



BAT32A237用户手册

基于ARM® Cortex®-M0+的超低功耗32位微控制器

V1.0.3

请注意以下有关CMS知识产权政策

* 中微半导体(深圳)股份有限公司(以下简称本公司)已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体(深圳)股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

文档使用说明

本手册是BAT32A237微控制器产品的技术参考手册，技术参考手册是有关如何使用本系列产品的应用说明资料，包含各个功能模块的结构、功能描述、工作模式以及寄存器配置等详细信息。

技术参考手册是针对这一系列产品所有功能模块的说明，若要了解产品的特征说明(即功能搭载情况)，可参考相应的数据手册。

数据手册信息如下：

BAT32A237xx: BAT32A237_数据手册_vx.x.x. pdf

通常在芯片选型的初期，首先要看数据手册，以评估该产品是否能够满足设计上的功能需求；在基本选定所需产品后，需要查看技术参考手册，以确定各功能模块的工作模式是否符合要求；在确定选型进入编程设计阶段时，需要详细阅读技术参考手册，以获知各项功能的具体实现方式和寄存器配置。在设计硬件时可参考数据手册以获得电压，电流，驱动能力以及管脚分配等信息。

关于Cortex-M0+核心、SysTick定时器和NVIC的详细说明，请参照对应ARM的文档。

目录

文档使用说明.....	2
第1章 CPU	19
1.1 概述	19
1.2 Cortex-M0+内核特性.....	19
1.3 调试特性	19
1.4 SWD接口引脚.....	21
1.5 ARM参考文档	22
第2章 引脚功能	23
2.1 端口功能	23
2.2 端口复用功能	23
2.3 控制端口功能的寄存器.....	24
2.3.1 端口模式寄存器(PMxx).....	26
2.3.2 端口寄存器(Pxx).....	27
2.3.3 上拉电阻选择寄存器(PUxx).....	28
2.3.4 端口输入模式寄存器(PIMxx).....	29
2.3.5 端口输出模式寄存器(POMxx).....	30
2.3.6 端口模式控制寄存器(PMCxx).....	31
2.3.7 外围I/O重定向寄存器0(PIOR0).....	32
2.3.8 外围I/O重定向寄存器1(PIOR1).....	34
2.3.9 外围I/O重定向寄存器2(PIOR2).....	35
2.3.10 外围I/O重定向寄存器3(PIOR3).....	36
2.4 未使用引脚的处理	37
2.5 使用复用功能时的寄存器设置	38
2.5.1 使用复用功能时的基本思想	38
2.5.2 使用的端口功能和复用功能的寄存器设置例子.....	39
第3章 系统结构	50
3.1 概述	50
3.2 系统地址划分	51
第4章 时钟发生电路	53
4.1 时钟发生电路的功能	53
4.2 时钟发生电路的结构	55
4.3 控制时钟发生电路的寄存器.....	58
4.3.1 时钟运行模式控制寄存器(CMC)	59
4.3.2 系统时钟控制寄存器(CKC).....	60
4.3.3 时钟运行状态控制寄存器(CSC).....	61
4.3.4 振荡稳定时间计数器的状态寄存器(OSTC).....	62
4.3.5 振荡稳定时间选择寄存器(OSTS).....	64
4.3.6 外围允许寄存器0、1(PER0、PER1).....	65
4.3.7 副系统时钟提供模式控制寄存器(OSMC).....	70
4.3.8 高速内部振荡器的频率选择寄存器(HOCODIV).....	71
4.3.9 高速内部振荡器的微调寄存器(HIOTRM).....	72
4.4 系统时钟振荡电路	73
4.4.1 X1振荡电路	73
4.4.2 XT1振荡电路	73
4.4.3 高速内部振荡器	77
4.4.4 低速内部振荡器	77

4.5	时钟发生电路的运行	78
4.6	时钟控制	80
4.6.1	高速内部振荡器的设置例子	80
4.6.2	X1振荡电路的设置例子	82
4.6.3	XT1振荡电路的设置例子	83
4.6.4	CPU时钟的状态转移图	84
4.6.5	CPU时钟转移前的条件和转移后的处理	90
4.6.6	CPU时钟和主系统时钟的切换所需时间	92
4.6.7	时钟振荡停止前的条件	93
第5章	硬件除法器	94
5.1	特点	94
5.2	功能描述	94
5.3	硬件除法器的寄存器	94
5.3.1	被除数寄存器(DIVIDEND)	95
5.3.2	除数寄存器(DIVISOR)	95
5.3.3	商寄存器(QUOTIENT)	95
5.3.4	余数寄存器(REMAINDER)	95
5.3.5	状态寄存器(STATUS)	96
第6章	通用定时器单元Timer4	97
6.1	通用定时器单元的功能	99
6.1.1	独立通道运行功能	99
6.1.2	多通道联动运行功能	101
6.1.3	8位定时器运行功能(只限于单元0的通道1和通道3)	102
6.1.4	LIN-bus支持功能(只限于单元0的通道3)	102
6.2	通用定时器单元的结构	103
6.2.1	通用定时器单元寄存器列表	106
6.2.2	定时器计数寄存器mn (TCRmn)	107
6.2.3	定时器数据寄存器mn(TDRmn)	109
6.3	控制通用定时器单元的寄存器	110
6.3.1	外围允许寄存器0 (PER0)	111
6.3.2	定时器时钟选择寄存器m(TPSm)	112
6.3.3	定时器模式寄存器mn(TMRmn)	115
6.3.4	定时器状态寄存器mn(TSRmn)	119
6.3.5	定时器通道允许状态寄存器m(TEm)	120
6.3.6	定时器通道开始寄存器m (TSm)	121
6.3.7	定时器通道停止寄存器m(TTm)	122
6.3.8	定时器输入输出选择寄存器(TIOS0,TIOS1)	123
6.3.9	定时器输出允许寄存器m(TOEm)	125
6.3.10	定时器输出寄存器m(TOm)	126
6.3.11	定时器输出电平寄存器m(TOLm)	127
6.3.12	定时器输出模式寄存器m(TOMm)	128
6.3.13	输入切换控制寄存器(ISC)	129
6.3.14	噪声滤波器允许寄存器(NFEN1)	130
6.3.15	控制定时器输入/输出引脚端口功能的寄存器	131
6.4	通用定时器单元的基本规则	132
6.4.1	多通道联动运行功能的基本规则	132
6.4.2	8位定时器运行功能的基本规则(只限于通道1和通道3)	134
6.5	计数器的运行	135

6.5.1	计数时钟(F_{TCLK})	135
6.5.2	计数器的开始时序.....	137
6.5.3	计数器的运行.....	138
6.6	通道输出(TOmn引脚)的控制	143
6.6.1	TOmn引脚输出电路的结构	143
6.6.2	TOmn引脚的输出设置	144
6.6.3	通道输出运行的注意事项	145
6.6.4	TOmn位的一次性操作	149
6.6.5	有关开始计数时的定时器中断和TOmn引脚输出	150
6.7	定时器输入(TImn)的控制.....	151
6.7.1	TImn引脚输入电路的结构.....	151
6.7.2	噪声滤波器	151
6.7.3	操作通道输入时的注意事项	152
6.8	通用定时器单元的独立通道运行功能	153
6.8.1	作为间隔定时器/方波输出的运行	153
6.8.2	作为外部事件计数器的运行	157
6.8.3	作为分频器的运行(只限于单元0 的通道0).....	160
6.8.4	作为输入脉冲间隔测量的运行.....	163
6.8.5	作为输入信号高低电平宽度测量的运行	166
6.8.6	作为延迟计数器的运行	170
6.9	通用定时器单元的多通道联动运行功能	173
6.9.1	作为单触发脉冲输出功能的运行	173
6.9.2	作为PWM功能的运行	180
6.9.3	作为多重PWM输出功能的运行	186
6.10	使用通用定时器单元时的注意事项	194
6.10.1	使用定时器输出时的注意事项.....	194
第7章	定时器A	195
7.1	定时器A的功能.....	195
7.2	定时器A的结构.....	196
7.3	控制定时器A的寄存器	197
7.3.1	外围允许寄存器1(PER1)	197
7.3.2	副系统时钟提供模式控制寄存器(OSMC)	198
7.3.3	定时器A计数寄存器0(TA0)	199
7.3.4	定时器A控制寄存器0(TACR0)	200
7.3.5	定时器AI/O控制寄存器0(TAIOC0)	201
7.3.6	定时器A控制寄存器0(TAMR0).....	203
7.3.7	定时器A事件引脚选择寄存器0(TAISR0)	204
7.3.8	端口模式寄存器x(PMx).....	205
7.4	定时器A的运行.....	206
7.4.1	重加载寄存器和计数器的改写.....	206
7.4.2	定时器模式	207
7.4.3	脉冲输出模式.....	208
7.4.4	事件计数器模式	209
7.4.5	脉宽测量模式.....	210
7.4.6	脉冲周期测量模式.....	211
7.4.7	与EVENTC的协作	212
7.4.8	各模式的输出设置.....	212
7.5	使用定时器A时的注意事项	213

7.5.1	计数的开始和停止控制	213
7.5.2	标志的存取(TACR0寄存器的TEDGF位和TUNDF位)	213
7.5.3	计数寄存器的存取	213
7.5.4	模式的变更	213
7.5.5	TAO引脚和TAIO引脚的设置步骤	214
7.5.6	不使用定时器A的情况	214
7.5.7	定时器A运行时钟的停止	214
7.5.8	深度睡眠模式(事件计数器模式)的设置步骤	215
7.5.9	深度睡眠模式中(只限于事件计数器模式)的功能限制	215
7.5.10	通过TSTOP位进行强制的计数停止	215
7.5.11	数字滤波器	215
7.5.12	选择F _{IL} 作为计数源的情况	215
第8章 定时器B		216
8.1	定时器B的功能	216
8.2	定时器B的结构	217
8.3	控制定时器B的寄存器	218
8.3.1	外围允许寄存器1(PER1)	219
8.3.2	定时器B模式寄存器(TBMR)	220
8.3.3	定时器B计数控制寄存器(TBCNTC)	221
8.3.4	定时器B控制寄存器(TBCR)	222
8.3.5	定时器B中断允许寄存器(TBIER)	223
8.3.6	定时器B状态寄存器(TBSR)	224
8.3.7	定时器BI/O控制寄存器(TBIOR)	226
8.3.8	定时器B计数器(TB)	228
8.3.9	定时器B通用寄存器A、B、C、D	229
8.3.10	端口寄存器和端口模式寄存器	231
8.4	定时器B的运行	232
8.4.1	有关多个模式和功能的共同事项	232
8.4.2	定时器模式(输入捕捉功能)	237
8.4.3	定时器模式(输出比较功能)	240
8.4.4	PWM模式	244
8.4.5	相位计数模式	248
8.5	定时器B中断	251
8.6	使用定时器B时的注意事项	253
8.6.1	相位计数模式中的相位差、重叠和脉宽	253
8.6.2	模式的切换	253
8.6.3	计数源的切换	253
8.6.4	TBIO0引脚和TBIO1引脚的设置步骤	254
8.6.5	外部时钟TBCLK0和TBCLK1	254
8.6.6	SFR的读写存取	255
8.6.7	停止计数时的输入捕捉运行	255
第9章 定时器C		256
9.1	定时器C的功能	256
9.2	定时器C的结构	257
9.3	控制定时器C的寄存器	258
9.3.1	外围允许寄存器1(PER1)	258
9.3.2	定时器C计数寄存器(TC)	259
9.3.3	定时器C计数缓冲寄存器(TCBUF)	259

9.3.4	定时器C控制寄存器1(TCCR1).....	260
9.3.5	定时器C控制寄存器1(TCCR2).....	261
9.3.6	定时器C状态寄存器(TCSR).....	262
9.4	定时器C的运行.....	263
9.4.1	计数源.....	263
9.4.2	定时器C开始计数的动作.....	263
9.4.2.1	选择Timer M的信号作为触发时的设置和动作.....	264
9.4.2.2	选择软件触发时的设置和动作.....	265
9.4.3	定时器C计数停止的动作.....	266
9.4.3.1	选择比较器1作为触发时的设置和动作.....	266
9.4.3.2	软件触发时的设置和动作.....	266
9.4.4	输入捕捉动作.....	267
9.4.5	定时器C计数复位动作.....	268
9.4.6	定时器C的中断.....	270
9.5	定时器C使用时的注意事项.....	271
9.5.1	寄存器的读写.....	271
9.5.2	上溢中断.....	271
9.5.3	输入捕捉和定时器C计数复位动作.....	271
9.5.4	定时器C和定时器M, 比较器1联动时的步骤.....	271
第10章	定时器M.....	272
10.1	定时器M的功能.....	272
10.2	定时器M的结构.....	273
10.3	控制定时器M的寄存器.....	274
10.3.1	外围允许寄存器1(PER1).....	275
10.3.2	定时器M EVENTC 寄存器 (TMELC).....	276
10.3.3	定时器M启动寄存器(TMSTR).....	277
10.3.4	定时器M模式寄存器(TMMR).....	278
10.3.5	定时器M PWM功能选择寄存器(TMPMR).....	279
10.3.6	定时器M功能控制寄存器(TMFCR).....	280
10.3.7	定时器 M 输出主允许寄存器1(TMOER1).....	281
10.3.8	定时器M输出主允许寄存器2(TMOER2).....	282
10.3.9	定时器M输出控制寄存器(TMOCR).....	283
10.3.10	定时器M数字滤波器功能选择寄存器i(TMDFi)(i=0、1).....	286
10.3.11	定时器M控制寄存器i(TMCri)(i=0、1).....	288
10.3.12	定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1).....	293
10.3.13	定时器MI/O控制寄存器Ci(TMIORCi)(i=0、1).....	295
10.3.14	定时器M状态寄存器0(TMSR0).....	297
10.3.15	定时器M状态寄存器1(TMSR1).....	301
10.3.16	定时器M中断允许寄存器i(TMIERi)(i=0、1).....	307
10.3.17	定时器MPWM功能输出电平控制寄存器i(TMPOCRi)(i=0、1).....	308
10.3.18	定时器M计数器i(TMi)(i=0、1).....	309
10.3.19	定时器M通用寄存器Ai、Bi、Ci、Di.....	311
10.3.20	端口模式寄存器(PMxx, PMCxx).....	320
10.4	有关多个模式的共同事项.....	321
10.4.1	计数源.....	321
10.4.2	缓冲器运行.....	322
10.4.3	同步运行.....	325
10.4.4	脉冲输出的强制截止.....	326

10.4.5	从事件联动控制器(EVENTC)输入的事件	328
10.4.6	向事件联动控制器(EVENTC)/数据传送控制器(DMA)输出的事件	329
10.5	定时器M的运行	330
10.5.1	输入捕捉功能	330
10.5.2	输出比较功能	334
10.5.3	PWM功能	341
10.5.4	复位同步PWM模式	345
10.5.5	互补PWM模式	348
10.5.6	PWM3模式	352
10.6	定时器M中断	356
10.7	使用定时器M时的注意事项	358
10.7.1	SFR的读写存取	358
10.7.2	模式的切换	358
10.7.3	计数源	359
10.7.4	输入捕捉功能	359
10.7.5	TMIOAi、TMIOBi、TMIOCi、TMIODi引脚的设置步骤(i=0、1)	359
10.7.6	外部时钟TMCLK	360
10.7.7	互补PWM模式	360
10.8	PWMOP	364
10.8.1	PWMOP的功能	365
10.8.2	PWMOP的寄存器	365
10.8.3	PWMOP的运行	371
10.8.3.1	输出强制截止	371
10.8.3.2	硬件解除(HS_SEL=0)	372
10.8.3.3	软件解除(HS_SEL=1)	381
10.8.3.4	Hazard对策	387
10.8.3.5	输出强制截止源检出状态和未检出状态	388
10.8.3.6	定时器M的计数器的值到达0000H时的时序图	389
10.8.3.7	设置步骤	391
10.8.4	注意事项	392
第11章	实时时钟	393
11.1	实时时钟的功能	393
11.2	实时时钟的结构	393
11.3	控制实时时钟的寄存器	395
11.3.1	外围允许寄存器0(PER0)	396
11.3.2	实时时钟选择寄存器(RTCCL)	397
11.3.3	实时时钟控制寄存器0(RTCC0)	398
11.3.4	实时时钟控制寄存器1(RTCC1)	399
11.3.5	时钟误差校正寄存器(SUBCUD)	401
11.3.6	秒计数寄存器(SEC)	402
11.3.7	分钟计数寄存器(MIN)	402
11.3.8	小时计数寄存器(HOUR)	403
11.3.9	日计数寄存器(DAY)	405
11.3.10	星期计数寄存器(WEEK)	406
11.3.11	月计数寄存器(MONTH)	407
11.3.12	年计数寄存器(YEAR)	407
11.3.13	闹钟分钟寄存器(ALARMWM)	408
11.3.14	闹钟小时寄存器(ALARMWH)	408

11.3.15	闹钟星期寄存器(ALARMWW).....	409
11.3.16	端口模式寄存器和端口寄存器.....	409
11.4	实时时钟的运行.....	410
11.4.1	实时时钟的运行开始.....	410
11.4.2	开始运行后睡眠模式的转移.....	411
11.4.3	实时时钟计数器的读写.....	412
11.4.4	实时时钟的闹钟设置.....	414
11.4.5	实时时钟的1Hz输出.....	415
11.4.6	实时时钟的时钟误差校正例子.....	416
第12章	15位间隔定时器.....	418
12.1	15位间隔定时器的功能.....	418
12.2	15位间隔定时器的结构.....	418
12.3	控制15位间隔定时器的寄存器.....	419
12.3.1	外围允许寄存器0(PER0).....	419
12.3.2	实时时钟选择寄存器(RTCCL).....	420
12.3.3	15位间隔定时器的控制寄存器(ITMC).....	421
12.4	15位间隔定时器的运行.....	422
12.4.1	15位间隔定时器的运行时序.....	422
12.4.2	从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移.....	423
第13章	时钟输出/蜂鸣器输出控制电路.....	424
13.1	时钟输出/蜂鸣器输出控制电路的功能.....	424
13.2	时钟输出/蜂鸣器输出控制电路的结构.....	425
13.3	控制时钟输出/蜂鸣器输出控制电路的寄存器.....	425
13.3.1	时钟输出选择寄存器n(CKSn).....	425
13.3.2	控制时钟输出/蜂鸣器输出引脚端口功能的寄存器.....	427
13.4	时钟输出/蜂鸣器输出控制电路的运行.....	428
13.4.1	输出引脚的运行.....	428
13.5	时钟输出/蜂鸣器输出控制电路的注意事项.....	428
第14章	看门狗定时器.....	429
14.1	看门狗定时器的功能.....	429
14.2	看门狗定时器的结构.....	429
14.3	控制看门狗定时器的寄存器.....	431
14.3.1	看门狗定时器的允许寄存器(WDTE).....	431
14.3.2	LOCKUP控制寄存器(LOCKCTL)及其保护寄存器(PRCR).....	432
14.3.3	WDTCFG配置寄存器(WDTCFG0/1/2/3).....	433
14.4	看门狗定时器的运行.....	434
14.4.1	看门狗定时器的运行控制.....	434
14.4.2	看门狗定时器上溢时间的设置.....	435
14.4.3	看门狗定时器窗口打开期间的设置.....	436
14.4.4	看门狗定时器间隔中断的设置.....	437
14.4.5	WDTCFG未配置时看门狗定时器的运行.....	437
第15章	A/D转换器.....	438
15.1	A/D转换器的功能.....	438
15.2	控制A/D转换器的寄存器.....	440
15.2.1	外围允许寄存器0(PER0).....	441
15.2.2	A/D转换器的模式寄存器0(ADM0).....	442
15.2.3	A/D转换器的模式寄存器1(ADM1).....	447
15.2.4	A/D转换器的模式寄存器2(ADM2).....	448

15.2.5	A/D转换器的触发模式寄存器(ADTRG)	449
15.2.6	模拟输入通道指定寄存器(ADS)	450
15.2.7	12位A/D转换结果寄存器(ADCR)	452
15.2.8	8位A/D转换结果寄存器(ADCRH).....	453
15.2.9	转换结果比较上限值设置寄存器(ADUL)	454
15.2.10	转换结果比较下限值设置寄存器(ADLL).....	454
15.2.11	A/D采样时间控制寄存器(ADNSMP)	455
15.2.12	A/D采样时间延长寄存器(ADSMPWAIT)	456
15.2.13	A/D测试寄存器(ADTES)	457
15.2.14	A/D状态寄存器(ADFLG).....	458
15.2.15	A/D充放电控制寄存器(ADNDIS).....	459
15.2.16	控制模拟输入引脚端口功能的寄存器	460
15.3	输入电压和转换结果	461
15.4	A/D转换器的运行模式	462
15.4.1	软件触发模式(选择模式、连续转换模式).....	462
15.4.2	软件触发模式(选择模式、单次转换模式).....	463
15.4.3	软件触发模式(扫描模式、连续转换模式).....	464
15.4.4	软件触发模式(扫描模式、单次转换模式).....	465
15.4.5	硬件触发无等待模式(选择模式、连续转换模式)	466
15.4.6	硬件触发无等待模式(选择模式、单次转换模式)	467
15.4.7	硬件触发无等待模式(扫描模式、连续转换模式)	468
15.4.8	硬件触发无等待模式(扫描模式、单次转换模式)	469
15.4.9	硬件触发等待模式(选择模式、连续转换模式).....	470
15.4.10	硬件触发等待模式(选择模式、单次转换模式).....	471
15.4.11	硬件触发等待模式(扫描模式、连续转换模式).....	472
15.4.12	硬件触发等待模式(扫描模式、单次转换模式).....	473
15.5	转换器的设置流程图	474
15.5.1	软件触发模式的设置	474
15.5.2	硬件触发无等待模式的设置	475
15.5.3	硬件触发等待模式的设置	476
15.5.4	选择温度传感器的输出电压/内部基准电压时的设置.....	477
15.5.5	测试模式的设置	478
第16章	D/A转换器.....	479
16.1	D/A转换器的功能	479
16.2	D/A转换器的结构	480
16.3	控制D/A转换器的寄存器.....	481
16.3.1	外围允许寄存器1(PER1)	481
16.3.2	D/A转换器的模式寄存器(DAM).....	482
16.3.3	D/A转换值设置寄存器i(DACSi)(i=0、1)	482
16.3.4	事件输出目标选择寄存器n(ELSELRn)、n=00~21.....	483
16.3.5	控制模拟输入引脚端口功能的寄存器	483
16.4	D/A转换器的运行	484
16.4.1	通常模式的运行	484
16.4.2	实时输出模式的运行	485
16.4.3	D/A转换值的输出时序.....	486
16.5	使用D/A转换器时的注意事项	487
第17章	比较器.....	488
17.1	比较器的功能	488

17.2	比较器的结构	489
17.3	控制比较器的寄存器	491
17.3.1	外围允许寄存器1(PER1)	492
17.3.2	比较器模式设置寄存器(COMPMDR)	493
17.3.3	比较器滤波控制寄存器(COMPFR).....	494
17.3.4	比较器输出控制寄存器(COMPOCR)	496
17.3.5	比较器内置基准电压控制寄存器(CVRCTL)	498
17.3.6	比较器内置基准电压选择寄存器(CiRVM)	499
17.3.7	比较器0的输入信号选择控制寄存器(CMPSEL0)	500
17.3.8	比较器1的输入信号选择控制寄存器(CMPSEL1)	501
17.3.9	控制模拟输入引脚端口功能的寄存器	502
17.4	运行说明	503
17.4.1	比较器i的数字滤波器(i=0、1)	505
17.4.2	比较器i中断(i=0、1).....	505
17.4.3	向联动控制器(EVENTC)输出的事件信号	506
17.4.4	比较器i的输出(i=0、1)	507
17.4.5	比较器时钟的停止和提供	507
第18章	可编程增益放大器(PGA).....	508
18.1	可编程增益放大器的功能	508
18.2	可编程增益放大器的结构	508
18.3	可编程增益放大器的寄存器	509
18.3.1	外围允许寄存器1(PER1)	509
18.3.2	可编程增益放大器控制寄存器(PGAnCTL)	510
18.3.3	控制模拟输入引脚端口功能的寄存器	510
18.4	可编程增益放大器的运行	511
18.4.1	可编程增益放大器的开始运行步骤.....	511
18.4.2	可编程增益放大器的停止运行步骤.....	512
第19章	通用串行通信单元	513
19.1	通用串行通信单元的功能	515
19.1.1	3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21).....	515
19.1.2	UART(UART0~UART2)	516
19.1.3	简易I ² C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21).....	517
19.2	通用串行通信单元的结构	518
19.2.1	移位寄存器	521
19.2.2	串行数据寄存器mn(SDRmn)的低8位或者低9位	521
19.3	控制通用串行通信单元的寄存器.....	523
19.3.1	外围允许寄存器0(PER0)	524
19.3.2	串行时钟选择寄存器m(SPSm)	525
19.3.3	串行模式寄存器mn(SMRmn).....	526
19.3.4	串行通信运行设置寄存器mn(SCRmn).....	528
19.3.5	串行数据寄存器mn(SDRmn)	530
19.3.6	串行标志清除触发寄存器mn(SIRmn).....	532
19.3.7	串行状态寄存器mn(SSRmn)	533
19.3.8	串行通道开始寄存器m(SSm).....	535
19.3.9	串行通道停止寄存器m(STm)	536
19.3.10	串行通道允许状态寄存器m(SEm).....	537
19.3.11	串行输出允许寄存器m(SOEm)	538
19.3.12	串行输出寄存器m(SOm).....	539

19.3.13	串行输出电平寄存器m(SOLm)	540
19.3.14	输入切换控制寄存器(ISC)	542
19.3.15	噪声滤波器允许寄存器0(NFEN0)	543
19.3.16	控制串行输入/输出引脚端口功能的寄存器	544
19.4	运行停止模式	545
19.4.1	以单元为单位停止运行的情况	545
19.4.2	按通道停止运行的情况	546
19.5	3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信的运行	547
19.5.1	主控发送	548
19.5.2	主控接收	557
19.5.3	主控的发送和接收	565
19.5.4	从属发送	573
19.5.5	从属接收	581
19.5.6	从属的发送和接收	587
19.5.7	传送时钟频率的计算	596
19.5.8	在3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信过程中发生错误时的处理步骤	598
19.6	从属选择输入功能的时钟同步串行通信的运行	599
19.6.1	从属发送	602
19.6.2	从属接收	612
19.6.3	从属的发送和接收	619
19.6.4	传送时钟频率的计算	629
19.6.5	在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤	630
19.7	UART(UART0~UART2)通信的运行	631
19.7.1	UART发送	632
19.7.2	UART接收	641
19.7.3	波特率的计算	648
19.7.4	在UART(UART0~UART2)通信过程中发生错误时的处理步骤	652
19.8	LIN通信的运行	653
19.8.1	LIN发送	653
19.8.2	LIN接收	656
19.9	简易I ² C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信的运行	661
19.9.1	地址段发送	662
19.9.2	数据发送	667
19.9.3	数据接收	670
19.9.4	停止条件的产生	674
19.9.5	传送速率的计算	675
19.9.6	在简易I ² C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信过程中发生错误时的处理步骤	677

第20章 串行接口IICA 678

20.1	串行接口IICA的功能	678
20.2	串行接口IICA的结构	681
20.2.1	IICA移位寄存器n(IICAn)	681
20.2.2	从属地址寄存器n(SVAn)	682
20.2.3	SO锁存器	682
20.2.4	唤醒控制电路	682
20.2.5	串行时钟计数器	682
20.2.6	中断请求信号发生电路	682
20.2.7	串行时钟控制电路	682
20.2.8	串行时钟等待控制电路	682

20.2.9	应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路	683
20.2.10	数据保持时间校正电路	683
20.2.11	开始条件生成电路	683
20.2.12	停止条件生成电路	683
20.2.13	总线状态检测电路	683
20.3	控制串行接口IICA的寄存器	684
20.3.1	外围允许寄存器0(PER0)	685
20.3.2	IICA控制寄存器n0(IICCTLn0)	685
20.3.3	IICA状态寄存器n(IICSn)	690
20.3.4	IICA标志寄存器n(IICFn)	693
20.3.5	IICA控制寄存器n1(IICCTLn1)	695
20.3.6	IICA低电平宽度设置寄存器n(IICWLn)	697
20.3.7	IICA高电平宽度设置寄存器n(IICWHn)	697
20.3.8	端口模式寄存器x(PMx)	698
20.4	I2C总线模式的功能	699
20.4.1	引脚结构	699
20.4.2	通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法	700
20.5	I ² C总线的定义和控制方法	702
20.5.1	开始条件	703
20.5.2	地址	704
20.5.3	传送方向的指定	704
20.5.4	应答(ACK)	705
20.5.5	停止条件	706
20.5.6	等待	707
20.5.7	等待的解除方法	709
20.5.8	中断请求(INTIICAn)的产生时序和等待控制	710
20.5.9	地址匹配的检测方法	711
20.5.10	错误的检测	711
20.5.11	扩展码	712
20.5.12	仲裁	713
20.5.13	唤醒功能	715
20.5.14	通信预约	718
20.5.15	其他注意事项	722
20.5.16	通信运行	723
20.5.17	I ² C中断请求(INTIICAn)的产生时序	731
20.6	时序图	751
第21章	CAN 控制器	766
21.1	概要描述	766
21.1.1	特征	766
21.1.2	功能概述	767
21.1.3	配置	768
21.2	CAN协议	769
21.2.1	帧格式	769
21.2.2	帧类型	770
21.2.3	数据帧和远程帧	771
21.2.4	错误帧	778
21.2.5	过载帧	779
21.3	功能	780

21.3.1	总线优先级设定	780
21.3.2	位填充	780
21.3.3	多主设备	780
21.3.4	多点广播	780
21.3.5	CAN 睡眠模式/CAN停止模式功能	780
21.3.6	错误控制功能	781
21.3.7	波特率控制功能	786
21.4	与目标系统的连接	789
21.5	CAN控制器的内部寄存器	790
21.5.1	CAN控制器配置	790
21.5.2	寄存器访问类型	792
21.5.3	寄存器位配置	801
21.6	位设置/清除功能	805
21.7	控制寄存器	807
21.7.1	外围时钟选择寄存器 (PER0)	807
21.7.2	CAN全局模块控制寄存器(C0GMCTRL)	808
21.7.3	CAN全局模块时钟选择寄存器(C0GMCS)	810
21.7.4	CAN全局自动块传输控制寄存器 (C0GMABT)	811
21.7.5	CAN全局自动块传输延时设置寄存器(C0GMABTD)	813
21.7.6	CAN模块屏蔽寄存器(C0MASKaL, C0MASKaH) (a = 1, 2, 3, or4)	814
21.7.7	CAN模块控制寄存器(C0CTRL)	816
21.7.8	CAN模块上次错误代码寄存器 (C0LEC)	820
21.7.9	CAN模块信息寄存器(C0INFO)	821
21.7.10	CAN模块错误计数器寄存器(C0ERC)	822
21.7.11	CAN模块中断使能寄存器(C0IE)	823
21.7.12	CAN模块中断状态寄存器(C0INTS)	825
21.7.13	CAN模块位率缩放寄存器(C0BRP)	826
21.7.14	CAN模块位率寄存器(C0BTR)	827
21.7.15	CAN模块上次输入指针寄存器(C0LIPT)	829
21.7.16	CAN模块接收历史列表寄存器 (C0RGPT)	830
21.7.17	CAN模块上次输出指针寄存器(C0LOPT)	832
21.7.18	CAN模块发送历史列表寄存器(C0TGPT)	833
21.7.19	CAN模块时间戳寄存器(C0TS)	835
21.7.20	CAN报文数据字节寄存器 (C0MDBxm) (x = 0 到7), (C0MDBzm) (z = 01, 23, 45,67)	837
21.7.21	CAN报文数据长度寄存器m (C0MDLCm)	839
21.7.22	CAN报文配置寄存器 (C0MCONFm)	840
21.7.23	CAN报文ID寄存器m (C0MIDLm and C0MIDHm)	842
21.7.24	CAN报文控制寄存器m (C0MCTRLm)	843
21.7.25	串行通讯引脚选择寄存器1(PIOR3)	846
21.7.26	端口模式寄存器0, 5 (PM0, PM5)	846
21.8	CAN控制器初始化	847
21.8.1	CAN模块初始化	847
21.8.2	报文缓存的初始化	847
21.8.3	重新定义报文缓存	847
21.8.4	从初始化模式过渡到操作模式	849
21.8.5	重设置CAN模块错误计数器C0ERC	849
21.9	报文接收	850
21.9.1	报文接收	850
21.9.2	接收数据读	851

21.9.3	接收历史列表功能.....	852
21.9.4	屏蔽功能.....	854
21.9.5	多缓冲区接收块功能.....	855
21.9.6	远程帧接收.....	856
21.10	报文发送.....	857
21.10.1	报文发送.....	857
21.10.2	发送历史列表功能.....	859
21.10.3	自动块传输 (ABT).....	861
21.10.4	传输中止处理.....	863
21.10.5	远程帧传输.....	863
21.11	省电模式.....	864
21.11.1	CAN睡眠模式.....	864
21.11.2	CAN停止模式.....	866
21.11.3	省电模式举例.....	867
21.12	中断功能.....	868
21.13	诊断功能和特殊操作模式.....	869
21.13.1	仅接收模式.....	869
21.13.2	单次模式.....	870
21.13.3	自检模式.....	871
21.13.4	操作模式下的接收/发送操作.....	872
21.14	时间戳功能.....	873
21.14.1	时间戳功能.....	873
21.15	波特率设置.....	875
21.15.1	波特率设置.....	875
21.15.2	波特率设置的代表性示例.....	879
21.16	CAN控制器的操作.....	883
第22章	IrDA.....	908
22.1	IrDA的功能.....	908
22.2	控制IrDA的寄存器.....	909
22.2.1	外围允许寄存器0(PER0).....	909
22.2.2	IrDA控制寄存器(IRCR).....	910
22.3	IrDA的运行.....	911
22.3.1	IrDA通信的操作步骤.....	911
22.3.2	发送.....	912
22.3.3	接收.....	912
22.3.4	高电平脉宽的选择.....	913
22.4	使用IrDA时的注意事项.....	914
第23章	增强型DMA.....	915
23.1	DMA的功能.....	915
23.2	DMA的结构.....	917
23.3	控制DMA的寄存器.....	918
23.3.1	DMA控制数据区和DMA向量表的分配.....	919
23.3.2	控制数据的分配.....	920
23.3.3	向量表.....	922
23.3.4	外围允许寄存器1(PER1).....	924
23.3.5	DMA控制寄存器j(DMACRj)(j=0~39).....	925
23.3.6	DMA块大小寄存器j(DMBLSj)(j=0~39).....	926
23.3.7	DMA传送次数寄存器j(DMACTj)(j=0~39).....	927

23.3.8	DMA传送次数重加载寄存器j(DMRLDj)(j=0~39)	928
23.3.9	DMA源地址寄存器j(DMSARj)(j=0~39).....	929
23.3.10	DMA目标地址寄存器j(DMDARj)(j=0~39)	929
23.3.11	DMA启动允许寄存器i(DMAENi)(i=0~4)	930
23.3.12	DMA基址寄存器(DMABAR).....	932
23.4	DMA的运行	933
23.4.1	启动源	933
23.4.2	正常模式	934
23.4.3	重复模式	937
23.4.4	链传送	941
23.5	使用DMA时的注意事项	943
23.5.1	DMA控制数据和向量表的设置	943
23.5.2	DMA控制数据区和DMA向量表区的分配	943
23.5.3	DMA的执行时钟数	944
23.5.4	DMA的响应时间	945
23.5.5	DMA的启动源	945
23.5.6	待机模式中的运行	946
第24章	联动控制器(EVENTC).....	947
24.1	EVENTC的功能	947
24.2	EVENTC的结构	947
24.3	控制寄存器	948
24.3.1	输出目标选择寄存器n(ELSELRn)(n=00~21)	949
24.4	EVENTC的运行	952
第25章	中断功能	954
25.1	中断功能的种类	954
25.2	中断源和结构	955
25.3	控制中断功能的寄存器	961
25.3.1	中断请求标志寄存器(IF00~IF31)	961
25.3.2	中断屏蔽标志寄存器(MK00~MK31).....	962
25.3.3	外部中断上升沿允许寄存器(EGP0、EGP1)、外部中断下降沿允许寄存器(EGN0、EGN1).....	965
25.4	中断处理的操作	967
25.4.1	可屏蔽中断请求的接受	967
25.4.2	不可屏蔽中断请求的接受	967
第26章	键中断功能.....	968
26.1	键中断的功能	968
26.2	键中断的结构	969
26.3	控制键中断的寄存器	970
26.3.1	键返回模式寄存器(KRM)	970
26.3.2	端口模式寄存器(PMx).....	970
第27章	待机功能	971
27.1	待机功能	971
27.2	睡眠模式	972
27.2.1	睡眠模式的设置	972
27.2.2	睡眠模式的解除	975
27.3	深度睡眠模式	976
27.3.1	深度睡眠模式的设置	976
27.3.2	深度睡眠模式的解除	979

第28章	复位功能	980
28.1	确认复位源的寄存器	985
28.1.1	复位控制标志寄存器(RESF)	985
第29章	上电复位电路	987
29.1	上电复位电路的功能	987
29.2	上电复位电路的结构	988
29.3	上电复位电路的运行	989
第30章	电压检测电路	992
30.1	电压检测电路的功能	992
30.2	电压检测电路的结构	993
30.3	控制电压检测电路的寄存器	994
30.3.1	电压检测寄存器(LVIM)	994
30.3.2	电压检测电平寄存器(LVIS)	995
30.4	电压检测电路的运行	998
30.4.1	用作复位模式时的设置	998
30.4.2	用作中断模式时的设置	1000
30.4.3	用作中断&复位模式时的设置	1002
30.5	电压检测电路的注意事项	1007
第31章	安全功能	1009
31.1	安全功能的概要	1009
31.2	安全功能使用的寄存器	1010
31.3	安全功能的运行	1010
31.3.1	闪存CRC运算功能(高速CRC)	1010
31.3.1.1	闪存CRC控制寄存器(CRC0CTL)	1011
31.3.1.2	闪存CRC运算结果寄存器(PGCRCL)	1012
31.3.2	CRC运算功能(通用CRC)	1014
31.3.2.1	CRC输入寄存器(CRCIN)	1014
31.3.2.2	CRC数据寄存器(CRCD)	1015
31.3.3	RAM奇偶校验错误检测功能	1016
31.3.3.1	RAM奇偶校验错误控制寄存器(RPECTL)	1016
31.3.4	SFR保护功能	1018
31.3.4.1	SFR保护控制寄存器(SFRGD)	1018
31.3.5	频率检测功能	1019
31.3.5.1	定时器输入选择寄存器0(TIS0)	1019
31.3.6	A/D测试功能	1020
31.3.6.1	A/D测试寄存器(ADTES)	1022
31.3.6.2	模拟输入通道指定寄存器(ADS)	1022
31.3.7	输入/输出引脚的数字输出信号电平检测功能	1023
31.3.7.1	端口模式选择寄存器(PMS)	1023
31.3.8	产品唯一身份标识寄存器	1024
第32章	温度传感器	1025
32.1	温度传感器的功能	1025
32.2	温度传感器的寄存器	1025
32.2.1	温度传感器校准数据寄存器 TSN25	1025
32.2.2	温度传感器校准数据寄存器 TSN125	1025
32.3	温度传感器的使用说明	1026
32.3.1	温度传感器的使用原理	1026

32.3.2	温度传感器的使用方法	1027
第33章	选项字节	1028
33.1	选项字节的功能.....	1028
33.1.1	用户选项字节(000C0H~000C2H/010C0H~010C2H).....	1028
33.1.2	闪存数据保护选项字节(000C3H/010C3H, 500004H~500005H).....	1030
33.2	用户选项字节的格式	1031
33.3	闪存数据保护选项字节的格式	1037
第34章	FLASH控制.....	1038
34.1	FLASH控制功能描述.....	1038
34.2	FLASH存储器结构	1038
34.3	控制FLASH的寄存器.....	1039
34.3.1	Flash写保护寄存器(FLPROT)	1039
34.3.2	FLASH 操作控制寄存器(FLOPMD1,FLOPMD2)	1040
34.3.3	Flash擦除控制寄存器(FLERMD)	1040
34.3.4	Flash 状态寄存器(FLSTS).....	1041
34.3.5	Flash 全片擦除时间控制寄存器(FLCERCNT).....	1041
34.3.6	Flash 页擦除时间控制寄存器(FLSERCNT)	1042
34.3.7	Flash 写入时间控制寄存器(FLPROCNT).....	1043
34.4	FLASH操作方法.....	1044
34.4.1	页擦除(sector erase)	1044
34.4.2	全片擦除(chip erase).....	1045
34.4.3	编程(word program).....	1045
34.5	闪存读取	1046
34.6	FLASH操作的注意事项	1046
附录	修订记录.....	1047

第1章 CPU

1.1 概述

本章节简单介绍本制品搭载的ARM Cortex-M0+内核的特性及调试特性，详情请参考ARM相关文档。

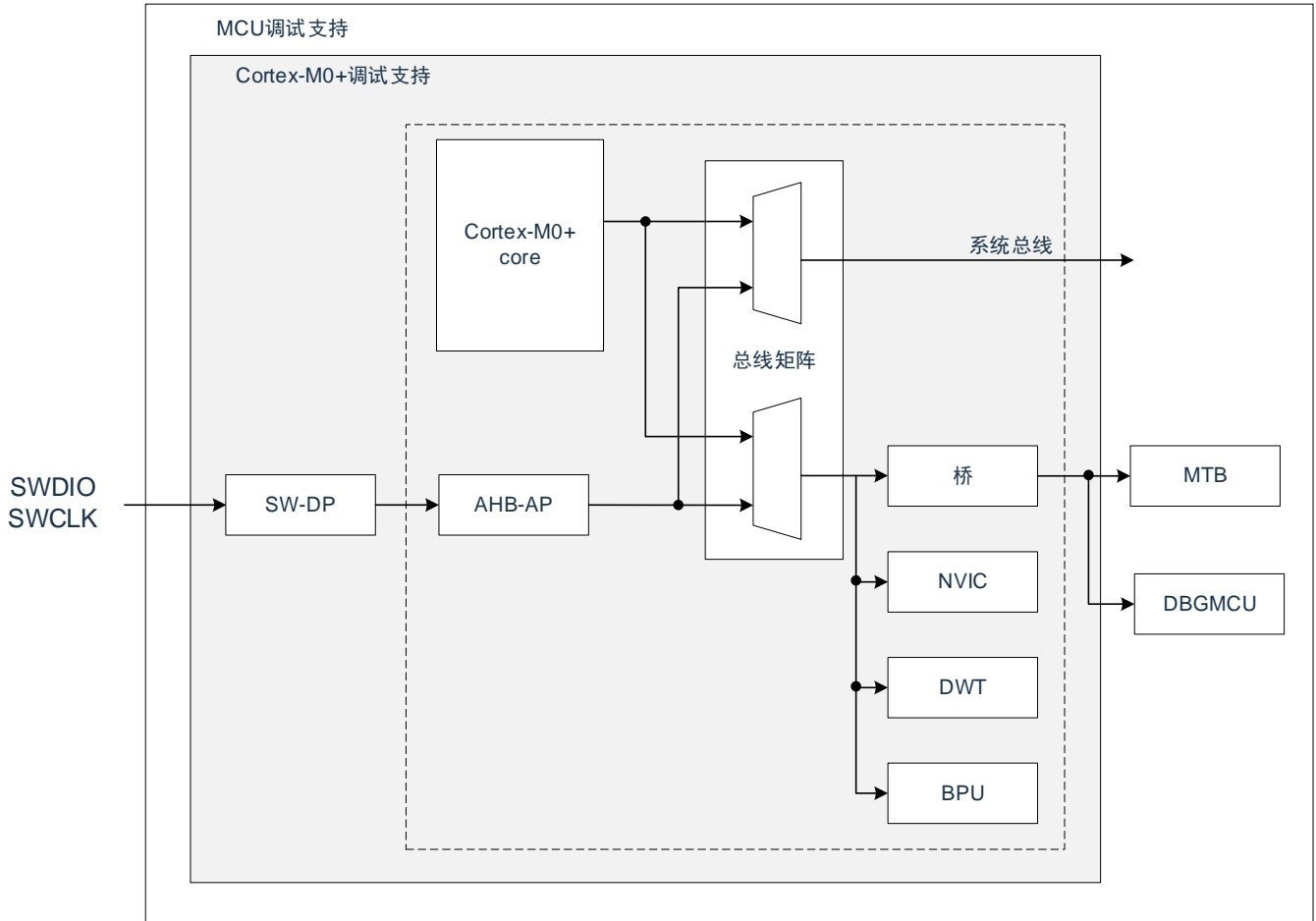
1.2 Cortex-M0+内核特性

- ARM Cortex-M0+处理器是32位RISC内核，采用2级流水线，支持特权模式和用户模式
- 存储器保护单元(MPU)支持8个独立的分区(region)保护
- 单周期硬件乘法器
- 嵌套向量中断控制器(NVIC)
 - 1个不可屏蔽中断(NMI)
 - 支持32个可屏蔽中断请求(IRQ)
 - 4个中断优先级
- 系统定时器SysTick是一个24位倒计时定时器，可选择F_{CLK}或F_{IL}计数时钟
- 向量表偏移寄存器(VTOR)
 - 软件可以写VTOR将向量表起始地址重新定位到不同的位置
 - 该寄存器的默认值为0x0000_0000,低8位写忽略，读为零，也就是偏移量256字节对齐

1.3 调试特性

- 2线SWD调试接口
- 支持暂停、恢复和单步执行程序
- 访问处理器的内核寄存器和特殊功能寄存器
- 4个硬件断点(BPU)
- 无限个软件断点(BKPT指令)
- 2个数据观察点(DWT)
- 内核执行的时候访问存储器
- 微跟踪缓存(MTB)提供简单的指令执行跟踪方案
 - MTB与用户SRAM(0x2000_0000-0x2000_2FFF区域)共用
 - MTB控制寄存器的基地址为0x4001_9000

图1-1: Cortex-M0+的调试框图



注意: SWD在Deep Sleep模式下不能工作, 请在active和sleep模式下进行调试操作。

1.4 SWD接口引脚

本产品的2个GPIO可用作SWD接口引脚，这些引脚在所有的封装里都存在。

表1-1: SWD调试端口引脚

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	P137
SWDIO	串行数据输入/输出	P40

不使用SWD功能时，可以通过设置debug停止控制寄存器(DBGSTOPCR)来禁用SWD。

Bit No.	31	30	29	28	27	26	25	24
DBGSTOPCR	-	-	-	-	-	-	-	SWDIS
默认值	0	0	0	0	0	0	0	0
Bit No.	23	22	21	20	19	18	17	16
DBGSTOPCR	-	-	-	-	-	-	-	-
默认值	0	0	0	0	0	0	0	0
Bit No.	15	14	13	12	11	10	9	8
DBGSTOPCR	-	-	-	-	-	-	-	-
默认值	0	0	0	0	0	0	0	0
Bit No.	7	6	5	4	3	2	1	0
DBGSTOPCR	-	-	-	-	-	-	FRZEN1	FRZEN0
默认值	0	0	0	0	0	0	0	0

SWDIS	SWD 调试接口禁用
0	SWD 调试接口使能。在连接调试器的状态下，P40 不能用作 GPIO(因为此时该 IOBUF 的 ENO 和 DOUT 由调试器控制)
1	SWD 调试接口禁用。P40 可用作 GPIO

FRZEN0	在调试器连接的状态下，并且 CPU 处于调试状态时(HALTED=1)，定时器系周边模块动作/停止 ^{注1}
0	周边动作
1	周边停止

FRZEN1	在调试器连接的状态下，并且 CPU 处于调试状态时(HALTED=1)，通信系周边模块动作/停止 ^{注2}
0	周边动作
1	周边停止

注1: 本制品的定时器系周边模块包括: 通用定时器单元Timer4, 定时器A, 定时器B, 定时器C和定时器M

注2: 本制品的通信系周边模块包括: 通信串行通信单元, 串行IICA

1.5 ARM参考文档

Cortex®-M0+ 内核中内置的调试功能是ARM®CoreSight设计套件的一部分。相关文档请参考：

- Cortex®-M0+技术参考手册(TRM)
- ARM®调试接口V5
- ARM® CoreSight设计套件版本r1p1技术参考手册
- ARM® CoreSight™ MTB-M0+ Technical Reference Manual

第2章 引脚功能

2.1 端口功能

参见各产品系列的数据手册。

2.2 端口复用功能

参见各产品系列的数据手册。

2.3 控制端口功能的寄存器

通过以下寄存器控制端口。

- 端口模式寄存器(PMxx)
- 端口寄存器(Pxx)
- 上拉电阻选择寄存器(PUxx)
- 端口输入模式寄存器(PIMx)
- 端口输出模式寄存器(POMx)
- 端口模式控制寄存器(PMCxx)
- 外围I/O重定向寄存器(PIORx)
- 全局数字输入禁止寄存器(GDIDIS)

注意：分配的寄存器和位因产品而不同。有关各产品分配的寄存器和位，请参照表2-1。必须给未分配的位设置初始值。

表2-1：各产品分配的PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx寄存器及其位

端口		位名						64 引脚	48 引脚	44 引脚	40 引脚	32 引脚	24 引脚
		PMxx 寄存器	Pxx 寄存器	PUxx 寄存器	PIMxx 寄存器	POMxx 寄存器	PMCxx 寄存器						
端口0	0	PM00	P00	PU00	—	POM00	PMC00 ^{注1}	○	○	○	○	○	○
	1	PM01	P01	PU01	PIM01	—	PMC01 ^{注1}	○	○	○	○	○	○
	2	PM02	P02	PU02	—	POM02	PMC02	○	—	—	—	—	—
	3	PM03	P03	PU03	PIM03	POM03	PMC03	○	—	—	—	—	—
	4	PM04	P04	PU04	PIM04	POM04	PMC04	○	—	—	—	—	—
	5	PM05	P05	PU05	—	—	—	○	—	—	—	—	—
	6	PM06	P06	PU06	—	—	—	○	—	—	—	—	—
端口1	0	PM10	P10	PU10	PIM10	POM10	PMC10	○	○	○	○	○	○
	1	PM11	P11	PU11	—	POM11	PMC11	○	○	○	○	○	○
	2	PM12	P12	PU12	—	—	PMC12 ^{注2}	○	○	○	○	○	○
	3	PM13	P13	PU13	—	POM13	PMC13 ^{注2}	○	○	○	○	○	○
	4	PM14	P14	PU14	PIM14	POM14	PMC14 ^{注2}	○	○	○	○	○	○
	5	PM15	P15	PU15	PIM15	POM15	—	○	○	○	○	○	○
	6	PM16	P16	PU16	PIM16	—	—	○	○	○	○	○	—
	7	PM17	P17	PU17	PIM17	POM17	—	○	○	○	○	○	—
端口2	0	PM20	P20	—	—	—	PMC20	○	○	○	○	○	○
	1	PM21	P21	—	—	—	PMC21	○	○	○	○	○	○
	2	PM22	P22	—	—	—	PMC22	○	○	○	○	○	○
	3	PM23	P23	—	—	—	PMC23	○	○	○	○	○	—
	4	PM24	P24	—	—	—	PMC24	○	○	○	○	—	—
	5	PM25	P25	—	—	—	PMC25	○	○	○	○	—	—
	6	PM26	P26	—	—	—	PMC26	○	○	○	—	—	—
	7	PM27	P27	—	—	—	PMC27	○	○	○	—	—	—

注1：只限于24~48引脚的产品。

注2：只限于24~64引脚的产品。

表2-2: 各产品分配的PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx寄存器及其位

端口		位名						64 引脚	48 引脚	44 引脚	40 引脚	32 引脚	24 引脚
		PMxx 寄存器	Pxx 寄存器	PUxx 寄存器	PIMxx 寄存器	POMxx 寄存器	PMCxx 寄存器						
端口3	0	PM30	P30	PU30	PIM30	POM30	—	○	○	○	○	○	—
	1	PM31	P31	PU31	PIM30 ^{注1}	POM31 ^{注1}	—	○	○	○	○	○	—
端口4	0	PM40	P40	PU40	—	—	—	○	○	○	○	○	○
	1	PM41	P41	PU41	—	—	—	○	○	○	—	—	—
	2	PM42	P42	PU42	—	—	—	○	—	—	—	—	—
	3	PM43	P43	PU43	—	—	—	○	—	—	—	—	—
端口5	0	PM50	P50	PU50	PIM50	POM50	—	○	○	○	○	○	○
	1	PM51	P51	PU51	—	POM51	—	○	○	○	○	○	○
	2	PM52	P52	PU52	—	—	—	○	—	—	—	—	—
	3	PM53	P53	PU53	—	—	—	○	—	—	—	—	—
	4	PM54	P54	PU54	—	—	—	○	—	—	—	—	—
	5	PM55	P55	PU55	PIM55	POM55	—	○	—	—	—	—	—
端口6	0	PM60	P60	—	—	—	—	○	○	○	○	—	—
	1	PM61	P61	—	—	—	—	○	○	○	○	—	—
	2	PM62	P62	—	—	—	—	○	○	○	—	—	—
	3	PM63	P63	—	—	—	—	○	○	○	—	—	—
端口7	0	PM70	P70	PU70	—	—	—	○	○	○	○	○	—
	1	PM71	P71	PU71	—	POM71	—	○	○	○	—	—	—
	2	PM72	P72	PU72	—	—	—	○	○	○	○	○	○
	3	PM73	P73	PU73	—	—	—	○	○	○	○	○	○
	4	PM74	P74	PU74	PIM74 ^{注1}	POM74	—	○	○	—	○	○	—
	5	PM75	P75	PU75	—	—	—	○	○	—	○	—	—
	6	PM76	P76	PU76	—	—	—	○	—	—	—	—	—
	7	PM77	P77	PU77	—	—	—	○	—	—	—	—	—
端口12	0	PM120	P120	PU120	—	—	PMC120	○	○	○	○	○	—
	1	—	P121	—	—	—	—	○	○	○	○	○	○
	2	—	P122	—	—	—	—	○	○	○	○	○	○
	3	—	P123	—	—	—	—	○	○	○	○	—	—
	4	—	P124	—	—	—	—	○	○	○	○	—	—
端口13	0	—	P130	—	—	—	—	○	○	—	—	—	—
	6	PM136	P136	PU136	—	—	—	○	○	—	○	○	○
	7	PM137	P137	PU137	—	—	—	○	○	○	○	○	○
端口14	0	PM140	P140	PU140	—	—	—	○	○	—	○	—	—
	1	PM141	P141	PU141	—	—	—	○	—	—	—	—	—
	6	PM146	P146	PU146	—	—	PMC146	○	○	○	—	—	—
	7	PM147	P147	PU147	—	—	PMC147	○	○	○	○	○	○

注1: 只限于32引脚的产品。

2.3.1 端口模式寄存器(PMxx)

这是以位为单位设置端口输入/输出的寄存器。在产生复位信号后，这些寄存器的值变为“FFH”。当将端口引脚用作复用功能的引脚时，必须参照“2.5 使用复用功能时的寄存器设置”进行设置。

寄存器地址=基址+偏址；PM寄存器的基址为0x40040000，偏址见下图。

图2-1：端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	0x320	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	0x321	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	0x322	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	0x323	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	0x324	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	0x325	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	0x326	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	0x327	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	0x32C	FFH	R/W
PM13	PM137	PM136	1	1	1	1	1	PM130	0x32D	FEH	R/W
PM14	PM147	PM146	1	1	1	1	PM141	PM140	0x32E	FFH	R/W

PMmn	Pmn引脚的输入/输出模式的选择(m=0~7、12~14, n=0~7)
0	输出模式(用作输出端口(输出缓冲器ON))
1	输入模式(用作输入端口(输出缓冲器OFF))

注意：必须给未分配的位设置初始值。

2.3.2 端口寄存器(Pxx)

这是以位为单位设置端口输出锁存器的值的寄存器。在输入模式中读引脚电平，而在输出模式中读端口的输出锁存器的值。在产生复位信号后，这些寄存器的值变为“00H”。

寄存器地址=基址+偏址；端口寄存器的基址为0x40040000，偏址见下图。

图2-2：端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	0x300	00H(输出锁存器)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	0x301	00H(输出锁存器)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	0x302	00H(输出锁存器)	R/W
P3	0	0	0	0	0	0	P31	P30	0x303	00H(输出锁存器)	R/W
P4	0	0	0	0	P43	P42	P41	P40	0x304	00H(输出锁存器)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	0x305	00H(输出锁存器)	R/W
P6	0	0	0	0	P63	P62	P61	P60	0x306	00H(输出锁存器)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	0x307	00H(输出锁存器)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	0x30C	不定值	R/W ^{注1}
P13	P137	P136	0	0	0	0	0	P130	0x30D	00H(输出锁存器)	R/W
P14	P147	P146	0	0	0	0	P141	P140	0x30E	00H(输出锁存器)	R/W

Pmn	m=0~7、12~14, n=0~7	
	输出数据的控制(输出模式)	输入数据的读取(输入模式)
0	输出“0”。	输入低电平。
1	输出“1”。	输入高电平。

注意：

1. P121~P124是只读位。
2. 必须给未分配的位设置初始值。

2.3.3 上拉电阻选择寄存器(PUxx)

内部上拉电阻的选择寄存器。只能对通过上拉电阻选择寄存器指定使用内部上拉电阻的引脚并且POMmn位为“0”而且设置为输入模式(PMmn=1)的位，以位为单位使用内部上拉电阻。对于设置为输出模式的位，与上拉电阻选择寄存器的设置无关，不连接内部上拉电阻。当用作复用功能的输出引脚或者设置为模拟功能时也相同。

在产生复位信号后，这些寄存器的值变为“00H”(只有PU4为“01H”，PU13为“80H”)。

寄存器地址=基址+偏址；PU寄存器的基址为0x40040000，偏址见下图。

图2-3：上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	0x030	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	0x031	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	0x033	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	0x034	01H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	0x035	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	0x037	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	0x03C	00H	R/W
PU13	PU137	PU136	0	0	0	0	0	0	0x03D	80H	R/W
PU14	PU147	PU146	0	0	0	0	PU141	PU140	0x03E	00H	R/W

PUmn	Pmn引脚的内部上拉电阻的选择(m=0、1、3~5、7、12~14, n=0~7)
0	不连接内部上拉电阻。
1	连接内部上拉电阻。

注意：必须给未分配的位设置初始值。

2.3.4 端口输入模式寄存器(PIMxx)

这是以位为单位设置输入缓冲器的寄存器。在和不同电位的外部设备进行串行通信时，能选择TTL输入缓冲器。在产生复位信号后，这些寄存器的值变为“00H”。

寄存器地址=基址+偏址；PIM寄存器的基址为0x40040000，偏址见下图。

图2-4：端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PIM0	0	0	0	PIM04	PIM03	0	PIM01	0	0x040	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	0	0	0	PIM10	0x041	00H	R/W
PIM3	0	0	0	0	0	0	PIM31 ^{注1}	PIM30	0x043	00H	R/W
PIM5	0	0	PIM55	0	0	0	0	PIM50	0x045	00H	R/W
PIM7	0	0	0	PIM74 ^{注1}	0	0	0	0	0x047	00H	R/W

PIMmn	Pmn引脚的输入缓冲器的选择(m=0、1、3、5、7, n=0~7)
0	施密特输入缓冲器
1	TTL输入缓冲器

注1：只限于32引脚的产品。

注2：必须给未分配的位设置初始值。

2.3.5 端口输出模式寄存器(POMxx)

这是以位为单位设置输出模式的寄存器。在和不同电位的外部设备进行串行通信以及和同电位的外部设备进行简易I²C通信时，能给SDAxx引脚选择N沟道漏极开路输出模式。

在产生复位信号后，这些寄存器的值变为“00H”。

寄存器地址=基址+偏址；POM寄存器的基址为0x40040000，偏址见下图。

注意：对于设置N沟道漏极开路输出模式(POMmn=1)的位，不连接内部上拉电阻。

图2-5：端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
POM0	0	0	0	POM04	POM03	POM02	0	POM00	0x050	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	0	POM11	POM10	0x051	00H	R/W
POM3	0	0	0	0	0	0	POM31 _{注1}	POM30	0x053	00H	R/W
POM5	0	0	POM55	0	0	0	POM51	POM50	0x055	00H	R/W
POM7	0	0	0	POM74	0	0	POM71	0	0x057	00H	R/W

POMmn	Pmn引脚的输出模式的选择(m=0、1、3、5、7, n=0~7)
0	通常的输出模式
1	N沟道漏极开路输出模式

注1：只限于32引脚的产品。

注2：必须给未分配的位设置初始值。

2.3.6 端口模式控制寄存器(PMCxx)

PMC寄存器以位为单位设置的数字输入/输出或者模拟输入。

在产生复位信号后，这些寄存器的值变为“FFH”。

寄存器地址=基址+偏址；PMC寄存器的基址为0x40040000，偏址见下图。

图2-6: 端口模式控制寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W								
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01 _{注1}	PMC00 _{注1}	0x060	FFH	R/W								
PMC1	1	1	1	1	1	1	PMC11	PMC10	0x061	FFH	R/W								
PMC2	<table border="1" style="width:100%; text-align:center;"> <tr> <td>PMC27</td><td>PMC26</td><td>PMC25</td><td>PMC24</td><td>PMC23</td><td>PMC22</td><td>PMC21</td><td>PMC20</td> </tr> </table>								PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	0x062	FFH	R/W
PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20												
PMC12	1	1	1	1	1	1	1	PMC120	0x06C	FFH	R/W								
PMC14	<table border="1" style="width:100%; text-align:center;"> <tr> <td>PMC147</td><td>PMC146</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td> </tr> </table>								PMC147	PMC146	1	1	1	1	1	1	0x06E	FFH	R/W
PMC147	PMC146	1	1	1	1	1	1												

PMCmn	Pmn引脚的数字输入/输出或者模拟输入的选择(m=0、1、2、12、14, n=0~7)
0	数字输入/输出(模拟输入以外的复用功能)
1	模拟输入

注1：只限于32~48引脚的产品。

注2：必须给未分配的位设置初始值。

2.3.7 外围I/O重定向寄存器0(PIOR0)

这是设置允许或者禁止外围I/O重定向功能的寄存器0。外围I/O重定向功能切换被分配了复用功能的端口。在产生复位信号后，此寄存器的值变为“00H”。

图2-7：外围I/O重定向寄存器0(PIOR0)的格式

地址：0x40040877 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PIOR0	PIOR07	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

PIOR0	功能	64pin		48pin		44pin		40pin		32pin		24pin	
		设定值		设定值		设定值		设定值		设定值		设定值	
		0	1	0	1	0	1	0	1	0	1	0	1
bit7(PIOR07)	INTP8	P74/P42	P00	P74	P00	必须设置为0(初始值)							
	INTP10	P76/P05	P01										
	INTP11	P77/P06	P20										
bit6(PIOR06)	RxD2	必须设置为0(初始值)		P14	P14	P14	P14	P14	P14	P14	P14	P14	P14
	TxD2			P13	P10	P13	P10	P13	P10	P13	P10	P13	P10
	SCL20			P15	-	P15	-	P15	-	P15	-	P15	-
	SDA20			P14	-	P14	-	P14	-	P14	-	P14	-
	SDI20			P14	-	P14	-	P14	-	P14	-	P14	-
	SDO20			P13	-	P13	-	P13	-	P13	-	P13	-
bit5(PIOR05)	RXD1	必须设置为0(初始值)		P01	P73	P01	P73	P01	P73	P01	P73	P01	P73
	TXD1			P00	P72	P00	P72	P00	P72	P00	P72	P00	P72
bit4(PIOR04)	CLKBUZ1	P141	P55	必须设置为0(初始值)									
	INTP5	P16	P12										
bit3(PIOR03)	CLKBUZ0	P140	P31	P140	P31	-	P31	P140	P31				
bit2(PIOR02)	SCLA0	P60	P14	P60	P14	P60	P14	P60	P14	P31	P14	P01	P14
	SDAA0	P61	P15	P61	P15	P61	P15	P61	P15	P74	P15	P00	P15
bit1(PIOR01)	INTP10	P76	P05	P01	P01	P01	P01	P01	P01	P01	P01	必须设置为0(初始值)	
	INTP11	P77	P06	P20	P20	P20	P20	P20	P20	P20	P20		
	RXD2	P14	P76	必须设置为0(初始值)									
	TxD2	P13	P77										
	SCL20	P15	-										
	SDA20	P14	-										
	SDI20	P14	-										
	SDO20	P13	-										
	SCLK20	P15	-										
	TXD0	P51	P17	P51	P17	P51	P17	P51	P17	P51	P17		
	RXD0	P50	P16	P50	P16	P50	P16	P50	P16	P50	P16		
	SCL00	P30	-	P30	-	P30	-	P30	-	P30	-		
	SDA00	P50	-	P50	-	P50	-	P50	-	P50	-		
SDO00	P51	P17	P51	-	P51	-	P51	-	P51	-			
SDI00	P50	P16	P50	-	P50	-	P50	-	P50	-			
SCLK00	P30	P55	P30	-	P30	-	P30	-	P30	-			
bit0(PIOR00)	INTP1	P50	P52	必须设置为0(初始值)									
	INTP2	P51	P53										

	INTP3	P30	P54		
	INTP4	P31	P55		
	INTP8	P74	P42		
	INTP9	P75	P43		

2.3.8 外围I/O重定向寄存器1(PIOR1)

这是设置允许或者禁止外围I/O重定向功能寄存器1。外围I/O重定向功能切换被分配了复用功能的端口。在产生复位信号后，此寄存器的值变为“00H”。

图2-8：外围I/O重定向寄存器1(PIOR1)的格式

地址：0x40040879H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PIOR1	0	0	0	0	PIOR13	PIOR12	PIOR11	PIOR10

PIOR13	PIOR12	定时器ATAO引脚的选择
0	0	和P30复用(32-64引脚产品)，和P72复用（24引脚产品）。
0	1	和P50复用。
1	0	和P00复用。
1	1	禁止设置。

PIOR11	PIOR10	定时器ATAIO引脚的选择
0	0	和P01复用。
0	1	和P31复用(32-64引脚产品)，和P73复用（24引脚产品）。
1	0	和P41复用(只限于48、64引脚的产品)。其它引脚产品禁止设置。
1	1	和P06复用(只限于64引脚的产品)。其它引脚产品禁止设置。

2.3.9 外围I/O重定向寄存器2(PIOR2)

这是设置允许或者禁止外围I/O重定向功能寄存器2。外围I/O重定向功能切换被分配了复用功能的端口。在产生复位信号后，此寄存器的值变为“00H”。

图2-9：外围I/O重定向寄存器2(PIOR2)的格式

地址：0x40040875 复位后：00H R/W

符号 7 6 5 4 3 2 1 0

PIOR2	PIOR27	PIOR26	PIOR25	PIOR24	PIOR23	PIOR22	PIOR21	PIOR20
-------	--------	--------	--------	--------	--------	--------	--------	--------

PIOR2	功能	64pin		48pin		44pin		40pin		32pin		24pin	
		设定值		设定值		设定值		设定值		设定值		设定值	
		0	1	0	1	0	1	0	1	0	1	0	1
bit7(PIOR27)	TMIOC0	x	x	x	x	x	x	x	x	x	x	P16	P13
bit6(PIOR26)	TMIOD0	P15	P17	P15	P17	P15	P17	P15	P17	P15	P17	-	-
bit5(PIOR25)	TMIOD1	P11	P51	P11	P51	P11	P51	P11	P51	P11	P51	P11	P51
bit4(PIOR24)	TMIOC1	P13	P50	P13	P50	P13	P50	P13	P50	P13	P50	P13	P50
bit3(PIOR23)	TMIOB1	P10	P30	P10	P30	P10	P30	P10	P30	P10	P30	-	-
bit2(PIOR22)	TMIOA1	P12	P16	P12	P16	P12	P16	P12	P16	P12	P16	-	-
bit1(PIOR21)	VCOUT1	P31	P70	P31	P70	P31	P70	P31	P70	P31	P70	P147	P72
bit0(PIOR20)	VCOUT0	P120	P71	P120	P71	P120	P71	P120	P70	P120	P73	P12	P73

注：x表示该位必须设置为初始值0。

2.4 未使用引脚的处理

各未使用引脚的处理如表2-3所示。

表2-3: 各未使用引脚的处理

引脚名	输入/输出	未使用时的推荐连接方法
P00~P06	输入/输出	输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
P10~P17		
P20~P27		输入: 单独通过电阻连接V _{DD} 或者V _{SS} 。 输出: 置为开路。
P30、P31		输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
P40		输入: 单独通过电阻连接EV _{DD} 或者置为开路。 输出: 置为开路。
P41~P43		输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
P50~P55		
P60~P63		输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 将端口的输出锁存器置“0”并且置为开路, 或者将端口的输出锁存器置“1”并且单独通过电阻连接EV _{DD} 或者EV _{SS} 。
P70~P77		输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
P120		
P121~P124	输入	单独通过电阻连接V _{DD} 或者V _{SS} 。
P130	输出	置为开路。
P136	输入/输出	输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
P137		输入: 单独通过电阻连接V _{DD} 或者置为开路。 输出: 置为开路。
P140~P147		输入: 单独通过电阻连接EV _{DD} 或者EV _{SS} 。 输出: 置为开路。
RESETB	输入	直接或者通过电阻连接V _{DD} 。

备注: 对于没有EV_{DD}、EV_{SS}引脚的产品, 必须将EV_{DD}替换为V_{DD}并且将EV_{SS}替换为V_{SS}。

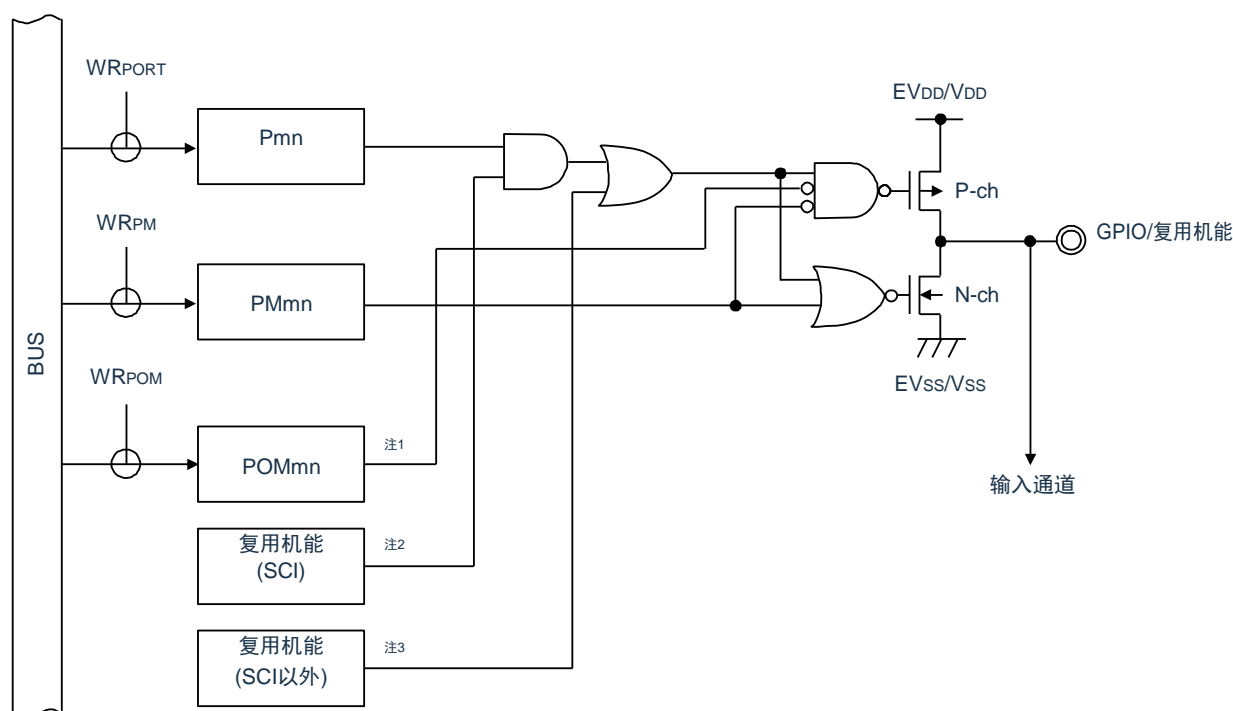
2.5 使用复用功能时的寄存器设置

2.5.1 使用复用功能时的基本思想

首先，必须对与模拟功能复用的引脚，通过端口模式控制寄存器(PMCxx)设置是用作模拟功能还是用作数字输入/输出。

用作数字输入/输出时输出电路的基本结构如图2-11所示。与端口的输出锁存器输出复用的SCI功能的输出被输入到AND门，AND门的输出被输入到OR门，OR门的其他输入连接复用的非SCI功能(定时器、RTC、时钟/蜂鸣器的输出、IICA等)的输出。当将这样的引脚用作端口功能或者复用功能时，不使用的复用功能不能影响要使用的功能的输出。此时的设置基本思想如表2-4所示。

图2-11：端口的输出基本结构



注1：当没有POM寄存器时，此信号为Low电平(0)。

注2：当没有复用功能时，此信号为High电平(1)。

注3：当没有复用功能时，此信号为Low电平(0)。

表2-4：设置的基本思想

使用的引脚输出功能	不使用的复用功能的输出设置		
	端口功能	SCI的输出功能	SCI以外的输出功能
端口输出功能	—	High电平输出(1)	Low电平输出(0)
SCI的输出功能	High(1)	—	Low电平输出(0)
SCI以外的输出功能	Low(0)	High电平输出(1)	Low电平输出(0) ^注

注：因为1个引脚有可能复用多个SCI以外的输出功能，所以需要将不使用的复用功能的输出置为Low电平(0)。有关具体的设置方法，请参照“2.5.2 使用的端口功能和复用功能的寄存器设置例子”。

2.5.2 使用的端口功能和复用功能的寄存器设置例子

使用的端口功能和复用功能的寄存器设置例子(64管脚制品)如表2-5~表2-10所示。

表2-5: 使用P00~P06引脚功能时的寄存器设置例子

管脚名称	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI/CAN的输出功能	SCI/CAN以外
P00	P00	输入	—	×	0	1	×	×	—
		输出	×	0	0	0	0/1	—	(TAO)=0
		N沟道漏极开路输出		1	0	0	0/1		
	TI00	输入	—	—	0	1	×	—	—
	TBCLK0	输入	—	×	0	1	×	×	—
	(TAO)	输出	PIOR13 PIOR12=10B	0	0	0	0	—	—
	(INTP8)	输入	PIOR07=1	—	0	1	×	—	—
P01	P01	输入	—	—	0	1	×	—	—
		输出	×	—	0	0	0/1	—	TAIO=0 TO00=0
	TO00	输出	—	—	0	0	0	—	TAIO=0
	TBCLK1	输入	—	—	0	1	×	—	—
	TAIO	输入	PIOR13 PIOR12=00B	—	0	1	×	—	—
		输出	—	—	0	0	0	—	TO00=0
(INTP10)	输入	PIOR07=1	—	0	1	×	—	—	
P02	P02	输入	—	×	0	1	×	×	—
		输出	×	0	0	0	0/1	TxD1/SDO10=1 CTxD0=1	—
		N沟道漏极开路输出		1	0	0	0/1		
	ANI11	模拟输入	—	×	1	1	×	×	—
	TxD1	输出	—	0/1	0	0	1	CTxD0=1	—
	SDO10	输出	—	0/1	0	0	1	CTxD0=1	—
	VCIN10	模拟输入	—	×	1	1	×	×	—
CTxD0	输出	—	0/1	0	0	1	TxD1/SDO10=1	—	
P03	P03	输入	—	×	0	1	×	×	—
		输出	×	0	0	0	0/1	SDA10=1	—
		N沟道漏极开路输出		1	0	0	0/1		
	ANI10	模拟输入	—	×	1	1	×	×	—
	SDI10	输入	—	×	0	1	×	×	—
	RxD1	输入	—	×	0	1	×	×	—
	SDA10	输入/输出	—	1	0	0	1	—	—
	VCIN11	模拟输入	—	×	1	1	×	×	—
CRxD0	输入	—	×	0	1	×	×	—	
P04	P04	输入	—	×	0	1	×	—	—
		输出	×	0	0	0	0/1	SCLK10/SCL10=1	—
		N沟道漏极开路输出		1	0	0	0/1		

	ANI13	模拟输入	—	×	1	1	×	×	—
	SCLK10	输入	—	×	0	1	×	—	—
		输出		0/1	0	0	1	—	—
	SCL10	输出	—	0/1	0	0	1	—	—
P05	P05	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
	(INTP10)	输入	PIOR01=1	—	—	1	×	—	—
P06	P06	输入	—	—	—	1	×	—	—
		输出	×	—	—	0	0/1	—	(TAIO)=0
	(TAIO)	输入	PIOR13 PIOR12=11B	—	—	1	×	—	—
		输出		—	—	0	0	—	—
	(INTP11)	输入	PIOR01=1	—	—	1	×	—	—

表2-6: 使用P10~P17引脚功能时的寄存器设置例子(1/3)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P10	P10	输入	—	×	0	1	×	—	
		输出	—	0	0	0	0/1	SCLK11/SCLK11=1	TMIOB1=0
		N沟道漏极开路输出		1	0	0	0/1		
	ANI9	模拟输入	—	×	1	1	×	×	—
	SCLK11	输入	—	×	0	1	×	—	—
		输出	—	0/1	0	0	1	—	TMIOB1=0
	SCL11	输出	—	0/1	0	0	1	—	TMIOB1=0
	TMIOB1	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0	SCLK11/SCLK11=1	—
TMIOD1	输入	PIOR37=1 PIOR36=0	×	0	1	×	—	—	
	输出		0	0	0	0	SCLK11/SCLK11=1	—	
P11	P11	输入	—	×	0	1	×	—	
		输出	—	0	0	0	0/1	SDA11=1	TMIOD1=0
		N沟道漏极开路输出		1	0	0	0/1		
	ANI8	模拟输入	—	×	1	1	×	×	—
	SI11	输入	—	×	0	1	×	—	—
	SDA11	输入/输出	—	1	0	0	1	—	TMIOD1=0
	RxD0	输入	PIOR35=1	×	0	1	1	—	TMIOD1=0
	TMIOD1	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0	SDA11=1	—
	TMIOA1	输入	PIOR37=0 PIOR36=1	×	0	1	×	—	—
输出		0		0	0	0	SDA11=1	—	
TMIOC1	输入	PIOR37=1 PIOR36=0	×	0	1	×	—	—	
	输出		0	0	0	0	SDA11=1	—	
P12	P12	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	SDO11=0	TMIOA1=0
	SDO11	输出	—	—	—	0	1	—	TMIOA1=0
	TxD0	输出	PIOR35=1	0/1	—	0	1	—	TMIOA1=0
	TMIOA1	输入	—	×	—	1	×	—	—
		输出		0	—	0	0	SDO11=0	—
	TMIOB0	输入	PIOR37=0 PIOR36=1	—	—	1	×	—	—
		输出		—	—	0	0	SDO11=0	—
	TMIOB1	输入	PIOR37=1 PIOR36=0	—	—	1	×	—	—
输出		—		—	0	0	SDO11=0	—	
(INTP5)	输入	PIOR04=1	—	—	1	×	—	—	

表2-6: 使用P10~P17引脚功能时的寄存器设置例子(2/3)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P13	P13	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	TxD2/SDO20=1	TMIOC1=0
		N沟道漏极开路输出		1	—	0	0/1		
	TxD2	输出	PIOR01=0	0/1	—	0	1	—	TMIOC1=0
	SDO20	输出	PIOR01=0	0/1	—	0	1	—	TMIOC1=0
	TMIOC1	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0	TxD2/SDO20=1	—
	TMIOD1	输入	PIOR37=0 PIOR36=1	×	—	1	×	—	—
		输出		0	—	0	0	TxD2/SDO20=1	—
	TMIOA1	输入	PIOR37=1 PIOR36=0	×	—	1	×	—	—
输出		0		—	0	0	TxD2/SDO20=1	—	
P14	P14	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	SDA20=1	TMIOB0=0、 (SCLA0)=0
		N沟道漏极开路输出		1	—	0	0/1		
	RxD2	输入	PIOR01=0	×	—	1	×	—	—
	SDI20	输入	PIOR01=0	×	—	1	×	—	—
	SDA20	输入/输出	PIOR01=0	1	—	0	1	—	TMIOB0=0、 (SCLA0)=0
	TMIOB0	输入	—	×	—	1	×	—	—
		输出	—	1	—	0	0	SDA20=1	(SCLA0)=0
	TMIOC1	输入	PIOR37=0 PIOR36=1	×	—	1	×	—	—
		输出		1	—	0	0	SDA20=1	(SCLA0)=0
TMIOB0	输入	PIOR37=1 PIOR36=0	×	—	1	×	—	—	
	输出		1	—	0	0	SDA20=1	(SCLA0)=0	
(SCLA0)	输入/输出	PIOR02=1	1	—	0	0	SDA20=1	TMIOB0=0	

表2-6: 使用P10~P17引脚功能时的寄存器设置例子(3/3)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P15	P15	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	SCLK20/SCL20 =1	TMIOD0=0 (SDAA0)=0 CLKBUZ1=0
		N沟道漏极开路输出		1	—	0	0/1		
	SCLK20	输入	PIOR01=0	×	—	1	×		
		输出		0/1	—	0	1	—	TMIOD0=0 (SDAA0)=0 CLKBUZ1=0
	SCL20	输出	PIOR01=0	0/1	—	0	1	—	—
	TMIOD0	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0	SCLK20/SCL20 =1	(SDAA0)=0 CLKBUZ1=0
TMIOB0	输入	PIOR37=1 PIOR36=0	×	—	1	×	—	—	
	输出		0	—	0	0	SCLK20/SCL20 =1	(SDAA0)=0 CLKBUZ1=0	
(SDAA0)	输入/输出	PIOR02=1	1	—	0	0	SCLK20/SCL20 =1	TMIOD0=0 CLKBUZ1=0	
P16	P16	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	TO01=0 TMIOC0=0 (TMIOA1)=0
	TI01	输入	—	—	—	1	×	—	—
	TO01	输出	—	—	—	0	0	—	TMIOC0=0、 (TMIOA1)=0
	INTP5	输入	PIOR04=0	—	—	1	×	—	—
	TMIOC0	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0	—	TO01=0 (TMIOA1)=0
	(SDI00)	输入	PIOR01=1	—	—	1	×	—	—
(RxD0)	输入	PIOR01=1 PIOR06=0	—	—	1	×	—	—	
(TMIOA1)	输入	PIOR22=1	×	—	1	×	—	—	
	输出		—	—	0	0	—	TO01=0 TMIOC0=0	
P17	P17	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	(TxD0)/(SDO00) =1	TO02=0 TMIOA0=0 (TMIOD0)=0
		N沟道漏极开路输出		1	—	0	0/1		
	TI02	输入	—	×	—	1	×		
	TO02	输出	×	0	—	0	0	(TxD0)/(SDO00) =1	TMIOA0=0 (TMIOD0)=0
	TMIOA0	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0	(TxD0)/(SDO00) =1	TO02=0 (TMIOD0)=0
	TMCLK	输入	—	×	—	1	×	—	—
(SDO00)	输出	PIOR01=1	0/1	—	0	1	—	TO02=0、 TMIOA0=0、 (TMIOD0)=0	
(TxD0)	输出	PIOR01=1 PIOR06=0	0/1	—	0	1	—	TO02=0 TMIOA0=0 (TMIOD0)=0	
(TMIOD0)	输入	PIOR26=1	×	—	1	×	—	—	
	输出		0	—	0	0	(TxD0)/(SDO00) =1	TO02=0 TMIOA0=0	

表2-7：使用P20~P27引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	PMCxx	PMxx	Pxx
	功能名称	输入/输出				
P20	P20	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI0	模拟输入	—	1	1	×
	AVREFP	基准电压输入	—	1	1	×
	VCIN12	模拟输入	—	1	1	×
	(INTP11)	输入	PIOR07=1	0	1	1
P21	P21	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI1	模拟输入	—	1	1	×
	AVREFM	基准电压输入	—	1	1	×
	VCIN13	模拟输入	—	1	1	×
P22	P22	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI2	模拟输入	—	1	1	×
	ANO0	模拟输出	—	1	1	×
	VCIN0	模拟输入	—	1	1	×
	PGA0IN	模拟输入	—	1	1	×
P23	P23	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI3	模拟输入	—	1	1	×
	ANO1	模拟输出	—	1	1	×
	PGA0GND	模拟输入	—	1	1	×
P24	P24	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI4	模拟输入	—	1	1	×
	PGA1IN	模拟输入	—	1	1	×
P25	P25	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI5	模拟输入	—	1	1	×
	PGA1GND	模拟输入	—	1	1	×
P26	P26	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI6	模拟输入	—	1	1	×
P27	P27	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI7	模拟输入	—	1	1	×

表2-8: 使用P30~P43引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P30	P30	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	SCLK00/SCL00=1	RTC1HZ=0 TAO=0 (TMIOB1)=0
		N沟道漏极开路输出		1	—	0	0/1		
	INTP3	输入	PIOR00=0 PIOR05=0	×	—	1	×	—	—
	RTC1HZ	输出	×	0	—	0	0	SCLK00/SCL00=1	TAO=0 (TMIOB1)=0
	SCLK00	输入	PIOR01=0	×	—	1	×	—	—
		输出		0/1	—	0	1	—	RTC1HZ=0 TAO=0 (TMIOB1)=0
	SCL00	输出	PIOR01=0	0/1	—	0	0	—	RTC1HZ=0 TAO=0 (TMIOB1)=0
	TAO	输出	PIOR13 PIOR12=00B	0	—	0	0	SCLK00/SCL00=1	RTC1HZ=0 (TMIOB1)=0
	(TMIOB1)	输入	PIOR23=1	×	—	1	×	—	—
输出		0		—	0	0	SCLK00/SCL00=1	RTC1HZ=0 TAO=0	
P31	P31	输入	—	—	—	1	×	—	—
		输出	×	—	—	0	0/1	—	TO03=0 VCOUT1=0 (CLKBUZ0)=0 (TAIO)=0
	TI03	输入	—	—	—	1	×	—	—
	TO03	输出	×	—	—	0	0	—	VCOUT1=0 (CLKBUZ0)=0 (TAIO)=0
	INTP4	输入	PIOR00=0 ^{注13} PIOR05=0 ^{注14}	—	—	1	×	—	—
	(TAIO)	输入	PIOR11 PIOR10=01B	—	—	1	×	—	—
		输出		—	—	0	0	—	TO03=0 VCOUT1=0 (CLKBUZ0)=0
	(CLKBUZ0)	输出	PIOR03=1	—	—	0	0	—	TO03=0 (TAIO)=0 VCOUT1=0
VCOUT1	输出	PIOR21=0	—	—	0	0	—	TO03=0 (TAIO)=0 (CLKBUZ0)=0	
P40	P40	输入	—	×	—	1	×	—	—
		输出	×	—	—	0	0/1	—	—
	SWDIO	输入	—	×	—	1	×	—	—
		输出	×	—	—	0	×	—	—
P41	P41	输入	—	—	—	1	×	—	—
		输出	×	—	—	0	0/1	—	(TAIO)=0
	(TAIO)	输入	PIOR11 PIOR10=10B	—	—	1	×	—	—
		输出		—	—	0	0	—	—
P42	P42	输入	—	—	—	1	×	—	—
		输出	×	—	—	0	0/1	—	—
	(INTP8)	输入	PIOR00=1 PIOR07=0	—	—	1	×	—	—
	P43	P43	输入	—	—	—	1	×	—
输出			×	—	—	0	0/1	—	—
(INTP9)		输入	PIOR00=1	—	—	1	×	—	—

表2-9: 使用P50~P55引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI/CAN的 输出功能	SCI/CAN以外
P50	P50	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	SDA00=1	TBIO0=0 (TAO)=0 (TMIOC1)=0
		N沟道漏极 开路输出		1	—	0	0/1		
	INTP1	输入	PIOR00=0	×	—	1	×	—	—
	SDI00	输入	PIOR01=0	×	—	1	×	—	—
	RxD0	输入	PIOR01=0	×	—	1	×	—	—
	SDA00	输入/输出	PIOR01=0	1	—	0	1	—	TBIO0=0 (TAO)=0 (TMIOC1)=0
	(CRxD0)	输入	PIOR33=1	×	—	1	×	—	—
	TBIO0	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0	SDA00=1	(TAO)=0 (TMIOC1)=0
	(TAO)	输出	PIOR13 PIOR12=01B	0	—	0	0	SDA00=1	TBIO0=0 (TMIOC1)=0
(TMIOC1)	输入	PIOR24=1	×	—	1	×	—	—	
	输出		0	—	0	0	SDA00=1	TBIO0=0 (TAO)=0	
P51	P51	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	TxD0/SDO00=1	TBIO1=0 (TMIOD1)=0
		N沟道漏极 开路输出		1	—	0	0/1		
	INTP2	输入	PIOR01=0	×	—	1	×	—	—
	SDO00	输出	PIOR01=0	0/1	—	0	1	—	TBIO1=0 (TMIOD1)=0
	TxD0	输出	PIOR01=0	0/1	—	0	1	—	TBIO1=0 (TMIOD1)=0
	(CTxD0)	输出	PIOR33=1	0/1	—	0	1	TxD0/SDO00=1	TBIO1=0 (TMIOD1)=0
	TBIO1	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0	TxD0/SDO00=1	(TMIOD1)=0
(TMIOD1)	输入	PIOR25=1	×	—	1	×	—	—	
	输出		0	—	0	0	TxD0/SDO00=1	TBIO1=0	
P52	P52	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	—	—
		N沟道漏极 开路输出		1	—	0	0/1		
(INTP1)	输入	PIOR00=1	×	—	1	×	—	—	
P53	P53	输入	—	×	—	1	×	—	—
		输出	×	0	—	0	0/1	—	—
		N沟道漏极 开路输出		1	—	0	0/1		
(INTP2)	输入	PIOR00=1	×	—	1	×	—	—	
P54	P54	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	—	—
		N沟道漏极 开路输出		1	—	0	0/1		
(INTP3)	输入	PIOR00=1	×	—	1	×	—	—	
P55	P55	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	(SCLK00)=1	(CLKBUZ1)=0

		N沟道漏极 开路输出		1	—	0	0/1		
	(INTP4)	输入	PIOR00=1	×	—	1	×	—	—
	(CLKBUZ1)	输出	PIOR04=1	0	—	0	0	(SCLK00)=1	—
	(SCLK00)	输入	PIOR01=1	×	—	1	×	—	—
		输出		0/1	—	0	1	—	(CLKBUZ1)=0

表2-10: 使用P60~P63引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的 输出功能	SCI以外
P60	P60	输入	—	—	—	1	×	—	—
		N沟道漏极 开路输出 (6V耐压)	×	—	—	0	0/1	—	SCLA0=0
	SCLA0	输入/输出	PIOR02=0	—	—	0	0	—	—
P61	P61	输入	—	—	—	1	×	—	—
		N沟道漏极 开路输出 (6V耐压)	×	—	—	0	0/1	—	SDAA0=0
	SDAA0	输入/输出	PIOR02=0	—	—	0	0	—	—
P62	P62	输入	—	—	—	1	×	—	—
		N沟道漏极 开路输出 (6V耐压)	×	—	—	0	0/1	—	—
	SS00	输入	—	—	—	1	×	—	—
P63	P63	输入	—	—	—	1	×	—	—
		N沟道漏极 开路输出 (6V耐压)	×	—	—	0	0/1	—	—

表2-11: 使用P70~P77引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P70	P70	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	SCLK21/SCL21=1	(VCOUT1)=0
	KR0	输入	—	—	—	1	×	—	—
	SCLK21	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	1	—	(VCOUT1)=0
	SCL21 (VCOUT1)	输出	—	—	—	0	1	—	(VCOUT1)=0
P71	P71	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	SDA21=1	(VCOUT0)=0
		N沟道漏极 开路输出	—	1	—	0	0/1		
	KR1	输入	—	×	—	1	×	—	—
	SDI21	输入	—	×	—	1	×	—	—
	SDA21 (VCOUT0)	输入/输出	—	0/1	—	0	1	—	(VCOUT0)=0
	输出	PIOR20=1	—	—	0	0	SDA21=1	—	
P72	P72	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	SDO21=1	—
	KR2	输入	—	—	—	1	×	—	—
	SDO21	输出	—	—	—	0	1	—	—
P73	P73	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	SDO01=1	—
	KR3	输入	—	—	—	1	×	—	—
	SDO01	输出	—	—	—	0	1	—	—
P74	P74	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	SDA01=1	—
		N沟道漏极 开路输出	—	1	—	0	0/1		
	KR4	输入	—	—	—	1	×	—	—
	INTP8	输入	PIOR00=0 PIOR07=0	—	—	1	×	—	—
	SDI01	输入	—	—	—	1	×	—	—
SDA01	输入/输出	—	0/1	—	0	1	—	—	
P75	P75	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	SCLK01/SCL01=1	—
	KR5	输入	—	—	—	1	×	—	—
	INTP9	输入	PIOR00=0	—	—	1	×	—	—
	SCLK01	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	1	—	—
SCL01	输出	—	—	—	0	1	—	—	
P76	P76	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
	KR6	输入	—	—	—	1	×	—	—
	INTP10	输入	PIOR01=0 PIOR07=0	—	—	1	×	—	—
	(RxD2)	输入	PIOR01=1	—	—	1	×	—	—
P77	P77	输入	—	—	—	1	×	—	—
		输出	×	—	—	0	0/1	(TxD2)=1	—

	KR7	输入	—	—	—	1	×	—	—
	INTP11	输入	PIOR01=0、 PIOR07=0	—	—	1	×	—	—
	(TxD2)	输出	PIOR01=1	—	—	0	1	—	—

图2-12：使用P120引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P120	P120	输入	—	—	0	1	×	—	—
		输出	—	—	0	0	0/1	—	—
	ANI14	模拟输入	—	—	1	1	×	—	—
	VCOOUT0	输出	PIOR20=0	—	0	0	0	—	—

图2-13：使用P121~P124引脚功能时的寄存器设置例子

引脚名	使用的功能		CMC (EXCLK、OSCSEL、EXCLKS、 OSCSELS)		Pxx
	功能名称	输入/输出			
P121	P121	输入	00xx/10xx/11xx	×	
	X1	—	01xx	—	
P122	P122	输入	00xx/10xx/11xx	×	
	X2	—	01xx	—	
	EXCLK	输入	11xx	—	
P123	P123	输入	xx00/xx10/xx11	×	
	XT1	—	xx01	—	
P124	P124	输入	xx00/xx10/xx11	×	
	XT2	—	xx01	—	
	EXCLKS	输入	xx11	—	

图2-14：使用P130~P147引脚功能时的寄存器设置例子

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出	
	功能名称	输入/输出						SCI的输出功能	SCI以外
P130	P130	输出	—	—	—	—	0/1	—	—
P136	P136	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
P137	P137	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
	SWCLK	输入	—	—	—	1	×	—	—
P140	P140	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	CLKBUZ0=0
	CLKBUZ0	输出	PIOR03=0	—	—	0	0	—	—
P141	P141	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	CLKBUZ1=0
	CLKBUZ1	输出	PIOR04=0	—	—	0	0	—	—
	INTP7	输入	—	—	—	1	×	—	—
P146	P146	输入	—	—	0	1	×	—	—
		输出	—	—	0	0	0/1	—	—
	ANI15	模拟输入	—	—	1	1	×	—	—
P147	P147	输入	—	—	0	1	×	—	—
		输出	—	—	0	0	0/1	—	—
	ANI12	模拟输入	—	—	1	1	×	—	—
	VREF0	模拟输入	—	—	1	1	×	—	—

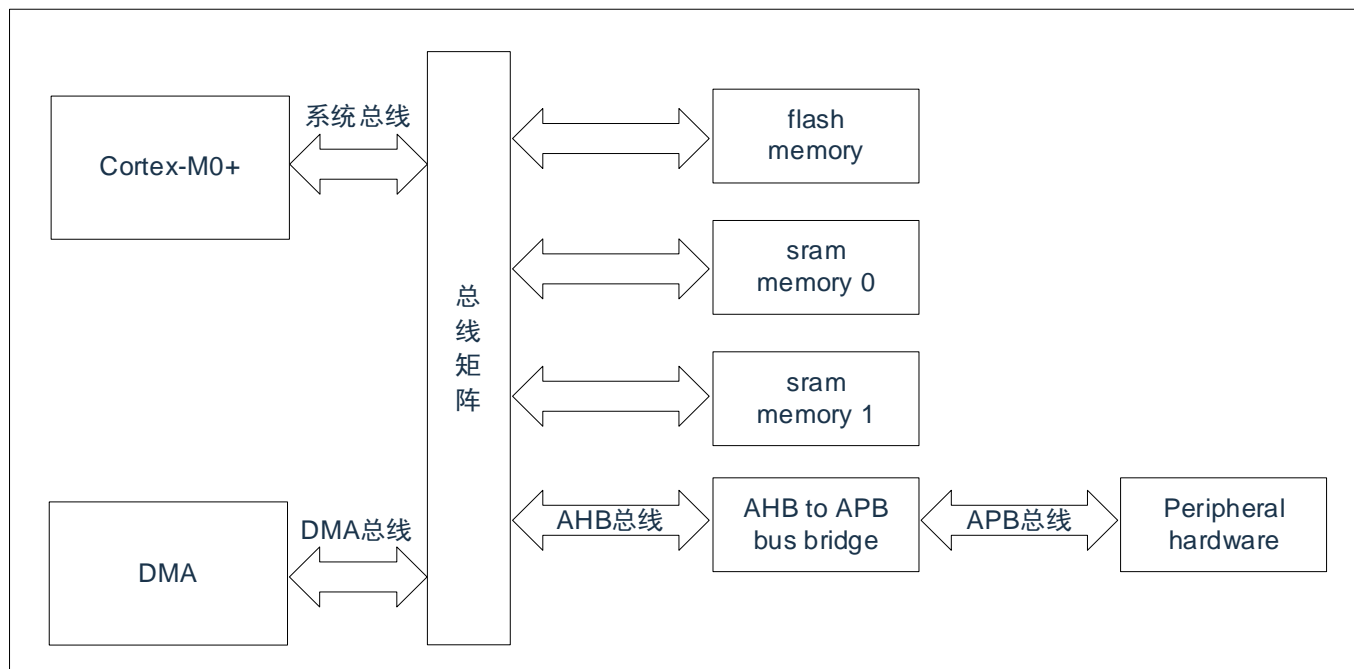
第3章 系统结构

3.1 概述

本产品系统由以下部分组成：

- 2个AHB总线Master：
 - Cortex-M0+
 - 增强型DMA
- 4个AHB总线Slaves：
 - FLASH存储器
 - SRAM存储器0
 - SRAM存储器1
 - AHB to APB Bridge，包含所有APB接口外设

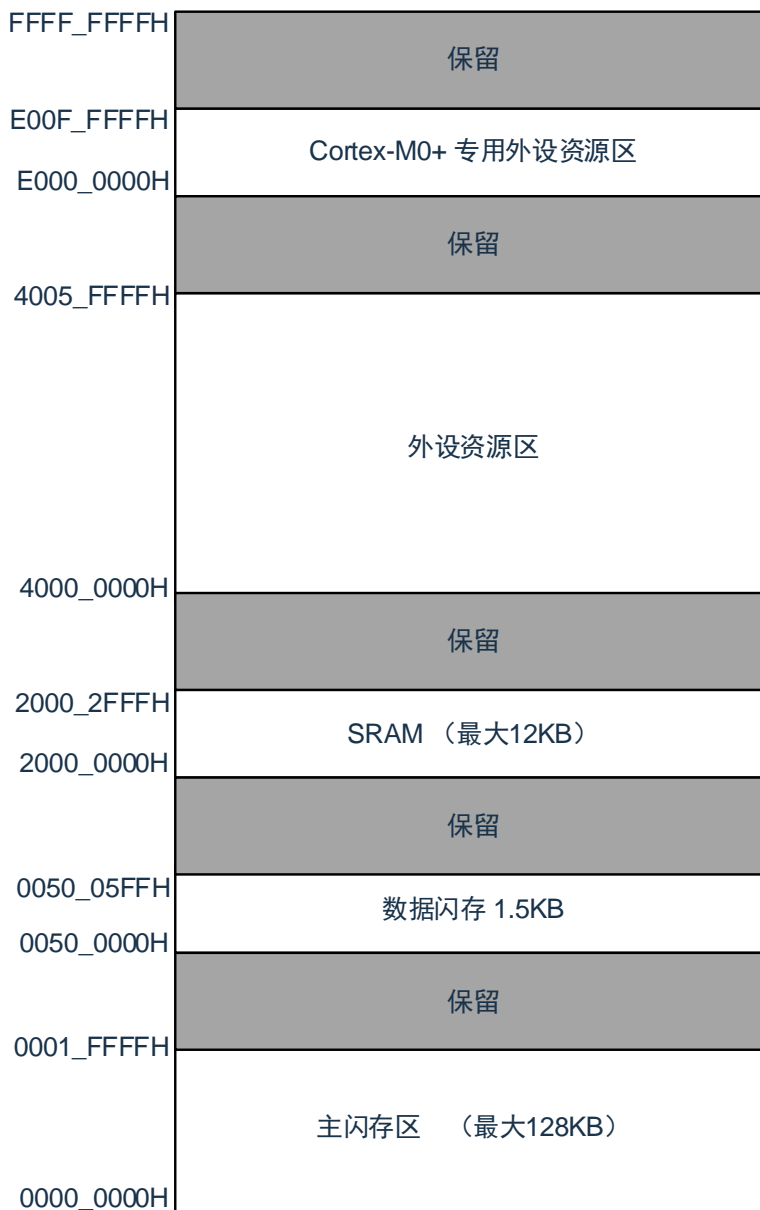
图3-1：系统结构示意图



- 系统总线：此总线连接Cortex-M0+内核的系统总线(外设总线)到总线矩阵，总线矩阵协调着内核和DMA间的访问。
- DMA总线：此总线将DMA的AHB主控接口与总线矩阵相联，总线矩阵协调着CPU和DMA到SRAM、闪存和外设的访问。
- 总线矩阵：总线矩阵协调内核系统总线和DMA主控总线之间的访问仲裁，仲裁采用固定优先级，DMA优先级高。
- AHB to APB Bridge：AHB to APB Bridge 在AHB和APB总线间提供同步连接。有关连接到每个桥的不同外设的地址映射请参考表3-1。

3.2 系统地址划分

图3-2: 地址区域划分示意图



外设地址分配

表3-1：外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_4FFF	保留	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	中断控制	
0x4000_7000 - 0x4001_8FFF	保留	
0x4001_9000 - 0x4001_9FFF	MTB 控制	详见第 1 章 调试支持
0x4001_A000 - 0x4001_FFFF	保留	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1000 - 0x4002_1001	看门狗定时器	
0x4002_1002 - 0x4002_1800	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见第 31 章 安全功能
0x4002_1C00 - 0x4002_1FFF	时钟控制	
0x4002_2000 - 0x4003_FFFF	保留	
0x4004_0000 - 0x4004_0FFF	GPIO	
0x4004_1100 - 0x4004_19FF	串行通信单元	
0x4004_1A00 - 0x4004_1CFF	串行接口 IICA	
0x4004_1D00 - 0x4004_1FFF	定时器阵列	
0x4004_2000 - 0x4004_21FF	保留	
0x4004_2200 - 0x4004_25FF	定时器 A	
0x4004_2600 - 0x4004_29FF	定时器 B	
0x4004_2A00 - 0x4004_2BFF	定时器 M	
0x4004_2C00 - 0x4004_2FFF	定时器 C	
0x4004_3000 - 0x4004_31FF	保留	
0x4004_3200 - 0x4004_32FF	通用 CRC	详见第 31 章 安全功能
0x4004_3300 - 0x4004_33FF	保留	
0x4004_3400 - 0x4004_37FF	联动控制器	
0x4004_3800 - 0x4004_3BFF	比较器	
0x4004_3C00 - 0x4004_3FFF	定时器 M	
0x4004_4000 - 0x4004_46FF	IrDA	
0x4004_4700 - 0x4004_4AFF	DA 转换器	
0x4004_4B00 - 0x4004_4EFF	键中断	
0x4004_4F00 - 0x4004_4FFF	实时时钟	
0x4004_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_5AFF	CAN 控制器	
0x4004_5B00 - 0x4004_5BFF	中断控制	
0x4008_0000 - 0x4008_01FF	硬件除法器	
0x4008_0200 - 0xDFFF_FFFF	保留	

第4章 时钟发生电路

用于主系统时钟的谐振器连接引脚/外部时钟输入引脚、用于副系统时钟的谐振器连接引脚/外部时钟输入引脚的有无因产品而不同。

	24、32引脚的产品	36、40、48、52、64引脚的产品
X1、X2引脚	○	○
EXCLK引脚	○	○
XT1、XT2引脚	—	○
EXCLKS引脚	—	○

4.1 时钟发生电路的功能

时钟发生电路是产生给CPU和外围硬件提供时钟的电路。有以下3种系统时钟和时钟振荡电路。

(1) 主系统时钟

a) X1振荡电路

能通过给X1引脚和X2引脚连接谐振器使 $F_X=1\sim 20\text{MHz}$ 的时钟振荡，并且能通过进入深度睡眠模式或者设置MSTOP位(时钟运行状态控制寄存器CSC的bit7)使振荡停止。

b) 高速内部振荡器(高速OCO)

能通过选项字节(000C2H)从 $F_{\text{HOCO}}=64\text{MHz}$ 、48MHz、32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz和1MHz(TYP.)中选择频率进行振荡。当选择64MHz作为 F_{HOCO} 时， F_{IH} 分频为32MHz；当选择48MHz或者更低的频率作为 F_{HOCO} 时， F_{IH} 不被分频而和 F_{HOCO} 的频率相同。在解除复位后，CPU一定以此高速内部振荡器时钟开始运行。能通过进入深度睡眠模式或者设置HIOSTOP位(CSC寄存器的bit0)使振荡停止。能通过高速内部振荡器的频率选择寄存器(HOCODIV)更改选项字节设置的频率。有关频率设置，请参照“图4-10：高速内部振荡器的频率选择寄存器(HOCODIV)的格式”。

另外，能由EXCLK/X2/P122引脚提供外部主系统时钟($F_{\text{EX}}=1\sim 20\text{MHz}$)，并且能通过进入深度睡眠模式或者设置MSTOP位将外部主系统时钟的输入置为无效。

能通过设置MCM0位(系统时钟控制寄存器(CKC)的bit4)进行高速系统时钟(X1时钟或者外部主系统时钟)和高速内部振荡器时钟的切换。

(2) 副系统时钟

- XT1振荡电路

能通过给XT1引脚和XT2引脚连接32.768KHz的谐振器使 $F_{XT}=32.768\text{KHz}$ 的时钟振荡，并且能通过设置XTSTOP位(时钟运行状态控制寄存器(CSC)的bit6)使振荡停止。

另外，能由EXCLKS/XT2/P124引脚提供外部副系统时钟($F_{EXS}=32.768\text{KHz}$)，并且能通过设置XTSTOP位将外部副系统时钟的输入置为无效。

(3) 低速内部振荡器时钟(低速OCO)

能使 $F_{IL}=15\text{KHz}$ (TYP.)的时钟振荡。

不能将低速内部振荡器时钟用作CPU时钟。

SysTick定时器使用低速内部振荡器时钟作为外部参考时钟。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 看门狗定时器
- 实时时钟
- 15位间隔定时器
- 定时器A

当选项字节(000C0H)的bit4(WDTON)或者副系统时钟提供模式控制寄存器(OSMC)的bit4(WUTMMCK0)为“1”时，低速内部振荡器振荡。

但是，在WDTON位为“1”并且WUTMMCK0位为“0”而且选项字节(000C0H)的bit0(WDSTBYON)为“0”时，如果进入深度睡眠模式或者睡眠模式，低速内部振荡器就停止振荡。

注意：只有在使用固定周期中断功能时，才能选择低速内部振荡器时钟(F_{IL})作为实时时钟的计数时钟。

备注： F_X ：X1时钟振荡频率

F_{HOCO} ：高速内部振荡器的时钟频率(最大64MHz)

F_{IH} ：高速内部振荡器的时钟频率(最大48MHz)注

F_{EX} ：外部主系统时钟频率

F_{XT} ：XT1时钟振荡频率

F_{EXS} ：外部副系统时钟频率

F_{IL} ：低速内部振荡器的时钟频率

注：由硬件进行控制，当将 F_{HOCO} 设置为64MHz时，使时钟频率和 F_{HOCO} 的2分频相同；当设置为48MHz或者更低的频率时，使时钟频率和 F_{HOCO} 相同。要给定定时器M提供64MHz时，必须将 F_{CLK} 设置为 F_{IH} 。

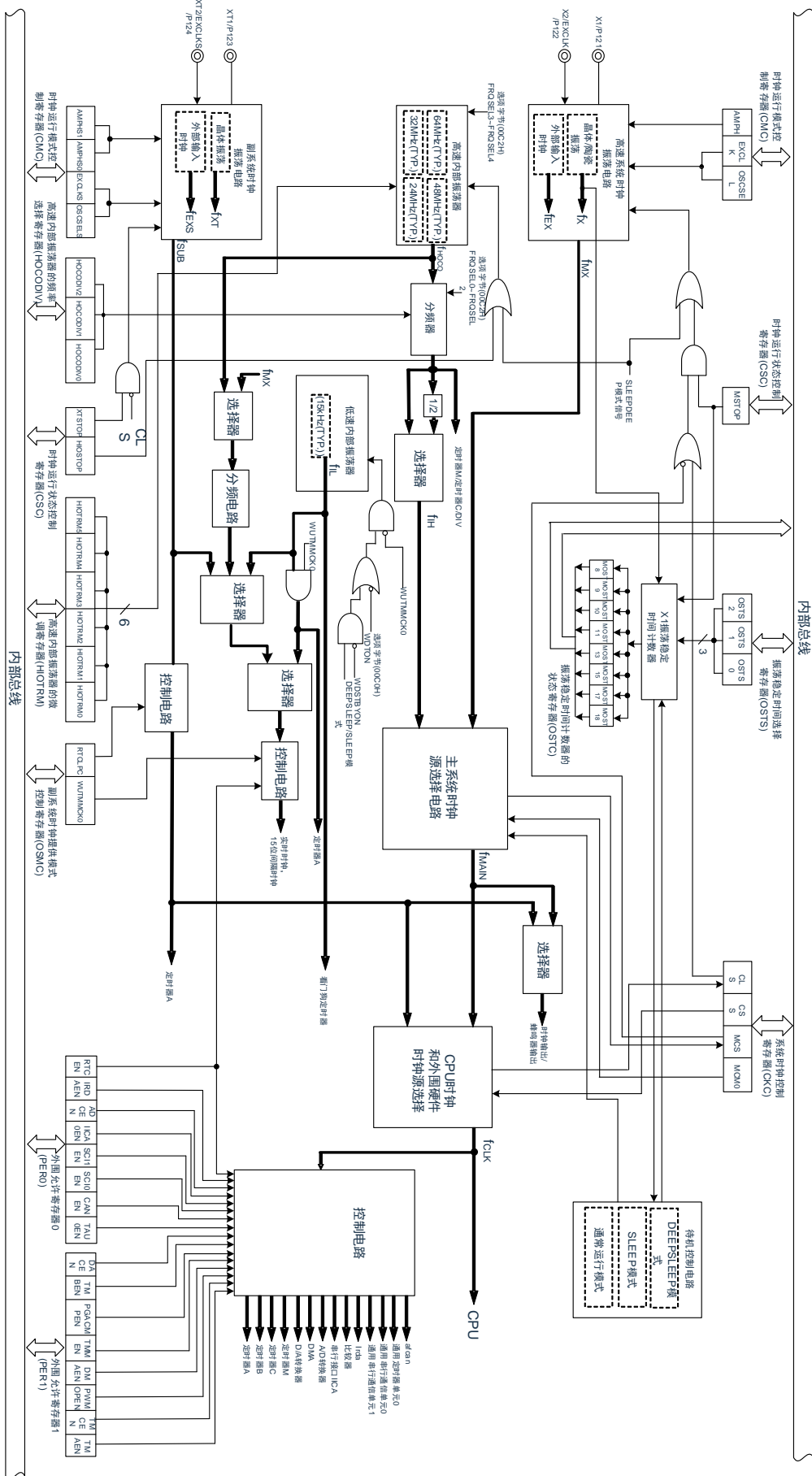
4.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表4-1：时钟发生电路的结构

项目	结构
控制寄存器	时钟运行模式控制寄存器(CMC) 系统时钟控制寄存器(CKC) 时钟运行状态控制寄存器(CSC) 振荡稳定时间计数器的状态寄存器(OSTC) 振荡稳定时间选择寄存器(OSTS) 外围允许寄存器0、1(PER0、PER1) 副系统时钟提供模式控制寄存器(OSMC) 高速内部振荡器的频率选择寄存器(HOCODIV) 高速内部振荡器的微调寄存器(HIOTRM)
振荡电路	X1振荡电路 XT1振荡电路 高速内部振荡器 低速内部振荡器

图4-1: 时钟发生电路的框图



备注: F_X : X1时钟振荡频率

F_{HOCO} : 高速内部振荡器的时钟频率(最大64MHz)

F_{IH} : 高速内部振荡器的时钟频率(最大48MHz)^注

F_{EX} : 外部主系统时钟频率

F_{MX} : 高速系统时钟频率

F_{MAIN} : 主系统时钟频率

F_{XT} : XT1时钟振荡频率

F_{EXS} : 外部副系统时钟频率

F_{SUB} : 副系统时钟频率

F_{CLK} : CPU/外围硬件的时钟频率

F_{IL} : 低速内部振荡器的时钟频率

注: 由硬件进行控制, 当将 F_{HOCO} 设置为64MHz时, 使时钟频率和 F_{HOCO} 的2分频相同; 当设置为48MHz或者更低的频率时, 使时钟频率和 F_{HOCO} 相同。要给定时器M提供64MHz时, 必须将 F_{CLK} 设置为 F_{IH} 。

4.3 控制时钟发生电路的寄存器

通过以下寄存器控制时钟发生电路。

- 时钟运行模式控制寄存器(CMC)
- 系统时钟控制寄存器(CKC)
- 时钟运行状态控制寄存器(CSC)
- 振荡稳定时间计数器的状态寄存器(OSTC)
- 振荡稳定时间选择寄存器(OSTS)
- 外围允许寄存器0、1(PER0、PER1)
- 副系统时钟提供模式控制寄存器(OSMC)
- 高速内部振荡器的频率选择寄存器(HOCODIV)
- 高速内部振荡器的微调寄存器(HIOTRM)

注意：分配的寄存器和位因产品而不同。必须给未分配的位设置初始值。

4.3.1 时钟运行模式控制寄存器(CMC)

这是设置X1/P121、X2/EXCLK/P122、XT1/P123、XT2/EXCLKS/P124引脚的运行模式以及选择振荡电路增益的寄存器。

在解除复位后，只能通过8位存储器操作指令写1次CMC寄存器。能通过8位存储器操作指令读此寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图4-2: 时钟运行模式控制寄存器(CMC)的格式

地址: 40020400H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS ^注	OSCSELS ^注	0	AMPHS1 ^注	AMPHS0 ^注	AMPH

EXCLK	OSCSEL	高速系统时钟引脚的运行模式	X1/P121引脚	X2/EXCLK/P122引脚
0	0	输入端口模式	输入端口	
0	1	X1振荡模式	连接晶体或者陶瓷谐振器。	
1	0	端口模式	输入端口	
1	1	外部时钟输入模式	输入端口	外部时钟输入

EXCLKS	OSCSELS	副系统时钟引脚的运行模式	XT1/P123引脚	XT2/EXCLKS/P124引脚
0	0	输入端口模式	输入端口	
0	1	XT1振荡模式	连接晶体谐振器。	
1	0	输入端口模式	输入端口	
1	1	外部时钟输入模式	输入端口	外部时钟输入

AMPHS1	AMPHS0	XT1振荡电路的振荡模式选择
0	0	低功耗振荡(默认)
0	1	通常的振荡
1	0	超低功耗振荡
1	1	禁止设置。

AMPH	X1时钟振荡频率的控制
0	$1\text{MHz} \leq F_x \leq 10\text{MHz}$
1	$10\text{MHz} < F_x \leq 20\text{MHz}$

注: EXCLKS位, OSCSELS位, AMPHS1位和AMPHS0位只在上电复位时被初始化, 而在其他复位时保持不变。

注意:

1. 在解除复位后，只能通过8位存储器操作指令写1次CMC寄存器。当以初始值(“00H”)使用CMC寄存器时，为了防止程序失控时的误动作(如果误写“00H”)以外的值就不能恢复)，必须在解除复位后将CMC寄存器置“00H”。
2. 在解除复位后并且在通过设置时钟运行状态控制寄存器(CSC)开始X1或者XT1振荡前，必须设置CMC寄存器。
3. 当X1时钟振荡频率超过10MHz时，必须将AMPH位置“1”。
4. 必须在解除复位后并且在选择 F_{IH} 作为 F_{CLK} 的状态(将 F_{CLK} 切换为 F_{MX} 或者 F_{SUB} 前的状态)下设置AMPH位、AMPHS1位和AMPHS0位。
5. 必须通过软件对 F_{XT} 的振荡稳定时间进行计数。
6. 系统时钟的频率上限为48MHz，但是X1振荡电路的频率上限为20MHz。

备注: F_x : X1时钟振荡频率

4.3.2 系统时钟控制寄存器(CKC)

这是选择CPU/外围硬件时钟和主系统时钟的寄存器。通过8位存储器操作指令设置CKC寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图4-3：系统时钟控制寄存器(CKC)的格式

地址：40020404H 复位后：00H R/W^{注1}

符号	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0

CLS	CPU/外围硬件时钟(F _{CLK})的状态
0	主系统时钟(F _{MAIN})
1	副系统时钟(F _{SUB})

CSS ^{注2}	CPU/外围硬件时钟(F _{CLK})的选择
0	主系统时钟(F _{MAIN})
1	副系统时钟(F _{SUB})

MCS	主系统时钟(F _{MAIN})的状态
0	高速内部振荡器时钟(F _{IH})
1	高速系统时钟(F _{MX})

MCM0 ^{注2}	主系统时钟(F _{MAIN})的运行控制
0	选择高速内部振荡器时钟(F _{IH})作为主系统时钟(F _{MAIN})。
1	选择高速系统时钟(F _{MX})作为主系统时钟(F _{MAIN})。

注1：bit7和bit5是只读位。

注2：禁止在将CSS位置“1”的状态下更改MCM0位的值。

备注：F_{HOCO}：高速内部振荡器的时钟频率(最大64MHz)

F_{IH}：高速内部振荡器的时钟频率(最大48MHz)^注

F_{MX}：高速系统时钟频率

F_{MAIN}：主系统时钟频率

F_{SUB}：副系统时钟频率

注：由硬件进行控制，当将F_{HOCO}设置为64MHz时，使时钟频率和F_{HOCO}的2分频相同；当设置为48MHz或者更低的频率时，使时钟频率和F_{HOCO}相同。要给定定时器M提供64MHz时，必须将F_{CLK}设置为F_{IH}。

注意：

1. 必须将bit0~3置“0”。
2. 给CPU和外围硬件提供CSS位设置的时钟。如果更改CPU时钟，就同时更改外围硬件的时钟(实时时钟、15位间隔定时器、时钟输出/蜂鸣器输出和看门狗定时器除外)。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。
3. 如果将副系统时钟用作外围硬件时钟，就无法保证A/D转换器和IICA的运行。有关外围硬件的运行特性，请参照各外围硬件的章节和数据手册的电气特性。
4. 要选择F_{HOCO}作为定时器M的计数源时，必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将F_{CLK}设置为F_{IH}。如果要将F_{CLK}改为F_{IH}以外的时钟，就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

4.3.3 时钟运行状态控制寄存器(CSC)

这是控制高速系统时钟、高速内部振荡器时钟和副系统时钟(低速内部振荡器时钟除外)运行的寄存器。通过8位存储器操作指令设置CSC寄存器。

在产生复位信号后，此寄存器的值变为“C0H”。

图4-4：时钟运行状态控制寄存器(CSC)的格式

地址：40020401H	复位后：C0H	R/W						
符号	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP

MSTOP	高速系统时钟的运行控制		
	X1振荡模式	外部时钟输入模式	输入端口模式
0	X1振荡电路运行	EXCLK引脚的外部时钟有效	输入端口
1	X1振荡电路停止	EXCLK引脚的外部时钟无效	

XTSTOP	副系统时钟的运行控制		
	XT1振荡模式	外部时钟输入模式	输入端口模式
0	XT1振荡电路运行	EXCLKS引脚的外部时钟有效	输入端口
1	XT1振荡电路停止	EXCLKS引脚的外部时钟无效	

HIOSTOP	高速内部振荡器时钟的运行控制
0	高速内部振荡器运行
1	高速内部振荡器停止

注意：

1. 在解除复位后，必须在设置时钟运行模式控制寄存器(CMC)后设置CSC寄存器。
2. 在解除复位后并且在将MSTOP位置“0”前，必须设置振荡稳定时间选择寄存器(OSTS)。但是，当以初始值使用OSTS寄存器时，不需要设置OSTS寄存器。
3. 要通过设置MSTOP位开始X1振荡时，必须通过振荡稳定时间计数器的状态寄存器(OSTC)确认X1时钟的振荡稳定时间。
4. 要通过设置XSTOP位开始XT1振荡时，必须通过软件等待副系统时钟所需的振荡稳定时间。
5. 不能通过CSC寄存器停止被选择为CPU/外围硬件时钟(F_{CLK})的时钟。
6. 有关用于停止时钟振荡(外部时钟输入无效)的寄存器标志设置和停止前的条件，请参照表4-2。

表4-2：时钟停止方法

时钟	时钟停止前的条件(外部时钟输入无效)	设置CSC寄存器的标志
X1时钟	CPU/外围硬件时钟以高速系统时钟以外的时钟运行。 (CLS=0并且MCS=0, 或者CLS=1)	MSTOP=1
外部主系统时钟		
XT1时钟	CPU/外围硬件时钟以副系统时钟以外的时钟运行。 (CLS=0)	XTSTOP=1
外部副系统时钟		
高速内部振荡器时钟	CPU/外围硬件时钟以高速内部振荡器时钟以外的时钟运行。 (CLS=0并且MCS=1, 或者CLS=1)	HIOSTOP=1

4.3.4 振荡稳定时间计数器的状态寄存器(OSTC)

这是表示X1时钟的振荡稳定时间计数器计数状态的寄存器。能在以下情况下确认X1时钟的振荡稳定时间：

- 当CPU时钟为高速内部振荡器时钟或者副系统时钟并且开始X1时钟的振荡时
- 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除睡眠模式时

能通过8位存储器操作指令读OSTC寄存器。

通过复位信号的产生、进入深度睡眠模式或者MSTOP位(时钟运行状态控制寄存器(CSC)的bit7)为“1”，此寄存器的值变为“00H”。

备注：在以下情况下，振荡稳定时间计数器开始计数：

- 当X1时钟开始振荡(EXCLK、OSCSEL=0、1→MSTOP=0)时
- 当解除深度睡眠模式时

图4-5: 振荡稳定时间计数器的状态寄存器(OSTC)的格式

地址: 40020402H 复位后: 00H R

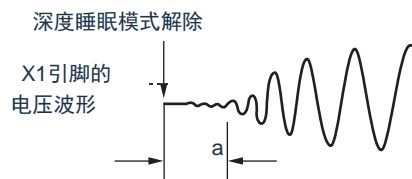
符号 7 6 5 4 3 2 1 0

OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
------	-------	-------	--------	--------	--------	--------	--------	--------

MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	振荡稳定时间状态		
								F _x =10MHz	F _x =20MHz	
0	0	0	0	0	0	0	0	小于2 ⁸ /F _x	小于25.6us	小于12.8us
1	0	0	0	0	0	0	0	至少2 ⁸ /F _x	至少25.6us	至少12.8us
1	1	0	0	0	0	0	0	至少2 ⁹ /F _x	至少51.2us	至少25.6us
1	1	1	0	0	0	0	0	至少2 ¹⁰ /F _x	至少102us	至少51.2us
1	1	1	1	0	0	0	0	至少2 ¹¹ /F _x	至少204us	至少102us
1	1	1	1	1	0	0	0	至少2 ¹³ /F _x	至少819us	至少409us
1	1	1	1	1	1	0	0	至少2 ¹⁵ /F _x	至少3.27ms	至少1.63ms
1	1	1	1	1	1	1	0	至少2 ¹⁷ /F _x	至少13.1ms	至少6.55ms
1	1	1	1	1	1	1	1	至少2 ¹⁸ /F _x	至少26.2ms	至少13.1ms

注意:

1. 在经过上述时间后, 各位从MOST8位开始依次变为“1”并且保持“1”的状态。
2. 振荡稳定时间计数器只在振荡稳定时间选择寄存器(OSTS)所设振荡稳定时间内进行计数。在以下情况下, OSTS寄存器的振荡稳定时间的设置值必须大于通过OSTC寄存器确认的计数值。
 - 当CPU时钟为高速内部振荡器时钟或者副系统时钟并且要开始X1时钟的振荡时
 - 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时(因此必须注意, 解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态)
3. X1时钟的振荡稳定时间不包含时钟开始振荡前的时间(下图a)。



备注: F_x: X1时钟振荡频率

4.3.5 振荡稳定时间选择寄存器(OSTS)

这是选择X1时钟的振荡稳定时间的寄存器。

如果使X1时钟振荡，就在X1振荡电路运行(MSTOP=0)后自动等待OSTS寄存器设置的时间。

如果将CPU时钟从高速内部振荡器时钟或者副系统时钟切换到X1时钟，或者如果CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式，就必须通过振荡稳定时间计数器的状态寄存器(OSTC)确认是否经过振荡稳定时间。

能通过OSTC寄存器确认OSTS寄存器事先设置的时间。

通过8位存储器操作指令设置OSTS寄存器。在产生复位信号后，此寄存器的值变为“07H”。

图4-6：振荡稳定时间选择寄存器(OSTS)的格式

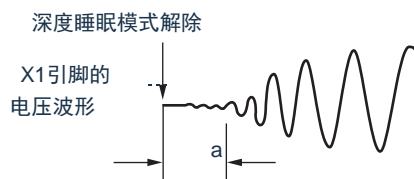
地址：40020403H 复位后：07H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择	
				F _X =10MHz	F _X =20MHz
0	0	0	2 ⁸ /F _X	25.6us	12.8us
0	0	1	2 ⁹ /F _X	51.2us	25.6us
0	1	0	2 ¹⁰ /F _X	102us	51.2us
0	1	1	2 ¹¹ /F _X	204us	102us
1	0	0	2 ¹³ /F _X	819us	409us
1	0	1	2 ¹⁵ /F _X	3.27ms	1.63ms
1	1	0	2 ¹⁷ /F _X	13.1ms	6.55ms
1	1	1	2 ¹⁸ /F _X	26.2ms	13.1ms

注意：

1. 要更改OSTS寄存器的设置时，必须在将时钟运行状态控制寄存器(CSC)的MSTOP位置“0”前进行更改。
2. 振荡稳定时间计数器只在OSTS寄存器所设振荡稳定时间内进行计数。
在以下情况下，OSTS寄存器的振荡稳定时间的设置值必须大于在开始振荡后通过OSTC寄存器确认的计数值。
 - 当CPU时钟为高速内部振荡器时钟或者副系统时钟并且要开始X1时钟的振荡时
 - 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时(因此必须注意，解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态)
3. X1时钟的振荡稳定时间不包含时钟开始振荡前的时间(下图a)。



备注：F_X：X1时钟振荡频率

4.3.6 外围允许寄存器0、1(PER0、PER1)

这是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

当使用由这些寄存器控制的以下外围功能时，必须在进行外围功能的初始设置前将对应位置“1”。

- 实时时钟、15位间隔定时器
- IrDA
- A/D转换器
- 串行接口IICA0
- 通用串行通信单元1
- 通用串行通信单元0
- afcan
- Timer40
- D/A转换器
- 定时器B
- 比较器
- 定时器M
- 增强型DMA
- PWMOP
- 定时器C
- 定时器A

通过8位存储器操作指令设置PER0寄存器和PER1寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图4-7：外围允许寄存器0(PER0)的格式(1/3)

地址：40020420H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN [‡]	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN

RTCEN	提供实时时钟(RTC)和15位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 • 不能写实时时钟(RTC)和15位间隔定时器使用的SFR。 • 实时时钟(RTC)和15位间隔定时器处于复位状态。
1	提供输入时钟。 • 能读写实时时钟(RTC)和15位间隔定时器使用的SFR。

注：RTCEN位只在上电复位时被初始化，而在其他复位时保持不变。

图4-7：外围允许寄存器0(PER0)的格式(2/3)

地址：40020420H 复位后：00H R/W

符号 7 6 5 4 3 2 1 0

PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN
------	-------	--------	-------	--------	--------	--------	--------	--------

IRDAEN	提供串行接口IICA1的输入时钟的控制
0	停止提供输入时钟。 •不能写IRDA使用的SFR。 •IRDA处于复位状态。
1	提供输入时钟。 •能读写IRDA使用的SFR。

ADCEN	提供A/D转换器的输入时钟的控制
0	停止提供输入时钟。 •不能写A/D转换器使用的SFR。 •A/D转换器处于复位状态。
1	提供输入时钟。 •能读写A/D转换器使用的SFR。

IICA0EN	提供串行接口IICA0的输入时钟的控制
0	停止提供输入时钟。 •不能写串行接口IICA0使用的SFR。 •串行接口IICA0处于复位状态。
1	提供输入时钟。 •能读写串行接口IICA0使用的SFR。

SCI1EN	提供通用串行通信单元1的输入时钟的控制
0	停止提供输入时钟。 •不能写通用串行通信单元1使用的SFR。 •通用串行通信单元1处于复位状态。
1	提供输入时钟。 •能读写通用串行通信单元1使用的SFR。

SCI0EN	提供通用串行通信单元0的输入时钟的控制
0	停止提供输入时钟。 •不能写通用串行通信单元0使用的SFR。 •通用串行通信单元0处于复位状态。
1	提供输入时钟。 •能读写通用串行通信单元0使用的SFR。

图4-7：外围允许寄存器0(PER0)的格式(3/3)

地址：40020420H 复位后：00H R/W

符号 7 6 5 4 3 2 1 0

PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN
------	-------	--------	-------	--------	--------	--------	--------	--------

CAN0EN	提供CAN模块的输入时钟的控制
0	停止提供输入时钟。 •不能写CAN0使用的SFR。 •CAN0处于复位状态。
1	提供输入时钟。 •能读写CAN0使用的SFR。

TM40EN	提供Timer4的输入时钟的控制
0	停止提供输入时钟。 •不能写通用定时器单元0使用的SFR。 •通用定时器单元0处于复位状态。
1	提供输入时钟。 •能读写通用定时器单元0使用的SFR。

图4-8: 外围允许寄存器1(PER1)的格式(1/2)

地址: 4002081AH 复位后: 00H R/W

符号 7 6 5 4 3 2 1 0

PER1	DACEN	TMBEN	PGACMPEN	TMMEN ^{注1}	DMAEN	PWMOPEN	TMCEN ^{注2}	TMAEN
------	-------	-------	----------	---------------------	-------	---------	---------------------	-------

DACEN	提供D/A转换器的输入时钟的控制
0	停止提供输入时钟。 •不能写D/A转换器使用的SFR。 •D/A转换器处于复位状态。
1	提供输入时钟。 •能读写D/A转换器使用的SFR。

TMBEN	提供定时器B的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器B使用的SFR。 •定时器B处于复位状态。
1	提供输入时钟。 •能读写定时器B使用的SFR。

PGACMPEN	提供放大器和比较器的输入时钟的控制
0	停止提供输入时钟。 •不能写放大器和比较器使用的SFR。 •放大器比较器处于复位状态。
1	提供输入时钟。 •能读写放大器和比较器使用的SFR。

TMMEN ^注	提供定时器M的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器M使用的SFR。 •定时器M处于复位状态。
1	提供输入时钟。 •能读写定时器M使用的SFR。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将F_{CLK}设置为F_{IH}。如果要将F_{CLK}改为F_{IH}以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注2: 如果选择F_{HOCO}作为TMC的计数源时, 必须在将外围允许寄存器1(PER1)的bit1(TMCEN)置位前将F_{CLK}设置为F_{IH}。如果要将F_{CLK}改为F_{IH}以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit1(TMCEN)后进行更改。

图4-8: 外围允许寄存器1(PER1)的格式(2/2)

地址: 4002081AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN ^{注1}	DMAEN	PWMOPEN	TMCEN ^{注2}	TMAEN

DMAEN	提供DMA的输入时钟的控制
0	停止提供输入时钟。 •DMA不能运行。
1	提供输入时钟。 •DMA能运行。

PWMOPEN	PWM截止电路的输入时钟的控制
0	停止提供输入时钟。 •不能写PWM截止电路使用的SFR。 •PWM截止电路处于复位状态。
1	提供输入时钟。 •能读写PWM截止电路使用的SFR。

TMCEN	提供定时器C的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器C使用的SFR。 •定时器C处于复位状态。
1	提供输入时钟。 •能读写定时器C使用的SFR。

TMAEN	提供定时器A的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器A使用的SFR。 •定时器A处于复位状态。
1	提供输入时钟。 •能读写定时器A使用的SFR。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注2: 如果选择 F_{HOCO} 作为TMC的计数源时, 必须在将外围允许寄存器1(PER1)的bit1(TMCEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit1(TMCEN)后进行更改。

4.3.7 副系统时钟提供模式控制寄存器(OSMC)

OSMC寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

如果将RTCLPC位置“1”，就在深度睡眠模式或者CPU以副系统时钟运行的睡眠模式中停止给实时时钟和15位间隔定时器以外的外围功能提供时钟，因此能降低功耗。

另外，能通过OSMC寄存器选择实时时钟和15位间隔定时器的运行时钟。

通过8位存储器操作指令设置OSMC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图4-9：副系统时钟提供模式控制寄存器(OSMC)的格式

地址：40020423H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	深度睡眠模式和CPU以副系统时钟运行的睡眠模式中的设置
0	允许给外围功能提供副系统时钟 (有关允许运行的外围功能，请参照表27-1~表27-3)。
1	停止给实时时钟和15位间隔定时器以外的外围功能提供副系统时钟。

WUTMMCK0	实时时钟、15位间隔定时器和定时器A的运行时钟的选择
0	<ul style="list-style-type: none"> 副系统时钟为实时时钟和15位间隔定时器的运行时钟。 不能选择低速内部振荡器作为定时器A的计数源。
1	<ul style="list-style-type: none"> 低速内部振荡器时钟为实时时钟和15位间隔定时器的运行时钟。 能选择低速内部振荡器或者副系统时钟作为定时器A的计数源。

4.3.8 高速内部振荡器的频率选择寄存器(HOCODIV)

这是更改选项字节(000C2H)设置的高速内部振荡器频率的寄存器。但是，能选择的频率因选项字节(000C2H)的FRQSEL4位和FRQSEL3位的值而不同。

通过8位存储器操作指令设置HOCODIV寄存器。

在产生复位信号后，此寄存器的值变为选项字节(000C2H)的FRQSEL2~FRQSEL0位的设置值。

图4-10：高速内部振荡器的频率选择寄存器(HOCODIV)的格式

地址：40021C20H 复位后：选项字节(000C2H)的FRQSEL2~FRQSEL0位的设置值 R/W

符号	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

			高速内部振荡器时钟频率的选择			
HOCODIV2	HOCODIV1	HOCODIV0	FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	$F_{IH}=24\text{MHz}$	$F_{IH}=32\text{MHz}$	$F_{IH}=48\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=32\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	0	1	$F_{IH}=12\text{MHz}$	$F_{IH}=16\text{MHz}$	$F_{IH}=24\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=16\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	1	0	$F_{IH}=6\text{MHz}$	$F_{IH}=8\text{MHz}$	$F_{IH}=12\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=8\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	1	1	$F_{IH}=3\text{MHz}$	$F_{IH}=4\text{MHz}$	$F_{IH}=3\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=4\text{MHz}$ $F_{HOCO}=64\text{MHz}$
1	0	0	禁止设置。	$F_{IH}=2\text{MHz}$	禁止设置。	$F_{IH}=2\text{MHz}$ $F_{HOCO}=64\text{MHz}$
1	0	1	禁止设置。	$F_{IH}=1\text{MHz}$	禁止设置。	$F_{IH}=1\text{MHz}$ $F_{HOCO}=64\text{MHz}$
上述以外			禁止设置。			

注意：

1. 必须在选择高速内部振荡器时钟(F_{IH})作为CPU/外围硬件时钟(F_{CLK})的状态下设置HOCODIV寄存器。
2. 在通过HOCODIV寄存器更改频率后，经过以下转移时间之后进行频率切换：
 - 以更改前的频率，最多进行3个时钟的运行。
 - 以更改后的频率，最多等待3个CPU/外围硬件的时钟。

4.3.9 高速内部振荡器的微调寄存器(HIOTRM)

这是校正高速内部振荡器精度的寄存器。能使用高精度的外部时钟输入的定时器等的高速内部振荡器频率的自测量和精度校正。通过8位存储器操作指令设置HIOTRM寄存器。

注意：如果在校正精度后温度和V_{DD}引脚的电压发生变化，频率就发生变化。

在温度和V_{DD}引脚的电压发生变化的情况下，需要在要求频率的精度前或者定期地进行校正。

图4-11：高速内部振荡器的微调寄存器(HIOTRM)的格式

地址：40021C00H 复位后：^注 R/W

	符号	7	6	5	4	3	2	1	0
HIOTRM		0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	高速内部振荡器
0	0	0	0	0	0	最低速
0	0	0	0	0	1	↑
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
• • •						
1	1	1	1	1	0	▼
1	1	1	1	1	1	最高速

注：复位值是发货时的调整值。

备注：HIOTRM寄存器的每1位能对高速内部振荡器的时钟精度进行0.05%左右的校正。

4.4 系统时钟振荡电路

4.4.1 X1振荡电路

X1振荡电路通过连接X1引脚和X2引脚的晶体谐振器或者陶瓷谐振器(1~20MHz)进行振荡。也能输入外部时钟,此时必须给EXCLK引脚输入时钟信号。

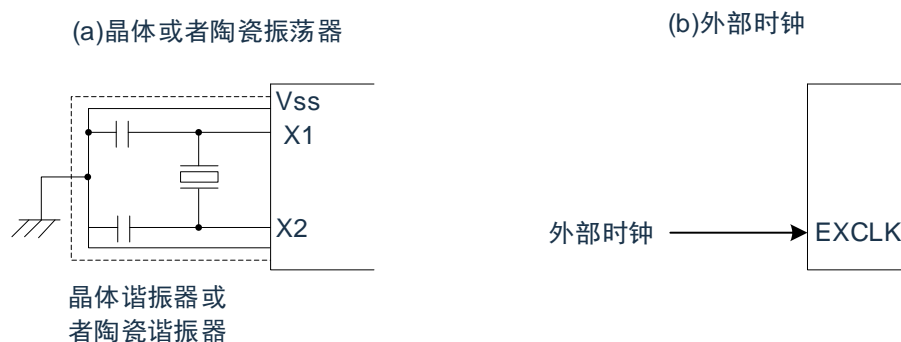
当使用X1振荡电路时,必须对时钟运行模式控制寄存器(CMC)的bit7和bit6(EXCLK、OSCSEL)进行以下的设置:

- 晶体或者陶瓷振荡: EXCLK、OSCSEL=0、1
- 外部时钟输入: EXCLK、OSCSEL=1、1

当不使用X1振荡电路时,必须设置为输入端口模式(EXCLK、OSCSEL=0、0)。而且,当也不用作输入端口时,请参照“表2-3:各未使用引脚的处理”。

X1振荡电路的外接电路例子如图4-12所示。

图4-12: X1振荡电路的外接电路例子



注意事项如下页所示。

4.4.2 XT1振荡电路

XT1振荡电路通过连接XT1引脚和XT2引脚的晶体谐振器(32.768KHz(TYP.))进行振荡。当使用XT1振荡电路时,必须将时钟运行模式控制寄存器(CMC)的bit4(OSCSELS)置“1”也能输入外部时钟,此时必须给EXCLKS引脚输入时钟信号。

当使用XT1振荡电路时,必须对时钟运行模式控制寄存器(CMC)的bit5和bit4(EXCLKS、OSCSELS)进行以下的设置:

- 晶体振荡: EXCLKS、OSCSELS=0、1
- 外部时钟输入: EXCLKS、OSCSELS=1、1

当不使用XT1振荡电路时,必须设置为输入端口模式(EXCLKS、OSCSELS=0、0)。而且,当也不用作输入端口时,请参“表2-3:各未使用引脚的处理”。XT1振荡电路的外接电路例子如图4-13所示。

图4-13: XT1振荡电路的外接电路例子



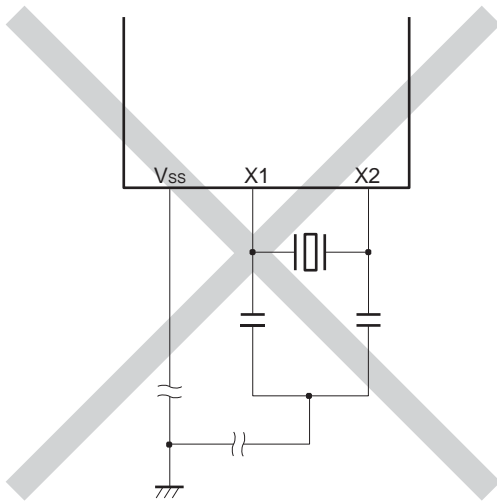
注意：当使用X1振荡电路和XT1振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图4-12和图4-13中的虚线部分进行布线：

- 必须尽量缩短布线。
- 不能和其他的信号线交叉，并且不能接近有变化的大电流流过的布线。
- 必须始终保持振荡电路的电容器接地点和V_{SS}同电位，而且不能给大电流流过的接地图形接地。
- 不能从振荡电路取出信号。

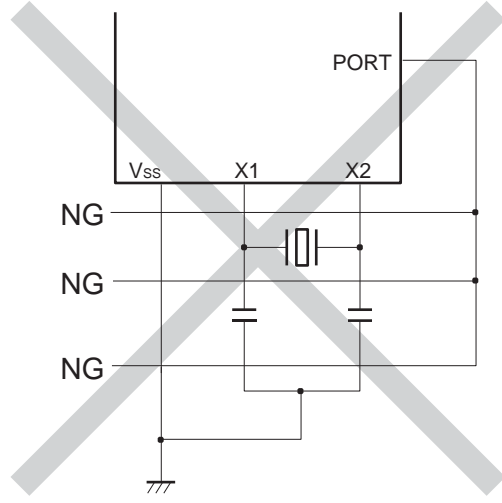
不正确的谐振器连接例子如图4-14所示。

图4-14：不正确的谐振器连接例子(1/2)

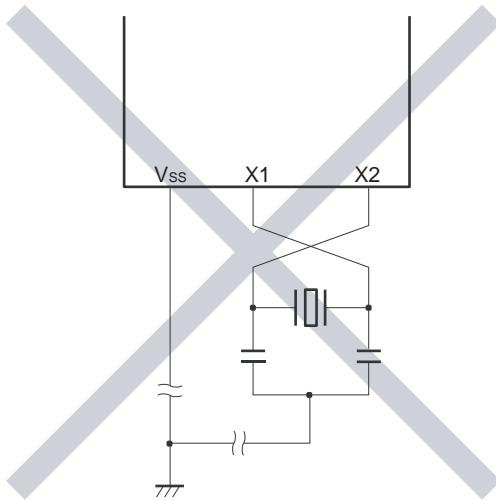
(a)连接电路的布线太长



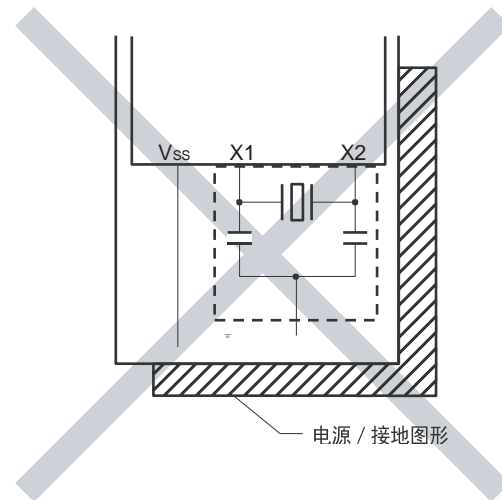
(b)信号线交叉



(c)X1和X2的信号线交叉布线



(d)X1和X2的布线下方有电源或者接地图形

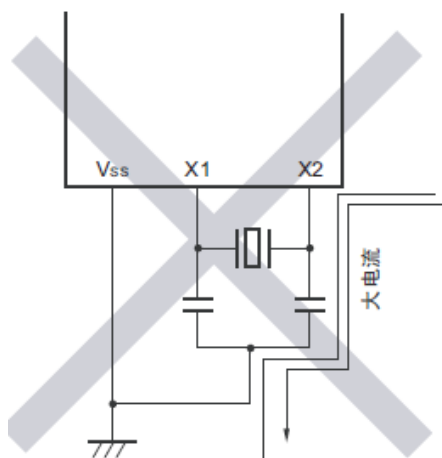


注：在多层板或者双面板中，不能在X1引脚、X2引脚和谐振器的布线区(图中虚线部分)下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

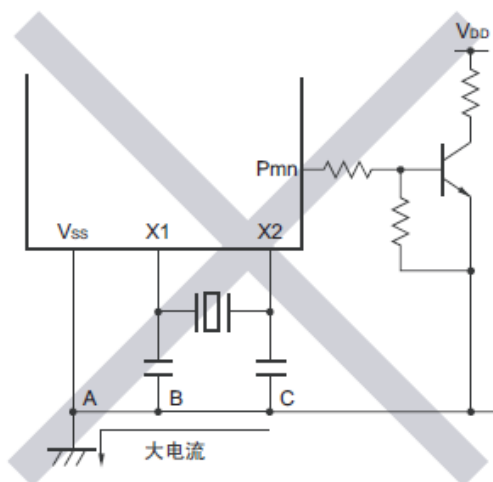
备注：在使用副系统时钟的情况下，请阅读时分别用XT1和XT2代替X1和X2，并且在XT2侧插入串联电阻。

图4-14: 不正确的谐振器连接例子(2/2)

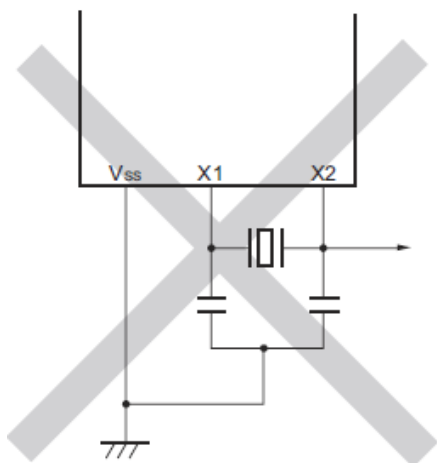
(e)有变化的大电流接近信号线



(f)振荡电路的接地线有电流流过
(A点、B点、C点的电位发生变化)



(g)取出信号



注意：当X2和XT1并行布线时，X2的串扰噪声会叠加到XT1而导致误动作。

备注：在使用副系统时钟的情况下，请阅读时分别用XT1和XT2代替X1和X2，并且在XT2侧插入串联电阻。

4.4.3 高速内部振荡器

BAT32A237内置高速内部振荡器。能通过选项字节(000C2H)从64MHz、48MHz、32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz和1MHz中选择频率。当选择64MHz时，CPU时钟为2分频时钟。能通过时钟运行状态控制寄存器(CSC)的bit0(HIOSTOP)控制振荡。

在解除复位后，高速内部振荡器自动开始振荡。

4.4.4 低速内部振荡器

BAT32A237内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器、实时时钟、15位间隔定时器的时钟和定时器A的时钟，以及SysTick定时器的外部参考时钟，但是不能用作CPU时钟。

当选项字节(000C0H)的bit4(WDTON)或者副系统时钟提供模式控制寄存器(OSMC)的bit4(WUTMMCK0)为“1”时，低速内部振荡器振荡。

当看门狗定时器停止运行并且WUTMMCK0位不为“0”时，低速内部振荡器继续振荡。但是，如果看门狗定时器运行而WUTMMCK0位为“0”，就在WDSTBYON位为“0”并且处于睡眠模式、深度睡眠模式时低速内部振荡器停止振荡。在看门狗定时器运行时，即使程序失控，低速内部振荡器时钟也不停止运行。

4.5 时钟发生电路的运行

时钟发生电路产生以下所示各种时钟，并且控制待机模式等CPU的运行模式(参照图4-1)。

主系统时钟：F_{MAIN}

高速系统时钟：F_{MX}

X1时钟：F_X

外部主系统时钟：F_{EX}

高速内部振荡器时钟：F_{IH}

副系统时钟：F_{SUB}

XT1时钟：F_{XT}

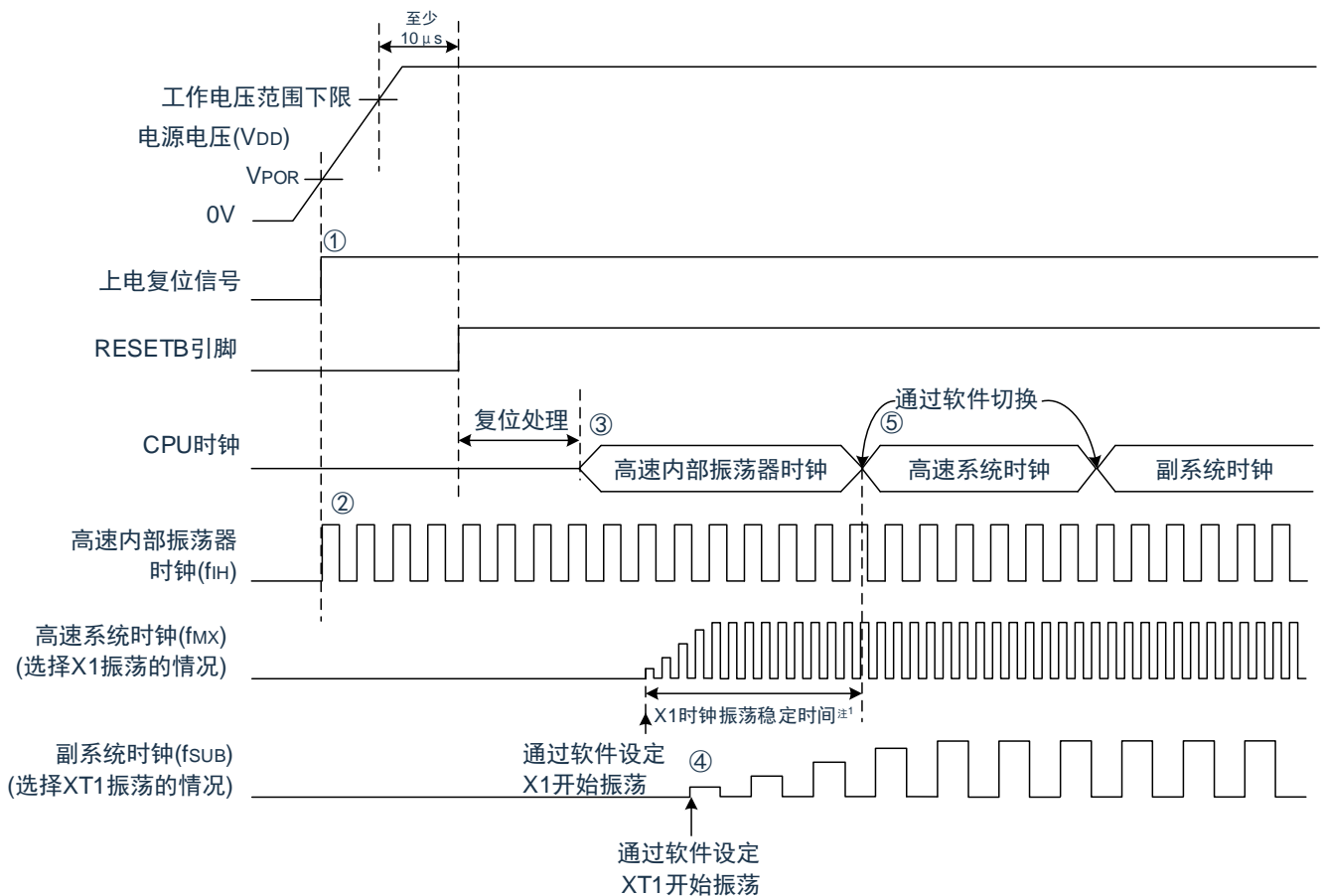
外部副系统时钟：F_{EXS}

低速内部振荡器时钟：F_{IL}

CPU/外围硬件时钟：F_{CLK}

BAT32A237在解除复位后，CPU通过高速内部振荡器的输出开始运行。接通电源时的时钟发生电路的运行如图4-15所示。

图4-15: 接通电源时的时钟发生电路的运行



- ① 在接通电源后，通过上电复位(POR)电路产生内部复位信号。
但是，在达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态(上图是使用外部复位时的例子)。
- ② 如果解除复位，高速内部振荡器就自动开始振荡。
- ③ 在解除复位后，进行电压稳定等待和复位处理，然后CPU以高速内部振荡器时钟开始运行。
- ④ 必须通过软件设置X1时钟或者XT1时钟的开始振荡(参照“4.6.2 X1振荡电路的设置例子”和“4.6.3 XT1振荡电路的设置例子”)。
- ⑤ 如果要将CPU时钟切换到X1时钟或者XT1时钟，就必须在等待时钟振荡稳定后通过软件设置切换(参照“4.6.2 X1振荡电路的设置例子”和“4.6.3 XT1振荡电路的设置例子”)。

注1: 当解除复位时，必须通过振荡稳定时间计数器的状态寄存器(OSTC)确认X1时钟的振荡稳定时间。
注意: 如果使用EXCLK引脚输入的外部时钟，就不需要振荡稳定等待时间。

4.6 时钟控制

4.6.1 高速内部振荡器的设置例子

在解除复位后，CPU/外围硬件时钟(F_{CLK})一定以高速内部振荡器时钟运行。能通过选项字节(000C2H)的FRQSEL0~FRQSEL4位，从64MHz、48MHz、32MHz、24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz和1MHz中选择高速内部振荡器的频率。另外，能通过高速内部振荡器的频率选择寄存器(HOCODIV)更改频率。

【选项字节的设置】

地址：000C2H

选项 字节 (000C2H)	7	6	5	4	3	2	1	0
	1	1	1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的频率	
					F_{HOCO}	F_{IH}
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	48MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	32MHz	16MHz
0	0	0	0	1	24MHz	12MHz
0	1	0	1	0	32MHz	8MHz
0	0	0	1	0	24MHz	6MHz
0	1	0	1	1	32MHz	4MHz
0	0	0	1	1	24MHz	3MHz
0	1	1	0	0	32MHz	2MHz
0	1	1	0	1	32MHz	1MHz
上述以外					禁止设置。	

【高速内部振荡器的频率选择寄存器(HOCODIV)的设置】

地址: 0x40021C20

符号	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	高速内部振荡器时钟频率的选择			
			FRQSEL4=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	F _{IH} =24MHz	F _{IH} =32MHz	F _{IH} =48MHz F _{HOCO} =48MHz	F _{IH} =32MHz F _{HOCO} =64MHz
0	0	1	F _{IH} =12MHz	F _{IH} =16MHz	F _{IH} =24MHz F _{HOCO} =48MHz	F _{IH} =16MHz F _{HOCO} =64MHz
0	1	0	F _{IH} =6MHz	F _{IH} =8MHz	F _{IH} =12MHz F _{HOCO} =48MHz	F _{IH} =8MHz F _{HOCO} =64MHz
0	1	1	F _{IH} =3MHz	F _{IH} =4MHz	F _{IH} =6MHz F _{HOCO} =48MHz	F _{IH} =4MHz F _{HOCO} =64MHz
1	0	0	禁止设置。	F _{IH} =2MHz	F _{IH} =3MHz F _{HOCO} =48MHz	F _{IH} =2MHz F _{HOCO} =64MHz
1	0	1	禁止设置。	F _{IH} =1MHz	禁止设置。	F _{IH} =1MHz F _{HOCO} =64MHz
上述以外			禁止设置。			

4.6.2 X1振荡电路的设置例子

在解除复位后，CPU/外围硬件时钟(F_{CLK})一定以高速内部振荡器时钟运行。此后，如果改为X1振荡时钟，就通过振荡稳定时间选择寄存器(OSTS)、时钟运行模式控制寄存器(CMC)和时钟运行状态控制寄存器(CSC)进行振荡电路的设置和振荡开始的控制，并且通过振荡稳定时间计数器的状态寄存器(OSTC)等待振荡稳定。在等待振荡稳定后通过系统时钟控制寄存器(CKC)将X1振荡时钟设置为 F_{CLK} 。

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

- ① 将CMC寄存器的OSCSSEL位置“1”，当 F_x 大于等于10MHz时，将AMPH位置“1”，使X1振荡电路运行。

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSSEL1	EXCLKS0	OSCSSEL0	0	AMPHS10	AMPHS00	AMPH0/1

- ② 通过OSTS寄存器选择解除深度睡眠模式时的X1振荡电路的振荡稳定时间。

例)要通过10MHz谐振器至少等待102us时，必须设置为以下的值。

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS20	OSTS11	OSTS00

- ③ 将CSC寄存器的MSTOP位清“0”，使X1振荡电路开始振荡。

	7	6	5	4	3	2	1	0
CSC	MSTOP0	XTSTOP1	0	0	0	0	0	HIOSTOP0

- ④ 通过OSTC寄存器等待X1振荡电路的振荡稳定。

例)要通过10MHz谐振器至少等待102us时，必须等到各位变为以下的值。

	7	6	5	4	3	2	1	0
OSTC	MOST81	MOST91	MOST101	MOST110	MOST130	MOST150	MOST170	MOST180

- ⑤ 通过CKC寄存器的MCM0位将X1振荡时钟设置为CPU/外围硬件时钟。

	7	6	5	4	3	2	1	0
CKC	CLS0	CSS0	MCS0	MCM01	0	0	0	0

4.6.3 XT1振荡电路的设置例子

在解除复位后，CPU/外围硬件时钟(F_{CLK})一定以高速内部振荡器时钟运行。此后，如果改为XT1振荡时钟，就通过副系统时钟提供模式控制寄存器(OSMC)、时钟运行模式控制寄存器(CMC)和时钟运行状态控制寄存器(CSC)进行振荡电路的设置和振荡开始的控制，并且通过系统时钟控制寄存器(CKC)将XT1振荡时钟设置为 F_{CLK} 。

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

- ① 在深度睡眠模式或者CPU以副系统时钟运行的睡眠模式中，当只要使实时时钟和15位间隔定时器以副系统时钟运行(超低消费电流)时，必须将RTCLPC位置“1”。

	7	6	5	4	3	2	1	0
OSMC	RTCLPC0/1	0	0	WUTMMCK00	0	0	0	0

- ② 将CMC寄存器的OSCSELS位置“1”，使XT1振荡电路运行。

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL0	EXCLKS0	OSCSELS1	0	AMPHS10/1	AMPHS00/1	AMPHO

AMPHS0位和AMPHS1位：设置XT1振荡电路的振荡模式。

- ③ 将CSC寄存器的XTSTOP位清“0”，使XT1振荡电路开始振荡。

	7	6	5	4	3	2	1	0
CSC	MSTOP1	XTSTOP0	0	0	0	0	0	HIOSTOP0

- ④ 必须通过软件和定时器功能等，等待副系统时钟所需的振荡稳定时间。

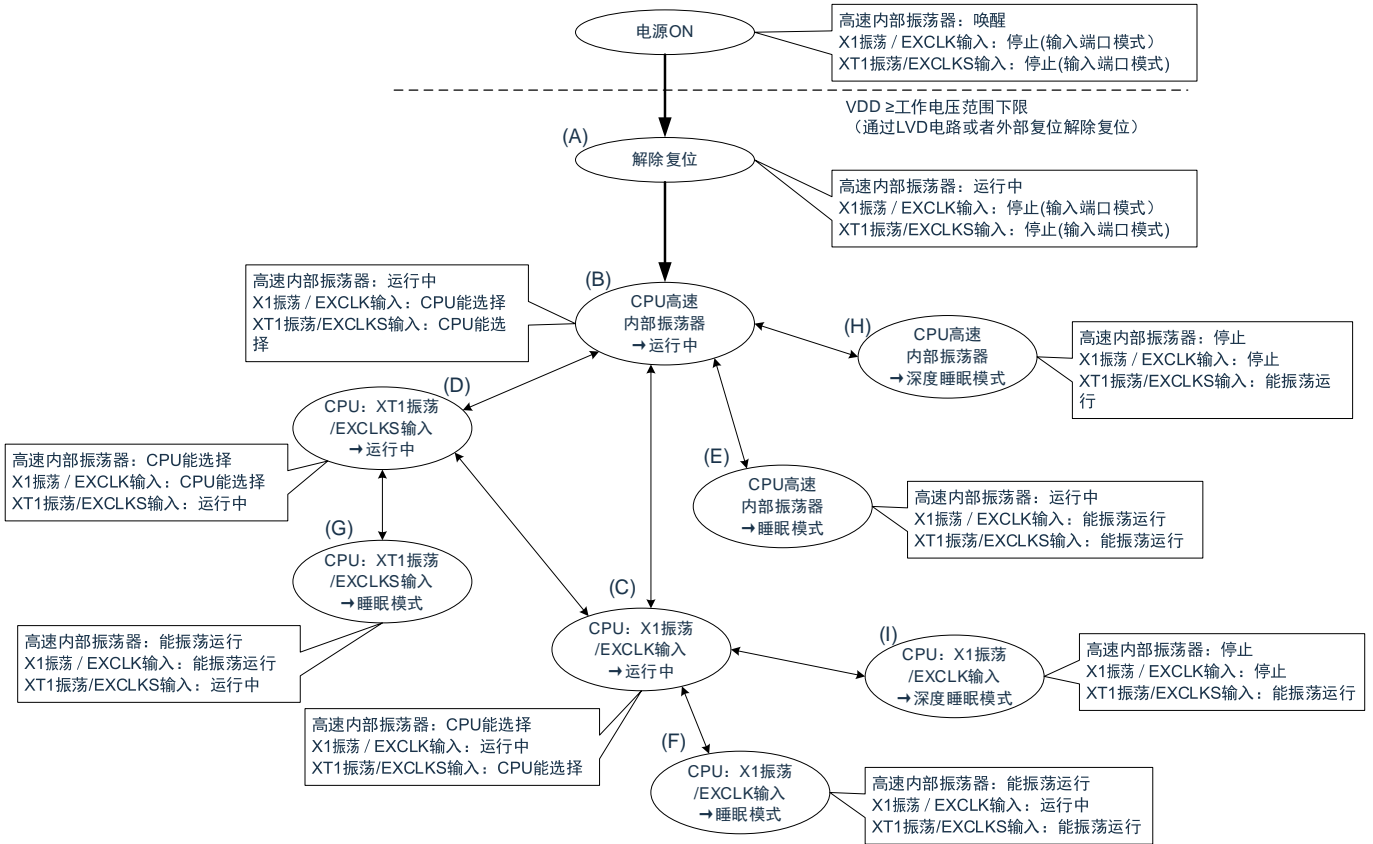
- ⑤ 通过CKC寄存器的CSS位将XT1振荡时钟设置为CPU/外围硬件时钟。

	7	6	5	4	3	2	1	0
CKC	CLS0	CSS0	MCS0	MCM01	0	0	0	0

4.6.4 CPU时钟的状态转移图

本产品的CPU时钟状态转移图如图4-16所示。

图4-16: CPU时钟的状态转移图



CPU时钟的转移和SFR寄存器的设置例子等如表4-3所示。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(1/5)

(1) 在解除复位(A)后, CPU转移到高速内部振荡器时钟运行(B)。

状态转移	SFR寄存器的设置
(A)→(B)	不需要设置SFR寄存器(解除复位后的初始状态)。

(2) 在解除复位(A)后, CPU转移到高速系统时钟运行(C)。

(CPU在解除复位后立即以高速内部振荡器时钟运行(B))

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志			CMC寄存器 ^{注1}			OSTS 寄存器	CSC 寄存器 MSTOP	OSTC 寄存器	CKC 寄存器 MCM0
	EXCLK	OSCSEL	AMPH	EXCLK	OSCSEL	AMPH				
(A)→(B)→(C) (X1时钟: $1\text{MHz} \leq F_x \leq 10\text{MHz}$)	0	1	0				^{注2}	0	需要确认	1
(A)→(B)→(C) (X1时钟: $10\text{MHz} < F_x \leq 20\text{MHz}$)	0	1	1				^{注2}	0	需要确认	1
(A)→(B)→(C) (外部主时钟)	1	1	×				^{注2}	0	不需要确认	1

注1: 在解除复位后, 只能通过8位存储器操作指令写1次时钟运行模式控制寄存器(CMC)。

注2: 必须对振荡稳定时间选择寄存器(OSTS)的振荡稳定时间进行以下的设置:

- 期待的振荡稳定时间计数器的状态寄存器(OSTC)的振荡稳定时间 ≤ OSTS寄存器设置的振荡稳定时间

注意: 必须在电源电压达到设置的时钟可运行电压(参照数据手册)后设置时钟。

(3) 在解除复位(A)后, CPU转移到副系统时钟运行(D)。

(CPU在解除复位后立即以高速内部振荡器时钟运行(B))

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志				CMC寄存器 ^注				CSC 寄存器 XTSTOP	振荡稳定的 等待	CKC 寄存器 CSS
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	EXCLKS	OSCSELS	AMPHS1	AMPHS0			
(A)→(B)→(D) (XT1时钟)	0	1	0/1	0/1					0	需要	1
(A)→(B)→(D) (外部副时钟)	1	1	×	×					0	需要	1

注: 在解除复位后, 只能通过8位存储器操作指令写1次时钟运行模式控制寄存器(CMC)。

备注:

1. ×: 忽略
2. 表4-3的(A)~(I)对应图4-16的(A)~(I)。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(2/5)

(4) CPU从高速内部振荡器时钟运行(B)转移到高速系统时钟运行(C)。

(SFR寄存器的设置顺序) →

SFR寄存器的设置标志 状态转移	CMC寄存器 ^{注1}			OSTS 寄存器	CSC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B)→(C) (X1时钟: $1\text{MHz} \leq F_x \leq 10\text{MHz}$)	0	1	0	^{注2}	0	需要确认	1
(B)→(C) (X1时钟: $10\text{MHz} < F_x \leq 20\text{MHz}$)	0	1	1	^{注2}	0	需要确认	1
(B)→(C) (外部主时钟)	1	1	×	^{注2}	0	不需要确认	1

如果已设置就不需要。 在高速系统时钟运行中不需要。

注1: 在解除复位后, 只能设置1次时钟运行模式控制寄存器(CMC)。如果已设置就不需要。

注2: 必须对振荡稳定时间选择寄存器(OSTS)的振荡稳定时间进行以下的设置:

- 期待的振荡稳定时间计数器的状态寄存器(OSTC)的振荡稳定时间 ≤ OSTC寄存器设置的振荡稳定时间

注意: 必须在电源电压达到设置的时钟可运行电压(参照数据手册)后设置时钟。

(5) CPU从高速内部振荡器时钟运行(B)转移到副系统时钟运行(D)。

(SFR寄存器的设置顺序) →

SFR寄存器的设置标志 状态转移	CMC寄存器 ^注			CSC 寄存器	振荡稳定的等待	CKC 寄存器
	EXCLKS	OSCSELS	AMPHS1、0	XTSTOP		CSS
(B)→(D) (XT1时钟)	0	1	00: 低功耗振荡 01: 通常振荡 10: 超低功耗振荡	0	需要	1
(B)→(D) (外部副时钟)	1	1	×	0	需要	1

如果已设置就不需要。 在副系统时钟运行中不需要。

注: 在解除复位后, 只能通过8位存储器操作指令写1次时钟运行模式控制寄存器(CMC)。如果已设置就不需要。

备注:

1. ×: 忽略
2. 表4-3的(A)~(I)对应图4-16的(A)~(I)。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(3/5)

(6) CPU从高速系统时钟运行(C)转移到高速内部振荡器时钟运行(B)。

(SFR寄存器的设置顺序)

状态转移	SFR寄存器的设置标志	CSC寄存器	振荡精度稳定的等待	CKC寄存器
		HIOSTOP		
(C)→(B)		0	注	0

在高速内部振荡器时钟运行中不需要。

注: FRQSEL4=0时: 45us~65us

FRQSEL4=1时: 45us~135us

备注: 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(4/4)

(7) CPU从高速系统时钟运行(C)转移到副系统时钟运行(D)。

(SFR寄存器的设置顺序)

状态转移	SFR寄存器的设置标志	CSC寄存器	振荡精度稳定的等待	CKC寄存器
		XTSTOP		
(C)→(D)		0	需要	1

在副系统时钟运行中不需要。

(8) CPU从副系统时钟运行(D)转移到高速内部振荡器时钟运行(B)。

(SFR寄存器的设置顺序)

状态转移	SFR寄存器的设置标志	CSC寄存器	振荡精度稳定的等待	CKC寄存器
		HIOSTOP		
(D)→(B)		0	注	0

在高速内部振荡器时钟运行中不需要。

注: FRQSEL4=0时: 45us~65us

FRQSEL4=1时: 45us~135us

备注:

1. 表4-3的(A)~(I)对应图4-16的(A)~(I)。
2. 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(4/5)

(9) CPU从副系统时钟运行(D)转移到高速系统时钟运行(C)。

(SFR寄存器的设置顺序)

状态转移	SFR寄存器的设置标志	OSTS寄存器	CSC寄存器	OSTC寄存器	CKC寄存器
			MSTOP		CSS
(D)→(C) (X1时钟: $1\text{MHz} \leq F_x \leq 10\text{MHz}$)		注	0	需要确认	0
(D)→(C) (X1时钟: $10\text{MHz} < F_x \leq 20\text{MHz}$)		注	0	需要确认	0
(D)→(C) (外部主时钟)		注	0	不需要确认	0

在高速系统时钟运行中不需要。

注: 必须对振荡稳定时间选择寄存器(OSTS)的振荡稳定时间进行以下的设置:

期待的振荡稳定时间计数器的状态寄存器(OSTC)的振荡稳定时间 ≤ OSTS寄存器设置的振荡稳定时间

注意: 必须在电源电压达到设置的时钟可运行电压(参照数据手册)后设置时钟。

(10) CPU在高速内部振荡器时钟运行中(B)转移到睡眠模式(E)。

CPU在高速系统时钟运行中(C)转移到睡眠模式(F)。

CPU在副系统时钟运行中(D)转移到睡眠模式(G)。

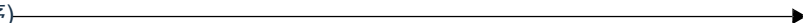
状态转移	设置内容
(B)→(E) (C)→(F) (D)→(G)	执行WFI指令。

备注: 表4-3的(A)~(I)对应图4-16的(A)~(I)。

表4-3: CPU时钟的转移和SFR寄存器的设置例子(5/5)

(11) CPU在高速内部振荡器时钟运行中(B)转移到深度睡眠模式(H)。

CPU在高速系统时钟运行中(C)转移到深度睡眠模式(I)。

(设置顺序) 

状态转移		设置内容		
(B)→(H)		停止	—	SCR寄存器bit2(SLEEPDEEP)置为1, 并执行WFI指令。
(C)→(I)	X1振荡	不能在深度睡眠模式中运行的外围功能。	设置OSTS寄存器。	
	外部时钟		—	

备注: 表4-3的(A)~(I)对应图4-16的(A)~(I)。

4.6.5 CPU时钟转移前的条件和转移后的处理

CPU时钟转移前的条件和转移后的处理如下所示。

表4-4：有关CPU时钟的转移(1/3)

CPU时钟		转移前的条件	转移后的处理
转移前	转移后		
高速内部振荡器时钟	X1时钟	X1振荡稳定。 • OSCSEL=1, EXCLK=0, MSTOP=0 • 经过振荡稳定时间后	如果停止高速内部振荡器的振荡(HIOSTOP=1), 就能减小工作电流。
	外部主系统时钟	将EXCLK引脚输入的外部时钟置为有效。 • OSCSEL=1, EXCLK=1, MSTOP=0	
	XT1时钟	XT1振荡稳定。 • OSCSELS=1, EXCLKS=0, XTSTOP=0 • 经过振荡稳定时间后	
	外部副系统时钟	将EXCLKS引脚输入的外部时钟置为有效。 • OSCSELS=1, EXCLKS=1, XTSTOP=0	
X1时钟	高速内部振荡器时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	能停止X1的振荡(MSTOP=1)。
	外部主系统时钟	不能转移。	—
	XT1时钟	XT1振荡稳定。 • OSCSELS=1, EXCLKS=0, XTSTOP=0 • 经过振荡稳定时间后	能停止X1的振荡(MSTOP=1)。
	外部副系统时钟	将EXCLKS引脚输入的外部时钟置为有效。 • OSCSELS=1, EXCLKS=1, XTSTOP=0	能停止X1的振荡(MSTOP=1)。
外部主系统时钟	高速内部振荡器时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	能将外部主系统时钟的输入置为无效(MSTOP=1)。
	X1时钟	不能转移。	—
	XT1时钟	XT1振荡稳定。 • OSCSELS=1, EXCLKS=0, XTSTOP=0 • 经过振荡稳定时间后	能将外部主系统时钟的输入置为无效(MSTOP=1)。
	外部副系统时钟	将EXCLKS引脚输入的外部时钟置为有效。 • OSCSELS=1, EXCLKS=1, XTSTOP=0	能将外部主系统时钟的输入置为无效(MSTOP=1)。

表4-4：有关CPU时钟的转移(2/2)

CPU时钟		转移前的条件	转移后的处理
转移前	转移后		
XT1时钟	高速内部振荡器时钟	高速内部振荡器正在振荡并且选择高速内部振荡器时钟作为主系统时钟。 • HIOSTOP=0, MCS=0	能停止XT1的振荡(XTSTOP=1)。
	X1时钟	X1振荡稳定并且选择高速系统时钟作为主系统时钟。 • OSCSEL=1, EXCLK=0, MSTOP=0 • 经过振荡稳定时间后 • MCS=1	
	外部主系统时钟	将EXCLK引脚输入的外部时钟置为有效并且选择高速系统时钟作为主系统时钟。 • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	外部副系统时钟	不能转移。	—
外部副系统时钟	高速内部振荡器时钟	高速内部振荡器正在振荡并且选择高速内部振荡器时钟作为主系统时钟。 • HIOSTOP=0, MCS=0	能将外部副系统时钟的输入置为无效(XTSTOP=1)。
	X1时钟	X1振荡稳定并且选择高速系统时钟作为主系统时钟。 • OSCSEL=1, EXCLK=0, MSTOP=0 • 经过振荡稳定时间后 • MCS=1	
	外部主系统时钟	将EXCLK引脚输入的外部时钟置为有效并且选择高速系统时钟作为主系统时钟。 • OSCSEL=1, EXCLK=1, MSTOP=0 • MCS=1	
	XT1时钟	不能转移。	—

4.6.6 CPU时钟和主系统时钟的切换所需时间

能通过设置系统时钟控制寄存器(CKC)的bit6和bit4(CSS、MCM0)进行CPU时钟的切换(主系统时钟↔副系统时钟)和主系统时钟的切换(高速内部振荡器时钟↔高速系统时钟)。

在改写CKC寄存器后不立即进行实际的切换，而是在更改CKC寄存器后仍然以切换前的时钟继续运行数个时钟(参照表4-5~表4-7)。

能通过CKC寄存器的bit7(CLS)来判断CPU是以主系统时钟还是以副系统时钟运行。能通过CKC寄存器的bit5(MCS)来判断主系统时钟是以高速系统时钟还是以高速内部振荡器时钟运行。

如果切换CPU时钟，就同时切换外围硬件时钟。

表4-5：切换主系统时钟所需要的最长时间

时钟A	切换方向	时钟B	备注
F _{IH}	↔	F _{MX}	参照表4-6。
F _{MAIN}	↔	F _{SUB}	参照表4-7。

表4-6：F_{IH}↔F_{MX}所需要的最大时钟数

切换前的设置值		切换后的设置值	
MCM0		MCM0	
		0 (F _{MAIN} =F _{IH})	1 (F _{MAIN} =F _{MX})
0 (F _{MAIN} =F _{IH})	F _{MX} ≥ F _{IH}		2个时钟
	F _{MX} < F _{IH}		2个F _{IH} /F _{MX} 时钟
1 (F _{MAIN} =F _{MX})	F _{MX} ≥ F _{IH}	2个F _{MX} /F _{IH} 时钟	
	F _{MX} < F _{IH}	2个时钟	

表4-7：F_{MAIN}↔F_{SUB}所需要的最大时钟数

切换前的设置值		切换后的设置值	
CSS		CSS	
		0 (F _{CLK} =F _{MAIN})	1 (F _{CLK} =F _{SUB})
0 (F _{CLK} =F _{MAIN})			1+2个F _{MAIN} /F _{SUB} 时钟
1 (F _{CLK} =F _{SUB})		3个时钟	

备注：

1. 表4-6和表4-7中的时钟数是切换前的CPU时钟数。
2. 表4-6和表4-7中的时钟数是舍入小数部分的时钟数。

例主系统时钟从高速系统时钟切换到高速内部振荡器时钟的情况(选择F_{IH}=8MHz、F_{MX}=10MHz振荡的情况)

$$2F_{MX}/F_{IH}=2(10/8)=2.5 \rightarrow 3 \text{个时钟}$$

4.6.7 时钟振荡停止前的条件

用于停止时钟振荡(外部时钟输入无效)的寄存器标志设置和停止前的条件如下所示。

表4-8: 时钟振荡停止前的条件和标志设置

时钟	时钟停止前的条件(外部时钟输入无效)	SFR寄存器的标志设置
高速内部振荡器时钟	MCS=1或者CLS=1 (CPU以高速内部振荡器时钟以外的时钟运行)	HIOSTOP=1
X1时钟	MCS=0或者CLS=1	MSTOP=1
外部主系统时钟	(CPU以高速系统时钟以外的时钟运行)	
XT1时钟	CLS=0	XTSTOP=1
外部副系统时钟	(CPU以副系统时钟以外的时钟运行)	

第5章 硬件除法器

硬件除法器是为了支持高性能运算而搭载的专用硬件。该硬件除法器是32位有符号整数除法器，输出32位有符号的商和余数结果。

5.1 特点

- 32位有符号(2的补码)整数除法计算
- 32位有符号被除数，32位有符号除数
- 32位有符号商和32位有符号余数输出
- 写除数寄存器自动触发除法计算
- 除0警告标志
- 指示运算中的BUSY标志
- 有计算结束的中断请求
- 4或8个CPU时钟周期完成一次计算
 - 倍速状态下4个CPU时钟周期完成一次计算
 - 非倍速状态下8个CPU时钟周期完成一次计算

5.2 功能描述

使用硬件除法器时，需要先设置被除数寄存器(DIVIDEND)，然后再设置除数寄存器(DIVISOR)，因为对除数寄存器的写操作会自动触发除法计算。通过查询状态寄存器(STATUS)的BUSY位或使用计算结束的中断可以知道什么时候结算结束。计算结果可以通过商(QUOTIENT)和余数(REMAINDER)寄存器读出。

注意：在计算期间请不要写被除数或除数寄存器，也不要读商或余数寄存器，否则结果不可预知。

5.3 硬件除法器的寄存器

硬件除法器的寄存器如下所示：

寄存器基地址：DIV_BASE = 4008_0000H；

寄存器名称	寄存器描述	R/W	复位值	寄存器地址
DIVIDEND	被除数寄存器	R/W	0000_0000H	DIV_BASE+00H
DIVISOR	除数寄存器	R/W	0000_0000H	DIV_BASE+04H
QUOTIENT	商寄存器	R	0000_0000H	DIV_BASE+08H
REMAINDER	余数寄存器	R	0000_0000H	DIV_BASE+0CH
STATUS	状态寄存器	R	0000_0000H	DIV_BASE+10H

R: read only, W: write only, R/W: both read and write

5.3.1 被除数寄存器(DIVIDEND)

被除数寄存器是保存被除数的寄存器，其数值作为32位有符号整数参与除法运算。

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

5.3.2 除数寄存器(DIVISOR)

除数寄存器是保存除数的寄存器，其数值作为32位有符号整数参与除法运算。对该寄存器的写操作会自动触发除法计算。

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

5.3.3 商寄存器(QUOTIENT)

该寄存器在除法计算完成后，保存除法计算结果的商，其数值作为32位有符号整数。

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

5.3.4 余数寄存器(REMAINDER)

该寄存器在除法计算完成后，保存除法计算结果的余数，其数值作为32位有符号整数。

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

5.3.5 状态寄存器(STATUS)

通过状态寄存器可以查询硬件除法器的状态，包括除零标志和BUSY标志。

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

DIVBYZERO	用于指示除零的情况，每次写除数寄存器时更新。
0	除数不是0。
1	除数是0

BUSY	用于指示除法运算的状态。
0	除法运算完成
1	除数运算进行中

第6章 通用定时器单元Timer4

本制品搭载一个通用定时器单元，含有4个通道。通用定时器单元的通道数因产品而不同。参见下表：

表6-1：各产品具有的定时器通道

单元	通道	24 引脚	32,36,40,48,52,64引脚
单元0	通道0	○	○
	通道1	-	○
	通道2	-	○
	通道3	○	○

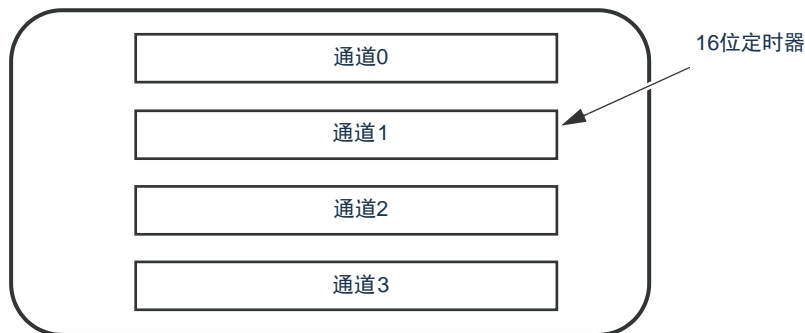
说明：

1. 本章下文中的标号“m”代表单元号，当搭载多个Timer4单元时有区分，本产品只搭载了一个通用定时器Timer4，固定 m=0。
2. 本章下文中的标号“n”代表通道号 (在本章中 n=0 ~ 3)，通道 0 ~ 3 的定时器输入 / 输出引脚的有无因产品而不同。详细内容请参照“表6-3：各产品具有的定时器输入/输出引脚”。
3. 本章的下述内容主要针对64引脚产品进行说明。

通用定时器单元有4个16位定时器。

各16位定时器称为“通道”，既能分别用作独立的定时器，也能组合多个通道用作高级的定时器功能。

通用定时器单元



有关各功能的详细内容，请参照下表。

独立通道运行功能	多通道联动运行功能
<ul style="list-style-type: none"> • 间隔定时器(→参照6.8.1) • 方波输出(→参照6.8.1) • 外部事件计数器(→参照6.8.2) • 分频器^注(→参照6.8.3) • 输入脉冲间隔的测量(→参照6.8.4) • 输入信号的高低电平宽度的测量(→参照6.8.5) • 延迟计数器(→参照6.8.6) 	<ul style="list-style-type: none"> • 单触发脉冲输出(→参照6.9.1) • PWM 输出(→参照6.9.2) • 多重PWM输出(→参照6.9.3)

注：只限于通用定时器单元 0 的通道 0。

能将单元0的通道1和通道3的16位定时器用作2个8位定时器(高位和低位)。通道1和通道3能用作 8位定时器的功能如下：

- 间隔定时器(高8位和低8位定时器)/方波输出(只限于低 8位定时器)
- 外部事件计数器(只限于低8位定时器)
- 延迟计数器(只限于低8位定时器)

能通过单元0的通道3和通用串行通信单元的UART0的协调，实现LIN-bus通信。

6.1 通用定时器单元的功能

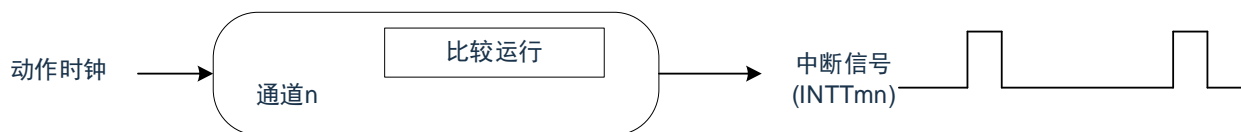
通用定时器单元有以下功能：

6.1.1 独立通道运行功能

独立通道运行功能是不受其他通道运行模式的影响而能独立使用任意通道的功能。

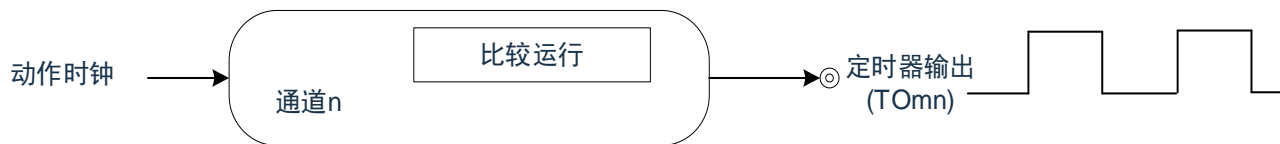
(1) 间隔定时器

能用作以固定间隔产生中断(INTTmn)的基准定时器。



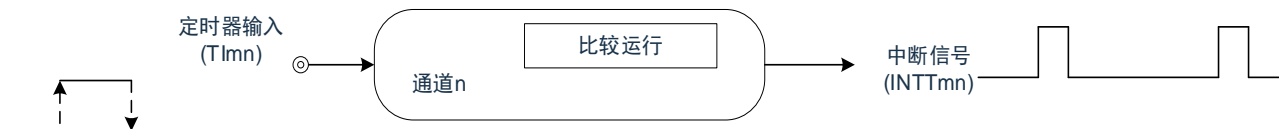
(2) 方波输出

每当产生 INTTmn中断时，就进行交替运行并且从定时器的输出引脚(TOmn)输出50%占空比的方波。



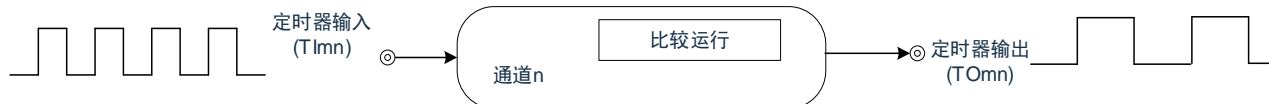
(3) 外部事件计数器

对定时器输入引脚(TImn)的输入信号的有效边沿进行计数，如果达到规定次数，就能用作产生中断的事件计数器。



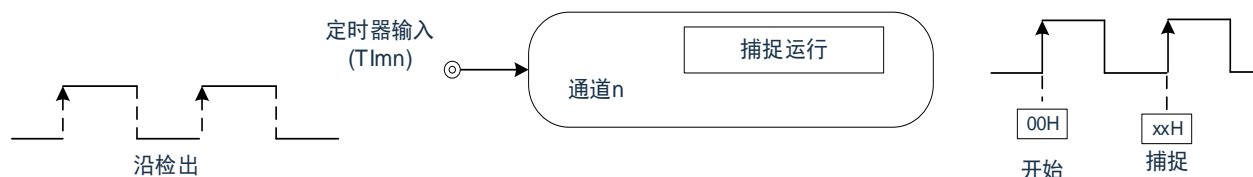
(4) 分频器功能(只限于单元0的通道0)

对定时器输入引脚(TI00)的输入时钟进行分频，然后从输出引脚(TO00)输出。



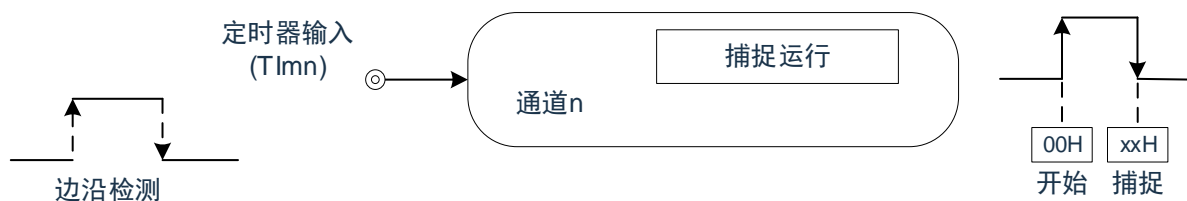
(5) 输入脉冲间隔的测量

在定时器输入引脚(TImn)的输入脉冲信号的有效边沿开始计数并且在下一个脉冲的有效边沿捕捉计数值，从而测量输入脉冲的间隔。



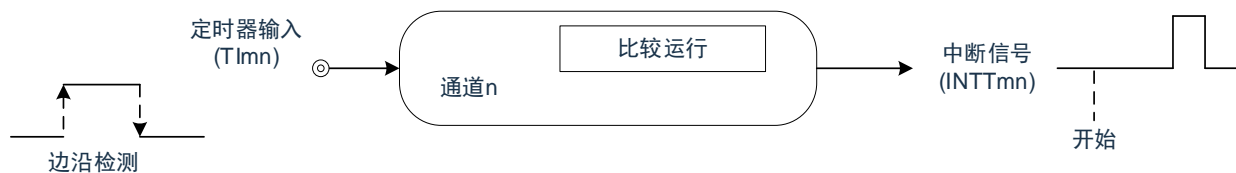
(6) 输入信号的高低电平宽度的测量

在定时器输入引脚(TImn)的输入信号的一个边沿开始计数并且在另一个边沿捕捉计数值，从而测量输入信号的高低电平宽度。



(7) 延迟计数器

在定时器输入引脚(TImn)的输入信号的有效边沿开始计数并且在经过任意延迟期间后产生中断。



备注:

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. 通道 0 ~ 3 的定时器输入 / 输出引脚的有无因产品而不同。详细内容请参照“表6-3: 各产品具有的定时器输入/输出引脚”。

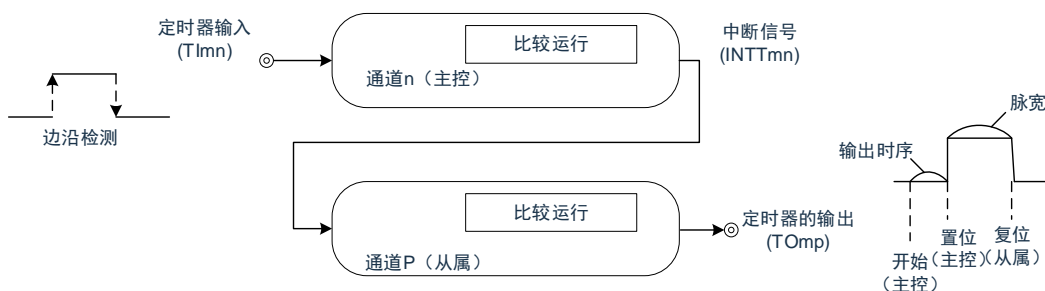
6.1.2 多通道联动运行功能

多通道联动运行功能是将主控通道(主要控制周期的基准定时器)和从属通道(遵从主控通道运行的定时器)组合实现的功能。

多通道联动运行功能可用作以下模式。

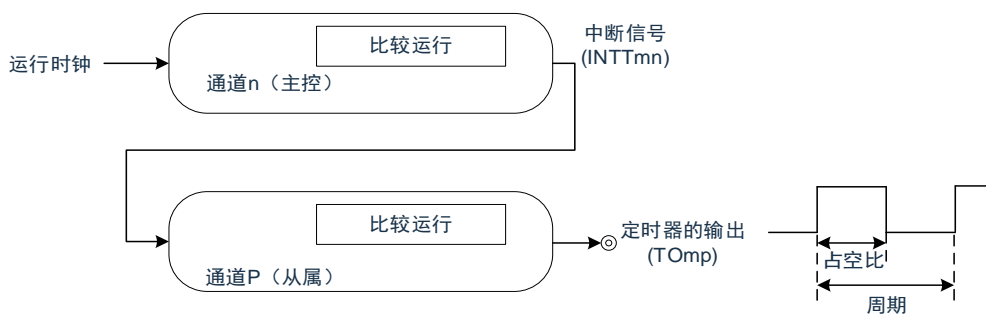
(1) 单触发脉冲输出

将2个通道成对使用，生成能任意设置输出时序和脉宽的单触发脉冲。



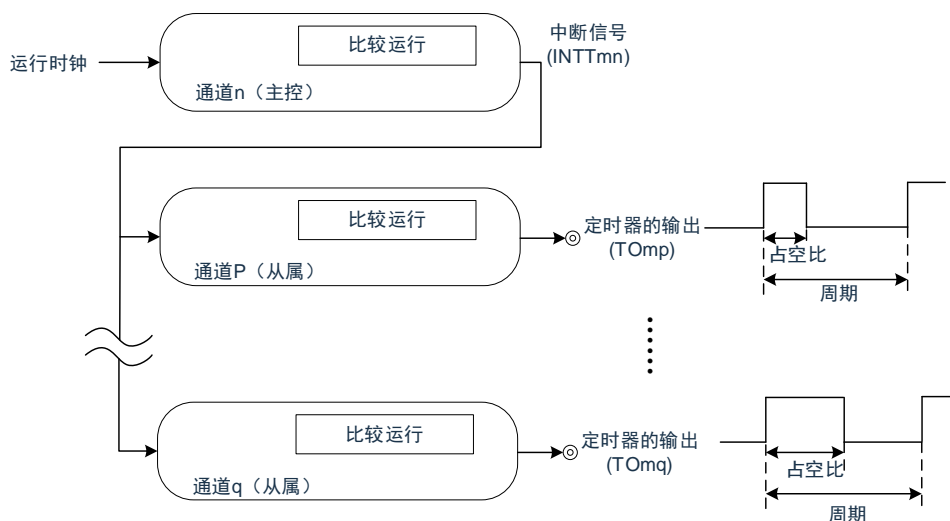
(2) PWM(Pulse Width Modulation)输出

将2个通道成对使用，生成能任意设置周期和占空比的脉冲。



(3) 多重 PWM(Pulse Width Modulation)输出

能通过扩展 PWM 功能并且使用1个主控通道和多个从属通道，以固定周期生成最多3种任意占空比的 PWM信号。



注意：有关多通道联动运行功能规则的详细内容，请参照“6.4.1 多通道联动运行功能的基本规则”。

备注：m：单元号(m=0,1) n：通道号(n=0~3)p、q：从属通道号(n<p<q≤3)

6.1.3 8位定时器运行功能(只限于单元0的通道1和通道3)

8位定时器运行功能是将16位定时器通道用作2个 8位定时器通道的功能。只能使用单元0的通道1和通道3。

注意：在使用 8位定时器运行功能时，有几个规则。

详细内容请参照“6.4.2 8 位定时器运行功能的基本规则(只限于通道1和通道3)”。

6.1.4 LIN-bus支持功能(只限于单元0的通道3)

通过通用定时器单元检查LIN-bus通信中的接收信号是否适合LIN-bus通信表格。

(1) 唤醒信号的检测

在 UART0 串行数据输入引脚(RxD0)的输入信号的下降沿开始计数并且在上升沿捕捉计数值，从而测量低电平宽度。如果该低电平宽度大于等于某固定值，就认为是唤醒信号。

(2) 间隔段的检测

在检测到唤醒信号后，从UART0串行数据输入引脚(RxD0)的输入信号的下降沿开始计数并且在上升沿捕捉计数值，从而测量低电平宽度。如果该低电平宽度大于等于某固定值，就认为是间隔段。

(3) 同步段脉宽的测量

在检测到间隔段后，测量 UART0 串行数据输入引脚(RxD0)的输入信号的低电平宽度和高电平宽度。根据以此方式测量的同步段的位间隔，计算波特率。

备注：有关 LIN-bus支持功能的运行设置，请参照“6.3.13 输入切换控制寄存器(ISC)”和“6.8.5 作为输入信号高低电平 宽度测量的运行”。

6.2 通用定时器单元的结构

通用定时器单元由以下硬件构成。

表6-2: 通用定时器单元的结构

项目	结构
计数器	定时器计数寄存器mn(TCRmn)
寄存器	定时器数据寄存器mn(TDRmn)
定时器的输入	TI00~TI03 ^{注1} 、RxD0引脚(用于LIN-bus)
定时器的输出	TO00~TO03 ^{注1} 、输出控制电路
控制寄存器	<单元设置部的寄存器> <ul style="list-style-type: none"> • 外围允许寄存器0(PER0) • 定时器时钟选择寄存器m(TPSm) • 定时器通道允许状态寄存器m(TEm) • 定时器通道开始寄存器m(TSm) • 定时器通道停止寄存器m(TTm) • 定时器输入选择寄存器0(TIS0) • 定时器输出允许寄存器m(TOEm) • 定时器输出寄存器m(TOm) • 定时器输出电平寄存器m(TOLm) • 定时器输出模式寄存器m(TOMm)
	<每个通道的寄存器> <ul style="list-style-type: none"> • 定时器模式寄存器mn(TMRmn) • 定时器状态寄存器mn(TSRmn) • 输入切换控制寄存器(ISC) • 噪声滤波器允许寄存器1、2(NFEN1、NFEN2) • 端口模式控制寄存器(PMCxx)^{注2} • 端口模式寄存器(PMxx)^{注2} • 端口寄存器(Pxx)^{注2}

注1: 通道 0 ~ 3 的定时器输入 / 输出引脚的有无因产品而不同。详细内容请参照“表6-3: 各产品具有的定时器输入/输出引脚”。

注2: 设置的端口模式控制寄存器(PMCxx)、端口模式寄存器(PMxx)和端口寄存器(Pxx)因产品而不同。详细内容请参照“2.5 使用复用功能时的寄存器设置”。

备注: m: 单元号(m=0)n: 通道号(n=0~3)

通用定时器单元的各通道的定时器输入/输出引脚的有无因产品而不同。

表6-3: 各产品具有的定时器输入/输出引脚

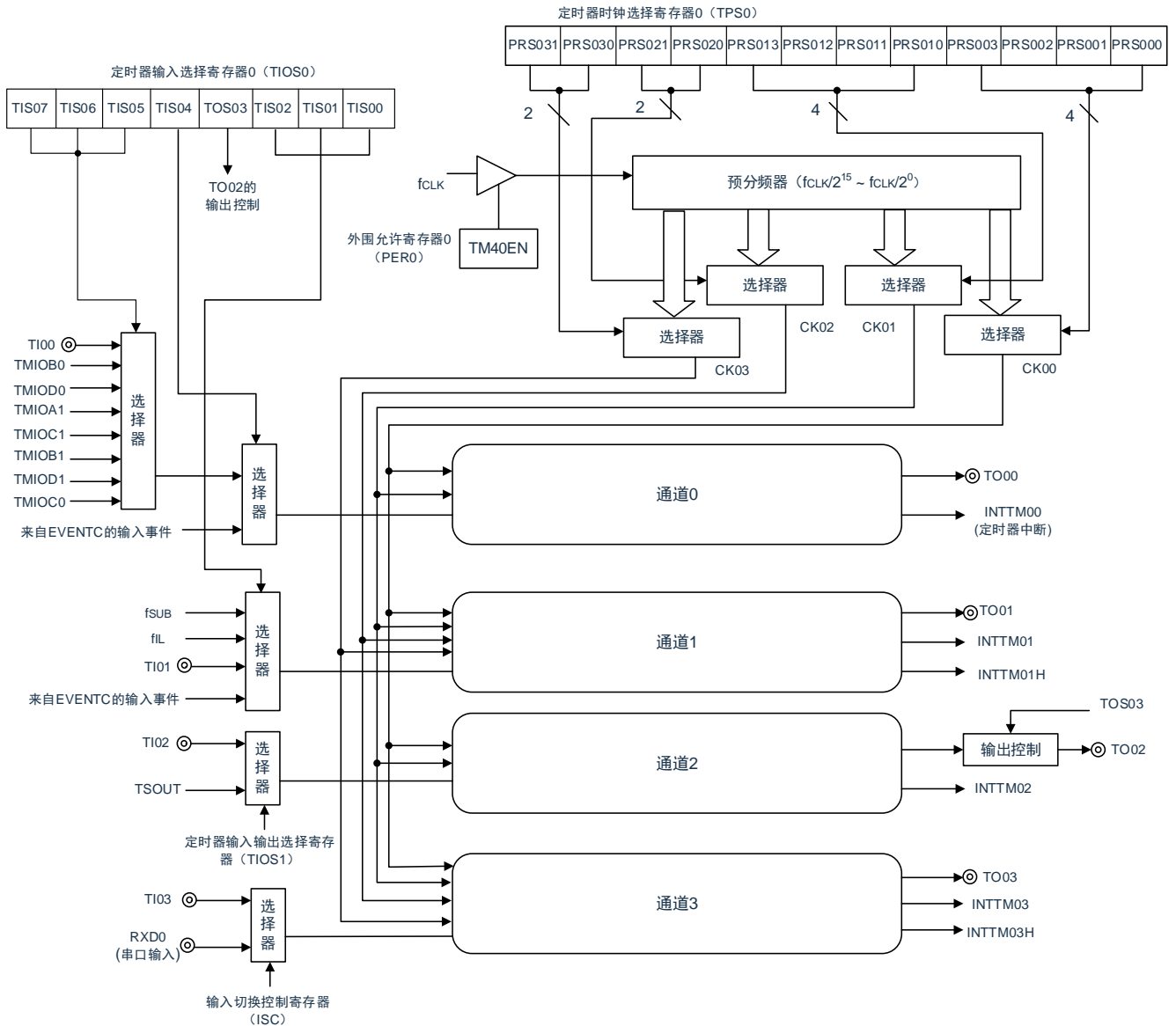
定时器阵列单元的通道		各产品的输入/输出引脚的有无	
		32,36,40,48,52,64引脚	
单元0	通道0	TI00/TO00	
	通道1	TI01/TO01	
	通道2	TI02/TO02	
	通道3	TI03/TO03	

备注:

1. 在定时器的输入和定时器的输出被同一个引脚复用时，只能用作定时器的输入或者定时器的输出。
2. -: 没有内置通道。

通用定时器单元的框图如图6-1所示。

图6-1：通用定时器单元0的整体框图



备注：F_{SUB}：副系统时钟频率

F_{IL}：低速内部振荡器时钟频率

6.2.1 通用定时器单元寄存器列表

寄存器基址：0x40041C00

偏移地址	寄存器名	读写属性	位宽	复位值
0x180	TCR00	R	16	FFFFH
0x182	TCR01	R	16	FFFFH
0x184	TCR02	R	16	FFFFH
0x186	TCR03	R	16	FFFFH
0x190	TMR00	R/W	16	0000H
0x192	TMR01	R/W	16	0000H
0x194	TMR02	R/W	16	0000H
0x196	TMR03	R/W	16	0000H
0x1A0	TSR00	R	16	0000H
0x1A0	TSR00L	R	8	00H
0x1A2	TSR01	R	16	0000H
0x1A2	TSR01L	R	8	00H
0x1A4	TSR02	R	16	0000H
0x1A4	TSR02L	R	8	00H
0x1A6	TSR03	R	16	0000H
0x1A6	TSR03L	R	8	00H
0x1B0	TE0	R	16	0000H
0x1B0	TE0L	R	8	00H
0x1B2	TS0	R/W	16	0000H
0x1B2	TS0L	R/W	8	00H
0x1B4	TT0	R/W	16	0000H
0x1B4	TT0L	R/W	8	00H
0x1B6	TPS0	R/W	16	0000H
0x1B8	TO0	R/W	16	0000H
0x1B8	TO0L	R/W	8	00H
0x1BA	TOE0	R/W	16	0000H
0x1BA	TOE0L	R/W	8	00H
0x1BC	TOL0	R/W	16	0000H
0x1BC	TOL0L	R/W	8	00H
0x1BE	TOM0	R/W	16	0000H
0x1BE	TOM0L	R/W	8	00H
0x318	TDR00	R/W	16	0000H
0x31A	TDR01	R/W	16	0000H
0x31A	TDR01L	R/W	8	00H
0x31B	TDR01H	R/W	8	00H
0x364	TDR02	R/W	16	0000H
0x366	TDR03	R/W	16	0000H
0x366	TDR03L	R/W	8	00H
0x367	TDR03H	R/W	8	00H

6.2.2 定时器计数寄存器mn (TCRmn)

TCRmn 寄存器是对计数时钟进行计数的16位只读寄存器。与计数时钟的上升沿同步进行递增或者递减计数。

通过定时器模式寄存器mn(TMRmn)的MDmn3~MDmn0位来选择运行模式，进行递增和递减计数的切换(参照“6.3.3 定时器模式寄存器mn(TMRmn)”)。

图6-2: 定时器计数寄存器mn(TCRmn)的表格

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

注： m: 单元号(m=0) n: 通道号(n=0~3)

能通过读定时器计数寄存器 mn(TCRmn)来读计数值。

- 在以下情况下，计数值变为“FFFFH”。
- 当产生复位信号时
- 当清除外围允许寄存器0(PER0)的TM4mEN位时
- 在PWM输出模式中从属通道的计数结束时
- 在延迟计数模式中从属通道的计数结束时
- 在单触发脉冲输出模式中主控/从属通道的计数结束时
- 在多重PWM输出模式中从属通道的计数结束时

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

注意：即使读TCRmn寄存器，也不将计数值捕捉到定时器数据寄存器 mn(TDRmn)。

如下所示，TCRmn寄存器的读取值因运行模式和运行状态而不同。

表6-4：各运行模式中的定时器计数寄存器mn(TCRmn)的读取值

运行模式	计数方式	定时器计数寄存器mn(TCRmn)的读取值 ^注			
		解除复位后更改运行模式时的值	计数暂停(TTmn=1)时的值	计数暂停(TTmn=1)后更改运行模式时的值	单次计数后等待开始触发时的值
间隔定时器模式	递减计数	FFFFH	停止时的值	不定值	—
捕捉模式	递增计数	0000H	停止时的值	不定值	—
事件计数器模式	递减计数	FFFFH	停止时的值	不定值	—
单次计数模式	递减计数	FFFFH	停止时的值	不定值	FFFFH
捕捉& 单次计数模式	递增计数	0000H	停止时的值	不定值	TDRmn寄存器的捕捉值+1

注：表示通道n处于定时器运行停止状态(TEmn=0)和计数允许状态(TSmn=1)时的TCRmn寄存器的读取值。将此值保持在TCRmn寄存器，直到开始计数为止。

备注：m：单元号(m=0) n：通道号(n=0~3)

6.2.3 定时器数据寄存器mn(TDRmn)

这是能进行捕捉功能和比较功能切换使用的16位寄存器。通过定时器模式寄存器mn(TMRmn)的MDmn3~MDmn0位来选择运行模式，进行捕捉功能和比较功能的切换。

能随时改写TDRmn寄存器。

能以16位为单位读写此寄存器。

在8位定时器模式中(定时器模式寄存器m1、m3(TMRm1、TMRm3)的SPLIT位为“1”)，能以8位为单位读写TDRm1寄存器和TDRm3寄存器，其中TDRm1H和TDRm3H用作高8位，TDRm1L和TDRm3L用作低8位。

在产生复位信号后，TDRmn寄存器的值变为“0000H”。

图6-3: 定时器数据寄存器mn(TDRmn)(n=0、2、4、5、6、7)的表格



图6-4: 定时器数据寄存器mn(TDRmn)(n=1、3)的表格
(TDR01H可支持8bit操作) (TDR01L可支持8bit操作)



(i) 定时器数据寄存器mn(TDRmn)用作比较寄存器的情况

从TDRmn寄存器的设置值开始递减计数，当计数值变为“0000H”时，产生中断信号(INTTMmn)。保持TDRmn寄存器的值，直到被改写为止。

注意：即使输入捕捉触发信号，设置为比较功能的TDRmn寄存器也不进行捕捉运行。

(ii) 定时器数据寄存器mn(TDRmn)用作捕捉寄存器的情况

通过输入捕捉触发，将定时器计数寄存器 mn(TCRmn)的计数值捕捉到TDRmn寄存器。

能选择TIMn引脚的有效边沿作为捕捉触发信号。通过定时器模式寄存器mn(TMRmn)来设置捕捉触发的选择。

备注：m：单元号(m=0) n：通道号(n=0~3)

6.3 控制通用定时器单元的寄存器

控制通用定时器单元的寄存器如下所示：

- 外围允许寄存器0(PER0)
- 定时器时钟选择寄存器m(TPSm)
- 定时器模式寄存器mn(TMRmn)
- 定时器状态寄存器mn(TSRmn)
- 定时器通道允许状态寄存器m(TEm)
- 定时器通道开始寄存器m(TSm)
- 定时器通道停止寄存器m(TTm)
- 定时器输入输出选择寄存器(TIOS0,TIOS1)
- 定时器输出允许寄存器m(TOEm)
- 定时器输出寄存器m(TOm)
- 定时器输出电平寄存器m(TOLm)
- 定时器输出模式寄存器m(TOMm)
- 输入切换控制寄存器(ISC)
- 噪声滤波器允许寄存器1(NFEN1)
- 端口模式控制寄存器(PMCxx)
- 端口模式寄存器(PMxx)
- 端口寄存器(Pxx)

注意：分配的寄存器和位因产品而不同。必须给未分配的位设置初始值。

备注：m：单元号(m= 0) n：通道号(n=0~3)

6.3.1 外围允许寄存器0 (PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用通用定时器单元0时，必须将 bit0 (TM40EN)置“1”。通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，PER0寄存器的值变为“00H”。

图6-5: 外围允许寄存器0(PER0)的表格

地址: 0x40020420 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN ^注	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN

TM4EN	通用定时器单元0的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> 不能写通用定时器单元0使用的SFR。 通用定时器单元0处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> 能读写通用定时器单元0使用的SFR。

注意: 要设置通用定时器单元时, 必须先在 TM4mEN 位为“1”的状态下设置以下的寄存器。当 TM4mEN 位为“0”时, 定时器阵列单元的控制寄存器的值为初始值, 忽视写操作(定时器输入输出选择寄存器0 (TIOS0)、输入切换控制寄存器(ISC)、噪声滤波器允许寄存器1 (NFEN1)、端口模式控制寄存器 PMCx、端口模式寄存器 PMx和端口寄存器Px除外)。

- 定时器状态寄存器 mn(TSRmn)
- 定时器通道允许状态寄存器m(TEm)
- 定时器通道开始寄存器m(TSm)
- 定时器通道停止寄存器m(TTm)
- 定时器输出允许寄存器m(TOEm)
- 定时器输出寄存器m(TOm)
- 定时器输出电平寄存器m(TOLm)
- 定时器输出模式寄存器m(TOMm)

6.3.2 定时器时钟选择寄存器m(TPSm)

TPSm寄存器是16位寄存器，选择提供给各通道的2种或者4种公共运行时钟(CKm0、CKm1、CKm2、CKm3)。通过TPSm寄存器的bit3 ~ 0选择CKm0，通过TPSm寄存器的bit7 ~ 4选择CKm1。另外，只有通道1和通道3才能选择CKm2和CKm3，通过TPSm寄存器的bit9 ~ 8选择CKm2，通过TPSm寄存器的bit13和bit12选择CKm3。

只有在以下情况下才能改写定时器运行中的TPSm寄存器。

能改写PRSm00 ~ PRSm03位的情况 (n=0 ~ 3):

选择CKm0作为运行时钟(CKSmn1、CKSmn0=0、0)的通道全部处于停止状态(TEmn=0)。能改写PRSm10 ~ PRSm13位的情况 (n=0 ~ 3):

选择CKm2作为运行时钟(CKSmn1、CKSmn0=0、1)的通道全部处于停止状态(TEmn=0)。能改写PRSm20位和PRSm21位的情况(n=1、3):

选择CKm1作为运行时钟(CKSmn1、CKSmn0=1、0)的通道全部处于停止状态(TEmn=0)。能改写PRSm30位和PRSm31位的情况(n=1、3):

选择CKm3作为运行时钟(CKSmn1、CKSmn0=1、1)的通道全部处于停止状态(TEmn=0)。

通过16位存储器操作指令设置TPSm寄存器。在产生复位信号后，TPSm寄存器的值变为“0000H”。

图6-6: 定时器时钟选择寄存器m(TPSm)的表格 (1/2)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		运行时钟(CKmk)的选择 [‡] (k=0、1)				
					F _{CLK} =2MHz	F _{CLK} =4MHz	F _{CLK} =8MHz	F _{CLK} =20MHz	F _{CLK} =32MHz
0	0	0	0	F _{CLK}	2MHz	4MHz	8MHz	20MHz	32MHz
0	0	0	1	F _{CLK} /2	1MHz	2MHz	4MHz	10MHz	16MHz
0	0	1	0	F _{CLK} /2 ²	500KHz	1MHz	2MHz	5MHz	8MHz
0	0	1	1	F _{CLK} /2 ³	250KHz	500KHz	1MHz	2.5MHz	4MHz
0	1	0	0	F _{CLK} /2 ⁴	125KHz	250KHz	500KHz	1.25MHz	2MHz
0	1	0	1	F _{CLK} /2 ⁵	62.5KHz	125KHz	250KHz	625KHz	1MHz
0	1	1	0	F _{CLK} /2 ⁶	31.3KHz	62.5KHz	125KHz	313KHz	500KHz
0	1	1	1	F _{CLK} /2 ⁷	15.6KHz	31.3KHz	62.5KHz	156KHz	250KHz
1	0	0	0	F _{CLK} /2 ⁸	7.81KHz	15.6KHz	31.3KHz	78.1KHz	125KHz
1	0	0	1	F _{CLK} /2 ⁹	3.91KHz	7.81KHz	15.6KHz	39.1KHz	62.5KHz
1	0	1	0	F _{CLK} /2 ¹⁰	1.95KHz	3.91KHz	7.81KHz	19.5KHz	31.25KHz
1	0	1	1	F _{CLK} /2 ¹¹	977Hz	1.95KHz	3.91KHz	9.77KHz	15.6KHz
1	1	0	0	F _{CLK} /2 ¹²	488Hz	977Hz	1.95KHz	4.88KHz	7.81KHz
1	1	0	1	F _{CLK} /2 ¹³	244Hz	488Hz	977Hz	2.44KHz	3.91KHz
1	1	1	0	F _{CLK} /2 ¹⁴	122Hz	244Hz	488Hz	1.22KHz	1.95KHz
1	1	1	1	F _{CLK} /2 ¹⁵	61.0Hz	122Hz	244Hz	610Hz	977Hz

注：在更改选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)的情况下，必须停止通用定时器单元(TTm=000FH)。即使在选择运行时钟(F_{MCK})或者TIMn引脚输入信号的有效边沿时，也需要停止通用定时器单元。

注意：

1. 必须将bit15、14、11、10置“0”。
2. 如果选择F_{CLK}(无分频)作为运行时钟(CKmk)并且将TDRnm置“0000H”(n=0、1, m=0~3)，就不能使用通用定时器单元的中断请求。

备注：

1. F_{CLK}：CPU/外围硬件的时钟频率
2. TPSm寄存器选择的时钟波形从上升沿开始只有1个F_{CLK}周期为高电平(m=1~15)。详细内容请参照“6.5.1计数时钟(F_{TCLK})”。

图6-7: 定时器时钟选择寄存器m(TPSm)的表格(2/2)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRSm21	PRSm20	运行时钟(CKm2)的选择 ^注					
		F _{CLK} =2MHz	F _{CLK} =4MHz	F _{CLK} =8MHz	F _{CLK} =20MHz	F _{CLK} =32MHz	
0	0	F _{CLK} /2	1MHz	2MHz	4MHz	10MHz	16MHz
0	1	F _{CLK} /2 ²	500KHz	1MHz	2MHz	5MHz	8MHz
1	0	F _{CLK} /2 ⁴	125KHz	250KHz	500KHz	1.25MHz	2MHz
1	1	F _{CLK} /2 ⁶	31.3KHz	62.5KHz	125KHz	313KHz	500KHz

PRSm31	PRSm30	运行时钟(CKm3)的选择 ^注					
		F _{CLK} =2MHz	F _{CLK} =4MHz	F _{CLK} =8MHz	F _{CLK} =20MHz	F _{CLK} =32MHz	
0	0	F _{CLK} /2 ⁸	7.81KHz	15.6KHz	31.3KHz	78.1KHz	125KHz
0	1	F _{CLK} /2 ¹⁰	1.95KHz	3.91KHz	7.81KHz	19.5KHz	31.3KHz
1	0	F _{CLK} /2 ¹²	488Hz	977Hz	1.95KHz	4.88KHz	7.81KHz
1	1	F _{CLK} /2 ¹⁴	122Hz	244Hz	488Hz	1.22KHz	1.95KHz

注：在更改选择为FCLK的时钟(更改系统时钟控制寄存器(CKC)的值)的情况下，必须停止通用定时器单元(TTm=000FH)。即使在选择运行时钟(fMCK)或者TImn引脚输入信号的有效边沿时，也需要停止通用定时器单元。

注意：必须将bit15、14、11、10置“0”。

如果在8位定时器模式中使用通道1和通道3并且将CKm2和CKm3作为运行时钟，就能通过间隔定时器功能实现表6-5所示的间隔时间。

表6-5: 运行时钟CKSm2和CKSm3能设置的间隔时间

时钟		间隔时间 ^注 (F _{CLK} =32MHz)			
		10us	100us	1ms	10ms
CKm2	F _{CLK} /2	○	—	—	—
	F _{CLK} /2 ²	○	—	—	—
	F _{CLK} /2 ⁴	○	○	—	—
	F _{CLK} /2 ⁶	○	○	—	—
CKm3	F _{CLK} /2 ⁸	—	○	○	—
	F _{CLK} /2 ¹⁰	—	○	○	—
	F _{CLK} /2 ¹²	—	—	○	○
	F _{CLK} /2 ¹⁴	—	—	○	○

注：○ 包含5%以内的误差。

备注：

1. F_{CLK}: CPU/外围硬件的时钟频率
2. 有关TPSm寄存器所选F_{CLK}/2r波形的详细内容，请参照“6.5.1计数时钟(F_{TCLK})”。

6.3.3 定时器模式寄存器mn(TMRmn)

TMRmn寄存器是设置通道n运行模式的寄存器，进行运行时钟(F_{MCK})的选择、计数时钟的选择、主控/从属的选择、16位/8位定时器的选择(只限于通道1和通道3)、开始触发和捕捉触发的设置、定时器输入有效边沿的选择以及运行模式(间隔、捕捉、事件计数器、单次计数、捕捉 & 单次计数)的设置。

禁止在运行中($TE_{mn}=1$)改写TMRmn寄存器。但是，能在一部分的功能运行中($TE_{mn}=1$)改写bit7和bit6(CIS_{mn1} 、 CIS_{mn0})(详细内容请参照“6.8通用定时器单元的独立通道运行功能”和“6.9定时器阵列单元的多通道联动运行功能”)。

通过16位存储器操作指令设置TMRmn寄存器。在产生复位信号后，TMRmn寄存器的值变为“0000H”。

注意：TMRmn寄存器的bit11因通道而不同。

TMRm2: MASTERmn位($n=2$)

TMRm1、TMRm3: SPLITmn位($n=1、3$)

TMRm0: 固定为“0”。

图6-8: 定时器模式寄存器mn(TMRmn)的表格(1/4)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn ($n=2$)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
TMRmn ($n=1,3$)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
TMRmn ($n=0$)	CKS mn1	CKS mn0	0	CCS mn	0注1	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKSmn1	CKSmn0	通道n运行时钟(F_{MCK})的选择
0	0	定时器时钟选择寄存器m(TPSm)设置的运行时钟CKm0
0	1	定时器时钟选择寄存器m(TPSm)设置的运行时钟CKm2
1	0	定时器时钟选择寄存器m(TPSm)设置的运行时钟CKm1
1	1	定时器时钟选择寄存器m(TPSm)设置的运行时钟CKm3

运行时钟(F_{MCK})用于边沿检测电路。通过设置CCSmn位来产生采样时钟和计数时钟(F_{TCLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。

CCSmn	通道n计数时钟(F_{TCLK})的选择
0	CKSmn0位和CKSmn1位指定的运行时钟(F_{MCK})
1	TImn引脚输入信号的有效边沿 • 单元0的情况: 通道0: TIS0选择的输入信号的有效边沿 通道1: TIS0选择的输入信号的有效边沿 通道3: ISC选择的输入信号的有效边沿

计数时钟(F_{TCLK})用于计数器、输出控制电路和中断控制电路。

注1: bit11是只读位, 固定为“0”, 忽视写操作。

注意:

1. 必须将 bit13、5、4置“0”。
2. 要更改选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)时, 即使选择了CKSmn0位和CKSmn1位指定的运行时钟(F_{MCK})或者TImn引脚输入信号的有效边沿作为计数时钟(F_{TCLK}), 也必须停止定时器阵列单元(TTm=00FFH)。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-9: 定时器模式寄存器mn(TMRmn)的表格 (2/4)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{注1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(TMRmn(n=2)的bit11)

MASTERmn	通道n的独立通道运行/多通道联动运行(从属或者主控)的选择
0	用作独立通道运行功能或者多通道联动运行功能的从属通道。
1	用作多通道联动运行功能的主控通道。

只能将通道2设置为主控通道(MASTERmn=1)。
通道0固定为“0”(因为通道0为最高位的通道, 所以与此位的设置无关, 用作主控通道)。对于用作独立通道运行功能的通道, 将MASTERmn位置“0”。

(TMRmn(n=1、3)的bit11)

SPLITmn	通道1和通道3的8位定时器/16位定时器的运行选择
0	用作16位定时器。 (用作独立通道运行功能或者多通道联动运行功能的从属通道)
1	用作8位定时器。

STSmn2	STSmn1	STSmn0	通道n的开始触发和捕捉触发的设置
0	0	0	只有软件触发开始有效(不选择其他触发源)。
0	0	1	将TImn引脚输入的有效边沿用于开始触发和捕捉触发。
0	1	0	将TImn引脚输入的双边沿分别用于开始触发和捕捉触发。
1	0	0	使用主控通道的中断信号(多通道联动运行功能的从属通道的情况)。
上述以外			禁止设置。

注1: bit11是只读位, 固定为“0”, 忽视写操作。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-10: 定时器模式寄存器mn(TMRmn)的表格(3/4)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{注1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CISmn1	CISmn0	TImn引脚的有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	双边沿(测量低电平宽度时) 开始触发: 下降沿, 捕捉 触发: 上升沿
1	1	双边沿(测量高电平宽度时) 开始触发: 上升沿, 捕捉 触发: 下降沿

当STSmn2~STSmn0位不为“010B”并且使用双边沿指定时, 必须将CISmn1~CISmn0位置“10B”。

注1: bit11是只读位, 固定为“0”, 忽视写操作。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-11: 定时器模式寄存器mn(TMRmn)的表格 (4/4)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=2)	CKS mn1	CKS mn0	0	CCS mn	MAS TERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=1,3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n=0)	CKS mn1	CKS mn0	0	CCS mn	0 ^{注1}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MD mn3	MD mn2	MD mn1	通道n运行模式的设置	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出(主控)	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出 (从属)	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设置。		
各模式的运行因MDmn0位而变(参照下表)。					

运行模式(MDmn3~MDmn1位的设置 (参照上表))	MD mn0	开始计数和中断的设置
<ul style="list-style-type: none"> 间隔定时器模式(0、0、0)捕捉 模式(0、1、0) 	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断(定时器的输出也发生变化)。
<ul style="list-style-type: none"> 事件计数器模式(0、1、1) 	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。
<ul style="list-style-type: none"> 单次计数模式^{注2}(1、0、0) 	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效 ^{注3} 。此时不产生中断。
<ul style="list-style-type: none"> 捕捉&单次计数模式(1、1、0) 	0	在开始计数时不产生定时器中断(定时器的输出也不发生变化)。计数运行中的开始触发无效。此时不产生中断。

注1: bit11是只读位, 固定为“0”, 忽视写操作。

注2: 在单次计数模式中, 不控制开始计数时的中断输出(INTTMmn)和TOMn输出。

注3: 如果在运行中产生开始触发(TSmn=1), 就对计数器进行初始化并且重新开始计数(不产生中断请求)

备注: m: 单元号(m=0) n: 通道号(n=0~3)

6.3.4 定时器状态寄存器mn(TSRmn)

TSRmn寄存器是表示通道n计数器的上溢状态的寄存器。

TSRmn寄存器只在捕捉模式(MDmn3~MDmn1=010B)和捕捉&单次计数模式(MDmn3~MDmn1=110B)中有效。有关各运行模式中的OVF位的变化和置位/清除条件，请参照表6-6。

通过16位存储器操作指令读TSRmn寄存器。

能用TSRmnL并且通过8位存储器操作指令读TSRmn寄存器的低8位。在产生复位信号后，TSRmn寄存器的值变为“0000H”。

图6-12: 定时器状态寄存器mn(TSRmn)的表格

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

OVF	通道n的计数器上溢状态
0	没有发生上溢。
1	发生上溢。
如果OVF位为“1”，就在下一次计数不发生上溢并且捕捉到计数值时清除此标志(OVF=0)。	

备注：m：单元号(m=0, 1) n：通道号(m=0时：n=0~3, m=1时：n=0~7)

表6-6: 各运行模式中的OVF位的变化和置位/清除条件

定时器运行模式	OVF位	置位/清除条件
•捕捉模式	清除	在捕捉时没有发生上溢的情况
•捕捉&单次计数模式	置位	在捕捉时发生上溢的情况
•间隔定时器模式 •事件计数器模式 •单次计数模式	清除 置位	— (不能使用)

备注：即使计数器发生上溢，OVF位也不立即发生变化，而在此后的捕捉时发生变化。

6.3.5 定时器通道允许状态寄存器m(TEM)

TEM寄存器是表示各通道定时器运行的允许或者停止状态的寄存器。

TEM寄存器的各位对应定时器通道开始寄存器m(TSm)和定时器通道停止寄存器m(TTm)的各位。如果将TSm寄存器的各位置“1”，就将TEM寄存器的对应位置“1”。如果将TTm寄存器的各位置“1”，就将其对应位清“0”。

通过16位存储器操作指令读TEM寄存器。

能用TEML并且通过8位存储器操作指令读TEM寄存器的低8位。在产生复位信号后，TEM寄存器的值变为“0000H”。

图6-13：定时器通道允许状态寄存器m(TEM)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEM	0	0	0	0	TEH _{m3}	0	TEH _{m1}	0	0	0	0	0	TEM ₃	TEM ₂	TEM ₁	TEM ₀

备注：m=0

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

备注：m=1

TEH _{m3}	通道3为8位定时器模式时的高8位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

TEH _{m1}	通道1为8位定时器模式时的高8位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

TEM _n	通道n的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

在通道1和通道3为8位定时器模式时，TEM₁和TEM₃表示低8位定时器的运行允许或者停止状态。

备注：m：单元号(m=0) n：通道号(n=0~3)

6.3.6 定时器通道开始寄存器 m (TSm)

TSm寄存器是对定时器计数寄存器mn(TCRmn)进行初始化并且设置各通道计数运行开始的触发寄存器。如果将各位置“1”，定时器通道允许状态寄存器m(TEm)的对应位就被置“1”。因为TSmn位、TSHm1位和TSHm3位是触发位，所以如果变为运行允许状态(TEmn、TEHm1、TEHm3=1)，就立即清除TSmn位、TSHm1位和TSHm3位。

通过16位存储器操作指令设置TSm寄存器。

能用TSmL并且通过8位存储器操作指令设置TSm寄存器的低8位。在产生复位信号后，TSm寄存器的值变为“0000H”。

图6-14: 定时器通道开始寄存器 m (TSm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSH m3	0	TSH m1	0	0	0	0	0	TSm 3	TSm 2	TSm 1	TSm 0

TSHm3	通道3 为8 位定时器模式时的高8 位定时器的运行允许(开始)触发
0	没有触发。
1	将TEHm3 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm3 寄存器的计数，就进入间隔定时器模式(参照“6.5.2 计数器的开始时序”的表6-6)。

TSHm1	通道1 为8 位定时器模式时的高8 位定时器的运行允许(开始)触发
0	没有触发。
1	将TEHm1 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm1 寄存器的计数，就进入间隔定时器模式(参照“6.5.2 计数器的开始时序”的表6-6)。

TSmn	通道n 的运行允许(开始)触发
0	没有触发。
1	将TEmn 位置“1”，进入计数允许状态。计数允许状态下的TCRmn寄存器的计数开始因各运行模式而不同(参照“6.5.2 计数器的开始时序”的表6-6)。在通道1 和通道3 为8 位定时器模式时，TSm1 和TSm3 为低8 位定时器的运行允许(开始)触发。

注意:

1. 必须将bit15 ~ 12、10、8 ~ 4置“0”。
2. 在从不使用TImn引脚输入的功能切换到使用TImn引脚输入的功能时，从设置定时器模式寄存器mn(TMRmn)到将TSmn (TSHm1、TSHm3)位置“1”为止，需要以下期间的等待：
TImn引脚噪声滤波器有效时(TNFENmn=1)：4个运行时钟(F_{MCK})
TImn引脚噪声滤波器无效时(TNFENmn=0)：2个运行时钟(F_{MCK})

备注:

1. TSm 寄存器的读取值总是“0”。
2. m: 单元号 (m= 0) n: 通道号 (n=0 ~ 3)

6.3.7 定时器通道停止寄存器m(TTm)

TTm寄存器是设置各通道计数停止的触发寄存器。

如果将各位置“1”，定时器通道允许状态寄存器m(TEm)的对应位就被清“0”。因为TTmn位、TTHm1位和TTHm3位是触发位，所以如果变为运行停止状态(TEmn、TEHm1、TEHm3=0)，就立即清除TTmn位、TTHm1位和TTHm3位。

通过16位存储器操作指令设置TTm寄存器。

能用TTmL并且通过8位存储器操作指令设置TTm寄存器的低8位。在产生复位信号后，TTm寄存器的值变为“0000H”。

图6-15：定时器通道停止寄存器m(TTm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTH _{m3}	0	TTH _{m1}	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	通道3为8位定时器模式时的高8位定时器的运行停止触发
0	没有触发。
1	将TEHm3位清“0”，进入计数停止状态。

TTHm1	通道1为8位定时器模式时的高8位定时器的运行停止触发
0	没有触发。
1	将TEHm1位清“0”，进入计数停止状态。

TTmn	通道n的运行停止触发
0	没有触发。
1	将TEmn位清“0”，进入计数停止状态。 在通道1和通道3为8位定时器模式时，TTm1和TTm3为低8位定时器的运行停止触发。

注意：必须将bit15~12、10、8~4置“0”。

备注：

1. TTm寄存器的读取值总是“0”。
2. m：单元号(m=0)n：通道号(n=0~3)

6.3.8 定时器输入输出选择寄存器(TIOS0,TIOS1)

TIOS0寄存器选择单元0的通道0和通道1的定时器输入以及通道2的定时器输出。通过8位存储器操作指令设置TIOS0寄存器。在产生复位信号后，TIOS0寄存器的值变为“00H”。

图6-16: 定时器输入选择寄存器0(TIOS0)的表格

地址: 0x40020474

复位后: 00H

R/W

符号

	7	6	5	4	3	2	1	0
TIOS0	TIS07	TIS06	TIS05	TIS04	TOS03	TIS02	TIS01	TIS00

TIS07	TIS06	TIS05	通道0使用的定时器输入的选择
0	0	0	定时器输入引脚(TI00)的输入信号
0	0	1	TimerM的输出信号TMIOB0(不经过PWMOP)
0	1	0	TimerM的输出信号TMIOD0(不经过PWMOP)
0	1	1	TimerM的输出信号TMIOA1(不经过PWMOP)
1	0	0	TimerM的输出信号TMIOC1(不经过PWMOP)
1	0	1	TimerM的输出信号TMIOB1(不经过PWMOP)
1	1	0	TimerM的输出信号TMIOD1(不经过PWMOP)
1	1	1	TimerM的输出信号TMIOC0(不经过PWMOP)

TIS04	通道0使用的定时器输入的选择
0	通过TIS07~TIS05选择的输入信号
1	ELC的事件输入信号

TOS03	通道2的定时器输出的使能
0	允许输出
1	禁止输出(输出固定为0)

TIS02	TIS01	TIS00	通道1使用的定时器输入的选择
0	0	0	定时器输入引脚(TI01)的输入信号
0	0	1	EVENTC的事件输入信号
0	1	0	定时器输入引脚(TI01)的输入信号
0	1	1	
1	0	0	低速内部振荡器时钟(F _{IL})
1	0	1	副系统时钟(F _{SUB})
上述以外			禁止设置。

注意:

1. 选择的定时器输入的高低电平宽度需要大于等于 $1/F_{MCK}+10ns$ 。因此，在选择 F_{SUB} 作为 F_{CLK} 时(CKC寄存器的CSS=1)，不能将TIS02位置“1”。
2. 在通过定时器输入选择寄存器0(TIOS0)选择ELC的事件输入信号时，必须通过定时器时钟选择寄存器0(TPS0)选择 F_{CLK} 。

TIOS1寄存器选择单元0的通道2的定时器输入。通过8位存储器操作指令设置TIOS1寄存器。在产生复位信号后，TIOS1寄存器的值变为“00H”。

图6-17: 定时器输入选择寄存器 1 (TIOS1)的表格

地址: 0x40020475	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
TIOS1	0	0	0	0	0	0	0	TIS10

TIS10	通道2 使用的定时器输入的选择
0	定时器输入引脚(TI02)的输入信号
1	CAN Controller的TSOUT输出

6.3.9 定时器输出允许寄存器m(TOEm)

TOEm寄存器是设置允许或者禁止各通道定时器输出的寄存器。

对于允许定时器输出的通道n，无法通过软件改写后述的定时器输出寄存器m(TOm)的TOmn位的值，并且由计数运行的定时器输出功能反映的值从定时器的输出引脚(TOmn)输出。

通过16位存储器操作指令设置TOEm寄存器。

能用TOEmL并且通过8位存储器操作指令设置TOEm寄存器的低8位。在产生复位信号后，TOEm寄存器的值变为“0000H”。

图6-18：定时器输出允许寄存器m(TOEm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	0	0	0	0	TOE m3	TOE m2	TOE m1	TOE m0

TOEmn	通道n的定时器输出的允许/禁止
0	禁止定时器输出。 定时器的运行不反映到TOmn位，固定输出。 能写TOmn位，并且从TOmn引脚输出TOmn位设置的电平。
1	允许定时器输出。 定时器的运行反映到TOmn位，产生输出波形。忽视TOmn位的写操作。

注意：必须将bit15~4置“0”。

备注：m：单元号(m=0)n：通道号(n=0~3)

6.3.10 定时器输出寄存器m(TOm)

TOm寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚(TOm_n)输出。

只有在禁止定时器输出 (TOEm_n=0)时才能通过软件改写此寄存器的TOm_n位。当允许定时器输出时 (TOEm_n=1)，忽视通过软件的改写操作，而只通过定时器的运行更改其值。

要将TI00、TO00、TI01/TO01、TI02/TO02、TI03/TO03 引脚用作端口功能时，必须将相应的TOm_n位置“0”。

通过16位存储器操作指令设置TOm寄存器。

能用TOmL并且通过8位存储器操作指令设置TOm寄存器的低8位。在产生复位信号后，TOm寄存器的值变为“0000H”。

图6-19: 定时器输出寄存器m(TOm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	0	0	0	0	TOm3	TOm2	TOm1	TOm0

TOm _n	通道n的定时器输出
0	定时器的输出值为“0”。
1	定时器的输出值为“1”。

注意：必须将bit15~4置“0”。

备注：m：单元号(m=0)n：通道号(n=0~3)

6.3.11 定时器输出电平寄存器m(TOLm)

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

当允许定时器输出 (TOEmn=1)并且使用多通道联动运行功能 (TOMmn=1)时, 在定时器输出信号的置位和复位时序, 反映此寄存器进行的各通道 n 的反相设置。在主导通道输出模式 (TOMmn=0)中, 此寄存器的设置无效。

通过 16 位存储器操作指令设置 TOLm 寄存器。

能用 TOLmL 并且通过 8 位存储器操作指令设置 TOLm 寄存器的低 8 位。在产生复位信号后, TOLm 寄存器的值变为“0000H”。

图6-20: 定时器输出电平寄存器m(TOLm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOLm3	TOLm2	TOLm1	0

TOLmn	通道n的定时器输出电平的控制
0	正逻辑输出(高电平有效)
1	反相输出(低电平有效)

注意: 必须将bit15~4和bit0置“0”。

备注:

1. 如果在定时器运行中改写此寄存器的值, 就在下一次定时器输出信号发生变化时反相定时器的输出逻辑, 而不是在改写后立即反相。
2. m: 单元号(m=0)n: 通道号(n=0~3)

6.3.12 定时器输出模式寄存器m(TOMm)

TOMm寄存器是控制各通道定时器输出模式的寄存器。当用作独立通道运行功能时，将所用通道的对应位置“0”。

当用作多通道联动运行功能(PWM 输出、单触发脉冲输出和多重 PWM 输出)时，将主控通道的对应位置“0”并且将从属通道的对应位置“1”。

当允许定时器输出 (TOEmn=1)时，在定时器输出信号的置位和复位时序，反映此寄存器进行的各通道 n 的设置。

通过 16 位存储器操作指令设置 TOMm 寄存器。

能用 TOMmL 并且通过 8 位存储器操作指令设置 TOMm 寄存器的低 8 位。在产生复位信号后，TOMm 寄存器的值变为“0000H”。

图6-21：定时器输出模式寄存器m(TOMm)的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOM _{m3}	TOM _{m2}	TOM _{m1}	0

TOMmn	通道n的定时器输出模式的控制
0	主控通道输出模式(通过定时器中断请求信号(INTTMmn)进行交替输出)
1	从属通道输出模式(通过主控通道的定时器中断请求信号(INTTMmn)将输出置位，并且通过从属通道的定时器中断请求信号(INTTMmp)对输出进行复位)

注意：必须将bit15~4和bit0置“0”。

备注：m：单元号(m=0)；n：通道号 n=0~3(主控通道时：n=0、2)

p：从属通道号

n=0：p=1、2、3

n=2：p=3

(有关主控通道和从属通道关系的详细内容，请参照“6.4.1多通道联动运行功能的基本规则”)

6.3.13 输入切换控制寄存器(ISC)

ISC寄存器的ISC1位和ISC0位用于通道3和通用串行通信单元的协调来实现LIN-bus通信。如果将ISC1位置“1”，就选择串行数据输入引脚(RxD0)的输入信号作为定时器的输入。

有关SSIE00位的设置，请参照“19.3.14输入切换控制寄存器(ISC)”。通过8位存储器操作指令设置ISC寄存器。

在产生复位信号后，ISC寄存器的值变为“00H”。

图6-22: 输入切换控制寄存器(ISC)的表格

地址: 0x40040473

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0

SSIE00	CSI00通信的从属模式中的通道0的SSI00引脚输入设置
0	SSI00引脚输入无效。
1	SSI00引脚输入有效。

ISC1	通用定时器单元0的通道3的输入切换
0	将TI03引脚的输入信号用作定时器的输入(通常运行)。
1	将RxD0引脚的输入信号用作定时器的输入(检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽)。

ISC0	外部中断(INTP0)的输入切换
0	将INTP0引脚的输入信号用作外部中断的输入(通常运行)。
1	将RxD0引脚的输入信号用作外部中断的输入(检测唤醒信号)。

注意: 必须将bit6~2置“0”。

备注: 要使用LIN-bus进行通信时, 必须将ISC1位置“1”, 选择RxD0引脚的输入信号。

6.3.14 噪声滤波器允许寄存器(NFEN1)

NFEN1 寄存器设置噪声滤波器是否用于各通道定时器输入引脚的输入信号。对于需要消除噪声的引脚，必须将对应位置“1”，使噪声滤波器有效。当噪声滤波器有效时，在通过对象通道的运行时钟 (F_{MCK})进行同步后检测 2 个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟(F_{MCK})进行同步^注。

通过 8 位存储器操作指令设置 NFEN1 寄存器。在产生复位信号后，NFEN1 寄存器的值变为“00H”。

注：详细内容请参照“6.5.1(2)选择TImn引脚输入信号的有效边沿的情况($CCSmn=1$)”、“6.5.2计数器的开始时序”和“6.7定时器输入(TImn)的控制”。

图6-23：噪声滤波器允许寄存器1(NFEN1)的表格

地址: 0x40040471	复位后: 00H				R/W			
符号	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	TI03引脚或者RxD0引脚的输入信号噪声滤波器的使用与否 ^注
0	噪声滤波器OFF
1	噪声滤波器ON

TNFEN02	TI02引脚的输入信号噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON

TNFEN01	TI01引脚的输入信号噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON

TNFEN00	TI00引脚的输入信号噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON

注：能通过设置输入切换控制寄存器(ISC)的ISC1位来切换适用的引脚。

ISC1=0：可选择是否使用TI03引脚的噪声滤波器。

ISC1=1：可选择是否使用RxD0引脚的噪声滤波器。

备注：通道0~3的定时器输入/输出引脚的有无因产品而不同。详细内容请参照“表6-3：各产品具有的定时器输入/输出引脚”。

6.3.15 控制定时器输入/输出引脚端口功能的寄存器

在使用通用定时器单元时，必须设置与对象通道复用的端口功能的控制寄存器(端口模式寄存器(PMxx)、端口寄存器(Pxx)和端口模式控制寄存器(PMCxx))。详细内容请参照“2.3.1端口模式寄存器(PMxx)”、“2.3.2端口寄存器(Pxx)”和“2.3.6端口模式控制寄存器(PMCxx)”。

设置的端口模式寄存器(PMxx)、端口寄存器(Pxx)和端口模式控制寄存器(PMCxx)因产品而不同。详细内容请参照“2.5使用复用功能时的寄存器设置”。

在将定时器输出引脚的复用端口用作定时器的输出时，必须将各端口对应的端口模式控制寄存器(PMCxx)的位、端口模式寄存器(PMxx)的位和端口寄存器(Pxx)的位置“0”。

(例)将P01/TO00用作定时器输出的情况

将端口模式控制寄存器0的PMC01位置“0”。

将端口模式寄存器0的PM01位置“0”。

将端口寄存器0的P01位置“0”。

在将定时器输入引脚的复用端口用作定时器的输入时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”并且将端口模式控制寄存器(PMCxx)的位置“0”。此时，端口寄存器(Pxx)的位可以是“0”或者“1”。

(例)将P00/TI00用作定时器输入的情况

将端口模式控制寄存器0的PMC00位置“0”。

将端口模式寄存器0的PM00位置“1”。

将端口寄存器0的P00位置“0”或者“1”。

6.4 通用定时器单元的基本规则

6.4.1 多通道联动运行功能的基本规则

多通道联动运行功能是将主控通道(主要对周期进行计数的基准定时器)和从属通道(遵从主控通道运行的定时器)组合实现的功能,使用时需要遵守几个规则。

多通道联动运行功能的基本规则如下所示。

- 1) 只能将偶数通道(通道0、通道2)设置为主控通道。
- 2) 通道0以外的任何通道都能设置为从属通道。
- 3) 只能将主控通道的低位通道设置为从属通道。

例在将通道0设置为主控通道时,能将通道1开始的通道(通道1、通道2、通道3)设置为从属通道。

- 4) 能对1个主控通道设置多个从属通道。
- 5) 当使用多个主控通道时,不能设置跨越主控通道的从属通道。

例在将通道0和通道2设置为主控通道时,能将通道1设置为主控通道0的从属通道,而不能将通道3设置为主控通道0的从属通道。

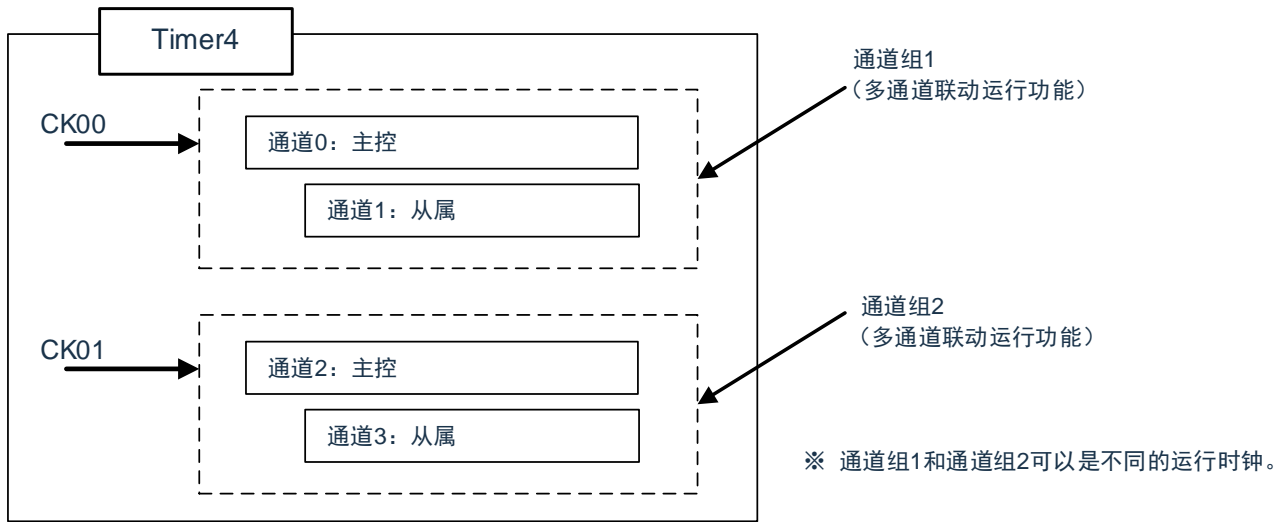
- 6) 和主控通道联动的从属通道需要设置相同的运行时钟。和主控通道联动的从属通道的CKSmn0位和CKSmn1位(定时器模式寄存器mn(TMRmn)的bit15和bit14)的值需要是相同的设置值。
- 7) 主控通道能将INTTMmn(中断)、开始软件触发和计数时钟传给低位通道。
- 8) 从属通道能将主控通道的INTTMmn(中断)、开始软件触发和计数时钟用作源时钟,但是不能将自己的INTTMmn(中断)、开始软件触发和计数时钟传给低位通道。
- 9) 主控通道不能将其他高位主控通道的INTTMmn(中断)、开始软件触发和计数时钟用作源时钟。
- 10) 为了同时启动要联动的通道,需要同时设置联动通道的通道开始触发位(TSmn)。
- 11) 只有联动的全部通道或者主控通道才能使用计数运行中的TSmn位的设置。不能只使用从属通道的TSmn位的设置。
- 12) 为了同时停止要联动的通道,需要同时设置联动通道的通道停止触发位(TTmn)。
- 13) 在联动运行时,因为主控通道和从属通道需要相同的运行时钟,所以不能选择CKm2/CKm3。
- 14) 定时器模式寄存器m0(TMRm0)没有主控位而固定为“0”。但是,因为通道0是最高位的通道,所以在联动运行时能将通道0用作主控通道。

多通道联动运行功能的基本规则是适用于通道组群(形成1个多通道联动运行功能的主控通道和从属通道的集合)的规则。

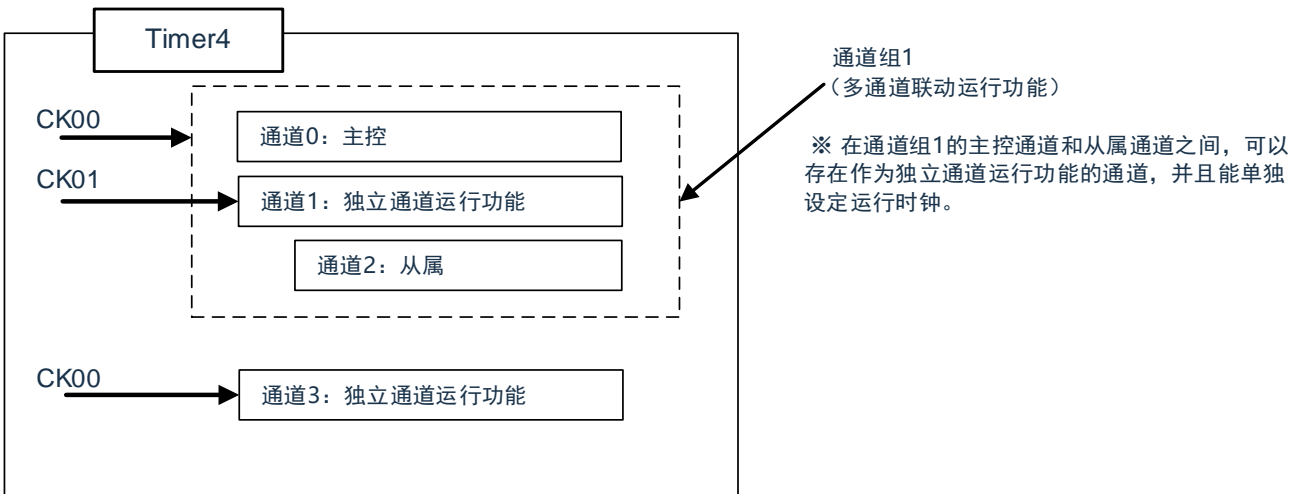
如果设置2个或者更多的相互不联动的通道群,通道群之间就不适用上述的基本规则。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

例1



例2



6.4.2 8位定时器运行功能的基本规则(只限于通道1和通道3)

8位定时器运行功能是将16位定时器的通道用作2个8位定时器的通道的功能。

只有通道1和通道3才能使用8位定时器运行功能，使用时需要遵守几个规则。

8位定时器运行功能的基本规则如下所示。

- 1) 8位定时器运行功能只适用于通道1和通道3。
- 2) 当用作8位定时器时，将定时器模式寄存器mn(TMRmn)的SPLIT位置“1”。
- 3) 高8位定时器能用作间隔定时器功能。
- 4) 在开始运行时，高8位定时器输出INTTm1H/INTTm3H(中断)(和MDmn0位为“1”的运行相同)。
- 5) 高8位定时器的运行时钟的选择取决于低位TMRmn寄存器的CKSmn1位和CKSmn0位的设置。
- 6) 对于高8位定时器，通过操作TSHm1/TSHm3位来开始通道的运行，并且通过操作TTHm1/TTHm3位来停止通道的运行。能通过TEHm1/TEHm3位确认通道的状态。
- 7) 低8位定时器的运行取决于TMRmn寄存器的设置，有以下3种支持低8位定时器运行的功能：
 - 间隔定时器功能
 - 外部事件计数器功能
 - 延迟计数功能
- 8) 对于低8位定时器，通过操作TSm1/TSm3位来开始通道的运行，并且通过操作TTm1/TTm3位来停止通道的运行。能通过TEm1/TEm3位确认通道的状态。
- 9) 在16位定时器运行时，TSHm1/TSHm3/TTHm1/TTHm3位的操作无效。通过操作TSm1/TSm3位和TTm1/TTm3位使通道1和通道3运行。TEHm3位和TEHm1位不变。
- 10) 8位定时器功能不能使用联动运行功能(单触发脉冲、PWM和多重PWM)。

备注：m：单元号(m=0)n：通道号(n=1、3)

6.5 计数器的运行

6.5.1 计数时钟(F_{TCLK})

通用定时器单元的计数时钟(F_{TCLK})能通过定时器模式寄存器mn(TMRmn)的CCSmn位选择以下任意一个时钟:

- CKSmn0位和CKSmn1位指定的运行时钟(F_{MCK})
- TImn引脚输入信号的有效边沿

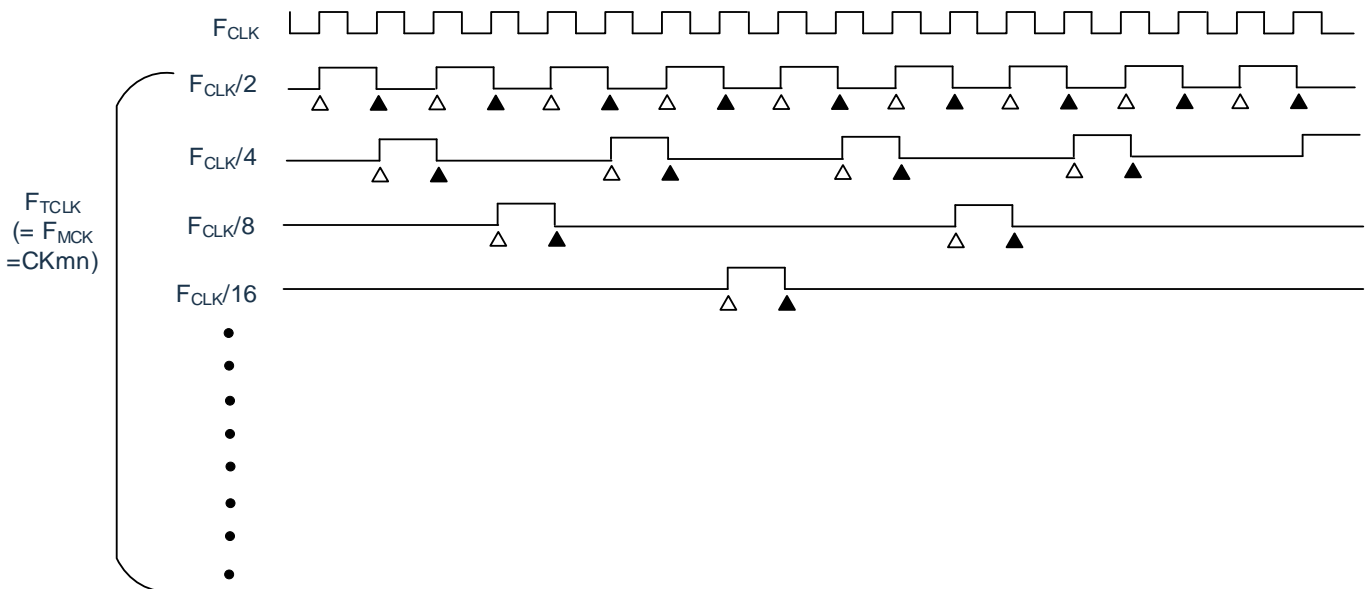
通用定时器单元被设计为与 F_{CLK} 同步运行, 因此计数时钟(F_{TCLK})的时序如下。

(1) 选择CKSmn0位和CKSmn1位指定的运行时钟(F_{MCK})的情况($CCSmn=0$)

根据定时器时钟选择寄存器m(TPSm)的设置, 计数时钟(F_{TCLK})为 $F_{CLK} \sim F_{CLK}/2^{15}$ 。但是, 当选择 F_{CLK} 的分频时, TPSm寄存器选择的时钟是从上升沿开始只有1个 F_{CLK} 周期为高电平的信号。当选择 F_{CLK} 时, 固定为高电平。

为了取得与 F_{CLK} 的同步, 定时器计数寄存器mn(TCRmn)从计数时钟的上升沿开始延迟1个 F_{CLK} 时钟后进行计数, 出于方便而将其称为“在计数时钟的上升沿进行计数”。

图6-24: F_{CLK} 和计数时钟(F_{TCLK})的时序($CCSmn=0$ 的情况)



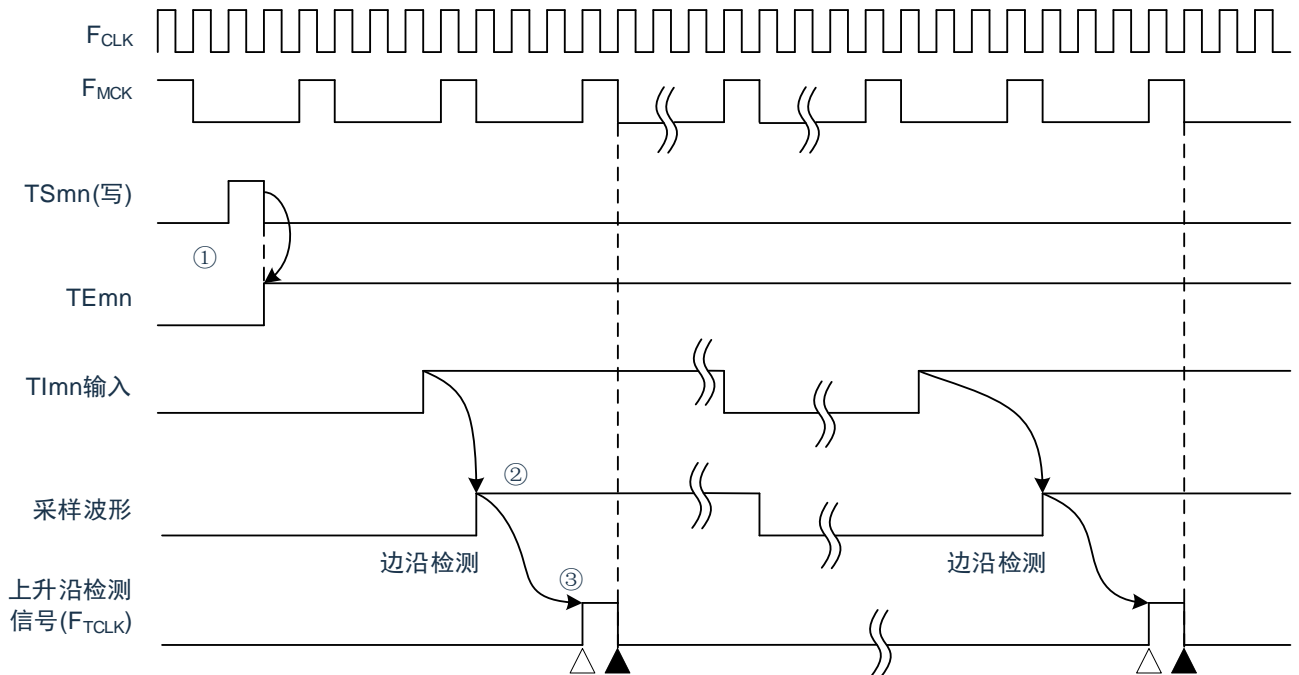
备注:

1. Δ : 计数时钟的上升沿
 \blacktriangle : 同步、计数器的递增/递减
2. F_{CLK} : CPU/外围硬件的时钟

(2) 选择TImn引脚输入信号的有效边沿的情况(CCSmn=1)

计数时钟(F_{TCLK})是检测TImn引脚输入信号的有效边沿并且与下一个 F_{MCK} 上升沿同步的信号。实际上，这是比TImn引脚的输入信号延迟了1~2个 F_{MCK} 时钟的信号(在使用噪声滤波器时，延迟3~4个 F_{MCK} 时钟)。为了取得与 F_{CLK} 的同步，定时器计数寄存器mn(TCRmn)从计数时钟的上升沿开始延迟1个 F_{CLK} 时钟后进行计数，出于方便而将其称为“在TImn引脚输入信号的有效边沿进行计数”。

图6-25: 计数时钟(F_{TCLK})的时序(CCSmn=1, 未使用噪声滤波器的情况)



- ① 通过将TSmn位置位来开始定时器的运行，并且等待TImn输入的有效边沿。
- ② 通过 F_{MCK} 对TImn输入的上升沿进行采样。
- ③ 在采样信号的上升沿检测边沿，并且输出检测信号(计数时钟)。

备注:

- 1. \triangle : 计数时钟的上升沿
 \blacktriangle : 同步、计数器的递增/递减
- 2. F_{CLK} : CPU/外围硬件的时钟
 F_{MCK} : 通道n的运行时钟
- 3. 输入脉冲间隔的测量、输入信号高低电平的测量、延迟计数器和单触发脉冲输出功能的TImn输入也是同样的波形。

6.5.2 计数器的开始时序

通过将定时器通道开始寄存器m(TSm)的TSMn位置位，定时器计数寄存器mn(TCRmn)进入运行允许状态。从计数允许状态到定时器计数寄存器mn(TCRmn)开始计数为止的运行如表6-7所示。

表6-7：从计数允许状态到定时器计数寄存器mn(TCRmn)开始计数为止的运行

定时器的运行模式	将TSMn位置“1”后的运行
• 间隔定时器模式	从检测到开始触发(TSMn=1)到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器，并且通过后续的计数时钟进行递减计数(参照“6.5.3(1)间隔定时器模式的运行”)。
• 事件计数器模式	通过给TSMn位写“1”，将TDRmn寄存器的值装入TCRmn寄存器。 如果检测到Timn的输入边沿，就通过后续的计数时钟进行递减计数。(参照“6.5.3(2)事件计数器模式的运行”)。
• 捕捉模式	从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器，并且通过后续的计数时钟进行递增计数(参照“6.5.3(3)捕捉模式的运行(输入脉冲的间隔测量)”)。
• 单次计数模式	通过在定时器停止运行(TEmn=0)的状态下给TSMn位写“1”，进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器，并且通过后续的计数时钟进行递减计数(参照“6.5.3(4)单次计数模式的运行”)。
• 捕捉&单次计数模式	通过在定时器停止运行(TEmn=0)的状态下给TSMn位写“1”，进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器，并且通过后续的计数时钟进行递增计数(参照“6.5.3(5)捕捉&单次计数模式的运行(高电平宽度的测量)”)。

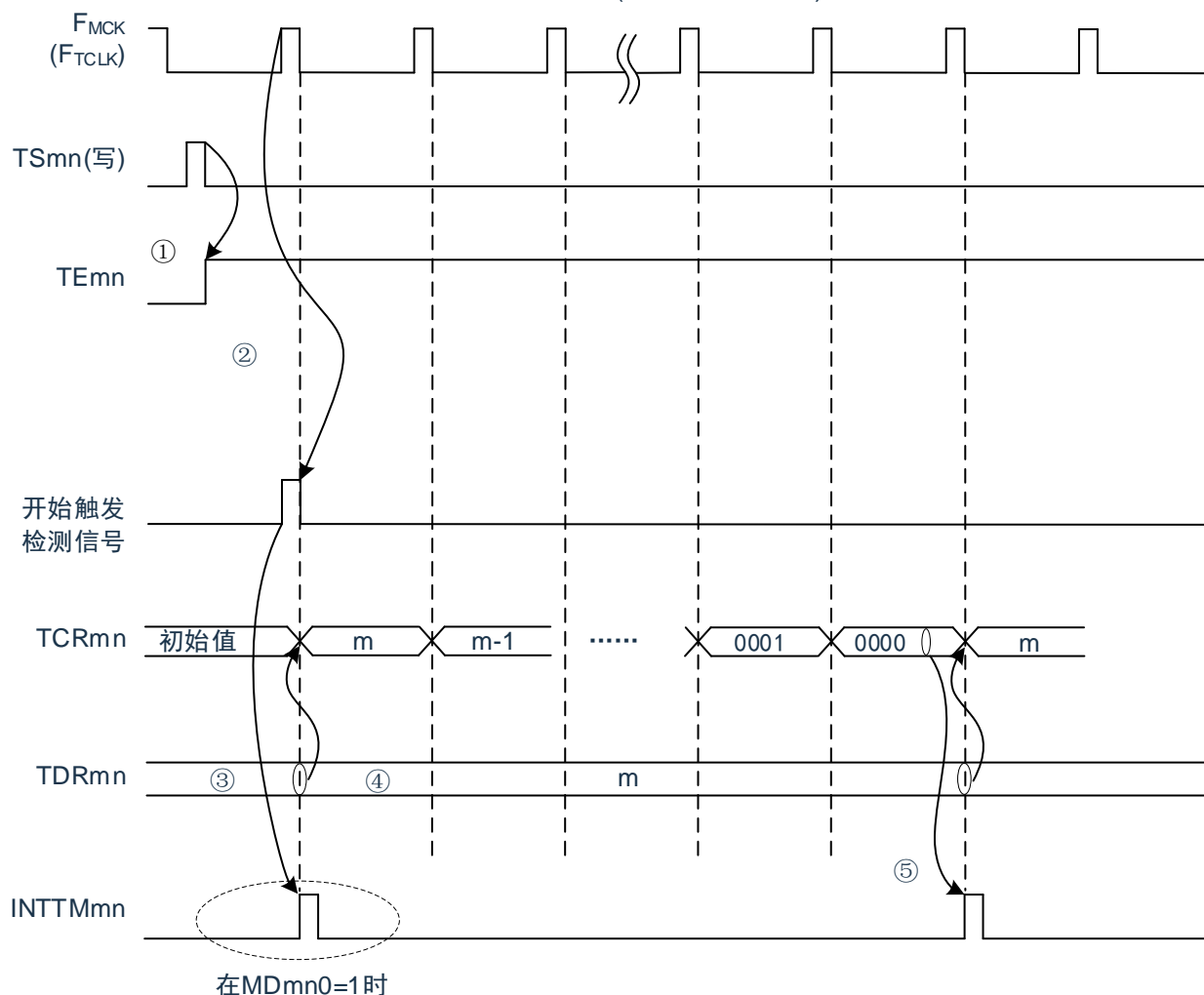
6.5.3 计数器的运行

以下说明各模式的计数器运行。

(1) 间隔定时器模式的运行

- ① 通过给TSmn位写“1”，进入运行允许状态(TEmn=1)。定时器计数寄存器mn(TCRmn)保持初始值，直到产生计数时钟为止。
- ② 通过允许运行后的第1个计数时钟(F_{MCK})产生开始触发信号。
- ③ 当MDmn0位为“1”时，通过开始触发信号产生INTTMmn。
- ④ 通过允许运行后的第1个计数时钟将定时器数据寄存器mn(TDRmn)的值装入TCRmn寄存器，并且以间隔定时器模式开始计数。
- ⑤ 如果TCRmn寄存器递减计数到“0000H”，就通过下一个计数时钟(F_{MCK})产生INTTMmn，并且在将定时器数据寄存器mn(TDRmn)的值装入TCRmn寄存器后继续计数。

图6-26：运行时序 (间隔定时器模式)



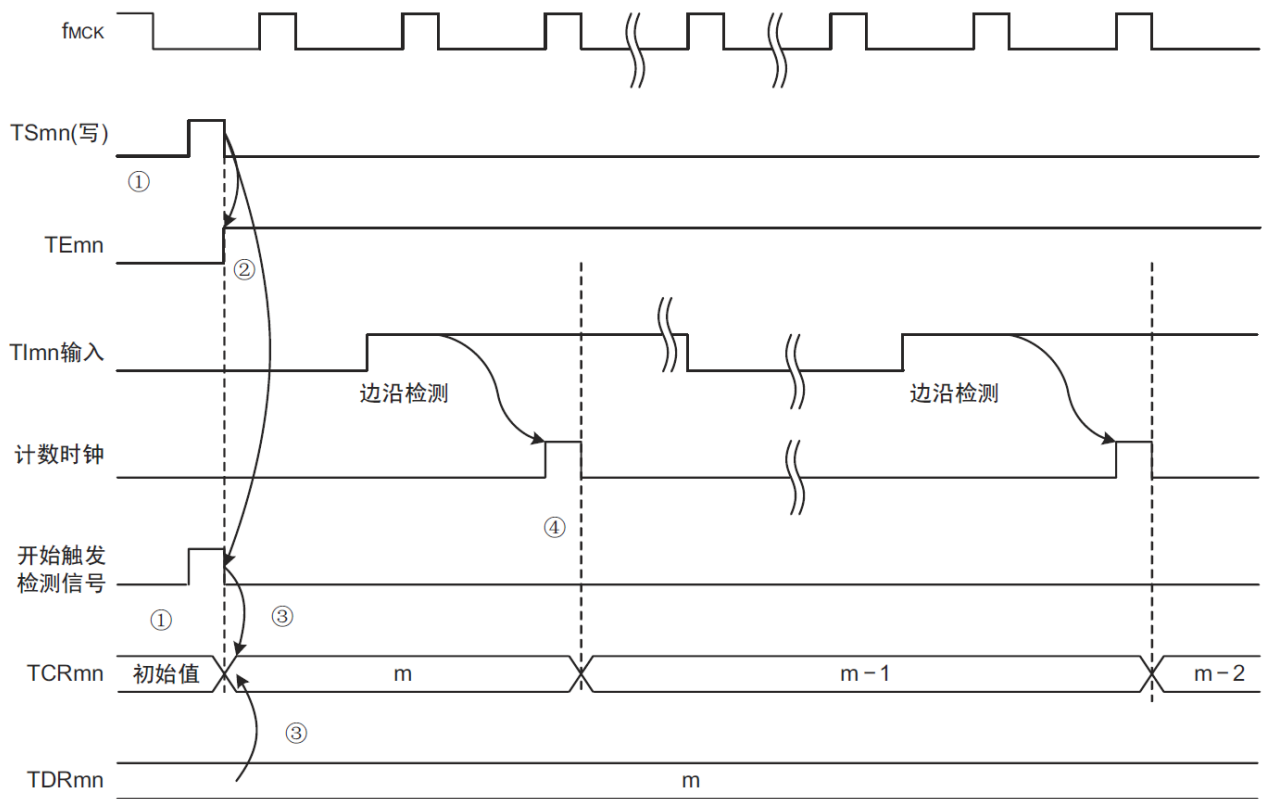
注意：因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

备注：F_{MCK}、开始触发检测信号和INTTMmn与F_{CLK}同步并且在1个时钟内有效。

(2) 事件计数器模式的运行

- ① 在运行停止状态(TE_{mn}=0)的期间, 定时器计数寄存器mn(TCR_{mn})保持初始值。
- ② 通过给TS_{mn}位写“1”, 进入运行允许状态(TE_{mn}=1)。
- ③ 在TS_{mn}位和TE_{mn}位都变为“1”的同时将定时器数据寄存器mn(TDR_{mn})的值装入TCR_{mn}寄存器, 并且开始计数。
- ④ 此后, 在TI_{mn}输入的有效边沿, 通过计数时钟对TCR_{mn}寄存器的值进行递减计数。

图6-27: 运行时序(事件计数器模式)

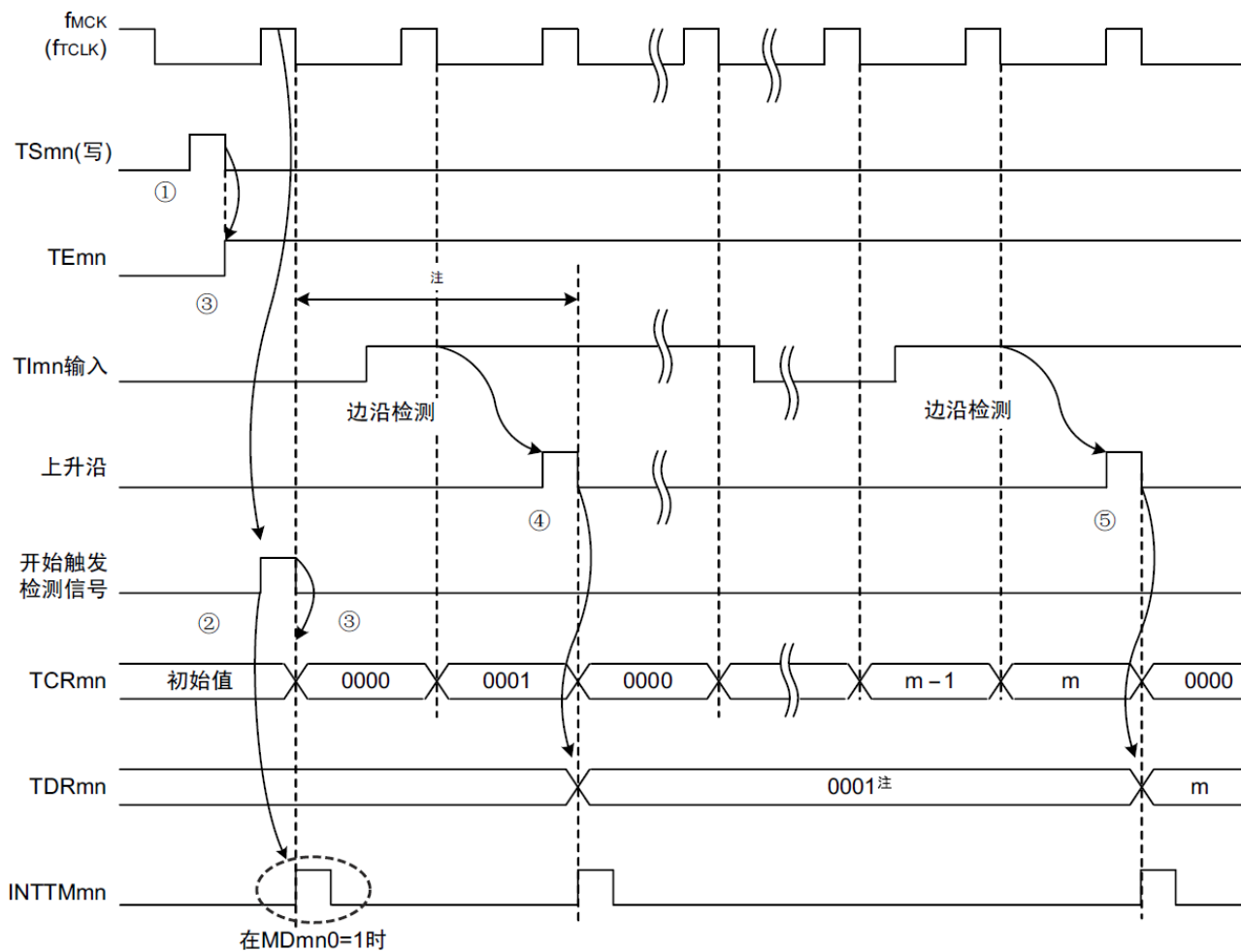


备注: 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从TI_{mn}输入开始再延迟2个F_{MCK}周期(合计3~4个周期)。1个周期的误差是因为TI_{mn}输入与计数时钟(F_{MCK})不同步。

