



HC32L15 系列

32 位 ARM[®] Cortex[®]-M0+ 微控制器

用户手册

前 言

非常感谢大家对华大半导体产品的支持和信赖。

使用本系列产品前, 请系统阅读本手册和“数据手册”。

➤ 本手册的目的和对象读者

本手册主要介绍本系列的功能、操作事项和使用方法。对象读者为使用本系列实际开发产品的工程师。

本手册介绍外设功能的构成和操作说明, 但不包括该系列的规格说明。关于芯片规格, 详情参见其对应的“数据手册”。

➤ 样本程序和开发环境

华大半导体提供外设功能运行用的样本程序和本系列所需的开发环境说明。关于华大微控制器的运行规格和使用方法, 请联系本公司。

➤ 微控制器支持信息: <http://www.hdsc.com.cn/list.aspx?node=73&f=cn>

- 本手册的记载内容如有变动, 恕不另行通知。订购前建议用户咨询销售代表。
- 本手册记载的信息仅供参考, 诸如功能概要和应用电路示例, 旨在说明华大半导体器件的使用方法和操作示例; 对于建立在该基础上的器件使用, 华大半导体不保证器件的正常工作。如果用户根据该信息使用器件实行相关开发, 用户应承担因此引发的责任。基于上述信息的使用引起的任何损失, 华大半导体不承担任何责任。
- 本手册内的任何技术信息, 包括功能介绍和原理图, 不应理解为使用和执行任何知识产权的许可, 诸如专利权或著作权, 或华大半导体的其他权利或第三方权利, 华大半导体也不保证使用该信息不侵犯任何第三方知识产权或其他权利。因使用该信息引起的对第三方知识产权或其他权利的侵权行为, 华大半导体不承担任何责任。
- 本手册所介绍的产品旨在一般用途而设计、开发和制造, 包括但不限于一般的工业使用、通常办公使用、个人使用和家庭使用, 不旨在以下设计、开发和制造: 使用中伴随着致命风险或危险, 若不加以特别高度安全保障, 有可能导致对公众产生危害, 甚至直接死亡、人身伤害、严重物质损失或其他损失(即核设施的核反应控制、航空飞行控制、空中交通控制、公共交通控制、医用维系生命系统、核武器系统的导弹发射控制); 需要极高可靠性的应用领域(比如海底中转器和人造卫星)。注意上述领域内对使用该产品引起的用户和/或第三方的任何索赔或者损失, 华大半导体不承担任何责任。
- 半导体器件存在一定的故障发生概率。请用户对器件和设备采取冗余设计、消防设计、过电流等防护措施, 及其它异常操作防护措施等安全设计, 保证即使半导体器件发生故障的情况下, 也不会造成人身伤害、社会损害或者重大损失。
- 本手册内记载的公司名称和商标名称是各个公司的商标或注册商标。

目 录

前 言.....	2
目 录.....	3
1. 系统.....	15
1.1 总线架构	15
1.1.1 总线架构图	17
1.1.2 存储器架构	18
1.1.3 外围地址映射	19
1.2 Cortex-M0+ 架构	20
1.2.1 选型配置	22
1.3 模式	23
2. 时钟.....	24
2.1 概述	24
2.2 配置	25
2.2.1 源时钟	25
2.2.2 主时钟	25
2.2.3 内部总线时钟	25
2.2.4 其他时钟	26
2.3 操作	27
2.3.1 时钟模式选择	27
2.3.2 内部总线时钟分频控制	28
2.3.3 振荡稳定等待时间	28
2.3.4 中断源	30
2.4 时钟设定步骤举例	31
2.5 寄存器	33
2.5.1 系统时钟模式控制寄存器 (CTRL)	33
2.5.2 系统时钟模式状态寄存器 (CLK_STB)	34
2.5.3 基础时钟分频寄存器 (BC_DIV)	35
2.5.4 APB0 时钟分频寄存器 (APB0_CTRL)	36
2.5.5 APB1 时钟分频寄存器 (APB1_CTRL)	37
2.5.6 定时器时钟设置寄存器 (SWDG_CTRL)	37
2.5.7 时钟稳定等待时间寄存器 (CSWT)	39
2.5.8 调试断点监视定时器控制寄存器 (DBWDT_CTRL)	41
2.5.9 中断使能寄存器 (INT_CTRL)	41
2.5.10 中断状态寄存器 (INT_PEND)	42
2.5.11 中断清零寄存器 (INT_CLR)	43
2.6 注意事项	45
3. 外设时钟门控.....	46
3.1 概述	46
3.2 配置	47
3.3 操作	48
3.4 寄存器	49
3.4.1 外设时钟控制寄存器 0 (CLKEN0)	49
3.4.2 外设复位控制寄存器 0 (RSTCTL0)	50
3.4.3 外设时钟控制寄存器 1 (CLKEN1)	51
3.4.4 外设复位控制寄存器 1 (RSTCTL1)	52
3.4.5 外设时钟控制寄存器 2 (CLKEN2)	53

3.4.6	外设复位控制寄存器 2 (RSTCTL2)	54
3.5	注意事项	56
4.	时钟调校.....	57
4.1	概述	57
4.2	配置及框图	58
4.3	操作	59
4.4	设置流程	60
4.5	寄存器	64
4.5.1	内部高速 RC 时钟调校写保护寄存器 (HICR_TRMKEY)	64
4.5.2	内部高速 RC 时钟频率调校寄存器 (HICR_FTRM)	65
4.5.3	内部高速 RC 时钟输出频率选择寄存器 (HICR_FSEL)	66
4.5.4	内部高速 RC 时钟分频设置寄存器 (HICR_FDIV)	67
4.5.5	外部高速晶振时钟控制寄存器 (HSXT_PWRCTL)	68
4.5.6	外部低速晶振时钟控制寄存器 (LSXT_PWRCTL)	69
4.6	注意事项	70
5.	时钟监视器.....	71
5.1	概述	71
5.2	配置	72
5.2.1	时钟失效检测	72
5.2.2	时钟频率异常检测	73
5.3	操作	74
5.3.1	时钟失效检测	74
5.3.2	时钟频率异常检测	75
5.4	设置步骤	77
5.5	寄存器	79
5.5.1	时钟监视器控制寄存器 (FCM_CTL)	79
5.5.2	时钟监视器状态寄存器 (FCM_STR)	81
5.5.3	频率检测窗口上限设置寄存器 (AFDWH_CTL)	81
5.5.4	频率检测窗口下限设置寄存器 (AFDWL_CTL)	82
5.5.5	频率检测计数值寄存器 (AFDC_CTL)	82
5.6	注意事项	83
6.	复位.....	84
6.1	概述	84
6.2	配置	85
6.3	操作	86
6.3.1	复位源	86
6.3.2	MCU 内部复位	89
6.3.3	复位顺序	91
6.3.4	释放复位后的操作	92
6.4	寄存器	93
6.4.1	复位源寄存器 (STAT)	93
7.	低电压检测.....	96
7.1	概述	96
7.2	工作原理	96
7.2.1	原理框图	96
7.2.2	建立时间	96
7.2.3	响应时间	96
7.2.4	迟滞功能	98
7.3	操作流程	99

7.3.1	LVD 输出作为中断	99
7.3.2	LVD 输出作为复位	100
7.4	寄存器	101
7.4.1	LVD1 控制寄存器 1(LVD_CTL1)	101
7.4.2	LVD1 控制寄存器 2(LVD_CTL2)	102
7.4.3	LVD1 状态寄存器 1(LVD_STR).....	103
7.4.4	LVD1 中断清零寄存器 1(LVD_CLR).....	103
8.	低功耗模式.....	104
8.1	概述	104
8.1.1	Run 模式概述	104
8.1.2	Idle 模式概述	105
8.1.3	Standby 模式概述	106
8.1.4	Deep Standby 模式概述.....	106
8.2	状态转换图	107
8.3	Idle 模式相关操作	108
8.3.1	Idle 模式时钟工作状态	108
8.3.2	Idle 模式功能	108
8.4	Standby 模式相关操作	109
8.4.1	Standby 模式时钟工作状态	109
8.4.2	RTC 模式功能	109
8.4.3	STOP 模式功能	110
8.5	Deep Standby 模式相关操作	111
8.5.1	Deep Standby 模式时钟工作状态	111
8.5.2	Deep Standby 模式供电情况	111
8.5.3	Deep RTC 模式功能	112
8.5.4	Deep STOP 模式功能	112
8.6	低功耗模式设置与唤醒	114
8.6.1	各个模式唤醒源	114
8.6.2	Idle 模式设置与唤醒	114
8.6.3	RTC 模式设置与唤醒	115
8.6.4	STOP 模式设置与唤醒	115
8.6.5	Deep RTC 模式设置与唤醒	116
8.6.6	Deep STOP 模式设置与唤醒	116
8.7	寄存器	118
8.7.1	低功耗模式控制寄存器(LPM_CTL)	118
8.7.2	LDO 控制寄存器(LDO_CTL).....	120
8.7.3	时钟控制寄存器 1(CLK_CTL1)	121
8.7.4	时钟控制寄存器 2(CLK_CTL2)	121
8.7.5	Deep Standby 模式 RAM 控制寄存器(DS_RAMP)	122
8.7.6	唤醒复位源状态寄存器(WAKE_RSR)	122
8.7.7	唤醒中断源状态寄存器(WAKE_ISR)	123
8.7.8	唤醒中断源使能寄存器(WAKE_IER)	123
8.7.9	唤醒引脚中断高低电平设置寄存器(WAKE_LVR)	124
8.7.10	备份寄存器(BUR01-BUR16)	124
9.	电源结构和电源域.....	125
9.1	概述	125
9.2	电源结构	125
9.3	供电形式	126
9.4	电源域划分	127
9.4.1	VDD 电源域	128
9.4.2	VCORE 电源域	128

9.4.3	VCORE_CPU 电源域	129
9.4.4	VCORE_RAM 电源域	129
9.5	电源切换流程	130
9.5.1	PSW 初始状态	130
9.5.2	上电流程	130
9.5.3	切换到 Standby 模式	130
9.5.4	切换到 Deep Standby 模式	130
10.	中断控制器.....	131
10.1	概要	131
10.2	构成	132
10.3	异常和中断源向量	133
10.4	寄存器	135
10.4.1	IRQ_ST02 中断请求状态寄存器	136
10.4.2	IRQ00PEND 中断请求状态寄存器	137
10.4.3	IRQ01PEND 中断请求状态寄存器	138
10.4.4	IRQ02PEND 中断请求状态寄存器	139
10.4.5	IRQ04PEND 中断请求状态寄存器	140
10.4.6	IRQ05PEND 中断请求状态寄存器	141
10.4.7	IRQ06PEND 中断请求状态寄存器	142
10.4.8	IRQ07/09/11/15/17/19PEND 中断请求状态寄存器	143
10.4.9	IRQ08/10/12/16/18/20PEND 中断请求状态寄存器	144
10.4.10	IRQ24PEND 中断请求状态寄存器	145
10.4.11	IRQ25PEND 中断请求状态寄存器	146
10.4.12	IRQ26PEND 中断请求状态寄存器	147
10.4.13	IRQ27PEND 中断请求状态寄存器	148
10.4.14	IRQ31PEND 中断请求状态寄存器	149
10.5	使用注意事项	151
11.	外部中断和 NMI 控制部分.....	152
11.1	概要	152
11.2	框图	152
11.3	操作	153
11.3.1	外部中断控制部分的操作	153
11.3.2	NMI 控制部分的操作	155
11.3.3	唤醒 Stop 和 RTC 模式	156
11.4	寄存器	158
11.4.1	外部中断使能寄存器 (INTEN)	158
11.4.2	外部中断源寄存器 (INTSRC)	159
11.4.3	外部中断源清除寄存器 (INTCLR)	159
11.4.4	外部中断电平寄存器 (INTLVL)	160
11.4.5	非屏蔽中断寄存器 (NMISRC)	161
11.4.6	非屏蔽中断清除寄存器 (NMICLR)	161
12.	I/O 端口	163
12.1	概要	163
12.2	I/O 的结构、框图及操作说明	164
12.2.1	I/O 端口结构	164
12.2.2	I/O 端口的初始选择功能	167
12.2.3	重定位功能	167
12.2.4	FN_SEL 输出的固定优先级	168
12.3	设定顺序示例	170
12.3.1	I/O 端口的设置	170

12.4	寄存器	171
12.4.1	端口功能设定寄存器(FNx)	173
12.4.2	上拉设定寄存器(PUx)	174
12.4.3	端口方向设定寄存器(DIRx).....	175
12.4.4	端口输入数据寄存器(GETx).....	175
12.4.5	端口输出数据寄存器(SETx)	176
12.4.6	模拟输入设定寄存器(ANALOG).....	177
12.4.7	扩展功能引脚设定寄存器(FN_SELx)	178
12.4.8	端口开漏功能设定寄存器(ODR)	205
12.4.9	LVDI 模拟输入设置寄存器(LVDIE).....	206
12.4.10	特殊端口设定寄存器(SPS)	207
12.5	注意事项	208
13.	快速 GPIO	210
13.1	概要	210
13.2	配置	211
13.3	操作	214
13.4	寄存器	216
13.4.1	快速 GPIO 输入数据寄存器 (FGETx)	217
13.4.2	快速 GPIO 输出数据寄存器 (FSETx)	218
13.4.3	快速 GPIO 输出使能寄存器 (FSETEx)	219
13.5	位操作地址	220
14.	AES 硬件加速器	221
14.1	概述	221
14.2	操作	221
14.2.1	操作共同点	221
14.2.2	加密流程	221
14.2.3	标准解密流程	222
14.3	寄存器	224
14.3.1	AES 控制寄存器 (AES_CON)	225
14.3.2	AES 数据寄存器 (AES_DATA)	227
14.3.3	AES 密钥寄存器 (AES_KEY)	229
14.3.4	AES 随机数寄存器 (AES_RNG)	231
14.3.5	AES 时钟分频寄存器 (AES_DIV_EN)	232
14.4	启停时序	233
15.	调试接口	234
15.1	概要	234
15.2	引脚说明	234
15.2.1	调试引脚	234
15.2.2	引脚的初始功能	234
15.2.3	SW-DP 引脚的内部上拉	235
16.	微追踪缓存数据观察点和跟踪单元(MTB_DWT)	236
16.1	概述	236
16.2	架构框图	237
16.3	配置以及设定步骤示例	238
16.3.1	MTB_DWT 的配置	238
16.4	寄存器	240
16.4.1	MTB_DW 地址比较起始追踪寄存器 (CMP_ADDR_START).....	241
16.4.2	MTB_DWT 数据比较起始追踪寄存器 (CMP_DATA_START)	241
16.4.3	MTB_DWT 屏蔽数据比较起始追踪寄存器(CMP_MASK_START)	242

16.4.4	MTB_DWT 地址比较停止追踪寄存器(CMP_ADDR_STOP).....	243
16.4.5	MTB_DWT 数据比较停止追踪寄存器(CMP_DATA_STOP).....	243
16.4.6	MTB_DWT 屏蔽数据比较停止追踪寄存器 (CMP_MASK_STOP).....	244
16.4.7	MTB_DWT 功能寄存器 (FCT).....	245
16.4.8	外围 ID0 ~ 外围 ID7 寄存器 (PIDx).....	246
16.4.9	元件 ID0 ~ 元件 ID3 寄存器 (CIDx).....	247
17.	闪存.....	248
17.1	概况	248
17.2	模块图	248
17.3	闪存存储构成	249
17.4	操作	250
17.4.1	闪存操作	250
17.4.2	编程或擦除时闪存操作	254
17.4.3	闪存控制状态标志和中断	255
17.4.4	注意事项	255
17.5	寄存器	256
17.5.1	闪存操作控制寄存器 (FOCR)	256
17.5.2	闪存锁存寄存器 0(FLC0)	258
17.5.3	闪存锁存寄存器 1(FLC1)	258
17.5.4	闪存状态寄存器 (FSTR)	259
17.5.5	闪存等待周期和中断控制寄存器 FWCIICR	260
17.5.6	闪存 Tnvs 时间设置寄存器 Tnvs	261
17.5.7	闪存 Tprog 时间设置寄存器 Tprog	261
17.5.8	闪存 Tpgs 时间设置寄存器 Tpgs	261
17.5.9	闪存编程 Trcv 时间设置寄存器 Trcv_pgm	262
17.5.10	闪存页擦除 Trcv 时间设置寄存器 Trcv_pers	262
17.5.11	闪存片擦除 Trcv 时间设定寄存器 Trcv_cers	262
17.5.12	闪存页擦除 Terase 设定时间寄存器 Tpers	263
17.5.13	闪存片擦除 Terase 时间设定寄存器 Tcers	263
17.6	闪存安全概况	264
17.7	闪存安全保护功能	265
17.8	闪存编程	266
17.8.1	编程方式	266
17.8.2	通过 JTAG/SWD 编程	266
17.8.3	通过串行下载方式编程	266
17.8.4	用户闪存编程	269
18.	唯一识别码寄存器.....	270
18.1	概要	270
18.2	寄存器	270
18.2.1	唯一识别码寄存器 0 (UIDR0)	270
18.2.2	唯一识别码寄存器 1 (UIDR1)	271
18.2.3	唯一识别码寄存器 2 (UIDR2)	271
19.	监视定时器.....	272
19.1	概要	272
19.2	构成框图	273
19.3	操作说明	275
19.3.1	软件监视定时器	275
19.3.2	硬件监视定时器	278
19.3.3	软件监视定时器和硬件监视定时器的差异	280
19.4	设定顺序示例	281

19.5	操作示例	284
19.6	寄存器一览	287
19.6.1	软件监视定时器载入寄存器(WDT_LD)	288
19.6.2	软件监视定时器数值寄存器(WDT_VL)	289
19.6.3	软件监视定时器控制寄存器(WDT_CTL)	290
19.6.4	软件监视定时器清空寄存器(WDT_INTCLR)	292
19.6.5	软件监视定时器中断状态寄存器(WDT_STA)	292
19.6.6	软件监视定时器窗口监视模式控制寄存器(WDT_MDC)	293
19.6.7	软件监视定时器锁定寄存器(WDT_LCK)	294
19.6.8	硬件监视定时器载入寄存器(WDT_LD)	295
19.6.9	硬件监视定时器数值寄存器(WDT_VL)	296
19.6.10	硬件监视定时器控制寄存器(WDT_CTL)	297
19.6.11	硬件监视定时器清空寄存器(WDT_CLR)	298
19.6.12	硬件监视定时器中断状态寄存器(WDT_STA)	298
19.6.13	硬件监视定时器锁定寄存器(WDT_LCK)	299
19.7	注意事项	300
20.	双定时器	301
20.1	概要	301
20.2	构成	302
20.3	操作	303
20.3.1	定时器工作模式	303
20.3.2	初始状态	307
20.3.3	中断操作	307
20.4	设定步骤例	308
20.5	寄存器	310
20.5.1	载入寄存器(TIMERXLOAD) X=1 或 2	311
20.5.2	数值寄存器(TIMERXVALUE) X=1 或 2	312
20.5.3	控制寄存器(TIMERXCONTROL) X=1 或 2	313
20.5.4	中断清除寄存器(TIMERXINTCLR) X=1 或 2	315
20.5.5	中断状态寄存器(TIMERXRIS) X=1 或 2	315
20.5.6	屏蔽中断状态寄存器(TIMERXMIS) X=1 或 2	316
20.5.7	背景载入寄存器(TIMERXBGLOAD) X=1 或 2	317
21.	实时时钟 (RTC)	318
21.1	构成	318
21.2	RTC 复位	318
22.	RTC 计数模块	320
22.1	概述	320
22.2	框图	321
22.3	操作示例	323
22.4	RTC 计数模块复位	334
22.5	闰年修正	337
22.6	时间修改错误	339
22.7	寄存器	343
22.7.1	控制寄存器 1 (RTCCTL1)	344
22.7.2	控制寄存器 2 (RTCCTL2)	351
22.7.3	计数周期设置寄存器 (CCS)	353
22.7.4	日期寄存器 (DATE)	354
22.7.5	小时寄存器 (HOUR)	354
22.7.6	分钟寄存器 (MIN)	355
22.7.7	秒寄存器 (SEN)	355

22.7.8	年寄存器 (YEAR).....	356
22.7.9	月寄存器 (MON).....	356
22.7.10	星期寄存器 (WEEK).....	357
22.7.11	日闹钟寄存器 (ADATE).....	358
22.7.12	小时闹钟寄存器 (AHOUR).....	359
22.7.13	分闹钟寄存器 (AMIN)	360
22.7.14	秒闹钟寄存器 (ASEN).....	361
22.7.15	年闹钟寄存器 (AYEAR).....	362
22.7.16	月闹钟寄存器 (AMON).....	363
22.7.17	定时器设置寄存器 (TVAL).....	364
22.8	使用注意事项	365
23.	RTC 时钟控制模块.....	366
23.1	概述	366
23.2	框图	367
23.3	操作	369
23.3.1	频率校正	369
23.3.2	RTCCO 外部引脚输出	371
23.4	设置过程	372
23.5	寄存器	373
23.5.1	时钟选择寄存器 (SCLKI).....	374
23.5.2	时钟选择状态寄存器 (SCLKS).....	374
23.5.3	频率校正值设置寄存器(CALVAL)	375
23.5.4	频率校正使能寄存器 (CALEN)	375
23.5.5	分频系数设置寄存器 (CLKDIVX).....	376
23.5.6	分频输出使能寄存器 (CLKDIVC).....	377
23.5.7	频率校正周期寄存器 (CALCYC)	378
23.5.8	RTCCO 输出选择寄存器 (RTCODIV).....	379
24.	复合定时器 I/O 选择功能	380
24.1	概要	380
24.2	结构	382
24.3	I/O 模式	383
24.3.1	引脚	383
24.3.2	I/O 模式	385
24.4	寄存器	400
24.4.1	I/O 选择寄存器(CTSEL0123).....	400
24.4.2	I/O 选择寄存器(CTSEL4567).....	401
24.4.3	同时软件启动寄存器(CTSSSR)	402
25.	复合定时器.....	404
25.1	概要	404
25.2	框图	406
25.3	操作	409
25.4	32 位模式的操作	411
25.5	复合定时器中断	413
25.6	寄存器	414
25.7	注意事项	416
25.8	功能描述	418
25.8.1	PWM 定时器功能	418
25.8.2	PPG 定时器功能	429
25.8.3	重载定时器功能	441
25.8.4	PWC 定时器功能	451

26. 蜂鸣器.....	463
26.1 概述	463
26.2 配置框图	463
26.3 操作	464
26.4 设置程序实例	465
26.5 寄存器	466
26.5.1 蜂鸣器控制寄存器(BUZ_CTL)	466
26.5.2 蜂鸣器计数重载寄存器(BUZ_CNT)	467
26.6 使用注意事项	467
27. 带隙基准源(BGR).....	468
27.1 BGR 的概要	468
27.2 寄存器	468
27.2.1 寄存器列表	468
27.2.2 BGR 控制寄存器 (BGR_CTL)	469
28. 12 位 A/D 转换器.....	470
28.1 概要	470
28.2 架构框图	471
28.3 操作说明	472
28.3.1 A/D 转换操作	472
28.3.2 区间比较功能	474
28.4 设定步骤示例	478
28.4.1 单次采样转换设定步骤示例	478
28.4.2 连续采样转换设定步骤示例	479
28.4.3 区间比较功能设定步骤示例	480
28.4.4 设置采样转换时间	481
28.5 寄存器	482
28.5.1 A/D 控制寄存器 1(ADC_CTL1)	483
28.5.2 A/D 控制寄存器 2(ADC_CTL2)	485
28.5.3 A/D 控制寄存器 3(ADC_CTL3)	487
28.5.4 A/D 转换结果寄存器(ADC_RESULT)	488
28.5.5 A/D 区间比较控制寄存器 (ADCMPCTL)	489
28.5.6 A/D 区间比较上限阈值寄存器(ADCMPH)	491
28.5.7 A/D 区间比较下限阈值寄存器(ADCMPL)	492
28.5.8 A/D 区间比较阈值范围外标志寄存器(ADCMPF)	493
28.5.9 A/D 区间比较中断标志寄存器(ADCMPIF)	494
29. LCD 控制器.....	495
29.1 概述	495
29.2 配置	497
29.2.1 LCD 驱动电压产生单元	498
29.2.2 引脚	503
29.3 操作	504
29.3.1 LCD 驱动波形	505
29.3.2 显示数据存储	517
29.4 设置步骤图	518
29.5 寄存器	519
29.5.1 LCD 控制寄存器 0 (LCDCR0)	519
29.5.2 LCD 控制寄存器 1 (LCDCR1)	520
29.5.3 LCD 控制寄存器 2 (LCDCR2)	521
29.5.4 LCD 控制寄存器 3 (LCDCR3)	522

29.5.5	LCD 驱动电压控制寄存器 (LCDDVC)	522
29.5.6	LCD COM 输出使能寄存器 (LCDCE0)	523
29.5.7	LCD SEG 输出使能寄存器 (LCDSE0 ~ LCDSE4)	524
29.5.8	LCD 显示数据存储寄存器 (LCDRAM00 ~ LCDRAM39)	525
29.6	注意事项	525
30.	电压比较器.....	526
30.1	概要	526
30.2	配置	527
30.3	操作以及设定步骤示例	528
30.3.1	电压比较器的滤波响应时间	528
30.3.2	电压比较器迟滞功能	529
30.3.3	设定步骤示例	529
30.4	寄存器	531
30.4.1	控制寄存器列表	531
30.4.2	电压比较器控制寄存器-1(VC_CTL1)	532
30.4.3	电压比较器控制寄存器 2(VC_CTL2)	534
30.4.4	电压比较器控制寄存器 3(VC_CTL3)	536
31.	多功能串行通讯接口.....	537
32.	UART (异步串行接口).....	538
32.1	UART (异步串行接口)概要	538
32.2	中断	539
32.2.1	接收中断及标志设置时序	539
32.2.2	发送中断发生和标志设置时序	541
32.3	UART 工作模式	542
32.4	专用波特率发生器	548
32.5	波特率补偿及设置	553
32.6	工作模式 0 (异步正常模式)设置步骤和编程流程	555
32.7	工作模式 1 (异步多处理器模式)设置步骤和程序流程	557
32.8	UART(异步串行接口)寄存器	559
32.8.1	串行控制寄存器(CR)	560
32.8.2	串行模式寄存器(MR)	562
32.8.3	串行状态寄存器(SR)	564
32.8.4	接收数据寄存器/发送数据寄存器(RXDR/TXDR)	566
32.8.5	波特率发生器寄存器(BRS1,BRS0)	568
32.8.6	扩展通信控制寄存器 (ECR)	570
32.8.7	波特率补偿寄存器(COMP1,COMP0)	572
33.	SPI(串行外设接口)	573
33.1	SPI (串行外设接口)概要	573
33.2	中断	574
33.2.1	接收中断产生和标志位设置时序	575
33.2.2	发送中断产生和标志位设置时序	577
33.2.3	定时器中断产生和标志位设置时序	578
33.2.4	片选错误中断产生和标志位设置时序	579
33.3	操作	581
33.3.1	SPI 传输(I)	581
33.3.2	SPI 传输 (II)	590
33.4	串行定时器操作	599
33.5	串行片选操作	604
33.6	专用波特率发生器	609

33.6.1	波特率设置	609
33.6.2	串行外设接口 SPI 设置顺序和程序流程	612
33.7	SPI (串行外设接口) 的寄存器	613
33.7.1	串行控制寄存器 (CR)	615
33.7.2	串行模式寄存器(MR)	618
33.7.3	串行状态寄存器(SR)	621
33.7.4	扩展通信控制寄存器(ECR)	623
33.7.5	接收数据寄存器/发送数据寄存器(RXDR/TXDR)	625
33.7.6	串行辅助控制寄存器(EACR)	627
33.7.7	串行定时器寄存器(TMR)	630
33.7.8	串行定时器比较寄存器(TMC)	631
33.7.9	串行片选控制/状态寄存器(CSCR)	632
33.7.10	串行片选时间寄存器(CSTMG3-0)	636
33.7.11	串行片选格式寄存器(CSFR2-0)	639
33.7.12	发送字节数寄存器(TBYTE3-0)	647
33.7.13	波特率生成器寄存器(BRS1,BRS0)	648
33.8	SPI 使用注意事项	649
34.	I²C 接口 (I²C 通信控制接口)	650
34.1	I ² C 接口概要	650
34.2	I²C 接口操作	651
34.2.1	I ² C 接口中断	651
34.2.2	I ² C 总线操作	652
34.2.3	主机模式	653
34.2.4	从机模式	666
34.2.5	总线错误	669
34.3	专用波特率发生器	670
34.3.1	波特率选择	670
34.3.2	波特率计算方法	670
34.3.3	各总线时钟频率对应的重载值和波特率	671
34.3.4	重载计数器功能	671
34.3.5	计数开始	671
34.4	I²C 通信操作流程示例	672
34.5	I²C 接口寄存器	675
34.5.1	串行模式寄存器(MR)	676
34.5.2	I ² C 总线控制寄存器(BC)	678
34.5.3	I ² C 总线状态寄存器(BS)	684
34.5.4	串行状态寄存器(SR)	689
34.5.5	接收数据寄存器/发送数据寄存器(RXDR/TXDR)	691
34.5.6	I ² C 总线扩展控制寄存器(EBCR)	693
34.5.7	波特率发生器寄存器(BRS1,BRS0)	695
34.5.8	7 位从机地址屏蔽寄存器 (SAMSK)	696
34.5.9	7 位从机地址寄存器 (SA)	697
35.	SCI7816 主控器	698
35.1	简介	698
35.2	ETU 计数器的使用	699
35.3	SCI7816 通信速率设置	699
35.4	SCI7816 字符等待时间 (CWT)	700
35.5	SCI7816 块保护时间 (BGT)	700
35.6	起始位采样	701
35.7	软件操作流程	702
35.7.1	初始化配置	702

35.7.2	发送（中断方式）	702
35.7.3	发送（轮询方式）	703
35.7.4	接收（中断方式）	703
35.7.5	接收（轮询方式）	703
35.8	寄存器定义	704
35.8.1	SCI7816 控制寄存器(CON).....	705
35.8.2	SCI7816 模式配置寄存器(MODE).....	708
35.8.3	SCI7816 状态寄存器(STATUS).....	710
35.8.4	SCI7816 FIFO 状态寄存器(FIFO_STATUS).....	712
35.8.5	SCI7816 FIFO 深度寄存器(FIFO_DEPTH)	715
35.8.6	SCI7816 收发数据寄存器(DATA).....	715
35.8.7	SCI7816 校验数据寄存器(EDC_DATA)	716
35.8.8	SCI7816 波特率配置寄存器(SBDR).....	717
35.8.9	SCI7816 ETU 计数寄存器(ETU_CNT).....	717
35.8.10	SCI7816 强制高电平配置寄存器 0(F1_NUM).....	718
35.8.11	SCI7816 时钟复位寄存器(CLK_RST)	719
35.8.12	SCI7816 中断使能寄存器(INT_EN)	720
35.8.13	SCI7816 唤醒中断标志寄存器(WAKEUP_INT)	721
附录 1 寄存器映射		722
Flash I/F	723	
Unique ID.....	724	
Clock/Reset.....	725	
HWDG	727	
SWDG.....	728	
Dual Timer	729	
Composite Timer.....	730	
Composite timer I/O select	731	
12-bit ADC.....	732	
OSC_CTL	733	
EXTI	734	
IRQC	735	
LCDC	737	
GPIO	738	
LVD.....	744	
Low Power mode	745	
SPI.....	747	
I ² C	748	
UART	749	
RTC	750	
Peripheral clock gating.....	751	
Buzzer	752	
Voltage Comparator	752	
SCI7816	753	
AES	754	
MTB_DWT	755	
Fast GPIO.....	757	
附录 2 封装尺寸		759
版本记录 & 联系方式		763

1. 系统

1.1 总线架构

本节介绍本系列的总线架构。

本系列的总线使用 AHB 总线矩阵电路来实现多层总线。以下介绍主控和从动架构。

- 主控架构
 - Cortex-M0+ CPU(AHB-Lite 系统总线)
- 从动架构
 - 内部闪存
 - 内部 SRAM(MTB 共享)
 - AHB-AHB 总线架桥
 - AHB-APB 总线架桥(APB0, 1)

关于总线架构图，参考图 1.1.1-1。

特征

● RAM 架构

用户 SRAM 区域可以和 MTB SRAM 区域共享。两块区域的划分由用户配置决定。

● APB 扩展总线

APB1 外围总线是扩展 APB 总线(不包含 APB0)，而下列功能最初是基于 AMBA3.0 单独追加的(不包括 APB0)。

- 支持半字(16 位)访问和字节(8 位)访问。
- 可使用半字和字节访问所支持的寄存器。
- 关于所支持的寄存器，可参考“寄存器映射”。
- 增加了读-修改-写(RMW)信号。
- 位带操作时，通常要产生 HMASTLOCK 信号。
- 增加 RMW 信号是为了防止在位带操作时进行读-修改-写处理而错误清除无关标志。对应的标志在进行读-修改-写处理的读取操作时读“1”，忽略写“1”操作。这样可以避免标志在“读-修改-写”顺序的读取操作后被马上置位，却在下一次写操作时被错误地清零。
- 对应的标志和寄存器记述如下：“读-修改-写”时，与位值无关，读“1”。

- 优先级
 - 总线权优先级由轮询调度机制决定。
- 端模式
 - 这个系列采用小端模式。

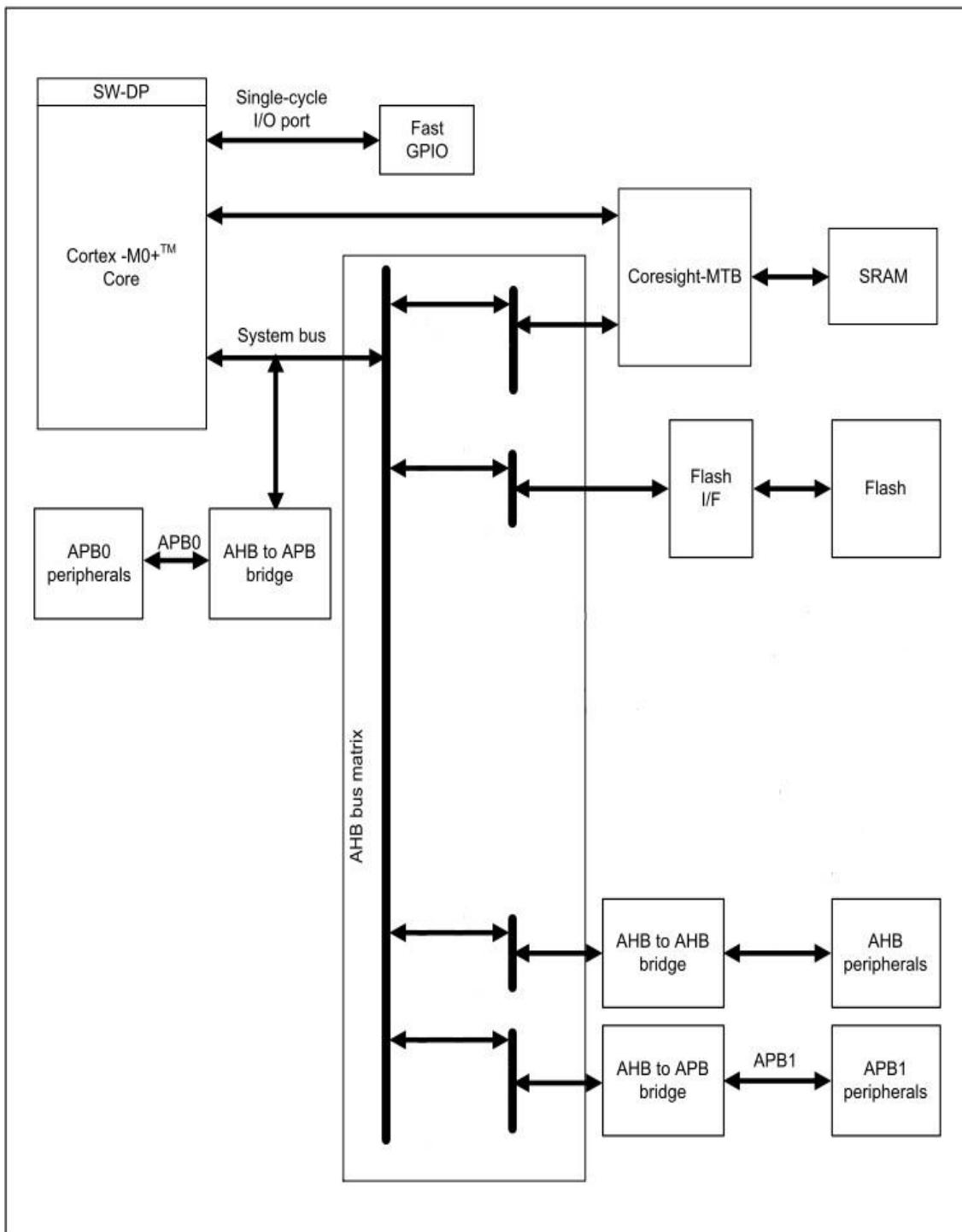
注意事项

- 针对禁止 RMW 操作的寄存器，不可进行位带操作。
- 通过软件而不是位带操作进行读-修改-写时，不输出 RMW 信号。因此，此时即使寄存器支持 RMW 过程，读取时仍然可以读标志值，注意在写入时，不要错误地清除无关标志。
- 关于位带操作，详情参见"Cortex-M3 技术参考手册"。此位带操作与 Cortex-M3 兼容。

1.1.1 总线架构图

图 1.1.1-1 是本系列的总线架构框图。

图 1.1.1-1 总线架构框图



1.1.2 存储器架构

本节介绍本系列的存储器架构。

本系列拥有 4GB 的地址空间。

存储器架构定义最大 1MB 的 FLASH 区、最大 512KB 的片上 SRAM 区。

1.1.3 外围地址映射

表 1.1.3-1 显示的是本系列的外设地址。

表 1.1.3-1 外设地址

起始地址	结束地址	总线	外设功能
0x4000_0000	0x4000_0FFF	AHB	闪存控制寄存器
0x4000_1000	0x4000_FFFF		保留
0x4001_0000	0x4001_0FFF	APB0	时钟/复位控制
0x4001_1000	0x4001_1FFF		硬件监视定时器
0x4001_2000	0x4001_2FFF		软件监视定时器
0x4001_3000	0x4001_4FFF		保留
0x4001_5000	0x4001_5FFF		双定时器
0x4001_6000	0x4001_FFFF		保留
0x4002_0000	0x4002_4FFF		保留
0x4002_5000	0x4002_5FFF	APB1	基本定时器
0x4002_6000	0x4002_6FFF		保留
0x4002_7000	0x4002_7FFF		A/D转换器
0x4002_8000	0x4002_DFFF		保留
0x4002_E000	0x4002_EFFF		时钟调校控制
0x4002_F000	0x4002_FFFF		保留
0x4003_0000	0x4003_0FFF		端口中断/不可屏蔽中断控制
0x4003_1000	0x4003_1FFF		中断监视单元
0x4003_2000	0x4003_2FFF		LCD控制器
0x4003_3000	0x4003_3FFF		端口控制
0x4003_4000	0x4003_4FFF		保留
0x4003_5000	0x4003_57FF		低电压检测/电源模式/带隙基准源
0x4003_5800	0x4003_6FFF		保留
0x4003_7000	0x4003_8FFF		多功能通讯串口
0x4003_9000	0x4003_AFFF		保留
0x4003_B000	0x4003_BFFF		实时时钟
0x4003_C000	0x4003_C0FF		SRAM奇偶校验
0x4003_C100	0x4003_C1FF		外设时钟门控
0x4003_C200	0x4003_C2FF		蜂鸣器
0x4003_C300	0x4003_C3FF		模拟电压比较器
0x4003_C400	0x4003_C7FF		保留
0x4003_C800	0x4003_C8FF		智能卡SCI7816主控
0x4003_C900	0x4003_FFFF		保留
0x4004_0000	0x4006_AFFF	AHB	保留
0x4006_B000	0x4006_BFFF		AES 加速器
0x4006_C000	0x41FF_FFFF		保留

1.2 Cortex-M0+ 架构

本节介绍本系列的内核架构。

注意：功能模块的架构根据产品不同而不同。详情请参考“1.2.1 选型配置”。

- Cortex-M0+ 内核

本系列搭载高性能 32 位处理器内核(ARM Cortex-M0+ 内核)。本外围手册没有详细描述 Cortex-M0+ 内核，详情参见“Cortex-M0+ 技术参考手册”。

- Cortex-M0+ 内核版本

关于 Cortex-M0+ 内核版本，参照“数据手册”。

- NVIC(嵌套向量中断控制器)

本系列可使用 1 个 NMI(不可屏蔽中断)和 32 个外围中断(IRQ0 ~ IRQ31)*¹。而且，中断优先级寄存器(从 0xE000E400 开始)由 2 位构成，并可以设定 4 级中断优先级。

关于外围中断，详情参见“中断”一章；关于 NMI 操作，参见“外部中断及 NMI 控制部分”一章。

NMI 引脚可以兼用作其他功能。复位后的初始值设定为通用口，NMI 输入被屏蔽了，可通过端口设定来使能 NMI 功能。详情参见“I/O 端口”一章和“外部中断及 NMI 控制部分”一章。

*¹：“Cortex-M0+ 技术参考手册”定义的异常类型：将 IRQ 视为外部中断。本手册为了区分 IRQ 和来自外部终端的中断，也定义了异常类型：将 IRQ 定义为外围中断。

当高优先级的中断在处理过程中，外设产生的中断会在 NVIC 内部等待。当取消 NVIC 内部等待的中断请求时，清除外设中断请求，通过清除 NVIC 内部中断等待寄存器（从 0xE000E280 开始）来清除 NVIC 内部的等待中断请求。

- SysTick 定时器

SysTick 定时器是系统定时器，被集成在 NVIC 中，可用于操作系统的任务管理。

本产品通过把 HCLK 八分频生成 STCLK，并且设置 SysTick 校正值寄存器(0xE000E01C)如下所示：

[bit31]: NOREF = 0

[bit30]: SKEW = 1

[bit23:0]: TENMS = 0x00C350 (50000)*¹

*¹ :若 HCLK 的 1/8 时钟输入到 STCLK 且 HCLK 的频率为 40MHz(1/8 时频率 为 5MHz)时，TENMS 的值设定为对应 10ms 的值。

因为时钟控制部分的 HCLK 可能变为其它频率，所以 TENMS 的值不一定始终是 10ms。所以，需根据 HCLK 的频率计算合适的中断时序。

- DWT(数据观察点与跟踪)

本系列搭载 DWT，用作调试功能。

DWT 有 2 个比较器可以用作数据地址匹配和指令地址匹配。

- BPU(断点单元)

BPU 提供指令读取时的断点支持。

- MTB(微追踪模块)

本系列搭载 Cortex-M0+ 处理器的一个可选部分-MTB，支持指令追踪。

MTB_DWT 控制追踪的起始。

- SW-DP

本系列搭载 SW-DP，支持串行线协议。

- ROM 表

ROM 表为外部的调试工具提供调试元件的地址信息。

- 单周期 I/O

本系列搭载单周期 I/O 用于高速访问。

1.2.1 选型配置

表 1.2.1-1 对应本系列 Cortex-M0+内核选型配置。

表 1.2.1-1 选型配置表

功能	
中断	32
端模式	小端
系统定时器	支持
数据观察点与跟踪比较器	2
断点比较器	4
中止调试	支持
乘法器	快速
单周期 I/O	支持
唤醒中断控制器	不支持
向量表位移寄存器	支持
非特权级/特权级	不支持
存储器保护单元	不支持
复位所有寄存器	支持
取指位宽	32
微追踪模块	支持
调试接口	仅串行线协议
多级串行线调试	不支持

1.3 模式

本节介绍工作模式。

本产品线使用如下工作模式。

- 用户模式

- 内部 ROM(Flash)启动: CPU 从 Flash 处取得复位向量, 启动运行。

- 串行编程器模式

- 使能 Flash 串行编程。

*: 关于该模式, 详情参见"闪存编程手册"。

上电复位、RSTB 端子输入复位完成时, 工作模式也就确定了。

*: 关于功耗控制模式和时钟选择模式, 参见"低功耗模式"及"时钟"一章。

工作模式设定方法

工作模式的设定根据 MODE 引脚输入进行。

MODE	功能
0	用户模式内部 ROM(Flash)启动
1	串行编程器模式

启动顺序

启动过程中决定工作模式的过程如下:

- 1) MODE 引脚的采样

根据 MODE 引脚输入设定工作模式。接通电源复位、RSTB 端子输入复位时对这些输入进行采样。

在复位信号释放之前确定 MODE 引脚的输入。

- 2) 工作模式的确定和模式数据的保持

复位时采样的 MODE 值将一直保持不变, 直至下次复位到来。

保持的 MODE 决定工作模式。因此, 复位解除后无论 MODE 如何变化, 对工作模式均无影响。

2. 时钟

2.1 概述

本节为时钟产生单元的概述。

时钟产生单元为 MCU 提供不同类型的时钟。

- 源时钟是 MCU 内部 RC 时钟和外部时钟的总称。源时钟有以下四种类型：
 - 外部高速时钟 (HSXT)
 - 外部低速时钟 (LSXT)
 - 内部高速 RC 时钟 (HICR)
 - 内部低速 RC 时钟 (LICR)
- 主时钟是 MCU 总线的工作时钟源，可以是四种源时钟的一种。

基础时钟是主时钟的分频时钟。基础时钟再分频可以得到各个总线时钟。

在本章，基础时钟和总线时钟统称为内部总线时钟，内部总线时钟包括以下三种类型：

- 基础时钟 (HCLK)
- APB0 总线时钟 (PCLK0)
- APB1 总线时钟 (PCLK1)

以下是时钟产生单元的基本功能：

- 可以设置外部高速时钟的稳定时间，时钟稳定后可以产生中断
- 可以设置外部低速时钟的稳定时间，时钟稳定后可以产生中断
- 外部高速时钟和外部低速时钟发生异常可以复位或产生中断
- 可以选择主时钟来源
- 可以设置内部总线时钟的分频比例
- 可以停止提供 APB1 总线时钟
- 可以设置软件监视定时器计数时钟的分频比例
- 可以停止提供硬件监视定时器和软件监视定时器的计数时钟
- 调试模式下，硬件监视定时器和软件监视定时器可以停止工作

2.2 配置

本节将介绍时钟产生单元的配置。

2.2.1 源时钟

源时钟是 MCU 内部 RC 时钟和外部时钟的总称。源时钟有以下四种类型。

- 外部高速时钟（HSXT）
 - HSXT 可以由外部高速时钟引脚（XHI, XHO）连接晶振产生，也可以由外部高速时钟引脚 XHI 直接输入时钟产生。
- 外部低速时钟（LSXT）
 - LSXT 可以由外部低速时钟引脚（XLI, XLO）连接晶振产生，也可以由外部低速时钟引脚 XLI 直接输入时钟产生。
- 内部高速 RC 时钟（HICR）
 - HICR 由内部高速 RC 振荡器产生。HICR 的输出频率可以配置为 2MHz, 4MHz, 8MHz 和 16MHz，默认值为 4MHz。
- 内部低速 RC 时钟（LICR）
 - LICR 由内部低速 RC 振荡器产生。

2.2.2 主时钟

主时钟是内部所有总线的时钟来源。主时钟由源时钟选择得到，可以是四种源时钟的一种。

2.2.3 内部总线时钟

下面将介绍 MCU 内部产生的总线时钟。

- 基本时钟（HCLK, FCLK）
 - HCLK 和 FCLK 统称为基本时钟。HCLK 和 FCLK 均作为 CPU 的时钟源。
 - HCLK 是 AHB 总线的时钟。
 - CPU 使用的 HCLK 和 FCLK 统称为基础时钟，基础时钟由主时钟分频得到，分频比例从 1/1 到 1/16。
 - 在 RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下，基础时钟自动停止。
 - 在空闲模式下，HCLK 停止提供给 CPU，但此时 FCLK 不停止。
- APB0 总线时钟（PCLK0）

- PCLK0 是 APB0 总线上的外设时钟。
 - PCLK0 由基础时钟分频得到，分频比例从 1/1 到 1/8。
 - 在 RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下，PCLK0 自动停止。
- APB1 总线时钟 (PCLK1)
- PCLK1 是 APB1 总线上的外设时钟。
 - PCLK1 由基础时钟分频得到，分频比例从 1/1 到 1/8。
 - 在 RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下，PCLK1 自动停止。
 - PCLK1 可以通过设置寄存器停止。

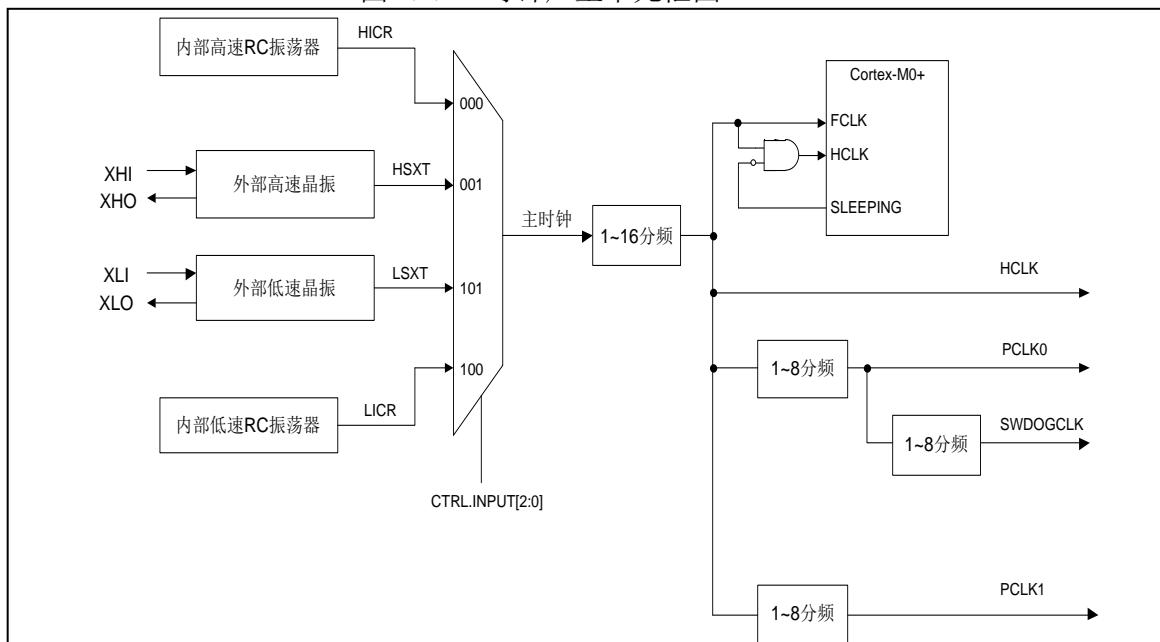
2.2.4 其他时钟

软件监视定时器计数时钟 (SWDOGCLK)，SWDOGCLK 由 PCLK0 分频得到，分频比例从 1/1 到 1/8。

在 RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下，SWDOGCLK 自动停止。

图 2.2.4-1 是时钟产生单元的系统框图。

图 2.2.4-1 时钟产生单元框图



2.3 操作

2.3.1 时钟模式选择

时钟模式定义

MCU 的时钟模式由系统时钟模式控制寄存器所选的源时钟确定。时钟模式有 HSXT 时钟模式、LSXT 时钟模式、HICR 时钟模式和 LICR 时钟模式四种。

- HSXT 时钟模式

HSXT 时钟模式下，外部高速时钟（HSXT）作为主时钟使用。是 CPU 和大部分外设总线时钟。

在这种时钟模式下，外部低速时钟（LSXT）和内部高速时钟（HICR）的状态，取决于系统时钟控制寄存器 CTRL 的 SUB_EN 位和 HC_EN 位。内部低速时钟（LICR）的状态不能通过寄存器控制。

- LSXT 时钟模式

LSXT 时钟模式下，外部低速时钟（LSXT）作为主时钟使用。是 CPU 和大部分外设的总线时钟。

在这种时钟模式下，外部高速时钟（HSXT）和内部高速时钟（HICR）由硬件控制自动停止工作。内部低速时钟（LICR）的状态不能通过寄存器控制。

- HICR 时钟模式

HICR 时钟模式下，内部高速时钟（HICR）作为主时钟使用。是 CPU 和大部分外设的总线时钟。

在这种时钟模式下，外部低速时钟（LSXT）和外部高速时钟（HSXT）的状态，取决于系统时钟控制寄存器 CTRL 的 SUB_EN 位和 MAIN_EN 位。内部低速时钟（LICR）的状态不能通过寄存器控制。

- LICR 时钟模式

LICR 时钟模式下，内部低速时钟（LICR）作为主时钟使用。是 CPU 和大部分外设总线时钟。

在这种时钟模式下，外部高速时钟（HSXT）和内部高速时钟（HICR）由硬件控制自动停止工作，外部低速时钟（LSXT）的状态，取决于系统时钟控制寄存器 CTRL 的 SUB_EN 位。内部低速时钟（LICR）的状态不能通过寄存器控制。

2.3.2 内部总线时钟分频控制

内部总线时钟各自独立，可分别设定基本时钟的分频比例，来选择各电路的最佳工作频率。

内部总线时钟的最大频率依据产品而不同，请参考产品的“数据手册”。

要设定内部总线时钟的分频比例，可以使用基本时钟分频寄存器，APB0 分频寄存器和 APB1 分频寄存器。详细描述请参考本章第 5 节“寄存器”。

总线时钟分频设定

- 软件复位不清零分频设定值，分频比例仍维持软件复位发生前的状态。
- 将主时钟从初始状态变更为较高速的源时钟之前，务必设定分频比例。
- 分频后的时钟频率超过内部总线的最大工作频率时，不能保证正常工作。

2.3.3 振荡稳定等待时间

时钟源需要经过一段振荡稳定等待时间后才能处于稳定运行状态。在振荡稳定等待时间内，内部及外部时钟停止供给，直到时钟稳定等待时间寄存器设定的稳定等待时间结束。当振荡稳定等待时间结束，对应的振荡器准备完毕，时钟便可作为主时钟使用。

振荡稳定等待的计数方法

外部时钟（HSXT/LSXT）的振荡稳定等待计数有两种方法：

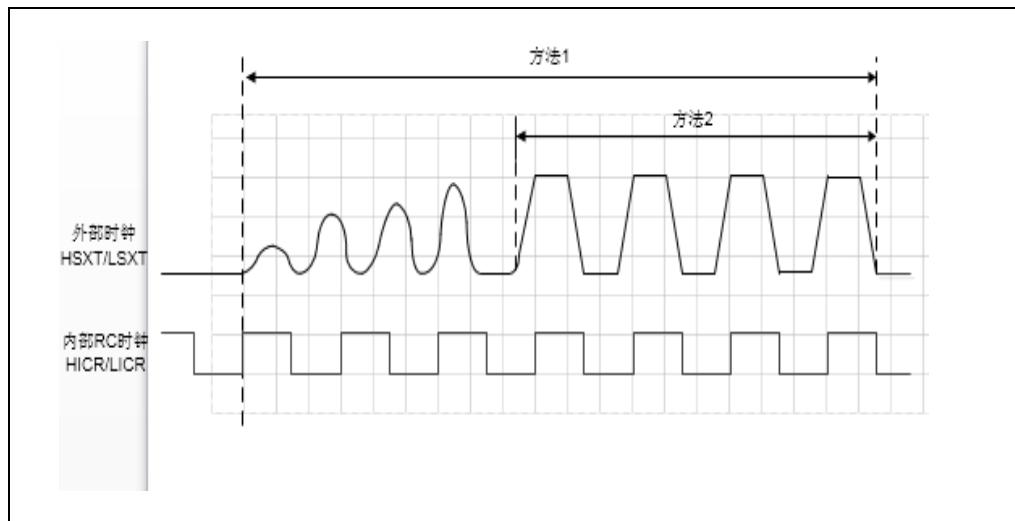
- 1) 内部 RC 时钟 (HICR/LICR) 作为计数时钟，经过一定的时间，认为外部时钟 (HSXT/LSXT) 处于稳定状态。
- 2) 外部时钟 (HSXT/LSXT) 作为计数时钟，经过一定数量的时钟上升沿，认为外部时钟 (HSXT/LSXT) 处于稳定状态。

振荡稳定等待计数方法，取决于时钟监视功能的开启与关闭。

- 外部高速时钟频率异常检测功能 (AFD) 关闭，同时外部高速时钟失效检测功能(MCFD) 关闭。此时选择方法 2。
- 外部高速时钟频率异常检测功能 (AFD) 打开，或者外部高速时钟失效检测功能(MCFD) 打开。此时选择方法 1。
- 外部低速时钟失效检测功能(SCFD)关闭，此时选择方法 2。
- 外部低速时钟失效检测功能(SCFD)打开，此时选择方法 1。

以下是两种计数方法的示意图。

图 2.3.3-1 振荡稳定等待的计数方法



振荡稳定等待时间的设定

- 外部高速时钟 (HSXT)

时钟振荡稳定等待时间寄存器的 MAINWT 位 (CSWT: MAINWT) 用来设定 HSXT 的振荡稳定等待时间。

- 外部低速时钟 (LSXT)

时钟振荡稳定等待时间寄存器的 SUBWT 位 (CSWT: SUBWT) 用来设定 LSXT 的振荡稳定等待时间。

需要等待振荡稳定的情况

- 软件使能振荡后

将系统时钟模式控制寄存器 (CTRL) 的 MAIN_EN、SUB_EN 和 HC_EN 位使能后，各振荡器开始振荡稳定等待过程。

- 使用 RTC 中断或外部中断退出 RTC 模式时

系统时钟将回到 RTC 模式之前的时钟模式，因 RTC 模式下停止的时钟源，硬件自动开始振荡稳定等待过程。

- 使用外部中断退出 STOP 模式时

系统时钟将回到 STOP 模式之前的时钟模式，因 STOP 模式下所有时钟源停止工作，硬件自动开始振荡稳定等待过程。

注意：

- 请在使能时钟前变更各振荡稳定等待时间的设定值。
- 软件复位后不等待振荡稳定等待时间。
- 振荡稳定计数完成后，振荡稳定等待完成标志置起，此标志与振荡器的实际状态无关。如

果振荡稳定等待时间设定的太短，有可能在振荡器的振荡稳定之前，振荡稳定等待时间就结束了。

- HSXT 和 LSXT 的稳定等待时间与晶振的种类(水晶、陶瓷等)有关，针对所用晶振，应选择适当的振荡稳定等待时间。

振荡稳定等待的优先级

在某个时钟振荡稳定等待时间内启动其他时钟，硬件将按照一定的优先级依次进行振荡稳定等待。当优先级高的时钟振荡稳定完成后，再启动优先级低的时钟振荡稳定等待。

不同时钟模式下，各时钟的优先级如下描述。

- HSXT 时钟模式

- MCFD 或 AFD 使能

LICR -> LSXT -> HICR -> HSXT

- MCFD 和 AFD 都禁止

LICR -> LSXT -> HSXT -> HICR

- LSXT 时钟模式

LICR -> LSXT

- HICR 时钟模式

LICR -> LSXT -> HICR -> HSXT

- LICR 时钟模式

LICR -> LSXT

2.3.4 中断源

时钟产生单元有以下中断源。

- HSXT 振荡稳定等待完成中断

HSXT 振荡稳定等待完成后，中断发生。

- LSXT 振荡稳定等待完成中断

LSXT 振荡稳定等待完成后，中断发生。

- AFD 中断（HSXT 频率异常检测）

使能 HSXT 频率异常检测功能，当检测到 HSXT 频率不在设定范围之内时发生中断。

- MCFD 中断（HSXT 失效检测）

使能 HSXT 失效检测功能，当检测到 HSXT 停止振荡时发生中断。

- SCFD 中断（LSXT 失效检测）

使能 LSXT 失效检测功能，当检测到 LSXT 停止振荡时发生中断。

2.4 时钟设定步骤举例

图 2.4-1 上电→HICR 运行模式→其他时钟模式

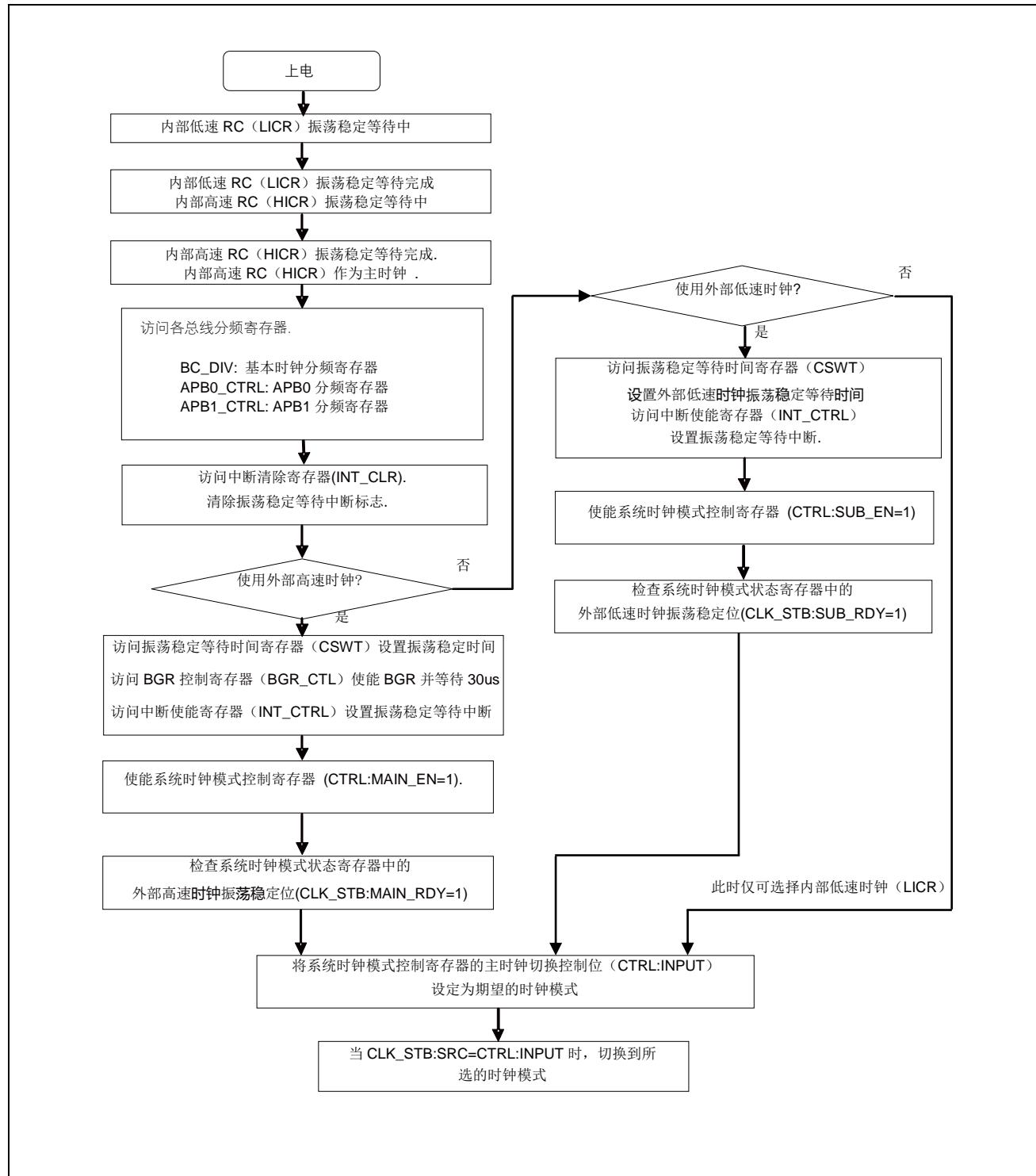
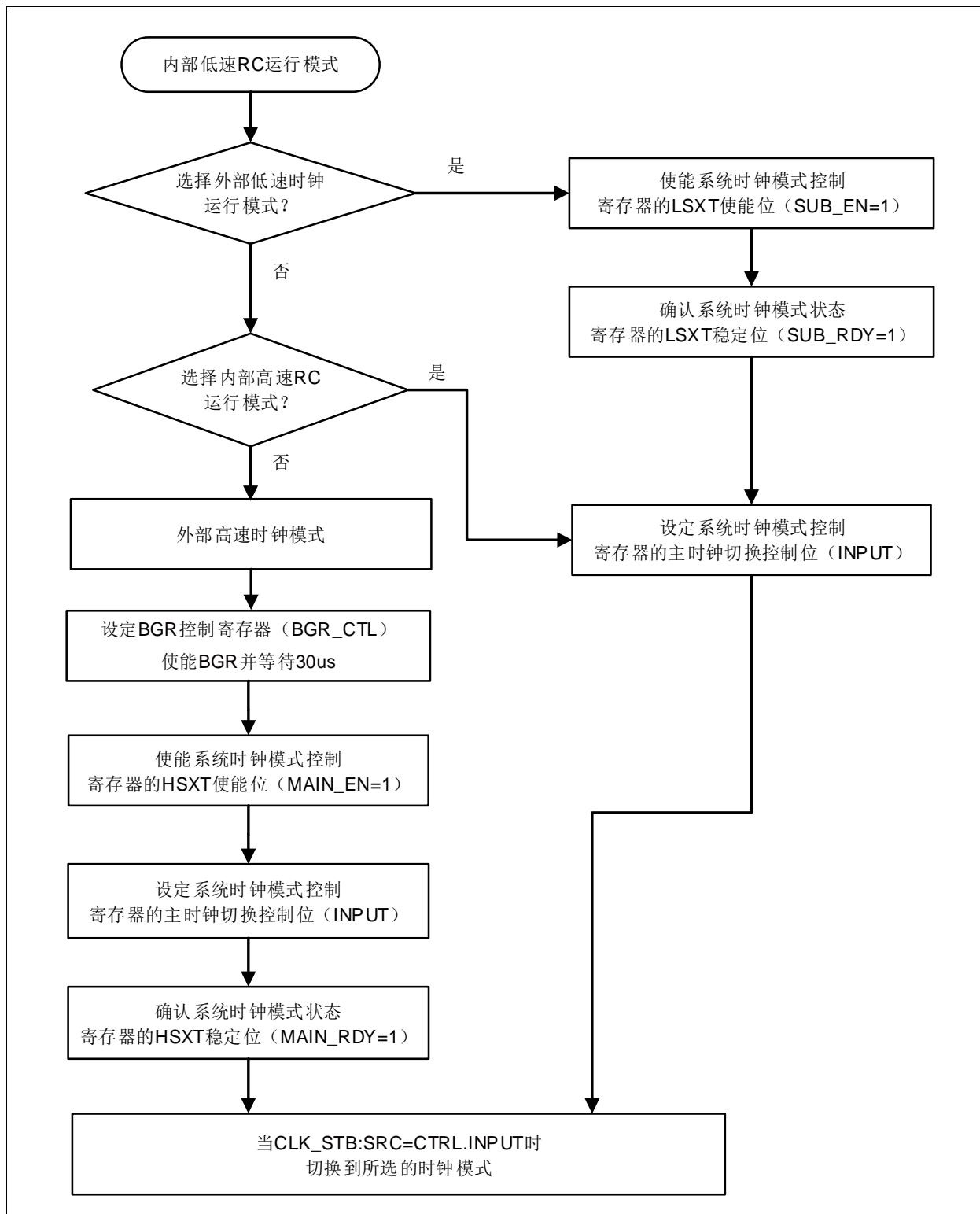


图 2.4-2 LICR 运行模式→其他时钟模式



注意：

- 图 2.4-2 中没有描述振荡稳定等待时间，中断和总线时钟分频比例的寄存器设置，请在此流程前做好相应配置。

- 在低速时钟模式 (LSXT 和 LICR) 下, 高速时钟 (HSXT 和 HICR) 由硬件控制自动停止, 此时使能高速时钟 (MAIN_EN=1 或 HC_EN=1) 不会启动相应时钟。
- 如果设置的外部时钟 (LSXT 和 HSXT) 振荡稳定等待时间较短, 在各振荡器稳定前振荡稳定等待时间已经结束时, 时钟监视模块可能会复位系统。
- 在高速时钟模式(HICR)下的不同频率时钟切换时, 建议先把时钟切换到低速时钟(LICR), 然后再切换回高速时钟模式(HICR).如从 HICR 4MHz 切换到 HICR 16MHz 时, 先切换到 LICR, 然后配置 HICR 到 16MHz, 等待稳定时间, 最后切换时钟到 HICR 16MHz.

2.5 寄存器

表 2.5-1 时钟产生单元的寄存器列表

缩写	寄存器名称	章节
CTRL	系统时钟模式控制寄存器	2.5.1
CLK_STB	系统时钟模式状态寄存器	2.5.2
BC_DIV	基础时钟分频寄存器	2.5.3
APB0_CTRL	APB0时钟分频寄存器	2.5.4
APB1_CTRL	APB1时钟分频寄存器	2.5.5
SWDG_CTRL	定时器时钟设置寄存器	2.5.6
C SWT	时钟稳定等待时间寄存器	2.5.7
DBWDT_CTRL	调试断点监视定时器控制寄存器	2.5.8
INT_CTRL	中断使能寄存器	2.5.9
INT_PEND	中断状态寄存器	2.5.10
INT_CLR	中断清零寄存器	2.5.11

2.5.1 系统时钟模式控制寄存器 (CTRL)

系统时钟模式控制寄存器可以选择当前主时钟, 使能或禁止时钟振荡。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	INPUT[2:0]	保留	SUB_EN	保留	MAIN_EN	HC_EN		
属性	R/W	-	R/W	-	R/W	R/W		
初始值	000	-	0	-	0	1		

寄存器功能

[bit7:5] INPUT[2:0]: 主时钟切换控制位

bit7	bit6	bit5	说明
0	0	0	内部高速RC时钟 (初始值)

0	0	1	外部高速时钟
1	0	0	内部低速RC时钟
1	0	1	外部低速时钟
其他			禁止设定

[bit4] 保留位，读数据为 0，写无效。

[bit3] SUB_EN：外部低速时钟振荡使能位

bit	说明
0	禁止外部低速时钟振荡（初始值）
1	使能外部低速时钟振荡

[bit2] 保留位，读数据为 0，写无效。

[bit1] MAIN_EN：外部高速时钟振荡使能位

bit	说明
0	禁止外部高速时钟振荡（初始值）
1	使能外部高速时钟振荡

[bit0] HC_EN：内部高速 RC 时钟振荡使能位

bit	说明
0	禁止内部高速RC时钟振荡
1	使能内部高速RC时钟振荡（初始值）

注意：

- 软件复位不会复位该寄存器。
- 变更时钟模式时，请先使能目标时钟振荡，再写主时钟切换控制位 INPUT。
- 打开 HSXT 时钟频率异常检测功能，或打开 HSXT 时钟失效检测功能后，内部高速 RC 时钟自动使能，此时与 HC_EN 位的设置无关。
- 当时钟处于振荡稳定等待状态时，不要打开其他时钟。
- 使能外部高速时钟振荡前，先访问 BGR 控制寄存器 BGR_CTL 使能 BGR 并等待 BGR 稳定。

2.5.2 系统时钟模式状态寄存器（CLK_STB）

系统时钟模式状态寄存器显示当前主时钟的选择状态和各时钟振荡稳定等待状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	SRC[2:0]	保留	SUB_RDY	保留	MAIN_RDY	HC_RDY		

属性	R	-	R	-	R	R
初始值	000	-	0	-	0	1

寄存器功能

[bit7:5] SRC[2:0]: 主时钟状态位

bit7	bit6	bit5	说明
0	0	0	内部高速RC时钟（初始值）
0	0	1	外部高速时钟
1	0	0	内部低速RC时钟
1	0	1	外部低速时钟
其他		禁止设定	

[bit4] 保留位，数据为 0，写无效。

[bit3] SUB_RDY: 外部低速时钟振荡稳定位

bit	说明
0	稳定等待中或振荡停止状态（初始值）
1	稳定状态

[bit2] 保留位，读数据为 0，写无效。

[bit1] MAIN_RDY: 外部高速时钟振荡稳定位

bit	说明
0	稳定等待中或振荡停止状态（初始值）
1	稳定状态

[bit0] HC_RDY: 内部高速 RC 时钟振荡稳定位

bit	说明
0	稳定等待中或振荡停止状态
1	稳定状态（初始值）

注意：

- 软件复位不会复位该寄存器。

2.5.3 基础时钟分频寄存器 (BC_DIV)

基础时钟分频寄存器用于设置基础时钟的分频比例。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留							DIV
属性			-			-	R/W	

初始值			-			-	3'b000	
-----	--	--	---	--	--	---	--------	--

寄存器功能

[bit7:3]保留位，读数据为0，写无效。

[bit2:0] DIV[2:0]: 基础时钟分频比例设置位

Bit2	Bit1	Bit0	说明
0	0	0	1/1 (初始值)
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	0	0	1/6
1	0	1	1/8
1	1	0	1/16
1	1	1	禁止设定

注意：

- 软件复位不会复位该寄存器。

2.5.4 APB0 时钟分频寄存器 (APB0_CTRL)

APB0 时钟分频寄存器用于设置 APB0 总线的分频比例。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留						DIV	
属性							R/W	
初始值							2'b00	

寄存器功能

[bit7:2]保留位，读数据为0，写无效。

[bit1:0] DIV[1:0]: APB0 时钟分频比例设置位

Bit1	Bit0	说明
0	0	1/1 (初始值)
0	1	1/2
1	0	1/4
1	1	1/8

注意：

- 软件复位不会复位该寄存器。

2.5.5 APB1 时钟分频寄存器 (APB1_CTRL)

APB1 时钟分频寄存器用于设置 APB1 总线的时钟使能，总线复位和分频比例。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	EN	保留		RST	保留		DIV	
属性	R/W	-		R/W	-		R/W	
初始值	1	-		0	-		2'b00	

寄存器功能

[bit7] EN: APB1 时钟使能位

bit	说明
0	禁止APB1时钟PCLK1
1	使能APB1时钟PCLK1 (初始值)

[bit6:5] 保留位，读数据为 0，写无效。

[bit4] RST: APB1 复位控制位

bit	说明
0	不复位APB1总线 (初始值)
1	复位APB1总线

[bit3:2] 保留位，读数据为 0，写无效。

[bit1:0] DIV[1:0]: APB1 时钟分频比例设置位

Bit1	Bit0	说明
0	0	1/1 (初始值)
0	1	1/2
1	0	1/4
1	1	1/8

注意：

- 软件复位不会复位该寄存器。

2.5.6 定时器时钟设置寄存器 (SWDG_CTRL)

定时器时钟设置寄存器用于设置双定时器的时钟使能，软/硬件监视定时器的时钟使能和软件监视定时器的时钟分频比例。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留	DTCKE	HWCKE	SWCKE	保留	DIV		

属性	-	R/W	R/W	R/W	-	R/W
初始值	-	1	1	1	-	2'b00

寄存器功能

[bit7]保留位，读数据为0，写无效。

[bit6] DTCKE: 双定时器时钟使能位

bit	说明
0	禁止双定时器的总线访问时钟
1	使能双定时器的总线访问时钟（初始值）

[bit5] HWCKE: 硬件监视定时器时钟使能位

bit	说明
0	禁止硬件监视定时器的总线访问时钟
1	使能硬件监视定时器的总线访问时钟（初始值）

[bit4] SWCKE: 软件监视定时器时钟使能位

bit	说明
0	禁止软件监视定时器的总线访问时钟
1	使能软件监视定时器的总线访问时钟（初始值）

[bit3:2]保留位，读数据为0，写无效。

[bit1:0] DIV[1:0]: 软件监视定时器计数时钟分频比例设置位

Bit1	Bit0	说明
0	0	PCLK0的1/1分频（初始值）
0	1	PCLK0的1/2分频
1	0	PCLK0的1/4分频
1	1	PCLK0的1/8分频

注意：

- 软件复位不会复位该寄存器。

2.5.7 时钟稳定等待时间寄存器（CSWT）

时钟稳定等待时间寄存器用于设置外部高速时钟（HSXT）和外部低速时钟（LSXT）的稳定等待时间。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	SUBWT					MAINWT		
属性	R/W					R/W		
初始值	4'b0000					4'b0000		

寄存器功能

[bit7:4] SUBWT[3:0]: LSXT 的稳定等待时间设置位

Bit7	Bit6	Bit5	Bit4	说明	LICR=32kHz时 等待时间
0	0	0	0	22个LICR或LSXT周期(初始值)	约0.12ms
0	0	0	1	24个LICR或LSXT周期	约0.5ms
0	0	1	0	26个LICR或LSXT周期	约2ms
0	0	1	1	28个LICR或LSXT周期	约8ms
0	1	0	0	29个LICR或LSXT周期	约16ms
0	1	0	1	210个LICR或LSXT周期	约32ms
0	1	1	0	211个LICR或LSXT周期	约64ms
0	1	1	1	212个LICR或LSXT周期	约128ms
1	0	0	0	213个LICR或LSXT周期	约256ms
1	0	0	1	214个LICR或LSXT周期	约512ms
1	0	1	0	215个LICR或LSXT周期	约1s
1	0	1	1	216个LICR或LSXT周期	约2s
1	1	0	0	217个LICR或LSXT周期	约4s
1	1	0	1	218个LICR或LSXT周期	约8s
1	1	1	0	219个LICR或LSXT周期	约16s
1	1	1	1	220个LICR或LSXT周期	约32s

[bit3:0] MAINWT[3:0]: HSXT 的稳定等待时间设置位

Bit3	Bit2	Bit1	Bit0	说明	HICR=4MHz时 等待时间
0	0	0	0	28个HICR或HSXT周期（初始值）	约64us
0	0	0	1	210个HICR或HSXT周期	约256us
0	0	1	0	212个HICR或HSXT周期	约1ms
0	0	1	1	214个HICR或HSXT周期	约4.1ms
0	1	0	0	215个HICR或HSXT周期	约8.2ms
0	1	0	1	216个HICR或HSXT周期	约16.4ms
0	1	1	0	217个HICR或HSXT周期	约32.8ms
0	1	1	1	218个HICR或HSXT周期	约65.6ms
1	0	0	0	219个HICR或HSXT周期	约131ms
1	0	0	1	220个HICR或HSXT周期	约262ms
1	0	1	0	221个HICR或HSXT周期	约524ms
1	0	1	1	222个HICR或HSXT周期	约1s
1	1	0	0	223个HICR或HSXT周期	约2s
1	1	0	1	22个HICR或HSXT周期	约1us
1	1	1	0	24个HICR或HSXT周期	约4us
1	1	1	1	26个HICR或HSXT周期	约16us

注意：

- 软件复位不会复位该寄存器。
- 禁止在时钟稳定等待过程中改写该寄存器的值，否则不能保证稳定等待时间的正确性。

2.5.8 调试断点监视定时器控制寄存器 (DBWDT_CTRL)

调试断点监视定时器控制寄存器用来控制调试工具在断点时监视定时器的状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	HWWDG_BRK	保留	SWWDG_BRK		保留			
属性	R/W	-	R/W		-			
初始值	0	-	0		-			

寄存器功能

[bit7] HWWDG_BRK: 调试中断时硬件监视定时器控制位

bit	说明
0	调试中断时，硬件监视定时器停止计数（初始值）
1	调试中断时，硬件监视定时器继续计数

[bit6] 保留位，读数据为 0，写无效。

[bit5] SWWDG_BRK: 调试中断时软件监视定时器控制位

bit	说明
0	调试中断时，软件监视定时器停止计数（初始值）
1	调试中断时，软件监视定时器继续计数

[bit4:0] 保留位，读数据为 0，写无效。

注意：

- 软件复位不会复位该寄存器。

2.5.9 中断使能寄存器 (INT_CTRL)

中断使能寄存器用来使能或禁止中断。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留	FCSEN	SFCMIE	MFC MIE	保留	SUBEN	MAINEN	
属性	-	R/W	R/W	R/W	-	R/W	R/W	
初始值	-	0	0	0	-	0	0	

寄存器功能

[bit7:6] 保留位，读数据为 0，写无效。

[bit5] FCSEN: 外部高速时钟 (HSXT) 频率异常中断使能位

bit	说明
0	禁止HSXT频率异常中断（初始值）

1	使能HSXT频率异常中断
---	--------------

[bit4] SFCMIE: 外部低速时钟 (LSXT) 失效中断使能位

bit	说明
0	禁止LSXT失效中断 (初始值)
1	使能LSXT失效中断

[bit3] MFCMIE: 外部高速时钟 (HSXT) 失效中断使能位

bit	说明
0	禁止HSXT失效中断 (初始值)
1	使能HSXT失效中断

[bit2] 保留位, 读数据为 0, 写无效。

[bit1] SUBEN: 外部低速时钟 (LSXT) 振荡稳定等待结束中断使能位

bit	说明
0	禁止LSXT振荡稳定等待结束中断 (初始值)
1	使能LSXT振荡稳定等待结束中断

[bit0] MAINEN: 外部高速时钟 (HSXT) 振荡稳定等待结束中断使能位

bit	说明
0	禁止HSXT振荡稳定等待结束中断 (初始值)
1	使能HSXT振荡稳定等待结束中断

2.5.10 中断状态寄存器 (INT_PEND)

中断状态寄存器用于显示当前的中断状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留		FCS	SCFD	MCFD	保留	SUB	MAIN
属性	-		R	R	R	-	R	R
初始值	-		0	0	0	-	0	0

寄存器功能

[bit7:6] 保留位, 读数据为 0, 写无效。

[bit5] FCS: 外部高速时钟 (HSXT) 频率异常中断状态位

bit	说明
0	没有发生HSXT频率异常中断 (初始值)
1	有HSXT频率异常中断

[bit4] SCFD: 外部低速时钟 (LSXT) 失效中断状态位

bit	说明
0	没有发生LSXT失效中断（初始值）
1	有LSXT失效中断

[bit3] MCFD: 外部高速时钟（HSXT）失效中断状态位

bit	说明
0	没有发生HSXT失效中断（初始值）
1	有HSXT失效中断

[bit2] 保留位，读数据为 0，写无效。

[bit1] SUB: 外部低速时钟（LSXT）振荡稳定等待结束中断状态位

bit	说明
0	没有发生LSXT振荡稳定等待结束中断（初始值）
1	有LSXT振荡稳定等待结束中断

[bit0] MAIN: 外部高速时钟（HSXT）振荡稳定等待结束中断状态位

bit	说明
0	没有发生HSXT振荡稳定等待结束中断（初始值）
1	有HSXT振荡稳定等待结束中断

2.5.11 中断清零寄存器（INT_CLR）

中断清零寄存器用于清除中断标志。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留		FCS	SCFD	MCFD	保留	SUB	MAIN
属性	-		W	W	W	-	W	W
初始值	-		0	0	0	-	0	0

寄存器功能

[bit7:6] 保留位，读数据为 0，写无效。

[bit5] FCS: 外部高速时钟（HSXT）频率异常中断清除位

bit	说明
写0	HSXT频率异常中断标志无影响（初始值）
写1	清除HSXT频率异常中断
读	固定值0

[bit4] SCFD: 外部低速时钟（LSXT）失效中断清除位

bit	说明

写0	LSXT失效中断标志无影响（初始值）
写1	清除LSXT失效中断
读	固定值0

[bit3] MCFD: 外部高速时钟（HSXT）失效中断清除位

bit	说明
写0	HSXT失效中断标志无影响（初始值）
写1	清除HSXT失效中断
读	固定值0

[bit2] 保留位读数据为0，写无效。

[bit1] SUB: 外部低速时钟（LSXT）振荡稳定等待结束中断清除位

bit	说明
写0	LSXT振荡稳定等待结束中断标志无影响（初始值）
写1	清除LSXT振荡稳定等待结束中断
读	固定值0

[bit0] MAIN: 外部高速时钟（HSXT）振荡稳定等待结束中断清除位

bit	说明
写0	HSXT振荡稳定等待结束中断标志无影响（初始值）
写1	清除HSXT振荡稳定等待结束中断
读	固定值0

注意：

- 当本寄存器复位后，中断状态寄存器 INT_PEND 中的各状态标志位也同时清零。

2.6 注意事项

- 关于振荡稳定等待时间

- 外部高速时钟和外部低速时钟的振荡稳定等待时间与晶振的种类(水晶、陶瓷等)有关，针对所用晶振，应选择适当的振荡稳定等待时间。
- 请在时钟振荡稳定后，再使用相应的时钟。
- 禁止在时钟稳定等待过程中改写时钟稳定等待时间寄存器（CSWT）的值。

- 不受时钟产生单元控制的外设

以下外设的工作不受时钟产生单元的控制，详细介绍请参考相关章节。

- 时钟监视器：请参考“时钟监视器”
- 监视定时器：请参考“监视定时器”
- RTC：请参考“实时时钟”

- 关于时钟模式的切换

- 请按照以下步骤切换时钟模式：
- 设定各时钟的振荡稳定等待时间
- 将希望使用的时钟振荡使能位置 1
- 等待希望使用的时钟振荡状态稳定位为 1
- 改写主时钟切换控制位
- 等待主时钟选择状态位等于主时钟切换控制位

- 关于时钟模式和振荡稳定位的关系

- 内部高速 RC 或外部高速时钟运行模式切换到其他时钟模式时寄存器 CTRL 中的振荡使能位写 1 后，相应时钟开始振荡。在设置的时间后，可以检查寄存器 CLK_STB 中的振荡稳定位为 1。
- 内部低速 RC 或外部低速时钟运行模式切换到其他时钟模式时寄存器 CTRL 中的 MAIN_EN 或 HC_EN 位写 1 后，相应时钟不会开始振荡。必须在 MAIN_EN 或 HC_EN 写 1 后，再改写主时钟切换控制位，当振荡稳定等待时间结束后，相应的振荡稳定位会置 1。

- 使用中断唤醒 standby 模式后，时钟模式将切换到时钟切换控制位 INPUT 对应的模式。
- 发生软件复位以外的其他复位时，主时钟将切换到内部高速 RC 时钟（HICR），同时外部高速时钟（HSXT）和外部低速时钟（LSXT）停止工作。
- 使用外部高速晶振时钟前，必须访问 BGR 控制寄存器使能 BGR 并等待 BGR 稳定。

3. 外设时钟门控

3.1 概述

外设时钟门控模块可以分别停止各外设的工作时钟，以便降低系统功耗。

同时，外设时钟门控模块还可以分别复位各外设模块，复位结果与发生总线复位相同。

外设时钟门控模块的工作时钟为 APB1 的总线时钟 PCLK1，请在使用本模块前设置 APB1 时钟分频寄存器，使能 APB1 总线时钟。

外设时钟的初始状态

表 3.1-1 外设时钟的初始状态

外设	初始状态	说明
IO	提供时钟	同时控制所有IO
ADC	提供时钟	同时控制所有通道
多功能串口	提供时钟	分别控制每个通道
电压比较器	提供时钟	-
复合定时器	提供时钟	通道0~通道3，通道4~通道7，分两组控制
LCD	提供时钟	-
ISO7816/IC卡	提供时钟	-
AES	提供时钟	-
蜂鸣器	提供时钟	-

3.2 配置

本节将介绍外设时钟门控模块的配置。

框图

图 3.2-1 外设时钟门控模块框图

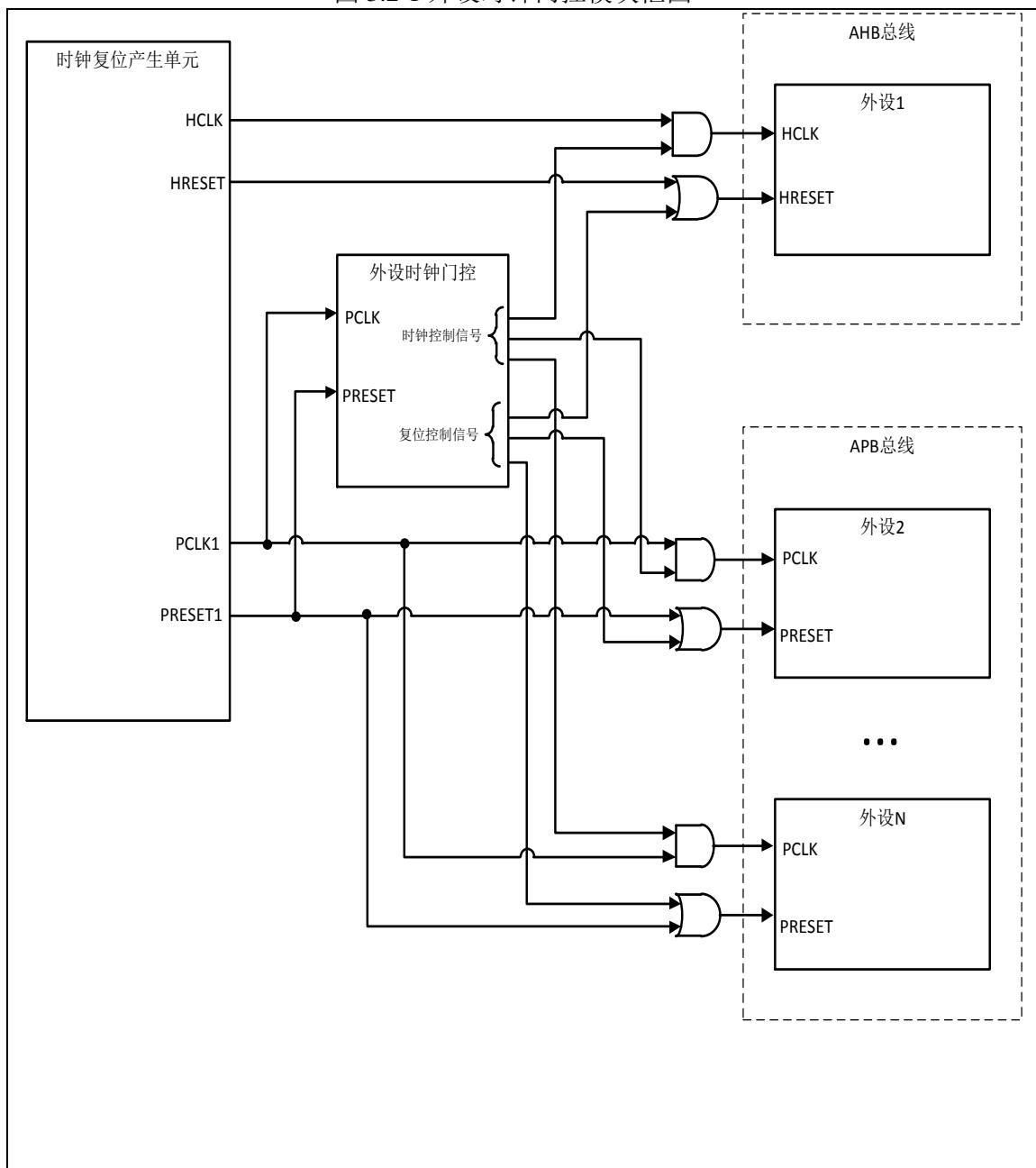


图 3.2-1 所示的系统框图中，外设时钟门控模块产生外设的时钟使能信号和复位信号，该模块连接于 APB1 总线。

若要停止某外设的时钟，则将该模块中的时钟控制寄存器对应的位写 0；反之，若要提供时钟，则写 1。

若要复位某外设，则将该模块中的复位控制寄存器对应的位写 1；反之，若要释放复位，则写 0。

3.3 操作

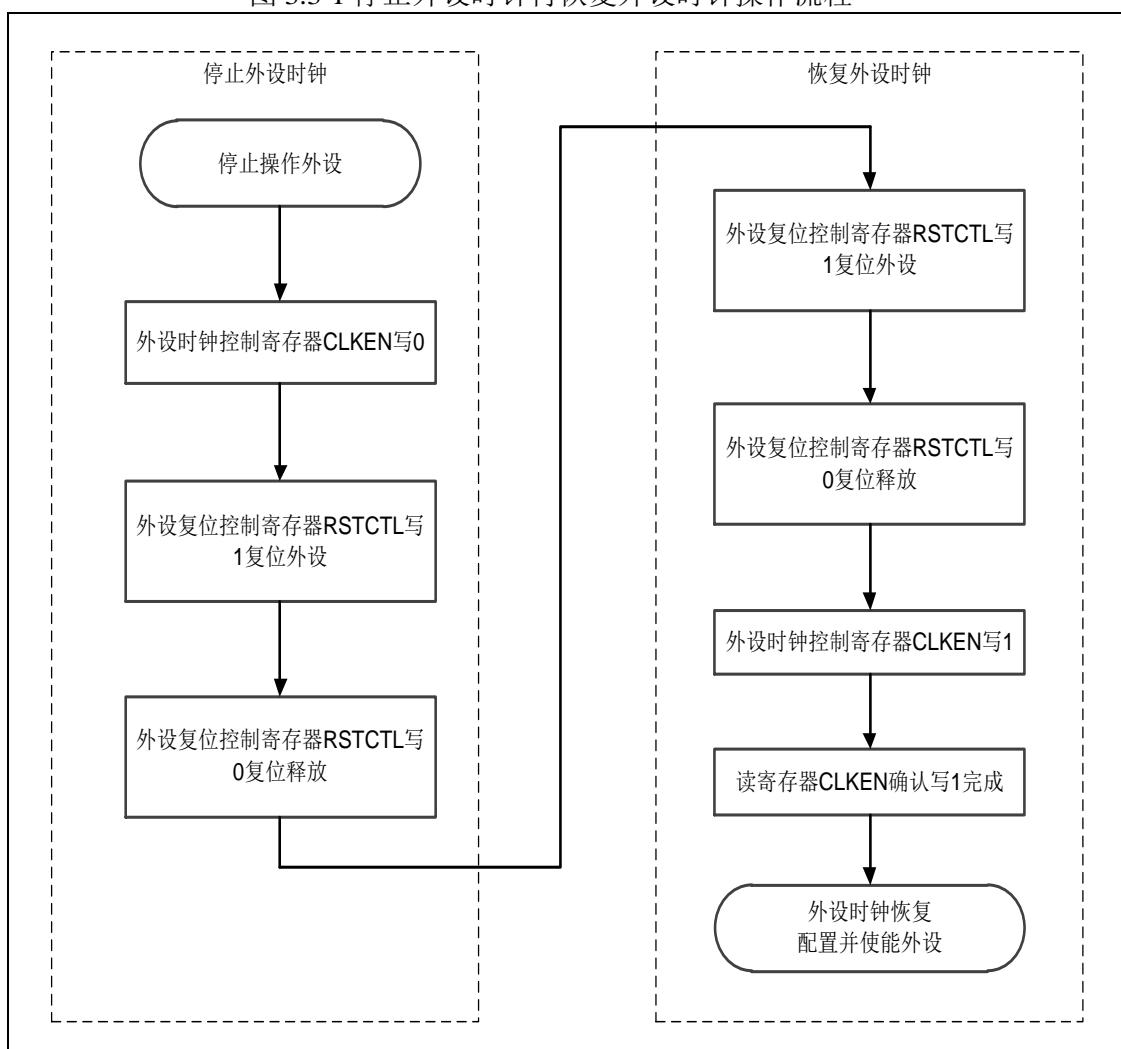
外设时钟门控模块中的寄存器复位源是 APB1 总线复位信号，即 PRESET1。

当系统复位完成后，本模块中的寄存器同时被复位，此时需要重新设置各寄存器。

停止时钟然后恢复时钟操作流程

图 3.3-1 是通过设置外设时钟门控模块停止外设时钟后，再重新恢复外设时钟的操作流程。

图 3.3-1 停止外设时钟再恢复外设时钟操作流程



3.4 寄存器

表 3.4-1 外设时钟门控模块的寄存器列表

缩写	寄存器名称	章节
CLKEN0	外设时钟控制寄存器0	3.4.1
RSTCTL0	外设复位控制寄存器0	3.4.2
CLKEN1	外设时钟控制寄存器1	3.4.3
RSTCTL1	外设复位控制寄存器1	3.4.4
CLKEN2	外设时钟控制寄存器2	3.4.5
RSTCTL2	外设复位控制寄存器2	3.4.6

3.4.1 外设时钟控制寄存器 0 (CLKEN0)

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	保留			GIOCLK K	保留			
属性					R/W			
初始值					1			

位	23	22	21	20	19	18	17	16
字段名	保留					ADCCLK		
属性						R/W		
初始值						1		

位	15	14	13	12	11	10	9	8
字段名	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0						
字段名	保留		MSCCLK[5:0]											
属性	-													
初始值	6'b111111													

寄存器功能

[bit31:29] 保留位,读数据为 0, 写无效。

[bit28] GIOCLK: GPIO/快速 GPIO 时钟控制位

bit	说明
0	停止IO端口的总线访问时钟

1	使能IO端口的总线访问时钟（初始值）
---	--------------------

[bit27:17]保留位,读数据为0,写无效。

[bit16] ADCCLK: ADC 时钟控制位

bit	说明
0	停止ADC的总线访问时钟
1	使能ADC的总线访问时钟（初始值）

[bit15:6]保留位,读数据为0,写无效。

[bit5:0] MSCCLK[5:0]: 多功能串口时钟控制位

多功能串口通道与各控制位的对应关系为:

Bit5: 多功能串口通道 6

Bit4: 多功能串口通道 5

Bit3: 多功能串口通道 4

Bit2: 多功能串口通道 2

Bit1: 多功能串口通道 1

Bit0: 多功能串口通道 0

bit	说明
0	停止对应通道的总线访问时钟
1	使能对应通道的总线访问时钟（初始值）

3.4.2 外设复位控制寄存器 0 (RSTCTL0)

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段名	保留							
属性	-							
初始值	0							

位	15	14	13	12	11	10	9	8
字段名	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---

字段名	保留	MSCRST[5:0]
属性	-	R/W
初始值	-	6'b000000

寄存器功能

[bit31:17]保留位,读数据为 0, 写无效。

[bit16] ADCRST: ADC 复位控制位

bit	说明
0	释放ADC复位（初始值）
1	复位ADC

[bit15:6]保留位,读数据为 0, 写无效。

[bit5:0] MSCRST[5:0]: 多功能串口复位控制位

多功能串口通道与各控制位的对应关系为:

Bit5: 多功能串口通道 6

Bit4: 多功能串口通道 5

Bit3: 多功能串口通道 4

Bit2: 多功能串口通道 2

Bit1: 多功能串口通道 1

Bit0: 多功能串口通道 0

bit	说明
0	释放对应通道复位（初始值）
1	复位对应通道

3.4.3 外设时钟控制寄存器 1 (CLKEN1)

寄存器配置

位	31							8
字段名	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段名	保留					CTCLK[1:0]		
属性	-					R/W		
初始值	-					11		

寄存器功能

[bit31:5]保留位,读数据为 0, 写无效。

[bit4] VCCLK: 电压比较器时钟控制位

bit	说明
0	停止电压比较器的总线访问时钟
1	使能电压比较器的总线访问时钟（初始值）

[bit3:2]保留位,读数据为 0, 写无效。

[bit1:0] CTCLK[1:0]: 复合定时器时钟控制位

每四个复合定时器通道为一组进行控制, 通道与各控制位的对应关系为:

Bit1: 通道 4~通道 7

Bit0: 通道 0~通道 3

bit	说明
0	停止对应通道的总线访问时钟
1	使能对应通道的总线访问时钟（初始值）

3.4.4 外设复位控制寄存器 1 (RSTCTL1)

寄存器配置

位	31							8
字段名	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段名	保留			VCRST	保留			CTRST[1:0]
属性	-			R/W	-			R/W
初始值	-			0	-			00

寄存器功能

[bit31:5]保留位,读数据为 0, 写无效。

[bit4] VCRST: 电压比较器复位控制位

bit	说明
0	释放电压比较器复位（初始值）
1	复位电压比较器

[bit3:2]保留位,读数据为 0, 写无效。

[bit1:0] CTRST[1:0]: 复合定时器复位控制位

每四个复合定时器通道为一组进行控制, 通道与各控制位的对应关系为:

Bit1: 通道 4~通道 7

Bit0: 通道 0~通道 3

bit	说明
0	释放对应通道复位（初始值）
1	复位对应通道

3.4.5 外设时钟控制寄存器 2 (CLKEN2)

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段名	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段名	保留			SCICLK	保留			AESCLK
属性		-		R/W		-		R/W
初始值		-		1		-		1

位	7	6	5	4	3	2	1	0
字段名	保留			BUZCLK	保留			
属性		-		R/W	-			
初始值		-		1	-			

寄存器功能

[bit31:17]保留位,保留位,写无效。

[bit16] LCDCCLK: LCD 控制器时钟控制位

bit	说明
0	停止LCD控制器的总线访问时钟
1	使能LCD控制器的总线访问时钟（初始值）

[bit15:13]保留位,读数据为 0, 写无效。

[bit12] SCICLK: ISO7816/IC 卡时钟控制位

bit	说明
0	停止ISO7816/IC卡的总线访问时钟
1	使能ISO7816/IC卡的总线访问时钟（初始值）

[bit11:9]保留位,读数据为 0, 写无效。

[bit8] AESCLK: AES 时钟控制位

bit	说明
0	停止AES的总线访问时钟
1	使能AES的总线访问时钟（初始值）

[bit7:5]保留位,读数据为0,写无效。

[bit4] BUZCLK: 蜂鸣器时钟控制位

bit	说明
0	停止蜂鸣器的总线访问时钟
1	使能蜂鸣器的总线访问时钟（初始值）

[bit3:0]保留位,读数据为0,写无效。

3.4.6 外设复位控制寄存器 2 (RSTCTL2)

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段名	保留							
属性	-							
初始值	0							

位	15	14	13	12	11	10	9	8			
字段名	保留			SCIRST	保留			AESRST			
属性	R/W										
初始值	0										

位	7	6	5	4	3	2	1	0				
字段名	保留			BUZRST	保留							
属性	R/W											
初始值	0											

寄存器功能

[bit31:17]保留位,保留位,写无效。

[bit16] LCDRST: LCD 控制器复位控制位

bit	说明
0	释放LCD控制器复位（初始值）
1	复位LCD控制器

[bit15:13]保留位,读数据为0,写无效。

[bit12] SCIRST: ISO7816/IC 卡复位控制位

bit	说明
0	释放ISO7816/IC卡复位（初始值）
1	复位ISO7816/IC卡

[bit11:9]保留位,读数据为 0, 写无效。

[bit8] AESRST: AES 复位控制位

bit	说明
0	释放AES复位（初始值）
1	复位AES

注意:

- AESRST 不能复位 AES 分频寄存器 AES_DIV_EN

[bit7:5]保留位,读数据为 0, 写无效。

[bit4] BUZRST: 蜂鸣器复位控制位

bit	说明
0	释放蜂鸣器复位（初始值）
1	复位蜂鸣器

[bit3:0]保留位

读数据为 0, 写无效。

3.5 注意事项

- 访问总线时钟已经停止的外设

对这些外设的寄存器进行读或写操作将失败。CPU 读到的数据不是该寄存器的真实值，同时写操作也被禁止。

- 外设时钟/复位的初始化

本模块中的外设时钟控制寄存器和复位控制寄存器的复位源包括以下这些复位。当发生其中任意一种复位后，请重新设置各寄存器，以便降低系统功耗。关于复位的详细描述，请参考另一章节“复位”。

- 上电复位 (PWR)
- 低电压检测复位 (LVD)
- 外部引脚复位 (RSTB)
- 软件监视定时器复位 (SWWDG)
- 硬件监视定时器复位 (HWWDG)
- 外部高速时钟频率异常复位 (AF)
- 外部高速/低速时钟失效复位 (CF)
- 软件复位 (SW)
- 深度待机转换复位 (DSTB)
- APB1 总线复位 (写寄存器 APB1_CTRL)

4. 时钟调校

4.1 概述

说明内部高速 RC 时钟、外部高速振荡器和外部低速振荡器的调校功能。

内部高速 RC 时钟

由于生产工艺的不同，芯片内部高速 RC 时钟振荡器的频率精度有一定的偏差。可以通过调校功能，来减小由于不同工艺以及应用场景的温度变化造成的频率精度偏差的影响。

频率调校模块具有以下功能：

- 可以通过将频率调校值写入内部高速 RC 振荡器的频率调校寄存器（HICR_FTRM）不同比特位，来配置 4MHz/16MHz 两种频率的 RC 时钟频率。
- 可以使用复合定时器的 0 通道（ch.0），通过指定周期内的计数值来计算出内部高速 RC 时钟振荡器的频率

外部高速/低速振荡器

外部高速/低速晶振的种类不同，其稳定振荡的条件略有不同。可以配置外部高速晶振时钟控制寄存器（HSXT_PWRCTL）和外部低速晶振时钟控制寄存器（LSXT_PWRCTL）选择合适的振荡条件。

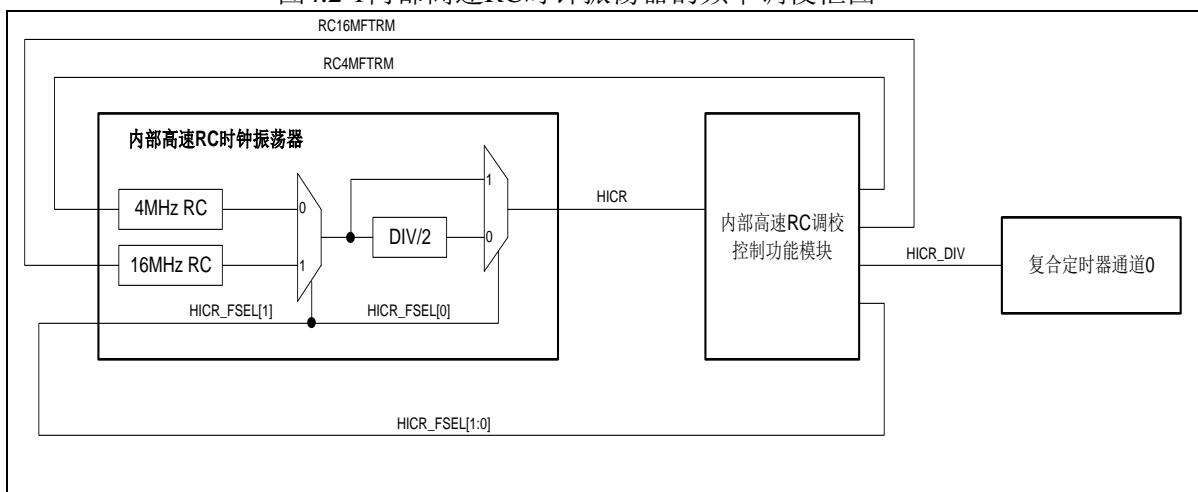
4.2 配置及框图

说明内部高速 RC 时钟振荡器频率调校功能配置及框图。

框图

图 4.2-1 是内部高速 RC 时钟振荡器的频率调校框图。

图4.2-1 内部高速RC时钟振荡器的频率调校框图



配置

- 内部高速 RC 时钟振荡器

出于功耗控制的考虑，本芯片的高速 RC 可以选择 16MHz 和 4MHz 其中一种，可以对其进行 2 分频。对于系统时钟来说，一共可以选择 2MHz/4MHz/8MHz/16MHz 等四种频率的内部高速 RC 时钟作为芯片的时钟源。内部高速 RC 时钟频率/分频的选择是由内部高速 RC 时钟振荡器的频率选择寄存器（HICR_FSEL[1:0]）来控制。

此外，可以通过内部高速 RC 时钟振荡器的频率调校寄存器(HICR_FTRM)的 RC16MFTRM 以及 RC4MFTRM 比特位，分别对 16MHz 和 4MHz 高速 RC 时钟振荡器的频率进行调校。

- 内部高速 RC 调校控制功能模块

该模块用于设置内部高速 RC 时钟振荡器的控制电路以及相关寄存器。此外，按照内部高速 RC 时钟振荡器分频设置寄存器(HICR_FDIV)所设比率所分频的高速 RC 时钟(HICR_DIV)输出至复合定时器通道 0 用于计算此高速 RC 时钟的频率。

- 复合定时器

该模块可以用通道 0 对分频后的内部高速 RC 进行计算，以计算出内部高速 RC 的时钟频率。

注意：

- 有关时钟的相关定义，请参考“时钟”的相关章节。

4.3 操作

本节说明内部高速 RC 时钟振荡器频率调校功能的相关操作。

频率调校

通过程序，将调校后的数值写入到内部高速 RC 时钟振荡器的频率调校寄存器（HICR_FTRM）的 RC16MFTRM 以及 RC4MFTRM 比特位，分别对 16MHz 和 4MHz 高速 RC 时钟振荡器的频率进行调校。

频率调校值的加载

系统复位期间，片上闪存内“RC 调校值”区域中的工厂预设值会加载到内部高速 RC 时钟振荡器的频率调校寄存器（HICR_FTRM）的 RC16MFTRM 以及 RC4MFTRM 比特位，使得在复位解除后，内部高速 RC 可以以一个比较准确的频率运行。

当系统复位解除后，任何需要改变频率的诉求，都必须通过修改频率调校寄存器（HICR_FTRM）的 RC16MFTRM 以及 RC4MFTRM 比特位来完成。

频率调校的方法

可以通过以下两种方法对内部高速 RC 的频率进行调校。

- 输出内部高速 RC 时钟到外部引脚，监测输出时钟的频率，通过程序调校。
- 使用复合定时器通道 0，经过相关设置后，通过在一定时间内复合定时器的计数值来计算内部高速 RC 时钟振荡器的频率，并通过程序来调校。

注意：

- 擦除整片片上闪存时，如果想再次使用内部高速 RC 的出厂预设值，请先将其保存在其他地方（如片上 SRAM），待片上闪存擦除后再写回相应地址空间。
- 在修改频率调校值后需要等待内部高速 RC 稳定时间。
- 有关片上闪存内“RC 调校值”区域，请参考“闪存”。

4.4 设置流程

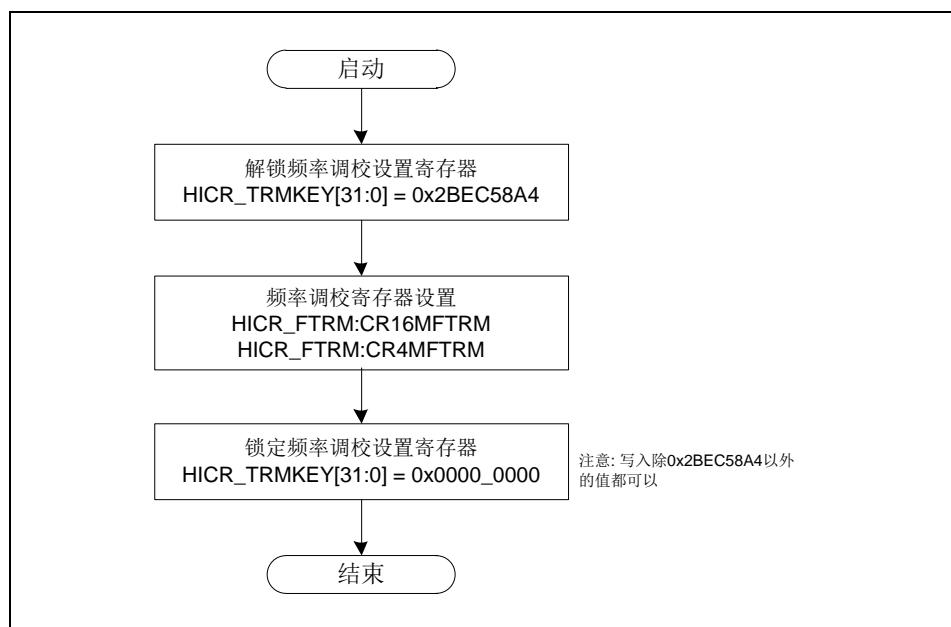
本节说明内部高速 RC 时钟振荡器频率调校功能的设置流程。

频率调校寄存器的设置流程：

- 1) 将解码值“0x2BEC58A4”写入内部高速 RC 频率设置寄存器的保护寄存器（HICR_TRMKEY）中，以解锁频率调校寄存器（HICR_FTRM）。
- 2) 根据需要，设置频率调校寄存器（HICR_FTRM）的 RC16MFTRM 以及 RC4MFTRM 比特位。
- 3) 将解码值“0x2BEC58A4”以外的值，写入内部高速 RC 频率设置寄存器的保护寄存器（HICR_TRMKEY）中，以锁定频率调校寄存器（HICR_FTRM）。

图 4.4-1 是内部高速 RC 时钟振荡器的频率调校设置流程图。

图 4.4-1 内部高速 RC 时钟振荡器的频率调校设置流程



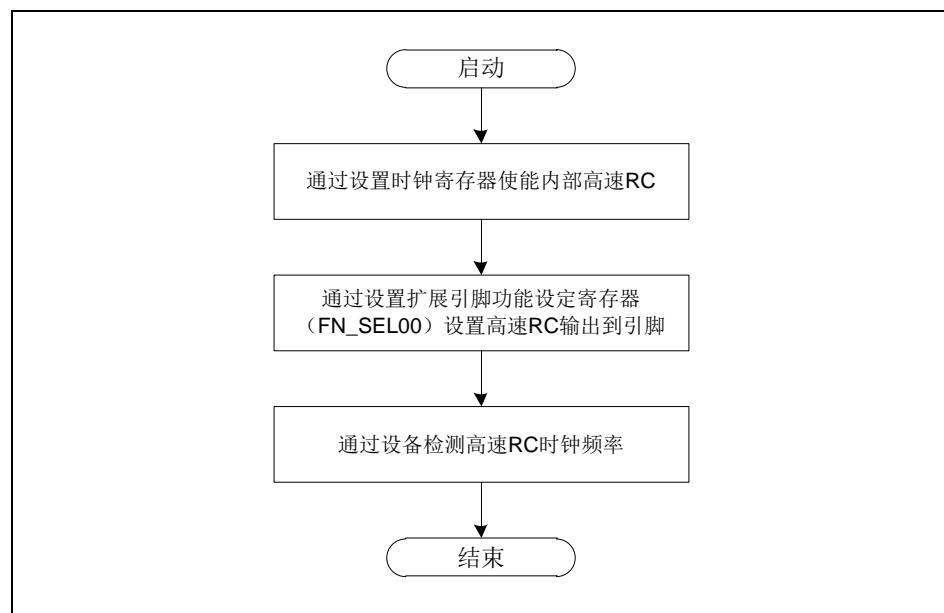
频率调校方法流程：

要进行频率调校，首先需要得到目前内部高速 RC 时钟振荡器的频率。如前所述，可以：

- 1) 直接输出内部高速 RC 时钟到外部引脚，使用仪器进行检测；

图 4.4-2 是将内部高速 RC 时钟振荡器输出至外部引脚的设置流程图。

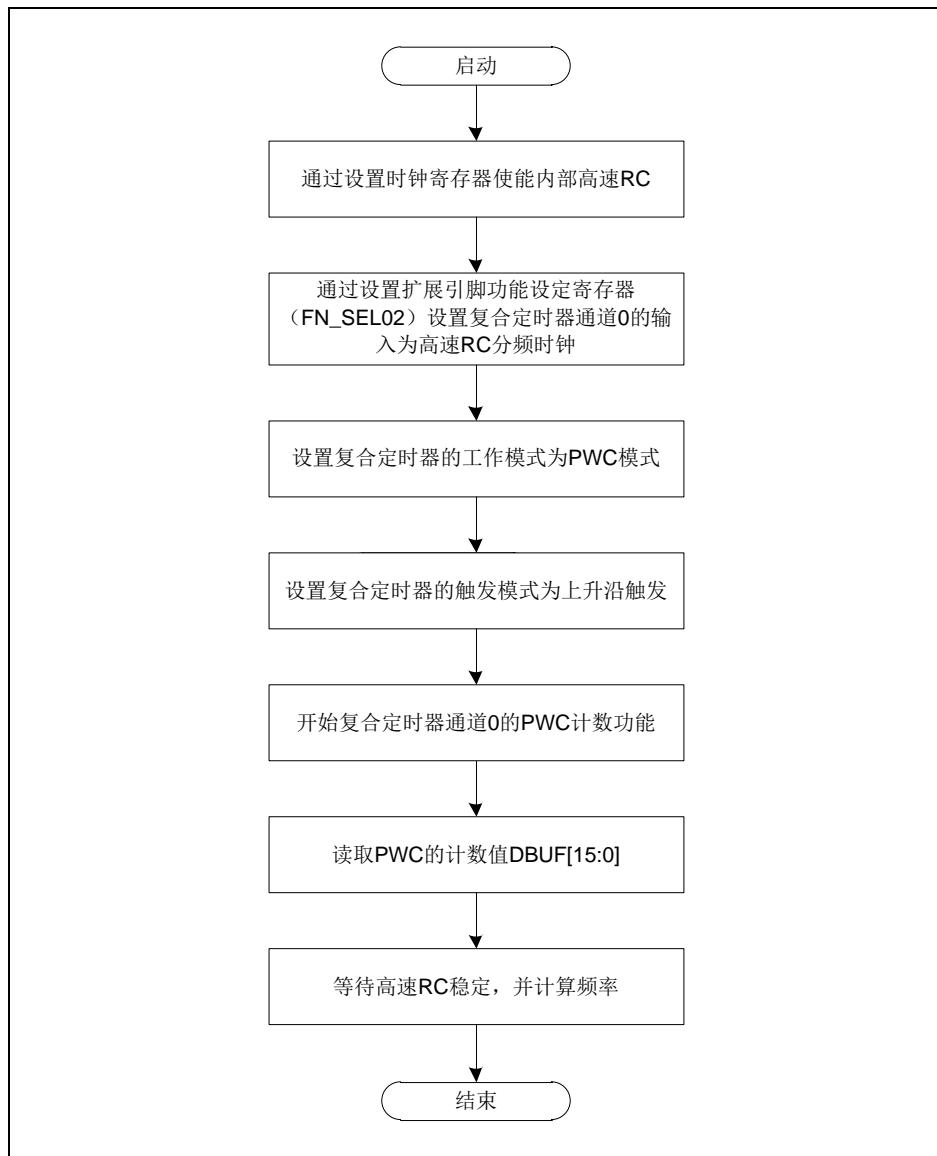
图 4.4-2 内部高速 RC 时钟振荡器输出至外部引脚设置流程



2) 使用复合定时器通道 0, 用一定时间内定时器的计数值来计算。

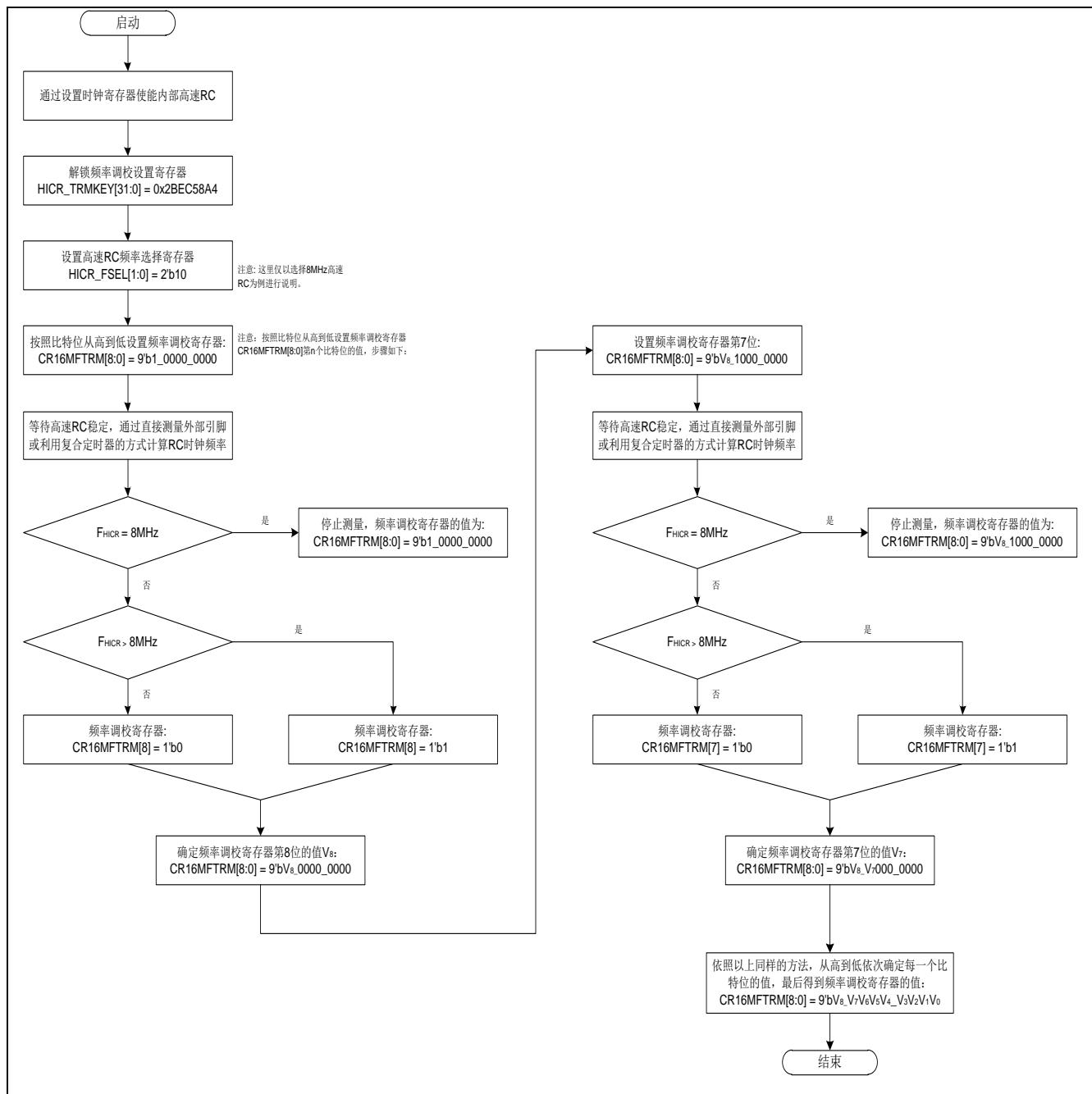
图 4.4-3 是利用基本定时计数器计算内部高速 RC 时钟振荡器频率的设置流程图。

图 4.4-3 利用复合定时器计算内部高速 RC 时钟频率的设置流程



3) 根据需要, 设置频率调校寄存器 (HICR_FTRM) 的 RC16MFTRM 以及 RC4MFTRM 比特位。

调校内部高速 RC 时钟振荡器的频率是通过二分法, 从高位到低位, 依次设置、计算、比较, 逐次逼近最优值, 现以 16MHz 二分频为 8MHz 作为 HICR 为例, 步骤如下:



4.5 寄存器

表 4.5-1 列出了相关寄存器如下。

表 4.5-1 时钟调校寄存器列表

缩写	寄存器名称	章节
HICR_TRMKEY	内部高速RC时钟调校寄存器写保护寄存器	4.5.1
HICR_FTRM	内部高速RC时钟频率调校寄存器	4.5.2
HICR_FSEL	内部高速RC时钟输出频率选择寄存器	4.5.3
HICR_FDIV	内部高速RC时钟分频设置寄存器	4.5.4
HSXT_PWRCTL	外部高速晶振时钟控制寄存器	4.5.5
LSXT_PWRCTL	外部低速晶振时钟控制寄存器	4.5.6

4.5.1 内部高速 RC 时钟调校写保护寄存器 (HICR_TRMKEY)

HICR_TRMKEY 寄存器控制 RC 时钟频率调校寄存器 (HICR_FTRM) 的写保护。

寄存器配置

位	31							0
字段名	TRMKEY[31:0]							
属性	R/W							
初始值	0x0000_0001							

寄存器功能

[bit31:0] TRMKEY[31:0]: RC 时钟频率调校寄存器写保护控制位

bit31:0	说明
读取	当读取值为0x0000_0000时, HICR_FTRM寄存器当前未被锁定 当读取值为0x0000_0001时, HICR_FTRM寄存器当前被锁定
写入0x2BEC58A4以外的值	锁定HICR_FTRM寄存器
写入0x2BEC58A4	解锁HICR_FTRM寄存器

4.5.2 内部高速 RC 时钟频率调校寄存器 (HICR_FTRM)

HICR_FTRM 寄存器调校内部高速 RC 时钟振荡器的频率。

寄存器配置

位	31		25	24				16
字段名	保留		CR16MFTRM[8:0]					
属性	-		R/W					
初始值	-		9b'0_1011_1101					

位	15		9	8				0
字段名	保留		CR4MFTRM[8:0]					
属性	-		R/W					
初始值	-		9b'0_1011_1101					

寄存器功能

[bit31:25] 保留：保留位,读数据为 0，写无效。

[bit24:16] CR16MFTRM[8:0]: 16MHz 内部高速 RC 时钟的频率调校设置位

寄存器值将会在复位后更新为 Flash 地址 0x0010_000C 的第 24~16 位数值

Bit24:16	说明
读取	读出目前的频率调校值
写入	设置内部高速16MHz RC时钟的频率调校值

[bit15:9] 保留：保留位，读数据为 0，写无效。

[bit8:0] CR4MFTRM[8:0]: 4MHz 内部高速 RC 时钟的频率调校设置位

寄存器值将会在复位后更新为 Flash 地址 0x0010_0010 的第 16~第 8 位数值

Bit8:0	说明
读取	读出目前的频率调校值
写入	设置内部高速4MHz RC时钟的频率调校值

4.5.3 内部高速 RC 时钟输出频率选择寄存器 (HICR_FSEL)

HICR_FSEL 寄存器选择内部高速 RC 时钟振荡器的输出频率。

寄存器配置

位	31							16
字段名	保留							
属性	-							
初始值	-							

位	15					2	1	0
字段名	保留						FSEL[1:0]	
属性	-						R/W	
初始值	-						2'b01'	

寄存器功能

[bit31:2] 保留：保留位，读数据为 0，写无效。

[bit1:0] FSEL[1:0]: 内部高速 RC 时钟振荡器频率输出选择位

Bit1:0	说明				
读取/写入	<p>FSEL[1]选择内部高速RC之中16MHz或者4MHz作为HICR输出 FSEL[1]=1'b0: 内部高速4MHz RC时钟振荡器作为HICR FSEL[1]=1'b1: 内部高速16MHz RC时钟振荡器作为HICR FSEL[0]决定内部高速RC时钟是否2分频后再输出 FSEL[0]=1'b0: 内部高速RC时钟振荡器2分频后再作为HICR FSEL[0]=1'b1: 内部高速RC时钟振荡器不分频即作为HICR 例如: FSEL[1:0]=2'b10, 表明内部高速16MHz RC时钟2分频后作为HICR, 所以, HICR的频率为8MHz。 HICR频率对应FSEL寄存器的真值表如下:</p> <table border="0"> <tr> <td>FSEL[1:0] = 2'b00: HICR频率为2MHz</td> </tr> <tr> <td>FSEL[1:0] = 2'b01: HICR频率为4MHz</td> </tr> <tr> <td>FSEL[1:0] = 2'b10: HICR频率为8MHz</td> </tr> <tr> <td>FSEL[1:0] = 2'b11: HICR频率为16MHz</td> </tr> </table>	FSEL[1:0] = 2'b00: HICR频率为2MHz	FSEL[1:0] = 2'b01: HICR频率为4MHz	FSEL[1:0] = 2'b10: HICR频率为8MHz	FSEL[1:0] = 2'b11: HICR频率为16MHz
FSEL[1:0] = 2'b00: HICR频率为2MHz					
FSEL[1:0] = 2'b01: HICR频率为4MHz					
FSEL[1:0] = 2'b10: HICR频率为8MHz					
FSEL[1:0] = 2'b11: HICR频率为16MHz					

4.5.4 内部高速 RC 时钟分频设置寄存器 (HICR_FDIV)

HICR_FDIV 寄存器设置作为复合定时器输入的内部高速 RC 时钟振荡器的分频比率。

寄存器配置

位	31							16
字段名	保留							
属性	-							
初始值	-							

位	15					3	2	0
字段名	保留						FDIV[2:0]	
属性	-						R/W	
初始值	-						3'b001'	

寄存器功能

[bit31:3] 保留：保留位，读数据为 0，写无效。

[bit2:0] FDIV[2:0]: 内部高速 RC 时钟振荡器分频比率选择位

bit2	bit1	bit0	说明
0	0	0	1/4
0	0	1	1/8
0	1	0	1/16
0	1	1	1/32
1	0	0	1/64
1	0	1	1/128
1	1	0	1/256
1	1	1	1/512

4.5.5 外部高速晶振时钟控制寄存器 (HSXT_PWRCTL)

寄存器配置

位	31					18	17	16
字段名	保留						HIBS[1:0]	
属性	-						R/W	
初始值	-						00	
位	15					2	1	0
字段名	保留						HXSEL[1:0]	
属性	-						R/W	
初始值	-						00	

寄存器功能

[bit31:18] 保留: 保留位, 读数据为 0, 写无效。

[bit17:16] HIBS[1:0]: 外部高速晶振偏置电流设置位

Bit17:16	说明
读取	读出目前的设置值
写入	设置外部高速晶振的偏置电流

[bit15:2] 保留: 保留位, 读数据为 0, 写无效。

[bit1:0] HXSEL[1:0]: 外部高速晶振频率范围设置位

Bit1:0	说明
读取	读出目前的设置值
写入	00 4MHz ~ 6MHz
	01 6MHz ~ 12MHz
	10 12MHz ~ 20MHz
	11 20MHz ~ 32MHz

4.5.6 外部低速晶振时钟控制寄存器（LSXT_PWRCTL）

寄存器配置

位	31					18	17	16
字段名	保留						LIBS[1:0]	
属性	-						R/W	
初始值	-						10	

位	15					2	1	0
字段名	保留						LCSEL[1:0]	
属性	-						R/W	
初始值	-						10	

寄存器功能

[bit31:18] 保留：保留位，读数据为 0，写无效。

[bit17:16] LIBS[1:0]: 外部低速晶振偏置电流设置位

Bit17:16	说明
读取	读出目前的设置值
写入	设置外部低速晶振的偏置电流

[bit15:2] 保留：保留位，读数据为 0，写无效。

[bit1:0] LCSEL[1:0]: 外部低速晶振振幅设置位

Bit1:0	说明
读取	读出目前的设置值
写入	设置外部低速晶振的振幅

4.6 注意事项

本节说明使用时钟调校功能时，需要注意的事项。

- 复合定时器

关于复合定时器的具体使用方法，请参考“复合定时器”的相关章节。

- 片上闪存的“RC 调校值”区的操作

内部高速 RC 时钟振荡器的频率调校预设值存储在片上闪存的“用户数据”区域。关于具体的值存储地址，请参考寄存器说明。

5. 时钟监视器

5.1 概述

时钟监视器包括以下两类功能。

时钟失效检测（CFD）

此功能检测外部高速时钟（HSXT）和外部低速时钟（LSXT）。当一定时间内（不可配置）检测不到 HSXT 或 LSXT 的上升沿时，则认为时钟已经失效，此时产生系统复位或中断（可以由寄存器选择）。

*不使用该功能时，建议上电复位之后关闭。

时钟频率异常检测（AFD）

此功能检测外部高速时钟（HSXT）。当 HSXT 的实际频率超出设定的范围后，产生系统复位或中断可以由寄存器选择）。

5.2 配置

5.2.1 时钟失效检测

图 5.2.1-1 是时钟失效检测的原理框图。

图 5.2.1-1 时钟失效检测的原理框图

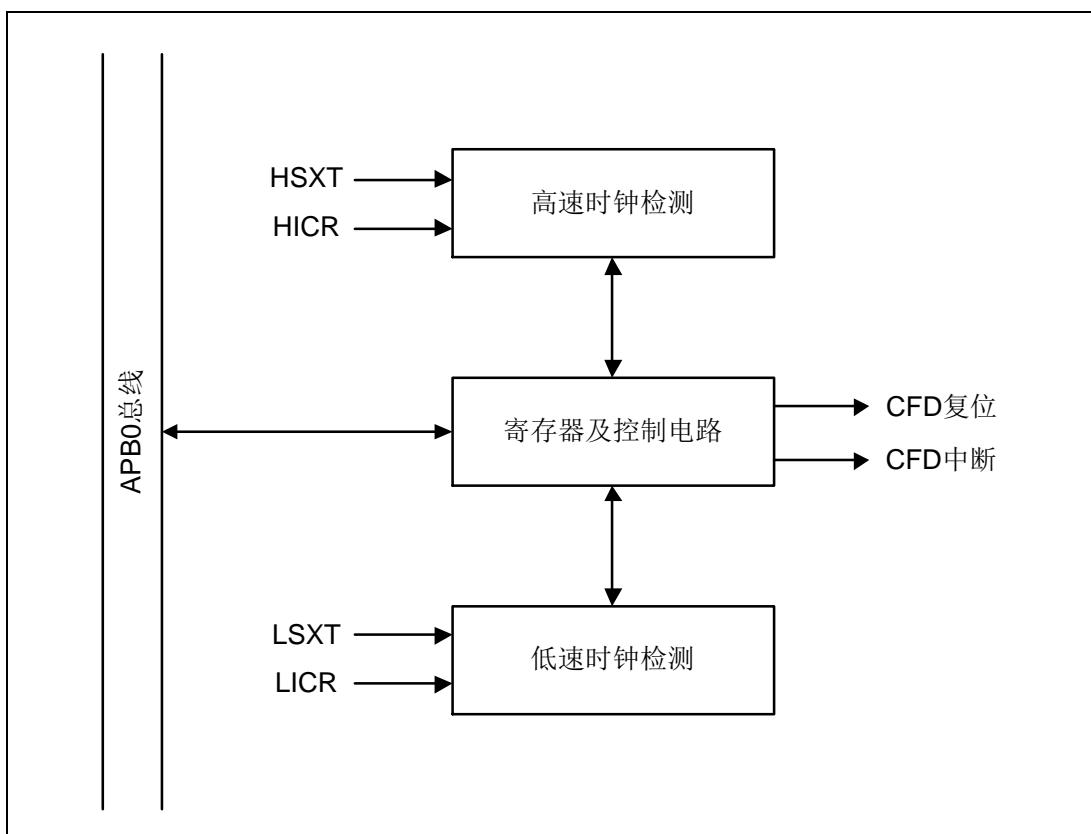


图 5.2.1-1 所示的系统框图中，包括以下模块。

高速时钟检测

该模块用内部高速 RC 时钟监控外部高速时钟，当外部高速时钟失效时产生报警信号。

低速时钟检测

该模块用内部低速 RC 时钟监控外部低速时钟，当外部低速时钟失效时产生报警信号。

寄存器及控制电路

该模块产生高速时钟检测和低速时钟检测电路的工作使能，中断使能和复位使能等控制信号，并控制复位和中断的输出。

该模块连接于 APB0 总线。

5.2.2 时钟频率异常检测

图 5.2.2-2 是时钟频率异常检测的原理框图。

图 5.2.2-2 时钟频率异常检测的原理框图

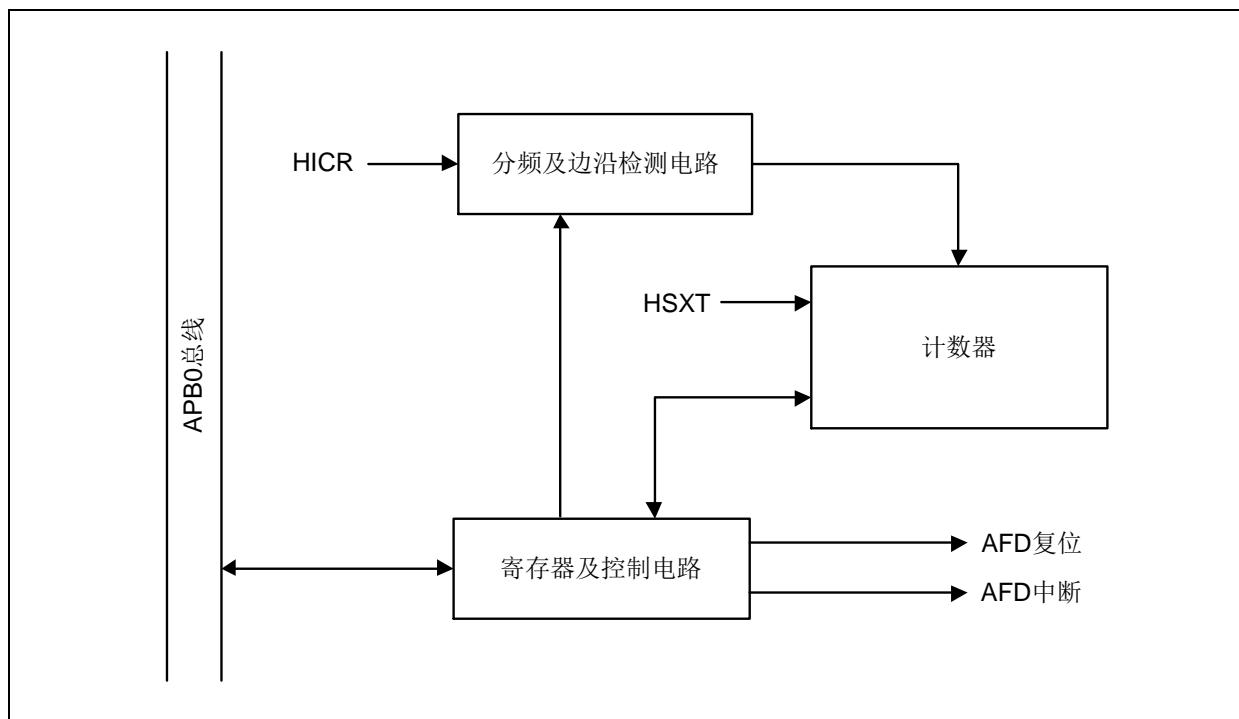


图 5.2.2-2 所示的系统框图中，包括以下模块。

分频及边沿检测电路

本模块将内部高速 RC 时钟分频，并检测其边沿。

计数器

外部高速时钟为计数时钟，计算分频后的内部高速 RC 时钟两个边沿之间的时间。当检测到的时间超过设定值后，产生报警信号。

寄存器及控制电路

该模块控制内部高速 RC 时钟的分频比例，计数器的使能，复位和中断的输出。

该模块连接于 APB0 总线。

5.3 操作

5.3.1 时钟失效检测

时钟失效检测功能的被检测时钟是外部高速时钟和外部低速时钟。在一定的时间内，如果被检测时钟没有上升沿，则认为该时钟已经失效，此时产生系统复位或中断。产生的复位称作 CFD 复位，产生的中断称作 CFD 中断。

使能或禁止 CFD 复位，请写寄存器 FCM_CTL 的 MFCMRE 和 SFCMRE 位，详细描述请参考第 5.5.1 节“时钟监视器控制寄存器”。

使能或禁止 CFD 中断，请写寄存器 INT_CTRL 的 MFCMIE 和 SFCMIE 位，详细描述请参考第 2.5.9 节“中断使能寄存器”。

时钟失效检测有以下特点：

- 外部高速时钟（HSXT）的检测和外部低速时钟（LSXT）的检测是独立的，可以单独配置
- 外部高速时钟检测的计数时钟是内部高速 RC 时钟（HICR），外部低速时钟检测的计数时钟是内部低速 RC 时钟（LICR）
- 时钟振荡稳定等待时间内，检测电路停止工作
- 时钟振荡稳定等待过程结束后，检测电路自动开始工作
- 外部高速晶振和外部低速晶振停止振荡之后，检测电路停止工作

图 5.3.1-1 HSXT 时钟失效举例

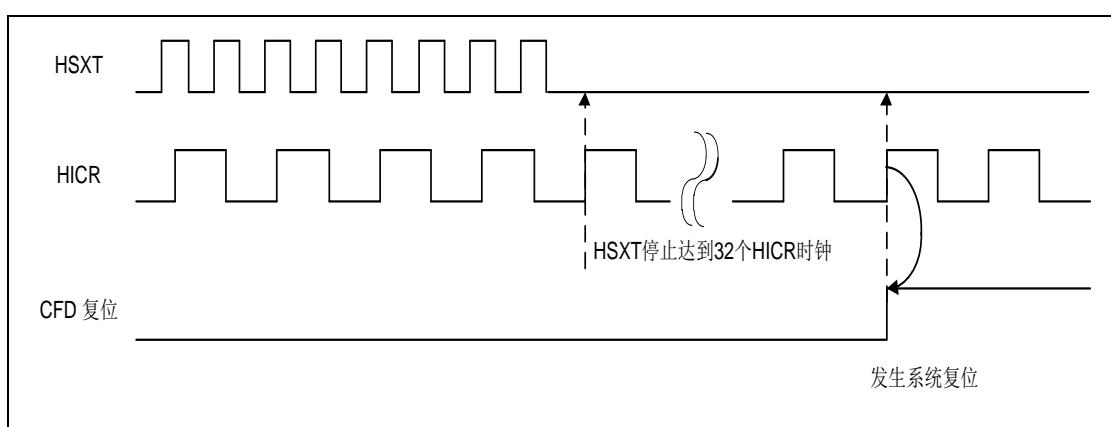
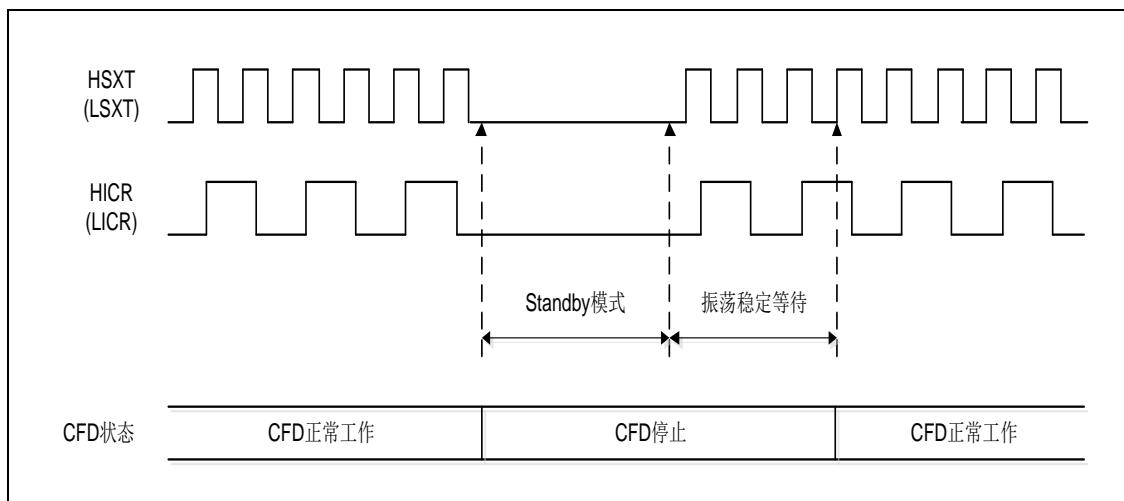


图 5.3.1-1 中，外部高速时钟 HSXT 停止振荡的时间达到 32 个 HICR 时钟后，产生 CFD 复位或中断。

外部低速时钟 LSXT 的检测原理与图 5.3.1-1 类似，不同点是计数时钟为 LICR。

图 5.3.1-2 时钟失效检测功能在 standby 模式下的状态



如图 5.3.1-2 所示：

- 1) standby 模式下，HSXT 和 LICR 都停止振荡，同时 CFD 功能自动停止。
- 2) 退出 standby 模式后，HSXT 和 HICR 开始振荡并启动时钟振荡稳定等待计数，此时 CFD 功能也自动停止。
- 3) 当振荡稳定等待结束后，CFD 功能打开。

5.3.2 时钟频率异常检测

时钟频率异常检测的被检测时钟是外部高速时钟。以外部高速时钟为计数时钟，计算内部高速 RC 分频时钟的两个上升沿。如果计数值超出设定的范围，则认为被检测时钟频率异常，此时产生系统复位或中断。产生的复位称作 AFD 复位，产生的中断称作 AFD 中断。工作原理如图 5.3.2-3 所示。

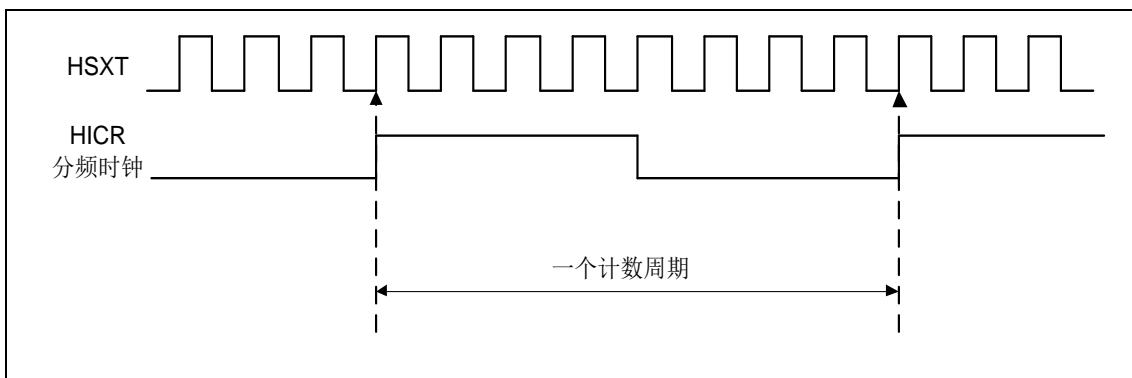
AFD 复位使能时，计数值第一次超出设定范围后，仅产生 AFD 中断。如果中断标志没有清零，当计数值再一次超出设定范围后，产生系统复位。

AFD 复位没有使能时，不会发生 AFD 系统复位。

时钟频率异常检测有以下特点：

- 被检测时钟只有外部高速时钟
- 时钟振荡稳定等待时间内，检测电路停止工作
- 外部高速晶振停止振荡之后，检测电路停止工作
- 需要软件启动该功能

图 5.3.2-3 时钟频率异常检测工作原理



HSXT 在一个计数周期内的计数值与 HICR 精度有关。因此，设置频率检测窗口寄存器时，需要考虑 HICR 的精度。

产品的 HICR 精度值，请参考“数据手册”。

计算方法：

考虑 HICR 精度后，一个计数周期内的计数值表达式为：

$$\text{计数值} = \frac{1}{\text{HICR分频时钟频率} \times (1 \pm \frac{\text{HICR精度}}{100})} \times \text{HSXT频率}$$

根据以上公式，可以得到计数值的上限 B 和下限 A。

设置频率检测窗口寄存器时，窗口下限应比 A 小，窗口上限应比 B 大。

计算举例：

假设 HSXT 频率为 4MHz，HICR 频率为 4MHz，HICR 精度 $\pm 2\%$ ，计数周期为 HICR 的 1024 分频（寄存器 FCM_CTL 的 AFDCC 默认值）。

$$\text{计数值 A} = \frac{1}{\frac{4 \times 10^6}{1024} \times (1 + \frac{2}{100})} \times 4 \times 10^6 \approx 1004$$

$$\text{计数值 B} = \frac{1}{\frac{4 \times 10^6}{1024} \times (1 - \frac{2}{100})} \times 4 \times 10^6 \approx 1045$$

如果频率检测窗口范围为 5%，则窗口设置值为：

$$\text{窗口下限} = 1004 \times (1 - 5\%) = 953.8$$

$$\text{窗口上限} = 1045 \times (1 + 5\%) = 1097.25$$

HSXT 频率低于 4MHz 时，计数值应小于 953.8；HSXT 频率高于 4MHz 时，计数值应大于 1097.25。所以，HSXT 频率范围为：

$$\text{频率下限} = 953.8 / (1 / (4 \times 0.98) \times 1024) \approx 3.65 \text{ MHz}$$

$$\text{频率上限} = 1097.25 / (1 / (4 \times 1.02) \times 1024) \approx 4.37 \text{ MHz}$$

5.4 设置步骤

图 5.4-1 时钟失效检测功能设置步骤

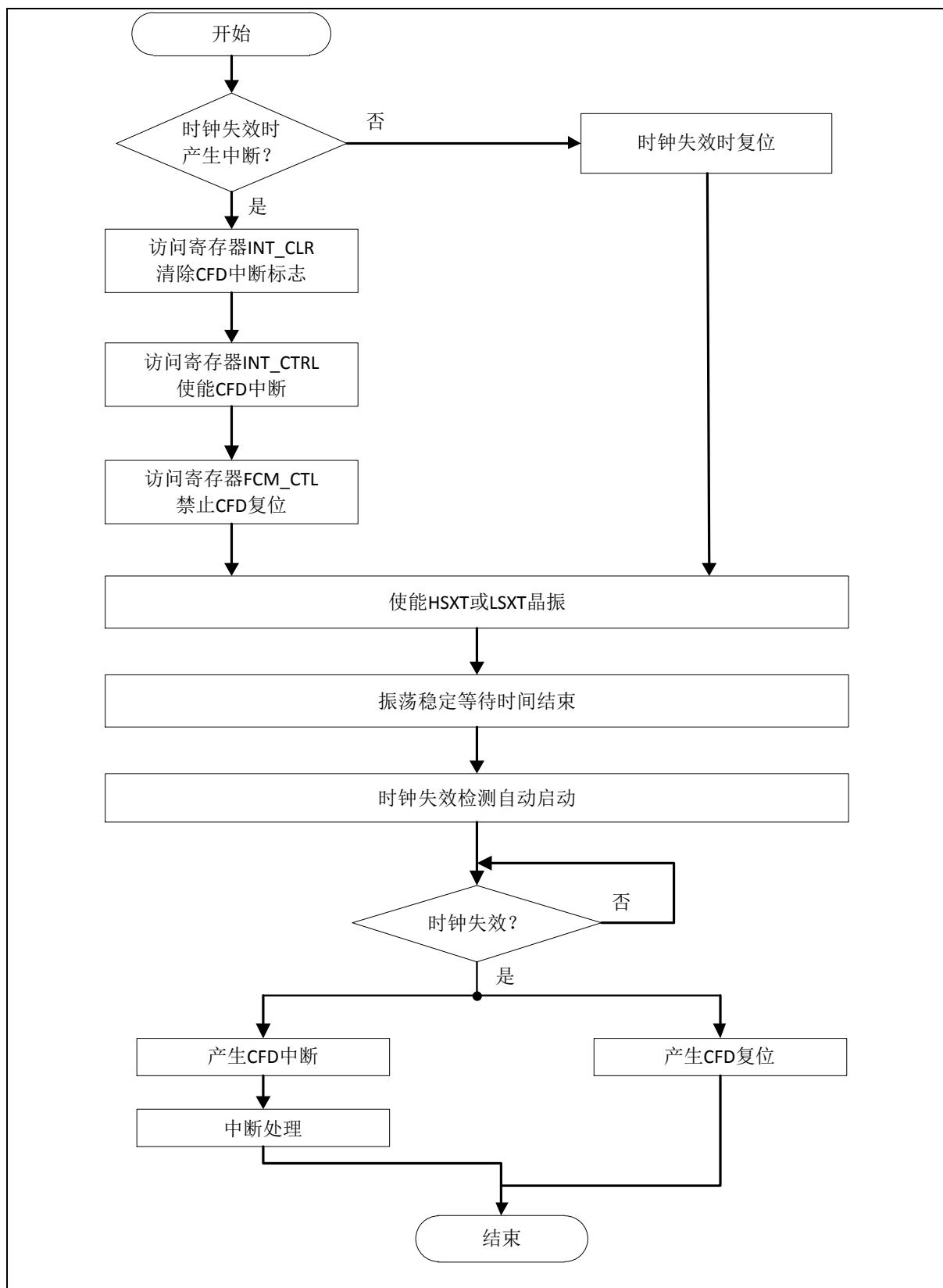
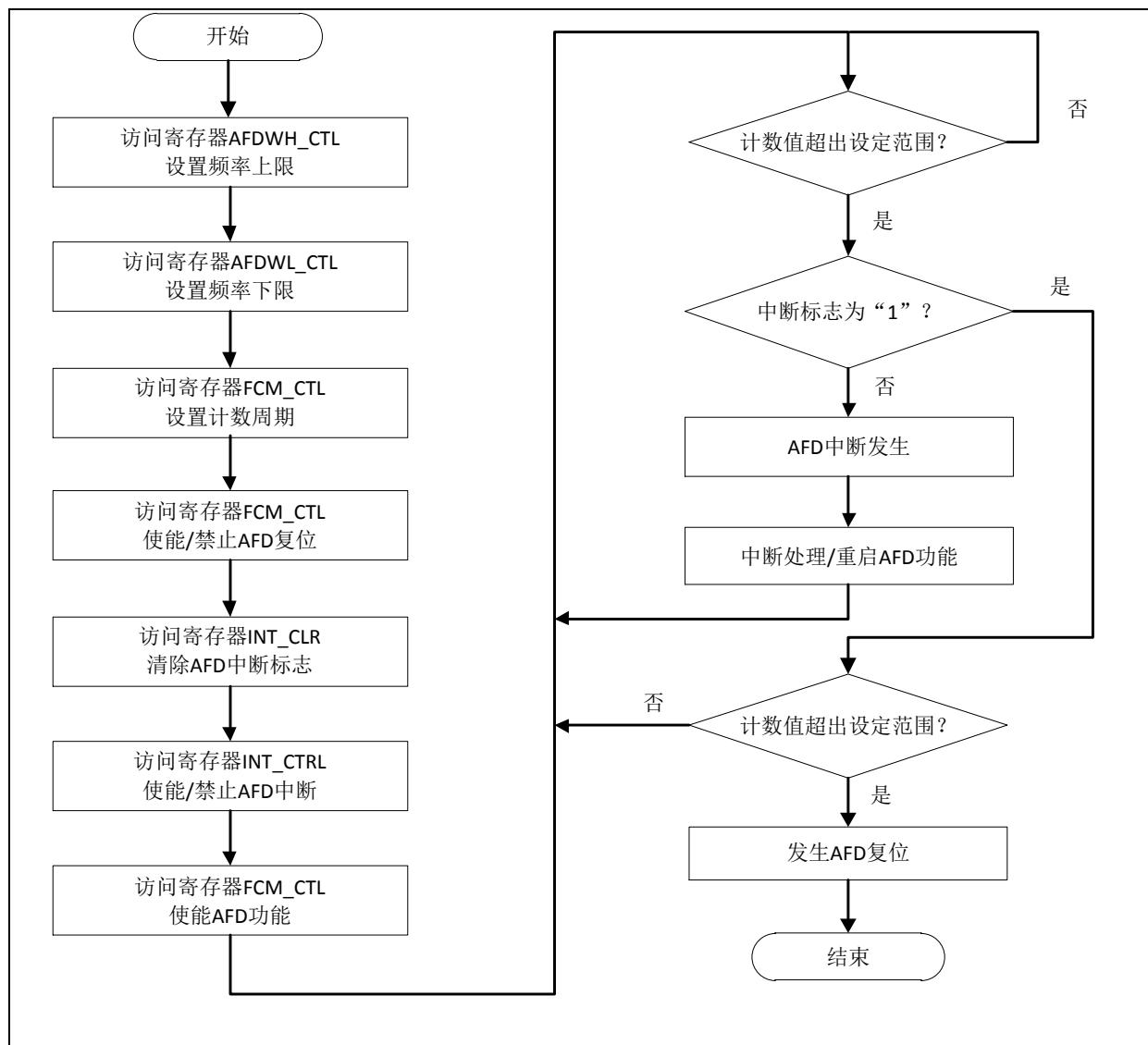


图 5.4-2 时钟频率异常检测功能设置步骤



5.5 寄存器

表 5.5-1 时钟监视器的寄存器列表

缩写	寄存器名称	章节
FCM_CTL	时钟监视器控制寄存器	5.5.1
FCM_STR	时钟监视器状态寄存器	5.5.2
AFDWH_CTL	频率检测窗口上限设置寄存器	5.5.3
AFDWL_CTL	频率检测窗口下限设置寄存器	5.5.4
AFDC_CTL	频率检测计数值寄存器	5.5.5

5.5.1 时钟监视器控制寄存器 (FCM_CTL)

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	保留		AFDCC		保留		AFDRE	AFDE
属性	-		R/W		-		R/W	R/W
初始值	-		111		-		0	0

位	7	6	5	4	3	2	1	0
字段名		保留		SFCMRE	MFCMRE	SFCME	MFCME	
属性		-		R/W	R/W	R/W	R/W	
初始值		-		1	1	1	1	

寄存器功能

[bit15]保留位，读数据为0，写无效。

[bit14:12] AFDCC[2:0]: AFD 计数周期设置位

Bit14:12		说明
写	101	内部高速RC时钟（HICR）的256分频
	110	内部高速RC时钟（HICR）的512分频
	111	内部高速RC时钟（HICR）的1024分频（初始值）
	其他	设置无效
读		读取当前寄存器的值

[bit11:10]保留位，读数据为0，写无效。

[bit9] AFDRE: AFD 复位使能位

bit		说明
写	0	禁止AFD复位功能（初始值）
	1	使能AFD复位功能

读	读取当前寄存器的值
---	-----------

[bit8] AFDE: AFD 功能使能位

bit		说明
写	0	禁止AFD功能 (初始值)
	1	使能AFD功能
读		读取当前寄存器的值

[bit7:4] 保留位，读数据为 0，写无效。

[bit3] SFCMRE: LSXT 失效检测复位使能位

bit		说明
写	0	禁止LSXT失效检测复位功能
	1	使能LSXT失效检测复位功能 (初始值)
读		读取当前寄存器的值

[bit2] MFCMRE: HSXT 失效检测复位使能位

bit		说明
写	0	禁止HSXT失效检测复位功能
	1	使能HSXT失效检测复位功能 (初始值)
读		读取当前寄存器的值

[bit1] SFCME: LSXT 失效检测功能使能位

不使用该功能时，建议上电复位之后关闭。

bit		说明
写	0	禁止LSXT失效检测功能
	1	使能LSXT失效检测功能 (初始值)
读		读取当前寄存器的值

[bit0] MFCME: HSXT 失效检测功能使能位

不使用该功能时，建议上电复位之后关闭。

bit		说明
写	0	禁止 HSXT 失效检测功能
	1	使能 HSXT 失效检测功能 (初始值)
读		读取当前寄存器的值

注意：

- 软件复位不会复位该寄存器。

5.5.2 时钟监视器状态寄存器 (FCM_STR)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留						SCFDF	MCFDF
属性	-						R	R
初始值	-						0	0

寄存器功能

[bit7:2] 保留位，读数据为 0，写无效。

[bit1] SCFDF: LSXT 失效标志位

bit	说明	
写	无效	
读	0	LSXT工作正常（初始值）
	1	检测到LSXT已失效

[bit0] MCFDF: HSXT 失效标志位

bit	说明	
写	无效	
读	0	HSXT工作正常（初始值）
	1	检测到HSXT已失效

注意：

- 读本寄存器的操作会同时清零该寄存器。

5.5.3 频率检测窗口上限设置寄存器 (AFDWH_CTL)

寄存器配置

位	15							0
字段名	AFDWH							
属性	R/W							
初始值	0xffff							

寄存器功能

[bit15:0] AFDWH[15:0]: 频率检测窗口上限设置位

bit	说明
写	可以写任意值
读	读取当前寄存器的值

注意：

- 软件复位不会复位该寄存器

5.5.4 频率检测窗口下限设置寄存器（AFDWL_CTL）

寄存器配置

位	15							0
字段名	AFDWL							
属性	R/W							
初始值	0x0000							

寄存器功能

[bit15:0] AFDWL[15:0]: 频率检测窗口下限设置位

bit	说明
写	可以写任意值
读	读取当前寄存器的值

注意：

- 软件复位不会复位该寄存器。

5.5.5 频率检测计数值寄存器（AFDC_CTL）

寄存器配置

位	15							0
字段名	AFDC							
属性	R							
初始值	0x0000							

寄存器功能

[bit15:0] AFDC[15:0]: 频率检测计数值设置位

bit	说明
写	无效
读	读取AFD计数器的当前计数值

注意：

- 有异常发生时，计数值会保留。
- 软件复位不会复位该寄存器。

5.6 注意事项

- 时钟监视器相关的中断设置，请参考“时钟”章节。
- 有关时钟故障检测和异常频率检测复位源的详细说明，参见“复位”章节。
- 发生 CFD 或 AFD 复位后，主时钟模式将切换到 HICR。同时，不要再使用有故障的时钟。
- 关于时钟频率异常检测的设置
 - 在使能 AFD (AFDE=1) 功能前，应先设置好计数周期 (AFDCC)，复位使能 (AFDRE) 和检测窗口 (AFDWH/AFDWL)。
 - 如果要改变 AFDCC/AFDRE/AFDWH/AFDWL 中的任意一项设置，请先停止 AFD 功能。

● 关于 AFD 中断和复位使能

使用 AFD 功能 (AFDE=1) 之前，请配置寄存器 INT_CTRL 中 AFD 中断使能位 (FCSEN)，和寄存器 FCM_CTL 中 AFD 复位使能位 (AFDRE)。

详细描述见表 5.6-1。

表 5.6-1 AFD 功能和 FCSEN, AFDRE 设置

	AFDE=1	AFDE=0
FCSEN=0 AFDRE=0	使能AFD功能 不产生中断 不发生复位	禁止AFD功能
FCSEN=0 AFDRE=1	使能AFD功能 不产生中断 检测到异常发生复位	禁止AFD功能
FCSEN=1 AFDRE=0	使能AFD功能 检测到异常产生中断 不发生复位	禁止AFD功能
FCSEN=1 AFDRE=1	使能AFD功能 第一次检测到异常产生中断 再次检测到异常发生复位	禁止AFD功能

● 关于关闭 CFD 功能和外部复位

选择 HSXT 或 LSXT 作为主时钟时，如果 CFD 功能关闭，时钟发生故障不会触发 CFD 复位，此时外部引脚复位也不能复位系统。所以，如果没有特殊原因，请不要关闭 CFD 功能。

● HICR 精度对 AFD 的影响

AFD 的检测结果受 HICR 精度的影响。设置 AFD 检测窗口值时，请将 HICR 的精度考虑在内。AFD 功能启动后，请不要改变 HICR 的频率。

6. 复位

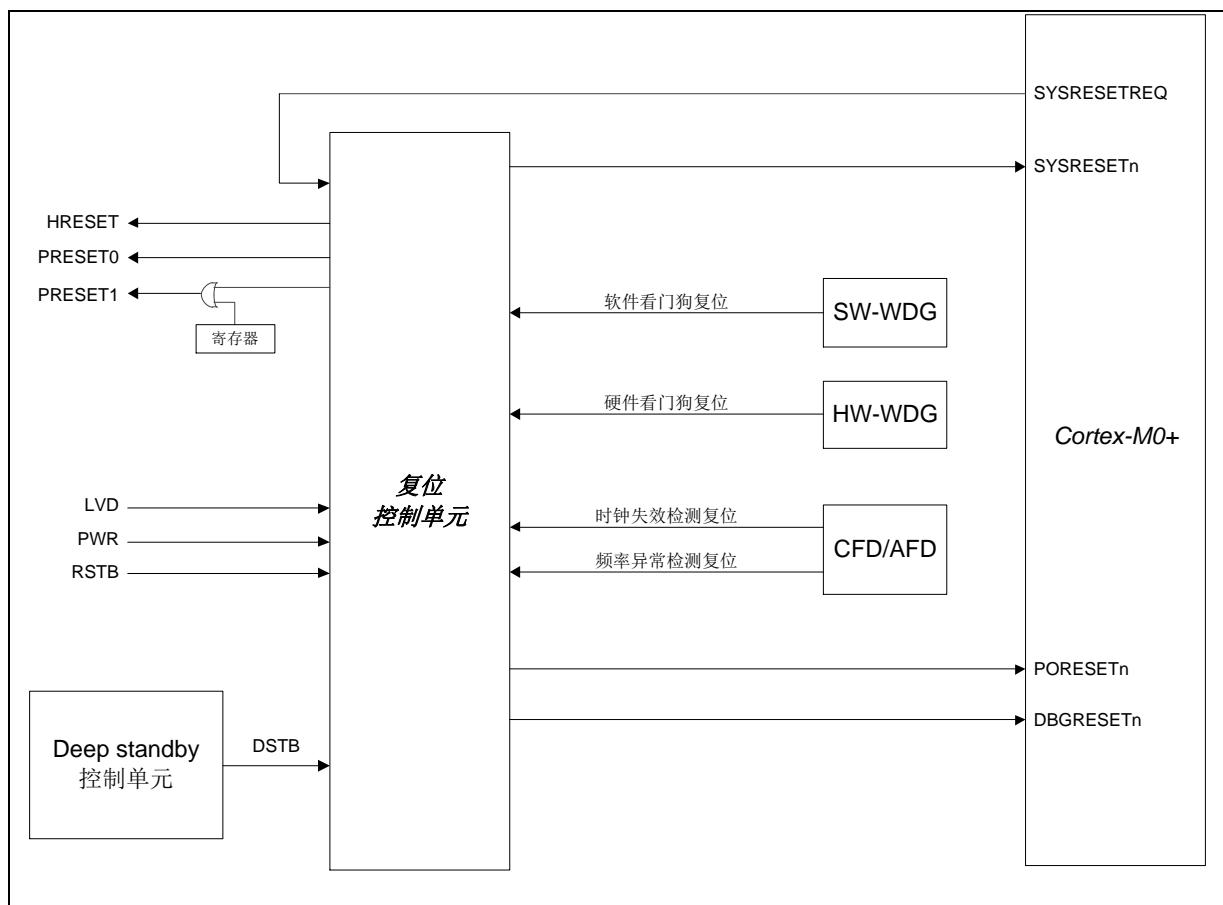
6.1 概述

MCU 有以下复位源。根据接收到的复位源，可发出复位命令并初始化芯片。

- 上电复位
- RSTB 引脚复位
- 低电压检测复位
- 硬件监视定时器复位
- 软件监视定时器复位
- 时钟失效检测复位
- 时钟频率异常检测复位
- 软件复位
- 深度待机转换复位

6.2 配置

图 6.2-1 复位系统原理框图



- | | |
|--------------|--|
| PWR: | 上电复位 |
| RSTB: | RSTB 引脚复位 |
| LVD: | 低电压检测复位 |
| HRESET: | AHB 总线复位 |
| PRESET0: | APB0 总线复位 |
| PRESET1: | APB1 总线复位。寄存器控制 APB1 总线复位，请参考第 2 章“时钟” |
| SYSRESETn: | Cortex-M0+的系统复位 |
| SYSRESETREQ: | Cortex-M0+内部复位控制寄存器的 SYSRESETREQ 位 |
| PORESETn: | Cortex-M0+的上电复位 |
| DBGRESETn: | Cortex-M0+的 SW-DP 复位 |
| DSTB: | 深度待机转换复位 |

6.3 操作

6.3.1 复位源

上电复位 (PWR)

产生条件	硬件检测到电源上升沿时，产生上电复位信号
清除条件	发生复位后，自动清除复位信号
初始化对象	所有寄存器和硬件
标志位	复位源寄存器 STAT 的 bit0 (PWR)

RSTB 引脚复位 (RSTB)

产生条件	外部引脚 RSTB 输入低电平 (低电平时间长度参考“数据手册”), 产生 RSTB 复位
清除条件	外部引脚 RSTB 输入高电平，清除复位信号
初始化对象	除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit1 (RSTB)

注意：

- RAM 中的数据不受 RSTB 复位影响。

低电压检测复位 (LVD)

产生条件	电源电压下降到一定阈值时，产生 LVD 复位
清除条件	电源电压上升到一定阈值时，清除复位信号
初始化对象	除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit3 (LVD)

软件监视定时器复位 (SWDGR)

产生条件	软件监视定时器计数值溢出时，产生 SWDGR 复位
清除条件	发生复位后，自动清除复位信号
初始化对象	除以下部分的所有寄存器和硬件 - 调试电路

	<ul style="list-style-type: none"> - 部分 RTC 寄存器 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit4 (SWWDG)

硬件监视定时器复位 (HWDGR)

产生条件	硬件监视定时器计数值溢出时，产生 HWDGR 复位
清除条件	发生复位后，自动清除复位信号
初始化对象	<ul style="list-style-type: none"> 除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit5 (HWWDG)

时钟失效检测复位 (CF)

产生条件	外部时钟 HSXT 或 LSXT 发生故障时，产生 CF 复位
清除条件	发生复位后，自动清除复位信号
初始化对象	<ul style="list-style-type: none"> 除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 部分时钟失效检测电路 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit6 (CF) 时钟监视器状态寄存器 FCM_STR 的 bit0(MCFDF)或 bit1(SCFDF)

时钟频率异常检测复位 (AF)

产生条件	外部时钟 HSXT 频率超出设定范围时，产生 FCS 复位
清除条件	发生复位后，自动清除复位信号
初始化对象	<ul style="list-style-type: none"> 除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 部分时钟频率异常检测电路 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit7 (AF)

软件复位 (SW)

产生条件	写 Cortex-M0+内部复位控制寄存器的 SYSRESETREQ 位，产生软件复位
清除条件	发生复位后，自动清除复位信号
初始化对象	除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 部分时钟控制电路 - 部分时钟失效检测电路 - 部分时钟频率异常检测电路 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	复位源寄存器 STAT 的 bit8 (SW)

深度待机转换复位 (DSTB)

产生条件	系统切换到深度待机模式时，产生 DSTB 复位
清除条件	退出深度待机模式后，清除复位信号
初始化对象	除以下部分的所有寄存器和硬件 - 调试电路 - 部分 RTC 寄存器 - 部分 GPIO 寄存器 - 复位源寄存器 (STAT) - 低电压检测电路 - 低功耗模式控制电路 - 备份寄存器 (BUR01 ~ BUR16)
标志位	深度待机复位唤醒源寄存器 (WAKE_RSR) 或 深度待机中断唤醒源寄存器 (WAKE_ISR) 详细描述请参考第 8 章 “低功耗模式”

注意：

- 关于 Cortex-M0+内部复位控制寄存器的详细描述，请参考“ARMv6-M 结构参考手册”。
- 复位源寄存器仅被上电复位信号复位。

6.3.2 MCU 内部复位

6.3.2.1 Cortex-M0+的复位

输入到 Cortex-M0+的复位有：PORRESETn, SYSRESETn 和 DBGRESETn

上电复位 PORRESETn

复位源	上电复位 (PWR) 低电压检测复位 (LVD) 深度待机转换复位 (DSTB)
-----	--

系统复位 SYSRESETn

复位源	上电复位 (PWR) 低电压检测复位 (LVD) RSTB 引脚复位 (RSTB) 硬件监视定时器复位 (HWDGR) 软件监视定时器复位 (SWDGR) 时钟失效检测复位 (CF) 时钟频率异常检测复位 (AF) 软件复位 (SW) 深度待机转换复位 (DSTB)
-----	---

SW-DP 复位 DBGRESETn

复位源	上电复位 (PWR) 低电压检测复位 (LVD) 深度待机转换复位 (DSTB)
-----	--

6.3.2.2 外设的复位

输入到外设电路的总线复位有：HRESET, PRESET0 和 PRESET1。

AHB 总线复位 HRESET

复位源	上电复位 (PWR) 低电压检测复位 (LVD) RSTB 引脚复位 (RSTB) 硬件监视定时器复位 (HWDGR) 软件监视定时器复位 (SWDGR) 时钟失效检测复位 (CF) 时钟频率异常检测复位 (AF) 软件复位 (SW) 深度待机转换复位 (DSTB)
-----	---

APB0 总线复位 PRESET0

复位源	上电复位 (PWR) 低电压检测复位 (LVD) RSTB 引脚复位 (RSTB) 硬件监视定时器复位 (HWDGR) 软件监视定时器复位 (SWDGR) 时钟失效检测复位 (CF) 时钟频率异常检测复位 (AF) 软件复位 (SW) 深度待机转换复位 (DSTB)
-----	---

APB1 总线复位 PRESET1

复位源	上电复位 (PWR) 低电压检测复位 (LVD) RSTB 引脚复位 (RSTB) 硬件监视定时器复位 (HWDGR) 软件监视定时器复位 (SWDGR) 时钟失效检测复位 (CF) 时钟频率异常检测复位 (AF) 软件复位 (SW) 深度待机转换复位 (DSTB) APB1 时钟分频寄存器 (APBC1_PSR)
-----	---

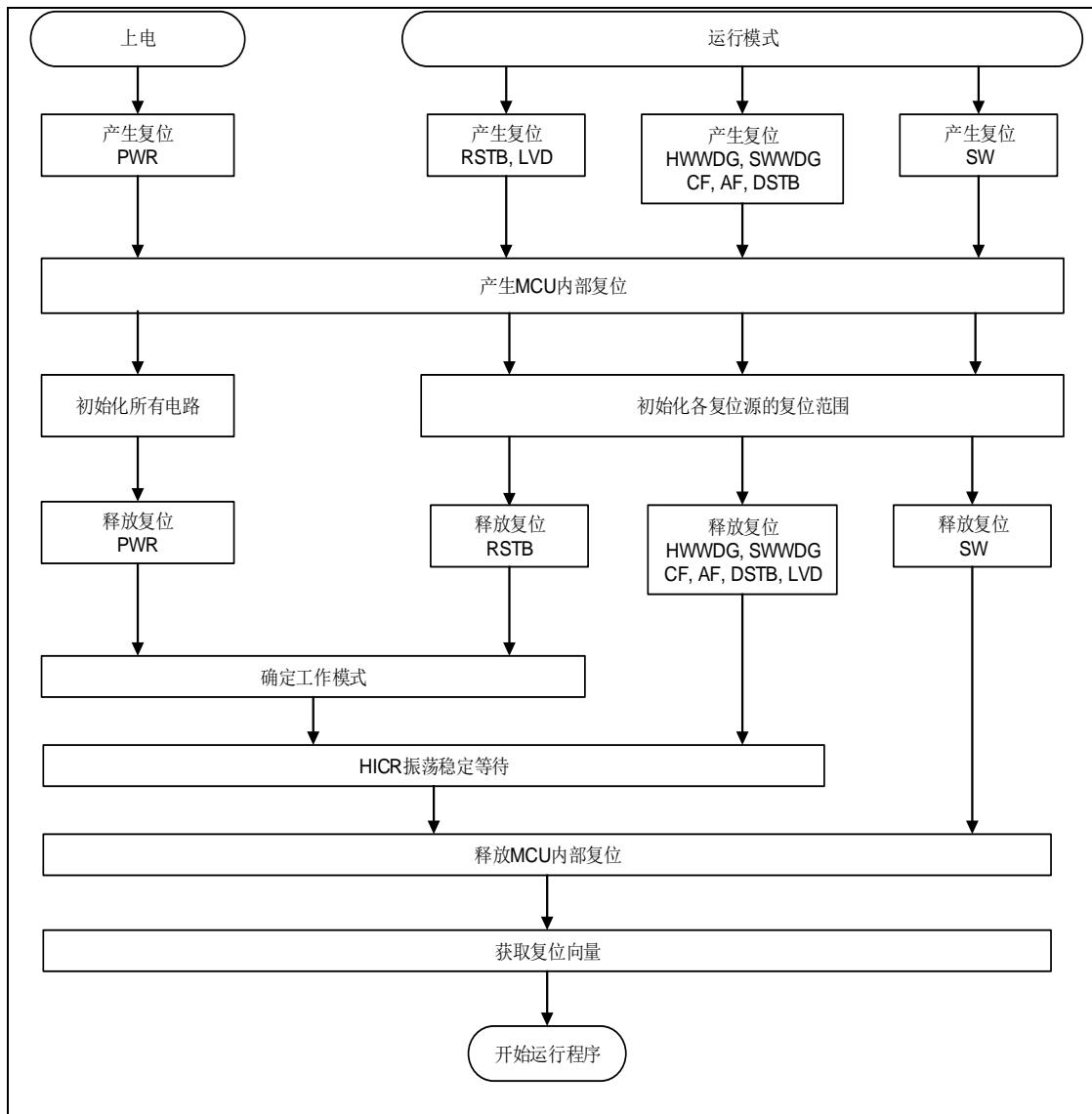
注意：

- 各外设的复位源可能略有不同，详细复位源描述请参考相关章节。

6.3.3 复位顺序

从复位开始到初始化操作完成的一系列动作，称为复位顺序。

图 6.3.3-1 复位状态切换图



关于工作模式

MODE 引脚的电平决定 MCU 的工作模式。当 PWR 和 RSTB 复位释放时工作模式将确定，其他复位源不会影响工作模式。

关于获取复位向量

MCU 内部复位清除后，CPU 开始获取复位向量。CPU 将获取的复位向量载入 PC 并开始运行程序。

复位清除

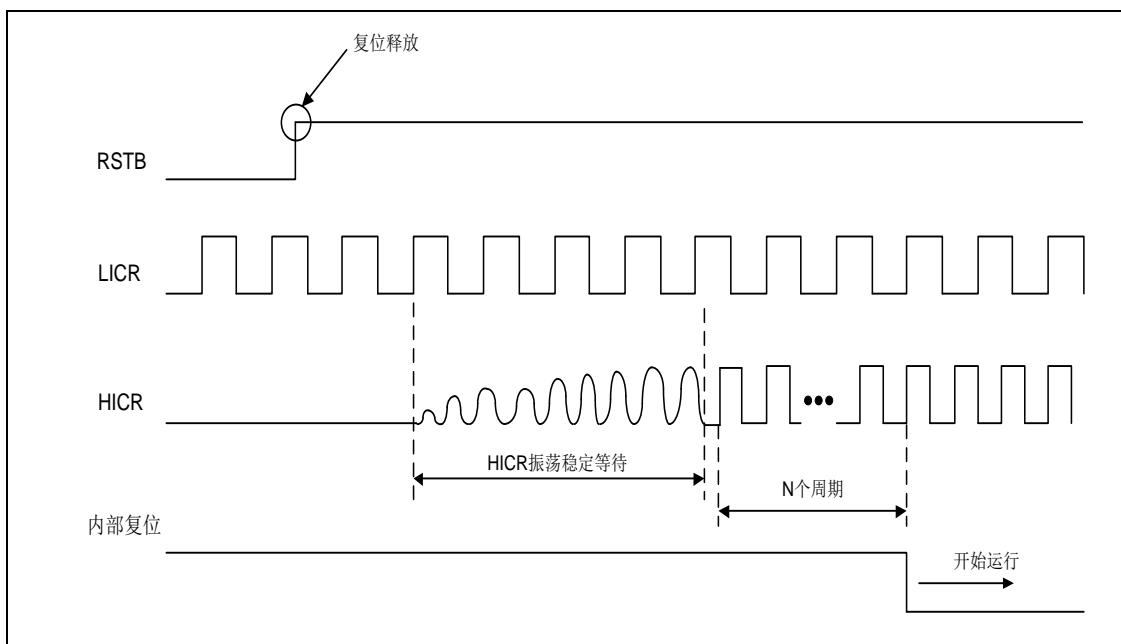
当复位源清除后，内部复位信号将在一定时间过后释放。例如，需要等待 HICR 振荡稳定。