(3) 捕捉模式的运行(输入脉冲的间隔测量)

- ① 通过给TSmn位写“1”，进入运行允许状态(TEmn=1)。
- ② 定时器计数寄存器mn(TCRmn)保持初始值，直到产生计数时钟为止。
- ③ 通过允许运行后的第1个计数时钟(F_{MCK})产生开始触发信号。然后，将“0000H”装入TCRmn寄存器并且以捕捉模式开始计数(当MDmn0位为“1”时，通过开始触发信号产生INTTMmn)。
- ④ 如果检测到TImn输入的有效边沿，就将TCRmn寄存器的值捕捉到TDRmn寄存器，并且产生INTTMmn中断。此时的捕捉值没有意义。TCRmn寄存器从“0000H”开始继续计数。
- ⑤ 如果检测到下一个TImn输入的有效边沿，就将TCRmn寄存器的值捕捉到TDRmn寄存器，并且产生INTTMmn中断。

图6-28: 运行时序(捕捉模式: 输入脉冲的间隔测量)



注：在开始前将时钟输入到TImn(有触发)时，即使没有检测到边沿也通过检测触发来开始计数，因此第1次捕捉时(④)的捕捉值不是脉冲间隔(在此例子中，0001：2个时钟间隔)，必须忽视。

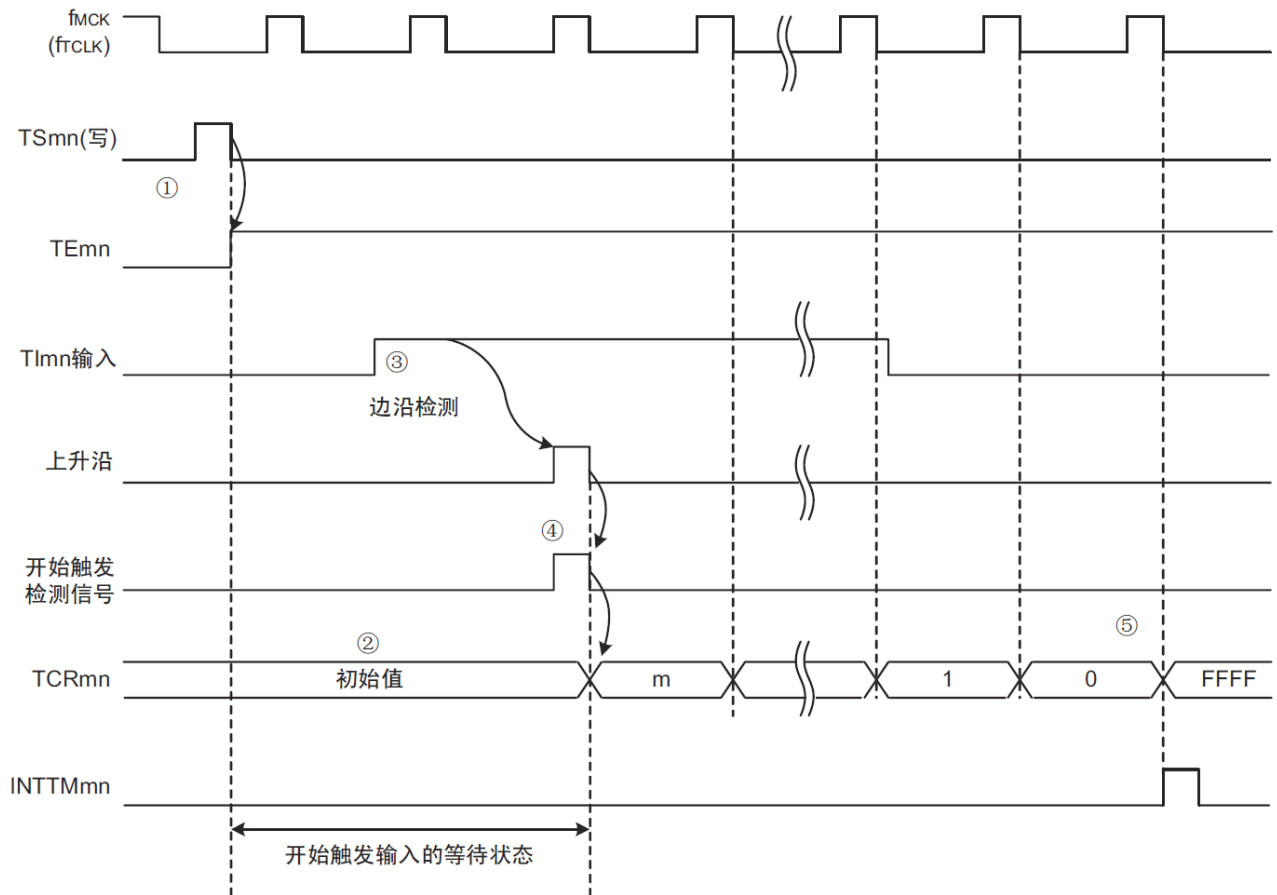
注意：因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TImn输入开始再延迟2个F_{MCK}周期(合计3~4个周期)。1个周期的误差是因为TImn输入与计数时钟(F_{MCK})不同步。

(4) 单次计数模式的运行

- ① 通过给TSMn位写“1”，进入运行允许状态(TEmn=1)。
- ② 定时器计数寄存器mn(TCRmn)保持初始值，直到产生开始触发信号为止。
- ③ 检测TIMn输入的上升沿。
- ④ 在产生开始触发信号后将TDRmn寄存器的值(m)装入TCRmn寄存器，并且开始计数。
- ⑤ 当TCRmn寄存器递减计数到“0000H”时，产生INTTMmn中断，并且TCRmn寄存器的值变为“FFFFH”，停止计数。

图6-29：运行时序(单次计数模式)

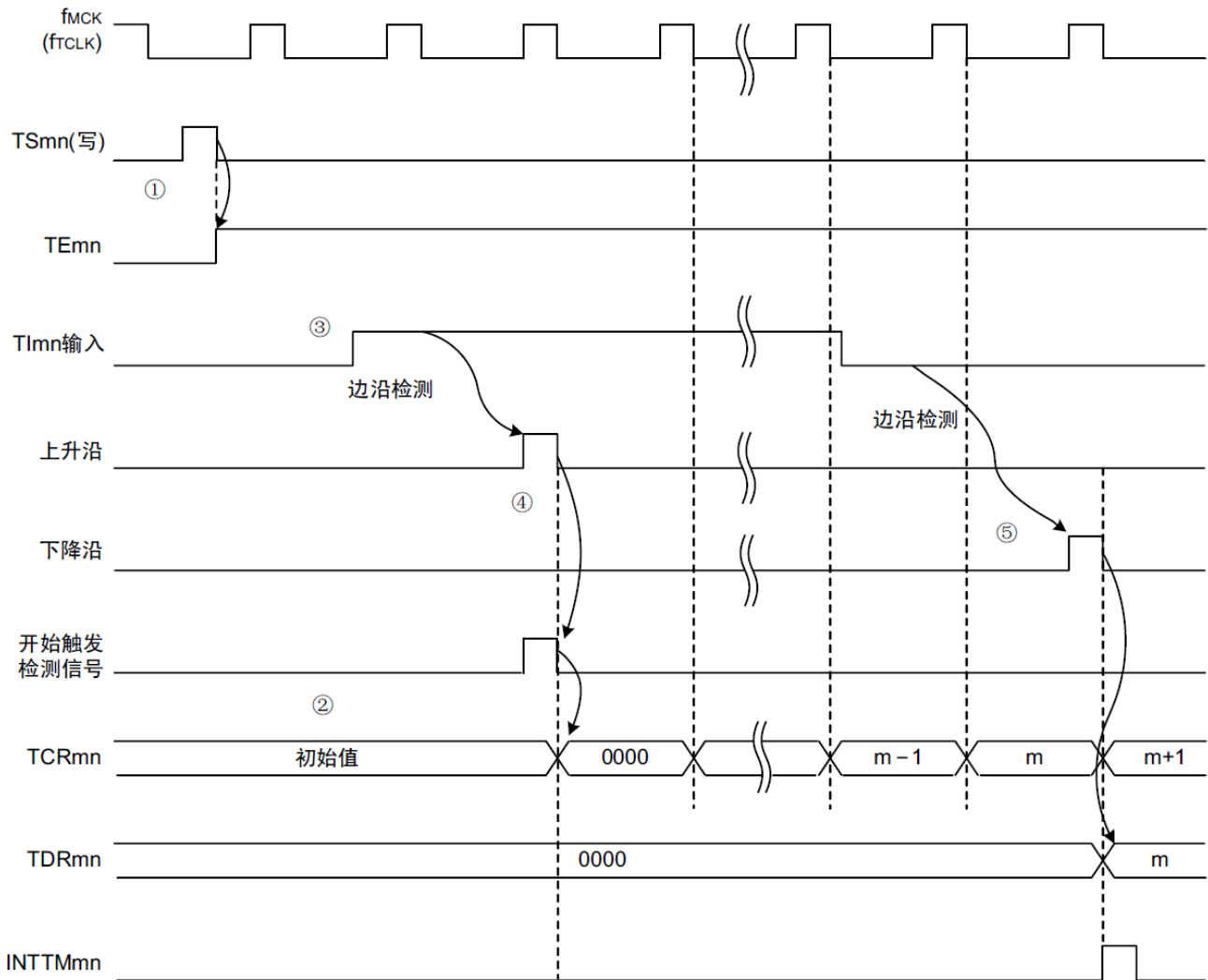


备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TIMn输入开始再延迟2个F_{MCK}周期(合计3~4个周期)。1个周期的误差是因为TIMn输入与计数时钟(F_{MCK})不同步。

(5) 捕捉&单次计数模式的运行(高电平宽度的测量)

- ① 通过给定时器通道开始寄存器m(TSm)的TSmn位写“1”，进入运行允许状态(TEmn=1)。
- ② 定时器计数寄存器mn(TCRmn)保持初始值，直到产生开始触发信号为止。
- ③ 检测TImn输入的上升沿。
- ④ 在产生开始触发信号后将“0000H”装入TCRmn寄存器，并且开始计数。
- ⑤ 如果检测到TImn输入的下沿，就将TCRmn寄存器的值捕捉到TDRmn寄存器，并且产生INTTMmn中断。

图6-30: 运行时序(捕捉&单次计数模式: 高电平宽度的测量)

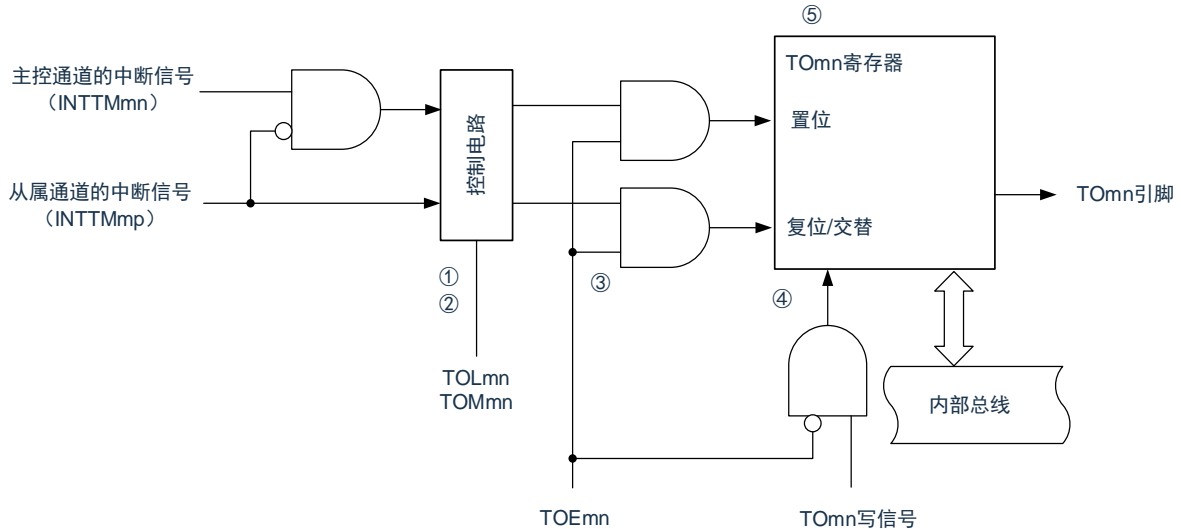


备注: 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从TImn输入开始再延迟2个F_{MCK}周期(合计3~4个周期)。1个周期的误差是因为TImn输入和计数时钟(F_{MCK})不同步。

6.6 通道输出(TOmn引脚)的控制

6.6.1 TOmn引脚输出电路的结构

图6-31：输出电路的结构



以下说明TOmn引脚的输出电路。

- ① 当TOMmn位为“0”(主控通道输出模式)时，忽视定时器输出电平寄存器m(TOLm)的设置值，只将INTTMmp(从属通道定时器中断)传给定时器输出寄存器m(TOm)。
- ② 当TOMmn位为“1”(从属通道输出模式)时，将INTTMmn(主控通道定时器中断)和INTTMmp(从属通道定时器中断)传给TOm寄存器。
此时，TOLm寄存器有效并且进行以下信号的控制：
TOLmn=0时：正相运行(INTTMmn→置位、INTTMmp→复位)
TOLmn=1时：反相运行(INTTMmn→复位、INTTMmp→置位)
当同时产生INTTMmn和INTTMmp时(PWM输出的0%输出)，优先INTTMmp(复位信号)而屏蔽INTTMmn(置位信号)。
- ③ 在允许定时器输出(TOEmn=1)的状态下，将INTTMmn(主控通道定时器中断)和INTTMmp(从属通道定时器中断)传给TOm寄存器。TOm寄存器的写操作(TOmn写信号)无效。
当TOEmn位为“1”时，除了中断信号以外，不改变TOmn引脚的输出。
要对TOmn引脚的输出电平进行初始化时，需要在设置为禁止定时器输出(TOEmn=0)后给TOm寄存器写值。
- ④ 在禁止定时器输出(TOEmn=0)的状态下，对象通道的TOmn位的写操作(TOmn写信号)有效。当定时器输出为禁止状态(TOEmn=0)时，不将INTTMmn(主控通道定时器中断)和INTTMmp(从属通道定时器中断)传给TOm寄存器。
- ⑤ 能随时读TOm寄存器，并且能确认TOmn引脚的输出电平。

备注：m：单元号(m=0)

n：通道号，n=0~3(主控通道：n=0、2)

p：从属通道号

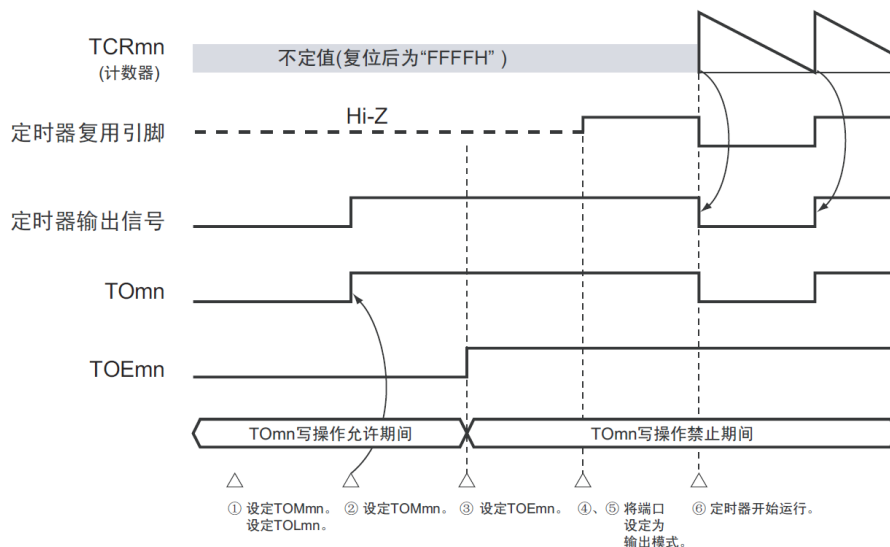
n=0：p=1、2、3

n=2：p=3

6.6.2 TOmn引脚的输出设置

从TOmn输出引脚的初始设置到定时器开始运行的步骤和状态变化如下所示。

图6-32：从设置定时器的输出到开始运行的状态变化



- ① 设置定时器输出的运行模式。
 - TOMmn位(0: 主控通道输出模式、1: 从属通道输出模式)
 - TOLmn位(0: 正逻辑输出、1: 负逻辑输出)
- ② 通过设置定时器输出寄存器m(TOm), 将定时器输出信号设置为初始状态。
- ③ 给TOEmn位写“1”, 允许定时器输出(禁止写TOm寄存器)。
- ④ 通过端口模式控制寄存器(PMCxx)将端口设置为数字输入/输出(参照“6.3.15控制定时器输入/输出引脚端口功能的寄存器”)。
- ⑤ 将端口的输入/输出设置为输出(参照“6.3.15控制定时器输入/输出引脚端口功能的寄存器”)。
- ⑥ 允许定时器运行(TSmn=1)。

备注：m：单元号(m=0)n：通道号(n=0~3)

6.6.3 通道输出运行的注意事项

1) 有关定时器运行中的TOM_m、TOE_m、TOL_m、TOM_m寄存器的设置值变更

定时器的运行(定时器计数寄存器mn(TCR_{mn})和定时器数据寄存器mn(TDR_{mn})的运行)和TOM_m输出电路相互独立。因此,定时器输出寄存器m(TO_m)、定时器输出允许寄存器m(TE_m)和定时器输出电平寄存器m(TOL_m)的设置值的变更不会影响定时器的运行,能在定时器运行中更改设置值。但是,为了在各定时器的运行中从TOM_m引脚输出期待的波形,必须设置为6.8和6.9所示的各运行的寄存器设置内容例子的值。

如果在产生各通道的定时器中断(INTT_{Mmn})信号前后更改除了TO_m寄存器以外的TE_m寄存器和TOL_m寄存器的设置值,就根据是在产生定时器中断(INTT_{Mmn})信号前更改还是在产生后更改,TOM_m引脚输出的波形可能不同。

备注:m:单元号(m=0)n:通道号(n=0~3)

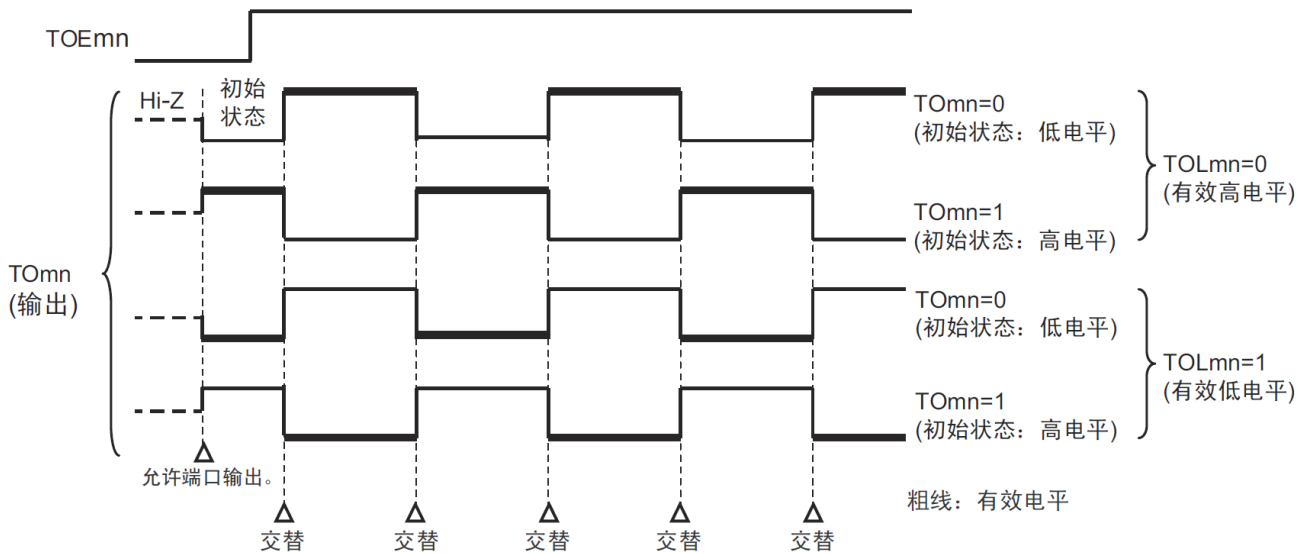
2) 有关TOM_m引脚的初始电平和定时器开始运行后的输出电平

在允许端口输出前并且在禁止定时器输出(TE_m=0)的状态下写定时器输出寄存器m(TO_m),在更改初始电平后设置为定时器输出允许状态(TE_m=1)时的TOM_m引脚输出电平的变化如下所示。

(a) 在主导通道输出模式(TOM_m=0)中开始运行的情况

在主导通道输出模式(TOM_m=0)中,定时器输出电平寄存器m(TOL_m)的设置无效。如果在设置初始电平后开始定时器的运行,就通过产生交替信号反相TOM_m引脚的输出电平。

图6-33: 交替输出时(TOM_m=0)的TOM_m引脚输出状态



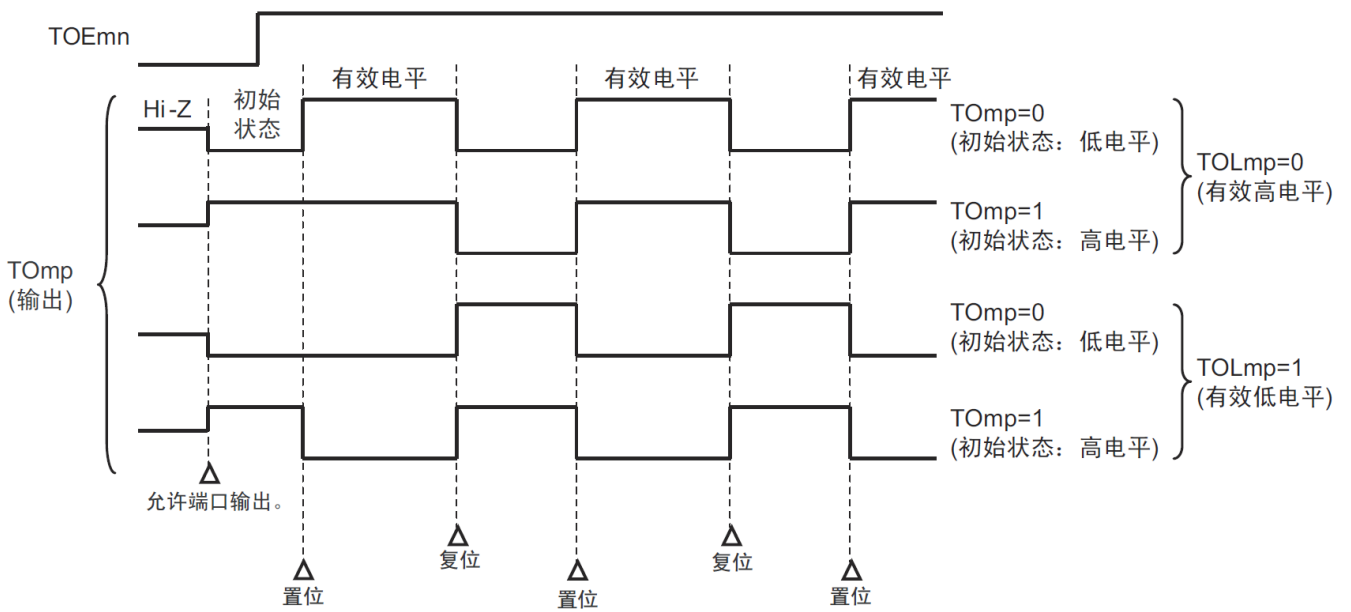
备注:

1. 交替: 反相TOM_m引脚的输出状态。
2. m: 单元号(m=0)n: 通道号(n=0~3)

(b) 在从属通道输出模式(TOMmn=1)中开始运行的情况(PWM输出)

在从属通道输出模式(TOMmn=1)中，有效电平取决于定时器输出电平寄存器m(TOLmn)的设置。

图6-34：PWM输出时(TOMmn=1)的TOmn引脚输出状态



备注:

1. 置位: TOmp引脚的输出信号从无效电平变为有效电平。
复位: TOmp引脚的输出信号从有效电平变为无效电平。
2. m: 单元号(m=0)n: 通道号(p=1~3)

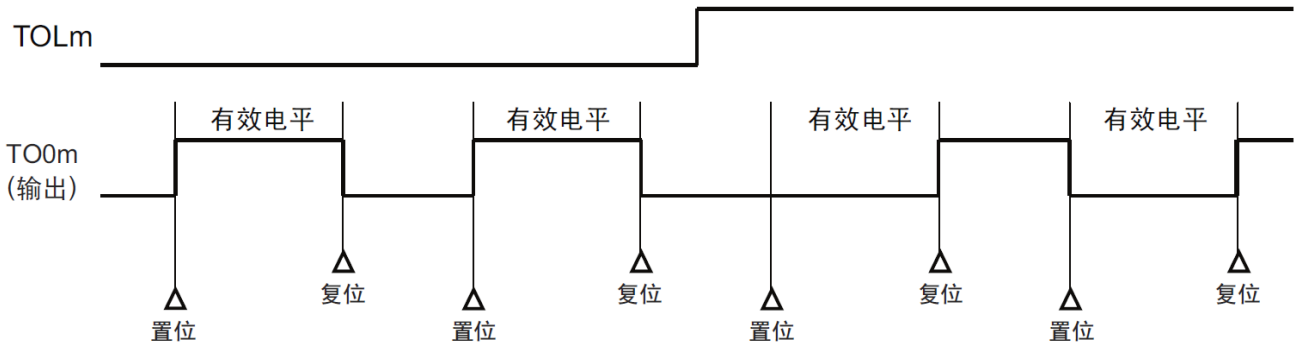
3) 有关从属通道输出模式(TOMmn=1)的TOMn引脚变化

(a) 在定时器运行中更改定时器输出电平寄存器m(TOLm)的设置的情况

如果在定时器运行中更改TOLm寄存器的设置，就在产生TOMn引脚变化条件时设置有效。无法通过改写TOLm寄存器来改变TOMn引脚的输出电平。

当TOMmn位为“1”时，在定时器运行中(TEmn=1)更改TOLm寄存器的值时的运行如下所示。

图6-35：在定时器运行中更改TOLm寄存器的内容时的运行



备注：

1. 置位：TOMn引脚的输出信号从无效电平变为有效电平。
复位：TOMn引脚的输出信号从有效电平变为无效电平。
2. m：单元号(m=0)n：通道号(n=0~3)

(b) 置位/复位时序

为了在PWM输出时实现0%和100%的输出，通过从属通道将产生主控通道定时器中断(INTTMmn)时的TOMn引脚/TOMn位的置位时序延迟1个计数时钟。

当置位条件和复位条件同时产生时，优先复位条件。

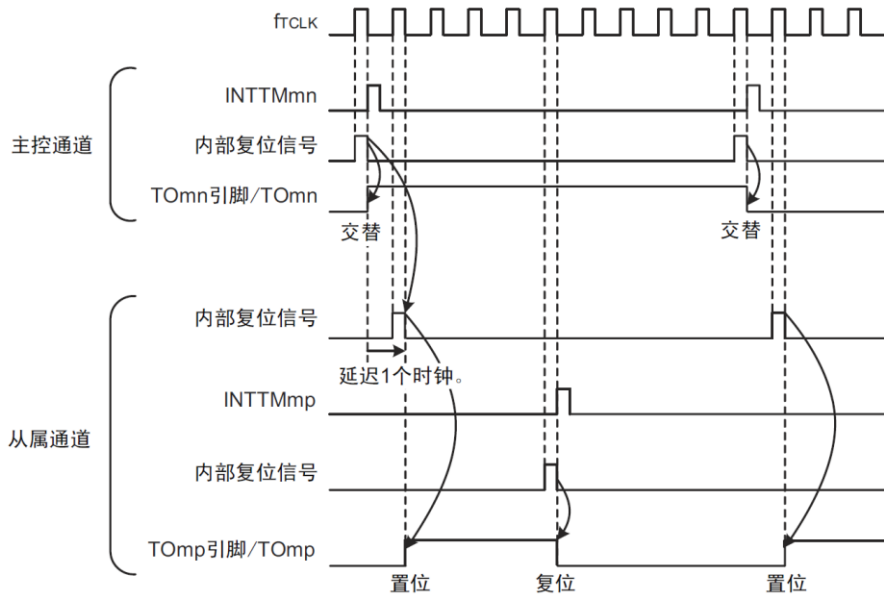
按照以下方法设置主控/从属通道时的置位/复位运行状态如图6-35所示。

主控通道：TOEmn=1、TOMmn=0、TOLmn=0

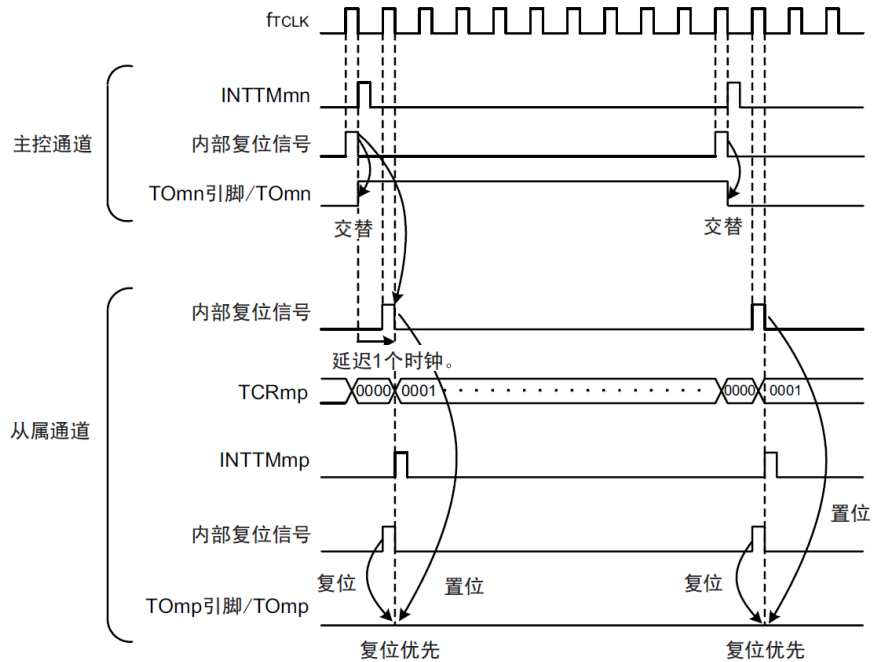
从属通道：TOEmp=1、TOMmp=1、TOLmp=0

图6-36：置位/复位时序运行状态

(1) 基本运行时序



(2) 0%占空比的运行时序



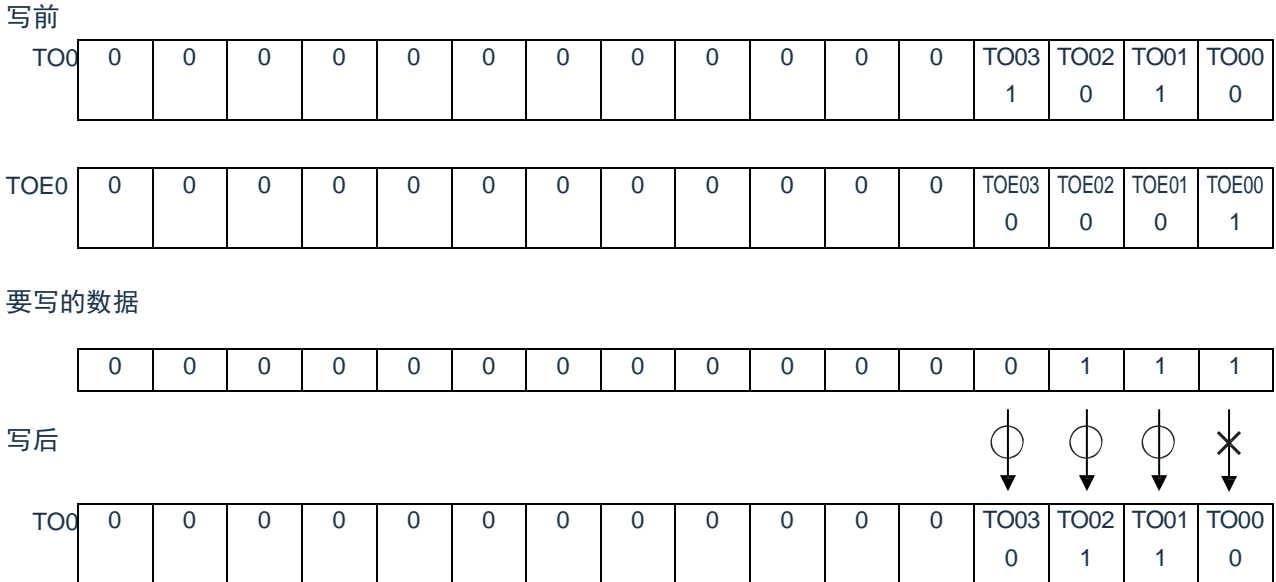
备注：

1. 内部复位信号：TOmn引脚的复位/交替信号
内部置位信号：TOmn引脚的置位信号
2. m：单元号(m=0)
n：通道号n=0~3(主控通道：n=0、2)
p：从属通道号
n=0：p=1、2、3
n=2：p=3

6.6.4 TOmn位的一次性操作

和定时器通道开始寄存器m(TSm)相同，定时器输出寄存器m(TOm)有全部通道的设置位(TOmn)，因此能一次性地操作全部通道的TOmn位。

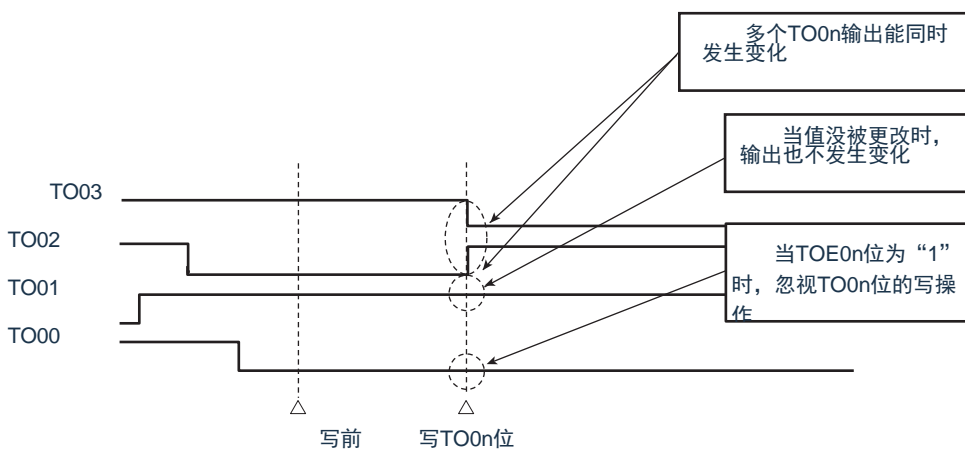
图6-37: TO0n位的一次性操作例子



只能写TOEmn位为“0”的TOmn位，忽视TOEmn位为“1”的TOmn位的写操作。

TOEmn位为“1”的TOmn(通道输出)不受写操作的影响，即使写TOmn位也被忽视，由定时器运行引起的输出变化正常进行。

图6-38: 一次性操作TO0n位时的TO0n引脚状态



备注: m: 单元号(m=0)n: 通道号(n=0~3)

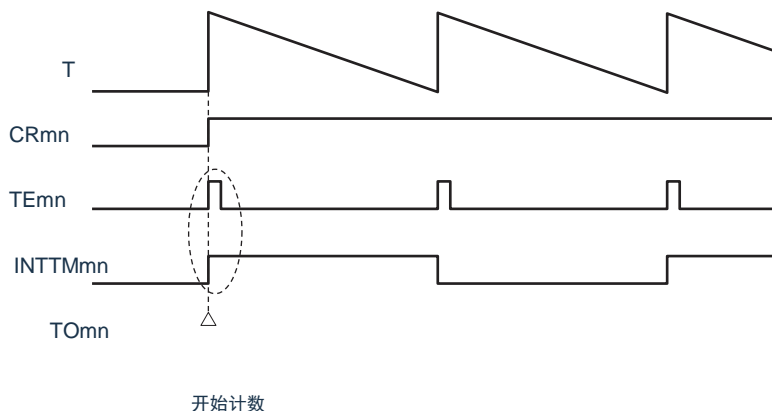
6.6.5 有关开始计数时的定时器中断和TOMn引脚输出

在间隔定时器模式或者捕捉模式中，定时器模式寄存器mn(TMRmn)的MDmn0位是设置是否在开始计数时产生定时器中断的位。

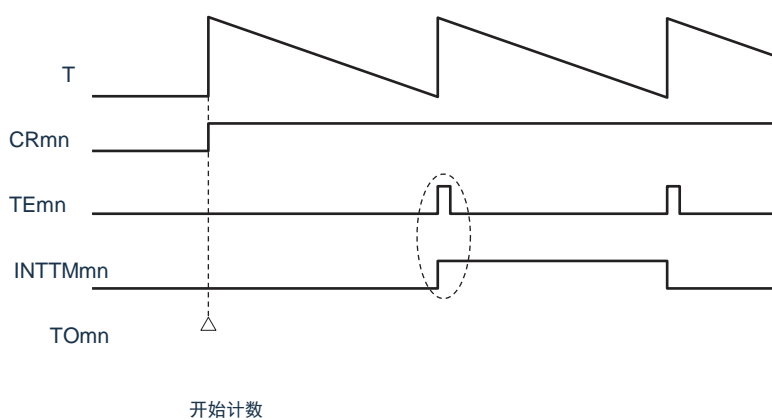
当MDmn0位为“1”时，能通过产生定时器中断(INTTMmn)得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和TOMn输出。设置为间隔定时器模式(TOEmn=1、TOMmn=0)时的运行例子如下所示。

图6-39：开始计数时的定时器中断和TOMn输出的运行例子

(a) MDmn0位为“1”的情况



(b) MDmn0位为“0”的情况



当MDmn0位为“1”时，在开始计数时输出定时器中断(INTTMmn)并且TOMn进行交替输出。

当MDmn0位为“0”时，在开始计数时不输出定时器中断(INTTMmn)并且TOMn也不发生变化，而在对1个周期进行计数后输出INTTMmn并且TOMn进行交替输出。

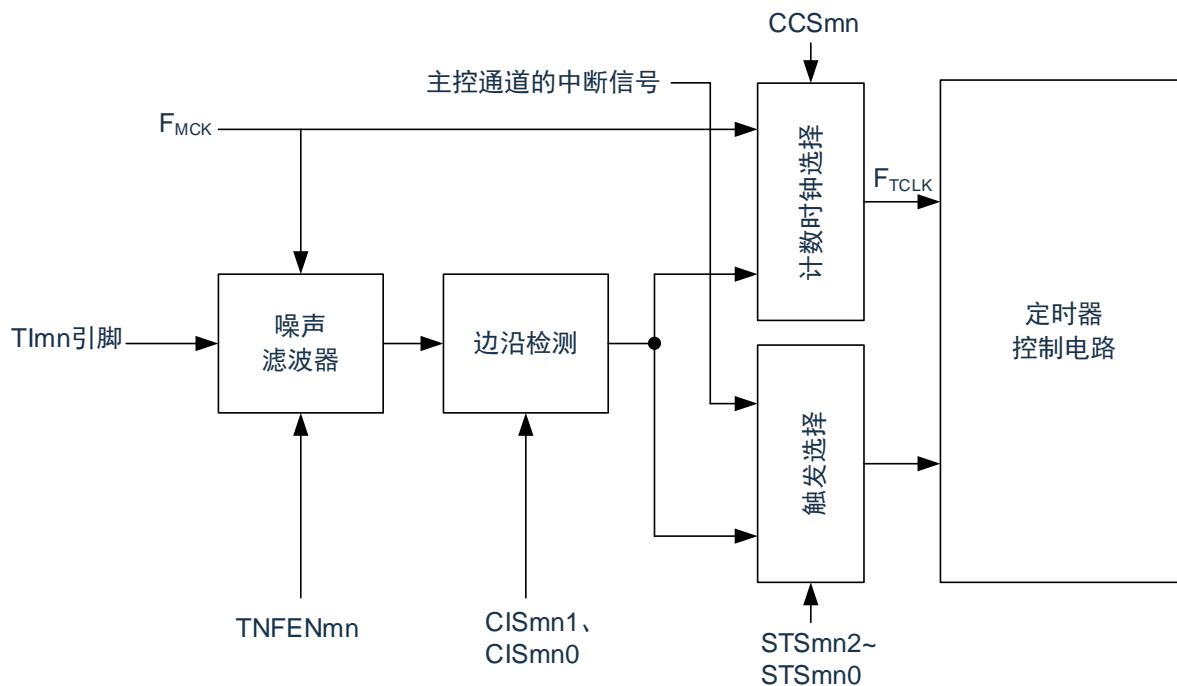
备注：m：单元号(m=0)n：通道号(n=0~3)

6.7 定时器输入(TImn)的控制

6.7.1 TImn引脚输入电路的结构

定时器输入引脚的信号通过噪声滤波器和边沿检测电路输入到定时器控制电路。对于需要消除噪声的引脚，必须将对应的引脚噪声滤波器置为有效。输入电路的结构如下所示。

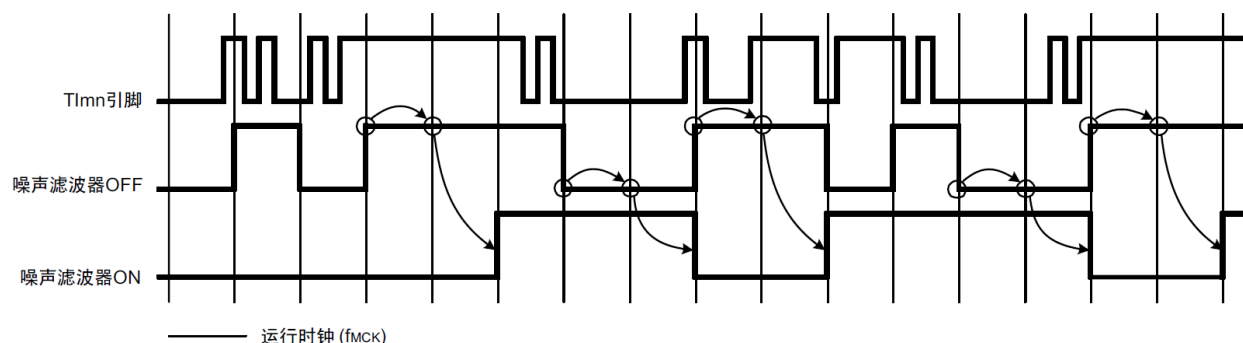
图6-40：输入电路的结构



6.7.2 噪声滤波器

当噪声滤波器无效时，只通过通道n的运行时钟(F_{MCK})进行同步；当噪声滤波器有效时，在通过通道n的运行时钟(F_{MCK})进行同步后检测2个时钟是否一致。TM4Imn输入引脚在噪声滤波器ON或者OFF的情况下，经过噪声滤波器电路后的波形如下所示。

图6-41：TImn输入引脚在噪声滤波器ON或者OFF情况下的采样波形



注意：TImn引脚的输入波形用于说明噪声滤波器ON或者OFF的运行。实际使用时，必须按照“AC特性”所示的TImn输入高低电平宽度进行输入。

6.7.3 操作通道输入时的注意事项

在设置为不使用定时器输入引脚时，不给噪声滤波器电路提供运行时钟。因此，从设置为使用定时器输入引脚到设置定时器输入引脚对应的通道运行允许触发，需要以下的等待时间。

(1) 噪声滤波器为OFF的情况

如果在定时器模式寄存器mn(TMRmn)的bit12(CCSmn)、bit9(STSmn1)和bit8(STSmn0)全都为“0”的状态下将任意一位置位，就必须至少在经过2个运行时钟(F_{MCK})周期后将定时器通道开始寄存器(TSm)的运行允许触发置位。

(2) 噪声滤波器为ON的情况

如果在定时器模式寄存器mn(TMRmn)的bit12(CCSmn)、bit9(STSmn1)和bit8(STSmn0)全都为“0”的状态下将任意一位置位，就必须至少在经过4个运行时钟(F_{MCK})周期后将定时器通道开始寄存器(TSm)的运行允许触发置位。

6.8 通用定时器单元的独立通道运行功能

6.8.1 作为间隔定时器/方波输出的运行

(1) 间隔定时器

能用作以固定间隔产生INTTMmn(定时器中断)的基准定时器。中断产生周期能用以下计算式进行计算:

$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟周期} \times (\text{TDRmn的设置值} + 1)$$

(2) 作为方波输出的运行

TOMn在产生INTTMmn的同时进行交替输出, 输出占空比为50%的方波。

TOMn输出方波的周期和频率能用以下计算式进行计算:

- TOMn输出的方波周期 = 计数时钟周期 \times (TDRmn的设置值 + 1) \times 2

- TOMn输出的方波频率 = 计数时钟频率 / {(TDRmn的设置值 + 1) \times 2}

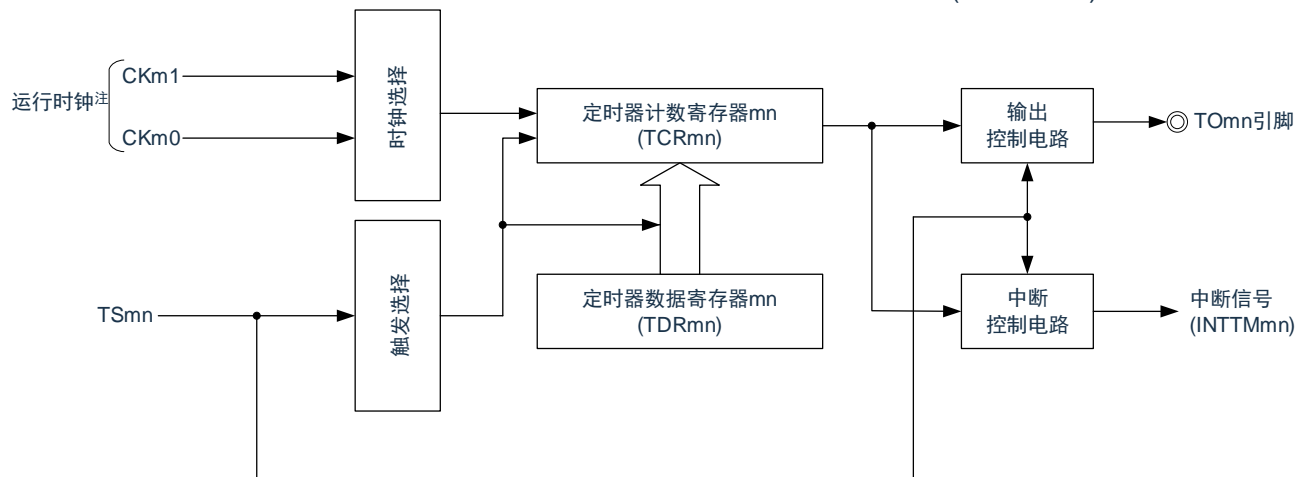
在间隔定时器模式中, 定时器计数寄存器mn(TCRmn)用作递减计数器。

在将定时器通道开始寄存器m(TSm)的通道开始触发位(TSmn、TSHm1、TSHm3)置“1”后, 通过第1个计数时钟将定时器数据寄存器mn(TDRmn)的值装入TCRmn寄存器。此时, 如果定时器模式寄存器n(TMRmn)的MDmn0位为“0”, 就不输出INTTMmn并且TOMn也不进行交替输出。如果TMRmn寄存器的MDmn0位为“1”, 就输出INTTMmn并且TOMn进行交替输出。然后, TCRmn寄存器通过计数时钟进行递减计数。

如果TCRmn变为“0000H”, 就通过下一个计数时钟输出INTTMmn并且TOMn进行交替输出。同时, 再次将TDRmn寄存器的值装入TCRmn寄存器。此后, 继续同样的运行。

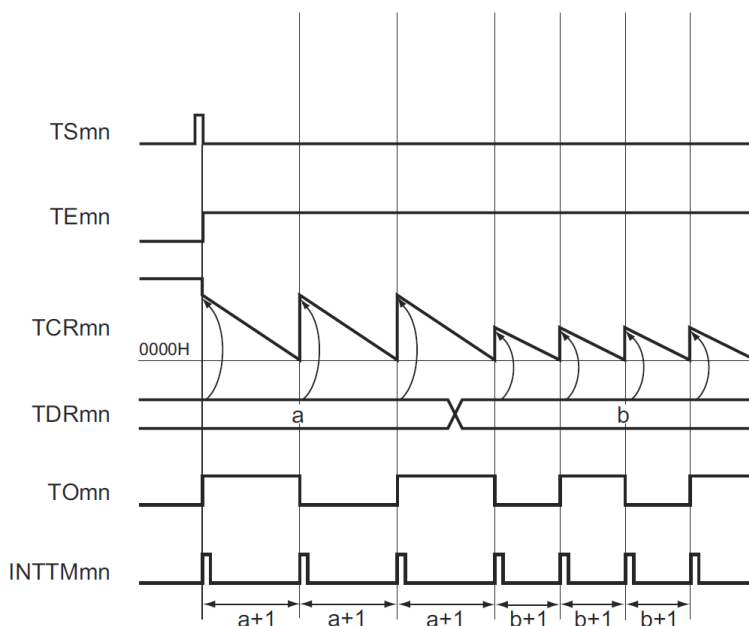
能随时改写TDRmn寄存器, 改写的TDRmn寄存器的值从下一个周期开始有效。

图6-42: 作为间隔定时器/方波输出运行的基本时序例子(MDmn0=1)



注: 在通道1和通道3时, 能从CKm0、CKm1、CKm2和CKm3中选择时钟。

图6-43: 作为间隔定时器/方波输出运行的基本时序例子(MDmn0=1)

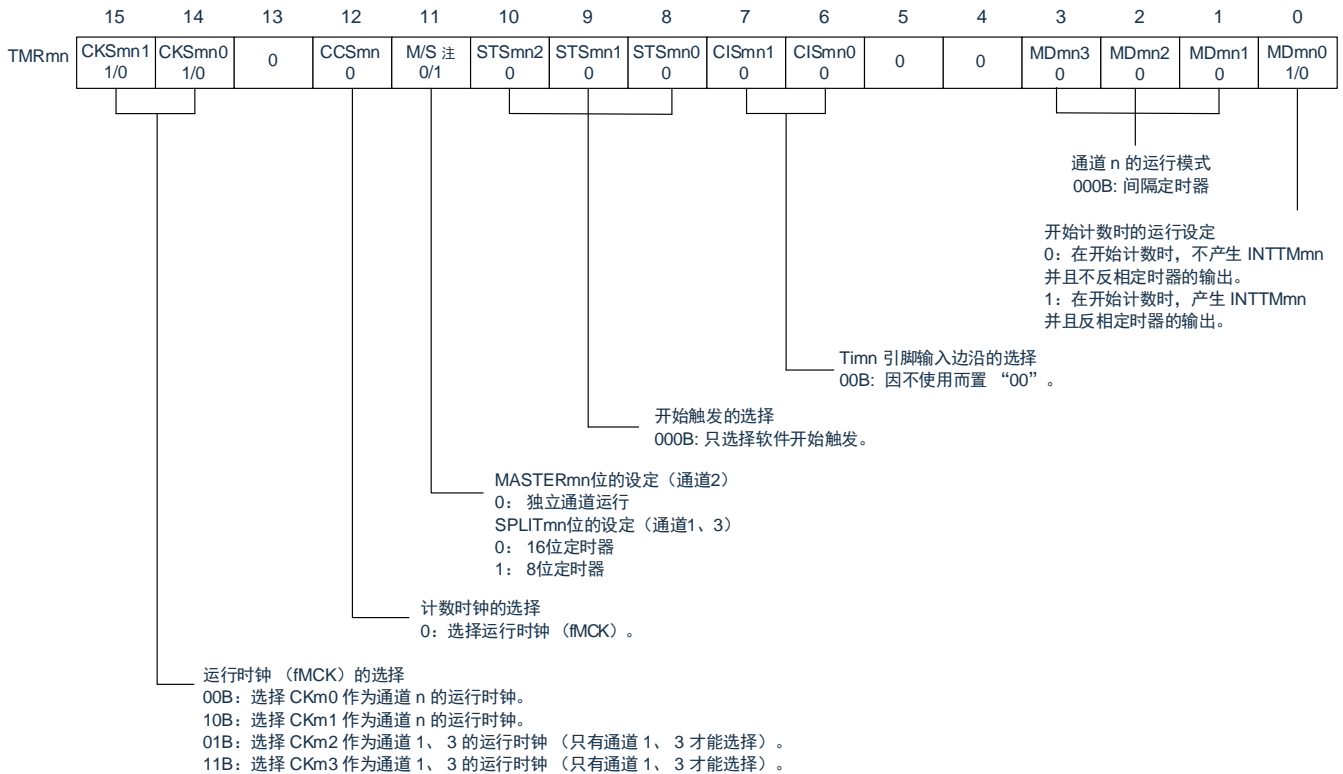


备注:

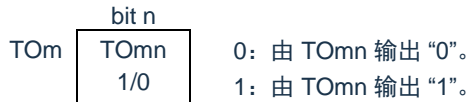
1. m: 单元号(m=0)n: 通道号(n=0~3)
2. TSmn: 定时器通道开始寄存器m(TSm)的bit n
 TEMn: 定时器通道允许状态寄存器m(TEm)的bit n
 TCRmn: 定时器计数寄存器mn(TCRmn)
 TDRmn: 定时器数据寄存器mn(TDRmn)
 TOMn: TOMn引脚输出信号

图6-44: 间隔定时器/方波输出时的寄存器设置内容例子

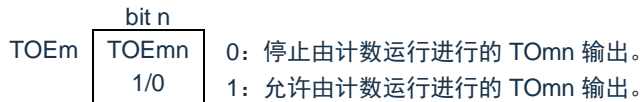
(a) 定时器模式寄存器mn(TMRmn)



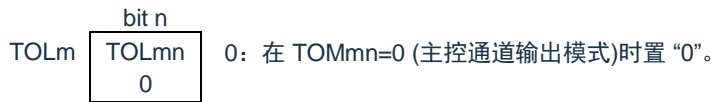
(b) 定时器输出寄存器 m (TOM)



(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



注: TMRm2: MASTERmn 位

TMRm1、TMRm3: SPLITmn位

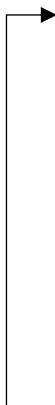
TMRm0: 固定为“0”。

备注: m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图6-45: 间隔定时器 / 方波输出功能时的操作步骤

	软件操作	硬件状态
TAU初始设定		定时器单元m的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0(PER0)的TM4mEN位置“1”。	定时器单元m的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m(TPSm)。 确定CKm0 ~ CKm3的时钟频率。	
通道初始设定	设定定时器模式寄存器mn(TMRmn)(确定通道的运行模式)。 给定时器数据寄存器mn(TDRmn)设定间隔(周期)值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始运行	使用TOMn输出的情况: 将定时器输出模式寄存器m(TOMm)的TOMmn位置“0” (主控通道输出模式)。 将TOLmn位置“0”。	TOMn引脚处于Hi-Z输出状态。
	设定TOMn位, 确定TOMn输出的初始电平。 将TOEmn位置“1”, 允许TOMn输出。 将端口寄存器和端口模式寄存器置“0”。	当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出TOMn初始设定的电平。 因为通道处于运行停止状态, 所以TOMn不变。 TOMn引脚输出TOMn设定的电平。
	(只在使用TOMn输出并且重新开始时, 将TOEmn位置“1”) 将TSMn(TSHm1、TSHm3)位置“1”。 因为TSMn(TSHm1、TSHm3)位是触发位, 所以自动返回到“0”。	TEmn(TEHm1、TEHm3)位变为“1”并且开始计数。 将TDRmn寄存器的值装入定时器计数寄存器mn(TCRmn)。当TMRmn寄存器的MDmn0位为“1”时, 产生INTTMmn并且TOMn进行交替输出。
	能任意更改TDRmn寄存器的设定值。 能随时读TCRmn寄存器。 不能使用TSRmn寄存器。 能更改TOM寄存器和TOEm寄存器的设定值。 禁止更改TMRmn寄存器、TOMmn位和TOLmn位的设定值。	计数器(TCRmn)进行递减计数。如果计数到“0000H”, 就再次将TDRmn寄存器的值装入TCRmn寄存器并且继续计数。当检测到TCRmn为“0000H”时, 产生INTTMmn并且TOMn进行交替输出。此后, 重复此运行。
停止运行	将TTmn(TTHm1、TTHm3)位置“1”。 因为TTmn(TTHm1、TTHm3)位是触发位, 所以自动返回到“0”。	TEmn(TEHm1、TEHmn)位变为“0”并且停止计数。 TCRmn寄存器保持计数值而停止计数。 TOMn输出不被初始化而保持状态。
	将TOEmn位置“0”并且给TOMn位设定值。	TOMn引脚输出TOMn位设定的电平。
TAU停止	要保持TOMn引脚输出电平的情况: 在给端口寄存器设定要维持的值后将TOMn位置“0”。 不需要保持TOMn引脚输出电平的情况: 不需要设定。	通过端口功能保持TOMn引脚的输出电平。
	将PER0寄存器的TM4mEN位置“0”。	定时器单元m的输入时钟处于停止提供状态。 对全部电路和各通道的SFR进行初始化。 (TOMn位变为“0”并且TOMn引脚变为端口功能)

重新开始运行



6.8.2 作为外部事件计数器的运行

能用作事件计数器，对检测到的TImn引脚输入的有效边沿(外部事件)进行计数，如果达到规定的计数值，就产生中断。规定的计数值能用以下计算式进行计算：

$$\text{规定的计数值} = \text{TDRmn的设置值} + 1$$

在事件计数器模式中，定时器计数寄存器mn(TCRmn)用作递减计数器。

通过将定时器通道开始寄存器m(TSm)的任意通道开始触发位(TSmn、TSHm1、TSHm3)置“1”，将定时器数据寄存器mn(TDRmn)的值装入TCRmn寄存器。

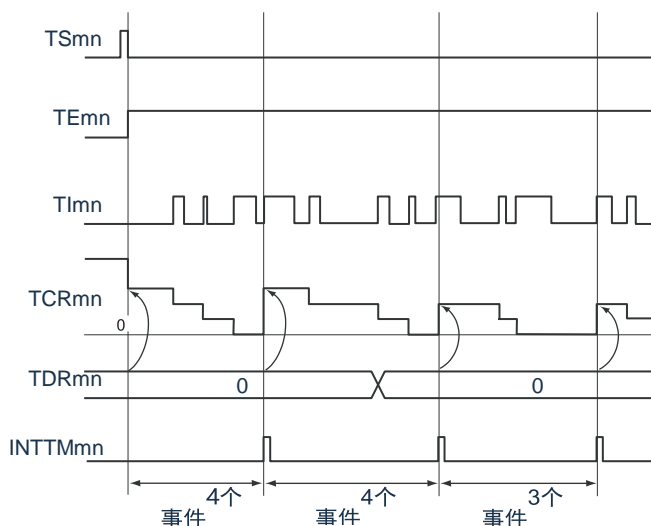
TCRmn寄存器在检测到TImn引脚输入的有效边沿的同时进行递减计数。如果TCRmn变为“0000H”，就再次装入TDRmn寄存器的值并且输出INTTMmn。

此后，继续同样的运行。

因为TOMn引脚根据外部事件输出不规则的波形，所以必须将定时器输出允许寄存器m(TOEm)的TOEmn位置“0”，停止输出。

能随时改写TDRmn寄存器，改写的TDRmn寄存器的值在下一个计数期间有效。

图6-46：作为外部事件计数器运行的基本时序例子

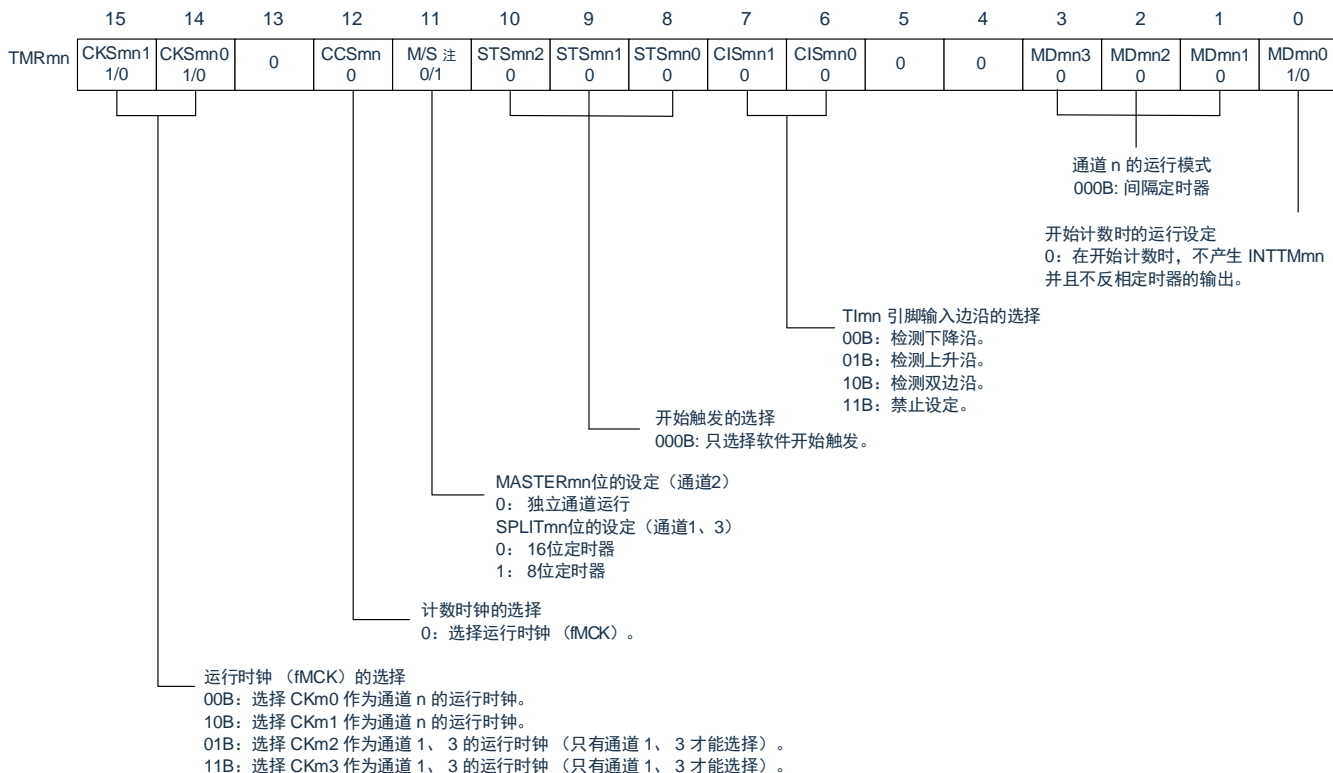


备注：

1. m：单元号(m=0) n：通道号(n=0~3)
2. TSmn：定时器通道开始寄存器m(TSm)的bit n
TE mn：定时器通道允许状态寄存器m(TEm)的bit n
TImn：TImn引脚输入信号
TCRmn：定时器计数寄存器mn(TCRmn)
TDRmn：定时器数据寄存器mn(TDRmn)

图6-47：外部事件计数器模式时的寄存器设置内容例子

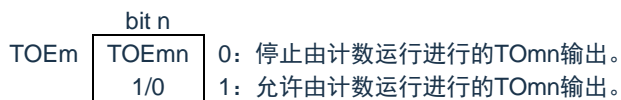
(a) 定时器模式寄存器mn(TMRmn)



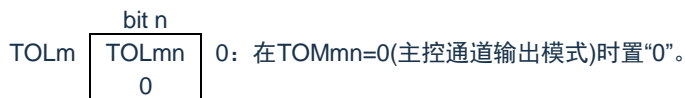
(b) 定时器输出寄存器 m(TOm)



(c) 定时器输出允许寄存器 m(TOEm)



(d) 定时器输出电平寄存器 m(TOLm)



(e) 定时器输出模式寄存器 m(TOMm)



注: TMRm2: MASTERmn位

TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 通道号 (n=0~3)

图6-48: 外部事件计数器功能时的操作步骤

	软件操作	硬件状态
Timer4初始设定		定时器单元m的输入时钟处于停止提供的状态。(停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0(PER0)的TM4mEN位置“1”。	定时器单元m的输入时钟处于提供状态, 各通道处于运行停止状态。(开始提供时钟, 能写各寄存器)
	设定定时器的时钟选择寄存器m(TPSm)。确定CKm0 ~ CKm3的时钟频率。	
通道初始设定	将噪声滤波器允许寄存器1(NFEN12)的对应位置“0”(OFF)或者“1”(ON)。设定定时器模式寄存器mn(TMRmn)(确定通道的运行模式)。给定定时器数据寄存器mn(TDRmn)设定计数值。将定时器输出允许寄存器m(TOEm)的TOEmn位置“0”。	通道处于运行停止状态。(提供时钟, 消耗一部分电力)
开始运行	将TSMn位置“1”。因为TSMn位是触发位, 所以自动返回到“0”。	TEmn位变为“1”并且开始计数。将TDRmn寄存器的值装入定时器计数寄存器mn(TCRmn), 进入TIMn引脚输入边沿的检测等待状态。
运行中	能任意更改TDRmn寄存器的设定值。能随时读TCRmn寄存器。不使用TSRmn寄存器。禁止更改TMRmn寄存器、TOMmn位、TOLmn位、TOMn位和TOEmn位的设定值。	每当检测到TIMn引脚的输入边沿时, 计数器(TCRmn)就进行递减计数, 如果计数到“0000H”就再次将TDRmn寄存器的值装入TCRmn寄存器并且继续计数。当检测到TCRmn为“0000H”时, 产生INTTMmn。此后, 重复此运行。
停止运行	将TTmn位置“1”。因为TTmn位是触发位, 所以自动返回到“0”。	TEmn位变为“0”并且停止计数。TCRmn寄存器保持计数值而停止计数。
Timer4停止	将PER0寄存器的TM4mEN位置“0”。	定时器单元m的输入时钟处于停止提供状态。对全部电路和各通道的SFR进行初始化。

重新开始运行

6.8.3 作为分频器的运行(只限于单元0 的通道0)

能对 TI00 引脚输入的时钟进行分频并且用作 TO00 引脚输出的分频器。

TO00 输出的分频时钟频率能用以下计算式进行计算：

- 选择上升沿或者下降沿的情况：
分频时钟频率=输入时钟频率/{(TDR00的设置值+1)×2}
- 选择双边沿的情况：
分频时钟频率≈输入时钟频率/(TDR00的设置值+1)

在间隔定时器模式中，定时器计数寄存器 00 (TCR00)用作递减计数器。

在将定时器通道开始寄存器 0 (TS0)的通道开始触发位 (TS00)置“1”后，通过检测到 TI00 的有效边沿将定时器数据寄存器 00 (TDR00)的值装入 TCR00 寄存器。此时，如果定时器模式寄存器 00 (TMR00)的 MD000 位为“0”，就不输出 INTTM00 并且 TO00 不进行交替输出；如果 TMR00 寄存器的 MD000 位为“1”，就输出 INTTM00 并且 TO00 进行交替输出。

然后，TCR00 寄存器通过 TI00 引脚输入的有效边沿进行递减计数。如果 TCR00 变为“0000H”，TO00就进行交替输出。同时，将 TDR00 寄存器的值装入 TCR00 寄存器并且继续计数。

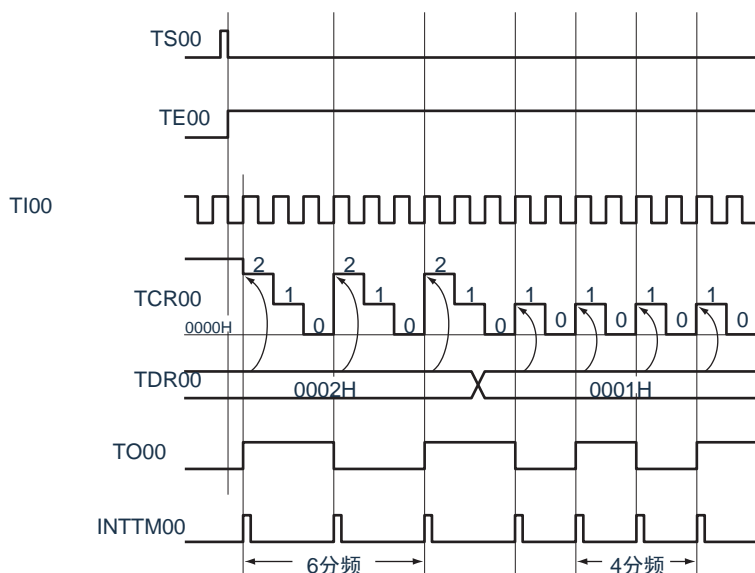
如果选择 TI00 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

$$TO00输出的时钟周期=理想的TO00输出时钟周期±运行时钟周期（误差）$$

能随时改写TDR00寄存器，改写的TDR00寄存器的值在下一个计数期间有效。

图6-49：作为分频器运行的基本时序例子(MD000=1)



备注：TS00：定时器通道开始寄存器0(TS0)的 bit 0

TE00：定时器通道允许状态寄存器0(TE0)的 bit 0

TI00：TI00引脚输入信号

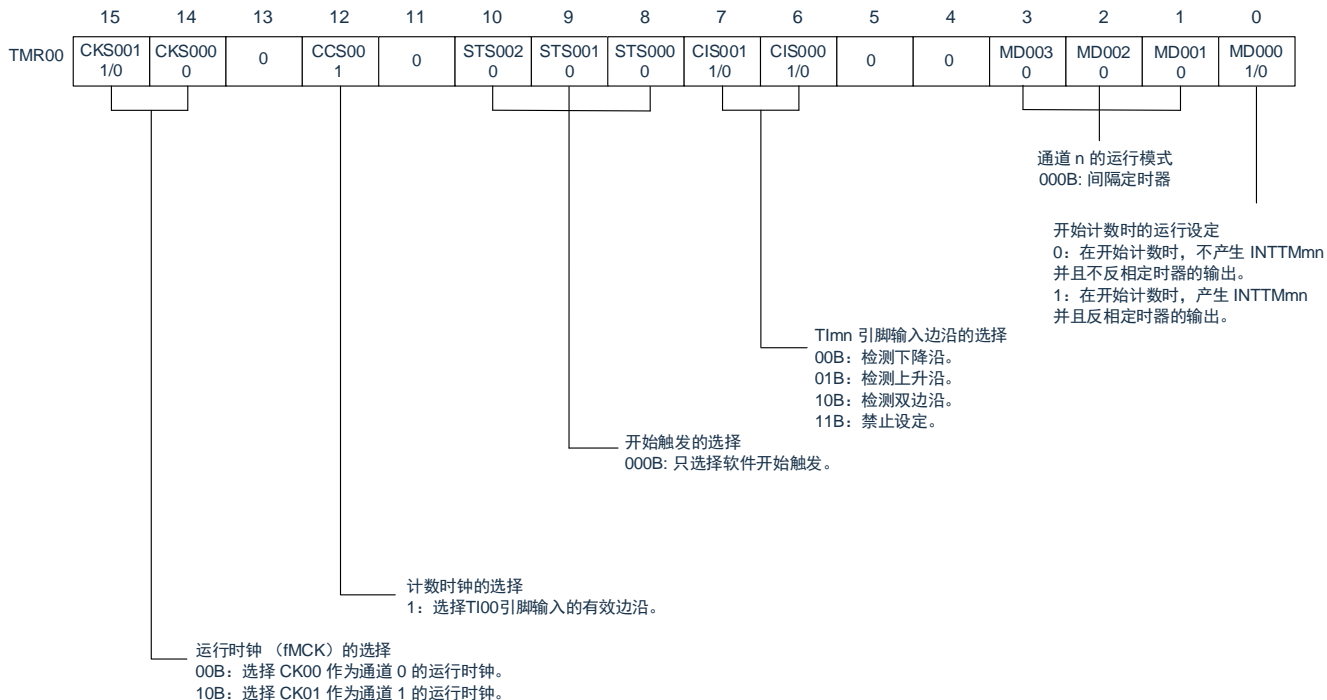
TCR00：定时器计数寄存器00(TCR00)

TDR00：定时器数据寄存器00(TDR00)

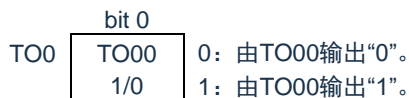
TO00：TO00引脚输出信号

图6-50: 作为分频器运行时的寄存器设置内容例子

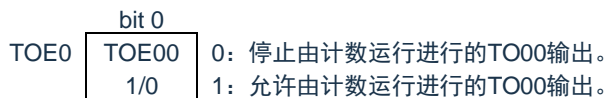
(a) 定时器模式寄存器00(TMR00)



(b) 定时器输出寄存器0(TO0)



(c) 定时器输出允许寄存器0(TOE0)



(d) 定时器输出电平寄存器0(TOLO)



(e) 定时器输出模式寄存器0(TOM0)



图6-51：分频器功能时的操作步骤

	软件操作	硬件状态
Timer4 初始设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
通道初始 设定	设定定时器时钟选择寄存器 0 (TPS0)。确定 CK00 ~ CK03 的时钟频率。	
	将噪声滤波器允许寄存器1(NFEN1)的 对应位置 “0” (OFF) 或者 “1” (ON)。 设定定时器模式寄存器 00 (TMR00) (确定通道的运行模式, 选择检测边沿)。 给定定时器数据寄存器 00 (TDR00) 设定间隔 (周期)值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始运行	将定时器输出模式寄存器 0 (TOM0) 的 TOM00 位置 “0” (主控通道输出模式)。 将 TOL00 位置 “0”。	TO00 引脚处于 Hi-Z 输出状态。
	设定 TO00 位并且确定 TO00 输出的初始电平 将 TOE00 位置 “1”, 允许 TO00 输出。 将端口寄存器和端口模式寄存器置 “0”	当端口模式寄存器为输出模式并且端口寄存器为 “0” 时, 输出 TO00 初始设定的电平。 因为通道处于运行停止状态, 所以TO00不变。 TO00 引脚输出 TO00 设定的电平。
运行中	将 TOE00 位置 “1” (只限于重新开始运行)。将 TS00 位置 “1”。	TE00 位变为 “1” 并且开始计数。
	因为 TS00 位是触发位, 所以自动返回到 “0”。	将 TDR00 寄存器的值装入定时器计数寄存器 00 (TCR00)。当 TMR00 寄存器的 MD000 位为 “1” 时, 产生 INTTM00 并且 TO00 进行交替输出。
停止运行	能任意更改 TDR00 寄存器的设定值。能随时读 TCR00 寄存器。 不使用 TSR00 寄存器。 能更改 TO0 寄存器和 TOE0 寄存器的设定值。 禁止更改 TMR00 寄存器、TOM00 位和 TOL00 位的设定值。	计数器 (TCR00) 进行递减计数。如果计数到 “0000H”, 就再次将 TDR00 寄存器的值装入 TCR00 寄存器并且继续计数。当检测到 TCR00 为 “0000H” 时, 产生 INTTM00 并且 TO00 进行交替输出。此后, 重复此运行。
	将 TT00 位置 “1”。	TE00 位变为 “0” 并且停止计数。 TCR00 寄存器保持计数值而停止计数。 TO00 输出不被初始化而保持状态。
Timer4 停止	因为 TT00 位是触发位, 所以自动返回到 “0”。	
	将 TOE00 位置 “0” 并且给 TO00 位设定值。	TO00 引脚输出 TO00 设定的电平。
重新 开始 运行	要保持 TO00 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 TO00 位置 “0”。	通过端口功能保持 TO00 引脚的输出电平。
	不需要保持 TO00 引脚输出电平的情况: 不需要设定。	
	将 PER0寄存器的 TM4mEN 位置 “0”。	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 (TO00 位变为 “0” 并且 TO00 引脚变为端口功能)

6.8.4 作为输入脉冲间隔测量的运行

能在TImn有效边沿捕捉计数值，测量TImn输入脉冲的间隔。在TEmn位为“1”的期间，也能将软件操作(TSmn=1)设置为捕捉触发，捕捉计数值。

脉冲间隔能用以下计算式进行计算：

$$TImn \text{ 输入脉冲间隔} = \text{计数时钟的周期} \times ((10000H \times \text{TSRmn:OVF}) + (\text{TDRmn的捕捉值} + 1))$$

注意：因为通过定时器模式寄存器mn(TMRmn)的CKSmn位选择的运行时钟对TImn引脚输入进行采样，所以产生1个运行时钟的误差。

在捕捉模式中，定时器计数寄存器mn(TCRmn)用作递增计数器。

如果将定时器通道开始寄存器m(TSm)的通道开始触发位(TSmn)置“1”，TCRmn寄存器就通过计数时钟从“0000H”开始递增计数。

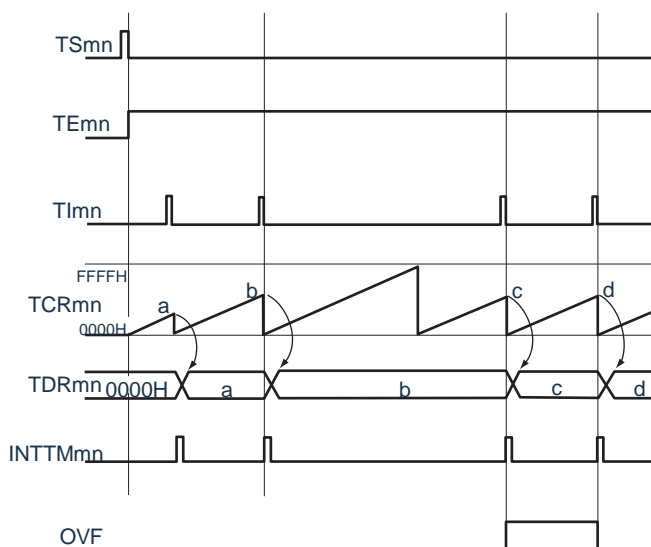
如果检测到TImn引脚输入的有效边沿，就将TCRmn寄存器的计数值传送(捕捉)到定时器数据寄存器mn(TDRmn)，同时将TCRmn寄存器清“0000H”，然后输出INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器mn(TSRmn)的OVF位置“1”。如果计数器没有发生上溢，就清除OVF位。此后，继续同样的运行。

在将计数值捕捉到TDRmn寄存器的同时，根据在测量期间是否发生上溢，更新TSRmn寄存器的OVF位，并且能确认捕捉值的上溢状态。

即使计数器进行了2个周期或者2个周期以上的完整计数，也认为发生上溢而将TSRmn寄存器的OVF位置“1”。但是，在发生2次或者2次以上的上溢时，无法通过OVF位正常测量间隔值。

将TMRmn寄存器的STSmn2~STSmn0位置“001B”，并且将TImn的有效边沿用于开始触发和捕捉触发。

图6-52：作为输入脉冲间隔测量的运行基本时序例子(MDmn0=0)

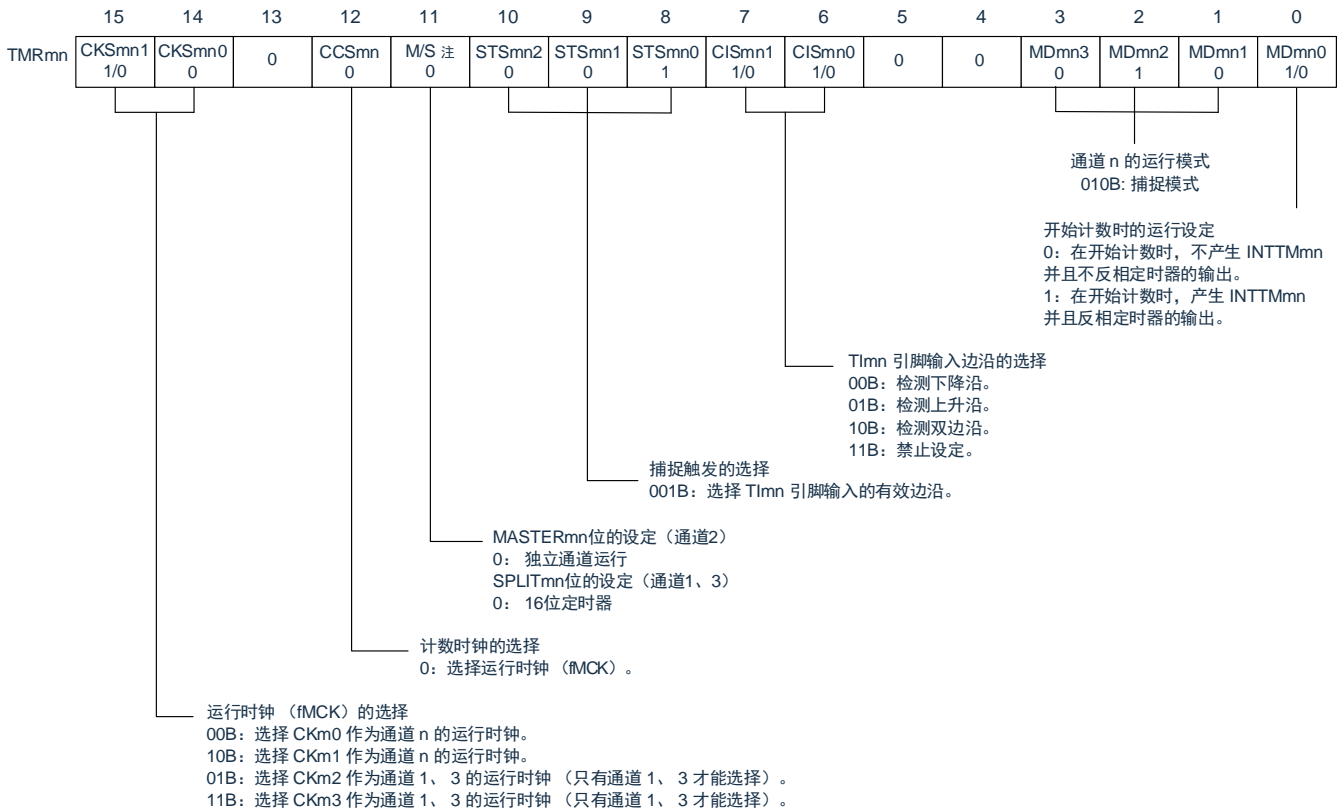


备注：

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. TSmn: 定时器通道开始寄存器m(TSm)的bit n
 TE mn: 定时器通道允许状态寄存器m(TEm)的bit n
 TImn: TImn引脚输入信号
 TCRmn: 定时器计数寄存器mn(TCRmn)
 TDRmn: 定时器数据寄存器mn(TDRmn)
 OVF: 定时器状态寄存器mn(TSRmn)的bit 0

图6-53: 测量输入脉冲间隔时的寄存器设置内容例子

(a) 定时器模式寄存器mn(TMRmn)



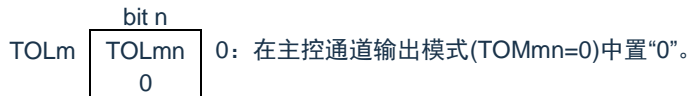
(b) 定时器输出寄存器m(TOm)



(c) 定时器输出允许寄存器m(TOEm)



(d) 定时器输出电平寄存器m(TOLm)



(e) 定时器输出模式寄存器m(TOMm)



注: TMRm2: MASTERmn位

TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-54：输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通 道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器1 (NFEN1) 的对应位 置 “0” (OFF) 或者 “1” (ON)。 设定定时器模式寄存器 mn (TMRmn) (确定通 道的运行模式)。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置 “1”。 因为 TSmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “1” 并且开始计数。 将定时器计数寄存器 mn (TCRmn) 清 “0000H”。 当 TMRmn 寄存器的 MDmn0 位为 “1” 时, 产生INTTMmn。
重新 开始 运行	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0位的设定值。 能随时读 TDRmn 寄存器。 能随时读 TCRmn 寄存器。 能随时读 TSRmn 寄存器。 禁止更改 TOMmn 位、TOLmn 位、TOmn 位和 TOEmn 位的设定值。	计数器 (TCRmn) 从 “0000H” 开始递增计 数, 如果检测到 TImn 引脚输入的有效边沿或者 将 TSmn 位置 “1”, 就将计数值传送 (捕 捉) 到定时器数据寄存器mn (TDRmn), 同时 将TCRmn寄存器清 “0000H” 并且产生 INTTMmn。 此时, 如果发生上溢, 就将定时器状态寄存器 mn (TSRmn) 的 OVF 位置位。如果没有发生 上溢, 就清除 OVF 位。此后, 重复此运行。
停止 运行	将 TTmn 位置 “1”。 因为 TTmn 位是触发位, 所以自动返回 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 保持 TSRmn 寄存器的 OVF 位。
Timer4 停止	将 PER0寄存器的 TM4mEN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

6.8.5 作为输入信号高低电平宽度测量的运行

注意：当用作LIN-bus支持功能时，必须将输入切换控制寄存器(ISC)的bit1(ISC1)置“1”，并且在以下说明中，请用RxD0代替TImn。

能通过TImn引脚输入的一个边沿开始计数并且在另一个边沿捕捉计数值，测量TImn的信号宽度(高低电平宽度)。TImn的信号宽度能用以下计算式进行计算。

$$\text{TImn输入的信号宽度} = \text{计数时钟的周期} \times ((10000\text{H} \times \text{TSRmn:OVF}) + (\text{TDRmn的捕捉值} + 1))$$

注意：因为通过定时器模式寄存器mn(TMRmn)的CKSmn位选择的运行时钟对TImn引脚输入进行采样，所以产生1个运行时钟的误差。

在捕捉&单次计数模式中，定时器计数寄存器mn(TCRmn)用作递增计数器。如果将定时器通道开始寄存器m(TSm)的通道开始触发位(TSmn)置“1”，TEmn位就变为“1”，并且进入TImn引脚的开始边沿检测等待状态。

如果检测到TImn引脚输入的开始边沿(在测量高电平宽度时为TImn引脚输入的上升沿)，就与计数时钟同步，从“0000H”开始递增计数。然后，如果检测到有效捕捉边沿(在测量高电平宽度时为TImn引脚输入的下降沿)，就在将计数值传送到定时器数据寄存器mn(TDRmn)的同时，输出INTTmn。此时，如果计数器发生上溢，就将定时器状态寄存器mn(TSRmn)的OVF位置位。如果计数器没有发生上溢，就清除OVF位。TCRmn寄存器的值变为“传送到TDRmn寄存器的值+1”而停止计数，并且进入TImn引脚的开始边沿检测等待状态。此后，继续同样的运行。

在将计数值捕捉到TDRmn寄存器的同时，根据在测量期间是否发生上溢，更新TSRmn寄存器的OVF位，并且能确认捕捉值的上溢状态。

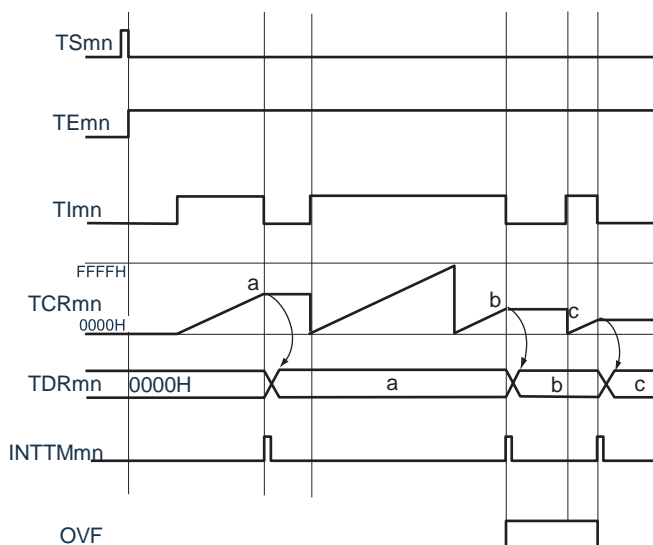
即使计数器进行了2个周期或者2个周期以上的完整计数，也认为发生上溢而将TSRmn寄存器的OVF位置“1”。但是，在发生2次或者2次以上的上溢时，无法通过OVF位正常测量间隔值。

能通过TMRmn寄存器的CISmn1位和CISmn0位来设置是测量TImn引脚的高电平宽度还是低电平宽度。此功能是以测量TImn引脚的输入信号宽度为目的，因此不能在TEmn位为“1”的期间将TSmn位置“1”。

TMRmn寄存器的CISmn1、CISmn0=10B：测量低电平宽度。

TMRmn寄存器的CISmn1、CISmn0=11B：测量高电平宽度。

图6-55: 作为输入信号高低电平宽度测量的运行基本时序例子

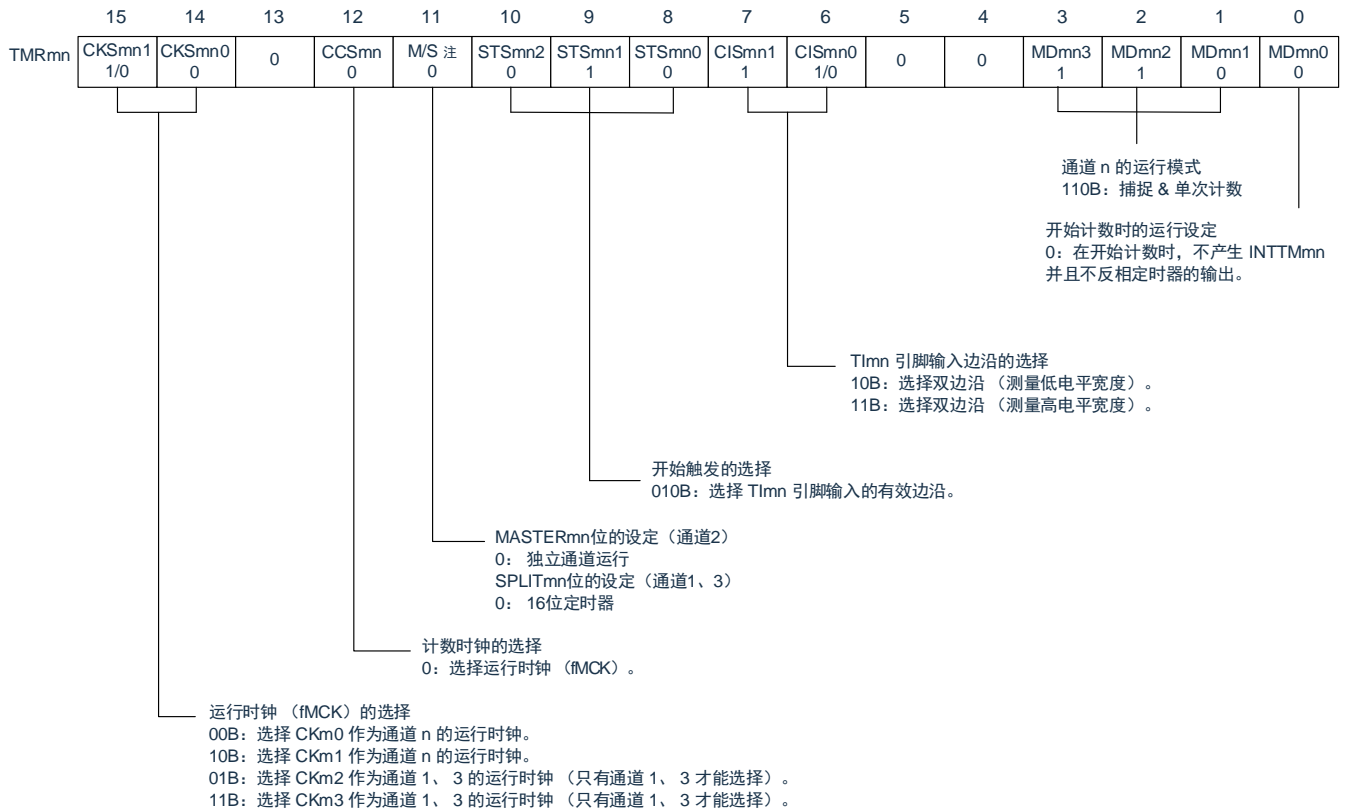


备注:

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. TSmn: 定时器通道开始寄存器 m(TSm)的bit n
TE mn: 定时器通道允许状态寄存器m(TEm)的 bit n
TI mn: TI mn引脚输入信号
TCRmn: 定时器计数寄存器mn(TCRmn)
TDRmn: 定时器数据寄存器mn(TDRmn)
OVF: 定时器状态寄存器mn(TSRmn)的bit0

图6-56：测量输入信号的高低电平宽度时的寄存器设置内容例子

(a) 定时器模式寄存器mn(TMRmn)



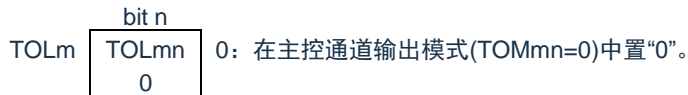
(b) 定时器输出寄存器m(TOm)



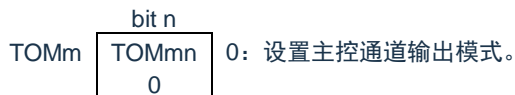
(c) 定时器输出允许寄存器m(TOEm)



(d) 定时器输出电平寄存器 m(TOLm)



(e) 定时器输出模式寄存器 m(TOMm)



注: TMRm2: MASTERmn位

TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-57: 输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0)的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通 道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器1 (NFEN1) 的对应位置 “0” (OFF) 或者 “1” (ON)。 设定定时器模式寄存器 mn (TMRmn) (确定通 道的运行模式)。 给定时器数据寄存器mn (TDRmn) 设定输出延迟 时间 将 TOEmn 位置 “0”, 并且停止 TOMn 的运 行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置 “1”。 因为 TSmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “1” 并且进入开始触发 (检测 TImn引脚输入的有效边沿或者将TSmn置 “1”) 的检测等待状态。
	检测 TImn 引脚输入的计数开始边沿。	将定时器计数寄存器 mn (TCRmn) 清 “0000H” 并 且开始递增计数。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 禁止更改 TMRmn 寄存器、TOMmn 位、TOLmn 位、TOMn 位和 TOEmn 位的设定值。	在检测到 TImn 引脚的开始边沿后, 计数器 (TCRmn) 从 “0000H” 开始递增计数。如果 检测到TImn 引脚的捕捉边沿, 就将计数值传送 到定时器数据寄存器 mn (TDRmn), 并且产 生 INTTMmn。 此时, 如果发生上溢, 就将定时器状态寄存器 mn (TSRmn) 的 OVF 位置位。如果没有发生 上溢, 就清除 OVF 位。TCRmn 寄存器在检测 到下一个 TImn引脚的开始边沿前停止计数。 此后, 重复此运行。
停止 运行	将 TTmn 位置 “1”。 因为 TTmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。保持 TSRmn 寄存器的 OVF 位。
Timer4 停止	将 PER0寄存器的 TM4mEN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

 重新
开始
运行

备注: m: 单元号(m=0) n: 通道号(n=0~3)

6.8.6 作为延迟计数器的运行

能通过TImn引脚输入的有效边沿检测(外部事件)开始递减计数, 并且以任意的设置间隔产生INTTMmn (定时器中断)。

在TEmn位为“1”的期间, 能通过软件将TSmn位置“1”, 开始递减计数, 并且以任意的设置间隔产生INTTMmn(定时器中断)。

中断产生周期能用以下计算式进行计算:

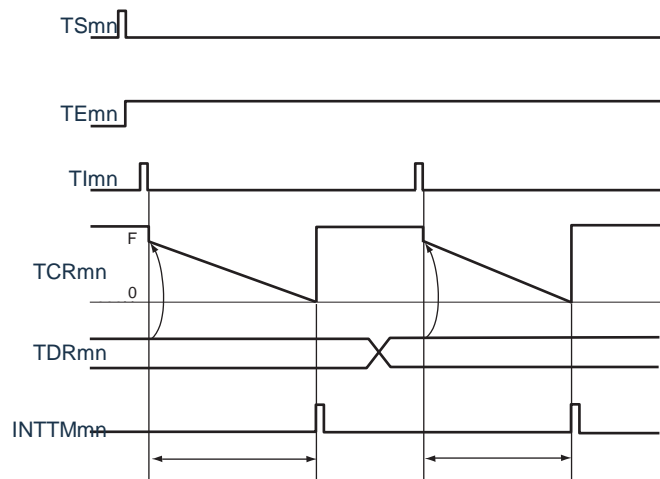
$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDRmn的设置值} + 1)$$

在单次计数模式中, 定时器计数寄存器mn(TCRmn)用作递减计数器。

如果将定时器通道开始寄存器m(TSm)的通道开始触发位(TSmn、TSHm1、TSHm3)置“1”, TEMn位、TEHm1位和TEHm3位就变为“1”, 并且进入TImn引脚的有效边沿检测等待状态。通过TImn引脚输入的有效边沿检测, 开始TCRmn寄存器的运行, 并且装入定时器数据寄存器mn(TDRmn)的值。TCRmn寄存器通过计数时钟, 从装入的TDRmn寄存器的值开始递减计数。如果TCRmn变为“0000H”, 就输出INTTMmn, 并且在检测到下一个TImn引脚输入的有效边沿前停止计数。

能随时改写TDRmn寄存器, 改写的TDRmn寄存器的值从下一个周期开始有效。

图6-58: 作为延迟计数器的运行基本时序例子

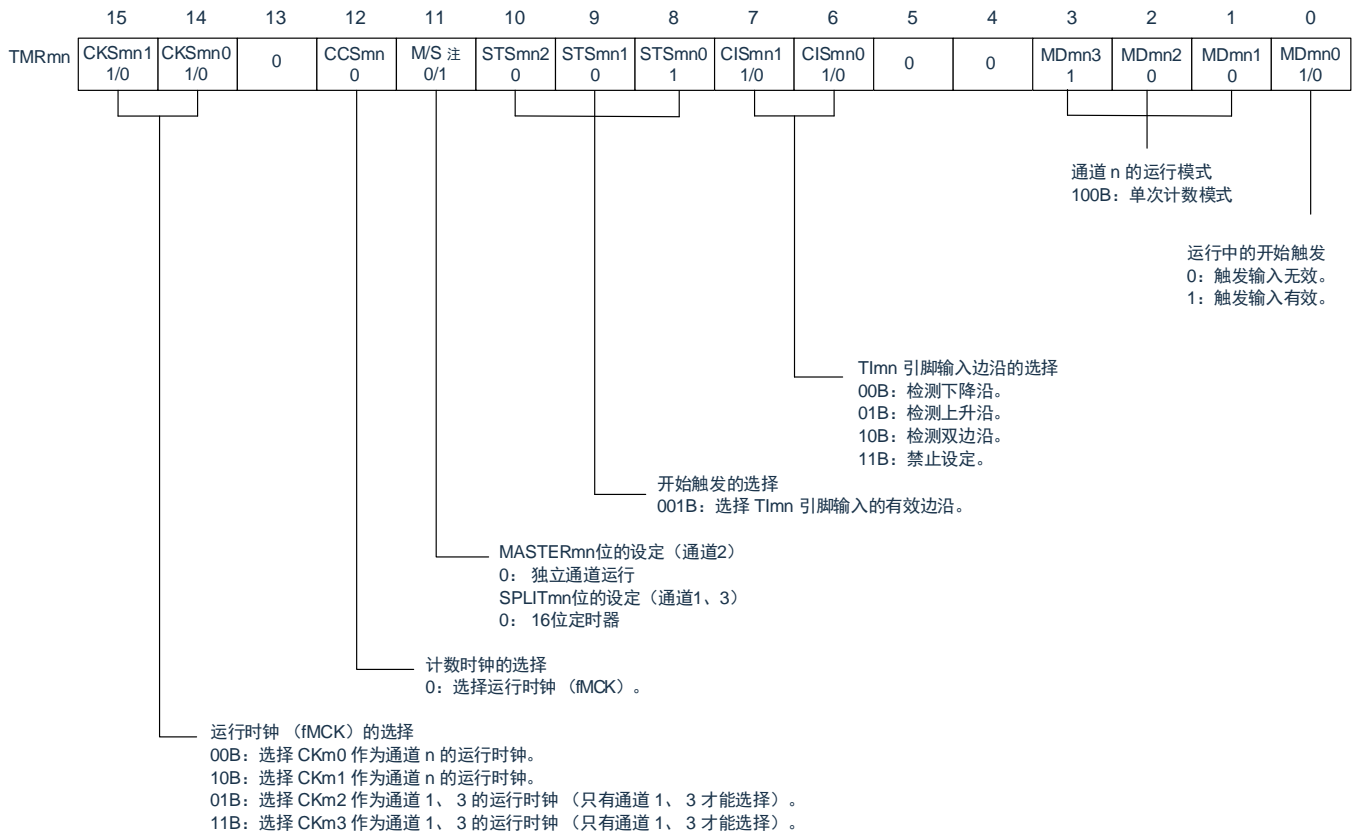


备注:

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. TSmn: 定时器通道开始寄存器m(TSm)的bit n
 TEMn: 定时器通道允许状态寄存器m(TEm)的bit n
 TImn: TImn引脚输入信号
 TCRmn: 定时器计数寄存器mn(TCRmn)
 TDRmn: 定时器数据寄存器mn(TDRmn)

图6-59: 延迟计数器功能时的寄存器设置内容例子

(a) 定时器模式寄存器mn(TMRmn)



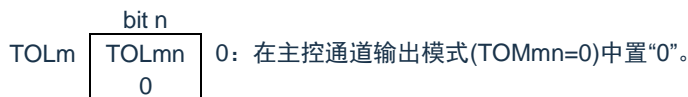
(b) 定时器输出寄存器m(TOm)



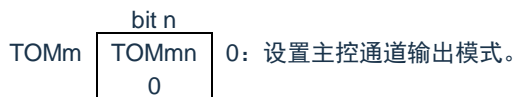
(c) 定时器输出允许寄存器m(TOEm)



(d) 定时器输出电平寄存器m(TOLm)



(e) 定时器输出模式寄存器m(TOMm)



注: TMRm2: MASTERmn位

TMRm1、TMRm3: SPLITmn位

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

图6-60: 延迟计数器功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通 道处于 运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器1 (NFEN1) 的 对应位 置 “0” (OFF) 或者 “1” (ON)。设定定 时器模式寄存器 mn (TMRmn) (确定通道 n 的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定输出延 迟 时间。 将 TOEmn 位置 “0” 并且停止 TOmn 的运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置 “1”。 因为 TSmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “1” 并且进入开始触发 (检测 TImn 引脚输入的有效边沿或者将 TSmn 位置 “1”) 的检测等 待状态。
	通过检测到下一个开始触发, 开始递减计数。 • 检测 TImn 引脚输入的有效边沿。 • 通过软件将 TSmn 位置 “1”。	将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn)。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。	计数器 (TCRmn) 进行递减计数。如果 TCRmn 计数到 “0000H”, 就产生 INTTMmn, 并且在检测到下 一次开始触发 (检测 TImn 引脚输入的有效边沿或 者将 TSmn 位置 “1”) 前 TCRmn 为 “0000H” 而停止 计数。
停止 运行	将 TTmn 位置 “1”。 运行 因为 TTmn 位是触发位, 所以自动返回到 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
Timer4 停止	将 PER0 寄存器的 TM4mEN位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注: m: 单元号(m=0) n: 通道号(n=0~3)

6.9 通用定时器单元的多通道联动运行功能

6.9.1 作为单触发脉冲输出功能的运行

将2个通道成对使用，能通过Tlmn引脚的输入生成任意延迟脉宽的单触发脉冲。延迟和脉宽能用以下计算式进行计算：

$$\begin{aligned} \text{延迟} &= \{ \text{TDRmn (主控) 的设置值} + 2 \} \times \text{计数时钟周期} \\ \text{脉宽} &= \{ \text{TDRmp (从属) 的设置值} \} \times \text{计数时钟周期} \end{aligned}$$

在单次计数模式中，主控通道运行并且对延迟进行计数。通过检测开始触发，主控通道的定时器计数寄存器mn(TCRmn)开始运行并且装入定时器数据寄存器mn(TDRmn)的值。TCRmn寄存器通过计数时钟，从装入的TDRmn寄存器的值开始递减计数。如果TCRmn变为“0000H”，就输出INTTMmn，并且在检测到下一个开始触发前停止计数。

在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的INTTMmn作为开始触发，从属通道的TCRmp寄存器开始运行并且装入TDRmp寄存器的值。TCRmp寄存器通过计数时钟，从装入的TDRmp寄存器值开始递减计数。如果计数值变为“0000H”，就输出INTTMmp，并且在检测到下一个开始触发(主控通道的INTTMmn)前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOmp的输出电平变为有效电平，如果TCRmp变为“0000H”，就变为无效电平。

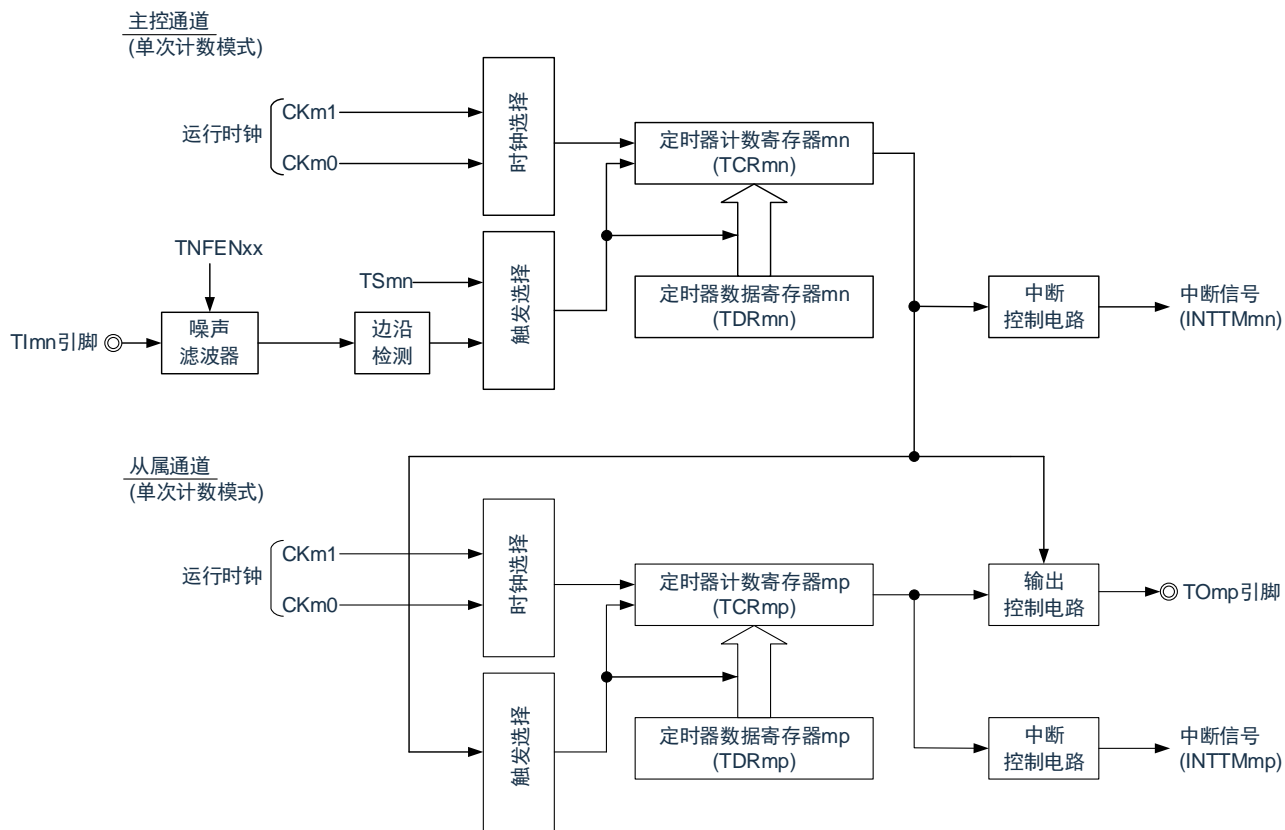
不使用Tlmn引脚输入也能将软件操作(TSmn=1)作为开始触发来输出单触发脉冲。

注意：因为主控通道的TDRmn寄存器和从属通道的TDRmp寄存器的装入时序不同，所以如果在计数过程中改写TDRmn寄存器和TDRmp寄存器，就可能与装入时序发生竞争，输出不正常的波形。必须在产生INTTMmn后改写TDRmn寄存器，并且在产生INTTMmp后改写TDRmp寄存器。

备注： m：单元号(m=0) n：主控通道号(n=0、2)

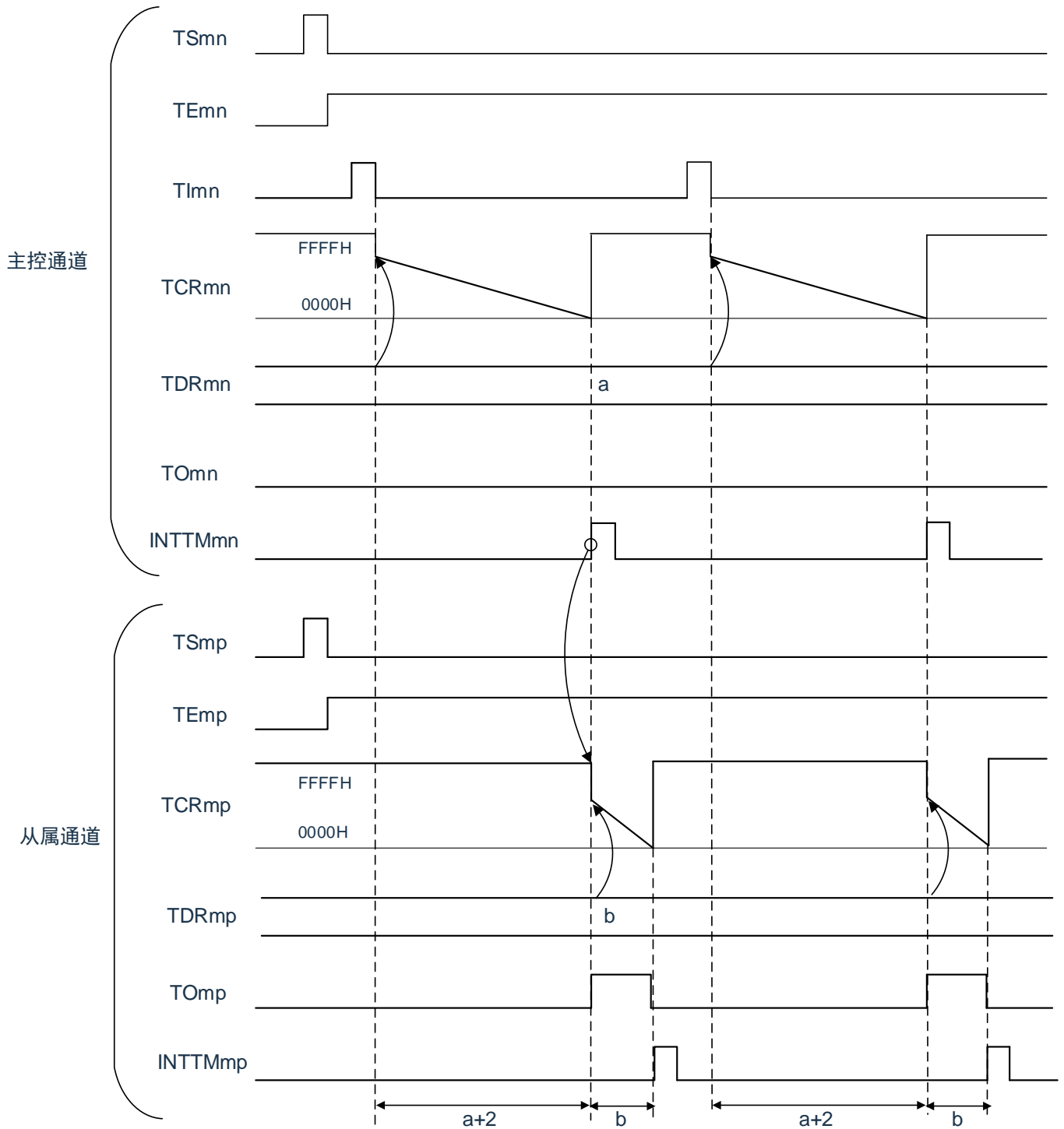
p：从属通道号(n=0：p=1、2、3，n=2：p=3)

图6-61：作为单触发脉冲输出功能运行的框图



备注：m：单元号(m=0) n：主控通道号(n=0、2)
 p：从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-62：作为单触发脉冲输出功能的运行基本时序例子

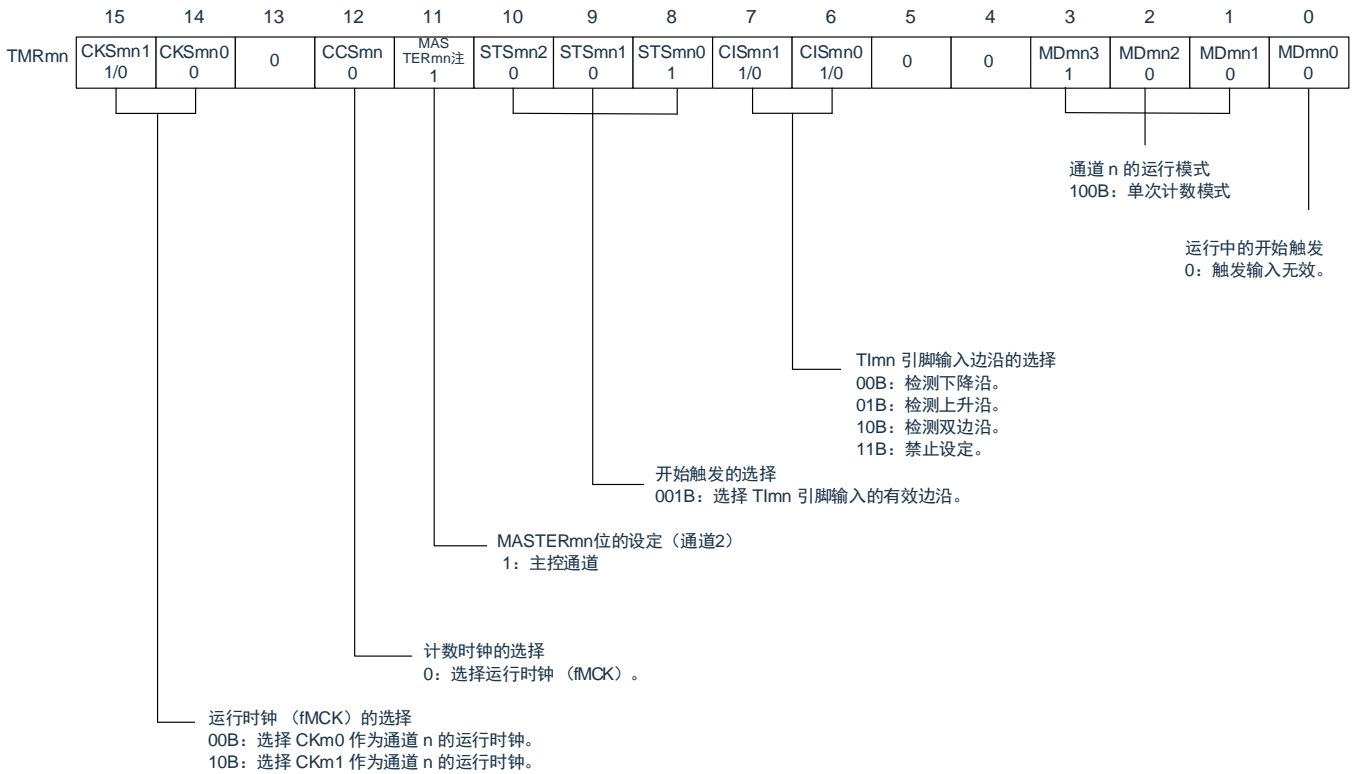


备注:

1. m: 单元号(m=0)n: 主控通道号(n=0、2) p: 从属通道号(n=0: p=1、2、3, n=2: p=3)
2. TSmn、TSmp: 定时器通道开始寄存器m(TSm)的bit n、p
 TEmn、TEmp: 定时器通道允许状态寄存器m(TEm)的bit n、p
 TImn、TImp: TImn引脚和TImp引脚的输入信号
 TCRmn、TCRmp: 定时器计数寄存器mn、mp(TCRmn、TCRmp)
 TDRmn、TDRmp: 定时器数据寄存器mn、mp(TDRmn、TDRmp)
 TOmn、TOmp: TOmn引脚和TOmp引脚的输出信号

图6-63：单触发脉冲输出功能时(主控通道)的寄存器设置内容例子

(a) 定时器模式寄存器 mn(TMRmn)



(b) 定时器输出寄存器 m(TOm)

bit n
TOm

TOmn
0

 0: 由 TOmn 输出“0”。

(c) 定时器输出允许寄存器 m(TOEm)

bit n
TOEm

TOEmn
0

 0: 停止由计数运行进行的 TOmn 输出。

(d) 定时器输出电平寄存器 m(TOLm)

bit n
TOLm

TOLmn
0

 0: 在 TOMmn=0(主控通道输出模式)时置“0”。

(e) 定时器输出模式寄存器 m(TOMm)

bit n
TOMm

TOMmn
0

 0: 设置主控通道输出模式。

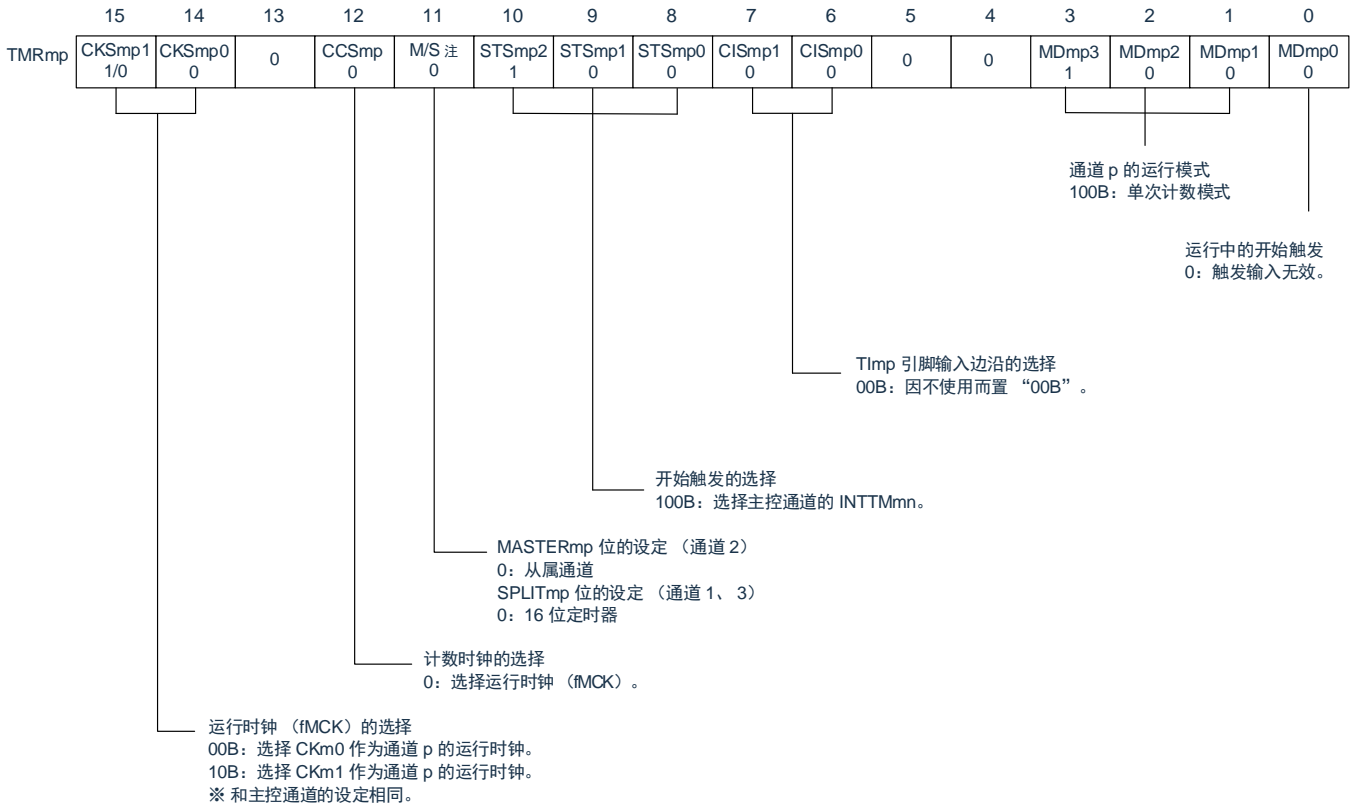
注：TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

备注：m: 单元号(m=0) n: 主控通道号(n=0、2)

图6-64：单触发脉冲输出功能时(从属通道)的寄存器设置内容例子

(a) 定时器模式寄存器 mp(TMRmp)



(b) 定时器输出寄存器 m(TOm)

	bit p	
TOm	TOmp 1/0	0: 由TOmp输出“0”。 1: 由TOmp输出“1”。

(c) 定时器输出允许寄存器 m(TOEm)

	bit p	
TOEm	TOEmp 1/0	0: 停止由计数运行进行的TOmp输出。 1: 允许由计数运行进行的TOmp输出。

(d) 定时器输出电平寄存器 m(TOLm)

	bit p	
TOLm	TOLmp 1/0	0: 正逻辑输出(高电平有效) 1: 负逻辑输出(低电平有效)

(e) 定时器输出模式寄存器 m(TOMm)

	bit p	
TOMm	TOMmp 1	1: 设置从属通道输出模式。

注：TMRm2: MASTERmp位

TMRm1、TMRm3: SPLITmp位

备注：m: 单元号(m=0) n: 主控通道号 (n=0、2)

p: 从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-65: 单触发脉冲输出功能时的操作步骤 (1/2)

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通道处于 运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置 “1”。 设定使用的 2 个通道的定时器模式寄存器 mn、 mp (TMRmn、TMRmp) (确定通道的运行模 式)。 给主控通道的定时器数据寄存器 mn (TDRmn) 设定输出延迟时间, 并且给从属通道的 TDRmp 寄存器设定脉宽。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp位置 “1” (从属通道输出模式)。 设定 TOLmp 位。 设定 TOmp 位并且确定 TOmp 输出的初始电平。 将 TOEmp 位置 “1”, 允许 TOmp 输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmp 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 “0” 时, 输出 TOmp 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 不变。 TOmp 引脚输出 TOmp 设定的电平。

图6-66: 单触发脉冲输出功能时的操作步骤 (2/2)

重新开始运行	开始运行	将TOEmp位（从属）置“1”（只限于重新开始运行） 将定时器通道开始寄存器m（TSm）的TSmn（主控）和TSmp（从属）位同时置“1”。 → 因为 TTmn 位和 TTmp 位是触发位，所以自动返回到“0”	TEmn 位和 TEm p 位都变为“1”，主控通道进入开始触发（检测TImn引脚输入的有效边沿或者将主控通道的TSmn位置“1”）的检测等待状态。计数器还处于停止状态。
		通过检测主控通道的开始触发，开始主控通道的计数。 检测TImn引脚输入的有效边沿。 通过软件将主控通道的TSmn位置“1”注。	主控通道开始计数。
	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0位的设定值。 禁止更改 TMRmp、TDRmn、TDRmp 寄存器以及TOMmn 位、TOMmp 位、TOLmn 位和TOLmp 位的设定值。 能随时读 TCRmn 寄存器和 TCRmp 寄存器。 不使用 TSRmn 寄存器和 TSRmp 寄存器。 能更改从属通道的 TOm 寄存器和 TOEm 寄存器的设定值。	主控通道通过检测开始触发（检测 TImn 引脚输入的有效边沿或者将主控通道的 TSmn 位置“1”），将TDRmn 寄存器的值装入定时器计数寄存器 mn（TCRmn），并且进行递减计数。如果 TCRmn 计数到“0000H”，就产生 INTTMmn，并且在下一次TImn 引脚输入前停止计数。 从属通道以主控通道的INTTMmn为触发，将TDRmp寄存器的值装入 TCRmp 寄存器并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1个计数时钟后，将TOmp的输出电平置为有效电平。 然后，如果 TCRmp 计数到“0000H”，就在将 TOmp 的输出电平置为无效电平后停止计数。 此后，重复此运行。	
	将 TTmn 位（主控）和 TTmp 位（从属）同时置“1”。 → 因为 TTmn 位和 TTmp 位是触发位，所以自动返回到“0”。	TEmn 位和 TEm p 位都变为“0”并且停止计数。 TCRmn 寄存器和 TCRmp 寄存器保持计数值而停止计数。TOmp 输出不被初始化而保持状态。	
	将从属通道的 TOEmp 位置“0”并且给 TOmp 位设定值。 →	TOmp 引脚输出 TOmp 设定的电平。	
要保持 TOmp 引脚输出电平的情况： → 在给端口寄存器设定要保持的值后将 TOmp 位置“0”。 不需要保持 TOmp 引脚输出电平的情况：不需要设定。 将 PER0寄存器的 TM4mEN 位置“0”。 →	通过端口功能保持 TOmp 引脚的输出电平。 定时器单元 m 的输入时钟处于停止提供状态。对全部电路和各通道的 SFR 进行初始化。		
Timer4 停止			

注：不能将从属通道的TSmn位置“1”。

备注：m：单元号（m=0） n：主控通道号（n=0、2）

p：从属通道号（n=0： p=1、2、3， n=2： p=3）

6.9.2 作为PWM功能的运行

将2个通道成对使用，能生成任意周期和占空比的脉冲。输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{TDRmn \text{ (主控) 的设置值} + 1\} \times \text{计数时钟周期} \\ \text{占空比}[\%] &= \{TDRmp \text{ (从属) 的设置值} / \{TDRmn \text{ (主控) 的设置值} + 1\} \} \times 100 \\ \text{0\%输出} &: TDRmp \text{ (从属) 的设置值} = 0000H \\ \text{100\%输出} &: TDRmp \text{ (从属) 的设置值} \geq \{TDRmn \text{ (主控) 的设置值} + 1\} \end{aligned}$$

备注：当TDRmp(从属)的设置值 > {TDRmn(主控)的设置值+1}时，占空比超过100%，但是为100%输出。

主控通道用作间隔定时器模式。如果将定时器通道开始寄存器m(TSm)的通道开始触发位(TSmn)置“1”，就输出中断(INTTMmn)，然后将定时器数据寄存器mn(TDRmn)的设置值装入定时器计数寄存器mn(TCRmn)，并且通过计数时钟进行递减计数。当计数到“0000H”时，在输出INTTMmn后再次将TDRmn寄存器的值装入TCRmn寄存器，并且进行递减计数。此后，在将定时器通道停止寄存器m(TTm)的通道停止触发位(TTmn)置“1”前，重复此运行。

当用作PWM功能时，主控通道进行递减计数，在计数到“0000H”为止的期间为PWM输出(TOmp)周期。从属通道用作单次计数模式。以主控通道的INTTMmn为开始触发，将TDRmp寄存器的值装入TCRmp寄存器，并且进行递减计数，计数到“0000H”为止。当计数到“0000H”时，输出INTTMmp，并且等待下一个开始触发(主控通道的INTTMmn)。

当用作PWM功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为PWM输出(TOmp)的占空比。

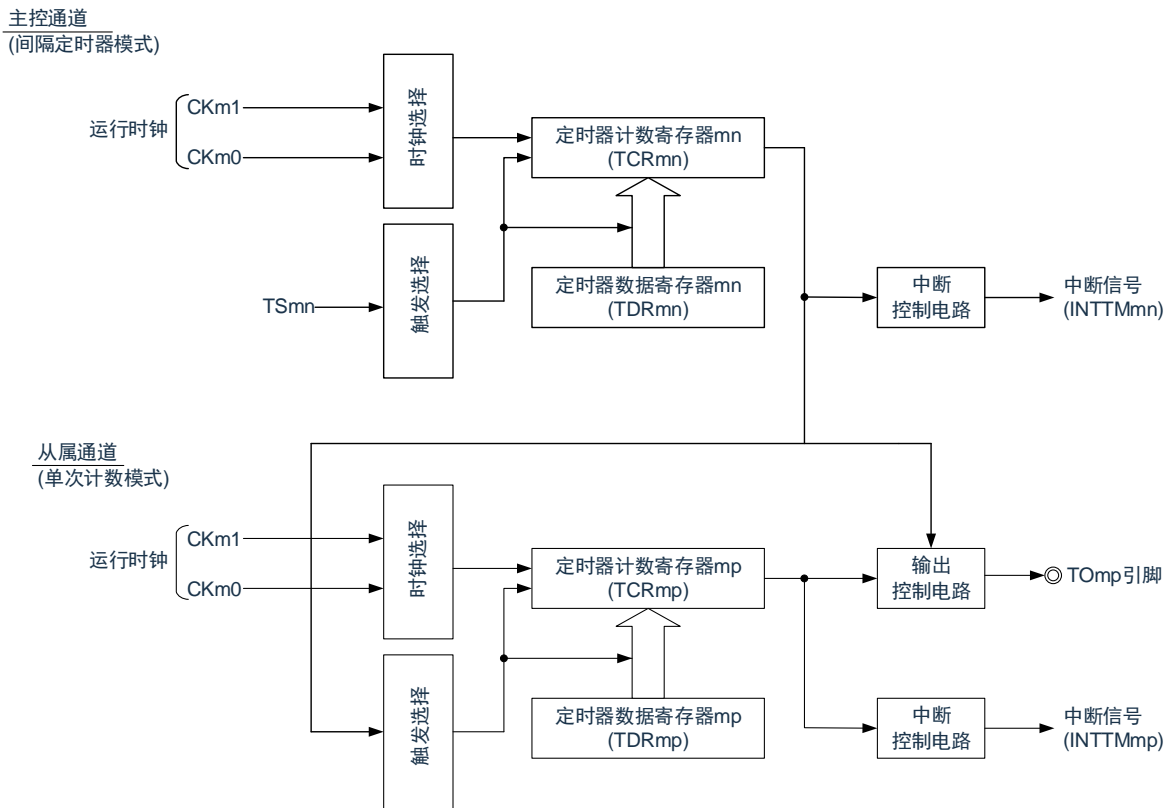
在从主控通道产生INTTMmn并且经过1个时钟后，PWM输出(TOmp)变为有效电平，并且在从属通道的TCRmp寄存器的值为“0000H”时变为无效电平。

注意：要同时改写主控通道的定时器数据寄存器mn(TDRmn)和从属通道的TDRmp寄存器时，需要2次写存取。因为在主控通道产生INTTMmn时将TDRmn寄存器和TDRmp寄存器的值装入TCRmn寄存器和TCRmp寄存器，所以如果分别在主控通道产生INTTMmn前后进行改写，TOmp引脚就不能输出期待的波形。因此，要同时改写主控的TDRmn寄存器和从属的TDRmp寄存器时，必须在主控通道产生INTTMmn后立即改写这2个寄存器。

备注：m：单元号(m=0)n：主控通道号(n=0、2)

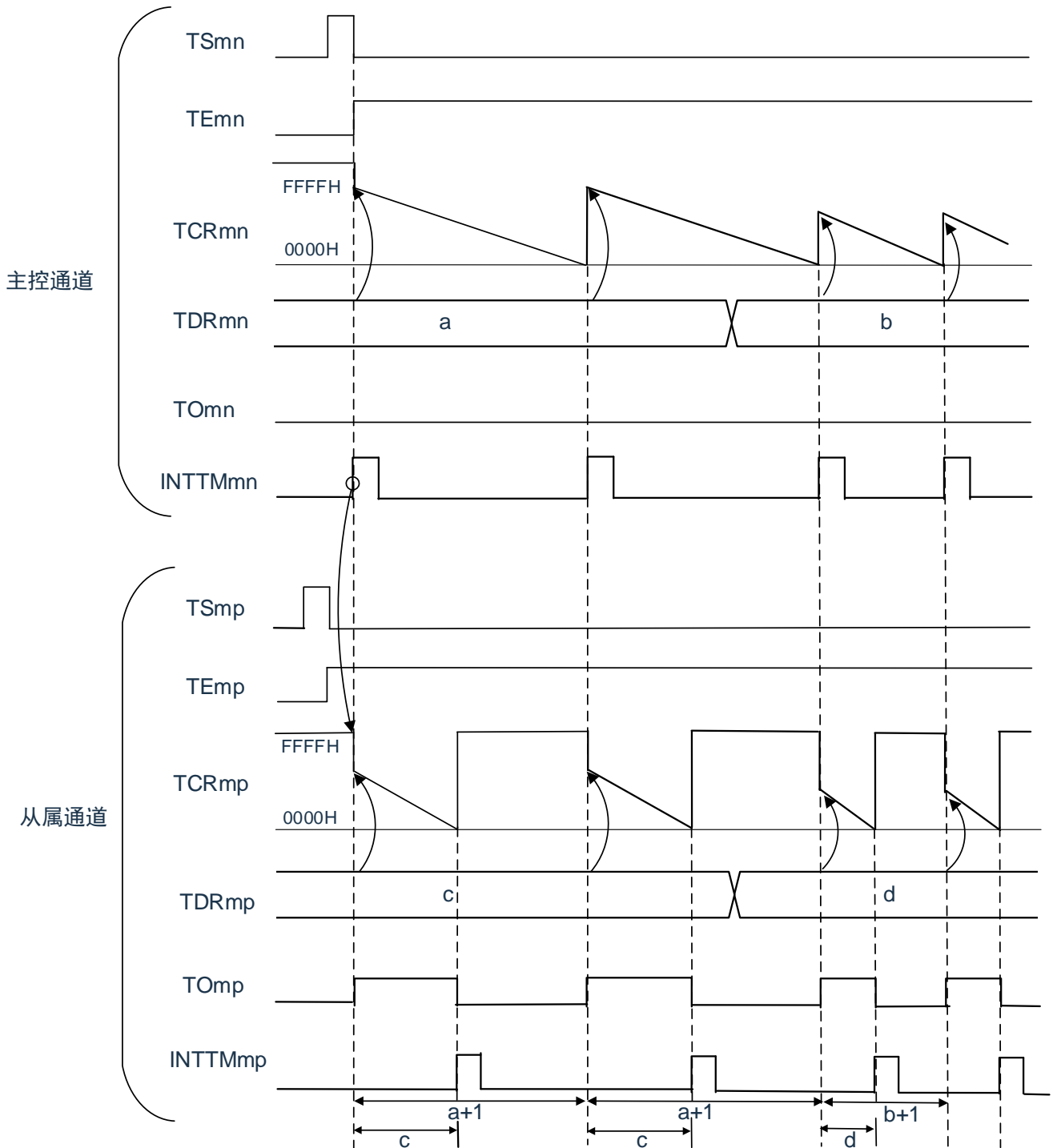
p：从属通道号(n=0：p=1、2、3，n=2：p=3)

图6-67: 作为PWM功能运行的框图



备注: m: 单元号(m=0)n: 主控通道号(n=0、2)
 p: 从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-68: 作为PWM功能的运行基本时序例子

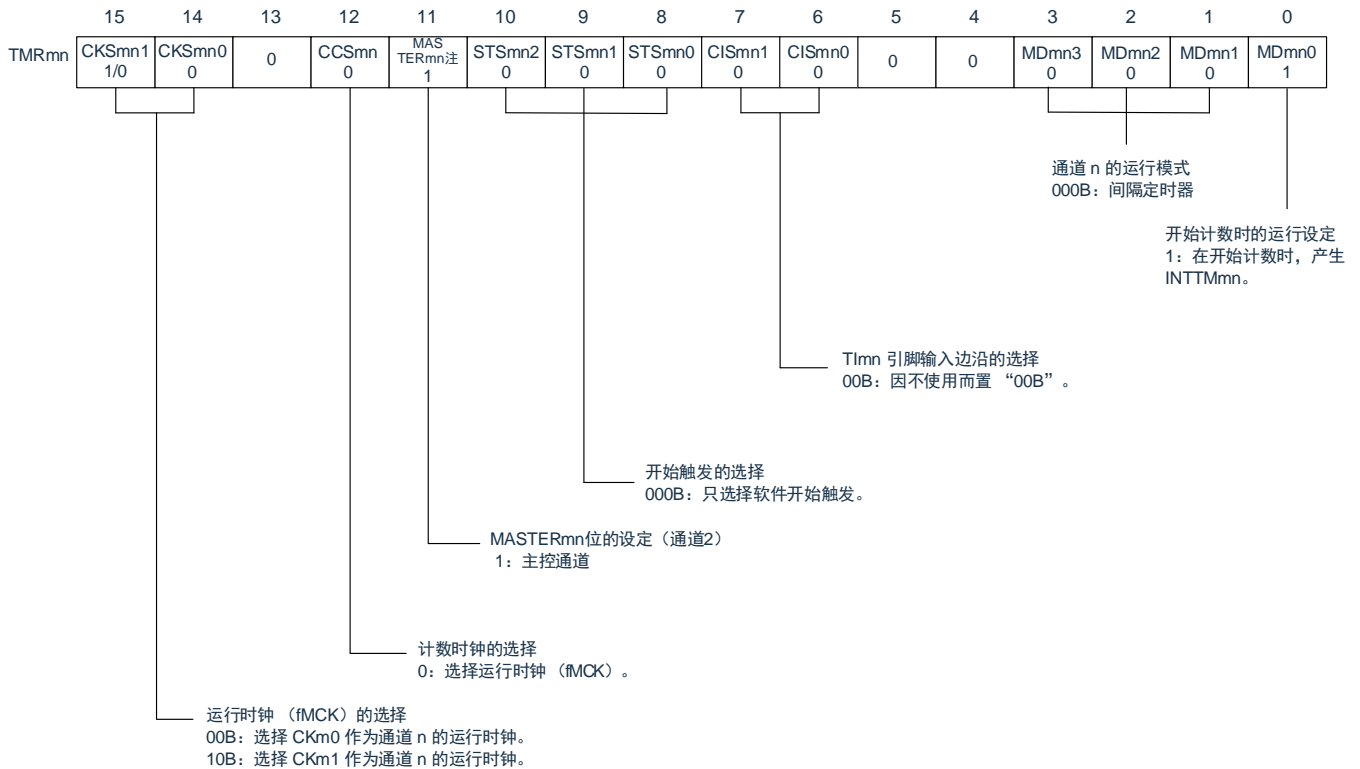


备注:

1. m: 单元号(m=0)n: 主控通道号(n=0、2) p: 从属通道号(n=0: p=1、2、3, n=2: p=3)
2. TSmn、TSmp: 定时器通道开始寄存器m(TSm)的bitn、p
TEmn、TEmp: 定时器通道允许状态寄存器m(TEm)的bitn、p
TCRmn、TCRmp: 定时器计数寄存器mn、mp(TCRmn、TCRmp)
TDRmn、TDRmp: 定时器数据寄存器mn、mp(TDRmn、TDRmp)
TOMn、TOMP: TOMn引脚和TOMP引脚的输出信号

图6-69: PWM功能时(主控通道)的寄存器设置内容例子

(a) 定时器模式寄存器 mn(TMRmn)



(b) 定时器输出寄存器 m(TOm)

bit n
TOm

TOmn
0

 0: 由TOmn输出“0”。

(c) 定时器输出允许寄存器 m(TOEm)

bit n
TOEm

TOEmn
0

 0: 停止由计数运行进行的TOmn输出。

(d) 定时器输出电平寄存器 m(TOLm)

bit n
TOLm

TOLmn
0

 0: 在TOMmn=0 (主控通道输出模式)时置“0”。

(e) 定时器输出模式寄存器 m(TOMm)

bit n
TOMm

TOMmn
0

 0: 设置主控通道输出模式。

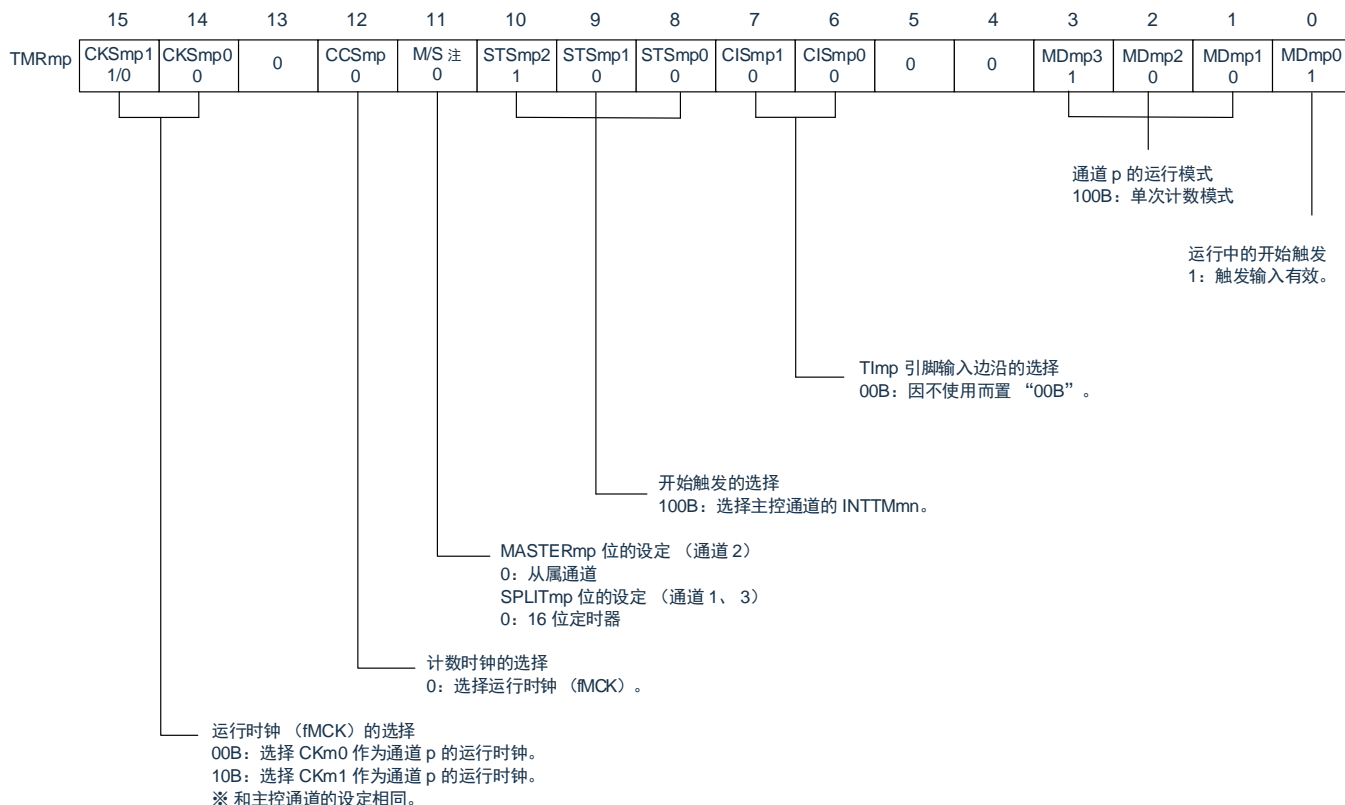
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

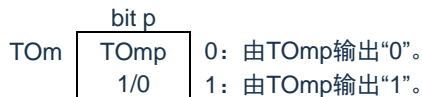
备注: m: 单元号(m=0) n: 主控通道号(n=0、2)

图6-70: PWM功能时(从属通道)的寄存器设置内容例子

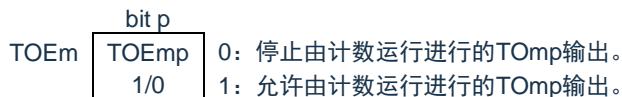
(a) 定时器模式寄存器 mp(TMRmp)



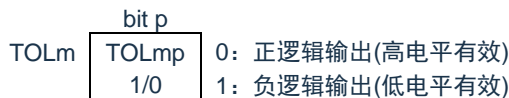
(b) 定时器输出寄存器 m(TOm)



(c) 定时器输出允许寄存器 m(TOEm)



(d) 定时器输出电平寄存器 m(TOLm)



(e) 定时器输出模式寄存器 m(TOMm)



注: TMRm2: MASTERmp位

TMRm1、TMRm3: SPLITmp位

备注: m: 单元号(m=0) n: 主控通道号(n=0、2)

p: 从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-71: PWM功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通道处于 运行停止状态。 (开始提供时钟, 能写各寄存器)
通道初 始设定	设定定时器时钟选择寄存器 m (TPSm)。确定 CKm0 ~ CKm3 的时钟频率。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp 位置 “1” (从属通道输出模式)。 设定 TOLmp 位。 设定 TOmp 位并且确定 TOmp 输出的初始电平。 将 TOEmp 位置 “1”, 允许 TOmp 输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmp 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 “0” 时, 输出 TOmp 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 不变。 TOmp 引脚输出 TOmp 设定的电平。
开始 运行	将 TOEmp 位 (从属) 置 “1” (只限于重新开 始运行)。 将定时器通道开始寄存器 m (TSm) 的 TSmn 位 (主控) 和 TSmp 位 (从属) 同时置 “1”。 因为 TSmn 位和 TSmp 位是触发位, 所以自动返 回到 “0”。	TEmn 位和 TEmp 位都变为 “1”。主控通道开始计数 并且产生 INTTMmn。以此为触发, 从属通道也开始计 数。
运行中	禁止更改 TMRmn 寄存器和 TMRmp 寄存器以及 TOMmn 位、TOMmp 位、TOLmn 位和 TOLmp 位的设定值。 能在主控通道产生 INTTMmn 后更改 TDRmn 寄 存器和 TDRmp 寄存器的设定值。 能随时读 TCRmn 寄存器和 TCRmp 寄存器。 不使用 TSRmn 寄存器和 TSRmp 寄存器。	主控通道将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn), 并且进行递减计数。如果 TCRmn 计 数到 “0000H”, 就产生 INTTMmn。同时, 将 TDRmn 寄存器的值装入 TCRmn 寄存器, 并且重新开 始递减计数。 从属通道以主控通道的 INTTMmn 为触发, 将 TDRmp 寄存器的值装入 TCRmp 寄存器, 并且计数器进行递减 计数。在从主控通道输出 INTTMmn 并且经过 1 个计数 时钟后, 将 TOmp 的输出电平置为有效电平。然后, 如 果 TCRmp 计数到 “0000H”, 就在将 TOmp 的输出电 平置为无效电平后停止计数。 此后, 重复此运行。
停止 运行	将 TTmn 位 (主控) 和 TTmp 位 (从属) 同时 置 “1”。 因为 TTmn 位和 TTmp 位是触发位, 所以自动返 回到 “0”。	TEmn 位和 TEmp 位都变为 “0” 并且停止计数。 TCRmn 寄存器和 TCRmp 寄存器保持计数值而停止计 数。 TOmp 输出不被初始化而保持状态。
	将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设定值。	TOmp 引脚输出 TOmp 设定的电平。
Timer4 停止	要保持 TOmp 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 TOmp 位置 “0”。	通过端口功能保持 TOmp 引脚的输出电平。
	不需要保持 TOmp 引脚输出电平的情况: 不需要设定。 将 PER0寄存器的 TM4mEN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。对全部 电路和各通道的 SFR 进行初始化。 (TOmp 位变为 “0” 并且 TOmp 引脚变为端口功能)

重新开始运行

备注: m: 单元号(m=0)n: 主控通道号(n=0、2)p: 从属通道号(n=0: p=1、2、3, n=2: p=3)

6.9.3 作为多重PWM输出功能的运行

这是通过扩展PWM功能并且使用多个从属通道进行不同占空比的多个PWM输出的功能。

例如，当将2个从属通道成对使用时，输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{\text{TDRmn (主控) 的设置值}+1\} \times \text{计数时钟周期} \\ \text{占空比1[\%]} &= \{\text{TDRmp (从属1) 的设置值}\} / \{\text{TDRmn (主控) 的设置值}+1\} \times 100 \\ \text{占空比2[\%]} &= \{\text{TDRmq (从属2) 的设置值}\} / \{\text{TDRmn (主控) 的设置值}+1\} \times 100 \end{aligned}$$

备注：当TDRmp(从属1)的设置值 > {TDRmn(主控)的设置值+1} 或者 {TDRmq(从属2)的设置值} > {TDRmn(主控)的设置值+1} 时，占空比超过100%，但是为100%输出。

在间隔定时器模式中，主控通道的定时器计数寄存器mn(TCRmn)运行并且对周期进行计数。在单次计数模式中，从属通道1的TCRmp寄存器运行并且对占空比进行计数以及从TOmp引脚输出PWM波形。以主控通道的INTTMmn为开始触发，将定时器数据寄存器mp(TDRmp)的值装入TCRmp寄存器并且进行递减计数。如果TCRmp变为“0000H”，就输出INTTMmp，并且在输入下一个开始触发(主控通道的INTTMmn)前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOmp的输出电平变为有效电平，如果TCRmp变为“0000H”，就变为无效电平。

和从属通道1的TCRmp寄存器相同，在单次计数模式中，从属通道2的TCRmq寄存器运行并且对占空比进行计数以及从TOmq引脚输出PWM波形。以主控通道的INTTMmn为开始触发，将TDRmq寄存器的值装入TCRmq寄存器并且进行递减计数。如果TCRmq变为“0000H”，就输出INTTMmq，并且在输入下一个开始触发(主控通道的INTTMmn)前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOmq的输出电平变为有效电平，如果TCRmq变为“0000H”，就变为无效电平。

当通过如此的运行将通道0用作主控通道时，最多能同时输出3种PWM信号。

注意：要同时改写主控通道的定时器数据寄存器mn(TDRmn)和从属通道1的TDRmp寄存器时，至少需要2次写存取。因为在主控通道产生INTTMmn时将TDRmn寄存器和TDRmp寄存器的值装入TCRmn寄存器和TCRmp寄存器，所以如果分别在主控通道产生INTTMmn前和产生后进行改写，TOmp引脚就不能输出期待的波形。因此，要同时改写主控的TDRmn寄存器和从属的TDRmp寄存器时，必须在主控通道产生INTTMmn后立即改写这2个寄存器(同样也适用于从属通道2的TDRmq寄存器)。

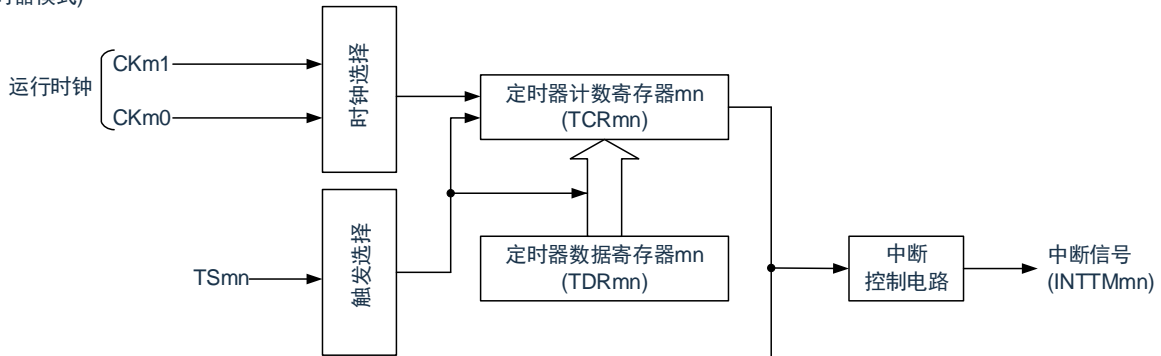
备注：m：单元号(m=0)n：主控通道号(n=0)

p：从属通道号

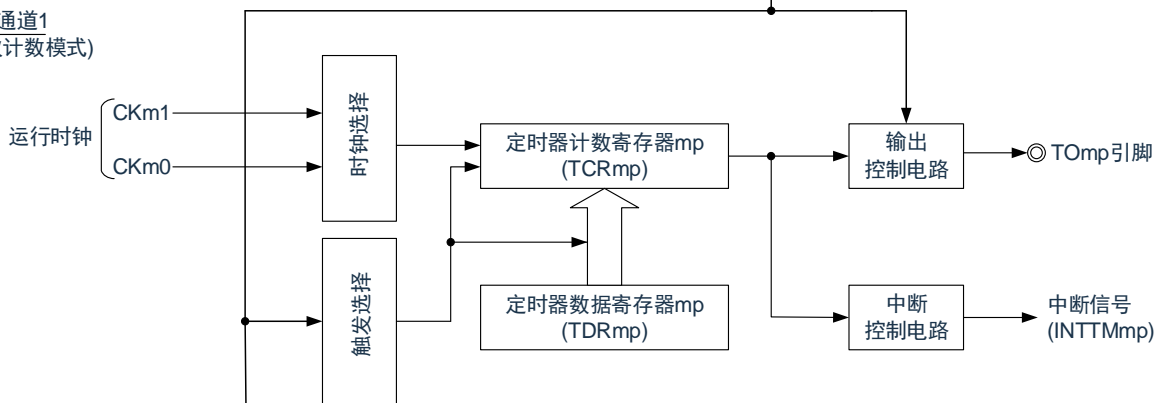
q：从属通道号n < p < q ≤ 3(p和q是大于n的整数)

图6-72：作为多重PWM输出功能运行的框图(输出2种PWM的情况)

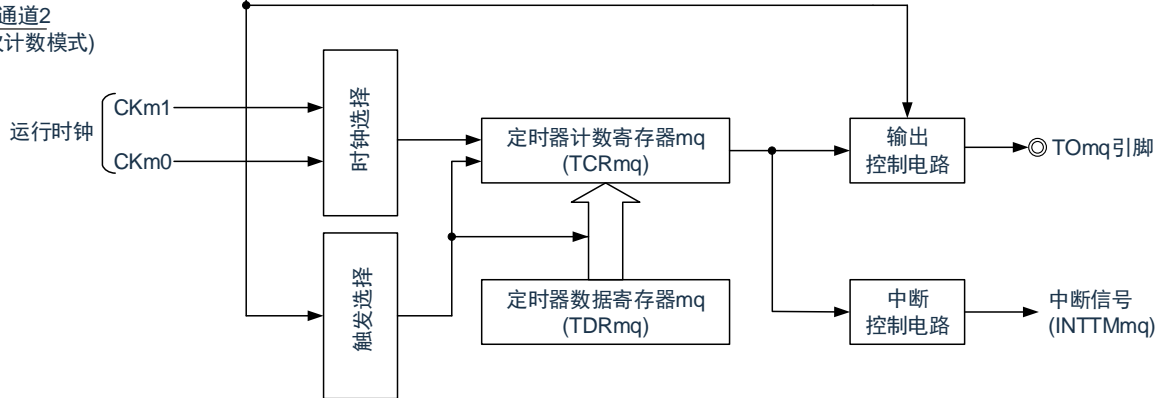
主控通道
(间隔定时器模式)



从属通道1
(单次计数模式)



从属通道2
(单次计数模式)

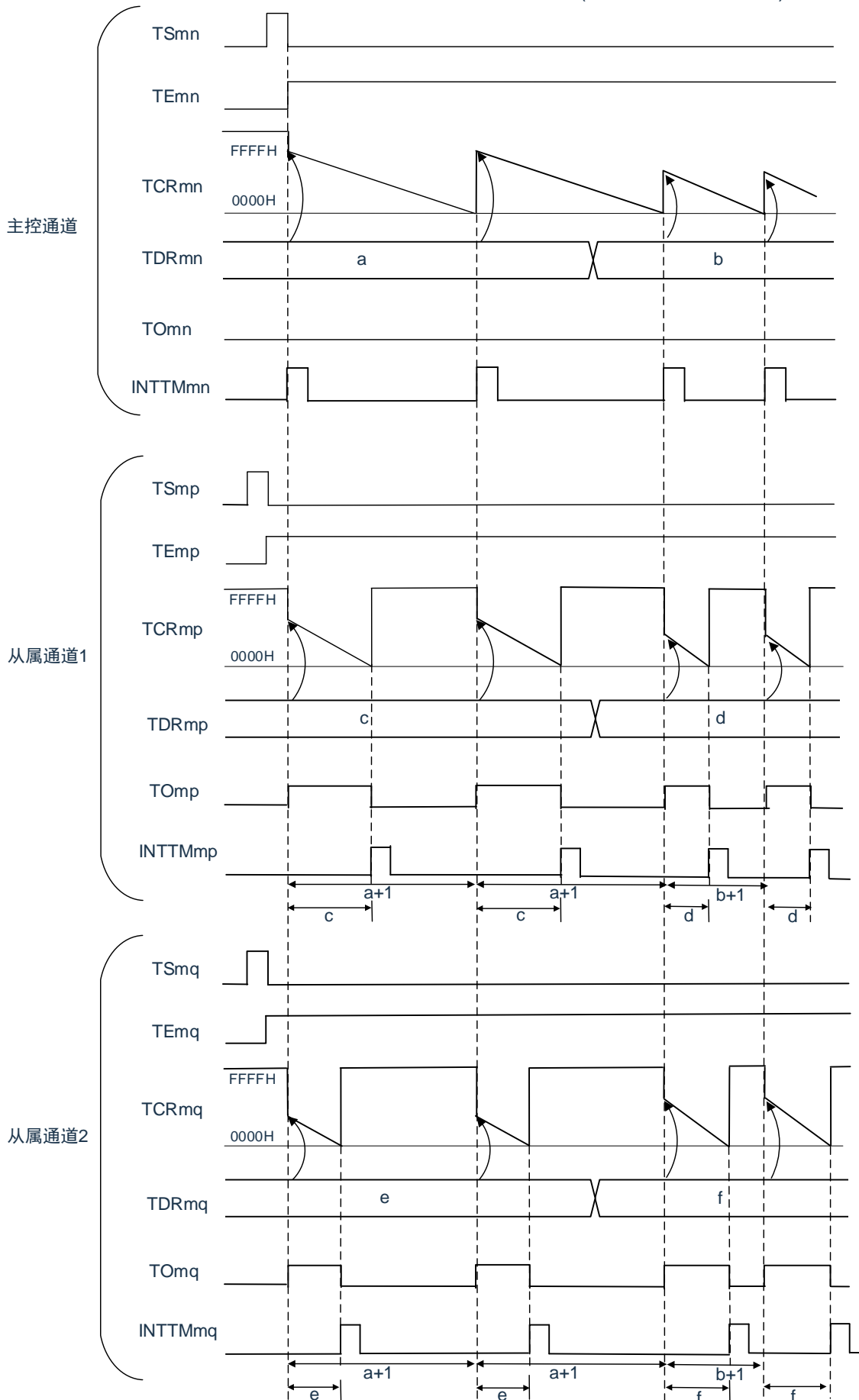


备注：m：单元号(m=0) n：主控通道号(n=0)

p：从属通道号 q：从属通道号

$n < p < q \leq 3$ (p和q是大于n的整数)

图6-73: 作为多重PWM输出功能的运行基本时序例子(输出2种PWM的情况)

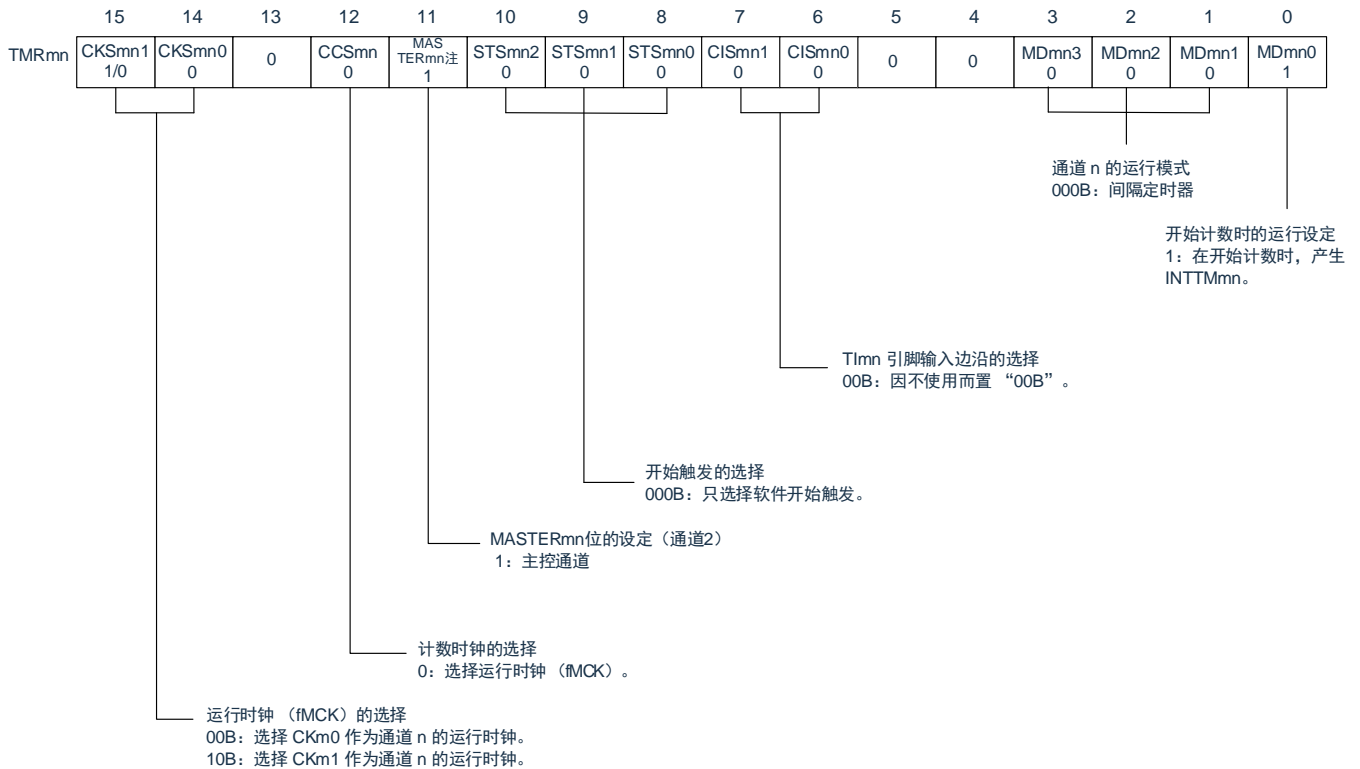


备注:

1. m: 单元号(m=0)n: 主控通道号(n=0)
p: 从属通道号q: 从属通道号
n < p < q ≤ 3 (p和q是大于n的整数)
2. TSmn、TSmp、TSmq: 定时器通道开始寄存器m(TSm)的bitn、p、q
TEmn、TEmp、TEmq: 定时器通道允许状态寄存器m(TEm)的bitn、p、q
TCRmn、TCRmp、TCRmq: 定时器计数寄存器mn、mp、mq(TCRmn、TCRmp、TCRmq)
TDRmn、TDRmp、TDRmq: 定时器数据寄存器mn、mp、mq(TDRmn、TDRmp、TDRmq)
TOmn、TOmp、TOmq: TOmn、TOmp、TOmq引脚的输出信号

图6-74：多重PWM输出功能时(主控通道)的寄存器设置内容例子

(a) 定时器模式寄存器mn(TMRmn)



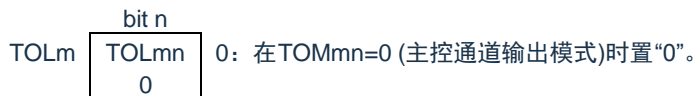
(b) 定时器输出寄存器m(TOm)



(c) 定时器输出允许寄存器m(TOEm)



(d) 定时器输出电平寄存器m(TOLm)



(e) 定时器输出模式寄存器m(TOMm)



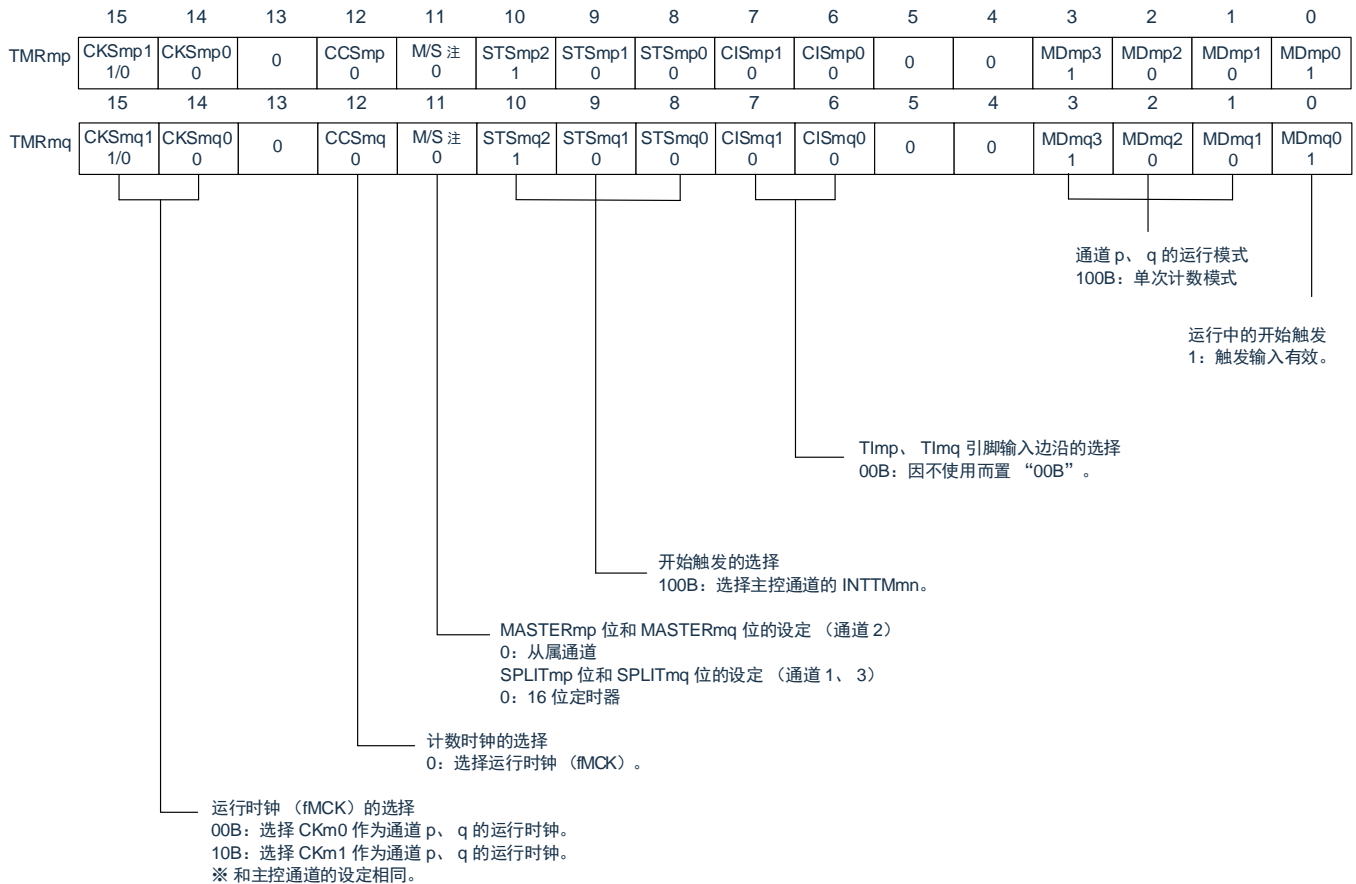
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 主控通道号(n=0)

图6-75: 多重PWM输出功能时(从属通道)的寄存器设置内容例子(输出2种PWM的情况)

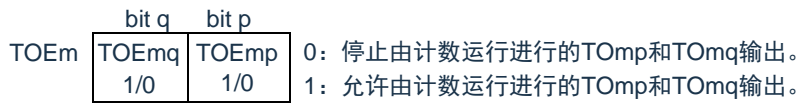
(a) 定时器模式寄存器mp、mq(TMRmp、TMRmq)



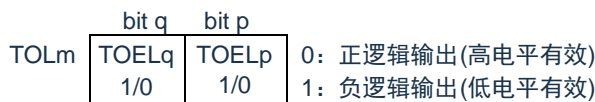
(b) 定时器输出寄存器m(TOm)



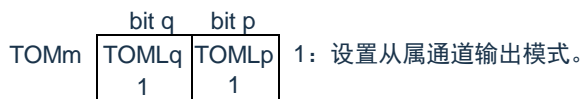
(c) 定时器输出允许寄存器m(TOEm)



(d) 定时器输出电平寄存器m(TOLm)



(e) 定时器输出模式寄存器m(TOMm)



注: TMRm2: MASTERmp位、MASTERmq位

TMRm1、TMRm3: SPLITmp位、SPLITmq位

备注: m: 单元号(m=0) n: 主控通道号(n=0)

p: 从属通道号 q: 从属通道号

n < p < q ≤ 3 (p和q是大于n的整数)

图6-76: 多重 PWM 输出功能时的操作步骤 (输出 2 种 PWM 的情况)(1/2)

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定使用的2个通道的定时器模式寄存器 mn、mp (TMRmn、TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 mn (TDRmn) 设定间隔 (周期) 值, 并且给从属通道的 TDRmp 寄存器和 TDRmq 寄存器设定占空比的值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp位和 TOMmq 位置 “1” (从属通道输出模式)。 将 TOLmp 位和 TOLmq 位置 “0”。 设定 TOmp 位和 TOmq 位, 并且确定 TOmp 和 TOmq 输出的初始电平。 将 TOEmp 位和 TOEmq 位置 “1”, 允许 TOmp 和 TOmq 的输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmp引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 “0” 时, 输出 TOmp 和 TOmq 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 和 TOmq 不变。 TOmp 引脚和 TOmq 引脚输出 TOmp 和 TOmq 设定的电平。

图6-76: 多重PWM输出功能时的操作步骤(输出2种PWM的情况)(2/2)

重新开始运行	开始运行	<p>(只在重新开始运行时将 TOEmp 位和 TOEmq 位 (从属) 置 “1”)</p> <p>将定时器通道开始寄存器 m (TSm) 的 TSmn 位 (主控)、TSmp 位和 TSmq 位 (从属) 同时置 “1”。</p> <p>因为 TSmn 位、TSmp 位和 TSmq 位是触发位, 所以自动返回到 “0”</p>	TEmn 位和 TEmq 位都变为 “1”。主控通道开始计数并且产生 INTTMmn。以此为触发, 从属通道也开始计数。
	运行中	<p>禁止更改 TMRmn、TMRmp、TMRmq 寄存器以及 TOMmn 位、TOMmp 位、TOMmq 位、TOLmn 位、TOLmp、TOLmq 位的设定值。能在主控通道产生 INTTMmn 后更改 TDRmn、TDRmp、TDRmq 寄存器的设定值。</p> <p>能随时读 TCRmn、TCRmp、TCRmq 寄存器。不使用 TSRmn、TSRmp、TSRmq 寄存器。</p>	<p>主控通道将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn), 并且进行递减计数。如果 TCRmn 计数到 “0000H”, 就产生 INTTMmn。同时, 将 TDRmn 寄存器的值装入 TCRmn 寄存器, 并且重新开始递减计数。</p> <p>从属通道 1 以主控通道的 INTTMmn 信号为触发, 将 TDRmp 寄存器的值传送到 TCRmp 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将 TOmp 的输出电平置为有效电平。然后, 如果计数到 “0000H”, 就在将 TOmp 的输出电平置为无效电平后停止计数。从属通道 2 以主控通道的 INTTMmn 信号为触发, 将 TDRmq 寄存器的值传送到 TCRmq 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将 TOMq 的输出电平置为有效电平。然后, 如果计数到 “0000H”, 就在将 TOMq 的输出电平置为无效电平后停止计数。此后, 重复此运行。</p>
	停止运行	<p>将 TTmn 位 (主控)、TTmp 位和 TTmq 位 (从属) 位同时置 “1”。</p> <p>因为 TTmn 位、TTmp 位和 TTmq 位是触发位, 所以自动返回到 “0”。</p>	<p>TEmn 位、TEmq 位和 TEmq 位都变为 “0” 并且停止计数。</p> <p>TCRmn、TCRmp、TCRmq 寄存器保持计数值而停止计数。</p> <p>TOmp 和 TOMq 输出不被初始化而保持状态。</p>
	Timer4 停止	<p>将从属通道的 TOEmp 位和 TOEmq 位置 “0” 并且给 TOmp 位和 TOMq 位设定值。</p>	TOmp 引脚和 TOMq 引脚输出 TOmp 和 TOMq 设定的电平。
		<p>要保持 TOmp 引脚和 TOMq 引脚的输出电平的情况: 在给端口寄存器设定要保留的值后将 TOmp 位和 TOMq 位置 “0”。</p> <p>不需要保持 TOmp 引脚和 TOMq 引脚的输出电平的情况: 不需要设定。</p>	通过端口功能保持 TOmp 和 TOMq 引脚的输出电平。
		将 PER0 寄存器的 TM4mEN 位置 “0”。	<p>定时器单元 m 的输入时钟处于停止提供状态。</p> <p>对全部电路和各通道的 SFR 进行初始化。</p> <p>(TOmp 位和 TOMq 位变为 “0” 并且 TOmp 引脚和 TOMq 引脚变为端口功能)</p>

备注: m: 单元号(m=0) n: 主控通道号(n=0)

p: 从属通道号 q: 从属通道号

n < p < q ≤ 3 (p和q是大于n的整数)

6.10 使用通用定时器单元时的注意事项

6.10.1 使用定时器输出时的注意事项

根据产品，分配了定时器输出功能的引脚也可能被分配其他复用功能的输出。在这种情况下使用定时器输出时，需要将其他复用功能的输出置初始值。

详细内容请参照“2.5 使用复用功能时的寄存器设置”。

第7章 定时器A

7.1 定时器A的功能

定时器A是能进行脉冲输出、外部输入的脉冲宽度和周期的测量以及对外部事件进行计数的16位定时器。

16位定时器由重加载寄存器和递减计数器构成，重加载寄存器和递减计数器分配在相同的地址。如果存取TA0寄存器，就能存取重加载寄存器和计数器。

定时器A的规格和框图分别如表7-1和图7-1所示。

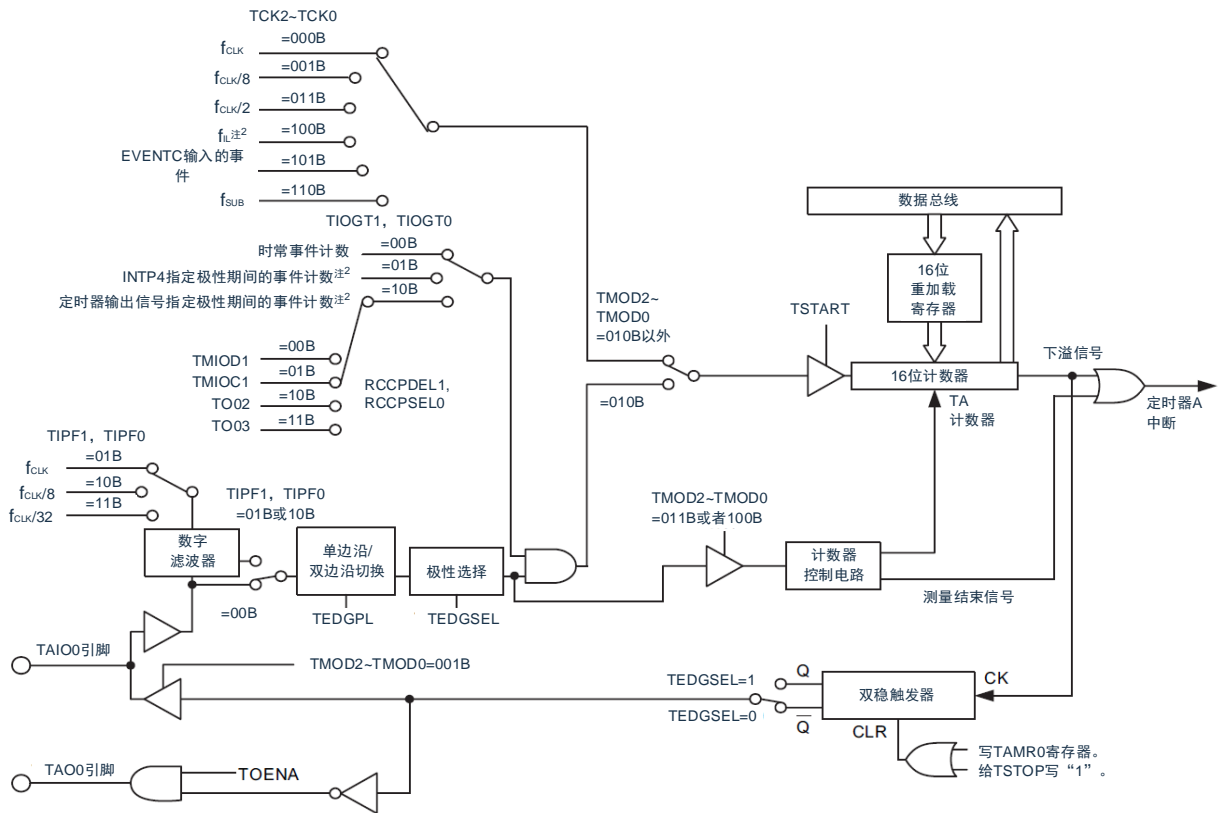
表7-1：定时器A的规格

项目		内容
运行模式	定时器模式	对计数源进行计数。
	脉冲输出模式	对计数源进行计数，在定时器发生下溢时输出极性相反的脉冲。
	事件计数器模式	对外部事件进行计数。 也能在深度睡眠模式中运行。
	脉宽测量模式	测量外部输入的脉冲宽度。
	脉冲周期测量模式	测量外部输入的脉冲周期。
计数源(运行时钟)		可选择 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/8$ 、 F_{IL} 、 F_{SUB} 或者EVENTC输入的事件。
中断		<ul style="list-style-type: none"> •当计数器发生下溢时 •在脉宽测量模式中外部输入(TAIO)的有效宽度测量结束时 •在脉冲周期测量模式中外部输入(TAIO)的设置边沿时
选择功能		•与EVENTC的协作：可选择EVENTC输入的事件作为计数源。

7.2 定时器A的结构

定时器A的框图和引脚结构分别如图7-1和表7-2所示。

图7-1：定时器A的框图



- 注1：要选择 F_{IL} 作为计数源时，必须将副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位置“1”。但是，当选择 F_{SUB} 作为实时时钟或者15位间隔定时器的计数源时，不能选择 F_{IL} 作为定时器A的计数源。
- 注2：能通过TAISR0寄存器的RCCPSEL2位选择极性。

表7-2：定时器A的引脚结构

引脚名	输入/输出	功能
INTP4	输入	定时器A的事件计数器模式控制
TAIO ^注	输入/输出	定时器A的外部事件输入和脉冲输出
TAO ^注	输出	定时器A的脉冲输出

注：能通过PIOR1寄存器的PIOR12位和PIOR13位选择TAO引脚的配置，并且能通过PIOR1寄存器的PIOR10位和PIOR11位选择TAIO引脚的配置。详细内容请参照“第2章 引脚功能”。

7.3 控制定时器A的寄存器

控制定时器A的寄存器如表7-3所示。

表7-3: 控制定时器A的寄存器

寄存器名	符号
外围I/O重定向寄存器1	PIOR1
外围允许寄存器1	PER1
副系统时钟提供模式控制寄存器	OSMC
定时器A计数寄存器0注	TA0
定时器A控制寄存器0	TACR0
定时器A/I/O控制寄存器0	TAIOC0
定时器A模式寄存器0	TAMR0
定时器A事件引脚选择寄存器0	TAISR0
端口寄存器x	Px
端口模式寄存器x	PMx

注：当存取TA0寄存器时，CPU不进入下一条指令的处理而处于CPU处理的等待状态。因此，当发生此等待时，指令执行的时钟数增加等待的时钟数。存取TA0寄存器时的读写等待时钟数都为1个时钟。

7.3.1 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器A时，必须将bit0(TMAEN)置“1”。

通过8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图7-2: 外围允许寄存器1(PER1)的格式

地址: 4002081AH 复位后: 00H RW

符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMOPEN	TMCEN	TMAEN

TMAEN	提供定时器A的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器A使用的SFR。 •定时器A处于复位状态。
1	提供输入时钟。 •能读写定时器A使用的SFR。

注意：要设置定时器A时，必须先将TMAEN位置“1”。当TMAEN位为“0”时，忽视定时器A的控制寄存器的写操作，而且读取值都为初始值(端口模式寄存器PMx和端口寄存器Px除外)。

7.3.2 副系统时钟提供模式控制寄存器(OSMC)

能通过WUTMMCK0位选择定时器A的运行时钟。

RTCLPC位是通过停止不需要的时钟功能来降低功耗的位。有关RTCLPC位的设置，请参照“第4章 时钟发生电路”。

通过8位存储器操作指令设置OSMC寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图7-3：副系统时钟提供模式控制寄存器(OSMC)的格式

地址：40020423H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	实时时钟、15位间隔定时器的运行时钟(F _{RTC})和定时器A的运行时钟的选择
0	副系统时钟(F _{SUB}) •副系统时钟为实时时钟和15位间隔定时器的运行时钟。 •不能选择低速内部振荡器作为定时器A的计数源。
1	低速内部振荡器时钟(F _{IL}) •低速内部振荡器时钟为实时时钟和15位间隔定时器的运行时钟。 •能选择低速内部振荡器或者副系统时钟作为定时器A的计数源。

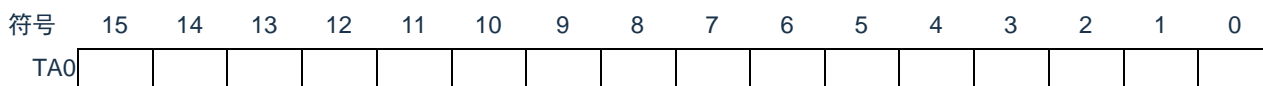
7.3.3 定时器A计数寄存器0(TA0)

这是16位寄存器。如果写此寄存器，就将数据写到重加载寄存器。如果读此寄存器，就读计数值。重加载寄存器和计数器的状态因TACR0寄存器的TSTART位的值而变。详细内容请参照“7.4.1 重加载寄存器和计数器的改写”。

通过16位存储器操作指令设置TA0寄存器。在产生复位信号后，TA0寄存器的值变为“FFFFH”。

图7-4：定时器A计数寄存器0(TA0)的格式

地址：40042300H 复位后：FFFFH R/W



—	功能	设置范围
bit15~0	16位计数器 ^{注1,2}	0000H~FFFFH

注1：如果给TACR0寄存器的TSTOP位写“1”，就强制停止16位计数器的计数，并且计数值变为“FFFFH”。

注2：如果TAMR0寄存器的TCK2~TCK0位的设置值不为“001B”(F_{CLK}/8)和“011B”(F_{CLK}/2)而TA0寄存器的值为“0000H”，就只在开始计数后立即向DMA和EVENTC产生1次请求信号。但是，TAO和TAIO进行交替输出。

在事件计数器模式中，与TCK2~TCK0位的值无关，如果TA0寄存器的值为“0000H”，就只在开始计数后立即向DMA和EVENTC产生1次请求信号，并且即使不在计数指定周期，TAO也进行交替输出。

如果TA0寄存器的值大于等于“0001H”，就在TA每次发生下溢时产生请求信号。

注意：当存取TA0寄存器时，CPU不进入下一条指令的处理而处于CPU处理的等待状态。因此，当发生此等待时，指令执行的时钟数增加等待的时钟数。存取TA0寄存器时的读写等待时钟数都为1个时钟。

7.3.4 定时器A控制寄存器0(TACR0)

TACR0寄存器是控制寄存器A的计数和停止以及表示定时器A状态的寄存器。

通过8位存储器操作指令设置TACR0寄存器。

在产生复位信号后，TACR0寄存器的值变为“00H”。

图7-5：定时器A控制寄存器0(TACR0)的格式

地址：40042240H 复位后：00H	R/W							
符号	7	6	5	4	3	2	1	0
TACR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

TUNDF	定时器A的下溢标志
0	没有发生下溢。
1	发生下溢。
[为“0”的条件] •当通过程序给此位写“0”时 [为“1”的条件] •当计数器发生下溢时	

TEDGF	有效边沿的判断标志
0	没有有效边沿。
1	有有效边沿。
[为“0”的条件] •当通过程序给此位写“0”时 [为“1”的条件] •在脉宽测量模式中外部输入(TAIO)的有效宽度测量结束时 •在脉冲周期测量模式中外部输入(TAIO)的设置边沿时	

TSTOP	定时器A的计数强制停止注1
如果给此位写“1”，就强制停止计数。读取值为“0”。	

TCSTF	定时器A的计数状态标志注2
0	停止计数。
1	正在计数。
[为“0”的条件] •当给TSTART位写“0”时(与计数源同步变为“0”) •当给TSTOP位写“1”时 [为“1”的条件] •当给TSTART位写“1”时(与计数源同步变为“1”)	

TSTART	定时器A的计数开始注2
0	停止计数。
1	开始计数。
通过给TSTART位写“1”开始计数；通过给TSTART位写“0”停止计数。如果将TSTART位置“1”(开始计数)，TCSTF位就与计数源同步变为“1”(正在计数)。另外，在给TSTART位写“0”后，TCSTF位与计数源同步变为“0”(停止计数)。详细内容请参照“7.5.1 计数的开始和停止控制”。	

注1：如果给TSTOP位写“1”(强制停止计数)，TSTART位和TCSTF位就同时被初始化，并且脉冲输出电平也被初始化。

注2：有关使用TSTART位和TCSTF位时的注意事项，请参照“7.5.1 计数的开始和停止控制”。

7.3.5 定时器AI/O控制寄存器0(TAIOC0)

TAIOC0寄存器是设置定时器A的输入/输出的寄存器。通过8位存储器操作指令设置TAIOC0寄存器。在产生复位信号后，TAIOC0寄存器的值变为“00H”。

图7-6：定时器AI/O控制寄存器0(TAIOC0)的格式

地址：40042241H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
TAIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	TAIO的计数控制 ^{注1,2}
0	0	总是对事件进行计数。
0	1	在INTP4指定的极性期间对事件进行计数。
1	0	在定时器输出信号指定的极性期间对事件进行计数。
上述以外		禁止设置。

TIPF1	TIPF0	TAIO输入滤波器的选择
0	0	没有滤波器。
0	1	有滤波器，通过F _{CLK} 进行采样。
1	0	有滤波器，通过F _{CLK} /8进行采样。
1	1	有滤波器，通过F _{CLK} /32进行采样。

这些位指定TAIO输入滤波器的采样频率。对TAIO引脚的输入进行采样，如果采样值连续3次相同，就确定此值为输入值。

TOENA	TAO输出的允许
0	禁止TAO输出(端口)。
1	允许TAO输出。

TEDGSEL	输入/输出的极性切换
功能因运行模式而不同(参照表7-4和表7-5)。	

注1：当使用INTP4或者定时器输出信号时，能通过TAISR0寄存器的RCCPSEL2位选择事件的计数极性。

注2：TIOGT0位和TIOGT1位只在事件计数器模式中有效。

表7-4: TAIO输入/输出的边沿和极性切换

运行模式	功能
定时器模式	不使用(输入/输出端口)。
脉冲输出模式	0: 从“H”电平开始输出(初始电平: “H”) 1: 从“L”电平开始输出(初始电平: “L”)
事件计数器模式	0: 在上升沿进行计数 1: 在下降沿进行计数
脉宽测量模式	0: 测量“L”电平宽度 1: 测量“H”电平宽度
脉冲周期测量模式	0: 在测量脉冲的上升沿到下一个上升沿之间进行测量 1: 在测量脉冲的下降沿到下一个下降沿之间进行测量

表7-5: TAO输出的极性切换

运行模式	功能
全部模式	0: 从“L”电平开始输出(初始电平: “L”) 1: 从“H”电平开始输出(初始电平: “H”)

7.3.6 定时器A控制寄存器0(TAMR0)

TAMR0寄存器是设置寄存器A的运行模式的寄存器。通过8位存储器操作指令设置TAMR0寄存器。在产生复位信号后，TAMR0寄存器的值变为“00H”。

图7-7：定时器A控制寄存器0(TAMR0)的格式

地址：40042242H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
TAMR0	0	TCK2	TCK1	TCK0	TEDGPL	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	定时器A的计数源选择 ^{注1,2}
0	0	0	F _{CLK}
0	0	1	F _{CLK} /8
0	1	1	F _{CLK} /2
1	0	0	F _{IL}
1	0	1	EVENTC输入的事件
1	1	0	F _{SUB}
上述以外			禁止设置。

TEDGPL	TAIO边沿极性的选择 ^{注5}
0	单边沿
1	双边沿

TMOD2	TMOD1	TMOD0	定时器A运行模式的选择 ^{注3}
0	0	0	定时器模式
0	0	1	脉冲输出模式
0	1	0	事件计数器模式
0	1	1	脉宽测量模式
1	0	0	脉冲周期测量模式
上述以外			禁止设置。

- 注1：如果选择事件计数器模式，就选择外部输入(TAIO)作为计数源，而与TCK0~TCK2位的设置无关。
- 注2：不能在计数过程中切换计数源。如果要切换计数源，就必须在TACR0寄存器的TSTART位和TCSTF位都为“0”(停止计数)时进行切换。
- 注3：只有在停止计数(TACR0寄存器的TSTART位和TCSTF位都为“0”(停止计数))时才能更改运行模式，不能在计数过程中进行更改。
- 注4：要选择F_{IL}作为计数源时，必须将副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位置“1”。但是，当选择F_{SUB}作为实时时钟或者12位间隔定时器的计数源时，不能选择F_{IL}作为定时器A的计数源。
- 注5：TEDGPL位只在事件计数器模式中有效。
- 注6：通过写TAMR0寄存器，对定时器A的TAO引脚和TAIO引脚的输出进行初始化。有关初始化时的输出电平，请参照“图7-6：定时器AI/O控制寄存器0(TAIOC0)的格式”的说明。

7.3.7 定时器A事件引脚选择寄存器0(TAISR0)

TAISR0寄存器是选择在事件计数器模式中控制事件计数期间的定时器以及设置极性的寄存器。通过8位存储器操作指令设置TAISR0寄存器。

在产生复位信号后，TAISR0寄存器的值变为“00H”。

图7-8：定时器A事件引脚选择寄存器0(TAISR0)的格式

地址：40042243H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
TAISR0	0	0	0	0	0	RCCPSEL2 ^注	RCCPSEL1 ^注	RCCPSEL0 ^注

RCCPSEL2 ^注	定时器输出信号和INTP4极性的选择
0	在“L”电平期间对事件进行计数。
1	在“H”电平期间对事件进行计数。

RCCPSEL1 ^注	RCCPSEL0 ^注	定时器输出信号的选择
0	0	TMIOD1
0	1	TMIOC1
1	0	TO02
1	1	TO03

注：RCCPSEL0~RCCPSEL2位只在事件计数器模式中有效。

7.3.8 端口模式寄存器x(PMx)

这是设置端口输入/输出的寄存器。

要将定时器输出引脚的复用端口(TAIO、TAO等)用作定时器的输出时，必须将各端口对应的端口模式寄存器(PMxx)的位和端口寄存器(Pxx)的位置“0”。

(例)将P01用作定时器输出TAIO的情况将端口模式寄存器0的PM01位置“0”。将端口寄存器0的P01位置“0”。

要将定时器输入引脚的复用端口(P01/TAIO等)用作定时器的输入时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”。此时，端口寄存器(Pxx)的位可以是“0”或者“1”。

(例)将P01用作定时器输入TAIO的情况将端口模式寄存器0的PM01位置“1”。将端口寄存器0的P01位置“0”或者“1”。

通过8位存储器操作指令设置PMxx寄存器。在产生复位信号后，这些寄存器的值变为“FFH”。

有关端口模式寄存器的格式，请参照“表2-1：各产品分配的PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx寄存器及其位”。

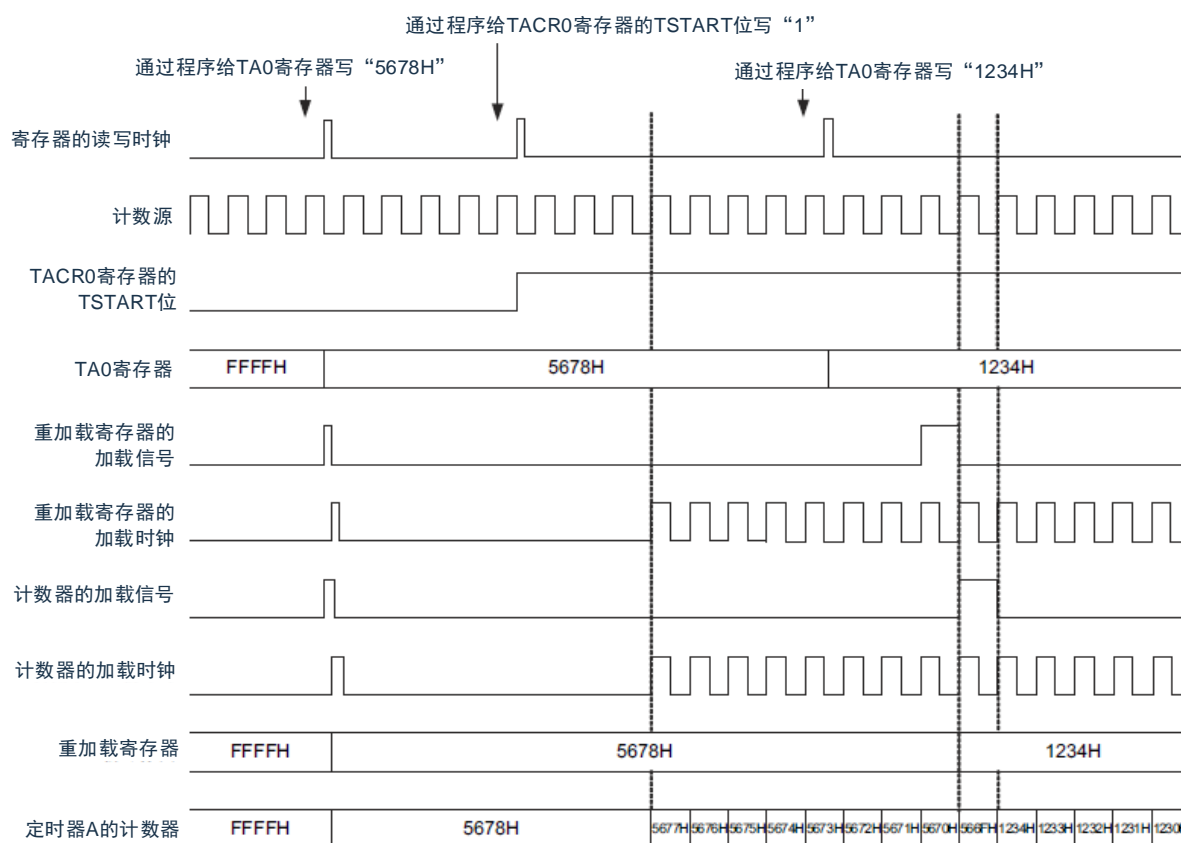
7.4 定时器A的运行

7.4.1 重加载寄存器和计数器的改写

与运行模式无关，重加载寄存器和计数器的改写时序因TACR0寄存器的TSTART位的值而变。当TSTART位为“0”(停止计数)时，直接写重加载寄存器和计数器；当TSTART位为“1”(开始计数)时，在与计数源同步写重加载寄存器后，与下一个计数源同步写计数器。

由TSTART位的值决定的改写时序图如图7-9所示。

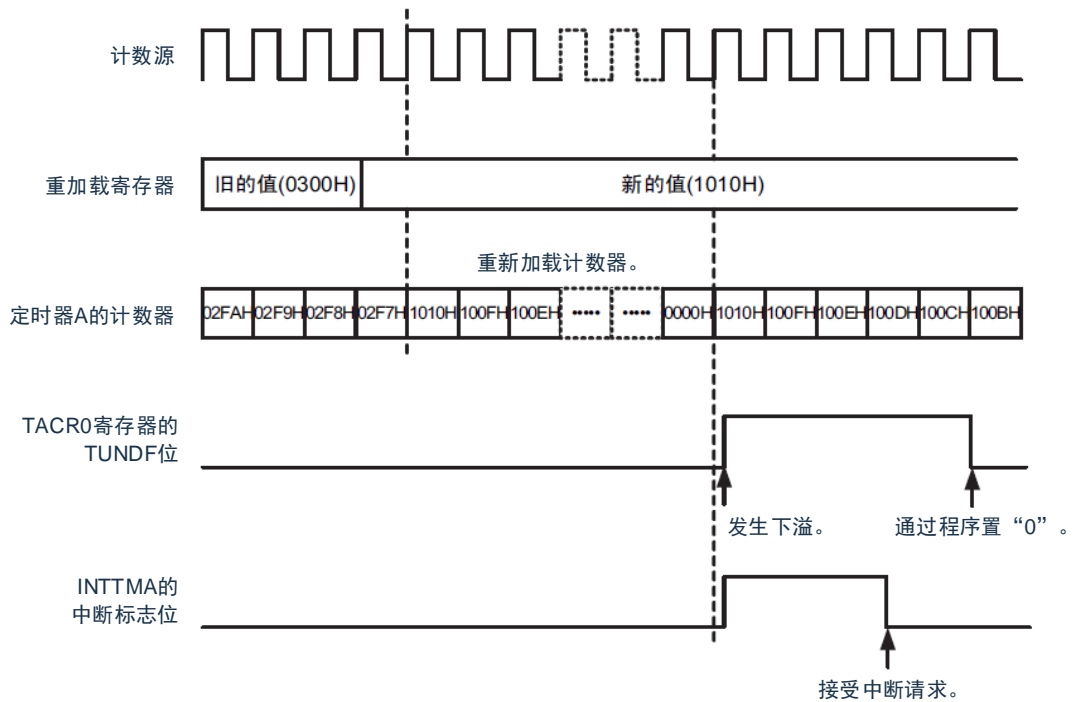
图7-9：由TSTART位的值决定的改写时序图



7.4.2 定时器模式

这是通过TAMR0寄存器的TCK0~TCK2位选择的计数源进行递减计数的模式。在定时器模式中，每当输入计数源时计数值就减1，如果计数值变为“0000H”并且输入下一个计数源，就发生下溢并且产生中断请求。定时器模式的运行例子如图7-10所示。

图7-10：定时器模式的运行例子



7.4.3 脉冲输出模式

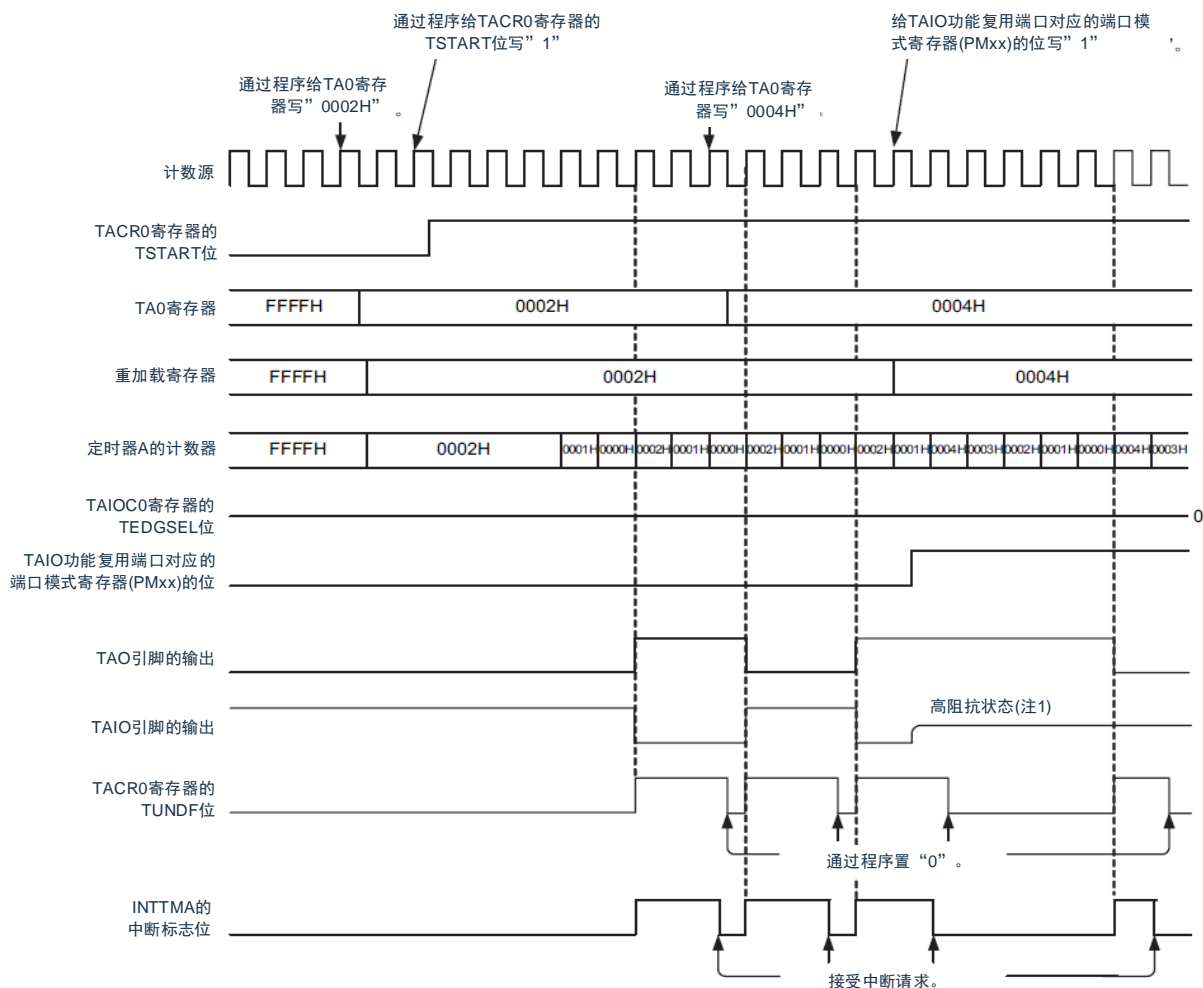
在此模式中，通过TAMR0寄存器的TCK0~TCK2位选择的计数源进行递减计数，每当发生下溢时，就将TAIO引脚和TAO引脚的输出电平反相输出。

在脉冲输出模式中，每当输入计数源时计数值就减1，如果计数值变为“0000H”并且输入下一个计数源，就发生下溢并且产生中断请求。

能从TAIO引脚和TAO引脚输出脉冲，并且每当发生下溢时就将输出电平进行反相。能通过TAIOC0寄存器的TOENA位停止TAO引脚的脉冲输出。

另外，能通过TAIOC0寄存器的TEDGSEL位选择输出电平。脉冲输出模式的运行例子如图7-11所示。

图7-11：脉冲输出模式的运行例子



注1：通过被选择为TAIO功能的端口输出允许控制，变为高阻抗状态。

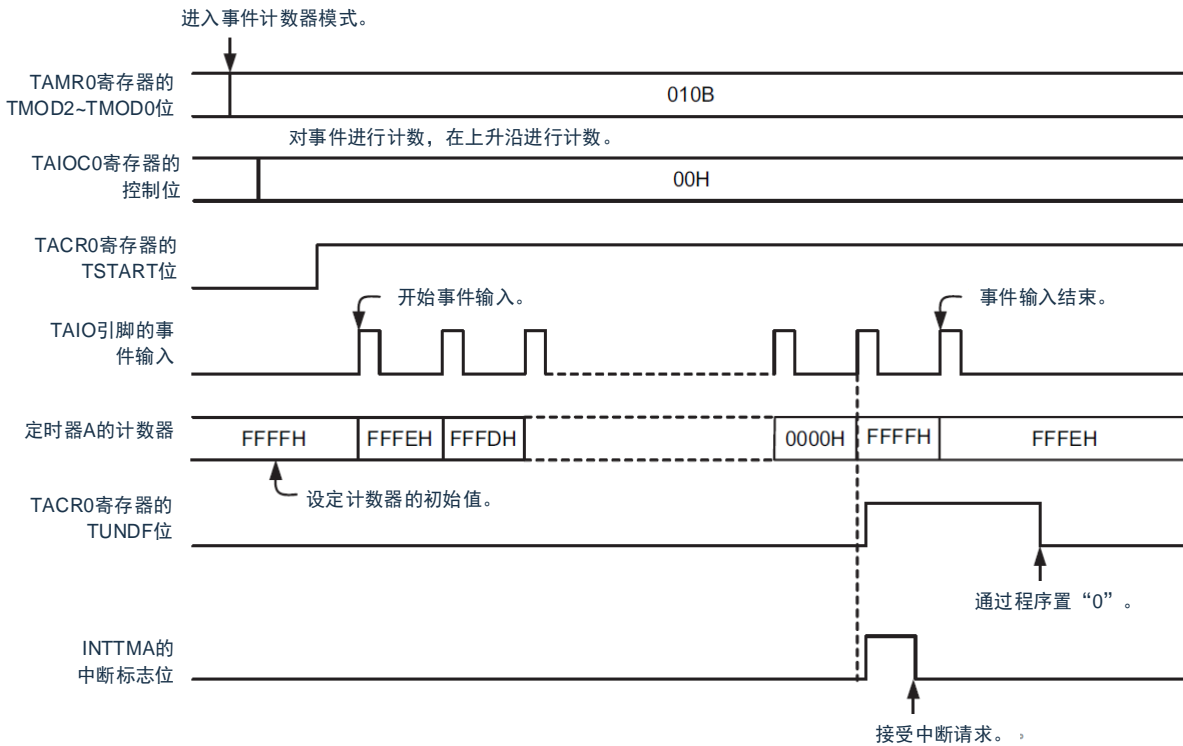
7.4.4 事件计数器模式

这是通过TAIO引脚输入的外部事件信号(计数源)进行递减计数的模式。

能通过TAIOC0寄存器的TIOGT0~TIOGT1位和TAISR0寄存器进行事件计数期间的各种设置，并且能通过TAIOC0寄存器的TIPF0~TIPF1位指定TAIO输入的滤波器功能。

即使在事件计数器模式中TAO引脚也能进行交替输出。要使用事件计数器模式时，请参照“7.5.5 TAO引脚和TAIO引脚的设置步骤”。事件计数器模式的运行例子1如图7-12所示。

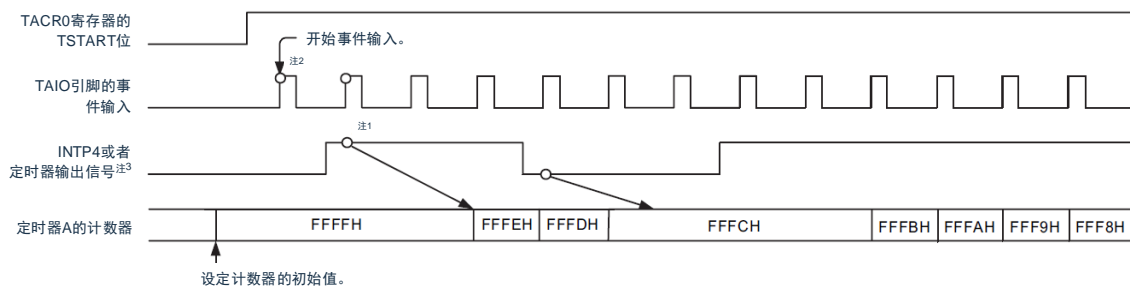
图7-12: 事件计数器模式的运行例子1



事件计数器模式中的指定时间计数(TAIOC0寄存器的TIOGT1位和TIOGT0位为“01B”或者“10B”)的运行例子如图7-13所示。

图7-13: 事件计数器模式的运行例子2

- 将运行模式设定为以下情况的时序例子
- TAMR0寄存器: TMOD2, 1, 0=010B(事件计数器模式)
- TAIOC0寄存器: TIOGT1, 0=01B(外部中断引脚制定期间的事件计数)
- TIPF1, 0=00B(没有滤波器)
- TEDGSEL=0(在上升沿进行计数)
- TAISR0寄存器: RCCPSEL2=1(对H期间进行计数)



以下注意事项限于事件计数模式的运行模式设定TAIOC0寄存器的TIOGT1和TIOGT0位为“01B”或者“10B”的情况。

- 注1. 为了进行同步控制，可能在反映到计数运行前发生2个计数源时钟的延迟。
- 注2. 开始计数后的2个计数源时钟可能根据前一次计数停止前的状态进行计数，要使开始计数后的2个计数源时钟变为无效时，必须给TACR0寄存器的TSTOP位写“1”，对内部电路进行初始化并且在运行设定后开始计数。
- 注3. 对于TAISR0寄存器的RCCPSEL1和RCCPSEL0位选择的定时器输出信号，不能将分配给该定时器输出功能的引脚用作定时器以外的复用功能的输出。

7.4.5 脉宽测量模式

这是测量TAIO引脚输入的外部信号脉宽的模式。

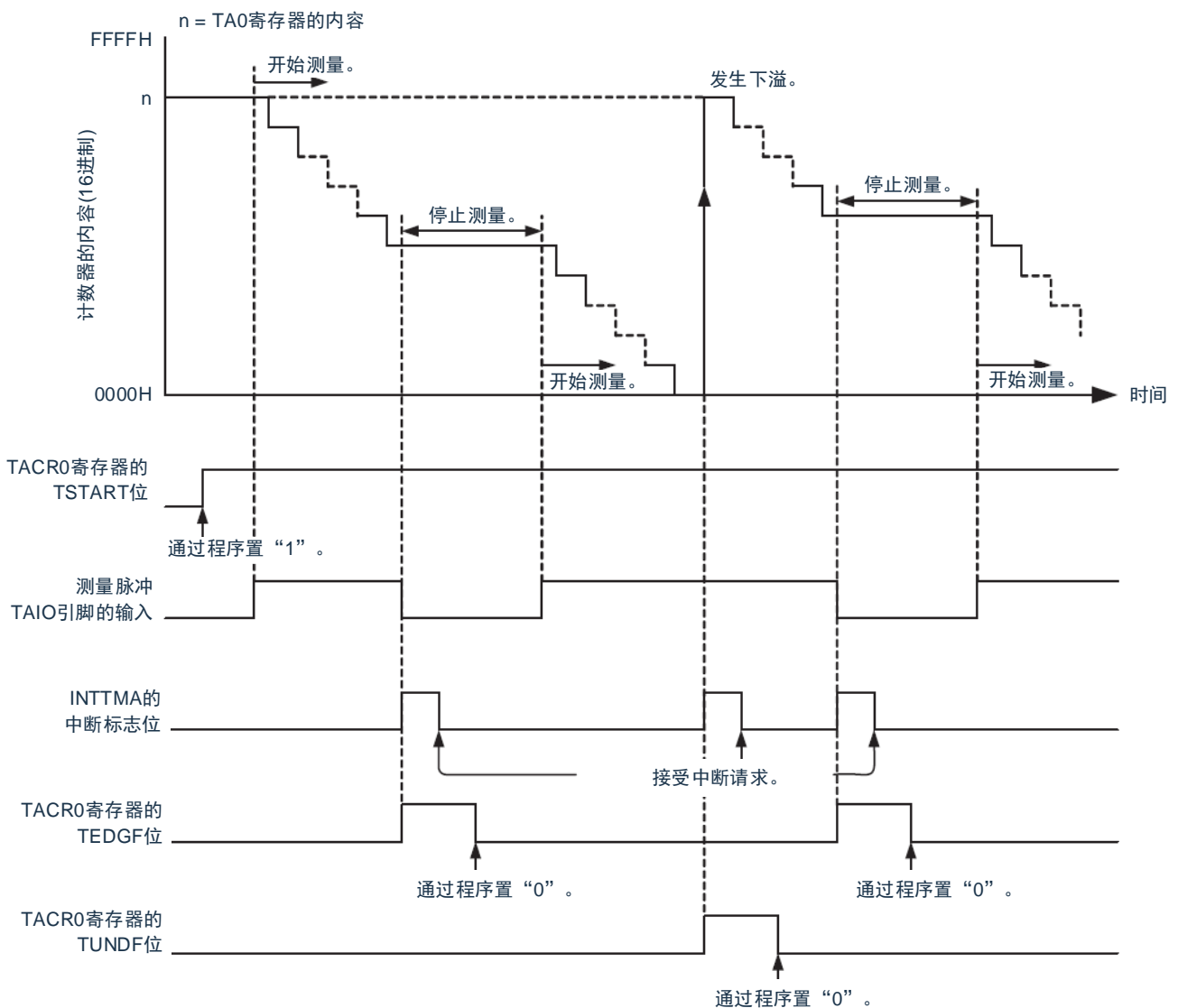
在脉宽测量模式中，如果给TAIO引脚输入TAIOC0寄存器的TEDGSEL位指定的电平，就通过选择的计数源开始递减计数。如果TAIO引脚输入的指定电平结束，计数器就停止计数，TACR0寄存器的TEDGF位变为“1”(有效边沿)并且产生中断请求。通过在计数器停止计数时读计数值进行脉宽数据的测量。如果在测量过程中计数器发生下溢，TACR0寄存器的TUNDF位就变为“1”(发生下溢)并且产生中断请求。

脉宽测量模式的运行例子如图7-14所示。

要存取TACR0寄存器的TEDGF位和TUNDF位时，请参照“7.5.2 标志的存取(TACR0寄存器的TEDGF位和TUNDF位)”。

图7-14：脉冲测量模式的运行例子

这是对测量脉冲的“H”电平进行测量的情况(TAIOC0寄存器的TEDGSEL=1).



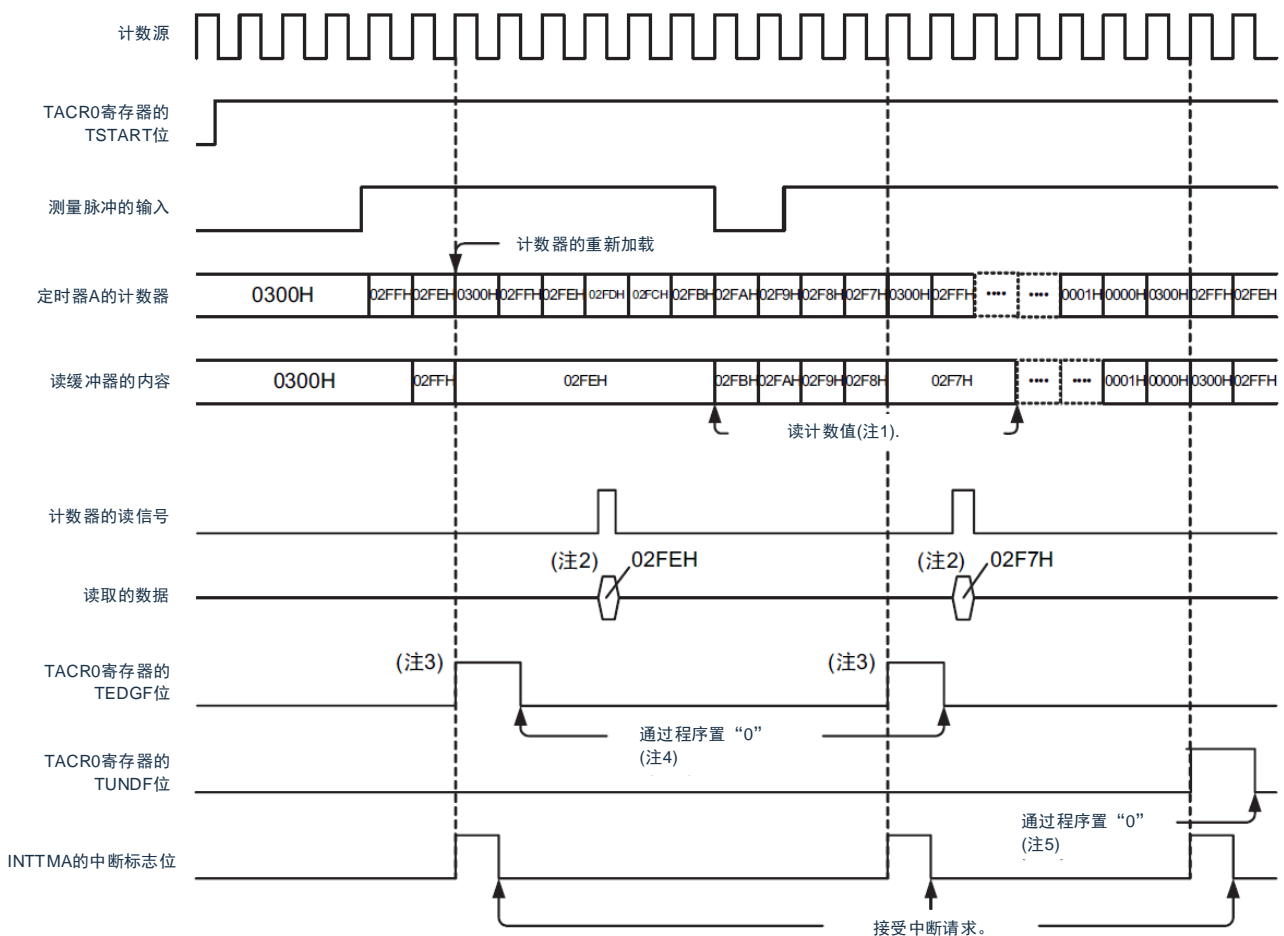
7.4.6 脉冲周期测量模式

这是测量TAIO引脚输入的外部信号脉冲周期的模式。

通过TAMR0寄存器的TCK0~TCK2位选择的计数源，计数器进行递减计数。如果给TAIO引脚输入TAIOC0寄存器的TEDGSEL位指定期间的脉冲，就在计数源的上升沿将计数值传送到读缓冲器，并且在下一个上升沿将重加载寄存器的值加载到计数器，同时TACR0寄存器的TEDGF位变为“1”(有有效边沿)并且产生中断请求。此时，读TAO寄存器(读缓冲器)，读取值和重加载值的差为输入脉冲的周期数据。周期数据被保持到读缓冲器为止。如果计数器发生下溢，TACR0寄存器的TUNDF位就变为“1”(发生下溢)并且产生中断请求。脉冲周期测量模式的运行例子如图7-15所示。

必须输入大于计数源2倍周期的脉冲，而且输入的“L”电平和“H”电平的宽度都必须大于计数源周期的脉冲。如果输入的脉冲周期和宽度不满足这些条件，输入的脉冲就可能被忽视。

图7-15: 脉冲周期测量模式的运行例子



这是TAO寄存器的初始值为“0300H”并且将TAIOC0寄存器的TEDGSEL位置“0”以及在测量脉冲上升前进行测量的情况。

- 注1.必须是从TEDGF位变为“1”(有有效边沿)到输入下一个有效边沿为止的期间读TAO寄存器。读缓冲器的内容被保持到读TAO寄存器为止，因此，如果在输入有效边沿前不读TAO寄存器，就保持以前周期的测量结果。
- 2.如果在脉冲周期测量模式中读TAO寄存器，读取值即为读缓冲器的内容。
- 3.如果在输入测量脉冲的有效边沿后输入外部脉冲的设定边沿，TACR0寄存器的TEDGF位就变为“1”(有有效边沿)。
- 4.要通过程序置“0”时，必须通过8位操作指令给TACR0寄存器的TEDGF位写“0”。
- 5.要通过程序置“0”时，必须通过8位操作指令给TACR0寄存器的TUNDF位写“0”。

7.4.7 与EVENTC的协作

能通过与EVENTC的协作，将EVENTC输入的事件设置为计数源。

通过TAMR0寄存器的TCK0~TCK2位，在ELC输入的事件上升沿进行计数。但是，在事件计数器模式中EVENTC输入不起作用。

EVENTC设置步骤如下所示。

- 开始运行的步骤
 - (1) 设置EVENTC的事件输出目标选择寄存器(ELSELRn)。
 - (2) 设置事件发生源的运行模式。
 - (3) 设置定时器A的模式。
 - (4) 开始定时器A的计数。
 - (5) 开始事件发生源的运行。
- 停止运行的步骤
 - (1) 停止事件发生源的运行。
 - (2) 停止定时器A的计数。
 - (3) 将EVENTC的事件输出目标选择寄存器(ELSELRn)置“0”。

7.4.8 各模式的输出设置

各模式中的TAO引脚和TAIO引脚的状态如表7-6和表7-7所示。

表7-6: TAO引脚的设置

运行模式	TAIOC0寄存器		TAO引脚的输出
	TOENA位	TEDGSEL位	
全部模式	1	1	反相输出
		0	正相输出
	0	0或者1	禁止输出

表7-7: TAIO引脚的设置

运行模式	TAIOC0寄存器		TAIO引脚的输入/输出
	PMXX位注	TEDGSEL位	
定时器模式	0或者1	0或者1	输入(不使用)
脉冲输出模式	0	0或者1	禁止输出(Hi-Z输出)
		1	正相输出
事件计数器模式	1	0	反相输出
		0或者1	输入
脉宽测量模式	1	0或者1	输入
脉冲周期测量模式			

注：这是和TAIO功能复用端口对应的端口模式寄存器(PMxx)的位。

7.5 使用定时器A时的注意事项

7.5.1 计数的开始和停止控制

- 事件计数模式或者将计数源设置为非EVENTC的情况

如果在计数停止过程中给TACR0寄存器的TSTART位写“1”(开始计数),就在3个计数源周期内TACR0寄存器的TCSTF位为“0”(停止计数)。除了TCSTF位以外,不能在TCSTF位变为“1”(正在计数)前存取定时器A的相关寄存器注。

如果在计数过程中给TSTART位写“0”(停止计数),就在3个计数源周期内TCSTF位为“1”。在TCSTF位变为“0”时停止计数。除了TCSTF位以外,不能在TCSTF位变为“0”前存取定时器A的相关寄存器注。必须在将TSTART位从“0”改为“1”前清除中断寄存器。详细内容请参照“第25章 中断功能”。

注:定时器A的相关寄存器:TA0、TACR0、TAIOC0、TAMR0、TAISR0

- 事件计数模式或者将计数源设置为EVENTC的情况

如果在计数停止过程中给TACR0寄存器的TSTART位写“1”(开始计数),就在2个CPU时钟周期内TACR0寄存器的TCSTF位为“0”(停止计数)。除了TCSTF位以外,不能在TCSTF位变为“1”(正在计数)前存取定时器A的相关寄存器注。

如果在计数过程中给TSTART位写“0”(停止计数),就在2个CPU时钟周期内TCSTF位为“1”。在TCSTF位变为“0”时停止计数。除了TCSTF位以外,不能在TCSTF位变为“0”前存取定时器A的相关寄存器注。必须在将TSTART位从“0”改为“1”前清除中断寄存器。详细内容请参照“第25章中断功能”。

注:定时器A的相关寄存器:TA0、TACR0、TAIOC0、TAMR0、TAISR0

7.5.2 标志的存取(TACR0寄存器的TEDGF位和TUNDF位)

如果通过程序给TACR0寄存器的TEDGF位和TUNDF位写“0”,这些位就变为“0”。但是,即使写“1”值也不变。如果对TACR0寄存器使用读-修改-写指令,就在指令执行过程中即使TEDGF位变为“1”(有有效边沿)并且TUNDF位变为“1”(发生下溢),也可能因时序而误将TEDGF位和TUNDF位置“0”。必须通过8位存储器操作指令存取TACR0寄存器。

7.5.3 计数寄存器的存取

在TACR0寄存器的TSTART位和TCSTF位都为“1”(正在计数)的情况下连续写TA0寄存器时,必须在各自的写操作之间至少间隔3个计数源时钟周期。

7.5.4 模式的变更

只有在停止计数(TACR0寄存器的TSTART位和TCSTF位都为“0”(停止计数))时才能更改定时器A的运行模式相关寄存器(TAIOC0、TAMR0、TAISR0),不能在计数过程中进行更改。

当更改定时器A的运行模式相关寄存器时,TEDGF位和TUNDF位的值为不定值。必须在给TEDGF位写“0”(没有有效边沿)并且给TUNDF位写“0”(没有发生下溢)后开始计数。

7.5.5 TAO引脚和TAIO引脚的设置步骤

在复位后，TAO引脚和TAIO引脚的复用I/O端口为输入端口。要从TAO引脚和TAIO引脚输出时，必须按照以下步骤进行设置。

更改步骤

- (1) 设置模式。
- (2) 设置初始值，允许输出。
- (3) 将TAO引脚和TAIO引脚对应的端口寄存器的位置“0”。
- (4) 将TAO引脚和TAIO引脚对应的端口模式寄存器的位设置为输出模式。
(从TAO引脚和TAIO引脚开始输出)
- (5) 开始计数(TACR0寄存器的TSTART=1)。

要从TAIO引脚输入时，必须按照以下步骤进行设置。

- (1) 设置模式。
- (2) 设置初始值，选择边沿。
- (3) 将TAIO引脚对应的端口模式寄存器的位设置为输入模式。
(从TAIO引脚开始输入)
- (4) 开始计数(TAMR0寄存器的TSTART=1)。
- (5) 等到TACR0寄存器的TCSTF位变为“1”(正在计数)。
(只限于事件计数器模式)
- (6) 从TAIO引脚输入外部事件。
- (7) 必须在第一次测量结束时进行测量值的无效处理(第二次及以后的测量值有效)。
(只限于脉宽测量模式和脉冲周期测量模式)

7.5.6 不使用定时器A的情况

当不使用定时器A时，必须将TAMR0寄存器的TMOD2~TMOD0位置“000B”(定时器模式)并且将TAIOC0寄存器的TOENA位置“0”(禁止TAO输出)。

7.5.7 定时器A运行时钟的停止

能通过PER1寄存器的TMAEN位控制定时器A时钟的提供或者停止。但是，不能在定时器A的时钟停止时存取以下的SFR，而必须在提供定时器A时钟的状态下进行存取。

TAO寄存器、TACR0寄存器、TAMR0寄存器、TAIOC0寄存器和TAISR0寄存器

7.5.8 深度睡眠模式(事件计数器模式)的设置步骤

要在深度睡眠模式中使事件计数器模式运行时，必须在提供定时器A的时钟后按照以下步骤转移到深度睡眠模式。

设置步骤

- (1) 设置运行模式。
- (2) 开始计数(TSTART=1、TCSTF=1)。
- (3) 停止提供定时器A的时钟。

要在深度睡眠模式中停止事件计数器模式时，必须按照以下步骤进行运行停止处理。

- (1) 提供定时器A的时钟。
- (2) 停止计数(TSTART=0、TCSTF=0)

7.5.9 深度睡眠模式中(只限于事件计数器模式)的功能限制

要在深度睡眠模式中使事件计数器模式运行时，不能使用数字滤波器功能。

7.5.10 通过TSTOP位进行强制的计数停止

不能在通过TACR0寄存器的TSTOP位强制停止计数器的计数后的1个计数源周期内存取以下的SFR。TAO寄存器、TACR0寄存器和TAMR0寄存器

7.5.11 数字滤波器

当使用数字滤波器时，不能在设置TAIOC寄存器的TIPF1位和TIPF0位后的5个数字滤波器时钟周期内开始定时器的运行。

另外，在使用数字滤波器的状态下，即使更改TAIOC寄存器的TEDGSEL位，也同样不能在5个数字滤波器时钟周期内开始定时器的运行。

7.5.12 选择F_{IL}作为计数源的情况

要选择F_{IL}作为计数源时，必须将副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位置“1”。但是，当选择F_{SUB}作为实时时钟或者15位间隔定时器的计数源时，不能选择F_{IL}作为定时器A的计数源。

第8章 定时器B

8.1 定时器B的功能

定时器B有以下3种模式：

- 定时器模式：
 - 输入捕捉功能 在上升沿、下降沿或者上升沿/下降沿的双边沿进行计数。
 - 输出比较功能 “L”电平输出、“H”电平输出或者交替输出
- PWM模式：能进行任意占空比的PWM输出。
- 相位计数模式：能自动测量2相编码器的计数值。

8.2 定时器B的结构

定时器B的框图和引脚结构分别如图8-1和表8-1所示。

图8-1：定时器B的框图

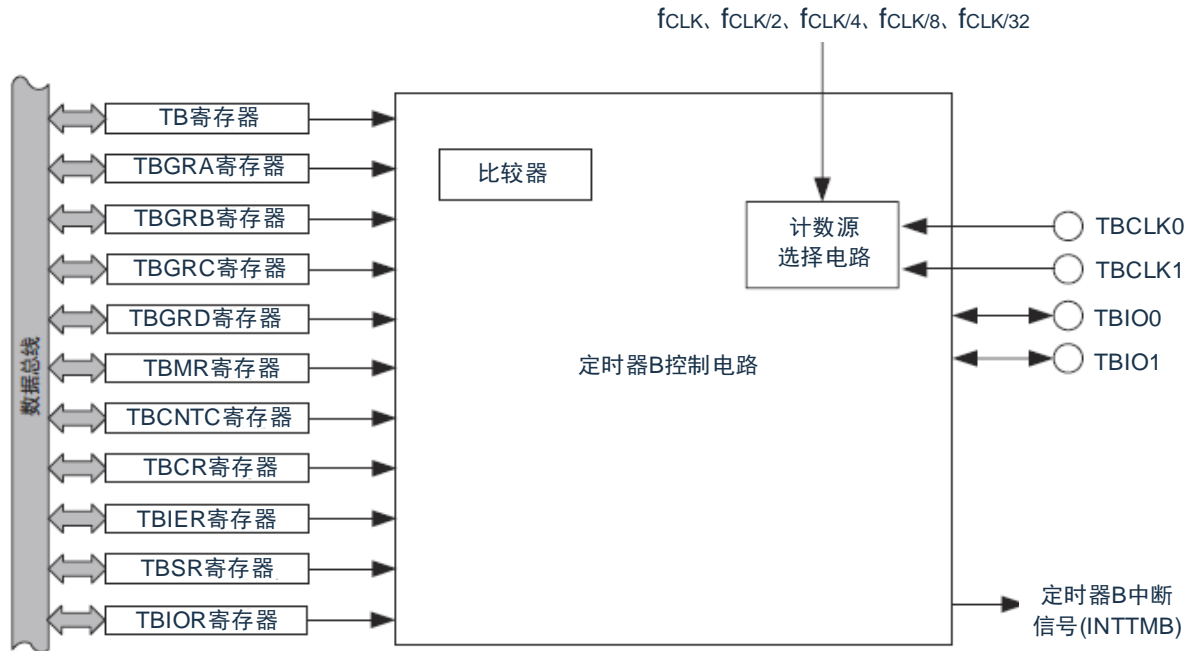


表8-1：定时器B的引脚结构

引脚名	复用的端口名	输入/输出	功能
TBCLK0	P00	输入	<ul style="list-style-type: none"> 相位计数模式 A相输入 非相位计数模式外部时钟0的输入
TBCLK1	P01	输入	<ul style="list-style-type: none"> 相位计数模式 B相输入 非相位计数模式外部时钟1的输入
TBIO0	P50	输入/输出	<ul style="list-style-type: none"> 定时器模式(输出比较功能) TBGRA输出比较的输出 定时器模式(输入捕捉功能) TBGRA输入捕捉的输入 PWM模式 PWM输出
TBIO1	P51	输入/输出	<ul style="list-style-type: none"> 定时器模式(输出比较功能) TBGRB输出比较的输出 定时器模式(输入捕捉功能) TBGRB输入捕捉的输入

8.3 控制定时器B的寄存器

控制定时器B的寄存器如表8-2所示。

表8-2：控制定时器B的寄存器

寄存器名	符号
外围允许寄存器1	PER1
定时器B模式寄存器	TBMR
定时器B计数控制寄存器	TBCNTC
定时器B控制寄存器	TBCR
定时器B中断允许寄存器	TBIER
定时器B状态寄存器	TBSR
定时器B I/O控制寄存器	TBIOR
定时器B计数器	TB
定时器B通用寄存器A	TBGRA
定时器B通用寄存器B	TBGRB
定时器B通用寄存器C	TBGRC
定时器B通用寄存器D	TBGRD
端口寄存器	Pxx
端口模式寄存器	PMxx

8.3.1 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器B时，必须将bit6(TMBEN)置“1”。

通过8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图8-2：外围允许寄存器1(PER1)的格式

地址：0x4002081A

复位后：00H

R/W

符号

7

6

5

4

3

2

1

0

PER1

DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMOPEN	TMCEN ^{注2}	TMAEN
-------	-------	----------	-------	-------	---------	---------------------	-------

TMBEN	提供定时器B的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器B使用的SFR。 •定时器B处于复位状态。
1	提供输入时钟。 •能读写定时器B使用的SFR。

注意：要设置定时器B时，必须先将TMBEN位置“1”。当TMBEN位为“0”时，忽视定时器B的控制寄存器的写操作，而且读取值都为初始值(端口模式寄存器(PMxx)和端口寄存器(Pxx)除外)。

8.3.2 定时器B模式寄存器(TBMR)

图8-3: 定时器B模式寄存器(TBMR)的格式

地址: 40042650H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TBMR	TBSTART	TBELCICE	TBDFCK1	TBDFCK0	TBDFB	TBDFA	TBMDF	TBPWM

TBSTART	TB计数的开始
0	停止计数, 并且对PWM输出信号(TBIO0引脚)进行初始化(PWM模式)。
1	开始计数。

TBELCICE	EVENTC输入捕捉请求的选择 ^{注1,2}
0	选择外部输入信号1/数字滤波器通过信号1。
1	选择EVENTC输入的事件(输入捕捉)。

TBDFCK1	TBDFCK0	数字滤波器功能所用时钟的选择 ^{注1}
0	0	$F_{CLK}/32$
0	1	$F_{CLK}/8$
1	0	F_{CLK}
1	1	TBCR寄存器的TBTK0~TBTK2位选择的时钟

TBDFB	TBIO1引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要5个数字滤波器的采样时钟周期进行边沿检测。	

TBDFA	TBIO0引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。
当有数字滤波器功能时, 最多需要5个数字滤波器的采样时钟周期进行边沿检测。	

TBMDF	相位计数模式的选择
0	递增计数
1	相位计数模式
当TBMDF位为“0”时, 计数器对TBCR寄存器的TBTK0~TBTK2位设置的计数源进行计数; 当TBMDF位为“1”时, 计数器对“表8-15TB寄存器的加减条件”所示的TBCLKj引脚(j=0、1)输入信号的相位进行计数。	

TBPWM	PWM模式的选择
0	定时器模式
1	PWM模式

注1: 不能在TBSTART位为“0”(停止计数)时设置此位。

注2: 要使EVENTC输入的事件(输入捕捉)有效时, 必须将TBIOR寄存器的TBIOR2位置“1”, 并且将TBIOR1位和TBIOR0位置“00B”(上升沿)。

8.3.3 定时器B计数控制寄存器(TBCNTC)

在相位计数模式中使用TBCNTC寄存器，设置相位计数模式的计数条件。

图8-4：定时器B计数控制寄存器(TBCNTC)的格式

地址：40042651H

复位后：00H

R/W

符号	7	6	5	4	3	2	1	0
TBCNTC	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

CNTEN7	计数的允许7
0	无效
1	递增计数 TBCLK0输入为“L”电平并且在TBCLK1输入的上升沿时

CNTEN6	计数的允许6
0	无效
1	递增计数 TBCLK1输入为“H”电平并且在TBCLK0输入的上升沿时

CNTEN5	计数的允许5
0	无效
1	递增计数 TBCLK0输入为“H”电平并且在TBCLK1输入的下降沿时

CNTEN4	计数的允许4
0	无效
1	递增计数 TBCLK1输入为“L”电平并且在TBCLK0输入的下降沿时

CNTEN3	计数的允许3
0	无效
1	递减计数 TBCLK1输入为“H”电平并且在TBCLK0输入的下降沿时

CNTEN2	计数的允许2
0	无效
1	递减计数 TBCLK0输入为“L”电平并且在TBCLK1输入的下降沿时

CNTEN1	计数的允许1
0	无效
1	递减计数 TBCLK1输入为“L”电平并且在TBCLK0输入的上升沿时

CNTEN0	计数的允许0
0	无效
1	递减计数 TBCLK0输入为“H”电平并且在TBCLK1输入的上升沿时

8.3.4 定时器B控制寄存器(TBCR)

必须在TBMR寄存器的TBSTART位为“0”(停止计数)的状态下写TBCR寄存器。

图8-5: 定时器B控制寄存器(TBCR)的格式

地址: 40042652H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TBCR	0	TBCCLR1	TBCCLR0	TBCKEG1	TBCKEG0	TBTCK2	TBTCK1	TBTCK0

TBCCLR1	TBCCLR0	TB寄存器的清除源选择
0	0	禁止清除。
0	1	在TBGRA的输入捕捉或者比较匹配时进行清除。
1	0	在TBGRB的输入捕捉或者比较匹配时进行清除。
上述以外		禁止设置。

TBCKEG1	TBCKEG0	外部时钟的有效边沿的选择 ^{注1,2}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在上升沿/下降沿的双边沿进行计数。
上述以外		禁止设置。

TBTCK2	TBTCK1	TBTCK0	计数源的选择 ^{注1}
0	0	0	F _{CLK}
0	0	1	F _{CLK} /2
0	1	0	F _{CLK} /4
0	1	1	F _{CLK} /8
1	0	0	F _{CLK} /32
1	0	1	TBCLK0的输入
1	1	1	TBCLK1的输入
上述以外			禁止设置。

注1: 在相位计数模式中, TBTCK0~TBTCK2位、TBCKEG0位和TBCKEG1位的设置无效, 优先相位计数模式的运行。

注2: TBCKEG0位和TBCKEG1位在TBTCK0~TBTCK2位的设置为外部时钟(TBCLK0、TBCLK1)时有效, 否则无效。

8.3.5 定时器B中断允许寄存器(TBIER)

图8-6: 定时器B中断允许寄存器(TBIER)的格式

地址: 40042653H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TBIER	0	0	0	0	TBOVIE	TBUDIE	TBIMIEB	TBIMIEA

TBOVIE	上溢中断的允许
0	禁止因TBOVF位产生的中断。
1	因TBOVF位产生的中断有效。

TBUDIE	下溢中断的允许
0	禁止因TBUDF位产生的中断。
1	因TBUDF位产生的中断有效。

TBIMIEB	输入捕捉/比较匹配的中断允许B
0	禁止因TBIMFB位产生的中断。
1	因TBIMFB位产生的中断有效。

TBIMIEA	输入捕捉/比较匹配的中断允许A
0	禁止因TBIMFA位产生的中断。
1	因TBIMFA位产生的中断有效。

备注: TBIMFA、TBIMFB、TBUDF、TBOVF: TBSR寄存器的位

8.3.6 定时器B状态寄存器(TBSR)

图8-7: 定时器B状态寄存器(TBSR)的格式

地址: 40042654H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TBSR	0	0	0	TBDIRF	TBOVF	TBUDF	TBIMFB	TBIMFA

TBDIRF	计数方向标志
0	TB寄存器进行递减计数。
1	TB寄存器进行递增计数。

TBOVF	上溢标志 ^{注1}
[为“0”的条件]读后写“0” ^{注2} 。[为“1”的条件] 参照“表8-3: 各标志为“1”的条件”。	

TBUDF	下溢标志
[为“0”的条件]读后写“0” ^{注2} 。[为“1”的条件] 参照“表8-3: 各标志为“1”的条件”。	

TBIMFB	输入捕捉/比较匹配标志B
[为“0”的条件]读后写“0” ^{注2,3} 。[为“1”的条件] 参照“表8-3: 各标志为“1”的条件”。	

TBIMFA	输入捕捉/比较匹配标志A
[为“0”的条件]读后写“0” ^{注2,3} 。[为“1”的条件] 参照“表8-3: 各标志为“1”的条件”。	

注1: 在定时器B的计数值从“FFFFH”变为“0000H”时, TBOVF位变为“1”。另外, 根据TBCR寄存器的TBCCLR0位和TBCCLR1位的设置, 如果在运行过程中因发生输入捕捉或者比较匹配而使定时器B的计数值从“FFFFH”变为“0000H”, TBOVF位也变为“1”。

注2: 写的结果如下:

当写“1”时, 此位不变。

在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。

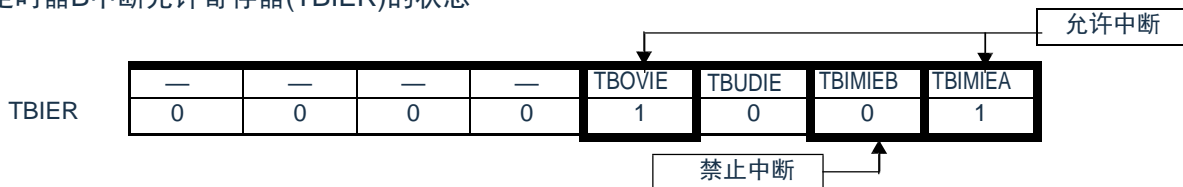
在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

但是, 要将定时器B的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器B中断允许寄存器(TBIER)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。

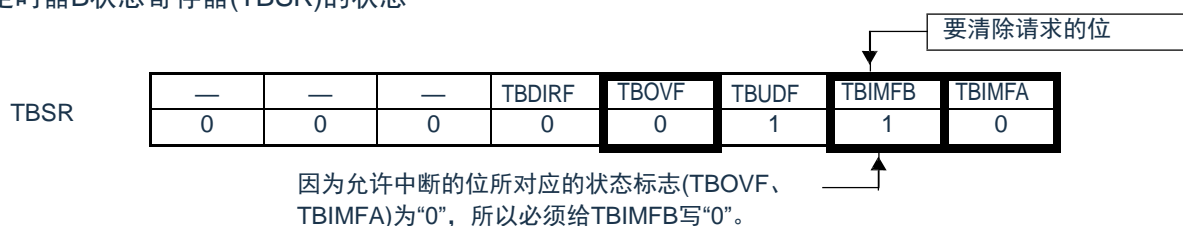
- (a) 必须在将定时器B中断允许寄存器(TBIER)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- (b) 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

(例)在TBIMIEA和TBOVIE为允许中断而TBIMIEB为禁止中断的状态下清除TBIMFB的情况

- 定时器B中断允许寄存器(TBIER)的状态



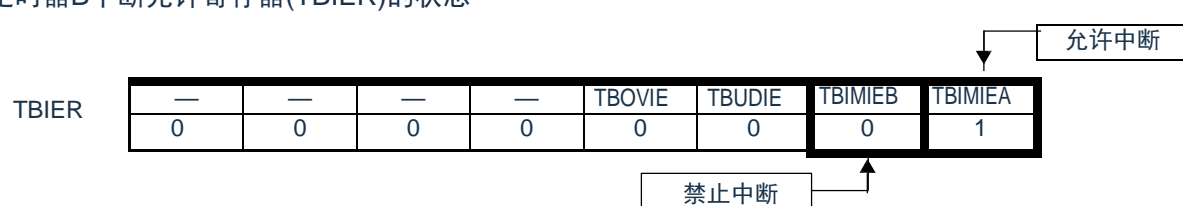
- 定时器B状态寄存器(TBSR)的状态



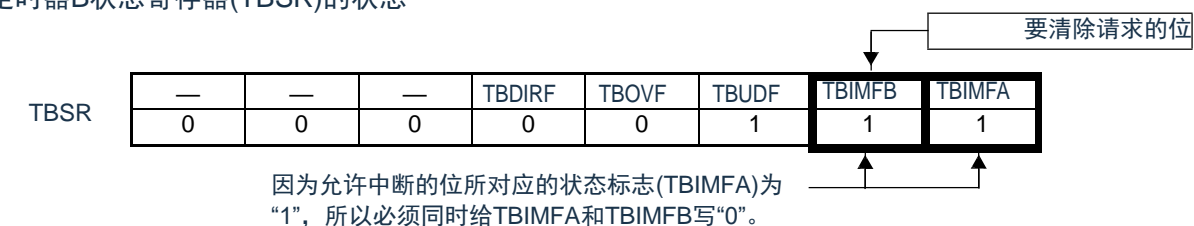
- (c) 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(例)在TBIMIEA为允许中断而TBIMIEB为禁止中断的状态下清除TBIMFB的情况

- 定时器B中断允许寄存器(TBIER)的状态



- 定时器B状态寄存器(TBSR)的状态



当使用DMA时，TBIMFA位和TBIMFB位在DMA传送结束后变为“1”。

表8-3: 各标志为“1”的条件

位符号	定时器模式 ^{注1}		PWM模式
	输入捕捉功能	输出比较功能	
TBOVF	当TB发生上溢时		
TBUDF	当TB发生下溢时(只限于相位计数模式)		
TBIMFB	TBIO1引脚的输入边沿 ^{注2}	当TB和TBGRB的值相同时	
TBIMFA	TBIO0引脚的输入边沿 ^{注2}	当TB和TBGRA的值相同时	

注1: 相位计数模式是定时器B计数寄存器的计数方法，能通过设置使用上述定时器模式和PWM模式。

注2: 这是通过TBIOR寄存器的TBIOj0位和TBIOj1位(j=0、1)选择的边沿。

8.3.7 定时器BI/O控制寄存器(TBIOR)

图8-8: 定时器BI/O控制寄存器(TBIOR)的格式

地址: 40042655H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TBIOR	TBBUFB	TBIOB2	TBIOB1	TBIOB0	TBBUFA	TBIOA2	TBIOA1	TBIOA0

TBBUFB	TBGRD寄存器功能的选择
0	不用作TBGRB寄存器的缓冲寄存器。
1	用作TBGRB寄存器的缓冲寄存器。

TBIOB2	TBGRB模式的选择 ^{注1,2}
0	输出比较功能
1	输入捕捉功能

TBIOB1	TBIOB0	TBGRB控制
0	0	禁止比较匹配的引脚输出。
0	1	输出“L”电平。
1	0	输出“H”电平。
1	1	进行交替输出。

通过输出比较功能进行TB寄存器和TBGRB寄存器的比较匹配输出。

TBIOB1	TBIOB0	TBGRB控制
0	0	TBIO1的上升沿
0	1	TBIO1的下降沿
1	0	TBIO1的双边沿
上述以外		禁止设置。

通过输入捕捉功能将TB寄存器的内容捕捉到TBGRB。

TBBUFA	TBGRG寄存器功能的选择
0	不用作TBGRA寄存器的缓冲寄存器。
1	用作TBGRA寄存器的缓冲寄存器。

TBIOA2	TBGRA模式的选择 ^{注1,2}
0	输出比较功能
1	输入捕捉功能

TBIOA1	TBIOA0	TBGRA控制
0	0	禁止比较匹配的引脚输出。
0	1	输出“L”电平。
1	0	输出“H”电平。
1	1	进行交替输出。
通过输出比较功能进行TB寄存器和TBGRA寄存器的比较匹配输出。		

TBIOA1	TBIOA0	TBGRA控制
0	0	TBIO0的上升沿
0	1	TBIO0的下降沿
1	0	TBIO0的双边沿
上述以外		禁止设置。
通过输入捕捉功能将TB寄存器的内容捕捉到TBGRA。		

注1：当TBIOj2位(j=A、B)为“1”(输入捕捉功能)时，TBGRj寄存器用作输入捕捉寄存器。

注2：当TBIOj2位(j=A、B)为“0”(输出比较功能)时，TBGRj寄存器用作比较匹配寄存器。在复位后设置TBIOj0位和TBIOj1位，并且在发生第1次比较匹配前，从TBIOj引脚输出以下电平：

当TBIOj1、TBIOj0=01B时，输出“H”电平。

当TBIOj1、TBIOj0=10B时，输出“L”电平。

当TBIOj1、TBIOj0=11B时，输出“L”电平。

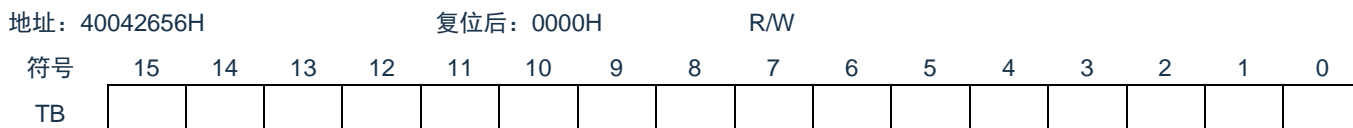
此TBIOR寄存器在定时器模式中控制输入/输出的引脚。在PWM模式中无效。必须在停止计数的状态(TBMR寄存器的TBSTART=0)下设定TBIOR寄存器。

8.3.8 定时器B计数器(TB)

TB寄存器通过16位内部总线和CPU连接，因此必须以16位为单位进行存取。TB寄存器能进行递增计数、自由运行、周期计数或者外部事件计数。能通过和对应的TBGRA寄存器、TBGRB寄存器的比较匹配或者输入捕捉到TBGRA寄存器和TBGRB寄存器，将TB寄存器清“0000H”(计数器清除功能)。

当TB寄存器发生上溢(“FFFFH”→“0000H”)时，TBSR寄存器的TBOVF位变为“1”；当TB寄存器发生下溢(“0000H”→“FFFFH”)时，TBSR寄存器的TBUDF位变为“1”。

图8-9：定时器B计数器(TB)的格式



—	功能	设置范围
bit15~0	在相位计数模式中进行递增/递减计数，在其他模式中进行递增计数。	0000H~FFFFH

8.3.9 定时器B通用寄存器A、B、C、D

(TBGRA、TBGRB、TBGRC、TBGRD)

TBGRA寄存器和TBGRB寄存器是16位可读写寄存器，有输出比较寄存器和输入捕捉寄存器的功能。通过TBIOR寄存器进行功能的转换。

当用作输出比较寄存器时，TBGRA寄存器和TBGRB寄存器的值总是与TB寄存器的值进行比较。如果值相同(比较匹配)，TBSR寄存器的TBIMFA位或者TBIMFB位就变为“1”。能通过TBIOR寄存器设置比较匹配的输

出。
当用作输入捕捉寄存器时，在检测到外部输入捕捉信号后保存TB寄存器的值。此时，TBSR寄存器的TBIMFA位或者TBIMFB位变为“1”。通过TBIOR寄存器选择输入捕捉信号的检测边沿。

TBGRC寄存器和TBGRD寄存器也能分别用作TBGRA寄存器和TBGRB寄存器的缓冲寄存器，并且能通过TBIOR寄存器的TBBUFA位和TBBUFB位选择此功能。

例如，如果将TBGRA寄存器设置为输出比较寄存器并且将TBGRC寄存器设置为TBGRA寄存器的缓冲寄存器，就在每次发生比较匹配A时将TBGRC寄存器的值传送到TBGRA寄存器。

如果将TBGRA寄存器设置为输入捕捉寄存器并且将TBGRC寄存器设置为TBGRA寄存器的缓冲寄存器，就在发生输入捕捉时将TB寄存器和TBGRA寄存器的值分别传送到TBGRA寄存器和TBGRC寄存器。

能以16位为单位读写TBGRA、TBGRB、TBGRC、TBGRD寄存器。

图8-10: 定时器B通用寄存器A、B、C、D(TBGRA、TBGRB、TBGRC、TBGRD)的格式

地址: 40042658H(TBGRA)、

4004265AH(TBGRB)、

40042660H(TBGRC)、

40042662H(TBGRD)

复位后: FFFFH

R/W



—	功能
bit15~0	因定时器模式和功能而不同。 TBGRA、TBGRB、TBGRC、TBGRD寄存器功能如表8-4所示。

备注: i=A、B、C、D

表8-4: TBGRA、TBGRB、TBGRC、TBGRD寄存器功能

模式和功能	寄存器	设置	功能
输入捕捉	TBGRA	TBIOR(TBIOA2=1) TBMR(TBPWM=0)	输入捕捉寄存器(保存TB寄存器的值)
	TBGRB	TBIOR(TBIOB2=1) TBMR(TBPWM=0)	输入捕捉寄存器(保存TB寄存器的值)
输出比较	TBGRA	TBIOR(TBIOA2=0) TBMR(TBPWM=0)	输出比较寄存器(保存和TB寄存器的比较值, 在比较匹配时TBIO0输出设置的值)
	TBGRB	TBIOR(TBIOB2=0) TBMR(TBPWM=0)	输出比较寄存器(保存和TB寄存器的比较值, 在比较匹配时TBIO1输出设置的值)
PWM	TBGRA	TBMR(TBPWM=1)	输出比较寄存器(在比较匹配时TBIO0输出“H”电平)
	TBGRB		输出比较寄存器(在比较匹配时TBIO0输出“L”电平)
共同	TBGRC	TBIOR(TBBUFA=0)	不使用。
	TBGRD	TBIOR(TBBUFB=0)	不使用。
	TBGRC	TBIOR(TBBUFA=1)	TBGRC的缓冲寄存器(和TBGRA进行传送) •当TBIOA2=1时 通过输入捕捉信号, 从TBGRA取前一次的输入捕捉值。 •当TBIOA2=0时 通过TB和TBGRA的比较匹配, 将下一个比较期待值传送到TBGRA。
	TBGRD	TBIOR(TBBUFB=1)	TBGRB的缓冲寄存器(和TBGRB进行传送) •当TBIOB2=1时 通过输入捕捉信号, 从TBGRB取前一次的输入捕捉值。 •当TBIOB2=0时 通过TB和TBGRB的比较匹配, 将下一个比较期待值传送到TBGRB。

注意: 如果将TBCR寄存器的TBTCK2~TBTCK0位置“000B”(F_{CLK})并且将比较值置“0000H”, 就只在开始计数后立即向DMA和EVENTC产生1次请求信号。如果比较值大于等于“0001H”, 就在每次比较匹配时产生请求信号。

8.3.10 端口寄存器和端口模式寄存器

在将定时器输出引脚的复用端口用作定时器的输出时，必须将各端口对应的端口模式寄存器(PMxx)的位和端口寄存器(Pxx)的位置“0”。

(例)P50/TBIO0用作定时器输出的情况将端口模式寄存器5的PM50位置“0”。将端口寄存器5的P50位置“0”。

在将定时器输入引脚的复用端口用作定时器的输入时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”。此时，端口寄存器(Pxx)的位可以是“0”或者“1”。

(例)P50/TBIO0用作定时器输入的情况

将端口模式寄存器5的PM50位置“1”。

将端口寄存器5的P50位置“0”或者“1”。

详细内容请参照“2.3.1 端口模式寄存器(PMxx)”、“2.3.2 端口寄存器(Pxx)”和“2.3.6 端口模式控制寄存器(PMCxx)”。

设置的端口模式寄存器 (PMxx)、端口寄存器 (Pxx)和端口模式控制寄存器 (PMCxx)因产品而不同。详细内容请参照“2.5 使用复用功能时的寄存器设置”。

8.4 定时器B的运行

8.4.1 有关多个模式和功能的共同事项

(1) 计数源

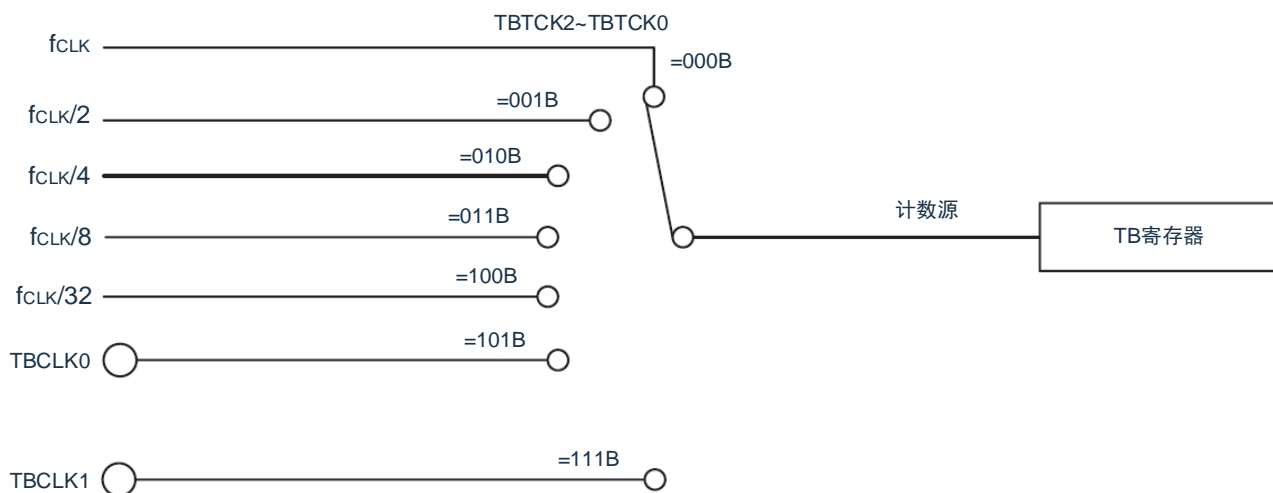
计数源的选择和框图分别如表8-5和图8-11所示。

在选择相位计数模式时，TBCR寄存器的TBTCK0~TBTCK2位、TBCKEG0位和TBCKEG1位的设置无效。

表8-5：计数源的选择

计数源	选择方法
F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$	通过TBCR寄存器的TBTCK0~TBTCK2位选择计数源。
TBCLK0引脚和TBCLK1引脚的外部输入信号	TBCR寄存器的TBTCK2~TBTCK0位为“101B”(TBCLK0输入)或者“111B”(TBCLK1输入)。通过TBCR寄存器的TBCKEG0位和TBCKEG1位选择有效边沿。端口模式寄存器的对应位为“1”(输入模式)。

图8-11：计数源的框图



备注 TBTCK2~TBTCK0：TBCR寄存器的位

TBCLKj引脚(j=0、1)输入的外部时钟的脉宽必须至少为3个定时器B的运行时钟(F_{CLK})周期。

(2) 缓冲器运行

能通过TBIOR寄存器的TBBUFA位和TBBUFB位，将TBGRC寄存器和TBGRD寄存器分别设置为TBGRA寄存器和TBGRB寄存器的缓冲寄存器。

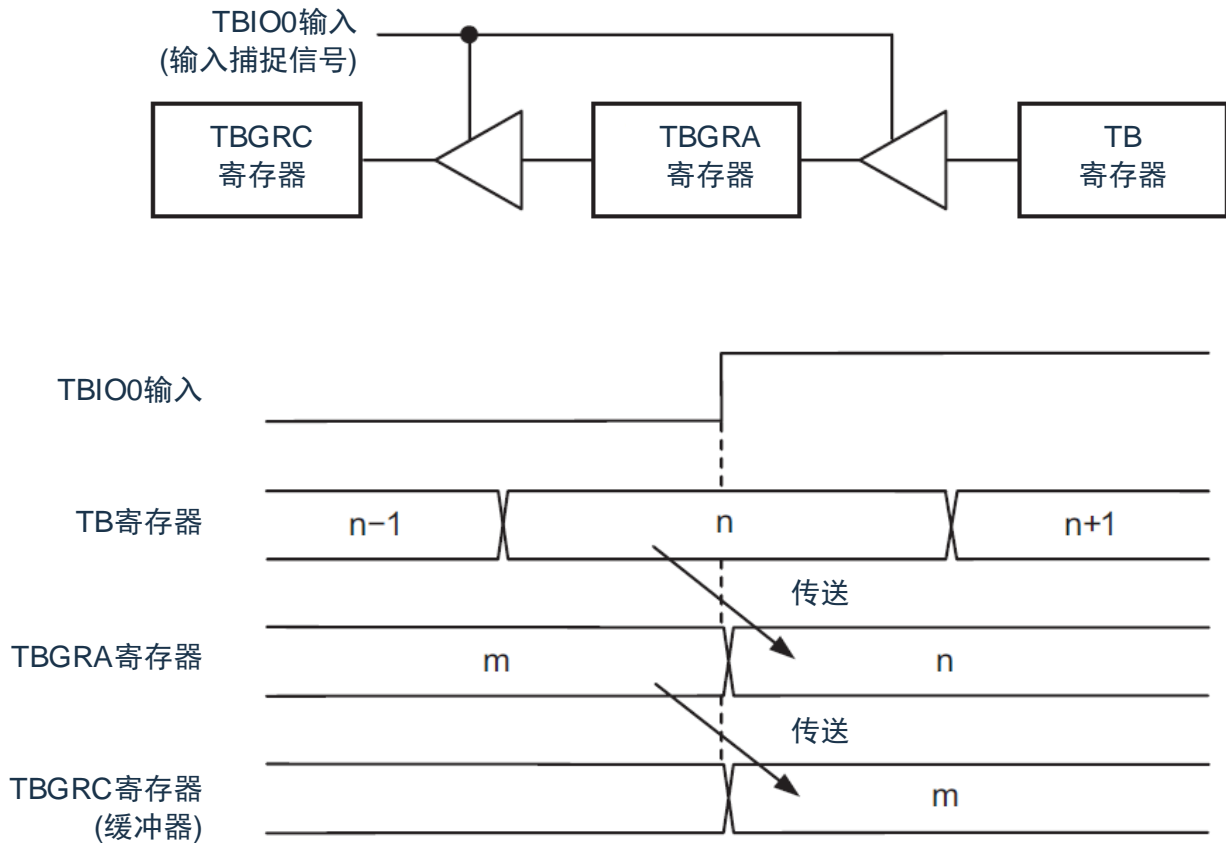
- TBGRA的缓冲寄存器：TBGRC寄存器
- TBGRB的缓冲寄存器：TBGRD寄存器缓冲器运行因定时器模式而不同。

各模式的缓冲器运行如表8-6所示，输入捕捉功能和输出比较功能的缓冲器运行分别如图8-12和图8-13所示。

表8-6：各模式的缓冲器运行

功能和模式	传送时序	传送的寄存器
输入捕捉功能	输入捕捉信号的输入	将TBGRA(TBGRB)寄存器的内容传送到缓冲寄存器。
输出比较功能	TB寄存器和TBGRA(TBGRB)寄存器的比较匹配	将缓冲寄存器的内容传送到TBGRA(TBGRB)寄存器。
PWM模式		

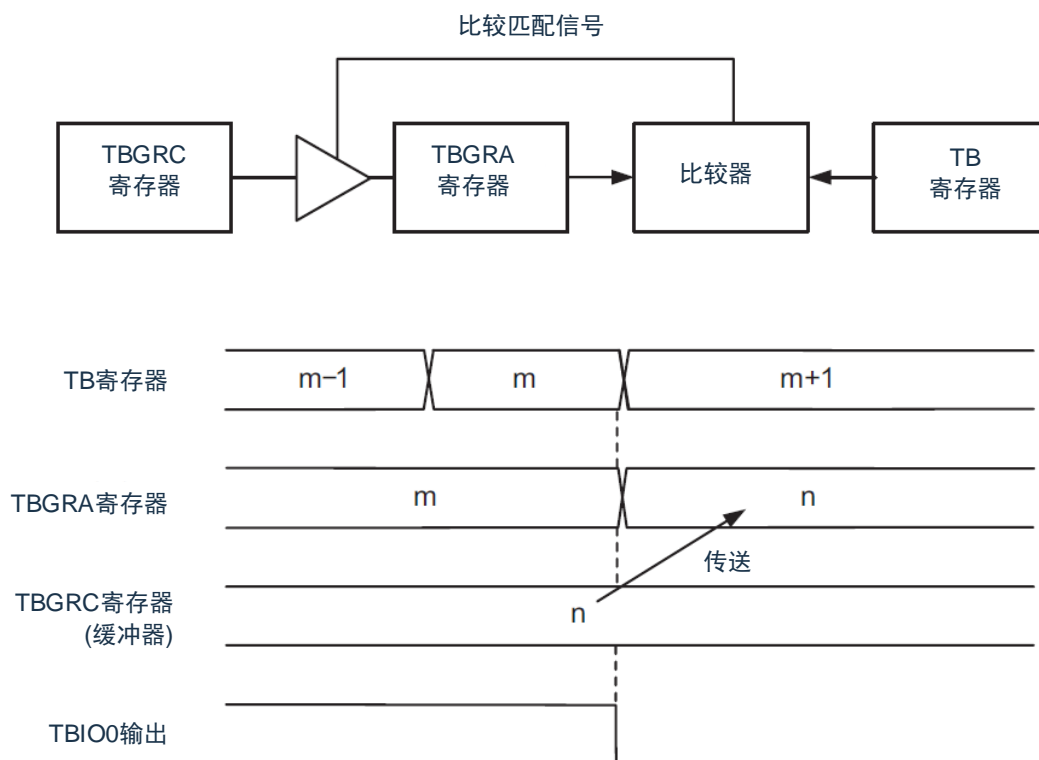
图8-12：输入捕捉功能的缓冲器运行



上图的条件如下：

- TBIOR寄存器的TBBUFA位为“1” (TBGRC寄存器为TBGRA寄存器的缓冲寄存器)。
- TBIOR寄存器的TBIOA2~TBIOA0位为“100B” (在上升沿输入捕捉)。

图8-13: 输出比较功能的缓冲器运行



上图的条件如下:

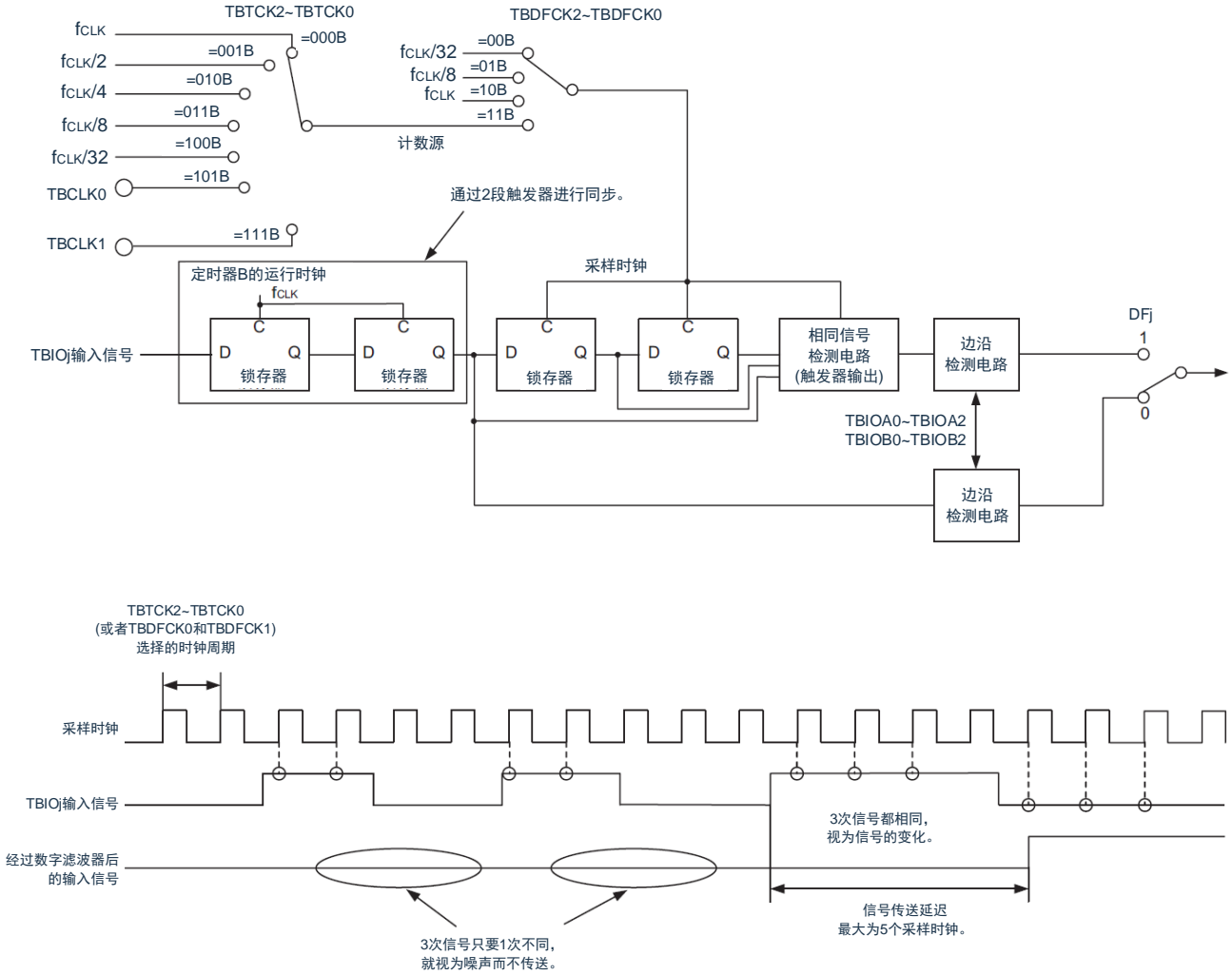
- TBIOR寄存器的TBBUFA位为“1” (TBGRC寄存器为TBGRA寄存器的缓冲寄存器)。
- TBIOR寄存器的TBIOA2~TBIOA0位为“001B” (在比较匹配时, 输出“L”电平)。

(3) 数字滤波器

对TBIOj输入(j=0、1)进行采样，如果信号3次相同，就视为电平已确定。必须通过TBMR寄存器选择数字滤波器的功能和采样时钟。

数字滤波器的框图如图8-14所示。

图8-14：数字滤波器的框图



备注 j=0, 1

TBCK2-TBCK0: TBMR寄存器的位
 TBDFCK0, TBDFCK1, TBDFCK2: TBMR寄存器的位
 TBIOA0-TBIOA2, TBIOB0-TBIOB2: TBIO寄存器的位

(4) 从EVENTC输入的事件

通过EVENTC输入的事件，定时器B进行输入捕捉运行B。此时，TBSR寄存器的TBIMFB位为“1”。要使用此功能时，必须选择定时器模式/相位计数模式的输入捕捉功能，并且将TBMR寄存器的TBELCICE位置“1”。在其他模式(定时器模式/相位计数模式的输出比较功能和PWM模式)中，此功能无效。

设置步骤

- (a) 将EVENTC事件链接目标设置为定时器B。
- (b) 将TBMR寄存器的TBELCICE位置“1”。

(5) 向EVENTC输出的事件

通过TBIMFA位和TBIMFB位向EVENTC输出的事件分别如表8-7和表8-8所示。

表8-7：通过TBIMFA位向EVENTC输出的事件

功能和模式	EVENTC源
输入捕捉功能 (TBPWM=0、TBIO2=1)	通过TBIOA0位和TBIOA1位设置的TBIO0边沿检测
输出比较功能 (TBPWM=0、TBIO2=0)	TB寄存器和TBGRA寄存器的比较匹配
PWM模式(TBPWM=1)	TB寄存器和TBGRA寄存器的比较匹配

备注：TBPWM：TBMR寄存器的位

TBIOA0、TBIOA1、TBIOA2：TBIOR寄存器的位

表8-8：通过TBIMFB位向EVENTC输出的事件

功能和模式	EVENTC源
输入捕捉功能 (TBPWM=0、TBIO12=1)	通过TBIOB0位和TBIOB1位设置的TBIO1边沿检测
输出比较功能 (TBPWM=0、TBIO12=0)	TB寄存器和TBGRB寄存器的比较匹配
PWM模式(TBPWM=1)	TB寄存器和TBGRB寄存器的比较匹配

备注：TBPWM：TBMR寄存器的位

TBIOB0、TBIOB1、TBIOB2：TBIOR寄存器的位

8.4.2 定时器模式(输入捕捉功能)

能在检测到输入捕捉/输出比较引脚(TBIO0、TBIO1)的输入边沿后，将TB寄存器的值传送到TBGRA寄存器和TBGRB寄存器。能从上升沿、下降沿和双边沿中选择检测边沿。能通过使用输入捕捉功能，测量脉宽和脉冲周期。