6.3.4 释放复位后的操作

PWR, LVD, RSTB, HWDGR, SWDGR, CF, AF

图 6.3.4-2 是 RSTB 引脚复位释放后的操作示意图。

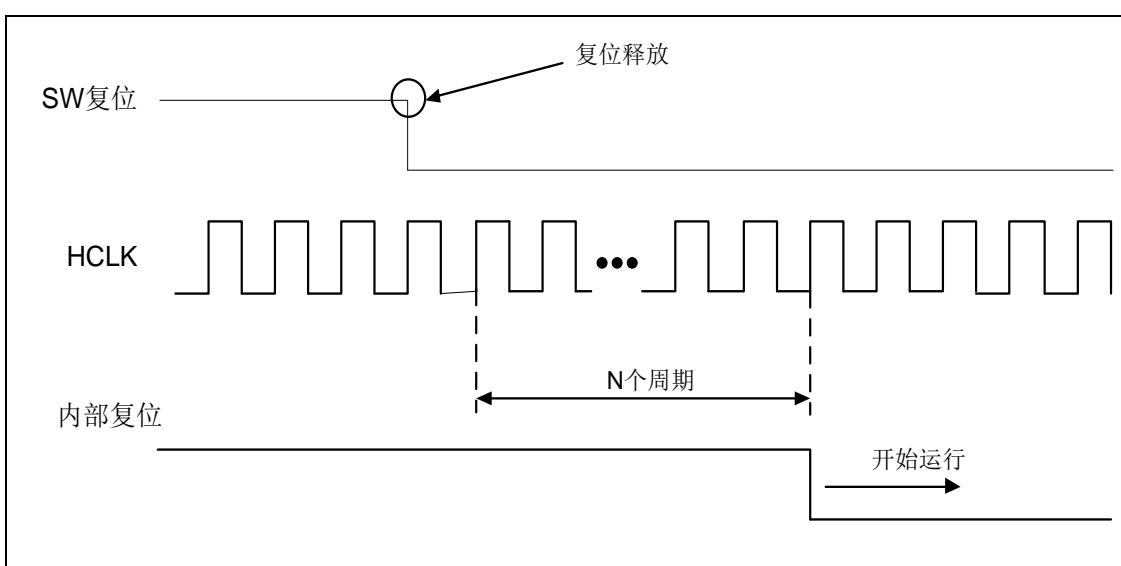
图 6.3.4-2 RSTB 引脚复位释放后的操作



软件复位

图 6.3.4-3 是软件复位释放后的操作示意图。

图 6.3.4-3 是软件复位释放后的操作



6.4 寄存器

表 6.4-1 寄存器列表

缩写	寄存器名称	章节
STAT	复位源寄存器	6.4.1

6.4.1 复位源寄存器 (STAT)

该寄存器显示当前发生的复位源。可以初始化该寄存器的复位有：上电复位、低电压检测复位和深度待机转换复位。

对该寄存器进行读操作后，所有位都会被清零。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	保留							SW
属性	-							R
初始值	-							0

位	7	6	5	4	3	2	1	0
字段名	AF	CF	HWWWDG	SWWDG	LVD	保留	RSTB	PWR
属性	R	R	R	R	R	-	R	R
初始值	0	0	0	0	0	-	0	1

寄存器功能

[bit15:9] 保留位, 读数据为 0, 写无效。

[bit8] SW: 软件复位标志, 此标志位显示复位由写 Cortex-M0+复位控制寄存器引起。发生软件复位时, 此标志位置 1。读操作清零该位。

Bit		说明
读	0	没有发生软件复位
	1	发生了软件复位
写		无效

[bit7] AF: HSXT 频率异常复位标志, 此标志位显示复位由外部高速时钟 HSXT 频率异常引起。

当 HSXT 频率超过设定的范围, 发生系统复位后, 此标志位置 1。

bit		说明
读	0	没有发生 FCS 复位
	1	发生了 FCS 复位
写		无效

[bit6] CF: HSXT 或 LSXT 时钟失效复位标志, 此标志位显示复位由外部高速时钟

HSXT 失效，或外部低速时钟 LSXT 失效引起。当 HSXT 或 LSXT 时钟失效，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生时钟失效复位
	1	发生了时钟失效复位
写		无效

[bit5] HWWDG：硬件监视定时器复位标志，此标志位显示复位由硬件监视定时器定时器溢出引起。当硬件监视定时器定时器溢出，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生硬件监视定时器复位
	1	发生了硬件监视定时器复位
写		无效

[bit4] SWWDG：软件监视定时器复位标志，此标志位显示复位由软件监视定时器定时器溢出引起。当软件监视定时器定时器溢出，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生软件监视定时器复位
	1	发生了软件监视定时器复位
写		无效

[bit3] LVD：低电压检测复位标志，此标志位显示复位由低电压检测电路引起。当电源电压低于设定值，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生低电压检测复位
	1	发生了低电压检测复位
写		无效

[bit2] 保留位，读数据为 0，写无效。

[bit1] RSTB：RSTB 引脚复位标志，此标志位显示复位由 RSTB 引脚引起。RSTB 引脚输入“L”电平，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生 RSTB 复位
	1	发生了 RSTB 复位
写		无效

[bit0] PWR: 上电复位标志

此标志位显示复位由上电复位或深度待机复位引起。

当检测到电源上升沿或进入深度待机模式，发生系统复位后，此标志位置 1。

bit		说明
读	0	没有发生上电复位
	1	发生了上电复位
写		无效

注意：

- 深度待机复位的详细描述，请参考“低功耗模式”章节。

7. 低电压检测

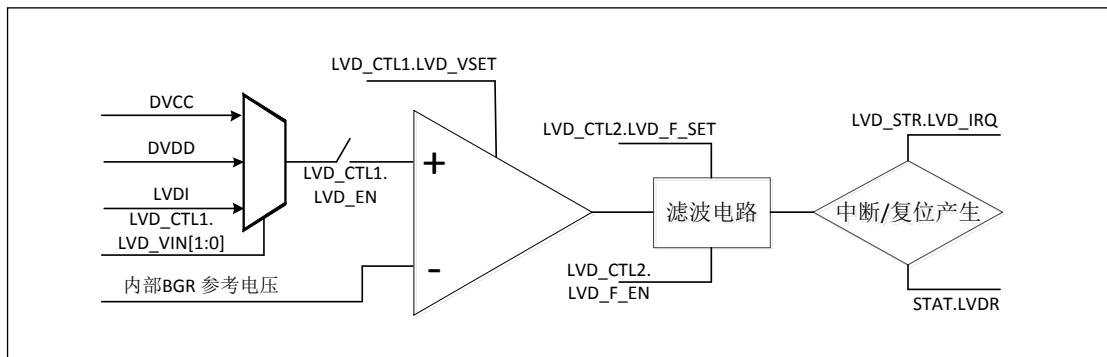
7.1 概述

本系列有一个 LVD，可以检测电源电压 DVCC、内部电压 DVDD 和外部端口 LVDI。LVD 阈值可调，带有迟滞和滤波功能，可配置产生中断或复位。

LVD 可以唤醒系统低功耗模式（Standby 和 Deep Standby）。

7.2 工作原理

7.2.1 原理框图



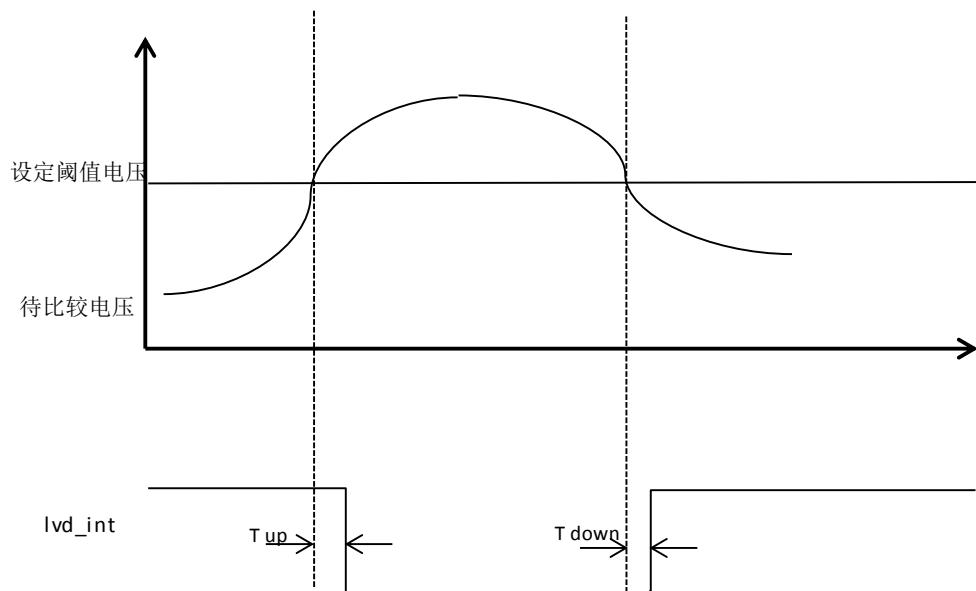
7.2.2 建立时间

当使用低电压检测器来监测外部电压或芯片工作电压变化时，需要内部 BGR 输出的参考电压为比较基准电压，BGR 的启动需要等待稳定时间。除此之外，低电压检测器也需要等待稳定时间，来建立低电压检测器的工作点。两者具体的稳定时间请参考数据手册。

7.2.3 响应时间

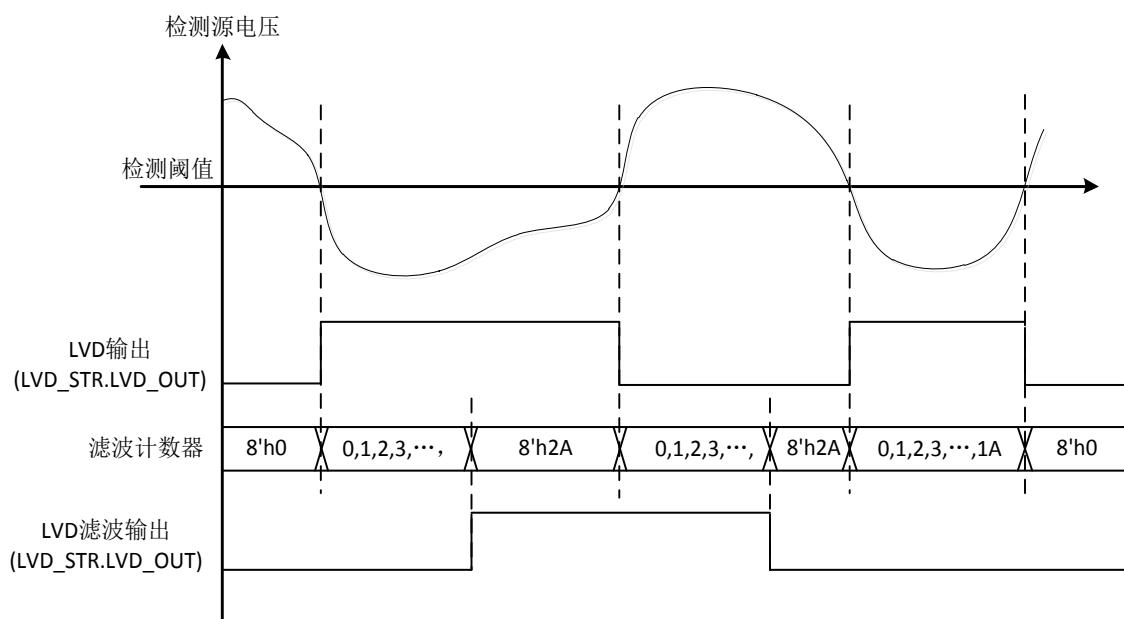
当低电压检测器的输入电压发生变换后，到输出结果产生翻转的这段时间被称为响应时间，小于响应时间的电压变化将被忽略。低电压检测器固有响应时间（比较器的响应时间）请参考数据手册，也可通过配置寄存器使能滤波电路。当滤波电路开启后，通过配置寄存器选择不同的滤波时间，以获得不同响应时间。

LVD 固定响应时间



如果开启了 LVD 滤波功能，可以通过设置 LVD_F_SET[2:0]获取不同的响应时间，下图为 LVD 滤波响应时间的示意图。

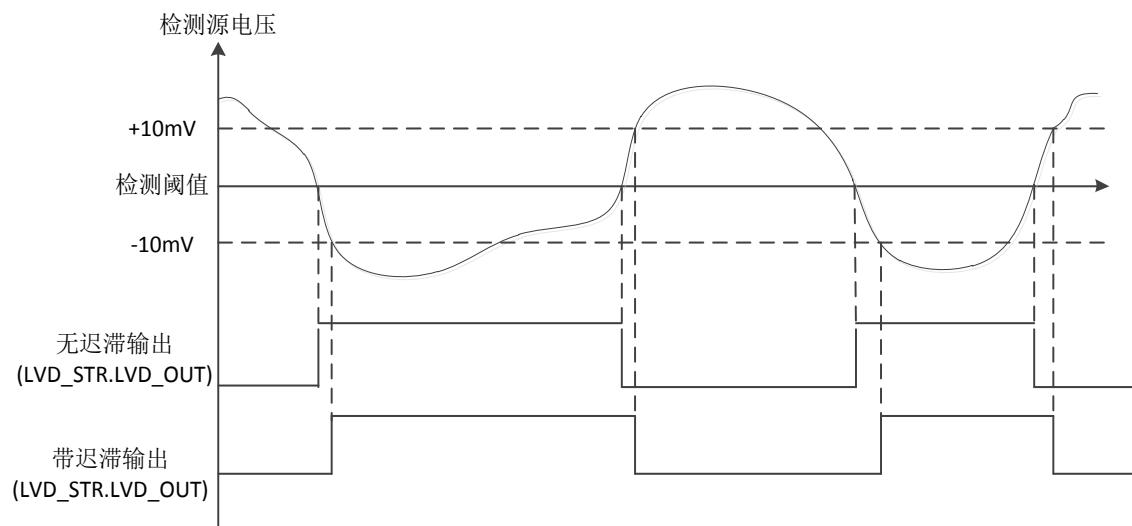
LVD 滤波响应时间



7.2.4 迟滞功能

通过配置寄存器可选择关闭或开启迟滞电路。若迟滞电路关闭，当输入电压高于 LVD 设定电压，结果则为“0”，当输入电压低于 LVD 设定电压，结果则为“1”；若迟滞电路开启，当输入电压高于设定电压+10mV 时，结果为“0”，当输入电压低于设定电压-10mV 时，结果为“1”。

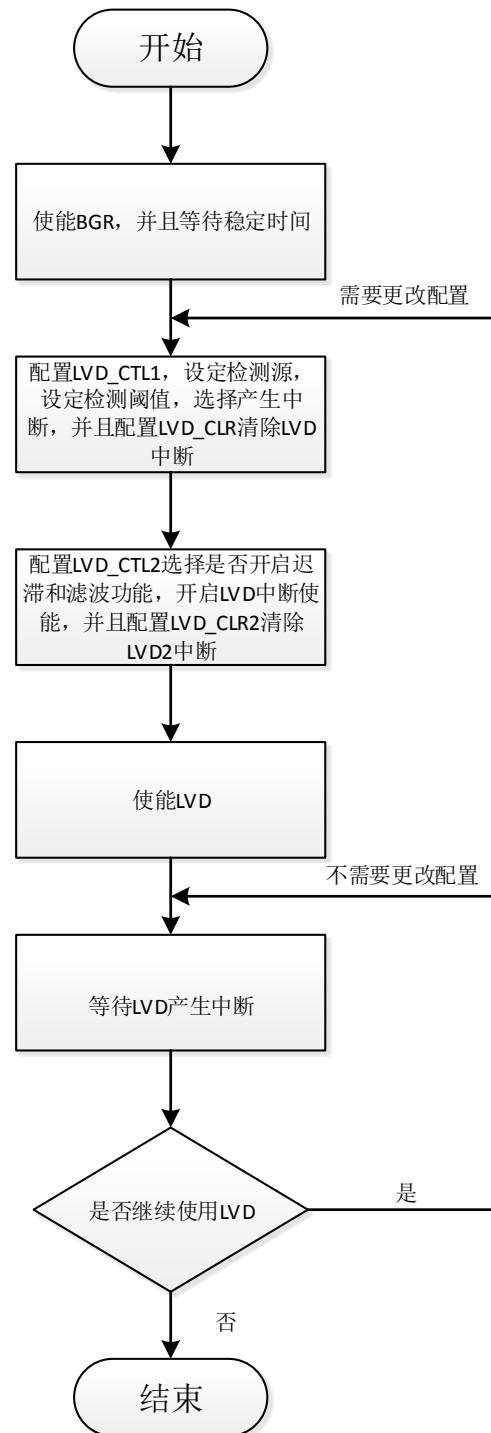
LVD迟滞功能示意图



7.3 操作流程

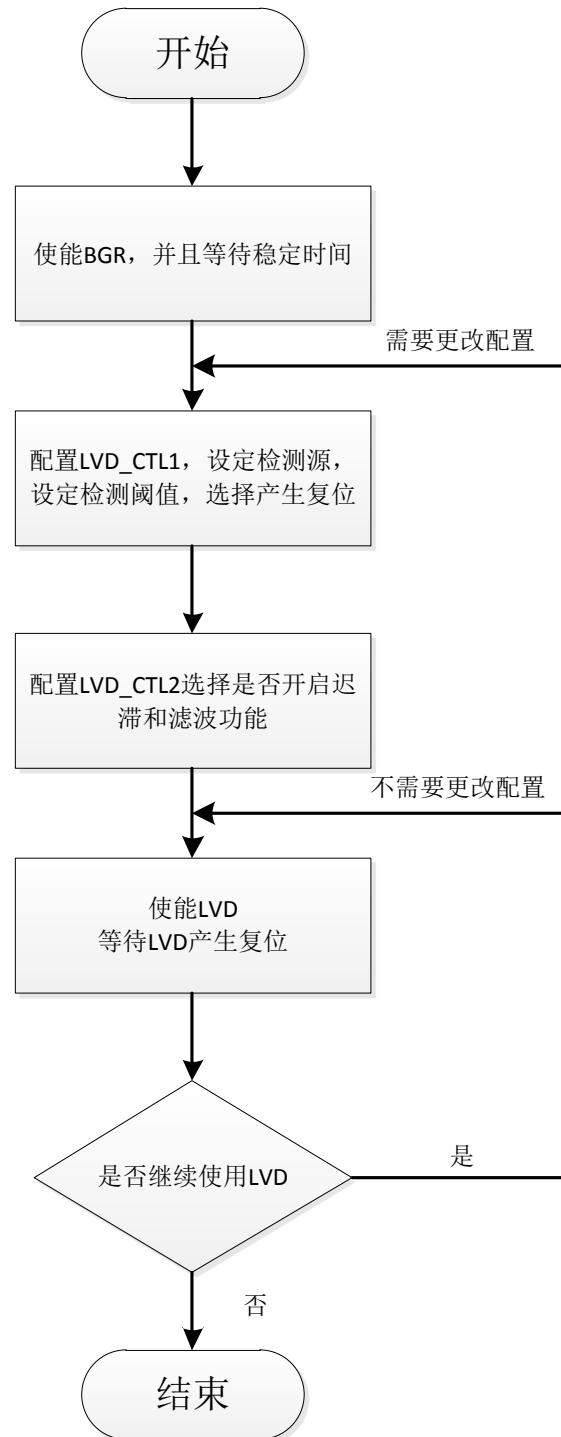
7.3.1 LVD 输出作为中断

LVD 产生中断流程如下图



7.3.2 LVD 输出作为复位

LVD 产生复位流程



7.4 寄存器

7.4.1 LVD1 控制寄存器 1(LVD_CTL1)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
字段名	-	-	-	-	-	-	-	-
初始值	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
字段名	LVD_VSET				LVD_VIN		LVD_SRST	LVD_EN
初始值	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LVD_EN: LVD 使能信号

- 0: 不使能 LVD
- 1: 使能 LVD

LVD_SRST: LVD 复位中断选择

- 0: LVD 输出中断信号
 - 1: LVD 输出复位信号
- LVD_VIN: LVD 检测信号来源选择
- 00: 检测 VCC
 - 01: 检测 VDD
 - 1X: 检测外部 PIN

LVD_VSET: LVD 低压检测阈值设置

- 0000: 2.00V±50mV
- 0001: 2.10V±50mV
- 0010: 2.20V±50mV
- 0011: 2.30V±50mV
- 0100: 2.40V±50mV
- 0101: 2.50V±50mV
- 0110: 2.61V±50mV
- 0111: 2.72V±50mV
- 1000: 2.83V±50mV
- 1001: 2.94V±50mV
- 1010: 3.05V±50mV
- 1011: 3.16V±50mV
- 1100: 3.27V±50mV
- 1101: 3.38V±50mV
- 1110: 3.49V±50mV
- 1111: 3.60V±50mV

注意:

- 本寄存器只会被上电复位初始化。

7.4.2 LVD1 控制寄存器 2(LVD_CTL2)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
字段名	-	-	-	-	-	-	-	-
初始值	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
字段名	-	LVD_IE		LVD_F_SET			LVD_F_EN	LVD_HYS_EN
初始值	0	0	0	0	0	0	0	0
R/W	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LVD_HYS_EN LVD 迟滞开关

0: 关闭迟滞开关

1: 打开迟滞开关, 典型值 20mV

LVD_F_EN: LVD 滤波开关

0: 滤波开关关闭

1: 滤波开关打开

LVD_F_SET: LVD 滤波时间设置

000: 16us (default)

001: 32us

010: 64us

011: 256us

100: 1ms

101: 4ms

110: 16ms

111: 64ms

LVD_IE: LVD 中断控制位

00: 不使能 LVD 中断

01: LVD 检测到电压低于检测阈值产生中断

10: LVD 检测到电压高于检测阈值产生中断

11: LVD 检测到电压高于检测阈值或低于检测阈值都产生中断

注意:

- 本寄存器只会被上电复位初始化。

7.4.3 LVD1 状态寄存器 1(LVD_STR)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
字段名	-	-	-	-	-	-	-	LVD_OUT
初始值	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
字段名	-	-	-	-	-	-	-	LVD_IRQ
初始值	0	0	0	0	0	0	0	0
R/W	R0	R						

LVD_IRQ: LVD 中断标志位

0: LVD 未产生中断

1: LVD 产生中断

LVD_OUT: LVD 输出原始状态, 经过滤波

0: 电压高于 LVD 检测电压

1: 电压低于 LVD 检测电压

注意:

- 本寄存器只会被上电复位初始化。

7.4.4 LVD1 中断清零寄存器 1(LVD_CLR)

bit	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
字段名	-	-	-	-	-	-	-	-
初始值	0	0	0	0	0	0	0	0
R/W	R0	R0	R0	R0	R0	R0	R0	R0

bit	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
字段名	-	-	-	-	-	-	-	LVD_CLR
初始值	0	0	0	0	0	0	0	1
R/W	R0	R1/W0						

LVD_CLR: LVD 中断清除位

W0: 写 0 清除中断

R1: 读取的值一直为 1

注意:

- 本寄存器只会被上电复位初始化。

8. 低功耗模式

8.1 概述

为了降低功耗，系统可以设置多种低功耗模式，Idle 模式、Standby 模式和 Deep Standby 模式。其中 Standby 模式又分为 RTC 模式和 STOP 模式，Deep Standby 模式又分为 Deep RTC 模式和 Deep STOP 模式。

按照 CPU 的相关操作具体分为如下工作模式：

- Run 模式
 - HICR Run 模式
 - HSXT Run 模式
 - LICR Run 模式
 - LSXT Run 模式
- 空闲模式
 - HICR Idle 模式
 - HSXT Idle 模式
 - LICR Idle 模式
 - LSXT Idle 模式
- Standby 模式
 - RTC 模式
 - STOP 模式
- Deep Standby 模式
 - Deep RTC 模式
 - Deep STOP 模式

8.1.1 Run 模式概述

Run 模式是指 CPU 可以工作运行，基本时钟由主控时钟分频得来，以用于为 CPU、AHB 和 APB 总线提供时钟，外设处于工作状态。

系统工作时钟可以进行切换，没有用到的时钟源可以关闭，以降低功耗。根据时钟源的不同 Run 模式可以分为以下几类：

- HICR Run 模式

在此工作模式下，HICR 作为系统时钟，频率可以选择 2/4/8/16MHz。如果不使用 HSXT 和 LSXT，可以将其关闭。LICR 在 Run 模式下是一直处于开启状态的。上电默认是 4M HICR Run 模式，如果发生了复位也将恢复到 4M HICR Run 模式。

- HSXT Run 模式

在此工作模式下，HSXT 作为系统时钟，时钟频率为 4-32MHz，取决于外接的晶振频率。如果不使用 HICR 和 LSXT，可以将其关闭。LICR 在 Run 模式下是一直处于开启状态的。

- LICR Run 模式

在此工作模式下，LICR 作为系统时钟，时钟频率为 32KHz。此工作模式下 HICR 和 HSXT 高频时钟是关闭的。

- LSXT Run 模式

在此工作模式下，LSXT 作为系统时钟，时钟频率为 32.768KHz。此工作模式下 HICR 和 HSXT 高频时钟是关闭的。LICR 在 Run 模式下是一直处于开启状态的。

8.1.2 Idle 模式概述

Idle 模式是指 CPU 停止执行指令，可以降低功耗，AHB 和 APB 时钟正常工作，大部分外设都处于工作状态。

根据时钟源的不同 Idle 模式可以分为以下几类：

- HICR Idle 模式

当 HICR 作为系统工作时钟源时，如果切换到 Idle 模式时，则进入到 HICR Idle 模式，频率可以选择 2/4/8/16MHz。如果不使用 HSXT 和 LSXT，可以将其关闭。LICR 在 Idle 模式下是一直处于开启状态的。

- HSXT Idle 模式

当 HSXT 作为系统工作时钟源时，如果切换到 Idle 模式时，则进入到 HSXT Idle 模式，时钟频率为 4-32MHz，取决于外接的晶振频率。如果不使用 HICR 和 LSXT，可以将其关闭。LICR 在 Idle 模式下是一直处于开启状态的。

- LICR Idle 模式

当 LICR 作为系统工作时钟源时，如果切换到 Idle 模式时，则进入到 LICR Idle 模式，时钟频率为 32KHz。此工作模式下 HICR 和 HSXT 高频时钟是关闭的。

- LSXT Idle 模式

当 LSXT 作为系统工作时钟源时，如果切换到 Idle 模式时，则进入到 LSXT Idle 模式，时钟频率为 32.768KHz。此工作模式下 HICR 和 HSXT 高频时钟是关闭的。LICR 在 Idle 模式

下是一直处于开启状态的。

8.1.3 Standby 模式概述

- RTC 模式

RTC 模式是 Standby 模式的一种，RTC 模式下会将 HSXT 和 HICR 关闭，LSXT 和 LICR 会根据应用选择只开一个或者是两个都开启。在 RTC 模式下除了 RTC、LCD、WDT、VC 和 LVD 工作外，其他功能都会关闭。

- STOP 模式

STOP 模式是 Standby 模式的一种，STOP 模式下所有的时钟都会关闭，在此模式下只有 VC 和 LVD 可以工作。

8.1.4 Deep Standby 模式概述

- Deep RTC 模式

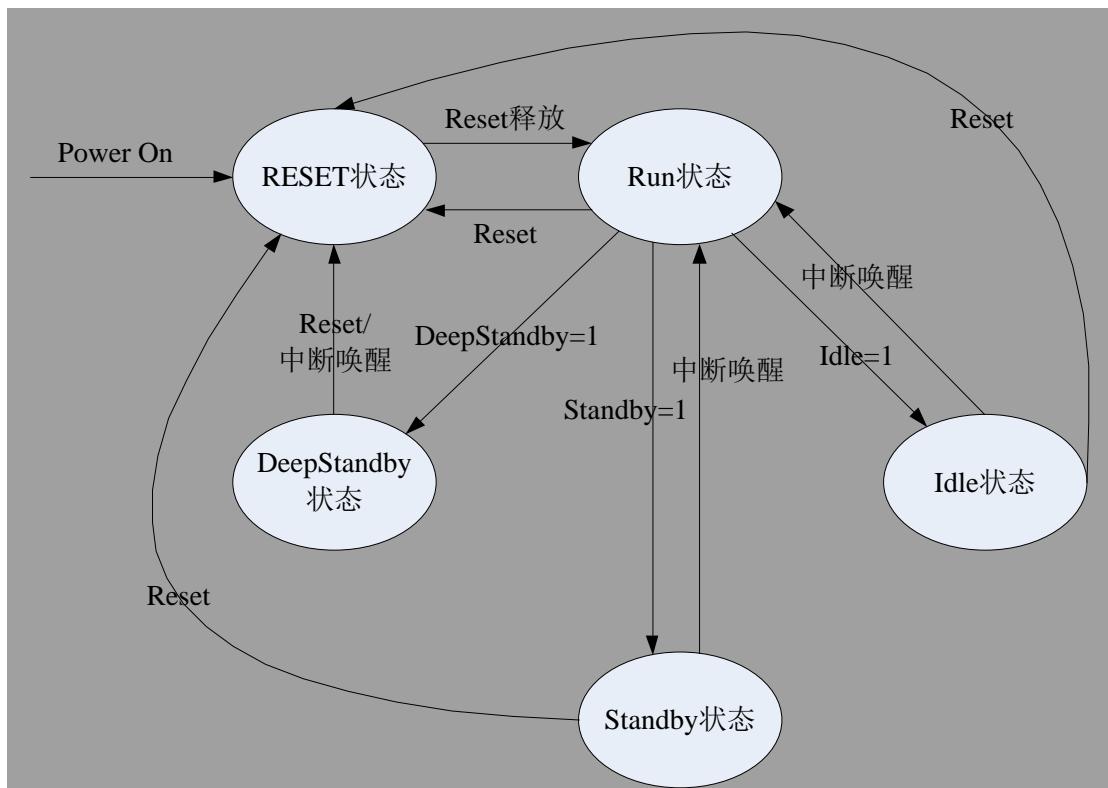
Deep RTC 模式是 Deep Standby 模式的一种，Deep RTC 模式下会将 HSXT 和 HICR 关闭，LSXT 和 LICR 会根据应用选择只开一个或者是两个都开启。此模式下只有 RTC、LCD、LVD 和用于唤醒的 GPIO 可以工作。片上 SRAM 会根据设置选择是否保留其中的数据。

- Deep STOP 模式

Deep STOP 模式是 Deep Standby 模式的一种，Deep Stop 模式下所有时钟都会关闭，在此模式下只有 LVD 和用于唤醒的 GPIO 可以工作。片上 SRAM 会根据设置选择是否保留其中的数据。

8.2 状态转换图

下图为 CPU 不同工作模式之间的状态转换图。



8.3 Idle 模式相关操作

8.3.1 Idle 模式时钟工作状态

下表给出了各个振荡器时钟以及 CPU、AHB 和 APB 总线时钟在不同 Idle 模式下的工作状态：

表 8.3.1-1 Idle 模式下时钟工作状态

	HICR Idle 模式	HSXT Idle 模式	LICR Idle 模式	LSXT Idle 模式
HICR	运行	根据设置决定	停止	停止
HSXT	根据设置决定	运行	停止	停止
LSXT	根据设置决定	根据设置决定	根据设置决定.	运行
LICR	运行	运行	运行	运行
CPU clock		停止		
AHB bus clock	HICR 时钟	HSXT 时钟	LICR 时钟	LSXT 时钟
APB bus clock	HICR 时钟	HSXT 时钟	LICR 时钟	LSXT 时钟

8.3.2 Idle 模式功能

- CPU 和片上存储

Idle 模式下 CPU 的时钟是停止的，AHB 和 APB 总线时钟可以正常执行。片上存储系统正常执行，数据保持。

- 外设

Idle 模式下大部分外设都可以工作，APB0 总线时钟在 Idle 模式下依然有效，APB1 BUS 时钟是否有效与 APBC1EN 的设置有关。

- RTC

RTC 在 Idle 模式下不受影响，RTC 会根据在进入 Idle 模式之前的设置正常运行。

- 时钟

每个时钟的状态请参考表 8.3.1-1。

- 复位和中断

复位和有效中断都可以唤醒 Idle 模式。

8.4 Standby 模式相关操作

8.4.1 Standby 模式时钟工作状态

Standby 模式包括了 RTC 模式和 STOP 模式。下表给出了各个振荡器时钟以及 CPU、AHB 和 APB 总线时钟在不同 Standby 模式下的工作状态：

表 8.4.1-1 Standby 模式下时钟工作状态

	RTC 模式	STOP 模式
HICR	停止	停止
HSXT	停止	
LICR	根据设置决定	
LSXT	根据设置决定	
CPU clock		
AHB bus clock	停止	
APB bus clock		

注意：

- RTC 模式下 LICR 和 LSXT 至少要开启一个，至于开启哪一个需要根据不同的设置来决定，也可以都开启。

8.4.2 RTC 模式功能

- CPU 和片上存储（SRAM 和 Flash）

在 RTC 模式下，CPU、AHB 和 APB 总线时钟会停止，片上存储时钟也会停止，但是片上存储中的数据会保持。

- 外设

在 RTC 模式下除了 RTC、LCD、WDT、VC 和 LVD 工作外，其他外设功能都会关闭。

- RTC

RTC 在 RTC 模式下不受影响，RTC 会根据在进入 RTC 模式之前的设置正常运行。

- 时钟

每个时钟的状态请参考表 8.4.1-1。

- 唤醒

参考表 8.6.1-1，各个模式的唤醒源。

- 总线

AHB 和 APB 总线在 RTC 模式下停止工作。

8.4.3 STOP 模式功能

- CPU 和片上存储 (SRAM 和 Flash)

在 STOP 模式下，CPU、AHB 和 APB 总线时钟会停止，片上存储时钟也会停止，但是片上存储中的数据会保持。

- 外设

在 STOP 模式下除了 VC 和 LVD 工作外，其他功能都会关闭。

- RTC

RTC 在 STOP 模式下停止工作。

- 时钟

每个时钟的状态请参考表 8.4.1-1。

- 唤醒

参考表 8.6.1-1，各个模式的唤醒源。

- 总线

AHB 和 APB 总线在 STOP 模式下停止工作。

8.5 Deep Standby 模式相关操作

8.5.1 Deep Standby 模式时钟工作状态

Deep Standby 模式包括了 Deep RTC 模式和 Deep STOP 模式。下表给出了各个振荡器时钟以及 CPU、AHB 和 APB 总线时钟在不同 Deep Standby 模式下的工作状态：

表 8.5.1-1 Deep Standby 模式下时钟工作状态

	Deep RTC 模式	Deep STOP 模式	
HICR	停止	停止	
HSXT	停止		
LICR	根据设置决定		
LSXT	根据设置决定		
CPU clock	停止		
AHB bus clock			
APB bus clock			

注意：

- Deep RTC 模式下 LICR 和 LSXT 至少要开启一个，至于开启哪一个需要根据不同的设置来决定，也可以都开启。

8.5.2 Deep Standby 模式供电情况

下表给出了不同功能模块在 Deep Standby 模式下的供电情况，以及从 Deep Standby 模式切换为复位模式时的初始化状态。

表 8.5.2-2 断电以及复位说明

	供电情况	复位状态
CPU	掉电	初始化
片上闪存	掉电	*1
片上 SRAM	掉电*2	*3
LVD	供电	初始化
RTC	供电	不初始化
LCD	供电	不初始化
GPIO	供电	不初始化
其他外设功能	掉电	初始化

*1: Flash 的数据保持，不会被初始化。

*2: Deep Standby 模式下片上 SRAM 是否掉电保持可配置。

*3: 如果片上 SRAM 断电则数据无法保持，如果不掉电，数据保持而不会被初始化。

8.5.3 Deep RTC 模式功能

- CPU 和片上存储（SRAM 和 Flash）

在 Deep RTC 模式下，CPU、AHB 和 APB 总线时钟会停止，片上存储时钟也会停止，片上闪存数据会保持，片上 SRAM 数据是否保持根据设置决定，片上 SRAM 配置为不断电则数据保持，片上 SRAM 断电则数据丢失。

- 外设

在 Deep RTC 模式下除了 RTC、LCD、和 LVD 工作外，其他功能都会关闭。

- RTC

RTC 在 Deep RTC 模式下不受影响，RTC 会根据在进入 Deep RTC 模式之前的设置正常运行。

- 时钟

每个时钟的状态请参考表 8.5.1-1。

- 唤醒

参考表 8.6.1-1，各个模式的唤醒源。

- 总线

AHB 和 APB 总线在 Deep RTC 模式下停止工作。

8.5.4 Deep STOP 模式功能

- CPU 和片上存储（SRAM 和 Flash）

在 Deep STOP 模式下，CPU、AHB 和 APB 总线时钟会停止，片上存储时钟也会停止，片上闪存数据会保持，片上 SRAM 数据是否保持根据设置决定，片上 SRAM 配置为不断电则数据保持，片上 SRAM 断电则数据丢失。

- 外设

在 Deep STOP 模式下除了 LVD 工作外，其他功能都会关闭。

- RTC

RTC 在 Deep STOP 模式下停止工作。

- 时钟

每个时钟的状态请参考表 8.5.1-1。

- 唤醒

参考表 8.6.1-1，各个模式的唤醒源。

- 总线

AHB 和 APB 总线在 Deep STOP 模式下停止工作。

8.6 低功耗模式设置与唤醒

8.6.1 各个模式唤醒源

表 8.6.1-1 各个模式唤醒源

唤醒方式	Idle 模式	RTC 模式	STOP 模式	Deep RTC 模式	Deep STOP 模式
复位唤醒	<ul style="list-style-type: none"> - RSTB 引脚复位 - LVD 复位 - 软件看门狗复位 - 硬件看门狗复位 - 时钟故障检测复位 - 异常频率检测复位 	<ul style="list-style-type: none"> - RSTB 引脚复位 - LVD 复位 - 硬件看门狗复位 	<ul style="list-style-type: none"> - RSTB 引脚复位 - LVD 复位 	<ul style="list-style-type: none"> - RSTB 引脚复位 - LVD 复位 	<ul style="list-style-type: none"> - RSTB 引脚复位 - LVD 复位
中断唤醒	<ul style="list-style-type: none"> - 各种外设产生的有效中断信号 	<ul style="list-style-type: none"> - NMI 中断 - 外部中断 - RTC 中断 - LVD 中断 - VC 中断 - 硬件看门狗 	<ul style="list-style-type: none"> - NMI 中断 - 外部中断 - LVD 中断 - VC 中断 	<ul style="list-style-type: none"> - 唤醒 IO 引脚中断 - RTC 中断 - LVD 中断 	<ul style="list-style-type: none"> - 唤醒 IO 引脚中断 - LVD 中断

8.6.2 Idle 模式设置与唤醒

- 配置进入 Idle 模式

1) 设置 Cortex-M0+系统控制寄存器的 SLEEPDEEP 比特位为 0

2) 执行 WFI 或者是 WFE 指令

进入何种 Idle 模式与当前系统时钟有关，可以通过查询 CLK_STB 寄存器的 SRC[2:0]位获取当前系统的时钟模式。

- 从 Idle 模式唤醒

复位和有效的中断都可以将系统唤醒，具体唤醒源可以参考表 8.6.1-1。通过复位方式唤醒只能唤醒到 HICR Run 模式，通过中断方式唤醒到何种工作模式与 SRC[2:0]的值相关，如下表所示：

	SRC=3'b000 (HICR)	SRC=3'b001 (HSXT)	SRC=3'b100 (LICR)	SRC=3'b101 (LSXT)
中断唤醒后的时钟模式	HICR Run 模式	HSXT Run 模式	LICR Run 模式	LSXT Run 模式

- 晶振稳定时间

如果是通过复位的方式唤醒，则需要等待 HICR 和 LICR 的时钟稳定时间，如果是通过中断的方式唤醒则不需要等待晶振的稳定时间。

8.6.3 RTC 模式设置与唤醒

- 配置进入 RTC 模式

- 进入 RTC 模式如果需要使用 LSXT, 使能 LSXT 后需要查询 CLK_STB 寄存器的 SUB_RDY 比特位是否为 1
- 配置 CLK_CTL1 寄存器的 RTCE 比特位为 1
- 配置 LPM_CTL 寄存器的 KEY 比特位为 0x1ACC, DSTBEN 比特位为 0, STBM 比特位为 2'b10, SPLV 比特位可以设置 IO 在 RTC 模式下的状态
- 设置 Cortex-M0+控制寄存器的 SLEEPDEEP 位为 1
- 执行 WFI 或 WFE 指令

- 从 RTC 模式唤醒

复位和有效的中断都可以将系统唤醒，具体唤醒源可以参考表 8.6.6-1。通过复位方式唤醒只能唤醒到 HICR Run 模式，通过中断方式唤醒到何种工作模式与 SRC[2:0]的值相关，如下表所示：

	SRC=3'b000 (HICR)	SRC=3'b001 (HSXT)	SRC=3'b100 (LICR)	SRC=3'b101 (LSXT)
中断唤醒后的时钟模式	HICR Run 模式	HSXT Run 模式	LICR Run 模式	LSXT Run 模式

- 晶振稳定时间

如果是通过复位的方式唤醒，则需要等待 HICR 和 LICR 的时钟稳定时间，如果是通过中断的方式唤醒则与系统时钟模式相关，具体的等待响应的稳定时间参考本系列的数据手册。

- LDO 稳定时间

LDO 从 RTC 模式切换到 Run 模式需要等待稳定时间，具体请看数据手册。

8.6.4 STOP 模式设置与唤醒

- 配置进入 STOP 模式

- 配置 CLK_CTL1 寄存器的 RTCE 比特位为 0
- 配置 LPM_CTL 寄存器的 KEY 比特位为 0x1ACC, DSTBEN 比特位为 0, STBM 比特位为 2'b10, SPLV 比特位可以设置 IO 在 STOP 模式下的状态
- 设置 Cortex-M0+控制寄存器的 SLEEPDEEP 位为 1
- 执行 WFI 或 WFE 指令

- 从 STOP 模式唤醒

和有效的中断都可以将系统唤醒，具体唤醒源可以参考表 8.6.1-1。通过复位方式唤醒只能

唤醒到 HICR Run 模式，通过中断方式唤醒到何种工作模式与 SRC[2:0]的值相关，如下表所示：

	SRC=3'b000 (HICR)	SRC=3'b001 (HSXT)	SRC=3'b100 (LICR)	SRC=3'b101 (LSXT)
中断唤醒后的时钟模式	HICR Run 模式	HSXT Run 模式	LICR Run 模式	LSXT Run 模式

- 晶振稳定时间

如果是通过复位的方式唤醒，则需要等待 HICR 和 LICR 的时钟稳定时间，如果是通过中断的方式唤醒则与系统时钟模式相关，具体的等待响应的稳定时间参考本系列的数据手册。

- LDO 稳定时间

LDO 从 STOP 模式切换到 Run 模式需要等待稳定时间，具体请看数据手册。

8.6.5 Deep RTC 模式设置与唤醒

- 配置进入 Deep RTC 模式

- 1) 进入 Deep RTC 模式如果需要使用 LSXT，使能 LSXT 后需要查询 CLK_STB 寄存器的 SUB_RDY 比特位是否为 1
- 2) 配置 CLK_CTL1 寄存器的 RTCE 比特位为 1
- 3) 配置 LPM_CTL 寄存器的 KEY 比特位为 0x1ACC, DSTBEN 比特位为 1, STBM 比特位为 2'b10, SPLV 比特位可以设置 IO 在 Deep RTC 模式下的状态
- 4) 设置 Cortex-M0+控制寄存器的 SLEEPDEEP 位为 1
- 5) 执行 WFI 或 WFE 指令

- 从 Deep RTC 模式唤醒

LVD 复位、RSTB 复位以及 RTC 中断、LVD 中断、唤醒 IO 引脚中断可以将系统唤醒。从 Deep RTC 模式唤醒系统会回到 HICR Run 模式。

- 晶振稳定时间

从 Deep RTC 模式唤醒到 Run 模式，会开启 HICR 和 LICR，所以需要等待 HICR 和 LICR 时钟的稳定时间。

- LDO 稳定时间

LDO 从 Deep RTC 模式切换到 Run 模式需要等待稳定时间，具体请看数据手册。

8.6.6 Deep STOP 模式设置与唤醒

- 配置进入 Deep Stop 模式

- 1) 配置 CLK_CTL1 寄存器的 RTCE 比特位为 0
 - 2) 配置 LPM_CTL 寄存器的 KEY 比特位为 0x1ACC, DSTBEN 比特位为 1, STBM 比特位为 2'b10, SPLV 比特位可以设置 IO 在 Deep STOP 模式下的状态
 - 3) 设置 Cortex-M0+控制寄存器的 SLEEPDEEP 位为 1
 - 4) 执行 WFI 或 WFE 指令
- 从 Deep STOP 模式唤醒

LVD 复位、RSTB 复位以及 LVD 中断、唤醒 IO 引脚中断可以将系统唤醒。从 Deep STOP 模式唤醒系统会回到 HICR Run 模式。
 - 晶振稳定时间

从 Deep STOP 模式唤醒到 Run 模式，会开启 HICR 和 LICR，所以需要等待 HICR 和 LICR 时钟的稳定时间。
 - LDO 稳定时间

LDO 从 Deep STOP 模式切换到 Run 模式需要等待稳定时间，具体请看数据手册。

8.7 寄存器

8.7.1 低功耗模式控制寄存器(LPM_CTL)

低功耗模式控制寄存器，用于控制进入 Standby 和 Deep Standby 模式。只有同时写入正确的 KEY 值 0x1ACC，SPLV、DSTBEN 和 STBM 才会被写入有效的值。

比特位	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
寄存器位	Key							
初值	0	0	0	0	0	0	0	0
属性	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
复位源	PRESET							
比特位	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
寄存器位	Key							
初值	0	0	0	0	0	0	0	0
属性	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
复位源	PRESET							
比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-							
初值	0							
属性	R0							
复位源	PRESET							
寄存器位	-	-	-	SPLV	-	DSTB EN	STBM	
初值	0	0	0	0	0	0	00	
属性	R0	R0	R0	R/W	R0	R/W	R/W	

[bit31:16] KEY: LPM_CTL 寄存器 KEY 值

用于释放 SPLV、DSTBEN 和 STBM 比特位的写权限

- 只有对 KEY 写入 0x1ACC 时，写入 SPLV、DSTBEN 和 STBM 的值才会有效
- 如果写入的 KEY 值不是 0x1ACC，则写入 SPLV、DSTBEN 和 STAM 的值无效
- 当读该寄存器时，KEY 值读取的值为 0x0000

[bit15:5]保留位,读取值永远为 0, 写操作无意义

[bit4] SPLV: Standby 模式下 PIN 设置,

在系统处于 RTC 模式、STOP 模式、Deep RTC 模式和 Deep STOP 模式时用于设置引脚的状态。

比特	描述
0	在 RTC 和 STOP 模式下引脚的状态保留, 在 Deep RTC 和 Deep STOP 模式下切换为 GPIO. [默认值]
1	在 RTC、STOP、Deep RTC 和 Deep STOP 模式下将引脚状态设置为高阻态

[bit3]保留位, 读取值永远为 0, 写操作无意义

[bit2] DSTBEN: Deep Standby 模式选择

此比特位用于选择进入 Standby 模式还是 Deep Standby 模式, 0 表示进入 Standby 模式, 1 表示进入 Deep Standby 模式。

[bit1:0] STBM: Standby 模式选择信号

此信号根据 DSTBEN 和 RTCE (CLK_CTL1) 确定进入 RTC 模式、STOP 模式、Deep RTC 模式还是 Deep STOP 模式, 详细信息见下表。

DSTBEN	STBM		CLK_CTL1:RTCE	Description
	bit1	bit0		
0	0	0	0	Run 模式[默认]
0	1	0	0	STOP 模式
0	0	0	1	禁止设置
0	0	1	0	禁止设置
0	0	1	1	禁止设置
0	1	0	1	RTC 模式
0	1	1	0	禁止设置
0	1	1	1	禁止设置
1	0	0	0	禁止设置
1	0	0	1	禁止设置
1	0	1	0	禁止设置
1	0	1	1	禁止设置
1	1	0	0	Deep STOP 模式
1	1	0	1	Deep RTC 模式
1	1	1	0	禁止设置
1	1	1	1	禁止设置

注意:

- 只有同时将 KEY 值写为 0x1ACC 时, 写入 SPLV、DSTBEN 和 STBM 的值才会有效。如果写入 KEY 的值不是 0x1ACC, 则写入 SPLV、DSTBEN 和 STBM 的值是无效的。

8.7.2 LDO 控制寄存器(LDO_CTL)

比特位	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
寄存器位	Key							
初值	0	0	0	0	0	0	0	0
属性	W/R0							
复位源	PONR							
比特位	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
寄存器位	Key							
初值	0	0	0	0	0	0	0	0
属性	W/R0							
复位源	PONR							
比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	AutoSWEEn
初值	0	0	0	0	0	0	0	1
属性	R0	R/W						
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R/W	R/W
复位源	PONR							

AutoSWEEn: LDO Standby 模式自动切换使能

0: 系统时钟为低频时钟时 LDO 不会自动切换到 Standby 模式

1: 系统时钟为低频时钟时 LDO 自动切换到 Standby 模式

Key: 需要配置 AutoSWEEn 时需要先写入正确的 Key, Key 值为 16h55AA

8.7.3 时钟控制寄存器 1(CLK_CTL1)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	-	-	-	RTCE
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R/W
复位源	PONR							

RTC 模式选择

RTCE: 0: 不选择 RTC 模式,
1: 选择 RTC 模式

注:

- 如果选用 LSXT 作为 RTC 的时钟源, 在写该寄存器之前必须先判断 CRG 中寄存器 CLK_STB.SUB_RDY 是否为 1。

8.7.4 时钟控制寄存器 2(CLK_CTL2)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	-	WDTC KE	LCDC KE	RTCCK E
初值	0	0	0	0	0	1	0	1
属性	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
复位源	PONR							

XXX 时钟使能

XXXCKE: 0: 不使能时钟
1: 使能时钟

8.7.5 Deep Standby 模式 RAM 控制寄存器(DS_RAMP)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	-	-	-	RAMP
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R/W
复位源	PONR							

控制 Deep Standby 模式时 RAM 是否带电

RAMP: 0: Deep Standby 模式下 SRAM 掉电
1: Deep Standby 模式下 SRAM 带电

8.7.6 唤醒复位源状态寄存器(WAKE_RSR)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0						
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	-	-	RELV DR	RERSTB
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R	R
复位源	PONR							

RSTB 引脚复位查询

RERSTB: 0: 未发生 RSTB 引脚复位
1: 发生了 RSTB 引脚复位

LVD 复位查询

RELVDR: 0: 未发生 LVD 复位
1: 发生了 LVD 复位

注:

- 该寄存器读清零。

8.7.7 唤醒中断源状态寄存器(WAKE_ISR)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	REW U3I	REW U2I	REWU1 I	REWU0 I	RELV DI	RERT CI
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R	R	R	R	R	R
复位源	PONR							

RERTCI: RTC 中断唤醒

RELVDI: LVD 中断唤醒

REWU0I-REWU3I: 唤醒引脚(WKUP0~3) 唤醒

注:

- 该寄存器读清零。

8.7.8 唤醒中断源使能寄存器(WAKE_IER)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	REW U3E	REW U2E	REWU1 E	REW U0E	RELVDE E	RERTCE E
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W
复位源	PONR							

RERTCE: RTC 唤醒使能

RELVDE: LVD 唤醒使能

REWU0E-REWU3E: 唤醒引脚(WKUP0~3) 唤醒使能

8.7.9 唤醒引脚中断高低电平设置寄存器(WAKE_LVR)

比特位	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
寄存器位	-	-	-	-	-	-	-	-
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R0	R0	R0	R0
复位源	PONR							
比特位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
寄存器位	-	-	-	-	WUI3LV	WUI2L V	WUI1 LV	WUI0L V
初值	0	0	0	0	0	0	0	0
属性	R0	R0	R0	R0	R/W	R/W	R/W	R/W
复位源	PONR							

WUIXLV：唤醒引脚(WKUP0~3) 唤醒中断高低电平有效设置：

0：低电平有效

1：高电平有效

8.7.10 备份寄存器(BUR01-BUR16)

BUR01-BUR16 为备份寄存器，当系统进入 Deep Standby 模式后数据可以保持。该寄存器只能被 PONR 复位。

比特位	31		24	23		16	15		8	7		0
寄存器位	BUR04			BUR03			BUR02			BUR01		
属性	R/W			R/W			R/W			R/W		
初始值	0x00			0x00			0x00			0x00		

比特位	31		24	23		16	15		8	7		0
寄存器位	BUR08			BUR07			BUR06			BUR05		
属性	R/W			R/W			R/W			R/W		
初始值	0x00			0x00			0x00			0x00		

比特位	31		24	23		16	15		8	7		0
寄存器位	BUR12			BUR11			BUR10			BUR09		
属性	R/W			R/W			R/W			R/W		
初始值	0x00			0x00			0x00			0x00		

比特位	31		24	23		16	15		8	7		0
寄存器位	BUR16			BUR15			BUR14			BUR13		
属性	R/W			R/W			R/W			R/W		
初始值	0x00			0x00			0x00			0x00		

9. 电源结构和电源域

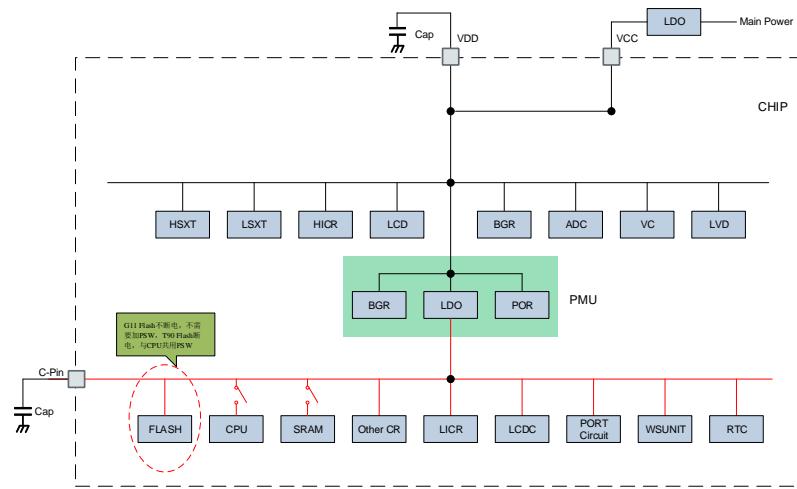
9.1 概述

本章主要讲述电源相关内容，包括电源结构、供电形式、电源域划分以及电源之间的切换。

9.2 电源结构

电源结构如下图所示。

图 9.2-1 电源结构

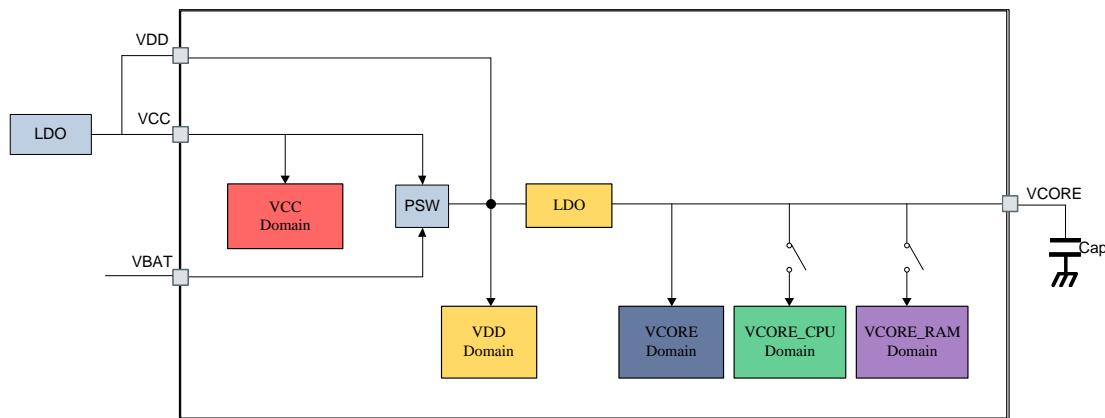


电源门控

当系统进入 Deep Standby 模式时，会切断 VCORE_CPU 电源域的供电，VCORE_RAM 电源域的供电是否被切断根据系统设置来确定。其他模块均属于不断电电源域，即使系统进入 Deep Standby 模式依然不会断电。

9.3 供电形式

图 9.3.1-1 供电形式

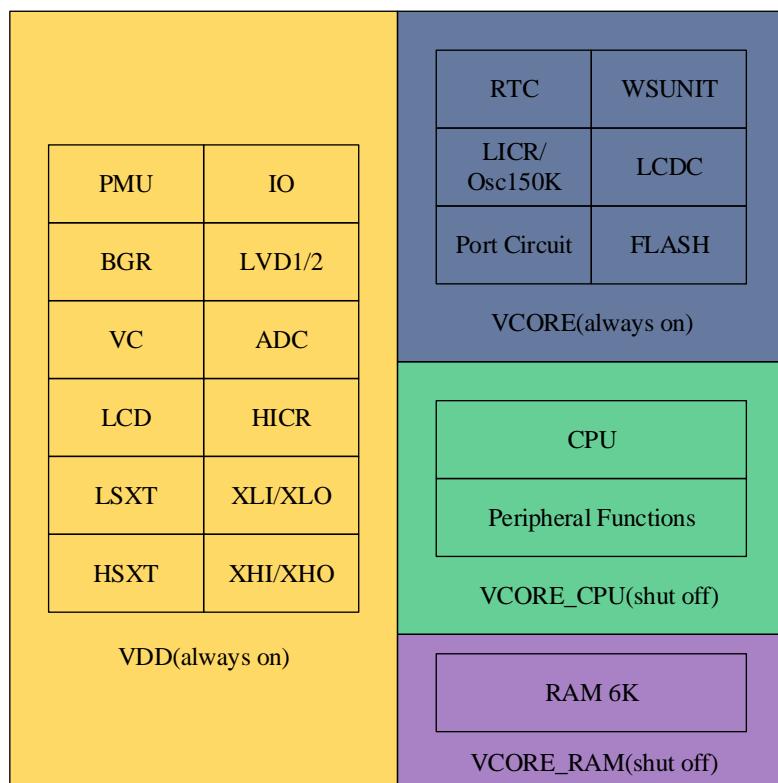


电源供电时，VDD 会和 VCC 一起绑定出去，同样 VCORE 上需要挂一个电容。此时 VCC 电源域和 VDD 电源域其实是同一个电源域。VCC 电源域下的电源切换电路会被关闭以节省功耗。

9.4 电源域划分

电源域划分框图如下图所示，VCC、VDD 和 VCORE 电源域都属于不断电的电源域，VCORE_CPU 和 VCORE_RAM 电源可以关断，用于节省功耗。

图 9.4-1 电源域划分



9.4.1 VDD 电源域

- PMU

PMU 为电源管理模块，包括 POR、LDO 以及 PSW。POR 复位会同时检测 VDD 和 LDO 的输出电压，当两个检测点的 POR 都释放时最终的 POR 才会释放。POR 会复位整个系统。LDO 输出电压给数字逻辑、部分模拟和存储系统供电，电压为 1.5V。

- IO

IO 包括 GPIO、RST IO 以及晶振 IO 等，所有 IO 都不会断电。RST IO 可以复位系统，外部中断可以唤醒 Standby 模式，唤醒 IO 可以唤醒 Deep Standby 模式。

- BGR/LVD/VC/ADC

BGR 给 LVD、VC 和 ADC 提供基准，LVD 可以产生中断或复位，检测阈值可调，VC 为电压比较器可以产生中断，ADC 为 12 位模数转换器，同样也可以产生中断。

- LCD

LCD 在 Deep RTC 模式下可以工作，用于显示相关内容。

- HICR

内部高速时钟，可以产生 4MHz 或 16MHz 的时钟，系统经过分频可以用作系统时钟。

- HSXT/LSXT

外部高速晶振和外部低速晶振，HSXT 支持 4MHz-32MHz，LSXT 接 32.768KHz，可以用作 RTC 等模块工作时钟。

9.4.2 VCORE 电源域

- RTC

实时时钟模块，时钟源可以选择外部 32.768K 晶振，也可以选择内部 32K CR，RTC 中断可以唤醒 RTC 模式和 Deep RTC 模式。在功耗很低的 Deep RTC 模式 RTC 依然可以工作。

- WSUNIT

WSUNIT 为低功耗管理模块，包括进出低功耗模式的控制，以及 LVD 和 PMU 的相关控制。

- LICR、OSC150K

内部低速 32K 时钟和 150K 时钟，32K 时钟可以作为系统时钟、RTC、LCD 等模块的工作时钟，150K 时钟作为 LVD 和 VC 滤波使用。

- LCDC

LCD 显示模块数字控制部分，控制 LCD COM 和 SEG 的波形，以及需要显示的内容。

- PORT Circuit

所有 GPIO 控制逻辑都不会断电, 另外在 Standby 模式下有 16 个 IO 可以产生外部中断将系统唤醒, 在 Deep Standby 模式下有 4 个唤醒 IO 可以产生中断将系统唤醒。

- Flash

由于 Flash 在 Deep Standby 模式下功耗很小所以将 Flash 放在不掉电电源域。

9.4.3 VCORE_CPU 电源域

- CPU

ARM Cortet-M0+ IP, 在 Deep Standby 模式下可以断电来降低功耗。

- Peripheral

APB 和 AHB 总线上的大部分外设, 在 Standby 模式下停止工作, 在 Deep Standby 模式下会断电来节省功耗。

9.4.4 VCORE_RAM 电源域

- SRAM

SRAM 的容量为 6K, 在 Deep Standby 模式下是否断电根据系统设置来决定, 如果断电 SRAM 的数据将会丢失, 如果不断电 SRAM 的数据可以继续保持。

9.5 电源切换流程

9.5.1 PSW 初始状态

VCORE_CPU 和 VCORE_RAM 电源域是可以断电的，需要通过 Power Gating 进行处理，默认 VCORE_CPU 和 VCORE_RAM 电源域是带电的，所以相应的 Power Switch 是导通的。

9.5.2 上电流程

1. 安装主电 VCC，系统上电
2. LDO 输出稳定的电压给数字系统
3. PMU 模块中的 POR 电路工作，产生 POR 复位
4. HICR 和 LICR 上电默认启动
5. 通过 HICR 将 POR 扩展 2ms
6. 系统复位释放，CPU 开始执行指令

9.5.3 切换到 Standby 模式

1. LVD 检测到 VCC 电压低于设置阈值，关闭大功耗外设
2. 系统保存重要数据到 SRAM
3. CPU 执行指令进入 Standby 模式
4. LDO 切换到低功耗模式

9.5.4 切换到 Deep Standby 模式

1. LVD 检测到 VCC 电压低于设置阈值，关闭大功耗外设
2. 系统保存重要数据到 SRAM
3. 根据需要决定 SRAM 是否带电
4. CPU 执行指令进入 Deep Standby 模式
5. CPU 和 RAM 电源域断电（如果 RAM 设置为断电）
6. LDO 切换到低功耗模式

10. 中断控制器

本章介绍中断控制器和外部中断请求。

10.1 概要

Cortex-M0+ 的 CPU 内核在自身内部搭载 NVIC(嵌套向量中断控制器)决定中断请求的优先级并向 CPU 发送请求。此处中断控制器的作用是将多个外设器件的中断信号集中并输入到 NVIC 的中断向量，并可使用中断请求状态寄存器确认中断请求是否发生。

NVIC(嵌套向量中断控制器)的特征:

- 包含 16 个 Cortex-M0+ 异常中断，1 个不可屏蔽中断 (NMI)，32 个可屏蔽的外设中断
- NVIC 与处理器内核接口紧密耦合，NVIC 不仅能处理嵌套的中断信息，还提供低延迟中断处理和有效的迟来中断处理机制
- 4 个可编程的中断优先级
- 操作系统控制寄存器

NVIC 管理包含内核异常在内的所有中断。关于异常和 NVIC，详情参见 ARM 公司《Cortex-M0+ 技术参考手册》的“异常”和“嵌套向量中断控制器”章节。

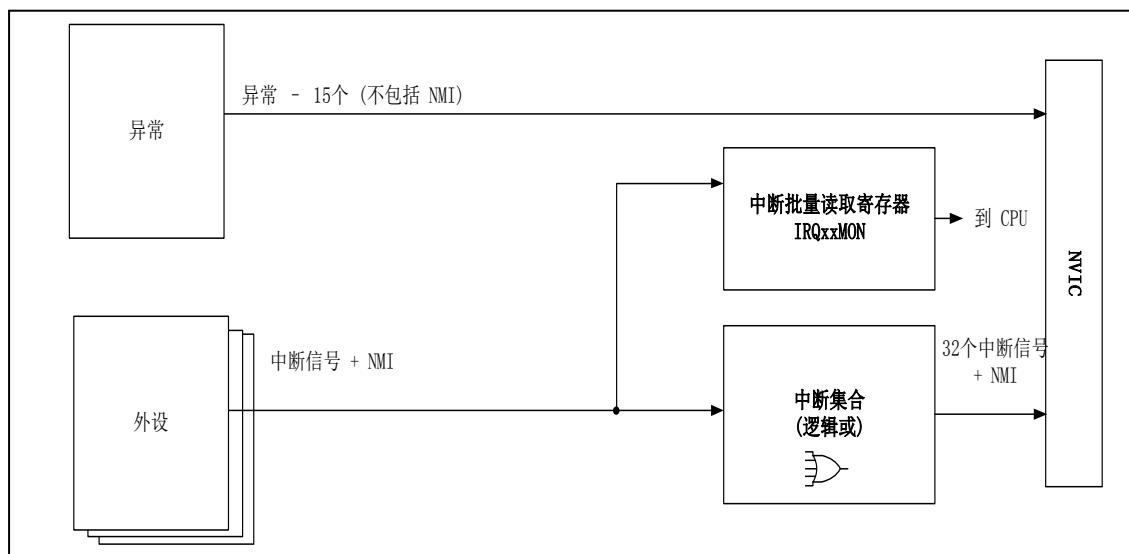
中断向量集中功能

将各外设资源的中断请求信号分为不同中断请求信号组，对应输入到 NVIC 的各中断向量中。例如，NMIX 外部引脚的中断请求信号 和硬件看门狗的中断信号经过逻辑或作为不可屏蔽中断 NMI 输入到 NVIC。通过读取中断请求状态寄存器可确认产生了哪个外设资源的中断请求。但是，不能清除中断标志位，可使用各外设功能的寄存器清除中断标志位。

10.2 构成

本节介绍中断控制器的构成。

图 10.2-1 中断控制器



10.3 异常和中断源向量

本节列出输入到 NVIC 的异常和中断源向量表。

表 10.3-1 异常和中断源向量

向量No.	IRQ No.	例外和中断源	向量 偏移
0	-	堆栈指针初始值	0x00
1	-	复位	0x04
2	-	不可屏蔽中断(NMI)/硬件看门狗定时器	0x08
3	-	硬件错误	0x0C
4	-	保留	0x10
5	-	保留	0x14
6	-	保留	0x18
7-10	-	保留	0x1C - 0x2B
11	-	SVCALL(管理程序请求)	0x2C
12	-	保留	0x30
13	-	保留	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	频率异常中断请求(FCS)/主时钟失效中断请求 (MCSV) /副时钟失效中断请求 (SCSV)	0x40
17	1	软件看门狗定时器	0x44
18	2	低电压检测(LVD)	0x48
19	3	保留	0x4C
20	4	外部引脚中断请求 ch.0 ~ ch.7	0x50
21	5	外部引脚中断请求 ch.8 ~ ch.15	0x54
22	6	双定时器中断请求	0x58
23	7	多功能串口 ch.0 的接收中断请求	0x5C
24	8	多功能串口 ch.0 的发送中断请求和状态中断请求	0x60
25	9	多功能串口 ch.1 的接收中断请求	0x64
26	10	多功能串口 ch.1 的发送中断请求和状态中断请求	0x68
27	11	多功能串口 ch.2 的接收中断请求	0x6C
28	12	多功能串口 ch.2 的发送中断请求和状态中断请求	0x70
29	13	保留	0x74
30	14	保留	0x78
31	15	多功能串口 ch.4 的接收中断请求	0x7C
32	16	多功能串口 ch.4 的发送中断请求, 状态中断请求	0x80
33	17	多功能串口 ch.5 的接收中断请求	0x84

向量No.	IRQ No.	例外和中断源	向量 偏移
34	18	多功能串口 ch.5 的发送中断请求和状态中断请求	0x88
35	19	多功能串口 ch.6 的接收中断请求	0x8C
36	20	多功能串口 ch.6 的发送中断请求和状态中断请求	0x90
37	21	保留	0x94
38	22	保留	0x98
39	23	保留	0x9C
40	24	外部主时钟 /外部副时钟/RTC/RAM校验中断请求	0xA0
41	25	A/D 转换器 unit0中断请求	0xA4
42	26	电压比较器中断请求	0xA8
43	27	SCI7816中断请求	0xAC
44	28	保留	0xB0
45	29	保留	0xB4
46	30	保留	0xB8
47	31	复合定时器 ch.0 ~ ch.7/flash 中断请求	0xBC

向量 No.4 ~ 15 的异常优先级可使用 NVIC 内置的系统处理器优先级寄存器(System Handlers Priority Register)(地址： 0xE000ED18、 0xE000ED1C、 0xE000ED20)进行配置。向量 No.16 以后的外部中断优先级可使用 NVIC 内置 的中断优先级寄存器(IRQ Priority Register)(地址： 0xE000E400 ~ 0xE000E42C)进行配置。

使用中断请求状态寄存器确认向量 No.2, No.16 ~ No.47 的中断源。关于其它异常及中断，参见《Cortex-M0+ 技术参考手册》的“NVIC”。

另外，对于向量 No.2, No.16 ~ No.47 的中断，批量读取的中断源可能是对各个外设模块内的多个中断源进行了逻辑或后的一个信号。详情参见各外设资源的中断。

10.4 寄存器

本节说明中断请求状态寄存器。

表 10.4-1 中断请求状态寄存器

寄存器缩写	寄存器名称	参考章节
IRQ_ST02	IRQ_ST02 中断请求状态寄存器	10.4.1
IRQ00PEND	IRQ00 中断请求状态寄存器	10.4.2
IRQ01PEND	IRQ01 中断请求状态寄存器	10.4.3
IRQ02PEND	IRQ02 中断请求状态寄存器	10.4.4
IRQ04PEND	IRQ04 中断请求状态寄存器	10.4.5
IRQ05PEND	IRQ05 中断请求状态寄存器	10.4.6
IRQ06PEND	IRQ06 中断请求状态寄存器	10.4.7
IRQ07PEND	IRQ07 中断请求状态寄存器	10.4.8
IRQ08PEND	IRQ08 中断请求状态寄存器	10.4.9
IRQ09PEND	IRQ09 中断请求状态寄存器	10.4.8
IRQ10PEND	IRQ10 中断请求状态寄存器	10.4.9
IRQ11PEND	IRQ11 中断请求状态寄存器	10.4.8
IRQ12PEND	IRQ12 中断请求状态寄存器	10.4.9
IRQ15PEND	IRQ15 中断请求状态寄存器	10.4.8
IRQ16PEND	IRQ16 中断请求状态寄存器	10.4.9
IRQ17PEND	IRQ17 中断请求状态寄存器	10.4.8
IRQ18PEND	IRQ18 中断请求状态寄存器	10.4.9
IRQ19PEND	IRQ19 中断请求状态寄存器	10.4.8
IRQ20PEND	IRQ20 中断请求状态寄存器	10.4.9
IRQ24PEND	IRQ24 中断请求状态寄存器	10.4.10
IRQ25PEND	IRQ25 中断请求状态寄存器	10.4.11
IRQ26PEND	IRQ26 中断请求状态寄存器	10.4.12
IRQ27PEND	IRQ27 中断请求状态寄存器	10.4.13
IRQ31PEND	IRQ31 中断请求状态寄存器	10.4.14

关于 NVIC 的寄存器，详情参见《 Cortex-M0 技术参考手册》的“嵌套向量中断控制器”章节。

10.4.1 IRQ_ST02 中断请求状态寄存器

IRQ_ST02 中断请求状态寄存器可读取配置到中断向量 No.2 的中断请求。

IRQ_ST02 显示硬件看门狗定时器和外部 NMIX 引脚中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															HWW DG
属性	R															NMI
初始值	0000000000000000															0

寄存器功能

[bit31: 2] Reserved: 保留位，保留位的读取值为 "0".

[bit1] HWWDG:

位	描述
0	无硬件看门狗定时器中断请求
1	有硬件看门狗定时器中断请求

[bit0] NMI:

位	描述
0	无外部NMIX引脚中断请求
1	有外部NMIX引脚中断请求

10.4.2 IRQ00PEND 中断请求状态寄存器

IRQ00PEND 中断请求状态寄存器可读取配置到中断向量 No.16 的中断请求。

IRQ00PEND 显示以下中断请求的状态：CSV 检测的频率异常、主时钟失效、副时钟失效。

寄存器配置

位	31																	8
字段名	Reserved																	
属性	R																	
初始值	0x000000																	

位	7	6	5	4	3	2	1	0	
字段名	Reserved				SCSV	MCSV	AFDI		
属性	R				R	R	R		
初始值	00000				0	0	0		

寄存器功能

[bit31: 3] Reserved: 保留位，保留位的读取值为 "0"。

[bit2] SCSV:

bit	描述
0	无副时钟失效中断请求
1	有副时钟失效中断请求

[bit1] MCSV:

bit	描述
0	无主时钟失效中断请求
1	有主时钟失效中断请求

[bit0] AFDI:

bit	描述
0	无CSV检测的频率异常中断请求
1	有CSV检测的频率异常中断请求

10.4.3 IRQ01PEND 中断请求状态寄存器

IRQ01PEND 中断请求状态寄存器可读取配置到中断向量 No.17 的中断请求。

IRQ01PEND 显示软件看门狗定时器中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved														SWWDG	
属性	R														R	
初始值	0000000000000000															

寄存器功能

[bit31: 1] Reserved: Reserved bits

保留位的读取值为"0".

[bit0] SWWDG:

Bit	描述
0	无软件看门狗定时器中断请求
1	有软件看门狗定时器中断请求

10.4.4 IRQ02PEND 中断请求状态寄存器

IRQ02PEND 中断请求状态寄存器可读取配置到中断向量 No.18 的中断请求。

IRQ02PEND 显示低电压检测（LVD）中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	22	21	20	19	18	17	16
字段名	Reserved														
属性	R														
初始值	0x0000														

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															LVD
属性	R															R
初始值	0000000000000000															0

寄存器功能

[bit31: 1] Reserved: 保留位

保留位的读取值为"0".

[bit0] LVD:

Bit	描述
0	无低电压检测（LVD）中断请求
1	有低电压检测（LVD）中断请求

10.4.5 IRQ04PEND 中断请求状态寄存器

IRQ04PEND 中断请求状态寄存器可读取配置到中断向量 No.20 的中断请求。

IRQ04PEND 显示外部中断 ch.0 到 ch.7 中断请求的状态。

寄存器配置

位	31														8
字段名	Reserved														
属性	R														
初始值	0x0000														
位	7	6	5	4	3	2	1	0							
字段名	EXTI7	EXTI6	EXTI5	EXTI4	EXTI3	EXTI2	EXTI1	EXTI0							
属性	R	R	R	R	R	R	R	R							
初始值	0	0	0	0	0	0	0	0							

寄存器功能

[bit31: 8] Reserved: 保留位

保留位的读取值为"0".

[bit7: 0] EXTx

Bit No.	Bit	描述
7	0	无外部中断ch.7的中断请求
	1	有外部中断ch.7的中断请求
6	0	无外部中断ch.6的中断请求
	1	有外部中断ch.6的中断请求
5	0	无外部中断ch.5的中断请求
	1	有外部中断ch.5的中断请求
4	0	无外部中断ch.4的中断请求
	1	有外部中断ch.4的中断请求
3	0	无外部中断ch.3的中断请求
	1	有外部中断ch.3的中断请求
2	0	无外部中断ch.2的中断请求
	1	有外部中断ch.2的中断请求
1	0	无外部中断ch.1的中断请求
	1	有外部中断ch.1的中断请求
0	0	无外部中断ch.0的中断请求
	1	有外部中断ch.0的中断请求

10.4.6 IRQ05PEND 中断请求状态寄存器

IRQ05PEND 中断请求状态寄存器可读取配置到中断向量 No.21 的中断请求。

IRQ05PEND 显示外部中断 ch.8 到 ch.15 中断请求的状态。

寄存器配置

位	31							8
字段名	Reserved							
属性	R							
初始值	0x0000							
位	7	6	5	4	3	2	1	0
字段名	EXTI15	EXTI14	EXTI13	EXTI12	EXTI11	EXTI10	EXTI9	EXTI8
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31: 8] Reserved: 保留位

保留位的读取值为"0".

[bit7: 0] EXTIx

Bit No.	Bit	描述
7	0	无外部中断ch.15的中断请求
	1	有外部中断ch.15的中断请求
6	0	无外部中断ch.14的中断请求
	1	有外部中断ch.14的中断请求
5	0	无外部中断ch.13的中断请求
	1	有外部中断ch.13的中断请求
4	0	无外部中断ch.12的中断请求
	1	有外部中断ch.12的中断请求
3	0	无外部中断ch.11的中断请求
	1	有外部中断ch.11的中断请求
2	0	无外部中断ch.10的中断请求
	1	有外部中断ch.10的中断请求
1	0	无外部中断ch.9的中断请求
	1	有外部中断ch.9的中断请求
0	0	无外部中断ch.8的中断请求
	1	有外部中断ch.8的中断请求

10.4.7 IRQ06PEND 中断请求状态寄存器

IRQ06PEND 中断请求状态寄存器可读取配置到中断向量 No.22 的中断请求。

IRQ06PEND 显示双定时器中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved												DTI			
属性	R												R			
初始值	0000000000000000												00			

寄存器功能

[bit31: 2] Reserved: 保留位

保留位的读取值为"0".

[bit1: 0] DTI:

Bit No.	Bit	描述
1	0	无双定时器DTI2的中断请求
	1	有双定时器DTI2的中断请求
0	0	无双定时器DTI1的中断请求
	1	有双定时器DTI1的中断请求

10.4.8 IRQ07/09/11/15/17/19PEND 中断请求状态寄存器

IRQ07PEND 中断请求状态寄存器可读取配置到中断向量 No.23 的中断请求。

IRQ09PEND 中断请求状态寄存器可读取配置到中断向量 No.25 的中断请求。

IRQ11PEND 中断请求状态寄存器可读取配置到中断向量 No.27 的中断请求。

IRQ15PEND 中断请求状态寄存器可读取配置到中断向量 No.31 的中断请求。

IRQ17PEND 中断请求状态寄存器可读取配置到中断向量 No.33 的中断请求。

IRQ19PEND 中断请求状态寄存器可读取配置到中断向量 No.35 的中断请求。

IRQ07PEND 显示多功能串口 ch.0 接收中断请求的状态。

IRQ09PEND 显示多功能串口 ch.1 接收中断请求的状态。

IRQ11PEND 显示多功能串口 ch.2 接收中断请求的状态。

IRQ15PEND 显示多功能串口 ch.4 接收中断请求的状态。

IRQ17PEND 显示多功能串口 ch.5 接收中断请求的状态。

IRQ19PEND 显示多功能串口 ch.6 接收中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															
属性	R															
初始值	0000000000000000															

寄存器功能

[bit31: 1] Reserved: Reserved bits, reserved bit reads "0".

[bit0] MSCI:

bit No.	bit	描述
0	0	无多功能串口相应通道接收中断请求
	1	有多功能串口相应通道接收中断请求

10.4.9 IRQ08/10/12/16/18/20PEND 中断请求状态寄存器

IRQ08PEND 中断请求状态寄存器可读取配置到中断向量 No.24 的中断请求。

IRQ10PEND 中断请求状态寄存器可读取配置到中断向量 No.26 的中断请求。

IRQ12PEND 中断请求状态寄存器可读取配置到中断向量 No.28 的中断请求。

IRQ14PEND 中断请求状态寄存器可读取配置到中断向量 No.30 的中断请求。

IRQ16PEND 中断请求状态寄存器可读取配置到中断向量 No.32 的中断请求。

IRQ18PEND 中断请求状态寄存器可读取配置到中断向量 No.34 的中断请求。

IRQ20PEND 中断请求状态寄存器可读取配置到中断向量 No.36 的中断请求。

IRQ08PEND 显示多功能串口 ch.0 的发送中断请求和状态中断请求的状态。

IRQ10PEND 显示多功能串口 ch.1 的发送中断请求和状态中断请求的状态。

IRQ12PEND 显示多功能串口 ch.2 的发送中断请求和状态中断请求的状态。

IRQ16PEND 显示多功能串口 ch.4 的发送中断请求和状态中断请求的状态。

IRQ18PEND 显示多功能串口 ch.5 的发送中断请求和状态中断请求的状态。

IRQ20PEND 显示多功能串口 ch.6 的发送中断请求和状态中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															MSCI
属性	R															R
初始值	0000000000000000															00

寄存器功能

[bit31: 2] Reserved: 保留位，保留位的读取值为 "0".

[bit1: 0] MSCI:

bit No.	bit	描述
1	0	无多功能串口相应通道的状态中断请求
	1	有多功能串口相应通道的状态中断请求
0	0	无多功能串口相应通道的发送中断请求
	1	有多功能串口相应通道的发送中断请求

10.4.10 IRQ24PEND 中断请求状态寄存器

IRQ24PEND 中断请求状态寄存器可读取配置到中断向量 No.40 的中断请求。

IRQ24PEND 显示以下中断请求的状态：RAM 奇偶校验、RTC、外部副时钟、外部主时钟。

寄存器配置

位	31																8
字段名	Reserved																
属性	R																
初始值	0x0000000																

位	7	6	5	4	3	2	1	0
字段名	RAMI	Reserved	RTCI	Reserved		SOSCI	MOSCI	
属性	R	R	R	R		R	R	
初始值	0	0	0	000		0	0	

寄存器功能

[bit31: 8] Reserved: 保留位, 保留位的读取值为 "0".

[bit7] RAMI:

bit	描述
0	无RAM奇偶校验中断请求
1	有RAM奇偶校验中断请求

[bit6] Reserved: 保留位, 保留位的读取值为 "0".

[bit5] RTCI:

bit	描述
0	无RTC中断请求
1	有RTC中断请求

[bit4: 2] Reserved: 保留位, 保留位的读取值为 "0".

[bit1] SOSCI:

bit	描述
0	无副时钟等待稳定完成的中断请求
1	有副时钟等待稳定完成的中断请求

[bit0] MOSCI:

bit	描述
0	无主时钟等待稳定完成的中断请求
1	有主时钟等待稳定完成的中断请求

10.4.11 IRQ25PEND 中断请求状态寄存器

IRQ25PEND 中断请求状态寄存器可读取配置到中断向量 No.41 的中断请求。

IRQ25PEND 显示 A/D 转换器的中断请求状态。

寄存器配置

位	31							8
字段名	Reserved							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段名	Reserved			ADCCMPI	Reserved			ADCCNVI
属性	R			R	R			R
初始值	000			0	000			0

寄存器功能

[bit31: 5] Reserved: 保留位, 保留位的读取值为 "0".

[bit4] ADCCMPI:

bit	描述
0	无A/D转换器比较结果的中断请求
1	有A/D转换器比较结果的中断请求

[bit3: 1] Reserved: 保留位, 保留位的读取值为 "0".

[bit0] ADCCNVI:

bit	描述
0	无A/D转换器转换中断的中断请求
1	有A/D转换器转换中断的额中断请求

10.4.12 IRQ26PEND 中断请求状态寄存器

IRQ26 中断请求状态寄存器（IRQ26PEND）可批量读取配置到中断向量 No.42 的中断请求。

IRQ26PEND 显示电压比较器的中断请求状态。

寄存器配置

位	31																																		8
字段名	Reserved																																		
属性	R																																		
初始值	0x000000																																		

位	7	6	5	4	3	2	1	0
字段名	Reserved		VCINT	Reserved				
属性	R		R	R				
初始值	00		0	00000				

寄存器功能

[bit31: 6] Reserved: 保留位, 保留位的读取值为 "0".

[bit5]VCINT:

bit	描述
0	无电压比较器的中断请求
1	有电压比较器的中断请求

[bit4: 0] Reserved: 保留位, 保留位的读取值为 "0".

10.4.13 IRQ27PEND 中断请求状态寄存器

IRQ27PEND 中断请求状态寄存器可读取配置到中断向量 No.43 的中断请求。

IRQ27PEND 显示 ISO7816 的中断请求状态。

寄存器配置

位	31								8
字段名	Reserved								
属性	R								
初始值	0x000000								

位	7	6	5	4	3	2	1	0		
字段名	Reserv ed	ICCI	Reserved							
属性	R	R	R							
初始值	0	0	000000							

寄存器功能

[bit31: 7] Reserved: 保留位, 保留位的读取值为 "0".

[bit6]ICCI:

bit	描述
0	无SCI7816的中断请求
1	有SCI7816的中断请求

[bit5: 0] Reserved: 保留位, 保留位的读取值为 "0".

10.4.14 IRQ31PEND 中断请求状态寄存器

IRQ31PEND 中断请求状态寄存器可读取配置到中断向量 No.47 的中断请求。

IRQ31PEND 显示 flash 和复合定时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名		Reserved		FLASHI		Reserved										
属性		R		R		R										
初始值		0000		0		000000000000										

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	CTIIRQ															
属性	R															
初始值	0x0000															

寄存器功能

[bit31: 28] Reserved: 保留位,保留位的读取值为 "0".

[bit27] FLASHI:

bit	描述
0	无flash中断请求
1	有flash中断请求

[bit26: 16] Reserved: 保留位,保留位的读取值 "0".

[bit15: 0] CTIIRQ

bit No.	bit	描述
15	0	无复合定时器 ch.7的IRQ1中断请求
	1	有复合定时器 ch.7的IRQ1中断请求
14	0	无复合定时器 ch.7的IRQ0中断请求
	1	有复合定时器 ch.7的IRQ0中断请求
13	0	无复合定时器 ch.6的IRQ1中断请求
	1	有复合定时器 ch.6的IRQ1中断请求
12	0	无复合定时器 ch.6的IRQ0中断请求
	1	有复合定时器 ch.6的IRQ0中断请求
11	0	无复合定时器 ch.5的IRQ1中断请求
	1	有复合定时器 ch.5的IRQ1中断请求
10	0	无复合定时器 ch.5的IRQ0中断请求
	1	有复合定时器 ch.5的IRQ0中断请求
9	0	无复合定时器 ch.4的IRQ1中断请求

bit No.	bit	描述
8	1	有复合定时器 ch.4的IRQ1中断请求
	0	无复合定时器 ch.4的IRQ0中断请求
7	1	有复合定时器 ch.4的IRQ0中断请求
	0	无复合定时器 ch.3的IRQ1中断请求
6	1	有复合定时器 ch.3的IRQ1中断请求
	0	无复合定时器 ch.3的IRQ0中断请求
5	1	有复合定时器 ch.2的IRQ1中断请求
	0	无复合定时器 ch.2的IRQ1中断请求
4	1	有复合定时器 ch.2的IRQ0中断请求
	0	无复合定时器 ch.2的IRQ0中断请求
3	1	有复合定时器 ch.1的IRQ1中断请求
	0	无复合定时器 ch.1的IRQ1中断请求
2	1	有复合定时器 ch.1的IRQ0中断请求
	0	无复合定时器 ch.1的IRQ0中断请求
1	1	有复合定时器 ch.0的IRQ1中断请求
	0	无复合定时器 ch.0的IRQ1中断请求
0	1	有复合定时器 ch.0的IRQ0中断请求
	0	无复合定时器 ch.0的IRQ0中断请求

10.5 使用注意事项

使用中断控制器时应该注意以下几点：

- 各外设资源的中断请求信号都是电平信号。退出中断处理时，须清零中断请求。
- NMIX 引脚与通用口共享引脚。复位后，初始值用作通用口，屏蔽 NMI 输入。要使用 NMI，端口设定时须使能 NMI。详情参见“外部中断及 NMI 控制部分”一章。
- 关于各外设资源的事件检测寄存器和中断使能寄存器的具体对应关系，参见各模块的章节。

11. 外部中断和 NMI 控制部分

本章介绍外部中断和 NMI 控制部分的功能和操作。

11.1 概要

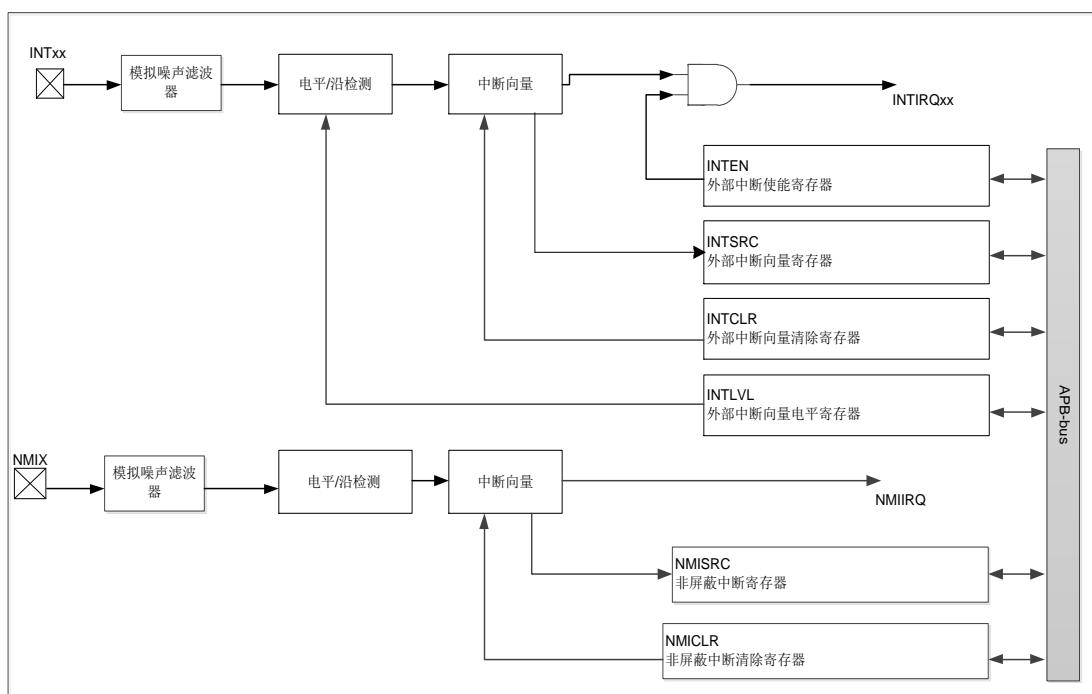
外部中断和 NMI 控制部分具有以下特征。

- 搭载最多 16 个外部中断输入引脚和 1 个 NMIX 输入引脚。
- 外部中断检测可选择“H”电平、L“电平”、上升沿和下降沿方式。
- 可使用外部中断输入或 NMI 输入唤醒待机模式。

11.2 框图

外部中断和 NMI 控制部分的框图如下。

图 11.2-1 外部中断和 NMI 控制部分框图



11.3 操作

本章介绍操作说明及设置步骤示例。

11.3.1 外部中断控制部分的操作

外部中断控制部分的操作概要

外部中断控制部分按照以下顺序将外部中断请求输出到中断控制器：

1. 配置外部中断电平寄存器 (INTLVL)，设定检测 INTxx 引脚输入信号的电平/沿。检测的电平/沿有以下 4 种选择：“H”电平，“L”电平，上升沿，下降沿。
2. 检测到的中断输入保留在中断向量内，可以通过外部中断源寄存器 (INTSRC) 读取，也可以通过外部中断源清除寄存器 (INTCLR) 清除。
3. 当外部中断通过外部中断使能寄存器 (INTEN) 使能时，该外部中断请求 (INTIRQxx) 则会被输出到中断控制器被 CPU 响应。

设置步骤

请按照以下步骤配置外部中断：

1. 使用外部中断使能寄存器 (INTEN) 禁止外部中断。
2. 使用外部中断源电平寄存器 (INTLVL) 设定检测条件 (有效沿或者电平)。
3. 使用外部中断源电平寄存器 (INTLVL)。
4. 使用外部中断源清除寄存器 (INTCLR) 清除外部中断源。
5. 使用外部中断使能寄存器 (INTEN) 使能外部中断。

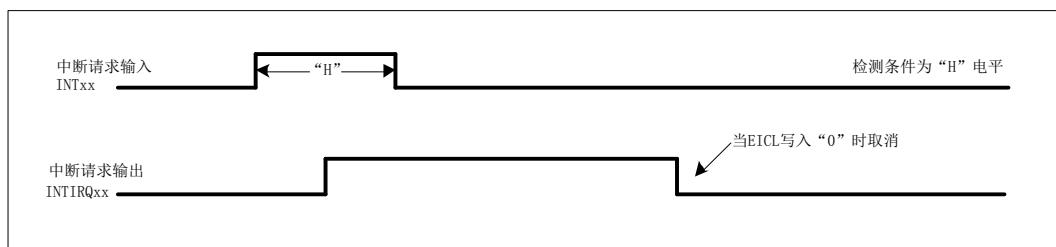
图11.3.1-1 外部中断设置步骤



外部中断请求的取消

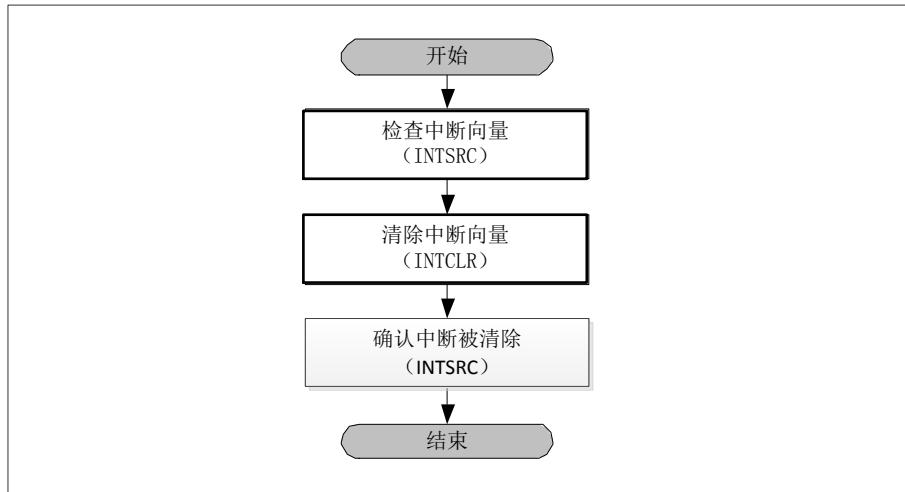
当检测到外部中断时，中断向量被保存在外部中断源寄存器（INTSRC）中。图 10.3.1-2 显示了高电平检测条件的波形。当外部中断检测条件设置为“H”电平或者“L”电平时，即使外部中断输入（INTxx）无效时，外部中断源仍保留在外部中断源寄存器（INTSRC）中。因此，外部中断请求仍会被输出到中断控制器。CPU 在接收到中断以后，应该使用外部中断源清除寄存器（INTCLR）清除外部中断源寄存器（INTSRC）。

当设定电平为检测条件时，在 INTxx 引脚输入被响应期间，CPU 即使通过外部中断清除寄存器（INTCLR）对部中断向量寄存器（INTSRC）清零，该位仍然会置“1”。



请按照以下步骤取消外部中断请求：

1. 读取并检查中断向量寄存器（INTSRC）。
2. 将外部中断源清除寄存器（INTCLR）相应位写“0”。
3. 读取外部中断源寄存器（INTSRC），并确定中断向量被清除。



11.3.2 NMI 控制部分的操作

NMI 控制部分概要

如果检测到 NMI 输入引脚(NMIX)有效电平/沿, NMI 控制部分则输出 NMI 中断请求到 CPU。

以下电平或者沿可以被检测(条件不能更改):

- Run 模式: 下降沿
- Sleep 模式: 下降沿
- RTC 模式: “L”电平
- Stop 模式: “L”电平
- Deep standby 模式: NMI 中断请求在该模式下不可用

注意:

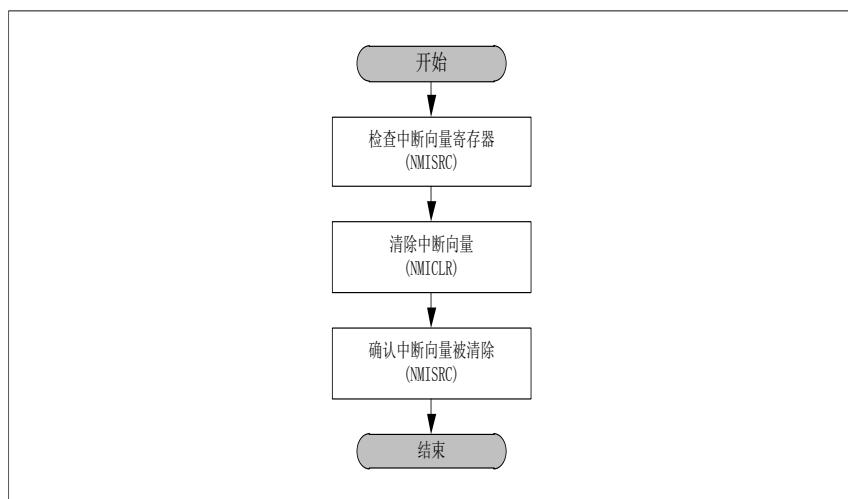
- NMI 中断请求不能用于唤醒 Deep standby 模式。但是, NMIX 输入引脚与 WKUP 输入引脚共享同一个引脚。因此, 可以通过 WKUP 输入引脚唤醒。详情请查看“低功耗模式”章的“Deep standby 模式下的操作”一节。

NMI 中断请求的取消

NMI 中断请求的取消, 与外部中断请求一样, 需要清除中断向量寄存器。请按照以下步骤清除 NMI 中断请求:

1. 读取并查看 NMI 向量寄存器 (NMISRC)。
2. 将 NMI 向量清除寄存器 (NMICLR) 相应位写 “0”。
3. 读取 NMI 向量寄存器 (NMISRC) 并确认该中断向量已清除。

图 11.3.1-2 取消 NMI 中断请求



11.3.3 唤醒 Stop 和 RTC 模式

本节介绍 Stop 模式和 RTC 模式的唤醒

概要

外部中断和 NMI 中断可用于唤醒 Stop 和 RTC 模式。

在这些模式中，INTxx 和 NMIX 引脚的输入信号为异步输入，芯片可以从这些模式回到 RUN 模式。

进入 Stop/RTC 模式之前的配置

为了使用外部中断请求唤醒，在进入 STOP/RTC 模式之前，需要配置用于唤醒的引脚及有效检测电平。

- 用于唤醒的引脚：中断请求输出使能（INTEN = 1）
- 不用于唤醒的引脚：中断请求输出禁止（INTEN = 0）

使用 NMI 不需要设置寄存器，只检测“L”电平。

唤醒 Stop/RTC 模式

对于外部中断请求，如果在用于唤醒的引脚上检测到事先设置的有效电平，芯片则退出 Stop/RTC 模式。

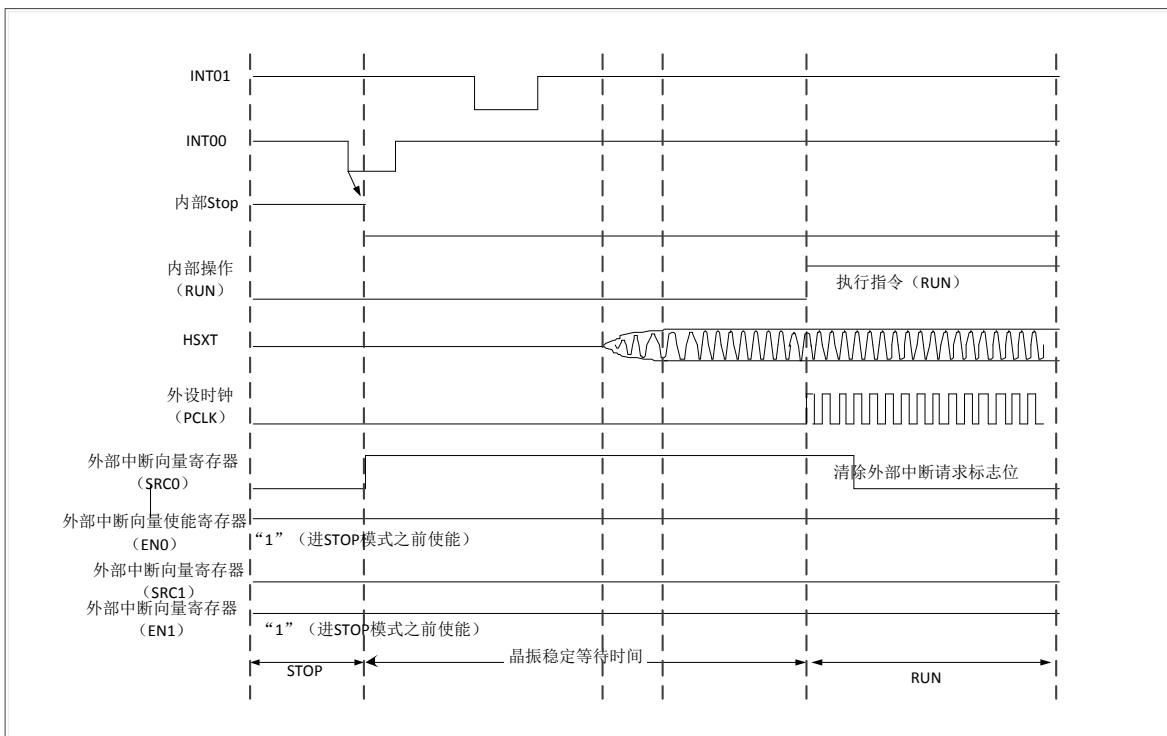
对于 NMI，如果在 Stop/RTC 模式下检测到“L”电平，芯片则退出 Stop/RTC 模式。

从 Stop/RTC 模式唤醒时的注意事项

从退出 Stop/RTC 模式到晶振稳定等待时间结束，这段时间内不可识别其他外部中断请求。（图 11.3.3--5 中，INT01 为不可识别的外部中断请求。）

因此，退出 Stop/RTC 模式后进行外部中断输入时，要等晶振稳定等待时间结束后再输入外部中断信号。

图 11.3.3-5 退出 Stop/RTC 模式



11.4 寄存器

本节介绍外部中断和 NMI 控制部分的寄存器。

外部中断和 NMI 控制部分寄存器列表

表 11.4-1 外部中断和 NMI 控制部分寄存器列表

缩写	寄存器名	参考
INTEN	外部中断使能寄存器	11.4.1
INTSRC	外部中断源寄存器	11.4.2
INTCLR	外部中断源清除寄存器	11.4.3
INTLVL	外部中断源电平寄存器	11.4.4
NMISRC	非屏蔽中断寄存器	11.4.5
NMICLR	非屏蔽中断清除寄存器	11.4.6

11.4.1 外部中断使能寄存器 (INTEN)

外部中断使能寄存器 (INTEN) 用于控制外部中断请求的屏蔽输出。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	EN[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:16]Reserved: 保留位

保留位的读取值为“0”。

[bit15:0] EN15 ~ EN0: 外部中断使能位

EN15~EN0 分别对应引脚 INT15 ~ INT00。

不能设定产品规格中没有定义的引脚的对应位。

bit	描述
0	禁止对应位引脚INTxx的外部中断请求
1	使能对应位引脚INTxx的外部中断请求

寄存器中写“1”对应的中断请求会被输出到中断控制器。写“0”的对应引脚虽然保留中断源，但不输出中断请求到中断控制器。

11.4.2 外部中断源寄存器 (INTSRC)

外部中断源寄存器(INTSRC)显示检测到外部中断请求。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	SRC[15:0]															
属性	R															
初始值	0xXXXX															

寄存器功能

[bit31:0]Reserved: 保留位，保留位的读取值为“0”。

[bit15:0] SRC15 ~ SRC0: 外部中断请求检测位

SRC15 ~ SRC0 分别对应引脚 INT15 ~ INT00。

产品规格中不存在的引脚的对应位不确定。.

bit	描述
0	未检测到对应引脚INTxx的外部中断请求
1	检测到对应引脚INTxx的外部中断请求
写	无效

注意：

- 使用 INTLVL 设定电平检测时，在有效电平从 INTxx 引脚输入期间，即使使用外部中断源清除寄存器(INTCLR)清零对应位，外部中断源寄存器 (INTSRC) 的对应位仍会再次置“1”。
- 初始化时，外部中断源寄存器 (INTSRC) 的对应位有可能置“1”，因此，将 GPIO 设定为外部中断引脚后，请清零外部中断源寄存器 (INTSRC)。

11.4.3 外部中断源清除寄存器 (INTCLR)

外部中断源清除寄存器 (INTCLR) 用于清除保留的中断源。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	Reserved															
属性	R/W															
初始值	0xFFFF															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	CLR[15:0]															
属性	R/W															
初始值	0xFFFF															

寄存器功能

[bit31:16]Reserved: 保留位

保留位的读取值为“1”。

[bit15:0] CLR[15:0]: 外部中断源清除位

CLR[15:0] 对应引脚 INT15 ~ INT00。

不能将产品规格中不存在的引脚对应位写“0”。

bit	Function															
写“0”	清除外部中断源															
写“1”	无效															
读	始终读“1”															

11.4.4 外部中断电平寄存器 (INTLVL)

外部中断电平寄存器 (INTLVL) 用于选择电平或者沿作为外部中断请求检测信号。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名 LVL	H15	L15	H14	L14	H13	L13	H12	L12	H11	L11	H10	L10	H9	L9	H8	L8
属性	R/W															
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名 LVL	H7	L7	H6	L6	H5	L5	H4	L4	H3	L3	H2	L2	H1	L1	H0	L0
属性	R/W															
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] LVLL15 ~ LVLL0 或 LVLH15 ~ LVLH0: 外部中断请求检测电平选择位

LVLL15 ~ LVLL0 和 LVLH15 ~ LVLH0 每两位组合 (LVLL 和 LVLH) 对应 INT15~INT00。

不能设定产品规格书中不存在的引脚的对应位。

一旦检测出该位选择的沿或者电平，则认为是一个外部中断请求。

LVLHx	LVLLx	描述
0	0	检测" L "电平
0	1	检测" H "电平
1	0	检测上升沿
1	1	检测下降沿

11.4.5 非屏蔽中断寄存器 (NMISRC)

非屏蔽中断寄存器 (NMISRC) 表示检测到不可屏蔽中断(NMI)请求。

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															SRC
属性	R															R
初始值	15'b0															0

寄存器功能

[bit15:1] Reserved: 保留位, 读取值没有定义。写无效。

[bit0] SRC: NMI 中断请求检测位

SRC 位对应引脚 NMIX。

bit	描述
0	未检测到NMI中断请求
1	检测到NMI中断请求
写	无效

11.4.6 非屏蔽中断清除寄存器 (NMICLR)

非屏蔽中断清除寄存器 (NMICLR) 清除保留的中断请求。

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	Reserved															CLR
属性	R															R/W
初始值	15'b0															1

寄存器功能

[bit15:1] Reserved: 保留位, 读取值没有定义。写无效。

[bit0] CLR: NMI 中断清除位, CLR 位对应引脚 NMIX。

bit	Function
写0	清除NMI中断请求
写1	无效
读	读取值为“1”

12. I/O 端口

12.1 概要

本节是 I/O 端口的概要。

本系列产品的 I/O 具有以下特征：

- 本系列 MCU 的 I/O 端口可复用作下列功能：

- GPIO

该通用 I/O 端口可读取输入电平并通过 CPU 设定输出电平。

- 快速 GPIO

快速 GPIO 功能可以在一个 CPU 时钟周期内读取或输出电平。具体细节请参考“快速通用 IO”章节

- 外设输入/输出

用作外设功能的数字输入输出信号端口。

- 特殊专用 I/O 端口

- 模拟输入端口

作为 A/D 转换器与电压比较器(VC)的模拟输入端口。

- 模拟输出端口

作为 LCDC 的模拟输出端口。

- 振荡器端口

- 各个引脚可如下设定：

- 可设置端口作为 GPIO 使用、作为外设的数字引脚使用、或作为特殊引脚使用。

- I/O 端口可以设置为输入口或者输出口。

- 可以使能或禁止上拉功能。

- 如果外设功能被分配到 2 个或更多的 I/O 端口，并且输入/输出相同的功能，则用户可以指定分配到其中的某个 I/O 端口 (重定位功能)。

- 根据寄存器设定，当 CPU 处于待机模式时，可把 I/O 端口设定到 Hi-Z 状态。

12.2 I/O 的结构、框图及操作说明

本节是 I/O 端口的构成、框图及操作说明。

12.2.1 I/O 端口结构

通过设定各 I/O 端口的寄存器，可以选择输入输出方向以及 GPIO/外设。图 12.2.1-1 详细说明了 I/O 端口。

图 12.2.1-1 I/O 端口框图

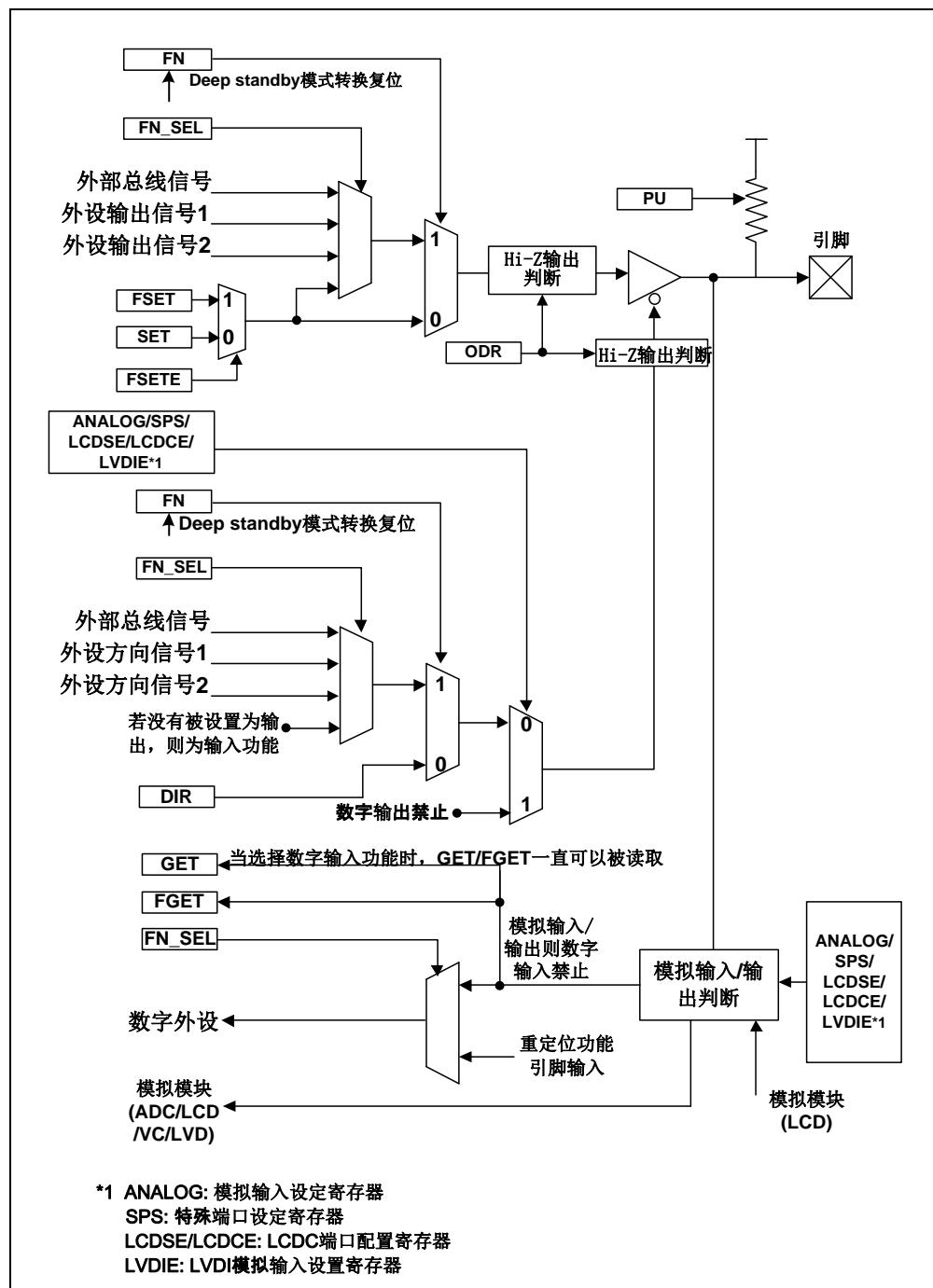


表 12.2.1-1 是寄存器的功能说明。

- 对于每个 I/O 端口，FN, DIR, GET, SET, PU 寄存器都拥有 1 位控制位，用来控制 I/O 端口的某项功能。
- 对于每个 I/O 端口，ANALOG 寄存器都拥有 1 位控制位，可将具有模拟输入功能的 I/O 端口设置为模拟输入引脚。
- SPS 寄存器是选择特殊引脚作为时钟引脚还是数字 IO 引脚的寄存器。
- 对于每个外设功能的 I/O 引脚，FN_SEL 寄存器都拥有一个控制位，并可以选择把外设模块的 引脚配置到相对应的 I/O 端口上。

表 12.2.1-1 寄存器功能说明

寄存器名称	功能说明
ANALOG	该寄存器可以设定 I/O端口是用作特殊引脚(模拟输入引脚)还是数字I/O引脚。
SPS	该寄存器可以设定 I/O端口被用作特殊功能引脚还是数字输入输出引脚。
FN	该寄存器可以设定 I/O端口是用作 GPIO 功能还是外设功能。
PU	I/O端口用作数字输入引脚和数字双向引脚时，该寄存器可以设定是连接还是切断 I/O端口的 上拉电阻。
ODR	该寄存器可以设定 I/O端口是否具有开漏功能 (open-drain)
DIR	I/O端口用作 GPIO 功能引脚时，该寄存器可以设定是用作输入引脚还是输出引脚。注：引脚选作外设功能的 I/O引脚时，此寄存器设定值无效。
GET	该寄存器可以读出 I/O端口的电平状态。 <ul style="list-style-type: none"> I/O端口用作数字输入引脚时，读出输入电平。 I/O端口用作数字输出引脚时，读出输出电平。 I/O端口用作模拟输入引脚时，始终读"0"。
SET	I/O端口用作GPIO功能的输出引脚时，该寄存器可以设定输出电平。 <ul style="list-style-type: none"> 清"0"时，输出低电平。 置"1"时，输出高电平。 注：引脚选作 GPIO输入引脚或外设功能的I/O引脚时，此寄存器设定值无效。
FN_SEL	该寄存器用于选择外设模块输入输出功能，并可进行重定位。 <ul style="list-style-type: none"> 设置外设输出引脚 可对 I/O端口的输出设定 ON/OFF。另外，可设定把外设功能的引脚重定位到 设置外设输入引脚 可设定把外设功能的每个引脚重定位到某个 I/O端口。 设置外设双向引脚 可设定把外设功能的每个引脚重定位到某个 I/O端口。

表 12.2.1-2 依据所选 I/O 端口功能，列出可支持的引脚功能及寄存器设定值。

表 12.2.1-2 I/O 端口功能和寄存器设定值

I/O 端口功能		ANAL OG/ SPS	FN	DIR	ODR	PU	FN_SEL
主功能	副功能						
特殊引脚 (模拟输入/输出，振荡器)	无	1	-	-	-	断开	*0
GPIO 功能输入 引脚	外设功能输入引脚	0	0	0	0	有效	*1
				0	1	断开	
				1	0	断开	
				1	1	断开	
选择外设功能 的输出引脚	GPIO 功能输入引脚(FB) 外设功能输入引脚(FB)	0	1	0	断开	*2	
				1	断开		
选择外设功能 的双向引脚	GPIO 功能输入引脚(FB) 外设功能输入引脚(FB)	1	-	0	有效	*3	
				1	断开		
外设功能输入 引脚	GPIO 功能输入引脚			0	有效	*4	
				1	断开		

注释：

-：指示寄存器设定值对引脚功能无影响。

有效：指示 PU 寄存器值是 0 时，上拉阻抗被切断。指示 PU 寄存器值是 1 时，上拉阻抗连接。

断开：指示不论 PU 寄存器的值如何，均断开上拉阻抗。

(FB)：指示 I/O 端口部分的输出信号提供反馈，并可从 GET 读出 I/O 端口的电平。信号也可用作外设功能输入。

*0：若 I/O 端口选择外设功能的输入引脚，设定无效。若 I/O 端口选择外设功能的输出引脚，设定无效。若 I/O 端口选择外设功能的双向引脚，设定无效。

*1：若 I/O 端口选择外设功能的输入引脚，设定有效。若 I/O 端口选择外设功能的输出引脚，设定无效。若 I/O 端口选择外设功能的双向引脚，设定无效。

*2：指示 I/O 端口选择外设功能的输出引脚。

*3：指示 I/O 端口选择外设功能的双向引脚。

*4：指示 I/O 端口既不选择外设功能的输出引脚也不选择双向引脚。

12.2.2 I/O 端口的初始选择功能

表 12.2.2-1 列出了复位后各 I/O 端口的初始选择功能。

表 12.2.2-1 释放复位后各 I/O 端口的初始选择功能

No	引脚	初始选择功能
1	SWCLK, SWDIO	选择 SWD(serial wire debug)引脚，使能上拉。
2	上述引脚以外的其他GPIO引脚	数字输入，输出为Hi-Z。

注意：

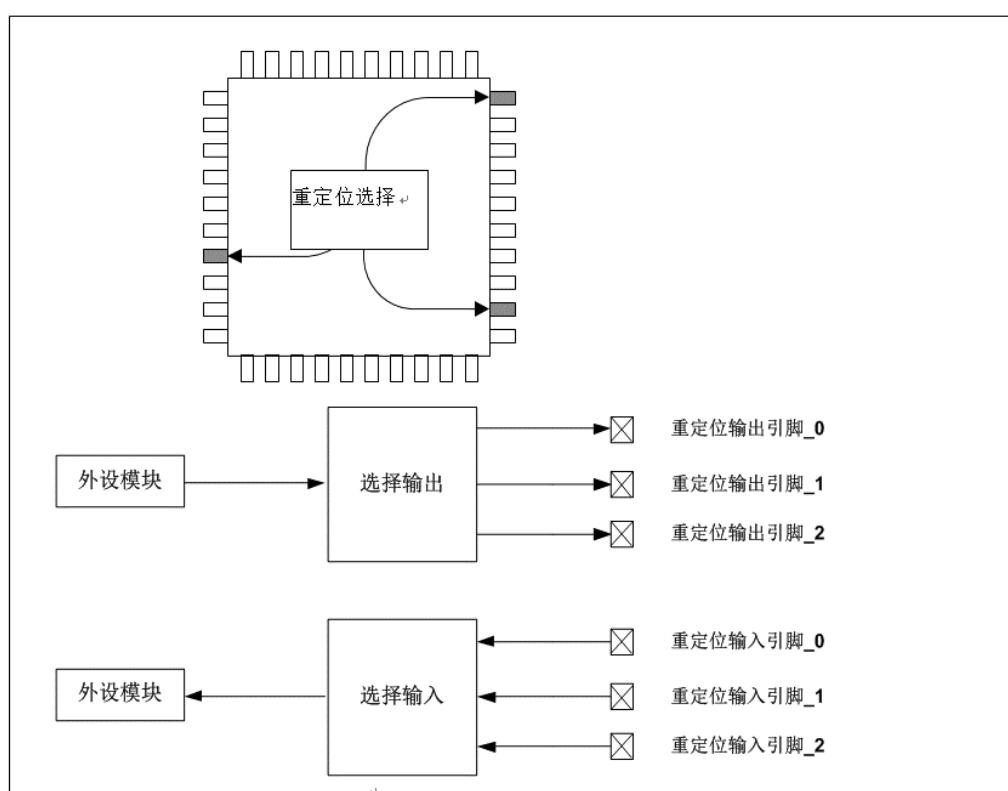
- 关于 GPIO 以外的引脚(MODE 引脚、RSTB 复位引脚)状态，参见“数据手册”。
- 复位时所有 FN_SEL 的输出选择值均为“不输出”。

12.2.3 重定位功能

部分外设功能的 I/O 拥有的引脚数不止一个(重定位引脚)。

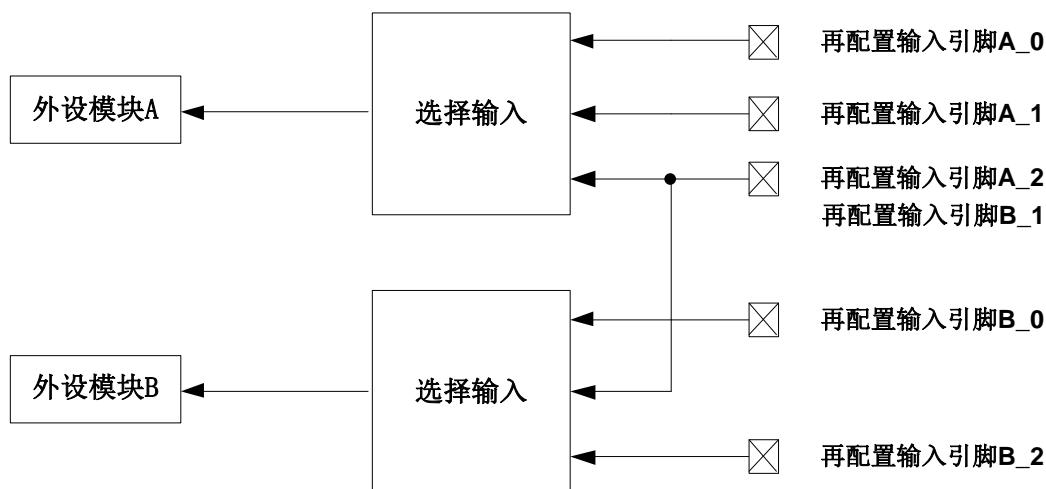
通过设定 FN_SEL 可以选择其中一个引脚。图 12.2.3-1 是重定位功能的示意图。

图 12.2.3-1 重定位功能示意图



注意：

- 外设功能和引脚的配置关系因产品而异。参见“数据手册”的引脚功能一览表。
- 即使同一个 I/O 端口连接到两个以上(包含两个)外设功能上作为输入口，也可以通过设定 FN_SEL 来完成所有的外设输入。以下图为例，通过选择“重定位输入引脚 A_2”和“重定位输入引脚 B_1”的输入，就可以同时使用这两个功能。因此，可以同时使用共用一个 I/O 端口的外部中断和多功能串行输入引脚。



- 即使 I/O 单元引脚设定为输出，因为不屏蔽输入，它仍然可以用作输入引脚。例如，定时器输出可以复用作外部中断输入。

12.2.4 FN_SEL 输出的固定优先级

在多个输出中仅可选择一个输出引脚功能，用于一个 I/O 端口。

如果通过 FN_SEL 寄存器设定了多个输出，则由固定优先级选择输出引脚。图 12.2.4-1 是输出引脚和固定优先级图。

图 12.2.4-1 输出引脚和固定优先级



FN_SEL 的固定优先顺序如表 12.2.4-1 所示。

表 12.2.4-1 FN_SEL 的固定优先顺序

高优先级	外设功能	应用引脚
↓	GPIO	输入/输出引脚
↓	特殊输入	SWD输入、NMI输入
↓	SWD	输出引脚
↓	多功能串行	输出引脚
↓	基本定时器输出	输出引脚
↓	电压比较器	输出引脚
↓	SCI7816	输出引脚
↓	蜂鸣器	输出引脚
↓	内置 CR时钟输出	输出引脚
↓	RTCO	输出引脚
低优先级	SUBOUT	输出引脚

注意：

- 固定优先顺序只是针对多个输出功能同时被设置的情况下；当设定为“输入”时则不遵循固定优先顺序。
- 但是，“特殊输入”功能比“输出”具有更高的优先级，当同一个端口被设置为“特殊输入”时，其输出功能无效。
- 若引脚用作外设功能的输入引脚，须禁止其它与此外设共用的输出设置。若没有选 FN_SEL 寄存器上的引脚输出时，引脚用作外部输入引脚。

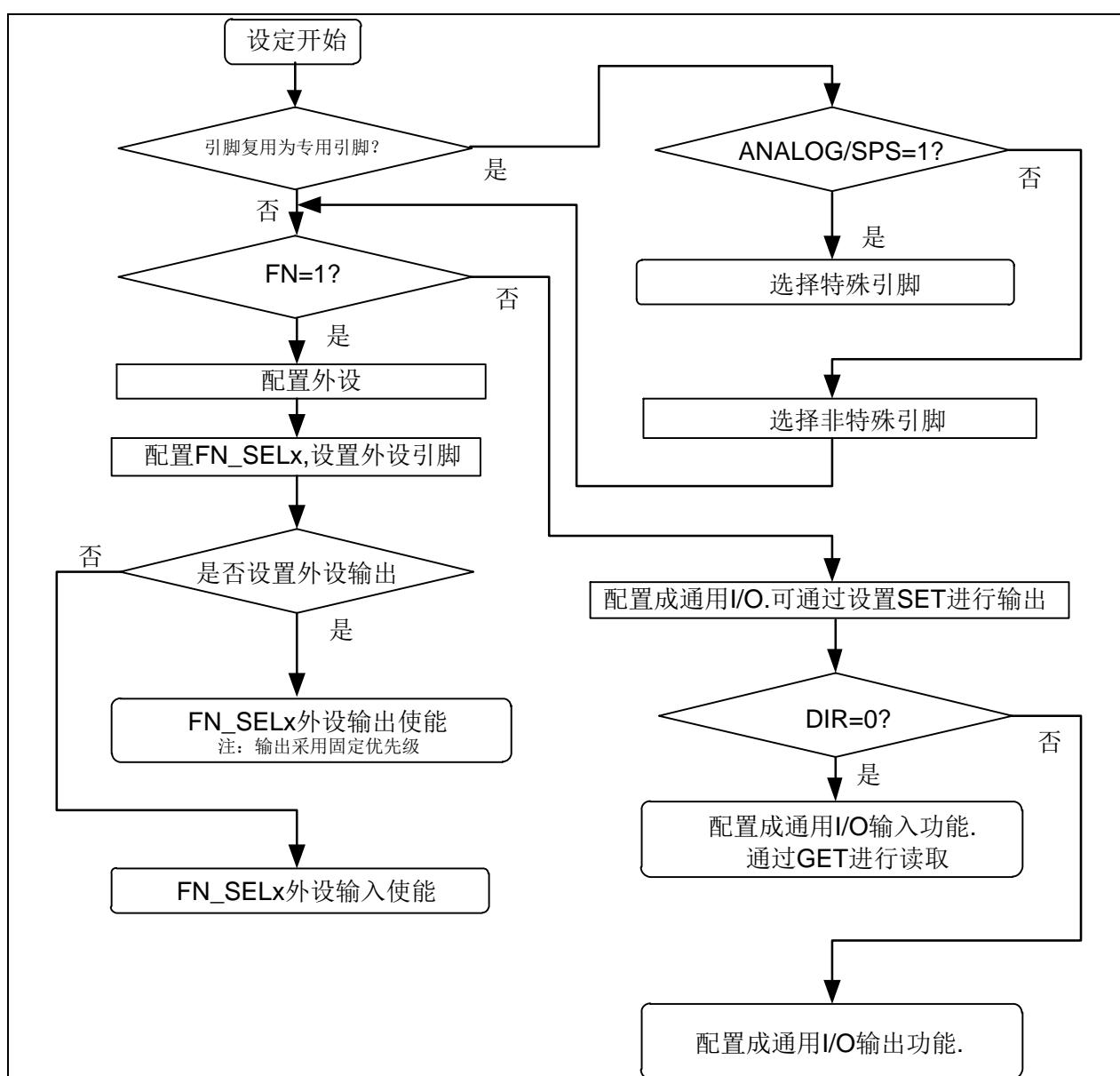
12.3 设定顺序示例

本节是 I/O 端口的设定顺序示例。

12.3.1 I/O 端口的设置

通过设置 I/O 端口各寄存器，可以选择 I/O 方向或者确定 I/O 端口作为 GPIO 或外设端口。图 12.3.1-1 是设定顺序示例。

图 12.3.1-1 I/O 端口的设定顺序示例



12.4 寄存器

本节提供了 I/O 端口的寄存器列表。寄存器一览如表 12.4-1 所示：

表 12.4-1 I/O 端口寄存器列表

缩写	名称	参考章节
FN0	端口功能设定寄存器 0	12.4.1
FN1	端口功能设定寄存器 1	
FN2	端口功能设定寄存器 2	
FN3	端口功能设定寄存器 3	
FN4	端口功能设定寄存器 4	
FN5	端口功能设定寄存器 5	
FN6	端口功能设定寄存器 6	
FN7	端口功能设定寄存器 7	
FN8	端口功能设定寄存器 8	
PU0	上拉设定寄存器 0	12.4.2
PU1	上拉设定寄存器 1	
PU2	上拉设定寄存器 2	
PU3	上拉设定寄存器 3	
PU4	上拉设定寄存器 4	
PU5	上拉设定寄存器 5	
PU6	上拉设定寄存器 6	
PU7	上拉设定寄存器 7	
PU8	上拉设定寄存器 8	
DIR0	端口 I/O 方向设定寄存器 0	12.4.3
DIR1	端口 I/O 方向设定寄存器 1	
DIR2	端口 I/O 方向设定寄存器 2	
DIR3	端口 I/O 方向设定寄存器 3	
DIR4	端口 I/O 方向设定寄存器 4	
DIR5	端口 I/O 方向设定寄存器 5	
DIR6	端口 I/O 方向设定寄存器 6	
DIR7	端口 I/O 方向设定寄存器 7	
DIR8	端口 I/O 方向设定寄存器 8	

缩写	名称	参考章节
GET0	端口输入数据寄存器 0	12.4.4
GET1	端口输入数据寄存器 1	
GET2	端口输入数据寄存器 2	
GET3	端口输入数据寄存器 3	
GET4	端口输入数据寄存器 4	
GET5	端口输入数据寄存器 5	
GET6	端口输入数据寄存器 6	
GET7	端口输入数据寄存器 7	
GET8	端口输入数据寄存器 8	
SET0	端口输出数据寄存器 0	12.4.5
SET1	端口输出数据寄存器 1	
SET2	端口输出数据寄存器 2	
SET3	输出数据寄存器 3	
SET4	端口输出数据寄存器 4	
SET5	端口输出数据寄存器 5	
SET6	端口输出数据寄存器 6	
SET7	端口输出数据寄存器 7	
SET8	端口输出数据寄存器 8	
ODR0	端口开漏输出寄存器0	12.4.6
ODR1	端口开漏输出寄存器1	
ODR2	端口开漏输出寄存器2	
ODR3	端口开漏输出寄存器 3	
ODR4	端口开漏输出寄存器4	
ODR5	端口开漏输出寄存器5	
ODR6	端口开漏输出寄存器6	
ODR7	端口开漏输出寄存器7	
ODR8	端口开漏输出寄存器8	
LVDIE	LVDI模拟输入设置寄存器	12.4.7
ANALOG	模拟输入设定寄存器	12.4.8
FN_SEL00	扩展功能引脚设定寄存器 00	12.4.9
FN_SEL02	扩展功能引脚设定寄存器 02	12.4.10
FN_SEL03	扩展功能引脚设定寄存器 03	12.4.11

FN_SEL04	扩展功能引脚设定寄存器 04	12.4.12
FN_SEL05	扩展功能引脚设定寄存器 05	12.4.13
FN_SEL06	扩展功能引脚设定寄存器 06	12.4.14
FN_SEL07	扩展功能引脚设定寄存器 07	12.4.15
FN_SEL08	扩展功能引脚设定寄存器 08	12.4.16
FN_SEL09	扩展功能引脚设定寄存器 09	12.4.17
SPS	特殊端口设定寄存器	12.4.18

12.4.1 端口功能设定寄存器(FNx)

FNx 寄存器用于选择引脚的功能。

寄存器配置

31	8	7	0	初始值	属性	支持端口
保留位		FN0		0x0000	R/W	P07 to P00
保留位		FN1		0x0000	R/W	P17 to P10
保留位		FN2		0x0000	R/W	P27 to P20
保留位		FN3		0x0003	R/W	P37--P34, P31--P30
保留位		FN4		0x0000	R/W	P47--P45, P41--40
保留位		FN5		0x0000	R/W	P57 to P50
保留位		FN6		0x0000	R/W	P67 to P60
保留位		FN7		0x0000	R/W	P74 to P70
保留位		FN8		0x0000	R/W	P80

寄存器功能

bit	31	8	7	0
Field	保留位			FNx

[bit31:8] 保留位，读永远是"0x0000"。写无效。

[bit7:0] FNx: 端口功能设定寄存器 x

设定引脚的使用方法:

Bit7:0	说明
读	读出寄存器的设定值。
写 0	该引脚用作 GPIO引脚。
写 1	该引脚用作外设功能的 I/O引脚。

注意:

- FNx 的"x"是通配符，指示 FN0,FN1,FN2.....。
- Px0 的"x"是通配符。Px0 指示 P00,P10,P20.....。
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如，FN0 的第 7

位对应 P07; FN0 的第 6 位对应 P06; FN0 的第 0 位对应 P00。

- P31 ~ P30 上选择 SWD 引脚, 初始值="1"。
- 若对产品上不存在引脚的位写值无效, 读值不定。

12.4.2 上拉设定寄存器(PUx)

PUx 寄存器用于使能引脚的上拉电阻。

寄存器配置

31	8	7	0	初始值	属性	支持端口
保留位	PU0		0x0000	R/W	P07 to P00	
保留位	PU1		0x0000	R/W	P17 to P10	
保留位	PU2		0x0000	R/W	P27 to P20	
保留位	PU3		0x0003	R/W	P37--P34, P31-P30	
保留位	PU4		0x0000	R/W	P47--P45, P41--40	
保留位	PU5		0x0000	R/W	P57 to P50	
保留位	PU6		0x0000	R/W	P67 to P60	
保留位	PU7		0x0000	R/W	P74 to P70	
保留位	PU8		0x0000	R/W	P80	

寄存器功能

bit	31	8	7	0
Field	保留位			PUx

[bit31:8]保留位, 读永远是"0x0000"。写无效。

[bit7:0] PUx: 端口上拉功能设定寄存器 x

设定引脚的使用方法:

Bit7:0	说明
读	读出寄存器的设定值。
写 0	断开引脚的上拉电阻。
写 1	若引脚处于输入状态 (GPIO或外设功能), 连接上拉电阻 若引脚处于输出状态, 则断开上拉电阻

注意:

- PUx 的"x"是通配符, 指示 PU0,PU1, PU2.....。
- Px0 的"x"是通配符。Px0 指示 P00, P10, P20.....。
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如, PU0 的第 7 位对应 P07; PU0 的第 6 位对应 P06; PU0 的第 0 位对应 P00。
- P31 ~ P30 上选择 SWD 引脚, 初始值="1"。
- 若使用 I2C 时若设定 PUx=0, 需使用外部上拉。
- 若对产品上不存在引脚的位写值无效, 读值不定。

12.4.3 端口方向设定寄存器(DIRx)

DIRx 寄存器用于选择引脚的输入/输出方向。

寄存器配置

31	8	7	0	初始值	属性	支持端口
保留位		DIR0	0x0000	R/W	P07 to P00	
保留位		DIR1	0x0000	R/W	P17 to P10	
保留位		DIR2	0x0000	R/W	P27 to P20	
保留位		DIR3	0x0000	R/W	P37--P34, P31-P30	
保留位		DIR4	0x0000	R/W	P47--P45, P41--40	
保留位		DIR5	0x0000	R/W	P57 to P50	
保留位		DIR6	0x0000	R/W	P67 to P60	
保留位		DIR7	0x0000	R/W	P74 to P70	
保留位		DIR8	0x0000	R/W	P80	

寄存器功能

bit	31	8	7	0
Field	保留位			DIRx

[bit31:8]保留位，读永远是"0x0000"。写无效。

[bit7:0] DIRx: 端口方向设定寄存器 x

设定引脚的使用方法:

Bit7:0	说明
读	读出寄存器的设定值。
写 0	用作GPIO输入 若引脚选作外设功能，本设定值无效
写 1	用作GPIO输出 若引脚选作外设功能，本设定值无效

注意:

- DIRx 的"x"是通配符，指示 DIR0,DIR1,DIR2.....。
- Px0 的"x"是通配符。Px0 指示 P00, P10, P20.....。
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如，DIR0 的第 7 位对应 P07; DIR0 的第 6 位对应 P06; DIR0 的第 0 位对应 P00。
- 若对产品上不存在引脚的位写值无效，读值不定。

12.4.4 端口输入数据寄存器(GETx)

GETx 寄存器用于读取引脚的数值。

寄存器配置

31	8	7	0	初始值	属性	支持端口
保留位		GET0		0x0000	R	P07 to P00
保留位		GET1		0x0000	R	P17 to P10
保留位		GET2		0x0000	R	P27 to P20
保留位		GET3		0x0000	R	P37--P34, P31-P30
保留位		GET4		0x0000	R	P47--P45, P41--40
保留位		GET5		0x0000	R	P57 to P50
保留位		GET6		0x0000	R	P67 to P60
保留位		GET7		0x0000	R	P74 to P70
保留位		GET8		0x0000	R	P80

寄存器功能

bit	31	8	7	0
Field	保留位			GETx

[bit31:8]保留位，读永远是"0x0000"。写无效。

[bit7:0] GETx: 端口输入数据寄存器 x

设定引脚的使用方法：

Bit7:0	说明
读0	不管引脚功能(FN/FN_SEL/DIR/SET)如何设定，该位表示引脚处于"L"电平输入状态或者"L"电平输出状态。 若使用ANALOG/SPS选择特殊引脚，因为数字输入被断开，所以始终读为0
读1	不管引脚功能(FN/FN_SEL/DIR/SET)如何设定，该位表示引脚处于"H"电平输入状态或者"H"电平输出状态。
写1	写无效

注意：

- GETx 的"x"是通配符，指示 GET0,GET1, GET2.....。
- Px0 的"x"是通配符。Px0 指示 P00, P10, P20.....。
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如，GET0 的第 7 位对应 P07； GET0 的第 6 位对应 P06； GET0 的第 0 位对应 P00。
- 若对产品上不存在引脚的位写值无效，读值不定。

12.4.5 端口输出数据寄存器(SETx)

SETx 寄存器用于设置引脚的输出数值。

寄存器配置

31	8	7	0	初始值	属性	支持端口
保留位		SET0		0x0000	R/W	P07 to P00
保留位		SET1		0x0000	R/W	P17 to P10
保留位		SET2		0x0000	R/W	P27 to P20

保留位	SET3	0x0000	R/W	P37--P34, P31-P30
保留位	SET4	0x0000	R/W	P47--P45, P41--40
保留位	SET5	0x0000	R/W	P57 to P50
保留位	SET6	0x0000	R/W	P67 to P60
保留位	SET7	0x0000	R/W	P74 to P70
保留位	SET8	0x0000	R/W	P80

寄存器功能

bit	31	8	7	0
Field	保留位			SETx

[bit31:8]保留位，读永远是"0x0000"。写无效。

[bit7:0] SETx: 端口输出数据设定寄存器 x

设定引脚的使用方法:

Bit7:0	说明
读	读出寄存器值
写 0	向GPIO输出"L"电平 若引脚选作GPIO输入或者外设功能的I/O引脚，设定值无效
写 1	向GPIO输出"H"电平 若引脚选作GPIO输入或者外设功能的I/O引脚，设定值无效

注意:

- SETx 的"x"是通配符，指示 SET0,SET1, SET2.....。
- Px0 的"x"是通配符。Px0 指示 P00, P10, P20.....。
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如，SET0 的第 7 位对应 P07; SET0 的第 6 位对应 P06; SET0 的第 0 位对应 P00。
- 若对产品上不存在引脚的位写值无效，读值不定。

12.4.6 模拟输入设定寄存器(ANALOG)

ANALOG 寄存器用作设置 ADC/VC 等模拟信号输入引脚。

寄存器功能

bit	31	16	15	0
Field	保留位			ANALOG[15:0]
属性	--			R/W
初始值	--			0x0000

[bit31:16]保留位，读永远是"0x0000"。写无效。

[bit15:0] ANALOG[15:0]: 模拟输入设定寄存器

设定引脚的使用方法:

bit15:0	说明
读	读出寄存器值
写 0	引脚用作数字 I/O
写 1	引脚用作模拟输入 (I/O单元处于输入/输出断开、上拉断开状态)

12.4.7 扩展功能引脚设定寄存器(FN_SELx)

当同一个引脚具有多个功能时，使用 FN_SELx 寄存器配置引脚功能。

寄存器配置

寄存器	初始值	属性	相关的功能模块
FN_SEL00	0x00000100	R/W	系统功能
FN_SEL02	0x00000000	R/W	
FN_SEL03	0x00000000	R/W	基本定时器
FN_SEL04	0x00000000	R/W	外部中断
FN_SEL05	0x00000000	R/W	
FN_SEL06	0x00000000	R/W	
FN_SEL07	0x00000000	R/W	多功能串行接口
FN_SEL08	0x00000000	R/W	
FN_SEL09	0x00000000	R/W	电压比较器 蜂鸣器 SCI7816

12.4.7.1 扩展功能引脚设定寄存器 00(FN_SEL00)

当同一个引脚具有多个功能时，FN_SEL00 寄存器可为引脚配置功能。

寄存器配置

位	31	--	24	
名称		保留位		
初始值		8'h00		
属性		R		

位	23	--	16	
名称		保留位		
初始值		8'h00		
属性		R		

位	15		9	8
名称		保留		SWDEN
初始值		-		1
属性		-		R/W

位	7 -- 6	5 -- 4	3	2 -- 1	0
名称	SUBOUT_E	RTCCOE	保留	CROUTE	NMIS
初始值	0	0	-	0	0
属性	R/W	R/W	-	R/W	R/W

寄存器功能

[bit31:9] 保留位，写无效

[bit8] SWDEN: 调试接口选择

Bit8	说明
读	读出寄存器值
写 0	不使用SWCLK与SWDIO引脚作为调试接口
写 1	使用SWCLK与SWDIO引脚作为调试接口

[bit7:6] SUBOUTE: RTC 时钟分频输出选择

Bit7:6	说明
读	读出寄存器值
写 00	不输出RTC时钟的分频
写 01	将SUBOUT_0用作RTC时钟分频输出
写 10	将SUBOUT_1用作RTC时钟分频输出
写 11	将SUBOUT_2用作RTC时钟分频输出

[bit5:4] RTCCOE: RTC 输出选择

Bit5:4	说明
读	读出寄存器值
写 00	不输出实时时钟
写 01	将RTCO_0用作实时时钟输出
写 10	将RTCO_1用作实时时钟输出
写 11	将RTCO_2用作实时时钟输出

[bit3]保留位，读永远是 0，写无效

[bit2:1] CROUTE: 内部高速 CR 输出选择

Bit2:1	说明
读	读出寄存器值
写 00	不输出内部高速晶振时钟
写 01	将IRCO_0用作内部高速时钟输出
写 10	将IRCO_1用作内部高速时钟输出
写 11	将IRCO_2用作内部高速时钟输出

[bit0] NMIS: 非屏蔽中断(Non-Masked interrupt)选择

Bit0	说明
读	读出寄存器值
写 0	不使用NMIX输入引脚
写 1	使用NMIX输入引脚

12.4.7.2 扩展功能引脚设定寄存器 02(FN_SEL02)

当同一个引脚具有多个功能时，FN_SEL02 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 30	29 -- 28	27 -- 26	25 -- 24
名称	保留位	TIOB3S[1:0]	TIOA3E[1:0]	TIOA3S[1:0]
初始值	0	0	0	0
属性	-	R/W	R/W	R/W

位	23	22 -- 20	19 -- 18	17 -- 16
名称	保留位	TIOB2S[2:0]	TIOA2E[1:0]	保留位
初始值	0	0	0	0
属性	-	R/W	R/W	-

位	15 -- 14	13 -- 12	11 -- 10	3 -- 2
名称	保留位	TIOB1S[1:0]	TIOA1E[1:0]	TIOA1S[1:0]
初始值	0	0	0	0
属性	-	R/W	R/W	R/W

位	7	6 -- 4	3 -- 2	1 -- 0
名称	保留位	TIOB0S[2:0]	TIOA0E[1:0]	保留位
初始值	0	0	0	0
属性	-	R/W	R/W	-

寄存器功能

[bit31:30] 保留位，读永远是 0，写无效

[bit29:28] TIOB3S: TIOB3 输入选择位

bit29:28	说明
读	读出寄存器值
写 00	基本定时器的ch3-TIOB的输入引脚使用 TIOB3_0
写 01	与写 00时相同
写 10	基本定时器的ch3-TIOB的输入引脚使用 TIOB3_1
写 11	基本定时器的ch3-TIOB的输入引脚使用 TIOB3_2

[bit27:26] TIOA3E: TIOA3E 输出选择位

bit27:26	说明
读	读出寄存器值
写 00	基本定时器的ch3-TIOA不进行输出
写 01	基本定时器的ch3-TIOA的输出引脚使用 TIOA3_0
写 10	基本定时器的ch3-TIOA的输出引脚使用 TIOA3_1
写 11	基本定时器的ch3-TIOA的输出引脚使用 TIOA3_2

[bit25:24] TIOA3S: TIOA3 输入选择位

bit25:24	说明
----------	----

读	读出寄存器值
写 00	基本定时器的ch3-TIOA的输入引脚使用 TIOA3_0
写 01	与写 00时相同
写 10	基本定时器的ch3-TIOA的输入引脚使用 TIOA3_1
写 11	基本定时器的ch3-TIOA的输入引脚使用 TIOA3_2

[bit23]保留位，读永远是 0，写无效

[bit22:20] TIOB2S: TIOB2 输入选择位

bit21:20	说明
读	读出寄存器值
写 000	基本定时器的ch2-TIOB的输入引脚使用TIOB2_0
写 001	与写 00时相同
写 010	基本定时器的ch2-TIOB的输入引脚使用TIOB2_1
写 011	基本定时器的ch2-TIOB的输入引脚使用TIOB2_2
写 100	基本定时器的ch2-TIOB的输入使用VC输出(需配置)
写 101	无效
写 110	无效
写 111	无效

[bit19:18] TIOA2E: TIOA2 输出选择位

bit19:18	说明
读	读出寄存器值
写 00	基本定时器的ch2-TIOA不进行输出
写 01	基本定时器的ch2-TIOA的输出引脚使用 TIOA2_0
写 10	基本定时器的ch2-TIOA的输出引脚使用 TIOA2_1
写 11	基本定时器的ch2-TIOA的输出引脚使用 TIOA2_2

[bit17:14]保留位，读永远是 0，写无效

[bit13:12] TIOB1S: TIOB1 输入选择位

bit13:12	说明
读	读出寄存器值
写 00	基本定时器的ch1-TIOB的输入引脚使用 TIOB1_0
写 01	与写 00 时相同
写 10	基本定时器的ch1-TIOB的输入引脚使用 TIOB1_1
写 11	基本定时器的ch1-TIOB的输入引脚使用 TIOB1_2

[bit11:10] TIOA1E: TIOA1E 输出选择位

bit11:10	说明
读	读出寄存器值。
写 00	基本定时器的ch1-TIOA不进行输出
写 01	基本定时器的ch1-TIOA的输出引脚使用 TIOA1_0
写 10	基本定时器的ch1-TIOA的输出引脚使用 TIOA1_1
写 11	基本定时器的ch1-TIOA的输出引脚使用 TIOA1_2

[bit9:8] TIOA1S: TIOA1 输入选择位

bit9:8	说明
读	读出寄存器值

写 00	基本定时器的ch1-TIOA的输入引脚使用 TIOA1_0
写 01	与写 00 时相同
写 10	基本定时器的ch1-TIOA的输入引脚使用 TIOA1_1
写 11	基本定时器的ch1-TIOA的输入引脚使用 TIOA1_2

bit7]保留位，读永远是 0，写无效

[bit6:4] TIOB0S: TIOB0 输入选择位

Bit6:4	说明
读	读出寄存器值
写 000	基本定时器的ch0-TIOB的输入引脚使用TIOB0_0
写 001	与写 000时相同
写 010	基本定时器的ch0-TIOB的输入引脚使用TIOB0_1
写 011	基本定时器的ch0-TIOB的输入引脚使用TIOB0_2
写 100	基本定时器的ch0-TIOB的输入使用VC输出(需配置)
写 101	无效
写 110	基本定时器的ch0-TIOB的输入引脚使用SUBOUT
写 111	基本定时器的ch0-TIOB的输入使用内部高速CR分频之后的时钟用于内部校准

[bit3:2] TIOA0E: TIOA0 输出选择位

bit3:2	说明
读	读出寄存器值
写 00	基本定时器的ch0-TIOA不进行输出
写 01	基本定时器的ch0-TIOA的输出引脚使用 TIOA0_0
写 10	基本定时器的ch0-TIOA的输出引脚使用 TIOA0_1
写 11	基本定时器的ch0-TIOA的输出引脚使用 TIOA0_2

[bit1:0]保留位，读永远是 0，写无效

注意：

- TIOA 偶数通道仅可输出，奇数通道既可输出又可输入。
- TIOB 仅输入。
- TIOA1, TIOA3, TIOA5, TIOA7(A 的奇数号)并非作为双向引脚使用，而是可选择作为输入引脚或输出引脚使用。
- 在 TIOA 的奇数通道选择输出时，输入设定被忽略。

例 1：TIOA1 作为输出引脚使用时

输出 TIOA1 到 TIOA1_0 时，选择 FN_SEL02.TIOA1E = 01

输出 TIOA1 到 TIOA1_1 时，选择 FN_SEL02.TIOA1E = 10

输出 TIOA1 到 TIOA1_2 时，选择 FN_SEL02.TIOA1E = 11

FN_SEL04.TIOA1S 的设定被忽略

所选择的引脚设置为 ANALOG=0, FN=1 (DIR 被忽略)

所选择的引脚复用的其他外设功能的输出须全部关闭

例 2: TIOA1 作为输入引脚使用时

选择 FN_SEL02.TIOA1E = 00

从 TIOA1_0 输入 TIOA1 时, 选择 FN_SEL02.TIOA1S = 00 或 01

从 TIOA1_1 输入 TIOA1 时, 选择 FN_SEL02.TIOA1S = 10

从 TIOA1_2 输入 TIOA1 时, 选择 FN_SEL02.TIOA1S = 11

所选择的引脚设置为 ANALOG=0, FN=1 (DIR 被忽略)。

所选择的引脚复用的其他外设功能引脚的输出须全部关闭

*当引脚被设置为输入时, 所选引脚 (GPIO、其他外设功能引脚) 复用的引脚输出可作为反馈输入至基本计时器。

12.4.7.3 扩展功能引脚设定寄存器 03(FN_SEL03)

当同一个引脚具有多个功能时，FN_SEL03 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 30	29 -- 28	27 -- 26	25 -- 24
名称	保留位	TIOB7S[1:0]	TIOA7E[1:0]	TIOA7S[1:0]
初始值	0	0	0	0
属性	-	R/W	R/W	R/W

位	23 -- 22	21 -- 20	19 -- 18	17 -- 16
名称	保留位	TIOB6S[2:0]	TIOA6E[1:0]	保留位
初始值	0	0	0	0
属性	-	R/W	R/W	-

位	15 -- 14	13 -- 12	11 -- 10	9 -- 8
名称	保留位	TIOB5S[1:0]	TIOA5E[1:0]	TIOA5S[1:0]
初始值	0	0	0	0
属性	-	R/W	R/W	R/W

位	7 -- 6	5 -- 4	3 -- 2	1 -- 0
名称	保留位	TIOB4S[2:0]	TIOA4E[1:0]	保留位
初始值	0	0	0	0
属性	-	R/W	R/W	-

寄存器功能

[bit31:30] 保留位，读永远是 0，写无效

[bit29:28] TIOB7S: TIOB7 输入选择位

bit29:28	说明
读	读出寄存器值
写 00	基本定时器的ch7-TIOB的输入引脚使用 TIOB7_0
写 01	与写 00时相同
写 10	基本定时器的ch7-TIOB的输入引脚使用 TIOB7_1
写 11	基本定时器的ch7-TIOB的输入引脚使用 TIOB7_2

[bit27:26] TIOA7E: TIOA7E 输出选择位

bit27:26	说明
读	读出寄存器值
写 00	基本定时器的ch7-TIOA不进行输出
写 01	基本定时器的ch7-TIOA的输出引脚使用 TIOA7_0
写 10	基本定时器的ch7-TIOA的输出引脚使用 TIOA7_1
写 11	基本定时器的ch7-TIOA的输出引脚使用 TIOA7_2

[bit25:24] TIOA7S: TIOA7 输入选择位

bit25:24	说明
读	读出寄存器值

写 00	基本定时器的ch7-TIOA的输入引脚使用 TIOA7_0
写 01	与写 00时相同
写 10	基本定时器的ch7-TIOA的输入引脚使用 TIOA7_1
写 11	基本定时器的ch7-TIOA的输入引脚使用 TIOA7_2

[bit23:22]保留位，读永远是 0，写无效

[bit21:20] TIOB6S: TIOB6 输入选择位

bit21:20	说明
读	读出寄存器值
写 00	基本定时器的ch6-TIOB的输入引脚使用TIOB6_0
写 01	与写 00时相同
写 10	基本定时器的ch6-TIOB的输入引脚使用TIOB6_1
写 11	基本定时器的ch6-TIOB的输入引脚使用TIOB6_2

[bit19:18] TIOA6E: TIOA6 输出选择位

bit19:18	说明
读	读出寄存器值
写 00	基本定时器的ch6-TIOA不进行输出
写 01	基本定时器的ch6-TIOA的输出引脚使用 TIOA6_0
写 10	基本定时器的ch6-TIOA的输出引脚使用 TIOA6_1
写 11	基本定时器的ch6-TIOA的输出引脚使用 TIOA6_2

[bit17:14]保留位，读永远是 0，写无效

[bit13:12] TIOB5S: TIOB5 输入选择位

bit13:12	说明
读	读出寄存器值
写 00	基本定时器的ch5-TIOB的输入引脚使用 TIOB5_0
写 01	与写 00 时相同
写 10	基本定时器的ch5-TIOB的输入引脚使用 TIOB5_1
写 11	基本定时器的ch5-TIOB的输入引脚使用 TIOB5_2

[bit11:10] TIOA5E: TIOA5E 输出选择位

bit11:10	说明
读	读出寄存器值。
写 00	基本定时器的ch5-TIOA不进行输出
写 01	基本定时器的ch5-TIOA的输出引脚使用 TIOA5_0
写 10	基本定时器的ch5-TIOA的输出引脚使用 TIOA5_1
写 11	基本定时器的ch5-TIOA的输出引脚使用 TIOA5_2

[bit9:8] TIOA5S: TIOA5 输入选择位

bit9:8	说明
读	读出寄存器值
写 00	基本定时器的ch5-TIOA的输入引脚使用 TIOA5_0
写 01	与写 00 时相同
写 10	基本定时器的ch5-TIOA的输入引脚使用 TIOA5_1
写 11	基本定时器的ch5-TIOA的输入引脚使用 TIOA5_2

[bit7:6]保留，读永远是 0，写无效

[bit5:4] TIOB4S: TIOB4 输入选择位

Bit5:4	说明
读	读出寄存器值
写 00	基本定时器的ch4-TIOB的输入引脚使用TIOB4_0
写 01	与写 00时相同
写 10	基本定时器的ch4-TIOB的输入引脚使用TIOB4_1
写 11	基本定时器的ch4-TIOB的输入引脚使用TIOB4_2

[bit3:2] TIOA4E: TIOA4 输出选择位

bit3:2	说明
读	读出寄存器值
写 00	基本定时器的ch4-TIOA不进行输出
写 01	基本定时器的ch4-TIOA的输出引脚使用 TIOA4_0
写 10	基本定时器的ch4-TIOA的输出引脚使用 TIOA4_1
写 11	基本定时器的ch4-TIOA的输出引脚使用 TIOA4_2

[bit1:0]保留位，读永远是 0，写无效

12.4.7.4 扩展功能引脚设定寄存器 04(FN_SEL04)

当同一个引脚具有多个功能时，FN_SEL04 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 30	29 -- 28	27 -- 26	25 -- 24
名称	EINT15S	EINT14S	EINT13S	EINT12S
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

位	23 -- 22	21 -- 20	19 -- 18	17 -- 16
名称	EINT11S	EINT10S	EINT09S	EINT08S
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

位	15 -- 14	13 -- 12	11 -- 10	3 -- 2
名称	EINT07S	EINT06S	EINT05S	EINT04S
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W
位	7 -- 6	5 -- 4	3 -- 2	1 -- 0
名称	EINT03S	EINT02S	EINT01S	EINT00S
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

寄存器功能

[bit31:30] EINT15S: 外部中断输入选择位

bit31:30	说明
读	读出寄存器值
写 00	外部中断的ch15 的输入引脚使用 INT15_0
写 01	与写 00 时相同
写 10	外部中断的ch15 的输入引脚使用 INT15_1
写 11	外部中断的ch15 的输入引脚使用 INT15_2

[bit29:28] EINT14S: 外部中断输入选择位

bit29:28	说明
读	读出寄存器值。
写 00	外部中断的ch14 的输入引脚使用 INT14_0
写 01	与写 00 时相同
写 10	外部中断的ch14 的输入引脚使用 INT14_1
写 11	外部中断的ch14 的输入引脚使用 INT14_2

[bit27:26] EINT13S: 外部中断输入选择位

bit27:26	说明
读	读出寄存器值
写 00	外部中断的ch13 的输入引脚使用 INT13_0
写 01	与写 00 时相同
写 10	外部中断的ch13 的输入引脚使用 INT13_1

写 11	外部中断的ch13 的输入引脚使用 INT13_2
------	---------------------------

[bit25:24] EINT12S: 外部中断输入选择位

bit25:24	说明
读	读出寄存器值
写 00	外部中断的ch12 的输入引脚使用 INT12_0
写 01	与写 00 时相同
写 10	外部中断的ch12 的输入引脚使用 INT12_1
写 11	外部中断的ch12 的输入引脚使用 INT12_2

[bit23:22] EINT11S: 外部中断输入选择位

bit23:22	说明
读	读出寄存器值
写 00	外部中断的ch11 的输入引脚使用 INT11_0
写 01	与写 00 时相同
写 10	外部中断的ch11 的输入引脚使用 INT11_1
写 11	外部中断的ch11 的输入引脚使用 INT11_2

[bit21:20] EINT10S: 外部中断输入选择位

bit21:20	说明
读	读出寄存器值
写 00	外部中断的ch10 的输入引脚使用 INT10_0
写 01	与写 00 时相同
写 10	外部中断的ch10 的输入引脚使用 INT10_1
写 11	外部中断的ch10 的输入引脚使用 INT10_2

[bit19:18] EINT09S: 外部中断输入选择位

bit19:18	说明
读	读出寄存器值
写 00	外部中断的ch9 的输入引脚使用 INT09_0
写 01	与写 00 时相同
写 10	外部中断的ch9 的输入引脚使用 INT09_1
写 11	外部中断的ch9 的输入引脚使用 INT09_2

[bit17:16] EINT08S: 外部中断输入选择位

bit17:16	说明
读	读出寄存器值
写 00	外部中断的ch8 的输入引脚使用 INT08_0
写 01	与写 00 时相同
写 10	外部中断的ch8 的输入引脚使用 INT08_1
写 11	外部中断的ch8 的输入引脚使用 INT08_2

[bit15:14] EINT07S: 外部中断输入选择位

bit15:14	说明
读	读出寄存器值
写 00	外部中断的ch7 的输入引脚使用 INT07_0
写 01	与写 00 时相同
写 10	外部中断的ch7 的输入引脚使用 INT07_1
写 11	外部中断的ch7 的输入引脚使用 INT07_2

[bit13:12] EINT06S: 外部中断输入选择位

bit13:12	说明
读	读出寄存器值
写 00	外部中断的ch6 的输入引脚使用 INT06_0
写 01	与写 00 时相同
写 10	外部中断的ch6 的输入引脚使用 INT06_1
写 11	外部中断的ch6 的输入引脚使用 INT06_2

[bit11:10] EINT05S: 外部中断输入选择位

bit11:10	说明
读	读出寄存器值
写 00	外部中断的ch5 的输入引脚使用 INT05_0
写 01	与写 00 时相同
写 10	外部中断的ch5 的输入引脚使用 INT05_1
写 11	保留

[bit9:8] EINT04S: 外部中断输入选择位

bit9:8	说明
读	读出寄存器值
写 00	外部中断的ch4 的输入引脚使用 INT04_0
写 01	与写 00 时相同
写 10	外部中断的ch4 的输入引脚使用 INT04_1
写 11	保留

[bit7:6] EINT03S: 外部中断输入选择位

bit7:6	说明
读	读出寄存器值
写 00	外部中断的ch3 的输入引脚使用 INT03_0
写 01	与写 00 时相同
写 10	外部中断的ch3 的输入引脚使用 INT03_1
写 11	外部中断的ch3 的输入引脚使用 INT03_2

[bit5:4] EINT02S: 外部中断输入选择位

bit5:4	说明
读	读出寄存器值
写 00	外部中断的ch2 的输入引脚使用 INT02_0
写 01	与写 00 时相同
写 10	外部中断的ch2 的输入引脚使用 INT02_1
写 11	外部中断的ch2 的输入引脚使用 INT02_2

[bit3:2] EINT01S: 外部中断输入选择位

bit3:2	说明
读	读出寄存器值
写 00	外部中断的ch1 的输入引脚使用 INT01_0
写 01	与写 00 时相同
写 10	外部中断的ch1 的输入引脚使用 INT01_1
写 11	保留

[bit1:0] EINT00S: 外部中断输入选择位

bit1:0	说明
读	读出寄存器值

写 00	外部中断的ch0 的输入引脚使用 INT00_0
写 01	与写 00 时相同
写 10	外部中断的ch0 的输入引脚使用 INT00_1
写 11	外部中断的ch0 的输入引脚使用 INT00_2

12.4.7.5 扩展功能引脚设定寄存器 05(FN_SEL05)

当同一个引脚具有多个功能时，FN_SEL05 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 28	27 -- 26	25 -- 24
名称	保留位	SCS10B[1:0]	SCK1B[1:0]
初始值	0	0	0
属性	-	R/W	R/W

位	23 -- 22	21 -- 20	19 -- 18	17 -- 16
名称	SOT1B[1:0]	SIN1S[1:0]	CTS1S[1:0]	RTS1E[1:0]
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

位	15 -- 10	9 -- 8
名称	保留位	SCK0B[1:0]
初始值	0	0
属性	-	R/W

位	7 -- 6	5 -- 4	3 -- 0
名称	SOT0B[1:0]	SIN0S[1:0]	保留位
初始值	0	0	0
属性	R/W	R/W	-

寄存器功能

[bit31:28]保留位，读永远是 0，写无效

[bit27:26] SCS10B: SCS10 I/O 选择位

bit27:26	说明
读	读出寄存器值
写 00	多功能串口的ch1-SCS10 的输入引脚使用 SCS10_0 不输出 (初始值)
写 01	多功能串口的ch1-SCS10 的输入引脚使用 SCS10_0 输出引脚使用 SCS10_0
写 10	多功能串口的ch1-SCS10 的输入引脚使用 SCS10_1 输出引脚使用 SCS10_1
写 11	多功能串口的ch1-SCS10 的输入引脚使用 SCS10_2 输出引脚使用 SCS10_2

[bit25:24] SCK1B: SCK1 输入输出选择位

Bit25:24	说明
读	读出寄存器值
写 00	多功能串口的ch1-SCK 的输入引脚使用 SCK1_0 不输出 (初始值)

写 01	多功能串口的ch1-SCK 的输入引脚使用 SCK1_0 输出引脚使用 SCK1_0
写 10	多功能串口的ch1-SCK 的输入引脚使用 SCK1_1 输出引脚使用 SCK1_1
写 11	多功能串口的ch1-SCK 的输入引脚使用 SCK1_2 输出引脚使用 SCK1_2

[bit23:22] SOT1B: SOT1B 输入输出选择位

Bit23:22	说明
读	读出寄存器值
写 00	多功能串口的ch1-SOT 的输入引脚使用 SOT1_0 不输出 (初始值)
写 01	多功能串口的ch1-SOT 的输入引脚使用 SOT1_0 输出引脚使用 SOT1_0
写 10	多功能串口的ch1-SOT 的输入引脚使用 SOT1_1 输出引脚使用 SOT1_1
写 11	多功能串口的ch1-SOT 的输入引脚使用 SOT1_2 输出引脚使用 SOT1_2

[bit21:20] SIN1S: SIN1S 输入选择位

Bit21:20	说明
读	读出寄存器值
写 00	多功能串口的ch1-SIN 的输入引脚使用 SIN1_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch1-SIN 的输入引脚使用 SIN1_1
写 11	多功能串口的ch1-SIN 的输入引脚使用 SIN1_2

[bit19:18] CTS1S: CTS1S 输入选择位

Bit19:18	说明
读	读出寄存器值
写 00	多功能串口的ch1-CTS 的输入引脚使用 CTS1_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch1-CTS 的输入引脚使用 CTS1_1
写 11	多功能串口的ch1-CTS 的输入引脚使用 CTS1_2

[bit17:16] RTS1E: RTS1E 输出选择位

bit17:16	说明
读	读出寄存器值
写 00	多功能串口的ch1-RTS 不进行输出 (初始值)
写 01	多功能串口的ch1-RTS 的输出引脚使用 RTS1_0
写 10	多功能串口的ch1-RTS 的输出引脚使用 RTS1_1
写 11	多功能串口的ch1-RTS 的输出引脚使用 RTS1_2

[bit15:10] 保留位, 读永远是 0, 写无效

[bit9:8] SCK0B: SCK0 输入输出选择位

bit9:8	说明
读	读出寄存器值
写 00	多功能串口的ch0-SCK 的输入引脚使用 SCK0_0 不输出 (初始值)

写 01	多功能串口的ch0-SCK 的输入引脚使用 SCK0_0 输出引脚使用 SCK0_0
写 10	多功能串口的ch0-SCK 的输入引脚使用 SCK0_1 输出引脚使用 SCK0_1
写 11	多功能串口的ch0-SCK 的输入引脚使用 SCK0_2 输出引脚使用 SCK0_2

[bit7:6] SOT0B: SOT0B 输入输出选择位

bit7:6	说明
读	读出寄存器值
写 00	多功能串口的ch0-SOT 的输入引脚使用 SOT0_0 不输出 (初始值)
写 01	多功能串口的ch0-SOT 的输入引脚使用 SOT0_0 输出引脚使用 SOT0_0
写 10	多功能串口的ch0-SOT 的输入引脚使用 SOT0_1 输出引脚使用 SOT0_1
写 11	多功能串口的ch0-SOT 的输入引脚使用 SOT0_2 输出引脚使用 SOT0_2

[bit5:4] SIN0S: SIN0S 输入选择位

bit5:4	说明
读	读出寄存器值
写 00	多功能串口的ch0-SIN 的输入引脚使用 SIN0_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch0-SIN 的输入引脚使用 SIN0_1
写 11	多功能串口的ch0-SIN 的输入引脚使用 SIN0_2

[bit3:0]保留位，读永远是 0，写无效

12.4.7.6 扩展功能引脚设定寄存器 06(FN_SEL06)

当同一个引脚具有多个功能时，FN_SEL06 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 24
名称	保留位
初始值	8'h00
属性	R

位	23 -- 16
名称	保留位
初始值	8h00
属性	R

位	15 -- 14	13 -- 12	11 -- 10	9 -- 8
名称	SCS22E[1:0]	SCS21E[1:0]	SCS20B[1:0]	SCK2B[1:0]
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

位	7 -- 6	5 -- 4	3 -- 0
名称	SOT2B[1:0]	SIN2S[1:0]	保留位
初始值	0	0	0
属性	R/W	R/W	-

寄存器功能

[bit31:16] 保留位，读永远是 0，写无效

[bit15:14] SCS22E: SCS22E 输出选择位

Bit15:14	说明
读	读出寄存器值
写 00	不输出多功能串口的ch2-SCS22 (初始值)
写 01	多功能串口的ch2- SCS22 的输出引脚使用 SCS22_0
写 10	多功能串口的ch2- SCS22 的输出引脚使用 SCS22_1
写 11	多功能串口的ch2- SCS22 的输出引脚使用 SCS22_2

[bit13:12] SCS21E: SCS21E 输出选择位

Bit13:12	说明
读	读出寄存器值
写 00	不输出多功能串口的ch2- SCS21 (初始值)
写 01	多功能串口的ch2- SCS21 的输出引脚使用 SCS21_0
写 10	多功能串口的ch2- SCS21 的输出引脚使用 SCS21_1
写 11	多功能串口的ch2- SCS21 的输出引脚使用 SCS21_2

[bit11:10] SCS20B: SCS20B 输入输出选择位

Bit11:10	说明
读	读出寄存器值

写 00	多功能串口的ch2-SCS20 的输入引脚使用 SCS20_0 不输出 (初始值)
写 01	多功能串口的ch2-SCS20 的输入引脚使用 SCS20_0 输出引脚使用 SCS20_0
写 10	多功能串口的ch2-SCS20 的输入引脚使用 SCS20_1 输出引脚使用 SCS20_1
写 11	多功能串口的ch2-SCS20 的输入引脚使用 SCS20_2 输出引脚使用 SCS20_2

[bit9:8] SCK2B: SCK2 输入输出选择位

bit9:8	说明
读	读出寄存器值
写 00	多功能串口的ch2-SCK 的输入引脚使用 SCK2_0 不输出 (初始值)
写 01	多功能串口的ch2-SCK 的输入引脚使用 SCK2_0 输出引脚使用 SCK2_0
写 10	多功能串口的ch2-SCK 的输入引脚使用 SCK2_1 输出引脚使用 SCK2_1
写 11	多功能串口的ch2-SCK 的输入引脚使用 SCK2_2 输出引脚使用 SCK2_2

[bit7:6] SOT2B: SOT2B 输入输出选择位

bit7:6	说明
读	读出寄存器值
写 00	多功能串口的ch2-SOT 的输入引脚使用 SOT2_0 不输出 (初始值)
写 01	多功能串口的ch2-SOT 的输入引脚使用 SOT2_0 输出引脚使用 SOT2_0
写 10	多功能串口的ch2-SOT 的输入引脚使用 SOT2_1 输出引脚使用 SOT2_1
写 11	多功能串口的ch2-SOT 的输入引脚使用 SOT2_2 输出引脚使用 SOT2_2

[bit5:4] SIN2S: SIN2S 输入选择位

bit5:4	说明
读	读出寄存器值
写 00	多功能串口的ch2-SIN 的输入引脚使用 SIN2_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch2-SIN 的输入引脚使用 SIN2_1
写 11	多功能串口的ch2-SIN 的输入引脚使用 SIN2_2

[bit3:0]保留位, 读永远是 0, 写无效

12.4.7.7 扩展功能引脚设定寄存器 07(FN_SEL07)

当同一个引脚具有多个功能时，FN_SEL07 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 28	27 -- 26	25 -- 24
名称	保留位	SCS50B[1:0]	SCK5B[1:0]
初始值	0	0	0
属性	-	R/W	R/W

位	23 -- 22	21 -- 20	19 -- 16
名称	SOT5B[1:0]	SIN5S[1:0]	保留位
初始值	0	0	0
属性	R/W	R/W	-

位	15 -- 14	13 -- 12	11 -- 10	9 -- 8
名称	保留位	SCS41E[1:0]	SCS40B[1:0]	SCK4B[1:0]
初始值	0	0	0	0
属性	-	R/W	R/W	R/W

位	7 -- 6	5 -- 4	3 -- 2	1 -- 0
名称	SOT4B[1:0]	SIN4S[1:0]	CTS4S[1:0]	RTS4E[1:0]
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

寄存器功能

[bit31:28] 保留位，读永远是 0，写无效

[bit27:26] SCS50B: SCS50B 输入输出选择位

Bit27:26	说明
读	读出寄存器值
写 00	多功能串口的ch5-SCS50 的输入引脚使用 SCS50_0 不输出 (初始值)
写 01	多功能串口的ch5-SCS50 的输入引脚使用 SCS50_0 输出引脚使用 SCS50_0
写 10	多功能串口的ch5-SCS50 的输入引脚使用 SCS50_1 输出引脚使用 SCS50_1
写 11	多功能串口的ch5-SCS50 的输入引脚使用 SCS50_2 输出引脚使用 SCS50_2

[bit25:24] SCK5B: SCK5 输入输出选择位

Bit25:24	说明
读	读出寄存器值
写 00	多功能串口的ch5-SCK 的输入引脚使用 SCK5_0 不输出 (初始值)
写 01	多功能串口的ch5-SCK 的输入引脚使用 SCK5_0 输出引脚使用 SCK5_0

写 10	多功能串口的ch5-SCK 的输入引脚使用 SCK5_1 输出引脚使用 SCK5_1
写 11	多功能串口的ch5-SCK 的输入引脚使用 SCK5_2 输出引脚使用 SCK5_2

[bit23:22] SOT5B: SOT5B 输入输出选择位

bit13:12	说明
读	读出寄存器值
写 00	多功能串口的ch5-SOT 的输入引脚使用 SOT5_0 不输出 (初始值)
写 01	多功能串口的ch5-SOT 的输入引脚使用 SOT5_0 输出引脚使用 SOT5_0
写 10	多功能串口的ch5-SOT 的输入引脚使用 SOT5_1 输出引脚使用 SOT5_1
写 11	多功能串口的ch5-SOT 的输入引脚使用 SOT5_2 输出引脚使用 SOT5_2

[bit21:20] SIN5S: SIN5S 输入选择位

Bit21:20	说明
读	读出寄存器值
写 00	多功能串口的ch5-SIN 的输入引脚使用 SIN5_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch5-SIN 的输入引脚使用 SIN5_1
写 11	多功能串口的ch5-SIN 的输入引脚使用 SIN5_2

[bit19:14] 保留位，读永远是 0，写无效

[bit13:12] SCS41E: SCS41E 输出选择位

Bit13:12	说明
读	读出寄存器值
写 00	不输出多功能串口的ch4-SCS41 (初始值)
写 01	多功能串口的ch4- SCS41 的输出引脚使用 SCS41_0
写 10	多功能串口的ch4- SCS41 的输出引脚使用 SCS41_1
写 11	多功能串口的ch4- SCS41 的输出引脚使用 SCS41_2

[bit11:10] SCS40B: SCS40B 输入输出选择位

Bit11:10	说明
读	读出寄存器值
写 00	多功能串口的ch4-SCS40 的输入引脚使用 SCS40_0 不输出 (初始值)
写 01	多功能串口的ch4-SCS40 的输入引脚使用 SCS40_0 输出引脚使用 SCS40_0
写 10	多功能串口的ch4-SCS40 的输入引脚使用 SCS40_1 输出引脚使用 SCS40_1
写 11	多功能串口的ch4-SCS40 的输入引脚使用 SCS40_2 输出引脚使用 SCS40_2

[bit9:8] SCK4B: SCK4 输入输出选择位

bit9:8	说明
读	读出寄存器值

写 00	多功能串口的ch4-SCK 的输入引脚使用 SCK4_0 不输出 (初始值)
写 01	多功能串口的ch4-SCK 的输入引脚使用 SCK4_0 输出引脚使用 SCK4_0
写 10	多功能串口的ch4-SCK 的输入引脚使用 SCK4_1 输出引脚使用 SCK4_1
写 11	多功能串口的ch4-SCK 的输入引脚使用 SCK4_2 输出引脚使用 SCK4_2