输入捕捉功能的规格如表8-9所示。

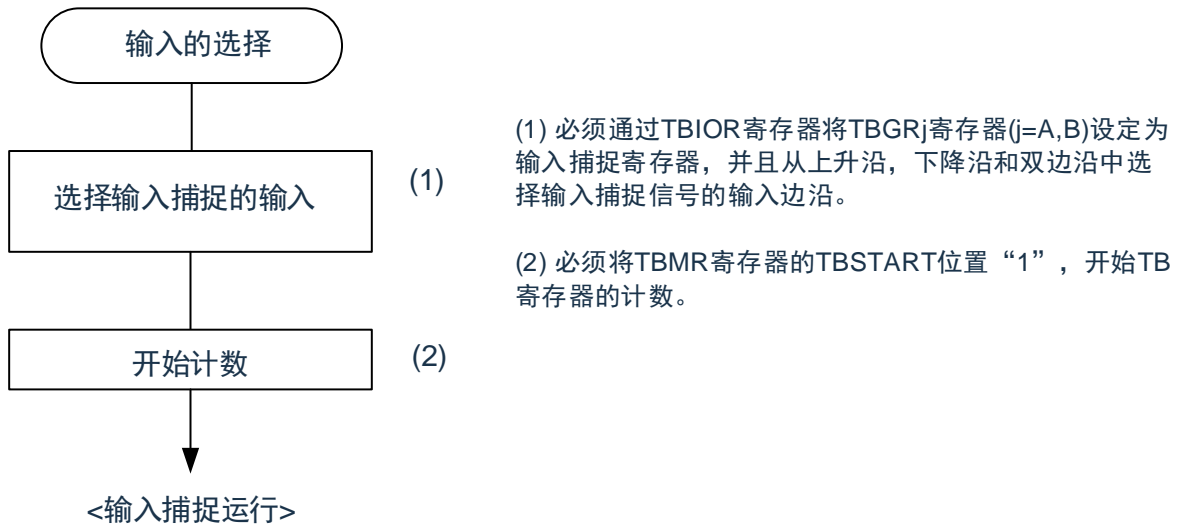
表8-9：输入捕捉功能的规格

项目	规格
计数源	F _{CLK} 、F _{CLK} /2、F _{CLK} /4、F _{CLK} /8、F _{CLK} /32 TBCLK0引脚和TBCLK1引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	当TBCR寄存器的TBCCLR1位和TBCCLR0位为“00B”(自由运行)时 1/fk × 65536fk：计数源的频率
计数开始条件	给TBMR寄存器的TBSTART位写“1”(开始计数)。
计数停止条件	给TBMR寄存器的TBSTART位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> •输入捕捉(TBIO0引脚和TBIO1引脚输入的有效边沿) •TB寄存器的上溢
TBIO0引脚和TBIO1引脚的功能	I/O端口或者输入捕捉的输入(按引脚进行选择)
TBCLK0引脚和TBCLK1引脚的功能	I/O端口或者外部时钟输入
读定时器	如果读TB寄存器，就能读到计数值。
写定时器	能写TB寄存器。
选择功能	<ul style="list-style-type: none"> •输入捕捉的输入引脚的选择 TBIO0引脚和TBIO1引脚中的1个或者2个引脚 •输入捕捉的输入有效边沿的选择 上升沿、下降沿或者双边沿 •将TB寄存器置“0000H”的时序 上溢或者输入捕捉 •缓冲器运行(参照“8.4.1(2)缓冲器运行”) •数字滤波器(参照“8.4.1(3)数字滤波器”) •通过EVENTC的事件输入信号(输入捕捉)进行的输入捕捉运行

(1) 输入捕捉运行的设置步骤例子

输入捕捉运行的设置步骤例子如图8-15所示。

图8-15: 输入捕捉运行的设置步骤例子

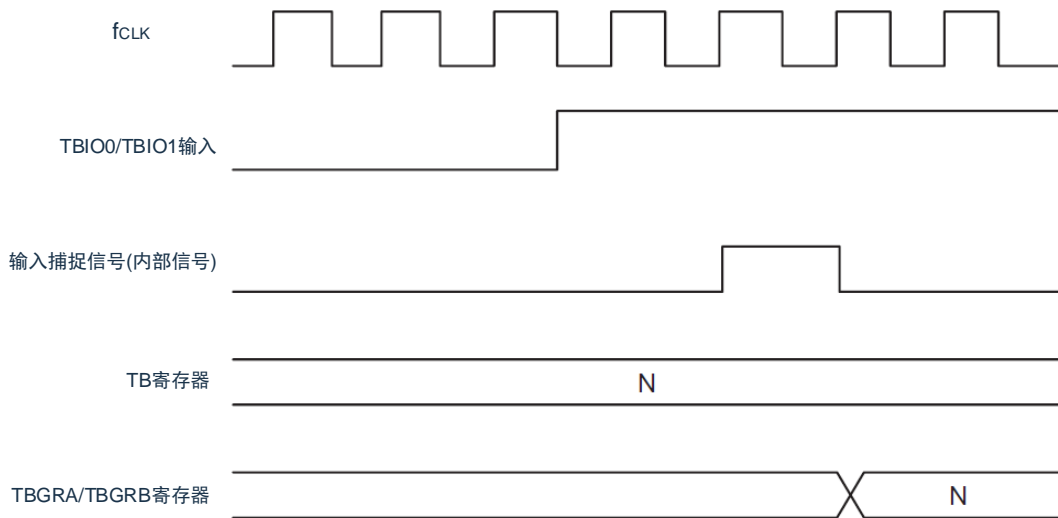


(2) 输入捕捉信号的时序

输入捕捉的输入能通过设置TBIOR寄存器选择上升沿、下降沿或者双边沿。输入捕捉的输入信号时序如图8-16所示。

在单边沿的情况下, 输入捕捉的输入信号脉宽必须至少为1.5个 F_{CLK} ; 在双边沿的情况下, 输入捕捉的输入信号脉宽必须至少为2.5个 F_{CLK} 。

图8-16: 输入捕捉的输入信号时序



(3) 运行例子

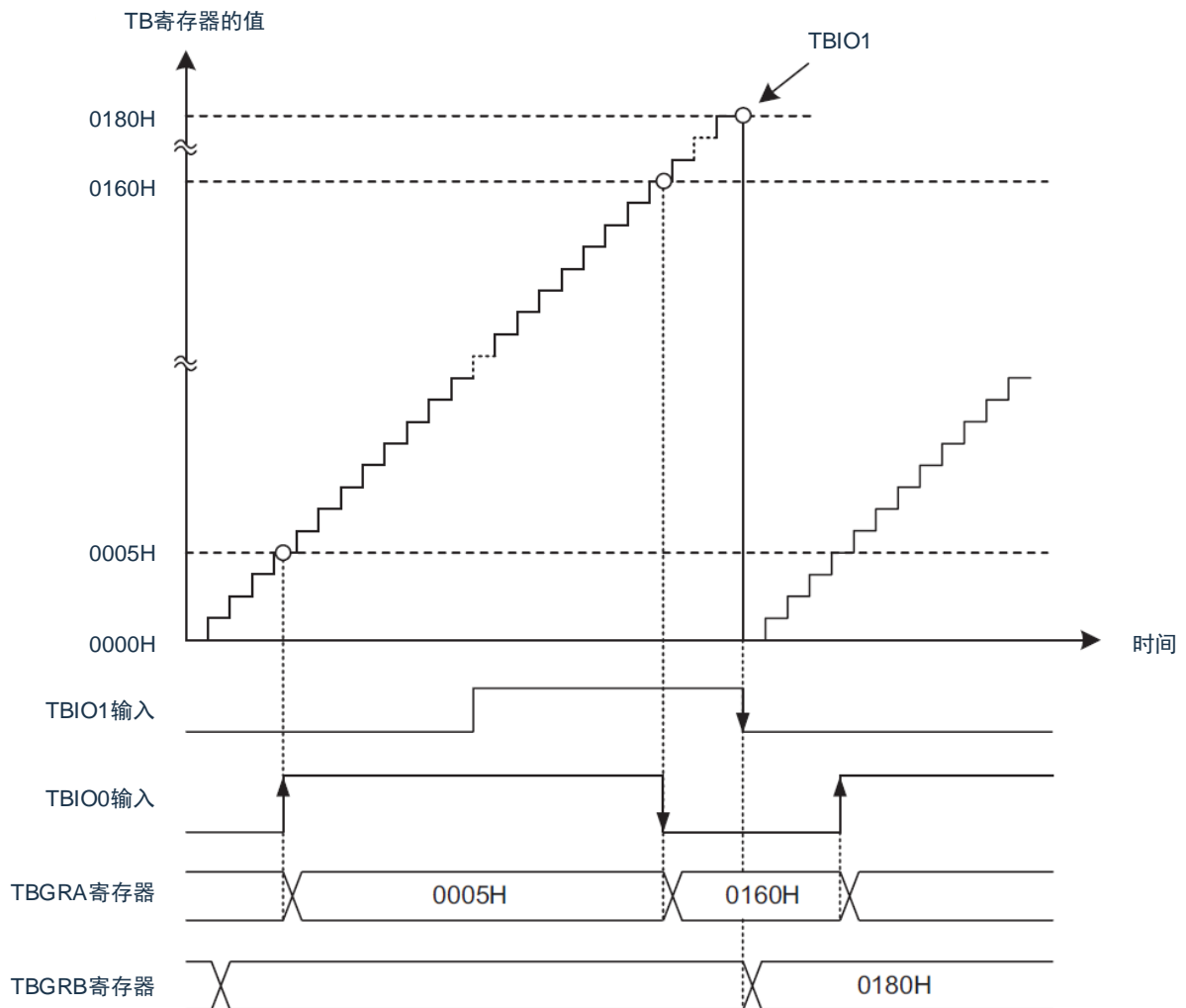
输入捕捉的运行例子如图8-17所示。

在此例子中，选择上升沿/下降沿的双边沿作为TBIO0引脚的输入捕捉的输入边沿，选择下降沿作为TBIO1引脚的输入捕捉的输入边沿，并且在TBGRB寄存器的输入捕捉时清除TB寄存器的计数器。

(a) 必须通过TBIOR寄存器将TBGRA寄存器和TBGRB寄存器设置为输入捕捉寄存器，并且从上升沿、下降沿和双边沿中选择输入捕捉信号的输入边沿。

(b) 必须将TBMR寄存器的TBSTART位置“1”，开始TB寄存器的计数。

图8-17：输入捕捉的运行例子



能通过设置TBCR寄存器的TBCCLR0位和TBCCLR1位，在发生输入捕捉A或者输入捕捉B时清除计数。图8-17是将TBCCLR1位和TBCCLR0位置“10B”时的运行例子。如果设置为通过运行过程中的输入捕捉进行计数清除并且在定时器的计数值为“FFFFH”时进行输入捕捉，就根据计数源和输入捕捉的运行时序，TBIMFA位和TBIMFB位的中断标志以及TBOVF位可能同时变为“1”。

8.4.3 定时器模式(输出比较功能)

这是检测TB寄存器的内容和TBGRA寄存器或者TBGRB寄存器的内容是否相同(比较匹配)的模式。如果内容相同,就从TBIO0引脚或者TBIO1引脚输出任意的电平。

输出比较功能的规格如表8-10所示。

表8-10: 输出比较功能的规格

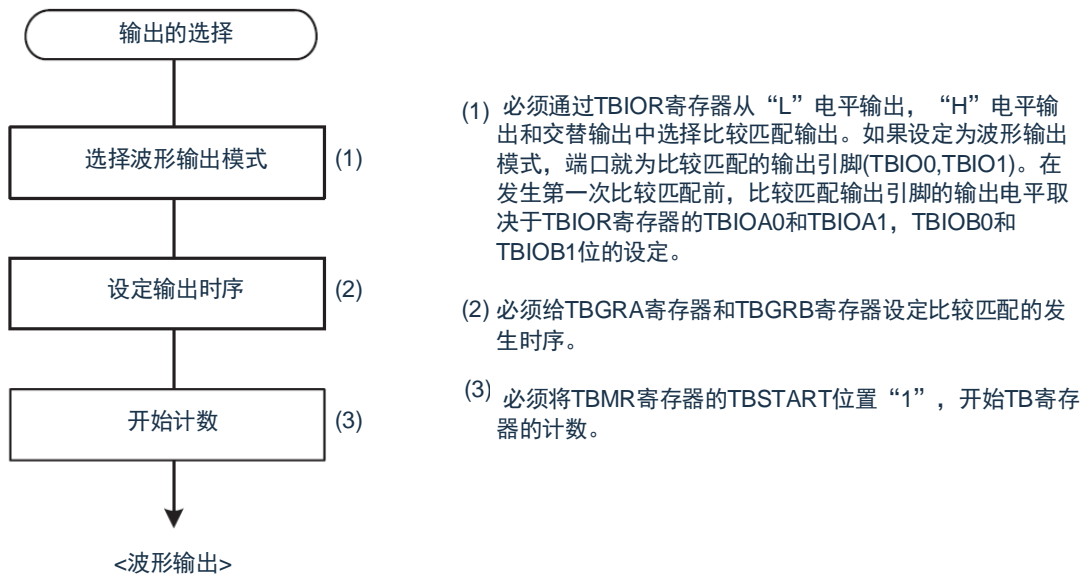
项目	规格
计数源	F _{CLK} 、F _{CLK} /2、F _{CLK} /4、F _{CLK} /8、F _{CLK} /32 TBCLKj引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	<ul style="list-style-type: none"> 当TBCR寄存器的TBCCLR1位和TBCCLR0位为“00B”(自由运行)时 1/fk × 65536fk: 计数源的频率 当TBCR寄存器的TBCCLR1位和TBCCLR0位为“01B”或者“10B”(在TBGRj比较匹配时将TB置“0000H”)时1/fk × (n+1)n: TBGRj寄存器的设置值
波形输出时序	比较匹配(TB寄存器和TBGRj寄存器的内容相同)
计数开始条件	给TBMR寄存器的TBSTART位写“1”(开始计数)。
计数停止条件	给TBMR寄存器的TBSTART位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TB寄存器和TBGRj寄存器的内容相同) TB寄存器的上溢
TBIO0引脚和TBIO1引脚的功能	I/O端口或者输出比较的输出(按引脚进行选择)
TBCLK0引脚和TBCLK1引脚的功能	I/O端口或者外部时钟输入
读定时器	如果读TB寄存器, 就能读到计数值。
写定时器	能写TB寄存器。
选择功能	<ul style="list-style-type: none"> 输出比较的输出引脚的选择 TBIO0引脚和TBIO1引脚中的1个或者2个引脚 比较匹配时的输出电平的选择 “L”电平输出、“H”电平输出或者电平反相输出 将TB寄存器置“0000H”的时序 上溢或者TBGRj寄存器的比较匹配 缓冲器运行(参照“8.4.1(2)缓冲器运行”)

备注: j=A、B

(1) 比较匹配的波形输出的设置步骤例子

比较匹配的波形输出的设置步骤如图8-18所示。

图8-18：比较匹配的波形输出的设置步骤

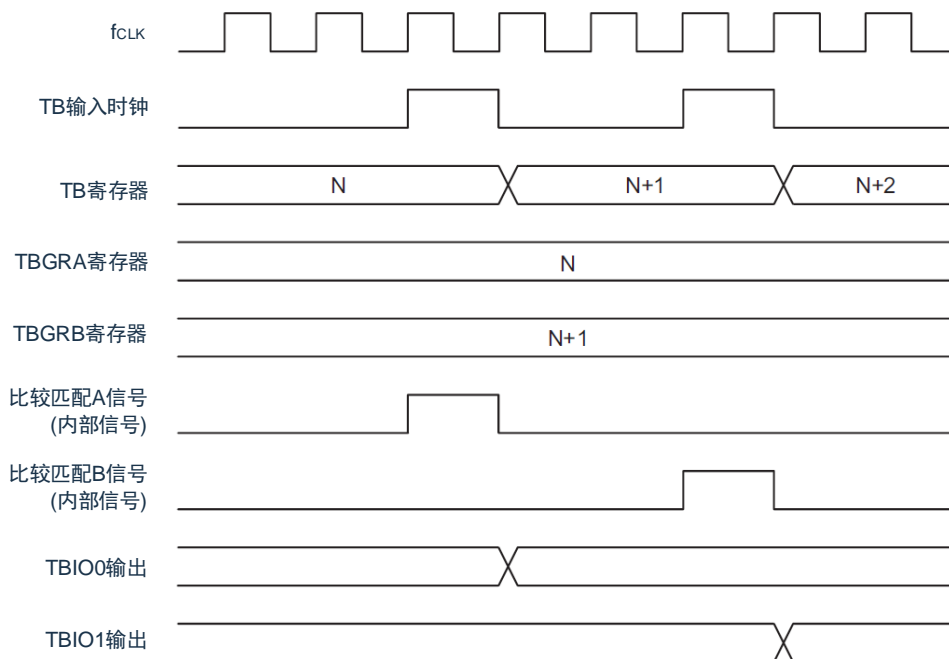


(2) 输出比较的输出时序

在TB寄存器的内容和TBGRA寄存器或者TBGRB寄存器的内容相同的最后状态(更新TB寄存器相同后的计数值时)产生比较匹配信号。在产生比较匹配信号后，从输出比较的输出引脚(TBIO0、TBIO1)输出TBIOR寄存器的输出设置值。从TB寄存器的内容和TBGRA寄存器或者TBGRB寄存器的内容相同到产生TB寄存器的输入时钟为止，不产生比较匹配信号。

输出比较的输出时序如图8-19所示。

图8-19：输出比较的输出时序

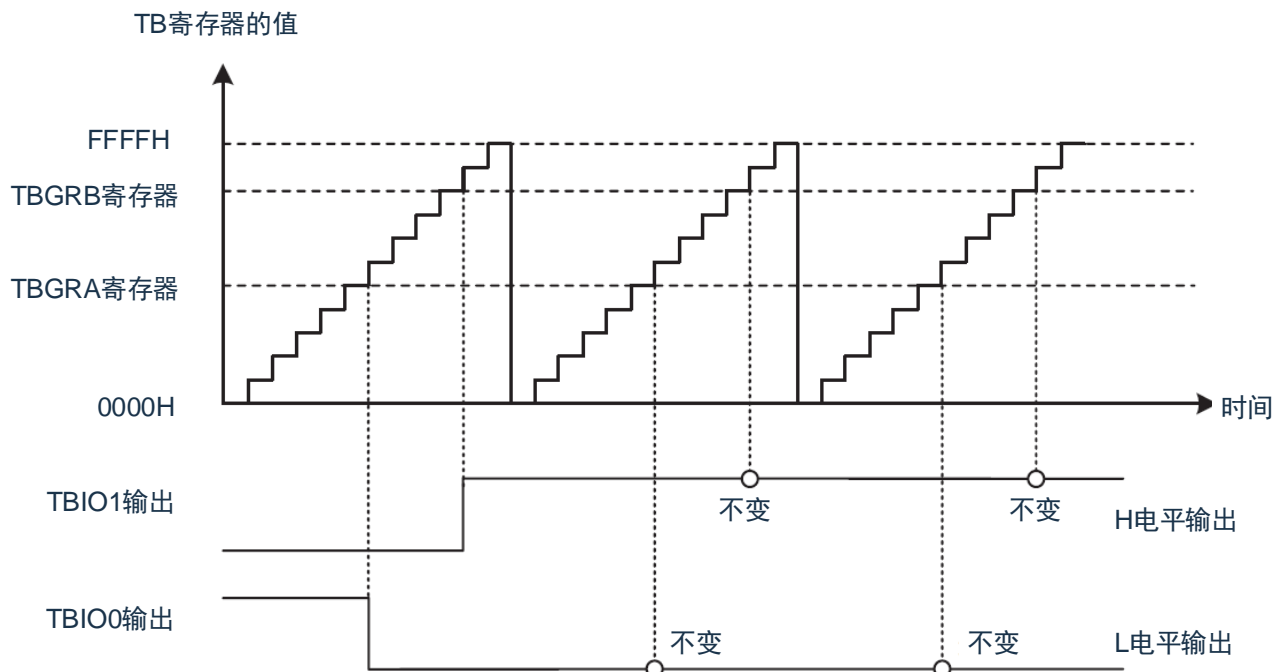


(3) 运行例子

“L”电平输出和“H”电平输出的运行例子如图8-20所示。

在此例子中，将TB寄存器设置为自由运行，并且在比较匹配A时输出“L”电平而在比较匹配B时输出“H”电平。如果设置的电平和引脚的电平相同，引脚的电平就不变。

图8-20：“L”电平输出和“H”电平输出的运行例子

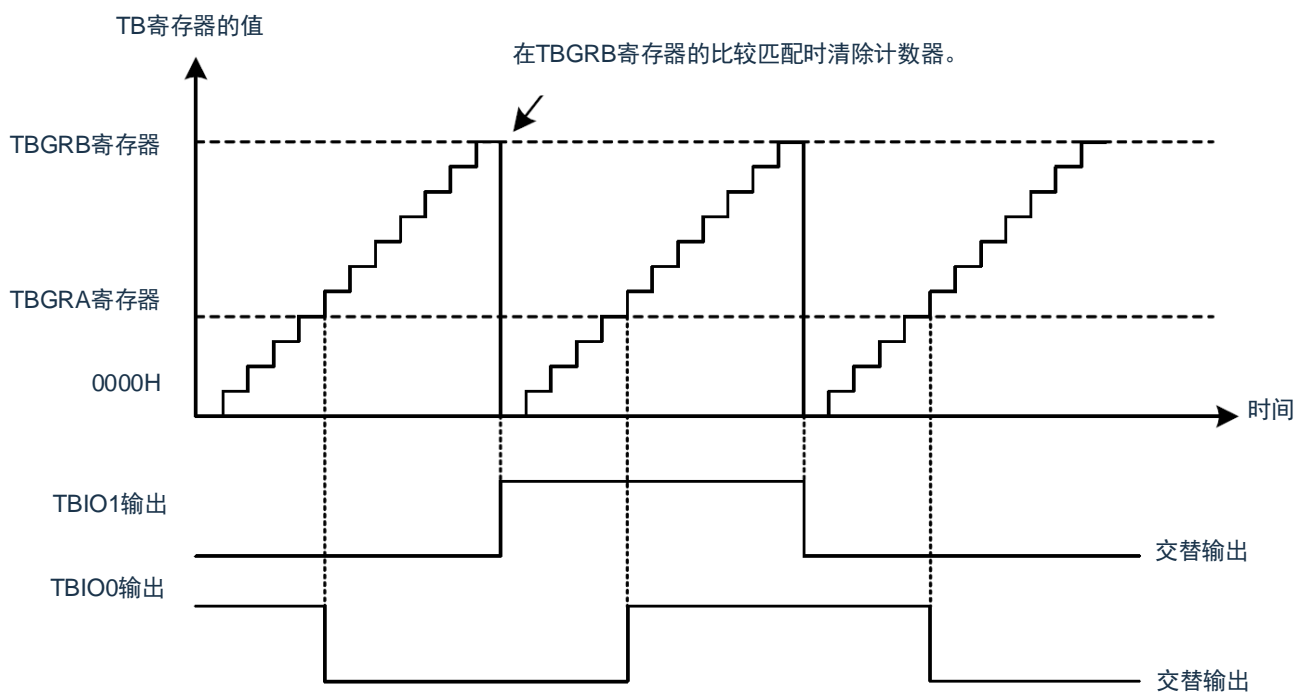


交替输出的运行例子如图8-21所示。在此例子中，将TB寄存器设置为周期计数运行(在比较匹配B时清除计数器)，并且在发生比较匹配A或者比较匹配B时都进行交替输出。

- (a) 必须通过TBIOR寄存器从“L”电平输出、“H”电平输出和交替输出中选择比较匹配输出。如果设置为波形输出模式，端口就为比较匹配的输出引脚(TBIO0、TBIO1)。
- (b) 必须给TBGRA寄存器和TBGRB寄存器设置比较匹配的发生时序。
- (c) 必须将TBMR寄存器的TBSTART位置“1”，开始TB寄存器的计数。

即使在运行过程中将TBSTART位置“0”，也不对比较匹配的输出引脚(TBIO0、TBIO1)进行初始化。要返回到初始值时，通过写TBIOR寄存器进行输出初始化(但是，只有在通过TBIOR寄存器的TBIOA0位、TBIOA1位、TBIOB0位和TBIOB1位设置为“L”电平输出或者“H”电平输出时才进行初始化)。通过设置TBCR寄存器的TBCCLR0位和TBCCLR1位，在输入捕捉/比较匹配(和TBGRA寄存器或者TBGRB寄存器相同)时对定时器B的计数值进行复位。此时，如果比较期待值为“FFFFH”，就和溢出相同从“FFFFH”变为“0000H”，并且TBOVF位为“1”。在使用定时器B的计数值和比较期待值的输出比较功能的模式中也相同。

图8-21：交替输出的运行例子



8.4.4 PWM模式

PWM模式配对使用TBGRA寄存器和TBGRB寄存器，从TBIO0输出引脚输出PWM波形。对于设置为PWM模式的输出引脚，TBIOR寄存器的输出设置无效。给TBGRA寄存器设置PWM波形的“H”电平输出时序，给TBGRB寄存器设置PWM波形的“L”电平输出时序。

能通过将TBGRA寄存器或者TBGRB寄存器的比较匹配设置为TB寄存器的计数器清除源，从TBIO0引脚输出0~100%占空比的PWM波形。

PWM模式的规格以及PWM输出引脚和寄存器的组合分别如表8-11和表8-12所示。

当TBGRA寄存器和TBGRB寄存器的设置值相同时，即使发生比较匹配，输出值也不变。

表8-11: PWM模式的规格

项目	规格
计数源	F _{CLK} 、F _{CLK} /2、F _{CLK} /4、F _{CLK} /8、F _{CLK} /32 TBCLK0, TBCLK1引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
PWM波形	<ul style="list-style-type: none"> 给TBGRA寄存器设置PWM波形的“H”电平输出时序。 给TBGRB寄存器设置PWM波形的“L”电平输出时序。
计数开始条件	给TBMR寄存器的TBSTART位写“1”(开始计数)。
计数停止条件	给TBMR寄存器的TBSTART位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TB寄存器和TBGRj寄存器的内容相同) TB寄存器的上溢
TBIO0引脚功能	PWM输出
TBIO1引脚功能	I/O端口
TBCLK0引脚和TBCLK1引脚的功能	I/O端口或者外部时钟输入
读定时器	如果读TB寄存器，就能读到计数值。
写定时器	能写TB寄存器。
选择功能	<ul style="list-style-type: none"> 将TB寄存器置“0000H”的时序上溢或者TBGRj寄存器的比较匹配 缓冲器运行(参照“8.4.1(2)缓冲器运行”)

备注: j=A、B

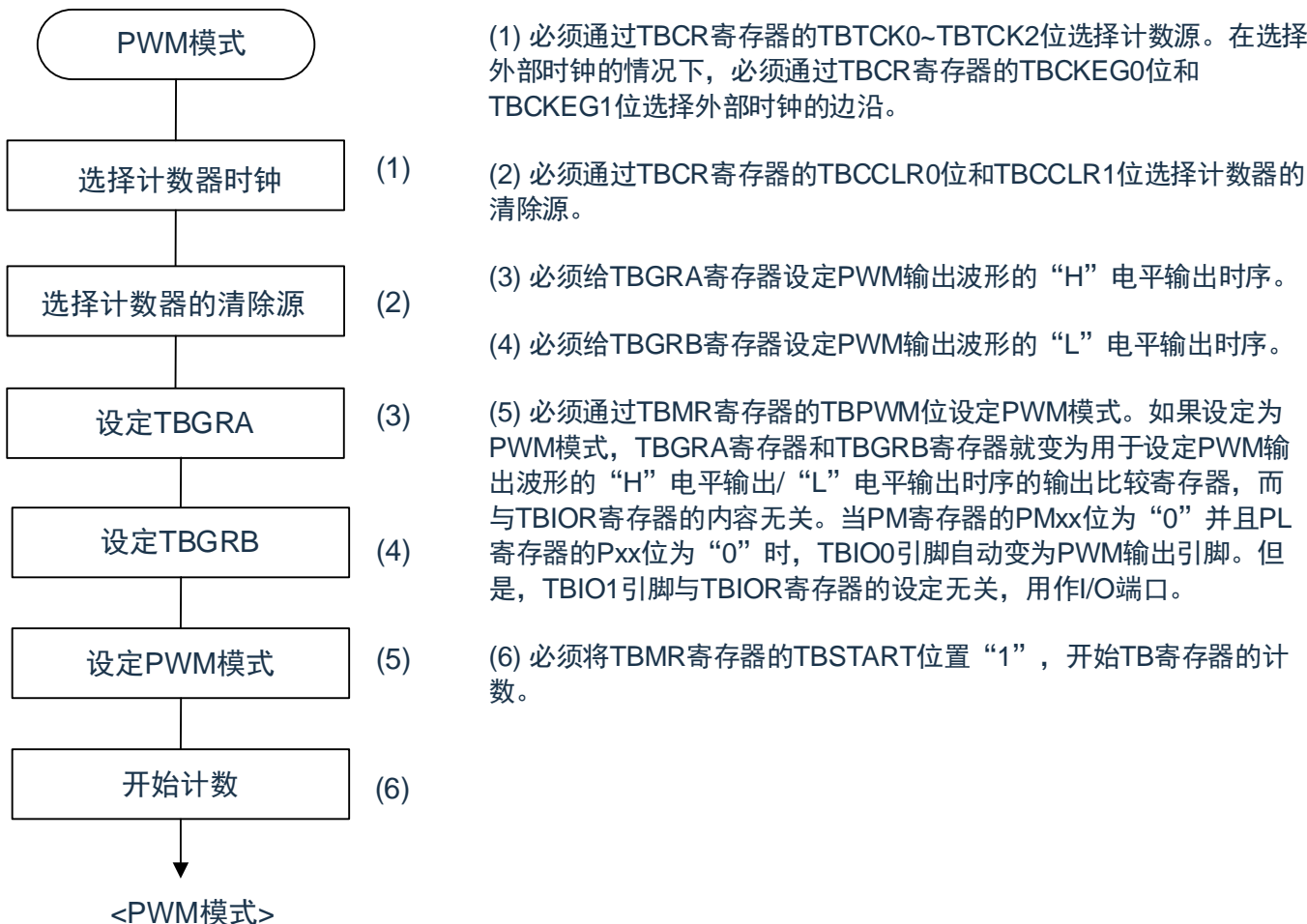
表8-12: PWM输出引脚和寄存器的组合

输出引脚	“H”电平输出	“L”电平输出
TBIO0	TBGRA	TBGRB
TBIO1	用作I/O端口。	

(1) PWM模式的设置步骤例子

PWM模式的设置步骤例子如图8-22所示。

图8-22: PWM模式的设置步骤例子



(1) 必须通过TBCR寄存器的TBTCK0~TBTCK2位选择计数源。在选择外部时钟的情况下，必须通过TBCR寄存器的TBCKEG0位和TBCKEG1位选择外部时钟的边沿。

(2) 必须通过TBCR寄存器的TBCCLR0位和TBCCLR1位选择计数器的清除源。

(3) 必须给TBGRA寄存器设定PWM输出波形的“H”电平输出时序。

(4) 必须给TBGRB寄存器设定PWM输出波形的“L”电平输出时序。

(5) 必须通过TBMR寄存器的TBPWM位设定PWM模式。如果设定为PWM模式，TBGRA寄存器和TBGRB寄存器就变为用于设定PWM输出波形的“H”电平输出/“L”电平输出时序的输出比较寄存器，而与TBIOR寄存器的内容无关。当PM寄存器的PMxx位为“0”并且PL寄存器的Pxx位为“0”时，TBIO0引脚自动变为PWM输出引脚。但是，TBIO1引脚与TBIOR寄存器的设定无关，用作I/O端口。

(6) 必须将TBMR寄存器的TBSTART位置“1”，开始TB寄存器的计数。

(2) 运行例子

PWM模式的运行例子(1)如图8-23所示。

在PM寄存器的PMxx位为“0”并且PL寄存器的Pxx位为“0”时，如果设置为PWM模式，TBIO0引脚就自动变为输出引脚，并且在TBGRA寄存器的比较匹配时输出“H”电平，在TBGRB寄存器的比较匹配时输出“L”电平。但是，TBIO1引脚与TBIOR寄存器的设置无关，用作I/O端口。

在此例子中，将TBGRA寄存器和TBGRB寄存器的比较匹配设置为TB寄存器的计数器清除源。TBIO0引脚的初始状态只取决于计数器的清除源，此对应关系如表8-13所示。

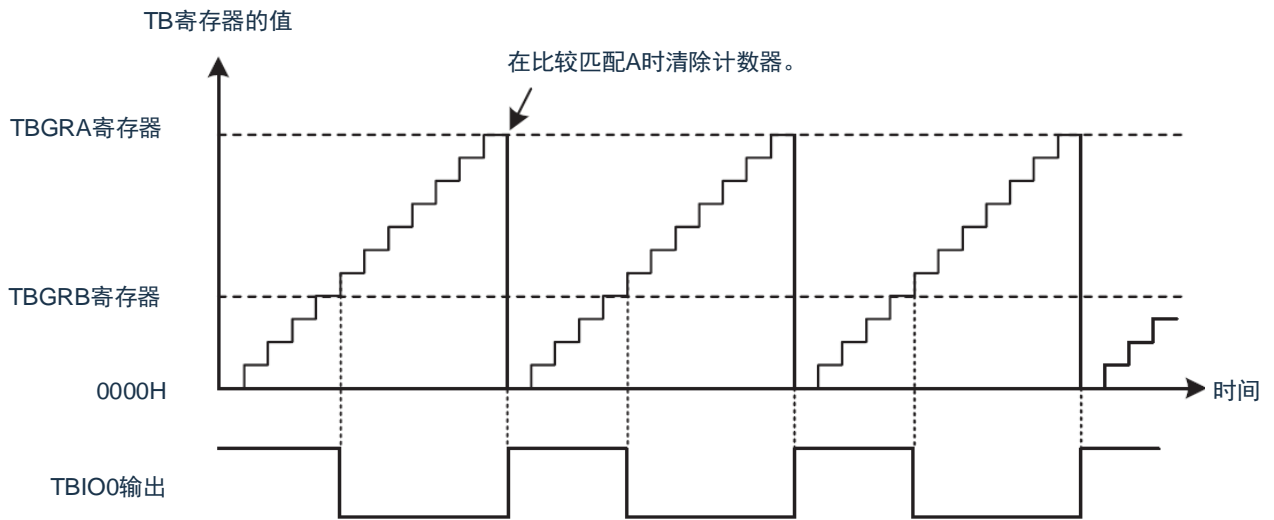
在TBMR寄存器的TBSTART位为“0”(停止计数)时进行此初始化。

表8-13: TBIO0引脚的初始状态和计数器清除源的对应关系

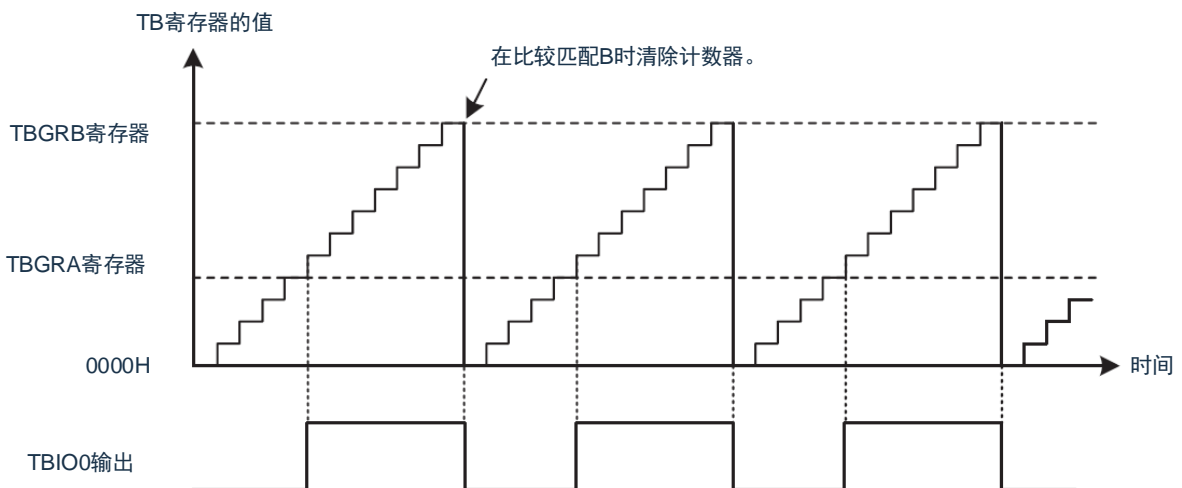
计数器清除源	TBIO0引脚的初始状态
TBGRA寄存器的比较匹配	“H”电平
TBGRB寄存器的比较匹配	“L”电平

当TBCR寄存器的TBCCLR1位和TBCCLR0位为“00B”(禁止清除)时，TBIO0引脚的初始状态为“H”电平。

图8-23: PWM模式的运行例子(1)



(a)在TBGRA寄存器的比较匹配时清除计数器。



(b)在TBGRB寄存器的比较匹配时清除计数器

在PWM模式中输出0%和100%占空比的PWM波形例子如图8-24所示。

当将TBGRB寄存器的比较匹配设置为计数器的清除源并且满足以下条件时，PWM波形的占空比为0%。

- TBGRA寄存器的设置值 > TBGRB寄存器的设置值

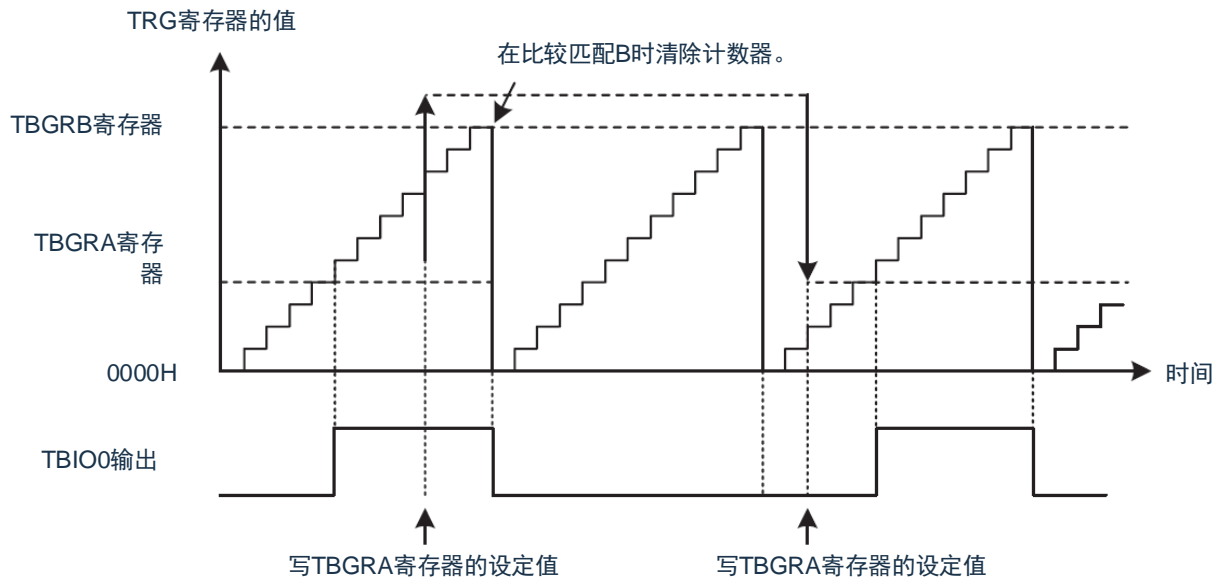
当将TBGRA寄存器的比较匹配设置为计数器的清除源并且满足以下条件时，PWM波形的占空比为100%。

- TBGRB寄存器的设置值 > TBGRA寄存器的设置值

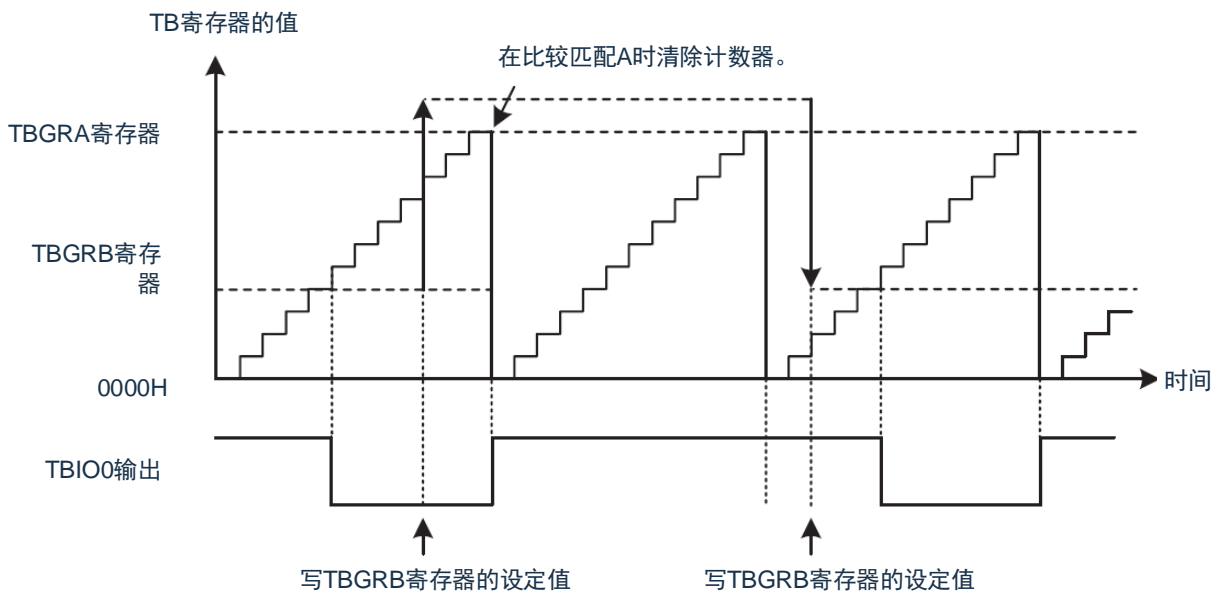
当满足以下条件时，即使发生比较匹配，输出值也不变。

- TBGRA寄存器的设置值 = TBGRB寄存器的设置值

图8-24：PWM模式的运行例子(2)



(a)占空比0%



(b)占空比100%

8.4.5 相位计数模式

相位计数模式检测2个TBCLK0引脚和TBCLK1引脚的外部输入信号的相位差，并且TB寄存器进行递增/递减计数。

在PM寄存器的PMxx位为“1”时，如果设置为相位计数模式，TBCLK0引脚和TBCLK1引脚就自动用作外部时钟输入引脚，并且TB寄存器根据TBCNTC寄存器的CNTEN0~CNTEN7位的设置进行加减计数，而与TBCR寄存器的TBTCK0~TBTCK2位、TBCKEG0位和TBCKEG1位的设置无关。但是，因为TBCR寄存器的TBCCLR0位和TBCCLR1位、TBIOR、TBIER、TBSR、TBGRA、TBGRB寄存器有效，所以能使用输入捕捉/输出比较功能、PWM输出功能和中断源。









根据CNTEN0~CNTEN7位的设置，TB寄存器在TBCLK0引脚和TBCLK1引脚的上升沿/下降沿的双边沿进行计数。

相位计数模式的规格以及TB寄存器的加减条件分别如表8-14和表8-15所示。

表8-14：相位计数模式的规格

项目	规格
计数源	TBCLK0/TBCLK1引脚的外部输入信号
计数	递增计数/递减计数
计数开始条件	给TBMR寄存器的TBSTART位写“1”(开始计数)。
计数停止条件	给TBMR寄存器的TBSTART位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> •输入捕捉(TBIO0/TBIO1输入的有效边沿) •比较匹配(TB寄存器和TBGRA/TBGRB寄存器的内容相同) •TB寄存器的上溢 •TB寄存器的下溢
TBIO0引脚功能	I/O端口、输入捕捉的输入、输出比较的输出或者PWM输出
TBIO1引脚功能	I/O端口、输入捕捉的输入或者输出比较的输出
TBCLK0引脚和TBCLK1引脚的功能	外部时钟输入
读定时器	如果读TB寄存器，就能读到计数值。
写定时器	能写TB寄存器。
选择功能	<ul style="list-style-type: none"> •计数器的加减条件的选择 通过TBCNTC寄存器的CNTEN0~CNTEN7位进行选择。 •能使用输入捕捉/输出比较功能和PWM功能。

表8-15：TB寄存器的加减条件

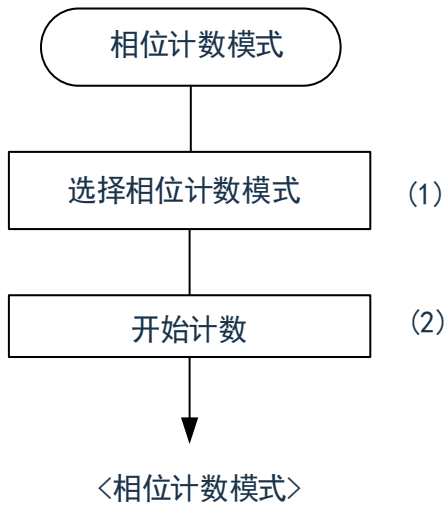
TBCLK1引脚		“H”		“L”	“H”		“L”	
TBCLK0引脚	“L”		“H”			“L”		“H”
TBCNTC寄存器的CNTEN0~CNTEN7位	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0
计数方向 ^注	+1	+1	+1	+1	-1	-1	-1	-1

注：表示TBCNTC寄存器的各位为“1”(递增计数或者递减计数)时的计数方向。为“0”(无效)时不进行计数。

(1) 相位计数模式的设置步骤例子

相位计数模式的设置步骤例子如图8-25所示。

图8-25：相位计数模式的设置步骤例子



(1) 必须将TBMR寄存器的TBMDf位置“1”，选择相位计数模式。

(2) 必须将TBMR寄存器的TBSTART位置“1”，开始TB寄存器的计数。

(2) 运行例子

相位计数模式的运行例子如图8-26~图8-29所示。

在相位计数模式中，根据TBCNTC寄存器的CNTEN0~CNTEN7位的设置，在TBCLK0引脚和TBCLK1引脚的上升沿(↑)/下降沿(↓)的双边沿进行加减计数。

图8-26：相位计数模式的运行例子1

· TBCNTC寄存器的值为“FFH”的情况

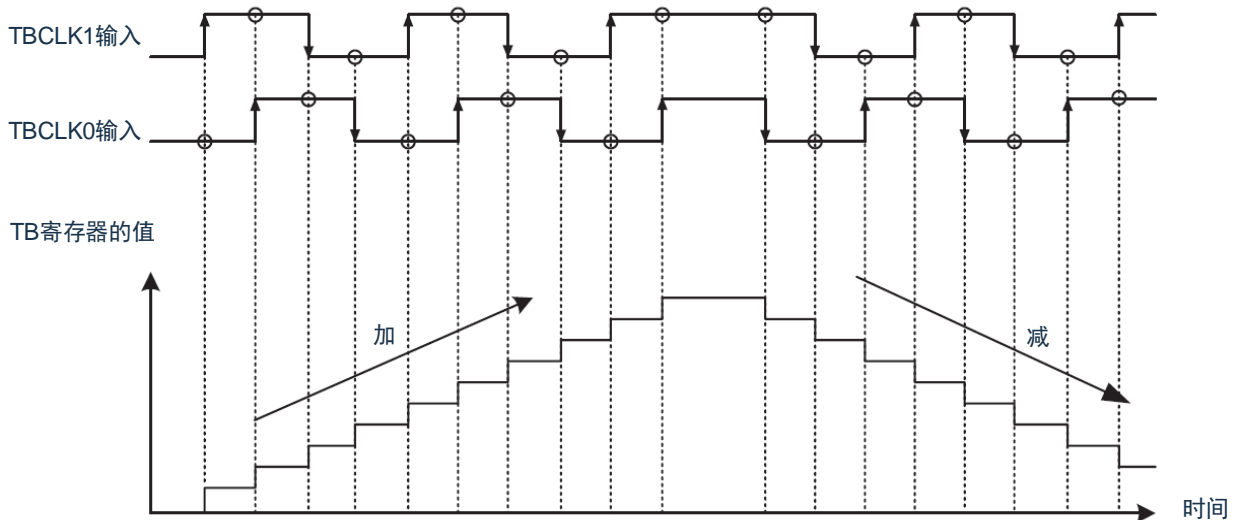


图8-27: 相位计数模式的运行例子2

· TBCNTC寄存器的值为“24H”的情况

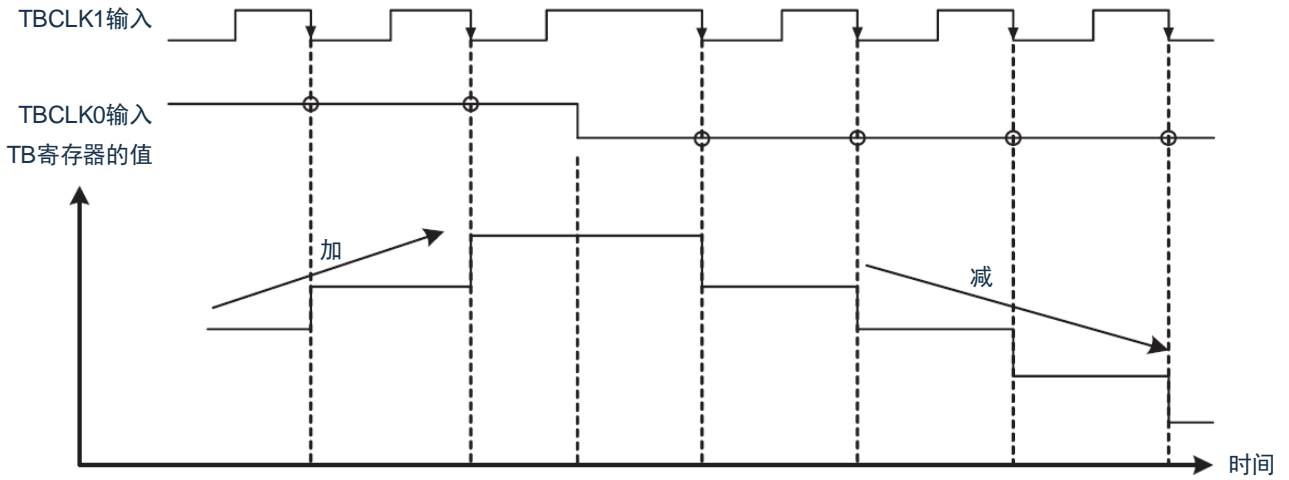


图8-28: 相位计数模式的运行例子3

· TBCNTC寄存器的值为“28H”的情况

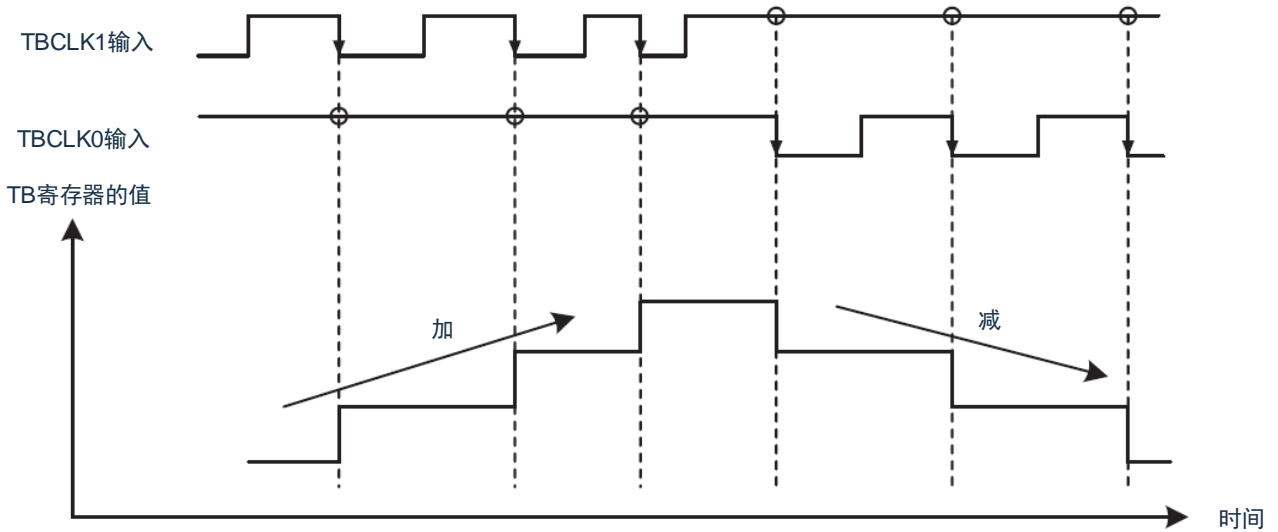
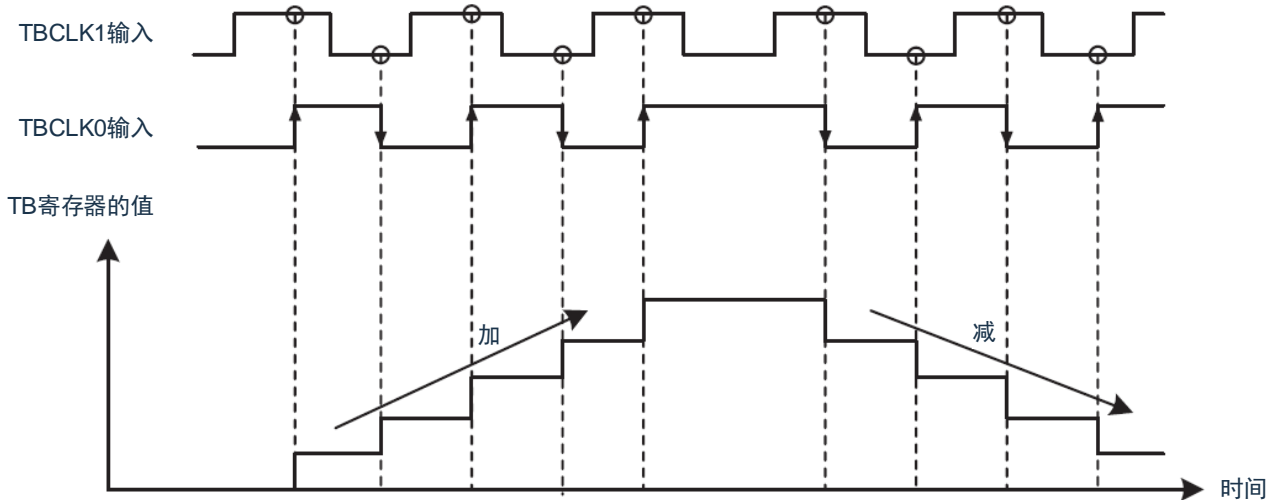


图8-29: 相位计数模式的运行例子4

· TBCNTC寄存器的值为“5AH”的情况



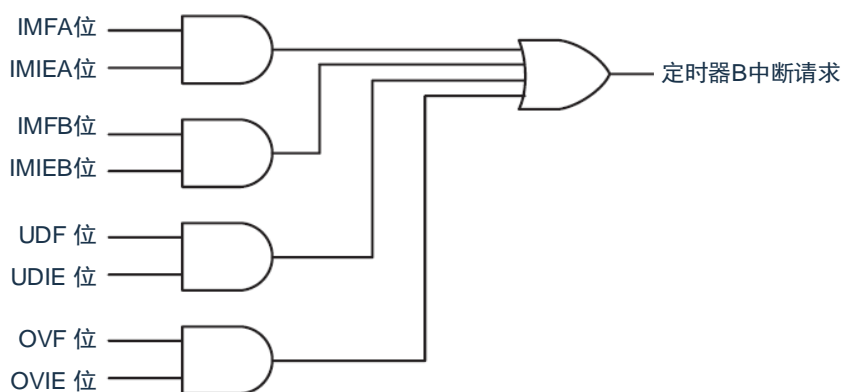
8.5 定时器B中断

定时器B从4个中断源中产生定时器B中断请求。定时器B中断的相关寄存器如表8-16所示，定时器B中断的框图如图8-30所示。

表8-16：定时器B中断的相关寄存器

	定时器B的状态寄存器	定时器B的中断允许寄存器	中断请求标志(寄存器)	中断屏蔽标志(寄存器)	优先级指定标志(寄存器)
定时器B	TBSR	TBIER	TBIF(IF2H)	TBMK(MK2H)	TBPR0(PR02H) TBPR1(PR12H)

图8-30：定时器B中断的框图



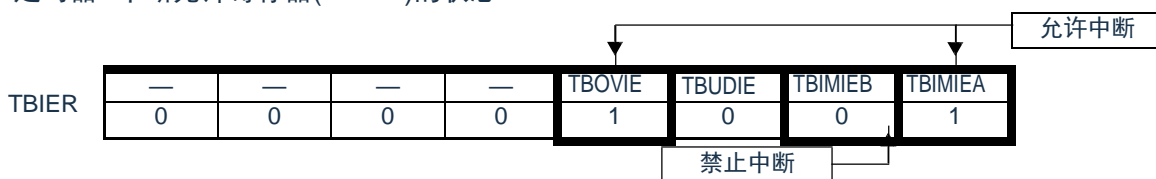
IMFA、IMFB、UDF、OVF：TBSR寄存器的位
IMIEA、IMIEB、UDIE、OVIE：TBIER寄存器的位

因为定时器B从多个中断请求源中产生1个中断请求(定时器B中断)，所以除了定时器M中断以外，和其他可屏蔽中断有以下的不同：

- 当TBSR寄存器的位为“1”并且对应的TBIER寄存器的位为“1”(允许中断)时，IF1D寄存器的Bit0位就变为“1”(有中断请求)。
- 当TBIER寄存器的多个位为“1”时，必须通过TBSR寄存器判断是哪个请求源产生的中断。
- 因为即使接受中断，TBSR寄存器的各位也不会自动变为“0”，所以必须在中断程序中将这位置“0”。
- 要将定时器B的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时，如果该中断被定时器B中断允许寄存器(TBIER)设置为禁止中断，就必须用以下(a)~(c)的任意一种方法置“0”。
 - 必须在将定时器B中断允许寄存器(TBIER)置“00H”(禁止全部中断)后给对象状态标志写“0”。
 - 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。

(例)在TBIMIEA和TBOVIE为允许中断而TBIMIEB为禁止中断的状态下清除TBIMFB的情况

- 定时器B中断允许寄存器(TBIER)的状态



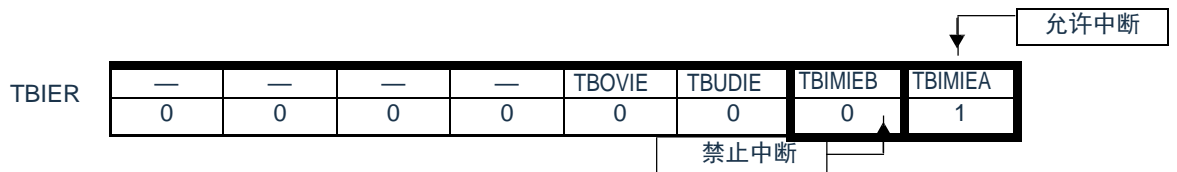
- 定时器B状态寄存器(TBSR)的状态



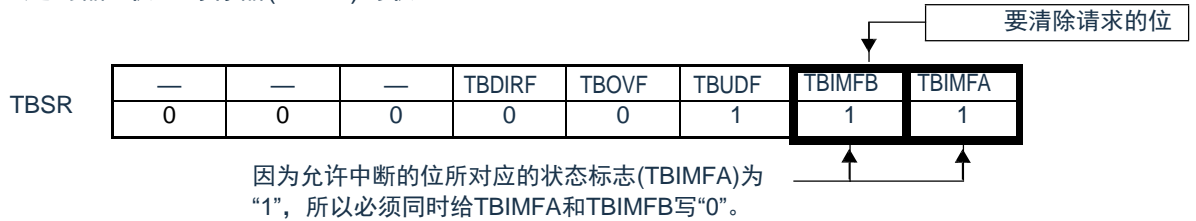
- (c) 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(例)在TBIMIEA为允许中断而TBIMIEB为禁止中断的状态下清除TBIMFB的情况

- 定时器B中断允许寄存器(TBIER)的状态



- 定时器B状态寄存器(TBSR)的状态

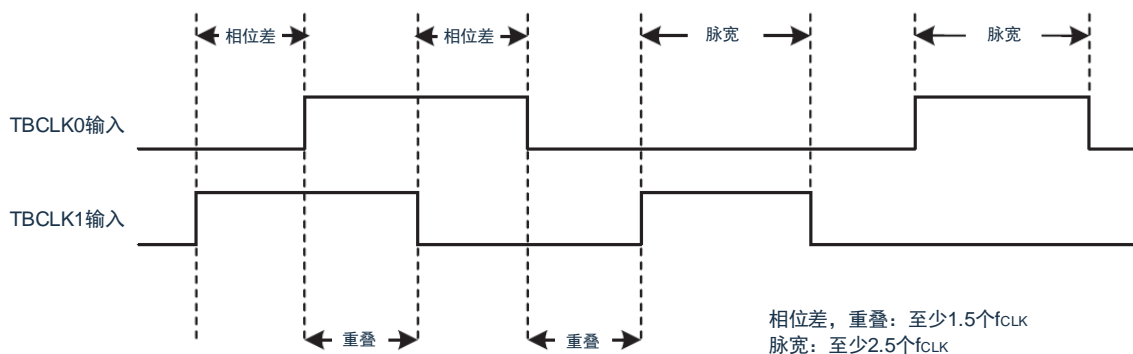


8.6 使用定时器B时的注意事项

8.6.1 相位计数模式中的相位差、重叠和脉宽

TBCLK0引脚和TBCLK1引脚的外部输入信号的相位差和重叠必须分别至少为1.5个 F_{CLK} ，脉宽必须至少为2.5个 F_{CLK} 。相位计数模式中的相位差、重叠和脉宽如图8-31所示。

图8-31：相位计数模式中的相位差、重叠和脉宽



8.6.2 模式的切换

- 要在运行过程中切换模式时，必须在将TBMR寄存器的TBSTART位置“0”(停止计数)后进行切换。
- 必须在切换模式后并且在开始运行前将IF1D寄存器的bit0置“0”。详细内容请参照“第25章 中断功能”。

8.6.3 计数源的切换

- 要切换计数源时，必须在停止计数后进行切换^注。变更步骤
 - (a) 将TBMR寄存器的TBSTART位置“0”(停止计数)。
 - (b) 更改TBCR寄存器的TBTCK0~TBTCK2位。

注：在计数过程中禁止改写的寄存器和位如下：

- (a) TBMR寄存器的TBSTART位以外的全部位
- (b) TBCNTC寄存器
- (c) TBCR寄存器
- (d) TBIOR寄存器

8.6.4 TBIO0引脚和TBIO1引脚的设置步骤

在复位后，TBIO0引脚和TBIO1引脚的复用I/O端口用作输入端口。

- 要从TBIO0引脚和TBIO1引脚输出时，必须按照以下步骤进行设置。

变更步骤

- 1) 进行模式、初始值和输出允许的设置(因为初始值和允许设置都通过SFR进行)。
- 2) 将TBIO0引脚和TBIO1引脚对应的端口寄存器的位置“0”。
- 3) 将TBIO0引脚和TBIO1引脚对应的端口模式寄存器的位设置为输出模式(从TBIO0引脚和TBIO1引脚开始输出)。
- 4) 开始计数(TBMR寄存器的TBSTART位为“1”)。

- 要将TBIO0引脚和TBIO1引脚对应的端口模式寄存器的位从输出模式改为输入模式时，必须按照以下步骤进行设置。

变更步骤

- 1) 将TBIO0引脚和TBIO1引脚对应的端口模式寄存器的位设置为输入模式(从TBIO0引脚和TBIO1引脚开始输入)。
- 2) 设置为输入捕捉功能。
- 3) 开始计数(TBMR寄存器的TBSTART位为“1”)。

- 在将TBIO0引脚和TBIO1引脚从输出模式切换为输入模式时，可能根据引脚的状态进行输入捕捉的运行。当不使用数字滤波器时，至少在经过2个CPU时钟周期后进行边沿检测；当使用数字滤波器时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

8.6.5 外部时钟TBCLK0和TBCLK1

TBCLK_j引脚(j=0、1)输入的外部时钟的脉宽必须至少为3个定时器B的运行时钟(F_{CLK})周期。

8.6.6 SFR的读写存取

要设置定时器B时，必须先将PER1寄存器的TMBEN位置“1”。当TMBEN位为“0”时，忽视定时器B的控制寄存器的写操作，而且读取值都为初始值(端口寄存器和端口模式寄存器除外)。

1) TBMR寄存器

在进行数字滤波器的时钟切换时，必须按照以下步骤进行设置：

- a) 在TBSTART位为“0”(停止计数)的状态下设置TBMR寄存器的TBDFEA位和TBDFEB位(TBIO0引脚和TBIO1引脚的数字滤波器功能选择位)、TBMR寄存器的TBDFCK0位和TBDFCK1位(数字滤波器功能的时钟选择位)。
- b) 将TBSTART位置“1”。

但是，在不设置数字滤波器并且不更改复位后为“00B”的TBDFCK1位和TBDFCK0位的情况下，能进行1次性设置。

除了外部输入引脚(TBIO0、TBIO1)以外，能选择EVENTC输入的事件作为输入捕捉的运行源。当不使用此功能时，必须将TBMR寄存器的TBELCICE位置“1”，设置输入捕捉功能(输入捕捉的有效边沿为上升沿(TBIOB2~TBIOB0=100B))。在使用PWM模式或者定时器模式的输出比较功能时(TBPWM=1、TBIOB2=0)，此功能无效。

2) TB寄存器

TBMR寄存器的写操作优先定时器B运行条件产生的计数复位。

8.6.7 停止计数时的输入捕捉运行

在输入捕捉模式中，如果在TBMR寄存器的TBSTART位为“0”(停止计数)时给TBIO0/TBIO1引脚输入TBIOj寄存器的TBIOj0位和TBIOj1位选择的边沿，就在TBIO0/TBIO1输入的有效边沿产生输入捕捉中断请求(j=A、B)。

第9章 定时器C

9.1 定时器C的功能

定时器C是一个可以通过软件，比较器1和定时器M触发输入捕捉功能的定时器。

动作如下：

计数开始：由软件或定时器M触发计数动作

计数停止：由软件或者比较器1的输出触发计数停止

输入捕捉：在比较器1的中断发生时，计数值传送到缓冲器里

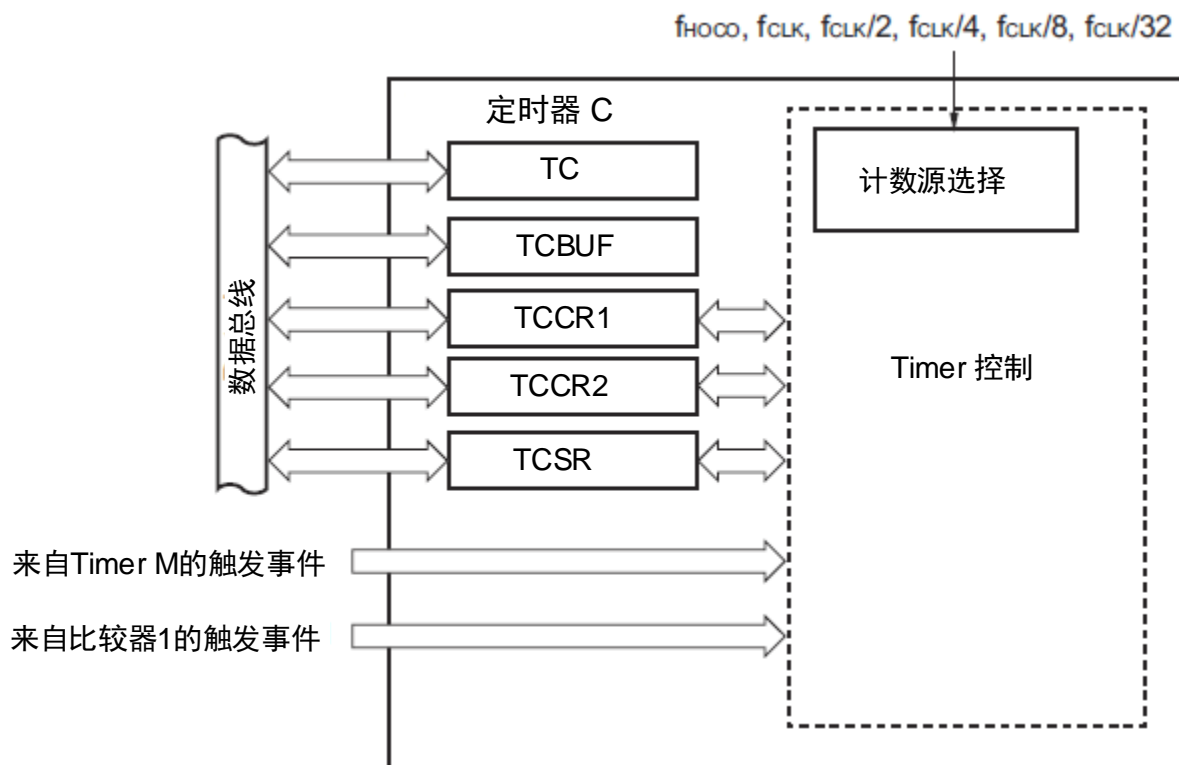
计数复位：由定时器M或者比较器1触发计数复位

Timer C的动作时钟可以选择 F_{CLK} 或者 F_{HOCO} 。

9.2 定时器C的结构

定时器C的框图如图9-1所示。

图9-1：定时器C的框图



9.3 控制定时器C的寄存器

控制定时器C的寄存器如表9-1所示。

表9-1: 控制定时器C的寄存器

寄存器名	符号
外围允许寄存器1	PER1
定时器C计数寄存器	TC
定时器C计数缓冲寄存器	TBUF0
定时器C控制寄存器1	TCCR1
定时器C控制寄存器2	TCCR2
定时器C状态寄存器	TCSR

9.3.1 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器C时，必须将bit1(TMCEN)置“1”。通过8位存储器操作指令设置PER1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图9-2: 外围允许寄存器1(PER1)的格式

地址: 0x4002081A 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMOPEN	TMCEN	TMAEN

TMCEN	提供定时器C的输入时钟的控制
0	停止提供输入时钟。 •不能写定时器C使用的SFR。 •定时器A处于复位状态。
1	提供输入时钟。 •能读写定时器C使用的SFR。

注意: 要设置定时器C时，必须先将TMCEN位置“1”。当TMCEN位为“0”时，忽视定时器C的控制寄存器的写操作，而且读取值都为初始值。

要选择 F_{HOCO} 作为定时器C的计数源时，必须在将外围允许寄存器1(PER1)的bit2(TMCEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要 F_{CLK} 改为 F_{IH} 以外的时钟，就必须在清除外围允许寄存器1(PER1)的bit2(TMCEN)后进行更改。

9.3.2 定时器C计数寄存器(TC)

这是16位寄存器。如果写此寄存器，就将数据写到重加载寄存器。如果读此寄存器，就读计数值。

图9-3: 定时器C计数寄存器(TC)的格式

地址: 0x40042C50 复位后: 0000H R/W

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

—	功能	设置范围
bit15~0	递增计数, 发生上溢时TCSR的TCOVF bit被置为1	0000H~FFFFH

注意: 选项字节(000C2H/010C2H)的FRQSEL4=1, 并且PER1的TMCEN=0时, 复位值不定。如果需要读出初期值, 需要在选择F_{CLK}并且TMCEN=1的条件下读出。

9.3.3 定时器C计数缓冲寄存器(TCBUF)

图9-4: 定时器C计数缓冲寄存器(TCBUF)的格式

地址: 0x40042C52 复位后: 0000H R

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

—	功能	设置范围
bit15~0	在比较器1的中断发生时, TC寄存器的值传送到缓冲寄存器里	0000H~FFFFH

注意: 选项字节(000C2H/010C2H)的FRQSEL4=1, 并且PER1的TMCEN=0时, 复位值不定。如果需要读出初期值, 需要在选择F_{CLK}并且TMCEN=1的条件下读出。

9.3.4 定时器C控制寄存器1(TCCR1)

图9-5: 定时器C控制寄存器1(TCCR1)的格式

地址: 0x40042C54复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCCR1	TCK2	TCK1	TCK0	START_MD	TRIG_MD_SW	TRIG_MD_HW	TM_TRIG	OVIE

TCK2	TCK1	TCK0	计数源选择 ^注
0	0	0	F _{CLK} , F _{HOCO} ^{注1}
0	0	1	F _{CLK} /2 ^{注2}
0	1	0	F _{CLK} /4 ^{注2}
0	1	1	F _{CLK} /8 ^{注2}
1	0	0	F _{CLK} /32 ^{注2}
上述以外			禁止设置

START_MD	计数开始源的选择
0	软件设置Timer C计数开始
1	Timer M的输出信号触发Timer C计数开始
START_MD=1时, 在TCCR2的TSTART位设置为1后, 计数开始	

TRIG_MD_SW	软件复位Timer C的有效信号
0	禁止软件复位Timer C的计数计数器
1	允许软件复位Timer C的计数计数器
START_MD=1时无效	

TRIG_MD_HW	通过Timer M的输出触发Timer C时的动作选择
0	复位Timer C后开始计数
1	Timer C开始计数
START_MD=0时无效	

TM_TRIG	来自Timer M的硬件触发的选择
0	TM0的计数开始动作(TSTART0设置为1)触发Timer C开始计数
1	TM1的计数开始动作(TSTART1设置为1)触发Timer C开始计数
START_MD=0时无效	

OVIE	允许上溢中断信号产生
0	禁止在TC寄存器上溢时产生中断
1	允许在TC寄存器上溢时产生中断

注: 在和Timer M联动时, 必须设置Timer C的动作时钟与Timer M的动作时钟频率一致。

注1: 选项字节(000C2H/010C2H)的FRQSEL4=0时选择F_{CLK}。选项字节(000C2H/010C2H)的FRQSEL4=1时选择F_{HOCO}。

注2: 选项字节(000C2H/010C2H)的FRQSEL4=1时不能设置。

注意: 选项字节(000C2H/010C2H)的FRQSEL4 = 1, 并且PER1的TMCEN=0时, 复位值不定。如果需要读出初期值, 需要在选择F_{CLK}并且TMCEN=1的条件下读出。

9.3.5 定时器C控制寄存器1(TCCR2)

图9-6: 定时器C控制寄存器1(TCCR2)的格式

地址: 0x40042C55复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCCR2	0	0	0	0	0	CMP_TCR1	CMP_TCR0	TSART

CMP1_TCR1	CMP1_TCR0	通过比较器 1 的输出触发 Timer C 时的动作选择
0	0	Timer C 计数停止
0	1	Timer C 计数值传送到缓冲寄存器里, 计数继续
1	0	Timer C 计数值变为 0000H, 计数继续
1	1	Timer C 计数值传送到缓冲寄存器里, Timer C 计数值变为 0000H, 计数继续

TSTART	Timer C动作开始控制 ^注
0	TC计数停止
1	TC计数开始

注: 如果来自CMP1的停止信号控制和TSTART控制产生了竞争, 来自比较器1的停止信号具有更高的优先级。

注意: 选项字节(000C2H/010C2H)的FRQSEL4=1, 并且PER1的TMCEN=0时, 复位值不定。如果需要读出初期值, 需要在选择F_{CLK}并且TMCEN=1的条件下读出。

9.3.6 定时器C状态寄存器(TCSR)

图9-7: 定时器C控制寄存器1(TCSR)的格式

地址: 0x40042C56 复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
TCSR	0	0	0	0	0	0	TCSB	TCOVF

TCSB	Timer C计数器状态标志位 ^{注1}
0	计数停止
1	计数中

TCOVF	Timer C计数器的上溢状态标志位 ^{注2,3}
0	没有发生上溢
1	发生了上溢

注1: 只读, 不可写。

注2: 只能写入0, 写1无效。

注3: Timer C计数器上溢和TCOVF写0同时发生时, 上溢具有较高的优先级。

注意: 选项字节(000C2H/010C2H)的FRQSEL4 = 1, 并且PER1的TMCEN=0时, 复位值不定。如果需要读出初期值, 需要在选择F_{CLK}并且TMCEN=1的条件下读出。

9.4 定时器C的运行

定时器C可以由定时器M的信号触发计数开始，比较器1的信号触发计数停止。

9.4.1 计数源

定时器C的动作时钟由选项字节和定时器C的分频设置决定。

(1) 定时器C的时钟源

- 选项字节(000C2H/010C2H)的FRQSEL4=1时，CPU/外围动作时钟(fcpu)选择高速振荡时钟 F_{HOCO} ，定时器C的时钟源选择 F_{HOCO} 。
- 选项字节(000C2H/010C2H)的FRQSEL4=0时或者CPU/外围动作时钟(fcpu)选择高速系统时钟(F_{MX})，定时器C的时钟源选择fcpu。

(2) 定时器C计数时钟源

使用TCCR1设置计数频率。

如果使用Timer M的信号触发Timer C计数开始，必须设置Timer C的动作时钟与Timer M的动作时钟频率一致。

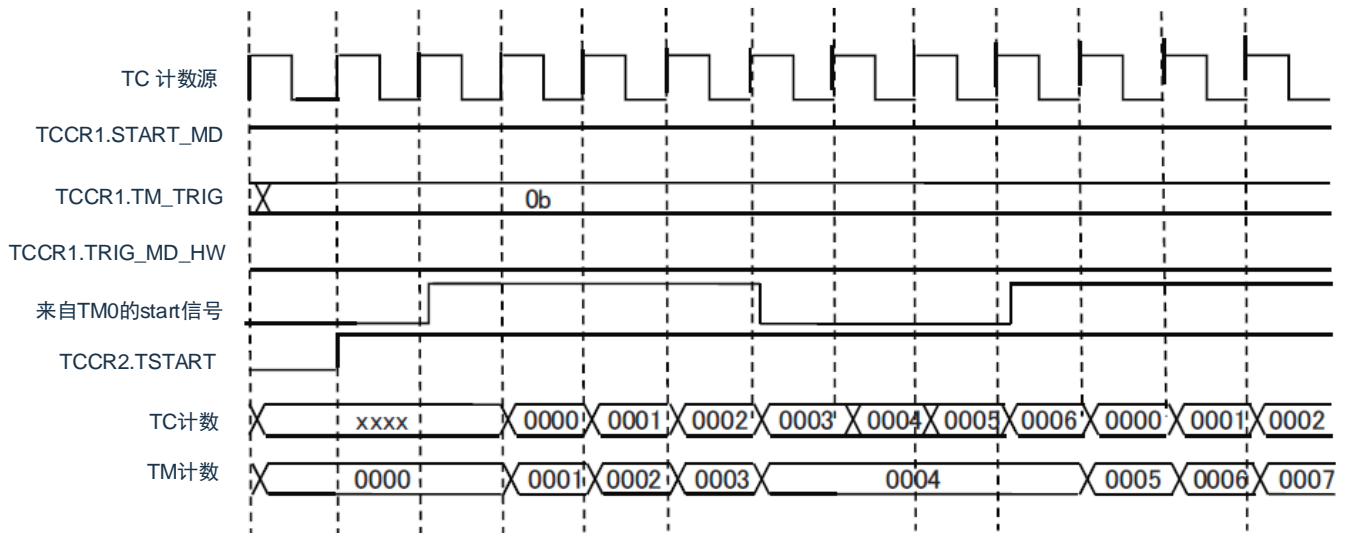
9.4.2 定时器C开始计数的动作

通过Timer M或者软件设置，开始Timer C的计数动作。

9.4.2.1 选择Timer M的信号作为触发时的设置和动作

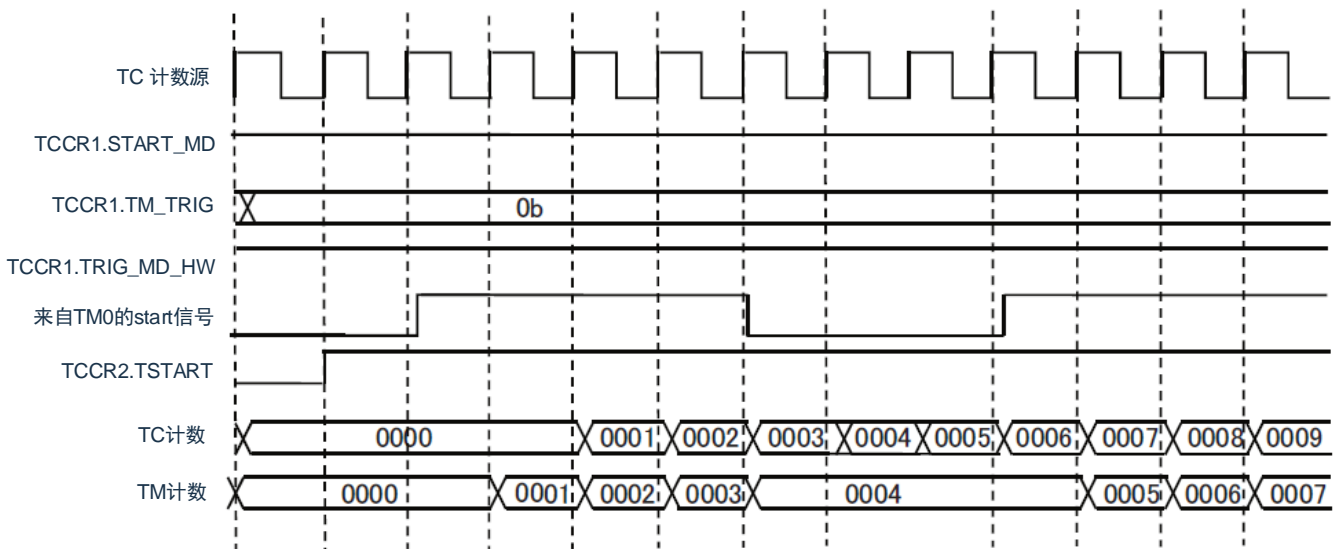
- (1) TRIG_MD_HW=0时, Timer C计数的复位和开始设置步骤:
 - (a) 选择Timer M输出信号作为计数开始的触发源: TCCR1.START_MD=1
 - (b) 选择Timer C的触发功能: TCCR1.TRIG_MD_HW=0
 - (c) 选择Timer M_0/1的触发信号: TCCR1.TC_TRIG=1/0
 - (d) Timer C计数开始: TCCR2.TSTART=1

图9-8: Timer C计数复位和开始的例子(TRIG_MD_HW=0)



- (2) TRIG_MD_HW=1时, Timer C计数的开始设置步骤:
 - (a) 选择Timer M输出信号作为计数开始的触发源: TCCR1.START_MD=1
 - (b) 选择Timer C的触发功能: TCCR1.TRIG_MD_HW=1
 - (c) 选择Timer M_0/1的触发信号: TCCR1.TC_TRIG=1/0
 - (d) Timer C计数开始: TCCR2.TSTART=1

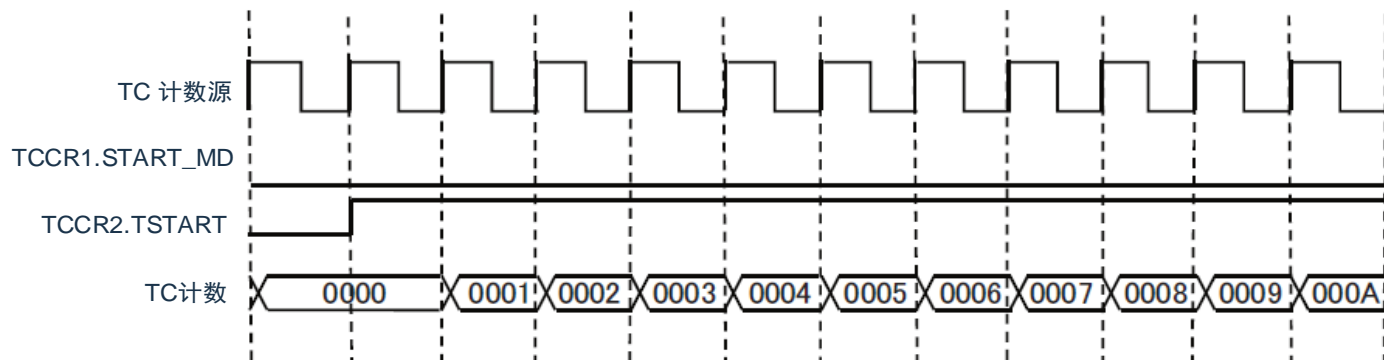
图9-9: Timer C计数开始的例子(TRIG_MD_HW=1)



9.4.2.2 选择软件触发时的设置和动作

1. 计数开始源选择软件触发: TCCR1.START_MD=0
2. Timer C计数开始: TCCR2.TSTART=1

图9-10: 软件触发Timer C计数开始的例子



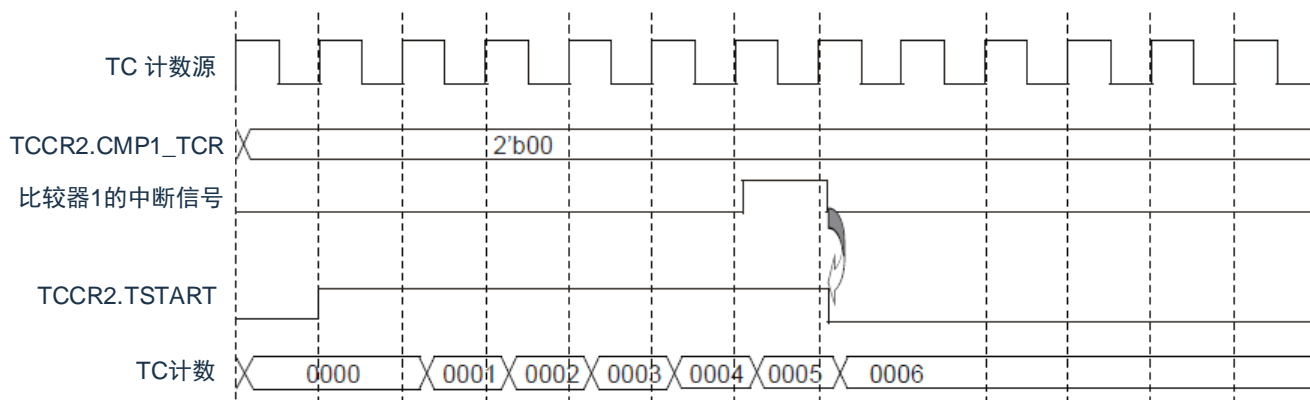
9.4.3 定时器C计数停止的动作

定时器C在计数中，可以通过比较器1的触发或者软件设置来停止计数动作。

9.4.3.1 选择比较器1作为触发时的设置和动作

1. 选择比较器1作为触发：TCCR2.CMP1_TCR=00
2. Timer C计数开始：TCCR2.TSTART=1

图9-11：选择比较器1触发Timer C停止的例子



9.4.3.2 软件触发时的设置和动作

1. TimerC计数开始：TCCR2.TSTART=1
2. 软件设置TCCR2.TSTART为0，Timer C计数停止

9.4.4 输入捕捉动作

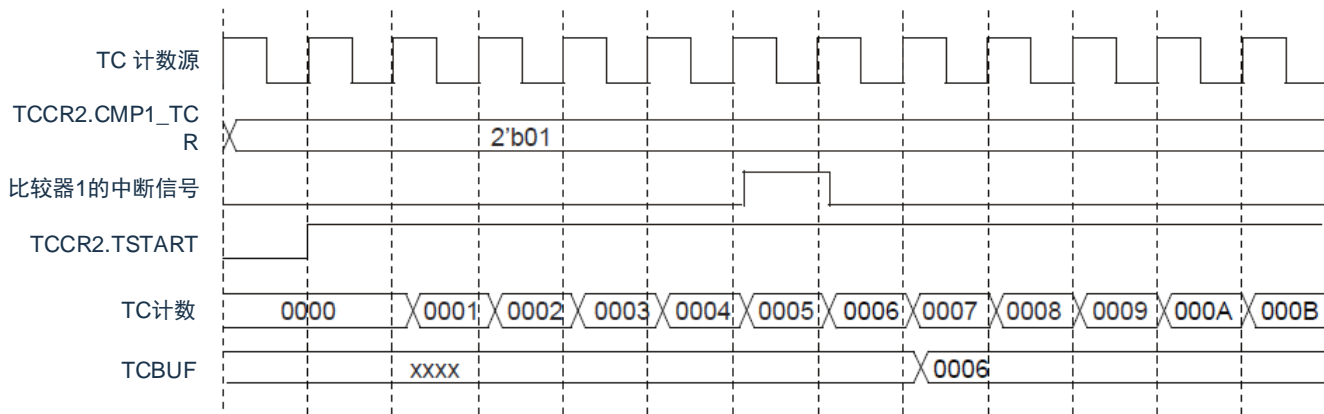
如果在Timer C动作过程中，比较器1产生了中断，Timer C的动作会发生变化。

(1) Case 1:

TCCR2.CMP1_TCR=01, Timer C的计数值传送到计数缓冲器里。

- TCCR2.CMP1_TCR=01(选择输入捕捉功能)
- TCCR2.TSTART=1(Timer C计数开始)

图9-12: 输入捕捉的动作例子

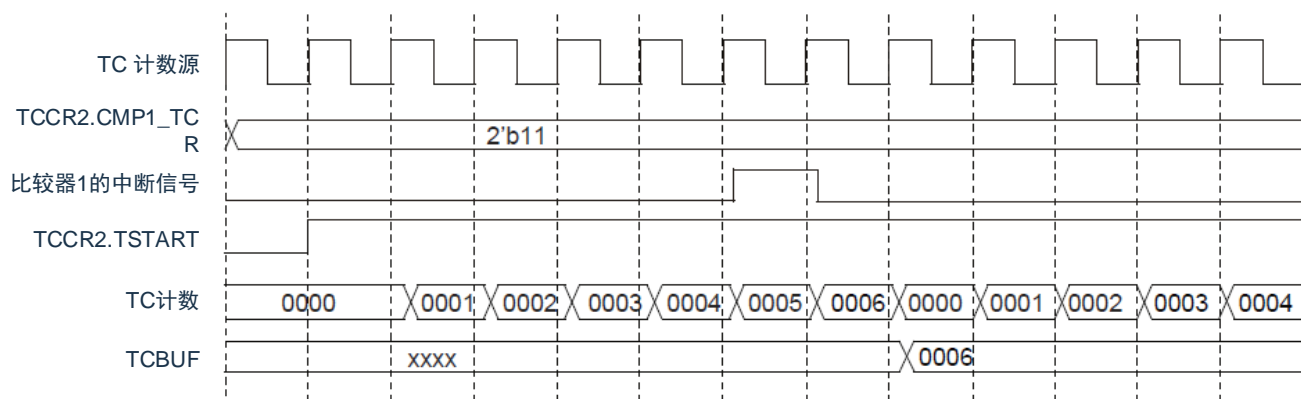


(2) Case 2:

TCCR2.CMP1_TCR=11, Timer C的计数值传送到计数缓冲器里, Timer C的计数值复位。

- TCCR2.CMP1_TCR=11(选择输入捕捉并且复位功能)
- TCCR2.TSTART=1(Timer C计数开始)

图9-13: 输入捕捉的动作例子(同时复位计数值)



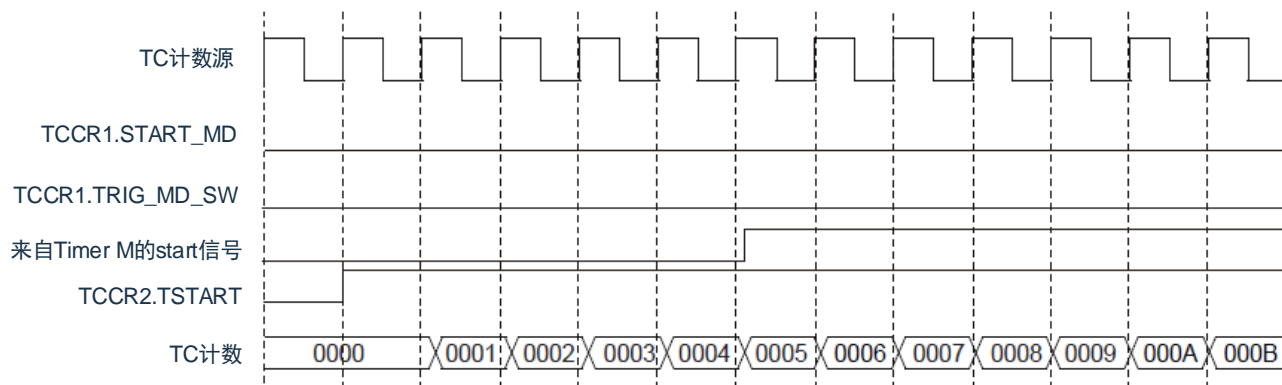
9.4.5 定时器C计数复位动作

使用软件设置Timer C动作开始的场合，Timer M的输出信号和比较器1的输出信号可以复位Timer C的计数器。

(1) TRIG_MD_SW=0，软件触发计数开始时，Timer M的输出信号不能对计数值进行复位。

- 软件触发计数开始：TCCR1.START_MD=0
- 允许软件复位计数器：TCCR1.TRIG_MD_SW=0
- Timer C计数开始：TCCR2.TSTART=1

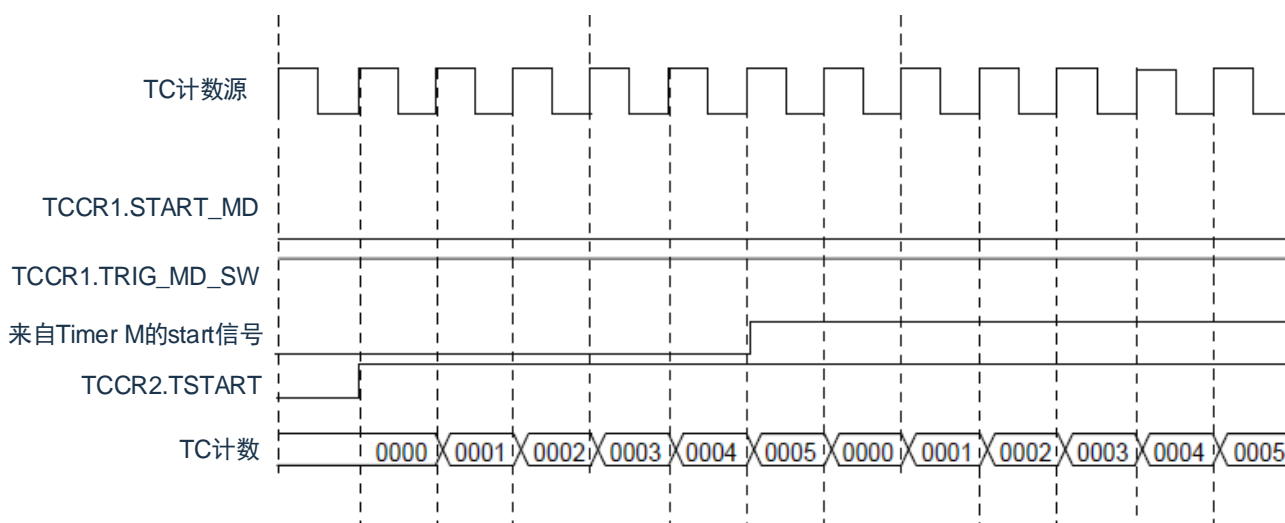
图9-14：Timer M触发Timer C计数复位的例子



(2) TRIG_MD_SW=1，软件触发计数开始时，Timer M的输出信号对计数值进行复位。

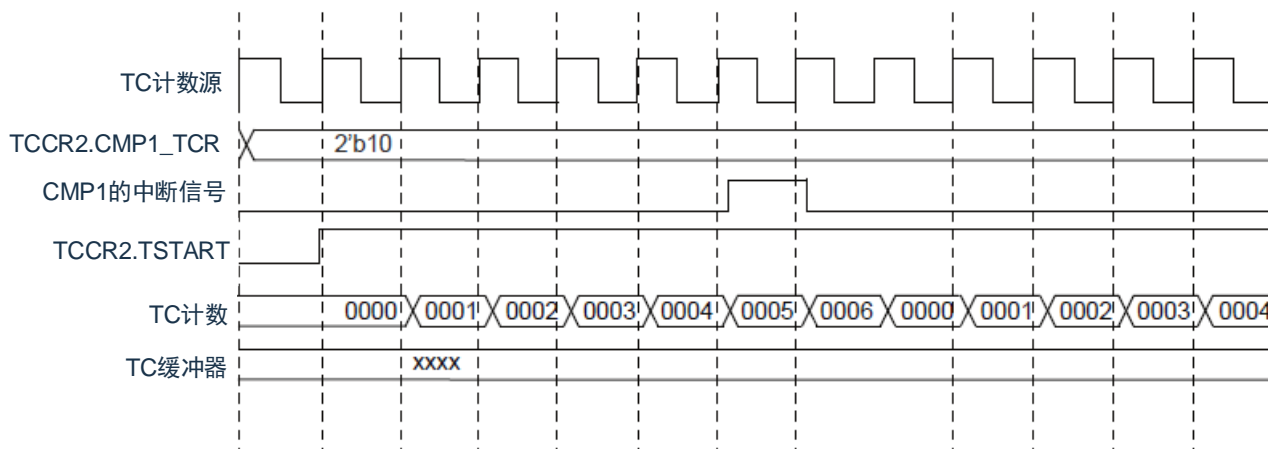
- 软件触发计数开始：TCCR1.START_MD=0
- 禁止软件复位计数器：TCCR1.TRIG_MD_SW=1
- Timer C计数开始：TCCR2.TSTART=1

图9-15：Timer M触发Timer C计数复位的例子



- (3) TCCR2.CMP1_TCR=10时, CMP1的输出信号对计数值进行复位。
- 计数值复位, 计数动作继续: TCCR2.CMP1_TCR=10(不能使用输入捕捉功能)
 - Timer C计数开始: TCCR2.TSTART=1

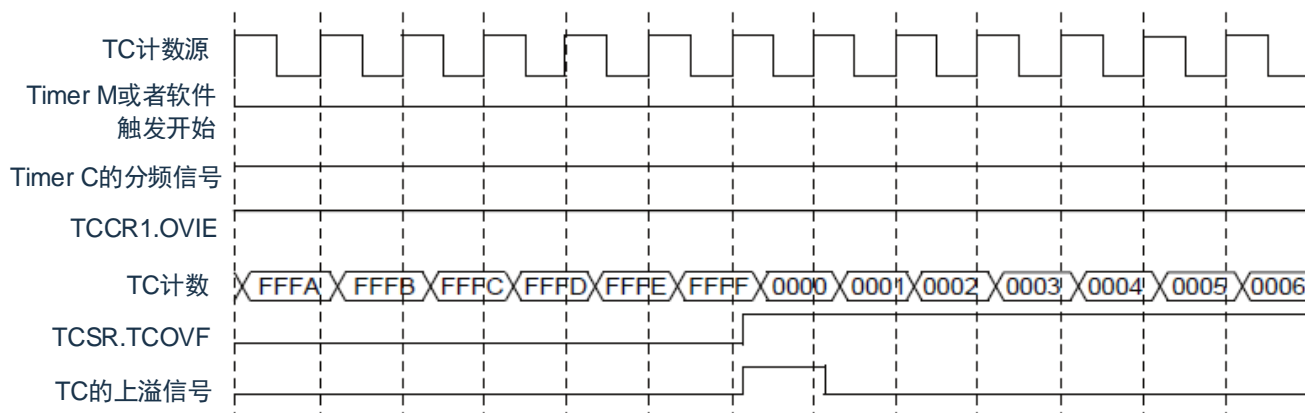
图9-16: CMP1触发Timer C计数复位的例子



9.4.6 定时器C的中断

定时器C的计数器发生上溢时，如果设置了TCCR1.OVIE=1，会产生上溢中断信号。

图9-17: Timer C上溢时中断产生的例子



9.5 定时器C使用时的注意事项

9.5.1 寄存器的读写

要设置定时器C时，必须先将PER1的TMCEN位置“1”。当TMCEN位为“0”时，忽视定时器C的控制寄存器的写操作，而且读取值都为初始值。

在时钟停止时，除TC,TCBUF寄存器不能写入外，其它的寄存器可以读写。

注意：计数动作中下列寄存器不能改写。

- TCCR1
- TCCR2

9.5.2 上溢中断

Timer C的计数值为FFFFH时，如果在上溢到0000H前外部输入的触发事件将计数器复位了的话，上溢中断不会产生。

9.5.3 输入捕捉和定时器C计数复位动作

即使设置TCSR.TCSB=0(计数停止)，Timer M和CMP1的输入信号仍然能触发Timer C的输入捕捉动作和计数复位动作。

9.5.4 定时器C和定时器M，比较器1联动时的步骤

定时器C和定时器M,比较器1联动时，设置步骤如下：

1. 提供比较器1的时钟输入：PGACMPEN=1
2. 允许比较器1中断产生和输出：详细请参考“第17章 比较器CMP”。
3. 提供定时器C的时钟输入：TMCEN=1
4. 设置TCCR1
5. 设置TCCR2
6. 设置定时器M，定时器M计数开始：TMOEN=1
7. 定时器C计数开始：TCCR2.START=1

备注：

1. Timer C和Timer M,CMP1联动时，必须设置Timer C的动作时钟与Timer M的动作时钟频率一致。
2. 设置寄存器时，必须先设置控制寄存器TCCR1，再设置TCCR2.TSTART。

第10章 定时器M

10.1 定时器M的功能

定时器M有以下4种模式：

- 定时器模式
 - 输入捕捉功能：以外部信号为触发，将计数值取到寄存器。
 - 输出比较功能：检测计数值和寄存器的值是否相同(能在检测时更改引脚的输出)。
 - PWM功能：连续输出任意的脉宽。

以下3种模式使用PWM功能：

- 复位同步PWM模式：这是输出锯齿波调制、无死区时间的三相波形(6个)的模式。
- 互补PWM模式：这是输出三角波调制、有死区时间的三相波形(6个)的模式。
- PWM3模式：这是输出同周期PWM波形(2个)的模式。

在定时器模式中，定时器M0和定时器M1具有同等的输入捕捉功能、输出比较功能和PWM功能，每个引脚能选择这些功能，并且能在定时器M0和定时器M1中组合使用这些功能。

复位同步PWM模式、互补PWM模式和PWM3模式通过定时器M0和定时器M1的计数器和寄存器的组合来输出波形，引脚功能取决于运行模式。

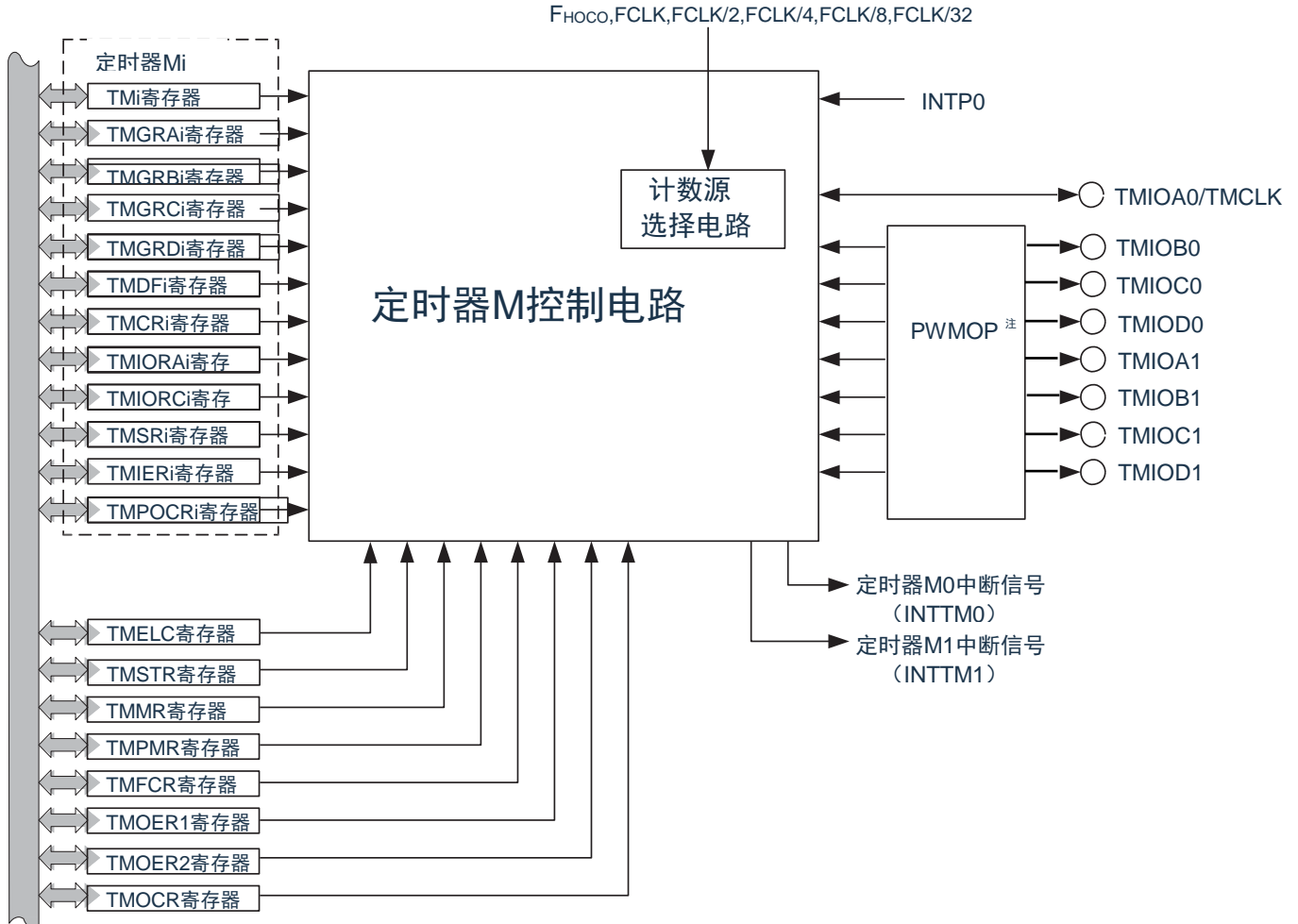
定时器M有4个输入/输出引脚。

定时器M的运行时钟为 F_{CLK} 或者 F_{HOCO} 。

10.2 定时器M的结构

定时器M的框图和引脚结构分别如图10-1和表10-1所示。

图10-1：定时器M的框图



注：只能截止输出

备注：i=0, 1

表10-1：定时器M的引脚结构

引脚名	复用的端口名	输入/输出	功能
TMIOA0/TMCLK	P17	输入/输出	功能因运行模式而不同，详细内容请参照各模式。
TMIOB0	P14	输入/输出	
TMIOC0	P16	输入/输出	
TMIOD0	P15	输入/输出	
TMIOA1	P12	输入/输出	
TMIOB1	P10	输入/输出	
TMIOC1	P13	输入/输出	
TMIOD1	P11	输入/输出	

10.3 控制定时器M的寄存器

控制定时器M的寄存器如表10-2所示。

表10-2：控制定时器M的寄存器

寄存器名	符号
外围允许寄存器1	PER1
定时器M EVENTC寄存器	TMELC
定时器M启动寄存器	TMSTR
定时器M模式寄存器	TMMR
定时器MPWM功能选择寄存器	TMPMR
定时器M功能控制寄存器	TMFCR
定时器M输出主允许寄存器1	TMOER1
定时器M输出主允许寄存器2	TMOER2
定时器M输出控制寄存器	TMOCR
定时器M数字滤波器功能选择寄存器0	TMDFO
定时器M数字滤波器功能选择寄存器1	TMDF1
定时器M控制寄存器0	TMCR0
定时器M I/O控制寄存器A0	TMIORA0
定时器M I/O控制寄存器C0	TMIORC0
定时器M状态寄存器0	TMSR0
定时器M中断允许寄存器0	TMIER0
定时器M PWM功能输出电平控制寄存器0	TMPOCR0
定时器M计数器0	TM0
定时器M通用寄存器A0	TMGRA0
定时器M通用寄存器B0	TMGRB0
定时器M通用寄存器C0	TMGRC0
定时器M通用寄存器D0	TMGRD0
定时器M控制寄存器1	TMCR1
定时器M I/O控制寄存器A1	TMIORA1
定时器M I/O控制寄存器C1	TMIORC1
定时器M状态寄存器1	TMSR1
定时器M中断允许寄存器1	TMIER1
定时器M PWM功能输出电平控制寄存器1	TMPOCR1
定时器M计数器1	TM1
定时器M通用寄存器A1	TMGRA1
定时器M通用寄存器B1	TMGRB1
定时器M通用寄存器C1	TMGRC1
定时器M通用寄存器D1	TMGRD1
端口寄存器	Pxx
端口模式寄存器	PMx,PMCx

10.3.1 外围允许寄存器1(PER1)

PER1 寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器M时，必须将bit4(TMMEN)置“1”。通过8位存储器操作指令设置PER1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图10-2：外围允许寄存器1 (PER1)的格式

地址:	0x4002081A	复位后: 00H	R/W					
符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN [‡]	TMMEN	DMAEN	PWMOPEN	TMCEN	TMAEN

TMMEN	提供定时器M的输入时钟的控制
0	停止提供输入时钟。 • 不能写定时器M使用的SFR。 • 定时器M处于复位状态。
1	提供输入时钟。 • 能读写定时器M使用的SFR。

PWMOPEN	PWM截止控制电路的输入时钟的控制
0	停止提供输入时钟。 • 不能写PWM截止电路使用的SFR。 • PWM截止电路处于复位状态。
1	提供输入时钟。 • 能读写PWM截止电路使用的SFR。

注意:

1. 要设置定时器M时，必须先将TMMEN位置“1”。当TMMEN 位为“0”时，忽视定时器M的控制寄存器的写操作，而且读取值都为初始值(端口模式寄存器1(PM1)和端口寄存器1(P1)除外)。
2. 要选择FHOCO作为定时器M的计数源时，必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将FCLK设置为F_{IH}。如果要将FCLK改为FIH以外的时钟，就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。
3. 要设置PWMOP时，必须先将PWMOPEN位置“1”。当 PWMOPEN位为“0”时，忽视PWMOP的控制寄存器的写操作，而且读取值都为初始值。详细参见“10.8 PWMOP”。

10.3.2 定时器M EVENTC 寄存器 (TMELC)

图10-3: 定时器M EVENTC寄存器(TMELC)的格式

地址:	0x40042A60	复位后: 00H ^注	R/W						
符号	7	6	5	4	3	2	1	0	
TMELC	0	0	ELCOBE1	ELCICE1	0	0	ELCOBE0	ELCICE0	

ELCOBE1	EVENTC事件输入1(用于强制截止定时器M的脉冲输出)的允许
0	禁止强制截止。
1	允许强制截止。

ELCICE1	EVENTC事件输入1(用于定时器M的输入捕捉D1)的选择
0	选择输入捕捉TMIOD1。
1	选择来自事件联动控制器(EVENTC)的事件输入1。

ELCOBE0	EVENTC事件输入0(用于强制截止定时器M的脉冲输出)的允许
0	禁止强制截止。
1	允许强制截止。

ELCICE0	EVENTC事件输入0(用于定时器M的输入捕捉D0)的选择
0	选择输入捕捉TMIOD0。
1	选择来自事件联动控制器(EVENTC)的事件输入0。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.3 定时器M启动寄存器(TMSTR)

能通过8位存储器操作指令设置TMSTR寄存器。请参照使用定时器M时的注意事项的“10.7.1(1) TMSTR寄存器”。

图10-4: 定时器 M 启动寄存器(TMSTR)的格式

地址:	0x40042A63	复位后: 0CH ^{注1}	R/W						
符号	7	6	5	4	3	2	1	0	
TMSTR	0	0	0	0	CSEL1	CSEL0	TSTART1	TSTART0	

CSEL1	TM1计数运行的选择 ^{注2}
0	在和TMGRA1寄存器比较匹配时停止计数。
1	在和TMGRA1寄存器比较匹配后还继续计数 ^{注3} 。

CSEL0	TM0计数运行的选择
0	在和TMGRA0寄存器比较匹配时停止计数。
1	在和TMGRA0寄存器比较匹配后还继续计数 ^{注3} 。

TSTART1	TM1计数的开始标志 ^{注4,5}
0	停止计数。
1	开始计数。

TSTART0	TM0计数的开始标志 ^{注6,7}
0	停止计数。
1	开始计数。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 不能用PWM3模式。

注3: 在使用输入捕捉功能时, 必须将此位置“1”。

注4: 当CSEL1位为“1”时, 必须给TSTART1位写“0”。

注5: 当CSEL1位为“0”并且产生比较匹配信号(TMIOA1)时, 此标志为“0”(停止计数)。

注6: 当CSEL0位为“1”时, 必须给TSTART0位写“0”。

注7: 当CSEL0位为“0”并且产生比较匹配信号(TMIOA0)时, 此标志为“0”(停止计数)。

10.3.4 定时器M模式寄存器(TMMR)

图10-5: 定时器M模式寄存器(TMMR)的格式

地址:	0x40042A64	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMMR	TMBFD1	TMBFC1	TMBFD0	TMBFC0	0	0	0	TMSYNC

TMBFD1	TMGRD1寄存器功能的选择 ^{注2}
0	通用寄存器
1	TMGRB1寄存器的缓冲寄存器

TMBFC1	TMGRC1寄存器功能的选择 ^{注2}
0	通用寄存器
1	TMGRA1寄存器的缓冲寄存器

TMBFD0	TMGRD0寄存器功能的选择 ^{注2}
0	通用寄存器
1	TMGRB0寄存器的缓冲寄存器

TMBFC0	TMGRC0寄存器功能的选择 ^{注2,3}
0	通用寄存器
1	TMGRA0寄存器的缓冲寄存器

TMSYNC	定时器M的同步 ^{注4}
0	TM0和TM1独立运行。
1	TM0和TM1同步运行。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在使用输出比较功能时, 如果通过TMIORCi寄存器(i=0、1)的IOj3(j=C或者D)位选择“0”(更改TMGRji寄存器的输出引脚), 就必须将TMMR寄存器的TMBFji位置“0”。

注3: 在互补PWM模式中, 必须将此位置“0”(通用寄存器)。

注4: 在复位同步PWM模式、互补PWM模式或者PWM3模式中, 必须将此位置“0”(TM0和TM1独立运行)。

10.3.5 定时器M PWM功能选择寄存器(TMPMR)

图10-6: 定时器M PWM功能选择寄存器(TMPMR)的格式[定时器模式]

地址: 0x40042A65 复位后: 00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMPMR	0	TMPWMD1	TMPWMC1	TMPWMB1	0	TMPWMD0	TMPWMC0	TMPWMB0

TMPWMD1	TMIOD1的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

TMPWMC1	TMIOC1的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

TMPWMB1	TMIOB1的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

TMPWMD0	TMIOD0的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

TMPWMC0	TMIOC0的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

TMPWMB0	TMIOB0的PWM功能选择
0	输入捕捉功能或者输出比较功能
1	PWM功能

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.6 定时器M功能控制寄存器(TMFCR)

图10-7: 定时器M功能控制寄存器(TMFCR)的格式

地址: 0x40042A66 复位后: 80H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMFCR	PWM3	STCLK	0	0	OLS1	OLS0	CMD1	CMD0

PWM3	PWM3模式的选择 ^{注2}
<ul style="list-style-type: none"> 在定时器模式中, 必须置“1”(非PWM3模式)。 在PWM3模式中, 必须置“0”(PWM3模式)。 在复位同步PWM模式和互补PWM模式中无效。 	

STCLK	外部时钟输入的选择
<ul style="list-style-type: none"> 定时器模式、复位同步PWM模式、互补PWM模式 0: 外部时钟输入无效 1: 外部时钟输入有效 在PWM3模式中, 必须置“0”(外部时钟输入无效)。 	

OLS1	反相输出电平的选择(复位同步PWM模式或者互补PWM模式)
<ul style="list-style-type: none"> 复位同步PWM模式、互补PWM模式 0: 初始输出“H”电平, “L”电平有效。 1: 初始输出“L”电平, “H”电平有效。 在定时器模式和PWM3模式中无效。 	

OLS0	正相输出电平的选择(复位同步PWM模式或者互补PWM模式)
<ul style="list-style-type: none"> 复位同步PWM模式、互补PWM模式 0: 初始输出“H”电平, “L”电平有效。 1: 初始输出“L”电平, “H”电平有效。 在定时器模式和PWM3模式中无效。 	

CMD1	CMD0	组合模式的选择 ^{注3、4}
<ul style="list-style-type: none"> 在定时器模式和PWM3模式中, 必须置“00B”(定时器模式或者PWM3模式)。 在复位同步PWM模式中, 必须置“01B”(复位同步PWM模式)。 互补PWM模式 		
CMD1 CMD0 1 0 : 互补PWM模式(在TM1发生下溢时, 将数据从缓冲寄存器传送到通用寄存器) 1 1 : 互补PWM模式(在TM0和TMGRA0寄存器比较匹配时, 将数据从缓冲寄存器传送到通用寄存器) 上述以外: 禁止设置。		

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在CMD1位和CMD0位为“00B”(定时器模式或者PWM3模式)时, PWM3位的设置有效。

注3: 必须在TMSTR寄存器的TSTART0位和TSTART1位都为“0”(停止计数)时写CMD0位和CMD1位。

注4: 当CMD1位和CMD0位为“01B”、“10B”或者“11B”时, 与TMPMR寄存器的设置无关, 为复位同步PWM模式或者互补PWM模式。

10.3.7 定时器 M 输出主允许寄存器1(TMOER1)

图10-8: 定时器M输出主允许寄存器1(TMOER1)的格式
[输出比较功能、PWM功能、复位同步PWM模式、互补PWM模式和PWM3模式]

地址: 0x40042A67 复位后: FFH^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMOER1	ED1	EC1	EB1	EA1	ED0	EC0	EB0	EA0

ED1	TMIOD1输出的禁止 ^{注2}
0	允许输出。
1	禁止输出(TMIOD1 引脚为I/O端口)。

EC1	TMIOC1输出的禁止 ^{注2}
0	允许输出。
1	禁止输出(TMIOC1引脚为I/O端口)。

EB1	TMIOB1输出的禁止 ^{注2}
0	允许输出。
1	禁止输出(TMIOB1引脚为I/O端口)。

EA1	TMIOA1输出的禁止 ^{注2,3}
0	允许输出。
1	禁止输出(TMIOA1引脚为I/O端口)。

ED0	TMIOD0输出的禁止 ^{注2}
0	允许输出。
1	禁止输出(TMIOD0引脚为I/O端口)。

EC0	TMIOC0输出的禁止 ^{注2}
0	允许输出。
1	禁止输出(TMIOC0引脚为I/O端口)。

EB0	TMIOB0输出的禁止
0	允许输出。
1	禁止输出(TMIOB0引脚为I/O端口)。

EA0	TMIOA0输出的禁止 ^{注3,4}
0	允许输出。
1	禁止输出(TMIOA0引脚为I/O端口)。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在PWM3模式中, 必须将此位置“1”。

注3: 当使用PWM功能时, 必须将此位置“1”。

注4: 在复位同步PWM模式和互补PWM模式中, 必须将此位置“1”。

10.3.8 定时器M输出主允许寄存器2(TMOER2)

图10-9: 定时器 M 输出主允许寄存器2(TMOER2)的格式
[PWM功能、复位同步PWM模式、互补PWM模式和PWM3模式]

地址: 0x40042A68 复位后: 00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMOER2	TMPTO	0	0	0	0	0	0	TMSHUTS

TMPTO	脉冲输出强制截止信号INTP0 引脚输入的有效 ^{注2}
0	脉冲输出强制截止输入无效。
1	脉冲输出强制截止输入有效(如果给INTP0引脚输入“L”电平, TMSHUTS位就为“1”)。

TMSHUTS	强制截止标志
0	不强制截止。
1	处于强制截止。

当通过INTP0引脚或者ELC输入事件来强制截止脉冲时, 此位变为“1”, 而且不自动清除。因此, 要停止强制截止脉冲时, 必须在停止计数(TSTARTi=0)的过程中给此位写“0”。即使在有效模式中给TMSHUTS位写“1”也强制截止脉冲。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将FCLK设置为FIH并且将TMMEN位置“1”后读初始值。

注2: 请参照“10.4.4 脉冲输出的强制截止”。

10.3.9 定时器M输出控制寄存器(TMOCR)

必须在TMSTR寄存器的TSTART0位和TSTART1位都为“0”(停止计数)时写TMOCR寄存器。

图10-10: 定时器M输出控制寄存器(TMOCR)的格式[输出比较功能]

地址: 0x40042A69 复位后: 00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TMIOD1初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOC1	TMIOC1初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOB1	TMIOB1初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOA1	TMIOA1初始输出电平的选择
0	初始输出“L”电平。
1	初始输出“H”电平。

TOD0	TMIOD0初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOC0	TMIOC0初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOB0	TMIOB0初始输出电平的选择 ^{注2}
0	初始输出“L”电平。
1	初始输出“H”电平。

TOA0	TMIOA0初始输出电平的选择
0	初始输出“L”电平。
1	初始输出“H”电平。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在TMOCR寄存器的引脚功能为波形输出的情况下设置TMOCR寄存器时, 输出初始的输出电平。

图10-11: 定时器M输出控制寄存器(TMOCR)的格式[PWM功能]

地址:	0x40042A69	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TMIOD1初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。

TOC1	TMIOC1初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。

TOB1	TMIOB1初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。

TOA1	TMIOA1初始输出电平的选择
必须置“0”。	

TOD0	TMIOD0初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。

TOC0	TMIOC0初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。
在复位同步PWM模式和互补PWM模式中有效。	

TOB0	TMIOB0初始输出电平的选择 ^{注2}
0	初始输出为无效电平。
1	初始输出为有效电平。

TOA0	TMIOA0初始输出电平的选择
必须置“0”。	

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在TMOCR寄存器的引脚功能为波形输出的情况下设置TMOCR寄存器时, 输出初始的输出电平。

图10-12: 定时器M输出控制寄存器(TMOCR)的格式[PWM3模式]

地址:	0x40042A69	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMOCR	TOD1	TOC1	TOB1	TOA1	TOD0	TOC0	TOB0	TOA0

TOD1	TMIOD1初始输出电平的选择
在PWM3模式中无效。	

TOC1	TMIOC1初始输出电平的选择
在PWM3模式中无效。	

TOB1	TMIOB1初始输出电平的选择
在PWM3模式中无效。	

TOA1	TMIOA1初始输出电平的选择
在PWM3模式中无效。	

TOD0	TMIOD0初始输出电平的选择
在PWM3模式中无效。	

TOC0	TMIOC0初始输出电平的选择
在PWM3模式中无效。	

TOB0	TMIOB0初始输出电平的选择 ^{注2}
0	初始输出“L”电平，“H”电平有效。 在TMGRB1比较匹配时输出“H”电平，在TMGRB0比较匹配时输出“L”电平。
1	初始输出“H”电平，“L”电平有效。 在TMGRB1比较匹配时输出“L”电平，在TMGRB0比较匹配时输出“H”电平。

TOA0	TMIOA0初始输出电平的选择
0	初始输出“L”电平，“H”电平有效。 在TMGRA1比较匹配时输出“H”电平，在TMGRA0比较匹配时输出“L”电平。
1	初始输出“H”电平，“L”电平有效。 在TMGRA1比较匹配时输出“L”电平，在TMGRA0比较匹配时输出“H”电平。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在TMOCR寄存器的引脚功能为波形输出的情况下设置TMOCR寄存器时, 输出初始的输出电平。

10.3.10 定时器M数字滤波器功能选择寄存器i(TMDFi)(i=0、1)

图10-13：定时器M数字滤波器功能选择寄存器i(TMDFi)(i=0、1)的格式[输入捕捉功能]

地址： 0x40042A6A(TMDF0)、0x40042A6B(TMDF1) 复位后：00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	数字滤波器功能的时钟选择 ^{注2}
0	0	$F_{CLK}/32$ ^{注3}
0	1	$F_{CLK}/8$ ^{注3}
1	0	F_{CLK} ^{注3}
1	1	计数源(TMCRi寄存器的TCK0~TCK2位选择的时钟)

PENB1	PENB0	TMIOB引脚的脉冲强制截止的控制
0	0	必须置“00B”。

DFD	TMIODi引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。

当有数字滤波器功能时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

DFC	TMIOCi引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。

当有数字滤波器功能时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

DFB	TMIOBi引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。

当有数字滤波器功能时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

DFA	TMIOAi引脚的数字滤波器功能的选择
0	没有数字滤波器功能。
1	有数字滤波器功能。

当有数字滤波器功能时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

注1：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将 F_{CLK} 设置为 F_{IH} 并且将TMMEN位置“1”后读初始值。

注2：必须在设置DFCK0位和DFCK1位后开始计数。

注3：当用户选项字节(000C2H)的FRQSEL4位为“1”时， $F_{CLK}/32$ 、 $F_{CLK}/8$ 和 F_{CLK} 分别为 $F_{HOCO}/32$ 、 $F_{HOCO}/8$ 和 F_{HOCO} 。

图10-14：定时器M数字滤波器功能选择寄存器i(TMDFi)(i=0、1)的格式

[PWM功能、复位同步PWM模式、互补PWM模式和PWM3模式]

 地址： 0x40042A6A(TMDF0)、0x40042A6B(TMDF1) 复位后：00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMDFi	DFCK1	DFCK0	PENB1	PENB0	DFD	DFC	DFB	DFA

DFCK1	DFCK0	TMIOA引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L”电平输出
1	1	“H”电平输出

如果在这些模式中不将对应的引脚用作定时器M的输出端口，就必须置“00B”(禁止强制截止)。而且，必须在停止计数的过程中设置这些位。

PENB1	PENB0	TMIOB引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L”电平输出
1	1	“H”电平输出

如果在这些模式中不将对应的引脚用作定时器M的输出端口，就必须置“00B”(禁止强制截止)。而且，必须在停止计数的过程中设置这些位。

DFD	DFC	TMIOC引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L”电平输出
1	1	“H”电平输出

如果在这些模式中不将对应的引脚用作定时器M的输出端口，就必须置“00B”(禁止强制截止)。而且，必须在停止计数的过程中设置这些位。

DFB	DFA	TMIOD引脚的脉冲强制截止的控制
0	0	禁止强制截止。
0	1	高阻抗输出
1	0	“L”电平输出
1	1	“H”电平输出

如果在这些模式中不将对应的引脚用作定时器M的输出端口，就必须置“00B”(禁止强制截止)。而且，必须在停止计数的过程中设置这些位。

注：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.11 定时器M控制寄存器i(TMCRi)(i=0、1)

在复位同步PWM模式和PWM3模式中不使用TMCR1寄存器。

图10-15: 定时器M控制寄存器i(TMCRi)(i=0、1)的格式[输入捕捉功能和输出比较功能]

地址: 0x40042A70(TMCR0)、0x40042A80(TMCR1) 复位后: 00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TMi计数器的清除选择
0	0	0	禁止清除(自由运行)。
0	0	1	在TMGRAi的输入捕捉/比较匹配时进行清除。
0	1	0	在TMGRBi的输入捕捉/比较匹配时进行清除。
0	1	1	同步清除(和其他定时器Mi的计数器同时清除) ^{注1}
1	0	1	在TMGRCi的输入捕捉/比较匹配时进行清除。
1	1	0	在TMGRDi的输入捕捉/比较匹配时进行清除。
上述以外			禁止设置。

CKEG1	CKEG0	外部时钟沿的选择 ^{注2}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设置。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	F_{CLK} 、 F_{HOCO} ^{注3}
0	0	1	$F_{CLK}/2$
0	1	0	$F_{CLK}/4$
0	1	1	$F_{CLK}/8$
1	0	0	$F_{CLK}/3$
1	0	1	TMCLK的输入 ^{注4}
上述以外			禁止设置。

注1: 在TMMR寄存器的TMSYNC位为“1”(TM0和TM1同步运行)时有效。

注2: 在TCK2~TCK0位为“101B”(TMCLK的输入)并且STCLK位为“1”(外部时钟输入有效)时有效。

注3: 当用户选项字节(000C2H)的FRQSEL4、FRQSEL3位均为“1”时, 选择 F_{HOCO} ; 其他值时, 选择 F_{CLK} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注4: 在TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)时有效。

图10-16: 定时器M控制寄存器(TMCRi)(i=0、1)的格式[PWM功能]

 地址: 0x40042A70(TMCR0)、0x40042A80(TMCR1) 复位后: 00H^{注1} R/W

符号	7	6	5	4	3	2	1	0
TMCRi	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TMi计数器的清除选择
必须置“001B”(在和TMGRAi寄存器比较匹配时清除TMi寄存器)。			

CKEG1	CKEG0	外部时钟沿的选择 ^{注2}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设置。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	F_{CLK} 、 F_{HOCO} ^{注3}
0	0	1	$F_{CLK}/2$ ^{注4}
0	1	0	$F_{CLK}/4$ ^{注4}
0	1	1	$F_{CLK}/8$ ^{注4}
1	0	0	$F_{CLK}/32$ ^{注4}
1	0	1	TMCLK的输入 ^{注5}
上述以外			禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将 F_{CLK} 设置为 F_{IH} 并且将TMMEN位置“1”后读初始值。

注2: 在TCK2~TCK0位为“101B”(TMCLK的输入)并且STCLK位为“1”(外部时钟输入有效)时有效。

注3: 当用户选项字节(000C2H)的FRQSEL4位为“0”时, 选择 F_{CLK} ; 当FRQSEL4位为“1”时, 选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注4: 不能在用户选项字节(000C2H)的FRQSEL4位为“1”时进行设置。

注5: 在TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)时有效。

图10-17: 定时器M控制寄存器0(TMCR0)的格式[复位同步PWM模式]

地址:	0x40042A70	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TM0计数器的清除选择
必须置“001B”(在和TMGRA0寄存器比较匹配时清除TM0寄存器)。			

CKEG1	CKEG0	外部时钟沿的选择 ^{注2}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设置。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	F_{CLK} 、 F_{HOCO} ^{注3}
0	0	1	$F_{CLK}/2$ ^{注4}
0	1	0	$F_{CLK}/4$ ^{注4}
0	1	1	$F_{CLK}/8$ ^{注4}
1	0	0	$F_{CLK}/32$ ^{注4}
1	0	1	TMCLK的输入 ^{注5}
上述以外			禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将 F_{CLK} 设置为 F_{IH} 并且将TMMEN位置“1”后读初始值。

注2: 在TCK2~TCK0位为“101B”(TMCLK的输入)并且STCLK位为“1”(外部时钟输入有效)时有效。

注3: 当用户选项字节(000C2H)的FRQSEL4位为“0”时, 选择 F_{CLK} ; 当FRQSEL4位为“1”时, 选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注4: 不能在用户选项字节(000C2H)的FRQSEL4位为“1”时进行设置。

注5: 在TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)时有效。

图10-18: 定时器M控制寄存器0(TMCR0)的格式[互补PWM模式]

地址:	0x40042A70	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TM0计数器的清除选择
必须置“000B”(禁止清除(自由运行))。			

CKEG1	CKEG0	外部时钟沿的选择 ^{注2,3}
0	0	在上升沿进行计数。
0	1	在下降沿进行计数。
1	0	在双边沿进行计数。
上述以外		禁止设置。

TCK2	TCK1	TCK0	计数源的选择
0	0	0	F_{CLK} 、 F_{HOCO} ^{注4}
0	0	1	$F_{CLK}/2$ ^{注5}
0	1	0	$F_{CLK}/4$ ^{注5}
0	1	1	$F_{CLK}/8$ ^{注5}
1	0	0	$F_{CLK}/32$ ^{注5}
1	0	1	TMCLK的输入 ^{注6}
上述以外			禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将 F_{CLK} 设置为 F_{IH} 并且将TMMEN位置“1”后读初始值。

注2: 在TCK2~TCK0位为“101B”(TMCLK的输入)并且STCLK位为“1”(外部时钟输入有效)时有效。

注3: 必须给TMCR0寄存器和TMCR1寄存器的TCK0~TCK2位、CKEG0位和CKEG1位设置相同的值。

注4: 当用户选项字节(000C2H)的FRQSEL4位为“0”时, 选择 F_{CLK} ; 当FRQSEL4位为“1”时, 选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注5: 不能在用户选项字节(000C2H)的FRQSEL4位为“1”时进行设置。

注6: 在TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)时有效。

图10-19: 定时器M控制寄存器0(TMCR0)的格式[PWM3模式]

地址:	0x40042A70	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TCK2	TCK1	TCK0

CCLR2	CCLR1	CCLR0	TM0计数器的清除选择
必须置“001B”(在和TMGRA0寄存器比较匹配时清除TM0寄存器)。			

CKEG1	CKEG0	外部时钟沿的选择
在PWM3模式中无效。		

TCK2	TCK1	TCK0	计数源的选择
0	0	0	F_{CLK} 、 F_{HOCO} ^{注2}
0	0	1	$F_{CLK}/2$ ^{注3}
0	1	0	$F_{CLK}/4$ ^{注3}
0	1	1	$F_{CLK}/8$ ^{注3}
1	0	0	$F_{CLK}/32$ ^{注3}
上述以外			禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将 F_{CLK} 设置为 F_{IH} 并且将TMMEN位置“1”后读初始值。

注2: 当用户选项字节(000C2H)的FRQSEL4位为“0”时, 选择 F_{CLK} ; 当FRQSEL4位为“1”时, 选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在定时器开始计数前将 F_{CLK} 设置为 F_{IH} 。

注3: 不能在用户选项字节(000C2H)的FRQSEL4位为“1”时进行设置。

10.3.12 定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1)

图10-20: 定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1)的格式[输入捕捉功能]

地址:	0x40042A71(TMIOA0)、0x40042A81H(TMIOA1)	复位后:	00H ^{注1}	R/W				
符号	7	6	5	4	3	2	1	0
TMIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TMGRB模式的选择 ^{注2}
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOB1	IOB0	TMGRB控制
0	0	在上升沿输入捕捉到TMGRBi。
0	1	在下降沿输入捕捉到TMGRBi。
1	0	在双边沿输入捕捉到TMGRBi。
上述以外		禁止设置。

IOA2	TMGRA模式的选择 ^{注3}
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOA1	IOA0	TMGRA控制
0	0	在上升沿输入捕捉到TMGRAi。
0	1	在下降沿输入捕捉到TMGRAi。
1	0	在双边沿输入捕捉到TMGRAi。
上述以外		禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 如果通过TMMR寄存器的TMBFDi位选择“1”(TMGRBi寄存器的缓冲寄存器), 就必须给TMIORAi寄存器的IOB2位和TMIORCi寄存器的IOD2位设置相同的值。

注3: 如果通过TMMR寄存器的TMBFCi位选择“1”(TMGRAi寄存器的缓冲寄存器), 就必须给TMIORAi寄存器的IOA2位和TMIORCi寄存器的IOC2位设置相同的值。

图10-21：定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1)的格式[输出比较功能]

地址：	0x40042A71(TMIOA0)、0x40042A81H(TMIOA1)		复位后：	00H ^{注1}	R/W			
符号	7	6	5	4	3	2	1	0
TMIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TMGRB模式的选择 ^{注2}
在使用输出比较功能时，必须置“0”(输出比较)。	

IOB1	IOB0	TMGRB控制
0	0	禁止比较匹配的引脚输出(TMIOBi引脚为I/O端口)。
0	1	在TMGRBi比较匹配时输出“L”电平。
1	0	在TMGRBi比较匹配时输出“H”电平。
1	1	在TMGRBi比较匹配时进行交替输出。

IOA2	TMGRA模式的选择 ^{注3}
在使用输出比较功能时，必须置“0”(输出比较)。	

IOA1	IOA0	TMGRA控制
0	0	禁止比较匹配的引脚输出(TMIOAi引脚为I/O端口)。
0	1	在TMGRAi比较匹配时输出“L”电平。
1	0	在TMGRAi比较匹配时输出“H”电平。
1	1	在TMGRAi比较匹配时进行交替输出。

- 注1：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。
- 注2：如果通过TMMR寄存器的TMBFDi位选择“1”(TMGRBi寄存器的缓冲寄存器)，就必须给TMIORAi寄存器的IOB2位和TMIORCi寄存器的IOD2位设置相同的值。
- 注3：如果通过TMMR寄存器的TMBFCi位选择“1”(TMGRAi寄存器的缓冲寄存器)，就必须给TMIORAi寄存器的IOA2位和TMIORCi寄存器的IOC2位设置相同的值。

10.3.13 定时器MI/O控制寄存器Ci(TMIORCi)(i=0、1)

图10-22: 定时器MI/O控制寄存器Ci(TMIORCi)(i=0、1)的格式[输入捕捉功能]

地址:	0x40042A72(TMIORC0)、0x40042A82(TMIORC1)	复位后:	88H ^{注1}	R/W				
符号	7	6	5	4	3	2	1	0
TMIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TMGRD寄存器功能的选择
在使用输入捕捉功能时, 必须置“1”(通用寄存器或者缓冲寄存器)。	

IOD2	TMGRD模式的选择 ^{注2}
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOD1	IOD0	TMGRD控制
0	0	在上升沿输入捕捉到TMGRDi。
0	1	在下降沿输入捕捉到TMGRDi。
1	0	在双边沿输入捕捉到TMGRDi。
上述以外		禁止设置。

IOC3	TMGRC寄存器功能的选择
在使用输入捕捉功能时, 必须置“1”(通用寄存器或者缓冲寄存器)。	

IOC2	TMGRC模式的选择 ^{注3}
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOC1	IOC0	TMGRC控制
0	0	在上升沿输入捕捉到TMGRCi。
0	1	在下降沿输入捕捉到TMGRCi。
1	0	在双边沿输入捕捉到TMGRCi。
上述以外		禁止设置。

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 如果通过TMMR寄存器的TMBFDi位选择“1”(TMGRBi寄存器的缓冲寄存器), 就必须给TMIORAi寄存器的IOB2位和TMIORCi寄存器的IOD2位设置相同的值。

注3: 如果通过TMMR寄存器的TMBFCi位选择“1”(TMGRAi寄存器的缓冲寄存器), 就必须给TMIORAi寄存器的IOA2位和TMIORCi寄存器的IOC2位设置相同的值。

图10-23: 定时器M I/O控制寄存器Ci(TMIORCi)(i=0、1)的格式[输出比较功能]

地址:	0x40042A72(TMIORC0)、0x40042A82(TMIORC1)		复位后:	88H ^{注1}	R/W			
符号	7	6	5	4	3	2	1	0
TMIORCi	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0

IOD3	TMGRD寄存器功能的选择
0	TMIOB输出寄存器 (参照“10.5.2(2)TMGRDi寄存器和TMGRDi寄存器(i=0、1)的输出引脚的变更”)
1	通用寄存器或者缓冲寄存器

IOD2	TMGRD模式的选择 ^{注2}
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOD1	IOD0	TMGRD控制
0	0	禁止比较匹配的引脚输出。
0	1	在TMGRDi比较匹配时输出“L”电平。
1	0	在TMGRDi比较匹配时输出“H”电平。
1	1	在TMGRDi比较匹配时进行交替输出。

IOC3	TMGRD寄存器功能的选择
0	TMIOA输出寄存器 (参照“10.5.2(2)TMGRDi寄存器和TMGRDi寄存器(i=0、1)的输出引脚的变更”)
1	通用寄存器或者缓冲寄存器

IOC2	TMGRD模式的选择 ^{注3}
在使用输出比较功能时, 必须置“0”(输出比较)。	

IOC1	IOC0	TMGRD控制
0	0	禁止比较匹配的引脚输出。
0	1	在TMGRDi比较匹配时输出“L”电平。
1	0	在TMGRDi比较匹配时输出“H”电平。
1	1	在TMGRDi比较匹配时进行交替输出。

- 注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将FCLK设置为FIH并且将TMMEN位置“1”后读初始值。
- 注2: 如果通过TMMR寄存器的TMBFDi位选择“1”(TMGRBi寄存器的缓冲寄存器), 就必须给TMIOA_i寄存器的IOB2位和TMIORC_i寄存器的IOD2位设置相同的值。
- 注3: 如果通过TMMR寄存器的TMBFCi位选择“1”(TMGRA_i寄存器的缓冲寄存器), 就必须给TMIOA_i寄存器的IOA2位和TMIORC_i寄存器的IOC2位设置相同的值。

10.3.14 定时器M状态寄存器0(TMSR0)

图10-24: 定时器M状态寄存器0(TMSR0)的格式[输入捕捉功能]

地址:	0x40042A73	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	上溢标志 ^{注2}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] 当TM0发生上溢时	

IMFD	输入捕捉/比较匹配标志D ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOD0引脚的输入边沿 ^{注4}	

IMFC	输入捕捉/比较匹配标志C ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOC0引脚的输入边沿 ^{注4}	

IMFB	输入捕捉/比较匹配标志B ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOB0引脚的输入边沿 ^{注5}	

IMFA	输入捕捉/比较匹配标志A ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOA0引脚的输入边沿 ^{注5}	

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在定时器M0的计数值从“FFFFH”变为“0000H”时, 上溢标志变为“1”。另外, 根据TMCRO寄存器的CCLR0~CCLR2位的设置, 如果在运行过程中因发生输入捕捉或者比较匹配而使定时器M0的计数值从“FFFFH”变为“0000H”, 上溢标志就变为“1”。

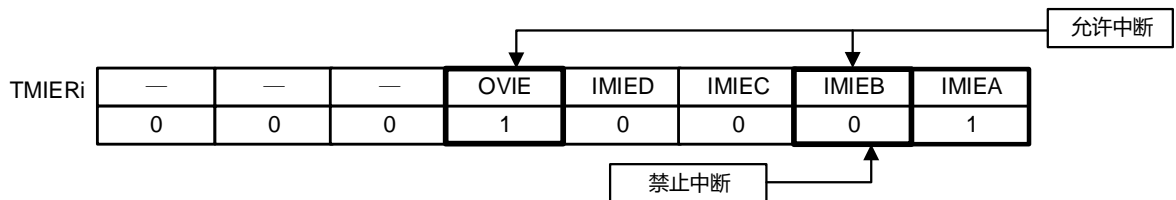
注3: 写的结果如下:

- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。但是, 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器M中断允许寄存器i(TMIERi)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。

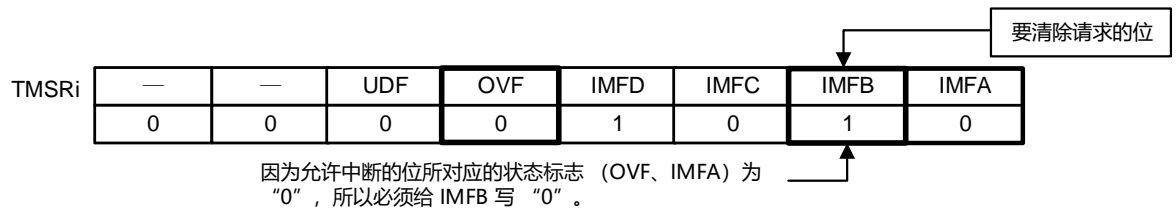
- 必须在将定时器M中断允许寄存器i(TMIERi)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时, 必须给对象状态标志写“0”。

(例)在IMIEA和OVIE为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



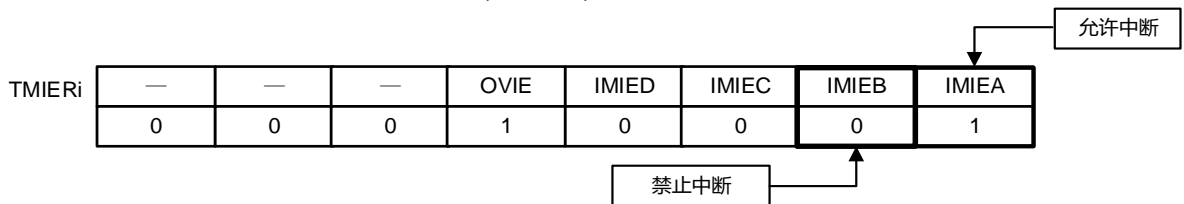
- 定时器M状态寄存器i(TMSRi)的状态



- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时, 必须同时给此状态标志和对象状态标志写“0”。

(例)在IMIEA为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



- 定时器M状态寄存器i(TMSRi)的状态



注4: 这是TMIORC0寄存器的IOk1位和IOk0位(k=C或者D)选择的边沿。

包括TMMR寄存器的TMBFk0位为“1”(TMGRk0为缓冲寄存器)的情况。

注5: 这是TMIORA0寄存器的IOj1位和IOj0位(j=A或者B)选择的边沿。

注6: 当使用DMA时, IMFA位、IMFB位、IMFC位和IMFD位在DMA传送结束后变为“1”。

图10-25: 定时器M状态寄存器0(TMSR0)的格式[输入捕捉以外的功能]

地址:	0x40042A73	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMSR0	0	0	0	OVF	IMFD	IMFC	IMFB	IMFA

OVF	上溢标志 ^{注3}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM0发生上溢时	

IMFD	输入捕捉/比较匹配标志D ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM0和TMGRD0的值相同时 ^{注4}	

IMFC	输入捕捉/比较匹配标志C ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM0和TMGRC0的值相同时 ^{注4}	

IMFB	输入捕捉/比较匹配标志B ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM0和TMGRB0的值相同时	

IMFA	输入捕捉/比较匹配标志A ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM0和TMGRA0的值相同时	

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 写的结果如下:

- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

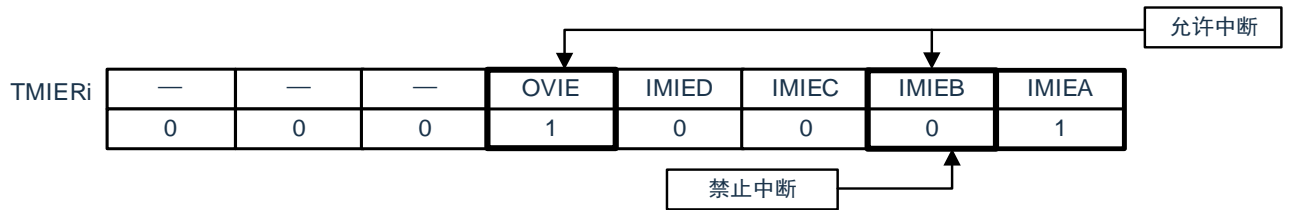
但是, 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器M中断允许寄存器i(TMIERi)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。

- 必须在将定时器M中断允许寄存器i(TMIERi)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志

为“0”时，必须给对象状态标志写“0”。

(例)在IMIEA和OVIE为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



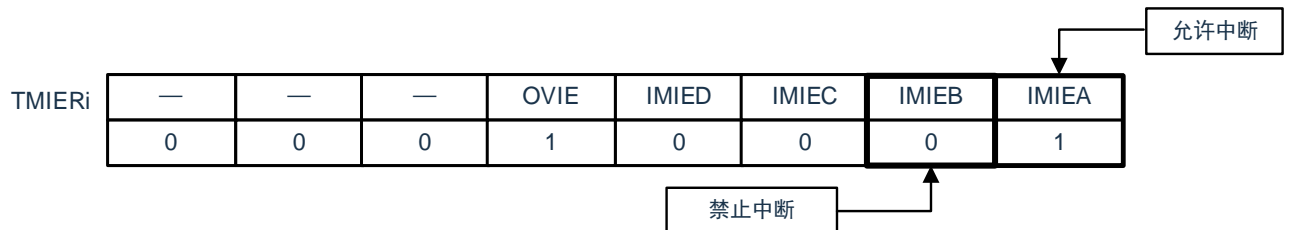
- 定时器M状态寄存器i(TMSRi)的状态



- c) 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(例)在IMIEA为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



- 定时器M状态寄存器i(TMSRi)的状态



注3: 在定时器M0的计数值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据TMCR0寄存器的CCLR0~CCLR2位的设置，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器M0的计数值从“FFFFH”变为“0000H”，上溢标志就变为“1”。

注4: 包括TMMR寄存器的TMBFk0位(k=C或者D)为“1”(TMGRk0为缓冲寄存器)的情况。

注5: 当使用DMA时，IMFA位、IMFB位、IMFC位和IMFD位在DMA传送结束后变为“1”。

10.3.15 定时器M状态寄存器1(TMSR1)

图10-26: 定时器M状态寄存器1(TMSR1)的格式[输入捕捉功能]

地址:	0x40042A83	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	下溢标志
在使用输入捕捉功能时无效。	

OVF	上溢标志 ^{注2}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] 当TM1发生上溢时	

IMFD	输入捕捉/比较匹配标志D ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOD1引脚的输入边沿 ^{注4}	

IMFC	输入捕捉/比较匹配标志C ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOC1引脚的输入边沿 ^{注4}	

IMFB	输入捕捉/比较匹配标志B ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOB1引脚的输入边沿 ^{注5}	

IMFA	输入捕捉/比较匹配标志A ^{注6}
[为“0”的条件] 读后写“0” ^{注3} 。 [为“1”的条件] TMIOA1引脚的输入边沿 ^{注5}	

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 在定时器M1的计数值从“FFFFH”变为“0000H”时, 上溢标志变为“1”。另外, 根据TMCR1寄存器的CCLR0~CCLR2位的设置, 如果在运行过程中因发生输入捕捉或者比较匹配而使定时器M1的计数值从“FFFFH”变为“0000H”, 上溢标志就变为“1”。

注3: 写的结果如下:

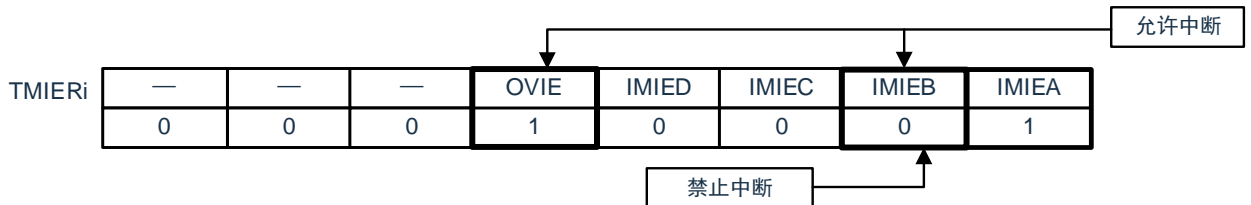
- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

但是, 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器M中断允许寄存器i(TMIERi)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。

- 必须在将定时器M中断允许寄存器i(TMIERi)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时, 必须给对象状态标志写“0”。

(例)在IMIEA和OVIE为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况。

- 定时器M中断允许寄存器i(TMIERi)的状态



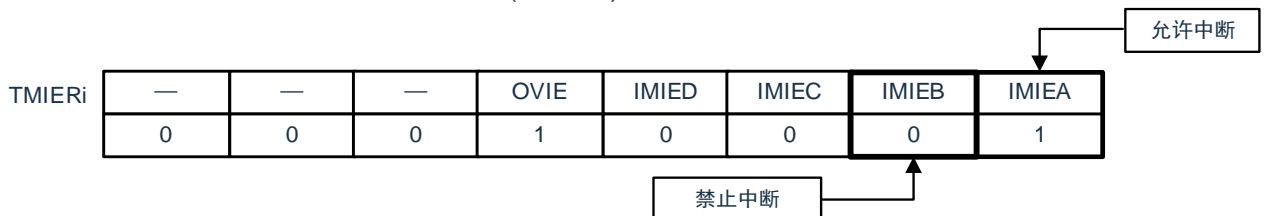
- 定时器M状态寄存器i(TMSRi)的状态



- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时, 必须同时给此状态标志和对象状态标志写“0”。

(例)在IMIEA为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况。

- 定时器M中断允许寄存器i(TMIERi)的状态



- 定时器M状态寄存器i(TMSRi)的状态



注4：这是TMIORC1寄存器的IOk1位和IOk0位(k=C或者D)选择的边沿。

包括TMMR寄存器的TMBFk1位为“1”(TMGRk1为缓冲寄存器)的情况。

注5：这是TMIORA1寄存器的IOj1位和IOj0位(j=A或者B)选择的边沿。

注6：当使用DMA时，IMFA位、IMFB位、IMFC位和IMFD位在DMA传送结束后变为“1”。

图10-27: 定时器M状态寄存器1(TMSR1)的格式[输入捕捉以外的功能]

地址:	0x40042A83	复位后: 00H ^{注1}	R/W					
符号	7	6	5	4	3	2	1	0
TMSR1	0	0	UDF	OVF	IMFD	IMFC	IMFB	IMFA

UDF	下溢标志
互补PWM模式的情况 [为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1发生下溢时在非互补PWM模式中无效。	

OVF	上溢标志 ^{注3}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1发生上溢时	

IMFD	输入捕捉/比较匹配标志D ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1和TMGRD1的值相同时 ^{注4}	

IMFC	输入捕捉/比较匹配标志C ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1和TMGRC1的值相同时 ^{注4}	

IMFB	输入捕捉/比较匹配标志B ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1和TMGRB1的值相同时	

IMFA	输入捕捉/比较匹配标志A ^{注5}
[为“0”的条件] 读后写“0” ^{注2} 。 [为“1”的条件] 当TM1和TMGRA1的值相同时	

注1: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

注2: 写的结果如下:

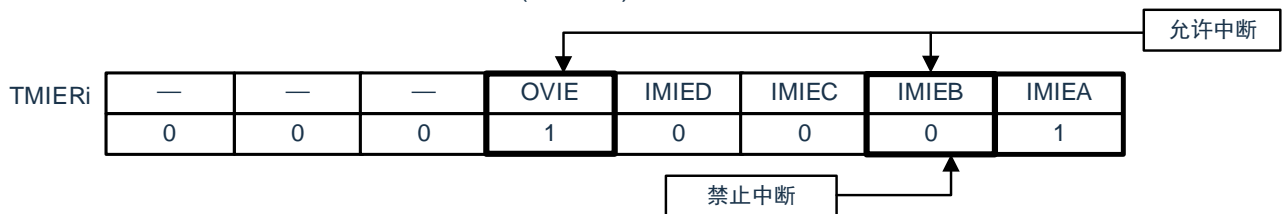
- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

但是, 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器M中断允许寄存器i(TMIERi)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。

- 必须在将定时器M中断允许寄存器i(TMIERi)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时, 必须给对象状态标志写“0”。

(例)在IMIEA和OVIE为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



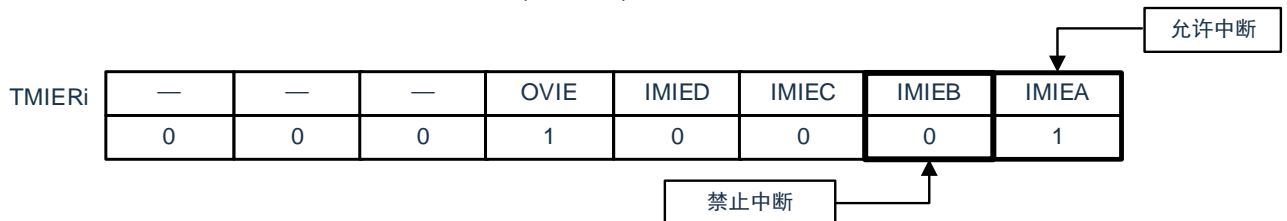
- 定时器M状态寄存器i(TMSRi)的状态



- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时, 必须同时给此状态标志和对象状态标志写“0”。

(例)在IMIEA为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



• 定时器M状态寄存器(TMSRi)的状态



注3: 在定时器M1的计数值从“FFFFH”变为“0000H”时，上溢标志变为“1”。另外，根据TMCR1寄存器的CCLR0~CCLR2位的设置，如果在运行过程中因发生输入捕捉或者比较匹配而使定时器M1的计数值从“FFFFH”变为“0000H”，上溢标志就变为“1”。

注4: 包括TMMR寄存器的TMBFk1位(k=C或者D)为“1”(TMGRk1为缓冲寄存器)的情况。

注5: 当使用DMA时，IMFA位、IMFB位、IMFC位和IMFD位在DMA传送结束后变为“1”。

10.3.16 定时器M中断允许寄存器i(TMIERi)(i=0、1)

图10-28: 定时器M中断允许寄存器i(TMIERi)(i=0、1)的格式

地址:	0x40042A74(TMIER0)、0x40042A84(TMIER1)		复位后: 00H ^注		R/W			
符号	7	6	5	4	3	2	1	0
TMIERi	0	0	0	OVIE	IMIED	IMIEC	IMIEB	IMIEA

OVIE	上溢/下溢中断的允许
0	禁止因OVF位和UDF位产生的中断(OVI)。
1	允许因OVF位和UDF位产生的中断(OVI)。

IMIED	输入捕捉/比较匹配的中断允许D
0	禁止因IMFD位产生的中断(IMID)。
1	允许因IMFD位产生的中断(IMID)。

IMIEC	输入捕捉/比较匹配的中断允许C
0	禁止因IMFC位产生的中断(IMIC)。
1	允许因IMFC位产生的中断(IMIC)。

IMIEB	输入捕捉/比较匹配的中断允许B
0	禁止因IMFB位产生的中断(IMIB)。
1	允许因IMFB位产生的中断(IMIB)。

IMIEA	输入捕捉/比较匹配的中断允许A
0	禁止因IMFA位产生的中断(IMIA)。
1	允许因IMFA位产生的中断(IMIA)。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.17 定时器MPWM功能输出电平控制寄存器i(TMPOCRi)(i=0、1)

只在使用PWM功能时，TMPOCRi寄存器的设置有效，否则TMPOCRi寄存器的设置无效。

图10-29：定时器MPWM功能输出电平控制寄存器i(TMPOCRi)(i=0、1)的格式[PWM功能]

地址：	0x40042A75(TMPOCR0)、0x40042A85(TMPOCR1)	复位后：00H ^注	R/W					
符号	7	6	5	4	3	2	1	0
TMPOCRi	0	0	0	0	0	POLD	POLC	POLB

POLD	PWM功能的输出电平控制D
0	TMIODi输出电平为“L”电平有效。
1	TMIODi输出电平为“H”电平有效。

POLC	PWM功能的输出电平控制C
0	TMIOCi输出电平为“L”电平有效。
1	TMIOCi输出电平为“H”电平有效。

POLB	PWM功能的输出电平控制B
0	TMIOBi输出电平为“L”电平有效。
1	TMIOBi输出电平为“H”电平有效。

注：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.18 定时器M计数器i(TM_i)(i=0、1)

[定时器模式]

必须以16位而不能以8位为单位存取TM_i寄存器。

[复位同步PWM模式和PWM3模式]

必须以16位而不能以8位为单位存取TM₀寄存器。在复位同步PWM模式和PWM3模式中，不使用TM₁寄存器。

[互补PWM模式(TM₀)]

必须以16位而不能以8位为单位存取TM₀寄存器。

[互补PWM模式(TM₁)]

必须以16位而不能以8位为单位存取TM₁寄存器。

图10-30：定时器M计数器i(TM_i)(i=0、1)的格式[定时器模式]



—	功能	设置范围
bit15~0	对计数源进行递增计数。 如果发生上溢，TMSR _i 寄存器的OVF位就变为“1”。	0000H~FFFFH

注：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

图10-31：定时器M计数器i(TM_i)(i=0、1)的格式[复位同步PWM模式和PWM3模式]



—	功能	设置范围
bit15~0	对计数源进行递增计数。 如果发生上溢，TMSR ₀ 寄存器的OVF位就变为“1”。	0000H~FFFFH

注：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

图10-32: 定时器M计数器i(TMi)(i=0、1)的格式[互补PWM模式(TM0)]

地址: 0x40042A76(TM0)、0x40042A86(TM1) 复位后: 0000H[±] R/W

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

TMi

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

—	功能	设置范围
bit15~0	必须设置死区时间。 对计数源进行递增计数或者递减计数。 如果发生上溢, TMSR0寄存器的OVF位就变为“1”。	0001H~FFFFH

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

图10-33: 定时器M计数器i(TMi)(i=0、1)的格式[互补PWM模式(TM1)]

地址: 0x40042A76(TM0)、0x40042A86(TM1) 复位后: 0000H[±] R/W

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

TMi

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

—	功能	设置范围
bit15~0	必须置“0000H”。对计数源进行递增计数或者递减计数。 如果发生上溢, TMSR1寄存器的UDF位就变为“1”。	0000H~FFFFH

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

10.3.19 定时器M通用寄存器Ai、Bi、Ci、Di(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)

[输入捕捉功能]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在使用输入捕捉功能时，以下寄存器无效：

TMOER1、TMOER2、TMOCR、TMPOCR0、TMPOCR1

当不使用数字滤波器时(TMDFi寄存器的DFj位为“0”)，TMIOji引脚输入的捕捉信号的脉宽必须至少为3个定时器M的运行时钟(F_{CLK})周期。

[输出比较功能]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在使用输出比较功能时，以下寄存器无效：

TMDF0、TMDF1、TMPOCR0、TMPOCR1

[PWM功能]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在使用PWM功能时，以下寄存器无效：

TMDF0、TMDF1、TMIORA0、TMIORC0、TMIORA1、TMIORC1

[复位同步PWM模式]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在复位同步PWM模式中，以下寄存器无效：

TMPMR、TMOCR^注、TMDF0、TMDF1、TMIORA0、TMIORC0、TMPOCR0、TMIORA1、TMIORC1、TMPOCR1

注：作为复位同步PWM模式和互补PWM模式中的TMIOC0的初始输出设置，只有TMOCR寄存器的TOC0位有效。

[互补PWM模式]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在互补PWM模式中，不使用TMGRC0寄存器。

在互补PWM模式中，以下寄存器无效：

TMPMR、TMOCR^注、TMDF0、TMDF1、TMIORA0、TMIORC0、TMPOCR0、TMIORA1、TMIORC1、TMPOCR1

注：作为复位同步PWM模式和互补PWM模式中的TMIOC0的初始输出设置，只有TMOCR寄存器的TOC0位有效。

因为不能在开始计数后直接写TMGRB0、TMGRA1、TMGRB1寄存器(禁止事项)，所以必须将TMGRD0、TMGRC1、TMGRD1寄存器用作缓冲寄存器。

但是，要写TMGRD0、TMGRC1、TMGRD1寄存器时，在将TMBFD0位、TMBFC1位和TMBFD1位置“0”(通用寄存器)后写这些寄存器。此后，能将TMBFD0位、TMBFC1位和TMBFD1位置“1”(缓冲寄存器)。

[PWM3模式]

必须以16位而不能以8位为单位存取TMGRAi~TMGRDi寄存器。在PWM3模式中，以下寄存器无效：

TMPMR、TMDf0、TMDf1、TMIORA0、TMIORC0、TMPOCR0、TMIORA1、TMIORC1、TMPOCR1在PWM3模式中，不使用TMGRC0、TMGRC1、TMGRD0、TMGRD1寄存器。但是，要将这些寄存器用作缓冲寄存器时，在将TMBFC0位、TMBFC1位、TMBFD0位和TMBFD1位置“0”(通用寄存器)后给TMGRC0、TMGRC1、TMGRD0、TMGRD1寄存器写值。此后，能将TMBFC0位、TMBFC1位、TMBFD0位和TMBFD1位置“1”(缓冲寄存器)。

图10-34：定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[输入捕捉功能]

地址： 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后：FFFFH^注 R/W
 0x40042B58(TMGRC0)、0x40042B5A(TMGRD0)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRC1)、0x40042B5E(TMGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi																
TMGRBi																
TMGRCi																
TMGRDi																

—	功能
bit15~0	参照“表10-3：使用输入捕捉功能时的TMGRji寄存器功能”。

注：当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时，复位后的值为不定值。当需要读初始值时，必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

表10-3：使用输入捕捉功能时的TMGRji寄存器功能

寄存器	设置	寄存器功能	输入捕捉的输入引脚
TMGRAi	—	通用寄存器，能读输入捕捉时的Tmi寄存器的值。	TMIOAi
TMGRBi			TMIOBi
TMGRCi	TMBFCi=0	通用寄存器，能读输入捕捉时的Tmi寄存器的值。	TMIOCi
TMGRDi	TMBFDi=0		TMIODi
TMGRCi	TMBFCi=1	通用寄存器，能读输入捕捉时的Tmi寄存器的值(参照“10.4.2 缓冲器运行”)。	TMIOAi
TMGRDi	TMBFDi=1		TMIOBi

备注：i=0、1，j=A、B、C、D

TMBFCi、TMBFDi：TMMR寄存器的位

图10-35: 定时器M通用寄存器Ai、Bi、Ci、Di

 (TMGRA_i、TMGRB_i、TMGRC_i、TMGRD_i)(i=0、1)的格式[输出比较功能]

 地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH^注 R/W
 0x40042B58(TMGRC0)、0x40042B5A(TMGRD0)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRC1)、0x40042B5E(TMGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRA _i																
TMGRB _i																
TMGRC _i																
TMGRD _i																

—	功能
bit15~0	参照“表10-4: 使用输出比较功能时的TMGR _{ji} 寄存器功能”。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

 表10-4: 使用输出比较功能时的TMGR_{ji}寄存器功能

寄存器	设置		寄存器功能	输出比较的输出引脚
	TMBF _{ki}	IO _{j3}		
TMGRA _i	—	—	通用寄存器, 必须写比较值。	TMIOA _i
TMGRB _i				TMIOB _i
TMGRC _i	0	1	通用寄存器, 必须写比较值。	TMIOC _i
TMGRD _i				TMIOD _i
TMGRC _i	1	1	缓冲寄存器, 必须写下一个比较值 (参照“10.4.2缓冲器运行”)。	TMIOA _i
TMGRD _i				TMIOB _i
TMGRC _i	0	0	TMIOA _i 输出控制	TMIOA _i
TMGRD _i			TMIOB _i 输出控制 (参照“10.5.2(2)TMGRC _i 寄存器和TMGRD _i 寄存器(i=0、1)的输出引脚的变更”)	

注意: 如果将TMCR_i寄存器的TCK2~TCK0位置“000B”(F_{CLK}、F_{HOCO})并且将比较值置“0000H”, 就只在开始计数后立即向DMA和ELC产生1次请求信号。如果比较值大于等于“0001H”, 就在每次比较匹配时产生请求信号。

备注: i=0、1, j=A、B、C、D, k=C、D

TMBF_{ki}: TMMR寄存器的位, IO_{j3}: TMIORC_i寄存器的位

图10-36: 定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[PWM功能]

 地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH^注 R/W
 0x40042B58(TMGRc0)、0x40042B5A(TMGRD0)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRc1)、0x40042B5E(TMGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi																
TMGRBi																
TMGRci																
TMGRDi																

—	功能
bit15~0	参照“表10-5: 使用PWM功能时的TMGRji寄存器功能”。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

表10-5: 使用PWM功能时的TMGRji寄存器功能

寄存器	设置	寄存器功能	PWM输出引脚
TMGRAi	—	通用寄存器, 必须设置PWM周期。	—
TMGRBi	—	通用寄存器, 必须设置PWM输出的变化点。	TMIOBi
TMGRci	TMBFCi=0	通用寄存器, 必须设置PWM输出的变化点。	TMIOci
TMGRDi	TMBFDi=0		TMIODi
TMGRci	TMBFCi=1	缓冲寄存器, 必须设置下一个PWM周期(参照“10.4.2缓冲器运行”)。	—
TMGRDi	TMBFDi=1	缓冲寄存器, 必须设置下一个PWM输出的变化点(参照“10.4.2缓冲器运行”)。	TMIOBi

注意: 如果将TMCRI寄存器的TCK2~TCK0位置“000B”(F_{CLK}、F_{HOCO})并且将比较值置“0000H”, 就只在开始计数后立即向DMA和ELC产生1次请求信号。如果比较值大于等于“0001H”, 就在每次比较匹配时产生请求信号。

备注: i=0、1, j=A、B、C、D

TMBFCi、TMBFDi: TMMR寄存器的位

图10-37: 定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[复位同步PWM模式]

 地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH^注 R/W
 0x40042B58(TMGRD0)、0x40042B5A(TMGRD1)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRD1)、0x40042B5E(TMGRD1)

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

—	功能
bit15~0	参照“表10-6: 复位同步PWM模式中的TMGRji寄存器功能”。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

表10-6: 复位同步PWM模式中的TMGRji寄存器功能

寄存器	设置	寄存器功能	PWM输出引脚
TMGRA0	—	通用寄存器, 必须设置PWM周期。	(TMIOC0在每个PWM周期进行反相输出)
TMGRB0	—	通用寄存器, 必须设置PWM1输出的变化点。	TMIOB0TMIOD0
TMGRD0	TMBFD0=0	(在复位同步PWM模式中不使用)	—
TMGRD0	TMBFD0=0		
TMGRA1	—	通用寄存器, 必须设置PWM2输出的变化点。	TMIOA1TMIOC1
TMGRB1	—	通用寄存器, 必须设置PWM3输出的变化点。	TMIOB1TMIOD1
TMGRD1	TMBFD1=0	(在复位同步PWM模式中不使用)	—
TMGRD1	TMBFD1=0		
TMGRD0	TMBFD0=1	缓冲寄存器, 必须设置下一个PWM1输出的变化点(参照“10.4.2缓冲器运行”)。	TMIOB0TMIOD0
TMGRD1	TMBFD1=1	缓冲寄存器, 必须设置下一个PWM3输出的变化点(参照“10.4.2缓冲器运行”)。	TMIOB1TMIOD1
TMGRD0	TMBFD0=1	缓冲寄存器, 必须设置下一个PWM1输出的变化点(参照“10.4.2缓冲器运行”)。	TMIOB0TMIOD0
TMGRD1	TMBFD1=1	缓冲寄存器, 必须设置下一个PWM3输出的变化点(参照“10.4.2缓冲器运行”)。	TMIOB1TMIOD1

注意: 如果将TMCR0寄存器的TCK2~TCK0位置“000B”(F_{CLK}、F_{HOCO})并且将比较值置“0000H”, 就只在开始计数后立即向DMA和ELC产生1次请求信号。如果比较值大于等于“0001H”, 就在每次比较匹配时产生请求信号。

备注: i=0、1, j=A、B、C、D

TMBFC0、TMBFD0、TMBFC1、TMBFD1: TMMR寄存器的位

图10-38: 定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[互补PWM模式]

地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH^注 R/W
 0x40042B58(TMGRD0)、0x40042B5A(TMGRD1)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRD1)、0x40042B5E(TMGRD1)

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

—	功能
bit15~0	参照“表10-7: 互补PWM模式中的TMGR _{ji} 寄存器功能”。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

表10-7: 互补PWM模式中的TMGR_{ji}寄存器功能

寄存器	设置	寄存器功能	PWM输出引脚
TMGRA0	—	通用寄存器, 必须在初始设置时设置PWM周期。 设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRA0的设置值 ≤ FFFFH-TM0寄存器的设置值 当TMSTR寄存器的TSTART0位和TSTART1位为“1”(开始计数)时, 不能写此寄存器。	(TMIOC0在每半个周期进行反相输出)
TMGRB0	—	通用寄存器, 必须在初始设置时设置PWM1输出的变化点。设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRB0的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 当TMSTR寄存器的TSTART0位和TSTART1位为“1”(开始计数)时, 不能写此寄存器。	TMIOB0TMIOD0
TMGRA1	—	通用寄存器, 必须在初始设置时设置PWM2输出的变化点。设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRA1的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 当TMSTR寄存器的TSTART0位和TSTART1位为“1”(开始计数)时, 不能写此寄存器。	TMIOA1TMIOC1
TMGRB1	—	通用寄存器, 必须在初始设置时设置PWM3输出的变化点。设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRB1的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 当TMSTR寄存器的TSTART0位和TSTART1位为“1”(开始计数)时, 不能写此寄存器。	TMIOB1TMIOD1
TMGRC0	—	(在互补PWM模式中不使用)	—
TMGRD0	TMBFD0=1	缓冲寄存器, 必须设置下一个PWM1输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRD0的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 在初始设置时必须设置TMGRB0寄存器的相同值。	TMIOB0TMIOD0
TMGRC1	TMBFC1=1	缓冲寄存器, 必须设置下一个PWM2输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRC1的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 在初始设置时必须设置TMGRA1寄存器的相同值。	TMIOA1TMIOC1
TMGRD1	TMBFD1=1	缓冲寄存器, 必须设置下一个PWM3输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围: TM0寄存器的设置值(计数初始值) ≤ TMGRD1的设置值 ≤ TMGRA0的设置值-TM0寄存器的设置值 在初始设置时必须设置TMGRB1寄存器的相同值。	TMIOB1TMIOD1

注意: 如果将TMCRI寄存器的TCK2~TCK0位置“000B”(F_{CLK}、F_{HOCO})并且将比较值置“0000H”, 就只在开始计数后立即向DMA和ELC产生1次请求信号。如果比较值大于等于“0001H”, 就在每次比较匹配时产生请求信号。

备注: i=0、1, j=A、B、C、D

TMBFD0、TMBFC1、TMBFD1: TMMR寄存器的位

图10-39: 定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[PWM3模式]

地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH^注 R/W
 0x40042B58(TMGRD0)、0x40042B5A(TMGRD1)、
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、
 0x40042B5C(TMGRD1)、0x40042B5E(TMGRD1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMGRAi																
TMGRBi																
TMGRCi																
TMGRDi																

—	功能
bit15~0	参照“表10-8: PWM3模式中的TMGRji寄存器功能”。

注: 当用户选项字节(000C2H)的FRQSEL4位为“1”并且PER1寄存器的TMMEN位为“0”时, 复位后的值为不定值。当需要读初始值时, 必须在将F_{CLK}设置为F_{IH}并且将TMMEN位置“1”后读初始值。

表10-8: PWM3模式中的TMGR_{ji}寄存器功能

寄存器	设置	寄存器功能	PWM输出引脚
TMGRA0	—	通用寄存器，必须设置PWM周期。 设置范围：TMGRA1寄存器的设置值 ≤ TMGRA0的设置值	TMIOA0
TMGRA1		通用寄存器，必须设置PWM输出的变化点(变为有效电平的时序)。设置范围：TMGRA1的设置值 ≤ TMGRA0寄存器的设置值	
TMGRB0		通用寄存器，必须设置PWM输出的变化点(返回到初始输出电平的时序)。 设置范围：TMGRB1寄存器的设置值 ≤ TMGRB0的设置值 ≤ TMGRA0寄存器的设置值	TMIOB0
TMGRB1		通用寄存器，必须设置PWM输出的变化点(变为有效电平的时序)。设置范围：TMGRB1的设置值 ≤ TMGRB0寄存器的设置值	
TMGRC0	TMBFC0=0	(在PWM3模式中不使用)	—
TMGRC1	TMBFC1=0		
TMGRD0	TMBFD0=0		
TMGRD1	TMBFD1=0		
TMGRC0	TMBFC0=1	缓冲寄存器，必须设置下一个PWM周期(参照“10.4.2缓冲器运行”)。设置范围：TMGRC1寄存器的设置值 ≤ TMGRC0的设置值	TMIOA0
TMGRC1	TMBFC1=1	缓冲寄存器，必须设置下一个PWM输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围：TMGRC1的设置值 ≤ TMGRC0寄存器的设置值	
TMGRD0	TMBFD0=1	缓冲寄存器，必须设置下一个PWM输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围：TMGRD1寄存器 ≤ TMGRD0的设置值 ≤ TMGRC0寄存器的设置值	TMIOB0
TMGRD1	TMBFD1=1	缓冲寄存器，必须设置下一个PWM输出的变化点(参照“10.4.2缓冲器运行”)。 设置范围：TMGRD1的设置值 ≤ TMGRD0寄存器的设置值	

注意：如果将TMCR0寄存器的TCK2~TCK0位置“000B”(F_{CLK}、F_{HOCO})并且将比较值置“0000H”，就只在开始计数后立即向DMA和EVENTC产生1次请求信号。如果比较值大于等于“0001H”，就在每次比较匹配时产生请求信号。

备注：i=0、1，j=A、B、C、D

TMBFC0、TMBFD0、TMBFC1、TMBFD1：TMMR寄存器的位

10.3.20 端口模式寄存器(PMxx, PMCxx)

这是设置端口的输入/输出或者模拟输入的寄存器。

在将定时器输出引脚的复用端口(Pxx/TMIOD1、Pxx/TMIOC1等)用作定时器的输出时，必须将各端口对应的端口模式寄存器(PMxx, PMCxx)的位和端口寄存器(Pxx)的位置“0”。

(例)P10/TMIOD1用作定时器输出的情况

将端口模式寄存器1的PM10和PMC10位置“0”。

将端口寄存器1的P10位置“0”。

在将定时器输入引脚的复用端口(P10/TMIOD1、P11/TMIOC1等)用作定时器的输入时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”以及各端口对应的端口模式寄存器(PMCxx)的位置“0”。此时，端口寄存器(Pxx)的位可以是“0”或者“1”。

(例)P10/TMIOD1用作定时器输入的情况

将端口模式寄存器1的PM10位置“1”。

将端口模式寄存器1的PMC10位置“0”。

将端口寄存器1的P10位置“0”或者“1”。

通过8位存储器操作指令设置PM1和PMC1寄存器。在产生复位信号后，寄存器的值变为“FFH”。

图10-40: 端口模式寄存器1(PM1,PMC1)的格式

地址: 0x40040321 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PMmn	Pmn 引脚的输入/输出模式的选择(m=1, n=0~7)
0	输出模式(输出缓冲器ON)
1	输入模式(输出缓冲器OFF)

地址: 0x40040061 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10

PMCmn	Pmn引脚的输入/输出模式的选择(m=1, n=0~7)
0	数字输入/输出(模拟输入以外的复用功能)
1	模拟输入

10.4 有关多个模式的共同事项

10.4.1 计数源

所有模式的计数源的选择方法都一样。但是，在PWM3模式中，不能选择外部时钟。

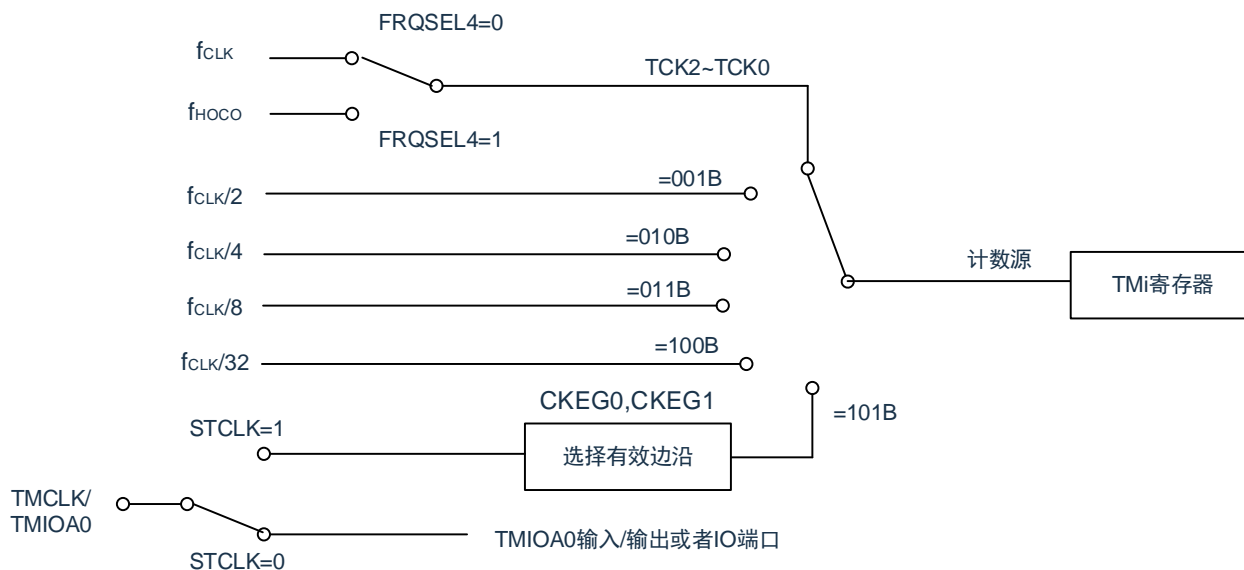
表10-9: 计数源的选择

计数源	选择方法
F_{CLK} 、 F_{HOCO} ^注 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$	通过TMCRi寄存器的TCK2~TCK0位选择计数源。
TMCLK引脚的外部输入信号	TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)。 TMCRi寄存器的TCK2~TCK0位为“101B”(计数源为外部时钟)。通过TMCRi寄存器的CKEG1~CKEG0位选择有效边沿。 TMCLK引脚的复用I/O端口的端口模式寄存器的位为“1”(输入模式)。

备注: $i=0, 1$

注: 当用户选项字节(000C2H)的FRQSEL4位为“0”时, 选择 F_{CLK} ; 当FRQSEL4位为“1”时, 选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

图10-41: 计数源的框图



备注: $i=0, 1$

TMCLK引脚输入的外部时钟的脉宽必须至少为3个定时器M的运行时钟(F_{CLK})周期。

10.4.2 缓冲器运行

能通过TMMR寄存器的TMBFCi(i=0、1)位和TMBFDi位，将TMGRCi寄存器和TMGRDi寄存器分别设置为TMGRAi寄存器和TMGRBi寄存器的缓冲寄存器。

- TMGRAi的缓冲寄存器：TMGRCi寄存器
- TMGRBi的缓冲寄存器：TMGRDi寄存器

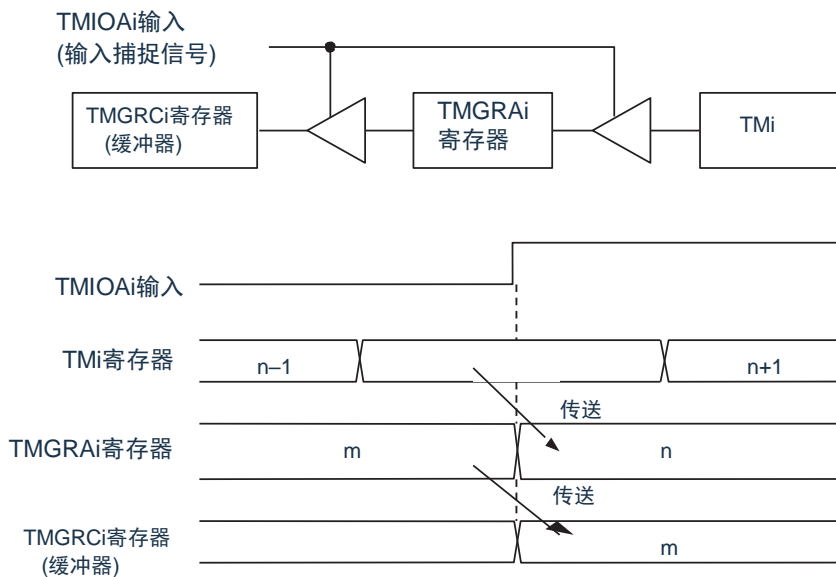
缓冲器运行因定时器模式而不同，各模式的缓冲器运行如表10-10所示。

表10-10：各模式的缓冲器运行

功能和模式		传送时序	传送的寄存器
定时器模式	输入捕捉功能	TMIOAi的输入信号 (输入捕捉信号的输入)	将TMGRAi寄存器的内容传送到TMGRCi寄存器(缓冲寄存器)。
		TMIOBi的输入信号 (输入捕捉信号的输入)	将TMGRBi寄存器的内容传送到TMGRDi寄存器(缓冲寄存器)。
	输出比较功能	TMi寄存器和TMGRAi寄存器的比较匹配	将TMGRCi寄存器(缓冲寄存器)的内容传送到TMGRAi寄存器。
		TMi寄存器和TMGRBi寄存器的比较匹配	将TMGRDi寄存器(缓冲寄存器)的内容传送到TMGRBi寄存器。
	PWM功能	TMi寄存器和TMGRAi寄存器的比较匹配	将TMGRCi寄存器(缓冲寄存器)的内容传送到TMGRAi寄存器。
		TMi寄存器和TMGRBi寄存器的比较匹配	将TMGRDi寄存器(缓冲寄存器)的内容传送到TMGRBi寄存器。
复位同步PWM模式	TM0寄存器和TMGRA0寄存器的比较匹配	将TMGRCi寄存器(缓冲寄存器)的内容传送到TMGRAi寄存器。 将TMGRDi寄存器(缓冲寄存器)的内容传送到TMGRBi寄存器。	
互补PWM模式	<ul style="list-style-type: none"> • 当TMFCR寄存器的CMD1位和CMD0位是“11B”时，为TM1寄存器的下溢。 • 当TMFCR寄存器的CMD1位和CMD0位是“10B”时，为TM0寄存器和TMGRA0寄存器的比较匹配。 	将TMGRC1寄存器(缓冲寄存器)的内容传送到TMGRA1寄存器。 将TMGRDi寄存器(缓冲寄存器)的内容传送到TMGRBi寄存器。	
PWM3模式	TM0寄存器和TMGRA0寄存器的比较匹配	将TMGRCi寄存器(缓冲寄存器)的内容传送到TMGRAi寄存器。 将TMGRDi寄存器(缓冲寄存器)的内容传送到TMGRBi寄存器。	

备注：i=0、1

图10-42: 输入捕捉功能的缓冲器运行

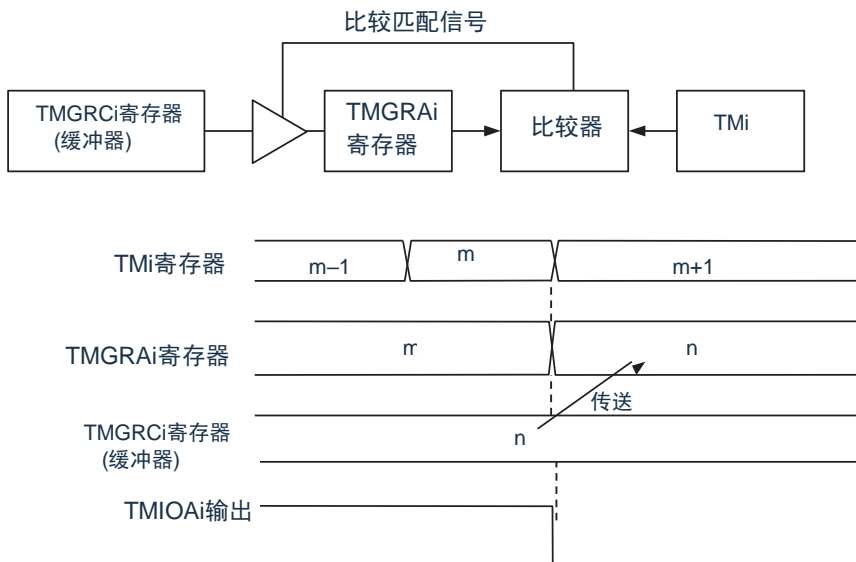


备注: $i=0, 1$

上图的条件如下:

- TMMR寄存器的TMBFCi位为“1”(TMGRci寄存器为TMGRAi寄存器的缓冲寄存器)
- TMIOAi寄存器的IOA2~IOA0位为“100B”(在上升沿输入捕捉)

图10-43: 输出比较功能的缓冲器运行



备注: $i=0, 1$

上图的条件如下:

- TMMR寄存器的TMBFCi位为“1”(TMGRci寄存器为TMGRAi寄存器的缓冲寄存器)
- TMIOAi寄存器的IOA2~IOA0位为“001B”(在比较匹配时, 输出“L”电平)

在定时器模式(输入捕捉功能和输出比较功能)中,必须进行以下的设置。将TMGR*Ci*(*i*=0、1)寄存器用作TMGR*Ai*寄存器的缓冲寄存器的情况:

- 必须将TMIOR*Ci*寄存器的IOC3位置“1”(通用寄存器或者缓冲寄存器)。
- 必须给TMIOR*Ci*寄存器的IOC2位和TMIOR*Ai*寄存器的IOA2位设置相同的值。

将TMGRD*i*寄存器用作TMGRB*i*寄存器的缓冲寄存器的情况:

- 必须将TMIOR*Ci*寄存器的IOD3位置“1”(通用寄存器或者缓冲寄存器)。
- 必须给TMIOR*Ci*寄存器的IOD2位和TMIOR*Ai*寄存器的IOB2位设置相同的值。

在使用输入捕捉功能时,即使将TMGR*Ci*寄存器和TMGRD*i*寄存器用作缓冲寄存器,在TMIOC*i*引脚和TMIOD*i*引脚的输入边沿,TMSR*i*寄存器的IMFC位和IMFD位也变为“1”。

当使用输出比较功能或者PWM功能时或者在复位同步PWM模式、互补PWM模式和PWM3模式中,即使将TMGR*Ci*寄存器和TMGRD*i*寄存器用作缓冲寄存器,在和TMI寄存器比较匹配时,TMSR*i*寄存器的IMFC位和IMFD位也变为“1”。

10.4.3 同步运行

使TM0寄存器与TM1寄存器同步。

- 同步预置

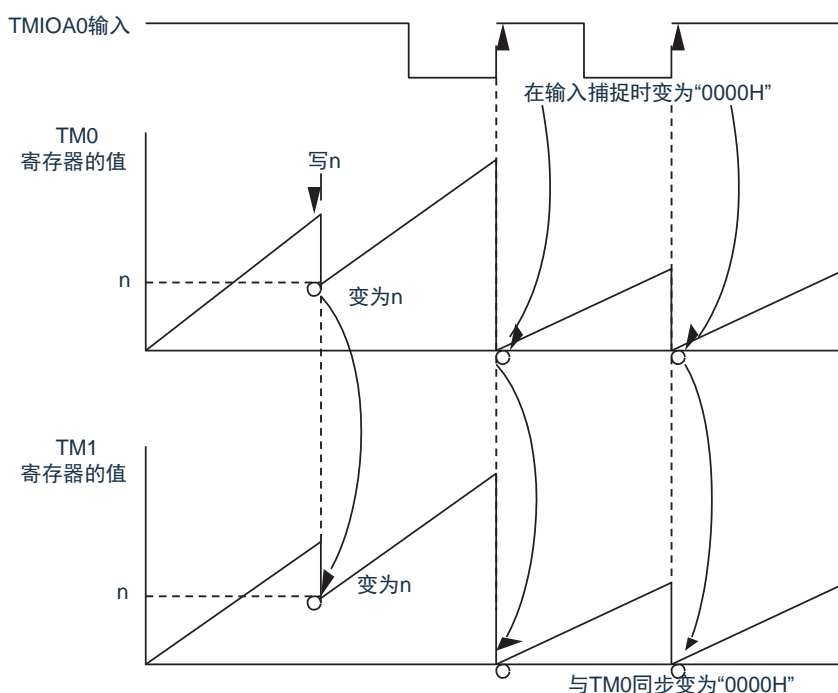
如果在TMMR寄存器的TMSYNC位为“1”(同步运行)时写Tm_i寄存器，数据就同时被写到TM0寄存器和TM1寄存器。

- 同步清除

当TMMR寄存器的TMSYNC位为“1”并且TMCR0寄存器的CCLR2~CCLR0位为“011B”(同步清除)时，TM0寄存器和TM1寄存器同时变为“0000H”。

同样，当TMMR寄存器的TMSYNC位为“1”并且TMCR1寄存器的CCLR2~CCLR0位为“011B”(同步清除)时，TM1寄存器和TM0寄存器同时变为“0000H”。

图10-44：同步运行



上图的条件如下：

TMMR寄存器的TMSYNC位为“1”(同步运行)

TMCR0寄存器的CCLR2~CCLR0位为“001B”(在输入捕捉时，将TM0置“0000H”)

TMCR1寄存器的CCLR2~CCLR0位为“011B”(与TM0同步，将TM1置“0000H”)

TMIOA0寄存器的IOA2~IOA0位为“100B”

TMFCR寄存器的CMD1位和CMD0位为“00B” } (在TMIOA0输入的上升沿输入捕捉)

TMFCR寄存器的PWM3位为“1”

10.4.4 脉冲输出的强制截止

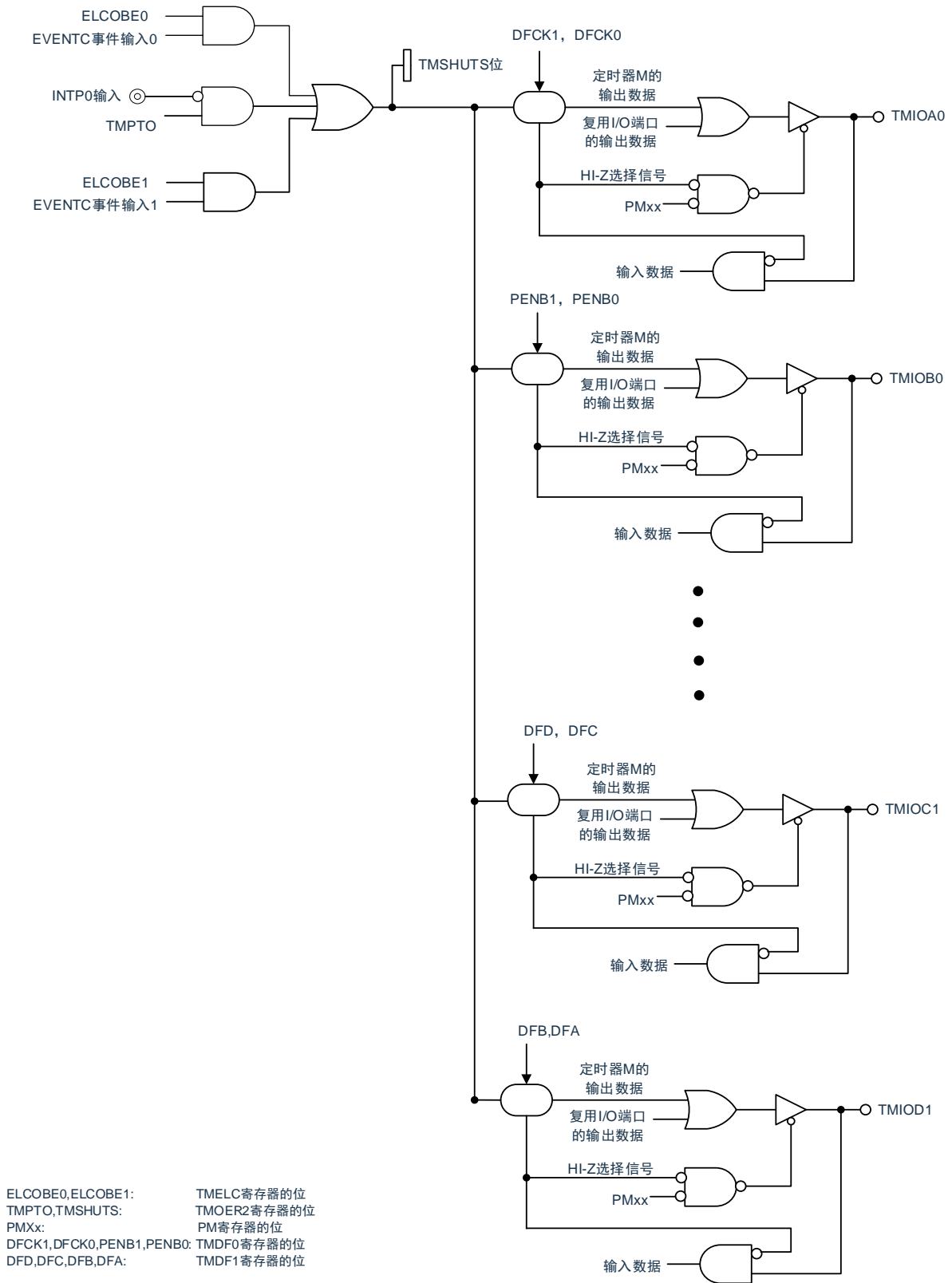
当使用PWM功能时或者在复位同步PWM模式、互补PWM模式和PWM3模式中，能通过INTP0引脚的输入来截止TMIO_j输出引脚($i=0, 1, j=A, B, C, D$)的脉冲输出。

如果将TMOER1寄存器的对应位置“0”(允许定时器M的输出)，就将这些功能或者模式中使用的输出引脚用作定时器M的输出引脚。当TMOER2寄存器的TMPTO位为“1”(脉冲输出强制截止信号INTP0引脚输入有效)时，TMDF0寄存器或者TMDF1寄存器的DFCK1、DFCK0、PENB1、PENB0、DFD、DFC、DFB、DFA位设置的输出值从用作定时器M输出端口的输出引脚输出。

在使用此功能时，必须进行以下的设置：

- 通过TMDF_i寄存器设置脉冲输出被强制截止时的引脚状态(高阻抗、“L”电平输出或者“H”电平输出)。
- 有关通过EVENTC事件输入进行的脉冲强制截止，请参照“10.4.5从事件联动控制器(EVENTC)输入的事件”。
- 在强制截止脉冲的输出时，TMOER2寄存器的TMSHUTS位为“1”。要中止强制截止脉冲的输出时，必须在停止计数的过程中($TSTART_i=0$)将TMSHUTS位置“0”。
- 将TMOER2寄存器的TMPTO位置“1”(脉冲输出强制截止信号INTP0引脚输入有效)。

图10-45：脉冲输出的强制截止



10.4.5 从事件联动控制器(EVENTC)输入的事件

通过EVENTC输入的事件，定时器M进行2种运行。

(a) TMIOD0/TMIOD1的输入捕捉

通过EVENTC输入的事件，定时器M进行TMIOD0/TMIOD1的输入捕捉。此时，TMSRi寄存器的IMFD位为“1”。

要使用此功能时，必须选择定时器模式的输入捕捉功能，并且将TMELC寄存器的ELCICE0位或者ELCICE1位置“1”。在其他模式(定时器模式的输出比较功能、PWM功能、复位同步PWM模式、互补PWM模式、PWM3模式)中，此功能无效。

(b) 强制截止脉冲输出的运行^注

通过EVENTC输入的事件，强制截止脉冲的输出。要使用此功能时，必须选择脉冲输出模式(PWM功能、复位同步PWM模式、互补PWM模式、PWM3模式)并且将ELCOBE0位或者ELCOBE1位置“1”。在使用定时器模式的输入捕捉功能时，此功能无效。

注：INTP0引脚的强制截止功能截止“L”电平输入期间的脉冲输出，但是EVENTC事件的强制截止功能对于EVENTC输入的1次事件，截止1次脉冲输出。

设置步骤

- 1) 将EVENTC事件联动目标设置为定时器M。
- 2) 将TMELC寄存器的ELCICEi位(i=0、1)和ELCOBEi位(i=0、1)置“1”。

10.4.6 向事件联动控制器(EVENTC)/数据传送控制器(DMA)输出的事件

定时器M的模式和向EVENTC/DMA输出的事件如表10-11所示。

表10-11: 定时器M的模式和向ELC/DMA输出的事件

使用模式	输出源	EVENTC	DMA
输入捕捉功能	通过TMIORA0寄存器的IOA1位和IOA0位设置的TMIOA0边沿检测	○	○
	通过TMIORA0寄存器的IOB1位和IOB0位设置的TMIOB0边沿检测	○	○
	通过TMIORC0寄存器的IOC1位和IOC0位设置的TMIOC0边沿检测	—	○
	通过TMIORC0寄存器的IOD1位和IOD0位设置的TMIOD0边沿检测	—	○
	通过TMIORA1寄存器的IOA1位和IOA0位设置的TMIOA1边沿检测	○	○
	通过TMIORA1寄存器的IOB1位和IOB0位设置的TMIOB1边沿检测	○	○
	通过TMIORC1寄存器的IOC1位和IOC0位设置的TMIOC1边沿检测	—	○
	通过TMIORC1寄存器的IOD1位和IOD0位设置的TMIOD1边沿检测	—	○
输出比较功能、 PWM功能、 复位同步PWM模式、互补 PWM模式、PWM3模式	TM0寄存器和TMGRA0寄存器的比较匹配	○	○
	TM0寄存器和TMGRB0寄存器的比较匹配	○	○
	TM0寄存器和TMGRC0寄存器的比较匹配	—	○
	TM0寄存器和TMGRD0寄存器的比较匹配	—	○
	TM1寄存器和TMGRA1寄存器的比较匹配	○	○
	TM1寄存器和TMGRB1寄存器的比较匹配	○	○
	TM1寄存器和TMGRC1寄存器的比较匹配	—	○
	TM1寄存器和TMGRD1寄存器的比较匹配	—	○
互补PWM模式	TM1寄存器的下溢	○	—

10.5 定时器M的运行

10.5.1 输入捕捉功能

这是测量外部信号的宽度和周期的功能。以TMIO_{ji}引脚(i=0、1, j=A、B、C、D)的外部信号为触发，将TM_i寄存器(计数器)的内容传送到TMGR_{ji}寄存器(输入捕捉)。因为TMIO_{ji}引脚和TMGR_{ji}寄存器组合使用，所以能按引脚选择为输入捕捉功能、或者其他模式和功能。

输入捕捉功能的框图和运行例子分别如图10-46和图10-47所示，输入捕捉功能的规格如表10-12所示。

图10-46：输入捕捉功能的框图

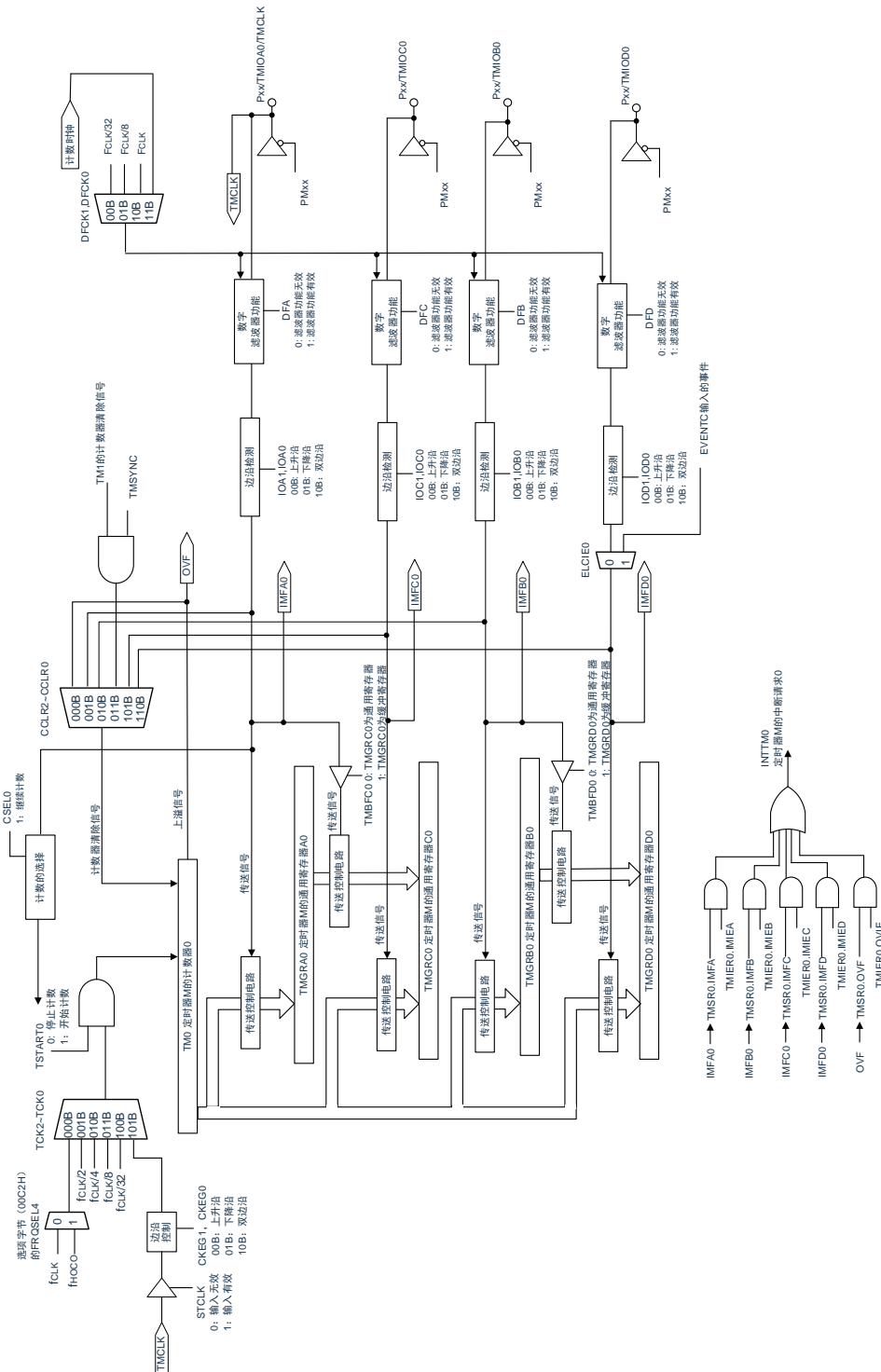


表10-12: 输入捕捉功能的规格

项目	规格
计数源	F_{HOCO} ^注 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$ TMCLK引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	当TMCRI寄存器的CCLR2~CCLR0位为“000B”(自由运行)时 $1/fk \times 65536$ fk: 计数源的频率
计数开始条件	给TMSTR寄存器的TSTARTi位写“1”(开始计数)。
计数停止条件	当TMSTR寄存器的CSELi位为“1”时, 给TSTARTi位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> 输入捕捉(TMIOj输入的上升沿) TMi的上溢
TMIOA0引脚功能	I/O端口、输入捕捉的输入或者TMCLK(外部时钟)输入
TMIOB0、TMIOC0、 TMIOD0、TMIOA1~ TMIOD1引脚功能	I/O端口或者输入捕捉的输入(按引脚进行选择)
INTP0引脚功能	不使用(输入专用端口或者INTP0中断输入)。
读定时器	如果读TMI寄存器, 就能读到计数值。
写定时器	<ul style="list-style-type: none"> 当TMMR寄存器的TMSYNC位为“0”(定时器M0和定时器M1独立运行)时能写TMI寄存器。 当TMMR寄存器的TMSYNC位为“1”(定时器M0和定时器M1同步运行)时如果写TMI寄存器, 数据就同时被写到TM0寄存器和TM1寄存器。
选择功能	<ul style="list-style-type: none"> 输入捕捉的输入引脚的选择 TMIOAi、TMIOBi、TMIOCi、TMIODi引脚中的1个或者多个引脚 输入捕捉的输入有效边沿的选择上升沿、下降沿或者双边沿 将TMi置“0000H”的时序上溢或者输入捕捉 缓冲器运行(参照“10.4.2缓冲器运行”) 同步运行(参照“10.4.3同步运行”) 数字滤波器 对TMIOj输入进行采样, 如果信号3次相同, 就视为电平已确定。 通过ELC输入的事件进行的输入捕捉运行

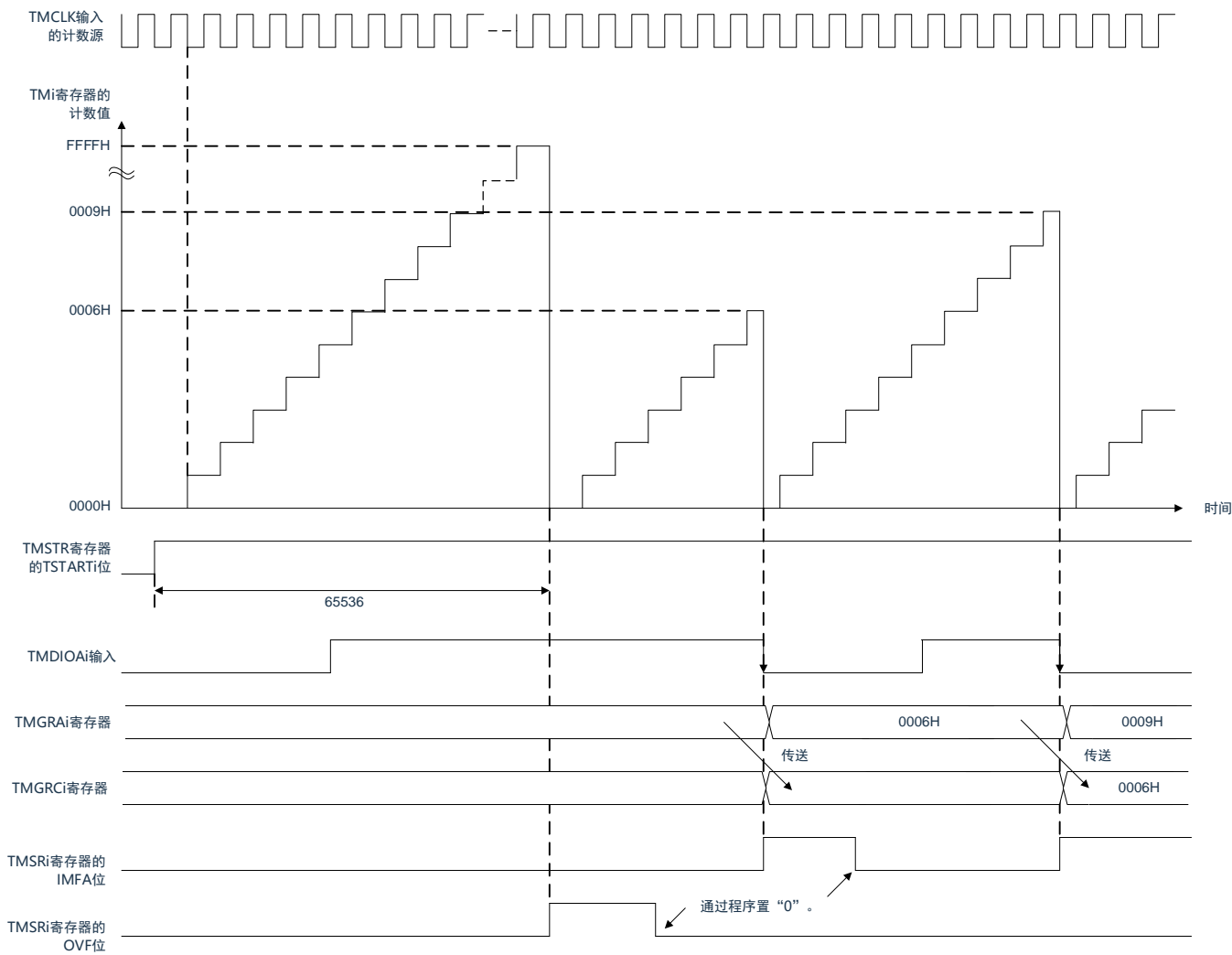
注: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

备注: i=0、1, j=A、B、C、D

(1) 运行例子

通过设置TMCRI寄存器(i=0、1)的CCLR0~CCLR2位，在发生输入捕捉或者比较匹配时对定时器Mi的计数值进行复位。图10-47是将CCLR2~CCLR0位置“001B”时的运行例子。如果设置为通过运行过程中的输入捕捉进行计数清除并且在定时器的计数值为“FFFFH”时进行输入捕捉，就根据计数源和输入捕捉的运行时序，TMSRI寄存器的IMFA~IMFD位和OVF位的中断标志可能同时变为“1”。

图10-47：输入捕捉功能的运行例子



备注：i=0, 1

上图的条件如下：
 TMCRI寄存器的CCLR2~CCLR0位为“001B”(在TMGRAi输入捕捉时，将TMi位置“0000H”)。
 TMCRI寄存器的TCK2~TCK0位为“101B”(计数源为TMCLK输入)。
 TMCRI寄存器的CKEG1位和CKEG0位为“01B”(在计数源的下降沿进行计数)。
 TMIOAI寄存器的IOA2~IOA0位为“101B”(在TMDIOAi输入的下降沿输入捕捉)。
 TMMR寄存器的TMBFCi位为“1”(TMGRCi寄存器为TMGRAi寄存器的缓冲寄存器)。

表10-13: 输出比较功能的规格

项目	规格
计数源	F_{HOCO} ^注 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$ TMCLK引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	<ul style="list-style-type: none"> 当TMCRI寄存器的CCLR2~CCLR0位为“000B”(自由运行)时 $1/fk \times 65536$ fk: 计数源的频率 当TMCRI寄存器的CCLR1~CCLR0位为“01B”或者“10B”(在TMGRji比较匹配时, 将TMI置“0000H”)时 $1/fk \times (n+1)$ n: TMGRji寄存器的设置值
波形输出时序	比较匹配(TMi寄存器和TMGRji寄存器的内容相同)
计数开始条件	给TMSTR寄存器的TSTARTi位写“1”(开始计数)。
计数停止条件	<ul style="list-style-type: none"> 当TMSTR寄存器的CSELi位为“1”时, 给TSTARTi位写“0”(停止计数)。输出比较的输出引脚保持停止计数前的输出电平。 在TMSTR寄存器的CSELi位为“0”并且发生TMGRAi的比较匹配时停止计数。输出比较的输出引脚保持比较匹配引起输出变化后的电平。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TMi寄存器和TMGRji寄存器的内容相同) TMi的上溢
TMIOA0引脚功能	I/O端口、输出比较的输出或者TMCLK(外部时钟)输入
TMIOB0、TMIOC0、 TMIOD0、TMIOA1~ TMIOD1引脚功能	I/O端口或者输出比较的输出(按引脚进行选择)
INTP0引脚功能	不使用(输入专用端口或者INTP0中断输入)。
读定时器	如果读TMi寄存器, 就能读到计数值。
写定时器	<ul style="list-style-type: none"> 当TMMR寄存器的TMSYNC位为“0”(定时器M0和定时器M1独立运行)时能写TMi寄存器。 当TMMR寄存器的TMSYNC位为“1”(定时器M0和定时器M1同步运行)时如果写TMi寄存器, 数据就同时被写到TM0寄存器和TM1寄存器。
选择功能	<ul style="list-style-type: none"> 输出比较的输出引脚的选择 TMIOAi、TMIOBi、TMIOCi、TMIODi引脚中的1个或者多个引脚 比较匹配时的输出电平的选择 “L”电平输出、“H”电平输出或者电平反相输出 初始输出电平的选择设置从开始计数到比较匹配为止的电平。 将TMI置“0000H”的时序 上溢或者TMGRAi寄存器的比较匹配 缓冲器运行(参照“10.4.2 缓冲器运行”) 同步运行(参照“10.4.3 同步运行”) TMGRCi和TMGRDi的输出引脚的变更 能将TMGRCi和TMGRDi分别用于TMIOAi引脚和TMIOBi引脚的输出控制。 能将定时器M用作内部定时器而不进行输出。

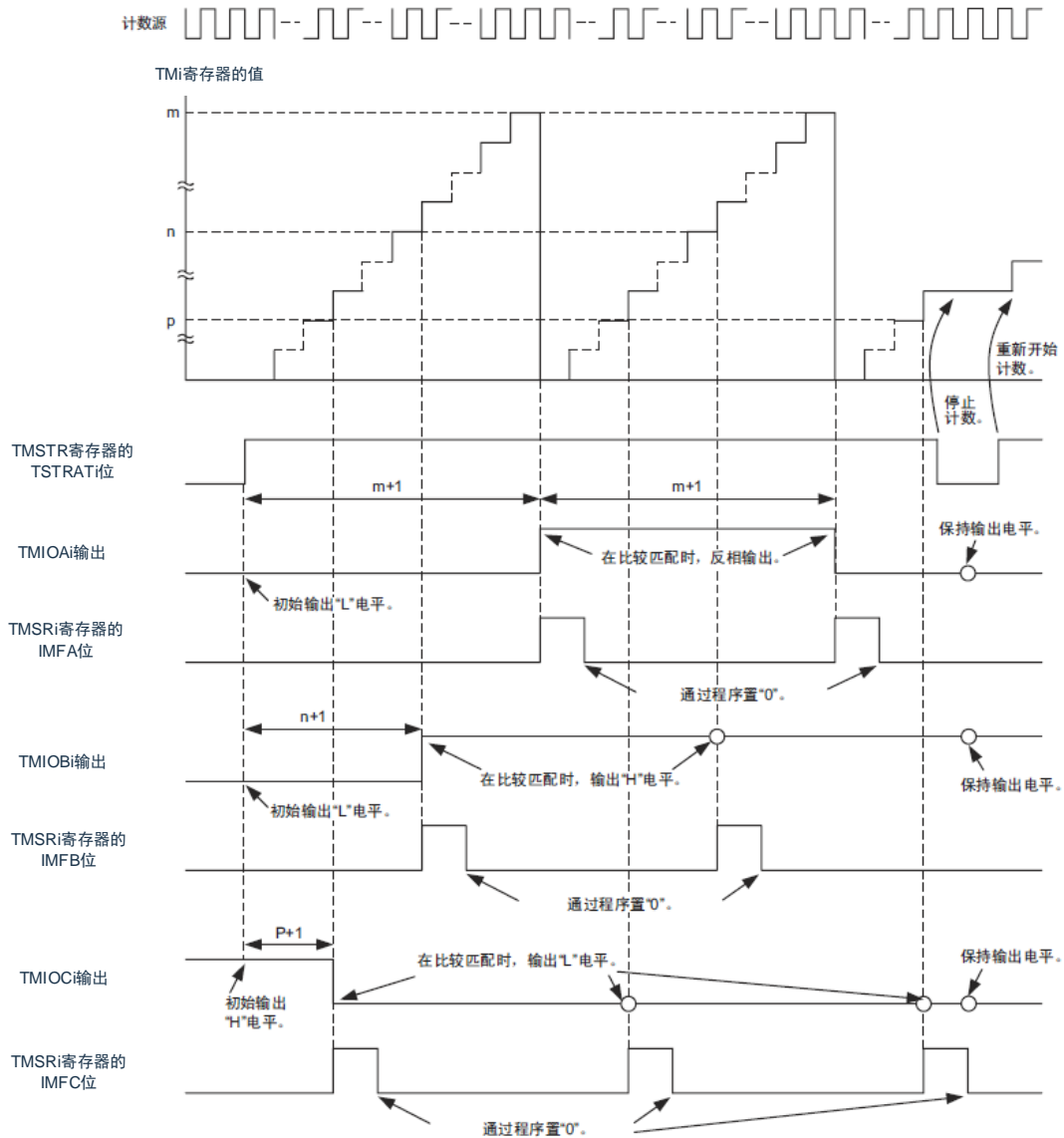
注: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

备注: $i=0, 1$, $j=A, B, C, D$

(1) 运行例子

通过设置TMCRI寄存器(i=0、1)的CCLR0~CCLR2位，在发生输入捕捉或者比较匹配时对定时器Mi的计数值进行复位。如果比较期待值为“FFFFH”，就和溢出相同从“FFFFH”变为“0000H”，并且上溢标志变为“1”。

图10-50：输出比较功能的运行例子



备注：i=0, 1

m: TMGRAi寄存器的设置值

n: TMGRBi寄存器的设置值

p: TMGRCi寄存器的设置值

上图的条件如下：

TMSTR寄存器的CSELi位为“1”（在比较匹配时，TMi不停止）。

TMMR寄存器的TMBFCi位和TMBFDi位为“0”（TMGRCi和TMGRDi不作为缓冲器运行）。

TMOER1寄存器的EAi位，EBi位和ECi位为“0”（允许TMIOAi, TMIOBi和TMIOCi的输出）。

TMCRi寄存器的CCLR2~CCLR0位为“001B”（在TMGRAi比较匹配时，将TMi位置“0000H”）。

TMOCR寄存器TOAi位和TOBi位为“0”（在比较匹配前，初始输出“L”电平），TOCi位为“1”（在比较匹配前，初始输出“H”电平）。

TMIORAi寄存器的IOA2~IOA0位为“011B”（在TMGRAi比较匹配时，TMIOAi反相输出）。

TMIORAi寄存器的IOB2~IOB0为“010B”（在TMGRBi比较匹配时，TMIOBi输出“H”电平）。

TMIORCi寄存器的IOC3~IOC0为“1001B”（在TMGRCi比较匹配时，TMIOCi输出“L”电平）。

TMIORCi寄存器的IOD3~IOD0为“1000B”（TMGRDi寄存器不控制TMIOBi引脚的输出，禁止比较匹配的引脚输出）。

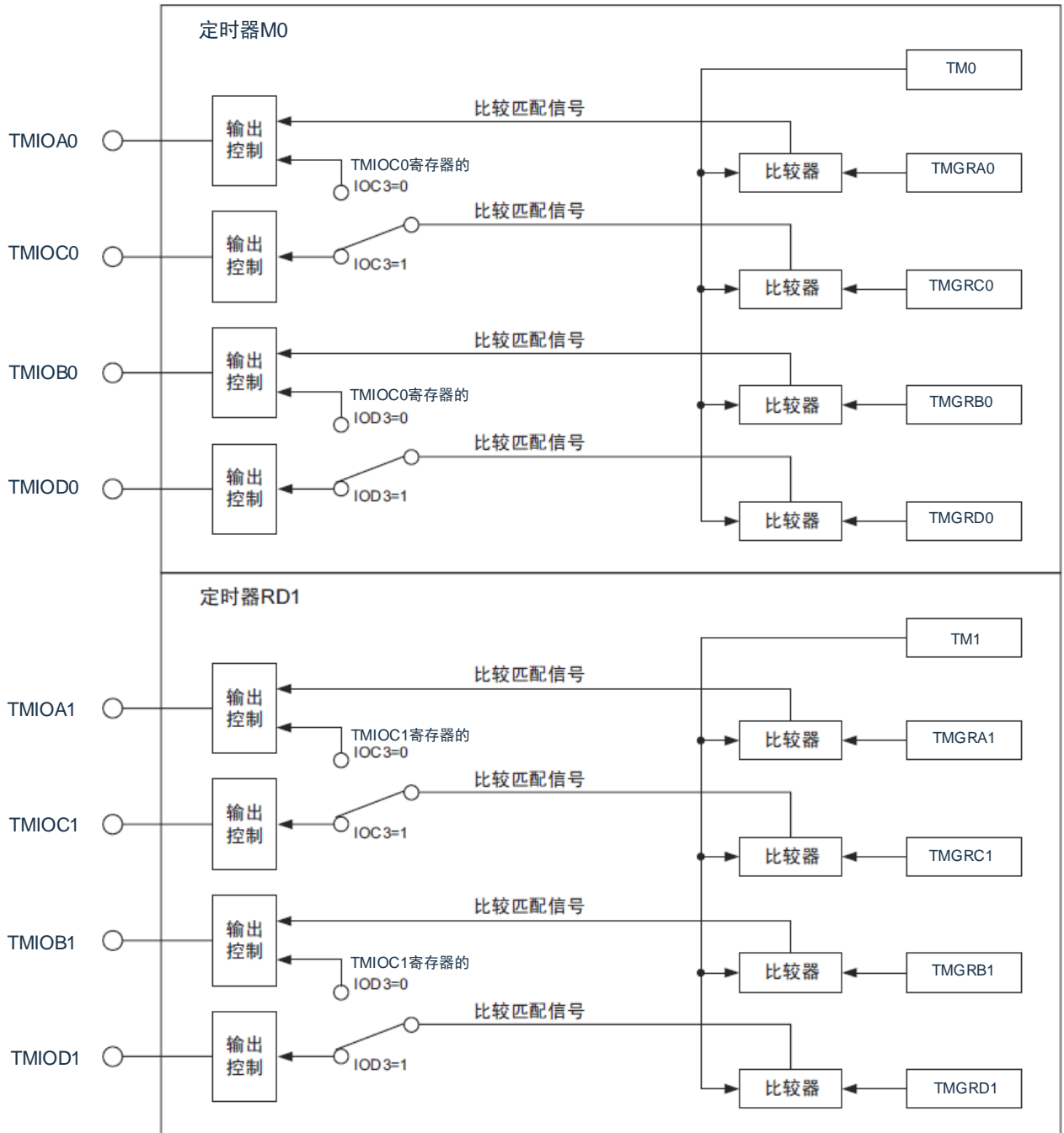
(2) TMGR*C*寄存器和TMGRD*i*寄存器的输出引脚的变更

能将TMGR*C*寄存器和TMGRD*i*寄存器分别用于TMIOA*i*引脚和TMIOB*i*引脚的输出控制。因此，能对各引脚进行如下的输出控制：

通过TMGRA*i*寄存器的值和TMGR*C*寄存器的值，控制TMIOA*i*输出。

通过TMGRB*i*寄存器的值和TMGRD*i*寄存器的值，控制TMIOB*i*输出。

图10-51：TMGR*C*和TMGRD*i*的输出引脚的变更

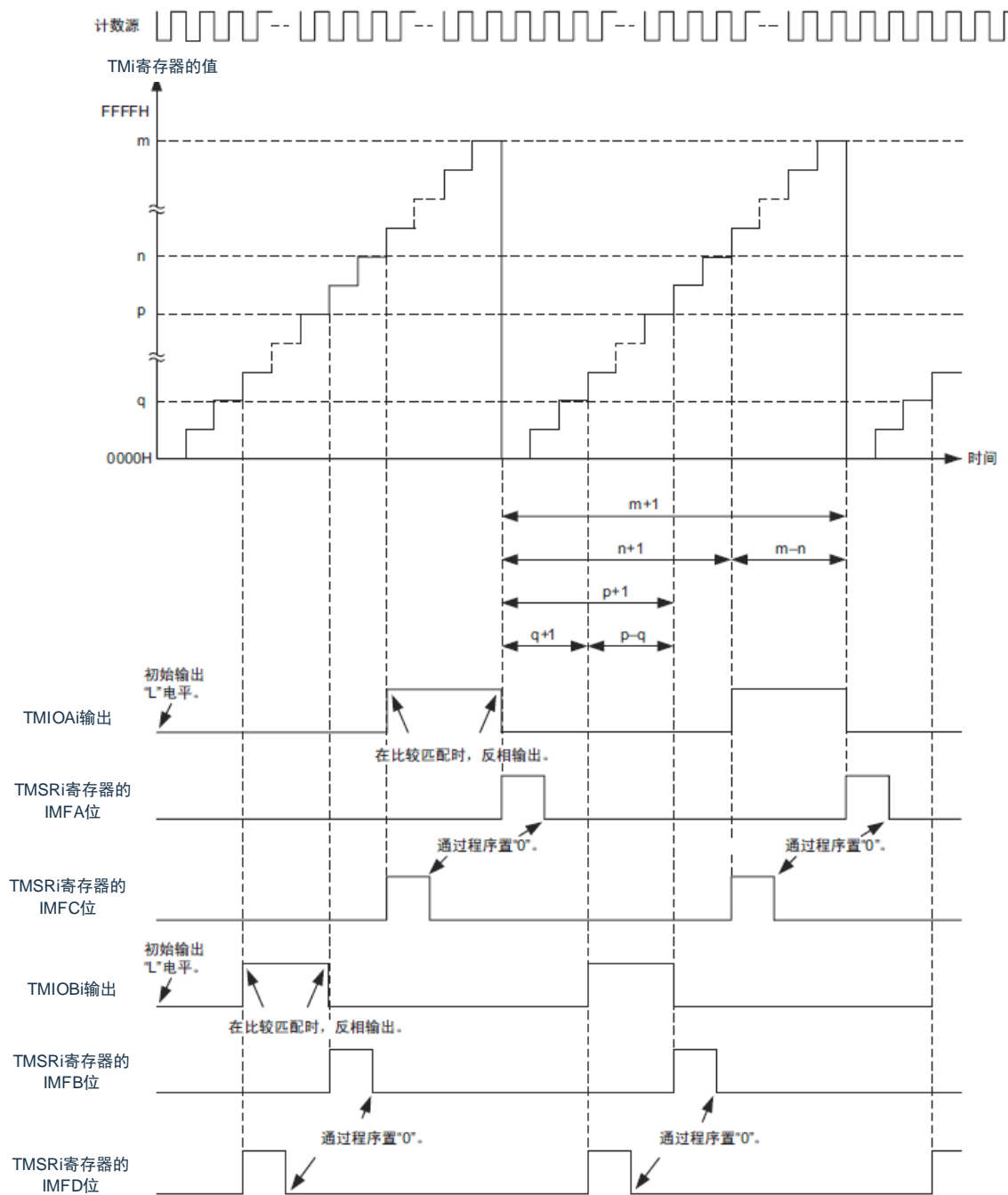


要更改TMGR*C*寄存器和TMGRD*i*寄存器的输出引脚时，必须进行以下的设置：

- 通过TMIORC*i*寄存器的IO*j*3位(*j*=C、D)选择“0”(TMGR*j*寄存器的输出引脚的变更)。
- 将TMMR寄存器的TMBF*j*位置“0”(通用寄存器)。
- 给TMGRA*i*寄存器和TMGR*C*寄存器设置不同的值，并且给TMGRB*i*寄存器和TMGRD*i*寄存器设置不同的值。

将TMGR*C*_i和TMGR*D*_i分别用于TMIOA*i*引脚和TMIOD*i*引脚的输出控制时的运行例子如图10-52所示。

图10-52：将TMGR*C*_i和TMGR*D*_i分别用于TMIOA*i*引脚和TMIOD*i*引脚的输出控制时的运行例子



备注：i=0, 1

- m: TMGRA*i*寄存器的设置值
- n: TMGRB*i*寄存器的设置值
- p: TMGR*C*_i寄存器的设置值
- q: TMGRD*i*寄存器的设置值

上图的条件如下：

TMSTR寄存器的CSELi位为“1”(在比较匹配时，TMi不停止)。

TMMR寄存器的TMBFCi位和TMBFDi位为“0”(TMGRCi和TMGRDi不作为缓冲器运行)。

TMOER1寄存器的EAI位，EBi位和ECi位为“0”(允许TMIOAi, TMIOBi和TMIOCi的输出)。

TMCRi寄存器的CCLR2~CCLR0位为“001B”(在TMGRAi比较匹配时，将TMi位置“0000H”)。

TMOCR寄存器TOAi位和TOBi位为“0”(在比较匹配前，初始输出“L”电平)。

TMIORAi寄存器的IOA2~IOA0位为“011B”(在TMGRAi比较匹配时，TMIOAi反相输出)。

TMIORAi寄存器的IOB2~IOB0为“011B”(在TMGRBi比较匹配时，TMIOBi反相输出)。

TMIORCi寄存器的IOC3~IOC0为“0011B”(在TMGRCi比较匹配时，TMIOAi反相输出)。

TMIORDi寄存器的IOD3~IOD0为“1000B”(在TMGRCi比较匹配时，TMIOBi反相输出)。

10.5.3 PWM功能

这是输出PWM波形的功能。通过定时器Mi(i=0、1)最多能输出3个同周期的PWM波形。通过使定时器M0与定时器M1同步，最多能输出6个同周期的PWM波形。

因为TMIOj_i引脚(j=B、C、D)和TMGRj_i寄存器组合使用，所以能按引脚选择为PWM功能、或者其他模式和功能(但是，无论将哪个引脚用作PWM功能，都要使用TMGRA_i寄存器，因此TMGRA_i寄存器不能用于其他模式)。

PWM功能的框图和规格分别如图10-53和表10-14所示，PWM功能的运行例子如图10-54和图10-55所示。

图10-53: PWM功能的框图

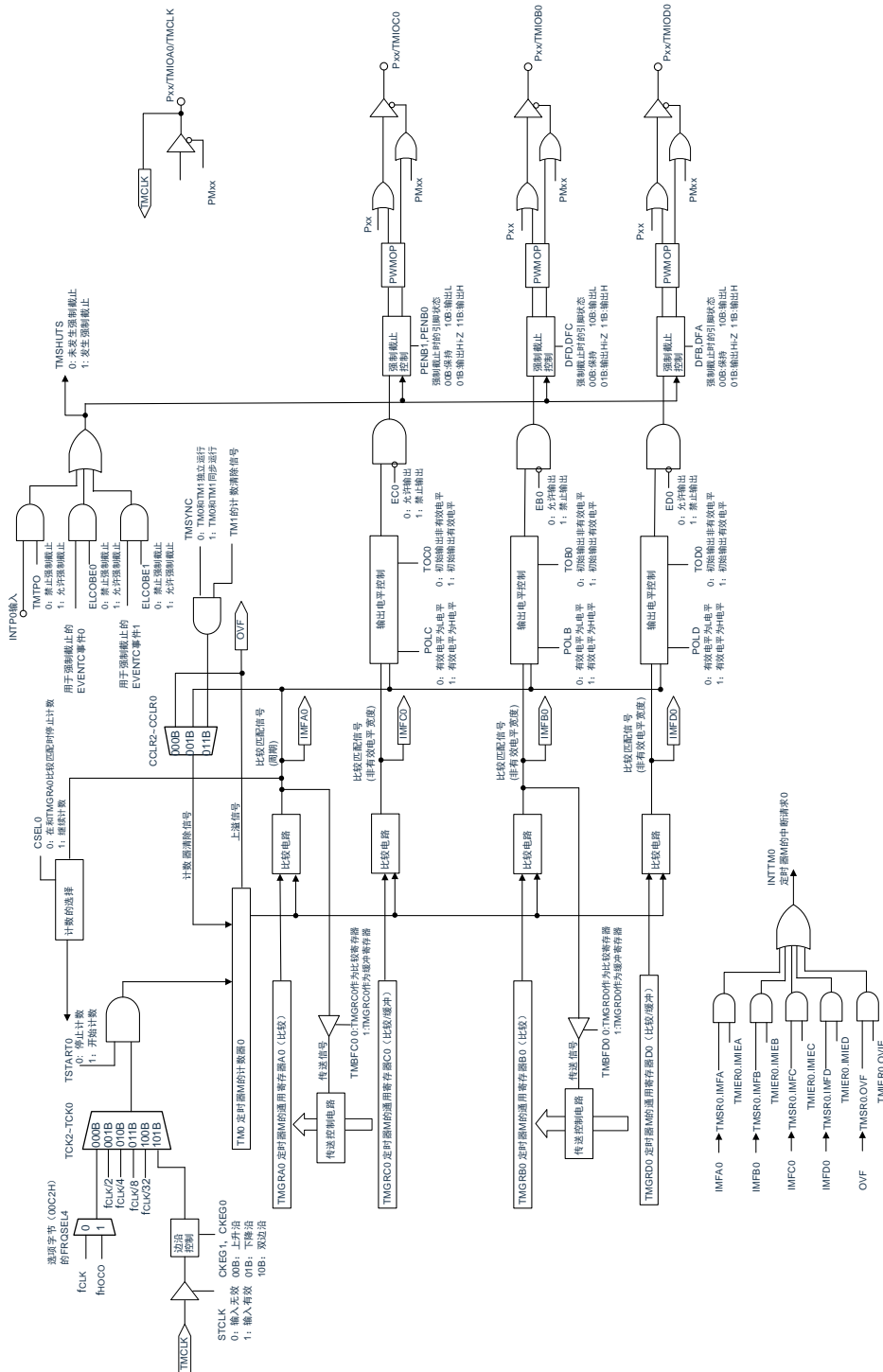
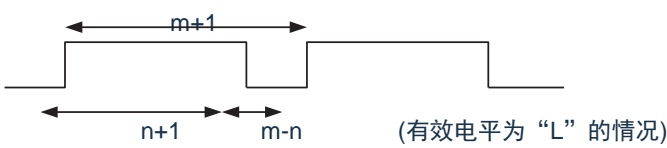


表10-14: PWM功能的规格

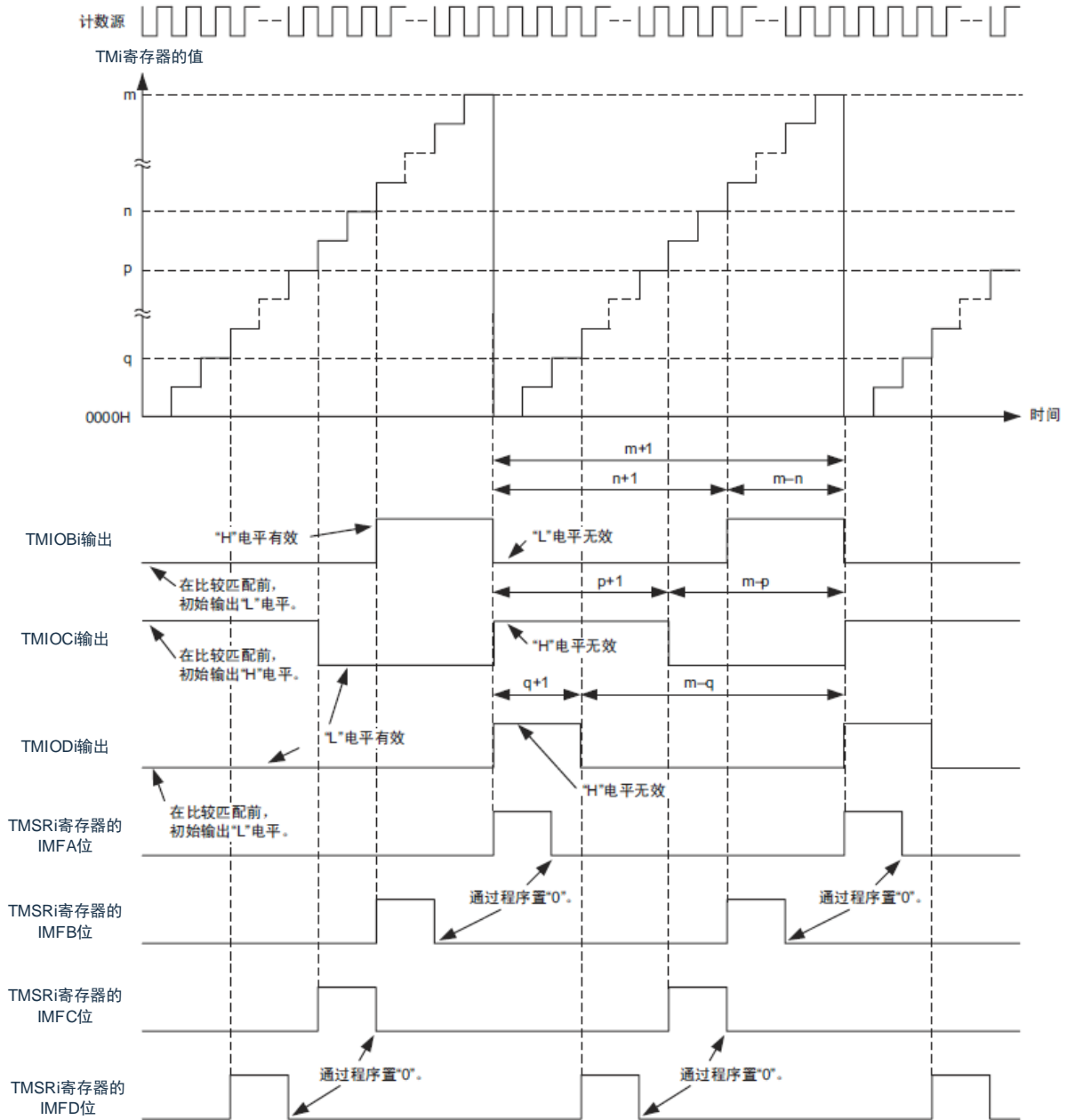
项目	规格
计数源	F_{HOCO} ^注 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$ TMCLK引脚的外部输入信号(能通过程序选择有效边沿)
计数	递增计数
PWM波形	PWM周期: $1/fk \times (m+1)$ 有效电平宽度: $1/fk \times (m-n)$ 无效电平宽度: $1/fk \times (n+1)$ fk: 计数源的频率 m: TMGRAi寄存器的设置值 n: TMGRji寄存器的设置值  (有效电平为“L”的情况)
计数开始条件	给TMSTR寄存器的TSTARTi位写“1”(开始计数)。
计数停止条件	<ul style="list-style-type: none"> 当TMSTR寄存器的CSELi位为“1”时, 给TSTARTi位写“0”(停止计数)。PWM输出引脚保持停止计数前的输出电平。 在TMSTR寄存器的CSELi位为“0”并且发生TMGRAi的比较匹配时停止计数。PWM输出引脚保持比较匹配引起输出变化后的电平。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TM_i寄存器和TMGR_{hi}寄存器的内容相同) TM_i的上溢
TMIOA0引脚功能	I/O端口或者TMCLK(外部时钟)输入
TMIOA1引脚功能	I/O端口
TMIOB0、TMIOC0、TMIOD0 TMIOB1、TMIOC1、TMIOD1 引脚功能	I/O端口或者PWM输出(按引脚进行选择)
INTP0引脚功能	脉冲输出强制截止信号的输入(输入专用端口或者INTP0中断输入)
读定时器	如果读TM _i 寄存器, 就能读到计数值。
写定时器	能写TM _i 寄存器。
选择功能	<ul style="list-style-type: none"> 通过定时器M_i进行的1~3个PWM输出引脚的选择 TMIOB_i、TMIOC_i、TMIOD_i引脚中的1个或者多个引脚 各引脚有效电平的选择 各引脚初始输出电平的选择 同步运行(参照“10.4.3同步运行”) 缓冲器运行(参照“10.4.2缓冲器运行”) 脉冲输出强制截止信号的输入(参照“10.4.4脉冲输出的强制截止”)

注: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

备注: i=0、1, j=B、C、D, h=A、B、C、D

(1) 运行例子

图10-54: PWM 功能的运行例子



备注: $i=0, 1$

m: TMGRAi寄存器的设置值

n: TMGRBi寄存器的设置值

p: TMGRCi寄存器的设置值

q: TMGRDi寄存器的设置值

上图的条件如下:

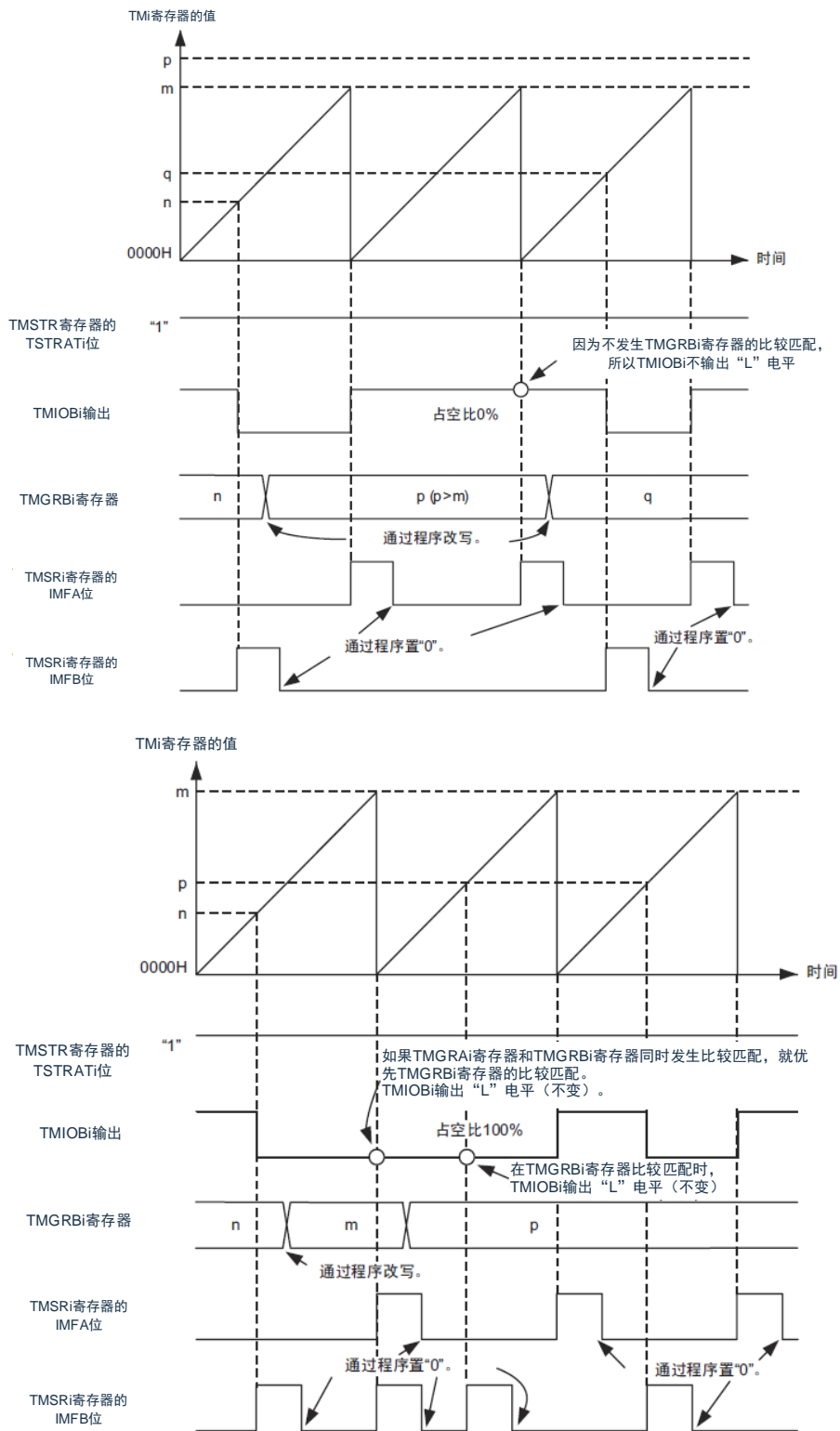
TMMR寄存器的TMBFCi位和TMBFDi位为“0”(TMGRCi和TMGRDi不作为缓冲器运行)。

TMOER1寄存器的EBi位和ECi位为“0”(允许TMIODi和TMIOCI的输出)。

TMOCR寄存器TOBi位和TOCi位为“0”(无效电平), TODi位为“1”(有效电平)。

TMPOCRi寄存器的POLB位为“1”(“H”电平有效), POLC位和POLD位为“0”(“L”电平有效)

图10-55: PWM功能的运行例子(占空比为0%和100%)



备注: i=0, 1

m: TMGRAi寄存器的设置值

上图的条件如下:

TMOER1寄存器的EBi位为"0"(允许TMIObi输出)。

TMPOCRi寄存器的POLB位为"0"("L"电平有效)

10.5.4 复位同步PWM模式

输出3个正相和3个反相(共6个)的同周期的PWM波形(三相、锯齿波调制、无死区时间)。

复位同步PWM模式的框图和运行例子分别如图10-56和图10-57所示，复位同步PWM模式的规格如表10-15所示。

有关占空比为0%和100%的PWM运行例子，请参照“图10-56：复位同步PWM模式的框图”

图10-56：复位同步PWM模式的框图

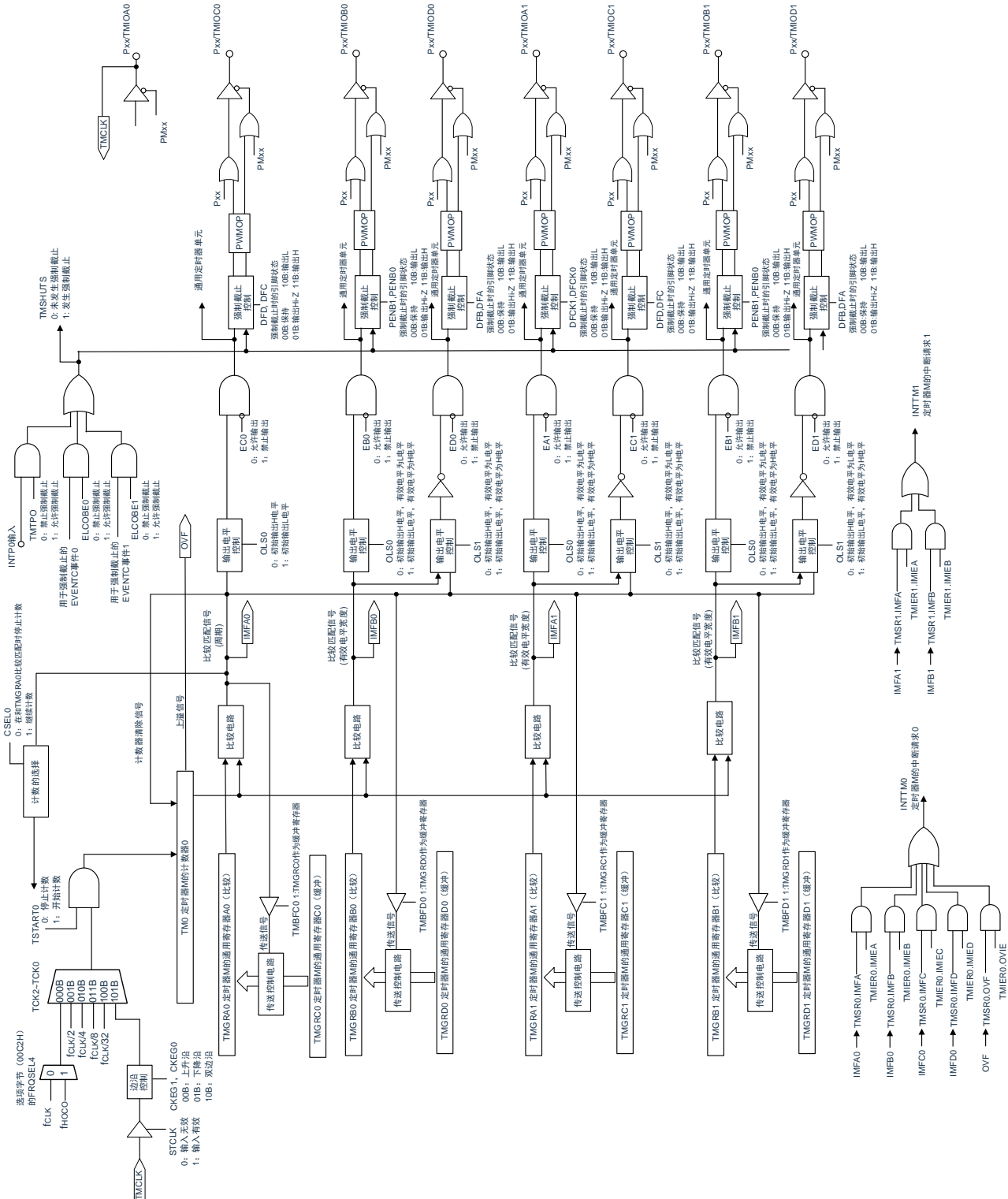
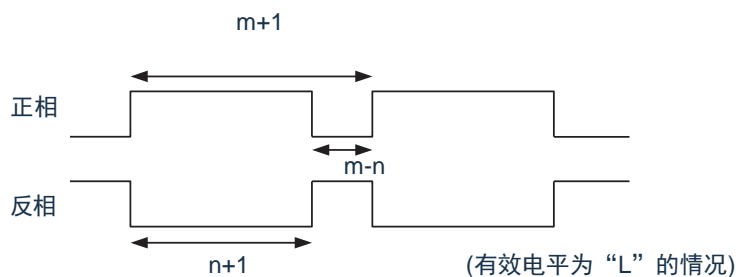


表10-15: 复位同步 PWM 模式的规格

项目	规格
计数源	F_{HOCO} ^注 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$ TMCLK引脚的外部输入信号(能通过程序选择有效边沿)
计数	TM0为递增计数(不使用TM1)。
PWM 波形	PWM 周期: $1/f_k \times (m+1)$ 正相有效电平宽度: $1/f_k \times (m-n)$ 反相有效电平宽度: $1/f_k \times (n+1)$ f_k : 计数源的频率 m : TMGRA0 寄存器的设置值 n : TMGRB0 寄存器的设置值 (PWM 输出 1) TMGRA1 寄存器的设置值 (PWM 输出 2) TMGRB1 寄存器的设置值 (PWM 输出 3)
计数开始条件	给TMSTR寄存器的TSTART0位写“1”(开始计数)。
计数停止条件	<ul style="list-style-type: none"> 当TMSTR寄存器的CSEL0位为“1”时, 给TSTART0位写“0”(停止计数)。 PWM输出引脚输出TMFCR寄存器的OLS0位和OLS1位选择的初始输出电平。 在TMSTR寄存器的CSEL0位为“0”并且发生TMGRA0的比较匹配时停止计数。 PWM输出引脚输出TMFCR寄存器的OLS0位和OLS1位选择的初始输出电平。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TM0寄存器和TMGRj0、TMGRA1、TMGRB1寄存器的内容相同) TM0的上溢
TMIOA0引脚功能	I/O端口或者TMCLK(外部时钟)输入
TMIOB0引脚功能	PWM输出1的正相输出
TMIOD0引脚功能	PWM输出1的反相输出
TMIOA1引脚功能	PWM输出2的正相输出
TMIOC1引脚功能	PWM输出2的反相输出
TMIOB1引脚功能	PWM输出3的正相输出
TMIOD1引脚功能	PWM输出3的反相输出
TMIOC0引脚功能	在每个PWM周期进行反相输出。
INTP0引脚功能	脉冲输出强制截止信号的输入(输入专用端口或者INTP0中断输入)
读定时器	如果读TM0寄存器, 就能读到计数值。
写定时器	能写TM0寄存器。
选择功能	<ul style="list-style-type: none"> 正反相有效电平和初始输出电平的选择 缓冲器运行(参照“10.4.2缓冲器运行”) 脉冲输出强制截止信号的输入(参照“10.4.4脉冲输出的强制截止”)

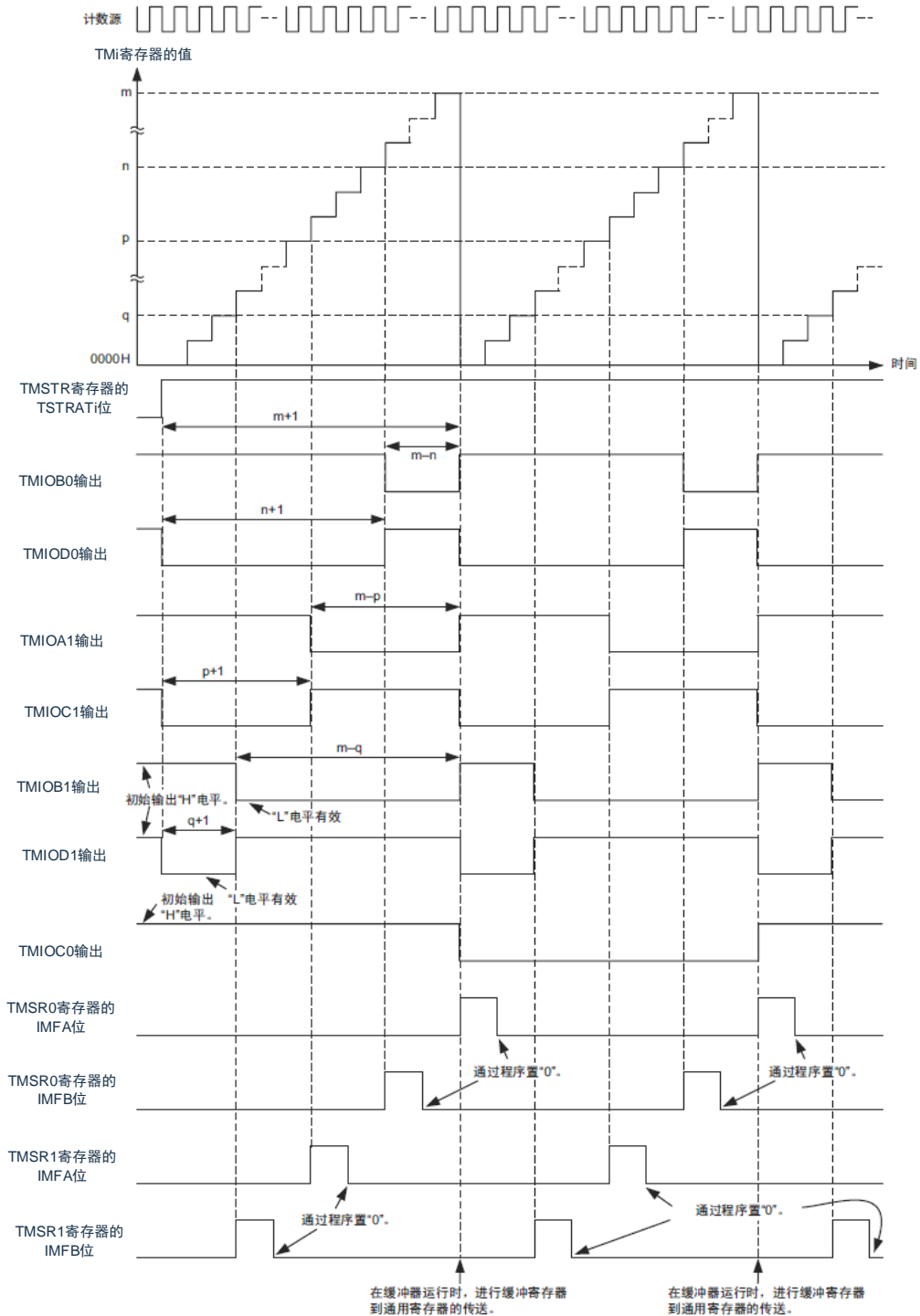


注: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

备注: j=A、B、C、D

(1) 运行例子

图10-57: 复位同步PWM模式的运行例子



备注: $i=0, 1$

- m: TMGRA0寄存器的设置值
- n: TMGRB0寄存器的设置值
- p: TMGRA1寄存器的设置值
- q: TMGRB1寄存器的设置值

上图的条件如下:

TMFCR寄存器的OLS1位和OLS0位为“0”(初始输出“H”电平,“L”电平有效)

10.5.5 互补PWM模式

输出3个正相和3个反相(共6个)的同周期的PWM波形(三相、三角波调制、有死区时间)。

互补PWM模式的框图如图10-58所示，互补PWM模式的规格如表10-16所示，互补PWM模式的输出模型和运行例子分别如图10-59和图10-60所示。

图10-58: 互补PWM模式的框图

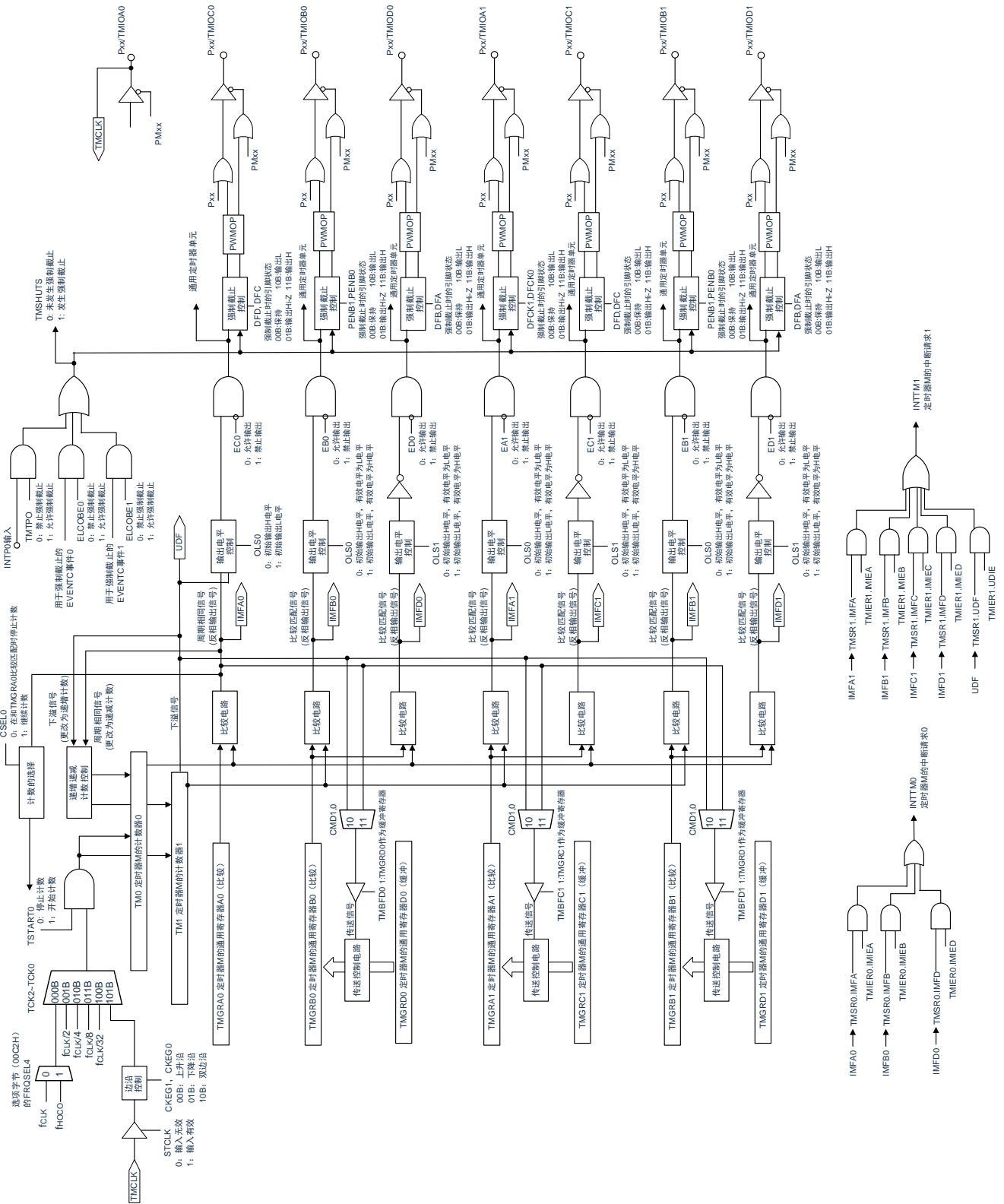
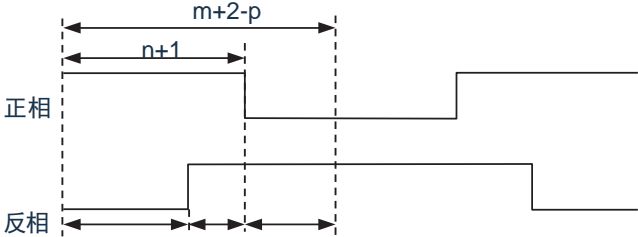


表10-16: 互补PWM模式的规格

项目	规格
计数源	F_{HOCO} ^{注1} 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$ TMCLK引脚的外部输入信号(能通过程序选择有效边沿) 必须给TMCR0寄存器的TCK0~TCK2位和TMCR1寄存器的TCK0~TCK2位设置相同的值(相同的计数源)。
计数	递增计数或者递减计数 如果在递增计数的过程中TM0寄存器和TMGRA0寄存器比较匹配, TM0和TM1就都变为递减计数; 如果在递减计数的过程中TM1寄存器从“0000H”变为“FFFFH”, TM0和TM1就都变为递增计数。
PWM波形	PWM周期: $1/fk \times (m+2-p) \times 2^{n+2}$ 死区时间: p 正相有效电平宽度: $1/fk \times (m-n-p+1) \times 2$ 反相有效电平宽度: $1/fk \times (n+1-p) \times 2$ fk: 计数源的频率 m: TMGRA0寄存器的设置值 n: TMGRB0寄存器的设置值(PWM输出1) TMGRA1寄存器的设置值(PWM输出2) TMGRB1寄存器的设置值(PWM输出3) p: TM0寄存器的设置值 
计数开始条件	给TMSTR寄存器的TSTART0位和TSTART1位写“1”(开始计数)。
计数停止条件	当TMSTR寄存器的CSEL0位为“1”时, 给TSTART0位和TSTART1位写“0”(停止计数) (PWM输出引脚输出TMFCR寄存器的OLS0位和OLS1位选择的初始输出电平)。
中断请求的产生时序	<ul style="list-style-type: none"> •比较匹配(TMi寄存器和TMGRji寄存器的内容相同) •TM1的下溢
TMIOA0引脚功能	I/O端口或者TMCLK(外部时钟)输入
TMIOB0引脚功能	PWM输出1的正相输出引脚
TMIOD0引脚功能	PWM输出1的反相输出引脚
TMIOA1引脚功能	PWM输出2的正相输出引脚
TMIOC1引脚功能	PWM输出2的反相输出引脚
TMIOB1引脚功能	PWM输出3的正相输出引脚
TMIOD1引脚功能	PWM输出3的反相输出引脚
TMIOC0引脚功能	在每1/2个PWM周期进行反相输出。
INTP0引脚功能	脉冲输出强制截止信号的输入(输入专用端口或者INTP0中断输入)
读定时器	如果读TMi寄存器, 就能读到计数值。
写定时器	能写TMi寄存器。
选择功能	<ul style="list-style-type: none"> •脉冲输出强制截止信号的输入(参照“10.4.4脉冲输出的强制截止”) •正反相有效电平和初始输出电平的选择 •缓冲寄存器的传送时序的选择

注1: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

注2: 在开始计数后, PWM周期是固定的。

备注: i=0、1, j=A、B、C、D

(1) 运行例子

图10-59: 互补PWM模式的输出模型

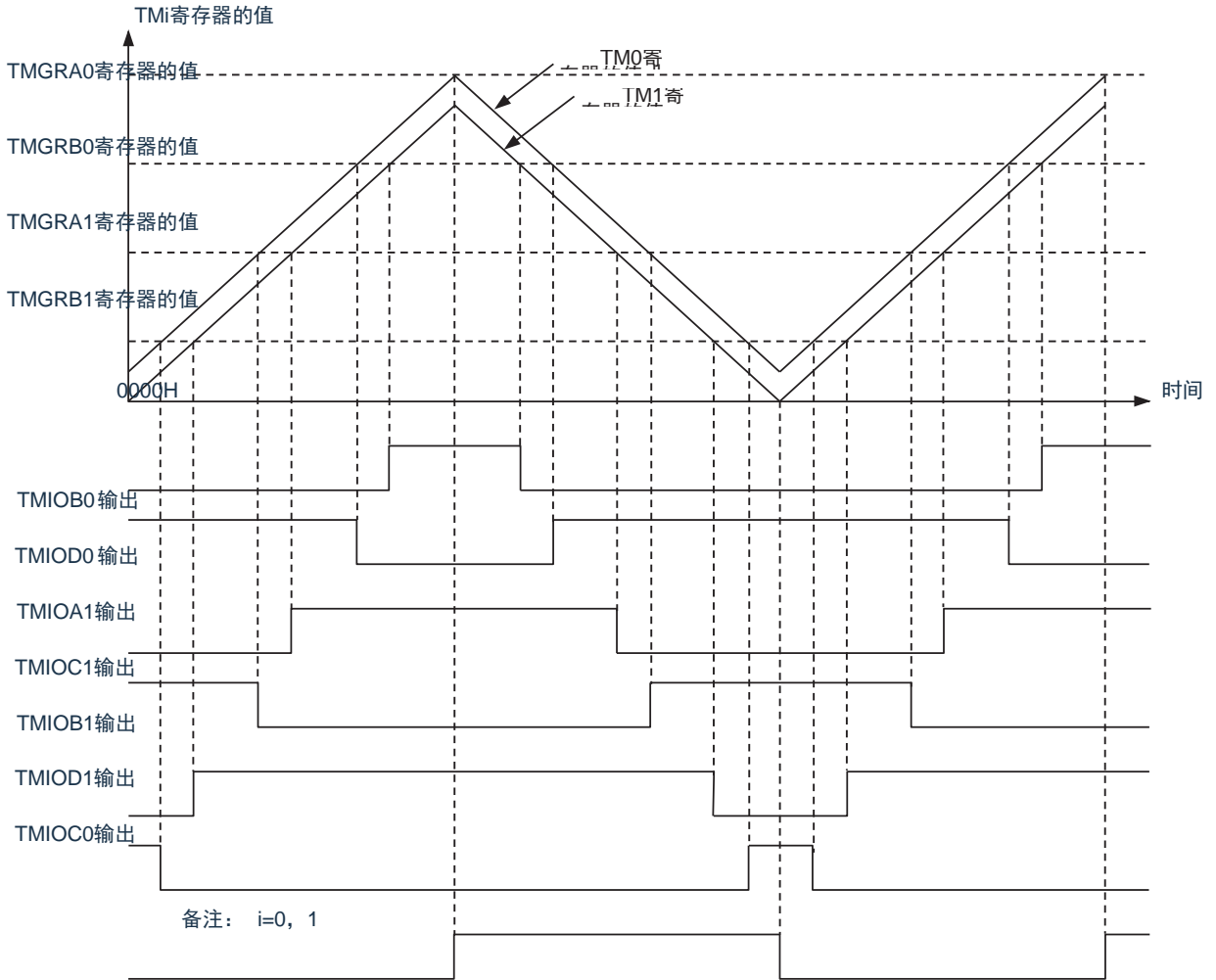
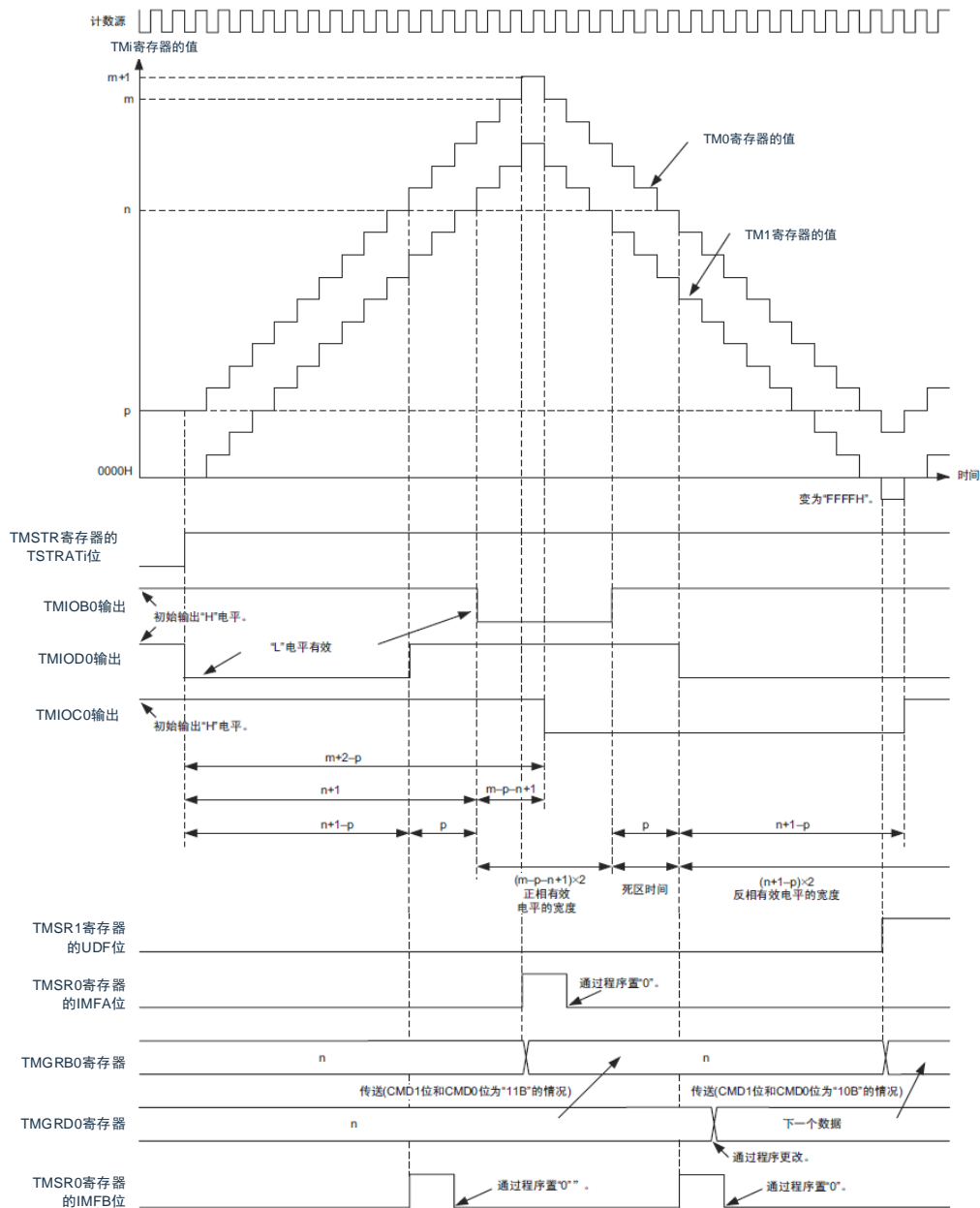


图10-60: 互补PWM模式的运行例子



备注: CMD0,CMD1:TMFCR寄存器的位
 $i=0, 1$
 m : TMGRA0寄存器的设置值
 n : TMGRB0寄存器的设置值
 p : TM0寄存器的设置值

上图的条件如下:

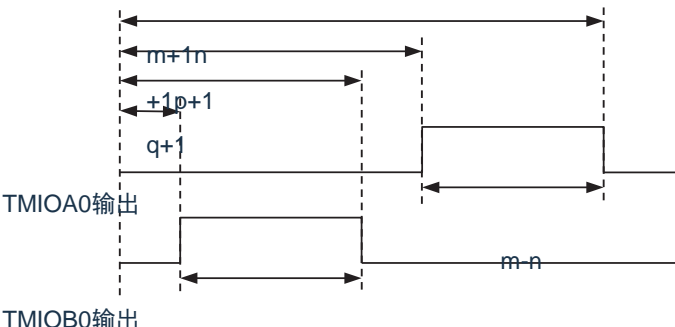
TMFCR寄存器的OLS1位和OLS0位为“0”(初始输出“H”电平,“L”电平有效)

(2) 缓冲寄存器的数据传送时序

TMGRD0、TMGRC1、TMGRD1寄存器到TMGRB0、TMGRA1、TMGRB1寄存器的数据传送在TMFCR寄存器的CMD1位和CMD0位为“10B”并且TM1发生下溢时进行数据传送。

在CMD1位和CMD0位为“11B”并且TM0寄存器和TMGRA0寄存器比较匹配时进行数据传送。

表10-17: PWM3模式的规格

项目	规格
计数源	$F_{HOCO}^{\#}$ 、 F_{CLK} 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$
计数	TM0为递增计数(不使用TM1)。
PWM波形	<p>PWM周期: $1/f_k \times (m+1)$ TMIOA0输出的有效电平宽度: $1/f_k \times (m-n)$ TMIOB0输出的有效电平宽度: $1/f_k \times (p-q)$ f_k: 计数源的频率 m: TMGRA0寄存器的设置值 n: TMGRA1寄存器的设置值 p: TMGRB0寄存器的设置值 q: TMGRB1寄存器的设置值</p> 
计数开始条件	给TMSTR寄存器的TSTART0位写“1”(开始计数)。
计数停止条件	<ul style="list-style-type: none"> 当TMSTR寄存器的CSEL0位为“1”时, 给TSTART0位写“0”(停止计数)。PWM输出引脚保持停止计数前的输出电平。 在TMSTR寄存器的CSEL0位为“0”并且发生TMGRA0的比较匹配时停止计数。PWM输出引脚保持比较匹配引起输出变化后的电平。
中断请求的产生时序	<ul style="list-style-type: none"> 比较匹配(TM_i寄存器和TMGR_{ji}寄存器的内容相同) TM0的上溢
TMIOA0引脚和TMIOB0引脚的功能	PWM输出
TMIOC0、TMIOD0、TMIOA1~TMIOD1引脚功能	I/O端口
INTP0引脚功能	脉冲输出强制截止信号的输入(输入专用端口或者INTP0中断输入)
读定时器	如果读TM0寄存器, 就能读到计数值。
写定时器	能写TM0寄存器。
选择功能	<ul style="list-style-type: none"> 脉冲输出强制截止信号的输入(参照“10.4.4脉冲输出的强制截止”) 各引脚有效电平的选择 缓冲器运行(参照“10.4.2缓冲器运行”)

注: 只有在用户选项字节(000C2H)的FRQSEL4位为“1”时才能选择 F_{HOCO} 。要选择 F_{HOCO} 作为定时器M的计数源时, 必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟, 就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

备注: $i=0、1$, $j=A、B、C、D$

上图的条件如下：

TMOCR寄存器TOA0位和TOB0位为“0”（初始输出“L”电平,在TMGRj1寄存器比较匹配时输出“H”电平，在TMGRj0寄存器比较匹配时输出“L”电平）。

TMMR寄存器的TMBFC0位为“1”（TMGRC0寄存器为TMGRA0寄存器的缓冲寄存器）

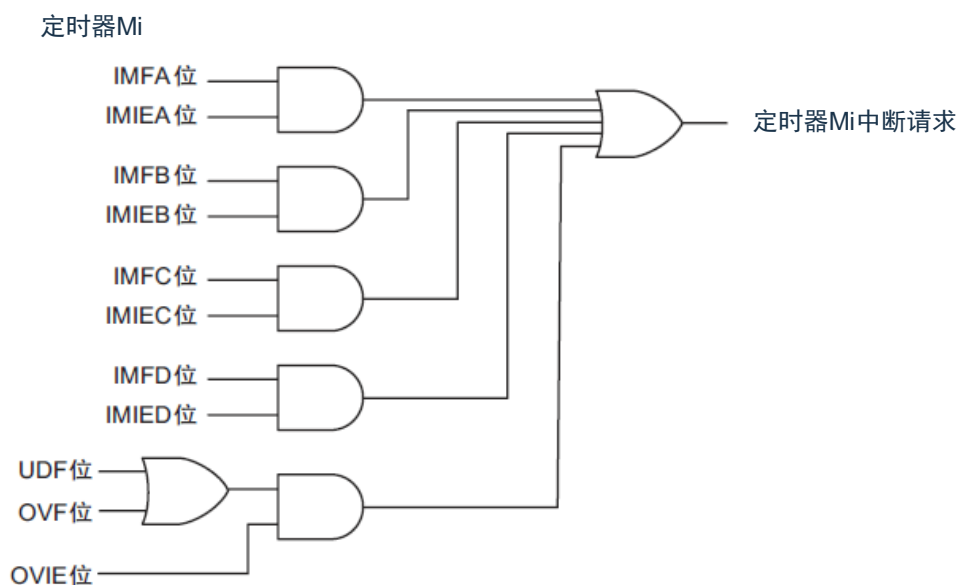
10.6 定时器M中断

定时器M从定时器M0和定时器M1的各6个中断源中产生定时器Mi(i=0、1)中断请求。定时器M中断的相关寄存器如表10-18所示，定时器M中断的框图如图10-63所示。

表10-18: 定时器M中断的相关寄存器

	定时器M的状态寄存器	定时器M的中断允许寄存器	中断请求标志(寄存器)	中断屏蔽标志(寄存器)
定时器M0	TMSR0	TMIER0	TMIF0(IF2H)	TMMK0(MK2H)
定时器M1	TMSR1	TMIER1	TMIF1(IF2H)	TMMK1(MK2H)

图10-63: 定时器M中断的框图

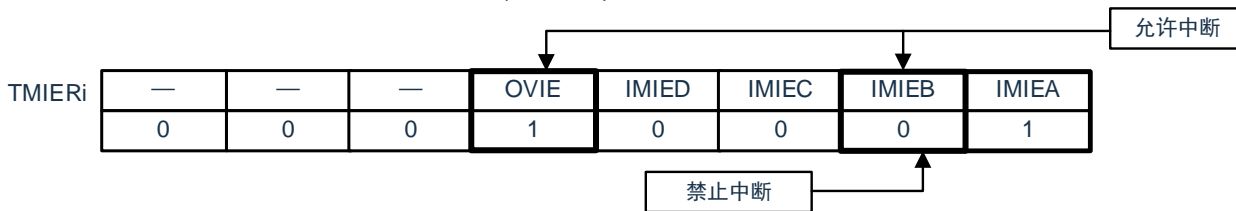


i = 0、1
 IMFA、IMFB、IMFC、IMFD、OVF、UDF：TMSRi寄存器的位
 IMIEA、IMIEB、IMIEC、IMIED、OVIE：TMIERi寄存器的位

因为定时器M从多个中断请求源中产生1个中断请求(定时器M中断)，所以除了定时器RG中断以外，和其他可屏蔽中断有以下不同：

- 当TMSRi寄存器的位为“1”并且对应的TMIERi寄存器的位为“1”(允许中断)时，IF2H寄存器的TMIFi位就变为“1”(有中断请求)。
 - 当TMIERi寄存器的多个位为“1”时，必须通过TMSRi寄存器判断是哪个请求源产生的中断。
 - 因为即使接受中断，TMSRi寄存器的各位也不会自动变为“0”，所以必须在中断程序中将这些位置“0”。
 - 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时，如果该中断被定时器M中断允许寄存器i(TMIE Ri)设置为禁止中断，就必须用以下(a)~(c)的任意一种方法置“0”。
 - (a) 必须在将定时器M中断允许寄存器i(TMIE Ri)置“00H”(禁止全部中断)后给对象状态标志写“0”。
 - (b) 当定时器M中断允许寄存器i(TMIE Ri)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时，必须给对象状态标志写“0”。
- (例)在IMIEA和OVIE为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



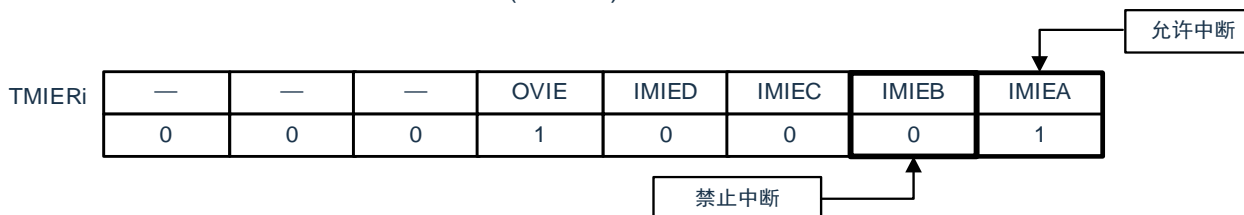
- 定时器M状态寄存器i(TMSRi)的状态



(c) 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(例)在IMIEA为允许中断而IMIEB为禁止中断的状态下清除IMFB的情况

- 定时器M中断允许寄存器i(TMIERi)的状态



- 定时器M状态寄存器i(TMSRi)的状态



10.7 使用定时器M时的注意事项

10.7.1 SFR的读写存取

要设置定时器M时，必须先将PER1寄存器的TMMEN位置“1”。当TMMEN位为“0”时，忽视定时器M的控制寄存器的写操作，而且读取值都为初始值(端口寄存器和端口模式寄存器除外)。

在计数过程中禁止改写以下寄存器：

TMELC寄存器、TMMR寄存器、TMPMR寄存器、TMFCR寄存器、TMOER1寄存器、TMOER2寄存器的TMPTO位、TMDFi寄存器、TMCRI寄存器、TMIORAi寄存器、TMIORCi寄存器、TMPOCRi寄存器

(1) TMSTR寄存器

- 能通过8位存储器操作指令设置TMSTR寄存器。
- 当TMSTR寄存器的CSELi位($i=0, 1$)为“0”(在Tmi寄存器和TMGRAi寄存器比较匹配时停止计数)时，即使给TSTARTi位写“0”(停止计数)，也不停止计数并且TSTARTi位也不变。只有在和TMGRAi寄存器比较匹配时，TSTARTi位才变为“0”(停止计数)。

在改写TMSTR寄存器时，如果要在CSELi位为“0”的情况下不影响计数而将CSELi位改为“1”，就必须给TSTARTi位写“0”。

如果在计数器停止计数的过程中给TSTARTi位写“1”，就可能开始计数。要通过程序停止计数时，必须在将CSELi位置“1”后给TSTARTi位写“0”。即使同时(使用1条指令)给CSELi位和TSTARTi位写“1”和“0”，也不能停止计数。

- 将TMIOji引脚($j=A, B, C, D$)用于定时器M输出时的计数停止期间的输出电平如表10-19所示。

表10-19: 停止计数时的TMIOji引脚($j=A, B, C, D$)的输出电平

停止计数的方法	停止计数时的TMIOji引脚的输出电平
在CSELi位为“1”时，通过给TSTARTi位写“0”来停止计数。	保持停止计数前的输出电平(在定时器M的互补PWM模式或者复位同步PWM模式中，输出TMFCR寄存器的OLS0位和OLS1位选择的初始输出电平)。
当CSELi位为“0”时，在Tmi寄存器和TMGRAi寄存器比较匹配时停止计数。	保持比较匹配引起输出变化后的电平(在定时器M的互补PWM模式或者复位同步PWM模式中，输出TMFCR寄存器的OLS0位和OLS1位选择的初始输出电平)。

备注： $i=0, 1$ ， $j=A, B, C, D$

(2) TMDFi寄存器($i=0, 1$)

必须在设置TMDFi寄存器的DFCK0位和DFCK1位后开始计数。

(3) Tmi寄存器($i=0, 1$)

如果Tmi寄存器的值变为“0000H”的时序和写Tmi寄存器的时序重叠，就优先写寄存器。

10.7.2 模式的切换

- 要在运行过程中切换模式时，必须在进入计数停止状态(将TSTART0位和TSTART1位置“0”)后进行切换。
- 在将TSTART0位和TSTART1位从“0”改为“1”前，必须将TMIF0位和TMIF1位置“0”。详细内容请参照“第25章 中断功能”。

10.7.3 计数源

- 要切换计数源时，必须在停止计数后进行切换。
[变更步骤]
 - 将TMSTR寄存器的TSTARTi位($i=0, 1$)置“0”(停止计数)。
 - 更改TMCRi寄存器的TCK0~TCK2位。
- 要选择 F_{HOCO} (64MHz或者48MHz)作为定时器M的计数源时，必须在将外围允许寄存器1(PER1)的bit4(TMMEN)置位前将 F_{CLK} 设置为 F_{IH} 。如果要将 F_{CLK} 改为 F_{IH} 以外的时钟，就必须在清除外围允许寄存器1(PER1)的bit4(TMMEN)后进行更改。

10.7.4 输入捕捉功能

- 输入捕捉信号的脉宽必须至少为3个定时器M的运行时钟周期。
- 在输入捕捉信号从TMIOj脚($j=A, B, C, D$)输入后，需要等待2~3个定时器M的运行时钟(F_{CLK})周期，然后将Tmi寄存器的值传送到TMGRj寄存器(没有数字滤波器的情况)。
- 在输入捕捉模式中，如果在TMSTR寄存器的TMTSTARTi位为“0”(停止计数)时给TMIOj脚输入TMIORk寄存器的IOj0位和IOj1位选择的边沿，就在TMIOj输入的有效边沿产生输入捕捉中断请求($i=0, 1, j=A, B, C, D, k=A, C$)。

10.7.5 TMIOAi、TMIOBi、TMIOCi、TMIODi引脚的设置步骤($i=0, 1$)

- 在复位后，TMIOAi、TMIOBi、TMIOCi、TMIODi引脚的复用I/O端口用作输入端口。
- 要从TMIOAi、TMIOBi、TMIOCi、TMIODi引脚输出时，必须按照以下步骤进行设置。
[变更步骤]
 - 进行模式和初始值的设置。
 - 将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚置为允许输出(TMOER1寄存器)。
 - 将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚对应的端口寄存器的位置“0”。
 - 将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚对应的端口模式寄存器的位设置为输出模式(从TMIOAi、TMIOBi、TMIOCi、TMIODi引脚开始输出)。
 - 开始计数(将TSTART0位和TSTART1位置“1”)。
- 要将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚对应的端口模式寄存器的位从输出模式改为输入模式时，必须按照以下步骤进行设置。
[变更步骤]
 - 将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚对应的端口模式寄存器的位设置为输入模式(从TMIOAi、TMIOBi、TMIOCi、TMIODi引脚开始输入)。
 - 设置为输入捕捉功能。
 - 开始计数(将TSTART0位和TSTART1位置“1”)。
- 在将TMIOAi、TMIOBi、TMIOCi、TMIODi引脚从输出模式切换为输入模式时，可能根据引脚的状态进行输入捕捉的运行。当不使用数字滤波器时，至少在经过2个CPU时钟周期后进行边沿检测；当使用数字滤波器时，最多需要5个数字滤波器的采样时钟周期进行边沿检测。

10.7.6 外部时钟TMCLK

TMCLK引脚输入的外部时钟的脉宽必须至少为3个定时器M的运行时钟周期。

复位同步PWM模式

- 当此模式用于马达控制时，必须在OLS0=OLS1的条件下使用。
- 要设置为复位同步PWM模式时，必须按照以下步骤进行设置。

[变更步骤]

- 1) 将TMSTR寄存器的TSTART0位置“0”(停止计数)。
- 2) 将TMFCR寄存器的CMD1位和CMD0位置“00B”(定时器模式、PWM模式和PWM3模式)。
- 3) 将CMD1位和CMD0位置“01B”(复位同步PWM模式)。
- 4) 重新设置定时器M的其他相关寄存器。

10.7.7 互补PWM模式

- 当此模式用于马达控制时，必须在OLS0=OLS1的条件下使用。
- 要更改TMFCR寄存器的CMD0位和CMD1位时，必须按照以下步骤进行更改。

[变更步骤：设置为互补PWM模式(包括重新设置)的情况，或者在互补PWM模式中更改缓冲寄存器到通用寄存器的数据传送时序的情况]

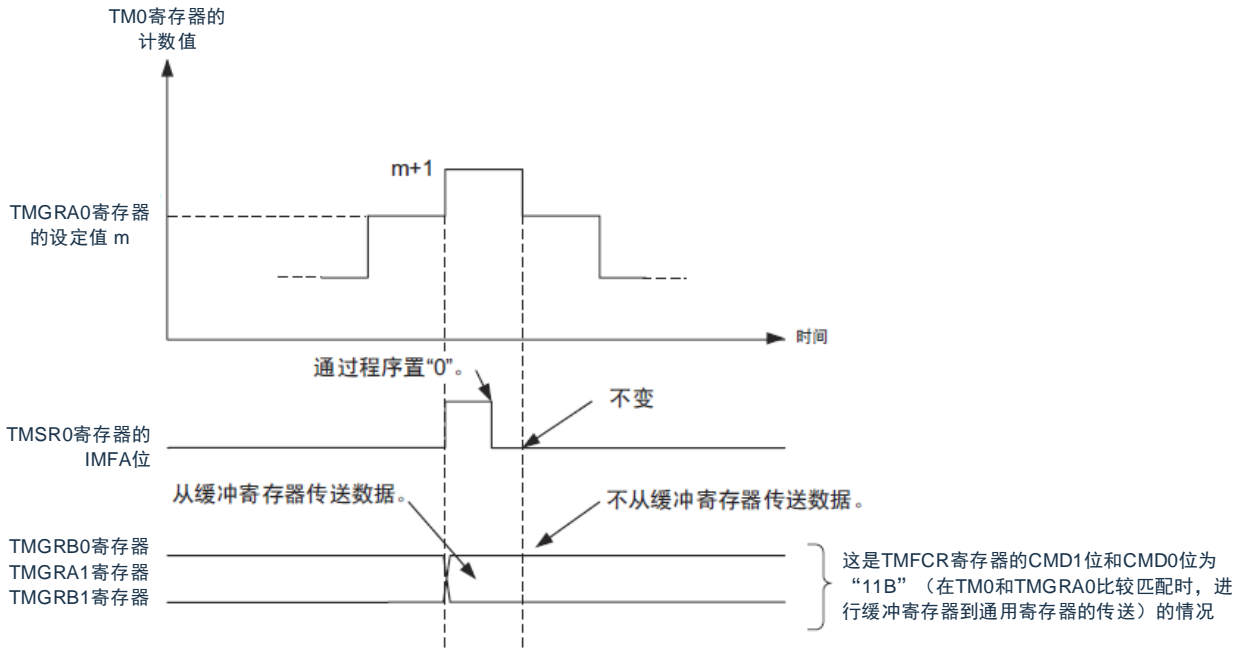
- 1) 将TMSTR寄存器的TSTART0位和TSTART1位都置“0”(停止计数)。
- 2) 将TMFCR寄存器的CMD1位和CMD0位置“00B”(定时器模式、PWM模式和PWM3模式)。
- 3) 将CMD1位和CMD0位置“10B”或者“11B”(互补PWM模式)。
- 4) 重新设置定时器M的其他相关寄存器。

[变更步骤：中止互补PWM模式的情况]

- 1) 将TMSTR寄存器的TSTART0位和TSTART1位都置“0”(停止计数)。
- 2) 将CMD1位和CMD0位置“00B”(定时器模式、PWM模式和PWM3模式)

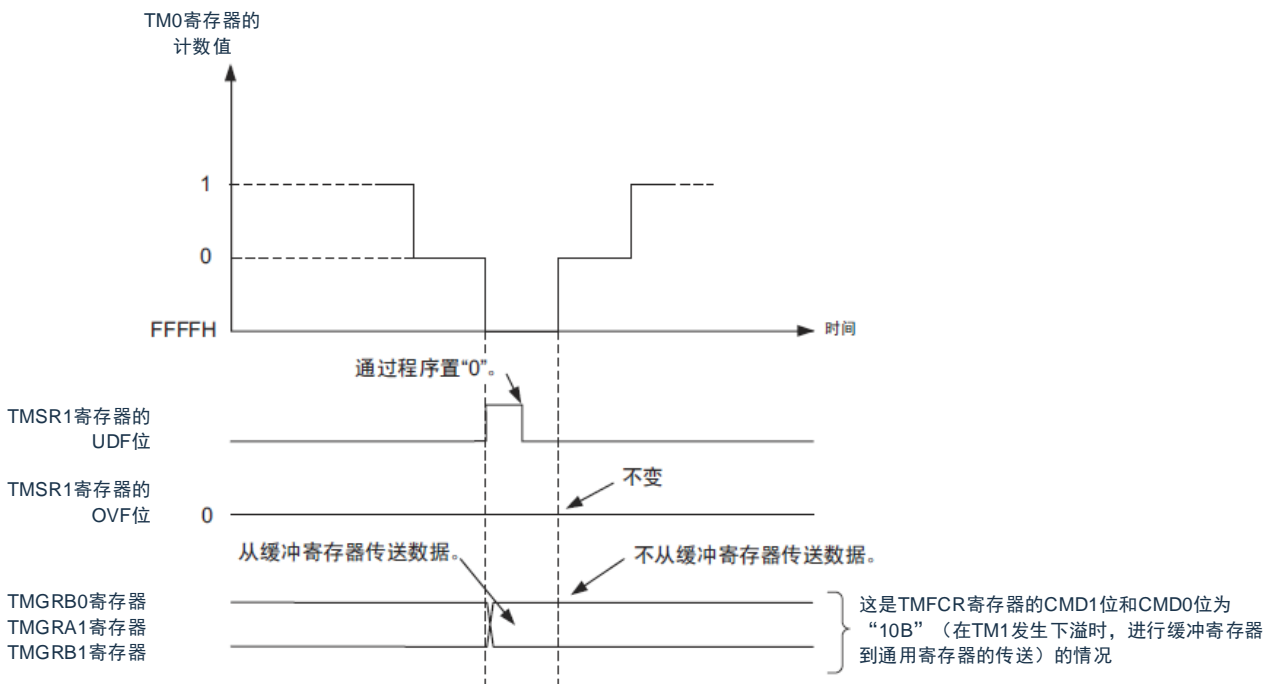
- 不能在运行过程中写TMGRA0、TMGRB0、TMGRA1、TMGRB1寄存器。
- 要更改PWM波形时，必须通过缓冲器运行将TMGRD0、TMGRC1、TMGRD1寄存器的写入值传送到TMGRB0、TMGRA1、TMGRB1寄存器。但是，在写TMGRD0、TMGRC1和TMGRD1时，必须在先将TMBFD0位、TMBFC1位和TMBFD1位置“0”(通用寄存器)，然后给这些寄存器写数据。此后，能将TMBFD0位、TMBFC1位和TMBFD1位置“1”(缓冲寄存器)。不能更改PWM周期。
- 假设TMGRA0寄存器的设置值为m，则TM0寄存器在从递增计数变为递减计数时进行m-1→m→m+1→m→m-1的计数。
- 在进行m→m+1的递增计数时，TMSRi寄存器的IMFA位变为“1”。当TMFCR寄存器的CMD1位和CMD0位为“11B”(互补PWM模式，在TM0和TMGRA0寄存器比较匹配时传送缓冲数据)时，将缓冲寄存器(TMGRD0、TMGRC1、TMGRD1)的内容传送到通用寄存器(TMGRB0、TMGRA1、TMGRB1)。在进行m+1→m→m-1的递减计数时，IMFA位不变，并且也不将数据传送到TMGRA0等寄存器。

图10-64：互补PWM模式中的TM0和TMGRA0寄存器比较匹配时的运行例子



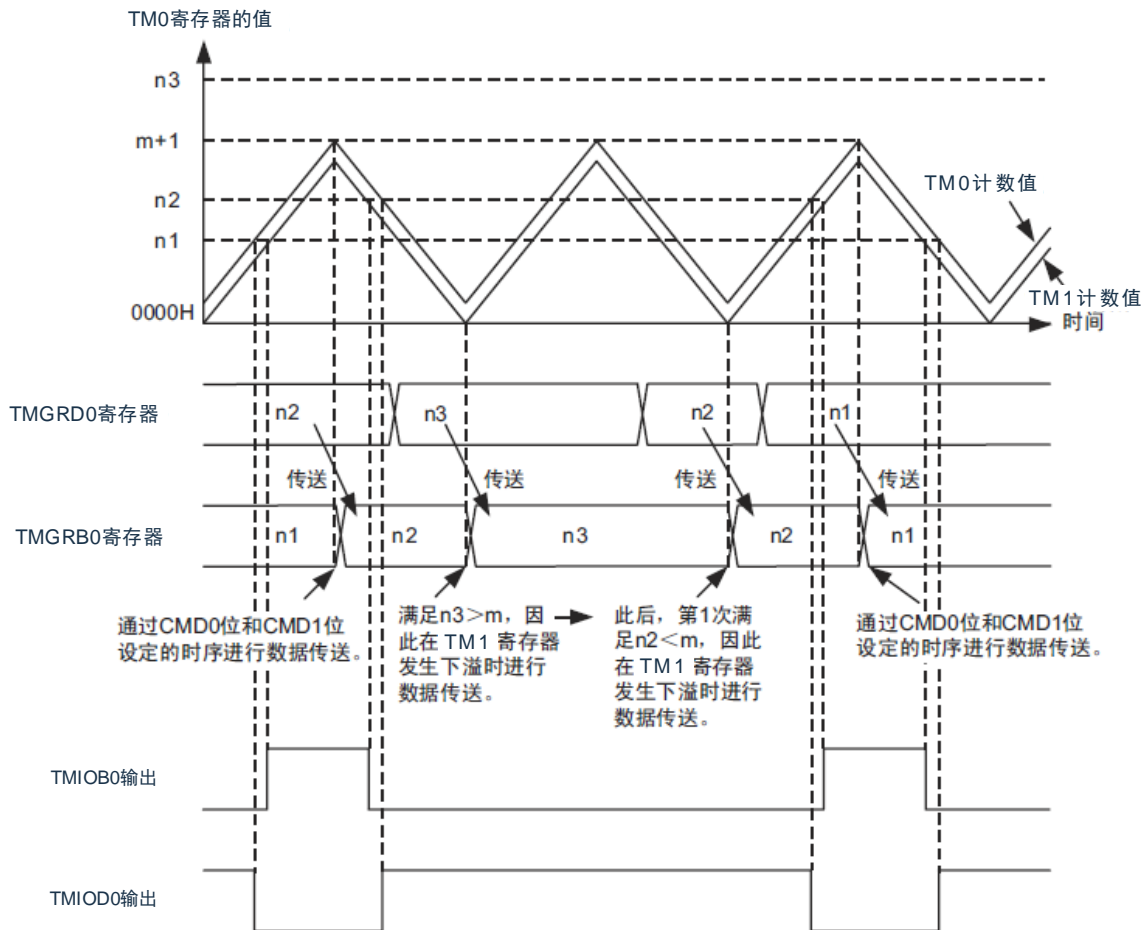
- 在从递减计数变为递增计数时，TM1进行1→0→FFFFH→0→1的计数。在进行1→0→FFFFH的递减计数时，TMSRi寄存器的UDF位变为“1”。当TMFCR寄存器的CMD1位和CMD0位为“10B”(互补PWM模式，在TM1发生下溢时传送缓冲数据)时，将缓冲寄存器(TMGRD0、TMGRC1、TMGRD1)的内容传送到通用寄存器(TMGRB0、TMGRA1、TMGRB1)。在进行FFFFH→0→1的递增计数时，不将数据传送到TMGRB0等寄存器，并且TMSRi寄存器的OVF位不变。

图10-65：互补PWM模式中的TM1发生下溢时的运行例子



- 必须通过TMFCR寄存器的CMD0位和CMD1位选择缓冲寄存器到通用寄存器的数据传送时序。但是，在0%占空比和100%占空比的情况下，与CMD0位和CMD1位的值无关，为以下的传送时序。
缓冲寄存器的值 \geq TMGRA0寄存器的值的情况(0%占空比)在TM1寄存器发生下溢时进行数据传送。此后，如果给缓冲寄存器设置值($0001H \leq$ 设置值 $<$ TMGRA0寄存器的值)，就在设置后TM1寄存器发生第1次下溢时将数据传送到通用寄存器。然后，通过CMD0位和CMD1位选择的时序进行数据传送。
但是，在缓冲寄存器的初始值为“FFFFH”时无法产生0%占空比的波形。要产生0%占空比的波形时，必须通过写缓冲寄存器使缓冲寄存器的值 \geq TMGRA0寄存器的值。

图10-66：互补PWM模式中的缓冲寄存器的值 \geq TMGRA0寄存器的值时的运行例子



如果给缓冲寄存器设置值(设置值 \geq TMGRA0寄存器的值)，就在TM1计数器发生下溢时将缓冲寄存器的值传送到通用寄存器，并且固定为正相100%占空比和反相0%占空比的输出电平，而与CMD0位的设置无关。

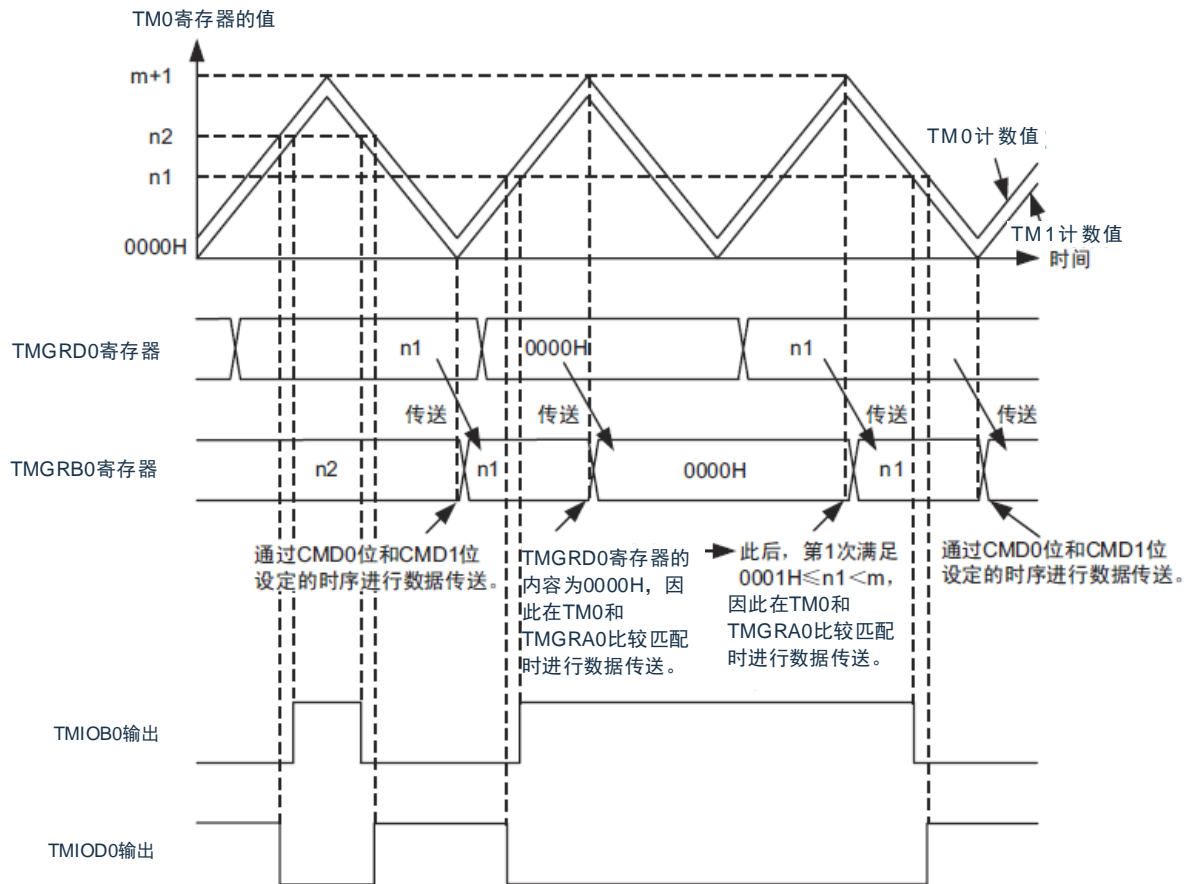
要解除固定的输出电平时，必须给缓冲寄存器设置值(TM0寄存器的值 \leq 设置值 \leq (TMGRA0的值-TM0寄存器的值))。在写缓冲器后，与CMD0位的设置无关，在TM1计数器发生下溢时将缓冲寄存器的值传送到通用寄存器并且输出PWM波形。在输出PWM波形后，通过CMD0位设置的时序将缓冲寄存器的值传送到通用寄存器。

但是，不能用缓冲寄存器的初始值“FFFFH”设置正相100%占空比和反相0%占空比的输出。也不能直接从正相100%占空比和反相0%占空比的输出改为正相0%占空比和反相100%占空比的输出。

缓冲寄存器的值为“0000H”的情况(100%占空比)在TM0和TMGRA0寄存器比较匹配时进行数据传送。

此后，如果给缓冲寄存器设置值($0001H \leq$ 设置值 $<$ TMGRA0寄存器的值)，就在设置后TM0和TMGRA0寄存器发生第1次比较匹配时，将数据传送到通用寄存器。然后，通过CMD0位和CMD1位选择的时序进行数据传送。

图10-67：互补PWM模式中的缓冲寄存器的值为“0000H”时的运行例子



如果给缓冲寄存器写“0000H”，就在TM0寄存器和TMGRA0寄存器比较匹配时将缓冲寄存器的值传送到通用寄存器，并且固定为正相0%占空比和反相100%占空比的输出电平，而与CMD0位的设置无关。

要解除固定的输出电平时，必须给缓冲寄存器设置值(TM0寄存器的值≤设置值≤(TMGRA0的值-TM0寄存器的值))。在写缓冲器后，与CMD0位的设置无关，在TM1计数器发生下溢时将缓冲寄存器的值传送到通用寄存器并且输出PWM波形。在输出PWM波形后，通过CMD0位设置的时序将缓冲寄存器的值传送到通用寄存器。

不能直接从正相0%占空比和反相100%占空比的输出改为正相100%占空比和反相0%占空比的输出。

10.8 PWMOP

PWMOP单元可以实现TimerM的输出强制截止功能。截止源可以从CMP0, INTPO以及EVENT中选择。这和TimerM自身的脉冲强制截止功能不同。

表10-20: 脉冲输出的强制截止和PWMOP的功能比较

	TimerM 脉冲输出的强制截止	PWMOP 的输出强制截止
支持强制截止的模式	PWM 功能 复位同步 PWM 模式 互补 PWM 模式 PWM3 模式	支持 TimerM 的所有模式 port 输出也可以被强制截止
强制截止的源	EVENTC 的事件输入 INTPO 输入	EVENTC 的事件输入 INTPO 输入 比较器 0 的输出
强制截止解除的源	停止计数器, 通过软件解除	硬件解除 软件解除(不需要停止计数器)
可以强制截止的引脚	Pxx/TMIOA0,Pxx/TMIOB0, Pxx/TMIOC0,Pxx/TMIOD0, Pxx/TMIOA1,Pxx/TMIOB1, Pxx/TMIOC1,Pxx/TMIOD1. (根据 TimerM 输出设置决定)	Pxx/TMIOA0,Pxx/TMIOB0, Pxx/TMIOC0,Pxx/TMIOD0, Pxx/TMIOA1,Pxx/TMIOB1, Pxx/TMIOC1,Pxx/TMIOD1. PORT 的输出也可以被强制截止
强制截止时的 PORT 状态	HI-Z L 电平 H 电平	HI-Z L 电平 H 电平 如果选择的是 PORT 输出的强制截止, 只能输出 HI-Z

注意: 同时使用脉冲强制截止和输出强制截止机能时, 不能选择同一源。

10.8.1 PWMOP的功能

PWMOP可以实现下列功能：

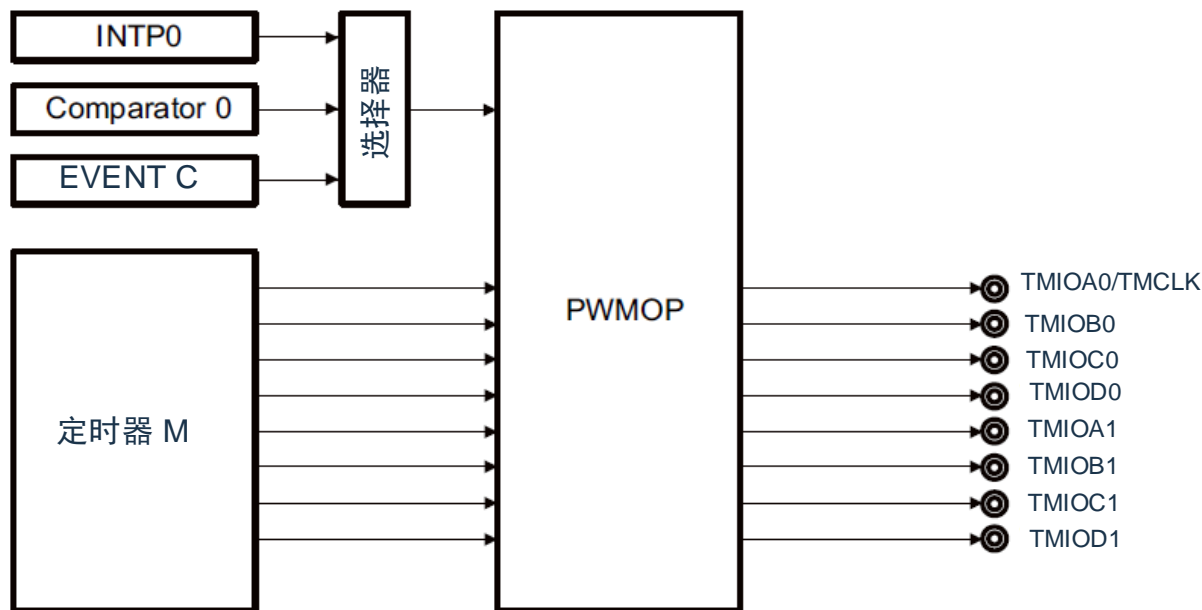
可以选择比较器0的输出，INTP0输入以及EVENTC的事件输入作为输出强制截止源。

选择比较器0的输出和INTP0输入作为源的时候，可以选择沿检出。

可以通过软件或硬件解除输出强制截止。

强制截止时，输出电平可以选择“H”电平，“L”电平和Hi-Z。

图10-68：PWMOP的框图



10.8.2 PWMOP的寄存器

PWMOP的寄存器如表10-21所示。

表10-21：PWMOP的控制寄存器

寄存器名	符号
PWMOP控制寄存器0	OPCTL0
PWMOP强制截止控制寄存器0	OPDF0
PWMOP强制截止控制寄存器1	OPDF1
PWMOP沿选择寄存器	OPEDGE
PWMOP状态寄存器	OPSR

(1) PWMOP控制寄存器0(OPCTL0)

图10-69: PWMOP控制寄存器0的格式

地址: 0x40043C58 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
OPCTL0	0	HAZAD_SET	IN_EG	IN_SEL1	IN_SELO	ACT	HZ_REL	HS_SEL

HAZAD_SET	输出强制截止hazard控制 ^{注1}
0	禁止hazard对策
1	允许hazard对策

IN_EG	输出强制截止的源/输出强制截止解除的源的沿的选择 ^{注2、3}
0	上升沿: 输出强制截止 下降沿: 输出强制截止解除
1	上升沿: 输出强制截止解除 下降沿: 输出强制截止

IN_SEL1	IN_SELO	截止源的选择 ^{注2、4、5}
0	0	不选择
0	1	比较器 0 的输出
1	0	INTP0 输入
1	1	来自 EVENTC 的事件输入

ACT	软件解除时的软件解除时序选择
0	通过软件将 HZ_SEL 置 1 时, 强制截止解除, 脉冲输出恢复
1	HZ_SEL 为 1 时, 强制截止解除, 脉冲输出恢复的时序如下: 定时器 M 互补 PWM 模式: 根据 OPEDGE 选择的 TMIOC0 的沿解除强制截止, 脉冲输出恢复 定时器 M 复位同步 PWM 模式: TM0 的计数值为 0000H 时, 脉冲输出恢复 上述模式以下: TM0 的计数值为 0000H 时, TMIOj0(j=A,B,C,D)强制截止解除 TM1 的计数值为 0000H 时, TMIOj1(j=A,B,C,D)强制截止解除 ^{注6}

HZ_REL	软件解除时: 输出强制截止解除控制 ^{注7}
0	保持输出强制截止(输出强制截止解除时, HZ_REL bit 变为 0)
1	强制截止解除, 脉冲输出恢复 ^{注8}
HZ_REL bit 的读出值根据状态不同: 通常状态时: 1/0 写入, 0 读出 输出强制截止时: 只能写入 1 和读出 1	

HS_REL	输出强制截止的模式选择
0	硬件解除： 使用硬件解除输出强制截止的时候，根据定时器 M 的动作模式的不同，时序也不同。 定时器 M 互补 PWM 模式：监测到解除源之后，根据 OPEDGE 选择的 TMIOCO 的沿解除强制截止。 定时器 M 复位同步 PWM 模式：TM0 的计数值为 0000H 时，脉冲输出恢复 TM0 的计数值为 0000H 时，TMIOj0(j=A,B,C,D)强制截止解除 TM1 的计数值为 0000H 时，TMIOj1(j=A,B,C,D)强制截止解除 ^{注9}
1	软件解除

注1：定时器M动作过程中不能改写

注2：在IN_EG设置后至少间隔三个时钟，再设置IN_SEL1和IN_SEL0。

注3：在选择比较器0输出和INTP0输入时使能。

注4：使用EVENTC解除强制截止时，必须选择软件解除(HS_SEL置为1)。使用比较器0输出和INTP0输入时没有限制。

注5：选择比较器0输出和INTP0输入的有效宽度必须大于一个时钟周期。

注6：TM0和TM1的计数值=0000H是指，TM0和TM1在动作过程中，计数器bit15~bit0全部为0的时刻。

注7：如果定时器M工作在输出比较功能，PWM功能或者PWM3模式下，使用2个通道和使用1个通道，截止输出的解除动作时不同的。

使用2通道时：

如果通过软件设置HZ_REL为1，输出截止bit(HZOF0,HZOF1)全部变为0，HZ_REL bit也变为0。

使用1通道时：

如果通过软件设置HZ_REL为1，当前使用的通道对应的输出截止bit(HZOF0或者HZOF1)变为0，HZ_REL bit也变为0。

注8：如果没有发生强制截止，不能设为1

注9：如果定时器M工作在输出比较功能，PWM功能或者PWM3模式下，发生强制截止的解除时，没有动作的通道不能解除强制输出截止状态。(HZOF0和HZOF1不会变为0)。

(2) PWMOP强制截止控制寄存器0(OPDF0)

图10-70: PWMOP强制截止控制寄存器0的格式

地址: 0x40043C59 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
OPDF0	DFD01	DFD00	DFC01	DFC00	DFB01	DFB00	DFA01	DFA00

DFD01	DFD00	TMIOD0 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFC01	DFC00	TMIOC0 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFB01	DFB00	TMIOB0 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFA01	DFA00	TMIOA0 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

注意:

1. 当TMIOj0(j=A,B,C,D)引脚用作PORT输出并且使能强制截止功能时, 必须选择Hi-Z输出。
2. 强制截止状态中不能更改寄存器的值。
3. 当使用PIOR来重定向TMIOji(j=A,B,C,D;i=0,1)引脚时, 对同一引脚只能进行单一设置。
例: 使用PIOR2选择P17作为TMIOD0引脚, 并且允许TMIOD0输出时, 必须设置DFA0n(n=0,1)为0(禁止强制截止功能)。

(3) PWMOP强制截止控制寄存器1(OPDF1)

图10-71：PWMOP强制截止控制寄存器1的格式

地址：0x40043C5A

复位后：00H R/W

符号	7	6	5	4	3	2	1	0
OPDF1	DFD11	DFD10	DFC11	DFC10	DFB11	DFB10	DFA11	DFA10

DFD11	DFD10	TMIOD1 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFC11	DFC10	TMIOC1 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFB11	DFB10	TMIOB1 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

DFA11	DFA10	TMIOA1 引脚输出强制截止控制
0	0	禁止强制截止功能
0	1	输出 Hi-Z
1	0	输出 L 电平
1	1	输出 H 电平

注意：

1. 当TMIOj1(j=A,B,C,D)引脚用作PORT输出并且使能强制截止功能时，必须选择Hi-Z输出。
2. 强制截止状态中不能更改寄存器的值。
3. 当使用PIOR来重定向TMIOji(j=A,B,C,D;i=0,1)引脚时，对同一引脚只能进行单一设置。

例：使用PIOR2选择P16作为TMIOA1引脚，并且允许TMIOA1输出时，必须设置DFC1n(n=0,1)为0(禁止强制截止功能)。

(4) PWMOP沿选择寄存器(OPEDGE)

在定时器M工作在互补PWM模式并且通过硬件解除输出强制截止时，可以通过OPEDGE寄存器设置解除时间点。

图10-72: PWMOP沿选择寄存器的格式

地址: 0x40043C5B 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
OPEDGE	-	-	-	-	-	-	EG1	EG0

EG1	EG0	输出强制截止解除的沿选择
0	0	在 TMIOC0 的上升沿解除
0	1	在 TMIOC0 的下降沿解除
1	0	在 TMIOC0 的上升沿或者下降沿解除
1	1	不使能沿选择，保持截止

(5) PWMOP状态寄存器(OPSR)

图10-73: PWMOP状态寄存器的格式

地址: 0x40043C5C 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	HZOF1	HZOF0	HZIF0

HZOF1	强制截止状态
0	通常 Timer 输出(TMIOA1, TMIOB1, TMIOC1, TMIOD1)
1	强制截止状态(TMIOA1, TMIOB1, TMIOC1, TMIOD1)

HZOF0	强制截止状态
0	通常 Timer 输出(TMIOA0, TMIOB0, TMIOC0, TMIOD0)
1	强制截止状态(TMIOA0, TMIOB0, TMIOC0, TMIOD0)

HZIF0	输出强制截止源的状态 ^{注1, 2}
0	输出强制截止源在阈值范围内
1	输出强制截止源超出了阈值范围

注1: 在通过设置IN_SEL1, IN_SEL0来选择INTP0输入, 比较器0输出作为强制截止要因前, 如果输出强制截止源超出了阈值范围, 在IN_SEL1和IN_SEL0设置后, HZIF0 bit会被设置为1, 而HZOF0和HZOF1不会被置起。

注2: 只在选择INTP0输入和比较器0输出时有效。

10.8.3 PWMOP的运行

可以选择比较器0的输出，INTP0输入以及EVENTC的事件输入作为输出强制截止源。

选择比较器0的输出和INTP0输入作为源的时候，可以选择沿检出。

10.8.3.1 输出强制截止

可以选择比较器0的输出，INTP0输入以及EVENTC的事件输入作为触发事件来强制截止TMIO_{ji}(j=A,B,C,D;i=0,1)的输出。

当触发事件被检测到时，定时器M的输出被强制截止，寄存器OPDF0/OPDF1的设置值被输出。详细动作请参见图10-75。

通过OPCTL0寄存器的HS_SELbit来选择通过软件或硬件来解除强制截止功能。

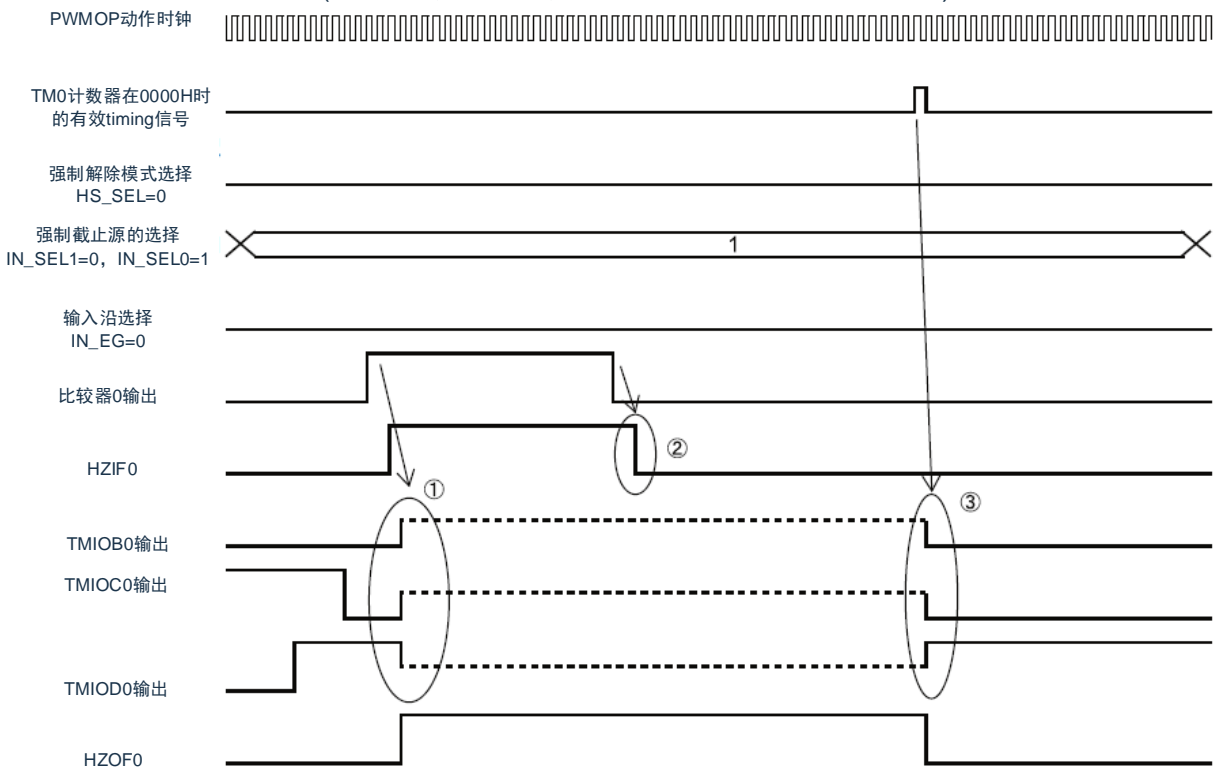
10.8.3.2 硬件解除(HS_SEL=0)

定时器M工作在不同模式时，解除时序不同：

(1) 互补PWM功能以外的输出场合

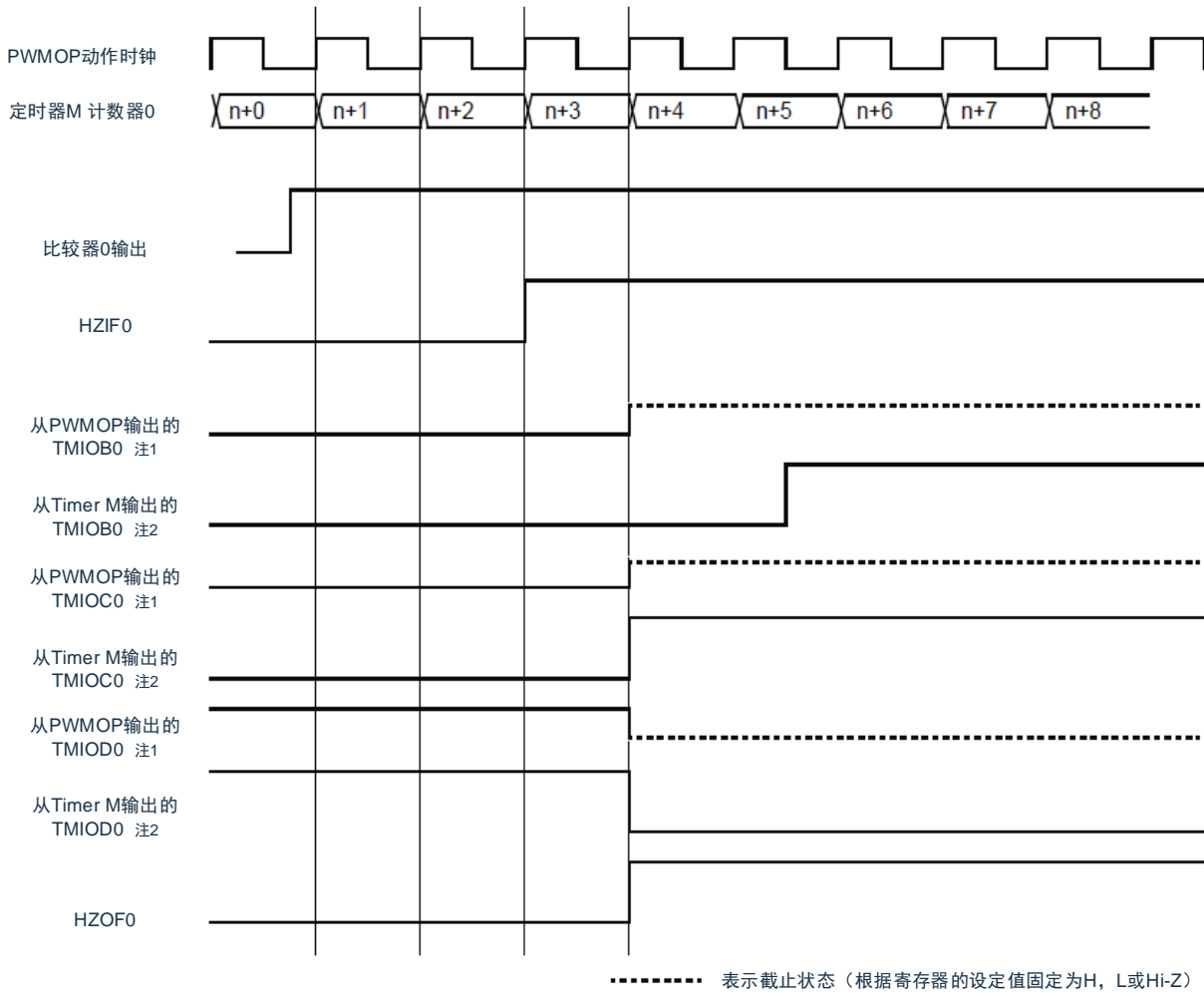
- 定时器M工作在输出比较功能，PWM功能或者PWM3模式下时，解除事件发生后，当TM0的计数值为0000H时，TMIOj0(j=A,B,C,D)的强制截止解除。当TM0的计数值为0000H时，TMIOj1(j=A,B,C,D)的强制截止解除。
- 定时器M工作复位同期PWM模式下时，解除事件发生后，当TM0的计数值为0000H时，所有的TMIO引脚的强制截止全部解除。

图10-74：输出强制截止/硬件解除输出强制截止的例子
(TMIOB0, TMIOC0, TMIOD0引脚被强制截止的例子)



- 1) 当比较器输出信号的上升沿被检测到时，TMIOB0，TMIOC0, TMIOD0引脚输出处于强制截止状态。
- 2) 比较器输出信号的下降沿被检测到后，HZIF0 bit被清除。
- 3) 在TDi计数值为0000H时，强制截止状态解除。

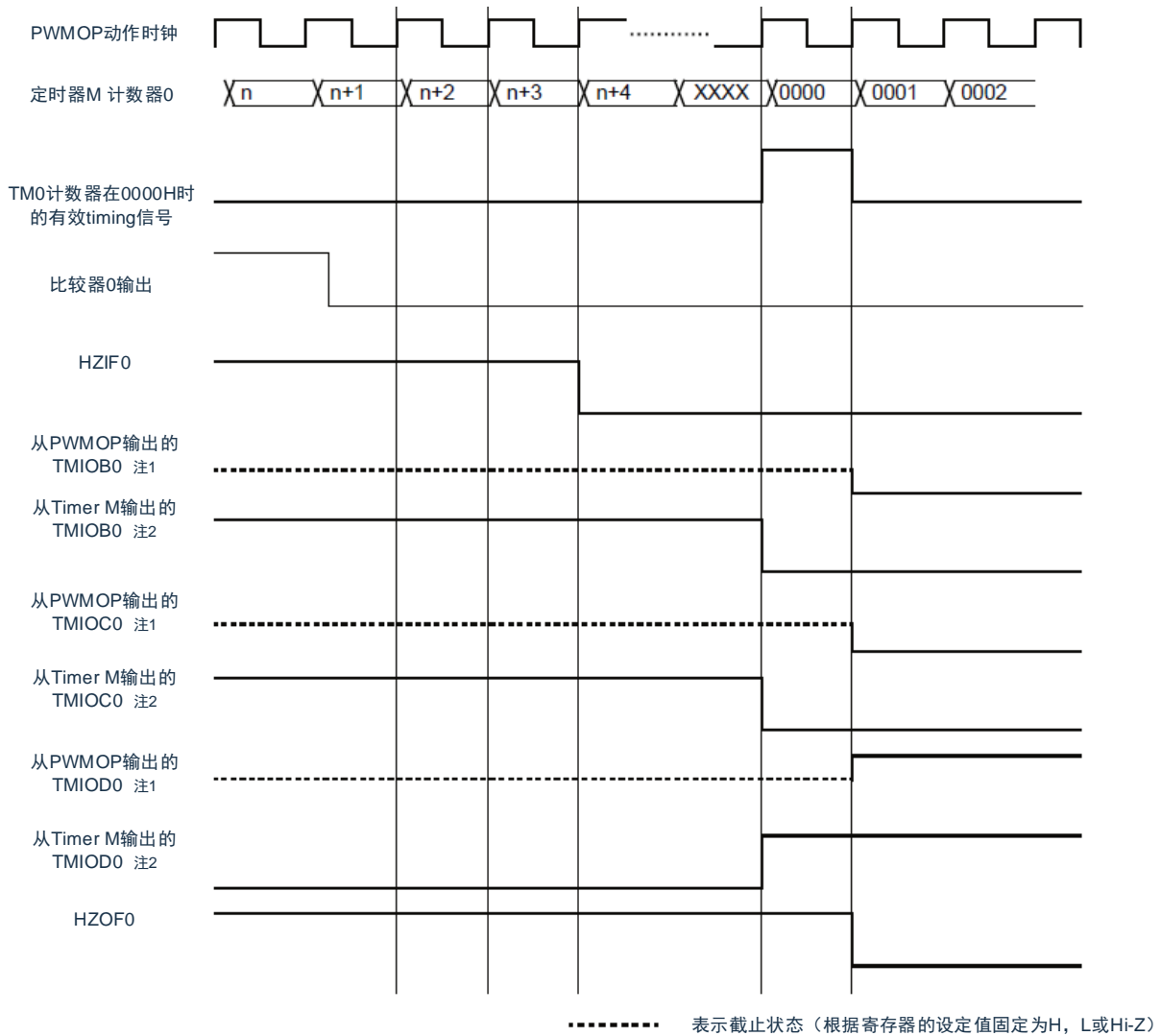
图10-75：强制截止的详细时序图



注1：从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2：从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

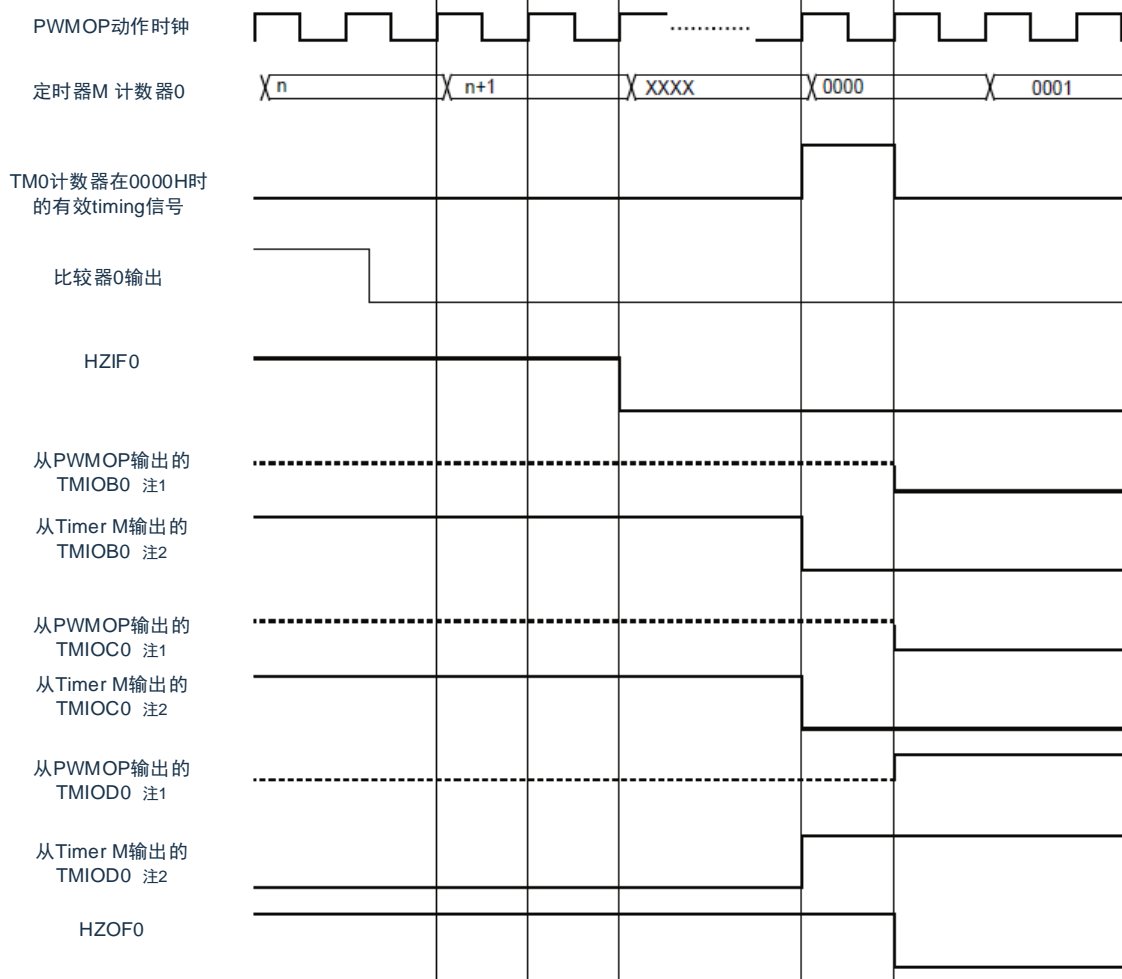
图10-76：强制截止解除的详细时序图(定时器M的计数源=Fclk)



注1：从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2：从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

图10-77：强制截止解除的详细时序图(定时器M的计数源= $F_{CLK}/2$)



----- 表示截止状态（根据寄存器的设定值固定为H, L或Hi-Z）

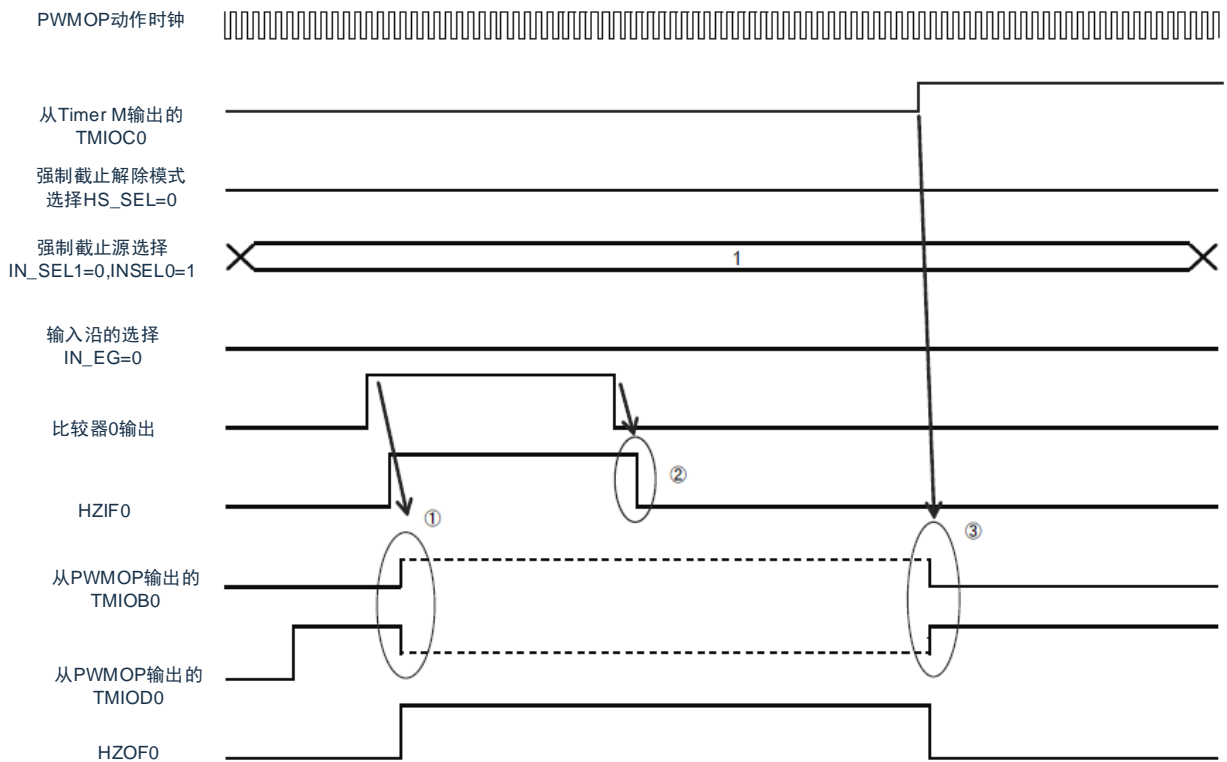
注1：从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2：从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

(2) 互补PWM功能输出的场合

监测到解除源之后，根据OPEDGE选择的TMIOC0的沿解除强制截止。

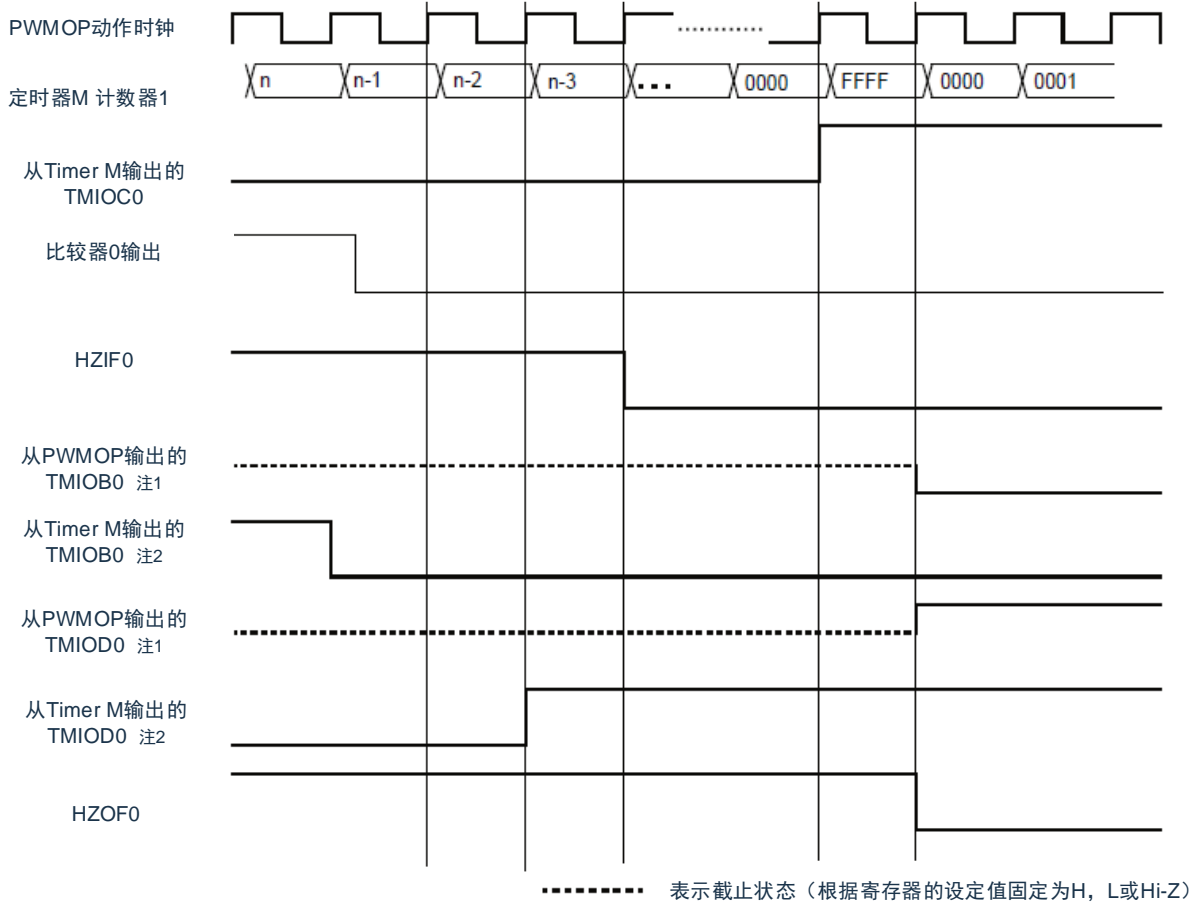
图10-78：硬件解除强制截止的例子(以TMIOB0, TMIOD0为例)



- 1) 当比较器0输出信号的上升沿被检测到时，TMIOB0, TMIOD0引脚的输出处于强制截止状态。
- 2) 当比较器0输出信号的下降沿被检测到后，HZIF0 bit被清除。
- 3) 强制截止状态在TMIOC0的上升沿发生时解除。

强制截止的详细时序请参考图10-75。

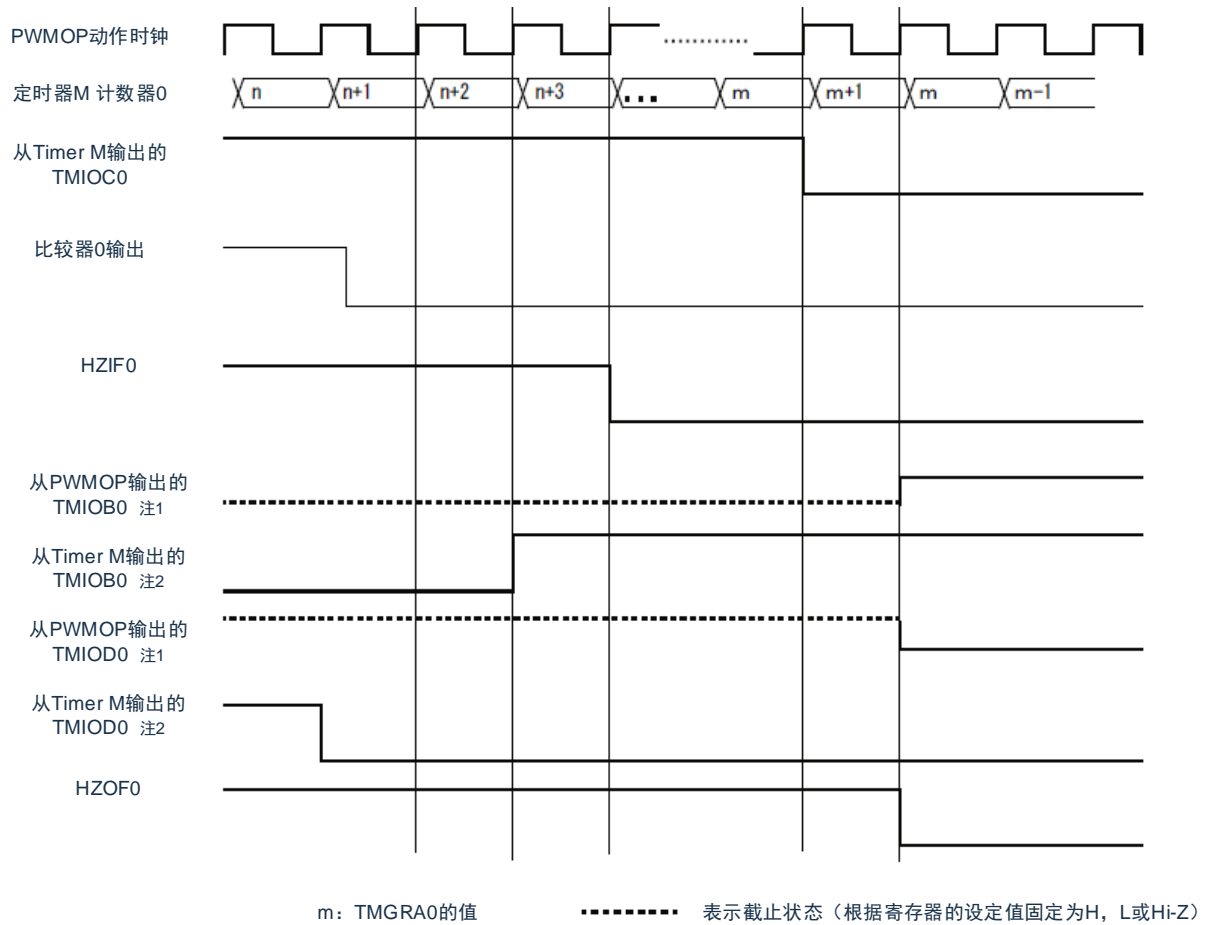
图10-79: 强制截止解除的详细时序图(定时器M的计数源= F_{CLK} , 递减计数)



注1: 从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2: 从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

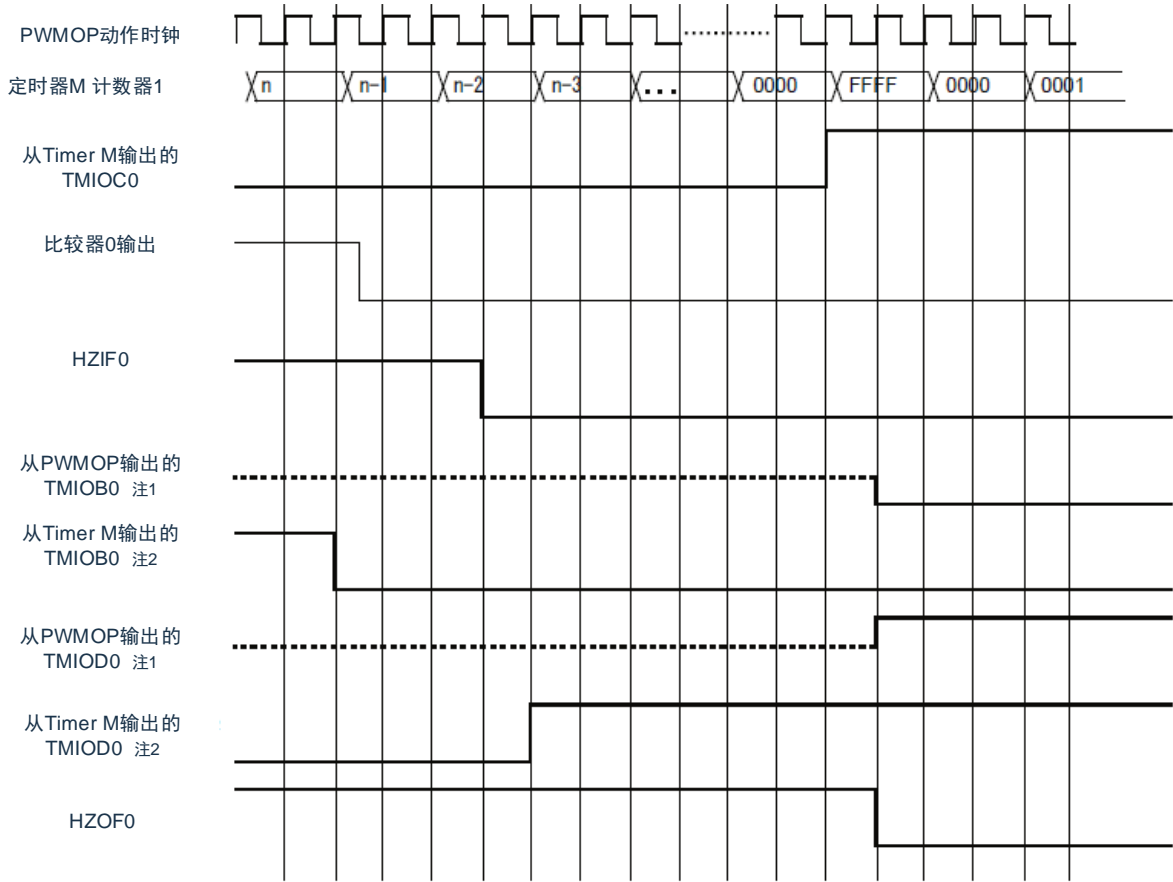
图10-80: 强制截止解除的详细时序图(定时器M的计数源=F_{CLK}, 计数器=TMGRA0)



注1: 从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2: 从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

图10-81：强制截止解除的详细时序图(定时器M的计数源= $F_{CLK}/2$ ，递减计数)

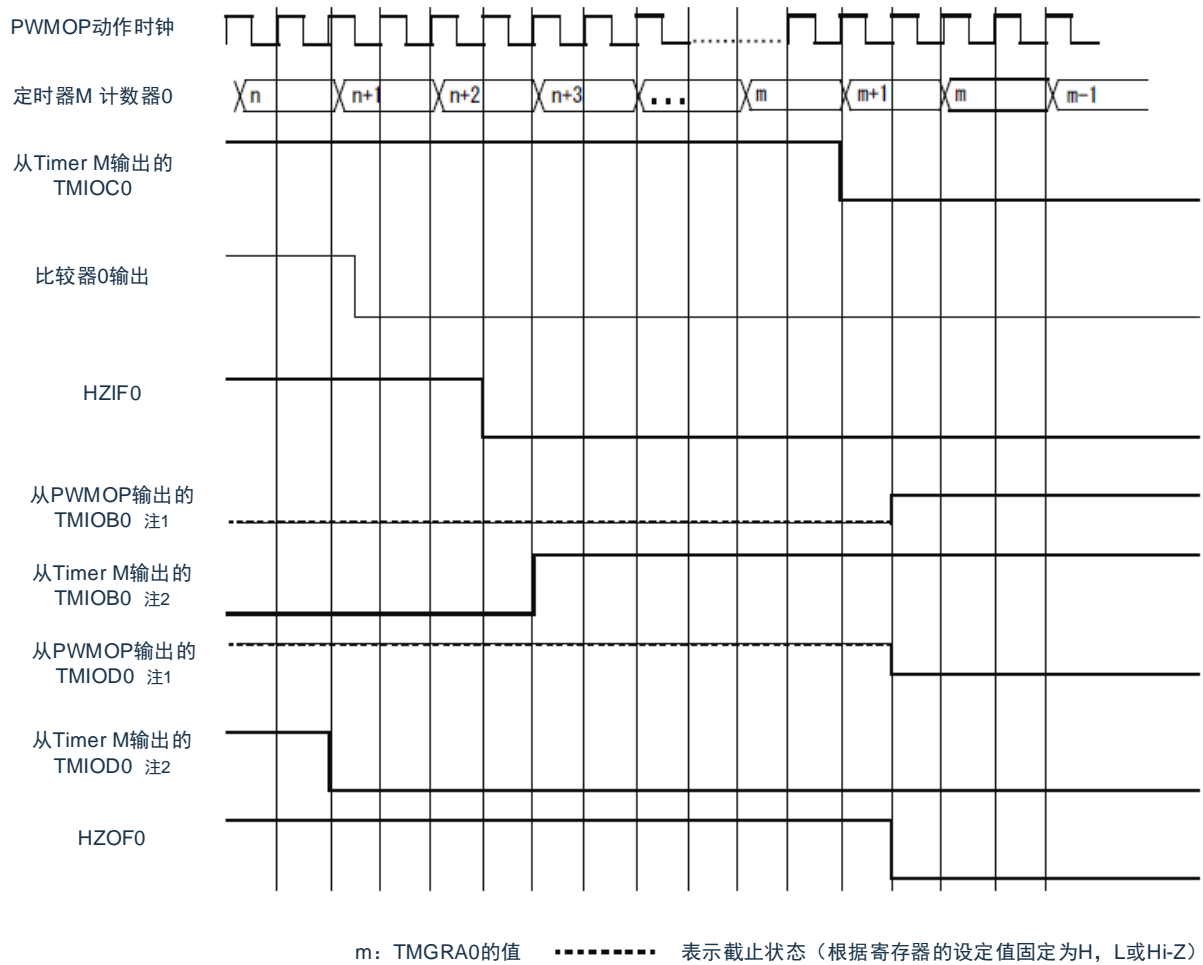


..... 表示截止状态（根据寄存器的设定值固定为H, L或Hi-Z）

注1：从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2：从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

图10-82: 强制截止解除的详细时序图(定时器M的计数源= $F_{CLK}/2$, 计数器=TMGRA0)



注1: 从PWMOP输出的TMIO*(*=B~D)表示复用定时器M功能的引脚的状态。

注2: 从Timer M输出的TMIO*(*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

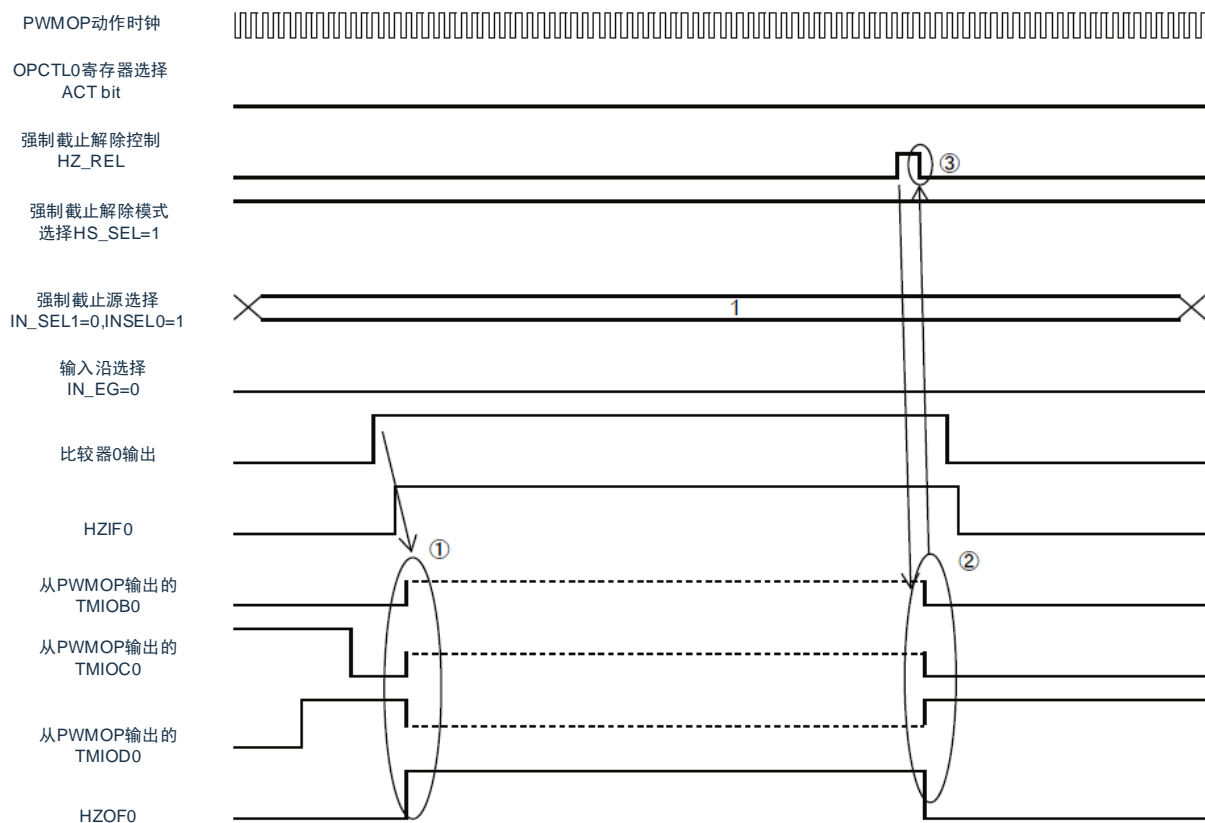
10.8.3.3 软件解除(HS_SEL=1)

OPCTL0的ACT设置不同时，强制截止解除时序不同。

(1) 使用软件立即解除截止(ACT=0)

如果设置了ACT=0，一旦设置了OPCTL0寄存器的HZ_REL bit为1，强制截止立即就被解除。解除后HZ_REL bit会被自动置0。

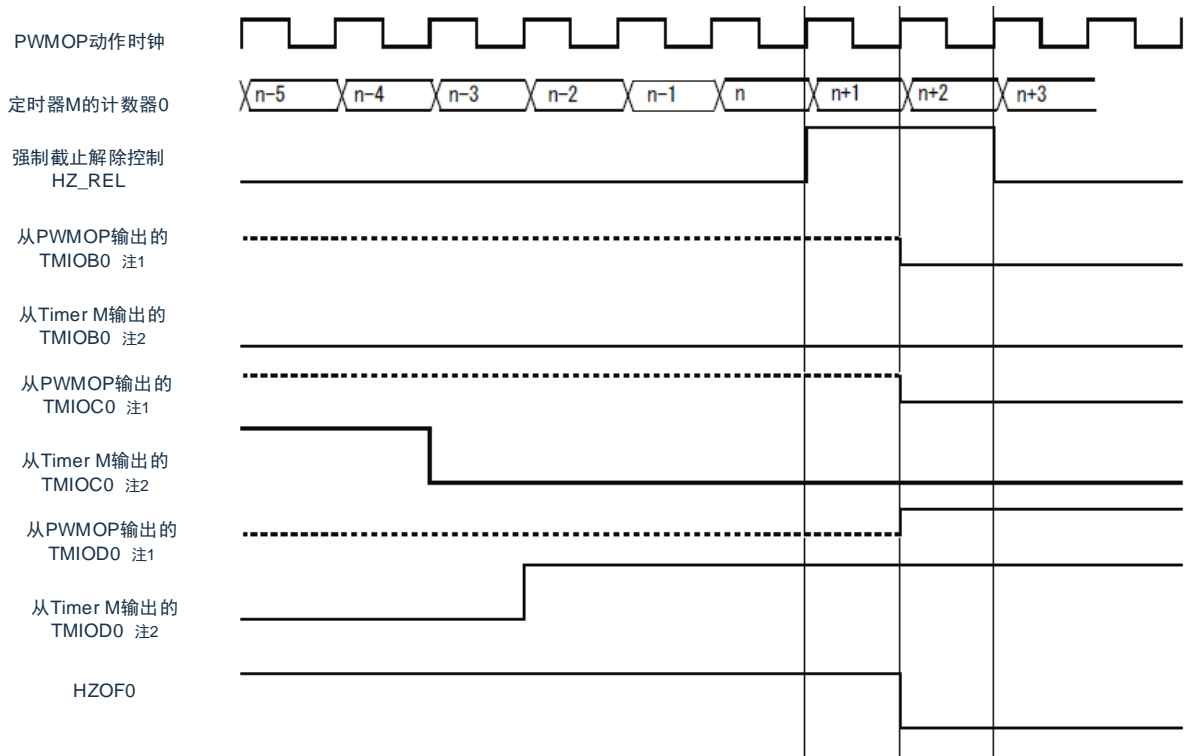
图10-83：软件解除强制截止的例子(以TMIOB0, TMIOC0, TMIOD0的例子)



- 当比较器0输出信号的上升沿被检测到时，TMIOB0, TMIOC0, TMIOD0引脚的输出处于强制截止状态。
- 设置HZ_SEL为1，强制截止立即解除。
- 强制截止解除后，HZ_REL bit变为0。

强制截止的详细时序请参考图10-75。

图10-84：强制截止解除的详细时序图



----- 表示截止状态（根据寄存器的设定值固定为H, L或Hi-Z）

(2) 软件解除时(ACT=1)

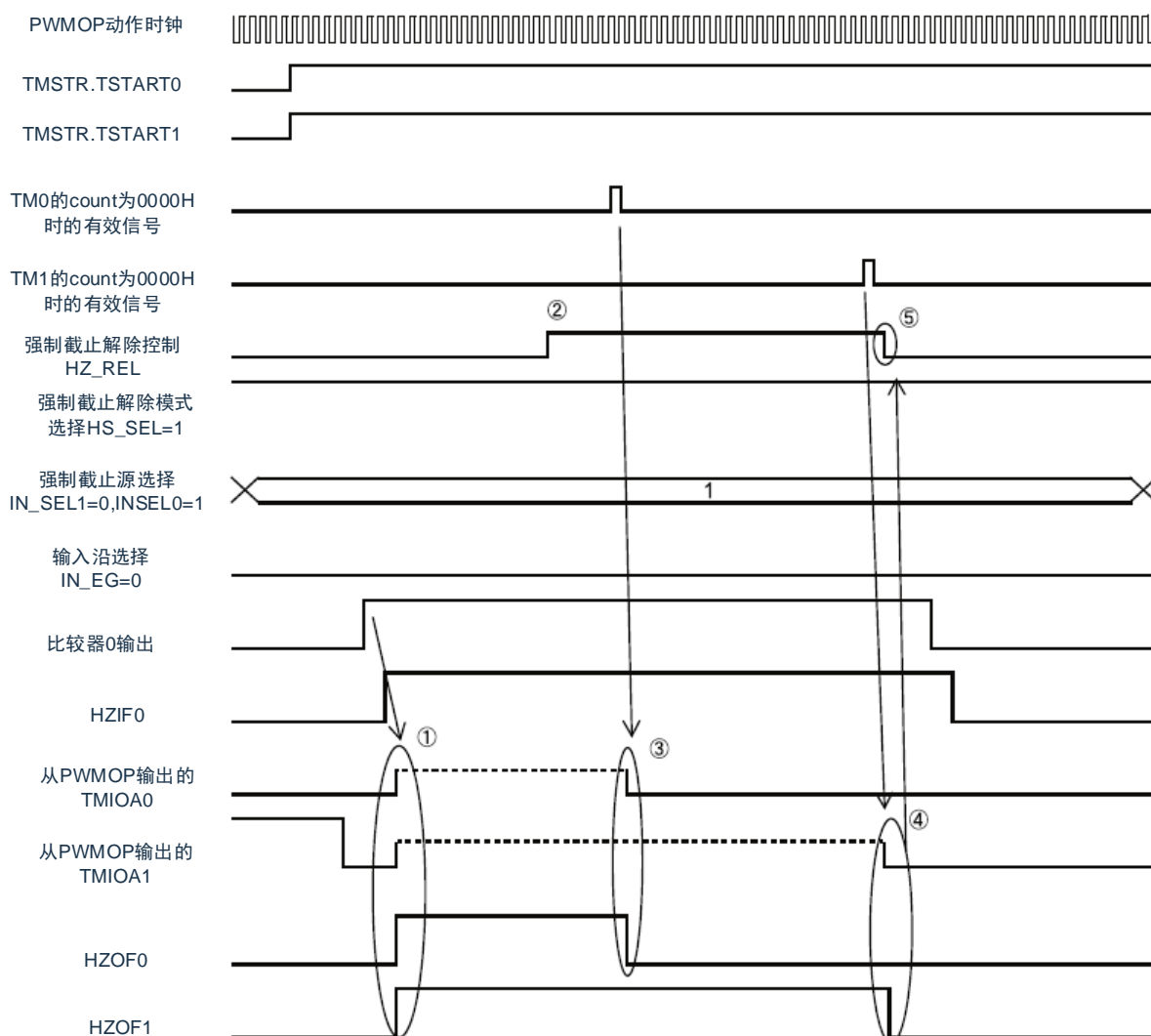
如果ACT设置为1，在设置OPCTL0的HZ_REL为1后，可以通过定时器M的信号解除强制截止状态。解除后HZ_REL被自动置0。

硬件解除截止时，在解除事件被检测到后，通过定时器M的信号来触发解除并且恢复输出。软件解除时，设置HZ_REL为1后通过定时器M的信号来触发解除并且恢复输出。解除时序是相同的。

a) 定时器M工作在输出比较功能，PWM功能，PWM3模式的场合：

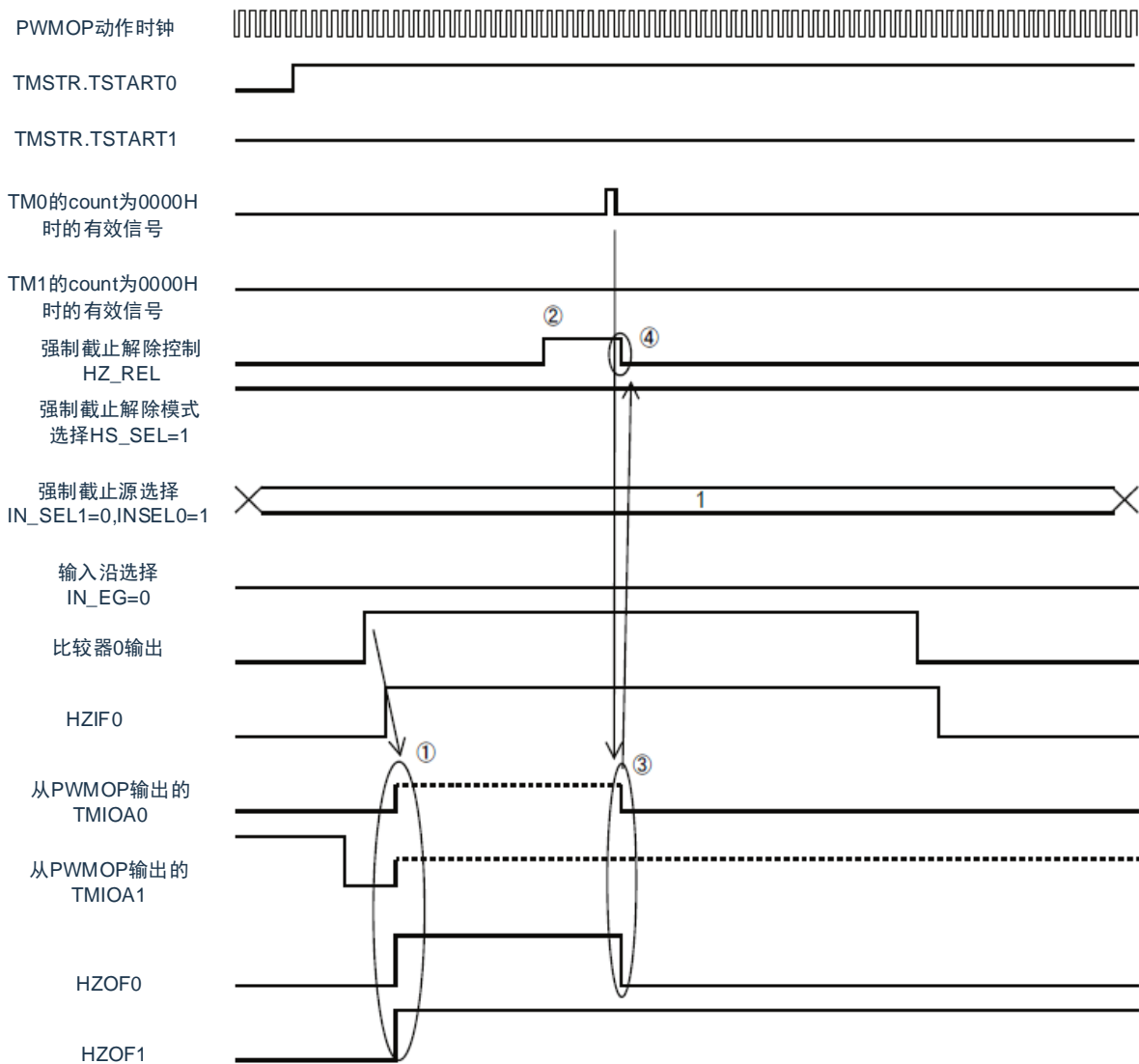
在HZ_REL置为1后，在TM0的计数值变为0000H时，TMIOA0, TMIOB0, TMIOC0, TMIOD0的输出强制截止状态被解除。在TM1的计数值变为0000H时，TMIOA1, TMIOB1, TMIOC1, TMIOD1的输出强制截止状态被解除。

图10-85: 软件解除强制截止的例子(定时器M, 2通道计数)



- 1) 当比较器0输出信号的上升沿被检测到时, TMIOA0, TMIOA1引脚的输出处于强制截止状态。
- 2) 设置HZ_REL为1, 等待各count计数值变为0000H。
- 3) TM0的计数值到达0000H时, TMIOA0的强制截止状态解除。
- 4) TM1的计数值到达0000H时, TMIOA1的强制截止状态解除。
- 5) 强制截止解除后, HZ_REL bit变为0。

图10-86：软件解除强制截止的例子(定时器M，1通道计数)



- a) 当比较器0输出信号的上升沿被检测到时，TMIOA0, TMIOA1引脚的输出处于强制截止状态。
- b) 设置HZ_REL为1，等待各count计数值变为0000H。
- c) TMO的计数值到达0000H时，TMIOA0的强制截止状态解除。
- d) 强制截止解除后，HZ_REL bit变为0。

强制截止的详细时序请参考图10-75。

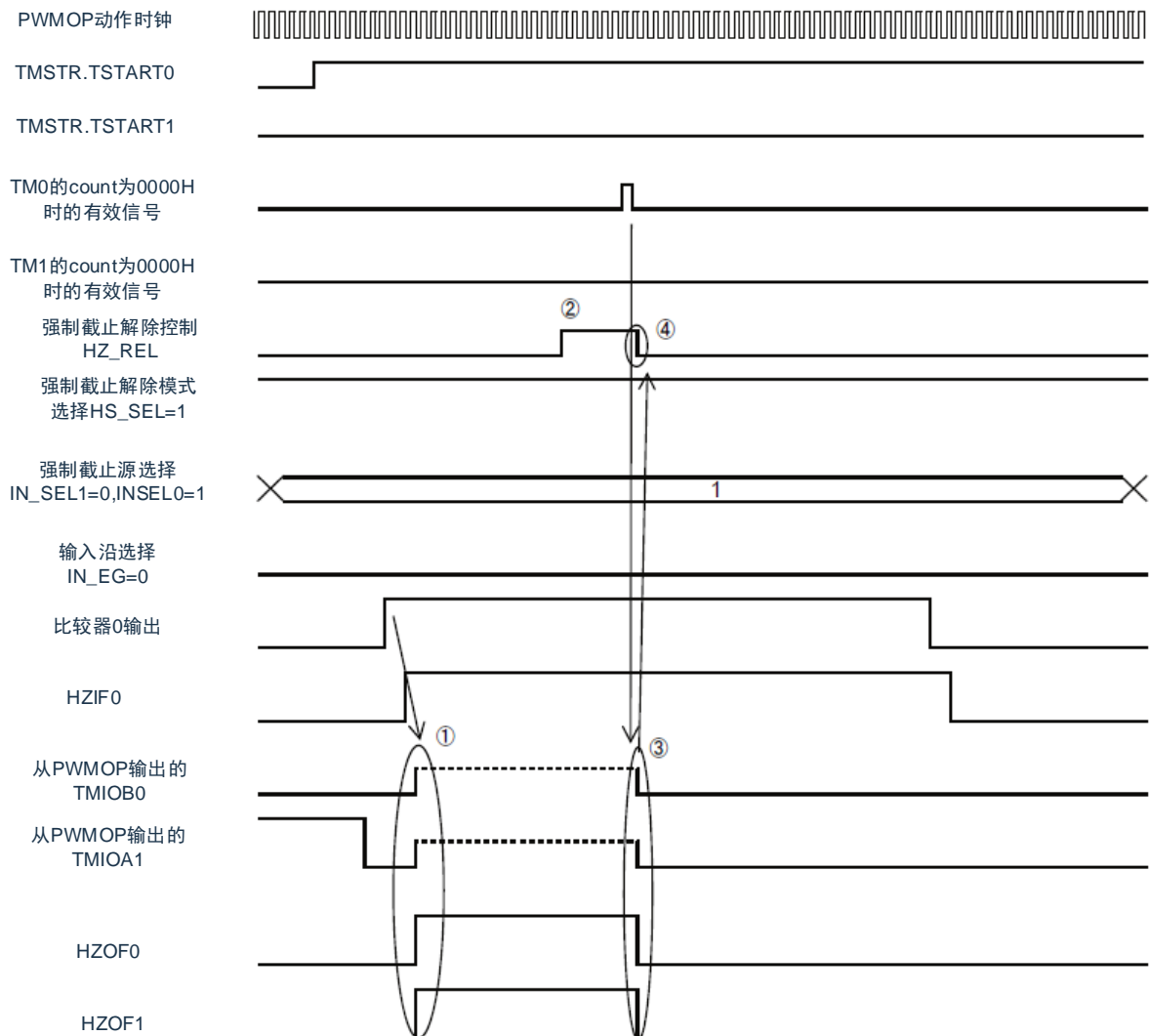
强制截止解除的详细时序请参考图10-76，图10-77。

HZ_REL自动清零的时序请参考图10-84。

(1) 定时器M工作在复位同步PWM模式的场合

设置HZ_REL为1，在TM0计数值为0000H时，全部TMIO引脚的输出强制截止都被解除。

图10-87：软件解除强制截止的例子



- a) 当比较器0输出信号的上升沿被检测到时，TMIOA0, TMIOA1引脚的输出处于强制截止状态。
- b) 设置HZ_REL为1，等待定时器M的通道0count计数值变为0000H。
- c) TM0的计数值到达0000H时，TMIOB0和TMIOA1的强制截止状态解除(通道1的动作不受影响)。
- d) 强制截止解除后，HZ_REL bit变为0。

强制截止的详细时序请参考图10-75。

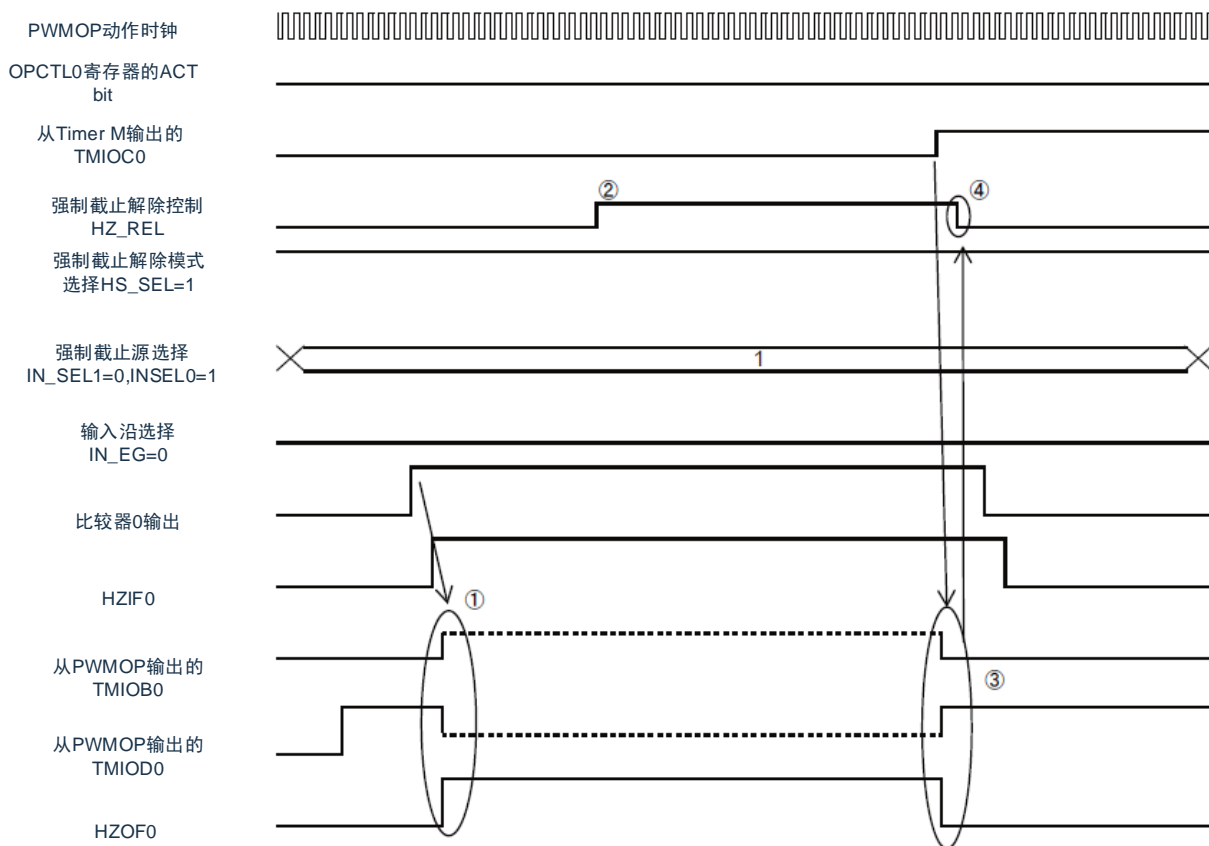
强制截止解除的详细时序请参考图10-76，图10-77。

HZ_REL自动清零的时序请参考图10-84。

(2) 定时器工作在互补PWM模式的场合

HZ_REL设置为1后，根据OPEDGE选择的TMIOC0的沿解除强制截止。

图10-88：软件解除强制截止的例子(以TMIOB0, TMIOD0为例)



- a) 当比较器0输出信号的上升沿被检测到时，TMIOB0, TMIOD0引脚的输出处于强制截止状态。
- b) 设置HZ_REL为1，等待TMIOC0的上升沿。
- c) TMIOC0的上升沿检出后，强制截止状态解除。
- d) 强制截止解除后，HZ_REL bit变为0。

强制截止的详细时序请参考图10-75。

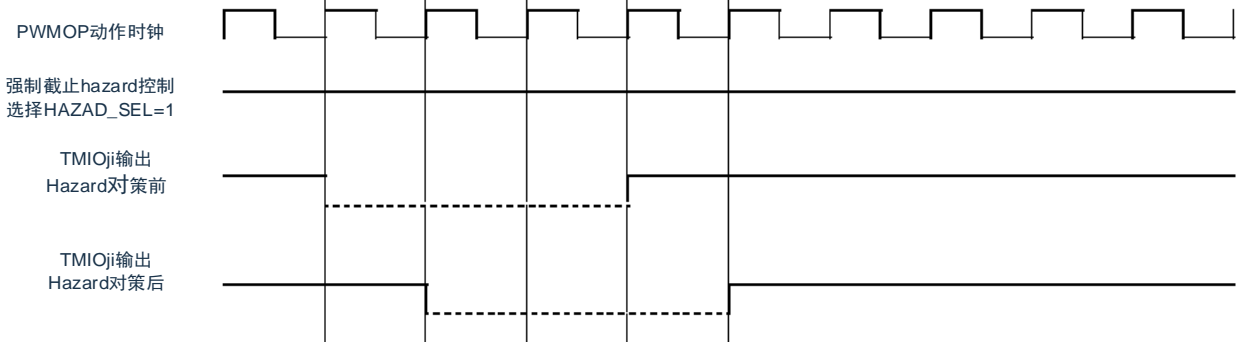
强制截止解除的详细时序请参考图10-79，图10-80，图10-81，图10-82。

HZ_REL自动清零的时序请参考图10-84。

10.8.3.4 Hazard对策

如果在强制截止状态/强制截止解除状态/定时器M动作中，TMIO引脚在复用功能和PORT功能之间切换时，存在误动作的风险。可以设置HAZAD_SET为1允许hazard对策来避免该风险。注意，使用Hazard对策时，定时器M的输出比禁止Hazard对策时延迟一个clock cycle。

图10-89: Hazard对策时序图



..... 表示截止状态（根据寄存器的设定值固定为H, L或Hi-Z）

备注：j=A,B,C,D；i=0,1

10.8.3.5 输出强制截止源检出状态和未检出状态

输出强制截止源是否检出由截止源选择bit(OPCTL0.IN_SEL1,OPCTL0.IN_SEL0)所选择信号(INTP0,CMP0)的电平决定。

如果设置OPCTL0.IN_EG为0，高电平为源检出状态，低电平为未检出状态。

如果设置OPCTL0.IN_EG为1，低电平为源检出状态，高电平为未检出状态。

备注：在通过设置IN_SEL1,IN_SEL0来选择INTP0输入,比较器0输出作为强制截止要因前，如果输出强制截止源超出了阈值范围，在IN_SEL1和IN_SEL0设置后，HZIF0 bit会被设置为1，而HZOF0和HZOF1不会被置起。

10.8.3.6 定时器M的计数器的值到达0000H时的时序图

硬件解除输出强制截止时，强制截止条件根据定时器M的动作模式不同而有差别。

- (1) 定时器M工作在输出比较功能时，计数值到达0000H的时序
 - 计数值=0000H，定时器M计数开始：不能解除强制截止
 - 定时器M在计数中，使用软件给计数器赋值0000H，解除强制截止状态
 - 定时器发生上溢，计数值变为0000H，解除强制截止状态
 - 与TMGRA0的值比较一致，计数值变为0000H，解除强制截止状态

- (2) 定时器M工作在PWM功能时，计数值到达0000H的时序
 - 计数值=0000H，定时器M计数开始：不能解除强制截止
 - 定时器M在计数中，使用软件给计数器赋值0000H，解除强制截止状态
 - 与TMGRA0的值比较一致，计数值变为0000H，解除强制截止状态

- (3) 定时器M工作在复位同步PWM模式时，计数值到达0000H的时序
 - 计数值=0000H，定时器M计数开始：不能解除强制截止
 - 定时器M在计数中，使用软件给计数器赋值0000H，解除强制截止状态
 - 与TMGRA0的值比较一致，计数值变为0000H，解除强制截止状态

- (4) 定时器M工作在PWM3模式时，计数值到达0000H的时序
 - 计数值=0000H，定时器M计数开始：不能解除强制截止
 - 定时器M在计数中，使用软件给计数器赋值0000H，解除强制截止状态
 - 与TMGRA0的值比较一致，计数值变为0000H，解除强制截止状态

- (5) 定时器M计数值到达0000H，并且定时器M停止工作时
 如果定时器M在计数值到达0000H的同时停止工作，不能解除强制截止。

图10-90：计数值=0000H时的判定时序
 (计数源= $F_{CLK}/2$ ，定时器M计数值到达0000H的同时计数器停止)

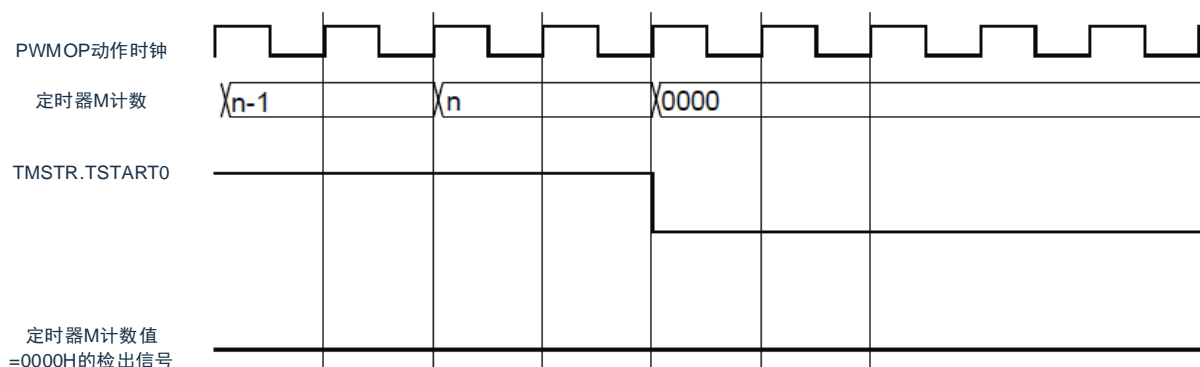
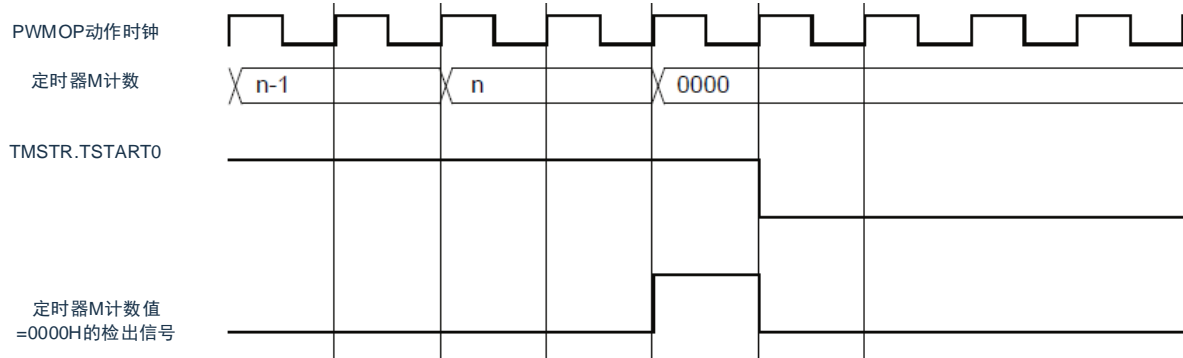


图10-91：计数值=0000H时的判定时序

(计数源= $F_{CLK}/2$ ，定时器M计数值到达0000H的下一个周期计数器停止)



10.8.3.7 设置步骤

PWMOP可以与定时器M联动，可以将PWMOP的设置追加在定时器M的设置中。步骤如下：

在Timer M的时钟和模式设置完后

- (1) 设置PWMOPEN为1
- (2) 设置OPCTL0寄存器
- (3) 设置OPEDGE寄存器
- (4) 设置OPDF0,OPDF1寄存器，定时器M动作开始
- (5) 等待HZOF1, HZOF0所示的强制截止状态
- (6) 强制截止解除

备注：PWMOP实现的功能是使用比较器0输出，外部中断输入INTP0和联动控制器对定时器M的输出进行强制截止。因此，PWMOP必须在定时器M动作的时候才能使用。如果只单独使用定时器M，PWMOP的相关寄存器需保持复位状态。

10.8.4 注意事项

(1) 定时器M的脉冲输出强制截止与PWMOP的输出强制截止同时工作时，优先级如下：

表10-22：强制截止时的优先级

		PWMOP 强制截止时的引脚状态			
		禁止	Hi-Z	L 电平	H 电平
定时器 M 强制截止时的 引脚状态	禁止	禁止	Hi-Z	L 电平	H 电平
	Hi-Z	Hi-Z	Hi-Z	L 电平	H 电平
	L 电平	L 电平	Hi-Z	L 电平	H 电平
	H 电平	H 电平	Hi-Z	L 电平	H 电平

- (2) 在互补PWM模式下，如果PWMOP处于输出强制截止状态时，定时器M也进入了脉冲输出强制截止。
- (3) 的状态，可能会对PWMOP产生解除截止的沿。
- (4) 使用EVENTC作为截止源的时候，必须使用软件解除截止状态(HS_SEL=1)
- (5) 使用输出截止hazard对策时，经由PWMOP的定时器M的输出会延迟一个时钟周期。
- (6) 使用输出截止hazard对策时，经由PWMOP的定时器M的输出引脚，可以在定时器计数动作中切换。
- (7) 复用功能和PORT功能。
- (8) 选择比较器0输出和INTP0输入的有效宽度必须大于一个时钟周期。

第11章 实时时钟

11.1 实时时钟的功能

实时时钟有以下功能。

- 持有年、月、星期、日、小时、分钟和秒的计数器，最长能计数到99年。
- 固定周期中断功能(周期：0.5秒、1秒、1分钟、1小时、1日、1个月)
- 闹钟中断功能(闹钟：星期、小时、分钟)
- 1Hz的引脚输出功能

11.2 实时时钟的结构

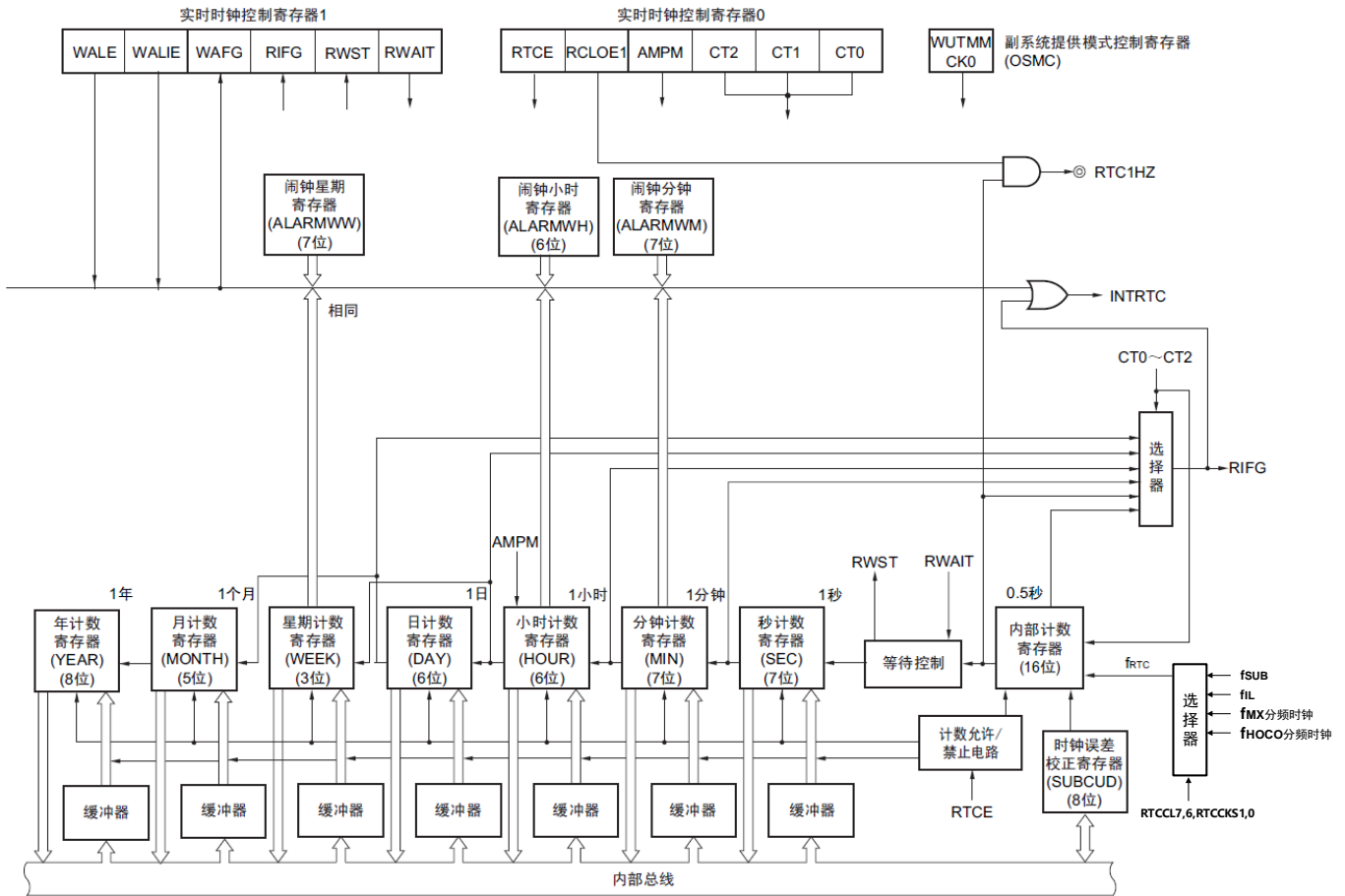
实时时钟由以下硬件构成。

表11-1：实时时钟的结构

项目	结构
计数器	内部计数器(16位)
控制寄存器	外围允许寄存器0(PER0.bit7)
	实时时钟选择寄存器(RTCCCL)
	实时时钟控制寄存器0(RTCC0)
	实时时钟控制寄存器1(RTCC1)
	秒计数寄存器(SEC)
	分钟计数寄存器(MIN)
	小时计数寄存器(HOUR)
	日计数寄存器(DAY)
	星期计数寄存器(WEEK)
	月计数寄存器(MONTH)
	年计数寄存器(YEAR)
	时钟误差校正寄存器(SUBCUD)
	闹钟分钟寄存器(ALARMWM)
	闹钟小时寄存器(ALARMWH)
闹钟星期寄存器(ALARMWW)	

备注：以上RTC控制寄存器的复位只受POR复位控制。

图11-1：实时时钟的框图



注意：只有在选择 F_{MX}/F_{HOCO} 的分频时钟(分周后 $\approx 32,768\text{KHZ}$)或者副系统时钟($F_{SUB}=32.768\text{KHZ}$)作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟($F_{IL}=15\text{KHz}$)时，只能使用固定周期中断功能。

选择 F_{IL} 时的固定周期中断间隔用以下计算式进行计算：固定周期(RTCC0寄存器选择的值) $\times F_{SUB}/F_{IL}$

11.3 控制实时时钟的寄存器

通过以下寄存器控制实时时钟。

- 外围允许寄存器0(PER0)
- 实时时钟选择寄存器(RTCCL)
- 实时时钟控制寄存器0(RTCC0)
- 实时时钟控制寄存器1(RTCC1)
- 秒计数寄存器(SEC)
- 分钟计数寄存器(MIN)
- 小时计数寄存器(HOUR)
- 日计数寄存器(DAY)
- 星期计数寄存器(WEEK)
- 月计数寄存器(MONTH)
- 年计数寄存器(YEAR)
- 时钟误差校正寄存器(SUBCUD)
- 闹钟分钟寄存器(ALARMWM)
- 闹钟小时寄存器(ALARMWH)
- 闹钟星期寄存器(ALARMWW)
- 端口模式寄存器(PMx)
- 端口寄存器(Px)

11.3.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用实时时钟时，必须将bit7(RTCEN)置“1”。通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-2: 外围允许寄存器0(PER0)的格式

地址: 0x40020420	复位后: 00H							R/W
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN
	RTCEN	提供实时时钟(RTC)和15位间隔定时器的输入时钟的控制						
	0	停止提供输入时钟。 • 不能写实时时钟(RTC)和15位间隔定时器使用的SFR。 • 实时时钟(RTC)和15位间隔定时器处于复位状态。						
	1	提供输入时钟。 • 能读写实时时钟(RTC)和15位间隔定时器使用的SFR。						

注意:

- 如果要使用实时时钟，必须先将在计数时钟(F_{RTC})振荡稳定的状态下将RTCEN位置“1”，然后设置以下的寄存器。当RTCEN位为“0”时，忽视实时时钟控制寄存器的写操作，而且读取值为初始值(实时时钟选择寄存器(RTCCL)、端口模式寄存器和端口寄存器除外)。
 - 实时时钟控制寄存器0(RTCC0)
 - 实时时钟控制寄存器1(RTCC1)
 - 秒计数寄存器(SEC)
 - 分钟计数寄存器(MIN)
 - 小时计数寄存器(HOUR)
 - 日计数寄存器(DAY)
 - 星期计数寄存器(WEEK)
 - 月计数寄存器(MONTH)
 - 年计数寄存器(YEAR)
 - 时钟误差校正寄存器(SUBCUD)
 - 闹钟分钟寄存器(ALARMWM)
 - 闹钟小时寄存器(ALARMWH)
 - 闹钟星期寄存器(ALARMWW)
- 能通过将副系统时钟提供模式控制寄存器(OSMC)的RTCLPC位置“1”，在深度睡眠模式或者以副系统时钟运行的睡眠模式中停止给实时时钟和15位间隔定时器以外的外围功能提供副系统时钟。

11.3.2 实时时钟选择寄存器(RTCCL)

能通过RTCCL选择实时时钟和15位间隔定时器的计数时钟(F_{RTC})。

图11-3: 实时时钟选择寄存器(RTCCL)的格式

地址: 0x4004047C 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	0	0	0	0	RTCKS1	RTCKS0

RTCCL7	实时时钟、15位间隔定时器的计数时钟的时钟源的选择
0	选择高速系统时钟(F_{MX})
1	选择高速内部振荡器(F_{HOCO})

RTCKS1	RTCKS0	RTCCL6	实时时钟、15位间隔定时器的计数时钟的运行时钟的选择
0	0	0/1	副系统时钟(F_{SUB})
0	1		低速内部振荡器时钟(F_{IL})(必须设置WUTMMCK0为1)
1	0	0	主时钟 F_{MX}/F_{HOCO} (通过RTCCL7选择)/1464
1	0	1	主时钟 F_{MX}/F_{HOCO} (通过RTCCL7选择)/976
1	1	0	主时钟 F_{MX}/F_{HOCO} (通过RTCCL7选择)/488
1	1	1	主时钟 F_{MX}/F_{HOCO} (通过RTCCL7选择)/244

11.3.3 实时时钟控制寄存器0(RTCC0)

这是设置实时时钟的运行开始或者停止、RTC1HZ引脚的控制、12/24小时系统和固定周期中断功能的8位寄存器。

通过8位存储器操作指令设置RTCC0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-4：实时时钟控制寄存器0(RTCC0)的格式

地址：0x40044F5D复位后：00H R/W

符号	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1 ^注	0	AMPM	CT2	CT1	CT0

RTCE	实时时钟的运行控制
0	停止计数器的运行。
1	开始计数器的运行。

RCLOE1 ^注	RTC1HZ引脚的输出控制
0	禁止RTC1HZ引脚的输出(1Hz)。
1	允许RTC1HZ引脚的输出(1Hz)。

AMPM	12小时系统/24小时系统的选择
0	12小时系统(表示上午或者下午)
1	24小时系统
<ul style="list-style-type: none"> 要更改AMPM位的值时，必须在将RWAIT位(实时时钟控制寄存器1(RTCC1)的bit0)置“1”后进行改写。如果更改AMPM位的值，小时计数寄存器(HOUR)的值就变为所设时间系统的对应值。 时间位的表示如表11-2所示。 	

CT2	CT1	CT0	固定周期中断(INTRTC)的选择
0	0	0	不使用固定周期中断功能。
0	0	1	0.5秒一次(与秒累加同步)
0	1	0	1秒一次(与秒累加同时)
0	1	1	1分钟一次(每分钟的00秒)
1	0	0	1小时一次(每小时的00分00秒)
1	0	1	1日一次(每日的00点00分00秒)
1	1	×	1个月一次(每月的1日上午00点00分00秒)
要在计数器运行中(RTCE=1)更改CT2~CT0位的值时，必须在通过中断屏蔽标志寄存器将INTRTC设置为禁止中断处理后进行改写，并且必须在改写后清除RIFG标志和RTCIF标志，然后再设置为允许中断处理。			

注：在32引脚的产品时，必须将RCLOE1位置“0”。

注意：

- 在RTCE位为“1”时，不能更改RCLOE1位。
- 在RTCE位为“0”时，即使将RCLOE1位置“1”也不输出1Hz。

备注：×：忽略

11.3.4 实时时钟控制寄存器1(RTCC1)

这是控制闹钟中断功能和计数器等待的8位寄存器。通过8位存储器操作指令设置RTCC1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-5: 实时时钟控制寄存器1(RTCC1)的格式(1/2)

地址: 0x40044F5E

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	闹钟的运行控制
0	一致运行无效。
1	一致运行有效。

要在计数器运行中(RTCE=1)并且WALIE位为“1”的情况下设置WALE位时，必须通过中断屏蔽标志寄存器将INTRTC设置为禁止中断处理后进行改写，并且必须在改写后清除WAFG标志和RTCIF标志。要设置各闹钟寄存器(RTCC1寄存器的WALIE标志、闹钟分钟寄存器(ALARMWM)、闹钟小时寄存器(ALARMWH)和闹钟星期寄存器(ALARMWW))时，必须将WALE位置“0”(一致运行无效)。

WALIE	闹钟中断(INTRTC)功能的运行控制
0	不产生闹钟一致中断。
1	产生闹钟一致中断。

WAFG	闹钟检测状态标志
0	闹钟不一致。
1	检测到闹钟一致。

这是表示检测到闹钟一致的状态标志。只在WALE位为“1”时有效，在检测到闹钟一致并且经过1个F_{RTC}时钟后变为“1”。
通过给此标志写“0”来清除此标志。写“1”的操作无效。

图11-5: 实时时钟控制寄存器1(RTCC1)的格式(2/2)

RIFG	固定周期中断状态标志
0	没有产生固定周期中断。
1	产生固定周期中断。

这是表示产生固定周期中断的状态标志。当产生固定周期中断时，此标志为“1”。
通过给此标志写“0”来清除此标志。写“1”的操作无效。

RWST	实时时钟的等待状态标志
0	计数器正在运行。
1	正处于计数器的读写模式。

这是表示RWAIT位的设置是否有效的状态。
必须在确认此标志为“1”后读写计数值。

RWAIT	实时时钟的等待控制
0	设置为计数器运行。
1	设置为SEC~YEAR计数器停止运行，进入计数器的读写模式。

此位控制计数器的运行。要读写计数值时，必须给此位写“1”。
因为内部计数器(16位)继续运行，所以必须在1秒内结束读写，然后返回到“0”。
从将RWAIT位置“1”到能读写计数值(RWST=1)为止，最少需要1个F_{RTC}时钟的时间。
如果在RWAIT位为“1”时发生内部计数器(16位)上溢，就保持发生上溢的状态，在RWAIT位变为“0”后进行递增计数。
但是，当写秒计数寄存器时，不保持发生上溢的状态。

备注:

1. 固定周期中断和闹钟一致中断使用相同中断源(INTRTC)。在同时使用这2个中断的情况下，当发生INTRTC中断时，可以通过确认固定周期中断状态标志(RIFG)和闹钟检测状态标志(WAFG)来判断发生的是哪个中断。
2. 如果写秒计数寄存器(SEC)，就清除内部计数器(16位)。

11.3.6 秒计数寄存器(SEC)

这是用0~59(十进制)表示秒计数值的8位寄存器。通过内部计数器(16位)的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。必须以BCD码设置十进制的00~59。

通过8位存储器操作指令设置SEC寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-7: 秒计数寄存器(SEC)的格式

地址: 0x40044F52复位后: 00H		R/W						
符号	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

注意: 要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

备注: 如果写秒计数寄存器(SEC)，就清除内部计数器(16位)。

11.3.7 分钟计数寄存器(MIN)

这是用0~59(十进制)表示分钟计数值的8位寄存器。通过秒计数器的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。在写操作过程中忽视秒计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的00~59。

通过8位存储器操作指令设置MIN寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-8: 分钟计数寄存器(MIN)的格式

地址: 0x40044F53复位后: 00H		R/W						
符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

注意: 要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

11.3.8 小时计数寄存器(HOUR)

这是用00~23或者01~12、21~32(十进制)表示小时计数值的8位寄存器。通过分钟计数器的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多2个 F_{RTC} 时钟后被写到计数器。在写操作过程中忽视分钟计数寄存器的上溢并且设置为写入值。

必须根据实时时钟控制寄存器0(RTCC0)的bit3(AMPM)设置的时间系统，以BCD码设置十进制的00~23或者01~12、21~32。

如果更改AMPM位的值，HOUR寄存器的值就变为所设时间系统的对应值。通过8位存储器操作指令设置HOUR寄存器。在产生复位信号后，此寄存器的值变为“12H”。

但是，如果在复位后将AMPM位置“1”，此寄存器的值就变为“00H”。

图11-9: 小时计数寄存器(HOUR)的格式

地址: 0x40044F54复位后: 12H								R/W
符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意:

1. 当选择AMPM位为“0”(12小时系统)时，HOUR寄存器的bit5(HOUR20)表示AM(0)/PM(1)。
2. 要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

AMPM 位的设置值、小时计数寄存器(HOUR)的值和时间的关系如表11-2所示。

表11-2: 时间位的表示

24小时表示(AMPM=1)		12小时表示(AMPM=0)	
时间	HOUR 寄存器	时间	HOUR 寄存器
0 时	00H	AM12 时	12H
1 时	01H	AM1时	01H
2 时	02H	AM2时	02H
3 时	03H	AM3时	03H
4 时	04H	AM4时	04H
5 时	05H	AM5 时	05H
6 时	06H	AM6 时	06H
7 时	07H	AM7 时	07H
8 时	08H	AM8 时	08H
9 时	09H	AM9 时	09H
10 时	10H	AM10 时	10H
11 时	11H	AM11 时	11H
12 时	12H	PM12 时	32H
13 时	13H	PM1 时	21H
14 时	14H	PM2 时	22H
15 时	15H	PM3 时	23H
16 时	16H	PM4 时	24H
17 时	17H	PM5 时	25H
18 时	18H	PM6 时	26H
19 时	19H	PM7 时	27H
20 时	20H	PM8 时	28H
21 时	21H	PM9 时	29H
22 时	22H	PM10 时	30H
23 时	23H	PM11 时	31H

当AMPM位为“0”时，HOUR寄存器的值为12小时表示；当AMPM位为“1”时，HOUR寄存器的值为24小时表示。

在12小时表示时，HOUR寄存器的bit5表示上午/下午。上午(AM)为“0”，下午(PM)为“1”。

11.3.9 日计数寄存器(DAY)

这是用1~31(十进制)表示日计数值的8位寄存器。通过小时计数器的上溢进行递增计数。计数器进行以下的计数。

- 01~31(1、3、5、7、8、10、12月)
- 01~30(4、6、9、11月)
- 01~29(2月闰年)
- 01~28(2月平常年)

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。在写操作过程中忽视小时计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的01~31。

通过8位存储器操作指令设置DAY寄存器。在产生复位信号后，此寄存器的值变为“01H”。

图11-10: 日计数寄存器(DAY)的格式

地址: 0x40044F56H 复位后: 01H RW

符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

注意: 要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

11.3.10 星期计数寄存器(WEEK)

这是用0~6(十进制)表示星期计数值的8位寄存器。与日计数器同步进行递增计数。

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。必须以BCD码设置十进制的00~06。

通过8位存储器操作指令设置WEEK寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-11：星期计数寄存器(WEEK)的格式

地址：0x40044F55H	复位后：00H R/W							
符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

注意：

1. 月计数寄存器(MONTH)和日计数寄存器(DAY)的对应值不自动保存到星期寄存器(WEEK)。必须在解除复位后进行以下的设置：

星期	WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

2. 要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

11.3.11 月计数寄存器(MONTH)

这是用1~12(十进制)表示月计数值的8位寄存器。通过日计数器的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。在写操作过程中忽视日计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的01~12。

通过8位存储器操作指令设置MONTH寄存器。在产生复位信号后，此寄存器的值变为“01H”。

图11-12：月计数寄存器(MONTH)的格式

地址：0x40044F57H	复位后：01H R/W							
符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

注意：要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

11.3.12 年计数寄存器(YEAR)

这是用0~99(十进制)表示年计数值的8位寄存器。通过月计数器(MONTH)的上溢进行递增计数。00、04、08、……、92、96是闰年。

在写时，数据先被写到缓冲器，在经过最多2个F_{RTC}时钟后被写到计数器。在写操作过程中忽视MONTH寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的00~99。通过8位存储器操作指令设置YEAR寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-13：年计数寄存器(YEAR)的格式

地址：0x40044F58H	复位后：00H R/W							
符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

注意：要在计数器运行中(RTCE=1)读写此寄存器时，必须按照“11.4.3 实时时钟计数器的读写”记载的步骤进行。

11.3.13 闹钟分钟寄存器(ALARMWM)

这是设置闹钟分钟的寄存器。

通过8位存储器操作指令设置ALARMWM寄存器。在产生复位信号后，此寄存器的值变为“00H”。

注意：必须以BCD码设置十进制的00~59。如果设置范围以外的值，就不检测闹钟。

图11-14：闹钟分钟寄存器(ALARMWM)的格式

地址：0x40044F5AH

复位后：00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

11.3.14 闹钟小时寄存器(ALARMWH)

这是设置闹钟小时的寄存器。

通过8位存储器操作指令设置ALARMWH寄存器。在产生复位信号后，此寄存器的值变为“12H”。

但是，如果在复位后将AMPM位置“1”，此寄存器的值就变为“00H”。

注意：必须以BCD码设置十进制的00~23或者01~12、21~32。如果设置范围以外的值，就不检测闹钟。

图11-15：闹钟小时寄存器(ALARMWH)的格式

地址：0x40044F5BH

复位后：12H R/W

符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

注意：当选择AMPM位为“0”(12小时系统)时，ALARMWH寄存器的bit5(WH20)表示AM(0)/PM(1)。

11.3.15 闹钟星期寄存器(ALARMWW)

这是设置闹钟星期的寄存器。

通过8位存储器操作指令设置ALARMWW寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-16: 闹钟星期寄存器(ALARMWW)的格式

地址: 0x40044F5CH

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

闹钟时间的设置例子如下所示。

闹钟设置时间	星期							12小时表示				24小时表示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	10时	1时	10分	1分	10时	1时	10分	1分
	W	W	W	W	W	W	W	10时	1时	10分	1分	10时	1时	10分	1分
每天 上午0时00分	0	1	2	3	4	5	6	1	2	0	0	0	0	0	0
每天 上午1时30分	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 上午11时59分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一~星期五 下午0时00分	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 下午1时30分	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 下午11时59分	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

11.3.16 端口模式寄存器和端口寄存器

在将RTC1HZ输出引脚的复用端口用1Hz输出时，必须将各端口对应的端口模式控制寄存器(PMCxx)的位、端口模式寄存器(PMxx)的位和端口寄存器(Pxx)的位置“0”。

设置的端口模式寄存器(PMxx)、端口寄存器(Pxx)和端口模式控制寄存器(PMCxx)因产品而不同。详细内容请参照“2.5 使用复用功能时的寄存器设置”。

11.4 实时时钟的运行

11.4.1 实时时钟的运行开始

图11-17: 实时时钟的运行开始步骤



注1: 必须先在计数时钟(F_{RTC})振荡稳定的状态下将RTCEN位置“1”。

注2: 这只是需要校正时钟误差的情况。有关校正值的计算方法, 请参照“11.4.6实时时钟的时钟误差校正例子”。

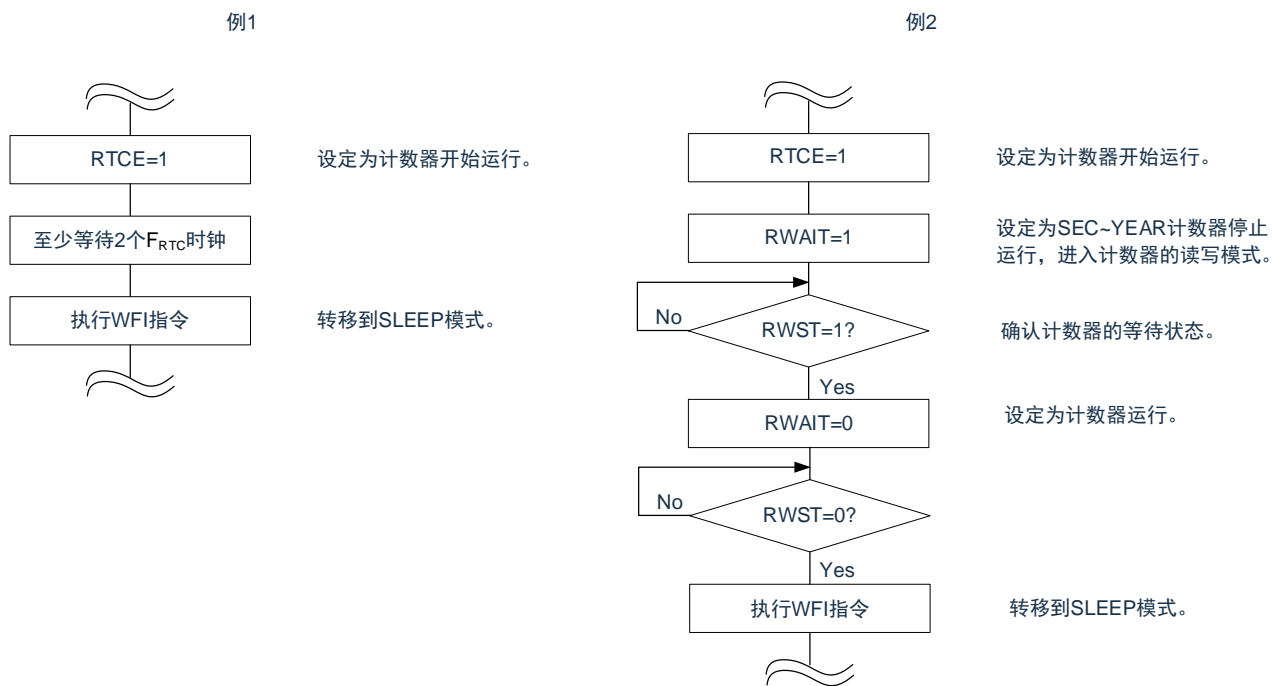
注3: 在RTCE位为“1”后不等待INTRTC位变为“1”而转移到睡眠模式的情况下, 请确认“11.4.2开始运行后睡眠模式的转移”的步骤。

11.4.2 开始运行后睡眠模式的转移

要在将RTCE位置“1”后立即转移到睡眠(包括深度睡眠)模式时,必须进行以下某种处理。但是,在将RTCE位置“1”后,如果要在发生INTRTC中断后转移到睡眠模式,就不需要这些处理。

- 在将RTCE位置“1”后至少经过2个计数时钟(F_{RTC})的时间之后转移到睡眠模式(参照图11-18的例1)。
- 在将RTCE位置“1”后将RWAIT位置“1”,通过轮询确认RWST位变为“1”。然后,将RWAIT位置“0”并且再次通过轮询确认RWST位变为“0”,然后转移到睡眠模式(参照图11-18的例2)。

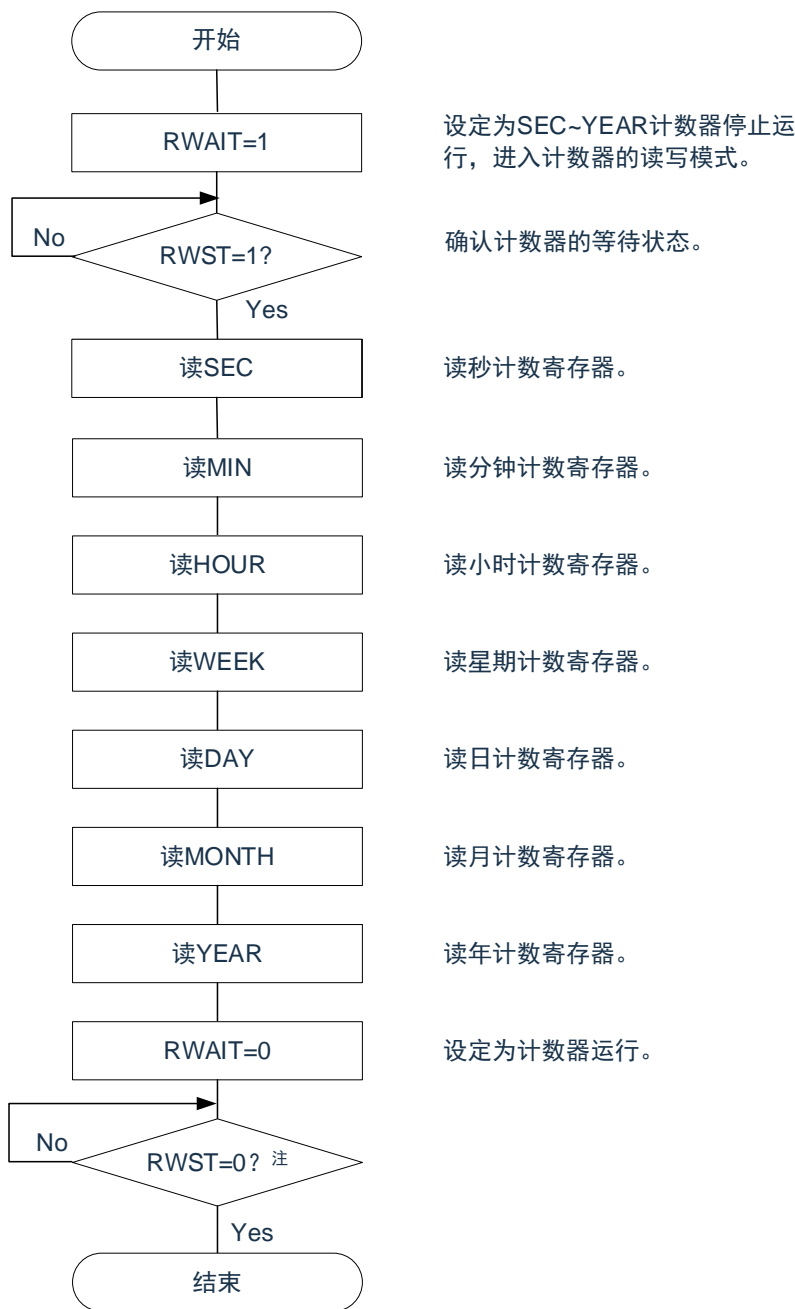
图11-18: 将RTCE位置“1”后的睡眠/深度睡眠模式的转移步骤



11.4.3 实时时钟计数器的读写

必须先将RWAIT位置“1”，然后读写计数器。必须在读写计数器后将RWAIT位置“0”。

图11-19: 实时时钟计数器的读操作步骤

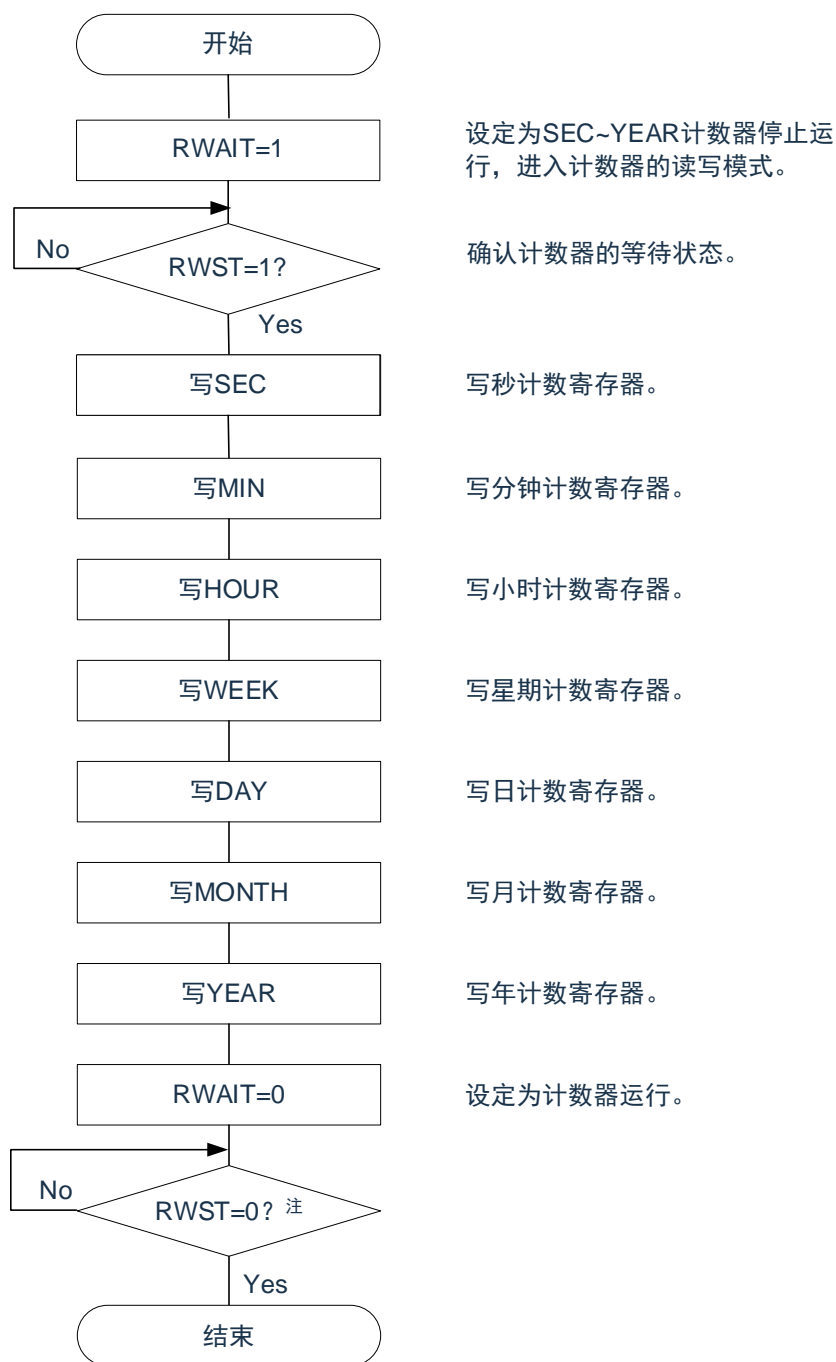


注：必须在转移到睡眠模式前，确认RWST位为“0”。

注意：必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。

备注：不限制秒/分钟/小时/星期/日/月/和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。

图11-20: 实时时钟计数器的读操作步骤



注：必须在转移到SLEEP模式前，确认RWST位为“0”。

注意：

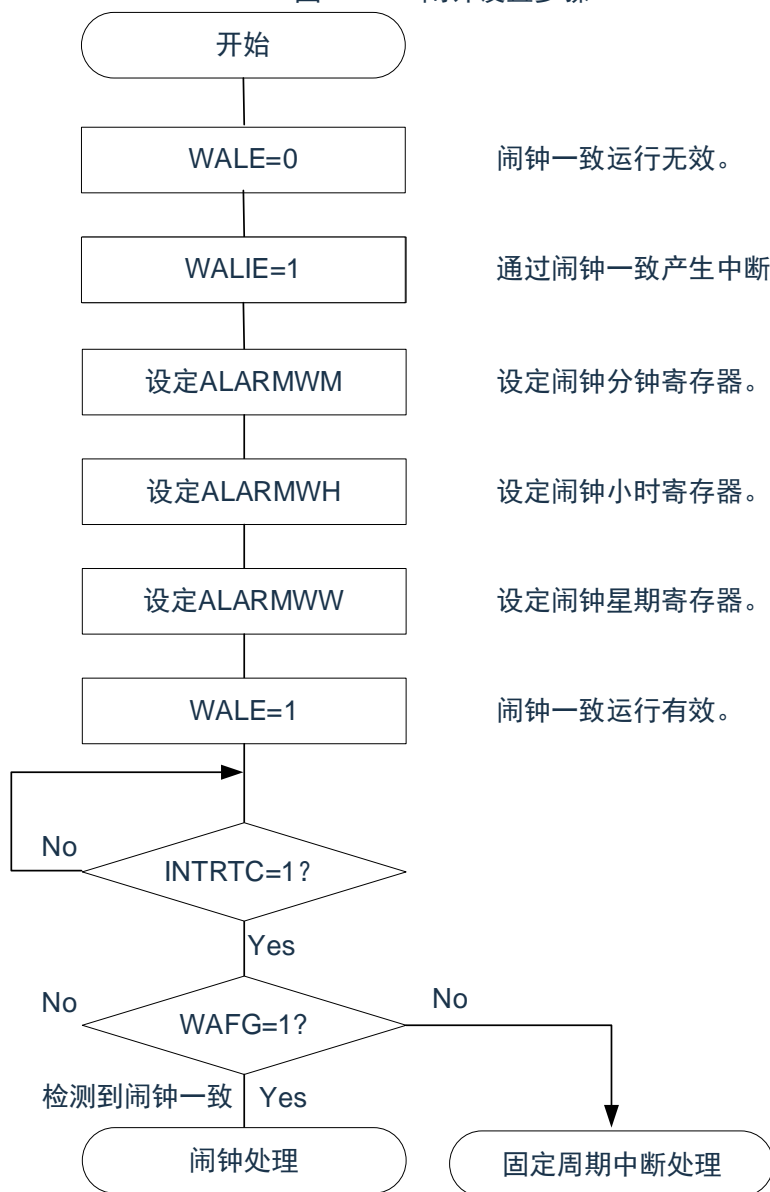
1. 必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。
2. 要在计数器运行中(RTCE=1)改写SEC、MIN、HOUR、WEEK、DAY、MONTH、YEAR寄存器时，必须在通过中断屏蔽标志寄存器将INTRTC设置为禁止中断处理后进行改写，并且必须在改写后清除WAFG标志、RIFG标志和RTCIF标志。

备注：不限制秒/分钟/小时/星期/日/月/和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。

11.4.4 实时时钟的闹钟设置

必须先将WALE位置“0”(闹钟运行无效), 然后设置闹钟时间。

图11-21: 闹钟设置步骤



备注:

1. 不限制闹钟分钟寄存器(ALARMW)、闹钟小时寄存器(ALARMWH)和闹钟星期寄存器(ALARMWW)的写操作顺序。
2. 固定周期中断和闹钟一致中断使用相同中断源(INTRTC)。在同时使用这2个中断的情况下, 能在发生INTRTC时通过确认固定周期中断状态标志(RIFG)和闹钟检测状态标志(WAFG)来判断发生的是哪个中断。

11.4.5 实时时钟的1Hz输出

图11-22: 1Hz输出的设置步骤



注意:

1. 必须先在计数时钟(F_{SUB})振荡稳定的状态下将RTCEN 位置“1”。
2. 部分封装不支持实时时钟的1Hz输出功能。

11.4.6 实时时钟的时钟误差校正例子

能通过给时钟误差校正寄存器设置值进行高精度的时钟快慢校正。

校正值的计算方法的例子

校正内部计数器(16位)的计数值时的校正值能用以下计算式进行计算。当校正范围超出-4165.6ppm~4165.6ppm的范围时，必须将DEV位置“0”。

(DEV=0的情况)

校正值^注=1分钟的校正计数值÷3=(振荡频率÷目标频率-1)×32768×60÷3

(DEV=1的情况)

校正值^注=1分钟的校正计数值=(振荡频率÷目标频率-1)×32768×60

注：校正值是根据时钟误差校正寄存器(SUBCUD)的bit12~0的值计算的时钟误差校正值。

(F12=0的情况)校正值={ (F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) - 1 } × 2

(F12=1的情况)校正值= -{ (/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1 } × 2

当(F12~F0)=(*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *)时，不进行时钟误差的校正。*是“0”或者“1”。

/F12~/F0是各位取反后的值(“000000000011”时，为“111111111100”)。

备注：

1. 校正值为2、4、6、8、……、8186、8188或者-2、-4、-6、-8、……、-8186、-8188。
2. 振荡频率是计数时钟(F_{RTC})的值，能用以下计算式进行计算：
时钟误差校正寄存器为初始值(“00H”)时的RTC1HZ引脚的输出频率0×32768
3. 目标频率是使用时钟误差校正寄存器进行校正后的频率。

校正例子

从32767.4Hz校正到32768Hz(32767.4Hz+18.3ppm)的例子

【振荡频率的测量】

在时钟误差校正寄存器(SUBCUD)为初始值("0000H")时, 通过从RTC1HZ引脚输出大约1Hz的信号来测量各产品的振荡频率^注。

注: 有关RTC1Hz输出的设置步骤, 请参照“10.4.5实时时钟的1Hz输出”。

【校正值的计算】

(RTC1HZ引脚的输出频率为0.9999817Hz的情况)

振荡频率=32768×0.9999817≈32767.4Hz

假设目标频率为32768Hz(32767.4Hz+18.3ppm)并且DEV=1。

适用DEV位为“1”时的校正值的计算式。

$$\begin{aligned}
 \text{校正值} &= 1 \text{ 分钟的校正计数值} = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \\
 &= (32767.4 \div 32768 - 1) \times 32768 \times 60 \\
 &= -36
 \end{aligned}$$

【(F12~F0)的设置值的计算】

(校正值= -36的情况)

因为校正值小于0(变快的情况), 所以F12=1。根据校正值计算(F11~F0)。

$$-(/F11 \sim /F0) - 1 \times 2 = -36$$

$$(/F11 \sim /F0) = 17$$

$$(/F11 \sim /F0) = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)$$

$$(F11 \sim F0) = (1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)$$

因此, 从32767.4Hz校正到32768Hz(32767.4Hz+18.3ppm)的情况如下:

如果通过DEV=1和校正值= -36(SUBCUD寄存器的bit12~0: 1,1,1,1,1,1,1,1,0,1,1,1,0)来设置校正寄存器, 就能校正到32768Hz(0ppm)。

第12章 15位间隔定时器

12.1 15位间隔定时器的功能

以事先设置的任意时间间隔产生中断(INTIT)，能用于从深度睡眠模式的唤醒。

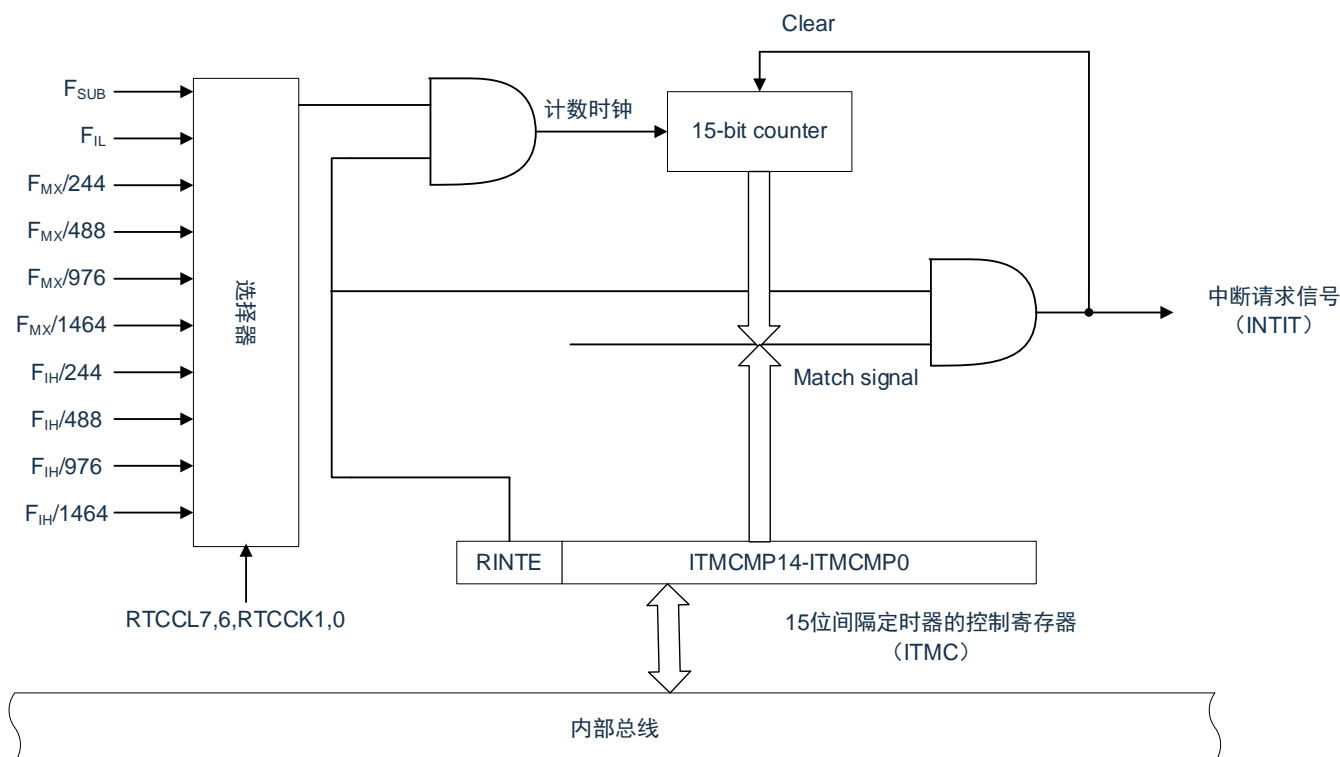
12.2 15位间隔定时器的结构

15位间隔定时器由以下硬件构成。

表12-1: 15位间隔定时器的结构

项目	结构
计数器	15位计数器
控制寄存器	外围允许寄存器0(PER0)
	实时时钟选择寄存器(RTCCCL)
	15位间隔定时器的控制寄存器(ITMC)

图12-1: 15位间隔定时器的框图



12.3 控制15位间隔定时器的寄存器

通过以下寄存器控制15位间隔定时器。

- 外围允许寄存器0(PER0)
- 实时时钟选择寄存器(RTCCCL)
- 15位间隔定时器的控制寄存器(ITMC)

12.3.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用15位间隔定时器时，必须将bit7(RTCEN)置“1”。通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图12-2: 外围允许寄存器0(PER0)的格式

地址: 0x40020420	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN

RTCEN	提供实时时钟(RTC)和15位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写实时时钟(RTC)和15位间隔定时器使用的SFR。 • 实时时钟(RTC)和15位间隔定时器处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> • 能读写实时时钟(RTC)和15位间隔定时器使用的SFR。

12.3.2 实时时钟选择寄存器(RTCCL)

能通过RTCCL选择实时时钟和15位间隔定时器的计数时钟(F_{RTC})。

图12-3: 实时时钟选择寄存器(RTCCL)的格式

地址: 0x4004047C 复位后: 00H R/W

RTCCL

RTCCL7	RTCCL6	0	0	0	0	RTCKS1	RTCKS0
--------	--------	---	---	---	---	--------	--------

RTCCL7	实时时钟、15位间隔定时器的计数时钟的时钟源的选择
0	选择高速系统时钟(F_{MX})
1	选择高速内部振荡器(F_{HOCO})

RTCKS1	RTCKS0	RTCCL6	实时时钟、15位间隔定时器的计数时钟的运行时钟的选择
0	0	0/1	副系统时钟(F_{SUB})
0	1		低速内部振荡器时钟(F_{IL})(必须设置WUTMMCK0为1)
1	0	0	主时钟 F_{MAX}/F_{HOCO} (通过RTCCL7选择)/1464
1	0	1	主时钟 F_{MAX}/F_{HOCO} (通过RTCCL7选择)/976
1	1	0	主时钟 F_{MAX}/F_{HOCO} (通过RTCCL7选择)/488
1	1	1	主时钟 F_{MAX}/F_{HOCO} (通过RTCCL7选择)/244

12.3.3 15位间隔定时器的控制寄存器(ITMC)

这是设置15位间隔定时器的运行开始和停止以及比较值的寄存器。通过16位存储器操作指令设置ITMC寄存器。

在产生复位信号后，此寄存器的值变为“7FFFH”。

图12-4：15位间隔定时器控制寄存器(ITMC)的格式

地址：0x40044F50 复位后：7FFFHR/W

符号 15 14~0

ITMC

RINTE	ITCMP14~ITCMP0
-------	----------------

RINTE	15位间隔定时器的运行控制
0	停止计数器的运行(清除计数)。
1	开始计数器的运行。

ITCMP14~ITCMP0	15位间隔定时器比较值的设置
0001H	这些位产生“计数时钟周期×(ITCMP设置值+1)”的固定周期中断。
•	
•	
7FFFH	禁止设置。
0000H	
ITCMP14~ITCMP0为“0001H”或者“7FFFH”时的中断周期例子	
•ITCMP14~ITCMP0=0001H, 计数时钟: $F_{SUB}=32.768\text{KHz}$	
$1/32.768[\text{KHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$	
• ITCMP14~ITCMP0=7FFFH, 计数时钟: $F_{SUB}=32.768\text{KHz}$	
$1/32.768[\text{KHz}] \times (32767+1) = 1000[\text{ms}]$	

注意：

1. 要将RINTE位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将INTIT设置为禁止中断处理后进行改写。要重新开始运行(从“0”改为“1”)时，必须在清除ITIF标志后设置为允许中断处理。
2. RINTE位的读取值在设置RINTE位后的1个计数时钟之后被反映。
3. 在从睡眠模式转移到通常运行模式后，如果要设置ITMC寄存器并且再次转移到睡眠模式，就必须在确认ITMC寄存器的写入值被反映后或者在设置ITMC寄存器后至少经过1个计数时钟之后再转移到睡眠模式。
4. 要更改ITCMP14~ITCMP0位的设置时，必须在RINTE位为“0”的状态下进行。但是，能在将RINTE位从“0”改为“1”或者从“1”改为“0”的同时更改ITCMP14~ITCMP0位的设置。

12.4 15位间隔定时器的运行

12.4.1 15位间隔定时器的运行时序

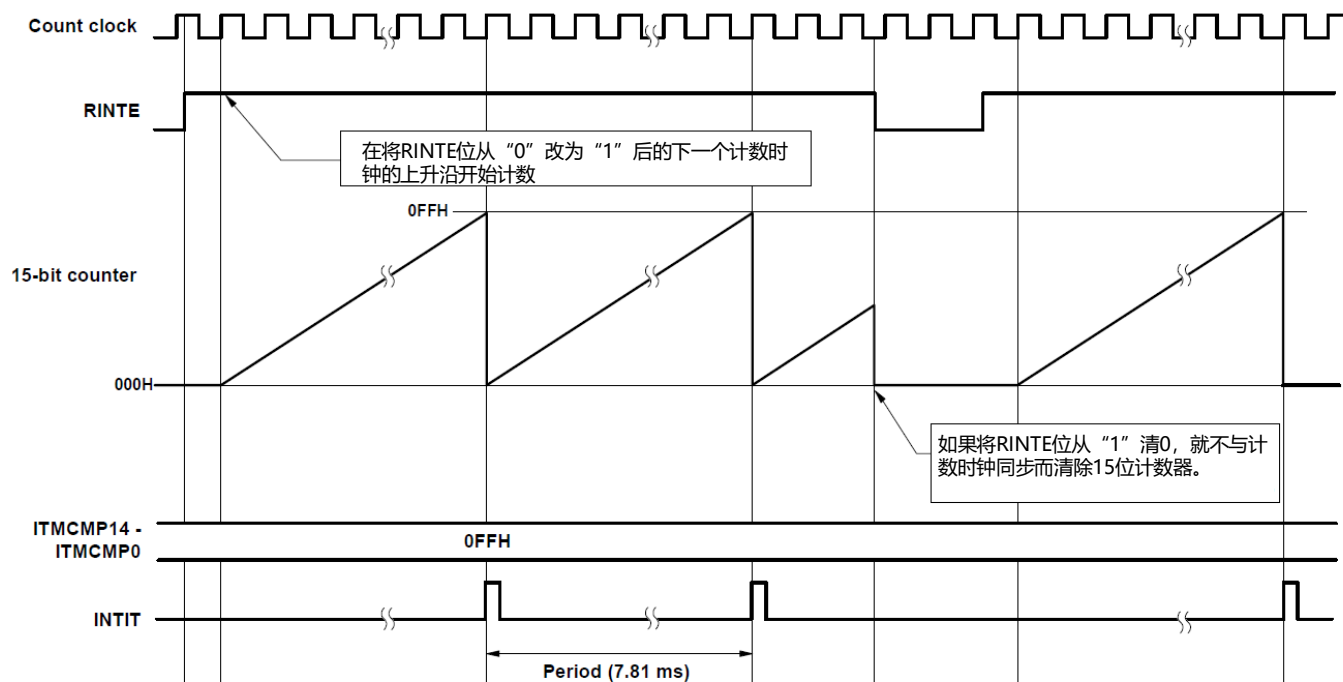
以ITCMP14~ITCMP0位设置的计数值为间隔，作为重复产生中断请求(INTIT)的15位间隔定时器运行。如果将RINTE位置“1”，15位计数器就开始计数。

当15位计数值和ITCMP14~ITCMP0位的设置值相同时，将15位计数值清“0”并且继续计数，同时产生中断请求信号(INTIT)。

15位间隔定时器的基本运行如图12-5所示。

图12-5：15位间隔定时器的运行时序

(ITCMP14~ITCMP0=0FFH, 计数时钟: FSUB=32.768KHz)

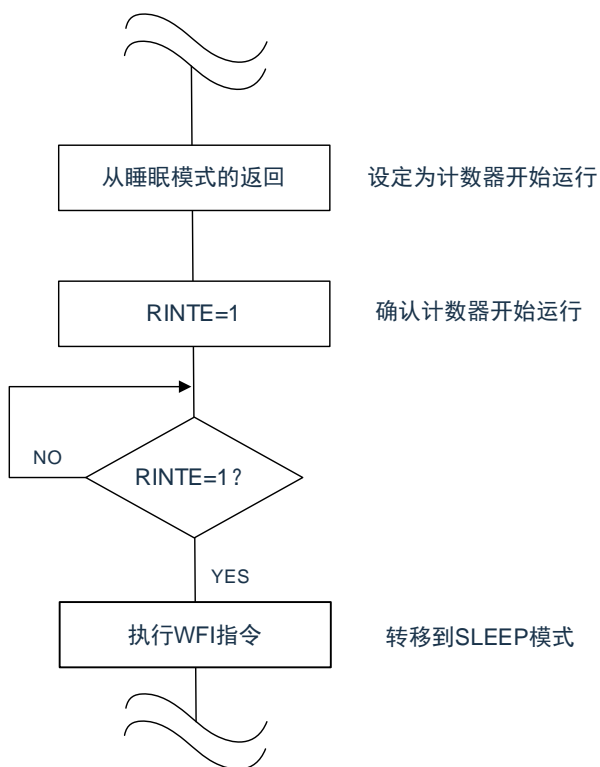


12.4.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

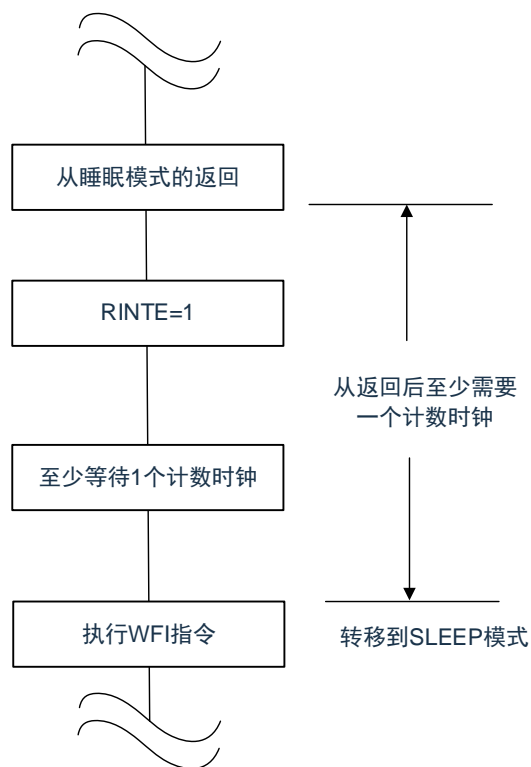
在从睡眠模式返回后，如果要将RINTE位置“1”并且再次转移到睡眠模式，就必须在将RINTE位置“1”后确认RINTE位的写入值被反映，或者在返回后至少经过1个计数时钟的时间之后再转移到睡眠模式。

- 在将RINTE位置“1”后，通过轮询确认RINTE位变为“1”，然后转移到睡眠模式(参照下图的例1)。
- 在将RINTE位置“1”后至少经过1个计数时钟的时间之后转移到睡眠模式(参照下图的例2)。

例1



例2



第13章 时钟输出/蜂鸣器输出控制电路

13.1 时钟输出/蜂鸣器输出控制电路的功能

时钟输出是输出提供给外围IC时钟的功能，蜂鸣器输出是输出蜂鸣器频率方波的功能。

能用1个引脚选择用作时钟输出或者蜂鸣器输出。

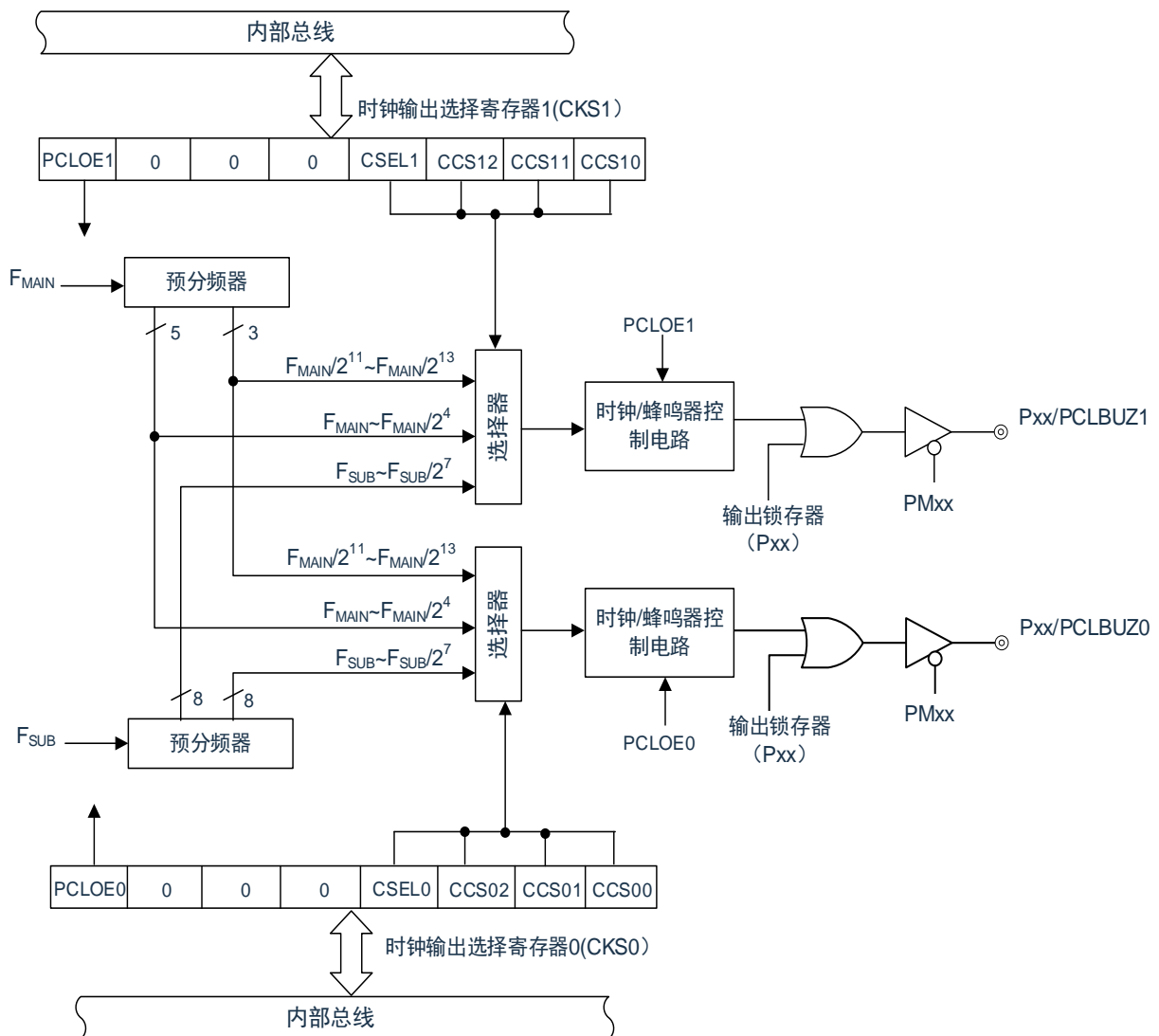
CLKBUZn引脚输出由时钟输出选择寄存器n(CKSn)选择的时钟。

时钟输出蜂鸣器输出控制电路的框图如图13-1所示。

注意：在副系统时钟提供模式控制寄存器(OSMC)的 RTCLPC位为“1时并且在CPU以副系统时钟(F_{SUB})运行的SLEEP模式中，不能从 CLKBUZn引脚输出副系统时钟(F_{SUB})。

备注：n=0、1。

图13-1：时钟输出/蜂鸣器输出控制电路的框图



注：有关能从CLKBUZ0引脚和CLKBUZ1引脚输出的频率，请参照“AC特性”。

13.2 时钟输出/蜂鸣器输出控制电路的结构

时钟输出/蜂鸣器输出控制电路由以下硬件构成。

表13-1: 时钟输出/蜂鸣器输出控制电路的寄存器

项目	寄存器列表
控制寄存器	时钟输出选择寄存器n(CKSn) 端口模式寄存器(PMmn)端口寄存器Pmn)

13.3 控制时钟输出/蜂鸣器输出控制电路的寄存器

13.3.1 时钟输出选择寄存器n(CKSn)

这是允许或者禁止时钟输出引脚或者蜂鸣器频率输出引脚(CLKBUNz)的输出以及设置输出时钟的寄存器。

通过CKSn寄存器选择CLKBUZn引脚输出的时钟。通过8位存储器操作指令设置CKSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图13-2: 时钟输出选择寄存器n(CKSn)的格式

地址: 0x40040FA5(CKS0)、0x40040FA6(CKS1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0	
CKSn	PCLOEn	0	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	CLKBUZn引脚输出允许/ 禁止的指定
0	禁止输出(默认值)。
1	允许输出。

CSELn	CCSn2	CCSn1	CCSn0		CLKBUZn引脚输出时钟的选择			
					F _{MAIN} =10MH z	F _{MAIN} =20MH z	F _{MAIN} =32MH z	F _{MAIN} =48MH z
0	0	0	0	F _{MAIN}	10MHz ^注	禁止设置 ^注	禁止设置 ^注	禁止设置 ^注
0	0	0	1	F _{MAIN} /2	5MHz	10MHz ^注	16MHz ^注	禁止设置 ^注
0	0	1	0	F _{MAIN} /2 ²	2.5MHz	5MHz	8MHz	12MHz
0	0	1	1	F _{MAIN} /2 ³	1.25MHz	2.5MHz	4MHz	6MHz
0	1	0	0	F _{MAIN} /2 ⁴	625KHz	1.25MHz	2MHz	3MHz
0	1	0	1	F _{MAIN} /2 ¹¹	4.88KHz	9.77KHz	15.63KHz	23.44KHz
0	1	1	0	F _{MAIN} /2 ¹²	2.44KHz	4.88KHz	7.81KHz	11.72KHz
0	1	1	1	F _{MAIN} /2 ¹³	1.22KHz	2.44KHz	3.91KHz	5.86KHz
1	0	0	0	F _{SUB}	32.768KHz			
1	0	0	1	F _{SUB} /2	16.384KHz			
1	0	1	0	F _{SUB} /2 ²	8.192KHz			
1	0	1	1	F _{SUB} /2 ³	4.096KHz			
1	1	0	0	F _{SUB} /2 ⁴	2.048KHz			
1	1	0	1	F _{SUB} /2 ⁵	1.024KHz			
1	1	1	0	F _{SUB} /2 ⁶	512Hz			
1	1	1	1	F _{SUB} /2 ⁷	256Hz			

注: 必须在16MHz以内的范围内使用输出时钟。详细内容请参照“AC特性”。

注意:

1. 输出时钟的切换必须在设置为禁止输出(PCLOEn=0)后进行。
2. 在选择主系统时钟(CSELn=0)时, 如果要转移到深度睡眠模式, 就必须在执行WFI指令前将PCLOEn置“0”; 在选择副系统时钟(CSELn=1)时, 因为能在副系统时钟提供模式控制寄存器(OSMC)的RTCLPC位为“0”时并且在深度睡眠模式中输出时钟, 所以能将PCLOEn置“1”。
3. 在副系统时钟提供模式控制寄存器(OSMC)的RTCLPC位为“1”时并且在CPU以副系统时钟(F_{SUB})运行的睡眠模式中, 不能从CLKBUZn引脚输出副系统时钟(F_{SUB})。

备注: n=0、1

F_{MAIN}: 主系统时钟频率

F_{SUB}: 副系统时钟频率

13.3.2 控制时钟输出/蜂鸣器输出引脚端口功能的寄存器

在用作时钟输出/蜂鸣器输出功能时，必须设置与对象通道复用的端口功能的控制寄存器(端口模式寄存器(PMxx)和端口寄存器(Pxx))。详细内容请参照“2.3.1 端口模式寄存器(PMxx)”和“2.3.2 端口寄存器(Pxx)”。

在将时钟输出/蜂鸣器输出引脚的复用端口用作时钟输出/蜂鸣器输出时，必须将各端口对应的端口模式寄存器(PMxx)的位和端口寄存器(Pxx)的位置“0”。

(例)将P140/INTP6/CLKBUZ0用作时钟输出/蜂鸣器输出的情况

将端口模式寄存器14的PM140位置“0”。

将端口寄存器14的P140位置“0”。

13.4 时钟输出/蜂鸣器输出控制电路的运行

能用1个引脚选择用作时钟输出或者蜂鸣器输出。

CLKBUZ0引脚输出由时钟输出选择寄存器0(CKS0)选择的时钟/蜂鸣器。

CLKBUZ1引脚输出由时钟输出选择寄存器1(CKS1)选择的时钟/蜂鸣器。

13.4.1 输出引脚的运行

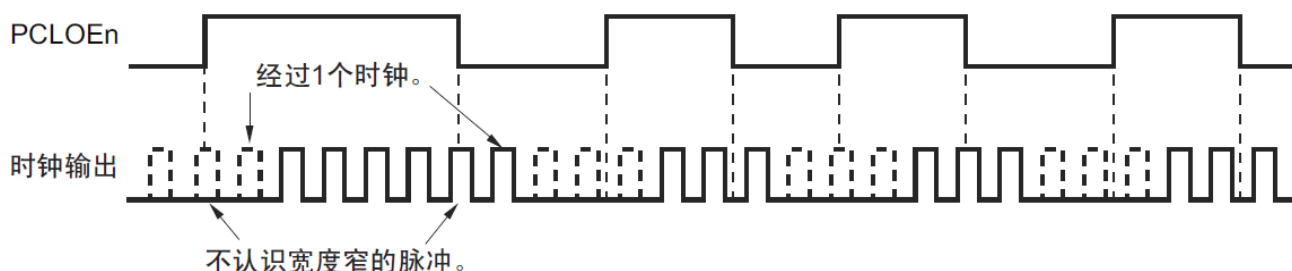
CLKBUZn引脚按照以下步骤进行输出：

- 1) 将用作CLKBUZ0 引脚的端口对应的端口模式寄存器(PMxx)和端口寄存器(Pxx)的位置“0”。
- 2) 通过CLKBUZn引脚的时钟输出选择寄存器(CKSn)的bit0~3(CCSn0~CCSn2、CSELn)选择输出频率(输出为禁止状态)。
- 3) 将CKSn寄存器的bit7(PCLOEn)置“1”，允许时钟/蜂鸣器的输出。

备注：

- ① 用作时钟输出时的控制电路在允许或者禁止时钟输出(PCLOEn位)后的1个时钟之后，开始或者停止时钟输出。此时不输出宽度窄的脉冲。通过PCLOEn位允许或者停止输出以及时钟输出的时序如图13-3所示。
- ② n=0、1

图13-3: CLKBUZn引脚的时钟输出时序



13.5 时钟输出/蜂鸣器输出控制电路的注意事项

当选择主系统时钟作为CLKBUZn输出(CSELn=0)时，如果在设置停止输出(PCLOEn=0)后的1.5个CLKBUZn引脚的输出时钟内转移到深度睡眠模式，CLKBUZn的输出宽度就变窄。

第14章 看门狗定时器

14.1 看门狗定时器的功能

看门狗定时器通过选项字节(000C0H)设置计数运行。看门狗定时器以低速内部振荡器时钟(F_{IL})运行。看门狗定时器用于检测程序失控。在检测到程序失控时，产生内部复位信号。

下述情况判断为程序失控。

- 当看门狗定时器的计数器发生上溢时
- 当对看门狗定时器的允许寄存器(WDTE)执行位操作指令时
- 当给WDTE寄存器写“ACH”以外的数据时
- 在窗口关闭期间给WDTE寄存器写数据时

当因看门狗定时器而发生复位时，将复位控制标志寄存器(RESF)的bit4(WDTRF)置“1”。有关RESF寄存器的详细内容，请参照“第28章 复位功能”。当达到上溢时间的75%+1/2 F_{IL} 时，能产生间隔中断。

14.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表14-1：看门狗定时器的结构

项目	结构
计数器	内部计数器(17位)
控制寄存器	看门狗定时器的允许寄存器(WDTE)

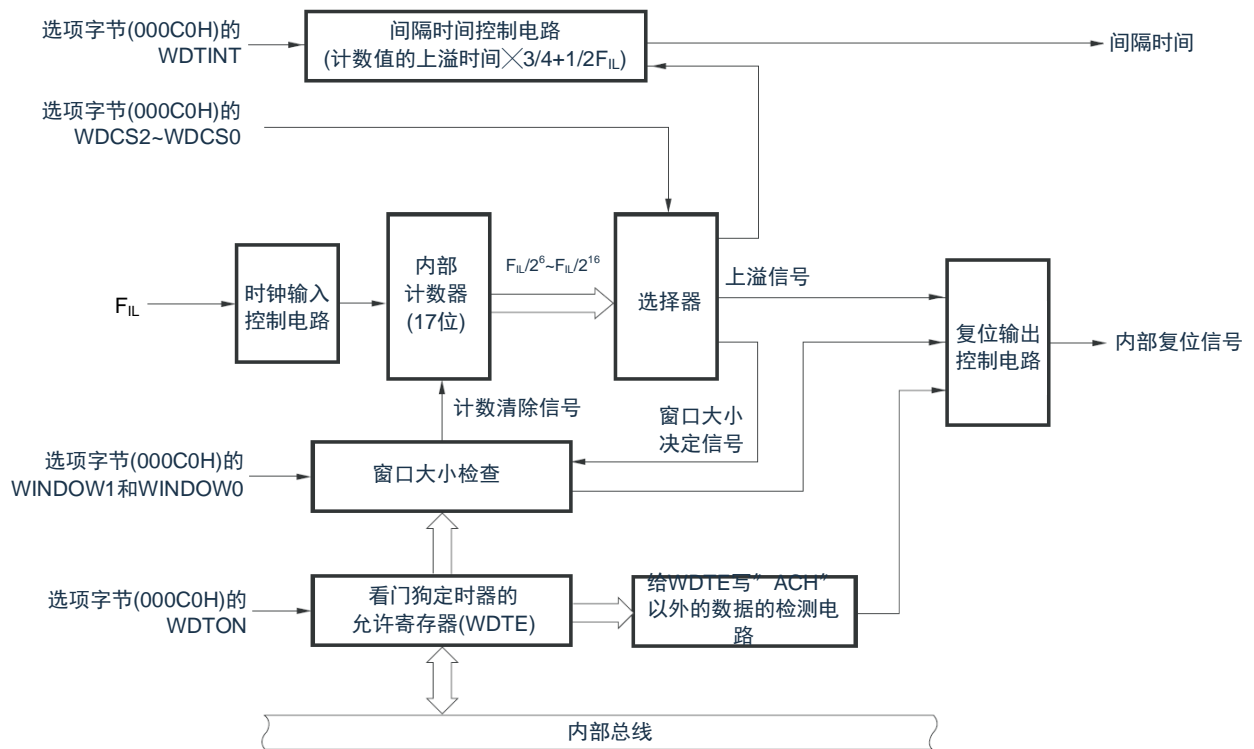
通过选项字节控制计数器的运行以及设置上溢时间、窗口打开期间和间隔中断。

表14-2：选项字节和看门狗定时器的设置内容

看门狗定时器的设置内容	选项字节(000C0H)
看门狗定时器的间隔中断的设置	bit7(WDTINT)
窗口打开期间的设置	bit6和bit5(WINDOW1、WINDOW0)
看门狗定时器的计数器运行控制	bit4(WDTON)
看门狗定时器的上溢时间的设置	bit3~1(WDCS2~WDCS0)
看门狗定时器的计数器运行控制(睡眠时)	bit0(WDSTBYON)

备注：有关选项字节，请参照“第33章 选项字节”。

图14-1：看门狗定时器的框图



备注：F_{IL}：低速内部振荡器的时钟频率

14.3 控制看门狗定时器的寄存器

通过看门狗定时器的允许寄存器(WDTE)控制看门狗定时器。

14.3.1 看门狗定时器的允许寄存器(WDTE)

通过给WDTE寄存器写“ACH”，清除看门狗定时器的计数器并且重新开始计数。通过8位存储器操作指令设置WDTE寄存器。在产生复位信号后，此寄存器的值变为“9AH”或者“1AH”^注。

图14-2：看门狗定时器的允许寄存器(WDTE)的格式

地址：0x40021001 复位后：9AH/1AH^注 R/W



注：WDTE寄存器的复位值因选项字节(000C0H)的WDTON位的设置值而不同。要使看门狗定时器运行时，必须将WDTON位置“1”。

WDTON位的设置值	WDTE寄存器的复位值
0(禁止看门狗定时器的计数运行)	1AH
1(允许看门狗定时器的计数运行)	9AH

注意：

1. 当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。
2. 当对WDTE寄存器执行位操作指令时，产生内部复位信号。
3. WDTE寄存器的读取值为“9AH/1AH”(和写入值(“ACH”)不同)。

14.3.2 LOCKUP控制寄存器(LOCKCTL)及其保护寄存器(PCR)

LOCKCTL寄存器是Cortex-M0+ LockUp功能是否引起看门狗定时器运行的配置寄存器，PCR是其写保护寄存器。

通过8位存储器操作指令设置LOCKCTL，PCR寄存器。

在产生复位信号后，LOCKCTL，PCR寄存器的值变为“00H”。

图14-3: LOCKUP控制寄存器(LOCKCTL)
及其保护寄存器(PCR)的格式(1/2)

地址: 40020405H 复位后: 01H R/W

符号	7	6	5	4	3	2	1	0
LOCKCTL	0	0	0	0	0	0	0	lockup_rst

lockup_rst	LOCKUP功能的配置
0	• LOCKUP不导致WDT复位
1	• LOCKUP导致WDT复位

图14-3: LOCKUP控制寄存器(LOCKCTL)
及其保护寄存器(PCR)的格式(2/2)

地址: 40020406H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PCR	PRTKEY[7:1]							PCR

PCR	LOCKUP控制寄存器写保护
0	• LOCKCTL寄存器不可写
1	• LOCKCTL寄存器可写

PRTKEY[7:1]	PCR的写保护
78H	• PCR可写
其它	• PCR不可写

14.3.3 WDTCFG配置寄存器(WDTCFG0/1/2/3)

WDTCFG配置寄存器是是否强制看门狗定时器运行的寄存器。

通过8位存储器操作指令设置WDTCFG寄存器。

在产生复位信号后，WDTCFG寄存器的值变为“00H”。

图14-4：WDTCFG配置寄存器(WDTCFG0/1/2/3)

地址：40020408H 复位后：00H R/W

WDTCFG0	WDTCFG0
---------	---------

地址：40020409H 复位后：00H R/W

WDTCFG1	WDTCFG1
---------	---------

地址：4002040AH 复位后：00H R/W

WDTCFG2	WDTCFG2
---------	---------

地址：4002040BH 复位后：00H R/W

WDTCFG3	WDTCFG3
---------	---------

WDTCFG0	WDTCFG1	WDTCFG2	WDTCFG3	看门狗定时器功能的配置
0x1A	0x2B	0x3C	0x4D	•复位后看门狗定时器的运行由选项字节决定 ^{注1注2}
其他				•复位后强制运行看门狗定时器

注意：详细配置参考33.2 用户选项字节章节。

14.4 看门狗定时器的运行

14.4.1 看门狗定时器的运行控制

(1) 当使用看门狗定时器时，通过选项字节(000C0H)设置以下内容：

- 必须将选项字节(000C0H)的bit4(WDTON)置“1”，允许看门狗定时器的计数运行(在解除复位后，计数器开始运行)(详细内容请参照第33章 选项字节)。

WDTON	看门狗定时器的计数器
0	禁止计数运行(解除复位后停止计数)。
1	允许计数运行(解除复位后开始计数)。

- 必须通过选项字节(000C0H)的bit3~1(WDCS2~WDCS0)设置上溢时间(详细内容请参照14.4.2和第33章)。
- 必须通过选项字节(000C0H)的bit6和bit5(WINDOW1、WINDOW0)设置窗口打开期间(详细内容请参照14.4.2和第33章)。

(2) 在解除复位后，看门狗定时器开始计数。

(3) 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器(WDTE)写“ACH”，就清除看门狗定时器并且重新开始计数。

(4) 此后，解除复位后第2次以后的WDTE寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写WDTE寄存器，就产生内部复位信号。

(5) 如果不给WDTE寄存器写“ACH”而超过上溢时间，就产生内部复位信号。以下情况会产生内部复位信号：

- 当对WDTE寄存器执行位操作指令时
- 当给WDTE寄存器写“ACH”以外的数据时

注意：

1. 只在解除复位后第1次写看门狗定时器的允许寄存器(WDTE)时，与窗口打开期间无关，只要在上溢时间前的任意时候写WDTE，就清除看门狗定时器并且重新开始计数。
2. 从给WDTE寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大2个 F_{IL} 时钟的误差。
3. 在计数值发生上溢前，都能清除看门狗定时器。
4. 如下所示，看门狗定时器在睡眠或者深度睡眠模式中的运行因选项字节(000C0H)的bit0(WDSTBYON)的设置值而不同。

	WDSTBYON=0	WDSTBYON=1
睡眠模式	停止看门狗定时器运行。	继续看门狗定时器运行。
深度睡眠模式		

当WDSTBYON位为“0”时，在解除睡眠或者深度睡眠模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。

当解除深度睡眠模式后以X1振荡时钟运行时，CPU在经过振荡稳定时间后开始运行。

如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除深度睡眠模式后，如果要以X1振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设置。

14.4.2 看门狗定时器上溢时间的设置

通过选项字节(000C0H)的bit3~1(WDCS2~WDCS0)设置看门狗定时器的上溢时间。

在发生上溢时，产生内部复位信号。如果在上溢时间前的窗口打开期间给看门狗定时器的允许寄存器(WDTE)写“ACH”，就清除计数并且重新开始计数。能设置的上溢时间如下所示。

表14-3: 看门狗定时器上溢时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($F_{IL}=20\text{KHz}(\text{MAX.})$ 的情况)
0	0	0	$2^6/F_{IL}(3.2\text{ms})$
0	0	1	$2^7/F_{IL}(6.4\text{ms})$
0	1	0	$2^8/F_{IL}(12.8\text{ms})$
0	1	1	$2^9/F_{IL}(25.6\text{ms})$
1	0	0	$2^{11}/F_{IL}(102.4\text{ms})$
1	0	1	$2^{13}/F_{IL}(409.6\text{ms})$
1	1	0	$2^{14}/F_{IL}(819.2\text{ms})$
1	1	1	$2^{16}/F_{IL}(3276.8\text{ms})$

备注： F_{IL} ：低速内部振荡器的时钟频率。

14.4.3 看门狗定时器窗口打开期间的设置

通过选项字节(000C0H)的bit6和bit5(WINDOW1、WINDOW0)设置看门狗定时器的窗口打开期间。窗口概要如下:

- 如果在窗口打开期间给看门狗定时器的允许寄存器(WDTE)写“ACH”，就清除看门狗定时器并且重新开始计数。
- 在窗口关闭期间，即使给WDTE寄存器写“ACH”，也会检测到异常并且产生内部复位信号。

注意：只在解除复位后第1次写WDTE寄存器时，与窗口打开期间无关，只要在上溢时间前的任意时候写WDTE，就清除看门狗定时器并且重新开始计数。

能设置的窗口打开期间如下所示。

表14-4：看门狗定时器窗口打开期间的设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	-	禁止设置
1	0	75%
1	1	100%

注意：当选项字节(000C0H)的bit0(WDSTBYON)为“0”时，与WINDOW1位和WINDOW0位的值无关，窗口打开期间为100%。

备注：当将上溢时间设置为 $2^9/F_{IL}$ 的情况时，窗口关闭时间和打开时间如下所示。

	窗口打开期间的设置	
	75%	100%
窗口关闭时间	0~12.8ms	无
窗口打开时间	12.8~25.6ms	0~25.6ms

<当窗口打开期间为75%时>

上溢时间:

$$2^9/F_{IL}(\text{MAX.})=2^9/20\text{KHz}(\text{MAX.})=25.6\text{ms}$$

窗口关闭时间:

$$0\sim 2^9/F_{IL}(\text{MIN.})\times(1-0.75)=0\sim 2^9/10\text{KHz}\times 0.25=0\sim 12.8\text{ms}$$

窗口打开时间:

$$2^9/F_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/F_{IL}(\text{MAX.})=12.8\sim 25.6\text{ms}$$

14.4.4 看门狗定时器间隔中断的设置

能通过设置选项字节(000C0H)的bit7(WDTINT)，在达到上溢时间的 $75\%+1/2F_{IL}$ 时产生间隔中断(INTWDTI)。

表14-5: 看门狗定时器间隔中断的设置

WDTINT	看门狗定时器间隔中断的使用/不使用
0	不使用间隔中断。
1	在达到上溢时间的 $75\%+1/2F_{IL}$ 时，产生间隔中断。

注意：当解除深度睡眠模式后以X1振荡时钟运行时，CPU在经过振荡稳定时间后开始运行。如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除深度睡眠模式后，如果要以X1振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设置。

备注：即使在产生INTWDTI后也继续计数(继续到给看门狗定时器的允许寄存器(WDTE)写“ACH”为止)。如果在上溢时间前不给WDTE寄存器写“ACH”，就产生内部复位信号。

14.4.5 WDTCFG未配置时看门狗定时器的运行

当WDTCFG未配置时，看门狗定时器的计时器自动开始运行，上溢时间由选项字节里的上溢时间控制位(WDCS2~WDCS0)决定。

第15章 A/D转换器

A/D转换器的模拟输入通道数因产品而不同。

引脚数	32引脚	48引脚	64引脚
模拟输入通道	10ch	15ch	16ch
	(ANI0~ANI3、ANI8~ANI12、ANI14)	(ANI0~ANI12) (ANI14~ANI15)	(ANI0~ANI15)

15.1 A/D转换器的功能

A/D转换器是将模拟输入转换为数字值的转换器，最多能控制19个模拟通道的A/D转换(16个引脚输入通道和3个内部通道)。

A/D转换器有以下的功能。

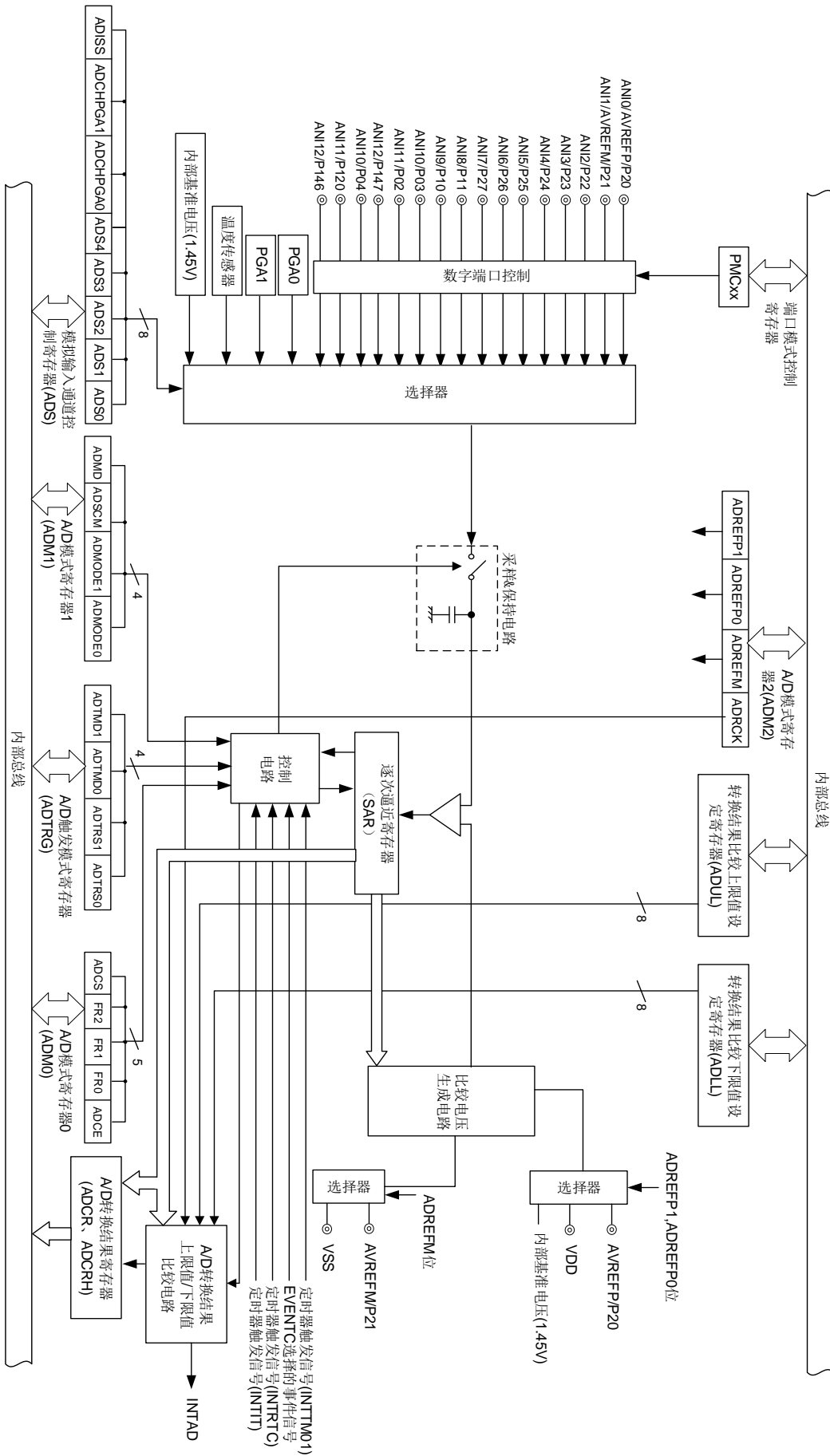
- 12位分辨率的A/D转换

从ANI0~ANI15, PGA0, PGA1和温度传感器中选择1个通道的模拟输入，重复进行12位分辨率的A/D转换。每结束1次A/D转换，就产生中断请求(INTAD)(选择模式的情况)。

能通过下述的模式组合设置各种A/D转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断电源的转换待机状态下，通过检测硬件触发来接通电源，在经过A/D电源稳定等待时间后自动开始转换。
通道选择模式	选择模式	选择1个通道的模拟输入，进行A/D转换。
	扫描模式	按顺序对4个通道的模拟输入进行A/D转换。能选择ANI0~ANI15中连续的4个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行1次A/D转换。
	连续转换模式	对所选通道进行连续的A/D转换，直到被软件停止为止。
采样时间	采样时钟5.5~255个ADCLK	采样时间可以通过ADNSMP寄存器来选择，默认使用13.5个转换时钟(F _{AD})。

图15-1: A/D转换器的框图



15.2 控制A/D转换器的寄存器

控制A/D转换器的寄存器如下所示：

寄存器基地址：CSC_BASE=4002_0420H; ADC_BASE=4004_5000H; PORT_BASE=4004_0000H

寄存器名称	寄存器描述	R/W	复位值	寄存器地址
PER0	外围使能寄存器0	R/W	00H	CSC_BASE+20H
ADM0	A/D转换器的模式寄存器0	R/W	00H	ADC_BASE+00H
ADM1	A/D转换器的模式寄存器1	R/W	00H	ADC_BASE+02H
ADM2	A/D转换器的模式寄存器2	R/W	00H	ADC_BASE+04H
ADTRG	A/D转换器的触发模式寄存器	R/W	00H	ADC_BASE+06H
ADS	模拟输入通道指定寄存器	R/W	00H	ADC_BASE+08H
ADLL	转换结果比较下限值设置寄存器	R/W	00H	ADC_BASE+0AH
ADUL	转换结果比较上限值设置寄存器	R/W	00H	ADC_BASE+0BH
ADNSMP	A/D转换器的采样时间控制寄存器	R/W	0dH	ADC_BASE+0CH
ADCR	12位A/D转换结果寄存器	R	0000H	ADC_BASE+0EH
ADCRH	8位A/D转换结果寄存器	R	00H	ADC_BASE+0FH
ADTES	A/D测试寄存器	R/W	00H	ADC_BASE+10H
ADNDIS	A/D转换器的充放电控制寄存器	R/W	00H	ADC_BASE+11H
ADSMPWAIT	A/D转换器的采样时间延长控制寄存器	R/W	00H	ADC_BASE+15H
ADFLG	A/D硬模块状态寄存器	R	00H	ADC_BASE+16H
PMC0	端口模式控制寄存器0	R/W	FFH	PORT_BASE+60H
PMC1	端口模式控制寄存器1	R/W	FFH	PORT_BASE+61H
PMC2	端口模式控制寄存器2	R/W	FFH	PORT_BASE+62H
PMC12	端口模式控制寄存器12	R/W	FFH	PORT_BASE+6CH
PMC14	端口模式控制寄存器14	R/W	FFH	PORT_BASE+6EH

R:read only, W:write only, R/W:both read and write

15.2.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用A/D转换器时，必须将bit5(ADCEN)置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-2：外围允许寄存器0(PER0)的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN

ADCEN	A/D转换器的输入时钟的控制
0	停止提供输入时钟。 •不能写A/D转换器使用的SFR。 •A/D转换器处于复位状态。
1	提供输入时钟。 •能读写A/D转换器使用的SFR。

注意：要设置A/D转换器时，必须先ADCEN位为“1”的状态下设置以下的寄存器。当ADCEN位为“0”时，A/D转换器的控制寄存器的值为初始值，忽视写操作(端口模式控制寄存器(PMCxx)除外)。

- A/D转换器的模式寄存器0(ADM0)
- A/D转换器的模式寄存器1(ADM1)
- A/D转换器的模式寄存器2(ADM2)
- A/D转换器的触发模式寄存器(ADTRG)
- 模拟输入通道指定寄存器(ADS)
- 转换结果比较下限值设置寄存器(ADLL)
- 转换结果比较上限值设置寄存器(ADUL)
- A/D采样时间控制寄存器 (ADNSMP)
- 12位A/D转换结果寄存器(ADCR)
- 8位A/D转换结果寄存器(ADCRH)
- A/D测试寄存器(ADTES)
- A/D充放电控制寄存器(ADNDIS)
- A/D采样时间延长控制寄存器(ADSMPWAIT)
- A/D硬模块状态寄存器(ADFLG)

15.2.2 A/D转换器的模式寄存器0(ADM0)

用于设置A/D转换时钟、转换开始或者停止的寄存器。通过8位存储器操作指令设置ADM0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图15-3: A/D转换器的模式寄存器0(ADM0)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADM0	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D转换运行的控制
0	停止转换运行。 [读时] 停止转换运行/待机状态
1	允许转换运行。 [读时] 软件触发模式时: 转换运行状态 硬件触发等待模式时: A/D电源等待稳定状态+转换运行状态

ADCE	A/D电压比较器的运行控制 ^{注2}
0	停止A/D电压比较器的运行。
1	允许A/D电压比较器的运行。

注1: 有关FR2~FR0位、SHT1~SHT0位和A/D转换的详细内容, 请参照“表15-3: A/D转换时间的选择”。

注2: A/D转换器开始运行需要1us稳定时间。在软件触发模式或者硬件触发无等待模式中, 在将ADCE位置“1”后至少经过1us时间, 然后将ADCS位置“1”, 则本次转换结果有效。如果等待时间小于1us而将ADCS位置“1”, 就必须忽视本次转换结果。在硬件触发等待模式中, 由设计保证1us的等待时间。

注意:

1. 必须在转换停止状态ADCS=0下更改FR2~FR0位。
2. 禁止进行ADCS=1、ADCE=0的设置。
3. 禁止通过8位操作指令将ADCS=0、ADCE=0的状态设置为ADCS=1、ADCE=1。必须按照“15.7 A/D转换器的设置流程图”的步骤进行设置。

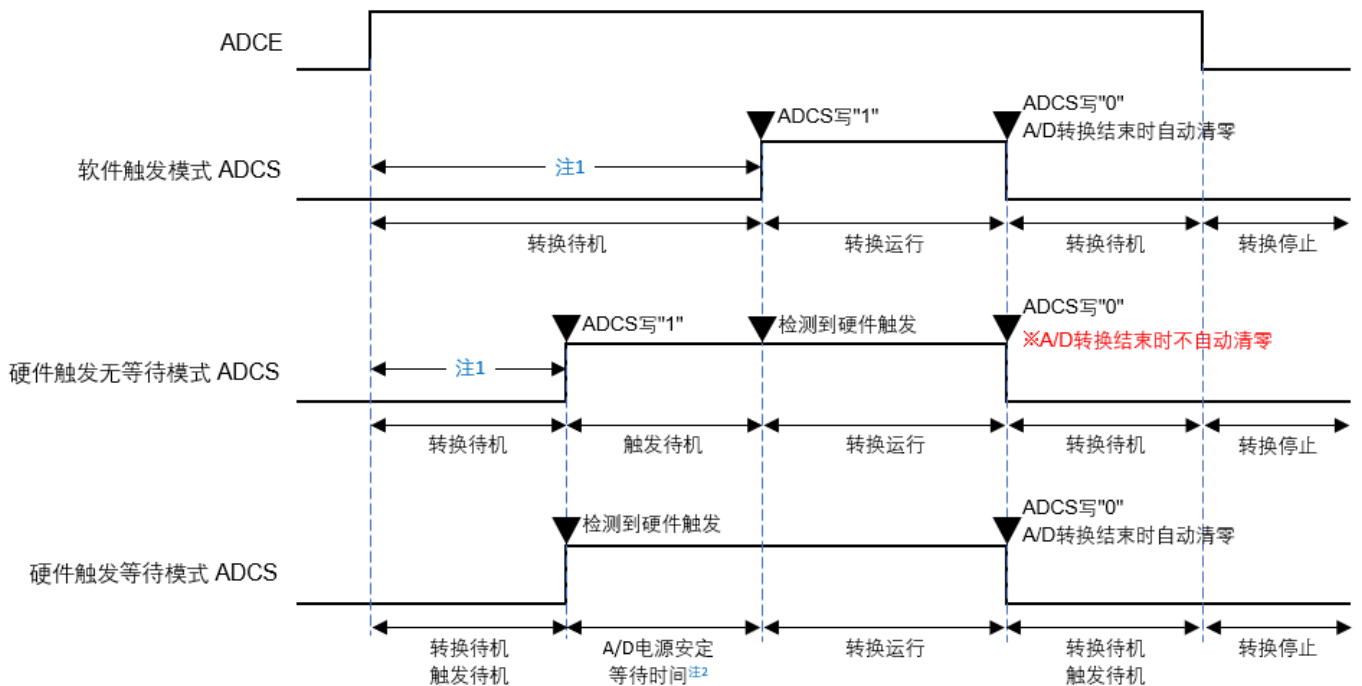
表15-1: ADCS位和ADCE位的设置

ADCS	ADCE	A/D转换运行
0	0	转换停止状态
0	1	转换待机状态
1	0	禁止设置。
1	1	转换运行状态

表15-2: DCS位的置位和清除条件

A/D转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给ADCS位写“1”时	当给ADCS位写“0”时
		单次转换模式		<ul style="list-style-type: none"> •当给ADCS位写“0”时 •在A/D转换结束时自动清“0”。
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		<ul style="list-style-type: none"> •当给ADCS位写“0”时 •当设置的4个通道转换结束时，自动清“0”。
硬件触发无等待模式	选择模式	连续转换模式	当给ADCS位写“1”时	当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时
硬件触发等待模式	选择模式	连续转换模式	当输入硬件触发时	当给ADCS位写“0”时
		单次转换模式		<ul style="list-style-type: none"> •当给ADCS位写“0”时 •在A/D转换结束时自动清“0”。
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		<ul style="list-style-type: none"> •当给ADCS位写“0”时 •当设置的4个通道转换结束时，自动清“0”。

图15-4: 使用A/D各种模式时的动作状态图



注1: 在软件触发模式或者硬件触发无等待模式中, 为了稳定内部电路, 从ADCE位上升到ADCS位上升的时间至少需要1s(TBD)。

注2: 在硬件触发等待模式, A/D电源稳定时间1us由设计保证。

注意:

1. 要使用硬件触发等待模式时, 禁止将ADCS位置“1”(当检测到硬件触发信号时, 自动切换为“1”)。但是, 为了设置为A/D转换待机状态, 能将ADCS位置“0”。
2. 必须在ADCS位为“0”(停止转换/转换待机状态)时改写ADCE位。
3. 为了结束A/D转换, 必须至少将硬件触发间隔设置为以下时间:
 硬件触发无等待模式时: 2个F_{CLK}时钟+A/D转换时间
 硬件触发等待模式时: 2个F_{CLK}时钟+A/D电源稳定等待时间+A/D转换时间

备注: F_{CLK}: CPU/外围硬件的时钟频率

表15-3: A/D转换时间的选择(1/2)

(1) 无A/D电源稳定等待时间(软件触发模式/硬件触发无等待模式)

A/D转换器的模式 寄存器0(ADM0)			A/D转换器的模式 寄存器 1(ADM1)		模式	转换时钟 ADCLK 的频率(F_{AD})	12位分辨率的转换时间	
FR2	FR1	FR0	ADMODE[1]	ADM0D[0]			转换时钟数	转换时间
0	0	0	0	0	高速变 换模式	$F_{CLK}/32$	45个ADCLK (采样时钟数: 13.5个 ADCLK)	$1440/F_{CLK}$
0	0	1				$F_{CLK}/16$		$720/F_{CLK}$
0	1	0				$F_{CLK}/8$		$360/F_{CLK}$
0	1	1				$F_{CLK}/4$		$180/F_{CLK}$
1	0	0				$F_{CLK}/2$		$90/F_{CLK}$
1	0	1				$F_{CLK}/1$		$45/F_{CLK}$
0	0	0	1	1	低电流 模式	$F_{CLK}/32$	54个ADCLK (采样时钟数: 13.5个 ADCLK)	$1728/F_{CLK}$
0	0	1				$F_{CLK}/16$		$864/F_{CLK}$
0	1	0				$F_{CLK}/8$		$432/F_{CLK}$
0	1	1				$F_{CLK}/4$		$216/F_{CLK}$
1	0	0				$F_{CLK}/2$		$108/F_{CLK}$
1	0	1				$F_{CLK}/1$		$54/F_{CLK}$

注意: 要将FR2~FR0位、ADMODE[1:0]位改写为不同数据时, 必须在转换停止状态(ADCS=0)下进行。

备注: F_{CLK} : CPU/外围硬件的时钟频率

表15-3: A/D转换时间的选择(2/2)

 (2) 有A/D电源稳定等待时间(硬件触发等待模式^{注1})

A/D转换器的模式 寄存器0(ADM0)			A/D转换器的模式 寄存器 1(ADM1)		模式	转换时钟 ADCLK的 频率(F_{AD})	A/D电源 稳定等待 时间	转换时钟数	A/D电源稳定 等待时间 +转换时间
FR2	FR1	FR0	ADM0DE[1]	ADM0D[0]					
0	0	0	0	0	高速变 换模式	$F_{CLK}/32$	1us	45个ADCLK (采样时钟数: 13.5 个ADCLK)	$1\mu s + 1440/F_{CLK}$
0	0	1				$F_{CLK}/16$			$1\mu s + 720/F_{CLK}$
0	1	0				$F_{CLK}/8$			$1\mu s + 360/F_{CLK}$
0	1	1				$F_{CLK}/4$			$1\mu s + 180/F_{CLK}$
1	0	0				$F_{CLK}/2$			$1\mu s + 90/F_{CLK}$
1	0	1				$F_{CLK}/1$			$1\mu s + 45/F_{CLK}$
0	0	0	1	1	低电流 模式	$F_{CLK}/32$	1us	54个ADCLK (采样时钟数: 13.5 个ADCLK)	$1\mu s + 1728/F_{CLK}$
0	0	1				$F_{CLK}/16$			$1\mu s + 864/F_{CLK}$
0	1	0				$F_{CLK}/8$			$1\mu s + 432/F_{CLK}$
0	1	1				$F_{CLK}/4$			$1\mu s + 216/F_{CLK}$
1	0	0				$F_{CLK}/2$			$1\mu s + 108/F_{CLK}$
1	0	1				$F_{CLK}/1$			$1\mu s + 54/F_{CLK}$

注1: 在连续转换模式时, 仅在第1次检测到硬件触发后, 发生A/D电源稳定等待时间(参照表15-3)。

注意:

1. 要将FR2~FR0位、ADM0DE[1:0]位改写为不同数据时, 必须在转换停止状态(ADCS=0)下进行。
2. 硬件触发等待模式中的转换时间包含检测到硬件触发后的A/D电源稳定等待时间。

备注: F_{CLK} : CPU/外围硬件的时钟频率。

15.2.3 A/D转换器的模式寄存器1(ADM1)

这是设置A/D转换模式的寄存器。

通过8位存储器操作指令设置ADM1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-5: A/D转换器的模式寄存器1(ADM1)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADM1	ADMD	0	0	0	ADSCM	0	ADMODE1	ADMODE0

ADMODE1	ADMODE0	A/D转换模式
0	0	高速变换模式
1	1	低电流模式
其他		禁止设置

ADSCM	A/D转换模式的设置
0	连续转换模式
1	单次转换模式

ADMD	A/D转换通道选择模式的设置
0	选择模式
1	扫描模式

注意:

1. 要改写ADM1寄存器时，必须在转换停止状态(ADCS=0, ADCE=0)下进行。
2. 为了正常结束A/D转换，必须至少将硬件触发间隔设置为以下时间:

硬件触发无等待模式时: 2个F_{CLK}时钟+A/D转换时间

硬件触发等待模式时: 2个F_{CLK}时钟+A/D电源稳定等待时间+A/D转换时间

备注: F_{CLK}: CPU/外围硬件的时钟频率

15.2.4 A/D转换器的模式寄存器2(ADM2)

通过8位存储器操作指令设置ADM2寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-6: A/D转换器的模式寄存器2(ADM2)的格式(1/3)

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	0	CHRDE	0

ADREFP1	ADREFP0	A/D转换器的正(+)基准电压源的选择
0	0	由V _{DD} 提供。
0	1	由P20/AV _{REFP} /ANI0提供。
1	0	由A/D转换器内部基准电压提供。
1	1	禁止设置

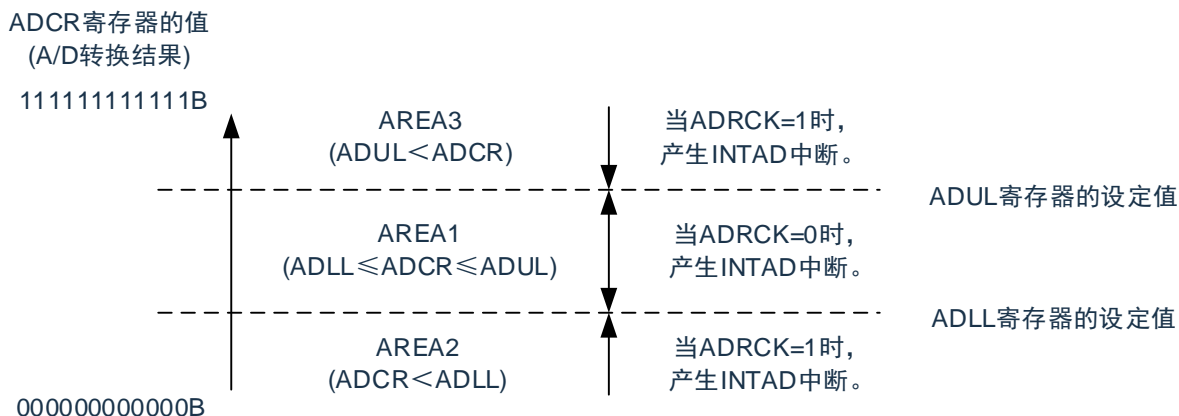
ADREFM	A/D转换器的负(-)基准电压源的选择
0	由V _{SS} 提供。
1	由P21/AV _{REFM} /ANI1提供。

ADRCK	转换结果上限值和下限值的检查
0	当ADLL寄存器 ≤ ADCR寄存器 ≤ ADUL寄存器(AREA1)时，产生中断信号(INTAD)。
1	当ADCR寄存器 < ADLL寄存器(AREA2)或者ADUL寄存器 < ADCR寄存器(AREA3)时，产生中断信号(INTAD)。

AREA1~AREA3的中断信号(INTAD)的产生范围如图15-7所示。

CHRDE	A/D转换器扫描模式时通道标识的输出使能
0	扫描模式时，不在转换结果中标识通道号
1	扫描模式时，转换结果的高四位(ADCR寄存器的[15:12])为此结果的通道号

图15-7: ADRCK位的中断信号产生范围



注意:

1. 要改写ADM2寄存器时，必须在转换停止状态(ADCS=0)下进行。
2. 当使用AVREFP和AVREFM时，必须将ANI0和ANI1设置为模拟输入，并且通过端口模式寄存器设置为输入模式。

备注: 当不发生INTAD时，A/D转换结果不保存到ADCR寄存器和ADCRH寄存器。

15.2.5 A/D转换器的触发模式寄存器(ADTRG)

这是设置A/D转换触发模式和硬件触发信号的寄存器。

通过8位存储器操作指令设置ADTRG寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-8: A/D转换器的触发模式寄存器(ADTRG)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADTRG	ADTMD1	ADTMD0	0	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	A/D转换触发模式的选择
0	0	软件触发模式
0	1	
1	0	硬件触发无等待模式
1	1	硬件触发等待模式

ADTRS1	ADTRS0	硬件触发信号的选择
0	0	定时器通道1的计数结束或者捕捉结束中断信号(INTTM01)
0	1	ELC选择的事件信号
1	0	实时时钟中断信号(INTRTC)
1	1	间隔定时器中断信号(INTIT)

注意:

1. 要改写ADTRG寄存器时，必须在转换停止状态(ADCS=0, ADCE=0)下进行。
2. 为了正常结束A/D转换，必须至少将硬件触发间隔设置为以下时间：
 硬件触发无等待模式时：2个 F_{CLK} 时钟+A/D转换时间
 硬件触发等待模式时：2个 F_{CLK} 时钟+A/D电源稳定等待时间+A/D转换时间

备注: F_{CLK} : CPU/外围硬件的时钟频率

15.2.6 模拟输入通道指定寄存器(ADS)

这是指定要进行A/D转换的模拟电压输入通道的寄存器。

通过8位存储器操作指令设置ADS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-9：模拟输入通道指定寄存器(ADS)的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADS	ADISS	ADCHPGA1	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0

◆ 选择模式(ADM1.ADMD=0)

ADS 寄存器设置值								CH 选择
ADISS	ADCHPGA1	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	0	ANI0(P20)
0	0	0	0	0	0	0	1	ANI1(P21)
0	0	0	0	0	0	1	0	ANI2(P22)
0	0	0	0	0	0	1	1	ANI3(P23)
0	0	0	0	0	1	0	0	ANI4(P24)
0	0	0	0	0	1	0	1	ANI5(P25)
0	0	0	0	0	1	1	0	ANI6(P26)
0	0	0	0	0	1	1	1	ANI7(P27)
0	0	0	0	1	0	0	0	ANI8(P11)
0	0	0	0	1	0	0	1	ANI9(P10)
0	0	0	0	1	0	1	0	ANI10(P03)
0	0	0	0	1	0	1	1	ANI11(P02)
0	0	0	0	1	1	0	0	ANI12(P147)
0	0	0	0	1	1	0	1	ANI13(P04)
0	0	0	0	1	1	1	0	ANI14(P120)
0	0	0	0	1	1	1	1	ANI15(P146)
0	0	0	1	1	1	1	1	ANI0~ANI15 通道全关
1	0	0	0	0	0	0	0	温度传感器的输出电压 ^{注1}
1	0	0	0	0	0	0	1	内部基准电压(1.45V)
0	1	0	0	0	0	0	0	PGA1
0	0	1	0	0	0	0	0	PGA0
其他禁止设置								

注1：如果选择内部基准电压(1.45V)作为比较器0或者比较器1的基准电压，就不能选择温度传感器输出。

◆ 扫描模式(ADM1.ADMD=1)

ADISS	ADS3	ADS2	ADS1	ADS0	模拟输入通道			
					扫描0	扫描1	扫描2	扫描3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
0	1	1	0	0	ANI12	ANI13	ANI14	ANI15
上述以外					禁止设置。			

注意:

1. 必须将bit4,bit5和bit6置“0”。
2. 对于由PMCx寄存器设置为模拟输入的端口, 才可以由ADS指定为模拟输入进行A/D转换。
3. 对于由端口模式控制寄存器(PMCxx)设置为数字输入/输出的引脚, 不能通过ADS寄存器进行设置。
4. 要改写ADISS位时, 必须在转换停止状态(ADCS=0、ADCE=0)下进行。
5. 当将AV_{REFP}用作A/D转换器的正(+)基准电压时, 不能选择ANI0作为A/D转换通道。
6. 当将AV_{REFM}用作A/D转换器的负(-)基准电压时, 不能选择ANI1作为A/D转换通道。
7. 在将ADISS位置“1”后, 不能使用第1次的转换结果。有关详细设置流程, 请参照“15.5.5选择温度传感器的输出电压/内部基准电压时的设置”。
8. 要转移到深度睡眠模式时或者要在CPU以副系统时钟运行中转移到睡眠模式时, 不能将ADISS位置“1”。

15.2.7 12位A/D转换结果寄存器(ADCR)

这是保存A/D转换结果的16位寄存器，此寄存器只可读。每当A/D转换结束时，就从逐次逼近寄存器(SAR)装入转换结果^注。

此寄存器的高4位在选择模式时读出值固定为“0”，在扫描模式时可由ADM2.CHRDE=1配置为此次转换结果的通道号。

通过16位存储器操作指令读ADCR寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

注：如果A/D转换结果的值不在A/D转换结果比较功能(通过ADRCK位和ADUL/ADLL寄存器进行设置(参照图15-7))的设置值范围内，就不保存A/D转换结果。

图15-10：12位A/D转换结果寄存器(ADCR)的格式

复位值：0000H R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	ADCH3	ADCH2	ADCH1	ADCH0	ADCR[11:0]											

注意：

1. 如果在仅仅需要8位分辨率A/D转换结果，可以通过ADCRH寄存器读取转换结果的高8位。
2. 当对ADCR寄存器进行16位存取时，能从bit11依次读转换结果的高12位。

◆ 选择模式(ADM1.ADMD=0)

ADCH0~3的读出值固定为4 ‘b0000

◆ 扫描模式(ADM1.ADMD=1)且ADM2.CHRDE=1，ADCH0~3的读出值与转换通道的关系如下：

ADCH3	ADCH2	ADCH1	ADCH0	转换通道标识
0	0	0	0	ANI0(P20)
0	0	0	1	ANI1(P21)
0	0	1	0	ANI2(P22)
0	0	1	1	ANI3(P23)
0	1	0	0	ANI4(P24)
0	1	0	1	ANI5(P25)
0	1	1	0	ANI6(P26)
0	1	1	1	ANI7(P27)
1	0	0	0	ANI8(P11)
1	0	0	1	ANI9(P10)
1	0	1	0	ANI10(P03)
1	0	1	1	ANI11(P02)
1	1	0	0	ANI12(P147)
1	1	0	1	ANI13(P04)
1	1	1	0	ANI14(P120)
1	1	1	1	ANI15(P146)

15.2.8 8位A/D转换结果寄存器(ADCRH)

这是保存A/D转换结果的8位寄存器，保存12位分辨率的高8位^注。

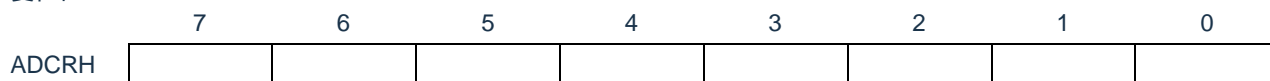
通过8位存储器操作指令读ADCRH寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注：如果A/D转换结果的值不在A/D转换结果比较功能(通过ADRCK位和ADUL/ADLL寄存器进行设置(参照图15-7))的设置值范围内，就不保存A/D转换结果。

图15-11：8位A/D转换结果寄存器(ADCRH)的格式

复位值：00H R



注意：必须在转换结束后并且在配置ADM0、ADS寄存器前读转换结果。否则，就可能读不到正确的转换结果。

15.2.9 转换结果比较上限值设置寄存器(ADUL)

这是用于检查A/D转换结果上限值的设置寄存器。

将A/D转换结果和ADUL寄存器的值进行比较，并且在A/D转换器的模式寄存器2(ADM2)的ADRCK位的设置范围内(参照图15-7)控制中断信号(INTAD)的产生。通过8位存储器操作指令设置ADUL寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

注意：

1. 只将12位A/D转换结果寄存器(ADCR)的高8位和ADUL寄存器以及ADLL寄存器进行比较。
2. 要改写ADUL寄存器和ADLL寄存器时，必须在转换停止状态(ADCS=0)下进行。
3. 在设置ADUL寄存器和ADLL寄存器时，必须使ADUL > ADLL。

图15-12：转换结果比较上限值设置寄存器(ADUL)的格式

复位值：FFH R/W

	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

15.2.10 转换结果比较下限值设置寄存器(ADLL)

这是用于检查A/D转换结果下限值的设置寄存器。

将A/D转换结果和ADLL寄存器的值进行比较，并且在A/D转换器的模式寄存器2(ADM2)的ADRCK位的设置范围内(参照图15-7)控制中断信号(INTAD)的产生。通过8位存储器操作指令设置ADLL寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图15-13：转换结果比较下限值设置寄存器(ADLL)的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

注意：

1. 只将12位A/D转换结果寄存器(ADCR)的高8位和ADUL寄存器以及ADLL寄存器进行比较。
2. 要改写ADUL寄存器和ADLL寄存器时，必须在转换停止状态(ADCS=0)下进行。
3. 在设置ADUL寄存器和ADLL寄存器时，必须使ADUL > ADLL。

15.2.11 A/D采样时间控制寄存器(ADNSMP)

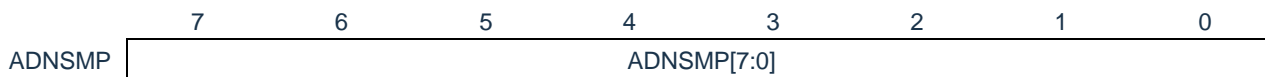
此寄存器可控制A/D采样时间。

通过8位存储器操作指令设置ADNSMP寄存器。

在产生复位信号后，此寄存器的值变为“0dH”。

图15-14: A/D采样时间控制寄存器(ADNSMP)的格式

复位值: 0dH R/W



采样时钟数设置:

ADNSMP[7:0]	采样时间	备注
8'h05	5.5 个 ADCLK	
8'h06	6.5 个 ADCLK	
8'h07	7.5 个 ADCLK	
8'h08	8.5 个 ADCLK	
8'h09	9.5 个 ADCLK	
8'h0a	10.5 个 ADCLK	
8'h0b	11.5 个 ADCLK	
8'h0c	12.5 个 ADCLK	
8'h0d	13.5 个 ADCLK	默认值
8'h0e	14.5 个 ADCLK	
8'h0f	15.5 个 ADCLK	
8'h10	16.5 个 ADCLK	
8'h11	17.5 个 ADCLK	
8'h12	18.5 个 ADCLK	
8'h13	19.5 个 ADCLK	
8'h14	20.5 个 ADCLK	
.....	
8'hff	255.5 个 ADCLK	

注意: 要改写ADNSMP寄存器时, 必须在转换停止状态(ADCS=0)下进行。

不同条件下, 各转换通道需要保证的采样时间:

A/D 转换模式	AVDD[V]	ANI0~ANI15[ns]	PGA0/PGA1[ns]
高速变换	4.5~5.5	211	633
	2.7~5.5	250	750
	2.4~5.5	422	1266
低电流变换	2.7~5.5	500	759
	2.4~5.5	844	1281
	1.8~5.5	1688	2563

15.2.12 A/D采样时间延长寄存器(ADSMPWAIT)

此寄存器用于延长A/D采样时间。

通过8位存储器操作指令设置ADSMPWAIT寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-15: A/D采样时间延长寄存器(ADSMPWAIT)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADSMPWAIT	0	0	0	0	0	0	0	ADSMPWAIT

ADSMPWAIT	A/D转换对象
0	为“0”时，A/D采样时间直接由ADNSMP寄存器进行设置
1	为“1”时任意延长A/D采样时间，当由“1”变“0”后，再由ADNSMP继续控制采样时间

注意：转换停止状态(ADCS=0)下设置ADSMPWAIT=1，在(ADCS=1)时可改写ADSMPWAIT为“0”。

15.2.13 A/D测试寄存器(ADTES)

此寄存器用于设置A/D转换器的测试模式。

通过8位存储器操作指令设置ADTES寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-16: A/D测试寄存器(ADTES)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	ADTES3	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D运行模式
0	0	0	通常转换
0	0	1	0 码的自诊断测试
0	1	1	半码的自诊断测试
1	0	1	全码的自诊断测试
上述以外			禁止设置。

15.2.14 A/D状态寄存器(ADFLG)

此寄存器表示A/D转换器的状态。

通过8位存储器操作指令读ADFLG寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-17: A/D状态寄存器(ADFLG)的格式

复位值: 00H R

	7	6	5	4	3	2	1	0
ADFLG	0	0	0	ADFLG4	ADFLG3	ADFLG2	ADFLG1	ADFLG0

ADFLG4	A/D转换状态
0	单次转换模式时，A/D转换未结束
1	单次转换模式时，转换结束(2个ADCLK后自动清零)。 连续转换模式时ADFLG4保持1'b0

ADFLG3	A/D转换状态
0	非A/D转换结束前1个ADCLK
1	A/D转换结束前1个ADCLK(1个ADCLK后自动清零)

ADFLG2	A/D转换状态
0	非A/D转换结束前2个ADCLK
1	A/D转换结束前2个ADCLK(1个ADCLK后自动清零)

ADFLG1	A/D转换状态
0	非逐次比较期间
1	逐次比较期间

ADFLG0	A/D转换状态
0	非A/D采样期间
1	A/D采样期间

15.2.15 A/D充放电控制寄存器(ADNDIS)

此寄存器用来控制A/D转换器的充放电动作及时间。

通过8位存储器操作指令读写ADNDIS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-18: A/D充放电控制寄存器(ADNDIS)的格式

复位值: 00HW

	7	6	5	4	3	2	1	0
ADNDIS	0	0	0	ADNDIS4	ADNDIS3	ADNDIS2	ADNDIS1	ADNDIS0

ADNDIS[4]	充放电控制
1'b0	放电
1'b1	充电

ADNDIS[3:0]	充放电时间
4'b0000	不进行充、放电
4'b0010	2个ADCLK
4'b0011	3个ADCLK
4'b0100	4个ADCLK
4'b0101	5个ADCLK
4'b0110	6个ADCLK
.....
4'b1111	15个ADCLK

注: 禁止设置充放电时间为1个ADCLK, 即ADNDIS[3:0]=4 'b0001

15.2.16 控制模拟输入引脚端口功能的寄存器

必须设置与A/D转换器模拟输入复用的端口功能的控制寄存器(端口模式控制寄存器(PMCxx))。详细内容请参照“2.3.6端口模式控制寄存器(PMCxx)”。

在将ANI0~ANI15引脚用作A/D转换器的模拟输入时, 必须将各端口对应的端口模式控制寄存器(PMCxx)的位置“1”。

15.3 输入电压和转换结果

模拟输入引脚(ANI0~ANI15)的模拟输入电压和理论上的A/D转换结果(12位A/D转换结果寄存器(ADCR))有以下表达式的关系。

$$ADCR = \text{INT}\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right) \text{ or } (ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

INT(): 将括号中的数值的整数部分返回的函数

V_{AIN}: 模拟输入电压

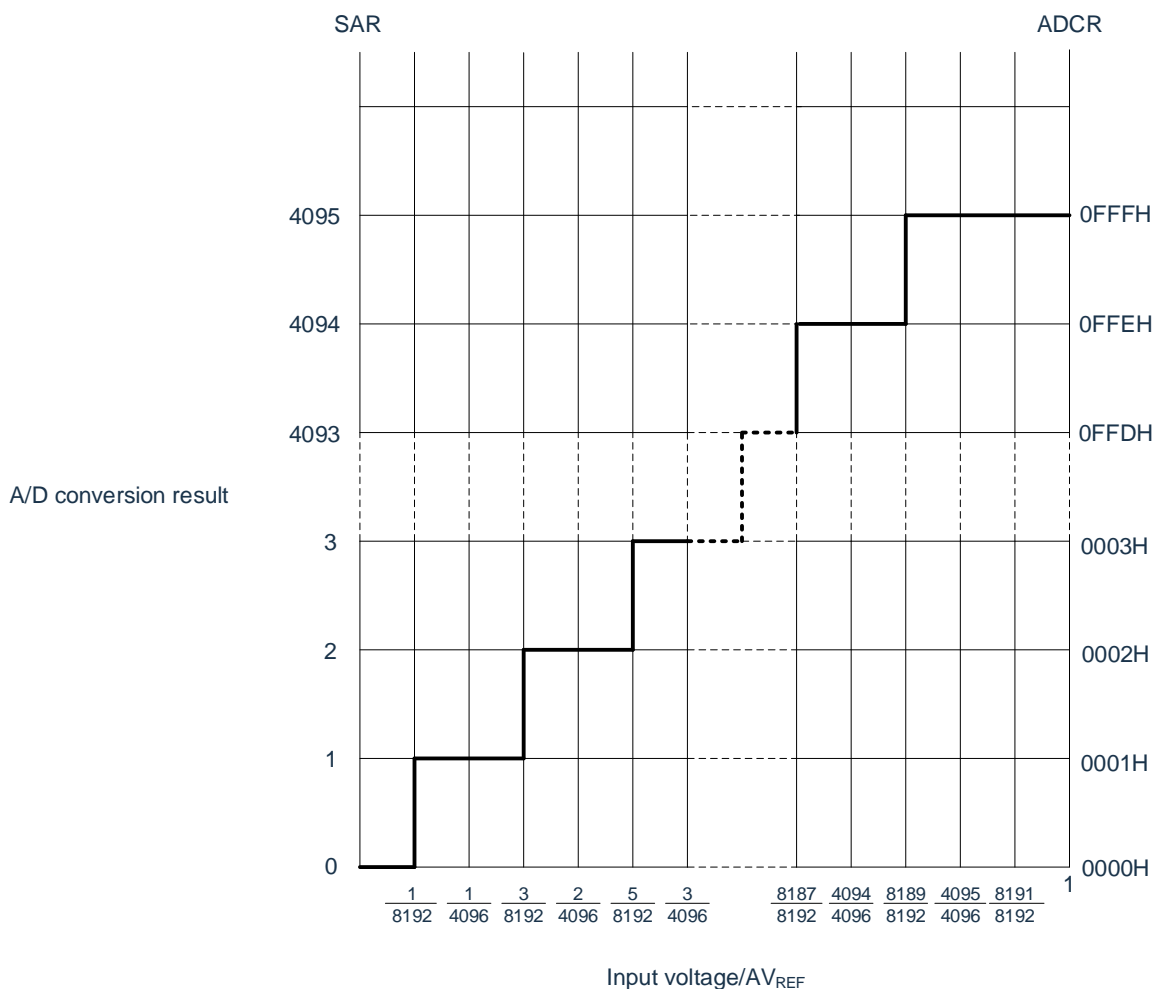
AV_{REF}: AVREF引脚电压

ADCR: A/D转换结果寄存器(ADCR)的值

SAR: 逐次逼近寄存器

模拟输入电压和A/D转换结果的关系如图15-19所示。

图15-19: 模拟输入电压和A/D转换结果的关系



备注: AV_{REF}是A/D转换器的正(+)基准电压, 可选择AV_{REFP}或者V_{DD}。

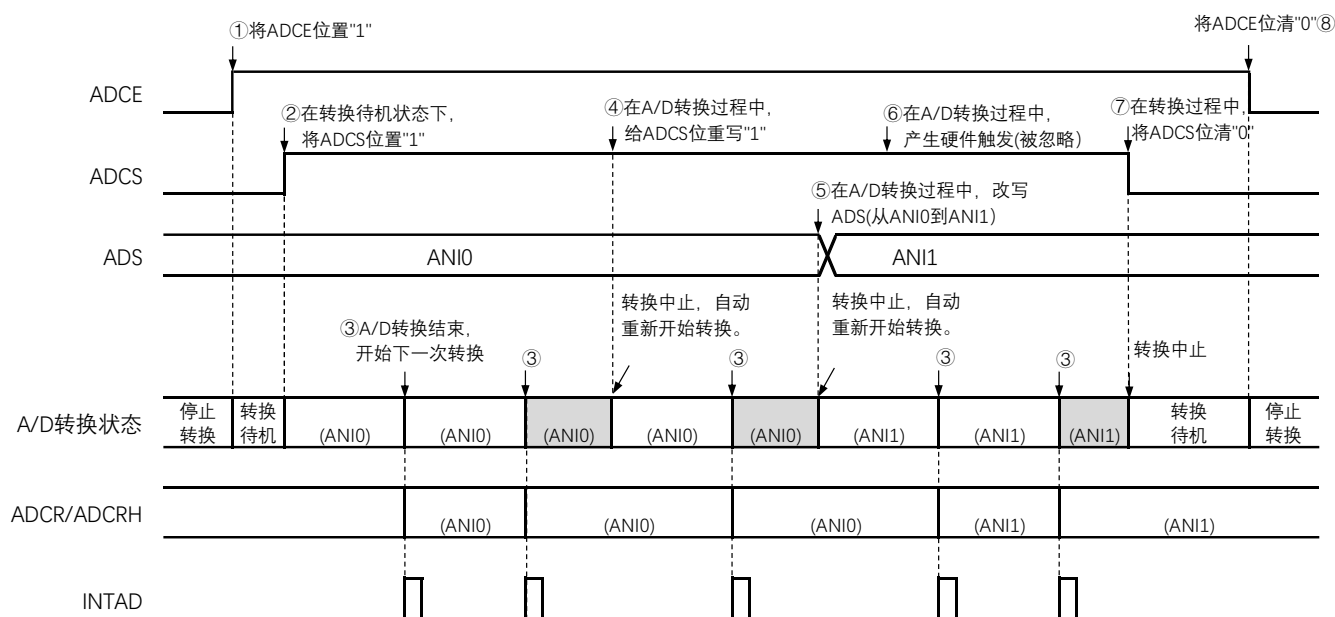
15.4 A/D转换器的运行模式

A/D转换器的各模式的运行如下所示。有关各模式的设置步骤，请参照“15.5A/D转换器的设置流程图”。

15.4.1 软件触发模式(选择模式、连续转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。在A/D转换结束后立即开始下一次A/D转换。
- ④ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对由ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑥ 即使在转换过程中输入硬件触发也不开始A/D转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

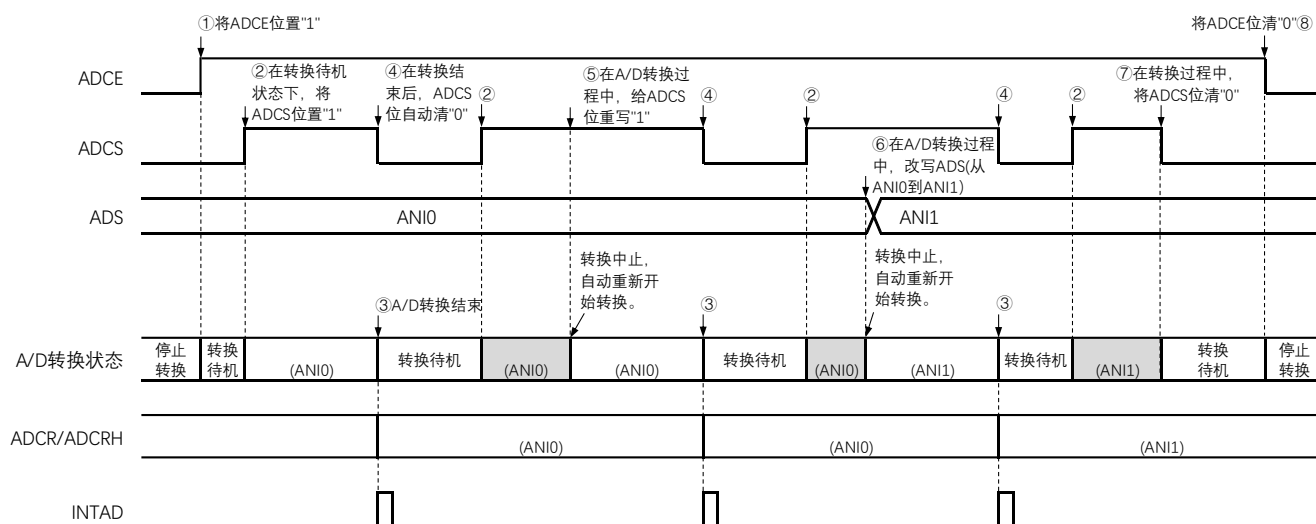
图15-20: 软件触发模式(选择模式、连续转换模式)的运行时序例子



15.4.2 软件触发模式(选择模式、单次转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。
- ④ 在A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。
- ⑤ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对由ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

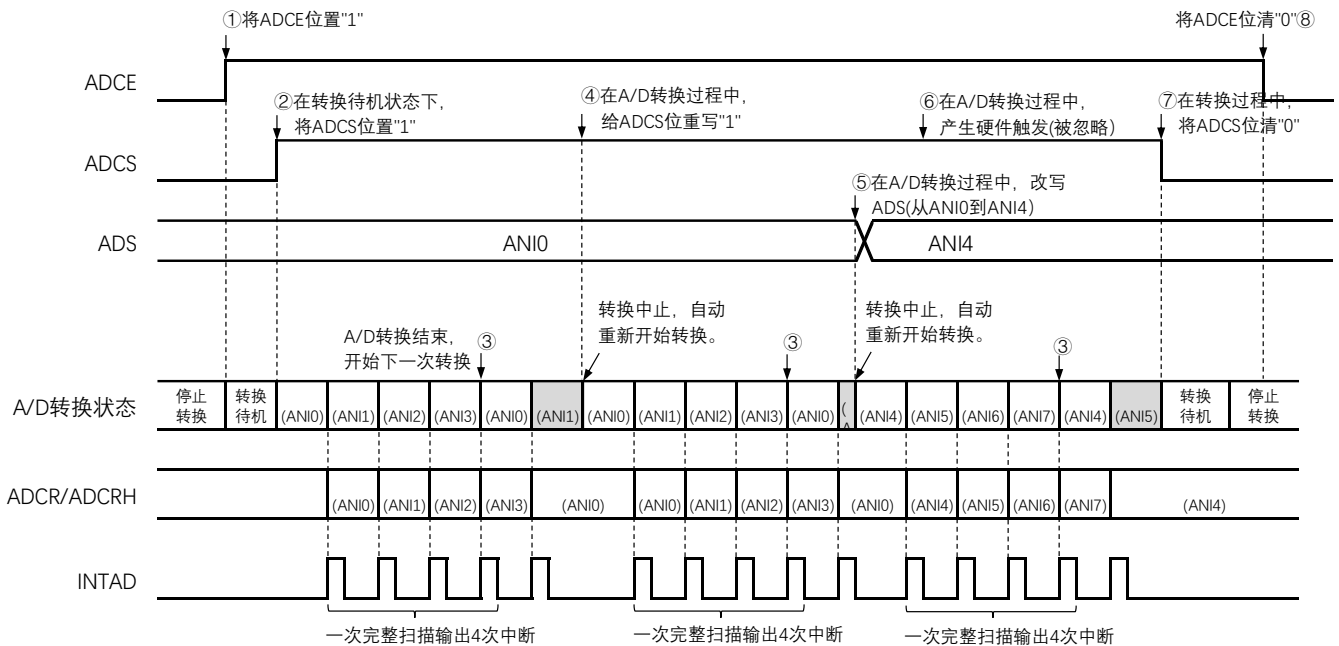
图15-21: 软件触发模式(选择模式、单次转换模式)的运行时序例子



15.4.3 软件触发模式(扫描模式、连续转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器(ADS)指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换(4个通道)。
- ④ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的最初通道进行A/D转换。
- ⑥ 即使在转换过程中输入硬件触发也不开始A/D转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

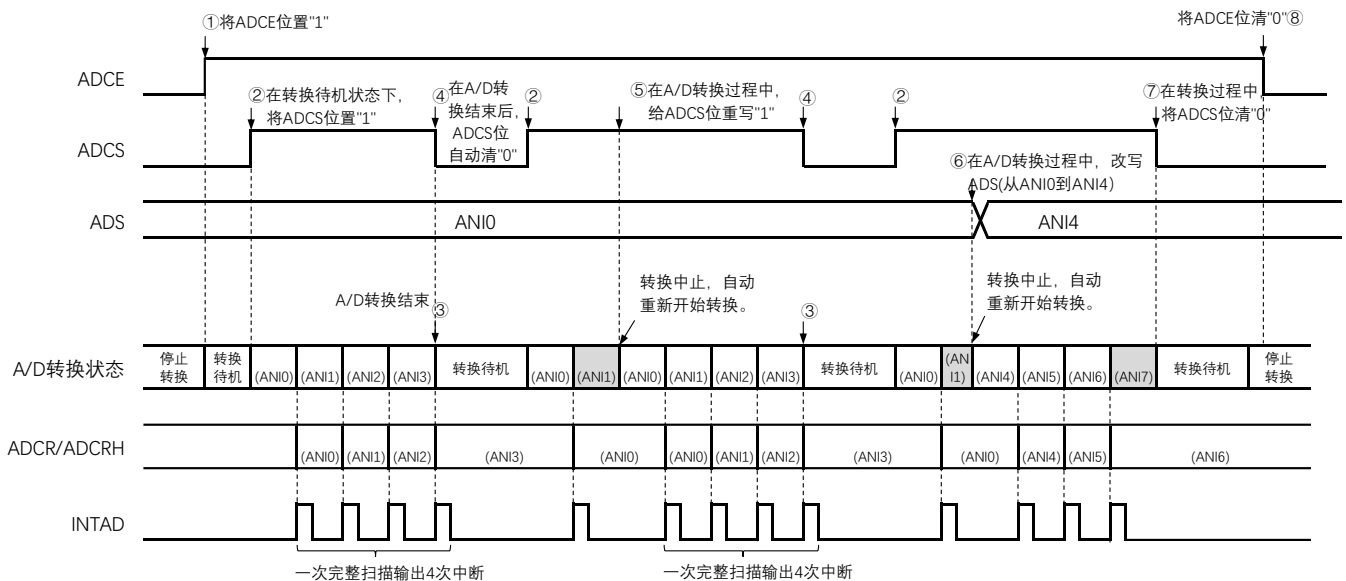
图15-22：软件触发模式(扫描模式、连续转换模式)的运行时序例子



15.4.4 软件触发模式(扫描模式、单次转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器(ADS)指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。
- ④ 在4个通道的A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。
- ⑤ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的最初通道进行A/D转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

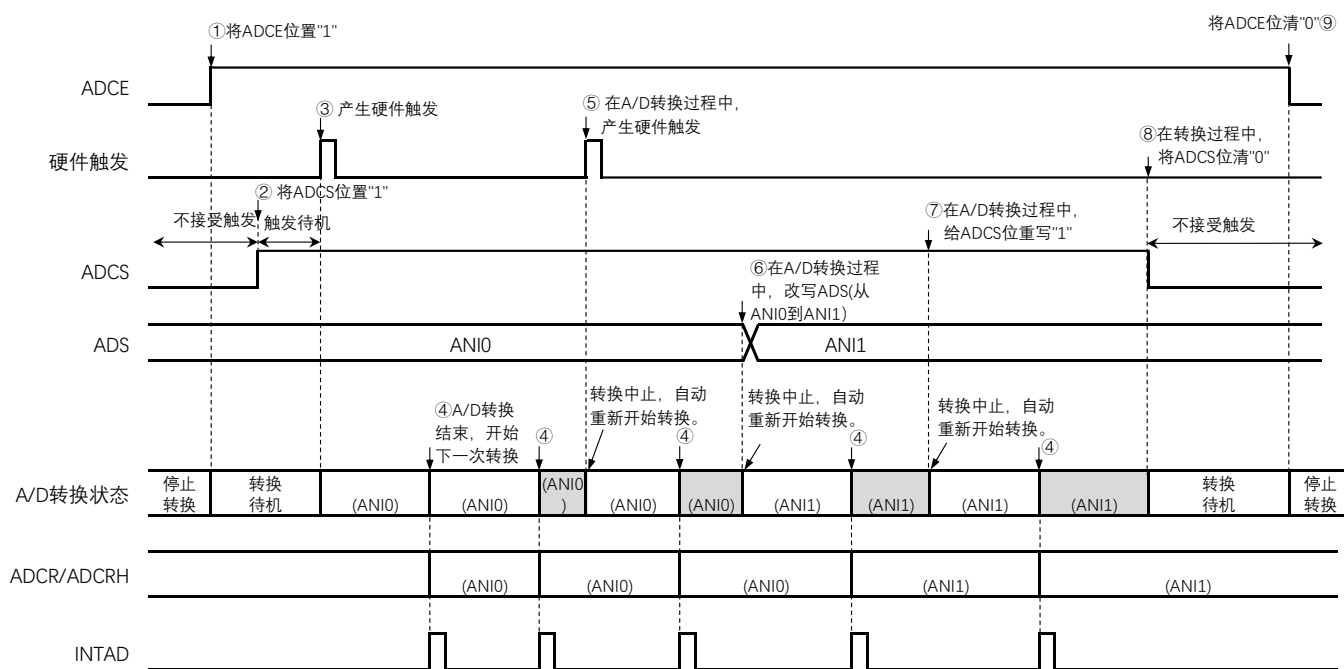
图15-23: 软件触发模式(扫描模式、单次转换模式)的运行时序例子



15.4.5 硬件触发无等待模式(选择模式、连续转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态(此阶段不开始转换)。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。
- ④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。在A/D转换结束后立即开始下一次A/D转换。
- ⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

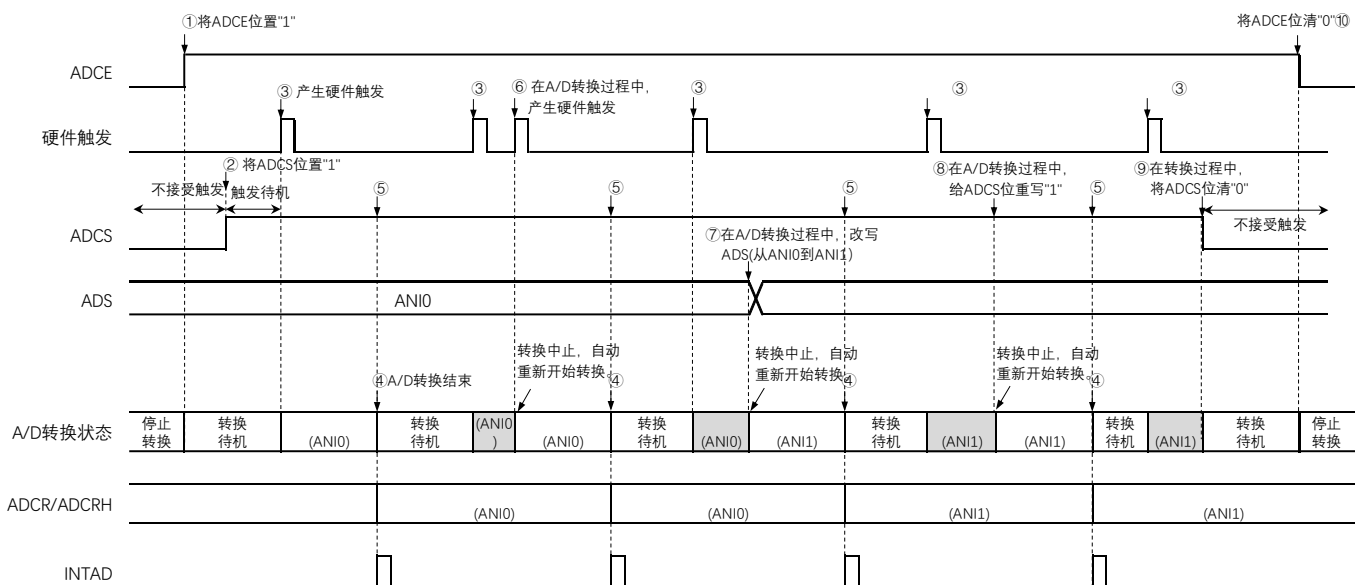
图15-24：硬件触发无等待模式(选择模式、连续转换模式)的运行时序例子



15.4.6 硬件触发无等待模式(选择模式、单次转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态(此阶段不开始转换)。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。
- ④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。
- ⑤ 在A/D转换结束后，ADCS位保持“1”的状态，进入A/D转换待机状态。
- ⑥ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。
- ⑦ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑧ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑨ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻停止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑩ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

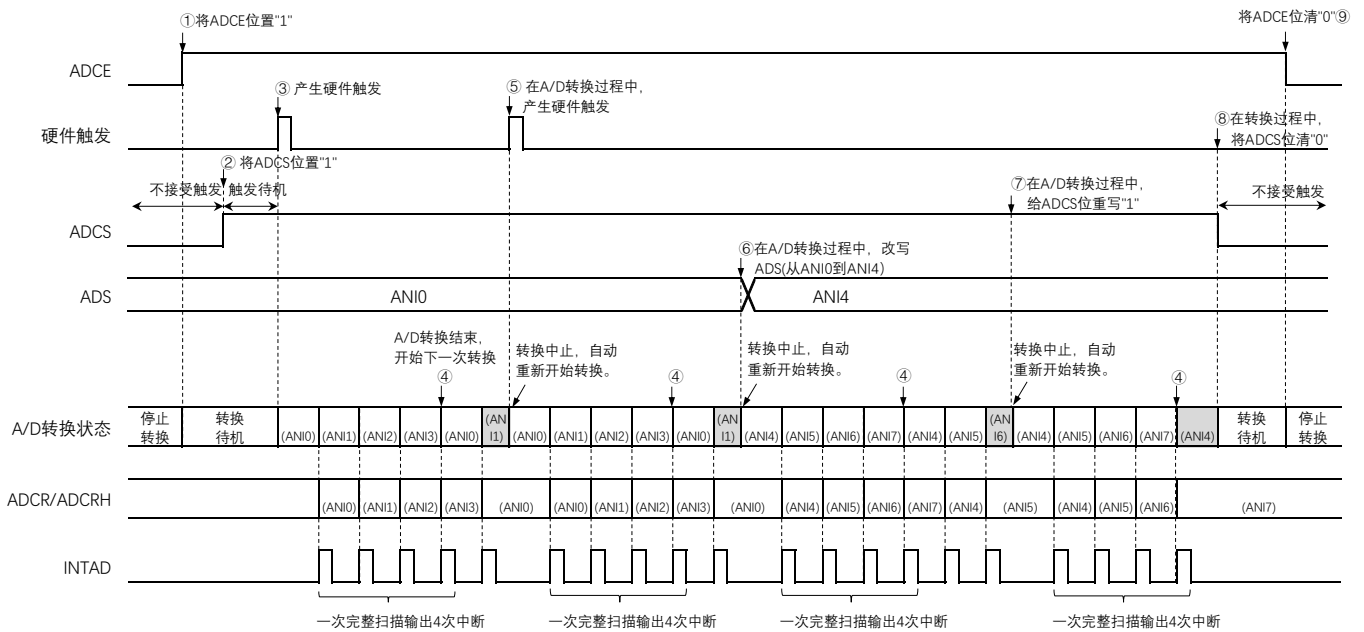
图15-25: 硬件触发无等待模式(选择模式、单次转换模式)的运行时序例子



15.4.7 硬件触发无等待模式(扫描模式、连续转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间(1us)进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态(此阶段不开始转换)。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ④ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换。
- ⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的通道进行A/D转换。
- ⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。
- ⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

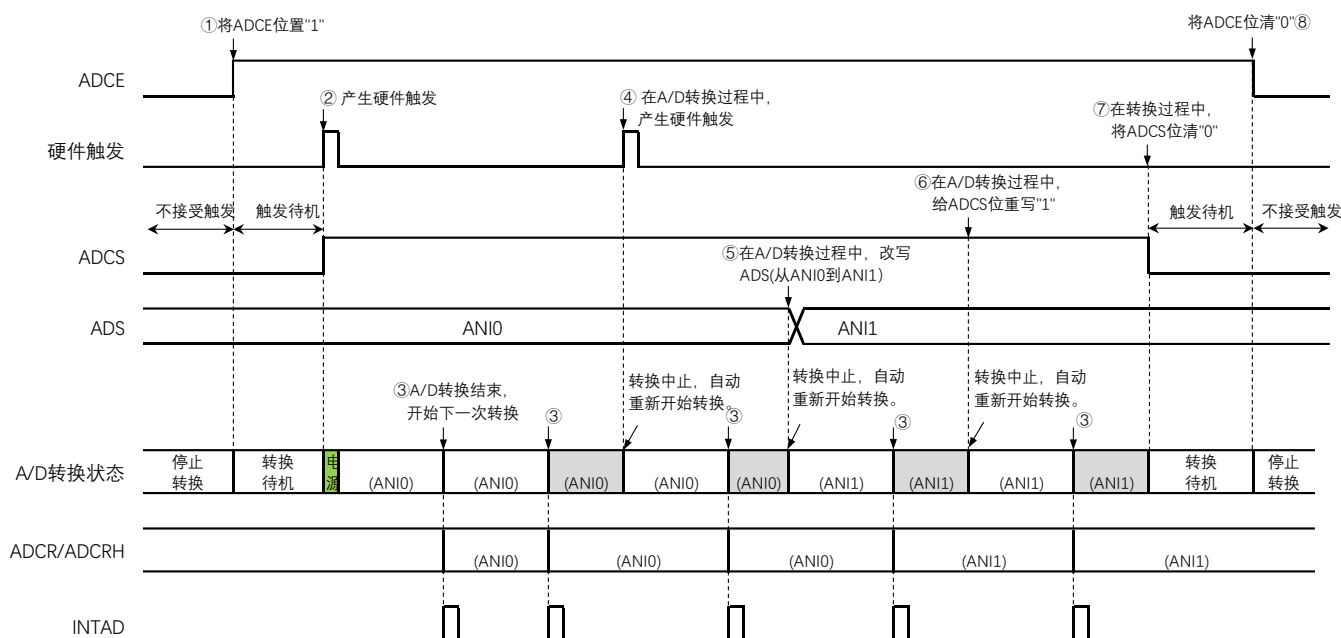
图15-26: 硬件触发无等待模式(扫描模式、连续转换模式)的运行时序例子



15.4.9 硬件触发等待模式(选择模式、连续转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。在A/D转换结束后立即开始下一次A/D转换(此时，不需要硬件触发)。
- ④ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑥ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图15-28：硬件触发等待模式(选择模式、连续转换模式)的运行时序例子

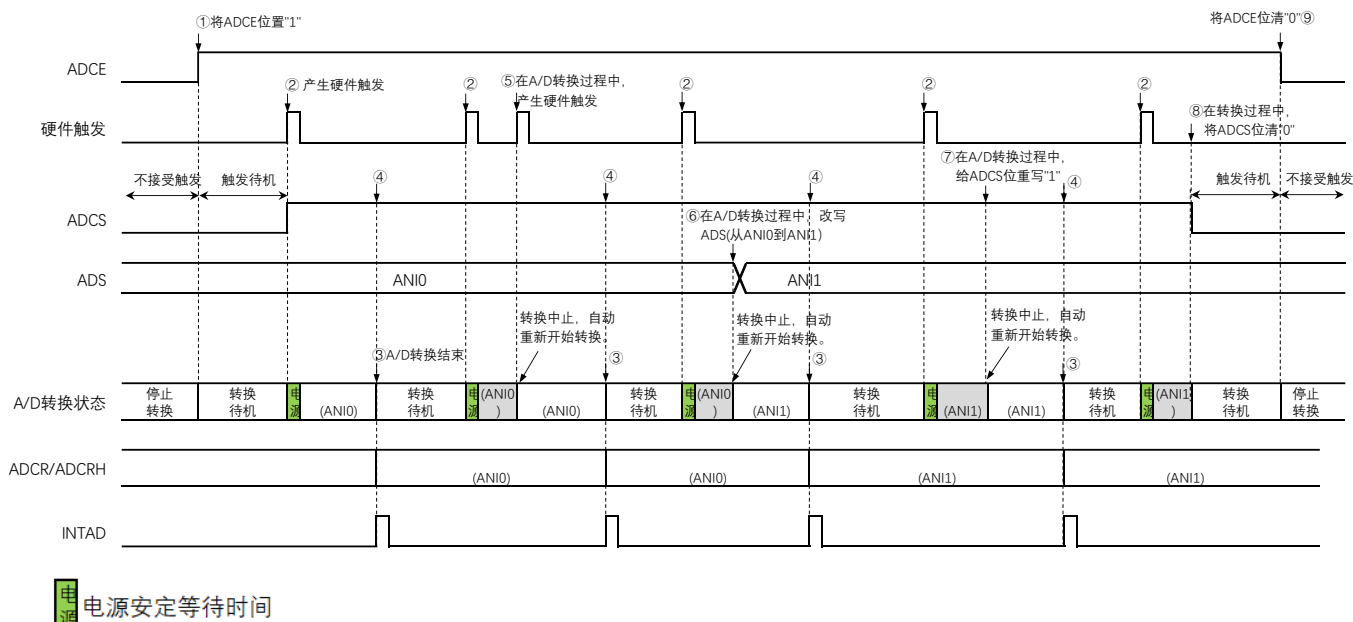


电源安定等待时间

15.4.10 硬件触发等待模式(选择模式、单次转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。
- ④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。
- ⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。
- ⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

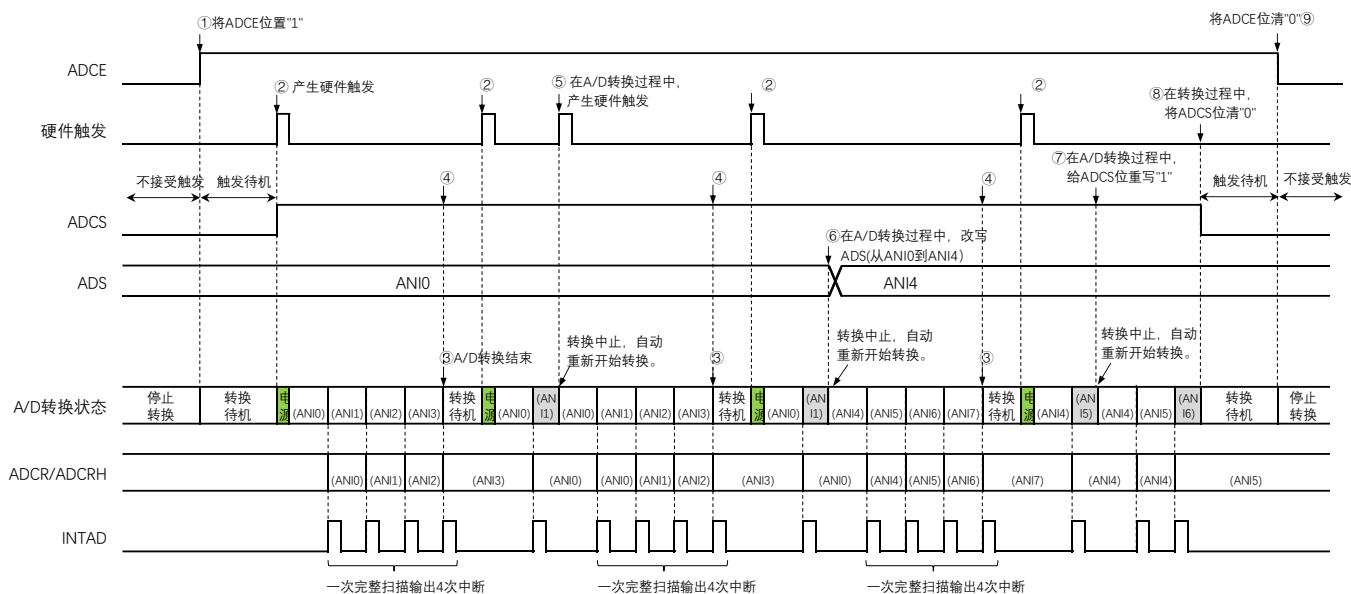
图15-29：硬件触发等待模式(选择模式、单次转换模式)的运行时序例子



15.4.12 硬件触发等待模式(扫描模式、单次转换模式)

- ① 在停止状态下，将A/D转换器的模式寄存器0(ADM0)的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器(ADS)指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。在输入硬件触发后自动将ADM0寄存器的ADCS位置“1”。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器(ADCR、ADCRH)，并且产生A/D转换结束中断请求信号(INTAD)。
- ④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新从最初的通道开始扫描转换。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的通道开始扫描转换。
- ⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后从最初的通道开始扫描转换。
- ⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图15-31：硬件触发等待模式(扫描模式、单次转换模式)的运行时序例子



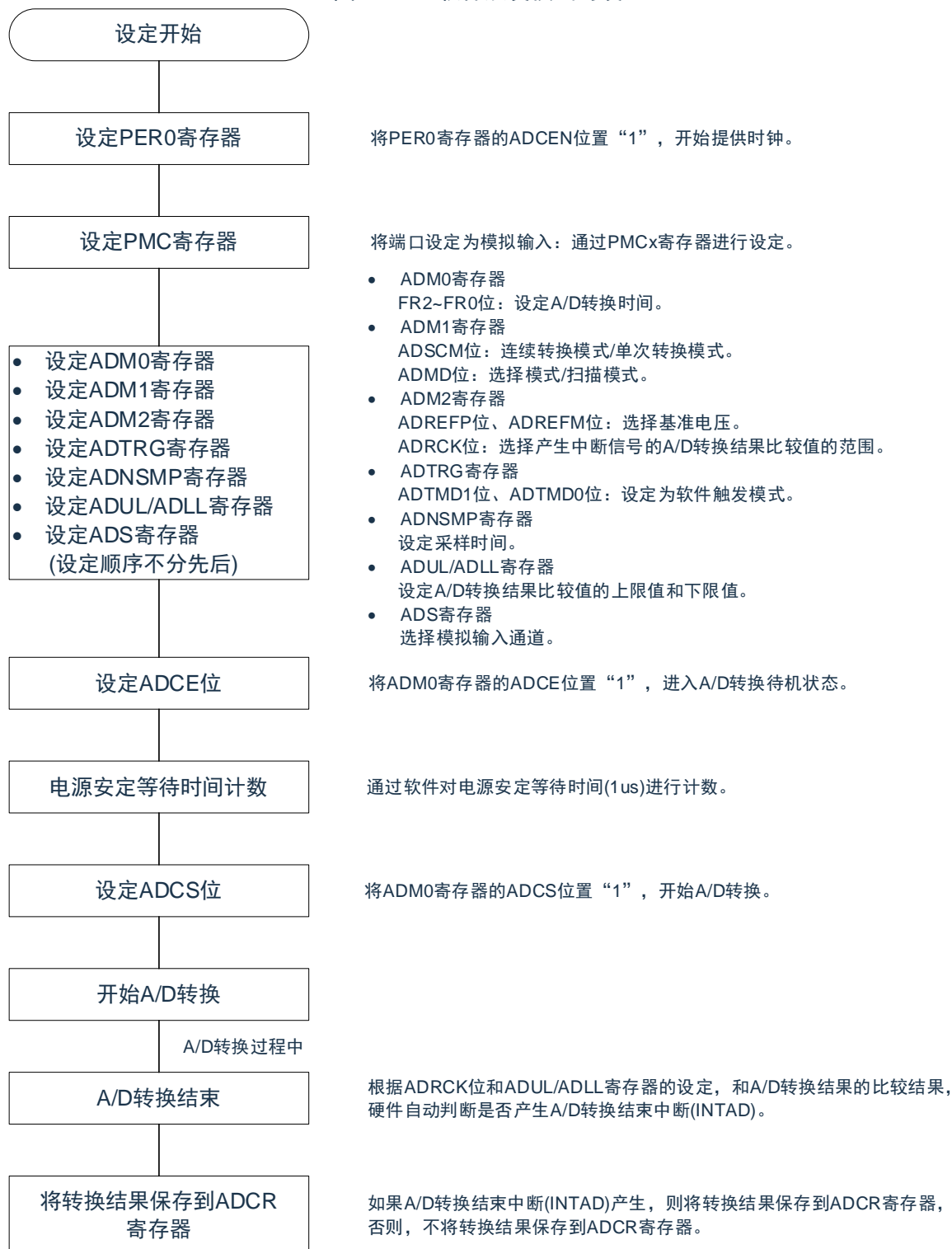
电源 电源安定等待时间

15.5 转换器的设置流程图

各运行模式的A/D转换器的设置流程图如下所示。

15.5.1 软件触发模式的设置

图15-32：软件触发模式的设置



15.5.2 硬件触发无等待模式的设置

图15-33: 硬件触发无等待模式的设置



15.5.3 硬件触发等待模式的设置

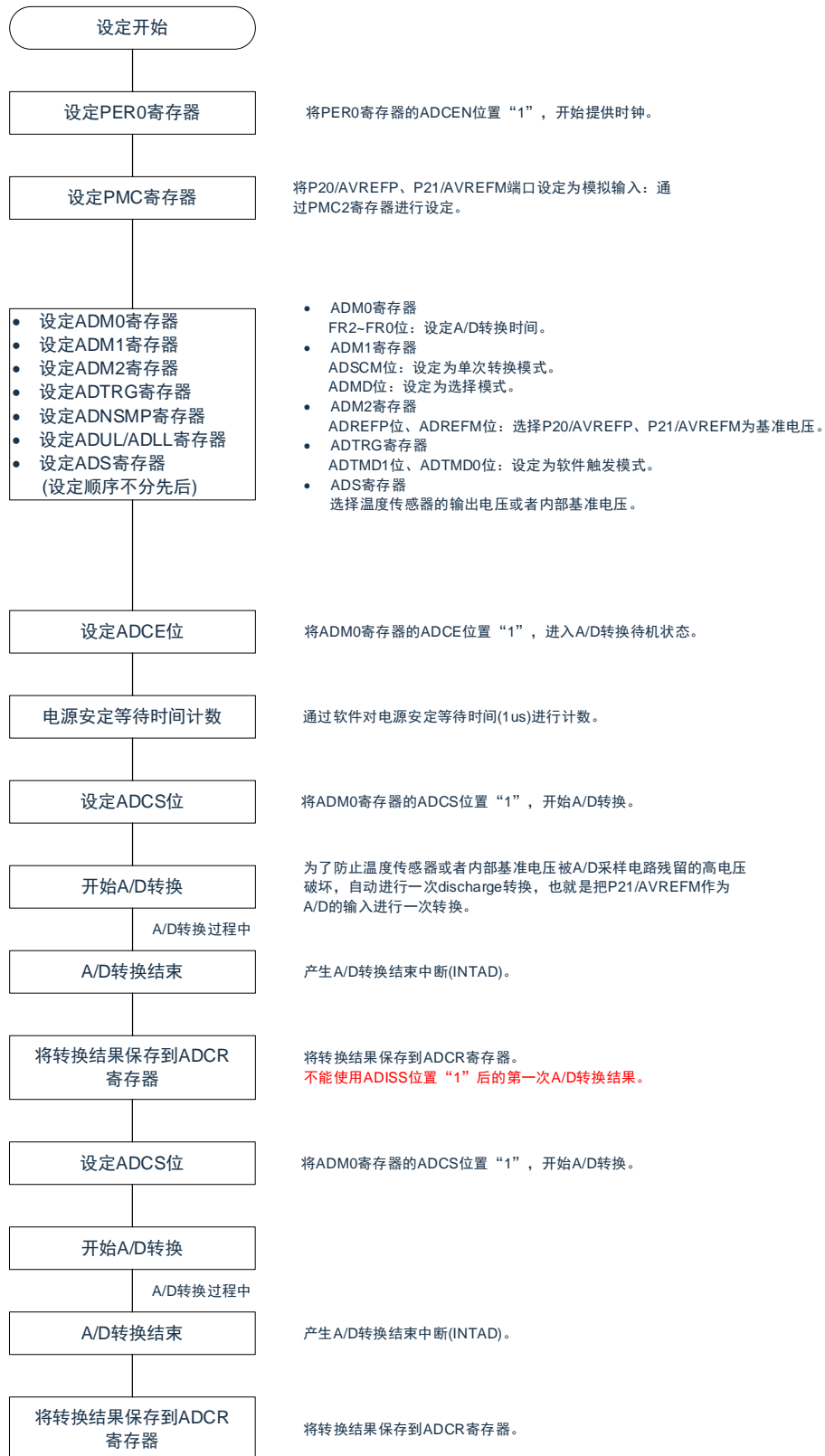
图15-34：硬件触发等待模式的设置



15.5.4 选择温度传感器的输出电压/内部基准电压时的设置

(以软件触发模式、单次转换模式为例)