[bit7:6] SOT4B: SOT4B 输入输出选择位

bit7:6	说明
读	读出寄存器值
写 00	多功能串口的ch4-SOT 的输入引脚使用 SOT4_0 不输出 (初始值)
写 01	多功能串口的ch4-SOT 的输入引脚使用 SOT4_0 输出引脚使用 SOT4_0
写 10	多功能串口的ch4-SOT 的输入引脚使用 SOT4_1 输出引脚使用 SOT4_1
写 11	多功能串口的ch4-SOT 的输入引脚使用 SOT4_2 输出引脚使用 SOT4_2

[bit5:4] SIN4S: SIN4S 输入选择位

bit5:4	说明
读	读出寄存器值
写 00	多功能串口的ch4-SIN 的输入引脚使用 SIN4_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch4-SIN 的输入引脚使用 SIN4_1
写 11	多功能串口的ch4-SIN 的输入引脚使用 SIN4_2

[bit3:2] CTS4S: CTS4S 输入选择位

Bit3:2	说明
读	读出寄存器值
写 00	多功能串口的ch4-CTS 的输入引脚使用 CTS4_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch4-CTS 的输入引脚使用 CTS4_1
写 11	多功能串口的ch4-CTS 的输入引脚使用 CTS4_2

[bit1:0] RTS4E: RTS4E 输出选择位

bit1:0	说明
读	读出寄存器值
写 00	多功能串口的ch4-RTS 不进行输出 (初始值)
写 01	多功能串口的ch4-RTS 的输出引脚使用 RTS4_0
写 10	多功能串口的ch4-RTS 的输出引脚使用 RTS4_1
写 11	多功能串口的ch4-RTS 的输出引脚使用 RTS4_2

12.4.7.8 扩展功能引脚设定寄存器 08(FN_SEL08)

当同一个引脚具有多个功能时，FN_SEL08 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 24
名称	保留位
初始值	8'h00
属性	R

位	23 -- 16
名称	保留位
初始值	8'h00
属性	R

位	15 -- 14	13 -- 12	11 -- 10	9 -- 8
名称	SCS62E[1:0]	SCS61E[1:0]	SCS60B[1:0]	SCK6B[1:0]
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

位	7 -- 6	5 -- 4	3 -- 2	1 -- 0
名称	SOT6B[1:0]	SIN6S[1:0]	CTS6S[1:0]	RTS6E[1:0]
初始值	0	0	0	0
属性	R/W	R/W	R/W	R/W

寄存器功能

[bit31:16] 保留位，读永远是 0，写无效

[bit15:14] SCS62E: SCS62E 输出选择位

Bit15:14	说明
读	读出寄存器值
写 00	不输出多功能串口的ch6- SCS62 (初始值)
写 01	多功能串口的ch6- SCS62 的输出引脚使用 SCS62_0
写 10	多功能串口的ch6- SCS62 的输出引脚使用 SCS62_1
写 11	多功能串口的ch6- SCS62 的输出引脚使用 SCS62_2

[bit13:12] SCS61E: SCS61E 输出选择位

Bit13:12	说明
读	读出寄存器值
写 00	不输出多功能串口的ch6- SCS61 (初始值)
写 01	多功能串口的ch6- SCS61 的输出引脚使用 SCS61_0
写 10	多功能串口的ch6- SCS61 的输出引脚使用 SCS61_1
写 11	多功能串口的ch6- SCS61 的输出引脚使用 SCS61_2

[bit11:10] SCS60B: SCS60B 输入输出选择位

Bit11:10	说明
读	读出寄存器值
写 00	多功能串口的ch6-SCS60 的输入引脚使用 SCS60_0 不输出 (初始值)

写 01	多功能串口的ch6-SCS60 的输入引脚使用 SCS60_0 输出引脚使用 SCS60_0
写 10	多功能串口的ch6-SCS60 的输入引脚使用 SCS60_1 输出引脚使用 SCS60_1
写 11	多功能串口的ch6-SCS60 的输入引脚使用 SCS60_2 输出引脚使用 SCS60_2

[bit9:8] SCK6B: SCK6 输入输出选择位

bit9:8	说明
读	读出寄存器值
写 00	多功能串口的ch6-SCK 的输入引脚使用 SCK6_0 不输出 (初始值)
写 01	多功能串口的ch6-SCK 的输入引脚使用 SCK6_0 输出引脚使用 SCK6_0
写 10	多功能串口的ch6-SCK 的输入引脚使用 SCK6_1 输出引脚使用 SCK6_1
写 11	多功能串口的ch6-SCK 的输入引脚使用 SCK6_2 输出引脚使用 SCK6_2

[bit7:6] SOT6B: SOT6B 输入输出选择位

bit7:6	说明
读	读出寄存器值
写 00	多功能串口的ch6-SOT 的输入引脚使用 SOT6_0 不输出 (初始值)
写 01	多功能串口的ch6-SOT 的输入引脚使用 SOT6_0 输出引脚使用 SOT6_0
写 10	多功能串口的ch6-SOT 的输入引脚使用 SOT6_1 输出引脚使用 SOT6_1
写 11	多功能串口的ch6-SOT 的输入引脚使用 SOT6_2 输出引脚使用 SOT6_2

[bit5:4] SIN6S: SIN6S 输入选择位

bit5:4	说明
读	读出寄存器值
写 00	多功能串口的ch6-SIN 的输入引脚使用 SIN6_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch6-SIN 的输入引脚使用 SIN6_1
写 11	多功能串口的ch6-SIN 的输入引脚使用 SIN6_2

[bit3:2] CTS6S: CTS6S 输入选择位

Bit3:2	说明
读	读出寄存器值
写 00	多功能串口的ch6-CTS 的输入引脚使用 CTS6_0 (初始值)
写 01	与写 00 时相同
写 10	多功能串口的ch6-CTS 的输入引脚使用 CTS6_1
写 11	多功能串口的ch6-CTS 的输入引脚使用 CTS6_2

[bit1:0] RTS6E: RTS6E 输出选择位

bit1:0	说明
读	读出寄存器值
写 00	多功能串口的ch6-RTS 不进行输出 (初始值)

写 01	多功能串口的ch6-RTS 的输出引脚使用 RTS6_0
写 10	多功能串口的ch6-RTS 的输出引脚使用 RTS6_1
写 11	多功能串口的ch6-RTS 的输出引脚使用 RTS6_2

12.4.7.9 扩展功能引脚设定寄存器 09(FN_SEL09)

当同一个引脚具有多个功能时，FN_SEL09 寄存器可为引脚配置功能。

寄存器配置

位	31 -- 24
名称	保留位
初始值	8'h00
属性	R

位	23 -- 16
名称	保留位
初始值	8'h00
属性	R

位	15 -- 14	13 -- 12	11 -- 10	9 -- 8
名称	保留位	ICRSTE[1:0]	ICIOB[1:0]	ICCKE[1:0]
初始值	0	0	0	0
属性	R	R/W	R/W	R/W

位	7 -- 6	5 -- 4	3 -- 2	1 -- 0
名称	保留位	BZOE[1:0]	保留位	VCOE[1:0]
初始值	0	0	0	0
属性	R	R/W	R	R/W

寄存器功能

[bit31:14] 保留位，读永远是 0，写无效

[bit13:12] ICRSTE: SCI7816 模块的 ICRST 信号输出选择位

Bit13:12	说明
读	读出寄存器值
写 00	不输出SCI7816模块的ICRST信号 (初始值)
写 01	SCI7816模块的ICRST输出引脚使用 ICRST_0
写 10	SCI7816模块的ICRST输出引脚使用 ICRST_1
写 11	SCI7816模块的ICRST输出引脚使用 ICRST_2

[bit11:10] ICIOB: SCI7816 模块的 ICIO 信号输入输出选择位

Bit11:10	说明
读	读出寄存器值
写 00	SCI7816模块的输入信号ICIO使用ICIO_0 不输出 (初始值)
写 01	SCI7816模块的输入信号ICIO使用ICIO_0 输出引脚使用ICIO_0
写 10	SCI7816模块的输入信号ICIO使用ICIO_1 输出引脚使用ICIO_1
写 11	SCI7816模块的输入信号ICIO使用ICIO_2 输出引脚使用 ICIO_2

[bit9:8] ICCKE: SCI7816 模块的 ICCK 信号输出选择位

Bit9:8	说明
读	读出寄存器值
写 00	不输出SCI7816模块的ICCK信号 (初始值)
写 01	SCI7816模块的ICCK输出信号使用ICCK_0
写 10	SCI7816模块的ICCK输出信号使用ICCK_1
写 11	SCI7816模块的ICCK输出信号使用ICCK_2

[bit7:6]保留位，读永远是0，写无效

[bit5:4] BZOE: 蜂鸣器模块的 BZ 信号输出选择位

Bit5:4	说明
读	读出寄存器值
写 00	不输出蜂鸣器模块的BZ信号 (初始值)
写 01	蜂鸣器模块的BZ输出信号使用BZ_0
写 10	蜂鸣器模块的BZ输出信号使用BZ_1
写 11	蜂鸣器模块的BZ输出信号使用BZ_2

[bit3:2]保留位，读永远是0，写无效

[bit1:0] VCOE: 电压比较器的 CAOUT 信号输出选择位

Bit1:0	说明
读	读出寄存器值
写 00	不输出电压比较器的CAOUT信号 (初始值)
写 01	电压比较器的CAOUT输出信号使用CAOUT_0
写 10	电压比较器的CAOUT输出信号使用CAOUT_1
写 11	电压比较器的CAOUT输出信号使用CAOUT_2

12.4.8 端口开漏功能设定寄存器(ODR)

ODR 寄存器用作设定端口是否具有开漏功能

寄存器功能

31 8	7 0	初始值	属性	支持端口
保留位	ODR0	0x0000	R/W	P07 to P00
保留位	ODR1	0x0000	R/W	P17 to P10
保留位	ODR2	0x0000	R/W	P27 to P20
保留位	ODR3	0x0000	R/W	P37--P34, P31-P30
保留位	ODR4	0x0000	R/W	P47--P45, P41--40
保留位	ODR5	0x0000	R/W	P57 to P50
保留位	ODR6	0x0000	R/W	P67 to P60
保留位	ODR7	0x0000	R/W	P74 to P70
保留位	ODR8	0x0000	R/W	P80

bit	31	8	7	0
Field	保留位			ODRx

[bit31:8]保留位，读永远是"0x0000"。写无效。

[bit7:0] ODRx: 端口开漏功能 (open-drain) 设定寄存器设定引脚的使用方法:

Bit7:0	说明
读	读出寄存器的设定值。
写 0	不将该端口设置为开漏 (open-drain) 功能。当该端口被SET寄存器或外设功能设为输出逻辑“高”时，该端口输出状态为“高”电平。
写 1	设置为开漏 (open-drain) 引脚。当端口被SET寄存器或外设功能设为输出逻辑“高”时，该端口输出状态为高阻

注意事项:

- ODRx 的"x"是通配符，指示 ODR0,ODR1, ODR2.....
- Px0 的"x"是通配符。Px0 指示 P00, P10, P20.....
- 寄存器中的各位分别设定到各引脚上。位配置和引脚顺序一一对应。例如，ODR0 的第 7 位对应 P07； ODR0 的第 6 位对应 P06； ODR0 的第 0 位对应 P00
- 若对产品上不存在引脚的位写值无效，读值不定

12.4.9 LVDIE 模拟输入设置寄存器(LVDIE)

LVDIE 寄存器用作选择 GPIO 功能或 LVDI 模拟输入功能。

寄存器配置

位	31	--	24
名称	保留位		
初始值	8'h00		
属性	R		

位	23	--	16
名称	保留位		
初始值	8'h00		
属性	R		

位	15	--	8
名称	保留位		
初始值	8'h00		
属性	R		

位	7	--	1	0
名称	保留位			
初始值	2'b00000000			
属性	R			

寄存器功能

[bit31:1]保留位，读永远是 0，写无效

[bit0] LVDIE: LVDI 模拟输入设置寄存器

bit0	说明
读	读出寄存器值
写 0	引脚用作数字 I/O
写 1	引脚用作模拟输入 (I/O 单元处于输入/输出断开、上拉断开状态)

12.4.10 特殊端口设定寄存器(SPS)

SPS 寄存器用作将端口设置为特殊功能的引脚

寄存器配置

位	31 -- 24
名称	保留位
初始值	8'h00
属性	R

位	23 -- 16
名称	保留位
初始值	8'h00
属性	R

位	15 -- 8
名称	保留位
初始值	8'h00
属性	R

位	7 -- 4	3 -- 2	1 -- 0
名称	保留位	MAINCS	SUBCS
初始值	2'b0000	01	01
属性	R	R/W	R/W

寄存器功能

[bit31:4] 保留位, 读永远是 0, 写无效

[bit3:2] MAINCS: XHI / XHO 功能设置选择位

Bit3:2	说明
读	读出寄存器值
写 00	将XHI与XHO设置为数字IO, 而不是高速时钟(晶振)引脚
写 01	将XHI与XHO设置为高速时钟(晶振)引脚(初始值)
写 10	禁止设置
写 11	将XHI设置为外部时钟输入引脚, 将XHO设置为数字IO引脚

[bit1:0] SUBCS: XLI / XLO 功能设置选择位

Bit1:0	说明
读	读出寄存器值
写 00	将XLI与XLO设置为数字IO, 而不是低速时钟(晶振)引脚
写 01	将XLI与XLO设置为低速时钟(晶振)引脚(初始值)
写 10	禁止设置
写 11	将XLI设置为外部时钟输入引脚, 将XLO设置为数字IO引脚

12.5 注意事项

本节介绍 I/O 端口的使用注意事项。

- LPM_CTL.SPLV 控制上拉阻抗的 ON/OFF

SPLV 信号在待机模式时可把引脚切换到 Hi-Z 状态。

 - SPLV=0 时正常操作

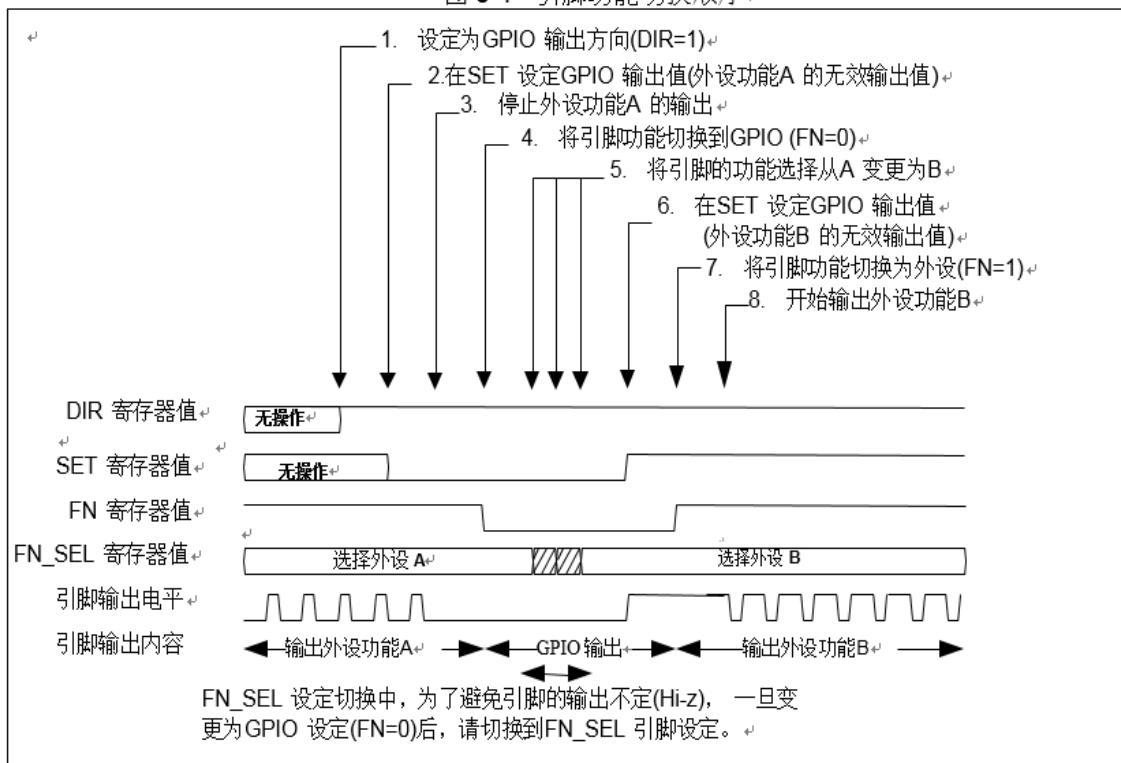
 - SPLV=1 时引脚为 Hi-Z 状态、输入切断、上拉断开。但是，SPLV 位不可用于设定外部中断、NMIX、SWD 的引脚。

关于 SPLV 位，详情参见“低功耗模式”一章中的 LPM_CTL 寄存器。

- 引脚功能切换顺序

使用 FN_SEL 寄存器进行外设功能的输出切换时，为防止引脚输出不定(Hi-Z)，请按照下图 12.5-1 中的切换顺序示例进行切换设定。

图 5-1 引脚功能切换顺序



- 保留位

保留位读为"0"。写入时，始终写"0"。

- 多功能通讯串口引脚组

当多功能通讯串口有多个输入输出时，每个输入输出需设定到同一组的端口上。“同一组的端口”是指引脚名后附的重定位功能号码相同，如"xxx_0"或"yyy_1"。

表 12.5-1 是设定示例。

表 12.5-1 多功能通讯串口设定例

串行数据输出	串行时钟输入/输出	串行数据输入	有效端口
SOT1_0 (端口 0)	SCK1_0 (端口 0)	SIN1_0 (端口 0) SIN1_1 (端口 1)	端口 0
	SCK1_1 (端口 1)	SIN1_0 (端口 0) SIN1_1 (端口 1)	禁止设定
SOT1_1 (端口 1)	SCK1_0 (端口 0)	SIN1_0 (端口 0) SIN1_1 (端口 1)	
	SCK1_1 (端口 1)	SIN1_0 (端口 0) SIN1_1 (端口 1)	

- 外设功能的输出

因为外设功能的输出引脚仅由 FN_SEL 设定决定，所以相同的外设功能输出不可配置到不同的引脚上。

(禁止示例): 配置 SOUT1_0 和 SOUT1_1 同时输出多功能通讯串口通道 1 的结果。

- 引脚设定和工作模式

关于 SWD 设定，参见“调试接口”一章。关于待机模式时和复位时的各引脚状态，参见“数据手册”。

- 产品规格和外设功能的引脚配置

配置到各引脚上的功能(GPIO、外设输入输出、特殊 I/O)因型号而异。关于各型号的引脚功能，请使用所用型号的数据手册中的引脚功能表确认。对于不存在匹配型号的引脚，严禁使用 FN_SEL 寄存器进行功能选择。

- Standby 模式下外部中断引脚的设置

在设置 SPLV=1 的条件下，芯片进入 standby 模式时，设置 FN=1 并选择外设功能，以使能外部中断分配引脚进行唤醒。

如果用于外部中断的引脚设置保持为 FN=0，将出现误操作。

13. 快速 GPIO

本章介绍快速 GPIO 端口。

13.1 概要

本节介绍快速 GPIO 的概要。

周期访问

- 可以在一个周期内访问的 GPIO 称为快速 GPIO。
- 快速 GPIO 可以在一个 HCLK 时钟周期内读取一个输入电平或者设置一个来自 CP 的输出电平。
- 初始化设置为 GPIO，需要通过寄存器设置选择快速 GPIO。

引脚分配

所有 I/O 端口均可作为 GPIO 或者快速 GPIO。

位操作

快速 GPIO 有专用电路支持位操作。

13.2 配置

本节介绍快速 GPIO 的配置、框图和功能。

快速 GPIO 的配置

图 13.2-1 显示快速 GPIO 的图框。

I/O 端口通过设置寄存器 FSETE 切换普通 GPIO 和快速 GPIO。

PUx、DIRx、ANALOG 的设置与普通 GPIO 相同。

图 13.2-1 快速 GPIO 框图

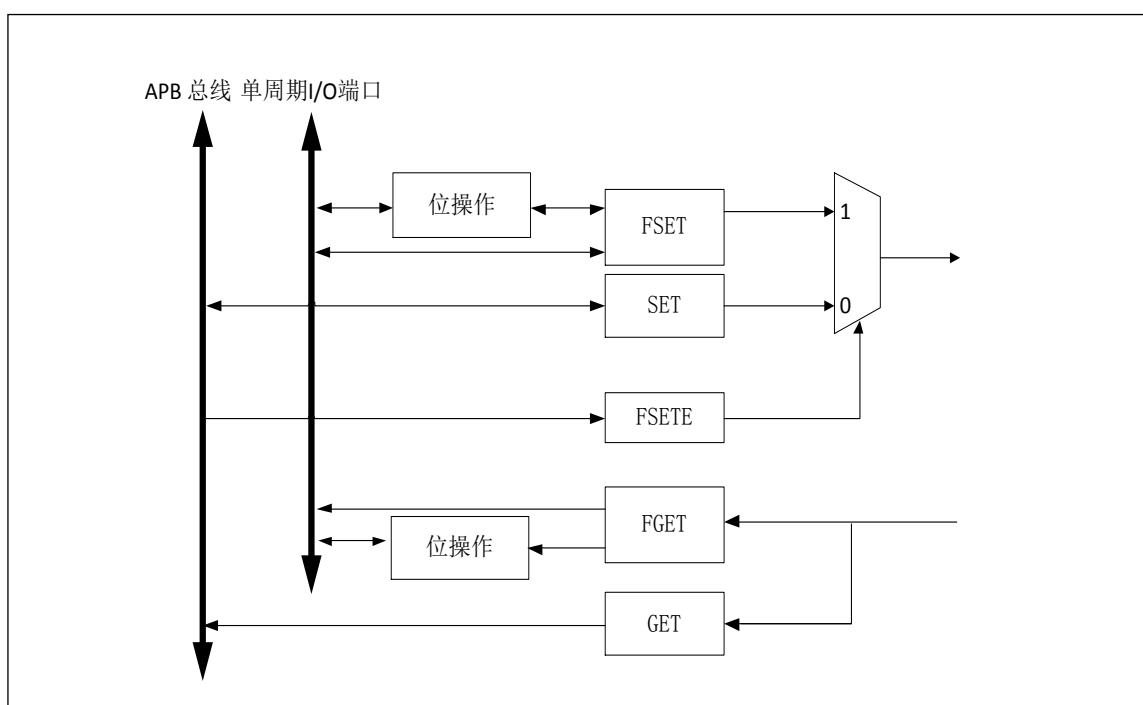


表 13.2-1 寄存器功能

寄存器名	功能
FGET	该寄存器读取快速GPIO端口的状态。 用法与普通GPIO的GETx相同，区别在于地址和只有一个周期延迟。 如果快速GPIO端口作为数字输入引脚，读取输入电平。 如果快速GPIO端口作为数字输出引脚，读取输出电平。 如果快速GPIO端口作为特殊端口，读取值总是“0”。
FSET	该寄存器在快速GPIO作为输出时设置输出电平。 用法与普通GPIO相同，区别在于地址和只有一个周期延迟。 当设置为“0”，则输出低电平。 当设置为“1”，则输出高电平。 注意事项： 如果引脚是输入状态或者普通GPIO或者外设的输入输出，那么设置无效。

寄存器名	功能
FSETE	该寄存器选择普通GPIO输出还是快速GPIO输出。 设置为“0”时，选择普通GPIO输出。 设置为“1”时，选择快速GPIO输出。 该寄存器设置不影响FGET。

快速 GPIO 访问

- 引脚设置为快速 GPIO 以后，可以使用 FGET 读取状态，也可使用 FSET 设置输出电平。用法与 GET/SET 相同。
- FGET/GET 和 FSET/SET 的位之间是一一对应的。

例如：

当 P00 设置为输入状态，访问 FGET0 的 bit0 读取 P00 的输入值。

当 P15 设置为快速 GPIO 并且为输出状态，则访问 FSET1 的 bit5 去设置 P15 的输出值。

位操作

- 概要

快速 GPIO 搭载在单周期 I/O 端口上。单周期 I/O 端口可以在一个周期访问数据，但是不支持位带操作。

快速 GPIO 通过特殊电路可以支持位操作。快速 GPIO 的每一位可以独立设置/清除。与位带操作类似，位操作也有额外的别名地址区域。

位操作只支持访问 FGET 和 FSET。

- 使用方法

这里介绍位操作的地址。

- 别名地址

别名地址是位操作的访问地址。其计算公式如下：

别名地址 = 基地址 + 偏移地址

- 基地址

基地址是 FGET、FSET 的具体地址，请参考第 5 节“位操作基地址”。

- 偏移地址

偏移地址是访问位的地址

表 13.2-2 位操作的偏移地址

访问位	偏移地址
bit0	0x0004
bit1	0x0008
bit2	0x0010
bit3	0x0020
bit4	0x0040
bit5	0x0080
bit6	0x0100
bit7	0x0200
禁止	其他

读取 FGET 仅读取“访问位”，其他位为“0”。例如，当 FGET0 为“0x07”时，“FGET0 基址 + 0x0010”的读取值为“0x04”。

写 FSET 只作用于“访问位”，忽略其他位。例如，当 FSET1 为“0x00”并写“0xFF”到“FSET1 基地址 + 0x0008”，FSET1 的值则变为“0x02”。

13.3 操作

本节介绍设置快速 GPIO 的操作示例。

设置快速 GPIO

图 13.3-1 显示输出设置的操作示例。

图 13.3-2 显示输入设置的操作示例。

图 13.3-1 快速 GPIO 输出设置操作示例

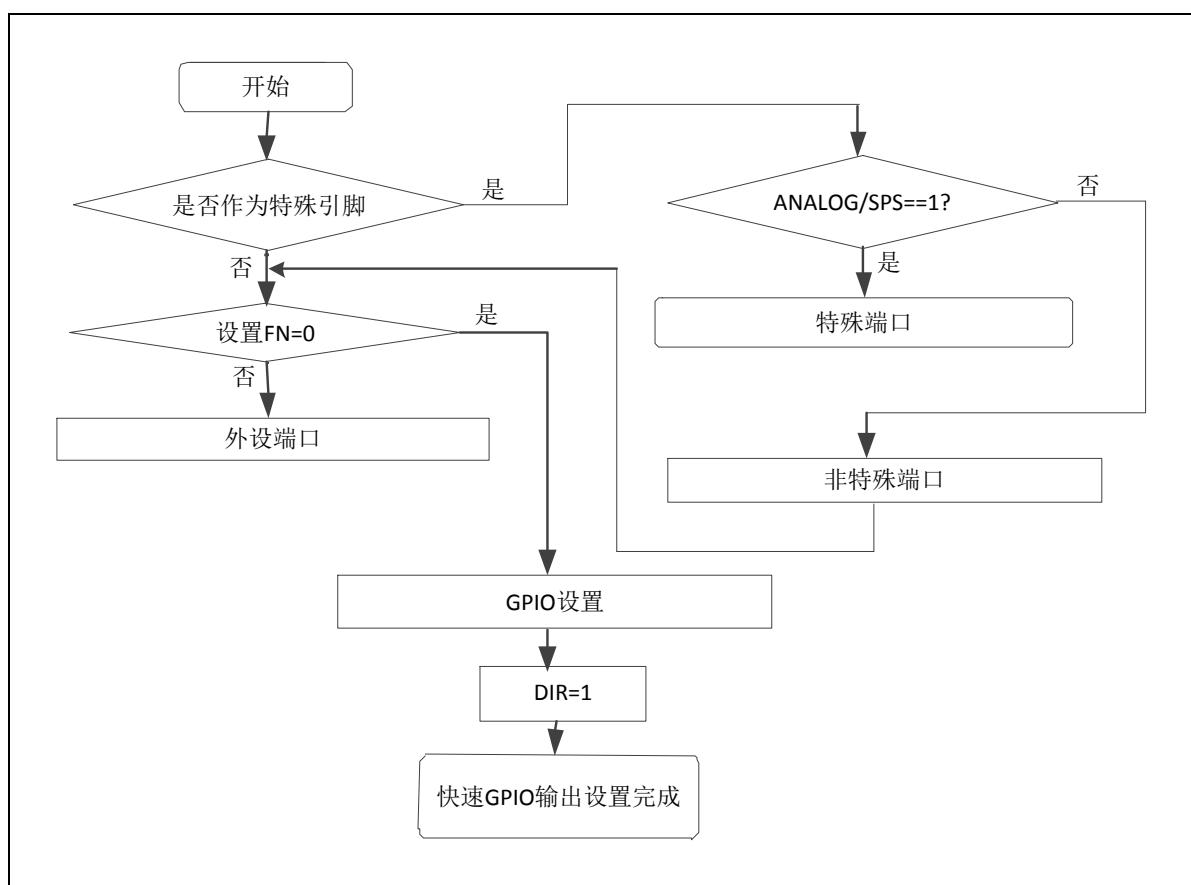
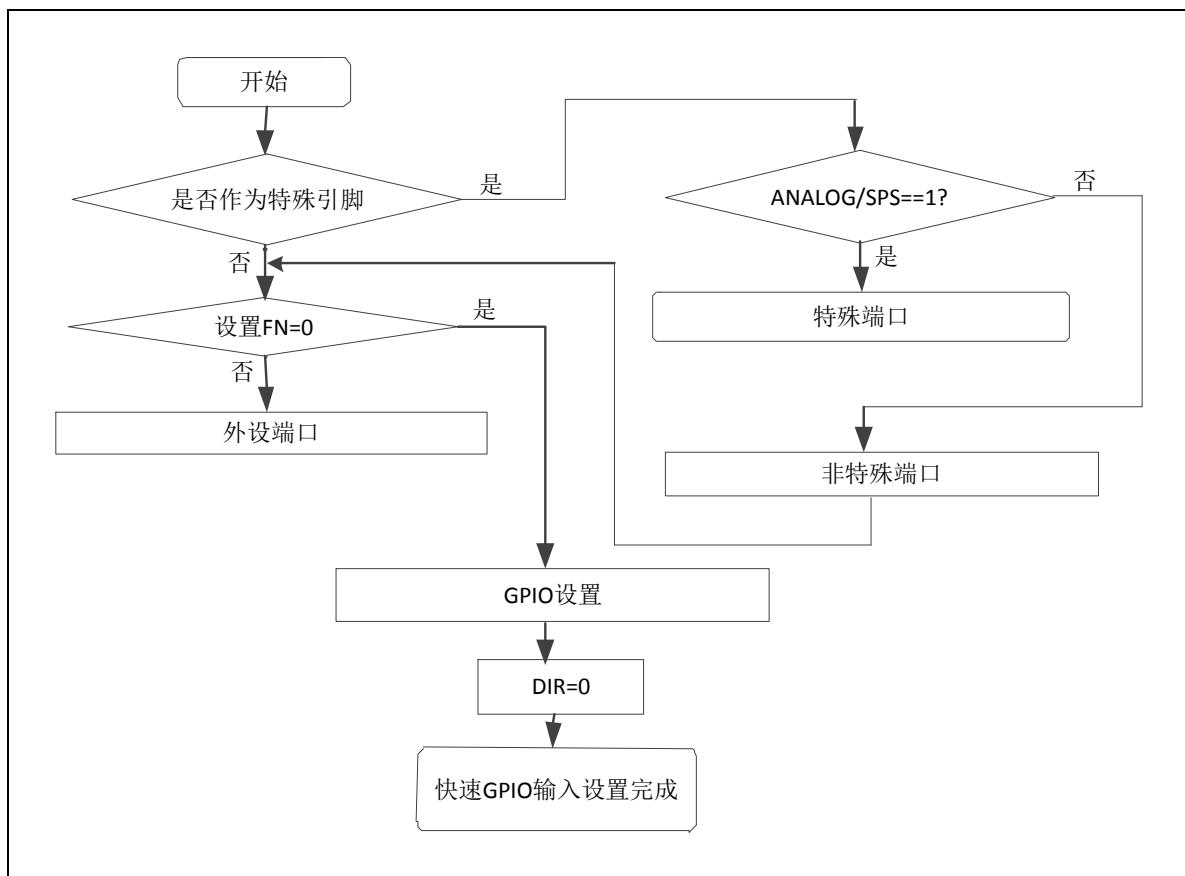


图 13.3-2 快速 GPIO 输入设置操作示例



13.4 寄存器

本节描述快速 GPIO 端口的寄存器。

表 13.4-1 快速 GPIO 寄存器列表

缩写	寄存器名	参考
FGET0	快速GPIO输入数据寄存器0	13.4.1
FGET1	快速GPIO输入数据寄存器1	
FGET2	快速GPIO输入数据寄存器2	
FGET3	快速GPIO输入数据寄存器3	
FGET4	快速GPIO输入数据寄存器4	
FGET5	快速GPIO输入数据寄存器5	
FGET6	快速GPIO输入数据寄存器6	
FGET7	快速GPIO输入数据寄存器7	
FGET8	快速GPIO输入数据寄存器8	
FSET0	快速GPIO输出数据寄存器0	13.4.2
FSET1	快速GPIO输出数据寄存器1	
FSET2	快速GPIO输出数据寄存器2	
FSET3	快速GPIO输出数据寄存器3	
FSET4	快速GPIO输出数据寄存器4	
FSET5	快速GPIO输出数据寄存器5	
FSET6	快速GPIO输出数据寄存器6	
FSET7	快速GPIO输出数据寄存器7	
FSET8	快速GPIO输出数据寄存器8	
FSETE0	快速GPIO输出数据使能寄存器0	13.4.3
FSETE1	快速GPIO输出数据使能寄存器1	
FSETE2	快速GPIO输出数据使能寄存器2	
FSETE3	快速GPIO输出数据使能寄存器3	
FSETE4	快速GPIO输出数据使能寄存器4	
FSETE5	快速GPIO输出数据使能寄存器5	
FSETE6	快速GPIO输出数据使能寄存器6	
FSETE7	快速GPIO输出数据使能寄存器7	
FSETE8	快速GPIO输出数据使能寄存器8	

13.4.1 快速 GPIO 输入数据寄存器 (FGETx)

快速 GPIO 输入数据寄存器 (FGETx) 显示引脚输入数据。

寄存器配置

bit31	8	7	0	初始值	属性	端口
Reserved		FGET0	0xXX	R	P07~P00	
Reserved		FGET1	0xXX	R	P17~P10	
Reserved		FGET2	0xXX	R	P27~P20	
Reserved		FGET3	0xXX	R	P37~P30	
Reserved		FGET4	0xXX	R	P47~P40	
Reserved		FGET5	0xXX	R	P57~P50	
Reserved		FGET6	0xXX	R	P67~P60	
Reserved		FGET7	0xXX	R	P77~P70	
Reserved		FGET8	0xXX	R	P87~P80	

寄存器功能

[bit31:8]Reserved: 保留位，保留位读取值为“0”。

[bit7:0]FGETx: 快速 GPIO 输入数据寄存器 x

读取快速 GPIO 输入数据

bit	描述
0	不论其他引脚功能如何设置 (FN/FN_SEL/DIR/FSET)，均表示引脚状态为低电平输入或者低电平输出。当ANALOG/SPS配置引脚为特殊引脚时，输入已切断，因此读取值为“0”。
1	不论其他引脚功能如何设置 (FN/FN_SEL/DIR/FSET)，均表示引脚状态为高电平输入或者高电平输出。
写	无效

注意：

- FGETx 中的 “x” 是一个通配符。FGETx 表示 FGET0、FGET1、FGET2 等。
- Px0 中的 “x” 也是一个通配符。Px0 表示 P00、P10、P20 等。
- 一个寄存器允许读取从 Px0 到 Px7 这 8 个端口的输入数据。
- 寄存器的每一位与引脚顺序一一对应。例如，FGET0 的第 7 位对应 P07，第 6 位对应 P06，第 0 位对应 P00。
- 产品中不存在的引脚对应位的读取值为“0”。

13.4.2 快速 GPIO 输出数据寄存器 (FSETx)

快速 GPIO 输出数据寄存器 (FSETx) 显示引脚输出数据。

寄存器配置

bit31	8	7	0	初始值	属性	端口
Reserved		FSET0		0x00	R/W	P07~P00
Reserved		FSET1		0x00	R/W	P17~P10
Reserved		FSET2		0x00	R/W	P27~P20
Reserved		FSET3		0x00	R/W	P37~P30
Reserved		FSET4		0x00	R/W	P47~P40
Reserved		FSET5		0x00	R/W	P57~P50
Reserved		FSET6		0xXX	R/W	P67~P60
Reserved		FSET7		0x00	R/W	P77~P70
Reserved		FSET8		0x00	R/W	P87~P80

寄存器功能

[bit31:8]Reserved: 保留位，保留位读取值为“0”。

[bit7:0]FSETx: 快速 GPIO 输出数据寄存器 x

设置快速 GPIO 输出数据

bit	描述
0	输出低电平到快速GPIO。 如果引脚配置为I/O输入，普通GPIO或者外设输入输出，设置则无效。
1	输出高电平到快速GPIO。 如果引脚配置为I/O输入，普通GPIO或者外设输入输出，设置则无效。

注意：

- FSETx 中的 “x” 是一个通配符。FSETx 表示 FSET0、FSET1、FSET2 等。
- Px0 中的 “x” 也是一个通配符。Px0 表示 P00、P10、P20 等..
- 一个寄存器允许读取从 Px0 到 Px7 这 8 个端口的输出数据。
- 寄存器的每一位与引脚顺序一一对应。例如，FSET0 的第 7 位对应 P07，第 6 位对应 P06，第 0 位对应 P00。
- 对于产品中不存在的引脚，相应位写无效，读取值没有定义。

13.4.3 快速 GPIO 输出使能寄存器 (FSETEx)

快速 GPIO 输出使能寄存器 (FSETEx) 选择端口输出为普通 GPIO 或者快速 GPIO。

寄存器配置

bit31	8	7	0	初始值	属性	端口
Reserved		FSETE0	0x00	R/W	P07~P00	
Reserved		FSETE1	0x00	R/W	P17~P10	
Reserved		FSETE2	0x00	R/W	P27~P20	
Reserved		FSETE3	0x00	R/W	P37~P30	
Reserved		FSETE4	0x00	R/W	P47~P40	
Reserved		FSETE5	0x00	R/W	P57~P50	
Reserved		FSETE6	0xXX	R/W	P67~P60	
Reserved		FSETE7	0x00	R/W	P77~P70	
Reserved		FSETE8	0x00	R/W	P87~P80	

寄存器功能

[bit31:8]Reserved: 保留位，保留位读取值为“0”。

[bit7:0]FSETEx: 快速 GPIO 输出使能寄存器

设置 I/O 端口输出来自普通 GPIO 还是快速 GPIO

bit	描述
0	选择SET输出到引脚。 如果引脚配置为I/O输入或者外设输入输出，设置则无效。
1	选择FSET输出到引脚。 如果引脚配置为I/O输入或者外设输入输出，设置则无效。

注意：

- FSETEx 寄存器搭载在 APB 总线上。
- FSETEx 中的 “x” 是一个通配符。FSETEx 表示 FPDOE0、FPDOE1、FPDOE2 等。
- Px0 中的 “x” 也是一个通配符。Px0 表示 P00、P10、P20 等。
- 寄存器的每一位与引脚顺序一一对应。例如，FSETE0 的第 7 位对应 P07，第 6 位对应 P06，第 0 位对应 P00。
- 不论 FSETEx 的值是什么，引脚值始终可以通过 FGETx 读取。
- 对于产品中不存在的引脚，相应位写无效，读取值没有定义。

13.5 位操作基地址

本节描述 FGETx 和 FSETx 位操作的基地址。

基地址列表

表 13.5-1 显示位操作的基地址

表 13.5-1 别名地址列表

名称	地址
FGET0 基地址	0xF801_0000
FGET1 基地址	0xF801_0400
FGET2 基地址	0xF801_0800
FGET3 基地址	0xF801_0C00
FGET4 基地址	0xF801_1000
FGET5 基地址	0xF801_1400
FGET6 基地址	0xF801_1800
FGET7 基地址	0xF801_1C00
FGET8 基地址	0xF801_2000
FSET0 基地址	0xF801_4000
FSET1 基地址	0xF801_4400
FSET2 基地址	0xF801_4800
FSET3 基地址	0xF801_4C00
FSET4 基地址	0xF801_5000
FSET5 基地址	0xF801_5400
FSET6 基地址	0xF801_5800
FSET7 基地址	0xF801_5C00
FSET8 基地址	0xF801_6000

14. AES 硬件加速器

14.1 概述

AES 硬件加速器可以执行 AES 算法标准的加密流程和解密流程，其执行结果完全符合“FIPS PUB 197”对算法原理的描述。

可提供一个安全的硬件实现架构，能够有效抵御 Side Channel Attack 中的差异功耗分析（DPA）和差异错误分析（DFA）的攻击。

具有两个功能：加密、标准解密。

14.2 操作

14.2.1 操作共同点

- 1) 上电后，首先必须通过 AHB 总线复位 hresetn 对本模块进行一次异步复位操作。时钟 hclk 必须在复位脱离前稳定有效，并且在后续运行中持续稳定。
- 2) 在 AES 加解密过程中，数据寄存器会改变，如果下次运算操作的数据就是本次运算的结果，那么就无需重新写入数据了。
- 3) 在用相同的密钥加解密大量数据的情况下，不需要重复写入密钥。
- 4) 为了有效抵御差异功耗分析，在每次异步复位后必须输入新的随机数，并建议在每次启动本模块运算前都输入新的随机数，输入的 64 位随机数不能为全 0。当然，即使不输入新的随机数也不会影响模块本身运算的正确性，只是可能会降低抵御差异功耗分析的能力。
- 5) 密钥写入分三种情况：如果密钥是 128bits，将密钥写入地址 0x4006_B020-0x4006_B02C；如果密钥是 192bits，将密钥写入地址 0x4006_B020-0x4006_B034；如果密钥是 256bits，将密钥写入地址 0x4006_B020-0x4006_B03C。
- 6) 判断模块运算结束的方法：不断读取 AES_CON.START，如果其值变为 0，则表示运算结束。
- 7) AES 最大工作频率为 20MHz，所以当 AHB 总线时钟速度高于 20MHz 时，请将 AES_DIV_EN.DIV_EN 设置为 1（HCLK 2 分频）以保证该模块可以正常工作。

14.2.2 加密流程

- 1) 将 64bits 随机数写入 AES_RNG 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的

第一次操作必须写入新的随机数，其余场合可以选择省略本步骤)。

- 2) 根据应用需求将加密密钥写入密钥寄存器(AES_KEY)中，除非复位或者重新写入，否则 AES_KEY 中的值将被保持。如果多次运算的密钥相同，则在第一次写入后即可忽略本步骤。
- 3) 将 128bits 明文写入数据寄存器(AES_DATA)中。
- 4) 设置 AES_CON 中的各位，包括：
 - a) 根据需要设置控制寄存器中的 AES_CON.KEY_SEL；
 - b) 将 AES_CON.FUNC_SEL 设置为 2'b00；
 - c) 根据需要设置 AES_CON.DUMMY_SEL；
 - d) 向控制寄存器中的 AES_CON.START 写入 1，启动模块进行运算；
以上的步骤 a, b, c, d 可同时进行。
- 5) 判断模块运算是否结束。
不断读取 AES_CON.START，如果其值变为 0，则表示运算结束。
- 6) 读取数据寄存器(AES_DATA)，获得 128 位密文。
- 7) 否则读取 AES_CON.DFA_STA，若 AES_CON.DFA_STA=0，则转到 8)，若 AES_CON.DFA_STA=1，则进行错误处理。
- 8) 如果要继续进行新的运算，回到步骤 1)，否则结束。

14.2.3 标准解密流程

- 1) 将 64bits 随机数写入 AES_RNG 中（在有抗 DPA 要求的应用场合下，每次本模块复位后的第一次操作必须写入新的随机数，其余场合可以选择省略本步骤)。
- 2) 根据应用需求将解密密钥写入密钥寄存器(AES_KEY)中，除非复位或者重新写入，否则 AES_KEY 中的值将被保持。如果多次运算的密钥相同，则在第一次写入后即可忽略本步骤。
- 3) 将 128bits 密文写入数据寄存器(AES_DATA)中。
- 4) 设置 AES_CON 中的各位，包括：
 - a) 根据需要设置控制寄存器中的 AES_CON.KEY_SEL；
 - b) 将 AES_CON.FUNC_SEL 设置为 2' b01；；
 - c) 根据需要设置 AES_CON.DUMMY_SEL；
 - d) 向控制寄存器中的 AES_CON.START 写入 1，启动模块进行运算
以上的步骤 a, b, c, d 可同时进行。
- 5) 判断模块运算是否结束。

不断读取 AES_CON.START，如果其值变为 0，则表示运算结束。

- 6) 读取数据寄存器(AES_DATA)，获得 128 位运算结果。
- 7) 否则读取 AES_CON.DFA_STA，若 AES_CON.DFA_STA=0，则转到 8)，若 AES_CON.DFA_STA=1，则进行错误处理。
- 8) 如果要继续进行新的运算，回到步骤 1)，否则结束。

14.3 寄存器

本章介绍 AES 硬件加速器相关的寄存器。

AES 寄存器列表

缩写	寄存器名称	章节
AES_CON	控制寄存器	14.3.1
AES_DATA	数据寄存器	14.3.2
AES_KEY	密钥寄存器	14.3.3
AES_RNG	随机数寄存器	14.3.4
AES_DIV_EN	时钟分频寄存器	14.3.5

14.3.1 AES 控制寄存器 (AES_CON)

控制寄存器用来配置 AES 硬件加速器的操作模式、密钥长度和启动等。

寄存器配置

Bit	31:13	12:11	10:9	8	7:5	4:3	2:1	0
Filed	Reserved	DUMMY_SEL	Reserved	DFA_STA	Reserved	KEY_SEL	FUNC_SEL	START
R/W	-	R/W	-	R	-	R/W	R/W	R/W
Default	-	11	-	0	-	10	00	0

寄存器功能

Bit[31:13] 保留位，写操作无效，读数据为未定义。

Bit[12:11] DUMMY_SEL: 假运算轮数选择

Bit12:11		描述
写	00	7轮 (56cycles)
	01	15轮 (120cycle)
	10	31轮 (248cycle)
	11	63轮 (504cycle)
读		读取当前值

Bit[10:9] 保留位，写操作无效，读数据为未定义。

Bit[8] DFA_STA: DFA 状态

Bit10		描述
读	0	DFA检测成功
	1	DFA检测失败

Bit[7:5] 保留位，写操作无效，读数据为未定义。

Bit[4:3] KEY_SEL: 密钥选择

Bit4:3		描述
写	00	选择128比特密钥
	01	选择192比特密钥
	10	选择256比特密钥
	11	选择128比特密钥
读		读取当前值

Bit[2:1] FUNC_SEL: 功能选择

Bit2:1		描述
写	00	加密运算
	01/10/11	解密运算
读		读取当前值

Bit[0] START: 启动/结束

Bit0*1		描述
写	0	本模块运算结束或未被启动
	1	启动本模块进行运算
读		读取当前值

*1: 本寄存器中的启动 (start) 位的操作方法是: 软件对本位写入 1 后, 本模块将启动运行, 本次运行结束后本模块硬件会自动将本位清 0, 软件查询到本位为 0 即表示本次运行完成。

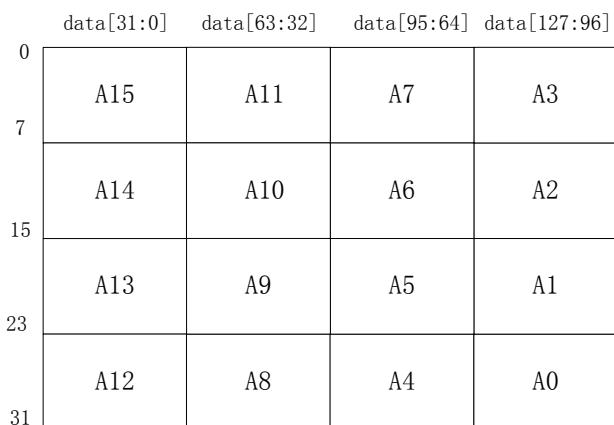
14.3.2 AES 数据寄存器 (AES_DATA)

数据寄存器由四个 32 位的寄存器组成 128 位数据，用于在模块运算前存放需要被加密的明文或者需要被解密的密文，并且，运算完成后存放加密后的密文或者解密后的明文。

加密运算		解密运算	
运算前	运算后	运算前	运算后
128位明文	128位密文	128位密文	128位明文

四个 32 位寄存器连接在一起组成一个 128 位的数据，读写操作时需要分别对四个寄存器进行操作。数据寄存器对应的操作顺序如下：

数据举例：128' hA15_A14_A13_A12_A11_A10_A9_A8_A7_A6_A5_A4_A3_A2_A1_A0



每个 32 位的寄存器成为了矩阵的一列，每一列由 4 个字节组成。

运算结束时 DFA 检测到错误，则 AES_DATA 被清为全 0；而检测成功，则 AES_DATA 不受影响。

寄存器配置

Bit	31:0
Filed	AES_DATA0
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_DATA1
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_DATA2
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_DATA3
R/W	R/W
Default	00000000000000000000000000000000

寄存器功能

Bit[31:0] AES_DATA0: AES 运算数据[31:0]

Bit[31:0] AES_DATA1: AES 运算数据[63:32]

Bit[31:0] AES_DATA2: AES 运算数据[95:64]

Bit[31:0] AES_DATA3: AES 运算数据[127:96]

写入: 需要运算的明文 (加密运算) 或密文 (解密运算);

读出: 本寄存器带极性存储, 读出的极性取决于存储时的极性 (运算过程中会反复存取, 因此启动运算之后的极性由最后一次运算存储时的极性决定)。

14.3.3 AES 密钥寄存器 (AES_KEY)

密钥寄存器由八个 32 位的寄存器组成，存放输入的初始密钥。当密钥为 128 位，输入的密钥只占用 key[127:0]；当密钥为 192 位，输入的密钥只占用 key[191:0]；当密钥为 256 位，输入的密钥将占用全部 key[255:0]。

写操作时需要分别对八个 32 位的寄存器进行操作。对应的操作顺序如下：

数据举例：

```
128' hK31_K30_K29_K28_K27_K26_K25_K24_K23_K22_K21_K20_K19
_K18_K17_K16_K15_K14_K13_K12_K11_K10_K9_K8_K7_K6_K5_K4_K3_K2_K1_K0
```

	key[31:0]	key[63:32]	key[95:64]	key[127:96]	key[159:128]	key[191:160]	key[223:192]	key[255:224]
0	K31	K27	K23	K19	K15	K11	K7	K3
7	K30	K26	K22	K18	K14	K10	K6	K2
15	K29	K25	K21	K17	K13	K9	K5	K1
23	K28	K24	K20	K16	K12	K8	K4	K0
31								

每个 32 位的寄存器成为了矩阵的一列，每一列由 4 个字节组成。

寄存器配置

Bit	31:0
Filed	AES_KEY0
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY1
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY2
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY3
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY4
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY5
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY6
R/W	R/W
Default	00000000000000000000000000000000

Bit	31:0
Filed	AES_KEY7
R/W	R/W
Default	00000000000000000000000000000000

寄存器功能

Bit[31:0] AES_KEY0: AES 密钥[31:0]

Bit[31:0] AES_KEY1: AES 密钥[63:32]

Bit[31:0] AES_KEY2: AES 密钥[95:64]

Bit[31:0] AES_KEY3: AES 密钥[127:96]

Bit[31:0] AES_KEY4: AES 密钥[159:128]

Bit[31:0] AES_KEY5: AES 密钥[191:160]

Bit[31:0] AES_KEY6: AES 密钥[223:192]

Bit[31:0] AES_KEY7: AES 密钥[255:224]

写入：用于运算的初始密钥；

读出：本寄存器带极性存储，读出的极性取决于存储时的极性。

14.3.4 AES 随机数寄存器 (AES_RNG)

本寄存器由 2 个 32 位寄存器组成 64 位数据，用于存放用于运算的随机数

寄存器配置

Bit	31:0
Filed	AES_RNG0
R/W	R/W (只能异或读写*2)
Default	XXXX0110XXXX0011XXXX1000XXXX1101*1

Bit	31:0
Filed	AES_RNG1
R/W	R/W (只能异或读写*2)
Default	XXXX0101XXXX1010XXXX1011XXXX0100*1

*1: X 表示不受复位信号影响

*2: 异或写入是指写数据与 AES_RNG 中对应的 bit 异或后再写入；异或读出是指输出数据只有最低 8 位有值，其余位为 0，且该 8 位数据为 AES_RNG[63:56] 与 AES_RNG [7:0] 异或的结果

寄存器功能

Bit[31:0] AES_RNG0: AES 运算随机数[31:0]

Bit[31:0] AES_RNG1: AES 运算随机数[63:32]

写入：运算使用的随机数；

读出：本寄存器带极性存储，读出的极性取决于存储时的极性（运算过程中会反复存取，因此启动运算之后的极性由最后一次运算存储时的极性决定）。

注意：

- 上电复位后必须重载真随机数，并建议在每次启动运算前都输入新的随机数，输入的 64 位随机数不能为全。

14.3.5 AES 时钟分频寄存器 (AES_DIV_EN)

本寄存器设置 AES 工作时钟的分频系数。

寄存器配置

Bit	31:1	0
Filed	Reserved	DIV_EN
R/W	-	R/W
Default	-	0

寄存器功能

Bit[31:1] 保留位，写操作无效，读数据为未定义。

Bit[0] DIV_EN: 分频使能

Bit0		描述
写	0	HCLK不分频
	1	HCLK 2分频
读		读取当前值

注意：

- 默认不分频，当 HCLK 高于 20MHz 时，请将 DIV_EN 设置为 1，AES 只能工作在 20MHz 以内。

14.4 启停时序

本章介绍 AES 硬件加速器的启停时序。

运行时间如下表所示：

密钥长度 (bit)	128	192	256
加密时间 (cycle)	164	196	228
解密时间 (cycle)	204	242	280

注意：

- 上表没计算 DUMMY_SEL 对应 cycle 数，需要根据 DUMMY_SEL 的设置加上对应的 cycle 数。

15. 调试接口

15.1 概要

本系列具有串行线调试端口(SW-DP)，将 ICE 连接到 SW-DP，便可进行系统调试。

另外，该系列还内置微追踪缓存(MTB)可用于记录程序流程中的变更。

关于 SW-DP 及系统调试，详情参照"Cortex -M0+技术参考手册"。

特征

为 SW-DP 配置了 2 个引脚。

2 个引脚的初始功能是串行线调试模式。

15.2 引脚说明

以下介绍引脚：

- 调试引脚
- 引脚初始功能
- SW-DP 引脚的内部上拉

15.2.1 调试引脚

串行线被配置了 2 个引脚 (SWCLK 和 SWDIO)。

表 15.2.1-1 列出了引脚功能。

表 15.2.1-1 调试模式时的 SW-DP 引脚功能一览表

引脚	功能
SWCLK	串行线时钟信号
SWDIO	串行线数据输入输出信号

15.2.2 引脚的初始功能

2 个 SW-DP 引脚也可以被用作 GPIO，但初始状态是调试功能。

注意：

- 关于调试功能的设置，详情请参考 "I/O 端口"一章。

表 15.2.2-1 指示复位解除后的初始状态和通过 FN 寄存器（端口功能设定寄存器）的设定可变更的功能。

注意：

- 关于 FN 寄存器，详情请参考“I/O 端口”一章。

表 15.2.2-1 调试引脚的初始功能和功能变更一览表

	引脚名称	初始状态的引脚功能	由FN寄存器设定变更的功能
SW-DP引脚	SWCLK	SWCLK	GPIO
	SWDIO	SWDIO	GPIO

15.2.3 SW-DP 引脚的内部上拉

本系列提供调试引脚的上拉功能。

用户可以通过设置 GPIO 的寄存器来控制上拉功能。

表 15.2.3-1 SW-DP 引脚的内部上拉使能/禁止一览表

	使能调试引脚时的上拉功能*
SWCLK	使能
SWDIO	使能

*：复位时上拉也是使能状态。

16. 微追踪缓存数据观察点和跟踪单元

(MTB_DWT)

16.1 概要

MTB_DWT 的功能是产生 ARM CoreSight MTB 的停止和起始信号来监控处于数据访问时的处理器地址和数据总线。同时，可以在 MTB 的记录中配置并监测观察点用来触发适当的响应。

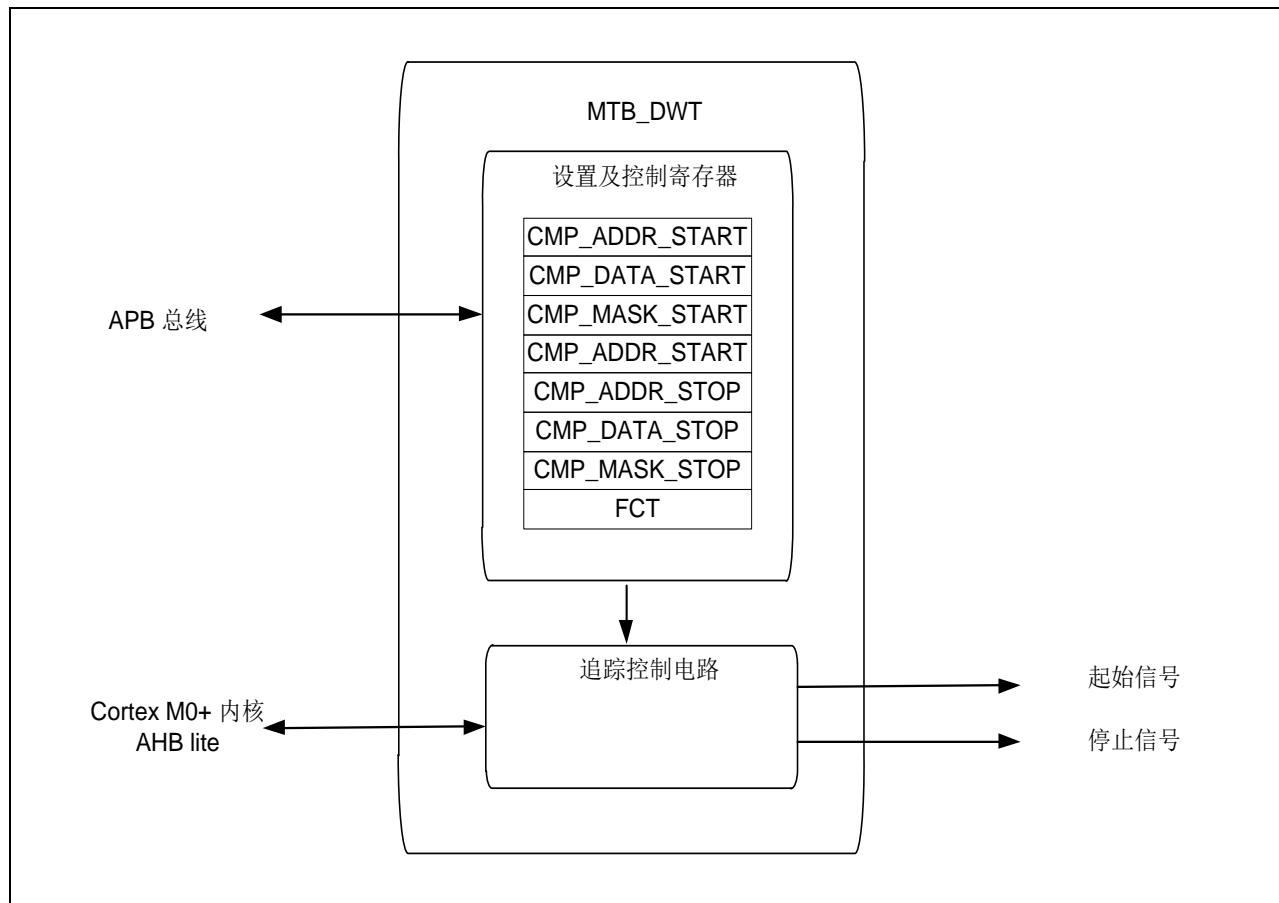
MTB_DWT 的特征

- 支持地址或地址+数据的可编程起始/停止记录
- 支持屏蔽数据的每一位
- 支持监测写/读以及写或读的操作
- 支持字节，半字以及字的监测
- 支持数据访问时的监测功能

16.2 架构框图

以下介绍了 MTB_DWT 的架构框图。

图 16.2-1 MTB_DWT 的架构框图



16.3 配置以及设定步骤示例

以下介绍了配置以及设定步骤示例。

16.3.1 MTB_DWT 的配置

以下介绍了 MTB_DWT 的配置：

MTB_DWT 的配置概要

MTB_DWT 只支持小端存储格式，当进行字节以及半字访问时， MTB_DWT 数据比较起始追踪寄存器(CMP_DATA_START) 以及 MTB_DWT 数据比较停止追踪寄存器(CMP_DATA_STOP) 没有用到的比特位如表 16.3-1 所示。表中 "○" 符号表示有效数据， "-" 符号表示无效数据。

可以通过设定寄存器 CMP_MASK_START/CMP_MASK_STOP 来屏蔽无效数据中相应的位。

表 16.3-1 AHB-Lite 字节通道定义

访问方式	地址段	相应的数据访问段			
	CMP_ADDR x*1 [1:0]	CMP_DATA x*2 [31:24]	CMP_DATA x*2 [23:16]	CMP_DATA x*2 [15:8]	CMP_DATA x*2 [7:0]
字节	00	-	-	-	○
	01	-	-	○	-
	10	-	○	-	-
	11	○	-	-	-
半字	00	-	-	○	○
	10	○	○	-	-
字	00	○	○	○	○

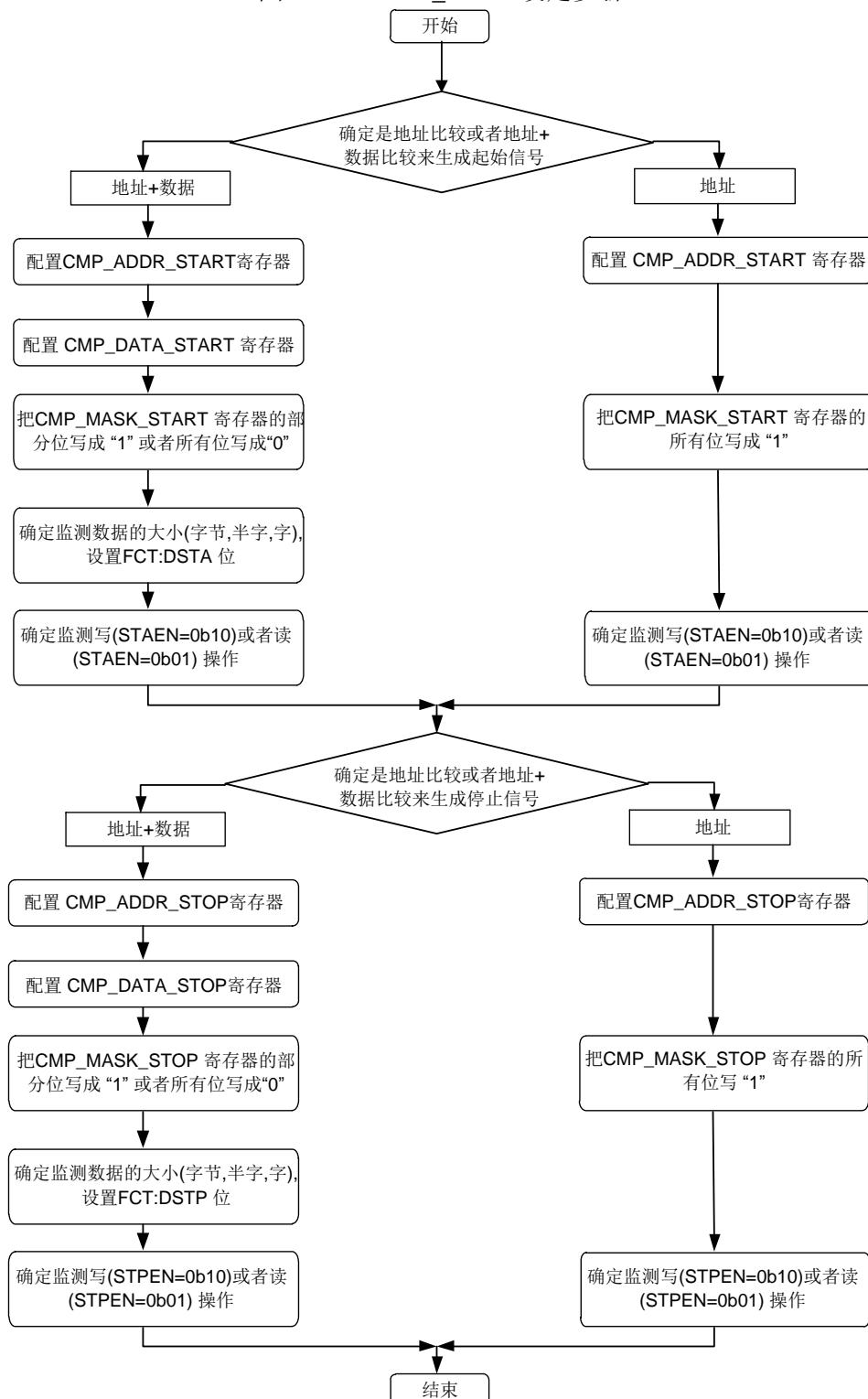
* 1.CMP_ADDRx: CMP_ADDR_START 寄存器或 CMP_ADDR_STOP 寄存器

* 2.CMP_DATAx: CMP_DATA_START 寄存器或 CMP_DATA_STOP 寄存器

设定步骤

MTB_DWT 的设定步骤示例如图 16.3-1 所示。

图 16.3-1 MTB_DWT 设定步骤



16.4 寄存器

以下介绍 MTB_DWT 的寄存器。

表 16.4-1 MTB_DWT 寄存器列表

寄存器简称	寄存器名称	参照章节
CMP_ADDR_START	MTB_DWT 地址比较起始追踪寄存器	16.4.1
CMP_DATA_START	MTB_DWT 数据比较起始追踪寄存器	16.4.2
CMP_MASK_START	MTB_DWT 屏蔽数据比较起始追踪寄存器	16.4.3
CMP_ADDR_STOP	MTB_DWT 地址比较停止追踪寄存器	16.4.4
CMP_DATA_STOP	MTB_DWT 数据比较停止追踪寄存器	16.4.5
CMP_MASK_STOP	MTB_DWT 屏蔽数据比较停止追踪寄存器	16.4.6
FCT	MTB_DWT 功能寄存器	16.4.7
PID4	外围 ID4 寄存器	16.4.8
PID5	外围 ID5 寄存器	16.4.9
PID6	外围 ID6 寄存器	
PID7	外围 ID7 寄存器	
PID0	外围 ID0 寄存器	
PID1	外围 ID1 寄存器	
PID2	外围 ID2 寄存器	
PID3	外围 ID3 寄存器	
CID0	元件 ID0 寄存器	
CID1	元件 ID1 寄存器	
CID2	元件 ID2 寄存器	
CID3	元件 ID3 寄存器	

16.4.1 MTB_DW 地址比较起始追踪寄存器 (CMP_ADDR_START)

MTB_DWT 地址比较起始追踪寄存器(CMP_ADDR_START) 可提供一个参考地址值来产生起始触发信号。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	ADCMP_STA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	ADCMP_STA[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[31:0] ADCMP_STA[31:0]: MTB_DWT 地址比较起始追踪位

地址比较的参考值用来产生 MTB 起始触发信号。

注意：

- 字访问时，3~0 位必须设置成 0x0, 0x4, 0x8 和 0xC 中的一种。
- 半字访问时，1~0 位必须设置成 0x0 和 0x2 中的一种。
- 该寄存器会被 PRESET0 复位，PRESET0 复位后，该寄存器需要重新配置。

16.4.2 MTB_DWT 数据比较起始追踪寄存器 (CMP_DATA_START)

The MTB_DWT 数据比较起始追踪寄存器(CMP_DATA_START) 可提供一个参考数据值来产生起始触发信号。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	DTCMP_STA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	DTCMP_STA[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[31:0] DTCMP_STA[31:0]: MTB_DWT 数据比较起始追踪位

数据比较的参考值用来产生 MTB 起始触发信号。

当所有位都被 CMP_MASK_START 寄存器屏蔽时, MTB 起始信号只由地址比较结果来确定;如果不是则 MTB 起始信号被数据(CMP_DATA_START) 和地址 (CMP_ADDR_START) 比较结果同时确定。

注意:

- 字节或者半字访问时, 根据表 3-1 设置 CMP_MASK_START 寄存器来屏蔽 CMP_DATA_START 寄存器中没有用到的位。
- 该寄存器会被 PRESET0 复位, PRESET0 复位后,该寄存器需要重新配置。

16.4.3 MTB_DWT 屏蔽数据比较起始追踪寄存器(CMP_MASK_START)

The MTB_DWT 屏蔽数据比较起始追踪寄存器(CMP_MASK_START) 定义了 MTB_DWT 数据比较起始追踪寄存器(CMP_DATA_START)中忽视的寄存器位。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名																
属性																
初始值																

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名																
属性																
初始值																

寄存器功能

[31:0] MSK_STA[31:0]: MTB_DWT 数据比较起始追踪寄存器屏蔽位

该寄存器会屏蔽用于起始 MTB 的参考数据值。

bit	功能
0	操作无影响 [初始值]
1	屏蔽对应的位

注意:

- 该寄存器的所有位被设置成“1”时, CMP_DATA_START 寄存器的值以及 FCT 寄存器中 DSTA 位的值会被忽视。该寄存器会被 PRESET0 复位, PRESET0 复位后,该寄存器需要重新配置。

16.4.4 MTB_DWT 地址比较停止追踪寄存器(CMP_ADDR_STOP)

MTB_DWT 地址比较停止追踪寄存器(CMP_ADDR_STOP) 可提供一个参考地址值来产生停止触发信号。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	ADCMP_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	ADCMP_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[31:0] ADCMP_STO[31:0]: MTB_DWT 地址比较停止追踪位

地址比较的参考值用来产生 MTB 停止触发信号。

注意：

- 字访问时，3~0 位必须设置成 0x0, 0x4, 0x8 和 0xC 中的一种。
- 半字访问时，1~0 位必须设置成 0x0 和 0x2 中的一种。
- 该寄存器会被 PRESET0 复位，PRESET0 复位后，该寄存器需要重新配置。

16.4.5 MTB_DWT 数据比较停止追踪寄存器(CMP_DATA_STOP)

MTB_DWT 数据比较停止追踪寄存器 (CMP_DATA_STOP) 可提供一个参考数据值来产生停止触发信号。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	DTCMP_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	DTCMP_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[31:0] DTCMP_STO[31:0]: MTB_DWT 数据比较停止追踪位

数据比较的参考值用来产生 MTB 停止触发信号。

当所有位都被 MTB_DWT 屏蔽数据比较停止追踪寄存器屏蔽时, MTB 停止信号只由地址比较结果来确定; 如果不是则 MTB 停止信号被数据(MTB_DWT 数据比较停止追踪寄存器)和地址(MTB_DWT 地址比较停止追踪寄存器)比较结果同时确定。

注意:

- 字节或者半字访问时, 根据表 16.3-1 设置 CMP_MASK_STOP 寄存器来屏蔽 CMP_DATA_STOP 寄存器中没有用到的位。
- 该寄存器会被 PRESET0 复位, PRESET0 复位后, 该寄存器需要重新配置。

16.4.6 MTB_DWT 屏蔽数据比较停止追踪寄存器 (CMP_MASK_STOP)

The MTB_DWT 屏蔽数据比较停止追踪寄存器 (CMP_MASK_STOP) 定义了 MTB_DWT 数据比较停止追踪寄存器(CMP_DATA_STOP)中忽视的寄存器位。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	MSK_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	MSK_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[31:0] MSK_STO[31:0]: MTB_DWT 数据比较停止追踪寄存器屏蔽位

该寄存器会屏蔽用于停止 MTB 的参考数据值。

bit	功能
0	操作无影响 [初始值]
1	屏蔽对应的位

注意:

- 该寄存器的所有位被设置成“1”时, CMP_DATA_STOP 寄存器的值以及 FCT 寄存器中 DSTP 位的值会被忽视。

- 该寄存器会被 PRESET0 复位，PRESET0 复位后,该寄存器需要重新配置。

16.4.7 MTB_DWT 功能寄存器 (FCT)

MTB_DWT 功能寄存器 (FCT) 控制读/写操作和数据大小。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	保留															
属性	-															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
字段名	保留								DSTP	DSTA	STPEN	STAEN					
属性	-								R/W	R/W	R/W	R/W					
初始值	00000000								00	00	00	00					

寄存器功能

[31:8] 保留位:读值是“0”，写操作无效。

[7:6] DSTP: 停止数据大小位，本位定义停止 MTB 功能的数据大小位。

bit7	bit6	功能
0	0	字节 [初始值]
0	1	半字
1	0	字
1	1	保留

[5:4] DSTA: 起始数据大小位，本位定义起始 MTB 功能的数据大小位。

bit5	bit4	功能
0	0	字节 [初始值]
0	1	半字
1	0	字
1	1	保留

[3:2] STPEN: 使能 MTB_DWT 停止 MTB 功能位

本位使能 MTB_DWT 停止 MTB 功能。

bit3	bit2	功能
0	0	禁止 MTB_DWT 停止 MTB 功能。 [初始值]
0	1	数据读取操作停止MTB。

1	0	数据写入操作停止MTB。
1	1	数据读取或写入操作停止MTB。

[1:0] STAEN: 使能 MTB_DWT 起始 MTB 功能位

本位使能 MTB_DWT 起始 MTB 功能。

bit1	bit0	功能
0	0	禁止 MTB_DWT起始 MTB 功能。 [初始值]
0	1	数据读取操作起始MTB。
1	0	数据写入操作起始MTB。
1	1	数据读取或写入操作起始MTB。

注意：

- 该寄存器会被 PRESET0 复位，PRESET0 复位后，该寄存器需要重新配置。

16.4.8 外围 ID0 ~ 外围 ID7 寄存器 (PIDx)

外围 ID0-7 寄存器 (PID0-7) 指示了外围 ID。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	PERIDx[31:16]															
属性	R															
初始值	0xFFFF															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	PERIDx[15:0]															
属性	R															
初始值	0xFFFF															

寄存器功能

[31:0] PERIDx[31:0]: 外围 IDx 位

本位是硬连接的特定值，通过外部调试用于自动识别流程中。

bit	功能
读取时	读取的是特定值 PID0, PID3 到 PID7: 0x00000000. PID1: 0x00000070 PID2: 0x0000000E
写入时	操作无影响

16.4.9 元件 ID0 ~元件 ID3 寄存器 (CIDx)

元件 ID0-3 寄存器 (CID0-3) 指示了元件 ID。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	CPNTIDx[31:16]															
属性	R															
初始值	0xXXXX															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	CPNTIDx[15:0]															
属性	R															
初始值	0xXXXX															

寄存器功能

[31:0] CPNTIDx[31:0]: 元件 IDx 位

本位是硬连接的特定值，通过外部调试用于自动识别流程中。

bit	功能
读取时	读取的是特定值。 CID0: 0x0000000D; CID1: 0x00000090; CID2: 0x00000005; CID3: 0x000000B1.
写入时	操作无影响

17. 闪存

17.1 概况

本产品闪存由 128KB 主要数据存储区及 4B (字节) 数据安全区，4 个字的用户数据区构成。闪存能够实现字节、半字、字编程。同时可以根据不同的寄存器设定来实现 512 字节的页擦除，或者整片擦除。

闪存有如下特点：

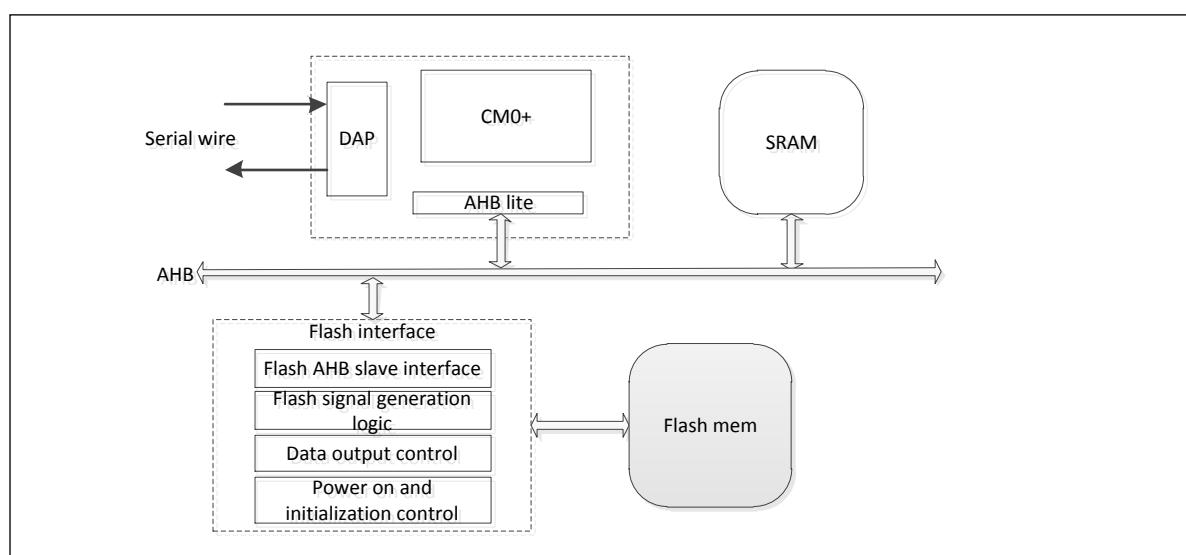
- 容量：高达 128KB 的主要数据存储区，每个页由 512 字节构成；4 字节安全数据（现 2 字节使用），4 个字用于 trimming 及测试
- 编程：8/16/32 位
- 读：8/16/32 位，0 周期等待@16Mhz
- 擦除：页、片
- 内置闪存安全

参考闪存安全章节

17.2 模块图

闪存接口控制由 AHB 从动接口、上电初始化逻辑、控制逻辑，输出控制逻辑构成。下面是一个简化图。

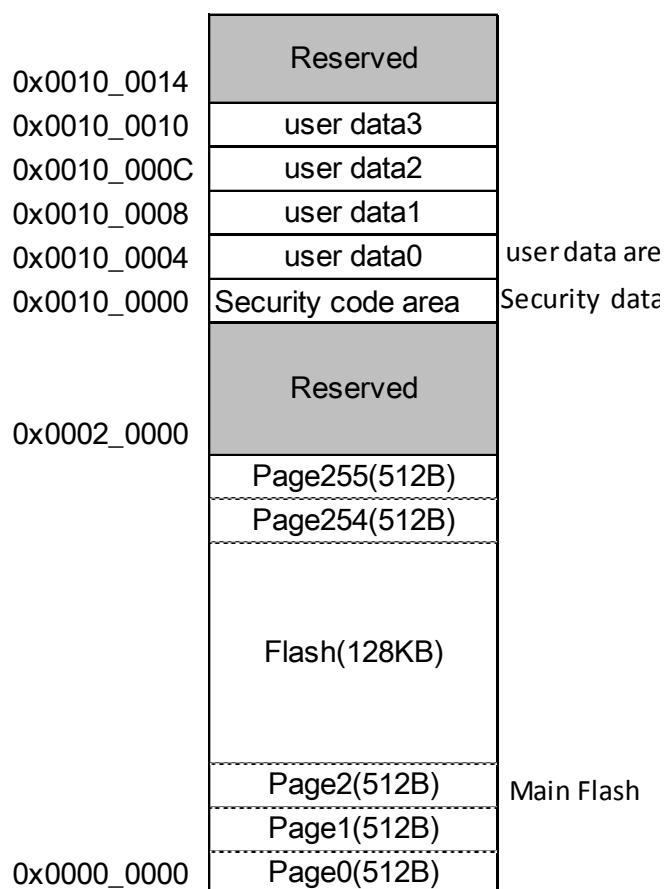
图 17.2-1 闪存控制图



17.3 闪存存储构成

闪存由 128K 字节数据区，4 字节安全数据区，以及 4 字用户数据和测试数据构成。

图 17.3-1 闪存存储布局



“用户数据” 区域中存储了包括内部高速 RC 时钟振荡器的频率调校预设值在内的保证芯片正常工作所需的 trimming 和测试数据。

当擦除片上闪存内的数据时，“用户数据” 区域中的数据会被同时擦除。为了保证芯片正常工作，请在对片上闪存做全区域擦除前，把“用户数据” 区域中的数据保存在其他区域，比如片上 SRAM 存储空间。若未对“用户数据”区域中的数据进行保存，会导致芯片无法正常使用。

4 字节安全代码数据区存放闪存安全数据。现只有低半字使用。

图 17.3-2 安全数据区



17.4 操作

本段将涉及闪存操作及注意事项。

17.4.1 闪存操作

闪存支持下面一系列操作。

读、复位操作

这段讲述读、复位操作。

闪存在上电完成或者编程、擦除操作完成之后，进入读、复位状态。读闪存地址空间内任一地址，闪存就将进入读、复位状态。读、复位是闪存的默认状态，闪存在上电或操作完成之后都将进入这个状态。CPU 可以访问闪存地址范围内的任一地址。

写操作

这段讲述写操作。

在开始写操作之前，检查闪存状态寄存器 FSTR 的 BUSY 位。对闪存控制寄存器 FOGR PRGM 位写 1，进入写状态，写任意闪存范围内地址，等待操作完成。

当闪存进入编程过程后，BUSY 位将会为高，完成编程后，该位自动清除。如果编程操作是在 RAM 中程序发起的话，CPU 在 BUSY 为高时不能访问闪存。否则，访问违例将出发，状态寄存器 ACCVIOFLG 标志位置 1。编程操作结果不可预知。如果程序在闪存中执行，CPU 将会停下来，直到操作完成，接着执行后面的程序。

当进行编程时，不可以对已经编程的地址编程。同时，对整个页进行编程时，建议采用字，半字编程，既能提高编程效率，同时也可避免累计编程高压周期 Thv。

闪存能够通过 RAM，或者闪存本身编程，但在闪存中编程时，不可以对程序执行页进行编程操作。下面分别是程序在闪存中和 RAM 中编程闪存的流程。

图 17.4.1-1 程序在闪存中编程闪存

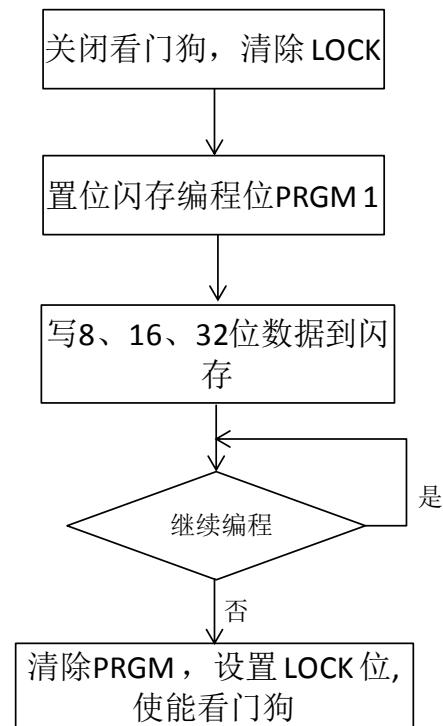
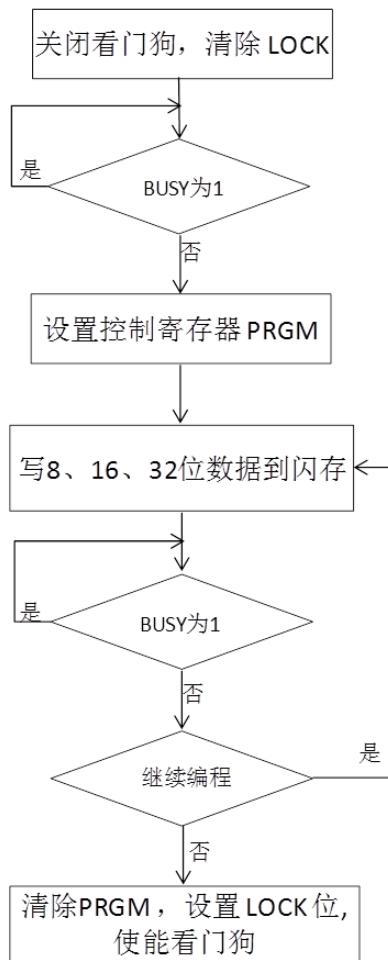


图 17.4.1-2 程序在 RAM 中编程闪存



页擦除

页擦除可以在闪存或者 RAM 中发起。首先需要设置闪存控制寄存器的 PERS 位，然后向需要擦除的页（512B）内的一个任意地址写任意数据（称为虚写操作），将会触发页擦除操作。页操作会将该页内的数据擦除。BUSY 位置 1 直到页擦除完成。如果程序在 RAM 中执行，不可以再对闪存进行操作。否则，访问违例发生，ACCVIOFLG 位置 1 。如果程序在闪存中执行，CPU 将会停止操作直到页擦除完成。程序将接着虚写之后继续执行。下面分别是 RAM 和闪存中执行页擦除的操作流程。

图 17.4.1-3 RAM 中执行页擦除

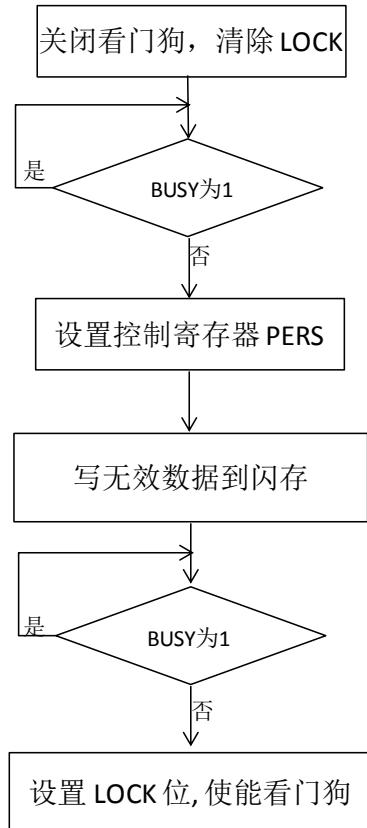
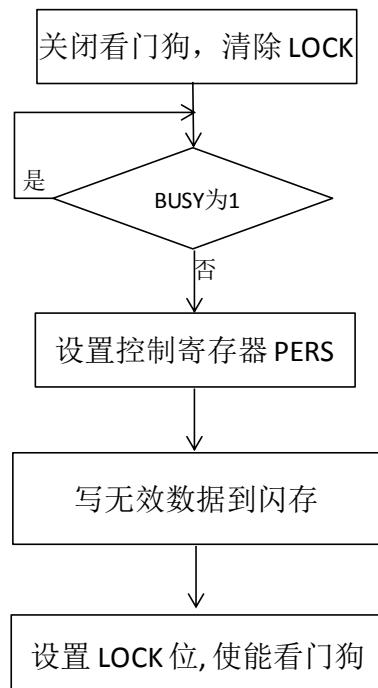


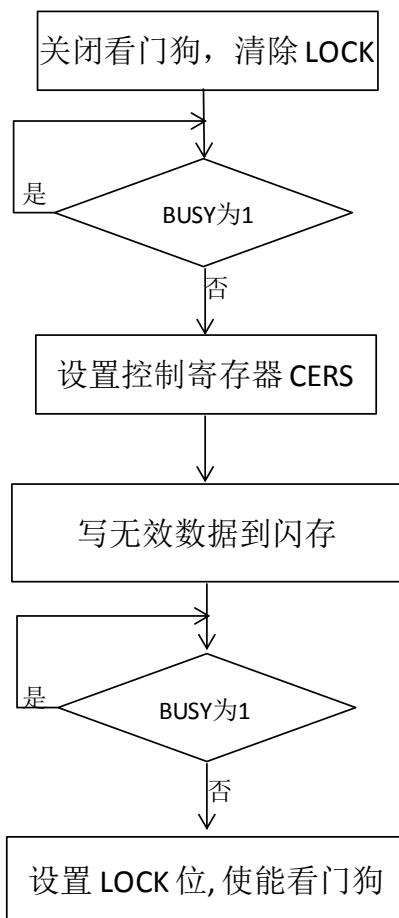
图 17.4.1-4 闪存中执行页擦除



片擦除

片擦除通过在 RAM 中执行程序，设置控制寄存器 CERS 位来完成。片擦除将擦除所有 128K 字节的主要数据区，4 字节的安全数据，4 个字的用户数据区。设置 CERS 之后，一个对闪存地址范围内任意地址的虚写操作将触发片擦除操作。在擦除期间，BUSY 位保持高直到擦除完成。在 BUSY 为高时，不可以对闪存进行操作。否则，访问违例将触发，ACCVIOFLG 置 1。程序执行结果不可预知。

图 17.4.1-5 RAM 中执行片擦除



当擦除存储于 FLASH 的数据时，”User datax”区域中的数据会被同时擦除。

为了保证芯片正常工作，请在对 FLASH 进行全芯片擦除前，把”User datax”区域中的数据存在其他区域，比如 RAM 存储空间。

若未对”User datax”区域中的数据进行保存，会导致芯片无法正常使用。 $(x=0,1,2,3)$

17.4.2 编程或擦除时闪存操作

当 $BUSY = 1$ ，从 RAM 中发起对闪存的写或者擦除，CPU 对闪存的操作将是无效的。这样会

导致访问违例发生，ACCVIOFLG 位置 1。同时，如果没有对闪存控制器中的相关位进行设置，就对闪存进行编程、擦除，也会使得 ACCVIOFLAG 置 1。如果写操作由闪存中程序自己发起，CPU 将等待闪存操作完成 (BUSY=0)，接着往下继续执行指令。

表 17.4.2-1 BUSY=1 闪存操作 (RAM 中发起)

闪存操作	总线访问	结果
片擦除	读	不确定返回数据，ACCVIOFLG 不置1.
	写	ACCVIOFLG置1。写无效。
页擦除	读	不确定返回数据，ACCVIOFLG 不置1.
	写	ACCVIOFLG置1。写无效。
编程	读	不确定返回数据，ACCVIOFLG 不置1.
	写	ACCVIOFLG置1。写无效。

在对闪存操作时，中断应当关闭。在进行擦除的时候，看门狗也应当关闭，像流程图里提到的一样。复位将打断擦除操作，后果不可预知。擦除完成之后再使能看门狗电路。

17.4.3 闪存控制状态标志和中断

闪存控制器有 4 个中断源，分别是 OPVIO,EER,ACCVIOFLG 和 KEYVIO. ACCVIOFLG 在访问违例时置位。如果中断使能位设置有效，ACCVIOFLG 标志位就将触发中断。当中断发生时，软件通过查询状态寄存器可以得知中断的类型，从而做出相应的操作。在擦除或者 8、16、32 位编程的过程中，对控制器寄存器任何写操作都是违例操作，会触发 ACCVIOFLG。

当 BUSY=1 时，对闪存地址空间进行编程（写）或者在未设定闪存操作寄存器之前的任何的闪存编程或者擦除都是违例操作，触发 ACCVIOFLG。

闪存寄存器可以在 BUSY=1 时读出。读寄存器不会触发违例。

当闪存控制寄存器写入不正确的密码时，KEYVIO 置高。如果 FIE 使能中断，中断将触发。

当对锁住的页或者程序执行的页进行编程或者擦除时，擦除错误标志位 (EER) 将置 1。如果中断使能 FIE 使能，中断发生。

操作设置违例 (OPVIO) 在中断使能后也将触发中断。操作违例指的是不止一位操作写入闪存控制寄存器 FOCR，闪存控制器不知如何操作。这时，没有任何操作发生。

17.4.4 注意事项

操作闪存时注意下列事项：

如果芯片在编程闪存的时候，复位发生，那么将无法保证写入数据的正确性。

因此，有必要在编程或者擦除时，制止不必要的复位，如看门狗复位的发生。

17.5 寄存器

本章节描述闪存控制寄存器，下面是寄存器的列表。

表 17.5-1 闪存控制寄存器列表

缩写	名称	参考
FOCR	闪存操作控制寄存器	17.5.1
FLC0	闪存锁存寄存器0	17.5.2
FLC1	闪存锁存寄存器1	17.5.3
FSTR	闪存状态寄存器	17.5.4
FWCICR	闪存等待周期和中断控制寄存器	17.5.5
Tnvs	Tnvs 时间设置寄存器	17.5.8
Tprog	Tprog 时间设置寄存器	17.5.9
Tpgs	Tpgs 时间设置寄存器	17.5.10
Trcv_pgm	编程 Trcv 时间设置寄存器	17.5.11
Trcv_pers	页擦除 Trcv 时间设置寄存器	17.5.12
Trcv_cers	片擦除 Trcv 时间设置寄存器	17.5.13
Tpers	页擦除 Terase 时间设置寄存器	17.5.14
Tcers	片擦除 Terase 时间设置寄存器	17.5.15

17.5.1 闪存操作控制寄存器 (FOCR)

闪存操作控制寄存器

位	15		4	3	2	1	0
字段名		-		DPSTB	CERS	PERS	PRGM
属性		-		R/W	R/W	R/W	R/W
初值		-		0	0	0	0

位[31:16] FRWPW: 闪存读、写密码，这几位时闪存读写操作密码。读这几位始终返回 0000h。

写入 DPSTB, CERS, PERS, PRGM 的数据只在写入正确的密码 1ACC 到这几位的同时，才能够有效。

值	描述
1ACC	解锁操作控制寄存器
其他	KEYVIO 触发

位[15:4] 预留位,读始终是 0.

位[3]DPSTB: 闪存深度休眠模式

用户设置该位，可以让闪存进入深度休眠模式。该位不能和 PRGM 、 PERS 、 CERS 同时设置。注意：程序在 RAM 中执行时才能设置该位，否则，程序在闪存中将无法继续执行。

值	描述
1	闪存进入深度休眠模式
0	闪存不进深度休眠模式（默认）

位[2]CERS: 片擦除操作

设置该位进入片擦除操作。 擦除完成后，该位自动清零。不需要软件操作。

值	描述
1	片擦除操作
0	没有片擦除操作（默认）

位[1]PERS: 页擦除操作

设置该位进入页擦除操作. 擦除完成后，该位自动清零。不需要软件操作。

值	描述
1	页擦除操作
0	没有页擦除操作（默认）

位[0]PRGM: 编程操作

设置该位进入编程操作。该位完成编程后不清零，软件需要手动清除。

值	描述
1	编程操作
0	无编程操作（默认）

17.5.2 闪存锁存寄存器 0(FLC0)

闪存锁存寄存器 0

位	31																0
字段名	lock2k[31:0]																
属性	R/W																
初值	32'hFFFFFFFFF																

17.5.3 闪存锁存寄存器 1(FLC1)

闪存锁存寄存器 1

位	63																32
字段名	lock2k[63:32]																
属性	R/W																
初值	32'hFFFFFFFFF																

字段名	位	描述
lock2k	63:0	闪存2k字节锁存寄存器 每一位 lock2k 锁住 2K 闪存空间。 0: 2KB 没有锁住，能够读、写、擦除； 1: 2KB 锁存住，只能读；

下表列出了 lock2k 每一位和它对应的 CPU 地址。

位	页	地址范围
lock2k[0]	Page0-page3	0x00000000-0x000007FF
lock2k[1]	Page4-page7	0x00000800-0x00000FFF
lock2k[2]	Page8-page11	0x00001000-0x000017FF
lock2k[3]	Page12-page15	0x00001800-0x00001FFF
...
lock2k[62]	Page248-page251	0x0001F000-0x0001F7FF
lock2k[63]	Page252-page255	0x0001F800-0x0001FFFF

位[n] 锁住页 n*4~n*4+3; 地址范围 n*(0800h) to n*(0800h)+7FF.

17.5.4 闪存状态寄存器 (FSTR)

闪存状态寄存器

位	31															16
字段名	FRWPW															
属性	R/W*															
初值	0000h															

位	15		5	4	3	2	1	0
字段名	-		OPVIO	EER	ACCVI OFLG	KEYVIO	BUSY	
属性	-		R/W	R/W	R/W	R/W	R/W	
初值	-		0	0	0	0	0	

位[31:16] FRWPW: 访问密码，这几位是状态寄存器访问密码。读出值始终是 FFFFh. 对状态寄存器的写只有在写入密码 1ACC 到 FRWPW 正确的情况下才有效。

值	描述
1ACC	解除这个寄存器
其他	KEYVIO 触发

位[15:5] 预留位,读始终是 0。

位[4]OPVIO: 闪存操作设置违例标志位。

这个位表示闪存控制寄存器操作设置违例。写 0 清除, 写 1 无效。该位置 1 表示控制寄存器 FOCR[2:0]有不止 1 位被设置。

值	描述
1	操作设置违例发生
0	无操作设置违例(默认)

位[3]EER: 闪存擦除设置错误标志位, 这位为高表示闪存擦除设置错误发生。写 0 清除, 写 1 无效。擦除错误发生在下列情况:

对锁存的页进行擦除或写;

对程序执行页进行擦除。

值	描述
1	擦除设置错误发生
0	无擦除设置错误 (默认)

位[2]ACCVIOFLG: 闪存访问违例标志位

这个位表示是否有访问违例发生。写 0 清除, 写 1 无效。

值	描述
1	访问违例发生
0	无访问违例 (默认)

位[1]KEYVIO: 闪存寄存器密码位错误标志位

操作闪存控制寄存器或者状态寄存器时, 密码输入不正确。写 0 清除, 写 1 无效。

值	描述
1	密码错误发生
0	无密码错误（默认）

位[0]BUSY: 闪存忙标志位

这位置 1 表示闪存在编程或者擦除过程中。完成后，自动清除。

值	描述
1	闪存忙
0	闪存不忙（默认）

17.5.5 闪存等待周期和中断控制寄存器 FWCICR

位	31				6	5	4	3	2	1	0
	Reserved						FWTC	Reserved			FIE
属性	R						R/W	R			R/W
初值	26'h0000000						0	3'b000			0

字段缩写	位	描述
预留位	31:6	预留位
FWTC	5: 4	闪存等待周期设置位 00: 读时0周期等待(HCLK频率≤16MHz时才可配置) 01: 读时1周期等待 1X: 读时2周期等待
预留位	3:1	预留位
FIE	0	中断使能位 0: 不是能中断 1: 使能中断

17.5.6 闪存 Tnvs 时间设置寄存器 Tnvs

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	1	0	0	0

字段缩写	位	描述
-	7:0	闪存Tnvs时间设置寄存器 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 ---Tnvs = 24*250ns(4Mhz) = 6us. 设置时请用这个公式 $Tnvs * 1/Fchip \geq 6\mu s$ Tnvs: 寄存器值; Fchip: 芯片工作频率。

17.5.7 闪存 Tprog 时间设置寄存器 Tprog

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	1	0	0	0

字段缩写	位	描述
-	7:0	闪存Tprog时间设置寄存器 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 ---Tprog = 24*250ns(4Mhz) = 6us 设置时请用这个公式 $Tprog * 1/Fchip = 6\sim 7.5\mu s$ Tprog: 寄存器值, Fchip: 芯片工作频率。

17.5.8 闪存 Tpgs 时间设置寄存器 Tpgs

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	1	0	0	0

字段缩写	位	描述
-	7:0	闪存Tpgs时间设置寄存器 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 ---Tpgs = 24*250ns(4Mhz) = 6us. 设置时请用这个公式 $Tpgs * 1/Fchip = 5\sim 6.5\mu s$, Tpgs: 寄存器值, Fchip: 芯片工作频率。

17.5.9 闪存编程 Trcv 时间设置寄存器 Trcv_pgm

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	1	0	0	0

字段缩写	位	描述
-	7:0	闪存编程Trcv时间设置寄存器, 闪存编程时恢复时间 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 ---Trcv_pgm = 24*250ns(4Mhz) = 6us 设置时请用这个公式Trcv_pgm*1/Fchip >= 5us, Trcv_pgm: 寄存器值 Fchip: 芯片工作频率

17.5.10 闪存页擦除 Trcv 时间设置寄存器 Trcv_pers

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	1	0	0	1

域名缩写	位	描述
-	7:0	闪存页擦除Trcv时间设定寄存器, 闪存页擦除恢复时间 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 --- Trcv_pers = 25*250ns(4Mhz) * 8= 50us 设置时请用这个公式Trcv_pers*1/Fchip*8>=50us, Trcv_pers : 寄存器值, Fchip : 芯片工作频率.

17.5.11 闪存片擦除 Trcv 时间设定寄存器 Trcv_cers

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	0	0	1	1	1

域名缩写	位	描述
-	7:0	闪存页擦除Trcv时间设定寄存器,闪存页擦除恢复时间 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 --- Trcv_cers = 7*250ns(4Mhz)*128 = 224us. 设置时请用这个 公式 Trcv_cers*1/Fchip*128>=200us, Trcv_cers : 寄存器值, Fchip : 芯片工作频率

17.5.12 闪存页擦除 Terase 设定时间寄存器 Tpers

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	0	1	0	0	0	0

域名缩写	位	描述
-	7:0	闪存页擦除Terase 时间设定寄存器 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 --- $Tpers = 16 * 250\text{ns}(4\text{Mhz}) * 1024 \sim 4\text{ms}$ 设置时请用这个公式 $Tpers * 1 / Fchip * 1024 = 4 \sim 5\text{ms}$ Tpers : 寄存器值, Fchip : 芯片工作频率

17.5.13 闪存片擦除 Terase 时间设定寄存器 Tcers

位	7	6	5	4	3	2	1	0
字段名	-							
属性	R/W							
初值	0	0	1	0	0	0	0	0

域名缩写	位	描述
-	7:0	闪存页擦除Terase 时间设定寄存器 请设置这个寄存器当改变芯片操作频率时。默认值是4MHz值 --- $Tcers = 32 * 250\text{ns}(4\text{Mhz}) * 4096 \sim 32\text{ms}$ 设置时请用这个公式 $Tcers * 1 / Fchip * 4096 = 30 \sim 40\text{ms}$ Tcers : 寄存器值, Fchip: 芯片工作频率。

注意:

- 当芯片工作频率变动时, 寄存器 Tnvs, Tprog 和 T* 需要设置。默认值是当芯片工作在 4Mhz 时设定的。下表提供了参考设置。

表 17.5.15-1 闪存时间设置寄存器推荐值设定

频率 寄存器	4Mhz	8Mhz	16Mhz	32Mhz
Tnvs	d'24	d'48	d'96	d'192
Tprog	d'24	d'48	d'96	d'192
Tpgs	d'24	d'48	d'96	d'192
Trcv_pgm	d'24	d'48	d'96	d'192
Trcv_pers	d'25	d'50	d'100	d'200
Trcv_cers	d'7	d'14	d'28	d'56
Tpers	d'16	d'32	d'64	d'128
Tcers	d'32	d'64	d'128	d'255

17.6 闪存安全概况

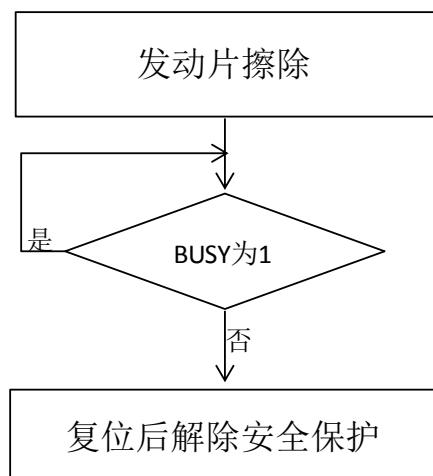
本节讲述存安全概况。

当保护码 0x0001 写入闪存安全数据区后，对闪存的访问就将受到限制。一旦闪存处于被保护状态，只有通过片擦除解除保护。这个功能适用于对程序和数据有安全保护要求的应用。

表 17.6-1 安全码与地址

地址	安全保护码
0x0010_0000	0x0001

图 17.6-1 解除安全功能流程



17.7 闪存安全保护功能

本节讲述闪存安全相关功能。

设置安全

将安全保护码 0x0001 写入闪存安全数据区地址。安全保护在复位或者上电复位之后有效。

解除安全

解除安全保护，对闪存主要数据存储区发动片擦除，复位芯片，安全保护解除如上页流程图。

安全使能后的操作

安全保护使能后，对不同模式操作将会不同。

图 17.7-1 给出了安全使能后不同模型下操作。

图 17.7-1 安全使能后不同模型下操作

模式	模式端口 MD0	访问闪存			调试接口
		片擦除	页擦除	读/写	
用户	0	使能	使能	使能	不能
串行下载	1	使能	不能	不能	不能

注意：

- 写保护码通常在完成闪存编程之后。这是为了避免编程过程中不必要的保护。
- 在用户模式下，闪存安全将无效，访问没有限制。但是，调试接口在安全使能之后是被锁死的。解除安全保护，用串行下载方式实施安全解除过程，因为调试口不可用。

17.8 闪存编程

17.8.1 编程方式

一个大概有 3 种编程方式，它们是：

- 1) 通过 JTAG/SWD 编程
- 2) 通过串行下载方式编程
- 3) 通过用户方法编程

17.8.2 通过 JTAG/SWD 编程

芯片可以通过 JTAG/SW 口进行闪存编程。JTAG 接口要求 4 个信号口，地以及 VCC 和 RSTB。SWD 需要 2 个信号口，加上地和 VCC，RSTB。现在只支持 SWD。

17.8.3 通过串行下载方式编程

串行下载器是由 HDSC 研发的板级编程工具。串行编程器支持 RS-232C 接口。

基本配置

串行下载通过时钟异步串行通信下载数据到用户系统的微控制器的内置闪存里。PC 和用户系统间是通过 RS-232C 线连接的。

在本系列，串行编程（UART 通信模式）能够使用外部高速晶振、外部高速时钟或者内置高速 CR 振荡器作为时钟源。

图 17.8.3-1 给出了 MCU 串行下载器的基本配置，表 17.8.3-1 列出了系统的配置。

图 17.8.3-1 串行下载器的基本配置

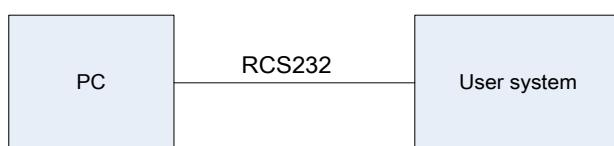


表 17.8.3-1 串行下载器配置

名字	功能	注释
串行下载器	软件	自行定义的编写闪存的软件.
RS-232C 线	市场有售	-

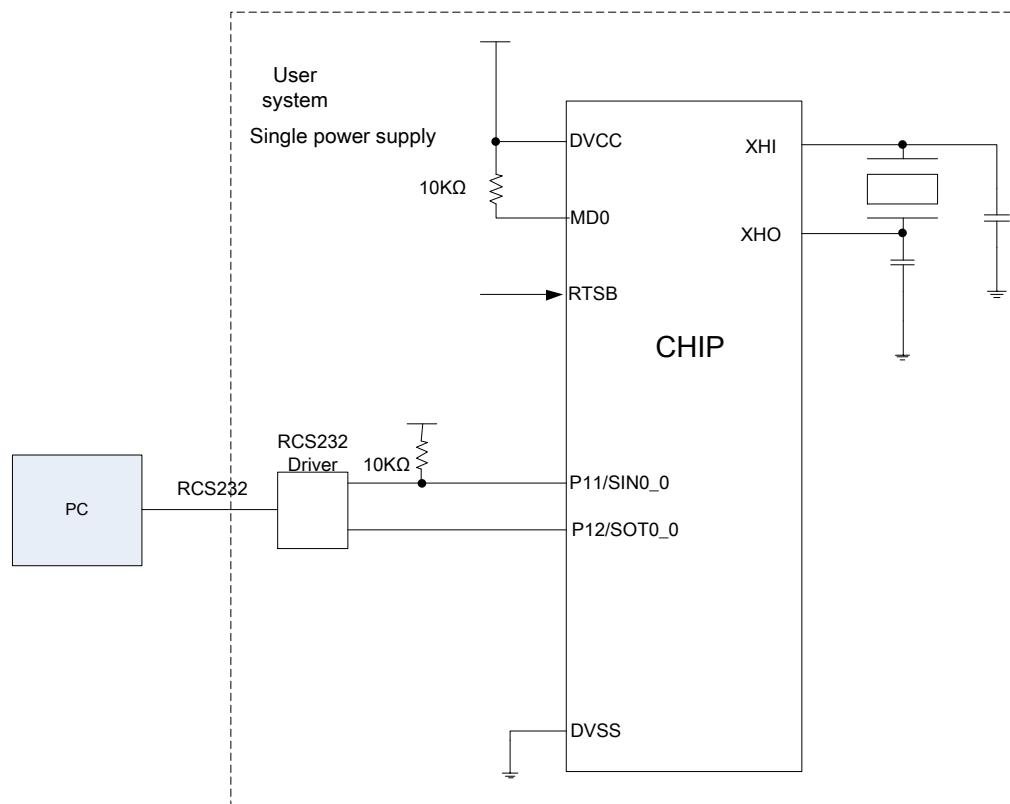
连接实例

下面展示串行下载器的一个连接实例。

当晶振使用时(如果配置)

下图展示了串行下载器在使用外部晶振作为时钟源与 RS-232C 驱动器之间进行通信的连接。

图 17.8.3-2 外部时钟使用时的外部串行下载器连接



电阻值只用作参考，请为不同系统选择合适的值。

表 17.8.3-2 时钟异步通信时的时钟频率和波特率

源频率	通信波特率
4MHz	9600bps
8MHz	19200bps
16MHz	38400bps

当内部 CR 振荡使用时（如果有）。

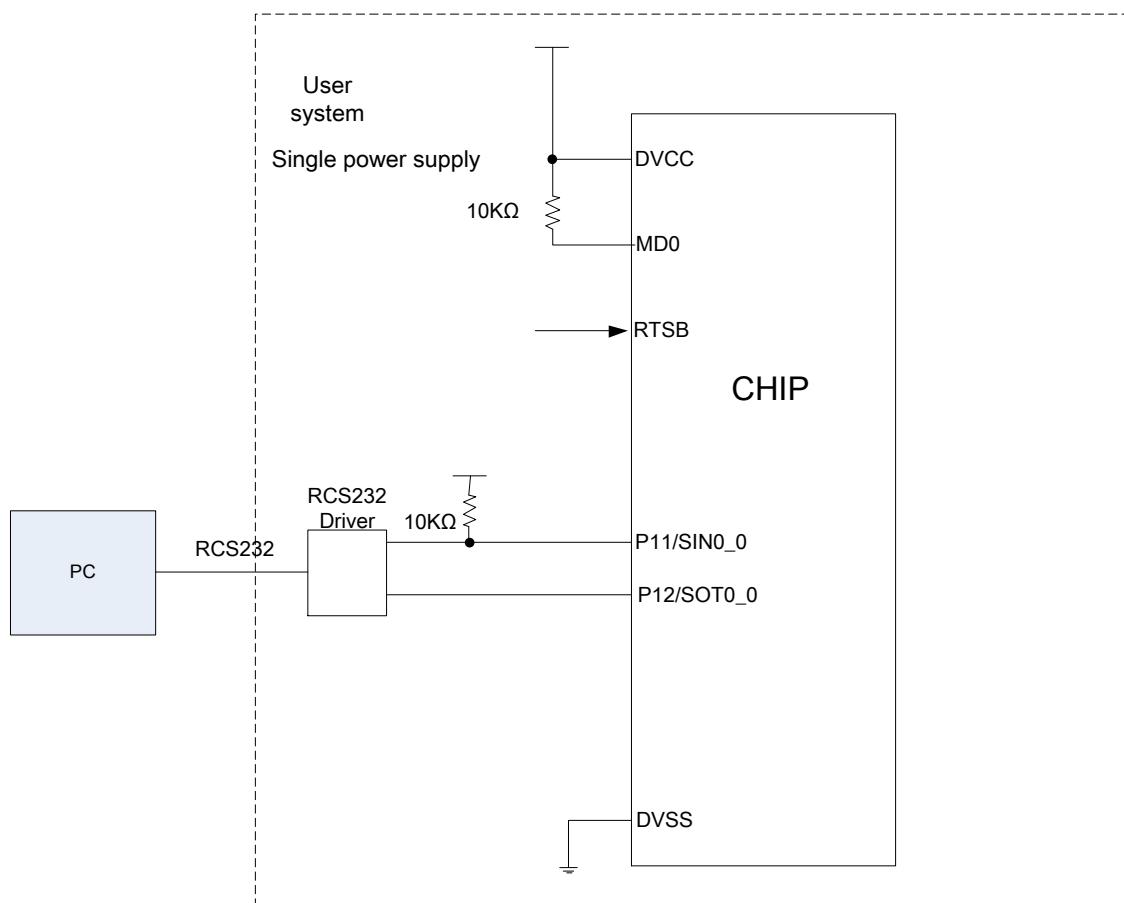
图 17.8.3-4 显示当内置高速 CR 振荡作为时钟振荡使用时的情况。

当既没有外部晶振也没有外部时钟连接到 XHI/XHO 引脚的时候，内置高速 CR 振荡器可以供通信使用。使用内置高速 CR 振荡器，有下列限制。

内置高速振荡器的振荡频率会随温度，电压的变动而抖动，从而超出能够允许的波特率错误范围。保持波特率在可接受的错误范围内。

对于使用内置高速振荡器，请参考有关产品的数据手册的内置 CR 振荡规范的相关内容。

图 17.8.3-3 使用内置高速 CR 振荡器时的串口下载器连接



电阻值仅供参考，请根据实际电路选择合适的值。

表 17.8.3-3 内部 CR 时钟源频率以及时钟异步串行通信波特率

CR 时钟频率	通信波特率
4MHz	9600bps
8MHz	19200bps
16MHz	38400bps

使用引脚

下面列出了串行编程需要的引脚。

表 17.8.3-4 串行编程使用引脚

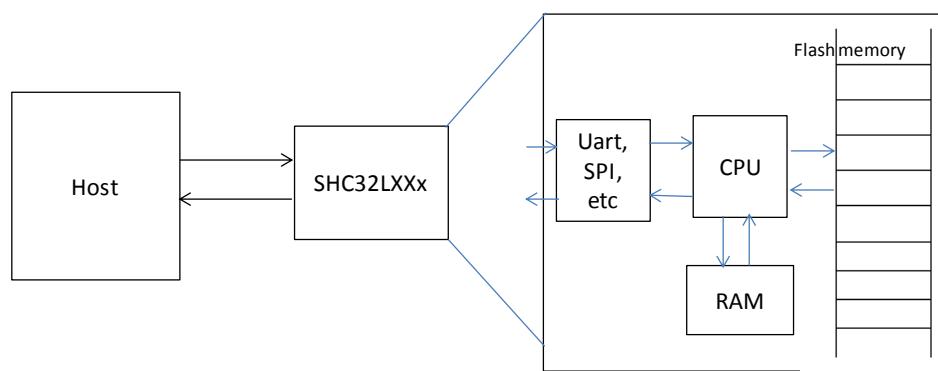
引脚名	功能	注释
MD0	模式引脚	当MD0=H 时, 实施复位或者打开电源, 进入串行编程模式。 当连接上拉或者下拉时, 避免长走线。
XHI/XHO	振荡引脚	主振荡时钟输入输出。参考产品数据手册, 查看可以用作串行编程模式的振荡时钟源(主时钟)频率
P12/SOT0_0	UART 串行输出	当通信模式设置为UART,这个引脚在串行编程模式启动后用做通信串行数据输出引脚。
P11/SIN0_0	UART 串行输入	当通信模式设置为UART,这个引脚在串行编程模式启动后用做通信串行数据输入引脚。
RSTB	复位引脚	
DVCC	电源	微控制器电源引脚
DVSS	地	

17.8.4 用户闪存编程

用户可以研发自己的闪存编程和擦出方式。写入闪存自身允许在线编程和外部用户编程方式。

用户可以通过任何可行的方式进行编程(例如, UART 或者 SPI)。因为这个方法是由用户自己开发的, 它完全可以用户化去满足各种需要编程、擦除、更新闪存的应用。

图 17.8.4-1 用户编程方法



18. 唯一识别码寄存器

18.1 概要

此功能概要阐述如下：

唯一识别码寄存器中存有预置的 80 位芯片唯一标识码，所有芯片的标识码的数值都不相同，可以有不同的用途，比如用于安全性的增强以及产品系列号。

该寄存器只能读取不能被用户写入，并且数值不会被复位或者上电下电改变。

18.2 寄存器

寄存器的配置以及功能阐释如下：

寄存器列表

寄存器简称	寄存器名称	参照章节
UIDR0	唯一识别码寄存器 0	18.2.1
UIDR1	唯一识别码寄存器1	18.2.2
UIDR2	唯一识别码寄存器2	18.2.3

18.2.1 唯一识别码寄存器 0 (UIDR0)

以下介绍了唯一识别码寄存器 0 的功能。

位	31	30	29	28	27	26	25	24
字段名	UID[31:24]							
属性	R							

位	23	22	21	20	19	18	17	16
字段名	UID[23:16]							
属性	R							

位	15	14	13	12	11	10	9	8
字段名	UID[15:8]							
属性	R							

位	7	6	5	4	3	2	1	0
字段名	UID[7: 0]							
属性	R							

[31:0] UID[31:0]：唯一识别码[31:0]

唯一识别码的 0 到 31 位。

18.2.2 唯一识别码寄存器 1 (UIDR1)

以下介绍了唯一识别码寄存器 1 的功能。

位	63	62	61	60	59	58	57	56
字段名	UID[63:56]							
属性	R							

位	55	54	53	52	51	50	49	48
字段名	UID[55:48]							
属性	R							

位	47	46	45	44	43	42	41	40
字段名	UID[47:40]							
属性	R							

位	39	38	37	36	35	34	33	32
字段名	UID[39:32]							
属性	R							

[31:0] UID[63:32] : 唯一识别码[63:32]

唯一识别码的 63 到 32 位。

18.2.3 唯一识别码寄存器 2(UIDR2)

以下介绍了唯一识别码寄存器 2 的功能。

位	31	30	29	28	27	26	25	24
字段名	保留位							
属性	R							

位	23	22	21	20	19	18	17	16
字段名	保留位							
属性	R							

位	79	78	77	76	75	74	73	72
字段名	UID[79:72]							
属性	R							

位	71	70	69	68	67	66	65	64
字段名	UID[71:64]							
属性	R							

[31:16] 保留位，读值无意义，写值无效。

[15:0] UID

[79:64] : 唯一识别码。

[79:64]，唯一识别码的 79 到 64 位。

19. 监视定时器

19.1 概要

本节说明监视定时器的概要。

监视定时器可以检测出用户程序是否失控。

如果在事先设定的间隔时间内不能清零监视定时器时，则会判断用户程序失控，并向 CPU 输出中断请求或者系统复位请求。

该中断请求称作监视器中断请求，而复位请求称作监视器复位请求。监视定时器工作时，在事先设定的间隔时间结束前，需要连续定时清零定时器。若发生用户程序挂起等异常动作，阻止了监视定时器进行定时清零，则该定时器会继续递减计数，发生下溢并输出监视器中断请求或者监视器复位请求。

如下所示，该系列有两类监视定时器：

- 软件监视定时器
 - 用户程序可启动软件监视定时器。
 - 计数时钟使用 APB 总线时钟的分频时钟。
 - 运行 CPU 程序时，该定时器计数；待机模式(Standby 模式、APB 时钟源时钟的振荡稳定等待时间)时 APB 时钟停止，该定时器停止计数。此时计数值保持，系统从待机模式返回时，继续计数。
 - 所有的复位均可停止软件监视定时器。
 - 提供窗口监视定时器模式。
- 硬件监视定时器
 - 上电后除了软件复位的所有复位释放后，如没有软件干预的话，硬件监视定时器启动。
 - 使用软件写寄存器可以停止硬件监视定时器。
 - 计数时钟使用内部低速的 CR 时钟(LICR)和外部低速时钟(LSXT)。
 - LICR/LSXT 工作时，该定时器计数；低功耗模式 STOP 模式时 LICR/LSXT 停止，则会停止计数。此时的计数值保持，从低功耗模式返回时，继续计数。

硬件/软件监视定时器

- 每个监视器有一个锁定寄存器，首先按照一定的步骤进行解锁，否则不可以访问监视定

时器的所有寄存器。

- 访问监视定时器清零寄存器可以重载监视定时器。
- 监视定时器第一次发生下溢时，产生中断请求。中断请求不清零的状态下，第二次发生下溢时，产生复位请求。可使用寄存器设定该功能。

19.2 构成框图

本节介绍监视定时器的框图。

图 19.2-1 软件监视定时器框图

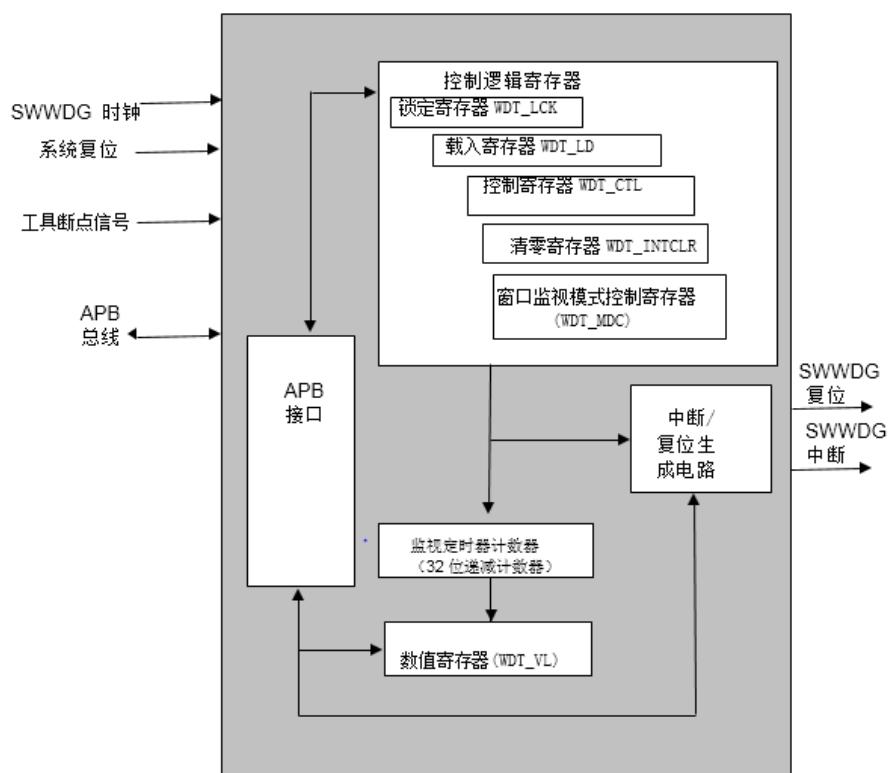
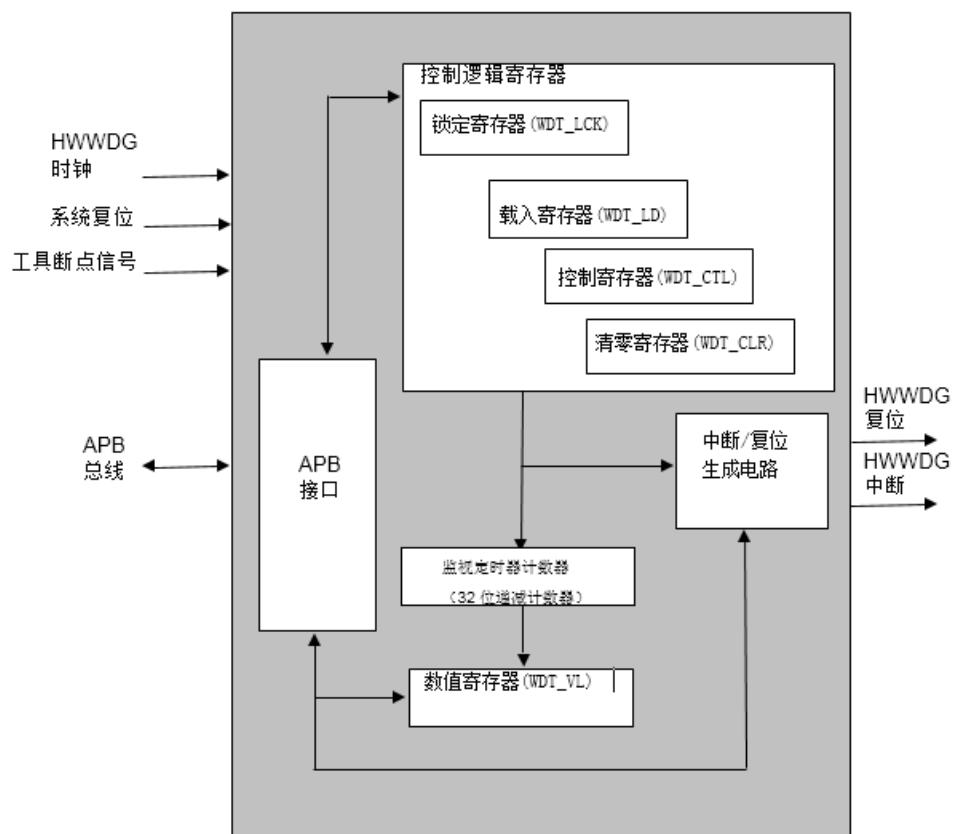


图 19.2-2 硬件监视定时器框图



19.3 操作说明

本节介绍监视定时器的操作。

监视定时器由以下部分构成。

19.3.1 软件监视定时器

控制逻辑寄存器

该电路控制软件监视定时器。该电路由载入寄存器、锁定寄存器、控制寄存器，清零寄存器和窗口监视模式控制寄存器构成。

1) 载入寄存器(WDT_LD)

该寄存器是 32 位寄存器，可以用来设定软件监视定时器的计数间隔周期。初始值是 "0xFFFFFFFF". 表 19.3.1-1 是间隔时间设定示例。

表 19.3.1-1 软件监视定时器的间隔时间设定示例

计数频率	间隔设定值	时间
32MHz	"0xFFFFFFFF" (初始值)	约134s
16MHz	"0xFFFFFFFF" (初始值)	约268s
32MHz	"0x0000FFFF"	约2.0ms
16MHz	"0x0000FFFF"	约4.1ms

2) 锁定寄存器(WDT_LCK)

该寄存器可以控制对软件监视定时器所有寄存器的访问。

本寄存器写"0x1ACCE551"可使能对软件监视定时器所有其它寄存器的写访问。

3) 控制寄存器(WDT_CTL)

该寄存器可设定软件监视器中断使能，软件监视器复位使能和窗口监视模式使能。

4) 清零寄存器(WDT_INTCLR)

该寄存器是软件监视定时器的清零寄存器。写任意 32 位值可从载入寄存器的设定值重新载入定时器，并继续计数。

5) 窗口监视模式控制寄存器(WDT_MDC)

该寄存器用作软件监视定时器的窗口监视模式的触发设置。

监视定时器计数器(32 位递减计数器)

该计数器是 32 位递减计数器。计数器值递减计数到"0"前访问清零寄存器(WDT_INTCLR)，可从载入寄存器(WDT_LD)的设定值重载计数值。

表 19.3.1-2 列出递减计数器的重载条件：

表 19.3.1-2 软件监视定时器的递减计数器的重载条件

重载条件
访问清零寄存器(WDT_INTCLR)
递减计数器计数到"0"时
修改载入寄存器(WDT_LD)
控制寄存器(WDT_CTL)写IE=0, 监视器停止; 写IE=1, 再启动。

数值寄存器(WDT_VL)

可读出监视定时器的计数值。

中断和复位产生电路

根据寄存器设定，当检测出监视定时器计数器发生下溢时，产生监视器中断或监视器复位。

中断状态寄存器(WDT_STA)

该寄存器指示软件监视器中断的状态。

软件监视定时器的启动

访问控制寄存器(WDT_CTL)时，可使能监视器中断和监视器复位。

表 19.3.1-3 列出监视器中断和监视器复位的设定组合。

表 19.3.1-3 软件监视器中断和复位的设定组合

中断	复位	操作说明
禁止	禁止	监视定时器不工作
使能	禁止	下溢时发生中断
禁止	使能	监视定时器不工作
使能	使能	第一次下溢时产生中断 第二次下溢

使能控制寄存器(WDT_CTL)的中断，可激活触发监视定时器。

软件监视定时器寄存器的重载和锁定

- 复位后的初始状态下，寄存器未被锁定。需要锁定时，请使用软件对 WDT_LCK 寄存器写入"0x1ACCE551"以外的值。
- 访问清零寄存器时，请对 WDT_LCK 寄存器写入"0x1ACCE551"解锁。
- 清零寄存器(WDT_INTCLR)赋任意值时，可重载载入寄存器(WDT_LD)的设定值。访问清零寄存器之后，寄存器不会自动锁定。请再次使用软件锁定。

软件监视定时器的停止

- 访问控制寄存器(WDT_CTL)，且监视器中断使能位写"0"，可停止软件监视定时器。
- 产生的复位可停止软件监视定时器。

窗口监视定时器模式

- 软件监视定时器有窗口监视模式
- 窗口监视模式用来判断计数器是否在正确的时间范围被软件重载

在如下情况中，具体的事件（中断或重置）会发生：

- 1) 当计数器发生下溢时
- 2) 当清零寄存器(WDT_INTCLR)在超出窗口时间被访问且计数器被清零时
- 3) 当数值寄存器(WDT_LD)在超出窗口时间被访问且计数器被重载时

19.3.2 硬件监视定时器

控制逻辑寄存器

该电路可以控制硬件监视定时器。该电路由载入寄存器、锁定寄存器、控制寄存器和清零寄存器构成。

1) 载入寄存器(WDT_LD)

该寄存器是 32 位寄存器，用于设定硬件监视定时器的计数间隔周期。初始值为 "0x0000FFFF" (16 位递减计数器=> 约 2048ms @ 32kHz(典型值))。关于计数时钟的内部 CR 频率，参考"数据手册"。

2) 锁定寄存器(WDT_LCK)

该寄存器可以控制对硬件监视定时器的寄存器的访问。该寄存器写"0x1ACCE551"时，可以访问控制寄存器(WDT_CTL)以外所有的寄存器。

3) 控制寄存器(WDT_CTL)

该寄存器可以设定监视器中断使能和监视器复位使能。要访问该寄存器，须向锁定寄存器写 "0x1ACCE551"后，再对锁定寄存器写"0xE5331AAE"。写入"0x1ACCE551"后若未能写入正确的值，须按顺序从头开始操作。

4) 清零寄存器(WDT_CLR)

该寄存器是硬件监视定时器的清零寄存器。

写任意 8 位值并写其反向值可从载入寄存器设定的设定值重载定时器计数器，然后继续计数。

监视定时器计数器(32 位递减计数器)

该计数器是 32 位递减计数器。计数器值递减计数到"0"前访问清零寄存器(WDT_CLR)，可从载入寄存器(WDT_LD)的设定值重载计数值。

表 19.3.2-1 列出递减计数器的重载条件。

表 19.3.2-1 硬件监视定时器递减计数器重载条件

重载条件
访问清零寄存器(WDT_CLR)
递减计数器计数到"0"时
重写载入寄存器(WDT_LD)
向控制寄存器(WDT_CTL): 写IE=0，监视器停止；写IE=1，再启动

数值寄存器(WDT_VL)

该寄存器可读出监视定时器当前的计数值。但是，工具断点时，只有监视定时器停止方可读出正确的数值。工具断点期间之外，因为是异步读取，有可能读取值不正确。此时，须采取读取 2 次并进行比较等措施。

中断和复位产生电路

根据寄存器的设定，该电路可在检测出监视定时器计数器的下溢时产生监视器中断或监视器复位。

中断状态寄存器(WDT_STA)

该寄存器指示硬件监视器中断的状态。

硬件监视定时器的启动

- 向锁定寄存器(WDT_LCK)写"0x1ACCE551"，并写入反向值"0xE5331AAE"，即可访问控制寄存器(WDT_CTL)。
- 访问控制寄存器(WDT_CTL)，使能监视器中断和监视器复位。

表 19.3.2-2 列出监视器中断和监视器复位的设定组合。

表 19.3.2-2 硬件监视器中断和复位的设定组合

中断	复位	操作说明
禁止	禁止	监视定时器不工作
使能	禁止	下溢时发生中断
禁止	使能	监视定时器不工作
使能	使能	第一次下溢时产生中断 第二次下溢时产生复位 [初始设定]

使能控制寄存器(WDT_CTL)的中断可触发启动硬件监视定时器。

硬件监视定时器寄存器的重载和锁定

清零寄存器(WDT_CLR)赋任意值可重载载入寄存器的设定值。重载后，再次锁定寄存器。此后，每次访问清零寄存器时，须进行解锁。

硬件监视定时器的停止

- WDT_LCK(锁定寄存器)先写"0x1ACCE551"，后写反向值"0xE5331AAE"，即可访问控制寄存器 (WDT_CTL)。
- 访问 WDT_CTL(控制寄存器)并向监视器中断使能位写"0"，可停止硬件监视定时器。

19.3.3 软件监视定时器和硬件监视定时器的差异

表 19.3.3-1 软件监视定时器和硬件监视定时器的差异

	软件监视定时器	硬件监视定时器
计数时钟	APB 时钟的分频	内部 LICR/外部 LSXT
数值寄存器的读值	同步读取。 —可读。	异步读取。 仅工具断点时可读取正确的值。 工具断点期间以外的读值有可能不正确。 正确。
监视器中断设定，复位设定初始值	禁止 —(监视定时器不动作)	使能 —(监视定时器动作)
寄存器锁定功能 初始状态	无锁定 —(软件启动后锁定)	锁定 —(硬件启动后锁定)
解锁	对锁定寄存器写 "0x1ACCE551"可使全部寄存器解锁	对锁定寄存器写"0x1ACCE551"可解锁除 WDT_CTL 以外的全部寄存器
WDT_CTL/ WDT_CTL 寄存器个别的闩锁解除	无	对锁定寄存器写"0xE5331AAE"可解锁 WDT_CTL 寄存器
再锁定条件	对 锁 定 寄 存 器 写 "0x1ACCE551"以外的值锁定 全部的寄存器。	WDT_CTL 以外的Lock 解锁后 <ul style="list-style-type: none"> · 对 WDT_LCK 写入"0x1ACCE551"或 "0xE5331AAE"以外的值 · 对 WDT_LD 写入 · 对 WDT_CTL 写入 · 对 WDT_CLR 再次写入 上记中的一项可再次将Lock 锁定。
		包括WDT_CTL 的Lock 解锁后 <ul style="list-style-type: none"> · 对 WDT_LCK 写入"0x1ACCE551"以外的值 · 对 WDT_LD 写入 · 对 WDT_CLR 写入 · 对 WDT_CTL 写入 上记中的一项可再次将Lock 锁定。
载入寄存器的初始值	0xFFFFFFFF	0x0000FFFF
清零寄存器位数	32 位	8 位
清零寄存器访问	写任意值清零	写任意值清零后,写任意值的反转值
窗口监视模式	可用	无

19.4 设定顺序示例

本节介绍监视定时器的设定顺序示例。

软件监视定时器

图 19.4-1 软件监视定时器的设定顺序示例 1

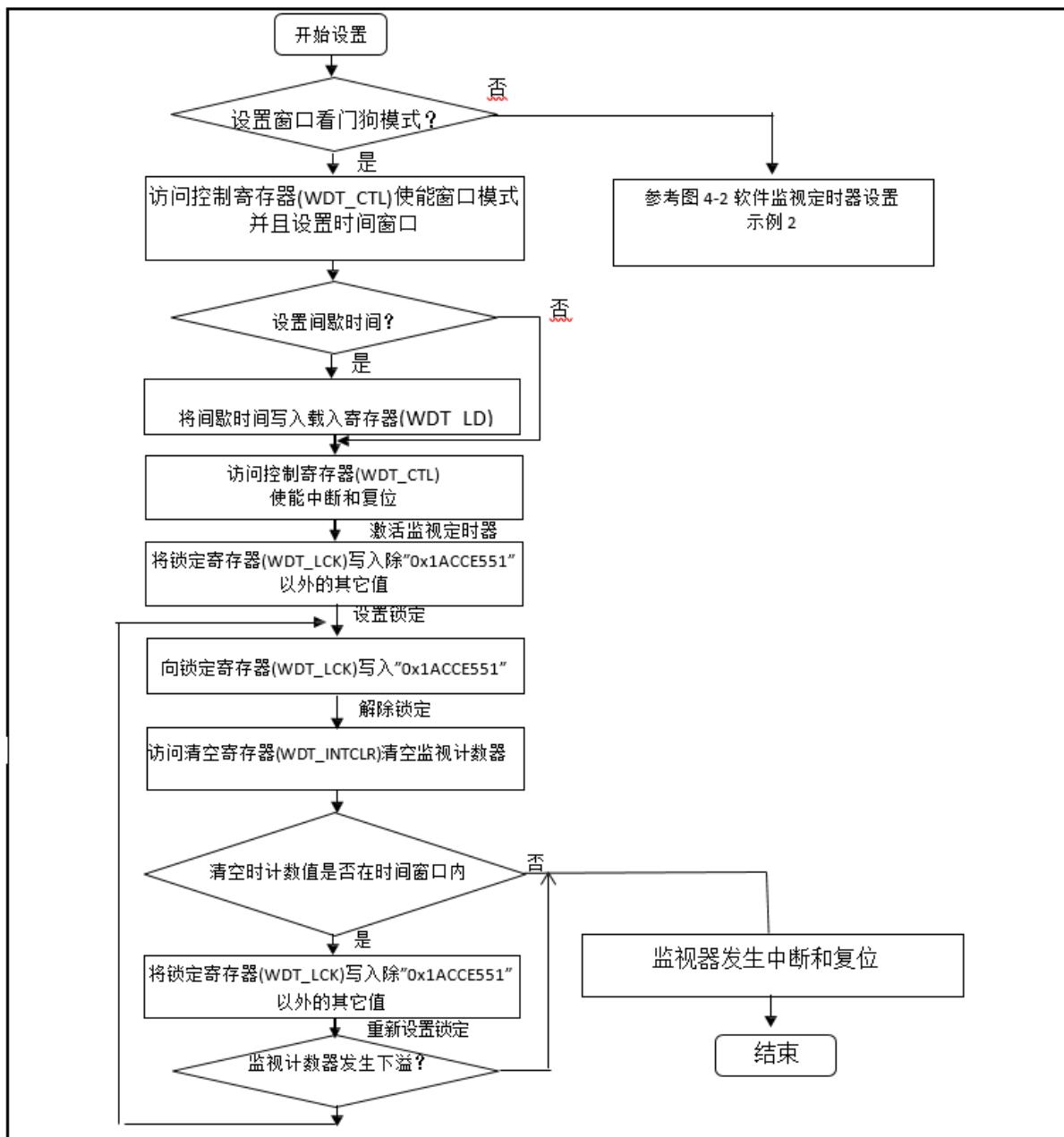
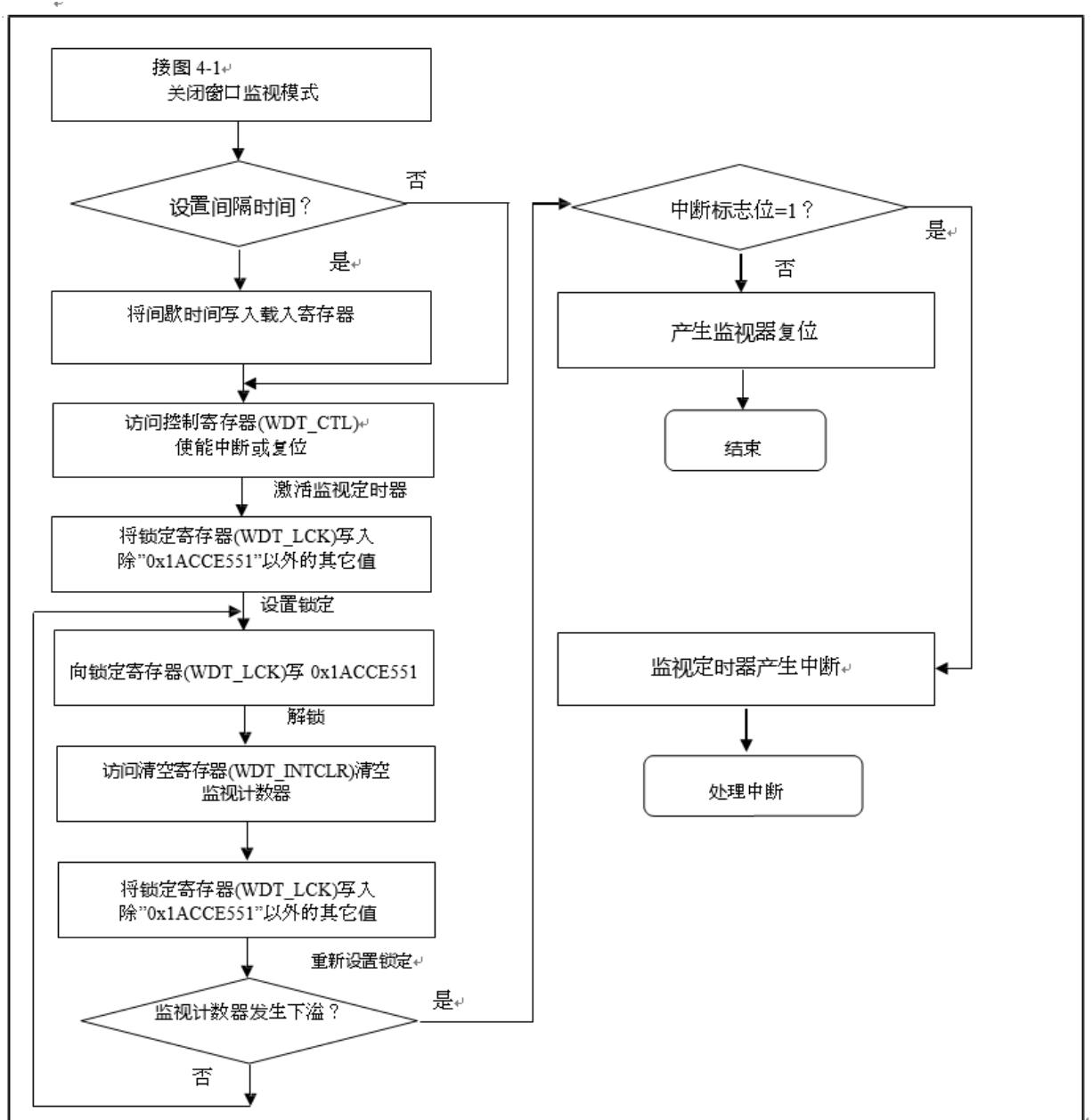
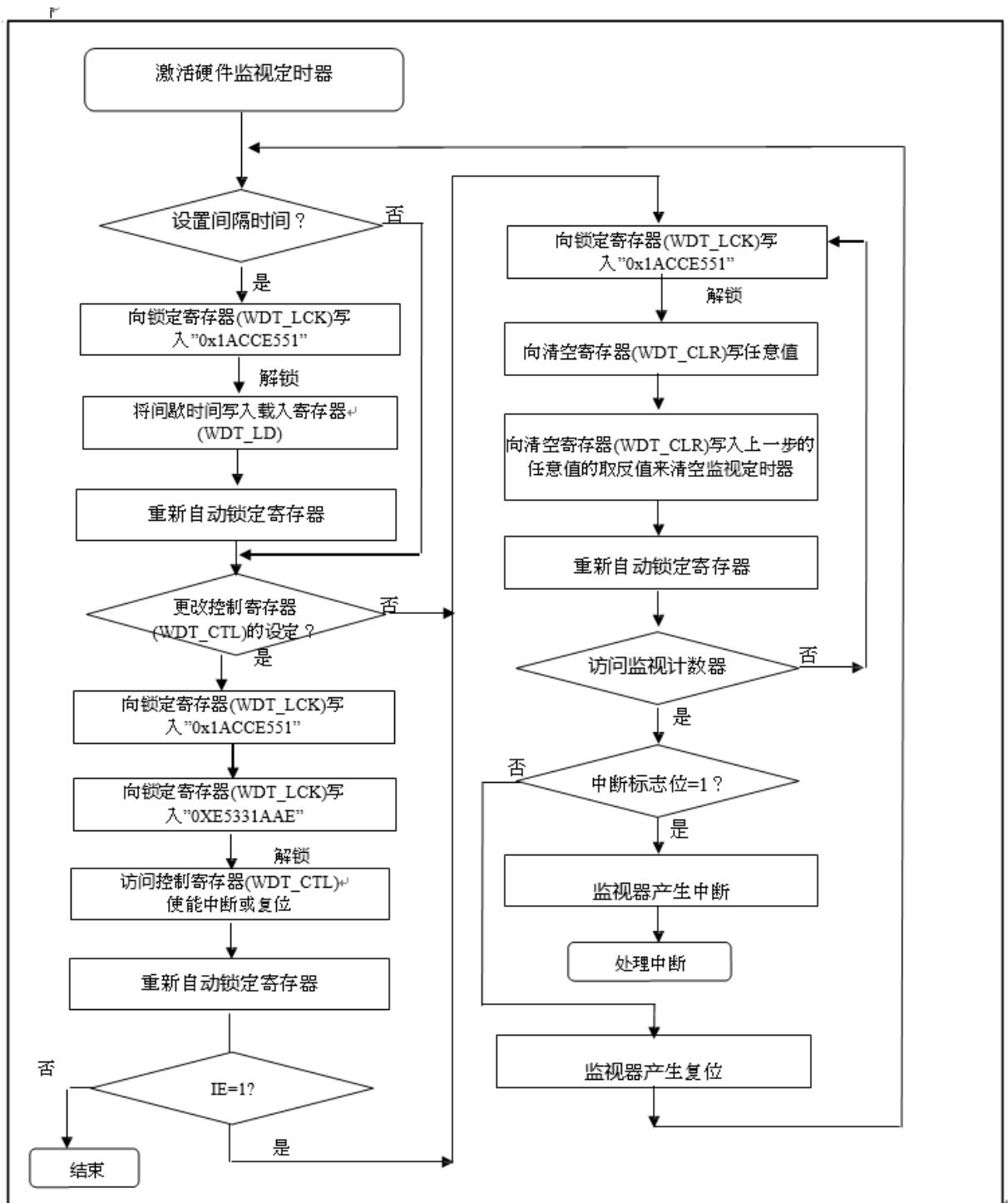


图 19.4-2 软件监视定时器的设定顺序示例 2



硬件监视定时器

图 19.4-3 硬件监视定时器的设定顺序示例

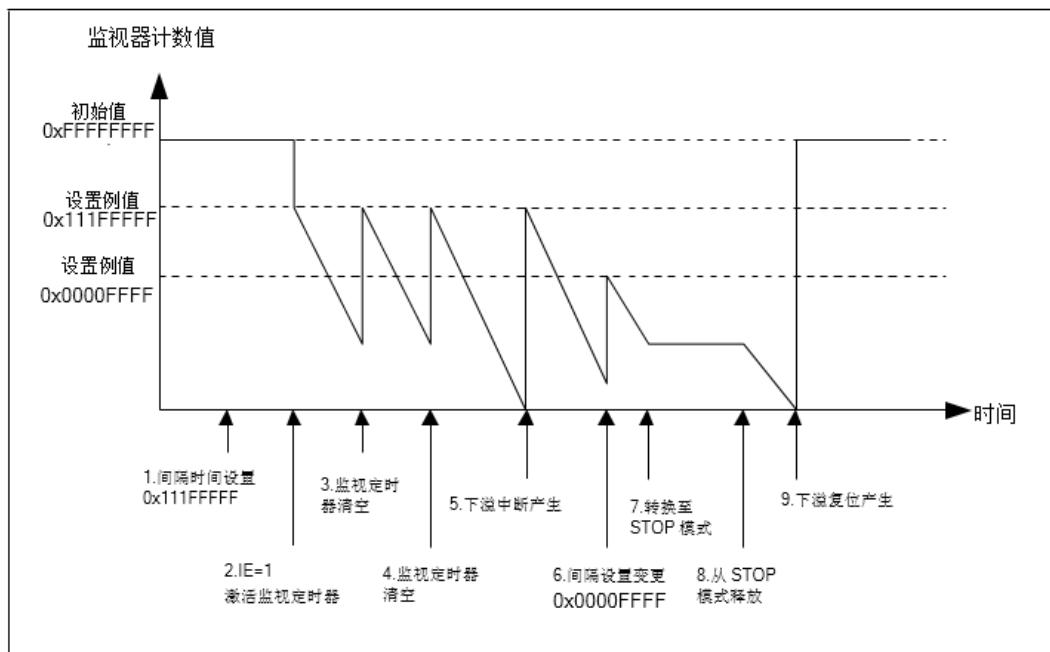


19.5 操作示例

本节是监视定时器的操作示例

软件监视定时器

图 19.5-1 软件监视定时器的操作示例

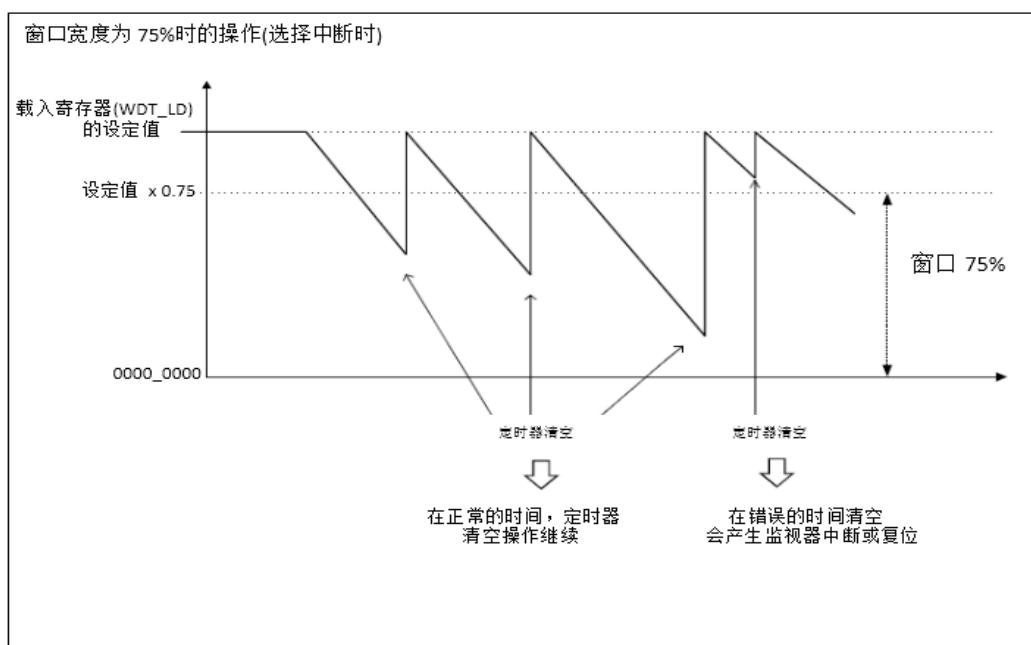


- 1) 启动前，设定 SWDG_CTRL, WDT_CTL。写 WDT_LD 寄存器，设定间隔时间。未激活监视定时器则不能反映间隔时间，此时计数值为初始值。
- 2) 访问 WDT_CTL 寄存器，写 IE=1，监视器启动。此时，可以反映间隔时间，从 1.处设定值开始递减计数。
- 3) 访问 WDT_INTCLR 寄存器并写任意值，可清零监视计数器。此时，设定值是 2.处的设定值。
- 4) 访问 WDT_INTCLR 寄存器并写任意值，可清零监视计数器。此时，设定值是 2.中的设定值。
- 5) 计数器不清零时，则产生下溢中断。此时，递减计数器设定值是 2.处的设定值。
- 6) 访问 WDT_LD 寄存器可变更间隔时间。此时，递减计数值清零到设定值。
- 7) 切换到 STOP 模式，软件监视器停止。
- 8) 退出停止模式，恢复递减计数，计数值不清零。
注意：振荡稳定等待结束后恢复递减计数，基础时钟启动。
- 9) WDT_INTCLR 寄存器的中断标志不清零并且发生第二次下溢时，产生软件监视器复位。因为发生复位，软件监视定时器停止操作。

注意：

- 访问各寄存器时须解除锁定寄存器。操作示例不予以赘述。

图 19.5-2 软件监视定时器的操作示例



注意：

- 当用载入寄存器(WDT_LD) 在窗口设置范围之外改变计数值时，
 - 中断和复位的产生遵循同样的方式
 - 当计数器发生下溢，会产生中断或复位
- 1) 在激活监视定时器之前，设置 SWDG_CTRL 和监视定时器控制寄存器(WDT_CTL)，向载入寄存器(WDT_LD)写入设定值并且设置一个反向值，设置控制寄存器(WDT_CTL)的 WDM 和 TMWD 位，此时 IE 必须为”0”，由于处在未激活状态，间歇时间无反应，计数值为初始值。设置窗口监视模式控制寄存器(WDT_MDC)，并且在窗口监视的模式中设置触发方式。
 - 2) 向监视定时器控制寄存器(WDT_CTL)寄存器的 IE 位写 1，启动监视定时器，然后间歇时间会起效，从步骤 1 中的设置值开始递减计数。
 - 3) 向清空寄存器(WDT_INTCLR)写任意值以清空监视定时器，当清空后的计数值处于窗口设置值之内（如上述操作示例，小于或等于 75%），清空动作会在正常时间进行，计数值变为步骤 1 中的设定值，操作继续。
 - 4) 向清空寄存器(WDT_INTCLR)写任意值以清空监视定时器，当清空后的计数值处于窗口设置值之外（如上述操作示例，大于 75%），清空动作会在错误时间进行，监视器产生中断

/复位。

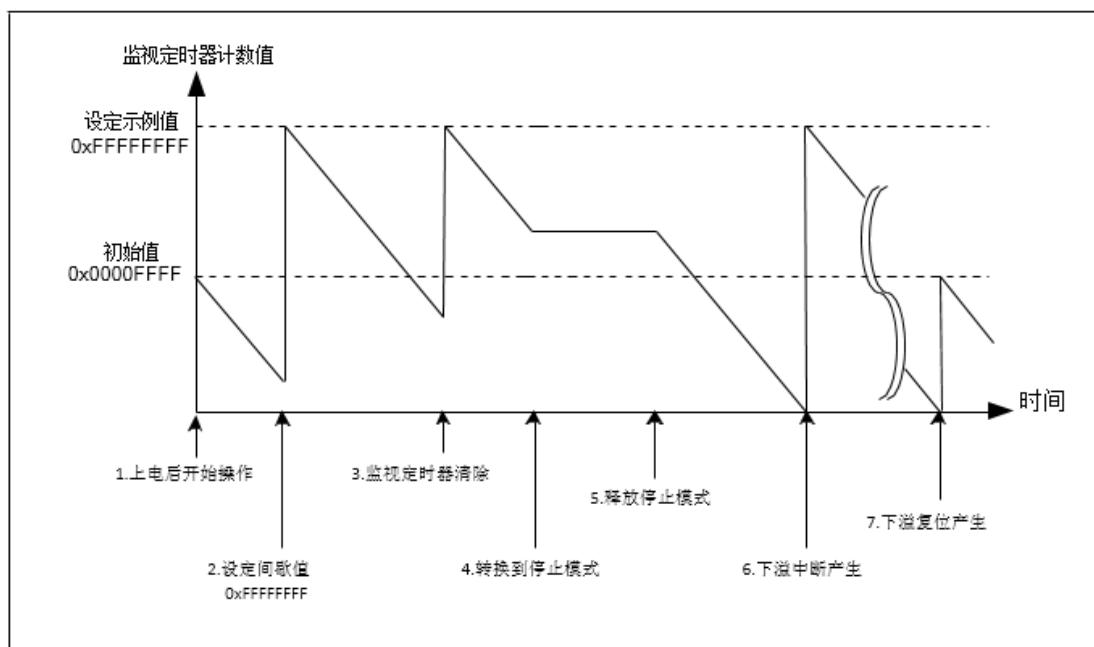
- 5) 当中断产生，监视定时器继续其操作（如上述示例）。
- 6) 当复位产生，监视定时器停止操作。

注意：

- 访问每一个寄存器时都需要释放锁定寄存器。该事项没有在上述的操作示例中提到

硬件监视定时器

图 19.5-3 硬件监视定时器的操作示例



- 1) 上电后，硬件监视定时器开始工作。计数值是初始值("0x0000FFFF")。
- 2) 写 WDT_LD 寄存器可变更间隔时间。此时，递减计数值清零到设定值。
- 3) WDT_CLR 寄存器赋任意值后写任意值的反向值，可以清零监视计数器。此时，设定值是 2 处的值。
- 4) 切换到停止模式。硬件监视定时器停止工作。
- 5) 退出停止模式。递减计数器恢复工作。不会清零计数值。

注意：LICR/LSXT 开始振荡时，输入 HWDG 时钟，恢复递减计数。

- 6) 不清除清零计数器时，则发生下溢中断。此时，设定值是 2 处的值。
- 7) WDT_CLR 寄存器的中断标志不清零并且发生第二次下溢时，产生硬件监视器复位。计数值返回到初始值，恢复递减计数。

注意：

- 访问各寄存器时，须解锁寄存器。操作示例不予以赘述。

19.6 寄存器一览

本节是监视定时器寄存器一览。

表 19.6-1 监视定时器寄存器一览

寄存器名称	说明	参考
WDT_LD	软件监视定时器载入寄存器	19.6.1
WDT_VL	软件监视定时器数值寄存器	19.6.2
WDT_CTL	软件监视定时器控制寄存器	19.6.3
WDT_INTCLR	软件监视定时器清零寄存器	19.6.4
WDT_STA	软件监视定时器中断状态寄存器	19.6.5
WDT_MDC	软件监视定时器窗口模式控制寄存器	19.6.6
WDT_LCK	软件监视定时器锁定寄存器	19.6.7
WDT_LD	硬件监视定时器载入寄存器	19.6.8
WDT_VL	硬件监视定时器数值寄存器	19.6.9
WDT_CTL	硬件监视定时器控制寄存器	19.6.10
WDT_CLR	硬件监视定时器清零寄存器	19.6.11
WDT_STA	硬件监视定时器中断状态寄存器	19.6.12
WDT_LCK	硬件监视定时器锁定寄存器	19.6.13

19.6.1 软件监视定时器载入寄存器(WDT_LD)

WDT_LD 寄存器用于设定软件监视定时器的周期。

寄存器配置

位	31	0
字段名	WDT_LD	
属性	R/W	
初值	0xFFFFFFFF	

寄存器功能

[bit31:0] WDT_LD: 间隔时间周期设定位

bit31:0	操作说明
写	设定软件监视定时器的周期。 初始值是"0xFFFFFFFF"。 写值最小为"1"。 写"0"时，发生中断。(根据设定可能会发生复位。)
读	读取设定值。 读取的初始值为"0xFFFFFFFF"。

注意：

- 监视定时器使能时若变更 WDT_LD 的值，WDT_LD 值反映到定时器计数器上，继续计数。
- 监视定时器停止时若变更 WDT_LD 的值，WDT_LD 值在监视定时器启动时反映到定时器计数器上。

19.6.2 软件监视定时器数值寄存器(WDT_VL)

WDT_VL 寄存器用于读出软件监视定时器当前的计数器值。

寄存器配置

位	31	0
字段名	WDT_VL	
属性	R	
初值	0xFFFFFFFF	

寄存器功能

[bit31:0] WDT_VL：计数器值位

bit31:0	说明
写时	无影响。
读时	读出当前监视计数器的计数值。启动前读取时，初始值为"0xFFFFFFFF"。

注意：

- 关于工具断点时监视定时器的设定，参考“时钟”章节的“调试断点监视定时器控制寄存器(DBWDT_CTRL)”。

19.6.3 软件监视定时器控制寄存器(WDT_CTL)

WDT_CTL 寄存器用于使能/禁止软件监视定时器的设定。

寄存器配置

bit	7	6	5	4	3	2	1	0
Field	保留位		WDM	TMWD		RSTE N	IE	
属性	-		R/W	R/W		R/W	R/W	
初始值	-		0	0	0	0	0	

寄存器功能

[bit7:5]保留位 这些位读"0b000"。 写时设定至"0b000"。

[bit4] WDM: 软件监视窗口模式使能位

bit	说明	
读	读出寄存器值	
写	0	窗口监视模式关闭
	1	窗口监视模式使能

注意：

- 当 IE="1"时，写此位无效。当 IE="0"时，此位可写
- 当 IE="1"时，可以同时将 IE 位与此位同时写
- 当 IE="0"时，同时写 IE 位与此位无效

[bit3:2] TMWD: 软件监视时间窗口设置位

bit	说明	
读	读出寄存器值	
写	00	处于 WDT_LD 周期之内或更小，重载被使能
	01	处于 WDT_LD 周期的 75%之内或更小，重载被使能
	10	处于 WDT_LD 周期的 50%之内或更小，重载被使能
	11	处于 WDT_LD 周期的 25%之内或更小，重载被使能

注意：

- 只有当 WDM="0"时，此位才有效
- 当 IE="1"时，此位不可写；当 IE="0"时，此位可写
- 当 IE="1"时，可以同时将 IE 位与此位同时写
- 当 IE="0"时，同时写 IE 位与此位无效

[bit1] RSTEN: 软件监视器复位使能位

bit	说明
读时	读出寄存器的值。
写"0"时	监视器复位禁止
写"1"时	监视器复位使能

注意：

- 当 WDM="1"时，此位设置无效

[bit0] IE: 软件监视定时器中断和计数器使能位

bit	说明
读时	读出寄存器的值。
写"0"时	监视器中断禁止 监视器计数禁止
写"1"时	监视器中断使能 监视器计数器使能

注意：

- IE 写"1"，监视器计数器载入 WDT_LD 的间隔周期值，软件监视定时器启动。
- IE 写"0"，监视器计数器停止操作。再次写"1"时，监视器计数器载入 WDT_LD 的周期值，再次启动。
- 仅使能 IE 可以启动监视定时器。仅使能 RSTEN，不能启动监视定时器。要启动监视定时器，须使 IE 有效。参见"3. 操作说明"。
- IE 写"0"可清零软件监视器中断状态寄存器(WDT_STA)的中断标志。

19.6.4 软件监视定时器清空寄存器(WDT_INTCLR)

WDT_INTCLR 寄存器用于清零软件监视定时器。

寄存器配置

位	31	0
字段名	WDT_INTCLR	
属性	R/W	
初值	0xFFFFFFFF	

寄存器功能

[bit31:0] WDT_INTCLR: 清零位

bit31:0	说明
读	读值不定。
写	写任意值 • 若监视定时器发生中断，则会清零监视定时器的中断。 • WDT_LD 寄存器的设定值重载到监视定时器计数器。

19.6.5 软件监视定时器中断状态寄存器(WDT_STA)

WDT_STA 寄存器用于指示软件监视定时器的中断状态。

寄存器配置

bit	7	1	0
Field	保留		STA
属性	-		
初始值	-		R 1'b0

寄存器功能

[bit7:1] 保留位 这些位读"0b00000000"。写时，设定"0b00000000"。

[bit0] STA: 软件监视器中断状态位

bit	说明
写	无效
读"0"	不发生监视器中断
读"1"	发生监视器中断

19.6.6 软件监视定时器窗口监视模式控制寄存器(WDT_MDC)

WDT_MDC 寄存器用于控制软件监视定时器的窗口监视模式。

寄存器配置

Bit	7		1	0
Field	保留位			TRG
属性	-			R/W
初始值	-			0

寄存器功能

[bit7:1] 保留位 这些位读"0b00000000"。写时，设定"0b00000000"。

[bit0] TRG: 软件监视定时器触发模式位

bit	说明	
读	读出寄存器值	
写	0	在下述条件下，会产生中断： - 当计数器发生下溢时 - 在窗口时间之外发生计数器清零(写WDT_INTCLR)时 - 在窗口时间之外发生计数器重载(写 WDT_LD)时
	1	在下述条件下，会产生复位： - 当计数器发生下溢时 - 在窗口时间之外发生计数器清零(写WDT_INTCLR)时 - 在窗口时间之外发生计数器重载(写 WDT_LD)时

注意：

- 只有当 WDM="1"时此位有效。
- 当设置 TRG="1"时，无论 WDT_CTL 寄存器的 RSTEN 位状态如何，都会产生复位。

19.6.7 软件监视定时器锁定寄存器(WDT_LCK)

WDT_LCK 寄存器用于控制对软件监视定时器所有寄存器的访问。

寄存器配置

位	31	0
字段名	WDT_LCK	
属性	R/W	
初值	0x00000000	

寄存器功能

[bit31:0] WDT_LCK: 软件监视定时器锁定寄存器

bit31:0	说明
写	写"0x1ACCE551"时: 软件监视定时器的所有寄存器解锁 写"0x1ACCE551"以外的值时: 软件监视定时器的所有寄存器的锁定有效
读	0x00000000: 解锁 0x00000001: 不解锁

注意：

- 初始值时，锁定无效。请在软件监视定时器启动后使能锁定。
- 解锁后，可访问清零寄存器(WDT_INTCLR)。访问清零寄存器(WDT_INTCLR)后，闩锁并非自动使能。请在清零顺序上加入以下步骤：解锁→清零→使能锁定的步骤。
- 未解锁的状态下访问软件监视定时器器的各寄存器时，可读出各寄存器的值，写无效。

19.6.8 硬件监视定时器载入寄存器(WDT_LD)

WDT_LD 寄存器用于设定硬件监视定时器的周期

寄存器配置

位	31	0
字段名	WDT_LD	
属性	R/W	
初值	0x0000FFFF	

寄存器功能

[bit31:0] WDT_LD: 间隔周期设定位

bit31:0	说明
写	设定硬件监视定时器的周期。 初始值是"0x0000FFFF"。 写值最小为"1"。 写"0"产生中断
读	可读取设定值。初始值读"0x0000FFFF"

注意：

- 监视定时器使能时若变更 WDT_LD 的值， WDT_LD 的值反映到定时器计数器，继续进行计数。
- 监视定时器停止时若变更 WDT_LD 的值， WDT_LD 值在监视定时器启动时反映到定时器计数器上。
- 软件复位和软件监视器复位不能清零该寄存器。
- 当监视定时器中断发生时修改 WDT_LD 的值，则监视定时器中断会被清除。

19.6.9 硬件监视定时器数值寄存器(WDT_VL)

WDT_VL 寄存器用于读出硬件监视定时器当前的计数器值。

寄存器配置

位	31	0
字段名	WDT_VL	
属性	R	
初值	0xFFFFFFFF	

寄存器功能

[bit31:0] WDT_VL：计数器值位

bit31:0	说明
读	可读出监视定时器计数器的当前计数值。 上电后，硬件监视定时器自动启动，所以读时递减计数已经开始。读出的值是上电后的值或复位后从初始值"0x0000FFFF"递减的值。
写	无影响

注意：

- 软件复位和软件监视器复位不能清零该寄存器。
- 本寄存器仅在工具断点期间监视定时器停止时可读取正确的值。关于工具断点时的监视定时器的设定，参考"时钟"一章的"调试断点监视定时器控制寄存器"。工具断点以外的动作时的读取值对于总线时钟是异步读取，有可能读取值不正确。此时，须采取读取 2 次后做比较等应对措施。

19.6.10 硬件监视定时器控制寄存器(WDT_CTL)

WDT_CTL 寄存器可使能/禁止设定硬件监视定时器。

寄存器配置

bit	7	6	5	4	3	2	1	0
Field		保留位				CLKS	RSTEN	IE
属性		-				R/W	R/W	R/W
初始值		-				0	1	1

寄存器功能

[bit7:3]保留位 这些位读"0b00000"。 写时，设定"0b00000"。

[bit2] CLKS: 计数时钟选择位

bit	说明
读	读出寄存器的值。
写"0"	计数时钟来自外部低速晶振(LSXT)
写"1"	计数时钟来自内部低速 RC(LICR)

[bit1] RSTEN: 硬件监视器复位使能位

bit	说明
读	读出寄存器的值。
写"0"	禁止监视器复位
写"1"	使能监视器复位

[bit0] IE: 硬件监视器中断和计数器使能位

bit	说明
读	读出寄存器的值。
写"0"	禁止监视器中断 禁止监视器计数器
写"1"	使能监视器中断 使能监视器计数器

注意：

- IE 写"0"可停止监视器计数器的操作。再次写"1"时，监视器计数器重载 WDT_LD 的周期值，计数器启动。
- 仅使能 IE 可启动监视定时器。仅使能 RSTEN 不能启动监视定时器。要使能监视定时器启动须使能 IE。
- 要访问本寄存器，锁定寄存器须写"0x1ACCE551"；并写反向值"0xE5331AAE"解锁。
- 软件复位和软件监视器复位不能清零该寄存器。
- IE 写"0"可清零硬件监视定时器中断状态寄存器(WDT_STA)的中断标志。

19.6.11 硬件监视定时器清空寄存器(WDT_CLR)

WDT_CLR 寄存器用于清零硬件监视定时器。

寄存器配置

位	7		0
字段名	WDT_CLR		
属性	R/W		
初值	0xXX		

寄存器功能

[bit7:0] WDT_CLR: 清零位

bit7:0	说明
读	读值不定
写	写任意 8 位值后，接着写任意值的反向值 - 监视定时器发生中断时，清零中断 - WDT_LD 寄存器的设定值重载到监视定时器计数器

注意：

- 软件复位和软件监视器复位不能清零该寄存器。

19.6.12 硬件监视定时器中断状态寄存器(WDT_STA)

WDT_STA 寄存器用于指示硬件监视定时器的中断状态。

寄存器配置

bit	7	1	0
Field	保留		
属性	-		
初始值	R 1'b0		

寄存器功能

[bit7:1] 保留位 这些位读"0b00000000"。 写时，设定"0b00000000"。

[bit0] STA: 硬件监视器中断状态位

bit	说明
写	无影响
读"0"	不发生硬件监视器中断
读"1"	发生硬件监视器中断

注意：

- 软件复位和软件监视器复位不能清零该寄存器。

19.6.13 硬件监视定时器锁定寄存器(WDT_LCK)

WDT_LCK 寄存器用于控制对硬件监视定时器所有寄存器的访问。

寄存器配置

位	31	0
字段名	WDT_LCK	
属性	R/W	
初值	0x00000001	

寄存器功能

[bit31:0] WDT_LCK: 硬件监视定时器锁定寄存器

bit31:0	说明
写	写"0x1ACCE551"时: 解锁控制寄存器以外所有的寄存器。 再写反向值"0xE5331AAE"时: 解锁所有的寄存器。 执行上述以外顺序或写其它值时: 所有寄存器锁定
读	0x00000000: 解锁 0x00000001: 不解锁

注意：

- 本寄存器不能被软件复位和软件监视器复位。
- 不能解锁时若访问硬件监视器的各寄存器，可读出各寄存器的值，写无效。

19.7 注意事项

本节介绍监视定时器的使用注意事项。

- 关于硬件监视定时器清零寄存器

先写任意 8 位值，后写任意值的反向值可以清零硬件监视定时器。除非写入正确的任意值的反转值，否则不能清零。即使不能清零，也可再次锁定寄存器。

- 配合使用调试工具

使用调试工具执行工具断点，根据寄存器的设定可以设定监视定时器的计数器的继续或停止。关于调试时监视定时器操作详情，参考“时钟”一章。

- 低功耗模式下的操作

进行低功耗模式设定时，须写 LPM_CTL 寄存器以免程序错误导致进入低功耗模式时监视定时器被停止，详见“低功耗模式”一章。

- 须使用复位源寄存器确认监视器复位的发生

详情参见“复位”一章中的复位源寄存器。

- 关于中断源

参见“中断”一章中的中断源寄存器。

- APB 时钟的分频时钟用作软件监视器的计数时钟

关于计数时钟的分频设定，参见“时钟”一章。

- 关于硬件监视器和中断处理

WDT_CTL 的 Lock 解锁前(WDT_CTL 以外的 Lock 解锁后)，硬件监视器使能其他中断并且转移到中断处理时，Lock 解锁次数在硬件上检测不出。

在中断处理的开始，请对 WDT_LCK 寄存器写值以锁定 Lock。

20. 双定时器

本章介绍双定时器的功能和操作。

20.1 概要

双定时器由 2 个可编程的 32/16 位递减计数器构成。计数器变为零时产生中断。

双定时器的概要

双定时器由 2 个可编程的自由运行计数器构成。每个定时器的操作相同。自由运行计数器可根据控制寄存器设定为 32 位或 16 位的计数器容量编程。另外，也可选择以下 3 种定时器模式中的一种模式进行工作。

- 自由运行模式

计数器变为零时，计数器回到最大值继续进行计数。

- 周期模式

计数器变为零时，计数器从载入寄存器重载后继续进行计数。

- 单次模式

对载入寄存器写值后，计数器载入新写的值。计数器变为零时停止计数，直到再次编程为止。

2 个自由运行计数器使用通用的定时器时钟(TMRCLK)操作。定时器时钟使用 APB 总线时钟(PCLK)。此外，各自的自由运行计数器带可 1, 16, 256 分频的预分频器。因此，使用每个预分频器，可控制各个自由运行计数器的计数速度。

写入载入寄存器后，定时器计数值载入计数器。若使能定时器计数器，定时器则以定时器时钟和预分频器设定的速度递减。定时器计数器动作中对载入寄存器写值的话，计数器则立即从新写入的值重新开始计数。

载入定时器计数的另一个方法是对背景载入寄存器写值。在这种情况下，写值后当前的计数值并不受到影响，计数器继续递减计数。等计数器变为零时，若是设定为周期模式，新载入值重载入定时器计数器。

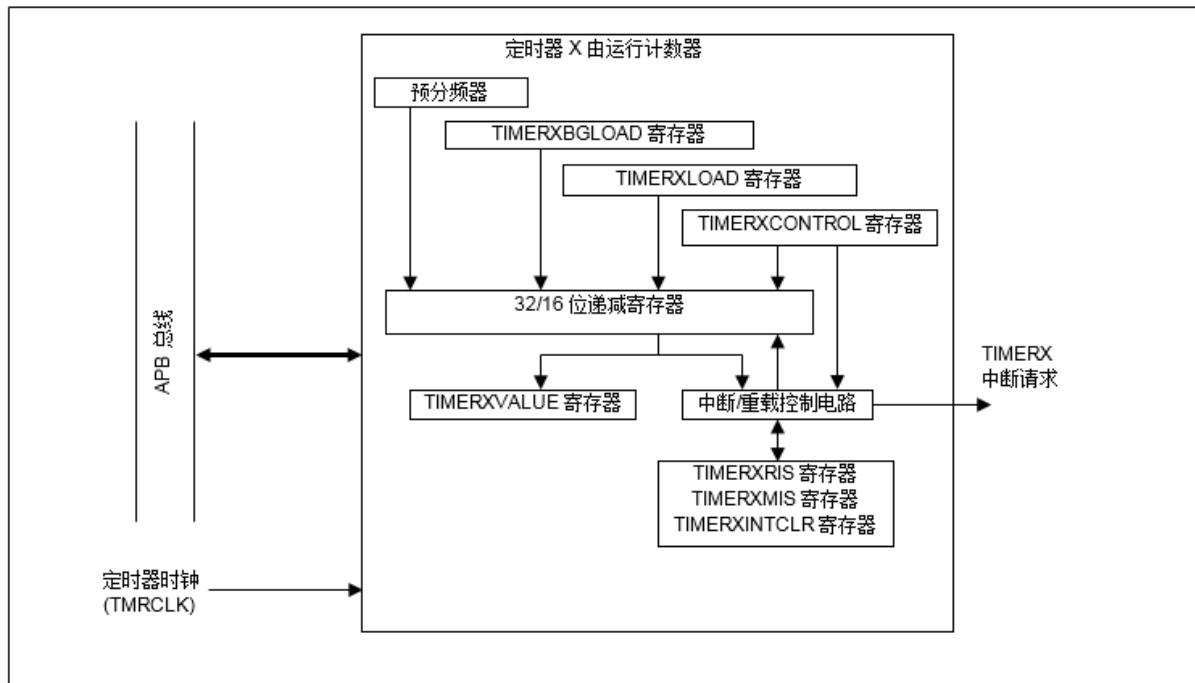
定时器计数器变为零时产生中断。在中断清除寄存器写值可清除中断。另外，可使用屏蔽中断寄存器屏蔽中断输出信号。

当前的计数值可随时从数值寄存器读取。

20.2 构成

以下介绍双定时器的构成。

图 20.2-1 双定时器的框图(X=1 或 2)



20.3 操作

以下介绍双定时器的操作。

20.3.1 定时器工作模式

工作模式可通过设定控制寄存器(TIMERXCONTROL)的模式位(TMRRMODE)和单次模式位(ONESHOT)从 3 个定时器模式中选择。

表 20.3.1-1 模式选择表

TMRRMODE	ONESHOT	选择模式
0	0	自由运行模式
1	0	周期模式
-	1	单次模式

计数器运行可通过设定控制寄存器的定时器容量位(TMRSIZE)决定 32 位或 16 位。

注意：

- 本章中寄存器名称中的“X”文字代表自由运行计数器 1 或 2。

自由运行模式

执行复位会将定时器的值初始化到 0xFFFFFFFF。之后若使能计数器，计数器则在定时器时钟(TMRCLK)的上升沿递减计数。也可在载入寄存器(TIMERXLOAD)写值，新写入的计数值可载入计数器。若使能计数器，计数器则从该载入值开始递减计数。

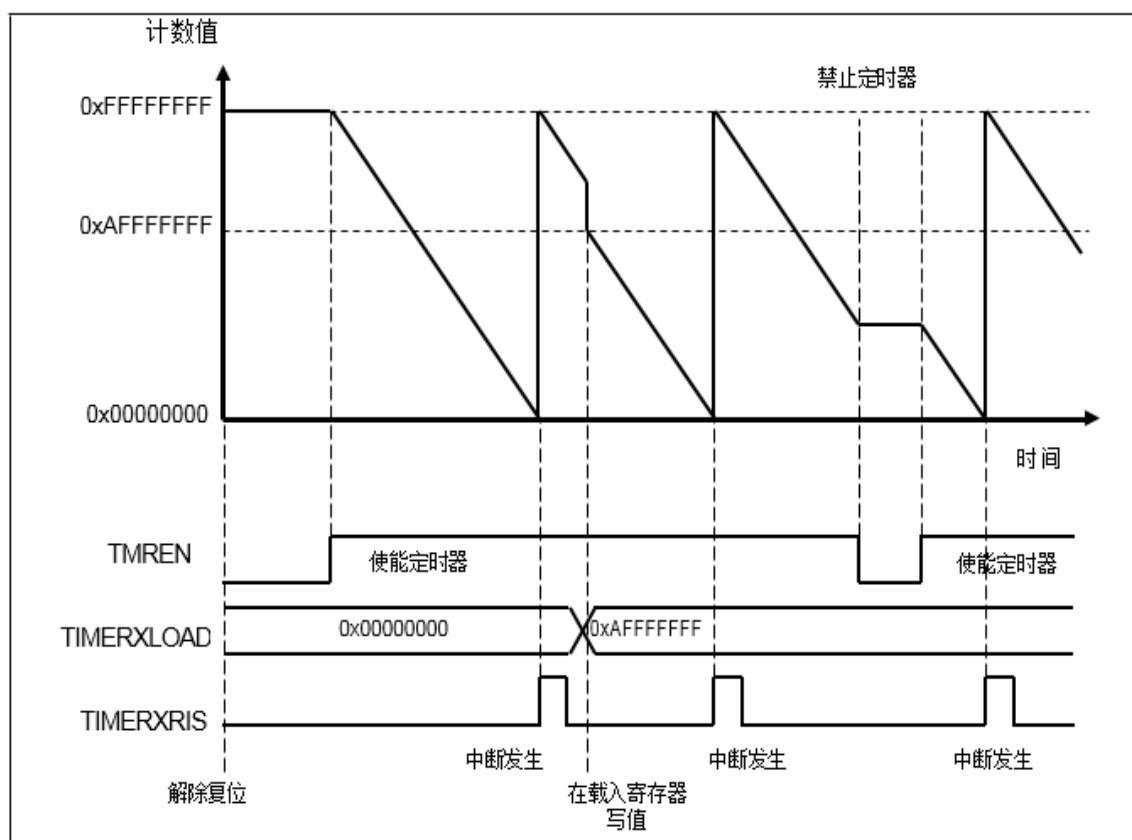
32 位模式下，计数变为零(0x00000000)时发生中断。然后无论载入寄存器的值是多少，计数值回到 0xFFFFFFFF。计数器再次开始递减计数，只要计数器处于使能状态，这一周期反复操作。

16 位模式下，仅计数器的低 16 位递减计数。计数变为 0x0000 时发生中断。然后无论载入寄存器的值是多少，计数值回到 0xFFFF。

如果清零控制寄存器(TIMERXCONTROL)的使能位(TMREN)，计数器变为禁止状态的话，计数器则停止并保持当前值。再次使能计数器时，计数器从当前值开始继续递减计数。

计数值可随时从数值寄存器(TIMERXVALUE)读取。

图 20.3.1-1 自由运行模式的操作(32 位模式)



周期模式

在载入寄存器(TIMERXLOAD)写入计数器初始值，该初始值载入计数器。若使能计数器，计数器则从该值开始递减计数。

32 位模式下，计数器的 32 位全部递减计数，计数变为零(0x00000000)时发生中断。然后计数器重新载入载入寄存器的值。计数器再次开始递减计数，只要计数器处于使能状态，这一周期反复操作。

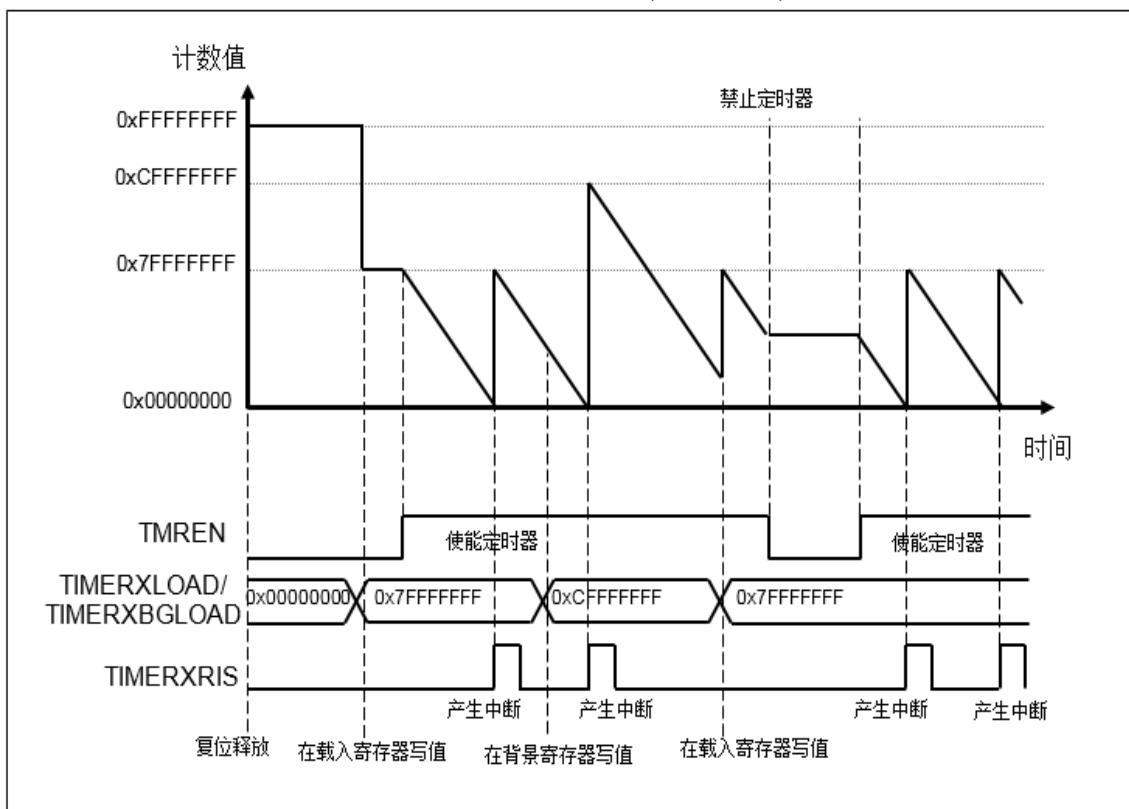
16 位模式下，仅计数器的低 16 位递减计数。计数变为 0x0000 时发生中断。然后计数器重新载入载入寄存器的值。计数器再次开始递减计数，只要计数器处于使能状态，这一周期反复操作。

计数器运行中若在背景载入寄存器(TIMERXBGLOAD)写入新值，则载入寄存器也更新为同一数值，但是计数器继续向零的方向递减计数。等计数器变为零时，再载入新值。只要定时器的设定是周期模式，在之后的每次重载时都使用该新载入值。

计数器运行中在载入寄存器写值，新写值会载入计数器，计数器值在下一个定时器时钟变为新载入的值。

如果清零控制寄存器(TIMERXCONTROL)的使能位(TMREN)，计数器变为禁止状态的话，计数器停止计数并保持当前值。再次使能计数器时，计数器从当前值开始继续递减计数。

图 20.3.1-2 周期模式的操作(32 位模式)



单次模式

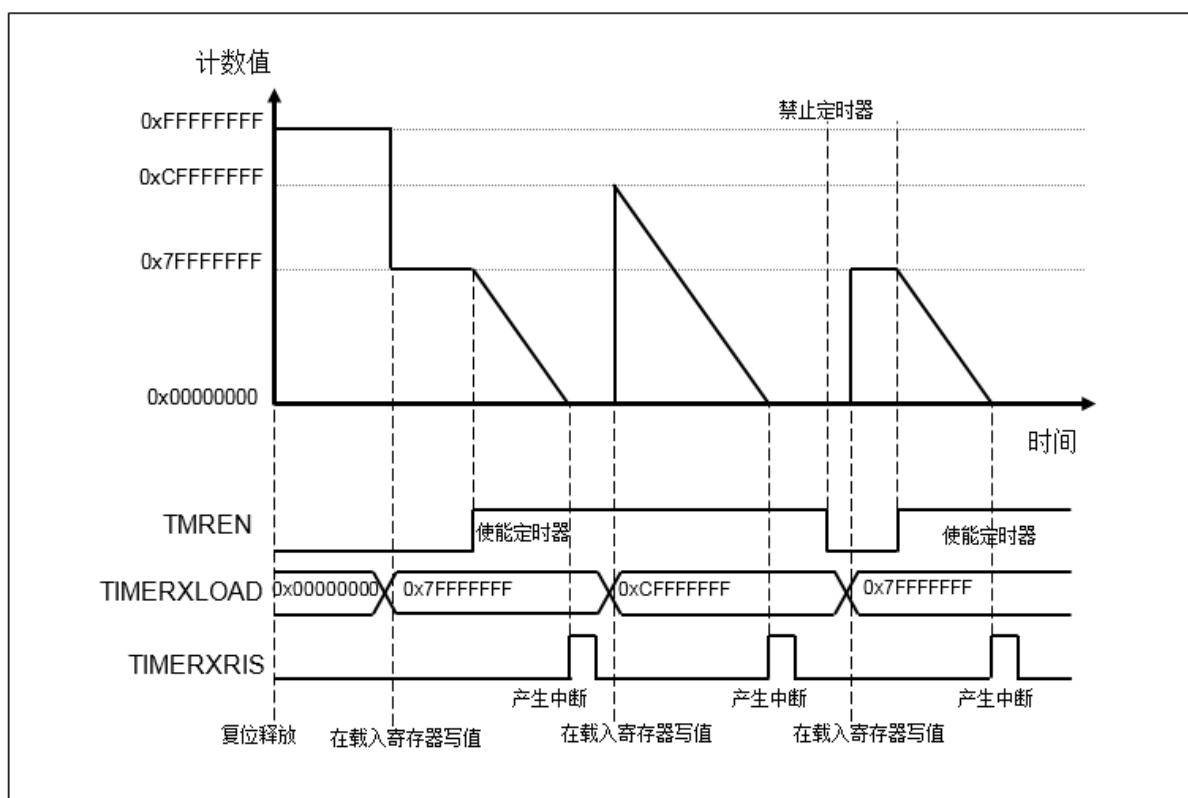
单次模式下，计数器以递减顺序开始计数，载入值写入载入寄存器(TIMERXLOAD)。若使能计数器的话，计数器从该值开始递减计数。

32 位模式下，计数器的 32 位全部递减计数，计数变为零(0x00000000)时发生中断。然后计数器停止运行。

16 位模式下，仅计数器的低 16 位递减计数。计数变为 0x0000 时发生中断。然后计数器停止动作。

单次模式下，只要在载入寄存器写入新值，计数器就可再次启动。计数器值在下一个定时器时钟变为新载入的值。

图 20.3.1-3 单次模式的操作(32 位模式)



20.3.2 初始状态

复位后，定时器初始化到以下状态。

- 禁止定时器计数器
- 自由运行模式选择
- 16 位计数器模式选择
- 预分频器 1 分频设定
- 中断清除、且中断使能状态
- 载入寄存器设定值为 0
- 计数器值设定为 0xFFFFFFFF

20.3.3 中断操作

以下介绍中断操作。

使能中断(INTEN=1)的条件下，计数器变为 0x00000000(32 位模式时)、或 0xXXXX0000(16 位模式时)时发生中断。16 位模式时计数器的高 16 位被忽略。

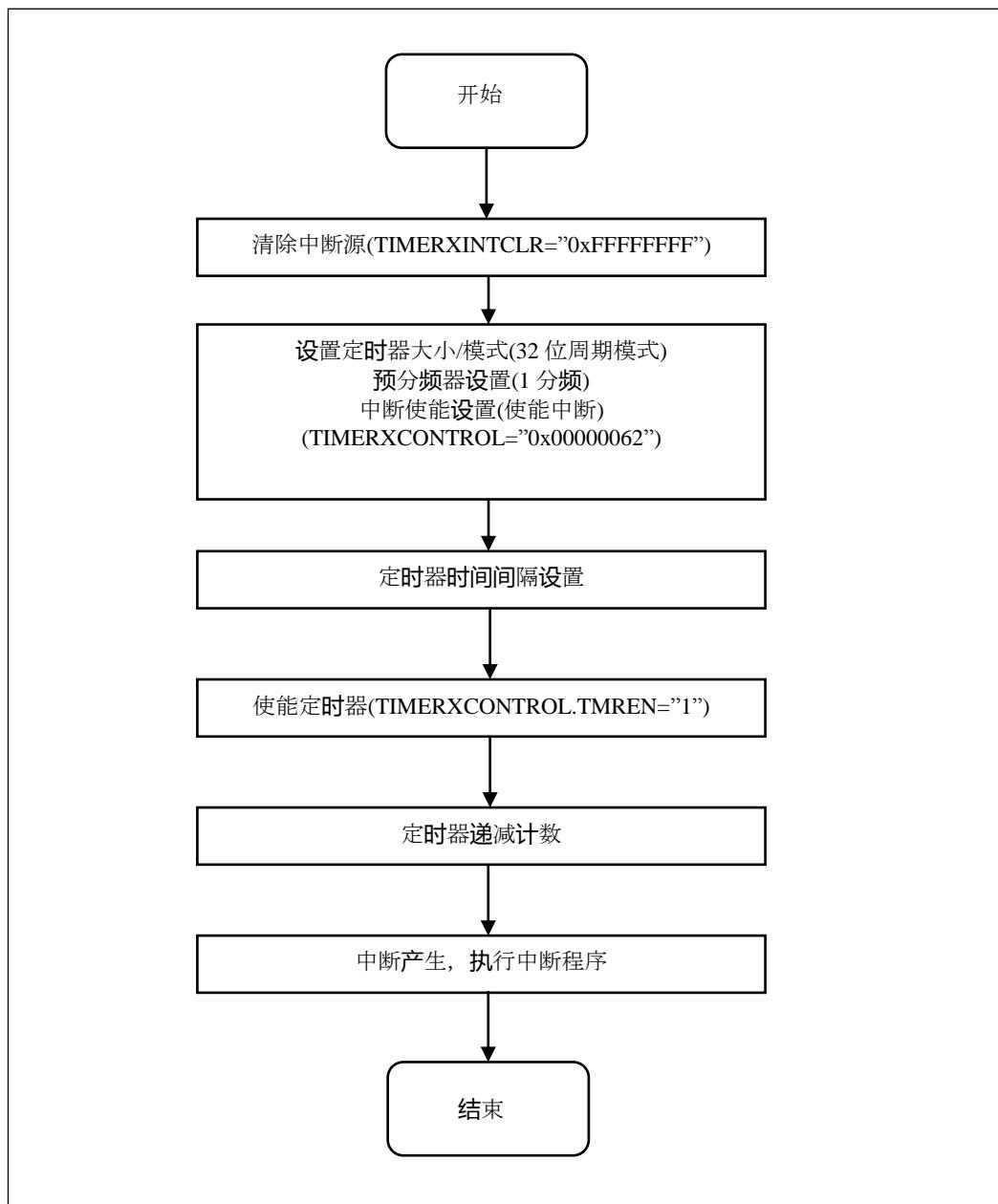
在中断清除寄存器(TIMERXINTCLR)写值可清除中断。定时器发生的中断信号可通过控制寄存器的中断许可位(INTEN)清 0 进行屏蔽。屏蔽前的实际中断状态可从中断状态寄存器(TIMERXRIS)读取。另外，被屏蔽的状态可从屏蔽中断状态寄存器(TIMERXMIS)读取。

20.4 设定步骤例

以下介绍双定时器的设定步骤范例。

双定时器设定步骤流程

图 20.4-1 周期模式设定步骤例



定时器时间间隔设定

各模式下定时器间隔的计算公式如表 20.4-1 所示。

表 20.4-1 定时器间隔计算式

模式	定时器间隔
自由运行32位	$(\text{PRESCALEDIV} / \text{TMRCLKFREQ}) \times 2^{32}$
自由运行16位	$(\text{PRESCALEDIV} / \text{TMRCLKFREQ}) \times 2^{16}$
周期&单次	$(\text{PRESCALEDIV} / \text{TMRCLKFREQ}) \times \text{TIMERXLOAD} + 1$

- TMRCLKFREQ 是定时器时钟(TMRCLK)的频率。
- PRESCALEDIV 是控制寄存器(TIMERXCONTROL)的位[3:2]设定的 1、16 或 256 的预分频器分频值。
- TIMERXLOAD 是载入寄存器(TIMERXLOAD)的值。

例如，TMRCLKFRAQ=40MHz, PRESCALEDIV=1 时，可如下所示计算设定 1ms 定时器间隔时的载入寄存器(TIMERXLOAD)的值。

$$\begin{aligned}\text{TIMERXLOAD} &= \text{定时器间隔} \times \text{TMRCLKFREQ} / \text{PRESCALEDIV} - 1 \\ &= 1\text{ms} \times 40\text{MHz} / 1 - 1 = 4 \times 10^4 - 1 = 0x00009C3F\end{aligned}$$

注意：

- 载入寄存器(TIMERXLOAD)的最小有效值为"1"。若将载入寄存器设定为"0"，立即发生中断。

20.5 寄存器

以下介绍双定时器使用的寄存器的构成和功能。

双定时器的寄存器一览：

寄存器简称	寄存器名称	参照章节
TIMER1LOAD	定时器 1 载入寄存器	20.5.1
TIMER1VALUE	定时器 1 数值寄存器	20.5.2
TIMER1CONTROL	定时器 1 控制寄存器	20.5.3
TIMER1INTCLR	定时器 1 中断清除寄存器	20.5.4
TIMER1RIS	定时器 1 中断状态寄存器	20.5.5
TIMER1MIS	定时器 1 屏蔽中断状态寄存器	20.5.6
TIMER1BGLOAD	定时器 1 背景载入寄存器	20.5.7
TIMER2LOAD	定时器 2 载入寄存器	20.5.1
TIMER2VALUE	定时器 2 数值寄存器	20.5.2
TIMER2CONTROL	定时器 2 控制寄存器	20.5.3
TIMER2INTCLR	定时器 2 中断清除寄存器	20.5.4
TIMER2RIS	定时器 2 中断状态寄存器	20.5.5
TIMER2MIS	定时器 2 屏蔽中断状态寄存器	20.5.6
TIMER2BGLOAD	定时器 2 背景载入寄存器	20.5.7

20.5.1 载入寄存器(TIMERXLOAD) X=1 或 2

载入寄存器(TIMERXLOAD)是 32 位寄存器，用于放入计数器的递减开始值。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TIMERXLOAD[31:16]															
属性	R/W															
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TIMERXLOAD[15:0]															
属性	R/W															
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[bit31:0] TIMERXLOAD: 定时器 X 载入位

在本寄存器直接写值后，计数器在下一个定时器时钟设定为新值。另外，如果设定是周期模式，当前的计数值变为零时，重载该值使用。

在背景寄存器(TIMERXBGLOAD)写值后，本寄存器虽然也被覆盖，但当前的计数并不受到影响。在载入寄存器或背景寄存器中的一个写值后，读取时总是读最后写入的寄存器值。也就是说，从载入寄存器和背景寄存器读取的是同样的值，该值是周期模式下计数器变为零时重载的值。

注意：

- 载入寄存器(TIMERXLOAD)的最小有效值为"1"。若将载入寄存器设定为"0"，立即发生中断。

20.5.2 数值寄存器(TIMERXVALUE) X=1 或 2

数值寄存器(TIMERXVALUE)是 32 位只读寄存器，用于显示递减计数器的当前值。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TIMERXVALUE[31:16]															
属性	R															
初始值	0xFFFF															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TIMERXVALUE[15:0]															
属性	R															
初始值	0xFFFF															

[bit31:0] TIMERXVALUE：定时器 X 数值位

在载入寄存器(TIMERXLOAD)写入新的载入值被载入后，本数值寄存器(TIMERXVALUE)立即反映出新的载入值。

注意：

- 16 位定时器模式时，数值寄存器(TIMERXVALUE)32 位的高 16 位不会设定为自动清 "0"。例如，定时器从 32 位模式变更到 16 位模式后，如果未发生对载入寄存器(TIMERXLOAD)的写入，数值寄存器的高 16 位残留的值并非是零。

20.5.3 控制寄存器(TIMERXCONTROL) X=1 或 2

控制寄存器(TIMERXCONTROL)用于控制定时器。

bit	31																												8
Field	Reserved																												
属性	-																												
初始值	0xXXXXXX																												

bit	7	6	5	4	3	2	1	0
Field	TMREN	TMRMOD	INTE	N	Reserved	TMRPRE	TMRSIZE	ONESHOT
属性	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
初始值	0	0	1	0	00	0	0	0

[bit31:8] Reserved: 保留位

写入时	无效
读取时	值未定

[bit7] TMREN: 使能位

bit	说明
0	禁止定时器 (初始值)
1	使能定时器

[bit6] TMRMODE : 模式位

bit	说明
0	自由运行模式 (初始值)
1	周期模式

[bit5] INTEN : 中断许可位

bit	说明
0	禁止中断
1	使能中断 (初始值)

[bit4] Reserved : 保留位

写入时	无效
读取时	值未定

[bit3:2] TMRPRE: 预分频器位

bit3	bit2	说明
------	------	----

0	0	时钟 1 分频 (初始值)
0	1	时钟 16 分频
1	0	时钟 256 分频
1	1	未定义，禁止使用

[bit1] TMRSIZE: 计数器容量位, 该用于选择 16/32 位计数器操作。

bit	说明
0	16 位计数器 (初始值)
1	32 位计数器

[bit0] ONESHOT: 单次模式位, 该位用于选择单次模式还是计数器返回模式(自由运行模式/周期模式)。可使用模式位(TMRMODE)进行自由运行模式还是周期模式的设定。

bit	说明
0	返回模式(自由运行模式/周期模式) (初始值)
1	单次模式

注意:

- 定时器运行中, 禁止变更计数器模式、容量和预分频器设定。如需变更设定, 先禁止定时器, 然后在各寄存器写入新的设定值。设定变更完成后, 再次使能定时器。务必按照此步骤进行设定变更, 否则器件运行可能会发生意外。

20.5.4 中断清除寄存器(TIMERXINTCLR) X=1 或 2

中断清除寄存器(TIMERXINTCLR)用于清除中断。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TIMERXINTCLR[31:16]															
属性	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
初始值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TIMERXINTCLR[15:0]															
属性	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
初始值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[bit31:0] TIMERXINTCLR: 中断清除位

在本寄存器写入任意值，都可清除自计数器的中断输出。

20.5.5 中断状态寄存器(TIMERXRIS) X=1 或 2

中断状态寄存器(TIMERXRIS)用于显示未被屏蔽的中断状态。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
属性	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值	0xXXXX															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Field	Reserved												TMRXINTST							
属性	-																			
初始值	0xXXXX																			

[bit31:1] Reserved: 保留位

写入时	无效
读取时	值未定

[bit0] TMRXINTST: 中断状态寄存器位

bit	说明
0	未发生自计数器的中断(初始值)
1	发生自计数器的中断

20.5.6 屏蔽中断状态寄存器(TIMERXMIS) X=1 或 2

屏蔽中断状态寄存器(TIMERXMIS)用于显示被屏蔽的中断状态。

bit	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6
Field	Reserved															
属性	-															
初始值	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															TMRXINTST
属性	-															R
初始值	0xXXXX															0

[bit31:1] Reserved: 保留位

写入时	无效
读取时	值未定

[bit0] TMRXMINTST: 屏蔽中断状态位, 本位是中断状态和控制寄存器(TIMERXCONTROL)的定时器中断许可位的理论“与”值。与本位相同的值连接到中断输出信号。

bit	说明
0	未发生自计数器的中断 (初始值)
1	发生自计数器的中断

20.5.7 背景载入寄存器(TIMERXBGLOAD) X=1 或 2

背景载入寄存器(TIMERXBGLOAD)是 32 位寄存器，用于载入递减计数开始值。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	TIMERXBGLOAD[31:16]															
属性	R/W															
初始值	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TIMERXBGLOAD[15:0]															
属性	R/W															
初始值	0x0000															

[bit31:0] TIMERXBGLOAD: 背景载入位

周期模式下，当前的计数值变为零时，需要使用本寄存器进行重载。自由运行模式和单次模式下不使用本寄存器。

对本寄存器写值和对载入寄存器(TIMERXLOAD)写值是不同的重载动作。差别在于对载入寄存器写值后，计数器立即从新值开始计数；而对本寄存器写值后，计数器并非立即从新值开始重新计数。

在载入寄存器或背景寄存器中的一个写值后，读取时总是读最后写入的寄存器值。也就是说，从载入寄存器和背景寄存器读取的是同样的值，该值是周期模式下计数器变为零时重载的值

21. 实时时钟 (RTC)

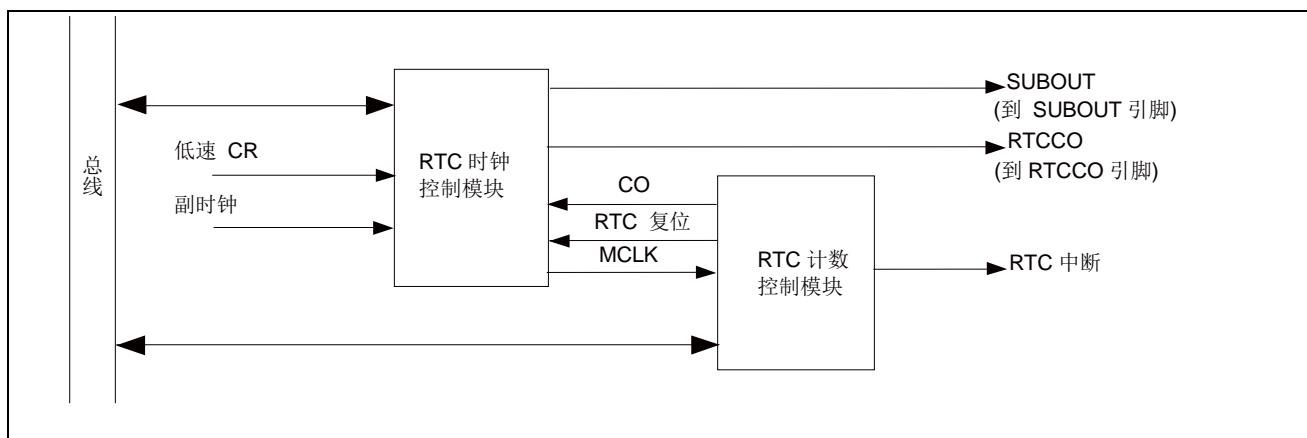
实时时钟 (Real-Time Clock, RTC) 由时钟控制模块和计数模块组成。

21.1 构成

本节介绍实时时钟的构成。

实时时钟的构成

图 21.1-1 实时时钟的构成



21.2 RTC 复位

本节介绍 RTC 的复位。

RTC 复位

RTC 有 4 种复位，不同复位初始化不同寄存器。

- 低电压检测复位/上电复位

初始化所有 RTC 寄存器。

- 系统复位

系统复位的复位源可以来自外部 RSTB 引脚输入、软件看门狗复位、硬件看门狗复位、时钟失效复位、时钟异常复位。

初始化 RTC 时钟控制模块所有寄存器。

初始化 RTC 计数模块部分寄存器，具体请参考“RTC 计数模块”章节的“RTC 计数模块复位”部分。

- RTC 复位

对 RTC 计数模块中的寄存器位 SWRST 写“1”，则产生 RTC 复位。

初始化 RTC 时钟控制模块部分寄存器，具体请参考“RTC 时钟控制模块”章节的“寄存器”部分中针对每个寄存器的相关警告。

初始化 RTC 计数模块部分寄存器，具体请参考“RTC 计数模块”章节的“RTC 计数模块复位”部分。

22. RTC 计数模块

本章介绍 RTC 计数模块的功能和操作。

22.1 概述

RTC 计数模块完成从 01 到 99 年间的年、月、日、小时、分、秒、星期的计数。具备闹钟和定时器功能。闹钟可以设置到具体的年/月/日/小时/分钟/秒，也可以分别设置到特定的年、月、日、小时、分钟或者秒。定时器最大计数周期可以设置到一天，通过设置具体的小时/分钟/秒既可以得到想要的时间间隔，也可以得到想要的间隔周期。

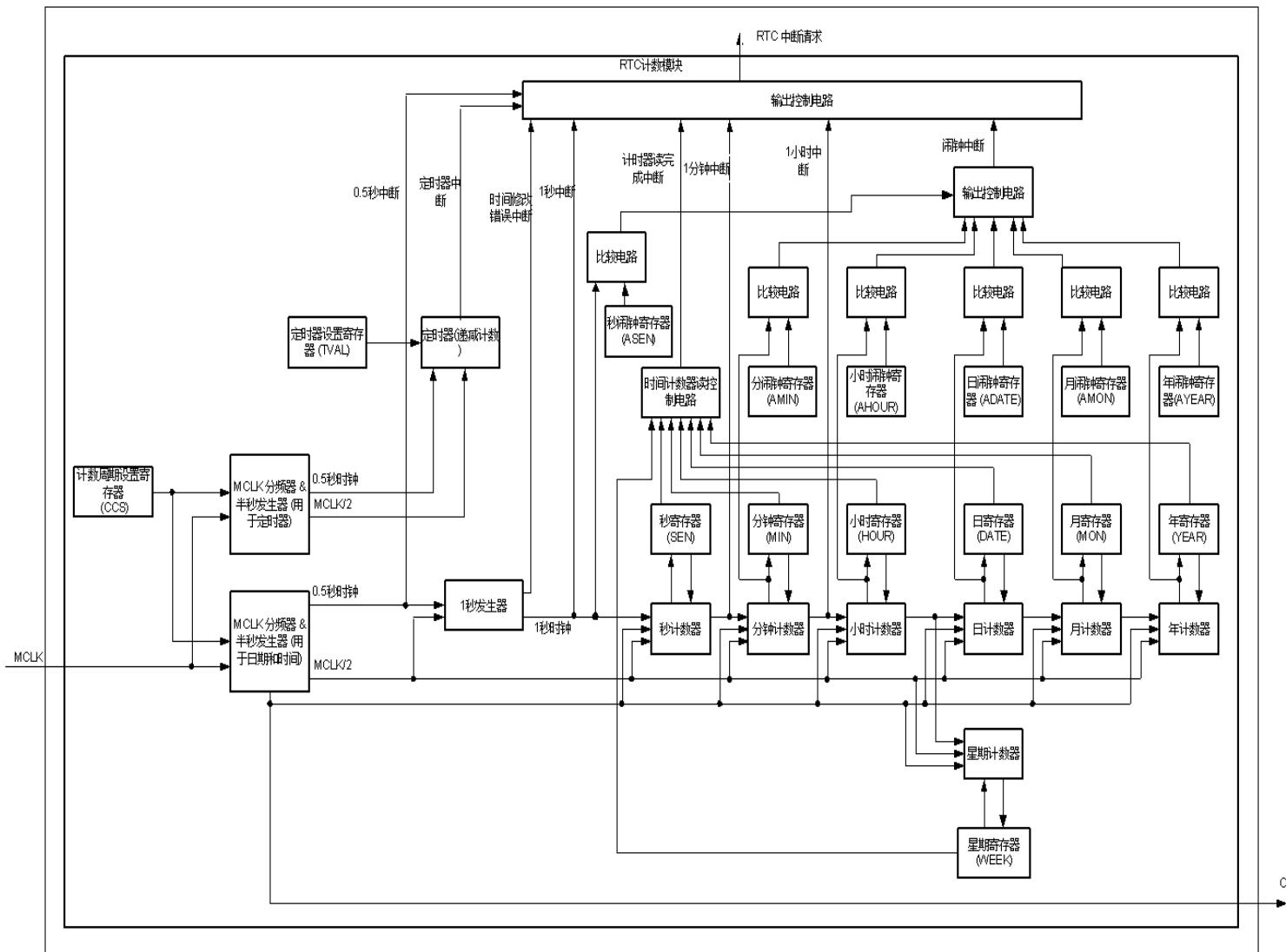
RTC 计数模块功能概述

- 日期和时间设置(年/月/日/小时/分钟/秒/星期)
- 从 01 到 99 年间的日期和时间计数（年、月、日、小时、分钟、秒、星期）
- 自动修正闰年
- 闹钟可以设置到具体的年/月/日/小时/分钟/秒，也可以分别设置到特定的年、月、日、小时、分钟或者秒
- 定时器最大计数可以设置到一天，通过设置具体的小时/分钟/秒既可以得到相应的单次定时或者周期定时
- 在 1 秒之内修改时间，计时器不会停止计数
- 有如下中断：
 - 闹钟中断
 - 每小时
 - 每分钟
 - 每秒
 - 每 0.5 秒
 - 定时器中断
 - 时间修改错误
 - 读时间完成

22.2 框图

图 22.2-1 显示 RTC 计数模块的框图。

图 22.2-1 RTC 计数模块框图



计数周期设置寄存器 (CCS)

该寄存器值为 0.5 秒计数值。当 RTC 开始操作或者 0.5 秒发生器的计数器为“0”时，加载该寄存器值到 0.5 秒发生器。

MCLK 分频器和半秒发生器 (用于定时器)

MCLK 分频器（用于定时器）产生一个 MCLK 2 分频的时钟。半秒发生器（用于定时器）基于 MCLK 分频器（用于定时器）产生的时钟进行 0.5 秒周期计数。

MCLK 分频器和半秒发生器 (用于日期和时间)

MCLK 分频器（用于日期和时间）产生一个 MCLK 的 2 分频时钟。半秒发生器（用于日期和时间）基于 MCLK 分频器（用于日期和时间）产生的时钟进行 0.5 秒周期计数。

定时器设置寄存器 (TVAL)

该寄存器值用于定时器单次定时或周期定时的计数。

定时器 (递减计数)

定时器加载定时器设置寄存器值，然后以 0.5 秒时钟周期开始递减计数。

1 秒发生器

该发生器对半秒发生器（用于日期和时间）输出的 0.5 秒时钟周期计数，然后输出 1 秒时钟脉冲。

秒、分、小时、日、月、年、星期 计数器

秒计数器、分计数器、小时计数器、日计数器、月计数器、年计数器、星期计数器分别对秒、分、小时、日、月、年、星期计数。

秒、分、小时、日、月、年、星期 寄存器

各寄存器分别表示秒、分、小时、日、月、年和星期信息。

计时器读控制电路

该电路报告读计时器的完成情况。

秒闹钟 (ASEN)、分闹钟(AMIN)、小时闹钟 (AHOUR)、日闹钟(ADATE)、月闹钟(AMON)、年闹钟寄存器 (AYEAR)

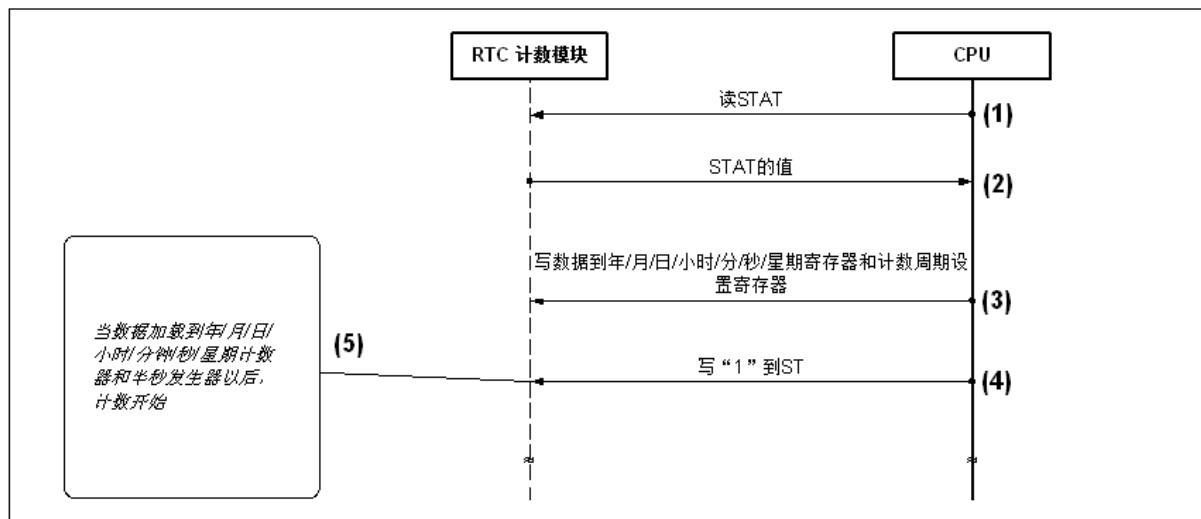
各寄存器分别存放秒、分、小时、日、月、年闹钟的值。当开启闹钟时，比较电路会比较闹钟寄存器的值和秒/分/小时/日/月/年计数器值，如果相等，则产生闹钟中断。

22.3 操作示例

本节介绍 RTC 计数模块的操作示例。

初始化时间设置过程示例

图 22.3-1 初始化时间设置过程



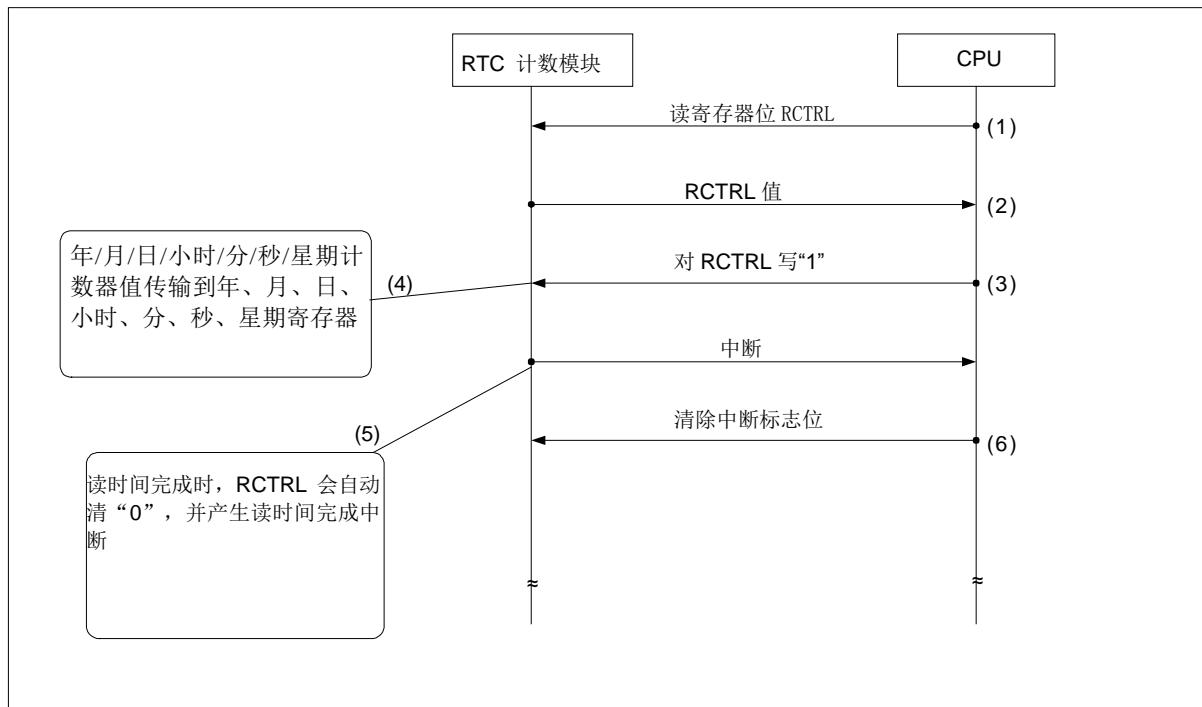
- 1) 读寄存器位 STAT.
- 2) 以下初始化时间流程可以进行的条件是 RTC 计数模块没有操作，即：STAT= “0”。STAT= “1” 时，请参考时间修改设置过程（计时继续）示例或时间修改设置过程（计时复位）示例。
- 3) 写想要的时间到年/月/日/小时/分钟/秒/星期寄存器(YEAR, MON, DATE, HOUR, MIN, SEN, 和 WEEK)。写想要的 0.5 秒计数值到计数周期设置寄存器(CCS)。
- 4) 写 ST= “1”。
- 5) 当设置 ST= “1” 以后，以下操作开始执行，计数器开始计数：
年/月/日/小时/分钟/秒/星期寄存器值加载到年/月/日/小时/分钟/秒/星期计数器。
计数周期设置寄存器值(CC)加载到半秒发生器，然后计数开始。

注意：

- 如果在系统复位或者 RTC 复位后需要继续计时，那么在设置 ST=1 之前应对 RCTRL 写“1”然后等待 RCTRL=0。
- 在上述复位之前，时间值会从年/月/日/小时/分/秒/星期计数器传输到年、月、日、小时、分、秒、星期寄存器。
- 当设置 ST=1 后，年、月、日、小时、分、秒、星期寄存器的值又传输到年/月/日/小时/分/秒/星期计数器，继续计时。

读时间的设置过程示例

图 22.3-2 读时间设置过程



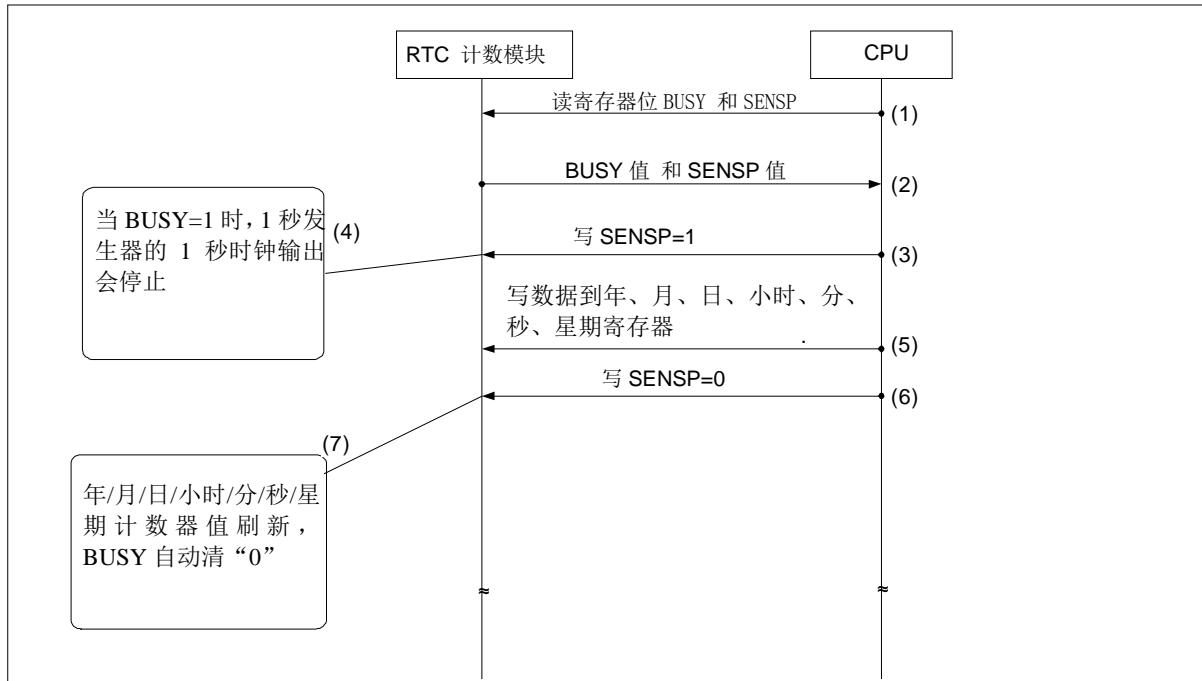
- 1) 读 RCTRL。
- 2) 当 RCTRL 为 “1” 时，须等到 RCTRL 变为 “0”。
- 3) 写 RCTRL=1。
- 4) 年/月/日/小时/分/秒/星期计数器值会传输到年、月、日、小时、分、秒、星期寄存器(YEAR, MON, DATE, HOUR, MIN, SEN, WEEK)。
- 5) 当上述操作完成时，RCTRL 自动清 “0”，并且年/月/日/小时/分/秒/星期计数器值的读完成中断产生。
- 6) 清除读完成中断标志位。

注意：

- 当 RCTRL="1"时，不要写“1”到 SENSP 和 SWRST。
- 当 RCTRL="1"时，不要设置 STOP 模式。
- 当对 RCTRL 写 “1”以后，在年/月/日/小时/分/秒/星期计数器值的读完成中断产生之前不要停止 RTC 计数模块（写 ST=0）。

时间修改设置过程示例（计时继续）

图 22.3-3 时间修改设置过程（计时继续）



- 1) 读 BUSY 和 SENSP。
- 2) 如果 BUSY= “1”，需要等待直到 BUSY=0，并且 SENSP=0。
如果不是，转到第 (3) 步。
- 3) 写 SENSP=1。
- 4) BUSY 自动置 “1”，1 秒发生器的 1 秒时钟输出停止。
- 5) 写想要的年、月、日、小时、分、秒、星期的值到年、月、日、小时、分、秒、星期寄存器(YEAR, MON, DATE, HOUR, MIN, SEN, WEEK)，此时应保持 SENSP=1。
- 6) 写 SENSP = 0。
- 7) 只有当新的年、月、日、小时、分、秒、星期寄存器值传输到年、月、日、小时、分、秒、星期计数器以后，BUSY 才自动清 “0”。

注意：

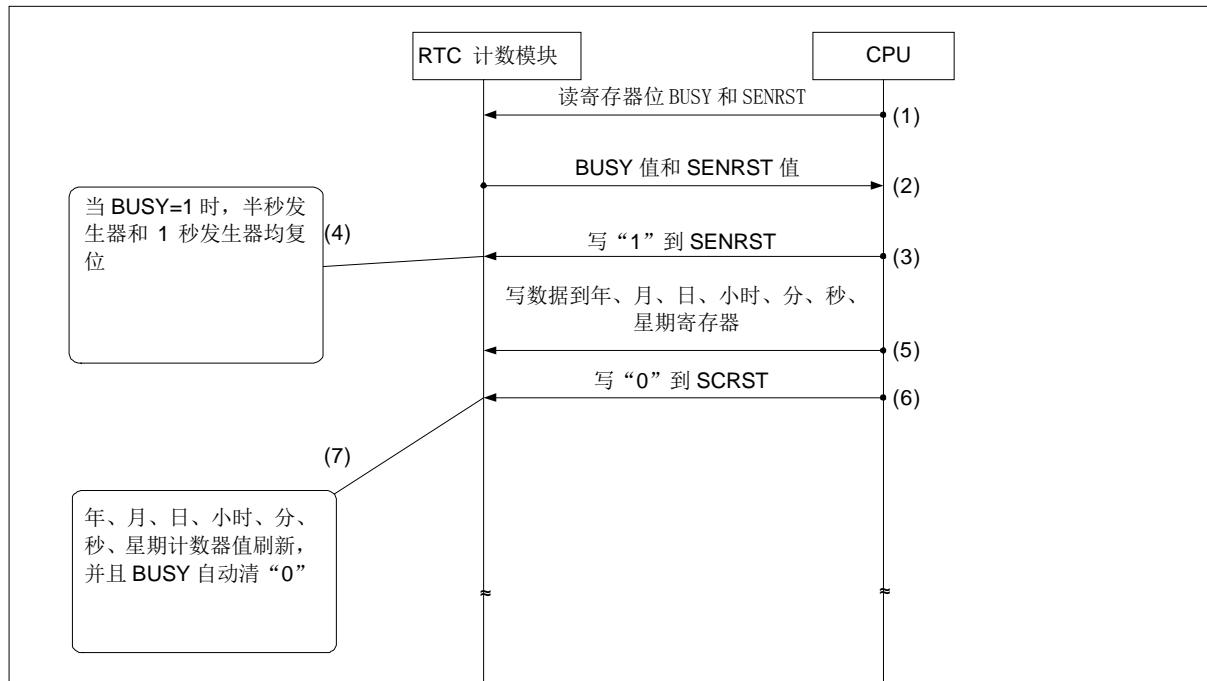
- 当 BUSY=1 和 SENSP=0 时，SENSP 不能写 “1”。
- 当 STAT=0 时，SENSP 不能写 “1”。
- 如果执行第 (3) 步到第 (6) 步之间的时间超过了 1 秒，连续计时将得不到保证。在这种情况下，将产生时间修改错误中断，且时间修改错误标志位将置 “1”。写 SENSP=0 以清除时间修改错误标志位，然后重新执行以上步骤。
- 当 SENSP=0 且 BUSY=1 时，表示数据在从年、月、日、小时、分、秒、星期寄存器 传输到年、月、日、小时、分、秒、星期计数器过程中，此时不能写年、月、日、小时、分、

秒、星期寄存器。

- 需要注意的是在 SENSP=1 的条件下，如果在年、月、日、小时、分、秒、星期计数器刷新之前写“1”到 RCTRL，年、月、日、小时、分、秒、星期寄存器的值会被年、月、日、小时、分、秒、星期计数器值覆盖。
- 在 BUSY=1 的条件下，数据不能正确从年、月、日、小时、分、秒、星期寄存器传输到年、月、日、小时、分、秒、星期计数器，因此年、月、日、小时、分、秒、星期计数值得不到保证。
- 当 BUSY=1 时，不允许写“0”到 ST。

时间修改设置过程示例 (计时复位)

图 22.3-4 间修改设置过程 (计时复位)



- 1) 读 BUSY 和 SENRST。
- 2) 如果 BUSY=1，须等待，直到 BUSY=0 并且 SENRST=0。
如果不是，转到步骤(3)。
- 3) 写 SENRST=1。
- 4) BUSY 将自动置 “1”。半秒发生器和 1 秒发生器都将复位。
- 5) 写想要的年、月、日、小时、分、秒、星期的值到年、月、日、小时、分、秒、星期寄存器(YEAR, MON, DATE, HOUR, MIN, SEN, WEEK)，此时应保持 SENSP=1。
- 6) 写 SENRST = 0。
- 7) 只有当新的年、月、日、小时、分、秒、星期寄存器值传输到年、月、日、小时、分、秒、星期计数器以后，BUSY 才自动清“0”。

注意：

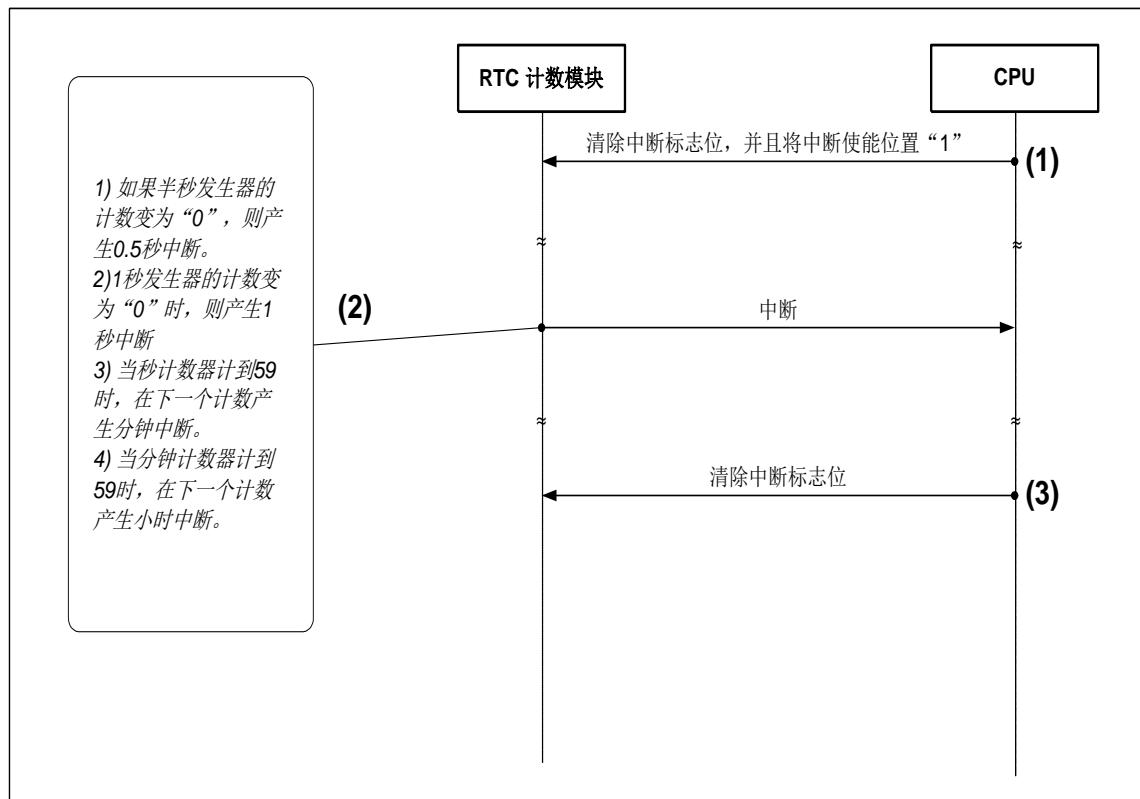
- 当 BUSY=1, SENRST=0 时，不能对 SENRST 写 1。
- 当 STAT=0 时，不允许写“0”到 SENRST。
- 当 SENSP=0 且 BUSY=1 时，表示数据在从年、月、日、小时、分、秒、星期寄存器传输到年、月、日、小时、分、秒、星期计数器过程中，此时不能写年、月、日、小时、分、秒、星期寄存器。.
- 需要注意的是在 SENRST=1 的条件下，如果在年、月、日、小时、分、秒、星期计数器刷新之前写“1”到 RCTRL，年、月、日、小时、分、秒、星期寄存器的值会被年、月、日、

小时、分、秒、星期计数器值覆盖。

- 在 BUSY=1 的条件下，数据不能正确从年、月、日、小时、分、秒、星期寄存器传输到年、月、日、小时、分、秒、星期计数器，因此年、月、日、小时、分、秒、星期计数值得不到保证。
- 当 BUSY=1 时，不允许写“0”到 ST。

每 0.5 秒/每秒/每分/每小时中断设置过程示例

图 22.3-5 每 0.5 秒/每秒/每分/每小时中断设置过程



1) 写 INTSSF/INTSF/INTMF/INTHF=0 以清除中断标志位。

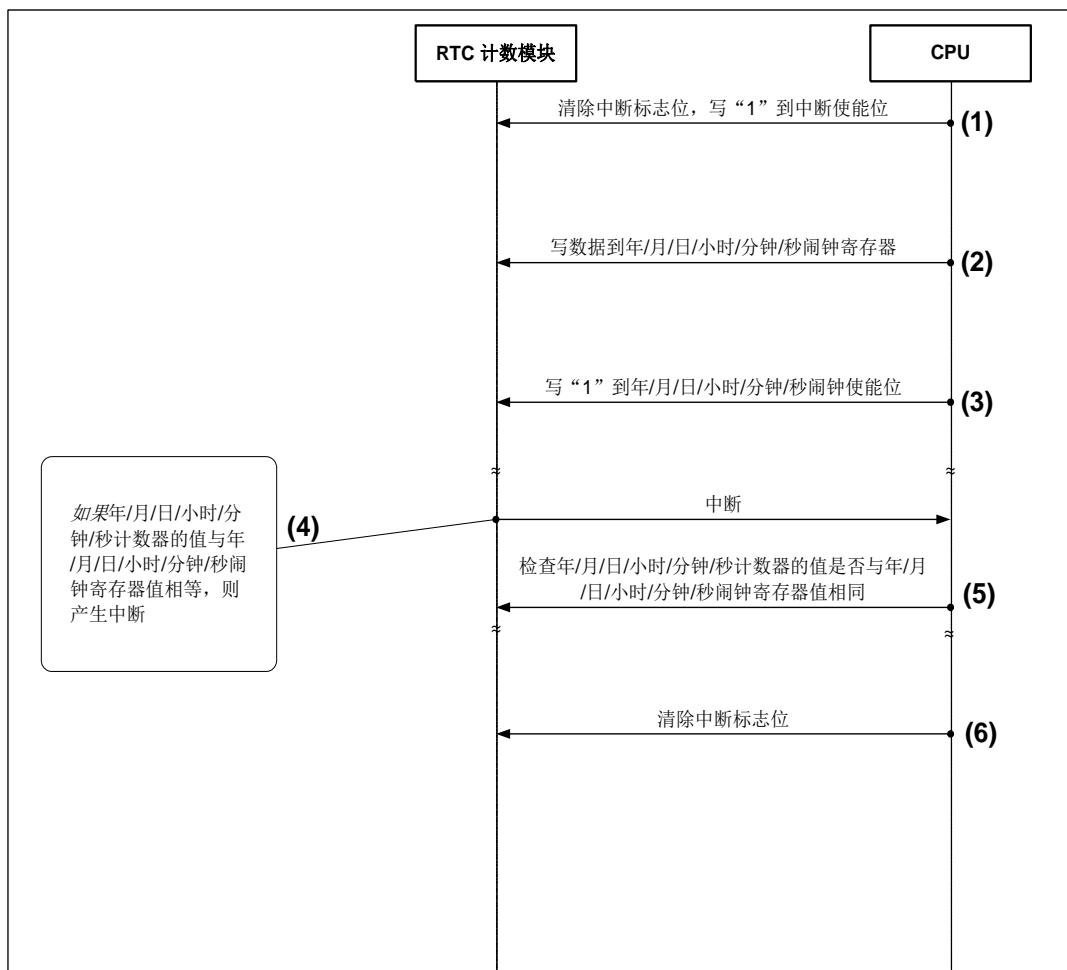
写“1”到中断使能位 INTSSIEN, INTSIEN, INTMIEN, or INTHIEN 以便能中断。

2) 每 0.5 秒/秒/分/小时就会产生中断。

3) 写 INTSSF/INTSF/INTMF/INTHF=0 清除中断标志位。

闹钟中断设置过程示例

图 22.3-6 闹钟中断设置过程



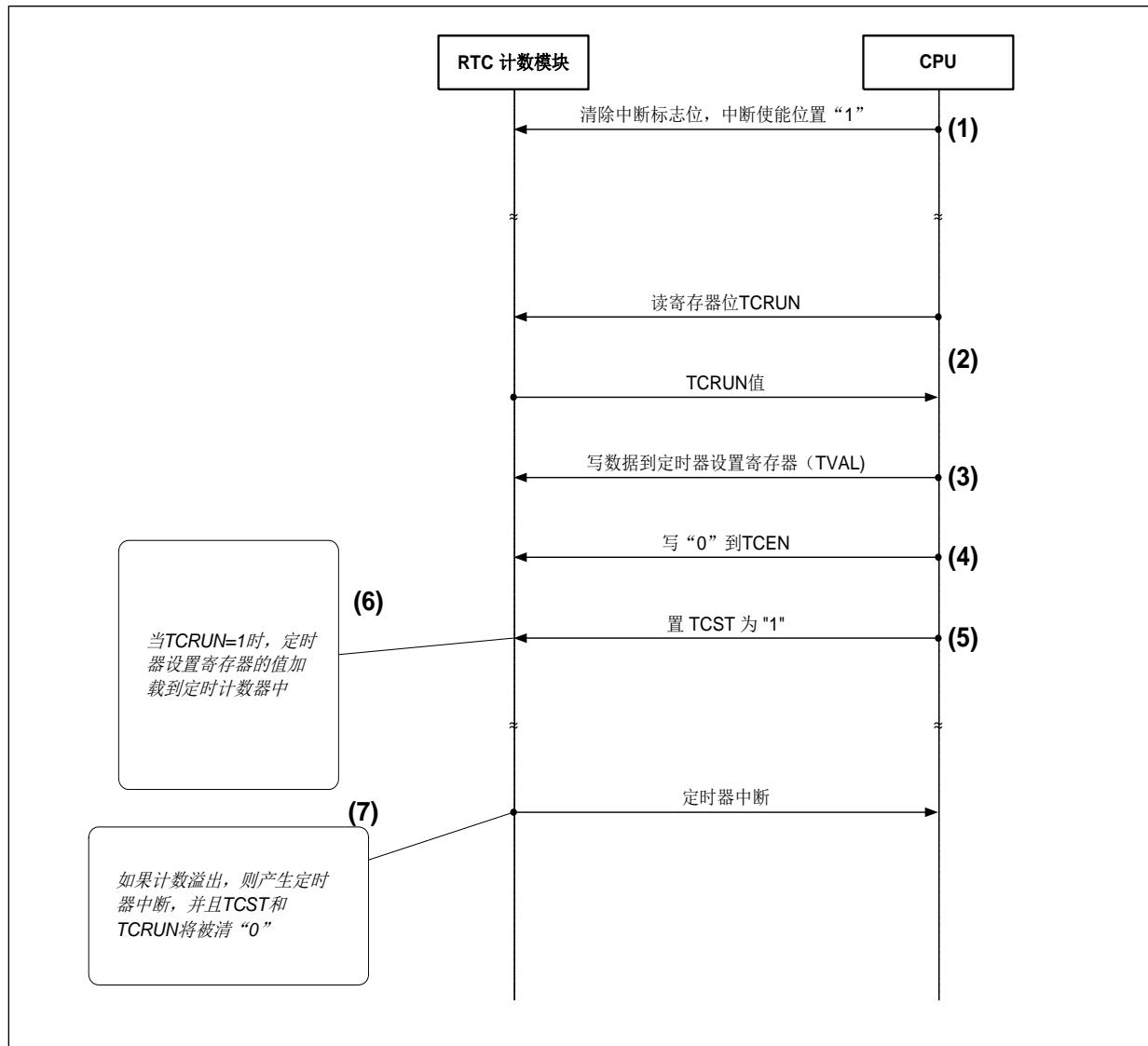
- 1) 写 INTALF=0 以清除中断标志位。
写 INTALIEN=1 以使能闹钟中断。
- 2) 写时间到年/月/日/小时/分钟/秒闹钟寄存器，闹钟中断将在相应时间产生。
- 3) 写“1”到年/月/日/小时/分钟/秒闹钟使能位。
- 4) RTC 计数模块的中断请求将在年/月/日/小时/分钟/秒闹钟寄存器值与年/月/日/小时/分钟/秒计数器值相等时产生。
- 5) 参照读时间设置过程示例读当前时间，检查年/月/日/小时/分钟/秒闹钟寄存器值与年/月/日/小时/分钟/秒计数器值是否相等。
- 6) 写 INTALF=0 以清除闹钟中断标志位。

注意：

- 中断可能在闹钟使能位置“1”后立即产生。通过读取时间检查年/月/日/小时/分钟/秒闹钟寄存器值与年/月/日/小时/分钟/秒计数器值是否相等。

定时器中断设置过程示例（单次定时）

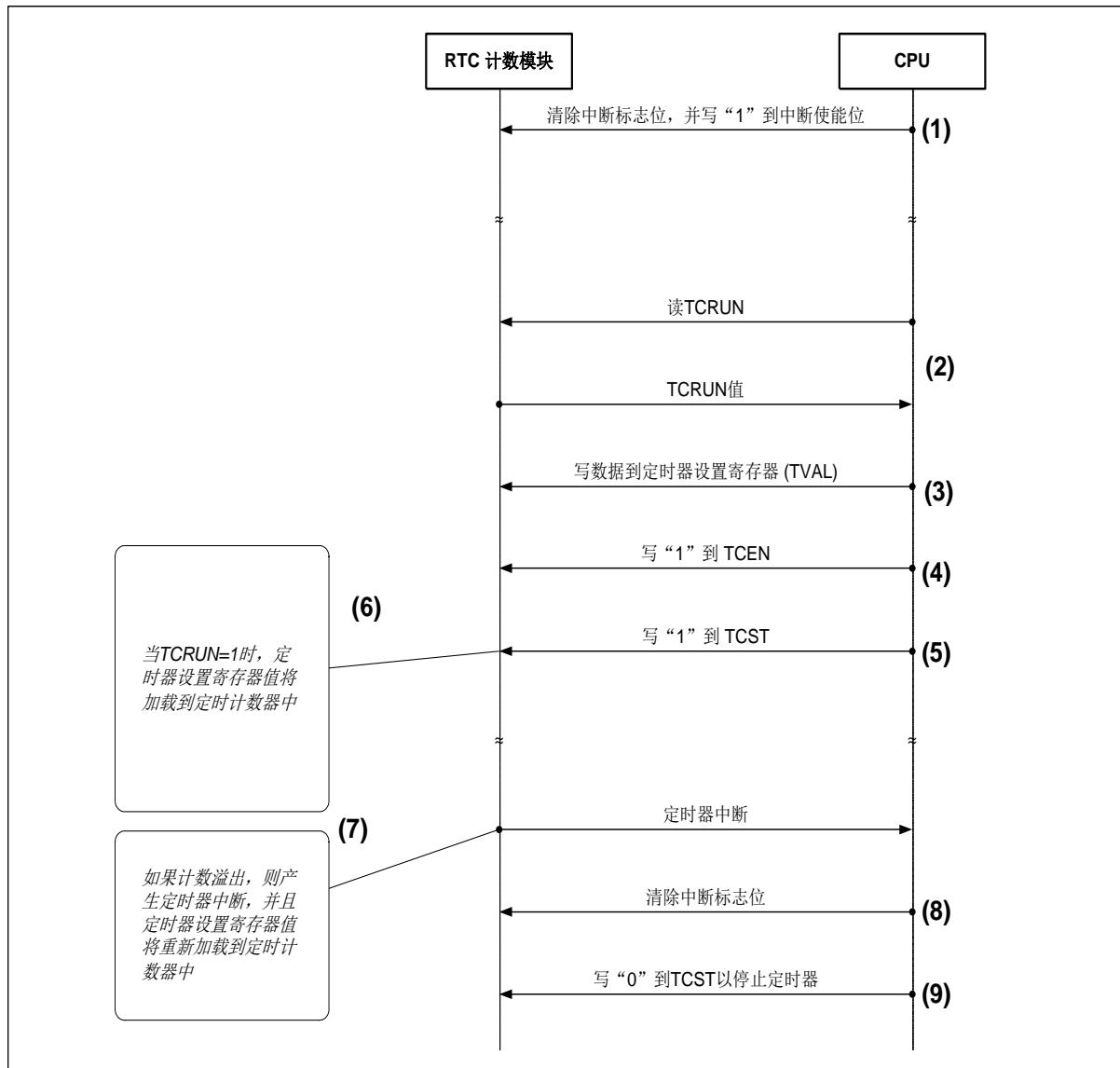
图 22.3-7 定时器中断设置过程（单次定时）



- 1) 写 INTTMF=0 以清除定时器中断标志位。
写 INTTMIEN=1 使能定时器中断。
 - 2) 读定时器操作位 (TCRUN) 并检查是否为 “0” (操作停止)。
 - 3) 写定时时间到定时设置寄存器(TVAL)。
 - 4) 写定时器控制位 TCEN=0。
 - 5) 写定时器开始位 TCST=1。
 - 6) 定时器设置寄存器值加载到定时器，然后递减计数开始。
 - 7) 如果递减计数溢出，则产生中断，并且 TCST 将清 “0”。定时器停止后 TCRUN 也将清 “0”。
- 注意：
- 只有当定时器没有工作 (TCRUN=0) 时，才能更改 TCEN 的设置。

定时器中断设置过程示例（周期定时）

图 22.3-8 定时器中断设置过程示例（周期定时）



- 1) 写 INTTMF=0 以清除定时器中断标志位。
写 INTTMIEN=1 使能定时器中断。
- 2) 读定时器操作位 (TCRUN) 并检查是否为 “0” (操作停止)。
- 3) 写定时时间到定时设置寄存器(TVAL)。
- 4) 写定时器控制位 TCEN=1。
- 5) 写定时器开始位 TCST=1。
- 6) 定时器设置寄存器值加载到定时器，然后递减计数开始。
- 7) 如果递减计数溢出，则产生中断，并且定时器设置寄存器值将重新加载到定时器，继续定时操作。
- 8) 写 INTTMF= 0 清除定时器中断标志位。

9) 写 $\text{TCST}=0$ 停止定时器操作。

注意：

- 只有当定时器没有工作 ($\text{TCRUN}=0$) 时，才能更改 TCEN 的设置。

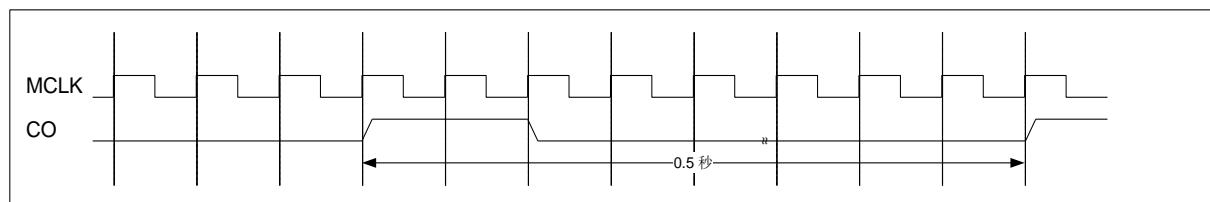
CO 外部引脚输出操作

RTC 计数模块有一个作为 0.5 秒时钟输出的 CO 外部引脚。

CO 外部引脚以 0.5 秒时钟 (CO) 周期输出。

图 22.3-9 显示 CO 外部引脚输出波形。

图 22.3-9 CO 外部引脚输出波形



22.4 RTC 计数模块复位

本节介绍复位操作。

低电压检测复位/上电复位操作

表 22.4-1 阴影部分是受低电压检测复位/上电复位控制的存器位。

0.5 秒发生器、1 秒发生器、定时器、年/月/日/小时/分/秒/星期计数器也受低电压检测复位/上电复位控制，只是在表中没有显示。

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
INTCRIE	INTERIE	INTALIE	INTTM1	INTTM2	INTHEN1	INTHEN2	INTSIE	INTSEN1	INTSEN2	INTCRF	INTERF	INTALF	INTTMF	INTHF	INTMF	INTSSF
RTCCCTL1	-	-	YEAE	MONE	DAYE	HOUE	MINE	SENE	-	BUSY	SENRST	SENSP	SWRST	STAT	OE	ST
RTCCCTL2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CCS	CC15	CC14	CC13	CC12	CC11	CC10	CC9	CC8	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
DATE/HOUR/MIN/SEN	-	-	HD1	HD0	LD3	LD2	LD1	LD0	-	-	HH1	HH0	LH3	LH2	LH1	LH0
YEAR/MON/WEEK	-	-	-	-	-	-	-	-	-	HS2	HS1	HS0	LS3	LS2	LS1	LS0
ADATE/AHOUR/AMIN/ASEN	-	-	HD1	HD0	LD3	LD2	LD1	LD0	-	-	HH1	HH0	LH3	LH2	LH1	LH0
AYEAR/AMON	-	-	-	-	HMON0	HMON3	HMON2	HMON1	HMON0	-	-	-	-	-	-	-
TVAL	VAL15	VAL14	VAL13	VAL12	VAL11	VAL10	VAL9	VAL8	VAL7	VAL6	VAL5	VAL4	VAL3	VAL2	VAL1	VAL0

系统复位操作

表 22.4-2 阴影部分是受系统复位控制的寄存器位。没有显示在表中的 1 秒发生器和定时器也受系统复位控制。

半秒发生器和年/月/日/小时/分钟/秒/星期计数器不受系统复位控制。

表 4-2 系统复位目标寄存器位

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RTCCCTL1	INTCRIE N	INTERIE N	INTALIE N	INTTMIE N	INTHIEN N	INTMIEN N	INTSIEN N	INTSSIEN N	INTCRF	INTERF	INTALF	INTTMF	INTHF	INTMF	INTSF	INTSSF
-	-	YEAE	MONE	DAYE	HOUE	MINE	SENE		BUSY	SENrst	SENsp	SWRST	STAT	OE	ST	
RTCCCTL2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCTRL
CCS	CC15	CC14	CC13	CC12	CC11	CC10	CC9	CC8	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
DATE/HOUR/ MIN/SEN	-	-	HD1	HD0	LD3	LD2	LD1	LD0	-	-	HH1	HH0	LH3	LH2	LH1	LH0
YEAR/MON/WE EK	-	-	HMI1	HMI0	LMI3	LMI2	LMI1	LMI0	-	HS2	HS1	HS0	LS3	LS2	LS1	LS0
ADATE/AHOUR/ AMIN/ASEN	-	-	HM0	HM0	LD3	LD2	LD1	LD0	-	HY3	HY2	HY1	HY0	LY3	LY2	LY0
AYEAR/AMON	-	-	-	-	-	-	-	-	-	HH1	HH0	LH3	LH2	LH1	LH0	
TVAL	VAL15	VAL14	VAL13	VAL12	VAL11	VAL10	VAL9	VAL8	VAL7	VAL6	VAL5	VAL4	VAL3	VAL2	VAL1	VAL0

RTC 复位操作

表 22.4-3 的阴影部分是受 RTC 复位的寄存器位。表中没有显示的 1 秒发生器和定时器也受 RTC 复位控制。

半秒发生器和年/月/日/小时/分钟/秒/星期计数器不受系统复位控制。

表 4-3 RTC 复位目标寄存器位

	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RTCCCTL1	INTCRIE_N	INTERIE_N	INTTALIE_N	INTTIME_N	INTHEN_N	INTMEN_N	INTSIEN_N	INTSSIEN_N	INTCRF_N	INTERF_N	INTALF_N	INTTMF_N	INTHF_N	INTSF_N	INTSSF_N	
RTCCCTL2	-	-	YEAE	MONE	DAVE	HOUe	MINE	SENE		BUSY	SENIRST	SENISP	SWRST	STAT	OE	ST
CCS	CC15	CC14	CC13	CC12	CC11	CC10	CC9	CC8	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
DATE/HOUR/ MIN/SEN	-	-	HD1	HD0	LD3	LD2	LD1	LD0	-	-	HH1	HH0	LH3	LH2	LH1	LH0
YEAR/MON/WEEK	-	-	HM01	HM00	LM13	LM12	LM11	LM10	-	HS2	HS1	HS0	LS3	LS2	LS1	LS0
ADATE/AHOUR/ AMIN/ASEN	-	-	HD1	HD0	LD3	LD2	LD1	LD0	-	HY3	HY2	HY1	HY0	LY3	LY2	LY1
AYEAR/AMON	-	-	HM01	HM00	LM3	LM2	LM1	LM0	-	-	HH1	HH0	LH3	LH2	LH1	LH0
TVAL	VAL15	VAL14	VAL13	VAL12	VAL11	VAL10	VAL9	VAL8	VAL7	VAL6	VAL5	VAL4	VAL3	VAL2	VAL1	VAL0

22.5 闰年修正

本节介绍 RTC 计数模块的闰年修正。

闰年修正

表 22.5-1 显示闰年和每个月的天数的关系。

表 22.5-1 闰年列表

年	闰年	月											
		1	2	3	4	5	6	7	8	9	10	11	12
01 to 03	否	31	28	31	30	31	30	31	31	30	31	30	31
04	是	31	29	31	30	31	30	31	31	30	31	30	31
05 to 07	否	31	28	31	30	31	30	31	31	30	31	30	31
08	是	31	29	31	30	31	30	31	31	30	31	30	31
09 to 11	否	31	28	31	30	31	30	31	31	30	31	30	31
12	是	31	29	31	30	31	30	31	31	30	31	30	31
13 to 15	否	31	28	31	30	31	30	31	31	30	31	30	31
16	是	31	29	31	30	31	30	31	31	30	31	30	31
17 to 19	否	31	28	31	30	31	30	31	31	30	31	30	31
20	是	31	29	31	30	31	30	31	31	30	31	30	31
21 to 23	否	31	28	31	30	31	30	31	31	30	31	30	31
24	是	31	29	31	30	31	30	31	31	30	31	30	31
25 to 27	否	31	28	31	30	31	30	31	31	30	31	30	31
28	是	31	29	31	30	31	30	31	31	30	31	30	31
29 to 31	否	31	28	31	30	31	30	31	31	30	31	30	31
32	是	31	29	31	30	31	30	31	31	30	31	30	31
33 to 35	否	31	28	31	30	31	30	31	31	30	31	30	31
36	是	31	29	31	30	31	30	31	31	30	31	30	31
37 to 39	否	31	28	31	30	31	30	31	31	30	31	30	31
40	是	31	29	31	30	31	30	31	31	30	31	30	31
41 to 43	否	31	28	31	30	31	30	31	31	30	31	30	31
44	是	31	29	31	30	31	30	31	31	30	31	30	31
45 to 47	否	31	28	31	30	31	30	31	31	30	31	30	31
48	是	31	29	31	30	31	30	31	31	30	31	30	31
49 to 51	否	31	28	31	30	31	30	31	31	30	31	30	31

年	闰年	月											
		1	2	3	4	5	6	7	8	9	10	11	12
52	是	31	29	31	30	31	30	31	31	30	31	30	31
53 to 55	否	31	28	31	30	31	30	31	31	30	31	30	31
56	是	31	29	31	30	31	30	31	31	30	31	30	31
57 to 59	否	31	28	31	30	31	30	31	31	30	31	30	31
60	是	31	29	31	30	31	30	31	31	30	31	30	31
61 to 63	否	31	28	31	30	31	30	31	31	30	31	30	31
64	是	31	29	31	30	31	30	31	31	30	31	30	31
65 to 67	否	31	28	31	30	31	30	31	31	30	31	30	31
68	是	31	29	31	30	31	30	31	31	30	31	30	31
69 to 71	否	31	28	31	30	31	30	31	31	30	31	30	31
72	是	31	29	31	30	31	30	31	31	30	31	30	31
73 to 75	否	31	28	31	30	31	30	31	31	30	31	30	31
76	是	31	29	31	30	31	30	31	31	30	31	30	31
77 to 79	否	31	28	31	30	31	30	31	31	30	31	30	31
80	是	31	29	31	30	31	30	31	31	30	31	30	31
81 to 83	否	31	28	31	30	31	30	31	31	30	31	30	31
84	是	31	29	31	30	31	30	31	31	30	31	30	31
85 to 87	否	31	28	31	30	31	30	31	31	30	31	30	31
88	是	31	29	31	30	31	30	31	31	30	31	30	31
89 to 91	否	31	28	31	30	31	30	31	31	30	31	30	31
92	是	31	29	31	30	31	30	31	31	30	31	30	31
93 to 95	否	31	28	31	30	31	30	31	31	30	31	30	31
96	是	31	29	31	30	31	30	31	31	30	31	30	31
97 to 99	否	31	28	31	30	31	30	31	31	30	31	30	31

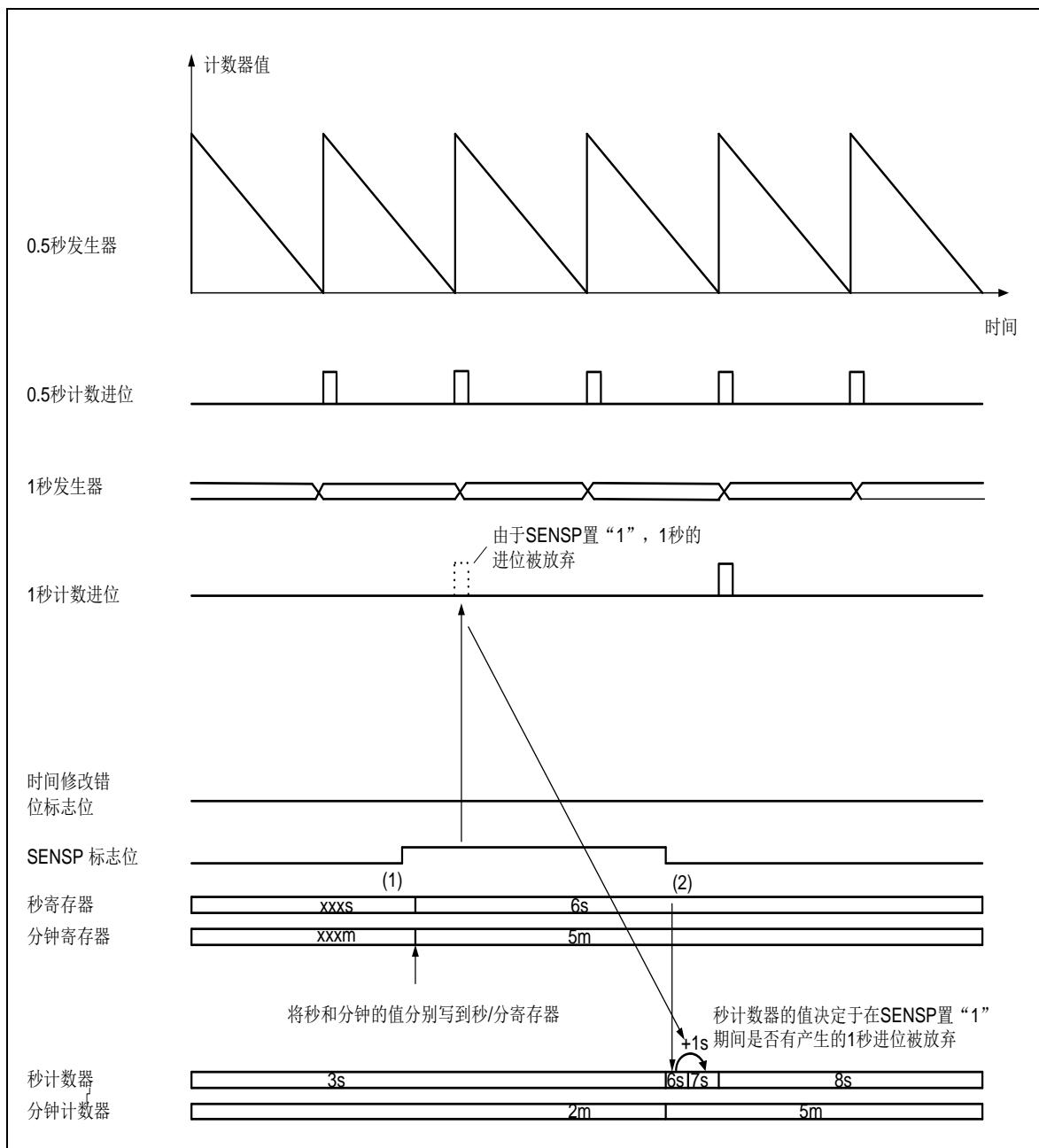
22.6 时间修改错误

本节介绍在计时继续的情况下修改时间发生的时间修改错误。

时间修改错误 1

下面例子中，在 SENSP 置“1”的时间修改（计时继续）期间 0.5 秒发生器的计数有 2 个进位。只修改秒计数器和分钟计数器。

图 22.6-1 只修改秒计数器和分钟计数器



1) SENSP 置“1”，秒寄存器写入 6 秒，分钟寄存器写入 5 分钟。

2) 在 SENSP 置“1”期间，有两个 0.5 秒进位。6 秒加载到秒计数器，紧接着加上 1 秒，计数

器变为 7 秒。

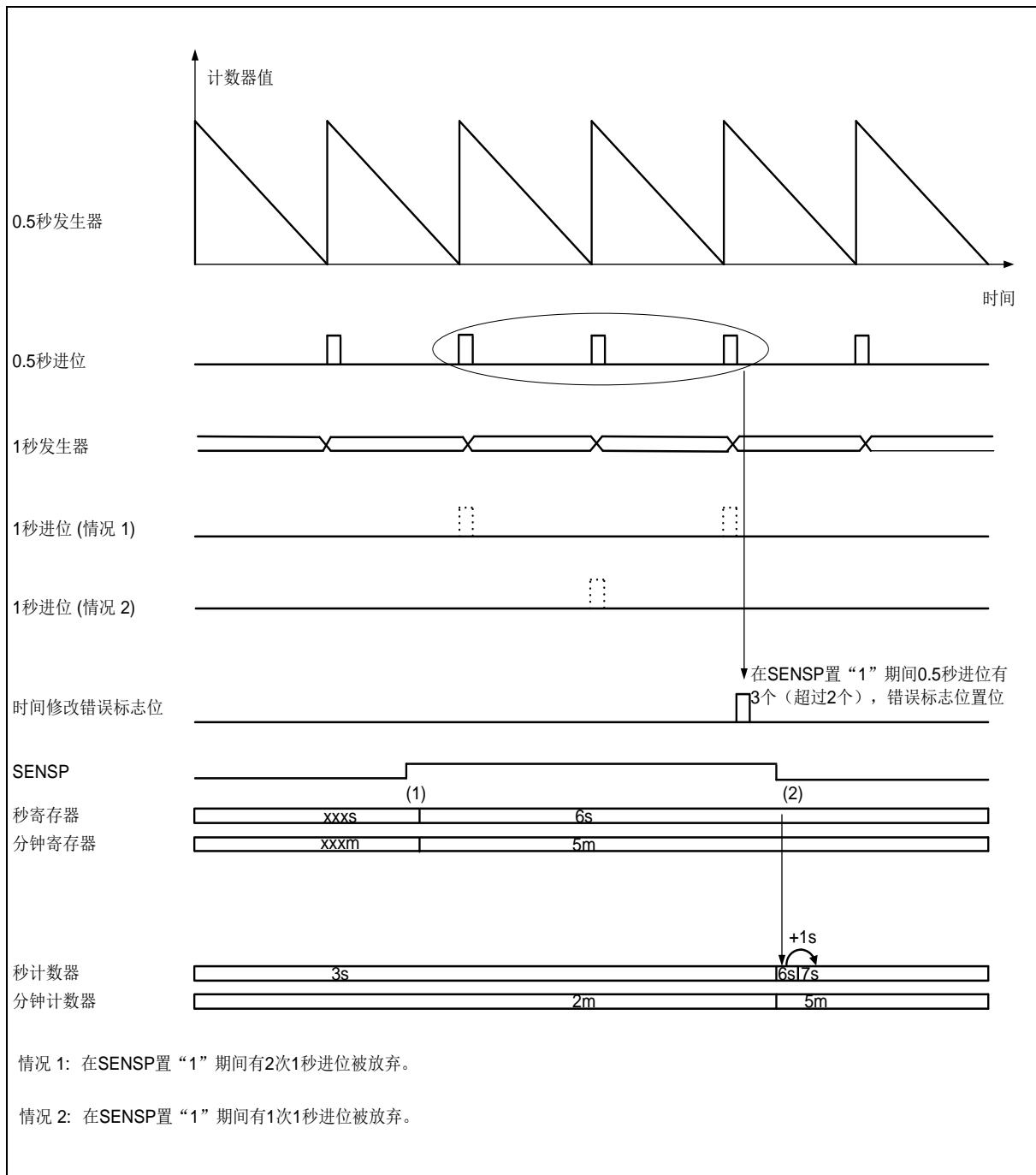
注意：

- 如果在 SENSP 置“1”期间有 2 次或者 2 次以下的 0.5 秒进位，那么时间修改错误标志位不会变为“1”。
- 因为在 SENSP 置“1”期间 1 秒进位被放弃，秒计数器不会计数，在 SENSP 清“0”时加载秒寄存器值到秒计数器，紧接着加上 1 秒。
- 如果在 SENSP 置“1”期间没有 1 秒进位被放弃，那么在 SENSP 清“0”时只是加载秒寄存器到秒计数器，不会再加 1 秒。

时间修改错误 2

下面例子中，在 SENSP 置“1”的时间修改（计时继续）期间 0.5 秒发生器的计数有 3 个进位。仅修改秒计数器和分钟计数器

图 22.6-2 仅修改秒计数器和分钟计数器



- 1) SENSP 置“1”，秒寄存器写入 6 秒，分钟寄存器写入 5 分钟。
- 2) 在 SENSP 置“1”期间有 3 次 0.5 秒进位，时间修改错误标志位置“1”。6 秒加载到秒计数器，紧接着加上 1 秒，变为 7 秒。

注意：

- 当时间错误标志位置“1”时，在第一种情况下，因为在 SENSP 置“1”期间有 2 次 1 秒进位被放弃，所以会产生 1 秒移位；在第二种情况下，因为在 SENSP 置“1”期间只有 1 次 1 秒进位被放弃，所以不会产生秒移位。但是，一旦发生时间修改错误，建议重新执行时间修改步骤，因为不能判断是哪种情况

22.7 寄存器

本节描述 RTC 计数模块的寄存器。

RTC 计数模块寄存器列表

表 22.7-1 RTC 计数模块寄存器列表

缩写	寄存器名	章节
RTCCTL1	控制寄存器 1	22.7.1
RTCCTL2	控制寄存器 2	22.7.2
CCS	计数周期设置寄存器	22.7.3
DATE	日寄存器	22.7.4
HOUR	小时寄存器	22.7.5
MIN	分钟寄存器	22.7.6
SEN	秒寄存器	22.7.7
YEAR	年寄存器	22.7.8
MON	月寄存器	22.7.9
WEEK	星期寄存器	22.7.10
ADATE	日闹钟寄存器	22.7.11
AHOUR	小时闹钟寄存器	22.7.12
AMIN	分钟闹钟寄存器	22.7.13
AYEAR	年闹钟寄存器	22.7.14
AMON	月闹钟寄存器	22.7.15
TVAL	定时器设置寄存器	22.7.16

22.7.1 控制寄存器 1 (RTCCTL1)

控制寄存器 1 用来控制 RTC 计数模块的操作。

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	INTCRI EN	INTERI EN	INTALI EN	INTTMI EN	INTHIE N	INTMIE N	INTSIE N	INTSSI EN
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	23	22	21	20	19	18	17	16
字段名	INTCRF	INTERF	INTALF	INTTM F	INTHF	INTMF	INTSF	INTSSF
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8
字段名	Reserved		YEAE	MONE	DAYE	HOUe	MINE	SENE
属性		R		R/W	R/W	R/W	R/W	R/W
初始值		000		0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	Reserved	BUSY	SENRS T	SENSP	SWRST	STAT	Reserve d	ST
属性	R	R	R/W	R/W	R/W	R	R	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31] INTCRIEN: 年/月/日/小时/分钟/秒/星期计数器值的读完成中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit30] INTERIEN: 时间修改错误中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit29] INTALIEN: 闹钟中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit28] INTTMIEN: 定时器中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit27] INTHIEN: 1 小时中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit26] INTMIEN: 1 分钟中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit25] INTSIEN: 1 秒中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit24] INTSSIEN: 0.5 秒中断使能位

bit	描述
0	中断禁止
1	中断使能

[bit23] INTCRF: 年/月/日/小时/分钟/秒/星期计数器值的读完成中断标志位

该位表示在用 RCTRL 读取时间时，年/月/日/小时/分钟/秒/星期计数器值到年/月/日/小时/分钟/秒/星期寄存器的传输完成。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	读年/月/日/小时/分钟/秒/星期计数器值未完成
	1	读年/月/日/小时/分钟/秒/星期计数器值已完成
写	0	清除标志位
	1	无效

[bit22] INTERF: 时间修改错误标志位

该位表示在时间修改 (SENSP=1) 期间，秒计数器计数异常。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无时间修改错误发生
	1	有时间修改错误发生
写	0	清除标志位
	1	无效

[bit21] INTALF: 闹钟中断标志位

该位表示年/月/日/小时/分钟/秒计数器值与年/月/日/小时/分钟/秒闹钟寄存器值相等。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无闹钟中断产生
	1	有闹钟中断产生
写	0	清除标志位
	1	无效

[bit20] INTTMF: 定时器溢出标志位

如果定时器的计数溢出，该位将置“1”。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无定时器计数溢出产生
	1	有定时器计数溢出产生
写	0	清除标志位
	1	无效

[bit19] INTHF: 1 小时中断标志位

计时到 1 小时，该位置“1”。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无 1 小时中断产生
	1	有 1 小时中断产生
写	0	清除标志位
	1	无效

[bit18] INTMF: 1 分钟中断标志位

计时到 1 分钟，该位置“1”。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无 1 分钟中断产生
	1	有 1 分钟中断产生
写	0	清除标志位
	1	无效

[bit17] INTSF: 1 秒中断标志位

计时到 1 秒钟，该位置“1”。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无 1 秒中断产生
	1	有 1 秒中断产生
写	0	清除标志位
	1	无操作

[bit16] INTSSF: 0.5 秒中断标志位

计时到 0.5 秒钟，该位置“1”。

在 RMW 访问模式下读该位值总是“1”。

bit		描述
读	0	无 0.5 秒中断产生
	1	有 0.5 秒中断产生
写	0	清除标志位
	1	无操作

[bit15:14] Reserved: 保留位，读取值总是“0”。

[bit13] YEAE: 年闹钟使能寄存器

该位在产生闹钟中断时使能年闹钟寄存器与年计数器的比较。

bit	描述
0	禁止年闹钟
1	使能年闹钟

[bit11] MONE: 月闹钟使能寄存器

该位在产生闹钟中断时使能月闹钟寄存器与月计数器的比较。

bit	描述

0	禁止月闹钟
1	使能月闹钟

[bit10] DAYE: 日闹钟使能寄存器

该位在产生闹钟中断时使能日闹钟寄存器与日计数器的比较。

bit	描述
0	禁止日闹钟
1	使能日闹钟

[bit9] HOUe: 小时闹钟使能寄存器

该位在产生闹钟中断时使能小时闹钟寄存器与小时计数器的比较。

bit	描述
0	禁止小时闹钟
1	使能小时闹钟

[bit8] MINE: 分闹钟使能寄存器

该位在产生闹钟中断时使能分闹钟寄存器与分计数器的比较。

bit	描述
0	禁止分闹钟
1	使能分闹钟

[bit7] SENE: 秒闹钟使能寄存器

该位在产生闹钟中断时使能秒闹钟寄存器与秒计数器的比较。

bit	描述
0	禁止秒闹钟
1	使能秒闹钟

[bit6] BUSY: 忙

该位表示时间修改正在进行中。

bit	描述
0	没有进行时间修改
1	满足以下条件之一时: · SENSP = "1" · SENRST = "1" 年/月/日/小时/分/秒/星期寄存器的值在向年/月/日/小时/分/秒/星期计数器传输过程中。

[bit5] SENRST: 半秒发生器/1 秒发生器复位

该位对半秒发生器/1 秒发生器（用于日期和时间）进行复位控制。

bit	描述
0	释放半秒发生器/1 秒发生器（用于日期和时间）的复位状态
1	复位半秒发生器/1 秒发生器（用于日期和时间）

当 RTC 工作 (STAT=1) 时，如果该位为“0”且 SENSP 为“0”，年/月/日/小时/分/秒/星期寄存器的值不能被刷新。SENSP 和 SENRST 能同时置“1”。当 RTC 停止 (STAT=0) 时，该位不能置“1”。

[bit4] SENS: 1 秒时钟输出停止位

该位控制 1 秒发生器的 1 秒时钟输出。

bit	描述
0	输出使能
1	禁止输出

当 RTC 工作 (STAT=1) 时，如果该位为“0”且 SENRST 为“0”，年/月/日/小时/分/秒/星期寄存器的值不能被刷新。SENSP 和 SENRST 能同时置“1”。当 RTC 停止 (STAT=0) 时，该位不能置“1”。

[bit3] SWRST: RTC 复位

“4.RTC 计数模块复位”中的表 4-3 显示了能被 RTC 复位的寄存器。

在 RMW 访问模式下读该位值总是“0”。

bit	描述
0	RTC 复位已完成
1	写“1”时，硬件将实施 RTC 复位

[bit2] STAT: RTC 计数模块操作位

该位表示 RTC 计数模块的操作状态。

在 RTC 计数模块在操作时如果将 ST 清“0”，RTC 计数模块将停止并且 STAT 也将变为“0”。

bit	描述
0	RTC 计数模块停止
1	RTC 计数模块工作

[bit1] Reserved: 保留位，读取值总为“0”。

[bit0] ST: 开始位，该位控制 RTC 计数模块的操作开始。

bit	描述
0	RTC 即将停止

1	年/月/日/小时/分/秒/星期寄存器的值向年/月/日/小时/分/秒/星期计数器传输，RTC 计数模块开始计数操作。
---	---

22.7.2 控制寄存器 2 (RTCCTL2)

控制寄存器 2 控制 RTC 计数模块的操作。

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	Reserved							
属性	R							
初始值	0x00							

位	23	22	21	20	19	18	17	16
字段名	Reserved							
属性	R							
初始值	0x00							

位	15	14	13	12	11	10	9	8
字段名	Reserved							
属性	R							
初始值	00000							

位	7	6	5	4	3	2	1	0
字段名	Reserved							
属性	R							
初始值	0000000							

寄存器功能

[bit31:11] Reserved: 保留位，保留位的读取值总是“0”。

[bit10] TCRUN: 定时器操作位

该位表示定时器的操作状态。

如果定时器控制位为“0”，当定时器溢出时，该位将被硬件清“0”。如果定时器控制位为“1”，该位将保持为“1”直到定时器开始位（TCST）变为“0”为止。

在定时器工作中如果将 TCST 清“0”，定时器将停止工作并且 TCRUN 将自动清“0”。

bit	描述
0	定时器没有工作
1	定时器在工作

[bit9] TCEN: 定时器控制位

该位选择定时器计数为对设置时间的单次计数（完成一次定时则停止）还是周期计数（一次定时完成后重新加载设置时间进行下一次定时）。

bit	描述

0	单次定时（对设置时间进行单次计数）
1	周期定时（对设置时间进行周期计数）

[bit8] TCST: 定时器开始位

该位控制定时器操作开始。

如果定时器控制位为“0”，当定时器的计数完成时，该位将被硬件清“0”。

查询定时器操作位（TCRUN）了解定时器操作状态。如果要重写计时器设置寄存器，将此位设置为“0”，以立即停止此位并重写计时器设置寄存器，然后将计时器设置寄存器复位为“1”，启动计时器设置寄存器的操作。

bit	描述
0	定时器停止工作
1	定时器开始工作

[bit7:1] Reserved: 保留位，保留位的读取值总为“0”。

[bit0] RCTRL: 年/月/日/小时/分/秒/星期计数器值读取控制位

如果该位置“1”，年/月/日/小时/分/秒/星期计数器的值将向年/月/日/小时/分/秒/星期寄存器传输，传输完成时，该位自动清“0”。

在 RMW 访问模式下读该位值总是“0”。

bit		描述
读	0	年/月/日/小时/分/秒/星期计数器的值向年/月/日/小时/分/秒/星期寄存器的传输已完成
	1	年/月/日/小时/分/秒/星期计数器的值在向年/月/日/小时/分/秒/星期寄存器传输过程中
写	0	无效
	1	年/月/日/小时/分/秒/星期计数器的值向年/月/日/小时/分/秒/星期寄存器的传输开始

22.7.3 计数周期设置寄存器 (CCS)

计数周期设置寄存器 (CCS) 存储半秒发生器（用于日期、时间和定时器）的重载计数值。

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	Reserved							
属性	R							
初始值	0x00							

位	23	22	21	20	19	18	17	16
字段名	CC23	CC22	CC21	CC20	CC19	CC18	CC17	CC16
属性	R/W							
初始值	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8
字段名	CC15	CC14	CC13	CC12	CC11	CC10	CC9	CC8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31:24] Reserved: 保留位，保留位的读取值总是“0”。

[bit23:0] CC23 ~ CC0: 计数周期设置位

该寄存器值作为半秒发生器的 0.5 秒计数值，当计数值变为“0”时，重新加载到半秒发生器。

通过以下公式计算计数周期寄存器 (CCS) 的设置值：

$$CCS = (0.5 [s] / (2 \times MCLK [s])) - 1$$

注意：

- 设置 CCS 寄存器时，确保 ST 的值为“0”（RTC 计数模块停止操作）以及 TCST 的值为“0”（定时器停止操作）。
- 设置该寄存器值应大于等于“7”，否则如果读年/月/日/小时/分/秒/星期计数器的值将得不到正确值。

22.7.4 日期寄存器 (DATE)

日期寄存器 (DATE) 显示 RTC 计数模块的日期信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved		HD1	HD0	LD3	LD2	LD1	LD0
属性	R		R/W	R/W	R/W	R/W	R/W	R/W
初始值	00	0	0	0	0	0	0	0

寄存器功能

[bit7:6] Reserved: 保留位，保留位的读取值总是“0”。

[bit5:4] HD1, HD0: 日期寄存器，存储日期的十位。

[bit3:0] LD3 ~ LD0: 日期寄存器，存储日期的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.5 小时寄存器 (HOUR)

小时寄存器 (HOUR) 显示 RTC 计数模块的小时信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved		HH1	HH0	LH3	LH2	LH1	LH0
属性	R		R/W	R/W	R/W	R/W	R/W	R/W
初始值	00	0	0	0	0	0	0	0

寄存器功能

[bit7:6] Reserved: 保留位，保留位的读取值为“0”。

[bit5:4] HH1, HH0: 小时寄存器，存储小时信息的十位。

0 ~ 2: 允许

3: 禁止设置

[bit3:0] LH3 ~ LH0: 小时寄存器个位，存储小时信息的个位

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.6 分钟寄存器 (MIN)

分钟寄存器 (MIN) 显示 RTC 计数模块的分钟信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved	HMI2	HMI1	HMI0	LMI3	LMI2	LMI1	LMI0
属性	R	R/W						
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] Reserved: 保留位，保留位的读取值为“0”。

[bit6:4] HMI2 ~ HMI0: 分钟寄存器，存储分钟信息的十位

0 ~ 5: 允许

6, 7: 禁止设置

[bit3:0] LMI3 ~ LMI0: 分钟寄存器，存储分钟信息的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.7 秒寄存器 (SEN)

秒寄存器 (SEN) 显示 RTC 计数模块的秒信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved	HS2	HS1	HS0	LS3	LS2	LS1	LS0
属性	R	R/W						
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] Reserved: 保留位，保留位读取值总为“0”。

[bit6:4] HS2 ~ HS0: 秒寄存器，存储秒信息的十位。

0 ~ 5: 允许

6, 7: 禁止设置

[bit3:0] LS3 ~ LS0: 秒寄存器，存储秒信息的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.8 年寄存器 (YEAR)

年寄存器 (YEAR) 显示 RTC 计数模块的年信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	HY3	HY2	HY1	HY0	LY3	LY2	LY1	LY0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7:4] HY3 ~HY0: 年寄存器，存储年信息的十位。

0 ~ 9: 允许

A ~ F: 禁止设置

[bit3:0] LY3 ~ LY0: 年寄存器，存储年信息的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.9 月寄存器 (MON)

月寄存器 (MON) 显示 RTC 计数模块的月信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved			HMON0	LMON3	LMON2	LMON1	LMON0
属性	R			R/W	R/W	R/W	R/W	R/W
初始值	000			0	0	0	0	0

寄存器功能

[bit7:5] Reserved: 保留位，保留位的读取值总为“0”。

[bit4] HMON0: 月寄存器，存储月寄存器的十位。

[bit3:0] LMON3 ~ LMON0: 月寄存器，存储月寄存器的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.10 星期寄存器 (WEEK)

星期寄存器 (WEEK) 显示 RTC 计数模块的星期信息。该寄存器值格式为 BCD。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved					WK2	WK1	WK0
属性	R					R/W	R/W	R/W
初始值	00000					0	0	0

寄存器功能

[bit7:3] Reserved: 保留位，保留位的读取值总是“0”。

[bit2:0] WK2 ~ WK0: 星期寄存器，存储星期信息。

"0": 星期天

"1": 星期一

"2": 星期二

"3": 星期三

"4": 星期四

"5": 星期五

"6": 星期六

"7": 禁止设置

22.7.11 日闹钟寄存器 (ADATE)

日闹钟寄存器 (ADATE) 显示闹钟设置的日期信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved		HD1	HD0	LD3	LD2	LD1	LD0
属性	R		R/W	R/W	R/W	R/W	R/W	R/W
初始值	00		0	0	0	0	0	0

寄存器功能

[bit7:6] Reserved: 保留位，保留位的读取值为“0”。

[bit5:4] HD1 ~ HD0: 日闹钟寄存器，存储日闹钟信息的十位。

[bit3:0] LD3 ~ LD0: 日闹钟寄存器，存储日闹钟信息的个位

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.12 小时闹钟寄存器 (AHOUR)

小时闹钟寄存器 (AHOUR) 显示闹钟设置的小时信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved		HH1	HH0	LH3	LH2	LH1	LH0
属性	R	R/W						
初始值	00	0	0	0	0	0	0	0

寄存器功能

[bit7:6] Reserved: 保留位，保留位的读取值为“0”。

[bit5:4] HH1, HH0: 小时闹钟寄存器，存储小时闹钟信息的十位。

0 ~ 2: 允许

3 ~ F: 禁止设置

[bit3:0] LH3 ~ LH0: 小时闹钟寄存器，存储小时闹钟信息的个位

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.13 分闹钟寄存器 (AMIN)

分闹钟寄存器 (AMIN) 显示闹钟设置的分钟信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved	HM2	HM1	HM0	LM3	LM2	LM1	LM0
属性	R	R/W						
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] Reserved: 保留位，保留位的读取值总是“0”。

[bit6:4] HM2 ~ HM0: 分闹钟寄存器，存储分闹钟信息的十位。

0 ~ 5: 允许

6, 7: 禁止设置

[bit3:0] LM3 ~ LM0: 分闹钟寄存器，存储分闹钟信息的个位

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.14 秒闹钟寄存器 (ASEN)

秒闹钟寄存器 (ASEN) 显示闹钟设置的秒信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved	HS2	HS1	HS0	LS3	LS2	LS1	LS0
属性	R	R/W						
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] Reserved: 保留位，保留位的读取值总是“0”。

[bit6:4] HS2 ~ HS0: 秒闹钟寄存器，存储秒闹钟的十位。

0 ~ 5: 允许

6, 7: 禁止设置

[bit3:0] LS3 ~ LS0: 秒闹钟寄存器，存储秒闹钟的个位。

0 ~ 9: 使能

A ~ F: 禁止设置

22.7.15 年闹钟寄存器 (AYEAR)

年闹钟寄存器 (AYEAR) 显示闹钟设置的年信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	HY3	HY2	HY1	HY0	LY3	LY2	LY1	LY0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7:4] HY3 ~ HY0: 年闹钟寄存器，存储年闹钟信息的十位。

0 ~ 9: 允许

A ~ F: 禁止设置

[bit3:0] LY3 ~ LY0: 年闹钟寄存器，存储年闹钟信息的个位

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.16 月闹钟寄存器 (AMON)

月闹钟寄存器 (AMON) 显示闹钟设置的月信息。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved		HMON0	LMON3	LMON2	LMON1	LMON0	
属性	R		R/W	R/W	R/W	R/W	R/W	
初始值	000		0	0	0	0	0	

寄存器功能

[bit7:5] Reserved: 保留位，保留位的读取值总是“0”。

[bit4] HMON0: 月闹钟寄存器，存储月闹钟信息的十位。

[bit3:0] LMON3 ~ LMON0: 月闹钟寄存器，存储月闹钟信息的个位。

0 ~ 9: 允许

A ~ F: 禁止设置

22.7.17 定时器设置寄存器 (TVAL)

定时器设置寄存器 (TVAL) 用于设置定时时间，该定时时间可用于单次定时或周期定时。值得大小可以从 1 秒到一天。

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	Reserved							
属性	R							
初始值	0x00							

位	23	22	21	20	19	18	17	16
字段名	Reserved							VAL17
属性	R							R/W
初始值	000000							0

位	15	14	13	12	11	10	9	8
字段名	VAL15	VAL14	VAL13	VAL12	VAL11	VAL10	VAL9	VAL8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	VAL7	VAL6	VAL5	VAL4	VAL3	VAL2	VAL1	VAL0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31:18] Reserved: 保留位，保留位的读取值总是“0”。

[bit17:0] VAL17 ~ VAL0: 定时器设置寄存器，该寄存器可以设置一天的定时时间用于单次定时或周期定时。

有效值为以 0.5 秒间隔从 1 秒到 1 天。

通过以下公式计算设置值：

$$\text{VAL [17:0]} = (\text{设置时间 [s]} \times 2) - 1$$

1 ~ 172799: 允许

0,172800 to 262143: 禁止设置

22.8 使用注意事项

使用 RTC 计数模块是注意以下几点：

- 满足频率条件： $PCLK \geq MCLK / 2$
- 当闹钟使能位(RTCCTL1:YEAE, RTCCTL1:MONE, RTCCTL1:DAYE, RTCCTL1:HOU, RTCCTL1:MINE, RTCCTL1:SENE)为“0”时，更改相应闹钟寄存器。
- 当闹钟使能位(RTCCTL1:YEAE, RTCCTL1:MONE, RTCCTL1:DAYE, RTCCTL1:HOU, RTCCTL1:MINE, RTCCTL1:SENE)置“1”后，中断可能立即产生。中断产生以后，读取并检查年/月/日/小时/分钟/秒计数器值。

23. RTC 时钟控制模块

本章介绍 RTC 时钟控制模块的功能和操作。

23.1 概述

本节介绍 RTC 时钟控制模块的功能概述。

RTC 时钟控制模块

RTC 时钟控制模块具有以下功能：

- 产生用于 RTC 计数模块的 RTC 时钟 (MCLK)。
- 选择副时钟或低速 CR 时钟作为输入时钟 (RIN_CLK)。
- 输出分频时钟到 SUBOUT 外部引脚。
- 输出 0.5 秒脉冲或者 1 秒脉冲到 RTCCO 外部引脚。
- 对输入时钟 (副时钟) 由温度引起的频率抖动进行频率校正。

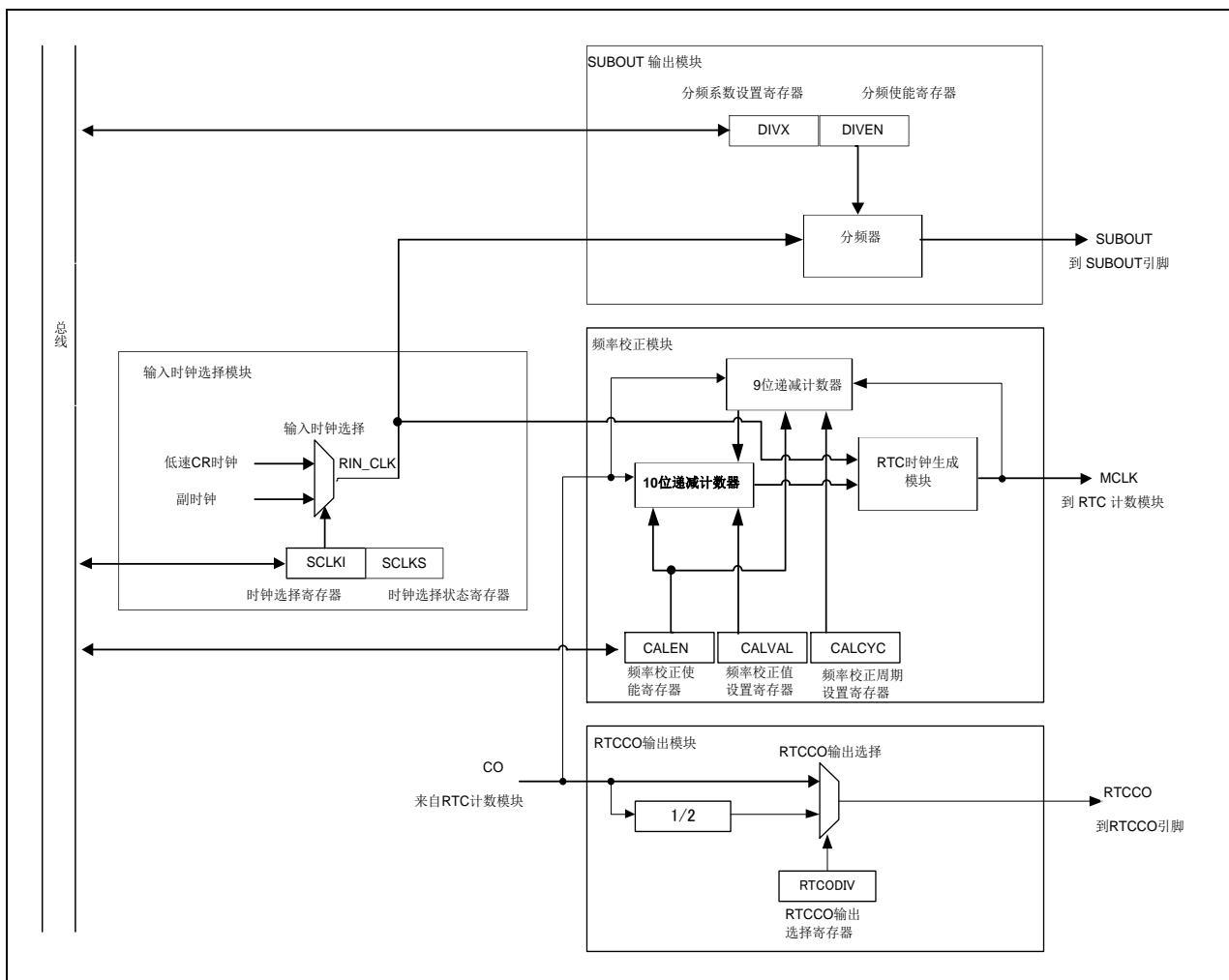
(以上频率校正功能是基于外部有温度传感器连接。)

23.2 框图

本节介绍 RTC 时钟控制模块的框图。

RTC 时钟控制模块框图

图 23.2-1 RTC 时钟控制模块框图



输入时钟选择模块

使用时钟选择寄存器（SCLKI）选择低速 CR 时钟或者副时钟作为输入时钟。

频率校正模块

频率校正模块根据 CALVAL 设置值屏蔽 RIN_CLK 一定周期数，以输出频率校正后的 MCLK。

SUBOUT 输出模块

SUBOUT 输出模块产生分频时钟作为 SUBOUT 外部引脚输出。

在 deep standby RTC 模式下，没有 SUBOUT 输出。

RTCCO 输出模块

RTCCO 模块输出来自 RTC 计数模块的 CO 信号或者 CO 信号的 2 分频到 RTCCO 外部引脚。

在 deep standby RTC 模式下， RTCCO 外部引脚没有输出。

23.3 操作

本节介绍 RTC 时钟控制模块的操作。

23.3.1 频率校正

存在频率抖动的 RIN_CLK 是被校正时钟，屏蔽该时钟一定数量的周期后输出的 MCLK 则为频率校正后的时钟。

频率校正的周期由频率校正周期设置寄存器（CALCYC）设置。

每次频率校正时，对 RIN_CLK 进行屏蔽的时钟周期个数由频率校正值设置寄存器（CALVAL）设置。

频率校正操作

图 23.3.1-1 频率校正操作示例（CCS=8190, CALCYC=19）

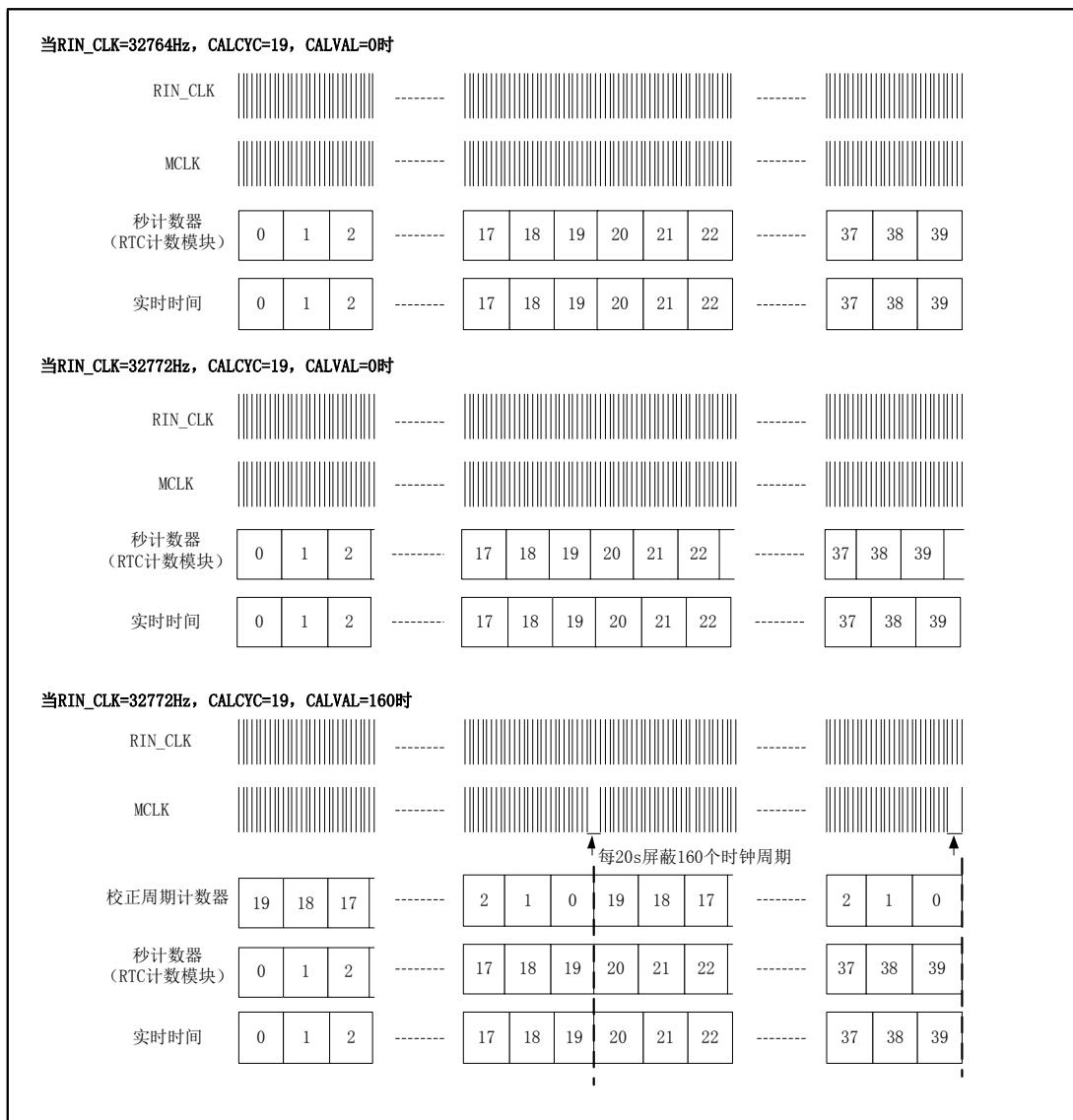
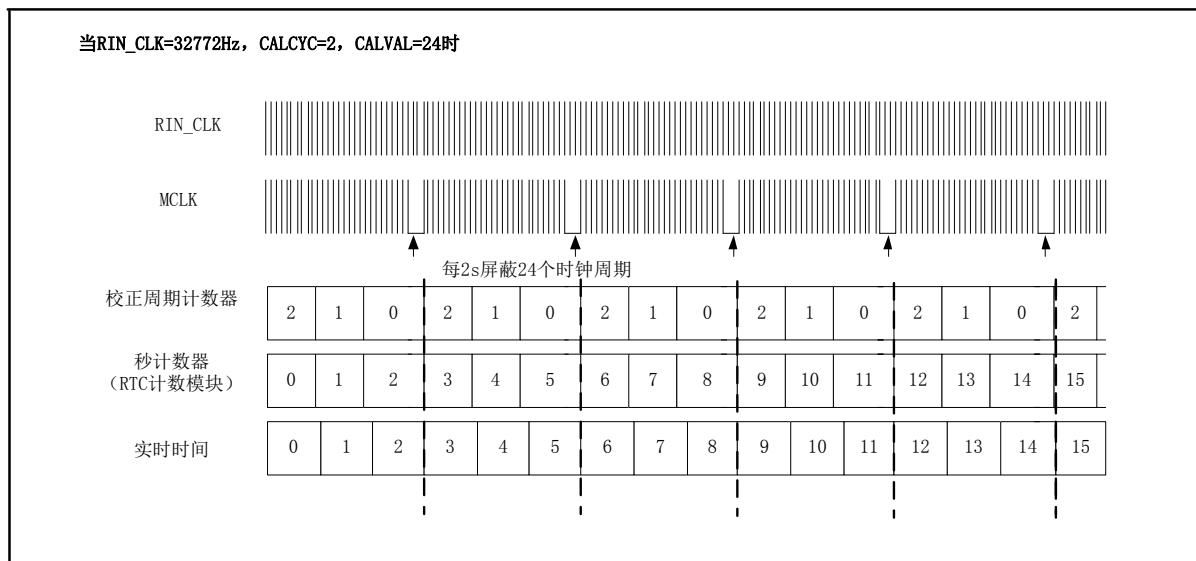


图 23.3.1-2 频率校正操作示例 (CCS=8190, CALCYC=3)



频率校正范围

表 23.3.1-1 频率校正范围示例 (CALCYC=19) (理想值)

CALVAL	CCS=8190		CCS=8189	
	校正率 [ppm]	副时钟频率[Hz]	校正率[ppm]	副时钟频率[Hz]
0	122.1	32764.00	244.1	32760.00
1	120.5	32764.05	242.6	32760.05
2	119.0	32764.10	241.1	32760.10
:	:	:	:	:
79	1.5	32767.95	123.6	32763.95
80	0.0	32768.00	122.1	32764.00
81	-1.5	32768.05	120.5	32764.05
:	:	:	:	:
150	-120.5	32771.95	1.5	32767.95
160	-122.1	32772.00	0.0	32768.00
161	-123.6	32772.05	-1.5	32768.05
:	:	:	:	:
318	-363.2	32779.90	-241.1	32775.90
319	-364.7	32779.95	-242.6	32775.95
320	-366.2	32780.00	-244.1	32776.00

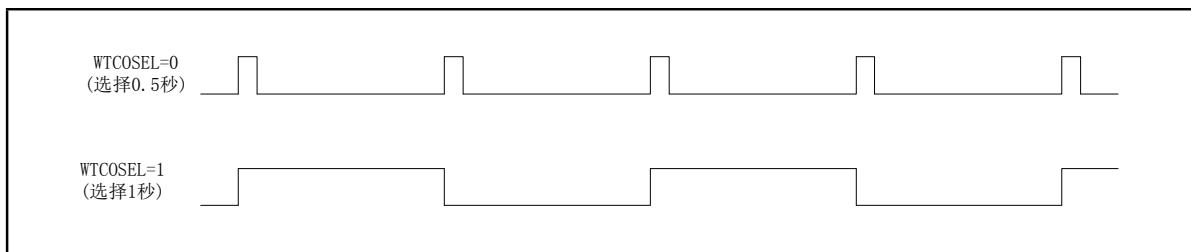
表 23.3.1-2 频率校正范围 (CALCYC=59) (理想值)

CALVAL	CCS=8190		CCS=8189	
	校正率[ppm]	副时钟频率 [Hz]	校正率[ppm]	副时钟频率[Hz]
0	122.1	32764.00	244.1	32760.00
1	121.6	32764.02	243.6	32760.02
2	121.1	32764.03	243.1	32760.03
:	:	:	:	:
23	0.5	32767.98	122.6	32763.98
24	0.0	32768.00	122.1	32764.00
24	-0.5	32768.02	121.6	32764.02
:	:	:	:	:
47	-121.6	32771.98	0.5	32767.98
48	-122.1	32772.00	0.0	32768.00
48	-122.6	32772.02	-0.5	32768.02
:	:	:	:	:
958	-365.2	32779.97	-243.1	32775.97
959	-365.7	32779.98	-243.6	32775.98
960	-366.2	32780.00	-244.1	32776.00

23.3.2 RTCCO 外部引脚输出

通过设置 RTCCO 时钟选择寄存器(RTCODIV)，选择输出到 RTCCO 外部引脚的信号来自 RTC 计数模块的 CO 信号（0.5 秒）还是 CO 的 2 分频信号（1 秒）。

图 23.3.2-1 RTCCO 外部引脚输出示例



CO 信号请参见“RTC 计数模块”章节。

23.4 设置过程

本节介绍 RTC 是时钟控制模块的设置过程。

频率校正设置过程 (时钟源为副时钟)

- 1) 输入时钟选择寄存器 (SCLKI) 写 “0” 以选择副时钟。
- 2) 读时钟选择状态寄存器 (SCLKS) 并等待其值变为 “10”，表示副时钟选择成功。
- 3) 写校正周期到频率校正周期设置寄存器 (CALCYC)。
- 4) 写校正值到频率校正值设置寄存器 (CALVAL)。

由以下公式获取 CALVAL 的设置值：

$$\text{CALVAL} = [\text{待校正频率} - (\text{CCS}+1) \times 4] \times (\text{CALCYC}+1)$$

- 5) 频率校正使能寄存器 (CALEN) 写 “1” 以开始频率校正。

SUBOUT 输出设置过程

- 1) 分频输出使能寄存器 (DIVEN) 写 “0”，分频器将停止操作，SUBOUT 外部引脚将输出低电平。
- 2) 读分频器状态寄存器(DIVRDY)，等待其值变为 “0”，表示分频器已停止操作。
- 3) 写分频系数到分频系数寄存器 (CLKDIVX)。
- 4) 写 “1” 到分频输出使能寄存器(DIVEN)，分频器开始输出。

23.5 寄存器

本节介绍 RTC 时钟控制模块的寄存器。

RTC 时钟控制模块寄存器列表

表 23.5-1 RTC 时钟控制模块寄存器列表

缩写	寄存器名	章节
SCLKI	时钟选择寄存器	23.5.1
SCLKS	时钟选择状态寄存器	23.5.2
CALVAL	频率校正值设置寄存器	23.5.3
CALEN	频率校正使能寄存器	23.5.4
CLKDIVX	分频系数设置寄存器	23.5.5
CLKDIVC	分频输出使能寄存器	23.5.6
CALCYC	校正周期设置寄存器	23.5.7
RTCODIV	RTCCO 输出选择寄存器	23.5.8

23.5.1 时钟选择寄存器 (SCLKI)

时钟选择寄存器(SCLKI)用于输入时钟(RIN_CLK)的选择。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved						SCLKI	
属性	R						R/W	
初始值	0000000						0	

寄存器功能

[bit7:1] Reserved: 保留位，保留位的读取值总是“0”。

[bit0] SCLKI: 输入时钟选择，该位用于以下输入时钟 (RIN_CLK) 的选择。

bit	描述
0	RIN_CLK 输入选择副时钟.
1	RIN_CLK 输入选择低速 CR 时钟

23.5.2 时钟选择状态寄存器 (SCLKS)

时钟选择状态寄存器 (SCLKS)显示输入时钟(RIN_CLK)的选择状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved						SCLKS	
属性	R						R	
初始值	000000						00	

寄存器功能

[bit7:2] Reserved: 保留位，保留位的读取值为“0”。

[bit1:0] SCLKS: 时钟选择状态，该位显示输入时钟 (RIN_CLK) 的选择状态。

bit1:0	描述
0x	RIN_CLK 没有输入
10	已选择副时钟作为 RIN_CLK 输入
11	已选择低速 CR 时钟作为 RIN_CLK 输入

注意：

- 软件复位、RTC 复位以及 APB1 总线复位不能初始化该寄存器。

23.5.3 频率校正值设置寄存器(CALVAL)

频率校正值设置寄存器(CALVAL)设置频率校正时需要对 RIN_CLK 进行屏蔽的时钟周期个数。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserve d				CALVAL			
属性	R				R/W			
初始值	0				0000000			

寄存器功能

[bit7] Reserved: 保留位，保留位的读取值为“0”。

[bit6:0] CALVAL: 频率校正值设置，频率校正周期在寄存器 CALCYC 中设置。

如果 CALCYC 设置为“19”，CALVAL 设置为“8”，那么 RIN_CLK 每 20s 将屏蔽 8 个时钟周期，输出 MCLK 到 RTC 计数模块。

CALVAL 设置值的计算方法请参见“4.操作过程”。

注意：

- 软件复位和 APB1 总线复位不能初始化该寄存器。

23.5.4 频率校正使能寄存器 (CALEN)

频率校正使能寄存器(CALEN)使能频率校正。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名				Reserved				CALEN
属性				R				R/W
初始值				0000000				0

寄存器功能

[bit7:1] Reserved: 保留位，保留位的读取值为“0”。

[bit0] CALEN: 频率校正使能

bit	描述
0	禁止频率校正
1	使能频率校正

注意：

- 软件复位和 APB1 总线复位不能初始化该寄存器。

23.5.5 分频系数设置寄存器 (CLKDIVX)

分频系数设置寄存器(CLKDIVX)设置分频系数。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved					DIVX		
属性	R					R/W		
初始值	0000					0000		

寄存器功能

[bit7:4] Reserved: 保留位，保留位的读取值为“0”。

[bit3:0] DIVX: 设置分频系数，根据设置的分频系数对输入时钟 (RIN_CLK) 进行分频，然后输出到 SUBOUT 外部引脚。

bit3:0	描述
0000	无分频
0001	2 分频
0010	4 分频
0011	8 分频
0100	16 分频
0101	32 分频
0110	64 分频
0111	128 分频
1000	256 分频
1001	512 分频
其它	禁止设置

注意：

- 当分频输出使能寄存器 (CLKDIVC) 中的 DIVEN 位和 DIVRDY 位均为“0”时，才能写数据到 CLKDIVX。

软件复位和 APB1 总线复位不能初始化该寄存器。

23.5.6 分频输出使能寄存器 (CLKDIVC)

分频输出使能寄存器 (CLKDIVC) 使能分频器的输出。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved						DIVRDY	DIVEN
属性	R						R	R/W
初始值	000000						0	0

寄存器功能

[bit7:2] Reserved: 保留位，保留位的读取值为“0”。

[bit1] DIVRDY: 分频状态，该位显示分频器的操作状态。

bit	描述
0	分频器为停止状态。SUBOUT 外部引脚输出为低
1	分频器在工作中

[bit0] DIVEN: 分频器使能位，该位使能分频操作。

bit	描述
0	停止分频操作
1	使能分频操作

注意：

- 软件复位和 APB1 总线不能初始化该寄存器。

23.5.7 频率校正周期寄存器 (CALCYC)

频率校正周期寄存器 (CALCYC) 设置频率校正的周期。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	Reserved							CALCYC
属性	R/W							R/W
初始值	0000000							0

位	15	14	13	12	11	10	9	8
字段名	CALCYC							
属性	R/W							
初始值	010011							

寄存器功能

[bit15:9] Reserved: 保留位，保留位的读取值为“0”。

[bit8:0] CALCYC: 频率校正周期，该寄存器的值为频率校正周期减“1”。

比如，如果设置为“0”，那么频率校正周期为 1s；如果设置为“19”，那么频率校正周期则为 20s。

注意：

- 软件复位和 APB1 总线复位不能初始化该寄存器。

23.5.8 RTCCO 输出选择寄存器 (RTCODIV)

RTCCO 输出选择寄存器 (RTCODIV) 选择 RTCCO 的输出为 CO 信号还是 CO 信号的 2 分频。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	Reserved							CODIV2
属性	R							R/W
初始值	0000000							0

寄存器功能

[bit7:1] Reserved: 保留位，保留位的读取值为“0”。

[bit0] CODIV2: 频率校正使能

bit	描述
0	输出为来自 RTC 计数模块的 CO 信号
1	输出为 CO 信号的 2 分频

注意：

- 软件复位或 APB1 总线复位不能初始化该寄存器。

24. 复合定时器 I/O 选择功能

本章介绍复合定时器的 I/O 选择功能。

24.1 概要

复合定时器 I/O 选择功能通过设定 I/O 模式决定复合定时器的信号(外部时钟/外部启动触发/波形)的输出或输入方法。

另外，通过切换定时器功能，复合定时器的每个通道都可作为以下定时器来使用：

- 16 位 PWM 定时器
- 16 位 PPG 定时器
- 16/32 位重载定时器
- 16/32 位 PWC 定时器

每 2 路通道可以选择以下 9 种 I/O 模式之一。

具有多路通道软件同时启动的功能，最大支持 8 路通道软件同时启动。

- I/O 模式 0：16 位定时器标准模式

复合定时器在每路通道单独工作。

- I/O 模式 1：定时器全模式

复合定时器偶数通道的信号单独分配到外部引脚的工作模式。

- I/O 模式 2：外部触发共享模式

该模式可同时输入外部启动触发信号到复合定时器的 2 个通道。使用此模式，可同时启动复合定时器的该 2 路通道。

- I/O 模式 3：其他通道触发共享模式

将另一个通道的外部信号用作外部启动触发进行启动的模式。通道 0 和通道 1 不可设定为此模式。

- I/O 模式 4：定时器启动/停止模式

使用偶数通道控制奇数通道的启动/停止的模式。奇数通道在偶数通道的输出信号的上升沿启动、下降沿停止。

- I/O 模式 5：软件同步启动模式

使用软件同时启动多个通道的模式。

- I/O 模式 6：软件启动定时器启动/停止模式

该模式使用偶数通道控制奇数通道的启动/停止。使用软件启动偶数通道。奇数通道在偶数通道的输出信号的上升沿启动、下降沿停止。

- I/O 模式 7：定时器启动模式

使用偶数通道控制奇数通道的启动的模式。奇数通道在偶数通道的输出信号的上升沿启动。

- I/O 模式 8：其他通道触发共享定时器启动/停止模式

将一个通道的外部信号用作另一个通道的外部启动触发进行启动的模式。通道 0 和通道 1 不可设定为此模式。

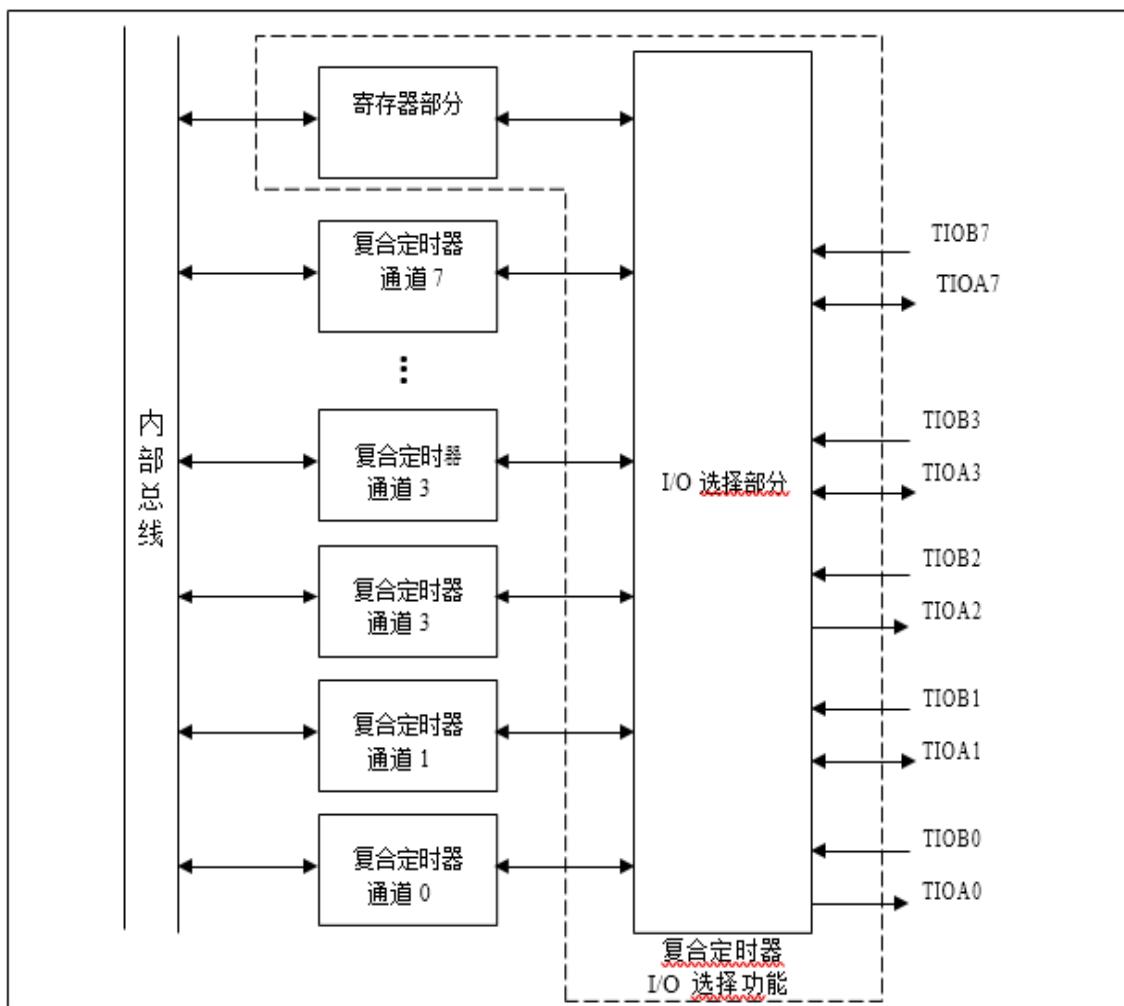
24.2 结构

复合定时器 I/O 选择功能由以下模块构成。

框图

复合定时器 I/O 选择功能的框图如图 24.2-1 所示。

图 24.2-1 复合定时器 I/O 选择功能的框图



- I/O 选择部分

为每路通道选择复合定时器的 I/O 模式的电路

- 复合定时器(通道 0 ~通道 7)

复合定时器通道通道 0 ~通道 7 (最多 8 通道)

- 寄存器部分

复合定时器 I/O 选择功能的寄存器部分

24.3 I/O 模式

本节介绍用于配置复合定时器 I/O 模式的引脚和各 I/O 模式。

24.3.1 引脚

以下介绍用于配置复合定时器 I/O 模式的引脚。

复合定时器的每路通道有 2 种外部引脚和 5 种内部信号。同时，复合定时器 I/O 选择功能有 2 种内部信号。将外部引脚和内部信号连接，将连接的信号(外部时钟(ECK 信号)/外部启动触发(TGIN 信号)/波形(TIN 信号) 输入或输出复合定时器。可通过设定复合定时器的 I/O 模式连接外部引脚和内部信号。使用的引脚和 I/O 信号因不同的 I/O 模式而异。

外部引脚

- TIOA 引脚

该引脚用于输出复合定时器的波形(TOUT 信号)、或输入外部启动触发(TGIN 信号)。

- TIOB 引脚

该引脚用于输入外部启动触发(TGIN 信号)/外部时钟(ECK 信号)/其他通道的波形(TIN 信号)。

内部信号

与上述的外部引脚连接或输入自其他通道的输出信号，使信号输入或输出至复合定时器。

- TOUT 信号

复合定时器的输出波形(16/32 位 PWC 定时器不可使用)。

- ECK 信号

复合定时器的外部时钟(16/32 位 PWC 定时器不可使用)。

选择外部时钟作为计数时钟时输入该信号。

- TGIN 信号

复合定时器的外部启动信号(16/32 位 PWC 定时器不可使用)。

若选定外部启动信号的有效沿后，复合定时器检测到此信号的有效边沿则启动。

- TIN 信号

复合定时器的输入波形。属于被测量波形(16/32 位 PWC 定时器不可使用)。

- DTRG 信号

复合定时器的触发输入。复合定时器在此信号的下降沿时停止工作。

- COUT 信号

复合定时器 I/O 选择功能的触发输出。输出至复合定时器其他通道的信号。

- CIN 信号

复合定时器 I/O 选择功能的触发输入。此信号来自复合定时器的其他通道的输入信号。

外部引脚和内部信号的连接

复合定时器的 I/O 模式的设定将外部引脚和内部信号连接。

I/O 模式和连接引脚的对应关系如表 24.3 1-1 所示。

表 24.3.1-1 I/O 模式和连接引脚的对应关系

I/O 模式	TIOAn (偶数通道)		TIOBn (偶数通道)		TIOAn+1 (奇数通道)		TIOBn+1 (奇数通道)	
	连接对象	I/O	连接对象	I/O	连接对象	I/O	连接对象	I/O
0	通道 n 的 TOUT	输出	通道 n 的 ECK/TGIN/ TIN	输入	通道 n+1 的 TOUT	输出	通道 n+1 的 ECK/TGIN/ TIN	输入
1	通道 n 的 TOUT	输出	通道 n 的 ECK	输入	通道 n 的 TGIN	输入	通道 n 的 TIN	输入
2	通道 n 的 TOUT	输出	通道 n/通道 n+1 的 ECK/TGIN/ TIN *1	输入	通道 n+1 的 TOUT	输出	不使用	
3	通道 n 的 TOUT	输出	不使用		通道 n+1 的 TOUT	输出		
4	通道 n 的 TOUT	输出	通道 n 的 ECK/TGIN/ TIN	输入	通道 n+1 的 TOUT	输出		
5	通道 n 的 TOUT	输出	不使用		通道 n+1 的 TOUT	输出		
6	通道 n 的 TOUT	输出			通道 n+1 的 TOUT	输出		
7	通道 n 的 TOUT	输出	通道 n 的 ECK/TGIN/ TIN	输入	通道 n+1 的 TOUT	输出	不使用	
8	通道 n 的 TOUT	输出	不使用		通道 n+1 的 TOUT	输出		

n : 偶数

通道 n: 偶数通道

通道 n+1: 奇数通道

*1: 使用外设时钟(PCLK)进行同步

24.3.2 I/O 模式

外部引脚的操作和复合定时器的启动/停止时序因 I/O 选择寄存器(CTSEL)设定而异。

I/O 模式 0 (16 位定时器标准模式)

此模式单独使用复合定时器的各通道。

此模式时，所用的外部引脚如表 24.3.2-1 所示。

表 24.3.2-1 设定为 I/O 模式 0 时所用的外部引脚

	偶数通道	奇数通道
输入引脚	1 路	1 路
输出引脚	1 路	1 路

所用的外部引脚的连接对象和输入输出信号如表 24.3.2-2 所示。

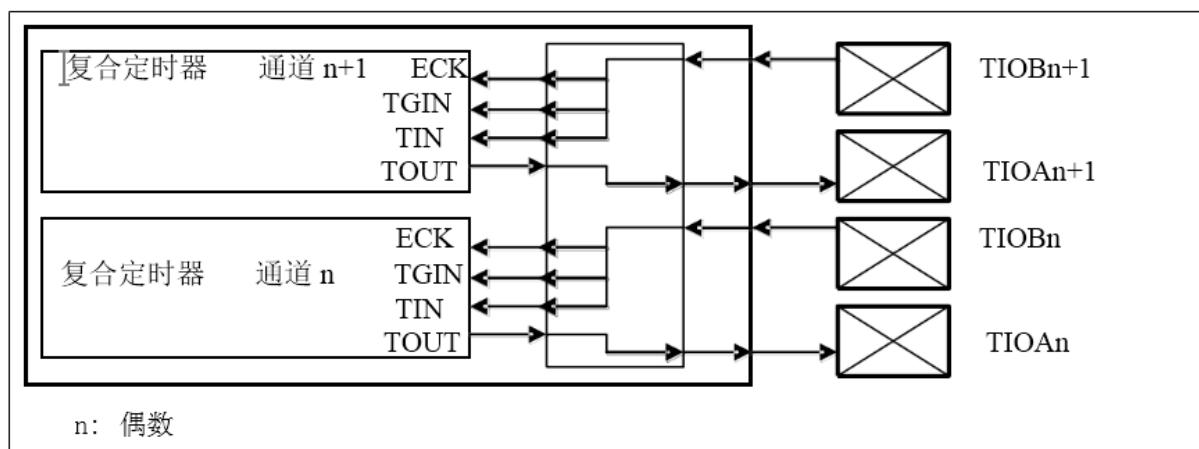
表 24.3.2-2 设定为 I/O 模式 0 时的外部引脚的连接对象和输入输出信号

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOA	输出	TOUT	输出复合定时器的波形
TIOB	输入	ECK/TGIN/TIN*	输入的信号作为以下之一使用 • 外部时钟(ECK 信号) • 外部启动触发(TGIN 信号) • 测定波形(TIN 信号)

*: 输入信号(ECK/TGIN/TIN 信号)的使用方法因复合定时器的定时器控制寄存器(TCR)设定而异。

模式 0 (16 位定时器标准模式)的框图如图 24.3.2-1 所示。

图 24.3.2-1 I/O 模式 0 (16 位定时器标准模式)的框图



模式 0 的连接如表 24.3.2-3 所示。

表 24.3.2-3 I/O 模式 0 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出
自 TIOBn 引脚的输入信号	作为 ECK/TGIN/TIN 输入至 通道 n
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出
自 TIOBn+1 引脚的输入信号	作为 ECK/TGIN/TIN 输入至 通道 n+1

n: 偶数

I/O 模式 1(定时器全模式)

此模式将偶数通道的信号分别分配到所有外部引脚。

设定为此模式时的使用外部引脚如表 24.3.2-4 所示。

表 24.3.2-4 设定为 I/O 模式 1 时的使用外部引脚

	偶数通道
输入引脚	3 路
输出引脚	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-5 所示。

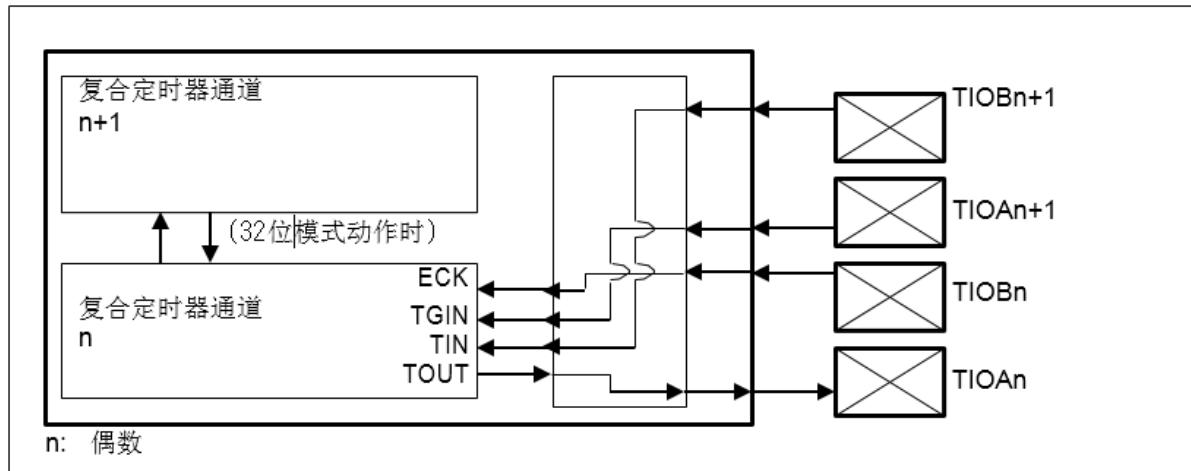
表 24.3.2-5 设定为 I/O 模式 1 时的外部引脚的连接对象和输入输出信号

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的TOUT	输出偶数通道的波形
TIOBn	输入	偶数通道的ECK	将外部时钟(ECK 信号)输入到偶数通道
TIOAn+1	输入	偶数通道的TGIN	将外部启动触发(TGIN 信号)输入到偶数通道
TIOBn+1	输入	偶数通道的TIN	将测定波形(TIN 信号)输入到偶数通道

n : 偶数

I/O 模式 1(定时器 全模式)的框图如图 24.3.2-2 所示。

图 24.3.2-2 I/O 模式 1(定时器 全模式)的框图



I/O 模式 1 的连接如表 24.3.2-6 所示。

表 24.3.2-6 I/O 模式 1 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出
自 TIOBn 引脚的输入信号	作为 ECK 信号输入至通道 n
TIOAn+1 引脚	作为 TGIN 信号输入至通道 n
TIOBn+1 引脚	作为 TIN 信号输入至通道 n

n：偶数

注意：

- 设定为此模式时，请将 GPIO 端口功能寄存器(FN)对应奇数通道的 TIOA 引脚(TIOA1, TIOA3, ...)
- 设定为端口输入模式。

I/O 模式 2 (外部触发共享模式)

在此模式下，2 路通道共享复合定时器的输入信号(ECK/TGIN/TIN)。

设定为此模式时使用的外部引脚如表 24.3.2-7 所示。

表 24.3.2-7 设定为 I/O 模式 2 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	1 路(2 路通道共享)	
输出引脚	1 路	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-8 所示。

表 24.3.2-8 设定为 I/O 模式 2 时的外部引脚的连接对象和输入输出信号

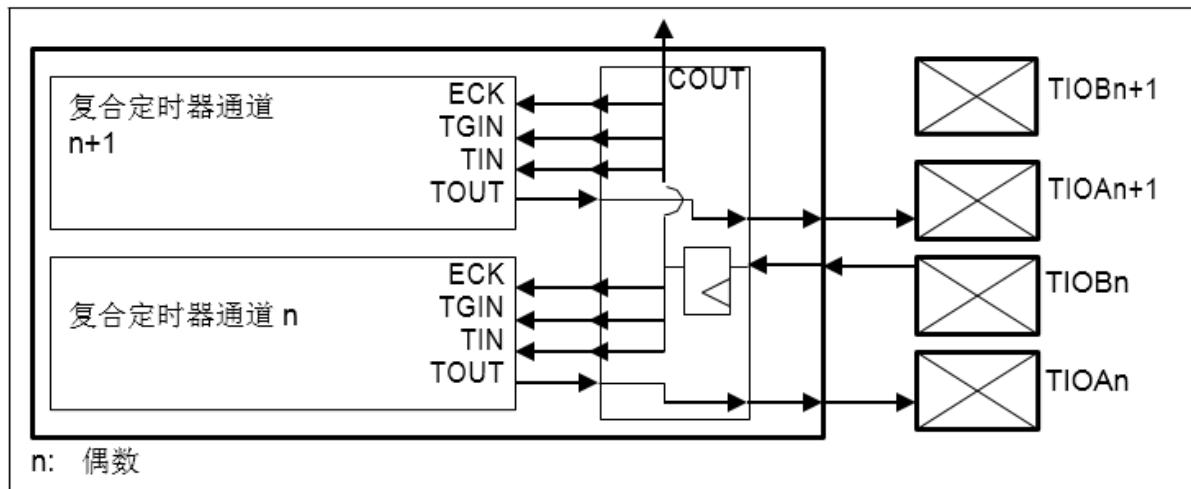
外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的TOUT	输出奇数通道的波形
TIOBn	输入	偶数/奇数通道的 ECK/TGIN/TIN*	输入偶数/奇数两方面的通道(使用外设时钟(PCLK)进行同步)作为以下之一使用 <ul style="list-style-type: none"> • 外部时钟(ECK 信号) • 外部启动触发(TGIN 信号) • 测定波形(TIN 信号)
TIOBn+1	-	-	不使用

n: 偶数

*: 输入信号(ECK/TGIN/TIN 信号)的使用方法因复合定时器的定时器控制寄存器(TCR)的设定而异。

I/O 模式 2 (外部触发共享模式)的框图如图 24.3.2-3 所示。

图 24.3.2-3 I/O 模式 2 (外部触发共享模式)的框图



I/O 模式 2 的连接如表 24.3.2-9 所示。

表 24.3.2-9 I/O 模式 2 的连接

连接源	连接对象	备注
通道 n 的TOUT 信号	从TIOAn 引脚输出	
自TIOBn 引脚的输入信号	<ul style="list-style-type: none"> · 作为ECK/TGIN/TIN 信号输入通道 n 和通道 n+1 · 作为COUT 信号输出到其他通道 	使用外设时钟 (PCLK) 进行同步化
通道 n+1 的TOUT 信号	从TIOAn+1 引脚输出	

n : 偶数

注意::

- 若将设定为此模式的通道的高位 2 路通道(n+2, n+3)设定为 I/O 模式 3(共享通道触发模式), 4 路通道可同时输入信号(ECK/TGIN/TIN)。
- (例: 将 通道 0 和 通道 1 设定为此模式, 若将 通道 2 和 通道 3 设定为 I/O 模式 3, 通道 0 ~ 通道 3 的 4 路通道可同时输入信号(ECK/TGIN/TIN)

I/O 模式 3 (其他通道触发共享模式)

在此模式下, 低 2 位端通道的 COUT 信号作为 CIN 信号输入, 作为 ECK/TGIN/TIN 信号使用。设定为此模式时使用的外部引脚如表 24.3.2-10 所示。

表 24.3.2-10 设定为 I/O 模式 3 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	不使用	
输出引脚	1 路	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-11 所示。

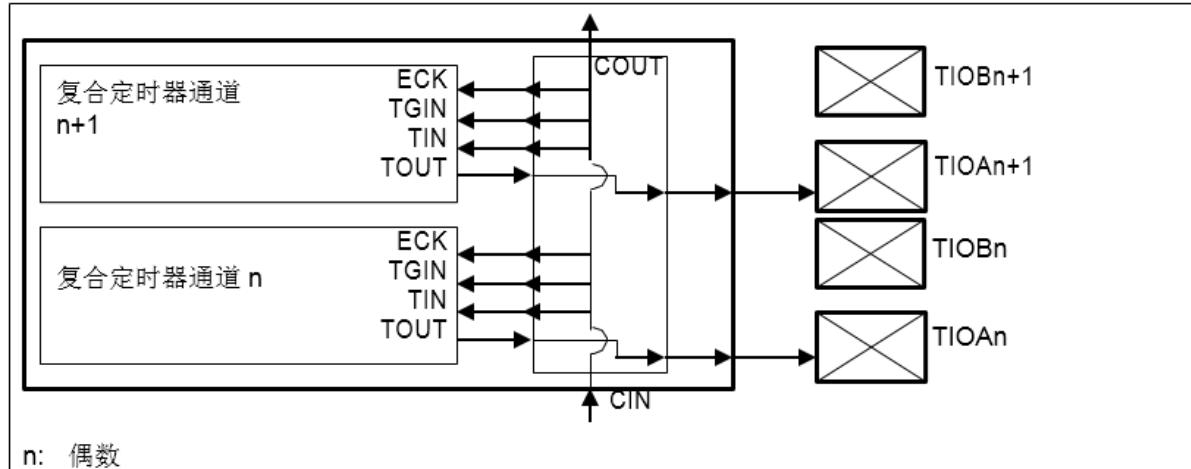
表 24.3.2-11 设定为 I/O 模式 3 时的外部引脚的连接对象和输入输出信号

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的TOUT	输出奇数通道的波形
TIOBn TIOBn+1	-	-	不使用

n : 偶数

I/O 模式 3 (其他通道触发共享模式)的框图如图 24.3.2-4 所示。

图 24.3.2-4 I/O 模式 3 (其他通道触发共享模式)的框图



I/O 模式 3 的连接如表 24.3.2-12 所示。

表 24.3.2-12 I/O 模式 3 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出
CIN 信号*	作为 ECK/TGIN/TIN 信号输入到 通道 n 和 通道 n+1 • 作为 COUT 信号输出到其他通道
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出

n : 偶数

*：从其他通道输入的 COUT 信号作为 CIN 信号。

可以被输入 通道 n/n+1 作为 ECK/TGIN/TIN 的 通道 n-2/n-1 信号如下：

I/O 模式 2 时使用外设时钟将 TIOBn-2 输入同步后的信号

I/O 模式 3 时从 通道 n-4/n-3 输入的触发信号

I/O 模式 4 时的 TIOAn-2 输出

I/O 模式 6 时的 TIOAn-2 输出

I/O 模式 7 时的 TIOAn-2 输出

I/O 模式 8 时从 通道 n-4/n-3 输入的触发信号

注意：

- 请使用复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0 位将触发输入沿设定为上升沿(EDGS1, EDGS0=0b01)。
- 设定为此模式的通道可将 2 路通道低位端(n-2, n-1)的 COUT 信号作为 CIN 信号输入

并使用。(例:如果将 通道 2, 通道 3 设定为此模式, 可使用 通道 0, 通道 1 的 COUT 信号)。通道 0 和通道 1 不可设定为此模式。

I/O 模式 4 (定时器启动/停止模式)

此模式可使用偶数通道控制奇数通道的启动/停止。

奇数通道在偶数通道输出波形(TOUT 信号)的上升沿启动、下降沿停止。

设定为此模式时使用的外部引脚如表 24.3.2-13 所示。

表 24.3.2-13 设定为 I/O 模式 4 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	1 路	不使用
输出引脚	1 路	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-14 所示。

表 24.3.2-14 设定为 I/O 模式 4 时的外部引脚的连接对象和输入输出信号

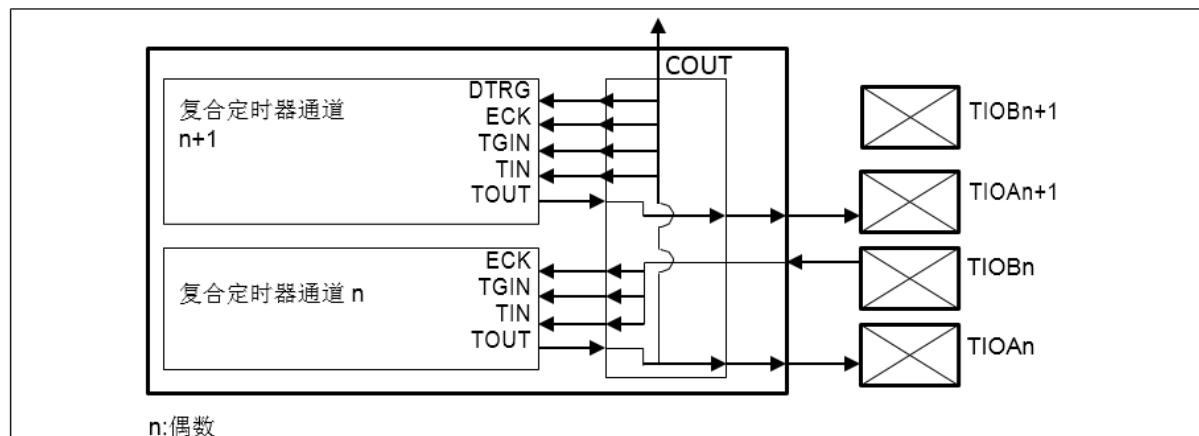
外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的TOUT	输出奇数通道的波形
TIOBn	输入	偶数通道的 ECK/TGIN/TIN*	输入偶数通道, 作为以下之一使用 <ul style="list-style-type: none"> • 外部时钟(ECK 信号) • 外部启动触发(TGIN 信号) • 测定波形(TIN 信号)
TIOBn+1	-	-	不使用

n: 偶数

*: 输入信号(ECK/TGIN/TIN 信号)的使用方法因复合定时器的定时器控制寄存器(TCR)的设定而异。

I/O 模式 4 (定时器启动/停止模式)的框图如图 24.3.2-5 所示。

图 24.3.2-5 I/O 模式 4 (定时器启动/停止模式)的框图



I/O 模式 4 的连接如表 24.3.2-15 所示。

表 24.3.2-15 I/O 模式 4 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出 作为 ECK/TGIN/TIN 和 DTRG 信号输入 通道 n+1 作为 COUT 信号输出到其他通道
自 TIOBn 引脚的输入信号	作为 ECK/TGIN/TIN 信号输入到 通道 n
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出

n : 偶数

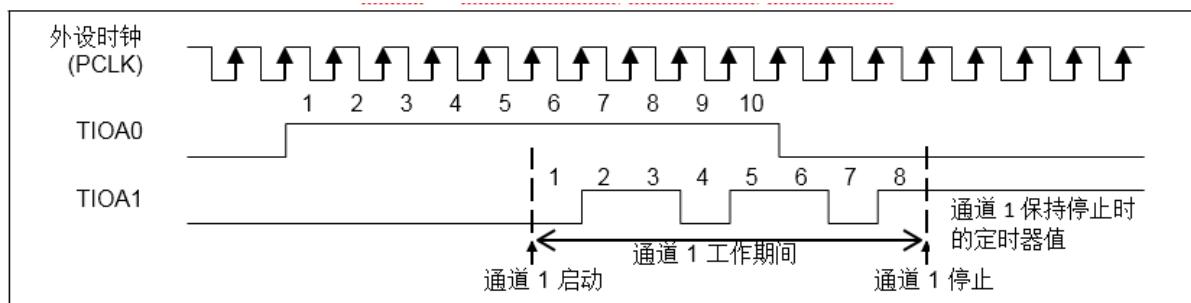
注意：

- 请使用复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0 位将奇数通道的触发输入沿设定为 上升沿(EDGS1, EDGS0=0b01)。
- 检测出 DTRG 信号的下降沿后，奇数通道停止工作。

图 24.3.2-6 显示的是选择 I/O 模式 4 (定时器启动/停止模式)且 通道 0 和 通道 1 用作 PWM 定时器时的操作例。

复合定时器 通道 0	设定值	复合定时器 通道 1	设定值
周期设定寄存器(CSR)	0x0010	周期设定寄存器(CSR)	0x0002
占空比设定寄存器(DUT)	0x0009	占空比设定寄存器(DUT)	0x0001
定时器控制寄存器(TCR)	0x0013	定时器控制寄存器(TCR)	0x0112

图 24.3.2-6 I/O 模式 4 (定时器启动/停止模式)的操作例



此模式下，可使用同时软件启动寄存器(CTSSSR)同时启动复数通道。

对应同时软件启动寄存器(CTSSSR)写"1"位的通道都同时启动。设定为此模式时使用的外部引脚如表 24.3.2-16 所示。

表 24.3.2-16 设定为 I/O 模式 5 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	不使用	
输出引脚	1 路	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-17 所示。

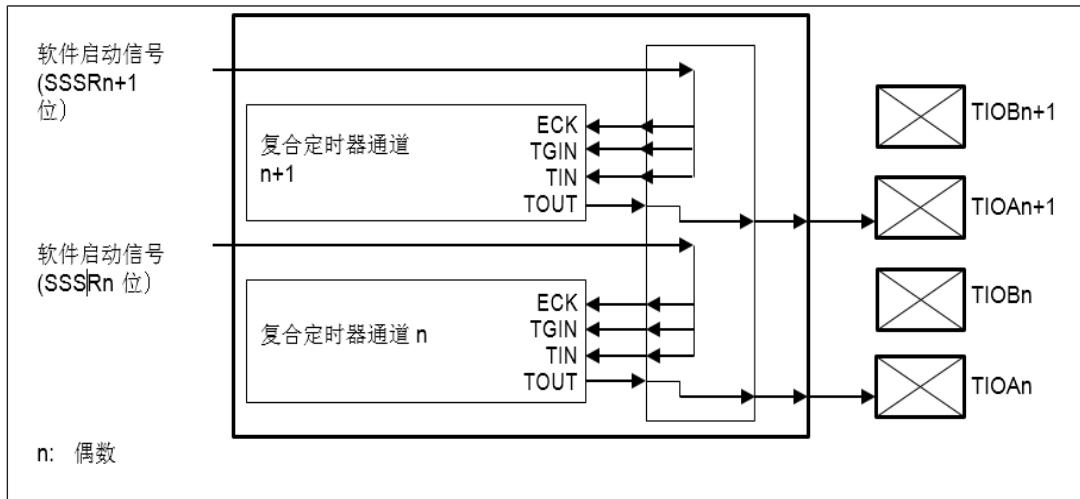
表 24.3.2-17 设定为 I/O 模式 5 时的外部引脚的连接对象和输入输出信号

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的 TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的 TOUT	输出奇数通道的波形
TIOBn TIOBn+1	-	-	不使用

n: 偶数

I/O 模式 5 (同时软件启动模式)的框图如图 24.3.2-7 所示。

图 24.3.2-7 I/O 模式 5 (同时软件启动模式)的框图



I/O 模式 5 的连接如表 24.3.2-18 所示。

表 24.3.2-18 I/O 模式 5 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出
软件启动信号 (在 CTSSSR 的 SSSR _n 位写"1")	作为 ECK/TGIN/TIN 信号输入 通道 n
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出
软件启动信号 (在 CTSSSR 的 SSSR _{n+1} 位写"1")	作为 ECK/TGIN/TIN 信号输入 通道 n+1

n: 偶数

CTSSSR: 同时软件启动寄存器

对同步软件启动寄存器(CTSSSR)写"1", 上升沿(ECK/TGIN/TIN 信号) 输入到写入位的对应通道。

注意:

- 请使用复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0 位将触发输入沿设定为上升沿(EDGS1, EDGS0=0b01)。

I/O 模式 6 (软件启动定时器启动/停止模式)

此模式使用偶数通道控制奇数通道的启动/停止。

对同时软件启动寄存器(CTSSSR)写"1"启动偶数通道。

在偶数通道的输出波形(TOUT 信号)上检测出上升沿时, 奇数通道启动;检测出下降沿时, 奇数通道停止。

设定为此模式时使用的外部引脚如表 24.3.2-19 所示。

表 24.3.2-19 设定为 I/O 模式 6 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	不使用	
输出引脚	1 路	1 路

使用外部引脚的连接对象和输入输出信号如表 24.3.2-20 所示。

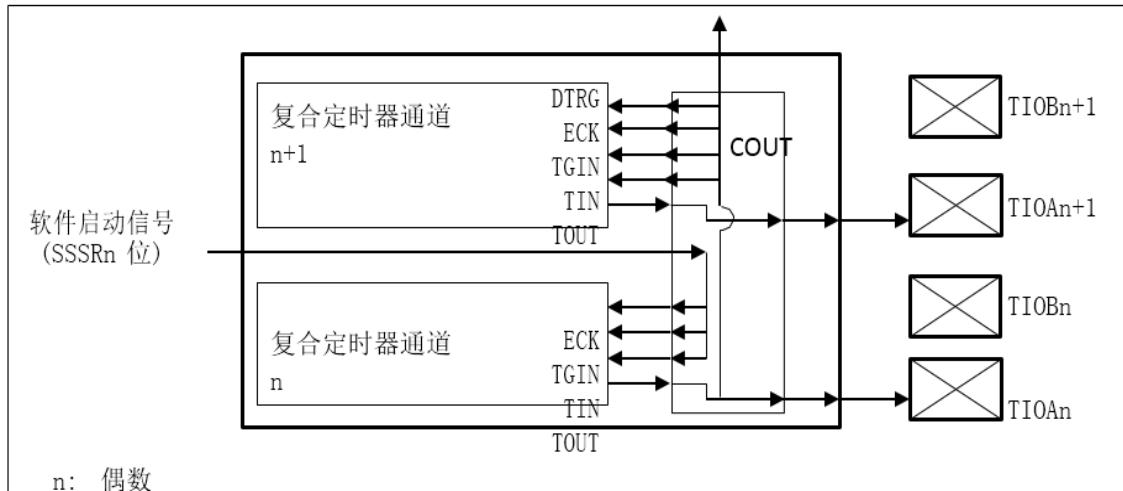
表 24.3.2-20 设定为 I/O 模式 6 时的外部引脚的连接对象和输入输出信号

外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的 TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的 TOUT	输出奇数通道的波形
TIOBn TIOBn+1	-	-	不使用

n：偶数

I/O 模式 6 (软件启动定时器启动/停止模式)的框图如图 24.3.23-8 所示。

图 24.3.2-8 I/O 模式 6 (软件启动定时器启动/停止模式)的框图



I/O 模式 6 的连接如表 24.3.2-21 所示。

表 24.3.2-21 I/O 模式 6 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出 作为 ECK/TGIN/TIN/DTRG 信号输入到通道 n+1 作为 COUT 信号输出到其他通道

软件启动信号 (对 CTSSSR 的 SSSRn 位写"1")	作为 ECK/TGIN/TIN 信号输入到 通道 n
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出

n: 偶数

CTSSSR: 同时软件启动寄存器

在需要同时软件启动寄存器(CTSSSR)启动的偶数通道的对应位写"1", 上升沿输入(ECK/TGIN/TIN 信号)到对应通道。

通道 n 的启动/停止时序与 I/O 模式 4 相同。

注意:

- 请在复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0 位将触发输入沿设定为上升沿(EDGS1, EDGS0=0b01)。
- 检测出 DTRG 信号的下降沿时, 奇数通道停止工作。

I/O 模式 7 (定时器启动模式)

在此模式下, 偶数通道的输出波形(TOUT 信号)作为奇数通道的输入信号(ECK/TGIN/TIN 信号)使用设定为此模式时使用的外部引脚如表 24.3.2-22 所示。

表 24.3.2-22 设定为 I/O 模式 7 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	1 路	不使用
输出引脚	1 路	1 路

使用的外部引脚的连接对象和输入输出信号如表 24.3.2-23 所示。

表 24.3.2-23 设定为 I/O 模式 7 时的外部引脚的连接对象和输入输出信号

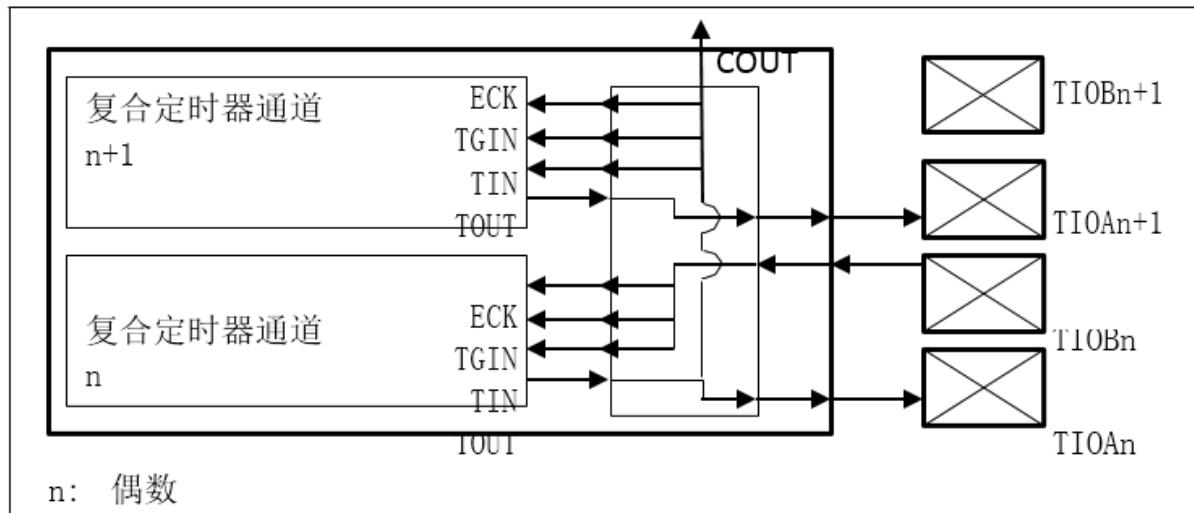
外部引脚	I/O	连接对象(内部信号)	输入输出信号
TIOAn	输出	偶数通道的 TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的 TOUT	输出奇数通道的波形
TIOBn	输入	偶数通道的 ECK/ TGIN/TIN*	输入偶数通道, 作为以下之一使用 <ul style="list-style-type: none"> • 外部时钟(ECK 信号) • 外部启动触发(TGIN 信号) • 测定波形(TIN 信号)
TIOBn+1	-	-	不使用

n: 偶数

*: 输入波形(ECK/TGIN/TIN 信号)的使用方法因定时器控制寄存器(TCR)的设定而异。

I/O 模式 7(定时器启动模式)时的框图如图 24.3.2-9 所示。

图 24.3.2-9 I/O 模式 7(定时器启动模式)的框图



I/O 模式 7 的连接如表 24.3.2-24 所示。

表 24.3.2-24 I/O 模式 7 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出 作为 ECK/TGIN/TIN 信号输入到 通道 n+1 作为 COUT 信号输出到其他通道
自 TIOBn 引脚的输入信号	作为 ECK/TGIN/TIN 信号输入到 通道 n
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出

n: 偶数

通道 n 的启动时序与 I/O 模式 4 相同

I/O 模式 8(其他通道触发共享定时器启动/停止模式)

在此模式下，2 路通道低位端通道的 COUT 信号作为 CIN 信号输入，作为外部启动触发(TGIN 信号)使用。

设定为此模式时使用的外部引脚如表 24.3.2-25 所示。

表 24.3.2-25 设定为 I/O 模式 8 时使用的外部引脚

	偶数通道	奇数通道
输入引脚	不使用	
输出引脚	1 路	1 路

使用的外部引脚的连接对象和 I/O 信号如表 24.3.2-26 所示。

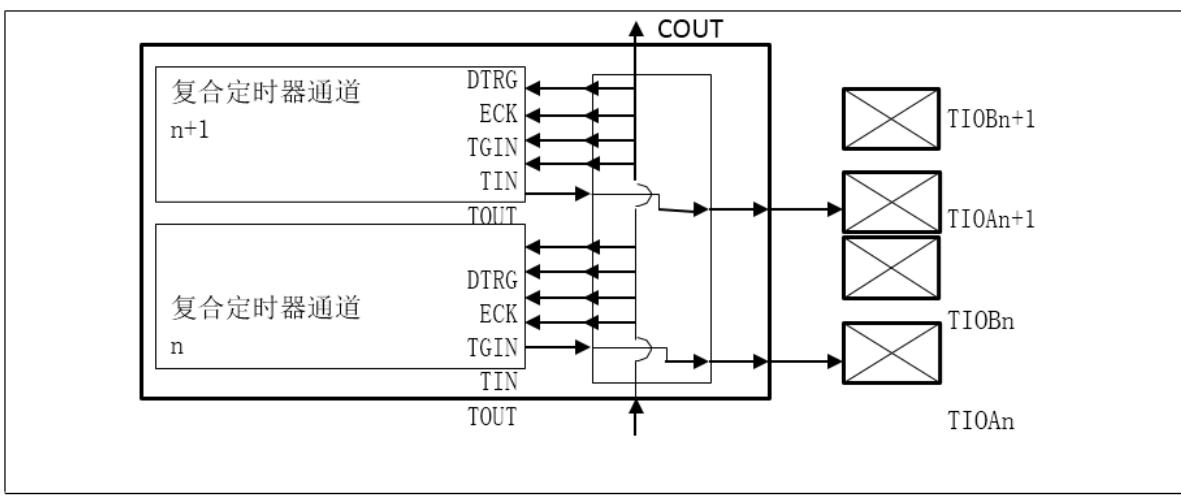
表 24.3.2-26 设定为 I/O 模式 8 时的外部引脚的连接对象和 I/O 信号

外部引脚	I/O	连接对象(内部信号)	I/O 信号
TIOAn	输出	偶数通道的 TOUT	输出偶数通道的波形
TIOAn+1	输出	奇数通道的 TOUT	输出奇数通道的波形
TIOBn TIOBn+1	-	-	不使用

n: 偶数

I/O 模式 8 (其他通道触发共享定时器启动/停止模式)时的框图如图 24.3.2-10 所示。

图 24.3.2-10 I/O 模式 8(其他通道触发共享定时器启动/停止模式)的框图



n: 偶数

I/O 模式 8 的连接如表 24.3.2-27 所示。

表 24.3.2-27 I/O 模式 8 的连接

连接源	连接对象
通道 n 的 TOUT 信号	从 TIOAn 引脚输出
通道 n+1 的 TOUT 信号	从 TIOAn+1 引脚输出
CIN 信号*	作为 ECK/TGIN/TIN 信号和 DTRG 信号输入到 通道 n 和 通道 n+1 作为 COUT 信号输出到其他通道

n: 偶数

*: 输入自其他通道的 COUT 信号作为 CIN 信号

可输入到 通道 n/n+1 作为 ECK, TGIN, TIN 的 通道 n-2/n-1 信号如下:

I/O 模式 2 时使用外设时钟将 TIOBn-2 输入同步后的信号

I/O 模式 3 时从 通道 n-4/n-3 输入的触发信号

I/O 模式 4 时的 TIOAn-2 输出

I/O 模式 6 时的 TIOAn-2 输出

I/O 模式 7 时的 TIOAn-2 输出

I/O 模式 8 时从 通道 n-4/n-3 输入的触发信号

注意：

- 设定为此模式的通道将 2 通道低位端(n-2, n-1)COUT 信号作为 CIN 信号输入并使用
(例：若将 通道 2, 通道 3 设定为此模式，可使用 通道 0, 通道 1 的 COUT 信号)。通道
0 和 通道 1 不可设定为此模式。
- 通道设定为此模式时，请使用复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0
位将触发输入沿设定为上升沿(EDGS1, EDGS0=0b01)。但使用复合定时器的定时器控制
寄存器(TCR)的 FUNC2 ~ FUNC0 位将定时器功能设定为 16/32 位 PWC 定时器(FUNC2
~ FUNC0=0b100)除外。
- 若检测到 DTRG 信号的下降沿，复合定时器停止工作。

24.4 寄存器

以下显示的是复合定时器 I/O 选择功能的寄存器一览。

复合定时器 I/O 选择功能的寄存器

表 24.4-1 复合定时器 I/O 选择功能的寄存器一览

寄存器简称	寄存器名称	参照章节
CTSEL0123	I/O 选择寄存器	24.4.1
CTSEL4567	I/O 选择寄存器	24.4.2
CTSSSR	同时软件启动寄存器	24.4.3

24.4.1 I/O 选择寄存器(CTSEL0123)

本寄存器设定复合定时器的 通道 0 ~ 通道 3 的 I/O 模式。

寄存器配置

bit	15	14	13	12	11	10	9	8
Field	SEL23	SEL23_2	SEL23_1	SEL23_0	SEL01_3	SEL01_2	SEL01_1	SEL01_0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit15:12] SEL23_3 ~ SEL23_0: 通道 2/通道 3 I/O 选择位

Bit15	Bit14	Bit13	Bit12	I/O 选择位
0	0	0	0	I/O 模式 0(16 位定时器标准模式)
0	0	0	1	I/O 模式 1(定时器全模式)
0	0	1	0	I/O 模式 2(外部触发共享模式)
0	0	1	1	I/O 模式 3(其他通道触发共享模式)
0	1	0	0	I/O 模式 4(定时器启动/停止模式)
0	1	0	1	I/O 模式 5(同时软件启动模式)
0	1	1	0	I/O 模式 6(软件启动定时器启动/停止模式)
0	1	1	1	I/O 模式 7(定时器启动模式)
1	0	0	0	I/O 模式 8(其他通道触发共享定时器启动/停止模式)
其他				禁止设定