图15-35：选择温度传感器的输出电压/内部基准电压时的设置



15.5.5 测试模式的设置

图15-36：测试模式的设置(VSS/half_VDD/VDD作为转换对象)



第16章 D/A转换器

D/A转换器的通道因产品而不同。

表16-1: D/A转换器的输出引脚

D/A输出引脚	64PIN	52PIN	48PIN	40PIN	36PIN	32PIN	24PIN
ANO0	○	○	○	○	○	○	○
ANO1	○	○	○	○	○	○	—

注意：本章的下述内容主要针对64引脚产品进行说明。

16.1 D/A转换器的功能

D/A转换器是将数字输入转换为模拟信号的8位分辨率的转换器，能控制2个通道(ANO0、ANO1)的模拟输出。

D/A转换器有以下功能：

- 8位分辨率×2ch
- R-2R梯形方式
- 模拟输出电压
 - 8位分辨率： $V_{DD} \times m8 / 256$ (m8：给DACSi寄存器设置的值)
- 运行模式
 - 通常模式
 - 实时输出模式

备注：i=0、1

16.3 控制D/A转换器的寄存器

通过以下寄存器控制D/A转换器。

- 外围允许寄存器1(PER1)
- D/A转换器的模式寄存器(DAM)
- D/A转换值设置寄存器0、1(DACS0、DACS1)
- 事件输出目标选择寄存器n(ELSELRn)、n=00~21

16.3.1 外围允许寄存器1(PER1)

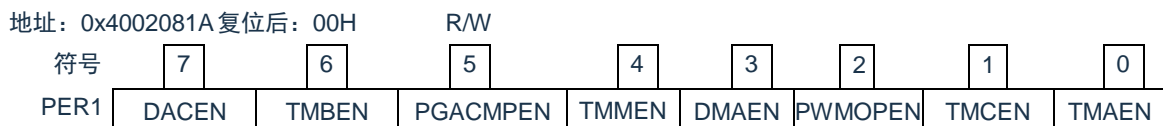
PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用D/A转换器时，必须将bit7(DACEN)置“1”。

通过8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图16-2：外围允许寄存器1(PER1)的格式



DACEN	D/A转换器的输入时钟的控制
0	停止提供输入时钟。 •不能写D/A转换器使用的SFR。 •D/A转换器处于复位状态。
1	提供输入时钟。 •能读写D/A转换器使用的SFR。

注意：要设置D/A转换器时，必须先将DACEN位置“1”。

当DACEN位为“0”时，忽视D/A转换器的控制寄存器的写操作，而且读取值都为初始值(端口模式寄存器2(PM2)和端口寄存器2P2)除外)。

16.3.2 D/A转换器的模式寄存器(DAM)

这是控制D/A转换器运行的寄存器。

通过8位存储器操作指令设置DAM寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图16-3: D/A转换器的模式寄存器(DAM)的格式

地址: 40044736H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEi	D/A转换器的转换运行的控制
0	停止D/A转换。
1	允许D/A转换。

DAMD _i	D/A转换器的运行模式的选择
0	通常运行模式
1	实时输出模式

备注: i=0、1

16.3.3 D/A转换值设置寄存器i(DACSi)(i=0、1)

这是设置在使用D/A转换器时输出到ANO0引脚和ANO1引脚的模拟电压值的寄存器。通过8位存储器操作指令设置DACSi寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图16-4: D/A转换值设置寄存器i(DACSi)(i=0、1)的格式

地址: 40044734H(DACS0)、40044735H(DACS1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

备注: D/A转换器的模拟输出电压(VANO_i)如下:

$$VANO_i = V_{DD} \times (DACSi) / 256$$

当不使用D/A转换器时，为了减小不必要的消费电流，必须将DACE_i位置“0”(禁止输出)并且将DACSi寄存器置“00H”，使R-2R的电阻没有电流流过。

16.3.4 事件输出目标选择寄存器n(ELSELRn)、n=00~21

当使用D/A转换器的实时输出模式时，以事件链接控制器的事件信号为启动触发，进行D/A转换。详细内容请参照“24.3.1 事件输出目标选择寄存器n(ELSELRn)(n=00~21)”。

16.3.5 控制模拟输入引脚端口功能的寄存器

必须设置与D/A转换器模拟输出复用的端口功能的控制寄存器(端口模式寄存器(PMxx)和端口模式控制寄存器(PMC))。详细内容请参照“2.3.1 端口模式寄存器(PMxx)”和“2.3.6 端口模式控制寄存器(PMCxx)”。

在将ANO0引脚和ANO1引脚用作D/A转换器的模拟输出时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”，并且通过端口模式控制寄存器(PMC)设置为模拟输出。

16.4 D/A转换器的运行

16.4.1 通常模式的运行

以DACSi寄存器的写操为启动触发，进行D/A转换。

其设置方法如下所示：

- ① 将PER1寄存器(外围允许寄存器1)的DACEN位置“1”，开始给D/A转换器提供输入时钟。
- ② 通过PMC寄存器(端口模式控制寄存器)将端口设置为模拟引脚。
- ③ 将DAM寄存器(D/A转换器的模式寄存器)的DAMDi位置“0”(通常模式)。
- ④ 给DACSi寄存器(D/A转换值设置寄存器i)设置ANOi引脚输出的模拟电压值。

以上①~④为初始设置。

- ⑤ 将DAM寄存器的DACEi位置“1”(允许D/A转换)。
开始D/A转换，在经过稳定时间后将④设置的模拟电压输出到ANOi引脚。
- ⑥ 此后，要进行D/A转换时，写DACSi寄存器。

在进行下一次D/A转换前，保持前一次D/A转换的结果。

如果将DAM寄存器的DACEi位置“0”(停止D/A转换)，就停止D/A转换。

注意：

1. 即使将DACEi位的设置值进行“1”→“0”→“1”的设置，也在经过最后置“1”后的稳定时间之后将DACSi寄存器设置的模拟电压输出到ANOi引脚。
2. 如果在稳定时间内改写DACSi寄存器，就中止转换并且以改写的值重新开始转换。

备注：i=0、1

16.4.2 实时输出模式的运行

D/A转换器的各通道以EVENTC的事件信号为启动触发，进行D/A转换。

其设置方法如下所示：

- ① 将PER1寄存器(外围允许寄存器1)的DACEN位置“1”，开始给D/A转换器提供输入时钟。
- ② 通过PMC寄存器(端口模式控制寄存器)将端口设置为模拟引脚。
- ③ 将DAM寄存器(D/A转换器的模式寄存器)的DAMDi位置“0”(通常模式)。
- ④ 给DACSi寄存器(D/A转换值设置寄存器i)设置ANOi引脚输出的模拟电压值。
- ⑤ 将DAM寄存器的DACEi位置“1”(允许D/A转换)。

开始D/A转换，在经过稳定时间后将③设置的模拟电压输出到ANOi引脚。

- ⑥ 通过事件输出目标选择寄存器n(ELSELRn、n=00~21)设置用于实时输出模式的触发信号。
- ⑦ 将DAM寄存器的DAMDi位置“1”(实时输出模式)。
- ⑧ 开始事件发生源的运行。

以上①~⑧为初始设置。

- ⑨ 此后，通过产生用于实时输出模式的触发信号，开始D/A转换，在经过稳定时间后将④设置的模拟电压输出到ANOi引脚。

在进行下一次D/A转换(产生用于实时输出模式的触发信号)前，给DACSi寄存器设置ANOi引脚输出的模拟电压值。

必须在进行下一次D/A转换(产生用于实时输出模式的触发信号)前给DACSi寄存器设置ANOi引脚输出的模拟电压值。

如果将DAM寄存器的DACEi位置“0”(停止D/A转换)，就停止D/A转换。

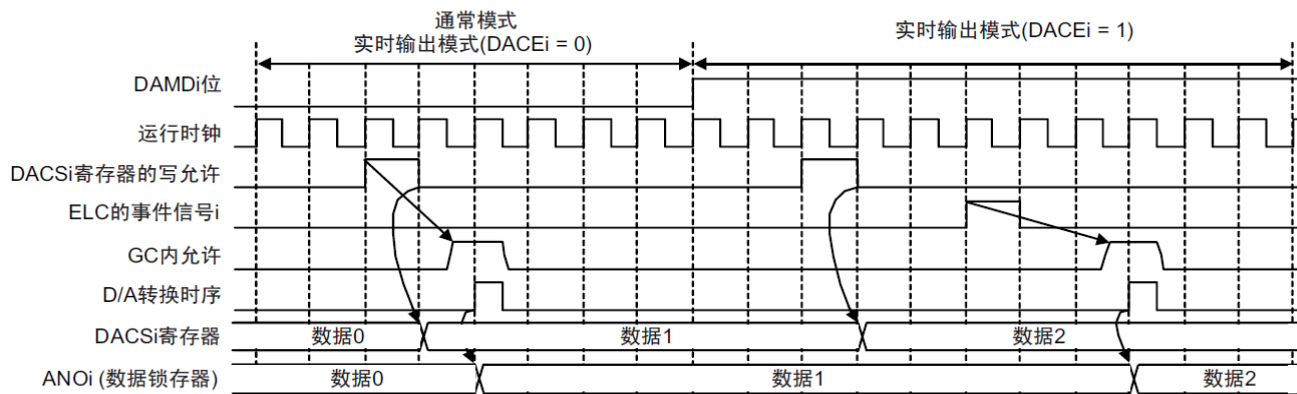
注意：

1. 即使将DACEi位的设置值进行“1”→“0”→“1”的设置，也在经过最后置“1”后的稳定时间之后将DACSi寄存器设置的模拟电压输出到ANOi引脚。
2. 用于相同通道实时输出模式的触发信号的产生间隔必须大于稳定时间。如果在稳定时间内产生用于实时输出模式的触发信号，就中止D/A转换并且重新开始转换。
3. 用于相同通道实时输出模式的触发信号的产生间隔必须大于3个F_{CLK}时钟。如果以小于等于3个F_{CLK}时钟的间隔连续产生启动触发，就只在产生第1个触发时进行D/A转换。

16.4.3 D/A转换值的输出时序

D/A转换值的输出时序如图16-5所示。

图16-5: D/A转换值输出时序图



备注: $i=0, 1$

- 通常运行模式和实时输出模式(不允许转换的情况)
在写DACSi寄存器的1个周期后(运行时钟)写数据锁存器(从ANOi引脚输出)。
- 实时输出模式(允许转换的情况)
在接受EVENTC事件信号后的3个周期后(运行时钟)写数据锁存器(从ANOi引脚输出)。

16.5 使用D/A转换器时的注意事项

使用D/A转换器时的注意事项如下所示。

- (1) 当通过PMC寄存器(端口模式控制寄存器)将端口设置为模拟引脚时, 和ANO0引脚、ANO1引脚复用的数字端口的输入/输出功能不工作。如果在通过ADPC寄存器将端口设置为模拟引脚的过程中读P2寄存器, 就在输入模式中读取值为“0”, 而在输出模式中读取值为P2的设置值。另外, 即使设置为数字输出模式, 也不将输出数据输出到引脚。
- (2) 在睡眠模式和深度睡眠模式中, D/A转换器继续运行。为了降低功耗, 必须将DACEi位清“0”, 并且在停止D/A转换后执行WFI指令。

备注: $i=0、1$

- (3) 在停止实时输出模式时(包括变为通常模式的情况)需要按照以下步骤进行:
 - 在停止触发输出源后至少等待3个时钟, 然后将DACEi位和DAMDi位置“0”。
 - 在将DACEi位和DAMDi位置“0”后将PER1寄存器的DACEN位置“0”(停止DAC)。
如果将DACEN位置“0”, 就清除DAC内部的全部寄存器。
因此, 在重新开始运行时, 需要设置各SFR。
- (4) 当允许D/A转换时, 不能对和ANO0引脚、ANO1引脚复用的模拟输入引脚进行A/D转换。
- (5) 在实时输出模式中, 必须在产生用于实时输出模式的触发信号前设置DACSi寄存器的值。
在触发信号有效期间, 不能更改DACSi寄存器的设置值。
- (6) 因为D/A转换器的输出阻抗高, 所以无法从ANO0引脚和ANO1引脚获得电流。在负载的输入阻抗低的情况下, 必须在负载和ANO0引脚、ANO1引脚之间插入跟踪放大器。另外, 必须尽量缩短跟踪放大器和负载之间的布线(因为输出阻抗高)。
如果布线太长, 就必须在布线的周围进行接地图形等的处理。
- (7) 如果要在实时输出模式有效时进入深度睡眠模式, 就必须在进入深度睡眠模式前禁止EVENTC的事件链接。

第17章 比较器

本产品内置2个通道的比较器。

17.1 比较器的功能

比较器有以下功能：

- CMP1的输入引脚可选择外部端口，内部基准电压以及内置DAC基准电压。
- 当电机停止时，可以通过比较U、V和W组合来检测电机位置，以实现无传感器电机控制。
- 通过切换一个比较器可以实现三相零交叉检测。
- 比较器0和比较器1的比较结果可以通过引脚输出(VCOUT0, VCOUT1)。

表17-1：比较器的功能概要

项目	内容
CMP	<ul style="list-style-type: none"> • 2通道比较器(CMP0和CMP1) • 比较器负端可以选择基准电压： CMP0的负端可选外部引脚输入，CMP0的内置基准电压和内部基准电压(1.45V) CMP1的负端可选外部引脚输入(4个)，CMP1的内置基准电压和内部基准电压(1.45V) • 负端的内部参考电压可设置(256steps) • CMP0的正端可选择PGA的输出 • CMP1的正端可以选择外部引脚输入(4个) • 正端输入电压>负端输入电压时，输出高电平 正端输入电压<负端输入电压时，输出低电平 • 数字滤波器的滤波宽度可选 • 输出反转功能 • 比较结果可以从引脚(VCOUT0, VCOUT1)输出 • 能检测比较器输出的有效边沿并且产生中断信号 • 与其他功能结合可检测电机初始位置，可控制高速/低速旋转 定时器的6相脉宽调制输出可在过流情况下设置/复位为高阻状态 • 与Timer4组合可输出TIMER WINDOW

17.2 比较器的结构

比较器的框图如图17-1所示。

图17-1：比较器0的框图

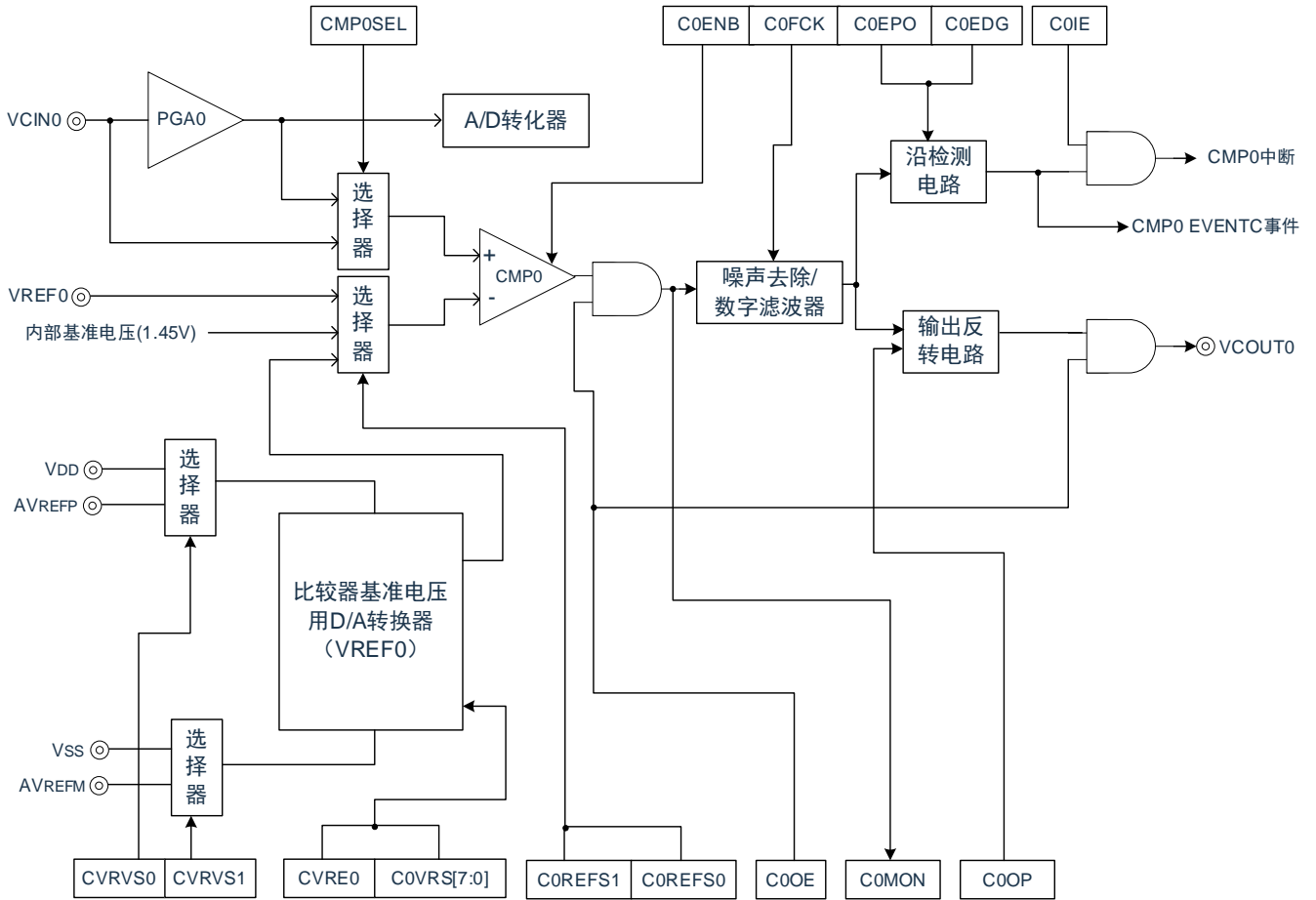
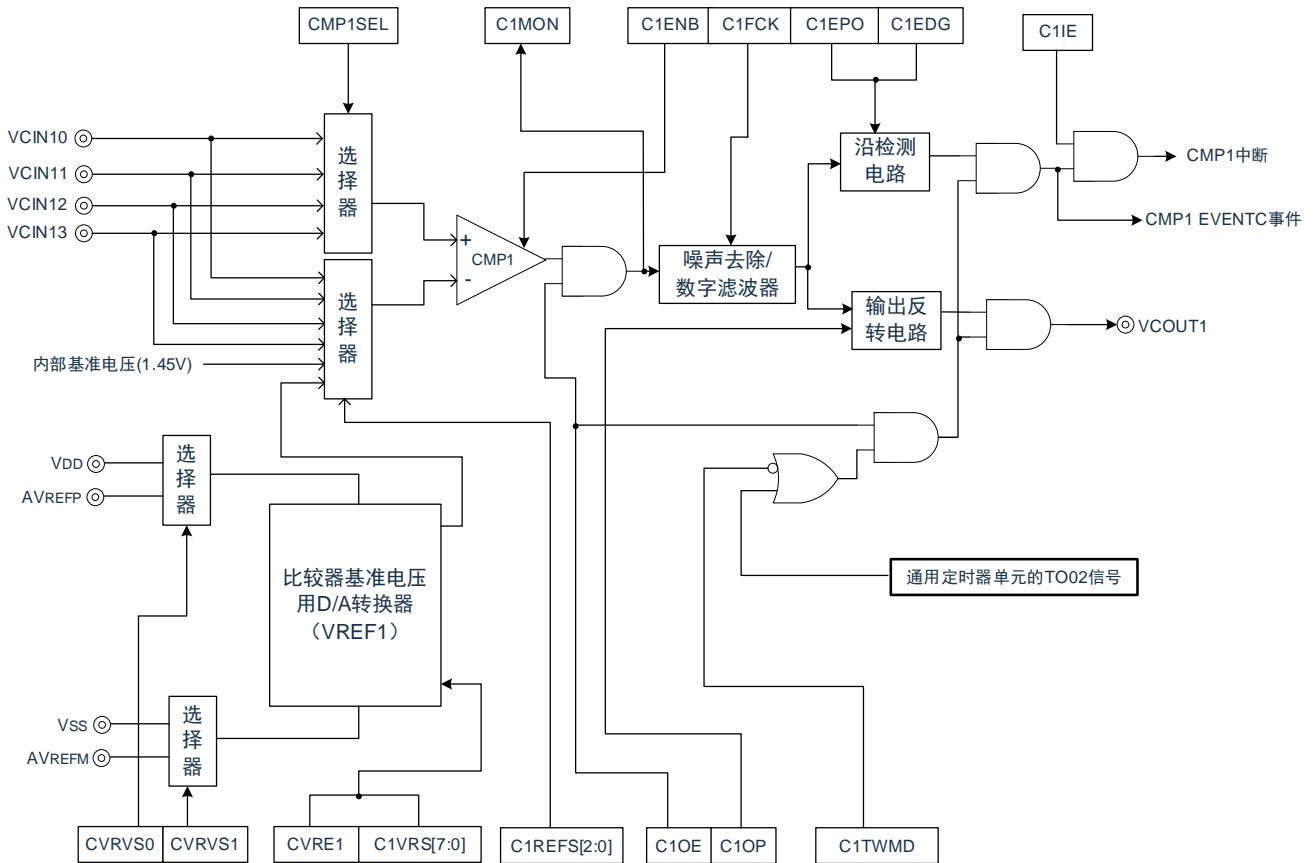


图17-2: 比较器1的框图



17.3 控制比较器的寄存器

控制比较器的寄存器如表17-2所示。

表17-2：控制比较器的寄存器

寄存器名	符号
外围允许寄存器1	PER1
比较器模式设置寄存器	COMPMDR
比较器滤波控制寄存器	COMPFIR
比较器输出控制寄存器	COMPOCR
比较器内置基准电压控制寄存器	CVRCTL
比较器内置基准电压选择寄存器0	C0RVM
比较器内置基准电压选择寄存器1	C1RVM
比较器0输入选择控制寄存器	CMPSEL0
比较器1输入选择控制寄存器	CMPSEL1
端口模式控制寄存器	PMCxx
端口模式寄存器	PMxx
端口寄存器	Pxx

17.3.1 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用比较器时，必须将bit5(PGACMPEN)置“1”。

通过8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图17-3：外围允许寄存器1(PER1)的格式

地址：0x4002081A 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMOPEN	TMCEN	TMAEN

PGACMPEN	比较器输入时钟的控制
0	停止提供输入时钟。 •不能写比较器使用的SFR。 •比较器处于复位状态。
1	提供输入时钟。 •能读写比较器使用的SFR。

注意：要设置比较器时，必须先将PGACMPEN位置“1”。

当PGACMPEN位为“0”时，忽视比较器的控制寄存器的写操作，并且读取值都为初始值(端口模式寄存器(PMxx)和端口寄存器(Pxx)除外)。

17.3.2 比较器模式设置寄存器(COMPMDR)

COMPMDR寄存器是设置比较器动作许可/禁止以及检测比较器输出的寄存器。

CiENB位在比较器输出许可(COMPOCR寄存器的CiOE位置“1”)时，禁止设置为“0”。

以下几种情况，禁止将CiENB位置“1”(i=0,1):

- CMP负端输入选择内置基准电压，而内置基准电压动作停止(CVRCTL寄存器的CVREi位为“0”)时
- CMP0的输入选择PGA的输出，而PGA动作停止(CVRCTL寄存器的CMPSEL0位为“1”且PGAEN寄存器的PGAEN位为“0”)时

通过8位存储器操作指令设置COMPMDR寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图17-4：比较器模式设置寄存器(COMPMDR)的格式

地址：40043840H 复位后：00H	R/W							
符号	7	6	5	4	3	2	1	0
COMPMDR	C1MON	0	0	C1ENB	C0MON	0	0	C0ENB

C1MON	比较器1的监视标志 ^{注1,2}
0	VCIN1 < 比较器1的基准电压，或者比较器1停止运行。
1	VCIN1 > 比较器1的基准电压

C1ENB	比较器1运行的允许
0	禁止比较器1的运行。
1	允许比较器1的运行。

C0MON	比较器0的监视标志 ^{注1,2}
0	VCIN0 < 比较器0的基准电压，或者比较器0停止运行。
1	VCIN0 > 比较器0的基准电压

C0ENB	比较器0运行的允许
0	禁止比较器0的运行。
1	允许比较器0的运行。

注1：在解除复位后立即变为“0”(初始值)，如果在允许比较器的运行后将C0ENB位和C1ENB位都置“0”，就为不定值。

注2：忽视此位的写入值。

17.3.3 比较器滤波控制寄存器(COMPfir)

COMPfir寄存器是数字滤波器的控制寄存器。通过8位存储器操作指令设置COMPfir寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图17-5：比较器滤波控制寄存器(COMPfir)的格式

地址：40043841H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
COMPfir	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	比较器1边沿检测的选择 ^{注1}
0	通过比较器1的单边沿检测产生中断请求。
1	通过比较器1的双边沿检测产生中断请求。

C1EPO	比较器1边沿极性的切换 ^{注1}
0	通过比较器1的上升沿产生中断请求。
1	通过比较器1的下降沿产生中断请求。

C1FCK1	C1FCK0	比较器1滤波器的选择 ^{注1}
0	0	比较器1没有滤波器。
0	1	比较器1有滤波器，通过F _{CLK} 进行采样。
1	0	比较器1有滤波器，通过F _{CLK} /8进行采样。
1	1	比较器1有滤波器，通过F _{CLK} /32进行采样。

C0EDG	比较器0边沿检测的选择 ^{注2}
0	通过比较器0的单边沿检测产生中断请求。
1	通过比较器0的双边沿检测产生中断请求。

C0EPO	比较器0边沿极性的切换 ^{注2}
0	通过比较器0的上升沿产生中断请求。
1	通过比较器0的下降沿产生中断请求。

C0FCK1	C0FCK0	比较器0滤波器的选择 ^{注2}
0	0	比较器0没有滤波器。
0	1	比较器0有滤波器，通过F _{CLK} 进行采样。
1	0	比较器0有滤波器，通过F _{CLK} /8进行采样。
1	1	比较器0有滤波器，通过F _{CLK} /32进行采样。

注1：如果更改C1FCK1~C1FCK0位、C1EPO位和C1EDG位，就可能产生比较器1的中断请求和向EVENTC输出的事件信号。必须在将EVENTC的ELSELR21寄存器(不链接比较器1的输出)置“0”后更改这些位。另外，必须将中断请求标志寄存器的IF清“0”。

如果将C1FCK1~C1FCK0位从“00B”(比较器1无滤波器)改为其他值(比较器1有滤波器)，就必须在更新滤波器的输出前经过4次采样后，使用比较器1的中断请求或者向EVENTC输出的事件信号。

注2：如果更改C0FCK1~C0FCK0位、C0EPO位和C0EDG位，就可能产生比较器0的中断请求和向EVENTC输出的事件信号。必须在将EVENTC的ELSELR20寄存器(不链接比较器0的输出)置“0”后更改这些位。另外，必须将中断请求标志寄存器的IF清“0”。

如果将C0FCK1~C0FCK0位从“00B”(比较器0无滤波器)改为其他值(比较器0有滤波器)，就必须在更新滤波器的输出前经过4次采样后，使用比较器0的中断请求或者向EVENTC输出的事件信号。

17.3.4 比较器输出控制寄存器(COMPOCR)

COMPOCR寄存器是设置比较器输出的极性，输出的许可/禁止以及中断输出的许可/禁止的控制寄存器。

以下的几种情况下，禁止将COMPOCR寄存器的CiOE位置“1”（输出许可）。(i=0,1)

- 比较器动作停止(COMPMDR寄存器的CiENB位为“0”)时
- CMP负端输入选择内置基准电压，而内置基准电压动作停止(CVRCTL寄存器的CVREi位为“0”)时
- CMP0的输入选择PGA的输出，而PGA动作停止(CVRCTL寄存器的CMPSEL0位为“1”且PGAEN寄存器的PGAEN位为“0”)时

通过8位存储器操作指令设置COMPOCR寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图17-6：比较器输出控制寄存器(COMPOCR)的格式

地址：40043842H 复位后：00H R/W

	7	6	5	4	3	2	1	0
COMPOCR	C1OTWMD	C1OP	C1OE	C1IE	0	COOP	COOE	COIE
C1OTWMD	比较器1的TIMER WINDOW输出模式控制位 ^{注1}							
0	比较器1普通输出模式(由C1OE控制)							
1	比较器1TIMER WINDOW输出模式(由TO02和C1OE共同控制)							
C1OP	VCOUT1输出极性的选择							
0	从VCOUT1进行比较器1的输出。							
1	从VCOUT1进行比较器1的反相输出。							
C1OE	VCOUT1引脚输出的允许 ^{注2}							
0	禁止比较器1的VCOUT1引脚输出。							
1	允许比较器1的VCOUT1引脚输出。							
C1IE	比较器1中断请求的允许 ^{注3}							
0	禁止比较器1的中断请求。							
1	允许比较器1的中断请求。							
COOP	VCOUT0输出极性的选择							
0	从VCOUT0进行比较器0的输出。							
1	从VCOUT0进行比较器0的反相输出。							
COOE	VCOUT0引脚输出的允许 ^{注4}							
0	禁止比较器0的VCOUT0输出到引脚。							
1	允许比较器0的VCOUT0输出到引脚 ^{注4,注8} 。							
COIE	比较器0中断请求的允许 ^{注5}							
0	禁止比较器0的中断请求。							
1	允许比较器0的中断请求。							

- 注1: 比较器1使用TIMER WINDOW模式时, 必须将寄存器COMPFIR的bit7(C1EDG)置为“1”。C1OE和C1OTWMD位不能同时设置, 要先设置C1OTWMD位之后, 再将C1OE位置为“1”。
- 注2: 当改写C1OE位时, 可能会产生比较器1中断请求和EVENTC事件。请在将EVENTC的ELSELR21寄存器设置为0(不链接到比较器1的输出)后改写此位。此外, 在改写C1OE位后请将中断请求标志寄存器的IF位初始化(无中断请求)。
- 注3: 比较器1的结果输出到引脚时, 必须将PIOR3寄存器的bit2(PIOR32)置为“1”。
- 注4: 当改写C0OE位时, 可能会产生比较器0中断请求和EVENTC事件。请在将EVENTC的ELSELR20寄存器设置为0(不链接到比较器0的输出)后改写此位。此外, 在改写C0OE位后请将中断请求标志寄存器的IF位初始化(无中断请求)。
- 注5: 设置C0OE位, C0OP位, 将比较器9的结果输入到PWM选项单元, 来控制PWM输出的强制切断。
- 注6: 如果将C1IE从“0”(禁止中断请求)改为“1”(允许中断请求), 中断请求标志寄存器的IF就可能变为“1”(有中断请求), 因此必须在将中断请求标志寄存器的IF清“0”后使用中断。
- 注7: 比较器0的结果输出到引脚时, 必须将PIOR3寄存器的bit1(PIOR31)置为“1”。
- 注8: 如果将C0IE从“0”(禁止中断请求)改为“1”(允许中断请求), 中断请求标志寄存器的IF就可能变为“1”(有中断请求), 因此必须在将中断请求标志寄存器的IF清“0”后使用中断。

17.3.5 比较器内置基准电压控制寄存器(CVRCTL)

CVRCTL寄存器是设置比较器的内置基准电压允许/停止动作的寄存器。

通过8位存储器操作指令设置CVRCTL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注意：在内置基准电压停止动作(CVREi=0)时，改写CVRCTL寄存器的CVRVSi位。

图17-7：比较器内置基准电压控制寄存器(CVRCTL)的格式

地址：40043843H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CVRCTL	0	0	CVRE1	CVRVS1	0	0	CVRE0	CVRVS0

CVRE1	内置基准电压1的控制位
0	禁止内置基准电压1的运行
1	允许内置基准电压1的运行

CVRVS1	内置基准电压的ground端选择位
0	内置基准电压的ground端选择V _{SS}
1	内置基准电压的ground端选择AV _{REFM} ^{注1}

CVRE0	内置基准电压0的控制位
0	禁止内置基准电压0的运行
1	允许内置基准电压0的运行

CVRVS0	内置基准电压的电源端选择位
0	内置基准电压的电源端选择V _{DD}
1	内置基准电压的电源端选择AV _{REFP} ^{注2}

注1：P21引脚同时兼用AVREFM和VCIN13，所以P21引脚作为CMP1的输入信号时，禁止将CVRVS1位置“1”。

注2：P20引脚同时兼用AVREFP和VCIN12，所以P20引脚作为CMP1的输入信号时，禁止将CVRVS0位置“1”。

17.3.6 比较器内置基准电压选择寄存器(CiRVM)

CiRVM寄存器是设置比较器的内置基准电压的寄存器。

在内置基准电压停止动作(CVREi=0)时, 改写CiRVM寄存器

通过8位存储器操作指令设置CVRCTL寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图17-8: 比较器内置基准电压选择寄存器i(CiRVM)的格式

地址: 400438434H(C0RVM), 400438435H(C1RVM), 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CiRVM	CiRVS7	CiRVS6	CiRVS5	CiRVS4	CiRVS3	CiRVS2	CiRVS1	CiRVS0

CiRVS7	CiRVS6	CiRVS5	CiRVS4	CiRVS3	CiRVS2	CiRVS1	CiRVS0	比较器的内置基准电压的设置
0	0	0	0	0	0	0	0	$\{(AV_{REFP}或V_{DD})/256\}x0$
0	0	0	0	0	0	0	1	$\{(AV_{REFP}或V_{DD})/256\}x1$
0	0	0	0	0	0	1	0	$\{(AV_{REFP}或V_{DD})/256\}x2$
0	0	0	0	0	0	1	1	$\{(AV_{REFP}或V_{DD})/256\}x3$
...								...
1	1	1	1	1	1	0	0	$\{(AV_{REFP}或V_{DD})/256\}x252$
1	1	1	1	1	1	0	1	$\{(AV_{REFP}或V_{DD})/256\}x253$
1	1	1	1	1	1	1	0	$\{(AV_{REFP}或V_{DD})/256\}x254$
1	1	1	1	1	1	1	1	$\{(AV_{REFP}或V_{DD})/256\}x255$

17.3.7 比较器0的输入信号选择控制寄存器(CMPSEL0)

CMPSEL0寄存器是比较器0的正端，负端的输入信号的选择寄存器。

在比较器0停止动作(C0ENB=0)时，改写CMPSEL0寄存器。

通过8位存储器操作指令设置CMPSEL0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图17-9：比较器0的输入信号选择控制寄存器(CMPSEL0)的格式

地址：4004384AH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CMPSEL0	CMP0SEL	0	0	0	0	0	C0REFS1	C0REFS0

CMP0SEL	比较器0的正端输入信号选择位
0	选择外部引脚(VCIN0引脚)
1	选择PGA输出信号

C0REFS1	C0REFS0	比较器0的负端输入信号选择位
0	0	选择内置基准电压VREF0
0	1	选择内部基准电压(1.45V)
1	0	选择外部引脚(IVREF0引脚)
1	1	禁止设置

17.3.8 比较器1的输入信号选择控制寄存器(CMPSEL1)

CMPSEL1寄存器是比较器1的正端，负端的输入信号的选择寄存器。

在比较器1停止动作(C1ENB=0)时，改写CMPSEL1寄存器。

通过8位存储器操作指令设置CMPSEL1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图17-10：比较器1的输入信号选择控制寄存器(CMPSEL1)的格式

地址：4004384BH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CMPSEL1	CMP1SEL1	CMP1SEL0	0	0	0	C0REFS1	C0REFS1	C0REFS0

CMP1SEL1	CMP1SEL0	比较器1的正端输入信号选择位
0	0	选择外部引脚(VCIN10引脚)
0	1	选择外部引脚(VCIN11引脚)
1	0	选择外部引脚(VCIN12引脚)
1	1	选择外部引脚(VCIN13引脚)

C0REFS2	C0REFS1	C0REFS0	比较器1的负端输入信号选择位
0	0	0	选择内置基准电压VREF1
0	0	1	选择内部基准电压(1.45V)
0	1	0	选择外部引脚(VCIN10引脚)
0	1	1	选择外部引脚(VCIN11引脚)
1	0	0	选择外部引脚(VCIN12引脚)
1	0	1	选择外部引脚(VCIN13引脚)
1	1	0	禁止设置
1	1	1	禁止设置

注意：切换CMP1的模拟输入时，为了防止两个输入信号之前的贯通电流，切换间隔必须在3us以上。

17.3.9 控制模拟输入引脚端口功能的寄存器

在将VCIN0引脚、VCIN10-VCIN13引脚和VREF0引脚引脚用作比较器的模拟输入时，必须将各端口对应的端口模式寄存器(PMxx)的位和端口控制寄存器(PMCxx)的位置“1”。

在使用VCOUT0和VCOUT1功能时，必须设置与对象通道复用的端口功能的控制寄存器(端口模式寄存器(PMxx)和端口寄存器(Pxx))以及外围IO重定向寄存器(PIOR2,PIOR3)。详细内容请参照“2.3.1端口模式寄存器(PMxx)”，“2.3.2端口寄存器(Pxx)”，“2.3.9外围I/O重定向寄存器2(PIOR2)”和“2.3.10外围I/O重定向寄存器3(PIOR3)”

17.4 运行说明

比较器0和比较器1能各自独立运行。设置方法和运行相同。

CMP0和PGA0可以组合起来联动。

比较器的独立运行和联动的设置步骤如表17-3所示。

表17-3: 比较器相关寄存器的设置步骤

步骤	寄存器	位	设置值
1	PGACTL	PGAVG0/1/2	选择增益 ^{注3}
2	PGACTL	PVRVS	0(Vss引脚选择) ^{注3}
3	PGACTL	PGAEN	1(允许运行) ^{注3}
4	等待PGA的稳定时间(最小10us)		
5	COMPSELi	CMP0SEL/CMP1SELi 1	比较器i正端输入选择
6	COMPSELi	CiREFS	比较器i负端输入选择
7	CiRVM	CiRVSn	设置内置基准电压的值
8	CVRCTL	CVRVSi	选择内置基准电压的电源和地
9	CVRCTL	CVREi	1(内置基准电压i允许运行)
10	等待基准电压的稳定时间(最小20us)		
11	将VCIN0、VCIN1x、IVREF0引脚(输入), PGAI(输入) ^{注3} 设置为模拟输入功能。 PMCxx VCIN0引脚,VCi和IVREFi引脚的功能选择将PMCxx位置“1”(模拟输入)。 将PMxx位置“1”(输入模式)。		
12	COMPMDR	CiENB	1(允许运行)
13	等待比较器的稳定时间(最小3us)		
14	COMPFIR	CiFCK	使用或者不使用数字滤波器, 选择采样时钟。
		CiEOP、CiEDG	选择中断请求的边沿检测条件(上升沿、下降沿或者双边沿)。
15	COMPOCR	CiOP、CiOE	设置VCOUTi的输出(选择极性, 设置允许或者禁止输出)。 请参照“17.4.4 比较器i的输出(i=0、1)”。
		CiIE	设置允许或者禁止中断请求的输出。 请参照“17.4.4 比较器i的输出(i=0、1)”。
		C1OTWMD	设置比较器1的TIMER WINDOW输出许可/禁止
16	MKxx ^{注1}	MKL	使用中断时: 选择屏蔽中断。
17	IFxx ^{注1}	IFL	使用中断时: 0(无中断请求: 初始化) ^{注2}

注1: MKxx, IFxx是比较器的中断控制寄存器, 详细内容请参考“第25章 中断功能”。

注2: 比较器设置后, 到运行稳定期间, 可能会产生不需要的中断请求, 必须将中断请求标志位初始化。

注3: 比较器0和PGA联动时, 必须要设置

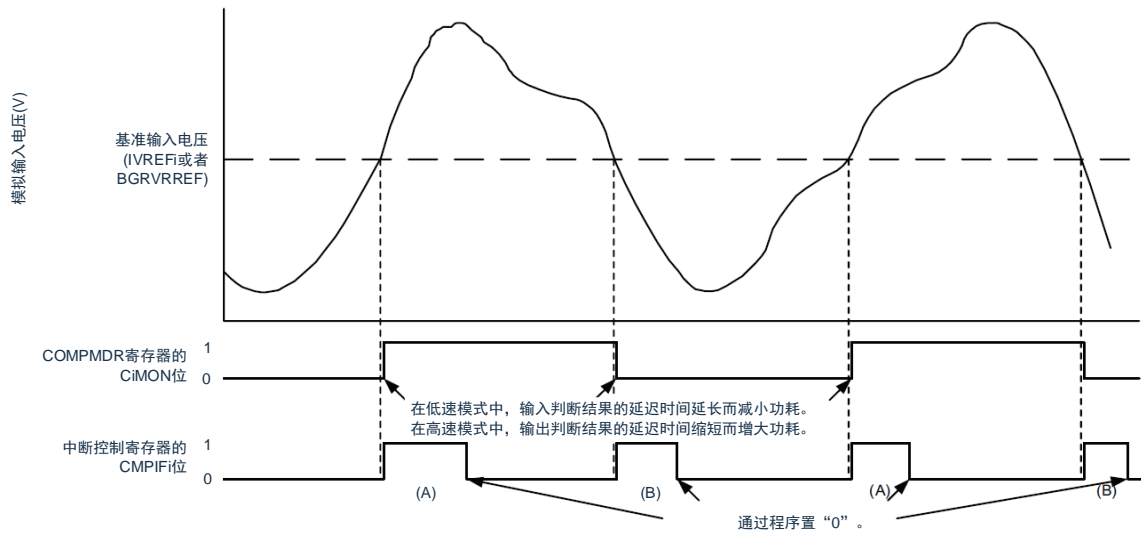
备注: i=0、1, n=0-7, x=0-3

比较器*i*(*i*=0、1)的运行例子如图17-11所示。在基本模式中，当模拟输入电压高于基准输入电压时，COMPMDR寄存器的CiMON位为“1”；当模拟输入电压低于基准输入电压时，CiMON位为“0”。

要使用比较器*i*中断时，必须将COMPOCR寄存器的CiIE位置“1”(允许中断请求)。此时，如果比较结果发生变化，就产生比较器*i*的中断请求。有关中断请求的详细内容，请参照“17.4.2 比较器*i*中断(*i*=0、1)”。

图17-11：比较器*i*(*i*=0、1)的运行例子(基本模式)

• 基本模式的运行例子



注意：上图是COMPFIIR寄存器的CiFCK1~CiFCK0位为“00B”(无滤波器)并且CiEDG位为“1”(双边沿)的情况(CiEDG位为“0”并且CiEPO位为“0”(上升沿)时的CMPIFi只限于(A)的变化，CiEDG位为“0”并且CiEPO位为“1”(下降沿)时的CMPIFi只限于(B)的变化)。

17.4.1 比较器i的数字滤波器(i=0、1)

比较器i内置数字滤波器，能通过COMPFIR寄存器的CiFCK1~CiFCK0位选择采样时钟。按各采样时钟对比较器i的输出信号进行采样，在电平3次都相同后的下一个采样时钟，数字滤波器输出此采样值。

图17-12是比较器i的数字滤波器的结果，图17-13是比较器i的数字比较器(i=0、1)的数字滤波器和中断运行例子。

图17-12: 比较器i(i=0、1)的数字滤波器和沿检出结构

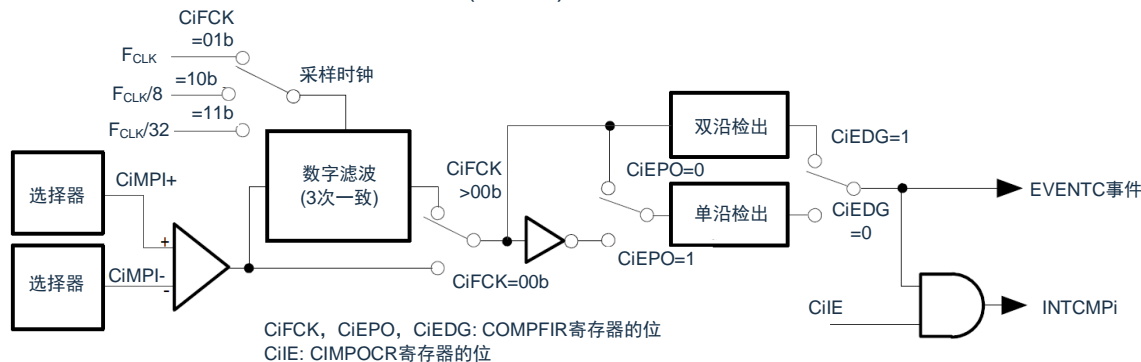
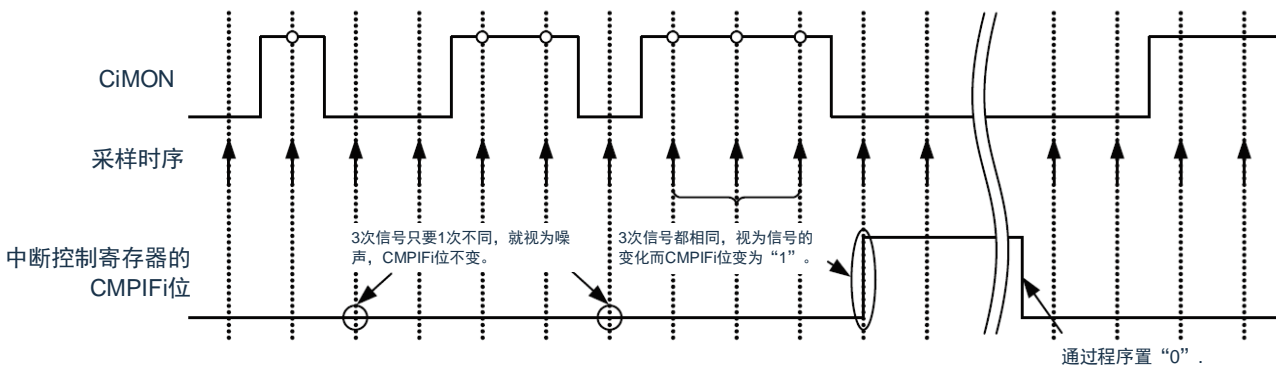


图17-13: 比较器i(i=0、1)的数字滤波器和中断运行例子



注意：上图是COMPFIR寄存器的CiFCK1~CiFCK0位为“01B”、“10B”或者“11B”(有数字滤波器)时的运行例子。

17.4.2 比较器i中断(i=0、1)

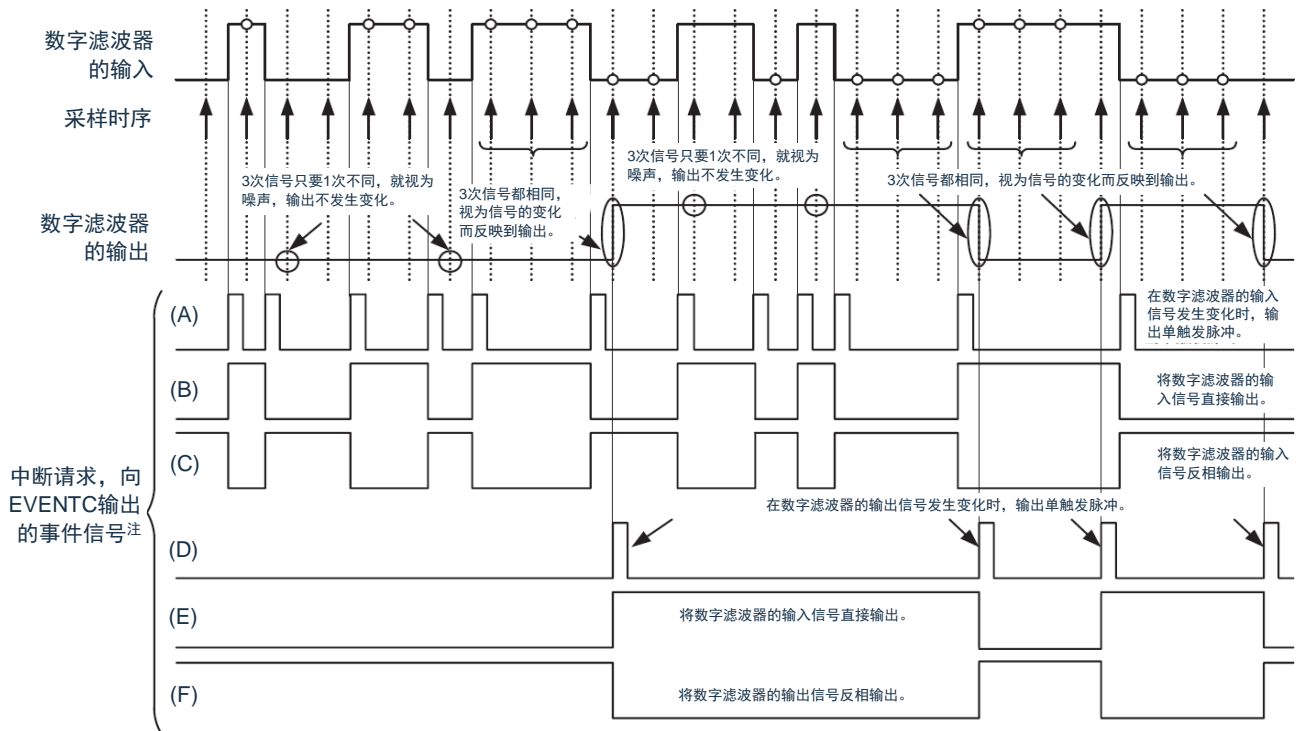
比较器产生比较器0和比较器1共2个中断请求。比较器i中断各有1个优先级指定标志、中断屏蔽标志、中断请求标志和中断向量。

要使用比较器i中断时，必须将COMPOCR寄存器的CiIE位置“1”(允许中断请求的输出)。通过COMPFIR寄存器设置中断请求的产生条件，还能给比较器的输出附加数字滤波器。数字滤波器可选择3种采样时钟。有关寄存器的设置和产生中断请求的对应，请参照“17.3.3 比较器滤波控制寄存器(COMPFIR)”和“17.3.4 比较器输出控制寄存器(COMPOCR)”。

17.4.3 向联动控制器(EVENTC)输出的事件信号

和中断请求的产生条件相同，通过检测COMPFIR寄存器设置的数字滤波器的输出边沿，产生向EVENTC输出的事件信号。但是，和中断请求不同的是与COMPOCR寄存器的CiIE位无关，总是向EVENTC输出事件信号。必须通过EVENTC的ELSELR20寄存器和ELSELR21寄存器设置事件输出目标的选择和事件链接的停止。

图17-14: 数字滤波器、中断请求和向EVENTC输出事件信号的运行



注：当CiIE位(i=0、1)为“1”时，中断请求和向EVENTC输出的事件信号为相同的波形。
当CiIE位(i=0、1)为“0”时，只有中断请求固定为“0”。

(A)、(B)、(C)的波形是COMPFIR寄存器的CiFCK位(i=0、1)为“00B”(无数字滤波器)的情况，(D)、(E)、(F)的波形是COMPFIR寄存器的CiFCK位(i=0、1)为“01B”、“10B”或者“11B”(有数字滤波器)的情况。(A)、(D)是CiEDG位为“1”(双边沿)的情况，(B)、(E)是CiEDG位为“0”并且CiEPO位为“0”(上升沿)的情况，(C)、(F)是CiEDG位为“0”并且CiEPO位为“1”(下降沿)的情况。

17.4.4 比较器i的输出(i=0、1)

能将比较器的比较结果输出到外部引脚，还能通过COMPOCR寄存器的CiOP位和CiOE位设置输出极性，(正相输出或者反相输出)以及是否允许输出。有关寄存器的设置和比较器输出的对应，请参照“17.3.4 比较器输出控制寄存器(COMPOCR)”。

要将比较器的比较结果输出到VCOUTi引脚时，必须按照以下的步骤设置端口(在复位后，端口默认为输入状态)：

- ① 设置比较器的模式(“表17-3：比较器相关寄存器的设置步骤”的步骤2~5)。
- ② 设置比较器的VCOUTi输出(设置COMPOCR寄存器，选择极性并且允许输出)。
- ③ 将VCOUTi的输出引脚对应的端口模式控制寄存器的位置“0”。
- ④ 将VCOUTi的输出引脚对应的端口寄存器的位置“0”。
- ⑤ 将VCOUTi的输出引脚对应的端口方向寄存器设置为输出(从引脚开始输出)。

17.4.5 比较器时钟的停止和提供

在通过设置外围允许寄存器1(PER1)来停止比较器时钟的情况下，必须按照以下步骤进行设置：

- ① 将COMPMDR寄存器的CiENB位置“0”(停止比较器的运行)。
- ② 将中断请求标志寄存器的IF位置“0”(清除比较器停止运行前不需要的中断)。
- ③ 将PER1寄存器的PGACMPEN位置“0”。

如果通过设置PER1寄存器来停止时钟，比较器的内部寄存器就全部被初始化，因此在要再次使用比较器时，必须按照表17-3的步骤设置寄存器。

注意：

1. 如果将比较器模式设置寄存器(COMPMDR)的比较器n基准电压选择位(CnVRF)置“1”(比较器n基准电压为内部基准电压(1.45V))，就不能通过A/D转换器对温度传感器的输出进行A/D转换。
2. 如果在以下某个状态下允许DMA启动，就开始DMA传送，并且在传送结束后产生中断。因此，必须根据需要在确认比较器的监视标志(CnMON)后置为允许DMA启动。
 - 设置为通过比较器的单边沿检测产生中断请求(CnEDG=0)并且通过比较器的上升沿产生中断请求(CnEPO=0)而且VCIN > VREF(或者内部基准电压1.45V)。
 - 设置为通过比较器的单边沿检测产生中断请求(CnEDG=0)并且通过比较器的下降沿产生中断请求(CnEPO=1)而且VCIN < VREF(或者内部基准电压1.45V)。
 (n=0、1)

第18章 可编程增益放大器(PGA)

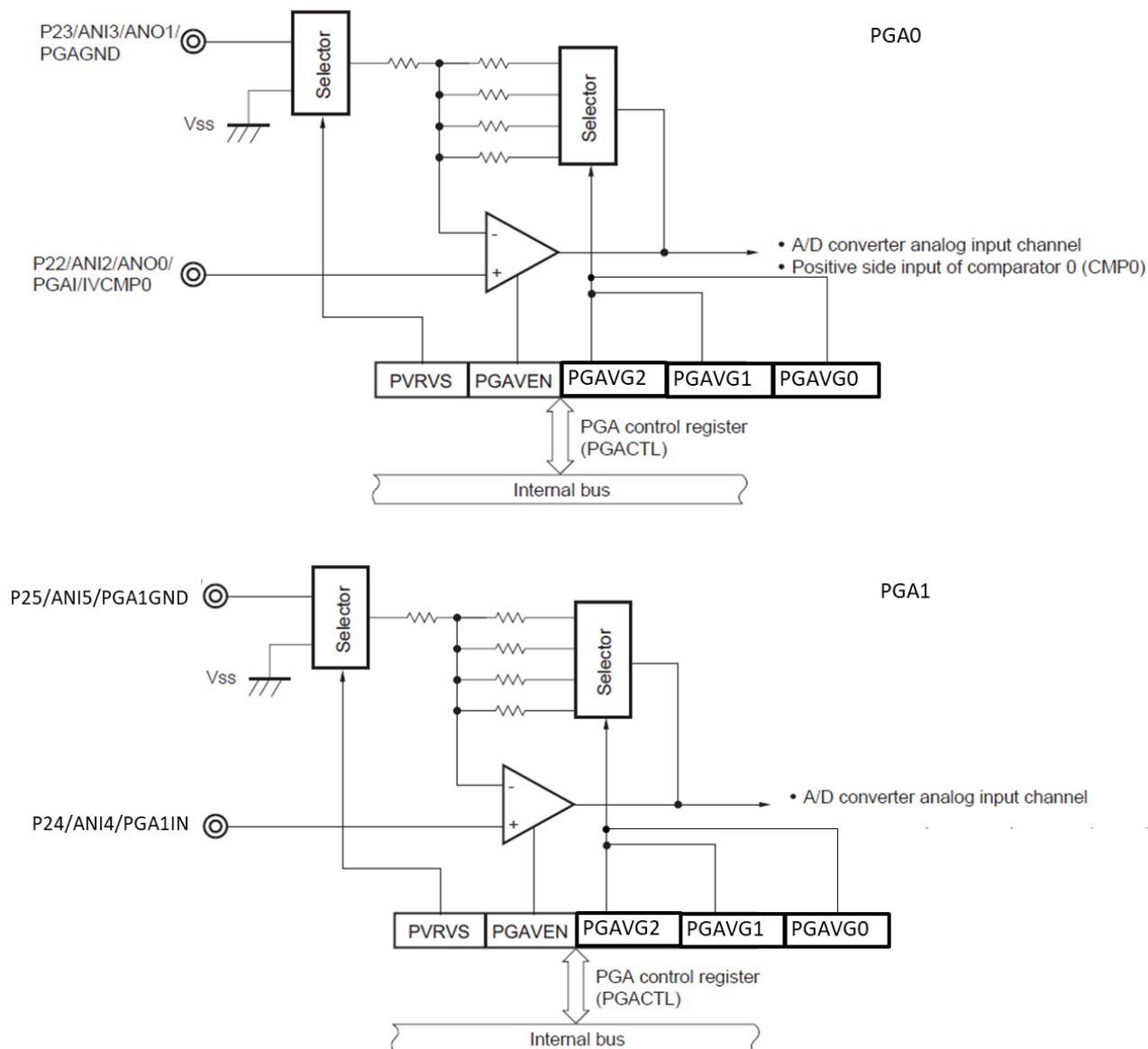
18.1 可编程增益放大器的功能

本产品内置两路可编程增益放大器(PGA0和PGA1), 具有如下功能:

- 每路PGA的放大增益有7种选择: 4倍, 8倍, 10倍, 12倍, 14倍, 16倍, 32倍
- 可选择外部引脚作为PGA负端反馈电阻的地
- PGA0的输出可被选为用于A/D转换器的模拟输入或者比较器0(CMP0)正端的模拟输入
- PGA1的输出可被选为用于A/D转换器的模拟输入

18.2 可编程增益放大器的结构

图18-1: 可编程增益放大器的框图



18.3 可编程增益放大器的寄存器

表18-1: 控制可编程增益放大器的寄存器

外围允许寄存器1	PER1
可编程增益放大器控制寄存器	PGACTL
端口模式控制寄存器2	PMC2
端口模式寄存器2	PM2

18.3.1 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用可编程增益放大器时，必须将此寄存器的bit5 (PGACMPEN)置“1”。

通过1位或者8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图18-2: 外围允许寄存器1(PER1)的格式

复位值: 00H									
		R/W							
符号	7	6	5	4	3	2	1	0	
PER1	DACEN	TIMERBEN	PGACMPEN	TIMERMEN	DTCEN	PWMOPEN	TIMERCEN	TIMERAEN	

PGACMPEN	比较器/可编程增益放大器的输入时钟的控制
0	停止提供输入时钟。 比较器或可编程增益放大器的寄存器不可写 比较器或可编程增益放大器处于复位状态
1	提供输入时钟。 比较器或可编程增益放大器的寄存器可读写

注意：在配置比较器或可编程增益放大器的寄存器之前，确认PGACMPEN的bit位先要置1。

如果PGACMPEN=0，对比较器或可编程增益放大器的控制寄存器进行写操作是无效的,并且所有的读出值为默认值。(端口模式寄存器2(PM2)和端口寄存器P2除外)

18.3.2 可编程增益放大器控制寄存器(PGAnCTL)

PGA0CTL和PGA1CTL寄存器用于控制可编程增益放大器开始工作，停止工作和放大倍数。

可通过1位或者8位存储器操作指令设置PGA0CTL和PGA1CTL寄存器。在产生复位信号后，此寄存器复位值为00H。

图18-3: PGA控制寄存器(PGAnCTL)的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
PGAnCTL	PGAnEN	-	-	-	PVRVS ^注	PGAnVG2	PGAnVG1	PGAnVG0

n=0,1

PGAnEN	可编程增益放大器运行控制
0	放大器停止工作
1	允许放大器工作

PVRVS ^注	反馈电阻地的选择
0	选择 V _{SS}
1	选择 PGAnGND 引脚

PGAnVG2	PGAnVG1	PGAnVG0	PGAn 增益
0	0	0	4 倍
0	0	1	8 倍
0	1	0	10 倍
0	1	1	12 倍
1	0	0	14 倍
1	0	1	16 倍
1	1	0	32 倍
其他			禁止设置

注: 24pin制品时, 设置此位为0。

注意: PGAnEN设置为1后, 可编程增益放大器运行需要10us的稳定时间。

18.3.3 控制模拟输入引脚端口功能的寄存器

在将PGA0IN引脚、PGA1IN引脚、PGA0GND引脚和PGA1GND引脚用作可编程增益放大器的模拟输入时, 必须将各端口对应的端口模式寄存器(PMxx)的位和端口数字/模拟控制寄存器(PMCxx)的位置“1”。

18.4 可编程增益放大器的运行

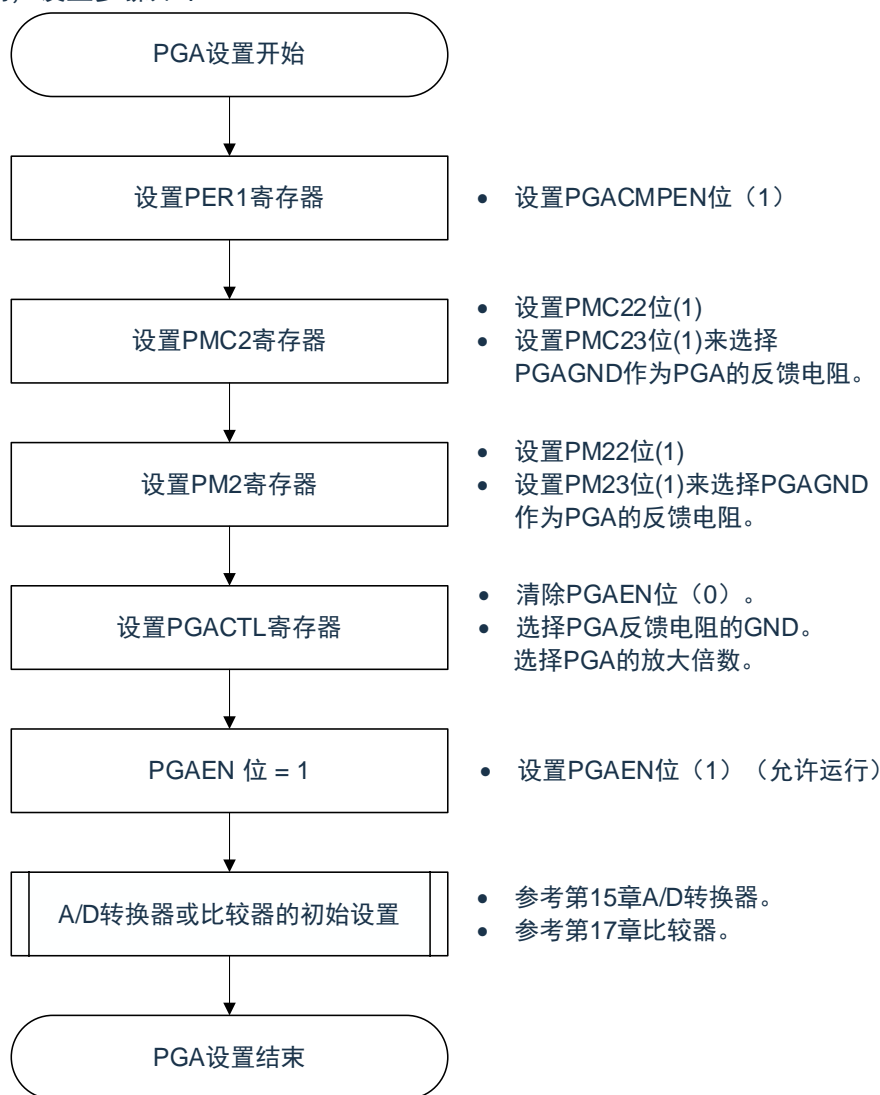
对PGA_{IN}引脚输入的模拟电压进行放大，放大增益有7种选择：4倍，8倍，10倍，12倍，14倍，16倍，32倍。

放大之后的电压可以用于A/D转换器的模拟输入，和比较器0(CMP0)的正端输入信号。

可编程增益放大器的开始运行和停止运行的步骤如下。

18.4.1 可编程增益放大器的开始运行步骤

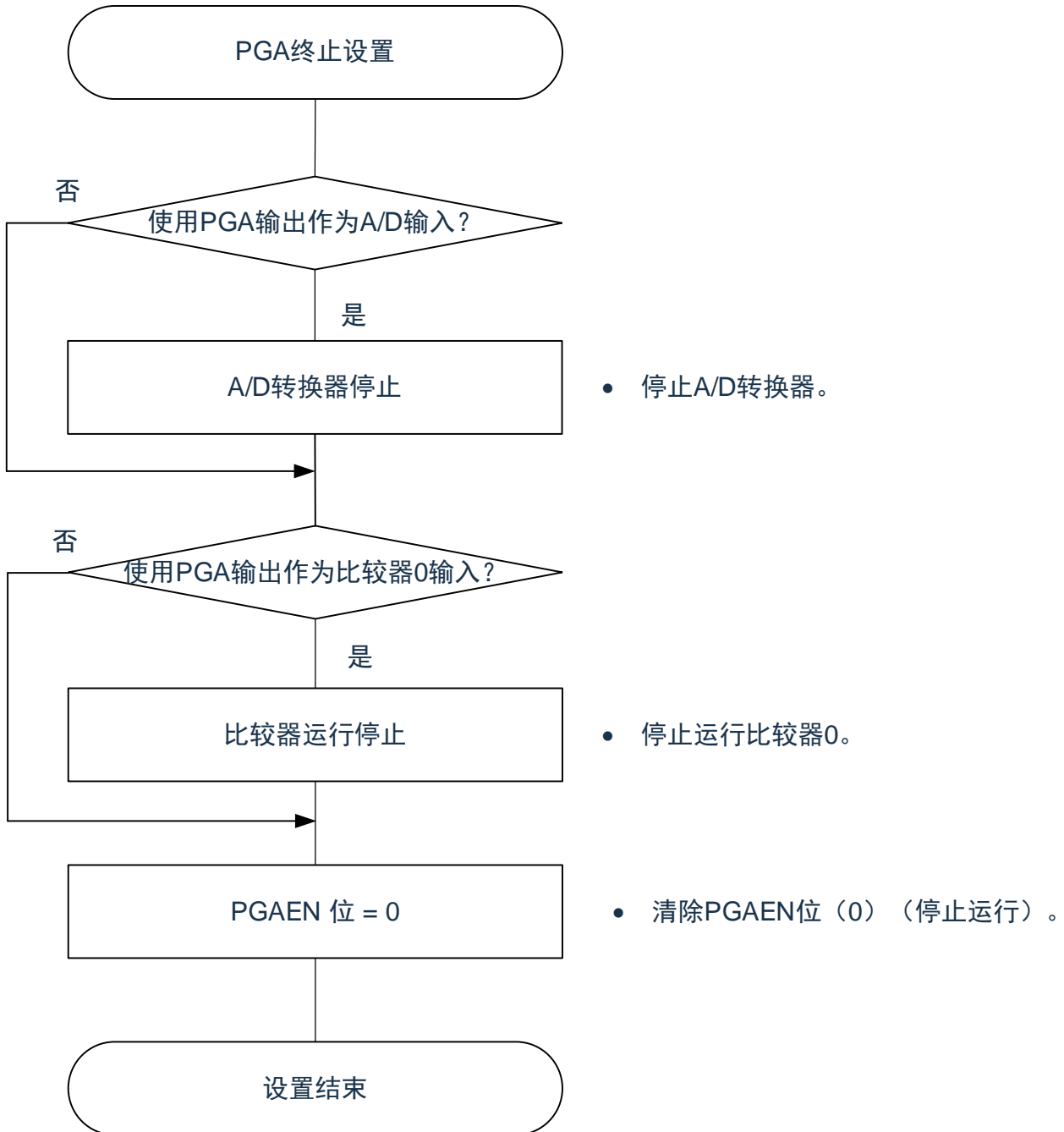
以PGA0为例，设置步骤如下：



注意：在设置PGAEN位为1后需要10us的PGA稳定时间。之后再启动A/D转换。

18.4.2 可编程增益放大器的停止运行步骤

以PGA0为例，设置步骤如下：



注意：

1. 当重新开始PGA和A/D转换或者放大器时，在设置PGAEN位为1后需要10us的PGA稳定时间。
2. 即使停止PGA操作，也可以使用直通的引脚进行A/D转换和比较器动作。

第19章 通用串行通信单元

通用串行通信单元的单元0有4个串行通道，单元1有2个串行通道，各通道能实现3线串行(SSPI)、UART和简易I²C的通信功能。

BAT32A237支持的各通道的功能分配如下：

○32引脚的产品

单元	通道	用作SSPI	用作UART	用作简易I ² C
0	0	SSPI00 (支持从属选择输入功能)	UART0(支持LIN-bus)	IIC00
	1	—		—
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	—		—

○40引脚的产品

单元	通道	用作SSPI	用作UART	用作简易I ² C
0	0	SSPI00 (支持从属选择输入功能)	UART0(支持LIN-bus)	IIC00
	1	SSPI01		IIC01
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	—		—

○48引脚的产品

单元	通道	用作SSPI	用作UART	用作简易I ² C
0	0	SSPI00 (支持从属选择输入功能)	UART0(支持LIN-bus)	IIC00
	1	SSPI01		IIC01
	2	—	UART1	—
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21

○64引脚产品

单元	通道	用作SSPI	用作UART	用作简易I ² C
0	0	SSPI00 (支持从属选择输入功能)	UART0(支持LIN-bus)	IIC00
	1	SSPI01		IIC01
	2	SSPI10	UART1	IIC10
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21

在单元0的通道0和通道1使用UART0时，不能使用SSPI00和SSPI01，但是能使用通道2和通道3的SSPI10、UART1和IIC10。

注意：本章的下述内容主要针64引脚产品的单元和通道结构进行说明。

19.1 通用串行通信单元的功能

BAT32A237支持的各串行接口的特征如下所示。

19.1.1 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

与主控设备输出的串行时钟(SCLK)同步进行数据的发送和接收。

这是使用1条串行时钟(SCLK)、1条发送串行数据(SDO)和1条接收串行数据(SDI)共3条通信线进行通信的时钟同步通信功能。

有关具体的设置例子，请参照“19.5 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信的运行”。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。

- 最大传送速率^注

主控通信：最大值 $F_{CLK}/2$

从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注：必须在满足SCLK周期时间(T_{KCY})特性的范围内使用。详细内容请参照数据手册。

19.1.2 UART(UART0~UART2)

这是通过串行数据发送(TxD)和串行数据接收(RxD)共2条线进行异步通信的功能。使用这2条通信线，按数据帧(由起始位、数据、奇偶校验位和停止位构成)与其他通信方进行异步(使用内部波特率)的数据发送和接收。能通过使用发送专用(偶数通道)和接收专用(奇数通道)共2个通道来实现全双工UART通信，而且还能通过组合通用定时器单元和外部中断(INTP0)来支持LIN-bus。

有关具体的设置例子，请参照“19.7 UART(UART0~UART2)通信的运行”。

[数据的发送和接收]

- 7位、8位或者9位的数据长度
- MSB/LSB优先的选择
- 发送和接收数据的电平设置、反相的选择
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误或者溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

UART0(单元0的通道0和通道1)支持LIN-bus。 [LIN-bus功能]

- | | | |
|--|---|------------------------|
| <ul style="list-style-type: none"> • 唤醒信号的检测 • 间隔段(BF)的检测 • 同步段的测量、波特率的计算 | } | 使用外部中断(INTP0)和通用定时器单元。 |
|--|---|------------------------|

19.1.3 简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)

这是通过串行时钟(SCL)和串行数据(SDA)共2条线与多个设备进行时钟同步通信的功能。因为此简易I²C是为了与EEPROM、闪存、A/D转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守AC规格，在操作控制寄存器的同时通过软件进行处理。有关具体的设置例子，请参照“19.9 简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收(只限于单主控的主控功能)
- ACK输出功能^注、ACK检测功能
- 8位数据长度(在发送地址时，用高7位指定地址，用最低位进行R/W控制)
- 开始条件和停止条件的手动产生

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK错误、溢出错误

※[简易I²C不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注：在接收最后的数据时，如果给SOEmn位(串行输出允许寄存器m(SOEm))写“0”来停止串行通信数据的输出，就不输出ACK。详细内容请参照“19.9.3(2) 处理流程”。

备注：在使用全功能的I²C总线时，请参照“第20章 串行接口IICA”。

19.2 通用串行通信单元的结构

通用串行通信单元由以下硬件构成。

表19-1: 通用串行通信单元的结构

项目	结构
移位寄存器	8位或者9位 ^{注1}
缓冲寄存器	串行数据寄存器mn(SDRmn)的低8位或者9位 ^{注1,2}
串行时钟输入/输出	SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚(用于3线串行I/O)、SCL00、SCL01、SCL10、SCL11、SCL20、SCL21引脚(用于简易I ² C)
串行数据输入	SDI00、SDI01、SDI10、SDI11、SDI20、SDI21引脚(用于3线串行I/O)、RxD0(用于支持LIN-bus的UART)、RxD1、RxD2引脚(用于UART)
串行数据输出	SDO00、SDO01、SDO10、SDO11、SDO20、SDO21引脚(用于3线串行I/O)、TxD0(用于支持LIN-bus的UART)、TxD1、TxD2引脚(用于UART)
串行数据输入/输出	SDA00、SDA01、SDA10、SDA11、SDA20、SDA21引脚(用于简易I ² C)
从属选择输入	SS00引脚(用于从属选择输入功能)
控制寄存器	<单元设置部的寄存器> <ul style="list-style-type: none"> • 外围允许寄存器0(PER0) • 串行时钟选择寄存器m(SPSm) • 串行通道允许状态寄存m(SEm) • 串行通道开始寄存器m(SSm) • 串行通道停止寄存器m(STm) • 串行输出允许寄存器m(SOEm) • 串行输出寄存器m(SOm) • 串行输出电平寄存器m(SOLm) • 串行待机控制寄存器m(SSCm) • 输入切换控制寄存器(ISC) • 噪声滤波器允许寄存器0(NFEN0)
	<各通道部的寄存器> <ul style="list-style-type: none"> • 串行数据寄存器mn(SDRmn) • 串行模式寄存器mn(SMRmn) • 串行通信运行设置寄存器mn(SCRmn) • 串行状态寄存器mn(SSRmn) • 串行标志清除触发寄存器mn(SIRmn)
	<ul style="list-style-type: none"> • 端口输入模式寄存器(PIMxx) • 端口输出模式寄存器(POMxx) • 端口模式寄存器(PMxx) • 端口寄存器(Pxx)

注1: 用作移位寄存器和缓冲寄存器的位数因单元和通道而不同。

- 24~64引脚产品并且mn=00、01: 低9位
- 上述以外: 低8位

注2: 根据通信模式, 能用以下SFR名称读写串行数据寄存器mn(SDRmn)的低8位。

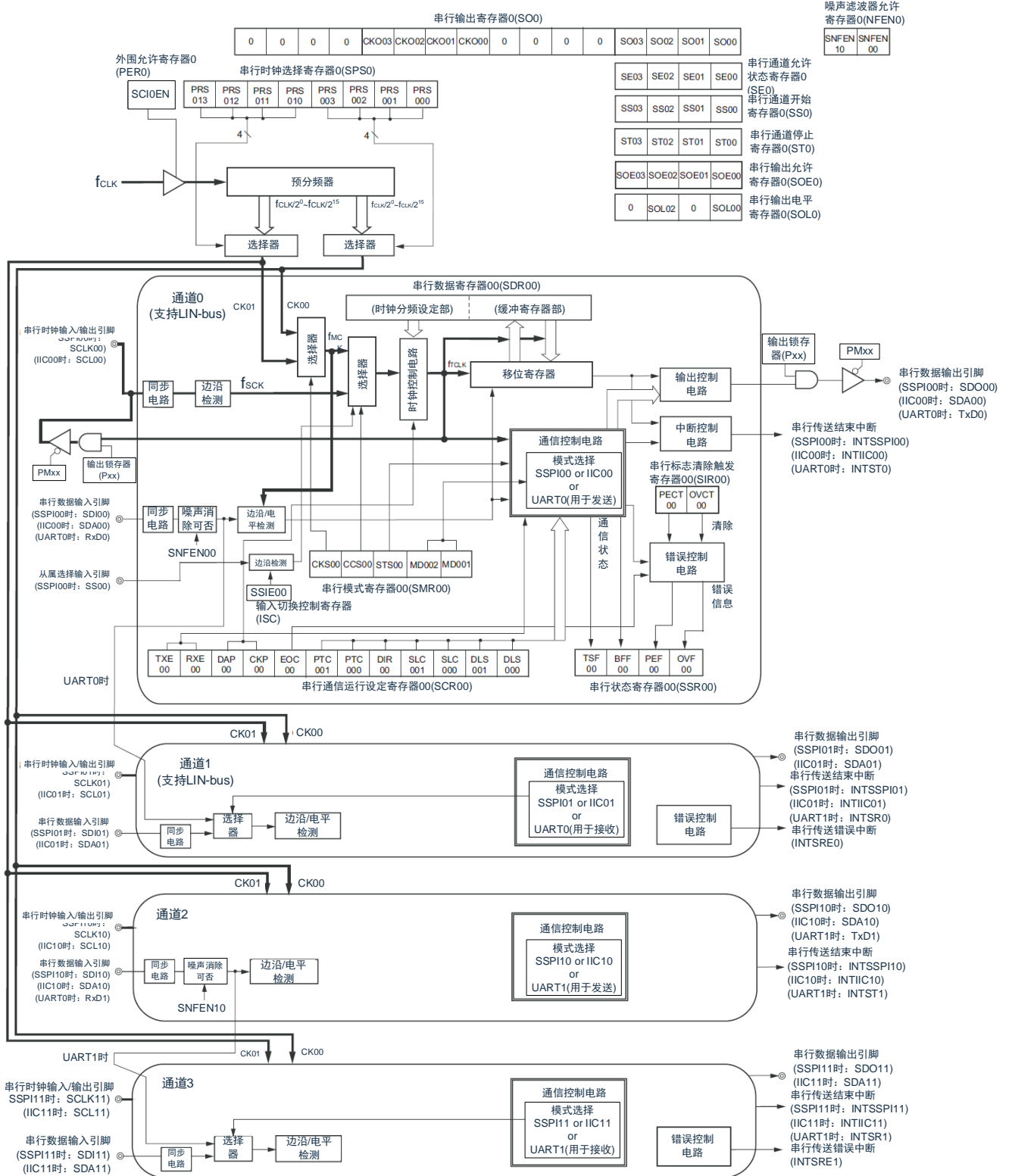
- SSPIp通信.....SIOp(SSPIp数据寄存器)
- UARTq接收.....RXDq(UARTq接收数据寄存器)
- UARTq发送.....TXDq(UARTq发送数据寄存器)
- IICr通信.....SIOr(IICr数据寄存器)

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)

q: UART号(q=0~2)r: IIC号(r=00、01、10、11、20、21)

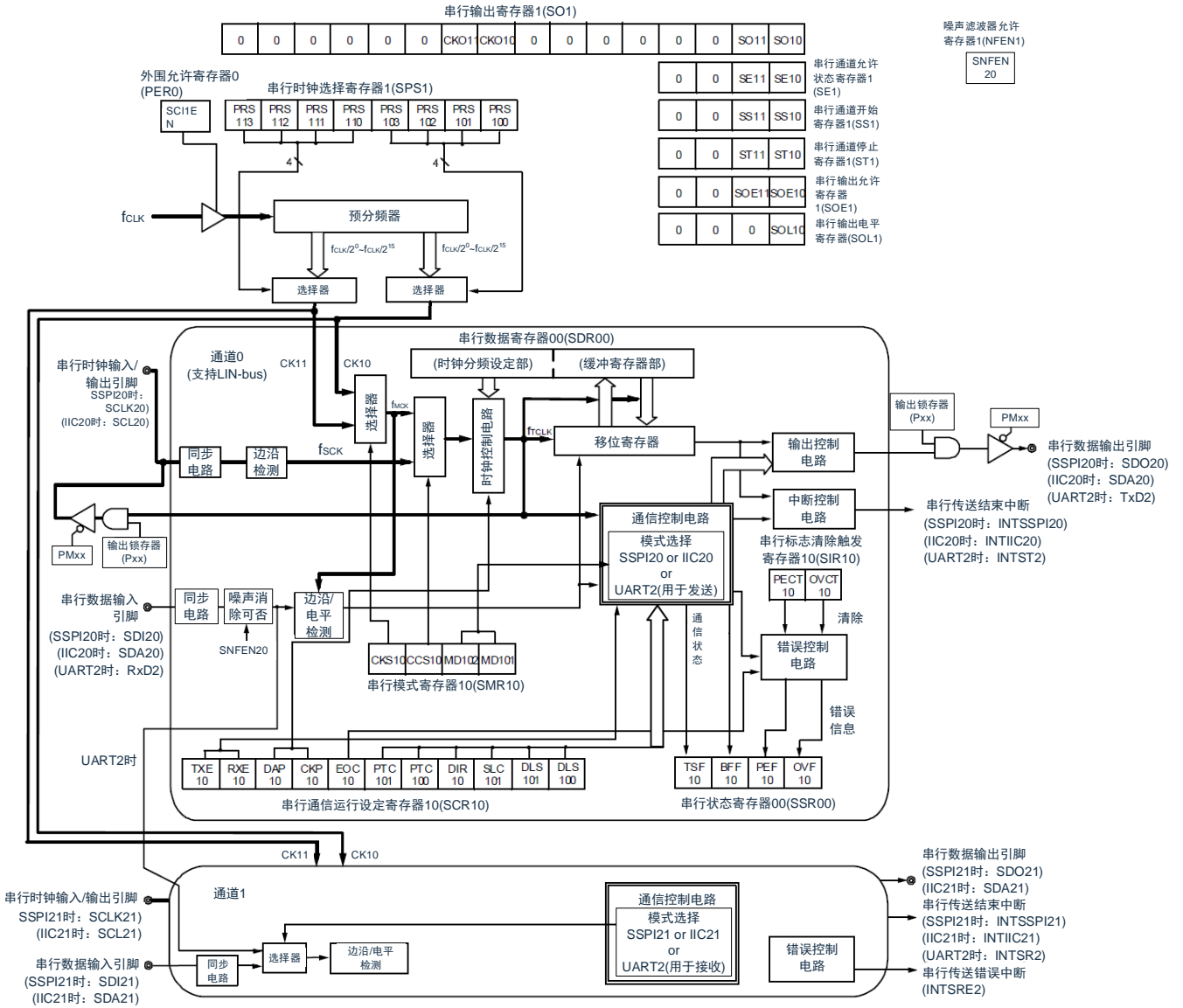
通用串行通信单元0的框图如图19-1所示。

图19-1：通用串行通信单元0的框图



通用串行通信单元1的框图如图19-2所示。

图19-2: 通用串行通信单元1的框图



19.2.1 移位寄存器

这是进行并行和串行相互转换的9位寄存器。

在以9位数据长度进行UART通信时，使用9位(bit0~8)^{注1}。在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出^{注1}。不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，使用串行数据寄存器mn(SDRmn)的低8位或者低9位。



19.2.2 串行数据寄存器mn(SDRmn)的低8位或者低9位

SDRmn寄存器是通道n的发送和接收数据寄存器(16位)。

bit8~0(低9位)^注或者bit7~0(低8位)用作发送和接收缓冲寄存器，bit15~9用作运行时钟(F_{MCK})的分频设置寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到低8位或者低9位；在发送数据时，将被传送到移位寄存器的发送数据设置到低8位或者低9位。

与数据的输出顺序无关，根据串行通信运行设置寄存器mn(SCRmn)的bit0和bit1(DLSmn0、DLSmn1)的设置，保存到低8位或者低9位的数据如下所示：

- 7位数据长度(保存在SDRmn寄存器的bit0~6)
- 8位数据长度(保存在SDRmn寄存器的bit0~7)
- 9位数据长度(保存在SDRmn寄存器的bit0~8)^{注1}

能以16位为单位读写SDRmn寄存器。

根据通信模式，能用以下SFR名称，以8位为单位读写SDRmn寄存器的低8位或者低9位^{注2}。

- SSPIp通信.....SDIOp(SSPIp数据寄存器)
- UARTq接收.....RXDq(UARTq接收数据寄存器)
- UARTq发送.....TXDq(UARTq发送数据寄存器)
- IICr通信.....SDIOr(IICr数据寄存器)

在产生复位信号后，SDRmn寄存器的值变为“0000H”。

注1：只有UART0支持9位数据长度。

注2：在运行停止(SEmn=0)时，禁止通过8位存储器操作指令改写SDRmn[7:0](否则，SDRmn[15:9]全部被清“0”)。

备注：

1. 在接收结束后，bit0~8中超过数据长度的部分的位为“0”。
2. m：单元号(m=0、1)n：通道号(n=0~3)p：SSPI号(p=00、01、10、11、20、21)
q：UART号(q=0~2)r：IIC号(r=00、01、10、11、20、21)

19.3 控制通用串行通信单元的寄存器

控制通用串行通信单元的寄存器如下所示：

- 外围允许寄存器0(PER0)
- 串行时钟选择寄存器m(SPSm)
- 串行模式寄存器mn(SMRmn)
- 串行通信运行设置寄存器mn(SCRmn)
- 串行数据寄存器mn(SDRmn)
- 串行标志清除触发寄存器mn(SDIRmn)
- 串行状态寄存器mn(SSRmn)
- 串行通道开始寄存器m(SSm)
- 串行通道停止寄存器m(STm)
- 串行通道允许状态寄存器m(SEm)
- 串行输出允许寄存器m(SOEm)
- 串行输出电平寄存器m(SOLm)
- 串行输出寄存器m(SOm)
- 输入切换控制寄存器(ISC)
- 噪声滤波器允许寄存器0(NFEN0)
- 端口输入模式寄存器(PIMx)
- 端口输出模式寄存器(POMx)
- 端口模式寄存器(PMx)
- 端口寄存器(Px)

备注：m：单元号(m=0、1)n：通道号(n=0~3)

19.3.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用通用串行通信单元0时，必须将bit2(SCI0EN)置“1”。

要使用通用串行通信单元1时，必须将bit3(SCI1EN)置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，PER0寄存器的值变为“00H”。

图19-5：外围允许寄存器0(PER0)的格式

地址：0x40020420 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	CAN0EN	TM40EN

SCI _m EN	提供通用串行通信单元m的输入时钟的控制
0	停止提供输入时钟。 •不能写通用串行通信单元m使用的SFR。 •通用串行通信单元m处于复位状态。
1	允许提供输入时钟。 •能读写通用串行通信单元m使用的SFR。

注意：要设置通用串行通信单元m时，必须先在此SCI_mEN位为“1”的状态下设置以下的寄存器。当SCI_mEN位为“0”时，忽视通用串行通信单元m的控制寄存器的写操作，而且读取值都为初始值(输入切换控制寄存器(ISC)、噪声滤波器允许寄存器0(NFEN0)、端口输入模式寄存器(PIMx)、端口输出模式寄存器(POMx)、端口模式寄存器(PMx)、端口模式控制寄存器(PMCx)以及端口寄存器(Px)除外)。

- 串行时钟选择寄存器m(SPSm)
- 串行模式寄存器mn(SMRmn)
- 串行通信运行设置寄存器mn(SCRmn)
- 串行数据寄存器mn(SDRmn)
- 串行标志清除触发寄存器mn(SIRmn)
- 串行状态寄存器mn(SSRmn)
- 串行通道开始寄存器m(SSm)
- 串行通道停止寄存器m(STm)
- 串行通道允许状态寄存器m(SEm)
- 串行输出允许寄存器m(SOEm)
- 串行输出电平寄存器m(SOLm)
- 串行输出寄存器m(SOm)
- 串行待机控制寄存器m(SSCm)

19.3.2 串行时钟选择寄存器m(PSm)

SPSm寄存器是16位寄存器，选择提供给各通道的2种公共运行时钟(CKm0、CKm1)。通过SPSm寄存器的bit7~4选择CKm1，通过bit3~0选择CKm0。

禁止在运行过程中(SEmn=1)改写SPSm寄存器。

通过16位存储器操作指令设置SPSm寄存器。

能用SPSmL并且通过8位存储器操作指令设置SPSm寄存器的低8位。

在产生复位信号后，SPSm寄存器的值变为“0000H”。

图19-6：串行时钟选择寄存器m(PSm)的格式

地址：40041126H(PS0)、40041566H(PS1)								复位后：0000H								R/W
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	运行时钟(CKmk)的选择 [‡]					
				F _{CLK}	F _{CLK} =2MHz	F _{CLK} =5MHz	F _{CLK} =10MHz	F _{CLK} =20MHz	F _{CLK} =48MHz
0	0	0	0	F _{CLK}	2MHz	5MHz	10MHz	20MHz	48MHz
0	0	0	1	F _{CLK} /2	1MHz	2.5MHz	5MHz	10MHz	24MHz
0	0	1	0	F _{CLK} /2 ²	500KHz	1.25MHz	2.5MHz	5MHz	12MHz
0	0	1	1	F _{CLK} /2 ³	250KHz	625KHz	1.25MHz	2.5MHz	6MHz
0	1	0	0	F _{CLK} /2 ⁴	125KHz	313KHz	625KHz	1.25MHz	3MHz
0	1	0	1	F _{CLK} /2 ⁵	62.5KHz	156KHz	313KHz	625KHz	1.5MHz
0	1	1	0	F _{CLK} /2 ⁶	31.3KHz	78.1KHz	156KHz	313KHz	750KHz
0	1	1	1	F _{CLK} /2 ⁷	15.6KHz	39.1KHz	78.1KHz	156KHz	375KHz
1	0	0	0	F _{CLK} /2 ⁸	7.81KHz	19.5KHz	39.1KHz	78.1KHz	187.5KHz
1	0	0	1	F _{CLK} /2 ⁹	3.91KHz	9.77KHz	19.5KHz	39.1KHz	93.75KHz
1	0	1	0	F _{CLK} /2 ¹⁰	1.95KHz	4.88KHz	9.77KHz	19.5KHz	46.9KHz
1	0	1	1	F _{CLK} /2 ¹¹	977Hz	2.44KHz	4.88KHz	9.77KHz	23.4KHz
1	1	0	0	F _{CLK} /2 ¹²	488Hz	1.22KHz	2.44KHz	4.88KHz	11.7KHz
1	1	0	1	F _{CLK} /2 ¹³	244Hz	610Hz	1.22KHz	2.44KHz	5.85KHz
1	1	1	0	F _{CLK} /2 ¹⁴	122Hz	305Hz	610Hz	1.22KHz	2.92KHz
1	1	1	1	F _{CLK} /2 ¹⁵	61Hz	153Hz	305Hz	610Hz	1.46KHz

注：要在通用串行通信单元(SCI)运行过程中更改被选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)时，必须在停止SCI的运行(串行通道停止寄存器m(STm)=000FH)后进行更改。

注意：必须将bit15~8置“0”。

备注：

1. F_{CLK}：CPU/外围硬件的时钟频率
2. m：单元号(m=0、1)
3. k=0、1

19.3.3 串行模式寄存器mn(SMRmn)

SMRmn寄存器是设置通道n运行模式的寄存器，进行运行时钟(F_{MCK})的选择、能否使用串行时钟(F_{SCLK})输入的指定、开始触发的设置、运行模式(SSPI、UART、简易I²C)的设置以及中断源的选择。另外，只在UART模式中设置接收数据的反相电平。

禁止在运行过程中(SEmn=1)改写SMRmn寄存器，但是能在运行过程中改写MDmn0位。

通过16位存储器操作指令设置SMRmn寄存器。

在产生复位信号后，SMRmn寄存器的值变为“0020H”。

图19-7：串行模式寄存器mn(SMRmn)的格式(1/2)

地址：40041110H(SMR00)~40041116H(SMR03) 复位后：0020H

R/W

40041550H(SMR10)~40041552H(SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn ^{注1}	0	SISmn ^{注1}	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	通道n运行时钟(F_{MCK})的选择
0	SPSm寄存器设置的运行时钟CKm0
1	SPSm寄存器设置的运行时钟CKm1
运行时钟(F_{MCK})用于边沿检测电路。通过设置CCSmn位和SDRmn寄存器的高7位，生成传送时钟(F_{TCLK})。	

CCSmn	通道n传送时钟(F_{TCLK})的选择
0	CKSmn位指定的运行时钟 F_{MCK} 的分频时钟
1	来自SCLKp引脚的输入时钟 F_{SCLK} (SSPI模式的从属传送)
传送时钟 F_{TCLK} 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当CCSmn位为“0”时，通过SDRmn寄存器的高7位进行运行时钟(F_{MCK})的分频设置。	

STSmn ^{注1}	开始触发源的选择
0	只有软件触发有效(在SSPI、UART发送、简易I ² C时选择)。
1	RxDq引脚的有效边沿(在UART接收时选择)
在将SSm寄存器置“1”后满足上述条件时，开始传送。	

注1：只限于SMR01、SMR03、SMR11寄存器。

注意：必须将bit13~9、7、4、3(SMR00、SMR02、SMR10寄存器为bit13~6、4、3)置“0”，并且将bit5置“1”。

备注：m：单元号(m=0、1)n：通道号(n=0~3)p：SSPI号(p=00、01、10、11、20、21)

q：UART号(q=0~2)r：IIC号(r=00、01、10、11、20、21)

图19-7：串行模式寄存器mn(SMRmn)的格式(2/2)

地址：40041110H(SMR00)~40041116H(SMR03) 复位后：0020H

R/W

40041550H(SMR10)~40041552H(SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSm _n ^{注1}	0	SISmn ₀ ^{注1}	1	0	0	MDmn2	MDmn1	MDmn0

SISmn ₀ ^{注1}	UART模式中的通道n接收数据的电平反相控制
0	将下降沿检测为起始位。 不将输入的通信数据进行反相。
1	将上升沿检测为起始位。 将输入的通信数据进行反相。

MDmn2	MDmn1	通道n运行模式的设置
0	0	SSPI模式
0	1	UART模式
1	0	简易I ² C模式
1	1	禁止设置。

MDmn0	通道n中断源的选择
0	传送结束中断
1	缓冲器空中断 (在数据从SDRmn寄存器传送到移位寄存器时发生)

在连续发送时，如果MDmn0位为“1”并且SDRmn的数据为空，就写下一个发送数据。

注1：只限于SMR01、SMR03、SMR11、寄存器。

注意：必须将bit13~9、7、4、3(SMR00、SMR02、SMR10寄存器为bit13~6、4、3)置“0”，并且将bit5置“1”。

备注：m：单元号(m=0、1)n：通道号(n=0~3)p：SSPI号(p=00、01、10、11、20、21)

q：UART号(q=0~2)r：IIC号(r=00、01、10、11、20、21)

19.3.4 串行通信运行设置寄存器mn(SCRmn)

SCRmn寄存器是通道n的通信运行设置寄存器，设置数据发送和接收模式、数据和时钟相位、是否屏蔽错误信号、奇偶检验位、起始位、停止位和数据长度等。

禁止在运行过程中(SEmn=1)改写SCRmn寄存器。

通过16位存储器操作指令设置SCRmn寄存器。

在产生复位信号后，SCRmn寄存器的值变为“0087H”。

图19-8: 串行通信运行设置寄存器mn(SCRmn)的格式(1/2)

地址: 40041118H(SCR00)~4004111EH(SCR03) 复位后: 0087H R/W
 40041558H(SCR10)~4004155AH(SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 ^{注1}	SLCmn0	0	1	DLSmn1 ^{注2}	DLSmn0

TXEmn	RXEmn	通道n运行模式的设置
0	0	禁止通信。
0	1	使能接收。
1	0	使能发送。
1	1	使能发送和接收。

DAPmn	CKPmn	SSPI 模式中的数据和时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4

在 UART 模式和简易 I²C 模式中，必须将 DAPmn 位和 CKPmn 位都置“0”。

EOCmn	错误中断信号(INTSREx(x=0~3))的屏蔽控制
0	禁止产生错误中断INTSREx(产生INTSRx)。
1	允许产生错误中断INTSREx(在发生错误时不产生INTSRx)。

在SSPI模式和简易I²C模式中或者在UART发送时，必须将EOCmn位置“0”^{注3}。

注1: 只限于SCR00、SCR02、SCR10寄存器。

注2: 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

注3: 在EOCmn位为“0”并且不使用SSPImn时，有可能产生错误中断INTSREn。

注意: 必须将bit3、6、11置“0”(也必须将SCR01、SCR03、SCR11寄存器的bit5置“0”), 并且将bit2置“1”。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)

图19-8: 串行通信运行设置寄存器mn(SCRmn)的格式(2/2)

 地址: 40041118H(SCR00)~4004111EH(SCR03) 复位后: 0087H R/W
 40041558H(SCR10)~4004155AH(SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 ^{注1}	SLC mn0	0	1	DLSm n1 ^{注2}	DLS mn0

PTCmn1	PTCmn0	UART模式中的奇偶校验位的设置	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出奇偶校验 ^{注3} 。	不判断奇偶校验。
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。

在SSPI模式和简易I²C模式中, 必须将PTCmn1位和PTCmn0位都置“0”。

DIRmn	SSPI和UART模式中的数据传送顺序的选择
0	进行MSB优先的输入/输出。
1	进行LSB优先的输入/输出。

在简易I²C模式中, 必须将DIRmn位置“0”。

SLCmn1 ^{注1}	SLCmn0	UART模式中的停止位的设置
0	0	无停止位
0	1	停止位长度=1位
1	0	停止位长度=2位(只限于mn=00、02、10)
1	1	禁止设置。

如果选择了传送结束中断, 就在传送完所有停止位后产生中断。
 在UART接收时或者在简易I²C模式中, 必须设置为1个停止位(SLCmn1、SLCmn0=0、1)。
 在SSPI模式中, 必须设置为无停止位(SLCmn1、SLCmn0=0、0)。
 在UART发送时, 必须设置为1位(SLCmn1、SLCmn0=0、1)或者2位(SLCmn1、SLCmn0=1、0)。

DLSmn1 ^{注2}	DLSmn0	SSPI和UART模式中的数据长度的设置
0	1	9位数据长度(保存在SDRmn寄存器的bit0~8)(只在UART模式中可选择)
1	0	7位数据长度(保存在SDRmn寄存器的bit0~6)
1	1	8位数据长度(保存在SDRmn寄存器的bit0~7)
其他		禁止设置。

在简易I²C模式中, 必须将DLSmn1位和DLSmn0位都置“1”。

注1: 只限于SCR00、SCR02、SCR10寄存器。

注2: 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

注3: 与数据的内容无关, 总是附加“0”。

注意: 必须将bit3、6、11置“0”(也必须将SCR01、SCR03、SCR11寄存器的bit5置“0”), 并且将bit2置“1”。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)

19.3.5 串行数据寄存器mn(SDRmn)

SDRmn寄存器是通道n发送和接收的数据寄存器(16位)。

SDR00、SDR01的bit8~0(低9位)或者SDR02、SDR03、SDR10、SDR11的bit7~0(低8位)用作发送和接收缓冲寄存器，bit15~9(高7位)用作运行时钟(F_{MCK})的分频设置寄存器。

如果将串行模式寄存器mn(SMRmn)的CCSmn位置“0”，由SDRmn寄存器的bit15~9(高7位)设置的运行时钟的分频时钟就用作传送时钟。

如果将CCSmn位置“1”，就必须将SDRmn的bit15~9(高7位)置“0000000B”。SCLKp引脚的输入时钟F_{SCLK}(SSPI模式的从属传送)为传送时钟。

SDRmn寄存器的低8位或者低9位用作发送和接收缓冲寄存器。在接收数据时，将移位寄存器转换的并行数据保存到低8位或者低9位；在发送数据时，将被传送到移位寄存器的发送数据设置到低8位或者低9位。

能以16位为单位读写SDRmn寄存器。但是，只有在运行停止(SEmn=0)时才能读写高7位。在运行中(SEmn=1)只能写SDRmn寄存器的低8位或者低9位，而且SDRmn寄存器的高7位的读取值总是“0”。

在产生复位信号后，SDRmn寄存器的值变为“0000H”。

图19-9：串行数据寄存器mn(SDRmn)的格式



SDRmn[15:9]							运行时钟分频的传送时钟设置
0	0	0	0	0	0	0	F _{MCK} /2
0	0	0	0	0	0	1	F _{MCK} /4
0	0	0	0	0	1	0	F _{MCK} /6
0	0	0	0	0	1	1	F _{MCK} /8
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	F _{MCK} /254
1	1	1	1	1	1	1	F _{MCK} /256

注意:

1. 必须将SDR02、SDR03、SDR10、SDR11寄存器的bit8置“0”。
2. 在使用UART时，禁止将SDRmn[15:9]设置为“0000000B”和“0000001B”。
3. 在使用简易I²C时，禁止将SDRmn[15:9]设置为“0000000B”，SDRmn[15:9]的设置值必须大于等于“0000001B”。
4. 在运行停止(SEmn=0)时，禁止通过8位存储器操作指令改写SDRmn[7:0](否则，SDRmn[15:9]全部被清“0”)。

备注:

1. 有关SDRmn寄存器的低8位或者低9位的功能，请参照“19.2 通用串行通信单元的结构”。
2. m: 单元号(m=0、1)n: 通道号(n=0~3)

19.3.6 串行标志清除触发寄存器mn(SIRmn)

这是用于清除通道n各错误标志的触发寄存器。

如果将各位(FECTmn、PECTmn、OVCTmn)置“1”，就将串行状态寄存器mn(SSRmn)的对应位(FEFmn、PEFmn、OVFmn)清“0”。因为SDIRmn寄存器是触发寄存器，所以如果清除SSRmn寄存器的对应位，也会立即清除SDIRmn寄存器。

通过16位存储器操作指令设置SIRmn寄存器。

能用SIRmnL并且通过8位存储器操作指令设置SIRmn寄存器的低8位。

在产生复位信号后，SIRmn寄存器的值变为“0000H”。

图19-10：串行标志清除触发寄存器mn(SIRmn)的格式

地址：40041108H(SIR00)~4004110EH(SIR03) 复位后：0000H R/W

40041548H(SIR10)~4004154AH(SIR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn ^{注1}	PECTmn	OVCmn

FECTmn ^{注1}	通道n帧错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的FEFmn位清“0”。

PECTmn	通道n奇偶校验错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的PEFmn位清“0”。

OVCTmn	通道n溢出错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的OVFmn位清“0”。

注1：只限于SIR01、SIR03、SIR11寄存器。

注意：必须将bit15~3(SIR00、SIR02、SIR10寄存器为bit15~2)置“0”。

备注：

1. m：单元号(m=0、1)n：通道号(n=0~3)
2. SIRmn寄存器的读取值总是“0000H”。

19.3.7 串行状态寄存器mn(SSRmn)

SSRmn寄存器表示通道n的通信状态和发生错误的情况。表示的错误为帧错误、奇偶校验错误和溢出错误。通过16位存储器操作指令读取SSRmn寄存器。

能用SSRmnL并且通过8位存储器操作指令读取SSRmn寄存器的低8位。

在产生复位信号后，SSRmn寄存器的值变为“0000H”。

图19-11：串行状态寄存器mn(SSRmn)的格式(1/2)

地址：40041100H(SSR00)~40041106H(SSR03)

复位后：0000H R

40041540H(SSR10)~40041542H(SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{注1}	PEF mn	OVF mn

TSFmn	通道n通信状态的表示标志
0	通信停止状态或者通信待机状态
1	通信运行状态
[清除条件] •当将STm寄存器的STmn位置“1”(通信停止状态)或者将SSm寄存器的SSmn位置“1”(通信待机状态)时 •当通信结束时 [置位条件] •当开始通信时	

BFFmn	通道n缓冲寄存器的状态表示标志
0	SDRmn寄存器没有保存有效数据。
1	SDRmn寄存器保存了有效数据。
[清除条件] •在发送过程中传送完从SDRmn寄存器到移位寄存器的发送数据时 •在接收过程中从SDRmn寄存器读完接收数据时 •当将STm寄存器的STmn位置“1”(通信停止状态)或者将SSm寄存器的SSmn位置“1”(通信允许状态)时 [置位条件] •在SCRmn寄存器的TXEmn位为“1”(各通信模式中的发送模式、发送和接收模式)的状态下给SDRmn寄存器写发送数据时 •在SCRmn寄存器的RXEmn位为“1”(各通信模式中的接收模式、发送和接收模式)的状态下将接收数据保存到SDRmn寄存器时 •当发生接收错误时	

注1：只限于SSR01、SSR03、SSR11寄存器。

备注：m：单元号(m=0、1)n：通道号(n=0~3)

图19-11: 串行状态寄存器mn(SSRmn)的格式(2/2)

地址: 40041100H(SSR00)~40041106H(SSR03)

复位后: 0000H R

40041540H(SSR10)~40041542H(SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{注1}	PEF mn	OVF mn

FEFmn ^{注1}	通道n帧错误的检测标志
0	没有发生错误。
1	发生错误(UART接收时)。
[清除条件]	
• 当给SIRmn寄存器的FECTmn位写“1”时	
[置位条件]	
• 在UART接收结束时没有检测到停止位时	

PEFmn	通道n奇偶校验错误的检测标志
0	没有发生错误。
1	发生错误(UART接收时)或者未检测到ACK(I ² C发送时)。
[清除条件]	
• 当给SIRmn寄存器的PECTmn位写“1”时	
[置位条件]	
• 在UART接收结束时发送数据的奇偶校验和奇偶校验位不同(奇偶校验错误)时	
• 在I ² C发送时并且在ACK接收时序从属方没有返回ACK信号(未检测到ACK)时	

OVFmn	通道n溢出错误的检测标志
0	没有发生错误。
1	发生错误。
[清除条件]	
• 当给SIRmn寄存器的OVCTmn位写“1”时	
[置位条件]	
• 在SCRmn寄存器的RXEmn位为“1”(各通信模式中的接收模式、发送和接收模式)的状态下, 虽然接收数据被保存在SDRmn寄存器, 但是没有读接收数据而写发送数据或者写下一个接收数据时	
• 在SSPI模式的从属发送或者从属发送和接收过程中未准备好发送数据时	

注1: 只限于SSR01、SSR03、SSR11寄存器。

注意: 如果在BFFmn位为“1”时写SDRmn寄存器, 就会破坏被保存的发送或者接收数据, 并且检测到溢出错误(OVEmn=1)。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)

19.3.8 串行通道开始寄存器m(SSm)

SSm寄存器是设置允许各通道的通信/开始计数的触发寄存器。

如果给各位(SSmn)写“1”，就将串行通道允许状态寄存器m(SEm)的对应位(SEmn)置“1”(运行允许状态)。因为SSmn位是触发位，所以如果SEmn位为“1”就立即清除SSmn位。

通过16位存储器操作指令设置SSm寄存器。

能用SSmL并且通过8位存储器操作指令设置SSm寄存器的低8位。

在产生复位信号后，SSm寄存器的值变为“0000H”。

图19-12: 串行通道开始寄存器m(SSm)的格式

地址:	40041122H(SS0)				复位后: 0000H				R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

地址:	40041562H(SS1)				复位后: 0000H				R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	通道n运行开始的触发
0	没有触发。
1	将SEmn位置“1”，转移到通信待机状态 ^注 。

注：如果在通信过程中将SSmn位置“1”，就停止通信进入待机状态。此时，控制寄存器和移位寄存器的值、SCLKmn引脚和SDOmn引脚、FEFmn标志、PEFmn标志和OVFmn标志保持状态。

注意：

1. 必须将SS0寄存器的bit15~4、SS1寄存器的bit15~2置“0”。
2. 在UART接收时，必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个 F_{MCK} 时钟，然后将SSmn置“1”。

备注：

1. m: 单元号(m=0、1)n: 通道号(n=0~3)
2. SSm寄存器的读取值总是“0000H”。

19.3.9 串行通道停止寄存器m(STm)

STm寄存器是设置允许各通道的通信/停止计数的触发寄存器。

如果给各位(STmn)写“1”，就将串行通道允许状态寄存器m(SEm)的对应位(SEmn)清“0”(运行停止状态)。因为STmn位是触发位，所以如果SEmn位为“0”就立即清除STmn位。

通过16位存储器操作指令设置STm寄存器。

能用STmL并且通过8位存储器操作指令设置STm寄存器的低8位。

在产生复位信号后，STm寄存器的值变为“0000H”。

图19-13: 串行通道停止寄存器m(STm)的格式

地址: 40041124H(ST0)		复位后: 0000H								R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00	

地址: 40041564H(ST1)		复位后: 0000H								R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10	

STmn	通道n运行的停止触发
0	没有触发。
1	将SEmn位清“0”，停止通信运行 ^注 。

注：控制寄存器和移位寄存器的值、SCLKmn引脚和SDOmn引脚以及FEFmn标志、PEFmn标志和OVFmn标志保持状态。

注意：必须将ST0寄存器的bit15~4、ST1寄存器的bit15~2置“0”。

备注：

1. m: 单元号(m=0、1)n: 通道号(n=0~3)
2. STm寄存器的读取值总是“0000H”。

19.3.10 串行通道允许状态寄存器m(SEm)

SEm寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

如果给串行开始允许寄存器m(SSm)的各位写“1”，就将其对应位置“1”。如果给串行通道停止寄存器m(STm)的各位写“1”，就将其对应位清“0”。

对于允许运行的通道n，无法通过软件改写后述的串行输出寄存器m(SOm)的CKOmn位(通道n的串行时钟输出)的值，而从串行时钟引脚输出由通信运行反映的值。

对于停止运行的通道n，能通过软件设置SOm寄存器的CKOmn位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令读取SEm寄存器。

能用SEmL并且通过8位存储器操作指令读取SEm寄存器的低8位。

在产生复位信号后，SEm寄存器的值变为“0000H”。

图19-14：串行通道允许状态寄存器m(SEm)的格式

地址：40041120H(SE0)		复位后：0000H		R												
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

地址：40041560H(SE1)		复位后：0000H		R												
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE10

SEmn	通道n运行的允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

备注：m：单元号(m=0、1)n：通道号(n=0~3)

19.3.11 串行输出允许寄存器m(SOEm)

SOEm寄存器设置允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道n，无法通过软件改写后述的串行输出寄存器m(SOm)的SOmn位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道n，能通过软件设置SOm寄存器的SOmn位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令设置SOEm寄存器。

能用SOEmL并且通过8位存储器操作指令设置SOEm寄存器的低8位。

在产生复位信号后，SOEm寄存器的值变为“0000H”。

图19-15: 串行输出允许寄存器m(SOEm)的格式

地址: 4004112AH	复位后: 0000H	R/W																			
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00				

地址: 4004156AH	复位后: 0000H	R/W																
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE11	SOE10		

SOE mn	通道n串行输出的允许或者停止
0	停止串行通信的输出。
1	允许串行通信的输出。

注意：必须将SOE0寄存器的bit15~4、SOE1寄存器的bit15~2置“0”。

备注：m：单元号(m=0、1)n：通道号(n=0~3)

19.3.12 串行输出寄存器m(SOm)

SOm寄存器是各通道串行输出的缓冲寄存器。

从通道n的串行数据输出引脚输出此寄存器的SOmn位的值。

从通道n的串行时钟输出引脚输出此寄存器的CKOmn位的值。

只有在禁止串行输出时(SOEmn=0)才能通过软件改写此寄存器的SOmn位。当允许串行输出(SOEmn=1)时，忽视通过软件的改写而只能通过串行通信更改此寄存器的SOmn位的值。

只有在通道停止运行时(SEmn=0)才能通过软件改写此寄存器的CKOmn位。当允许通道运行(SEmn=1)时，忽视通过软件的改写而只能通过串行通信更改此寄存器的CKOmn位的值。

要将串行接口的引脚用作端口功能等非串行接口功能时，必须将相应的CKOmn位和SOmn位置“1”。

通过16位存储器操作指令设置SOm寄存器。

在产生复位信号后，SOm寄存器的值变为“0F0FH”。

图19-16: 串行输出寄存器m(SOm)的格式

地址: 40041128H		复位后: 0F0FH		R/W																												
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
SO0	0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00																

地址: 40041568H		复位后: 0303H		R/W																												
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
SO1	0	0	0	0	0	0	CKO11	CKO10	0	0	0	0	0	0	SO11	SO10																

CKOmn	通道n的串行时钟输出															
0	串行时钟的输出值为“0”。															
1	串行时钟的输出值为“1”。															

SOmn	通道n的串行数据输出															
0	串行数据的输出值为“0”。															
1	串行数据的输出值为“1”。															

注意: 必须将SO0寄存器的bit15~12和bit7~4置“0”。

必须将SO1寄存器的bit15~10和bit7~2置“0”。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)

19.3.13 串行输出电平寄存器m(SOLm)

SOLm寄存器是设置各通道的数据输出电平反相的寄存器。

只有在UART模式中才能设置此寄存器。在SSPI模式和简易I2C模式中，必须将对应位置“0”。只在允许串行输出时(SOEmn=1)，将此寄存器的各通道n反相设置反映到引脚输出。在禁止串行输出时(SOEmn=0)，将SOmn位的值直接输出。禁止在运行过程中(SEmn=1)改写SOLm寄存器。

通过16位存储器操作指令设置SOLm寄存器。

能用SOLmL并且通过8位存储器操作指令设置SDOLm寄存器的低8位。

在产生复位信号后，SOLm寄存器的值变为“0000H”。

图19-17：串行输出电平寄存器m(SOLm)的格式

地址：40041134H		复位后：0000H												R/W		
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02	0	SOL00

地址：40041574H		复位后：0000H												R/W		
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL10

SOLmn	UART模式中的通道n发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

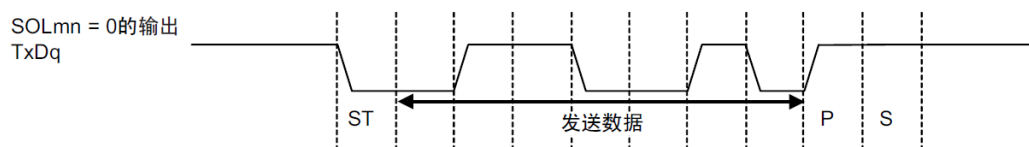
注意：必须将SOL0寄存器的bit15~3和bit1、SOL1寄存器的bit15~1置“0”。

备注：m：单元号(m=0、1)n：通道号(n=0、2)

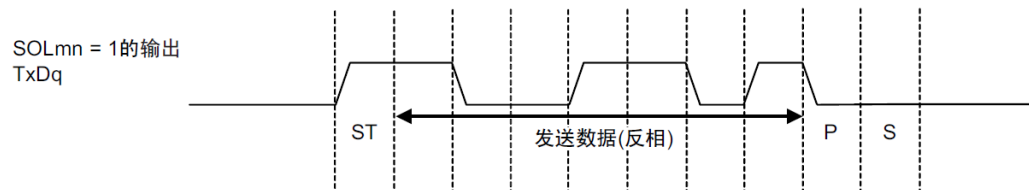
当进行UART发送时，发送数据的电平反相例子如图19-18所示。

图19-18：发送数据的电平反相例子

(a) 正相输出 (SOLmn = 0)



(b) 反相输出 (SOLmn = 1)



备注：m：单元号(m=0、1)n：通道号(n=0、2)

19.3.14 输入切换控制寄存器(ISC)

在通过UART0实现LIN-bus通信时，ISC寄存器的ISC1位和ISC0位用于外部中断和定时器阵列单元的协调。如果将bit0置“1”，就选择串行数据输入(RxD0)引脚的输入信号作为外部中断的输入(INTP0)，因此能通过INTP0中断检测唤醒信号。

如果将bit1置“1”，就选择串行数据输入(RxD0)引脚的输入信号作为定时器的输入，因此能通过定时器检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽。

SS1E00位在SSPI00通信的从属模式中控制通道0的SS00引脚输入。在给SS00引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收；在给SS00引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设置进行发送和接收。

通过8位存储器操作指令设置ISC寄存器。

在产生复位信号后，ISC寄存器的值变为“00H”。

图19-19：输入切换控制寄存器(ISC)的格式

地址：40040473H 复位后：00H	R/W							
符号	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0
SSDIE00	SSPI00通信的从属模式中通道0的SS00输入的设置							
0	SS00引脚输入无效。							
1	SS00引脚输入有效。							
ISC1	定时器Timer4的通道3的输入切换							
0	将TI03引脚的输入信号用作定时器的输入(通常运行)。							
1	将RxD0引脚的输入信号用作定时器的输入(检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽)。							
ISC0	外部中断(INTP0)的输入切换							
0	将INTP0引脚的输入信号用作外部中断的输入(通常运行)。							
1	将RxD0引脚的输入信号用作外部中断的输入(检测唤醒信号)。							

注意：必须将bit6~2置“0”。

19.3.15 噪声滤波器允许寄存器0(NFEN0)

NFEN0寄存器设置噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于SSPI或者简易I²C通信的引脚，必须将对应位置“0”，使噪声滤波器无效。对于用于UART通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟(F_{MCK})进行同步后检测2个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟(F_{MCK})进行同步。

通过8位存储器操作指令设置NFEN0寄存器。

在产生复位信号后，NFEN0寄存器的值变为“00H”。

图19-20: 噪声滤波器允许寄存器0(NFEN0)的格式

地址: 40040470H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	RxD2引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD2引脚时，必须将SNFEN20位置“1”。 当用作RxD2引脚以外的功能时，必须将SNFEN20位置“0”。	

SNFEN10	RxD1引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD1引脚时，必须将SNFEN10位置“1”。 当用作RxD1引脚以外的功能时，必须将SNFEN10位置“0”。	

SNFEN00	RxD0引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD0引脚时，必须将SNFEN00位置“1”。 当用作RxD0引脚以外的功能时，必须将SNFEN00位置“0”。	

注意：必须将bit7~5、3、1置“0”。

19.3.16 控制串行输入/输出引脚端口功能的寄存器

在使用通用串行通信单元时，必须设置与对象通道复用的端口功能的控制寄存器(端口模式寄存器(PMxx)、端口寄存器(Pxx)、端口输入模式寄存器(PIMxx)、端口输出模式寄存器(POMxx)和端口模式控制寄存器(PMCxx))。

详细内容请参照“2.3.1 端口模式寄存器(PMxx)”、“2.3.2 端口寄存器(Pxx)”、“2.3.4 端口输入模式寄存器(PIMxx)”、“2.3.5 端口输出模式寄存器(POMxx)”和“2.3.6 端口模式控制寄存器(PMCxx)”。

在将串行数据输出引脚或者串行时钟输出引脚的复用端口用作串行数据输出或者串行时钟输出时，必须将各端口对应的端口模式控制寄存器(PMCxx)的位和端口模式寄存器(PMxx)的位置“0”，并且将端口寄存器(Pxx)的位置“1”。

另外，当用于N沟道漏极开路输出模式时，必须将各端口对应的端口输出模式寄存器(POMxx)的位置“1”。

(例)P02用作串行数据输出的情况将端口模式控制寄存器0的PMC02位置“0”。将端口模式寄存器0的PM02位置“0”。将端口寄存器0的P02位置“1”。

在将串行数据输入引脚或者串行时钟输入引脚的复用端口用作串行数据输入或者串行时钟输入时，必须将各端口对应的端口模式寄存器(PMxx)的位置“1”，并且将端口模式控制寄存器(PMCxx)的位置“0”。此时，端口寄存器(Pxx)的位可以是“0”或者“1”。

另外，当用作TTL输入缓冲器时，必须将各端口对应的端口输入模式寄存器(PIMxx)的位置“1”。

(例)P03用作串行数据输入的情况将端口模式控制寄存器0的PMC03位置“0”。将端口模式寄存器0的PM03位置“1”。将端口寄存器0的P03位置“0”或者“1”。

19.4 运行停止模式

通用串行通信单元的各串行接口有运行停止模式。在运行停止模式中不能进行串行通信，因此能降低功耗。另外，在运行停止模式中能将用于串行接口的引脚用作端口功能。

19.4.1 以单元为单位停止运行的情况

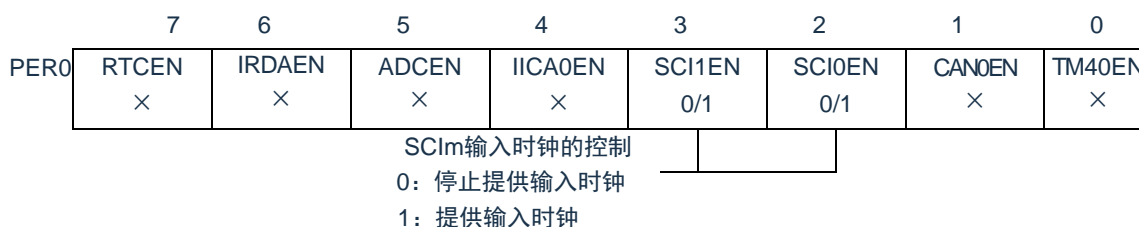
通过外围允许寄存器0(PER0)设置以单元为单位的停止运行。

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过给不使用的硬件停止提供时钟，以降低功耗和噪声。

要停止通用串行通信单元0时，必须将bit2(SCI0EN)置“0”；要停止通用串行通信单元1时，必须将bit3(SCI1EN)置“0”。

图19-21：以单元为单位停止运行时的外围允许寄存器0(PER0)的设置

(a)外围允许寄存器0(PER0).....只将要停止SCI_m的对应位置“0”。



注意：当SCI_mEN位为“0”时，忽视通用串行通信单元m的控制寄存器的写操作，而且读取值都为初始值。

但是，以下的寄存器除外：

- 输入切换控制寄存器(ISC)
- 噪声滤波器允许寄存器0(NFEN0)
- 端口输入模式寄存器(PIMx)
- 端口输出模式寄存器(POMx)
- 端口模式寄存器(PMx)
- 端口寄存器(Px)

备注：×：这是通用串行通信单元未使用的位(取决于其他外围功能的设置)。

0/1：根据用户的用途置“0”或者“1”。

19.5 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信的运行

这是通过串行时钟(SCLK)和串行数据(SDI和SDO)共3条线实现的时钟同步通信功能。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率^注

主控通信：最大值 $F_{CLK}/2$ (只限于SSPI00)

主控通信：最大值 $F_{CLK}/4$

从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注：必须在满足SCLK周期时间(T_{KCY})特性的范围内使用。详细内容请参照数据手册。

SCI0的通道0~3以及SCI1的通道0~1是支持3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)的通道。

3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)有以下6种通信运行：

- 主控发送 (参照19.5.1)
- 主控接收 (参照19.5.2)
- 主控的发送和接收 (参照19.5.3)
- 从属发送 (参照19.5.4)
- 从属接收 (参照19.5.5)
- 从属的发送和接收 (参照19.5.6)

19.5.1 主控发送

主控发送是指BAT32A237输出传送时钟并且将数据发送到其他设备的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDO00	SCLK01 SDO01	SCLK10 SDO10	SCLK11 SDO11	SCLK20 SDO20	SCLK21 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。					
错误检测标志	无					
传送数据长度	7位或者8位					
传送速率 ^注	Max.F _{CLK} /2[Hz](只限于SSPI00), F _{CLK} /4[Hz] Min.F _{CLK} /(2x2 ¹⁵ x128)[Hz] F _{CLK} : 系统时钟频率					
数据相位	能通过SCR _{mn} 寄存器的DAP _{mn} 位进行选择。 • DAP _{mn} =0: 在串行时钟开始运行时, 开始数据输出。 • DAP _{mn} =1: 在串行时钟开始运行的半个时钟前, 开始数据输出。					
时钟相位	能通过SCR _{mn} 寄存器的CKP _{mn} 位进行选择。 • CKP _{mn} =0: 正相 • CKP _{mn} =1: 反相					
数据方向	MSB优先或者LSB优先					

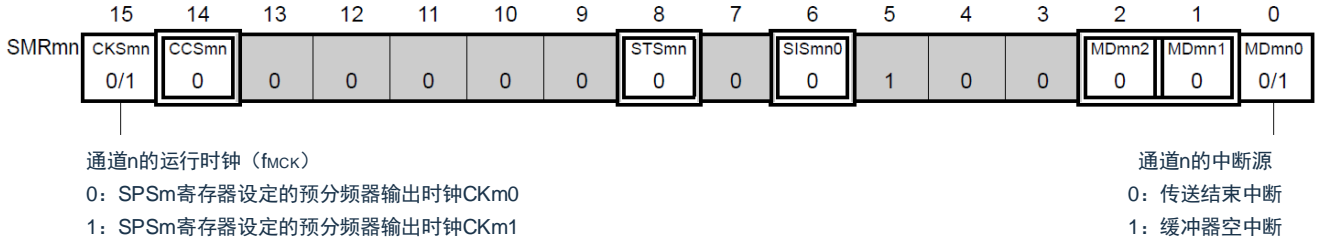
注: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

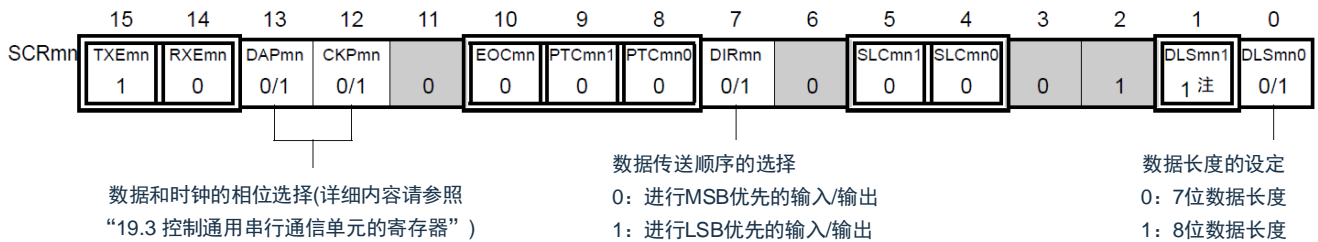
(1) 寄存器的设置

图19-23: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控发送时的寄存器设置内容例子

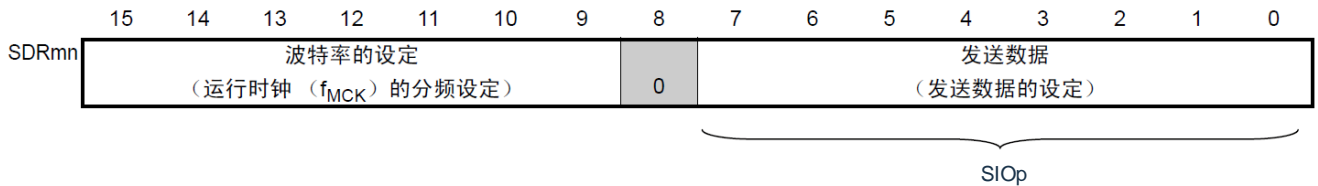
(a) 串行模式寄存器mn(SMRmn)



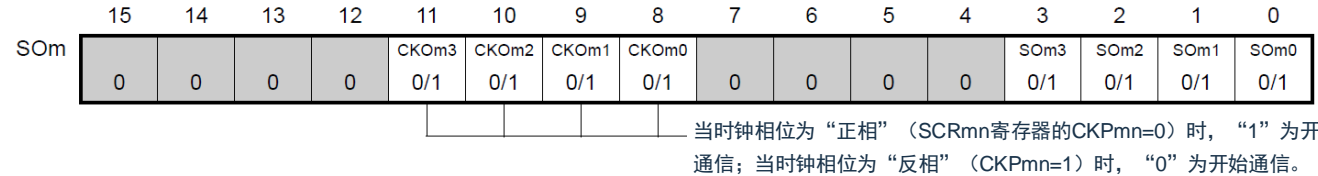
(b) 串行通信运行设定寄存器mn(SCRmn)



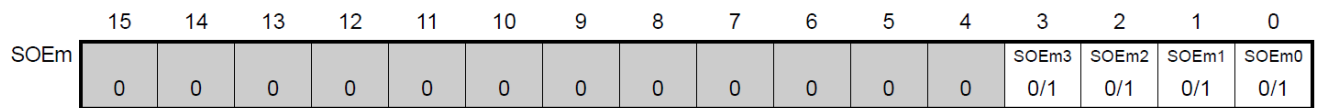
(c) 串行数据寄存器mn(SDRmn)



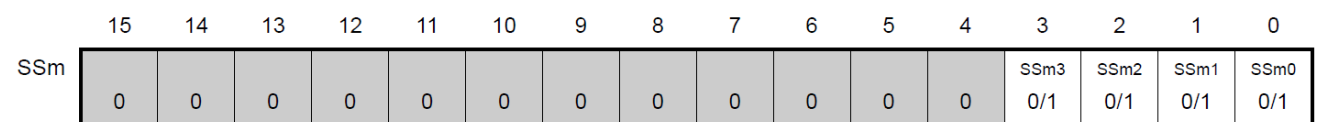
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



注: 只限于SCR00, SCR01寄存器,其他固定为“1”。

备注:

1. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11
2. □: 不能设置(设置初始值)。0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-24：主控发送的初始设置步骤

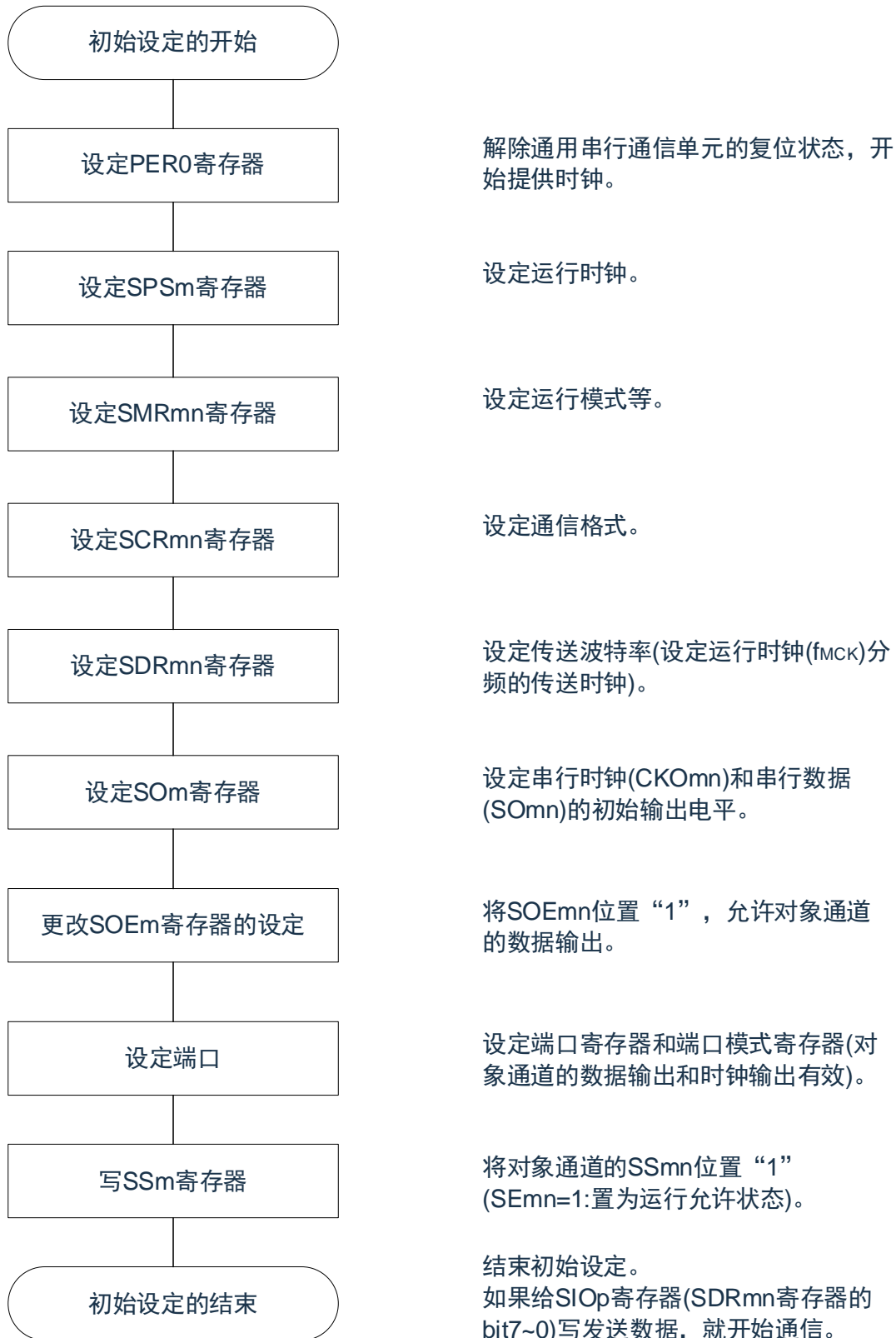


图19-25: 主控发送的中止步骤

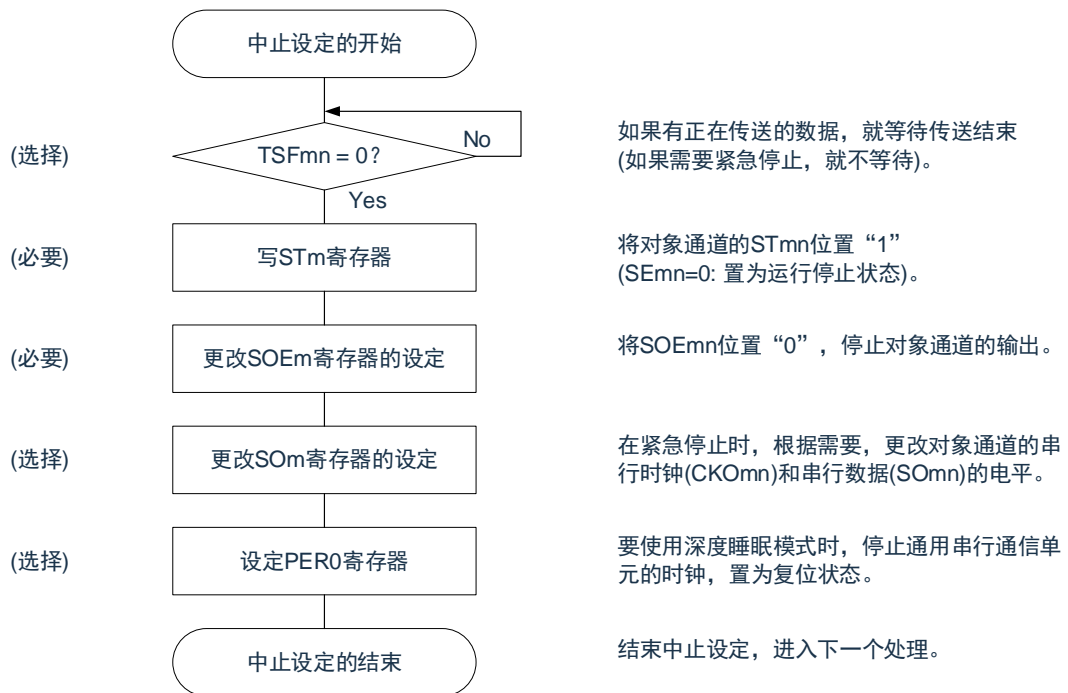
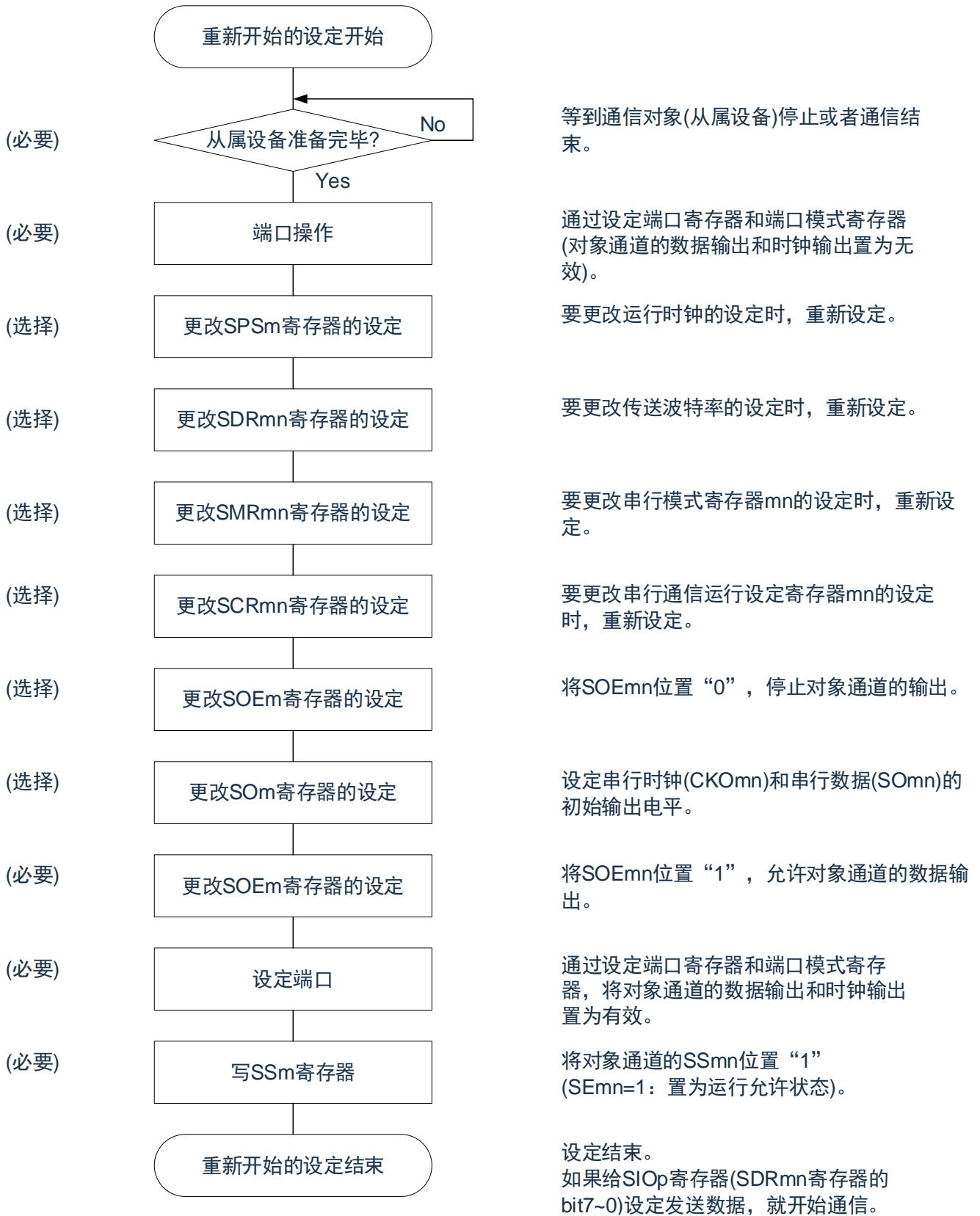


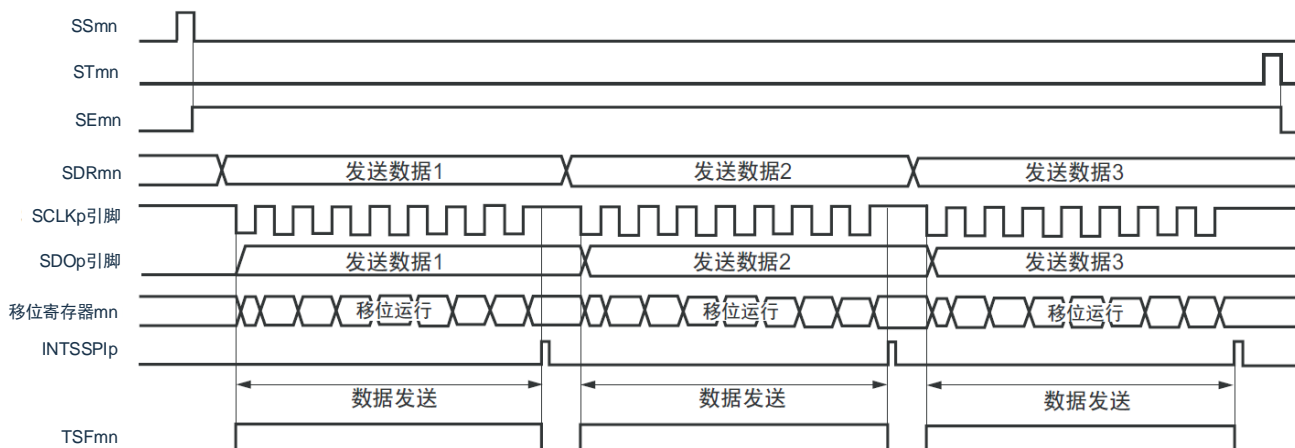
图19-26: 重新开始主控发送的设置步骤



备注: 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象(从属设备)停止或者通信结束后进行初始设置而不是进行重新开始设置。

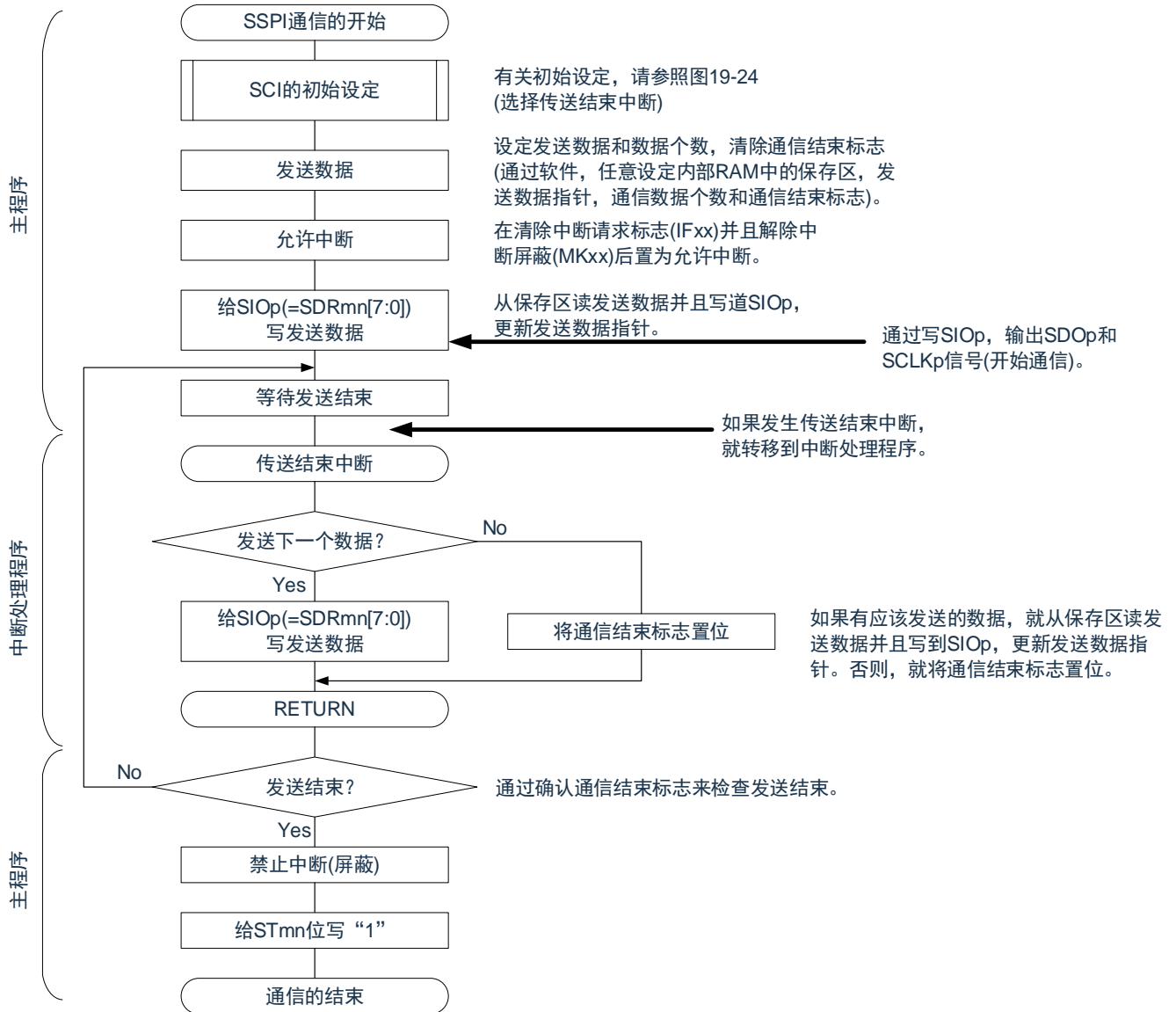
(3) 处理流程(单次发送模式)

图19-27: 主控发送(单次发送模式)的时序图(类型1: DAPmn=0、CKPmn=0)



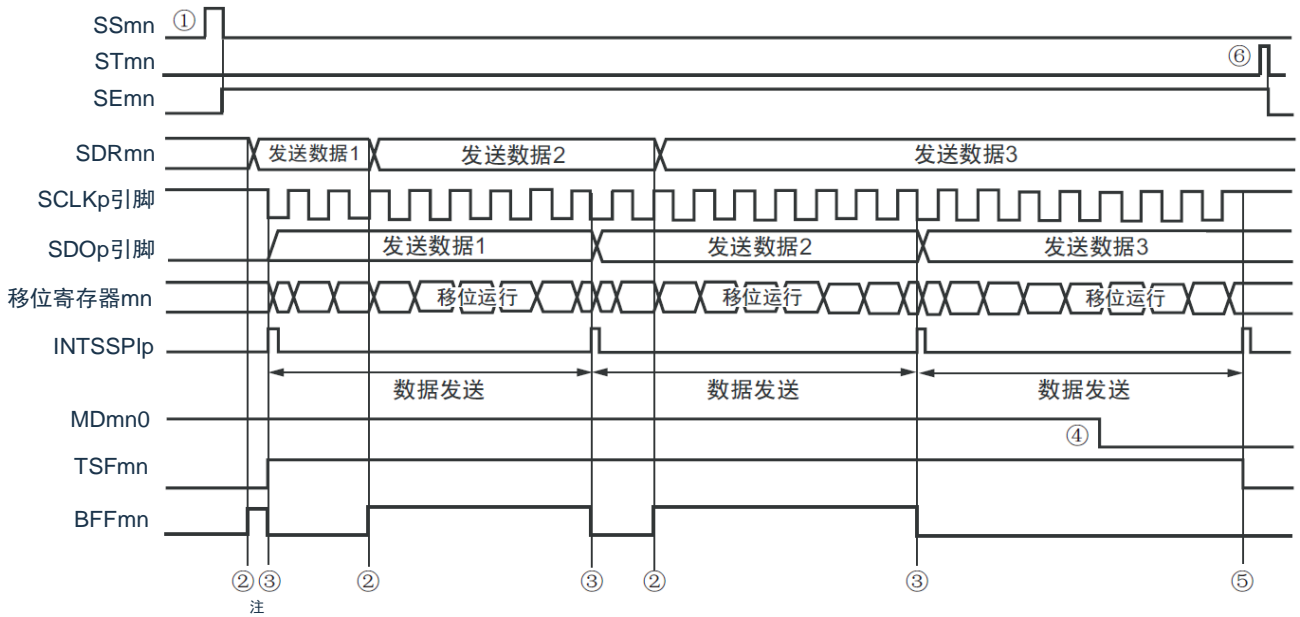
备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-28: 主控发送(单次发送模式)的流程图



(4) 处理流程(连续发送模式)

图19-29: 主控发送(连续发送模式)的时序图(类型1: DAPmn=0、CKPmn=0)

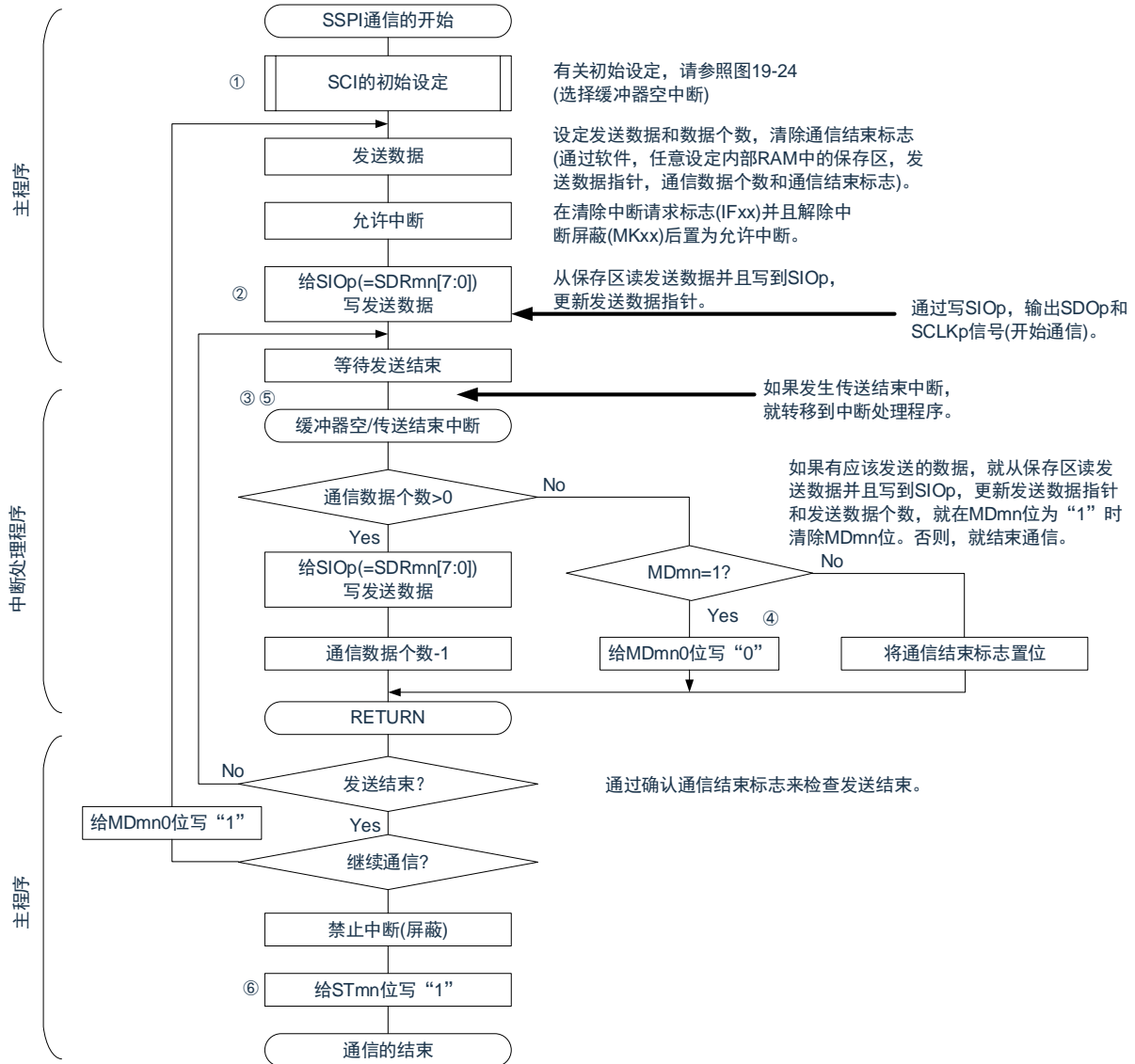


注: 如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据, 就重写发送数据。

注意: 即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是, 为了能赶上最后发送数据的传送结束中断, 必须在开始传送最后一位之前进行改写。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-30: 主控发送(连续发送模式)的流程图



备注: 图中的①~⑥对应“图19-29: 主控发送(连续发送模式)的时序图”中的①~⑥。

19.5.2 主控接收

主控接收是指BAT32A237输出传送时钟并且从其他设备接收数据的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDI00	SCLK01 SDI01	SCLK10 SDI10	SCLK11 SDI11	SCLK20 SDI20	SCLK21 SDI21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。					
错误检测标志	只有溢出错误检测标志(OVFmn)。					
传送数据长度	7位或者8位					
传送速率 ^注	Max.F _{CLK} /2[Hz](只限于SSPI00), F _{CLK} /4[Hz] Min.F _{CLK} /(2x2 ¹⁵ x128)[Hz] F _{CLK} : 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 					
数据方向	MSB优先或者LSB优先					

注: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

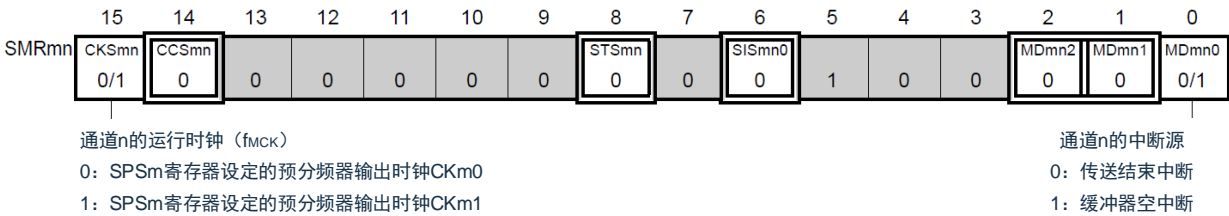
备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)

mn=00~03、10~11

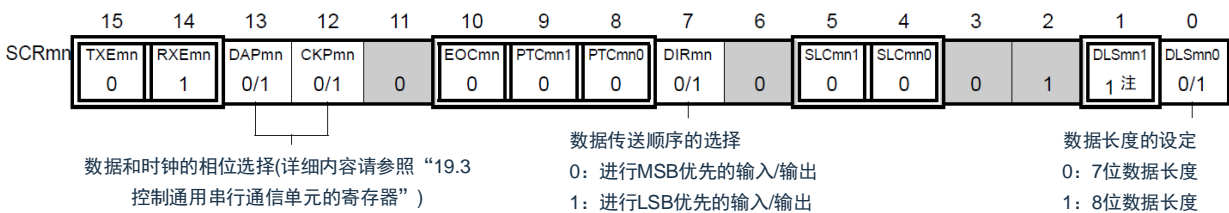
(1) 寄存器的设置

图19-31: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控接收时的寄存器设置内容例子

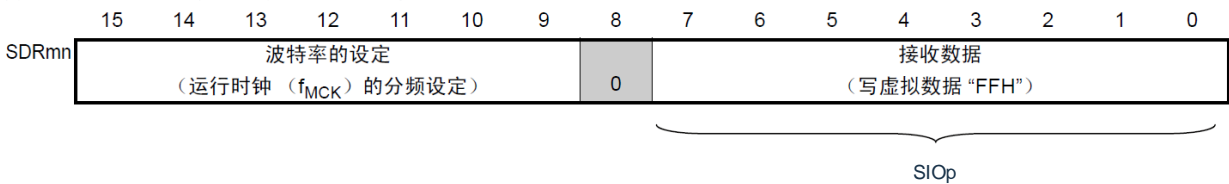
(a) 串行模式寄存器mn(SMRmn)



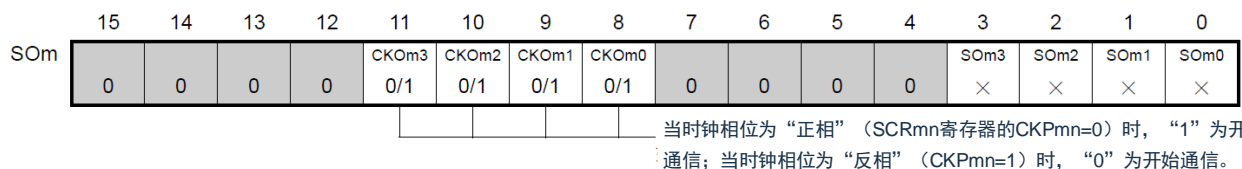
(b) 串行通信运行设定寄存器mn(SCRmn)



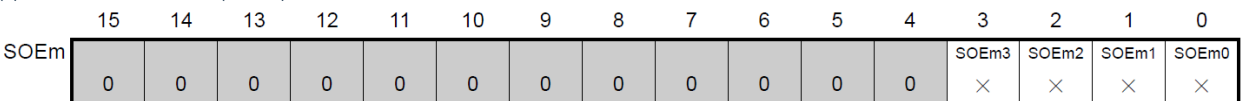
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



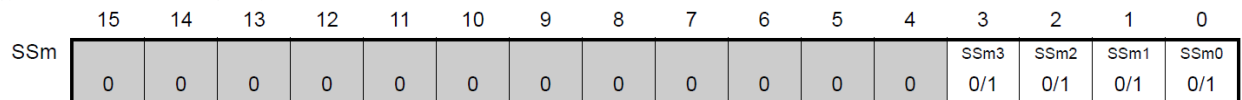
(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ...在此模式中不使用



(f) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。



注: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11
- : 在SSPI主控接收模式中为固定设置。■: 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-32: 主控接收的初始设置步骤

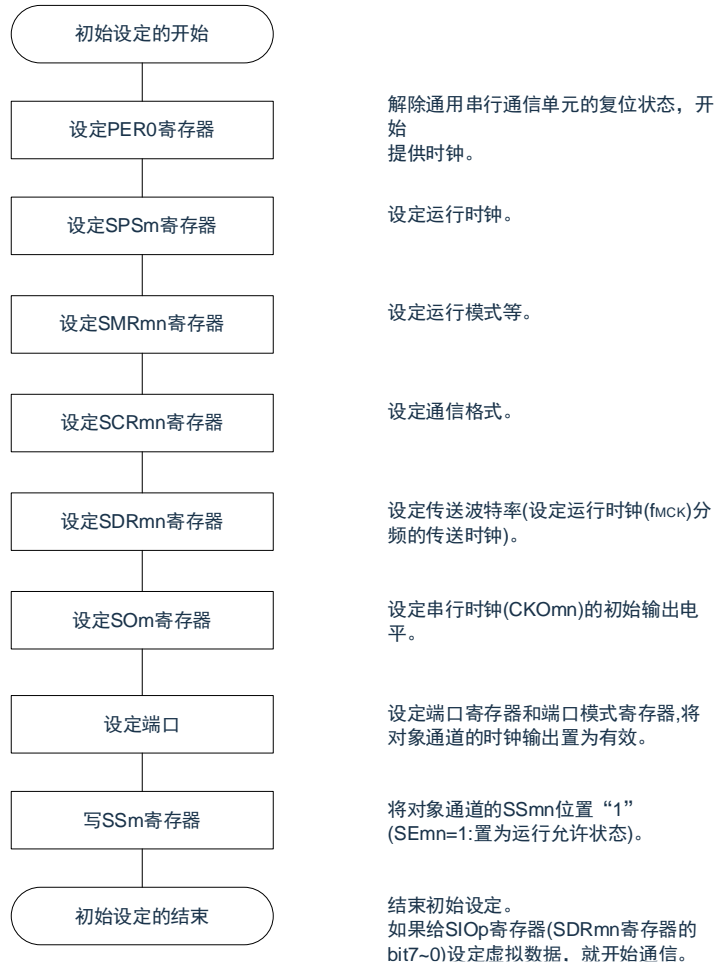


图19-33: 主控接收的中止步骤

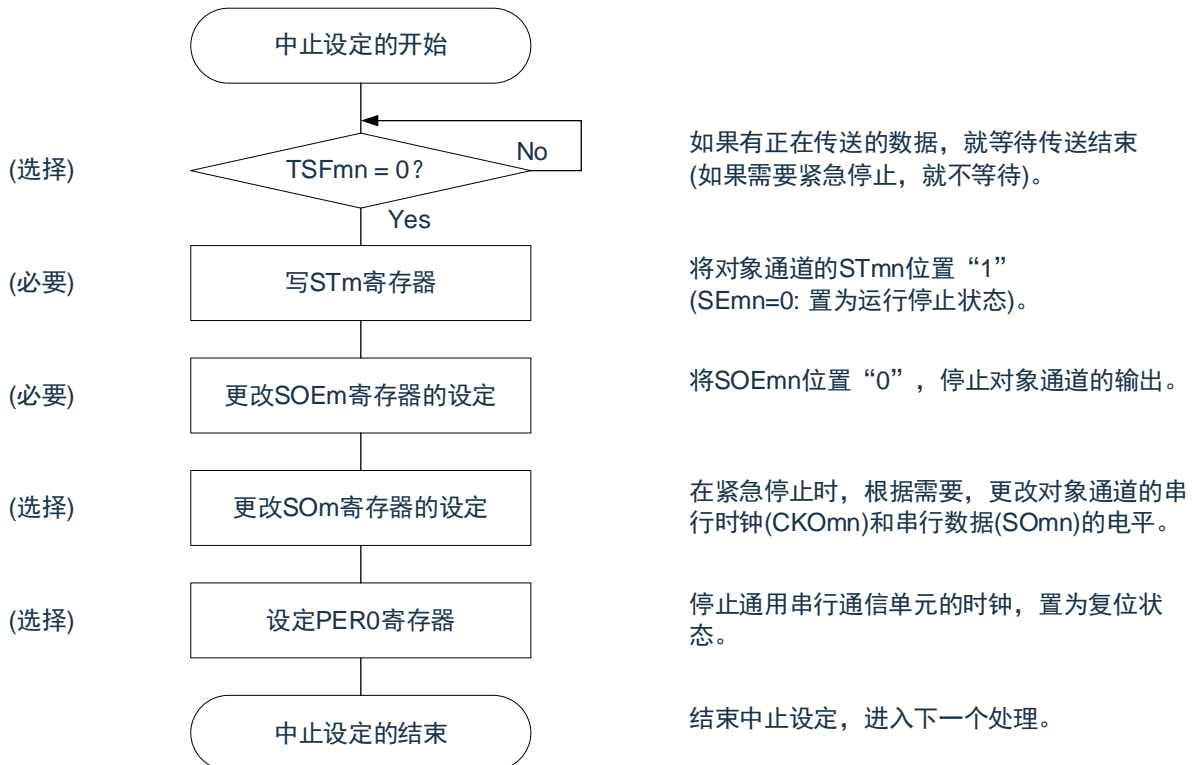


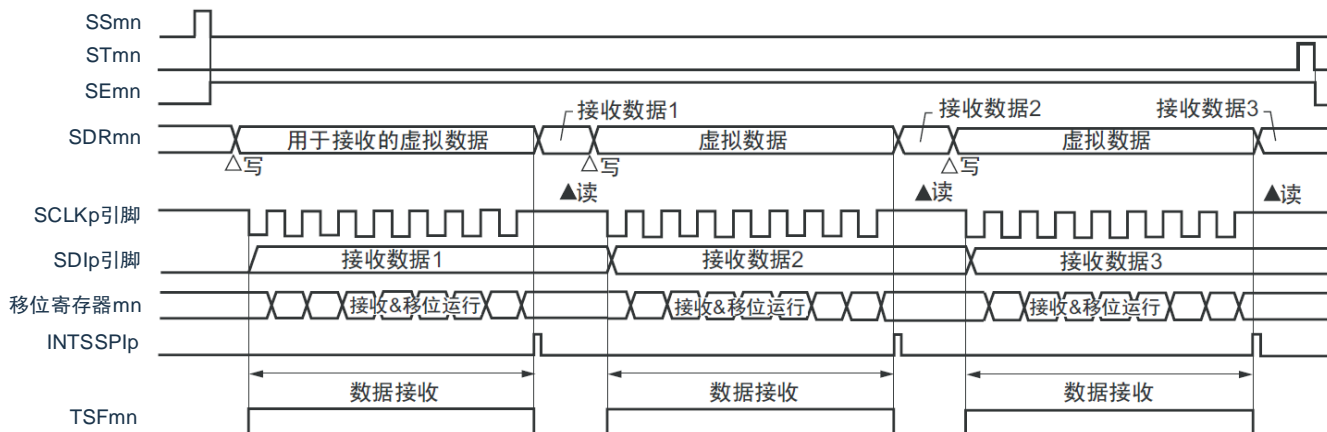
图19-34：重新开始主控接收的设置步骤



备注：如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象(从属设备)停止或者通信结束后进行初始设置而不是进行重新开始设置。

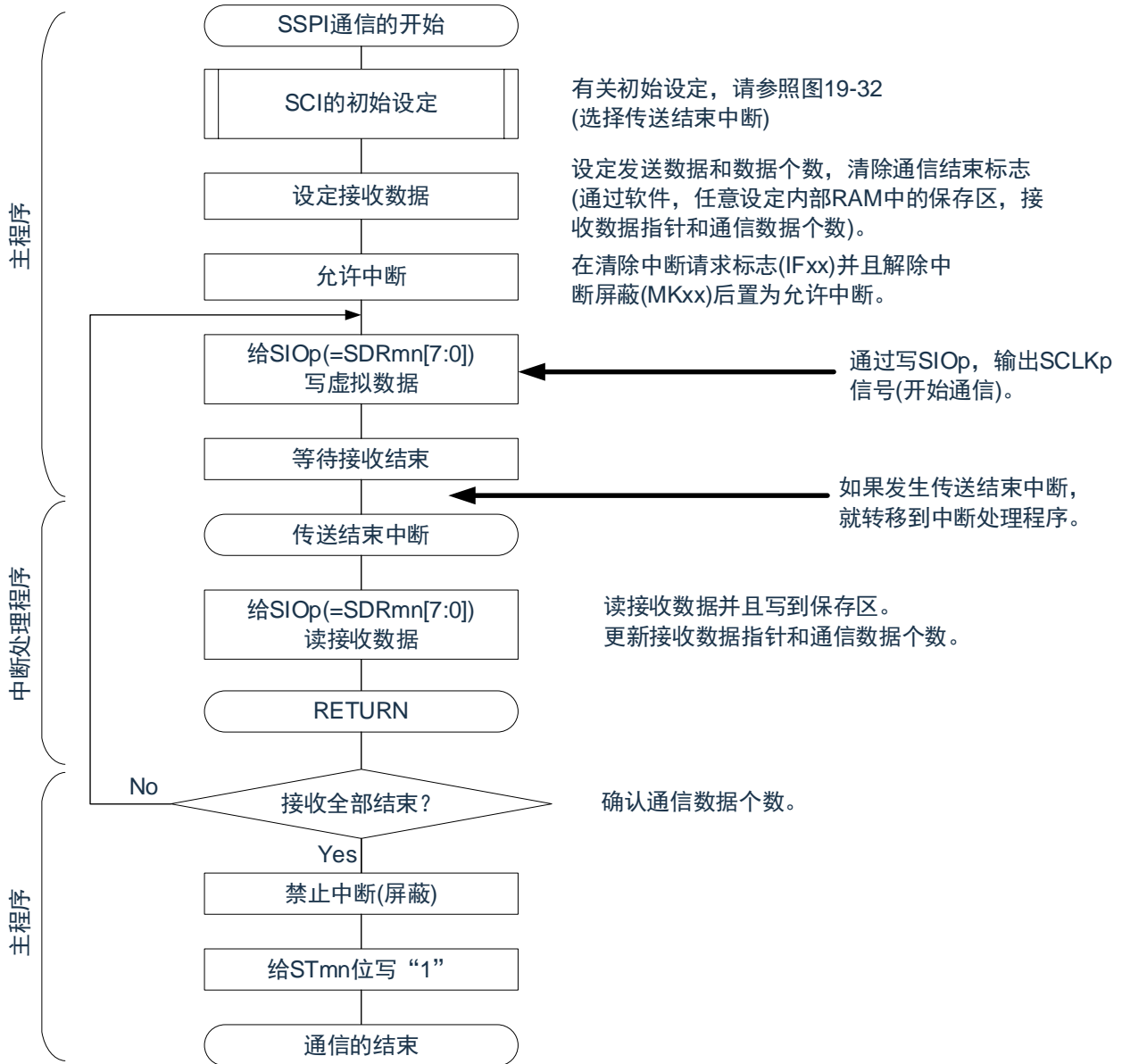
(3) 处理流程(单次接收模式)

图19-35: 主控接收(单次接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



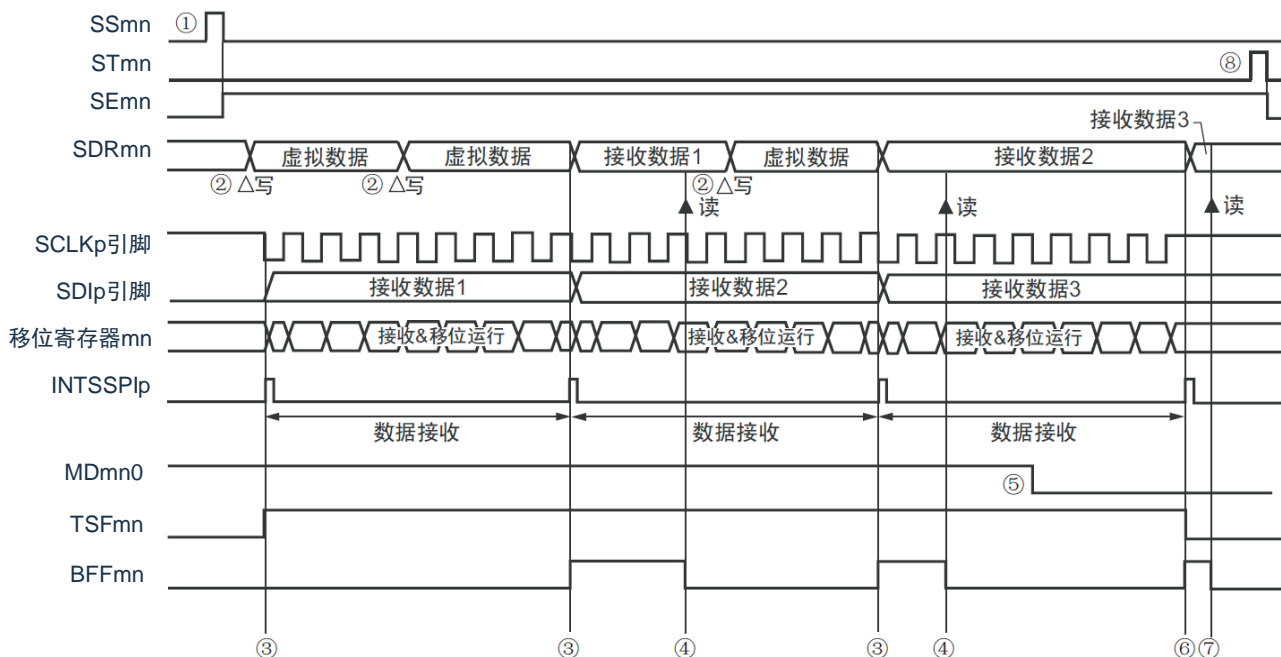
备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-36: 主控接收(单次接收模式)的流程图



(4) 处理流程(连续接收模式)

图19-37：主控接收(连续接收模式)的时序图(类型1：DAPmn=0、CKPmn=0)

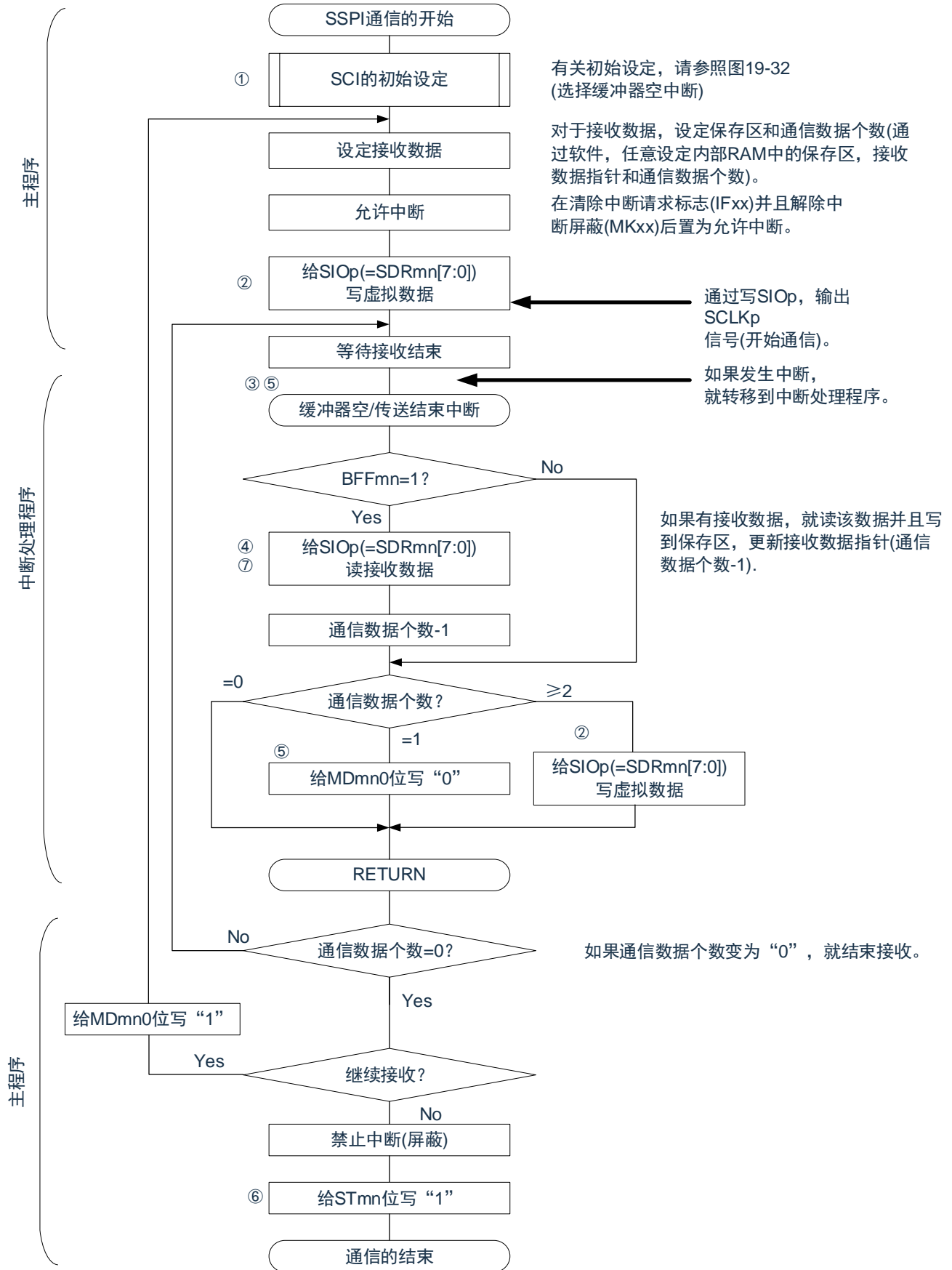


注意：即使在运行过程中也能改写MDmn0位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注：

1. 图中的①~⑧对应“图19-38：主控接收(连续接收模式)的流程图”中的①~⑧。
2. m：单元号(m=0、1)n：通道号(n=0~3)p：SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-38: 主控接收(连续接收模式)的流程图



备注: 图中的①~⑧对应“图19-37: 主控接收(连续接收模式)的时序图”中的①~⑧。

19.5.3 主控的发送和接收

主控的发送和接收是指BAT32A237输出传送时钟并且和其他设备进行数据发送和接收的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDI00 SDO00	SCLK01 SDI01 SDO01	SCLK10 SDI10 SDO10	SCLK11 SDI11 SDO11	SCLK20 SDI20 SDO20	SCLK21 SDI21 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位或者8位					
传送速率注	Max.F _{CLK} /2[Hz]（只限于SSPI00），F _{CLK} /4[Hz] Min.F _{CLK} /(2×2 ¹⁵ ×128)[Hz] F _{CLK} : 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时，开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

注：必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

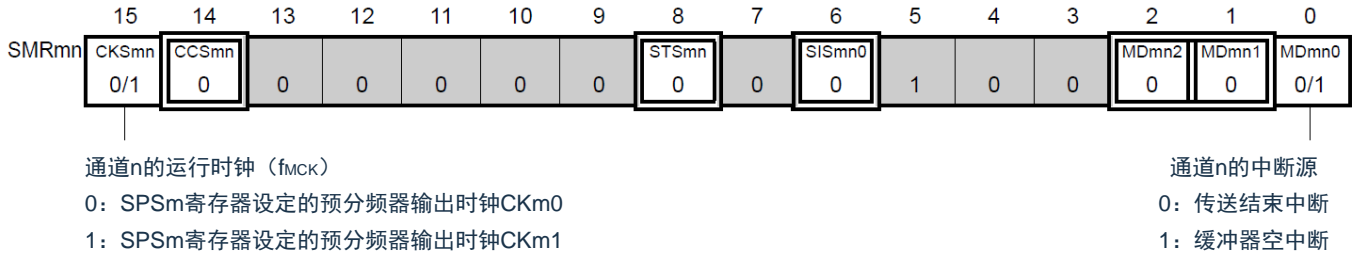
备注：m：单元号(m=0、1)n：通道号(n=0~3)p：SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

(1) 寄存器的设置

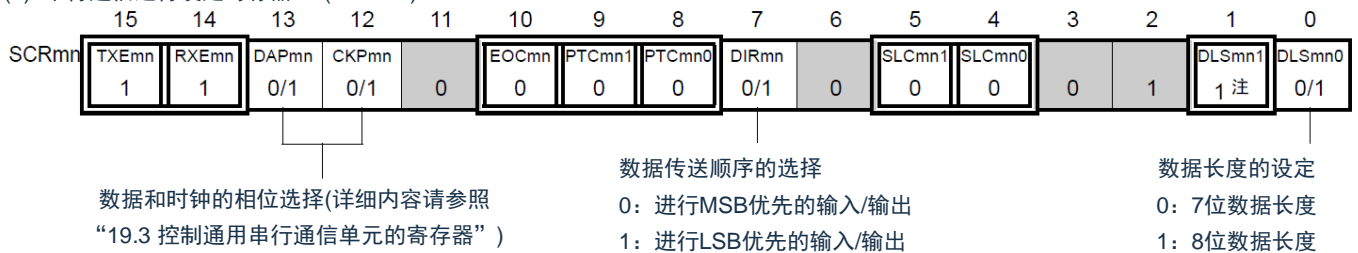
图19-39: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

主控发送和接收时的寄存器设置内容例子

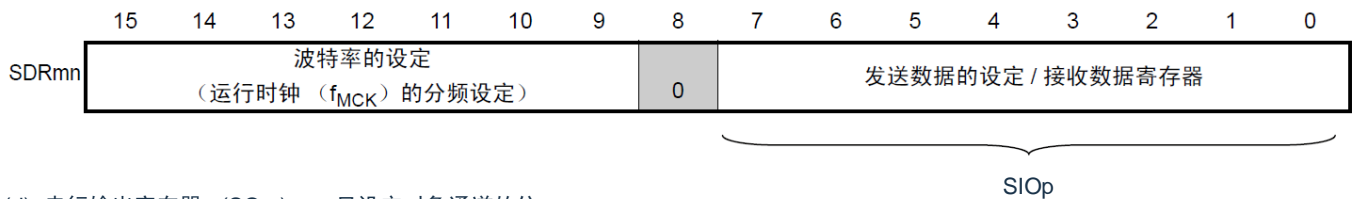
(a) 串行模式寄存器mn(SMRmn)



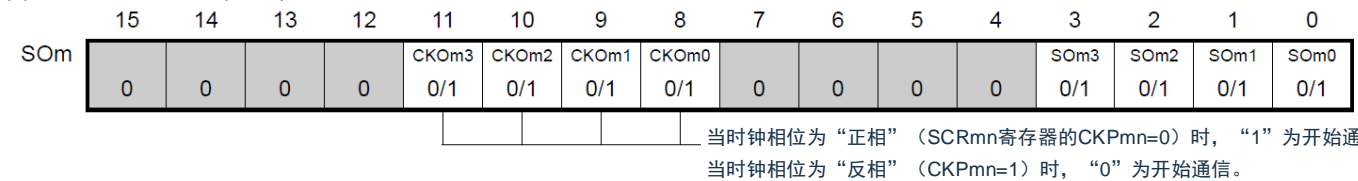
(b) 串行通信运行设定寄存器mn(SCRmn)



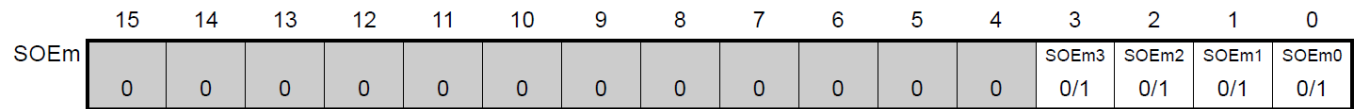
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



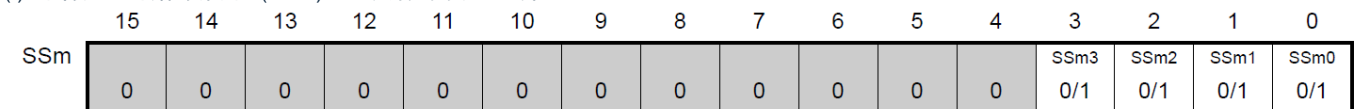
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



注: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11
- : 在SSPI主控发送和接收模式中为固定设置。■: 不能设置(设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-40: 主控发送和接收的初始设置步骤

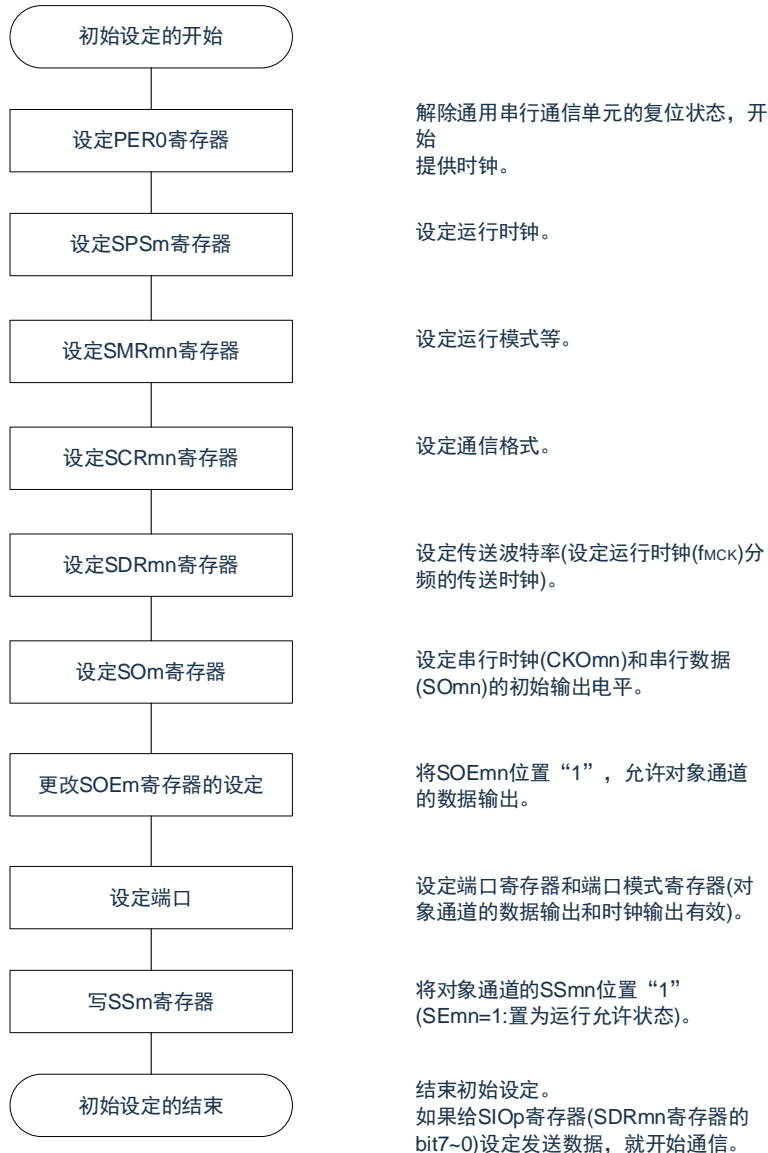


图19-41: 主控发送和接收的中止步骤

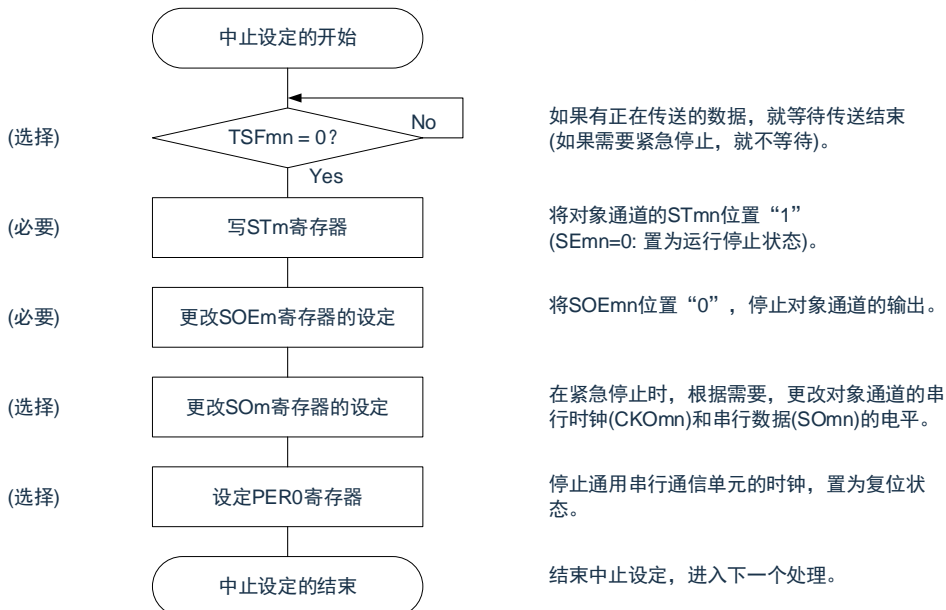
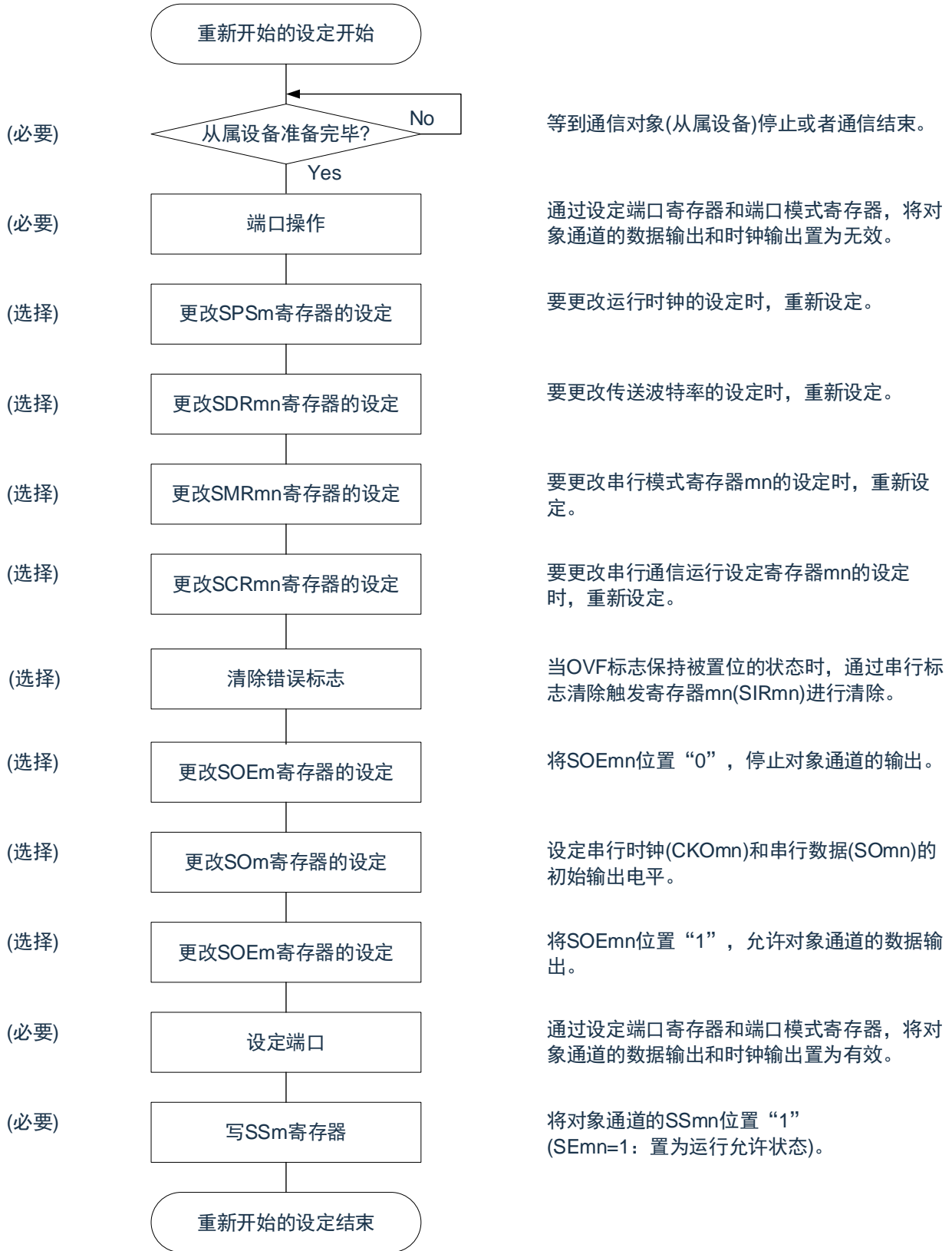
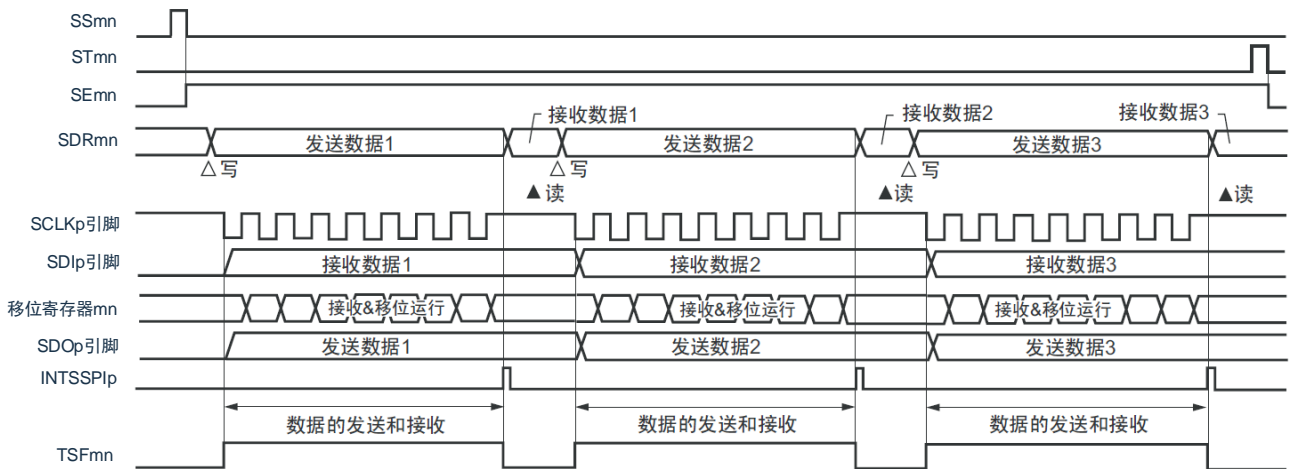


图19-42: 重新开始主控发送和接收的设置步骤



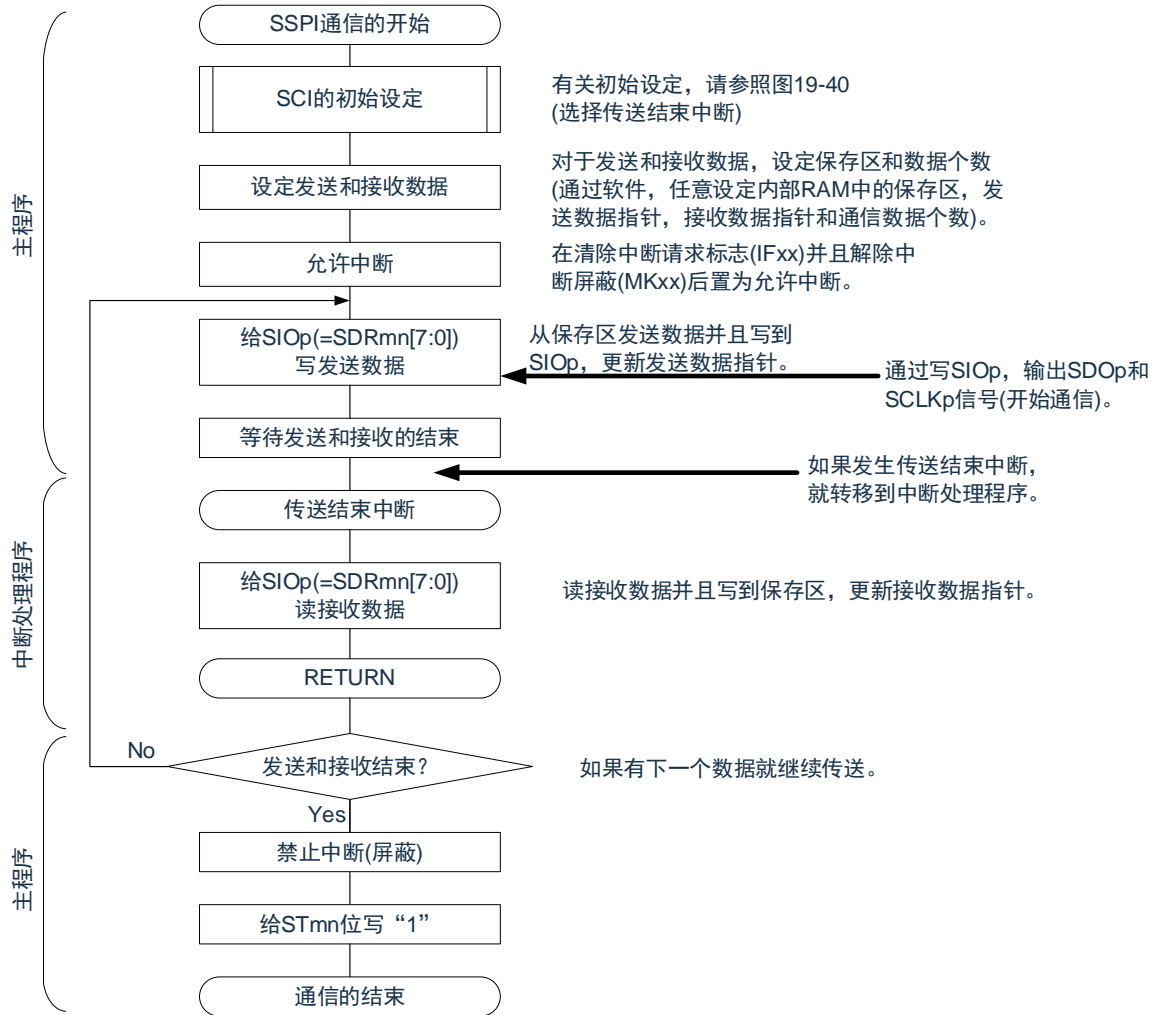
(3) 处理流程(单次发送和接收模式)

图19-43: 主控发送和接收(单次发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



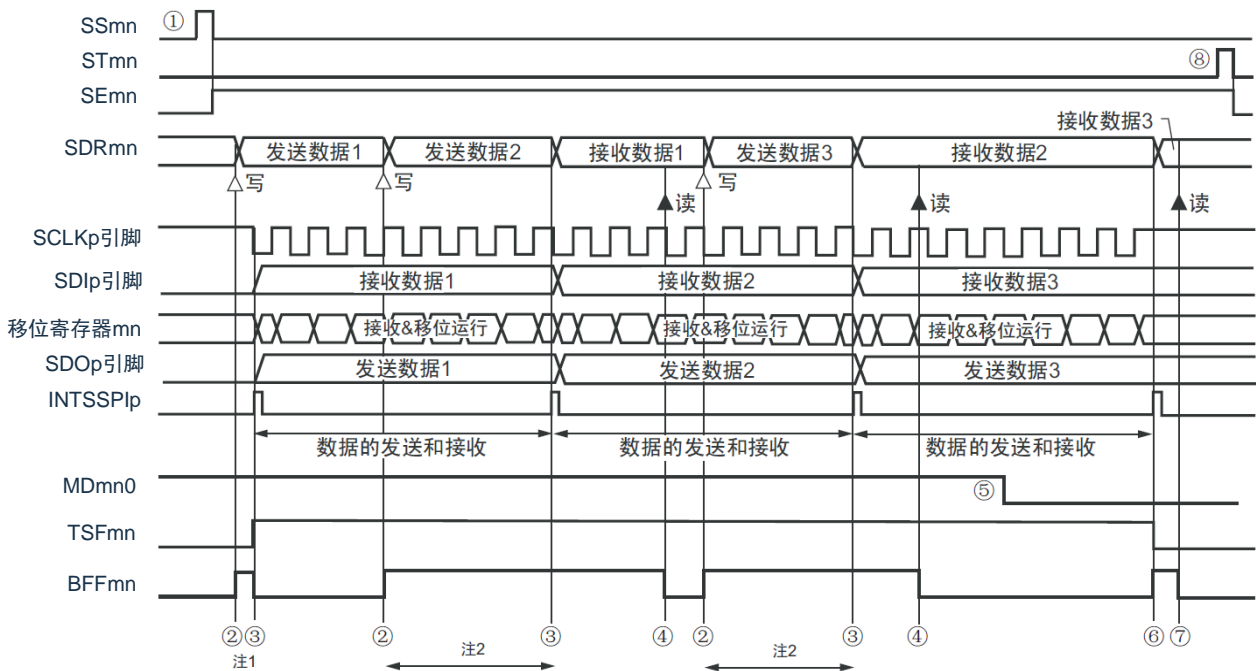
备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-44: 主控发送和接收(单次发送和接收模式)的流程图



(4) 处理流程(连续发送和接收模式)

图19-45: 主控发送和接收(连续发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



注1: 如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据, 就重写发送数据。

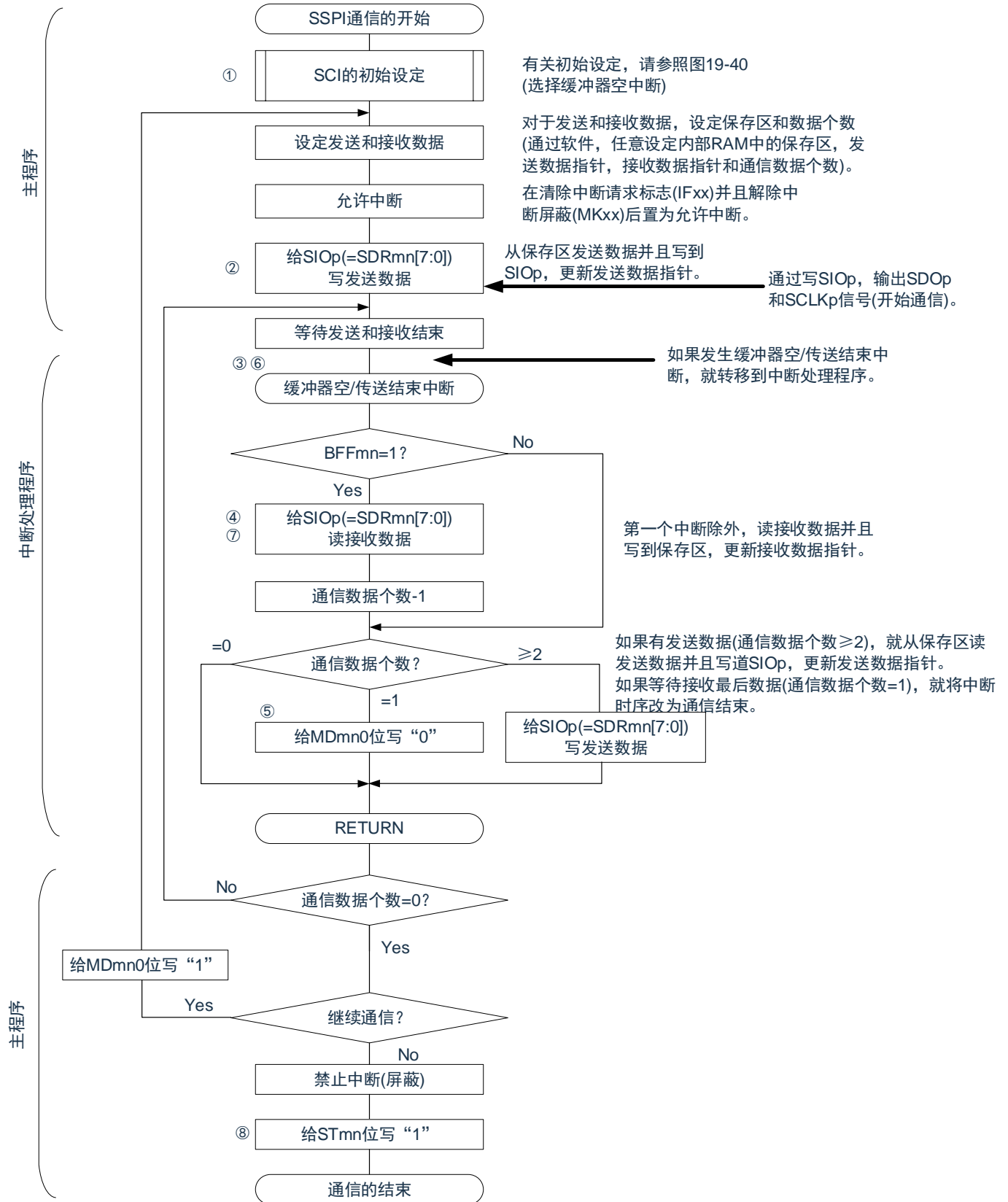
注2: 如果在此期间读取SDRmn寄存器, 就能读发送数据。此时, 不影响传送运行。

注意: 即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是, 为了能赶上最后发送数据的传送结束中断, 必须在开始传送最后一位之前进行改写。

备注:

1. 图中的①~⑧对应“图19-46: 主控发送和接收(连续发送和接收模式)的流程图”中的①~⑧。
2. m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-46: 主控发送和接收(连续发送和接收模式)的流程图



备注: 图中的①~⑧对应“图19-45: 主控发送和接收(连续发送和接收模式)的时序图”中的①~⑧。

19.5.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下BAT32A237微控制器将数据发送到其他设备的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDO00	SCLK01 SDO01	SCLK10 SDO10	SCLK11 SDO11	SCLK20 SDO20	SCLK21 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。					
错误检测标志	只有溢出错误检测标志(OVFmn)。					
传送数据长度	7位或者8位					
传送速率	Max.F _{MCK} /6[Hz] ^{注1,2}					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 					
数据方向	MSB优先或者LSB优先					

注1: 因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为F_{MCK}/6[Hz]。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

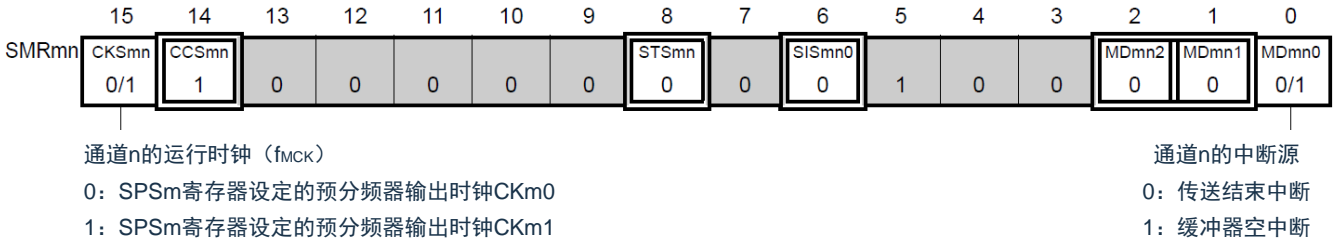
1. F_{MCK}: 对象通道的运行时钟频率
2. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

(1) 寄存器的设置

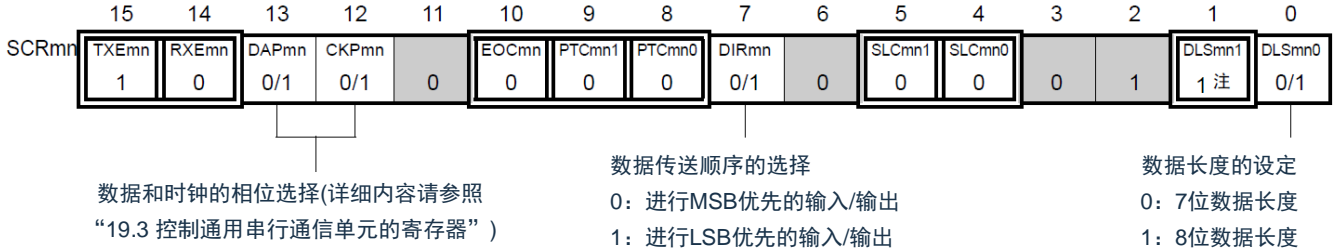
图19-47: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

从属发送时的寄存器设置内容例子

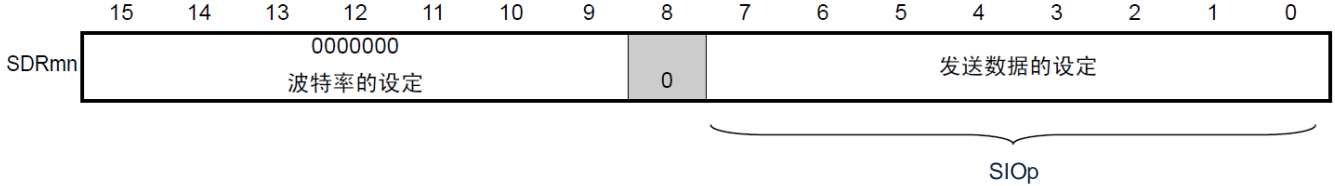
(a) 串行模式寄存器mn(SMRmn)



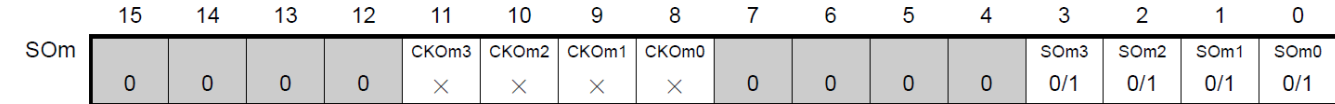
(b) 串行通信运行设定寄存器mn(SCRmn)



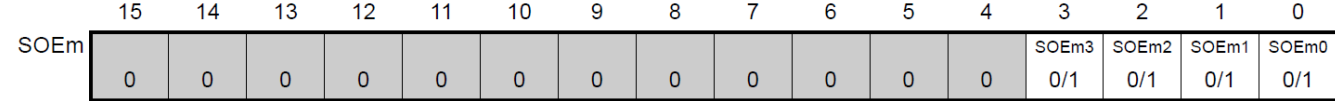
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



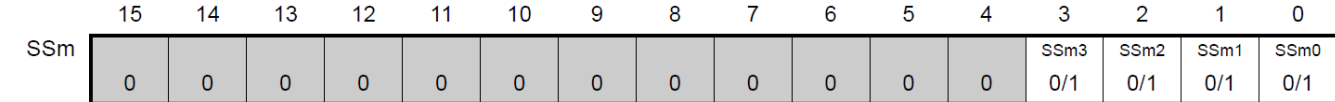
(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ...只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。



注: 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11
- : 在SSPI从属发送模式中为固定设置。■: 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-48: 从属发送的初始设置步骤



图19-49: 从属发送的中止步骤

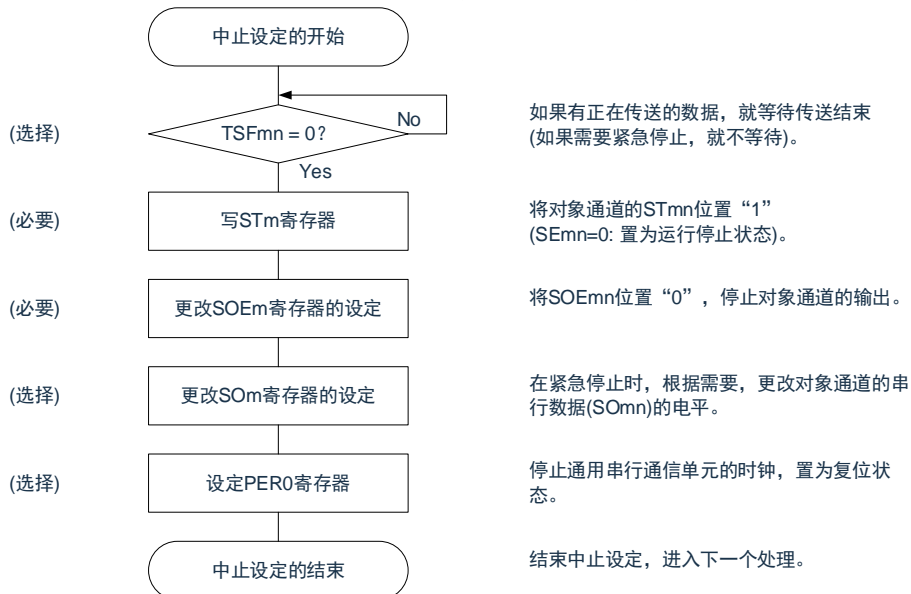
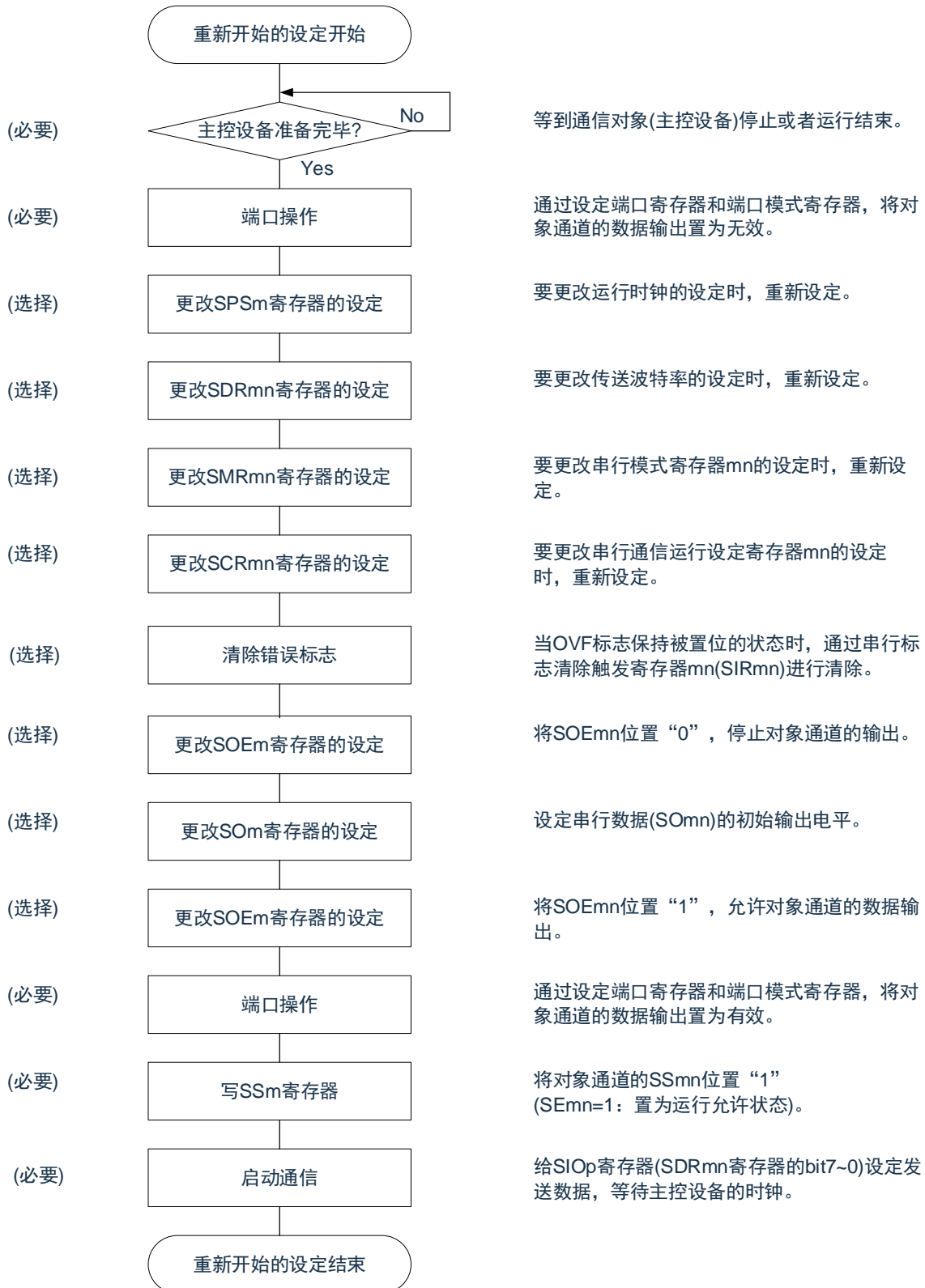


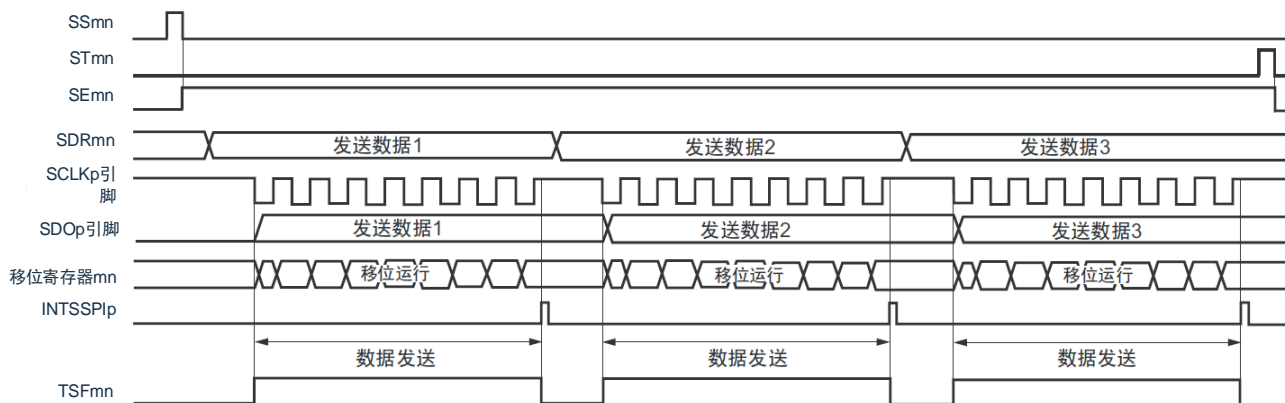
图19-50: 重新开始从属发送的设置步骤



备注: 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始的设置。

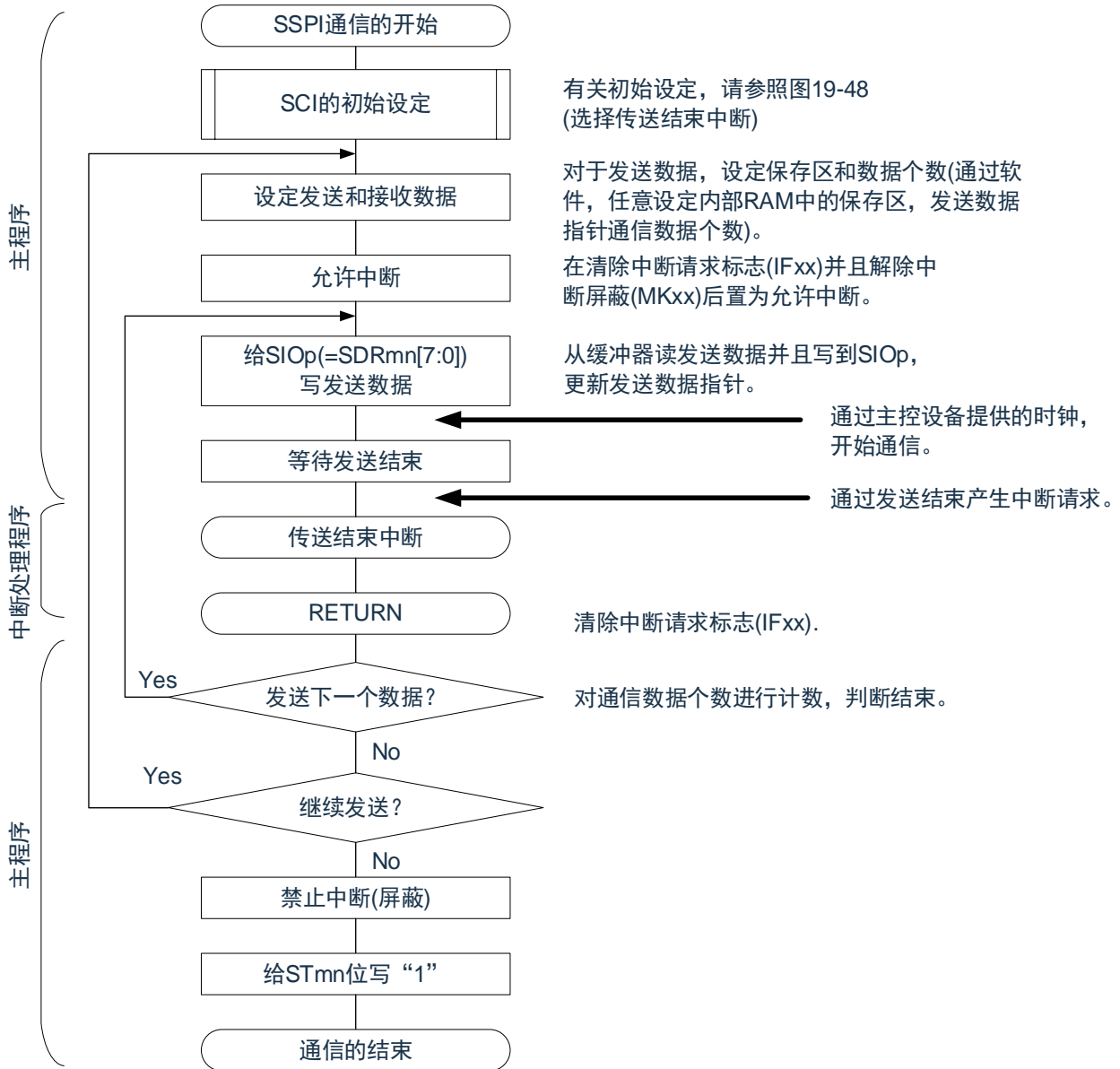
(3) 处理流程(单次发送模式)

图19-51: 从属发送(单次发送模式)的时序图(类型1: DAPmn=0、CKPmn=0)



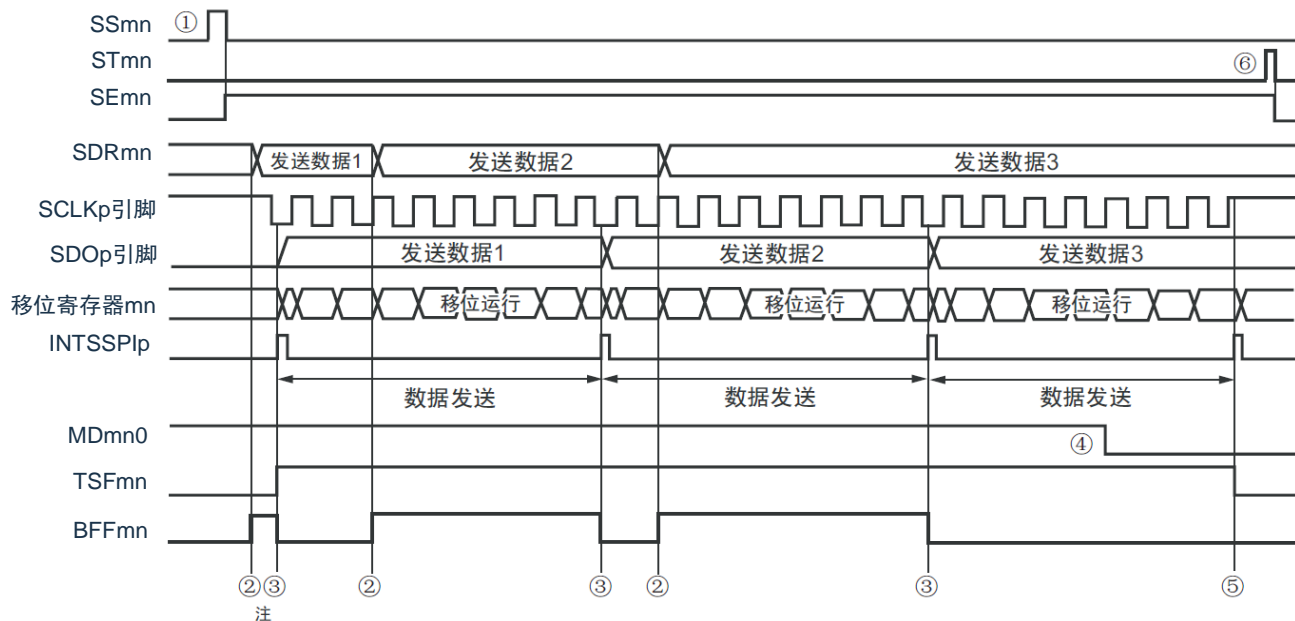
备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-52: 从属发送(单次发送模式)的流程图



(4) 处理流程(连续发送模式)

图19-53: 从属发送(连续发送模式)的时序图(类型1: DAPmn=0、CKPmn=0)

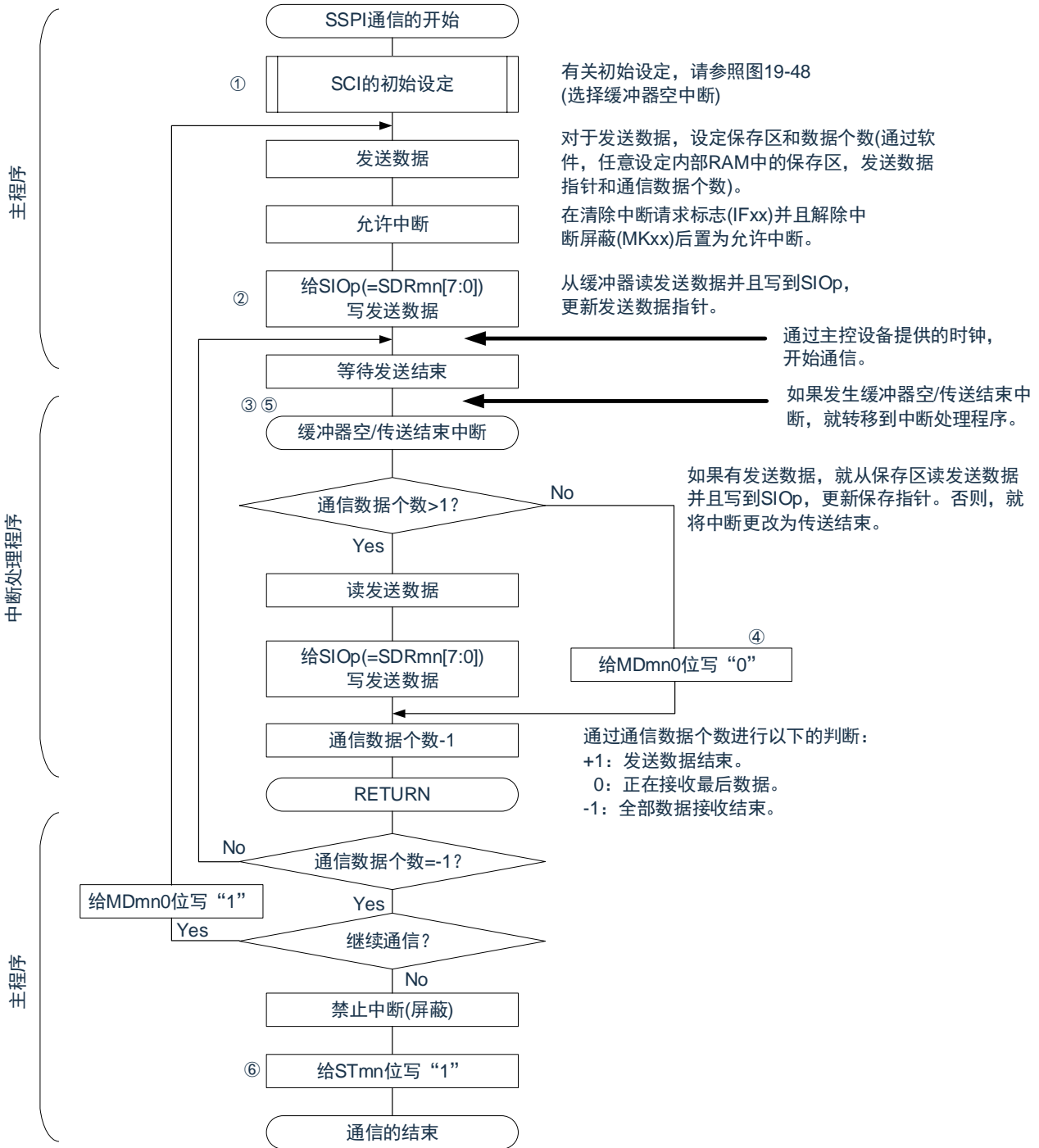


注: 如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据, 就重写发送数据。

注意: 即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是, 必须在开始传送最后一位之前进行改写。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-54：从属发送(连续发送模式)的流程图



备注：图中的①~⑥对应“图19-53：从属发送(连续发送模式)的时序图”中的①~⑥。

19.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下BAT32A237从其他设备接收数据的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDI00	SCLK01 SDI01	SCLK10 SDI10	SCLK11 SDI11	SCLK20 SDI20	SCLK21 SDI21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	只限于传送结束中断(禁止设置缓冲器空中断)。					
错误检测标志	只有溢出错误检测标志(OVFmn)。					
传送数据长度	7位或者8位					
传送速率	Max.fMCK/6[Hz] ^{注1,2}					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

注1: 因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为 $F_{MCK}/6$ [Hz]。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

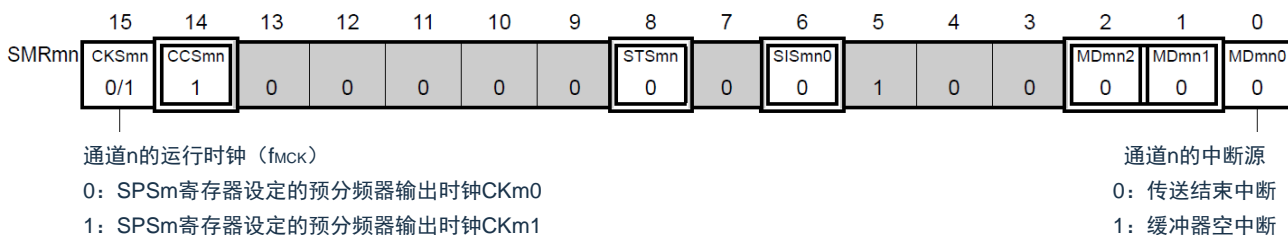
1. F_{MCK} : 对象通道的运行时钟频率
2. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

(1) 寄存器的设置

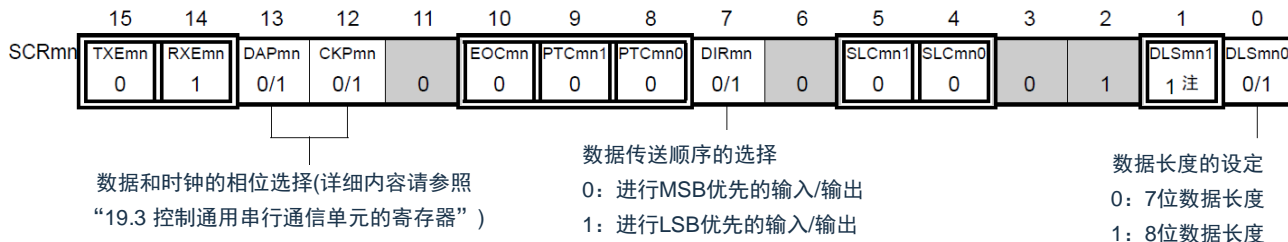
图19-55: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

从属接收时的寄存器设置内容例子

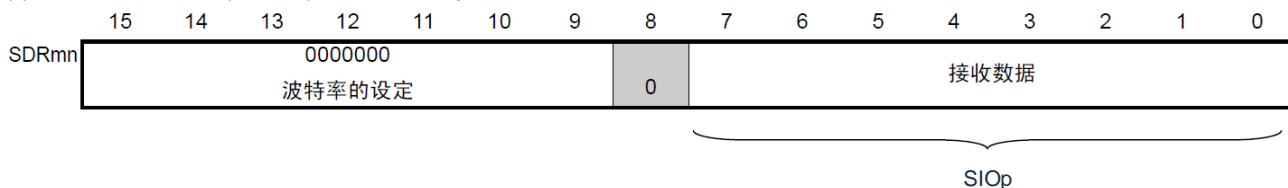
(a) 串行模式寄存器mn(SMRmn)



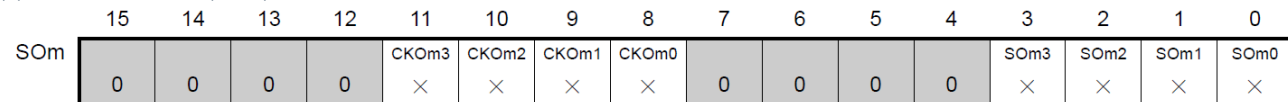
(b) 串行通信运行设定寄存器mn(SCRmn)



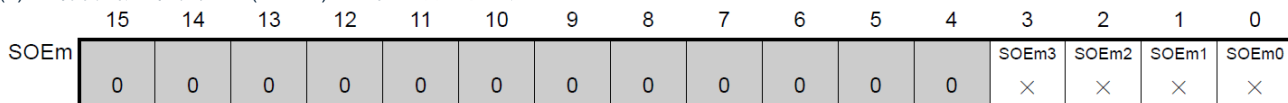
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



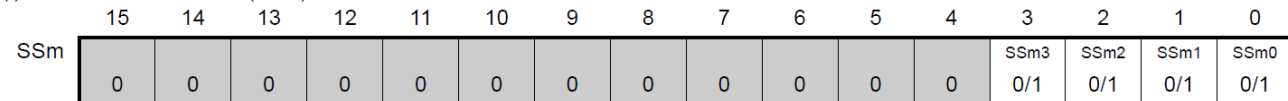
(d) 串行输出寄存器m(SOm) ...在此模式中不使用。



(e) 串行输出允许寄存器m(SOEm) ...在此模式中不使用。



(f) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。



注: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21、30、31)
mn=00~03、10~11
- : 在从属接收模式中为固定设置。■: 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-56: 从属接收的初始设置步骤

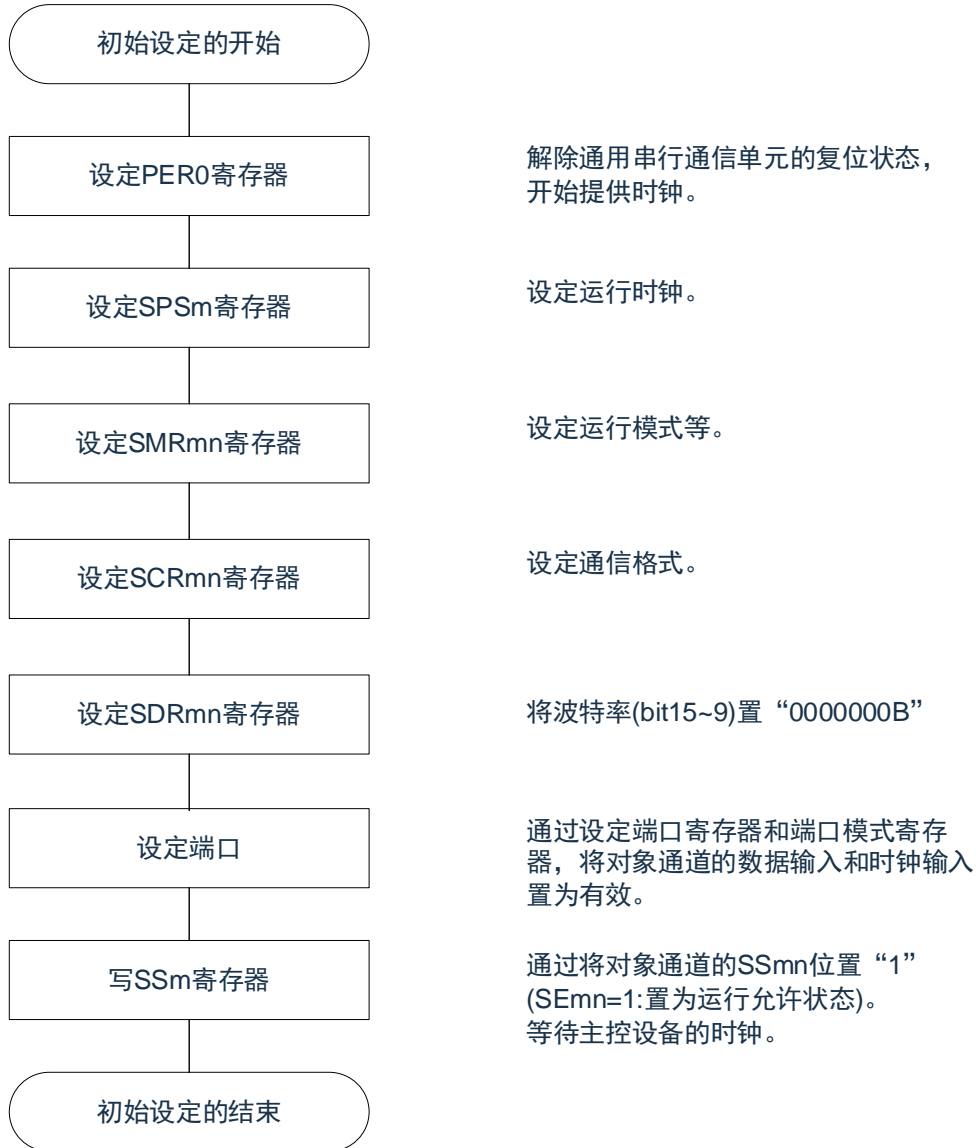


图19-57: 从属接收的中止步骤

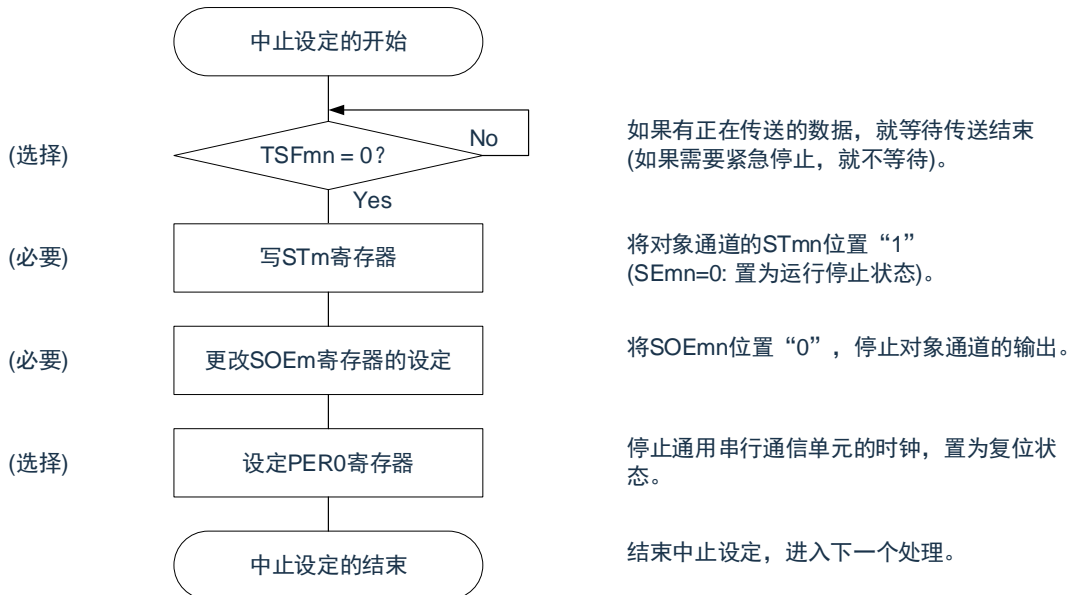


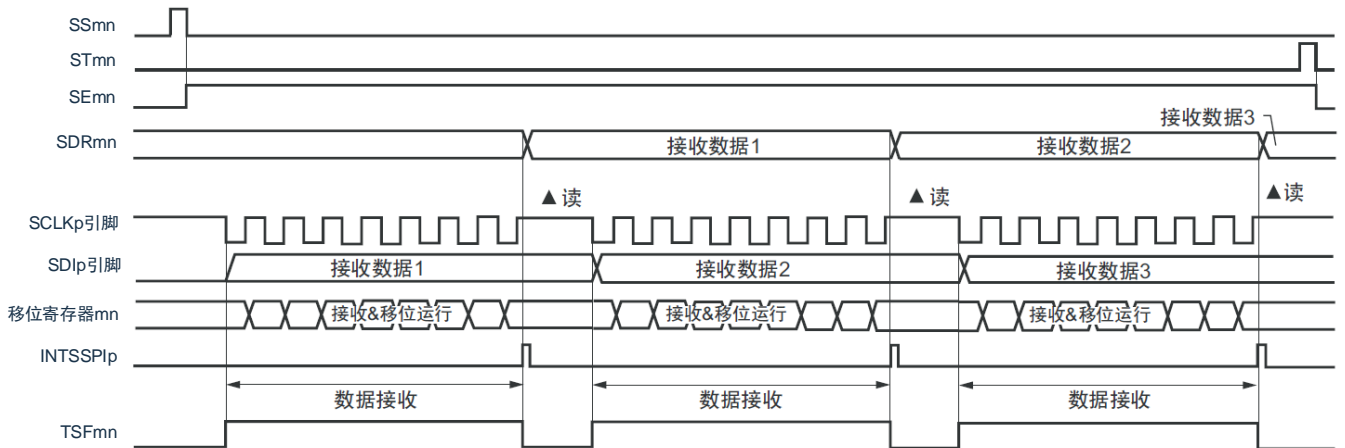
图19-58: 重新开始从属接收的设置步骤



备注: 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始的设置。

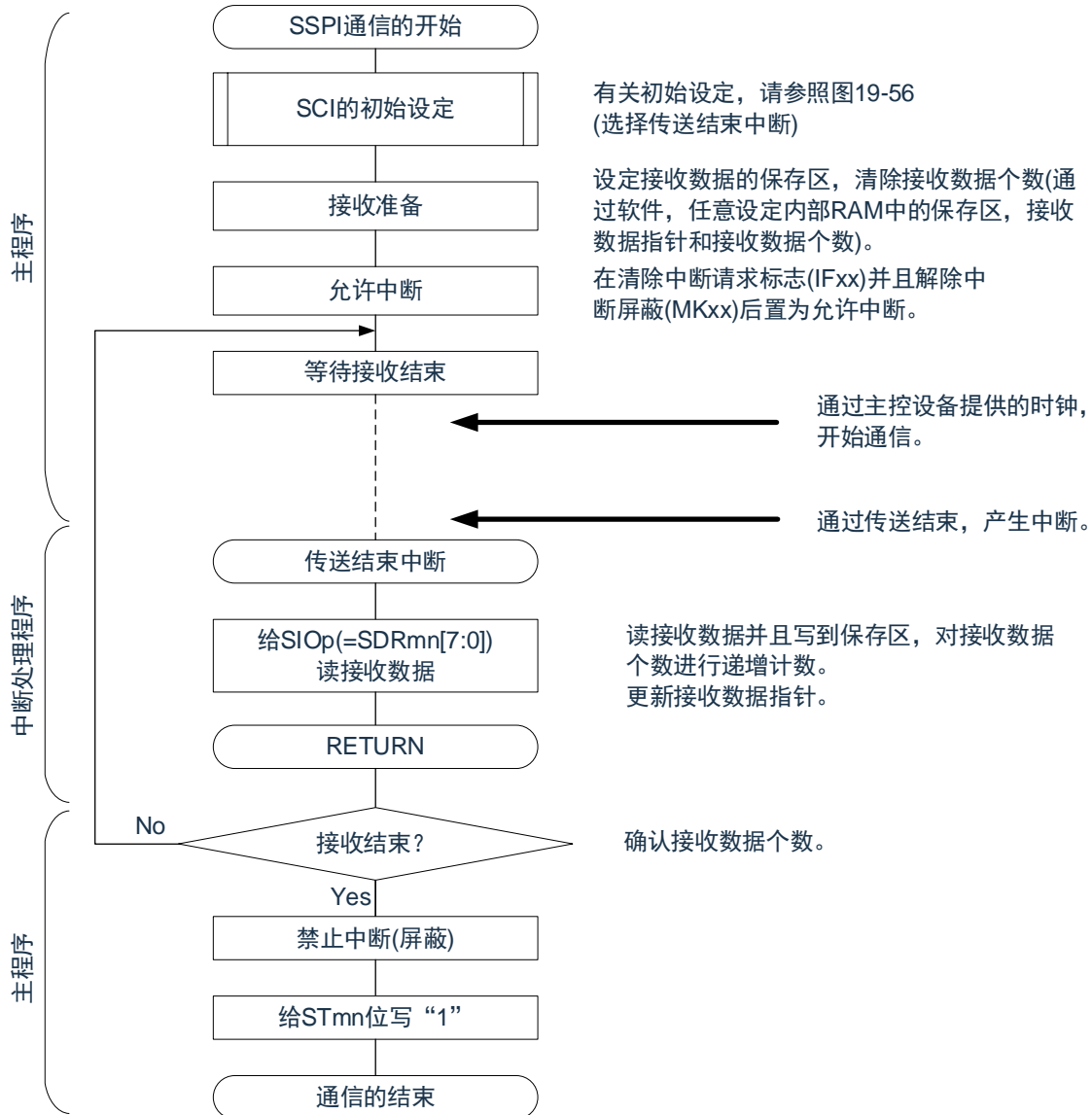
(3) 处理流程(单次接收模式)

图19-59: 从属接收(单次接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21、30、31)
mn=00~03、10~11

图19-60: 从属接收(单次接收模式)的流程图



19.5.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下BAT32A237微控制器和其他设备进行数据发送和接收的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00 SDI00 SDO00	SCLK01 SDI01 SDO01	SCLK10 SDI10 SDO10	SCLK11 SDI11 SDO11	SCLK20 SDI20 SDO20	SCLK21 SDI21 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。					
错误检测标志	只有溢出错误检测标志(OVFmn)。					
传送数据长度	7位或者8位					
传送速率	$\text{Max. } F_{\text{MCK}}/6[\text{Hz}]^{\text{注1,2}}$					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时, 开始数据输入/输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输入/输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

注1: 因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率 $F_{\text{MCK}}/6[\text{Hz}]$ 。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

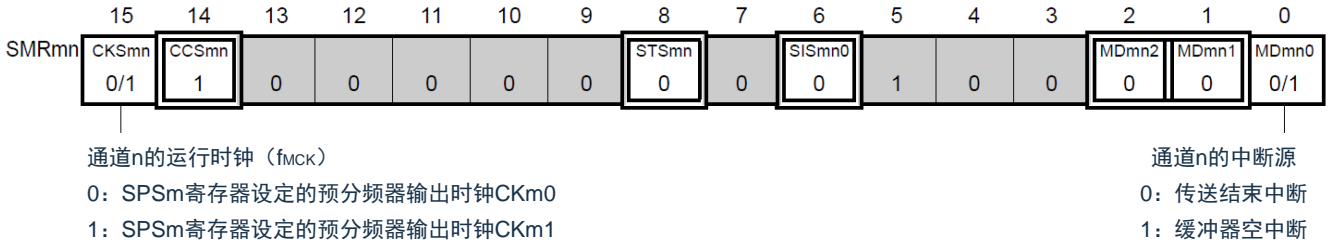
1. F_{MCK} : 对象通道的运行时钟频率
2. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

(1) 寄存器的设置

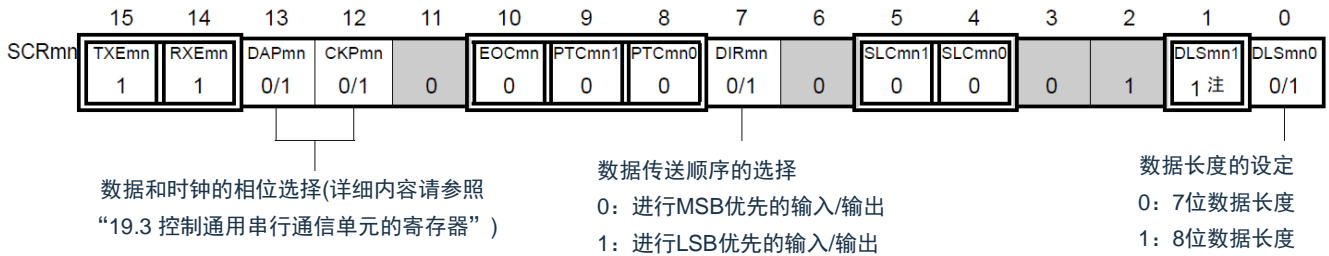
图19-61: 3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

从属发送和接收时的寄存器设置内容例子

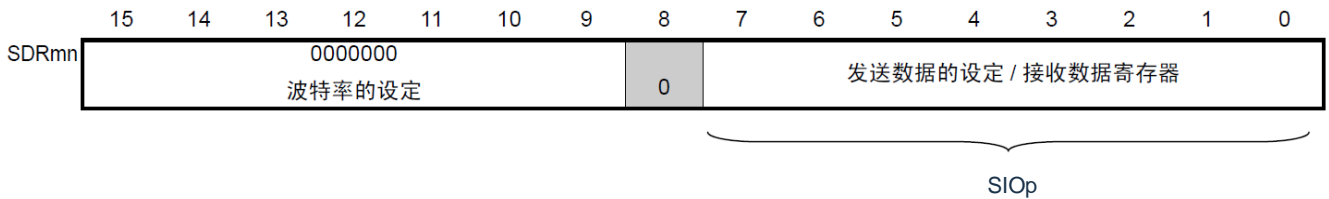
(a) 串行模式寄存器mn(SMRmn)



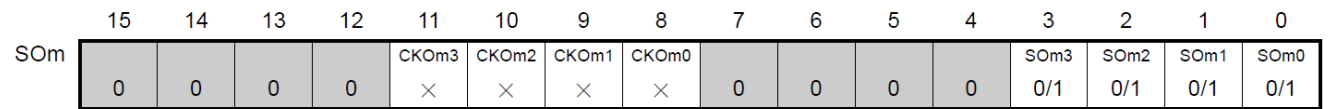
(b) 串行通信运行设定寄存器mn(SCRmn)



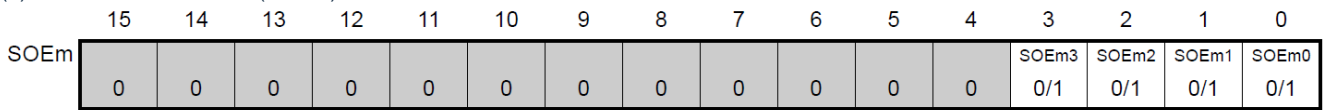
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



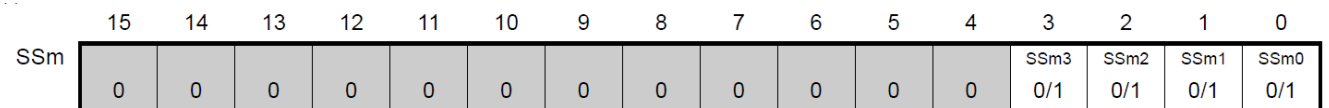
(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ...只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。



注: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

注意: 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11
- : 在SSPI从属发送和接收模式中为固定设置。■: 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-62：从属发送和接收的初始设置步骤



注意：在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

图19-63: 从属发送和接收的中止步骤

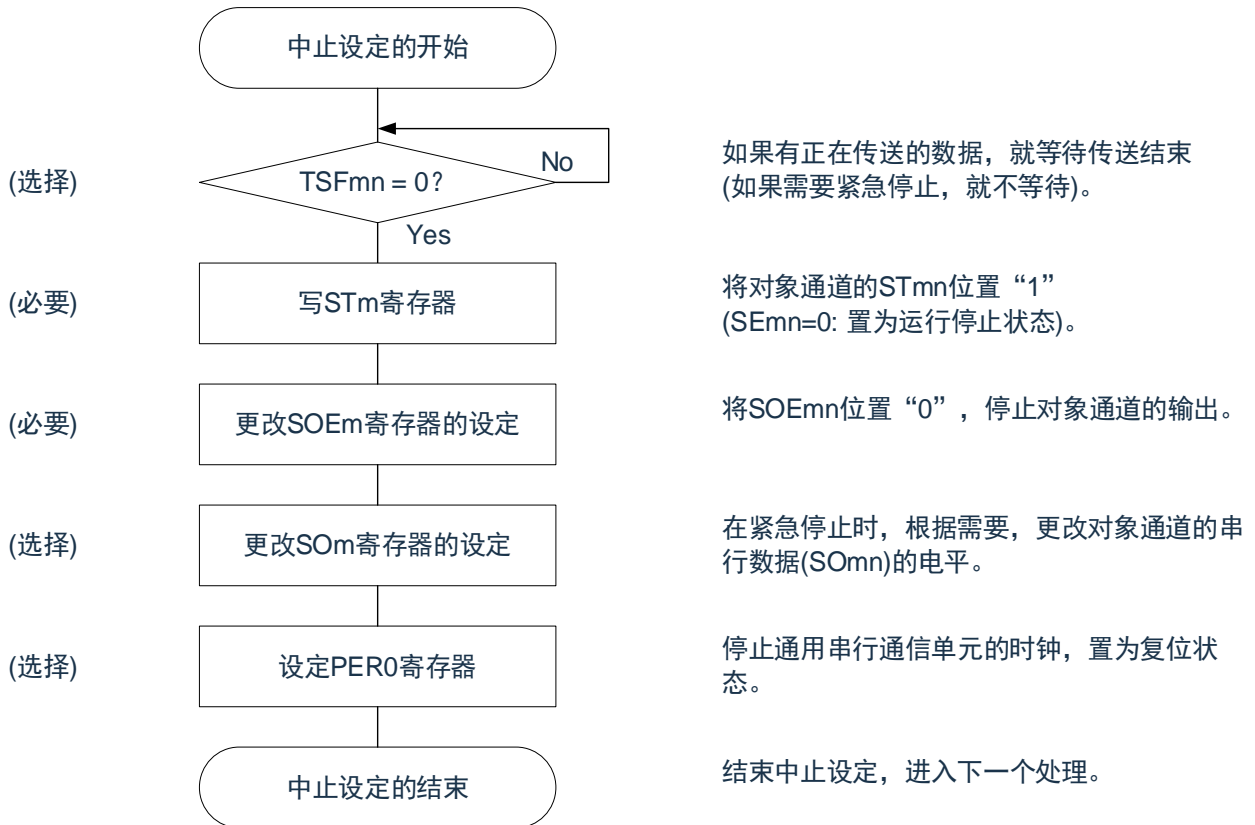
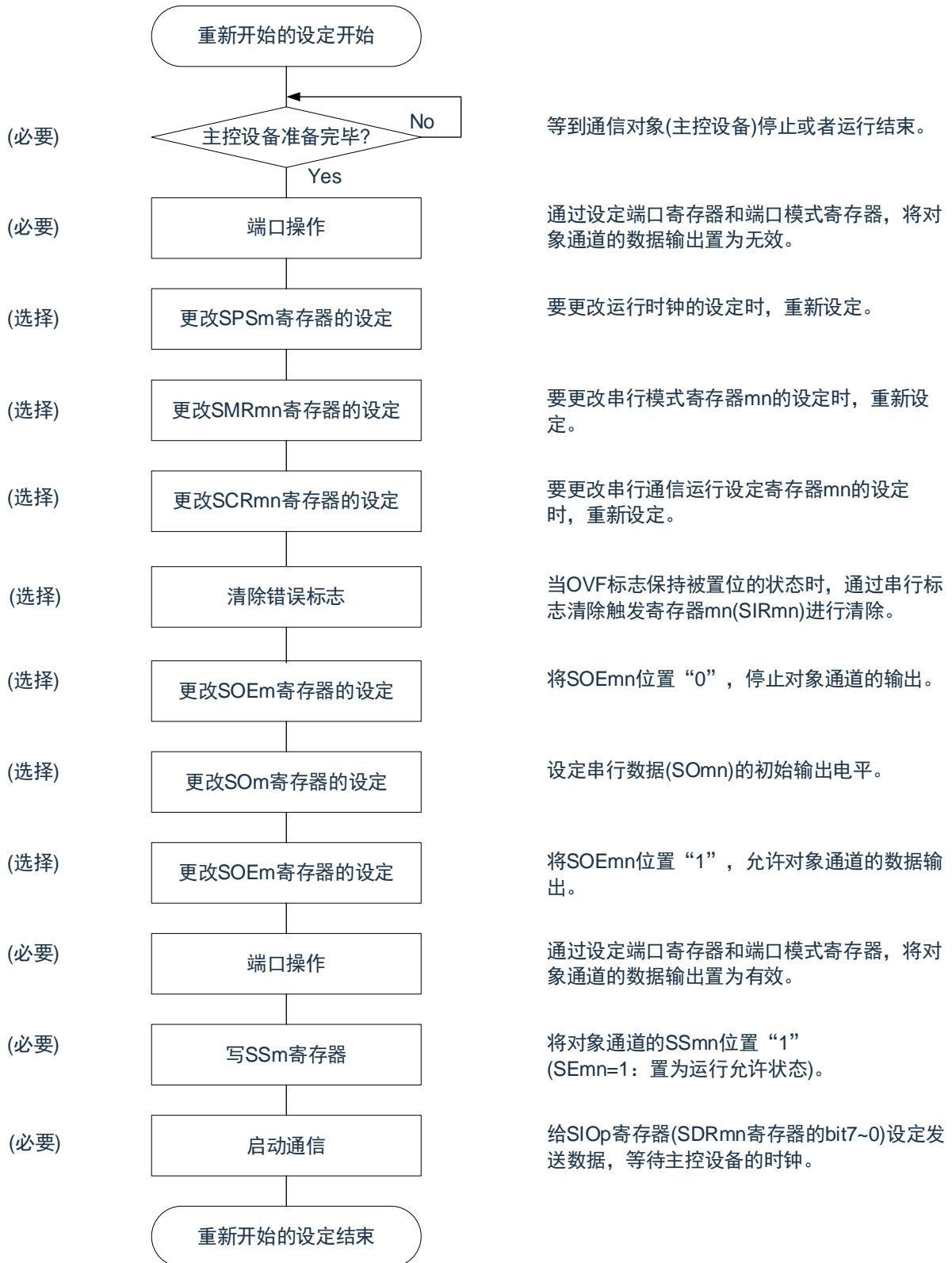


图19-64：重新开始从属发送和接收的设置步骤

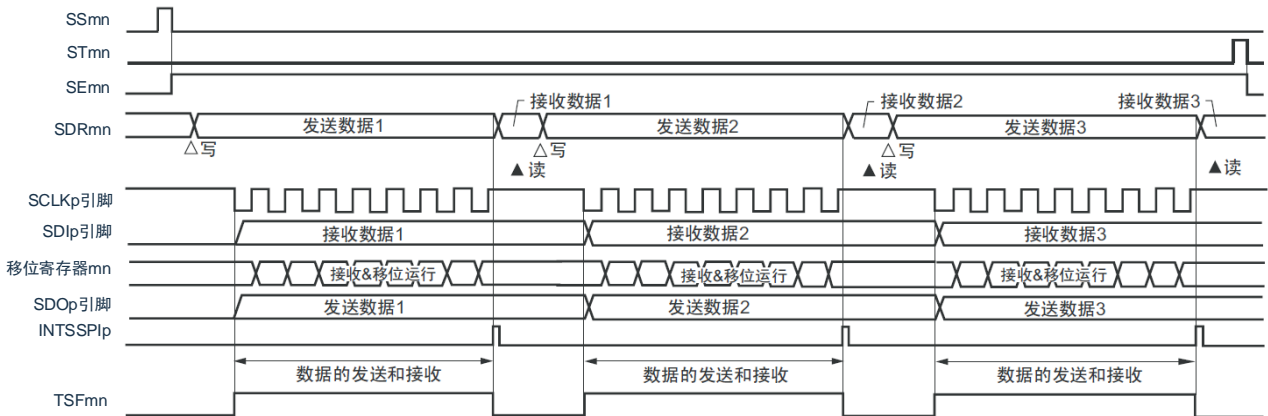


注意：

1. 在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。
2. 如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始的设置。

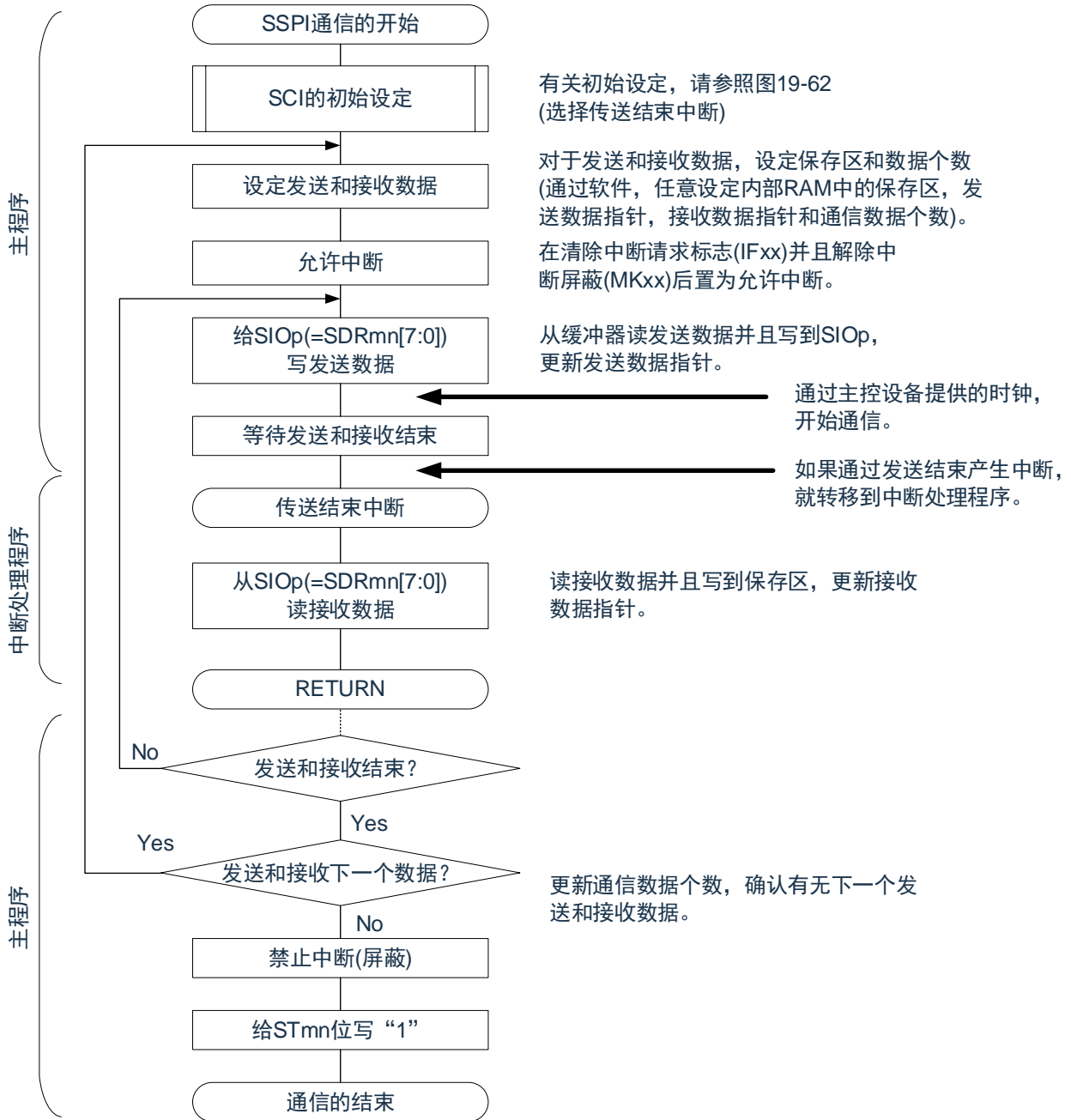
(3) 处理流程(单次发送和接收模式)

图19-65: 从属发送和接收(单次发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



备注: m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

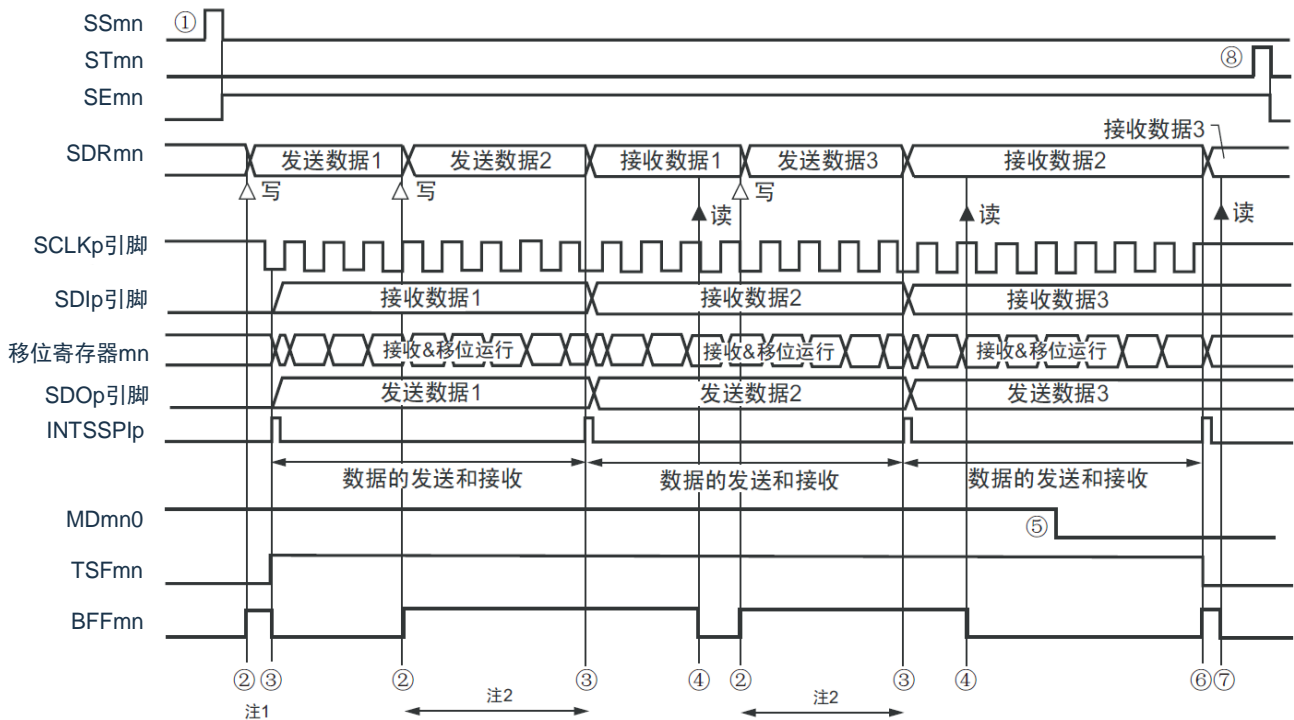
图19-66：从属发送和接收(单次发送和接收模式)的流程图



注意：在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

(4) 处理流程(连续发送和接收模式)

图19-67: 从属发送和接收(连续发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



注1: 如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据, 就重写发送数据。

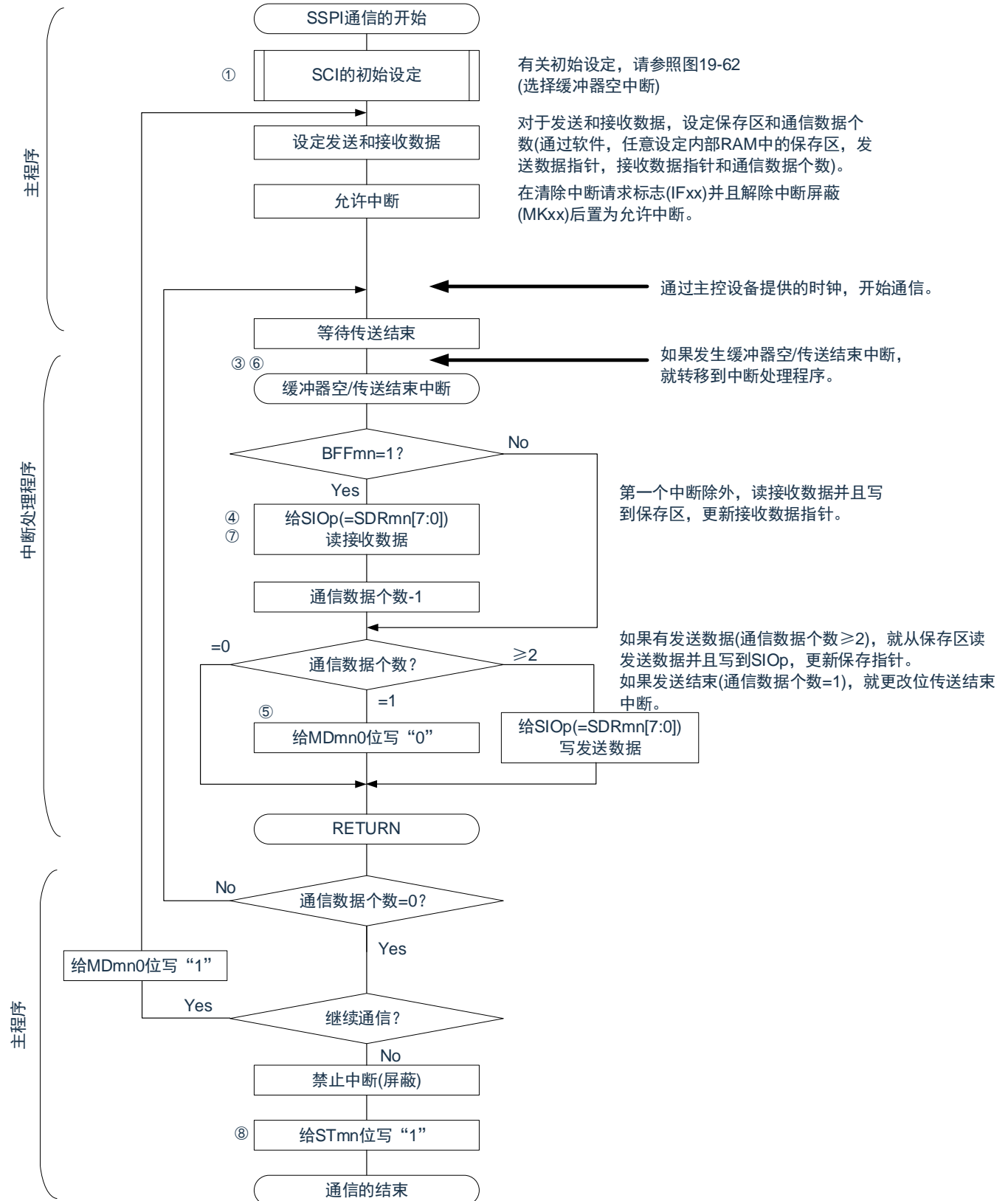
注2: 如果在此期间读取SDRmn寄存器, 就能读发送数据。此时, 不影响传送运行。

注意: 即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是, 为了能赶上最后发送数据的传送结束中断, 必须在开始传送最后一位之前进行改写。

备注:

1. 图中的①~⑧对应“图19-68: 从属发送和接收(连续发送和接收模式)的流程图”中的①~⑧。
2. m: 单元号(m=0、1)n: 通道号(n=0~3)p: SSPI号(p=00、01、10、11、20、21)
mn=00~03、10~11

图19-68: 从属发送和接收(连续发送和接收模式)的流程图



注意: 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注: 图中的①~⑧对应“图19-67: 从属发送和接收(连续发送和接收模式)的时序图”中的①~⑧。

19.5.7 传送时钟频率的计算

3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信的传送时钟频率能用以下计算式进行计算。

(1) 主控设备

$$\text{(传送时钟频率)} = \{\text{对象通道的运行时钟 (F}_{\text{MCK}}\text{) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) 从属设备

$$\text{(传送时钟频率)} = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\}^{\frac{1}{2}} [\text{Hz}]$$

注：容许的最大传送时钟频率为 $F_{\text{MCK}}/6$ 。

备注：因为SDRmn[15:9]的值为串行数据寄存器mn(SDRmn)的bit15~9的值(0000000B~1111111B)，所以为0~127。

运行时钟(F_{MCK})取决于串行时钟选择寄存器m(SPSm)和串行模式寄存器mn(SMRmn)的bit15(CKSmn)。

表19-2: 3线串行I/O运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟(F _{CLK}) ^注		
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	F _{CLK} =32MHz运行时	
0	X	X	X	X	0	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	1	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	1	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	1	F _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	1	0	F _{CLK} /2 ⁶	500KHz
	X	X	X	X	0	1	1	1	1	F _{CLK} /2 ⁷	250KHz
	X	X	X	X	1	0	0	0	0	F _{CLK} /2 ⁸	125KHz
	X	X	X	X	1	0	0	1	1	F _{CLK} /2 ⁹	62.5KHz
	X	X	X	X	1	0	1	0	0	F _{CLK} /2 ¹⁰	31.25KHz
	X	X	X	X	1	0	1	1	1	F _{CLK} /2 ¹¹	15.63KHz
	X	X	X	X	1	1	0	0	0	F _{CLK} /2 ¹²	7.81KHz
	X	X	X	X	1	1	0	1	1	F _{CLK} /2 ¹³	3.91KHz
	X	X	X	X	1	1	1	0	0	F _{CLK} /2 ¹⁴	1.95KHz
X	X	X	X	1	1	1	1	1	F _{CLK} /2 ¹⁵	977Hz	
1	0	0	0	0	X	X	X	X	X	F _{CLK}	32MHz
	0	0	0	1	X	X	X	X	X	F _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	X	F _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	X	F _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	X	F _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	X	F _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	X	F _{CLK} /2 ⁶	500KHz
	0	1	1	1	X	X	X	X	X	F _{CLK} /2 ⁷	250KHz
	1	0	0	0	X	X	X	X	X	F _{CLK} /2 ⁸	125KHz
	1	0	0	1	X	X	X	X	X	F _{CLK} /2 ⁹	62.5KHz
	1	0	1	0	X	X	X	X	X	F _{CLK} /2 ¹⁰	31.25KHz
	1	0	1	1	X	X	X	X	X	F _{CLK} /2 ¹¹	15.63KHz
	1	1	0	0	X	X	X	X	X	F _{CLK} /2 ¹²	7.81KHz
	1	1	0	1	X	X	X	X	X	F _{CLK} /2 ¹³	3.91KHz
	1	1	1	0	X	X	X	X	X	F _{CLK} /2 ¹⁴	1.95KHz
1	1	1	1	X	X	X	X	X	F _{CLK} /2 ¹⁵	977Hz	

注: 要更改被选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)时, 必须在停止通用串行通信单元(SCI)的运行(串行通道停止寄存器m(STm)=000FH)后进行更改。

备注:

1. X: 忽略
2. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

19.5.8 在3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信过程中发生错误时的处理步骤

在3线串行I/O(SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)通信过程中发生错误时的处理步骤如图19-69所示。

图19-69：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn (SDIRmn) 写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号(m=0、1)n：通道号(n=0~3)mn=00~03、10~11

19.6 从属选择输入功能的时钟同步串行通信的运行

SCI0的通道0是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择
- 发送和接收数据的电平设置

[时钟控制]

- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率^注

从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注：必须在满足SCLK周期时间(T_{KCY})特性的范围内使用。详细内容请参照数据手册。

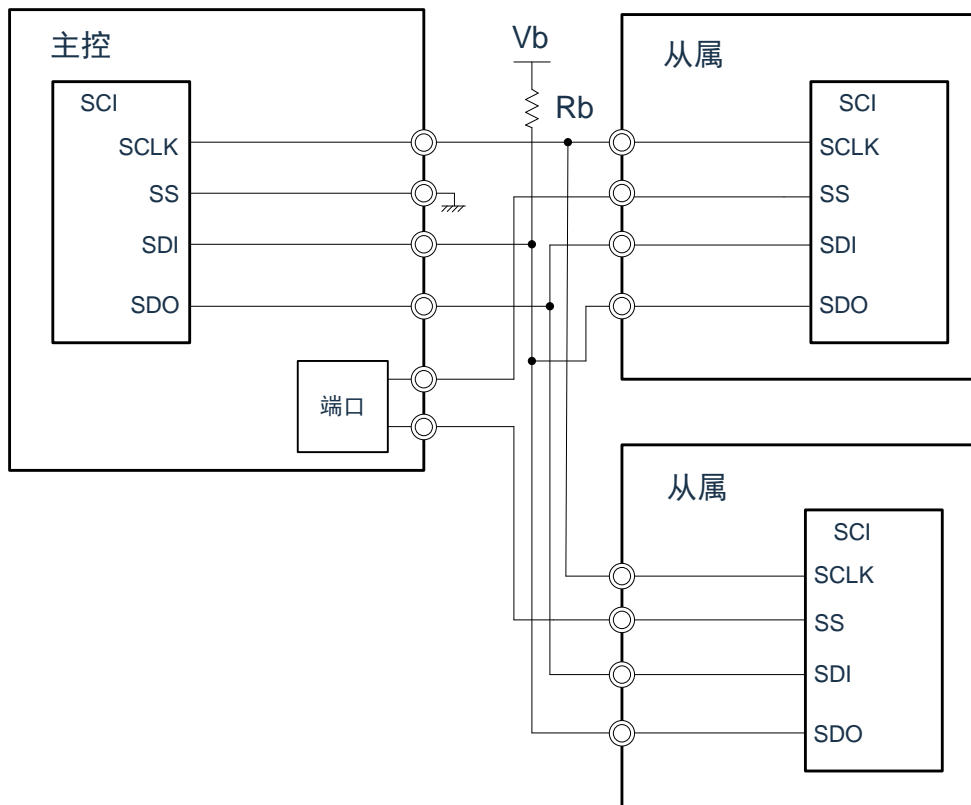
从属选择输入功能有以下3种通信运行：

- 从属发送 (参照19.6.1)
- 从属接收 (参照19.6.2)
- 从属的发送和接收 (参照19.6.3)

能通过使用从属选择输入功能，使1个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备(1个)进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制SDO引脚的输出。当被选择为通信对象的从属设备时，SDO引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO引脚变为高电平输出，因此在连接多个从属设备的环境下需要将SDO引脚设置为Nch-O.D并且将该节点上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

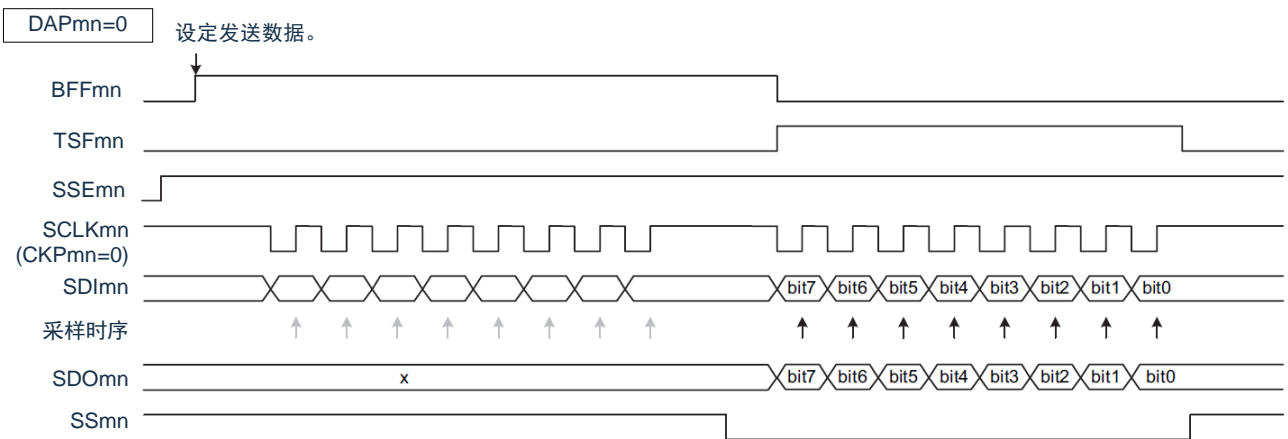
注意：必须通过端口的操作输出从属选择信号。

图19-70：从属选择输入功能的结构例子



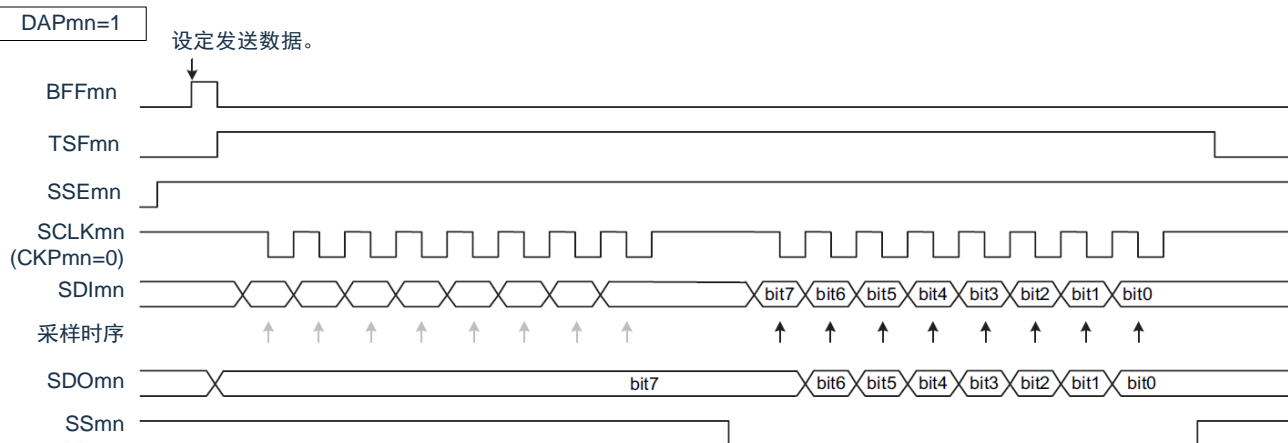
注意：将SDO00引脚选择为N沟道漏极开路输出模式。

图19-71：从属选择输入功能的时序图



在SSmn为高电平期间，即使在SCKmn(串行时钟)的下降沿也不进行发送，而且也不进行与上升沿同步的接收数据的采样。

在SSmn为低电平期间，与串行时钟的下降沿同步输出数据(移位)并且与上升沿同步接收数据。



当DAPmn位为“1”时，如果在SSmn为高电平期间设置发送数据，就将最初的数据(bit7)提供给数据输出。但是，即使在SCLKmn(串行时钟)的上升沿也不移位，而且也不进行与下降沿同步的接受数据的采样。如果SSmn变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注：m：单元号(m=0)n：通道号(n=0)

19.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下BAT32A237将数据发送到其他设备的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。
错误检测标志	只有溢出错误检测标志(OVFmn)。
传送数据长度	7位或者8位
传送速率	Max.F _{MCK} /6[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择功能的运行。

注1: 因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为F_{MCK}/6[Hz]。

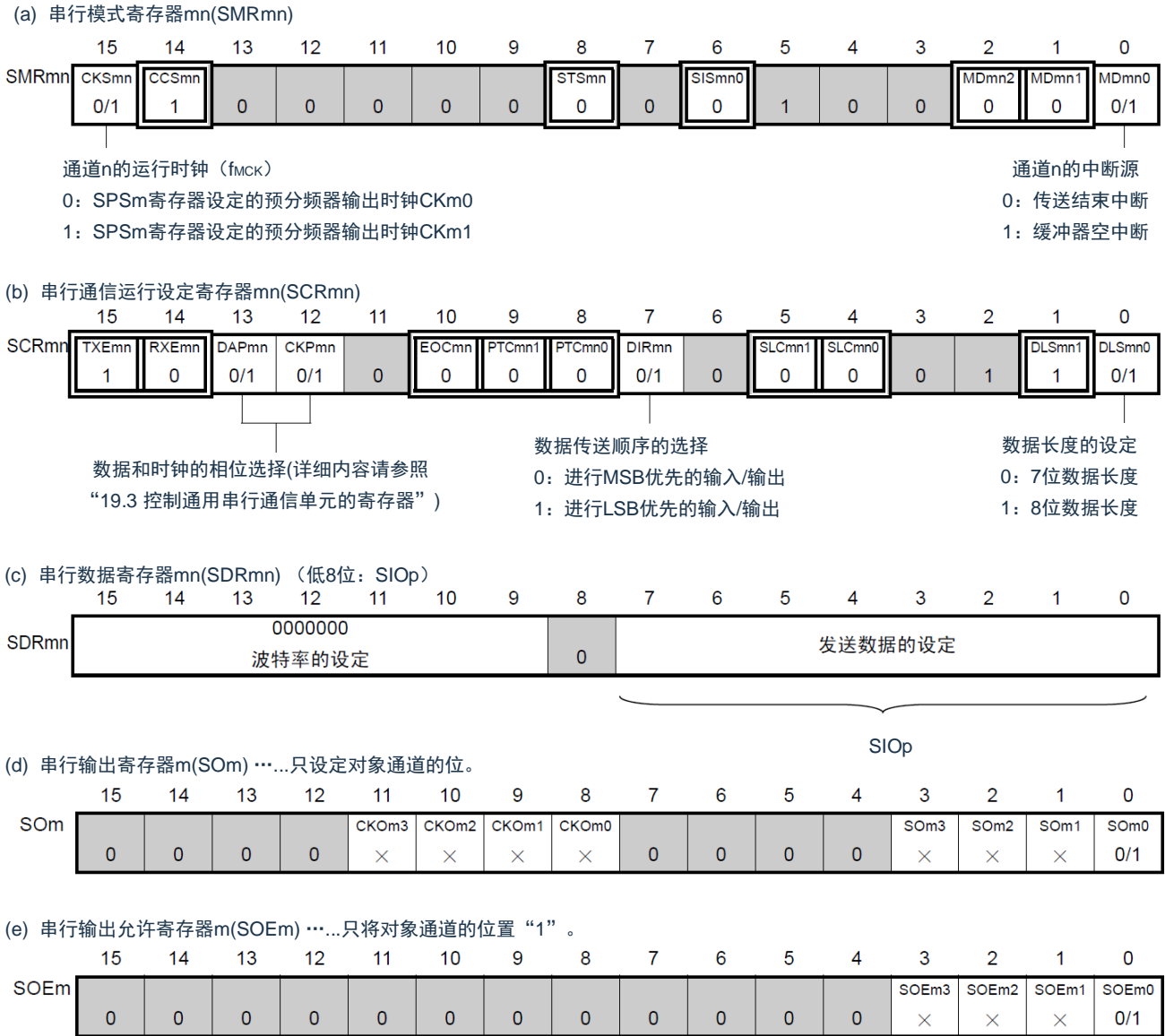
注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

1. F_{MCK}: 对象通道的运行时钟频率
2. m: 单元号(m=0)n: 通道号(n=0)

(1) 寄存器的设置

图19-72：从属选择输入功能(SSPI00)从属发送时的寄存器设置内容例子(1/2)



备注:

1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. : 在SSPI从属发送模式中为固定设置。 : 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图19-72：从属选择输入功能(SSPI00)从属发送时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

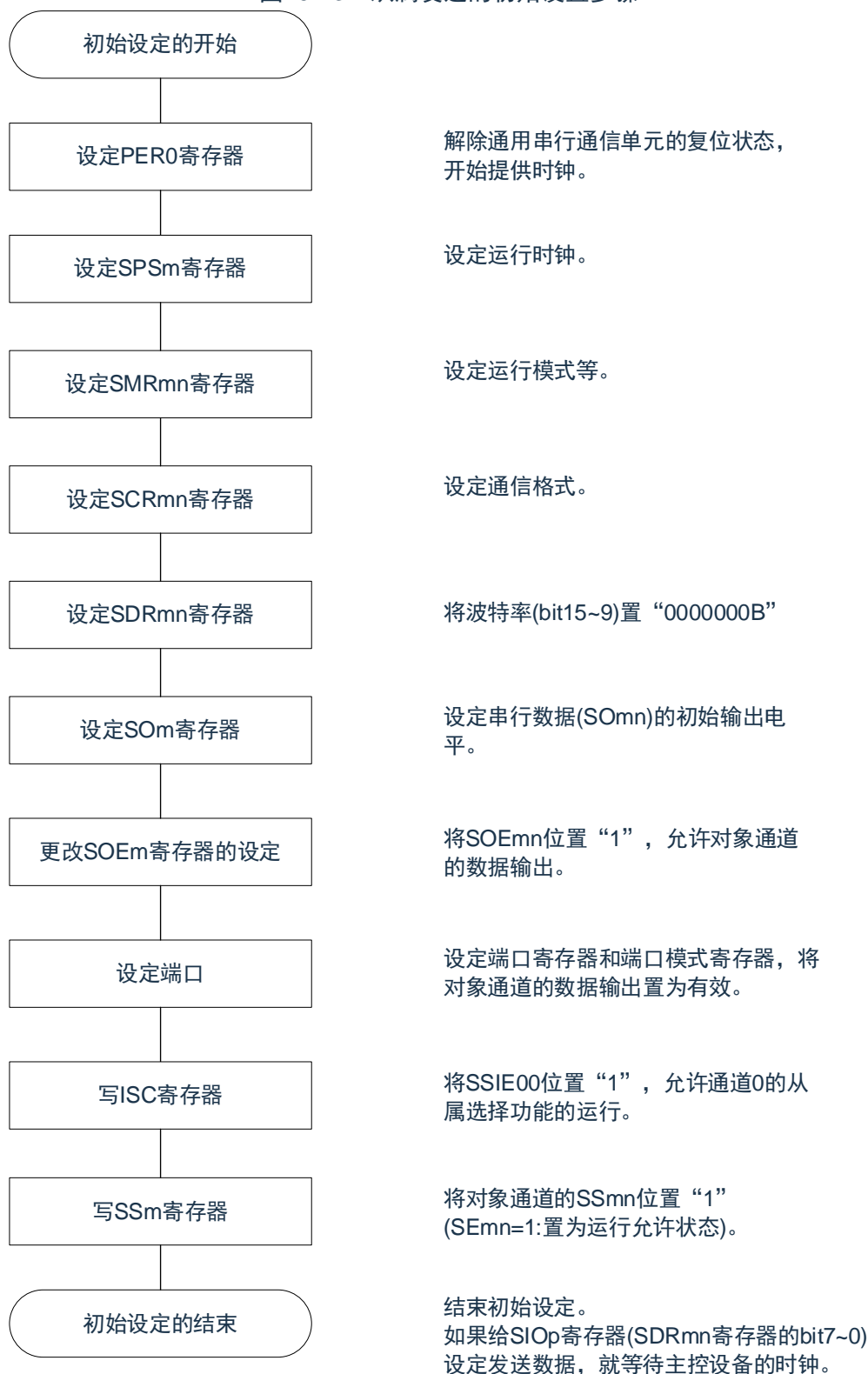
0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

备注:

1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. : 在SSPI从属发送模式中为固定设置。 : 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

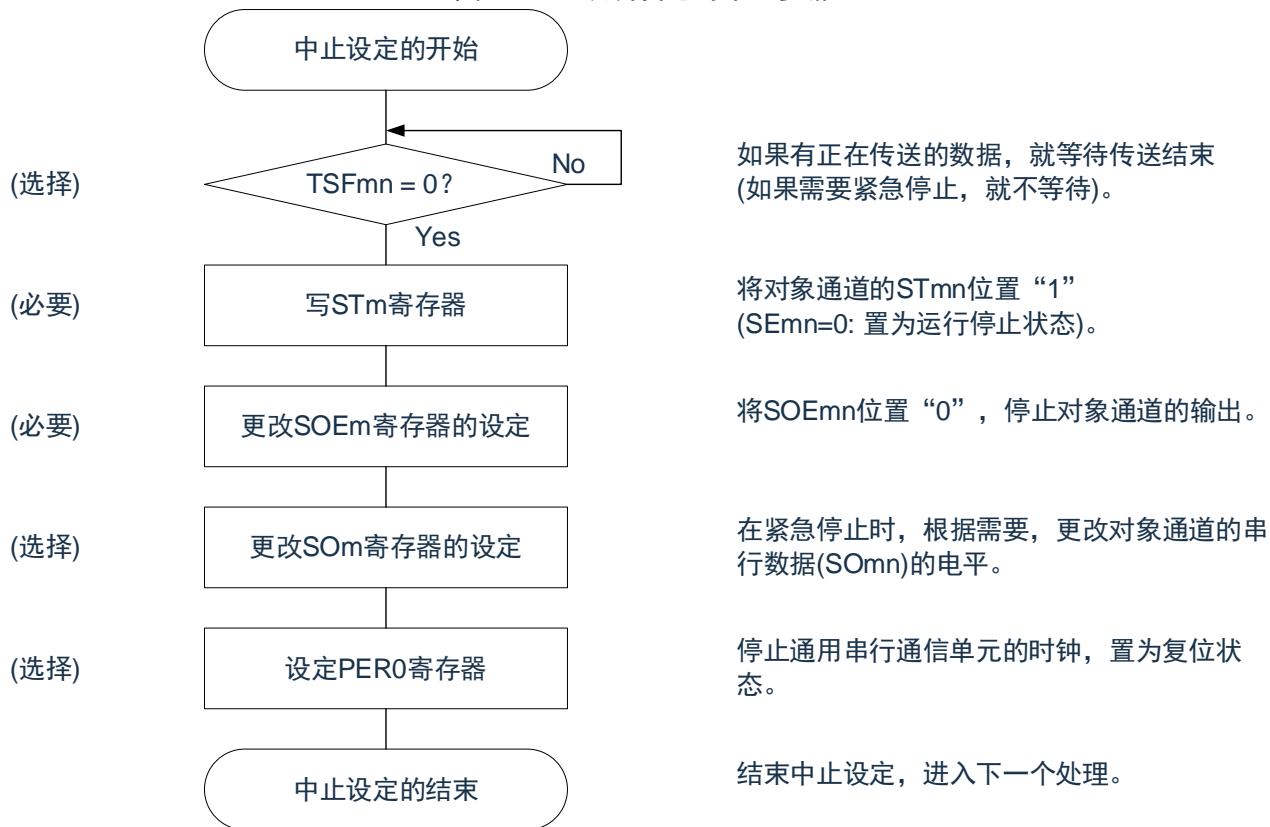
(2) 操作步骤

图19-73: 从属发送的初始设置步骤



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

图19-74：从属发送的中止步骤



备注：

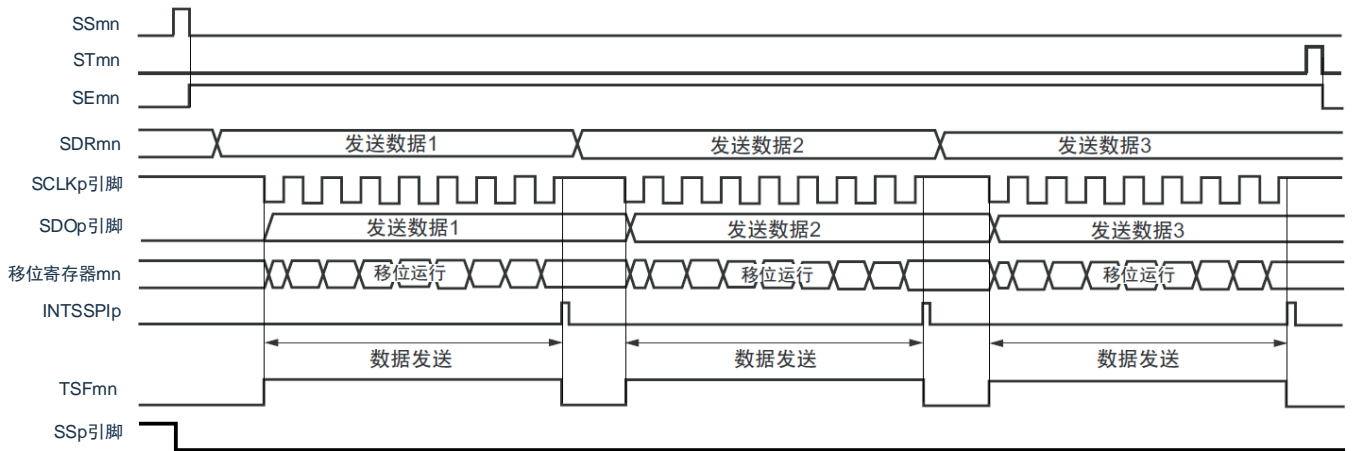
1. 如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始设置。
2. m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

图19-75: 重新开始从属发送的设置步骤



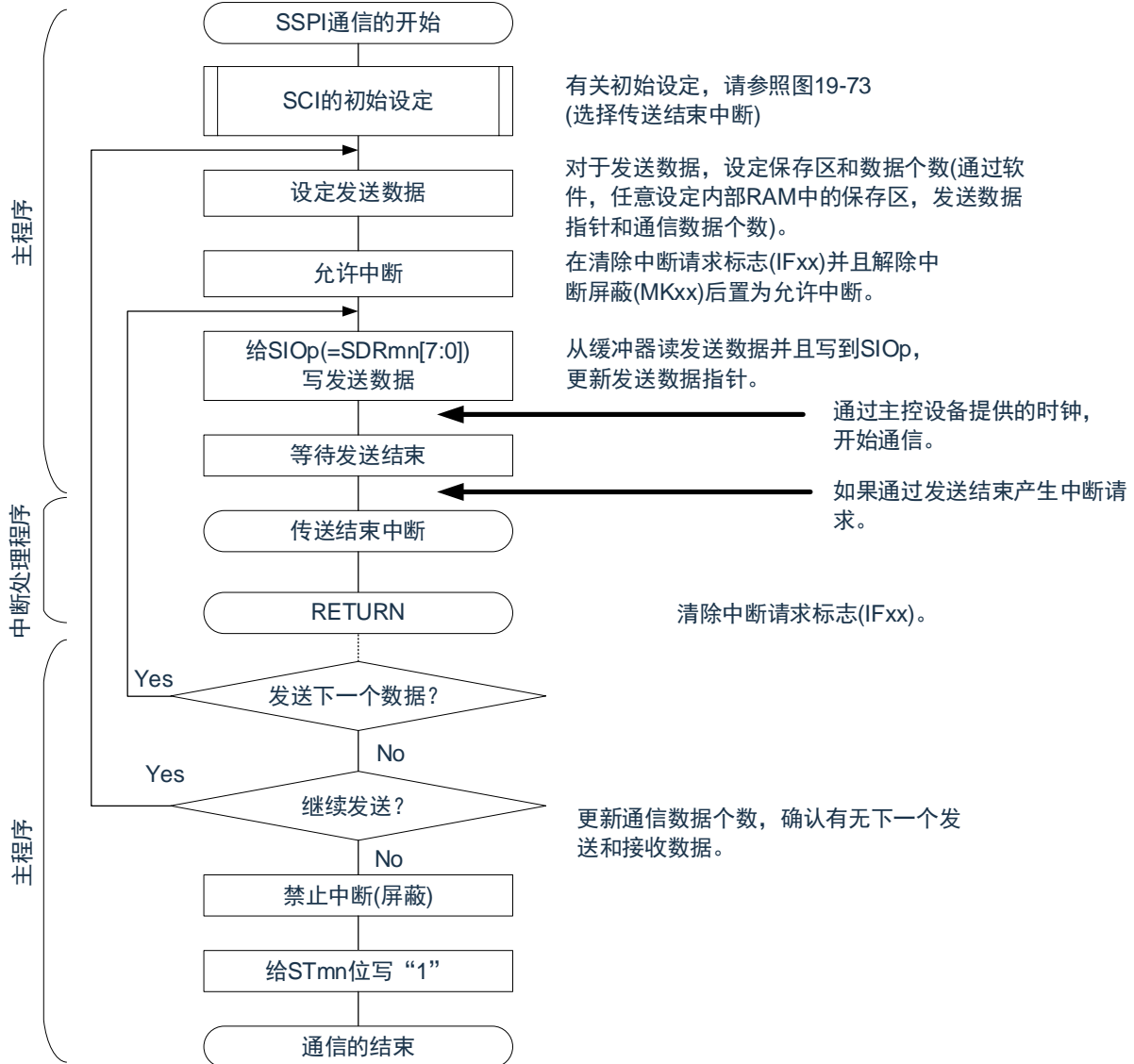
(3) 处理流程(单次发送模式)

图19-76: 从属发送(单次发送模式)的时序图(类型1: DAPmn=0、CKPmn=0)



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

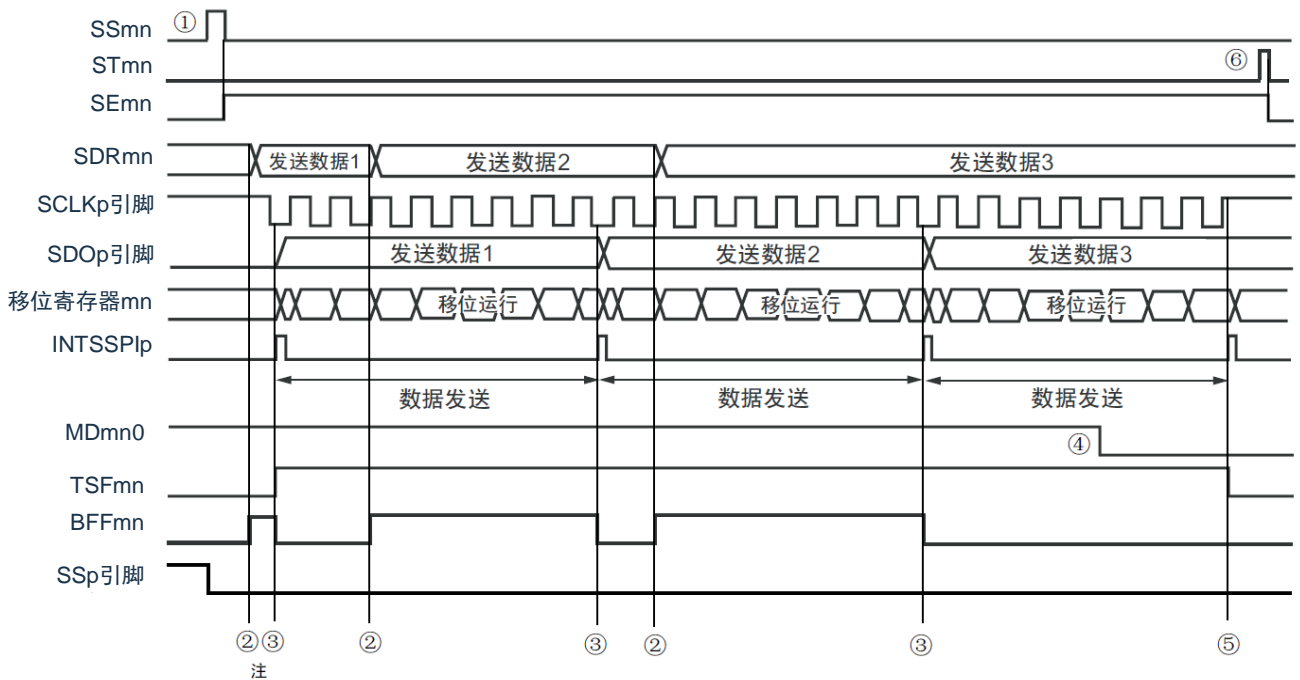
图19-77: 从属发送(单次发送模式)的流程图



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

(4) 处理流程(连续发送模式)

图19-78：从属发送(连续发送模式)的时序图(类型1：DAPmn=0、CKPmn=0)

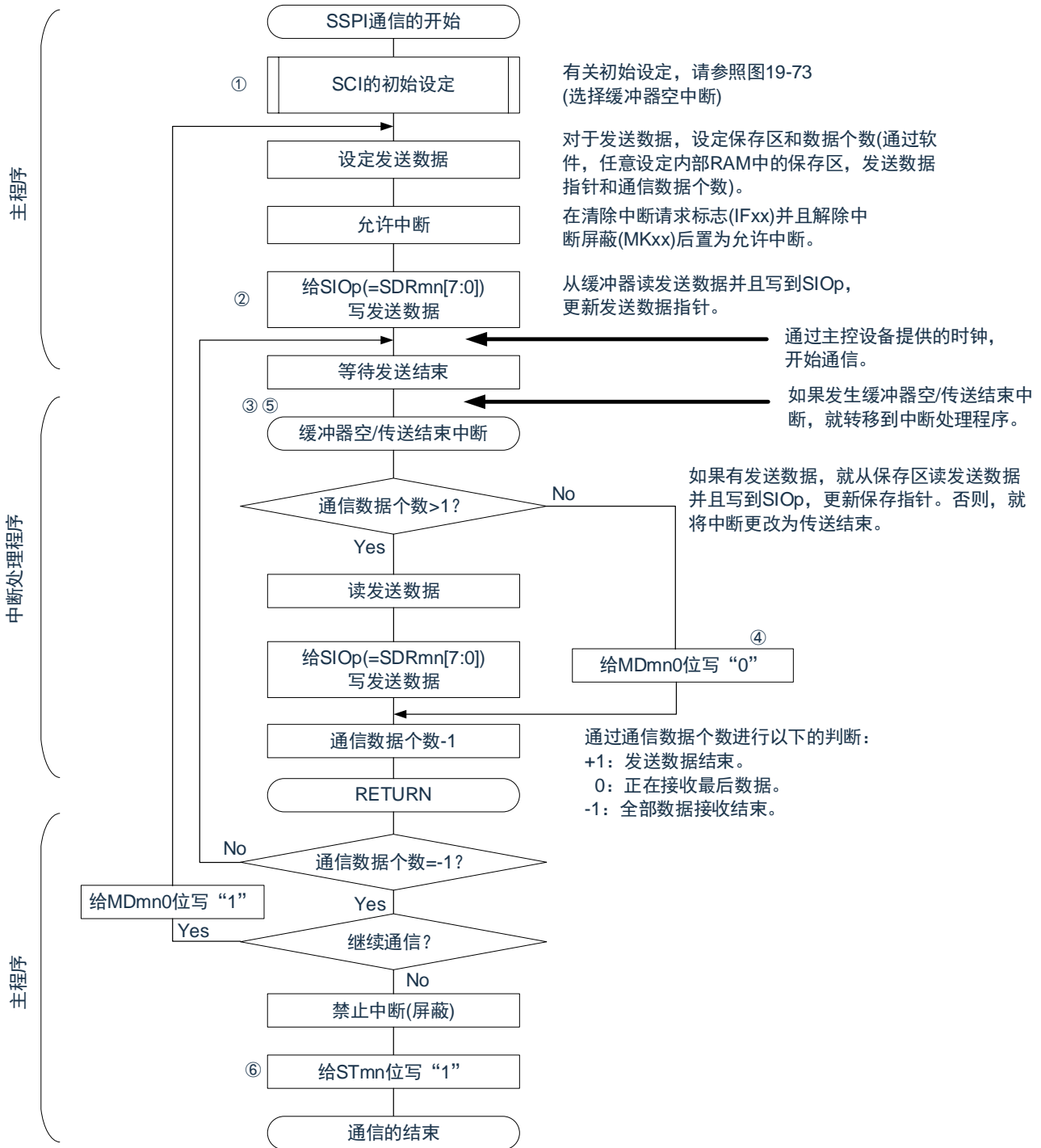


注：如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据，就重写发送数据。

注意：即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

图19-79: 从属发送(连续发送模式)的流程图



备注:

1. 图中的①~⑥对应“图19-78: 从属发送(连续发送模式)的时序图”中的①~⑥。
2. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

19.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下BAT32A237从其他设备接收数据的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDI00、SS00
中断	INTSSPI00
	只限于传送结束中断(禁止设置缓冲器空中断)。
错误检测标志	只有溢出错误检测标志(OVFmn)。
传送数据长度	7位或者8位
传送速率	$\text{Max}, F_{\text{MCK}}/6[\text{Hz}]^{\text{注}1,2}$
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注1: 因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为 $F_{\text{MCK}}/6[\text{Hz}]$ 。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

1. F_{MCK} : 对象通道的运行时钟频率
2. m: 单元号(m=0)n: 通道号(n=0)

(1) 寄存器的设置

图19-80: 从属选择输入功能(SSPI00)从属接收时的寄存器设置内容例子(1/2)



备注:

1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. □: 在从属接收模式中为固定设置。■: 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图19-80: 从属选择输入功能(SSPI00)从属接收时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

备注:

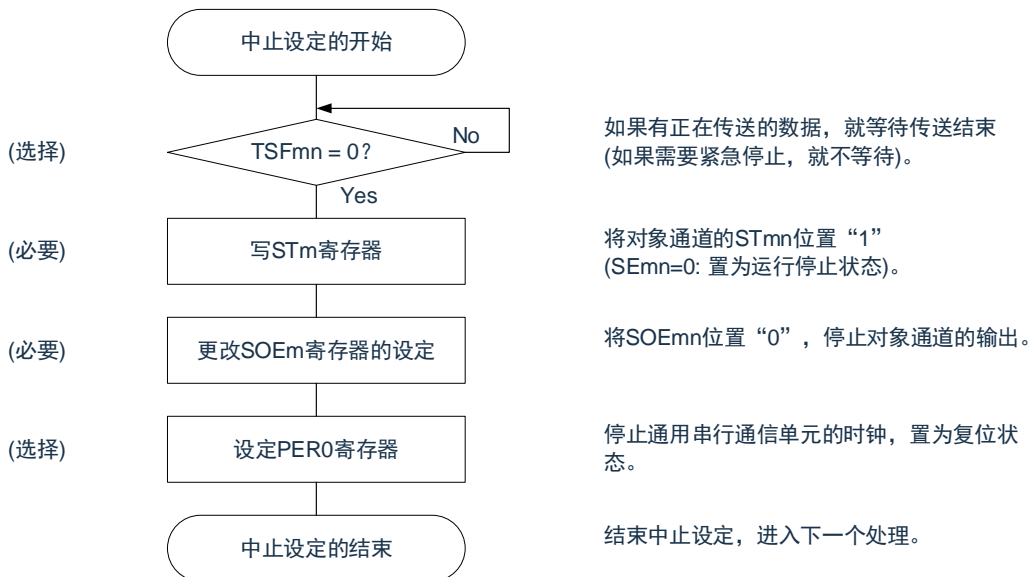
1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. : 在从属接收模式中为固定设置。 : 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-81：从属接收的初始设置步骤



图19-82：从属接收的中止步骤



备注：m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

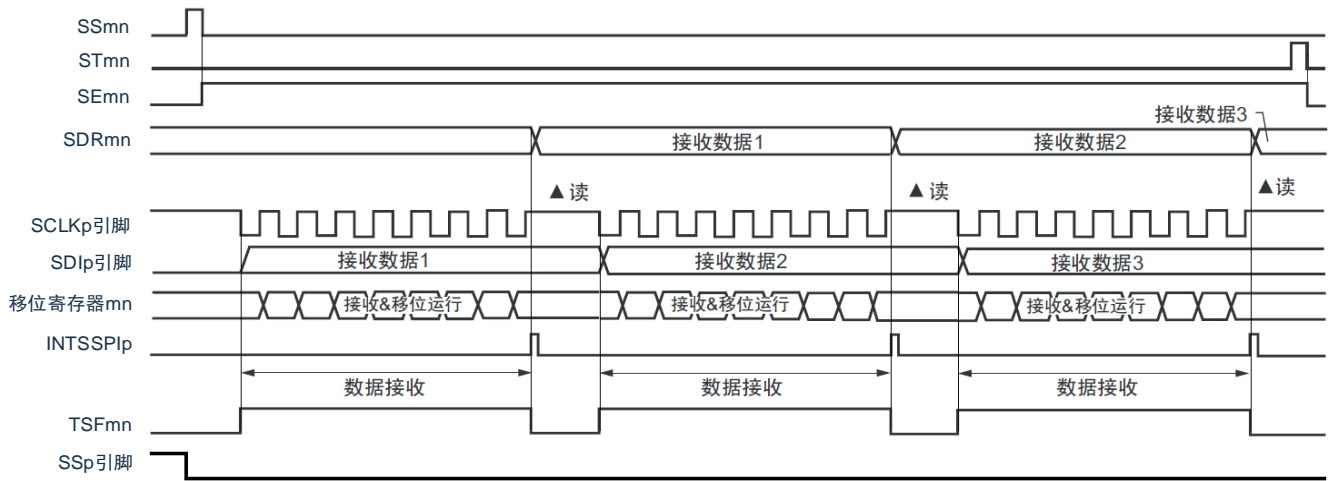
图19-83: 重新开始从属接收的设置步骤



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

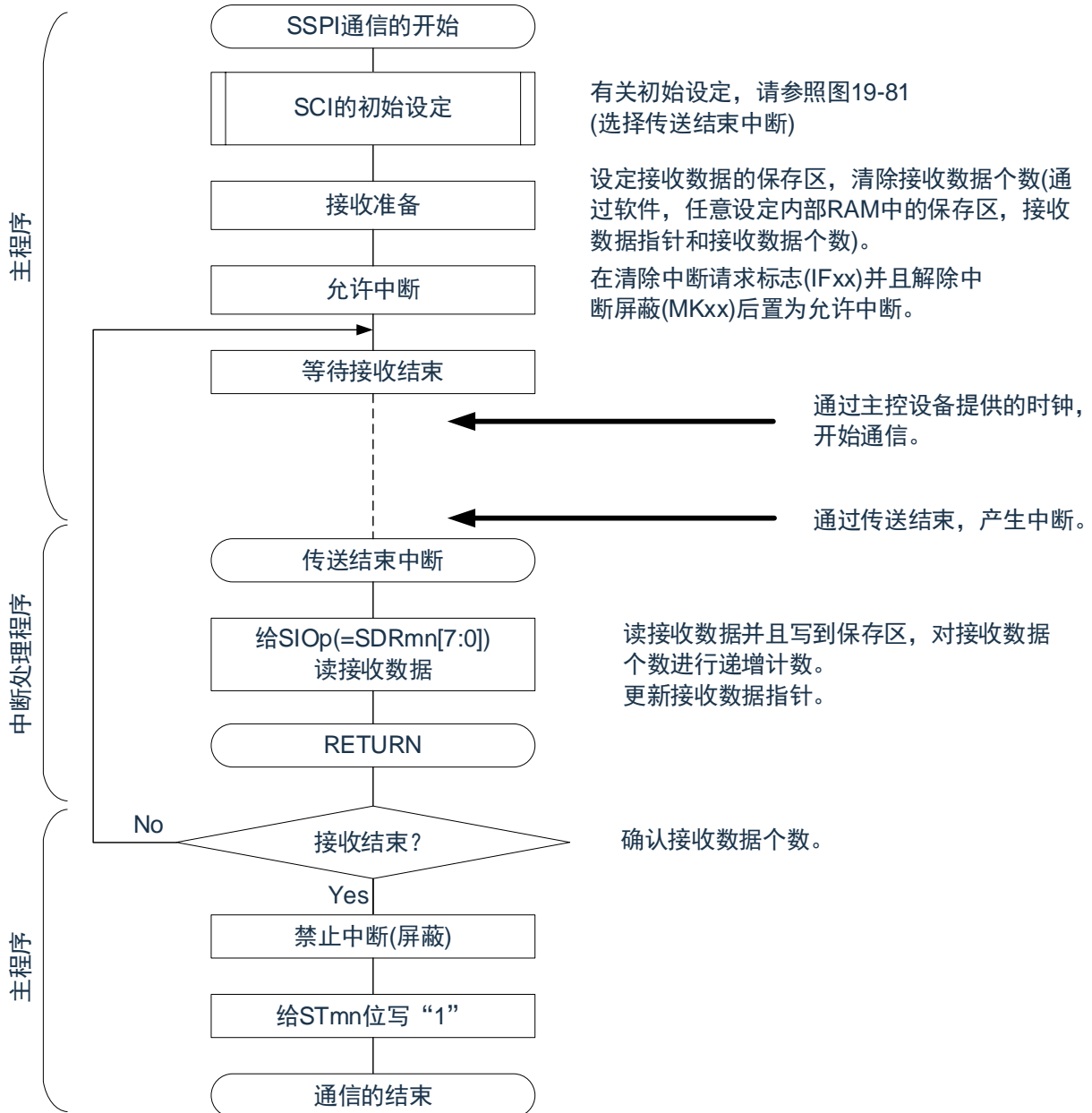
(3) 处理流程(单次接收模式)

图19-84：从属接收(单次接收模式)的时序图(类型1：DAPmn=0、CKPmn=0)



备注：m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

图19-85: 从属接收(单次接收模式)的流程图



19.6.3 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下BAT32A237和其他设备进行数据发送和接收的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDI00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。
错误检测标志	只有溢出错误检测标志(OVFmn)。
传送数据长度	7位或者8位
传送速率	Max.F _{MCK} /6[Hz] ^{注1,2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注1: 因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为F_{MCK}/6[Hz]。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

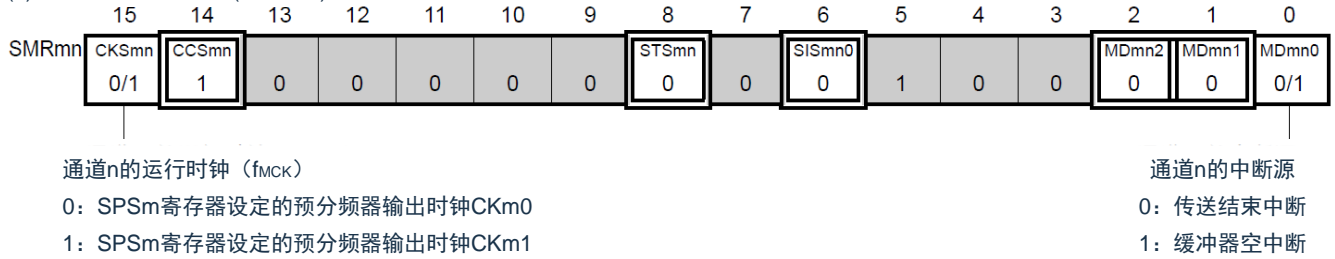
备注:

1. F_{MCK}: 对象通道的运行时钟频率
2. m: 单元号(m=0)n: 通道号(n=0)

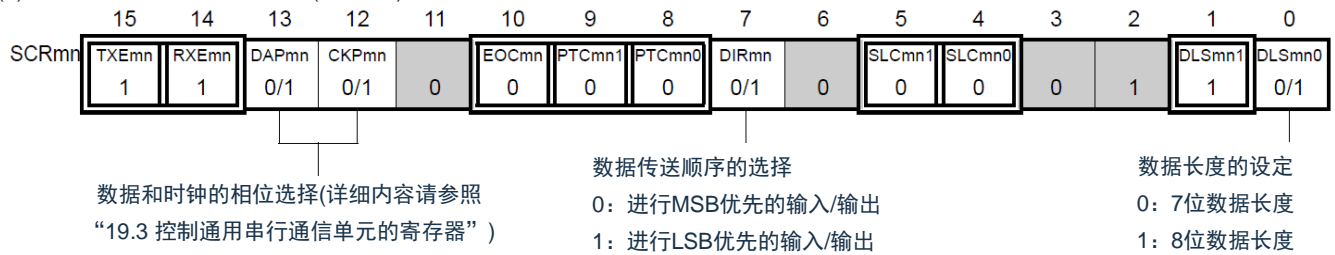
(1) 寄存器的设置

图19-86: 从属选择输入功能(SSPI00)从属发送和接收时的寄存器设置内容例子(1/2)

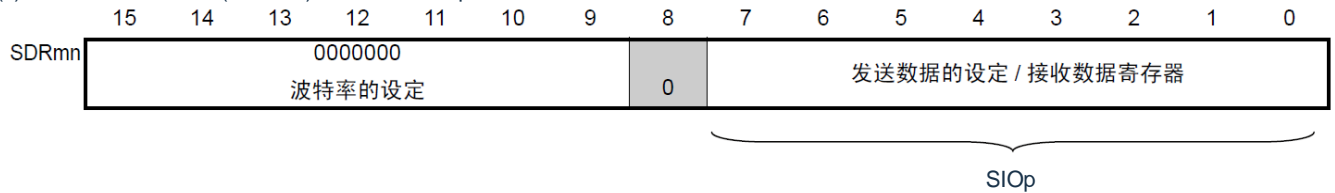
(a) 串行模式寄存器mn(SMRmn)



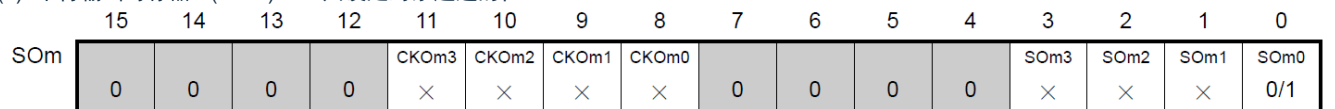
(b) 串行通信运行设定寄存器mn(SCRmn)



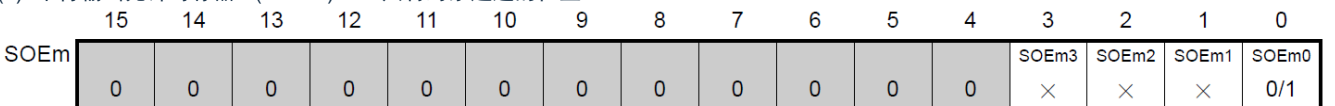
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ...只将对象通道的位置“1”。



注意: 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注:

1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. □: 在从属接收模式中为固定设置。■: 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

图19-86：从属选择输入功能(SSPI00)从属发送和接收时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													×	×	×	0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSIE00						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

注意：在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

备注：

1. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)
2. : 在从属接收模式中为固定设置。 : 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下，设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

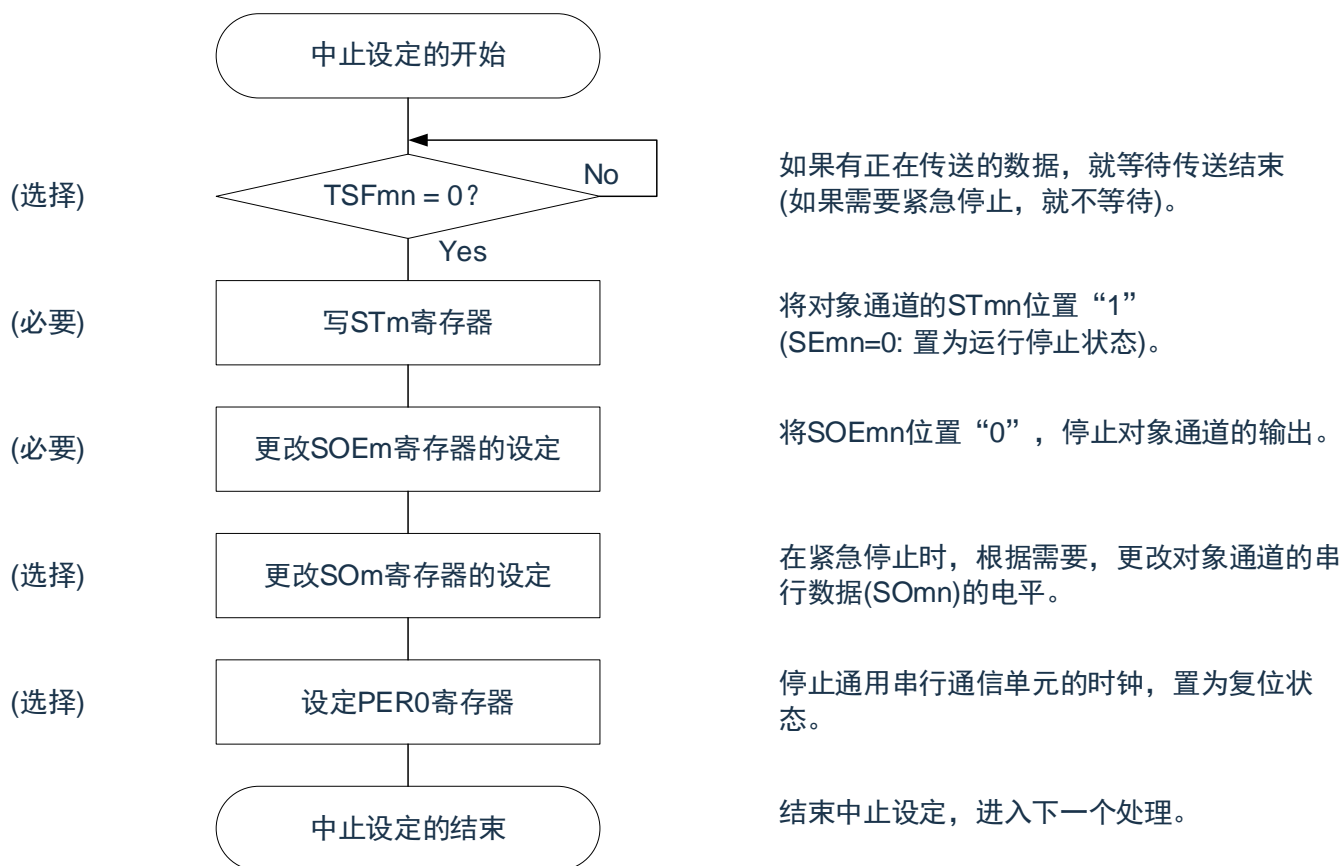
图19-87：从属发送和接收的初始设置步骤



注意：在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

备注：m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

图19-88: 从属发送和接收的中止步骤



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

图19-89: 重新开始从属发送和接收的设置步骤

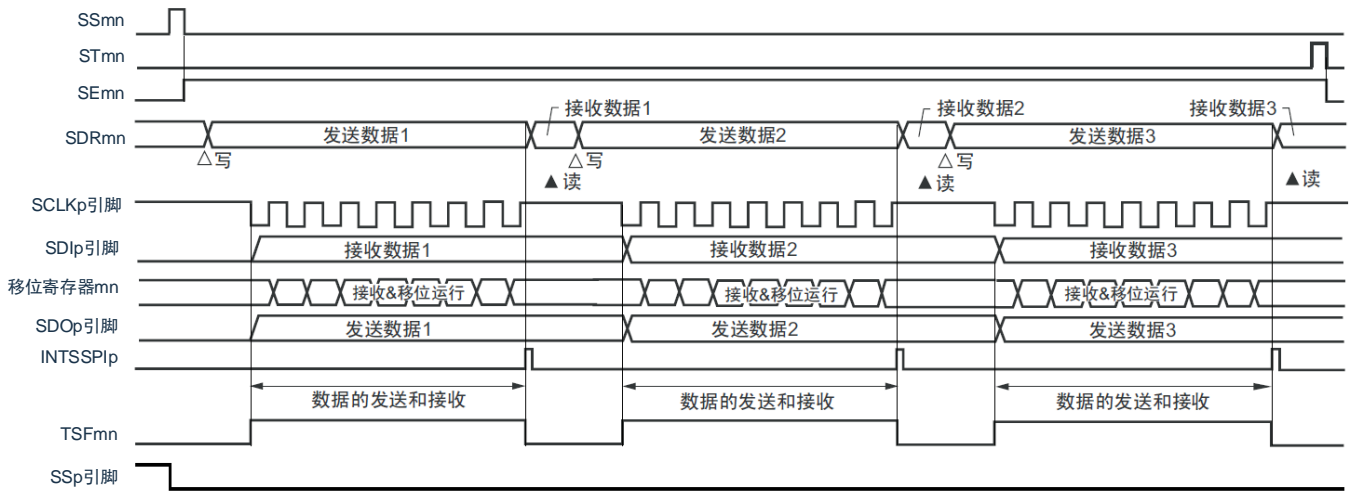


注意:

1. 在 主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。
2. 如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始设置。

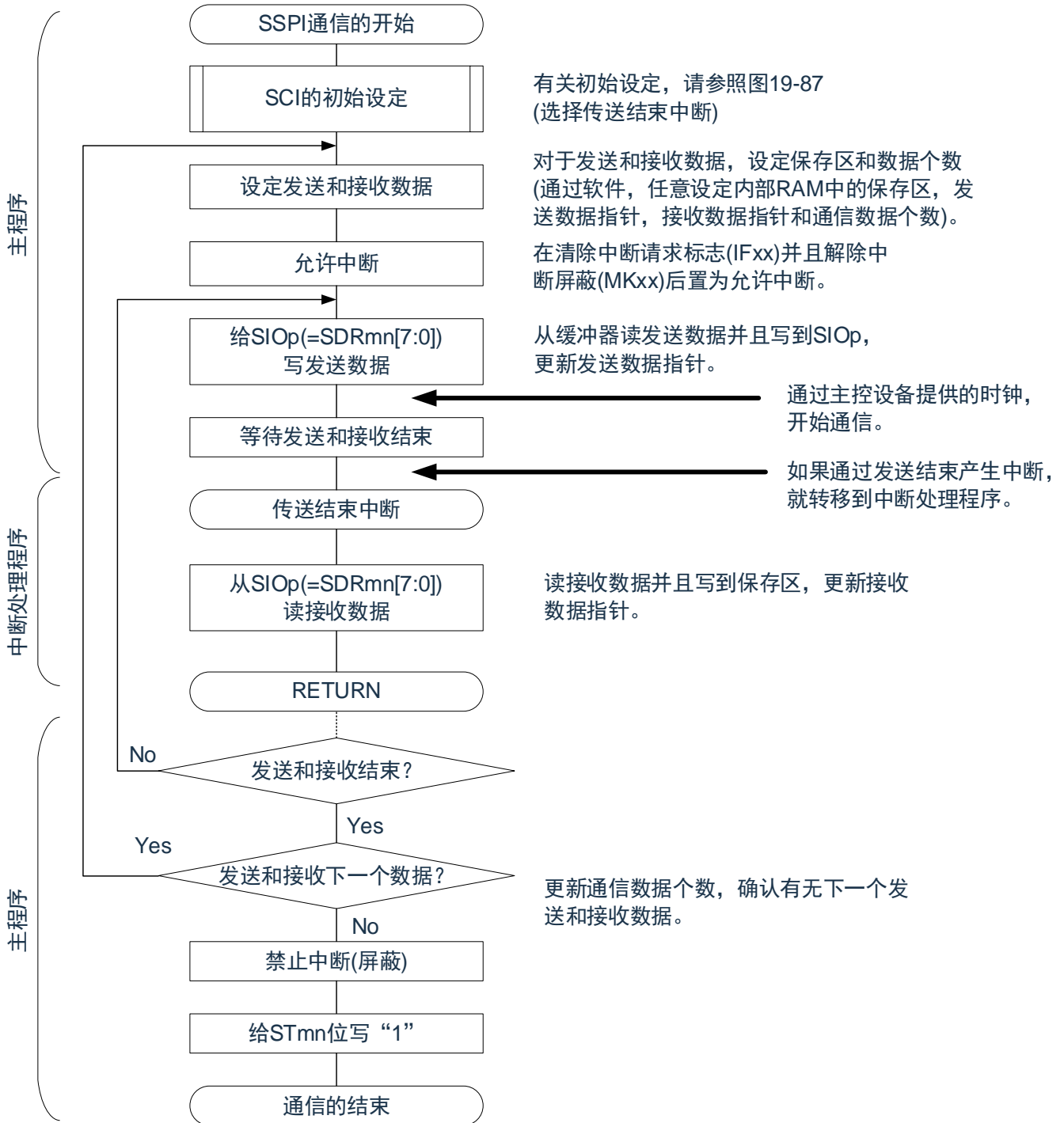
(3) 处理流程(单次发送和接收模式)

图19-90: 从属发送和接收(单次发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

图19-91: 从属发送和接收(单次发送和接收模式)的流程图

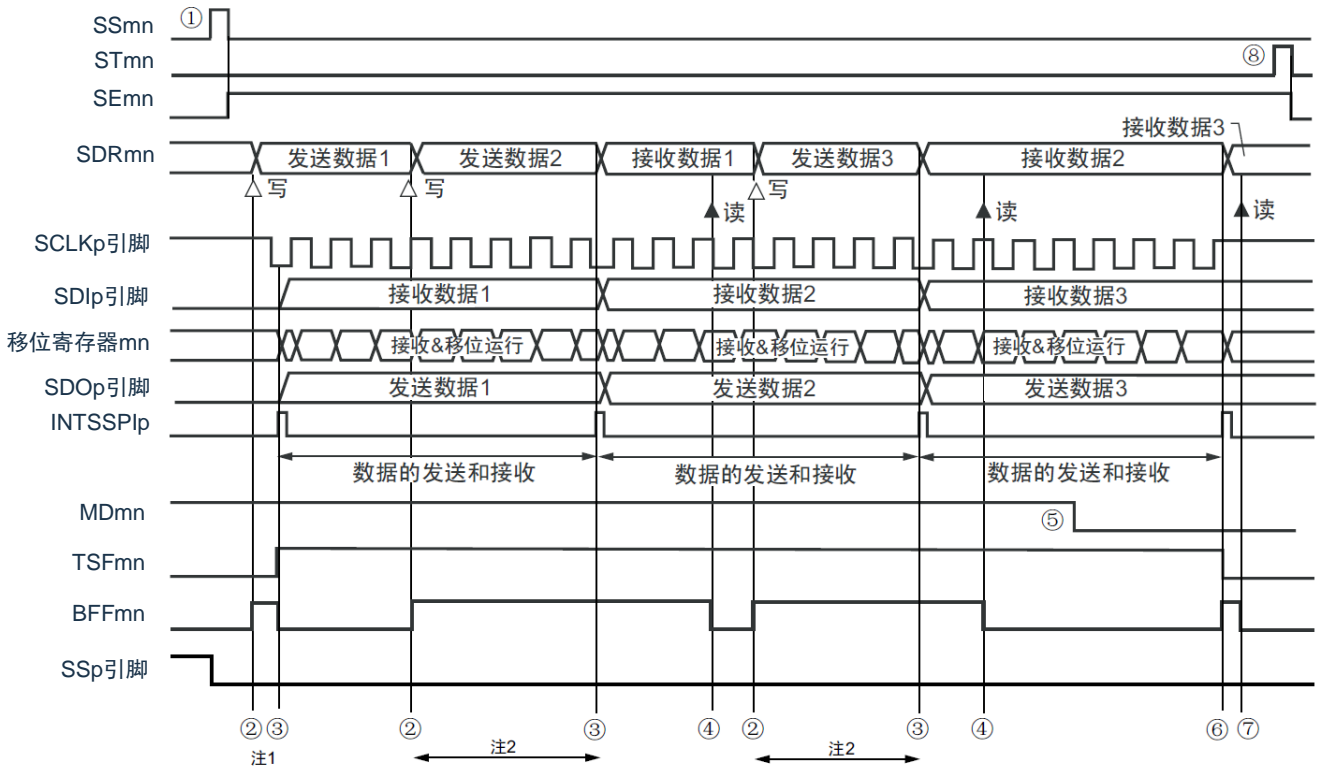


注意: 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注: m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

(4) 处理流程(连续发送和接收模式)

图19-92: 从属发送和接收(连续发送和接收模式)的时序图(类型1: DAPmn=0、CKPmn=0)



注1: 如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据, 就重写发送数据。

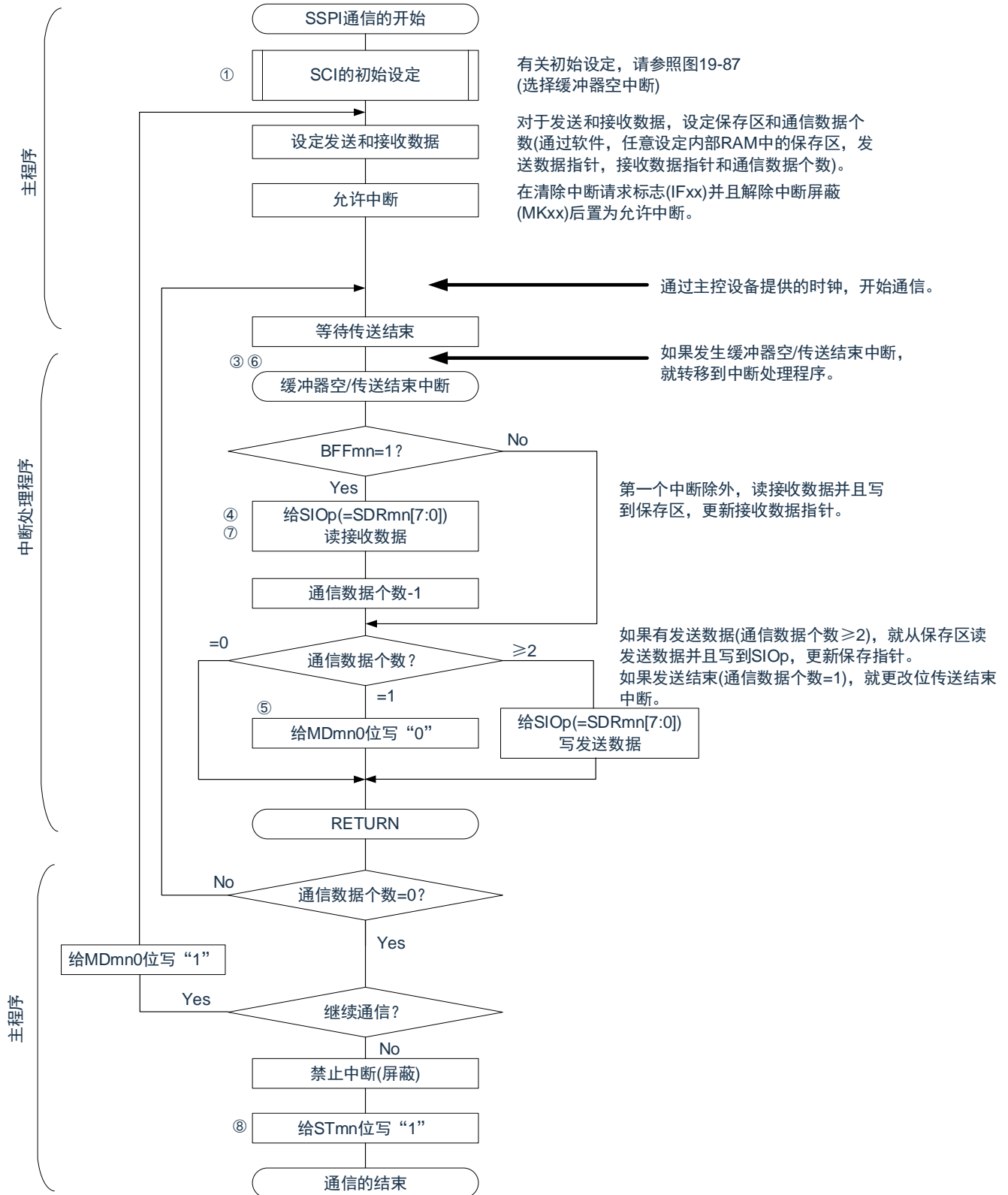
注2: 如果在此期间读取SDRmn寄存器, 就能读发送数据。此时, 不影响传送运行。

注意: 即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是, 为了能赶上最后发送数据的传送结束中断, 必须在开始传送最后一位之前进行改写。

备注:

1. 图中的①~⑧对应“图19-93: 从属发送和接收(连续发送和接收模式)的流程图”中的①~⑧。
2. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

图19-93: 从属发送和接收(连续发送和接收模式)的流程图



注意: 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注:

1. 图中的①~⑧对应“图19-92: 从属发送和接收(连续发送和接收模式)的时序图”中的①~⑧。
2. m: 单元号(m=0)n: 通道号(n=0)p: SSPI号(p=00)

19.6.4 传送时钟频率的计算

从属选择输入功能(SSPI00)通信的传送时钟频率能用以下计算式进行计算。

(1) 从属设备

$$(\text{传送时钟频率}) = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\}^{\text{注}}[\text{Hz}]$$

注：容许的最大传送时钟频率为 $F_{MCK}/6$ 。

备注：m：单元号(m=0)n：通道号(n=0)p：SSPI号(p=00)

表19-3：从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟(F_{MCK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$F_{CLK}=32\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	F_{CLK}	32MHz
	X	X	X	X	0	0	0	1	$F_{CLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$F_{CLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$F_{CLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$F_{CLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$F_{CLK}/2^5$	1KHz
	X	X	X	X	0	1	1	0	$F_{CLK}/2^6$	500KHz
	X	X	X	X	0	1	1	1	$F_{CLK}/2^7$	250KHz
	X	X	X	X	1	0	0	0	$F_{CLK}/2^8$	125KHz
	X	X	X	X	1	0	0	1	$F_{CLK}/2^9$	62.5KHz
	X	X	X	X	1	0	1	0	$F_{CLK}/2^{10}$	31.25KHz
	X	X	X	X	1	0	1	1	$F_{CLK}/2^{11}$	15.63KHz
	X	X	X	X	1	1	0	0	$F_{CLK}/2^{12}$	7.81KHz
	X	X	X	X	1	1	0	1	$F_{CLK}/2^{13}$	3.91KHz
	X	X	X	X	1	1	1	0	$F_{CLK}/2^{14}$	1.95KHz
X	X	X	X	1	1	1	1	$F_{CLK}/2^{15}$	977Hz	

注：要更改被选择为 F_{CLK} 的时钟(更改系统时钟控制寄存器(CKC)的值)时，必须在停止通用串行通信单元(SCI)的运行(串行通道停止寄存器m(STm)=000FH)后进行更改。

备注：

1. X：忽略
2. m：单元号(m=0)n：通道号(n=0)

19.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如图19-94所示。

图19-94：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。 →	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。	-	判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn (SDIRmn) 写“1”。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号(m=0)n：通道号(n=0)

19.7 UART(UART0~UART2)通信的运行

这是通过串行数据发送(TxD)和串行数据接收(RxD)共2条线进行异步通信的功能。使用这2条通信线，按数据帧(由起始位、数据、奇偶校验位和停止位构成)与其他通信方进行异步(使用内部波特率)的数据发送和接收。能通过使用发送专用(偶数通道)和接收专用(奇数通道)共2个通道来实现全双工异步UART通信，而且还能通过组合UART0、通用定时器Timer4(通道3)和外部中断(INTP0)来支持LIN-bus。

[数据的发送和接收]

- 7位、8位或者9位的数据长度^注
- MSB/LSB优先的选择
- 发送和接收数据的电平设置(选择电平是否反相)
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位的检测功能

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误和溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

UART0(单元0的通道0和通道1)支持LIN-bus。[LIN-bus功能]

- | | | |
|--|---|----------------------------|
| <ul style="list-style-type: none"> • 唤醒信号的检测 • 间隔段(BF)的检测 • 同步段的测量、波特率的计算 | } | 使用外部中断(INTP0)和Timer4(通道3)。 |
|--|---|----------------------------|

注：只有UART0支持9位数据长度。

UART0使用SCI0的通道0和通道1。

UART1使用SCI0的通道2和通道3。

UART2使用SCI1的通道0和通道1。

各通道任意选择一个功能使用，除了所选功能以外，其他功能不能运行。

例如，在单元0的通道0和通道1使用UART0时，不能使用SSPI00和IIC01。但是，在使用UART0的同时，不同通道的通道2和通道3能使用SSPI10、UART1或者IIC10。

注意：当用作UART时，发送方(偶数通道)和接收方(奇数通道)只能用于UART。

UART有以下4种通信运行：

- UART发送 (参照19.7.1)
- UART接收 (参照19.7.2)
- LIN发送(只限于UART0) (参照19.8.1)
- LIN接收(只限于UART0) (参照19.8.2)

19.7.1 UART发送

UART发送是BAT32A237微控制器将数据异步发送到其他设备的运行。

UART使用的2个通道中的偶数通道用于UART发送。

UART	UART0	UART1	UART2
对象通道	SCI0的通道0	SCI0的通道2	SCI1的通道0
使用的引脚	TxD0	TxD1	TxD2
中断	INTST0	INTST1	INTST2
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。		
错误检测标志	无		
传送数据长度	7位、8位或者9位 ^{注1}		
传送速率	Max.F _{MCK} /6[bps](SDRmn[15:9]≥2)、Min.F _{CLK} /(2×2 ¹⁵ ×128)[bps] ^{注2}		
数据相位	正相输出(默认值:高电平)。反相输出(默认值:低电平)。		
奇偶校验位	可选择以下内容: <ul style="list-style-type: none"> •无奇偶校验位。 •附加零校验。 •附加偶校验。 •附加奇校验。 		
停止位	可选择以下内容: <ul style="list-style-type: none"> •附加1位。 •附加2位。 		
数据方向	MSB优先或者LSB优先		

注1: 只有UART0支持9位数据长度。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

1. F_{MCK}: 对象通道的运行时钟频率 F_{CLK}: 系统时钟频率
2. m: 单元号(m=0、1)n: 通道号(n=0、2)mn=00、02、10

图19-95: UART(UART0~UART2)的UART发送时的寄存器设置内容例子(2/2)

(e) 串行输出寄存器m(SOm) ...只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
	0	0	0	0	×	×	×	×	0	0	0	0	×	0/1 注	×	0/1 注

0: 串行数据输出值为“0”
 1: 串行数据输出值为“1”

(f) 串行输出允许寄存器m(SOEm) ...只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

(g) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

注：在开始发送前，当对应通道的SOLmn位为“0”时，必须置“1”；当对应通道的SOLmn位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注：

1. m: 单元号(m=0、1)n: 通道号(n=0、2)q: UART号(q=0~2)mn=00、02、10
2. : 在UART发送模式中为固定设置。 : 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下，设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-96: UART发送的初始设置步骤

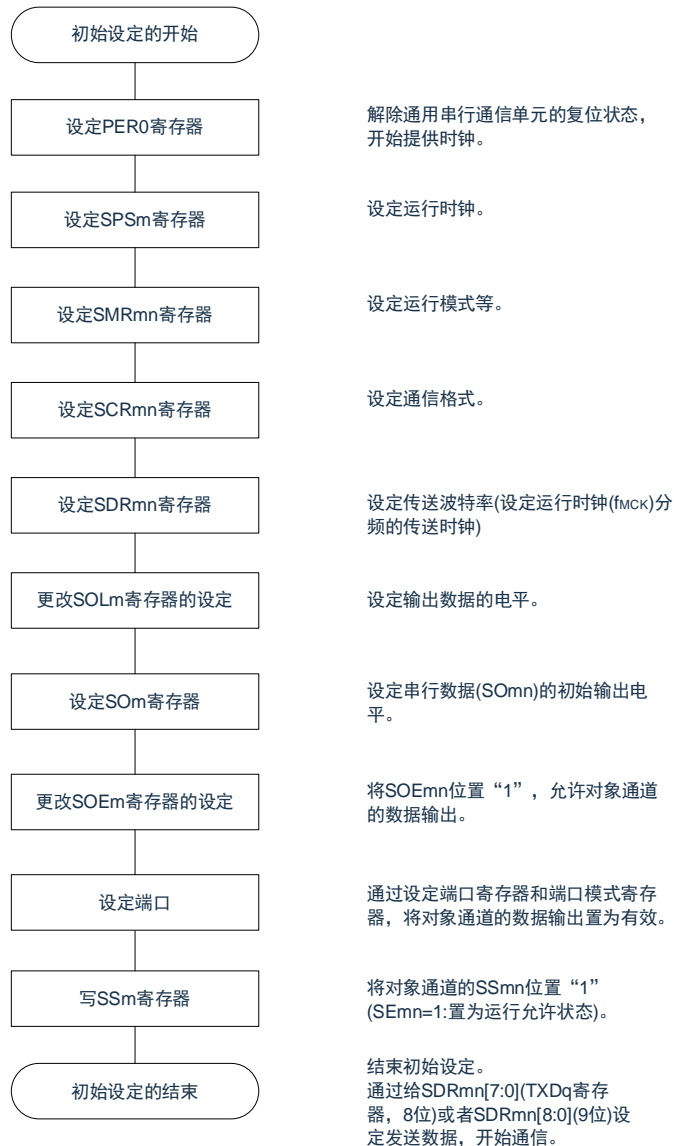


图19-97: UART发送的中止步骤

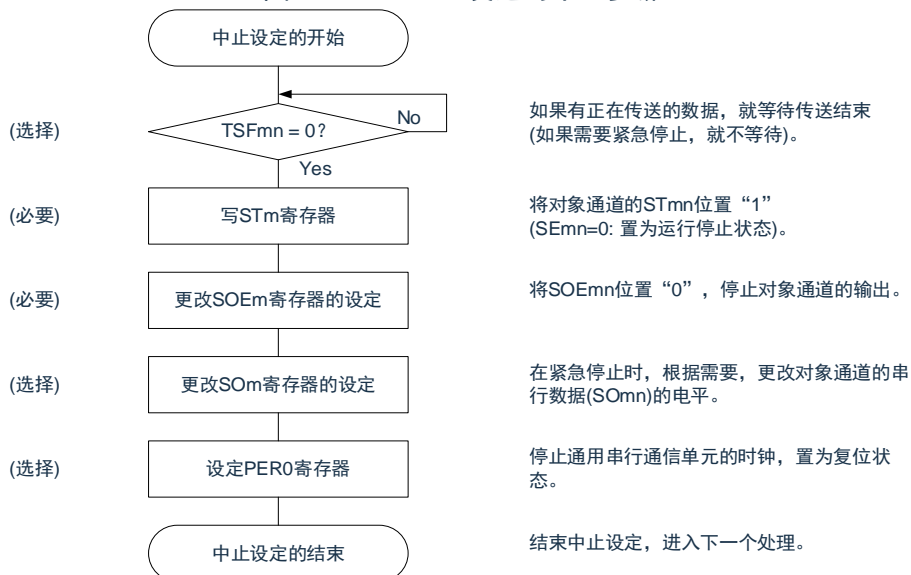
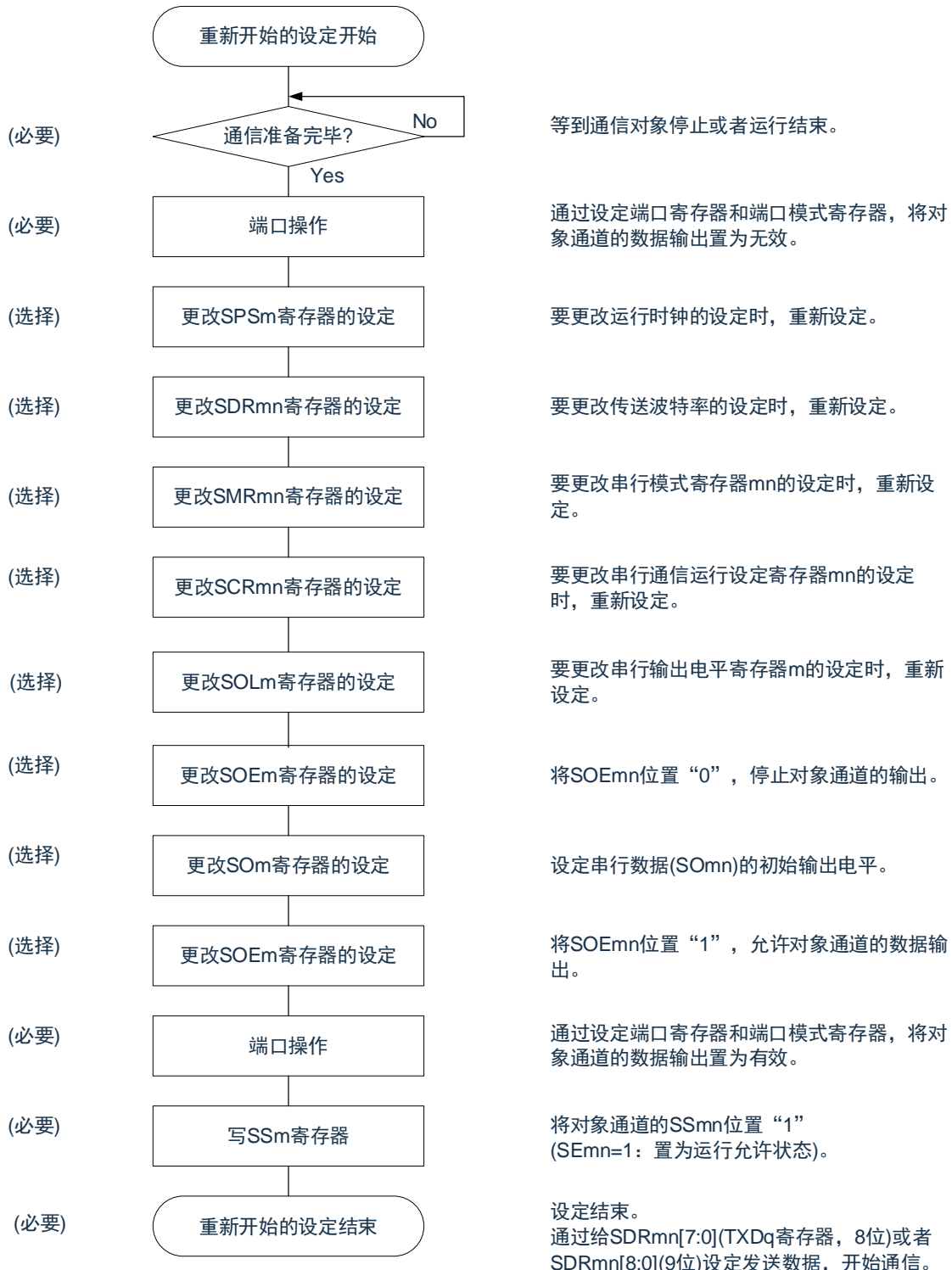


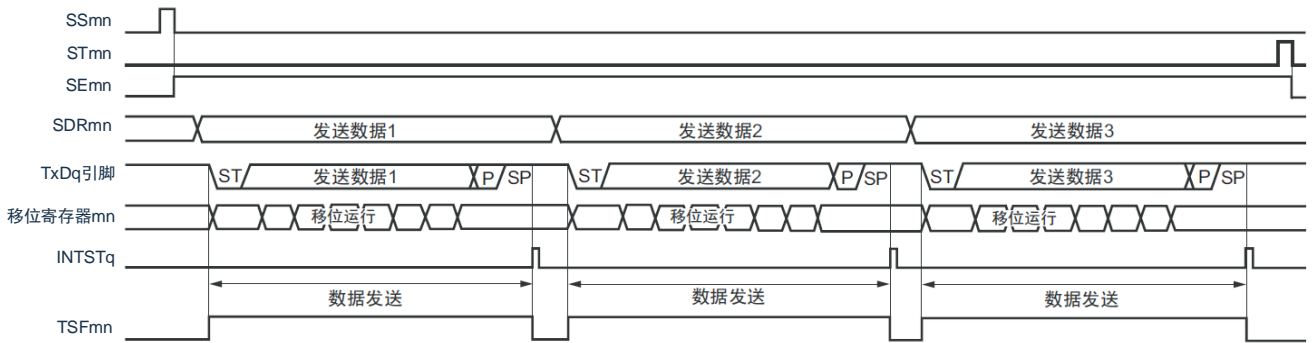
图19-98：重新开始UART发送的设置步骤



备注：如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设置而不是进行重新开始设置。

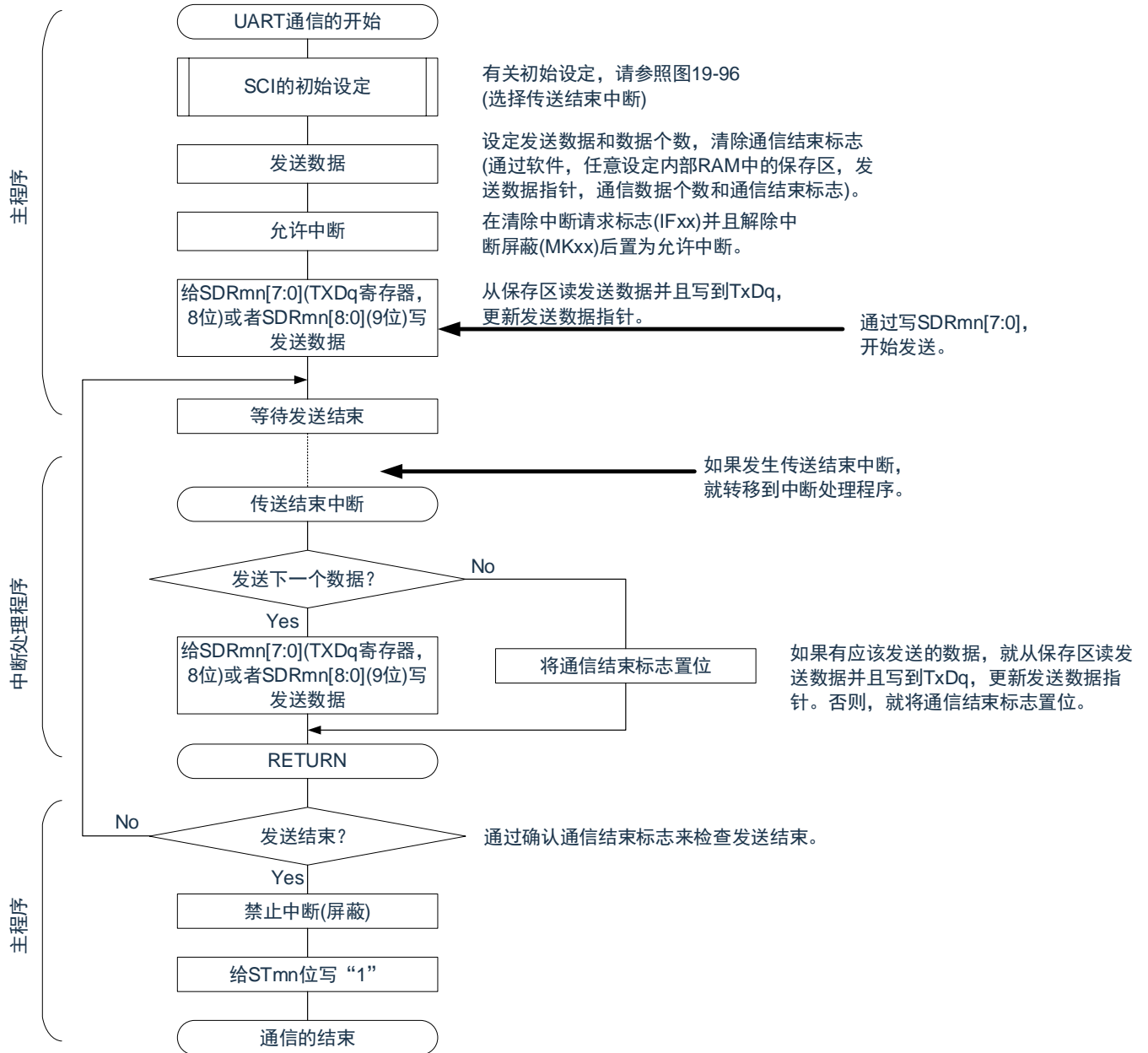
(3) 处理流程(单次发送模式)

图19-99: UART发送(单次发送模式)的时序图



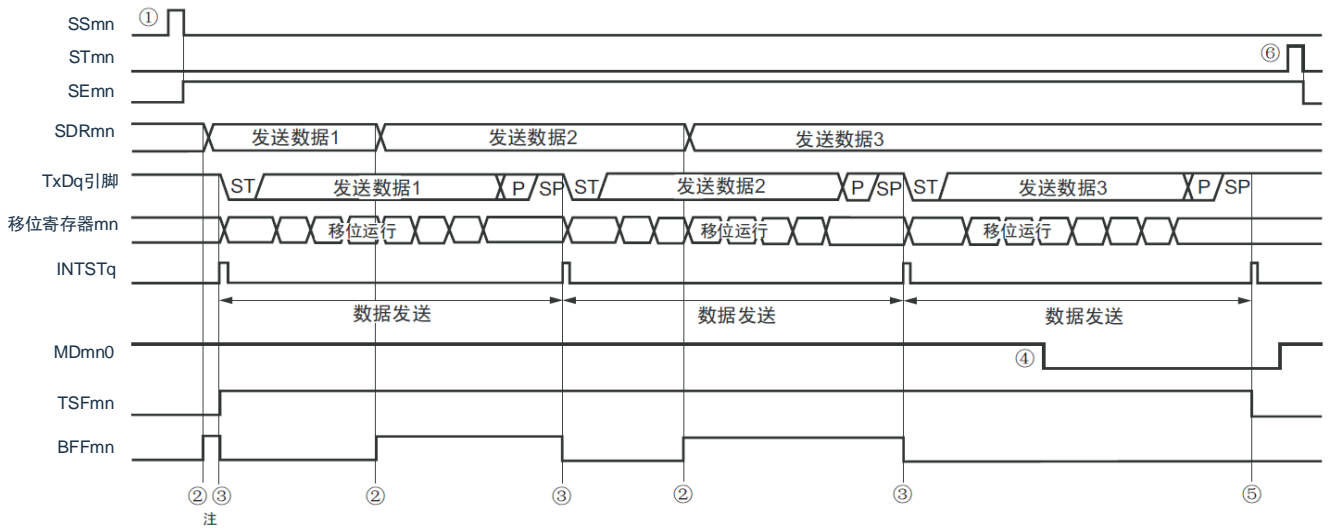
备注: m: 单元号(m=0、1)n: 通道号(n=0、2)q: UART号(q=0~2)mn=00、02、10

图19-100: UART发送(单次发送模式)的流程图



(4) 处理流程(连续发送模式)

图19-101: UART发送(连续发送模式)的时序图

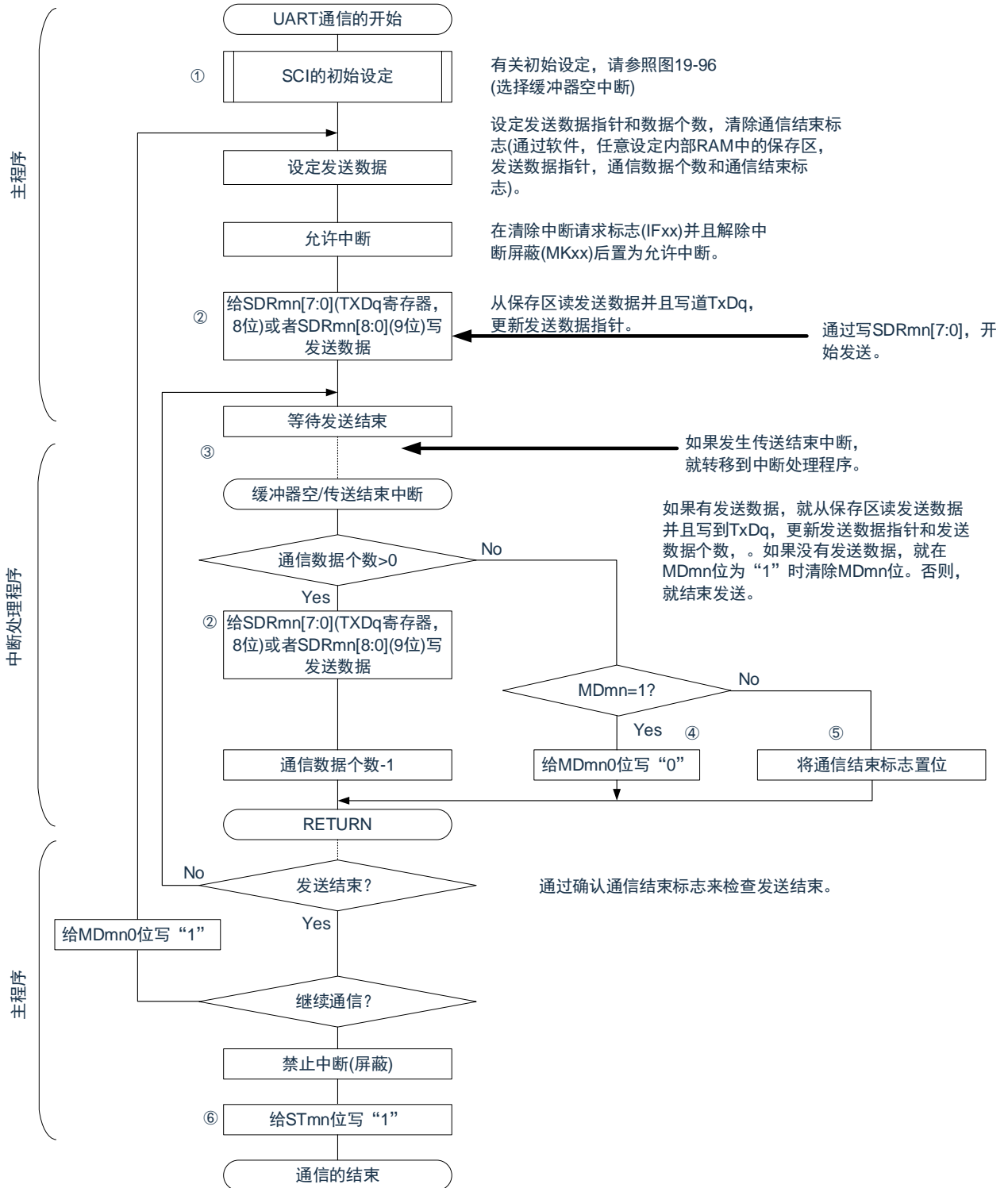


注：如果在串行状态寄存器mn(SSRmn)的BFFmn位为“1”期间(有效数据保存在串行数据寄存器mn(SDRmn)时)给SDRmn寄存器写发送数据，就重写发送数据。

注意：即使在运行中也能改写串行模式寄存器mn(SMRmn)的MDmn0位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：m：单元号(m=0、1)n：通道号(n=0、2)q：UART号(q=0~2)mn=00、02、10

图19-102: UART发送(连续发送模式)的流程图



备注: 图中的①~⑥对应“图19-101: UART发送(连续发送模式)的时序图”中的①~⑥。

19.7.2 UART接收

UART接收是BAT32A237微控制器其他设备异步接收数据的运行。

UART使用的2个通道中的奇数通道用于UART接收。但是，需要设置奇数通道和偶数通道的SMR寄存器。

UART	UART0	UART1	UART2
对象通道	SCI0的通道1	SCI0的通道3	SCI1的通道1
使用的引脚	RxD0	RxD1	RxD2
中断	INTSR0	INTSR1	INTSR2
	只限于传送结束中断(禁止设置缓冲器空中断)。		
错误中断	INTSRE0	INTSRE1	INTSRE2
错误检测标志	<ul style="list-style-type: none"> •帧错误检测标志(FEFmn) •奇偶校验错误检测标志(PEFmn) •溢出错误检测标志(OVFmn) 		
传送数据长度	7位、8位或者9位 ^{注1}		
传送速率	Max. $F_{MCK}/6[\text{bps}](\text{SDRmn}[15:9] \geq 2)$ 、 Min. $F_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$		
数据相位	正相输出(默认值: 高电平)。反相输出(默认值: 低电平)。		
奇偶校验位	可选择以下内容: <ul style="list-style-type: none"> •无奇偶校验位(无奇偶校验)。 •附加零校验(无奇偶校验)。 •偶校验 •奇校验 		
停止位	附加1位。		
数据方向	MSB优先或者LSB优先		

注1: 只有UART0支持9位数据长度。

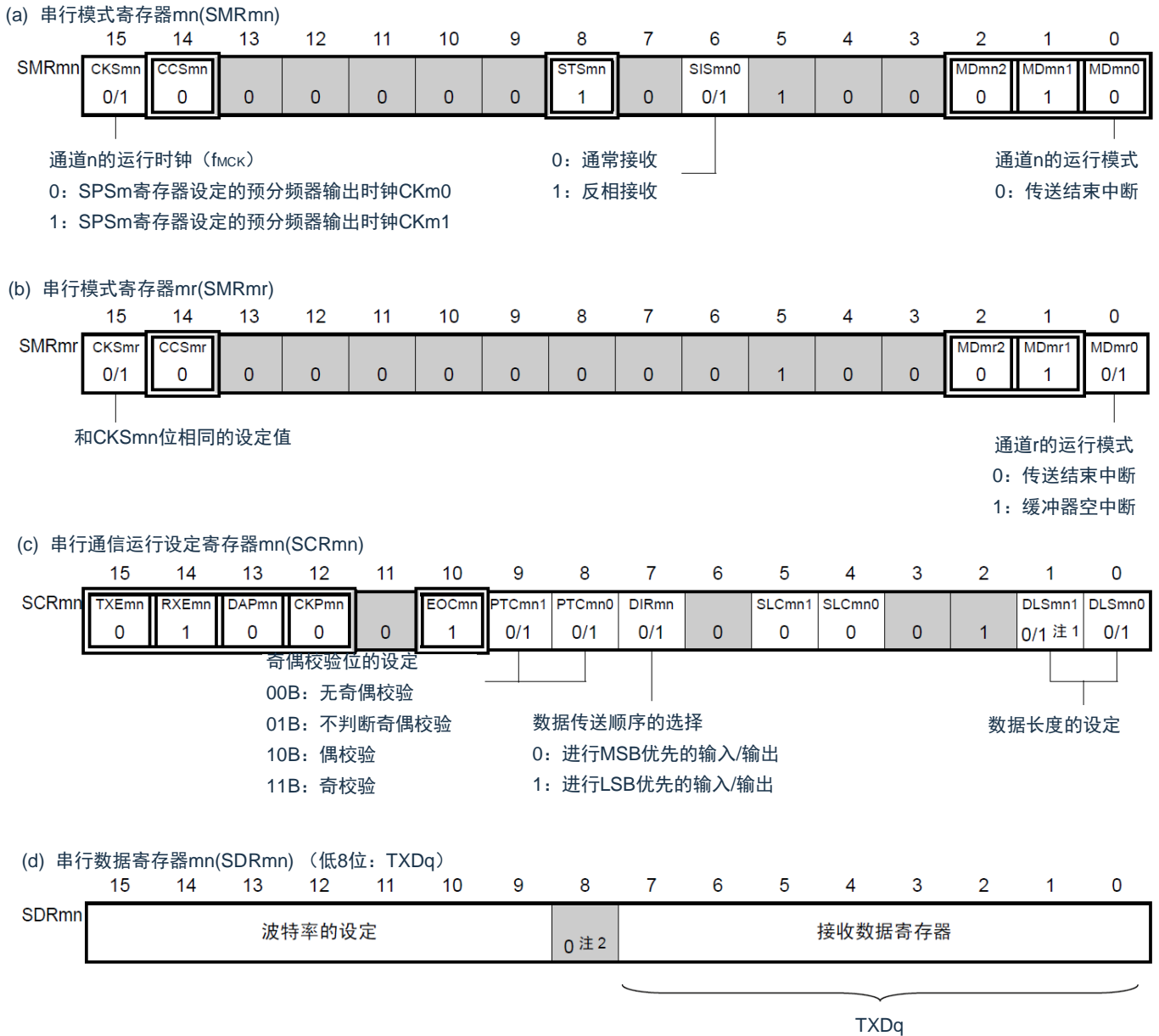
注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注:

1. F_{MCK} : 对象通道的运行时钟频率
 F_{CLK} : 系统时钟频率
2. m: 单元号(m=0、1)n: 通道号(n=1、3)mn=01、03、11

(1) 寄存器的设置

图19-103: UART(UART0~UART2)的UART接收时的寄存器设置内容例子(1/2)



注1: 只限于SCR01寄存器, 其他固定为“1”。

注2: 当进行9位数据长度的通信时, SDRm1寄存器的bit0~8为发送数据的设置区。只有UART0才能进行9位数据长度的通信。

注意: 在UART接收时, 还必须设置与通道n成对的通道r的SMRmr寄存器。

备注:

1. m: 单元号(m=0、1)n: 通道号(n=1、3)mn=01、03、11
 r: 通道号(r=n-1)q: UART号(q=0~2)
2. □: 在UART接收模式中为固定设置。■: 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

图19-103: UART(UART0~UART2)的UART接收时的寄存器设置内容例子(2/2)

(e) 串行输出寄存器m(SOm) ...在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
	0	0	0	0	×	×	×	×	0	0	0	0	×	×	×	×

(f) 串行输出允许寄存器m(SOEm) ...在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	×

(g) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	×

备注:

1. m: 单元号(m=0、1)
2. : 在UART接收模式中为固定设置。 : 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图19-104: UART接收的初始设置步骤



注意: 必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个F_{MCK}时钟, 然后将SSmn位置“1”。

图19-105: UART接收的中止步骤

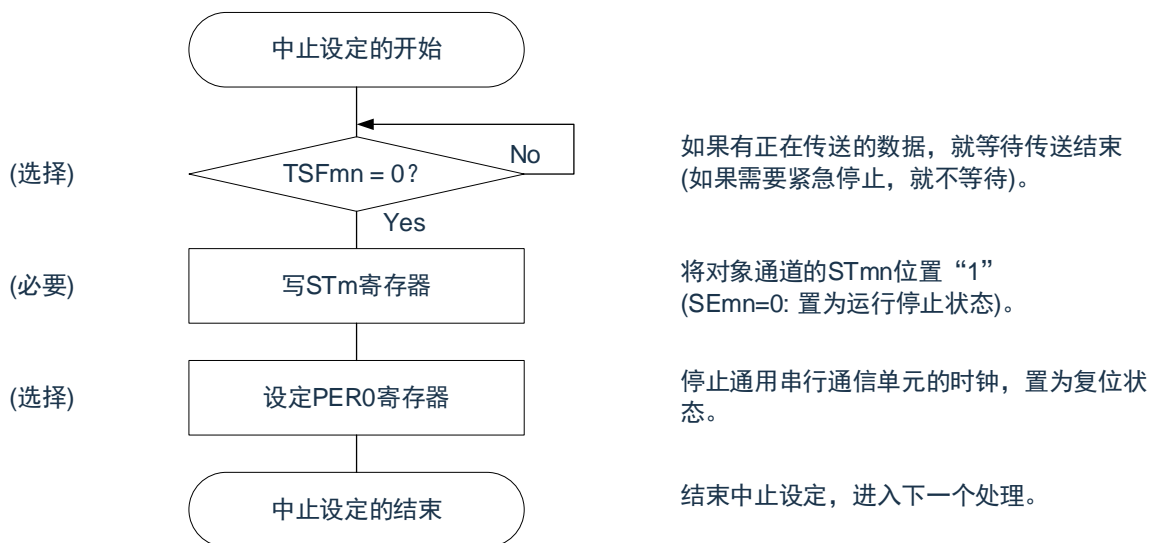


图19-106: 重新开始UART接收的设置步骤

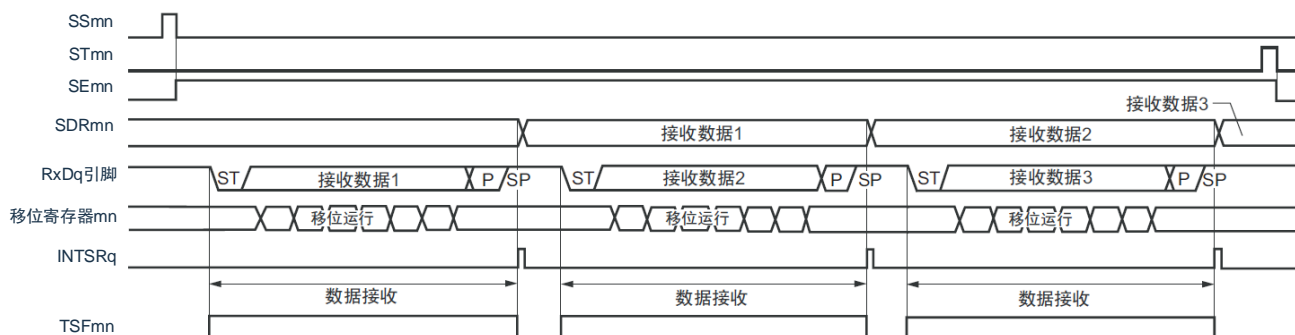


注意: 必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个 F_{MCK} 时钟, 然后将SSmn位置“1”。

备注: 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象停止或者通信结束后进行初始设置而不是进行重新开始设置。

(3) 处理流程

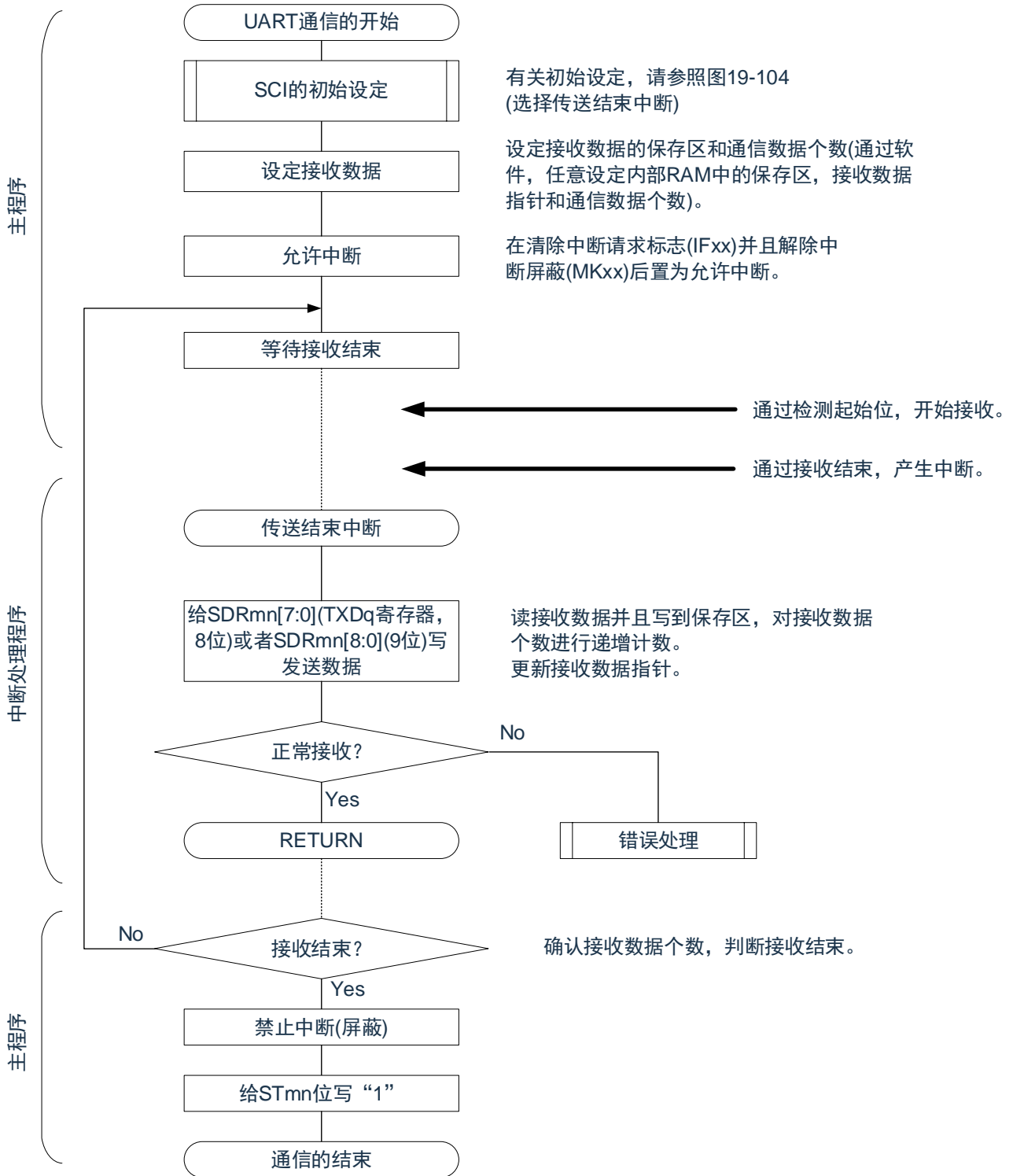
图19-107: UART接收的时序图



备注: m: 单元号(m=0、1)n: 通道号(n=1、3)mn=01、03、11

r: 通道号(r=n-1)q: UART号(q=0~2)

图19-108: UART接收的流程图



19.7.3 波特率的计算

(1) 波特率的计算式

UART(UART0~UART2)通信的波特率能用以下计算式进行计算:

$$\text{(波特率)} = \{\text{对象通道的运行时钟 (F}_{MCK}\text{) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

注意: 禁止将串行数据寄存器mn(SDRmn)的SDRmn[15:9]设置为“0000000B”和“0000001B”。

备注:

1. 因为在使用UART时SDRmn[15:9]的值为SDRmn寄存器的bit15~9的值(0000010B~1111111B), 所以为2~127。
2. m: 单元号(m=0、1)n: 通道号(n=0~2)mn=00~03、10~11

运行时钟(F_{MCK})取决于串行时钟选择寄存器m(SPSm)和串行模式寄存器mn(SMRmn)的bit15(CKSmn位)。

表19-4: UART运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟(F _{MCK}) ^注	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	F _{CLK} =32MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	500KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	250KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	125KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	62.5KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	31.25KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	15.63KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	7.81KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	3.91KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.95KHz
X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	977Hz	
1	0	0	0	0	X	X	X	X	F _{CLK}	32MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	500KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	250KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	125KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	62.5KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	31.25KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	15.63KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	7.81KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	3.91KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.95KHz
1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	977Hz	

注: 要更改被选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)时, 必须在停止通用串行通信单元(SCI)的运行(串行通道停止寄存器m(STm)=000FH)后进行更改。

备注:

1. X: 忽略
2. m: 单元号(m=0、1)n: 通道号(n=0~2)mn=00~03、10~11

(2) 发送时的波特率误差

UART(UART0~UART2)通信发送时的波特率误差能用以下计算式进行计算，必须将发送方的波特率设置在接收方的波特率容许范围内。

$$(\text{波特率误差}) = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100[\%]$$

$F_{CLK}=32\text{MHz}$ 时的UART波特率的设置例子如下所示。

UART波特率 (目标波特率)	$F_{CLK}=32\text{MHz}$			
	运行时钟(F_{MCK})	SDRmn[15:9]	波特率的计算值	与目标波特率的误差
300bps	$F_{CLK}/2^9$	103	300.48bps	+0.16%
600bps	$F_{CLK}/2^8$	103	600.96bps	+0.16%
1200bps	$F_{CLK}/2^7$	103	1201.92bps	+0.16%
2400bps	$F_{CLK}/2^6$	103	2403.85bps	+0.16%
4800bps	$F_{CLK}/2^5$	103	4807.69bps	+0.16%
9600bps	$F_{CLK}/2^4$	103	9615.38bps	+0.16%
19200bps	$F_{CLK}/2^3$	103	19230.8bps	+0.16%
31250bps	$F_{CLK}/2^3$	63	31250.0bps	$\pm 0.0\%$
38400bps	$F_{CLK}/2^2$	103	38461.5bps	+0.16%
76800bps	$F_{CLK}/2$	103	76923.1bps	+0.16%
153600bps	F_{CLK}	103	153846bps	+0.16%
312500bps	F_{CLK}	50	313725bps	$\pm 0.39\%$

备注：m：单元号(m=0、1)n：通道号(n=0、2)mn=00、02、10

(3) 接收时的波特率容许范围

UART(UART0~UART2)通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设置在接收方的波特率容许范围内。

$\text{(可接收的最大波特率)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$
$\text{(可接收的最小波特率)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$

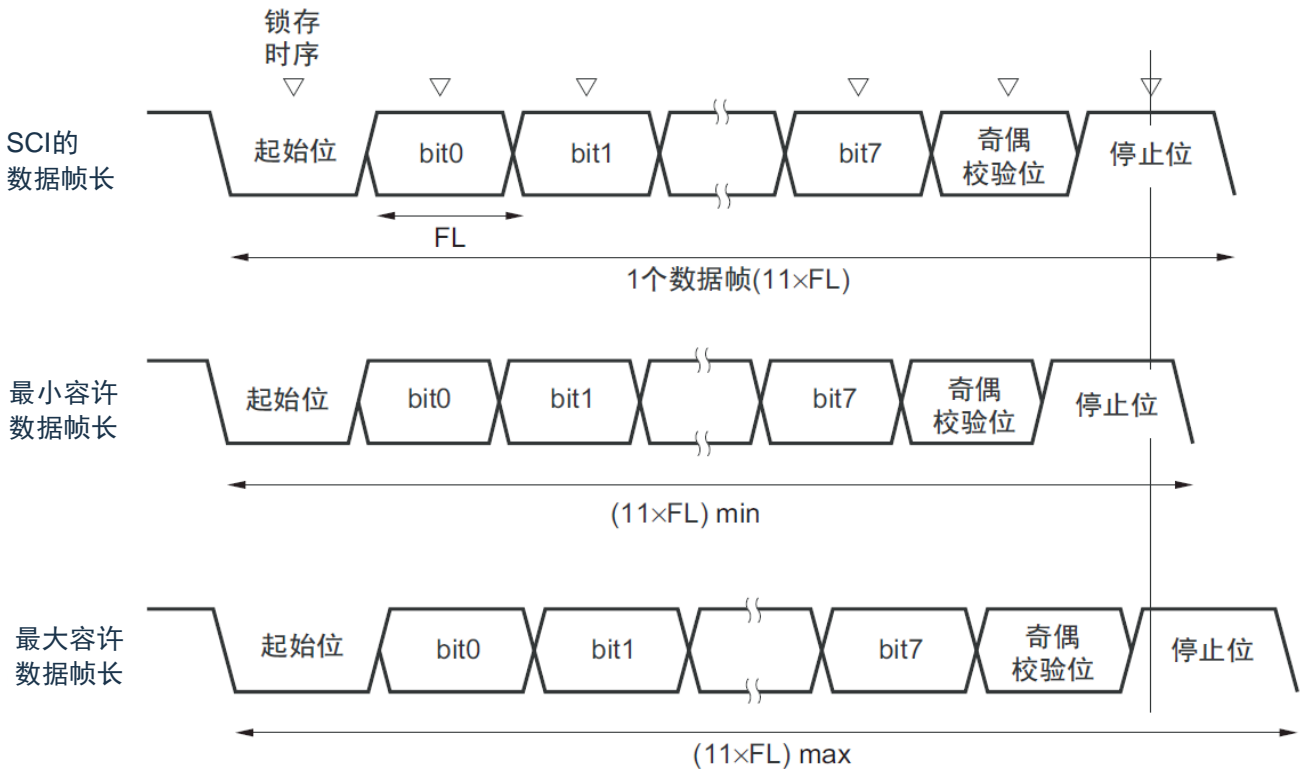
Brate: 接收方的波特率的计算值(参照“19.7.4(1)波特率的计算式”)

k : SDRmn[15:9]+1

Nfr: 1个数据帧的帧长[位]=(起始位)+(数据长度)+(奇偶校验位)+(停止位)

备注: m: 单元号(m=0、1)n: 通道号(n=1、3)mn=01、03、11

图19-109: 接收时的波特率容许范围(1个数据帧=11位的情况)



如图19-109所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器mn(SDRmn)的bit15~9设置的分频比。如果最后的数据(停止位)能赶上此锁存时序，就能正常接收。

19.7.4 在UART(UART0~UART2)通信过程中发生错误时的处理步骤

在UART(UART0~UART2)通信过程中发生错误时的处理步骤如图19-110和图19-111所示。

图19-110：发生奇偶校验错误或者溢出错时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn(SDRmn)。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn(SSRmn)。		判断错误种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn(SDIRmn)写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

图19-111：发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn (SIRmn)。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m (STm) 的STmn位置“1”。	串行通道允许状态寄存器m (SEm) 的SEmn位为“0”并且通道n为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存m (SSm) 的SSmn位置“1”。	串行通道允许状态寄存器m (SEm) 的SEmn位为“1”并且通道n为可运行状态。	

备注：m：单元号(m=0、1)n：通道号(n=0~3)mn=00~03、10~11

19.8 LIN通信的运行

19.8.1 LIN发送

在UART发送中，UART0支持LIN通信。

LIN发送使用单元0的通道0。

UART	UART0	UART1	UART2
LIN通信的支持	能	不能	不能
对象通道	SCI0的通道0	—	—
使用的引脚	TxD0	—	—
中断	INTST0	—	—
	可选择传送结束中断(单次传送模式)或者缓冲器空中断(连续传送模式)。		
错误检测标志	无		
传送数据长度	8位		
传送速率 ^注	Max.F _{MCK} /6[bps](SDR00[15:9]≥2)、Min.F _{CLK} /(2×2 ¹⁵ ×128)[bps]		
数据相位	正相输出(默认值:高电平)。 反相输出(默认值:低电平)。		
奇偶校验位	无奇偶校验位。		
停止位	附加1位。		
数据方向	LSB优先		

注：必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用，并且在LIN通信中经常使用2.4/9.6/19.2kbps。

备注：F_{MCK}：对象通道的运行时钟频率

F_{CLK}：系统时钟频率

LIN是Local Interconnect Network的简称，是为降低汽车网络成本的低速(1~20kbps)串行通信协议。LIN通信是单主控通信，一个主控设备最多可连接15台从属设备。

LIN从属设备用于开关、传动装置和传感器等的控制，这些装置通过LIN连接到主控设备。

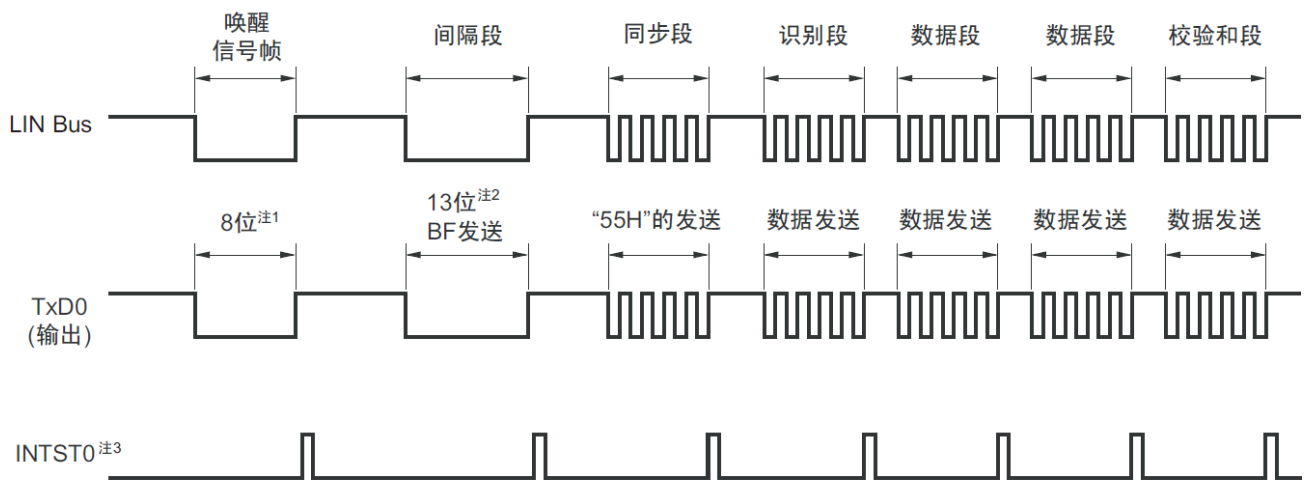
LIN主控一般连接CAN(Controller Area Network)等的网络。

LIN总线是单线方式的总线，通过符合ISDO9141的收发器连接各节点。

根据LIN协议，主控设备发送附加波特率信息的帧，从属设备接收此帧并且校正与主控设备的波特率误差。因此，如果从属设备的波特率误差不大于±15%，就能进行通信。

LIN的发送操作的概要如图19-112所示。

图19-112: LIN的发送操作



注1: 为了满足唤醒信号的规定，设置波特率并且通过发送“80H”的数据进行对应。

注2: 间隔段规定为13位宽的低电平输出，因此假设主传送使用的波特率为N[bps]，间隔段使用的波特率如下：

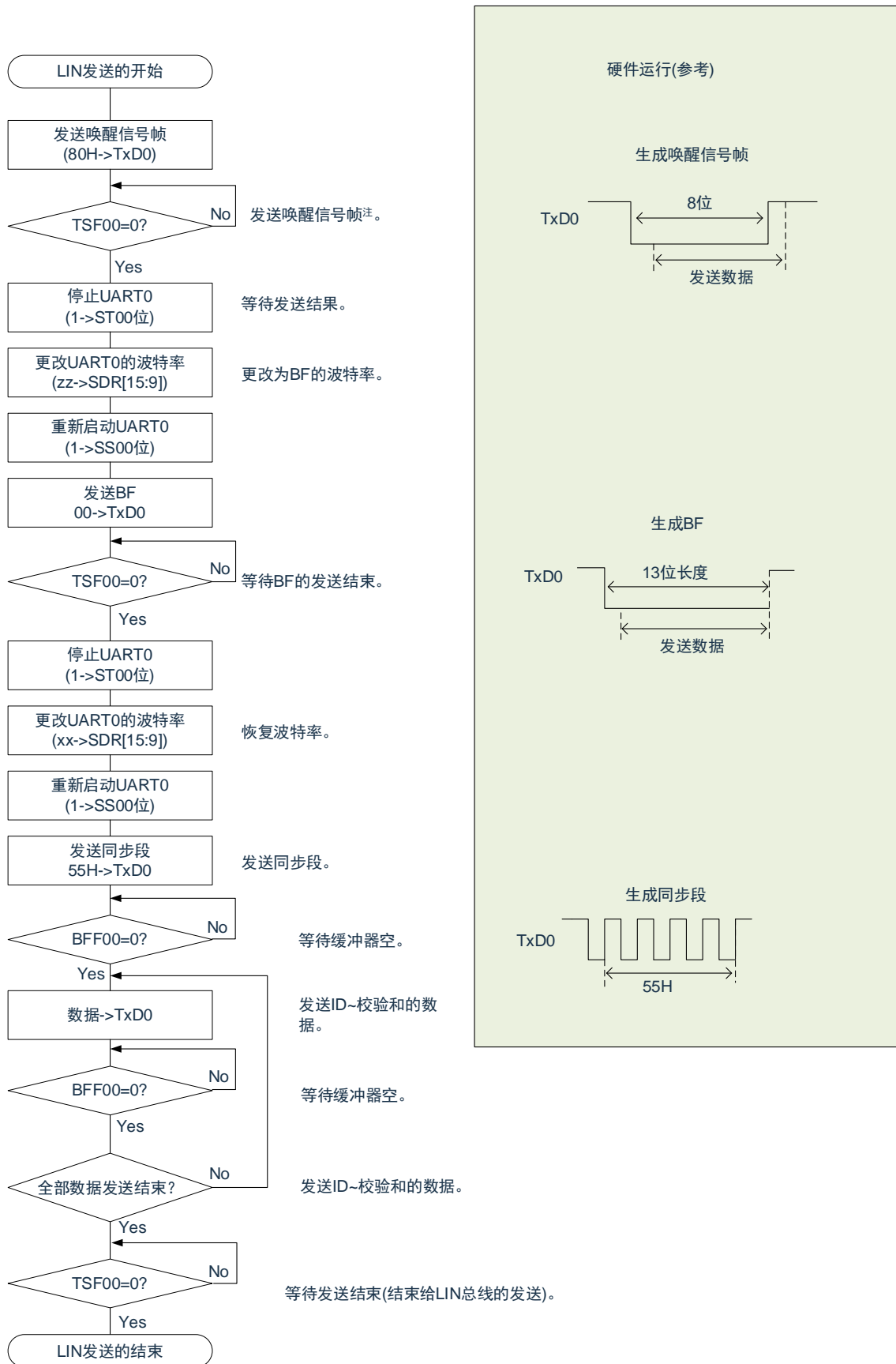
$$(间隔段的波特率)=9/13 \times N$$

通过此波特率发送“00H”的数据，生成间隔段。

注3: 在各数据发送结束时输出INTST0，而且在BF发送时也输出INTST0。

备注: 由软件控制各段间的间隔。

图19-113: LIN发送的流程图



注：只限于从LIN-bus睡眠状态启动的情况。

备注：这是从结束UART的初始设置并且允许从属发送开始的流程。

19.8.2 LIN接收

在UART接收中，UART0支持LIN通信。

LIN接收使用单元0的通道1。

UART	UART0	UART1	UART2	UART3
LIN通信的支持	能	不能	不能	不能
对象通道	SCI0的通道1	—	—	—
使用的引脚	RxD0	—	—	—
中断	INTSR0	—	—	—
	只限于传送结束中断(禁止设置缓冲器空中断)。			
错误中断	INTSRE0	—	—	—
错误检测标志	<ul style="list-style-type: none"> • 帧错误检测标志(FEF01) • 溢出错误检测标志(OVF01) 			
传送数据长度	8位			
传送速率 ^注	Max.F _{MCK} /6[bps](SDR01[15:9]≥2)、Min.F _{CLK} /(2×2 ¹⁵ ×128)[bps]			
数据相位	正相输出(默认值:高电平)。反相输出(默认值:低电平)。			
奇偶校验位	无奇偶校验位(不进行奇偶校验)。			
停止位	附加1位。			
数据方向	LSB优先			

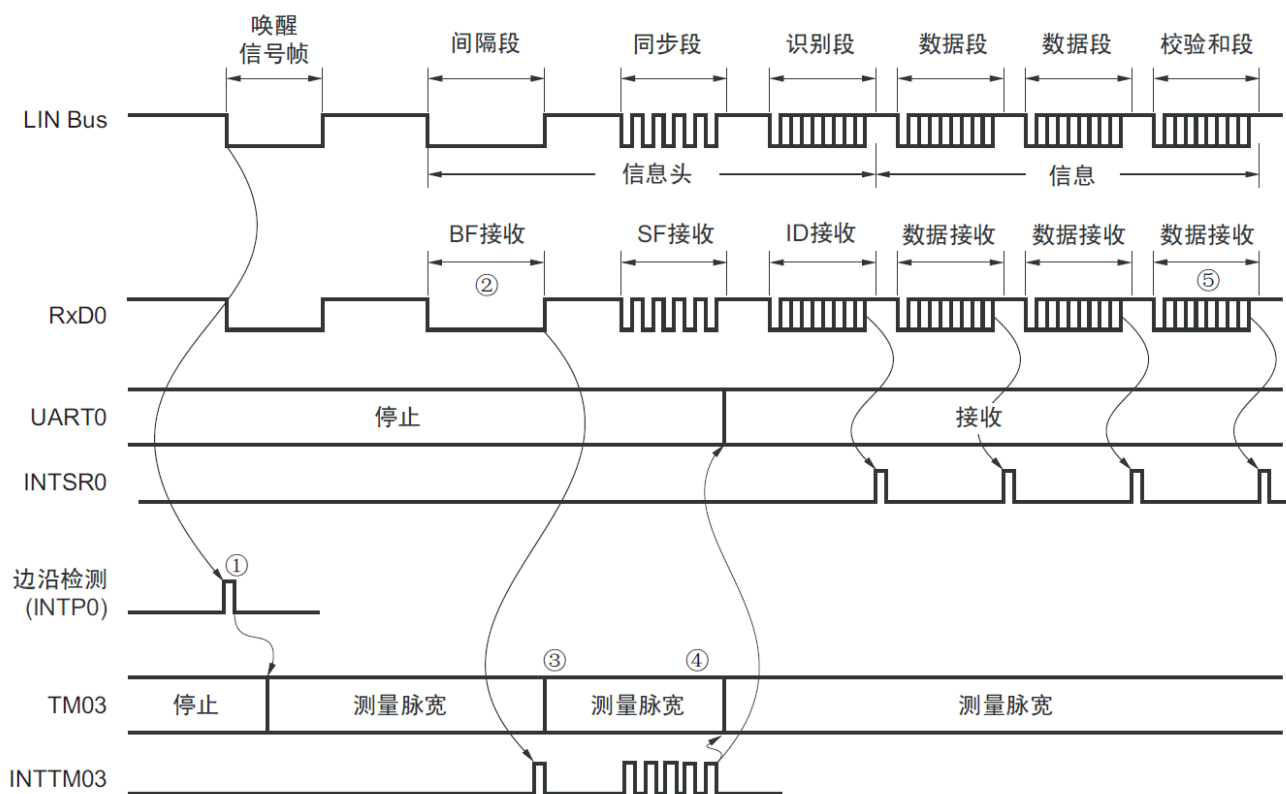
注：必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注：F_{MCK}：对象通道的运行时钟频率

F_{CLK}：系统时钟频率

LIN接收操作的概要如图19-114所示。

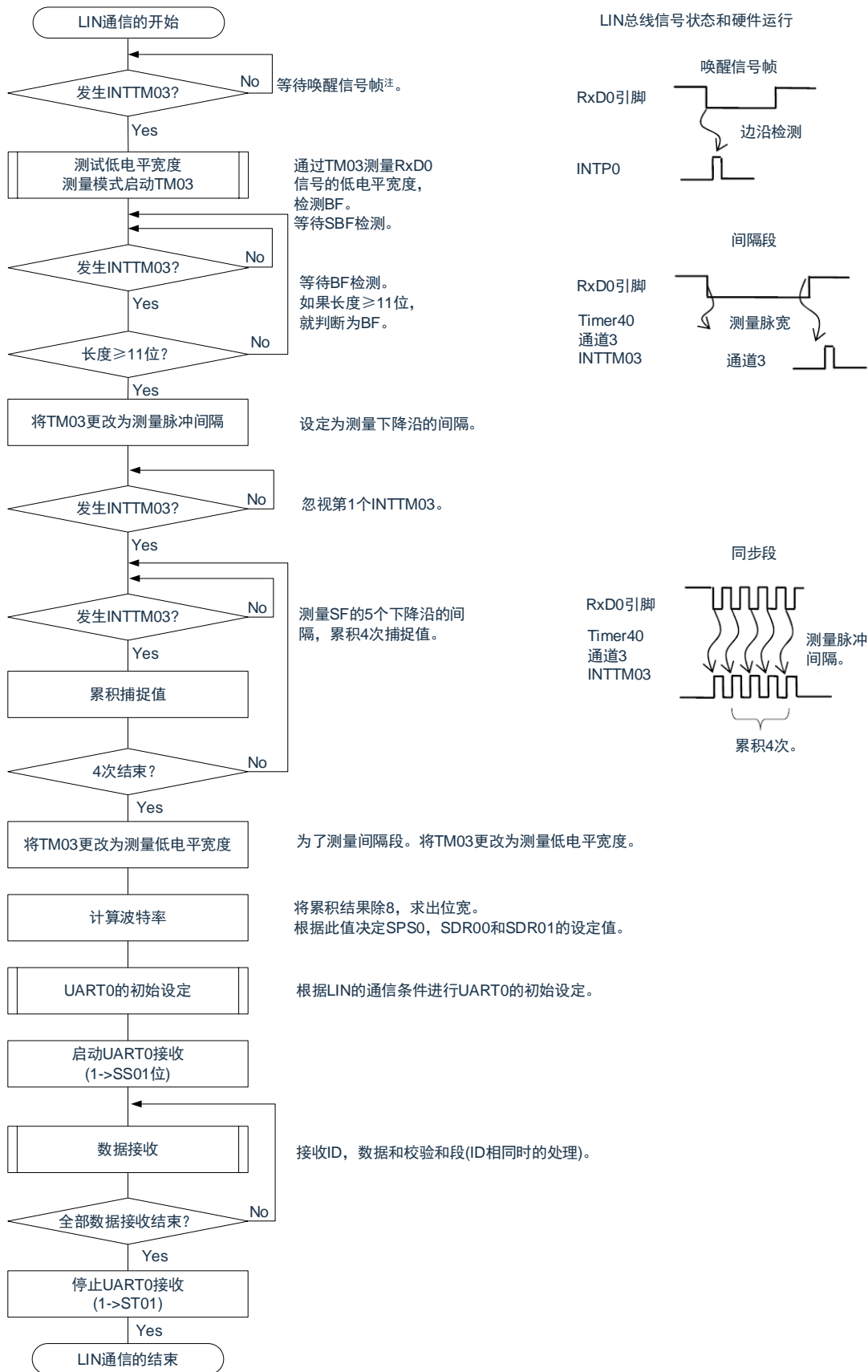
图19-114: LIN接收操作



信号处理的流程如下：

- ① 通过检测引脚的中断边沿(INTP0)来检测唤醒信号。当检测到唤醒信号时，为了测量BF的低电平宽度，将TM03设置为测量脉宽，然后进入BF接收等待状态。
- ② 如果检测到BF的下降沿，TM03就开始测量低电平宽度，并且在BF的上升沿进行捕捉。根据捕捉到的值判断是否为BF信号。
- ③ 当BF接收正常结束时，必须将TM03设置为测量脉冲间隔，并且测量4次同步段的RxD0信号下降沿的间隔(参照“6.8.4 作为输入脉冲间隔测量的运行”)。
- ④ 根据同步段(SF)的位间隔计算波特率误差。然后，必须在暂停UART0运行后调整(重新设置)波特率。
- ⑤ 必须通过软件区分校验和段。还必须通过软件在接收校验和段后对UART0进行初始化并且再次设置为BF接收等待状态。

图19-115: LIN接收的流程图



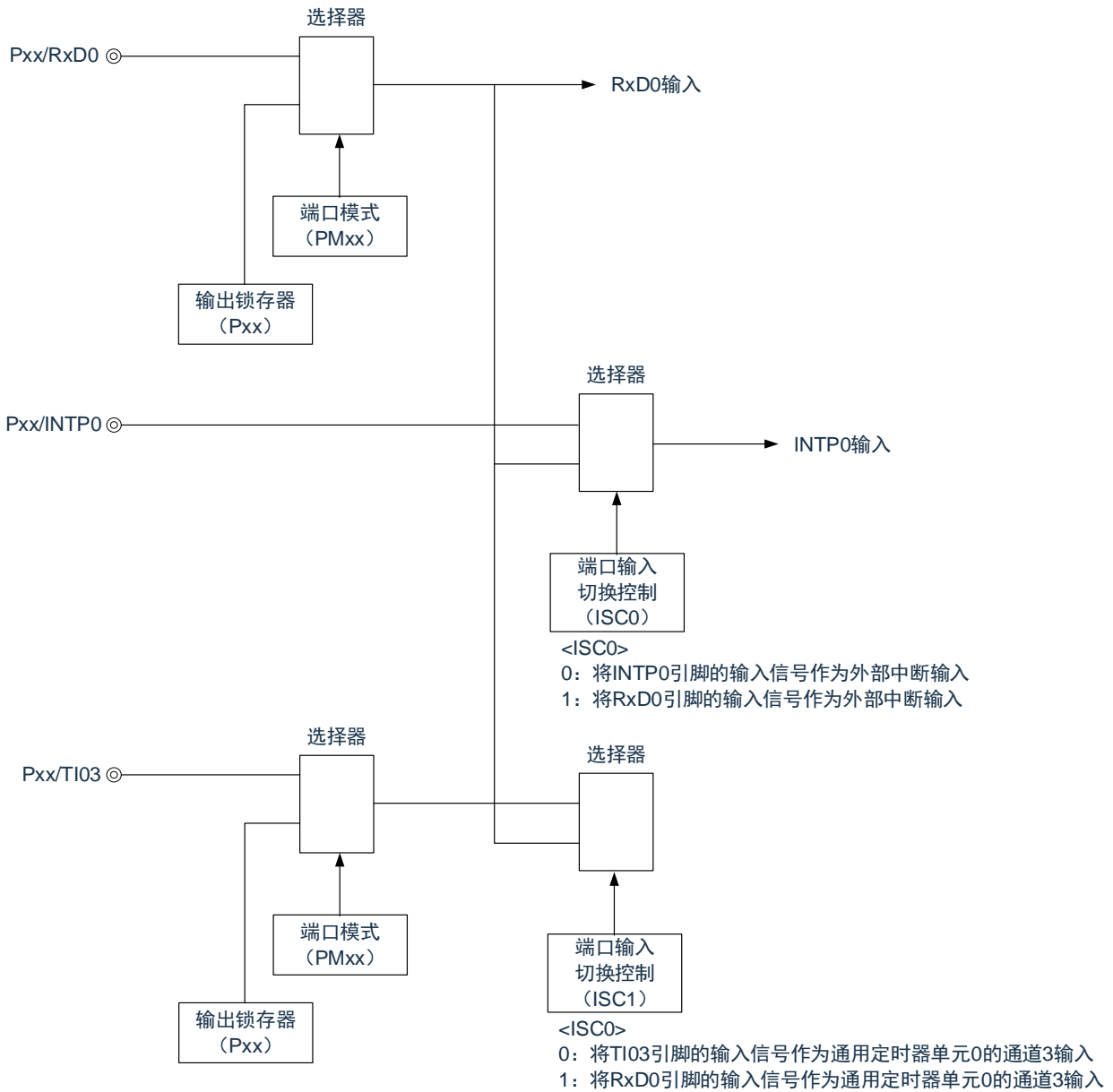
注：只在睡眠状态下才需要。

用于LIN接收操作的端口结构图如图19-116所示。

通过外部中断(INTP0)的边沿检测,接收LIN主控发送的唤醒信号。能通过通用定时器单元的外部事件捕捉运行,测量LIN主控发送的同步段的长度以及计算波特率误差。

通过端口输入切换控制(ISC0/ISC1),能不在外部连线而将用于接收的端口输入(RxD0)的输入源输入到外部中断(INTP0)和定时器阵列单元。

图19-116: 用于LIN接收操作的端口结构图



备注: ISC0、ISC1: 输入切换控制寄存器(ISC)的bit0和bit1(参照图19-19)

用于LIN通信运行的外围功能总结如下：

<使用的外围功能>

- 外部中断(INTPO)：唤醒信号的检测
用途：检测唤醒信号的边沿和通信的开始。
- 通用定时器单元的通道3：波特率误差的检测、间隔段(BF)的检测
用途：检测同步段(SF)的长度，并且通过将其长度除以位数来检测波特率误差(通过捕捉模式测量RxD0输入边沿的间隔)。测量低电平宽度，判断是否为间隔段(BF)。
- 通用串行通信单元0(SCIO)的通道0和通道1(UART0)

19.9 简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信的运行

这是通过串行时钟(SCL)和串行数据(SDA)共2条线与多个设备进行时钟同步通信的功能。因为此简易I²C是为了与EEPROM、闪存、A/D转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守AC规格，在操作控制寄存器的同时通过软件进行处理。

[数据的发送和接收]

- 主控发送、主控接收(只限于单主控的主控功能)
- ACK输出功能^注、ACK检测功能
- 8位数据长度(在发送地址时，用高7位指定地址，用最低位进行R/W控制)
- 通过软件产生开始条件和停止条件。

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK错误

※[简易I²C不支持的功能]

- 从属发送、从属接收
- 多主控功能(仲裁失败检测功能)
- 等待检测功能

注：在接收最后的数据时，如果给SDOEmn位(SDOEm寄存器)写“0”来停止串行通信数据的输出，就不输出ACK。详细内容请参照“19.9.3(2)处理流程”。

备注：m：单元号(m=0、1)n：通道号(n=0~3)mn=00~03、10~11

SCI0的通道0~3以及SCI1的通道0~1是支持简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)的通道。

简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21、IIC30、IIC31)有以下4种通信运行：

- 地址段发送 (参照19.9.1)
- 数据发送 (参照19.9.2)
- 数据接收 (参照19.9.3)
- 停止条件的产生 (参照19.9.4)

19.9.1 地址段发送

地址段发送是为了特别指定传送对象(从属设备)而在I²C通信时最先进行的发送运行。在产生开始条件后,将地址(7位)和传送方向(1位)作为1帧进行发送。

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCL00 SDA00 ^{注1}	SCL01 SDA01 ^{注1}	SCL10 SDA10 ^{注1}	SCL11 SDA11 ^{注1}	SCL20 SDA20 ^{注1}	SCL21 SDA21 ^{注1}
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断(不能选择缓冲器空中断)。					
错误检测标志	ACK错误检测标志(PEFmn)					
传送数据长度	8位(将高7位作为地址并且将低1位作为R/W控制进行发送)					
传送速率 ^{注2}	Max.F _{MCK} /4[Hz](SDRmn[15:9]≥1) F _{MCK} : 对象通道的运行时钟频率 但是,必须在I ² C的各模式中满足以下条件: •Max.1MHz(增强型快速模式) •Max.400KHz(快速模式) •Max.100KHz(标准模式)					
数据电平	正相输出(默认值:高电平)。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位(用于ACK接收)。					
数据方向	MSB优先					

注1:要通过简易I²C进行通信时,必须通过端口输出模式寄存器(POMxx)设置N沟道漏极开路输出模式(POMxx=1)。详细内容请参照“2.3 控制端口功能的寄存器”和“2.5 使用复用功能时的寄存器设置”。

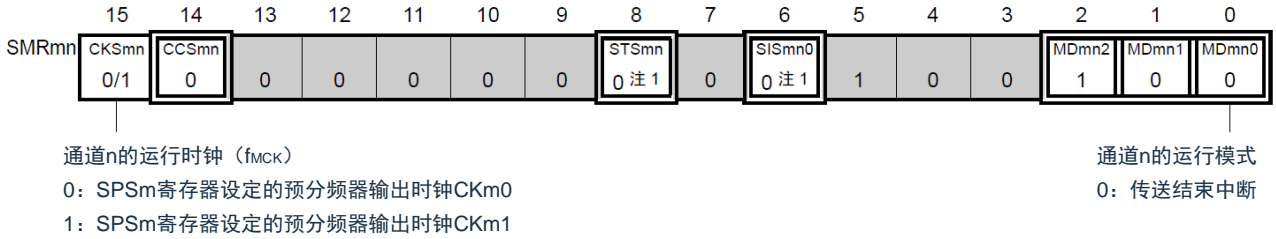
注2:必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

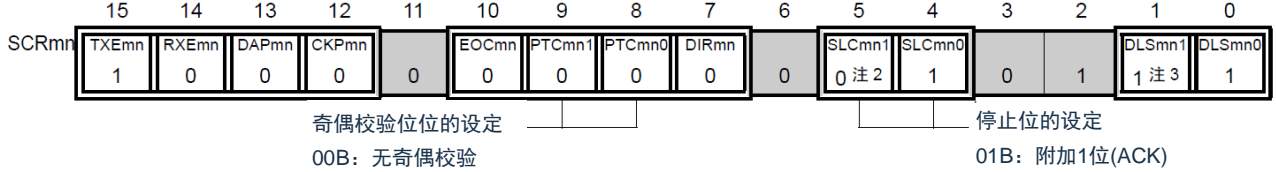
(1) 寄存器的设置

图19-117: 简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)地址段发送时的寄存器设置内容例子

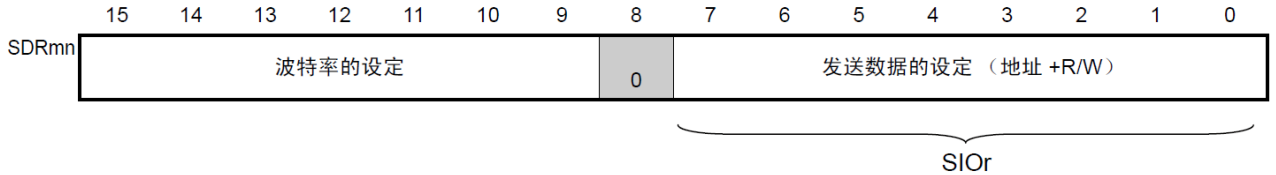
(a) 串行模式寄存器mn(SMRmn)



(b) 串行通信运行设定寄存器mn(SCRmn)



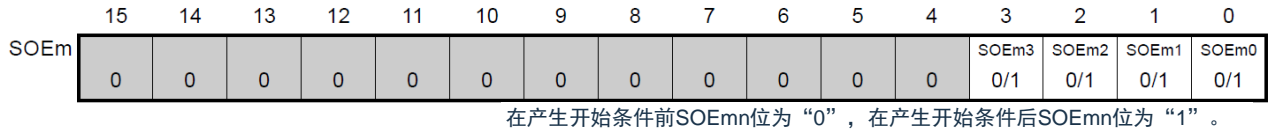
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOr)



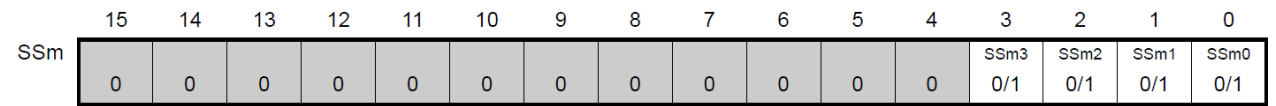
(d) 串行输出寄存器m(SOm)



(e) 串行输出允许寄存器m(SOEm)



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



- 注1: 只限于SMR00、SMR03、SMR11。
 注2: 只限于SCR00、SCR02、SCR10。
 注3: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。
 备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)r: IIC号(r=00、01、10、11、20、21)
 mn=00~03、10~11
- : 在IIC模式中为固定设置。■: 不能设置(设置初始值)。
 ×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

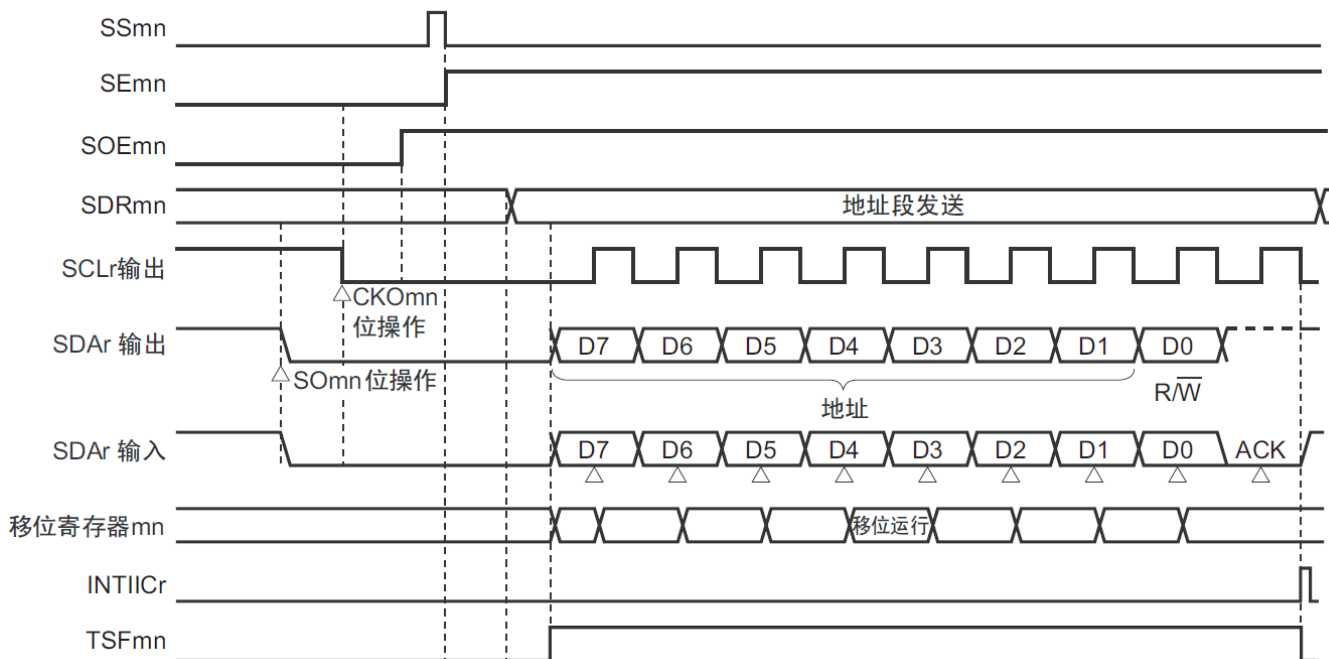
图19-118: 地址段发送的初始设置步骤



备注：在初始设置结束时，简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)为禁止输出并且处于运行停止状态。

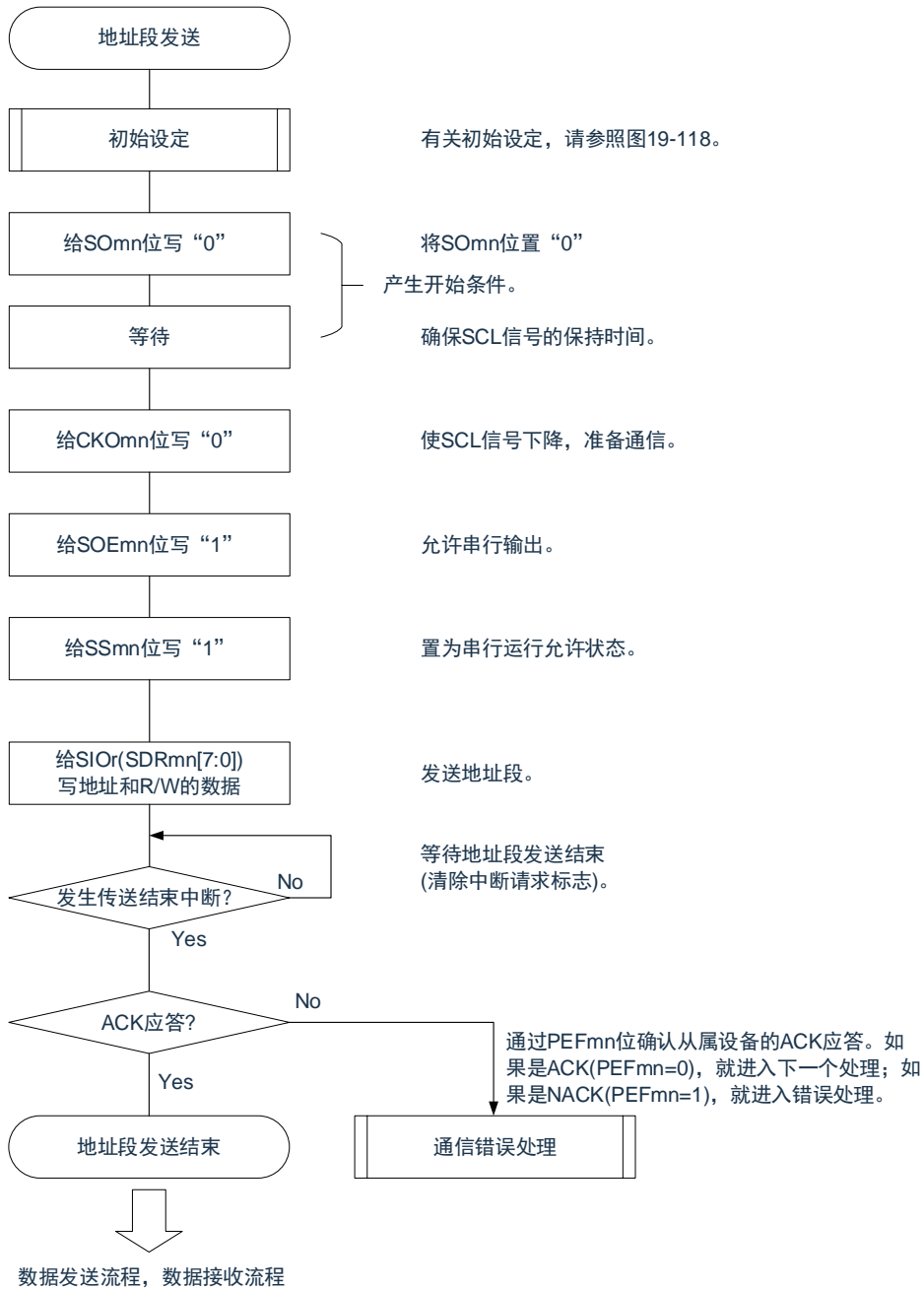
(3) 处理流程

图19-119: 地址段发送的时序图



备注: m: 单元号(m=0、1)n: 通道号(n=0~3)r: IIC号(r=00、01、10、11、20、21)
mn=00~03、10~11

图19-120: 地址段发送的流程图



19.9.2 数据发送

数据发送是在发送地址段后给该传送对象(从属设备)发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCL00 SDA00 ^{注1}	SCL01 SDA01 ^{注1}	SCL10 SDA10 ^{注1}	SCL11 SDA11 ^{注1}	SCL20 SDA20 ^{注1}	SCL21 SDA21 ^{注1}
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断(不能选择缓冲器空中断)。					
错误检测标志	ACK错误标志(PEFmn)					
传送数据长度	8位					
传送速率 ^{注2}	Max.F _{MCK} /4[Hz](SDRmn[15:9]≥1) F _{MCK} : 对象通道的运行时钟频率 但是, 必须在I ² C的各模式中满足以下条件: •Max.1MHz(增强型快速模式) •Max.400KHz(快速模式) •Max.100KHz(标准模式)					
数据电平	正相输出(默认值: 高电平)。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位(用于ACK接收)。					
数据方向	MSB优先					

注1: 要通过简易I²C进行通信时, 必须通过端口输出模式寄存器(POMxx)设置N沟道漏极开路输出模式(POMxx=1)。详细内容请参照“2.3 控制端口功能的寄存器”和“2.5 使用复用功能时的寄存器设置”。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

(1) 寄存器的设置

图19-121: 简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)数据发送时的寄存器设置内容例子的寄存器设置内容例子

(a) 串行模式寄存器mn(SMRmn).....在数据发送和接收的过程中不操作此寄存器。

SMRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0注1	0	0注1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器mn(SCRmn).....在数据发送和接收的过程中不操作TXEmn位和RXEmn位以外的位。

SCRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0注2	1	0	1	1注3	1

(c) 串行数据寄存器mn(SDRmn) (低8位: SIO_r)在数据发送和接收的过程中只有低8位有效(SIO_r)。

SDRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	波特率的设定注4								0	发送数据的设定							
	SIO _r																

(d) 串行输出寄存器m(SOm)在数据发送和接收的过程中不操作此寄存器。

SOm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5

(e) 串行输出允许寄存器m(SOEm)在数据发送和接收的过程中不操作此寄存器。

SOEm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

(f) 串行通道开始寄存器m(SSm)在数据发送和接收的过程中不操作此寄存器。

SSm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

注1: 只限于SMR01、SMR03、SMR11寄存器。

注2: 只限于SCR00、SCR02、SCR10寄存器。

注3: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

注4: 因为已经在发送地址段时设置, 所以不需要设置。

注5: 在通信运行过程中, 值因通信数据而变。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)r: IIC号(r=00、01、10、11、20、21)
mn=00~03、10~11
- ☐: 在IIC模式中为固定设置 ■ : 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 处理流程

图19-122: 数据发送的时序图

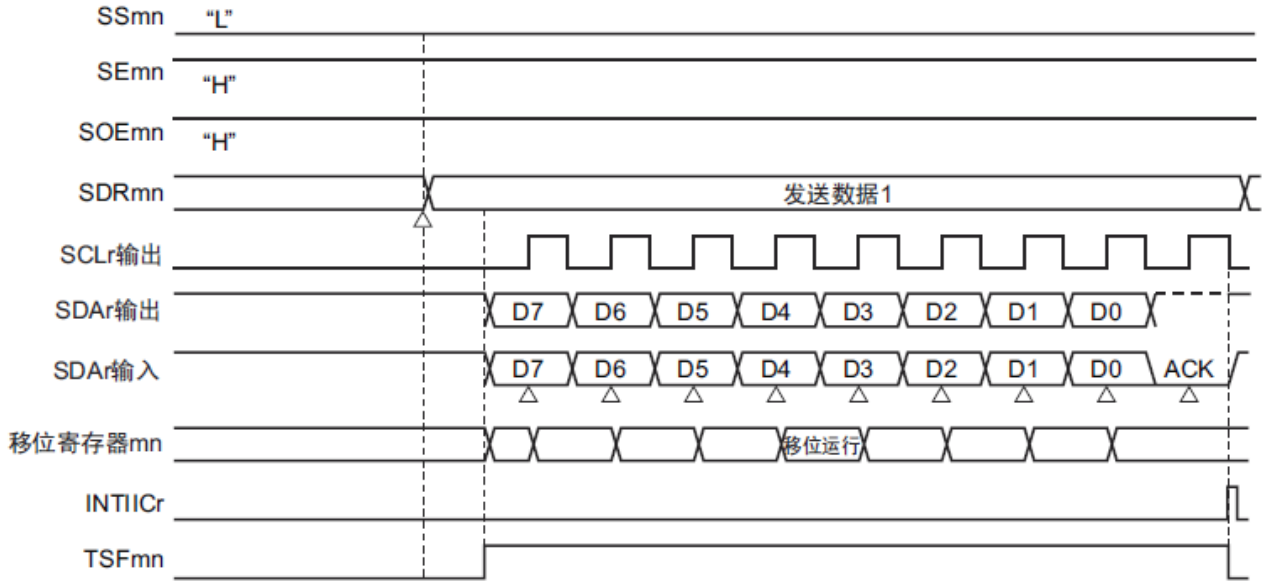
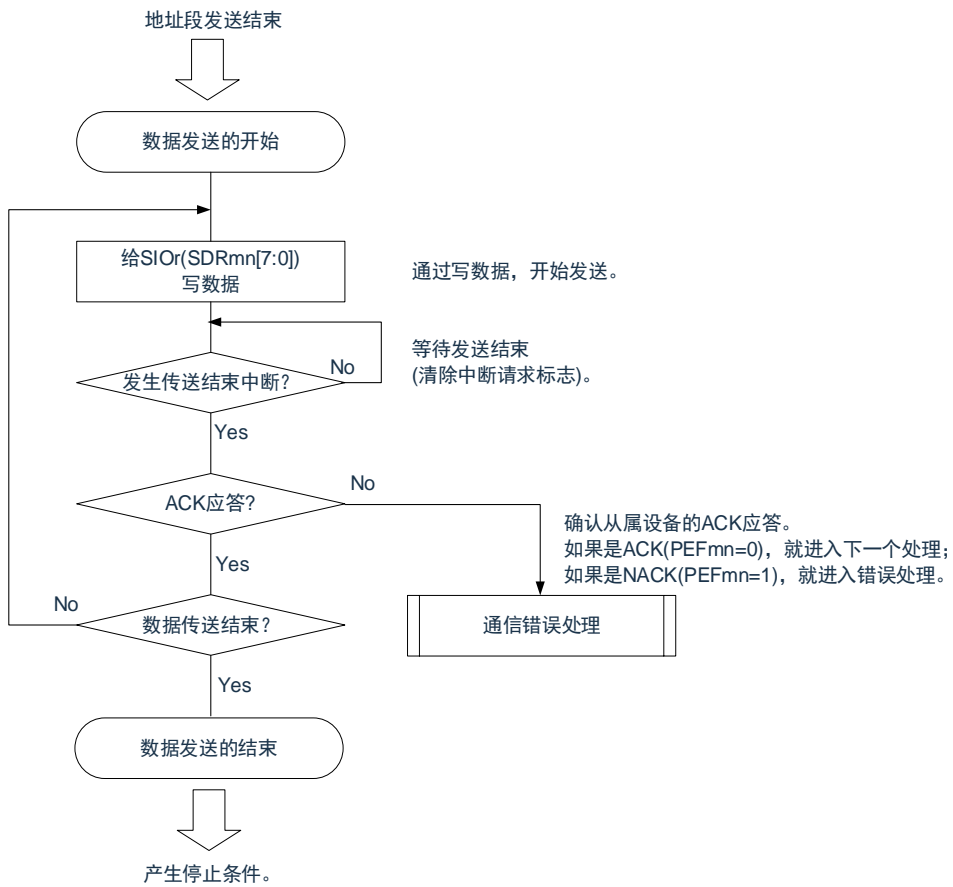


图19-123: 数据发送的流程图



19.9.3 数据接收

数据接收是在发送地址段后从传送对象(从属设备)接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCL00 SDA00 ^{注1}	SCL01 SDA01 ^{注1}	SCL10 SDA10 ^{注1}	SCL11 SDA11 ^{注1}	SCL20 SDA20 ^{注1}	SCL21 SDA21 ^{注1}
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断(不能选择缓冲器空中断)。					
错误检测标志	只有溢出错误检测标志(OVFmn)。					
传送数据长度	8位					
传送速率注2	Max.F _{MCK} /4[Hz](SDRmn[15:9]≥1) F _{MCK} : 对象通道的运行时钟频率 但是, 必须在I ² C的各模式中满足以下条件: •Max.1MHz(增强型快速模式) •Max.400KHz(快速模式) •Max.100KHz(标准模式)					
数据电平	正相输出(默认值: 高电平)。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位(ACK发送)。					
数据方向	MSB优先					

注1: 要通过简易I²C进行通信时, 必须通过端口输出模式寄存器(POMxx)设置N沟道漏极开路输出模式(POMxx=1)。详细内容请参照“2.3 控制端口功能的寄存器”和“2.5 使用复用功能时的寄存器设置”。

注2: 必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注: m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

(1) 寄存器的设置

图19-124: 简易I2C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)数据接收时的寄存器设置内容例子的寄存器设置内容例子

(a) 串行模式寄存器mn(SMRmn).....在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0注1	0	0注1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器mn(SCRmn).....在数据发送和接收的过程中不操作TXEmn位和RXEmn位以外的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0注2	1	0	1	1注3	1

(c) 串行数据寄存器mn(SDRmn) (低8位: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDRmn	波特率的设定注4								0	虚拟发送数据的设定 (FFH)							
	SIOr																

(d) 串行输出寄存器m(SOm)在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5

(e) 串行输出允许寄存器m(SOEm)在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(f) 串行通道开始寄存器m(SSm)在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

注1: 只限于SMR01、SMR03、SMR11寄存器。

注2: 只限于SCR00、SCR02、SCR10寄存器。

注3: 只限于SCR00寄存器和SCR01寄存器, 其他固定为“1”。

注4: 因为已经在发送地址段时设置, 所以不需要设置。

注5: 在通信运行过程中, 值因通信数据而变。

备注:

- m: 单元号(m=0、1)n: 通道号(n=0~3)r: IIC号(r=00、01、10、11、20、21)
mn=00~03、10~11

- : 在IIC模式中为固定设置。 : 不能设置(设置初始值)。

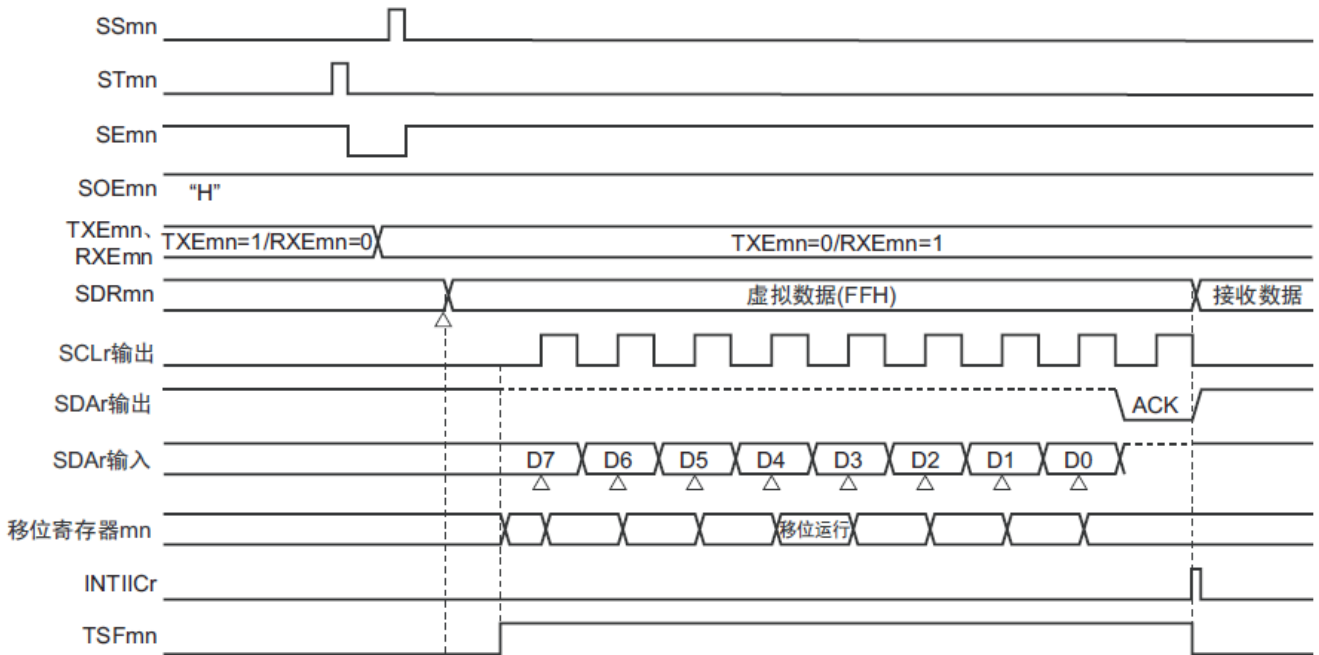
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下, 设置初始值)。

0/1: 根据用户的用途置“0”或者“1”。

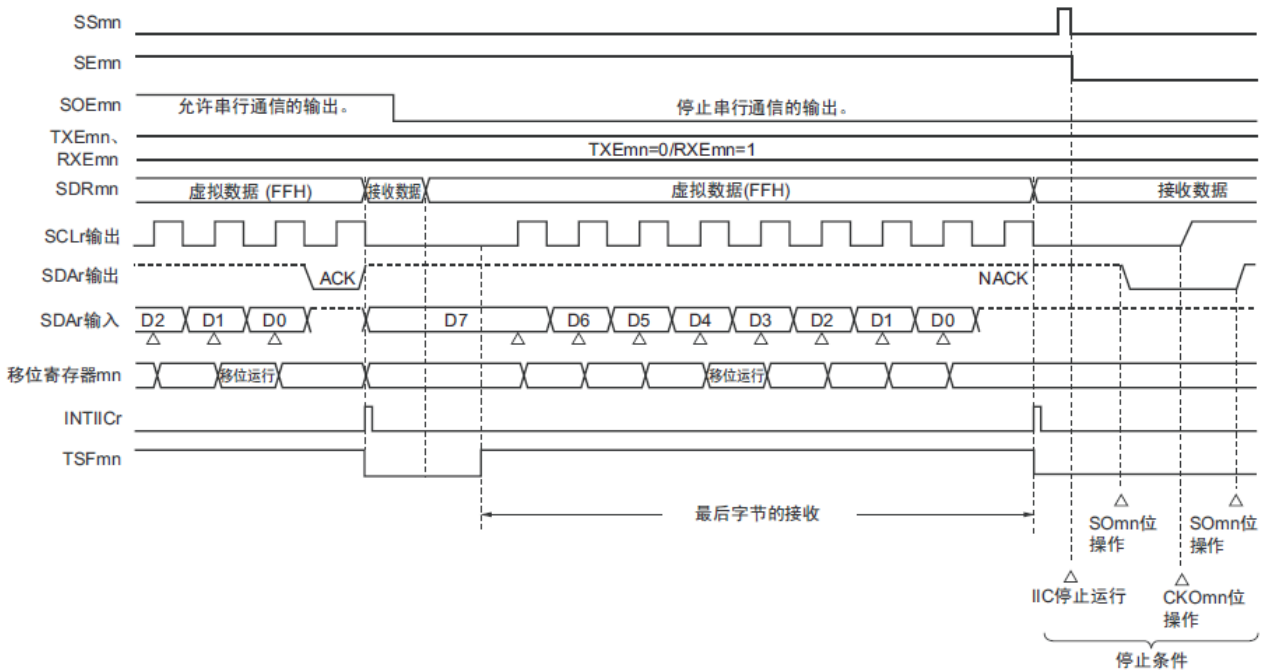
(2) 处理流程

图19-125: 数据接收的时序图

(a) 开始接收数据的情况

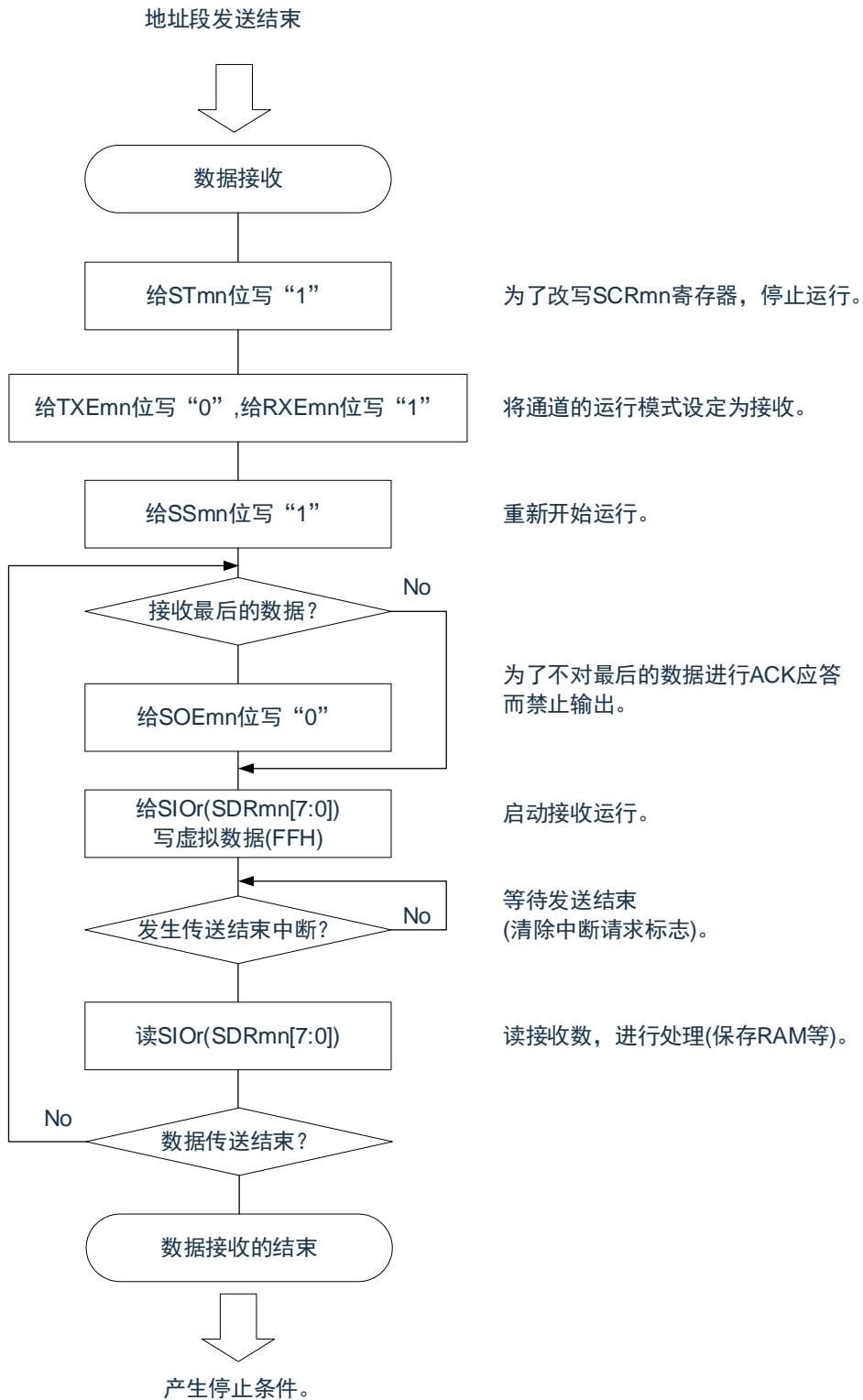


(b) 接收最后数据的情况



备注: m: 单元号(m=0、1)n: 通道号(n=0~3)r: IIC号(r=00、01、10、11、20、21)
mn=00~03、10~11

图19-126: 数据接收的流程图



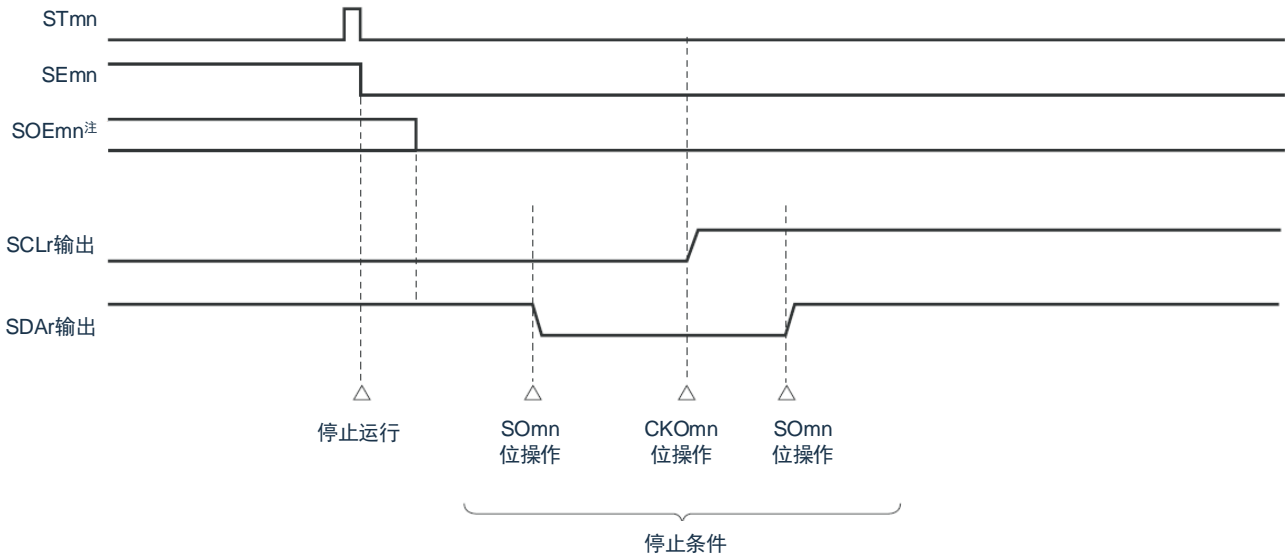
注意：在接收最后的数据时不输出ACK(NACK)。此后，先通过将串行通道停止寄存器m(STm)的STmn位置“1”来停止运行，然后产生停止条件来结束通信。

19.9.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。

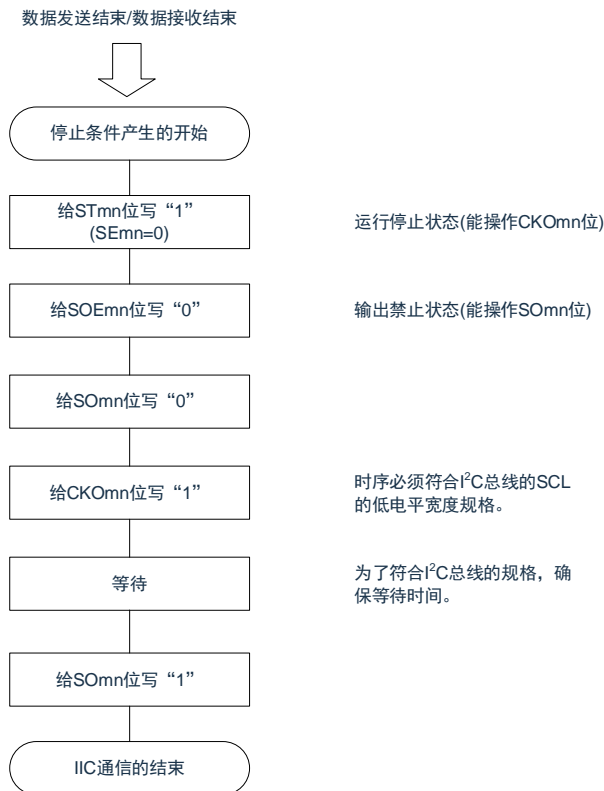
(1) 处理流程

图19-127：产生停止条件的时序图



注：在接收时，已经在接收最后的数据前将串行输出允许寄存器m(SOEm)的SOEmn位置“0”。

图19-128：产生停止条件的流程图



19.9.5 传送速率的计算

简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信的传送速率能用以下计算式进行计算。

$$\text{(传送速率)} = \{\text{对象通道的运行时钟 (F}_{\text{MCK}}\text{) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

注意：禁止将SDRmn[15:9]设置为“0000000B”，SDRmn[15:9]的设置值必须大于等于“0000001B”。简易I²C输出的SCL信号的占空比为50%。在I²C总线规格中，SCL信号的低电平宽度大于高电平宽度。因此，如果设置为快速模式的400kbps或者增强型快速模式的1Mbps，SCL信号输出的低电平宽度就小于I²C总线的规格值。必须给SDRmn[15:9]设置能满足I²C总线规格的值。

备注：

1. 因为SDRmn[15:9]的值为串行数据寄存器(SDRmn)的bit15~9的值(0000001B~1111111B)，所以为1~127。
2. m：单元号(m=0、1)n：通道号(n=0~3)mn=00~03、10~11

运行时钟(F_{MCK})取决于串行时钟选择寄存器m(SPSm)和串行模式寄存器mn(SMRmn)的bit15(CKSmn位)。

表19-5: 简易I²C运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟(F _{MCK}) ^注		
	CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00	F _{CLK} =32MHz运行时	
0	X	X	X	X	0	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	0	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	0	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	1	F _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	1	0	F _{CLK} /2 ⁶	500KHz
	X	X	X	X	0	1	1	1	1	F _{CLK} /2 ⁷	250KHz
	X	X	X	X	1	0	0	0	0	F _{CLK} /2 ⁸	125KHz
	X	X	X	X	1	0	0	1	1	F _{CLK} /2 ⁹	62.5KHz
	X	X	X	X	1	0	1	0	0	F _{CLK} /2 ¹⁰	31.25KHz
	X	X	X	X	1	0	1	1	1	F _{CLK} /2 ¹¹	15.63KHz
1	0	0	0	0	X	X	X	X	X	F _{CLK}	32MHz
	0	0	0	1	X	X	X	X	X	F _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	X	F _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	X	F _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	X	F _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	X	F _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	X	F _{CLK} /2 ⁶	500KHz
	0	1	1	1	X	X	X	X	X	F _{CLK} /2 ⁷	250KHz
	1	0	0	0	X	X	X	X	X	F _{CLK} /2 ⁸	125KHz
	1	0	0	1	X	X	X	X	X	F _{CLK} /2 ⁹	62.5KHz
	1	0	1	0	X	X	X	X	X	F _{CLK} /2 ¹⁰	31.25KHz
	1	0	1	1	X	X	X	X	X	F _{CLK} /2 ¹¹	15.63KHz
上述以外									禁止设置。		

注：要更改被选择为F_{CLK}的时钟(更改系统时钟控制寄存器(CKC)的值)时，必须在停止通用串行通信单元(SCI)的运行(串行通道停止寄存器m(STm)=000FH)后进行更改。

备注：

1. X: 忽略
2. m: 单元号(m=0、1)n: 通道号(n=0~3)mn=00~03、10~11

F_{MCK}=F_{CLK}=32MHz时的I²C传送速率的设置例子如下所示。

I ² C传送模式 (期待的传送速率)	F _{CLK} =32MHz			
	运行时钟(F _{MCK})	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100KHz	F _{CLK} /2	79	100KHz	0.0%
400KHz	F _{CLK}	41	380KHz	5.0% ^注
1MHz	F _{CLK}	18	0.84MHz	16.0% ^注

注：因为SCL信号的占空比为50%，所以不能将误差设置为“0”%左右。

19.9.6 在简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信过程中发生错误时的处理步骤

在简易I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)通信过程中发生错误时的处理步骤如图19-129和图19-130所示。

图19-129：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SDIRmn) 写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

图19-130：在简易I²C模式中发生ACK错误时的处理步骤

软件操作	硬件状态	备注
读串行状态寄存器mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn (SDIRmn)。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m (STm)的STmn位置“1”。	串行通道允许状态寄存器m (SEm)的SEmn位为“0”并且通道n为运行停止状态。	因为没有返回 ACK，所以从属设备没有做好接收的准备。因此，生成停止条件并且释放总线，再次从开始条件开始通信，或者也能生成重新开始条件并且从地址发送开始重新进行。
生成停止条件。		
生成开始条件。		
将串行通道开始寄存器m (SSm)的SSmn位置“1”。	串行通道允许状态寄存器m (SEm)的SEmn位为“1”并且通道n为可运行状态。	

备注：m：单元号(m=0、1)n：通道号(n=0~3)r：IIC号(r=00、01、10、11、20、21)
mn=00~03、10~11

第20章 串行接口IICA

20.1 串行接口IICA的功能

串行接口IICA有以下3种模式。

(1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

(2) I²C总线模式(支持多主控)

此模式通过串行时钟(SCLAn)和串行数据总线(SDAAn)的2条线，与多个设备进行8位数据传送。符合I²C总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的I²C总线控制部分。

因为串行接口IICA的SCLAn引脚和SDAAn引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

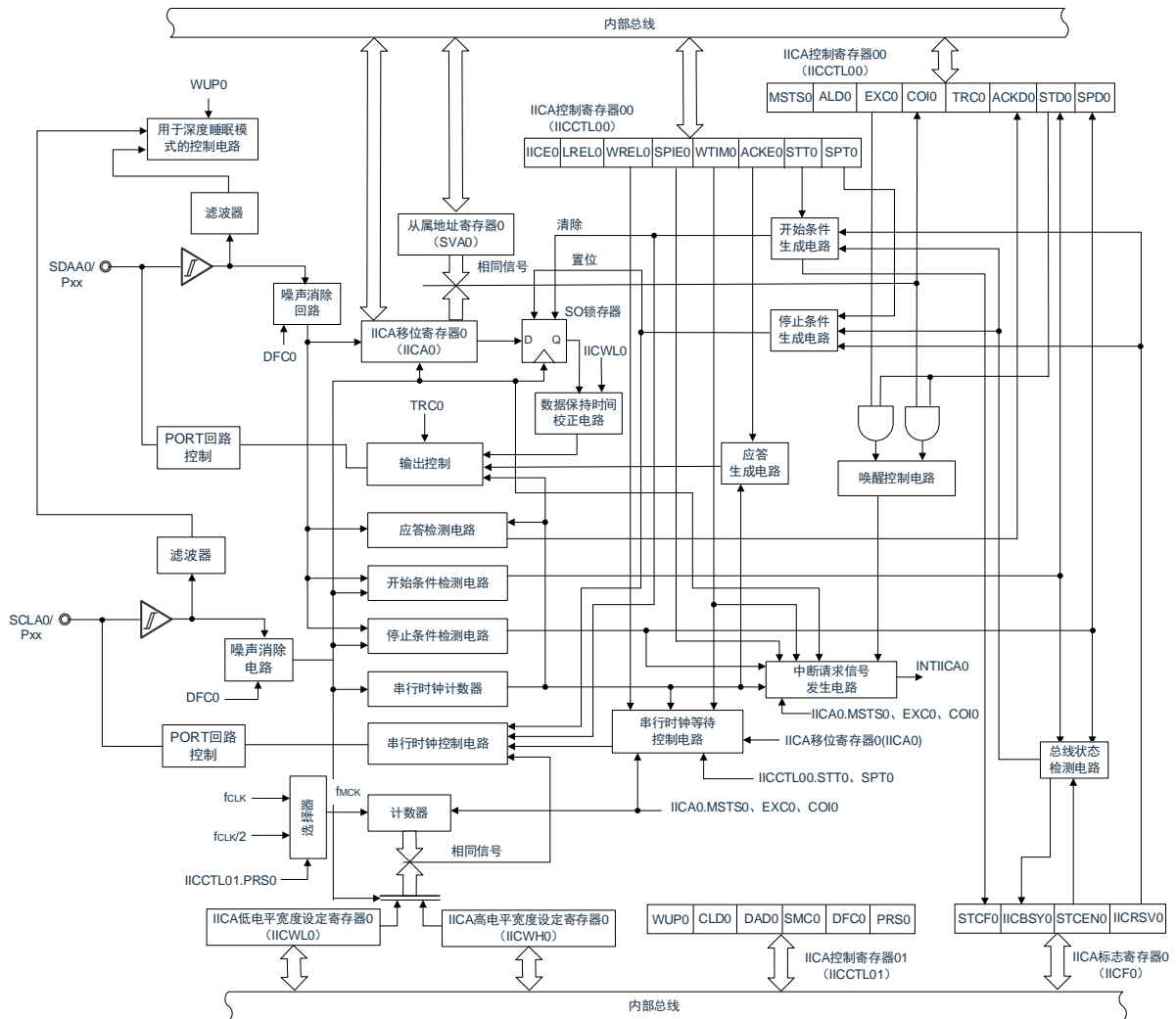
(3) 唤醒模式

在深度睡眠模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号(INTIICAn)解除深度睡眠模式。通过IICA控制寄存器n1(IICCTLn1)的WUPn位进行设置。

串行接口IICA的框图如图20-1所示。

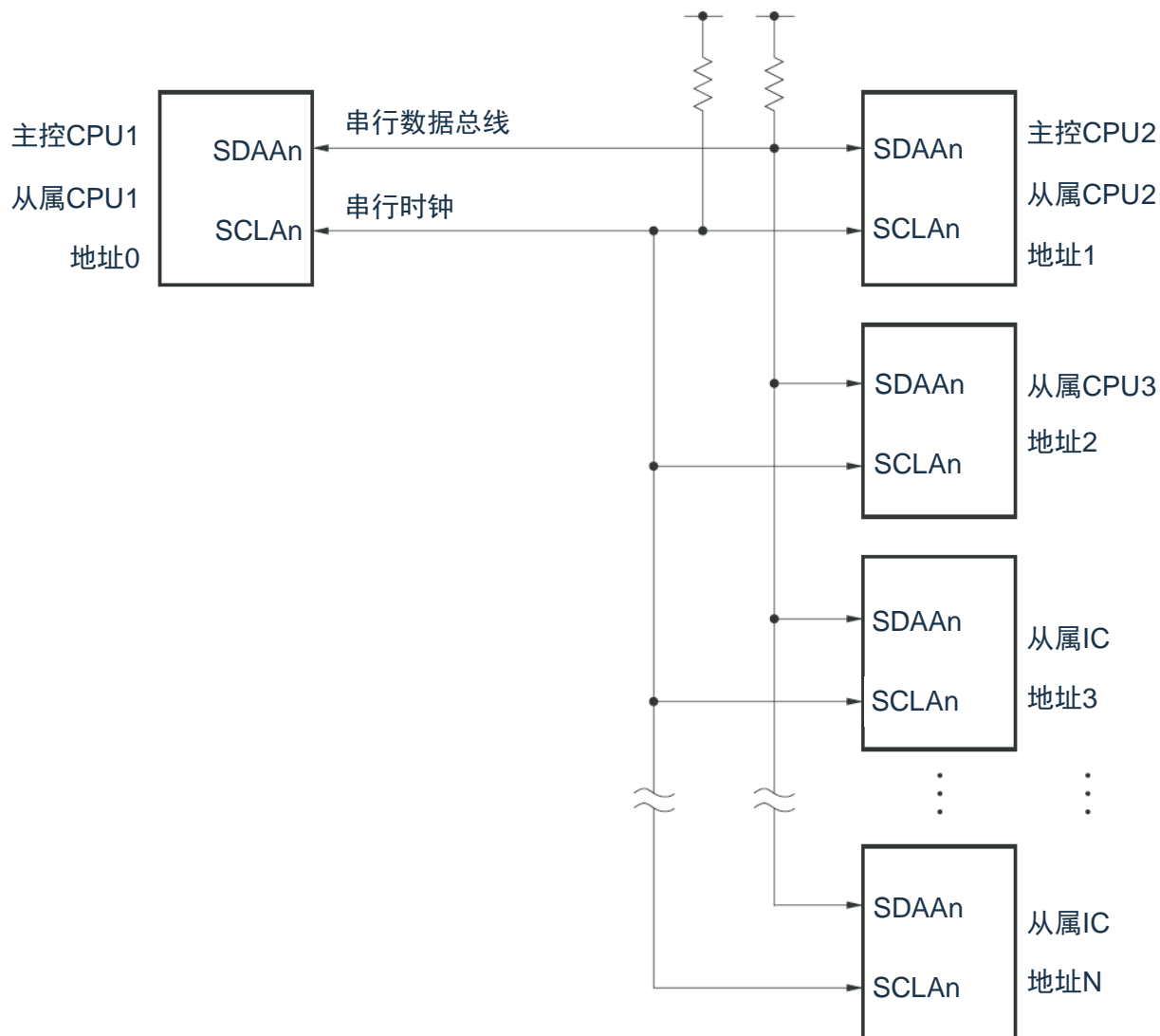
备注：n=0

图20-1：串行接口IICA的框图



串行总线的结构例子如图20-2所示。

图20-2: I²C总线的串行总线结构例子



备注: n=0

20.2 串行接口IICA的结构

串行接口IICA由以下硬件构成。

表20-1: 串行接口IICA的结构

项目	结构
寄存器	IICA移位寄存器n(IICAn)从属地址寄存器n(SVAn)
控制寄存器	外围允许寄存器0(PER0) IICA控制寄存器n0(IICCTLn0) IICA状态寄存器n(IICSn) IICA标志寄存器n(IICFn) IICA控制寄存器n1(IICCTLn1) IICA低电平宽度设置寄存器n(IICWLn) IICA高电平宽度设置寄存器n(IICWHn) 端口模式寄存器x(PMx) 端口寄存器x(Px)

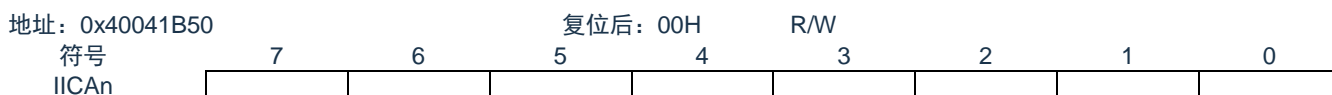
备注: n=0

20.2.1 IICA移位寄存器n(IICAn)

IICAn寄存器是与串行时钟同步进行8位串行数据和8位并行数据相互转换的寄存器，用于发送和接收。能通过读写IICAn寄存器来控制实际的发送和接收。

在等待期间，通过写IICAn寄存器来解除等待，开始传送数据。通过8位存储器操作指令设置IICAn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图20-3: IICAn移位寄存器n(IICAn)的格式



注意:

1. 在数据传送过程中，不能给IICAn寄存器写数据。
2. 只能在等待期间读写IICAn寄存器。除了等待期间以外，禁止在通信状态下存取IICAn寄存器。但是，在主控设备的情况下，能在将通信触发位(STTn)置“1”后写一次IICAn寄存器。
3. 当预约通信时，必须在检测到由停止条件产生的中断后给IICAn寄存器写数据。

备注: n=0

20.2.2 从属地址寄存器n(SVAn)

这是在用作从属设备时保存7位本地站地址{A6,A5,A4,A3,A2,A1,A0}的寄存器。

通过8位存储器操作指令设置SVAn寄存器。但是，在STDn位为“1”(检测到开始条件)时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图20-4：从属地址寄存器n(SVAn)的格式

地址：0x40041A34	复位后：00H			R/W				
符号	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 ^注

注：bit0固定为“0”。

20.2.3 SO锁存器

SO锁存器保持SDAAn引脚的输出电平。

20.2.4 唤醒控制电路

当设置在从属地址寄存器n(SVAn)的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求(INTIICAn)。

20.2.5 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了8位数据的发送和接收。

20.2.6 中断请求信号发生电路

此电路控制产生中断请求信号(INTIICAn)。

由以下2种触发产生I²C中断请求。

- 第8个或者第9个串行时钟的下降(通过WTIMn位进行设置)
- 因检测到停止条件而产生中断请求(通过SPIEn位进行设置)。

备注：WTIMn位：IICA控制寄存器n(IICCTLn0)的bit3

SPIEn位：IICA控制寄存器n(IICCTLn0)的bit4

20.2.7 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到SCLAn引脚的时钟。

20.2.8 串行时钟等待控制电路

此电路控制等待时序。

20.2.9 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

20.2.10 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

20.2.11 开始条件生成电路

如果将STTn位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下(IICRSVn位=1)并且没有释放总线(IICBSYn位=1)时，忽视开始条件请求并且将STCFn位置“1”。

20.2.12 停止条件生成电路

如果将SPTn位置“1”，此电路就生成停止条件。

20.2.13 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。但是，在刚运行时不能立即检测总线状态，因此必须通过STCENn位设置总线状态检测电路的初始状态。

备注：

1. STTn位：IICA控制寄存器n0(IICCTLn0)的bit1
SPTn位：IICA控制寄存器n0(IICCTLn0)的bit0
IICRSVn位：IICA标志寄存器n(IICFn)的bit0
IICBSYn位：IICA标志寄存器n(IICFn)的bit6
STCFn位：IICA标志寄存器n(IICFn)的bit7
STCENn位：IICA标志寄存器n(IICFn)的bit1
2. n=0

20.3 控制串行接口IICA的寄存器

通过以下8种寄存器控制串行接口IICA。

- 外围允许寄存器0(PER0)
- IICA控制寄存器n0(IICCTLn0)
- IICA标志寄存器n(IICFn)
- IICA状态寄存器n(IICSn)
- IICA控制寄存器n1(IICCTLn1)
- IICA低电平宽度设置寄存器n(IICWLn)
- IICA高电平宽度设置寄存器n(IICWHn)
- 端口模式寄存器(PMx)
- 端口寄存器(Px)

备注：n=0

20.3.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。

通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用串行接口IICAn时，必须将bit4(IICA0EN)置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图20-5：外围允许寄存器0(PER0)的格式

地址：0x40020420 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN ^注	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN

IICAnEN	提供串行接口IICAn的输入时钟的控制
0	停止提供输入时钟。 •不能写串行接口IICAn使用的SFR。 •串行接口IICAn处于复位状态。
1	允许提供输入时钟。 •能读写串行接口IICAn使用的SFR。

注意：要设置串行接口IICA时，必须先在IICAnEN位为“1”的状态下设置以下的寄存器。当IICAnEN位为“0”时，串行接口IICA的控制寄存器的值为初始值，忽视写操作(端口模式寄存器(PMx)和端口寄存器(Px)除外)。

- IICA控制寄存器n0(IICCTLn0)
- IICA标志寄存器n(IICFn)
- IICA状态寄存器n(IICSn)
- IICA控制寄存器n1(IICCTLn1)
- IICA低电平宽度设置寄存器n(IICWLn)
- IICA高电平宽度设置寄存器n(IICWHn)

备注：n=0

20.3.2 IICA控制寄存器n0(IICCTLn0)

这是允许或者停止I²C运行、设置等待时序以及设置其他I²C运行的寄存器。

通过8位存储器操作指令设置IICCTLn0寄存器。但是，必须在IICEn位为“0”时或者在等待期间设置SPIEn位、WTIMn位和ACKEn位，而且在将IICEn位从“0”置为“1”时能同时设置这些位。

在产生复位信号后，此寄存器的值变为“00H”。

备注：n=0

图20-6: IICA控制寄存器n0(IICCTLn0)的格式(1/4)

地址: 0x40041A30

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I ² C运行的允许
0	停止运行。对IICA状态寄存器n(IICSn)进行复位 ^{注1} , 并且停止内部运行。
1	允许运行。
必须在SCLAn线和SDAAn线为高电平的状态下将此位置“1”。	
清除条件(IICEn=0)	置位条件(IICEn=1)
•通过指令清除。 •当复位时	•通过指令置位。

LRELn ^{注2,3}	通信的退出
0	通常运行
1	退出当前的通信, 进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况下使用。 SCLAn线和SDAAn线变为高阻状态。 IICA控制寄存器n0(IICCTLn0)和IICA状态寄存器n(IICSn)中的以下标志被清“0”: •STTn•SPTn•MSTSn•EXCn•COIn•TRCn•ACKDn•STDn
变为退出通信的待机状态, 保持到满足以下的通信参加条件为止。 •在检测到停止条件后作为主控设备启动。 •在检测到开始条件后地址匹配或者接收到扩展码。	
清除条件(LRELn=0)	置位条件(LRELn=1)
•在执行后自动清除。 •当复位时	•通过指令置位。

WRELn ^{注2,3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下(TRCn=1)的第9个时钟等待期间将WRELn位(解除等待)置位, SDAAn线就变为高阻抗状态(TRCn=0)。	
清除条件(WRELn=0)	置位条件(WRELn=1)
•在执行后自动清除。 •当复位时	•通过指令置位。

注1: 对IICA移位寄存器n(IICAn)、IICA标志寄存器n(IICFn)的STCFn位和IICBSYn位以及IICA控制寄存器n1(IICCTLn1)的CLDn位和DADn位进行复位。

注2: 在IICEn位为“0”的状态下, 此位的信号无效。

注3: LRELn位和WRELn位的读取值总是“0”。

注意: 如果在SCLAn线为高电平、SDAAn线为低电平并且数字滤波器为ON(IICCTLn1寄存器的DFCn=1)时允许I²C运行(IICEn=1), 就立即检测开始条件。此时, 必须在允许I²C运行(IICEn=1)后连续通过位存储器操作指令将LRELn位置“1”。

备注: n=0

图20-6: IICA控制寄存器n0(IICCTLn0)的格式(2/4)

SPIEn ^{注1}	允许或者禁止停止条件检测产生的中断请求	
0	禁止	
1	允许	
当IICA控制寄存器n1(IICCTLn1)的WUPn位为“1”时, 即使将SPIEn位置“1”也不产生停止条件中断。		
清除条件(SPIEn=0)		置位条件(SPIEn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

WTIMn ^{注1}	等待和中断请求的控制	
0	在第8个时钟的下降沿产生中断请求信号。 主控设备: 在输出8个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入8个时钟后, 将时钟置为低电平, 然后等待主控设备。	
1	在第9个时钟的下降沿产生中断请求信号。 主控设备: 在输出9个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入9个时钟后, 将时钟置为低电平, 然后等待主控设备。	
在地址传送期间, 与此位的设置无关, 在第9个时钟的下降沿产生中断; 在地址传送结束后, 此位的设置有效。主控设备在地址传送期间的第9个时钟下降沿进入等待状态。接收到本地站地址的从属设备在产生应答(ACK)后的第9个时钟下降沿进入等待状态, 但是接收到扩展码的从属设备在第8个时钟下降沿进入等待状态。		
清除条件(WTIMn=0)		置位条件(WTIMn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

ACKEn ^{注1,2}	应答控制	
0	禁止应答。	
1	允许应答。在第9个时钟期间将SDAAn线置为低电平。	
清除条件(ACKEn=0)		置位条件(ACKEn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

注1: 在IICEn位为“0”的状态下, 此位的信号无效。必须在此期间设置此位。

注2: 在地址传送过程中不是扩展码时, 设置值无效。当为从属设备并且地址匹配时, 与设置值无关而生成应答。

备注: n=0

图20-6: IICA控制寄存器n0(IICCTLn0)的格式(3/4)

STTn ^{注1,2}	开始条件的触发	
0	不生成开始条件。	
1	当总线被释放时(待机状态, IICBSYn位为“0”): 如果将此位置“1”, 就生成开始条件(作为主控设备的启动)。当第三方正在通信时: <ul style="list-style-type: none"> •允许通信预约功能的情况(IICRSVn=0) 用作开始条件预约标志。如果将此位置“1”, 就在释放总线后自动生成开始条件。 •禁止通信预约功能的情况(IICRSVn=1) 即使将此位置“1”, 也清除STTn位并且将STTn清除标志(STCFn)置“1”, 不生成开始条件。等待状态(主控设备): 在解除等待后生成重新开始条件。 	
有关置位时序的注意事项: <ul style="list-style-type: none"> •主控接收: 禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 •主控发送: 在应答期间, 可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。 •禁止与停止条件的触发(SPTn)同时置“1”。 •在将STTn位置“1”后, 禁止在满足清除条件前再次将此位“1”。 		
清除条件(STTn=0)		置位条件(STTn=1)
<ul style="list-style-type: none"> •在禁止通信预约的状态下将STTn位置“1”。 •在仲裁失败时 •主控设备生成开始条件。 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位为“0”(停止运行时) •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

注1: 在IICEn位为“0”的状态下, 此位的信号无效。

注2: STTn位的读取值总是“0”。

备注:

1. 如果在设置数据后读bit1(STTn), 此位就变为“0”。
2. IICRSVn: IICA标志寄存器n(IICFn)的bit0
 STCFn: IICA标志寄存器n(IICFn)的bit7
3. n=0

图20-6: IICA控制寄存器n0(IICCTLn0)的格式(4/4)

SPTn ^注	停止条件的触发	
0	不生成停止条件。	
1	生成停止条件(作为主控设备的传送结束)。	
有关置位时序的注意事项: <ul style="list-style-type: none"> •主控接收: 禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 •主控发送: 在应答期间, 可能无法正常生成停止条件。必须在输出第9个时钟后的等待期间将此位置“1”。 •禁止与开始条件的触发(STTn)同时置“1”。 •只有在主控设备的情况下才能将SPTn位置“1”。 •在WTIMn位为“0”时, 必须注意: 如果在输出8个时钟后的等待期间将SPTn位置“1”, 就在解除等待后的第9个时钟的高电平期间生成停止条件。必须在输出8个时钟后的等待期间将WTIMn位从“0”置为“1”并且在输出第9个时钟后的等待期间将SPTn位置“1”。 •在将SPTn位置“1”后, 禁止在满足清除条件前再次将此位置“1”。 		
清除条件(SPTn=0)		置位条件(SPTn=1)
<ul style="list-style-type: none"> •当仲裁失败时 •在检测到停止条件后自动清除。 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位为“0”(停止运行)时 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

注: SPTn位的读取值总是“0”。

注意: 在IICA状态寄存器n(IICSn)的bit3(TRCn)为“1”(发送状态)时, 如果在第9个时钟将IICCTLn0寄存器的bit5(WRELn)置“1”来解除等待, 就在清除TRCn位(接收状态)后将SDAAn线置为高阻抗。必须通过写IICA移位寄存器n进行TRCn位为“1”(发送状态)时的等待解除。

备注: n=0

20.3.3 IICA状态寄存器n(IICSn)

这是表示I²C状态的寄存器。

只有在STTn位为“1”并且等待期间，才能8位存储器操作指令读IICSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

注意：在深度睡眠模式中允许地址匹配唤醒功能(WUPn=1)状态下，禁止读IICSn寄存器。在WUPn位为“1”的状态下，与INTIICAn中断请求无关，如果将WUPn位从“1”改为“0”(停止唤醒运行)，就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许(SPIEn=1)因检测到停止条件而产生的中断，并且在检测到中断后读IICSn寄存器。

备注：STTn：IICA控制寄存器n0 (IICCTLn0) 的bit1

WUPn：IICA控制寄存器n1 (IICCTLn1) 的bit7

图20-7：IICA状态寄存器n(IICSn)的格式(1/3)

地址：0x40041B51

复位后：00H R

符号	7	6	5	4	3	2	1	0
IICSn	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n

MSTS _n	主控状态的确认标志
0	从属状态或者通信待机状态
1	主控通信状态
清除条件(MSTS _n =0)	
<ul style="list-style-type: none"> 当检测到停止条件时 当ALD_n位为“1”(仲裁失败)时 因LREL_n位为“1”(退出通信)而进行的清除 当IICEn位从“1”变为“0”(停止运行)时 当复位时 	
置位条件(MSTS _n =1)	
<ul style="list-style-type: none"> 当生成开始条件时 	

ALD _n	仲裁失败的检测
0	表示未发生仲裁或者赢得仲裁。
1	表示仲裁失败。清除MSTS _n 位。
清除条件(ALD _n =0)	
<ul style="list-style-type: none"> 在读IICSn寄存器后自动清除注。 当IICEn位从“1”变为“0”(停止运行)时 当复位时 	
置位条件(ALD _n =1)	
<ul style="list-style-type: none"> 当仲裁失败时 	

注：即使对IICSn寄存器以外的位执行位存储器操作指令，也清除此位。因此，在使用ALD_n位时，必须在读其他位前先读ALD_n位的数据。

备注：

1. LREL_n：IICA控制寄存器n0(IICCTLn0)的bit6
IICEn：IICA控制寄存器n0(IICCTLn0)的bit7
2. n=0

图20-7: IICA状态寄存器n(IICSn)的格式(2/3)

EXCn	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件(EXCn=0)		置位条件(EXCn=1)
<ul style="list-style-type: none"> •当检测到开始条件时 •当检测到停止条件时 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位从“1”变为“0”(停止运行)时 •当复位时 		<ul style="list-style-type: none"> •当接收的地址数据的高4位为“0000”或者“1111”时(在第8个时钟的上升沿置位)

COIn	地址匹配的检测	
0	地址不同。	
1	地址相同。	
清除条件(COIn=0)		置位条件(COIn=1)
<ul style="list-style-type: none"> •当检测到开始条件时 •当检测到停止条件时 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位从“1”变为“0”(停止运行)时 •当复位时 		<ul style="list-style-type: none"> •当接收地址和本地站地址(从属地址寄存器n(SVAn))相同时(在第8个时钟的上升沿置位)

TRCn	发送/接收的状态检测	
0	处于接收状态(发送状态除外)。将SDAAn线置为高阻抗。	
1	处于发送状态。设置为能将SON锁存器的值输出到SDAAn线(在第1字节的第9个时钟的下降沿以后有效)。	
清除条件(TRCn=0)		置位条件(TRCn=1)
<主控设备和从属设备> <ul style="list-style-type: none"> •当检测到停止条件时 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位从“1”变为“0”(停止运行)时 •因WRELn位为“1”(解除等待)而进行的清除注 •当ALDn位从“0”变为“1”(仲裁失败)时 •当复位时 •不参加通信的情况(MSTSn、EXCn、COIn=0) <主控设备> <ul style="list-style-type: none"> •当第1字节的LSB(传送方向指示位)输出“1”时 <从属设备> <ul style="list-style-type: none"> •当检测到开始条件时 •当第1字节的LSB(传送方向指示位)输入“0”时 		<主控设备> <ul style="list-style-type: none"> •当生成开始条件时 •当第1字节(地址传送)的LSB(传送方向指示位)输出“0”(主控发送)时 <从属设备> <ul style="list-style-type: none"> •当主控设备的第1字节(地址传送)的LSB(传送方向指示位)输入“1”(从属发送)时

注：在IICA状态寄存器n(IICSn)的bit3(TRCn)为“1”(发送状态)时，如果在第9个时钟将IICA控制寄存器n0(IICCTLn0)的bit5(WRELn)置“1”来解除等待，就在清除TRCn位(接收状态)后将SDAAn线置为高阻抗。必须通过写IICA移位寄存器n进行TRCn位为“1”(发送状态)时的等待解除。

备注：

1. LRELn: IICA控制寄存器n0(IICCTLn0)的bit6
IICEn: IICA控制寄存器n0(IICCTLn0)的bit7
2. n=0

图20-7: IICA状态寄存器n(IICSn)的格式(3/3)

ACKDn	应答(ACK)的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件(ACKDn=0)		置位条件(ACKDn=1)
<ul style="list-style-type: none"> •当检测到停止条件时 •当下一个字节的第1个时钟上升时 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位从“1”变为“0”(停止运行时) •当复位时 		<ul style="list-style-type: none"> •在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平

STDn	开始条件的检测	
0	未检测到开始条件。	
1	检测到开始条件，表示处于地址传送期间。	
清除条件(STDn=0)		置位条件(STDn=1)
<ul style="list-style-type: none"> •当检测到停止条件时 •在地址传送后的下一个字节的第1个时钟上升时 •因LRELn位为“1”(退出通信)而进行的清除 •当IICEn位从“1”变为“0”(停止运行时) •当复位时 		<ul style="list-style-type: none"> •当检测到开始条件时

SPDn	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件，主控设备结束通信并且已释放总线。	
清除条件(SPDn=0)		置位条件(SPDn=1)
<ul style="list-style-type: none"> •在将此位置位后，在检测到开始条件后的地址传送字节的第1个时钟上升时 •当WUPn位从“1”变为“0”时 •当IICEn位从“1”变为“0”(停止运行时) •当复位时 		<ul style="list-style-type: none"> •当检测到停止条件时

备注:

1. LRELn: IICA控制寄存器n0(IICCTLn0)的bit6
IICEn: IICA控制寄存器n0(IICCTLn0)的bit7
2. n=0

20.3.4 IICA标志寄存器n(IICFn)

这是设置I²C运行模式以及表示I²C总线状态的寄存器。

通过8位存储器操作指令设置IICFn寄存器。但是，只能读STTn清除标志(STCFn)和I²C总线状态标志(IICBSYn)。

通过IICRSVn位设置允许或者禁止通信预约功能，并且通过STCENn位设置IICBSYn位的初始值。只有在禁止I²C运行(IICA控制寄存器n0(IICCTLn0)的bit7(IICEn)=0)时才能写IICRSVn位和STCENn位。在允许运行后，只能读IICFn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图20-8: IICA标志寄存器n(IICFn)的格式

 地址: 0x40041B52 复位后: 00H R/W^注

符号	7	6	5	4	3	2	1	0
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn清除标志
0	发行开始条件。
1	无法发行开始条件而清除STTn标志。
清除条件(STCFn=0)	置位条件(STCFn=1)
<ul style="list-style-type: none"> •因STTn位为“1”而进行的清除 •当IICEn位为“0”(停止运行时) •当复位时 	<ul style="list-style-type: none"> •在设置为禁止通信预约(IICRSVn=1)的状态下无法发行开始条件而将STTn位清“0”时

IICBSYn	I2C总线状态标志
0	总线释放状态(STCENn=1时的通信初始状态)
1	总线通信状态(STCENn=0时的通信初始状态)
清除条件(IICBSYn=0)	置位条件(IICBSYn=1)
<ul style="list-style-type: none"> •当检测到停止条件时 •当IICEn位为“0”(停止运行时) •当复位时 	<ul style="list-style-type: none"> •当检测到开始条件时 •STCENn位为“0”时的IICEn位的置位

STCENn	初始开始允许触发
0	在允许运行(IICEn=1)后, 通过检测停止条件来允许生成开始条件。
1	在允许运行(IICEn=1)后, 不检测停止条件而允许生成开始条件。
清除条件(STCENn=0)	置位条件(STCENn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当检测到开始条件时 •当复位时 	<ul style="list-style-type: none"> •通过指令置位。

IICRSVn	通信预约功能禁止位
0	允许通信预约。
1	禁止通信预约。
清除条件(IICRSVn=0)	置位条件(IICRSVn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 	<ul style="list-style-type: none"> •通过指令置位。

注: bit6和bit7是只读位。

注意:

1. 只有在停止运行(IICEn=0)时才能写STCENn位。
2. 如果STCENn位为“1”, 就与实际的总线状态无关而认为总线为释放状态(IICBSYn=0), 因此为了避免在发行第1个开始条件(STTn=1)时破坏其他通信, 需要确认没有正在通信的第三方。
3. 只有在停止运行(IICEn=0)时才能写IICRSVn。

备注:

1. STTn: IICA控制寄存器n0(IICCTLn0)的bit1
2. IICEn: IICA控制寄存器n0(IICCTLn0)的bit7

20.3.5 IICA控制寄存器n1(IICCTLn1)

这是用于设置I²C运行模式以及检测SCLAn引脚和SDAAn引脚状态的寄存器。

通过8位存储器操作指令设置IICCTLn1寄存器。但是，只能读CLDn位和DADn位。

除了WUPn位以外，必须在禁止I²C运行(IICA控制寄存器n0(IICCTLn0)的bit7(IICEn)=0)时设置IICCTLn1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图20-9: IICA控制寄存器n1(IICCTLn1)的格式(1/2)

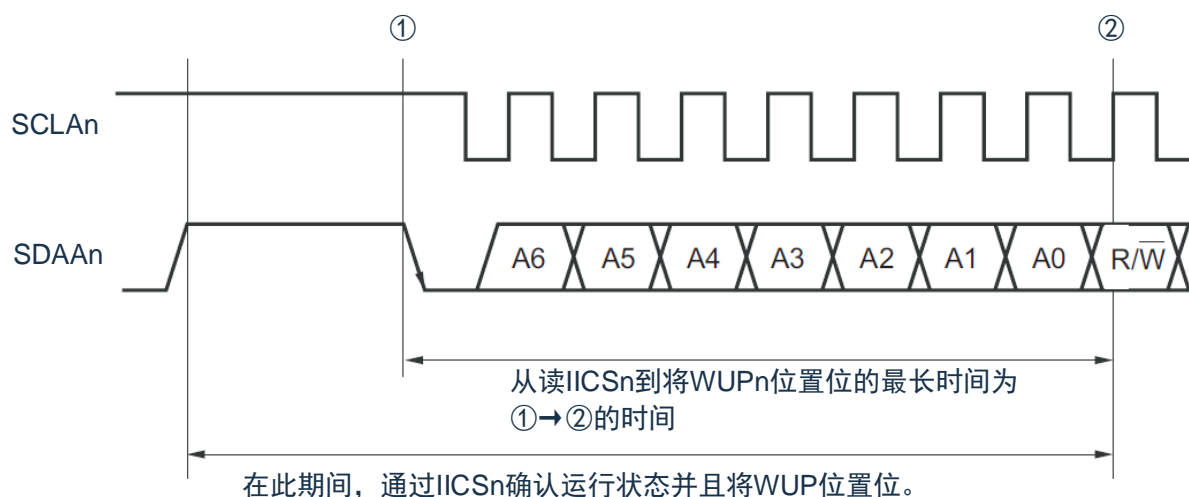
地址: 0x40041A31 复位后: 00H R/W^{注1}

符号	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	地址匹配唤醒的控制
0	在深度睡眠模式中，停止地址匹配唤醒功能的运行。
1	在深度睡眠模式中，允许地址匹配唤醒功能的运行。
<p>要通过将WUPn位置“1”来转移到深度睡眠模式时，必须在将WUPn位置“1”后至少经过3个F_{MCK}时钟，然后执行深度睡眠指令(参照“图20-21: 将WUPn位置“1”时的流程”)。在地址匹配或者接收到扩展码后，必须将WUPn位清“0”。能通过将WUPn位清“0”来参加后续的通信(需要在将WUPn位清“0”后解除等待以及写发送数据)。</p> <p>在WUPn位为“1”的状态下，地址匹配或者接收到扩展码时的中断时序与WUPn位为“0”时的中断时序相同(根据时钟产生采样误差的延迟差)。另外，当WUPn位为“1”时，即使将SPIEn位置“1”也不产生停止条件中断。</p>	
清除条件(WUPn=0)	置位条件(WUPn=1)
•通过指令清除(在地址匹配或者接收到扩展码后)。	•通过指令置位(MSTSn=0、EXCn=0、COIn=0并且STDn=0(不参加通信)) ^{注2} 。

注1: bit4和bit5是只读位。

注2: 在以下所示的期间，需要确认IICA状态寄存器n(IICSn)的状态并且将其置位。



备注: n=0

图20-9: IICA控制寄存器n1(IICCTLn1)的格式(2/2)

CLDn	SCLAn引脚的电平检测(只在IICEn位为“1”时有效)	
0	检测到SCLAn引脚为低电平。	
1	检测到SCLAn引脚为高电平。	
清除条件(CLDn=0)		置位条件(CLDn=1)
•当SCLAn引脚为低电平时 •当IICEn位为“0”(停止运行时) •当复位时		•当SCLAn引脚为高电平时

DADn	SDAAn引脚的电平检测(只在IICEn位为“1”时有效)	
0	检测到SDAAn引脚为低电平。	
1	检测到SDAAn引脚为高电平。	
清除条件(DADn=0)		置位条件(DADn=1)
•当SDAAn引脚为低电平时 •当IICEn位为“0”(停止运行时) •当复位时		•当SDAAn引脚为高电平时

SMCn	运行模式的切换
0	在标准模式中运行(最大传送速率: 100kbps)。
1	在快速模式(最大传送速率: 400kbps)或者增强型快速模式(最大传送速率: 1Mbps)中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器OFF
1	数字滤波器ON
必须在快速模式或者增强型快速模式中使用数字滤波器。 数字滤波器用于消除噪声。 无论是将DFCn位置“1”还是清“0”，传送时钟都不变。	

PRSn	运行时钟(F _{MCK})的控制
0	选择F _{CLK} (1MHz ≤ F _{CLK} ≤ 20MHz)。
1	选择F _{CLK} /2(20MHz < F _{CLK})。

注意:

- IICA运行时钟(F_{MCK})的最大工作频率为20MHz(Max.)。只有在F_{CLK}超过20MHz时才必须将IICA控制寄存器n1(IICCTLn1)的bit0(PRSn)置“1”。
- 在设置传送时钟的情况下，必须注意F_{CLK}的最小工作频率。串行接口IICA的F_{CLK}最小工作频率取决于运行模式。
 快速模式: F_{CLK}=3.5MHz(Min.)
 增强型快速模式: F_{CLK}=10MHz(Min.)
 标准模式: F_{CLK}=1MHz(Min.)

备注:

- IICEn: IICA控制寄存器n0(IICCTLn0)的bit7
- n=0

20.3.6 IICA低电平宽度设置寄存器n(IICWLn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号低电平宽度(TLOW)和SDAAn引脚信号。

通过8位存储器操作指令设置IICWLn寄存器。

必须在禁止I²C运行(IICA控制寄存器n0(IICCTLn0)的bit7(IICEn)=0)时设置IICWLn寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

有关IICWLn寄存器的设置方法，请参照“20.4.2 通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法”。

数据保持时间为IICWLn所设时间的1/4。

图20-10: IICA低电平宽度设置寄存器n(IICWLn)的格式



20.3.7 IICA高电平宽度设置寄存器n(IICWHn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号高电平宽度和SDAAn引脚信号。

通过8位存储器操作指令设置IICWHn寄存器。

必须在禁止I²C运行(IICA控制寄存器n0(IICCTLn0)的bit7(IICEn)=0)时设置IICWHn寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

图20-11: IICA高电平宽度设置寄存器n(IICWHn)的格式



备注:

1. 有关主控方传送时钟的设置方法，请参照20.4.2(1)；有关从属方IICWLn寄存器和IICWHn寄存器的设置方法，请参照20.4.2(2)。
2. n=0

20.3.8 端口模式寄存器x(PMx)

此寄存器设置端口的输入/输出。

在将Pxx/SCLA0引脚用作时钟输入/输出并且将Pxx/SDAA0引脚用作串行数据输入/输出时，必须将端口模式寄存器PMx和端口输出锁存器Px置“0”。

当IICEn位(IICA控制寄存器n0(IICCTLn0)的bit7)为“0”时，Pxx/SCLA0引脚和Pxx/SDAA0引脚为低电平输出(固定)，因此必须在将IICEn位置“1”后切换到输出模式。

通过8位存储器操作指令设置PMx寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

端口模式寄存器的详细内容请参考“2.3.1 端口模式寄存器(PMxx)”。

20.4 I²C总线模式的功能

20.4.1 引脚结构

串行时钟引脚(SCLAn)和串行数据总线引脚(SDAAn)的结构如下。

(1) SCLAn.....串行时钟的输入/输出引脚

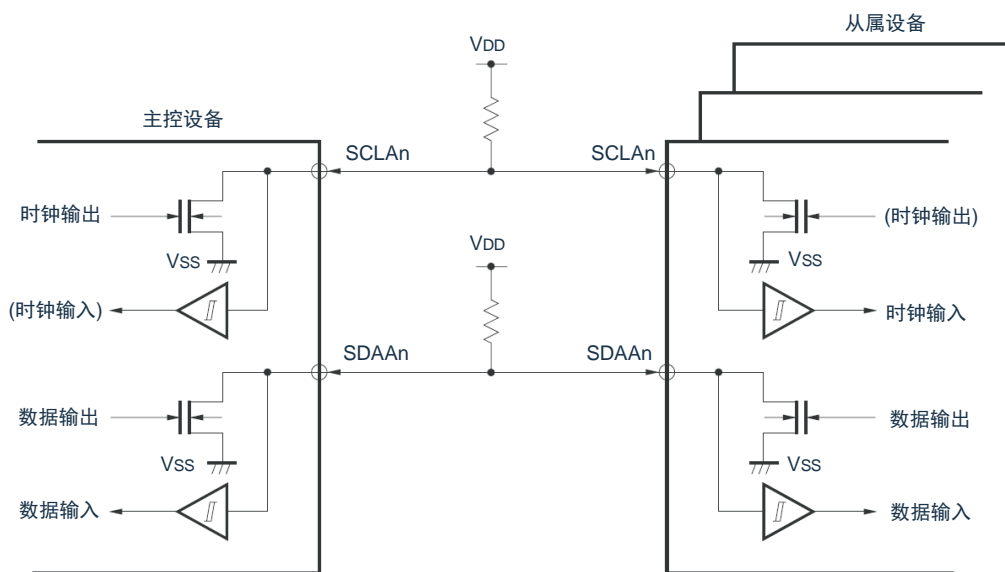
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

(2) SDAAn.....串行数据的输入/输出复用引脚

主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

因为串行时钟线和串行数据总线的输出为N沟道漏极开路输出，所以需要外接上拉电阻。

图20-12：引脚结构图



备注：n=0

20.4.2 通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法

(1) 主控方传送时钟的设置方法

$$\text{传送时钟} = \frac{F_{MCK}}{IICWL + IICWH + F_{MCK}(T_R + T_F)}$$

此时，IICWLn寄存器和IICWHn寄存器的最佳设置值如下：

(全部设置值的小数部分都舍入)

- 快速模式

$$IICWLn = \frac{0.52}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left(\frac{0.48}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

- 标准模式

$$IICWLn = \frac{0.47}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left(\frac{0.53}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

- 增强型快速模式

$$IICWLn = \frac{0.50}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left(\frac{0.50}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

(2) 从属方IICWLn寄存器和IICWHn寄存器的设置方法

(全部设置值的小数部分都舍入)

- 快速模式

$$IICWLn = 1.3\mu s \times F_{MCK}$$

$$IICWHn = (1.2\mu s - T_R - T_F) \times F_{MCK}$$

- 标准模式

$$IICWLn = 4.7\mu s \times F_{MCK}$$

$$IICWHn = (5.3\mu s - T_R - T_F) \times F_{MCK}$$

- 增强型快速模式

$$IICWLn = 0.50\mu s \times F_{MCK}$$

$$IICWHn = (0.50\mu s - T_R - T_F) \times F_{MCK}$$

注意：

- IICA运行时钟(F_{MCK})的最大工作频率为20MHz(Max.)。只有在 F_{CLK} 超过20MHz时才必须将IICA控制寄存器n1(IICCTLn1)的bit0(PRSn)置“1”。
- 在设置传送时钟的情况下，必须注意 F_{CLK} 的最小工作频率。串行接口IICA的 F_{CLK} 最小工作频率取决于运行模式。

快速模式： $F_{CLK} = 3.5\text{MHz}(\text{Min.})$

增强型快速模式： $F_{CLK} = 10\text{MHz}(\text{Min.})$

标准模式： $F_{CLK} = 1\text{MHz}(\text{Min.})$

备注:

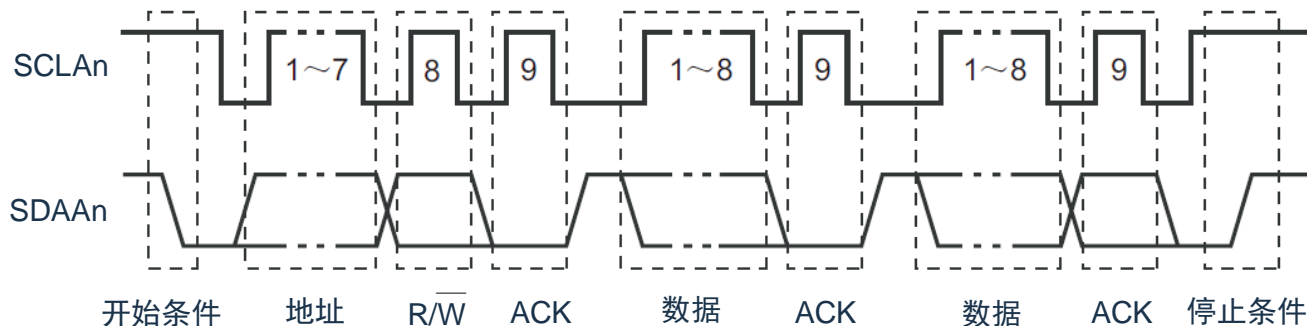
1. 因为SDAAn信号和SCLAn信号的上升时间(T_R)和下降时间(T_F)因上拉电阻和布线电容而不同, 所以必须各自计算。
2. IICWLn: IICA低电平宽度设置寄存器n
IICWHn: IICA高电平宽度设置寄存器n
 T_F : SDAAn信号和SCLAn信号的下降时间
 T_R : SDAAn信号和SCLAn信号的上升时间
 F_{MCK} : IICA运行时钟频率
3. $n=0$

20.5 I²C总线的定义和控制方法

以下说明I²C总线的串行数据通信格式和使用的信号。

I²C总线的串行数据总线上生成的“开始条件”、“地址”、“数据”和“停止条件”的各传送时序如图20-13所示。

图20-13: I²C总线的串行数据传送时序



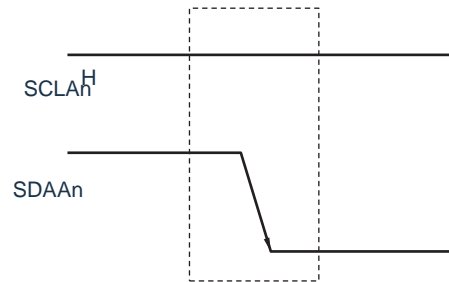
主控设备生成开始条件、从属地址和停止条件。

主控设备和从属设备都能生成应答(ACK)(在一般情况下,接收方输出8位数据)。主控设备连续输出串行时钟(SCLAn)。但是,从属设备能延长SCLAn引脚的低电平期间并且插入等待。

20.5.1 开始条件

在SCLAn引脚为高电平时，如果SDAAn引脚从高电平变为低电平，就生成开始条件。SCLAn引脚和SDAAn引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。在用作从属设备时，能检测到开始条件。

图20-14：开始条件



在检测到停止条件(SPDn: IICA状态寄存器n(IICSn)的bit0=1)的状态下，如果将IICA控制寄存器n0(IICCTLn0)的bit1(STTn)置“1”，就输出开始条件。如果检测到开始条件，就将IICSn寄存器的bit1(STDn)置“1”。

备注：n=0

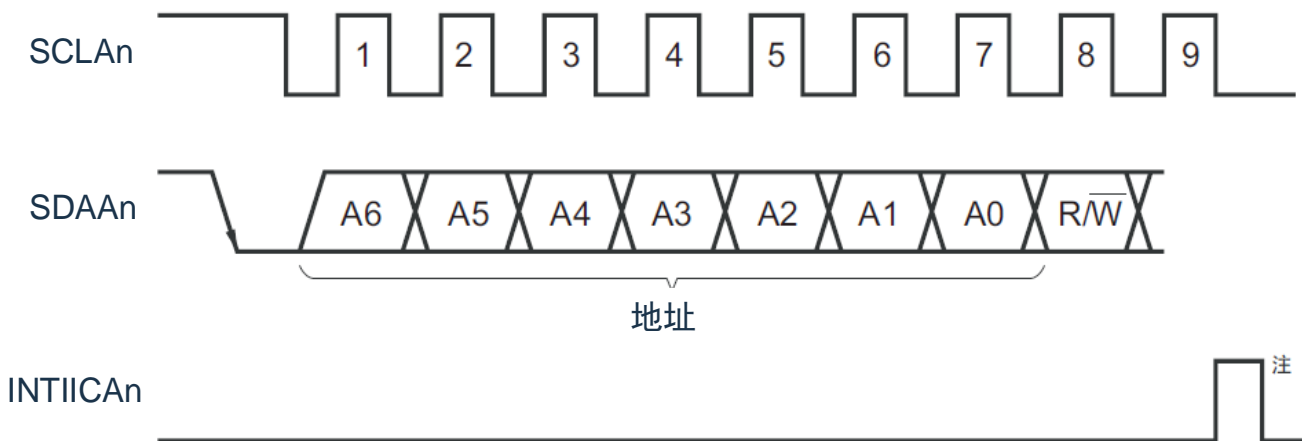
20.5.2 地址

开始条件的后续7位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的7位数据。因此，总线上的从属设备需要设置完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查7位数据是否和从属地址寄存器n(SVAn)的内容相同。此时，如果7位数据和SVAn寄存器的值相同，该从属设备就被选中，在从属设备生成开始条件或者停止条件前，与主控设备进行通信。

图20-15：地址



注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

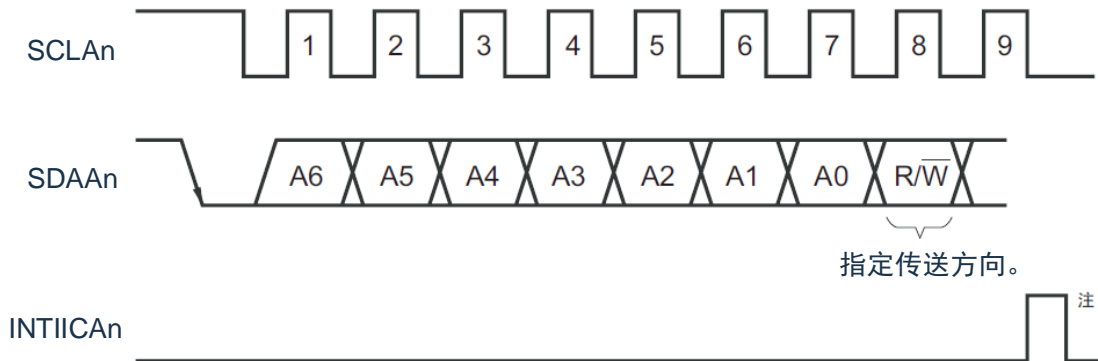
如果将从属地址和“20.5.3 传送方向的指定”中说明的传送方向构成的8位数据写到IICA移位寄存器n(IICAn)，就输出地址。接收到的地址被写到IICAn寄存器。从属地址分配在IICAn寄存器的高7位。

20.5.3 传送方向的指定

主控设备在7位地址之后发送1位指定传送方向的数据。

当此传送方向指定为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定为“1”时，表示主控设备从从属设备接收数据。

图20-16：传送方向的指定



注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

备注：n=0

20.5.4 应答(ACK)

能通过应答(ACK)确认发送方和接收方的串行数据状态。接收方在每次接收到8位数据时返回应答。

通常，发送方在发送8位数据后接收应答。当接收方返回应答时，认为已正常接收，继续处理。能通过IICA状态寄存器n(IICSn)的bit2(ACKDn)确认应答的检测。在主导设备为接收状态下接收到最后的数据时，不返回应答而生成停止条件。在从属设备接收数据后不返回应答时，主导设备输出停止条件或者重新开始条件，中止发送。不返回应答的原因如下：

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

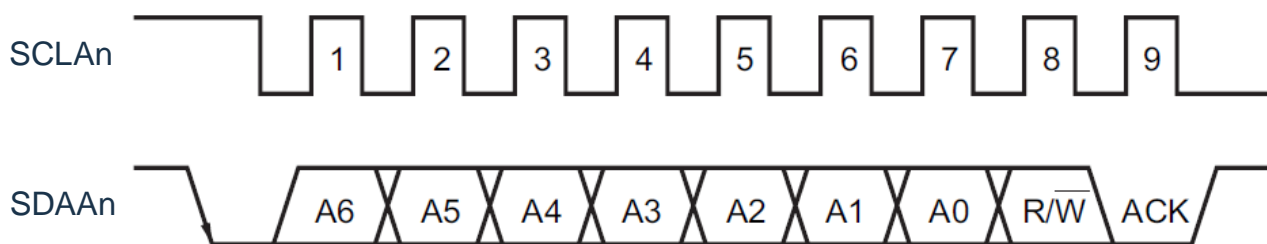
接收方在第9个时钟将SDAAn线置为低电平，生成应答(正常接收)。

通过将IICA控制寄存器n0(IICCTLn0)的bit2(ACKEn)置“1”，变为能自动生成应答的状态。通过7位地址信息后续的第8位数据设置IICSn寄存器的bit3(TRCn)。在接收(TRCn=0)的情况下，通常必须将ACKEn位置“1”。

在从属接收运行过程中(TRCn=0)不能接收数据或者不需要下一个数据时，必须将ACKEn位清“0”，通知主控方不能接收数据。

在主导接收运行过程中(TRCn=0)不需要下一个数据时，为了不生成应答，必须将ACKEn位清“0”，通知从属发送方数据的结束(停止发送)。

图20-17: 应答



当接收到本地站的地址时，与ACKEn位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答(NACK)。

在接收到扩展码时，通过事先将ACKEn位置“1”，生成应答。接收数据时的应答生成方法因等待时序的设置而不同，如下所示。

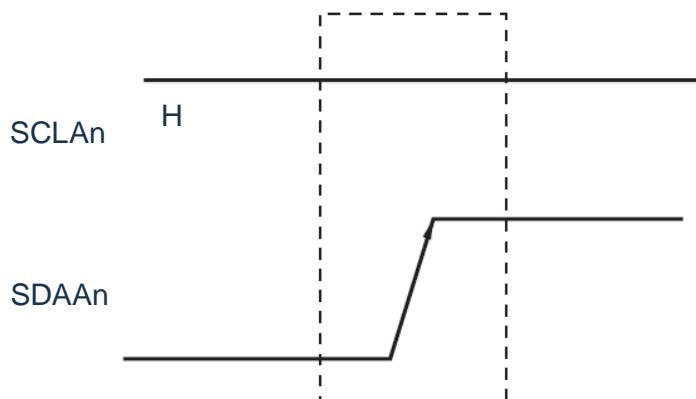
- 当选择8个时钟的等待时(IICCTLn0寄存器的bit3(WTIMn)=0)：通过在解除等待前将ACKEn位置“1”，与SCLAn引脚的第8个时钟下降沿同步生成应答。
- 当选择9个时钟的等待时(IICCTLn0寄存器的bit3(WTIMn)=1)：通过事先将ACKEn位置“1”，生成应答。

备注：n=0

20.5.5 停止条件

在SCLAn引脚为高电平时，如果SDAAn引脚从低电平变为高电平，就生成停止条件。停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图20-18: 停止条件



如果将IICA控制寄存器n0(IICCTLn0)的bit0(SPTn)置“1”，就生成停止条件。如果检测到停止条件，就将IICA状态寄存器n(IICSn)的bit0(SPDn)置“1”，并且在IICCTLn0寄存器的bit4(SPIEn)为“1”时产生INTIICAn。

备注：n=0

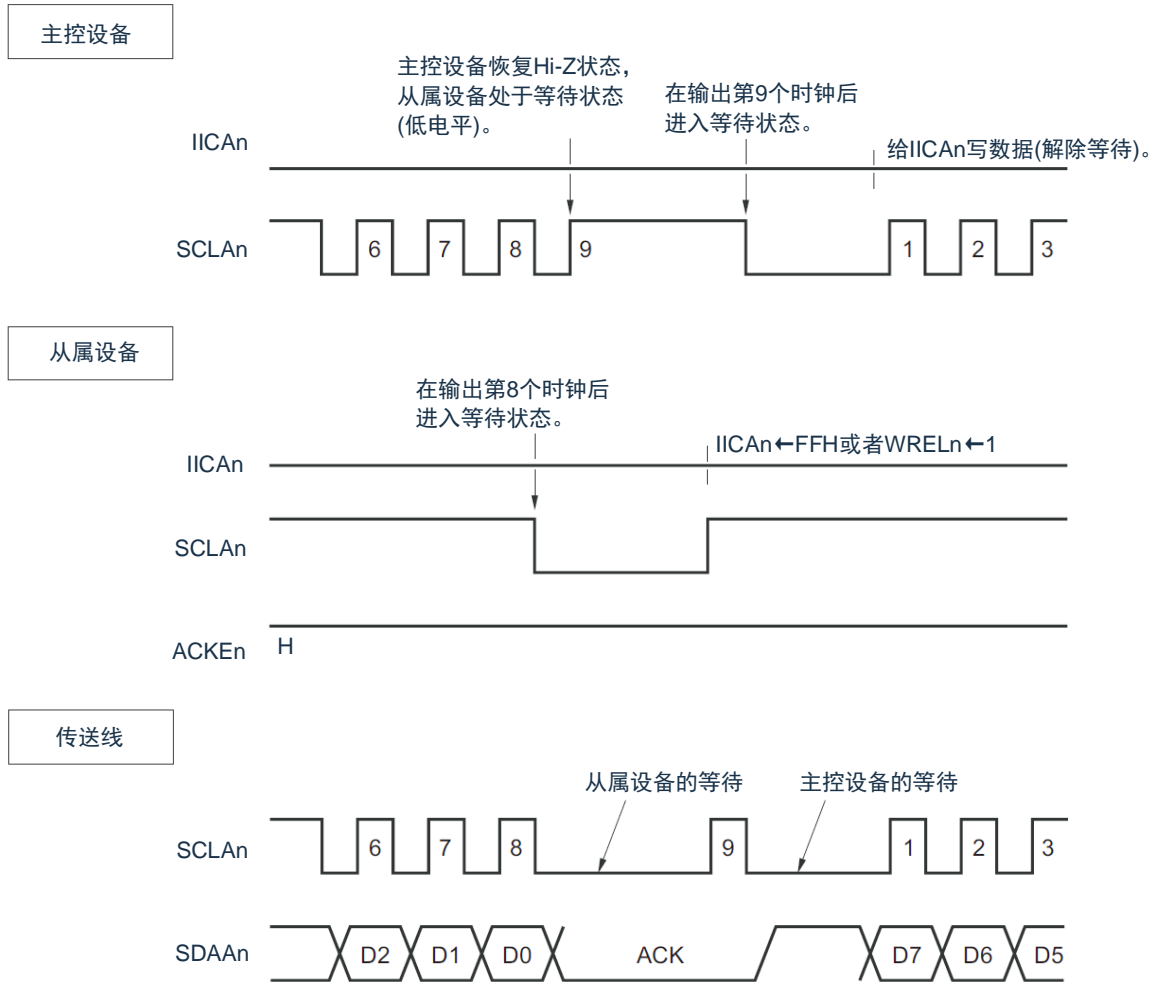
20.5.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送/接收(等待状态)。

通过将SCLAn引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图20-19：等待(1/2)

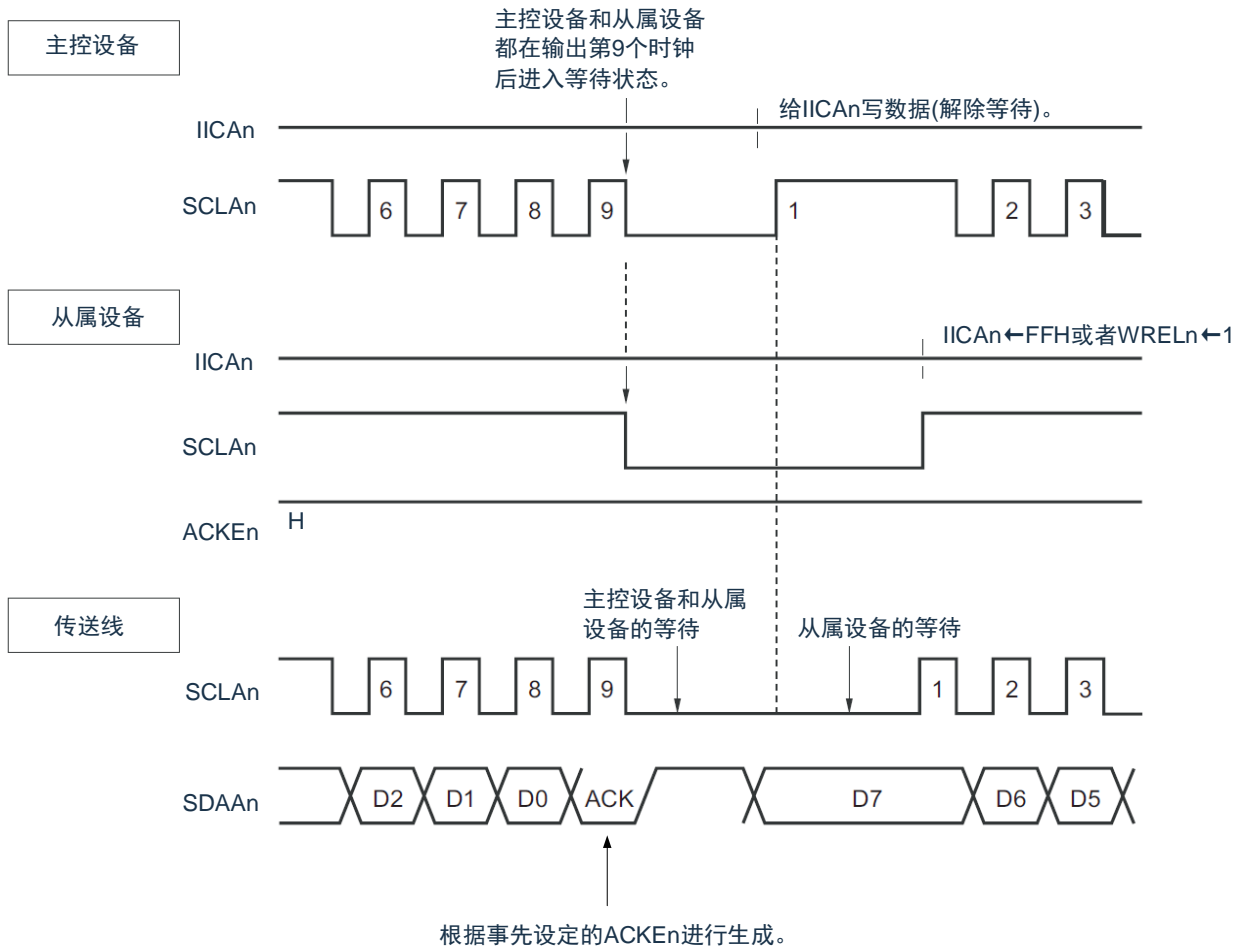
- (1) 主控设备为9个时钟等待，从属设备为8个时钟等待的情况
(主控设备：发送，从属设备：接收，ACKEn=1)



备注：n=0

图20-19: 等待(2/2)

(2) 主控设备和从属设备都为9个时钟等待的情况
 (主控设备: 发送, 从属设备: 接收, ACKEn=1)



备注: ACKEn: IICA控制寄存器n0(IICCTLn0)的bit2
 WRELn: IICA控制寄存器n0(IICCTLn0)的bit5

通过设置IICA控制寄存器n0(IICCTLn0)的bit3(WTIMn)自动产生等待状态。通常, 在接收方, 如果IICCTLn0寄存器的bit5(WRELn)为“1”或者给IICA移位寄存器n(IICAn)写“FFH”, 就解除等待; 在发送方, 如果给IICAn寄存器写数据, 就解除等待。主控设备还能通过以下方法解除等待:

- 将IICCTLn0寄存器的bit1(STTn)置“1”。
- 将IICCTLn0寄存器的bit0(SPTn)置“1”。

备注: n=0

20.5.7 等待的解除方法

在一般情况下，I²C能通过以下的处理来解除等待。

给IICA移位寄存器n(IICAn)写数据。

将IICA控制寄存器n0(IICCTLn0)的bit5(WRELn)置位(解除等待)。

将IICCTLn0寄存器的bit1(STTn)置位(生成开始条件)^注。

将IICCTLn0寄存器的bit0(SPTn)置位(生成停止条件)^注。

注：只限于主控设备。

如果执行了这些等待的解除处理，I²C就解除等待，重新开始通信。要在解除等待后发送数据(包括地址)时，必须给IICAn寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将IICCTLn0寄存器的bit5(WRELn)置“1”。要在解除等待后生成重新开始条件时，必须将IICCTLn0寄存器的bit1(STTn)置“1”。要在解除等待后生成停止条件时，必须将IICCTLn0寄存器的bit0(SPTn)置“1”。对于一次等待只能执行一次解除处理。

例如，如果在通过将WRELn位置“1”来解除等待后给IICAn寄存器写数据，SDAAn线的变化时序与IICAn寄存器的写时序就可能发生冲突，导致将错误的值输出到SDAAn线。除了这些处理以外，在中途中止通信的情况下，如果将IICEn位清“0”，就停止通信，因此能解除等待。在I²C总线状态因噪声而被死锁的情况下，如果将IICCTLn0寄存器的bit6(LRELn)置“1”，就退出通信，因此能解除等待。

注意：如果在WUPn位为“1”时执行等待的解除处理，就不解除等待。

备注：n=0

20.5.8 中断请求(INTIICAn)的产生时序和等待控制

通过设置IICA控制寄存器n0(IICCTLn0)的bit3(WTIMn)，在表20-2所示的时序产生INTIICAn并且进行等待控制。

表20-2: INTIICAn的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 ^{注1,2}	8 ^{注2}	8 ^{注2}	9	8	8
1	9 ^{注1,2}	9 ^{注2}	9 ^{注2}	9	9	9

注1: 只有在接收的地址和从属地址寄存器n(SVAn)的设置地址相同时，从属设备才在第9个时钟的下降沿产生INTIICAn信号并且进入等待状态。

此时，与IICCTLn0寄存器的bit2(ACKEn)的设置无关，生成应答。接收到扩展码的从属设备在第8个时钟的下降沿产生INTIICAn。如果在重新开始后地址不同，就在第9个时钟的下降沿产生INTIICAn，但是不进入等待状态。

注2: 如果接收的地址和从属地址寄存器n(SVAn)的内容不同并且未接收到扩展码，就不产生INTIICAn并且也不进入等待状态。

备注: 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属运行: 与WTIMn位无关，根据上述注1和注2的条件决定中断和等待的时序。
- 主控运行: 与WTIMn位无关，在第9个时钟的下降沿产生中断和等待的时序。

(2) 数据接收

- 主控运行/从属运行: 通过WTIMn位决定中断和等待的时序。

(3) 数据发送

- 主控运行/从属运行: 通过WTIMn位决定中断和等待的时序。

备注: n=0

(4) 等待的解除方法

等待的解除方法有以下4种:

- 给IICA移位寄存器n(IICAn)写数据。
- 将IICA控制寄存器n0(IICCTLn0)的bit5(WRELn)置位(解除等待)。
- 将IICCTLn0寄存器的bit1(STTn)置位(生成开始条件)^注。
- 将IICCTLn0寄存器的bit0(SPTn)置位(生成停止条件)^注。

注: 只限于主控设备。

当选择8个时钟的等待(WTIMn=0)时，需要在解除等待前决定是否生成应答。

(5) 停止条件的检测

如果检测到停止条件，就产生INTIICAn(只限于SPIEn=1的情况)。

20.5.9 地址匹配的检测方法

在I²C总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。能通过硬件自动检测地址匹配。当主控设备发送的从属地址和从属地址寄存器n(SVAn)的设置地址相同或者只接收到扩展码时，产生INTIICAn中断请求。

20.5.10 错误的检测

在I²C总线模式中，因为发送过程中的串行数据总线(SDAAn)的状态被取到发送器件的IICA移位寄存器n(IICAn)，所以能通过将开始发送前和发送结束后的IICA数据进行比较来检测发送错误。此时，如果2个数据不同，就判断为发生了发送错误。

备注：n=0

20.5.11 扩展码

- (1) 当接收地址的高4位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志(EXCn)置“1”，并且在第8个时钟的下降沿产生中断请求(INTIICAn)。不影响保存在从属地址寄存器n(SVAn)的本地站地址。
- (2) 当SVAn寄存器的设置值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求(INTIICAn)。
 - 高4位数据相同：EXCn=1
 - 7位数据相同：COIn=1

备注：EXCn：IICA状态寄存器n(IICSn)的bit5

COIn：IICA状态寄存器n(IICSn)的bit4

- (3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将IICA控制寄存器n0(IICCTLn0)的bit6(LRELn)置“1”，进入下一次通信的待机状态。

表20-3：主要扩展码的位定义

从属地址	R/W位	说明
0000000	0	全呼地址
11110xx	0	10位从属地址的指定(地址认证时)
11110xx	1	10位从属地址的指定(在地址相同后发行读命令时)

备注：

1. 有关上述以外的扩展码，请参照NXP公司发行的I2C总线规格书。
2. n=0

20.5.12 仲裁

当多个主控设备同时生成开始条件时(在STDn位变为“1”前将STTn位置“1”的情况), 边调整时钟边进行主控设备的通信, 直到数据不同为止。此运行称为仲裁。

在仲裁失败时, 仲裁失败的主控设备将IICA状态寄存器n(IICSn)的仲裁失败标志(ALDn)置“1”, 并且将SCLAn线和SDAAn线都置为高阻抗状态, 释放总线。

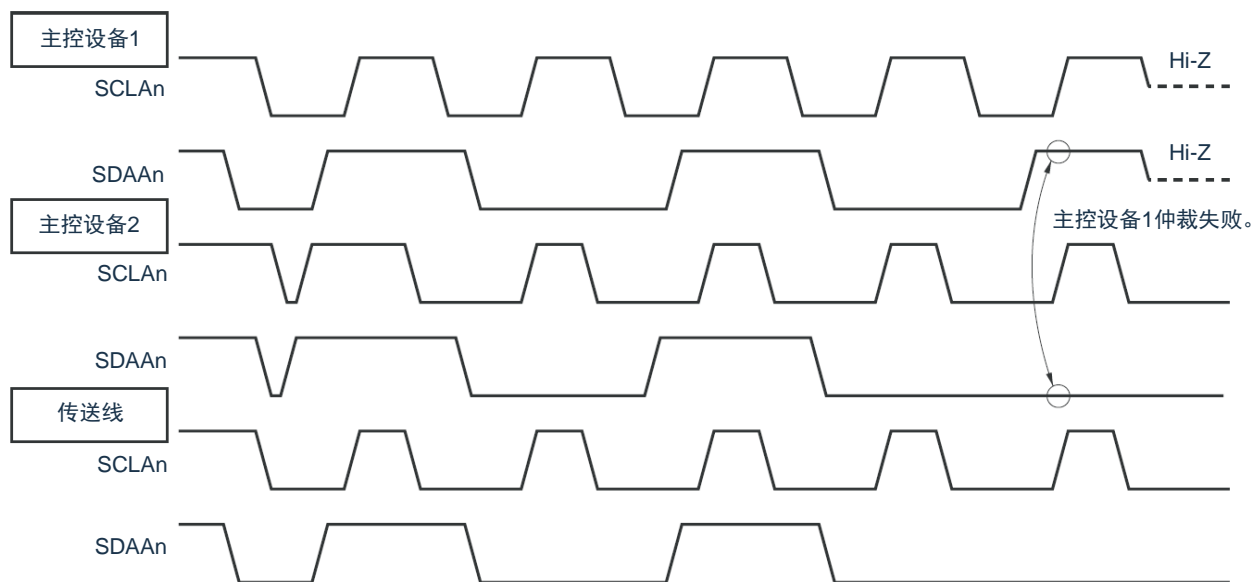
在发生下一次中断请求时(例如: 在第8或者第9个时钟检测到停止条件), 用软件通过ALDn位为“1”来检测仲裁的失败。

有关中断请求的产生时序, 请参照“20.5.8 中断请求(INTIICAn)的产生时序和等待控制”。

备注: STDn: IICA状态寄存器n(IICSn)的bit1

STTn: IICA控制寄存器n0(IICCTLn0)的bit1

图20-20: 仲裁时序例子



备注: n=0

表20-4: 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时(SPIEn=1) ^{注2}
想要生成重新开始条件, 但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件, 但是检测到停止条件。	在生成停止条件时(SPIEn=1) ^{注2}
想要生成停止条件, 但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件, 但是SCLAn为低电平。	

注1: 当WTIMn位(IICA控制寄存器n0(IICCTLn0)的bit3)为“1”时, 在第9个时钟的下降沿产生中断请求; 当WTIMn位为“0”并且接收到扩展码的从属地址时, 在第8个时钟的下降沿产生中断请求。

注2: 当有可能发生仲裁时, 必须在主控运行时将SPIEn位置“1”。

备注:

1. SPIEn: IICA控制寄存器n0(IICCTLn0)的bit4
2. n=0

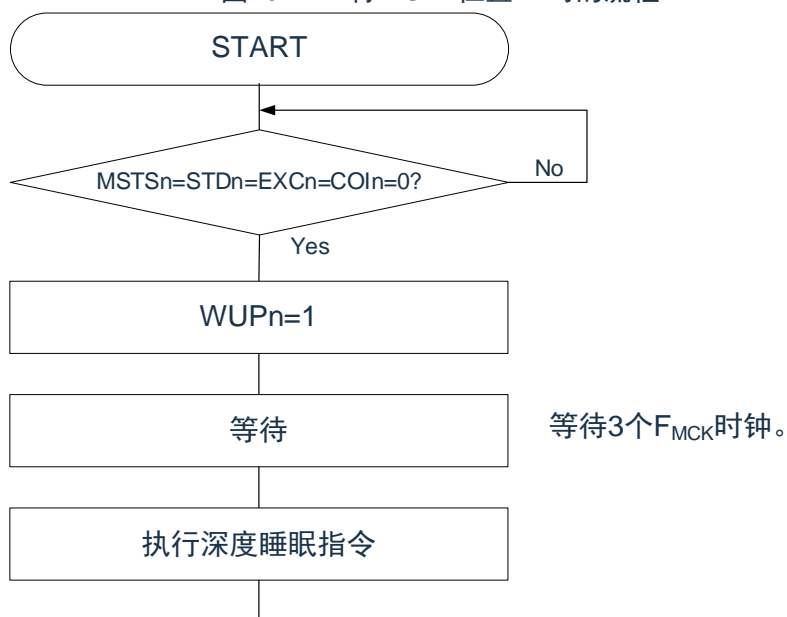
20.5.13 唤醒功能

这是I²C的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号(INTIICAn)的功能。在地址不同的情况下不产生不需要的INTIICAn信号，能提高处理效率。如果检测到开始条件，就进入唤醒待机状态。因为主控设备(已经生成开始条件的情况)也有可能因仲裁失败而变为从属设备，所以在发送地址的同时进入唤醒待机状态。

要在深度睡眠模式中使用唤醒功能时，必须将WUPn位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号(INTIICAn)。在产生此中断后，通过指令将WUPn位清“0”，返回到通常运行。

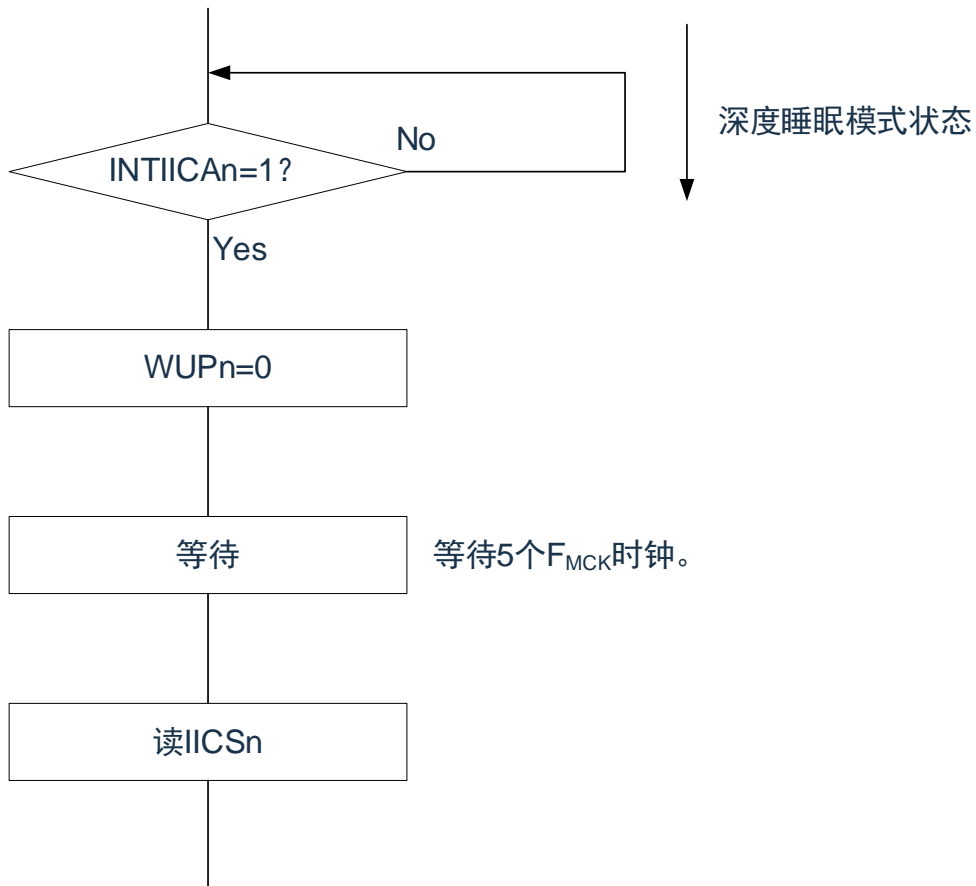
将WUPn位置“1”时的流程如图20-21所示，通过地址匹配将WUPn位置“0”时的流程如图20-22所示。

图20-21：将WUPn位置“1”时的流程



备注：n=0

图20-22: 通过地址匹配将WUPn位置“0”时的流程(包括接收扩展码)



在确认串行接口IICA的运行状态后，
根据要执行的内容进行处理。

除了串行接口IICA产生的中断请求(INTIICAn)以外，必须通过以下的流程解除深度睡眠模式。

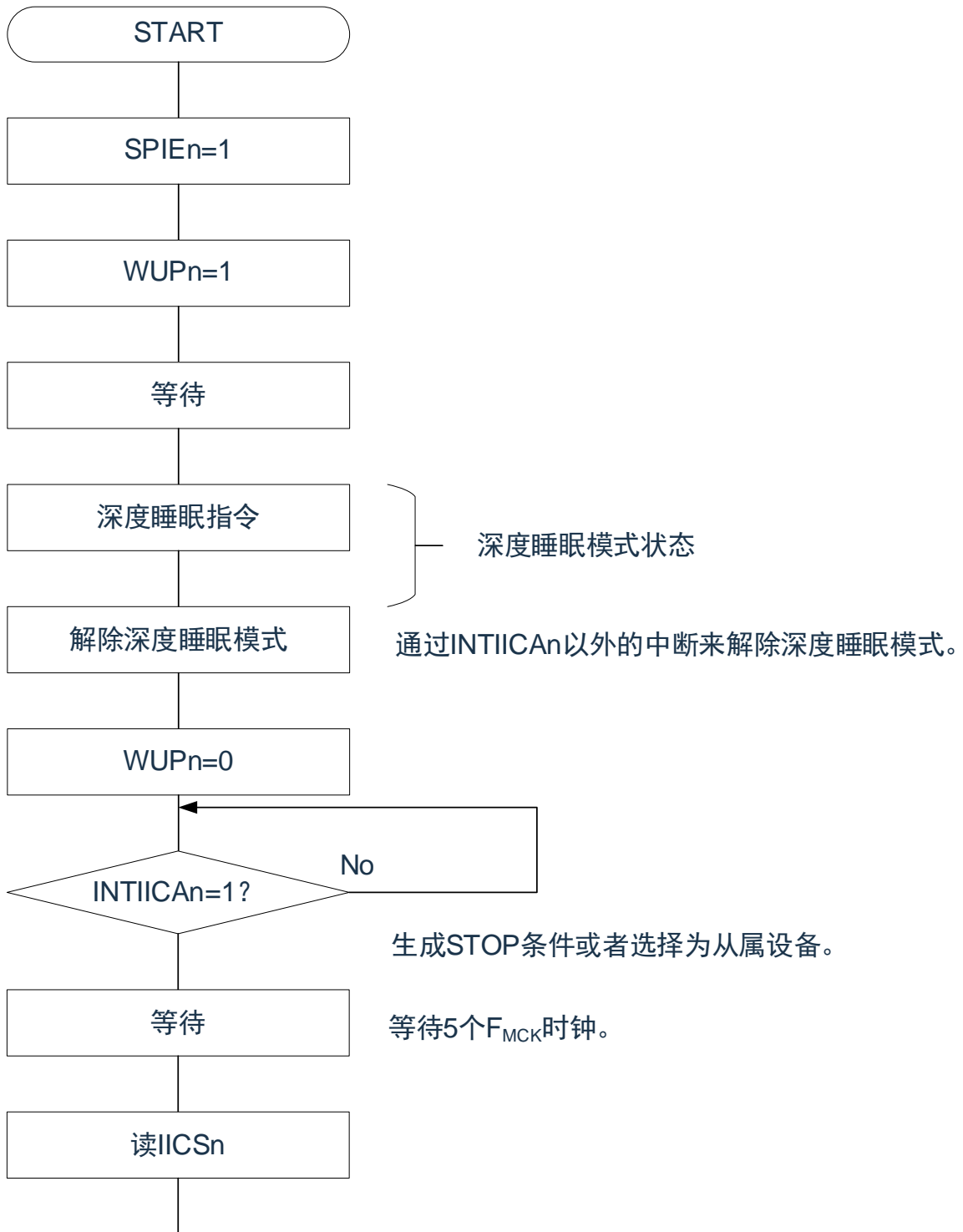
- 下一次IIC通信为主控设备运行的情况：图20-23的流程
- 下一次IIC通信为从属设备运行的情况：

通过INTIICAn中断返回的情况：和图20-22的流程相同。

通过INTIICAn中断以外的中断返回的情况：必须在产生INTIICAn中断前保持WUPn位为“1”的状态继续运行。

备注：n=0

图20-23: 在通过INTIICAn以外的中断来解除深度睡眠模式后作为主控设备运行的情况



在确认串行接口IICA的运行状态后，
根据要执行的内容进行处理。

备注：n=0

20.5.14 通信预约

(1) 允许通信预约功能的情况(IICA标志寄存器n(IICFn)的bit0(IICRSVn)=0)

要在不加入总线的状态下进行下一次主控通信时，能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下2种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时(不返回应答而将IICA控制寄存器n0(IICCTLn0)的bit6(LRELn)置“1”，退出通信后释放了总线)

如果在不加入总线的状态下将IICCTLn0寄存器的bit1(STTn)置“1”，就在释放总线后(检测到停止条件)自动生成开始条件，进入等待状态。

将IICCTLn0寄存器的bit4(SPIEn)置“1”，在通过产生的中断请求信号(INTIICAn)检测到总线的释放(检测到停止条件)后，如果给IICA移位寄存器n(IICAn)写地址，就自动作为主控设备开始通信。在检测到停止条件前，给IICAn寄存器写的的数据无效。

当将STTn位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时.....生成开始条件
- 总线未处于释放状态(待机状态)时.....通信预约

在将STTn位置“1”并且经过等待时间后，通过MSTSn位(IICA状态寄存器n(IICSn)的bit7)确认是否作为通信预约运行。

必须通过软件确保以下计算式计算的等待时间：

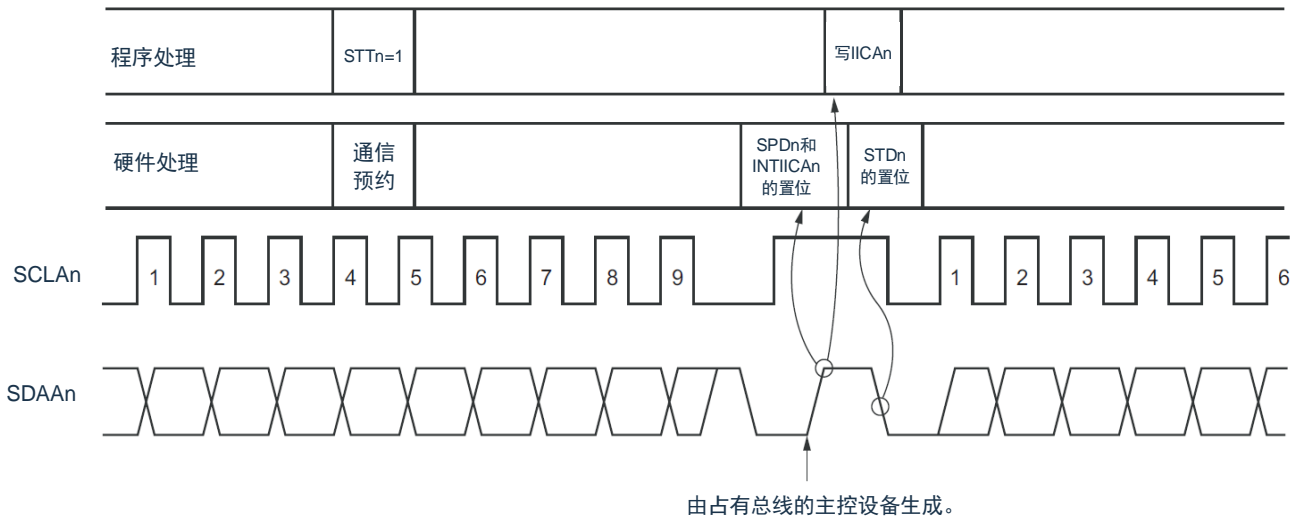
从将STTn位置“1”到确认MSTSn标志为止的等待时间：
 $(IICWLn的设置值+IICWHn的设置值+4)/F_{MCK}+T_F \times 2$

备注：

1. IICWLn: IICA低电平宽度设置寄存器n
 IICWHn: IICA高电平宽度设置寄存器n
 T_F : SDAAn信号和SCLAn信号的下降时间
 F_{MCK} : IICA运行时钟频率
2. n=0

通信预约的时序如图20-24所示。

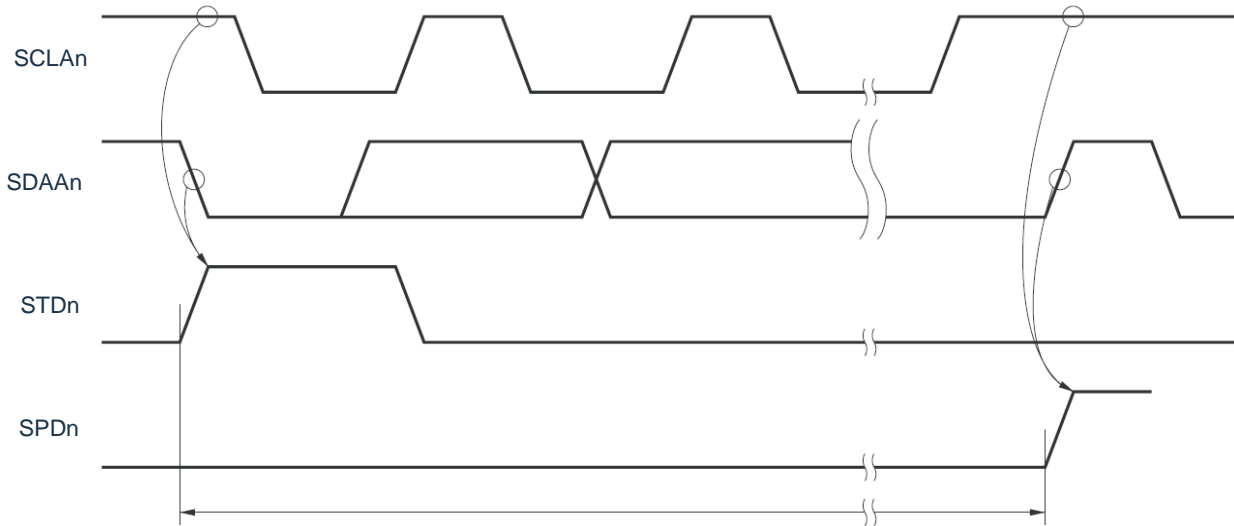
图20-24：通信预约的时序



- 备注：IICAn：IICA移位寄存器
- STTn：IICA控制寄存器n0(IICCTLn0)的bit1
- STDn：IICA状态寄存器n(IICSn)的bit1
- SPDn：IICA状态寄存器n(IICSn)的bit0

通过图20-25所示的时序接受通信预约。在IICA状态寄存器n(IICSn)的bit1(STDn)变为“1”后并且在检测到停止条件前，将IICA控制寄存器n0(IICCTLn0)的bit1(STTn)置“1”进行通信预约。

图20-25：通信预约的接受时序

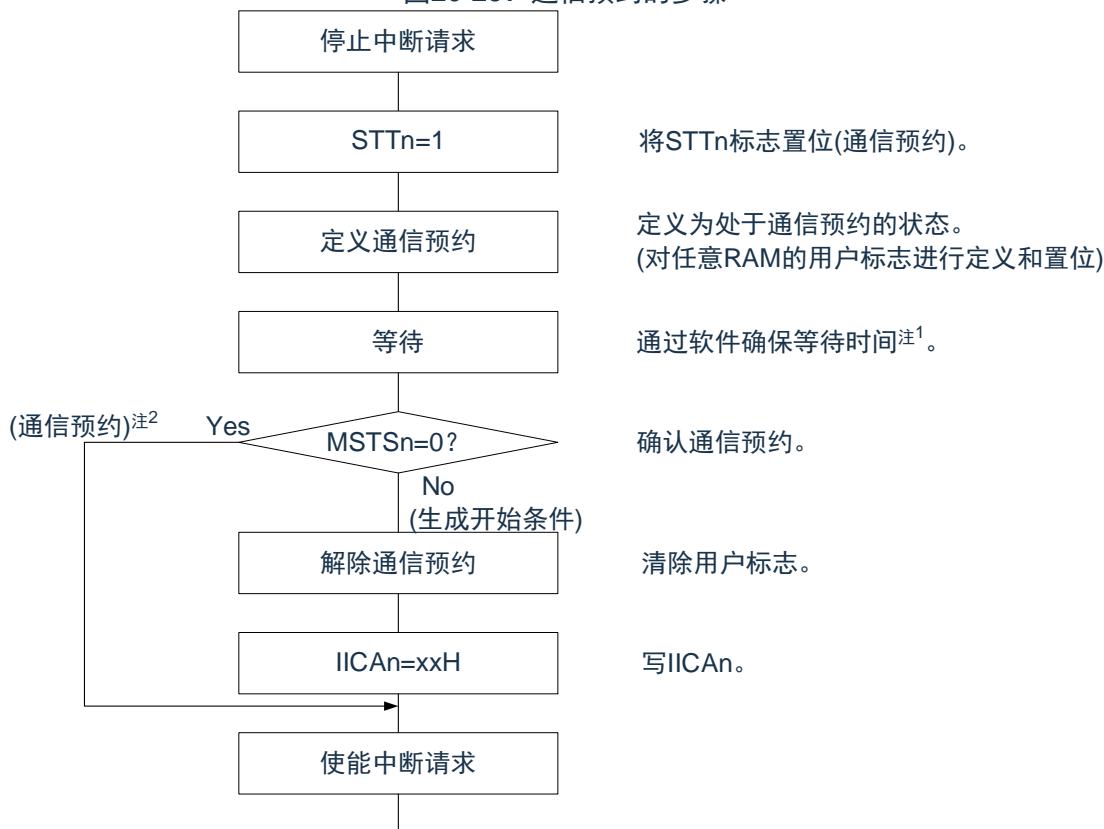


待机状态(在此期间，能通过将STTn位置“1”进行通信预约)

通信预约的步骤如图20-26所示。

备注：n=0

图20-26: 通信预约的步骤



注1: 等待时间如下: $(IICWLn\text{的设置值} + IICWHn\text{的设置值} + 4) / F_{MCK} + T_F \times 2$

注2: 在通信预约运行时, 通过停止条件中断请求来写IICA移位寄存器n(IICAn)。

备注:

1. STTn: IICA控制寄存器n0(IICCTLn0)的bit1
 MSTSn: IICA状态寄存器n(IICSn)的bit7
 IICAn: IICA移位寄存器n
 IICWLn: IICA低电平宽度设置寄存器n
 IICWHn: IICA高电平宽度设置寄存器n
 T_F : SDAAn信号和SCLAn信号的下降时间
 F_{MCK} : IICA运行时钟频率
2. n=0

(2) 禁止通信预约功能的情况(IICA标志寄存器n(IICFn)的bit0(IICRSVn)=1)

在总线通信过程中，如果不参加此通信的状态下将IICA控制寄存器n0(IICCTLn0)的bit1(STTn)置“1”，就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下2种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时(不返回应答而将IICCTLn0寄存器的bit6(LRELn)置“1”，退出通信后释放了总线)

能通过STCFn(IICFn寄存器的bit7)来确认是生成了开始条件还是拒绝了请求。因为从STTn位为“1”到将STCFn位置“1”为止需要5个 F_{MCK} 时钟的时间，所以必须通过软件确保此时间。

备注：n=0

20.5.15 其他注意事项

(1) STCENn位为“0”的情况

在刚允许I²C运行(IICEn=1)后, 与实际的总线状态无关而视为通信状态(IICBSYn=1)。要在没有检测到停止条件的状态下进行主控通信时, 必须先生成停止条件, 在释放总线后进行主控通信。对于多主控, 在总线未释放(未检测到停止条件)的状态下不能进行主控通信。按照以下顺序生成停止条件:

- ① 设置IICA控制寄存器n1(IICCTLn1)。
- ② 将IICA控制寄存器n0(IICCTLn0)的bit7(IICEn)置“1”。
- ③ 将IICCTLn0寄存器的bit0(SPTn)置“1”。

(2) STCENn位为“1”的情况

在刚允许I²C运行(IICEn=1)后, 与实际的总线状态无关而视为释放状态(IICBSYn=0)。因此在生成第1个开始条件(STTn=1)时, 为了不破坏其他的通信, 需要确认总线已被释放。

(3) 正在和其他设备进行I²C通信的情况

在SDAAn引脚为低电平并且SCLAn引脚为高电平时, 如果允许I²C运行而且中途参加通信, I²C的宏就视为SDAAn引脚从高电平变为低电平(检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的I²C通信。为了避免这种情况, 必须按照以下顺序启动I²C:

- ① 将IICCTLn0寄存器的bit4(SPIEn)清“0”, 禁止在检测到停止条件时产生中断请求信号(INTIICAn)。
- ② 将IICCTLn0寄存器的bit7(IICEn)置“1”, 允许I²C运行。
- ③ 等待检测到开始条件。
- ④ 在返回应答前(将IICEn位置“1”后的4~72个F_{MCK}时钟内)将IICCTLn0寄存器的bit6(LRELn)置“1”, 强制将检测置为无效。

(4) 在将STTn位和SPTn位(IICCTLn0寄存器的bit1和bit0)置位后, 禁止清“0”前的再置位。

(5) 如果进行了通信预约, 就必须将SPIEn位(IICCTLn0寄存器的bit4)置“1”, 在检测到停止条件时产生中断请求。在产生中断请求后, 通过给IICA移位寄存器n(IICAn)写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因在开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测MSTSn位(IICA状态寄存器n(IICSn)的bit7)时, 不需要将SPIEn位置“1”。

备注: n=0

20.5.16 通信运行

在此通过流程图表示以下3个运行步骤。

(1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设置”和“通信处理”。在启动时执行“初始设置”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在I²C总线的多主控系统中，只根据I²C总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内(1帧)为高电平，就将总线作为释放状态而参加通信。此流程大体分为“初始设置”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设置”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