[bit11:8] SEL01_3 ~ SEL01_0: 通道 0/通道 1 I/O 选择位

Bit11	Bit10	Bit9	Bit8	I/O 选择位
0	0	0	0	I/O 模式 0(16 位定时器标准模式)
0	0	0	1	I/O 模式 1(定时器全模式)
0	0	1	0	I/O 模式 2(外部触发共享模式)
0	0	1	1	I/O 模式 3(其他通道触发共享模式)
0	1	0	0	I/O 模式 4(定时器启动/停止模式)
0	1	0	1	I/O 模式 5(同时软件启动模式)
0	1	1	0	I/O 模式 6(软件启动定时器启动/停止模式)
0	1	1	1	I/O 模式 7(定时器启动模式)
1	0	0	0	I/O 模式 8(其他通道触发共享定时器启动/停止模式)
其他				禁止设定

注意：

- 通道 0 和 通道 1 是复合定时器的最低位通道，不可使用需要设置更低位端通道的模式。
因此禁止设定以下模式。
 - I/O 模式 3(其他通道触发共享模式)
 - I/O 模式 8(其他通道触发共享定时器启动/停止模式)
- 使用复合定时器的定时器控制寄存器(TCR)的 FUNC2 ~ FUNC0 位将复合定时器设定为复位模式(FUNC2 ~ FUNC0=0b000)后，再重写本寄存器。

24.4.2 I/O 选择寄存器(CTSEL4567)

本寄存器设定复合定时器的 通道 4 ~ 通道 7 的 I/O 模式

寄存器配置

bit	15	14	13	12	11	10	9	8
Field	SEL67_3	SEL67_2	SEL67_1	SEL67_0	SEL45_3	SEL45_2	SEL45_1	SEL45_0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit15:12] SEL67_3 ~ SEL67_0: 通道 6/通道 7 I/O 选择位

Bit15	Bit14	Bit13	Bit12	I/O 选择位
0	0	0	0	I/O 模式 0(16 位定时器标准模式)
0	0	0	1	I/O 模式 1(定时器 全模式)

0	0	1	0	I/O 模式 2 (外部触发共享模式)
0	0	1	1	I/O 模式 3 (其他通道触发共享模式)
0	1	0	0	I/O 模式 4 (定时器启动/停止模式)
0	1	0	1	I/O 模式 5 (同时软件启动模式)
0	1	1	0	I/O 模式 6 (软件启动定时器启动/停止模式)
0	1	1	1	I/O 模式 7 (定时器启动模式)
1	0	0	0	I/O 模式 8 (其他通道触发共享定时器启动/停止模式)
其他				禁止设定

[bit11:8] SEL45_3 ~ SEL45_0: 通道 4/通道 5 I/O 选择位

Bit11	Bit10	Bit9	Bit8	I/O 选择位
0	0	0	0	I/O 模式 0(16 位定时器标准模式)
0	0	0	1	I/O 模式 1(定时器 全模式)
0	0	1	0	I/O 模式 2 (外部触发共享模式)
0	0	1	1	I/O 模式 3 (其他通道触发共享模式)
0	1	0	0	I/O 模式 4 (定时器启动/停止模式)
0	1	0	1	I/O 模式 5 (同时软件启动模式)
0	1	1	0	I/O 模式 6 (软件启动定时器启动/停止模式)
0	1	1	1	I/O 模式 7 (定时器启动模式)
1	0	0	0	I/O 模式 8 (其他通道触发共享定时器启动/停止模式)
其他				禁止设定

注意：

- 使用复合定时器的定时器控制寄存器(TCR)的 FUNC2 ~ FUNC0 位将复合定时器设定为复位模式(FUNC2 ~ FUNC0=0b000)后，再重写本寄存器。

24.4.3 同时软件启动寄存器(CTSSSR)

本寄存器使用软件同时启动复合定时器。

最多可同时启动 8 路对应写"1"位的通道。

寄存器配置

bit	7	6	5	4	3	2	1	0
Field	SSSR7 to SSSR0							
属性	W							
初始值	0							

寄存器功能

[bit7:0] SSSR7 ~ SSSR0: 同步软件启动位

SSSRx	同步软件启动位
0	写"0"无效
1	启动复合定时器的 通道 x

x: 7 ~ 0

注意:

- 当设定下述之外的模式时, 切勿对本寄存器写值。
- I/O 模式 5 (同时软件启动模式)
- I/O 模式 6 (软件启动定时器启动/停止模式)(仅限偶数通道)
- 对于使用本寄存器启动的通道, 请使用复合定时器的定时器控制寄存器(TCR)的 EDGS1, EDGS0 位将触发输入沿设定为上升沿(EDGS1, EDGS0=0b01)。

25. 复合定时器

本章介绍了复合定时器的功能和操作。

25.1 概要

通过设定定时器控制寄存器的 FUNC2, 1, 0 位, 复合定时器可从 16 位 PWM 定时器, 16 位 PPG 定时器, 16/32 位重载定时器, 16/32 位 PWC 定时器中选择一个定时器功能。本节是可选择的各定时器的概要。

模式设定和各种定时器功能的关系:

FUNC2, FUNC1, FUNC0 位的设定	功能
0b000	复位模式
0b001	16 位 PWM 定时器
0b010	16 位 PPG 定时器
0b011	16/32 位重载定时器
0b100	16/32 位 PWC 定时器

- 复位模式

复位模式是指复合定时器的初始复位后的状态(各寄存器都是初始值)。在设置切换到不同的定时器功能 T32 位前, 确认设置为复位模式。然而, 如果是初始复位后, 则无需在设置定时器功能或 T32 位前设置为复位模式。

- 16 位 PWM 定时器

该定时器由 16 位递减计数器、带周期设置缓冲器的 16 位数据寄存器、带占空比设置缓冲器的 16 位比较寄存器及引脚控制模块构成。

由于周期和占空比数据保存到带缓冲器的寄存器内, 因此在定时器工作期间可改写数据。

16 位递减计数器的计数时钟可从 8 种内部时钟(机器时钟的 1/4/16/128/256/512/1024/2048 分频)和 3 种外部事件(上升沿、下降沿、双沿检测)中选择。可选择因下溢而停止计数的单次模式和重载后重复计数的连续模式。

关于 16 位 PWM 定时器的启动, 可从软件触发和 3 种外部事件(上升沿、下降沿、双沿检测)中选择。

- 16 位 PPG 定时器

该定时器由 16 位递减计数器、“H”宽度设置的 16 位数据寄存器、“L”宽度设置的 16 位

数据寄存器及引脚控制模块构成。

16 位递减计数器的计数时钟可从 8 种内部时钟(机器时钟的 1/4/16/128/256/512/1024/2048 分频)和 3 种 外部事件(上升沿、下降沿、双沿检测)中选择。

可选择因下溢而停止计数的单次模式和重载后重复计数的连续模式。

关于 16 位 PPG 定时器的启动，可从软件触发和 3 种外部事件(上升沿、下降沿、双沿检测)中选择。

- 16/32 位重载定时器

该定时器由 16 位递减计数器、16 位重载寄存器及引脚控制模块构成。

16 位递减计数器的计数时钟可从 8 种内部时钟(机器时钟的 1/4/16/128/256/512/1024/2048 分频)和 3 种 外部事件(上升沿、下降沿、双沿检测)中选择。

可选择因下溢而停止计数的单次模式和重载后重复计数的连续模式。

关于 16/32 位重载定时器的启动，可从软件触发和 3 种外部事件(上升沿、下降沿、双沿检测)中选择。

具有门控功能，此门控功能仅在外部输入有效电平输入时才执行递减计数。有效电平可从（高电平或低电平）两种电平中选择。

- 16/32 位 PWC 定时器

该定时器由 16 位递增计数器、测量输入引脚及控制寄存器构成。

通过外部脉冲输入，测定任意事件之间的时间间隔。

参考计数时钟可从 8 种内部时钟(1/4/16/128/256/512/1024/2048 分频)中选择。

测量模式：

H 脉宽($\uparrow \sim \downarrow$) / L 脉宽($\downarrow \sim \uparrow$)

上升周期($\uparrow \sim \uparrow$) / 下降周期($\downarrow \sim \downarrow$)

沿间测量(\uparrow 或 $\downarrow \sim \downarrow$ 或 \uparrow)

测量完成时可产生中断请求。

可选择单次测量或连续测量。

25.2 框图

图 25.2-1~25.2-4 是各模式下复合定时器的框图。

图 25.2-1 16 位 PWM 定时器的框图

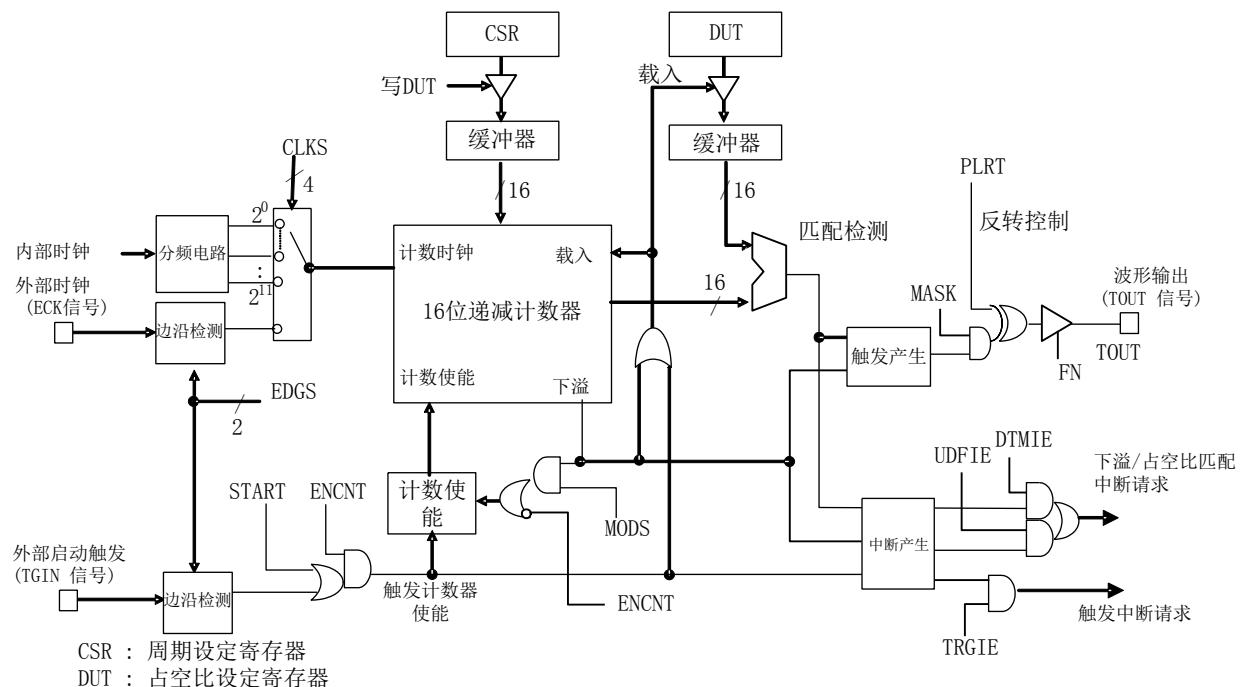


图 25.2-2 16 位 PPG 定时器的框图

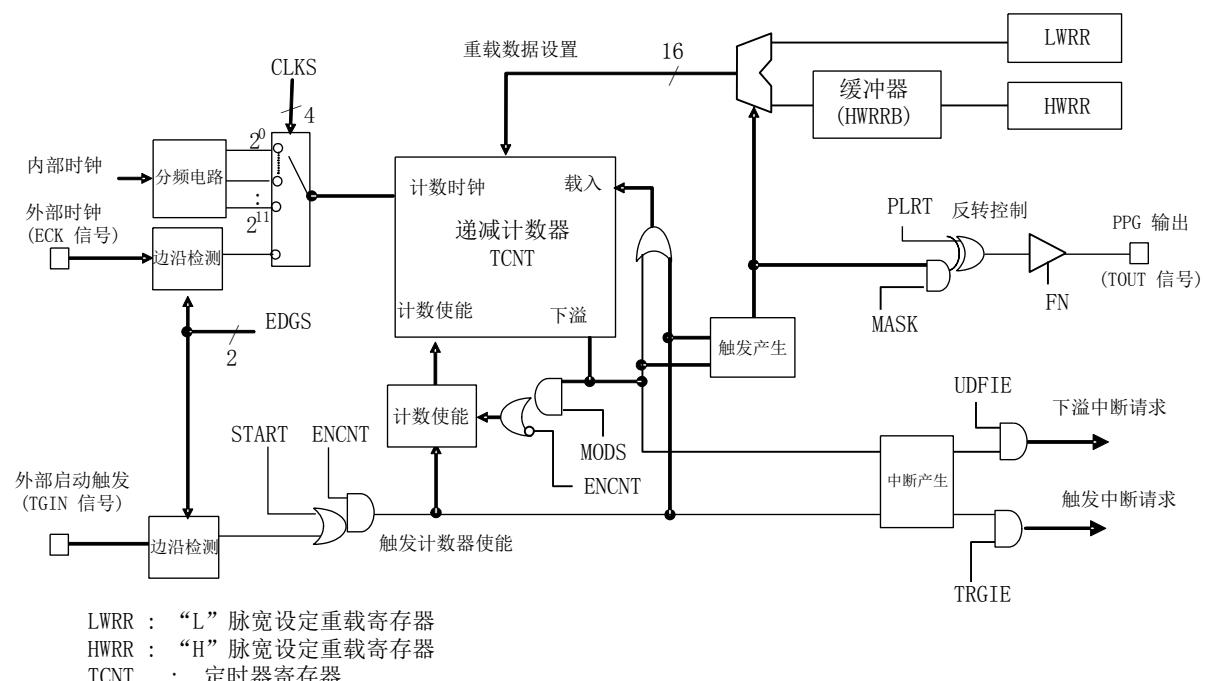
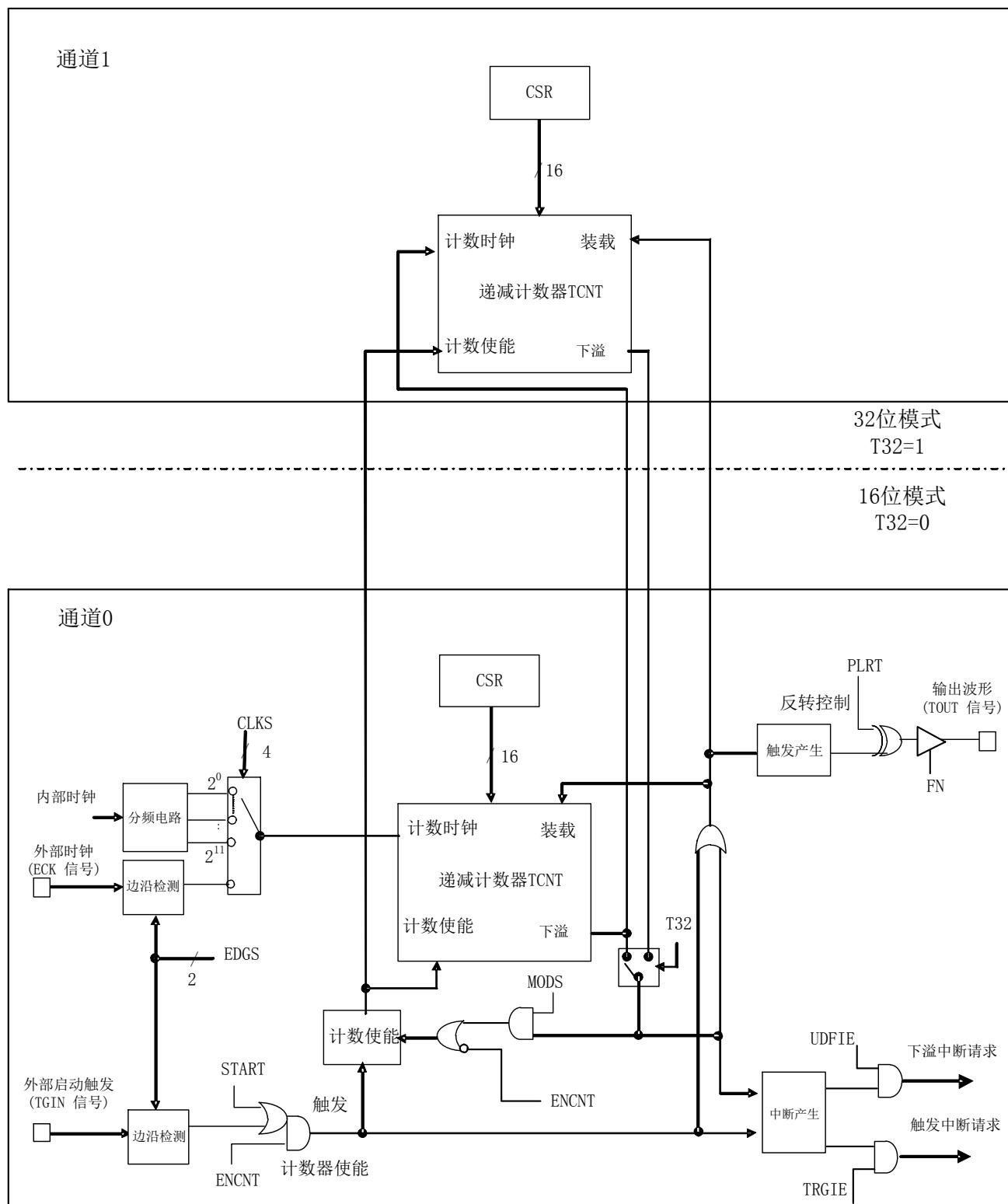
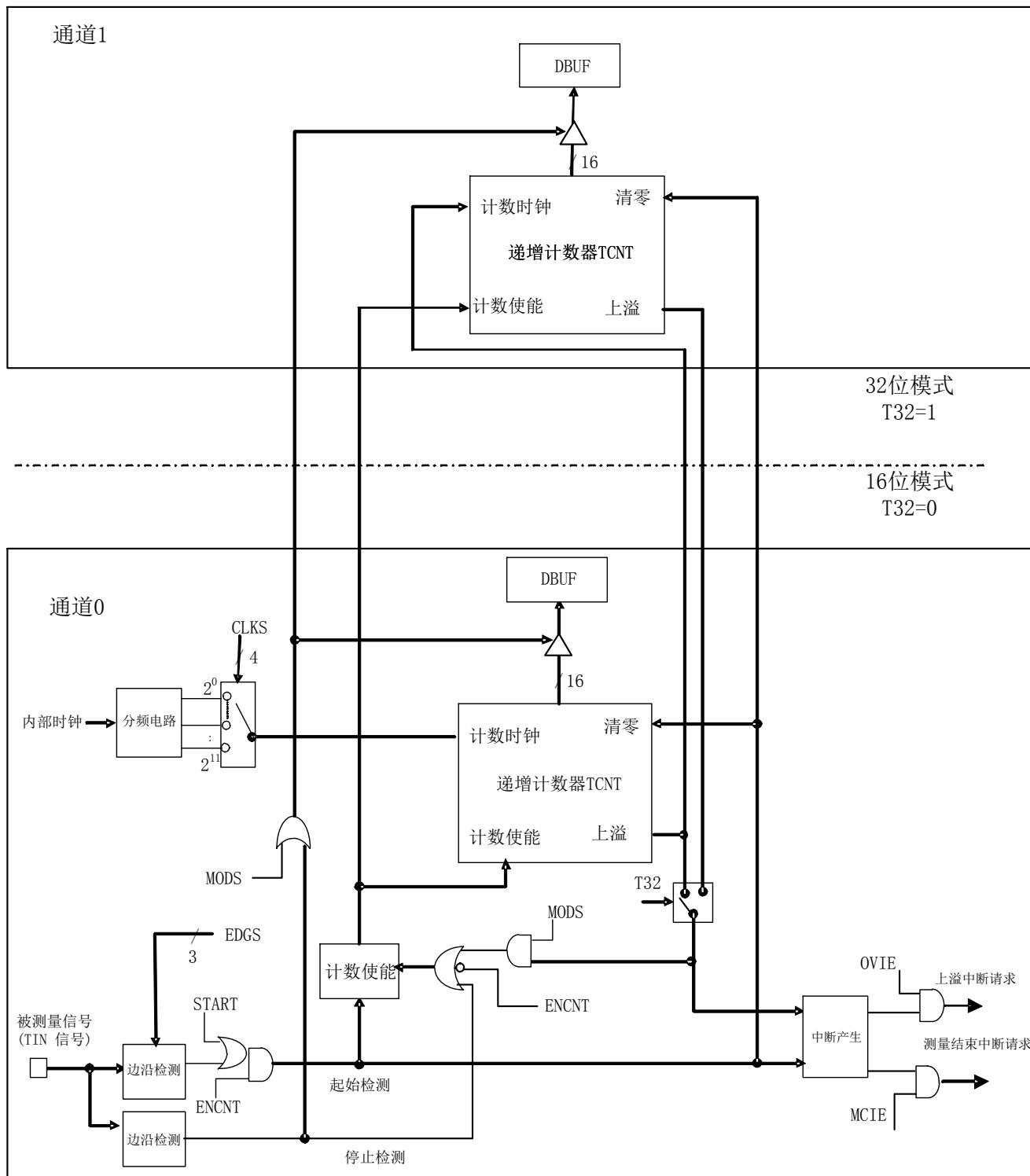


图 25.2-3 16/32 位重载定时器(通道 1, 0)的框图



CSR : 复合定时器 1 周期设置寄存器
 TCNT : 复合定时器 1 定时器寄存器
 CSR : 复合定时器 0 周期设置寄存器
 TCNT : 复合定时器 0 定时器寄存器

图 25.2-4 16/32 位 PWC 定时器(通道 1, 0)的框图



DBUF : 复合定时器 1 数据缓存寄存器

TCNT : 复合定时器 1 定时器寄存器

DBUF : 复合定时器 0 数据缓存寄存器

TCNT : 复合定时器 0 T定时器寄存器

25.3 操作

本节介绍复合定时器的操作。

复合定时器的操作

- 复位模式

复位模式是指复合定时器的初始复位后的状态(各寄存器都是初始值)。在设置切换到不同的定时器功能或 T32 位前，确认先设置为复位模式。然而，如果是初始复位后，则无需在设置定时器功能或 T32 位前设置为复位模式。在 32 位模式下，将该模式设为偶数通道时，奇数通道也同时发生复位，因此，无需对奇数通道设定复位模式。

- 16 位 PWM 定时器

16 位 PWM 定时器从触发启动时周期设定的值开始递减计数。首先，输出“L”电平。16 位递减计数器的值与占空比设定寄存器的值匹配时，则输出“H”电平。然后，计数器发生下溢时，输出重新取反，再输出“L”电平。这样以来，周期和占空比可产生任意波形。

- 16 位 PPG 定时器

16 位 PPG 定时器从触发启动时“L”脉宽设定重载寄存器的值开始递减计数。首先，输出“L”电平。发生下溢时，输出“H”电平。定时器继续从“H”脉宽设定重载寄存器的值开始递减计数，再发生下溢时，输出“L”电平。这样以来，可产生任意“L”脉宽和“H”脉宽的波形。

- 16 位重载定时器

16 位重载定时器从触发启动时周期设定的值开始递减计数。16 位递减计数器发生下溢时，设定中断标志。输出电平根据 MODS 位的设定在每次下溢时取反跳变，计数开始时，输出“H”，发生下溢时，输出“L”电平。

- 32 位重载定时器

基本操作与 16 位重载定时器相同，但该定时器使用一个偶数通道和一个奇数通道作为 32 位重载定时器进行操作。这时，偶数通道用作低位 16 位定时器，奇数通道用作高位 16 位定时器。中断控制和输出波形控制仅取决于偶数通道的设定。设定周期时，先写高位寄存器(奇数通道)，再写低位寄存器(偶数通道)。读取定时器值时，先读低位寄存器(偶数通道)，再读高位寄存器(奇数通道)。

- 16 位 PWC 定时器

PWC 定时器通过设定输入信号的测量起始沿启动 16 位递增计数器，和测量结束沿停止计数器。这期间的计数值以脉宽形式保存到数据缓冲寄存器。

- 32 位 PWC 定时器

基本操作与 16 位 PWC 定时器相同，但该定时器使用一个偶数通道和一个奇数通道作为 32 位 PWC 定时器。这时，偶数通道作为低位 16 位计数器，奇数通道作为高位 16 位计数器。中断控制仅取决于偶数通道的设定。读取测定值或计数值时，先读低位寄存器(偶数通道)，再读高位寄存器(奇数通道)。

25.4 32 位模式的操作

重载定时器和 PWC 使用 2 路通道时，其工作在 32 位模式。本节介绍 32 位模式的基本功能和操作。

- 32 位模式的功能

该功能由复合定时器的 2 路通道实现 32 位数据的重载定时器或 32 位数据的 PWC 定时器操作。在偶数通道读取低 16 位定时器/计数器的值时，同时在奇数通道读取高 16 位定时器/计数器的值，也可在运行期间读取定时器/计数器的值。

- 32 位模式的设定

首先，将偶数通道的 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位设为"0b000"实现复位。然后，同 16 位模式时的操作，选择重载定时器或 PWC 定时器并设置运行。这时，向 TCR 寄存器的 T32 位写"1"设定到 32 位工作模式。奇数通道的 T32 位保持"0"。无需设定复位模式。并且，重载定时器时，在奇数通道的 PWM 周期设置寄存器中，设置 32 位中的高 16 位的重载值。然后，在偶数通道的 PWM 周期设置寄存器中，设置低 16 位的重载值。

在写 T32 位后，立刻切换到 32 位工作模式，因此，2 路通道应在计数停止状态下改变设置。将偶数通道的 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位设为"0b000"进入复位模式后，可将 32 位模式切换为 16 位模式。这样以来，偶数和奇数 2 路通道的状态被复位，各个通道可在 16 位模式下进行设定。

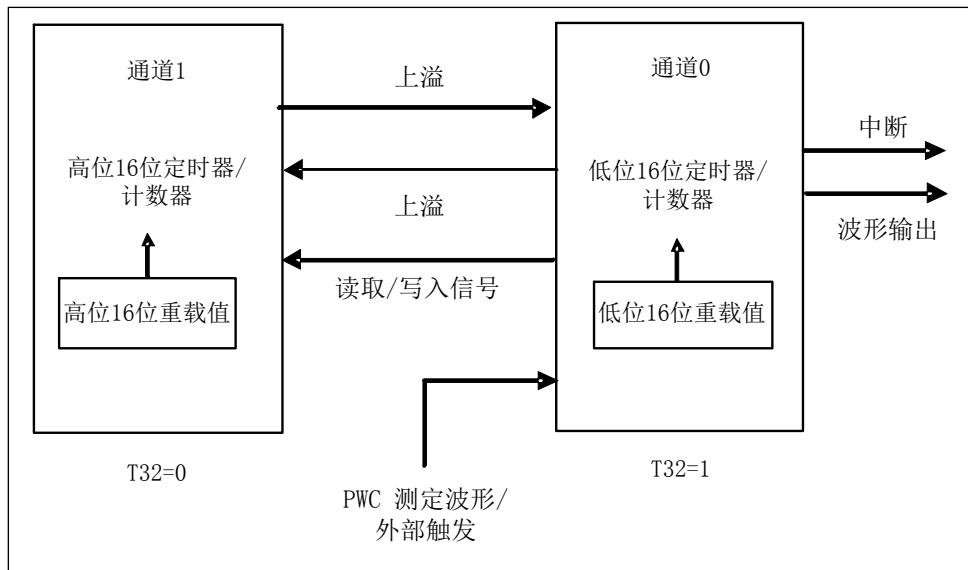
- 32 位模式的操作

设为 32 位模式后，通过控制偶数通道启动重载定时器或 PWC 定时器时，偶数通道的定时器/计数器用作低 16 位工作，奇数通道的定时器/计数器用作高 16 位工作。

32 位模式下的操作取决于偶数通道的设定。因此，忽略奇数通道的设定(重载定时器时的周期设定寄存器除外)。定时器启动、波形输出、中断信号均是偶数通道设置和应用(奇数通道屏蔽且固定为"L")。

图 25.4-1 是通道 0、通道 1 时的结构。

图 25.4-1 32 位模式操作的结构(通道 0, 通道 1)



25.5 复合定时器中断

本节介绍了复合定时器各功能下的中断请求标志、中断使能位及中断源，各功能的中断控制位和中断源。

表 25.5-1 是各功能下的中断控制位和中断源。

表 25.5-1 各模式下的中断控制位和中断源

		状态控制寄存器(STA)			
		中断请求标志位	中断请求使能位	中断源	中断源输出信号
PWM 定时器功能	UDFIR:bit0	UDFIE:bit4	检出下溢		IRQ0
	DTMIR:bit1	DTMIE:bit5	检出占空比匹配		
	TRGIR:bit2	TGIE:bit6	检出定时器启动触发		IRQ1
PPG 定时器功能	UDFIR:bit0	UDFIE:bit4	检出下溢		IRQ0
	TRGIR:bit2	TGIE:bit6	检出定时器启动触发		IRQ1
重载定时器功能	UDFIR:bit0	UDFIE:bit4	检出下溢		IRQ0
	TRGIR:bit2	TGIE:bit6	检出定时器启动触发		IRQ1
PWC 定时器功能	OVIR:bit0	OVIE:bit4	检出上溢		IRQ0
	MCIR:bit2	MCIE:bit6	检出测定完成		IRQ1

25.6 寄存器

本节介绍了复合定时器在各模式下的寄存器列表。

16 位 PWM 定时器选择时的寄存器一览表

表 25.6-1 16 位 PWM 定时器选择时的寄存器一览表

寄存器简称	寄存器名称	参考章节
TCR	定时器控制寄存器	25.8.1.6
TCR2	定时器控制寄存器2	25.8.1.6
STA	状态控制寄存器	25.8.1.6
CSR	PWM 周期设定寄存器	25.8.1.7
DUT	PWM 占空比设定寄存器	25.8.1.8
TCNT	定时器寄存器	25.8.1.9

16 位 PPG 定时器选择时的寄存器一览表

表 25.6-2 16 位 PPG 定时器选择时的寄存器一览表

寄存器简称	寄存器名称	参考章节
TCR	定时器控制寄存器	25.8.2.6
TCR2	定时器控制寄存器2	25.8.2.6
STA	状态控制寄存器	25.8.2.6
LWRR	"L"脉宽设定重载寄存器	25.8.2.7
HWRR	"H"脉宽设定重载寄存器	25.8.2.8
TCNT	定时器寄存器	25.8.2.9

重载定时器选择时的寄存器一览表

表 25.6-3 重载定时器选择时的寄存器一览表

寄存器简称	寄存器名称	参考章节
TCR	定时器控制寄存器	25.8.3.3
TCR2	定时器控制寄存器2	25.8.3.3
STA	状态控制寄存器	25.8.3.3
CSR	周期设定寄存器	25.8.3.4
TCNT	定时器寄存器	25.8.3.5

PWC 定时器选择时的寄存器一览表

表 25.6-4 PWC 定时器选择时的寄存器一览表

寄存器简称	寄存器名称	参考章节
TCR	定时器控制寄存器	25.8.4.2
TCR2	定时器控制寄存器2	25.8.4.2
STA	状态控制寄存器	25.8.4.2
DBUF	数据缓冲寄存器	25.8.4.3

25.7 注意事项

本节介绍复合定时器的使用注意事项。

使用各定时器时设定的注意事项

- 操作期间，禁止改写 TCR2 寄存器，TCR 寄存器中的以下位。这些位应在启动前或停止后进行改写

[TCR2 bit 8], [TCR bit 14, 13, 12]	CLKS3 ~ CLKS0: 时钟选择位
[bit 10, 9, 8]	EDGS2, EDGS1, EDGS0: 测量边沿选择位
[bit 7]	T32: 32 位定时器选择位 (重载定时器 PWC 功能被选择时)
[bit 6, 5, 4]	FUNC2 ~ FUNC0: 定时器功能选择位
[bit 2]	MODS: 测量模式(单次/连续)选择位

- 将 TCR 寄存器的 FUNC2 ~ FUNC0 位设为"0b000"实现复位模式时，复合定时器的全部寄存器恢复初值。因此，需要重新设定所有寄存器。
- 将 TCR 寄存器的 FUNC2 ~ FUNC0 位设为"0b000"进入复位模式时，忽略对 TCR 寄存器的 FUNC2~ FUNC0 位除外的设置且将其初始化。

16 位 PWM/PPG/重载定时器的使用注意事项

- 中断请求标志设定时序和清零时序重合时，标志设定优先，清零操作无效。
- 递减计数器的载入时序和计数时序重合时，优先执行载入操作。
- 通过 TCR 寄存器的 FUNC2, FUNC1 和 FUNC0 位设定定时器功能后，再设定周期、占空比、"H"脉宽、"L"脉宽。
- 单次模式下，计数结束后检测出重新启动时，重载计数值后重新启动。

PWC 定时器的使用注意事项

- 如果将计数启动使能位(ENCNT)设为"1"，计数器被清零。因此，使能启动前，计数器中的数据无效
- 从系统复位 / 复位模式中同时设定 PWC 模式(FUNC="0b100")和测定起始(ENCNT=1)时，有时操作结果取决于之前测定信号的状态。
- 连续测定模式下设定重启时，若同时检测出测定起始沿，则立刻从"0x0001"开始计数。

- 开始计数操作后重启时，根据时序有时发生以下操作：
 - 在脉宽单次测量模式中，如果遇到测量完成边沿，则定时器会重启并且等待下一个测量起始沿，然而测量完成标志(MCIR)会被置位。
 - 在脉宽连续测量模式中，如果遇到测量完成边沿，则定时器会重启并且等待下一个测量起始沿，然而测量完成标志(MCIR)会被置位，并且此时的测定结果传输到 DBUF。

如上所述，在定时器工作期间需要重启定时器时，请留意上述的标志位操作和中断控制操作。

25.8 功能描述

本节介绍复合定时器各功能。

25.8.1 PWM 定时器功能

通过设定复合定时器的定时器控制寄存器 FUNC2, 1, 0 位后，可从 16 位 PWM 定时器，16 位 PPG 定时器，16/32 位重载定时器，16/32 位 PWC 定时器中选择一种定时器功能。本节介绍设定了 PWM 时的定时器功能。

1. 16 位 PWM 定时器
2. 单次操作
3. 中断及时序图
4. 输出波形
5. PWM 定时器操作流程图
6. 选择 PWM 定时器时的定时器控制寄存器(TCR, TCR2)， 状态控制寄存器(STA)
7. PWM 周期设定寄存器(CSR)
8. PWM 占空比设定寄存器(DUT)
9. 定时器寄存器(TCNT)

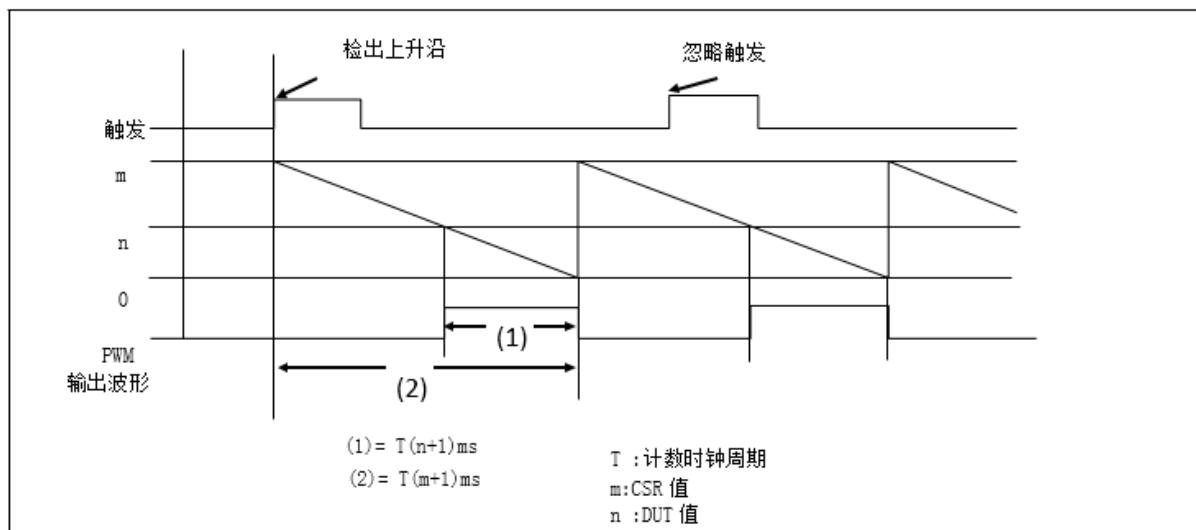
25.8.1.1 16 位 PWM 定时器操作

PWM 操作中，检测出触发后，可单次或连续输出设定周期的波形。通过改变 CSR 值可以控制输出脉冲的周期。此外，改变 DUT 值可控制占空比。向 CSR 写数据后，必须向 DUT 写值。

连续操作

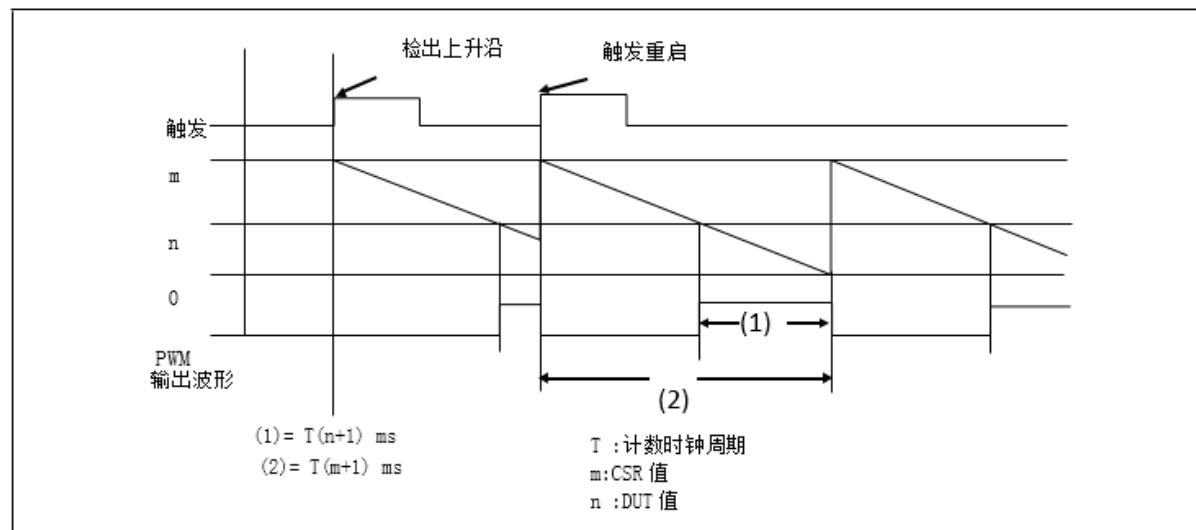
禁止重启(RSTEN="0")

图 25.8.1-1 PWM 操作时序图(禁止重启)



使能重启(RSTEN="1")

图 25.8.1-2 PWM 操作时序图(使能重启)



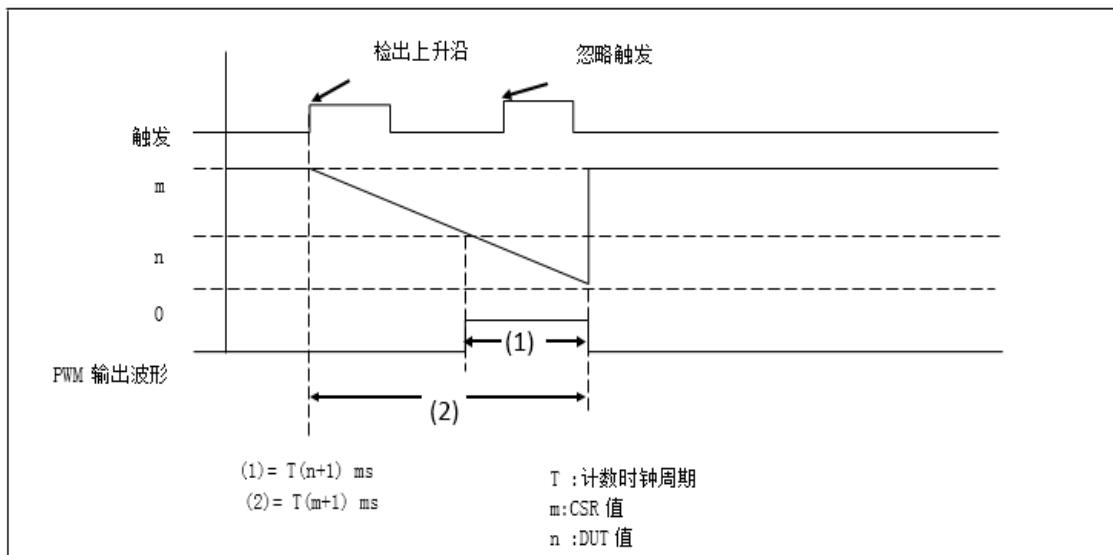
25.8.1.2 单次操作

单次操作时，定时器被通过触发会输出一个任意宽度的脉冲。当使能重启时，如果操作期间检出边沿，则重载计数器。

单次操作

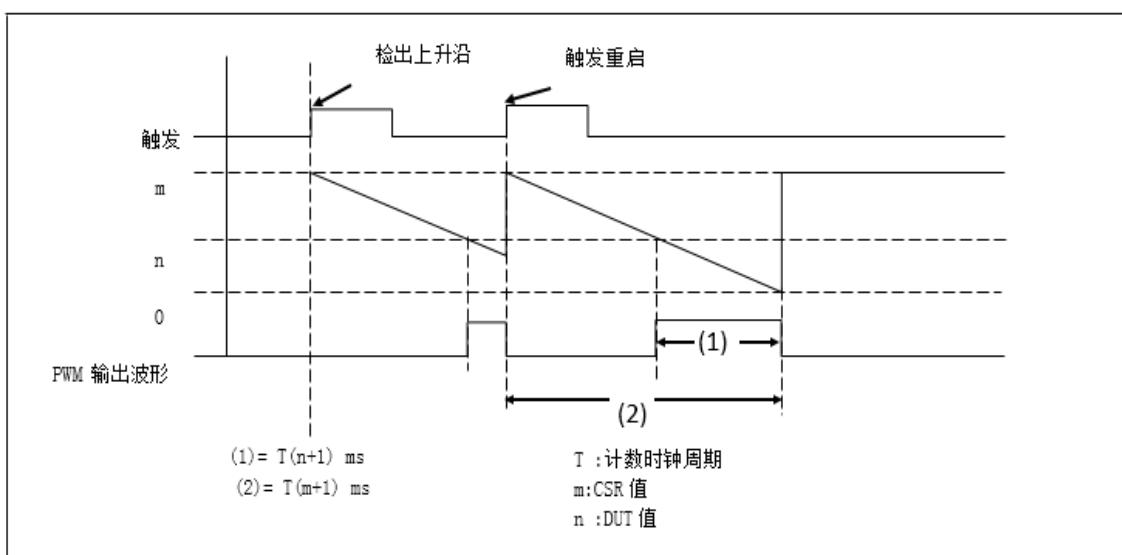
禁止重启(RSTEN="0")

图 25.8.1-3 单次操作时序图(禁止触发重启)



使能重启(RSTEN="1")

图 25.8.1-4 单次操作时序图(使能触发重启)



25.8.1.3 中断及时序图

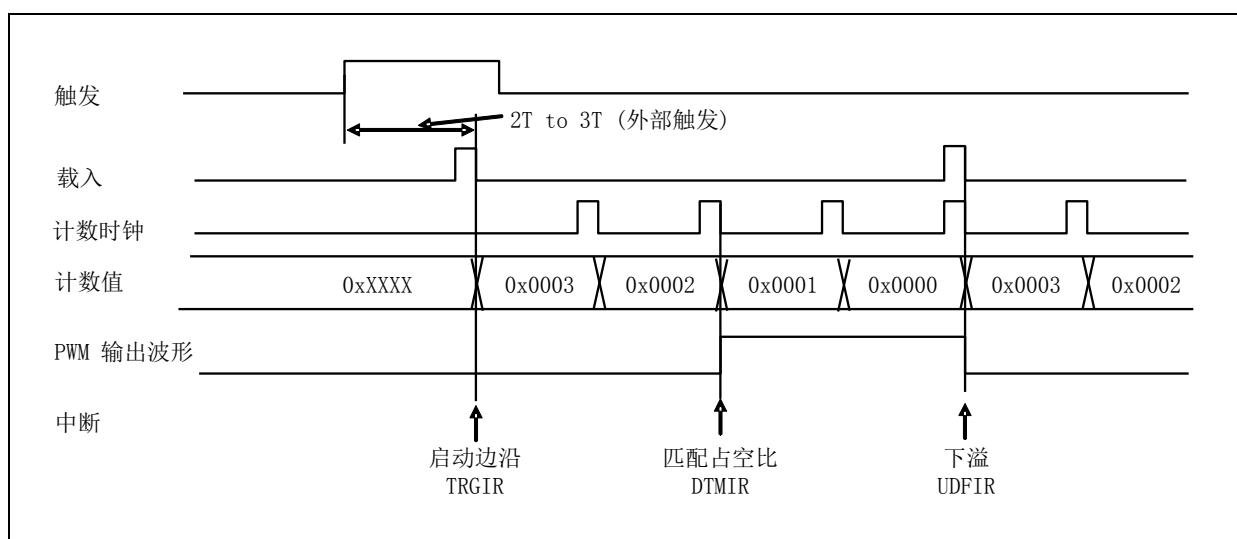
本节介绍中断源和时序图。

中断源和时序图(PWM 输出:正常极性)

输入触发后到载入计数器值的这段时间，软件触发需要 1T，外部触发需要 2T ~ 3T(T:机器周期)。

图 25.8.1-5 是周期设定值=3，占空比值=1 时的中断源和时序图。

图 25.8.1-5 PWM 定时器的中断源和时序图



25.8.1.4 输出波形

本节介绍说明了 PWM 输出。

PWM 输出全为“L”或全为“H”的输出方法

图 25.8.1-6 和图 25.8.1-7 分别是 PWM 输出全设为“L”或全设为“H”的输出方法。

图 25.8.1-6 PWM 输出全设为“L”电平示例

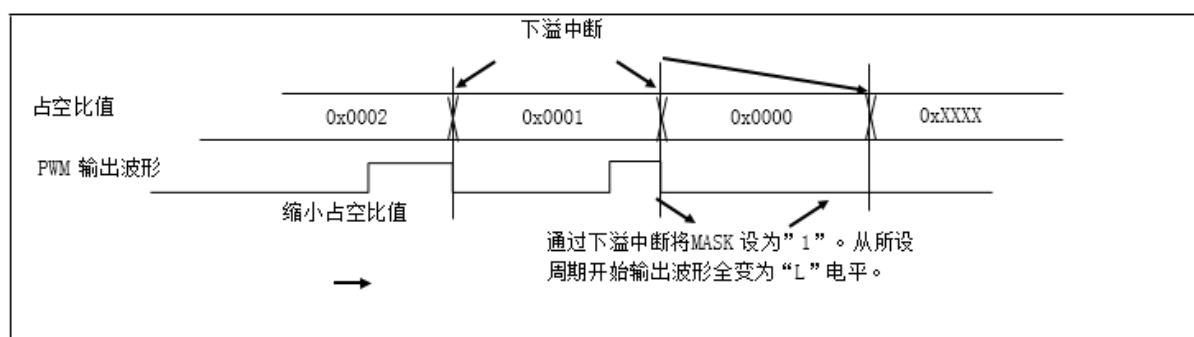
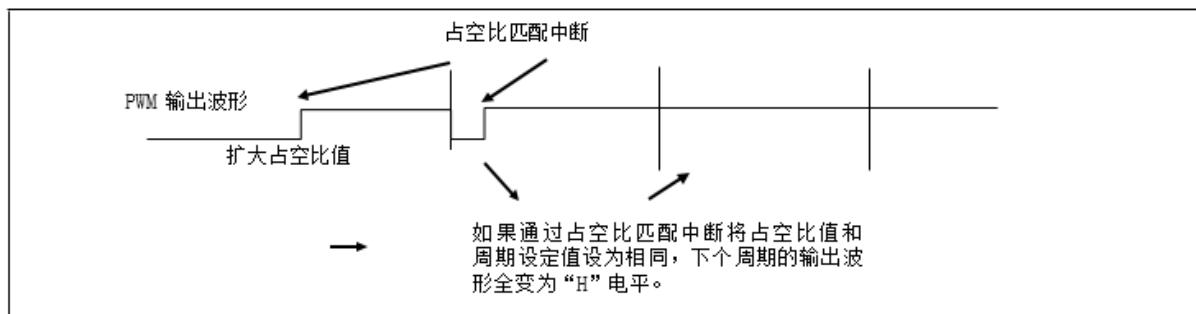


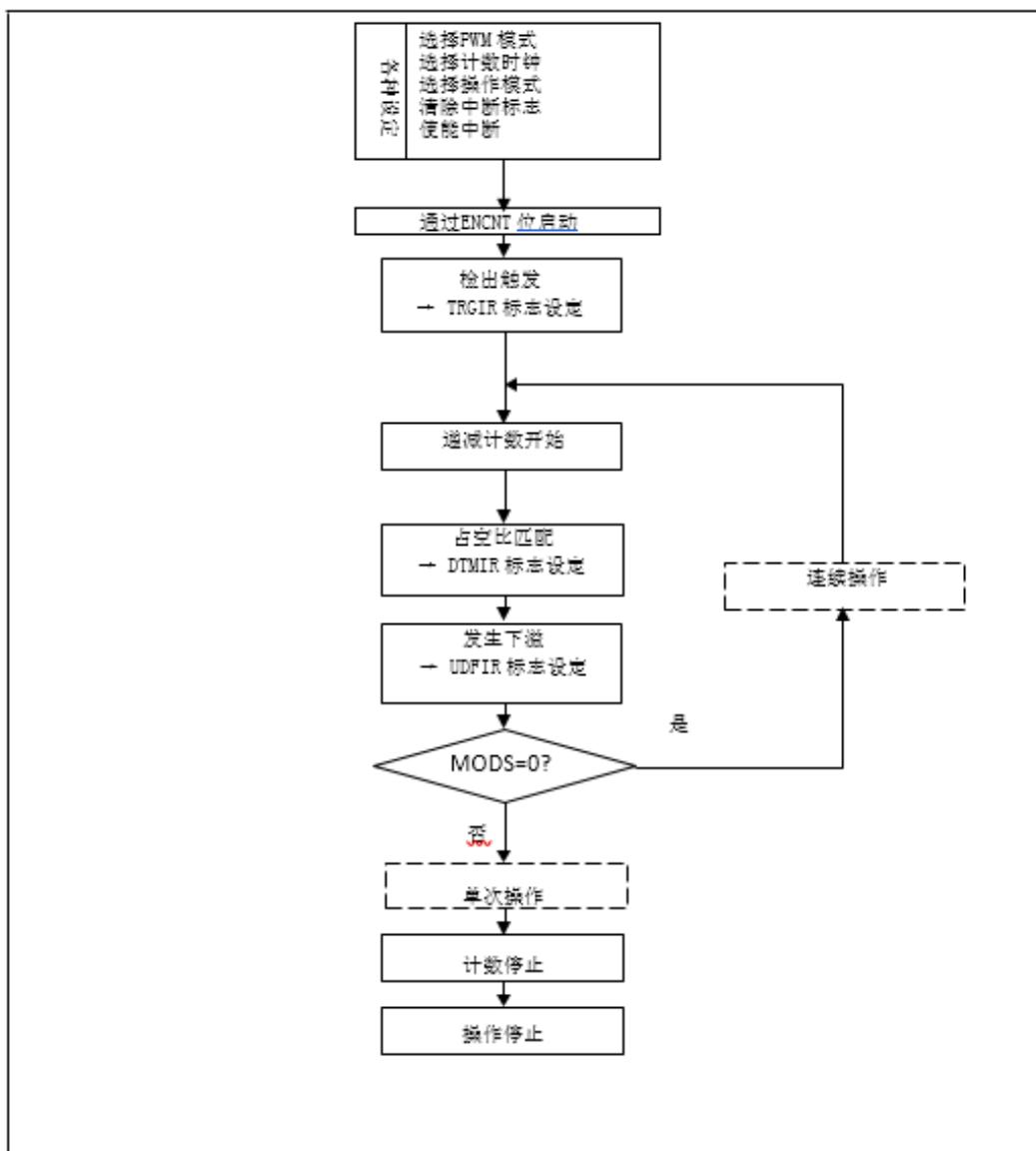
图 25.8.1-7 PWM 输出全设为 "H" 电平示例



25.8.1.5 PWM 定时器操作流程图

本节介绍 PWM 定时器的操作流程。

PWM 定时器操作流程图



25.8.1.6 选择 PWM 定时器时的定时器控制寄存器(TCR, TCR2), 状态控制寄存器(STA)

定时器控制寄存器(TCR)控制 PWM 定时器。注意 PWM 定时器操运行期间有些无法改写的位置。

定时器控制寄存器(TCR 高位字节)

bit	15	14	13	12	11	10	9	8
Field	res	CLKS2	CLKS1	CLKS0	RSTEN	MASK	EDGS1	EDGS
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit15] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit14:12, TCR2:bit8] CLKS3 ~ CLKS0:计数时钟选择位

选择 16 位递减计数器的计数时钟。

改变设定后, 计数时钟立刻得以改变。因此, 应在计数停止状态(ENCNT="0")下改变 CLKS3 ~ CLKS0。但是, 可向 ENCNT 位写"1"的同时进行改变。

CLKS3	CLKS2	CLKS1	CLKS0	说明
0	0	0	0	φ
0	0	0	1	$\varphi/4$
0	0	1	0	$\varphi/16$
0	0	1	1	$\varphi/128$
0	1	0	0	$\varphi/256$
0	1	0	1	外部时钟(上升沿事件)
0	1	1	0	外部时钟(下降沿事件)
0	1	1	1	外部时钟(双沿事件)
1	0	0	0	$\varphi/512$
1	0	0	1	$\varphi/1024$
1	0	1	0	$\varphi/2048$
其他				禁止设定

[bit11] RSTEN: 重启使能位该位通过软件触发或触发输入执行重启。

bit	说明
0	禁止重启
1	使能重启

[bit10] MASK:脉冲输出屏蔽位

控制 PWM 输出波形的输出波形电平。

该位清"0"时, 原样输出 PWM 波形。

该位为"1"时, 与周期和占空比设定值无关, 将 PWM 输出为"L"电平。

注意：

- Bit3 的 PLRT 取反输出时，如果将 MASK 设为"1"，则输出为"H"电平。

bit	说明
0	正常输出
1	固定为"L"电平输出

[bit9:8] EDGS1 ~ EDGS0:触发输入沿选择位

作为外部启动源，选择输入波形对应的有效沿，设定触发条件。

设为初始值或"0b00"时，由于尚未选择输入波形对应的有效沿，因此通过外部波形无法启动。

注意：

- 与 EDGS1, EDGS0 设定无关，如果将 START 位设为"1"，则软件触发有效。

应在计数停止状态(ENCNT="0")下改变 EDGS1, EDGS0。但是，可向 ENCN T 位写"1"的同时进行改变。

bit9	bit8	说明
0	0	触发输入无效
0	1	上升沿
1	0	下降沿
1	1	双沿

定时器控制寄存器(TCR 低位字节)

bit	7	6	5	4	3	2	1	0
Field	res	FUNC2	FUNC1	FUNC0	PLRT	MODS	ENCNT	START
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit7] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit6:4] FUNC2 ~ FUNC0:定时器功能选择位

该位选择定时器功能。

若将 FUNC2, FUNC1, FUNC0 位设为"0b001"，则选择 PWM 功能。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCN T 位写"1"的同时进行改变。

bit6	bit5	bit4	说明
0	0	0	复位模式
0	0	1	选择 PWM 功能
0	1	0	选择 PPG 功能

0	1	1	选择重载定时器功能
1	0	0	选择 PWC 功能
1	0	1	禁止设定
1	1	0	
1	1	1	

[bit3] PLRT: 输出极性指定位

设定 PWM 输出极性

极性	复位后	占空比匹配	下溢
正常	"L"输出		
取反	"H"输出		

bit	说明
0	正常极性
1	取反极性

[bit2] MODS: 模式选择位

该位选择连续输出脉冲或单次输出脉冲。

应在定时器停止期间(ENCNT="0")执行改变。但是，可向 ENCNT 位写"1"的同时进行改变。

bit	说明
0	连续操作
1	单次操作

[bit1] ENCNT: 计数操作使能位

该位使能递减计数器的操作。

计数器处于操作使能状态(ENCNT 位为"1")时，若向其写"0"，则计数器停止。

bit	说明
0	停止
1	使能操作

[bit0] START: 软件触发位

ENCNT 位为"1"时，若向 START 位写"1"，则开始软件触发。

START 位始终读"0"。

注意：

- 即使同时将 ENCNT 位和 START 位设为"1"，仍可启动软件触发。

- 与 EDGS1, EDGS0 的设定无关, 若向 START 位写"1", 则软件触发有效。

bit	说明
0	无效
1	软件启动开始

定时器控制寄存器 2 (TCR2)

bit	1	14	13	1	11	1	9	8
Field				r				CLKS3
属性				R/W				R/W
初始值				0b00000000				0

注意:

- 该寄存器配置到 STA 寄存器的高位。
- [bit15:9] res: 保留位读值为"0"。写寄存器时将该位写"0"。
- [bit8] CLKS3: 计数时钟选择位
参考“定时器控制寄存器(TCR)”的“计数时钟选择位”。

状态控制寄存器(STA)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	DTMIE	UDFIE	res	TRGIR	DTMIR	UDFIR
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

注意:

- 该寄存器的高位字节配置给 TCR2 寄存器。
- [bit7] res: 保留位读值为"0"。写寄存器时将该位写"0"。
- [bit6] TGIE: 触发中断请求使能位
控制 bit2 TRGIR 的中断请求。
若使能 TGIE 位后设定 bit2 TRGIR 位, 那么 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

- [bit5] DTMIE: 占空比匹配中断请求使能位
控制 bit1 DTMIR 的中断请求。
若使能 DTMIE 位后设定 bit1 DTMIR 位, 那么 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit4] UDFIE: 下溢中断请求使能位

控制 bit0 UDFIR 的中断请求。

若使能 UDFIE 位后设定 bit0 UDFIR 位，那么 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit3] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit2] TRGIR: 触发中断请求位

检测出软件触发或触发输入时，TRGIR 位设为"1"。

向 TRGIR 位写"0"可将其清零。

向 TRGIR 位写"1"不影响该位的值。

通过读-修改-写类指令始终读"1"。

bit	说明
0	清除中断源
1	检出中断源

[bit1] DTMIR: 占空比匹配中断请求位

计数值与占空比设定值匹配时，DTMIR 位为"1"。

向 DTMIR 位写"0"将其清零。

向 DTMIR 位写"1"不影响该位的值。

通过读-修改-写类指令始终读"1"。

bit	说明
0	清除中断源
1	检出中断源

[bit0] UDFIR: 下溢中断请求位

计数值按"0x0000"→"0xFFFF"下溢时，UDFIR 位为"1"。

向 UDFIR 位写"0"将其清零。

向 UDFIR 位写"1"不影响该位的值。

通过读-修改-写类指令始终读"1"。

bit	说明

0	清除中断源
1	检出中断源

25.8.1.7 PWM 周期设定寄存器 (CSR)

PWM 周期设定寄存器(CSR)是用于设定周期的缓冲寄存器。启动时/下溢时向定时器寄存器传输数据。

bit	15	0
Field	CSR[15:0]	
属性	R/W	
初始值	0xXXXX	

该寄存器是设定周期缓冲寄存器。启动时/下溢时向定时器寄存器传输数据。

周期设定寄存器的初始设定时或重写时，写到周期设定寄存器后，必须向占空比设定寄存器写值。

通过 16 位数据访问 CSR 寄存器。

CSR 寄存器使用 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位在设定 PWM 功能后设定周期。

25.8.1.8 PWM 占空比设定寄存器(DUT)

PWM 占空比设定寄存器(DUT)是用于设定占空比的缓冲寄存器。下溢时，从缓冲器传输数据。

bit	15	0
Field	DUT[15:0]	
属性	R/W	
初始值	0xXXXX	

该寄存器是用于设定占空比的缓冲寄存器。下溢时，从缓冲器传输数据。

如果周期设定寄存器的值和占空比设定寄存器的值相同，正常极性时全输出"H"，取反极性时全输出"L"。

切勿设为 CSR<DUT，PWM 输出不定

- 按 16 位数据访问 DUT 寄存器。
- DUT 寄存器使用 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位在设定 PWM 功能后设定占空比。

25.8.1.9 定时器寄存器(TCNT)

定时器寄存器(TCNT)读取 16 位递减计数器的值。

bit	15	0
Field	TCNT[15:0]	

属性	R	
初始值	0x0000	

读取 16 位递减计数器的值

按 16 位数据读取 TCNT 寄存器

25.8.2 PPG 定时器功能

复合定时器通过设定定时器控制寄存器的 FUNC2, 1, 0 位, 从 16 位 PWM 定时器, 16 位 PPG 定时器, 16/32 位重载定时器, 16/32 位 PWC 定时器中选择一种定时器功能。本节介绍设定 PPG 后的定时器功能。

1. 16 位 PPG 定时器操作
2. 连续操作
3. 单次操作
4. 中断源和时序图
5. PPG 定时器操作流程
6. 选择 PPG 定时器时的定时器控制寄存器(TCR, TCR2), 状态控制寄存器(STA)
7. "L"脉宽设定重载寄存器(LWRR)
8. "H"脉宽设定重载寄存器(HWRR)
9. 定时器寄存器(TCNT)

25.8.2.1 16 位 PPG 定时器操作

PPG 定时器操作时, 将输出脉冲的 "L" 脉宽和 "H" 脉宽分别设定到相应的重载寄存器, 可控制任意的输出脉冲。

操作概要

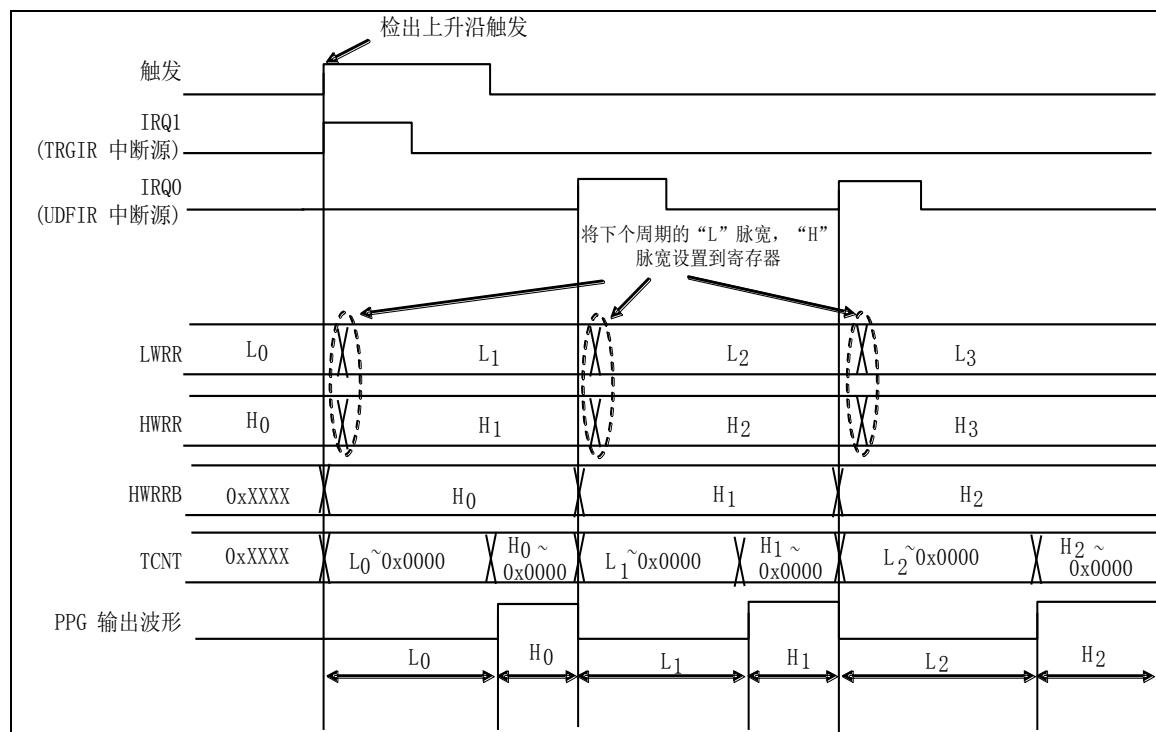
16 位长度的重载寄存器包含 "L" 脉宽设定和 "H" 脉宽设定, 和一个 "H" 脉宽设定缓冲器(LWRR, HWRR, HWRRB)。

通过启动触发, 向 16 位递减计数器最初载入 LWRR 设定值, 同时向 HWRRB 传输 HWRR 设定值。将 PPG 输出电平设为 "L" 电平, 每个计数时钟进行递减计数。检出下溢时, HWRRB 值重载入计数器, 取反 PPG 输出波形并继续递减计数。再次检出下溢时, 再取反 PPG 输出波形, 将 LWRR 设定值重载入计数器, 将 HWRR 设定值传输到 HWRRB。

通过上述操作, 输出波形变为持有对应各种重载寄存器值的 "L" 脉宽和 "H" 脉宽的脉冲输出。

重载寄存器写入时序

检出启动触发时以及设定下溢中断源(UDFIR)后，切换到下一个周期前这段时间，向重载寄存器 LWRR, HWRR 写数据。这时设定的数据成为下一个周期的设定数据。检出启动触发时“H”脉宽计数完成时的下溢时，LWRR, HWRR 中的设定数据分别自动传输到 TCNT 和 HWRRB。 “L”脉宽计数完成时的下溢时，传输到 HWRRB 的数据自动重载到 TCNT。



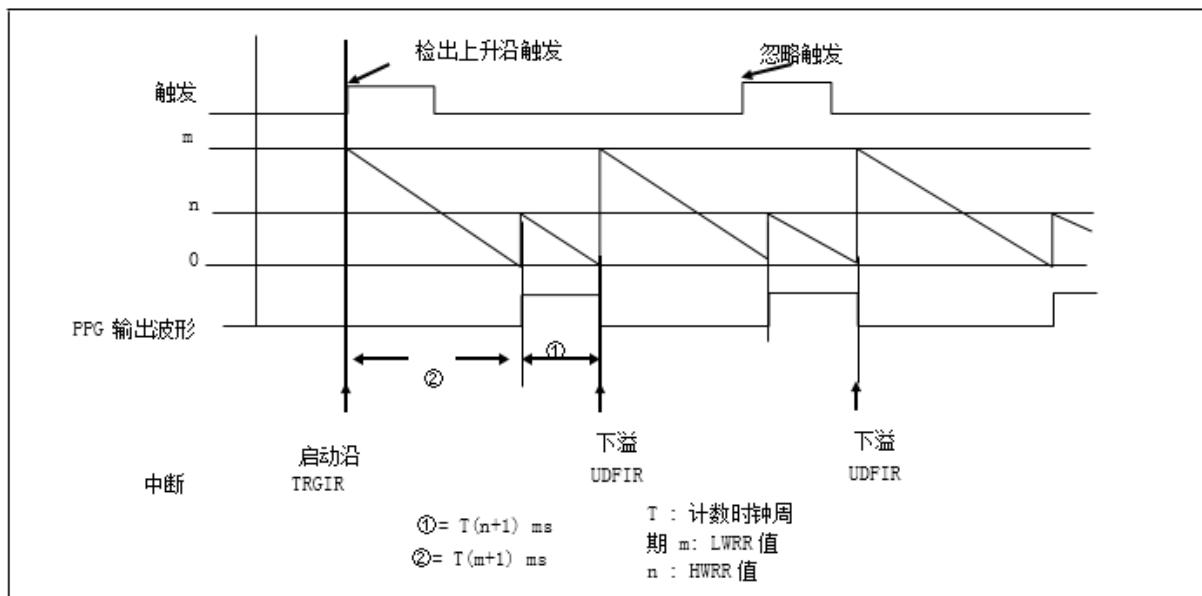
25.8.2.2 连续操作

连续操作时，根据各中断源的设定期序更新“L”脉宽和“H”脉宽后，可连续输出任意脉冲。使能重启时，如果操作期间检出边沿，则重载计数器。

连续操作

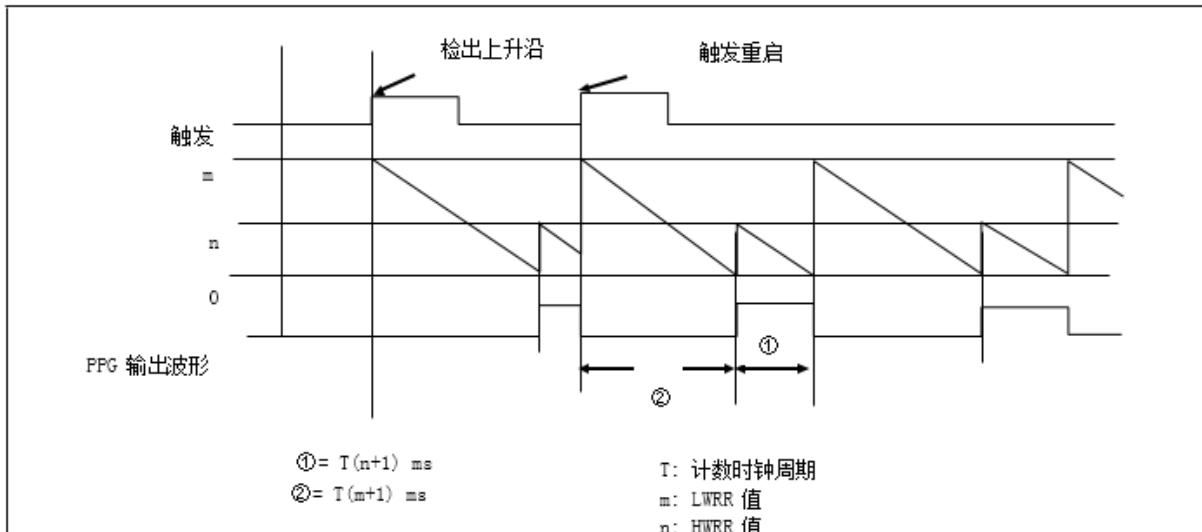
禁止重启(RSTEN="0")

图 25.8.2-8 PPG 操作时序图(禁止重启)



使能重启(RSTEN="1")

图 25.8.2-9 PPG 操作时序图(使能重启)



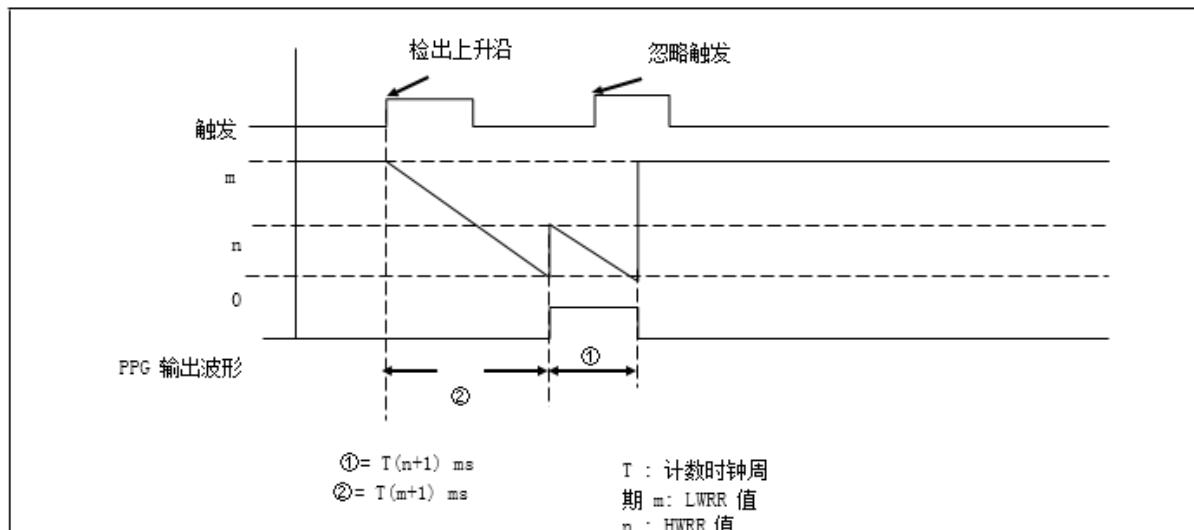
25.8.2.3 单次操作

单次操作时，触发后可输出任意宽的单一脉冲。使能重启时，如果操作期间检出边沿，则重载计数器。

单次操作

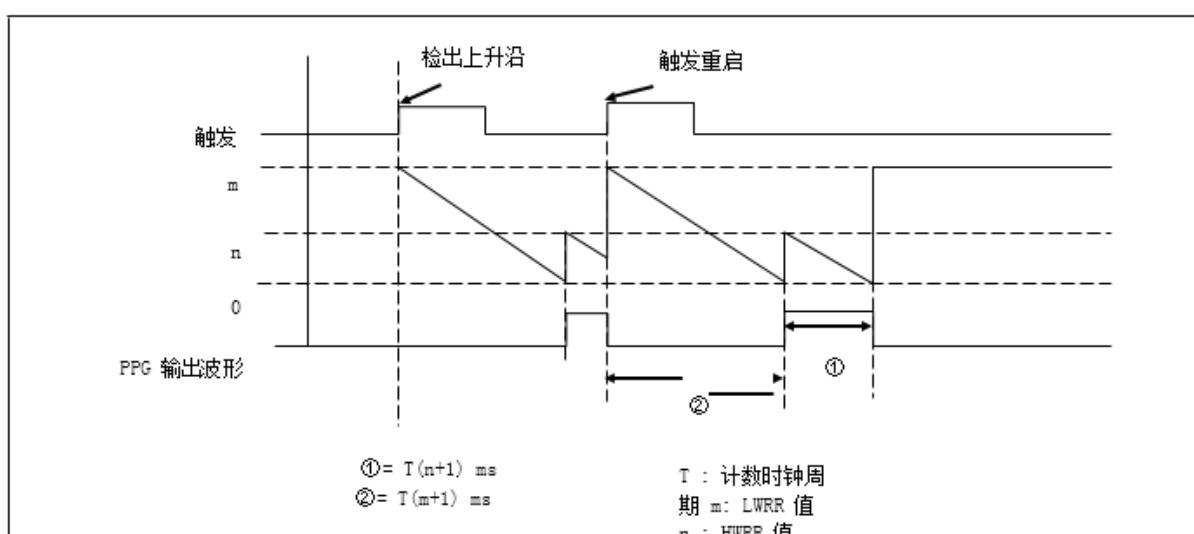
禁止重启(RSTEN="0")

图 25.8.2-10 单次操作时序图(禁止触发重启)



使能重启(RSTEN="1")

图 25.8.2-11 单次操作时序图(使能触发重启)



重载值和脉宽的关系

将写到 16 位数据的重载寄存器的值+1 后，乘以计数时钟周期的值即为输出脉宽。因此，重载寄存器值为"0x0000"时，为计数时钟 1 个周期的脉宽。此外，重载寄存器值为"0xFFFF"时，

为计数时钟 65536 个周期的脉宽。以下是脉宽计算公式：

$$PL = T \times (L+1)$$

PL: "L" 脉宽

$$PH = T \times (H+1)$$

PH: "H" 脉宽

T: 计数时钟周期

L: LWRR 值

H: HWRR 值

25.8.2.4 中断源和时序图

本节介绍中断源和时序图。

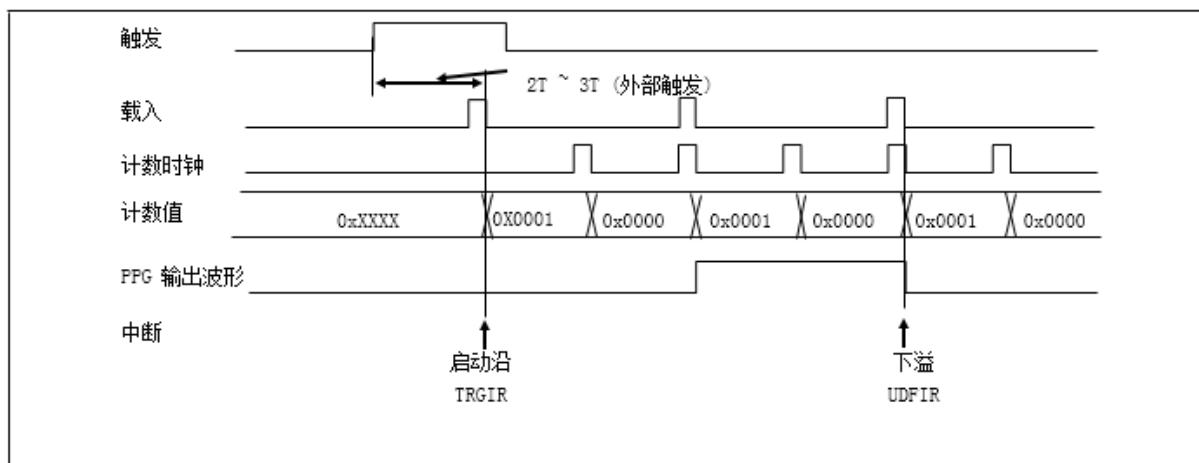
中断源和时序图(PPG 输出:正常极性)

触发之后到载入计数值前，软件触发需要 1T，外部触发需要 2T ~ 3T (T:机器周期)。

检出 PPG 启动触发时和检出"H"电平输出时的下溢时，设定中断源。

图 25.8.2-12 是"L"脉宽设定值="1"，"H"脉宽设定值="1"时的中断源和时序图。

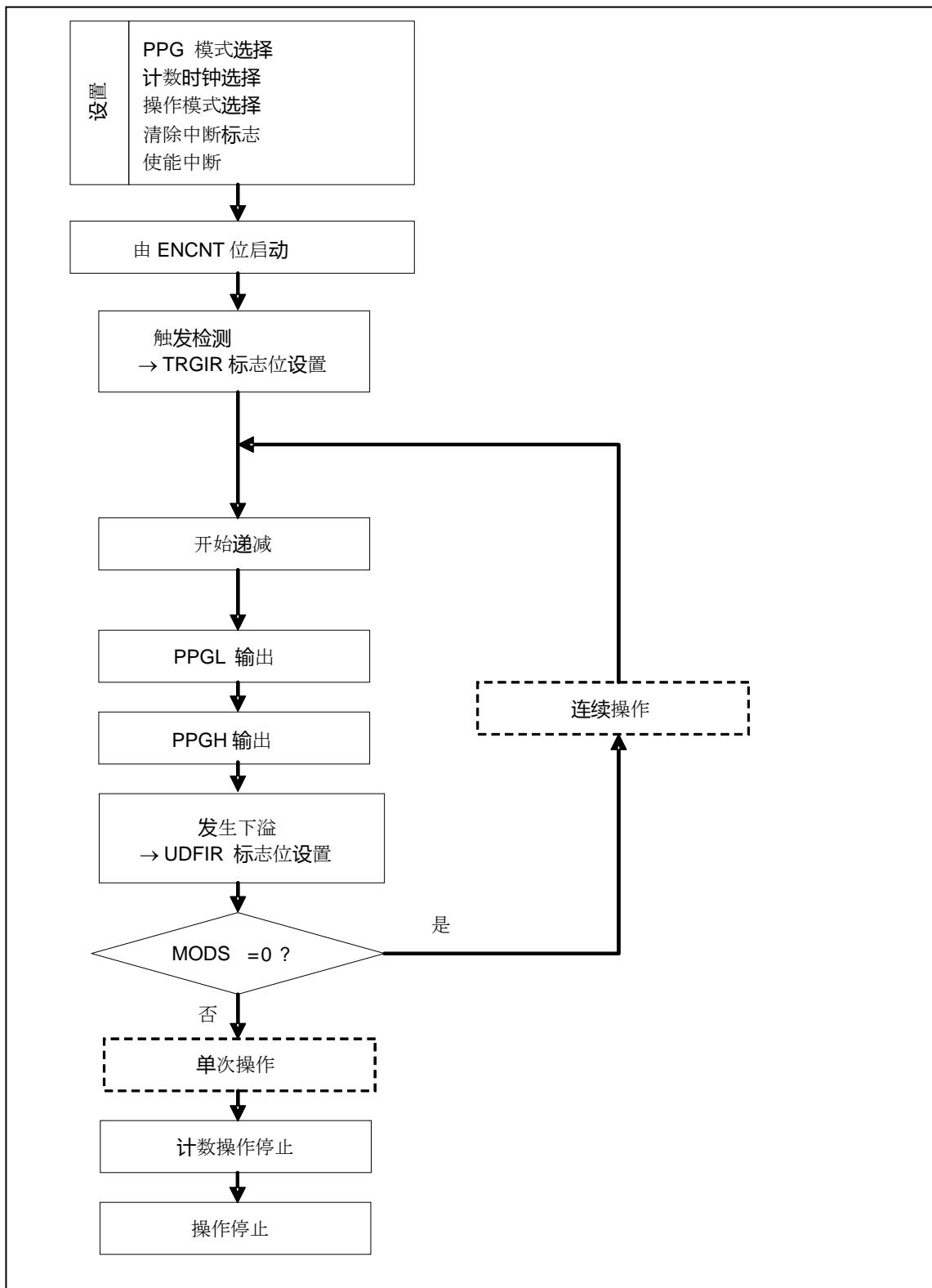
图 25.8.2-12 PPG 定时器的中断源和时序图



25.8.2.5 PPG 定时器操作流程

本节是 PPG 定时器的操作流程。

PPG 定时器操作流程



25.8.2.6 选择 PPG 定时器时的定时器控制寄存器(TCR, TCR2),状态控制寄存器(STA)

定时器控制寄存器(TCR)控制 PPG 定时器。注意 PPG 定时器操作期间一些位无法改写。

定时器控制寄存器(TCR 高位字节)

bit	15	14	13	12	11	10	9	8
Field	res	CLKS2	CLKS1	CLKS0	RSTEN	MASK	EDGS1	EDGS0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit15] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit14:12, TCR2:bit8] CLKS3 ~ CLKS0: 计数时钟选择位

选择 16 位递减计数器的计数时钟。

更改设定后，计数时钟立刻得以改变。因此，计数停止状态(ENCNT="0")下应改变 CLKS3 ~ CLKS0。但向 ENCNT 位写"1"的同时可进行改变。

CLKS3	CLKS2	CLKS1	CLKS0	说明
0	0	0	0	φ
0	0	0	1	$\varphi/4$
0	0	1	0	$\varphi/16$
0	0	1	1	$\varphi/128$
0	1	0	0	$\varphi/256$
0	1	0	1	外部时钟(上升沿事件)
0	1	1	0	外部时钟(下降沿事件)
0	1	1	1	外部时钟(双沿事件)
1	0	0	0	$\varphi/512$
1	0	0	1	$\varphi/1024$
1	0	1	0	$\varphi/2048$
其他				禁止设定

[bit11] RSTEN: 重启使能位该位使能软件触发或触发输入时的重启。

bit	说明
0	禁止重启
1	使能重启

[bit10] MASK: 脉冲输出屏蔽位

控制 PPG 输出波形的输出波形电平。

该位为"0"时，原样输出 PPG 波形。

该位为"1"时，与周期和占空比设定值无关，将 PPG 输出屏蔽为"L"输出。

注意：

- Bit3 的 PLRT 设为取反输出时，如果将 MASK 设为"1"，屏蔽为"H"输出。

bit	说明
0	正常输出
1	固定为"L"输出

[bit9:8] EDGS1 ~ EDGS0: 触发输入沿选择位

作为外部启动源，选择输入波形对应的有效沿，设定触发条件。

设为初始值或"0b00"时，由于处于未选择输入波形对应的有效沿，因此无法通过外部波形启动。

注意：

- 与 EDGS1, EDGS0 的设定无关，如果向 START 位写"1"，软件触发有效。

应在计数停止状态(ENCNT="0")下改变 EDGS1, EDGS0。但可向 ENCTN 位写"1"的同时进行改变。

bit9	bit8	说明
0	0	触发输入无效
0	1	上升沿
1	0	下降沿
1	1	双沿

定时器控制寄存器(TCR 低位字节)

bit	7	6	5	4	3	2	1	0
Field	res	FUNC2	FUNC1	FUNC0	PLRT	MODS	ENCNT	START
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit7] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit6:4] FUNC2 ~ FUNC0: 定时器功能选择位

该位选择定时器功能。

如果将 FUNC2, FUNC1, FUNC0 位设为"0b010"，则选择 PPG 功能。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCTN 位写"1"的同时进行改变。

			说明
0			复位模式
0			选择PWM 功能
0			选择PPG 功能
0			选择重载定时器功能
1			选择PWC 功能

1	0	1	
1	1	0	
1	1	1	禁止设定

[bit3] PLRT: 输出极性指定位

设定 PPG 输出极性。

极性	复位后	"L"脉宽计数结束	"H"脉宽计数结束
正常极性	"L"输出		
取反极性	"H"输出		

bit	说明
0	正常极性
1	取反极性

[bit2] MODS: 模式选择位

该位选择连续输出脉冲还是单次输出脉冲。

定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变。

bit	说明
0	连续操作
1	单次操作

[bit1] ENCNT: 计数操作使能位

该位使能递减计数器的操作。

计数器处于操作使能状态(ENCNT 位="1")时, 若向其写"0", 则计数器停止操作。

bit	说明
0	停止
1	使能操作

[bit0] START: 软件触发位

ENCNT 位为"1"时, 若向 START 位写"1", 则软件触发有效。

START 位始终读"0"。

注意:

- 即使同时将 ENCNT 位和 START 位设为"1", 软件触发仍然有效。
- 与 EDGS1,0 的设定无关, 若向 START 位写"1", 则软件触发有效。

bit	说明
0	无效

1	软件启动开始
---	--------

定时器控制寄存器 2 (TCR2 高位字节)

bit	15	14	13	12	11	10	9	8
Field				res				CLKS3
属性				R/W				R/W 0

初始值: 0b00000000

注意:

- 该寄存器配置到 STA 寄存器高位。

[bit15:9] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit8] CLKS3:计数时钟选择位

参考"定时器控制寄存器(TCR 高位字节)"的"计数时钟选择位"。

状态控制寄存器(STA)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	res	UDFIE	res	TRGIR	res	UDFIR
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初始值: 0 0 0 0 0 0 0 0 0

注意:

- 该寄存器的高位字节分配给 TCR2 寄存器。

[bit7] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit6] TGIE: 触发中断请求使能位

该位控制 bit2 TRGIR 的中断请求。

如果使能 TGIE 位后设定 bit2 TRGIR 位，则 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit5] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit4] UDFIE: 下溢中断请求使能位

该位控制 bit0 UDFIR 的中断请求。

如果使能 UDFIE 位后设定 bit0 UDFIR 位，则 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit3] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit2] TRGIR: 触发中断请求位

软件触发或检出触发输入时，TRGIR 位为"1"。

向 TRGIR 位写"0"可将其清零。

即使向 TRGIR 位写"1"也不会影响位值。

读-修改-写类指令始终读"1"。

bit	说明
0	清零中断源
1	检出中断源

[bit1] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit0] UDFIR: 下溢中断请求位

从"H"脉宽设定值开始计数的期间，计数值如"0x0000"→"0xFFFF"发生下溢时，UDFIR 位设为"1"。

向 UDFIR 位写"0"可将其清零。

即使向 UDFIR 位写"1"也不会影响位值。

读-修改-写类指令始终读"1"。

bit	说明
0	清零中断源
1	检出中断源

25.8.2.7 "L"脉宽设定重载寄存器(LWRR)

"L"脉宽设定重载寄存器(LWRR)用于设定 PPG 输出波形的"L"脉宽。检出启动触发时或"H"脉宽计数完成后下溢时向定时器寄存器传输数据。

bit	15		0
Field		LWRR[15:0]	
属性		R/W	
初始值		0xXXXX	

该寄存器用于设定 PPG 输出波形的"L"脉宽。检出启动触发时或"H"脉宽计数完成后下溢时向定时器寄存器传输数据。

按 16 位数据访问 LWRR 寄存器。

LWRR 寄存器在使用 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位设定 PPG 功能后设定"L"脉宽。

25.8.2.8 "H"脉宽设定重载寄存器(HWRR)

"H"脉宽设定重载寄存器(HWRR)是用于设定 PPG 输出波形"H"脉宽的缓冲寄存器。检出启动触发时或"H"脉宽计数完成后下溢时从 HWRR 向缓冲寄存器传输数据。"L"脉宽计数完成时发生下溢时，从缓冲寄存器向定时器寄存器传输数据。

bit	15		0
Field		HWRR[15:0]	
属性		R/W	
初始值		0xXXXX	

该寄存器设定 PPG 输出波形的"H"脉宽。检出启动触发时或"H"脉宽计数完成后发生下溢时，从 HWRR 向缓冲寄存器传输数据。"L"脉宽计数完成后发生下溢时，从缓冲寄存器向定时器寄存器传输数据。

按 16 位数据访问 HWRR 寄存器。

HWRR 寄存器在使用 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位设定 PPG 功能后设定"H"脉宽。

25.8.2.9 定时器寄存器(TCNT)

定时器寄存器(TCNT)读取 16 位递减计数器值。

bit	15		0
Field		TCNT[15:0]	
属性		R	
初始值		0x0000	

读取 16 位递减计数器的值

按 16 位数据访问 TCNT 寄存器。

25.8.3 重载定时器功能

复合定时器通过设定定时器控制寄存器的 FUNC2, 1, 0 位可从 16 位 PWM 定时器, 16 位 PPG 定时器, 16/32 位重载定时器, 16/32 位 PWC 定时器中选择一种定时器功能。本节介绍设定重载定时器时的定时器功能。

1. 16 位重载定时器的操作
2. 重载定时器操作流程
3. 选择选择重载定时器时的定时器控制寄存器(TCR, TCR2), 状态控制寄存器(STA)
4. 周期设定寄存器(CSR)
5. 定时器寄存器(TCNT)

25.8.3.1 16 位重载定时器的操作

重载定时器操作中，递减操作与计数时钟同步，从周期设定寄存器中的设定值开始递减计数。计数值为"0"时，停止计数或自动载入周期设定值并在停止递减计数前继续操作。

选择内部时钟时的计数操作

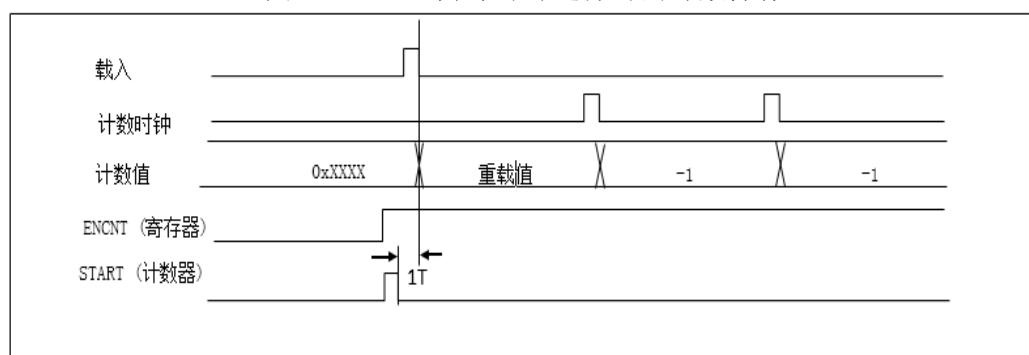
如需使能计数的同时开始计数操作，应向定时器控制寄存器的 ENCNT 位和 START 位都写 "1"。当定时器处于工作状态时(ENCNT="1")， START 位有效并且作为触发输入，与工作模式无关。

如果使能计数操作后通过软件触发或外部触发启动定时器，则将周期设定寄存器的值载入计数器并开始递减计数。

设定计数器启动触发后，周期设定寄存器的数据载入计数器前，需要 1T (T:机器周期)的时间。

图 25.8.3-13 介绍通过软件触发启动计数器的方法及其操作。

图 25.8.3-13 内部时钟选择时的计数操作



下溢操作

计数值从"0x0000"变为"0xFFFF"时，发生下溢。因此在计数为 (周期设定寄存器的设定值+1)

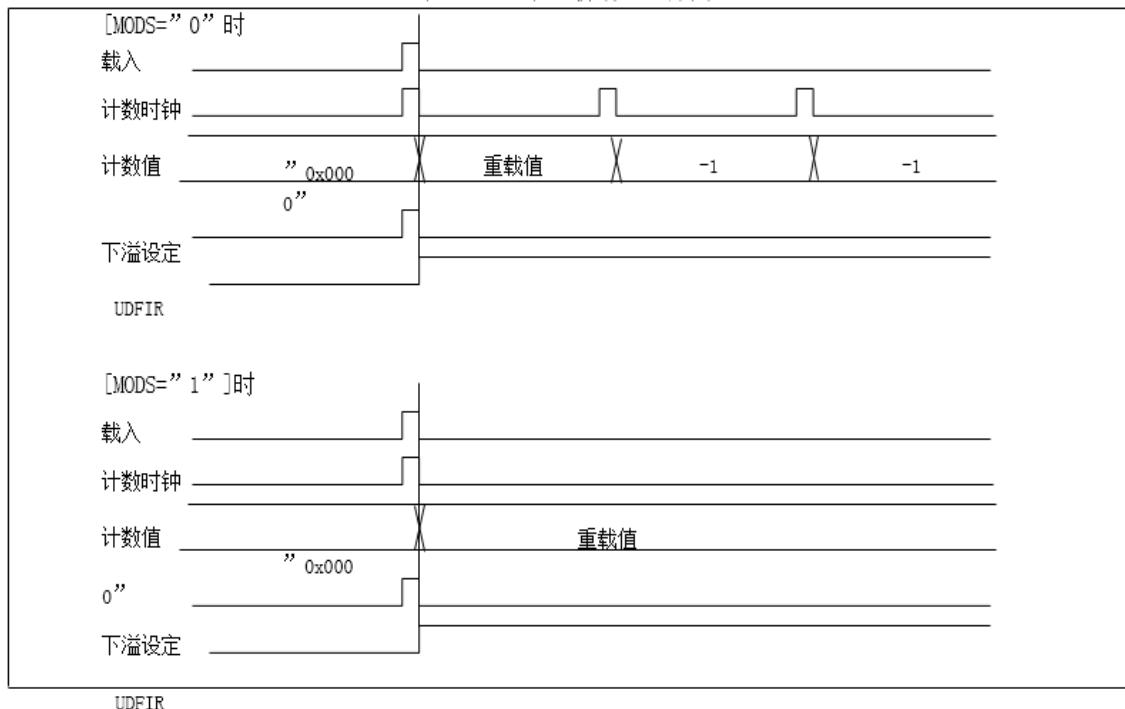
时发生下溢。

发生下溢时，周期设定寄存器(CSR)的内容载入计数器。定时器控制寄存器(TCR)的 MODS 位为"0" 时，计数操作继续。MODS 位为"1"时，载入的计数值保持不变，计数停止。

下溢时，设定状态控制寄存器(STA)的 UDFIR 位。这时，如果 UDFIE 位为"1"，发生中断请求。

图 25.8.3-14 介绍下溢操作时序图。

图 25.8.3-14 下溢操作的时序图

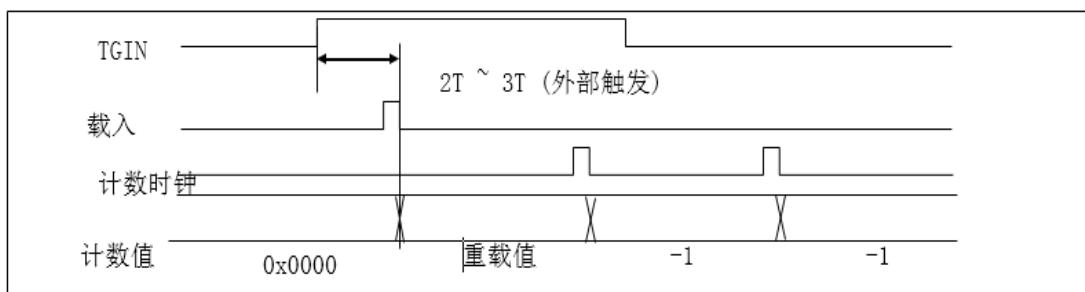


输入引脚功能

TGIN 引脚可用于触发输入。如果有效沿输入到 TGIN 引脚，那么将周期设定寄存器载入计数器并开始计数操作。触发生效后，载入计数器值前，需要 2T ~ 3T(T:机器周期)。将定时器控制寄存器 2 的门控使能位设置为“1”(GTIE=1)时，TGIN 引脚可用于触发输入。

图 25.8.3-15 是将有効沿指定为上升沿时的触发输入操作。

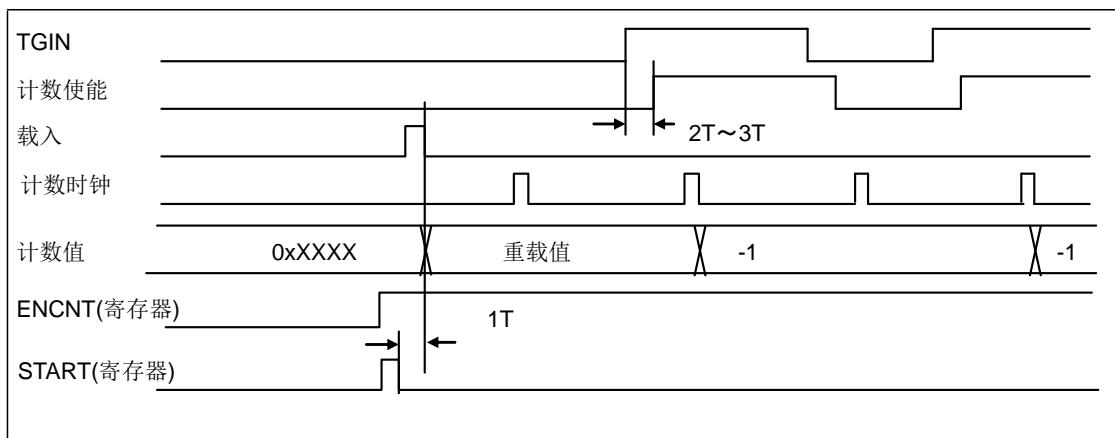
图 25.8.3-15 触发输入操作



定时器控制寄存器 2 的门控输入使能位被设置为“1” (GTIE=1)时，通过软件触发使能计数操作和启动计时器可将周期设定寄存器值装载入计数器并在有效电平输入 TGIN 引脚时使用计数时钟进行递减计数。在有效电平输入和使能计数之间，需要 2T 至 3T (T: 机器周期) 的时间。

图 25.8.3-16 所示为指定无效电平被设置为“高”时的递减操作。

图 25.8.3-16 使能门控输入时(GTIE=1)的操作

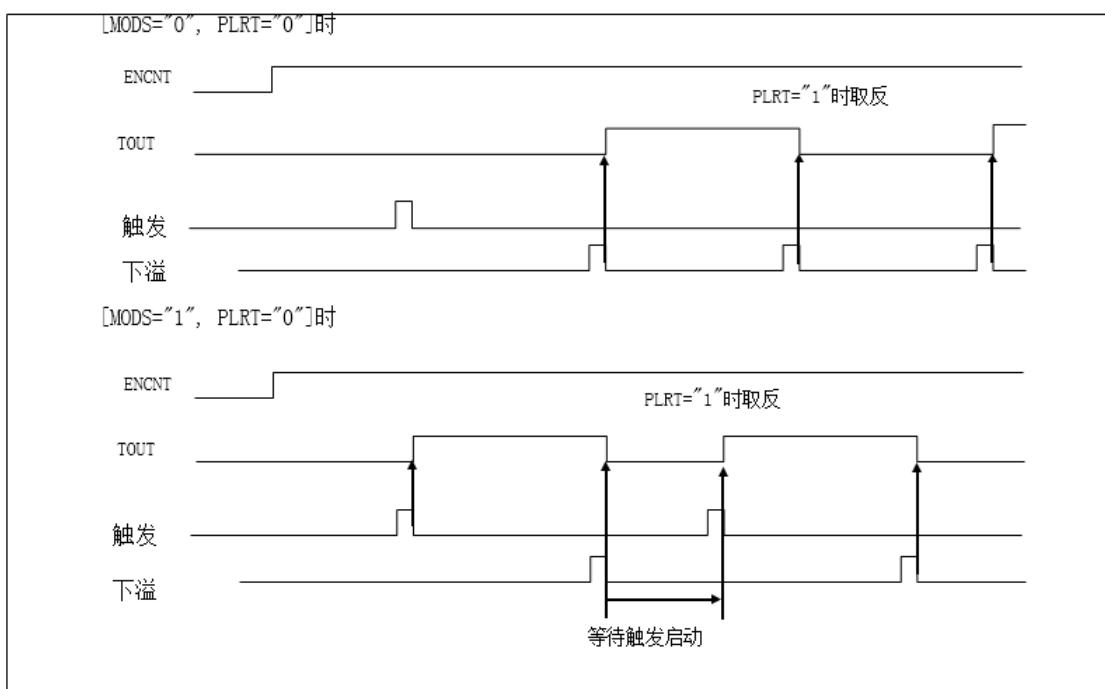


输出引脚功能

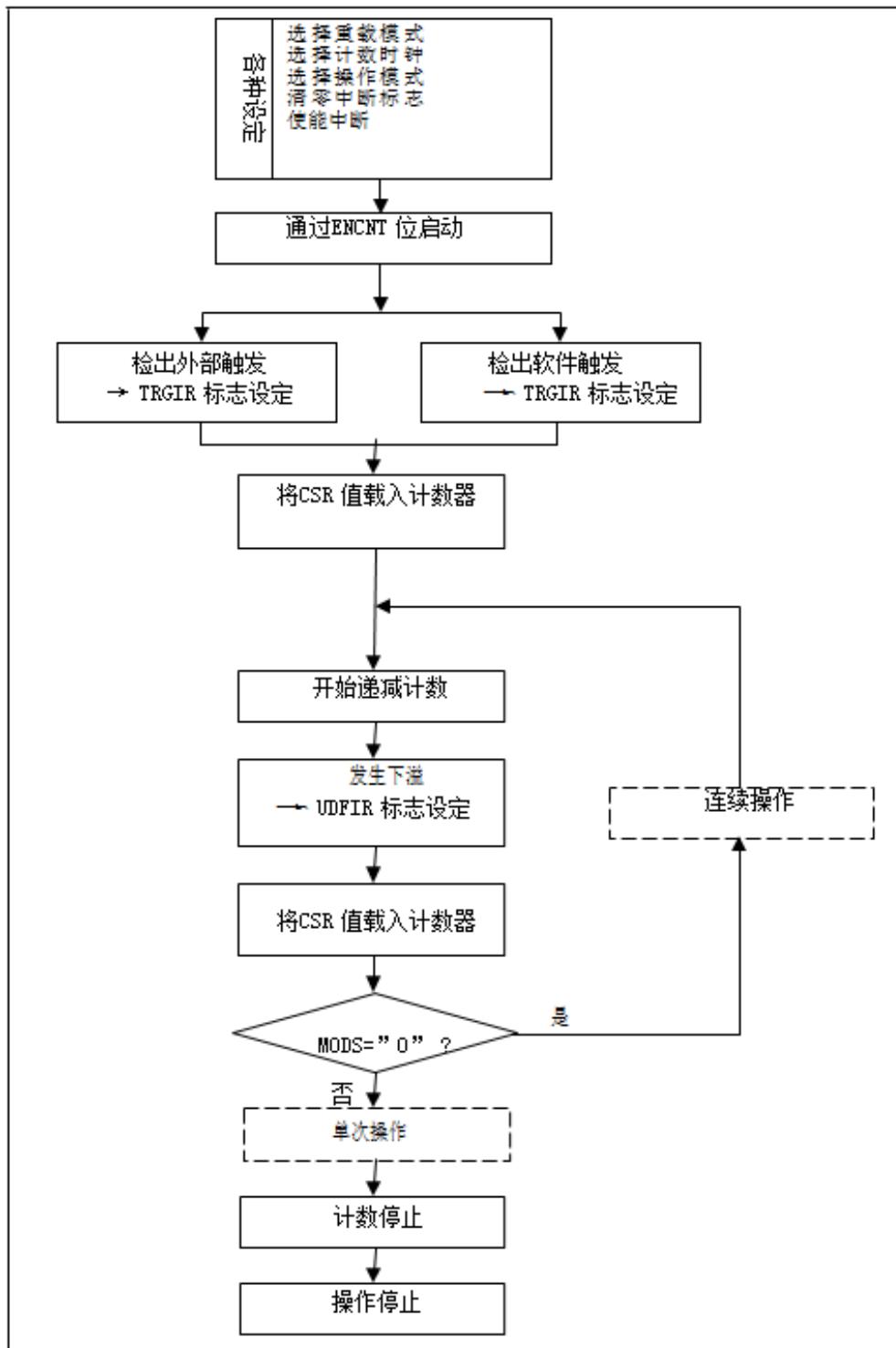
TOUT 输出引脚在重载模式时通过下溢取反输出，单次模式时作为脉冲输出指示正在计数。输出极性可由定时器控制寄存器(TCR)的 PLRT 位设定。PLRT="0"时，跳转输出初始值为"0"，单次脉冲输出在计数期间输出"1"。如果设为 PLRT="1"，输出波形取反。

图 25.8.3-17 是输出引脚功能的时序图。

图 25.8.3-17 输出引脚功能时序图



25.8.3.2 重载定时器操作流程



25.8.3.3 选择重载定时器时的定时器控制寄存器(TCR, TCR2), 状态控制寄存器(STA)

定时器控制寄存器(TCR)控制定时器操作

定时器控制寄存器(TCR 高位字节)

bit	15	14	13	12	11	10	9	8
Field	res	CLKS2	CLKS1	CLKS0		res	EDGS1	EDGS0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0b00	0	0	0

[bit15] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit14:12, TCR2:bit8] CLKS3 ~ CLKS0: 计数时钟选择位

选择 16 位递减计数器的计数时钟。

设定改变后, 计数时钟立刻改变。因此, 应在计数停止状态下(ENCNT="0")改变 CLKS3 ~ CLKS0。但可向 ENCNT 位写"1"的同时进行改变。

CLKS3	CLKS2	CLKS1	CLKS0	说明
0	0	0	0	φ
0	0	0	1	$\varphi/4$
0	0	1	0	$\varphi/16$
0	0	1	1	$\varphi/128$
0	1	0	0	$\varphi/256$
0	1	0	1	外部时钟(上升沿事件)
0	1	1	0	外部时钟(下降沿事件)
0	1	1	1	外部时钟(双沿事件)
1	0	0	0	$\varphi/512$
1	0	0	1	$\varphi/1024$
1	0	1	0	$\varphi/2048$
其他			禁止设定	

[bit11:10] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit9:8] EDGS1 ~ EDGS0: 触发输入沿和门控功能电平选择位

选择触发输入时(GTIE=0)时, 这些位选择输入波形对应的有效沿作为外部启动源并设定触发条件。

选择触发输入时(GTIE=0)且设为初始值或"0b00"时, 由于尚未选择输入波形对应的有效沿, 因此无法通过外部波形启动。

选择门控功能时(GTIE=1)时, 这些位选择输入波形对应的有效沿作为外部启动源, 当指定的边沿有效时递减计数。

只能在计数停止时(ENCNT=0)时改变 EDGS1, EDGS0 的值，也可在设置 ENCNT=1 的同时进行变更。

注意事项：

与 EDGS1, EDGS0 设定无关，如果向 START 位写"1"，软件触发有效。

应在计数停止状态(ENCNT="0")下改变 EDGS1, EDGS0。但可向 ENCNT 位写"1"的同时进行改变。

bit9	bit8	说明	
		选择触发输入时 (GTIE=0)	选择门控功能时 (GTIE=1)
0	0	触发输入无效	低电平
0	1	外部触发(上升沿)	高电平
1	0	外部触发(下降沿)	低电平
1	1	外部触发(双沿)	高电平

定时器控制寄存器(TCR 低位字节)

bit	7	6	5	4	3	2	1	0
Field	T32	FUNC2	FUNC1	FUNC0	PLRT	MODS	ENCNT	START
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit7] T32: 32 位定时器选位

该位选择 32 位定时器功能

将 FUNC2, FUNC1, FUNC0 位设为"0b011"，选择重载定时器功能时，如果将 T32 位设为"1"，则进入 32 位定时器模式

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变(参考“4.32 位模式的操作”)

bit	说明
0	16 位定时器模式
1	32 位定时器模式

[bit6:4] FUNC2 ~ FUNC0: 定时器功能选择位

该位选择定时器功能。

将 FUNC2, FUNC1, FUNC0 位设为"0b011"时，选择重载定时器功能。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变。

bit6	bit5	bit4	说明
0	0	0	复位模式
0	0	1	选择 PWM 功能

0	1	0	选择 PPG 功能
0	1	1	选择重载定时器功能
1	0	0	选择 PWC 功能
其他		禁止设定	

[bit3] PLRT: 输出极性指定位

该位选择定时器输出的正常电平还是取反电平。

通过与 bit2 MODS 的组合生成如下输出波形。

MODS	PLRT	输出波形
0	0	计数开始时输出 "L" 电平
0	1	计数开始时输出 "H" 电平
1	0	计数期间输出 "H" 电平方波
1	1	计数期间输出 "L" 电平方波

bit	说明
0	正常极性
1	取反极性

[bit2] MODS: 模式选择位

MODS 位设为 0"时，进入模式。计数值发生"0000"→"0xFFFF"的下溢的同时，重载寄存器值载入计数器并继续计数

MODS 位设为"1"时，进入单次模式。计数值发生"0x0000"→"0xFFFF"的下溢时，停止操作。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变。

bit	说明
0	重载模式
1	单次模式

[bit1] ENCNT: 定时器使能位

该位使能递减计数器的操作。

计数器处于操作使能状态(ENCNT 位="1")时，如果向其写"0"，则计数器停止。

bit	说明
0	停止
1	使能操作

[bit0] START: 软件触发位

ENCNT 位为"1"时，如果向 START 位写"1"，则软件触发有效。

START 位始终读"0"。

注意：

- 即使同时向 ENCNT 位和 START 位写"1"，软件触发仍然有效。
- 与 EDGS1, EDGS0 的设定无关，如果将 START 位设为"1"，则软件触发有效。

bit	说明
0	无效
1	软件启动开始

定时器控制寄存器 2 (TCR2 高位字节)

bit	15	14	13	12	11	10	9	8
Field	GTIE			res				CLKS3
属性	R/W			R/W				R/W
初始	0			0b000000				0

注意：

- 该寄存器配置到 STA 寄存器高位

[bit15] GTIE:门控功能使能位

此位选择外部因素引脚的触发输入功能或门控功能。

触发输入功能：当从外部因素引脚输入有效边沿时，递减计数开始。

门控功能：当从外部因素引脚输入有效边沿时，递减计数继续。

位	说明
0	触发输入功能
1	门控功能

[bit14:9] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit8] CLKS3:计数时钟选择位

参考"8.3.3.1 定时器控制寄存器(TCR 高位字节)"的"计数时钟选择位"

状态控制寄存器(STA)

bit	7	6	5	4	3	2	1	0
Field	res	TGIE	res	UDFIE	res	TRGIR	res	UDFIR
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

注意：

- 该寄存器的高位字节配置给 TCR2 寄存器。

[bit7] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit6] TGIE: 触发中断请求使能位

控制 bit2 TRGIR 的中断请求。

如果使能 TGIE 位后设定 bit2 TRGIR 位，则 CPU 发生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit5] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit4] UDFIE: 下溢中断请求使能位

控制 bit0 UDFIR 的中断请求。

如果使能 UDFIE 位后设定 bit0 UDFIR 位，则 CPU 发生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit3] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit2] TRGIR: 触发中断请求位

软件触发或检出触发输入时，TRGIR 位设为"1"。

向 TRGIR 位写"0"将其清零。

即使向 TRGIR 位写"1"也不影响位值。

读-修改-写类指令始终读"1"。

bit	说明
0	清除中断源
1	检出中断源

[bit1] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit0] UDFIR: 下溢中断请求位

从"H"脉宽设定值开始计数，计数期间，计数值发生"0x0000"→"0xFFFF"的下溢时，UDFIR 位设为"1"。

向 UDFIR 位写"0"将其清零。

即使向 UDFIR 位写"1"也不影响位值。

读-修改-写类指令始终读"1"。

bit	说明
0	清除中断源
1	检出中断源

25.8.3.4 周期设定寄存器(CSR)

周期设定寄存器(CSR)用于保持计数初始值。32 位模式下，偶数通道存储低 16 位的计数初始值。奇数通道存储高 16 位的计数初始值。复位后的初始值不定。务必通过 16 位数据传输命令访问寄存器。

bit	15		0
Field		CSR[15:0]	
属性		R/W	
初始值		0xXXXX	

该寄存器用于设定周期。下溢时向定时器寄存器传输数据。

按 16 位数据访问 CSR 寄存器。通过 TCR 寄存器的 FUNC2, FUNC1, FUNC0 位设定重载定时器功能后，CSR 寄存器设定周期。

32 位模式下，向 CSR 寄存器写数据时，先访问高 16 位数据(奇数通道的数据)，再访问低 16 位的数据(偶数通道的数据)。

25.8.3.5 定时器寄存器(TCNT)

定时器寄存器(TCNT)可读出定时器的计数值。32 位模式下，偶数通道的读出值为低 16 位的计数值，奇数通道读出值为高 16 位的计数值。该寄存器初始值不定。

务必通过 16 位数据传输命令读取该寄存器。

bit	15		0
Field		TCNT[15:0]	
属性		R 0xXXXX	
初始值			

读取 16 位递减计数器的值

按 16 位数据访问 TCNT 寄存器。

32 位模式下读取 TCNT 寄存器时，先读低 16 位数据(偶数通道的数据)，再读高 16 位数据(奇数通道的数据)。

25.8.4 PWC 定时器功能

复合定时器通过设定定时器控制寄存器的 FUNC2, 1, 0 位，可从 16 位 PWM 定时器，16 位 PPG 定时器，16/32 位重载定时器，16/32 位 PWC 定时器中选择一种定时器功能。

本节介绍设定 PWC 时的定时器功能。

1. PWC 定时器的操作
2. 选择 PWC 定时器时的定时器控制寄存器(TCR, TCR2), 状态控制寄存器(STA)
3. 数据缓冲寄存器(DBUF)

25.8.4.1 PWC 定时器的操作

PWC 定时器具有脉宽测定功能。可选择 5 种计数时钟，通过计数器测定输入脉冲的任意事件的时间间隔和周期。本节介绍脉宽测定功能中的基本功能和操作。

脉宽测定功能

启动后，将计数器设为"0x0000"，输入设定后的测定起始沿前，不执行计数操作。如果检出测定起始沿，"0x0001"开始递增计数，如果检出测定完成沿，则停止计数。这期间的计数值作为脉宽保存到寄存器。

测定完成时或上溢发生时，可产生中断请求。测定完成后，根据测定模式如下操作：

- 单次测定模式时，停止操作
- 连续测定模式时，计数器值传输到缓冲寄存器后，停止计数，直到再次输入测定起始沿

图 25.8.4-18 脉宽测定操作(单次测定模式/"H"脉宽测定)

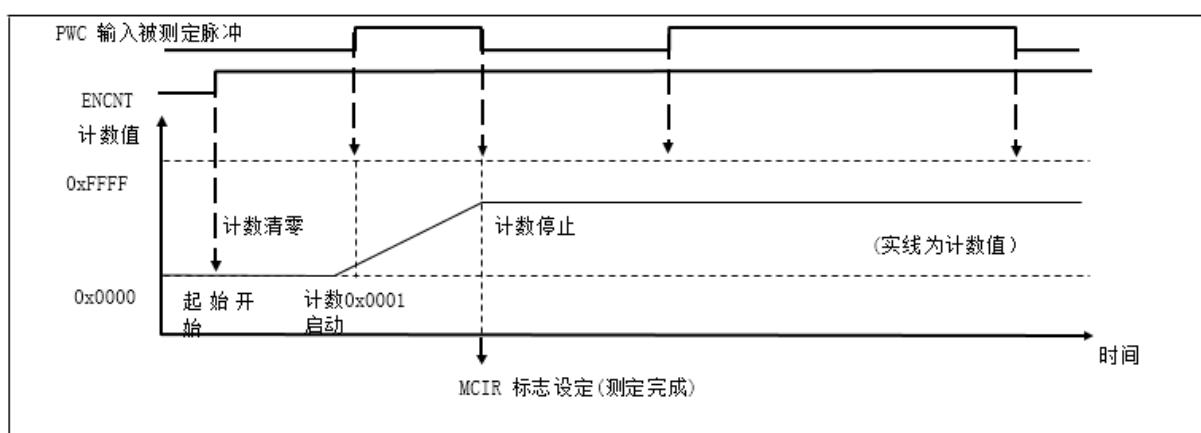
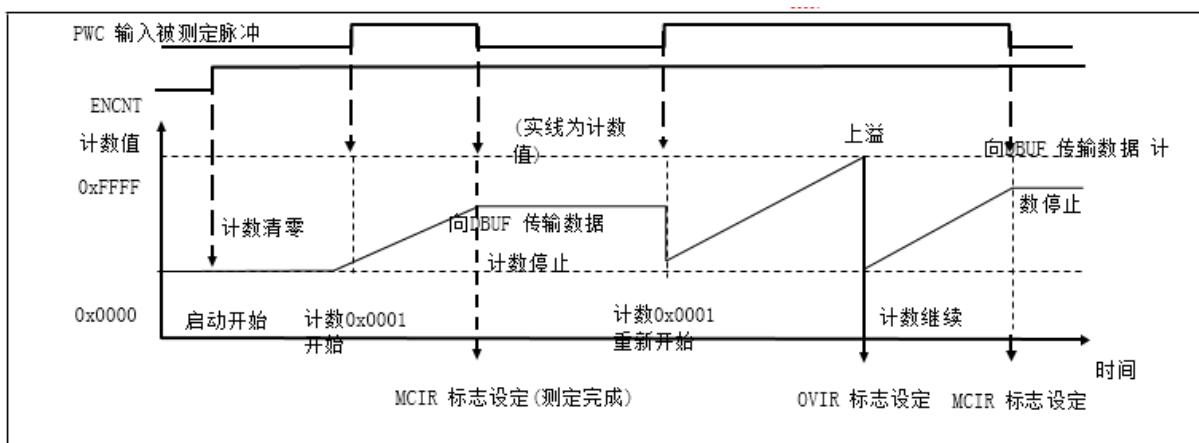


图 25.8.4-19 脉宽测定操作(连续测定模式/"H"脉宽测定)



计数时钟的选择

通过设定 TCR2 寄存器 bit8: CLKS3, TCR 寄存器 bit14 ~ 12:CLKS2, CLKS1, CLKS0, 可选择 8 种计数器的计数时钟。

以下是可选择的计数时钟：

TCR2, TCR 寄存器	选择的内部计数时钟
CLKS3, CLKS2, CLKS1, CLKS0 位	
0b0000	机器时钟(初始值)
0b0001	4 分频的机器时钟
0b0010	16 分频的机器时钟
0b0011	128 分频的机器时钟
0b0100	256 分频的机器时钟
0b0101	禁止设定
0b0110	
0b0111	
0b1000	512 分频的机器时钟
0b1001	1024 分频的机器时钟
0b1010	2048 分频的机器时钟
其他	禁止设定

复位后的初始值为选择机器时钟。

务必在启动计数器前选择计数时钟。

工作模式的选择

设定 TCR 来选择各工作模式/测定模式。

工作模式的设定 ... TCR bit10 ~ 8: EDGS2, EDGS1, EDGS0 (选择测定沿)

测定模式的设定 ... TCR bit2: MODS (选择单次测定/连续测定)

以下是工作模式选择一览表：

工作模式		MODS	EDGS2	EDGS1	EDGS0
$\uparrow \sim \downarrow$ H 脉宽测定	连续测定:缓冲器有效	0	0	0	0
	单次测定:缓冲器无效	1	0	0	0
$\uparrow \sim \uparrow$ 上升间周期测定	连续测定:缓冲器有效	0	0	0	1
	单次测定:缓冲器无效	1	0	0	1
$\downarrow \sim \downarrow$ 下降间周期测定	连续测定:缓冲器有效	0	0	1	0
	单次测定:缓冲器无效	1	0	1	0
$\uparrow \text{或} \downarrow \sim \uparrow \text{或} \downarrow$ 全部沿测定	连续测定:缓冲器有效	0	1	1	1
	单次测定:缓冲器无效	1	1	1	1
$\downarrow \sim \uparrow$ L 脉宽测定	连续测定:缓冲器有效	0	1	0	0
	单次测定:缓冲器无效	1	1	0	0
禁止设定		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

复位后的初始值为单次测定模式下选择"H"脉宽测定。

务必在启动计数器前选择工作模式。

脉宽测定的启动/停止

关于各操作的启动/重启/强行停止，应设定 TCR 的 bit1:ENCNT 位。

向 ENCNT 位写"1"后，脉宽测定的启动/重启有效，向 ENCNT 位写"0"后，强行停止有效。

ENCNT	功能
1	脉宽测定的启动/重启
0	脉宽测定的强行停止

启动后的操作

脉宽测定模式启动后，测定起始沿输入前，不执行计数。检出测定起始沿后，16 位递增计数器从"0x0001"开始计数。

重启

启动后，工作期间再次启动(ENCNT 位为"1"的状态下再次写"1")被认为为是重启。重启动时，执行以下操作：

- 在等待测定起始沿的到来时：不影响操作。
- 测定期间：计数清零为"0x0000"后，再次进入等待测定起始沿的状态。这时，如果测定完成沿检测和重启同时执行，则设定测定完成标志(MCIR)，连续测定模式时，测定结果传

输到 DBUF。

停止

单次测定模式下，计数器发生上溢或测定完成时，自动停止计数操作，因此无需特别留意。

如果想在连续测定模式或需要在其自动停止前使其停止，则需要强行停止。

计数器清零和初始值

16 位递增计数器在以下情况下清零为"0x0000":

- 复位时
- 向 TCR 的 bit1: ENCNT 位写"1"时(包含重启时的情况)

16 位递增计数器在以下情况下初始化为"0x0001":

- 检出测定起始沿时

脉宽测定操作

● 单次测定和连续测定

脉宽测定包含单次模式和连续模式。各模式由 TCR 的 MODS 位选择(参考"8.4.1.3 工作模式的选择")。以下是两种模式的区别:

单次测定模式:

当第 1 次的测定完成沿到来时，计数器停止，STA 中的测定完成标志(MCIR)设定后，不再进行测定。但是，如果在此时重启，则会进入等待测定起始的状态。

连续测定模式:

当输入测定完成沿到来时，计数器停止计数，STA 中的测定完成标志(MCIR)被设定，再次输入测定起始沿前，停止计数。如果再次输入测定起始沿，则将计数器初始化为"0x0001"后开始测定。测定完成时，计数器的测定结果传输到 DBUF。

务必在计数器停止期间选择/改变测定模式。

● 测定结果数据

单次测定模式和连续测定模式下，测定结果和计数器值的处理以及 DBUF 的功能存在差异。两种模式中测定结果的区别如下:

单次测定模式:

如果工作期间读 DBUF，则得到当前的测量计数值。

如果测定完成后读 DBUF，则得到测定结果数据。

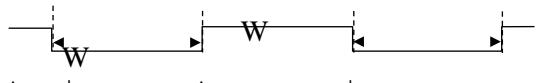
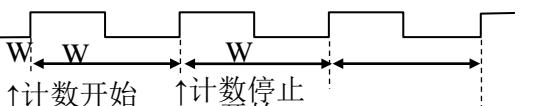
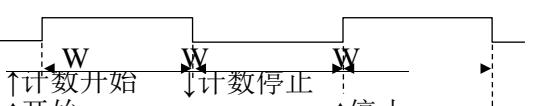
连续测定模式:

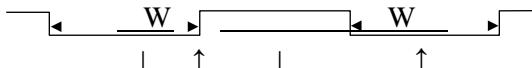
测定完成时，计数器内的测定结果传输到 DBUF。此时读 DBUF，可获取最后一次的测定结果。之前的测量结果可以在测量过程中获取。但是测定期间的计数值无法被读取。

连续测定模式下，尚未读取测定结果但下次测定已经完成时，上次的测定结果被新的测定结果代替。这时，STA 中的错误标志(ERR)置位。一旦读取 DBUF，则错误标志(ERR)清零。

测定模式和计数操作

根据输入脉冲的测定位置可选择 5 种测定模式，如下所示：

测定模式	EDGS2, 1, 0	测定内容(W:测定脉宽)
"H"脉宽测定	0b000	 <p>↑ ↓ 计数停止 ↑ 计数开始 ↓ 停止 开始</p> <p>测定"H"电平期间的脉宽。 计数(测定)开始:检出上升沿时 计数(测定)完成:检出下降沿时</p>
上升沿间周期测定	0b001	 <p>↑ 计数开始 ↑ 计数停止 ↑ 开始</p> <p>测定上升沿间的周期。 计数(测定)开始:检出上升沿时 计数(测定)完成:检出上升沿时</p>
下降沿间周期测定	0b010	 <p>↓ 停止 ↓ 开始 ↓ 开始</p> <p>测定下降沿间的周期。 计数(测定)开始:检出下降沿时 计数(测定)完成:检出下降沿时</p>
全部沿间脉宽测定	0b011	 <p>↑ 计数开始 ↓ 计数停止 ↑ 停止</p> <p>↑ 开始 ↑ 开始</p> <p>测定连续输入沿间的宽度。 计数(测定)开始:检出边沿时 计数(测定)完成:检出边沿时</p>

"L"脉宽测定	0b100	 测定"L"电平期间的宽度。 计数(测定)开始:检出下降沿时 计数(测定)完成:检出上升沿时
---------	-------	---

无论哪种测定模式，测定启动时计数器清零"0x0000"后，输入测定起始沿前，计数器不计数。

从输入测定起始沿开始，到输入测定完成沿的这段时间，各计数时钟连续递增计数。

连续测定模式下，执行所有的沿间脉宽测定和周期测定时，最后的完成沿同时也是下一个测定的起始沿。

脉宽/周期计算方法

测定完成后，可按以下方法从 DBUF 得到的测定结果数据算出被测定脉宽/周期。

$$TW = n \times t$$

 TW : 被测定脉宽/周期

 n : DBUF 内的测定结果数据

 t : 计数时钟周期

中断请求发生

可产生 2 个中断请求：

- 计数器上溢中断请求

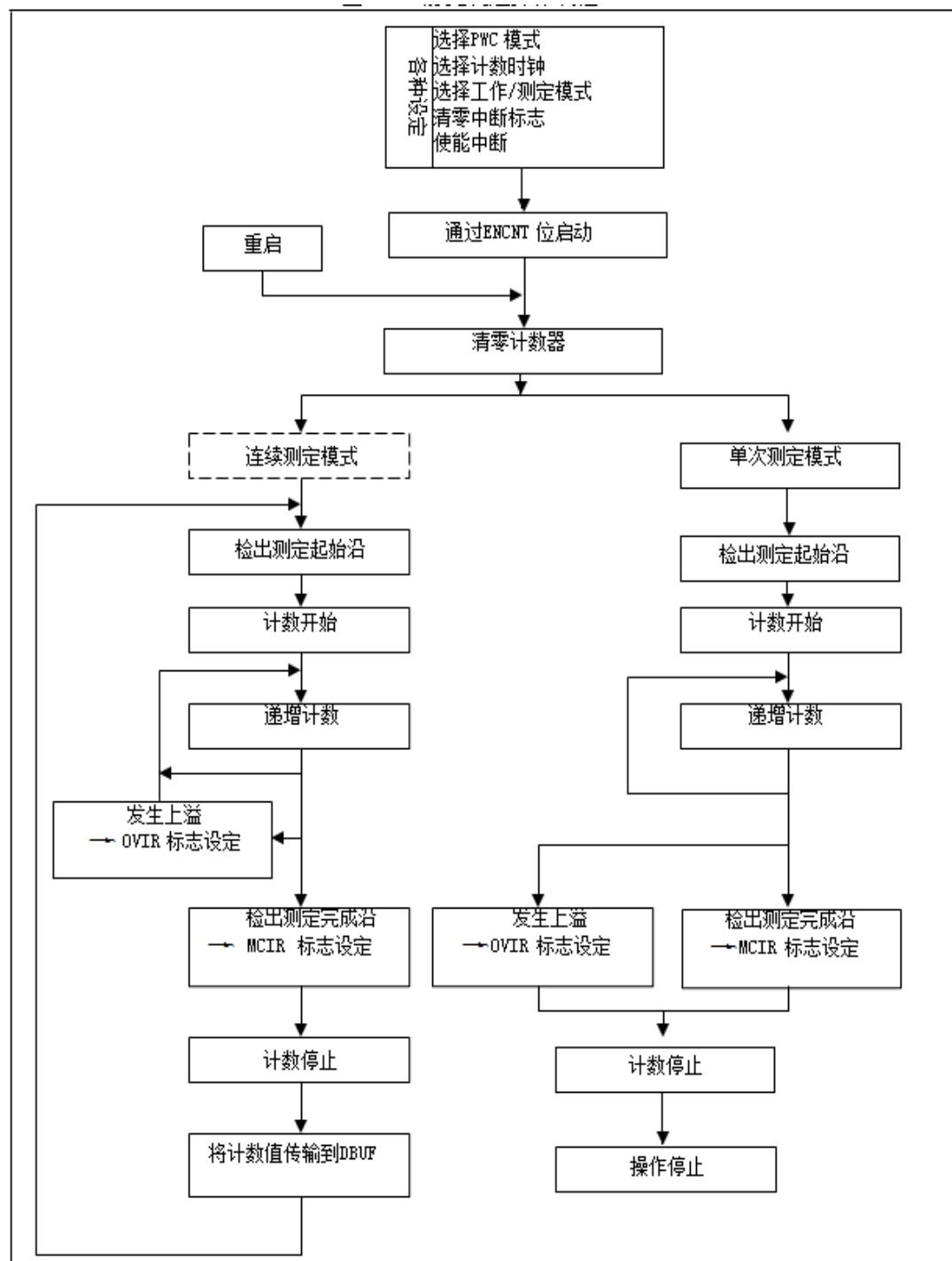
测定期间，如果计数递增引起上溢，则上溢标志(OVIR)置位，如果使能上溢中断请求，则发生中断请求。

- 测定完成中断请求

如果检出测定完成沿，则 STA 中的测定完成标志(MCIR)置位，如果使能测定完成中断请求，则产生中断请求。

读取测定结果 DBUF 后，测定完成标志(MCIR)自动清零。

脉宽测定操作流程



25.8.4.2 选择 PWC 定时器时的定时器控制寄存器(TCR, TCR2),状态控制寄存器(STA)

定时器控制寄存器(TCR)控制定时器操作。

定时器控制寄存器(TCR 高位字节)

bit	15	14	13	12	11	10	9	8
Field	res	CLKS2	CLKS1	CLKS0	res	EDGS2	EDGS1	EDGS
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit15] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit14:12, TCR2:bit8] CLKS3 ~ CLKS0: 计数时钟选择位

选择 16 位递减计数器的计数时钟。

更改设定后, 计数时钟立刻得以改变。因此, 应在计数停止状态(ENCNT="0")更改 CLKS3 ~ CLKS0。但可向 ENCNT 位写"1"的同时进行改变。

CLKS3	CLKS2	CLKS1	CLKS0	说明
0	0	0	0	φ
0	0	0	1	$\varphi/4$
0	0	1	0	$\varphi/16$
0	0	1	1	$\varphi/128$
0	1	0	0	$\varphi/256$
0	1	0	1	禁止设定
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	$\varphi/512$
1	0	1	0	$\varphi/1024$
其它				禁止设定

[bit11] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit10:8] EDGS2 ~ EDGS0: 测定沿选择位

设定测定沿的条件。

应在计数停止状态(ENCNT="0")下更改 EDGS2, EDGS1, EDGS0。但可向 ENCNT 位写"1"的同时进行改变。

bit10	bit9	bit8	说明
0	0	0	"H"脉宽测定($\uparrow \sim \downarrow$)
0	0	1	上升沿间周期测定($\uparrow \sim \uparrow$)

0	1	0	下降沿间周期测定(↓ ~ ↓)
0	1	1	全部沿间脉宽测定(↑或↓ ~ ↓或↑)
1	0	0	"L"脉宽测定(↓ ~ ↑)
1	0	1	禁止设定
1	1	0	
1	1	1	

定时器控制寄存器(TCR 低位字节)

bit	7	6	5	4	3	2	1	0
Field	T32	FUNC2	FUNC1	FUNC0	res	MODS	ENCNT	res
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit7] T32: 32 位定时器选择位

该位选择 32 位定时器功能。

将 FUNC2, FUNC1, FUNC0 位设为"0b100"后，选择 PWC 功能时，如果将 T32 位设为"1"，则进入 32 位 PWC 模式。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变(参考 "4.32 位模式的操作")。

bit	说明
0	16 位定时器模式
1	32 位定时器模式

[bit6:4] FUNC2 ~ FUNC0: 定时器功能选择位

选择定时器功能。

将 FUNC2, FUNC1, FUNC0 位设为"0b100"时，选择 PWC 定时器功能。

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变。

bit6	bit5	bit4	说明
0	0	0	复位模式
0	0	1	选择PWM 功能
0	1	0	选择PPG 功能
0	1	1	选择重载定时器功能
1	0	0	选择PWC 功能
1	0	1	禁止设定
1	1	0	
1	1	1	

[bit3] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit2] MODS: 模式选择位

应在定时器停止期间(ENCNT="0")进行改变。但可向 ENCNT 位写"1"的同时进行改变。

bit	说明
0	连续测定模式(缓冲寄存器有效)
1	单次测定模式(1 次测定后停止)

[bit1] ENCNT: 定时器使能位

该位使能递增计数器的启动或重启。

计数器为操作使能状态(ENCNT 位="1")时, 向该位写"1"会重启, 同时计数器清零, 等待下一个测定起始沿。

计数器为操作使能状态(ENCNT 位="1")时, 向其写"0", 则计数器停止。

bit	说明
0	停止
1	使能操作

bit0] res: 保留位读值为"0"。写寄存器时将该位写"0"。

定时器控制寄存器 2 (TCR2 高位字节)

bit	15	14	13	12	11	10	9	8
Field				res				CLKS3
属性				R/W				R/W
初始值				0b0000000				0

注意:

- 该寄存器配置到 STA 寄存器的高位。

[bit15:9] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit8] CLKS3: 计数时钟选择位

参考"8.4.2.1 定时器控制寄存器(TCR 高位字节) "的"计数时钟选择位"。

状态控制寄存器(STA)

bit	7	6	5	4	3	2	1	0
Field	ERR	MCIE	res	OVIE	res	MCIR	res	OVIR
属性	R	R/W	R/W	R/W	R/W	R	R/W	R/W
初始值	0	0	0	0	0	0	0	0

注意：

- 该寄存器的高位字节配置给 TCR2 寄存器。

[bit7] ERR: 错误标志位

连续测定模式时，未读取 DBUF 寄存器的测定结果期间，该标志指示下次测定已经完成。这时，DBUF 寄存器值更新为新的测定结果，前一次的测定结果会丢失。

无论 ERR 位的值为多少，测定继续。

ERR 位为只读，写操作无效。

读取测定结果(DBUF)后，ERR 位被清零。

bit	说明
0	正常状态
1	下次测定结果覆盖未读取的测定结果

[bit6] MCIE: 测定完成中断请求使能位

该位控制 bit2 MCIR 的中断请求。

如果使能 MCIE 位后设定 bit2 MCIR 位，则 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit5] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit4] OVIE: 上溢中断请求使能位

该位控制 bit0 OVIR 的中断请求。

如果使能 OVIE 位后设定 bit0 OVIR 位，则 CPU 产生中断请求。

bit	说明
0	禁止中断请求
1	使能中断请求

[bit3] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit2] MCIR: 测定完成中断请求位

该位标志测定完成，测定完成时标志为"1"。

读取测定结果(DBUF)可将 MCIR 位清零。

MCIR 位为只读，写操作无效。

Bit	说明
0	读取测定结果(DBUF)
1	检出中断源

[bit1] res: 保留位读值为"0"。写寄存器时将该位写"0"。

[bit0] OVIR: 上溢中断请求位

计数值发生"0xFFFF"→"0x0000"的上溢时，标志为"1"。

向 OVIR 位写"0"将其清零。

即使向 OVIR 位写"1"也不影响位值。

读-修改-写类指令始终读"1"。

bit	说明
0	清除中断源
1	检出中断源

25.8.4.3 数据缓冲寄存器(DBUF)

数据缓冲寄存器(DBUF)可读取 PWC 定时器的测定值或计数值。32 位模式下，偶数通道为低 16 位的值，奇数通道为高 16 位的值。

必须通过 16 位数据传输指令读取寄存器。

bit	15		0
Field		DBUF[15:0]	
属性		R	
初始值		0x0000	

在连续测定模式或单次测定模式下，DBUF 寄存器都为只读。写操作无效。

连续测定模式时(TCR bit3 MODS="1")，该寄存器是保持上次测定结果的缓冲寄存器。

单次测定模式时(TCR bit3 MODS="0")，可通过 DBUF 寄存器直接访问递增计数器。计数期间仍可读取计数值。测定完成后，原样保存测定结果。

按 16 位数据访问 DBUF 寄存器。

26. 蜂鸣器

26.1 概述

本节讲述蜂鸣器的概述。

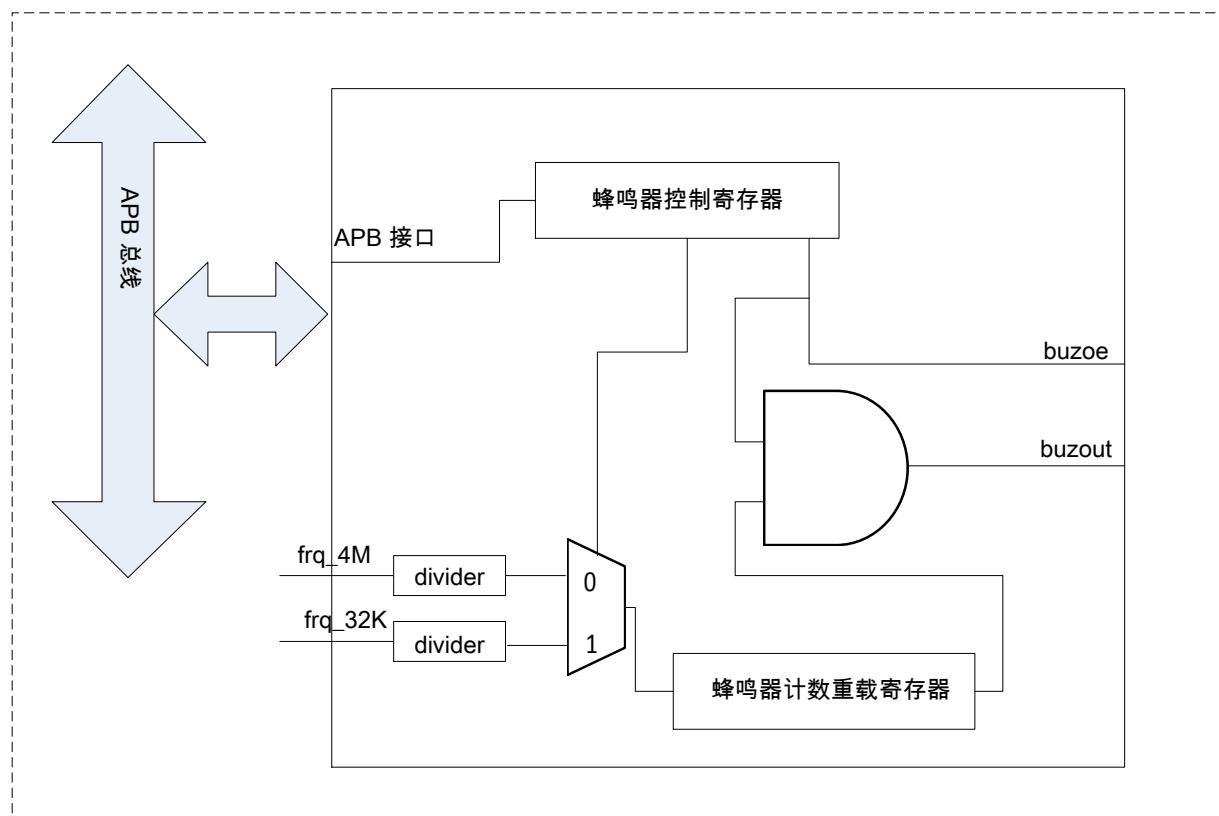
蜂鸣器产生特定频率的方波，在有蜂鸣器的系统中，用来作为确认声响。它有下面这些特点：

- 频率可调
- 时钟源可选(内部 HICR(freq_4M) 和 LICR(freq_32K))

26.2 配置框图

本节显示蜂鸣器的内部框图。

图 26.2-1 蜂鸣器内部结构



buzout: 蜂鸣器输出；

buzoe: 蜂鸣器输出使能。

26.3 操作

本节讲述蜂鸣器的配置操作。

蜂鸣器输出选择

- 控制寄存器 / 逻辑

这个电路控制蜂鸣器的输出频率。蜂鸣器的输出频率可以通过下面的公式得到：

$$\text{蜂鸣器输出} = \frac{\text{时钟源}}{2 * (256 - \text{CNT})}$$

- 蜂鸣器控制寄存器

设置这个寄存器，选择计数时钟源。内部 32K 的 4 分频，内部 HICR (2/4/8/16M) 的 512 分频都可以选做时钟源。

触发蜂鸣器输出，设置寄存器 BUZ_ST 位。

- 蜂鸣器计数重载寄存器

这个寄存器用做存放蜂鸣器计数器的初始值 CNT。当溢出发生时，这个值将作为蜂鸣器计数器的初值，重新被加载。

下表显示蜂鸣器的输出频率。

蜂鸣器输出频率与计数器初值 CNT 关系

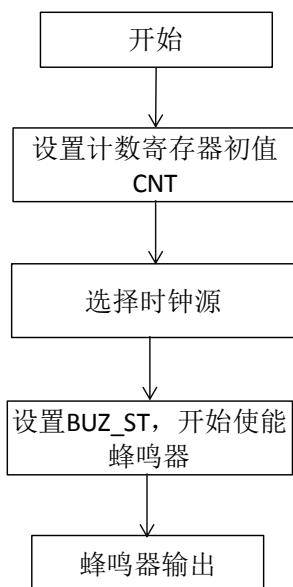
表 26.3-1 蜂鸣器输出频率与计数器初值 CNT 关系

外部时钟	时钟源	buzzout (不同CNT值)							
		FFh	FEh	FDh	FCh	FBh	FAh	00h
32K	8K	4K	2K	1.33K	1K	800	667	...	15.6
2M	4K	2K	1K	667	512	400	333	...	7.8
4M	8K	4K	2K	1.33K	1K	800	667	...	15.6
8M	16K	8K	4K	2.66K	2K	1.6K	1.33k	...	31
16M	32K	16K	8K	5.33K	4K	3.2K	2.66k	...	62

26.4 设置程序实例

本节讲述设置流程实例。

图 26.4-1 蜂鸣器设定程序



26.5 寄存器

本节讲述蜂鸣器寄存器。

表 26.5 -1 蜂鸣器寄存器列表

缩写	寄存器名 6	参看
BUZ_CTL	蜂鸣器输出控制寄存器	26.5.1
BUZ_CNT	蜂鸣器计数重载寄存器	26.5.2

26.5.1 蜂鸣器控制寄存器(BUZ_CTL)

蜂鸣器控制寄存器控制蜂鸣器的输出。

寄存器配置

位	7		2	1	0
字段		Reserved		BUZ_ST	CLK_SEL
属性		-		R/W	R/W
初值		-		0	0

寄存器功能

[bit7:2] Reserved: 预留位, 读出值为 5"0b000"

写时请设置为 "0b000".

[bit1]BUZ_ST: 蜂鸣器开始位

位		注释
读		返回寄存器值。
写	0	蜂鸣器不开始。
写	1	蜂鸣器开始。

[bit0]CLK_SEL: 蜂鸣器时钟选择位

位		注释
读		返回寄存器值
写	0	内部 HICR 4M 的 512 分频作为时钟源。
写	1	内部 LICR 的 4 分频作为蜂鸣器时钟源。

注意:

- 当设置了 BUZ_ST 位后, 蜂鸣器自动输出方波。

26.5.2 蜂鸣器计数重载寄存器(BUZ_CNT)

蜂鸣器计数重载寄存器，计算计数的周期，在溢出之后，重新装载计数初值。

寄存器配置

位	7	0
字段	BUZ_CNT	
属性	R/W	
初值	0x00	

寄存器功能

[bit7:0] 蜂鸣器计数重载寄存器

Bit7:0	描述
读	蜂鸣器当前计数值。
写	将初始值写入蜂鸣器计数器，在计数溢出之后，自动重新装载。

注意：

- 在蜂鸣器工作时写本寄存器，这个值在下一次溢出之后有效。

26.6 使用注意事项

本小节讲述使用蜂鸣器时的注意事项。

蜂鸣器计数重载寄存器能够在操作中改动。设定程序只是一个例子。用户可以照着做，或者自己根据情况改动。

27. 带隙基准源(BGR)

本章介绍带隙基准源(BGR)的功能以及操作方式。

27.1 BGR 的概要

以下介绍了 BGR 的概要。

BGR 的功能

BGR 是一个参考电压生成电路，主要用于给 A/D 转换器和电压比较器输出参考电压。

同时，在使用低电压监测电路(LVD)，电压比较器，外部高速晶振以及 A/D 转换器时，必须使能 BGR。

关于 BGR 具体使用方式，请参考“时钟”，“低电压检测”，“12 位 A/D 转换器”以及“电压比较器”。

27.2 寄存器

以下介绍 BGR 的寄存器功能。

27.2.1 寄存器列表

以下列出了 BGR 的控制寄存器。

表 27.2.1-1 BGR 控制寄存器列表

寄存器简称	寄存器名称	参照章节
BGR_CTL	BGR 控制寄存器	27.2.2

27.2.2 BGR 控制寄存器 (BGR_CTL)

BGR 控制寄存器(BGR_CTL) 控制 BGR 的设定。

寄存器配置

位	31								8
字段名	保留								
属性	-								
初始值	-								
位	7	6	5	4	3	2	1	0	
字段名	保留							BGR_EN	
属性	-							R/W	
初始值	-							0	

寄存器功能

[31:1] 保留位，读值未定，写操作无效。

[0] BGR_EN: BGR 使能位，本位控制 BGR 的使能。

Bit	功能
0	不使能 BGR。 (初始值)
1	使能 BGR。

注意：

- 该寄存器仅被上电复位清零。
- 使能 BGR 后，需要等待建立时间，具体建立时间请参考数据手册。

28. 12 位 A/D 转换器

本章介绍了 12 位 A/D 转换器的功能和操作。

28.1 概要

12 位 A/D 转换器的功能是按照 RC 逐次逼近法将模拟输入电压转换为 12 位数值。

12 位 A/D 转换器的特征:

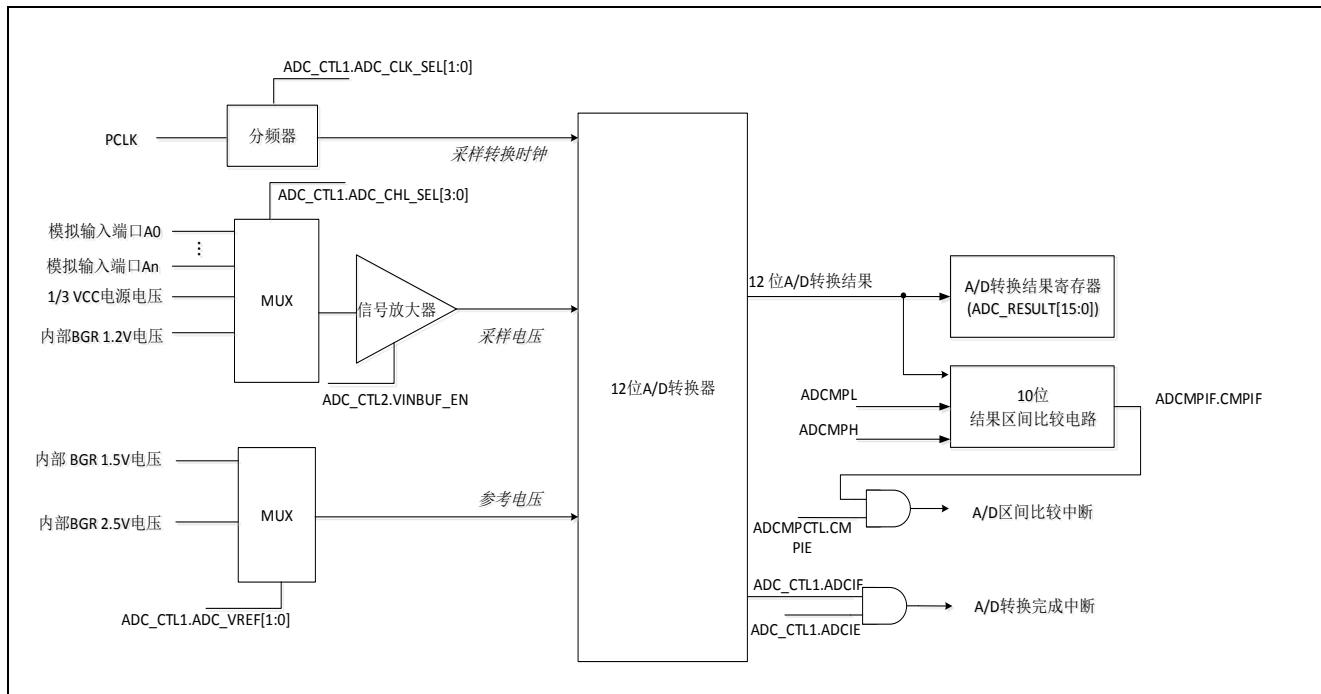
- 单调无失码的 12 位分辨率
- 使用采样保持电路的 RC 逐次逼近法进行比较转换
- 可通过软件选择内置参考电压 (1.5 V 或 2.5 V)
- 单独可配置外部端口模拟电压输入通道
- 支持采样转换电源电压以及内部参考电压
- 可选转换时钟源
- 支持单次采样模式和连续采样模式
- 可变更 A/D 转换数据的配置 (可选择向 MSB 端移位或者向 LSB 端移位)
- 内置输入信号放大器可采样外部微弱信号
- A/D 区间比较功能
- 转换结果的高 10 位(位 11:2)进行比较
- 可配置上限/下限数据
- 可选择在区间范围内或者范围外检测转换结果
- 当选择检测结果在区间范围外时，可识别“超出上限值”或者是“低于下限值”
- 可选择区间检测次数
- 有以下两种中断源:
 - A/D 转换完成中断
 - A/D 区间比较中断

28.2 架构框图

以下描绘了 12 位 A/D 转换器的框图。

12 位 A/D 转换架构框图

图 28.2-1 12 位 A/D 转换器架构框图



28.3 操作说明

以下介绍了 12 位 A/D 转换器的功能及操作。

28.3.1 A/D 转换操作

A/D 转换器可进行两种转换：单次采样转换和连续采样转换。

28.3.1.1 操作概要

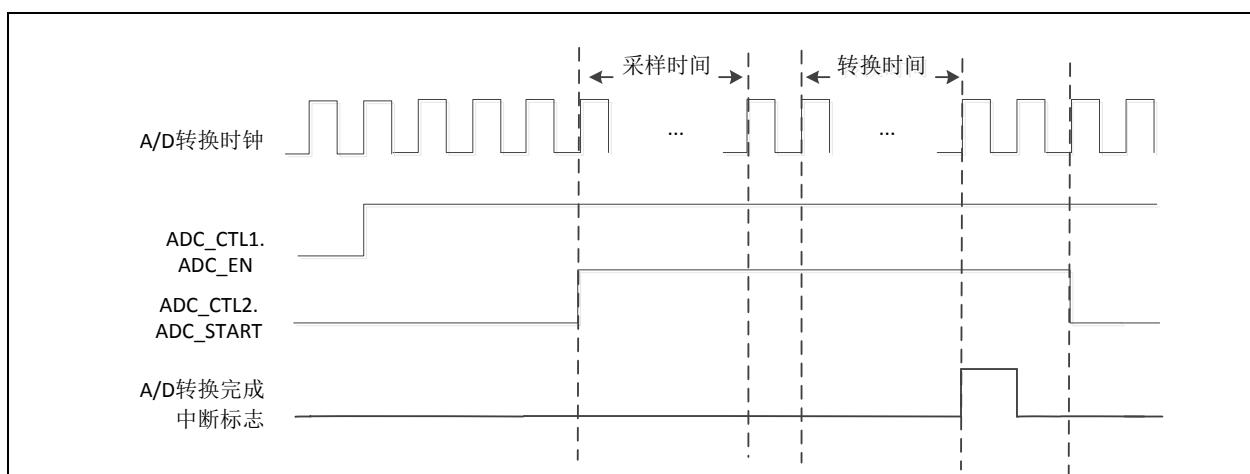
A/D 转换器可通过软件启动。使用软件启动时，设置 A/D 控制寄存器 1(ADC_CTL1)中的 ADC_EN 位后，再设置 A/D 控制寄存器 2(ADC_CTL2)中的 ADC_START 为 "1"，即开始转换。当转换结束后，转换完成标志位 ADCIF 置 "1"。

28.3.1.2 单次采样转换操作

以下介绍单次采样转换操作。

当 ADC_CTL2 寄存器中的 ADC_MODE = 1 时，单次采样转换模式有效。当转换完成时，A/D 操作停止，ADC_CTL2 中的 ADC_START 位会被硬件清零。

图 28.3-1 单次采样转换模式操作

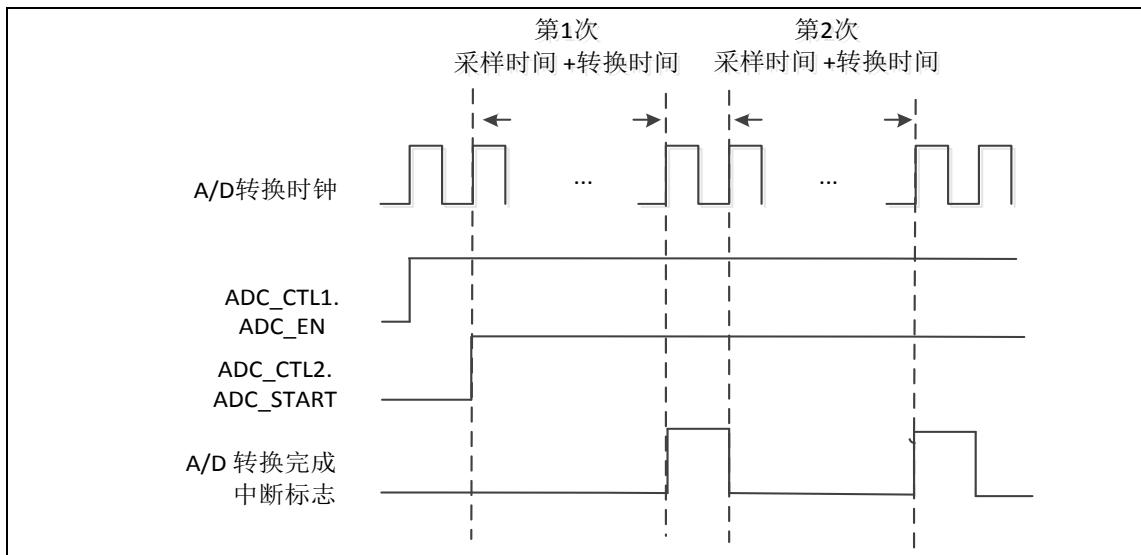


28.3.1.3 连续采样转换操作

以下介绍连续采样转换操作。

当 ADC_CTL2 寄存器中的 ADC_MODE = 0 时，连续采样转换模式有效。每次当转换完成时，A/D 转换操作继续，ADC_CTL2 中的 ADC_START 位不会被硬件清零。

图 28.3-2 连续采样转换模式操作



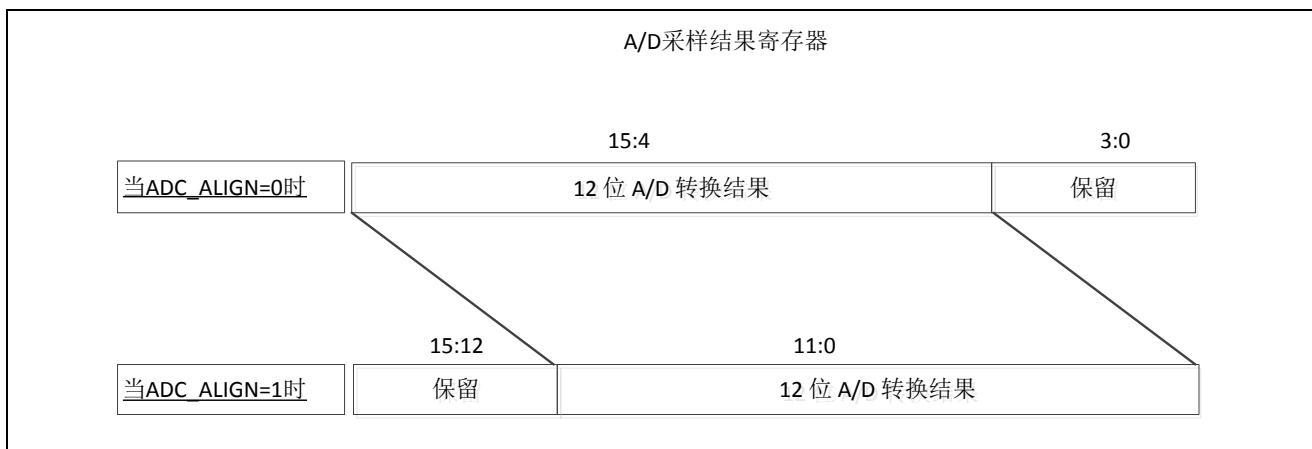
28.3.1.4 A/D 结果的配置选择

以下介绍了 A/D 结果的配置选择。

A/D 转换器可以通过配置 A/D 控制寄存器 2 (ADC_CTL2) 中的 ADC_ALIGN 位来改变转换结果的移位方向（图 28.3-3）

设置 ADC_ALIGN 位为"1"时，从 A/D 采样结果寄存器(ADC_RESULT)读取的 12 位 A/D 转换结果会移位到最右端。

图 28.3-3 A/D 结果配置



28.3.2 区间比较功能

区间比较功能可以判断 A/D 转换器的结果是否在已设定好的区间范围内且产生中断请求。

把 A/D 区间比较控制寄存器(ADCMPCCTL)中的 CMPEN 位置 1，就可以使能区间比较功能。

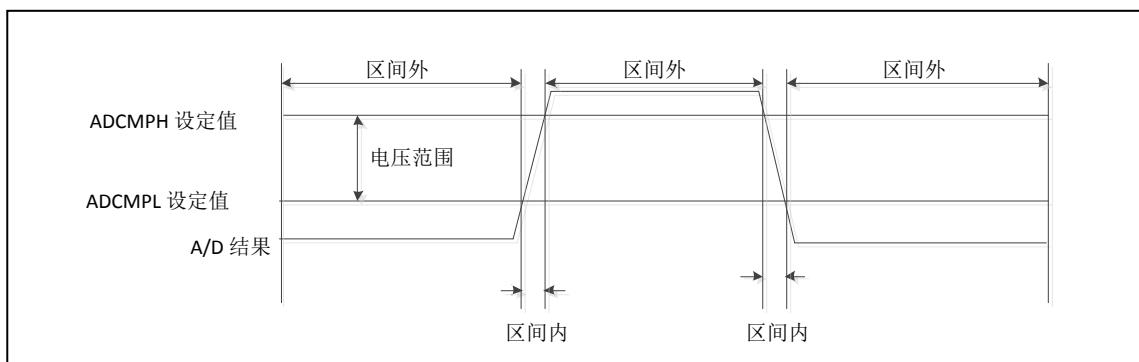
A/D 输出结果的高 10 位会跟 A/D 区间比较上限阈值寄存器(ADCMPH)和 A/D 区间比较下限阈值寄存器(ADCMPL)中设定的值进行比较，并且可以通过设置 ADCMPCCTL 中的 CMPTRG 位来选择检测在设定值的区间内还是区间外。

表 28.3-1 描述了区间比较的条件，图 28.3-4 描绘了区间比较的操作。

表 28.3-1 区间比较条件

区间比较结果	设定条件为检测区间外 (CMPTRG="0")	设定条件为检测区间内 (CMPTRG="1")
区间外 (超出上限阈值) A/D 结果 > 上限设定值	检测	不检测
区间内 A/D 结果 \geq 下限设定值 且 A/D 结果 \leq 上限设定值	不检测	检测
区间外 (低于下限阈值) A/D 结果 < 下限设定值	检测	不检测

图 28.3-4 区间比较的操作



连续检测功能可以连续的进行区间比较以此来减少噪声的干扰。当连续检测的次数满足 ADCMPCCTL 中的 CMPCNT[2:0]位设定的次数时，区间比较标准位 CMPIF 会被置 "1"。如果此时区间比较的中断使能位 CMPIE 是 "1"，就会产生中断请求。

在连续检测时，区间比较的结果只要有一次没有被检测到，那么连续检测次数会被清零并且重新开始计算检测次数。

关于连续检测条件，见表 28.3-2。

表 28.3-2 连续检测条件

项目	描述
连续检测操作	只要区间比较功能使能(CMPEN设置为1), 检测功能就会一直运作
连续检测次数	通过设置CMPCNT[2:0],连续检测次数可以设置范围为1次~7次 通过读取连续检测次数 (CMPCNT_ST[2:0], 可以确定当前满足条件的已检测的次数)
检测次数清零条件	禁止区间比较功能(CMPEN设置为0) 区间比较的结果没有被检测到
检测次数递增条件	区间比较的结果被检测到。 但是, 当检测的次数已经达到了CMPCNT[2:0]的设定值时, 检测会停止, 并且检测次数会跟CMPCNT[2:0]保持一致。

注意:

- 当检测条件设定为区间外 (ADCMPCTL.CMPTRG=0)时,即使区间比较的结果从超出上限值变为低于下限值, 连续检测次数仍然不会被清零并且会继续进行检测。
- 为了初始化连续检测区间比较结果的次数, 当不需要使用 A/D 转换功能时禁止区间比较功能, 然后再次使能区间比较功能。

当设定检测条件为区间外(ADCMPCTL.CMPTRG=0)时, 当前状态为超出上限值还是低于下限值可以从 CMPPF 位中读取。

关于 A/D 区间比较阈值范围外标志位的判断条件, 见表 28.3-3。

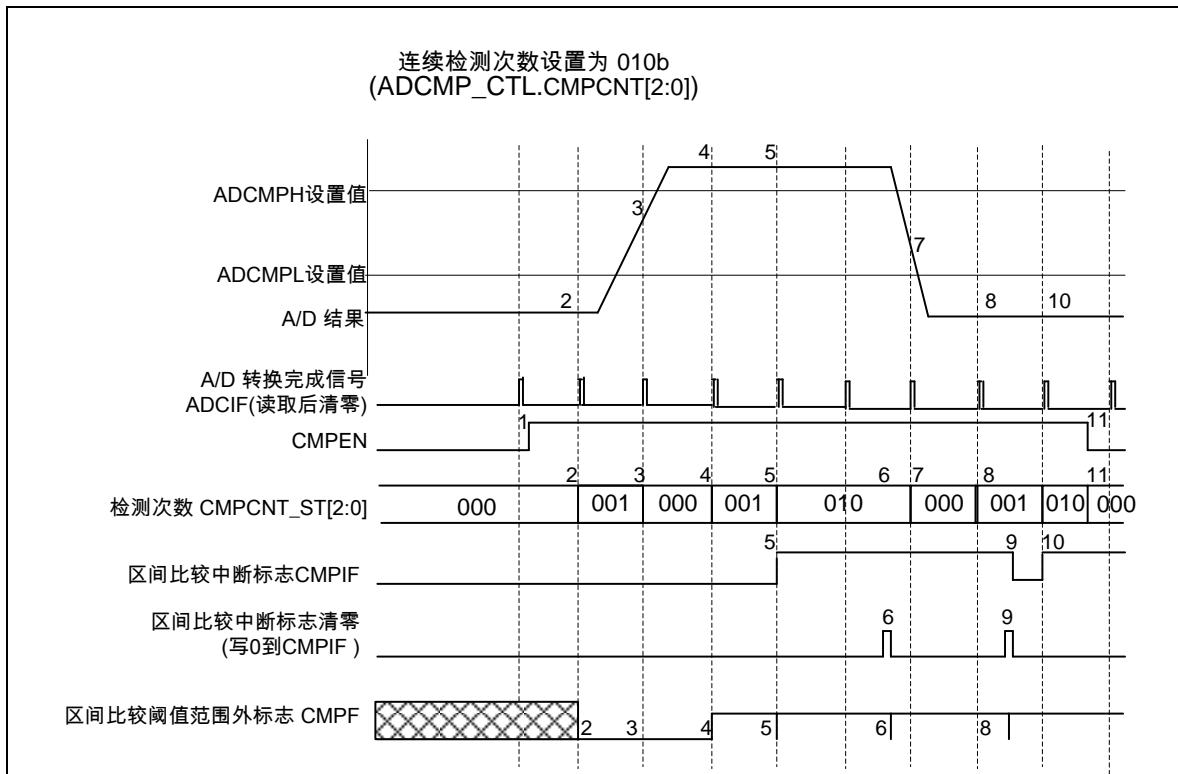
表 28.3-3 A/D 区间比较阈值范围外标志位判断条件

区间比较结果	区间比较阈值范围外标志位(CMPPF)	
	设定条件为检测区间外时 (CMPTRG="0")	设定条件为检测区间内时 (CMPTRG="1")
区间外 (超出上限阈值) A/D 结果 > 上限设定值	"1"	保持
区间内 A/D 结果 \geq 下限设定值 且 A/D 结果 \leq 上限设定值	保持	保持
区间外 (低于下限阈值) A/D 结果 < 下限设定值	"0"	保持

当区间比较中断请求标志位(CMPIF)为 1 的时候, 区间比较阈值范围外标志位(CMPPF) 的值会保持不变。

关于区间比较功能的操作示例(检测条件为区间范围外), 见图 28.3-5。

图 28.3-5 区间比较功能操作示例



以下是上图中区间比较功能的相关注解：

当区间比较功能被禁止($CMPEN=0$)时，连续检测次数($CMPCNT_ST$)被初始化到"000"。

当区间比较功能使能($CMPEN=1$)时，区间比较操作就会开始。

当区间比较的结果低于下限值时，连续检测次数($CMPCNT_ST$)开始递增。

同时，比较阈值范围外标志表明当前是低于下限值($CMPF="0"$)。

在连续检测次数 ($CMPCNT_ST$) 变为 "010" 之前，区间比较结果落在区间内没有被检测到。所以，连续检测次数($CMPCNT_ST$)被初始化到 "000"。

同时，区间阈值范围外标志位($CMPF$)的值保持。

因为区间比较结果超出了上限值，连续检测次数($CMPCNT_ST$)递增。

同时，比较阈值范围外标志表明当前是超出上限限值($CMPF="1"$)。

由于区间比较检测次数达到了设定值 ($CMPCNT_ST = "010"$)，区间比较中断请求标志位($CMPIF$)置 1。

同时，比较阈值范围外标志($CMPF$)的状态会从区间比较中断标志位($CMPIF$)置 1 的地方开始保持，直到区间比较中断标志位($CMPIF$)清零。

当连续检测的状态冲突时，连续检测条件满足状态引起的区间比较中断置位拥有比区间比较中断标志位清零更高的优先级。此时区间比较中断标志会置位($CMPIF = "1"$)并且比较阈值范围

外标志(CMPF)会再次表明相应状态。

当区间比较的结果在范围内时，尽管此时区间比较中断标志处于置位的状态(CMPIF="1")，连续检测次数仍然会被初始化(CMPCNT_ST = "000")。

而且尽管此时区间比较中断标志处于置位的状态(CMPIF = "1")，连续检测次数(CMPCNT_ST)依然会根据低于下限值的区间比较结果递增。但是处于区间比较中断标志处于置位的状态时(CMPIF = "1")，比较阈值范围外标志(CMPF)的值会保持。

写"0"到区间比较中断标志位会把区间比较中断标志(CMPIF = "0")清零。

同时，比较阈值范围外标志(CMPF)的保持状态也会被释放。

由于区间比较检测次数达到了设定值(CMPCNT_ST = "010")，区间比较中断请求标志位(CMPIF)置1。

同时，比较阈值范围外标志(CMPF)的状态会从区间比较中断标志位(CMPIF)置1的地方开始保持，直到区间比较中断标志位(CMPIF)清零。

当区间比较功能被禁止(CMPEN = "0")时，连续检测次数(CMPCNT_ST)被初始化到 "000"。

但是不管区间比较中断标志位(CMPIF)还是比较阈值范围外标志位(CMPF)都不会被清零。

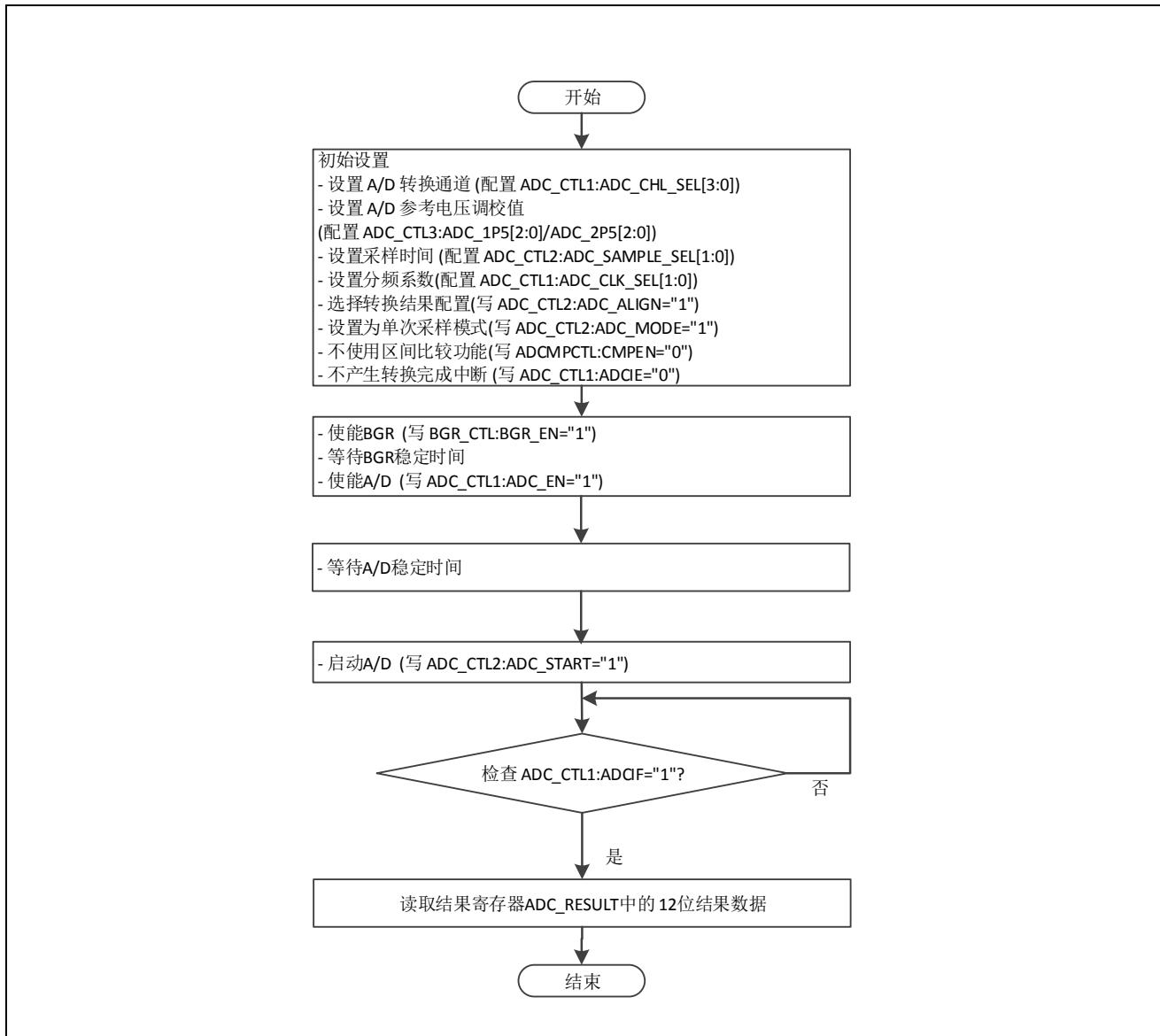
28.4 设定步骤示例

以下介绍了 12 位 A/D 转换器的设定步骤示例。

28.4.1 单次采样转换设定步骤示例

以下描绘了单次采样转换设定步骤示例。

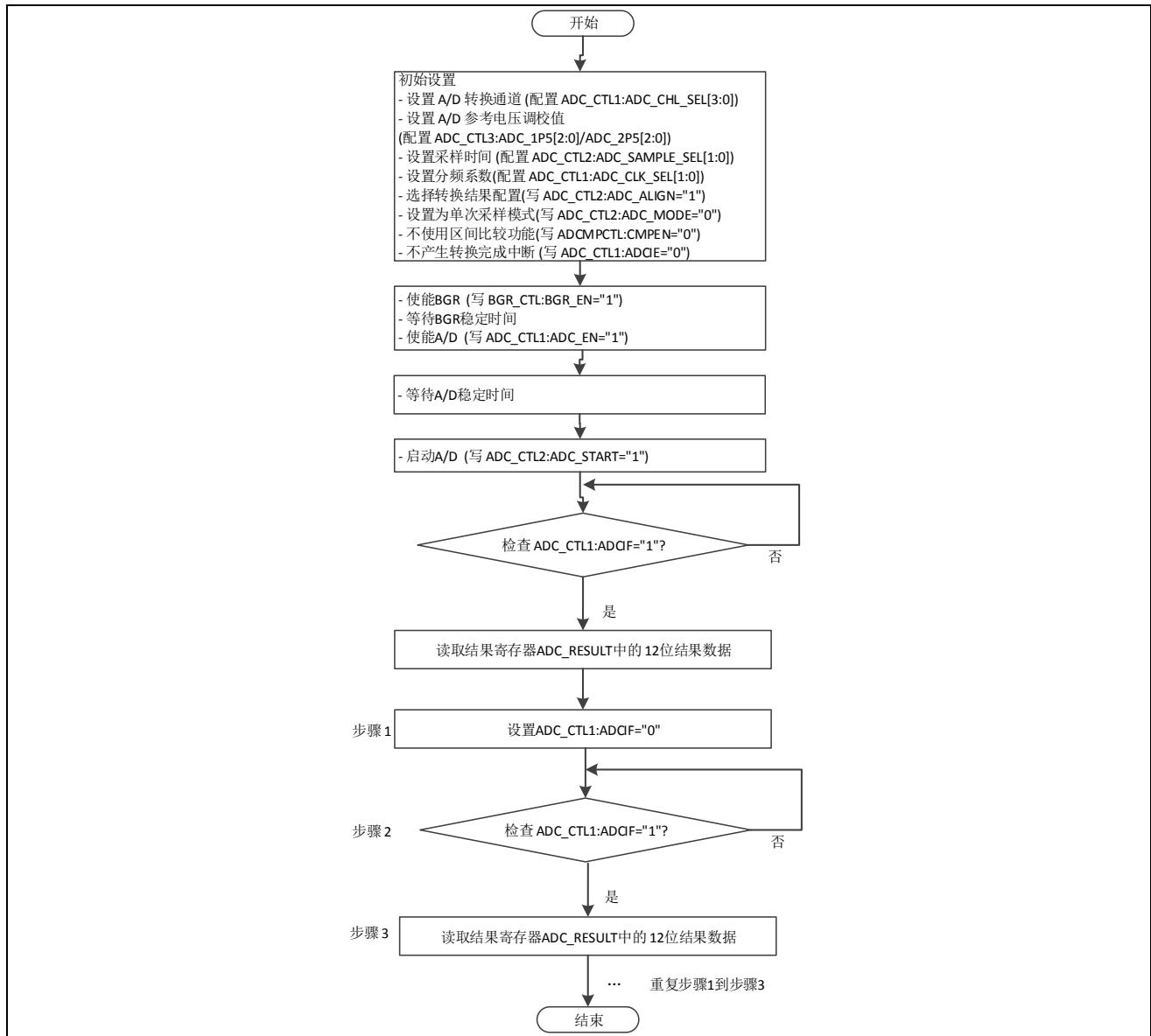
图 28.4-1 单次采样转换设定步骤示例



28.4.2 连续采样转换设定步骤示例

以下描绘了连续采样转换设定步骤示例。

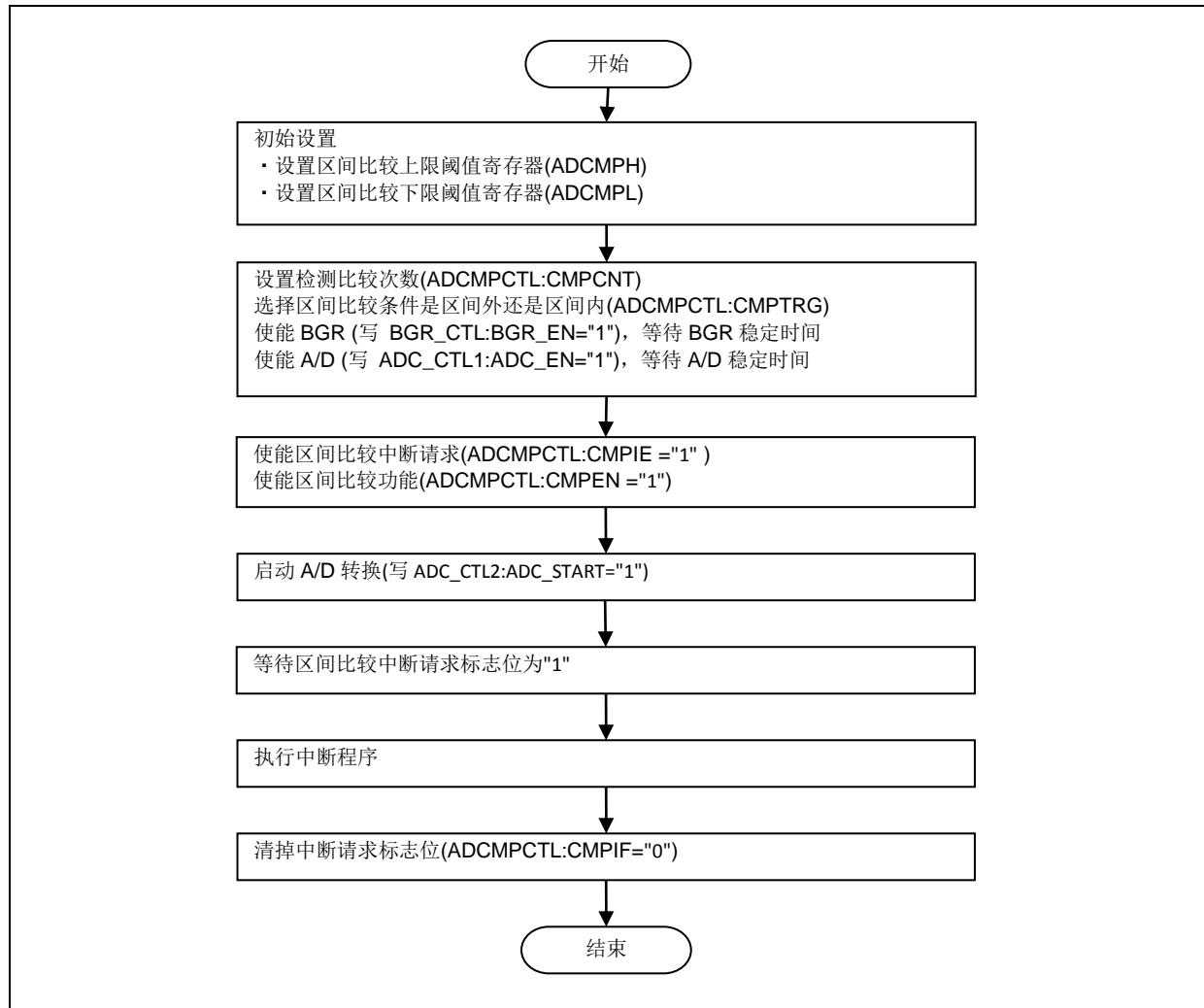
图 28.4-2 连续采样转换设定步骤示例



28.4.3 区间比较功能设定步骤示例

以下描绘了区间比较功能设定步骤示例。

图 28.4-3 区间比较功能设定步骤示例



28.4.4 设置采样转换时间

以下介绍了如何设置并计算采样转换时间。

设置采样时间示例

采样时间可以通过 A/D 控制寄存器 1 和 A/D 控制寄存器 2 来配置(ADC_CTL1/ADC_CTL2)。

采样时间 = 采样时钟周期 \times ADC_SAMPLE_SEL 的设置值

采样时钟周期 = APB1 总线时钟(PCLK1)周期 \times (ADC_CLK_SEL 设置的分频系数)

设置转换时间示例

转换时间可以通过 A/D 控制寄存器 1 来配置(ADC_CTL1)。

转换时间 = 转换时钟周期 \times (15 + 2)

转换时钟周期=APB1 总线时钟(PCLK1)周期 \times (ADC_CLK_SEL 设置的分频系数)

计算采样转换时间示例 (当 PCLK1 = 16 MHz (即一个周期 62.5 ns)时)

1) 采样时间

当 ADC_CLK_SEL = 2(分频系数:4), ADC_SAMPLE_SEL = 00 (3 个采样时钟周期)时采样
时间= $62.5 \text{ ns} \times 4 \times 3 = 750\text{ns}$

2) 转换时间

当 ADC_CLK_SEL = 2(分频系数:4)时
转换时间 = $62.5 \text{ ns} \times 4 \times (15+2) = 4250\text{ns}$

注意：

- ADC 的最高转换频率为 4MHz。

28.5 寄存器

以下介绍了 12 位 A/D 转换器所用寄存器。

表 28.5-1 12 位 A/D 转换器列表

寄存器简称	寄存器名称	参照章节
ADC_CTL1	A/D 控制寄存器1	28.5.1
ADC_CTL2	A/D控制寄存器2	28.5.2
ADC_CTL3	A/D控制寄存器3	28.5.3
ADC_RESULT	A/D 转换结果寄存器	28.5.4
ADCMPCTL	A/D 区间比较控制寄存器	28.5.5
ADCMPH	A/D区间比较上限阈值寄存器	28.5.6
ADCmpl	A/D区间比较下限阈值寄存器	28.5.7
ADCMPF	A/D区间比较阈值范围外标志寄存器	28.5.8
ADCMPIF	A/D区间比较中断标志寄存器	28.5.9

28.5.1 A/D 控制寄存器 1(ADC_CTL1)

以下描述了 A/D 控制寄存器 1 的功能(ADC_CTL1)。

寄存器配置

位	31		18	17	16
字段名	保留		ADCI_F	ADCI_E	
属性	-		R/W	R/W	
初始值	-		0	0	

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	保留			ADC_CHL_SEL			ADC_VREF		保留			ADC_CLK_SEL		ADC_EN		
属性	-			R/W			R/W		-			R/W		R/W		
初始值	-			1111			01		-			00		0		

寄存器功能

[31:18] 保留位，读值未定，写操作无效。

[17] ADCIF: A/D 转换中断请求标志位

位	描述	
	读	写
0	A/D 转换未完成 (初始值)	清零
1	A/D转换已完成	值不变且对其余寄存器无影响

[16] ADCIE: A/D 转换中断请求使能位

位	描述
0	禁止A/D中断请求(初始值)
1	使能A/D中断请求

当 A/D 转换中断请求标志位 (ADCIF) = “1” 且 A/D 转换中断请求使能位(ADCIE)=“1”时, 中断请求才会产生。

[15:12] 保留位，读值未定，写操作无效。

[11:8] ADC_CHL_SEL[3:0]: A/D 采样电压通道选择位

关于具体的端口请看产品的 "数据手册"。

3:0	描述
0000	通道A0
0001	通道A1
0010	通道A2

0011	通道A3
0100	通道A4
0101	通道A5
0110	通道A6
0111	通道A7
1000	通道A8
1001	通道A9
1010	通道A10
1011	通道A11
1100	1/3 VCC电压(需要设置ADC_CTL2.VINBUF_EN为0)
1101	保留
1110	内部BGR 1.2V电压(需要设置ADC_CTL2.VINBUF_EN为0)
1111	不选择任一通道(初始值)

[7:6] ADC_VREF [1:0]: A/D 参考电压选择位

1:0	描述
00	选择内部BGR 2.5V电压作为参考电压
01	选择内部BGR 1.5V 电压作为参考电压(初始值)

[5:3] 保留位, 读值未定, 写操作无效。

[2:1] ADC_CLK_SEL [1:0]: A/D 采样转换时钟选择位

1:0	描述
00	选择APB1总线时钟PCLK1 (初始值)*
01	选择APB1总线时钟PCLK1的1/2分频时钟
10	选择APB1总线时钟PCLK1的1/4分频时钟
11	选择APB1总线时钟PCLK1的1/8分频时钟

注意事项:主时钟频率≤4MHz 时, 该位才能设置为 2'b00。

[0] ADC_EN: A/D 使能位

位	描述
0	关闭 A/D转换器(初始值)
1	使能A/D 转换器

28.5.2 A/D 控制寄存器 2(ADC_CTL2)

以下描述了 A/D 控制寄存器 2 的功能(ADC_CTL2)。

寄存器配置

位	31							8
字段名					保留			
属性					-			
初始值					-			

位	7	6	5	4	3	2	1	0
字段名	ADC_SAMPLE_SEL	VINBUF_EN	保留	ADC_MODE	ADC_ALIGN	保留	ADC_START	
属性	R/W	R/W	-	R/W	R/W	-	R/W	
初始值	10	0	-	0	0	-	0	

寄存器功能

[31:8] 保留位，读值未定，写操作无效。

[7:6] ADC_SAMPLE_SEL [1:0]: A/D 采样时钟周期选择位

1:0	描述
00	选择 3个时钟周期进行采样
01	保留
10	选择6个时钟周期进行采样(初始值)
11	选择9个时钟周期进行采样

[5] VINBUF_EN: A/D 输入信号放大器使能位

本位在采样 A/D 转换器输入电压时使能信号放大器，可以采样外部微弱信号。

位	描述
0	使能A/D 信号放大器功能(初始值)
1	禁止A/D信号放大器功能

[4] 保留位，读值未定，写操作无效。

[3] ADC_MODE: A/D 采样模式选择位

位	描述
0	连续采样转换模式(初始值)
1	单次采样转换模式

[2] ADC_ALIGN: A/D 结果配置选择位

位	描述
0	配置转换结果左对齐。 (初始值)

1	配置转换结果右对齐
---	-----------

1] 保留位，读值未定，写操作无效。

[0] ADC_START: A/D 采样启动位

位	描述
0	停止 A/D 采样(初始值)
1	启动 A/D采样

在单次采样转换模式下，ADC_START 会在一次采样转换结束后自动清零。该位在 ADC_CTL1.ADC_EN=1 时设置才有效。

28.5.3 A/D 控制寄存器 3(ADC_CTL3)

以下描述了 A/D 控制寄存器 3 的功能(ADC_CTL3)。

寄存器配置

位	31																16
字段名	KEY																
属性	R/W*																
初值	0000h																

位	15		8	7:5	4:2	1	0
字段名	-		ADC_1P5[2:0]	ADC_2P5[2:0]	保留		
属性	-		R/W	R/W	-		
初值	-		XXX	XXX	-		

寄存器功能

位[31:16] KEY: ADC_CTL3 读、写密码。读这几位始终返回 0000h。写入 ADC_CTL3 的数据只在写入正确的密码 16'hD27A 到这几位的同时，才能够有效。

[15:8] 保留位，读值未定，写操作无效。

[7:5] ADC_1P5[2:0]: ADC 内部 1.5V 参考电压微调位

寄存器值将会在上电复位后更新为 Flash 地址 0x0010_0004 的第 18~16 位数值,每次使用 ADC 前，需要从 FLASH 读取调校好的出厂值。

Bit7:5	说明
读取	读出目前的设置值
写入	设置ADC内部1.5V参考电压

[4:2] ADC_2P5[2:0]: ADC 内部 2.5V 参考电压微调位

寄存器值将会在上电复位后更新为 Flash 地址 0x0010_0004 的第 21~19 位数值,每次使用 ADC 前，需要从 FLASH 读取调校好的出厂值。

Bit7:5	说明
读取	读出目前的设置值
写入	设置ADC内部2.5V参考电压

[1:0] 保留位，读值未定，写操作无效。

28.5.4 A/D 转换结果寄存器(ADC_RESULT)

A/D 转换结果寄存器(ADC_RESULT)存储了 12 位数字转换结果。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	保留															
属性	-															
初始值	-															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	ADC_RESULT															
属性	R															
初始值	0x0000															

寄存器功能

[31:16] 保留位，读值未定，写操作无效。

[15: 0] ADC_RESULT[15:0]: A/D 转换结果位

ADC_RESULT[15:0] = {4'h0, A/D 结果[11:0]}(当 ADC_CTL2.ADC_ALIGN = “1”时)

ADC_RESULT[15:0] = {A/D 结果[11:0],4'h0}(当 ADC_CTL2.ADC_ALIGN = “0”时)

28.5.5 A/D 区间比较控制寄存器 (ADCMPCCTL)

以下描述了 A/D 区间比较控制寄存器(ADCMPCCTL)的功能。

寄存器配置

位	31						16
字段名				保留			
属性				-			
初始值				-			

位	15		13	12			8
字段名		CMPCNT_ST			保留		
属性		R			-		
初始值		000			-		

位	7		5	4	3	2	1 0
字段名		CMPCNT		CMPTR G	CMPIE	CMPEN	保留
属性		R/W		R/W	R/W	R/W	-
初始值		001		0	0	0	-

寄存器功能

[31:8] 保留位，读值未定，写操作无效。

[15:13]: CMPCNT_ST[2:0]: A/D 转换结果比较检测次数状态位

2:0	描述
000	当前已检测状态数：0次
001	当前已检测状态数：1次
010	当前已检测状态数：2次
011	当前已检测状态数：3次
100	当前已检测状态数：4次
101	当前已检测状态数：5次
110	当前已检测状态数：6次
111	当前已检测状态数：7次

[7:5]: CMPCNT[2:0]: A/D 转换结果比较检测次数位

2:0	描述
000	禁止设置
001	设定的比较检测次数：1次
010	设定的比较检测次数：2次
011	设定的比较检测次数：3次
100	设定的比较检测次数：4次
101	设定的比较检测次数：5次

110	设定的比较检测次数: 6次
111	设定的比较检测次数: 7次

当转换结果比较检测次数状态次数(CMPCNT_ST)跟转换结果比较检测次数(CMPCNT)一样时, 区间比较中断标志位 (CMPIF) 会被置 “1”。此时 CMPCNT_ST 会停止并跟 CMPCNT 保持一致。

注意:

- 在使能区间比较功能后(CMPEN="1"), 不要改变转换结果比较检测次数位 (CMPCNT)。
- 不要给转换结果比较检测次数位 (CMPCNT)设置成 “000” 。

[4] CMPTRG: A/D 区间比较检测条件(区间内外)选择

位	描述
0	区间外检测(初始值)
1	区间内检测

A/D 区间比较检测条件位的功能是在检测比较时, 选择上限阈值位(CDH) 和下限阈值位(CDL) 所定义的区间内还是区间外进行检测比较。

当选择区间外检测(CMPTRG="0")时, 检测条件为:

A/D 转换结果 > 上限阈值位(CDH) 或, A/D 转换结果 < 下限阈值位(CDL)

当选择区间内检测 (CMPTRG = "1")时, 检测条件为:

A/D 转换结果 \leq 上限阈值位 (CDH) 且, A/D 转换结果 \geq 下限阈值位 (CDL)

当选择区间外检测 (CMPTRG = "0")时, 可以通过读取 A/D 区间比较阈值范围外标志位 (CMPIF) 来判断当前转换结果是高于上限阈值还是低于下限阈值。

[3] CMPIE: 区间比较中断请求使能位

位	描述
0	区间比较中断禁止(初始值)
1	区间比较中断使能

当区间比较中断标志位 (CMPIF) 是 “1” 且区间比较中断请求使能 (CMPIE="1"), 中断请求才会产生。

[2] CMPEN: 区间比较使能位

位	描述
0	禁止A/D区间比较功能(初始值)
1	使能A/D区间比较功能

当区间比较使能位 (CMPEN) 为 "0" 时, 区间比较功能禁止。同时, A/D 转换结果比较检测次数状态次数(CMPCNT_ST)会被初始化到 “000”。

[1:0] 保留位, 读值未定, 写操作无效。

28.5.6 A/D 区间比较上限阈值寄存器(ADCMPH)

A/D 区间比较上限阈值寄存器(ADCMPH)用来设置区间比较时的上限值。

寄存器配置

位	31	30	29	28	27	26	25	24
字段名	CDH11	CDH10	CDH9	CDH8	CDH7	CDH6	CDH5	CDH4
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	23	22	21	20	19	18	17	16
字段名	CDH3	CDH2	保留					
属性	R/W	R/W	R					
初始值	0	0	000000					

寄存器功能

[32:22] CDH11 到 CDH2: 上限阈值位

本位定义了区间比较时的上限阈值。

31:22	描述
上限阈值	

[bit21:16] 保留位，读值为“0”，写操作无效。

注意：

- 当区间比较功能使能后，A/D 转换结果的高 10 位(11:2) 会跟此寄存器(ADCMPH)进行比较，低 2 位不会进行比较。

28.5.7 A/D 区间比较下限阈值寄存器(ADCMPL)

A/D 区间比较下限阈值寄存器(ADCMPL) 用来设置区间比较时的下限值。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	CDL11	CDL10	CDL9	CDL8	CDL7	CDL6	CDL5	CDL4
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CDL3	CDL2				保留		
属性	R/W	R/W				R		
初始值	0	0			000000			

寄存器功能

[15:6] CDL11 到 CDL2: 下限阈值位

本位定义了区间比较时的下限阈值。

15:6	描述
下限阈值	

[bit5:0] 保留位, 读值为“0”, 写操作无效。

注意:

- 当区间比较功能使能后, A/D 转换结果的高 10 位(11:2) 会跟此寄存器(ADCMPL)进行比较, 低 2 位不会进行比较。

28.5.8 A/D 区间比较阈值范围外标志寄存器(ADCMPF)

A/D 区间比较阈值范围外标志寄存器(ADCMPF)用来指示当检测条件为区间范围外时，检测的转换结果是高于上限阈值还是低于下限阈值。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	保留															
属性	-															
初始值	-															

位	15														1	0
字段名	保留															CMPF
属性	-															R
初始值	-															0

寄存器功能

[31:1] 保留位，读值未定，写操作无效。

[0] CMPF: A/D 区间比较阈值范围外标志位

位	描述
0	A/D 结果 < 下限阈值
1	A/D 结果 > 上限阈值

设置为区间外检测时(CMPTRG="0"),当区间比较结果落在区间范围内, A/D 区间比较阈值范围外标志位 CMPF 保持之前的值。

当区间比较中断标志位 (CMPIF) 为 “1”时, A/D 区间比较阈值范围外标志位(CMPF) 不会更新并保持之前的值,即使在设置为区间外检测条件下(CMPTRG="0")检测到了区间外的转换结果。

设置为区间内检测时(CMPTRG = "1"), A/D 区间比较阈值范围外标志位无意义并保持之前的值。

28.5.9 A/D 区间比较中断标志寄存器(ADCMPIF)

区间比较中断标志寄存器(ADCMPIF)指示了由满足检测次数的区间比较结果产生的中断标志。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段名	保留															
属性	-															
初始值	-															

位	15														1	0
字段名	保留															
属性	-															
初始值	-															

寄存器功能

[31:1] 保留位，读值未定，写操作无效。

[0] CMPIF: 区间比较中断标志位

位	描述	
	读	写
0	A/D区间比较中断清除	清零
1	检测的A/D转换结果满足设定好的条件，产生了中断标志	无效且不影响其他设定

当 CMPIF 位以及区间比较中断请求使能位(CMPIE) 都为 "1"时,区间比较中断才会产生。

注意：

- 当软件清零 (写 CMPIF =”0”)与硬件置位同时发生时，硬件置位具有更高优先级。

29. LCD 控制器

29.1 概述

LCD 控制器的功能如下。

支持 1/8 占空比、1/6 占空比、1/4 占空比、1/3 占空比、1/2 占空比和静态显示模式。

- 1/8 占空比模式

Common 信号可以选择引脚 COM0 ~ COM7, segment 信号最多 36 个

LCD 显示数据存储器 (LCDRAM) 最大 36×8 bits

支持 1/3 偏压和 1/4 偏压

- 1/6 占空比模式

Common 信号可以选择引脚 COM0 ~ COM5, segment 信号最多 38 个

LCD 显示数据存储器 (LCDRAM) 最大 38×6 bits

支持 1/3 偏压和 1/4 偏压

- 1/4 占空比模式

Common 信号可以选择引脚 COM0 ~ COM3, segment 信号最多 40 个

LCD 显示数据存储器 (LCDRAM) 最大 40×4 bits

支持 1/2 偏压和 1/3 偏压

- 1/3 占空比模式

Common 信号可以选择引脚 COM0 ~ COM2, segment 信号最多 40 个

LCD 显示数据存储器 (LCDRAM) 最大 40×3 bits

支持 1/2 偏压和 1/3 偏压

- 1/2 占空比模式

Common 信号可以选择引脚 COM0 ~ COM1, segment 信号最多 40 个

LCD 显示数据存储器 (LCDRAM) 最大 40×2 bits

支持 1/2 偏压和 1/3 偏压

- 静态模式

Common 信号仅支持引脚 COM0, segment 信号最多 40 个

LCD 显示数据存储器 (LCDRAM) 最大 40×1 bit

偏压不可选择

LD 驱动电压可以来自内部电阻分压、外部电阻分压和外部电容。

选择内部电阻分压时，支持对比度调节，同时可以选择多种充电模式。

充电模式有：大电流模式和快速充电模式。

- 大电流模式

此模式可以得到较好的显示效果，但功耗相对较大快速充电模式。

- 快速充电模式

此模式可以选择合适的充电时间，兼顾显示效果和功耗。

LD 工作时钟可以选择外部低速晶振（LSXT）和内部低速 RC（LICR）

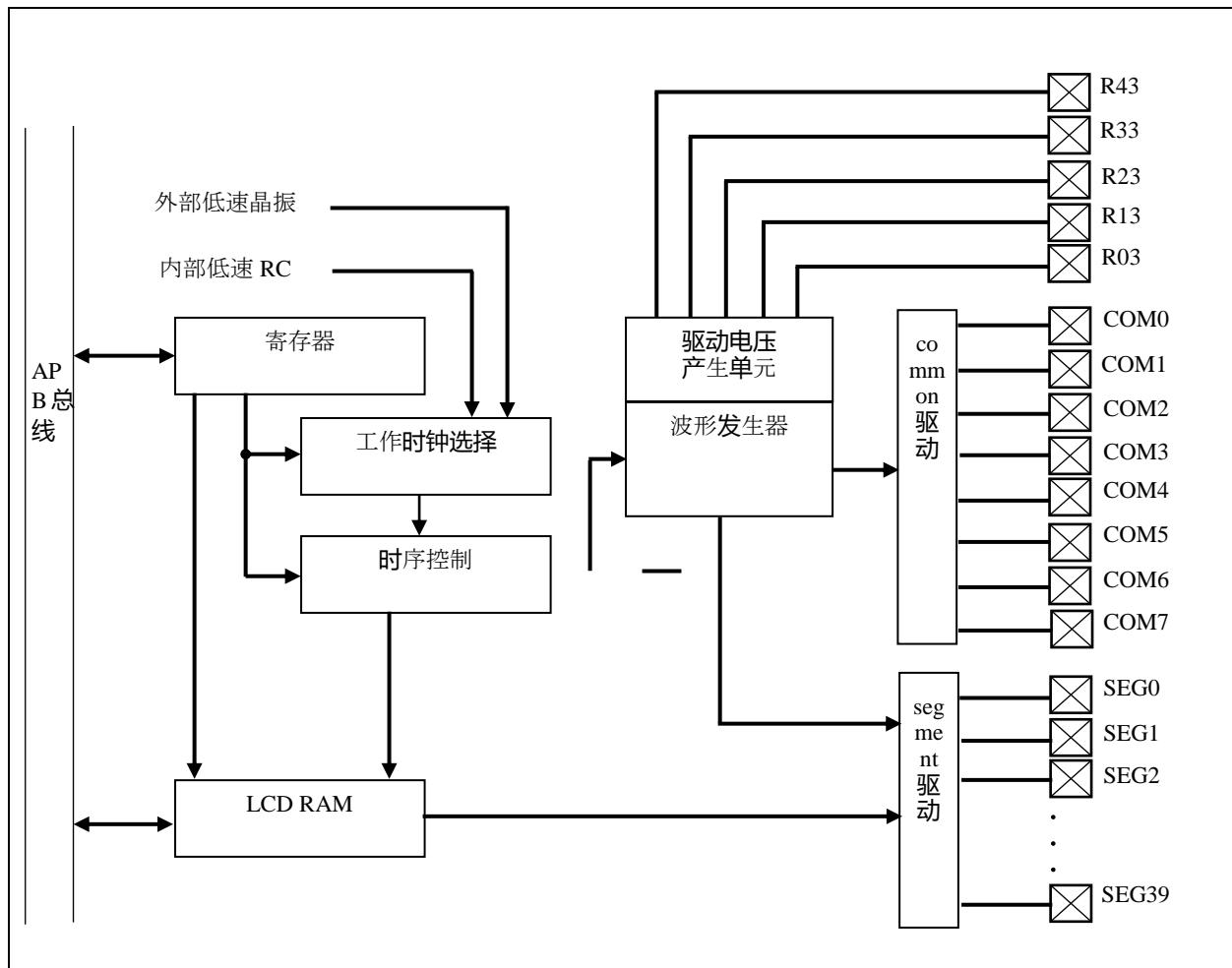
支持 standby 模式和 deep standby 模式下工作

支持 A 类和 B 类 LCD 驱动波形

29.2 配置

LCD 控制器的配置如图 29.2-1 所示。

图 29.2-1 LCD 控制器配置图



- **工作时钟选择**

可以选择外部低速晶振或内部低速 RC 时钟作为 LCD 的工作时钟

- **时序控制**

时序控制模块根据当前的寄存器设置和 LCD 时钟产生 common 和 segment 的原始波形

- **波形发生器**

波形发生器将 common 和 segment 的原始波形转换为 AC 波形

- **Common/Segment 驱动**

本模块用来驱动外部引脚 COM0 ~ COM7 和 SEG0 ~ SEG39

- **LCD RAM**

LCD RAM 用来存储 LCD 的显示数据，也可以用作普通数据寄存器使用

LCD RAM 中的数据改写后，LCD 显示的数据立即发生变化

- 驱动电压产生单元

本模块由内部电阻、外部电阻或外部电容产生 LCD 的驱动电压

29.2.1 LCD 驱动电压产生单元

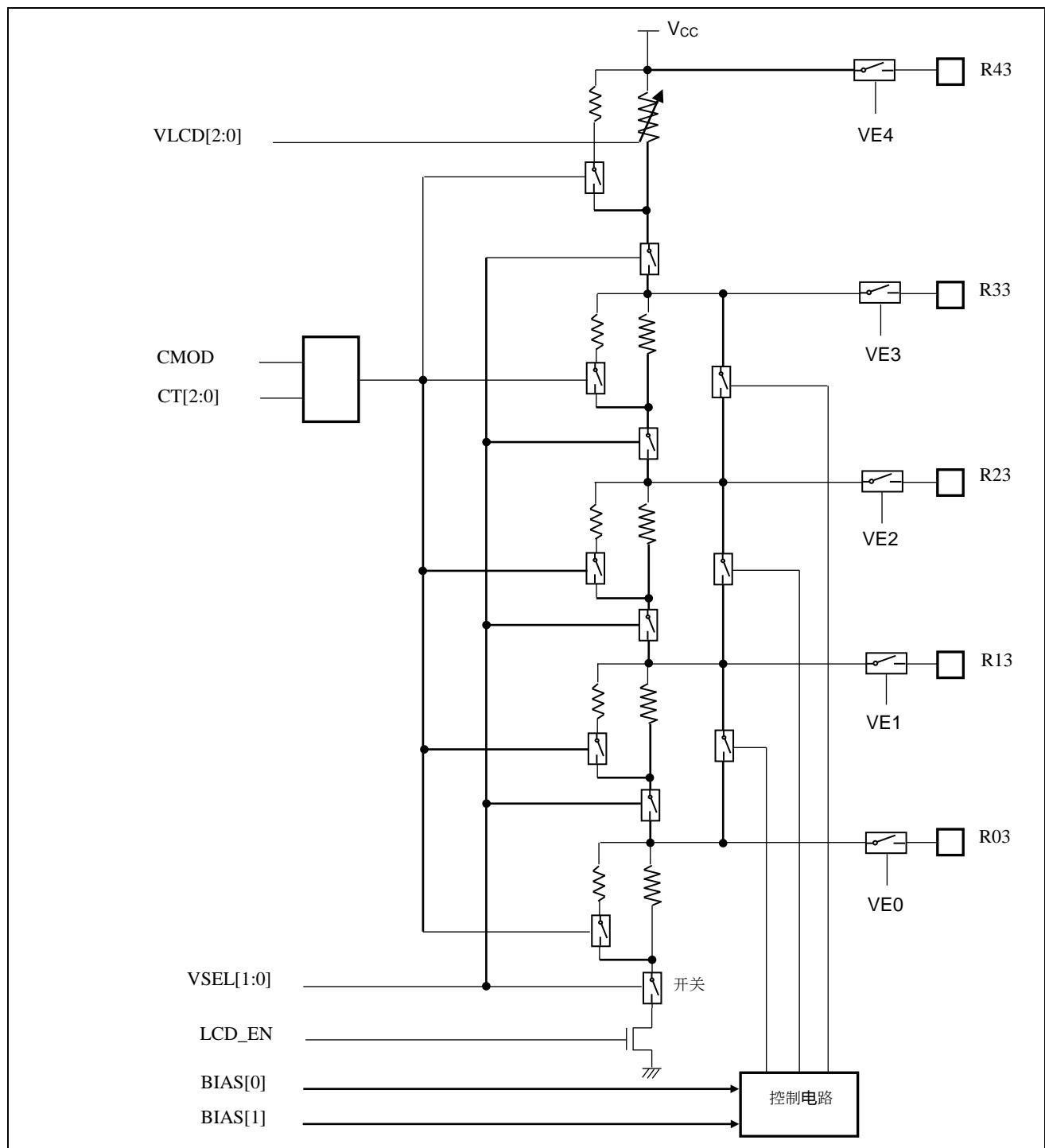
LCD 驱动电压可以来自内部电阻、外部电阻或外部电容。

内部电阻分压

LCD 控制器内部集成了电阻分压电路，要使用内部电阻分压模式，寄存器 LCDCR2 的 VSEL 位需要写“00”。

选择内部电阻分压模式时，写寄存器 LCDDVC 中各位为“0”，R03 ~ R43 可以作为 GPIO 使用。

图 29.2-2 内部电阻分压电路示意图

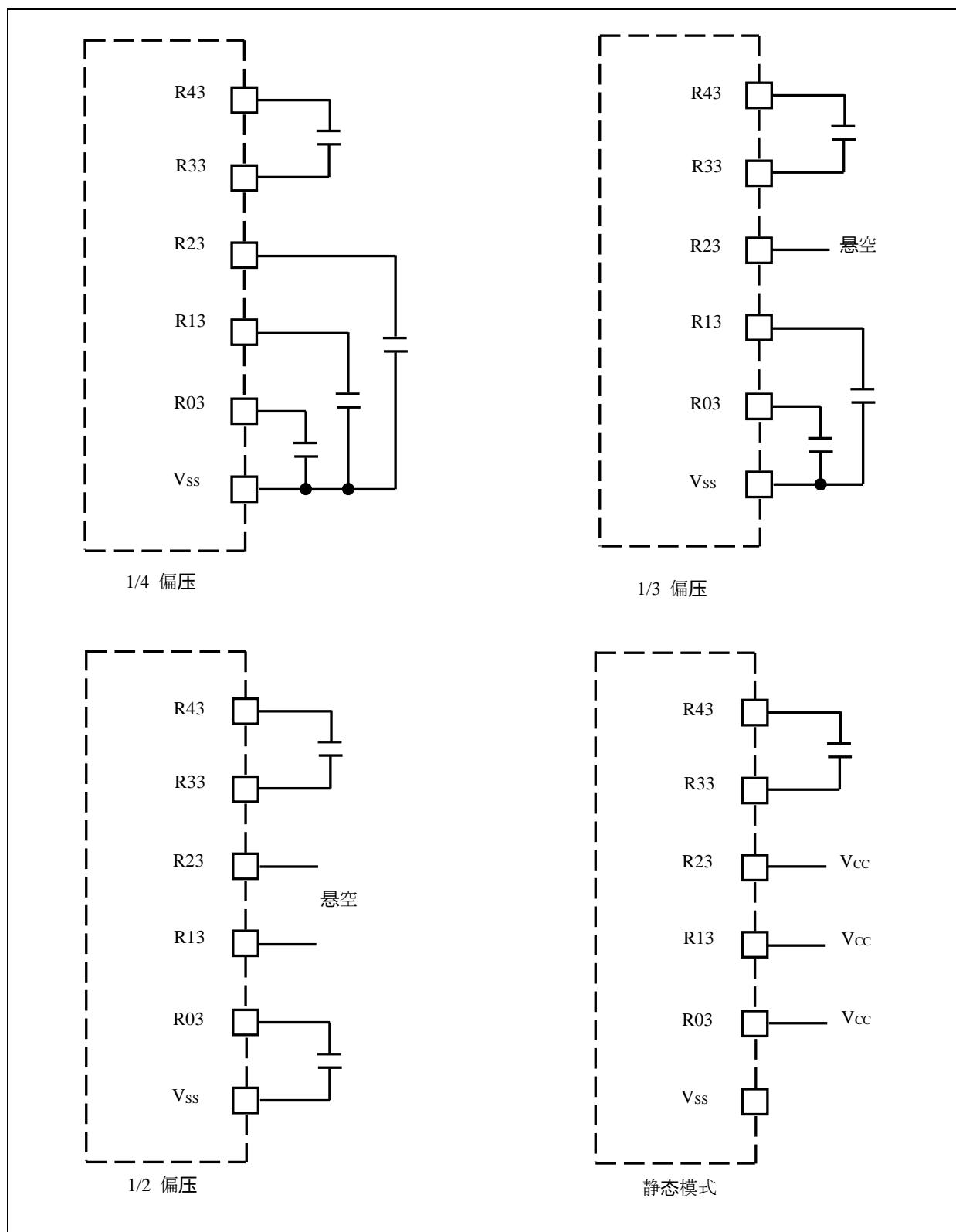


外部电容

要使用外部电容模式，寄存器 LCDCR2 的 VSEL 位需要写“01”。选择外部电容模式时，寄存器 LCDDVC 中各位必须写“1”，R03 ~ R43 不能作为 GPIO 使用。

外部电容模式时，不同偏压对应的外部连接如图 29.2-3 所示。

图 29.2-3 外部电容模式连接图

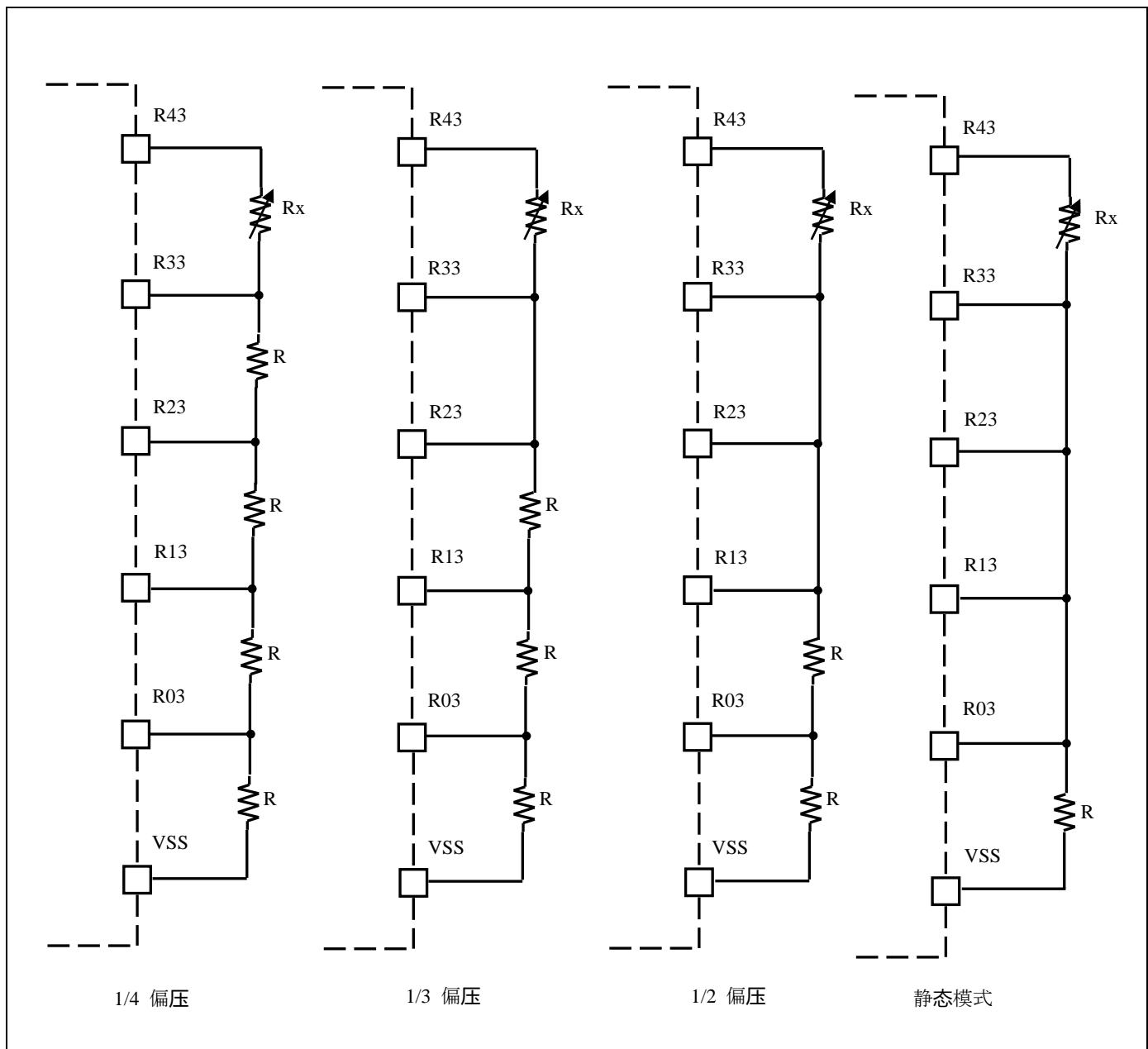


外部电阻分压

要使用外部电阻模式，寄存器 LCDCR2 的 VSEL 位需要写“10”。选择外部电阻模式时，寄存器 LCDDVC 中各位必须写“1”，R03 ~ R43 不能作为 GPIO 使用。

外部电阻模式时，不同偏压对应的外部连接如图 29.2-4 所示。

图 29.2-4 外部电阻模式连接图



注意：

- Rx 为可调电阻，用于调节 LCD 显示亮度。
- 请根据使用的 LCD 屏幕选择合适的电阻 R。

29.2.2 引脚

LCD 控制器相关的引脚有，8 个 common 信号输出引脚 COM0 ~ COM7, 40 个 segment 信号输出引脚 SEG0 ~ SEG39, 5 个 LCD 驱动电压输入引脚 R03 ~ R43。所有引脚都可以作为通用输入输出端口（GPIO）。

- 当这些引脚用作 LCD 功能时，寄存器 LCDDVC、LCDCE0 和 LCDSE0 ~ LCDSE4 中相应位写“1”。
- 当这些引脚用作 GPIO 功能时，寄存器 LCDDVC、LCDCE0 和 LCDSE0 ~ LCDSE4 中相应位写“0”。

引脚 COM0 ~ COM7

1/8 占空比模式下，COM0 ~ COM7 作为 common 信号输出引脚。

1/6 占空比模式下，COM0 ~ COM5 作为 common 信号输出引脚，COM6 和 COM7 引脚可以作为 segment 信号输出或 GPIO。如果作为 segment 信号输出，请写寄存器 LCDSE4 中的相应位为“1”，此时寄存器 LCDCE0 中的 COME6 和 COME7 可以忽略。

1/4、1/3 和 1/2 占空比模式下，COM0 ~ COM3 可以作为 common 信号输出引脚，COM4 ~ COM7 引脚可以作为 segment 信号输出或 GPIO。如果作为 segment 信号输出，请写寄存器 LCDSE4 中的相应位为“1”，此时寄存器 LCDCE0 中的 COME4 ~ COME7 可以忽略。

静态模式下，仅 COM0 作为 common 信号输出引脚，COM1 ~ COM3 自动作为 GPIO，COM4 ~ COM7 引脚可以作为 segment 信号输出或 GPIO。如果作为 segment 信号输出，请写寄存器 LCDSE4 中的相应位为“1”，此时寄存器 LCDCE0 中的 COME4 ~ COME7 可以忽略。

引脚 SEG0 ~ SEG39

1/8 占空比模式下，引脚 SEG0 ~ SEG35 可以作为 segment 信号输出引脚。

1/6 占空比模式下，引脚 SEG0 ~ SEG35、SEG38 和 SEG39 可以作为 segment 信号输出引脚。

1/4、1/3、1/2 占空比和静态模式下，引脚 SEG0 ~ SEG39 可以作为 segment 信号输出引脚。

引脚 R03 ~ R43

R03 ~ R43 是 LCD 驱动电压输入引脚。

使用外部电阻分压或外部电容产生 LCD 驱动电压时，请写寄存器 LCDDVC 中的 VE0 ~ VE4 为“1”。此时引脚 R03 ~ R43 不允许用作 GPIO。

29.3 操作

LCD 控制器的工作模式。

不同占空比条件下可用的偏压如表 29.3-1 所示, 请设置寄存器 LCDCR0 选择合适的占空比和偏压。

表 29.3-1 占空比与偏压的关系

占空比	1/2 偏压	1/3 偏压	1/4 偏压
1/2 占空比	O	O	X
1/3 占空比	O	O	X
1/4 占空比	O	O	O
1/6 占空比	X	O	O
1/8 占空比	X	O	O

O: 允许设置

X: 禁止设置

不同系统运行模式下 LCD 控制器的状态

LCD 控制器可以在工作模式和低功耗模式下工作, 详细信息请参考表 29.3-2。

表 29.3-2 系统运行模式与 LCD 的关系

系统运行模式	LCD状态	
工作模式	O	
空闲模式	O	
Standby模式	Standby RTC模式	O
	Standby STOP模式	X
Deep Standby模式	Deep standby RTC模式	O
	Deep standby STOP模式	X

O: 允许工作

X: 禁止工作

29.3.1 LCD 驱动波形

为了保护 LCD 面板，LCD 控制器输出的驱动波形，每隔一段时间液晶两端的电压反向一次。

LCD 控制器可以输出两种驱动波形：A 类和 B 类。

- A 类驱动波形中，COM 和 SEG 的电压差在一个 common 扫描周期内完成反向。
- B 类驱动波形中，COM 和 SEG 的电压差在两个 common 扫描周期内完成反向。

1/8 占空比模式：

1/8 占空比，1/3 偏压，A 类/B 类波形

1/8 占空比，1/4 偏压，A 类/B 类波形

1/6 占空比模式：

1/6 占空比，1/3 偏压，A 类/B 类波形

1/6 占空比，1/4 偏压，A 类/B 类波形

1/4 占空比模式：

1/4 占空比，1/3 偏压，A 类/B 类波形

1/4 占空比，1/4 偏压，A 类/B 类波形

1/3 占空比模式：

1/3 占空比，1/2 偏压，A 类/B 类波形

1/3 占空比，1/3 偏压，A 类/B 类波形

1/2 占空比模式：

1/2 占空比，1/2 偏压，A 类/B 类波形

1/2 占空比，1/3 偏压，A 类/B 类波形

静态模式：

引脚 COM0 可以作为 common 信号输出，SEG0 到 SEG39 可以作为 segment 输出。此模式下，偏压和驱动波形由硬件固定，不可以软件配置。

29.3.1.1 1/8 占空比模式输出波形举例

1/8 占空比模式下，LCD 可以配置为 1/3 偏压或 1/4 偏压，A 类波形或 B 类波形。

图 29.3-1 为 1/8 占空比，1/4 偏压，A 类波形。图 29.3-2 为 1/8 占空比，1/4 偏压，B 类波形。

下面的例子中，LCDRAM00 数据为 0x01，COM0 和 SEG0 的电压差最大，所以 COM0 和 SEG0 之间的液晶将被点亮。

表 29.3-3 1/8 占空比模式下 LCDRAM00 中的数据

Segment	LCDRAM00							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEG0	0	0	0	0	0	0	0	1

图 29.3-1 1/8 占空比, 1/4 偏压, A 类波形

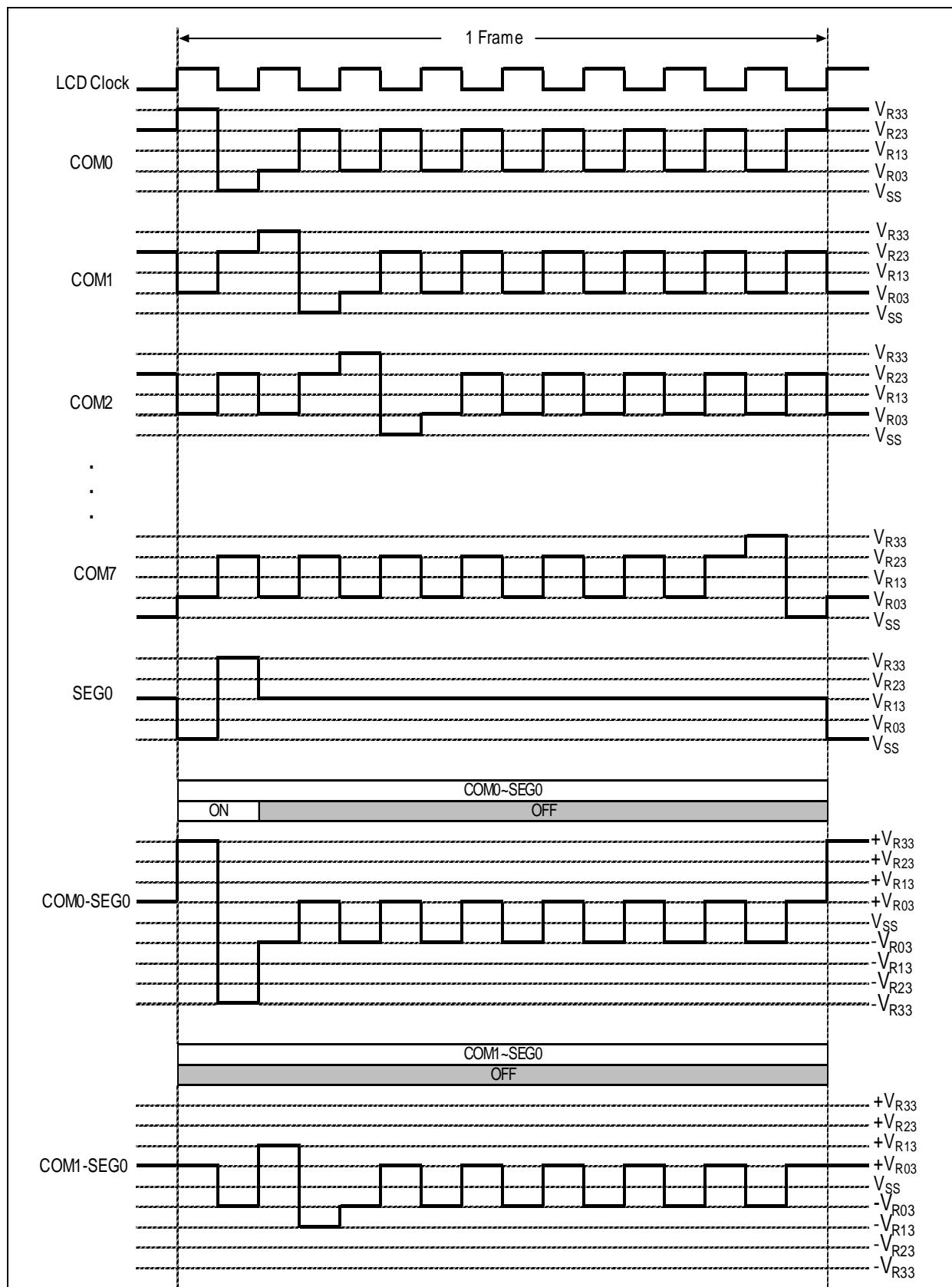
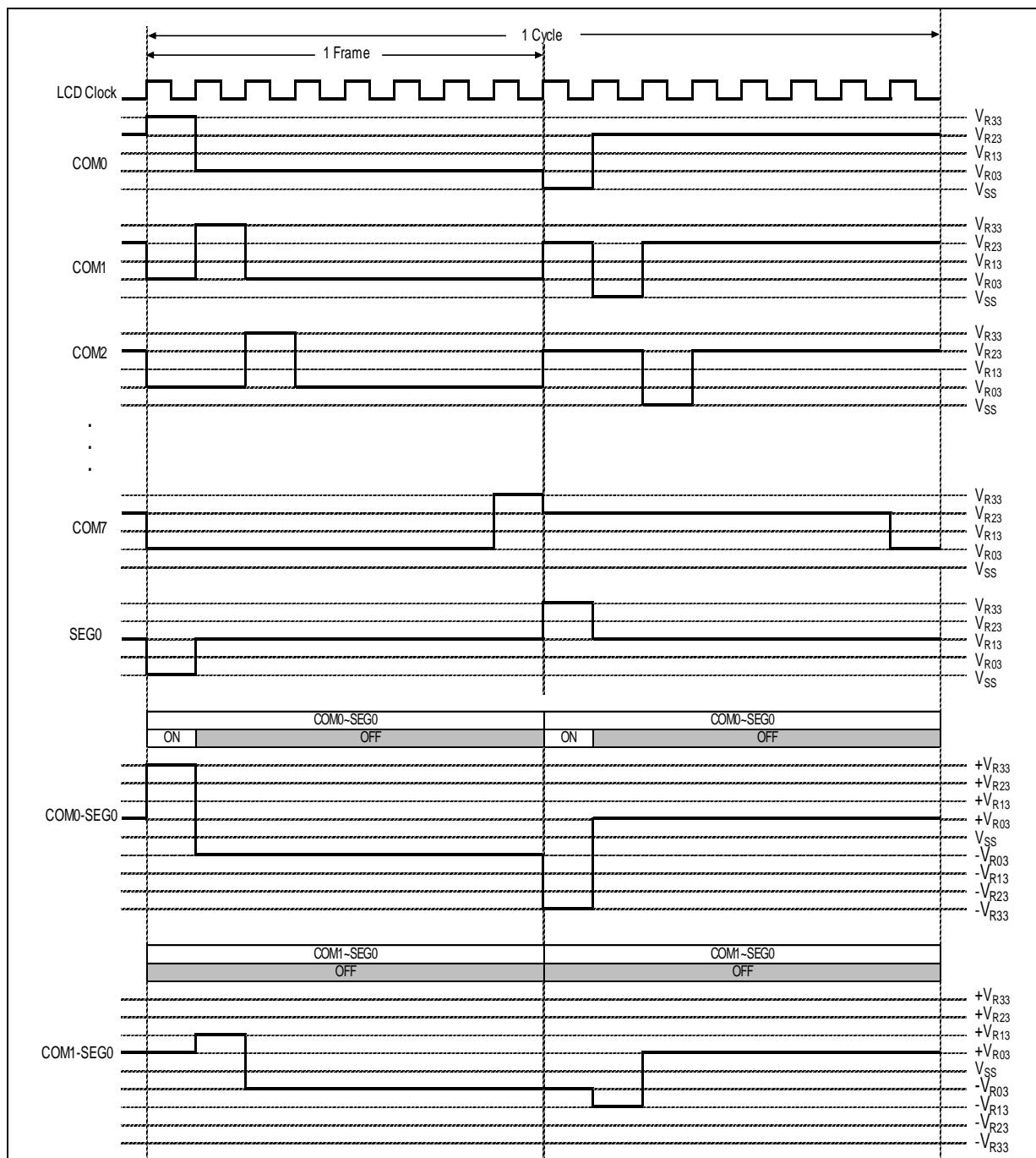


图 29.3-2 1/8 占空比, 1/4 偏压, B 类波形



29.3.1.2 1/6 占空比模式输出波形举例

1/6 占空比模式下，LCD 可以配置为 1/3 偏压或 1/4 偏压，A 类波形或 B 类波形。

图 29.3-3 为 1/6 占空比，1/3 偏压，A 类波形。图 29.3-4 为 1/6 占空比，1/3 偏压，B 类波形。

下面的例子中，LCDRAM00 数据为 0x01，COM0 和 SEG0 的电压差最大，所以 COM0 和 SEG0 之间的液晶将被点亮。

表 29.3-4 1/6 占空比模式下 LCDRAM00 中的数据

Segment	LCDRAM00							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEG0	-	-	0	0	0	0	0	1

图 29.3-3 1/6 占空比, 1/3 偏压, A 类波形

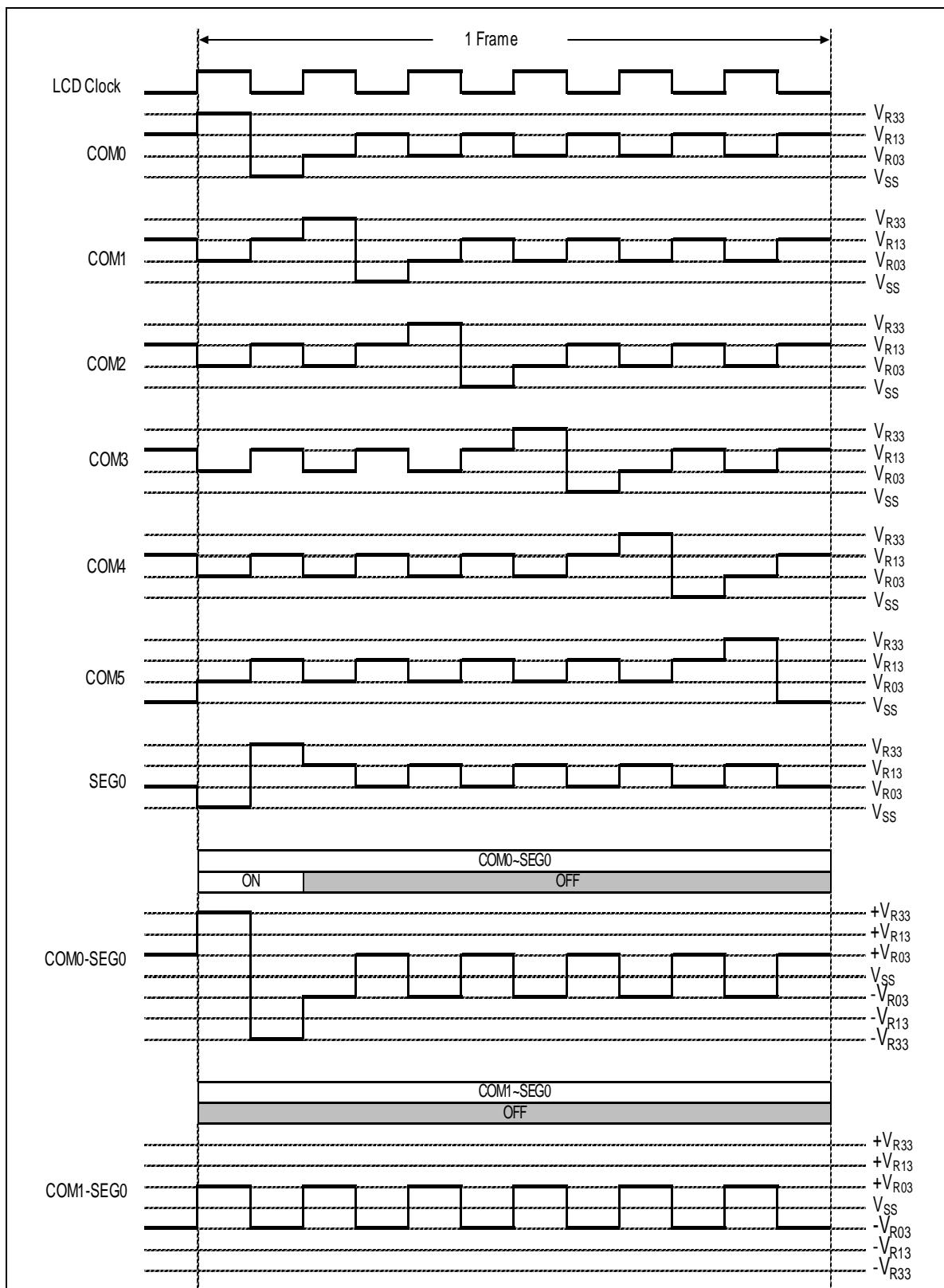
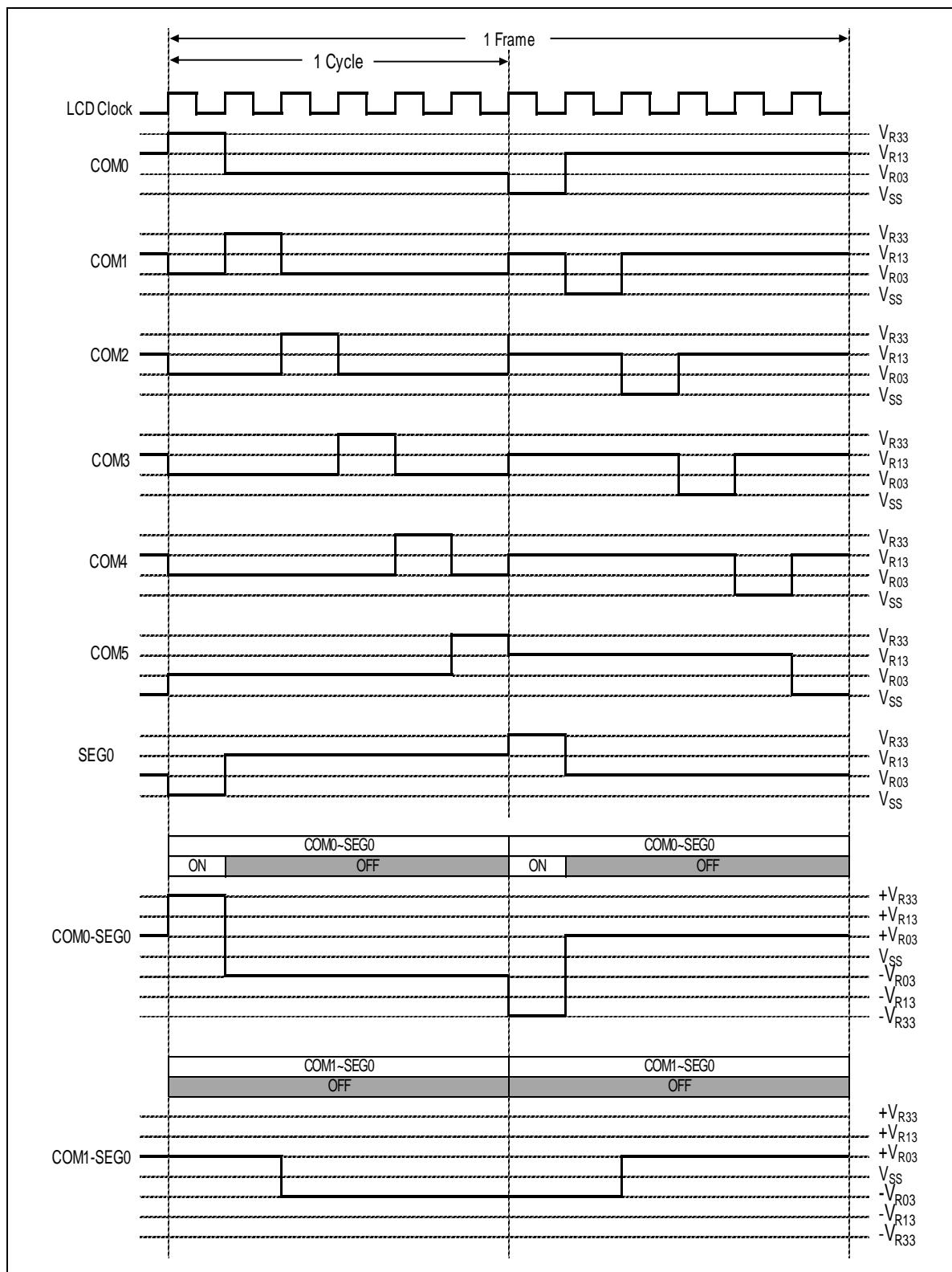


图 29.3-4 1/6 占空比, 1/3 偏压, B 类波形



29.3.1.3 1/4 占空比模式输出波形举例

1/4 占空比模式下，LCD 可以配置为 1/2 偏压或 1/3 偏压，A 类波形或 B 类波形。

图 29.3-5 为 1/4 占空比，1/2 偏压，A 类波形。图 28.3.4-6 为 1/4 占空比，1/2 偏压，B 类波形。

下面的例子中，LCDRAM00 数据为 0x05，COM0 和 SEG0、COM2 和 SEG0 的电压差最大，所以 COM0 和 SEG0 以及 COM2 和 SEG0 之间的液晶将被点亮。

表 29.3-5 1/4 占空比模式下 LCDRAM00 中的数据

Segment	LCDRAM00							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEG0	-	-	-	-	0	1	0	1

图 29.3-5 1/4 占空比, 1/2 偏压, A 类波形

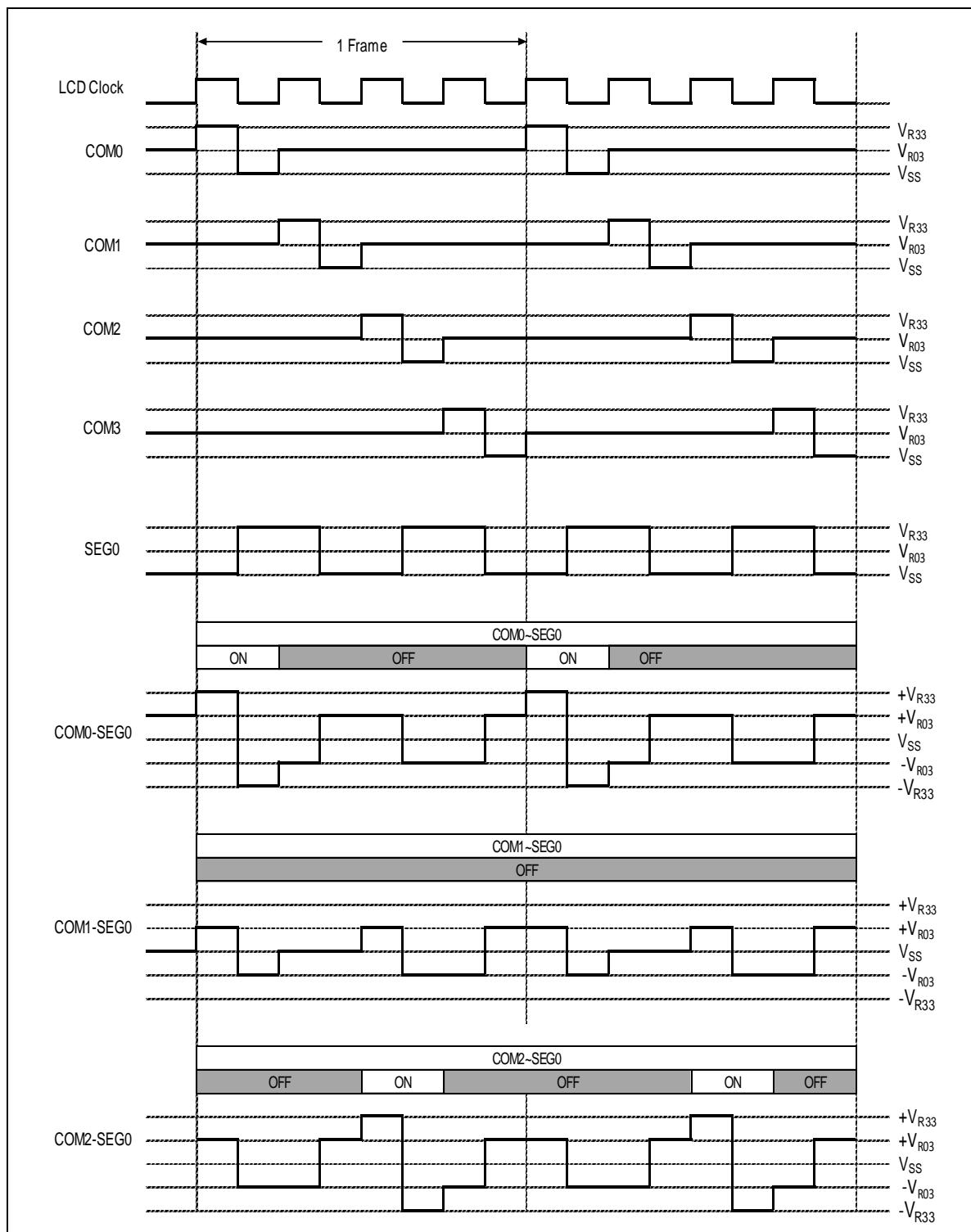
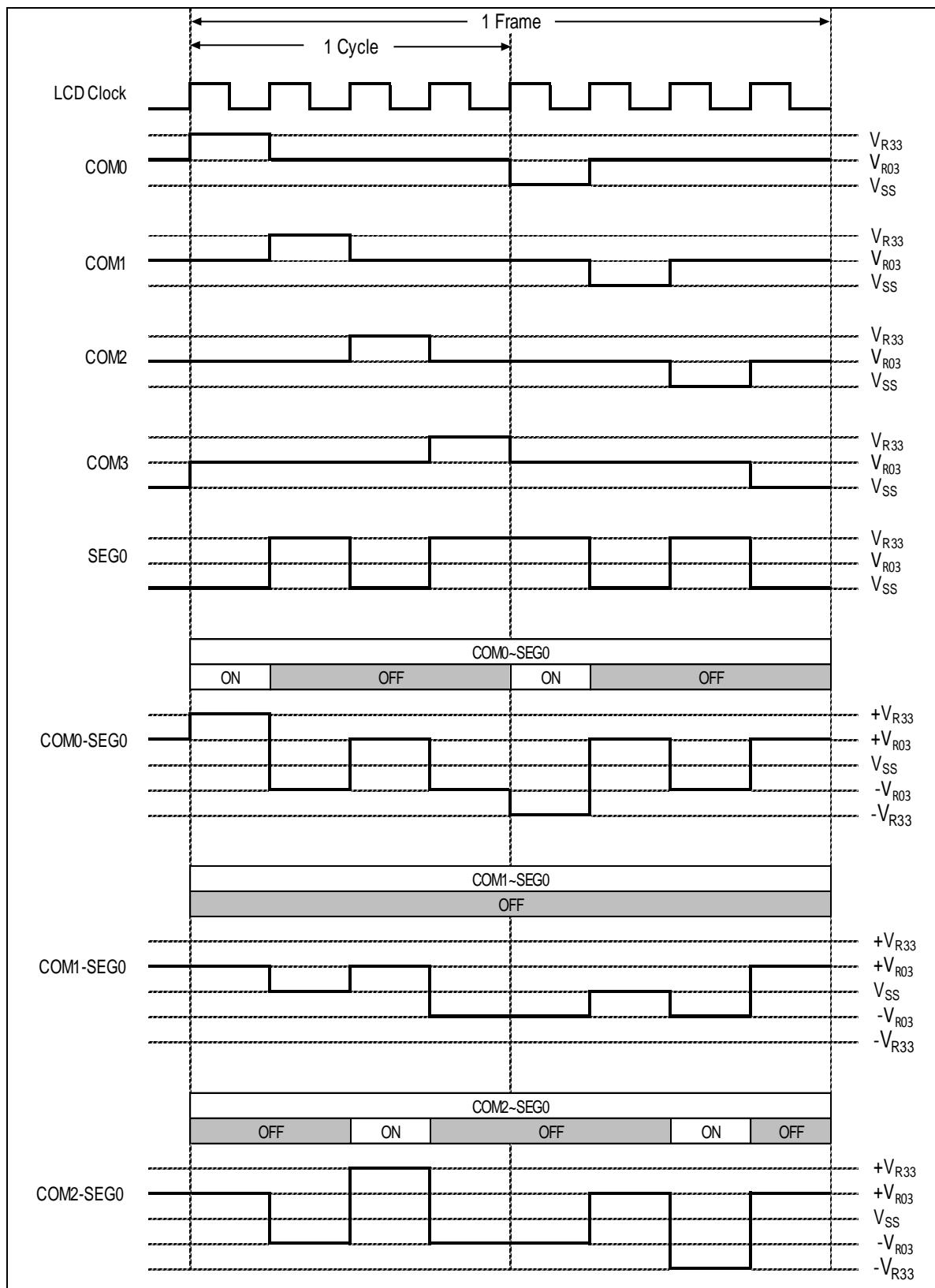


图 29.3-6 1/4 占空比, 1/2 偏压, B 类波形



29.3.1.4 静态模式输出波形举例

静态显示模式下，COM0 可以作为 common 信号输出引脚。偏压和输出波形类型不可选择。

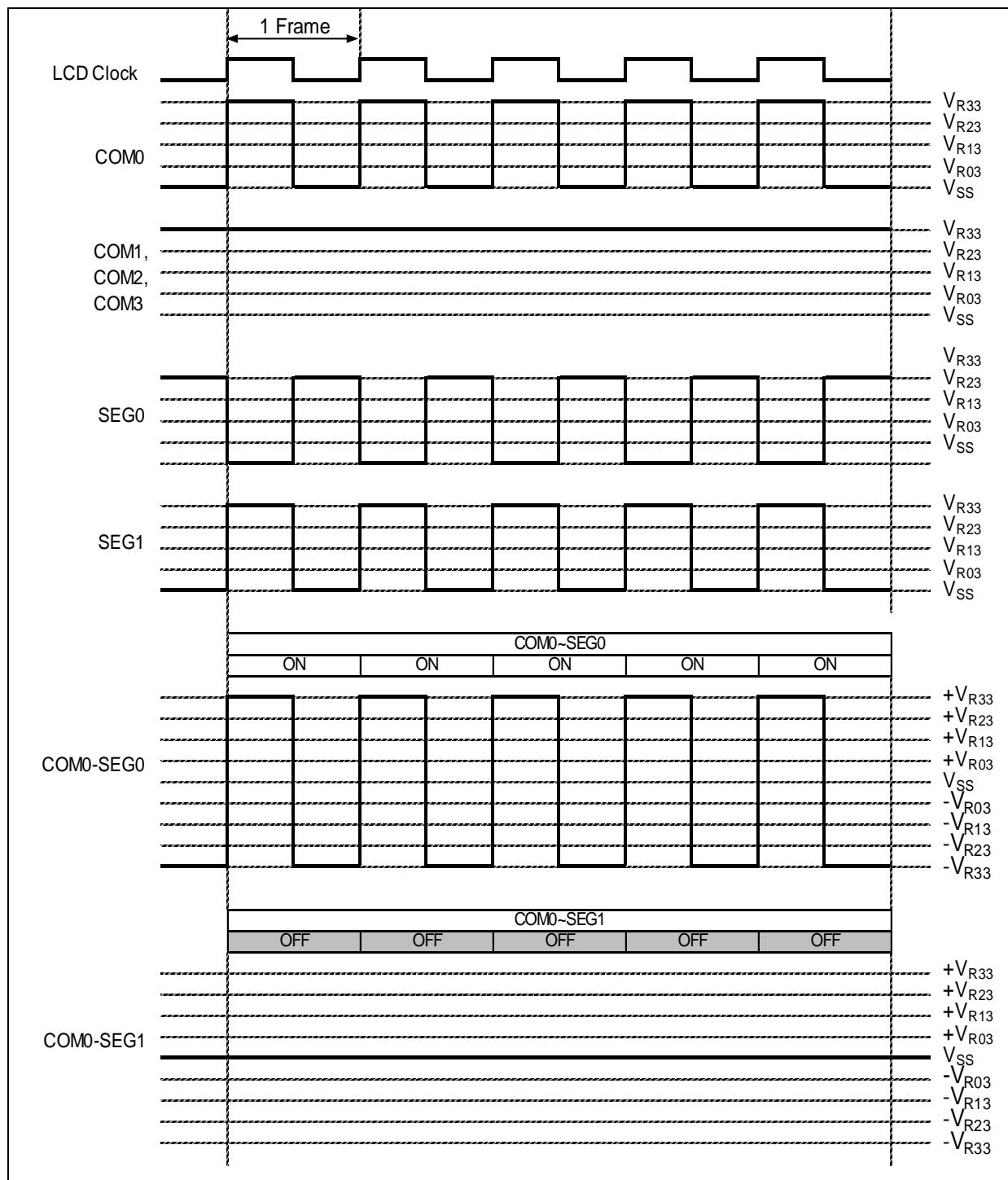
图 29.3-7 是静态显示模式的输出波形，表 29.3-6 是 LCDRAM00 和 LCDRAM01 的数据。

COM0 和 SEG0 之间的电压差最大，所以二者之间的液晶点亮；COM0 和 SEG1 之间的电压差为 0，所以二者之间的液晶熄灭。

表 29.3-6 静态模式下 LCDRAM00 和 LCDRAM01 中的数据

SEG0	LCDRAM00							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	-	-	-	-	-	-	-	1
SEG1	LCDRAM01							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	-	-	-	-	-	-	-	0

图 29.3-7 静态模式输出波形



29.3.2 显示数据存储

不同显示模式下，LCD 显示数据存储寄存器（LCDRAM）的可用空间如下：

- 1/8 占空比模式下，存储空间最大为 36×8 bits
- 1/6 占空比模式下，存储空间最大为 38×6 bits
- 1/4 占空比模式下，存储空间最大为 40×4 bits
- 1/3 占空比模式下，存储空间最大为 40×3 bits
- 1/2 占空比模式下，存储空间最大为 40×2 bits
- 静态显示模式下，存储空间最大为 40×1 bit

LCDRAM 的数据与 common 信号同步读出，并输出到 segment 引脚。例如，1/4 占空比模式下，COM0 有效时 LCDRAM 的 bit0 一起读出并分别赋给 SEG0 ~ SEG39；COM1 有效时 LCDRAM 的 bit1 一起读出并分别赋给 SEG0 ~ SEG39。

LCDRAM 中写“1”对应的 SEG 将在屏幕上显示，LCDRAM 中写“0”对应的 SEG 将不显示。

LCD 控制器开始工作后，将不受 CPU 干预。没有用作 LCD 功能的引脚可以作为通用输入输出端口使用，此时没有用到 LCDRAM 可以作为普通数据寄存器使用。

图 29.3-8 是不同占空比条件下 common/segment 和 LCDRAM 各位的关系图，每个 LCDRAM 对应一个 SEG。

以 1/8 占空比模式为例，LCDRAM00 ~ LCDRAM35 的所有位都有效。其中，bit0 在 COM0 有效时读取，bit1 在 COM1 有效时读取，以此类推。

图 29.3-8 不同占空比时 common/segment 和 LCDRAM 各位的关系

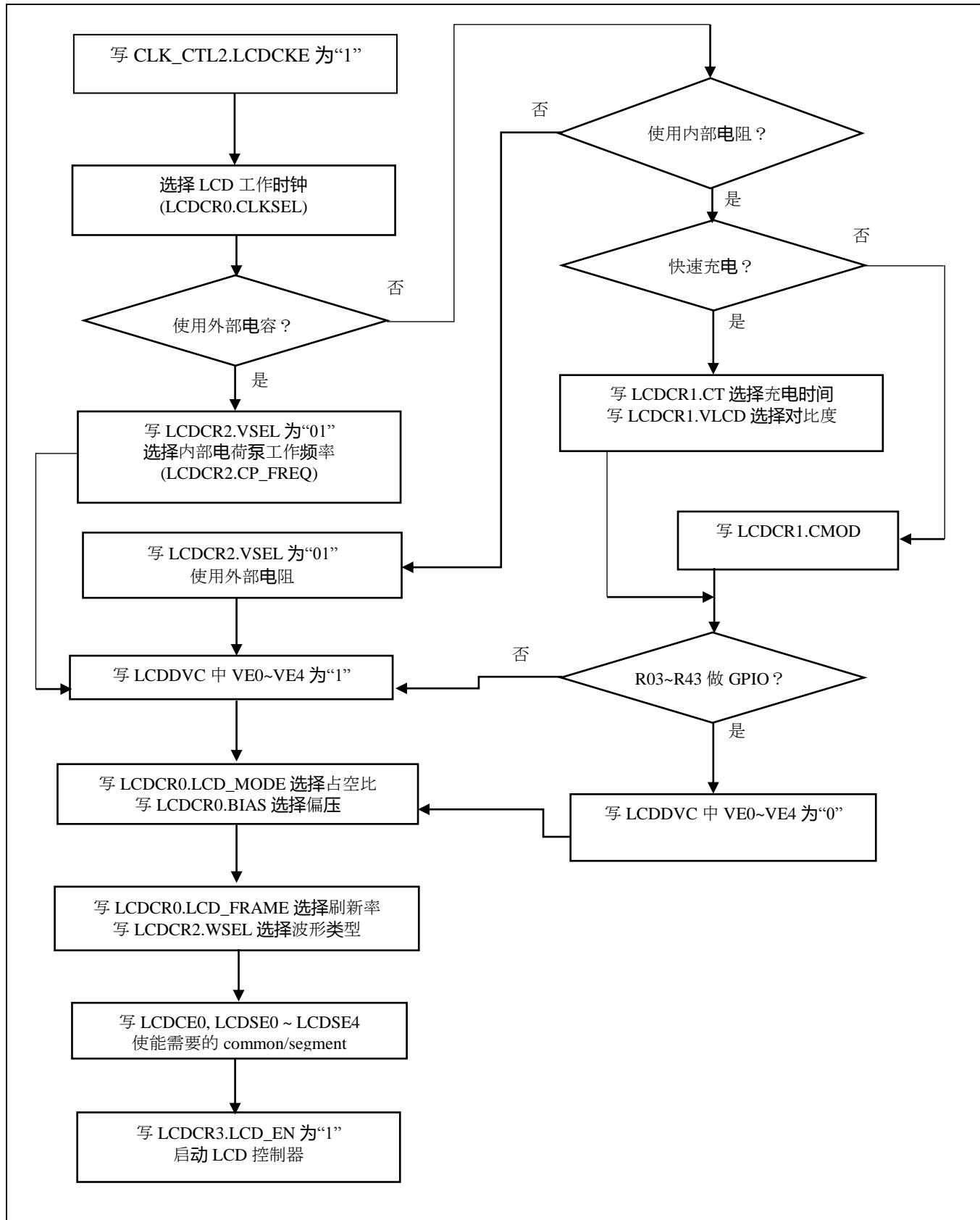
LCDRAM00	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG0
LCDRAM01	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG1
LCDRAM02	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG2
.
LCDRAM36	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG36
LCDRAM37	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG37
LCDRAM38	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG38
LCDRAM39	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SEG39
	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	

The diagram illustrates the mapping of LCDRAM bits to common pins for different duty cycles. Arrows point from each bit column to its corresponding common pin. Annotations indicate the effective pins for static mode and various duty cycle ratios:

- 静态模式的有效端口 (Static mode effective pins): COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7.
- 1/2 占空比的有效端口 (1/2 duty cycle effective pins): COM1, COM2.
- 1/3 占空比的有效端口 (1/3 duty cycle effective pins): COM2, COM3.
- 1/4 占空比的有效端口 (1/4 duty cycle effective pins): COM3, COM4.
- 1/6 占空比的有效端口 (1/6 duty cycle effective pins): COM4, COM5.
- 1/8 占空比的有效端口 (1/8 duty cycle effective pins): COM5, COM6.

29.4 设置步骤图

图 29.4-1 LCD 控制器设置步骤



29.5 寄存器

表 29.5-1 LCD 控制器寄存器列表

缩写	寄存器名称	章节
LCDCR0	LCD控制寄存器0	29.5.1
LCDCR1	LCD控制寄存器1	29.5.2
LCDCR2	LCD控制寄存器2	29.5.3
LCDCR3	LCD控制寄存器3	29.5.4
LCDDVC	LCD驱动电压控制寄存器	29.5.5
LCDCE0	LCD COM输出使能寄存器	29.5.6
LCDSE0~LCDSE4	LCD SEG输出使能寄存器0 ~ 4	29.5.7
LCDRAM00~LCDRAM39	LCD显示数据存储寄存器0 ~ 39	29.5.8

29.5.1 LCD 控制寄存器 0 (LCDCR0)

LCDCR0 是 LCD 的通用设置寄存器。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	LCD_MODE[2:0]			BIAS[1:0]		LCD_FRAME[1:0]		CLKSE_L
属性	R/W			R/W		R/W		R/W
初始值	000			00		00		0

寄存器功能

[bit7:5] LCD_MODE[2:0]: LCD 显示模式选择

Bit2:0	说明
000	静态显示 (初始值)
001	1/2 占空比
010	1/3 占空比
011	1/4 占空比
100	禁止设定, 写无效
101	1/6 占空比
110	禁止设定, 写无效
111	1/8 占空比

[bit4:3] BIAS[1:0]: 偏压选择

Bit1:0	说明
00	1/2 偏压 (初始值)
01	1/3 偏压
10	1/4 偏压
11	禁止设定, 写无效

注意：

- 偏压和占空比的对应关系，请参考表 29.3-1。

[bit2:1] LCD_FRAME[1:0]: LCD 刷新频率选择

Bit1:0	说明
00	128Hz (初始值)
01	256Hz
10	512Hz
11	1kHz

[bit0] CLKSEL: LCD 工作时钟选择

Bit	说明
0	外部低速晶振 (LSXT) 作为LCD工作时钟 (初始值)
1	内部低速RC (LICR) 作为LCD工作时钟

29.5.2 LCD 控制寄存器 1 (LCDCR1)

LCDCR1 是内部电阻模式时的设置寄存器。外部电阻模式和外部电容模式下，可以忽略本寄存器。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留	CMOD	CT[2:0]			VLCD[2:0]		
属性	-	R/W	R/W			R/W		
初始值	-	0	000			111		

寄存器功能

[bit7] 保留位，读数据为 0，写无效。

[bit6] CMOD: LCD 充电模式选择

Bit	说明
0	快速充电模式 (初始值)
1	大电流充电模式

注意：

- 在 B 类波形下，使用大电流充电模式可以得到较好的显示效果。

[bit5:3] CT[2:0]: 快速充电模式下，大电流充电时间选择

Bit2:0	说明
000	1个LCD工作时钟周期 (初始值)
001	2个LCD工作时钟周期
010	3个LCD工作时钟周期
011	4个LCD工作时钟周期
100	5个LCD工作时钟周期
101	6个LCD工作时钟周期
110	7个LCD工作时钟周期

111	8个LCD工作时钟周期
-----	-------------

[bit2:0] VLCD[2:0]: 对比度选择

Bit2:0	说明
000	60%
001	65%
010	70%
011	75%
100	80%
101	85%
110	90%
111	100% (初始值)

注意：

- 仅可在大电流充电模式下改变对比度，否则会影响显示效果。

29.5.3 LCD 控制寄存器 2 (LCDCR2)

LCDCR2 是 LCD 的通用设置寄存器。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	WSEL	VSEL[1:0]		保留			CP_FREQ[1:0]	
属性	R/W	R/W		-			R/W	
初始值	1	00		-			00	

寄存器功能

[bit7] WSEL: LCD 驱动波形选择

Bit	说明
0	A类波形
1	B类波形 (初始值)

[bit6:5] VSEL[1:0]: LCD 驱动电压来源选择

Bit1:0	说明
00	内部电阻 (初始值)
01	外部电容
10	外部电阻
11	禁止设定，写无效

[bit4:2] 保留位，读数据为 0，写无效。

[bit1:0] CP_FREQ[1:0]: 外部电容模式下，内部电荷泵工作频率选择

Bit1:0	说明
00	2KHz (初始值)
01	4KHz
10	8KHz
11	16KHz

29.5.4 LCD 控制寄存器 3 (LCDCR3)

LCDCR3 是 LCD 的通用设置寄存器。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留		LCD_BUF[1:0]		保留			LCD_EN
属性	-		R/W		-			R/W
初始值	-		00		-			0

寄存器功能

[bit7:6] 保留位,读数据为 0, 写无效。

[bit5:4] LCD_BUF[1:0]: COM/SEG 引脚驱动能力选择

Bit1:0	说明
00	正常驱动能力 (初始值)
01	2倍驱动能力
10	3倍驱动能力
11	4倍驱动能力

[bit3:1] 保留位,读数据为 0, 写无效。

[bit0] LCD_EN: LCD 工作使能位

Bit	说明
0	停止LCD控制器, LCD屏幕显示关闭
1	启动LCD控制器, LCD屏幕显示开启

29.5.5 LCD 驱动电压控制寄存器 (LCDDVC)

LCDDVC 是驱动电压输入引脚 R03 ~ R43 功能选择寄存器

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	保留			VE4	VE3	VE2	VE1	VE0
属性	-			R/W	R/W	R/W	R/W	R/W
初始值	-			0	0	0	0	0

寄存器功能

[bit7:5] 保留位,读数据为 0, 写无效。

[bit4] VE4: R43 引脚功能选择位

Bit	说明
0	R43作为通用输入输出端口 (GPIO) (初始值)
1	R43作为LCD驱动电压输入端口

[bit3] VE3: R33 引脚功能选择位

Bit	说明
0	R33作为通用输入输出端口（GPIO）（初始值）
1	R33作为LCD驱动电压输入端口

[bit2] VE2: R23 引脚功能选择位

Bit	说明
0	R23作为通用输入输出端口（GPIO）（初始值）
1	R23作为LCD驱动电压输入端口

[bit1] VE1: R13 引脚功能选择位

Bit	说明
0	R13作为通用输入输出端口（GPIO）（初始值）
1	R13作为LCD驱动电压输入端口

[bit0] VE0: R03 引脚功能选择位

Bit	说明
0	R03作为通用输入输出端口（GPIO）（初始值）
1	R03作为LCD驱动电压输入端口

注意：

- 仅在内部电阻模式下，R03 ~ R43 可以作为 GPIO。
- 外部电阻模式和外部电容模式下，R03 ~ R43 不可作为 GPIO，VE0 ~ VE4 必须写“1”。

29.5.6 LCD COM 输出使能寄存器 (LCDCE0)

LCDCE0 是引脚 COM0 ~ COM7 功能选择寄存器。可以作为 LCD common 信号输出或 PIO。

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	COME 7	COME 6	COME 5	COME 4	COME 3	COME 2	COME 1	COME 0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7:6] COME[7:6]: COM/SEG 双功能端口输出选择位

仅在 1/8 占空比模式下写有效，其他模式下写无效。

非 1/8 占空比模式下，引脚 COM7 和 COM6 作为 GPIO 或 SEGxx，取决于寄存器 LCDSE4。

Bit	说明
0	1/8占空比模式下：COM7和COM6的common信号输出关闭（初始值）
1	1/8占空比模式下：COM7和COM6输出common信号

[bit5:4] COME[5:4]: COM/SEG 双功能端口输出选择位

仅在 1/8 或 1/6 占空比模式下写有效，其他模式下写无效。

非 1/8 或 1/6 占空比模式下，引脚 COM5 和 COM4 作为 GPIO 或 SEGxx，取决于寄存器 LCDSE4。

Bit	说明
0	1/8或1/6占空比模式下，COM5和COM4的common信号输出关闭（初始值）
1	1/8或1/6占空比模式下，COM5和COM4输出common信号

[bit3:0] COME[3:0]: COM3 ~ COM0 端口输出选择位

Bit	说明
0	COMn的common信号输出关闭（初始值）
1	COMn输出common信号

其中，n=0~3。

例如：COME0 控制引脚 COM0 的功能选择

29.5.7 LCD SEG 输出使能寄存器 (LCDSE0 ~ LCDSE4)

LCDSE0 ~ LCDSE4 是引脚 SEG0~SEG39 功能选择寄存器。可以作为 LCD segment 信号输出或 GPIO。

寄存器配置

LCDSE4:

位	7	6	5	4	3	2	1	0
字段名	SEGE3 9	SEGE3 8	SEGE3 7	SEGE3 6	SEGE3 5	SEGE3 4	SEGE3 3	SEGE3 2
属性	R/W							
初始值	0	0	0	0	0	0	0	0

LCDSE3:

位	7	6	5	4	3	2	1	0
字段名	SEGE3 1	SEGE3 0	SEGE2 9	SEGE2 8	SEGE2 7	SEGE2 6	SEGE2 5	SEGE2 4
属性	R/W							
初始值	0	0	0	0	0	0	0	0

LCDSE2:

位	7	6	5	4	3	2	1	0
字段名	SEGE2 3	SEGE2 2	SEGE2 1	SEGE2 0	SEGE1 9	SEGE1 8	SEGE1 7	SEGE1 6
属性	R/W							
初始值	0	0	0	0	0	0	0	0

LCDSE1:

位	7	6	5	4	3	2	1	0
字段名	SEGE1 5	SEGE1 4	SEGE1 3	SEGE1 2	SEGE1 1	SEGE1 0	SEGE9	SEGE8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初始值	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

LCDSE0:

位	7	6	5	4	3	2	1	0
字段名	SEGE7	SEGE6	SEGE5	SEGE4	SEGE3	SEGE2	SEGE1	SEGE0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

Bit	说明
0	SEGn的segment信号输出关闭（初始值）
1	SEGn输出segment信号

其中， n=0~39。

例如： SEGE0 控制引脚 SEG0

29.5.8 LCD 显示数据存储寄存器（LCDRAM00 ~ LCDRAM39）

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	LCDRAMxx							
属性	R/W							
初始值	0x00							

注意：

- 寄存器 LCDRAMxx 中各位与 COM/SEG 的关系，请参考“显示数据存储”一节。

29.6 注意事项

外部低速晶振的稳定等待时间结束后，再将 LCD 工作时钟切换到外部低速晶振（LCDR0.CLKSEL）。

如果要改变 LCD 的占空比、偏压等设置，必须首先停止 LCD 控制器（写 LCDCR3.LCD_EN 为“0”）。禁止在 LCD 运行过程中改变设置。

30. 电压比较器

本章介绍了电压比较器的功能以及操作方法。

30.1 概要

以下介绍了电压比较器的概要。

电压比较器的功能

电压比较器是一个可以单独进行电压比较的功能模块，用来监测和比较模拟输入电压(正端)和参考输入电压(负端)。

当正端输入电压高于负端输入电压时，电压比较器输出高电平；反之，则输出低电平。

正端输入电压可以从外部端口选择输入，负端输入电压可以选择从外部端口或者内部 BGR 参考电压输入。

特征

- 支持电压监测功能
- 支持负端输入电压从外部端口或者内部 BGR 参考电压输入
- 支持比较器输入迟滞功能
- 支持比较器输出滤波功能
- 支持电压转换功能，可设置 64 档
- 支持比较器结果输出到端口
- 电压比较器输出可作为复合定时器的触发输入
- 以下三种条件可以产生中断：
 - 电压比较器输出高电平
 - 电压比较器输出上升沿
 - 电压比较器输出下降沿

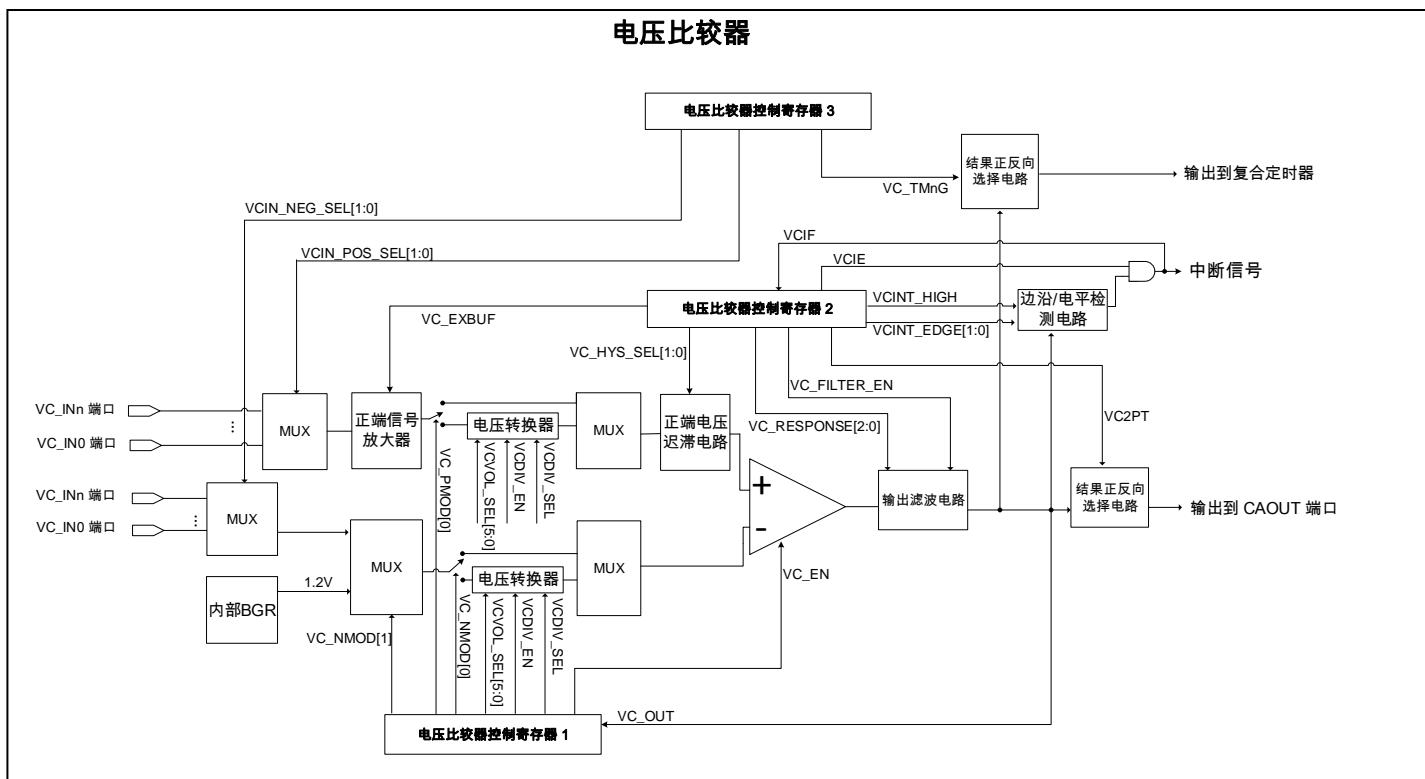
30.2 配置

以下介绍了电压比较器的配置。

电压比较器的框图

电压比较器的框图如图 30.2-1 所示。

图 30.2-1 电压比较器的框图



电压比较器

电压比较器的功能是监测并比较 2 路模拟输入电压然后输出比较结果。其中正端模拟电压可以从外部 VC_INx 端口输入，负端模拟参考电压可以从以下两种电压选择：外部 VC_INx 端口或者内部 BGR 参考电压。

边沿/电平检测电路

边沿/电平检测电路的功能是监测电压比较器的输出结果是上升沿，下降沿或者高电平以便产生中断信号。

电压转换器

通过电压转换器可以把输入电压进行分压处理，正端电压和负端电压都可进行电压转换。

正端电压迟滞电路

当使能迟滞电路时，电压比较器会输出迟滞后的比较结果。

输出滤波电路

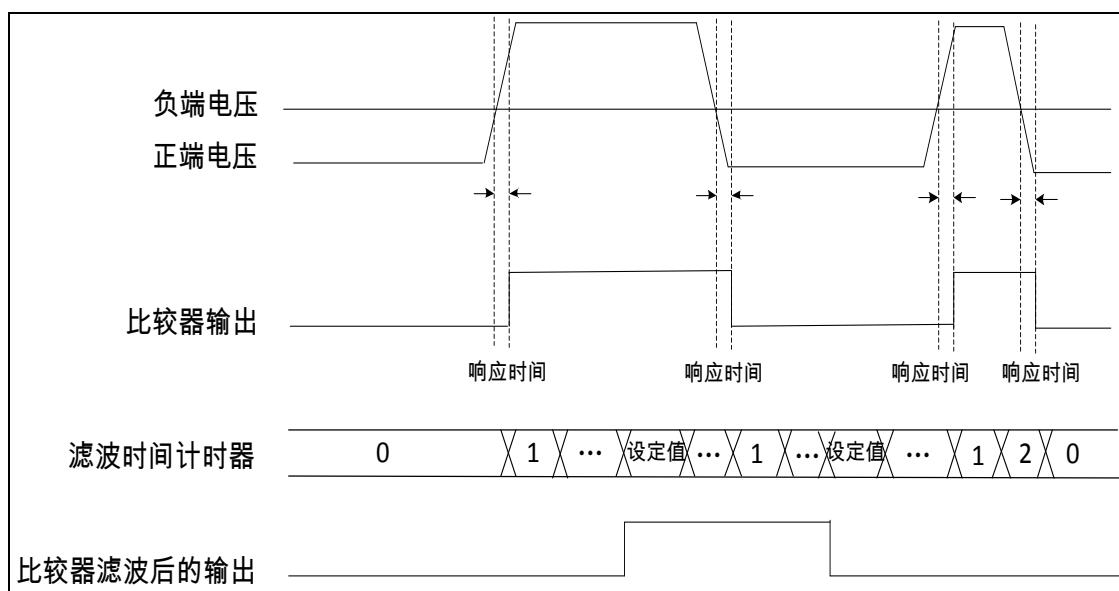
电压比较器的输出可以通过滤波电路进行滤波处理，滤波时间可以由寄存器配置。

30.3 操作以及设定步骤示例

30.3.1 电压比较器的滤波响应时间

当输入电压变化时，电压比较器的结果会经过一段响应时间再改变。除此之外，还可以通过配置电压比较器控制寄存器 2 中的 VC_FILTER_EN 位来使能滤波功能，并通过配置 VC_RESPONSE[2:0]位来选择不同的滤波时间。设置了滤波时间后，如果电压比较器的输入在设定时间内有变化，比较器的输出不会改变。

图 30.3-1 电压比较器滤波响应时间



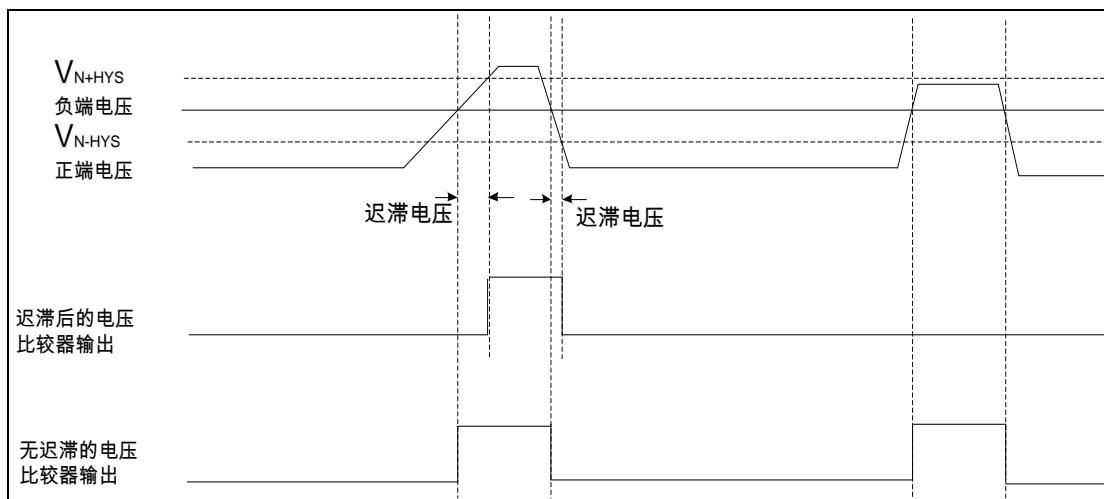
*具体的响应时间参考本系列的“数据手册”。

30.3.2 电压比较器迟滞功能

电压比较器迟滞功能可以通过电压比较器控制寄存器 2 中的 VC_HYS_EN 位进行使能。

迟滞功能使能后，当正端输入电压高于负端输入电压+迟滞电压(V_N+HYS)时，电压比较器输出高；当正端输入电压低于负端输入电压-迟滞电压(V_N-HYS)时，电压比较器输出低。

图 30.3-2 电压比较器迟滞功能



30.3.3 设定步骤示例

以下介绍了设定步骤示例，本示例选择外部端口输入作为正端电压输入，内部 BGR 作为负端参考电压输入。

电压比较器的设定 (正端电压从外部输入，负端电压从内部 BGR 输入)

图 30.3-3 电压比较器的设定

15	14	13	8
VC_CTL1	VCDIV_SEL	VCDIV_EN	VCVOL_SEL
0	0		100000

7	6	5	4	3	2	1	0
VC_CTL1	-	VC_PMOD	VC_NMODY	VC_OUT	VC_EN		
-		00	10		Δ	0	

15	13	12	11	10	9	8
VC_CTL2	-	VC_HYS_SEL	VCIF	VCIE	VC2PT	
000		00	Δ	0	0	

7	6	5	4	3	2	1	0
VC_CTL	VCINT_HI	VCINT_E	VC_EX	VC_RESPONSE	VC_FILTER		
2	GH	DGE	BUF	000		0	

VC_CTL3	7	6	5	4	3	2	1	0
	00		00		0	0	0	0

Δ:Bit to be used

0: Set it to “0”

1: Set it to “1”

注意:

- 当使能电压比较器后，需要等待建立时间，具体的建立时间请参考数据手册。

设定步骤示例

1. 使能 BGR(BGR_CTL:BGR_EN = 1)。
2. 等待 BGR 建立时间。
3. 清除中断标志位(VC_CTL2.VCIF=0)。
4. 如图 3-3 所示配置电压比较器控制寄存器。

(正端电压:外部端口 VC_IN0 输入电压,

负端电压:内部 BGR 1.2V 参考电压

检测到上升沿产生中断)

如果需要，使能电压比较器中断请求(VC_CTL2:VCIE = 1)。

5. 使能电压比较器(VC_CTL1:VC_EN = 1)。
6. 等待电压比较器建立时间。
7. 读取结果

30.4 寄存器

以下介绍了电压比较器的寄存器功能。

30.4.1 控制寄存器列表

以下是电压比较器控制寄存器列表。

表 30.4-1 电压比较器控制寄存器列表

寄存器简称	寄存器名称	参照章节
VC_CTL1	电压比较器控制寄存器 1	30.4.2
VC_CTL2	电压比较器控制寄存器 2	30.4.3
VC_CTL3	电压比较器控制寄存器 3	30.4.4

30.4.2 电压比较器控制寄存器-1(VC_CTL1)

寄存器配置

位	15	14	13				8
字段名	VCDIV_SEL	VCDIV_EN		VCVOL_SEL			
属性	R/W	R/W		R/W			
初始值	0	0		100000			

位	7	6	5	4	3	2	1	0
字段名	-		VC_PMOD		VC_NMODY		VC_OUT	VC_EN
属性	-		R/W		R/W		R	R/W
初始值	00		00		00		0	0

寄存器功能

[31:16] 保留，读值未定，写操作无影响。

[15] VCDIV_SEL:电压转换源选择位

位	功能
0	正端输入电压进行电压转换(初始值)
1	负端输入电压进行电压转换

[14] VCDIV_EN:电压转换器使能位

位	功能
0	关闭电压转换器(初始值)
1	使能电压转换器

[13:8] VCVOL_SEL[5:0]: 通过电压转换器转换后的电压选择位

本位可以选择当电压转换器使能后，经过电压转换器分压后的电压。

位5:0	功能
000000	设置转换后电压为1/64
000001	设置转换后电压为2/64
000010	设置转换后电压为3/64
.....
100000	设置转换后电压为33/64(初始值)
.....
111101	设置转换后电压为62/64
111110	设置转换后电压为63/64
111111	设置转换后电压为64/64

[7:6] 保留，读值未定，写操作无影响。

[5:4] VC_PMODY[1:0]: 正端电压选择位，本位选择正端电压源是否需要经过电压转换。

位1:0	功能
00	外部输入电压不经过转换器作为比较器正端电压(初始值)。
01	外部输入电压经过转换器作为比较器正端电压
10	保留
11	保留

[3:2] VC_NMOD[1:0]: 负端电压选择位, 本位选择负端电压源:外部端口或者内部 BGR 参考电源; 同时选择是否需要经过电压转换。

位1:0	功能
00	外部参考输入电压不经过转换器作为比较器负端电压(初始值)。
01	外部参考输入电压经过转换器作为比较器负端电压。
10	内部BGR参考输入电压不经过转换器作为比较器负端电压。
11	内部BGR参考输入电压经过转换器作为比较器负端电压。

[1] VC_OUT: 输出状态位, 本位表明电压比较器的输出结果状态。

位	功能
写	写值到 VC_OUT无影响。
读 "0"	表明电压比较器输出"低"(正端电压 < 负端电压). (初始值)
读 "1"	表明电压比较器输出"高"(正端电压 > 负端电压).

[0] VC_EN: 电压比较器使能位, 本位使能电压比较器。

位	功能
0	关闭电压比较器 (初始值)
1	使能电压比较器

30.4.3 电压比较器控制寄存器 2(VC_CTL2)

寄存器配置

位	31		13	12	11	10	9	8
字段名	保留			VC_HYS_SEL			VCIF	VCIE
属性	-			R/W			R/W	R/W
初始值	-			00			0	0
位	7	6	5	4	3	2	1	0
字段名	VCINT_HIGH	VCINT_EDGE[1:0]			保留	VC_RESPONSE[2:0]		
属性	R/W	R/W			-	R/W		
初始值	0	00			-	000		

寄存器功能

[31:13] 保留位,读值未定,写操作无影响。

[12:11] VC_HYS_SEL[1:0]: 电压比较器迟滞电压选择位,本位选择电压比较器迟滞电压。

位 1:0	功能
00	无迟滞(初始值)
01	使能迟滞功能,迟滞电压为 10mV
10	使能迟滞功能,迟滞电压为 20mV
11	使能迟滞功能,迟滞电压为 30mV

[10] VCIF: 电压比较器中断标志位

位	功能	
	读	写
0	未产生电压比较器中断(初始值)	中断标志位清零
1	产生电压比较器中断	无效且对其他寄存器无作用

注意:

- 软件清零与硬件置位同时发生时,硬件置位具有更高优先级。

[9] VCIE: 电压比较器中断请求使能位

位	功能
0	禁止产生电压比较器中断请求(初始值)
1	使能电压比较器中断请求

当 VCIF 为 “1”且 VCIE=“1”时,中断请求才会产生。

[8] VC2PT: 电压比较器结果输出选择位,本位选择电压比较器输出到端口的相位。

位	功能
0	电压比较器正向结果输出到端口(初始值).
1	电压比较器反向结果输出到端口

[7] VCINT_HIGH: 电压比较器输出电平/边沿检测选择位,本位选择电压比较器输出边沿/电平检测。

位	功能
0	边沿检测产生中断标志位(初始值)。
1	高电平检测产生中断标志位

[6:5] VCINT_EDGE[1:0]: 电压比较器输出上升沿/下降沿检测选择位, 本位选择电压比较器输出上升沿/下降沿检测。

位	功能
00	当检测到上升沿或者下降沿时不产生中断标志(初始值).
01	检测到下降沿时产生中断标志
10	检测到上升沿时产生中断标志
11	检测到上升沿或者下降沿时产生中断标志

[4] 保留位,读值未定, 写操作无影响。

[3:1] VC_RESPONSE[2:0]: 电压比较器滤波时间选择位, 本位选择电压比较器滤波时间。

位	功能
000	设置滤波时间为 13us(初始值).
001	设置滤波时间为 27us
010	设置滤波时间为 53us
011	设置滤波时间为 213us
100	设置滤波时间为 852us
101	设置滤波时间为 3.41ms
110	设置滤波时间为 13.64ms
111	设置滤波时间为 54.56ms

注意:

- 以上所列时间为标准条件下的时间, 具体请参考数据手册。

[0] VC_FILTER_EN: 电压比较器滤波使能位, 本位使能电压比较器滤波功能。

位	功能
0	禁止电压比较器滤波功能.(初始值)
1	使能电压比较器滤波功能.

30.4.4 电压比较器控制寄存器 3(VC_CTL3)

寄存器配置

位	31	保留						8
字段名		-						
属性		-						
初始值		-						
位	7	6	5	4	3	2	1	0
字段名	VCIN_POS_SEL[1:0]	VCIN_NEG_SEL[1:0]		保留	VC_TM2G	保留	VC_TM0G	
属性	R/W	R/W		-	R/W	-	R/W	
初始值	00	00		-	0	-	0	

寄存器功能

[31:8] 保留, 读值未定, 写操作无影响。

[7:6] VCIN_POS_SEL[1:0]: 电压比较器外部正端输入电压源选择位, 本位选择电压比较器的外部正端输入电压源。

位	功能
00	选择端口 VC_IN0 作为电压比较器正端输入(初始值).
01	选择端口 VC_IN1 作为电压比较器正端输入
10	选择端口 VC_IN2 作为电压比较器正端输入
11	选择端口 VC_IN3 作为电压比较器正端输入

[5:4] VCIN_NEG_SEL[1:0]: 电压比较器外部负端输入电压源选择位, 本位选择电压比较器的外部负端输入电压源。

位	功能
00	选择端口 VC_IN0 作为电压比较器负端输入(初始值).
01	选择端口 VC_IN1 作为电压比较器负端输入
10	选择端口 VC_IN2 作为电压比较器负端输入
11	选择端口 VC_IN3 作为电压比较器负端输入

[3] 保留, 读值未定, 写操作无影响.

[2] VC_TM2G: 复合定时器 2 计数触发信号选择位, 本位选择复合定时器 2 的计数触发信号。

位	功能
0	电压比较器结果直接作为复合定时器 2 的计数触发信号(初始值).
1	电压比较器取反结果作为复合定时器 2 的计数触发信号

[1] 保留, 读值未定, 写操作无影响。

[0] VC_TM0G: 复合定时器 0 计数触发信号选择位, 本位选择复合定时器 0 的计数触发信号。

位	功能
0	电压比较器结果直接作为复合定时器0的计数触发信号(初始值).
1	电压比较器取反结果作为复合定时器0的计数触发信号

31. 多功能串行通讯接口

本章概述多功能串行通讯接口。

接口模式

根据工作模式的设置，多功能串行通讯接口可以选择以下通讯模式：

- UART0 (异步标准串行接口)
- UART1 (异步多处理器接口)
- SPI (串行外设接口/时钟同步通讯)
- I2C (I2C 总线接口)

注意：

- 关于各种接口详情，参见各章的说明。

接口模式切换

通过各串口通讯时，应在启动通讯前使用表 31-1 所示串口模式寄存器 (MR) 设置操作模式。

表 31-1 接口模式切换

MODE2	MODE1	MODE0	模式
0	0	0	UART0 (异步标准串行接口)
0	0	1	UART1 (异步多处理器接口)
0	1	0	SPI (串行外设接口/时钟同步通讯)
1	0	0	I2C (I2C 总线接口)
其他值			禁止设置

注意：

- 如果一个串口正在用于发送或接收操作时切换操作模式，不能保证正确发送和接收。
- 切换当前操作模式时，发出可编程清除命令(CR:PCL=1)或禁用 I2C (SAMSK:EN=0)，然后继续切换操作模式。设置操作模式后，设置各寄存器。
- 禁止使用表 31-1 没有列举的设置。

32. UART (异步串行接口)

本章说明支持多功能串口工作模式 0 和 1 的 UART (异步串行接口) 功能。

32.1 UART (异步串行接口) 概要

UART (异步串行接口) 是用于与外部设备异步通信（启动/停止同步）的通用数据通信串口，支持双向通信功能（正常模式）和主机/从机通信功能（多处理器模式：同时支持主机模式和从机模式）。

UART (异步串行接口) 功能

		功能
1	数据	全双工缓冲器。
2	串行输入	在总线时钟执行 3 次过采样，接收值取决于采样值的多数。
3	传输形式	异步
4	波特率	专用波特率发生器(由 15 位重载计数器构成) 可以选择外部时钟作为重载计数器工作时钟 波特率补偿
5	数据长度	位 5 -9 位 (正常模式), 7/8 位 (多处理器模式)
6	信号方式	NRZ (非归零翻转码), 反向 NRZ
7	检出启动位	与启动位的下降沿同步 (在 NRZ 系统中) 与启动位上升沿同步 (在反向 NRZ 系统中)
8	检出接收错误	帧错误 溢出错误 奇偶校验错误(正常模式)
9	硬件流控制	基于 CTS/RTS 的自动发送/接收控制
10	中断请求	接收中断(接收完成后发生帧错误、溢出错误或奇偶校验错误) 发送中断 (发送数据为空, 发送总线空闲)
11	主控/从动通信功能 (多处理器模式)	使能一 (主机) 至多 (从机) 通信。 (支持主机和从机系统。)

32.2 中断

UART 产生收/发中断，以下情况可产生中断请求：

- 在接收数据寄存器(RXDR)中收到数据或发生数据接收错误时
- 发送数据从发送数据寄存器(TXDR)传输至发送移位寄存器，并启动数据发送时
- 发送总线空闲（无数据发送）时

UART 中断

表 32.2-1 所示为 UART 中断控制位和中断源之间的关系。

表 32.2-1 UART 中断控制位和中断源

中断类型	中断请求标志位	标志寄存器	工作模式		中断源	中断源使能位	清除中断请求标志的操作
			0	1			
接收	RDFF	SR			接收到单字节	CR:RXIE	读取接收数据寄存器(RXDR)中数据
	ORF	SR			溢出错误		将接收错误标志清除位(SR:RECLR)设置为 1
	FEF	SR			帧错误		
	PEF	SR		x	奇偶校验错误		
发送	TDEF	SR			发送数据寄存器为空	CR:TXIE	向发送数据寄存器(TXDR)写入数据
	TBIF	SR			无数据发送	CR:TBIE	

32.2.1 接收中断及标志设置时序

接收完成 (SR:RDFF=1) 或接收错误发生 (SR:PEF,ORF,FEF=1) 时产生接收中断。

接收中断及标志设置时序

检测至首个停止位时，接收数据储存在接收数据寄存器 (RXDR) 中。完成数据接收 (SR:RDFF=1) 或数据接收错误发生 (SR:PEF, ORF, FEF=1) 时，设置各标志。若使能接收中断 (SR:RXIE=1)，将发生接收中断。

注意：

- 若出现接收错误，接收数据寄存器 (RXDR) 中的数据无效。

图 32.2-1 RDFF (接收数据寄存器已满) 标志位的设置时序

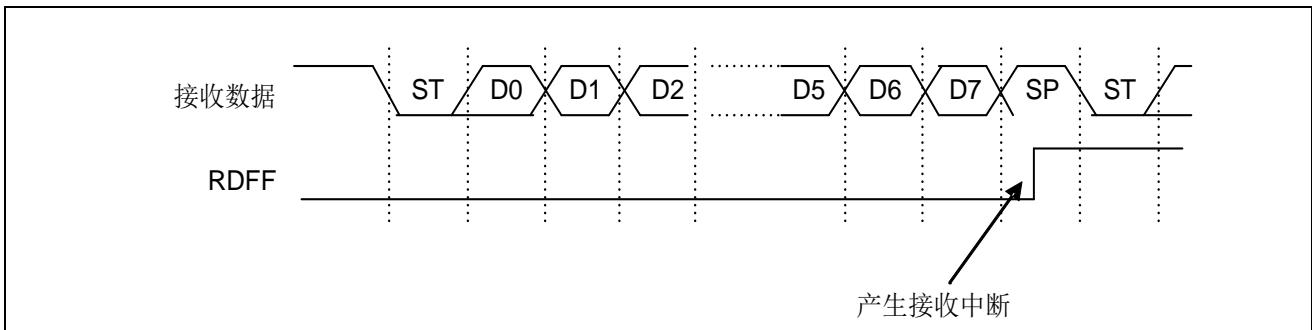
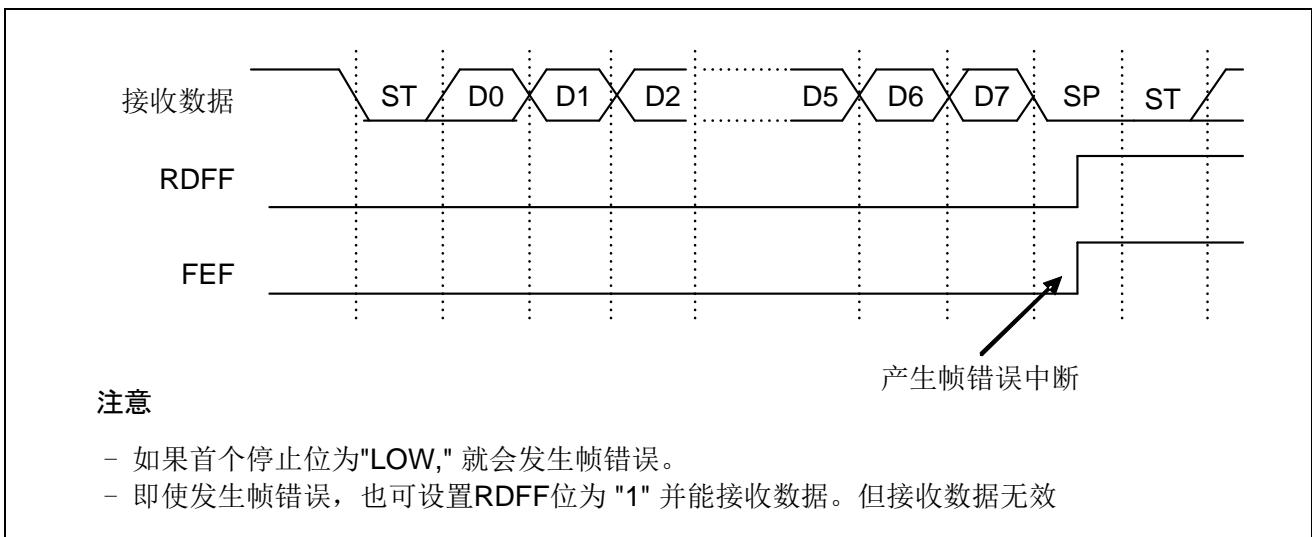


图 32.2-2 FEF (帧错误) 标志位设置时序

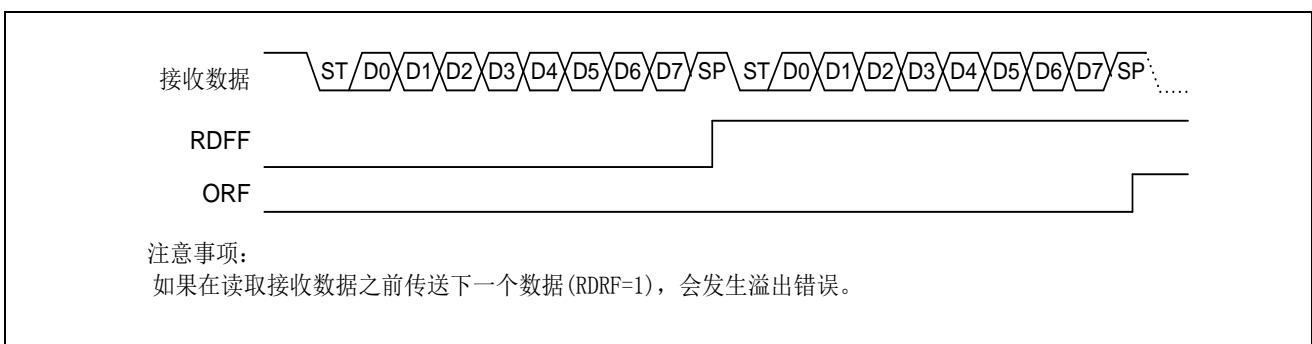


注意:

在接收进程中, 若在停止位采样点或在 1 到 2 个总线时钟之前检测到以下内容, 相关边沿将无效, 这可能会禁用下一个数据的正常接收。连续输出帧时, 帧之间需要充分的时间间隔。

- 串行数据下降沿 (ECR:INV=0 时)
- 串行数据上升沿 (ECR:INV=1 时)

图 32.2-3 ORF (溢出错误) 标志位设置时序



32.2.2 发送中断发生和标志设置时序

关于发送数据从发送数据寄存器 (TXDR) 传输至发送移位寄存器 (SR:TDEF = 1) 并启动发送，且未执行发送(SR:TBIF = 1) 时，发生发送中断。

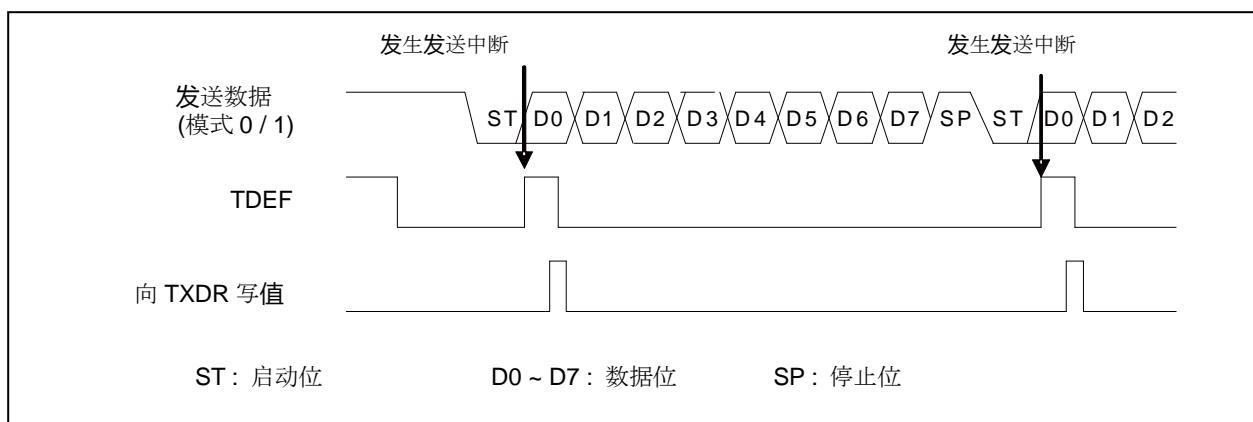
发送中断及标志设置时序

- 发送数据空标志 (SR:TDEF) 设置时序

数据从发送数据寄存器 (TXDR) 传输至发送移位寄存器后，可将下一个数据写入 TXDR (SR:TDEF = 1)。

若在此进程中使能发送中断 (CR:TXIE = 1)，将发生发送中断。由于 SR:TDEF 位为只读位，将数据写入发送数据寄存器 (TXDR) 时，SR:TDEF 位被清除为 0。

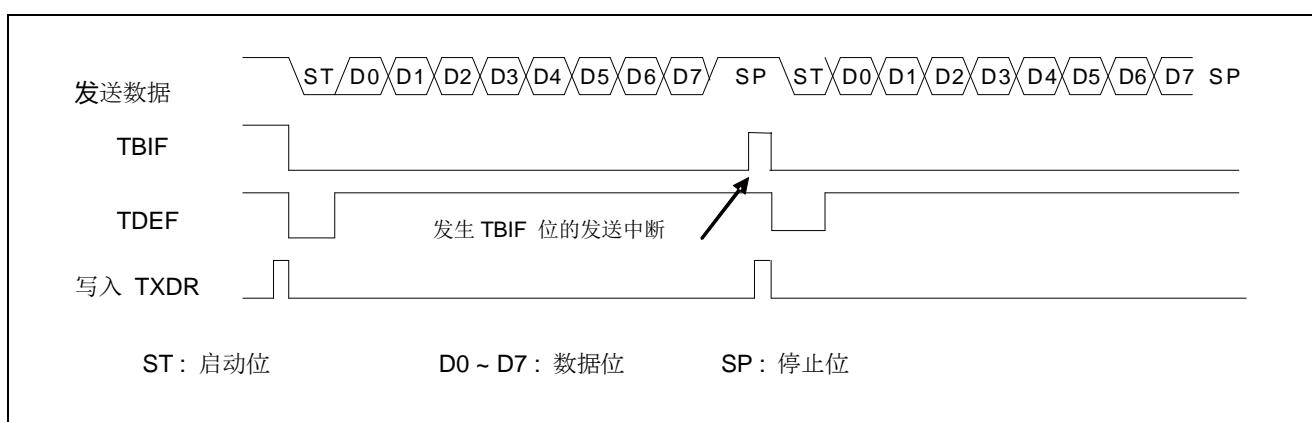
图 32.2-4 发送数据空标志(SR:TDEF) 的设置时序



- 发送总线空闲标志 (SR:TBIF) 设置时序

若发送数据寄存器为空 (SR:TDEF=1) 且没有数据发送，则 SR:TBIF 位将被设置为 1。若在此进程中使能发送总线空闲中断 (CR:TBIE = 1)，将发生发送中断。将发送数据写入发送数据寄存器 (TXDR) 时，将清除 SR:TBIF 位和发送中断请求。

图 32.2-5 发送总线空闲标志(TBIF) 的设置时序



32.3 UART 工作模式

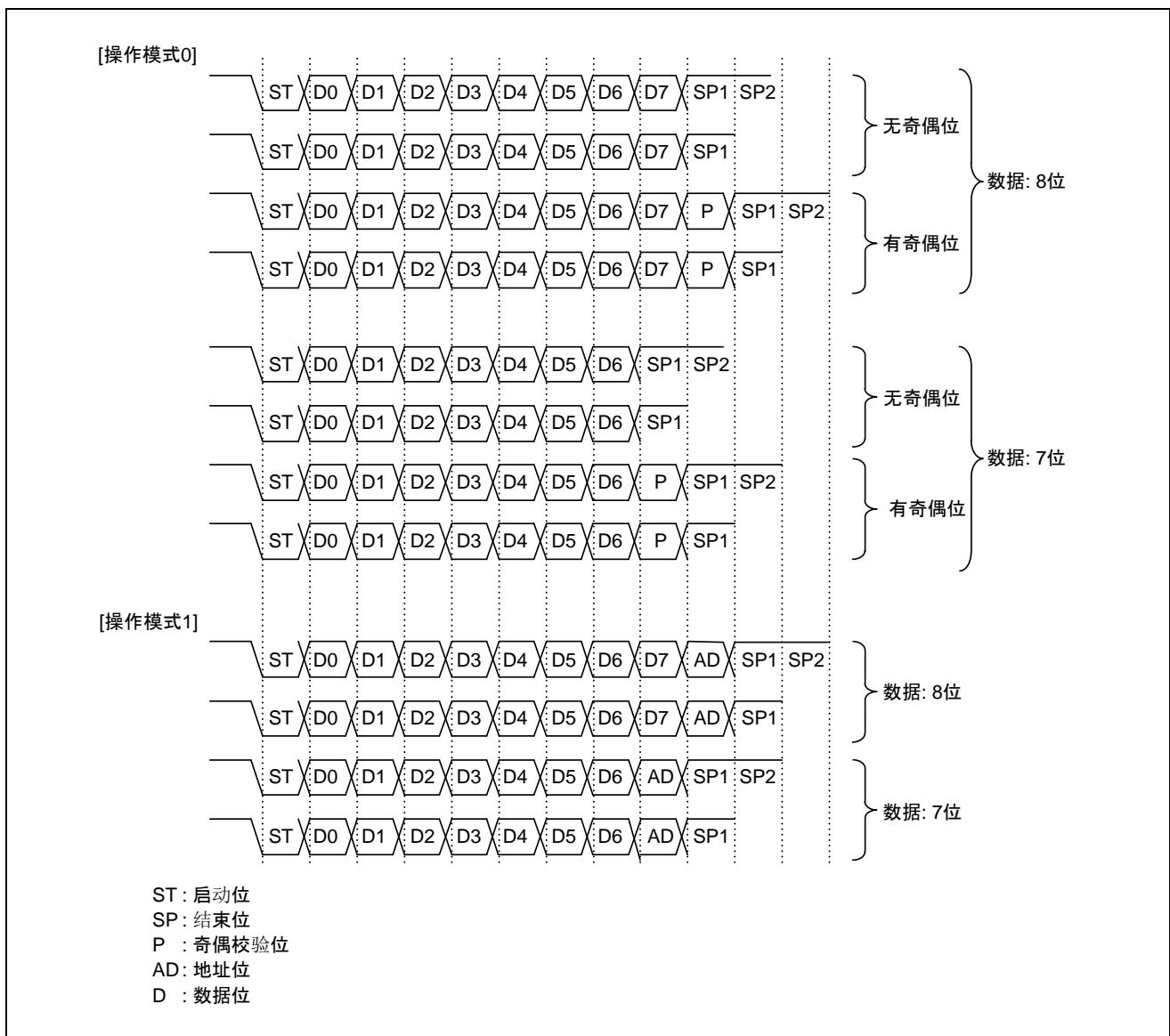
UART 执行模式 0 的双向串行异步通信,和模式 1 的主控/从动多处理器通信。

UART 操作

- 发送/接收数据的格式

- 发送/接收数据始终采用启动位启动, 接下来按指定数据位长度进行数据发送/接收, 并以至少一位长停止位结束。
- 串行模式寄存器 (MR) 的 TDS 位确定数据发送方向 (LSB 优先或 MSB 优先)。使用奇偶校验时, 将奇偶校验位始终放置在最后一个数据位和首个停止位之间。
- 在工作模式 0 (正常模式) 中, 可以选择使用或不使用奇偶校验。
- 在工作模式 1 (多处理器模式) 中, 不增加奇偶校验, 而是增加 AD 位。
- 图 32.3-1 所示为工作模式 0 和 1 的发送/接收数据格式。

图 32.3-1 发送/接收数据格式示例 (工作模式 0/1)



注意：

- 上图所示为数据长度设置为 7 位或 8 位时的格式。(在工作模式 0 中，数据长度可设置范围为 5 到 9 位。)
- 若串行模式寄存器 (MR) 的 TDS 位被设置为 1 (MSB 优先)，按照 D7, D6, D5, ... D1, 和 D0(P)的顺序处理此位。
- 若数据长度设置为 X 位，使能发送/接收数据寄存器 (TXDR/RXDR) 的下 X 位。

● 数据发送

- 串行状态寄存器 (SR) 的发送数据空标志位 (TDEF) 为 1 时，可将发送数据写入发送数据寄存器(TXDR)。

- 将发送数据写入发送数据寄存器 (TXDR) 时，发送数据空标志位 (SR:TDEF) 设置为 0。
- 将串行控制寄存器 (CR:TXE) 的发送使能位设置为 1，将发送数据载入发送移位寄存器，然后通过启动位按顺序启动发送。
- 启动发送时，再次将发送数据空标志位 (SR:TDEF) 设置为 1。若使能发送中断 (CR:TXIE=1)，将产生发送中断。在中断处理进程中，下一个发送数据设置可写入发送数据寄存器。

注意：

- 由于发送数据空标志位 (SR:TDEF) 的初始设置为 1，使能发送中断 (CR:TXIE) 时将立即发生发送中断。

- 数据接收

- 使能接收 (CR:RXE=1) 时，接口执行接收。
- 如果检出起始位，则根据设定到扩展通信控制寄存器(ESCR:PEN, P, DL2, DL1, DL0)和串行模式寄存器(SMR:BDS)的数据格式接收 1 帧数据。起始位的检出条件为:经过噪声滤波器(对串行数据输入在总线时钟进行 3 次采样，取决于多数)处理后，检出下降沿 (ESCR:INV="0"时)或上升沿(ESCR:INV="1"时)，在采样点检测出经过滤波处理的数据为 "L"。
- 完成单帧接收后，接收数据满标志位 (SR:RDFF) 被设置为 1。若使能接收中断 (CR:RXIE=1)，将产生接收中断。
- 要读取接收数据，在完成单帧数据接收后对接收数据进行读取，并检查串行状态寄存器 (SR) 的错误标志状态。若发生接收错误，则需要处理。
- 读取接收数据，将接收数据满标志位 (SR:RDFF) 清除为 0。

注意：

- 接收数据寄存器已满标志位 (SR:RDFF) 设置为 1 且未发生任何接收错误 (SR:PEF, ORF, FEF=0) 时，接收数据寄存器 (RXDR) 中的数据有效。
- 尽管配置有噪声滤波器（通过总线时钟三次采样串行数据输入后，采用择多值），若有噪声经过滤波器，可能接收到错误的数据。解决办法是板设计时，不允许噪声经过此滤波器或执行通信时经过滤波器的噪声不会造成任何问题（例如，增加结束时数据总量的检验并在发生错误时再次发送数据）。
- 在接收进程中，若在停止位采样点或在 1 到 2 个总线时钟之前检测到以下内容，相关边沿将无效，这可能会禁用下一个数据的正常接收。连续输出帧时，帧之间需要充

分的时间间隔。

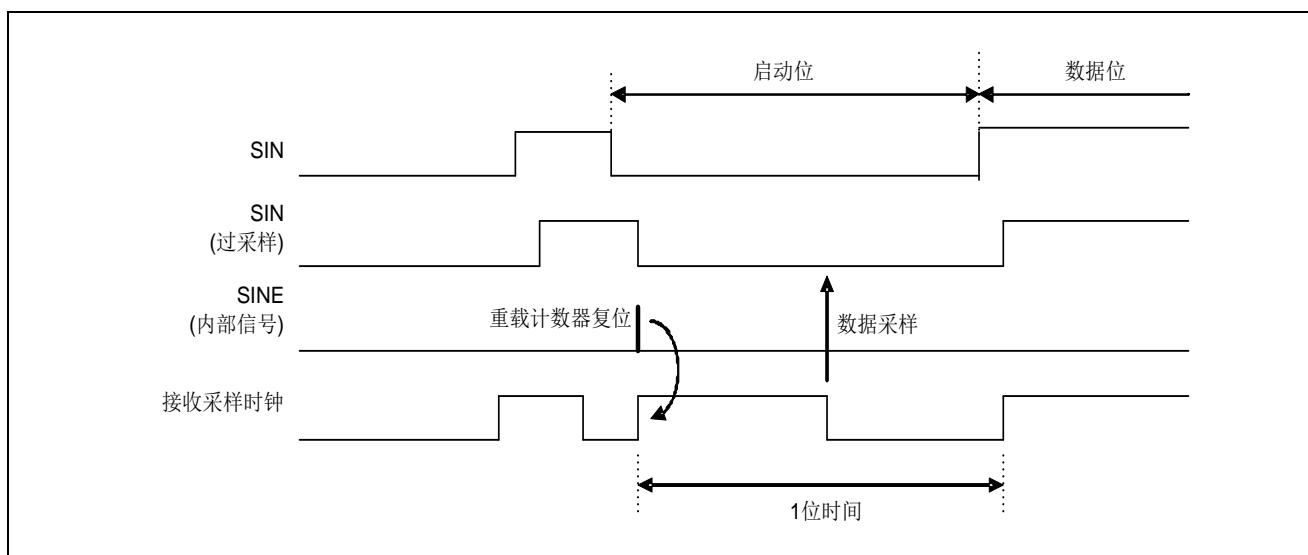
- 串行数据下降沿 (ECR:INV=0 时)
- 串行数据上升沿 (ECR:INV=1 时)
-

- 时钟选择

- 可使用内部时钟或外部时钟。
- 使用外部时钟时, 将 MR:EXT 设置为 1。外部时钟需由波特率发生器分频。从 SCK 输入外部时钟。

- 启动位检测

- 在异步模式中, 基于 SIN 信号下降沿的检测识别启动位。因此, 即使使能接收 (CR:RXE=1), 除非输入 SIN 信号下沿, 否则不会启动接收。
- 检测到启动位下降沿后, 波特率发生器的接收重载寄存器将立即复位并重载, 以启动递减计数。因此, 始终在数据中段进行采样。



- 停止位

- 可选择 1 位 - 4 位长。
- 如果检出首个停止位, 接收数据满标志位 (SR:RDFF) 立即设置为 1。

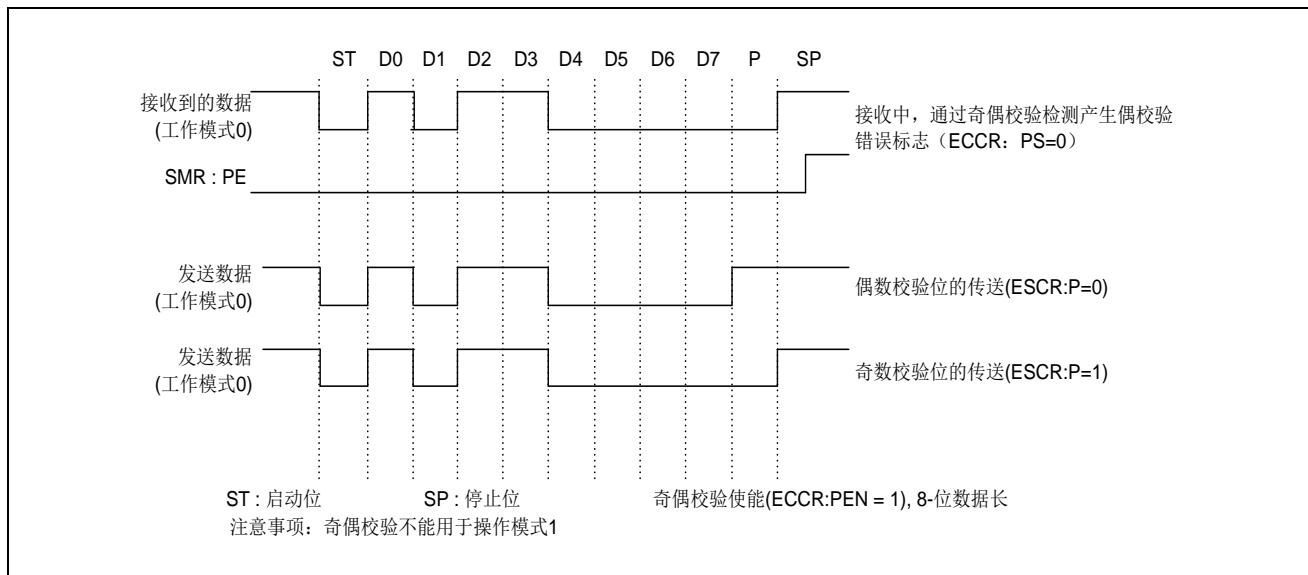
- 错误检测

- 在工作模式 0 中, 可以检测奇偶校验错误、溢出错误和帧错误。
- 在工作模式 1 中, 可以检测溢出错误和帧错误, 但是无法检测奇偶校验错误。

- 奇偶校验位

- 只能在工作模式 0 中增加奇偶校验位。可使用奇偶校验使能位(ECR:PE)指示是否使用奇偶校验及奇偶校验选择位(ECR:P)设置偶数校验或奇数校验。
- 在工作模式 1 中不能使用奇偶校验。
- 图 32.3-2 所示为使能奇偶校验时的发送/接收数据。

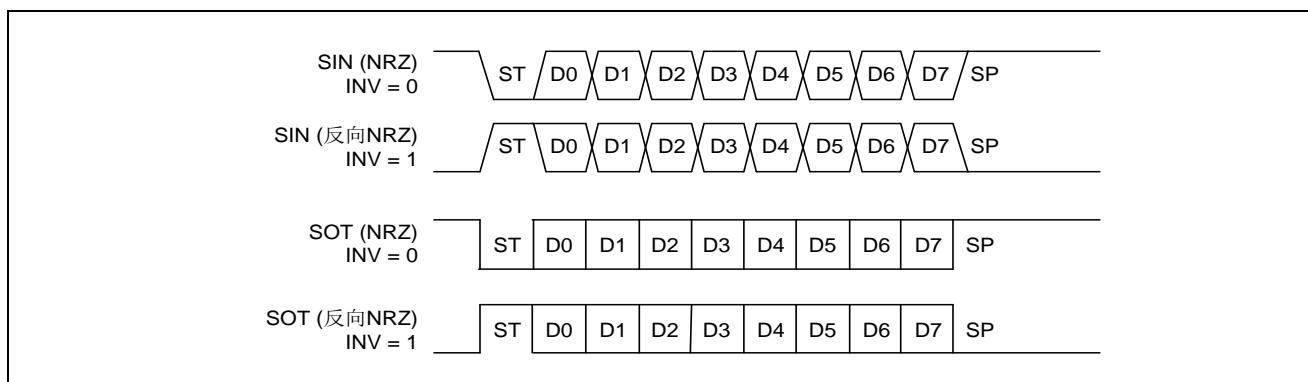
图 32.3-2 使能奇偶校验时的操作



- 数据信号系统

- 通过设置扩展通信控制寄存器的 INV 位，可选择 NRZ (不归零) 信号系统(ECR:INV=0) 或反向 NRZ 信号系统(ECR:INV=1)。
- 图 32.3-3 所示为 NRZ 和反向 NRZ 信号系统。

图 32.3-3 NRZ (不归零)信号方式以及反向 NRZ 信号方式



- 数据传输系统

- 数据位传输方式可选择 LSB 优先或 MSB 优先。

- 硬件流控制

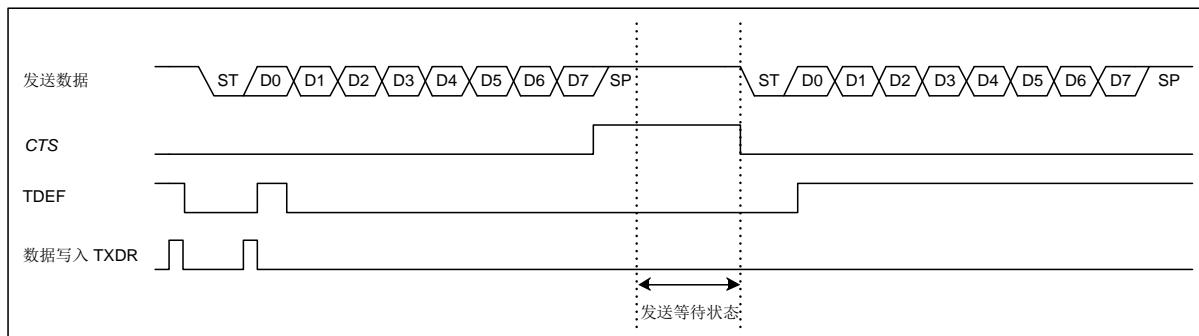
使能流控制 (ECR:FCEB=1) 后，UART 执行硬件流控制。

- 在数据发送进程中

如果数据发送后 CTS 为 HIGH，即使发送缓冲器包含数据 (TDEF=0) 也不发送下一个数据，此进程要等到 CTS 被设置为 LOW。需要发送等待时，在完成停止位发送之前在 CTS 输入 HIGH。即使在发送进程中 CTS 输入 HIGH，发送也会持续继续到停止位。

图 32.3-4 数据发送进程中的硬件流控制

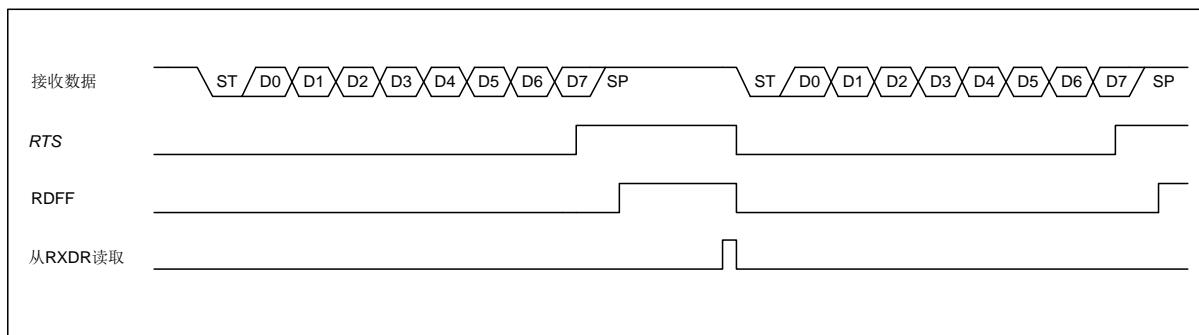
(MR:SBLS=0, ECR:ESBLS=INV=PE=DL2=DL1=DL0=0)



- 在数据接收进程中接收到停止位前一位数据时，“HIGH”输出至 RTS。读取接收数据后，将 LOW 输出至 RTS。

图 32.3-5 数据接收进程中的硬件流控制

(MR:SBLS=0, ECR:ESBLS=INV=PE=DL2=DL1=DL0=0)



注意：

- 禁用接收操作 (RXE=0) 时，RTS 信号固定为 LOW。
- 执行可编程复位 (CR:PCL=1)，将清除 RTS 信号清除为 LOW。

32.4 专用波特率发生器

UART 发送/接收的时钟源可选择以下任意一种：

- 1) 专用波特率发生器(重载计数器)
- 2) 外部时钟输入到波特率发生器(重载计数器)

选择波特率

选择以下两种波特率中的一种：

- 专用波特率发生器(重载计数器)将内部时钟分频后得到的波特率

在波特率生成器寄存器 1, 0 (BGR1, BGR0)中设定 15 位的重载值后，可选择波特率。

各重载计数器按照设置值分频内部时钟。

要设置时钟源，选择内部时钟 (BRS1:EXT=0)。

- 专用波特率发生器(重载计数器)将外部时钟分频后得到的波特率

使用外部时钟作为重载计数器的时钟源。从 SCK 输入外部时钟。

在波特率生成器寄存器 1, 0 (BGR1, BGR0)中设定 15 位重载值后，可选择波特率。

各重载计数器通过设置值分频外部时钟。

要设置时钟源，选择使用外部时钟和波特率发生器时钟 (BRS1:EXT=1)。

本模式用于使用非标准频率振荡器的情况。

注意：

- 若选择外部时钟 (BRS1:EXT=1)，必须暂停重载计数器 (BRS1/0=15' h00)。
- 若选择外部时钟 (BRS1:EXT=1)，HIGH 信号和 LOW 信号必须有至少两个总线时钟的宽度。

波特率设置

下文说明如何设置波特率和串行时钟频率计算结果。

- 计算波特率

使用波特率发生器寄存器 1 和 0 (BRS1 和 BRS0) 设置 15 位重载计数器。

通过以下公式获得波特率：

- 1) 重载值

$$V = \phi / b - 1$$

V : 重载值

b : 波特率

ϕ : 总线时钟频率或者外部时钟频率

2) 计算示例

总线时钟 16 MHz 的内部时钟,设置波特率为 19200 bps 设置重载值如下:

重载值:

$$V = (16 \times 1000000) / 19200 - 1 = 832$$

因此,实际波特率为:

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

3) 波特率误差

按以下公式计算波特率误差.

$$\text{误差}(\%) = (\text{计数值} - \text{目标值}) / \text{目标值} \times 100$$

例如: 设置 20 MHz 总线时钟和 153600 bps 的波特率:

$$\text{重载值} = (20 \times 1000000) / 153600 - 1 = 129$$

$$\text{波特率(计算值)} = (20 \times 1000000) / (129 + 1) = 153846 \text{ (bps)}$$

$$\text{误差}(\%) = (153846 - 153600) / 153600 \times 100 = 0.16 (\%):$$

$$B = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

注意:

- 如果重载值设为"0", 则重载计数器停止。
 - 重载值为偶数时, 接收串行时钟一个周期内的"L"比"H"宽; 重载值为奇数时, 串行时的 HIGH 信号和 LOW 信号宽度相同。
 - 通常重载值应大于 3, 但是, 有时因波特率误差和重载值的设置导致无法正常接收数据。
 - 在重载值设为小于 3 的情况, 配合波特率补偿功能可以有效改善传输。如系统频率为 32KHz 时, 为了得到 9.6K 的波特率, 必须配合波特率补偿功能才能确保准确的收发数据。
 - 当然, 在重载值大于 3 的情况下, 如果配合波特率补偿功能亦可得到更精准的波特率。
- 各总线时钟频率对应的重载值和波特率。

表 32.4-1 重载值与波特率

波特率 (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	重载 值	误差	重载 值	误差								
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	-	-	-	7	0	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
5000 00	15	0	19	0	31	0	39	0	47	0	63	0
4608 00	-	-	-	-	-	-	-	-	51	0.16	-	-

2500 00	31	0	39	0	63	0	79	0	95	0	127	0
2304 00	-	-	-	-	-	-	86	-0.2 2	103	0.16	138	-0.08
1536 00	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
1250 00	63	0	79	0	127	0	159	0	191	0	255	0
1152 00	-	-	86	-0.2 2	138	-0.0 8	173	-0.2 2	207	0.16	277	-0.08
7680 0	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.1 6	416	-0.08
5760 0	138	-0.0 8	173	-0.2 2	277	-0.0 8	346	0.06	416	-0.0 8	555	-0.08
3840 0	207	0.16	259	0.16	416	-0.0 8	520	-0.0 3	624	0	832	0.04
2880 0	277	-0.0 8	346	<0.0 1	554	-0.0 1	693	0.06	832	0.03	1110	0.01
1920 0	416	-0.0 8	520	-0.0 3	832	0.03	104 1	-0.0 3	124 9	0	166 6	-0.02
1041 7	767	<0.0 1	959	<0.0 1	153 5	<0.0 1	191 9	<0.0 1	230 3	<0.0 1	307 1	<0.0 1
9600	832	0.04	104 1	-0.0 3	166 6	-0.0 2	208 3	0.03	249 9	0	333 2	0.01
7200	1110	<0.0 1	138 8	<0.0 1	222 1	<0.0 1	277 7	<0.0 1	333 2	<0.0 1	444 3	0.01
4800	166 6	-0.0 2	208 2	0.02	333 2	<0.0 1	416 6	<0.0 1	499 9	0	666 6	<0.0 1
2400	333 2	<0.0 1	416 6	<0.0 1	666 6	<0.0 1	833 2	<0.0 1	999 9	0	133 32	<-0. 01
1200	666 6	<0.0 1	833 4	0.02	133 32	<0.0 1	166 66	<0.0 1	199 99	0	266 66	<0.0 1
600	133 32	<0.0 1	166 66	<0.0 1	266 66	<0.0 1	-	-	-	-	-	-
300	266 66	<0.0 1	-	-	-	-	-	-	-	-	-	-

重载值: BRS1/0 寄存器设置值(十进制)

误差 : 波特率误差 (%)

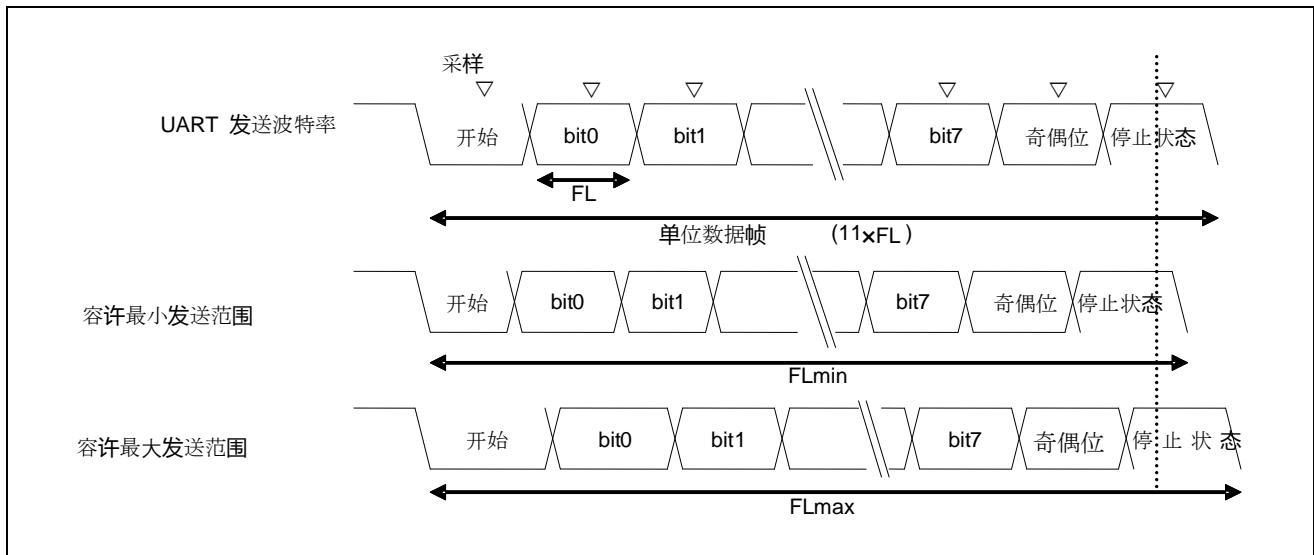
有关表 31.4-1 和表 31.4-2 中未说明的频率, 根据 “31.4 波特率设置” 进行计算。关于系统频率, 参考数据手册。

- 数据接收时的波特率容许范围

接收时，以下是对发送波特率的容许范围。

接收时的波特率误差按以下公式计算，误差不得超出容许误差范围。

图 32.4-1 接收时的波特率容许范围



如上图所示，检出启动位后，通过 BRS1/0 寄存器中设置的计数器确定接收数据的采样时序。如果包括停止位在内的位顺序与采样时序匹配，则可成功接收数据。

如果将这用于 11 位接收，那么理论上应如下所示：

如果将采样时序的余量设为总线时钟()的 1 个时钟周期，则容许最小传输率(FLmin)为：

$$FL_{min} = (11 \text{bits} \times (V+1) - (V+1)/2 + 2)/f = (21V + 25)/2f \text{ (s)} \quad V: \text{重载值}, f: \text{总线时钟}$$

因此，接收数据目标的允许最大波特率(BGmax)为：

$$BG_{max} = 11/FL_{min} = 22f/(21V+25) \text{ (bps)} \quad V: \text{重载值}, f: \text{总线时钟}$$

按最大容许传输率(FLmax)接收数据时，对第 11 位接收的起始点采样。

因此，容许最大传输率(FLmax)为：

$$10/11 \times FL_{max} = (11 \text{bits} \times (V+1) - (V+1)/2)/f \quad V: \text{重载值}, f: \text{总线时钟}$$

$$FL_{max} = (21/20 \times 11 \times (V+1))/f$$

如果将采样时序的余量(Φ)设为 2 个时钟周期，则容许最大传输率(FLmax)为：

$$10/11 \times FL_{max} = (11 \text{bits} \times (V+1) - (V+1)/2 - 2)/f \quad V: \text{重载值}, f: \text{总线时钟}$$

$$FL_{max} = (21/20 \times 11 \times (V+1) - 44/20)/f = (231V + 187)/20f \text{ (s)} \quad V: \text{重载值}, f: \text{总线时钟}$$

因此，可接收的最小容许波特率(BGmin)为：

$$BG_{min} = 11/FL_{max} = 220 / (231V + 187) \text{ (bps)} \quad V: \text{重载值}, f: \text{总线时钟}$$

从上述最小/最大波特率值的计算公式可算出 UART 和目标之间的波特率容许误差，

如下所示：

重载值 (V)	容许最大波特率误差	容许最小波特率误差
3	0%	0
10	+2.98%	-3.08%
50	+4.37%	-4.40%
100	+4.56%	-4.58%
200	+4.66%	-4.67%
32767	+4.76%	-4.76%

注意：

- 接收精度取决于 1 帧的位数、总线时钟和重载值。总线时钟和分频比越大，精确度越高。
- 上述情况是指在无波特率补偿功能的情况下，如果结合波特率补偿，可以使得在重载值较小的情况下允许更大误差范围的波特率误差而保证正常的数据传输。

- 外部时钟

波特率发生器寄存器(BRS)的 EXT 位为"1"时，波特率发生器分频外部时钟(从 SCK 引入的时钟)。

注意：

- 外部时钟信号与 UART 上的内部时钟同步。因此，若外部时钟无法实现同步，则工作不稳定。

- 重载计数器的功能

重载计数器有两种：发送重载计数器和接收重载计数器，都用作专用波特率发生器。各重载计数器包含 15 位的重载寄存器，并通过外部时钟或内部时钟产生发送时钟和接收时钟。

- 启动计数

将重载值写入波特率发生器寄存器 1, 0 (BRS1 或 BRS0) 时，重载计数器启动计数。

- 重新启动

重载计数器在以下条件下重新启动计数：

- 发送和接收公用重载计数器，可编程复位(CR:PCL 位)
- 接收重载计数器，在异步模式下检出启动位下降沿

32.5 波特率补偿及设置

UART 支持波特率补偿功能，用它来优化每一位的精度。

补偿精度为一个 UART 工作时钟，波特率补偿不可单独使用，必须配合波特率发生器。

按位对 UART 传输信号频率进行补偿

波特率补偿

补偿的精度是 UART 工作时钟。由 COMP1.COE 位来定义是否开启波特率补偿，由 COMP1/COMP0 的低 10 位来配置对每一位是否补偿，为 0 表示不补偿(波特率设置值)，为 1 表示补偿(即波特率设置值加 1)。

需要做波特率补偿时，对波特率配置寄存器的计算值取整即可，余数部分舍弃。

通常在系统频率较低时且要求波特率相对较高的情况下，需要做波特率补偿。

表 32.5-1 波特率补偿与 UART 数据对应表

寄存器		补偿(LSB 模式)					补偿(MSB 模式)				
名称	补偿位	9 位	8 位	7 位	6 位	5 位	9 位	8 位	7 位	6 位	5 位
COMP0	0	起始	起始	起始	起始	起始	起始	起始	起始	起始	起始
	1	D0	D0	D0	D0	D0	D8	D7	D6	D5	D4
	2	D1	D1	D1	D1	D1	D7	D6	D5	D4	D3
	3	D2	D2	D2	D2	D2	D6	D5	D4	D3	D2
	4	D3	D3	D3	D3	D3	D5	D4	D3	D2	D1
	5	D4	D4	D4	D4	D4	D4	D3	D2	D1	D0
	6	D5	D5	D5	D5	无效	D3	D2	D1	D0	无效
	7	D6	D6	D6	无效	无效	D2	D1	D0	无效	无效
COMP1	8	D7	D7	无效	无效	无效	D1	D0	无效	无效	无效
	9	D8	无效	无效	无效	无效	D0	无效	无效	无效	无效

波特率补偿设置

- 设置示例

假定 LSB 模式，外设时钟为 32.768KHz，波特率为 9600. $(32.768 \times 10^3) / 9600 = 3.413$ 个外设时钟位。

因此设置 BRS1/0 = 取整($32.768 \times 10^3 / 9600 - 1$) = 2，每一位的误差约为 0.413(靠近 0.5). 为了保证数据收发有效，可以隔一位补偿一次。可以对接收 UART 接收的波特率补偿寄存器值设置为 0x82AA 或者 0x8155，UART 发送的波特率补偿寄存器值设置为 0x8155。计算方式

如下：

数据位	总误差	补偿值	最终误差
启动位	-0.413	0	-0.413
D0	-0.826	1	0.176
D1	-0.237	0	-0.237
D2	-0.650	1	0.350
D3	-0.063	0	-0.476
D4	-0.889	1	0.111
D5	-0.302	0	-0.302
D6	-0.715	1	0.285
D7	-0.128	0	-0.128
D8	0.541	1	0.459

因此补偿寄存器设置为 0x82AA.

假定外设时钟为 32.768KHz, 波特率为 4800. $(32.768 \times 10^3) / 4800 = 6.827$ 个外设时钟位。

因此设置 BRS1/0 = 取整($32.768 \times 10^3 / 4800 - 1$) = 6, 每一位的误差约为 0.827, 补偿值设置为 0x82AA。

结论：补偿值得设置就是指在计算总误差大于 0.5 时，将对应的补偿值设置为 1，否则设置为 0。

32.6 工作模式 0 (异步正常模式)设置步骤和编程流程

工作模式 0 使能异步双向串行通信。

CPU 至 CPU 连接

在工作模式 0 (正常模式) 中选择双向通信。

图 32.6-1 和图 32.6-2 为 2 个 CPU 相互连接示例。

图 32.6-1 UART 工作模式 0 中的双向通信连接示例（禁用流控制）

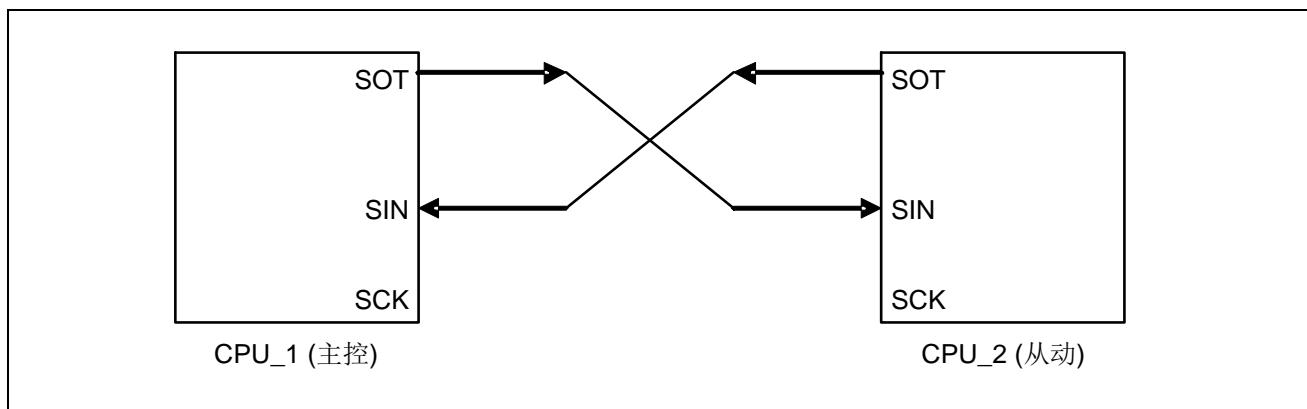
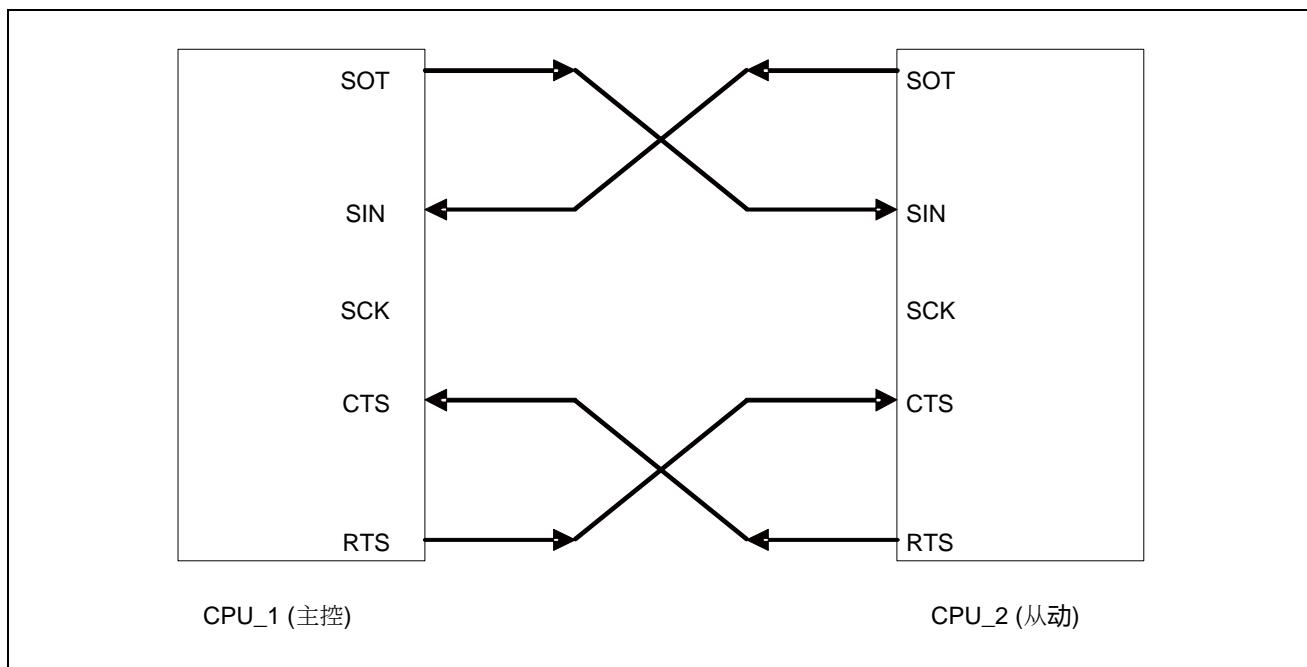
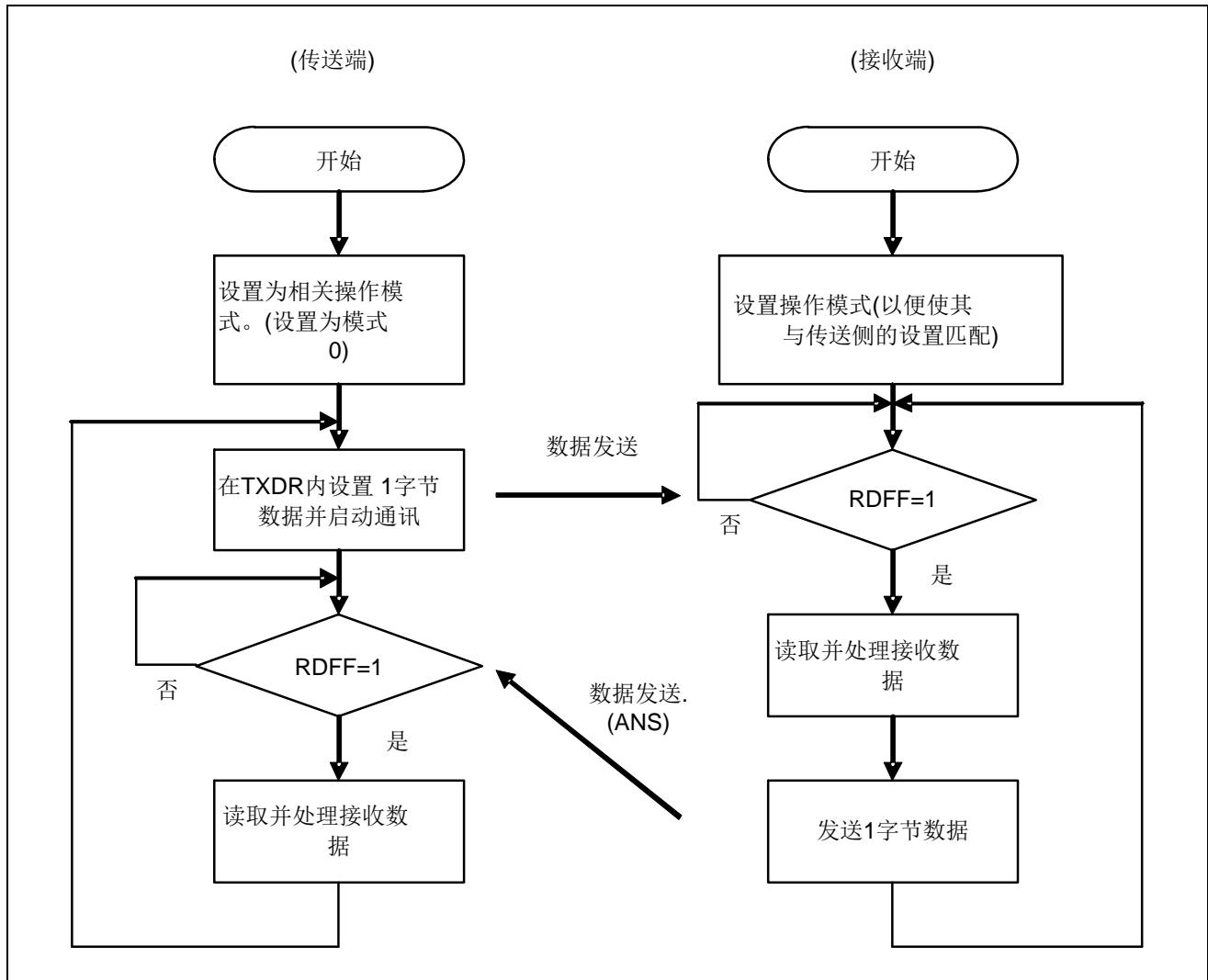


图 32.6-2 UART 工作模式 0 中的双向通信连接示例（具有流控制）



流程图

图 32.6-3 双向通信流程示例



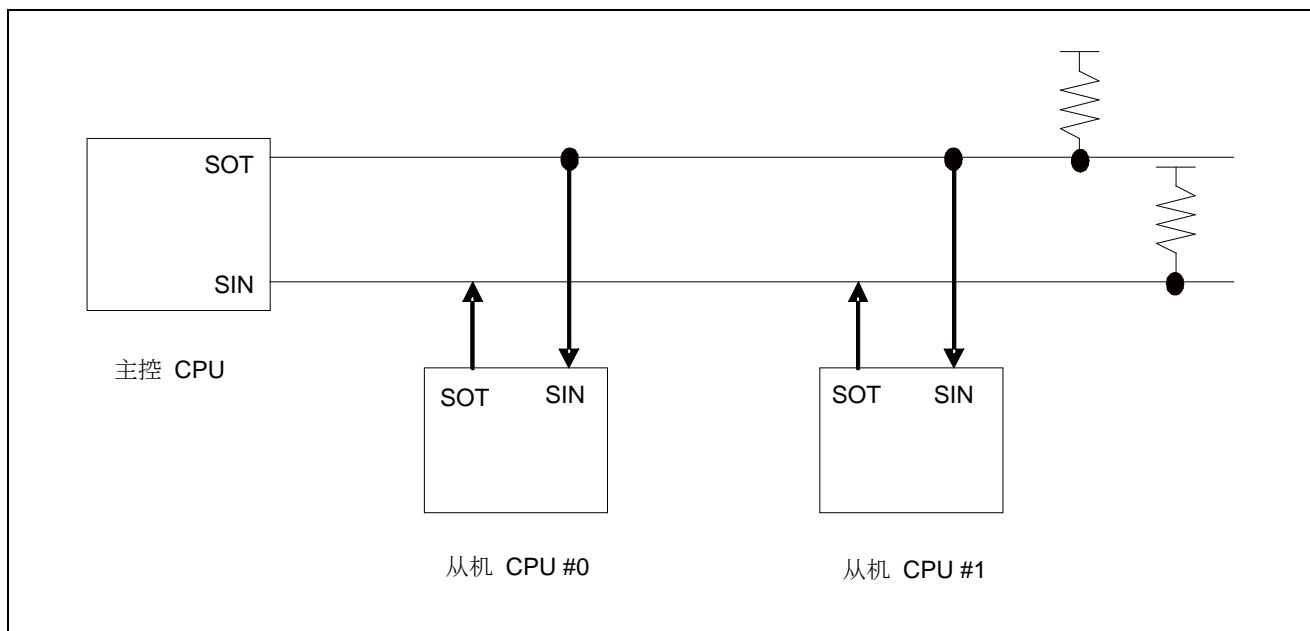
32.7 工作模式 1(异步多处理器模式)设置步骤和程序流程

在工作模式 1(多处理器模式)中，使能与多个 CPU 连接的主机/从机通信。具有主机功能或从机功能。

CPU 至 CPU 连接

如图 32.7-1 所示的在主机/从机通信中，通信系统配置有两条与主机 CPU 和多个从机 CPU 连接的公用通信线路。UART 可用作主控或从属。

图 32.7-1 UART 上的主机/从机通信连接示例



功能选择

在主机/从机通信中，选择工作模式和数据传输系统，如表 32.7-1 所示。.

表 32.7-1 选择主机/从机通信功能

	工作模式		数据	奇偶校验	停止状态位	位描述
	主机模式 CPU	从机模式 CPU				
地址收/发	模式 1 (地址位发送)	模式 1 (地址位接收)	AD=1 + 7 位或 8 位地址	关闭	1 位或 2 位	LSB 优先或 MSB 优先
			AD=0 + 7 位或 8 位数据			

注意：

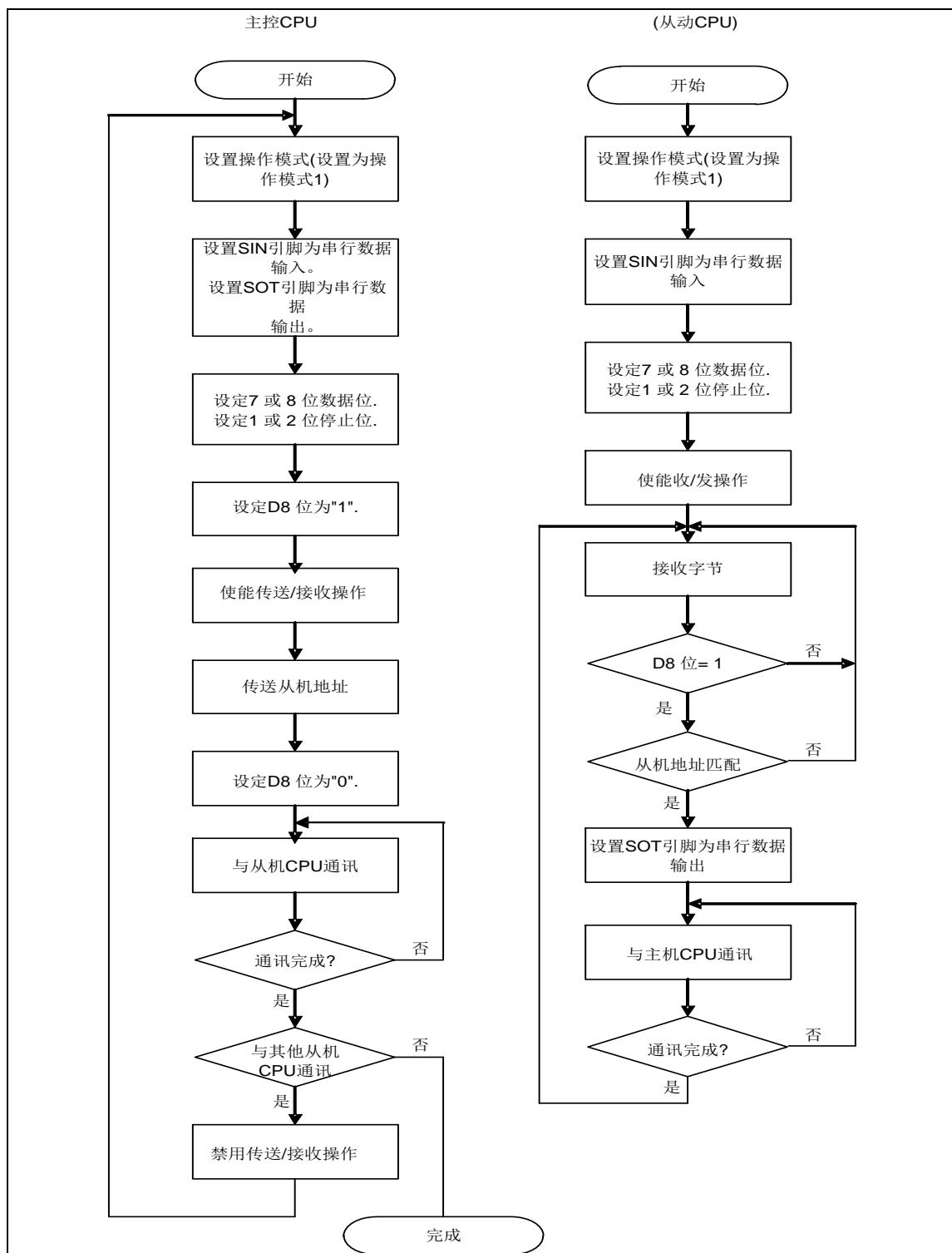
- 在工作模式 1 时，发送/接收数据 (TXDR/RXDR) 以字访问模式工作。

通信程序

主机 CPU 发送地址数据时，通信启动。地址数据为 D8 位为 1 的数据集，用于选择进行通信的从机 CPU。

各从机 CPU 根据程序判断地址，若该地址与分配地址匹配，则与主机 CPU 进行通信。

图 32.7-2 所示为主机/从机通信的流程图（多处理器模式）



32.8 UART(异步串行接口)寄存器

以下是 UART(异步串行接口)寄存器一览表。

UART (异步串行接口)寄存器列表

表 32.8-1 UART (异步串行接口)寄存器列表

	位 15	位 8	位 7	位 0
UAR T	CR (串行控制寄存器)	MR (串行模式寄存器)		
	SR (串行状态寄存器)	ECR (扩展通信控制寄存器)		
	RXDR1/TXDR1(收/发数据寄存器 1)	RXDR0/TXDR0 (收/发数据寄存器 0)		
	BRS1 (波特率发生器寄存器 1)	BRS0 (波特率发生器寄存器 0)		
	COMP1(波特率补偿寄存器 1)	COMP0(波特率补偿寄存器 0)		

表 32.8-2 UART (异步串行接口)寄存器位信息

	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CR/ MR	PCL	-	-	RXI E	TXI E	TBI E	RXE	TXE	MO DE2	MO DE1	MO DE0	-	SBL S	TDS	-	DOE
SR/ ECR	REC LR	-	PEF	FEF	ORF	RDF F	TDE F	TBI F	FCE B	ESB LS	INV	PE	PS	DL2	DL1	DL0
TXDR / (RXD R)							D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0	
BRS1/ BRS0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
COMP	COE	-	-	-	-	-	CE9	CE8	CE7	CE6	CE5	CE4	CE3	CE2	CE1	CE0

模式选择

UART(异步串行接口)支持 2 种不同的工作模式。取决于串行模式寄存器(MR)的 MODE2, MODE1, MODE0。注意，在配置其他串口寄存器之前配置模式选择寄存器。

表 32.8-3 UART(异步串行接口)工作模式选择

工作模式	MODE2	MODE1	MODE0	Type
0	0	0	0	UART0 (异步单从机模式)
1	0	0	1	UART1 (异步多处理器模式)

32.8.1 串行控制寄存器(CR)

串行控制寄存器(CR)使能/禁止收/发、使能/禁止收/发中断、使能/禁止发送总线空闲中断、执行 UART 复位。

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	PCL	-	-	RXIE	TXIE	TBIE	RXE	TXE	(MR)		
属性	R/W	-	-	R/W	R/W	R/W	R/W	R/W			
初始值	0	-	-	0	0	0	0	0			

寄存器功能

[bit15] PCL: 编程清零位, 初始化 UART 的内部状态.