用作I²C总线从属设备的例子如下所示。

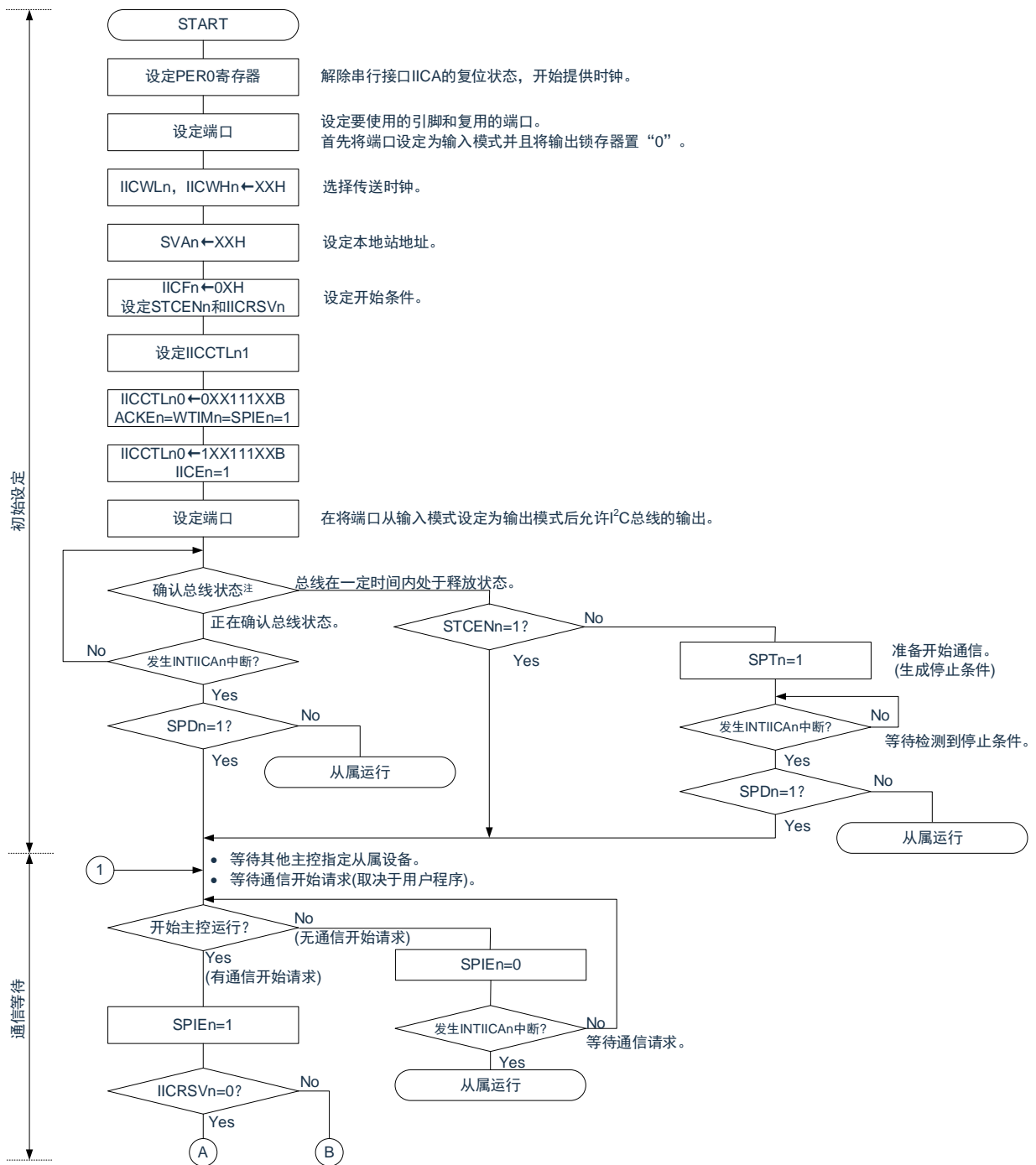
当用作从属设备时，通过中断开始运行。在启动时执行“初始设置”部分，然后通过“通信等待”，等待INTIICAn中断的发生。如果发生INTIICAn中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注：n=0

(2) 多主控系统的主控运行

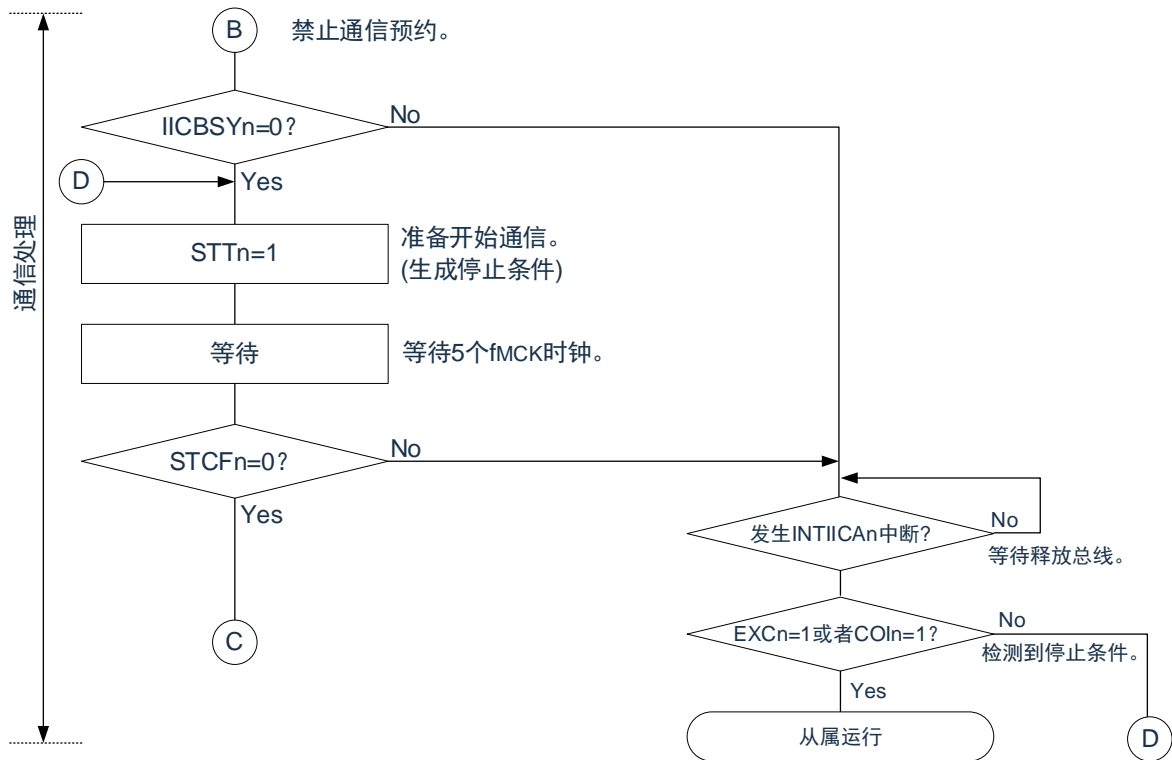
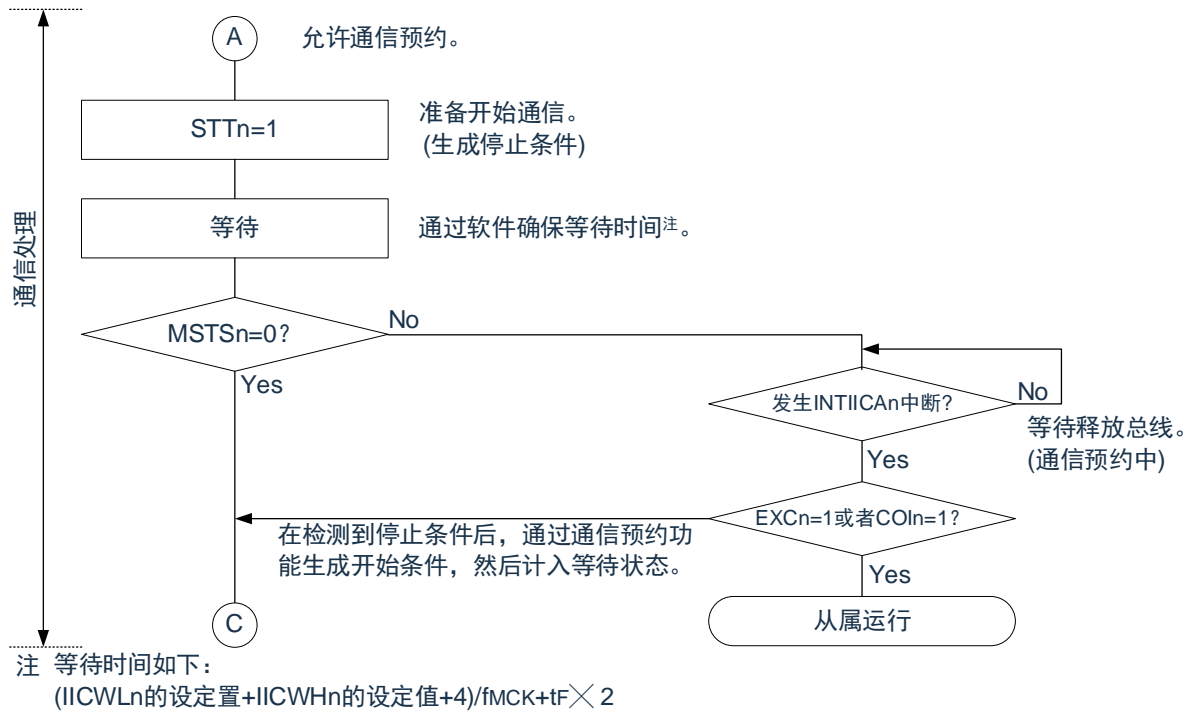
图20-28: 多主控系统的主控运行(1/3)



注: 必须确认总线在一定时间内(例如, 1帧)处于释放状态(CLDn位=1、DADn位=1)。当SDAAn引脚固定为低电平时, 必须根据通信中的产品的规格, 判断是否释放I²C总线(SCLAn引脚和SDAAn引脚为高电平)。

备注: n=0

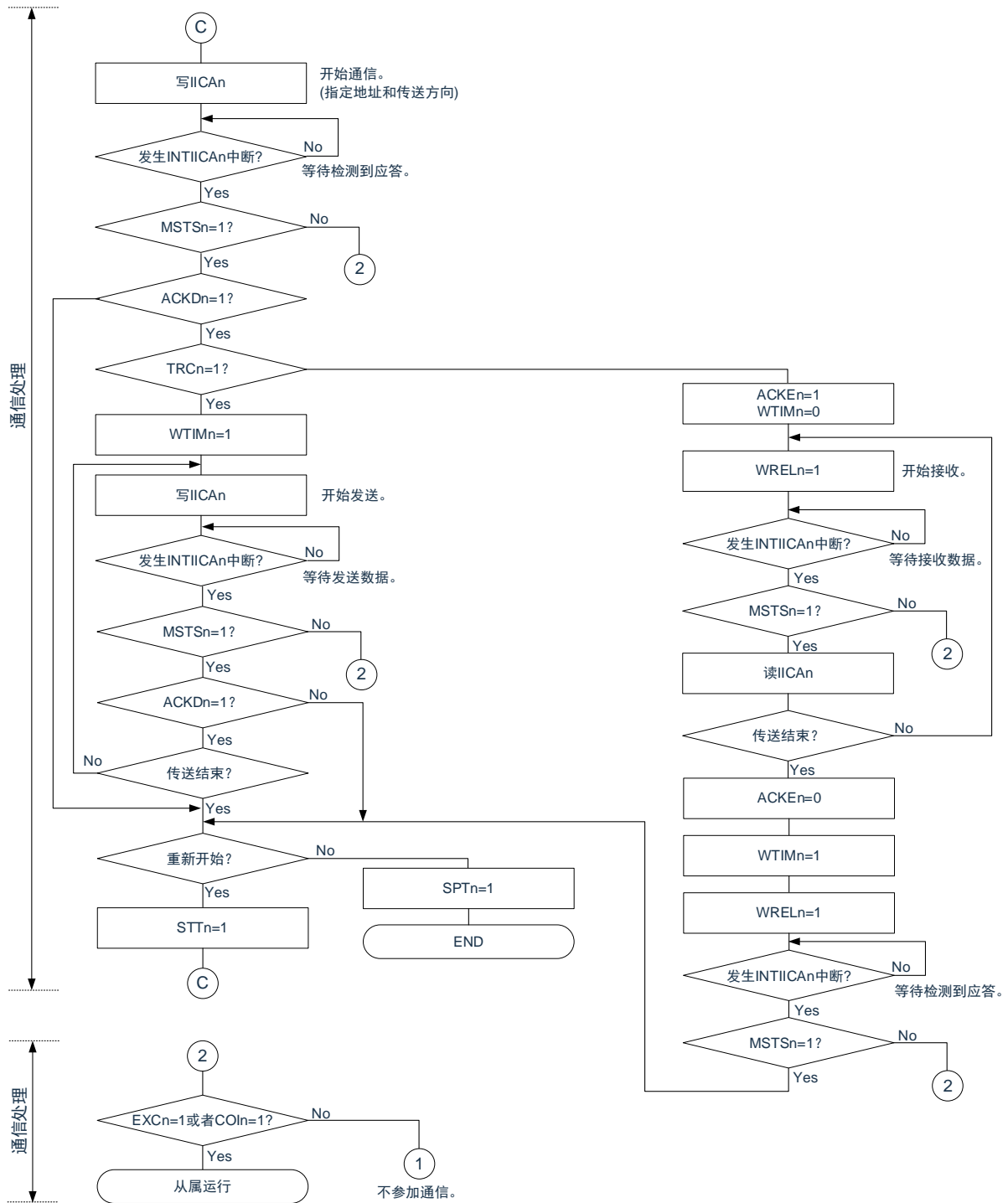
图20-28：多主控系统的主控运行(2/3)



备注：

1. IICWLn: IICA低电平宽度设置寄存器n
 IICWHn: IICA高电平宽度设置寄存器n
 T_F : SDAAn信号和SCLAn信号的下降时间
 f_{MCK} : IICA运行时钟频率
2. n=0

图20-28: 多主控系统的主控运行(3/3)



备注:

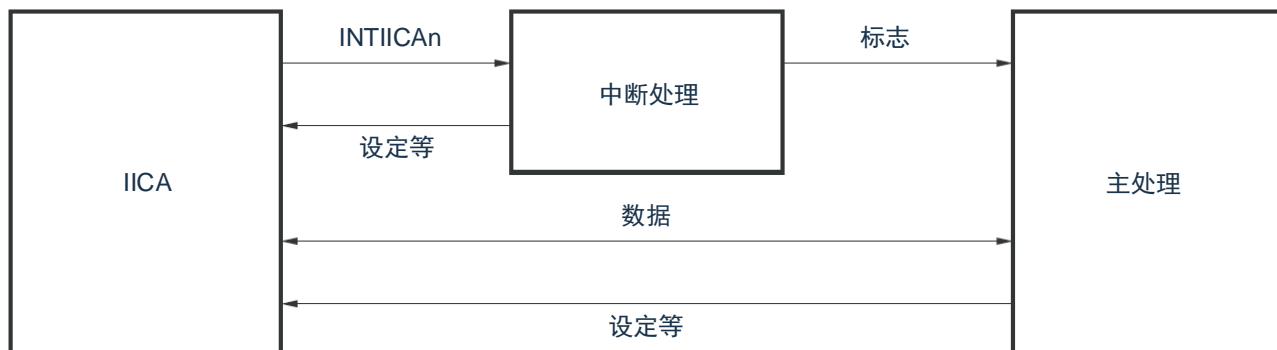
1. 传送和接收的格式必须符合通信中的产品的规格。
2. 在多主控系统中用作主控设备的情况下，必须在每次发生INTIICAn中断时读MSTSn位，确认仲裁结果。
3. 在多主控系统中用作从属设备的情况下，必须在每次发生INTIICAn中断时通过IICA状态寄存器n(IICSn)和IICA标志寄存器n(IICFn)确认状态，决定以后的处理。
4. n=0

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过INTIICAn中断进行处理(需要对通信中的停止条件检测等的运行状态进行很大的变更处理)。

在此说明中，假设数据通信不支持扩展码，INTIICAn中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下3个标志并且代替INTIICAn将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下2种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态(有效地址的检测~停止条件的检测，未检测到主控设备的应答，地址不同)

② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和INTIICAn中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第1个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第1个数据(地址匹配被解释为下一次数据请求)。

③ 通信方向标志

此标志表示通信方向，和TRCn位的值相同。

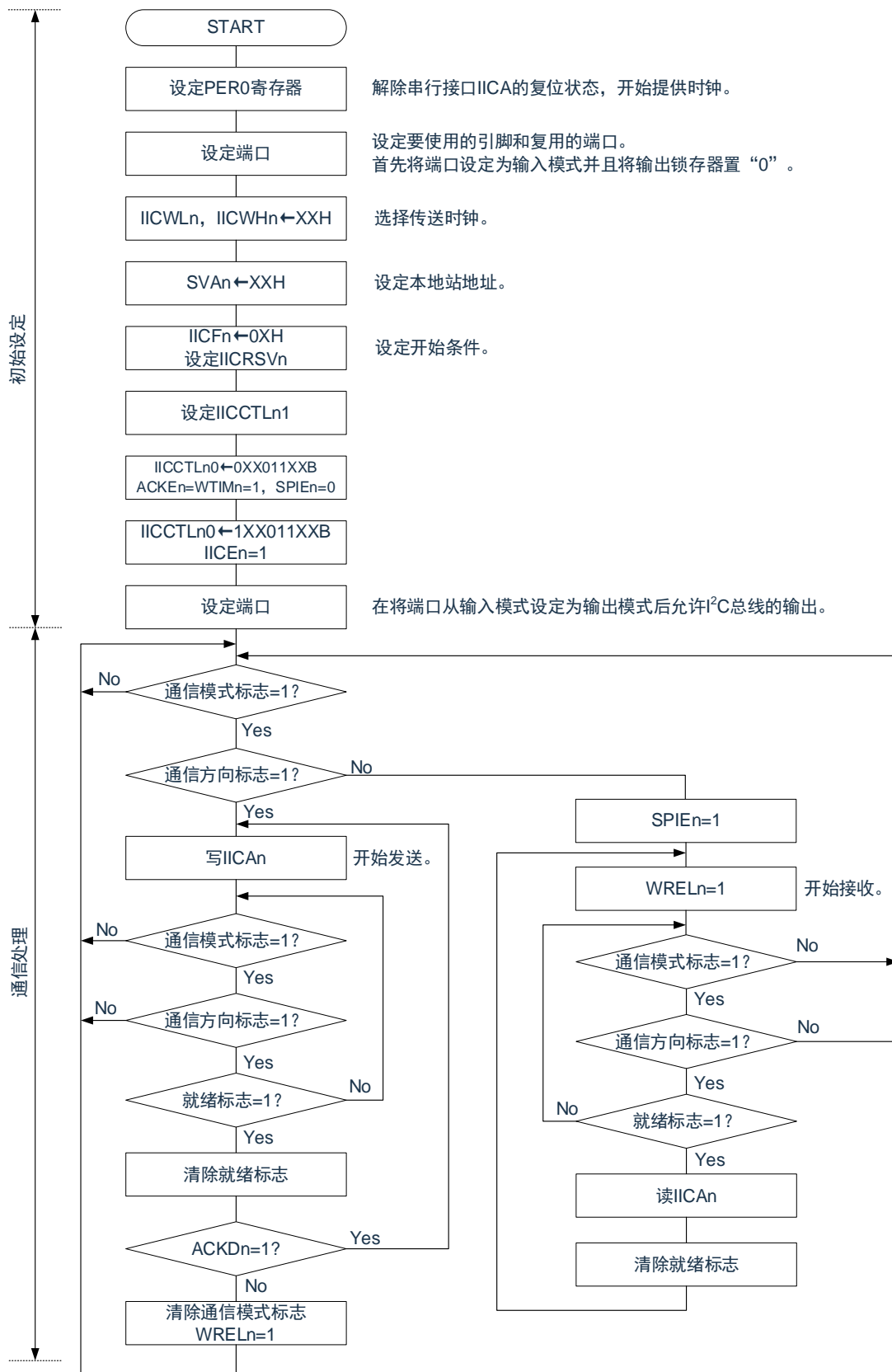
备注：n=0

从属运行的主处理部的运行如下所示。

启动串行接口IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信(因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态)。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图20-29: 从属运行步骤(1)



备注:

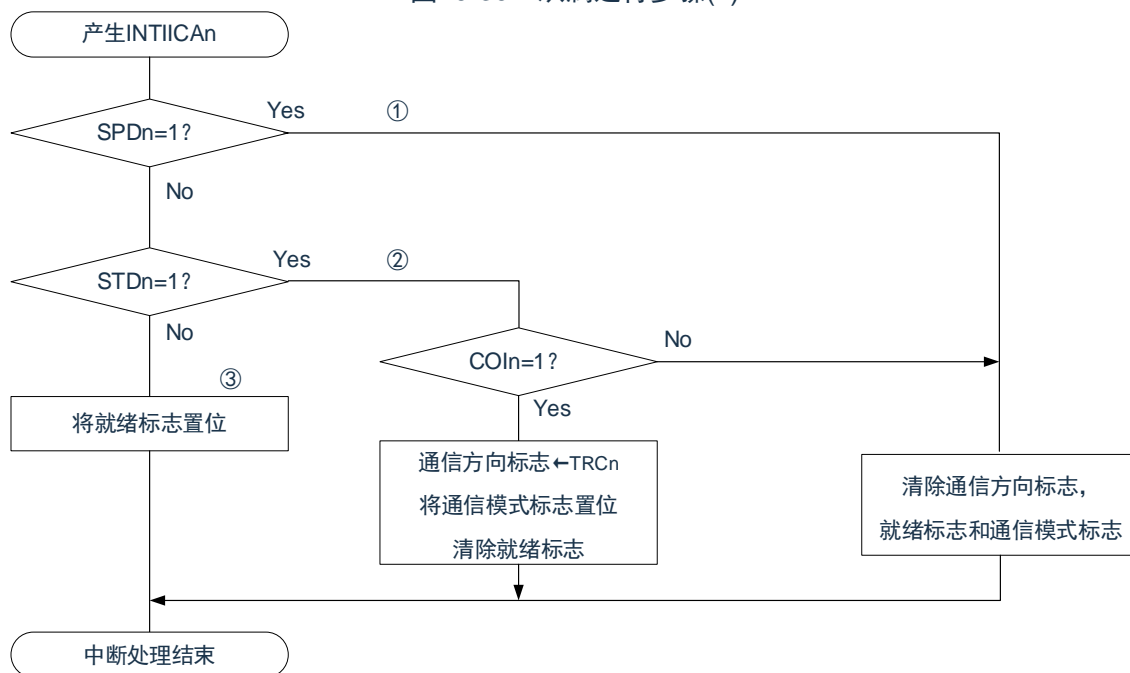
1. 传送和接收的格式必须符合通信中的产品的规格。
2. n=0

从属设备通过INTIICAn中断进行处理的步骤例子如下所示(在此假设没有用扩展码进行处理)。通过INTIICAn中断确认状态并且进行以下处理。

- ① 如果生成停止条件，就结束通信。
- ② 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设置为通信模式并且解除等待，然后从中断返回(清除就绪标志)。
- ③ 当发送和接收数据时，只要将就绪标志置位，I²C总线就保持等待状态并且从中断返回。

备注：上述的①~③对应“图20-30：从属运行步骤(2)”的①~③。

图20-30：从属运行步骤(2)



备注：n=0

20.5.17 I²C中断请求(INTIICAn)的产生时序

数据的发送和接收时序、INTIICAn中断请求信号的产生时序以及产生INTIICAn信号时的IICA状态寄存器n(IICSn)的值如下所示。

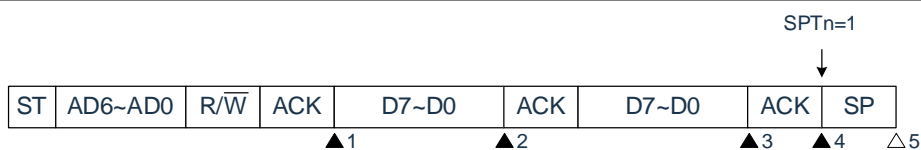
备注:

1. ST: 开始条件
AD6~AD0: 地址
 $\overline{R/W}$: 传送方向的指定
ACK: 应答
D7~D0: 数据
SP: 停止条件
2. n=0

(1) 主控运行

(a) Start~Address~Data~Data~Stop(发送和接收)

(i) WTIMn=0的情况

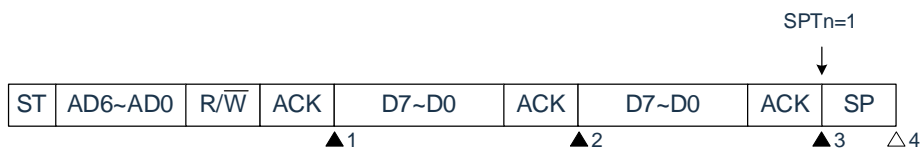


- ▲1:IICSn=1000X110B
- ▲2:IICSn=1000X000B
- ▲3:IICSn=1000X000B(将WTIMn位置“1”)注
- ▲4:IICSn=1000XX00B(将SPTn位置“1”)
- △5:IICSn=00000001B

注 为了生成停止条件，必须将WTIMn位置“1”并且更改INTIICAn中断请求信号的产生时序。

- 备注
- ▲ 一定产生
 - △ 只在SPIEn位为“1”时产生。
 - X 任意

(ii) WTIMn=1的情况



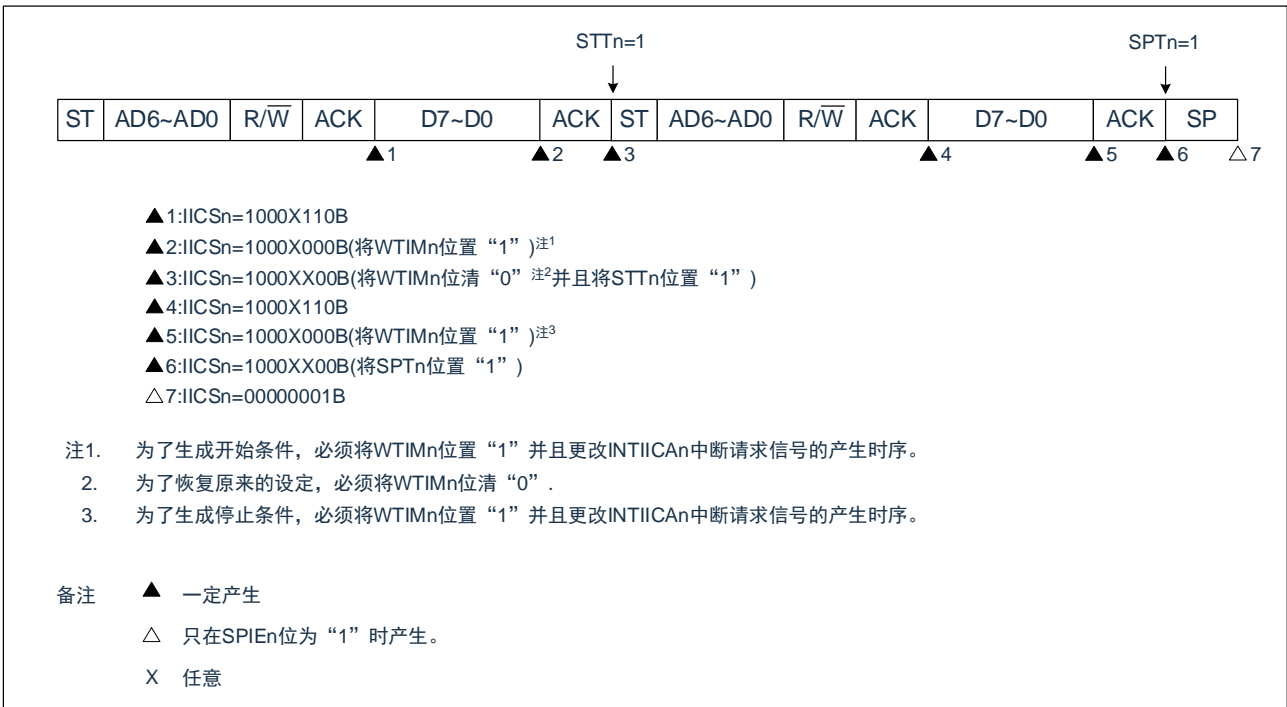
- ▲1:IICSn=1000X110B
- ▲2:IICSn=1000X100B
- ▲3:IICSn=1000XX00B(将SPTn位置“1”)
- △4:IICSn=00000001B

- 备注
- ▲ 一定产生
 - △ 只在SPIEn位为“1”时产生。
 - X 任意

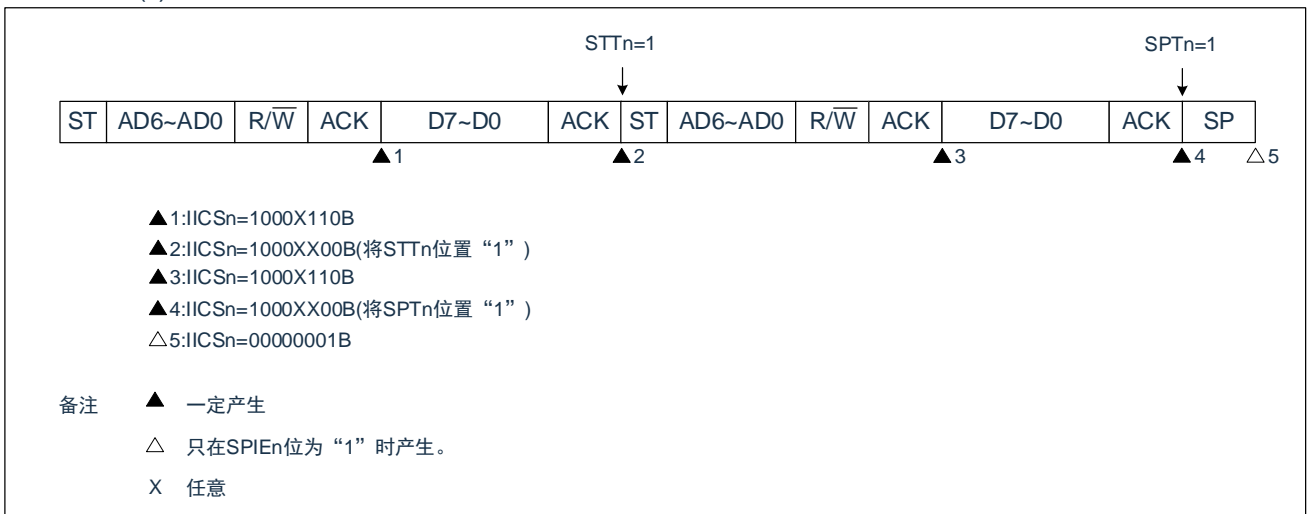
备注：n=0

(b) Start~Address~Data~Start~Address~Data~Stop(重新开始)

(i) WTIMn=0的情况



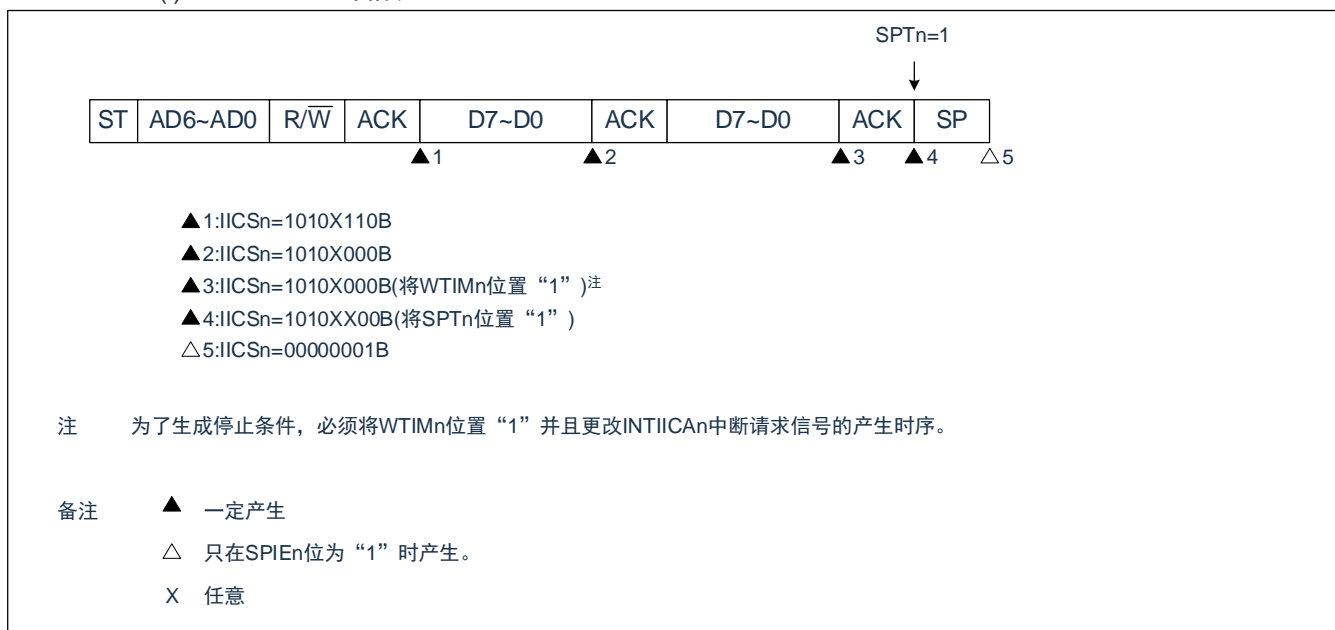
(ii) WTIMn=1的情况



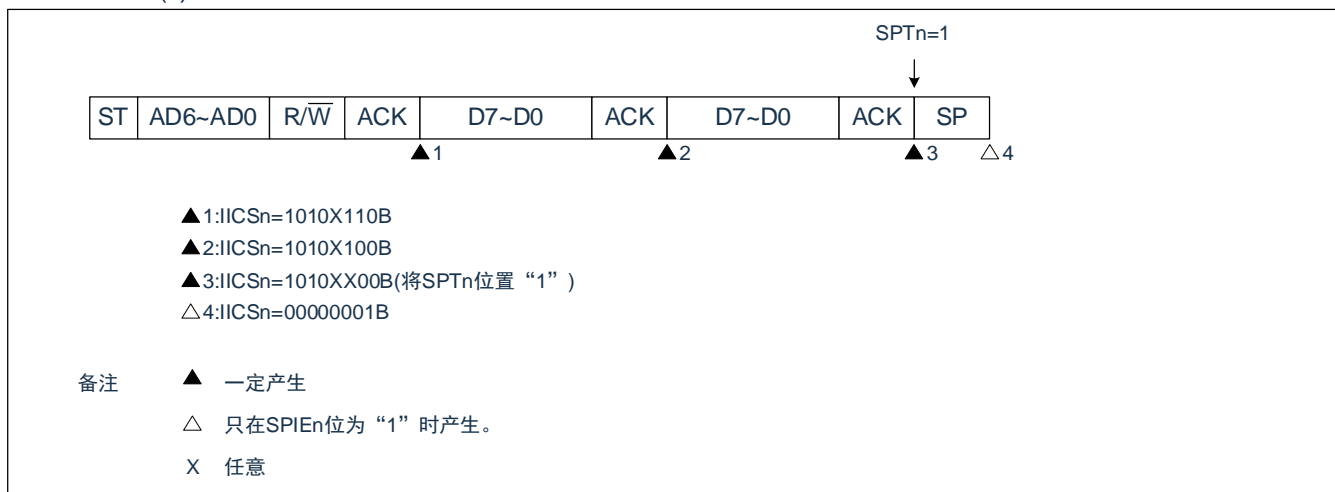
备注: n=0

(c) Start~Code~Data~Data~Stop(发送扩展码)

(i) WTIMn=0的情况



(ii) WTIMn=1的情况



备注：n=0

(2) 从属运行(接收从属地址的情况)

(a) Start~Address~Data~Data~Stop

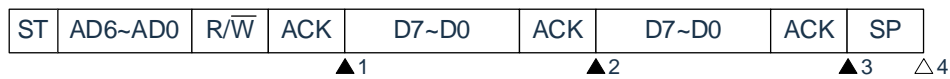
(i) WTIMn=0的情况



- ▲1:IICSn=0001X110B
- ▲2:IICSn=0001X000B
- ▲3:IICSn=0001X000B
- △4:IICSn=00000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

(ii) WTIMn=1的情况



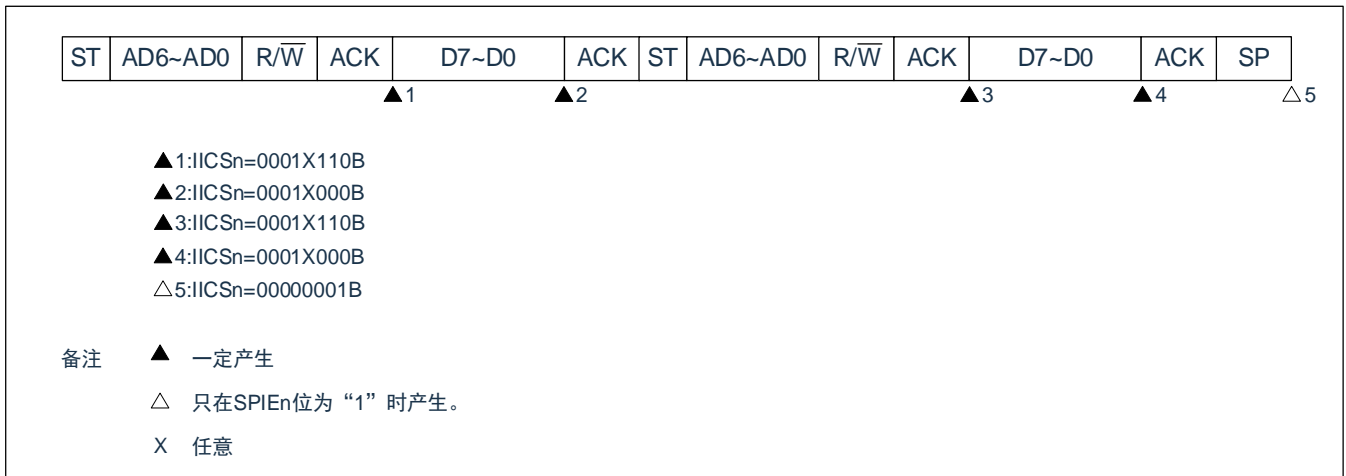
- ▲1:IICSn=0001X110B
- ▲2:IICSn=0001X100B
- ▲3:IICSn=0001XX00B
- △4:IICSn=00000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

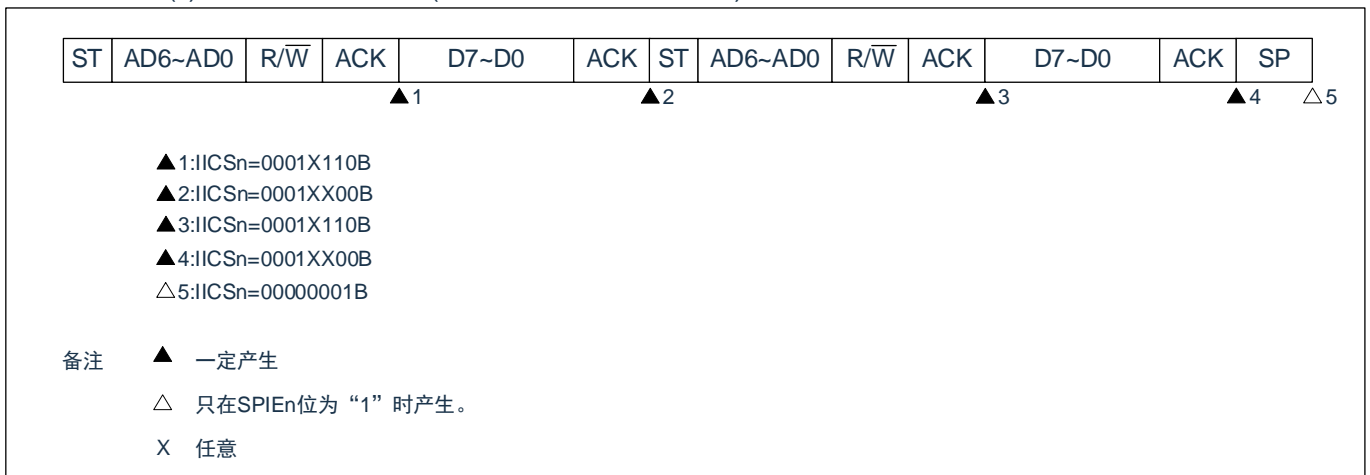
备注: n=0

(b) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况(在重新开始后SVAn相同)



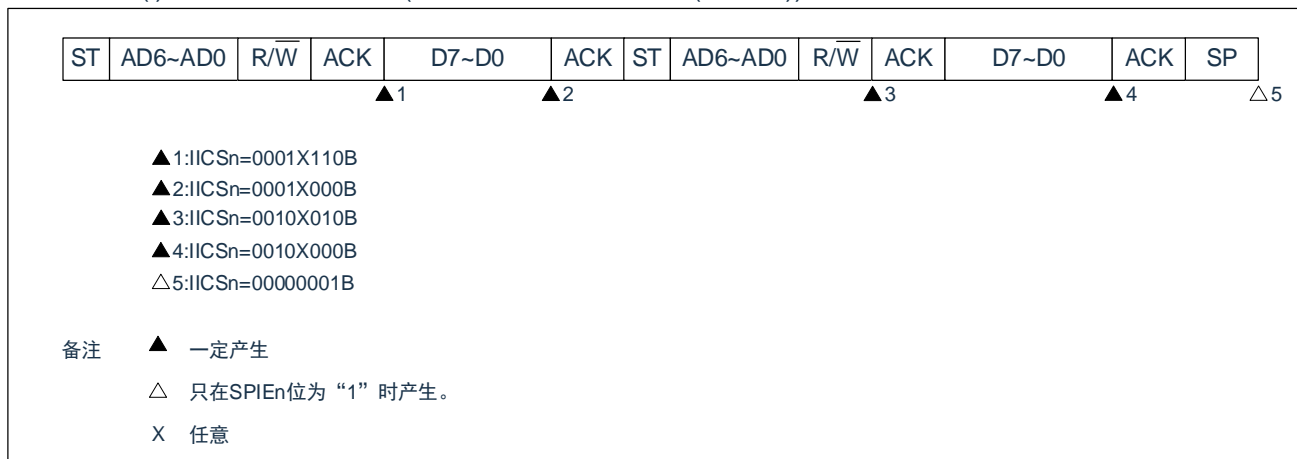
(ii) WTIMn=1的情况(在重新开始后SVAn相同)



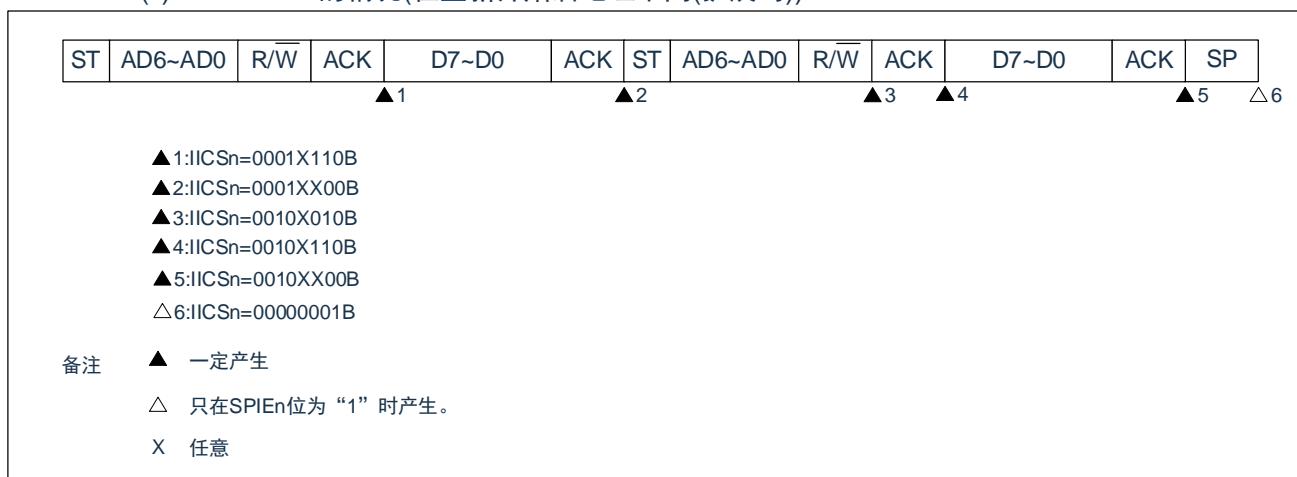
备注: n=0

(c) Start~Address~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况(在重新开始后地址不同(扩展码))



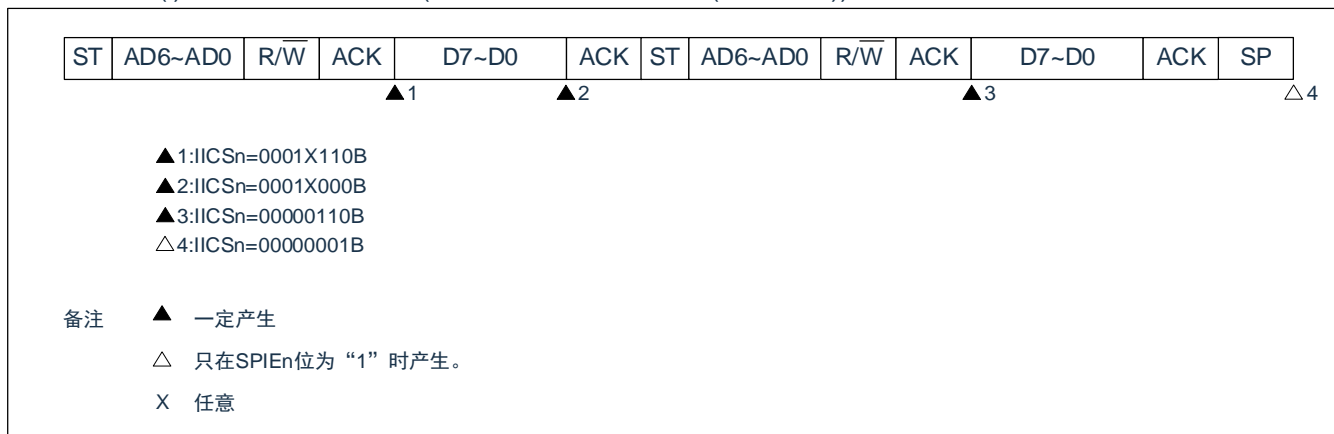
(ii) WTIMn=1的情况(在重新开始后地址不同(扩展码))



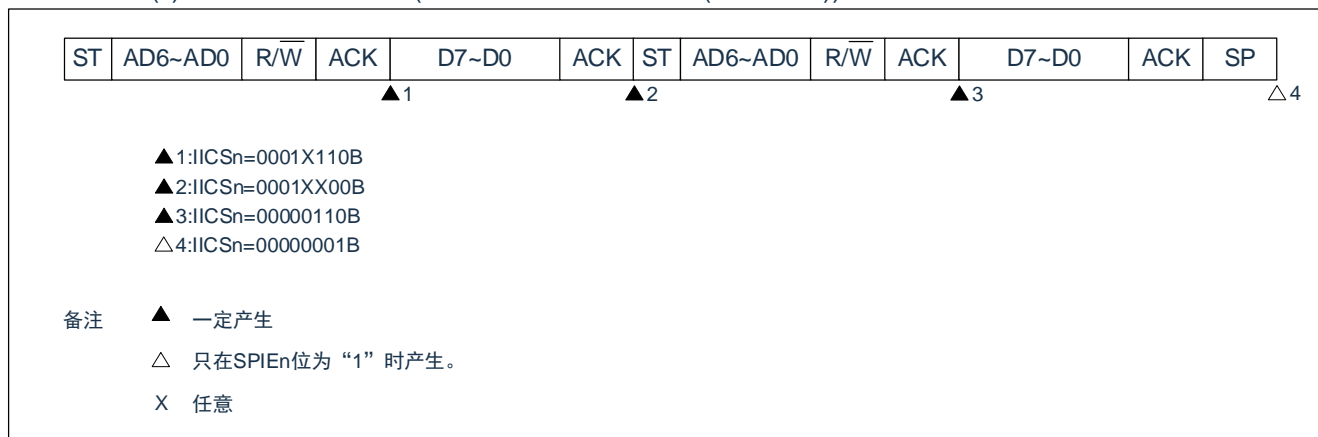
备注: n=0

(d) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况(在重新开始后地址不同(非扩展码))



(ii) WTIMn=1的情况(在重新开始后地址不同(非扩展码))



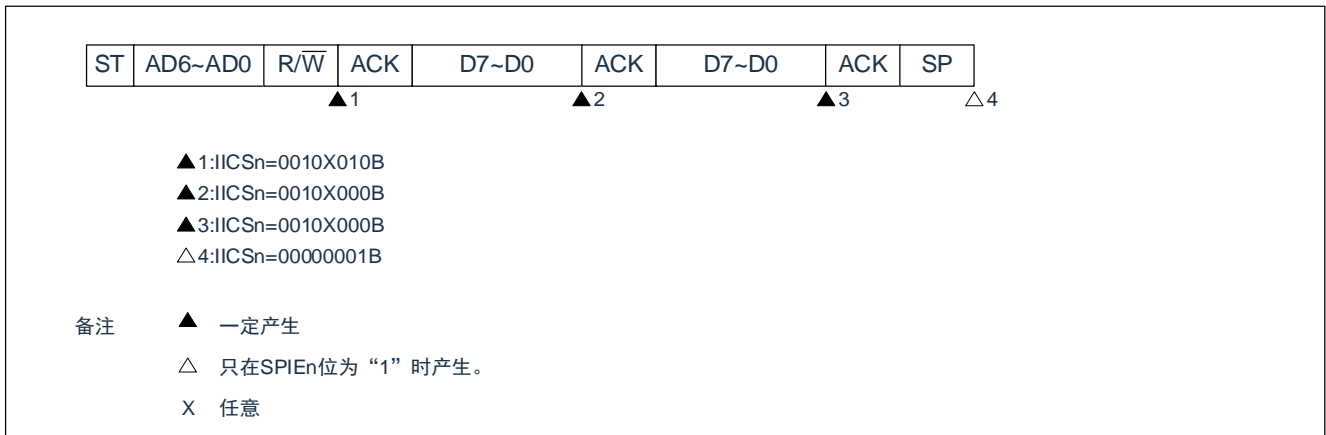
备注: n=0

(3) 从属运行(接收扩展码的情况)

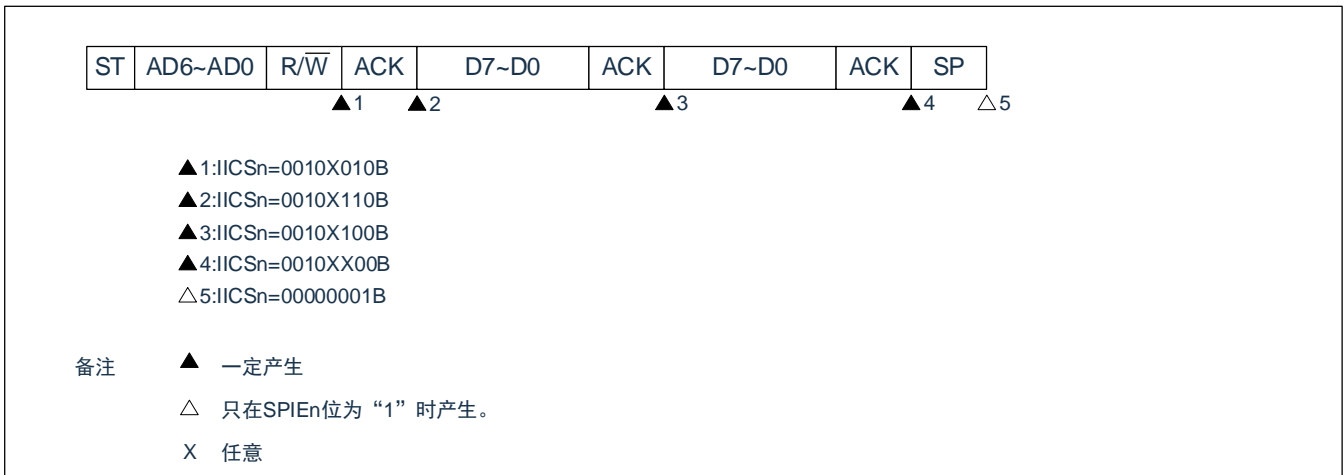
在接收扩展码时，始终参加通信。

(a) Start~Code~Data~Data~Stop

(i) WTIMn=0的情况



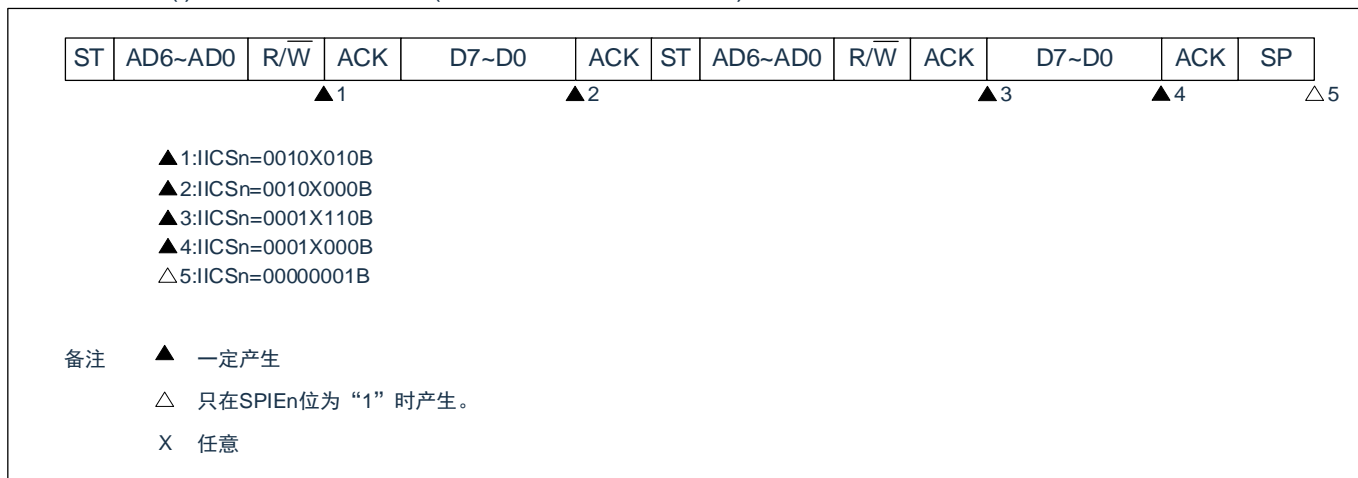
(ii) WTIMn=1的情况



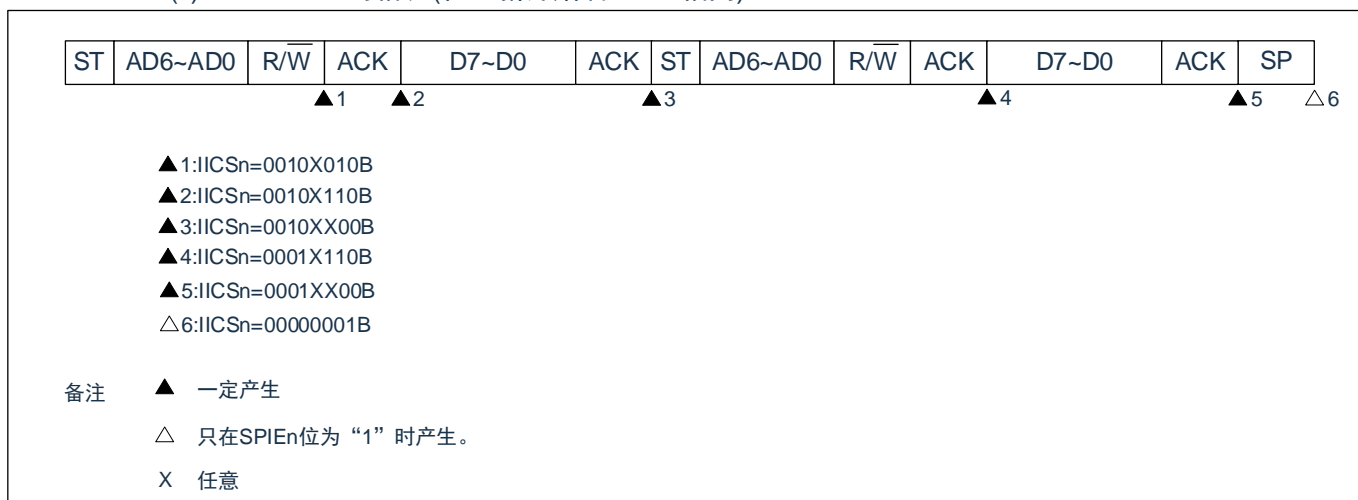
备注：n=0

(b) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况(在重新开始后SVAn相同)



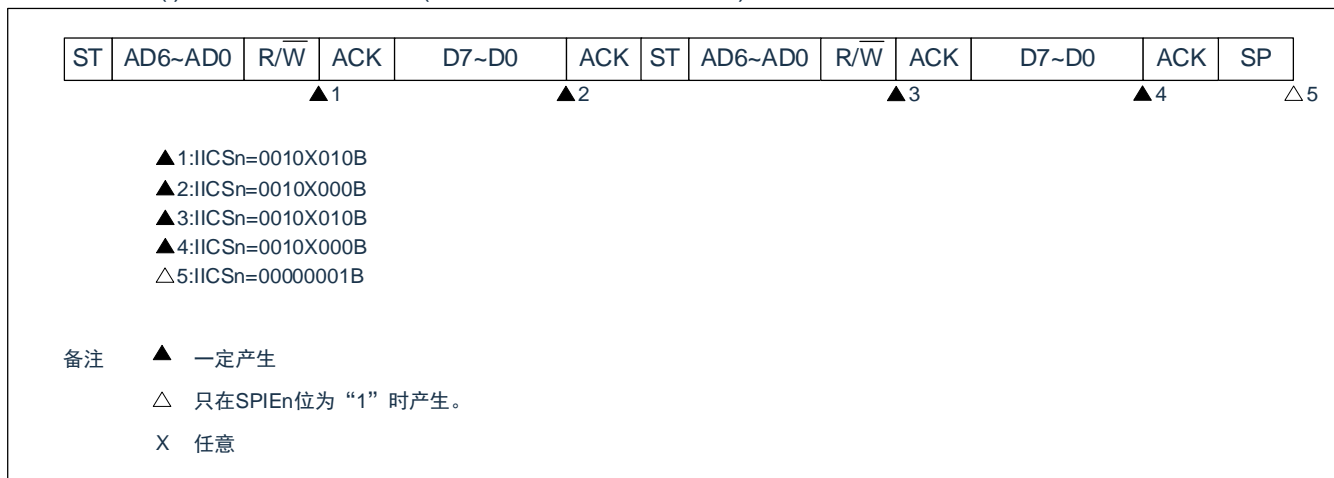
(ii) WTIMn=1的情况(在重新开始后SVAn相同)



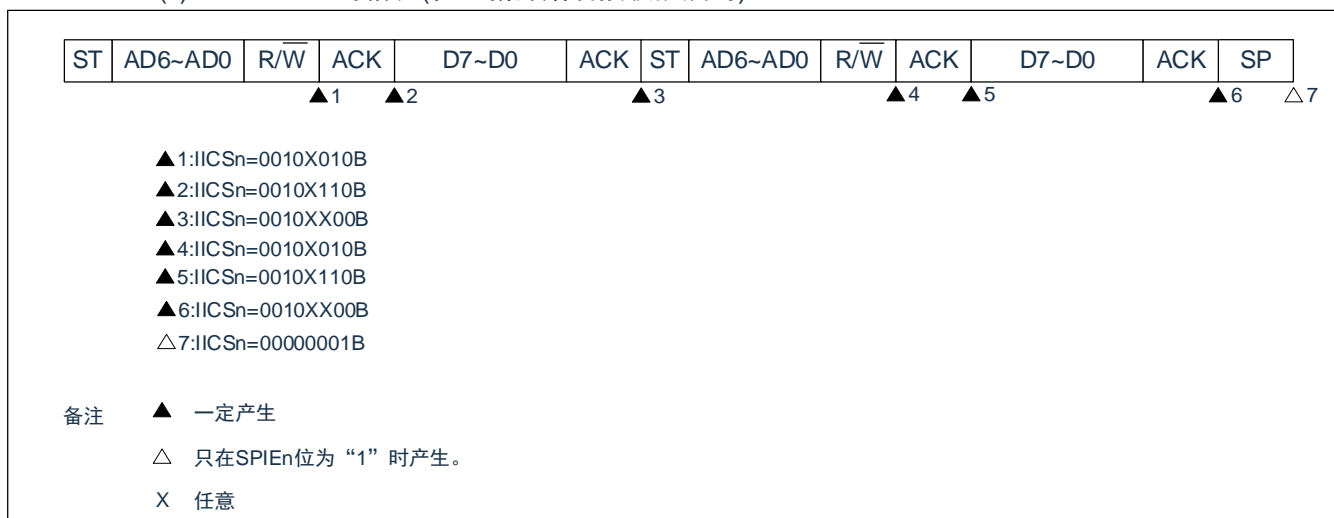
备注: n=0

(c) Start~Code~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况(在重新开始后接收扩展码)



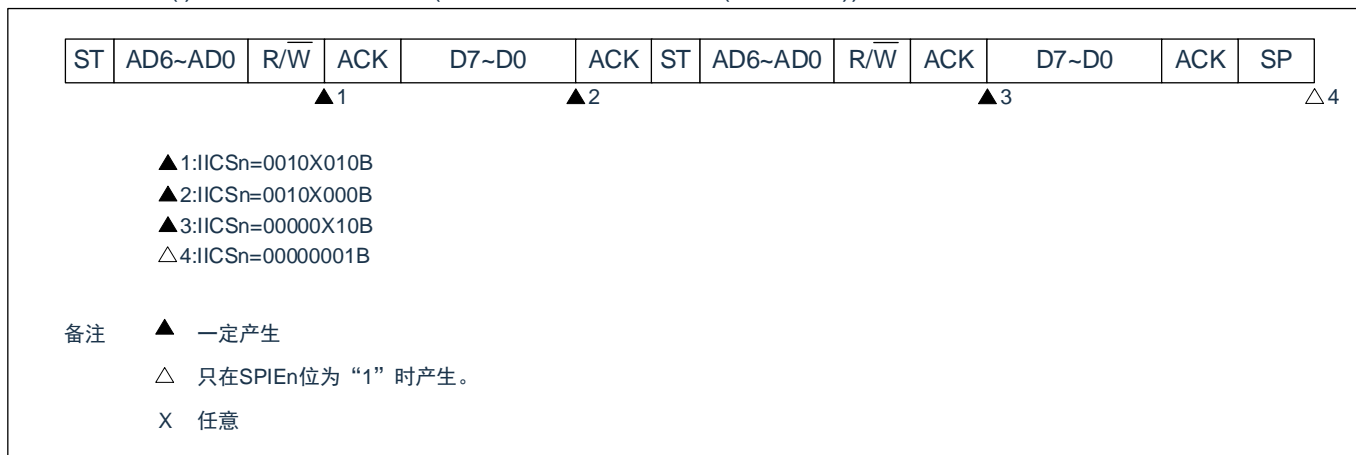
(ii) WTIMn=1的情况(在重新开始后接收扩展码)



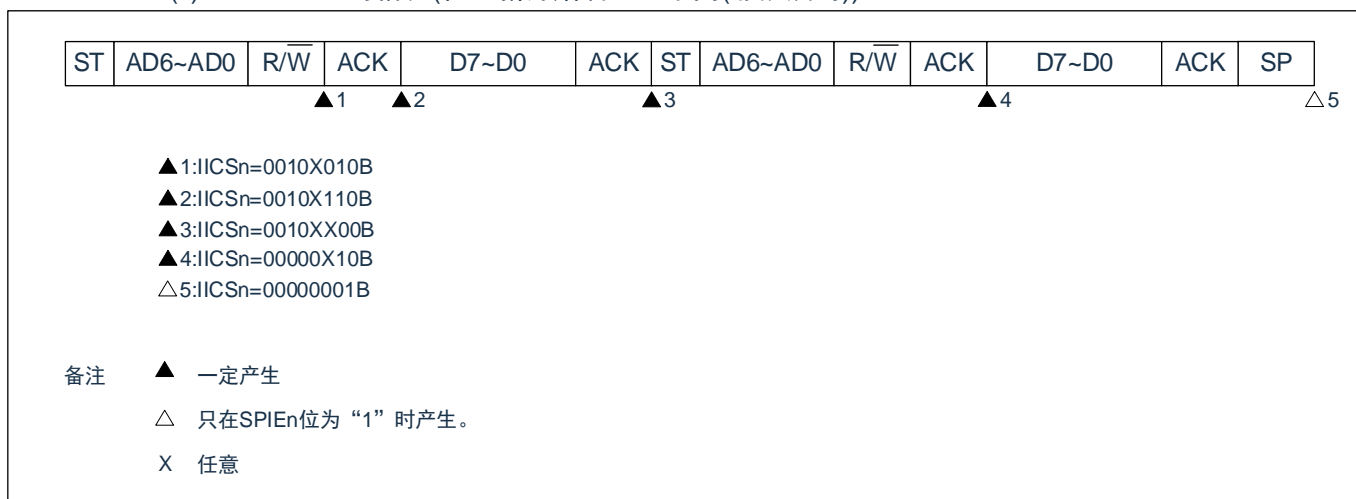
备注: n=0

(d) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况(在重新开始后地址不同(非扩展码))



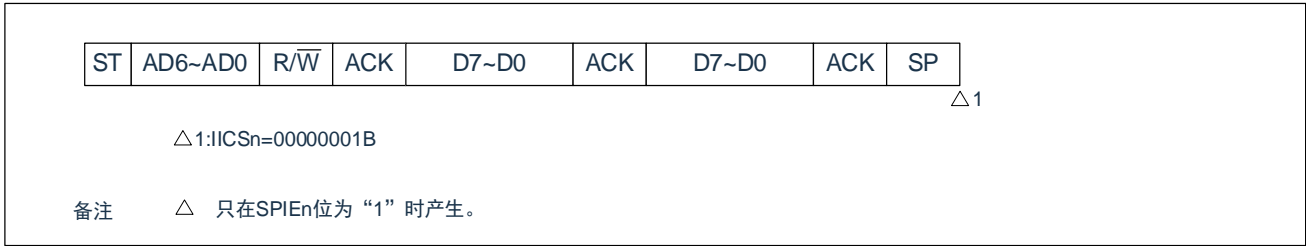
(ii) WTIMn=1的情况(在重新开始后地址不同(非扩展码))



备注: n=0

(4) 不参加通信的运行

(a) Start~Code~Data~Data~Stop

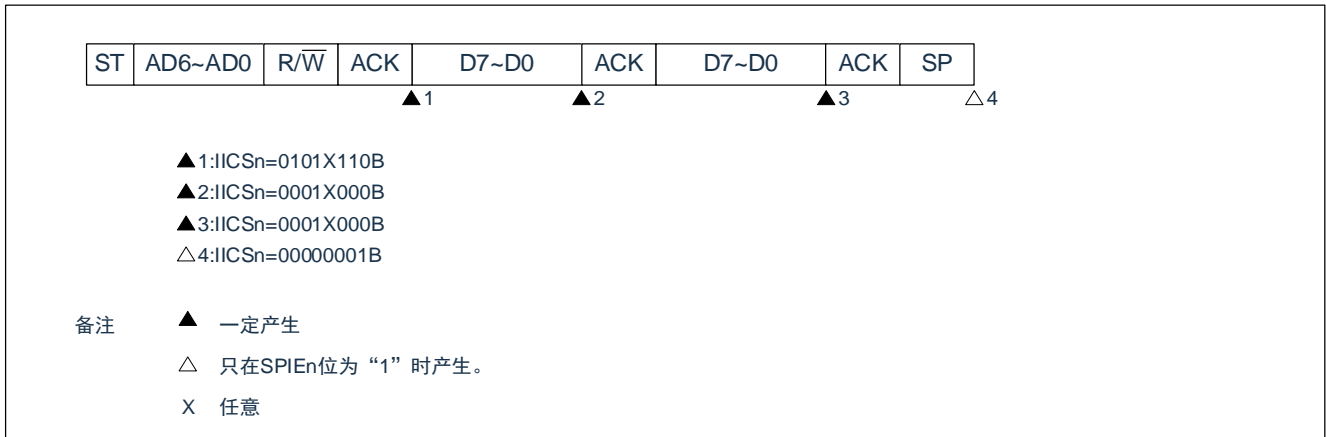


(5) 仲裁失败的运行(在仲裁失败后作为从属设备运行)

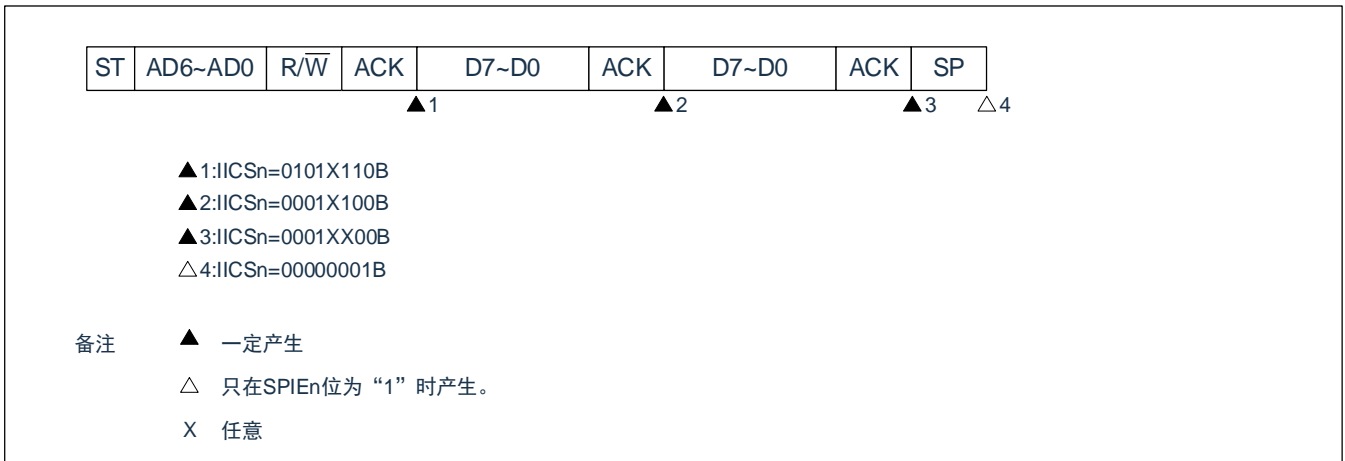
在多主控系统中用作主控设备时，必须在每次产生INTIICAn中断请求信号时读MSTS_n位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况

(i) WTIM_n=0的情况



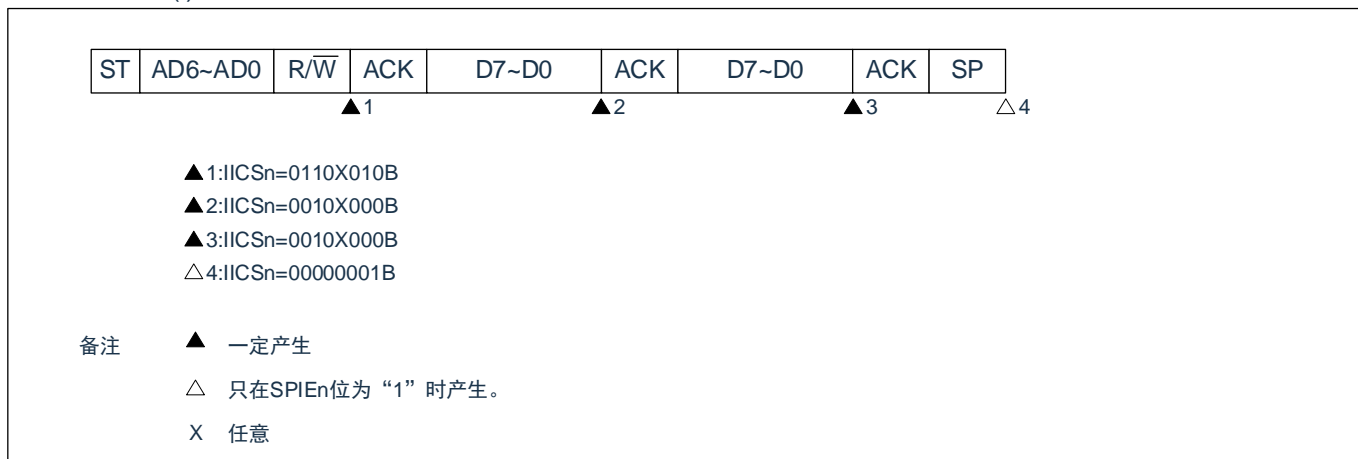
(ii) WTIM_n=1的情况



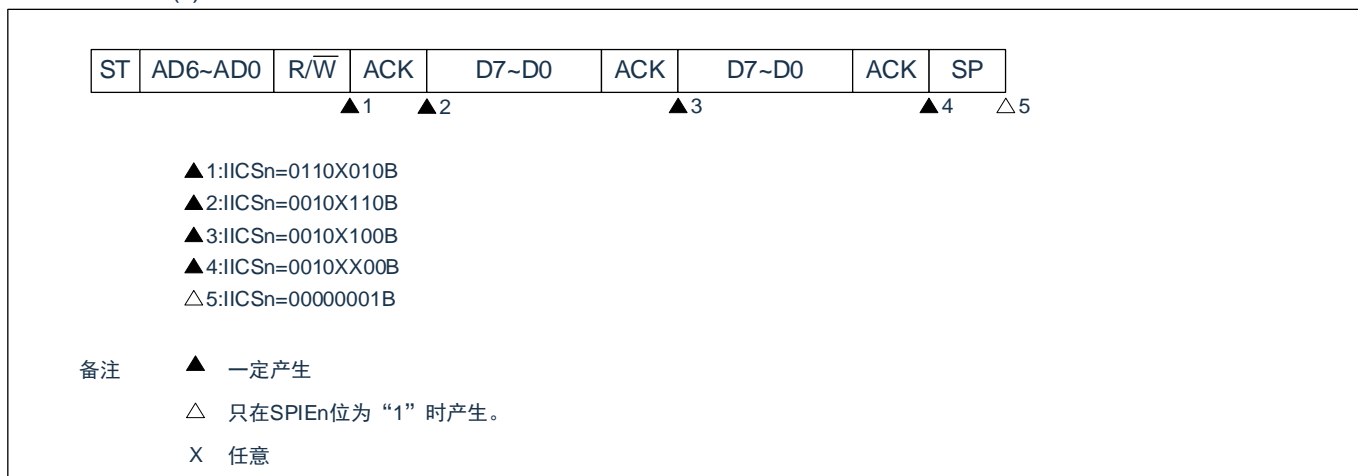
备注：n=0

(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIMn=0的情况



(ii) WTIMn=1的情况

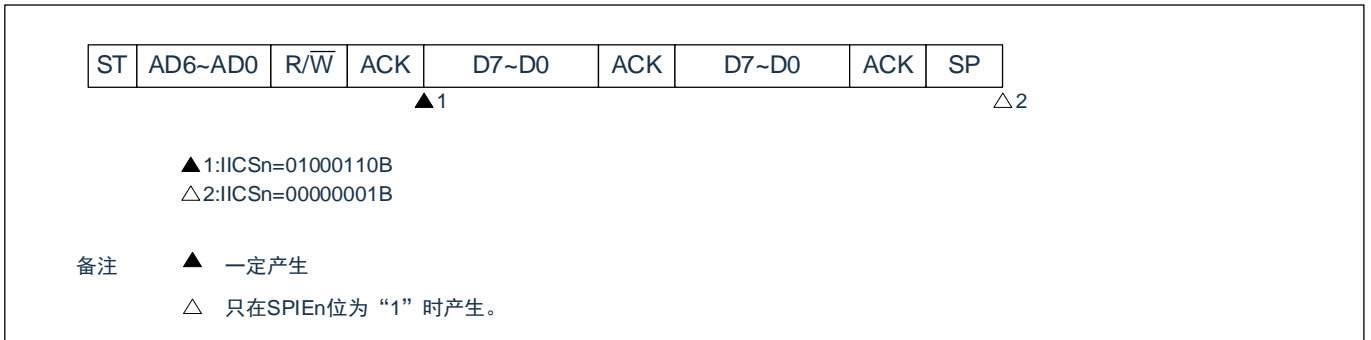


备注：n=0

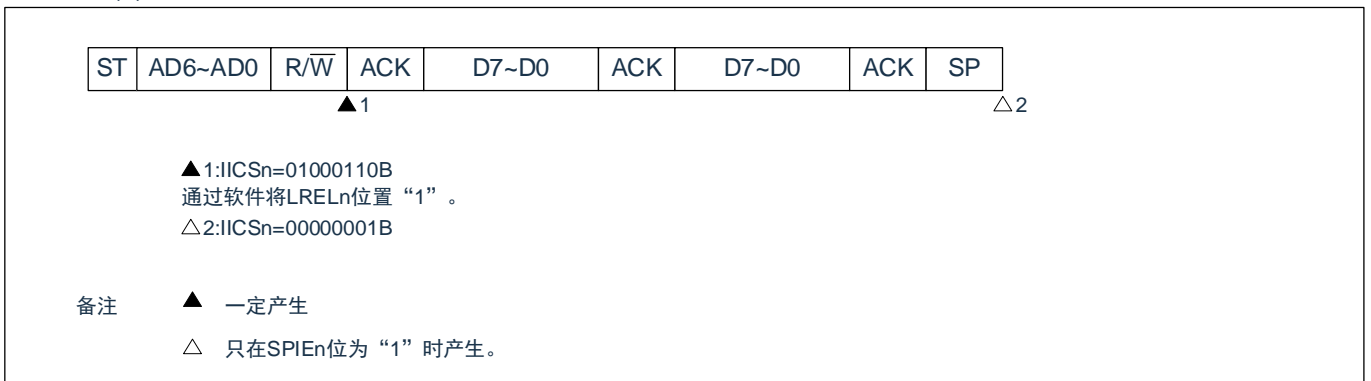
(6) 仲裁失败的运行(在仲裁失败后不参加通信)

在多主控系统中用作主控设备时，必须在每次产生INTIICAn中断请求信号时读MSTS_n位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况(WTIM_n=1)



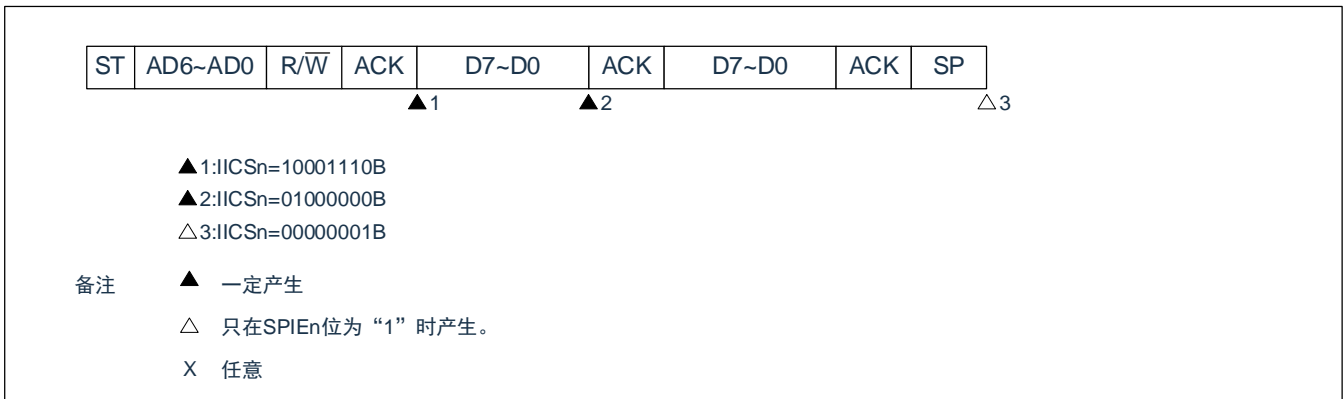
(b) 在发送扩展码的过程中仲裁失败的情况



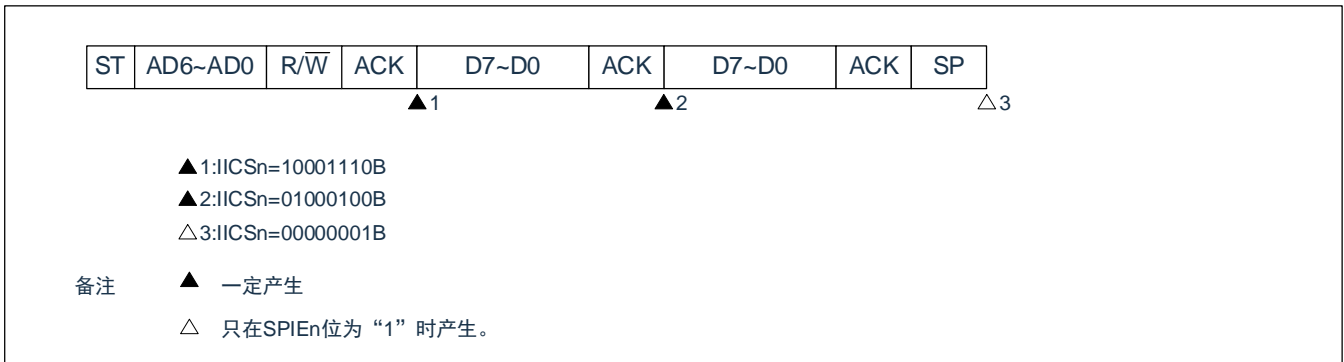
备注：n=0

(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0的情况



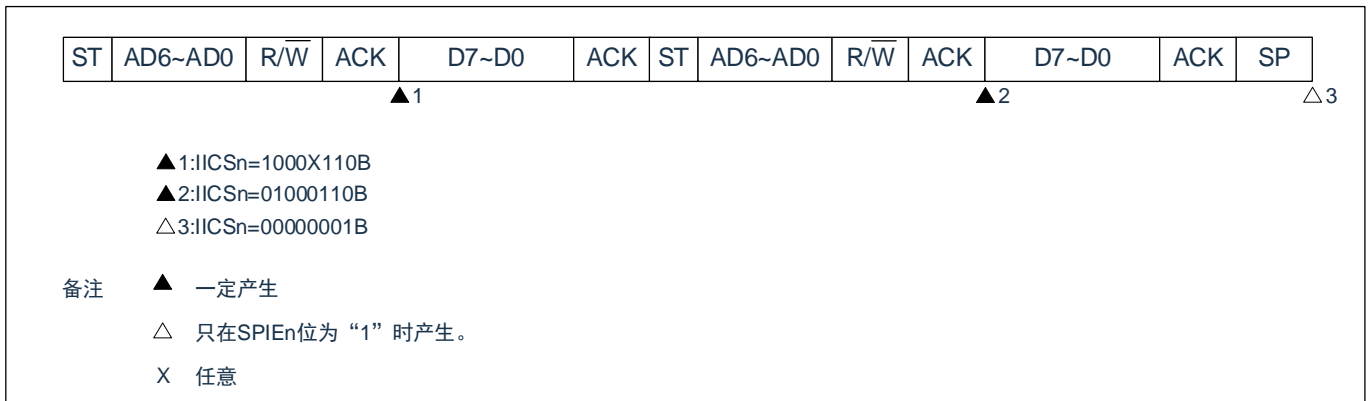
(ii) WTIMn=1的情况



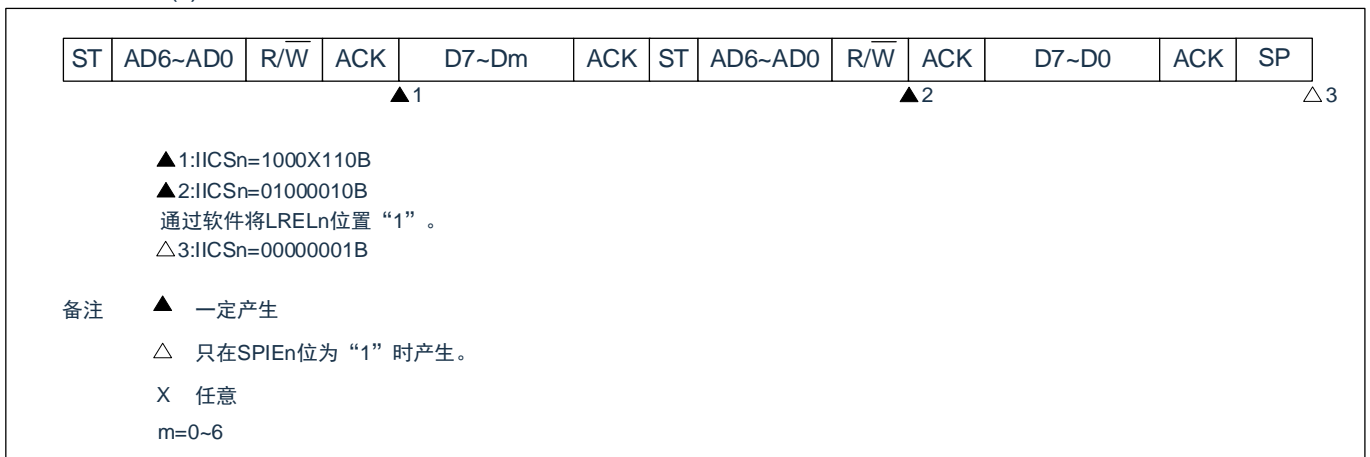
备注：n=0

(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码(例如, SVAn不同)

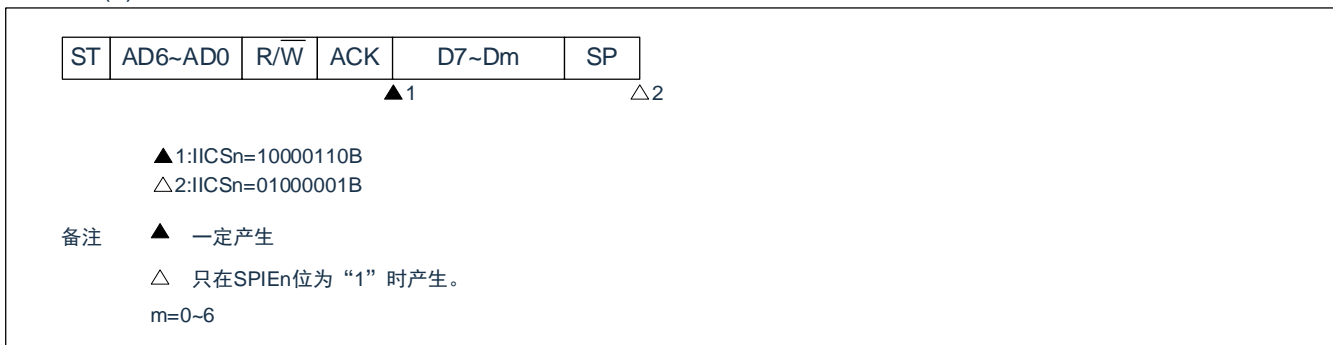


(ii) 扩展码



备注: n=0

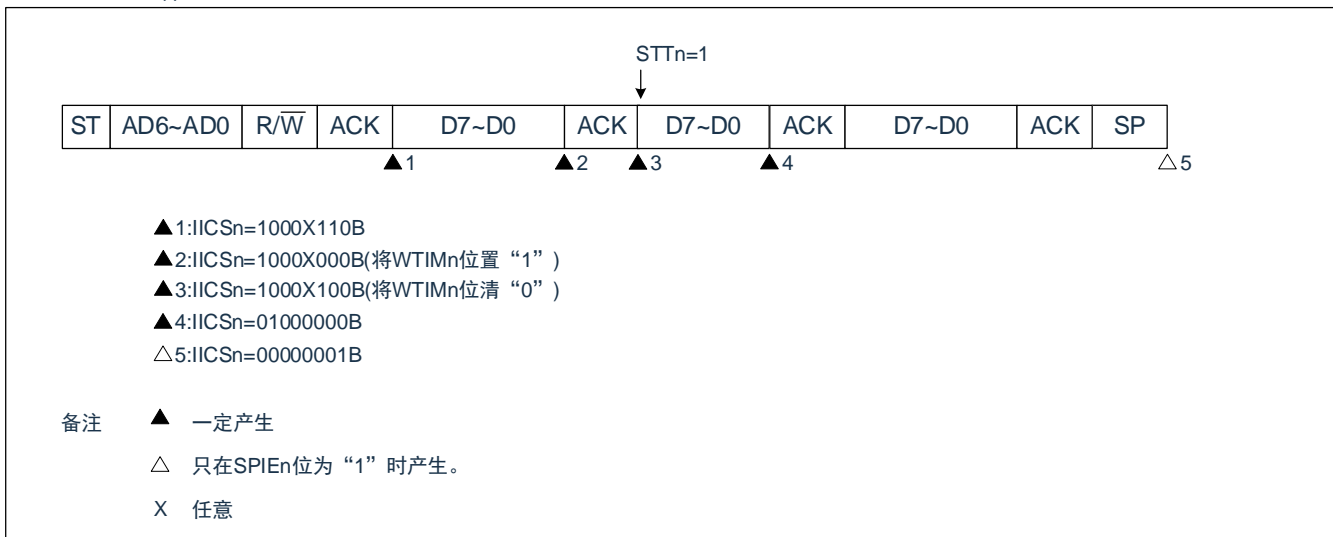
(e) 在传送数据时因停止条件而仲裁失败的情况



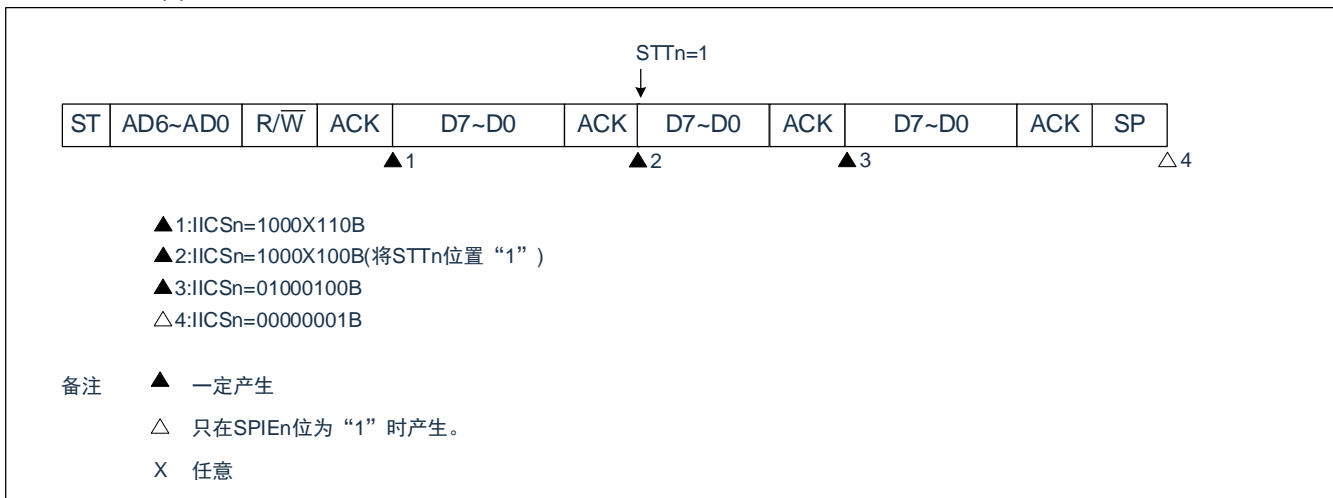
备注：n=0

(f) 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



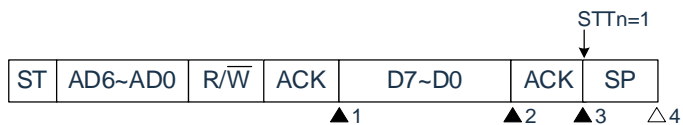
(ii) WTIMn=1的情况



备注：n=0

(g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

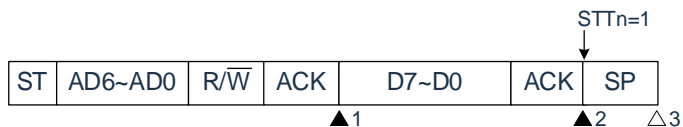
(i) WTIMn=0的情况



- ▲1:IICSn=1000X110B
- ▲2:IICSn=1000X000B(将WTIMn位置“1”)
- ▲3:IICSn=1000XX00B(将STTn位置“1”)
- △4:IICSn=01000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

(ii) WTIMn=1的情况



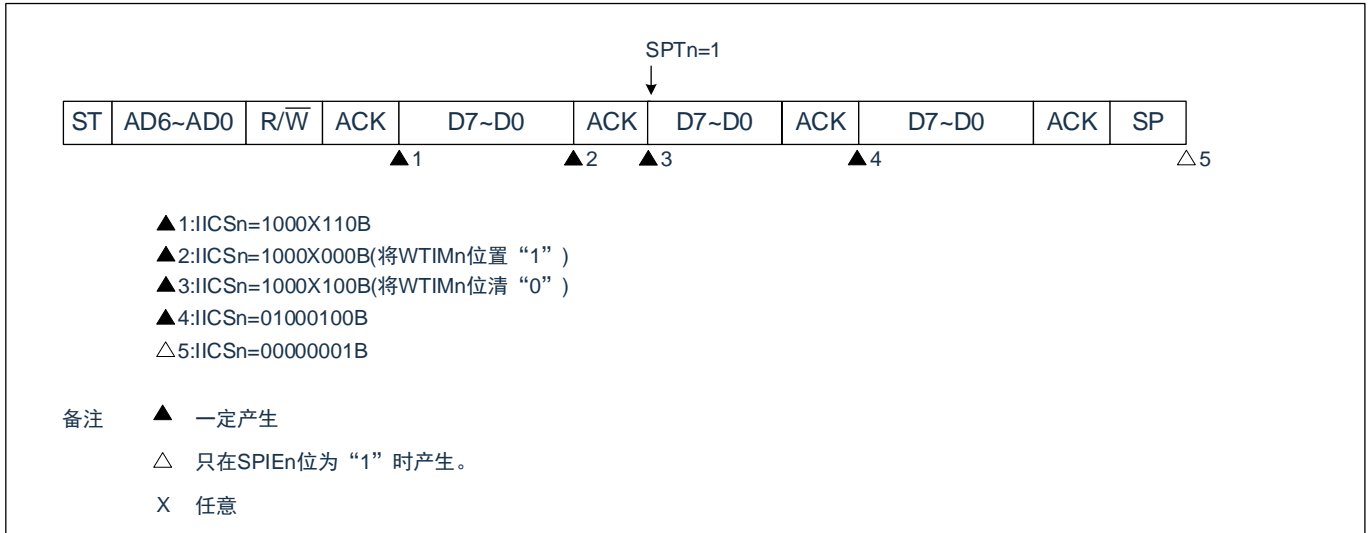
- ▲1:IICSn=1000X110B
- ▲2:IICSn=1000XX00B(将STTn位置“1”)
- △3:IICSn=01000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

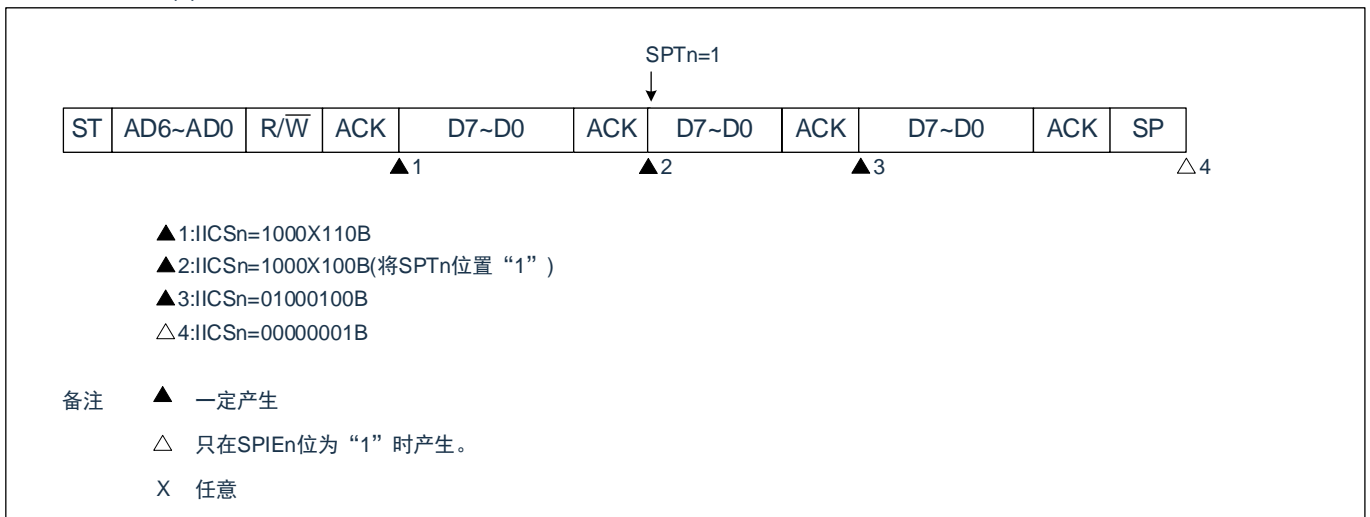
备注: n=0

(h) 在想要生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



(ii) WTIMn=1的情况



备注: n=0

20.6 时序图

在I²C总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。主控设备在从属设备地址之后发送表示数据传送方向的TRCn位(IICA状态寄存器n(IICSn)的bit3)，开始与从属设备进行串行通信。数据通信的时序图如图20-31和图20-32所示。

与串行时钟(SCLAn)的下降沿同步进行IICA移位寄存器n(IICAn)的移位，并且将发送数据传送到SO锁存器，以MSB优先从SDAAn引脚输出数据。

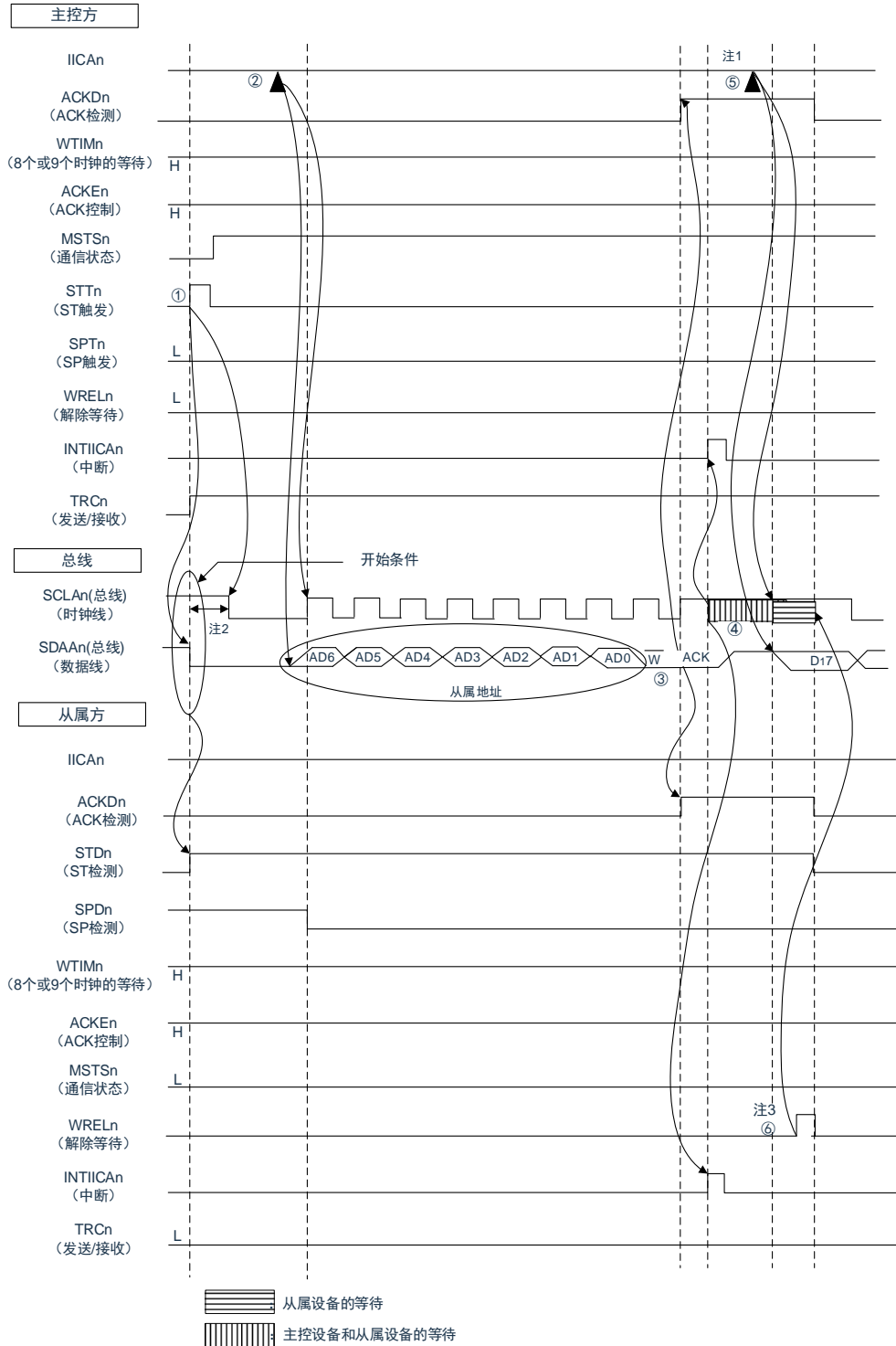
在SCLAn的上升沿将SDAAn引脚输入的数据取到IICAn。

备注：n=0

图20-31：主控设备→从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待)(1/4)

(1) 开始条件~地址~数据



注1：要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

注2：从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。

注3：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图20-31的“(1)开始条件~地址~数据”的①~⑥的说明如下：

- ①如果在主控方将开始条件触发置位(STTn=1)，总线数据线(SDAAn)就下降，生成开始条件(通过SCLAn=1使SDAAn从“1”变为“0”)。此后，如果检测到开始条件，主控方就进入主控通信状态(MSTS_n=1)，在经过保持时间后总线时钟线下降(SCLAn=0)，结束通信准备。
- ②如果主控方给IICA移位寄存器n(IICAn)写地址+W(发送)，就发送从属地址。
- ③在从属方，如果接收地址和本地站地址(SVAn的值)相同^注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ④主控方在第9个时钟的下降沿产生中断(INTIICAn：地址发送结束中断)。相同地址的从属设备进入等待状态(SCLAn=0)，并且产生中断(INTIICAn：地址匹配中断)^注。
- ⑤主控方给IICAn寄存器写发送数据，解除主控方的等待。
- ⑥如果从属方解除等待(WRELn=1)，主控方就开始给从属方传送数据。

注：如果发送的地址和从属地址不同，从属方就不给主控方返回ACK(NACK：SDAAn=1)，并且不产生INTIICAn中断(地址匹配中断)，也不进入等待状态。但是，主控方对于ACK和NACK都产生INTIICAn中断(地址发送结束中断)。

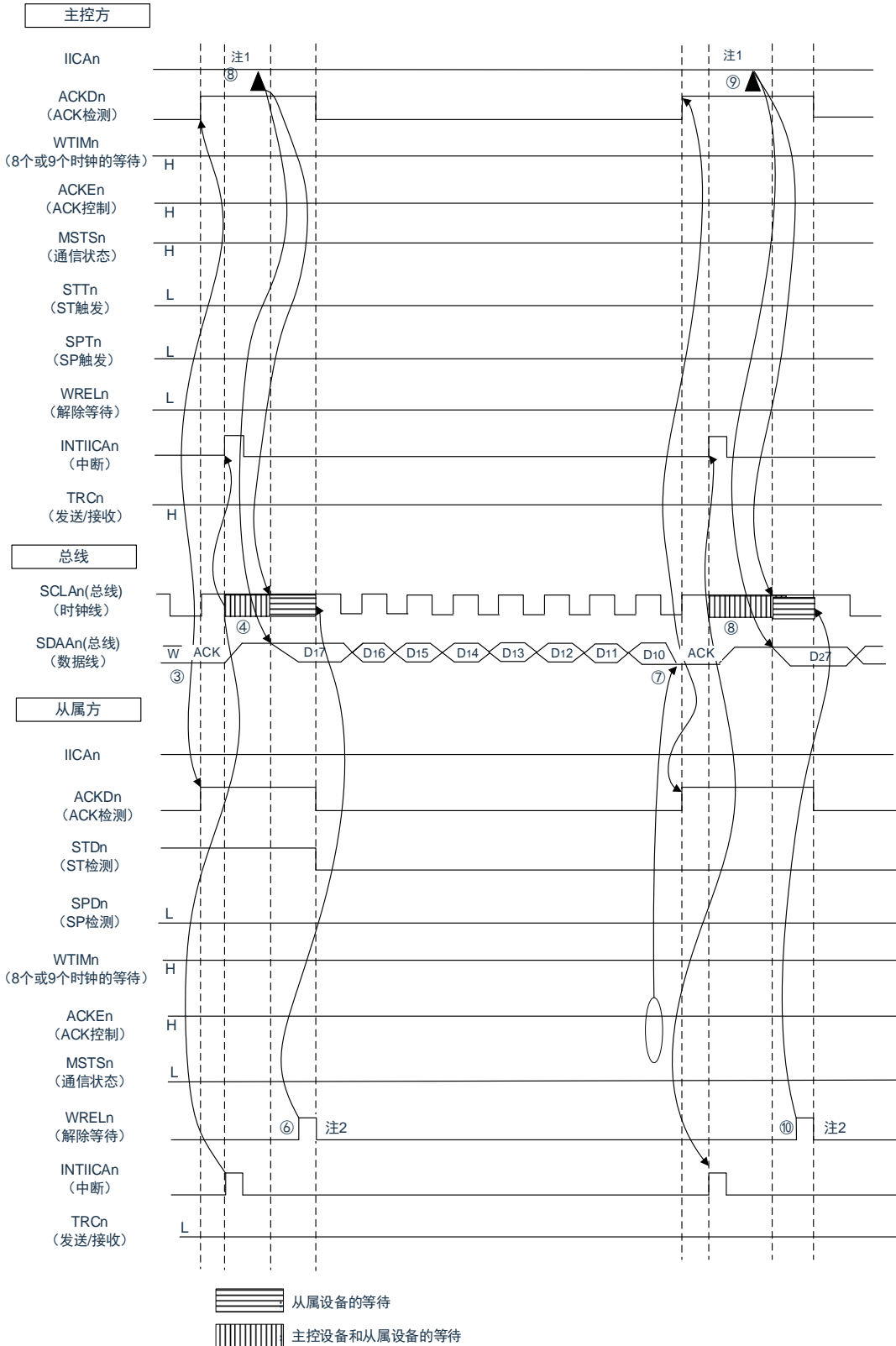
备注：

1. 图20-31的①~⑮是通过I²C总线进行数据通信的一系列运行步骤。
图20-31的“(1)开始条件~地址~数据”说明步骤①~⑥。
图20-31的“(2)地址~数据~数据”说明步骤③~⑩。
图20-31的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图20-31：主控设备→从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待)(2/4)

(2) 地址~数据~数据



注1：要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。
 注2：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图20-31的“(2)地址~数据~数据”的③~⑩的说明如下:

- ③在从属方, 如果接收地址和本地站地址(SVAn的值)相同^注, 就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ④主控方在第9个时钟的下降沿产生中断(INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 地址匹配中断)^注。
- ⑤主控方给IICA移位寄存器n(IICAn)写发送数据, 解除主控方的等待。
- ⑥如果从属方解除等待(WRELn=1), 主控方就开始给从属方传送数据。
- ⑦在数据传送结束后, 因为从属方的ACKEn位为“1”, 所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ⑧主控方和从属方在第9个时钟的下降沿进入等待状态(SCLAn=0), 并且都产生中断(INTIICAn: 传送结束中断)。
- ⑨主控方给IICAn寄存器写发送数据, 解除主控方的等待。
- ⑩如果从属方读接收数据并且解除等待(WRELn=1), 主控方就开始给从属方传送数据。

注: 如果发送的地址和从属地址不同, 从属方就不给主控方返回ACK(NACK: SDAAn=1), 并且不产生INTIICAn中断(地址匹配中断), 也不进入等待状态。但是, 主控方对于ACK和NACK都产生INTIICAn中断(地址发送结束中断)。

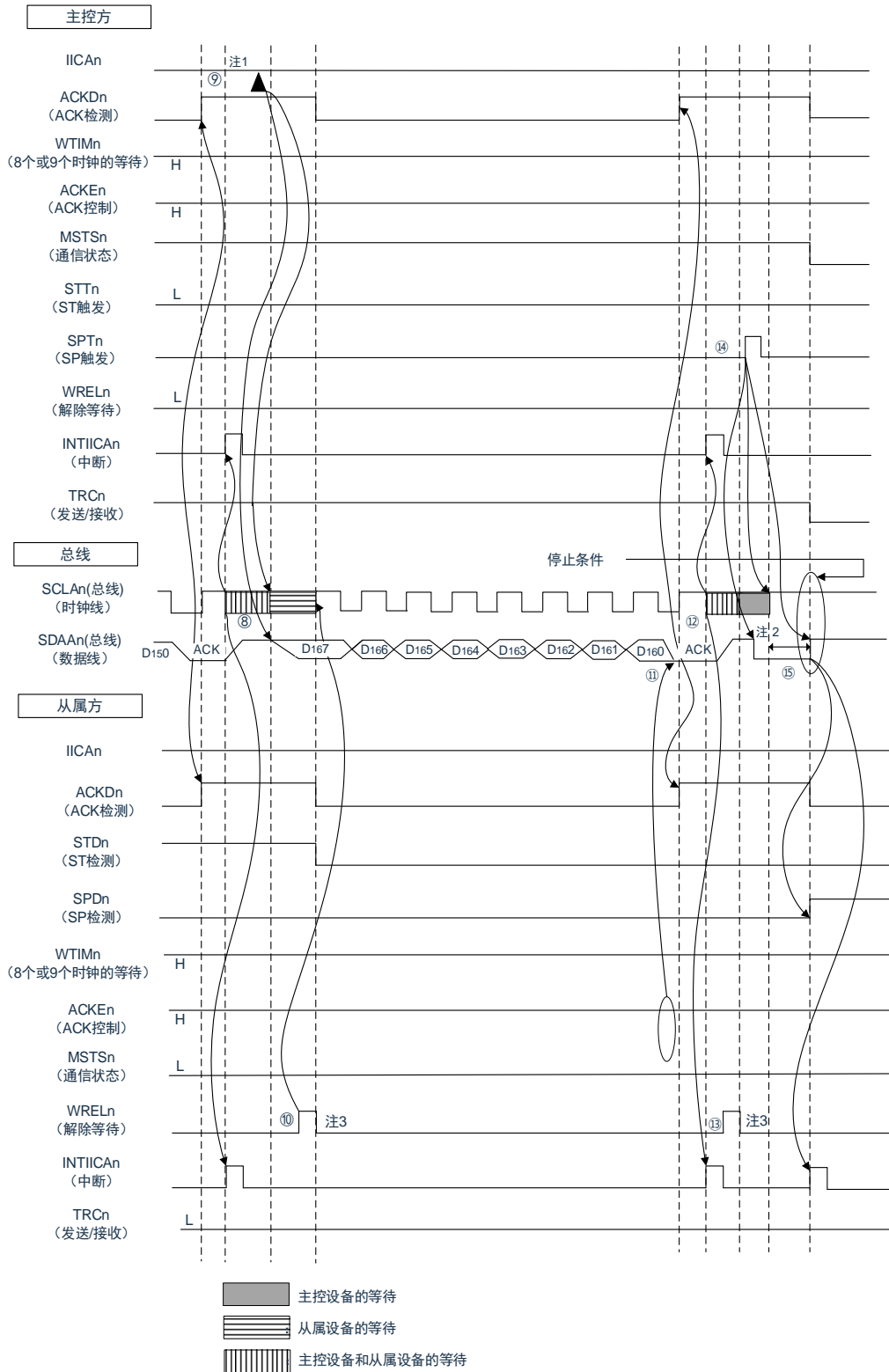
备注:

1. 图20-31的①~⑮是通过I²C总线进行数据通信的一系列运行步骤。
图20-31的“(1)开始条件~地址~数据”说明步骤①~⑥。
图20-31的“(2)地址~数据~数据”说明步骤③~⑩。
图20-31的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图20-31：主控设备→从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待)(3/4)

(3) 数据~数据~停止条件



注1：要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

注2：在发行停止条件后，从SCLAn引脚信号上升到生成停止条件的的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。

注3：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图20-31的“(3)数据~数据~停止条件”的⑦~⑮的说明如下：

- ⑦在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ⑧主控方和从属方在第9个时钟的下降沿进入等待状态(SCLAn=0)，并且都产生中断(INTIICAn：传送结束中断)。
- ⑨主控方给IICA移位寄存器n(IICAn)写发送数据，解除主控方的等待。
- ⑩如果从属方读接收数据并且解除等待(WRELn=1)，主控方就开始给从属方传送数据。
- ⑪在数据传送结束后，从属方(ACKEn=1)通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ⑫主控方和从属方在第9个时钟的下降沿进入等待状态(SCLAn=0)，并且都产生中断(INTIICAn：传送结束中断)。
- ⑬从属方读接收数据，解除等待(WRELn=1)。
- ⑭如果在主控方将停止条件触发置位(SPTn=1)，就清除总线数据线(SDAAn=0)并且将总线时钟线置位(SCLAn=1)，在经过停止条件的准备时间后将总线数据线置位(SDAAn=1)，生成停止条件(通过SCLAn=1使SDAAn从“0”变为“1”)。
- ⑮如果生成停止条件，从属方就检测到停止条件并且产生中断(INTIICAn：停止条件中断)。

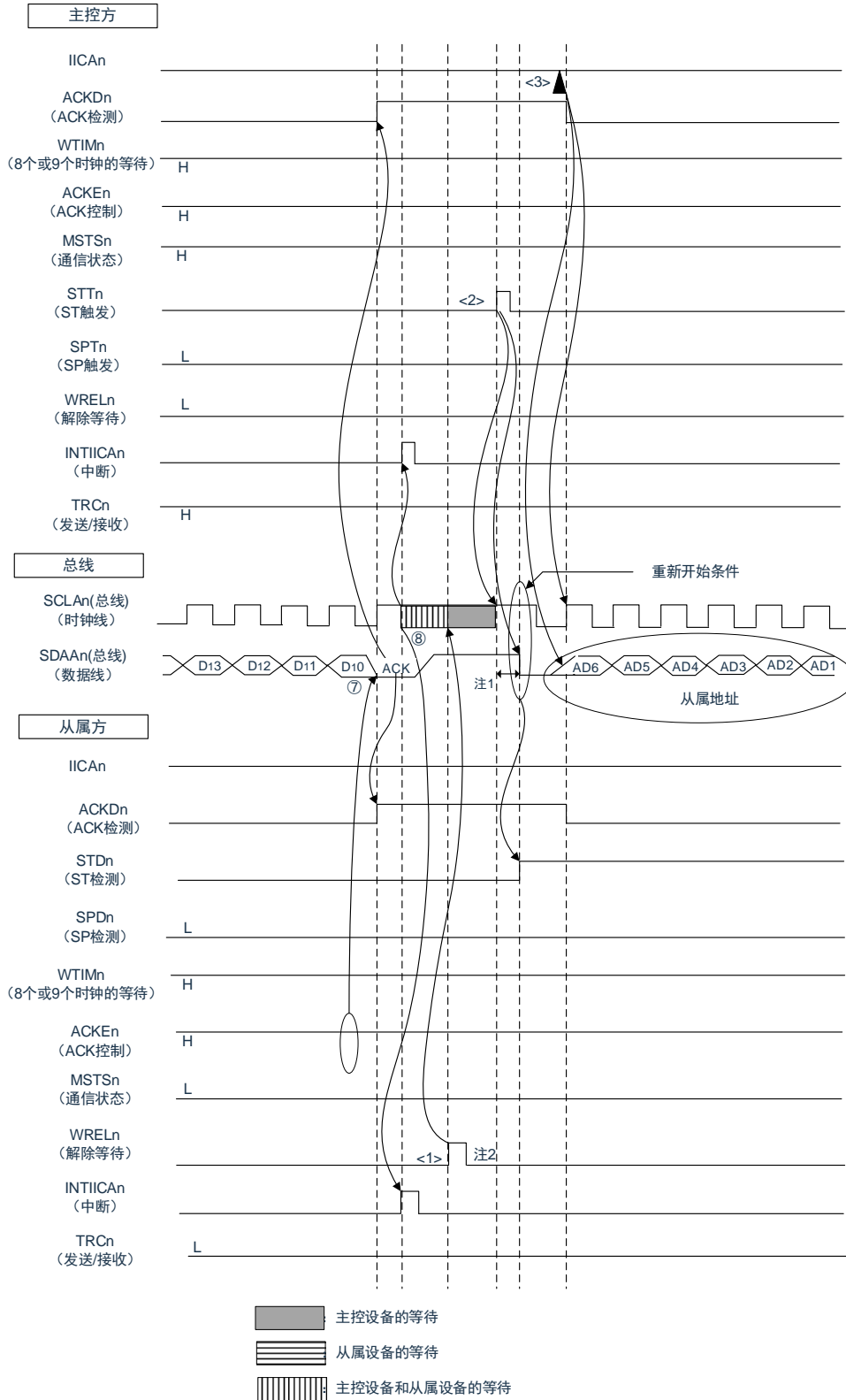
备注：

1. 图20-31的①~⑮是通过I²C总线进行数据通信的一系列运行步骤。
 图20-31的“(1)开始条件~地址~数据”说明步骤①~⑥。
 图20-31的“(2)地址~数据~数据”说明步骤③~⑩。
 图20-31的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图20-31：主控设备→从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待)(4/4)

(4) 数据~重新开始条件~地址



注1：在发行重新开始条件后，从SCLAn引脚信号上升到生成开始条件的时间，在设置为标准模式时至少为4.7us，在设置为快速模式时至少为0.6us。

注2：要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图20-31的“(4)数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行<1>~<3>，从而返回到步骤③的数据发送步骤。

⑦在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。

⑧主控方和从属方在第9个时钟的下降沿进入等待状态(SCLAn=0)，并且都产生中断(INTIICAn：传送结束中断)。

<1>从属方读接收数据，解除等待(WRELn=1)。

<2>如果在主控方再次将开始条件触发置位(STTn=1)，总线时钟线就上升(SCLAn=1)，而且在经过重新开始条件的准备时间后总线数据线下降(SDAAn=0)，生成开始条件(通过SCLAn=1使SDAAn从“1”变为“0”)。然后，如果检测到开始条件，就在经过保持时间后总线时钟线下降(SCLAn=0)，结束通信准备。

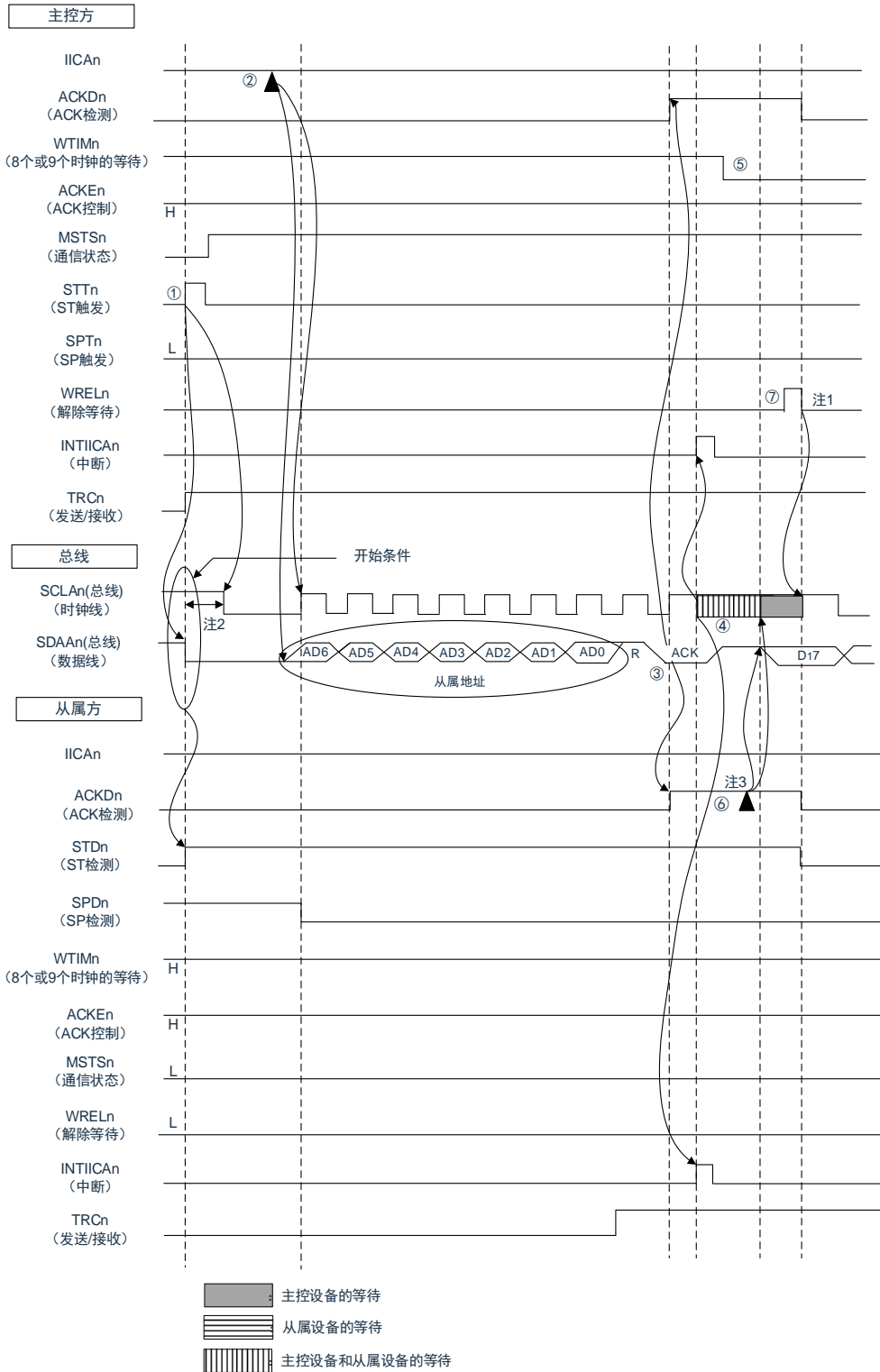
<3>如果主控方给IICA移位寄存器n(IICAn)写地址+R/W(发送)，就发送从属地址。

备注：n=0

图20-32：从属设备→主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待)(1/3)

(1) 开始条件~地址~数据



注1：要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注2：从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。

注3：要解除从属方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

图20-32的“(1)开始条件~地址~数据”的①~⑦的说明如下：

- ①如果在主控方将开始条件触发置位(STTn=1)，总线数据线(SDAAn)就下降，生成开始条件(通过SCLAn=1使SDAAn从“1”变为“0”)。此后，如果检测到开始条件，主控方就进入主控通信状态(MSTS_n=1)，在经过保持时间后总线时钟线下降(SCLAn=0)，结束通信准备。
- ②如果主控方给IICA移位寄存器n(IICAn)写地址+R(接收)，就发送从属地址。
- ③在从属方，如果接收地址和本地站地址(SVAn的值)相同^注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ④主控方在第9个时钟的下降沿产生中断(INTIICAn：地址发送结束中断)。相同地址的从属设备进入等待状态(SCLAn=0)，并且产生中断(INTIICAn：地址匹配中断)^注。
- ⑤主控方将等待时序改为第8个时钟(WTIMn=0)。
- ⑥从属方给IICAn寄存器写发送数据，解除从属方的等待。
- ⑦主控方解除等待(WRELn=1)，开始来自从属设备的数据传送。

注：如果发送的地址和从属地址不同，从属方就不给主控方返回ACK(NACK：SDAAn=1)，并且不产生INTIICAn中断(地址匹配中断)，也不进入等待状态。但是，主控方对于ACK和NACK都产生INTIICAn中断(地址发送结束中断)。

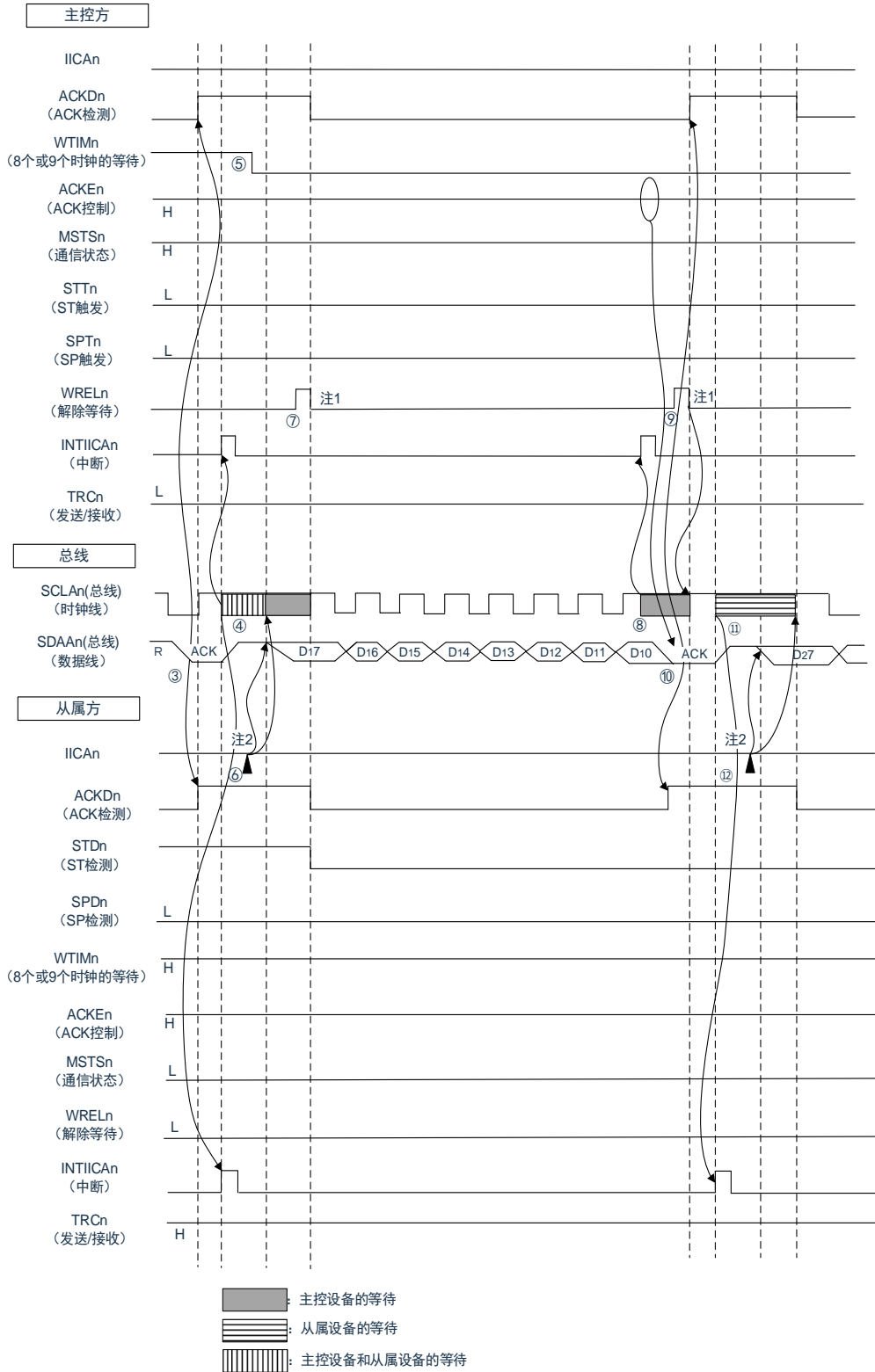
备注：

1. 图20-32的①~⑱是通过I²C总线进行数据通信的一系列运行步骤。
 图20-32的“(1)开始条件~地址~数据”说明步骤①~⑦。
 图20-32的“(2)地址~数据~数据”说明步骤③~⑫。
 图20-32的“(3)数据~数据~停止条件”说明步骤⑧~⑱。
2. n=0

图20-32：从属设备→主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待)(2/3)

(2) 地址~数据~数据



注1：要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注2：要解除从属方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

图20-32的“(2)地址~数据~数据”的③~⑫的说明如下:

- ③在从属方, 如果接收地址和本地站地址(SVAn的值)相同^注, 就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ④主控方在第9个时钟的下降沿产生中断(INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 地址匹配中断)^注。
- ⑤主控方将等待时序改为第8个时钟(WTIMn=0)。
- ⑥从属方给IICA移位寄存器n(IICAn)写发送数据, 解除从属方的等待。
- ⑦主控方解除等待(WRELn=1), 开始来自从属设备的数据传送。
- ⑧主控方在第8个时钟的下降沿进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 传送结束中断)。因为主控方的ACKEn位为“1”, 所以通过硬件给从属方发送ACK。
- ⑨主控方读接收数据, 解除等待(WRELn=1)。
- ⑩从属方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ⑪从属方在第9个时钟的下降沿进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 传送结束中断)。
- ⑫如果从属方给IICAn寄存器写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。

注: 如果发送的地址和从属地址不同, 从属方就不给主控方返回ACK(NACK: SDAA_n=1), 并且不产生INTIICAn中断(地址匹配中断), 也不进入等待状态。但是, 主控方对于ACK和NACK都产生INTIICAn中断(地址发送结束中断)。

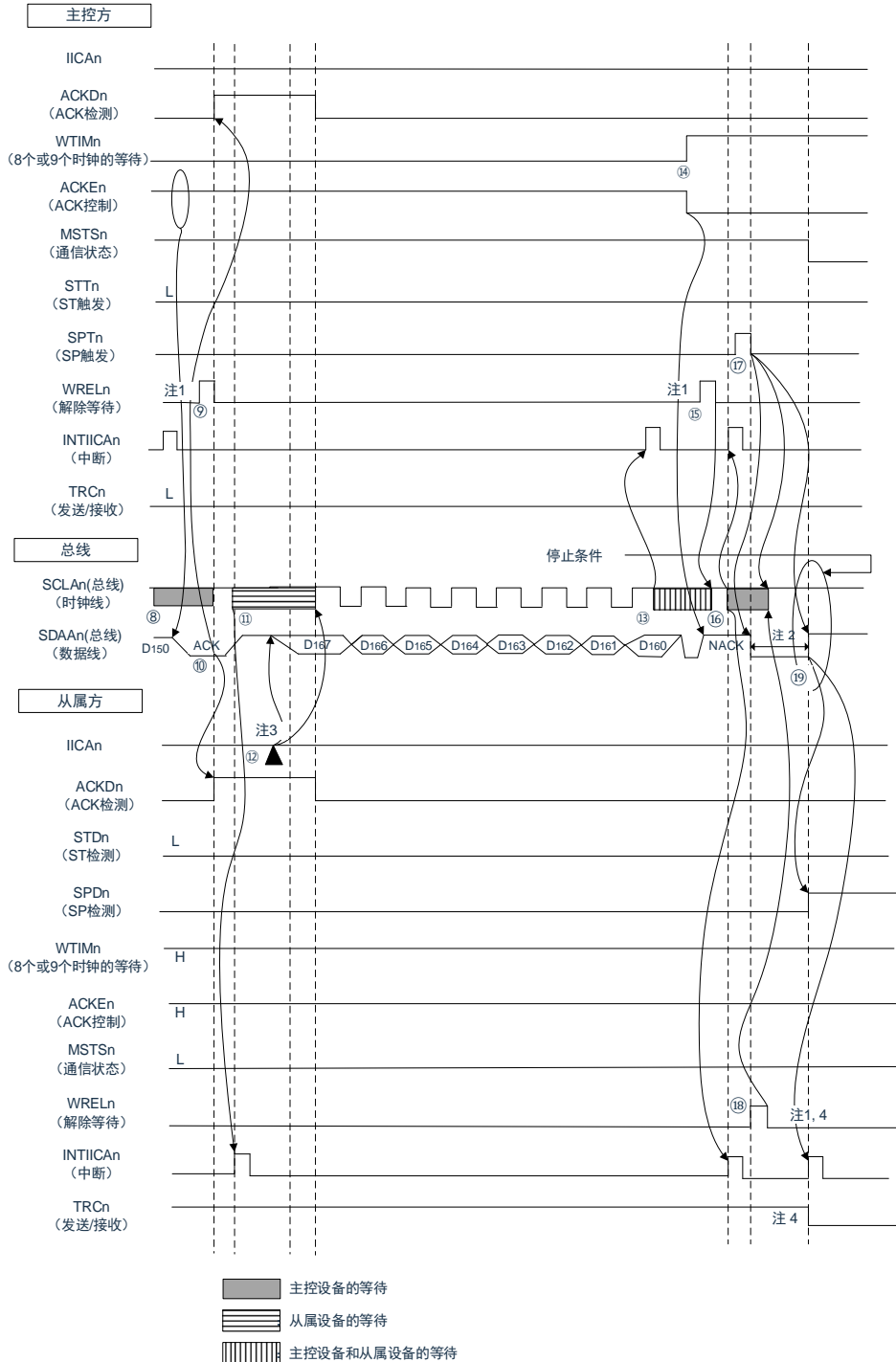
备注:

1. 图20-32的①~⑰是通过I²C总线进行数据通信的一系列运行步骤。
 图20-32的“(1)开始条件~地址~数据”说明步骤①~⑦。
 图20-32的“(2)地址~数据~数据”说明步骤③~⑫。
 图20-32的“(3)数据~数据~停止条件”说明步骤⑧~⑱。
2. n=0

图20-32: 从属设备→主控设备的通信例子

(主控设备: 选择8个→9个时钟的等待, 从属设备: 选择9个时钟的等待)(3/3)

(3) 数据~数据~停止条件



注1: 要解除等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注2: 在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的时间, 在设置为标准模式时至少为4.0us, 在设置为快速模式时至少为0.6us。

注3: 要解除从属方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注4: 在从属方的发送期间, 如果通过WRELn位的置位来解除等待, 就清除TRCn位。

图20-32的“(3)数据~数据~停止条件”的⑧~⑱的说明如下:

- ⑧主控方在第8个时钟的下降沿进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 传送结束中断)。因为主控方的ACKEn位为“0”, 所以通过硬件给从属方发送ACK。
- ⑨主控方读接收数据, 解除等待(WRELn=1)。
- ⑩从属方在第9个时钟的上升沿检测到ACK(ACKDn=1)。
- ⑪从属方在第9个时钟的下降沿进入等待状态(SCLAn=0), 并且产生中断(INTIICAn: 传送结束中断)。
- ⑫如果从属方给IICA移位寄存器n(IICAn)写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。
- ⑬主控方在第8个时钟的下降沿产生中断(INTIICAn: 传送结束中断), 并且进入等待状态(SCLAn=0)。因为进行ACK控制(ACKEn=1), 所以此阶段的总线数据线变为低电平(SDAAn=0)。
- ⑭主控方设置为NACK应答(ACKEn=0), 并且将等待时序改为第9个时钟(WTIMn=1)。如果主控方解除等待(WRELn=1), 从属方就在第9个时钟的上升沿检测到NACK(ACKDn=0)。
- ⑮主控方和从属方在第9个时钟的下降沿进入等待状态(SCLAn=0), 并且都产生中断(INTIICAn: 传送结束中断)。
- ⑯如果主控方发行停止条件(SPTn=1), 就清除总线数据线(SDAAn=0), 并且解除主控方的等待。此后, 主控方处于待机状态, 直到将总线时钟线置位(SCLAn=1)为止。
- ⑰从属方在确认NACK后停止发送, 为了结束通信, 解除等待(WRELn=1)。如果解除从属方的等待, 就将总线时钟线置位(SCLAn=1)。
- ⑱如果主控方确认到总线时钟线被置位(SCLAn=1), 就在经过停止条件准备时间后将总线数据线置位(SDAAn=1), 然后发行停止条件(通过SCLAn=1使SDAAn从“0”变为“1”)。如果生成停止条件, 从属方就检测到停止条件, 并且产生中断(INTIICAn: 停止条件中断)。

第21章 CAN 控制器

21.1 概要描述

这款芯片具有片上CAN控制器(控制器局域网)功能,并且符合ISO 11898中标准的CAN协议.

21.1.1 特征

- 符合 ISO 11898并且按照 ISO/DIS 16845 (CAN符合性)来测试
- 使用标准帧和扩展帧来实现接收和发送
- 通信速度:最大1 Mbps. (CAN输入时钟 ≥ 8 MHz)
- 1个通道有16个报文缓存
- 接收/发送历史列表功能
- 自动块传输功能
- 多缓存接收块功能
- 每个通道四种模式的屏蔽设置

21.1.2 功能概述

表21-1列出了CAN控制器的功能。

表21-1：功能概述

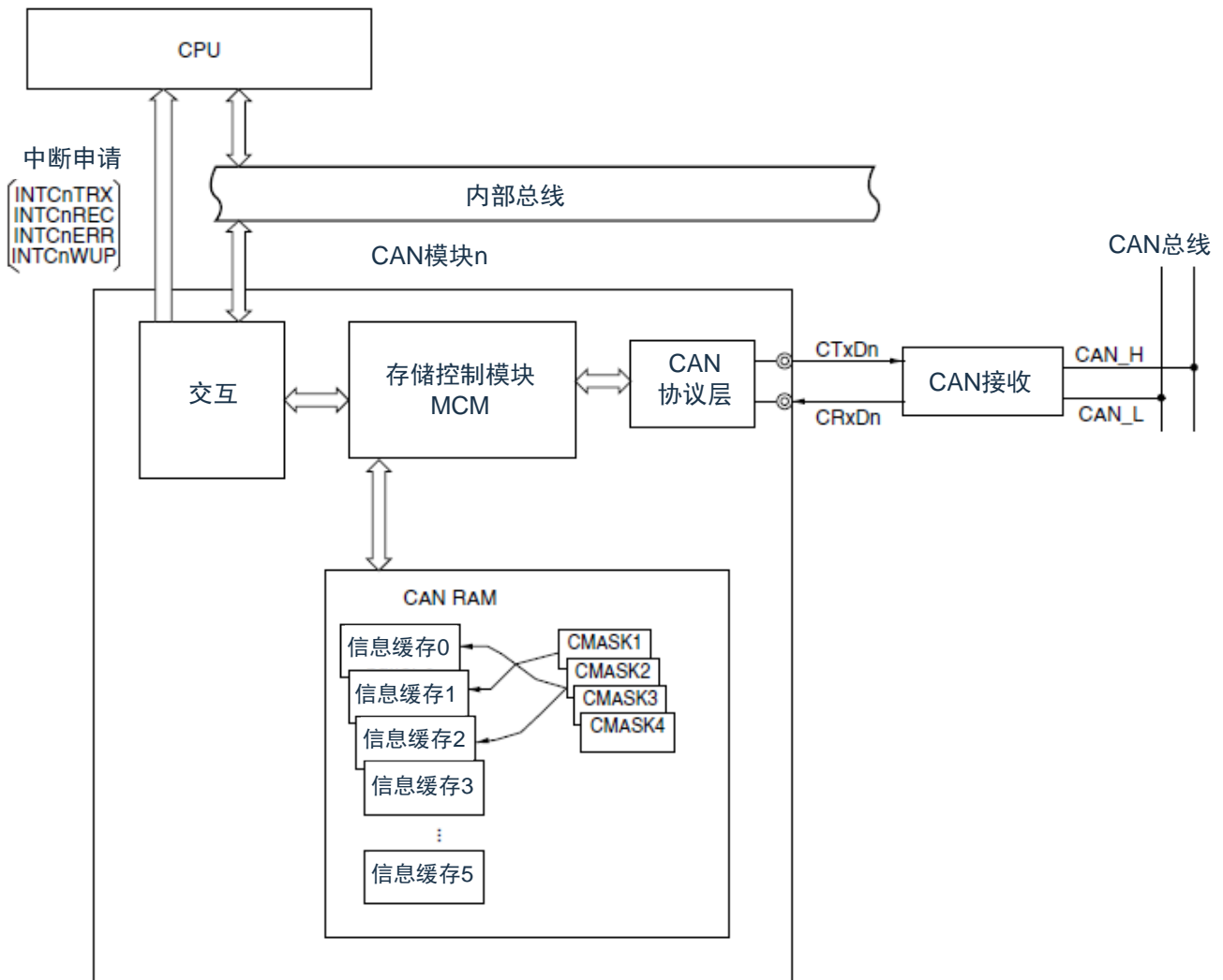
功能	细节
协议	ISO 11898 CAN协议 (标准和扩展帧发送/接收)
波特率	最大： 1 Mbps (CAN 输入时钟 ≥ 8MHz)
数据储存	信息储存在CAN RAM
信息数	- 一个通道 16 个报文缓存 - 每一个报文缓存既可以被设置成接收报文缓存也能够被设置成发送报文缓存
报文接收	- Unique ID 可以被设置到每一个报文缓存. - 每个通道有四种 Mask 设置 - 每次报文被接收储存在到报文缓存后，都会生成一个接收中断。 - 两个以上的接收报文缓存能被用于作为接收 FIFO(多缓存接收块功能) - 接收历史列表功能
报文发送	- Unique ID 可以被设置到每一个报文缓存 - 每个缓存都能关联到发送 完成中断 - 报文缓冲区 0 到 7 被指定为传输报文缓冲区，可用于自动块传输。报文传输间隔是可编程的(自动块传输功能(以下简称"ABT") - 传输历史列表功能
远程帧处理	远程帧处理通过传输报文缓存
时间戳功能	- 当使用 16 位计时器时，可以为报文接收设置时间戳功能。 时间戳捕获触发是可以选择的(可以侦测报文帧的 SOF 或 EOF)
诊断功能	- 可读的错误计数器 - 用于验证总线连接的"有效协议操作标志" - 只收模式 - 单次模式 - CAN 协议错误类型的解码 - 自检模式
从总线关闭状态释放	- 软件可以设置从总线关闭状态中强制释放(忽略时序约束) - 不从总线关闭状态自动释放(必须重新启用软件设置)。
省电模式	- CAN 睡眠模式 (可以被 CAN 总线唤醒) - CAN 停止模式(不能被 CAN 总线唤醒)

21.1.3 配置

CAN控制器由下面四个模块组成

- (1) 交互
这个模块提供内部的交互总线,在CAN模块和主机之间发送和接收信号。
- (2) 内存控制模块(MCM)
此功能块控制对 CAN 协议层和 CAN 模块中的 CAN RAM 的访问。
- (3) CAN协议层
此功能块涉及 CAN 协议的操作及其相关设置。
- (4) CAN RAM
这是 CAN 内存功能块, 用于存储报文ID、报文数据等。

图21-1: CAN模块n的框图



备注: n = 0

21.2 CAN协议

CAN(控制器局域网)是一种高速多路通信协议，用于汽车应用(C类)的实时通信。CAN由ISO11898规定。有关详细信息，请参阅ISO11898规范。

CAN规范通常分为两层：物理层和数据链接层，反过来数据链接层包含逻辑链接和介质访问控制。这些层的组成如下图所示：

图21-2：层的组成



注1：CAN控制器规范

21.2.1 帧格式

(1) 标准帧格式

- 标准帧格式使用11位标识符，这意味着它可以处理2048条信号。

(2) 扩展格式帧

- 扩展格式帧使用29位(11位 + 18位)识别符，这可以处理2048 x 218条信号。
- 在仲裁字段中为 SRR 和 IDE 位设置"隐性电平"(CMOS 电平等于"1")时设置扩展格式帧。

21.2.2 帧类型

下表中的四个帧类型运用在CAN协议中。

表21-2: 帧类型

帧类型	描述
数据帧	用于传输数据的帧
远程帧	去申请数据帧的信号
错误帧	用于报告错误的信号
过载帧	用于设置一段延时来等待下一个数据帧或远程帧

(3) 总线值

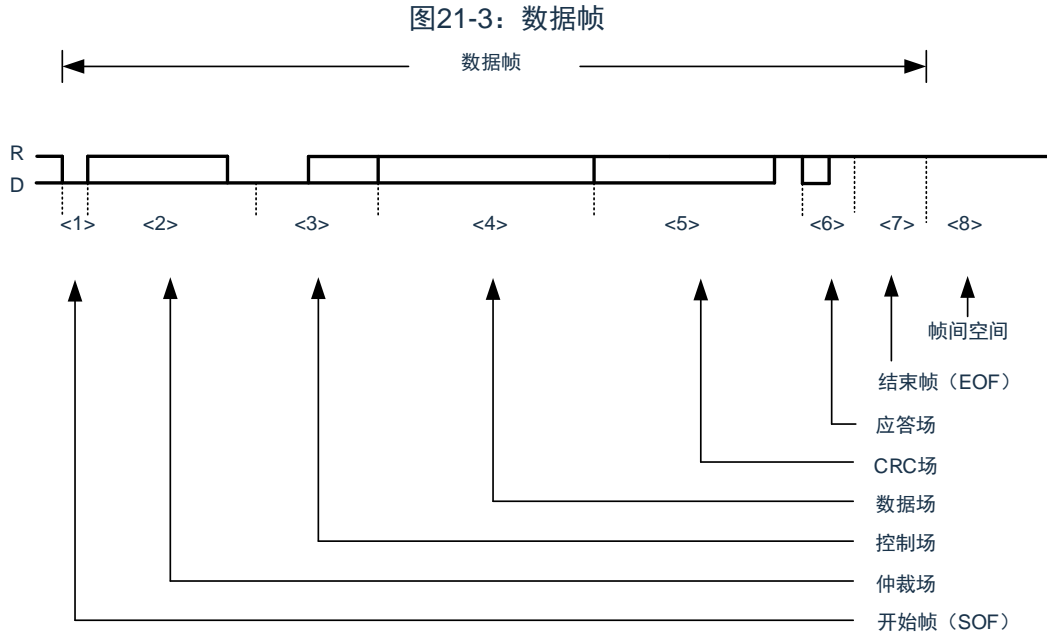
总线值分为显性和隐性两种

- 显性值是逻辑0
- 隐性值是逻辑1
- 当总线同时出现显性和隐性时，总线的状态显示为显性

21.2.3 数据帧和远程帧

(4) 数据帧

数据帧由7个不同的位场组成

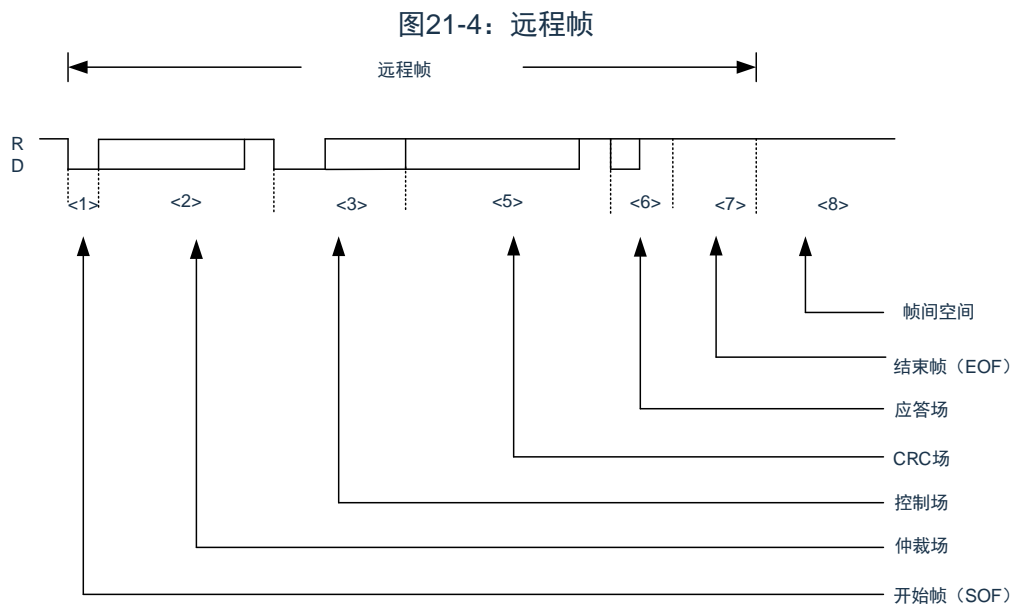


备注: D: 显性=0

R: 隐性=1

(5) 远程帧

远程帧由6个位场主成.



备注:

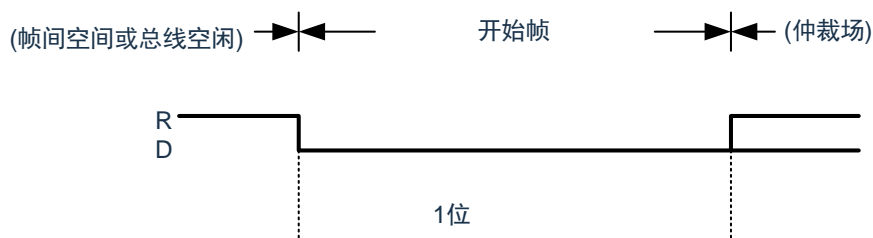
1. 即使控制位场的数据长度不是“0000B”，数据位场也不会发送.
2. D: 显性=0
R: 隐性=1

(6) 位场描述

<1> 开始帧(SOF)

开始帧位场位于数据帧或远程帧的开始.

图21-5: 开始帧(SOF)



备注: D: 显性=0

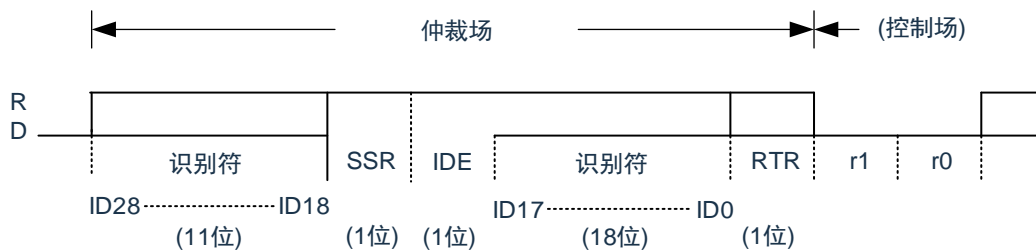
R: 隐性=1

- 如果在总线空闲状态中检测到显性位, 则执行硬同步(当前 TQ 被指定为 SYNC 段)
- 如果在此类硬同步后在采样点采样显性位, 则该位将分配为 SOF。如果检测到隐性位, 协议层将返回到总线空闲状态, 并将前面的显性脉冲视为仅干扰。在这种情况下, 不会生成错误帧。

<2> 仲裁场

仲裁场用来设置优先级，数据帧/远程帧和帧格式

图21-6: 仲裁场(在标准格式)



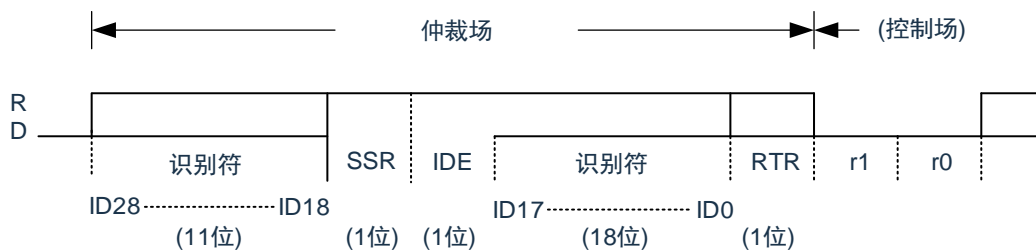
注意:

1. ID18到ID28是识别符。
2. 识别符是MSB先传输。

备注: D: 显性=0

R: 隐性=1

图21-7: 仲裁场 (在扩展帧模式)



注意:

1. ID18 到ID28是识别符。
2. 识别符是MSB优先发送。

备注: D: 显性=0

R: 隐性=1

表21-3: RTR 帧设置

帧类型	RTR位
数据帧	0(D)
远程帧	1(R)

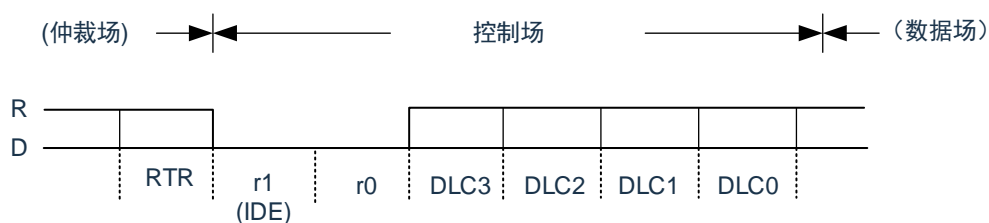
表21-4: 帧格式设置(IDE 位) 和识别符(ID)位数设置

帧格式	SRR位	IDE位	位数
标准帧格式	None	0(D)	11位
扩展帧格式	1(R)	1(R)	29位

<3> 控制场

控制场设置“DLC”作为数据场中的数据字节数(DLC=0到8).

图21-8: 控制场



备注: D: 显性=0

R: 隐性=1

在标准格式帧, 控制场的IDE位和r1位相同

表21-5: 数据长度设置

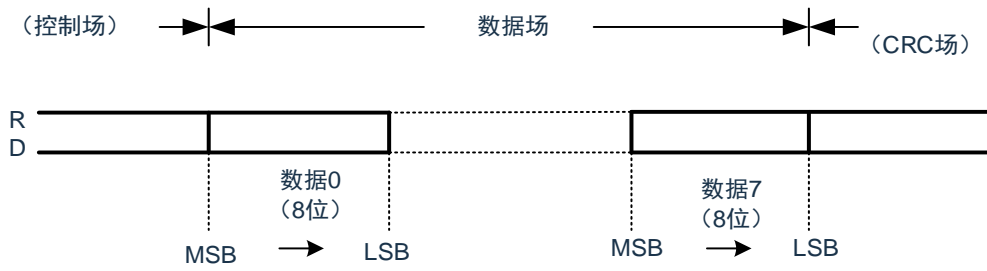
数据长度代码				数据字节计算
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0字节
0	0	0	1	1字节
0	0	1	0	2字节
0	0	1	1	3字节
0	1	0	0	4字节
0	1	0	1	5字节
0	1	1	0	6字节
0	1	1	1	7字节
1	0	0	0	8字节
其他值				8 字节(无论DLC0到DLC3是什么值)

注意: 在远程帧中, 即使数据长度代码不是(0000B), 也没有数据场

<4> 数据场

数据场包含控制字段设置的数据量(字节单位)。最多可设置8个数据单元。

图21-9: 数据场



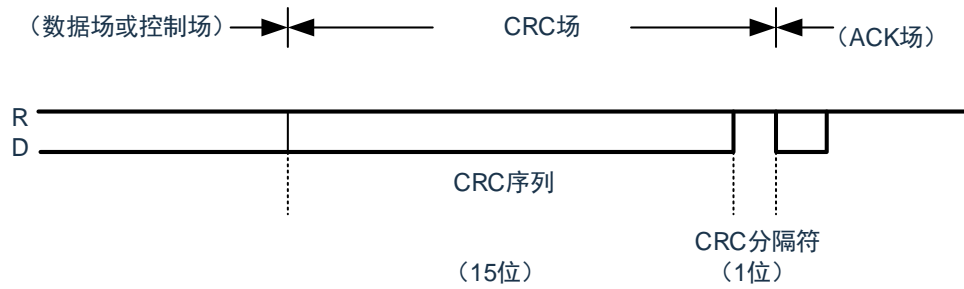
备注: D: 显性=0

R: 隐性=1

<5> CRC场

CRC场是一个16位的场, 用来检测数据传输中的错误。

图21-10: CRC场



备注: D: 显性=0

R: 隐性=1

- 用于生成 15 位 CRC 序列的多项式 $P(X)$ 表示如下:
- $P(X)=X^{15}+X^{14}+X^{10}+X^8+X^7+X^4+X^3+1$
- 发送节点: 发送开始帧、仲裁场、控制场和数据场的数据(位填充之前)计算的 CRC 序列
- 接收节点:使用排除接收数据中的填充位的数据位与 CRC 字段中的 CRC 序列, 将计算的 CRC 序列进行比较。如果两个 CRC 序列不匹配, 节点将发出错误帧。

<8> 帧间空间

帧间空间用于插入于数据帧，远程帧，错误帧，或过载帧之间，来区分两帧。

- 总线状态的不同基于错误的状态的不同

(a) 错误活动节点

帧间空间包含3位间隙场和一个总线空闲场。

图21-13: 帧间空间(错误活动节点)



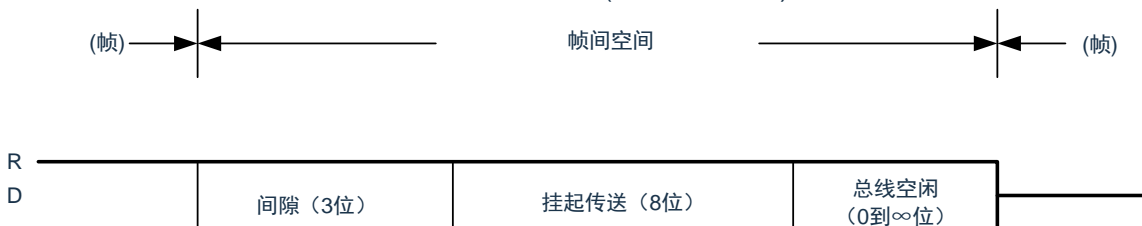
备注:

1. Bus 空闲: 总线没有被使用的节点状态
2. D: 显性=0
R: 隐性=1

(b) 错误主动节点

帧间空间包含一个间隙场，一个挂起场和一个总线空闲场。

图21-14: 帧间空间(错误主动节点)



备注:

1. 总线空闲: 总线没有被任何节点使用。
挂起传送: 在错误被动情况,8个隐形的位会在节点被发送.
2. D: 显性=0
R: 隐形=1

一般情况下，间隙场是3个位,如果发送节点在间隙场的第三个位侦测到了一个显性位,此时无论如何都会执行发送操作.

- 错误状态操作

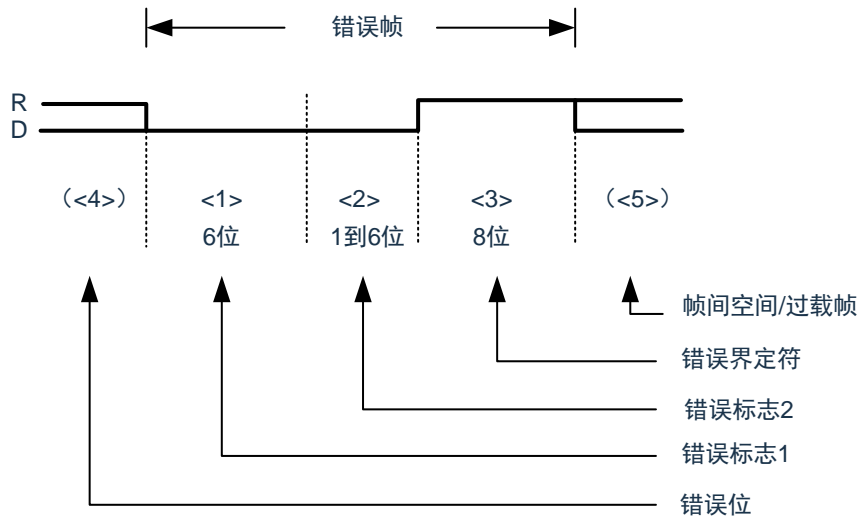
表21-6: 错误状态下的操作

错误状态	操作
错误主动	3个位间隙后可以立即被传输.
错误被动	在间隙后可以发送8个位.

21.2.4 错误帧

如果侦测到一个错误,会在接下来的节点发送错误帧.

图21-15: 错误帧



备注: D: 显性=0
R: 隐性=1

表21-7: 错误帧的定义

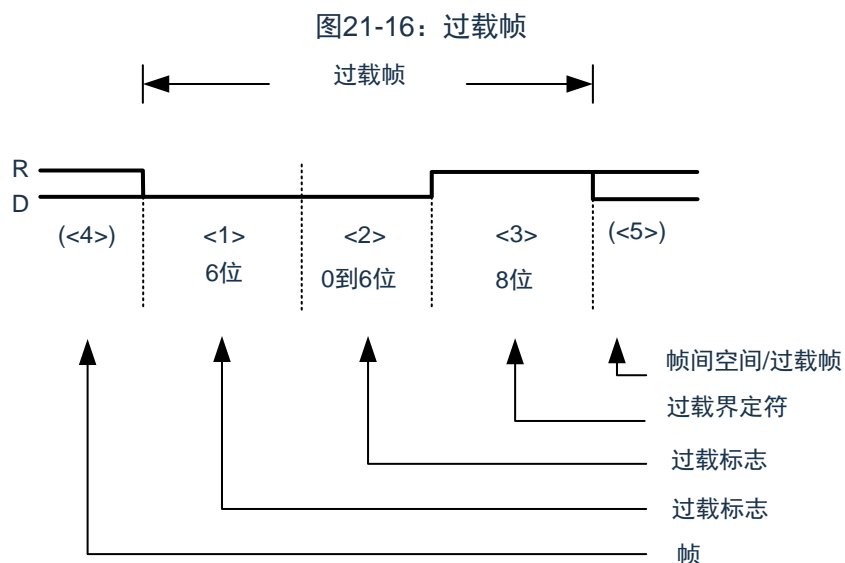
No.	名字	位数	定义
<1>	错误标志1	6	错误主动节点: 连续的输出6个显性位. 错误被动节点: 连续的输出6个隐性位. 当一个节点正在输出被动错误标志时,另一节点输出一个显性位,此时错误被动标志不会被清除,直到连续的6个相同极性的位被侦测.
<2>	错误标志2	0~6	节点接受错误标志1侦测位填充错误,然后发出这个错误标志.
<3>	错误界定符	8	连续的输出8个隐性位. 如果在第8个位侦测到一个显性位,下一个位会发出一个过载帧.
<4>	错误位	-	侦测到错误. 紧接着错误位后发出错误标志. 如果是CRC错误,错误位在ACK界定符后接着被发出.
<5>	帧间空间/过载帧	-	一个帧间空间或则过载帧此时被发出

21.2.5 过载帧

过载帧在下面几种情况发出。

- 当接收节点没有完成接收操作^{注1}
- 在间隙期间，如果前两个位被侦测显性位
- 如果显性位在结束帧的最后一位(第7位)或在错误界定符/过载界定符的最后一位(第8位)

注1：CAN内部速度足够快来处理所有接受到的帧，从而不会产生过载帧。



备注：D：显性=0

R：隐性=1

表21-8：过载帧的场定义

No	名字	位数	定义
<1>	过载标志	6	连续的输出6个显性位
<2>	其他节点的过载标志	0 到6	在帧间空间接收到过载标志，发出过载标志
<3>	过载界定符	8	连续的输出8个隐形位 如果在第8个位侦测到显性位，接着下一个位发出过载帧
<4>	帧	-	在结束帧后输出，错误界定符，过载界定符.
<5>	帧间空间/过载帧	-	此时发出帧间空间或者过载帧

21.3 功能

21.3.1 总线优先级设定

- (1) 当节点开始传输:
 - 当总线空闲, 节点输出数据开始
- (2) 当大于一个节点开始传输:
 - 节点从仲裁场的第一个位持续输出最长的显性位来捕获总线的优先级(如果显性位和隐性位同时传输, 显性位作为总线值).
 - 传输节点比较其输出仲裁字段和总线上的数据电平.

表21-9: 总线优先级设定

电平匹配	持续传输
电平不匹配	当不匹配被侦测就停止传输, 然后在下一个位开始接收

- (3) 数据帧和远程帧的优先级
 - 当数据帧和远程帧同时在总线上, 数据帧更有优先级, 因为数据帧的RTR位, 仲裁帧的最后一个位是显性位

注意: 如果扩展格式数据帧和标准格式远程帧在总线冲突了(如果ID18到ID28 相同), 标准格式远程帧占有优先级。

21.3.2 位填充

位填充是当有5个连续的相同位出现时, 通过插入一位反转数补充位来建立同步, 从而阻止爆发性的错误。

表21-10: 位填充

发送	当数据帧或远程帧发送的时候, 如果在帧开始和ACK场, 连续出现5个相同的位, 一个与数据位相反的数据插入在接下来的位。
接收	当接收数据帧或远程帧, 当开始帧和ACK场之间出现连续相同的5个位, 删除这个位后继续接收数据

21.3.3 多主设备

标识符决定总线的优先级(节点捕获到传输功能)任一节点可以作为总线的主设备

21.3.4 多点广播

即使只有一个传输节点, 两个或更多的节点可以同时接收相同的数据, 因为相同的标识符可以在两个或更多的节点被设置

21.3.5 CAN 睡眠模式/CAN停止模式功能

CAN睡眠/CAN停止模式让CAN控制器进入等待模式, 从而获得更低的功耗。

CAN 睡眠模式能从总线操作中唤醒, 但是CAN停止模式不能被总线操作唤醒(CAN停止模式被CPU控制)。

21.3.6 错误控制功能

(4) 错误类型

表21-11：错误类型

类型	错误描述		侦测状态	
	检测方法	检测条件	发送/接收	场/帧
位错误	比较输出电平和总线电平	不匹配的电平	发送/接收节点	总线上在开始帧和结束帧之间的数据的位, 错误帧和过载帧
填充错误	在填充位检测接收数据	6个输出电平连续相同的位	接收节点	开始帧到CRC时序
CRC错误	从接收的数据和接收的CRC时序中比较CRC	CRC不匹配	接收节点	CRC场
格式错误	在混合的格式中检测场/帧	侦测混合的格式错误	接收节点	CRC界定符ACK场 结束帧 错误帧 过载帧
ACK错误	在发送节点中检测ACK间隙	在ACK间隙中检测隐形位	发送节点	ACK间隙

(5) 错误帧输出时序

表21-12：错误帧输出时序

类型	输出时序
位错误, 填充错误, 格式错误, ACK错误	在错误侦测后的输出时序开始输出错误帧
CRC错误	在ACK界定符后的输出时序开始输出错误帧

(6) 错误的处理

发送节点在错误帧后重新发送数据帧或远程帧 (在单次模式不会重新发送帧)。

(7) 错误状态

(a) 错误状态的类型

接下来的三个错误状态由CAN规范定义

- 错误主动
- 错误被动
- 总线关闭

错误类型由CAN错误计数寄存器(C0ERC)的TEC0到TEC7(发送错误计数位)和REC0到REC6(接收错误计数位)的值决定,如表 21-13所示。

当前的错误由CAN模块的信息寄存器(C0INFO)决定。

当每一个计数器变得相等或大于错误警告水平(96)时, C0INFO 寄存器的TECS0或RECS0被设置为1。在这种情况下,总线的状态必须被测试,因为此时可能存在严重的故障。错误计数器的值为128或更大时,表明处于错误被动状态,寄存器C0INFO的TECS1或RECS1被置位1。

- 如果传输错误计数器的值大于等于256(实际上,传输错误计数器并不会表明一个值大于或等于256),总线关闭状态到达,C0INFO寄存器的BOFF位被置为1。
- 在启动时,如果总线上只有一个节点被激活(例如:一个特殊的情况,比如当总线只连接到本地站点),ACK不会被返回,即使数据被传输。因此,重复的传输错误帧和数据。在错误被动状态,传输错误计数器不会增加,总线关闭状态不会出现。

表 21-13: 错误状态的类型

类型	操作	错误计数器的值	C0 INFO寄存器	获得错误状态的操作细节
错误主动	发送	0-95	TECS1, TECS0 =00	- 输出一个主动错误标志 (6个持续的显性位)
	接收	0-95	RECS1, RECS0 =00	
	发送	96-127	TECS1, TECS0 =01	
	接收	96-127	RECS1, RECS0 =01	
错误被动	发送	128-255	TECS1, TECS0 =11	- 输出一个被动错误标志(6个持续的隐形位) - 在传输中接着间隙(挂起传输)发送 8 个隐形位。
	接收	128或更多	RECS1, RECS0 =11	
总线关闭	发送	256 或更多 (没有指定) ^{注1}	BOFF =1, TECS1, TECS0 =11	- 通讯已经不可能. 接收帧时报文不会被储存, 接下来的<1>, <2>, 和<3>被完成. <1> TSOUT 绑定. <2> REC 增加/减少. <3> VALID 位被设置. - 如果 CAN 模块进入初始化模式, 到操作模式的转换申请被制定, 当 11 个持续的隐形位被检测到 128 次, 错误计数器被复位到 0, 错误主动状态会被记录

注1: 当BOFF位的值是1, 传输错误计数器的值(TEC)无效, 如果一个错误让错误传输计数器的值增加了 8, 当计数器的值在248到255的范围内, 计数器不会再增加, 总线关闭状态不会被认定。

(b) 错误计数器

当错误发生，错误计数器往上计数；当成功的接受和发送，错误计数器往下计数。当错误被侦测时，错误计数器的值立即更新。

表21-14：错误计数器

状态	传输错误计数器 (TEC0 ~TEC7)	接收错误计数器 (REC0 ~REC6)
接收节点侦测到一个错误 (除了主动错误标志或过载标志).	没有改变	+1 (当REPS 位=0)
错误帧的错误标志发出后接收节点侦测到显性位	没有改变	+8 (当REPS 位=0)
发送节点发出错误标志 [如期望的, 在下面几种情况下错误计数器不会改变] <1> 在被动状态ACK错误被侦测而且当被动错误标志发出时显性位没有被侦测 <2>在仲裁场填充错误被侦测, 发送一个隐形位作为填充, 但是显性位被侦测	+8	没有改变
在主动错误标志或过载标志输出时检测到位错误(错误-主动发送节点)	+8	没有改变
在主动错误标志或过载标志输出时检测到位错误(错误-主动接收节点)	没有改变	+8 (当REPS 位 =0)
当节点从主动误差标志或重载标志开始检测 14 个连续的显性位, 然后检测 8 个连续的显性位. 当节点在被动错误标记后检测到 8 连续显性位时.	+8 (传输时)	+8 (接收时,当REPS 位 =0)
传输节点完成传输时无错误 (± 0 如果错误计数器 =0)	-1	没有改变
接收节点完成接收时无错误	没有改变	- -1 ($1 \leq \text{REC6}$ 到 $\text{REC0} \leq 127$, 当REPS位=0) - ± 0 (REC6 到 $\text{REC0} = 0$, 当REPS位=0) 值119到127被设置 (当REPS 位=1)

(c) 位错误在间歇中发生

过载帧产生

警告： 如果发生错误，则根据发生错误之前传输错误计数器和接收错误对的内容控制错误标志输出(主动或被动)。输出错误标志后，错误计数器的值将递增。

(8) 总线关闭恢复

当CAN模块处于总线关闭状态,CAN模块永久设置它的输出信号(CTxD)为隐性位.

CAN 模块在以下总线关闭恢复序列中从总线关闭状态中恢复.

<1> 申请进入CAN初始化模式

<2> 申请进入CAN操作模式

(a) 从正常的恢复序列中恢复

(b) 跳过恢复序列强制恢复操作

(a) 按照正常的恢复序列从总线关闭状态中恢复

CAN模块第一次发布进入初始化申请(参考时序 <1> 在图21-17).这个申请会立即得到应答,并且C0CTRL的OPMODE位会被清0.分析这个故障的处理会导致总线关闭状态,使用软件去重新定义CAN模块和报文缓存,或则通过清除GOM位来停止CAN操作.

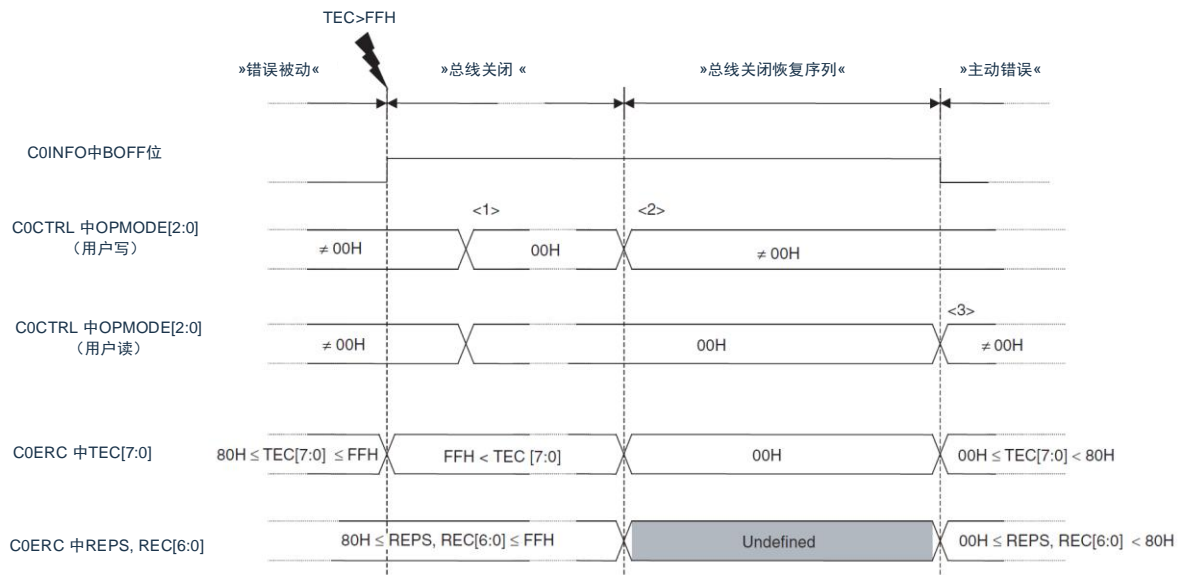
接下来,用户申请从初始状态到操作模式(参考时序<2>在图21-17).这启动一个操作让CAN模块从总线关闭状态中恢复. CAN 协议 ISO 11898 定义了模块从总线关闭状态中恢复的条件,需要检测11个连续的隐性位128次.此时,将模式更改为操作模式的请求保持挂起状态,直到恢复条件得到满足.满足恢复条件时.(参考时序 <3>在图21-17),CAN 模块可以进入它请求的操作模式.在CAN模块进入此操作模式之前,它将保持初始化模式.通过读取C0CTRL寄存器的OPMODE位,可以确认要请求的操作模式的完成..

在总线关闭期间和总线关闭恢复序列中,C0INFO寄存器的BOFF位保持设置(设置为1).在总线关闭恢复序列中,接收错误计数器(REC[6:0])计算总线上检测到11连续隐性位的次数.因此,可以通过读取REC_6:0来检查恢复状态.

注意:

1. 如果通过进入初始化模式并重新进入任何操作模式而中断总线关闭恢复序列,则总线关闭恢复序列将从这一点从头开始重新启动,等待阶段将再次为128次11个隐性位.
 - a) 在总线关闭恢复序列中,每次检测到11个连续的隐性位时,REC [6:0]都会计数(+1).即使在总线关闭期间,CAN 模块也可以进入CAN 睡眠模式或CAN 停止模式.要启动总线关闭恢复序列,必须转移到初始化模式一次.
2. 但是,当CAN模块处于CAN睡眠模式或CAN停止模式时,不接受向初始化模式过渡的请求,因此必须首先释放CAN睡眠模式.在这种情况下,一旦CAN睡眠模式释放,总线关闭恢复序列就会启动,并且无需过渡到初始化模式.如果CAN模块在总线关闭期间在睡眠模式下检测到CAN总线上的显性边缘,则睡眠模式将离开,总线关闭恢复序列将启动(在提供CAN时钟的状态下,必须在显性边沿之后通过软件清除PSMODE)。

图21-17: 通过正常恢复序列从总线关闭状态恢复操作



(b) 跳过总线关闭恢复序列的强制恢复操作

无论总线状态如何，CAN 模块都可以通过跳过总线关闭恢复序列强制从总线关闭状态释放。这是过程

首先，CAN 模块请求进入初始化模式。有关此时要注意的操作和点，请参阅 (a) 通过正常恢复序列从总线关闭状态恢复操作。

接下来，模块请求进入操作模式。同时，COCTRL 寄存器的 CCERC 位必须设置为 1。

因此，跳过了 CAN 协议 ISO 11898 定义的总线关闭恢复序列，模块立即进入操作模式。在这种情况下，模块在监视了 11 个连续的隐性位后连接到 CAN 总线。有关详细信息，请参阅图 21-82。

注意：此功能不是由 CAN 协议 ISO 11898 定义的。使用此功能时，要彻底评估其对网络系统的影响。

(9) 初始化模式下初始化 CAN 模块错误计数器寄存器 (COERC)

如果需要初始化 CAN 模块错误计数器寄存器 (COERC) 和 CAN 模块信息寄存器 (COINFO) 以调试或评估程序，则可以通过在初始化模式。初始化完成后，CCERC 位将自动清除为 0。

注意：

1. 此功能仅在初始化模式下启用。即使 CCERC 位在 CAN 操作模式下设置为 1，也不会初始化寄存器 COERC 和 COINFO。
2. CCERC 位可以设置为与请求进入 CAN 操作模式同时设置。

21.3.7 波特率控制功能

(10) 预缩放

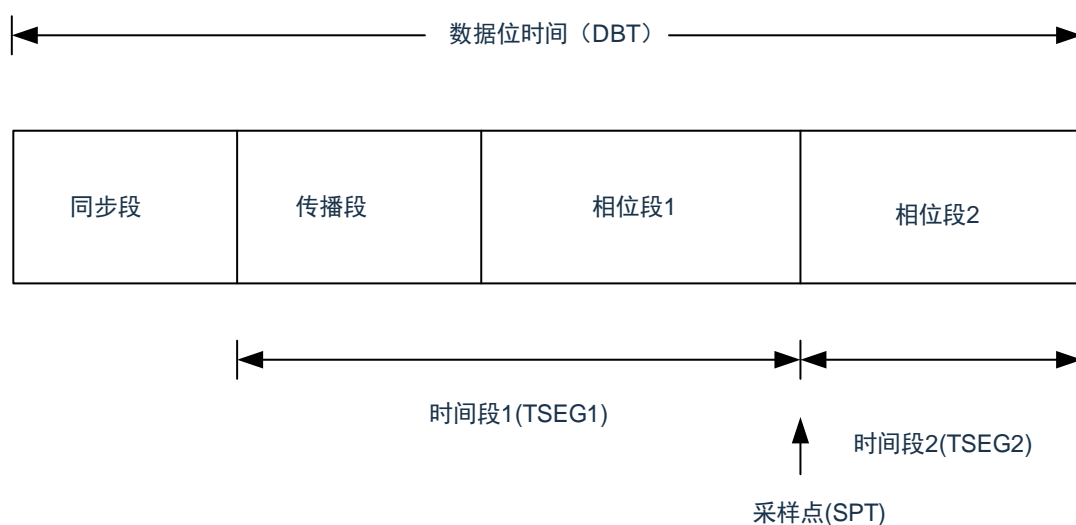
CAN控制器具有一个预缩放，用于划分提供给CAN的时钟(F_{CAN})。此预缩放生成从CAN模块系统时钟(F_{CANMOD})派生的CAN协议层基本时钟(F_{TQ})，分频1到256(请参阅21.6(12)CAN比特率预缩放寄存器(COBRP))

(11) 数据位时间(8-25 时间量化)

一个数据位时间定义如下图21-18.

CAN 控制器将时间段 1、时间段 2 和重新同步跳转宽度 (SJW) 作为数据位时间的参数，如图21-18. 时间段1等效于由 CAN 协议规范定义的传播(道具)段和相位段1的总和。时间段2等效于相位段2..

图21-18: 时段设置



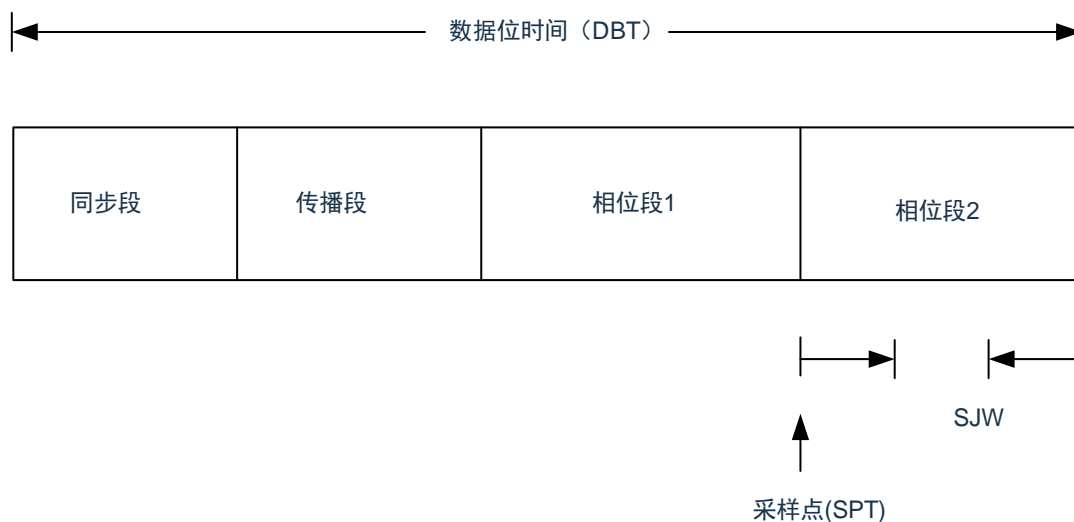
段名称	可设置的范围	关于设置为 CAN 规范的说明
时间段 1(TSEG1)	2TQ-16TQ	—
时间段2(TSEG2)	1TQ-8TQ	CAN 控制器的IPT是0TQ。因此，为了符合CAN协议规范，必须在此处设置长度小于或等于相段1。这意味着时段1减去1TQ的长度是时段的可设置上限2.
重新同步跳转宽度 (SJW)	1TQ-4TQ	时间段1长度减去 1TQ 或 4TQ，以较小者为准

备注：IPT：信息处理时间

TQ：时间份额

参考：CAN 标准 ISO 11898 规范定义了构成数据位时间的段，如图21-19.

图21-19：参考：由 CAN 规范定义的数据位时间配置



段名称	段长度	描述
同步段	1	此段从硬同步建立时电平从隐性变为显性的边缘开始.
传播段	可编程从 1到8或则更多	此段吸收输出缓冲区、CAN 总线和输入缓冲区的延迟。设置此段的长度，以便在相位 1 开始之前返回 ACK。 传播段时间 \geq (输出缓冲延时) + 2x(CAN总线延时) + (输入缓冲延时)
相位段1	可编程1到8	此段补偿数据比特时间的错误。此段越长，允许范围越宽，但通信速度越慢.
相位段2	相位段1或则IPT, 取最大	
SJW	可编程从1TQ到段1的长度或则4TQ,取最小	此宽度设置在重新同步期间扩展或收缩相段的上限

备注：IPT：信息处理时间

TQ：时间份额

(12) 同步数据位

- 接收节点通过总线上的电平更改建立同步，因为它没有异步信号。
- 传输节点在传输节点的位时间同步传输数据

(a) 硬件同步

当接收节点检测到帧间空间中的帧开始时，将建立此同步。

- 在总线上检测到下降沿时，TQ表示同步段，下一段是传播段。在这种情况下，无论 SJW 如何，都会建立同步。

图21-20：总线空闲期间在识别显性位时的硬同步



(b) 重同步

如果在接收期间总线上检测到电平变化，则再次建立同步(仅当之前对隐性电平进行了采样时)。

- 边缘的相位误差由检测到的边缘和同步段的相对位置给出。

<相位误差信号>

0: 如果边沿在同步段内

正: 如果边缘位于采样点之前(相位误差)

负: 如果边缘位于采样点之后(相位误差)

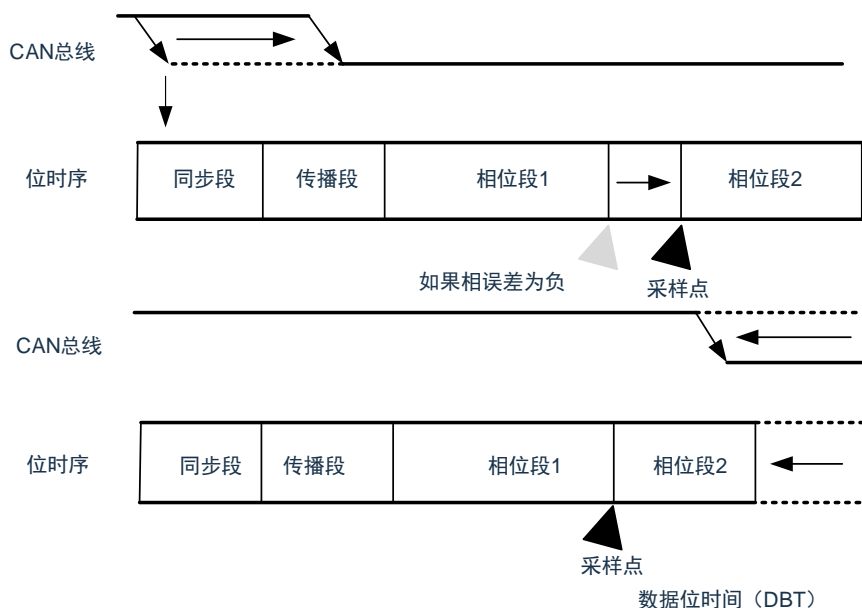
如果相位误差为正: 通过指定的 SJW, 相位段 1 较长。

如果相位误差为负: 相位段 2 由指定的 SJW 缩短。

- 由于发送节点与接收节点之间的波特速率"差异", 接收节点数据的采样点相对移动。

图21-21：重同步

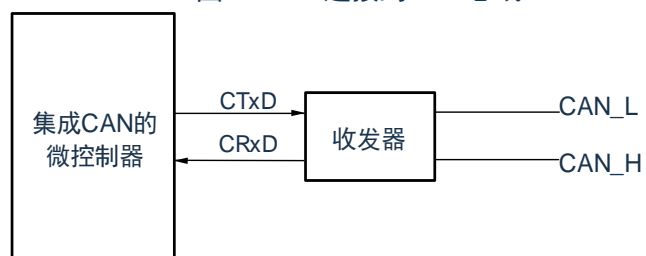
如果相误差为正



21.4 与目标系统的连接

集成 CAN 的微控制器必须使用外部收发器连接到 CAN 总线。

图21-22：连接到CAN总线



21.5 CAN控制器的内部寄存器

21.5.1 CAN控制器配置

表21-15: CAN控制器寄存器列表 (1/2)

项	寄存器名称
控制寄存器	外围使能寄存器 0(PER0)
	串行通信引脚选择寄存器 (PIOR3)
	端口寄存器 0, 5 (P0, P5)
	端口模式寄存器 0, 5 (PM0, PM5)
CAN全局寄存器	CAN全局模块控制寄存器 (COGMCTRL)
	CAN全局模块时钟选择寄存器(COGMCS)
	CAN全局自动块传输控制寄存器 (COGMABT)
	CAN全局自动块传输延时设置寄存器 (COGMABTD)
CAN模块寄存器	CAN模块屏蔽1寄存器L (COMASK1L)
	CAN模块屏蔽1寄存器 H (COMASK1H)
	CAN 模块屏蔽2寄存器L (COMASK2L)
	CAN 模块屏蔽2寄存器H (COMASK2H)
	CAN 模块屏蔽3寄存器 L (COMASK3L)
	CAN 模块屏蔽3寄存器 H (COMASK3H)
	CAN 模块屏蔽4寄存器 L (COMASK4L)
	CAN模块屏蔽4寄存器 H (COMASK4H)
	CAN 模块控制寄存器(COCTRL)
	CAN模块上次错误代码寄存器(COLEC)
	CAN 模块信息寄存器 (COINFO)
	CAN模块错误计数寄存器(COERC)
	CAN模块中断使能寄存器(COIE)
	CAN 模块中断状态寄存器 (COINTS)
	CAN模块位率缩放寄存器 (COBRP)
	CAN模块位率寄存器(COBTR)
	CAN模块最后一个输入指针寄存器(COLIPT)
	CAN模块接收历史记录列表寄存器 (CORGPT)
	CAN 模块最后一个输出指针寄存器(COLOPT)
	CAN 模块发送历史记录列表寄存器(COTGPT)
CAN模块时间戳寄存器(COTS)	

备注: CAN全局寄存器通过CGM<寄存器功能>识别。

CAN模块寄存器通过C<寄存器功能>识别。

报文缓冲寄存器由CM<寄存器功能>识别。

表21-15: CAN控制器寄存器列表 (2/2)

项	寄存器名称
报文缓存寄存器	CAN报文数据字节01寄存器m (C0MDB01m)
	CAN 报文数据字节0寄存器m (C0MDB0m)
	CAN报文数据字节1寄存器m (C0MDB1m)
	CAN报文数据字节23寄存器m (C0MDB23m)
	CAN 报文数据字节2寄存器m (C0MDB2m)
	CAN报文数据字节3寄存器 m (C0MDB3m)
	CAN 报文数据字节45寄存器m (C0MDB45m)
	CAN 报文数据字节4寄存器 m (C0MDB4m)
	CAN报文数据字节5寄存器 m (C0MDB5m)
	CAN报文数据字节67寄存器m (C0MDB67m)
	CAN 报文数据字节6寄存器m (C0MDB6m)
	CAN 报文数据字节7寄存器 m (C0MDB7m)
	CAN报文数据长度寄存器m (C0MDLm)
	CAN报文配置寄存器m (C0MCONFm)
	CAN报文ID寄存器L m (C0MIDLm)
	CAN报文ID寄存器H m (C0MIDHm)
	CAN报文控制寄存器m (C0MCTRLm)

备注:

1. CAN全局寄存器通过CGM<寄存器功能>识别。
模块寄存器通过 C<寄存器功能>识别。
报文缓冲寄存器由CM<寄存器功能>识别。
2. m = 0 到15

21.5.2 寄存器访问类型

CAN控制器的外围I/O寄存器被定位在0x40045400到 0x400455FF.

表21-16: 寄存器访问类型 (1/9)

地址	寄存器名称	Symbol	读/写	位操作单元		默认值
				8	16	
0x40045400H	CAN0 全局模块控制寄存器	C0GMCTRL	读/写	-	√	0000H
0x40045406H	CAN0全局自动块传输控制寄存器	C0GMABT		-	√	0000H
0x40045408H	CAN0全局自动块传输延时设置	C0GMABTD		√	-	00H
0x40045402H	CAN0全局模块时钟选择寄存器	C0GMCS		√	-	0FH
0x40045440H	CAN0模块屏蔽1寄存器	C0MASK1L		-	√	没有定义
0x40045442H		C0MASK1H				没有定义
0x40045444H	CAN0模块屏蔽2寄存器	C0MASK2L		-	√	没有定义
0x40045446H		C0MASK2H				没有定义
0x40045448H	CAN0模块屏蔽3寄存器	C0MASK3L		-	√	没有定义
0x4004544AH		C0MASK3H				没有定义
0x4004544CH	CAN0模块屏蔽4寄存器	C0MASK4L		-	√	没有定义
0x4004544EH		C0MASK4H				没有定义
0x40045450H	CAN0模块控制寄存器	C0CTRL		-	√	0000H
0x40045452H	CAN0模块最后错误代码寄存器	C0LEC		√	-	00H
0x40045453H	CAN0 模块信息寄存器	C0INFO	读	√	-	00H
0x40045454H	CAN0模块错误计数寄存器	C0ERC		-	√	0000H
0x40045456H	CAN0模块中断使能寄存器	C0IE	读/写	-	√	0000H
0x40045458H	CAN0模块中断状态寄存器	C0INTS		-	√	0000H
0x4004545AH	CAN0模块位率缩放寄存器	C0BRP		√	-	FFH
0x4004545CH	CAN0模块位率寄存器	C0BTR		-	√	370FH
0x4004545EH	CAN0模块最后输入指针寄存器	C0LIPT	读	√	-	没有定义
0x40045460H	CAN0模块接收历史列表寄存器	C0RGPT	读/写	-	√	xx02H
0x40045462H	CAN0模块最后输出指针寄存器	C0LOPT	读	√	-	没有定义
0x40045464H	CAN0模块发送历史列表寄存器	C0TGPT	读/写	-	√	xx02H
0x40045466H	CAN0模块时间戳寄存器	C0TS		-	√	0000H

表21-16: 寄存器访问类型(2/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值
				8	16	
0x40045500H	CAN0报文数据字节01寄存器00	C0MDB0100	读/写	-	√	没有定义
0x40045500H	CAN0报文数据字节0寄存器00	C0MDB000		√	-	没有定义
0x40045501H	CAN0报文数据字节1寄存器00	C0MDB100		√	-	没有定义
0x40045502H	CAN0报文数据字节23寄存器00	C0MDB2300		-	√	没有定义
0x40045502H	CAN0报文数据字节2寄存器00	C0MDB200		√	-	没有定义
0x40045503H	CAN0报文数据字节3寄存器00	C0MDB300		√	-	没有定义
0x40045504H	CAN0报文数据字节45寄存器00	C0MDB4500		-	√	没有定义
0x40045504H	CAN0报文数据字节4寄存器00	C0MDB400		√	-	没有定义
0x40045505H	CAN0报文数据字节5寄存器00	C0MDB500		√	-	没有定义
0x40045506H	CAN0报文数据字节67寄存器00	C0MDB6700		-	√	没有定义
0x40045506H	CAN0报文数据字节6寄存器00	C0MDB600		√	-	没有定义
0x40045507H	CAN0报文数据字节7寄存器00	C0MDB700		√	-	没有定义
0x40045508H	CAN0 报文数据长度寄存器00	C0MDLC00		√	-	0000xxxxB
0x40045509H	CAN0 报文配置寄存器00	C0MCONF00		√	-	没有定义
0x4004550AH	CAN0报文ID寄存器00	C0MIDL00		-	√	没有定义
0x4004550CH		C0MIDH00		-	√	没有定义
0x4004550EH	CAN0报文控制寄存器00	C0MCTRL00		-	√	00x00000 000xx000B
0x40045510H	CAN0报文数据字节01寄存器01	C0MDB0101		-	√	没有定义
0x40045510H	CAN0报文数据字节0寄存器01	C0MDB001		√	-	没有定义
0x40045511H	CAN0报文数据字节1寄存器01	C0MDB101		√	-	没有定义
0x40045512H	CAN0报文数据字节23寄存器01	C0MDB2301		-	√	没有定义
0x40045512H	CAN0报文数据字节2寄存器01	C0MDB201		√	-	没有定义
0x40045513H	CAN0报文数据字节3寄存器01	C0MDB301		√	-	没有定义
0x40045514H	CAN0报文数据字节45寄存器01	C0MDB4501		-	√	没有定义
0x40045514H	CAN0报文数据字节4寄存器01	C0MDB401		√	-	没有定义
0x40045515H	CAN0报文数据字节5寄存器01	C0MDB501		√	-	没有定义
0x40045516H	CAN0报文数据字节67寄存器01	C0MDB6701		-	√	没有定义
0x40045516H	CAN0报文数据字节6寄存器01	C0MDB601		√	-	没有定义
0x40045517H	CAN0报文数据字节7寄存器01	C0MDB701		√	-	没有定义
0x40045518H	CAN0 报文数据长度寄存器01	C0MDLC01		√	-	0000xxxxB
0x40045519H	CAN0 报文配置寄存器01	C0MCONF01		√	-	没有定义
0x4004551AH	CAN0报文ID寄存器01	C0MIDL01		-	√	没有定义
0x4004551CH		C0MIDH01	-	√	没有定义	
0x4004551EH	CAN0报文控制寄存器01	C0MCTRL01	-	√	00x00000 000xx000B	

表21-16: 寄存器访问类型(3/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值
				8	16	
0x40045520H	CAN0报文数据字节01寄存器02	C0MDB0102	读/写	-	√	没有定义
0x40045520H	CAN0报文数据字节0寄存器02	C0MDB002		√	-	没有定义
0x40045521H	CAN0报文数据字节1寄存器02	C0MDB102		√	-	没有定义
0x40045522H	CAN0报文数据字节23寄存器02	C0MDB2302		-	√	没有定义
0x40045522H	CAN0报文数据字节2寄存器02	C0MDB202		√	-	没有定义
0x40045523H	CAN0报文数据字节3寄存器02	C0MDB302		√	-	没有定义
0x40045524H	CAN0报文数据字节45寄存器02	C0MDB4502		-	√	没有定义
0x40045524H	CAN0报文数据字节4寄存器02	C0MDB402		√	-	没有定义
000F625H	CAN0报文数据字节5寄存器02	C0MDB502		√	-	没有定义
0x40045526H	CAN0报文数据字节67寄存器02	C0MDB6702		-	√	没有定义
0x40045526H	CAN0报文数据字节6寄存器02	C0MDB602		√	-	没有定义
0x40045527H	CAN0报文数据字节7寄存器02	C0MDB702		√	-	没有定义
0x40045528H	CAN0报文数据长度寄存器02	C0MDLC02		√	-	0000xxxxB
0x40045529H	CAN0 报文配置寄存器02	C0MCONF02		√	-	没有定义
0x4004552AH	CAN0报文ID寄存器02	C0MIDL02		-	√	没有定义
0x4004552CH		C0MIDH02		-	√	没有定义
0x4004552EH	CAN0报文控制寄存器02	C0MCTRL02		-	√	00x00000 000xx000B
0x40045530H	CAN0报文数据字节01寄存器03	C0MDB0103		-	√	没有定义
0x40045530H	CAN0报文数据字节0寄存器03	C0MDB003		√	-	没有定义
0x40045531H	CAN0报文数据字节1寄存器03	C0MDB103		√	-	没有定义
0x40045532H	CAN0报文数据字节23寄存器03	C0MDB2303		-	√	没有定义
0x40045532H	CAN0报文数据字节2寄存器03	C0MDB203		√	-	没有定义
0x40045533H	CAN0报文数据字节3寄存器03	C0MDB303		√	-	没有定义
0x40045534H	CAN0报文数据字节45寄存器03	C0MDB4503		-	√	没有定义
0x40045534H	CAN0报文数据字节4寄存器03	C0MDB403		√	-	没有定义
0x40045535H	CAN0报文数据字节5寄存器03	C0MDB503		√	-	没有定义
0x40045536H	CAN0报文数据字节67寄存器03	C0MDB6703		-	√	没有定义
0x40045536H	CAN0报文数据字节6寄存器03	C0MDB603		√	-	没有定义
0x40045537H	CAN0报文数据字节7寄存器03	C0MDB703		√	-	没有定义
0x40045538H	CAN0报文数据长度寄存器03	C0MDLC03		√	-	0000xxxxB
0x40045539H	CAN0 报文配置寄存器03	C0MCONF03		√	-	没有定义
0x4004553AH	CAN0报文ID寄存器03	C0MIDL03		-	√	没有定义
0x4004553CH		C0MIDH03	-	√	没有定义	
0x4004553EH	CAN0报文控制寄存器03	C0MCTRL03	-	√	00x00000 000xx000B	

表21-16: 寄存器访问类型 (4/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值		
				8	16			
0x40045540H	CAN0报文数据字节01寄存器04	C0MDB0104	读/写	-	√	没有定义		
	0x40045540H	CAN0报文数据字节0寄存器04		C0MDB004	√	-	没有定义	
	0x40045541H	CAN0报文数据字节1寄存器04		C0MDB104	√	-	没有定义	
0x40045542H	CAN0报文数据字节23寄存器04	C0MDB2304		-	√	没有定义		
	0x40045542H	CAN0报文数据字节2寄存器04		C0MDB204	√	-	没有定义	
	0x40045543H	CAN0报文数据字节3寄存器04		C0MDB304	√	-	没有定义	
0x40045544H	CAN0报文数据字节45寄存器04	C0MDB4504		-	√	没有定义		
	0x40045544H	CAN0报文数据字节4寄存器04		C0MDB404	√	-	没有定义	
	0x40045545H	CAN0报文数据字节5寄存器04		C0MDB504	√	-	没有定义	
0x40045546H	CAN0报文数据字节67寄存器04	C0MDB6704		-	√	没有定义		
	0x40045546H	CAN0报文数据字节6寄存器04		C0MDB604	√	-	没有定义	
	0x40045547H	CAN0报文数据字节7寄存器04		C0MDB704	√	-	没有定义	
0x40045548H	CAN0报文数据长度寄存器04	C0MDLC04		√	-	0000xxxxB		
0x40045549H	CAN0 报文配置寄存器04	C0MCONF04		√	-	没有定义		
0x4004554AH	CAN0报文ID寄存器04	C0MIDL04		-	√	没有定义		
0x4004554CH		C0MIDH04		-	√	没有定义		
0x4004554EH	CAN0报文控制寄存器04	C0MCTRL04		-	√	00x00000 000xx000B		
0x40045550H	CAN0报文数据字节01寄存器05	C0MDB0105		读/写	-	√	没有定义	
	0x40045550H	CAN0报文数据字节0寄存器05			C0MDB005	√	-	没有定义
	0x40045551H	CAN0报文数据字节1寄存器05			C0MDB105	√	-	没有定义
0x40045552H	CAN0报文数据字节23寄存器05	C0MDB2305			-	√	没有定义	
	0x40045552H	CAN0报文数据字节2寄存器05			C0MDB205	√	-	没有定义
	0x40045553H	CAN0报文数据字节3寄存器05			C0MDB305	√	-	没有定义
0x40045554H	CAN0报文数据字节45寄存器05	C0MDB4505			-	√	没有定义	
	0x40045554H	CAN0报文数据字节4寄存器05			C0MDB405	√	-	没有定义
	0x40045555H	CAN0报文数据字节5寄存器05			C0MDB505	√	-	没有定义
0x40045556H	CAN0报文数据字节67寄存器05	C0MDB6705			-	√	没有定义	
	0x40045556H	CAN0报文数据字节6寄存器05			C0MDB605	√	-	没有定义
	0x40045557H	CAN0报文数据字节7寄存器05			C0MDB705	√	-	没有定义
0x40045558H	CAN0报文数据长度寄存器05	C0MDLC05			√	-	0000xxxxB	
0x40045559H	CAN0 报文配置寄存器05	C0MCONF05	√		-	没有定义		
0x4004555AH	CAN0报文ID寄存器05	C0MIDL05	-		√	没有定义		
0x4004555CH		C0MIDH05	-		√	没有定义		
0x4004555EH	CAN0报文控制寄存器05	C0MCTRL05	-		√	00x00000 000xx000B		

表21-16: 寄存器访问类型 (5/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值
				8	16	
0x40045560H	CAN0报文数据字节01寄存器06	C0MDB0106	读/写	-	√	没有定义
0x40045560H	CAN0报文数据字节0寄存器06	C0MDB006		√	-	没有定义
0x40045561H	CAN0报文数据字节1寄存器06	C0MDB106		√	-	没有定义
0x40045562H	CAN0报文数据字节23寄存器06	C0MDB2306		-	√	没有定义
0x40045562H	CAN0报文数据字节2寄存器06	C0MDB206		√	-	没有定义
0x40045563H	CAN0报文数据字节3寄存器06	C0MDB306		√	-	没有定义
0x40045564H	CAN0报文数据字节45寄存器06	C0MDB4506		-	√	没有定义
0x40045564H	CAN0报文数据字节4寄存器06	C0MDB406		√	-	没有定义
0x40045565H	CAN0报文数据字节5寄存器06	C0MDB506		√	-	没有定义
0x40045566H	CAN0报文数据字节67寄存器06	C0MDB6706		-	√	没有定义
0x40045566H	CAN0报文数据字节6寄存器06	C0MDB606		√	-	没有定义
0x40045567H	CAN0报文数据字节7寄存器06	C0MDB706		√	-	没有定义
0x40045568H	CAN0报文数据长度寄存器06	C0MDLC06		√	-	0000xxxxB
0x40045569H	CAN0 报文配置寄存器06	C0MCONF06		√	-	没有定义
0x4004556AH	CAN0报文ID寄存器06	C0MIDL06		-	√	没有定义
0x4004556CH		C0MIDH06		-	√	没有定义
0x4004556EH	CAN0报文控制寄存器06	C0MCTRL06		-	√	00x00000 000xx000B
0x40045570H	CAN0报文数据字节01寄存器07	C0MDB0107		-	√	没有定义
0x40045570H	CAN0报文数据字节0寄存器07	C0MDB007		√	-	没有定义
0x40045571H	CAN0报文数据字节1寄存器07	C0MDB107		√	-	没有定义
0x40045572H	CAN0报文数据字节23寄存器07	C0MDB2307		-	√	没有定义
0x40045572H	CAN0报文数据字节2寄存器07	C0MDB207		√	-	没有定义
0x40045573H	CAN0报文数据字节3寄存器07	C0MDB307		√	-	没有定义
0x40045574H	CAN0报文数据字节45寄存器07	C0MDB4507		-	√	没有定义
0x40045574H	CAN0报文数据字节4寄存器07	C0MDB407		√	-	没有定义
0x40045575H	CAN0报文数据字节5寄存器07	C0MDB507		√	-	没有定义
0x40045576H	CAN0报文数据字节67寄存器07	C0MDB6707		-	√	没有定义
0x40045576H	CAN0报文数据字节6寄存器07	C0MDB607		√	-	没有定义
0x40045577H	CAN0报文数据字节7寄存器07	C0MDB707		√	-	没有定义
0x40045578H	CAN0报文数据长度寄存器07	C0MDLC07		√	-	0000xxxxB
0x40045579H	CAN0 报文配置寄存器07	C0MCONF07		√	-	没有定义
0x4004557AH	CAN0报文ID寄存器07	C0MIDL07		-	√	没有定义
0x4004557CH		C0MIDH07		-	√	没有定义
0x4004557EH	CAN0报文控制寄存器07	C0MCTRL07	-	√	00x00000 000xx000B	

表21-16: 寄存器访问类型 (6/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值	
				8	16		
0x40045580H	CAN0报文数据字节01寄存器08	C0MDB0108	读/写	-	√	没有定义	
	0x40045580H	CAN0报文数据字节0寄存器08		C0MDB008	√	-	没有定义
	0x40045581H	CAN0报文数据字节1寄存器08		C0MDB108	√	-	没有定义
0x40045582H	CAN0报文数据字节23寄存器08	C0MDB2308		-	√	没有定义	
	0x40045582H	CAN0报文数据字节2寄存器08		C0MDB208	√	-	没有定义
	0x40045583H	CAN0报文数据字节3寄存器08		C0MDB308	√	-	没有定义
0x40045584H	CAN0报文数据字节45寄存器08	C0MDB4508		-	√	没有定义	
	0x40045584H	CAN0报文数据字节4寄存器08		C0MDB408	√	-	没有定义
	0x40045585H	CAN0报文数据字节5寄存器08		C0MDB508	√	-	没有定义
0x40045586H	CAN0报文数据字节67寄存器08	C0MDB6708		-	√	没有定义	
	0x40045586H	CAN0报文数据字节6寄存器08		C0MDB608	√	-	没有定义
	0x40045587H	CAN0报文数据字节7寄存器08		C0MDB708	√	-	没有定义
0x40045588H	CAN0报文数据长度寄存器08	C0MDLC08		√	-	0000xxxxB	
0x40045589H	CAN0 报文配置寄存器08	C0MCONF08		√	-	没有定义	
0x4004558AH	CAN0报文ID寄存器08	C0MIDL08		-	√	没有定义	
0x4004558CH		C0MIDH08		-	√	没有定义	
0x4004558EH	CAN0报文控制寄存器08	C0MCTRL08		-	√	00x00000 000xx000B	
0x40045590H	CAN0报文数据字节01寄存器09	C0MDB0109		-	√	没有定义	
	0x40045590H	CAN0报文数据字节0寄存器09		C0MDB009	√	-	没有定义
	0x40045591H	CAN0报文数据字节1寄存器09		C0MDB109	√	-	没有定义
0x40045592H	CAN0报文数据字节23寄存器09	C0MDB2309		-	√	没有定义	
	0x40045592H	CAN0报文数据字节2寄存器09		C0MDB209	√	-	没有定义
	0x40045593H	CAN0报文数据字节3寄存器09		C0MDB309	√	-	没有定义
0x40045594H	CAN0报文数据字节45寄存器09	C0MDB4509		-	√	没有定义	
	0x40045594H	CAN0报文数据字节4寄存器09		C0MDB409	√	-	没有定义
	0x40045595H	CAN0报文数据字节5寄存器09		C0MDB509	√	-	没有定义
0x40045596H	CAN0报文数据字节67寄存器09	C0MDB6709		-	√	没有定义	
	0x40045596H	CAN0报文数据字节6寄存器09		C0MDB609	√	-	没有定义
	0x40045597H	CAN0报文数据字节7寄存器09		C0MDB709	√	-	没有定义
0x40045598H	CAN0报文数据长度寄存器09	C0MDLC09		√	-	0000xxxxB	
0x40045599H	CAN0 报文配置寄存器09	C0MCONF09	√	-	没有定义		
0x4004559AH	CAN0报文ID寄存器09	C0MIDL09	-	√	没有定义		
0x4004559CH		C0MIDH09	-	√	没有定义		
0x4004559EH	CAN0报文控制寄存器09	C0MCTRL09	-	√	00x00000 000xx000B		

表21-16: 寄存器访问类型 (7/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值	
				8	16		
0x400455A0H	CAN0报文数据字节01寄存器10	C0MDB0110	读/写	-	√	没有定义	
	0x400455A0H	CAN0报文数据字节0寄存器10		C0MDB010	√	-	没有定义
	0x400455A1H	CAN0报文数据字节1寄存器10		C0MDB110	√	-	没有定义
0x400455A2H	CAN0报文数据字节23寄存器10	C0MDB2310		-	√	没有定义	
	0x400455A2H	CAN0报文数据字节2寄存器10		C0MDB210	√	-	没有定义
	0x400455A3H	CAN0报文数据字节3寄存器10		C0MDB310	√	-	没有定义
0x400455A4H	CAN0报文数据字节45寄存器10	C0MDB4510		-	√	没有定义	
	0x400455A4H	CAN0报文数据字节4寄存器10		C0MDB410	√	-	没有定义
	0x400455A5H	CAN0报文数据字节5寄存器10		C0MDB510	√	-	没有定义
0x400455A6H	CAN0报文数据字节67寄存器10	C0MDB6710		-	√	没有定义	
	0x400455A6H	CAN0报文数据字节6寄存器10		C0MDB610	√	-	没有定义
	0x400455A7H	CAN0报文数据字节7寄存器10		C0MDB710	√	-	没有定义
0x400455A8H	CAN0报文数据长度寄存器10	C0MDLC10		√	-	0000xxxxB	
0x400455A9H	CAN0 报文配置寄存器10	C0MCONF10		√	-	没有定义	
0x400455AAH	CAN0报文ID寄存器10	C0MIDL10		-	√	没有定义	
0x400455ACH		C0MIDH10		-	√	没有定义	
0x400455AEH	CAN0报文控制寄存器10	C0MCTRL10		-	√	00x00000 000xx000B	
0x400455B0H	CAN0报文数据字节01寄存器11	C0MDB0111		-	√	没有定义	
	0x400455B0H	CAN0报文数据字节0寄存器11		C0MDB011	√	-	没有定义
	0x400455B1H	CAN0报文数据字节1寄存器11		C0MDB111	√	-	没有定义
0x400455B2H	CAN0报文数据字节23寄存器11	C0MDB2311		-	√	没有定义	
	0x400455B2H	CAN0报文数据字节2寄存器11		C0MDB211	√	-	没有定义
	0x400455B3H	CAN0报文数据字节3寄存器11		C0MDB311	√	-	没有定义
0x400455B4H	CAN0报文数据字节45寄存器11	C0MDB4511		-	√	没有定义	
	0x400455B4H	CAN0报文数据字节4寄存器11		C0MDB411	√	-	没有定义
	0x400455B5H	CAN0报文数据字节5寄存器11		C0MDB511	√	-	没有定义
0x400455B6H	CAN0报文数据字节67寄存器11	C0MDB6711		-	√	没有定义	
	0x400455B6H	CAN0报文数据字节6寄存器11		C0MDB611	√	-	没有定义
	0x400455B7H	CAN0报文数据字节7寄存器11		C0MDB711	√	-	没有定义
0x400455B8H	CAN0报文数据长度寄存器11	C0MDLC11		√	-	0000xxxxB	
0x400455B9H	CAN0 报文配置寄存器11	C0MCONF11		√	-	没有定义	
0x400455BAH	CAN0报文ID寄存器11	C0MIDL11		-	√	没有定义	
0x400455BCH		C0MIDH11		-	√	没有定义	
0x400455BEH	CAN0报文控制寄存器11	C0MCTRL11	-	√	00x00000 000xx000B		

表21-16: 寄存器访问类型 (8/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值	
				8	16		
0x400455C0H	CAN0报文数据字节01寄存器12	C0MDB0112	读/写	-	√	没有定义	
	0x400455C0H	CAN0报文数据字节0寄存器12		C0MDB012	√	-	没有定义
	0x400455C1H	CAN0报文数据字节1寄存器12		C0MDB112	√	-	没有定义
0x400455C2H	CAN0报文数据字节23寄存器12	C0MDB2312		-	√	没有定义	
	0x400455C2H	CAN0报文数据字节2寄存器12		C0MDB212	√	-	没有定义
	0x400455C3H	CAN0报文数据字节3寄存器12		C0MDB312	√	-	没有定义
0x400455C4H	CAN0报文数据字节45寄存器12	C0MDB4512		-	√	没有定义	
	0x400455C4H	CAN0报文数据字节4寄存器12		C0MDB412	√	-	没有定义
	0x400455C5H	CAN0报文数据字节5寄存器12		C0MDB512	√	-	没有定义
0x400455C6H	CAN0报文数据字节67寄存器12	C0MDB6712		-	√	没有定义	
	0x400455C6H	CAN0报文数据字节6寄存器12		C0MDB612	√	-	没有定义
	0x400455C7H	CAN0报文数据字节7寄存器12		C0MDB712	√	-	没有定义
0x400455C8H	CAN0报文数据长度寄存器12	C0MDLC12		√	-	0000xxxxB	
0x400455C9H	CAN0 报文配置寄存器12	C0MCONF12		√	-	没有定义	
0x400455CAH	CAN0报文ID寄存器12	C0MIDL12		-	√	没有定义	
0x400455CCH		C0MIDH12		-	√	没有定义	
0x400455CEH	CAN0报文控制寄存器12	C0MCTRL12		-	√	00x00000 000xx000B	
0x400455D0H	CAN0报文数据字节01寄存器13	C0MDB0113		-	√	没有定义	
	0x400455D0H	CAN0报文数据字节0寄存器13		C0MDB013	√	-	没有定义
	0x400455D1H	CAN0报文数据字节1寄存器13		C0MDB113	√	-	没有定义
0x400455D2H	CAN0报文数据字节23寄存器13	C0MDB2313		-	√	没有定义	
	0x400455D2H	CAN0报文数据字节2寄存器13		C0MDB213	√	-	没有定义
	0x400455D3H	CAN0报文数据字节3寄存器13		C0MDB313	√	-	没有定义
0x400455D4H	CAN0报文数据字节45寄存器13	C0MDB4513		-	√	没有定义	
	0x400455D4H	CAN0报文数据字节4寄存器13		C0MDB413	√	-	没有定义
	0x400455D5H	CAN0报文数据字节5寄存器13		C0MDB513	√	-	没有定义
0x400455D6H	CAN0报文数据字节67寄存器13	C0MDB6713		-	√	没有定义	
	0x400455D6H	CAN0报文数据字节6寄存器13		C0MDB613	√	-	没有定义
	0x400455D7H	CAN0报文数据字节7寄存器13		C0MDB713	√	-	没有定义
0x400455D8H	CAN0报文数据长度寄存器13	C0MDLC13		√	-	0000xxxxB	
0x400455D9H	CAN0 报文配置寄存器13	C0MCONF13		√	-	没有定义	
0x400455DAH	CAN0报文ID寄存器13	C0MIDL13		-	√	没有定义	
0x400455DCH		C0MIDH13		-	√	没有定义	
0x400455DEH	CAN0报文控制寄存器13	C0MCTRL13	-	√	00x00000 000xx000B		

表21-16: 寄存器访问类型 (9/9)

地址	寄存器名	Symbol	读/写	位操作单元		默认值
				8	16	
0x400455E0H	CAN0报文数据字节01寄存器14	C0MDB0114	读/写	-	√	没有定义
0x400455E0H	CAN0报文数据字节0寄存器14	C0MDB014		√	-	没有定义
0x400455E1H	CAN0报文数据字节1寄存器14	C0MDB114		√	-	没有定义
0x400455E2H	CAN0报文数据字节23寄存器14	C0MDB2314		-	√	没有定义
0x400455E2H	CAN0报文数据字节2寄存器14	C0MDB214		√	-	没有定义
0x400455E3H	CAN0报文数据字节3寄存器14	C0MDB314		√	-	没有定义
0x400455E4H	CAN0报文数据字节45寄存器14	C0MDB4514		-	√	没有定义
0x400455E4H	CAN0报文数据字节4寄存器14	C0MDB414		√	-	没有定义
0x400455E5H	CAN0报文数据字节5寄存器14	C0MDB514		√	-	没有定义
0x400455E6H	CAN0报文数据字节67寄存器14	C0MDB6714		-	√	没有定义
0x400455E6H	CAN0报文数据字节6寄存器14	C0MDB614		√	-	没有定义
0x400455E7H	CAN0报文数据字节7寄存器14	C0MDB714		√	-	没有定义
0x400455E8H	CAN0报文数据长度寄存器14	C0MDLC14		√	-	0000xxxxB
0x400455E9H	CAN0 报文配置寄存器14	C0MCONF14		√	-	没有定义
0x400455EAH	CAN0报文ID寄存器14	C0MIDL14		-	√	没有定义
0x400455ECH		C0MIDH14		-	√	没有定义
0x400455EEH	CAN0报文控制寄存器14	C0MCTRL14		-	√	00x00000 000xx000B
0x400455F0H	CAN0报文数据字节01寄存器15	C0MDB0115		-	√	没有定义
0x400455F0H	CAN0报文数据字节0寄存器15	C0MDB015		√	-	没有定义
0x400455F1H	CAN0报文数据字节1寄存器15	C0MDB115		√	-	没有定义
0x400455F2H	CAN0报文数据字节23寄存器15	C0MDB2315		-	√	没有定义
0x400455F2H	CAN0报文数据字节2寄存器15	C0MDB215		√	-	没有定义
0x400455F3H	CAN0报文数据字节3寄存器15	C0MDB315		√	-	没有定义
0x400455F4H	CAN0报文数据字节45寄存器15	C0MDB4515		-	√	没有定义
0x400455F4H	CAN0报文数据字节4寄存器15	C0MDB415		√	-	没有定义
0x400455F5H	CAN0报文数据字节5寄存器15	C0MDB515		√	-	没有定义
0x400455F6H	CAN0报文数据字节67寄存器15	C0MDB6715		-	√	没有定义
0x400455F6H	CAN0报文数据字节6寄存器15	C0MDB615		√	-	没有定义
0x400455F7H	CAN0报文数据字节7寄存器15	C0MDB715		√	-	没有定义
0x400455F8H	CAN0报文数据长度寄存器15	C0MDLC15		√	-	0000xxxxB
0x400455F9H	CAN0 报文配置寄存器15	C0MCONF15		√	-	没有定义
0x400455FAH	CAN0报文ID寄存器15	C0MIDL15		-	√	没有定义
0x400455FCH		C0MIDH15	-	√	没有定义	
0x400455FEH	CAN0报文控制寄存器15	C0MCTRL15	-	√	00x00000 000xx000B	

21.5.3 寄存器位配置

表21-17: CAN全局寄存器位配置

地址	Symbol	位7/15	位6/14	位5/13	位4/12	位3/11	位2/10	位1/9	位0/8
0x40045400H	COGMCTRL(W)	0	0	0	0	0	0	0	ClearGOM
0x40045401H		0	0	0	0	0	0	SetEFSD	SetGOM
0x40045400H	COGMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
0x40045401H		MBON	0	0	0	0	0	0	0
0x40045406H	COGMABT(W)	0	0	0	0	0	0	0	ClearABTTRG
0x40045407H		0	0	0	0	0	0	SetABTCLR	SetABTTRG
0x40045406H	COGMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
0x40045407H		0	0	0	0	0	0	0	0
0x40045408H	COGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0
0x40045402H	COGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

注意: 实际的寄存器地址计算参考下面的公式:

寄存器地址=全局寄存器区域偏移(依赖CH)+ 上表中列出的偏移地址

备注: (R): 读的时候

(W): 写的时候

表21-18: CAN 模块寄存器的位配置 (1/2)

地址	Symbol	位7/15	位6/14	位5/13	位4/12	位3/11	位2/10	位1/9	位0/8
0x40045440H	C0MASK1L	CM1ID[7:0]							
0x40045441H		CM1ID[15:8]							
0x40045442H	C0MASK1H	CM1ID[23:16]							
0x40045443H		0	0	0	CM1ID[28:24]				
0x40045444H	C0MASK2L	CM2ID[7:0]							
0x40045445H		CM2ID[15:8]							
0x40045446H	C0MASK2H	CM2ID[23:16]							
0x40045447H		0	0	0	CM2ID[28:24]				
0x40045448H	C0MASK3L	CM3ID[7:0]							
0x40045449H		CM3ID[15:8]							
0x4004544AH	C0MASK3H	CM3ID[23:16]							
0x4004544BH		0	0	0	CM3ID[28:24]				
0x4004544CH	C0MASK4L	CM4ID[7:0]							
0x4004544DH		CM4ID[15:8]							
0x4004544EH	C0MASK4H	CM4ID[23:16]							
0x4004544FH		0	0	0	CM4ID[28:24]				
0x40045450H	C0CTRL(W)	ClearCCE RC	ClearAL	ClearVALI D	ClearPSM ODE1	ClearPSM ODE0	ClearOPM ODE2	ClearOPM ODE1	ClearOPM ODE0
0x40045451H		SetCCERC	SetAL	0	SetPSMOD E1	SetPSMOD E0	SetOPMOD E2	SetOPMOD E1	SetOPMOD E0
0x40045450H	C0CTRL(R)	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0
0x40045451H		0	0	0	0	0	0	RSTAT	TSTAT
0x40045452H	C0LEC(W)	0	0	0	0	0	0	0	0
0x40045452H	C0LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
0x40045453H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
0x40045454H	C0ERC	TEC[7:0]							
0x40045455H		REPS	REC[7:0]						
0x40045456H	C0IE(W)	0	0	ClearCIE5	ClearCIE4	ClearCIE3	ClearCIE2	ClearCIE1	ClearCIE0
0x40045457H		0	0	SetCIE5	SetCIE4	SetCIE3	SetCIE2	SetCIE1	SetCIE0
0x40045456H	C0IE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
0x40045457H		0	0	0	0	0	0	0	0
0x40045458H	C0INTS(W)	0	0	ClearCINT S5	ClearCINT S4	ClearCINT S3	ClearCINT S2	ClearCINT S1	ClearCINT S0
0x40045459H		0	0	0	0	0	0	0	0

注意: 实际的寄存器地址计算参考下面的公式:

寄存器地址=全局寄存器区域偏移(依赖CH)+ 上表中列出的偏移地址

备注: (R): 读时

(W): 写时

表21-18: CAN 模块寄存器的位配置 (2/2)

地址	Symbol	位7/15	位6/14	位5/13	位4/12	位3/11	位2/10	位1/9	位0/8
0x40045458H	C0INTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
0x40045459H		0	0	0	0	0	0	0	0
0x4004545AH	C0BRP	TQPRS[7:0]							
0x4004545CH	C0BTR	0	0	0	0	TSEG1[3:0]			
0x4004545DH		0	0	SJW[1:0]		0	TSEG2[2:0]		
0x4004545EH	C0LIPT	LIPT[7:0]							
0x40045460H	C0RGPT(W)	0	0	0	0	0	0	0	ClearROVF
0x40045461H		0	0	0	0	0	0	0	0
0x40045460H	C0RGPT(R)	0	0	0	0	0	0	RHPM	ROVF
0x40045461H		RGPT[7:0]							
0x40045462H	C0LOPT	LOPT[7:0]							
0x40045464H	C0TGPT(W)	0	0	0	0	0	0	0	ClearTOVF
0x40045465H		0	0	0	0	0	0	0	0
0x40045464H	C0TGPT(R)	0	0	0	0	0	0	THPM	TOVF
0x40045465H		TGPT[7:0]							
0x40045466H	C0TS(W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	ClearTSEN
0x40045467H		0	0	0	0	0	Set TSLOCK	SetTSSEL	SetTSEN
0x40045466H	C0TS(R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
0x40045467H		0	0	0	0	0	0	0	0

注意: 实际的寄存器地址计算参考下面的公式:

寄存器地址=全局寄存器区域偏移(依赖CH)+ 上表中列出的偏移地址

备注: (R): 读时

(W): 写时

表 21-19: 报文缓冲区寄存器的位配置

地址	Symbol	位7/15	位6/14	位5/13	位4/12	位3/11	位2/10	位1/9	位0/8
0x400455x0H	C0MDB01m	报文数据(字节0)							
0x400455x1H		报文数据(字节1)							
0x400455x0H	C0MDB0m	报文数据(字节0)							
0x400455x1H	C0MDB1m	报文数据(字节1)							
0x400455x2H	C0MDB23m	报文数据(字节2)							
0x400455x3H		报文数据(字节3)							
0x400455x2H	C0MDB2m	报文数据(字节2)							
0x400455x3H	C0MDB3m	报文数据(字节3)							
0x400455x4H	C0MDB45m	报文数据(字节4)							
0x400455x5H		报文数据(字节5)							
0x400455x4H	C0MDB4m	报文数据(字节4)							
0x400455x5H	C0MDB5m	报文数据(字节5)							
0x400455x6H	C0MDB67m	报文数据(字节6)							
0x400455x7H		报文数据(字节7)							
0x400455x6H	C0MDB6m	报文数据(字节6)							
0x400455x7H	C0MDB7m	报文数据(字节7)							
0x400455x8H	C0MDLcM	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
0x400455x9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
0x400455xAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x400455xBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
0x400455xCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
0x400455xDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
0x400455xEH	C0MCTRLm(W)	0	0	0	ClearMOW	ClearIE	ClearDN	ClearTRQ	ClearRDY
0x400455xFH		0	0	0	0	SetIE	0	SetTRQ	SetRDY
0x400455xEH	C0MCTRLm(R)	0	0	0	MOW	IE	DN	TRQ	RDY
0x400455xFH		0	0	MUC	0	0	0	0	0

注意: 实际的寄存器地址计算参考下面的公式:

寄存器地址=全局寄存器区域偏移(依赖CH)+ 上表中列出的偏移地址

备注:

- (R): 读时
(W): 写时
- m = 0 到15
x = 0 到F

21.6 位设置/清除功能

CAN 控制寄存器包括可通过 CPU 和 CAN 接口设置或清除其位的寄存器。

如果直接写入以下寄存器，则会发生操作错误。不要直接通过位操作、读取/修改/写入或直接写入目标值来写入任何值。

- CAN 全局控制寄存器(C0GMCTRL)
- CAN全局自动块传输控制寄存器 (C0GMABT)
- CAN 模块控制寄存器(C0CTRL)
- CAN 模块中断使能寄存器 (C0IE)
- CAN模块中断中断状态寄存器(C0INTS)
- CAN 模块接收历史列表寄存器(C0RGPT)
- CAN 模块发送历史列表寄存器(C0TGPT)
- CAN 模块时间戳寄存器(C0TS)
- CAN 报文控制寄存器 (C0MCTRLm)

备注：m = 0到15

上述寄存器中的所有 16 位都可以通过常用方法读取。使用下面的图21-23 中描述的过程设置或清除这些寄存器中较低的 8 位。

上述寄存器中较低 8 位的设置或清除与较高8 位结合执行(参见图21-24 中的写入操作后的 16 位数据)。图 21-23 显示了设置位或清除位的值与相应寄存器中的设置/清除/无更改操作的关系。

图21-23：位设置/清除操作例子

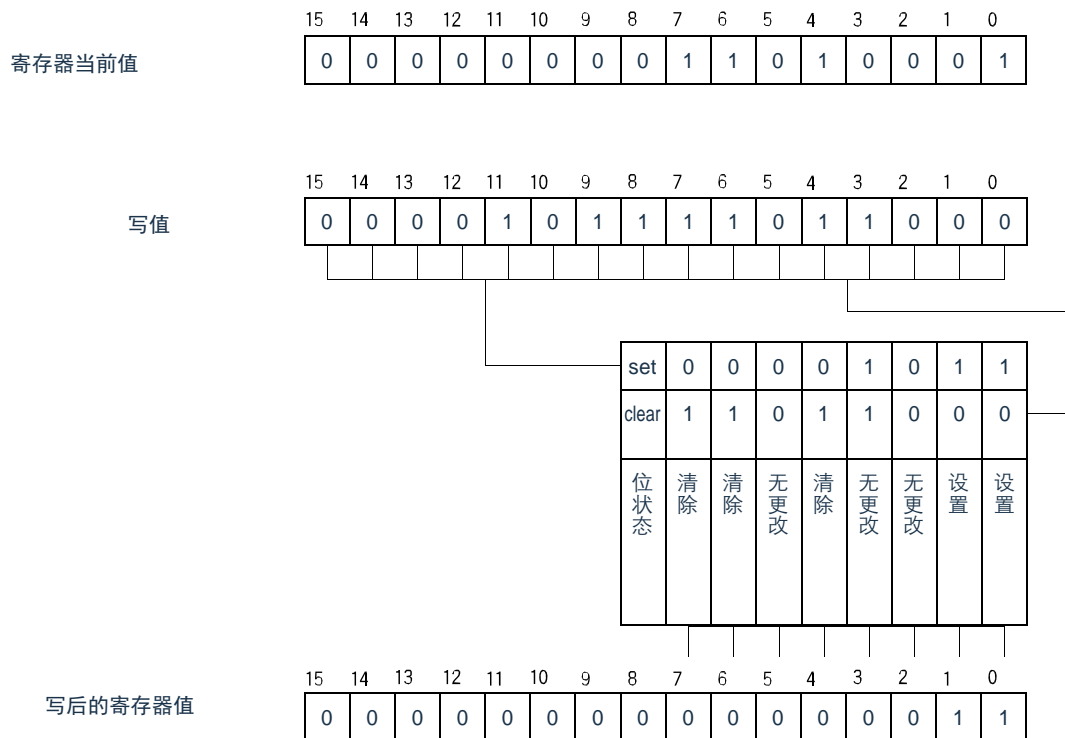


图21-24：写操作中16-位数据

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set7	Set6	Set5	Set4	Set3	Set2	Set1	Set0	Clear7	Clear6	Clear5	Clear4	Clear3	Clear2	Clear1	Clear0

Setn	Clearn	设置/清除后的n状态
0	0	无更改
0	1	0
1	0	1
1	1	无更改

备注：n = 0 到7

21.7 控制寄存器

备注：m = 0 到15

21.7.1 外围时钟选择寄存器 (PER0)

PER0 用于启用或禁用每个外围硬件宏。PER0 可通过 8 位内存操作指令进行设置。重置信号将此寄存器清除到 00H。

注意：在开始操作每个外围硬件设备之前，设置PCKSEL寄存器。

图21-25：外围时钟选择寄存器的格式 (PER0)

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	CAN0EN	TM40EN

CAN0EN	提供CAN模块的输入时钟的控制
0	停止提供输入时钟。 •不能写CAN0使用的SFR。 •CAN0处于复位状态。
1	提供输入时钟。 •能读写CAN0使用的SFR。

21.7.2 CAN全局模块控制寄存器(C0GMCTRL)

C0GMCTRL 寄存器用于控制CAN模块的操作。

图21-26: CAN全局模块控制寄存器格式 (C0GMCTRL)(1/2)

(a) 读

	15	14	13	12	11	10	9	8
C0GMCTRL	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(b) 写

	15	14	13	12	11	10	9	8
C0GMCTRL	0	0	0	0	0	0	Set EFSD	Set GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear GOM

(a) 读

MBON	报文缓存寄存器,传输/接收历史列表寄存器访问的位使能
0	禁止对报文缓存寄存器和发送/接收历史列表寄存器的读写
1	使能对报文缓存寄存器和发送/接收历史列表寄存器的读写

注意:

1. 当 MBON 位被清除(到 0), 软件访问报文缓冲区(C0MDB0m, C0MDB1m, C0MDB01m, C0MDB2m, C0MDB3m, C0MDB23m, C0MDB4m, C0MDB5m, C0MDB45m, C0MDB6m, C0MDB7m, C0MDB67m, C0MDLCm, C0MCONFm, C0MIDLm, C0MIDHm, and C0MCTRLm)或发送历史记录或接收历史记录寄存器(C0LOPT、C0TGPT、C0LIPT 和 C0RGPT)相关的会被禁用。
2. 此位是只读的, 当MBON 是0的时候对它写1, MBON的值不会改变, 对报文缓存寄存器或寄存器相关的发送历史或接收历史的访问保持禁止。

备注: 当 CAN 模块进入 CAN 睡眠模式/CAN 停止模式或 GOM 位清除(至 0)时, MBON 位将被清除(至 0)。

释放 CAN 睡眠模式/CAN 停止模式或设置 GOM 位时, MBON 位被设置为 1。

图21-26: CAN全局模块控制寄存器格式(C0GMCTRL)(2/2)

EFSD	位使能强制关断
0	禁止通过写GOM=0来关断
1	使能通过写GOM=0来关断.

注意: 要请求强制关闭, 必须在后续操作中清除GOM位为0, 在EFSD位设置为1后立即写入。如果在EFSD位设置为1后没有立即清除GOM位的情况下, 执行对另一个寄存器的访问(包括读取C0GMCTRL寄存器), 则EFSD位会被强制清除为0, 并且强制关闭请求会无效。执行DMA时, 可能会忽略强制关闭的请求。请务必读取EFSD位, 并在发出强制关闭请求之前确认已启用强制关闭。如果由于正在执行DMA而无法启用强制关闭, 建议暂时停止DMA。

GOM	全局操作模式位
0	CAN模块操作中被禁用
1	CAN模块被启动

注意: GOM位仅在初始化模式或紧接EFSD被设置为1后被清0。

(b) 写

SetEFSD	EFSD位设置
0	EFSD位没有改变
1	EFSD 被设置为1

SetGOM	ClearGOM	GOM位设置
0	1	GOM位清除为0
1	0	GOM位设置为1
其他值		没有改变GOM位

注意: 单独的设置GOM位和EFSD位。

21.7.3 CAN全局模块时钟选择寄存器(C0GMCS)

C0GMCS和C1GMCS用于选择CAN模块的系统时钟。

图21-27: CAN全局模块时钟选择寄存器格式(C0GMCS)

	7	6	5	4	3	2	1	0
C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN模块系统时钟 (F _{CANMOD})
0	0	0	0	F _{CAN/1}
0	0	0	1	F _{CAN/2}
0	0	1	0	F _{CAN/3}
0	0	1	1	F _{CAN/4}
0	1	0	0	F _{CAN/5}
0	1	0	1	F _{CAN/6}
0	1	1	0	F _{CAN/7}
0	1	1	1	F _{CAN/8}
1	0	0	0	F _{CAN/9}
1	0	0	1	F _{CAN/10}
1	0	1	0	F _{CAN/11}
1	0	1	1	F _{CAN/12}
1	1	0	0	F _{CAN/13}
1	1	0	1	F _{CAN/14}
1	1	1	0	F _{CAN/15}
1	1	1	1	F _{CAN/16} (默认值)

备注: F_{CAN}: 提供给CAN(F_{MAIN})的时钟

21.7.4 CAN全局自动块传输控制寄存器 (COGMABT)

COGMABT寄存器用于控制自动块传输(ABT)操作

图21-28: CAN全局自动块传输控制寄存器的格式(COGMABT)(1/2)

(a) 读

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

(b) 写

	15	14	13	12	11	10	9	8
COGMABT	0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	Clear ABTTRG

注意: 在将ABT的正常运行模式更改为初始化模式之前, 请确保将COGMABT寄存器设置为默认值(0000H), 并确认COGMABT寄存器肯定已初始化为默认值(0000H)。

(a) 读

ABTCLR	自动块传输清楚状态位
0	清除自动传输引擎完成
1	自动传输引擎正在被清除

备注:

1. 当ABTTRG 被清除为0时, 设置ABTCLR位为1。
当ABTTRG为1, ABTCLR为1时的操作是没有保证的。
2. 当通过将ABTCLR位设置为1清除自动块传输引擎时, 一旦请求的清除处理完成, ABTCLR 位将自动清除为0。

图21-28: CAN全局自动块传输控制寄存器的格式 (COGMABT)(2/2)

ABTTRG	自动块传输状态位
0	自动块传输被停止了
1	自动块传输正在执行

注意: 请勿在初始化模式下设置ABTTRG位(ABTTRG = 1)。如果在初始化模式下设置ABTTRG位, 则在CAN模块使用ABT进入正常操作模式后, 无法保证操作。在COCTRL.TSTAT为1时, 不要设置ABTTRG 位(1)。在设置ABTTRG之前, 请提前确认TSTAT = 0。

(b) 写

SetABTCLR	自动块传输引擎清除请求位
0	自动块传输引擎正处于空闲状态或正在操作
1	申请清除自动块传输。当自动块传输引擎被清除了，自动块传输在设置ABTTRG为1后从报文缓存0启动

SetABTTRG	ClearABTTRG	自动块传输开始位
0	1	申请停止自动块传输
1	0	申请启动自动块传输
其他值		ABTTRG位不改变.

注意：当从其他节点接收报文或发送ABT报文以外的报文(报文缓冲区8到15)时，即使ABTTRG位设置为1，也可能无法立即开始传输。

即使ABTTRG位已清除为0，传输也不会中止，直到当前正在传输的ABT报文传输完成(成功与否)。之后，传输中止。

21.7.5 CAN全局自动块传输延时设置寄存器(C0GMABTD)

C0GMABTD 寄存器用于设置，分配到ABT的报文缓冲区数据，在ABT正常操作模式下传输的时间间隔。

图21-29: CAN全局自动块传输延时设置寄存器的格式(C0GMABTD)

	7	6	5	4	3	2	1	0
C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	自动块传输的数据帧间隔 (单位: 数据位时间(DBT))
0	0	0	0	0 DBT (默认值)
0	0	0	1	25DBT
0	0	1	0	26DBT
0	0	1	1	27DBT
0	1	0	0	28DBT
0	1	0	1	29DBT
0	1	1	0	210DBT
0	1	1	1	211DBT
1	0	0	0	212DBT
其他值				禁止

注意:

1. 当ABTTRG为1时，不要改变C0GMABTD的内容。
2. ABT报文实际传输到CAN总线的时间，因从其他工作站传输的状态或请求传输ABT报文(报文缓冲区8到15)的方式而异。

21.7.6 CAN模块屏蔽寄存器(COMASKaL, COMASKaH) (a = 1, 2, 3, or 4)

COMASKaL和COMASKaH寄存器通过报文ID比较的屏蔽部分和使屏蔽部分的ID无效, 来扩展进入同一报文缓存的数量。

图21-30: CAN模块屏蔽寄存器的格式(COMASKaL, COMASKaH) (a = 1, 2, 3, or 4)(1/2)

- CAN模块屏蔽1寄存器(COMASK1L, COMASK1H)

	15	14	13	12	11	10	9	8
COMASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
COMASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN模块屏蔽2寄存器(COMASK2L, COMASK2H)

	15	14	13	12	11	10	9	8
COMASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
COMASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

图21-30: CAN模块屏蔽寄存器的格式(COMASKaL, COMASKaH) (a = 1, 2, 3, or 4)(2/2)

- CAN模块屏蔽3寄存器(COMASK3L, COMASK3H)

	15	14	13	12	11	10	9	8
COMASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
COMASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN模块屏蔽4寄存器(COMASK4L, COMASK4H)

	15	14	13	12	11	10	9	8
COMASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

	15	14	13	12	11	10	9	8
COMASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID0到CMID28	设置ID位的屏蔽模式
0	CMID0 到 CMID28设置的报文缓冲区的ID 位与接收的报文帧的ID 位进行比较。
1	CMID0 到 CMID28设置的报文缓冲区的ID位不与接收的报文帧的ID 位进行比较(它们被屏蔽)

备注: 屏蔽始终由29位的ID长度决定。如果将掩码分配给具有标准ID的邮件, 则忽略CMID0到CMID17位; 因此, 仅屏蔽已接收ID的CMID18到CMID28。相同的掩码可用于标准和扩展的ID。

21.7.7 CAN模块控制寄存器(COCTRL)

COCTRL寄存器用于控制CAN模块的操作模式。

图21-31：CAN模块控制寄存器的格式(COCTRL)(1/4)

(a) 读

	15	14	13	12	11	10	9	8
COCTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

(b) 写

	15	14	13	12	11	10	9	8
COCTRL	Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
	7	6	5	4	3	2	1	0
	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0

(a) 读

RSTAT	接收状态位
0	接收被停止
1	接收正在进行

备注：在以下条件(时序)，RSTAT位被设置为1：

- 接收帧的SOF位被侦测
- 传输帧中仲裁丢失出现

在以下条件(时序)，RSTAT位被清0：

- 在帧间空间的第二个位侦测到隐性位
- 到初始模式的转换，在帧间空间的第一个位

图21-31：CAN模块控制寄存器的格式(C0CTRL)(2/4)

TSTAT	传输状态位
0	传输被停止
1	传输正在进行中

备注：在以下条件(时序)，RSTAT位被设置为1：

- 发送帧的SOF位被侦测

在以下条件(时序)，RSTAT位被清0：

- 在过渡到总线关闭状态期间
- 传输帧中仲裁丢失出现
- 在帧间空间的第二个位侦测到了隐性位
- 到初始模式的转换，在帧间空间的第一个位出现

CCERC	错误计数器清除位
0	在初始化模式C0ERC和C0INFO寄存器没有被清除。
1	C0ERC和C0INFO寄存器在初始化模式被清除

备注：

1. CCERC 位用于清除 C0ERC 和 C0INFO 寄存器，以便从总线关闭状态重新初始化或强制恢复。此位只能在初始化模式下设置为 1。
2. 清除 C0ERC 和 C0INFO 寄存器后，CCERC 位也自动清除为 0。
3. 在请求将初始化模式更改为操作模式时，CCERC 位会被设置为1。
4. 从自检模式进入INIT模式后立即将 CCERC 位设置为1的时候，接收数据可能会被损坏。

AL	在仲裁丢失时设置操作的位
0	在单次模式下发生仲裁损失时，不执行重新传输
1	在单次模式下发生仲裁损失时执行重新传输。

备注：AL位仅在单次模式下有效。

VALID	有效的接收报文帧侦测位
0	自上次将 VALID 位清除为 0 以来，尚未收到有效的报文帧
1	将 VALID 位清除为 0 后，收到有效的报文帧

备注：

1. 检测有效的接收报文帧不依赖于接收报文缓冲区(数据帧)或发送报文缓冲区(远程帧)中的存储。
2. 在将初始化模式更改为操作模式之前，清除 VALID 位
3. 如果只有两个 CAN 节点连接到 CAN 总线，一个在正常操作模式下传输报文帧，另一个处在仅接收模式，则在传输节点进入错误被动状态之前，VALID 位不会被设置为1，因为在仅接收模式下不会生成应答。
4. 为了清除VALID位，先将Clear VALID位设为1，然后确认VALID位被清除没有，如果还是没有被清除，再次执行清除处理。

图21-31：CAN模块控制寄存器的格式(C0CTRL)(3/4)

PSMODE1	PSMODE0	省电模式
0	0	没有省电模式被选择
0	1	CAN睡眠模式
1	0	设置禁止
1	1	CAN停止模式

注意：

1. 转换到或则从CAN停止模式唤醒，都要通过CAN睡眠模式。
申请直接执行CAN停止模式的进出 是会被忽略的。
2. 在退出省电模式后，必须检查 C0GMCTRL 的 MBON 标志，然后再访问报文缓冲区。
3. CAN 睡眠模式的请求保持挂起，直到软件取消或进入适当的总线条件(总线空闲)。软件可以通过读取 PSMODE 来检查实际状态

OPMODE2	OPMODE1	OPMODE0	操作模式
0	0	0	没有选择操作模式(CAN模式在初始化模式)
0	0	1	正常的操作模式
0	1	0	有自动块传输的正常操作模式 (有ABT的正常操作模式)
0	1	1	仅接收模式
1	0	0	单次模式
1	0	1	自检模式
其他值			禁止设置

注意：转移到初始化模式或省电模式可能需要一些时间。请务必通过读取来验证模式更改是否成功，然后再继续。

备注：在CAN睡眠和CAN停止模式下OPMODE[2:0]位是只读的。

(b) 写

SetCCERC	设置CCERC位
1	设置CCERC位为1
0	CCERC保持

SetAL	ClearAL	设置AL位
0	1	AL 被清0.
1	0	AL 被设置为1
其他值		AL保持不变

图21-31：CAN模块控制寄存器的格式(C0CTRL)(4/4)

ClearVALID	设置VALID位
0	VALID保持
1	VALID 被清0.

Set PSMODE0	Clear PSMODE0	设置PSMODE0位
0	1	PSMODE0被清0.
1	0	PSMODE0被设置1.
其他值		PSMODE0不改变.

Set PSMODE1	Clear PSMODE1	设置PSMODE1位
0	1	PSMODE1清0
1	0	PSMODE1置1.
其他值		PSMODE1不改变.

Set OPMODE0	Clear OPMODE0	设置OPMODE0位
0	1	OPMODE0 清0.
1	0	OPMODE0 置1.
其他值		OPMODE0 不改变.

Set OPMODE1	Clear OPMODE1	设置OPMODE1位
0	1	OPMODE1 清0.
1	0	OPMODE1置1.
其他值		OPMODE1不改变.

Set OPMODE2	Clear OPMODE2	设置OPMODE2位
0	1	OPMODE2清0.
1	0	OPMODE2 置1.
其他值		OPMODE2不改变.

21.7.8 CAN模块上次错误代码寄存器 (COLEC)

COLEC 寄存器提供 CAN 协议的错误信息。

图21-32：CAN模块上次错误代码寄存器的格式(COLEC)

	7	6	5	4	3	2	1	0
COLEC	0	0	0	0	0	LEC2	LEC1	LEC0

备注：

1. 当CAN从操作模式切换到初始化模式后，COLEC寄存器的内容不会被清除。
2. 如果尝试通过软件写00H以外的值到COLEC寄存器，访问会被忽略。

LEC2	LEC1	LEC0	上次CAN协议错误信息
0	0	0	没有错误
0	0	1	填充错误
0	1	0	格式错误
0	1	1	ACK错误
1	0	0	位错误(CAN 模块尝试作为传输报文的一部分传输隐性位(仲裁字段除外)，但 CAN 总线上的值是显性位)
1	0	1	位错误(CAN模块尝试传输显性位作为传输报文、ACK 位、错误帧或重载帧的一部分，但 CAN 总线上的值是隐性位)
1	1	0	CRC错误
1	1	1	没有定义

21.7.9 CAN模块信息寄存器(COINFO)

COINFO寄存器表明CAN模块的状态

图21-33: CAN模块信息寄存器的格式(COINFO)

	7	6	5	4	3	2	1	0
COINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	总线关闭状态位	
0	非总线关闭状态(传输错误计数器 小于255)(传输计数器的值小于256)	
1	总线关闭状态(传输错误计数器大于255)(传输计数器的值等于或大于256)	

TECS1	TECS0	发送错误计数器状态位
0	0	发送错误计数器的值小于警告水平(<96).
0	1	发送错误计数器的值在警告范围内(96到127).
1	0	没有定义
1	1	发送错误计数器的值在错误被动或总线关闭状态内(≥ 128).

RECS1	RECS0	接收错误计数器状态位
0	0	接收错误计数器的值小于警告水平(<96)
0	1	接收错误计数器的值在警告范围内(96到127).
1	0	没有定义
1	1	接收错误计数器的值在错误被动范围内(≥ 128).

21.7.10 CAN模块错误计数器寄存器(C0ERC)

C0ERC寄存器记录发送/接收错误计数器的计数值。

图21-34：CAN模块错误计数器寄存器格式 (C0ERC)

	15	14	13	12	11	10	9	8
C0ERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	接收错误被动状态位
0	接收错误计数器不是错误被动 (<128)
1	接收错误计数器在错误被动范围内 (≥128)

REC6-REC0	接收错误计数器位
0-127	接收错误数。这些位反映接收错误计数器的状态。错误数由 CAN 协议定义。

备注：接收错误计数器的REC [6:0] 在接收错误被动状态(RECS [1:0] =11B)是无效的

TEC7-TEC0	发送错误计数器位
0-255	传输错误数。这些位反映传输错误计数器的状态。错误数由 CAN 协议定义。

备注：发送错误计数器的TEC [7:0] 在总线关闭状态(BOFF =1)是无效的

21.7.11 CAN模块中断使能寄存器(C0IE)

C0IE寄存器用于使能或禁止CAN模块的中断。

图21-35：CAN模块中断使能寄存器格式(C0IE)(1/2)

(a) 读

	15	14	13	12	11	10	9	8
C0IE	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) 写

	15	14	13	12	11	10	9	8
C0IE	0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
	7	6	5	4	3	2	1	0
	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

(a) 读

CIE5-CIE0	CAN模块中断使能位
0	与中断状态寄存器 C0INTS[5: 0]位对应的中断输出已禁用
1	与中断状态寄存器 C0INTS[5: 0]位对应的中断输出已启用

(b) 写

SetCIE5	ClearCIE5	设置CIE5位
0	1	CIE5 清0.
1	0	CIE5 置1.
其他值		CIE5 没有改变.

SetCIE4	ClearCIE4	设置CIE4位
0	1	CIE4 位清0.
1	0	CIE4 位置1.
其他值		CIE4 没有变化.

图21-35: CAN模块中断使能寄存器格式(C0IE)(2/2)

SetCIE3	ClearCIE3	设置CIE3位
0	1	CIE3清0.
1	0	CIE3置1.
其他值		CIE3 没有变化.

SetCIE2	ClearCIE2	设置CIE2位
0	1	CIE2 清0.
1	0	CIE2置1.
其他值		CIE2 没有变化.

SetCIE1	ClearCIE1	设置CIE1位
0	1	CIE1清0.
1	0	CIE1置1.
其他值		CIE1没有变化.

SetCIE0	ClearCIE0	设置CIE0位
0	1	CIE0清0.
1	0	CIE0置1.
其他值		CIE0没有变化.

21.7.12 CAN模块中断状态寄存器(COINTS)

COINTS寄存器表明CAN模块中断状态

图21-36: CAN模块中断状态寄存器格式(COINTS)

(a) 读

	15	14	13	12	11	10	9	8
COINTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) 写

	15	14	13	12	11	10	9	8
COINTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

(a) 读

CINTS5-CINTS0	CAN中断状态位
0	没有相关的中断源事件处于挂起状态.
1	一个相关的中断源事件在挂起.

Interrupt StatusBit	相关的中断源事件
CINTS5	从CAN睡眠模式唤醒 ^{注1}
CINTS4	仲裁丢失中断
CINTS3	CAN协议错误中断
CINTS2	CAN错误状态中断
CINTS1	有效报文帧接收到报文缓冲区 m 的完成中断
CINTS0	从报文缓冲区 m 正常完成报文帧传输的中断

注1: 仅当CAN模块通过CAN总线操作从CAN睡眠模式唤醒时, 才设置CINTS5位。当软件释放CAN睡眠模式时, CINTS5位不会被设置。

(b) 写

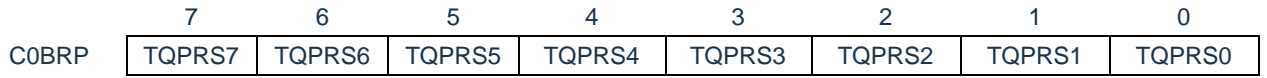
Clear CINTS5-CINTS0	CINTS0到CINTS5位
0	CINTS0到CINTS5没有改变
1	CINTS0到CINTS5被清0

注意: 当中断处理中需要确认每个状态时, 请用软件清除此寄存器的状态位, 因为这些位不会自动清除.

21.7.13 CAN模块位率缩放寄存器(C0BRP)

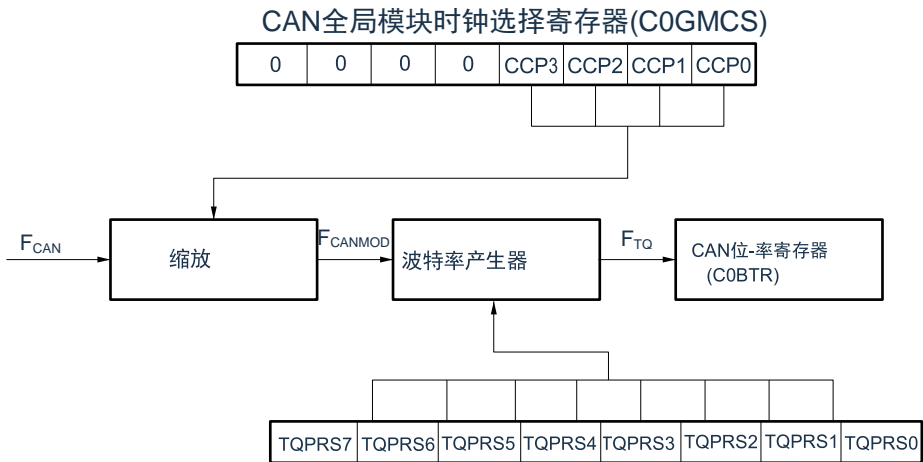
C0BRP 寄存器用于选择CAN协议层基本时钟(F_{TQ})。波特速率设置为 C0BTR 寄存器

图21-37: CAN模块位率缩放寄存器格式(C0BRP)



TQPRS7-TQPRS0	CAN协议层基本系统时钟(F_{TQ})
0	$F_{CANMOD}/1$
1	$F_{CANMOD}/2$
:	:
n	$F_{CANMOD}/(n+1)$
:	:
255	$F_{CANMOD}/256$ (默认值)

图21-38: CAN全局时钟



注意: C0BRP寄存器仅在初始化模式下能被写入

备注: F_{CAN} : 给CAN 提供时钟 (F_{MAIN}) F_{CANMOD} : CAN模块系统时钟

F_{TQ} : CAN协议层基本系统时钟

21.7.14 CAN模块位率寄存器(C0BTR)

C0BTR 寄存器用于控制波特速率的数据位时间。

图21-39: CAN模块位率寄存器格式(C0BTR)(1/2)

	15	14	13	12	11	10	9	8
C0BTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

SJW1	SJW0	同步跳宽的长度
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (默认值)

TSEG22	TSEG21	TSEG20	时间段2的长度
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (默认值)

图21-39: CAN模块位率寄存器格式(C0BTR)(2/2)

TSEG 13	TSEG 12	TSEG 11	TSEG 10	时间段1的长度
0	0	0	0	禁止设置
0	0	0	1	2TQ ^{注1}
0	0	1	0	3TQ ^{注1}
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (默认值)

注1: 这些设置必须在C0BRP寄存器为00H时执行

备注: $TQ = 1/F_{TQ}$ (F_{TQ} : CAN协议层基本系统时钟)



21.7.15 CAN模块上次输入指针寄存器(COLIPT)

COLIPT 寄存器指出上次存储数据帧或远程帧的报文缓冲区数。

图21-41：CAN模块上次输入指针寄存器格式(COLIPT)

	7	6	5	4	3	2	1	0
COLIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7-LIPT0	上次输入指针寄存器(COLIPT)
0 到15	读取 COLIPT 寄存器时，将读取由接收历史列表的最后一个输入指针 (LIPT) 索引的元素的内容。这些内容指示上次存储数据帧或远程帧的报文缓冲区数。

备注：如果数据帧或远程帧从未存储在报文缓冲区中，则 COLIPT 寄存器的读取值未定义。如果 CAN 模块从初始化模式更改为操作模式后，CORGPT 寄存器的 RHPM 位设置为 1，则 COLIPT 寄存器的读取值未定义。

21.7.16 CAN模块接收历史列表寄存器 (C0RGPT)

C0RGPT寄存器用于读取接收的历史列表。

图21-42: CAN模块接收历史列表寄存器格式 (C0RGPT)(1/2)

(a) 读

	15	14	13	12	11	10	9	8
C0RGPT	RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	RHPM	ROVF

(b) 写

	15	14	13	12	11	10	9	8
C0RGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ROVF

(a) 读

RGPT7-RGPT0	接收历史列表获取指针
0 到15	读取C0RGPT寄存器时，通过接收历史列表获取指针(RGPT)索引元素的内容被读。这些内容指示存储数据帧或远程帧的报文缓冲区数。

RHPM ^{注1}	接收历史列表指针匹配
0	接收历史列表至少有一个尚未读取的报文缓冲区号
1	接收历史列表没有尚未读取的报文缓冲区号

注1: 当RHPM 为1时, RGPT0 到 RGPT7的读值无效

ROVF ^{注1}	接收历史列表溢出位
0	将保留所有尚未读取的报文缓冲区编号。所有新数据帧或远程帧里被接收和储存的报文缓存号已经记录到接收历史列表(接收历史记录列表具有一个空元素)
1	自主机处理器上次为RHL提供服务以来(即读取 C0RGPT), 已存储了至少 23 个条目。前 22 个条目按顺序存储, 而每当存储新接收的报文时, 最后一个条目都可以被覆盖, 因为当设置ROVF 位时, 所有缓冲区编号都存储在 LIPT-1 的位置。因此, 接收的顺序现在不能完全恢复。

注1: 如果设置了 ROVF, 则 RHPM 不再在报文存储时清除, 但如果软件读取C0RGPT的所有条目, 则仍设置 RHPM

图21-42: CAN模块接收历史列表寄存器格式(C0RGPT)(2/2)

(b) 写

ClearROVF	设置ROVF位
0	ROVF不改变.
1	ROVF清0.

21.7.17 CAN模块上次输出指针寄存器(COLOPT)

C0LOPT 寄存器指示上次将数据帧或远程帧传输到的报文缓冲区数。

图21-43: CAN模块上次输出指针寄存器格式 (C0LOPT)

	7	6	5	4	3	2	1	0
C0LOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7-LOPT0	发送历史列表上次输出指针(LOPT)
0 到15	读取 C0LOPT 寄存器时，将读取由接收历史记录列表的最后一个出指针 (LOPT) 索引的元素的内容。这些内容指示上次将数据帧或远程帧传输到的报文缓冲区数。

备注：如果数据帧或远程帧从未从报文缓冲区传输，则从 C0LOPT 寄存器读取的值未定义。如果 CAN 模块从初始化模式更改为操作模式后，THPM 位设置为 1，则 C0LOPT 寄存器的读取值未定义。

21.7.18 CAN模块发送历史列表寄存器(C0TGPT)

C0TGPT寄存器用于读出发送历史列表。

图21-44: CAN模块发送历史列表寄存器格式(C0TGPT)(1/2)

(a) 读

	15	14	13	12	11	10	9	8
C0TGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

(b) 写

	15	14	13	12	11	10	9	8
C0TGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear TOVF

(a) 读

TGPT7-TGPT0	发送历史列表读指针
0 到15	读取 C0TGPT 寄存器时, 将读取发送历史列表的读取指针 (TGPT) 索引元素的内容。这些内容指示上次将数据帧或远程帧传输到的报文缓冲区数。

THPM1	Transmit History PointerMatch
0	发送历史列表至少有一个尚未读取的报文缓冲区号。
1	发送历史列表至没有尚未读取的报文缓冲区号。

注1: 当THPM 为1时, TGPT0到TGPT7的读值无效。

TOVF	发送历史列表溢出位
0	将保留所有尚未读取的报文缓冲区编号。新数据帧或远程帧传输到的报文缓冲区的所有编号都记录到传输历史记录列表(传输历史记录列表具有空元素)。
1	自主机处理器上次服务THL 以来, 已存储至少 7 个条目(即读取 C0TGPT)。前 6 个条目按顺序存储, 而每当新传输报文时, 最后一个条目都可以被覆盖, 因为当设置 TOVF 位时, 所有缓冲区编号都存储在 LOPT-1 的位置。因此, 传输序列现在不能完全恢复。

如果设置了 TOVF, 则在报文传输时不再清除THPM, 但如果软件读取CTGPT的所有条目, 则 THPM 仍被设置。

备注: 在具有 ABT 的正常操作模式下, 不会将报文缓冲区 0 传输到 7 的传输历史记录放入列表

图21-44: CAN模块发送历史列表寄存器格式(C0TGPT)(2/2)

(b) 写

ClearTOVF	设置TOVF位
0	TOVF没有改变
1	TOVF 清0.

21.7.19 CAN模块时间戳寄存器(COTS)

COTS寄存器用于控制时间戳功能

图21-45: CAN模块时间戳寄存器格式(COTS)(1/2)

(a) 读

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

(b) 写

	15	14	13	12	11	10	9	8
COTS	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

备注: 当 CAN 模块处于具有 ABT 的正常操作模式时, 不得使用时间戳功能的锁定功能。

(a) 读

TSLOCK	时间戳锁定功能使能位
0	时间戳锁定功能停止. 每次发生选定的时间戳捕获事件时, 都会切换 TSOUT 信号.
1	时间戳锁定功能使能. 每次发生选定的时间戳捕获事件时, 都会切换 TSOUT 信号。但是, 当数据帧正确接收到报文缓冲区 0 时, TSOUT 输出信号被锁定 ^{注1} .

注1: TSEN位自动清0.

图21-45: CAN模块时间戳寄存器格式(C0TS)(2/2)

TSSEL	时间戳捕获事件选择位
0	时间戳捕获事件在SOF.
1	时间戳捕获事件在EOF的最后一个位

TSEN	TSOUT信号操作设置位
0	TSOUT信号翻转操作禁止.
1	TSOUT信号翻转操作使能.

备注: 信号 TSOUT 从 CAN 宏输出到计时器资源, 具体取决于实现。请参阅第 6 章TIMER 阵列单元.

(b) 写

SetTSLOCK	ClearTSLOCK	TSLOCK设置位
0	1	TSLOCK 清0.
1	0	TSLOCK 置1.
其他值		TSLOCK 不改变.

SetTSSEL	ClearTSSEL	TSSEL设置位
0	1	TSSEL清0.
1	0	TSSEL置1.
其他值		TSSEL不改变

SetTSEN	ClearTSEN	TSEN设置位
0	1	TSEN 清0.
1	0	TSEN 置1.
其他值		TSEN 不改变.

21.7.20 CAN报文数据字节寄存器 (C0MDBxm) (x = 0 到7), (C0MDBzm) (z = 01, 23, 45,67)

C0MDBxm、C0MDBzm 寄存器用于存储发送/接收报文的数据。C0MDBxm 寄存器可以以 8 位为单位访问。C0MDBzm 寄存器可以访问 16 位单元的 C0MDBxm 寄存器。

图21-46: CAN报文数据字节寄存器格式(C0MDBxm) (x = 0 到7),
(C0MDBzm) (z = 01, 23, 45, 67)(1/2)

地址: 参考表21-16 复位值: 没有定义 R/W

- C0MDBxm寄存器

	7	6	5	4	3	2	1	0
C0MDB0m	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00
	7	6	5	4	3	2	1	0
C0MDB1m	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10
	7	6	5	4	3	2	1	0
C0MDB2m	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20
	7	6	5	4	3	2	1	0
C0MDB3m	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30
	7	6	5	4	3	2	1	0
C0MDB4m	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40
	7	6	5	4	3	2	1	0
C0MDB5m	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50
	7	6	5	4	3	2	1	0
C0MDB6m	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60
	7	6	5	4	3	2	1	0
C0MDB7m	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

备注: m = 0到15

图21-46: CAN报文数据字节寄存器格式(C0MDBxm) (x = 0 到7),
(C0MDBzm) (z = 01, 23, 45, 67)(2/2)

- C0MDBzm寄存器

	15	14	13	12	11	10	9	8
C0MDB01m	MDATA01	MDATA01	MDATA00	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01	MDATA01
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
C0MDB23m	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23	MDATA23
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
C0MDB45m	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45	MDATA45
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
C0MDB67m	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67
	15	14	13	12	11	10	9	8
	7	6	5	4	3	2	1	0
	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67	MDATA67
	7	6	5	4	3	2	1	0

备注: m = 0 到15

21.7.21 CAN报文数据长度寄存器m (C0MDLcM)

C0MDLcM 寄存器用于设置报文缓冲区的数据段的字节数。

图21-47: CAN报文数据长度寄存器m格式 (C0MDLcM)

地址: 参考表21-16 复位值: 0000xxxxB R/W

	7	6	5	4	3	2	1	0
C0MDLcM	0	0	0	0	MDLc3	MDLc2	MDLc1	MDLc0

MDLc3	MDLc2	MDLc1	MDLc0	发送/接收报文的 数据长度
0	0	0	0	0字节
0	0	0	1	1字节
0	0	1	0	2字节
0	0	1	1	3字节
0	1	0	0	4字节
0	1	0	1	5字节
0	1	1	0	6字节
0	1	1	1	7字节
1	0	0	0	8字节
1	0	0	1	设置禁止 (如果在传输过程中设置了这些位, 则传输 8 字节数据, 而不考虑传输数据帧时设置的 DLC 值。但是, 实际传输到 CAN 总线的 DLC 是设置为此寄存器的 DLC 值) ^{注1}
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

注1: 实际传输到 CAN 总线的数据和 DLC 值如下。

发送帧的类型	发送数据的长度	DLC传输
数据帧	被DLC制定的字节数 (无论如何, 8 字节如果 DLC ≥ 8)	MDLc[3:0]
远程帧	0字节	

注意:

1. 务必把位7设置为40000B.
2. 接收数据根据接收帧的DLC对应字节数(但是, 上限为8)储存在C0MDBxm 中。C0MDBxm没有数据储存就是未定义。

备注: m = 0 到15

21.7.22 CAN报文配置寄存器 (COMCONFm)

COMCONFm 寄存器用于指定报文缓冲区的类型并设置掩码。

图21-48: CAN报文配置寄存器格式(COMCONFm)(1/2)

地址: 参考表21-16 复位值: 没有定义 R/W

	7	6	5	4	3	2	1	0
COMCONFm	OVS	RTR	MT2	MT1	MT0	0	0	MA0

OVS	覆盖控制位
0	已收到数据帧 ^{注1} 的报文缓冲区不会被新接收的数据帧覆盖。新接收的数据帧将被丢弃。
1	已收到数据帧 ^{注1} 的报文缓冲区会被新接收的数据帧覆盖。

注1: "已收到数据帧的报文缓冲区"是接收报文缓冲区, 其 DN 位已设置为 1

备注: 不管OVS 位和 DN位的设置, 接收和存储远程帧。满足其他条件的远程帧(ID 匹配、RTR=0、TRQ=0)始终接收并存储在相应的报文缓冲区中(生成中断、设置 DN 标志、MDLC [3: 0] 位更新, 并记录到接收历史记录列表中)

RTR	远程帧申请位 ^{注1}
0	发送一个数据帧。
1	发送一个远程帧

注1: RTR 位指定从定义为传输报文缓冲区的报文缓冲区传输的报文帧的类型。即使已收到有效的远程帧, 已收到帧的传输报文缓冲区的 RTR 仍为 0。即使从 CAN 总线接收到匹配ID的远程帧, 传输报文缓冲区的 RTR 位已经设置为 1 从而可以传输远程帧, 远程帧也有可能不会被接收或存储(已生成中断, DN 标志已经设置, MDLC(3: 0)位已更新), 并记录到接收历史记录列表)。

MT2	MT1	MT0	报文缓存类型设置位
0	0	0	发送报文缓存
0	0	1	接收报文缓存(没有屏蔽设置)
0	1	0	接收报文缓存(1个屏蔽设置)
0	1	1	接收报文缓存(2个屏蔽设置)
1	0	0	接收报文缓存(3个屏蔽设置)
1	0	1	接收报文缓存(4个屏蔽设置)
其他值			禁止设置

备注: m = 0到15

图21-48: CAN报文配置寄存器格式(COMCONFm)(2/2)

MA0	报文缓存分配位
0	没有使用报文缓存
1	使用报文缓存

注意: 务必把位1和2写0.

备注: m = 0到15

21.7.23 CAN报文ID寄存器m (C0MIDLm and C0MIDHm)

C0MIDLm 和 C0MIDHm 寄存器用于设置标识符(ID)

图21-49: CAN报文ID寄存器m格式 (C0MIDLm and C0MIDHm)

地址: 参考表21-16	复位值: 没有定义								R/W
	15	14	13	12	11	10	9	8	
C0MIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	
	7	6	5	4	3	2	1	0	
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	
C0MIDHm	15	14	13	12	11	10	9	8	
	IDE	0	0	ID28	ID27	ID26	ID25	ID24	
	7	6	5	4	3	2	1	0	
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16	

IDE	格式模式指定位
0	标准格式模式(ID18到 ID28: 11位) ^{注1}
1	扩展格式模式 (ID0 到 ID28: 29位)

注1: ID0到ID17位没有使用.

ID0到ID28	报文ID
ID18 到ID28	11位标准ID值(当IDE =0)
ID0 到ID28	29位扩展ID值(当IDE =1)

注意:

1. 务必把C0MIDHm寄存器的位13和位14写0.
2. 确保根据给定位位置将ID值对齐到此寄存器中。请注意, 对于标准 ID, 必须将 ID 值移入ID11 到 ID28 位置。

备注: m = 0 到15

21.7.24 CAN报文控制寄存器m (COMCTRLm)

COMCTRLm 寄存器用于控制报文缓冲区的操作。

图21-50: CAN报文控制寄存器m格式 (COMCTRLm)(1/3)

地址: 参考表21-16 复位值: 000xx000B R/W

(a) 读

	15	14	13	12	11	10	9	8
COMCTRLm	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) 写

	15	14	13	12	11	10	9	8
COMCTRLm	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

(a) 读

MUC ^注	报文缓存数据更新位
0	CAN 模块未更新报文缓冲区(接收和存储)
1	CAN 模块正在更新报文缓冲区(接收和存储).

注: 在进行第一个接收和存储之前, MUC 位未定义

MOW	报文缓存覆盖状态位
0	新接收的数据帧不会覆盖报文缓冲区.
1	新接收的数据帧覆盖报文缓冲区

备注: 即使远程帧被接收并存储在传输报文缓存, DN也等于1, MOW位也不会被设置为1。

IE	报文缓存中断申请使能位
0	接收报文缓冲区: 有效报文接收完成中断已禁用。 传输报文缓冲区: 正常报文传输完成中断已禁用。
1	接收报文缓冲区: 有效报文接收完成中断启用。 传输报文缓冲区: 正常报文传输完成中断启用。

备注: m = 0 到15

图21-50: CAN报文控制寄存器m格式(C0MCTRLm)(2/3)

DN	报文缓存数据更新位
0	数据帧或远程帧不存储在报文缓冲区中。
1	数据帧或远程帧存储在报文缓冲区中。

TRQ	报文缓存发送申请位
0	报文缓冲区中没有挂起或正在传输的报文帧传输请求。
1	报文缓冲区保持报文帧的传输挂起或正在传输报文帧。

注意: 请勿同时设置 TRQ 位和 RDY 位 (1)。在设置 TRQ 位之前设置 RDY 位 (1)。

RDY	报文缓存就绪位
0	报文缓冲区可以由软件编写。CAN 模块无法写入报文缓冲区
1	忽略通过软件写入报文缓冲区(对 RDY、TRQ、DN 和 MOW 位的写入访问除外)。CAN 模块可以写入报文缓冲区。

注意:

1. 在报文传输过程中, 不要清除 RDY 位 (0)。按照传输中止过程清除 RDY 位 (0) 以重新定义报文缓冲区。
2. 当RDY位没有被清除时(尽管已经被清除), 再次清除一次。
3. 在写入报文缓冲区寄存器之前, 请确保已清除 RDY位。通过读取 RDY 位来执行此确认。但是, 不需要确认设置 TRQ 位、清除 DN 位、设置 RDY 位或清除 C0MCTRLm 寄存器的 MOW 位。

(b) 写

ClearMOW	设置MOW位
0	MOW位没有改变。
1	MOW位被清0。

SetIE	ClearIE	设置IE位
0	1	IE被清0。
1	0	IE 被置1。
其他值		IE 位不改变。

注意: 始终单独设置 IE 位和 RDY 位。

ClearDN	设置DN位
0	DN不改变。
1	DN被清0。

注意: 不要用软件将 DN 位设置为 1。请务必把位 10写0。

备注: m = 0 到15

图21-50: CAN报文控制寄存器m格式(C0MCTRLm)(3/3)

SetTRQ	ClearTRQ	TRQ设置位
0	1	TRQ 清0.
1	0	TRQ置1.
其他值		TRQ没有变化

注意: 当从其他节点接收报文或传输消报文时, 即使 TRQ 位设置为 1, 也可能无法立即开始传输。即使 TRQ 位已清除到 0, 传输也不会中止。如果有报文正在传输, 传输会继续, 直到传输完成(成功或不成功)。

SetRDY	ClearRDY	RDY设置位
0	1	RDY清0.
1	0	RDY 置1.
其他值		RDY不改变.

注意: 单独设置 IE 位和 RDY 位。

备注: m = 0到15

21.7.25 串行通讯引脚选择寄存器1(PIOR3)

PIOR3 寄存器用于将输入源切换到计时器阵列单元和 CAN 通信引脚。此寄存器可以8 位形式读取或写。
PIOR3 寄存器可用于选择在两个不同的端口上选择 CTxD、CRxD 引脚。

图21-51: 串行通讯引脚选择寄存器1格式(PIOR3)

地址: 0x4004087C 复位值: 00H R/W

	7	6	5	4	3	2	1	0
PIOR3	0	0	0	0	PIOR33	PIOR32	PIOR31	PIOR30

PIOR33	通讯引脚选择FCAN1	
	CTxD0	CRxD0
0	P02	P03
1	P51	P50

21.7.26 端口模式寄存器0, 5 (PM0, PM5)

PM0 和 PM5 寄存器用于将端口 0 和 5 设置为输入或输出。

使用 P02/CTxD0 或 P51/CTxD0 引脚进行串行数据输出时, 将 PM02 或 PM51 位清"0", 并将 P02 或 P51 的输出锁存设置为"1".

使用 P03/CRxD0 或 P50/CRxD0 引脚进行串行数据输入时, 将 PM03 或 PM50 位设置为"1". 此时, P03 或 P50 的输出锁存可能为"0"或"1".