位	说明	
	写入	读取
0	操作无效	
1	可编程清除	始终读“0”

设为"1"时:

- 直接将 UART 复位(软件复位), 但保持当前寄存器设置。这时断开收/发状态。
- 波特率发生器重载 BRS1/0 寄存器的设置值并重新启动。
- 全部收/发中断源(SR:PEF, FEF, ORF, RDFF, TDEF, TBIF)被初始化("0b0000011")。
- RTS 信号清除为"L"。

设为"0"时:

- 操作无效; 读取时, 总读为 0。

注意:

- 设为中断禁止后, 然后执行可编程清除指令。

[bit14:13]: 保留位, 读取: 读取值未定义; 写入: 无效。

[bit12] RXIE: 接收中断使能位

使能/禁止向 CPU 输出接收中断请求.

RXIE 位和接收数据标志位(SR:RDFF)为"1"或任一错误标志位(SR:PEF, ORF, FEF)为"1"时, 输出接收中断请求。

位	说明
0	禁止接收中断

1	使能接收中断
---	--------

[bit11] TXIE: 发送中断使能位

使能/禁止向 CPU 输出发送中断请求

TXIE 位和 SR:TDEF 位为"1"时，输出发送中断请求。

位	说明
0	禁止发送中断
1	使能发送中断

[bit10] TBIE: 发送总线空闲中断使能位

使能/禁止向 CPU 输出发送总线空闲中断请求。

TBIE 位和 TBIF 位为"1"时，输出发送总线空闲中断请求。

位	说明
0	禁止发送总线空闲中断
1	使能发送总线空闲中断

[bit9] RXE: 接收工作使能位，使能/禁止 UART 接收工作.

位	说明
0	禁止接收
1	使能接收

注意：

- 即使使能接收工作模式(RXE="1")，如果未输入启动位的下降沿(NRZ 格式(ECR:INV="0"))时，则不启动接收工作模式。(反向 NRZ 格式(ECR:INV="1")时，输入上升沿之前，不启动接收工作模式)
- 接收期间禁止接收工作模式(RXE="0")时，立刻停止接收工作模式。
- 禁止接收工作模式(RXE="0")时，RTS 信号固定为"L"

[bit8] TXE: 发送工作模式使能位，使能/禁止 UART 发送工作模式。

位	说明
0	禁止发送
1	使能发送

注意：

- 发送期间禁止发送工作模式(TXE="0")时，立刻停止发送工作模式。

32.8.2 串行模式寄存器(MR)

串行模式寄存器(MR)设置工作模式，选择传输方向、数据长、停止位长以及使能/禁止输出串行数据至引脚。

寄存器配置

位	15	..	8	7	6	5	4	3	2	1	0
字段名	(CR)			MOD E2	MOD E1	MOD E0	-	SBLS	TDS	Reserved	DOE
属性				R/W	R/W	R/W	-	R/W	R/W	-	R/W
初始值				0	0	0	0	0	0	0	0

寄存器功能

[bit7:5] MODE2, MODE1, MODE0: 工作模式设置位，设置异步串行接口的工作模式。

本章介绍工作模式 0(异步正常模式)、工作模式 1(异步多处理器模式)的寄存器及其工作。

位 7	位 6	位 5	说明
0	0	0	工作模式 0(异步正常模式)
0	0	1	工作模式 1(异步多处理器模式)
0	1	0	工作模式 2(时钟同步模式)
1	0	0	工作模式 3(I2C 模式)
其他值		禁止设置	

注意：

- 禁止上述以外的设置。
- 关于工作模式的切换，应执行编程清零(CR:PCL="1")后再切换。
- 转换工作模式后，正确设置各寄存器。

[bit4] 保留位，未定义位。读值为"0"。始终写"0"。

[bit3] SBLS: 停止位长选择位

设置停止位(发送数据的帧端标记)的位长

位	说明	
0	ECR:ESBLS=0	1 位
	ECR:ESBLS=1	3 位
1	ECR:ESBLS=0	2 位
	ECR:ESBLS=1	4 位

注意：

- 接收时，通常只检出停止位数据的第 1 位，应在禁止发送(CR:TXE="0")时设置该位。

[bit2] TDS: 传输方向选择位，该位选择从最低位传输串行数据(LSB 优先, TDS="0")或从最高

位传输数据(MSB 优先, TDS="1")。

位	说明
0	LSB 优先(从最低位开始传输)
1	MSB 优先(从最高位开始传输)

注意:

- 应在禁止收/发(CR:TXE=CR:RXE="0")时设置该位。

[bit1] 保留位, 未定义位。读值为"0"。始终写"0"。

[bit0] DOE: 串行数据输出使能位, 该位使能/禁止输出串行数据。

位	说明
0	禁止输出串行数据
1	使能输出串行数据

注意:

- 此位用作 SOT 时, GPIO 需要配置成选择外设功能。

32.8.3 串行状态寄存器(SR)

串行状态寄存器(SR)用于检查收/发状态、检查接收错误标志、清零接收错误标志

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	REC LR	-	PEF	FEF	ORF	RDF F	TDE F	TBIF	(ECR)		
属性	R/W	-	R	R	R	R	R	R			
初始值	0	-	0	0	0	0	1	1			

寄存器功能

[bit15] RECLR: 接收错误标志清零位，该位清零串行状态寄存器(SR)的 PEF, FEF, ORF 标志。

位	说明	
	写入	读取
0	无效.	始终返回"0"
1	清除接收错误标志位(PEF, FEF, ORF).	

[bit14] - : 保留位，未定义位。读值不定，写值无效。

[bit13] PEF: 奇偶错误标志位 (工作模式 0 时有效)

如果 ECR:PE="1"时接收发生奇偶错误，则该位置"1"，将串行状态寄存器(SR)的 RECLR 位设为"1"可将其清零。

PEF 位和 CR:RXIE 位为"1"时，输出接收中断请求。

该标志置位时，接收数据寄存器(RXDR)的数据无效。

位	说明
0	无奇偶校验错误
1	有奇偶校验错误

[bit12] FEF: 帧错误标志位

如果接收数据时发生帧错误，则该位为"1"，向串行状态寄存器(SR)的 RECLR 位写"1"可将其清零。

FEF 位和 CR:RXIE 位为"1"时，输出接收中断请求。

该标志置位时，接收数据寄存器(RXDR)的数据无效。

位	说明
0	无帧错误
1	有帧错误

[bit11] ORF: 溢出错误标志位

如果接收时发生溢出错误，则该位置"1"，向串行状态寄存器(SR)的 RECLR 位写"1"可将其清零。

ORF 位和 CR:RXIE 位为"1"时，输出接收中断请求。

该标志置位时，接收数据寄存器(RXDR)的数据无效。

位	说明
0	无溢出错误
1	有溢出错误

[bit10] RDFF: 接收数据满载标志位

该标志显示接收数据寄存器(RXDR)的状态。

如果收数据载入 RXDR，则该位置"1"，读取接收数据寄存器(RXDR)可将其清"0"。

RDFF 位和 CR:RXIE 位为"1"时，输出接收中断请求。

位	说明
0	接收数据寄存器 RXDR 放空
1	接收数据寄存器 RXDR 内有数据

[bit9] TDEF: 发送数据空标志位

该标志显示发送数据寄存器(TXDR)的状态。

如果向 TXDR 写发送数据，则该位清"0"，表示 TXDR 内存在有效数据。数据载入发送移位寄存器后开始发送时，该位为"1"，表示 TXDR 内没有有效数据。

TDEF 位和 CR:TXIE 位为"1"时，输出发送中断请求

如果串行控制寄存器(CR)的 PCL 位设为"1"，则 TDEF 位变为"1"。

位	说明
0	发送数据寄存器 TXDR 内有数据
1	发送数据寄存器放空

[bit8] TBIF:发送总线空闲标志位，该位表示 UART 没有执行发送工作模式

向发送数据寄存器(TXDR)写发送数据时，该位清"0"。

发送数据寄存器放空(TDEF="1")且未执行发送工作模式时，该位为"1"。

如果串行控制寄存器(CR)的 PCL 位为"1"，则 TBIF 位为"1"。

该位设为"1"且使能发送总线空闲中断(CR:TBIE="1")时，输出发送中断请求。

位	说明
0	正在发送数据
1	无数据发送

32.8.4 接收数据寄存器/发送数据寄存器(RXDR/TXDR)

接收数据和发送数据寄存器位于同一地址。读取时用作接收数据寄存器，写入时用作发送数据寄存器。

1) 接收数据寄存 (RXDR)

寄存器配置

位	15	...	9	8	7	6	5	4	3	2	1	0
字段名				D8	D7	D6	D5	D4	D3	D2	D1	D0
属性				R	R	R	R	R	R	R	R	R
初始值				0	0	0	0	0	0	0	0	0

寄存器功能

接收数据寄存器(RXDR)是串行数据接收用的 9 位数据缓冲寄存器。

发送到串行输入引脚(SIN 引脚)的串行数据信号通过移位寄存器进行转换后，保存在接收数据寄存器(RXDR)中。

根据数据长度将上位写"0"，如下所示：

数据长	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 位	X	X	X	X	X	X	X	X	X
8 位	0	X	X	X	X	X	X	X	X
7 位	0	0	X	X	X	X	X	X	X
6 位	0	0	0	X	X	X	X	X	X
5 位	0	0	0	0	X	X	X	X	X

(X 表示数据接收位。)

如果接收数据保存到接收数据寄存器(RXDR)，则接收数据满载标志位(SR:RDFF)置"1"。使能接收中断(SR:RXIE="1")后，产生接收中断请求。

只有在接收数据满载标志位(SR:RDFF)为"1"时才能读取接收数据寄存器(RXDR)。如果读取接收数据寄存器(RXDR)，则接收数据满载标志位(SR:RDFF)自动清"0"。

发生接收错误(SR:PEF, ORF, FEF 的任一位为"1")时，接收数据寄存器(RXDR)的数据无效。

工作模式 1(多处理器模式)时，进行 7 位或 8 位长度的操作并将已接收的 AD 位保存到 D8 位。

如果是 9 位长度传输并在工作模式 1 时，必须按 16 位读取 RXDR。

2) 发送数据寄存器(TXDR)

寄存器配置

位	15	...	9	8	7	6	5	4	3	2	1	0
字段名				D8	D7	D6	D5	D4	D3	D2	D1	D0
属性				W	W	W	W	W	W	W	W	W
初始值				1	1	1	1	1	1	1	1	1

寄存器功能

发送数据寄存器(TXDR)是串行数据发送用的 9 位数据缓冲寄存器。

使能发送工作模式(CR:TXE="1")时, 如果将发送数据写入发送数据寄存器(TXDR), 则发送数据将传输到发送移位寄存器。然后, 将发送数据转换为串行数据并从串行数据输出引脚(SOUT 引脚)发出。

根据数据长度, 如下表所示, 上位将依次变为无效:

数据长	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 位	X	X	X	X	X	X	X	X	X
8 位	无效	X	X	X	X	X	X	X	X
7 位	无效	无效	X	X	X	X	X	X	X
6 位	无效	无效	无效	X	X	X	X	X	X
5 位	无效	无效	无效	无效	X	X	X	X	X

(X 表示发送数据位)

如果发送数据写入发送数据寄存器(TXDR), 发送数据空标志(SR:TDEF)将被清"0"。

如果发送数据传输到发送移位寄存器并开始发送, 发送数据空标志(SR:TDEF)将被置"1"。

发送数据空标志(SR:TDEF)为"1"时, 可写入发送数据。如果使能发送中断, 将发生发送中断。

产生发送中断后或发送数据空标志(SR:TDEF)为"1"时, 执行数据发送。

发送数据空标志(SR:TDEF)为"0"时, 无法写入发送数据。

工作模式 1(多处理器模式)时, 进行 7 位或 8 位长度的操作, 并通过向 D8 为写值以发生 AD 位。

9 位长度的数据传输并且工作模式 1 时, 按 16 位写数据入 TXDR

注意:

- 发送数据寄存器为只写, 接收数据寄存器为只读。因为收/发寄存器分配到同一地址, 而读值/写值不同, 所以, 无法使用 INC/DEC 命令和其它读改写(RMW)命令。

32.8.5 波特率发生器寄存器(BRS1,BRS0)

波特率发生器寄存器 1,0 (BRS1, BRS0)用于设置串行时钟的分频比。此外，可选择外部时钟作为重载计数器的时钟源。

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	EXT	(BRS1)										(BRS0)				
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

按字或者半字进行写访问波特率发生器寄存器(BRS1, BRS0)。

波特率发生器寄存器设置串行时钟的分频比。

BRS1 对应高位， BRS0 对应低位， 可写入需要计数的重载值以及读取设置值。

向波特率发生器寄存器 1,0 (BRS1, BRS0)写重载值后， 重载计数器开始计数。

EXT 位（位 15）指定使用重载计数器的时钟源作为内部时钟或外部时钟。EXT="0"时， 选择内部时钟。EXT="1"时， 选择外部时钟。

[bit15] EXT: 外部时钟选择位

位	说明
0	使用内部时钟(APB 总线时钟)
1	使用外部时钟(从 SCK 输入).

[bit14:8] BRS1: 波特率发生器寄存器 1

位 14:8	说明
写入	写入重载计数器位 8 ~ 位 14
读取	读取 BRS1 的设置值

[bit7:0] BRS0: 波特率发生器寄存器 0

位 7:0	说明
写入	写入重载计数器位 0 ~ 位 7
读取	读取 BRS0 的设置值

注意：

- 按 16 位数据访问将数据写入波特率发生器寄存器(BRS1, BRS0)。
- 如果改变波特率发生器寄存器(BRS1, BRS0)的当前值， 只有在计数值达到"15h00"后， 才能重载新值。因此， 如需使新的设置值立刻生效， 需改变 BRS1/0 的设置值并执行编程清

零(PCL)。

- 重载值为偶数时，接收串行时钟一个周期内的“L”比“H”宽。为奇数时，“H”宽和“L”宽相同。
- 应将 BRS1/0 设置为 4 或更大值。注意，由于波特率误差和重载值的设置，可能无法正常接收数据。
- 如果波特率发生器正在工作时改变外部时钟设置(EXT="1")，那么向波特率发生器 1, 0(BRS1, BRS0)写"0"并执行程序清零(PCL)后，设置外部时钟(EXT="1")。

32.8.6 扩展通信控制寄存器 (ECR)

扩展通信控制寄存器(ECR)用于设置收/发数据长度、使能/禁止奇偶校验位、选择奇偶校验位、反向串行数据格式、选择停止位长度。

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	(SR)			FCEB	ESB LS	INV	PE	PS	DL2	DL1	DL0
属性				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值				0	0	0	0	0	0	0	0

寄存器功能

[bit7] FCEB:硬件流控制使能位

该位选择使能/禁止硬件流控制工作模式。

位	说明
0	禁止硬件流控制
1	使能硬件流控制

注意：

- 应在禁止数据收/发(CR:TXE="0", RXE="0")时设置该位。
- 只在希望执行硬件流控制时，设置该位为"1"。

[bit6] ESBLS: 扩展停止位长度选择位

设置停止位长度(发送数据的帧端标记)。

位	说明
0	MR:SBLS=0 1 位
	MR:SBLS=1 2 位
1	MR:SBLS=0 3 位
	MR:SBLS=1 4 位

注意：

- 接收时，只能检出停止位的第 1 位。应在禁止发送(CR:TXE="0")时设置该位。

[bit5] INV: 反向串行数据格式位，该位将串行数据格式选为 NRZ 格式或反向 NRZ 格式。

位	说明
0	NRZ 格式
1	反向 NRZ 格式

[bit4] PE: 奇偶校验使能位(仅工作模式 0 时有效)

该位设置是否添加奇偶校验位(发送时)以及检测奇偶校验位(接收时)。

位	说明
0	禁止奇偶校验.
1	使能奇偶校验.

注意：

- 工作模式"1"时，该位内部固定为"0"。

[bit3] PS: 奇偶方式选择位 (仅工作模式 0 时有效)

使能奇偶校验 (ECR:PE=1) 时，此位将设置为奇数校验 1 或偶数校验 "0"。

位	说明
0	偶数校验
1	奇数校验

[bit2:0] DL2, DL1, DL0: 数据长度选择位，设置收/发数据的数据长度。

位 2	位 1	位 0	说明
0	0	0	8 位长
0	0	1	5 位长
0	1	0	6 位长
0	1	1	7 位长
1	0	0	9 位长

注意：

- 禁止除上表外的设置。
- 工作模式 1 时，数据长设为 7 位和 8 位。禁止其他设置。

32.8.7 波特率补偿寄存器(COMP1,COMP0)

波特率补偿寄存器 1,0 (COMP1, COMP0)用于优化 UART 每一位的精度。补偿精度为 1 个 UART 反向时 (APB1 时钟)。设置为 1 时表示在原波特率基础上增加一个系统时钟，为 0 时表示输出原波特率。

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	CE	-	-	-	-	-	CE9	CE8	-	-	-	-	-	-	CE7-CE0	-
属性	R/W	R0	R0	R0	R0	R0	R/W	R/W								
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

波特率补偿用于优化 UART 的每一位的波特率。

COMP1 对应高位，COMP0 对应低位，可写入需要补偿的位及读取设置值。

向波特率补偿寄存器 1,0 (COMP1, COMP0)写补偿值并将 CE 置“1”后，位补偿功能生效。

[位 15] CE: 波特率补偿启停控制位，启动/停止波特率补偿功能。

位	说明
0	停止波特率补偿
1	启动波特率补偿

注意：

- 在 BRS1/0 设置值小于 4 时，必需开启波特率补偿功能，才有可能保证数据传输的有效性；
- 在 BRS1/0 设置值大于 4 时，建议开启波特率补偿功能，可有效优化 UART 的精度。

[位 9:0] CE9-CE0: 波特率对应位补偿使能

位 9:0	说明
0	对波特率对应位不补偿(维持波特率设置值)
1	对波特率对应位补偿(即波特率设置值加 1)

注意：

- 波特率补偿位与波特率设置值相加后为偶数时，关于接收串行时钟的“H”宽和“L”宽，“L”宽比“H”宽长 1 个总线时钟周期。为奇数时，串行时钟的“H”宽和“L”宽相同。

33. SPI(串行外设接口)

本章介绍工作模式 2 支持的 SPI(串行外设接口)功能。SPI 功能是多功能串行通讯接口的一部分。

33.1 SPI(串行外设接口)概要

SPI(串行外设串口)是通用串行数据通信接口，通过外部设备可以进行同步通信。

SPI 功能

		功能
1	数据缓冲	全双工双缓冲
2	传输系统	支持主/从模式 支持无片选信号模式(时钟同步模式)
3	波特率	专用波特率发生器(配置有 15 位重载计数器；主机模式操作时) 可输入外部时钟(从机模式操作时)
4	数据长度	5~16位
5	接收错误检测	溢出错误
6	中断请求	接收中断(接收结束，溢出错误) 发送中断(发送数据为空，发送总线空闲)
7	串行片选	单主机控制 建立/保持/取消选定时间可设置 可设置各通道有效电平
8	同步传输功能	数据按照同步串行定时器设置的周期自动发送。
9	定时器功能	搭载 16 位串行定时器。 操作时钟分频可从 1/1 到 1/256 中选择
10	同步模式	主机模式或者从机模式功能
11	引脚访问	串行数据输出引脚可设置到“1”

33.2 中断

SPI 中断可分为接收中断，发送中断和状态中断。下列情形下产生中断请求：

- 接收数据寄存器(RXDR)中接收了数据或者发生数据接收错误
- 发送数据从发送数据寄存器 (TXDR) 传输到发送移位寄存器且开始数据发送
- 发送总线空闲(无发送数据)
- 串行定时器比较值(TMC)与串行定时器计数值(TMR)相同

SPI 中断

SPI 中断控制位和中断源如表 33.2-1 所示。

表 33.2-1 SPI 中断控制位和中断源

中断类型	中断请求标志位	标志寄存器	中断源	中断源使能位	清零中断请求标志位
接收	RDF F	SR	单字节接收	CR:RXIE	读取接收数据(RXDR)
	ORF	SR	溢出错误		向接受错误标志清零位(SR:RECLR)写"1"
发送	TDE F	SR	发送寄存器为空	CR:TXIE	向发送数据(TXDR)写值
	TBIF	SR	发送总线空闲(无发送数据)	CR:TBIFE	
	CSE R	EAC R	从机模式(CR:MSS=1)或者串行芯片选择引脚为不活动的主机发送模式(CR:MSS=0)时,发送计数器等于或小于TBYTE 设置值且下一发送数据未写入TXDR(SR:TDEF=1)	EACR:CSEIE	向片选标志位(EACR:CSER)写“0”
状态	TIF	EAC R	串行定时器比较值(TMC)与串行定时器计数值(TMR)相同	EACR:TIF	向定时器中断标志位(EACR:TIF)写“0”.

33.2.1 接收中断产生和标志位设置时序

当接收完成(SR:RDFF=1)或者接收错误发生 (SR:ORF=1)时产生接收中断。

接收中断产生和标志位设置时序

检测到最终数据位时，接收数据保存到接收数据寄存器(RXDR)。数据接收完毕(SR:RDFF=1)或者发生接收错误(SR:ORF=1)时，设置各标志位。此时，若使能接收中断(SR:RXIE=1)，则会发生接收中断。

注意：

- 发生接收错误时，接收数据寄存器(RXDR)的数据会失效。

图 33.2-1 数据接收和标志位设置时序

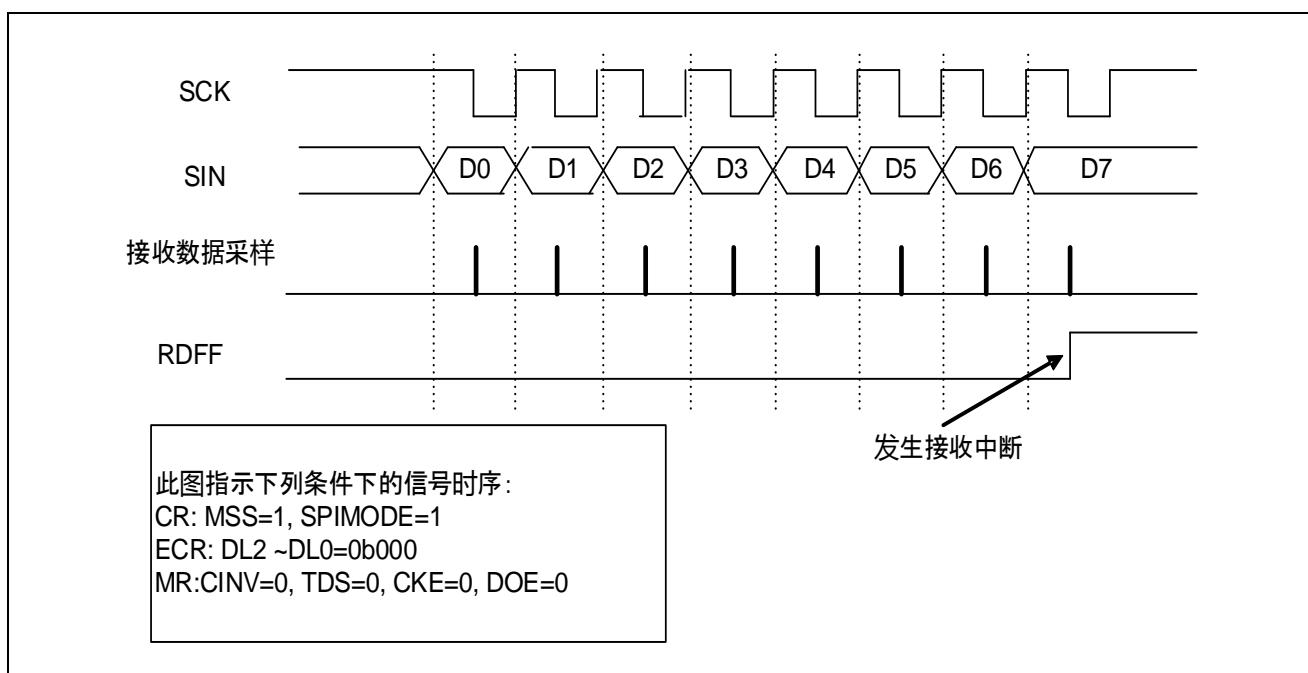
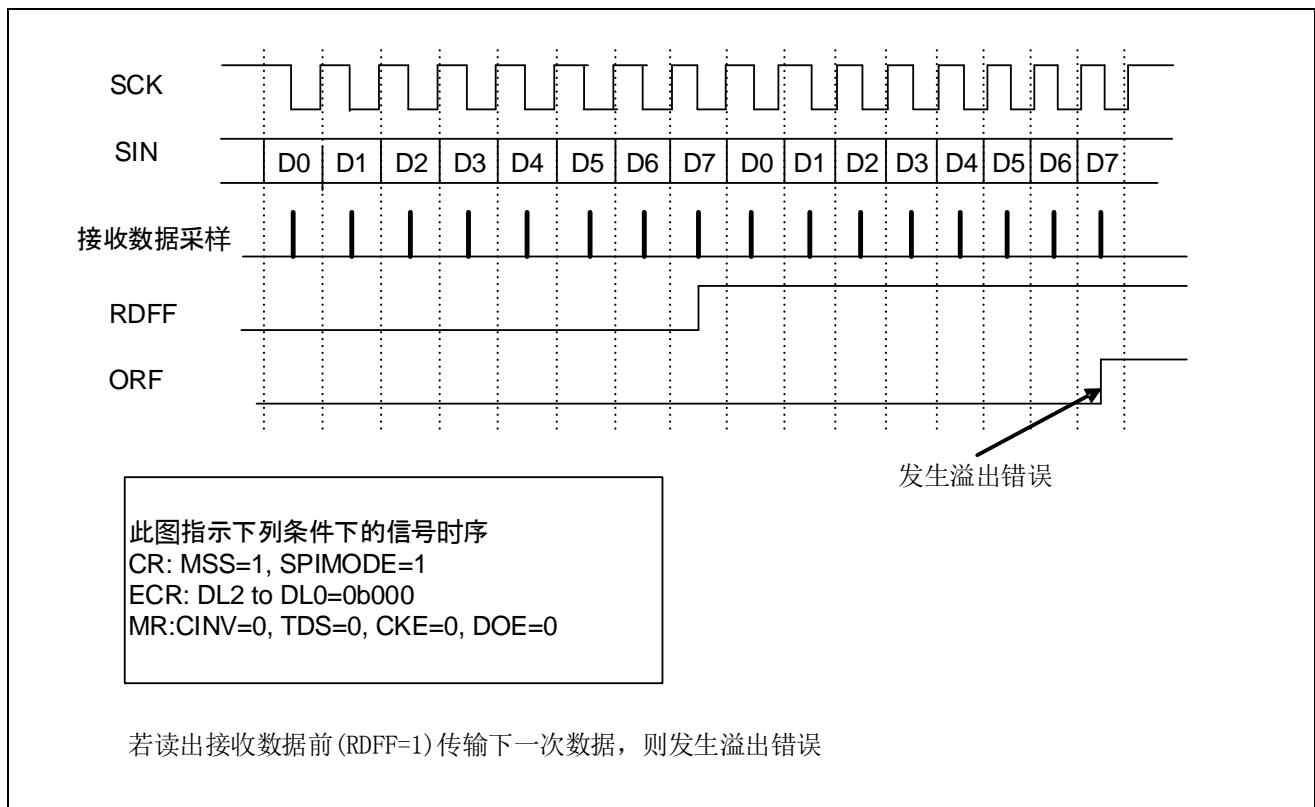


图 33.2-2 ORF (溢出错误)标志位设置时序



33.2.2 发送中断产生和标志位设置时序

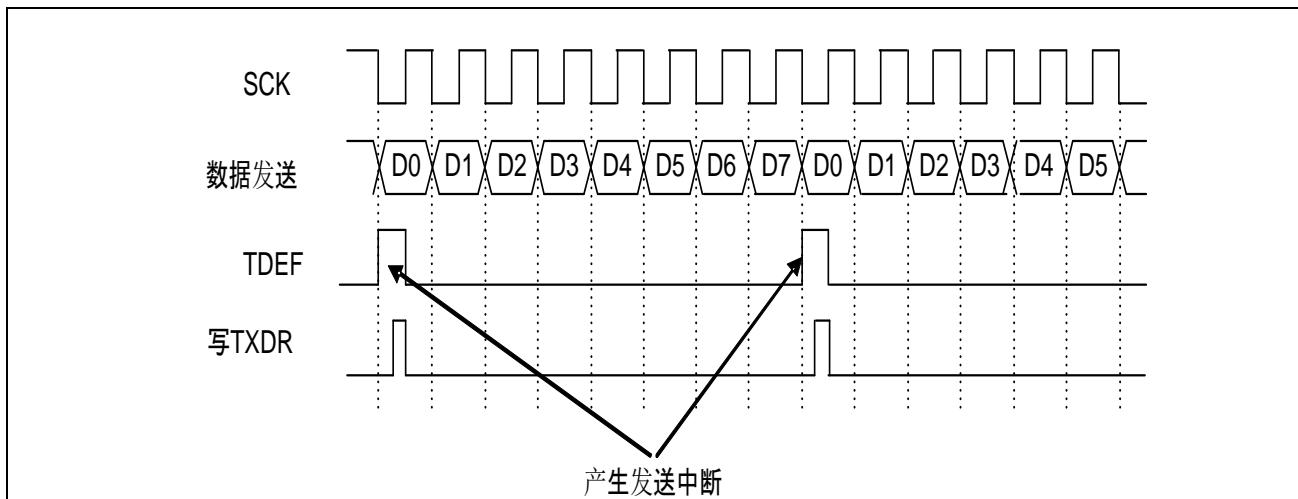
若发送数据从发送数据寄存器(TXDR)传送到发送移位寄存器(SR:TDEF=1)后，发送开始时或未执行发送操作(SR:TBIF=1)时，会产生发送中断。

发送中断产生和标志位设置时序

- 发送数据空标志(SR:TDEF)的设置时序

发送数据寄存器(TXDR)的数据传送到发送移位寄存器后，下一个数据可以写入 TXDR (SR:TDEF=1)。此时，若使能发送中断(CR:TXIE=1)，则产生发送中断。因为 SR:TDEF 位是只读，数据写入发送数据寄存器(TXDR)时可使 SR:TDEF 位清“0”。

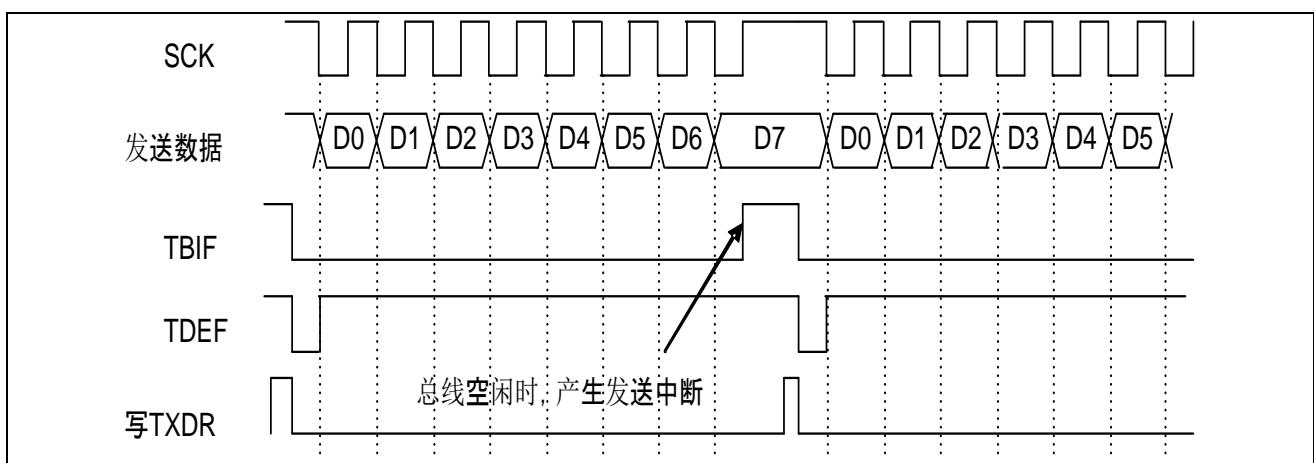
图 33.2-3 发送数据空标志(SR:TDEF)的设置时序



- 发送总线空闲标志(SR:TBIF)的设置时序

因发送数据寄存器为空(SR:TDEF=1)且无数据发送时，SR:TBIF 位置“1”。此时，使能发送总线空闲中断(CR:TBIFE=1)，则产生发送中断。当发送数据写入发送数据寄存器(TXDR)，会清零 SR:TBIF 位和发送中断请求。

图 33.2-4 发送总线空闲标志(TBIF)的设置时序 (CSCR:CSEN0=0, EACR:SYNTE=0)



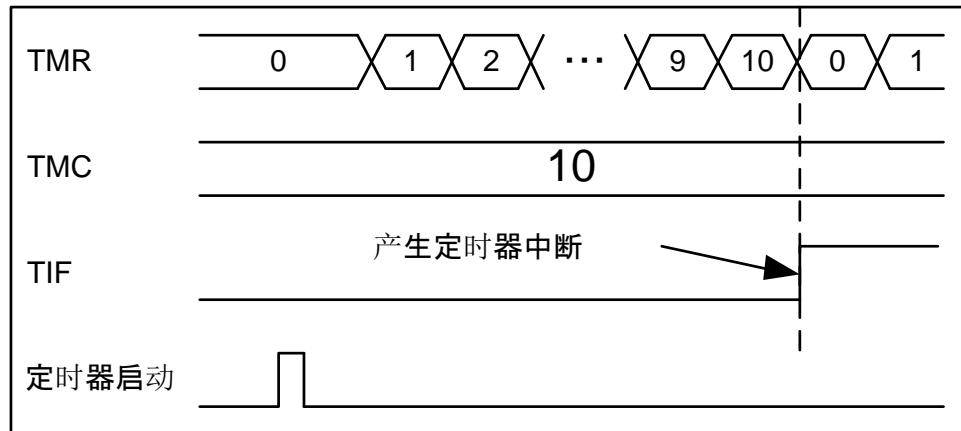
33.2.3 定时器中断产生和标志位设置时序

当串行定时器寄存器(TMR)的值与串行定时器比较寄存器(TMC)的值相同时产生定时器中断。

定时器中断产生和标志位设置时序

当串行定时器寄存器(TMR)的值与串行定时器比较寄存器(TMC)的值相同时，定时器中断标志位(EACR:TIF)被置“1”。此时，如果定时器中断使能(EACR:TXIE=1)，则产生状态中断。

图 33.2-5 定时中断发生时序



33.2.4 片选错误中断产生和标志位设置时序

主机模式(CR:MSS=0)时, 仅当发送帧计数值不大于发送字节数寄存器 TBYTE 的设置值, 且数据发送寄存器值不存在有效数据时, 才产生片选错误。从机(CR:MSS=1)模式下, 数据发送时, 如果片选引脚无效将发生片选错误。

片选错误中断产生和标志位设置时序

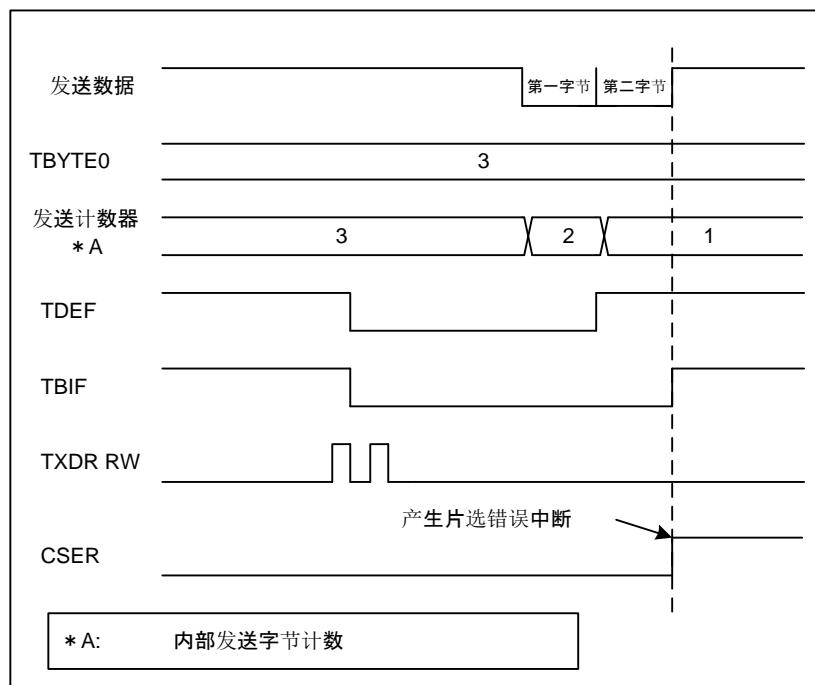
- 主机模式(CR:MSS=0)

满足下列条件之一时, 如果在 TBYTE 设置值的数据帧(SR:TDEF=1)之前, 使能发送字节错误(TBEE=1)且发送数据寄存器不存在有效数据(SR:TDEF=1), 则发生片选错误:

- 使用片选功能
- 使用串行计数器做同步数据传输

此时, 如果片选错误中断使能位使能 (EACR:CSEIE=1), 则产生片选错误中断请求。

图 33.2-6 片选错误中断产生时序(CSCR:CSEN0=0, EACR:SYNTE=1)



注意:

- 如果使用片选功能, 从发生片选错误开始直到片选取消选定时间结束后片选错误标志 (EACR:CSER) 置“1”。此外, 在保持时间内, 向发送数据寄存器(TXDR)写发送数据时而发送操作未开始, 当片选取消选定时间过去后片选错误标志(EACR:CSER) 置“1”。
- 如果片选错误标志(EACR:CSER) 置“1”, 即使向发送数据寄存器(TXDR)写发送数据, 发送操作也不会开始。
- 同步传输模式过程中, 如果片选错误标志(EACR:CSER) 置“1”, 即使下面条件满足发

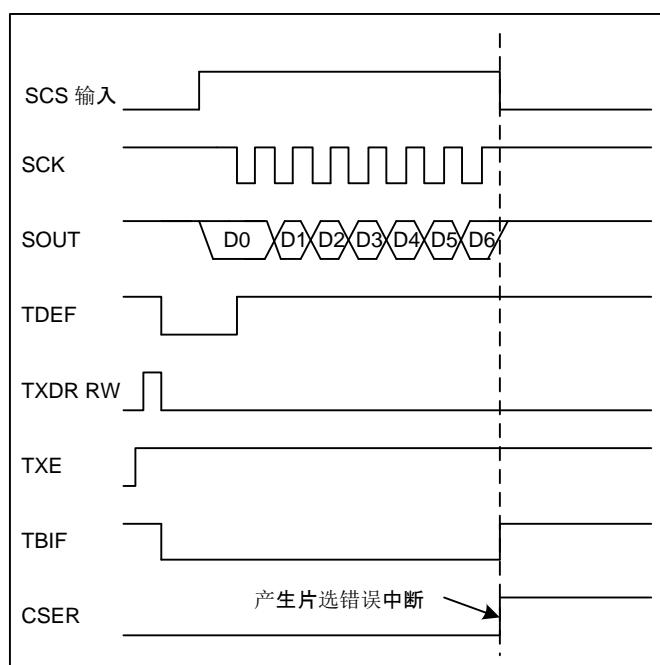
送操作也不会开始:

串行定时器计数值 (TMR)与串行定时器比较寄存器设定值相等。

- 从机模式 (CR:MSS=1)

当发送(SR:TBIF=1)时, 片选引脚变得无效将发生片选错误。此时, 如果片选错误中断使能(EACR:CSEIE=1), 则产生片选错误中断请求。

图 33.2-7 片选错误发生时序(CSLVS=0, :SPIMODE=0)



33.3 操作

数据传输采用时钟同步方式。

33.3.1 SPI 传输(I)

特征

	项目								说明							
1	串行时钟(SCK)的信号检测电平								"H"高电平							
2	发送数据输出时序								SCK 上升沿							
3	接收数据采样								SCK 下降沿							
4	数据长度								~16 位							

寄存器设置

SPI 传输(I)时必要的寄存器设置值如下所示：

表 33.3-1 SPI 传输(I) 寄存器设置

	位 15 14	位 13 14	位 12 13	位 11 12	位 10 11	位 9 10	位 8 11	位 7 12	位 6 13	位 5 14	位 4 15	位 3 16	位 2 17	位 1 18	位 0 19	
CR/ MR	PCL	MS S	SPIM ODE	RX IE	TXI E	TBIF E	RX E	TX E	MO DE2	MOD E1	MO DE0	-	CINV	TD S	CK E	DO E
	0	1/0	1	*	*	*	*	*	0	1	0	0	*	1/0	*	
SR/ ECR	REC LR	-	-	AW C	OR F	RDF F	TD EF	TBI F	SOP S	DL3	CSF E	DTR WS1	DTR WS0	DL 2	DL 1	DL 0
	0	-	-	*	-	-	-	-	0	*	-	*	*	*	*	
TXDR 1/0	D15	D1 4	D13	D1 2	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
RXDR 1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	D31	D3 0	D29	D2 8	D27	D26	D2 5	D2 4	D23	D22	D21	D20	D19	D1 8	D1 7	D1 6
TXDR 3/2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
BRS1/ BRS0	-	B14	B13	B1 2	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

1: 置"1"

0: 清"0"

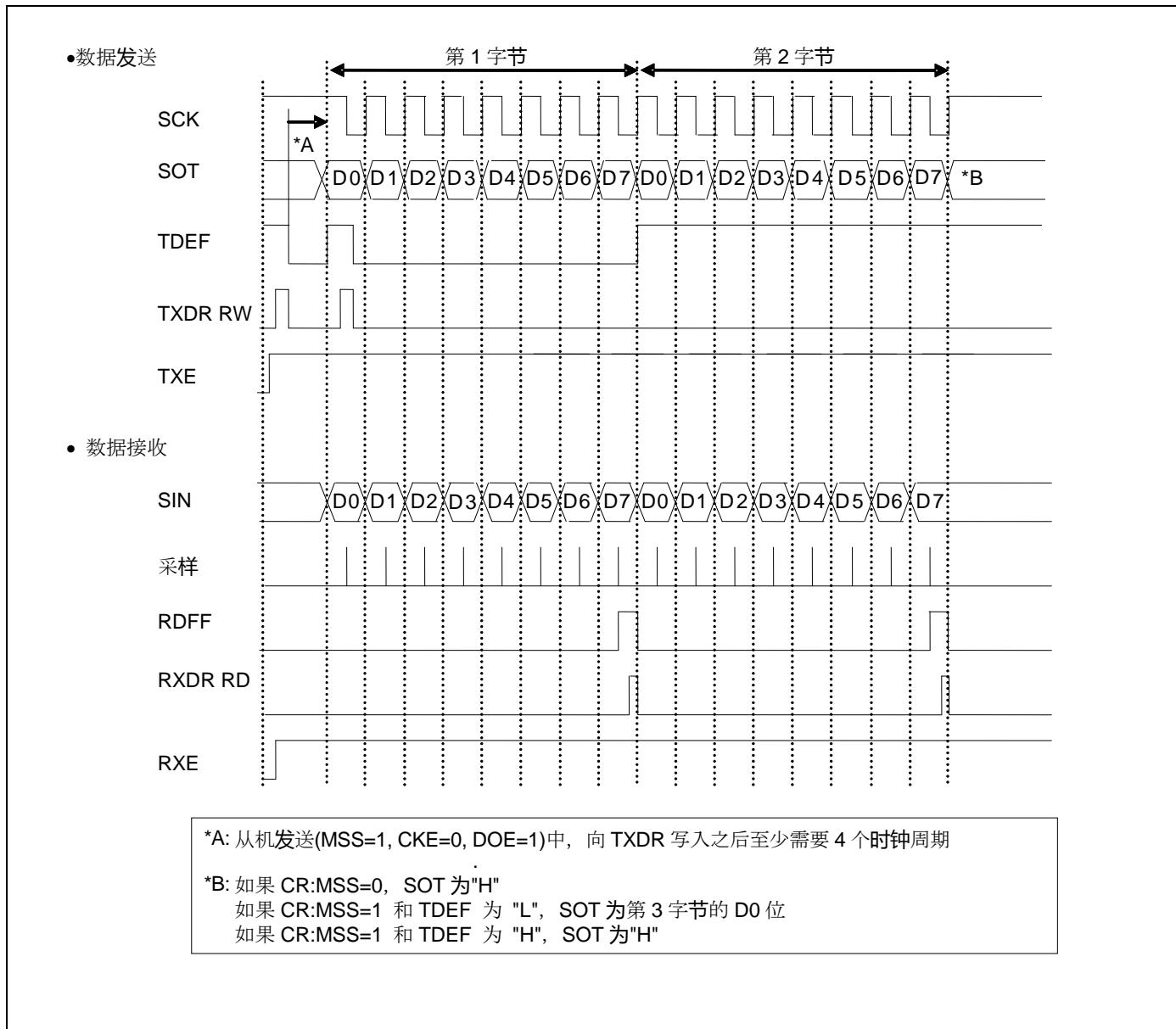
*: 用户自定义值

注意：

上述位的设置值(1/0)因主机模式操作、从机模式操作而异。按照下面进行设置：

- 主机模式操作时: CR:MSS=0, MR:CKE=1
- 从机模式操作时: CR:MSS=1, MR:CKE=0

SPI 传输(I)时序图(不使用片选引脚)



主机模式操作 (CR:MSS=0, MR:CKE=1)

- 数据发送
 - 若使能串行数据输出(MR:DOE=1), 并使能发送操作(CR:TXE=1)和禁止接收操作(CR:RXE=0), 且向 TXDR 写发送数据时, SR:TDEF=0。此后, 与串行时钟(SCK)输出的

上升沿同步，输出发送数据。

- 输出第一位发送数据时，SR:TDEF 位设置成 1。因此，如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。

- 数据接收

- 若禁止串行数据输出(MR:DOE=0)，并使能发送操作(CR:TXE=1)和使能接收操作(CR:RXE=1)，且向 TXDR 写入虚设的无效数据，则可在串行时钟(SCK)输出的下降沿处对接收数据进行采样。
- 接收最后位时，则 SR:RDFF=1。此时，如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后，SR:RDFF 清"0"。

注意：

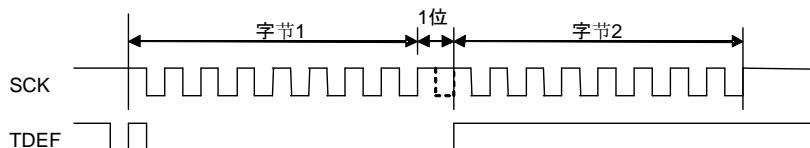
- 如果只进行接收操作，则在 TXDR 写入虚设数据，以输出串行时钟(SCK)。

- 收/发操作

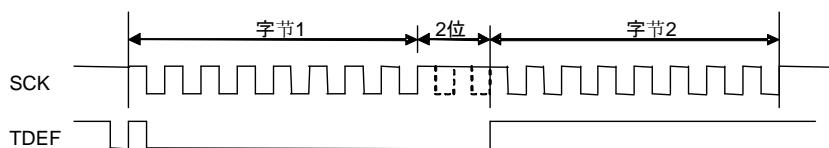
- 要同时进行收/发操作，须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
- 向 TXDR 写发送数据，则 SR:TDEF=0。之后，与串行时钟(SCK)输出的上升沿同步，输出发送数据。输出第一位发送数据时，SR:TDEF=1。如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 在串行时钟(SCK)输出下降沿时对接收数据进行采样。接收到接收数据的最后位时，SR:RDFF=1。如果使能接收中断(CR:RXIE=1)则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。

- 连续数据发送或接收等待操作
 - 对连续数据发送或接收进行(ECR:DTRWS1, ECR.DTRWS0)=(0, 0)以外的设置时，在帧之间将插入一个等待时间。

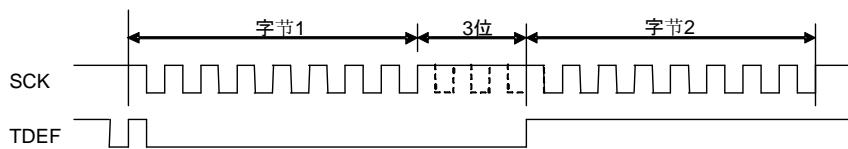
- ECR:DTRWS1, ECR:DTRWS0=01 (主控模式操作)



- ECR:DTRWS1, ECR:DTRWS0=10 (主控模式操作)



- ECR:DTRWS1, ECR:DTRWS0=11 主控模式操作)



从机模式操作 (CR:MSS=1, MR:CKE=0)

- 数据发送

- 若使能串行数据输出(MR:DOE=1)和使能发送操作(CR:TXE=1)，且向 TXDR 写入发送数据，则 SR:TDEF=0。之后，与串行时钟(SCK)输出的上升沿同步，输出发送数据。
- 若输出发送数据的第一位，则 SR:TDEF=1。如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。

注意：

- 若使能发送操作(CR:TXE=1)后，并在串行时钟(SCK)信号检测电平以外的时间向 TXDR 写入第一个发送数据，则不能输出第一个数据位，且数据发送可能失败。使能发送操作(CR:TXE=1)后，须在串行时钟(SCK)信号检测电平时向 TXDR 写入第一个发送数据。

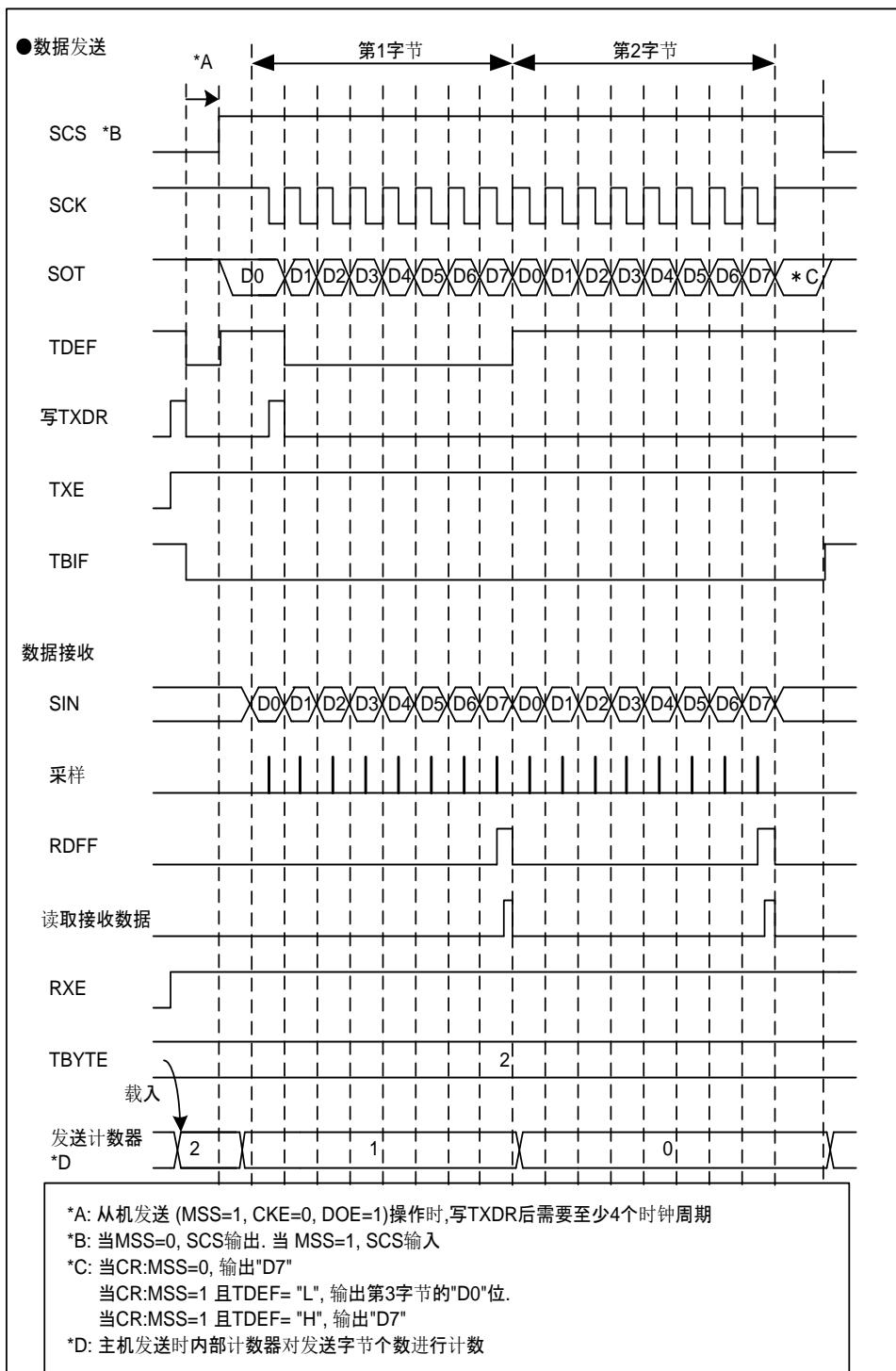
- 数据接收

- 若禁止串行数据输出(MR:DOE=0)和使能接收操作(CR:RXE=1)，在串行时钟输入(SCK)下降沿处对接收数据进行采样。
- 接收最后一位时，则 SR:RDFF=1。如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。

此时，可读出接收数据(RXDR)。

- 读取接收数据(RXDR)后，SR:RDFF 清"0"。
- 收/发操作
 - 要同时进行收/发操作，须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
 - 向 TXDR 写发送数据，则 SR:TDEF=0。此后，与串行时钟(SCK)输入的上升沿同步，输出发送数据。若输出发送数据的第 1 位，SR:TDEF=1，如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
 - 在串行时钟(SCK)输入下降沿处对接收数据进行采样。接收到接收数据的最后位时，SR:RDFF=1，如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。
- 从数据接收到数据发送的连续切换
 - 禁止串行数据输出(MR:DOE=0)，使能接收中断(CR:RXIE=1)，使能接收操作(CR:RXE=1)和使能发送操作(CR:TXE=1)。在串行时钟(SCK)的信号检测电平时向 TXDR 写入虚设无效数据，在串行时钟输入(SCK)下降沿处对接收数据进行采样。
 - 要连续进行接收操作，在接收中断请求后到下一个串行时钟(SCK)上升时的时间内向 TXDR 写入虚设数据。
 - 若要从接收操作切换到发送操作时，从接收中断请求后到下一次串行时钟(SCK)上升时使能串行数据输出(MR:DOE=1)，禁止接收中断(CR:RXIE=0)和禁止接收操作(CR:RXE=0)，向 TXDR 写入发送数据且接收操作结束后，输出发送数据与串行时钟的上升沿同步。

SPI 传输 (I) 时序图(有片选引脚)



主机模式操作 (CR:MSS=0, MR:CKE=1, CSCR:CSOE=1, CSCR:CSENn*=1)

*: “n” 是所使用的片选引脚编号。

● 数据发送

- 若使能串行数据输出(MR:DOE=1), 并使能发送操作(CR:TXE=1)和禁止接收操作(CR:RXE=0), 且向 TXDR 写发送数据, 则 SR:TDEF=0。据此, 输出第 1 位的同时片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。串行时钟输出后, 与串行时钟(SCK)输出的上升沿同步, 输出发送数据。
- 输出第一位发送数据时, SR:TDEF 位设置为 1。因此, 若使能发送中断(CR:TXIE=1), 则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。
- 当完成 TBYTE 所设置的数据发送次数后, 串行时钟停止。
- 当串行时钟停止后, 等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而, 如果片选引脚使能保持 (CSCR:CSAH=1), 片选引脚(SCS)一直保持有效状态。

● 数据接收

- 若禁止串行数据输出(MR:DOE=0), 并使能发送操作(CR:TXE=1)和使能接收操作(CR:RXE=1), 并向 TXDR 写入虚设数据, 则片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。之后, 在串行时钟(SCK)输出的下降沿处对接收数据进行采样。
- 接收最后位时, 则 SR:RDFF=1。如果使能接收中断(CR:RXIE=1), 则输出接收中断请求。
- 此时, 可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后, SR:RDFF 清"0"。
- 当完成 TBYTE 所设置的数据接收次数后, 串行时钟停止。
- 当串行时钟停止后, 等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而, 如果片选引脚使能保持 (CSCR:CSAH=1), 片选引脚(SCS)一直保持有效状态。

注意:

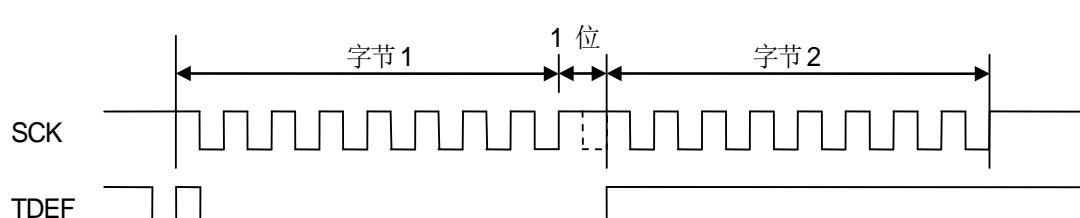
- 如果只进行接收操作, 须向 TXDR 写虚设数据以输出串行时钟(SCK)。

● 收/发操作

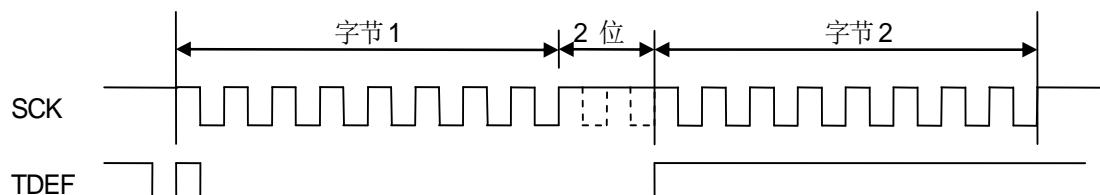
- 要同时进行收/发操作, 须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
- 向 TXDR 写发送数据, 则 SR:TDEF=0。据此, 输出第 1 位的同时片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。串行时钟输出后, 与串行时钟(SCK)输出的上升沿同步, 输出发送数据。输出第一位发送数据时, SR:TDEF=1, 如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。

- 在串行时钟(SCK)输出下降沿对接收数据进行采样。接收到接收数据的最后位时，SR:RDFF=1。如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。
- 当完成 TBYTE 所设置的数据接收次数后，串行时钟停止。
- 当串行时钟停止后，等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而，如果片选引脚使能保持 (CSCR:CSAH=1)，片选引脚(SCS)一直保持有效状态。
- 连续数据发送或接收等待操作
 - 对连续数据发送或接收进行(ECR:DTRWS1, ECR.DTRWS0)=(0, 0)以外的设置时，在帧间将插入一个等待。

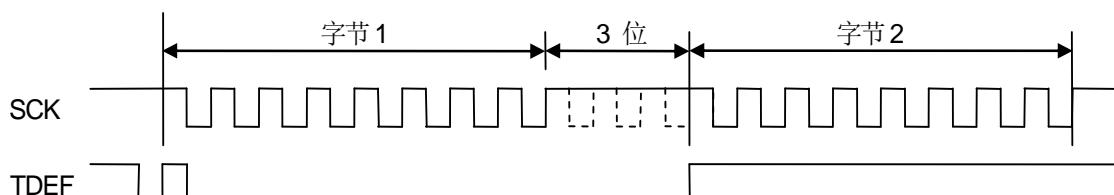
■ ECR.DTRWS1=0, ECR.DTRWS0=1 (主机模式操作)



■ ECR.DTRWS1=1, ECR.DTRWS0=0 (主机模式操作)



■ ECR.DTRWS1=1, ECR.DTRWS0=1 (主机模式操作)



从机模式操作 (CR:MSS=1, MR:CKE=0, CSCR:CSEN=1, CSCR:CSAH=0)

- 数据发送
 - 若使能串行数据输出(MR:DOE=1)和使能发送操作(CR:TXE=1)，且向 TXDR 写入发送数据，则 SR:TDEF=0。
 - 当片选引脚(SCS)变得有效时，启动发送数据输出，与串行时钟(SCK)输出的上升沿同步，

输出发送数据。

- 若输出发送数据的第 1 位，则 SR:TDEF=1。如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 当片选引脚(SCS)变得无效时,数据发送停止且串行输出引脚(SOT)变为高“H”.

注意：

- 如果只进行接收操作，须向 TXDR 写虚设无效数据以输出串行时钟(SCK)。若使能发送操作(CR:TXE=1)后，并在串行时钟(SCK)信号检测电平以外的时间向 TXDR 写入第一个发送数据，则不能输出第一个数据位，且数据发送可能失败。使能发送操作(CR:TXE=1)后，须在串行时钟(SCK)信号检测电平时向 TXDR 写入第一个发送数据。

- 数据接收

- 若禁止串行数据输出(MR:DOE=0)和使能接收操作(CR:RXE=1)，当片选引脚(SCS)变得有效时，在串行时钟输入(SCK)下降沿处对接收数据进行采样。
- 接收最后位时，则 SR:RDFF=1。如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后，SR:RDFF 清"0"。
- 如果片选引脚(SCS)变得无效时,数据接收停止。

- 收/发操作

- 要同时进行收/发操作，须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)，且向 TXDR 写发送数据，则 SR:TDEF=0，之后当片选引脚(SCS)有效时,数据发送启动。在片选引脚建立时间过去后，与串行时钟(SCK)输入的上升沿同步，开始输出发送数据。当输出发送数据的第 1 位时，SR:TDEF=1，如果使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 在串行时钟(SCK)输入下降沿处对接收数据进行采样。接收到接收数据的最后一位时，SR:RDFF=1，如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。
- 当片选引脚(SCS)变得无效时,串行时钟输出停止且串行输出引脚(SOT)变为高“H”。

33.3.2 SPI 传输 (II)

特征

	项目	说明
1	串行时钟(SCK)的信号检测电平	"L"低电平
2	发送数据输出时序	SCK 下降沿
3	接收数据的采样	SCK 上升沿
4	数据长度	~ 16 位

寄存器设置

SPI 传输(II)时必要的寄存器设置值如下所示

表 33.3-2 SPI 传输(II)寄存器设置

	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CR/ MR	PCL	MS S	SPIM ODE	RX IE	TX IE	TBIF E	RX E	TX E	MO DE2	MO DE1	MOD E0	-	CINV	TD S	CK E	DO E
	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SR/ ECR	REC LR	-	-	AW C	OR F	RDF F	TD EF	TBI F	SOP S	DL3	CSFE	DTR WS1	DTR WS0	DL 2	DL 1	DL 0
	0	-	-	*	-	-	-	-	0	*	-	*	*	*	*	*
TXDR 1/0	D15	D1 4	D13	D1 2	D1 1	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RXDR 1/0	D31	D3 0	D29	D2 8	D2 7	D26	D2 5	D2 4	D23	D22	D21	D20	D19	D1 8	D1 7	D1 6
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TXDR 3/2	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RXDR 3/2	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
BRS1/ BRS0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

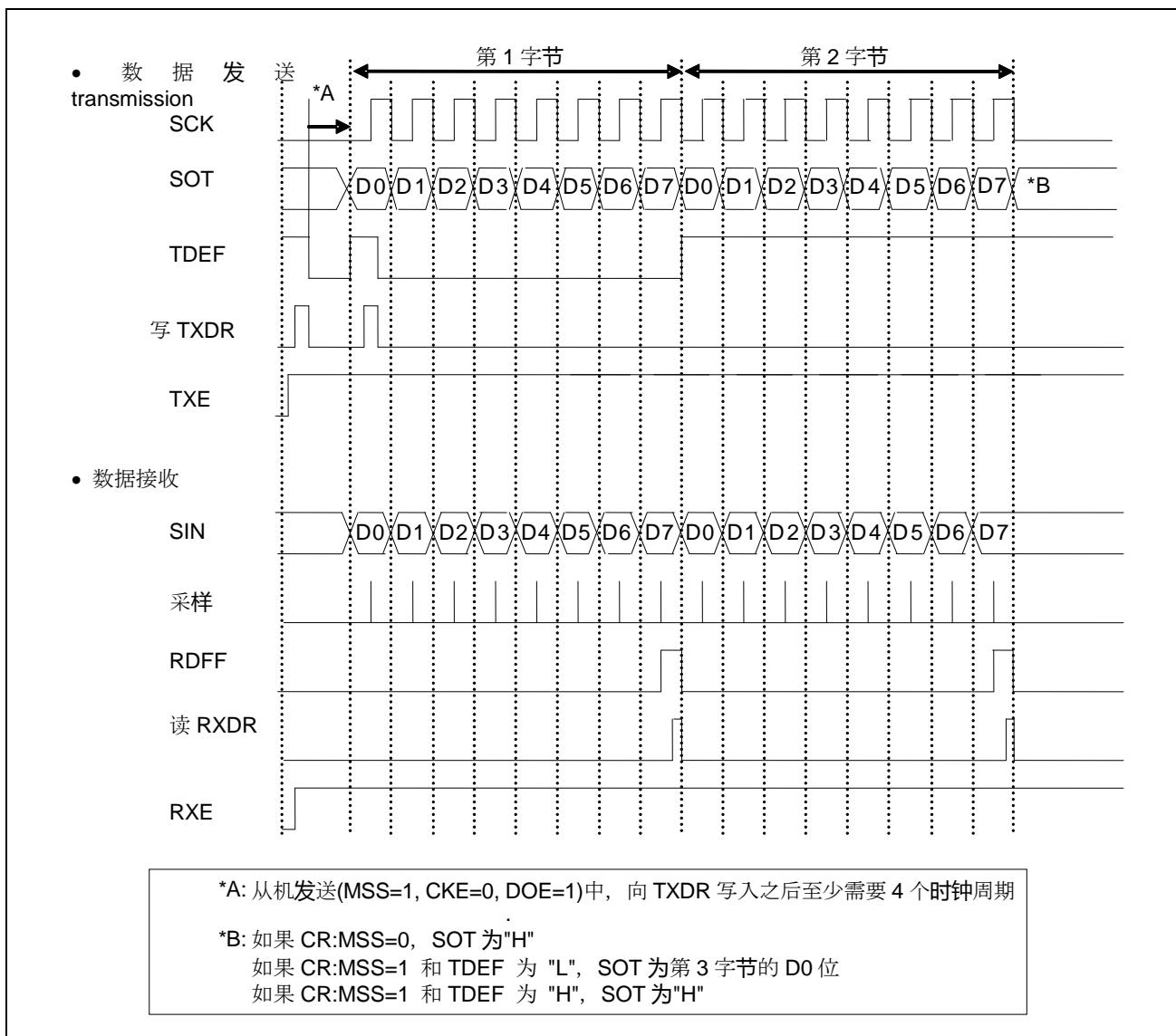
1: 置"1", 0:清"0", *:用户自定义值

注意：

上述位的设置值(1/0)因主机模式操作、从机模式操作而异。按照下面进行设置：

- 主机模式操作时: CR:MSS=0, MR:CKE=1
- 从机模式操作时: CR:MSS=1, MR:CKE=0

SPI 传输(II)时序图(无片选引脚)



主机模式操作(CR:MSS=0, MR:CKE=1)

- 数据发送

- 使能串行数据输出(MR:DOE=1), 使能发送操作(CR:TXE=1)和禁止接收操作(CR:RXE=0), 且向 TXDR 写发送数据, 则 SR:TDEF=0。据此, 与串行时钟(SCK)输出下降沿同步, 输出发送数据。
- 在第一个数据输出时, SR:TDEF 设置为 1。因此, 若使能发送中断(CR:TXIE=1), 则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。

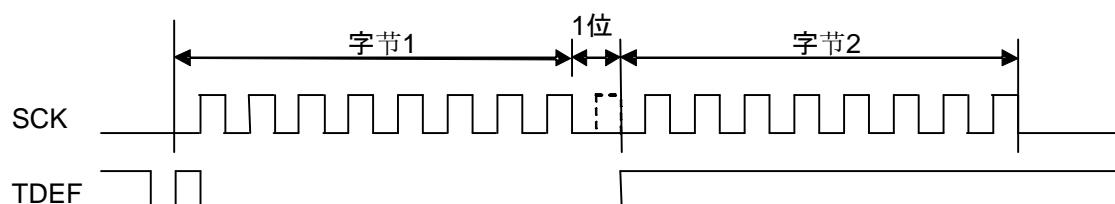
注意:

- 如果只进行接收操作, 须向 TXDR 写虚设数据以输出串行时钟(SCK)。

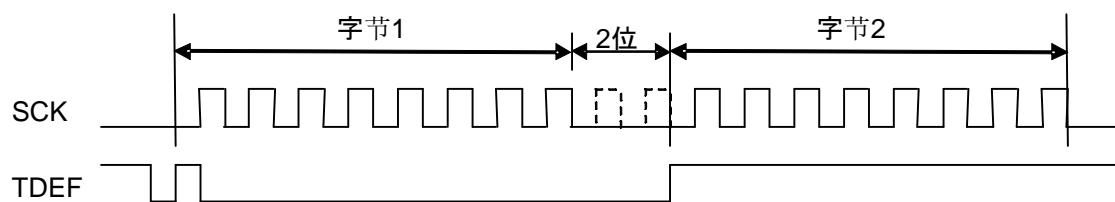
- 数据接收

- 禁止串行数据输出(MR:DOE=0), 使能发送操作(CR:TXE=1)和使能接收操作(CR:RXE=1), 并向 TXDR 写入虚设数据, 则在串行时钟输出(SCK)的上升沿对接收数据进行采样。
 - 接收最后一位时, SR:RDFF=1。若使能接收中断(CR:RXIE=1), 则输出接收中断请求。此时, 可读出接收数据(RXDR)。
 - 读取接收数据(RXDR)后, SR:RDFF 清"0"。
- 收/发操作
- 要同时进行收/发操作, 须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
 - 向 TXDR 写发送数据, 则 SR:TDEF=0。此后, 与串行时钟(SCK)输出下降沿同步, 输出发送数据。在第一个数据输出时, SR:TDEF=1, 若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。
在串行时钟(SCK)输出的上升沿处对接收数据进行采样。接收到接收数据的最后一位时, SR:RDFF=1。若使能接收中断(CR:RXIE=1), 则输出接收中断请求。此时, 可读出接收数据(RXDR)。读出接收数据后, SR:RDFF 清"0"。

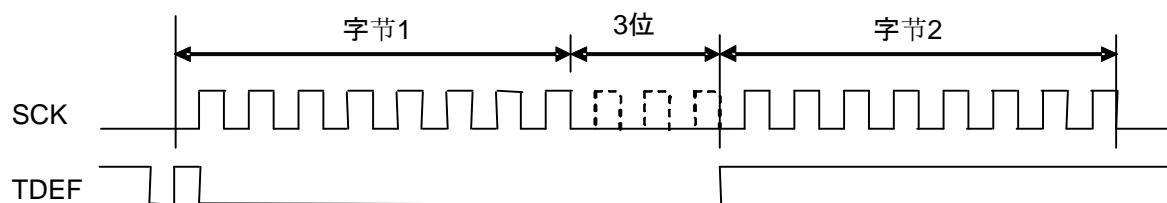
- 连续数据发送或接收等待操作
 - 对连续数据发送或接收进行(ECR:DTRWS1, ECR.DTRWS0)=(0, 0)以外的设置时，在帧间插入一个等待时间。
 - ECR:DTRWS1, ECR:DTRWS0=01 (主机模式操作)



- ECR:DTRWS1, ECR:DTRWS0=10 (主机模式操作)



- ECR:DTRWS1, ECR:DTRWS0=11 (主机模式操作)



从机模式操作(CR:MSS=1, MR:CKE=0)

- 数据发送
 - 使能串行数据输出(MR:DOE=1)和使能发送操作(CR:TXE=1)，且向 TXDR 写入发送数据，则 SR:TDEF=0。之后，与串行时钟(SCK)输入下降沿同步，输出发送数据。
 - 输出发送数据的第 1 位时，SR:TDEF=1。若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。

注意：

- 若使能发送操作(CR:TXE=1)后，并在串行时钟(SCK)信号检测电平以外的时间向 TXDR 写入第一个发送数据，则不能输出第一位数据，且数据发送可能失败。使能发送操作(CR:TXE=1)后，须在串行时钟(SCK)信号检测电平时向 TXDR 写入第一个发送

数据。

- 数据接收

- 禁止串行数据输出(MR:DOE=0)和使能接收操作(CR:RXE=1)，在串行时钟输入(SCK)的上升沿处对接收数据进行采样。
- 接收最后一位时，SR:RDFF=1。若使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后，SR:RDFF 清"0"。

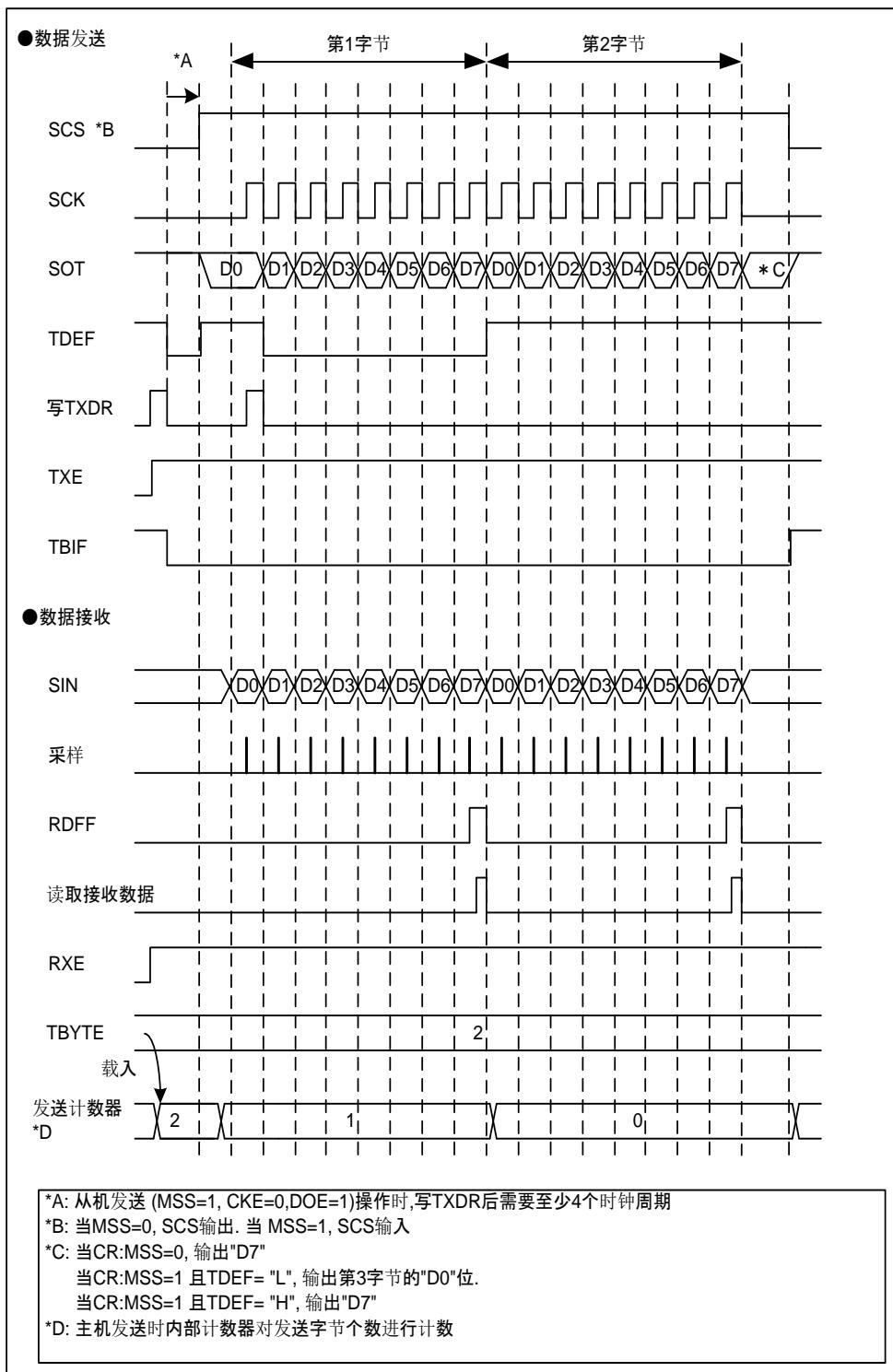
- 收/发操作

- 要同时进行收/发操作，须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
- 向 TXDR 写发送数据，则 SR:TDEF=0。之后，与串行时钟(SCK)输入下降沿同步，输出发送数据。输出发送数据的第 1 位时，SR:TDEF=1，若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 在串行时钟(SCK)输入的上升沿处对接收数据进行采样。接收到接收数据的最后一位时，SR:RDFF=1，若使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。

- 数据接收到数据发送的连续操作

- 禁止串行数据输出(MR:DOE=0)，使能接收中断(CR:RXIE=1)，使能接收操作(CR:RXE=1)和使能发送操作(CR:TXE=1)。在串行时钟(SCK)的信号检测电平时，向 TXDR 写入虚设数据，在串行时钟输入(SCK)上升沿处对接收数据进行采样。
- 要连续进行接收操作，在接收中断请求后到下一个串行时钟(SCK)上升时的时间内向 TXDR 写入虚设数据。
- 若要从接收操作切换到发送操作时，从接收中断请求后到下一次串行时钟(SCK)上升时的时间内使能串行数据输出(MR:DOE=1)，禁止接收中断(CR:RXIE=0)和禁止接收操作(CR:RXE=0)，向 TXDR 写入发送数据且接收操作结束后，与串行时钟的下降沿同步输出发送数据。

SPI 传输(II)时序图(有片选引脚)



主机模式操作(CR:MSS=0, MR:CKE=1, CSCR:CSOE=1, CSCR:CSENn*=1)

*: “n” 是所使用的片选引脚编号

● 数据发送

- 若使能串行数据输出(MR:DOE=1), 并使能发送操作(CR:TXE=1)和禁止接收操作(CR:RXE=0), 且向 TXDR 写发送数据, 则 SR:TDEF=0。据此, 输出第 1 位的同时片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。串行时钟输出后, 与串行时钟(SCK)输出的下降沿同步, 输出发送数据
- 输出第一个数据时, SR:TDEF 设置为 1。因此, 若使能发送中断(CR:TXIE=1), 则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。
- 当完成 TBYTE 所设置的数据发送次数后, 串行时钟停止。
- 当串行时钟停止后, 等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而, 如果片选引脚使能保持 (CSCR:CSAH=1), 片选引脚 (SCS) 会一直保持有效状态。

● 数据接收

- 若禁止串行数据输出(MR:DOE=0), 并使能发送操作(CR:TXE=1)和使能接收操作(CR:RXE=1), 并向 TXDR 写入虚设数据, 片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。之后, 在串行时钟(SCK)输出的上升沿处对接收数据进行采样。
- 接收最后一位时, SR:RDFF=1。若使能接收中断(CR:RXIE=1), 则输出接收中断请求。此时, 可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后, SR:RDFF 清"0"。
- 当完成 TBYTE 所设置的数据接收次数后, 串行时钟停止。
- 当串行时钟停止后, 等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而, 如果片选引脚使能保持(CSCR:CSAH=1), 片选引脚(SCS)会一直保持有效状态。

注意:

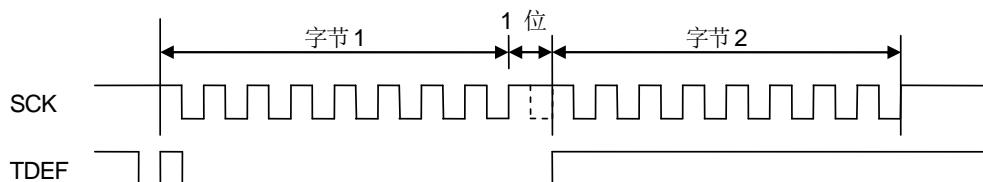
- 要仅进行接收操作, 须向 TXDR 写虚设数据以输出串行时钟(SCK)。

● 收/发操作

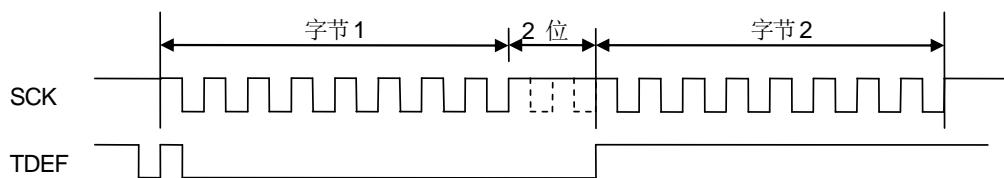
- 要同时进行收/发操作, 须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
- 向 TXDR 写发送数据, 则 SR:TDEF=0。据此, 输出第 1 位的同时片选引脚(SCS)变成有效。此后, 当片选引脚的建立时间过后, 串行时钟开始输出。串行时钟输出后, 与串行时钟(SCK)输出的下降沿同步, 输出发送数据。输出第一个数据时, SR:TDEF=1, 若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时, 可向寄存器写入第 2 个字节的发送数据。

- 在串行时钟(SCK)输出的上升沿对接收数据进行采样。接收到接收数据的最后一位时，SR:RDFF=1。如果使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。
- 当完成 TBYTE 所设置的数据接收次数后，串行时钟停止。
- 当串行时钟停止后，等到片选引脚保持时间之后片选引脚(SCS)变得无效。然而，如果片选引脚使能保持(CSCR:CSAH=1)，片选引脚(SCS)会一直保持有效状态。
- 连续数据发送或接收等待操作
 - 对连续数据发送或接收进行(ECR:DTRWS1, ECR.DTRWS0)=(0, 0)以外的设置时，在帧间将插入一个等待时间。

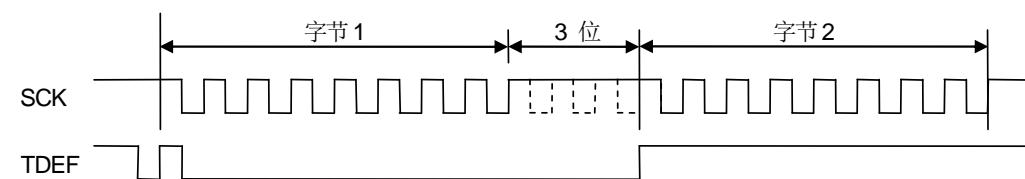
■ ECR.DTRWS1=0, ECR.DTRWS 0=1 (主机模式操作)



■ ECR.DTRWS 1=1, ECR.DTRWS 0=0 (主机模式操作)



■ ECR.DTRWS 1=1, ECR.DTRWS 0=1 (主机模式操作)



从机模式操作 (CR:MSS=1, MR:CKE=0, CSCR:CSEN=1, CSCR:CSAH=0)

- 数据发送
 - 若使能串行数据输出(MR:DOE=1)和使能发送操作(CR:TXE=1)，且向 TXDR 写发送数据，则 SR:TDEF=0。
 - 当片选引脚(SCS)变得有效时，启动发送数据输出，与串行时钟(SCK)输出的下降沿同步，输出发送数据。

- 输出发送数据的第 1 位时， SR:TDEF=1。若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 当片选引脚(SCS)变得无效时,数据发送停止且串行输出引脚(SOT)变为高“H”.

注意：

- 要仅进行接收操作，须向 TXDR 写虚设数据以输出串行时钟(SCK)。若使能发送操作(CR:TXE=1)后，并在串行时钟(SCK)信号检测电平以外的时间向 TXDR 写入第一个发送数据，则不能输出第一个数据位，且数据发送可能失败。使能发送操作(CR:TXE=1)后，须在串行时钟(SCK)信号检测电平时向 TXDR 写入第一个发送数据。

● 数据接收

- 若禁止串行数据输出(MR:DOE=0)和使能接收操作(CR:RXE=1)，当片选引脚(SCS)变得有效时，在串行时钟输入(SCK)上升沿处对接收数据进行采样。
- 接收最后一位时，则 SR:RDFF=1。若使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。
- 读取接收数据(RXDR)后，SR:RDFF 清"0"。
- 如果片选引脚(SCS)变得无效时,数据接收停止。

● 收/发操作

- 要同时进行收/发操作，须使能串行数据输出(MR:DOE=1)并使能收/发操作(CR:TXE, RXE=1)。
- 向 TXDR 写发送数据，则 SR:TDEF=0，之后当片选引脚(SCS)有效,数据收发启动。发送数据在片选引脚建立时间过去后，与串行时钟(SCK)输入的下降沿同步，开始输出发送数据。当输出发送数据的第 1 位时，SR:TDEF=1，若使能发送中断(CR:TXIE=1)则输出发送中断请求。此时，可向寄存器写入第 2 个字节的发送数据。
- 在串行时钟(SCK)输入上升沿处对接收数据进行采样。接收到接收数据的最后一位时，SR:RDFF=1，若使能接收中断(CR:RXIE=1)，则输出接收中断请求。此时，可读出接收数据(RXDR)。读出接收数据后，SR:RDFF 清"0"。
- 当片选引脚(SCS)变得无效时,串行时钟输出停止且串行输出引脚(SOT)变为高“H”。

33.4 串行定时器操作

串行定时器可以用作定时器功能或者同步传输功能。

串行定时器操作

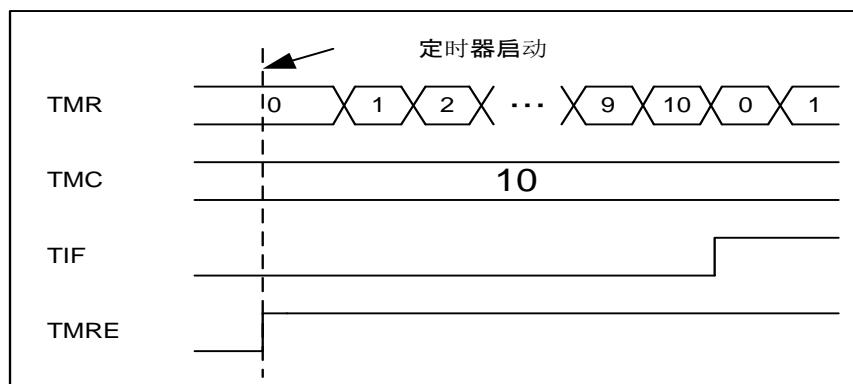
- 串行定时器启动方法

将串行定时器使能位 (EACR:TMRE)置 “1”启动串行定时器

启动串行定时器使能位 (EACR:TMRE)

当串行定时器使能位 (EACR:TMRE)置 “1”后，串行定时器启动，且串行定时器寄存器 (TMR)从 “0”开始计数。

图 33.4-1 启动串行定时器使能位(TMC=10, EACR:SYNTE=0))



- 串行定时器停止方法

将串行定时器使能位 (EACR:TMRE)置“1”停止串行定时器。这时，串行定时器寄存器(TMR)的值将保持。

- 定时器操作

当同步发送使能位 (EACR:SYNTE) 为 “0”时,串行定时器作为定时器工作。

当串行定时器(TMR)的值与串行定时器比较寄存器(TMC)相等时，定时器中断标志 (EACR:TIF) 置“1”，同时串行定时器(TMR)清 “0”并继续计数。

图 33.4-2 定时器操作(TMC=10, EACR:SYNTE=0))

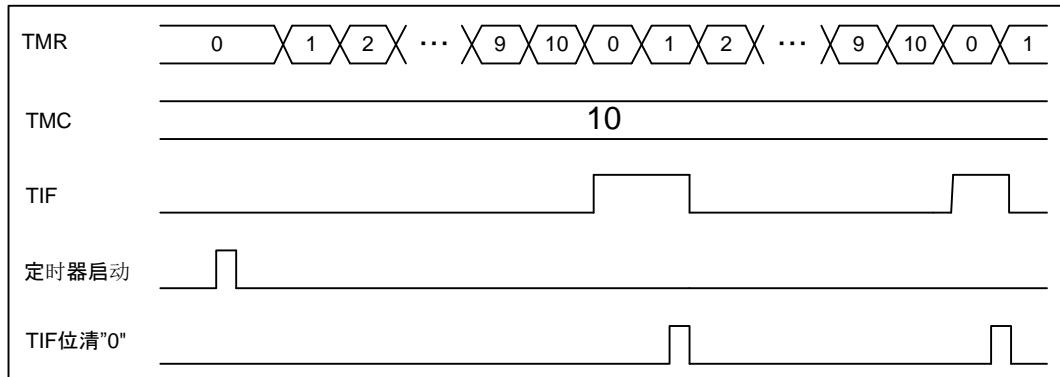


图 33.4-3 串行定时初始化流程

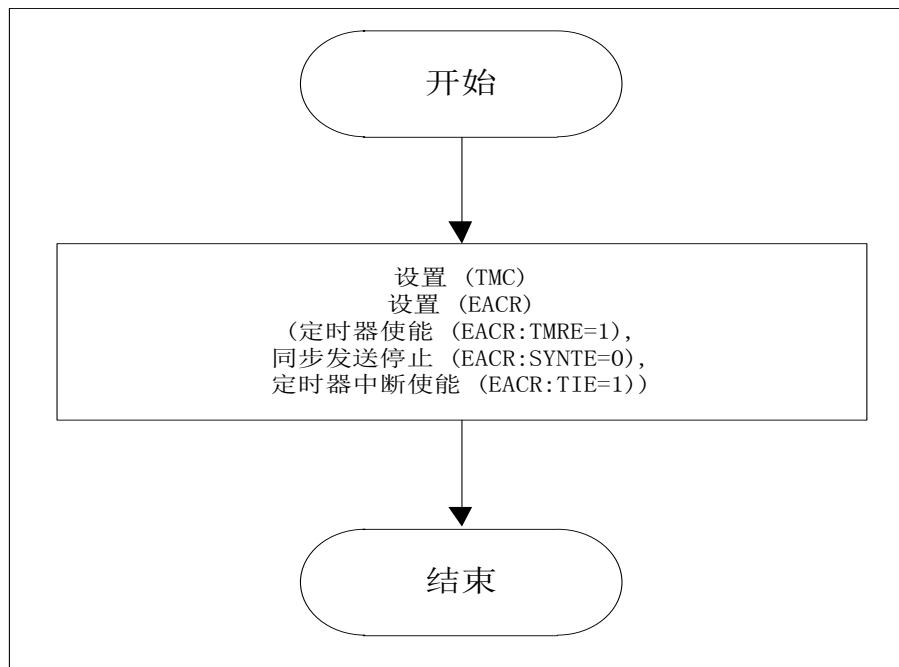
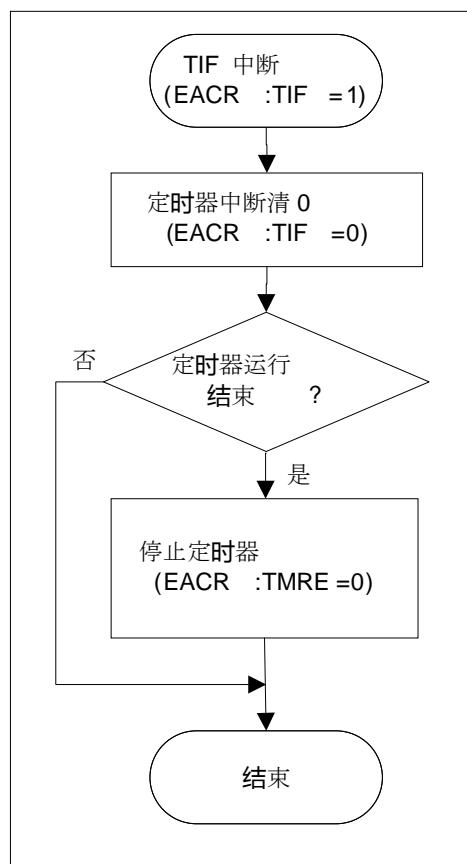


图 33.4-4 串行定时器中断处理流程



注意：

- 下列条件同时满足时，定时中断标志(EACR:TIF)固定为“1”。
- 同步传输模式禁止(EACR:SYNTE="0")时，定时比较寄存器(TMC)的值设为"0x0000"。
- 定时器操作时，定时器操作时钟分频比(EACR:CLKDIV) 设置为"0000"。

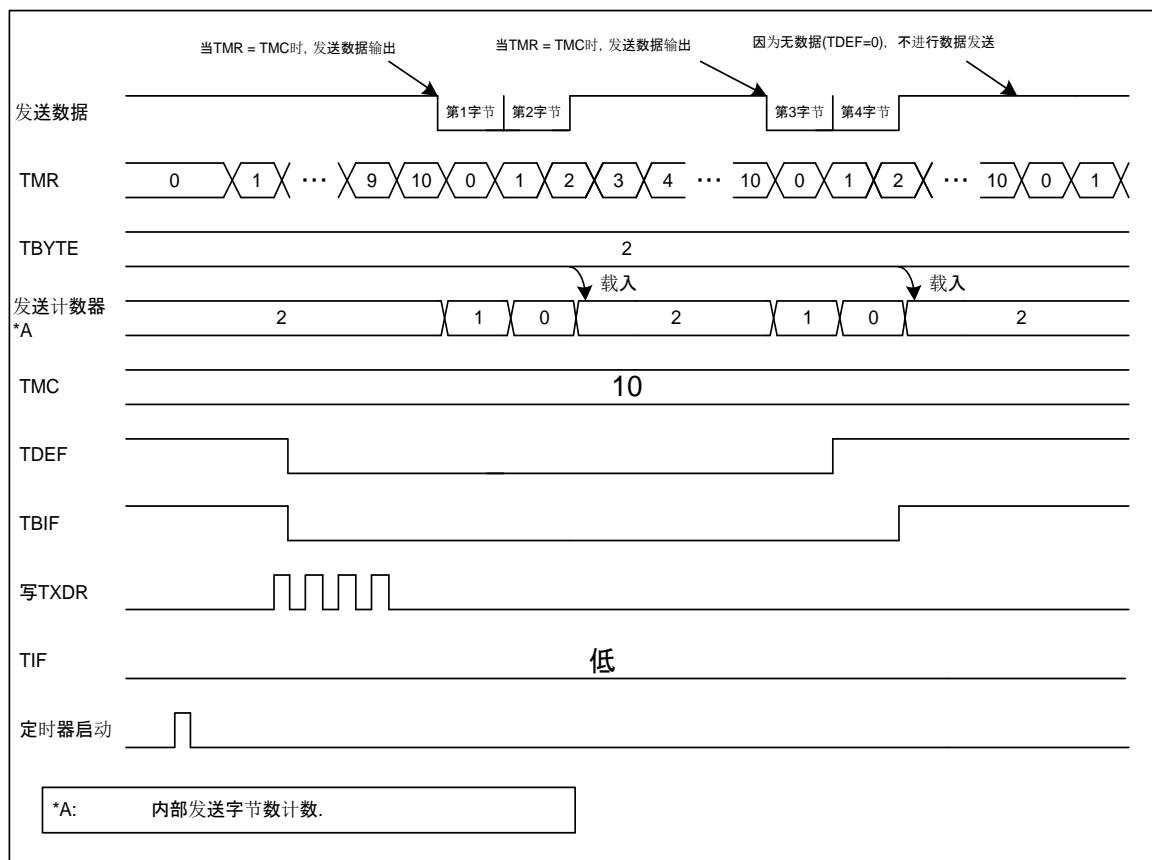
● 定时器同步发送

当同步发送使能位 (EACR:SYNTE) 为“1”时，串行定时器用作同步发送功能。

定时器同步发送的实现流程：

1. 在发送寄存器有数据时(SR:TDEF="0")情况下，当串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等时，发送操作启动的同时串行定时器寄存器(TMR) 清"0"。发送字节数寄存器(TBYTE0)设置的发送次数的数据将被发送。
2. 当字节数寄存器(TBYTE0)设置的数据发送次数完成后，发送操作停止。除非下一次串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等。

图 33.4-5 定时器进行同步发送操作 (TMR=10, TBYTE0=2, EACR:SYNTE=1)



下列条件之一的情况下，即使同步发送使能(EACR:SYNTE=1) 和串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等，同步发送操作也不会执行：

- 当发送操作禁止时 (CR:TXE=0)
- 从机操作模式(CR:MSS=1)
- 发送数据寄存器为空(SR:TDEF=1)

但是，如果发送数据寄存器为空 (SR:TDEF=1)，同步发送使能和串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等，在向发送寄存器写值后，将立即启动数据发送。

即使发送数据寄存器(TXDR)还有值，当已发送数据次数跟字节数寄存器 (TBYTE0) 设置的次数相等，发送操作也将停止，除非下一次串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等。

在当同步发送使能(EACR:SYNTE="1") 的发送(SR:TBIF=0)过程中，串行定时器寄存器(TMR)的值与串行定时器比较寄存器 (TMC) 相等也不影响当前的发送，当前发送操作继续执行，直到满足发送字节数寄存器 (TBYTE0) 设置的数目之后，发送操作才将停止。

发送操作将会下列情况打断：

- 可编程复位 (CR:PCL=1)
- 禁止发送 (CR:TXE=0)
- 片选错误标志发生(EACR:CSER=1)

通过停止串行数据输出 (MR:DOE=0)，使能发送 (CR:TXE=1) 和接收(CR:RXE=1)，并向接收寄存器 TXDR 写入虚设无效数据将执行同步接收操作。

图 33.4-6 定时器同步发送初始化流程

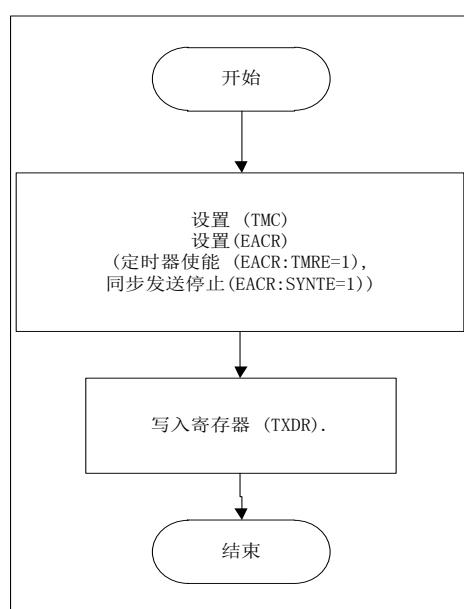
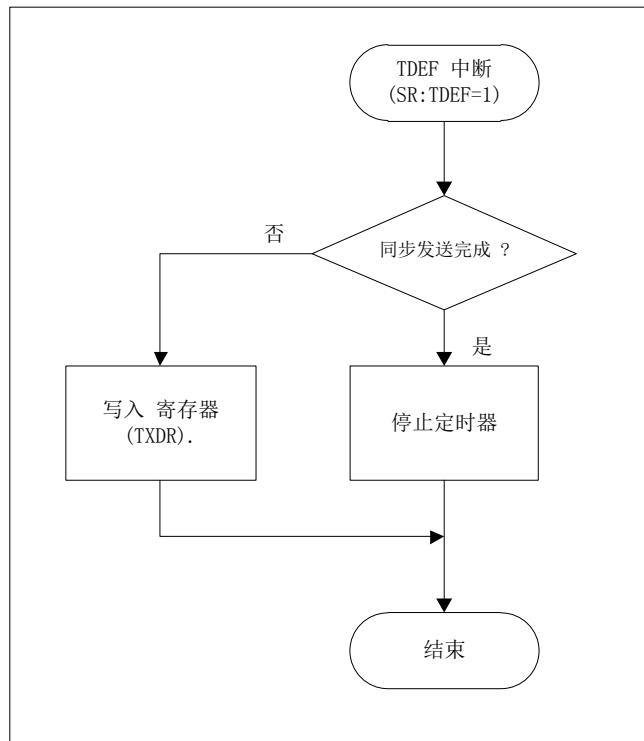


图 33.4-7 定时器同步发送中断控制流程



注意：

当发送寄存器(TXDR)无有效数据(SR:TDEF=1) 之前设置发送字节数寄存器(TBYTE0)，将按下列操作执行：

- 当发送字节错误使能(TBEE=1)时，会产生片选错误，片选错误标志位(EACR:CSER)置“1”。之后即使向发送寄存器(TXDR)写入数据，发送操作也不会执行。
- 当发送字节错误禁止(TBEE=0)时，发送操作将会在向发送寄存器(TXDR)写入数据后开始执行。

33.5 串行片选操作

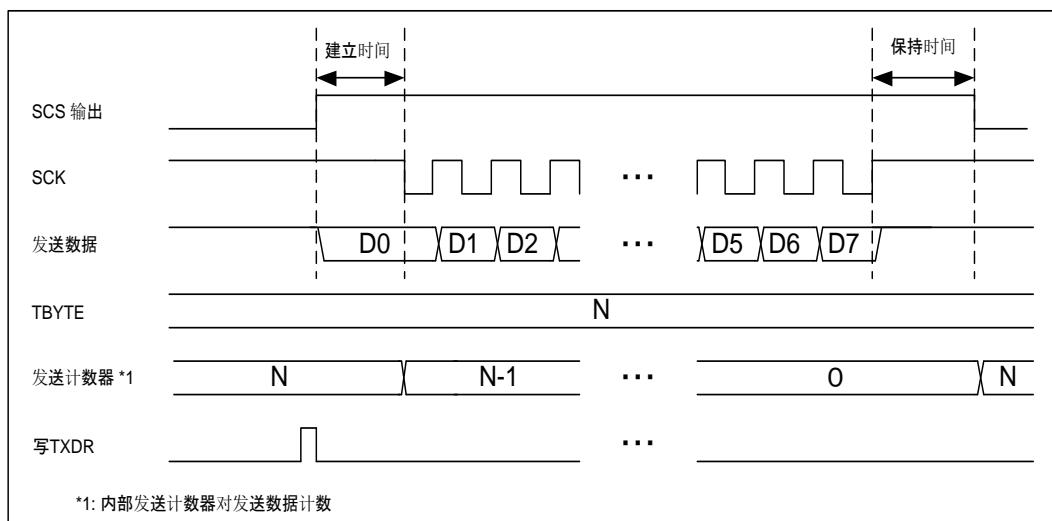
本节介绍片串行片选操作。

主机模式操作(CR:MSS=0)

主机模式(CR:MSS=0)下，片选操作按照下面执行：

1. 当片选操作使能位(CSCR:CSENn="1") 置位并使能发送(CR:TXE="1")，片选引脚有效。
2. 等待片选引脚建立时间过后，启动数据的收/发操作。
3. 当完成发送字节数寄存器(TBYTE)设置次数的数据收/发后，串行时钟停止。
4. 串行时钟停止之后，等待片选信号保持时间过后，片选引脚释放。

图 33.5-1 串行片选操作(主机模式发送(MSS=0), SPI 发送 (SPIMODE=1), CINV=0)



注意：

- 片选引脚有效状态时，如果禁止发送(CR:TXE="0")且软件复位(CR:PCL=1)，片选引脚释放。
- 在无发送数据(SR:TDEF=1)的情况下，片取消选定时间过后，如果片选引脚不保持有效状态 (CSCR:CSAH=0)，片选引脚将释放，发送总线切换到待机状态 (SR:TBIF=1)。
- 主机模式(CR:MSS=0)下设置 CSCR:CSENx 为“0”时，不管片选引脚状态如何，数据收/发操作都将执行。
- 当发送的数据数据帧数小于发送字节数寄存器(TBYTE)设置值时，如果发送寄存器(TXDR)无有效数据(SR:TDEF=1)时，将按照下面操作进行：
 - 如果传输字节错误使能(TBEE=1)，则会产生片选错误(EACR:CSER=1)。片选引脚在片选错误发生(EACR:CSER=1)后经过片选保持时间后将变得无效。片选错误发生标志

(EACR:CSER)置“1”后，即使向发送数据寄存器(TXDR)写入发送数据，发送操作也将停止执行。

- 如果传输字节错误禁止(TBEE=0)，发送操作将暂停，直到向发送数据寄存器(TXDR)写入发送数据。在这整个过程中片选引脚保持有效状态。当向发送数据寄存器(TXDR)写入发送数据后，发送操作将重新启动。

串行片选时序调整

主机模式(CR:MSS=0)下，在片选操作使能(CSCR:CSENn=“1”的情况下，建立时间，保持时间和取消选定时间都可以通过改写串行片选时间寄存器(CSTMG3-0)来调整。

- 建立时间

指从片选引脚有效到串行时钟输出的持续时间，参考图 33.5-2 和图 33.5-3。

此时间可以通过片选建立时间寄存器(CSTMG0:CSSDLY7-0)调整。

- 保持时间

指从串行时钟输出到片选引脚释放的持续时间，参考图 33.5-2 和图 33.5-3。

此时间可以通过片选保持时间寄存器(CSTMG1:CSHDLY7-0)调整。

- 取消选定时间

指从片选引脚释放到片选引脚再次有效之间所需要的最长时间。

在此时间内即使向发送数据寄存器(TXDR)写入了发送数据，片选引脚将保持无效状态直到取消选定时间结束。参考图 32.5-2 和图 32.5-3。

此时间可以通过片选取消选定时间寄存器(CSTMG3:2:CSDS15-0)调整。

图 33.5-2 时序调整(普通模式(SPI MODE=0), CINV=0)

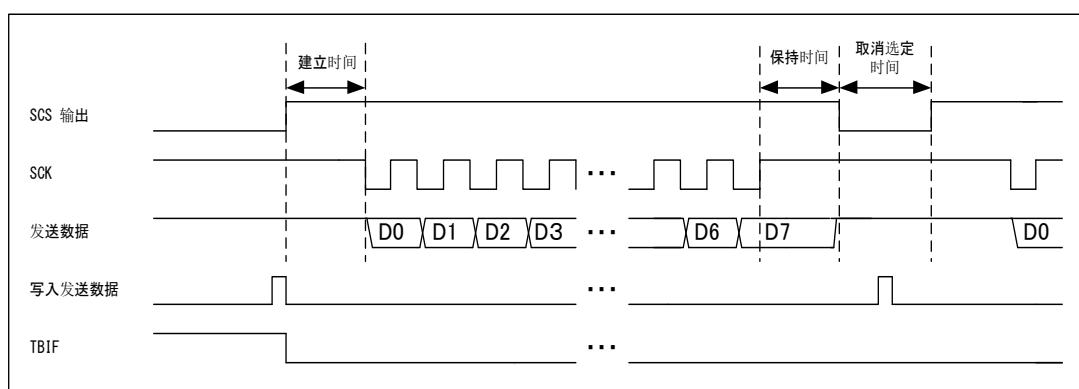
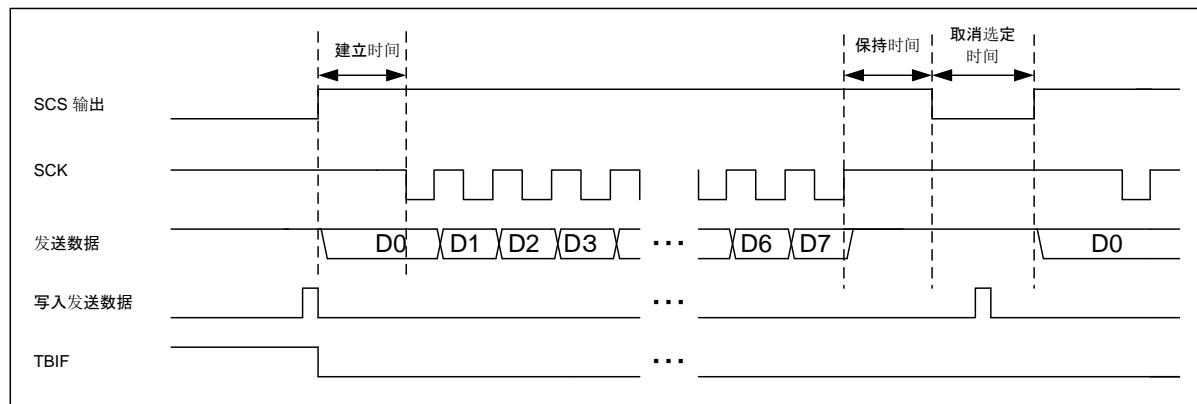


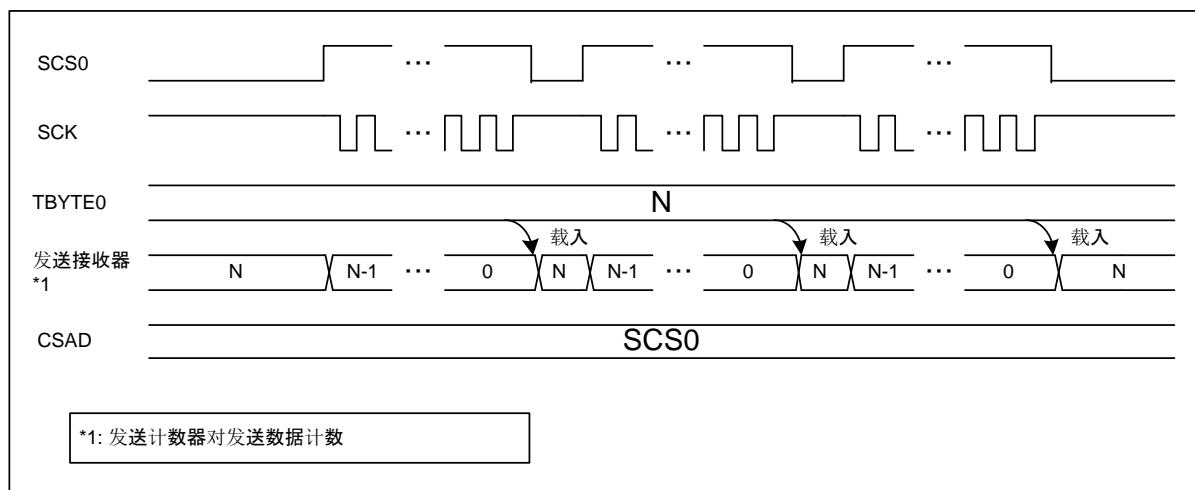
图 33.5-3 时序调整(SPI 传输(SPI MODE=1), CINV=0)



注意：

- 当无保持时间(CSTMG1:CSHDLY7:0=0x00) 的常规传输模式(CR:SPIMODE=0)下，片选引脚可能早于最后一位数据发送结束前释放，在此情况下需要增加保持时间(CSTMG1:CSHDLY7:0)来调节时序。
- 当无建立时间(CSTMG1:CSSDLY7:0=0x00) 的常规传输模式(CR:SPIMODE=0)下，片选引脚可能在第一位数据发送前释放，在此情况下需要增加建立时间(CSTMG1:CSSDLY7:0)来调节时序。
- 独立片选引脚操作(仅主机模式(CR:MSS=0)下支持)。
- 当串行片选有效不保持(CSCR:CSAH=0)时，每次完成了 TBYTE 配置次数的数据收/发，串行片选引脚都会变得无效。
- 关于串行片选有效保持(CSCR:CSAH=1)下的操作参考“串行片选有效保持操作”。

图 33.5-4 独立片选引脚操作(CSEN0=1, CSAH=0)



注意：

- 独立片选操作下的片选引脚时间调节(建立时间，保持时间和取消选定时间)功能依然有

效。

串行片选有效保持操作(CSCR:CSAH=1, 仅主机模式(CR:MSS=0)下支持)

启动发送时, 若将串行芯片选择活动状态保持位(CSCR:CSAH) 设置为 “1”, 则片选引脚处于“保持有效” 状态。

表 33.5-1 串行片选有效保持位 (CSCR:CSAH)

当前状态	当前 CSCR: CSAH 位	当前 SR: TDEF 位	下一状态
发送 (发送计数值< TBYTE)	0	-	片选引脚处于保持状态直到满足 TBYTE 所配置的数据发送次数完成。
	1		
发送 (TBYTE 设置 的数据发送次 数完成)	0	0	在保持时间过后, 片选引脚将释放; 在取消选定时间过后, 启动下一次数据发 送。
		1	在保持时间过后, 片选引脚将释放; 在取消选定时间过后, 等到向 TXDR 寄 存器写入时间后才启动下一次数据发送。
	1	1	片选引脚保持有效状态。
		0	片选引脚处于有效状态, 发送操作继续。 片选引脚保持有效状态直到再次向 TBYTE 写入发送数据个数。
片选错误发生 (EACR:CSER=1)	-	-	不管 CSAH 的设置, 在在保持时间过后, 片选引脚将被释放。
软件复位 (CR:PCL=1)	-	-	不管 CSAH 的设置, 片选引脚立即释放。
发送停止 (CR:TXE=0)			

注意:

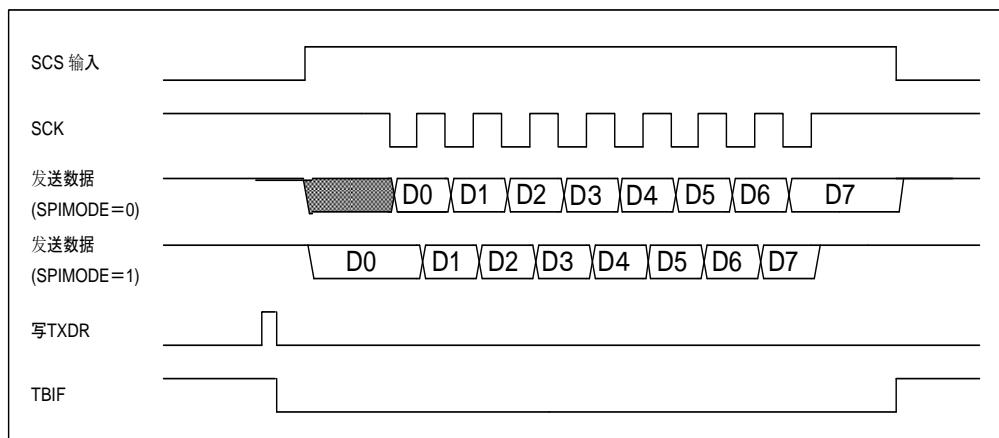
满足下列所有条件下, 片选引脚不会保持且片选保持时间后片选引脚无效, 而且发生片选错
误 (EACR:CSER=1)。

- 发送字节错误使能(EACR:TBEE=1)。
- TBYTE 设置的数据个数收/发未完成。
- 发送数据寄存器(TXDR)为空 (SR:TDEF=1)。

从机模式 (CR:MSS=1)

当片选引脚 0(SCS0)使能(CSCR:CSEN0="1") 且片选引脚输入引脚有效时, 数据收/发操作将与串行时钟(SCK)同步执行。之后, 当输入片选引脚变成无效时, 数据收/发操作结束。

图 33.5-5 从机模式下的片选操作(从机发送, CINV=0)



注意：

- 如果片选引脚输入处于无效状态, 即使有串行时钟 SCK 输入也不会启动数据传输。
- 在接收过程中, 如果片选输入引脚在最后一位数据采样前变成无效状态, 当前接收数据将被舍弃。
- 在发送过程中, 如果片选输入引脚变成无效状态, 数据发送将被舍弃, 同时片选错误标志产生(EACR:CSER=1)。
- 当发送数据寄存器为空(SR:TDEF=1)且片选输入引脚变成无效状态, 发送总线将处于待机状态(SR:TBIF=1)。
- 从机模式(CR:MSS=1)下, 当 CSCR:CSEN0 清“0”时, 不管片选引脚的状态如何, 数据收/发操作将被执行。

33.6 专用波特率发生器

专用波特率发生器只在主机模式下生效。

串行外设接口 SPI 波特率选择

主机模式与从机模式下的专用波特率发生器的设置方法不同。

- 主机操作模式

使用专用波特率发生器将内部时钟频率做出分频，并将该分频选择输出。

- 本发生器提供两个重载计数器，以分别产生接收和发送操作的串行时钟。
- 通过配置波特率生成器寄存器(BRS1 和 BRS0)来设置 15 位重载值。
- 内部时钟频率通过设置的重载值分频得到串行时钟频率。

- 从机操作模式

专用波特率在从机模式(CR:MSS=1)下无效(外部时钟通过时钟输入引脚 SCK 直接输入)。

33.6.1 波特率设置

以下介绍波特率设置方法并列出了串行时钟频率的计算结果。

波特率计算方法

通过波特率生成寄存器(BRS1 和 BRS0)来设置两个 15 位重载计数器。

波特率计算公式如下：

1) 重载值

$$V = \phi / b - 1$$

V：重载值； b：波特率； φ：总线频率

2) 计算示例

总线时钟 16 MHz 的内部时钟，设置波特率为 19200 bps 重载值设置如下：

重载值：

$$V = (16 \times 1000000) / 19200 - 1 = 832$$

因此，实际波特率为：

$$b = (16 \times 1000000) / (832 + 1) = 19208 \text{ bps}$$

3) 波特率误差

按以下公式计算波特率误差：

误差(%) = (计数值 - 目标值) / 目标值 × 100
 例如: 设置 20 MHz 总线时钟和 153600 bps 的波特率:
 重载值 = $(20 \times 1000000) / (129 + 1)$
 波特率(计算值) = $(20 \times 1000000) / (129 + 1) = 153846$ (bps)
 误差(%) = $(153846 - 153600) / 153600 \times 100 = 0.16\%$:
 $B = (16 \times 1000000) / (832 + 1) = 19208$ bps

注意:

- 如果重载值设为"0", 则重载计数器停止。
- 重载值为偶数时, 串行时钟的"L"宽和"H"宽取决于 MR:CINV 和 CR:SPIMODE 的设置。
奇数时, 串行时钟的"H"宽和"L"宽相同。
- 无片选传输模式(CR:SPIMODE=0) 且串行时钟检测电平为"H" (MR:CINV=0)高电平时, 或者片选传输模式(CR:SPIMODE=1)且串行时钟检测电平为"L" (MR:CINV=1)低电平时, 串行时钟的"H"高电平宽度将比"L"低电平宽度长 1 个总线时钟周期。
- 无片选传输模式(CR:SPIMODE=0) 且串行时钟检测电平为"L" (MR:CINV=1)低电平时, 或者片选传输模式(CR:SPIMODE=1)且串行时钟检测电平为"H" (MR:CINV=0)高电平时, 串行时钟的"L"低电平宽度将比"H"高电平宽度长 1 个总线时钟周期。
- 重载值设置需大于等于 3。

各总线时钟频率对应的重载值和波特率

表 33.6-1 重载值与波特率

波特率 (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	重载值	误差	重载值	误差	重载值	误差	重载值	误差	重载值	误差	重载值	误差
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	7	0	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	-0.22	103	0.16	-	-
153600	51	0.16	64	0.16	103	0.16	129	0.16	155	0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	-0.22	138	-0.08	173	-0.22	207	0.16	277	-0.08
76800	103	0.16	129	0.16	207	0.16	259	0.16	311	-0.16	416	-0.08
57600	138	-0.08	173	-0.22	277	-0.08	346	0.06	416	-0.08	555	-0.08
38400	207	0.16	259	0.16	416	-0.08	520	-0.03	624	0	832	0.04
28800	277	-0.08	346	0.06	554	-0.01	693	0.06	832	0.03	1110	0.01

19200	416	-0.08	520	-0.03	832	-0.03	1041	-0.03	1249	0	1666	-0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	-0.03	1666	-0.02	208	0.01	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	0.01
4800	1666	-0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8332	<0.01	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

重载值 : BRS1/0 寄存器设置值(十进制)

误差 : 波特率误差 (%)

如果不是上述值, 参考前面的公式设置。关于系统频率, 参考数据手册。

重载计数器的功能

重载计数器包含发送重载计数器和接收重载计数器, 用于专用波特率生成器。由重载值相应的 15 位寄存器构成, 由外部时钟或内部时钟生成收/发时钟。

计数开始

若向波特率生成器寄存器(BRS1, BRS0)写重载值, 则重载计数器开始计数。

重启

重载计数器在以下条件时重启:

- 收/发重载计数器共用
- 可编程复位(CR:PCL 位)

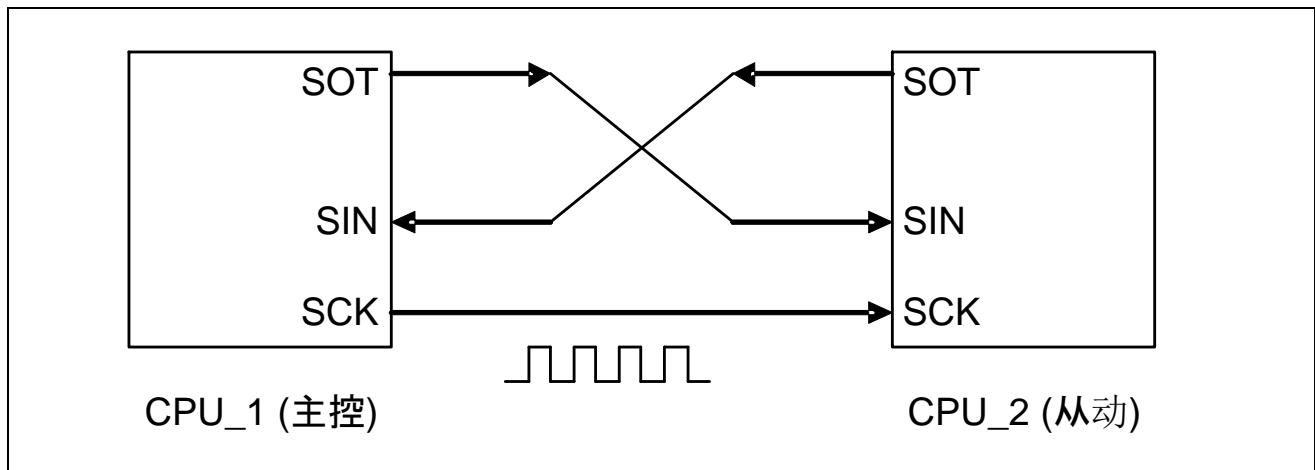
33.6.2 串行外设接口 SPI 设置顺序和程序流程

串行外设接口(SPI)支持双向同步传输。

CPU 互联

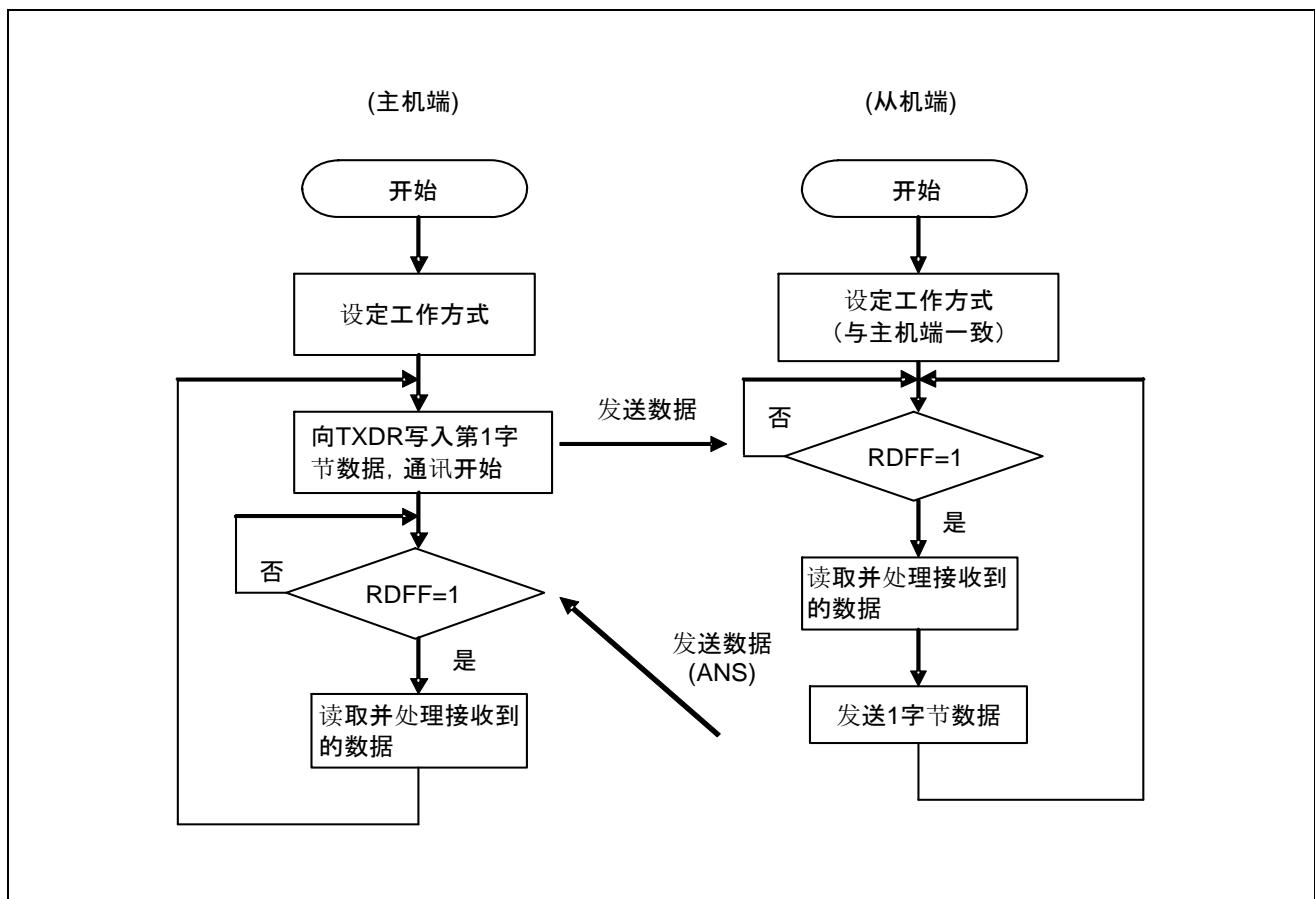
可选择双向通讯的 SPI 接口连接如图 33.6-1 所示。

图 33.6-1 双向通讯的 SPI 接口连接示意图



设置流程

图 33.6-2 双向通信流程图示例



33.7 SPI（串行外设接口）的寄存器

本节描述 SPI(串行外设接口)的寄存器

SPI(串行外设接口)的寄存器一览

表 33.7-1 SPI(串行外设接口)的寄存器一览

	位 15	位 8	位 7	位 0
SPI	CR (串行控制寄存器)		MR (串行模式寄存器)	
	SR (串行状态寄存器)		ECR (扩展通信控制寄存器)	
	RXDR1/TXDR1 (收/发数据寄存器 1)		RXDR0/TXDR0 (收/发数据寄存器 0)	
	EACR (串行辅助控制/状态寄存器)			
	TMR (串行定时器寄存器)			
	TMC (串行定时器比较寄存器)			
	CSCR (串行片选控制/状态寄存器)			
	CSTMG1 (串行片选时间寄存器 1)		CSTMG0 (串行片选时间寄存器 0)	
	CSTMG3 (串行片选时间寄存器 3)		CSTMG2 (串行片选时间寄存器 2)	
	-		TBYTE0(发送字节数寄存器 0)	
	BRS1 (波特率发生器寄存器 1)		BRS0 (波特率发生器寄存器 0)	

表 33.7-2 SPI(串行外设接口)位配置

	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
CR/ MR	PCL	MSS	SPI MO DE	RXI E	TXI E	TBI E	RXE	TXE	MO DE2	MO DE1	MO DE0	-	CIN V	TDS	CKE	DOE
SR/ ECR	REC LR	-	-	AW C	ORF	RDF F	TDE F	TBI F	SOP S	DL3	CSF E	DTR WS1	DTR WS0	DL2	DL1	DL0
TXDR 1/0 (RXD R1/0)	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
TXDR 3/2 (RXD R3/2)	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
EACR	-	-	TBE E	CSE IE	CSE R	-	-	TIF	TIE	SYN TE	-	CLK DIV 3	CLK DIV 2	CLK DIV 1	CLK DIV 0	TM RE
TMR	TM1 5	TM1 4	TM1 3	TM1 2	TM1 1	TM1 0	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
TMC	TC1 5	TC1 4	TC1 3	TC1 2	TC1 1	TC1 0	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
CSCR	-	-	-	-	-	-	CSA H	CST ODI V2	CST ODI V1	CST ODI V0	CSL VS	-	-	-	CSE N0	CSO E
CSTM G 1/0	CSS DLY 7	CSS DLY 6	CSS DLY 5	CSS DLY 4	CSS DLY 3	CSS DLY 2	CSS DLY 1	CSS DLY 0	CSH DLY 7	CSH DLY 6	CSH DLY 5	CSH DLY 4	CSH DLY 3	CSH DLY 2	CSH DLY 1	CSH DLY 0
CSTM G 3/2	CSD S 15	CSD S 14	CSD S 13	CSD S 12	CSD S 11	CSD S 10	CSD S 9	CSD S 8	CSD S 7	CSD S 6	CSD S 5	CSD S 4	CSD S 3	CSD S 2	CSD S 1	CSD S 0
TBYT E0	-	-	-	-	-	-	-	-	CS0 TD7	CS0 TD6	CS0 TD5	CS0 TD4	CS0 TD3	CS0 TD2	CS0 TD1	CS0 TD0
BRS1/ BRS0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

33.7.1 串行控制寄存器 (CR)

串行控制寄存器(CR)可使能/禁止收/发中断、发送空闲中断及收/发操作。另外，可连接 SPI 并复位 SPI。

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	PCL	MSS	SPIMO DE	RXI E	TXIE	TBIF E	RXE	TXE	(MR)		
属性	R/W	R/W	R/W	R/ W	R/W	R/W	R/W	R/W			
初始值	0	0	0	0	0	0	0	0			

寄存器功能

[位 15] PCL: 编程清零位，初始化 SPI 的内部状态

位	说明	
	写入	读取
0	无效	
1	编程清“0”	始终读“0”

设为"1"时：

直接将 SPI 复位(软件复位)并保持当前寄存器的设置。这时，收/发状态的操作立刻被切断。
波特率生成器重载 BRS1/0 寄存器的设置值后重新启动。

所有收/发中断源(SR:TDEF, TBIF, RDFF, ORF, TIF, CSER)被初始化。

所有的片选引脚处于无效状态。

设为"0"时：无效，始终读"0"。

注意：

- 首先设为中断禁止后，然后执行编程清零

[位 14] MSS: 主机模式/从机模式功能选择位，选择主机模式或从机模式

位	说明
0	主机模式
1	从机模式

注意：

- 选择从机模式时若 MR:CKE=0，则直接输入外部时钟
- 设置 MSS 位后，使能数据接收(RXE=1)

[位 13] SPIMODE: SPI 模式对应位，该位启动支持 SPI 通讯

位	说明
0	不使能 SPI 模式功能
1	使能 SPI 模式功能

注意：

- 此位的设置需在数据收/发停止的情况下(TXE=RXE=0)
- 此位在下列任一情况下都有效：
 - 片选引脚禁止(CSCR:CSEN0="0")
 - 从机操作模式(CR:MSS=1)

[位 12] RXIE: 接收中断使能位

该位可使能/禁止至 CPU 的接收中断请求输出。

若 RXIE 位和接收数据标志位(SR:RDFF)置"1", 或任何一个错误标志位(ORF)置"1", 则输出接收中断请求。

位	说明
0	禁止接收中断
1	使能接收中断

[位 11] TXIE: 发送中断使能位

该位可使能/禁止至 CPU 的发送中断请求输出。

若 TXIE 位和 SR:TDEF 位均置"1", 则会输出发送中断请求。

位	说明
0	禁止发送中断
1	使能发送中断.

[位 10] TBIFE: 发送总线空闲中断使能位

该位可使能/禁止至 CPU 的发送总线空闲中断请求输出。

如果 TBIFE 位和 SR:TBIF 位置"1", 则会输出发送总线空闲中断请求。

位	说明
0	禁止发送总线空闲中断
1	使能发送总线空闲中断

[位 9] RXE: 数据接收使能位, 该位可使能/禁止 SPI 的接收操作

位	说明
0	禁止接收
1	使能接收

接收时若禁止接收操作($RXE=0$)，则当前的数据接收会立即停止。

若设置 MSS 位和 MR:CINV 位后，可使能接收($RXE=1$)。

[位 8] TXE: 数据发送使能位

该位可使能/禁止 SPI 的发送操作。

位	说明
0	禁止发送
1	使能发送

注意：

- 发送时若禁止发送操作($TXE=0$)，则发送操作会立即停止。
- 主机模式($CR:MSS=0$)下使用串行片选($CSCR:CSEN=1$)，须执行软件复位 ($CR:PCL=1$)。

33.7.2 串行模式寄存器(MR)

串行模式寄存器(MR)可以选择工作模式，设置传输方向、数据长度、串行时钟反转，并使能/禁止至引脚的串行数据和时钟。

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	(CR)			MOD E2	MOD E1	MOD E0	-	CINV	TDS	CKE	DOE
属性				R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
初始值				0	0	0	-	0	0	0	0

寄存器功能

[位 7:5] MODE2, MODE1, MODE0: 工作模式设置位，这些位可设置工作模式

"010": 设置为工作模式 2 (SPI 模式)

"100": 设置为工作模式 3 (I²C 模式)

本章介绍工作模式 2(串行外设接口 SPI 模式)的寄存器和操作。

位 7	位 6	位 5	说明
0	1	0	工作模式 2 (SPI 模式)
1	0	0	工作模式 3 (I ² C 模式)
上述值以外		禁止设置	

注意：

- 禁止进行上述之外的设置。
- 要切换工作模式，需在执行可编程清零(CR:PCL=1)后面接着切换工作模式。
- 设置工作模式后，设置各个寄存器。

[位 4]保留：保留位，读取值为"0"。如果写入，需写"0"。

[位 3] CINV:串行时钟反转位

此位可以反转串行时钟格式

此位在主机模式(CR:MSS=0)下片选功能有效时生效。

清"0"时：

串行时钟输出的信号检测电平设置为高电平"H"。

同步传输时，在串行时钟的下降沿发送数据；而在 SPI 传输时，与串行时钟的上升沿同步进行输出。

同步传输时，在串行时钟的上升沿接收数据；而在 SPI 传输时，在串行时钟下降沿处进行采样。
置"1"时：

串行时钟输出的信号检测电平设置为低电平" L "

同步传输时，在串行时钟的上升沿发送数据；而在 SPI 传输时，与串行时钟下降沿同步进行输出。

同步传输时，在串行时钟的下降沿接收数据；而在 SPI 传输时，在串行时钟的上升沿处进行采样。

位	说明
0	信号检测高电平" H "格式
1	信号检测低电平" L "格式

注意：

- 禁止收/发(TXE=RXE=0)时，总是设置该位。
- 设置 CINV 位后，启动数据接收(CR:RXE=1)。
- 此位在下列情况下仍然配置有效：
 - 片选引脚功能禁止(CSCR:CSEN="0")
 - 从机模式(CR:MSS=1)

[位 2] TDS: 传输方向选择位，该位可以选择从传输串行数据的最低位(LSB 优先， TDS=0)进行传输还是从最高位(MSB 优先， TDS=1)进行传输。

位	说明
0	LSB 优先(优先传输最低位)
1	MSB 优先(优先传输最高位)

注意：

- 禁止收/发(CR:TXE=RXE=0)时，总是设置该位。
- 此位在下列情况下仍然配置有效：
 - 片选引脚功能禁止(CSCR:CSEN="0")
 - 从机模式(CR:MSS=1)

[位 1] CKE: 主机模式时的串行时钟输出使能位，该位可以控制串行时钟的 I/O 端口

位	说明
0	禁止串行时钟输出
1	使能串行时钟输出

注意：

- 用作 SCK 引脚时，须设置 GPIO 功能。

[位 0] DOE: 串行数据输出使能位，该位可以使能/禁止串行数据的输出

位	说明
0	禁止串行数据输出
1	使能串行数据输出

0	禁止串行数据输出
1	使能串行数据输出

注意：

- 用作 SOT 引脚时，须设置 GPIO 功能。

33.7.3 串行状态寄存器(SR)

串行状态寄存器(SR)用于确认收/发状态，确认接收错误标志及清零接收错误标志，及配置传输数据宽度。

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	RECLR	-	-	Reserved	ORF	RDF	TDEF	TBIF	(ECR)		
属性	R/W	-	-	-	R	R	R	R			
初始值	0	-	-	-	0	0	1	1			

寄存器功能

[位 15] RECLR:接收错误标志清零位，该位可以清零串行状态寄存器(SR)的 ORF 标志。

写"1"清零错误标志

写"0"无效

读取值总为 "0"。

位	说明	
	写	读
0	无影响	始终读 "0"
1	清除接收错误标志(ORF).	

[位 14:13] - : 保留位，读：值不定。写：无影响。

[位 12] AWC: 数据宽度控制位，选择接收数据寄存器(RXDR)和发送数据寄存器(TXDR)的访问宽度是 16 位或 32 位。

位	说明
0	16 位访问
1	32 位访问

注意：

- 在 TXDR 和 RXDR 无数据时(SR:TDEF=1, SR:RDFF=0)且接收和发送禁止(CR:TXE=RXE=0)时修改该位。

[位 11] ORF: 溢出错误标志位

数据接收时若发生溢出，该位置"1"。串行状态寄存器(SR)的 RECLR 置"1"则可清零该位若 ORF 位和 CR:RXIE 位均置"1"，则可输出接收中断请求。

若该标志置位，接收数据寄存器(RXDR)的数据无效。

位	说明
0	无溢出错误
1	有溢出错误

[位 10] RDFF: 接收数据满标志位

该标志指示接收数据寄存器(RXDR)的状态。

若 RXDR 载入接收数据，则该位置"1"。若读出接收数据寄存器(RXDR)，该位清"0"。

若 RDFF 位和 CR:RXIE 位均置"1"，则会输出接收中断请求。

位	说明
0	接收数据寄存器 RXDR 为空
1	接收数据寄存器 RXDR 有数据

[位 9] TDEF: 发送数据空标志位

该标志指示发送数据寄存器(TXDR)的状态。

若向 TXDR 写发送数据，该位清"0"，指示 TXDR 中存在有效数据。若数据载入到发送移位寄存器，则发送开始。该位置"1"，指示 TXDR 不存在有效数据。

TDEF 位和 CR:TXIE 位均置"1"时，则输出发送中断请求。

若串行控制寄存器(CR)的 PCL 位置"1"，则 TDEF 位也置"1"。

位	说明
0	发送数据寄存器 TXDR 有数据
1	发送数据寄存器 TXDR 为空

[位 8] TBIF: 发送总线空闲标志位

该位显示 SPI 未进行发送操作

向发送数据寄存器(TXDR)写数据，该位清"0"

因发送数据寄存器(TXDR)为空(TDEF=1)而未执行发送操作，该位置"1"。

若串行控制寄存器(CR)的 PCL 位置"1"，TDEF 位置"1"。

该位置"1"时若使能发送总线空闲中断(CR:TBIFE=1)，则可输出发送中断请求。

位	说明
0	发送中
1	无发送操作

33.7.4 扩展通信控制寄存器(ECR)

扩展通信控制寄存器(ECR)可以设置收/发数据长，及把串行数据输出固定到" H "。

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	-		SOP S	DL3	CSFE	DTR WS1	DTR WS0	DL2	DL1	DL0	
属性			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值			0	0	0	0	0	0	0	0	0

寄存器功能

[位 7] SOPS:串行输出引脚设置位

该位可把串行输出引脚固定到" H "。

该位写"1"时，SOT 引脚置" H "。此后，无需把该位清"0"。

读时，该位始终读"0"。

位	说明	
	写	读
0	无影响	
1	SOT 引脚配置为高" H " .	始终读"0"

注意：

- 发送串行数据时，无需设置该位。

[位 5] CSFE:串行片选样式使能位

使能/禁止串行片选样式。

当该位为“1”时，每一个片选引脚样式可以设置如下行为：

- 片选引脚有效电平
- 串行时钟的检测电平
- 选择是 SPI 模式或者普通模式
- 串行数据方向
- 串行数据长度

[位 4:3] DTRWS1, DTRWS0: 数据收/发等待时间选择位

主机模式时，这些位为连续的数据发送或接收设置等待计数。从机模式时，这些位设置到"00"：

设置"00"时：连续输出 SCK。

设置"01"时：1 位时间等待后，输出 SCK。

设置"10"时：2 位时间等待后，输出 SCK。

设置"11"时 :3 位时间等待后，输出 SCK。

位 4	位 3	说明
0	0	0 位
0	1	1 位
1	0	2 位
1	1	3 位

[位 6, 位 2:0] DL3, DL2, DL1, DL0:数据长选择位，这些位指定收/发数据的数据长。

DL3	DL2	DL1	DL0	说明
0	0	0	0	8 位长
0	0	0	1	5 位长
0	0	1	0	6 位长
0	0	1	1	7 位长
0	1	0	0	9 位长
0	1	0	1	10 位长
0	1	1	0	11 位长
0	1	1	1	12 位长
1	0	0	0	13 位长
1	0	0	1	14 位长
1	0	1	0	15 位长
1	0	1	1	16 位长
1	1	0	0	20 位长
1	1	0	1	24 位长
1	1	1	0	32 位长

注意：

- 禁止进行上述之外的设置。
- 下列情形之一需配置：
 - 片选引脚禁止 (CSCR:CSEN3-0="0000")。
 - 从机模式(CR:MSS=1)

33.7.5 接收数据寄存器/发送数据寄存器(RXDR/TXDR)

接收数据和发送数据寄存器配置到同一地址上。读时用作接收数据寄存器；写时用作发送数据寄存器。

接收数据寄存器(RXDR)

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	D1 5	D1 4	D1 3	D1 2	D1 1	D1 0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
属性	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

接收数据寄存器(RXDR)是接收串行数据的 16 位数据缓冲寄存器。

若串行数据信号发送到串行输入引脚(SIN 引脚)，该信号可以使用移位寄存器进行转换，并存储到接收数据寄存器(RXDR)。

根据数据长度设置，接收数据位按照从低到高的顺序 存储在 RXDR 中，其他高位为“0”。例如：“45”h 是接收到的 8 位数据，D7 到 D0 =”45”h, D15 到 D8 =0。

若接收数据存储到接收数据寄存器(RXDR)，接收数据满标志位(SR:RDFF)置“1”。使能接收中断时(SR:RXIE=1)，产生接收中断请求。

接收数据满标志位(SR:RDFF)置“1”时，须读出接收数据寄存器(RXDR)。读接收数据寄存器(RXDR)时，接收数据满标志位(SR:RDFF)自动清“0”。

产生接收错误(SR:ORF)时，接收数据寄存器(RXDR)的数据无效。

发送数据寄存器(TXDR)

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	D1 5	D1 4	D1 3	D1 2	D1 1	D1 0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
属性	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
初始值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

寄存器功能

发送数据寄存器(TXDR)是用于发送串行数据的 16 位数据缓冲寄存器。

若使能发送操作(CR:TXE=1)，并把发送数据写入发送数据寄存器(TXDR)，则发送数据传输到发送移位寄存器。这些数据转换为串行数据，并从串行数据输出引脚(SOT 引脚)输出。

根据数据长度设置,发送数据位按照从低到高的顺序存储在 TXDR 中, 其他高位为“0”。例如：“45”h 是发送寄存器 TXDR 接收到的 8 位数据, D7 到 D0 =”45”h, D15 到 D8 =0。

发送数据写入发送数据寄存器(TXDR)时, 发送数据空标志(SR:TDEF)清"0"。

若发送数据传输到发送移位寄存器, 发送传输开始。发送完成时, 发送数据空标志(SR:TDEF)置"1"。

若发送数据空标志(SR:TDEF)置"1", 可以写入下一次发送数据。使能发送中断的话, 发送中断发生。仅当发送中断发生后或发送数据空标志(SR:TDEF)置"1", 方可写入下一次发送数据。发送数据空标志(SR:TDEF)清"0"时, 不能向发送数据寄存器(TXDR)写入发送数据。

注意:

- 发送数据寄存器是只写型寄存器, 而接收数据寄存器是只读型寄存器。因为两个寄存器配置到同一地址, 所以读值与写值不同。所以, 不可以使用 INC/DEC 指令和读-修改-写(RMW)指令。

33.7.6 串行辅助控制寄存器(EACR)

串行辅助控制寄存器(EACR)用于控制串行测试，配置串行定时器启动方法，禁止/使能定时器中断，禁止/使能同步传输，设置串行定时器的分频比和启停串行定时器。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	Reserved	TBEE	CSEIE	CSER	-	-	TIF	
属性	-	R/W	R/W	R/W	-	-	R/W	
初始值	00	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	TXIE	SYNTE	-	CLKDI V3	CLKDI V2	CLKDI V1	CLKDI V0	TMRE
属性	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[位 15:14] 保留位，读“0”，写“0”。

[位 13] TBEE: 传输字节错误使能位，主机模式(CR:MSS=0)下，使能/禁止片选错误发生。

关于细节参考“33.2.4 片选错误产生和标志位设置时序”。

位	说明
0	主机模式(CR:MSS=0)下禁止片选错误产生.
1	主机模式(CR:MSS=0)下允许片选错误产生.

注意：

- 在禁止数据收/发(CR:TXE=RXE=“0”)时，才能修改该位。

[位 12] CSEIE: 片选错误中断使能位

该位用于禁止/使能片选错误中断请求输出到 CPU。

当 CSEIE 位和片选错误标志位 (CSER) 都是“1”，输出片选中断请求到 CPU。

位	说明
0	禁止片选错误中断输出
1	使能片选错误中断输出

[位 11] CSER: 片选错误标志位

该位用于标志是否发生片选错误。

关于细节参考“33.2.4 片选错误产生和标志位设置时序”。

当 CSEIE 位和片选错误标志位 (CSER) 都是“1”，输出片选中断请求到 CPU。

向该位写入“0”时清“0”该标志位，写“1”无效。

位	说明
0	无片选错误
1	有片选错误

注意：

- 该位可以通过软件复位 (CR:PCL=“1”)清 “0”。
- 支持 RMW 操作读取“1”。
- 当从机模式(CR:MSS=1)下，不使用片选功能 (CSCR:CSEN0=0)，该位不会置 “1”。
- 当片选错误发生后，需要停止数据发送和向该位写 “0” 以清除该标志位。如果需要重启数据传输，需向该位写 “0”，使能数据发送 (CR:TXE=1)和写发送数据寄存器(TXDR)。
- 如果从机模式片选引脚输入时因为总线时钟噪声或者其他因素导致该位可能被置 “1”。在此情况下，重启发送操作。

[位 8] TIF:定时器中断标志位

当串行定时器寄存器(TMR)和串行定时器比较寄存器(TMC)的值相等时，串行定时器寄存器(TMR)被清 “0”，定时器中断标志位(TIF)被置位为 “1”。

当定时器中断标志位(TIF) 和定时器中断使能位(TIE) 都是 “1”，输出状态中断标志请求到 CPU。

向该位写入 “0” 时清 “0” 该标志位，写 “1” 无效。

位	说明
0	无定时器中断请求
1	有定时器中断请求

注意：

- 该位可以通过软件复位 (CR:PCL=“1”)清 “0”。
- 支持 RMW 操作读取“1”。
- 当同步发送使能位(SYNTE)为“1”时，该位将不会被置位。

[位 7] TXIE: 定时器中断使能位

该位禁止/使能向 CPU 发送定时器中断请求。

当定时器中断标志位(TIF)和定时器中断使能位(TXIE) 都是 “1”，输出状态中断请求到 CPU。

位 7	说明
0	禁止定时器中断输出
1	使能定时器中断输出

[位 6] SYNT: 同步发送使能位

该位禁止/使能同步发送。

当该位为“1”和串行定时器寄存器(TMR)的值与串行定时器比较寄存器(TMC) 的值相同时，启动同步发送操作。

该位为“0”时，停止同步发送操作，串行定时器可以作为计数器使用。

位	说明
0	使能同步发送。串行定时器可被用作普通计数器。
1	使能同步发送。串行定时器不能用作普通计数器。

注意：

- 只能在串行定时器使能位(TMRE=0)的时候允许改变该位。
- 如果在同步发送使能(SYNTE=1)下发送禁止(CR:TXE=0)，即使串行定时器寄存器(TMR)的值与串行定时器比较寄存器(TMC) 的值相同，数据发送也不会启动。
- 从机模式(CR:MSS=“1”)该位内部固定为“0”。

[位 4:1] CLKDIV3:0: 定时器操作时钟分频控制位

这 4 位设置串行时钟的分频比

位 4	位 3	位 2	位 1	定时器工作时钟						
				分频比	总线时钟= 8MHz	总线时钟= 10MHz	总线时钟= 16MHz	总线时钟= 20MHz	总线时钟= 24MHz	总线时钟= 32MHz
0	0	0	0	1	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	0	1	2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	0	1	0	4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	0	1	1	8	1μs	800ns	500ns	400ns	333.33ns	250ns
0	1	0	0	16	2μs	1.6μs	1μs	800ns	666.67ns	500ns
0	1	0	1	32	4μs	3.2μs	2μs	1.6μs	1.33μs	1μs
0	1	1	0	64	8μs	6.4μs	4μs	3.2μs	2.67μs	2μs
0	1	1	1	128	16μs	12.8μs	8μs	6.4μs	5.33μs	4μs
1	0	0	0	256	32μs	25.6μs	16μs	12.8μs	10.67μs	8μs

注意：

- 在串行定时器使能位(TMRE=0 时)，可设置/改变这些位。
- 禁止设置除上述以外的其他组合。

[位 0]TMRE: 串行定时器使能位

此位使能/禁止串行定时器。

位	说明
0	停止串行定时器。在停止的时候，串行定时器寄存器(TMR)保持之前的值
1	该位从“0”变为“1”时，初始化串行定时器寄存器的值到“0”并启动串行定时器计数

注意：

- 当利用串行定时器执行同步发送操作，在发送使能之前配置该位为“1”。

33.7.7 串行定时器寄存器(TMR)

串行定时器寄存器(TMR)指当前串行定时器的计数值。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10	TMR9	TMR8
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[位 15:0] TMR[15:0]: 定时器计数位，这些位表明串行定时器的当前计数值

在定时器工作过程中，串行定时器寄存器的值在每一个(EACR:CLKDIV3:0)设置的时钟周期内增加 1. (EACR:CLKDIV3:0)。

注意：

- 启动定时器时，该位被初始化为“0”并开始计数。

33.7.8 串行定时器比较寄存器(TMC)

串行定时器比较寄存器(TMC)用于设置串行定时器的比较值。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	TMC15	TMC14	TMC13	TMC12	TMC11	TMC10	TMC9	TMC8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	TMC7	TMC6	TMC5	TMC4	TMC3	TMC2	TMC1	TMC0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

寄存器功能

[位 15:0] TMC15:0: 比较寄存器设置位，设置串行定时器的比较值。

此位的值与串行定时器寄存器 (TMR) 的值比较相等后串行定时器寄存器 (TMR) 清“0”并继续计数。

如果同步传输功能禁止(EACR:SYNTE=“0”),则定时器中断标志位(EACR:TIF)设置为“1”，如果同步传输功能开启(EACR:SYNTE=“1”),数据发送操作将启动

执行以下操作的间隔按照下面公式计算：(TMC:TMC+1) × 定时器操作时钟(由 EACR:CLKDIV3:0 配置)

EACR:TIF 标志位置“1”。

与串行时钟同步的数据发送操作启动。

注意：

- 同时满足下列条件时，定时器中断标志(EACR:TIF) 固定为“1”。
- 同步发送禁止(EACR:SYNTE=“0”)。
- 寄存器值配置为“0x0000”。
- 定时器工作中。
- 定时器工作时钟分频设置(EACR:CLKDIV)为 “0b0000”。
- 只有在串行定时器停止 (EACR:TMRE=“0”)时，才可以改变本寄存器值。

33.7.9 串行片选控制/状态寄存器(CSCR)

串行片选控制/状态寄存器(CSCR)用于选择片选引脚的启动/结束，指示有效片选引脚的状态和保持片选引脚状态，片选引脚极性，和使能/禁止片选输出。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	CSAS1	CSAS0	CSAE1	CSAE0	CSAD1	CSAD0	CSAH	CSTODI V2
属性	R/W	R/W	R/W	R/W	R	R	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CSTODI V1	CSTODI V0	CSLVS	CSEN3	CSEN2	CSEN1	CSEN0	CSOE
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	0	0	0	0	0

寄存器功能

[位 15:14]CSAS1-0: 串行片选有效开始位，选择串行片选引脚的开始引脚

在向 TXDR 写入发送数据，启动数据发送操作(CR:TXE= "0" 到 CR:TXE= "1")后，此位设置对应的串行片选有效。

位 15:14	说明
00	片选开始引脚是 SCS0
01	片选开始引脚是 SCS1
10	片选开始引脚是 SCS2
11	片选开始引脚是 SCS3

注意：

- 在数据收/发禁止(CR:TXE=RXE= "0")时配置该位。
- 在片选有效启动位(CSCR:CSAS1-0) 与片选有效结束位(CSCR:CSAE1-0)相同时，片选操作仅在所设置的引脚上执行。
- 从机模式(CR:MSS=1)下无效。
- 只有在片选功能使能(CSEN=1)下的片选引脚变为活动状态。
- 仅在主机模式下(CR:MSS= "0")和片选使能(CSEN=1)有效的情况下，该位配置有效。

[位 13:12]CSAE1-0: 串行片选有效结束位

选择串行片选引脚的结束引脚。

在此位设置的片选引脚变为活动状态时，片选引脚有效开始位(CSAS1,CSAS0) 的配置引脚有效。

位 13:12	说明
---------	----

00	片选结束引脚是 SCS0
01	片选结束引脚是 SCS1
10	片选结束引脚是 SCS2
11	片选结束引脚是 SCS3

注意：

- 在数据收/发禁止(CR:TXE=RXE= "0")时配置该位。
- 在片选有效启动位(CSCR:CSAS1-0) 与片选有效结束位(CSCR:CSAE1-0)相同时，片选操作仅在所设置的引脚上执行。
- 从机模式(CR:MSS=1)下无效。
- 只有在片选功能使能(CSEN=1)下的片选引脚变为活动状态。
- 仅在主机模式下(CR:MSS= "0")和片选使能(CSEN=1)有效的情况下，该位配置有效。

[位 11:10]CSAD1-0: 指示有效片选引脚的状态

指示当前有效的片选引脚信息

位 11:10	说明
00	SCS0 有效
01	SCS1 有效
10	SCS2 有效
11	SCS3 有效

注意：

- 当片选引脚无效时，指示即将出现的下一个有效的片选引脚。
- 在从机模式(CR:MSS="1")或者发送禁止(CR:TXE="0")时该位固定为 “00”。
- 软件复位(CR:PCL=1)将该位复位为 “00”。

[位 9] CSAH: 串行片选有效保持位

选择片选引脚的有效状态是否保持。

细节参考“5.串行片选操作”的 “串行片选有效保持操作 (CSCR:CSAH=1,仅主机模式 (CR:MSS=0) 下支持)” 部分。

位	说明
0	不保持片选引脚的有效状态
1	保持片选引脚的有效状态

注意：

- 当发送禁止(CR:TXE="0") 或者软件复位(CR:PCL="1")后，片选引脚变成无效状态而不管 CSAH 的设置。
- 当片选错误发生(EACR:CSER=1)，片选引脚变成无效状态而不管 CSAH 的设置。

[位 8:6] CSTODIV2:0: 串行片选时序操作时钟分频控制位

设置片选时序操作时钟的分频比

位 8	位 7	位 6	串行片选时序操作时钟						
			分频比	总线时钟 =8MHz	总线时钟 =10MHz	总线时钟 =16MHz	总线时钟 =20MHz	总线时钟 =24MHz	总线时钟 =32MHz
0	0	0	1	125ns	100ns	62.5ns	50ns	41.67ns	31.25ns
0	0	1	2	250ns	200ns	125ns	100ns	83.33ns	62.5ns
0	1	0	4	500ns	400ns	250ns	200ns	166.67ns	125ns
0	1	1	8	1μs	800ns	500ns	400ns	333.33ns	250ns
1	0	0	16	2μs	1.6μs	1μs	800ns	666.67ns	500ns
1	0	1	32	4μs	3.2μs	2μs	1.6μs	1.33μs	1μs
1	1	0	64	8μs	6.4μs	4μs	3.2μs	2.67μs	2μs

注意：

- 这些位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 从机模式(CR:MSS="1")设置这些位无效。
- 禁止上述组合以外的配置。

[位 5]CSLVS: 串行片选极性设置位

选择串行片选引脚在无效状态是的引脚状态为“高”或者“低”。

此位仅对配置 SCS0 引脚有效。

位	说明
0	设置片选引脚无效状态时为‘低’
1	设置片选引脚无效状态时为‘高’

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。

[位 4:1]CSEN3-0:片选使能位，此 4 位用于使能/禁止对应的片选引脚

CSEN3 位对应的是 SCS3 引脚，CSEN2 位对应的是 SCS2 引脚，CSEN1 位对应的是 SCS1 引脚，CSEN0 位对应的是 SCS0 引脚。

从机模式(CR:MSS=1)下，仅 CSEN0 位可以使能/禁止片选引脚。

位	说明
0	禁止片选引脚操作
1	使能片选引脚操作

注意：

- 在数据收/发禁止(CR:TXE=RXE= "0")时配置该位。
- 在禁止片选 CSEN3-0=“0000” 的主机模式(CR:MSS=0)下，直接进行数据收/发操作，无需片选引脚。
- 在禁止片选 CSEN0=“0” 的从机模式(CR:MSS=1)下，直接进行数据收/发操作，无需片选引脚。
- 禁止使能不用的片选引脚。

[位 0]CSOE: 片选输出使能位，此位禁止/使能片选引脚输出。

位	说明
0	禁止所有的片选引脚输出
1	使能所有的片选引脚输出

注意：

- 在数据收/发禁止(CR:TXE=RXE= "0")时配置该位。
- 从机模式(CR:MSS=1)时，该位需设置为“0”。

33.7.10 串行片选时间寄存器(CSTMG3-0)

串行片选时间寄存器(CSTMG3-0)用于设置片选引脚的建立时间，保持时间和取消选定时间。

片选时间寄存器(CSTMG1, CSTMG0) 配置

位	15	14	13	12	11	10	9	8
字段名	CSSDL Y7	CSSDL Y6	CSSDL Y5	CSSDL Y4	CSSDL Y3	CSSDL Y2	CSSDL Y1	CSSDL Y0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CSHDL Y7	CSHDL Y6	CSHDL Y5	CSHDL Y4	CSHDL Y3	CSHDL Y2	CSHDL Y1	CSHDL Y0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

片选时序寄存器(CSTMG1, CSTMG0) 功能

[位 15:8]CSSDLY7- CSSDLY0: 串行片选建立时间位

设置从片选引脚有效到串行时钟输出的等待时间。如果该位设置为“0x00”，片选有效和串行时钟同时输出。

位 15:8	建立时间
0x00	片选有效的同时，输出串行时钟.
0x01	1×串行片选操作时钟
0x02	2×串行片选操作时钟
:	:
0xFE	254×串行片选操作时钟
0xFF	255×串行片选操作时钟

注意：

- 只有收/发禁止(CR:TXE=RXE= "0")时才能配置该位。
- 从机模式(CR:MSS="1")下设置无效。

[位 7:0] CSHDLY[7:0]: 串行片选保持时间位

设置从时钟输出结束到片选引脚变成无效的时间。如果该位设置为“0x00”，片选引脚在同步时钟输出结束的同时变成无效。

位 7:0	保持时间

0x00	片选引脚在同步时钟输出结束的同时变成无效
0x01	1×串行片选操作时钟
0x02	2×串行片选操作时钟
:	:
0xFE	254×串行片选操作时钟
0xFF	255×串行片选操作时钟

注意：

- 只有收/发禁止(CR:TXE=RXE= "0")时才能配置该位
- 从机模式(CR:MSS="1")下设置无效。

串行片选时间寄存器(CSTMG3, CSTMG2)配置

位	15	14	13	12	11	10	9	8
字段名	CSDS15	CSDS14	CSDS13	CSDS12	CSDS11	CSDS10	CSDS9	CSDS8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CSDS7	CSDS6	CSDS5	CSDS4	CSDS3	CSDS2	CSDS1	CSDS0
属性	R/W							
初始值	0	0	0	0	0	0	0	0

串行片选时间寄存器(CSTMG3, CSTMG2)功能

[位 15:0] CSDS[15:0]: 片选取消选定时间位

设置从片选引脚从无效状态到再次变为有效状态所需的时间。

位 15:0	最小取消选定时间
0x0000	无最小取消选定时间 (5 个总线时钟周期)
0x0001	1×串行片选操作时钟
0x0002	2×串行片选操作时钟
:	:
0xFFFFE	65534×串行片选操作时钟
0xFFFF	65535×串行片选操作时钟

注意：

- 在收/发禁止(CR:TXE=RXE= "0")时配置该位。
- 从机模式(CR:MSS="1")下设置无效。

- 不管该寄存器的设置值是多少，片选引脚从无效状态到有效状态需要至少 5 个总线时钟(或者更多)的时间。
- 不要同时设置 CSTMG2:CSDS=0x0001 和 CSCR:CSTODIV=0b000。

33.7.11 串行片选格式寄存器(CSFR2-0)

串行片选格式寄存器(CSFR)用于选择片选引脚的活动电平、时钟反向、SPI模式、数据方向以及数据长度。

寄存器配置(CSFR1,CSFR0)

位	15	14	13	12	11	10	9	8
字段名	CS2 CSLVS	CS2 CINV	CS2 SPI	CS2 TDS	CS2 DL3	CS2 DL2	CS2 DL1	CS2 DL0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	CS1 CSLVS	CS1 CINV	CS1 SPI	CS1 TDS	CS1 DL3	CS1 DL2	CS1 DL1	CS1 DL0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	0	0	0	0	0	0	0

寄存器功能

[位 15]CS2CSLVS: 串行片选 SCS2 极性设置位，设置串行片选引脚在无效状态下的电平极性是高“H”或者低“L”，在片选格式有效(ECR:CSFE=1)的情况下。

此位仅对 SCS2 引脚有效。

位	说明
0	SCS2 在无效状态时，电平为“L”。
1	SCS2 在无效状态时，电平为“H”。

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 14]CS2CINV:片选引脚 SCS2 下的串行时钟反转位，反转串行时钟形式。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS2 有效。

位	说明
0	信号检测电平为高"H"
1	信号检测电平为低"L"

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。

- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。
- 设为“0”时：
 - 信号检测电平为高"H"
 - 同步模式下，发送数据输出与串行时钟的下降沿同步；SPI模式下，发送数据输出与串行时钟的上升沿同步
 - 同步模式下，接收数据输出与串行时钟的上升沿同步；SPI模式下，接收数据输出与串行时钟的下降沿同步
- 设为"1"时：
 - 信号检测电平为低"L"
 - 同步模式下，发送数据输出与串行时钟的上升沿同步；SPI模式下，发送数据输出与串行时钟的下降沿同步
 - 同步模式下，接收数据输出与串行时钟的下降沿同步；SPI模式下，接收数据输出与串行时钟的上升沿同步

[位 13] CS2 SPI: 片选引脚 2 下的模式选择位

片选引脚 2 下的 SPI 模式选择设置。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS2 有效。

位	说明
0	普通同步模式
1	SPI 模式

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 12] CS2TDS: 片选引脚 2 下的方向选择位

设置传输方式是低位优先(LSB)还是高位优先(MSB)。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS2 有效。

位	说明
0	LSB 优先 (最低位先传输)
1	MSB 优先 (最高位先传输)

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。

- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 11:8]CS2 DL3, DL2, DL1, DL0: 片选引脚 2 下的数据长度选择位

设置通过 SCS2 引脚传输数据的长度，在片选引脚形式使能(ECR:CSFE=1)时，此位有效。此位仅对片选引脚 SCS2 有效。

位 11	位 10	位 9	位 8	说明
0	0	0	0	8 位长
0	0	0	1	5 位长
0	0	1	0	6 位长
0	0	1	1	7 位长
0	1	0	0	9 位长
0	1	0	1	10 位长
0	1	1	0	11 位长
0	1	1	1	12 位长
1	0	0	0	13 位长
1	0	0	1	14 位长
1	0	1	0	15 位长
1	0	1	1	16 位长

注意：

- 禁止设置除上述之外的其他值。
- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 7]CS1CSLVS: 串行片选 1 极性设置位

设置串行片选引脚在无效状态下的电平极性是高“H” 或者低“L”，在片选格式有效(ECR:CSFE=1)的情况下。

此位仅对 SCS1 引脚有效。

位	说明
0	SCS1 在无效状态时，电平为“L”。
1	SCS1 在无效状态时，电平为“H”。

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 6]CS1CINV:片选引脚 1 下的串行时钟反转位

反转串行时钟形式。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS1 有效。

位	说明
0	信号检测电平为高"H"
1	信号检测电平为低"L"

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
 - 此位在从机模式(CR:MSS=1)无效。
 - 在片选引脚禁止(ECR:CSFE=0)时，此位无效。
- 设为“0”时：
 - 信号检测电平为高"H"
 - 同步模式下，发送数据输出与串行时钟的下降沿同步；SPI模式下，发送数据输出与串行时钟的上升沿同步
 - 同步模式下，接收数据输出与串行时钟的上升沿同步；SPI模式下，接收数据输出与串行时钟的下降沿同步
 - 设为"1"时：
 - 信号检测电平为低"L"
 - 同步模式下，发送数据输出与串行时钟的上升沿同步；SPI模式下，发送数据输出与串行时钟的下降沿同步
 - 同步模式下，接收数据输出与串行时钟的下降沿同步；SPI模式下，接收数据输出与串行时钟的上升沿同步

[位 5] CS1SPI: 片选引脚 1 下的模式选择位

片选引脚 1 下的 SPI 模式选择设置。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS1 有效。

位	说明
0	普通同步模式
1	SPI 模式

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 4] CS1TDS: 片选引脚 1 下的方向选择位

设置传输方式是低位优先(LSB)还是高位优先(MSB)。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS1 有效。

位	说明
0	LSB 优先 (最低位先传输)
1	MSB 优先 (最高位先传输)

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE=“0”)时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 3:0]CS1 DL3, DL2, DL1, DL0: 片选引脚 1 下的数据长度选择位

设置通过 SCS1 引脚传输数据的长度，在片选引脚形式使能(ECR:CSFE=1)时，此位有效。此位仅对片选引脚 SCS1 有效。

注意：

- 禁止设置除上述之外的其他值。

位 3	位 2	位 1	位 0	说明
0	0	0	0	8 位长
0	0	0	1	5 位长
0	0	1	0	6 位长
0	0	1	1	7 位长
0	1	0	0	9 位长
0	1	0	1	10 位长
0	1	1	0	11 位长
0	1	1	1	12 位长
1	0	0	0	13 位长
1	0	0	1	14 位长
1	0	1	0	15 位长
1	0	1	1	16 位长

此位只能在数据收/发停止(CR:TXE=RXE=“0”)时进行设置

此位在从机模式(CR:MSS=1)无效

在片选引脚禁止(ECR:CSFE=0)时，此位无效。

寄存器配置(CSFR2)

位	15-8	7	6	5	4	3	2	1	0
字段名	-	CS3 CSLVS	CS3 CINV	CS3 SPI	CS3 TDS	CS3 DL3	CS3 DL2	CS3 DL1	CS3 DL0
属性	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值		1	0	0	0	0	0	0	0

寄存器功能

[位 15-8]保留位，读 0，写入无效

[位 7]CS3CSLVS: 串行片选 3 极性设置位

设置串行片选引脚在无效状态下的电平极性是高“H”或者低“L”，在片选格式有效

(ECR:CSFE=1)的情况下。

此位仅对 SCS3 引脚有效。

位	说明
0	SCS3 在无效状态时，电平为“L”。
1	SCS3 在无效状态时，电平为“H”。

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 6]CS3CINV:片选引脚 3 下的串行时钟反转位

反转串行时钟形式。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS3 有效。

位	说明
0	信号检测电平为高"H"
1	信号检测电平为低"L"

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
 - 此位在从机模式(CR:MSS=1)无效。
 - 在片选引脚禁止(ECR:CSFE=0)时，此位无效。
- 设为“0”时：
 - 信号检测电平为高"H"
 - 同步模式下，发送数据输出与串行时钟的下降沿同步；SPI 模式下，发送数据输出与串行时钟的上升沿同步
 - 同步模式下，接收数据输出与串行时钟的上升沿同步；SPI 模式下，接收数据输出与串行时钟的下降沿同步
 - 设为"1"时：
 - 信号检测电平为低"L"
 - 同步模式下，发送数据输出与串行时钟的上升沿同步；SPI 模式下，发送数据输出与串行时钟的下降沿同步
 - 同步模式下，接收数据输出与串行时钟的下降沿同步；SPI 模式下，接收数据输出与串行时钟的上升沿同步

[位 5] CS3SPI: 片选引脚 3 下的模式选择位

片选引脚 3 下的 SPI 模式选择设置。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS3 有效。

位	说明
0	普通同步模式
1	SPI 模式

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 4] CS3TDS: 片选引脚 3 下的方向选择位

设置传输方式是低位优先(LSB)还是高位优先(MSB)。在片选引脚形式使能(ECR:CSFE=1)时，此位有效。

此位仅对片选引脚 SCS3 有效。

位	说明
0	LSB 优先 (最低位先传输)
1	MSB 优先 (最高位先传输)

注意：

- 此位只能在数据收/发停止(CR:TXE=RXE="0")时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

[位 3:0]CS3 DL3, DL2, DL1, DL0: 片选引脚 3 下的数据长度选择位

设置通过 SCS3 引脚传输数据的长度，在片选引脚形式使能(ECR:CSFE=1)时，此位有效。此位仅对片选引脚 SCS3 有效。

位 3	位 2	位 1	位 0	说明
0	0	0	0	8 位长
0	0	0	1	5 位长
0	0	1	0	6 位长
0	0	1	1	7 位长
0	1	0	0	9 位长
0	1	0	1	10 位长
0	1	1	0	11 位长
0	1	1	1	12 位长
1	0	0	0	13 位长
1	0	0	1	14 位长
1	0	1	0	15 位长
1	0	1	1	16 位长

注意：

- 禁止设置除上述之外的其他值。
- 此位只能在数据收/发停止(CR:TXE=RXE=“0”)时进行设置。
- 此位在从机模式(CR:MSS=1)无效。
- 在片选引脚禁止(ECR:CSFE=0)时，此位无效。

33.7.12 发送字节数寄存器(TBYTE3-0)

发送字节数寄存器(TBYTE3-0) 用于设置对应片选有效下的数据传输个数。

寄存器配置

位	15	14	13	12	11	10	9	8
字段名	TBYTE1							
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	TBYTE0							
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8
字段名	TBYTE3							
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段名	TBYTE2							
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

在对应的片选引脚有效的情况下，传输发送字节数寄存器 TBYTE 所定义的个数的数据。当所定义的个数发送完成，片选引脚释放为无效状态。

片选引脚 0(SCS0) 对应于 TBYTE0; 片选引脚 1(SCS1)对应于 TBYTE1; 片选引脚 2(SCS2) 对应于 TBYTE2; 片选引脚 3(SCS3)对应于 TBYTE3。

当片选禁止(CSCR:CSEN3-0="0000")时，发送字节数寄存器 0(TBYTE0)应用于与传输同步定时器，由定时器启动同步传输操作后，发送字节数寄存器 0(TBYTE0)所设置的数据个数将被传输。

如在传输过程中(SR:TBIF=0)改变该值。该值将在之前所设置的个数个数据传输完成后生效。

TBYTE	说明
写入	向 TBYTE 写值.
读取	TBYTE 设置的值

注意：

- 当该值设置为(00)h，传输个数为 8。
- 此位在从机模式(CR:MSS=1)无效。

33.7.13 波特率生成器寄存器(BRS1,BRS0)

波特率生成器寄存器 1,0 (BRS1, BRS0)设置串行时钟的分频比

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	-	(BRS1)							(BRS0)							
属性	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

寄存器功能

按字或者半字进行写访问波特率生成器寄存器(BRS1, BRS0)。

波特率生成器寄存器设置串行时钟的分频比。

BRS1 对应高位， BRS0 对应低位， 可写入需要计数的重载值以及读取设置值。

向波特率生成器寄存器 1,0 (BRS1, BRS0)写重载值后， 重载计数器开始计数。

[位 15] 保留位， 写入无效， 读取值不定

[位 14:8] BRS1: 波特率生成器寄存器 1

位 14:8	说明
写入	写入重载计数器位 8 ~ 位 14
读取	读取 BRS1 的设置值

[位 7:0] BRS0: 波特率生成器寄存器 0

位 7:0	说明
写入	写入重载计数器位 0 ~ 位 7
读取	读取 BRS0 的设置值

注意：

- 按 16/32 位写访问波特率生成器寄存器(BRS1, BRS0)。
- 变更波特率生成器寄存器(BRS1, BRS0)的设置值时， 计数值变为"15h00"后， 重载新的设置值。因此，如需使新的设置值立刻生效，变更 BRS1/0 的设置值后，应执行编程清零(PCL)。
- 重载值为奇数时，串行时钟的"H"宽和"L"宽相同。重载值为偶数时：
 - 如果 MR:CINV="0"， "H"宽比"L"宽长 1 个总线时钟周期
 - 如果 MR:CINV="1"， "L"宽比"H"宽长 1 个总线时钟周期
- 应向 BRS1/0 设置大于 3 的值。

33.8 SPI 使用注意事项

本节介绍使用注意事项。

- 当主机模式(CR:MSS="0")下的正常传送模式(CR:SPIMODE="0")下使用片选时，设置建立时间需满足下列条件：

$$\text{保持时间} + \text{建立时间} < \text{波特率转换值} - 2 \times t_{CYCP}$$

$$\text{波特率转换值}/2 < \text{保持时间} + 3 \times t_{CYCP}$$

波特率转换值：依据波特率对应的时间(定义)

t_{CYCP} : APB 总线周期

计算示例：

假如波特率: 1 [Mbps] (波特率对应的时间值是: 1 [μs]), 外设总线时钟: 48 [MHz] (周期: 约 20 [ns]) 且 CSCR:CSTODIV="0", 保持时间和建立时间按照如下计算:

- 保持时间:

$$CSTMG:CSHDLY \times t_{CYCP} \times 2CSCR:CSTODIV = CSTMG:CSHDLY \times 20[\text{ns}]$$

- 建立时间:

$$CSTMG:CSSDLY \times t_{CYCP} \times 2CSCR:CSTODIV = CSTMG:CSSDLY \times 20[\text{ns}]$$

依据上述公式, CSTMG:CSHDLY 与 CSTMG:CSSDLY 的对应组合为下表 表 33.8-.

表 33.8-1 建立时间和保持时间的设置条件(计算示例)

CSTMG:CSHDLY 值	CSTMG:CSSDLY 值
23 及以上	任意值
22	25 及以下
21	26 及以下
20	27 及以下
:	:
1	46 及以下
0	47 及以下

- 当主机模式(CR:MSS=0) 且 SPI 传输(CR:SPIMODE=1), 传输个数为“1”(TBYTE=1) 和片选保持功能开启时按照下面条件使用 SPI:

配置为“无串行数据收/发的等待”(ECR:DTRWS1, DTRWS0 ="00")

34. I²C 接口 (I²C 通信控制接口)

本章介绍多功能串行接口的工作模式 3 支持的 I²C 功能。

34.1 I²C 接口概要

I²C 接口(I²C 通信控制接口)支持 I²C 总线，可用作 I²C 总线上的主机/从机设备。

I²C 接口(I²C 通信控制接口)的功能

		功能
1	数据缓冲	半双工双缓冲器
2	串行输入	最多过滤串行时钟/串行数据输入中 2 个总线时钟的噪声
3	传输模式	同步
4	波特率	具有专用波特率发生器(由 15 位重载计数器构成)
5	数据长	8 位
6	信号方式	NRZ(非归零)
7	中断请求	<ul style="list-style-type: none">· 接收中断· 发送中断· 状态中断请求
8	I ² C	<ul style="list-style-type: none">· 主机/从机收/发功能· 仲裁功能· 时钟同步功能· 传输方向检测功能· 反复启动条件的发生和检测功能· 总线错误检测功能· 通用寻址功能· 作为主机/从机的 7 位寻址· 传输以及总线错误时可发生中断· 10 位寻址功能可通过程序支持

34.2 I²C 接口操作

34.2.1 I²C 接口中断

关于 I²C 接口的中断，以下中断源可使其发生中断请求。

- 收/发第 1 个字节后且收/发数据完成后
- 停止条件
- 反复启动条件

表 34.2-1 介绍 I²C 接口的中断控制位和中断源。

表 34.2-1 I²C 接口的中断控制位和中断

中断类型	中断请求标志位	标志寄存器	中断源	中断源使能位	中断请求标志清零
状态	INTF	BC	收/发第 1 个字节后 ^{*1}	BC:INTFE	中断标志位 (BC:INTF)清"0"
			收/发数据后 ^{*1}		
			总线错误检测 (EBCR:BCE=0)		
			仲裁丢失检测		
			保留地址检测		
			NACK 接收		
接收	STPCC	BS	停止条件	BC:CNDDIE	STPCC 清"0"
			反复启动检测		ITSCC 清"0"
发送	TDEF	SR	保留地址接收	MR:RXIE	读取接收数据 (RXDR)
			数据接收完成		
			FRIIE="1" 时检出接收空闲		
接收	RDFF	SR	溢出错误	MR:TXIE	接收错误标志位(SR:RECLR) 置"1"
			发送寄存器放空		
发送	ORF	SR	发送缓冲放空标志设置位(SR:TEFS)置"1"	MR:TXIE	向发送数据 (TXDR)写值

^{*1} 收/发数据时，如需产生 BC:INTF 标志，应在设置 BC:INTF 之前将 SR:TDEF 位设为"1"。

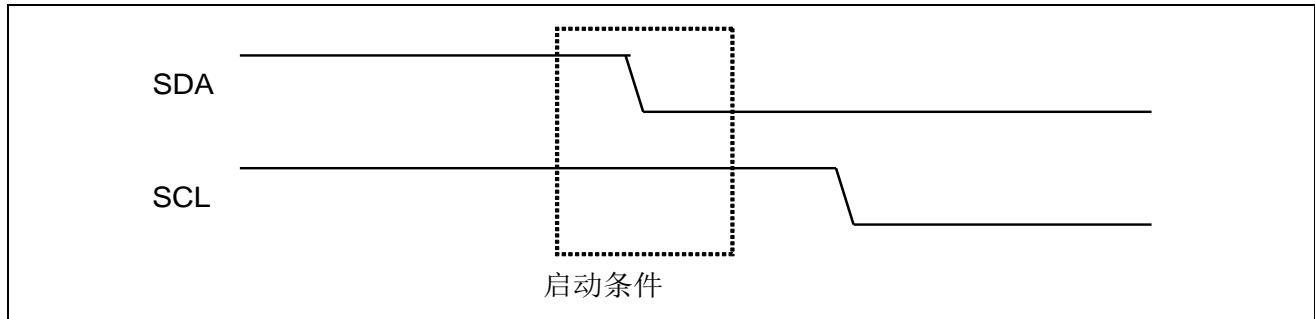
34.2.2 I²C 总线操作

I²C 接口使用 2 条双向总线、一条串行数据线(SDA)和一条串行时钟线(SCL)进行通信。

I²C 总线启动条件

以下是 I²C 总线的启动条件：

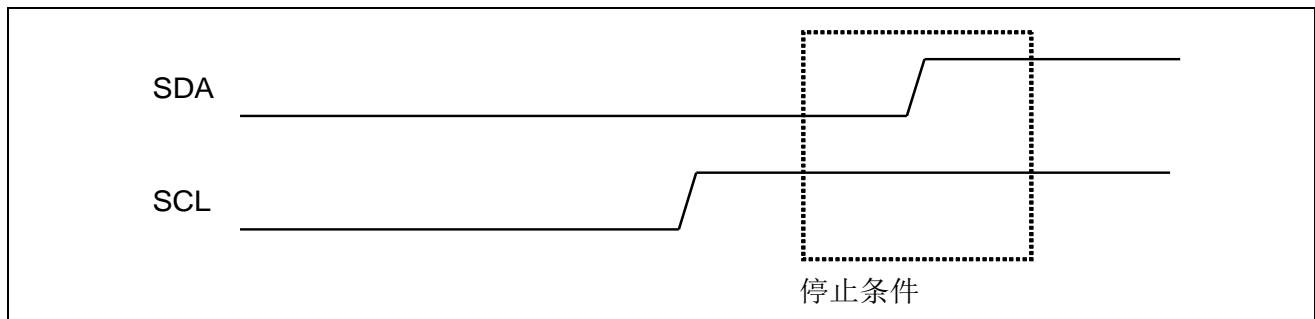
图 34.2-1 启动条件



I²C 总线停止条件

以下是 I²C 总线的停止条件：

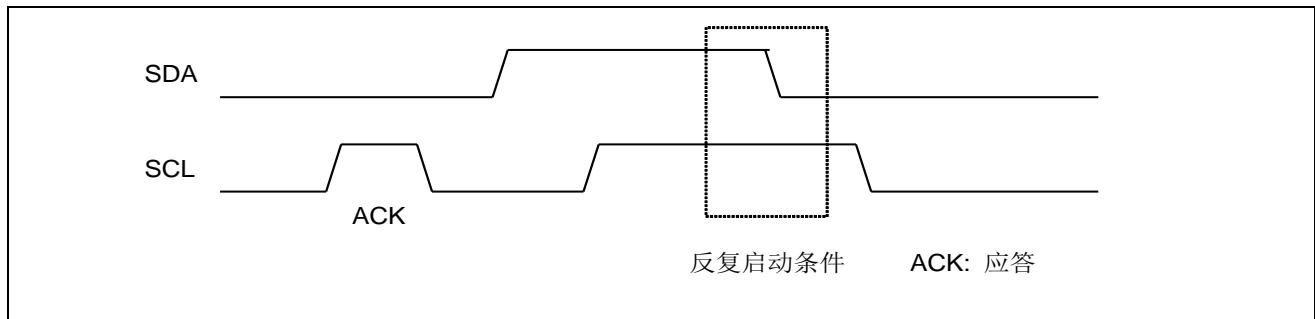
图 34.2-2 停止条件



I²C 总线反复启动条件

以下是 I²C 总线的反复启动条件：

图 34.2-3 反复启动条件



34.2.3 主机模式

主机模式时 I²C 总线产生启动条件并向 I²C 总线输出时钟。I²C 总线处于空闲状态(SCL="H", SDA="H")时, 如果将 BC 寄存器的 MSS 位设为"1", 则变为主机模式, BC 寄存器的 OFITS 位为"1"。

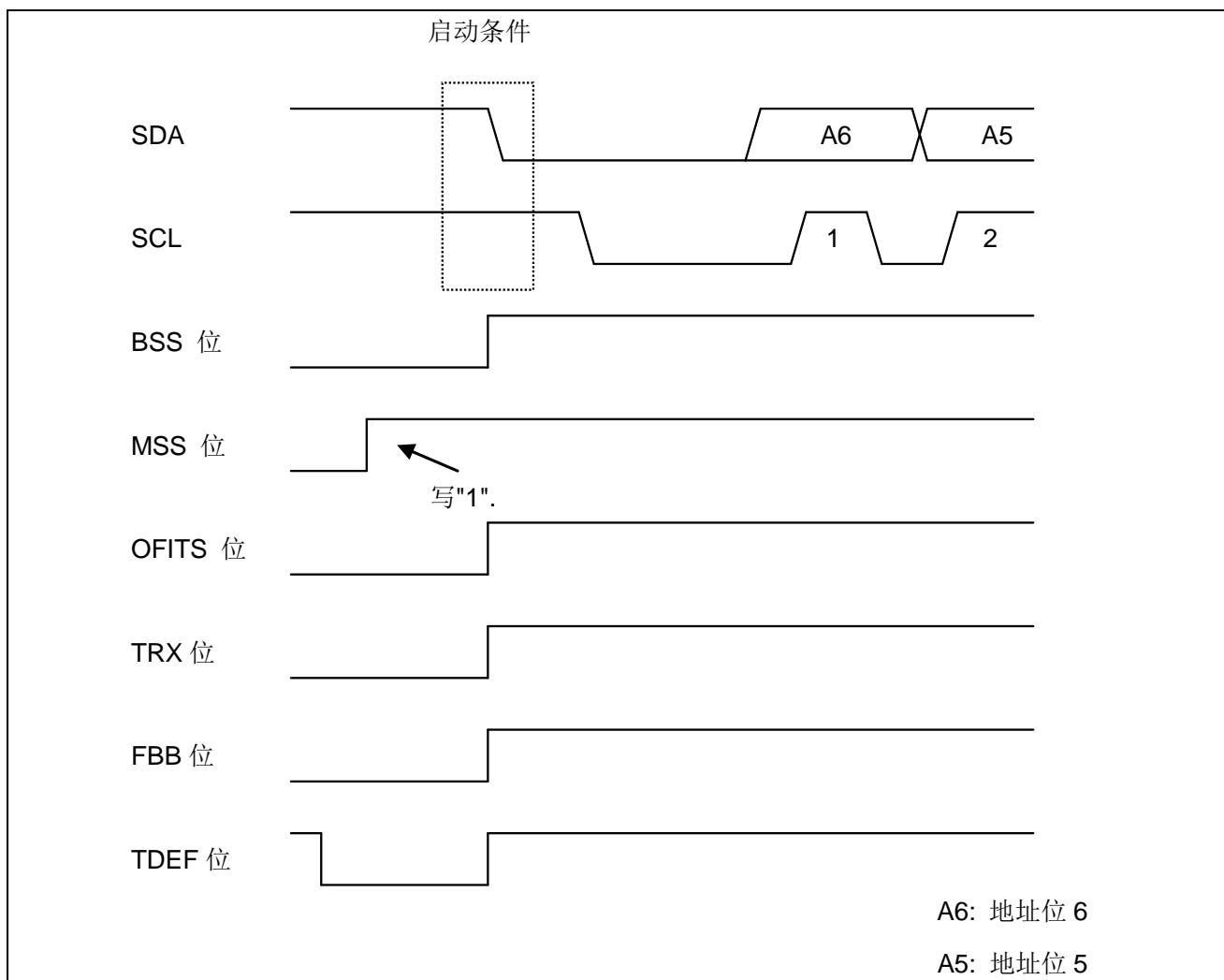
34.2.3.1 产生启动条件

以下条件下产生启动条件:

SDA="H", SCL="H", SAMS:EN=1 和 BS:BSS=0 时, 向 BC:MSS 位写入"1"

如果向 I²C 总线输出启动条件, 则将 BC:OFITS 位设为"1"。然后, 如果接收启动条件, 则 BS:BSS 位设为"1", 表示 I²C 总线正在执行通信(参考图 34.2-4)。

图 34.2-4 启动条件输出以及各位的关系



注意:

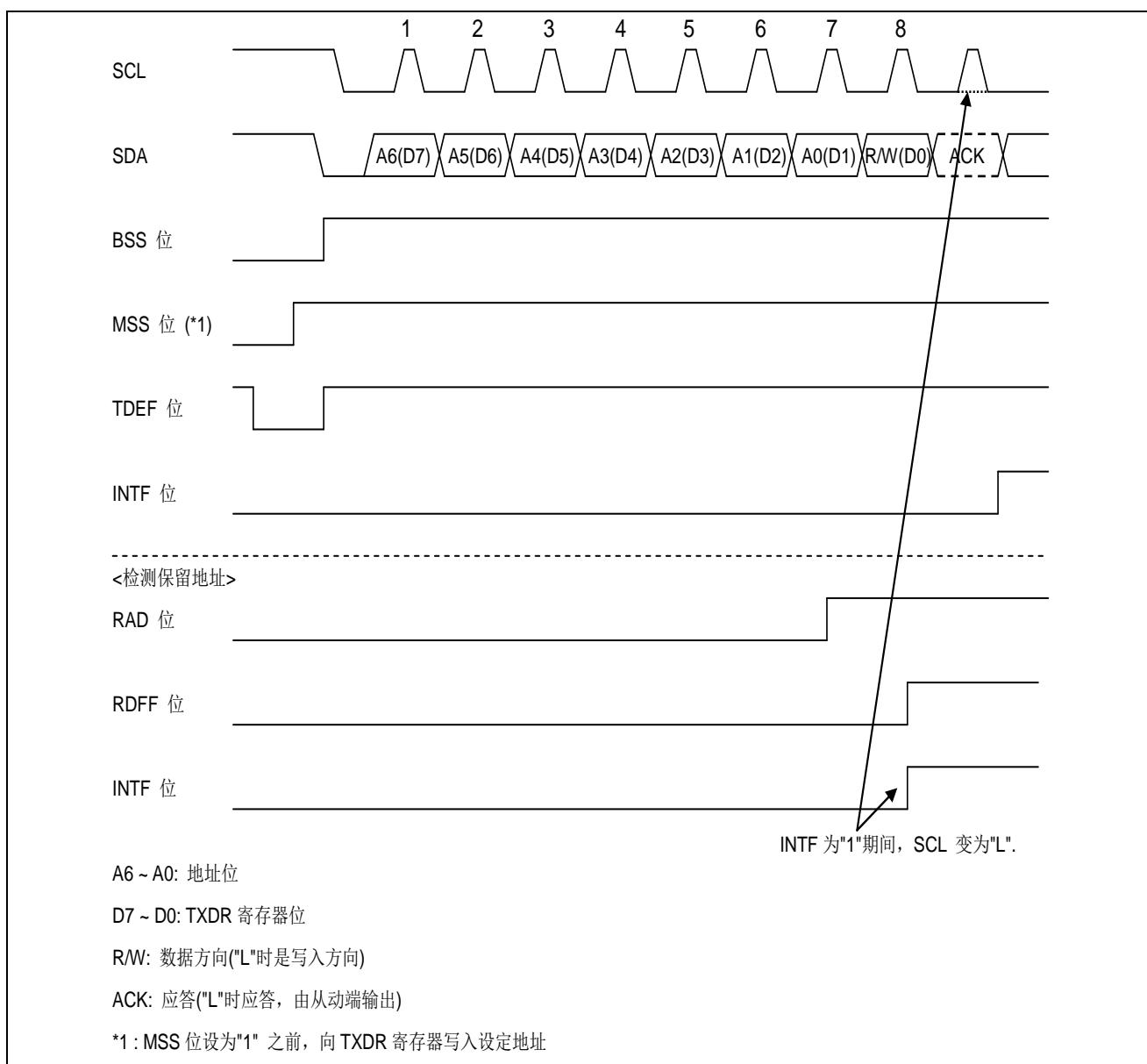
- 工作模式 3 (I²C 模式)时, 总线时钟在频率不低于 8 MHz 时使用。还应注意的是, 禁止将波特率生成器设置为 400 kbps 以上。

34.2.3.2 从机地址输出

如果输出启动条件, 那么 TXDR 寄存器内的数据将作为地址输出, 数据从 bit7 位开始。bit0 位用作数据方向位(R/W), 数据方向位(R/W)清"0"时, 数据表示写入方向(主机→从机)。将 BC:MSS 位或 BC:OFITS 位置"1"之前, 先在 TXDR 寄存器中设置地址。

关于地址和数据方向的输出时序, 参考图 34.2-5。

图 34.2-5 地址和数据方向



34.2.3.3 第 1 个字节发送后的应答接收

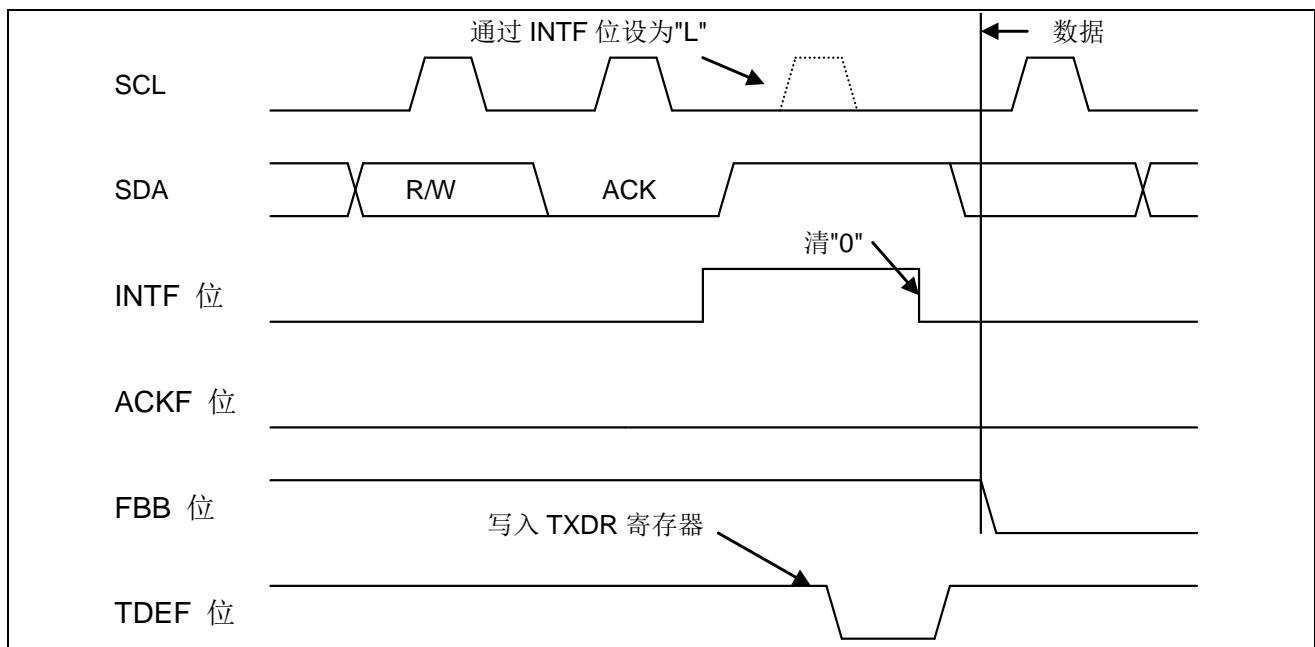
I²C 接口会在数据方向位 (R/W) 输出后，接收到一个从机的应答。下面是具体操作示例：

表 34.2-2 接收应答后的操作(BS:RAD 位="0",)

刚刚接收应答后的操作	
应答: ACK	应答: NACK
如果 SR:TDEF 位为"1"，将 BC:INTF 位设为"1"后等待。如果 SR:TDEF 位为"0"，则 BC:INTF 位为"0"保持不变，不等待。	将 BC:INTF 位设为"1"后等待

- BS:RAD 位为"0"时，接收应答后，如果 SR:TDEF 位为"1"，则中断标志(BC:INTF)置"1"并将 SCL 保持为"L"后等待。如果向中断标志写"0"，那么中断标志清"0"后解除等待状态。如果 SR:TDEF 位为"0"，接收 ACK 时，中断标志不置"1"的状态下 SCL 生成时钟。
- BS:RAD 位为"1"时，接收保留地址后(应答前)，将中断标志(BC:INTF)设为"1"并将 SCL 保持为"L"后等待。读取 RXDR 寄存器后，设置 BC:ACKE 位和发送数据，如果向中断标志写"0"，则中断标志清"0"后解除等待状态。
- 接收后的应答设置到 BS:ACKF 位。等待期间确认 BS:ACKF 位，如果是 NACK，则向 BC:MSS 位写"0"或向 BC:OFITS 位写"1"，使其生成停止条件或反复启动条件。这时，BC:INTF 位自动清"0"。

图 34.2-6 应答(BS:RAD="0", ACK 应答时)



以下是对于地址的等待时序：

- BS:RAD 位为"0"时，在接收应答后
 - BS:RAD 位为"1"时，在接收应答前
- 与 BC:WTSEL 无关。

图 34.2-7 应答(BS:RAD="0", NACK 应答时)

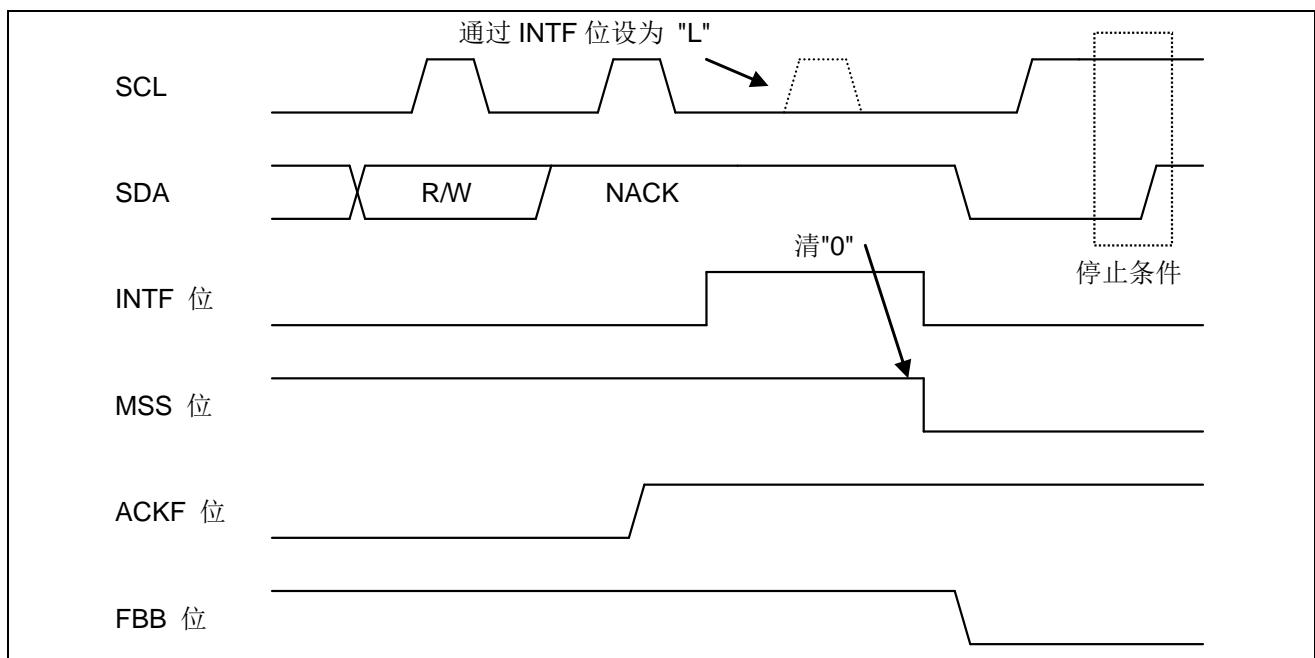


图 34.2-8 应答 (BS:RAD="1", ACK 应答时)

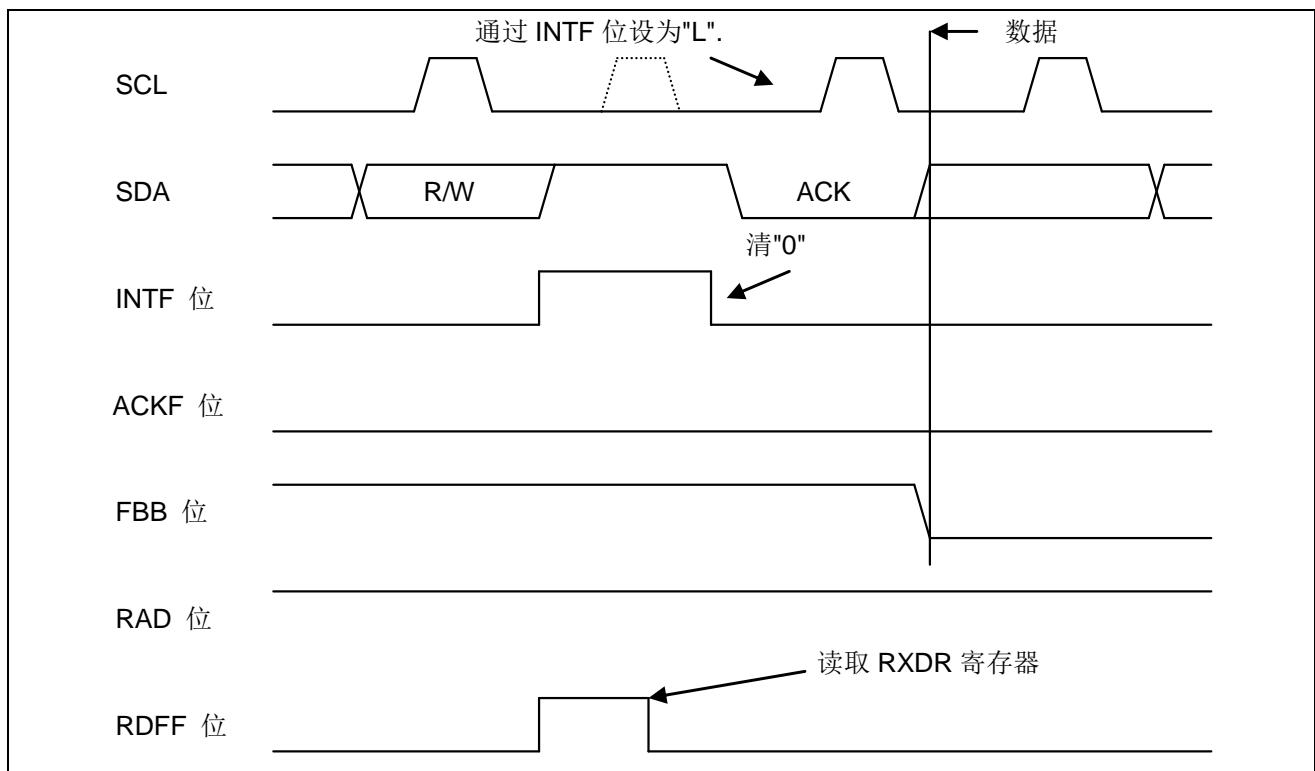
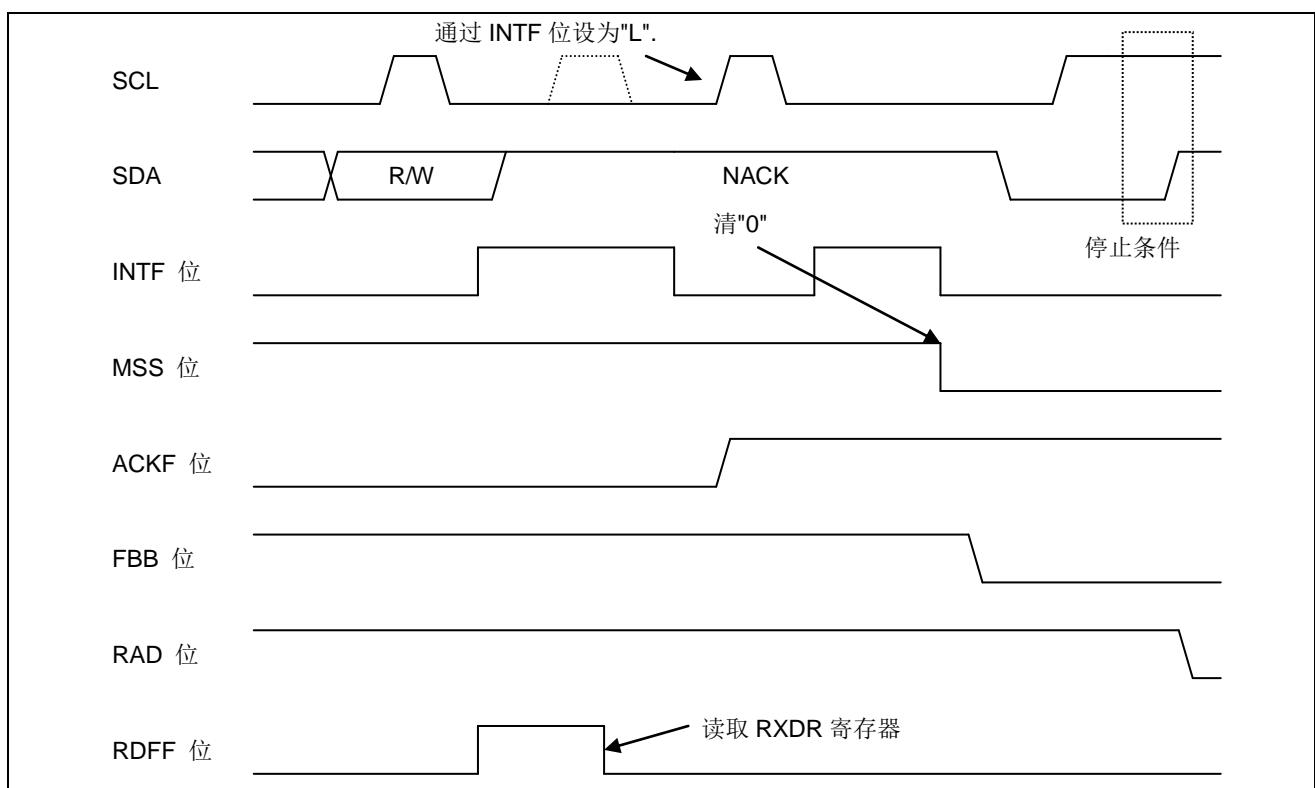


图 34.2-9 应答 (BS:RAD="1", NACK 应答时)



34.2.3.4 主机端发送数据

数据方向位(R/W)为"0"时，从主机端发送数据。每发送 1 个字节，从机端响应 ACK 或 NACK。因 BC:WTSEL 位的设置产生等待，如下所示：

表 34.2-3 主机端发送数据时的 BC:WTSEL 位

WTSEL 位	操作
0	第 2 个字节后，SR:TDEF 位为"1"或检出仲裁丢失时应答后，将中断标志(BC:INTF)设为"1"并将 SCL 保持在"L"后进入等待状态。
1	第 2 个字节后，SR:TDEF 位为"1"或检出仲裁丢失时主机端发送 1 字节数据后，将中断标志(BC:INTF)设为"1"并将 SCL 保持在"L"后进入等待状态。

但是，以下情况下，不取决于 BC:WTSEL 的设置，应答后设置中断标志(BC:INTF)。

- 除停止条件设置(BC:MSS="0", OFITS="1")时以外接收 NACK 时

以下是向从机端发送数据时的步骤示例：

向保留地址以外的地址发送数据时

1. 将从机地址(含数据方向位)设置到 TXDR 寄存器后，向 BC:MSS 位写"1"。
2. 发送从机地址后接收 ACK，中断标志(BC:INTF)置"1"。
3. 向 TXDR 寄存器写发送数据。
4. 更新 BC:WTSEL 位的同时向中断标志(BC:INTF)写"0"并解除 I2C 总线的等待状态。
5. 发送 1 字节后 BC:WTSEL="0"时，接收应答后 BC:WTSEL="1"时，刚刚发送 1 字节后将中断标志设为"1"并等待 I2C 总线。重复执行步骤 3~5，直到全部指定数据发送完成。不过，如果在等待状态已释放且 BC:WTSEL=1 的情况下收到了 NACK，将会在接收到应答后产生另外一个中断，且总线进入等待状态。
6. 向 BC:MSS 位写"0"或向 BC:OFITS 位写"1"，使其产生停止条件或反复启动条件。

向保留地址发送数据时

1. 作为从机地址将保留地址设置到 TXDR 寄存器后，向 BC:MSS 位设为"1"。
2. 发送从机地址后，中断标志(BC:INTF)置"1"。
3. 读取 RXDR 寄存器后确认保留地址。(*1)
4. 向 TXDR 寄存器写发送数据。
5. 更新 BC:WTSEL 位的同时，将中断标志(BC:INTF)清"0"，解除 I2C 总线等待状态。
6. 发送 1 字节后 BC:WTSEL="0"时，应答接收后，BC:WTSEL="1"时，刚发送 1 字节后将中断标志设为"1"并等待 I2C 总线。重复执行步骤 4~6，直到全部指定数据发送完成。不过，

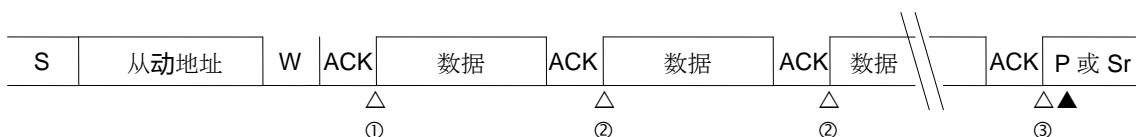
如果在等待状态已释放且 BC:WTSEL=1 的情况下收到了 NACK, 将会在接收到应答后产生另外一个中断, 且总线进入等待状态。

7. 向 BC:MSS 位写"0"或向 BC:OFITS 位写"1", 使其产生停止条件或反复启动条件。

*¹ 满足以下任一条件时, 将 BC:ACKE 位和 BC:WTSEL 位设为"1"后, 以检验下一个数据需要的是主机操作还是从机操作。

- 激活多主机模式且保留地址为全呼
- 检测到仲裁丢失后且接口可作为从机工作

图 34.2-10 产生主机端的中断 1(BC:WTSEL="0", BS:RAD="0")



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTF="1"引起中断

▲ : CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接应收应答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节 + 接收应答 ACK 时产生中断.

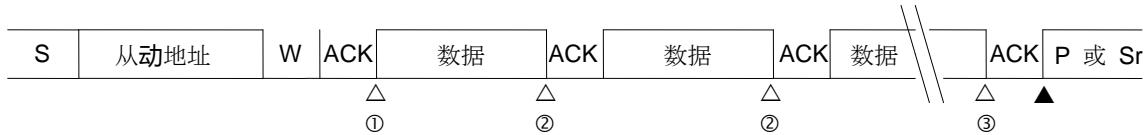
-向 TXDR 寄存器写发送数据后, INTF="0"

③发送 1 字节 + 接收应答 ACK 时产生中断.

- 设置 MSS ="0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

图 34.2-11 产生主机端的发送中断 2(BC:WTSEL="1", BS:RAD="0" ACK 应答)



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTE="1"引起中断

①发送从动地址 + 发送方向位+ 接收应答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节时产生中断

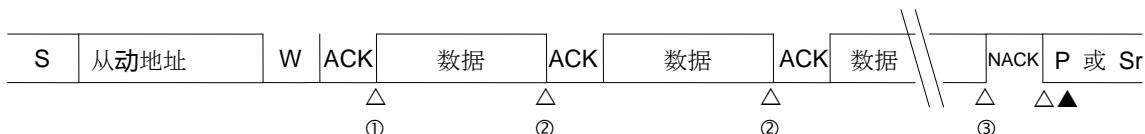
-向 TXDR 寄存器写发送数据后, INTF="0"

③发送 1 字节时产生中断

- 设置 MSS = "0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

图 34.2-12 产生主机端的发送中断 3(BC:WTSEL="1", BS:RAD="0" NACK 应答)



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTE="1"引起中断

▲ : CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接收应答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节时产生中断.

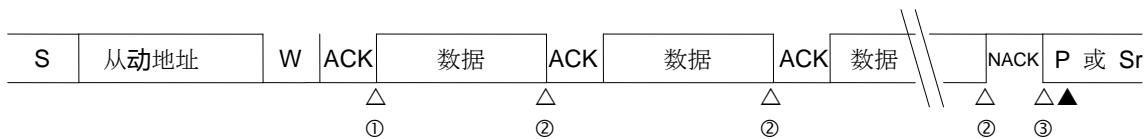
-向 TXDR 寄存器写发送数据后, INTF="0"

③发送 1 字节时产生中断

- 设置 MSS = "0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

图 34.2-13 产生主机端的发送中断 4(BC:WTSEL="1", BS:RAD="0" NACK 应答中)



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTF="1"引起中断

▲ : CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接应收答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节时产生中断.

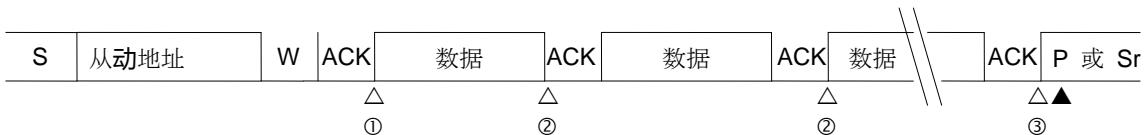
-向 TXDR 寄存器写发送数据后, INTF="0"

③通过 NACK 应答产生中断

- 设置 MSS = "0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

图 34.2-14 产生主机端的发送中断 5(BC:WTSEL="1"->"0", BS:RAD="0" ACK 应答)



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTF="1"引起中断

▲ : CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接应收答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节时产生中断.

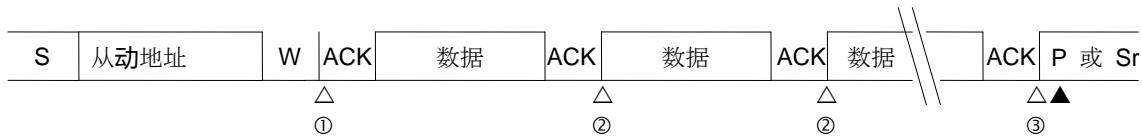
-向 TXDR 寄存器写发送数据后, WTSEL="0"和 INTF="0"

③发送 1 字节时产生中断.

- 设置 MSS = "0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

图 34.2-15 产生主机端的中断 6(BC:WTSEL="0", BS:RAD="1")



S: 启动条件

W: 数据方向位(写方向)

P: 停止条件

Sr: 反复启动条件

△ : INTF="1"引起中断

▲ : CNDDIE="1"引起中断

①发送从动地址 (保留地址) + 发送方向位+ 接收应答 ACK 时产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

②发送 1 字节 + 接收 ACK 应答时, 产生中断

-向 TXDR 寄存器写发送数据后, INTF="0"

③发送 1 字节 + 接收 ACK 应答时, 产生中断

- 设置 MSS ="0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF 位为 1".

34.2.3.5 主机端接收数据

数据方向位(R/W)为"1"时，接收自从机端发送的数据。

主机端执行以下操作：

- 如果 SR:TDEF 位是"1"，每接收 1 字节时发生等待(BC:INTF="1", SR:RDFF="1")。这时，如果 BC:WTSEL 位为"1"，那么处于等待前，如果 BC:WTSEL 位为"0"，那么位于等待后，通过设置 BC 寄存器的 ACKE 位应答 ACK 或 NACK。
- 如果 SR:TDEF 位清"0"，通过设置 BC 寄存器的 ACKE 位应答 ACK 时，不发生等待(BC:INTF="0")并接收下一条数据，如果应答 NACK 则发生等待(BC:INTF="1")。

关于中断引起的等待，参考下表：

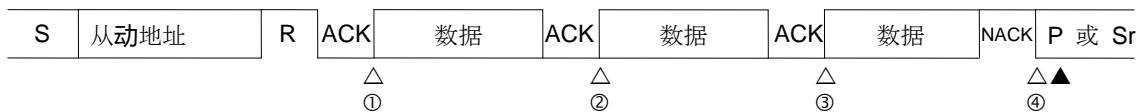
表 34.2-3 主机接收数据时的 BC:WTSEL 位

WTSEL 位	操作
0	第 2 字节之后，SR:TDEF 位为"1"时应答后，将中断标志(BC:INTF)设为"1"，将 SCL 设为"L"后设为等待状态。
1	第 2 字节之后，SR:TDEF 位为"1"时主机接收 1 字节数据后，将中断标志(BC:INTF)设为"1"，将 SCL 设为"L"后设为等待状态。

以下是自从机端接收数据时的步骤示例：

1. 将从机地址(含数据方向位)设置到 TXDR 寄存器后，将 BC:MSS 位设为"1"。
2. 发送从机地址后接收 ACK，中断标志(BC:INTF)为"1"。
3. 更新 BC:WTSEL 位的同时，将中断标志位(BC:INTF)清"0"，解除 I²C 总线等待。
4. 接收 1 字节后 BC:WTSEL="0"时发送应答后，BC:WTSEL="1"时刚接收 1 字节后，将中断标志设为"1"并等待 I²C 总线。接收指定的数据数前，重复 3.~4.的操作。
5. 接收最后的数据后，输出 NACK，将 BC:MSS 位设为"0"或将 BC:OFITS 位设为"1"，使其产生停止条件或反复启动条件。

图 34.2-16 主机端产生接收中断 1 (BC:WTSEL="0", BS:RAD="0")



△ :INTE="1"引起中断

▲ :CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接收应答ACK时产生中断

-如果INTF="0", 中断将清"0".

②接收 1 字节+发送应答 ACK 发生中断

- 读取接收数据后, INTF="0".

③接收 1 字节+发送应答 ACK 发生中断

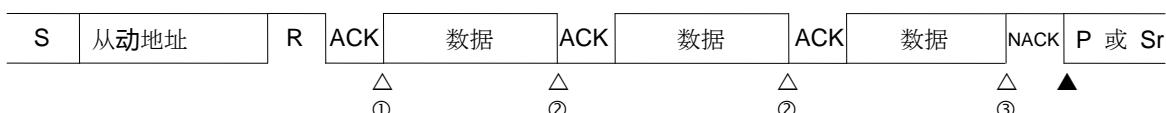
-读取接收数据后, ACKE="0", INTF="0".

④接收 1 字节+发送应答 ACK 发生中断. ,

- 设置MSS ="0", 或者 MSS="1"且OFITS="1"

*) 产生中断标志(INTF) 时, TDEF位为1".

图 34.2-17 主机端产生接收中断 2 (BC:WTSEL="1", BS:RAD="0")



△ :INTE="1"引起中断

▲ :CNDDIE="1"引起中断

①发送从动地址 + 发送方向位+ 接收应答ACK时产生中断

-如果INTF="0", 中断将清"0".

②接收 1 字节引起中断

- 读取接收数据后, INTF="0".

③接收 1 字节引起中断

-读取接收数据后, ACKE="0", MSS ="0", 或者 MSS="1"且 OFITS="1"

*) 产生中断标志(INTF) 时, TDEF位为1".

34.2.3.6 仲裁丢失

主机端的数据和其他主机的数据发生冲突，接收到的数据并不是已发送的数据时，视为仲裁丢失。此时，BC:MSS 位清"0"、BS:ABL 位置"1"后，可作为从机模式工作。

以下条件时，可 BS:ABL 位清"0"。

- BC:MSS 位为"1"
- BC:INTF 位清"0"
- BS:ABL 位="1", BS:STPCC 位="1"时，BS:STPCC 位清"0"
- 禁止 I²C 接口操作(SAMSK:EN 位="0")

发生仲裁丢失时，根据 BC:WTSEL 的设置将中断标志(BC:INTF)置"1"并将 I²C 总线的 SCL 保持在"L"。

34.2.3.7 主机模式等待

满足以下两个条件时，主机模式进入等待状态，同时 BS:BSS 位保持 "1"。BS:BSS 位清"0"后，发送启动条件。

- BS:BSS 位为"1"时，BC:MSS 位也设置为"1"
- 接口不作为从机模式时

主机模式是否是等待状态，可由 BC:MSS 位和 BC:OFITS 位进行判断(如果 BC:MSS="1", BC:OFITS="0"，则处于等待状态)。BC:MSS 位设为"1"后，作为从机模式工作时，将 BS:ABL 位设为"1"并将 BC:MSS 位设为"0"，BC:OFITS 位设为"1"。

34.2.4 从机模式

从机模式下检出(反复)启动条件后，与 SA 寄存器和 SAMS 寄存器的组合所接收的地址匹配时，执行 ACK 应答并作为从机模式工作。

注意

- 当 EBCR:BEC 清"0"，如果在传输地址数据或者响应数据过程中再次检测到启动条件将产生总线错误(BC:BERF = 1)，接收操作将停止。在此状况下，必须清除中断标志(BC:INTF)再重新由主机启动传输操作。

34.2.4.1 从机地址匹配检测

一旦检出(反复)启动条件，则将下一条数据的 7 位作为地址进行接收。关于 SAMS 寄存器设为"1"的位，比较 SA 寄存器和接收地址的各位后，如果二者匹配，则输出 ACK。

表 34.2-4 对从机地址输出 ACK 后的操作

刚刚应答后的操作	
应答: ACK	应答: NACK
如果 SR:TDEF 位为"1"，则将 BC:INTF 位设为"1"后等待。如果 SR:TDEF 位为"0"，则 BC:INTF 位保持"0"且不等待。	BC:INTF 位保持"0"且不等待。

- 保留地址检测

在第 1 个字节与保留地址("0000xxxx"或"1111xxxx")匹配时，接收第 8 位数据后，将 BC:INTF 位设为"1"并等待 I²C 总线。这时，读取接收数据后，如下设置：

- 如要作为从机端工作，将 BC:ACKE 设为"1"后确认数据方向位(BS:TRX)，如果是发送方向，则将发送数据写入 TXDR 并清零 BC:INTF 位。然后，作为从机端工作。
- 如无需作为从机端工作，将 BC:ACKE 设为"0"并清零 BC:INTF 位。输出应答后不作为从机工作。

34.2.4.2 数据方向位

接收地址后，开始接收数据方向位，该位用作判断数据收/发。该位清"0"时，显示主机端的发送数据，从机端接收数据。

34.2.4.3 从机端接收

从机地址匹配且数据方向位清"0"时，表示从机模式接收数据。以下是从机模式接收数据示例：

1. 发送 ACK 后，将中断标志(BC:INTF)设为"1"并等待 I²C 总线。使用 BC:MSS 位、BC:OFITS 位和 BS:FBSS 位判断从机地址匹配引起的中断，将 BC:ACKE 位设为"1"，中断标志(BC:INTF)设为"0"后解除 I²C 总线等待状态(参考表 34.2-4)。

2. 接收 1 字节数据后, 根据 BC:WTSEL 的设置将中断标志(BC:INTF)设为"1"并等待 I²C 总线。
3. 从 RXDR 寄存器读取接收数据, 设置 BC:ACKE 位后将中断标志(BC:INTF)清"0", 解除 I²C 总线等待状态。
4. 检出停止条件或反复启动条件前, 重复上述 2. ~ 3.的操作。

图 34.2-18 从机端接收中断 1(BC:WTSEL="0", BS:RAD="0")

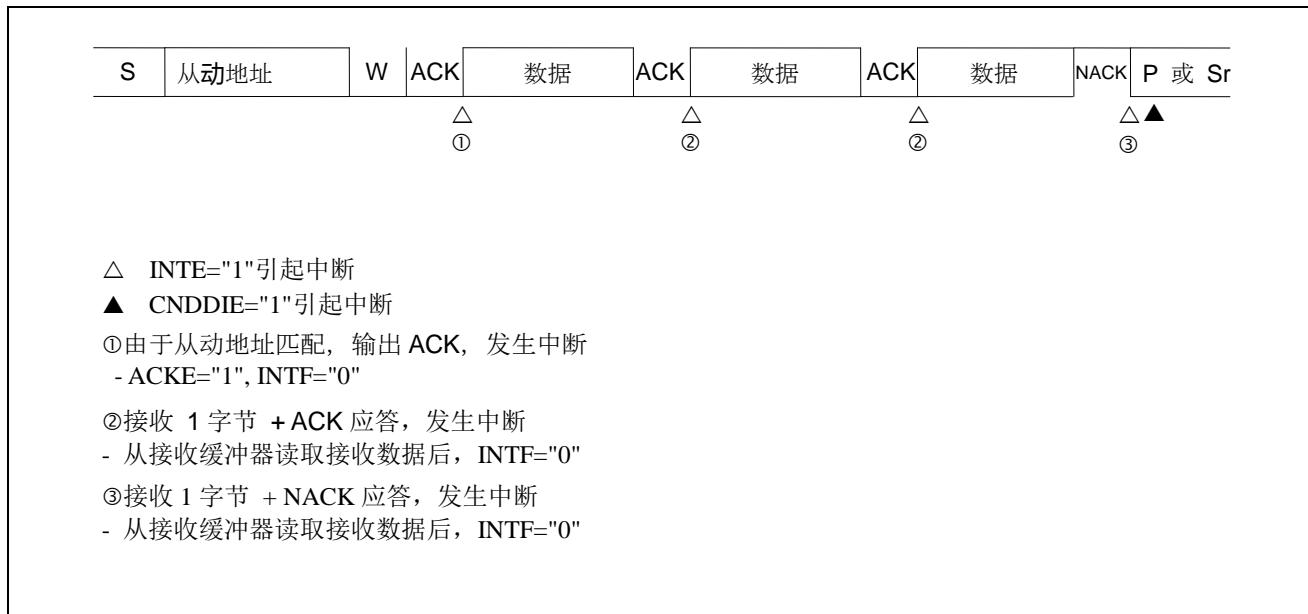
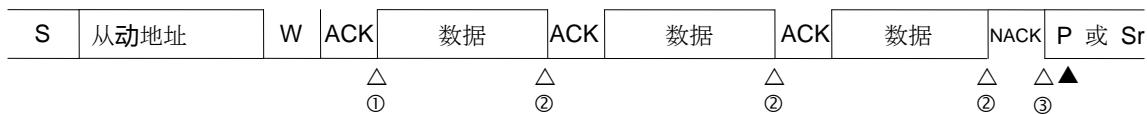


图 34.2-19 从机端接收中断 2(BC:WTSEL="1", BS:RAD="0")



图 34.2-20 从机端接收中断 3(BC:WTSEL="1", BS:RAD="0")



△ : INTF="1"引起中断

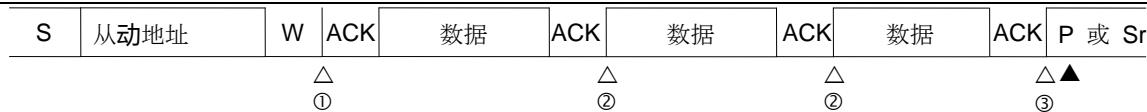
▲ : CNDDIE="1"引起中断

①由于从动地址匹配, 输出 ACK, 发生中断
- ACKE="1", INTF="0"

②接收 1 字节后发生中断
- 从接收缓冲器读取接收数据后, INTF="0"

③ NACK 应答后发生中断
- INTF="0"

图 34.2-21 从机端接收中断 4(BC:WTSEL="0", BS:RAD="1")



△ INTF="1"引起中断

▲ CNDDIE="1"引起中断

①保留地址("0000xxxx"或"1111xxxx")匹配时, 发生中断
- 读取接收数据, ACKE="1", INTF="0"

②接收 1 字节 + 应答 ACK 输出时发生中断
- INTF="0"

③接收 1 字节 + 应答 ACK 输出时发生中断
- INTF="0"时发生中断

34.2.4.4 从机端发送

从机地址匹配, 数据方向位为"1"时, 表示从机端发送数据。通过设置 BC:WTSEL, 发送 1 字节后或 ACK 应答后, 将中断标志(BC:INTF)设为"1"并产生等待(参考表 34.2-4)。

可通过 BS:ACKF 位确认主机端输出的 ACK 应答。如果从主机端收到 NACK 应答, 表示主机端无法正常接收或数据接收完成。BC:WTSEL="1"时, 检出 NACK 时, 发生中断, I²C 进入等待状态。

34.2.5 总线错误

I²C 总线上收/发数据期间，如果检出停止条件或(反复)启动条件，则视为总线错误进行处理。

34.2.5.1 总线错误发生条件

以下条件时，总线错误将 BC:BERF 位设为"1":

- 第 1 字节传输期间，检出(反复)启动条件或停止条件
- 数据的第 2 ~ 9 (应答)位时检出(反复)启动条件或停止条件

34.2.5.2 总线错误操作

EBCR:BEC=0

收/发操作时，若中断标志(BC:INTF)置"1"，则确认 BC:BERF 位，BC:BERF 位为"1"时，执行错误处理。将 BC:INTF 位设为"0"可清零 BC:BERF 位。

总线错误时，BC:INTF 位设为"1"，但将 I2C 总线的 SCL 设为"L"后不进入等待状态。

EBCR:BEC=1

收/发操作时，若中断标志(BC:STPCC 或 BC:ITSCC)置"1"，则确认 BC:BERF 位。BC:BERF 位为"1"时，执行错误处理。执行下列操作可清零 BC:BERF 位：

- 如果 BC:INTF=1，向 BC:INTF 位写 "0"
- 如果 BC:STPCC=1，向 BC:STPCC 位写 "0"
- 如果 BC:ITSCC=1，向 BC:ITSCC 位写 "0"

34.3 专用波特率发生器

34.3.1 波特率选择

专用波特率发生器(重载计数器)分频内部时钟所得到的波特率

使用波特率发生器寄存器 1, 0 (BRS1, BRS0)设置 15 位重载值后，可选择波特率。

重载计数器通过设置值分频内部时钟。

34.3.2 波特率计算方法

两个寄存器(BRS1 和 BRS0)组成的设置 15 位重载计数器的波特率设置值。

波特率计算公式如下：

1) 重载值

$$V = \phi / b - 1$$

V: 重载值 b: 波特率 ϕ : 总线时钟频率

有时在 I²C 总线的信号上升沿时间不会产生预设的波特率，应调整重载值

2) 计算示例

总线时钟为 16MHz，波特率为 400bps 时的重载值如下：

重载值：

$$V = (16 \times 1000000)/400000 - 1 = 39$$

因此，波特率为：

$$b = (16 \times 1000000) / (39+1) = 400 \text{ kbps}$$

注意：

- 以 16 位为单位写访问波特率发生器寄存器 1, 0 (BRS1, BRS0)。
- SAMSK 寄存器的 SAMSK:EN 位为"0"时，设置波特率发生器寄存器。
- I²C 模式时，总线时钟应不低于 8MHz，禁止设置超出 400kbps 的波特率发生器。
- 重载值为"0"时，重载计数器停止工作。

34.3.3 各总线时钟频率对应的重载值和波特率

表 34.3-1 重载值和波特率

波特率 [bps]	8 MHz	10 MHz	16 MHz	20 MHz	24 MHz	32 MHz
	重载值	重载值	重载值	重载值	重载值	重载值
1000000	禁止设置					
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

该数值是 I²C 总线的 SCL 上升为 0s 时的情况。I²C 总线的 SCL 上升延迟时，波特率比以上数值延迟。

如果不是上述值，参考前面的公式设置。关于系统频率，参考数据手册。

34.3.4 重载计数器功能

由重载值对应的 15 位寄存器构成，由内部时钟生成收/发时钟。此外，从波特率发生器寄存器(BRS1, BRS0)可读取发送重载计数器的计数值。

34.3.5 计数开始

如果向波特率发生器寄存器(BRS1, BRS0)写重载值，则重载计数器开始计数。

34.4 I²C 通信操作流程示例

图 34.4-1 I²C 流程示例 1/3

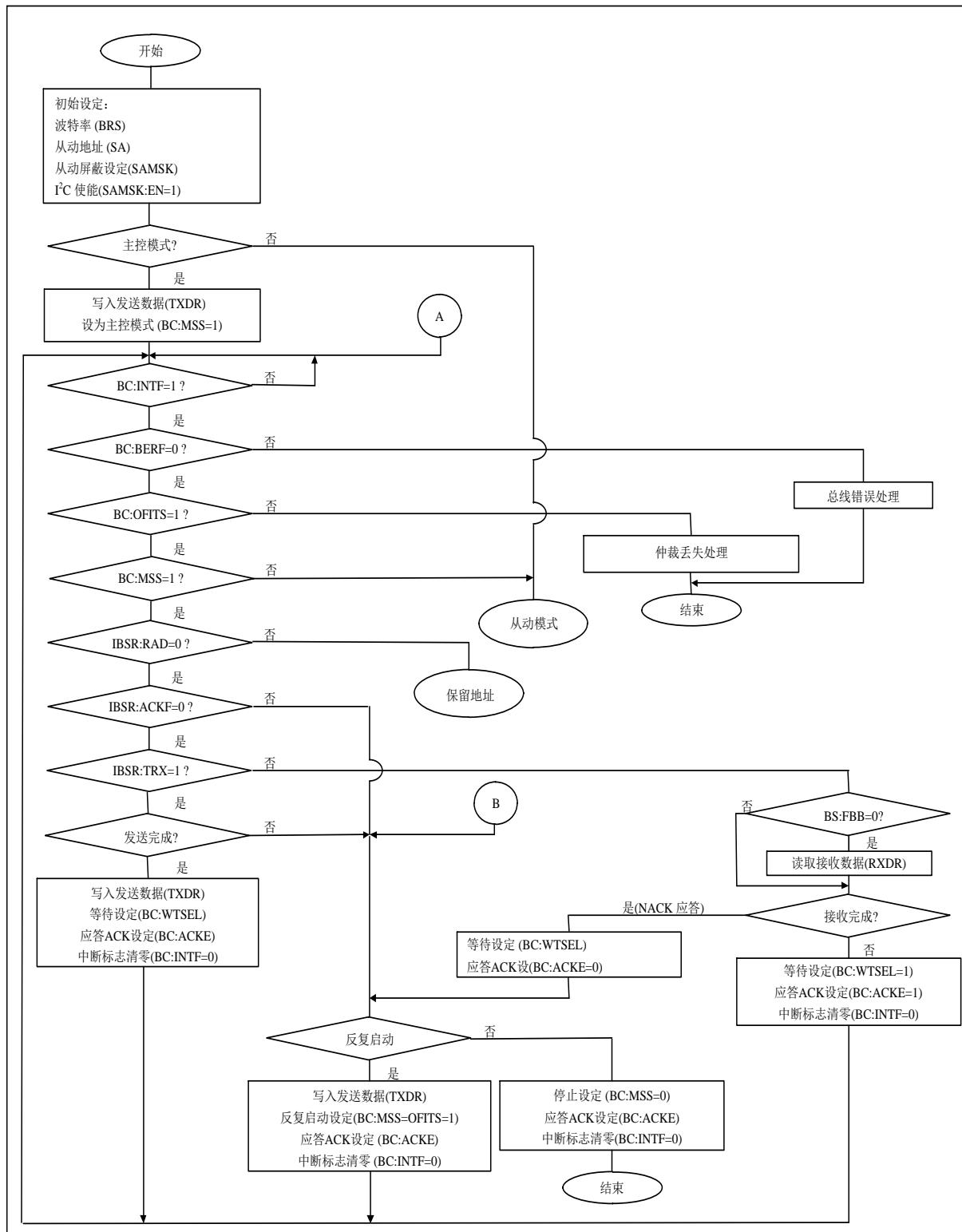


图 34.4-2 I²C 流程示例 2/3

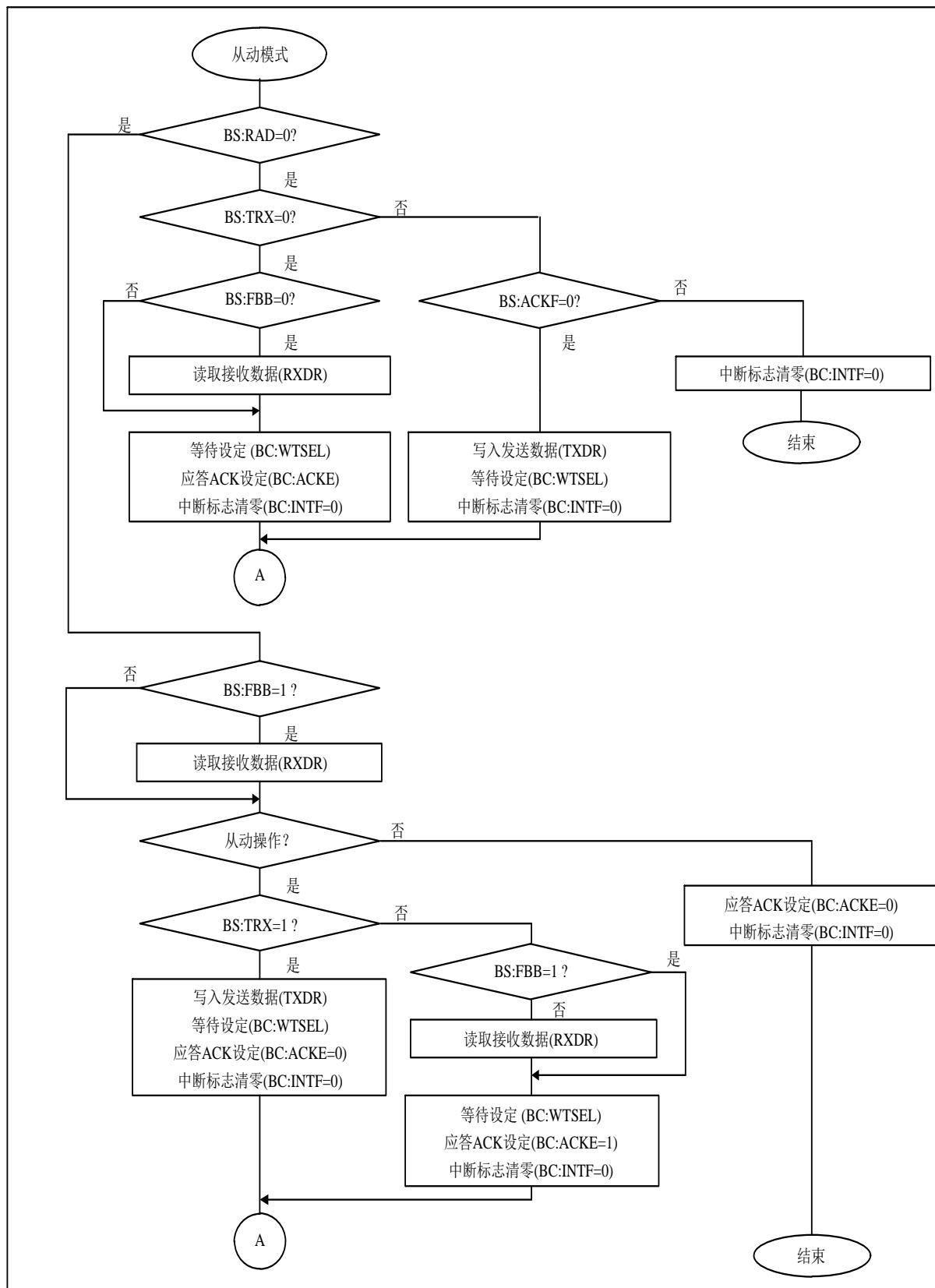
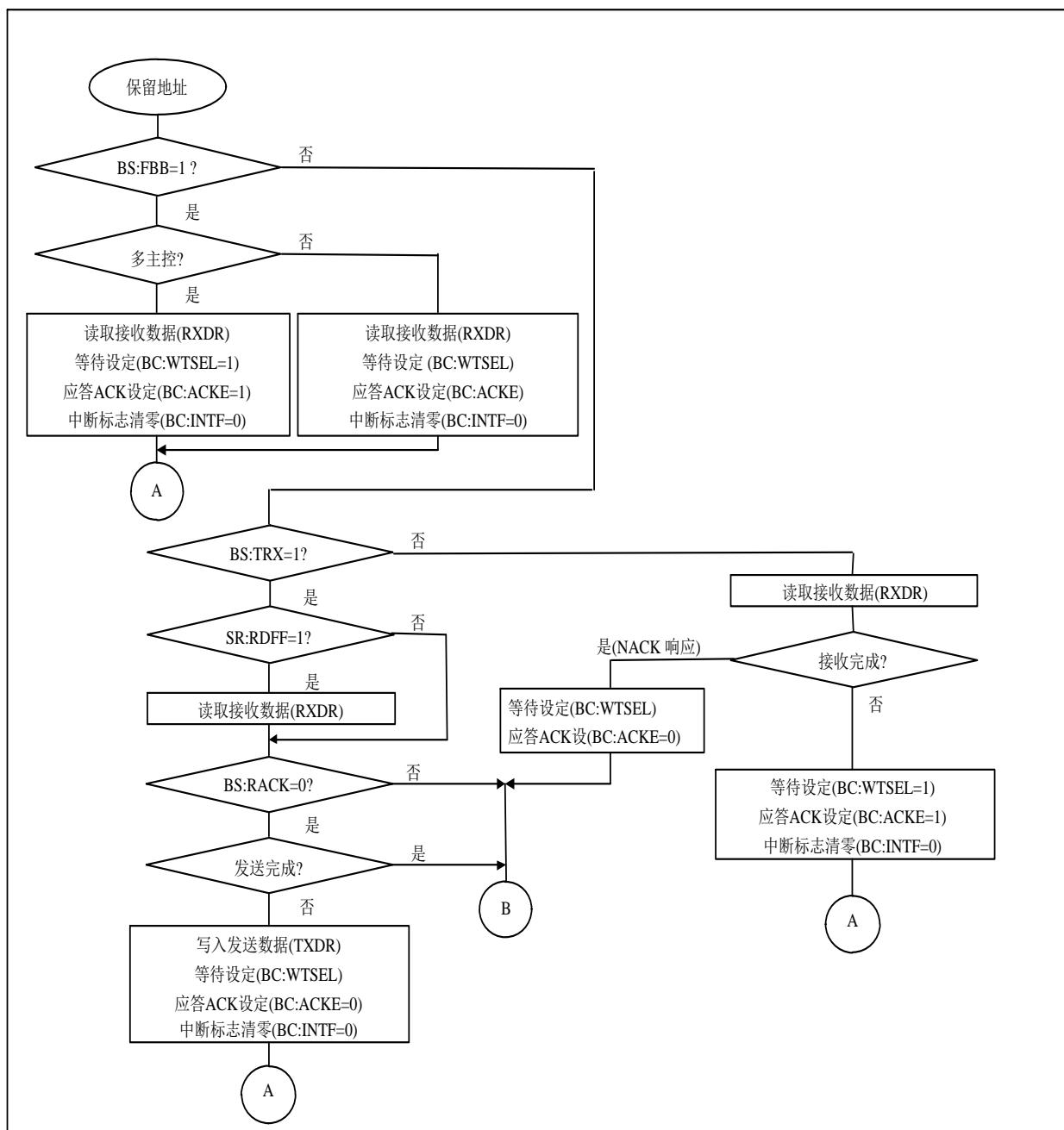


图 34.4-3 I²C 流程示例 3/3



注意：

- 以上流程为 I²C 模式时的操作设置概略。需配合适当的应用需要，考虑错误处理等情况。

34.5 I²C 接口寄存器

I²C 接口寄存器一览表

表 34.5-1 I²C 接口寄存器一览表

	位 15	位 8	位 7	位 0
I ² C	BC (I ² C 总线控制寄存器)			MR (串行模式寄存器)
	SR (串行状态寄存器)			BS (I ² C 总线状态寄存器)
	-			RXDR/TXDR (收/发数据寄存器)
	EBCR (扩展 I ² C 总线状态寄存器)			-
	BRS1 (波特率发生器寄存器 1)			BRS0 (波特率发生器寄存器 0)
	SAMS (7 位从机地址屏蔽寄存)			SA (7 位从机地址寄存器)

表 34.5-2 I²C 接口的位配置

	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
BC/ MR	MSS	OFI TS	AC KE	WT SEL	CN DDI E	INT FE	BER F	INT F	MO DE2	MO DE1	MO DE0	-	RXI E	TXI E	-	-
SR/ BS	REC LR	TEF S	-	-	ORF	RDF F	TDE F	-	FBS S	AC KF	RA D	TRX	ABL	ITS CC	STP CC	BSS
RXDR / TXDR	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
EBCR/ NFCR	-	-	SDA S	SCL S	SDA 0C	SCL 0C	DOE	BEC	-	-	-	-	-	-	-	-
BRS1/ BRS0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
SAMS K/ SA	EN	MS K6	MS K5	MS K4	MS K3	MS K2	MS K1	MS K0	AE	AD DR6	AD DR5	AD DR4	AD DR3	AD DR2	AD DR1	AD DR0

34.5.1 串行模式寄存器(MR)

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	(BC)			MOD E2	MOD E1	MO DE0	保留	RXI E	TXIE	保留	
属性				R/W	R/W	R/W	-	R/W	R/W	-	
初始值				0	0	0	-	0	0	-	

寄存器功能

[位 7:5] MODE2, MODE1, MODE0: 工作模式设置位，这些位设置工作模式

* 本节介绍工作模式 3 (I²C 模式)时的寄存器。

位 7	位 6	位 5	说明
0	0	0	工作模式 0 (异步 UART 模式)
0	0	1	工作模式 1 (异步多处理器 UART 模式)
0	1	0	工作模式 2 (SPI 模式)
1	0	0	工作模式 3 (I ² C 模式)
其他值		禁止设置	

注意：

- 如需切换工作模式，禁止 I²C (SAMSKE:EN="0")后再切换。
- 设置工作模式后，设置各寄存器。

[位 4] 保留位，读取值"0". 需写入 "0"

[位 3] RXIE: 接收中断使能位

使能/禁止向 CPU 输出接收中断请求。

RXIE 位和接收数据标志位(SR:RDFF)为"1"时或错误标志位(SR:ORF)置"1"时，输出接收中断请求。

位	说明
0	禁止接收中断
1	使能接收中断

注意：

- 使用 I²C 总线控制寄存器(BC)的 INTF 位接收数据时，应将该位清"0"。

[位 2] TXIE:发送中断使能位

使能/禁止向 CPU 输出发送中断请求

TXIE 位和 SR:TDEF 位都为"1"时，输出发送中断请求。

位	说明
0	禁止发送中断
1	使能发送中断

说明：使用 I²C 总线控制寄存器(BC)的 INTF 位发送数据时，应将该位清"0"。

[位 1:0] 保留位，读取值"00". 需写入 "00"

34.5.2 I²C 总线控制寄存器(BC)

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	MSS	OFITS	ACK E	WTS EL	CND DIE	INTF E	BER F	INTF	(MR)		
属性	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W			
初始值	0	0	0	0	0	0	0	0			

寄存器功能

[位 15] MSS:主机/从机选择位

若将该位设为"1", 则 I²C 总线为空闲状态(SAMSK:EN="1", BS:BSS="0")时, 进入主机模式。

BS 寄存器的 BSS 位为"1"时, 若将该设为"1", 则 BS:BSS 位清"0"前, 等待产生启动条件。等待期间, 从机地址匹配并用作从机端时, 该位清"0", BS 寄存器的 ABL 位为"1"。

主机操作期间(MSS="1", OFITS="1")中断标志(INTF)置"1"时, 若将该位清"0", 则产生停止条件。

MSS 位在下述条件时被清零:

- 禁止 I²C 接口操作(SAMSK:EN 位="0")
- 发生仲裁丢失时
- 检出总线错误(BERF 位="1")
- INTF="1"时, 向 MSS 位写"0"

以下是 MSS 位和 OFITS 位的关系:

MSS 位	OFITS 位	状态
0	0	空闲
0	1	从机地址匹配或对保留地址应答 ACK ^{*1} , 正在执行从机操作(从机模式)
1	0	主机操作待机中
1	1	主机操作期间(主机模式)

*1) ACK 应答:表示应答区间 I²C 总线的 SDA 为"L"。

位	说明
0	选择从机模式
1	选择主机模式

注意：

- 如果 MSS 位设置为 "1"，则只有在 MSS 位为 "1" 且 INTF 位为 "1" 时才必须将 MSS 位设置为 "0"。如果在 OFITS 位为 "1" 时 MSS 位设置为 "0"，则 INTF 位也清除为 "0"。
- 如果在 OFITS 位为 "1" 时 MSS 位设置为 "0"，则 INTF 位也清除为 "0"。
- 主机操作期间，即使 MSS 位清"0"，OFITS 位为"1"期间，仍读出"1"。

[位 14] OFITS：操作标志/反复启动条件发生位

该位的意思因读/写而异。

读取	写入
OFITS 位	OFITS 位

OFITS 位表示当前作为主机模式或从机模式操作。

OFITS 位的设置条件：

- 将启动条件输出到 I2C 总线时(主机模式)
- 从机地址与主机端发送的地址匹配时(从机模式)
- 检出保留地址并对其 ACK 应答时(MSS="0"时变为从机模式)

OFITS 位的复位条件：

<主机模式>

- 检出停止条件
- 检出仲裁丢失
- 检出总线错误
- 禁止 I2C 接口操作(SAMSK:EN 位="0")

<从机模式>

- (反复)检出启动条件
- 检出停止条件
- 保留地址检出状态(BS:RAD="1")下无应答时
- 禁止 I2C 接口操作(SAMSK:EN 位="0")
- 发生总线错误(BERF 位="1")

主机模式时，如果向该位写"1"，则执行反复启动。清"0"无效。

位	说明	
	写入	读取
0	无效	无效
1	产生反复启动条件	I ² C 操作期间

注意：

- 应在主机模式的中断期间(MSS="1", OFITS="1", INTF="1")，向 OFITS 位写"1"。OFITS 位为"1"时，若向 OFITS 位写"1"，则 INTF 位清"0"。
- 从机模式(MSS="0", OFITS="1")时、禁止向该位写"1"。
- OFITS 位= "1", MSS 位= "0"时，MSS 位优先。
- 通过读-修改-写类命令读取 OFITS 位。
- 满足下面两个条件时，INTF 位为"1"，等待 I²C 总线(SCL="L")。如需产生反复启动条件，需再次向 OFITS 位写"1"并清零 INTF 位。
 1. 第 8 位的主机模式中断期间(MSS="1", OFITS="1", INTF="1", WTSEL="1")，向 OFITS 位写"1"时
 2. 在第 9 位接收 NACK 时

[位 13] ACKE:数据字节 ACK 使能位

若将该位设为"1"，则根据 ACK 时序输出"L"。

应在下面任一条件时变更该位：

- OFITS="1", INTF 位= "1"时
- OFITS="0"时

以下条件时该位无效：

- 向保留地址除外的地址区应答(自动生成)
- 发送数据时(BS:RAD="0", BS:TRX="1", BS:FBSS="0")

位	说明
0	禁止应答
1	使能应答

[位 12] WTSEL: 等待选择位

该位在应答前或应答后产生中断(INTF="1")，选择是否达到 I²C 总线。

WTSEL 位在以下条件时无效。

1. 对第 1 字节^{*1}发生中断时(INTF="1")
2. 检出保留地址时(BS:FBSS="1", BS:RAD="1")

^{*1} 第 1 字节:指(反复)启动条件后的数据

位	说明
0	应答后等待(9 位)
1	数据收/发完成后等待(8 位)

[位 11] CNDDIE: 条件中断检测使能位

主机模式或从机模式时(OFITS="1")，检出停止条件或反复启动条件时，该位使能中断发生。BS 寄存器的 ITSCC 位或 STPCC 位为"1"且该位也为"1"时，发生中断。

位	说明
0	禁止反复启动或停止条件中断
1	使能反复启动或停止条件中断

[位 10] INTFE: 中断使能位

主机模式或从机模式时，该位使能数据收/发以及总线错误时的中断(INTF="1")。

位	说明
0	禁止中断
1	使能中断

[位 9] BERF: 总线错误标志位，该位表示 I²C 总线上检出错误。

BERF 位的置位条件：

- 第 1 字节^{*1} 传输期间，检出启动条件或停止条件
- 第 2 字节以后，在数据的第 2 ~ 9(ACK 应答)位检出(反复)启动条件或停止条件

^{*1} 第 1 字节：指(反复)启动条件后的数据

BERF 位的复位条件：

EBCR:BEC=0 且 BC:BERF="1" 时，向 INTF 位写"0"时

禁止 I²C 接口操作(SAMSK:EN=0)。

EBCR:BEC=1 且 BC:INTF=1 时，向 BC:INTF 位写"0"时

EBCR:BEC=1 且 BC:STPCC=1 时，向 BC:STPCC 位写"0"时

EBCR:BEC=1 且 BC:ITSCC=1 时，向 BC:ITSCC 位写"0"时

位	说明
0	无错误
1	检出错误

注意：

在下列情况下，中断标志(INTF 位)为"1"时，需确认该位，如果该位为"1"，则无法正常收/发数据，应重新发送等。

- EBCR:BEC=0 时 BC:BERF="1"
- EBCR:BEC=1 时反复启动条件确认位(BS:ITSCC)= "1"
- EBCR:BEC=1 时结束条件确认位(BS:STPCC)= "1"

[位 8] INTF: 中断标志位

主机模式或从机模式时，数据收/发第 8 位或第 9 位(ACK)后，或总线错误时，该标志设为"1"。若非总线错误时，INTF 位为"1"时，将 SCL 设为"L"，INTF 位为"0"时，解除 SCL 的"L"状。

INTF 位的置位条件：

<第 8 位>

- 第 1 字节检出保留地址时
- WTSEL 为"1"，第 2 字节之后检出仲裁丢失时
- WTSEL 为"1"，主机操作期间，第 2 字节以后 SR:TDEF 位为"1"时
- WTSEL 为"1"，从机操作期间，第 2 字节以后 SR:TDEF 位为"1"时
- WTSEL 为"1"，从机发送期间，第 2 字节以后 SR:TDEF 位为"1"时
- WTSEL 为"1"，从机接收时

<第 9 位>

- 在第 1 字节检出仲裁丢失时
- 若非设置停止条件输出(主机操作期间向 MSS 位写"0")时，接收 NACK 时
- 设为 WTSEL="0" 时，第 2 字节以后检出仲裁丢失时
- EBCR:BEC=1 且 BS:BERF=1 时
- 在第 1 字节未检出保留地址，在主机模式或从机模式的发送方向(BS:TRX="1")，SR:TDEF 位为"1"时
- 在第 1 字节未检出保留地址，在主机模式或从机模式的接收方向(BS:TRX="0")，SR:TDEF 位为"1"时
- 设为 WTSEL="0" 时，主机模式操作期间，第 2 字节以后 SR:TDEF 位为"1"时
- 设为 WTSEL="0" 时，从机发送期间，第 2 字节以后 SR:TDEF 位为"1"时
- 设为 WTSEL="0" 时，从机接收时。但是，在检出保留地址后的第 1 字节执行从机接收的状态下，在第 9 位不发生中断

<其他位>

- 在 EBCR:BEC=0 时检出总线错误

INTF 位的复位条件：

INTF 位清"0"。

INTF 位为"1"，OFITS 位为"1"时，MSS 位清"0"

INTF 位为"1"，OFITS 位为"1"时，向 OFITS 位写"1"

向 INTF 位写"1"无效

位	说明
---	----

	写入	读取
0	INTF 位清“0”。	没有发生中断请求
1	无影响	发生中断请求

注意：

- INTF 标志为"1"时，若 INTF 标志清"0"，则解除 I2C 总线等待状态。
- 将 SAMSK:EN 位设为"0"时，根据接收时序，有时 SR:RDFF 位和 INTF 位为"1"。这种情况下。应读取接收数据并清零 INTF 位。
- 使用读-修改-写类命令读出"1"。
- 启动条件发出时(BC:MSS=1)，请将本位置"1"。

34.5.3 I²C 总线状态寄存器(BS)

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	(SR)			FBSS	ACK F	RAD	TRX	ABL	ITSC C	STP CC	BSS
属性				R	R	R	R	R	R/W	R/W	R
初始值				0	0	0	0	0	0	0	0

寄存器功能

[位 7] FBSS: 第一节位

该位标志第 1 字节。

FBSS 位的设置条件:

- 检出(反复)启动条件时

FBSS 位的清零条件:

- 第 2 字节的收/发
- 检出停止条件
- 禁止 I²C 接口操作(SAMSK:EN 位="0")
- EBCR:BEC=0 时, 检出总线错误(BC:BERF 位="1")

位	说明
0	除第一节外
1	第一节收/发期间

[位 6] ACKF: ACK 应答标志位

该位表示在第 1 字节, 主机模式时或从机模式时接收的 ACK 应答。

ACKF 位的更新条件:

- 第一字节时的 ACK 应答
- 主机模式或从机模式时的数据应答

ACKF 位的清零条件(ACKF 位="0"):

- 检出(反复)启动条件
- 禁止 I²C 接口操作(SAMSK:EN 位="0")
- EBCR:BEC=0 时, 检出总线错误(BC:BERF 位="1")

位	说明
0	接收" L ".
1	接收" H ".

[位 5] RAD:保留地址检测位

该位表示检出保留地址。

RAD 位的设置条件(RAD="1"):

- 第 1 字节为(0000xxxx)或(1111xxxx)。"x"表示"0"或"1"。

RAD 位的复位条件(RAD="0"):

- 检出(反复)启动条件
- 检出停止条件
- 禁止 I2C 接口操作(SAMSK:EN 位="0")
- EBCR:BEC=0 时，检出总线错误(BC:BERF 位="1")

在第 1 字节 RAD 位为"1"时，该第 1 字节的第 8 位的 SCL 下降时，将中断标志(BC:INTF)设为"1"并将 SCL 设为"L"。这时，读取接收数据并作为从机端工作时，将 BC:ACKE 设为"1"，将中断标志(BC:INTF)清"0"。然后，如果 TRX 位清"0"，则作为从机接收数据。如果中途不想接收数据，则将 BC:ACKE 位清"0"。那么之后不接收数据。

位	说明
0	未检出保留地址
1	检出保留地址

注意：

- 数据传输期间，将 BC:ACKE 清"0"时，检出停止条件或反复启动条件前，禁止将 BC:ACKE 设为"1"。
- 检出保留地址后引起中断时，确认从机发送后，应设置 BC:ACKE="0"。

[位 4] TRX:数据方向位

该位表示数据方向。

TRX 位的置位条件:

- 主机模式下发生(反复)启动条件
- 从机模式下第 1 字节的第 8 位为"1"时(作为从机端发送方向)

TRX 位的复位条件:

- 仲裁丢失发生(ABL="1")
- 从机模式下优先字节的第 8 位为"0"时(作为从机端接收方向)
- 主机模式下优先字节的第 8 位为"1"时(作为主机端接收方向)
- 检出停止条件
- 非主机模式下检出(反复)启动条件
- 禁止 I2C 接口操作(SAMSK:EN 位="0")

- EBCR:BEC=0 时, 检出总线错误(BC:BERF 位="1")

位	说明
0	接收方向
1	发送方向

[位 3] ABL: 仲裁丢失标志位

该位表示仲裁丢失。

ABL 位的置位条件:

- 主机模式时, 输出数据和接收数据不一致时
- BC:MSS 位设为"1", 但作为从机工作时
- EBCR:BEC=0 时, 主机模式时, 在第 2 字节以后数据的第 1 位检出反复启动条件时
- EBCR:BEC=0 时, 主机模式时, 检出反复启动条件
- EBCR:BEC=1 时, 主机模式时, 在第 2 字节以后数据的第 1 位检出停止条件时
- EBCR:BEC=1 时, 主机模式时, 检出停止条件时(除在响应时检出停止条件外)
- 主机模式时, 想使其发生反复启动条件但无法发生时
- 主机模式时, 想使其发生停止条件但无法发生时

ABL 位的复位条件:

- BC:MSS 位设为"1"
- BC:INTF 位清 0"
- ABL 位="1", STPCC 位="1" 时, STPCC 位设为"0"
- 禁止 I2C 接口操作(SAMSK:EN 位="0")
- EBCR:BEC=0 时, 检出错误(BC:BERF 位="1")

位	说明
0	未发生仲裁丢失
1	发生仲裁丢失

[位 2] ITSCC:反复启动条件确认位

该位主机模式或从机模式时, 检出反复启动条件。

ITSCC 位的置位条件:

- EBCR:BEC=0 时, 从机模式或主机模式下, 操作期间 ACK 应答后, 检出反复启动条件
- EBCR:BEC=1 时, 从机模式或主机模式下, 在第一字节的第一位时检出反复启动条件

ITSCC 位的复位条件:

- ITSCC 位清"0"
- BC:MSS 位设为"1"

- 禁止 I2C 接口操作(SAMSK:EN 位="0")

该位直接写入"1"无效

位	说明
0	未检出反复启动条件
1	检出反复启动条件

注意：

- 检出保留地址后，作为从机模式执行接收操作期间，没有 ACK 应答时，结束从机模式，因此即使检出下一个反复启动条件，该位也不置"1"。
- 使用读-修改-写类命令读出"1"。

[位 1] STPCC:停止条件确认位

该位表示主机模式或从机模式时检出停止条件。

STPCC 位的置位条件：

- EBCR:BEC=0 时，从机模式或主机模式下，操作期间检出停止条件时
- EBCR:BEC=1 时，在下列情形之一检出停止条件：
 - BC:OFITS=0 下，第一字节时
 - 从机模式时
 - 主机模式时(除在响应时检出停止条件外)
- 主机模式下，发生停止条件后发生仲裁丢失时

STPCC 位的复位条件：

- 该位清"0"
- BC:MSS 位为"1"
- 禁止 I2C 接口操作(SAMSK:EN 位="0")

该位直接写入"1"无效

位	说明	
0	未检出停止条件	
1	主机	检出停止条件或输出停止条件输出时发生仲裁丢失
	从机	检出停止条件

注意：

- 检出保留地址后，作为从机模式接收期间，未有 ACK 应答时，结束从机模式，因此即使检出下一个停止条件，该位也不置"1"。
- 使用读-修改-写类命令读出"1"。
- 下列条件满足时，即使检出停止条件，该位不会被置"1"，主机操作也不会停止：

- 当 EBCR:BEC=1 时
- 主机模式时
- 在响应期间

[位 0] BSS: 总线状态标志位

该位表示总线状态

BSS 位的设置条件:

- 检出 I2C 总线的 SDA 或 SCL 为 "L" 时

BSS 位的复位条件:

- 检出停止条件时
- 禁止 I2C 接口操作(SAMSK:EN 位 = "0")
- EBCR:BEC=0 时，检出总线错误(BC:BERF 位 = "1")

位	说明
0	总线空闲状态
1	总线收/发状态

34.5.4 串行状态寄存器(SR)

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	REC LR	TEFS	-	-	ORF	RDF F	TDE F	-	(BS)		
属性	R/W	R/W	R	R/W	R	R	R	R			
初始值	0	0	0	0	0	0	1	1			

寄存器功能

[位 15] RECLR:接收错误标志清零位

该位清零串行状态寄存器(SR)的 ORF 位

位	说明	
	写入	读取
0	无效	始终读"0"
1	接收错误标志位(ORF)位清零	

[位 14] TEFS:发送缓冲放空标志设置位

该位设置串行状态寄存器(SR)的 TDEF 位。

此位设置为 "0" 时对操作无效

位	说明	
	写入	读取
0	无效	始终读"0"
1	发送数据放空标志位 (TDEF) 置位	

注意：

- 仅在 BC:INTF 位为"1"时， 该位可以置位"1"。

[位 11] ORF: 溢出错误标志位

如果接收时发生上溢，则该位为"1"， 如果向串行状态寄存器(SR)的 RECLR 位写"1"， 则该位清零。

ORF 位和 MR:RXIE 位为"1"时， 输出接收中断请求。

该标志置位时， 接收数据寄存器(RXDR)无效。

位	说明
0	无溢出错误
1	有溢出错误

[位 10] RDFF: 接收数据满载标志位

该标志表示接收数据寄存器(RXDR)的状态。

MR:RXIE 位和接收数据标志位(RDFF)为"1"时，输出接收中断请求。

如果接收数据载入 RXDR，则该位为"1"，如果读取接收数据寄存器(RXDR)，则该位清"0"

根据数据第 8 位的 SCL 下降时序设置该位。

即使应答 NACK (*1)，该位仍被置位。

在 8 个时钟计数期间，如果读取 RXDR，则计数器复位到"0"后，再次计数 8 个时钟。

*1) NACK 应答:应答期间，I²C 总线的 SDA 为"H"

位	说明
0	接收数据寄存器 RXDR 放空
1	接收数据寄存器 RXDR 内有数据

注意：

- 满足以下所有条件时，发送 ACK 后将 SCL 设为"L"，如果 RDFF 位清"0"，则 SCL 解除"L"状态。
 - 第 2 字节以后的接收操作期间(BS:TRX="0")，RDFF 位为"1"时
 - BC:WTSEL="0"
- 满足以下所有条件时，刚接收 1 字节数据后将 SCL 设为"L"，如果 RDFF 位清"0"，则 SCL 解除"L"状态。
 - 第 2 字节以后的接收操作期间(BS:TRX="0")，RDFF 位为"1"时
 - BC:WTSEL="1"

[位 9] TDEF: 发送数据放空标志位

该标志指示发送数据寄存器(TXDR)的状态。

MR:TXIE 位和 TDEF 位为"1"时，输出发送中断请求。

如果向 TXDR 写发送数据，则该标志为"0"，表示 TXDR 内存在有效数据。如果数据载入发送移位寄存器后开始发送，则该标志为"1"，表示 TXDR 内没有有效数据。

如果向串行状态寄存器(SR)的 TEFS 位写"1"，则该标志置位。检出仲裁丢失或总线错误时，用于将 TDEF 位设为"1"的情况。

位	说明
0	发送数据寄存器 TXDR 内有数据
1	发送数据寄存器 TXDR 放空

34.5.5 接收数据寄存器/发送数据寄存器(RXDR/TXDR)

接收数据寄存器(RXDR)和发送数据寄存器(TXDR)配置在同一地址。读取时，用作接收数据寄存器；写入时，用作发送数据寄存器。

接收数据寄存器(RXDR)

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名				D7	D6	D5	D4	D3	D2	D1	D0
属性				R	R	R	R	R	R	R	R
初始值				0	0	0	0	0	0	0	0

寄存器功能

接收数据寄存器(RXDR)是串行数据接收用的数据缓冲寄存器

串行数据线(SDA 引脚)发送的串行数据信号经由移位寄存器转换后，保存到接收数据寄存器(RXDR)。

收到第 1 字节^{*1}时，接收地址并不收纳到接收数据寄存器(RXDR)内。但是，如果第 1 字节是保留地址，接收地址被收纳到接收数据寄存器(RXDR)内。此时，最低位(RXDR:D0)为数据方向位。

如果接收数据保存到接收数据寄存器(RXDR)，则接收数据满载标志位(SR:RDFF)置"1"。

一旦读取接收数据寄存器(RXDR)，接收数据满载标志位(SR:RDFF)自动清"0"。

*1) 第 1 字节:(反复)启动条件后的数据。

发送数据寄存器(TXDR)

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名				D7	D6	D5	D4	D3	D2	D1	D0
属性				W	W	W	W	W	W	W	W
初始值				1	1	1	1	1	1	1	1

寄存器功能

发送数据寄存器(TXDR)是串行数据发送用的数据缓冲寄存器。

发送数据寄存器(TXDR)的值作为 MSB 优先，输出到串行数据线(SDA 引脚)。

发送第 1 字节时，最低位(TXDR:D0)为数据方向位。

如果发送数据写到发送数据寄存器(TXDR)，则发送数据放空标志(SR:TDEF)清"0"。

数据传输到发送用移位寄存器时，发送数据放空标志(SR:TDEF)置"1"。

如果数据放空标志(SR:TDEF)为"0"，则无法将发送数据写到发送数据寄存器(TXDR)。

注意：

- 发送数据寄存器为只写寄存器，接收数据寄存器为只读寄存器。两个寄存器配置到同一地址，因此读值和写值不同。无法使用执行 INC/DEC 命令等读-修改-写(RMW)操作的命令。

34.5.6 I²C 总线扩展控制寄存器(EBCR)

I²C 总线扩展控制寄存器(EBCR)用于控制 SDA/SCL 的输出和发生总线错误后操作行为。

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	保留	SDA S	SCL S	SDA 0C	SCL0 C	DOE	BEC	-	-	-	-
属性	-	R	R	R/W	R/W	R/W	R/W	R/W			
初始值	-	0	0	1	1	0	0				

寄存器功能

[位 15:14] 保留位，读取值为"0". 需写入"0".

[位 13] SDAS: SDA 状态位，该位指示经过滤波后的 SDA 的检测电平

位	说明
0	SDA 为低电平"L".
1	SDA 为高电平"H".

注意：

- 仅在使能 I²C (SAMSK:EN=1)时有效，如果禁止 I²C(SAMSK:EN=0)，该位固定返回值"0"。

[位 12] SCLS: SCL 状态位，该位指示经过滤波后的 SCL 的检测电平

位	说明
0	SCL 为低电平"L".
1	SCL 为高电平"H".

注意：

- 仅在使能 I²C (SAMSK:EN=1)时有效。如果禁止 I²C(SAMSK:EN=0)，该位固定返回值"0"。

[位 11] SDA0C: SDA 输出控制位

当串行输出控制使能(DOE=1)时，该位控制 SDA 的输出.

位	说明
0	SDA 输出为低电平"L".
1	SDA 输出为高电平"H".

[位 10] SCL0C: SCL 输出控制位

当串行输出控制使能(DOE=1)时，该位控制 SCL 的输出.

位	说明
0	SCL 输出为低电平"L".

1	SCL 输出为高电平 "H".
---	-----------------

[位 9] DOE: 串行输出控制位

该位禁止/使能 SCL 和 SDA 串行输出

当该位为"1"时,执行下面操作:

SDA 的输出由 SDA 输出控制位(SDA0C)控制.

SCL 的输出由 SCL 输出控制位(SCL0C)控制.

位	说明
0	禁止串行输出控制
1	使能串行输出控制

注意:

- 在 BC:MSS=0 和 BC:OFITS=0 时, 该位必须设置为 "1"。

[位 8] BEC: 总线错误控制位

在产生总线错误(BS:BERF=1)后, 该位选择继续还是停止 I2C 操作。

位	说明
0	I ² C 操作停止
1	I ² C 继续操作

注意:

- 当 EBCR:BEC=0, 如果在检测到开始条件后的传输地址数据或者位 2 到位 9 响应过程中检测到反复开始条件将发生总线错误(BC:BERF=1)和接收中止。因此, 下一数据将接收不到。在此情况下, 需主机清除中断标志位(BC:INTF), 并重启开始条件检测。

34.5.7 波特率发生器寄存器(BRS1,BRS0)

波特率发生器寄存器 1, 0 (BRS1, BRS0)设置串行时钟的分频比。

寄存器配置

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段名	-	(BRS1)										(BRS0)				
属性	-	R/ W														
初始值	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

波特率发生器寄存器设置串行时钟的分频比。

BRS1 和 BRS0 分别对应高位和低位，支持写入重载值和读取 BRS1/0 设置值。

如果向波特率发生器寄存器 1, 0 (BRS1, BRS0)写重载值，则重载计数器开始计数。

[位 15] -: 保留位，读取值不定，写入无效。

[位 14:8] BRS1: 波特率发生器寄存器 1

位 14:8	说明
写入	写入重载计数器位 8 ~ 位 14
读取	读取 BRS1 的设置值

[位 7:0] BRS0: 波特率发生器寄存器 0

位 7:0	说明
写入	写入重载计数器位 0 ~ 位 7
读取	读取 BRS0 的设置值

注意：

- 以 16 位为单位写访问波特率发生器寄存器(BRS1, BRS0)。
- SAMSK 寄存器的 EN 位为"0"时，设置波特率发生器寄存器。
- 与主机模式和从机模式无关，设置波特率。
- 工作模式 3 (I²C 模式)时，标准模式或快速模式时，总线时钟应大于 8MHz，禁止波特率发生器的设置超出 400kbps。

34.5.8 7 位从机地址屏蔽寄存器 (SAMSK)

7 位从机地址屏蔽寄存器(SAMSK)设置是否比较从机地址的各位。

寄存器配置

位	15	14	13	12	11	10	9	8	7	...	0
字段名	EN	MSK 6	MSK 5	MSK 4	MSK 3	MSK 2	MSK 1	MSK 0	(SA)		
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
初始值	0	1	1	1	1	1	1	1			

寄存器功能

[位 15] EN: I²C 接口操作使能位

该位使能/禁止 I²C 接口操作

清"0": I²C 接口变为禁止状态。

置"1": I²C 接口变为可工作状态

位	说明
0	禁止
1	使能

注意:

- 即使 BS 寄存器的 BERF 位设为"1", 该位也不清"0"。
- 该位清"0"时, 设置波特率发生器。
- 该位清"0"时, 设置 7 位从机地址和 7 位从机地址屏蔽寄存器。
- 禁止 I²C 接口(EN="0")后, 收/发立刻被禁止。
- 向 BC:MSS 位写"0"并发生停止条件后, 禁止 I²C 接口操作时, 确认发生停止条件后, 禁止操作(EN="0")。
- 发送期间, EN 位清"0"时, 有时 I²C 总线的 SDA/SCL 产生脉冲。

[位 14:8] MSK6:0:从机地址屏蔽位

该位决定是否将 7 位从机地址和接收地址设置在比较范围内。

置"1"对应的位:比较

清"0"对应的位:作为已匹配位处理

位 14:8	说明
0	不进行位比较
1	进行位比较.

注意:

- 只有在 SAMSK.EN 位="0"时配置该寄存器。

34.5.9 7 位从机地址寄存器 (SA)

7 位从机地址寄存器(SA)设置从机地址。

寄存器配置

位	15	...	8	7	6	5	4	3	2	1	0
字段名	(SAMSK)			AE	ADD R6	ADD R5	ADD R4	ADD R3	ADD R2	ADD R1	ADD R0
属性				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值				0	0	0	0	0	0	0	0

寄存器功能

[位 7] AE:从机地址使能位

该位使能从机地址检测。

清"0":未检出从机地址。

置"1":根据 SA, SAMSK 的设置对第一字节进行比较。

位	说明
0	禁止从机地址检测
1	使能从机地址检测

[位 6:0] ADDR6 to ADDR0: 7 位从机地址

关于 7 位从机地址寄存器(SA), 如果使能从机地址检测(AE="1"), 则检出(反复)启动条件后, 接收的 7 位数据与该寄存器进行比较, 如果所有位都匹配, 则用作从机模式, 输出 ACK。这时, 接收的从机地址设置到该寄存器(AE="0"时, 不输出 ACK)。

SAMSK 寄存器内设为"0"的地址位不在比较范围内。

位 6:0	说明
	7 位从机地址

注意:

- 禁止设置保留地址。
- 只有在 SAMSK.EN 位="0"时允许修改该寄存器。

35. SCI7816 主控器

本章介绍了 SCI7816 主控器的功能和使用方式。

35.1 简介

SCI7816 为异步半双工串行通讯端口，符合 SCI/IEC 7816-3 标准，功能和特性如下：

- 支持 T=0 和 T=1 的传输模式；
- 收发自动转换；
- 支持重试次数设置；
- 支持额外的基本时间单位(ETU, elementary time unit)设置；
- 支持 ETU 计数器计数；
- 支持硬件 LRC/CRC(纵向冗余校验/循环冗余校验)计算；
- 起始位采用 16 次采样判决算法；
- 每一个 Bit 数据三次采样并遵从择多判决算法；
- 支持 SCI7816 接口软复位；
- 支持接收数据未取走情况下，被新数据覆盖（overrun）标志；
- 支持 SCI7816 接口状态（忙 busy/空闲 idle）标志。

SCI7816 接口有以下中断源：

- 发送/接收数据完成（包括成功和失败）；
- ETU 计数器中断。

35.2 ETU 计数器的使用

SCI7816 提供 ETU 计数器的功能，软件可以根据需要设置 ETU 计数器，产生中断后做相应的处理。

使用 ETU 计数器的情况有以下几种：

- 计算字符超时等待时间(CWT)
- 计算块保护等待时间(BGT)
- 计算额外的保护时间（额外 ETU）

CWT 应用：软件按需要设置 ETU 计数器为 CWT 的值，如果接收两个字节数据之间间隔超过 CWT，则会产生 ETU 计数器中断，软件可以在 ETU 计数器中断服务函数中做相应的超时处理。

BGT 应用：T=1 模式下，软件按需要设置 ETU 计数器为 BGT 的值，在接收到一个包数据后，必须等 ETU 计数器中断产生后，才可以发送数据包，用以满足块保护时间的要求。

额外的保护时间应用：SCI7816 提供 0/1/2/4 四种额外 ETU 的可选配置，但如果有其他需要，可以通过设置 ETU 计数器并等待中断的方法，满足特殊的额外 ETU 需求。

35.3 SCI7816 通信速率设置

SCI7816 模块通信速率由 SBDR 寄存器控制。该寄存器的值为一个 ETU 所对应的外部时钟周期数。具体寄存器设置与 7816 模块通信速率的对应关系如下表所示：

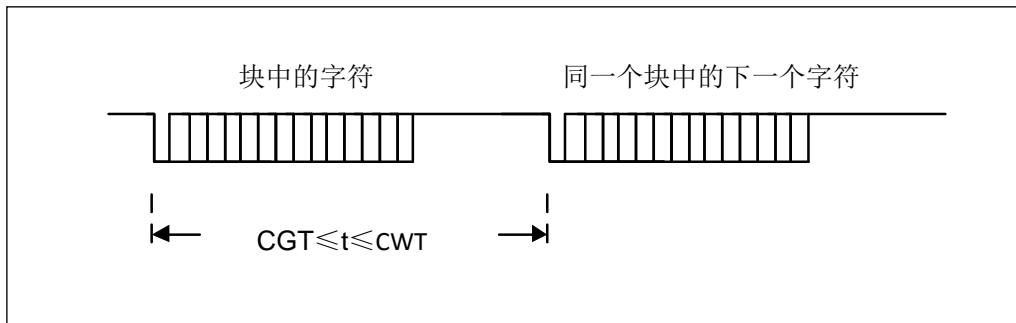
表 35.3-1 SCI7816 通信速率设置

Fi/Di	SBDR[13:0]	波特率(3.57MHz)	波特率(5MHz)
372	0x0174	9.6bps	13.4kbps
64	0x0040	57.6kbps	80.7kbps
32	0x001F	115.2kbps	161.3kbps
16	0x0010	230.4kbps	320.6kbps

35.4 SCI7816 字符等待时间 (CWT)

字符等待时间 CWT 表示 SCI7816 相同传输方向上连续两个字符间起始沿的最大延迟。

图 35.4-1 7816 字符等待时间



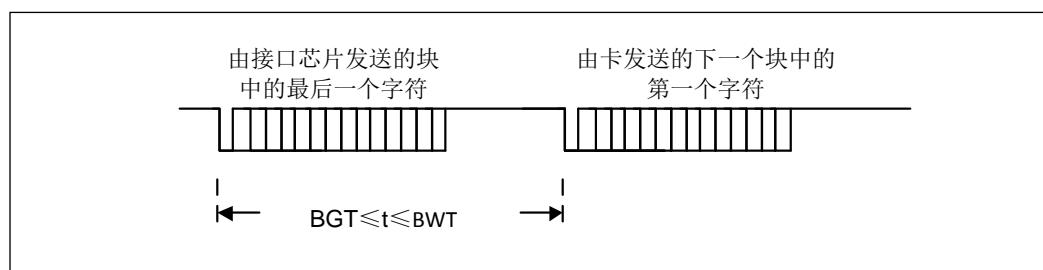
在 $T = 0$ 模式下, 字符等待时间 $CWT = WI * 960 * \frac{F}{f}$ ($WI=0\sim255$, 默认为 10), 该默认值为 9600 ETU;

在 $T = 1$ 模式下, 字符等待时间 $CWT = 11etu + 2^{CWI}$ ($CWI=0\sim15$, 默认为 13), 该默认值为 8203 ETU;

35.5 SCI7816 块保护时间 (BGT)

在 $T=1$ 模式下, 块保护时间 BGT 表示, 两个反向字符第一个沿之间的最小延迟。如下图所示:

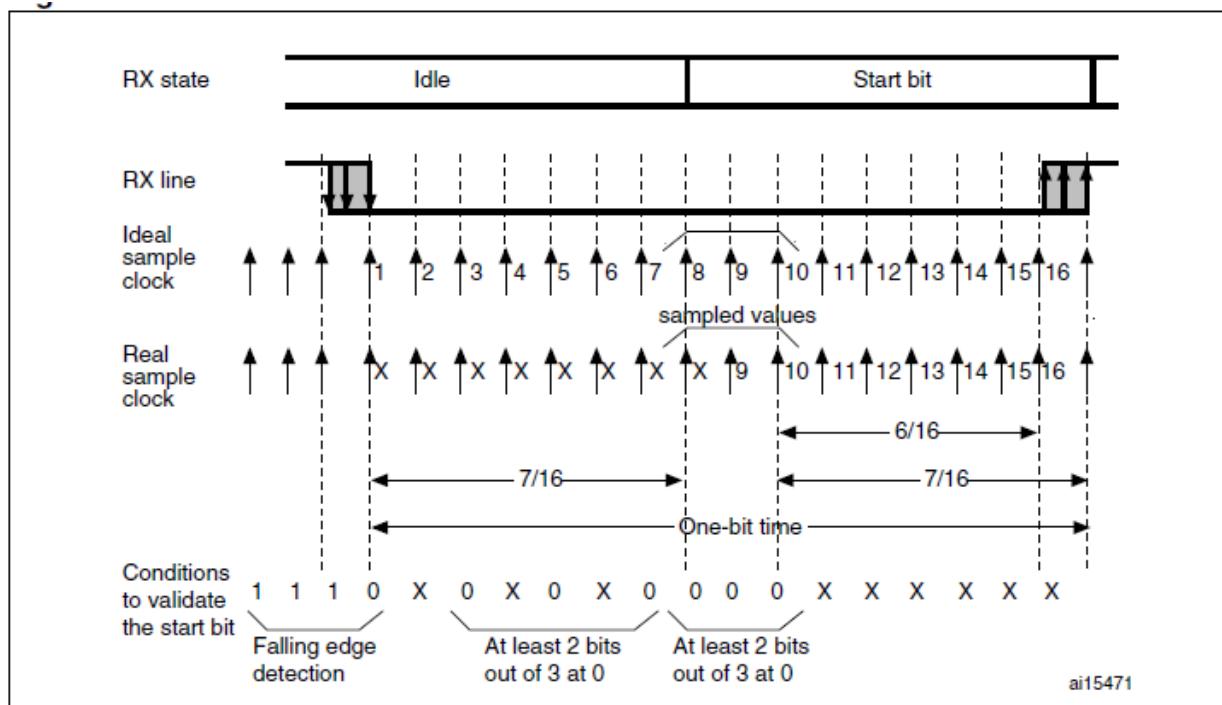
图 35.5-1 SCI7816 块保护时间



35.6 起始位采样

起始位采样采用 16 次采样，采样图形如下：

图 35.6-1 采样判决方式



采样判决方式说明如下：

- 1) 两组采样点中，全部 6 个采样点均为 0（第 3、5、7 采样点处为 0，第 8、9、10 采样点处为 0），则起始位采样有效，不置起 STATUS.RX_NE 标志。
- 2) 对两组采样点中，两组里的 3 个采样点至少有 2 个为 0（第 3、5、7 采样点和第 8、9、10 采样点），起始位采样有效，并置起 STATUS.RX_NE 标志（第一种 6 个采样点全为 0 的情况除外）。
- 3) 如果以上采样条件不满足，退出起始位采样，并使 7816 接口回到等待状态。

35.7 软件操作流程

35.7.1 初始化配置

1. 清除 STATUS 寄存器里的中断标志;
2. 设置 MODE 寄存器;
3. 配置中断优先级，通过 AIE.7816E 使能 7816 中断;
4. 设置 CON 寄存器，使能 SCI7816 模块，及配置其他功能;
5. 配置 SBDRL 寄存器;
6. 配置 ETU_CNT 寄存器。

35.7.2 发送（中断方式）

1. 初始化配置;
2. 将待发送数据写入 DATA 寄存器中;
3. 每向 DATA 寄存器写一次数据，查询一次 FIFO_STATUS 寄存器的 TX_FUL 标志，如果没有置位，继续写入，如果置位，停止写入;
4. 当 FIFO 中的数据全部发送成功，则会产生中断，软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功;
5. 数据发送失败（超过重试设置），也会产生中断，软件应首先读取 STATUS，判断中断是否是 TX_ERR，之后读取 FIFO_DEPTH，计算还有几个数没发出去，然后清空 FIFO，之后清除中断标志，并读取中断标志寄存器确认中断清除成功;
6. 如继续发送，则重复步骤 2-5;
7. 数据发送过程结束后，硬件自动把 7816 模块设成硬件接收模式，不需要软件设置。

35.7.3 发送（轮询方式）

1. 初始化设置；
2. 将待发送数据写入 DATA 寄存器中；
3. 每向 DATA 寄存器写一次数据，查询一次 FIFO_STATUS 寄存器的 TX_FUL 标志，如果没有置位，继续写入，如果置位，停止写入；
4. 软件查询 STATUS 寄存器标志，若 FIFO 中数据全部传输成功则硬件会置 TX_FIN 标志，程序查询到标志置起后，需把相应标志清 0，并读取 STATUS 标志寄存器确认清除成功；
5. 软件查询 STATUS 寄存器标志，如果查询到错误标志，软件应首先读取 STATUS 寄存器，判断中断是否是 TX_ERR，之后读取 FIFO_DEPTH，计算还有几个数没发出去，然后清空 FIFO，之后清除中断标志，并读取中断标志寄存器确认中断清除成功；
6. 如继续发送，重复步骤 2-5；
7. 数据发送过程结束后，硬件自动把 7816 模块设成硬件接收模式，不需要软件设置。

35.7.4 接收（中断方式）

1. 初始化配置；
2. 当接收了一个字节数据成功或失败之后，会产生中断请求，若数据接收成功则硬件会置 RX_FIN 标志，若失败则置 RX_ERR 标志；软件应首先清除中断标志，并读取中断标志寄存器确认中断清除成功；
3. 若接收成功，则软件应从 DATA 寄存器中读取收到的数据，不断查询 FIFO_STATUS 寄存器的 RX_EPTB 标志，直到 FIFO 为空；若接收失败，则进行出错处理；
4. 继续接收重复步骤 2-3。

35.7.5 接收（轮询方式）

1. 初始化设置；
2. 软件查询 STATUS 标志寄存器，当接收了一个字节数据成功或失败之后，会产生相应标志位，若数据接收成功则硬件会置 RX_FIN 标志，若失败则置 RX_ERR 标志；软件读取相应标志位后需将相应标志清 0，并读取 STATUS 标志寄存器确认清除成功；
3. 若接收成功，则软件应从 DATA 寄存器中读取收到的数据，不断查询 FIFO_STATUS 寄存器的 RX_EPTB 标志，直到 FIFO 为空；若接收失败，则进行出错处理；
4. 继续接收重复步骤 2-3。

35.8 寄存器定义

SCI7816 寄存器共 13 个，列表如下：

表 35.8-1 SCI7816 寄存器组

寄存器名称	寄存器描述	参考章节
CON	SCI7816 控制寄存器	35.8.1
MODE	SCI7816 模式配置寄存器	35.8.2
STATUS	SCI7816 状态寄存器	35.8.3
FIFO_STATUS	SCI7816 FIFO状态寄存器	35.8.4
FIFO_DEPTH	SCI7816 FIFO深度寄存器	35.8.5
DATA	SCI7816 收发数据寄存器	35.8.6
EDC_DATA	SCI7816 校验数据寄存器	35.8.7
SBDR	SCI7816 波特率配置寄存器	35.8.8
ETU_CNT	SCI7816 ETU 计数寄存器	35.8.9
F1_NUM	SCI7816强制高电平配置寄存器	35.8.10
CLK_RST	SCI7816 时钟复位寄存器	35.8.11
INT_EN	SCI7816 主机中断使能寄存器	35.8.12
WAKEUP_INT	SCI7816 唤醒中断标志寄存器	35.8.13

35.8.1 SCI7816 控制寄存器(CON)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	TX_FIF O_CLR	RX_FIF O_CLR	ETU_C NT_EN	EDC _EN	RETR Y_EN	PARITY_C HECK_EN	F1_EN	EN
类型	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	1	1	0	0	0	1

寄存器功能

[bit7] TX_FIFO_CLR:发送 FIFO 清零位

Bit7		描述
写	0	无效, 写1之后TX_FIFO_CLR位自动清零;
	1	软件置位后, 清空SCI7816 发送缓存中的数据, 发送FIFO中的所有状态标志位恢复默认值

[bit6] RX_FIFO_CLR:接收 FIFO 清零位

Bit6		描述
写	0	无效, 写1之后RX_FIFO_CLR位自动清零;
	1	软件置位后, 清空SCI7816 接收缓存中的数据, 接收FIFO中的所有状态标志位恢复默认值

[bit5] ETU_CNT_EN: ETU 计数器功能使能位

Bit5		描述
写	0	ETU计数器功能关闭;
	1	ETU计数器功能使能

注意:

- 在 ETU 功能打开后, 以收发数据的下降沿为计时开始点, 以 ETU 为单位计数, 若达到设置的 ETU 计数值, 则产生标志位。该功能用以辅助 7816 协议的超时处理, 可以代替定时器计时。

[bit4] EDC_EN: EDC 功能使能位

Bit4		描述
写	0	EDC(error detection code)功能无效
	1	EDC(error detection code)功能有效

注意:

- 仅在 T=1 模式下有效。在该功能打开后, 在接收到校验正确的数据, 或者发送数据时, 硬件会自动计算 EDC 的值并保存在 EDC_DATA 寄存器中。

[bit3] RETRY_EN: 自动重发功能使能位

Bit3		描述

写	0	自动重收发功能无效
	1	自动重收发功能有效

[bit2] PARITY_CHECK_EN: 校验检查使能位

Bit2		描述
写	0	校验位检查功能无效
	1	校验位检查功能有效

[bit1] F1_EN: SCI7816 强制输出高电平功能控制信号位

Bit1		描述
写	0	强制输出高电平功能无效
	1	强制输出高电平功能有效

注意：

- 打开此功能后，接收数据时，在奇偶校验错时，请求重发的一个 ETU 的低电平之前和之后，SCI7816 硬件强制输出一个时钟周期的高电平；发送数据时，在起始位开始前和数据位之后，SCI7816 硬件强制输出一个时钟周期的高电平。在该功能不打开的情况下，IO PAD 在输入状态时，只能靠上拉保持高电平，因此从电平转换的上升时间或下降时间有可能超出规范（具体数据依赖于系统设计方案）；打开强制输出高电平功能，可以在电平转换时先将 IO PAD 电平强制拉高，之后切换为上拉控制，这样可以改善电平转换时间。

[bit0] EN: SCI7816 模块使能控制信号位

Bit0		描述
写	0	SCI7816硬件关闭
	1	SCI7816硬件使能

注意：EN 清零后，所有状态位恢复默认值。

注意：

- 如出现 FIFO 异常，需要使用 TX/RX_FIFO_CLR 功能清空 FIFO，则软件在置位跟接收或发送对应的 TX/RX_FIFO_CLR 控制位后，需要查询等待该位自动清零后，FIFO 方可正常使用。

RETRY_EN	PARITY_CHECK_EN	功能
0	0	不做校验检查，不重试，不会产生错误标志
0	1	做校验检查，不重试，如校验出错则产生错误标志
1	0	-
1	1	做校验检查，出错重试，如重试次数超出设置，则产生错误标志

在 PARITY_CHECK_EN=1 且 RETRY_EN=1 时，在接收数据时，若接收到奇偶校验错误的字节（且未超过重试次数设置），将自动在 10.5-11.5 ETU 处拉低 IO 一个 ETU 的时间，用来请求重发，且重试次数加 1；若超过重试次数设置，则硬件不再请求重发，并产生接收错误标志

/中断。

在 PARITY_CHECK_EN=1 且 RETRY_EN=1 时，在发送数据时，若在 11ETU 处采样到对方的重发请求（且未超过重试次数设置），则硬件自动重发之前发送的最后一个字节数据，且重试次数加 1；若超过重试次数设置，则硬件不再重发，并产生发送错误标志/中断。

在 PARITY_CHECK_EN=1 且 RETRY_EN=0 时，在接收数据时，若接收到奇偶校验错误的字节，硬件不请求重发，并产生接收错误标志/中断。

在 PARITY_CHECK_EN=1 且 RETRY_EN=0 时，在发送数据时，若在 11ETU 处采样到对方的重发请求，硬件不重发，并产生发送错误标志/中断。

在 PARITY_CHECK_EN=0 且 RETRY_EN=0 时，在接收数据时，不校验接收的数据，硬件不请求重发，不产生接收错误标志/中断。

在 PARITY_CHECK_EN=0 且 RETRY_EN=0 时，在发送数据时，若在 11ETU 处采样到对方的重发请求，硬件不重发，不产生发送错误标志/中断。

35.8.2 SCI7816 模式配置寄存器(MODE)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	RETRY_TIMES			EDC_MODE	--	CDCV_T	PARIT_Y	T_MOD_E
类型	R/W	R/W	R/W	R/W	--	R/W	R/W	R/W
初始值	0	0	0	0	--	0	0	0

寄存器功能

[bit7:5] RETRY_TIMES[2: 0]: 自动重收发次数设定位

本位在 PARITY_CHECK_EN=1 且 RETRY_EN=1 时有效。

Bit7:5		描述
写	000	允许无限次自动重收发，出错标志不会置位；
	001	允许一次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位；
	010	允许二次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位；

	111	允许七次自动重收发，如仍有奇偶校验错则停止收发，出错标志被置位；

[bit4] EDC_MODE: EDC 校验模式位

Bit4		描述
写	0	LRC校验
	1	CRC校验

注意：

- 仅 T=1 模式下有效。LRC 校验为 1 字节，存在 EDC_DATA[15:8]寄存器中；CRC 校验为 2 字节，高字节存在 EDC_DATA[7:0]中，低字节存在 EDC_DATA[15:8]中。CRC16 校验算法参照 SCI/IEC 13239，算法多项式为 $x^{16}+x^{12}+x^5+1$ ，初始值为全 FFFF，输出取反。

[bit3] 保留位,写操作无效，读数据无意义。

[bit2] CDCVT: 传输方向选择位

Bit2		描述
写	0	正向传输
	1	反向传输

[bit1] PARITY: 奇偶校验选择位

Bit1		描述
写	0	偶校验
	1	奇校验

[bit0] T_MODE: 传输模式选择位

Bit0		描述
写	0	T=0模式
	1	T=1模式

注意：

- T=1 模式下，PARITY_CHECK 和 RETRY 功能不可用，收发数据字符间隔默认为 11ETU (T=0 为 12ETU)。

35.8.3 SCI7816 状态寄存器(STATUS)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	RX_NE	RX_O_VR	STAT	ETU_CNT_FIN	TX_E_RR	TX_FI_N	RX_E_RR	RX_FIN
类型	R	R	R	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] RX_NE: 接收起始位噪声标志位

接收数据开始后，起始位采样有效，但 6 个采样点并非全部为 0，则会置位，可做查询使用。

在下一次接收数据的起始位后，如果起始位无效，或者 6 个采样点全部为 0（无噪声），则该标志自动清零。

[bit6] RX_OVR: 接收 buffer 覆盖指示位

接收数据时，若没有及时取走 buffer 中的数据，在正确接到新的数据时，在接收起始位下降沿开始 11.5 个 ETU 时自动置位，表明有数据被覆盖；置位 RX_FIFO_CLR 可将此位清零。

Bit6		描述
读	0	SCI7816 接收 buffer 没有被覆盖
	1	SCI7816 接收 buffer 被覆盖了

[bit5] STAT: SCI7816 工作状态位

在接收数据时，从接收起始位下降沿 12 个 ETU 内一直为 1，12 个 ETU 后自动清零，从接收新的字节开始，再次被自动置位。

在发送数据时，从发送起始位下降沿 12 个 ETU 内一直为 1，12 个 ETU 后自动清零，直到发送新的字节开始，再次被自动置位。

Bit5		描述
读	0	空闲(idle)
	1	忙(busy)

[bit4] ETU_CNT_FIN: ETU 计数器中断标志位

以收发数据的下降沿为计时开始点，以 ETU 为单位计数，若达到设置的 ETU 计数值，则置位该标志位。置位后如果有新的数据收发，则标志位被清零并重新开始计数。

Bit4		描述
写	0	ETU计数器中断标志位清零
	1	无效

[bit3] TX_ERR: SCI7816 数据发送失败的中断标志位

T=0 模式下，因奇偶校验错误，发送重试次数超过设置的次数，则此标志在最后一次发送起始位下降沿后 11.5 ETU 时被置位。

T=1 模式下，该标志位无效。

Bit3		描述
写	0	SCI7816数据发送失败的中断标志位清零
	1	无效

[bit2] TX_FIN: SCI7816 数据发送成功的中断标志位

在成功发送一个字节数据，起始位下降沿后 11.5 ETU (T=0) / 10.5 ETU (T=1) 时被置位

Bit2		描述
写	0	SCI7816数据发送成功的中断标志位清零
	1	无效

[bit1] RX_ERR: SCI7816 数据接收失败的中断标志位

T=0 模式下，因奇偶校验错误，接收重试次数超过设置的次数，则此标志在最后一次接收起始位下降沿后 11.5 ETU 时被置位。

T=1 模式下，该标志位无效

Bit1		描述
写	0	SCI7816数据接收失败的中断标志位清零
	1	无效

[bit0] RX_FIN: SCI7816 数据接收成功的中断标志位

在接收成功一个字节数据，起始位下降沿后 11.5 ETU (T=0) 或 10.5 ETU (T=1) 时被置位。

Bit0		描述
写	0	SCI7816数据接收成功的中断标志位清零
	1	无效

35.8.4 SCI7816 FIFO 状态寄存器(FIFO_STATUS)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	TX_EP T	TX_U DR	TX_F UL	TX_EP TB	RX_O VR	RX_U DR	RX_F UL	RX_EP TB
类型	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit7] TX_EPT: 发送 FIFO 满溢标志位

发送 FIFO 已满时，再写入 DATA 寄存器数据（属于非法操作），可以将该位置位，此时发送 FIFO 中已有数据被覆盖；置位 TX_FIFO_CLR 可将此位清零。

Bit7	描述	
读	0	SCI7816 发送FIFO没有overrun
	1	SCI7816 发送FIFO发生overrun

[bit6] TX_UDR: 发送 FIFO 为空时读取指示位

发送 FIFO 为空时，读取 FIFO 中的数据（属于非法操作），可以将该位置位；置位 TX_FIFO_CLR 可将此位清零。

Bit6	描述	
读	0	SCI7816 发送FIFO没有underrun
	1	SCI7816 发送FIFO发生underrun

[bit5] TX_FUL: 发送 FIFO 为满指示位

发送 FIFO 差一个字节数据未满时，再写入 DATA 寄存器一个字节数据，可以将该位置位；在成功发送一个字节数据，发送起始位下降沿开始 11.5 个 ETU 时自动清零，或者置位 TX_FIFO_CLR 可将此位清零。

Bit5	描述	
读	0	SCI7816 发送FIFO非满
	1	SCI7816 发送FIFO为满

[bit4] TX_EPTB: 发送 FIFO 非空指示位

发送 FIFO 为空时，写入 DATA 寄存器待发送的数据，可以将该位置位；正确发送完 FIFO 中的所有数据，在成功发送最后一个字节数据，起始位下降沿开始 11.5 个 ETU 时自动清零，或者置位 TX_FIFO_CLR 可将此位清零。

Bit4	描述	
读	0	SCI7816 发送FIFO为空
	1	SCI7816 发送FIFO非空

[bit3] RX_OVR: 接收 FIFO 满溢标志位

接收 FIFO 已满时，再正确接到一个字节数据，在接收起始位下降沿开始 11.5 个 ETU 时自动置位，此时接收 FIFO 中已有数据被覆盖；置位 RX_FIFO_CLR 可将此位清零。

Bit3		描述
读	0	SCI7816 接收FIFO没有overrun
	1	SCI7816 接收FIFO发生overrun

[bit2] RX_UDR: SCI7816 接收 FIFO 为空时读取指示位

接收 FIFO 为空时，读取 FIFO 中的数据(属于非法操作)，可以将该位置位；置位 RX_FIFO_CLR 可将此位清零。

Bit2		描述
读	0	SCI7816 接收FIFO没有underrun
	1	SCI7816 接收FIFO发生underrun

[bit1] RX_FUL: 接收 FIFO 为满指示位

接收 FIFO 差一个字节数据未满时，再正确接到一个字节数据，在接收起始位下降沿开始 11.5 个 ETU 时自动置位；从接收 FIFO 中读走一个字节数据，或者置位 RX_FIFO_CLR 可将此位清零。

Bit1		描述
读	0	SCI7816 接收FIFO非满
	1	SCI7816 接收FIFO为满

[bit0] RX_EPTB: 接收 FIFO 非空指示位

接收 FIFO 为空时，正确接到一个字节数据，从起始位下降沿开始 11.5 个 ETU 时自动置位；将接收 FIFO 中数据全部读走，或者置位 RX_FIFO_CLR 可将此位清零。

Bit0		描述
读	0	SCI7816 接收FIFO为空
	1	SCI7816 接收FIFO非空

注意：

对 DATA 寄存器进行读写可能影响 FIFO_STATUS 所有状态位，请注意以下操作条件：

- 在 RX_EPTB=0 时，不可以读取 DATA 寄存器；
- 在 RX_FUL=1 时，需要立刻读取 DATA 寄存器中的数据；
- 在 TX_FUL=1 时，不可以写入 DATA 寄存器；
- 在 RX_UDR/RX_OVR/TR_UDR/TX_OVR 任何一位被置位后，软件需要立刻置位 RX_FIFO_CLR 或 TX_FIFO_CLR，清空 FIFO 将所有状态位恢复默认值。

对 DATA 寄存器进行读写，影响 FIFO_STATUS 状态位有一定延时（硬件需要做跨时钟同步），软件需要限制单次操作 DATA 寄存器的次数，不可超过 FIFO 本身的深度，具体操作如下：

- 查询 RX_EPTB 从 0 变 1 时，只可以读出一个字节数据；
- 查询 RX_EPTB=1 时，可以通过查询 RX_FIFO_DEPTH 得到接收 FIFO 中存储的字节数，再全部读出；
- 查询 RX_FUL=1 时，最多读出 X 字节数据（X=FIFO 大小）；
- 查询 TX_EPTB=0 时，最多可以写入 X 字节数据（X=FIFO 大小）；
- 查询 TX_FUL=0 时，可以通过查询 TX_FIFO_DEPTH 得到发送 FIFO 中存储的字节数，继续写入 N 字节数据（N=X-TX_FIFO_DEPTH）。

35.8.5 SCI7816 FIFO 深度寄存器(FIFO_DEPTH)

寄存器配置

位	[7:4]	[3:0]
字段名	TX_FIFO_DEPTH	RX_FIFO_DEPTH
类型	R	R
初始值	4'b0000	4'b0000

寄存器功能

[bit7:4] TX_FIFO_DEPTH [3: 0]: 发送 FIFO 深度位

本位表示发送 FIFO 中存储的待发送数据字节数。

[bit3:0] RX_FIFO_DEPTH [3: 0]: 接收 FIFO 深度位

本位表示接收 FIFO 中存储的待发送数据字节数。

35.8.6 SCI7816 收发数据寄存器(DATA)

寄存器配置

位	[7:0]
字段名	DATA
类型	R/W
初始值	8'b0000_0000

寄存器功能

[bit7:0] DATA [7: 0]: 数据缓存寄存器

本位表示存放待发送的数据字节，以及接收到的数据；二者地址映射在一起。

Bit7:0	描述
读	已接收到的数据字节
写	待发送的数据字节

注意：

- 本寄存器用于接收和发送；
- 在向发送缓冲器中写数据前必须确认发送缓冲器不满(FIFO_STATUS.TX_FUL =0)，否则将不能把数据写入发送缓冲器中。从接收缓冲器中读出数据前应确认接收缓冲器非空(FIFO_STATUS.RX_EPTB=1)，否则可能读出不可预知的数据。

35.8.7 SCI7816 校验数据寄存器(EDC_DATA)

寄存器配置

位	[15:0]
字段名	EDC_DATA
类型	R/W
初始值	16'b0000_0000_0000_0000

寄存器功能

[bit15:0] EDC_DATA [15: 0]: 校验数据寄存器

T=1 模式下，存储接收到或待发送的 CRC 校验值或者 LRC 校验值。

操作流程如下：

发送时写入 DATA 寄存器的数据，硬件会自动进行计算，并将计算结果随时更新至 EDC_DATA 寄存器中。软件可以将整个块数据(除 EDC 字节外)全部写入 DATA 寄存器，再读取 EDC_DATA 寄存器，将运算结果写入 DATA 寄存器，此时 EDC_DATA 寄存器将自动恢复默认值 0；

接收时所有接收到的数据，硬件会自动进行计算，并将计算结果随时更新至 EDC_DATA 寄存器中。如果所有数据均接收正确，在整个块数据接收完成后，EDC_DATA 寄存器最终计算结果应该为 0。软件可以在完成一个块数据接收后查询 EDC_DATA 来校验数据，也可以在接收过程中查询 EDC_DATA 的值并与软件计算结果进行对比。

35.8.8 SCI7816 波特率配置寄存器(SBDR)

寄存器配置

位	[15:14]	[13:0]
字段名	ETU_SEL	SBDR
类型	R/W	R/W
初始值	2'b00	14'b00_0000_0000_0000

寄存器功能

[bit15:14] ETU_SEL [15:14]: SCI7816 发送模式增加 extra ETU 配置位

无 extra ETU 情况下, T=0 默认为 12ETU, T=1 默认为 11ETU。

Bit15:14		描述
写	00	无extra ETU
	01	增加一个extra ETU
	10	增加二个extra ETU
	11	增加四个extra ETU

[bit13:0] SBDR [13:0]:波特率配置位

SBDR = Fi/D, 默认值 0x174, 表示 1 个 ETU 单位为 372 CLK。若 SCI7816 输入时钟不是 3.57MHz, 需要根据频率比例计算相应的 SBDR 值。

35.8.9 SCI7816 ETU 计数寄存器(ETU_CNT)

寄存器配置

位	[15:0]
字段名	ETU_CNT
类型	R/W
初始值	16'd9600

寄存器功能

[bit15:0] ETU_CNT [15:0]: SCI7816 ETU 计数寄存器

计算单位为 1 个 ETU 时间, 默认值 0x2580, 表示 9600 ETU

注意事项:

当 ETU 计数器使能开启 (CON.ETU_CNT_EN =1) 时, 接收数据从收发数据的有效下降沿起始, ETU 计数器以 ETU 为单位计数开始计数, 当计数器的值达到设置的 ETU 计数值 (ETU_CNT[15:0]) 时, 则在 STATUS 寄存器中产生标志位, 并且产生 ETU_CNT_INT 中断输出。当下一次收发数据的有效下降沿到来时, 相应的标志位和中断会被自动清除。软件也可以通过对 STATUS.ETU_CNT_FIN 写 0 将中断清除。

35.8.10 SCI7816 强制高电平配置寄存器 0(F1_NUM)

寄存器配置

位	--	2:0
字段名	--	F1_NUM
类型	--	R/W
初始值	--	3'b000

寄存器功能

[bit7: 3] 保留位,写操作无效, 读数据无意义。

[bit2:0] F1_NUM [2:0]: SCI7816 强制高电平的时钟数配置位

在控制寄存器中打开强制输出高电平功能后, 该寄存器用来控制硬件强制高电平的时钟数。

Bit2:0		描述
写	000	强制输出1个时钟周期高电平
	001	强制输出2个时钟周期高电平
	010	强制输出3个时钟周期高电平
	011	强制输出4个时钟周期高电平
	100	强制输出5个时钟周期高电平
	101	强制输出6个时钟周期高电平
	110	强制输出7个时钟周期高电平
	111	强制输出8个时钟周期高电平

35.8.11 SCI7816 时钟复位寄存器(CLK_RST)

寄存器配置

位	7	6	5	4	3	2	1	0
字段名	--	--	--	CLK_SEL	CLK_EN	RESET	RESET_EN	
类型	--	--	--	R/W	R/W	R/W	R/W	
初始值	--	--	--	2'b00	0	0	0	

寄存器功能

[bit7: 5] 保留位, 写操作无效, 读数据无意义

[bit4: 3] CLK_SEL: 7816 主机输出时钟频率选择位

Bit4:3	描述	
写	00	系统时钟
	01	1/2 系统时钟
	10	1/4 系统时钟
	11	1/8 系统时钟

[bit2] CLK_EN: SCI7816 主机输出时钟驱动使能位

Bit2	描述	
写	0	SCI7816 输出时钟关闭
	1	SCI7816 输出时钟使能

[bit1] RESET: SCI7816 主机输出复位信号

Bit1	描述	
写	0	SCI7816 主机输出复位信号为低电平, 从机复位
	1	SCI7816 主机输出复位信号为高电平, 从机工作

[bit0] RESET_EN: SCI7816 主机输出复位使能

Bit0	描述	
写	0	SCI7816 复位输出使能关闭
	1	SCI7816 复位输出使能

35.8.12 SCI7816 中断使能寄存器(INT_EN)

寄存器配置

位	7:3	2	1	0
字段名	--	INT_EN	ETU_CNT_IN_T_EN	WAKEUP_INT_EN
类型	--	R/W	R/W	R/W
初始值	--	0	0	0

寄存器功能

[bit7: 3] 保留位，写操作无效，读数据无意义

[bit2] INT_EN: SCI7816 发送/接收中断使能位

Bit2		描述
写	0	MCU不接受发送/接收中断
	1	MCU接受发送/接收中断

[bit1] ETU_CNT_INT_EN: SCI7816ETU 计数器中断使能

Bit1		描述
写	0	MCU不接受7816计数器中断
	1	MCU接受7816计数器中断

[bit0] WAKEUP_INT_EN: SCI7816 唤醒中断使能

Bit0		描述
写	0	MCU不接受7816唤醒中断
	1	MCU接受7816唤醒中断

35.8.13 SCI7816 唤醒中断标志寄存器(WAKEUP_INT)

寄存器配置

位	7:1	0
字段名	--	WAKEUP_INT
类型	--	R/W
初始值	--	0

寄存器功能

[bit7: 1] 保留位，写操作无效，读数据无意义

[bit0] WAKEUP_INT: SCI7816 唤醒中断标志位

当接收到一个有效的起始位下降沿时，该位被置位。

Bit0	描述	
写	0	SCI7816唤醒中断标志位清零
	1	无效

附录 1 寄存器映射

寄存器映射表按照模块或功能排列。

[如何阅读该表]

模块/功能名称及其基地址

时钟/复位 基地址 : 0x4001_0000

Base_Address + Address	寄存器			
	+3	+2	+1	+0
0x000	-	-	-	CTRL[B,H,W] 000-_0-01
0x004	-	-	-	CLK_STB[B,H,W] 000-_0-01
0x008	LPM_CTL[B,H,W] 0000_0000 0000_0000 ----_----_0_-000			
0x00C	-	-	-	STAT[B,H,W] ----_0 0000_--01

- : 保留区

寄存器复位后的初始值

"1"	:	初始值为 "1"
"0"	:	初始值为 "0"
"X"	:	初始值不定
"_"	:	保留位

寄存器名称

访问单位
(B : byte, H : half word, W : word)

最右边的寄存器地址(当 word 访问时, 寄存器"+0"列的称为数据的 LSB 侧)

注意:

- 寄存器列表用小端 (little-endian) 表示。
- 当进行数据访问时, 根据访问数据的大小, 其地址如下:
 - Word 访问 : 地址是 4 的倍数 (最低 2 位是"0")
 - Half word 访问 : 地址是 2 的倍数 (最低位是"0")
 - Byte 访问 : -
- 不可访问该列表中未标注的地址空间。

Flash I/F

Base_Address: 0x4000_0000

Base_Address	Register			
+ Address	3	2	1	0
0x000	FOCR[W]			
	0000_0000 0000_0000 0000_0000 0000_0000			
0x004	FLCR0[W]			
	1111_1111 1111_1111 1111_1111 1111_1111			
0x008	FLCR1[W]			
	1111_1111 1111_1111 1111_1111 1111_1111			
0x00C	FSTR[W]			
	0000_0000 0000_0000 0000_0000 0000_0000			
0x010	FWCICR[W]			
	0000_0000 0000_0000 0000_0000 0000_0000			
0x014	-			
	-			
0x018	-			
	-			
0x01C	-			
	-			
0x020	-			
	-			
0x024	-	-	-	TNVS[B,H,W]
	-	-	-	0001_1000
0x028	-	-	-	TPROG[B,H,W]
	-	-	-	0001_1000
0x02C	-	-	-	TPGS[B,H,W]
	-	-	-	0001_1000
0x030	-	-	-	TRCVPGM[B,H,W]
	-	-	-	0001_1000

Base_Address	Register			
+ Address	3	2	1	0
0x035	-	-	-	TRCVPERS[B,H,W]
	-	-	-	0001_1001
0x038	-	-	-	TRCVCERS[B,H,W]
	-	-	-	0000_0111
0x03C	-	-	-	TPERS[B,H,W]
	-	-	-	0001_0000
0x040	-	-	-	TCERS[B,H,W]
	-	-	-	0010_0000

Unique ID

Base_Address: 0x4000_0100

Base_Address	Register			
+ Address	3	2	1	0
0x000	UIDR0[B,H,W]			
	XXXX_XXXX XXXX_XXXX XXXX_XXXX XXXX_XXXX			
0x004	UIDR1[B,H,W]			
	XXXX_XXXX XXXX_XXXX XXXX_XXXX XXXX_XXXX			
0x008	-		UIDR2[B,H,W]	
	-		XXXX_XXXX XXXX_XXXX	

Clock/Reset

Base_Address: 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CTRL[W] 000-_0-01
0x004	-	-	-	CLK_STB[W] 000-_0-01
0x008	LPM_CTL[W] 0000_0000 0000_0000 ----_----_0_-000			
0x00C	-	-	STAT[W] ----_---0 0000_--01	
0x010	-	-	-	BC_DIV[W] ----_000
0x014	-	-	-	APB0_CTRL[W] ----_--00
0x018	-	-	-	APB1_CTRL[W] 1--0_--00
0x01C	-	-	-	-
0x020				SWDG_CTRL[W] -111_--00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	C SWT[W] 0000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x035	-	-	-	-
0x038	-	-	-	-
0x03C	-	-	-	-
0x040	-	-	FCM_CTL[W] -111--00 ----_1111	
0x044	-	-	-	FCM_STR[W] ----_--00
0x048	-	-	AFDWH_CTL[W] 1111_1111_1111_1111	
0x04C	-	-	AFDWL_CTL[W] 0000_0000_0000_0000	
0x050	-	-	AFDC_CTL[W] 0000_0000_0000_0000	
0x054	-	-	-	DBWDT_CTRL[W] 0-0-----
0x058 - 0x05C	-	-	-	-
0x060	-	-	-	INT_CTRL[W] --00_0-00
0x064	-	-	-	INT_PEND[W] --00_0-00
0x068	-	-	-	INT_CLR[W] --00_0-00
0x06C - 0xFFC	-	-	-	-

HWDG

Base_Address: 0x4001_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WDT_LD[W] 0000_0000 0000_0000 1111_1111 1111_1111			
0x004	WDT_VL[W] XXXX_XXXX XXXX_XXXX XXXX_XXXX XXXX_XXXX			
0x008	-	-	-	WDT_CTL[W] ----_011
0x00C	-	-	-	WDT_CLR[W] XXXXXXXX
0x010	-	-	-	WDT_STA[W] ----_---0
0x014	WDT_LCK[W] 0000_0000 0000_0000 0000_0000 0000_0001			
0x018 - 0xFFC	-			

SWDG

Base_Address: 0x4001_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WDT_LD[W] 1111_1111 1111_1111 1111_1111 1111_1111			
0x004	WDT_VL[W] 1111_1111 1111_1111 1111_1111 1111_1111			
0x008	-	-	-	WDT_CTL[W] ---0_0000
0x00C	WDT_INTCLR[W] XXXX_XXXX XXXX_XXXX XXXX_XXXX XXXX_XXXX			
0x010	-	-	-	WDT_STA[W] ----_---0
0x014 - 0x017	-	-	-	-
0x018	-	-	-	WDT_MDC[W] ----_---0
0x01C - 0xBFC	-	-	-	-
0xC00	WDT_LCK[W] 0000_0000 0000_0000 0000_0000 0000_0000			
0xC04 - 0xFFC	-			

Dual Timer

Base_Address: 0x4001_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	TIMER1LOAD[W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x004	TIMER1VALUE[W] 1111_1111 1111_1111 1111_1111 1111_1111			
0x008	TIMER1CTRL[W] ----_----_----_----_----_0010_0000			
0x00C	TIMER1INTCLR[W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0x010	TIMER1RIS[W] ----_----_----_----_----_----_----_0			
0x014	TIMER1MIS[W] ----_----_----_----_----_----_----_0			
0x018	TIMER1BGLoad[W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x020	TIMER2LOAD[W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x024	TIMER2VALUE[W] 1111_1111 1111_1111 1111_1111 1111_1111			
0x028	TIMER2CTRL[W] ----_----_----_----_----_----_0010_0000			
0x02C	TIMER2INTCLR[W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0x030	TIMER2RIS[W] ----_----_----_----_----_----_----_0			
0x035	TIMER2MIS[W] ----_----_----_----_----_----_----_0			
0x038	TIMER2BGLoad[W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x040 - 0xFFC	-			

Composite Timer

ch0	Base_Address: 0x4002_5000
ch1	Base_Address: 0x4002_5040
ch2	Base_Address: 0x4002_5080
ch3	Base_Address: 0x4002_50C0
ch4	Base_Address: 0x4002_5200
ch5	Base_Address: 0x4002_5240
ch6	Base_Address: 0x4002_5280
ch7	Base_Address: 0x4002_52C0

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	CSR/LWRR[H,W] xxxx_xxxx xxxx_xxxx	
0x004	-	-	DUT/HWRR/DBUF[H,W] xxxx_xxxx xxxx_xxxx	
0x008	-	-	TCNT[H,W] 0000_0000 0000_0000	
0x00C	-	-	TCR[B,H,W] -000_0000 -000_0000	
0x010	-	-	TCR2[B,H,W] ----_---0	STA[B,H,W] 0000_-000
0x014 - 0x03C				

Composite timer I/O select

Base_Address: 0x4002_5100

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	CTSEL0123[B,H,W]] 0000_0000	-
0x004 - 0x1FC	-	-	-	-
0x200	-	-	CTSEL4567[B,H,W]] 0000_0000	-
0x204 - 0xEF8	-	-	-	-
0xEFC	-	-	CTSSSR[B,H,W] ----_---- XXXX_XXXX	

12-bit ADC

Base_Address: 0x4002_7000

Base_Address	Register					
+ Address	3	2	1	0		
0x000	ADC_CTL1 [B,H,W] ----_----_--00 ----_0000 01--_0000					
0x004	-	-	-	ADC_CTL2[B,H,W] 100-_00-0		
0x008	-	-	-	ADC_CTL3[B,H,W] XXXX_XX--		
0x00C	-	-	ADC_Result [B,H,W] 0000_0000 0000_0000			
0x010	ADCMPIF [B,H,W] ----_----_----_----_----_0					
0x014	ADCMPPF [B,H,W] ----_----_----_----_----_0					
0x018	-	-	ADCMPCCTL [B,H,W] 000-_----_0010_00--			
0x01C	ADCMPPH [B,H,W] 0000_0000 00--_----		ADCMPL [B,H,W] 0000_0000 00--_----			
0x020—0xFFC	-					

OSC_CTL

Base_Address: 0x4002_E000

Address	+3	+2	+1	+0
0x000		HICR_TRMKEY[B,H,W] 00000000_00000000_00000000_00000001		
0x004		HICR_FTRM[B,H,W] 0000000_10111101_00000000_10111101		
0x008		-		
0x00C		-	HICR_FSEL[B,H,W] -----01	
0x010		-	HICR_FDIV[B,H,W] -----001	
0x014 - 0x01C		-		
0x020		HSXT_PWRCTL[B,H,W] -----_----00_-----_----00		
0x024 - 0x02C		-		
0x030		LSXT_PWRCTL[B,H,W] -----_----10_-----_----10		
0x035 - 0xFFC		-		

EXTI

Base_Address: 0x4003_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	INTEN[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x004	INTSRC[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0x008	INTCLR[B,H,W] 1111_1111 1111_1111 1111_1111 1111_1111			
0x00C	INTLVL[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x010	-			
0x014	-	-	NMISRC[B,H,W] ----_----_----_0	
0x018	-	-	NMICLR[B,H,W] ----_----_----_1	
0x01C - 0x0FC	-	-	-	-

IRQC

Base_Address: 0x4003_1000

Base_Address	Register			
+ Address	3	2	1	0
0x000 – 0x00C	-			
0x010	IRQ_ST02[B,H,W] ----_-----_-----_--00			
0x014	IRQ00PEND[B,H,W] ----_-----_-----_---000			
0x018	IRQ01PEND[B,H,W] ----_-----_-----_---0			
0x01C	IRQ02PEND[B,H,W] ----_-----_-----_---0			
0x020	-			
0x024	IRQ04PEND[B,H,W] ----_-----_-----_0000_0000			
0x028	IRQ05PEND[B,H,W] ----_-----_-----_0000_0000			
0x02C	IRQ06PEND[B,H,W] ----_-----_-----_--00			
0x030	IRQ07PEND[B,H,W] ----_-----_-----_---0			
0x035	IRQ08PEND[B,H,W] ----_-----_-----_--00			
0x038	IRQ09PEND[B,H,W] ----_-----_-----_---0			
0x03C	IRQ10PEND[B,H,W] ----_-----_-----_--00			

Base_Address	Register			
+ Address	3	2	1	0
0x040	IRQ11PEND[B,H,W] ----_-----_-----_---0			
0x044	IRQ12PEND[B,H,W] ----_-----_-----_---00			
0x048 - 0x04C	-			
0x050	IRQ15PEND[B,H,W] ----_-----_-----_---0			
0x054	IRQ16PEND[B,H,W] ----_-----_-----_---00			
0x058	IRQ17PEND[B,H,W] ----_-----_-----_---0			
0x05C	IRQ18PEND[B,H,W] ----_-----_-----_---00			
0x060	IRQ19PEND[B,H,W] ----_-----_-----_---0			
0x064	IRQ20PEND[B,H,W] ----_-----_-----_---00			
0x068 – 0x070	-			

Base_Address	Register			
+ Address	3	2	1	0
0x074	IRQ24PEND[B,H,W] ----_-----_-----_00_---00			
0x078	IRQ25PEND[B,H,W] ----_-----_-----_0_---0			
0x07C	IRQ26PEND[B,H,W] ----_-----_-----_0_---			
0x080	IRQ27PEND[B,H,W] ----_-----_-----_0_---			
0x084-0x08C	-			
0x090	IRQ31PEND[B,H,W] ----_0_-----_0000_0000_0000_0000			
0x094 – 0xFFC	-			

LCDC

Base_Address: 0x4003_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	LCDCR3[B,H,W]	LCDCR2[B,H,W]	LCDCR1[B,H,W]	LCDCR0[B,H,W]
	--00_---0	100_---00	0000_0111	0000_0000
0x004	-	-	-	LCDDVC[B,H,W] ---0_0000
	-	-	-	LCDCE0[B,H,W] 0000_0000
0x008	-	-	-	LCDSE0[B,H,W] 0000_0000
	-	-	-	LCDSE1[B,H,W] 0000_0000
0x00C	LCDSE3[B,H,W]	LCDSE2[B,H,W]	LCDSE1[B,H,W]	LCDSE0[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x010	-	-	-	LCDSE4[B,H,W] 0000_0000
	-	-	-	LCDSE5[B,H,W] 0000_0000
0x014	LCDRAM3[B,H,W] 1	LCDRAM2[B,H,W]	LCDRAM1[B,H, W]	LCDRAM0[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x018	LCDRAM7[B,H,W] 1	LCDRAM6[B,H,W]	LCDRAM5[B,H, W]	LCDRAM4[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x01C	LCDRAM11[B,H, W]	LCDRAM10[B,H,W]	LCDRAM9[B,H, W]	LCDRAM8[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x020	LCDRAM15[B,H, W]	LCDRAM14[B,H,W]	LCDRAM13[B,H, W]	LCDRAM12[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x024	LCDRAM19[B,H, W]	LCDRAM18[B,H,W]	LCDRAM17[B,H, W]	LCDRAM16[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x028	LCDRAM23[B,H, W]	LCDRAM22[B,H,W]	LCDRAM21[B,H, W]	LCDRAM20[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x02C	LCDRAM27[B,H, W]	LCDRAM26[B,H,W]	LCDRAM25[B,H, W]	LCDRAM24[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x030	LCDRAM31[B,H, W]	LCDRAM30[B,H,W]	LCDRAM29[B,H, W]	LCDRAM28[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x035	LCDRAM35[B,H, W]	LCDRAM35[B,H,W]	LCDRAM33[B,H, W]	LCDRAM32[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x038	LCDRAM39[B,H, W]	LCDRAM38[B,H,W]	LCDRAM37[B,H, W]	LCDRAM36[B,H,W]
	0000_0000	0000_0000	0000_0000	0000_0000
0x03C - 0xFFC	-			

GPIO

Base_Address: 0x4003_3000

Base_Address	Register			
+ Address	3	2	1	0
0x000	-	-	-	FN0[B,H,W] 0000_0000
0x004	-	-	-	FN1[B,H,W] 0000_0000
0x008	-	-	-	FN2[B,H,W] 0000_0000
0x00C	-	-	-	FN3[B,H,W] 0000_--11
0x010	-	-	-	FN4[B,H,W] 000_--00
0x014	-	-	-	FN5[B,H,W] 0000_0000
0x018	-	-	-	FN6[B,H,W] 0000_0000
0x01C	-	-	-	FN7[B,H,W] ---0_0000
0x020	-	-	-	FN8[B,H,W] ----_--0
0x024--0x0FF	-			
0x100	-	-	-	PU0[B,H,W] 0000_0000
0x104	-	-	-	PU1[B,H,W] 0000_0000
0x108	-	-	-	PU2[B,H,W] 0000_0000
0x10C	-	-	-	PU3[B,H,W] 0000_--11

Base_Address	Register			
+ Address	3	2	1	0
0x110	-	-	-	PU4[B,H,W] 000-_--00
0x114	-	-	-	PU5[B,H,W] 0000_0000
0x118	-	-	-	PU6[B,H,W] 0000_0000
0x11C	-	-	-	PU7[B,H,W] ---0_0000
0x120	-	-	-	PU8[B,H,W] ----_---0
0x124--0x1FF	-			
0x200	-	-	-	DIR0[B,H,W] 0000_0000
0x204	-	-	-	DIR1[B,H,W] 0000_0000
0x208	-	-	-	DIR2[B,H,W] 0000_0000
0x20C	-	-	-	DIR3[B,H,W] 0000_--00
0x210	-	-	-	DIR4[B,H,W] 000-_--00
0x214	-	-	-	DIR5[B,H,W] 0000_0000
0x218	-	-	-	DIR6[B,H,W] 0000_0000
0x21C	-	-	-	DIR7[B,H,W] ---0_0000
0x220	-	-	-	DIR8[B,H,W] ----_---0
0x224--0x2FF	-			

Base_Address	Register			
+ Address	3	2	1	0
0x300	-	-	-	GET0[B,H,W] 0000_0000
0x304	-	-	-	GET1[B,H,W] 0000_0000
0x308	-	-	-	GET2[B,H,W] 0000_0000
0x30C	-	-	-	GET3[B,H,W] 0000_--00
0x310	-	-	-	GET4[B,H,W] 000_--00
0x314	-	-	-	GET5[B,H,W] 0000_0000
0x318	-	-	-	GET6[B,H,W] 0000_0000
0x31C	-	-	-	GET7[B,H,W] ---0_0000
0x320--0x3FF	-			
0x400	-	-	-	SET0[B,H,W] 0000_0000
0x404	-	-	-	SET1[B,H,W] 0000_0000
0x408	-	-	-	SET2[B,H,W] 0000_0000
0x40C	-	-	-	SET3[B,H,W] 0000_--00
0x410	-	-	-	SET4[B,H,W] 000_--00

Base_Address	Register			
	3	2	1	0
+ Address	3	2	1	0
0x414	-	-	-	SET5[B,H,W] 0000_0000
0x418	-	-	-	SET6[B,H,W] 0000_0000
0x41C	-	-	-	SET7[B,H,W] ---0_0000
0x420	-	-	-	SET8[B,H,W] ----_---0
0x424--0x4FF	-			
0x500	-	-	ANALOG[B, H, W] 0000_0000_0000_0000	
0x504--0x57F	-			
0x580	-	-	-	SPS[B, H, W] ----_0101
0x584--0x5FF	-			
0x600	-	-	FN_SEL00[B, H, W] 0000_---1_0000_-000	
0x604	-			

Base_Address	Register			
+ Address	3	2	1	0
0x608	FN_SEL02[B, H, W] --00_0000_-000_00--_--00_0000_-000_00--			
0x60C	FN_SEL03[B, H, W] --00_0000_-00_00--_--00_0000_-00_00--			
0x610	FN_SEL04[B, H, W] 0000_0000_0000_0000_0000_0000_0000			
0x614	FN_SEL05[B, H, W] ----_0000_0000_----_--00_0000_----			
0x618	FN_SEL06[B, H, W] ----_----_----_0000_0000_0000_----			
0x61C	FN_SEL07[B, H, W] ----_0000_0000_----_--00_0000_0000			
0x620	FN_SEL08[B, H, W] ----_----_----_0000_0000_0000_0000			
0x624	-	-	FN_SEL09[H,W] --00_0000_-00_-00	
0x628--0x6FF	-			
0x700	-	-	-	ODR0[B, H, W] 0000_0000
0x704	-	-	-	ODR1[B, H, W] 0000_0000
0x708	-	-	-	ODR2[B, H, W] 0000_0000
0x70C	-	-	-	ODR3[B, H, W] 0000_-00
0x710	-	-	-	ODR4[B, H, W] 000_-00

Base_Address	Register			
+ Address	3	2	1	0
0x714	-	-	-	ODR5[B, H, W] 0000_0000
0x718	-	-	-	ODR6[B, H, W] 0000_0000
0x71C	-	-	-	ODR7[B, H, W] ---0_0000
0x720	-	-	-	ODR8[B, H, W] ----_---0
0x724--0x73F	-			
0x740	-	-	-	LVDIE[B, H, W] ----_---0
0x800--0x8FF	-			
0x900	-	-	-	FSETE0[B, H, W] 0000_0000
0x904	-	-	-	FSETE1[B, H, W] 0000_0000
0x908	-	-	-	FSETE2[B, H, W] 0000_0000
0x90C	-	-	-	FSETE3[B, H, W] 0000_--00
0x910	-	-	-	FSETE4[B, H, W] 000_--00
0x914	-	-	-	FSETE5[B, H, W] 0000_0000
0x918	-	-	-	FSETE6[B, H, W] 0000_0000
0x91C	-	-	-	FSETE7[B, H, W] ---0_0000
0x920	-	-	-	FSETE8[B, H, W] ----_---0

LVD

Base_Address: 0x4003_5000

Base_Address	Register			
+ Address	3	2	1	0
0x000	-	-	-	LVD_CTL1[B,H,W] 0000_0000
0x004	-	-	-	LVD_CTL2[B,H,W] -000_0000
0x008	-	-	LVD_STR[B,H,W] ----_--0 ----_---0	
0x00C	-	-	-	LVD_CLR[B,H,W] ----_---1
0x010	-			
0x014	-	-	-	LVD2_CTL[B,H,W] ----_0111
0x018	-	-	LVD2_STR [B,H,W] ----_--00 ----_---0	
0x01C	-	-	-	LVD2_CLR[B,H,W] ----_---1

Low Power mode

Base_Address: 0x4003_5000

Base_Address	Register			
+ Address	3	2	1	0
0x020	LDO_CTL[W] 0000_0000 0000_0000 ---- ---- --00			
0x024 – 0x02C	-			
0x030	-	-	-	BGR_CTL[B,H,W] ----_---0
0x035 – 0x0FC	-			

Base_Address	Register			
+ Address	3	2	1	0
0x104	-	-	-	CLK_CTL2[B,H,W] ---0_0101
0x108	-			
0x10C	-	-	-	DS_RAMP[B,H,W] ----_---0
0x110 - 0x1FC	-			

Base_Address	Register			
+ Address	3	2	1	0
0x200	-	-	-	WAKE_RSR[B,H,W] ----_--00
0x204	-	-	-	WAKE_ISR [B,H,W] --00_0000
0x208	-	-	-	WAKE_IER [B,H,W] --00_0000
0x20C	-	-	-	WAKE_LVR[B,H,W]]----_0000
0x210 - 0x6FC	-			

Base_Address	Register			
+ Address	3	2	1	0
0x700	BUR0104[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x704	BUR0508[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x708	BUR0912[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x70C	BUR1316[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x710 - 0xFFC	-			

SPI

ch0	Base_Address: 0x4003_8000
ch1	Base_Address: 0x4003_8100
ch2	Base_Address: 0x4003_8200
ch4	Base_Address: 0x4003_7000
ch5	Base_Address: 0x4003_7100
ch6	Base_Address: 0x4003_7200

Base_Address	Register			
+ Address	3	2	1	0
0x000	-	-	CR/BC[B,H,W]	MR[B,H,W]
			0--0_0000	000-_00-0
0x004	-	-	SR[B,H,W]	ECR/BS[B,H,W]
			0-00_0011	0000_0000
0x008	-	-	RXDR/TXDR[H,W]	
			0000_0000 0000_0000	
0x00C	-	-	BRS1[B,H,W]	BRS0[B,H,W]
			0000_0000	0000_0000
0x010	-	-	-	-
0x014	-	-	-	-
0x018	-	-	-	-
0x01C	-	-	CSTMG1/EBCR[B,H,W]	CSTMG0[B,H,W]
			0000_0000	0000_0000
0x020	-	-	CSTMG3[B,H,W]	CSTMG2[B,H,W]
			0000_0000	0000_0000
0x024	-	-	EACR[B,H,W]	
			--00_0-0 00-0_0000	
0x028	-	-	TMR[B,H,W]	
			0000_0000 0000_0000	
0x02C	-	-	TMC[B,H,W]	
			0000_0000 0000_0000	
0x030	-	-	CSCR[B,H,W]	
			0000_0000 0010_0000	
0x035	-	-	CSFR1[B,H,W]	CSFR0[B,H,W]
			1000_0000	1000_0000
0x038	-	-	-	CSFR2[B,H,W]
				1000_0000
0x03C	-	-	TDTCNT1[B,H,W]	TDTCNT0[B,H,W]
			0000_0000	0000_0000
0x040	-	-	TDTCNT3[B,H,W]	TDTCNT2[B,H,W]
			0000_0000	0000_0000
0x044—0x0FC	-	-	-	-

I²C

ch0	Base_Address: 0x4003_8000
ch1	Base_Address: 0x4003_8100
ch2	Base_Address: 0x4003_8200
ch4	Base_Address: 0x4003_7000
ch5	Base_Address: 0x4003_7100
ch6	Base_Address: 0x4003_7200

Base_Address	Register			
+ Address	3	2	1	0
0x000	-	-	CR/BC[B,H,W]	MR[B,H,W]
			0-0_0000	000-_00-0
0x004	-	-	SR[B,H,W]	ECR/BS[B,H,W]
			0-00_0011	0000_0000
0x008	-	-	RXDR/TXDR[H,W]	
			0000_0000 0000_0000	
0x00C	-	-	BRS1[B,H,W]	BRS0[B,H,W]
			0000_0000	0000_0000
0x010	-	-	SAMSK[B,H,W]	ADDR[B,H,W]
			-----	-----
0x014	-	-	-	-
0x018	-	-	-	-
0x01C	-	-	CSTMG1/EBCR[B,H,W]	-
			0000_0000	-
0x020—0x0FC	-	-	-	-

UART

ch0	Base_Address: 0x4003_8000
ch1	Base_Address: 0x4003_8100
ch2	Base_Address: 0x4003_8200
ch4	Base_Address: 0x4003_7000
ch5	Base_Address: 0x4003_7100
ch6	Base_Address: 0x4003_7200

Base_Address	Register			
+ Address	3	2	1	0
0x000	-	-	CR/BC[B,H,W]	MR[B,H,W]
			0--0_0000	000-00-0
0x004	-	-	SR[B,H,W]	ECR/BS[B,H,W]
			0-00_0011	0000_0000
0x008	-	-	RXDR/TXDR[H,W]	
			0000_0000 0000_0000	
0x00C	-	-	BRS1[B,H,W]	BRS0[B,H,W]
			0000_0000	0000_0000
0x010—0x04C	-	-	-	-
0x050	-	-	COMP1[B,H,W]	COMP0[B,H,W]
			0----0 00000000	
0x054 - 0x0FC	-	-	-	

RTC

Base_Address: 0x4003_B000

Address	+3	+2	+1	+0	
0x000	RTCCTL1[B,H,W] 0000_0000 0000_0000 --0_0000 -000_00-0				
0x004	RTCCTL2[B,H,W] ----_----_----_----_000 ----_---0				
0x008	-	CCS[B,H,W] 0000_0000 0000_0000 0000_0000			
0x00C	DATE[B,H,W] --00_0000	HOUR[B,H,W] --00_0000	MIN[B,H,W] -000_0000	SEN[B,H,W] -000_0000	
0x010	-	YEAR[B,H,W] 0000_0000	MON[B,H,W] ---0_0000	WEEK[B,H,W] ----_000	
0x014	ADATE[B,H,W]]--00_0000	AHOUR[B,H,W] --00_0000	AMIN[B,H,W] -000_0000	ASEN[B,H,W] -000_0000	
0x018	-	AYEAR[B,H,W] 0000_0000	AMON[B,H,W] ---0_0000	-	
0x01C	-	TVAL[B,H,W] ----_--00 0000_0000 0000_0000			
0x020	-	-	SCLKS[B,H,W] ----_--00	SCLKI[B,H,W] ----_---0	
0x024	-	CALEN[B,H,W] ----_---0	CALVAL[B,H,W] ----_--00 0000_0000		
0x028	-	-	CLKDIVC[B,H,W]]----_--00	CLKDIVX[B,H,W] ----_0000	
0x02C	-	-	CALCYC[B,H,W] ----_---0 0001_0011		
0x030	-	-	-	RTCODIV[B,H,W] ----_---0	
0x030 - 0xFFC	-				

Peripheral clock gating

Base_Address: 0x4003_C100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x00	CLKEN0[B,H,W] ---1-----1----11_1111			
0x04	RSTCTL0[B,H,W] ----0-----00_0000			
0x08 - 0x0C	-			
0x10	CLKEN1[B,H,W] ----1-----1--11			
0x14	RSTCTL1[B,H,W] ----0-----0--00			
0x18 - 0x1C	-			
0x20	CLKEN2[B,H,W] ----1---1---1---1----			
0x24	RSTCTL2[B,H,W] ----0---0---0---0----			
0x28 - 0xFC	-			

Buzzer

Base_Address: 0x4003_C200

Base_Address	Register			
+ Address	3	2	1	0
0x00	-	-	-	BUZ_CTL[B,H,W]
	-	-	-	----_--00
0x04	-	-	-	BUZ_CNT[B,H,W]
	-	-	-	0000_0000
0x08 – 0xFC	-	-	-	-

Voltage Comparator

Base_Address: 0x4003_C300

Base_Address	Register			
+ Address	3	2	1	0
0x00	-	-		VC_CTL1[B,H,W]
	-	-		0010_0000 --00_0000
0x04	-	-		VC_CTL2[B,H,W]
	-	-		--0_0000 0000_0000
0x08	-	-	-	VC_CTL3[B,H,W]
	-	-	-	0000_-0-0
0x0C – 0xFC	-	-	-	-

SCI7816

Base_Address: 0x4003_C800

Base_Address	Register			
+ Address	3	2	1	0
0x00	-	-	-	SCI_CON[B,H,W]
	-	-	-	0011_0001
0x04	-	-	-	SCI_MODE[B,H,W]
	-	-	-	0000_0000
0x08	-	-	-	SCI_STATUS[B,H,W]
	-	-	-	0000_0000
0x0C	-	-	-	SCI_FIFO_STATUS[B,H,W]
	-	-	-	0000_0000
0x10	-	-	-	SCI_FIFO_DEPTH[B,H,W]
	-	-	-	0000_0000
0x14	-	-	-	SCI_DATA[B,H,W]
	-	-	-	0000_0000
0x18	-	-		SCI_EDC_DATA[B,H,W]
	-	-		0000_0000 0000_0000
0x1C	-	-		SCI_SBDR[B,H,W]
	-	-		0000_0001 0111_0100
0x20	-	-		SCI_ETU_CNT[B,H,W]
	-	-		0010_0101 1000_0000
0x24	-	-	-	SCI_F1_NUM[B,H,W]
	-	-	-	----_000
0x28	-	-	-	SCI_CLK_RST[B,H,W]
	-	-	-	---0_0000
0x2C	-	-	-	SCI_INT_EN[B,H,W]
	-	-	-	----_000
0x30	-	-	-	SCI_WAKEUP_INT[B,H,W]
	-	-	-	----_--0
0x35 – 0xFC	-	-	-	-

AES

Base_Address: 0x4006_B000

Base_Address	Register					
+ Address	3	2	1	0		
0x000	-	-	AES_CON[B,H,W]			
	-	-	---1_1100 1-01_0000			
0x004 – 0x00C	-					
0x010 – 0x01C	AES_DATA[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000					
0x020 – 0x03C	AES_KEY[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000					
0x040	AES_RNG0[B,H,W] XXXX_0110 XXXX_0011 XXXX_1000 XXXX_1101					
	AES_RNG1[B,H,W] XXXX_0101 XXXX_1010 XXXX_1011 XXXX_0100					
0x048 – 0x04F	-					
0x050	-	-	-	AES_INT_CON[B,H, W]		
	-	-	-	----_--00		
0x054 – 0x05F	-					
0x060	-	-	-	AES_DIV_EN[B,H,W]		
	-	-	-	----_---0		
0x064 – 0x0FC	-					

MTB_DWT

Base_Address: 0xF000_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x004	CMP_DATA_START[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x008	CMP_MASK_START[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x014	CMP_DATA_STOP[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x018	CMP_MASK_STOP[B,H,W] 0000_0000 0000_0000 0000_0000 0000_0000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W] 0000_0000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFD4	PID5[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFD8	PID6[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFDC	PID7[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFE0	PID0[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFE4	PID1[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFE8	PID2[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFEC	PID3[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFF0	CID0[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFF4	CID1[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			
0xFF8	CID2[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx			

0xFFC	CID3[B,H,W] xxxx_xxxx xxxx_xxxx xxxx_xxxx xxxx_xxxx
-------	--

Fast GPIO

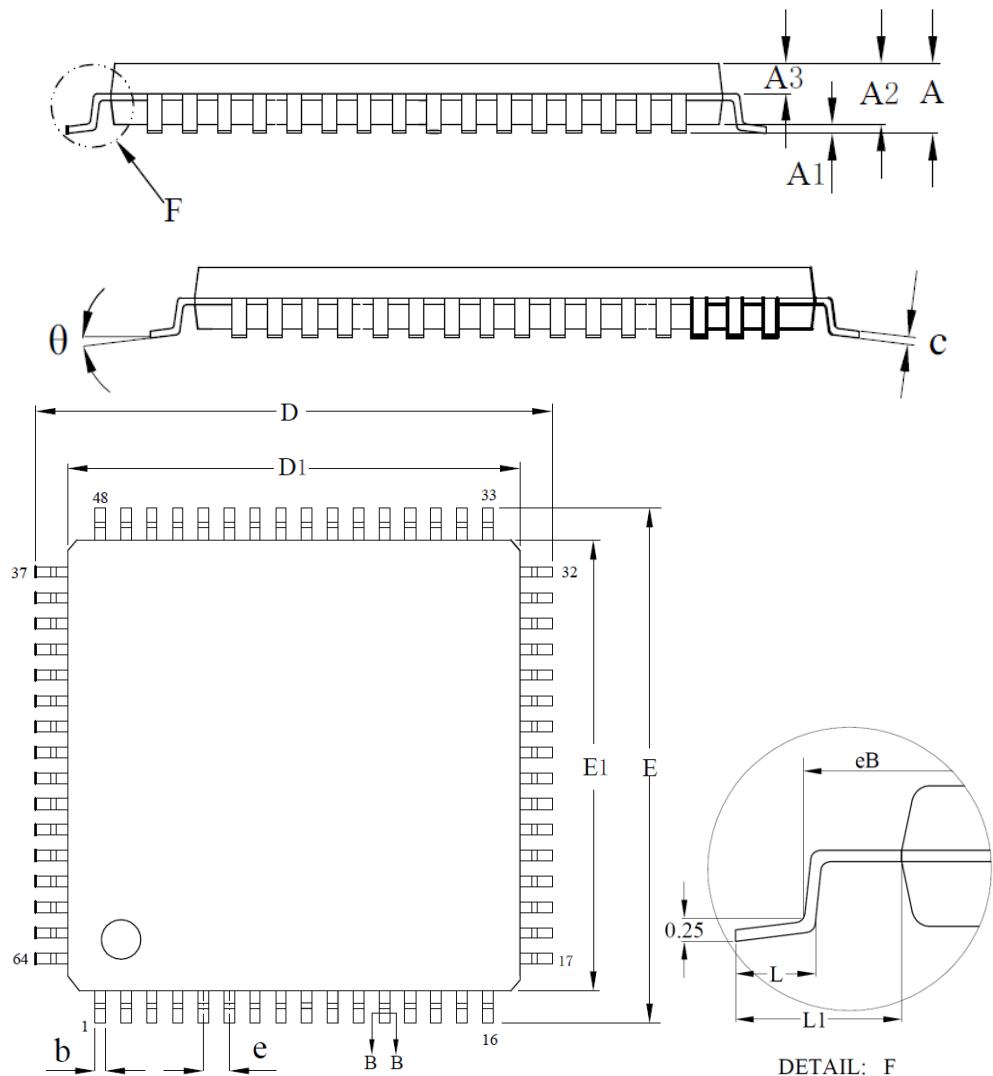
Base_Address: 0xF800_0000

Base_Address	Register			
	3	2	1	0
0x000	-	-	-	FGET0[B,H,W] xxxx_xxxx
0x004	-	-	-	FGET1[B,H,W] xxxx_xxxx
0x008	-	-	-	FGET2[B,H,W] xxxx_xxxx
0x00C	-	-	-	FGET3[B,H,W] xxxx_xxxx
0x010	-	-	-	FGET4[B,H,W] xxxx_xxxx
0x014	-	-	-	FGET5[B,H,W] xxxx_xxxx
0x018	-	-	-	FGET6[B,H,W] xxxx_xxxx
0x01C	-	-	-	FGET7[B,H,W] xxxx_xxxx
0x020	-	-	-	FGET8[B,H,W] xxxx_xxxx
0x024--0x03C	-			

Base_Address	Register			
+ Address	3	2	1	0
0x040	-	-	-	FSET0[B,H,W] 0000_0000
0x044	-	-	-	FSET1[B,H,W] 0000_0000
0x048	-	-	-	FSET2[B,H,W] 0000_0000
0x04C	-	-	-	FSET3[B,H,W] 0000_0000
0x050	-	-	-	FSET4[B,H,W] 0000_0000
0x054	-	-	-	FSET5[B,H,W] 0000_0000
0x058	-	-	-	FSET6[B,H,W] 0000_0000
0x05C	-	-	-	FSET7[B,H,W] 0000_0000
0x060	-	-	-	FSET8[B,H,W] 0000_0000

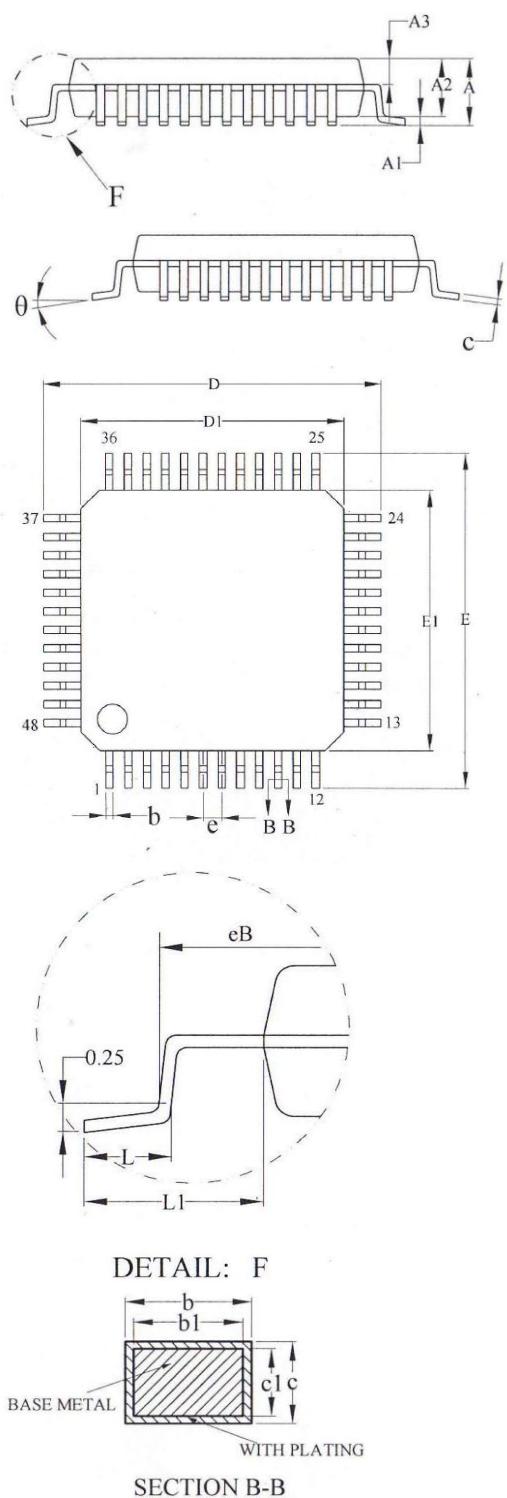
附录 2 封装尺寸

LQFP-64 封装



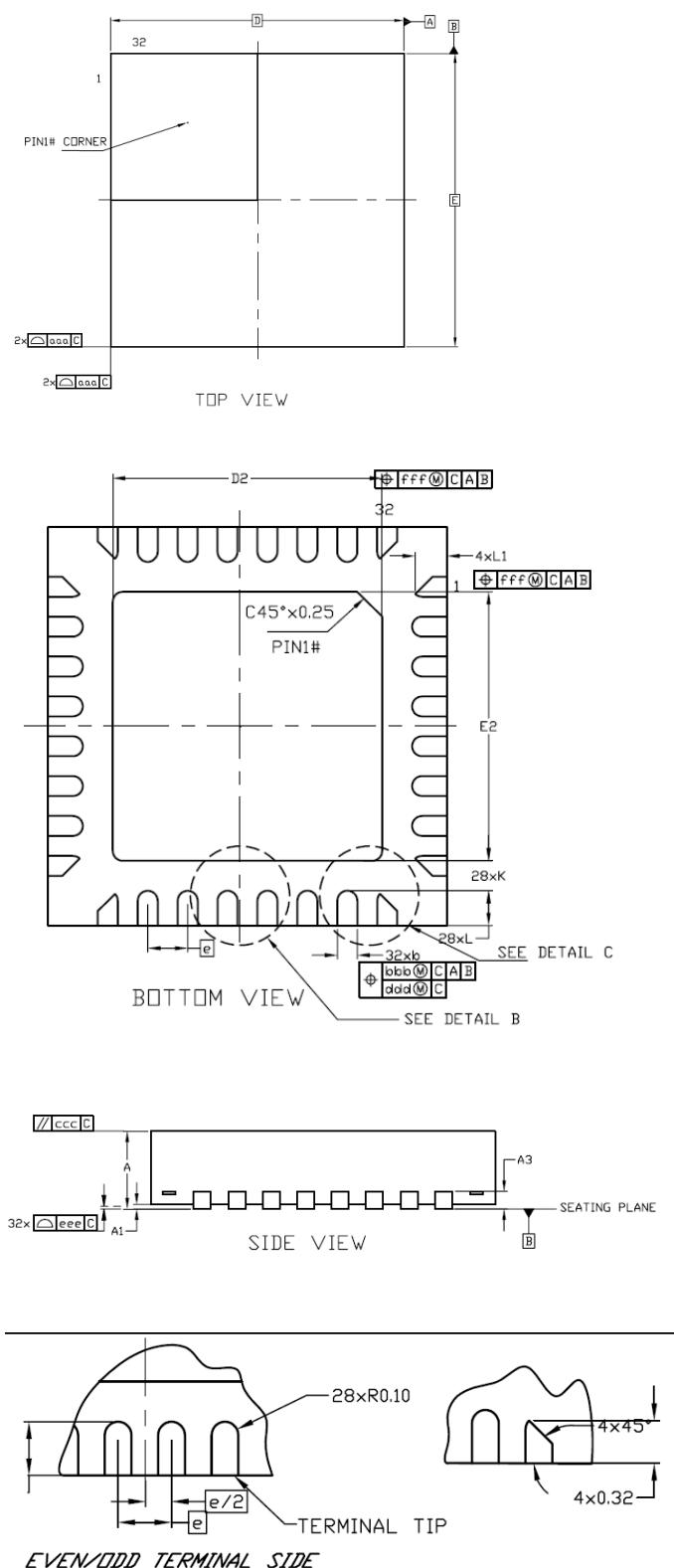
Symbol	LQFP64 (10x10)			LQFP64 (7x7)		
	Min	Nom	Max	Min	Nom	Max
A	--	--	1.60	--	--	1.60
A1	0.05	--	0.15	0.05	--	0.15
A2	1.35	1.40	1.45	1.35	1.40	1.45
A3	0.59	0.64	0.69	0.59	0.64	0.69
b	0.18	--	0.26	0.16	--	0.24
b1	0.17	0.20	0.23	0.15	0.18	0.21
c	0.13	--	0.17	0.13	--	0.17
c1	0.12	0.13	0.14	0.12	0.13	0.14
D	11.80	12.00	12.20	8.80	9.00	9.20
D1	9.90	10.00	10.10	6.90	7.00	7.10
E	11.80	12.00	12.20	8.80	9.00	9.20
E1	9.90	10.00	10.10	6.90	7.00	7.10
eB	11.25	--	11.45	8.10	--	8.25
e	0.50BSC			0.40BSC		
L	0.45	--	0.75	0.40	--	0.65
L1	1.00REF			1.00REF		
θ	0°	--	7°	0°	--	7°

LQFP48 封装



Symbol	Millimeter		
	Min	Nom	Max
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	--	0.26
b1	0.17	0.20	0.23
c	0.13	--	0.17
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	--	8.25
e	0.50BSC		
L	0.40	--	0.65
L1	1.00REF		
θ	0	--	7°

QFN32 封装



Symbol	QFN32		
	Min	Nom	Max
A	0.70	0.75	0.80
	0.85	0.90	0.95
A1	0	0.02	0.05
A3	--	0.20REF	--
b	0.15	0.20	0.25
D	4.0BSC		
E	4.0BSC		
D2	2.60	2.70	2.80
E2	2.60	2.70	2.80
e	0.40BSC		
L	0.30	0.35	0.40
L1	0.27	0.32	0.37
K	0.20	--	--
aaa	0.10		
bbb	0.07		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

版本记录 & 联系方式

版本	修改日期	修改内容摘要
Rev1.0	2018/07/12	初版发布。



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email : mcu@hdsc.com.cn

网址 : www.hdsc.com.cn

通信地址 : 上海市张江高科园区碧波路 572 弄 39 号

邮编 : 201203