可以使用 8 位内存操作指令设置 PM0、PM5 寄存器。将这些寄存器设置到 FFH 会产生重置信号。

注意: 串行数据输出和输入在每个通道的两个端口, 使用相应的寄存器选择任一端口。

备注: 引脚的定位取决于具体产品, 参考1.3引脚配置和2.1引脚功能列表.

图21-52: 端口模式寄存器0, 5格式(PM0, PM5)

地址: 0x40040320 复位值: FFH R/W

	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

地址: 0x40040325 复位值: FFH R/W

	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PMmn	PMmn pin I/O 引脚模式选择(m = 0, 5 ; n = 0 到7)
0	输出模式 (输出缓存开)
1	输入模式 (输出缓存关)

21.8 CAN控制器初始化

21.8.1 CAN模块初始化

在启用 CAN 模块操作之前，需要通过软件设置 C0GMCS 寄存器的 CCP[3: 0] 位来决定 CAN 模块的系统时钟。CAN 模块工作后，请勿更改 CAN 模块系统时钟的设置。

通过设置 C0GMCTRL 寄存器的 GOM 位启用 CAN 模块。

有关初始化 CAN 模块的过程，请参阅21.16 CAN 控制器的操作。

21.8.2 报文缓存的初始化

启用 CAN 模块后，报文缓冲区包含未定义的值。在将 CAN 模块从初始化模式切换到其中一种操作模式之前，需要为所有报文缓冲区(即使应用程序未使用)进行最小初始化。

- 清除C0MCTRLm 的RDY, TRQ, 和DN位(0)
- 清除C0MCONFm 的MA0 位(0)

备注：m = 0 到15

21.8.3 重新定义报文缓存

重新定义报文缓冲区意味着在接收或传发送报文时，更改报文缓冲区的 ID 和控制信息，而不会影响其他发送/接收操作。

(1) 在初始化模式重新定义报文缓存

将 CAN 模块置于初始化模式一次，然后在初始化模式下更改报文缓冲区的 ID 和控制信息。更改 ID 和控制信息后，将 CAN 模块设置为操作模式。

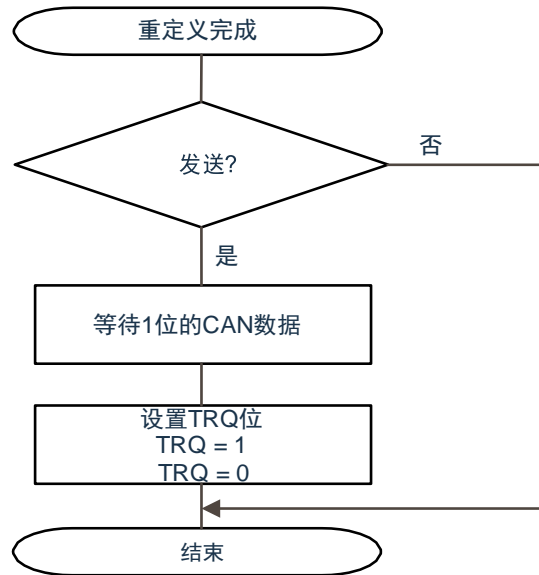
(2) 在接收中重新定义报文缓存

重新定义的操作如图21-66。

(3) 在发送中重新定义报文缓存

要重写已设置传输请求的传输报文缓冲区的内容，请执行传输中止处理(请参阅 21.10.4 (1)正常操作模式下自动块传输(ABT)外的传送中止处理和 21.10.4 (2) ABT传输在正常操作模式下伴有自动块传输外的传输中止处理)。确认传输已中止或完成，然后重新定义报文缓冲区。重新定义传输报文缓冲区后，请使用下面描述的过程设置传输请求。但是，当将传输请求设置为已重新定义的报文缓冲区而不中止正在进行的传输时，不需要 1 位等待时间。

图21-53: 重定义后给发送报文缓存设置发送请求(TRQ)



注意:

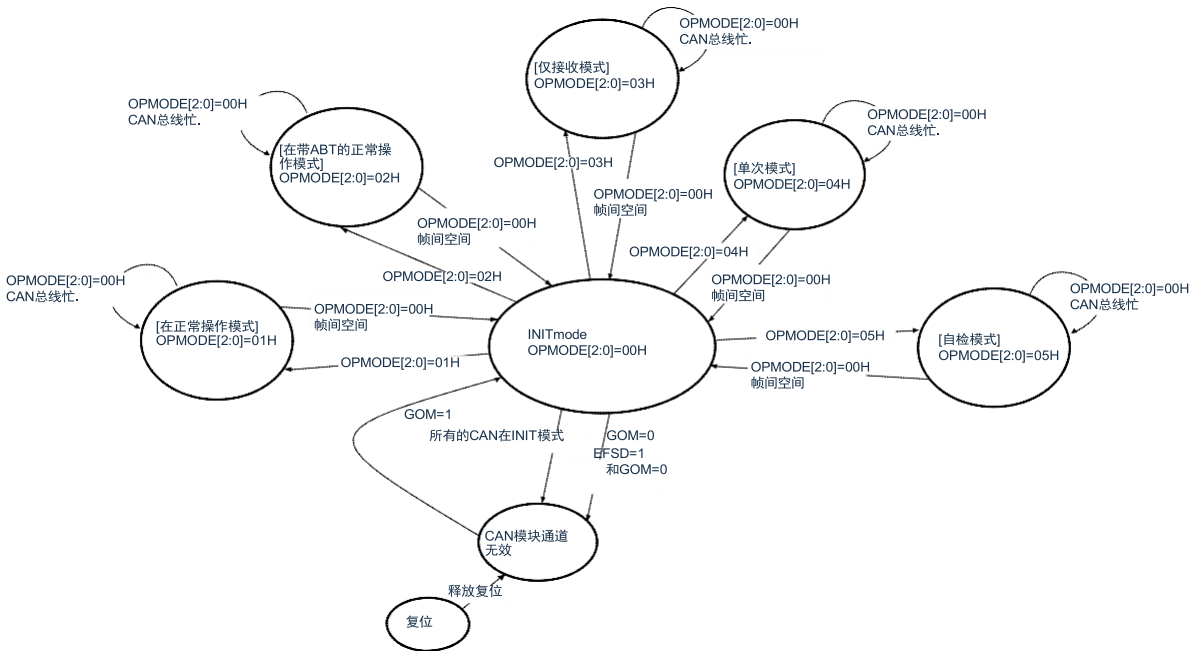
1. 收到报文时，将按照设置为每个接收报文缓冲区的ID和掩码执行接收滤波。如果未遵守图21-66中的过程，则报文缓冲区在重新定义后的内容可能与接收结果(接收筛选的结果)相矛盾。如果发生这种情况，请检查在重新定义后首先接收并存储在报文缓冲区中的ID和IDE是否为重新定义报文缓冲区后存储的ID和IDE。如果重新定义后未存储ID和IDE，请重新定义报文缓冲区。
2. 发送报文时，将根据设置为每个发送报文缓冲区的ID、IDE和RTR位检查传输优先级。选择具有最高优先级的传输报文缓冲区进行传输。如果未观察到图21-53中的过程，则ID没有最高优先级的报文可能在重新定义后传输。

21.8.4 从初始化模式过渡到操作模式

CAN 模块可切换到以下操作模式.

- 正常操作模式
- 正常操作模式伴ABT
- 仅接收模式
- 单次模式
- 自检模式

图21-54: 转换到操作模式



从初始化模式到操作模式的过渡由 C0CTRL 寄存器中的位字符串 OPMODE[2: 0] 控制。

从一种操作模式更改为另一种操作模式需要从两者之间转换到初始化模式。不要直接将一种操作模式更改为另一种操作模式，否则无法保证操作。

当CAN总线不在帧间空间(即帧接收或发送正在进行)且CAN模块在帧间空间(OPMODE[2: 0] 的值更改为 00H)时，申请从操作模式过渡到初始化模式会处于待定。发出将模式更改为初始化模式的请求后，读取 OPMODE [2: 0] 位，直到其值变为000B以确认模块是否已进入初始化模式(参见图21-64)。

21.8.5 重设置CAN模块错误计数器C0ERC

如果需要重置 CAN 模块错误计数器 C0ERC 和 CAN 模块信息寄存器 C0INFO，当从总线关闭状态重新初始化或强制恢复时，请在初始化模式下将 C0CTRL 寄存器的 CCERC 位设置为 1。当此位设置为 1 时，CAN 模块错误计数器 C0ERC 和 CAN 模块信息寄存器 C0INFO 将清除到其默认值。

21.9 报文接收

21.9.1 报文接收

在所有操作模式下，分析完整的报文缓冲区区域，以找到存储新接收报文的适当缓冲区。满足以下条件的所有报文缓冲区都包含在该评估(RX 搜索过程)中：

- 被使用了的报文缓存
(COMCONFm 的MA0位是1)
- 被设置为接收报文的缓存
(COMCONFm的MT[2:0]位设置为001B,010B,011B,100B,or101B.)
- 准备好接收
(COMCTRLm的RDY位设置为1)

当有两个或更多的报文缓存来接收一个报文，这个报文会储存在优先级更高的缓存。例如，当未屏蔽的接收报文缓冲区和链接到掩码 1 的接收报文缓冲区具有相同的 ID 时，接收的报文不会存储在链接到掩码 1 的报文缓冲区中，即使该报文缓冲区尚未收到报文，并且报文已经在未屏蔽的接收报文缓冲区中。换句话说，当设置为将报文存储在具有不同优先级的两个或多个报文缓冲区中时，具有最高优先级的报文缓冲区始终存储报文；报文不存储在优先级较低的报文缓冲区中。当具有最高优先级的报文缓冲区无法接收和存储报文(即 DN=1 指示已接收报文时，但由于 OWS = 0，重写被禁用)，这也适用于此情况。在这种情况下，报文实际上不会被接收存储在具有最高优先级的候选报文缓冲区中，但也不会存储在优先级较低的报文缓冲区中。

优先级	相同ID被设置情况下的储存	
1(高)	没有屏蔽的报文缓存	DN =0
		DN = 1 和OWS =1
2	链接到屏蔽1的报文缓存	DN =0
		DN = 1 和OWS =1
3	链接到屏蔽2的报文缓存	DN =0
		DN = 1 和OWS =1
4	链接到屏蔽3的报文缓存	DN =0
		DN = 1 和OWS =1
5(低)	链接到屏蔽4的报文缓存	DN =0
		DN = 1 和OWS =1

备注：m = 0 到15

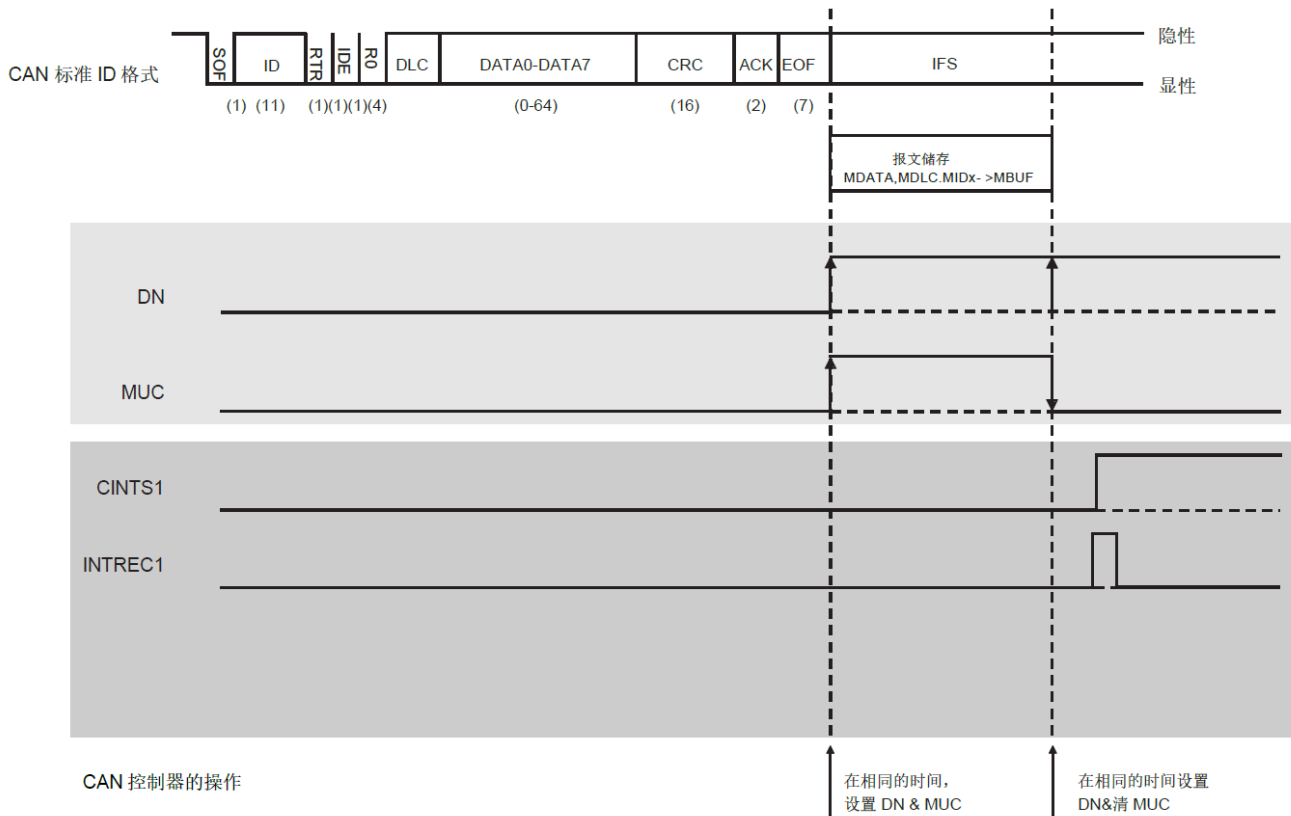
21.9.2 接收数据读

要在读取 CAN 报文缓冲区时保持数据一致性，请根据图21-76到图21-78执行数据读取。

在报文接收期间，CAN 模块将 COMCTRLm 寄存器的 DN 设置两次：在将数据存储到报文缓冲区的开始，再次是在此存储过程结束时。在此存储过程中，报文缓冲区 COMCTRLm 寄存器的 MUC 位会被设置。(参见图 21-55)。

接收历史列表也会在此存储过程之前更新。此外，在存储过程 (MUC_1) 期间，报文缓冲区的 COMCTRLm 寄存器的 RDY 位被锁定，以避免通过 CPU 进行巧合的数据 WR。请注意，当 CPU 访问报文缓冲区时，存储进程可能会受到干扰(延迟)。

图21-55： DN 和MUC位设置周期(对标准ID格式)



备注：m = 0 到15

21.9.3 接收历史列表功能

接收历史记录列表中的接收历史记录列表 (RHL) 函数记录接收和存储每个数据帧或远程帧的接收报文缓冲区的数量。RHL 由相当于最多 23 条报文的存储元素，最后一个报文入指针(LIPT)具有相应 C0LIPT 寄存器和接收历史记录列表获取指针(RGPT)具有相应 C0RGPT 寄存器。

在 CAN 模块从初始化模式过渡到其中一种操作模式后，RHL 立即处于未定义状态。

C0LIPT 寄存器包含由 LIPT 指针减去 1 的值指示的 RHL 元素的内容。因此，通过读取 C0LIPT 寄存器，可以检查首先接收和存储数据帧或远程帧的报文缓冲区数。LIPT 指针用作写入指针，指示 RHLa 报文缓冲区数的哪一部分被记录。每当接收和存储数据帧或远程帧时，相应的报文缓冲区编号将记录到 LIPT 指针指示的 RHL 元素。每次录制到 RHL 时，LIPT 指针将自动递增。这样，接收和存储帧的报文缓冲区的数量将按时间顺序记录。

RGPT 指针用作读取指针，用于从RHL读取记录的报文缓冲区编号。此指针指示CPU尚未读取的第一个 RHL元素。通过按软件读取C0RGPT 寄存器，可以读取已接收并存储数据帧或远程帧的报文缓冲区数。每次从 C0RGPT 寄存器读取报文缓冲区编号时，RGPT 指针将自动递增。

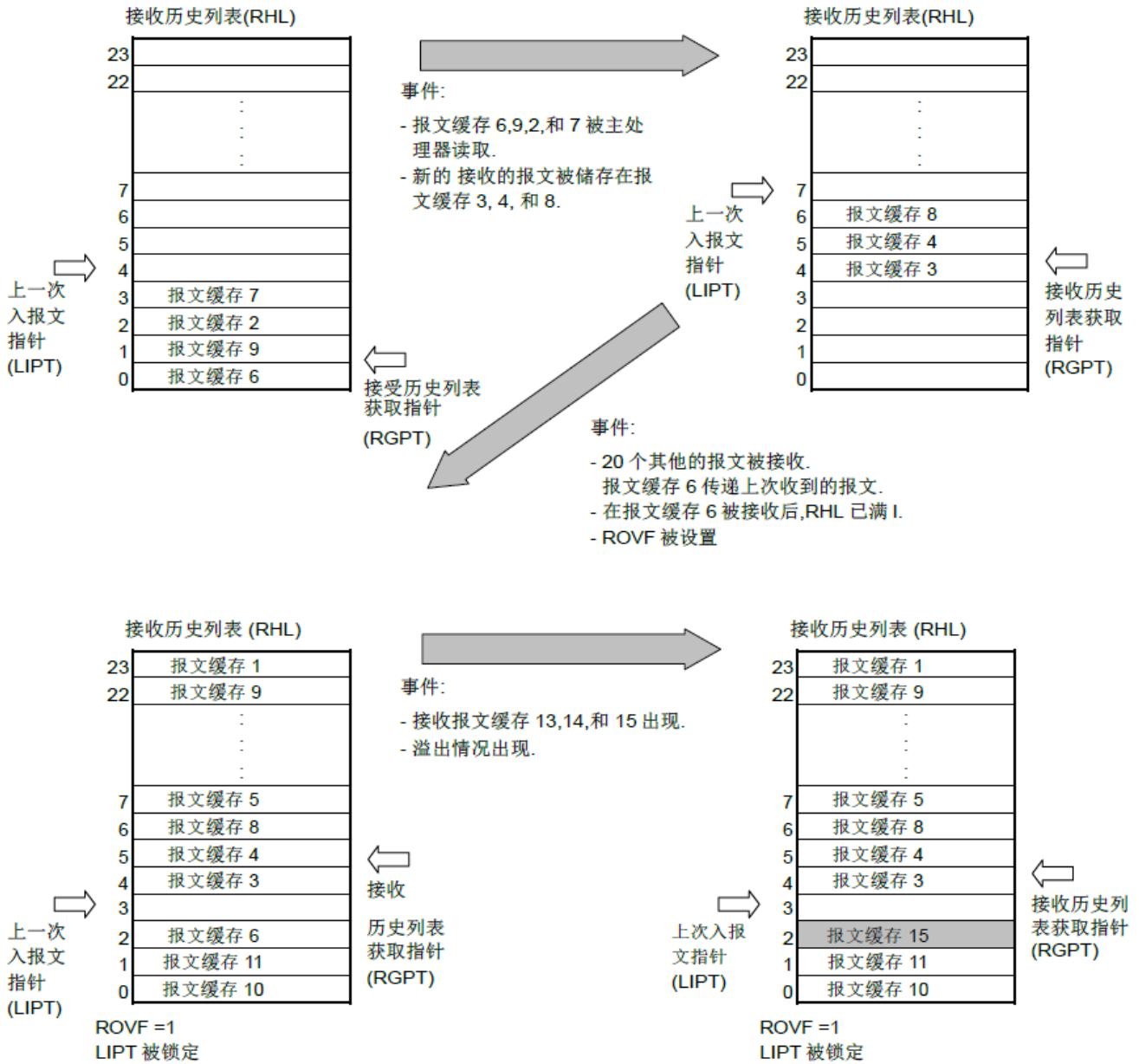
如果 RGPT 指针的值与 LIPT 指针的值匹配，则 C0RGPT 寄存器的 RHPM 位(接收历史记录列表指针匹配)设置为1。这表明RHL中未读取的报文缓冲区号未保留。如果记录新的报文缓冲区编号，LIPT 指针将递增，并且由于其值不再与 RGPT 指针的值匹配，因此将清除 RHPM 位。换句话说，未读报文缓冲区的数量存在于 RHL 中。

如果LIPT指针递增并匹配RGPT指针减去1的值，则C0RGPT 寄存器的ROVF位(接收历史记录列表溢出)设置为1。这表明RHL中充满了尚未读取的报文缓冲区数。当发生进一步的报文接收和存储时，最后一个记录的报文缓冲区号将被接收和存储新报文的报文缓冲区数覆盖。在这种情况下，在设置ROVF位(1)后，RHL中记录的报文缓冲区编号并不完全按时间顺序排列。但是，报文本身不会丢失，并且可以通过CPU搜索在 DN 位的帮助下在报文缓冲区内存中定位到。

注意：如果历史记录列表处于溢出条件(已设置ROVF)，则读取历史记录列表内容仍是可能的，直到历史记录列表为空(由RHPM标志指示)。然而，历史记录列表仍处于溢出状态，直到通过软件清除ROVF。如果未清除 ROVF，则 RHPM 标志也不会在新接收帧的报文上更新(清除)。这可能导致这种情况，即 RHPM 表示空历史记录列表，尽管已发生接收，而历史记录列表处于溢出状态(已设置 ROVF 和 RHPM)。

只要RHL包含23个或更少的条目，则保持发生顺序。如果主机处理器未读取RHL时，发生更多接收，则无法恢复完整的接收序列。

图21-56: 接收历史列表



备注: ROVF=1表示LIPT等于RGPT-1, 而报文缓冲区号存储到LIPT-1指示的元素。

21.9.4 屏蔽功能

对于任何用于接收的报文缓冲区，可以选择分配给四分之一的接收掩码(或无掩码)。

通过使用掩码功能，可以通过屏蔽位来减少报文ID比较，从而允许将多个不同的ID接收到一个缓冲区中。

当掩码函数生效时，接收报文中掩码定义为"1"的标识符位不会与报文缓冲区中的相应标识符位进行比较。

但是，这种比较会在被掩码定义为0的位中执行。

例如，所有具有标准格式ID的报文(其中ID25 到ID27 位为"0"，ID22 和ID24 位为"1")将存储在报文缓冲区14中。此示例的过程如下所示。

<1> 识别符被储存在报文缓存中

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

备注：x = 不用管

<2> 识别符被配置到报文缓存14中(例子)

(使用CAN 0报文ID寄存器L14和H14(C0MIDL14和C0MIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

ID25到ID27的ID已清除为"0"，ID24和ID22设置为"1"，已注册(初始化)到报文缓冲区14。

备注：报文缓冲区14设置为链接到掩码1的标准格式标识符(C0MCONF14寄存器的MT[2: 0]设置为010B)。

<3> CAN模块1掩码设置(mask1)(例子)

(使用CAN0模块掩码1寄存器L和H(C0MASK1L和C0MASK1H))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1
CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1
CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0				
1	1	1	1	1	1	1				

1: 没有比较 (屏蔽) 0: 比较

CMID24 到 CMID27和 CMID22 位被清除为"0"，CMID0 到 CMID21、CMID23 和CMID28位设置为"1"。

21.9.5 多缓冲区接收块功能

多缓冲区接收块 (MBRB) 功能用于将数据块按顺序存储在两个或多个报文缓冲区中, 而没有 CPU 交互, 通过将同一ID设置到具有相同报文缓冲区类型的两个或多个报文缓冲区。

例如, 假设相同的报文缓冲区类型设置为5个报文缓冲区(报文缓冲区10到14), 并且给他们设置相同的ID。如果接收到的第一条报文ID与缓存区ID相同, 则该报文存储在报文缓冲区10中。此时, 将设置报文缓冲区10的DN位, 禁止在收到后续报文时覆盖该报文缓冲区。

如果收到具有匹配ID的下一条报文, 则会接收该报文并将其存储在报文缓冲区11中。每次收到具有匹配ID的报文时, 它将按顺序(按升序顺序)存储在报文缓冲区12, 13和14中。即使收到由多条报文组成的数据块, 也会进行扫描后存储和接收报文, 而不会覆盖以前收到的匹配ID数据。

可以通过设置每个报文缓冲区的COMCTRLm寄存器的IE位来检查是否接收和存储数据块, 例如, 如果数据块由k报文组成, 则k报文缓冲区将初始化来接收数据块。报文缓冲区0到(k-2)中的IE位被清除为0(禁用中断), 报文缓冲区k-1中的IE位设置为1(启用中断)。在这种情况下, 当收到报文并存储在报文缓冲区k-1中, 指示MBRB已满时, 将发生接收完成中断。或者, 通过清除报文缓冲区0到(k-3)的IE位并设置报文缓冲区k-2的IE位, 可以发出MBRB即将溢出的警告。

将接收数据存储在MBRB的每个报文缓冲区中的基本条件与将数据存储在单个报文缓冲区中的条件相同。

注意:

1. 可以为每种相同类型的报文缓冲区配置MBRB。因此, 即使ID匹配但报文缓冲区类型不同的另一个MBRB的报文缓冲区存在空缺, 接收的报文也不会存储在该报文缓冲区中, 而是丢弃。
2. MBRB没有环形缓存结构。因此, 在MBRB配置中具有最高数字的报文缓冲区中存储报文后, 新接收的报文将不会存储在报文缓冲区中报文缓存数字最低的报文缓冲区中。
3. MBRB根据接收和存储条件来运行; 没有专用于MBRB的设置, 例如函数启用位。通过将相同的报文缓冲区类型和ID设置到两个或多个报文缓冲区, MBRB将自动配置。
4. 使用MBRB, "匹配ID"表示"掩码后匹配ID"。即使设置为每个报文缓冲区的ID不同, 如果掩码寄存器屏蔽的ID匹配, 它也被视为匹配的ID, 并且具有此ID的缓冲区被视为报文的存储目标。
5. 关于MBRBs的优先级请参考21.9.1 报文接收。

备注: m = 0 到15

21.9.6 远程帧接收

在所有操作模式下，当收到远程帧时，将从满足以下条件的所有报文缓冲区中搜索能存储远程帧的报文缓冲区。

- 用于报文缓存
(COMCONFm寄存器的MA0位被置1)
- 被设置为传输报文缓存
(COMCONFm寄存器的MT [2:0]位被设置为000B)
- 准备接收
(COMCTRLm寄存器的RDY位设置为1.)
- 设置为传输报文
(COMCONFm寄存器的RTR位清0)
- 未设置传输请求。
(COMCTRLm寄存器的TRQ位被清0)

接受远程帧后，如果接收的远程帧的ID与满足上述条件的报文缓冲区的ID匹配，则执行以下操作：

- COMDLCm寄存器的MDLC [3:0]位储存接收的DLC值
- 数据区间的COMDATA0m到COMDATA7m不被更新(保存接收前的数据)。
- COMCTRLm寄存器的DN位被置1。
- C0INTS寄存器的CINTS1位被置1 (如果接收和存储的报文缓冲区寄存器COMCTRLm中的IE位设置为1)。
- 输出接收完成中断(INTCREC)(如果接收和存储帧的报文缓冲区寄存器COMCTRLm中的IE位设置为1，并且C0IE寄存器的CIE1位设置为1)。
- 报文缓冲区编号被记录到接收历史记录列表中。

注意：当搜索到报文缓冲区用来接收和存储远程帧时，报文缓冲区COMCONFm寄存器的OWS位和COMCTRLm寄存器的DN位的覆盖控制不会受到影响。将忽略OWS的设置，并且在任何情况下都会设置DN。如果多个传输报文缓冲区具有相同的ID，并且接收的远程帧的ID与该ID匹配，则远程帧存储在报文缓存区数最低的缓冲区中。

备注：m = 0 到15

21.10 报文发送

21.10.1 报文发送

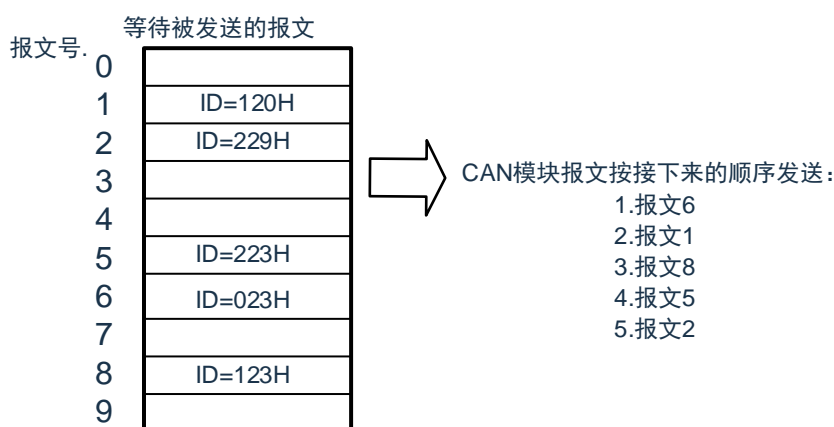
在所有操作模式下，如果TRQ位在满足以下条件的报文缓冲区中设置为1，则开始搜索要传输报文的报文缓冲区。

- 被用于报文缓存
(COMCONFm寄存器的MA0位置1)
- 设置为发送报文缓存
(COMCONFm寄存器的MT [2:0]位设置为000B.)
- 准备发送
(COMCTRLm寄存器的RDY位置1.)

CAN系统是一个多主通信系统。在这样的系统中，报文传输的优先级根据报文标识符(ID)确定，为了在有多多个报文等待传输时能实现软件传输处理，CAN 模块使用硬件来检测具有最高优先级的报文ID，并自动标识该报文，这消除了对软件优先级控制的需要。

发送优先级由标识符(ID)控制。

图21-57：报文处理例子



发送报文搜索后，发送具有挂起发送请求(TRQ位设置为1的报文缓冲区)的发送报文缓冲区的最高优先级的发送报文。

如果设置了新的发送请求，则将具有新发送请求的发送报文缓冲区与具有挂起发送请求的发送报文缓冲区进行比较。如果新的发送请求具有更高的优先级，则会传输该请求，除非已开始传输优先级较低的报文。但是，如果已开始传输优先级较低的报文，则稍后将发送新的发送请求。为了解决这种优先级反转效应，软件可以对低优先级报文执行传输中止请求。最高优先级根据以下规则确定。

优先级	条件	描述
1(高)	ID前11个位的值 [ID18 到 ID28]:	首先发送ID前11 位表示的最低值的报文帧。如果11位标准 ID 的值等于或小于29位扩展ID的前11 位, 则11 位标准ID的优先级高于具有29位扩展ID的报文帧。
2	帧类型	具有11位标准ID(RTR 位已清除为0)的数据帧的优先级高于具有标准ID的远程帧和具有扩展ID的报文帧。
3	ID类型	具有标准ID(ID 位已清除为 0)报文帧的优先级高于具有扩展ID的报文帧
4	ID低18个位的值[ID0 到 ID17]:	如果多个发送挂起的扩展ID报文帧在ID的前11 位中具有相同的值和相同的帧类型(等于 RTR 位值), 则首先传输其扩展ID低18 位中值最低的报文帧。
5(低)	报文缓存号	如果两个或多个报文缓冲区请求发送具有相同ID的报文帧, 则首先传输来自报文缓冲区中报文缓冲区编号最低的报文。

备注:

- 如果在带ABT的正常操作模式下自动块传输请求位ABTTRG设置为1, 则仅ABT报文缓存组中的一个报文缓存的TRQ位设置为1。如果ABT模式由ABTTRG位触发, 则ABT区域中的一个TRQ位设置为1(缓冲区0到7)。除了此TRQ位之外, 应用程序还可以请求其他不属于ABT区域的TX报文缓冲区传输(将TRQ设置为1)。在这种情况下, 间隔仲裁进程 (TX-search)评估TRQ位设置为1的所有TX报文缓冲区, 并选择具有最高优先级标识符的报文缓冲区为下一次传输。如果有2个或多个标识符具有最高优先级(即相同的标识符), 则首先传输位于最低报文缓冲区号的报文。
 成功传输报文帧后, 将执行以下操作:
 - 相应发送报文缓冲区的TRQ标志将自动清除到0。
 - C0INTS 寄存器的发送完成状态位CINTS0设置为1(如果相应发送报文缓冲区的中断启用位(IE)设置为1)。
 - 中断请求信号INTC0TRX输出(如果C0IE寄存器的CIE0位设置为1, 并且相应的发送报文缓冲区的中断启用位(IE)设置为1)。
- 更改发送缓冲区的内容时, 在更新缓冲区内容之前, 必须清除此缓冲区的RDY标志。与内部传输操作时一样, RDY 标志可能暂时锁定, 更改后必须由软件检查RDY的状态。
- m = 0 到15

21.10.2 发送历史列表功能

传输历史记录列表(THL)功能记录在传输历史记录列表中发送数据或远程帧的报文缓冲区的编号。THL包含相当于最多七条报文的存储元素、具有相应C0LOPT寄存器的最后一个出报文指针(LOPT)以及C0TGPT寄存器的发送历史记录列表获取指针(TGPT)。

在CAN模块从初始化模式过渡到其中一种操作模式后，THL 将立即处于未定义。

C0LOPT 寄存器包含由 LOPT 指针减去 1 的值指示的 THL 元素的内容。因此，通过读取C0LOPT寄存器，可以检查首先传输数据帧或远程帧的报文缓冲区数。LOPT指针用作写入指针，指示记录报文缓冲区编号的THL的哪一部分被记录。每当传输数据帧或远程帧时，相应的报文缓冲区编号将记录到LOPT指针指示的THL元素。每次记录THL时，LOPT 指针将自动递增。这样，被接收并存储的帧的报文缓冲区编号将按时间顺序记录。

TGPT指针用作读取指针，用于从THL读取记录的报文缓冲区编号。此指针指示CPU尚未读取的第一个THL元素。通过软件读取C0TGPT寄存器，可以读取已完成传输的报文缓冲区数。每次从 C0TGPT 寄存器读取报文缓冲区编号时，TGPT指针将自动递增。

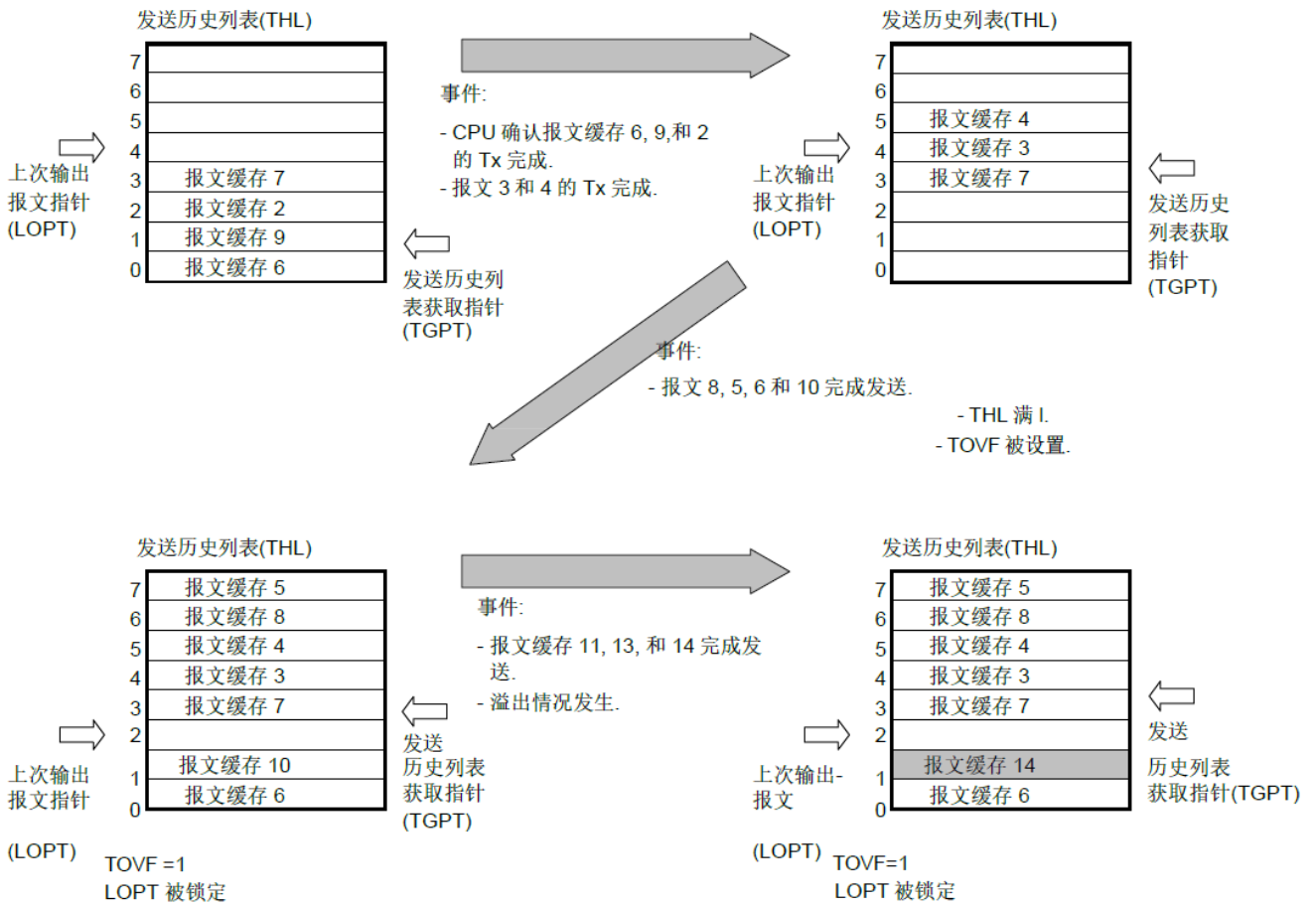
如果TGPT指针的值与LOPT指针的值匹配，则C0TGPT寄存器的THPM位(传输历史记录列表指针匹配)设置为1。这表明THL中没有保留未读取的报文缓冲区编号。如果记录新的报文缓冲区编号，则LOPT指针将递增，并且由于其值不再与TGPT指针的值匹配，因此将清除THPM位。换句话说，未读报文缓冲区的数量存在于THL中。

如果 LOPT 指针递增并匹配 TGPT 指针减去 1 的值，则 C0TGPT 寄存器的 TOVF 位(传输历史记录列表溢出)设置为 1。这表示THL中充满了尚未读取的报文缓冲区编号。如果收到并存储了新报文，则上次记录的报文缓冲区编号将被随后传输来的报文的报文缓冲区数覆盖。在TOVF位设置1后，THL中记录的报文缓冲区编号不能完全反映按时间顺序排列。但是，其他传输报文可以通过应用于所有传输报文缓冲区的CPU搜索找到，除非CPU尚未覆盖之前这些缓冲区之一中的传输对象。总共最多发生六次传输完成扫描，而不会THL溢出。

注意：如果历史记录列表处于溢出条件(TOVF被设置)，仍可以读取历史记录列表内容，直到历史记录列表为空(由 THPM标志指示)。然而，历史列表仍然处于溢出状态，直到TOVF被软件清除。如果未清除TOVF，则在成功传输新报文后也不会更新(清除)THPM标志。这可能会导致这种情况，即THPM指示空的历史记录列表，尽管已成功传输，而历史记录列表处于溢出状态(TOVF和THPM被设置)。

备注： m = 0 到15

图21-58: 发送历史列表



TOVF=1表示LOPT等于TGPT-1，而报文缓冲区号存储到LOPT-1指示的元素。

21.10.3 自动块传输 (ABT)

自动块传输(ABT)功能用于在没有CPU交互时成功的连续传输两个或多个数据帧。分配给ABT功能的传输报文缓冲区的最大数量为8(报文缓冲区编号0到7)。

通过将C0CTRL寄存器的OPMODE[2: 0]位设置为010B, "具有自动块传输功能的正常运行模式"(此处称为ABT模式)可以被选择。

要发出ABT传输请求, 请首先通过软件定义报文缓冲区。在用于ABT的所有报文缓冲区中设置MA0 位(1), 并通过将MT[2: 0]位设置为000B将所有缓冲区定义为传输报文缓冲区。请确保ABT每个报文缓冲区设置ID, 即使对所有报文缓冲区使用相同的ID也是如此。要使用两个或多个ID, 请使用C0MIDLm和C0MIDHm寄存器设置每个报文缓冲区的ID。在为ABT功能发出传输请求之前, 将C0MDLm和C0MDATA0m设置为C0MDATA7m。

完成ABT的报文缓冲区初始化后, 需要设置RDY 位为1。在ABT模式下, TRQ 位不由软件操作。

准备好ABT报文缓冲区的数据后, 将ABTTRG位设置为1。然后启动自动块传输。启动ABT时, 第一个报文缓冲区(报文缓冲区0)中的TRQ位将自动设置为1。报文缓冲区0的数据传输完成后, 自动设置下一个报文缓冲区的TRQ位, 报文缓冲区1。这样, 传输将依次执行。

在执行连续传输时自动设置传输请求(TRQ)的时间间隔内, 程序可以插入延迟时间。要插入的延迟时间由C0GMABTD 寄存器定义。延迟时间的单位是DBT(数据位时间)。DBT 取决于C0BRP和C0BTR 寄存器的设置。

在ABT区域内的传输对象中, 不评估传输ID的优先级。报文缓冲区0到7的数据按顺序传输。当从报文缓冲区7传输数据帧完成后, ABTTRG 位将自动清除为0, 并且ABT操作完成。

如果在ABT期间清除了ABT报文缓冲区的RDY位, 则不会从该缓冲区传输任何数据帧, ABT停止, 并清除ABTTRG位。之后, 可以通过软件将RDY和ABTTRG位设置为1, 从ABT停止的报文缓冲区恢复传输。为了不从停止ABT的报文缓冲区恢复传输, 可以在停止ABT模式并将ABTRG位清除为0时将ABTCLR位设置为1来重置内部ABT引擎。在这种情况下, 如果ABTCLR位被清除为0, 然后将ABTTRG位设置为1, 则从报文缓冲区0启动传输。

中断可用于检查数据帧是否从ABT的所有报文缓冲区传输。为此, 需要清除除最后一个报文缓冲区之外的每个报文缓冲区的C0MCTRLm 寄存器的IE 位为0。

如果将ABT功能(报文缓冲区8到15)以外的传输报文缓冲区分配给传输报文缓冲区, 则接下来要传输的报文由当前处于挂起状态的传输ABT报文缓冲区的传输ID的优先级确定, 并且是ABT功能所使用的缓冲区以外报文缓冲区的传输ID。

从 ABT 报文缓冲区传输的数据帧不记录在传输历史记录列表(THL)中。

注意:

1. 将ABTCLR位设置为1, 同时将ABTTRG位清除为0, 以便在缓冲区数0处恢复ABT操作。如果将ABTCLR位设置为1, 而ABTTRG位设置为1, 则不能保证后续操作。
2. 如果通过将ABTCLR位设置为1来清除自动块传输引擎, 则在清除请求的处理完成后, ABTCLR位将自动清除。
3. 请勿在初始化模式下设置ABTTRG位。如果在初始化模式下设置ABTTRG位, 则在模式从初始化模式更改为ABT模式后, 不能保证正确的操作。
4. 请勿在ABT的正常运行模式下由软件将ABT报文缓冲区的TRQ位设置为1。否则, 无法保证操作。
5. COGMABTD寄存器用于设置从前一条ABT报文完成到下一个ABT报文设置TRQ位的期间插入延迟时间, 当传输请求按ABT报文号顺序设置时, 在ABT模式下连续传输。报文实际传输到CAN总线的时间, 因从其他站点传输的状态以及ABT报文以外的报文发送请求的设置状态而异(报文缓冲区8到15)。
6. 如果对ABT报文以外的报文发出传输请求, 并且未在自动设置ABT传输请求的时间间隔(COGMABTD = 00H)中插入延迟时间, 则ABT报文以外的报文可能不依赖于ABT报文的优先级。
7. 当ABTTRG = 1 时, 不要清除RDY位到 0。
8. 如果在具有ABT的正常操作模式处于活动状态时, 从另一个节点收到报文, 则ABT区域的TX报文可能以一帧的延迟传输, 尽管COGMABTD 寄存器设置成了00H。

备注: m = 0 到15

21.10.4 传输中止处理

(1) 正常操作模式下自动块传输(ABT)外的传送中止处理

用户可以将COMCTRLm寄存器的TRQ位清0来中止传输请求，如果中止成功，将立即清除TRQ位。可以使用指示CAN总线传输状态的C0CTRL寄存器的TSTAT位和C0TGPT寄存器检查传输是否成功中止(有关详细信息，请参阅图21-72中的处理)。

(2) ABT传输在正常操作模式下伴有自动块传输外的传输中止处理

用户可以将C0GMABT寄存器的ABTTRG位清除到0以中止传输请求。检查C0GMABT寄存器的ABTTRG位 = 0后，将COMCTRLm寄存器的TRQ位清除为0。如果中止成功，将立即清除TRQ位。可以使用指示CAN总线上传输状态的C0CTRL寄存器的TSTAT位和C0TGPT寄存器检查传输是否成功中止(有关详细信息，请参阅图21-74中的处理)。

(3) ABT传输在正常操作模式下伴有自动块传输的传输中止处理

要中止已启动的ABT，请清除C0GMABT寄存器的ABTTRG位为0。在这种情况下，如果ABT报文当前正在传输，并且直到传输完成(成功与否)，则ABTTRG位将保持为1，并在传输完成后立即清除为0。这将中止ABT。

如果最后一次传输(在ABT之前)成功，则ABT的正常运行模式将保留，内部ABT指针指向要传输的下一个报文缓冲区。

如果传输错误，内部ABT指针的位置取决于TRQ位在最后一个传输的报文缓冲区中的状态。如果请求清除ABTTRG位时TRQ位设置为1，则内部ABT指针指向最后一个传输的报文缓冲区(有关详细信息，请参阅图21-73中的进程)。如果在清除ABTTRG位时将TRQ位清除为0，则内部ABT指针将递增(+1)，并指向ABT区域中的下一个报文缓冲区(有关详细信息，请参阅图21-74中的进程)。

注意：请确保通过将ABTRG清除为0来中止ABT，如果通过清除RDY位请求中止传输可能无法实现。

在ABT中止后恢复ABT的正常操作模式，并将ABTTRG位设置为1时，可以从下表确定要传输的下一个ABT报文缓冲区。

ABT报文缓存的状态 TRQ	成功传输后的中止	错误传输后中止
设置(1)	ABT区间的下一个报文缓存 ^{注1}	ABT区间相同的报文缓存
清除(0)	ABT区间的下一个报文缓存 ^{注1}	ABT区间的下一个报文缓存 ^{注1}

注1：仅当ABT区域中存在准备使用ABT的报文缓冲区时，才能执行上述恢复操作。例如，在报文缓冲区7的ABT正在处理的时候发出的中止请求，被视为ABT已完成，而不是中止，如果报文缓冲区7的传输已成功完成，ABT TRG已清除为0。如果ABT区域中下一个报文缓冲区中的RDY位被清除为0，则保留内部ABT指针，但即使ABT TRG设置为1，并且ABT立即结束，也不会执行恢复操作。

备注：m = 0 到15

21.10.5 远程帧传输

远程帧只能从传输报文缓冲区传输。通过COMCONFm寄存器的RTR位设置是数据帧或远程帧传输。设置RTR位为(1)设置远程帧传输。

备注：m = 0 到15

21.11 省电模式

21.11.1 CAN睡眠模式

CAN 睡眠模式可用于将CAN控制器设置为待机模式，以降低功耗。CAN 模块可以从所有操作模式进入CAN睡眠模式。CAN睡眠模式的释放将CAN模块返回到进入CAN睡眠模式前的操作模式。

在CAN睡眠模式下，CAN模块不传输报文，即使传输请求发出或挂起。

(1) 进入CAN睡眠模式

CPU通过将01B写入C0CTRL寄存器的PSMODE[1: 0] 位来发出CAN睡眠模式转换请求。

此转换请求仅在以下条件下应答。

- CAN模块已经在以下的操作模式
- 正常操作模式
- 正常操作模式伴ABT
- 仅接受模式
- 单次模式
- 自检模式
- 在上述所有操作模式下的停止模式
- CAN总线处于空闲状态 (帧间空间中的第 4 位是隐性的)^注
- 没有被挂起的传输申请

注：如果 CAN 总线固定为显性，则过渡到CAN睡眠模式的请求将处于挂起状态。

此外，从CAN停止模式到CAN睡眠模式的过渡也独立于CAN总线状态。

备注：如果睡眠模式请求处于挂起状态，同时在邮箱中收到一条报文，则睡眠模式请求不会取消，而是在邮件存储完成后立即执行。如果CPU将执行RX中断，这可能会导致睡眠模式下的FCAN。因此，如果在睡眠模式，中断必须使用MBON标志检查对报文缓冲区以及接收历史记录列表寄存器的访问。

如果未满足上述任何条件，CAN 模块将运行如下：

- 如果请求从初始化模式进入CAN睡眠模式，则忽略CAN睡眠模式转换请求，CAN 模块仍然保持在初始化模式。
- 如果CAN总线状态不是总线空闲(即CAN总线状态是传输或接收)，则在其中一种操作模式下请求CAN睡眠模式时，无法立即转换为CAN睡眠模式。在这种情况下，CAN睡眠模式转换请求保持挂起状态，直到CAN总线状态变为总线空闲(帧间空间的第4位是隐性位)。在从CAN睡眠模式请求到成功转换的这段时间内，PSMODE[1: 0] 位保持00B。当模块进入CAN睡眠模式时，PSMODE [1: 0] 位设置为01B。
- 如果在CAN模块处于其中一个操作模式时，同时发出向初始化模式的转换请求和向CAN睡眠模式转换的请求，则启用初始化模式转换的请求。CAN 模块在预定的时间进入初始化模式。此时，CAN睡眠模式请求不保持挂起状态，而是将被忽略。
- 即使未同时请求初始化模式和睡眠模式(即第一个请求未在第二次请求发出时授予)，初始化请求对睡眠模式请求具有优先级。请求初始化模式时，睡眠模式请求将被取消。当初始模式的挂起申请出现时，睡眠模式接下来的申请在发布时被删除。

(2) CAN睡眠模式的状态

CAN 模块在进入CAN睡眠模式后处于以下状态.

- 内部操作时钟已停止, 电源功耗最小.
- 侦测下降沿功能的CAN接收引脚(CRxD)保持有效来从CAN总线唤醒CAN模块.
- 要从 CPU 唤醒 CAN 模块, 可以将数据写入 CAN 模块控制寄存器 (C0CTRL) 的 PSMODE[1:0], 但无法写入其他 CAN 模块寄存器或位.
- 除了C0LIPT, C0RGPT, C0LOPT和C0TGPT, CAN模块的其他寄存器可以读.
- CAN报文缓存寄存器不能被写或读
- CAN0全局控制寄存器(C0GMCTRL)的MBON位被清除.
- 申请转换到初始化模式的申请没有应答, 而且被忽略.

(3) CAN睡眠模式释放

CAN 睡眠模式由以下事件释放:

- 当CPU写00B到C0CTRL寄存器的PSMODE [1:0] 位
- CAN 接收引脚(CRxD) 侦测到下降沿(即CAN总线电平从隐性到显性)

注意:

1. 即使下降沿出现在接收报文的SOF, 也不会被接收和存储。如果CPU在CAN睡眠模式的时候关闭了CAN 时钟。即使这样, CAN 睡眠模式也不会释放, PSMODE[1: 0] 将继续为01B, 除非再次提供CAN的时钟。除此之外, 之后将不会接收接收报文。
2. 如果在提供CAN时钟的状态下检测到CAN接收引脚(CRxD)上的下降沿, 则需要通过软件清除PSMODE0位(有关详细信息, 请参阅图21-81 中的处理)。

释放睡眠模式后, CAN模块将返回到请求CAN睡眠模式时的操作模式, C0CTRL 寄存器的PSMODE [1: 0] 位将重置为 00B。如果 CAN 睡眠模式通过CAN总线状态的更改释放, 则C0INTS寄存器的CINTS5位设置为1, 而不管C0IE寄存器的CIE位如何。CAN模块从CAN睡眠模式释放后, 通过自动检测CAN总线上的11个连续隐性电平位, 它再次进入CAN总线。用户应用程序必须等到MBON=1后, 再访问报文缓冲区。当CAN模块处于 CAN 睡眠模式时, 发出向初始化模式的转换请求, 该请求将被忽略。在进入初始化模式之前, CPU通过软件释放睡眠模式。

注意: 请注意, 如果CAN睡眠模式被CAN总线事件释放; 因此, 如果发生CAN总线事件, 即使在请求睡眠模式之后, 也可能发生唤醒中断。

备注: m = 0 到15

21.11.2 CAN停止模式

CAN停止模式可用于将CAN控制器设置为待机模式，以降低功耗。CAN模块只能从CAN睡眠模式进入CAN停止模式。CAN 停止模式的释放需要CAN模块处于CAN睡眠模式。

CAN停止模式只能通过将01B写入C0CTRL寄存器的PSMODE[1: 0]位，而不是通过CAN总线状态的更改来释放(进入CAN睡眠模式)。即使发出或挂起传输请求，也不会传输任何报文。

(1) 进入CAN停止模式

通过将11B写入C0CTRL寄存器的PSMODE[1: 0]位，发出CAN停止模式转换请求。仅当CAN模块处于CAN睡眠模式时，CAN停止模式请求才被应答。在其他模式下申请会被忽略。

注意：要将CAN模块设置为CAN停止模式，模块必须处于CAN睡眠模式。要确认模块处于睡眠模式，请检查 PSMODE[1: 0]是否为01B，然后请求CAN停止模式。如果在执行此过程时在CAN接收引脚(CRxD)处发生总线更改，则CAN睡眠模式将自动释放。在这种情况下，无法应答CAN停止模式转换请求(但是，在提供CAN时钟的状态下，在CAN接收引脚(CRxD)发生总线更改后，需要通过软件清除PSMODE0位)。

(2) CAN停止模式状态

进入CAN停止模式后，CAN模块处于以下状态之一。

- 内部操作时钟停止，功耗最小。
- 去唤醒CAN模块，数据可以写到CAN模块控制寄存器(C0CTRL)的PSMODE[1:0]位，不能写到CAN的其他寄存器。
- 除了C0LIPT, C0RGPT, C0LOPT和C0TGPT寄存器，其他CAN0寄存器都可以读。
- CAN0报文缓存寄存器不能被写或读。
- CAN0全局控制寄存器(C0GMCTRL)的MBON位被清除。
- 转到初始化模式的申请不被应答，会被忽略。

(3) 释放CAN停止模式

CAN 停止模式只能通过将01B写入C0CTRL寄存器的PSMODE [1: 0]位来释放。释放CAN停止模式后，CAN 模块进入CAN睡眠模式。

当CAN模块处于CAN停止模式时，请求转换到初始化模式，该请求将被忽略;在进入初始化模式之前。在进入初始化模式之前，CPU必须释放停止模式并随后释放CAN睡眠模式。不可能直接从CAN停止模式进入其他操作模式，不进CAN睡眠模式，该请求将被忽略。

备注：m = 0 到15

21.11.3 省电模式举例

在某些应用程序系统中，可能需要将CPU置于节能模式，以降低功耗。通过使用特定于CAN模块的省电模式和特定于CPU的省电模式，CAN总线可以从省电状态中唤醒CPU。

下面是使用省电模式的例子。

首先，将CAN模块置于CAN睡眠模式($PSMODE = 01B$)。接下来，将CPU置于省电模式。如果在此状态的CAN接收引脚(CRx_D)检测到从隐性到显性的边缘转换，CAN模块中的CINTS5位设置为1。如果C0CTRL寄存器的CIE5位设置为1，则生成唤醒中断(INTC0WUP)。CAN模块自动从CAN睡眠模式($PSMODE = 00B$)释放，并返回到正常操作模式(但是，在提供CAN时钟的状态下，在CAN接收引脚(CRx_D)检测到总线更改后，需要通过软件清除PSMODE0位。CPU响应INTC0WUP，可以释放自己的省电模式，并返回到正常运行模式。

为了进一步降低CPU的功耗，内部时钟(包括CAN模块的时钟)可能会停止。在这种情况下，在CAN模块置于CAN睡眠模式后，提供给CAN模块的操作时钟将停止。然后CPU进入一个省电模式，其中向CPU提供的时钟将停止。如果在此状态的CAN接收引脚(CRx_D)检测到从隐性到显性的边缘转换，CAN模块可以将CINTS5位设置为1并生成唤醒中断(INTC0WUP)，即使没有提供时钟。但是，其他功能不再工作，因为CAN模块的时钟供应已停止，并且模块保持CAN睡眠模式。CPU响应INTC0WUP，释放其省电模式，在振荡稳定时间过后恢复内部时钟(包括向CAN模块提供时钟)的电源，并开始执行指令。恢复时钟电源后，CAN模块会立即从CAN睡眠模式释放，并返回到正常操作模式($PSMODE=00B$)。

21.12 中断功能

CAN模块提供6个不同的中断源。

这些中断源的发生存储在中断状态寄存器中。从六个中断源生成四个单独的中断请求信号。当生成对应于两个或多个中断源的中断请求信号时，可以使用中断状态寄存器标识中断源。发生中断源后，软件必须将相应的中断状态位清除为 0。

表21-20: CAN模块中断源列表

No.	中断状态位		中断使能位		中断申请信号	中断源描述
	名字	寄存器	名字	寄存器		
1	CINTS0 ^注	C0INTS	CIE0 ^注	C0IE	INTC0TRX	报文帧成功的从报文缓存传输
2	CINTS1 ^注	C0INTS	CIE1 ^注	C0IE	INTC0REC	报文缓存接收到有效的报文帧
3	CINTS2	C0INTS	CIE2	C0IE	INTC0ERR	CAN模块状态错误中断 (补充1)
4	CINTS3	C0INTS	CIE3	C0IE		CAN模块协议错误中断(补充2)
5	CINTS4	C0INTS	CIE4	C0IE		CAN模块仲裁丢失中断
6	CINTS5	C0INTS	CIE5	C0IE	INTC0WUP	CAN模块从睡眠模式唤醒中断(补充3)

注：必须将相应报文缓冲区的 C0MCTRLm 寄存器中的 IE 位(报文缓冲区中断启用位)设置为1，以便该报文缓冲区参与中断生成。

补充：

1. 当发送/接收错误计数器处于警告级别或处于错误被动或总线关闭状态时，将生成此中断。
2. 当发生内容错误、形式错误、ACK 错误、位错误或 CRC 错误时，将生成此中断。
3. 当CAN模块从CAN睡眠模式唤醒时，将生成此中断，因为CAN接收引脚处检测到下降沿(CAN 总线从隐性转换到显性)。

备注：m = 0 到15

21.13 诊断功能和特殊操作模式

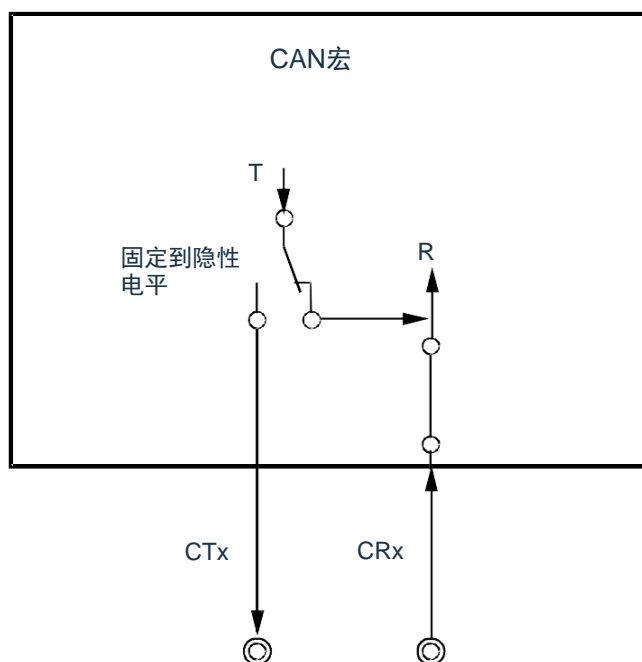
CAN 模块提供仅接收模式、单次模式和自检模式，以支持 CAN 总线诊断功能或特定 CAN 通信方法的操作。

21.13.1 仅接收模式

仅接收模式用于监视接收报文，而不会对CAN总线造成任何干扰，并可用于 CAN 总线分析节点。

例如，此模式可用于自动波特率检测。CAN 模块中的波特速率将更改，直到检测到“有效接收”，因此模块匹配中的波特速率(“有效接收”意味着在CAN协议层中接收了报文帧，而不会发生错误，并且具有适当的ACK连接到CAN总线的节点)。有效的接收不需要报文帧存储在接收报文缓冲区(数据帧)或发送报文缓冲区(远程帧)中。通过设置 COCTRL 寄存器(1)的VALID位来指示有效接收事件。

图21-59：CAN模式仅接收模式下的终端连接



在仅接收模式下，报文帧无法从 CAN 模块传输到 CAN 总线。报文缓存定义为发送报文缓存的转换申请处于挂起状态。

在仅接收模式下，CAN 模块中的CAN发送引脚 (CTxD) 固定到隐性电平。因此，当接收一个报文帧时，即使CAN总线错误被侦测到，也没有主动错误标志能从CAN 模块发送到CAN总线。由于无法从 CAN 模块发出传输，因此传输错误计数器 TEC 永远不会更新。因此，仅接收模式下的 CAN 模块不会进入总线关闭状态。

此外，在有效接收报文帧后，ACK 不会以此模式返回到 CAN 总线。在内部，本地节点识别它已经传输了应答(ACK)。无法将过载帧传输到 CAN 总线。

注意：如果只有两个CAN节点连接到CAN总线，并且其中一个节点在仅接收模式下运行，则CAN总线上没有ACK。由于缺少ACK，传输节点将传输主动错误标志，并重复发送报文帧。传输节点在发送报文帧16次后变为错误被动(假设错误计数器在开头为0，并且未发生其他错误)。传输第17次的报文帧后，传输节点将生成被动错误标志。仅接收模式下的接收节点此时检测到第一个有效的报文帧，并且VALID位首次设置为1。

21.13.2 单次模式

在单次模式下，CAN 协议中定义的自动再传输被关闭(根据 CAN 协议，由于仲裁丢失或错误发生而中止的报文帧传输必须重复(没有软件控制下))。单次模式的所有其他行为与正常操作模式相同。单次模式不能与正常规模模式伴 ABT 结合使用。

单次呆模式根据C0CTRL寄存器的AL位设置禁用了中止报文帧传输的再传输。当AL位被清除为0时，仲裁丢失时和错误发生时的重新传输将禁用。如果AL位设置为1，则错误发生时禁用重新传输，但在仲裁丢失时启用重新传输。因此，以下事件将定义为发送报文缓冲区的报文缓冲区中的TRQ位清除为0。

- 成功的发送了报文帧
- 发送报文帧的发生仲裁丢失
- 当发送报文帧的时候发生错误

通过分别检查C0INTS寄存器的CINTS4和CINTS3位，可以区分事件仲裁损失和错误发生，并且可以通过读取C0LEC寄存器的LEC[2: 0]位来识别错误类型。

报文帧成功传输后，C0INTS寄存器的传输完成中断位CINTS0设置为1。如果此时 C0IE 寄存器的 CIE0 位设置为 1，则输出中断请求信号。

在模拟时间触发的通信方法(例如 TTCAN级别1)时，可以使用单次模式。).

注意：AL 位仅在单次模式下有效。不影响其他操作模式下仲裁损失的再传输操作。

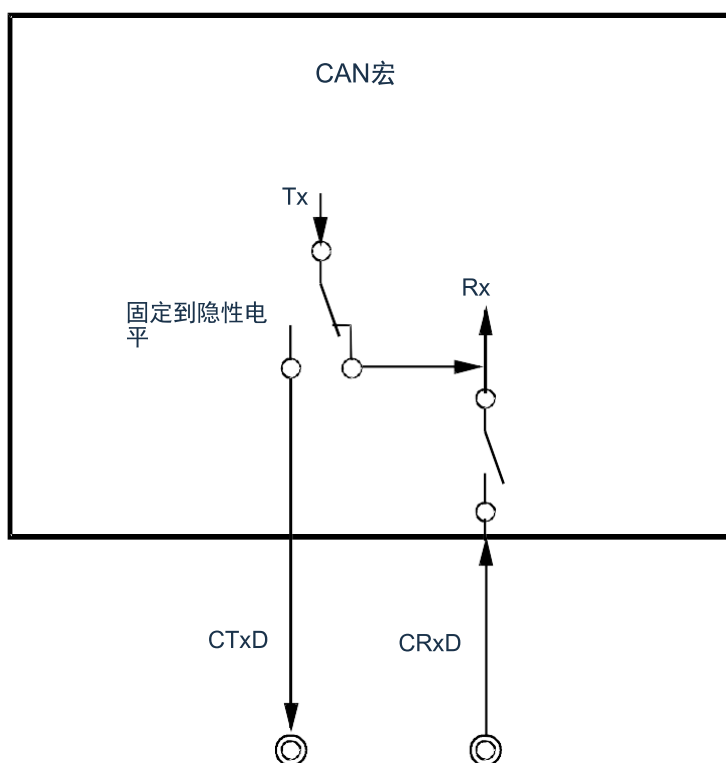
21.13.3 自检模式

在自检模式下，在不影响 CAN 总线的情况下无需将 CAN 节点连接到 CAN 总线，即可测试报文帧传输和接收。

在自检模式下，CAN模块与CAN总线完全断开，但传输和接收在内部循环。CAN发送引脚(CTxD)固定到隐性电平。

如果在 CAN 模块从自检模式进入 CAN 睡眠模式后检测到 CAN 接收引脚 (CRxD) 上游下降沿，则该模块以与其他操作模式相同的方式从 CAN 睡眠模式释放。(但是，要释放 CAN 睡眠模式，在提供 CAN 时钟的状态下，在检测到 CAN 接收引脚 (CRxD) 上的下降沿后，需要通过软件清除 PSMODE0 位)。要使模块保持CAN睡眠模式，需要将CAN接收引脚 (CRxD) 作为端口引脚。

图21-60：自检模式下CAN模块终端连接



21.13.4 操作模式下的接收/发送操作

表21-21显示每个操作模式下接收/发送操作概要。

表21-21：操作模式下发送/接收概要

操作模式	发送数据/远程帧	发送应答	发送错误/过载帧	传输重试	自动块传输(ABT)	设置VALID位	储存数据到报文缓存
初始化模式	否	否	否	否	否	否	否
正常操作模式	是	是	是	是	否	是	是
正常操作模式伴ABT	是	是	是	是	是	是	是
仅接收模式	否	否	否	否	否	是	是
单次模式	是	是	是	否 ^{注1}	否	是	是
自检模式e	是 ^{注2}	是 ^{注2}	是 ^{注2}	是 ^{注2}	否	是 ^{注2}	是 ^{注2}

注1：当仲裁丢失时，COCTRL 寄存器的 AL 位可以控制重新传输。

注2：每个信号不生成到外部，而是生成到CAN模块中。

21.14 时间戳功能

CAN 是异步的串行协议。连接到 CAN 总线的所有节点都有本地的自主时钟。因此，节点的时钟没有关系(即，时钟是异步的，可能有不同的频率)。

但是，在某些应用程序中，需要通过网络(= 全局时基)建立公共时基。为了建立全局时基，需要使用时间戳功能。时间戳功能的基本机制是捕获由CAN总线上的信号触发的计时器值。

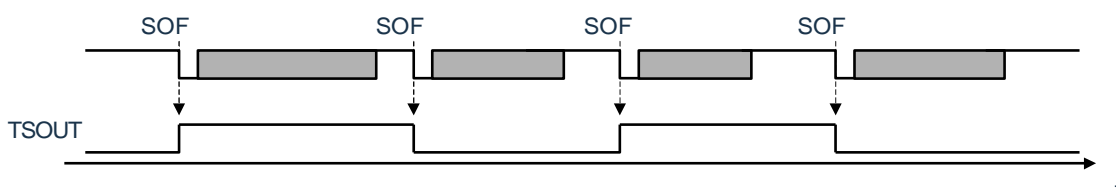
21.14.1 时间戳功能

CAN 控制器支持捕获由特定帧触发的计时器值。除了CAN控制器外，还使用了微控制器系统中的片上16位捕获定时器单元。16 位捕获定时器单元根据触发信号 (TSOUT) 捕获计时器值，用于捕获从 CAN 控制器接收数据帧时输出的计时器值。CPU可以通过读取捕获的值来检索捕获事件的发生时间，即从CAN总线接收的报文的时间戳。TSOUT信号可以从以下两个事件源中选择，并由 C0TS 寄存器的 TSSEL 位指定。

- SOF事件(帧开始) (TSSEL =0)
- EOF事件(结束帧的最后一个位) (TSSEL =1)

TSOUT信号使能需要设置C0TS 寄存器的TSEN位为1。

图21-61：捕获信号 TSOUT 的时序图



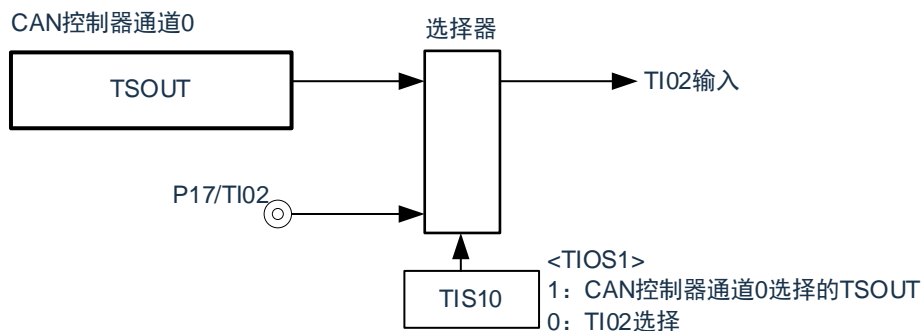
TSOUT 信号在数据帧接收期间发生选定事件时切换其电平(在上述时序图中，SOF 用作触发事件源)。要使用 TSOUT 信号捕获计时器值，运行时必须检测上升沿和下降沿的捕获信号。

此时间戳功能由 C0TS 寄存器的 TSLOCK 位控制。当 TSLOCK 被清除为 0 时，TSOUT 位在选定事件发生时切换。如果 TSLOCK 位设置为 1，则 TSOUT 会在选定事件发生时切换，但切换将停止，因为 TSEN 位在报文存储到报文缓冲区0时自动清除为0。这将抑制TSOUT的后续切换，以便将上次切换的时间戳值(=上次捕获)保存为报文缓冲区 0 中接收数据帧的时间戳值。

注意：使用TSLOCK位的时间戳功能是通过接收报文缓冲区0中的数据帧来停止切换TSOUT位。因此，报文缓冲区0必须设置为接收报文缓冲区。由于接收报文缓冲区无法接收远程帧，因此无法通过接收远程帧来停止TSOUT位的切换。当在报文缓冲区(报文缓冲区 0)以外的报文缓冲区中接收数据帧时，TSOUT 位的切换不会停止。由于这些原因，当 CAN 模块处于具有ABT的正常操作模式时，无法在报文缓冲区0中接收数据帧，因为报文缓冲区 0 必须设置为发送报文缓冲区。因此，在此操作模式下，无法使用TSLOCK位停止切换TSOUT位的功能。

通过切换输入源(使用TIS10)，可以将捕获触发信号(CAN 控制器通道0的TSOUT)输入到定时器阵列单元 0 的通道 2，而无需在外部连接CAN控制器通道0和TI02的TSOUT。

图21-62：输入源的切换



备注：

1. TIS10：定时器输入输出选择寄存器TIOS1的位0 (参考章节6.3.8).
2. 可用的引脚因产品而异。有关详细信息，请参阅 2.1 引脚功能列表.

21.15 波特率设置

21.15.1 波特率设置

确保设置在限制值范围内，以确保CAN控制器的正确操作，如下所示。

- (a) $5TQ \leq SPT$ (采样点) $\leq 17TQ$ $SPT = TSEG1 + 1TQ$
- (b) $8TQ \leq DBT$ (数据位时间) $\leq 25TQ$
 $DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT$
- (c) $1TQ \leq SJW$ (同步跳转宽度) $\leq 4TQ$ $SJW \leq DBT - SPT$
- (d) $4TQ \leq TSEG1 \leq 16TQ$ [3 (TSEG1 [3:0] ≤ 15 的设置值)]
- (e) $1TQ \leq TSEG2 \leq 8TQ$ [0 (TSEG2 [2:0] ≤ 7 的设置值)]

备注: $TQ = 1/F_{TQ}$ (f_{TQ} : CAN协议层基本系统时钟)

TSEG1[3:0]: CAN0位率寄存器(C0BTR)的位0到3

TSEG2[2:0]: CAN0位率寄存器(C0BTR)的位8到10

表21-22显示了满足上述条件的比特率的组合。

表21-22: 可设置比特率组合 (1/3)

有效的位率设置					C0BTR 寄存器设置值		采样点(单位%)
DBT长度	同步段	传播段	相位段1	相位段2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

表 21-22: 可设置比特率组合 (2/3)

有效的位率设置					C0BTR 寄存器设置值		采样点(单位%)
DBT长度	同步段	传播段	相位段1	相位段2	TSEG1[3:0]	TSEG2[2:0]	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

表 21-22: 可设置比特率组合 (3/3)

有效的位率设置					C0BTR 寄存器设置值		采样点(单位%)
DBT长度	同步段	传播段	相位段1	相位段2	TSEG1[3:0]	TSEG2[2:0]	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 ^注	1	2	2	2	0011	001	71.4
7 ^注	1	4	1	1	0100	000	85.7
6 ^注	1	1	2	2	0010	001	66.7
6 ^注	1	3	1	1	0011	000	83.3
5 ^注	1	2	1	1	0010	000	80.0
4 ^注	1	1	1	1	0001	000	75.0

注: 仅当 C0BRP 寄存器的值不是00H时, 使用7或更少的值设置DBT才有效。

注意: 表21-22中的值不能保证网络系统的操作。通过对振荡误差和CAN总线及CAN收发器的延时来检测网络系统的效率。

21.15.2 波特率设置的代表性示例

表21-23和表21-24显示波特率设置的代表性示例。

表21-23：波特率设置的代表性示例($F_{CANMOD} = 8\text{ MHz}$)(1/2)

设置波特率值 (单位:kbps)	COBRP的 划分比率	COBRP 寄存器设置 值	有效位率设置 (单位:kbps)					COBTR寄存器 设置值		采样点(单位:%)
			DBT长度	同步段	传播段	相位段1	相位段2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

注意：表21-23中的值不能保证网络系统的操作。通过对晶振荡误差和CAN总线及CAN收发器的延时来检测网络系统的效率。

表21-23: 波特率设置的代表性示例($F_{CANMOD} = 8 \text{ MHz}$)(2/2)

设置波特率值 (单位:kbps)	COBRP的 划分比率	COBRP 寄存器设置 值	有效位率设置 (单位:kbps)					COBTR寄存器 设置值		采样点(单位:%)
			DBT长度	同步段	传播段	相位段1	相位段2	TSEG1 [3:0]	TSEG2 [2:0]	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

注意: 表21-23中的值不能保证网络系统的操作。通过对振荡误差和CAN总线及CAN收发器的延时来检测网络系统的效率。

表21-24: 波特率设置的代表性示例($F_{CANMOD} = 16 \text{ MHz}$)(1/2)

设置波特率值 (单位:kbps)	COBRP的 划分比率	COBRP 寄存器设置 值	有效位率设置 (单位:kbps)					COBTR寄存器 设置值		采样点(单位:%)
			DBT长度	同步段	传播段	相位段1	相位段2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

注意: 表21-24中的值不能保证网络系统的操作。通过对振荡误差和CAN总线及CAN收发器的延时来检测网络系统的效率。

表21-24: 波特率设置的代表性示例($F_{CANMOD} = 16 \text{ MHz}$)(2/2)

设置波特率值 (单位:kbps)	COBRP的 划分比率	COBRP 寄存器设置 值	有效位率设置 (单位:kbps)					COBTR寄存器 设置值		采样点(单位:%)
			DBT长度	同步段	传播段	相位段1	相位段2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

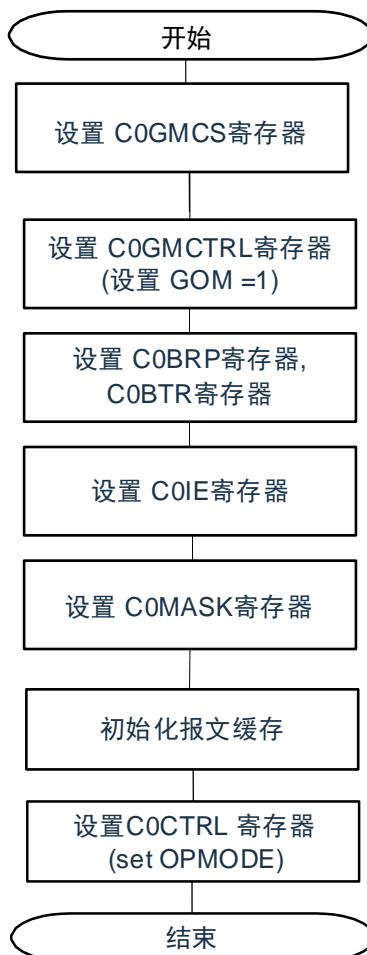
注意: 表21-24中的值不能保证网络系统的操作。通过对振荡误差和CAN总线及CAN收发器的延时来检测网络系统的效率。

21.16 CAN控制器的操作

本章的操作流程是CAN控制器的操作处理流程。开发请参考本章的处理流程。

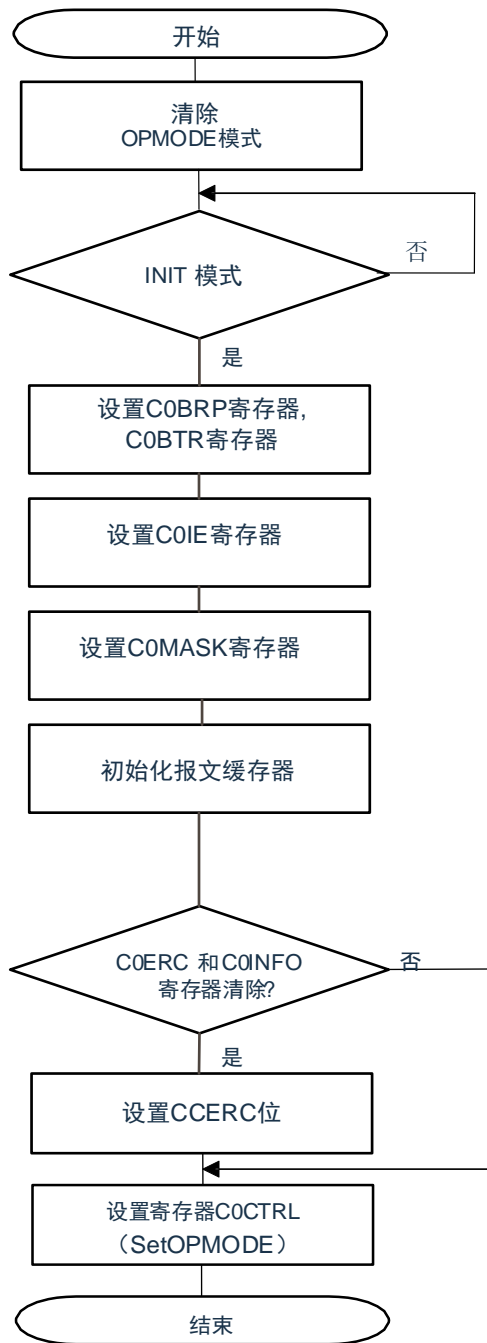
备注：m = 0 到15

图21-63：初始化



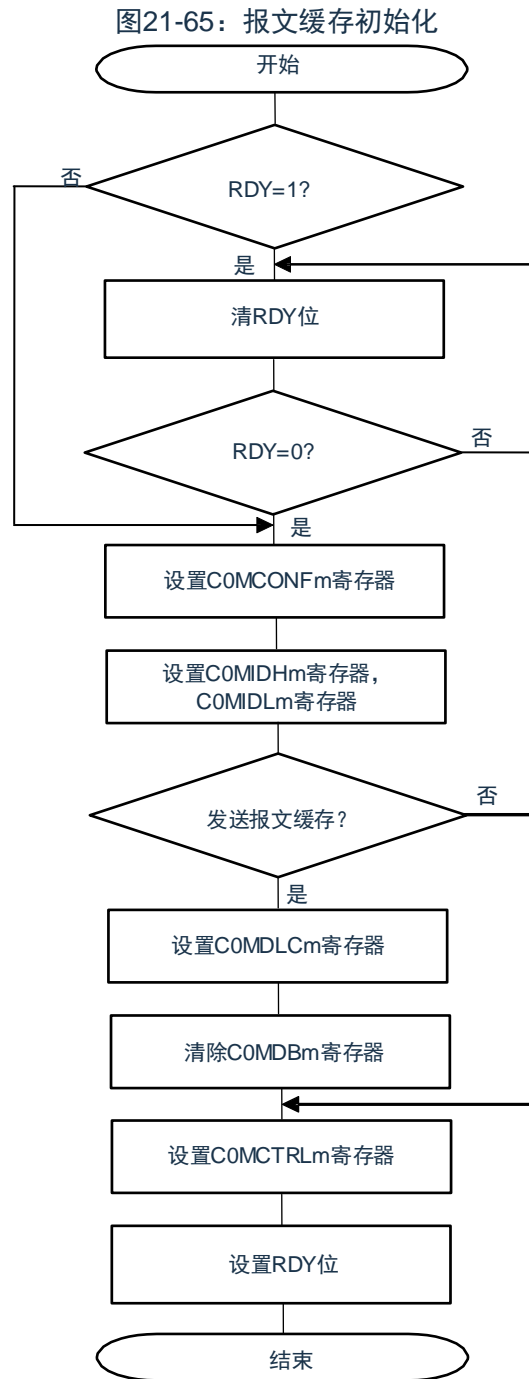
备注：OPMODE：正常的操作模式，正常操作模式伴ABT，仅接收模式，单次模式，自检模式

图21-64：重新初始化



注意：将CAN模块设置为初始化模式后，不要立即设置为另一个操作模式。如果需要立即将模块设置为其他操作模式，请访问COCTRL和COGMCTRL寄存器以外的寄存器(例如，设置报文缓冲区)。

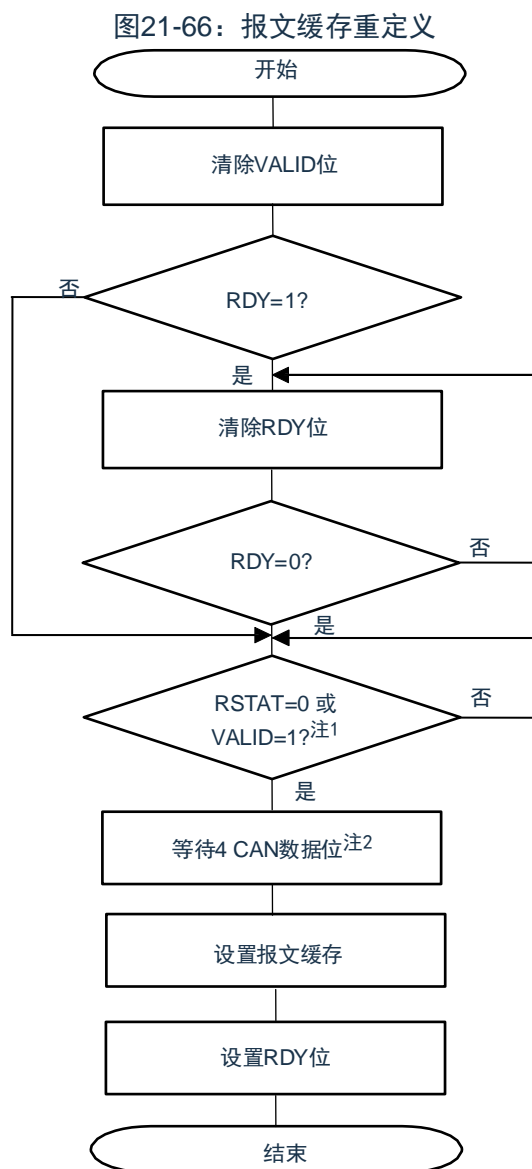
备注：OPMODE：正常操作模式，正常操作模式伴ABT，仅接收模式，单次模式,自检模式。



注意：

1. 在初始化报文缓存前必须清除RDY位.
2. 对没有使用的报文缓存参考下面的设置.
 - 清除COMCTRLm寄存器的RDY, TRQ和DN位为0.
 - 清除COMCONFm寄存器的MA0位为0.

图21-66展示接收报文缓存处理 (COMCONFm寄存器MT[2:0]位=001B到101B).



注1：确认收到报文是因为在完全收到报文后RDY 位被设置。

注2：通过等待额外的 4 CAN 数据位，避免在存储报文接收操作期间重新定义报文缓冲区。

图21-67显示传输过程中发送报文缓冲区的处理(COMCONFm寄存器MT[2:0]位=000B).

图21-67：发送过程中报文缓存重定义

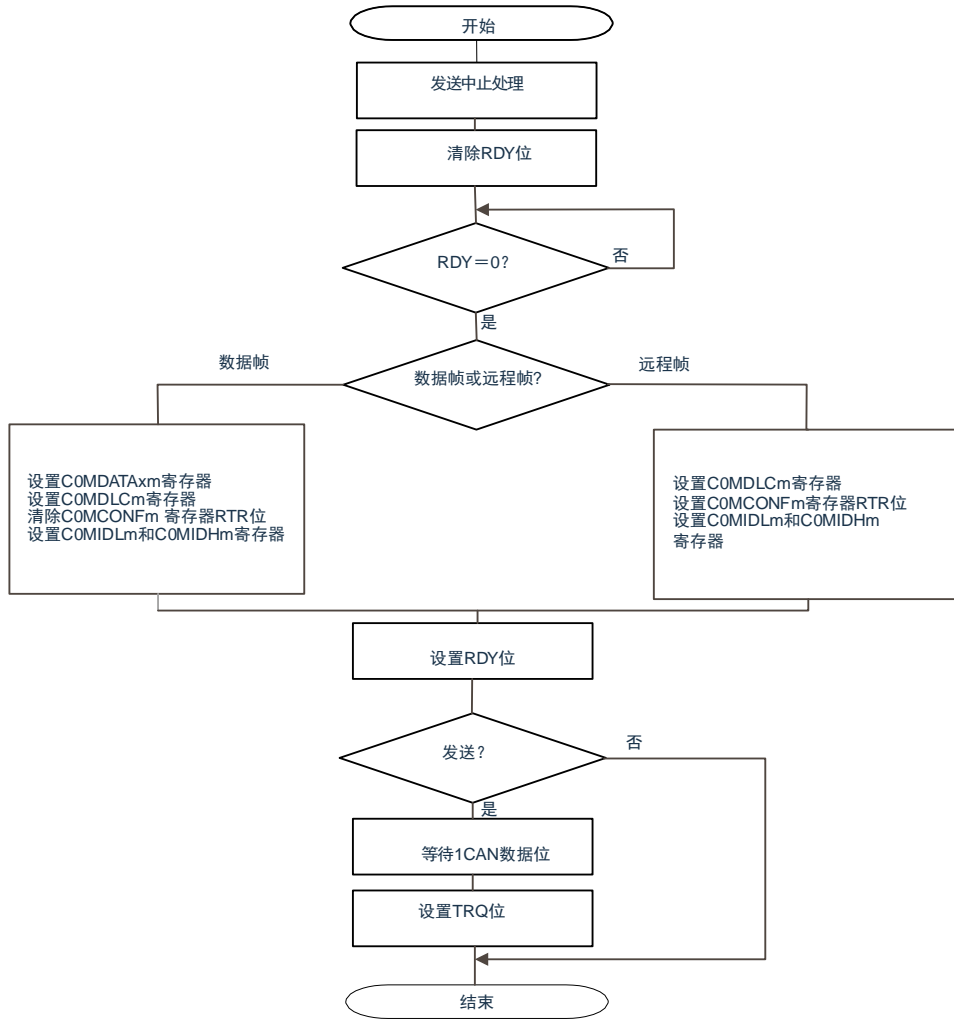
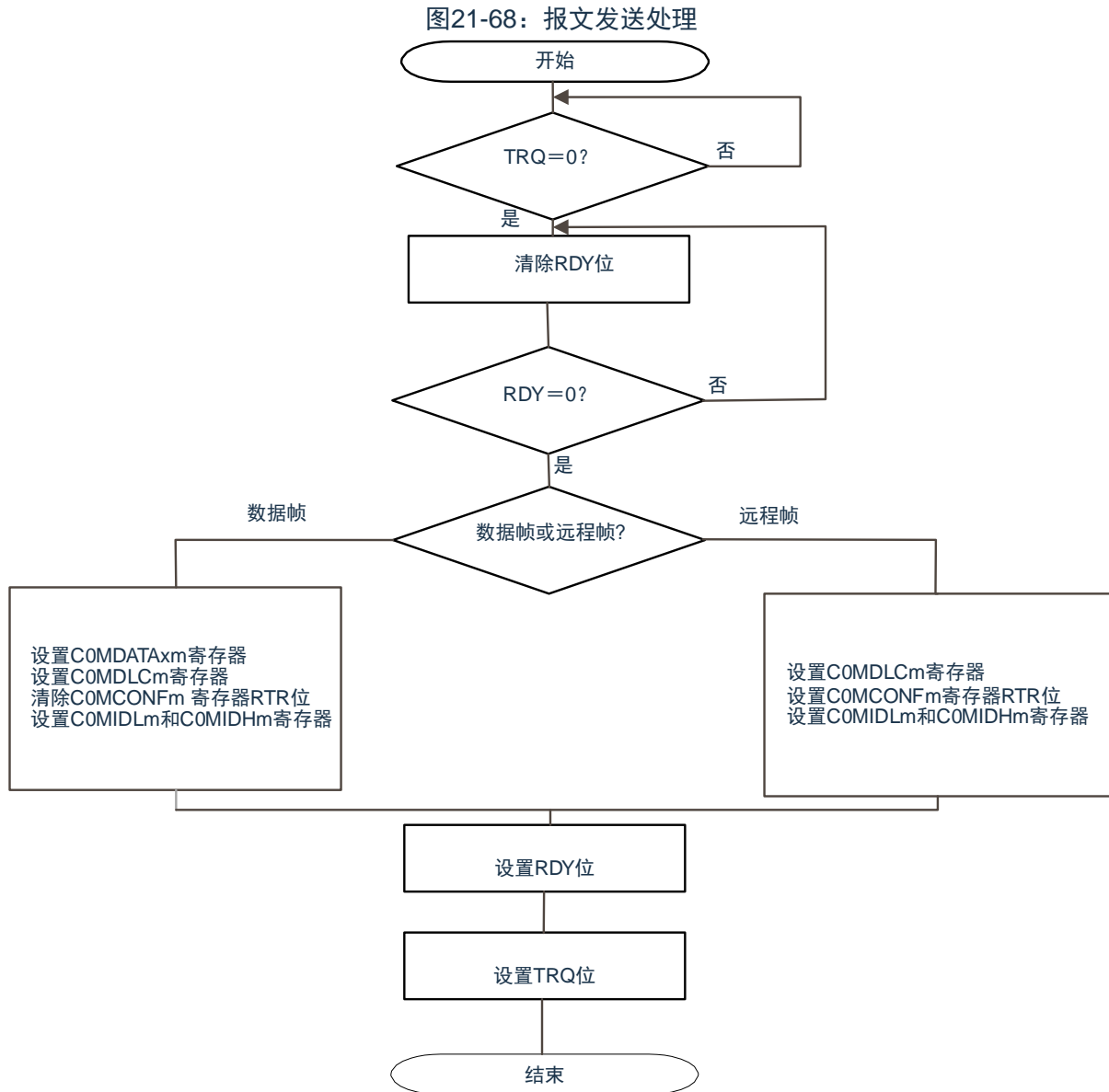


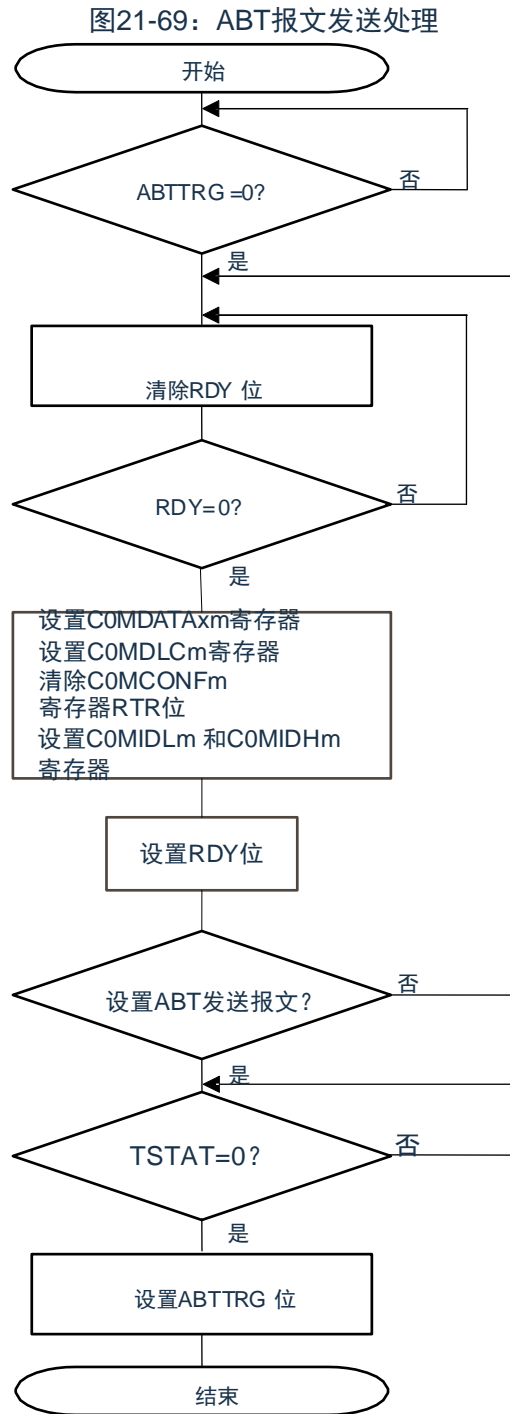
图21-68显示发送报文缓存处理 (COMCONFm寄存器MT[2:0]位=000B).



注意:

1. RDY设置后设置TRQ位.
2. RDY位和TRQ位不能同时设置.

图21-69显示发送报文缓存处理(COMCONFm寄存器MT[2:0]位=000B).

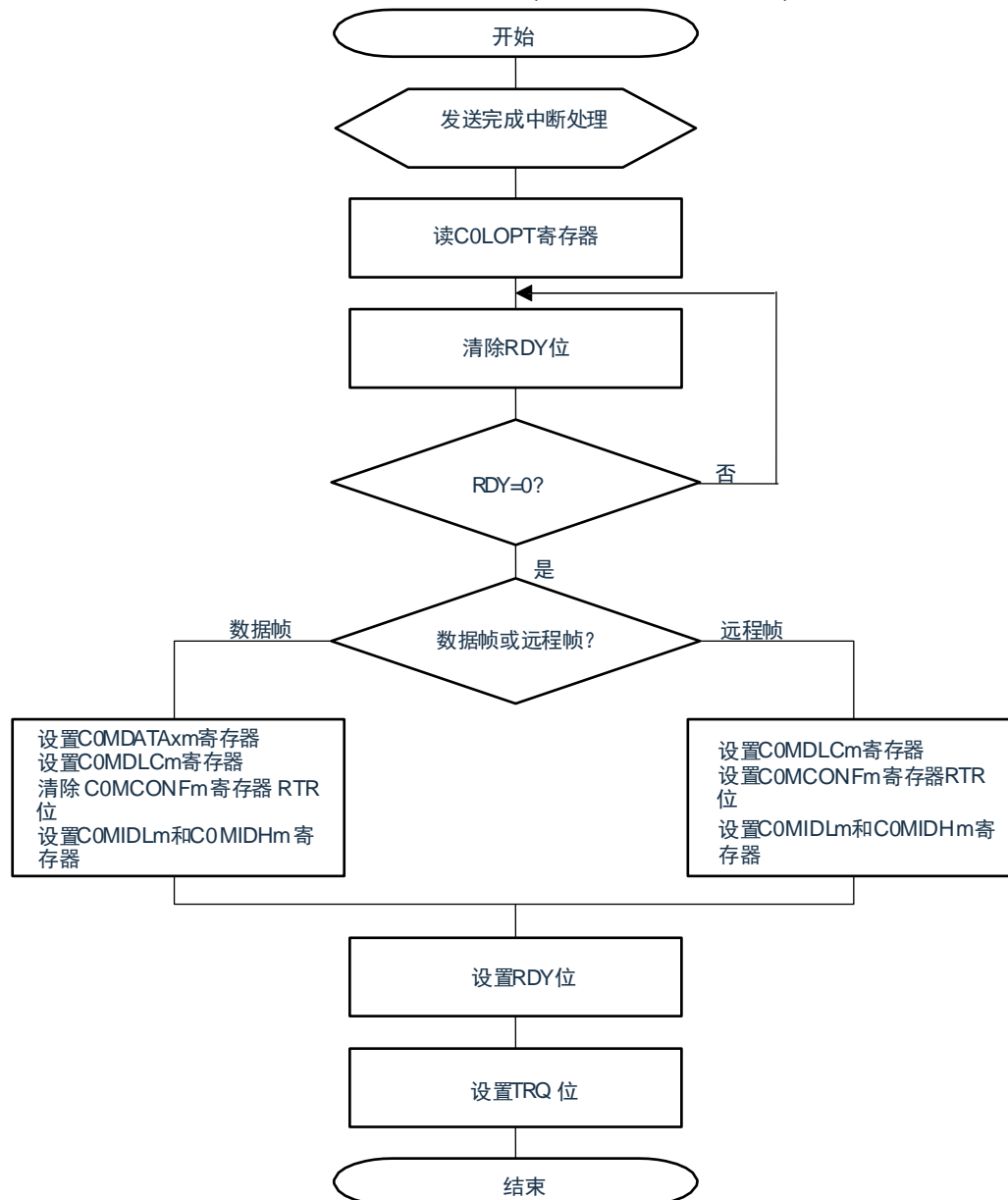


注意: 在TSTAT位清除为0后, 应将ABTTRG位设置为1。必须连续检查TSTAT位并将ABTTRG位设置为1。

备注: 此处理(使用ABS的正常操作模式)只能应用于报文缓冲区0到7。

对于ABT报文缓冲区以外的报文缓冲区, 请参阅图21-68。

图21-70：发送通过中断(使用C0LOPT寄存器)



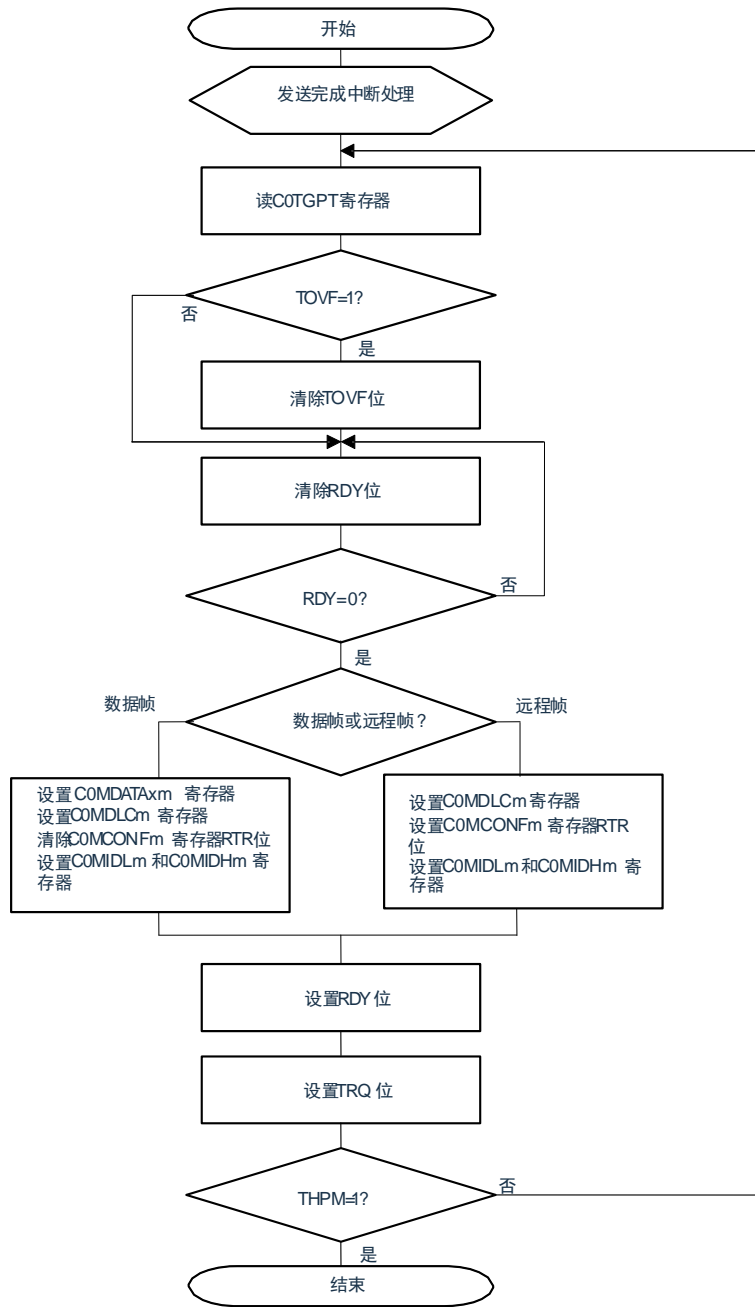
注意：

1. TRQ位在RDY设置后设置.
2. RDY位和TRQ位不能同时设置.

备注：在中断开始和结束时检查MBON标志，以便检查对报文缓冲区和TX历史记录列表寄存器的访问，以防执行挂起的睡眠模式。如果检测到MBON被清除，则在再次设置MBON后，必须丢弃结果和再次处理操作。

推荐在处理TX中断前删除一些睡眠模式请求。

图21-71: 通过中断发送(使用C0TGPT寄存器)



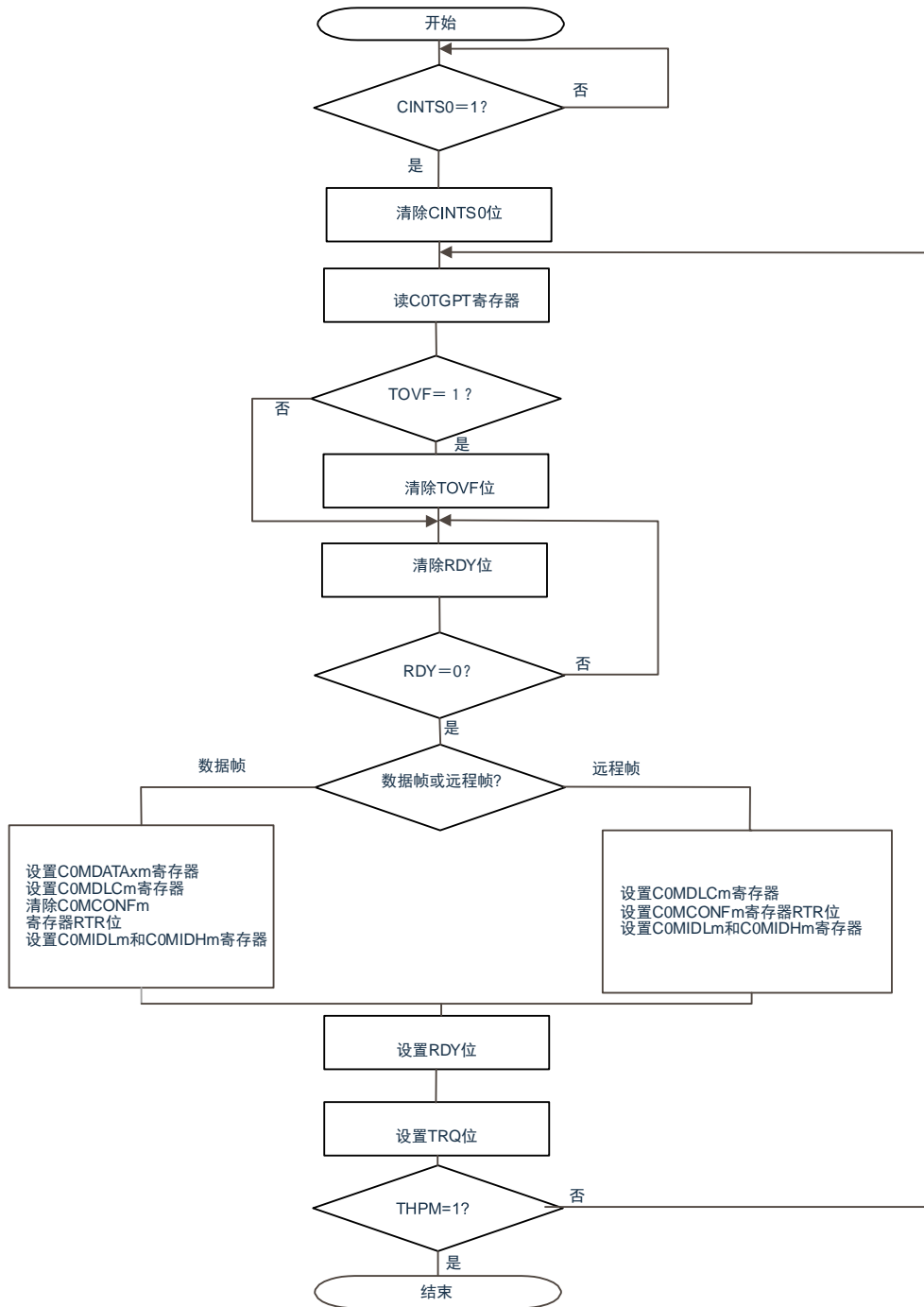
注意:

1. TRQ位的设置在设置RDY位后.
2. RDY位和TRQ位不能同时设置.

备注:

1. 在中断开始和结束时检查MBON标志, 以便检查对报文缓冲区和TX历史记录列表寄存器的访问, 以防执行挂起的睡眠模式. 如果检测到MBON被清除, 则在再次设置MBON后, 必须丢弃结果和再次处理操作. 推荐在处理TX中断前删除一些睡眠模式请求.
2. 如果TOVF被设置, 则传输历史记录列表不一致. 考虑去扫描所有已配置的传输缓冲区来完成传输.

图21-72：通过软件轮询传输



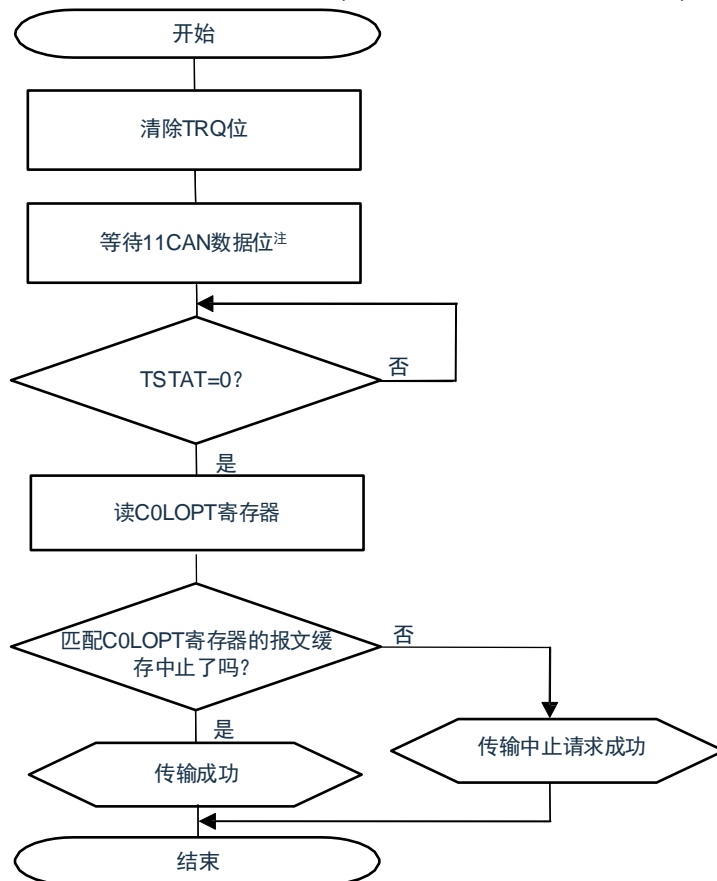
注意：

1. TRQ位的设置在设置RDY位后.
2. RDY位和TRQ位不能同时设置.

备注：

1. 在中断开始和结束时检查MBON标志，以便检查对报文缓冲区和TX历史记录列表寄存器的访问，以防执行挂起的睡眠模式。如果检测到MBON被清除，则在再次设置MBON后，必须丢弃结果和再次处理操作.
2. 如果TOVF被设置，则传输历史记录列表不一致。考虑去扫描所有已配置的传输缓冲区来完成传输.

图21-73：传输中止处理(除正常操作模式伴ABT以外)

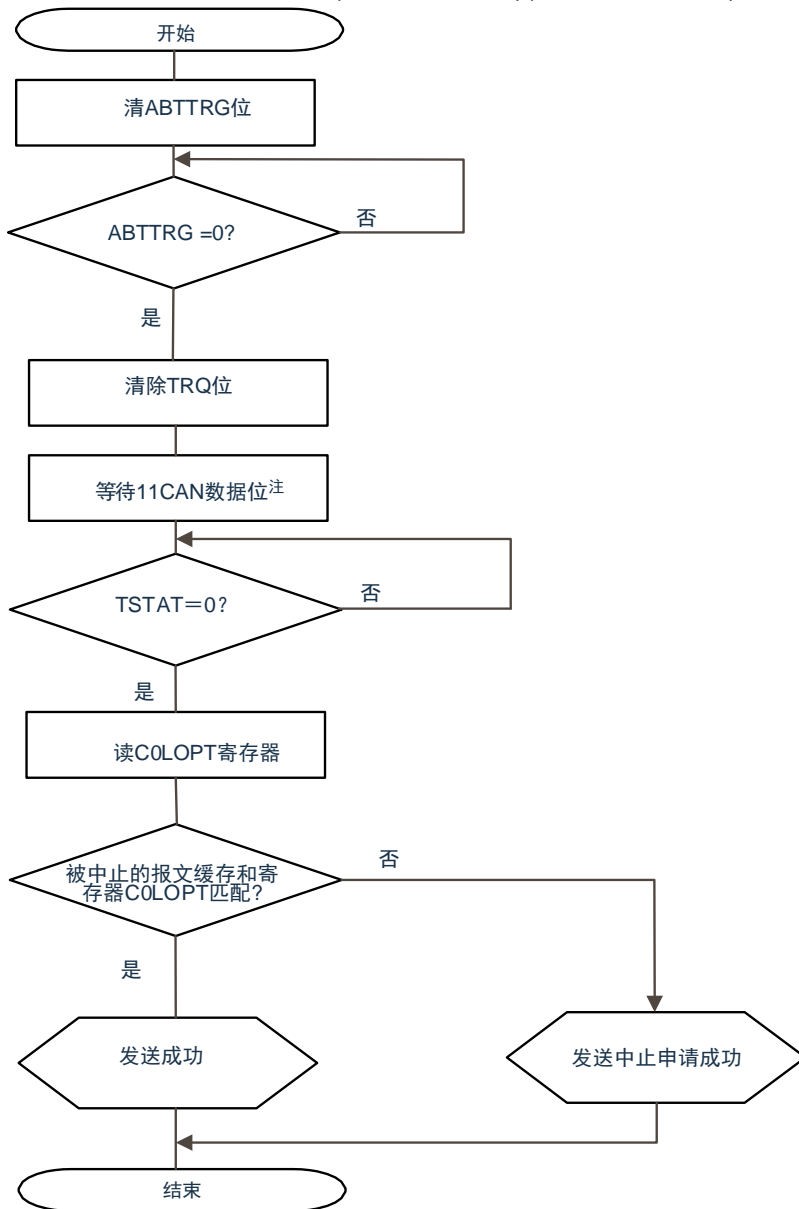


注：即使TRQ位被清除，也有可能在中止的情况下启动传输，因为在11位、帧间空间(3位)和挂起传输(8位)之间可能已接受对协议层的传输请求。

注意：

1. 通过清除TRQ位(而不是RDY位)执行传输请求中止处理。
2. 在发出睡眠模式转换请求之前，请确认没有剩余的传输请求使用此处理。
3. 用户应用程序可以定期检查 TSTAT 位，也可以在传输完成中断后进行检查。
4. 在传输中止处理进行中，不要执行新的传输请求，包括在其他报文缓冲区中。
5. 当传输在相同的报文缓存持续进行或者只有一个报文缓存被使用时，判断传输中止请求是否成功，可能引起冲突。在这种情况下，使用C0TGPT寄存器指示的历史信息等来判断。

图21-74：发送中止处理(ABT 传输除外)(正常操作伴ABT)



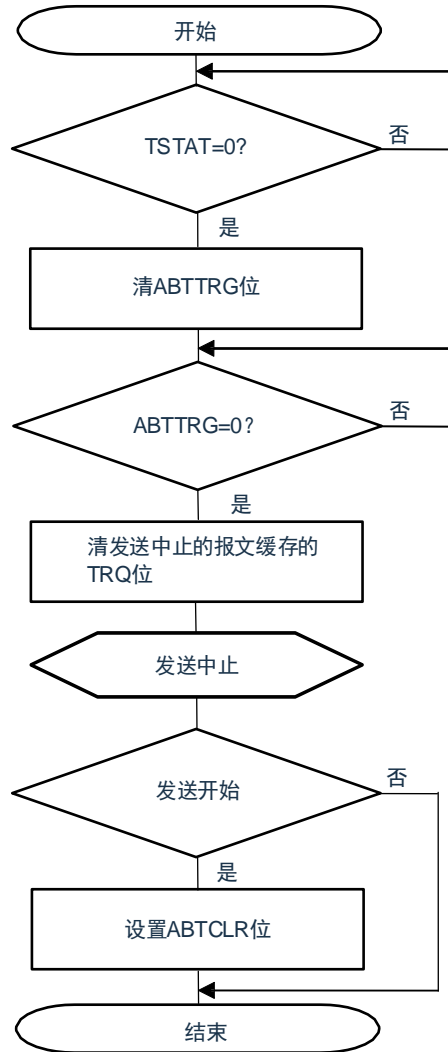
注：即使 TRQ 位被清除，也有可能在不中止的情况下启动发送，因为在 11 位、帧间空间(3 位)和挂起传输(8 位)之间可能已经接受了对协议层的传输请求..

注意：

1. 通过清除 TRQ 位(而不是 RDY 位)执行传输请求中止处理。
2. 在发出睡眠模式转换请求之前，确认没有传输请求使用此处理。
3. 用户应用程序可以定期检查 TSTAT 位，也可以在传输完成中断后进行检查。
4. 在传输中止处理进行中，不要执行新的传输请求，包括在其他报文缓冲区中。
5. 当同一报文缓冲区的传输是连续的或只使用一个报文缓冲区时，判断传输中止请求是否成功，都可能导致冲突。在这种情况下，使用C0TGPT寄存器指示的历史信息来判断。

图21-75显示当传输ABT报文缓存中止时发送报文停止没有跳过恢复的处理

图21-75: ABT发送中止处理(正常操作模式伴ABT)

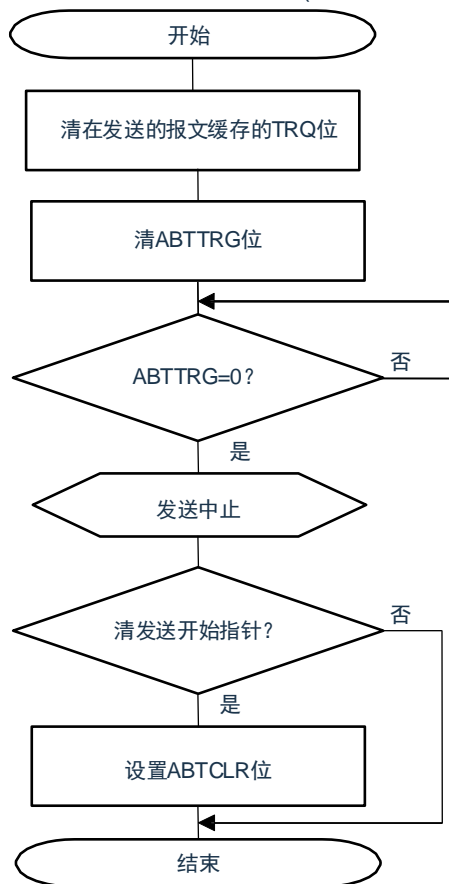


注意:

1. 正在进行ABT传输中止处理时, 不要设置任何传输请求.
2. 在 ABTTRG 位清除后(在 ABT 模式中止后)发出 CAN 睡眠模式/CAN 停止模式转换请求, 按照图 21-75或图21-76所示的过程。在 ABT 区域以外的区域中清除传输请求时, 请按照图21-74所示的过程操作。

图21-76显示当传输ABT报文缓存中止时发送报文停止跳过恢复的处理。

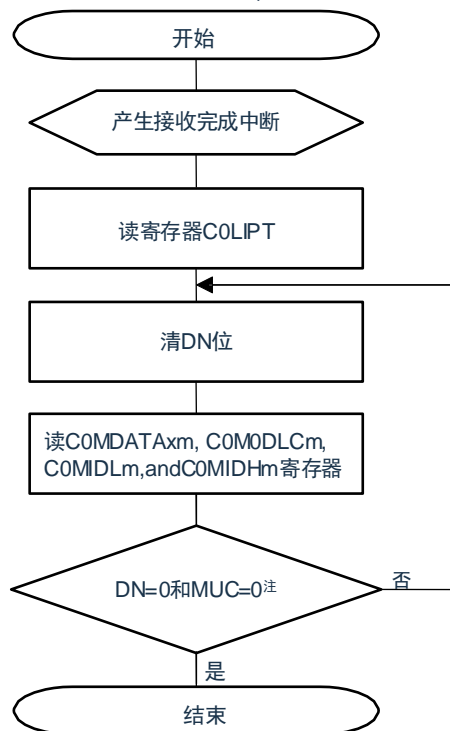
图21-76：ABT发送请求中止处理(正常操作模式伴ABT)



注意：

1. 在 ABT 传输中止处理正在进行时，不要设置任何传输请求。
2. 在 ABTTRG 清除(在 ABT 模式停止)后发出 CAN 睡眠模式/CAN 停止模式请求,按照图21-75或图21-76所示的过程。在 ABT 区域以外的区域中清除传输请求时，请按照图21-74所示的过程操作。

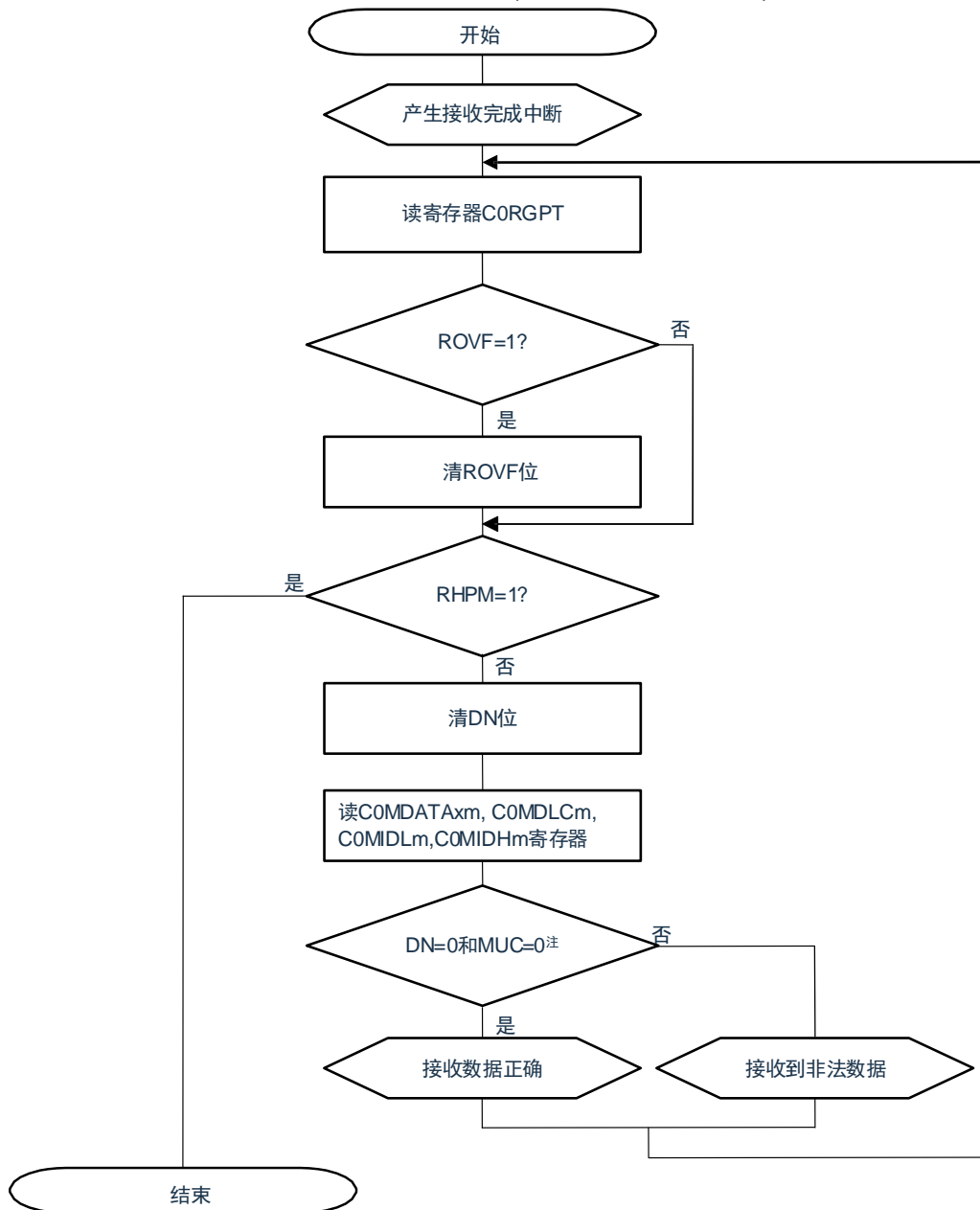
图21-77：通过中断接收(使用寄存器COLIPT)



注：使用读检查MUC和DN位

备注：要检查中断开始和结尾的 MBON 标志，以便检查对报文缓冲区和接收历史记录列表寄存器的访问，以防已挂起的睡眠模式的执行。如果检测到 MBON被清除，则在再次设置 MBON 后，必须丢弃处理的操作和结果然后再处理。在处理 RX 中断之前，建议取消任何睡眠模式请求。

图21-78: 通过中断接收(使用寄存器C0RGPT)

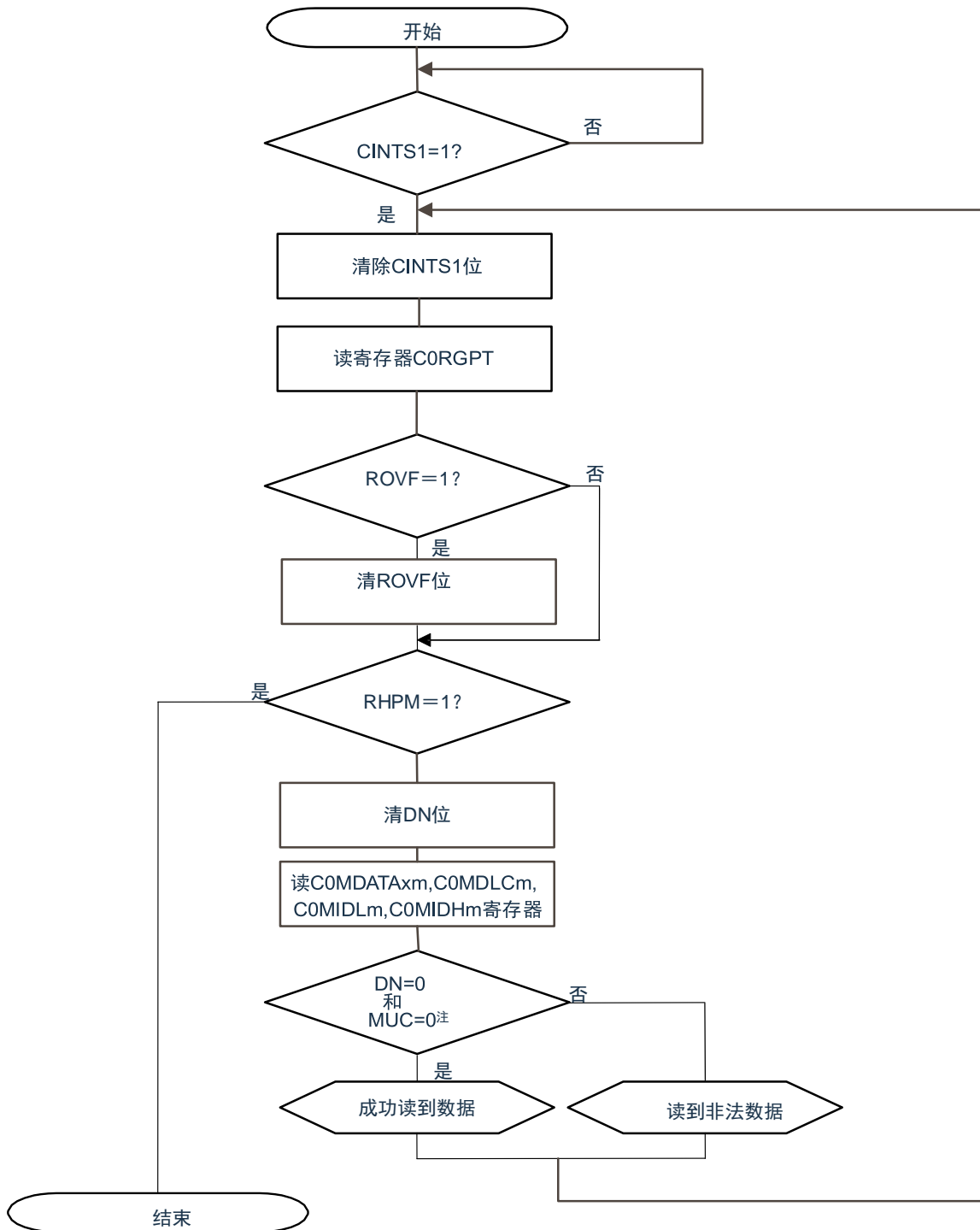


注：使用读检查MUC和DN位.

备注：

1. 要检查中断开始和结尾的 MBON 标志，以便检查对报文缓冲区和接收历史记录列表寄存器的访问，以防已挂起的睡眠模式的执行。如果检测到 MBON被清除，则在再次设置 MBON 后，必须丢弃处理的操作和结果然后再处理。在处理 RX 中断之前，建议取消任何睡眠模式请求。
2. 如果 ROVF 设置，则接收历史记录列表不一致。考虑扫描所有配置的接收缓冲区以进行接收。

图21-79：通过软件轮询接收

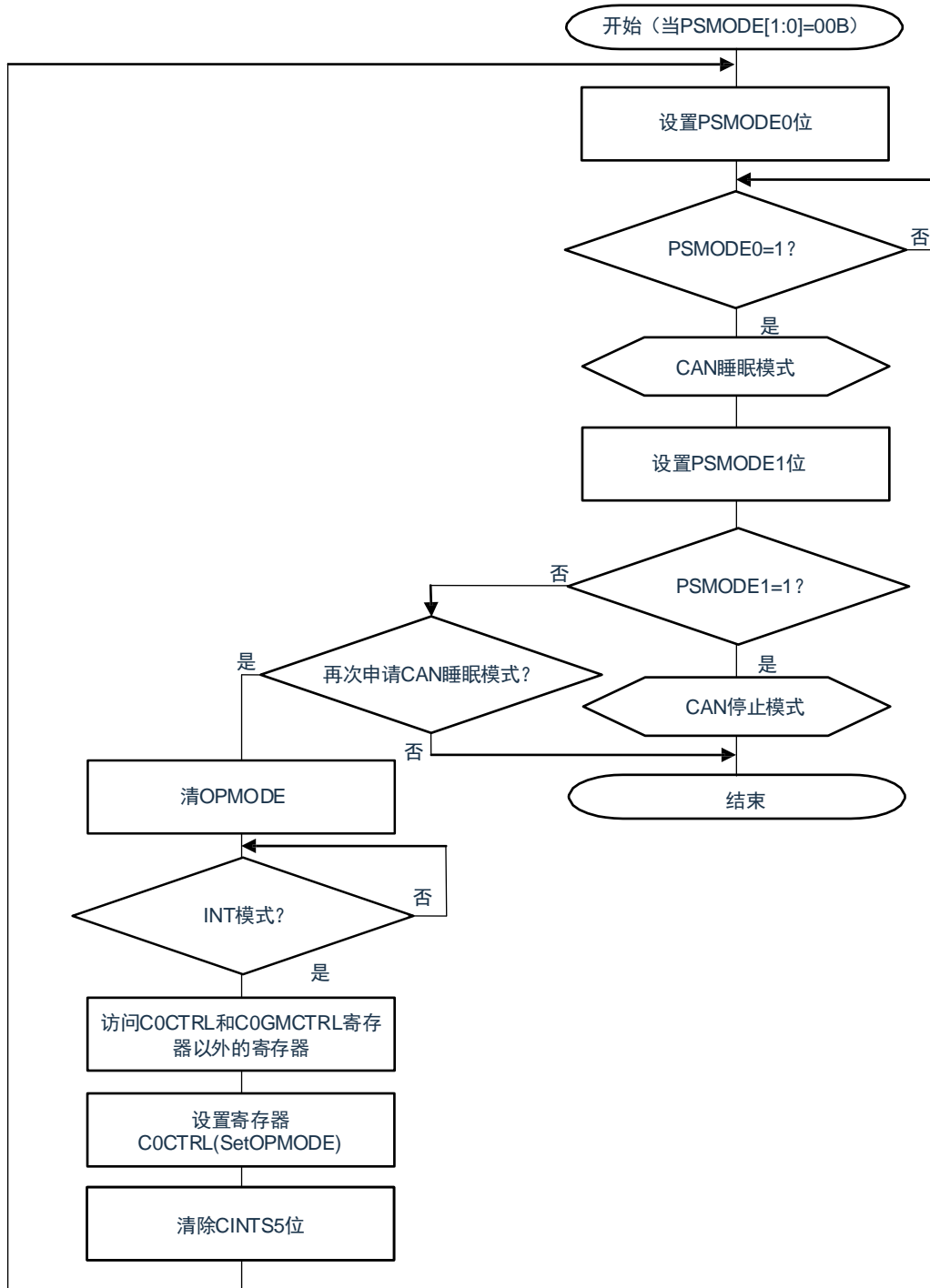


注：使用读检查MUC和DN位。

备注：

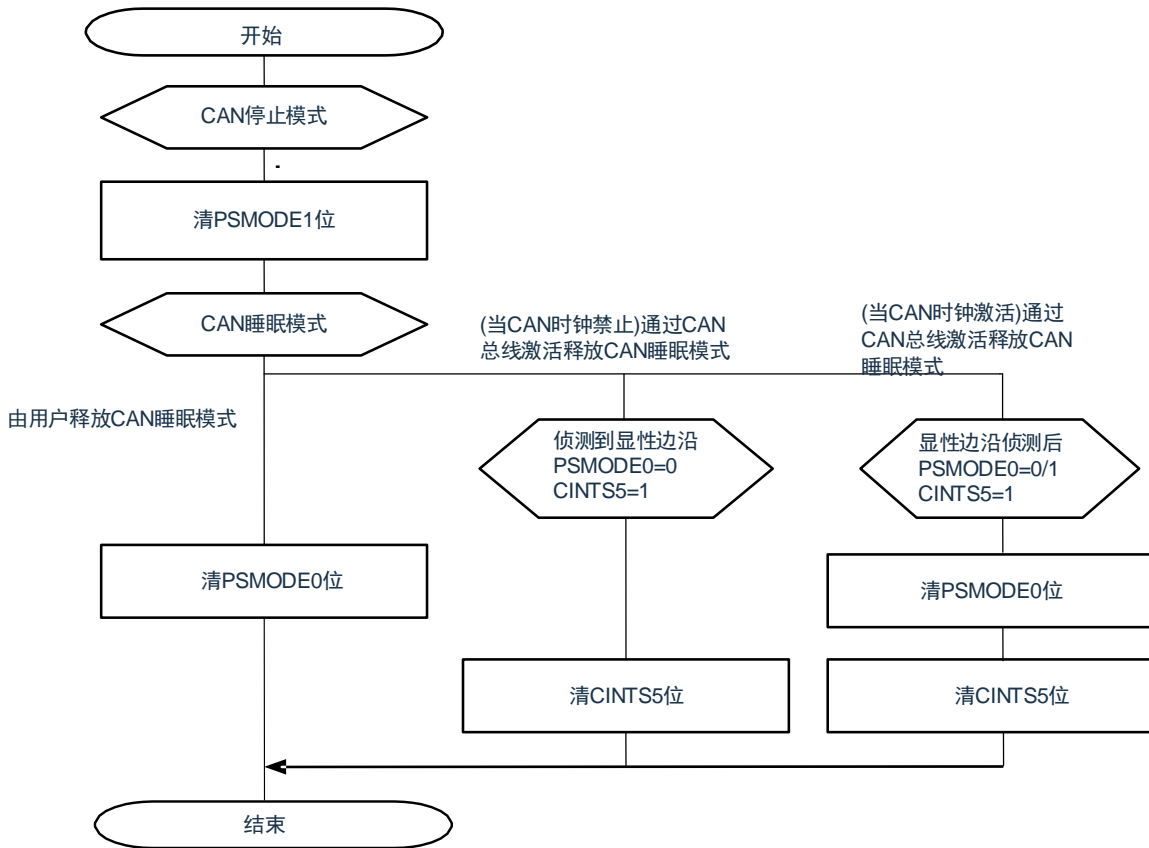
1. 要检查轮询开始和结束的 MBON 标志，以便检查对报文缓冲区和接收历史记录列表寄存器的访问，以防已挂起的睡眠模式被执行。如果检测到 MBON被清除，则在再次设置 MBON 后，必须丢弃处理的操作和结果然后再处理。
2. 如果ROVF设置，则接收历史记录列表不一致。考虑扫描所有配置的接收缓冲区以进行接收。

图21-80：设置CAN睡眠/停止模式



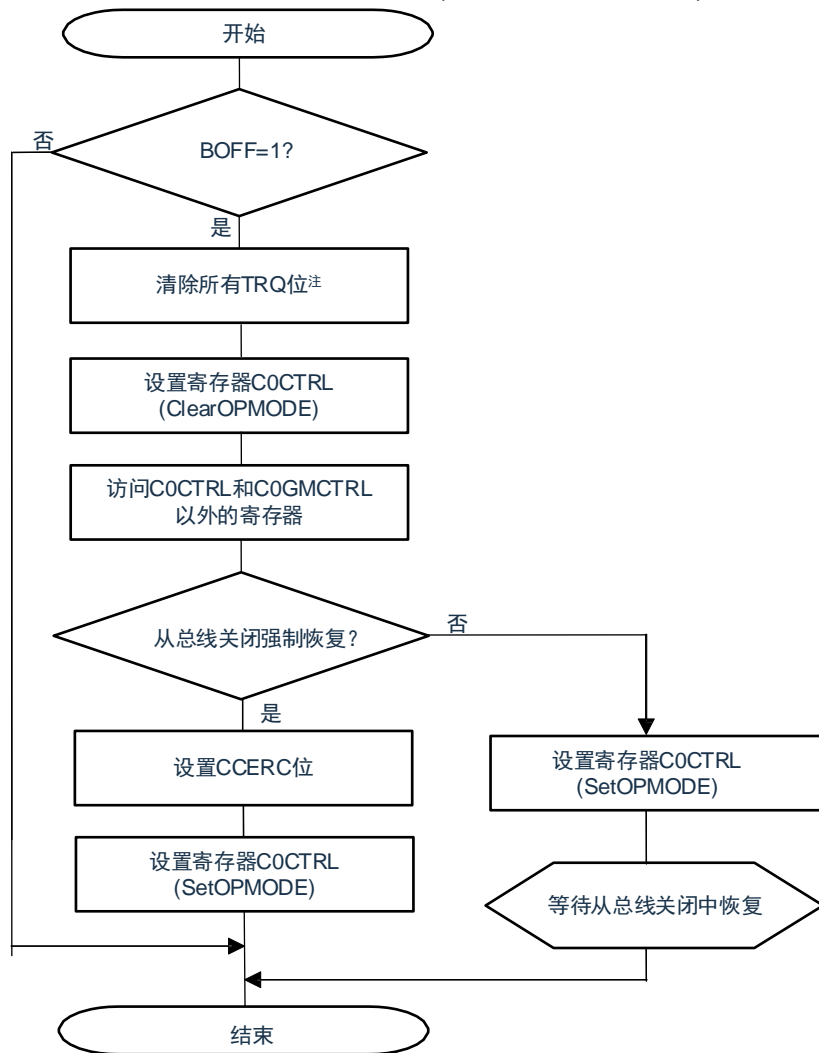
注意：要在请求 CAN 睡眠模式之前中止传输，请根据图21-73或图21-74执行。

图21-81：清CAN睡眠/停止模式



备注：“CAN时钟关闭”：通过 CPU 待机模式，CAN 模块时钟已关闭，CAN 模块处于睡眠模式。

图21-82：总线关闭恢复 (正常操作模式伴ABT)

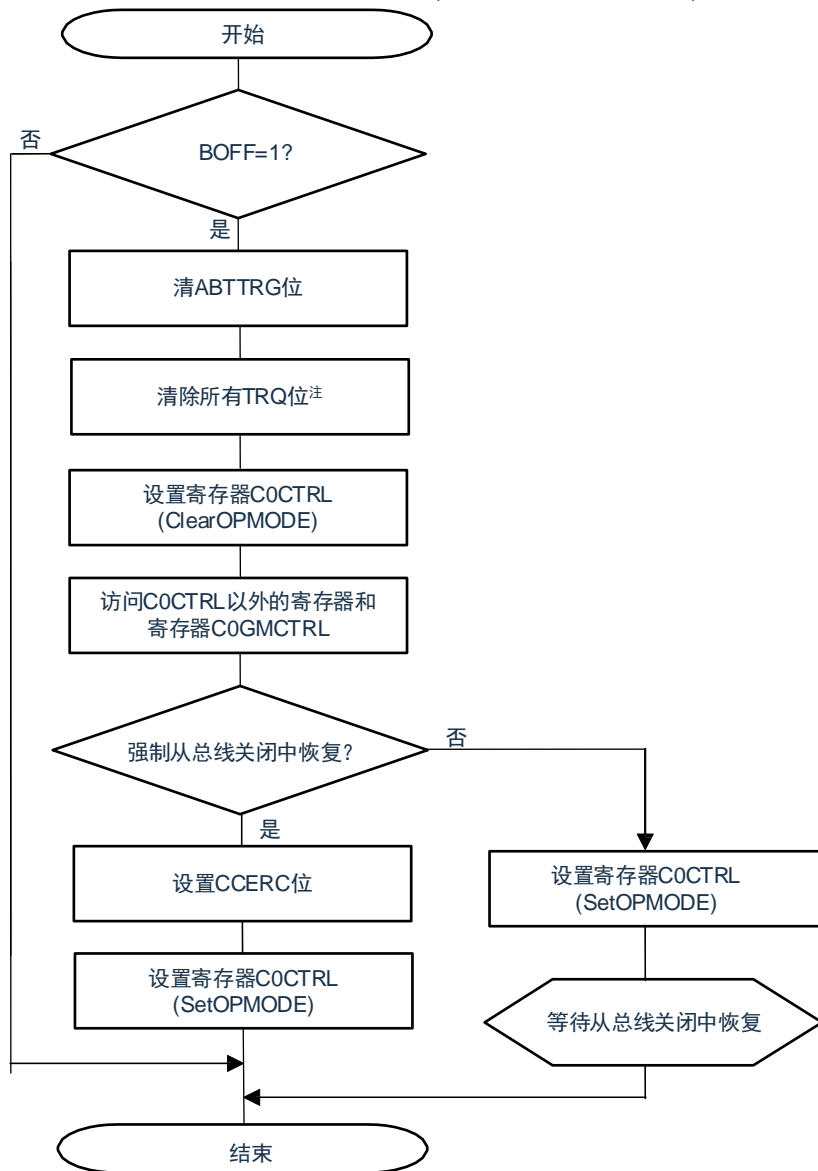


注：在启动总线关闭恢复序列之前通过清除 RDY 位来执行报文缓冲区的重新初始化时，清除所有 TRQ 位。

注意：当请求从初始化模式转换到其他任何操作模式，以在总线关闭恢复序列中再次执行总线关闭恢复序列时，将清除接收错误计数器。因此，有必要再次检测总线上的 11 个连续隐性位 128 次。

备注：OPMODE：正常操作模式, 正常操作模式伴ABT, 仅接收模式, 单次模式, 自检模式

图21-83: 总线关闭恢复(正常操作模式伴ABT)



注：在启动总线关闭恢复序列之前通过清除 RDY 位来执行报文缓冲区的重新初始化时，清除所有 TRQ 位。

注意：当请求从初始化模式转换到其他操作模式，以在总线关闭恢复序列中再次执行总线关闭恢复序列时，将清除接收错误计数器。因此，有必要再次检测总线上的 11 个连续隐性位 128 次。

备注：OPMODE：正常操作模式, 正常操作模式伴ABT, 仅接收模式, 单次模式, 自检模式

图21-84：正常关闭处理

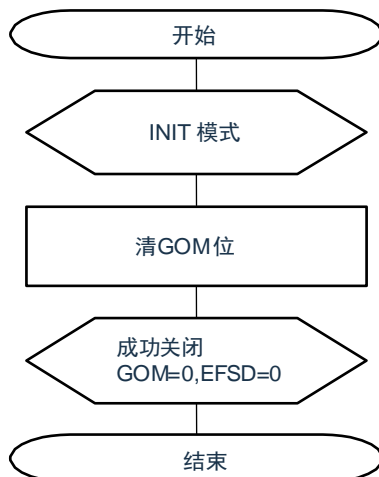
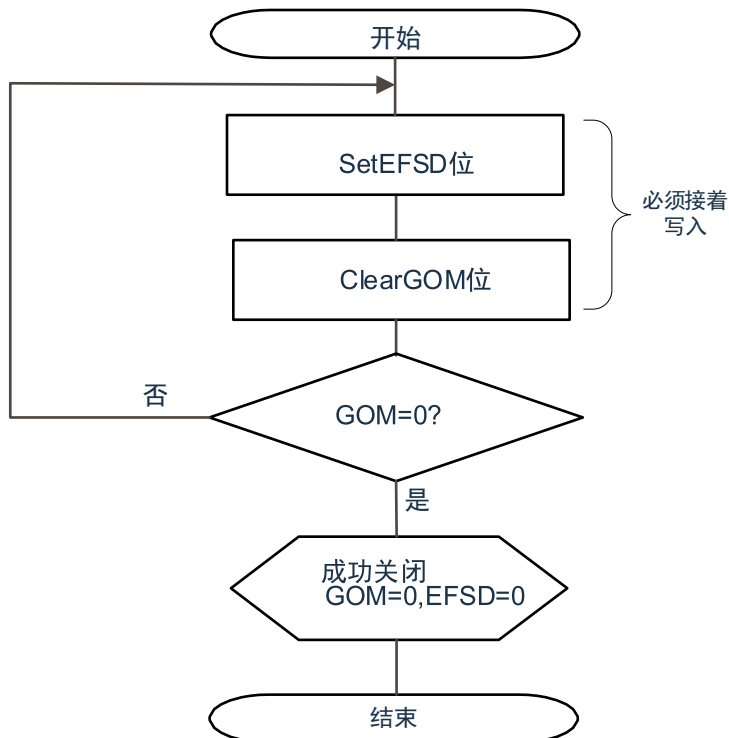


图21-85：强制关闭处理



注意：在设置 EFSD 位和清除 GOM 位之间，不要通过软件读取或写入任何寄存器。
 请注意，如果发生中断或 DMA，则不被视为顺序访问，强制关闭请求无效。

图21-86：错误处理

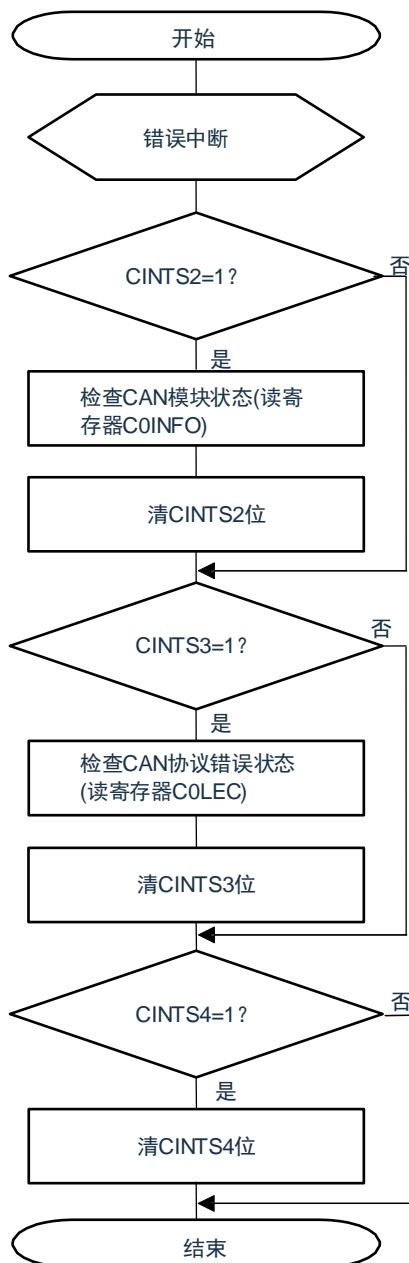
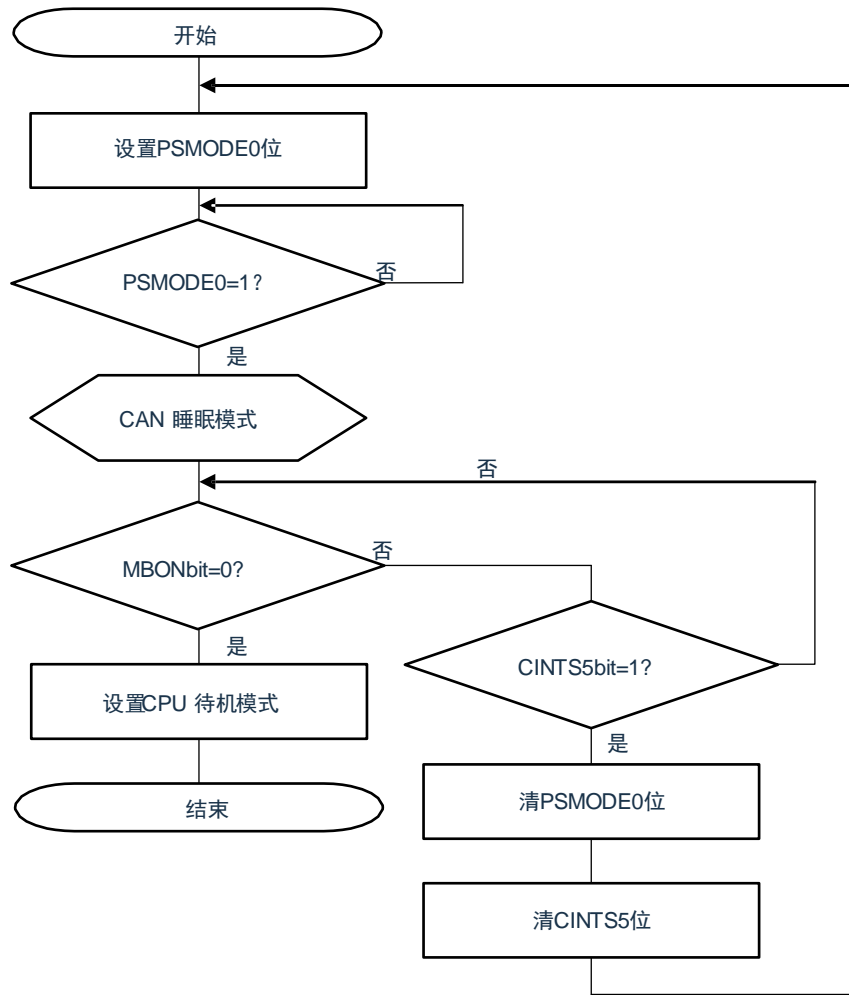
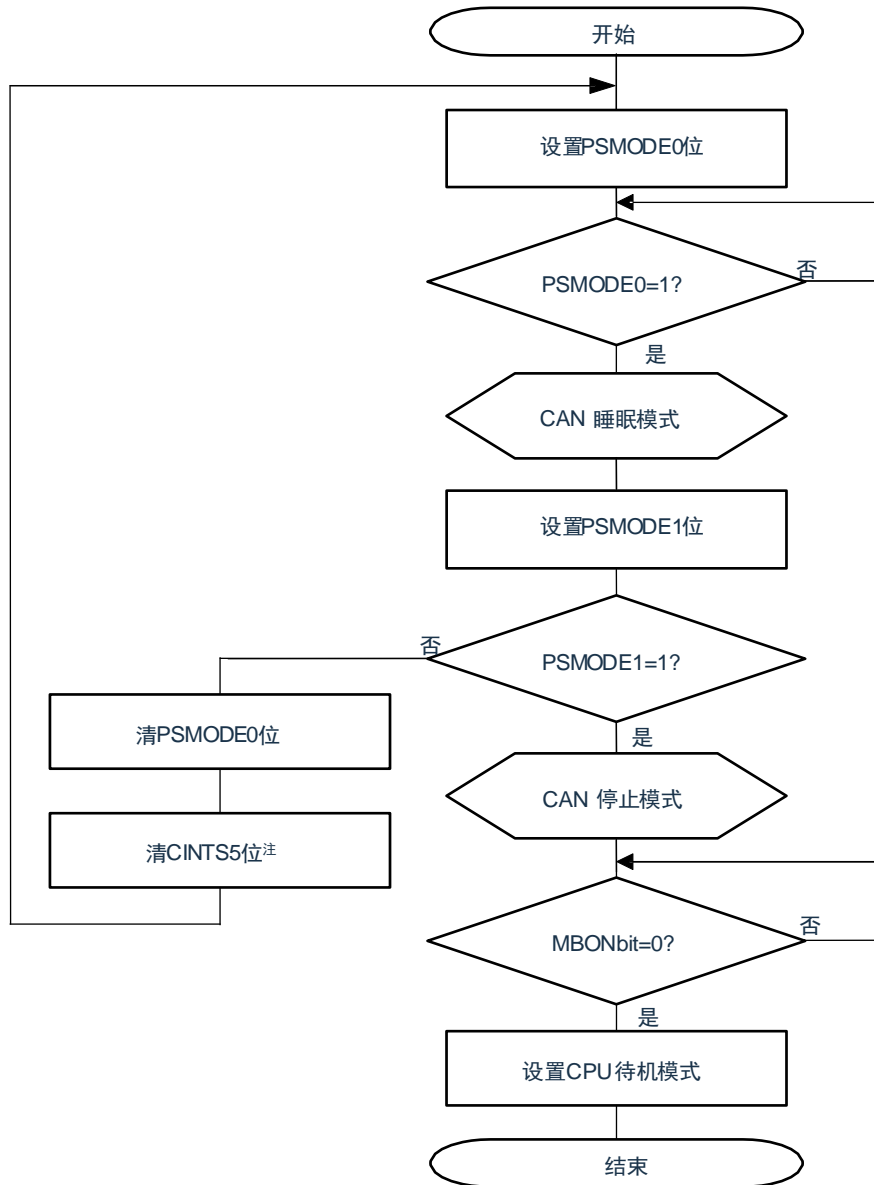


图21-87：设置CPU待机(从CAN睡眠模式)



注意：在CPU被设置进入CPU待机模式前，请检查CAN是否在睡眠模式。当检测CAN睡眠模式后，直到CPU被设置到待机模式，CAN睡眠模式可能会被CAN总线的唤醒而取消。

图21-88：设置CPU待机(从CAN停止模式)



注：在中断唤醒中

注意：CAN 停止模式只能通过将 01B 写入到C0CTRL寄存器的PSMODE[1: 0] 位，而不能通过CAN总线状态的更改来释放。

第22章 IrDA

IrDA通过与通用串行通信单元(SCI)合作, 实现发送和接收符合IrDA(InfraredDataAssociation(红外线数据协会))1.0协议的IrDA通信波形。

22.1 IrDA的功能

如果通过IRCR寄存器的IRE位将IrDA功能置为有效, SCI的TxD2信号和RxD2信号就能对符合IrDA1.0协议的波形进行编码或者解码(IrTxD/IrRxD引脚), 之后通过连接红外线发送/接收的发送器或者接收器, 实现支持IrDA1.0协议的红外线发送和接收。

在支持IrDA1.0协议的系统中, 以9600bps的传送速率开始通信后, 可根据需要改变传送速率。IrDA没有内置自动改变传送速率的功能, 所以必须通过软件更改设置以改变传送速率。

在选择高速内部振荡器($F_{IH}=24、12、6、3\text{MHz}$)时, 能设置以下的波特率。

- 115.2kbps、57.6kbps、38.4kbps、19.2kbps、9600bps、2400bps

IrDA与SCI的合作示意框图如图22-1所示。

图22-1: IrDA与SCI的合作示意框图

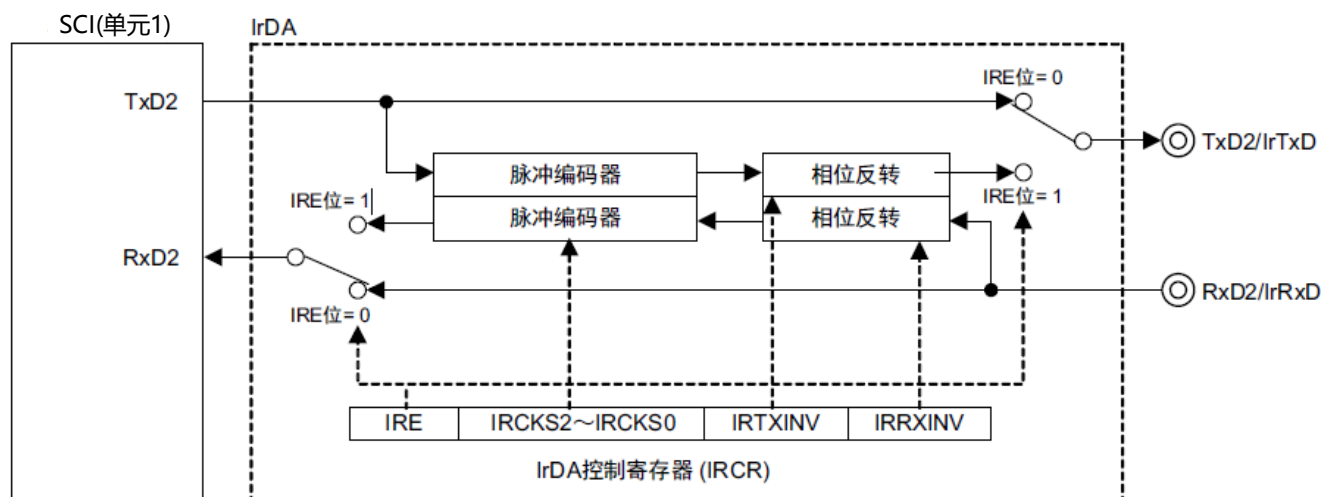


表22-1: IrDA的引脚结构

引脚名称	输入/输出	功能
IrTxD	输出	发送数据的输出引脚
IrRxD	输入	接收数据的输入引脚

22.2 控制IrDA的寄存器

通过以下寄存器控制IrDA功能。

- 外围允许寄存器0(PER0)
- IrDA控制寄存器(IRCR)

22.2.1 外围允许寄存器0(PER0)

PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用IrDA时，必须将bit6(IRDAEN)置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图22-2：外围允许寄存器0(PER0)的格式

地址：40020420H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	CAN0EN	TM40EN

IRDAEN	IrDA的输入时钟的控制
0	停止提供输入时钟。 •不能写IrDA使用的SFR。 •IrDA处于复位状态。
1	提供输入时钟。 •能读写IrDA使用的SFR。

注意：在设置IrDA时，必须先将IRDAEN位置“1”。当IRDAEN位为“0”时，忽视IrDA的控制寄存器的写操作，而且读取值全部为初始值。

22.2.2 IrDA控制寄存器(IRCR)

这是控制IrDA功能的寄存器。进行接收数据和发送数据的极性切换、IrDA的时钟选择、以及串行输入/输出引脚功能(通常的串行功能和IrDA功能)切换的选择。通过8位存储器操作指令设置IRCR寄存器。在产生复位信号后,此寄存器的值变为“00H”。

图22-3: IrDA控制寄存器(IRCR)的格式

地址: 40044000H

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IRCR	IRE	IRCKS2	IRCKS1	IRCKS0	IRTXINV	IRRXINV	0	0

IRE	IrDA的允许
0	串行输入/输出引脚用作通常的串行功能
1	串行输入/输出引脚用作IrDA功能

IRCKS2	IRCKS1	IRCKS0	IrDA的时钟选择
0	0	0	B3/16(B=位速率)
0	0	1	F _{CLK} /2
0	1	0	F _{CLK} /4
0	1	1	F _{CLK} /8
1	0	0	F _{CLK} /16
1	0	1	F _{CLK} /32
1	1	0	F _{CLK} /64
1	1	1	禁止设置

IRTXINV	IrTxD数据的极性切换
0	对发送数据进行IrTxD输出
1	反转发送数据后进行IrTxD输出

IRRXINV	IrRxD数据的极性切换
0	将IrRxD引脚的输入数据用作接收数据
1	将反转IrRxD引脚的输入数据后的数据用作接收数据

注意:

1. 必须将bit1和bit0置“0”。
2. 只有在IRE位为“0”时,才能设置IRCKS[2:0]位、IRTXINV位和IRRXINV位。

22.3 IrDA的运行

22.3.1 IrDA通信的操作步骤

(1) IrDA通信的初始设置流程

按照以下步骤进行IrDA的初始设置。

- ① 将PER0寄存器的IRDAEN位置“1”。
- ② 设置IRCR寄存器。
- ③ 设置SCI的相关寄存器(参照UART模式的设置步骤)。

(2) IrDA通信的停止流程

- ① 通过设置端口寄存器和端口模式寄存器，设置IrDA通信停止后的IrTxD引脚状态。

备注：在通过步骤3进行IrDA复位时，IrTxD引脚有可能因切换为通常的串行接口UART的数据输出，而改变输出状态。

- 从IrTxD引脚输出低电平的情况
 - 将端口寄存器置“0”。在进行此设置后，IrTxD引脚立即固定为低电平。
 - 从IrTxD引脚输出高电平的情况
 - 将端口寄存器置“1”。通过此设置，在步骤3的IrDA复位后，IrTxD引脚立即固定为高电平。
 - 将IrTxD引脚置为Hi-Z状态的情况将端口模式寄存器置“1”。在进行此设置后，IrTxD引脚立即变为Hi-Z状态。
- ② 将STm寄存器(SCI的相关寄存器)的STm0位和STm1位置“1”(停止SCI的通道0和通道1的运行)。
 - ③ 将PER0寄存器的IRDAEN位置“0”，进行IrDA复位。

不能在上述步骤以外的情况下将STm寄存器的STm0位和STm1位置“1”，或者将IrDA的IRE位置“0”。

(3) 发送IrDA帧错误时的步骤

在IrDA通信过程中发生帧错误时，为了设置为可接收后续数据的状态，必须进行以下设置。

- ① 将SCI的STm寄存器的STm1位置“1”(停止SCI的通道1的运行)。
- ② 将SCI的SSm寄存器的SSm1位置“1”(开始SCI的通道1的运行)。

备注：m：单元号(m=0)

有关SCI的帧错误处理，请参照“第19章 通用串行通信单元”。

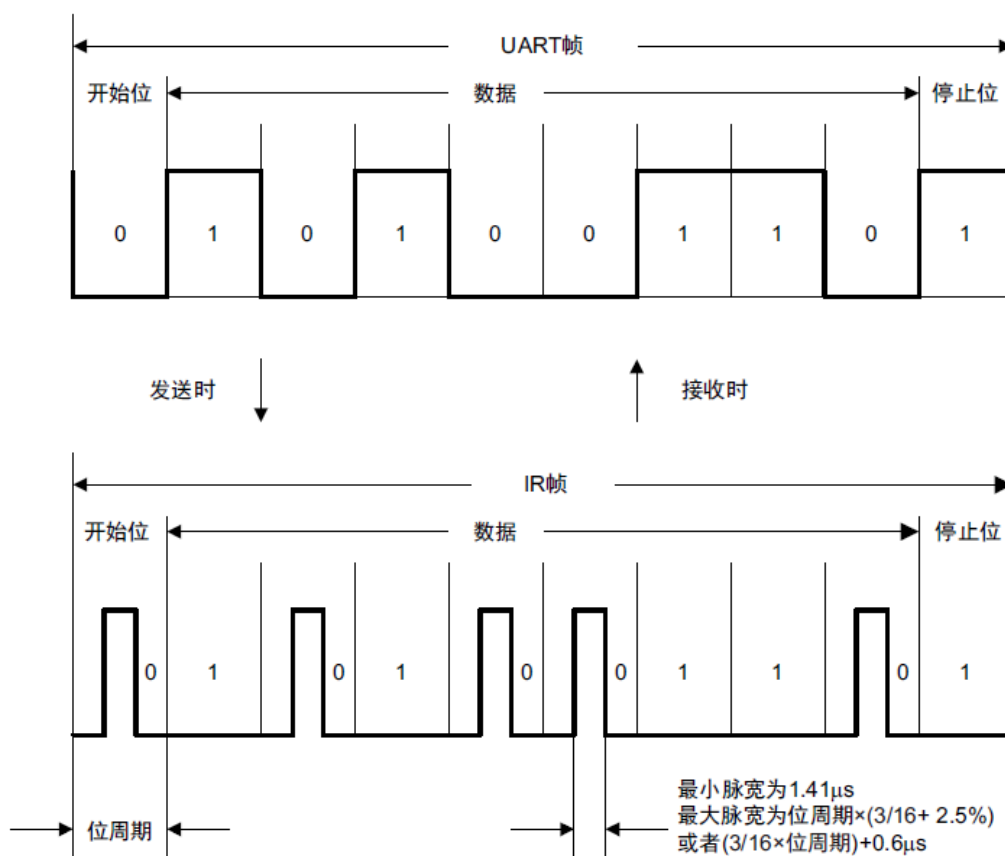
22.3.2 发送

在发送时，来自SCI的输出信号(UART帧)通过IrDA转换为IR帧(参照图22-4)。

在IRTXINV位为“0”并且串行数据为“0”时，输出位周期(1位位宽期间)×3/16的高电平脉冲(初始值)。另外，根据IRCKS2~IRCKS0位的设置值可更改高电平脉宽。按标准，规定高电平脉宽最小为1.41us，最大为(3/16+2.5%)位周期，或者(3/16×位周期)+0.6us。

在CPU或者外围硬件时钟(F_{CLK})为24MHz时，能设置的最小高电平脉宽为1.5us(满足上述规定的高电平脉宽不小于1.41us的条件)。另外，在串行数据为“1”时，不输出脉冲。

图22-4：IrDA的发送/接收运行图



22.3.3 接收

接收时，IR帧的数据通过IrDA转换为UART帧后，输入到SCI。在IRRXINV位为“0”，并且检测到高电平脉冲时，输出低电平数据。如果在1位期间内没有脉冲，就输出高电平数据。必须注意无法识别小于最小脉宽1.41us的脉冲。

22.3.4 高电平脉宽的选择

如果发送时的脉宽小于位速率 $\times 3/16$ ，可适用的IRCKS2~IRCKS0位的设置(最小脉宽)和设置时的高电平脉宽如表22-2所示。

表22-2: IRCKS2~IRCKS0位的设置值

F _{CLK} [MHz]	项目	<上段>位速率[kbps] <下段>位速率 $>3/16$ [us]					
		2.4	9.6	19.2	38.4	57.6	115.2
		78.13	19.53	9.77	4.87	3.26	1.63
1	IRCKS2~IRCKS0	001	001	001	_注1	_注1	_注1
	高电平脉宽[us]	2.00	2.00	2.00	_注1	_注1	_注1
2	IRCKS2~IRCKS0	010	010	010	010	010	_注1
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	_注1
3	IRCKS2~IRCKS0	011	011	011	011	011	_注1
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	_注1
4	IRCKS2~IRCKS0	011	011	011	011	011	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
6	IRCKS2~IRCKS0	100	100	100	100	100	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
8	IRCKS2~IRCKS0	100	100	100	100	100	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
12	IRCKS2~IRCKS0	101	101	101	101	101	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
16	IRCKS2~IRCKS0	101	101	101	101	101	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
24	IRCKS2~IRCKS0	110	110	110	110	110	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50

注1：“-”表示未达到通信标准。

注2：脉宽不能小于位速率 $\times 3/16$ 。

22.4 使用IrDA时的注意事项

1. 能通过外围允许寄存器设置允许或者禁止提供IrDA的运行时钟。初始状态为禁止提供时钟，因此无法存取寄存器。在设置寄存器前，必须通过外围允许寄存器设置为允许提供IrDA运行时钟的状态。
2. 睡眠模式中，IrDA功能持续运行。
3. 在IrDA通信过程中，禁止使用SCI的初始化功能(SS位=1)。
4. 只有在IRE位为“0”时，才能设置IRCR寄存器的IRRXINV位、IRTXINV位和IRCKS[2:0]位。

第23章 增强型DMA

23.1 DMA的功能

DMA是不使用CPU而在存储器之间进行数据传送的功能。通过外围功能中断启动DMA进行数据传送。当DMA和CPU同时访问FLASH, SRAM0, SRAM1或外围模块中的同一单元时, 其总线使用权高于CPU。当DMA和CPU分别访问FLASH, SRAM0, SRAM1或外围模块中的不同单元时, 两者互不干扰, 可以并行执行。

DMA的规格如表23-1所示。

表23-1: DMA的规格(1/2)

项目		规格
启动源		最大37个启动源
可分配的控制数据		40组
可传送的地址空间	地址空间	全地址范围空间
	源	全地址范围空间可选
	目标	全地址范围空间可选
最大传送次数	正常模式	65535次
	重复模式	65535次
最大传送块大小	正常模式 (8位传送)	65535字节
	正常模式 (16位传送)	131070字节
	正常模式 (32位传送)	262140字节
	重复模式	65535字节
传送单位		8位/16位/32位
传送模式	正常模式	在进行DMACTj寄存器从“1”变为“0”的传送后结束。
	重复模式	在DMACTj寄存器从“1”变为“0”的传送结束后, 对重复区的地址进行初始化, 在将DMRLDj寄存器的值重新加载到DMACTj寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表23-5: DMA启动源和向量地址”。

表23-1: DMA的规格(2/2)

项目		规格
中断请求	正常模式	在进行DMACTj寄存器从“1”变为“0”的数据传送时, 向CPU请求启动源的中断, 并进行中断处理。
	重复模式	在DMACRj寄存器的RPTINT位为“1”(允许产生中断)并且进行DMACTj寄存器从“1”变为“0”的数据传送时, 向CPU请求启动源的中断, 并进行中断处理。
传送开始		如果将DMAENi寄存器的DMAENi0~DMAENi7位置“1”(允许启动), 就在每次发生DMA启动源时开始传送数据。
传送停止	正常模式	<ul style="list-style-type: none"> •将DMAENi0~DMAENi7位置“0”(禁止启动)。 •当DMACTj寄存器从“1”变为“0”的数据传送结束时
	重复模式	<ul style="list-style-type: none"> •将DMAENi0~DMAENi7位置“0”(禁止启动)。 •当RPTINT位为“1”(允许产生中断)并且DMACTj寄存器从“1”变为“0”的数据传送结束时

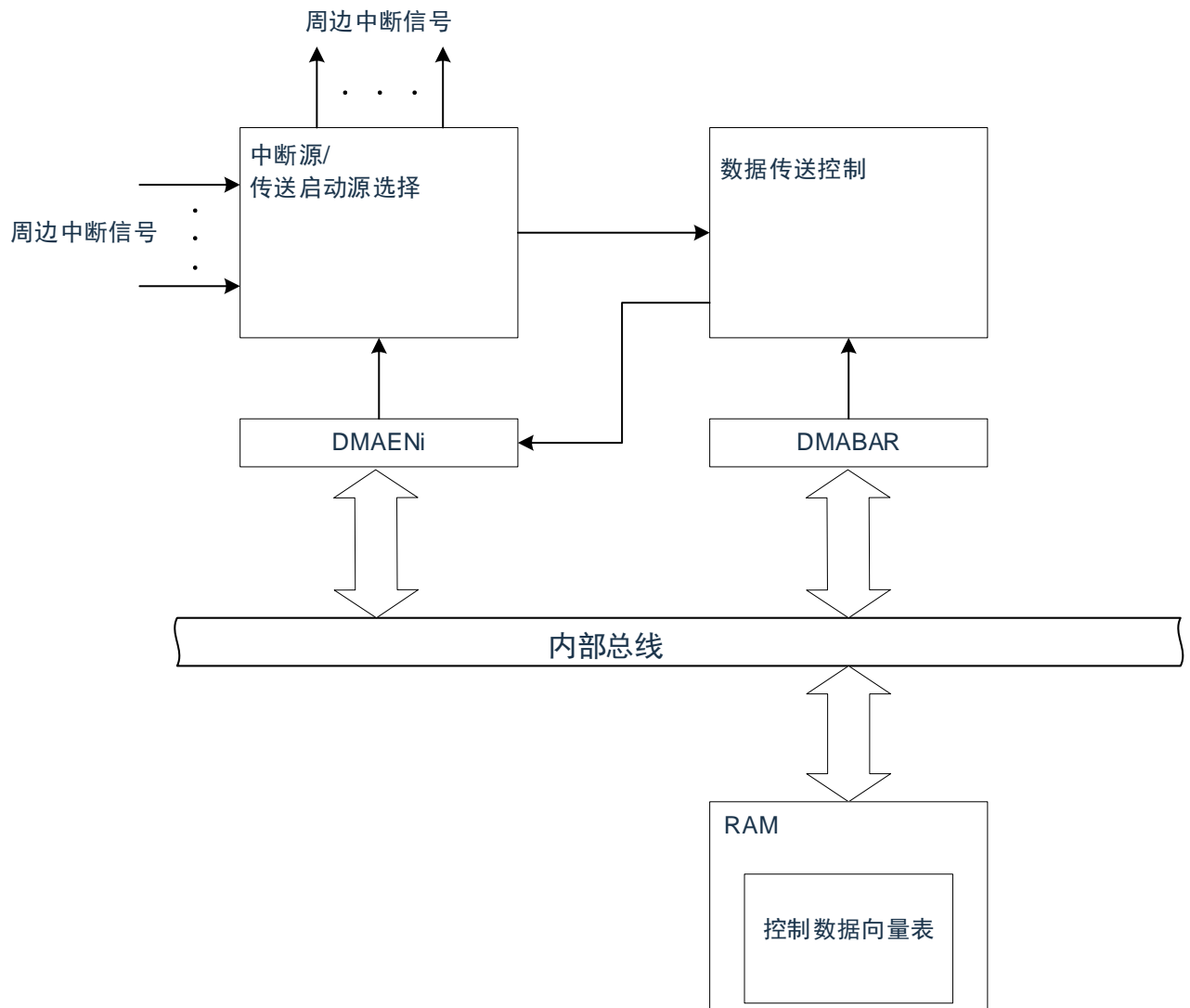
注: 在深度睡眠模式, 因为闪存停止运行, 所以不能作为DMA传送源。

备注: i=0~4、j=0~39

23.2 DMA的结构

DMA的框图如图23-1所示

图23-1: DMA的框图



23.3 控制DMA的寄存器

控制DMA的寄存器如表23-2所示。

表23-2: 控制DMA的寄存器

寄存器名	符号
外围允许寄存器1	PER1
DMA启动允许寄存器0	DMAEN0
DMA启动允许寄存器1	DMAEN1
DMA启动允许寄存器2	DMAEN2
DMA启动允许寄存器3	DMAEN3
DMA启动允许寄存器4	DMAEN4
DMA基址寄存器	DMABAR

DMA的控制数据如表23-3所示。

DMA的控制数据分配在RAM的DMA控制数据区。通过DMABAR寄存器设置DMA控制数据区和包含DMA向量表区(保存控制数据的起始地址)的704字节区域。

表23-3: DMA的控制数据

寄存器名	符号
DMA控制寄存器j	DMACRj
DMA块大小寄存器j	DMBLSj
DMA传送次数寄存器j	DMACTj
DMA传送次数重加载寄存器j	DMRLDj
DMA源地址寄存器j	DMSARj
DMA目标地址寄存器j	DMDARj

备注: j=0~39

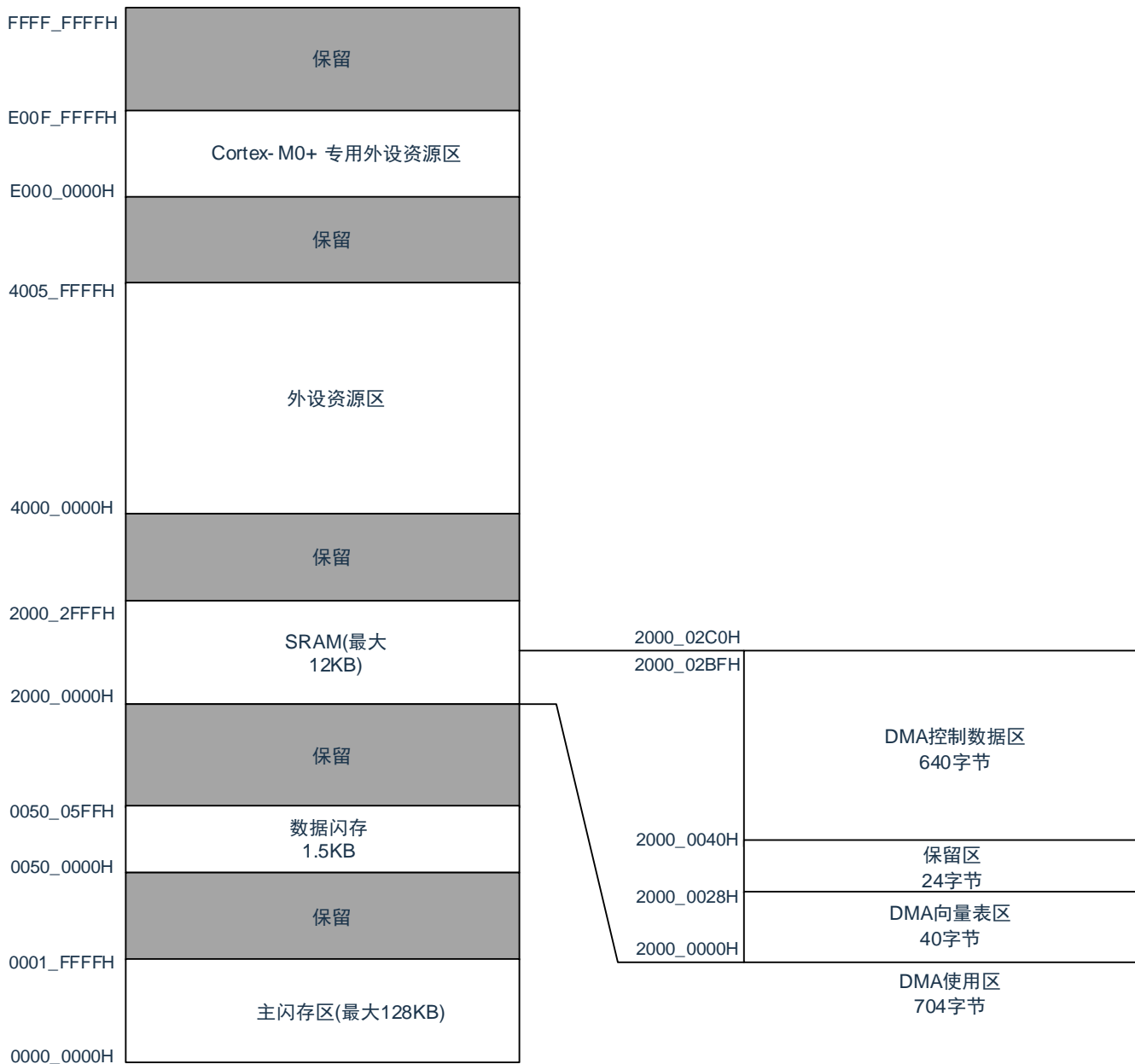
23.3.1 DMA控制数据区和DMA向量表区的分配

通过DMABAR寄存器将分配DMA的控制数据和向量表的704字节区域设置到RAM区。

DMABAR寄存器的设置值为“20000000H”时的存储器映像例子如图23-2所示。

DMA控制数据区的640字节中DMA不使用的空间能用作RAM。

图23-2: DMABAR寄存器的设置值为“20000000H”时的存储器映像例子



23.3.2 控制数据的分配

从起始地址开始，按照DMACR_j、DMBLS_j、DMACT_j、DMRLD_j、DMSAR_j、DMDAR_j(j=0~39)寄存器的顺序分配控制数据。

起始地址由DMABAR寄存器设置，低10位由各启动源分配的向量表分别设置。

控制数据的分配如图23-3所示。

注意：

1. 必须在对应的DMAEN_i(i=0~4)的DMAEN_{i0}~DMAEN_{i7}位为“0”(禁止启动)时更改DMACR_j、DMBLS_j、DMACT_j、DMRLD_j、DMSAR_j、DMDAR_j寄存器的数据。
2. 不能通过DMA传送进行DMACR_j、DMBLS_j、DMACT_j、DMRLD_j、DMSAR_j和DMDAR_j的存取。

图23-3：控制数据的分配(DMABAR设置为2000000H)

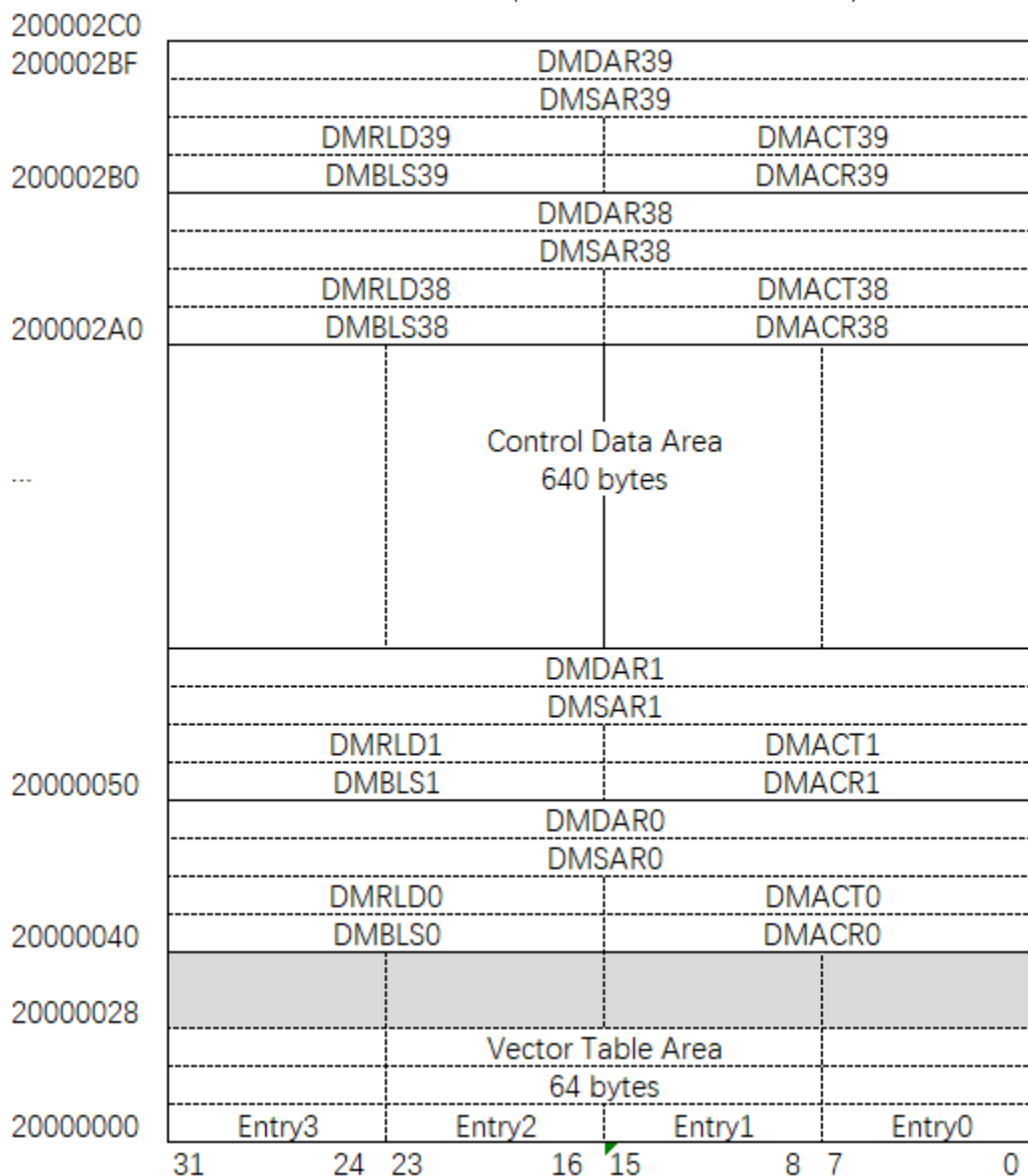


表23-4：控制数据的起始地址

j	地址	j	地址
19	baseaddr+170H	39	baseaddr+2B0H
18	baseaddr+160H	38	baseaddr+2A0H
17	baseaddr+150H	37	baseaddr+290H
16	baseaddr+140H	36	baseaddr+280H
15	baseaddr+130H	35	baseaddr+270H
14	baseaddr+120H	34	baseaddr+260H
13	baseaddr+110H	33	baseaddr+250H
12	baseaddr+100H	32	baseaddr+240H
11	baseaddr+F0H	31	baseaddr+230H
10	baseaddr+E0H	30	baseaddr+220H
9	baseaddr+D0H	29	baseaddr+210H
8	baseaddr+C0H	28	baseaddr+200H
7	baseaddr+B0H	27	baseaddr+1F0H
6	baseaddr+A0H	26	baseaddr+1E0H
5	baseaddr+90H	25	baseaddr+1D0H
4	baseaddr+80H	24	baseaddr+1C0H
3	baseaddr+70H	23	baseaddr+1B0H
2	baseaddr+60H	22	baseaddr+1A0H
1	baseaddr+50H	21	baseaddr+190H
0	baseaddr+40H	20	baseaddr+180H

备注：baseaddr：DMABAR寄存器的设置值

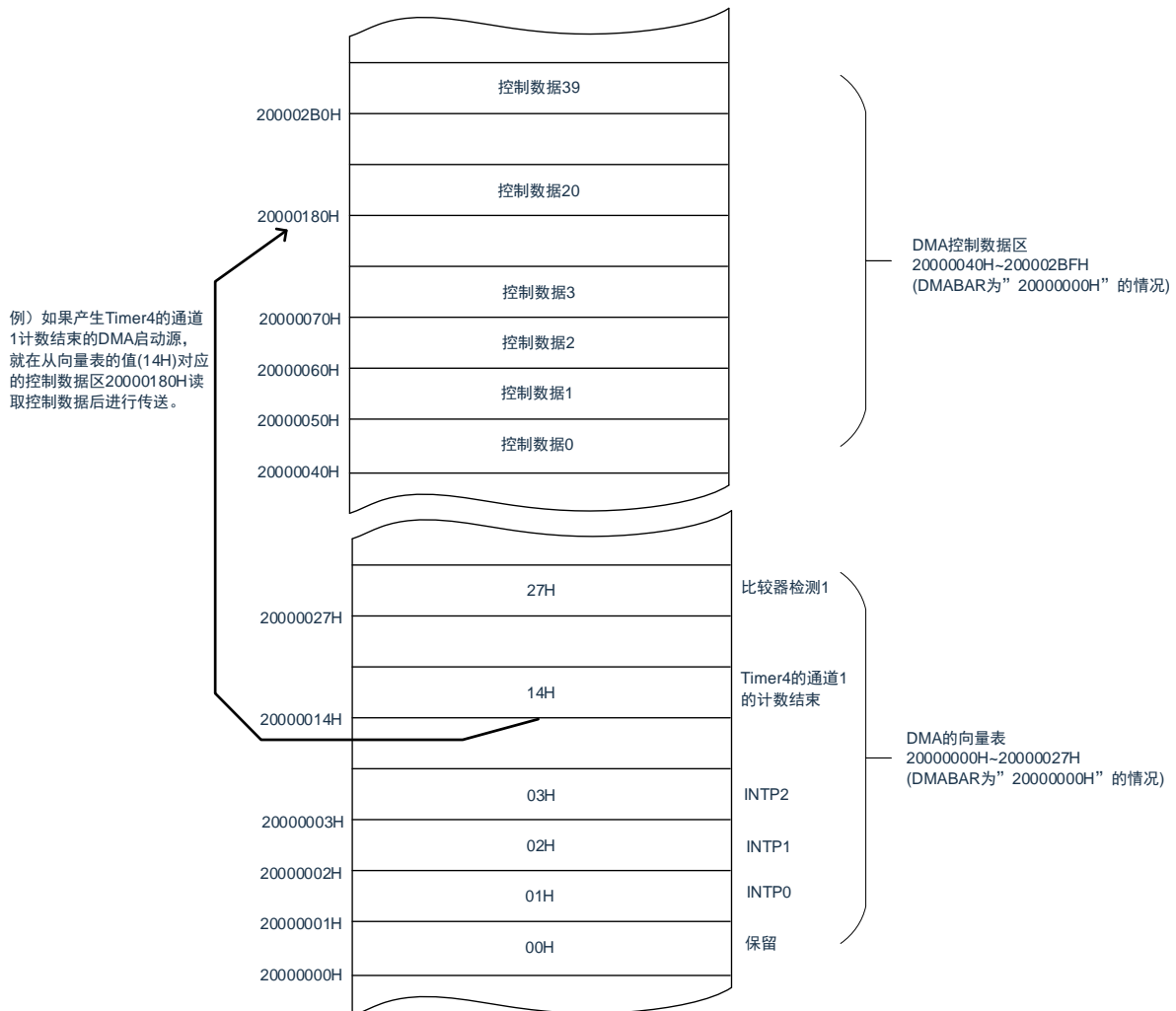
23.3.3 向量表

DMA一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在DMA控制数据区的控制数据。

DMA启动源和向量地址如表23-5所示。各启动源的向量表有1字节，保存“00H”~“27H”的数据，从40组的控制数据中选择1组数据。向量地址的高22位由DMABAR寄存器设置，低10位被分配了对应启动源的“00H”~“27H”。

注意：必须在对应的DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位为“0”(禁止启动)时更改设置在向量表中的DMA控制数据区的起始地址。

图23-4：控制数据的起始地址和向量表
DMABAR寄存器的设置值为“2000000H”的情况(例)



23.3.4 外围允许寄存器1(PER1)

PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用DMA时，必须将bit3(DMAEN)置“1”。

通过8位存储器操作指令设置PER1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图23-5：外围允许寄存器1(PER1)的格式

地址：0x4002081A 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER1	DACEN	TMBEN	PGACMPEN	TMMEN	DMAEN	PWMOPEN	TMCEN	TMAEN

DMAEN	提供DMA的输入时钟的控制
0	停止提供输入时钟。 •DMA不能运行。
1	提供输入时钟。 •DMA能运行。

23.3.5 DMA控制寄存器j(DMACRj)(j=0~39)

DMACRj寄存器控制DMA的运行模式。

图23-6: DMA控制寄存器j(DMACRj)的格式

地址: 参照“23.3.2控制数据的分配”。

复位后: 不定值 R/W

符号	15	14	13	12	11	10	9	8
DMACRj	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SZ		RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	传送数据长度的选择
00	8位
01	16位
10	32位
11	禁止设置

RPTINT	重复模式中中断的允许/禁止
0	禁止发生中断。
1	允许发生中断。

在MODE位为“0”(正常模式)时, RPTINT位的设置无效。

CHNE	链传送的允许/禁止
0	禁止链传送。
1	允许链传送。

必须将DMACR39寄存器的CHNE位置“0”(禁止链传送)。

DAMOD	传送目标地址的控制
0	固定
1	递增

在MODE位为“1”(重复模式)并且RPTSEL位为“0”(传送目标为重复区)时, DAMOD位的设置无效。

SAMOD	传送源地址的控制
0	固定
1	递增

在MODE位为“1”(重复模式)并且RPTSEL位为“1”(传送源为重复区)时, SAMOD位的设置无效。

RPTSEL	重复区的选择
0	传送目标为重复区。
1	传送源为重复区。

在MODE位为“0”(正常模式)时, RPTSEL位的设置无效。

MODE	传送模式的选择
0	正常模式
1	重复模式

注意: 不能通过DMA传送进行DMACRj寄存器的存取。

23.3.6 DMA块大小寄存器j(DMBLSj)(j=0~39)

此寄存器设置1次启动传送数据的块大小。

图23-7：DMA块大小寄存器j(DMBLSj)的格式

地址：参照“23.3.2控制数据的分配”。 复位后：不定值 R/W

符号	15	14	13	12	11	10	9	8
DMBLSj	DMBLSj15	DMBLSj14	DMBLSj13	DMBLSj12	DMBLSj11	DMBLSj10	DMBLSj9	DMBLSj8
	7	6	5	4	3	2	1	0
	DMBLSj7	DMBLSj6	DMBLSj5	DMBLSj4	DMBLSj3	DMBLSj2	DMBLSj1	DMBLSj0

DMBLSj	传送块大小		
	8位传送	16位传送	32位传送
00H	禁止设置	禁止设置	禁止设置
01H	1字节	2字节	4字节
02H	2字节	4字节	8字节
03H	3字节	6字节	12字节
•	•	•	•
•	•	•	•
•	•	•	•
FDH	253字节	506字节	1012字节
FEH	254字节	508字节	1016字节
FFH	255字节	510字节	1020字节
•	•	•	•
•	•	•	•
•	•	•	•
FFFFH	65535字节	131070字节	262140字节

注意：不能通过DMA传送进行DMBLSj寄存器的存取。

23.3.7 DMA传送次数寄存器j(DMACTj)(j=0~39)

此寄存器设置DMA的数据传送次数。每当启动1次DMA传送就减1。

图23-8: DMA传送次数寄存器j(DMACTj)的格式

地址: 参照“23.3.2控制数据的分配”。 复位后: 不定值 R/W

符号	15	14	13	12	11	10	9	8
DMACTj	DMACTj15	DMACTj14	DMACTj13	DMACTj12	DMACTj11	DMACTj10	DMACTj9	DMACTj8
	7	6	5	4	3	2	1	0
	DMACTj7	DMACTj6	DMACTj5	DMACTj4	DMACTj3	DMACTj2	DMACTj1	DMACTj0

DMACTj	传送次数
00H	禁止设置
01H	1次
02H	2次
03H	3次
⋮	⋮
FDH	253次
FEH	254次
FFH	255次
⋮	⋮
FFFFH	65535次

注意: 不能通过DMA传送进行DMACTj寄存器的存取。

23.3.8 DMA传送次数重加载寄存器j(DMRLDj)(j=0~39)

此寄存器设置重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到DMACT寄存器，所以设置值必须和DMACT寄存器的初始值相同。

图23-9: DMA传送次数重加载寄存器j(DMRLDj)的格式

地址：参照“23.3.2控制数据的分配”。 复位后：不定值 R/W

符号	15	14	13	12	11	10	9	8
DMRLDj	DMRLDj15	DMRLDj14	DMRLDj13	DMRLDj12	DMRLDj11	DMRLDj10	DMRLDj9	DMRLDj8
	7	6	5	4	3	2	1	0
	DMRLDj7	DMRLDj6	DMRLDj5	DMRLDj4	DMRLDj3	DMRLDj2	DMRLDj1	DMRLDj0

注意：不能通过DMA传送进行DMRLDj寄存器的存取。

23.3.9 DMA源地址寄存器j(DMSARj)(j=0~39)

此寄存器指定数据传送时的传送源地址。

当DMACRj寄存器的SZ位为“01”(16位传送)时, 忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”(32位传送)时, 忽视低2位而作为word地址进行处理。

图23-10: DMA源地址寄存器j(DMSARj)的格式

地址: 参照“23.3.2控制数据的分配”。 复位后: 不定值 R/W

符号	31	30	29	28	27	26	25	24
DMSARj	DMSARj31	DMSARj30	DMSARj29	DMSARj28	DMSARj27	DMSARj26	DMSARj25	DMSARj24
	23	22	21	20	19	18	17	16
	DMSARj23	DMSARj22	DMSARj21	DMSARj20	DMSARj19	DMSARj18	DMSARj17	DMSARj16
	15	14	13	12	11	10	9	8
	DMSARj15	DMSARj14	DMSARj13	DMSARj12	DMSARj11	DMSARj10	DMSARj9	DMSARj8
	7	6	5	4	3	2	1	0
	DMSARj7	DMSARj6	DMSARj5	DMSARj4	DMSARj3	DMSARj2	DMSARj1	DMSARj0

注意: 不能通过DMA传送进行DMSARj寄存器的存取。

23.3.10 DMA目标地址寄存器j(DMDARj)(j=0~39)

此寄存器指定数据传送时的传送目标地址。

当DMACRj寄存器的SZ位为“01”(16位传送)时, 忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”(32位传送)时, 忽视低2位而作为word地址进行处理。

图23-11: DMA目标地址寄存器j(DMDARj)的格式

地址: 参照“23.3.2控制数据的分配”。 复位后: 不定值 R/W

符号	31	30	29	28	27	26	25	24
DMDARj	DMDARj31	DMDARj30	DMDARj29	DMDARj28	DMDARj27	DMDARj26	DMDARj25	DMDARj24
	23	22	21	20	19	18	17	16
	DMDARj23	DMDARj22	DMDARj21	DMDARj20	DMDARj19	DMDARj18	DMDARj17	DMDARj16
	15	14	13	12	11	10	9	8
	DMDARj15	DMDARj14	DMDARj13	DMDARj12	DMDARj11	DMDARj10	DMDARj9	DMDARj8
	7	6	5	4	3	2	1	0
	DMDARj7	DMDARj6	DMDARj5	DMDARj4	DMDARj3	DMDARj2	DMDARj1	DMDARj0

注意: 不能通过DMA传送进行DMDARj寄存器的存取。

23.3.11 DMA启动允许寄存器i(DMAENi)(i=0~4)

这是控制允许或者禁止通过各中断源启动DMA的8位寄存器。中断源和DMAENi0~DMAENi7位的对应如表23-6所示。

能通过8位存储器操作指令设置DMAENi寄存器。

注意：

1. 必须在不产生对应该位的启动源的位置更改DMAENi0~DMAENi7位。
2. 不能通过DMA传送进行DMAENi寄存器的存取。
2. 分配的功能因产品而不同，必须将没有分配功能的位置“0”。

图23-12：DMA启动允许寄存器i(DMAENi)(i=0~4)的格式

地址：40005000H(DMAEN0)、40005001H(DMAEN1)、
40005002H(DMAEN2)、40005003H(DMAEN3)、
40005004H(DMAEN4)

						复位后：00H	R/W	
符号	7	6	5	4	3	2	1	0
DMAENi	DMAENj7	DMAENj6	DMAENj5	DMAENj4	DMAENj3	DMAENj2	DMAENj1	DMAENj0

DMAENi7	DMA启动的允许i7
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi7位变为“0”(禁止启动)。	

DMAENi6	DMA启动的允许i6
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi6位变为“0”(禁止启动)。	

DMAENi5	DMA启动的允许i5
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi5位变为“0”(禁止启动)。	

DMAENi4	DMA启动的允许i4
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi4位变为“0”(禁止启动)。	

DMAENi3	DMA启动的允许i3
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi3位变为“0”(禁止启动)。	

DMAENi2	DMA启动的允许i2
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi2位变为“0”(禁止启动)。	

DMAENi1	DMA启动的允许i1
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi1位变为“0”(禁止启动)。	

DMAENi0	DMA启动的允许i0
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi0位变为“0”(禁止启动)。	

表23-6：中断源和DMAENi0~DMAENi7位的对应

寄存器	DMAENi7位	DMAENi6位	DMAENi5位	DMAENi4位	DMAENi3位	DMAENi2位	DMAENi1位	DMAENi0位
DMAEN0	INTP6	INTP5	INTP4	INTP3	INTP2	INTP1	INTP0	保留
DMAEN1	UART2接收的传送结束/CSI21的传送结束或者缓冲器空/IIC21的传送结束	UART1发送的传送结束/CSI10的传送结束或者缓冲器空/IIC10的传送结束	UART1接收的传送结束/CSI11的传送结束或者缓冲器空/IIC11的传送结束	UART0发送的传送结束/CSI00的传送结束或者缓冲器空/IIC00的传送结束	UART0接收的传送结束/CSI01的传送结束或者缓冲器空/IIC01的传送结束	A/D转换结束	KEY输入	INTP7
DMAEN2	15比特时间定时器中断	定时器阵列单元0的通道3的计数结束或者捕捉结束	定时器阵列单元0的通道2的计数结束或者捕捉结束	定时器阵列单元0的通道1的计数结束或者捕捉结束	定时器阵列单元0的通道0的计数结束或者捕捉结束	保留	IICA0通信结束	UART2发送的传送结束/CSI20的传送结束或者缓冲器空/IIC20的传送结束
DMAEN3	TimerM的比较匹配A1	TimerM的比较匹配D0	TimerM的比较匹配C0	TimerM的比较匹配B0	TimerM的比较匹配A0	TimerC的上溢	保留	Flash擦除/写入结束
DMAEN4	比较器检测1	比较器检测0	TimerA的上溢	TimerB的比较匹配B	TimerB的比较匹配A	TimerM的比较匹配D1	TimerM的比较匹配C1	TimerM的比较匹配B1

注意：必须将没有分配功能的位置“0”。

备注：i=0~4

23.3.12 DMA基址寄存器(DMABAR)

这是32位寄存器，设置保存DMA控制数据区起始地址的向量地址以及DMA控制数据区的地址。

注意：

1. 必须在将全部的DMA启动源设置为禁止启动的状态下更改DMABAR寄存器。
2. 只能改写1次DMABAR寄存器。
2. 不能通过DMA传送进行DMABAR寄存器的存取。
3. 有关DMA控制数据区和DMA向量表区的分配，请参照“23.3.1DMA控制数据区和DMA向量表区的分配”的注意。
4. 设置该寄存器请保持1024Byte对齐，也就是低10位设置为零。DMA硬件忽略低10位。
5. 该寄存器只能WORD访问，BYTE和HALFWORD访问忽略。

图23-13: DMA基址寄存器(DMABAR)的格式

地址: 40005008H 复位后: 00000000H R/W

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

23.4 DMA的运行

DMA一旦启动，就从DMA控制数据区读控制数据，根据此控制数据进行数据传送，并且将数据传送后的控制数据回写到DMA控制数据区。能将40组控制数据保存到DMA控制数据区，并且进行40组数据的传送。传送模式有正常模式和重复模式，传送大小有8位传送，16位传送和32位传送。在DMACRj(j=0~39)寄存器的CHNE位为“1”(允许链传送)时，通过1个启动源读多个控制数据进行连续的数据传送(链传送)。

通过32位DMSARj寄存器和32位DMDARj寄存器分别指定传送源地址和传送目标地址。在数据传送后，根据控制数据递增或者固定DMSARj寄存器和DMDARj寄存器的值。

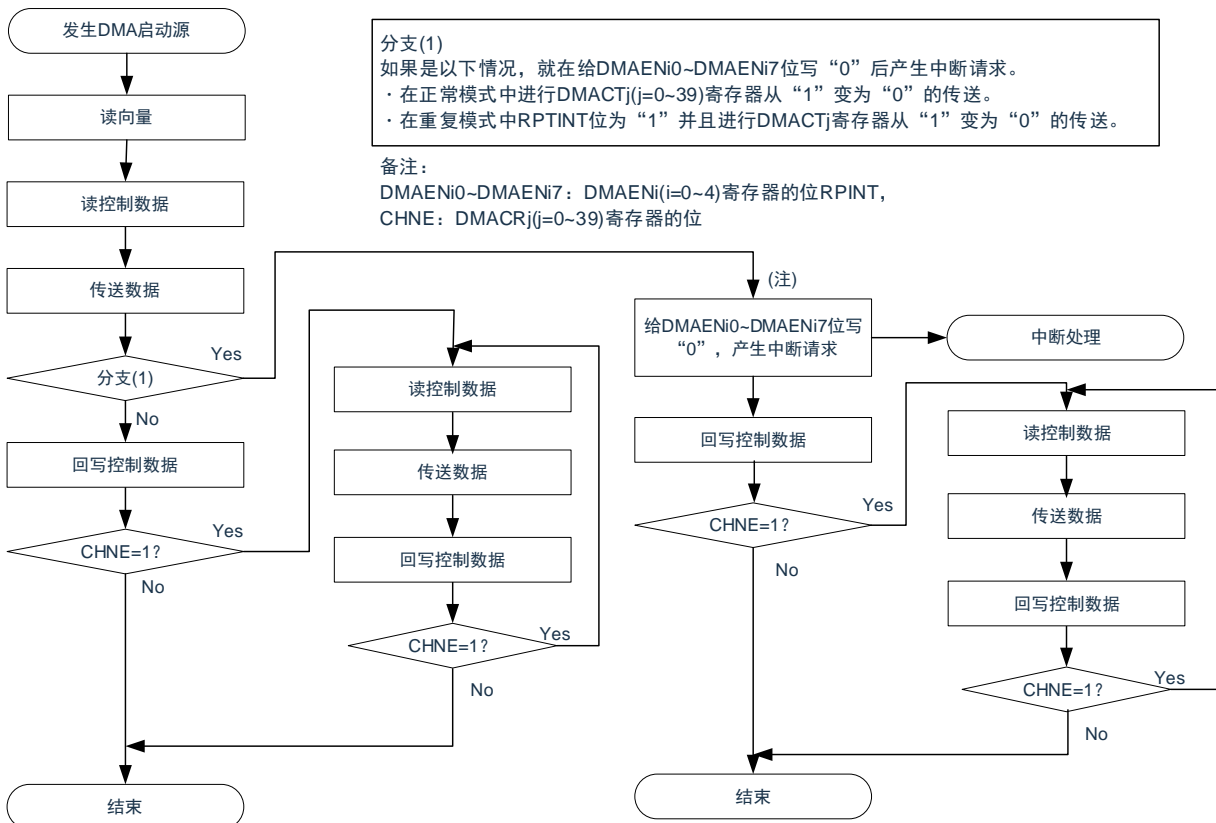
23.4.1 启动源

通过外围功能的中断信号启动DMA，并且通过DMAENi(i=0~4)寄存器选择启动DMA的中断信号。当数据传送(在进行链传送时，连续进行最初的传送)的设置如下述两种情况时，就在DMA运行中将对应的DMAENi寄存器的DMAENi0~DMAENi7位置“0”(禁止启动)。

- 在正常模式中，进行DMACTj(j=0~39)寄存器变为“0”的传送。
- 在重复模式中，DMACRj寄存器的RPTINT位为“1”(允许发生中断)并且进行DMACTj寄存器变为“0”的传送。

DMA的内部运行流程图如图23-14所示。

图23-14：DMA的内部运行流程图



注：在通过允许链传送(CHNE=1)的设置启动的数据传送中，不给DMAENi0~DMAENi7位写“0”并且不产生中断请求。

23.4.2 正常模式

在8位传送时，1次启动的传送数据为1~65535字节；在16位传送时，1次启动的传送数据为2~131070字节；在32位传送时，1次启动的传送数据为4~262140字节。传送次数为1~65535次。如果进行DMACTj(j=0~39)寄存器变为“0”的数据传送，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位置“0”(禁止启动)。

正常模式的寄存器功能和数据传送分别如表23-7和图23-15所示。

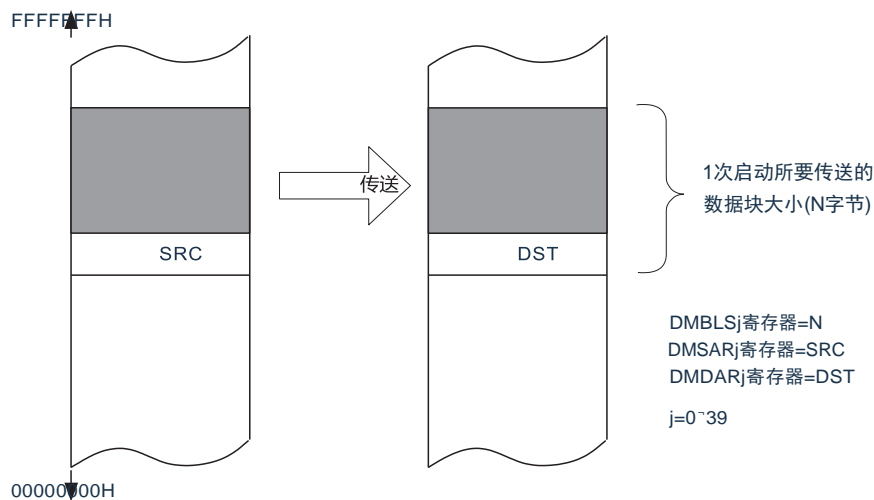
表23-7：正常模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	不使用 ^注 。
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

注：当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位(RPERDIS=0)时，必须进行初始化(00H)。

备注：j=0~39

图23-15：正常模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	固定	固定	SRC	DST
0	1	X	0	递增	固定	SRC+N	DST
1	0	X	0	固定	递增	SRC	DST+N
1	1	X	0	递增	递增	SRC+N	DST+N

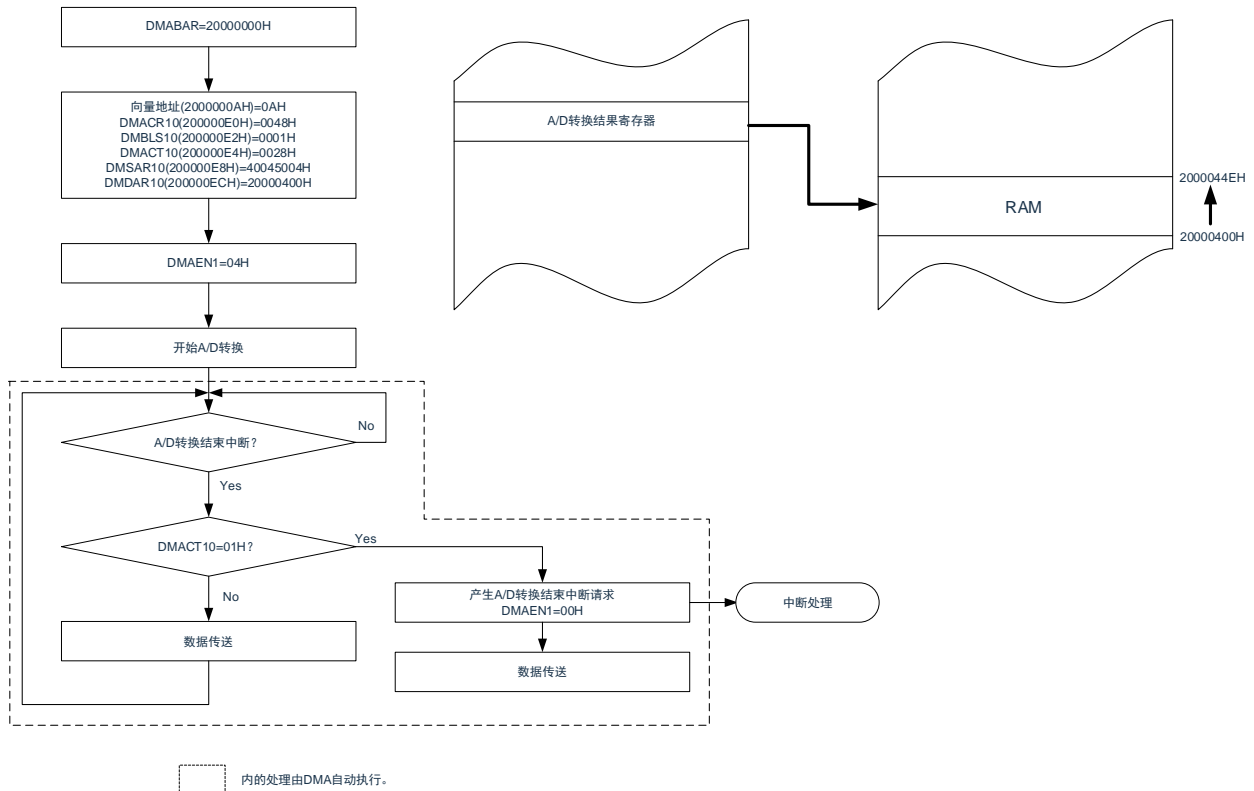
备注：X：“0”或者“1”

(1) 正常模式的使用例子1：连续取A/D转换结果

通过A/D转换结束中断启动DMA，并且将A/D转换结果寄存器的值传送到RAM。

- 向量地址分配在2000000AH，控制数据分配在200000E0H~200000EFH。
- 将A/D转换结果寄存器(40045004H, 40045005H)的2字节数据传送40次到RAM的20000400H~2000044FH的80字节。

图23-16：正常模式的使用例子1：连续取A/D转换结果



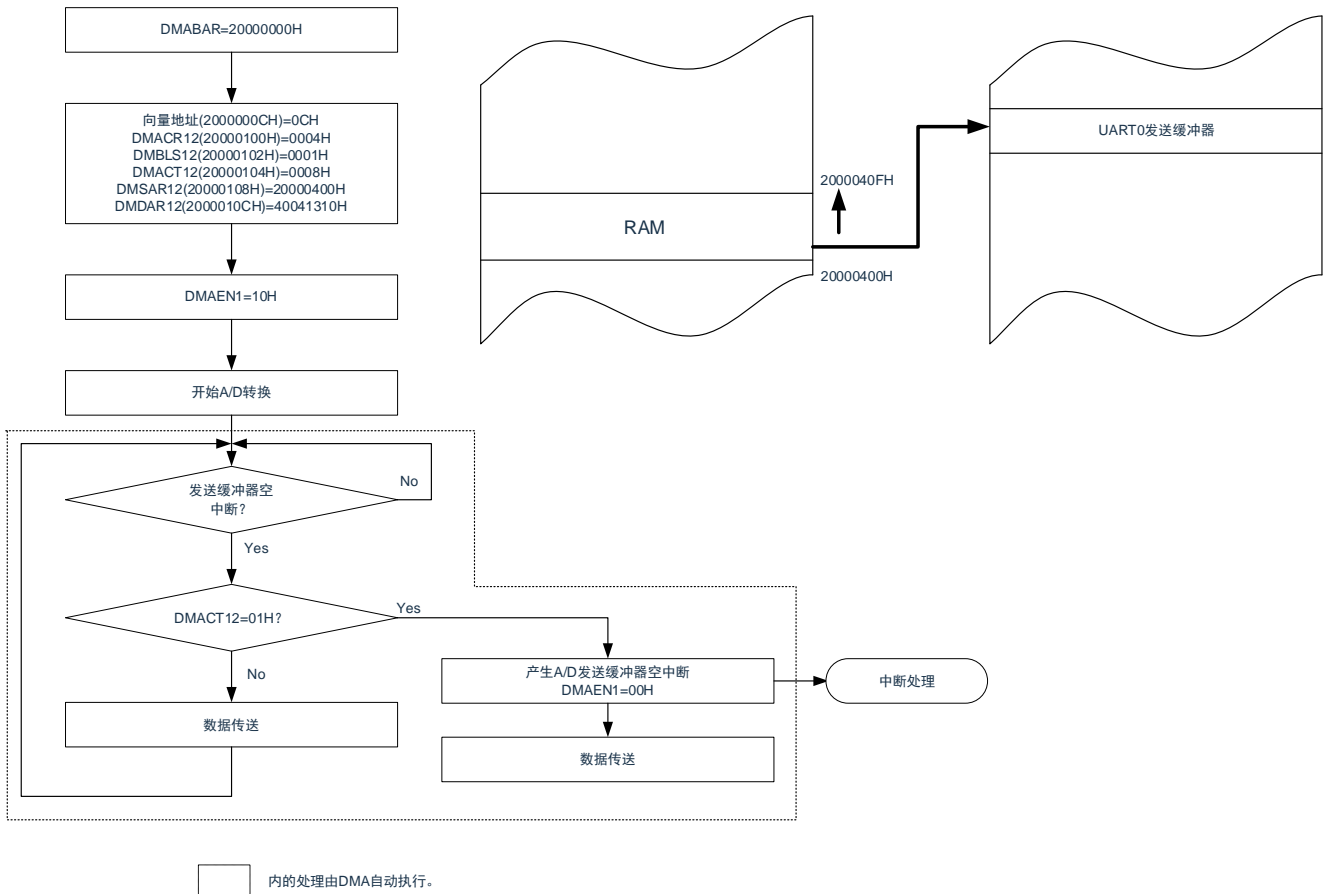
因为是正常模式，所以不使用DMRLD10寄存器的值。但是，当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位(RPERDIS=0)时，必须对DMRLD10寄存器进行初始化(0000H)。

(2) 正常模式的使用例子2: UART0连续发送

通过UART0的发送缓冲器空中断启动DMA, 并且将RAM的值传送到UART0的发送缓冲器。

- 向量地址分配在2000000CH, 控制数据分配在20000100H~2000010FH。
- 将RAM的20000400H~20000407H的8字节传送到UART0的发送缓冲器(40041310H)。

图23-17: 正常模式的使用例子2: UART0连续发送



因为是正常模式, 所以不使用DMRLD12寄存器的值。但是, 当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位(RPERDIS=0)时, 必须对DMRLD12寄存器进行初始化(0000H)。

必须通过软件开始第1次的UART0发送。通过发送缓冲器空中断启动DMA, 然后自动进行第2次以后的发送。

23.4.3 重复模式

1次启动的传送数据为1~65535字节。将传送源或者传送目标指定为重复区，传送次数为1~65535次。一旦指定次数的传送结束，就对DMACTj(j=0~39)寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当DMACRj寄存器的RPTINT位为“1”(允许发生中断)并且进行DMACTj寄存器变为“0”的数据传送时，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位置“0”(禁止启动)。当DMACRj寄存器的RPTINT位为“0”(禁止发生中断)时，即使进行DMACTj寄存器变为“0”的数据传送，也不产生中断请求，而且DMAENi0~DMAENi7位不变为“0”。

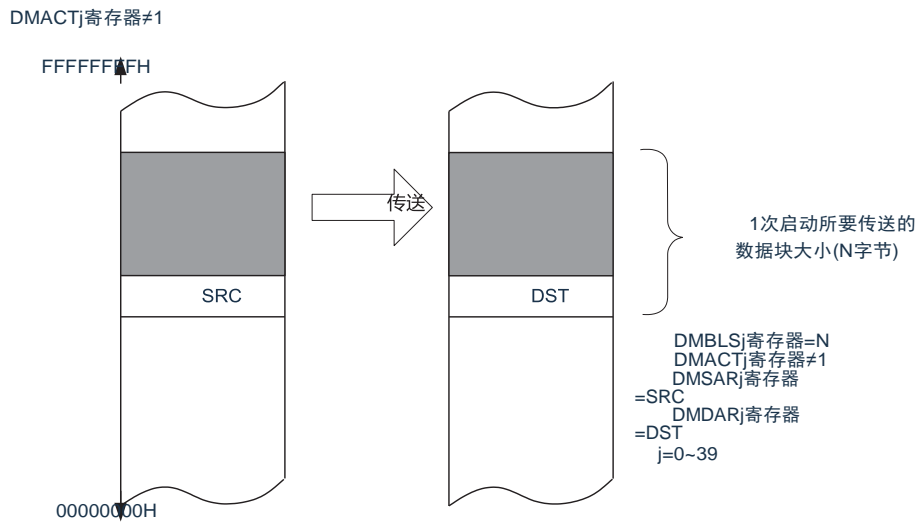
重复模式的寄存器功能和数据传送分别如表23-8和图23-18所示。

表23-8：重复模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	将此寄存器的值重新加载到DMACTj寄存器。 (对数据的传送次数进行初始化)
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

备注：j=0~39

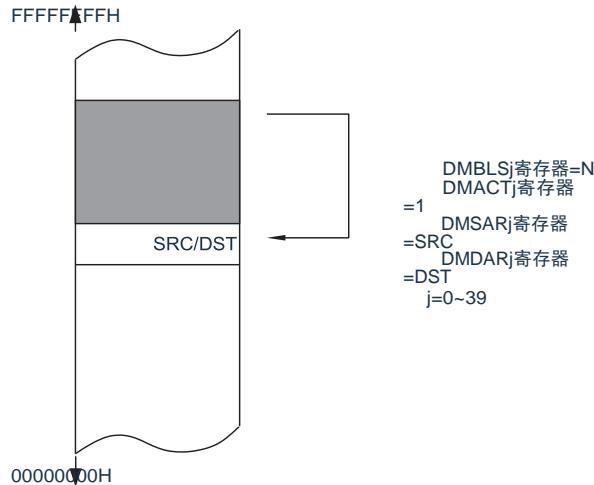
图23-18: 重复模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC+N	DST
1	X	1	1	重复区	递增	SRC+N	DST+N
X	0	0	1	固定	重复区	SRC	DST+N
X	1	0	1	递增	重复区	SRC+N	DST+N

备注: X: “0”或者“1”

DMACTj寄存器=1



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC	DST
1	X	1	1	重复区	递增	SRC	DST+N
X	0	0	1	固定	重复区	SRC	DST
X	1	0	1	递增	重复区	SRC+N	DST

备注: SRC0: 源地址的初始值 DST0: 目标地址的初始值 X: “0”或者“1”

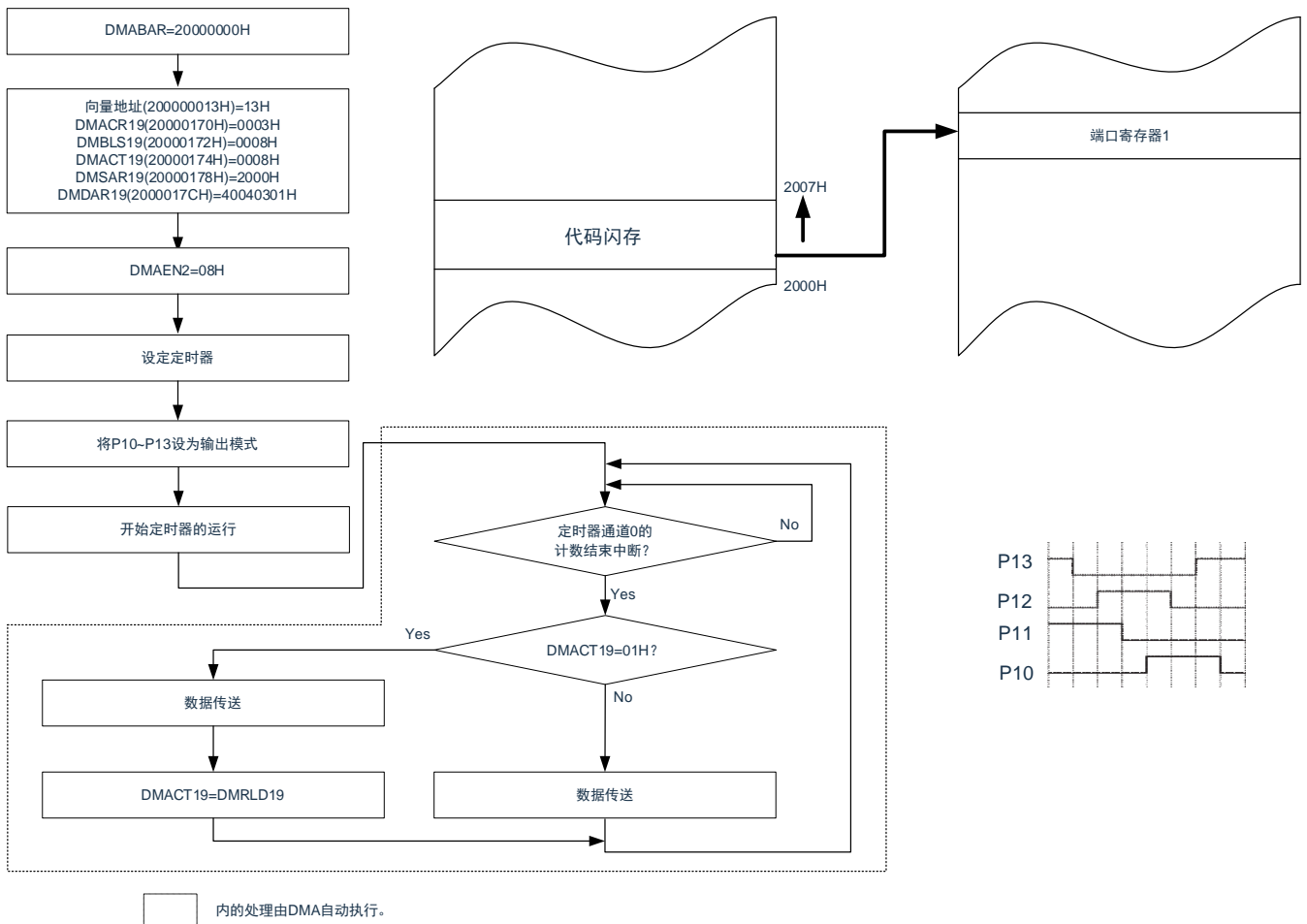
注意: 在使用重复模式时, 必须将重复区的数据长度设置在65535字节以内。

(1) 重复模式的使用例子1：使用端口的步进马达控制脉冲输出

使用Timer4的通道0间隔定时器功能启动DMA，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口。

- 向量地址分配在20000013H，控制数据分配在20000170H~2000017FH。
- 将代码闪存的02000H~02007H的8字节传送到端口寄存器1(40040301H)。
- 禁止重复模式中断。

图23-19：重复模式的使用例子1：使用端口的步进马达控制脉冲输出



要停止输出时，必须在停止定时器的运行后清除DMAEN2的bit3。

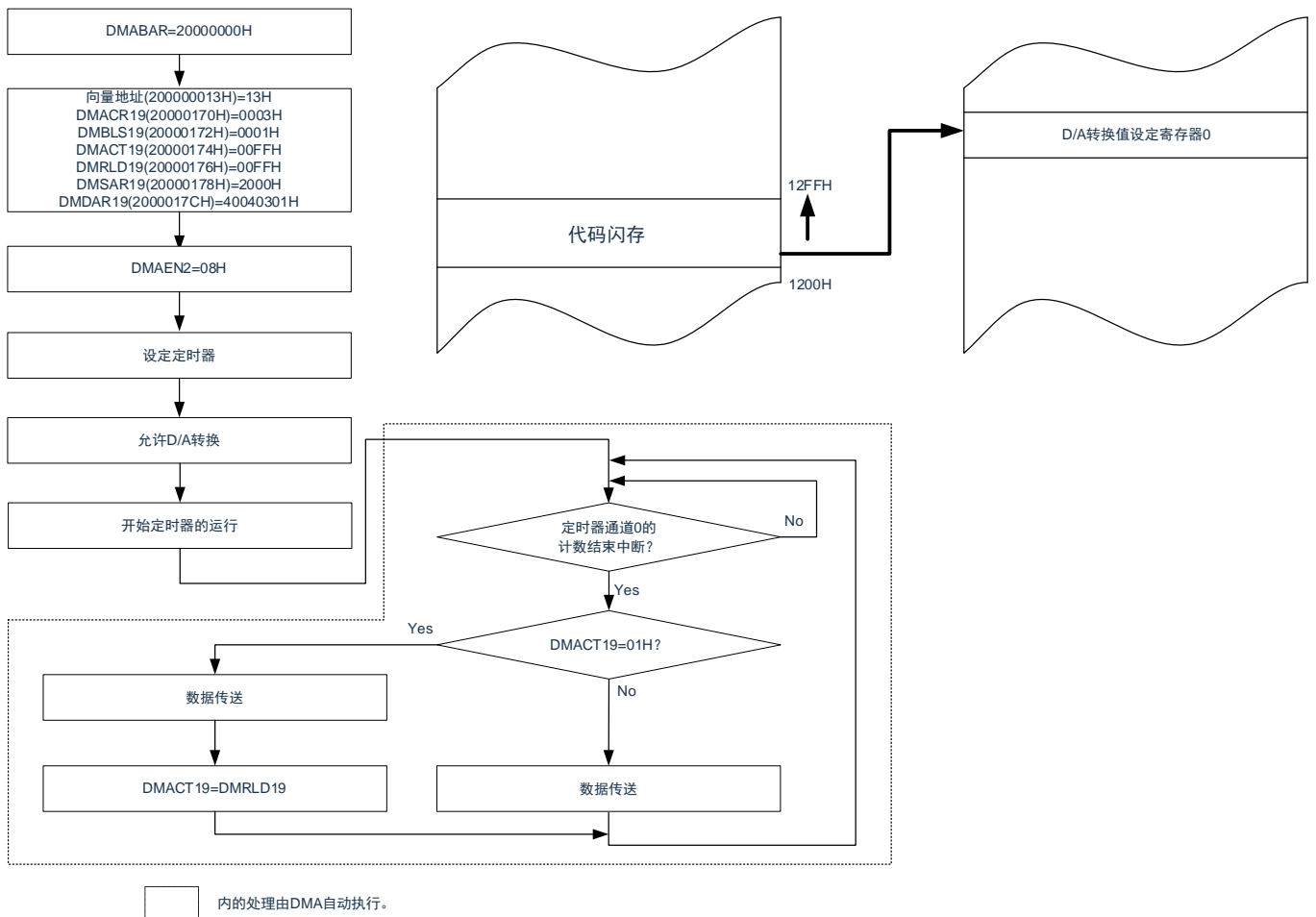
(2) 重复模式的使用例子2：使用8位D/A转换器的正弦波输出

使用使用Timer4的通道0间隔定时器功能并且通过中断启动DMA，将保存在数据闪存的正弦波表传送到8位D/A转换值设置寄存器0(40044734H)。

将定时器的间隔时间设置为D/A的输出准备时间。

- 向量地址分配在20000013H，控制数据分配在20000170H~2000017FH。
- 将数据闪存的1200H~12FEH的255字节数据传送到D/A转换值设置寄存器0(40044734H)。
- 禁止重复模式中断。

图23-20：重复模式的使用例子2：使用8位D/A转换器的正弦波输出



要停止输出时，必须在停止定时器的运行后清除DMAEN2的bit3。

23.4.4 链传送

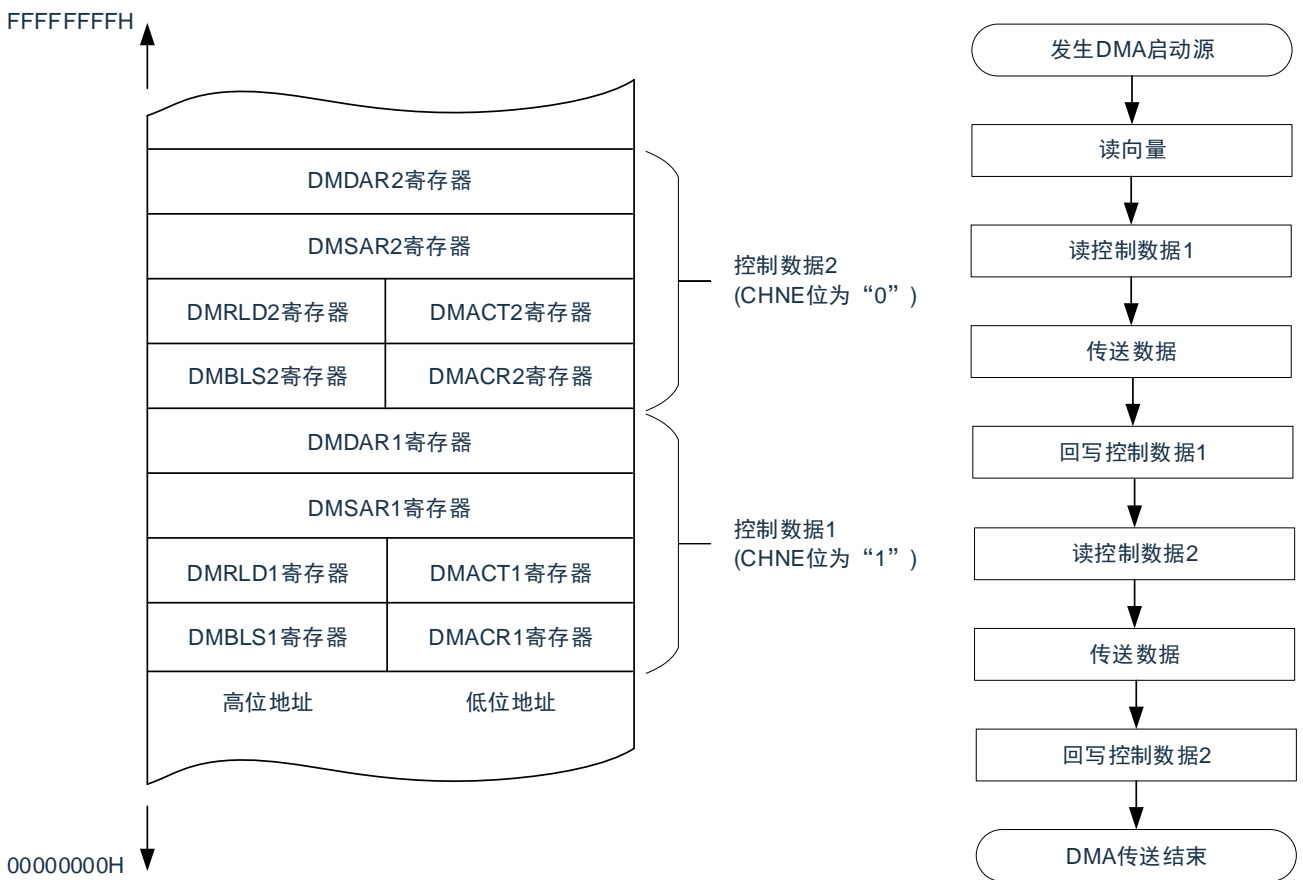
当DMACRj(j=0~38)寄存器的CHNE位为“1(允许链传送)”时，能通过1个启动源连续进行多个数据的传送。

DMA一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在DMA控制数据区的控制数据。如果读到的控制数据的CHNE位为“1”(允许链传送)，就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到CHNE位为“0”(禁止链传送)的控制数据传送结束为止。

在使用多个控制数据进行链传送时，第一个控制数据设置的传送次数有效，而第2个以后处理的控制数据的传送次数无效。

链传送的流程图如图23-21所示。

图23-21：链传送的流程图



注意：

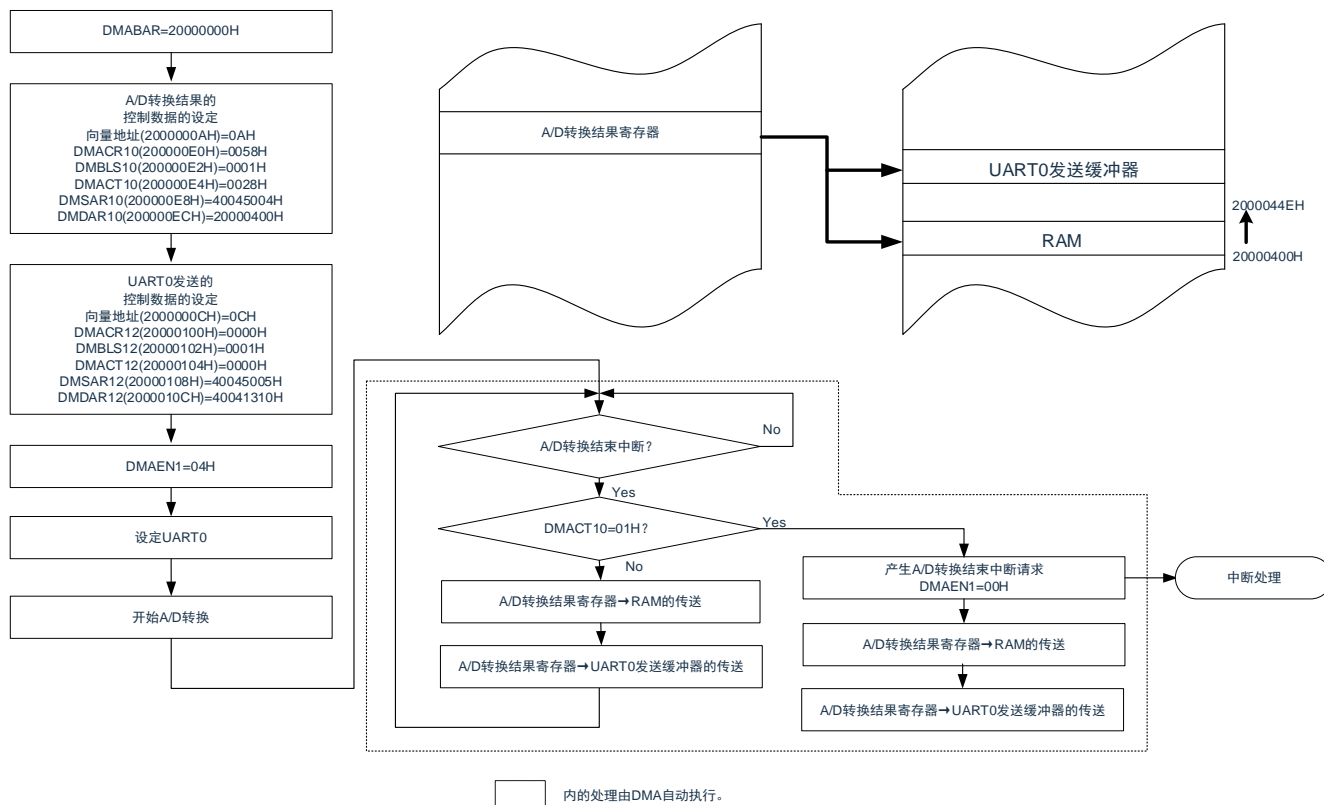
1. 必须将DMACR39寄存器的CHNE位置“0”(禁止链传送)。
2. 在链传送的第2次以后的数据传送时，DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位不变为“0”(禁止DMA启动)，并且不产生中断请求。

(1) 链传送的使用例子：连续取A/D转换结果进行UART0发送

通过A/D转换结束中断启动DMA，并且将A/D转换结果传送到RAM进行UART0发送。

- 向量地址分别为2000000AH和2000000CH。
- A/D转换结果的控制数据分配在200000E0H~200000EFH。
- UART0发送的控制数据分配在20000100H~2000010FH。
- 将A/D转换结果寄存器(40045004H, 40045005H)的2字节数据传送到RAM的20000400H~2000044FH，并且将A/D转换结果寄存器的高位1字节(40045005H)传送到UART0的发送缓冲器(40041310H)。

图23-22：链传送的使用例子：连续取A/D转换结果进行UART0发送



23.5 使用DMA时的注意事项

23.5.1 DMA控制数据和向量表的设置

- 必须在将全部的DMA启动源设置为禁止启动的状态下更改DMA基址寄存器(DMABAR)。
- 只能改写1次DMA基址寄存器(DMABAR)。
- 必须在对应的DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位为“0”(禁止DMA启动)时更改DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj寄存器的数据。
- 必须在对应的DMAENi(i=0~4)寄存器的DMAENi0~DMAENi7位为“0”(禁止DMA启动)时更改设置在向量表中的DMA控制数据区的起始地址。

23.5.2 DMA控制数据区和DMA向量表区的分配

能分配DMA控制数据和向量表的区域因产品和使用条件而不同。

- 堆栈区、DMA控制数据区和DMA向量表区不能重叠。
- 当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位(RPERDIS=0)时，即使在使用正常模式时也必须对DMRLD寄存器进行初始化(0000H)。

23.5.3 DMA的执行时钟数

DMA启动时的执行情况和所需的时钟数如表23-9所示。

表23-9: DMA启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注1	注2	注2

注1: 有关回写控制数据所需的时钟数, 请参照“表23-10: 回写控制数据所需的时钟数”。

注2: 有关读写数据所需的时钟数, 请参照“表23-11: 读写数据所需的时钟数”。

表23-10: 回写控制数据所需的时钟数

DMACR寄存器的设置				地址设置		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DMACTj 寄存器	DMRLDj 寄存器	DMSARj 寄存器	DMDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	重复区	固定	回写	回写	不回写	回写	2
X	1	0	1		递增	回写	回写	回写	回写	3

备注: j=0~23, X: “0”或者“1”

表23-11: 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器 (SFR)	扩展特殊功能寄存器(2ndSFR)	
					无等待	等待
读数据	1	2	4	1	1	1+等待数
写数据	1	—	—	1	1	1+等待数

23.5.4 DMA的响应时间

DMA响应时间如表23-12所示。DMA响应时间是指从检测到DMA启动源到开始DMA传送的时间，不包括DMA的执行时钟数。

表23-12: DMA的响应时间

	最短时间	最长时间
响应时间	3个时钟	23个时钟

但是，在以下情况下DMA的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部RAM执行指令的情况最长响应时间：20个时钟
- 存取发生等待的TimerA寄存器的情况最长响应时间：各条件的最长响应时间+1个时钟

备注：1个时钟： $1/F_{CLK}$ (F_{CLK} : CPU/外围硬件时钟)

23.5.5 DMA的启动源

- 不能在从输入DMA启动源到结束DMA传送的期间输入相同的启动源。
- 在产生DMA启动源的位置，不能操作该启动源对应的DMA启动允许位。
- 如果DMA启动源发送竞争，就在CPU接受DMA传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“23.3.3向量表”。
- 如果在以下某个状态下允许DMA启动，就开始DMA传送，并且在传送结束后产生中断。因此，必须根据需要在确认比较器的监视标志(CnMON)后置为允许DMA启动。
 - 设置为通过比较器注的单边沿检测产生中断请求(CnEDG=0)并且通过比较器的上升沿产生中断请求(CnEPO=0)而且IVCMP>IVREF(或者内部基准电压1.45V)。
 - 设置为通过比较器的单边沿检测产生中断请求(CnEDG=0)并且通过比较器的下降沿产生中断请求(CnEPO=1)而且IVCMP<IVREF(或者内部基准电压1.45V)。(n=0、1)

23.5.6 待机模式中的运行

状态	DMA运行
睡眠模式	能运行(禁止在低功耗RTC模式中运行)。
深度睡眠模式	能接受DMA启动源, 并进行DMA传送 ^{注1}

注1: 在深度睡眠模式中, 能在检测到DMA启动源后进行DMA传送, 并且在传送结束后返回到深度睡眠模式。但是, 因为在深度睡眠模式中代码闪存和数据闪存停止运行, 所以不能将闪存设置为传送源。

第24章 联动控制器(EVENTC)

24.1 EVENTC的功能

EVENTC将各外围功能输出的事件进行外围功能之间的相互链接。能通过事件链接不经过CPU而直接进行外围功能之间的协作运行。

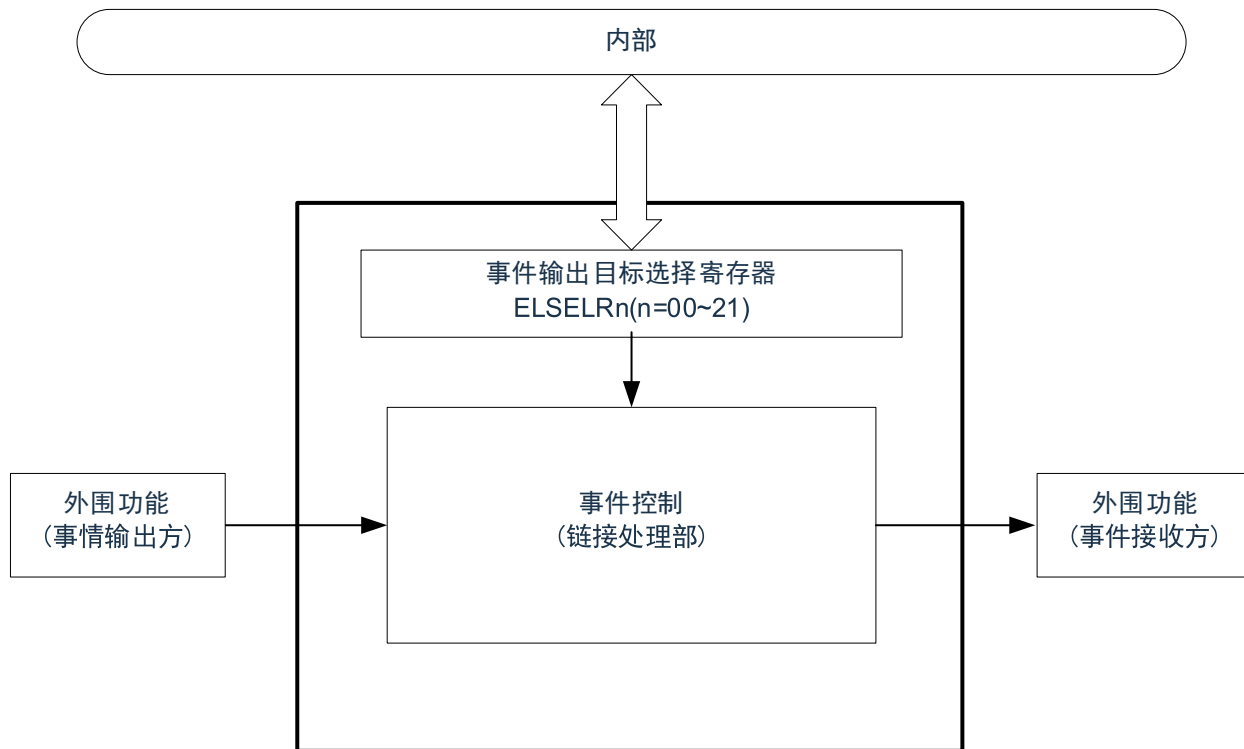
EVENTC有以下功能：

- 根据产品，能将21~22种外围功能的事件信号直接链接到指定的外围功能。
- 根据产品，能将事件信号用作10种外围功能中的1种外围功能运行的启动源。

24.2 EVENTC的结构

EVENTC的框图如图24-1所示。

图24-1：EVENTC的框图



24.3 控制寄存器

控制器寄存器如表24-1所示。

表24-1：控制EVENTC的寄存器

寄存器名	符号
事件输出目标选择寄存器00	ELSELR00
事件输出目标选择寄存器01	ELSELR01
事件输出目标选择寄存器02	ELSELR02
事件输出目标选择寄存器03	ELSELR03
事件输出目标选择寄存器04	ELSELR04
事件输出目标选择寄存器05	ELSELR05
事件输出目标选择寄存器06	ELSELR06
事件输出目标选择寄存器07	ELSELR07
事件输出目标选择寄存器08	ELSELR08
事件输出目标选择寄存器09	ELSELR09
事件输出目标选择寄存器10	ELSELR10
事件输出目标选择寄存器11	ELSELR11
事件输出目标选择寄存器12	ELSELR12
事件输出目标选择寄存器13	ELSELR13
事件输出目标选择寄存器14	ELSELR14
事件输出目标选择寄存器15	ELSELR15
事件输出目标选择寄存器16	ELSELR16
事件输出目标选择寄存器17	ELSELR17
事件输出目标选择寄存器18	ELSELR18
事件输出目标选择寄存器19	ELSELR19
事件输出目标选择寄存器20	ELSELR20
事件输出目标选择寄存器21	ELSELR21

24.3.1 输出目标选择寄存器n(ELSELRn)(n=00~21)

ELSELRn寄存器将各事件信号链接到事件接受方外围功能(链接目标外围功能)接受事件时的运行。不能将多个事件输入链接到相同的事件输出目标(事件接受方)。否则,事件接受方外围功能的运行可能不定而无法正常地接受事件信号。另外,不能将事件链接发生源和事件输出目标设置为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设置ELSELRn寄存器。

ELSELRn寄存器(n=00~21)和外围功能的对应如表24-2所示, ELSELRn寄存器(n=00~21)的设置值和链接目标外围功能接受事件时的运行的对应如表24-3所示。

图24-2: 事件输出目标选择寄存器n(ELSELRn)的格式

地址:	40043400H(ELSELR00)~40043415H(ELSELR21)	复位后:	00H	R/W				
符号	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn3 ^{注1}	ELSELn2	ELSELn1	ELSELn0	事件链接的选择
0	0	0	0	禁止事件链接。
0	0	0	1	选择所链接的外围功能1的运行 ^{注1} 。
0	0	1	0	选择所链接的外围功能2的运行 ^{注1} 。
0	0	1	1	选择所链接的外围功能3的运行 ^{注1} 。
0	1	0	0	选择所链接的外围功能4的运行 ^{注1} 。
0	1	0	1	选择所链接的外围功能5的运行 ^{注1} 。
0	1	1	0	选择所链接的外围功能6的运行 ^{注1} 。
0	1	1	1	选择所链接的外围功能7的运行 ^{注1} 。
1	0	0	0	选择所链接的外围功能8的运行 ^{注1} 。
1	0	0	1	选择所链接的外围功能9的运行 ^{注1} 。
上述以外				禁止设置。

注1: 请参照“表24-3: ELSELRn寄存器(n=00~21)的设置值和链接目标外围功能接受事件时的运行的对应”。

表24-2: ELSELRn寄存器(n=00~21)和外围功能的对应

寄存器名	事件发生源(事件输入n的输出源)	事件内容
ELSELR00	外部中断边沿检测0	INTP0
ELSELR01	外部中断边沿检测1	INTP1
ELSELR02	外部中断边沿检测2	INTP2
ELSELR03	外部中断边沿检测3	INTP3
ELSELR04	外部中断边沿检测4	INTP4
ELSELR05	外部中断边沿检测5	INTP5
ELSELR06	键返回信号检测	INTKR
ELSELR07	RTC固定周期/闹钟一致检测	INTRTC
ELSELR08	定时器M输入捕捉A0/比较匹配A0	INTTMM0
ELSELR09	定时器M输入捕捉B0/比较匹配B0	INTTMM0
ELSELR10	定时器M输入捕捉A1/比较匹配A1	INTTMM1
ELSELR11	定时器M输入捕捉B1/比较匹配B1	INTTMM1
ELSELR12	定时器M下溢	TMM下溢信号
ELSELR13	定时器A下溢/脉宽测量期间结束/脉冲周期测量期间结束	INTTMA
ELSELR14	定时器B输入捕捉A/比较匹配A	INTTMB
ELSELR15	定时器B输入捕捉B/比较匹配B	INTTMB
ELSELR16	Timer4通道00的计数结束/捕捉结束	INTTM00
ELSELR17	Timer4通道01的计数结束/捕捉结束	INTTM01
ELSELR18	Timer4通道02的计数结束/捕捉结束	INTTM02
ELSELR19	Timer4通道03的计数结束/捕捉结束	INTTM03
ELSELR20	比较器检测0	INTCMP0
ELSELR21	比较器检测1	INTCMP1

表24-3: ELSELRn寄存器(n=00~21)的设置值和链接目标外围功能接受事件时的运行的对应

ELSELRn寄存器的 ELSELn3~ELSELn0位	链接目标 No.	链接目标外围功能	接受事件时的运行
0001B	1	A/D转换器	开始A/D转换。
0010B	2	Timer4通道0的定时器输入 <small>注1</small>	延迟计数器、输入脉冲间隔的测量、外部事件计数器
0011B	3	Timer4 通道1的定时器输入 <small>注2</small>	延迟计数器、输入脉冲间隔的测量、外部事件计数器
0100B	4	定时器A	计数源
0101B	5	定时器B	TBIO1的输入捕捉
0110B	6	定时器M	TMIOD0的输入捕捉, 脉冲输出的强制截止
0111B	7	定时器M	TMIOD1的输入捕捉, 脉冲输出的强制截止
1000B	8	DA0 <small>注3</small>	实时输出
1001B	9	DA1 <small>注3</small>	实时输出

注1: 要选择Timer4通道0的定时器输入作为链接目标外围功能时, 必须先通过定时器时钟选择寄存器0(TPS0)将通道0的运行时钟设置为FCLK, 通过噪声滤波器允许寄存器1(NFEN1)将TI00引脚的噪声滤波器置为OFF(TNFEN0=0), 并且通过定时器输入选择寄存器0(TIS0)将通道0使用的定时器输入设置为联动控制器的事件输入信号。

注2: 要选择Timer4通道1的定时器输入作为链接目标外围功能时, 必须先通过定时器时钟选择寄存器0(TPS0)将通道1的运行时钟设置为FCLK, 通过噪声滤波器允许寄存器1(NFEN1)将TI01引脚的噪声滤波器置为OFF(TNFEN01=0), 并且通过定时器输入选择寄存器0(TIS0)将通道1使用的定时器输入设置为EVENTC的事件输入信号。

注3: 如果要在D/A转换的实时输出模式有效时进入深度睡眠状态, 就必须在进入深度睡眠模式之前禁止EVENTC的事件链接。

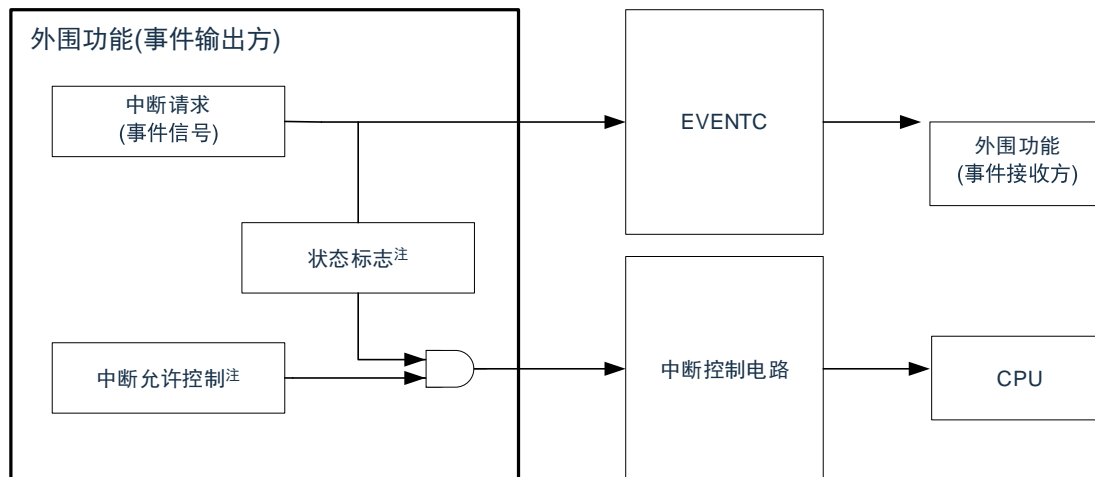
24.4 EVENTC的运行

将各外围功能产生的事件信号用作中断控制电路的中断请求所使用的路径和用作EVENTC事件所使用的路径相互独立。因此，各事件信号与中断控制无关，能用作事件接受方外围功能运行的事件信号。

中断处理和EVENTC的关系如图24-3所示。此图是以有中断请求状态标志和中断允许位(控制允许或者禁止)的外围功能为例的关系。

通过EVENTC接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行(参照“表24-3: ELSELRn寄存器(n=00~21)的设置值和链接目标外围功能接受事件时的运行的对应”)。

图24-3: 中断处理和EVENTC的关系



注：有些外围功能没有此功能。

接受事件的外围功能的响应如表24-4所示。

表24-4: 接受事件的外围功能的响应

事件接受目标No.	事件链接目标的功能	事件接受后的运行	响应
1	A/D转换器	A/D转换	EVENTC事件直接变为A/D转换的硬件触发。
2	Timer4 通道0的定时器输入	延迟计数器输入脉宽的测量外部 事件计数器	在从发生EVENTC事件经过3个或者4个F _{CLK} 周期 后进行边沿的检测。
3	Timer4 通道1的定时器输入	延迟计数器输入脉宽的测量外部 事件计数器	在从发生EVENTC事件经过3个或者4个F _{CLK} 周期 后进行边沿的检测。
4	定时器A	计数源	EVENTC事件直接变为定时器A的计数源。
5	定时器B	TBIOB的输入捕捉	在从发生EVENTC事件经过2个或者3个F _{CLK} 周期 后产生计数开始触发。
6	定时器M	TMIOD0的输入捕捉	在从发生EVENTC事件经过2个或者3个定时器M 的运行时钟周期后产生计数开始触发。
		脉冲输出的强制截止	在从发生EVENTC事件经过2个或者3个定时器M 的运行时钟周期后变为强制截止状态。
7	定时器M	TMIOD1的输入捕捉	在从发生EVENTC事件经过2个或者3个定时器M 的运行时钟周期后产生计数开始触发。
		脉冲输出的强制截止	在从发生EVENTC事件经过2个或者3个定时器M 的运行时钟周期后变为强制截止状态。
8	D/A转换器的通道0	实时输出(通道0)	在从发生EVENTC事件经过2个或者3个F _{CLK} 周期 后开始通道0的D/A转换。
9	D/A转换器的通道1	实时输出(通道1)	在从发生EVENTC事件经过2个或者3个F _{CLK} 周期 后开始通道1的D/A转换。

第25章 中断功能

Cortex-M0+处理器内置了嵌套向量中断控制器(NVIC)，支持最多32个中断请求(IRQ)输入，以及1个不可屏蔽中断(NMI)输入，另外，处理器还支持多个内部异常。

本系统中对32个中断请求(IRQ)输入和1个不可屏蔽中断(NMI)输入进行了扩展，最多能支持64个中断源，以及一个不可屏蔽中断源。本用户手册只对本系统中的扩展功能进行了说明，Cortex-M0+处理器内置NVIC的功能，请参考Cortex-M0+处理器的用户手册。

中断源的实际个数因产品而不同。

		24引脚	32引脚	36引脚	44引脚	40, 48, 52引脚	64引脚
可屏蔽中断	外部	9	12	11	10	12	13
	内部	33	33	33	33	33	33

25.1 中断功能的种类

中断功能有以下2种。

(1) 可屏蔽中断

这是接受屏蔽控制的中断。如果中断屏蔽标志寄存器没有打开，中断请求即使产生，也不会被响应。

可产生待机解除信号，解除深度睡眠模式、睡眠模式。

可屏蔽中断分为外部中断请求和内部中断请求。

(2) 不可屏蔽中断

这是不接受屏蔽控制的中断，中断请求一旦产生，CPU必须响应。

25.2 中断源和结构

中断源列表参照表25-1。

表25-1：中断源一览表(1/4)

中断处理	中断源编号	中断源		内部/外部	基本结构类型 ^{注1}	64引脚	52引脚	48引脚	44引脚	40引脚	36引脚	32引脚	24引脚		
		名称	触发												
可屏蔽	0	INTLVI	电压检测 ^{注2}	内部	(A)	○	○	○	○	○	○	○	○		
	1	INTP0	引脚输入边沿的检测	外部	(B)	○	○	○	○	○	○	○	○		
	2	INTP1	引脚输入边沿的检测			○	○	○	○	○	○	○	○	○	
	3	INTP2	引脚输入边沿的检测			○	○	○	○	○	○	○	○	○	
	4	INTP3	引脚输入边沿的检测			○	○	○	○	○	○	○	○	○	
	5	INTP4	引脚输入边沿的检测			○	○	○	○	○	○	○	○	○	
	6	INTP5	引脚输入边沿的检测			○	○	○	○	○	○	○	○	○	
	7	INTST2/INTSSPI20/INTIIC20	UART2发送的传送结束或者缓冲器空中断/SSPI20的传送结束或者缓冲器空中断/IIC20的传送结束	内部	(A)	○	○	○	○	○	○	○	○	○	
	8	INTSR2/INTSSPI21/INTIIC21	UART2接收的传送结束/SSPI21的传送结束或者缓冲器空中断/IIC21的传送结束			○	○	○	○	○ ^{注3}	○ ^{注3}	○ ^{注3}	○ ^{注3}	○ ^{注3}	
	9	INTSRE2	发生UART2接收的通信错误			○	○	○	○	○	○	○	○	○	
	10	INTST0/INTSSPI00/INTIIC00	UART0发送的传送结束或者缓冲器空中断/SSPI00的传送结束或者缓冲器空中断/IIC00的传送结束			○	○	○	○	○	○	○	○	○	○
	11	INTSR0/INTSSPI01/INTIIC01	UART0接收的传送结束/SSPI01的传送结束或者缓冲器空中断/IIC01的传送结束			○	○	○	○	○ ^{注4}	○ ^{注4}	○ ^{注4}	○ ^{注4}	○ ^{注4}	
12	INTSRE0	发生UART0接收的通信错误	○			○	○	○	○	○	○	○	○	○	

注1：基本构成类型(A)~(D)分别对应图25-1的(A)~(D)。

注2：这是将电压检测电平寄存器(LVIS)的bit7(LVIMD)置“0”的情况。

注3：只限于INTSR2。

注4：只限于INTSR0。

表25-1: 中断源一览表(2/4)

中断处理	中断源编号	中断源		内部/外部	基本结构类型 ^{注1}	64引脚	52引脚	48引脚	44引脚	40引脚	36引脚	32引脚	24引脚	
		名称	触发											
可屏蔽	13	INTST1/INTSSPI10/INTIC10	UART1发送的传送结束或者缓冲器空中断/SSPI10的传送结束或者缓冲器空中断/IIC10的传送结束	内部	(A)	○	○ ^{注2}	○ ^{注2}	○ ^{注2}	○ ^{注2}	○ ^{注2}	○ ^{注2}	○ ^{注2}	
	14	INTSR1/INTSSPI11/INTIC11	UART1接收的传送结束/SSPI11的传送结束或者缓冲器空中断/IIC11的传送结束			○	○	○	○	○	○	○	○	○
	15	INTSRE1	发生UART1接收的通信错误			○	○	○	○	○	○	○	○	○
	16	INTIICA0	IICA0通信结束			○	○	○	○	○	○	○	○	○
	17	INTTM00	定时器通道00的计数结束或者捕捉结束			○	○	○	○	○	○	○	○	○
	18	INTTM01	定时器通道01的计数结束或者捕捉结束			○	○	○	○	○	○	○	○	○
	19	INTTM02	定时器通道02的计数结束或者捕捉结束			○	○	○	○	○	○	○	○	○
	20	INTTM03	定时器通道03的计数结束或者捕捉结束			○	○	○	○	○	○	○	○	○
	21	INTAD	A/D转换结束			○	○	○	○	○	○	○	○	○
	22	INTRTC	实时时钟的固定周期/闹钟一致检测			○	○	○	○	○	○	○	○	○
	23	INTKR	键返回信号的检测	外部	(C)	○	○	○	○	○	-	-	-	
	24	INTCMP0	比较器检测0	内部	(A)	○	○	○	○	○	○	○	○	
	25	INTCMP1	比较器检测1			○	○	○	○	○	○	○	○	
	26	INTTMA	定时器A的下溢			○	○	○	○	○	○	○	○	
	27	INTTMM0	定时器M0的输入捕捉、比较匹配、上溢、下溢中断			○	○	○	○	○	○	○	○	
28	INTTMM1	定时器M1的输入捕捉、比较匹配、上溢、下溢中断	○			○	○	○	○	○	○	○		
29	INTTMB	定时器B的输入捕捉、比较匹配、上溢、下溢中断	○			○	○	○	○	○	○	○		
30	INTTMC	定时器C的上溢	○			○	○	○	○	○	○	○		
31	INTFL	Flash编程终了	○			○	○	○	○	○	○	○		

注1: 基本构成类型(A)~(D)分别对应图25-1的(A)~(D)。

注2: 只限于INTST1。

表25-1: 中断源一览表(3/4)

中断处理	中断源编号	中断源		内部/外部	基本结构类型 ^{注1}	64引脚	52引脚	48引脚	44引脚	40引脚	36引脚	32引脚	24引脚
		名称	触发										
可屏蔽	32	保留	-	-	-	-	-	-	-	-	-	-	-
	33	INTP6	引脚输入边沿的检测	外部	(B)	○	○	○	-	○	○	○	-
	34	INTP7				○	-	-	-	-	○	-	
	35	INTP8				○	○	○	○	○	○	○	
	36	INTP9				○	○	○	-	○	-	-	
	37	INTP10				○	○	○	○	○	○	○	
	38	INTP11				○	○	○	○	○	○	○	
	39	保留				-	-	-	-	-	-	-	-
	40	保留	-	-	-	-	-	-	-	-	-	-	
	41	INTC0ERR	CAN错误中断	内部	(A)	○	○	○	○	○	○	○	○
	42	保留	-	-	-	-	-	-	-	-	-	-	
	43	保留	-	-	-	-	-	-	-	-	-	-	
	44	INTTM01H	定时器通道01的计数结束或者捕捉结束(高8位定时器工作时)	内部	(A)	○	○	○	○	○	○	○	○
	45	保留	-	-	-	-	-	-	-	-	-	-	
	46	保留	-	-	-	-	-	-	-	-	-	-	
	47	INTTM03H	定时器通道03的计数结束或者捕捉结束(高8位定时器工作时)	内部	(A)	○	○	○	○	○	○	○	○
	48	INTDIV	除法器计算结束	内部	(A)	○	○	○	○	○	○	○	○
	49	保留	-	-	-	-	-	-	-	-	-	-	
	50	保留	-	-	-	-	-	-	-	-	-	-	
	51	保留	-	-	-	-	-	-	-	-	-	-	
	52	保留	-	-	-	-	-	-	-	-	-	-	
	53	保留	-	-	-	-	-	-	-	-	-	-	
	54	INTIT	间隔信号的检测	内部	(A)	○	○	○	○	○	○	○	○
	55	INTC0REC	CAN接收结束	内部	(A)	○	○	○	○	○	○	○	○
	56	INTC0WUP	CAN唤醒	内部	(A)	○	○	○	○	○	○	○	○
	57	INTC0TRX	CAN发送结束	内部	(A)	○	○	○	○	○	○	○	○
	58	保留	-	-	-	-	-	-	-	-	-	-	
	59	保留	-	-	-	-	-	-	-	-	-	-	
	60	保留	-	-	-	-	-	-	-	-	-	-	
	61	保留	-	-	-	-	-	-	-	-	-	-	
	62	保留	-	-	-	-	-	-	-	-	-	-	
	63	保留	-	-	-	-	-	-	-	-	-	-	

注1: 基本构成类型(A)~(D)分别对应图25-1的(A)~(D)。

表25-1: 中断源一览表(4/4)

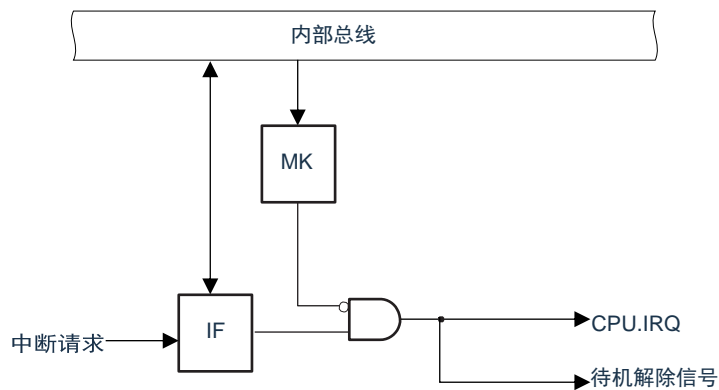
中断处理	中断源编号	中断源		内部/外部	基本结构类型 ^{注1}
		名称	触发		
不可屏蔽	—	INTWDT	看门狗定时器间隔中断 ^{注2}	内部	(D)

注1: 基本构成类型(A)~(D)分别对应图25-1的(A)~(D)。

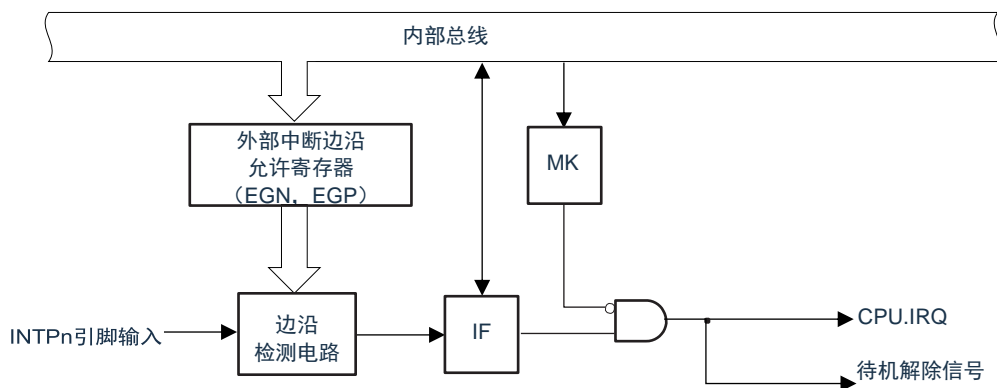
注2: 这是将选项字节(000C0H)的bit7(WDTINT)置“1”的情况。

图25-1：中断功能的基本结构

(A) 内部可屏蔽中断

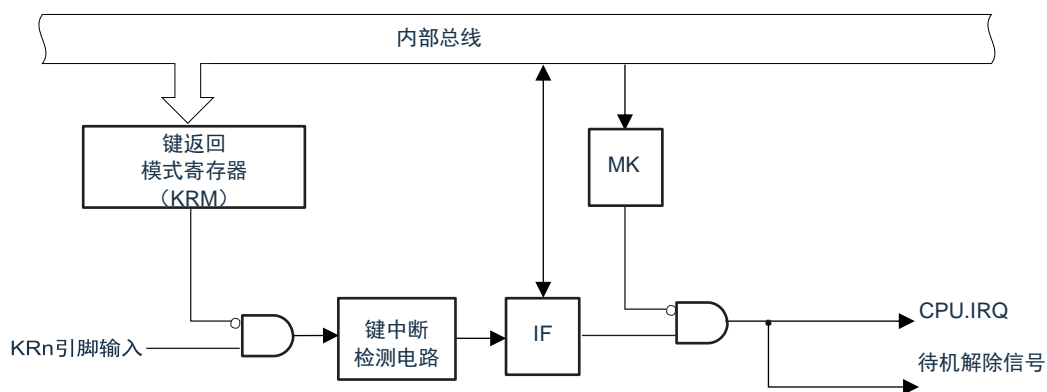


(B) 外部可屏蔽中断(INTPn)



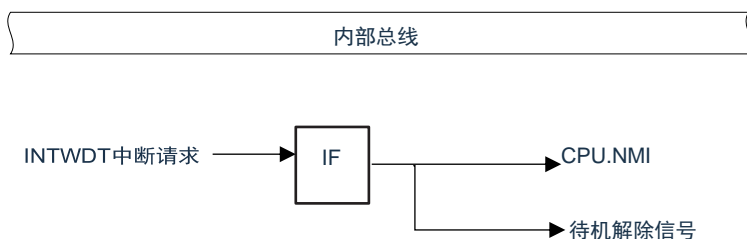
注：32引脚：n=0~8、10~11
 36引脚：n=0~6、8、10~11
 40、48、52引脚：n=0~6、8~11
 64引脚：n=0~11

(C) 外部可屏蔽中断(INTKR)



注：40引脚：n=0、2~5
 48引脚：n=0~5
 52、64引脚：n=0~7

(D) 不可屏蔽中断



注：不可屏蔽中断的中断请求标志IF没有实体寄存器，不能通过总线读写寄存器来产生中断请求。

25.3 控制中断功能的寄存器

通过以下4种寄存器控制中断功能。

- 中断请求标志寄存器(IF00~IF31)
- 中断屏蔽标志寄存器(MK00~MK31)
- 外部中断上升沿允许寄存器(EGP0、EGP1)
- 外部中断下降沿允许寄存器(EGN0、EGN1)

25.3.1 中断请求标志寄存器(IF00~IF31)

通过发生对应的中断请求或者执行指令，将中断请求标志置“1”。通过产生复位信号或者执行指令，将中断请求标志清“0”。

通过8位存储器操作指令设置IF00L~IF31L，IF00H~IF31H寄存器或通过32位存储器操作指令设置IF00~IF31寄存器。

在产生复位信号后，这些寄存器的值变为“0000_0000H”。

图25-2：中断请求标志寄存器(IFm)的格式(m=0~31)

地址：IF00：40006000H，IF01：40006004H，IF02：40006008H，IF03：4000600CH
 IF04：40006010H，IF05：40006014H，IF06：40006018H，IF07：4000601CH
 IF08：40006020H，IF09：40006024H，IF10：40006028H，IF11：4000602CH
 IF12：40006030H，IF13：40006034H，IF14：40006038H，IF15：4000603CH
 IF16：40006040H，IF17：40006044H，IF18：40006048H，IF19：4000604CH
 IF20：40006050H，IF21：40006054H，IF22：40006058H，IF23：4000605CH
 IF24：40006060H，IF25：40006064H，IF26：40006068H，IF27：4000606CH
 IF28：40006070H，IF29：40006074H，IF30：40006078H，IF31：4000607CH
 复位值：0000_0000H R/W

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

IFmL	编号0~31的中断源的中断请求标志
0	不产生中断请求信号。
1	产生中断请求，处于中断请求状态。

IFmH	编号32~63的中断源的中断请求标志
0	不产生中断请求信号。
1	产生中断请求，处于中断请求状态。

注1：中断源与中断请求标志寄存器的对应关系见表25-2。

注2：中断请求标志寄存器与CPU.IRQ的对应关系见图25-4。

注3：中断请求标志寄存器不会自清零，中断响应后必须将寄存器写0。

25.3.2 中断屏蔽标志寄存器(MK00~MK31)

中断屏蔽标志设置允许或者禁止对应的可屏蔽中断处理。

通过8位存储器操作指令设置MK00L~MK31L, MK00H~MK31H寄存器或通过32位存储器操作指令设置MK00~MK31寄存器。

在产生复位信号后, 这些寄存器的值变为“FFFF_FFFF”。

图25-3: 中断请求屏蔽寄存器(MK_m)的格式(m=0~31)

地址: MK00: 40006100H, MK01: 40006104H, MK02: 40006108H, MK03: 4000610CH

MK04: 40006110H, MK05: 40006114H, MK06: 40006118H, MK07: 4000611CH

MK08: 40006120H, MK09: 40006124H, MK10: 40006128H, MK11: 4000612CH

MK12: 40006130H, MK13: 40006134H, MK14: 40006138H, MK15: 4000613CH

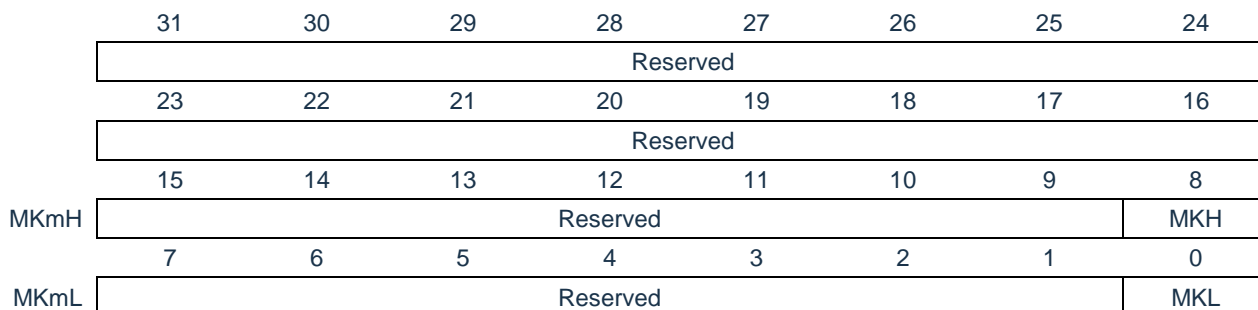
MK16: 40006140H, MK17: 40006144H, MK18: 40006148H, MK19: 4000614CH

MK20: 40006150H, MK21: 40006154H, MK22: 40006158H, MK23: 4000615CH

MK24: 40006160H, MK25: 40006164H, MK26: 40006168H, MK27: 4000616CH

MK28: 40006170H, MK29: 40006174H, MK30: 40006178H, MK31: 4000617CH

复位值: FFFF_FFFFH R/W



MK _m L	编号0~31的中断源的中断处理控制 ^{注1}
0	允许中断处理。
1	禁止中断处理。

MK _m H	编号32~63的中断源的中断处理控制 ^{注2}
0	允许中断处理。
1	禁止中断处理。

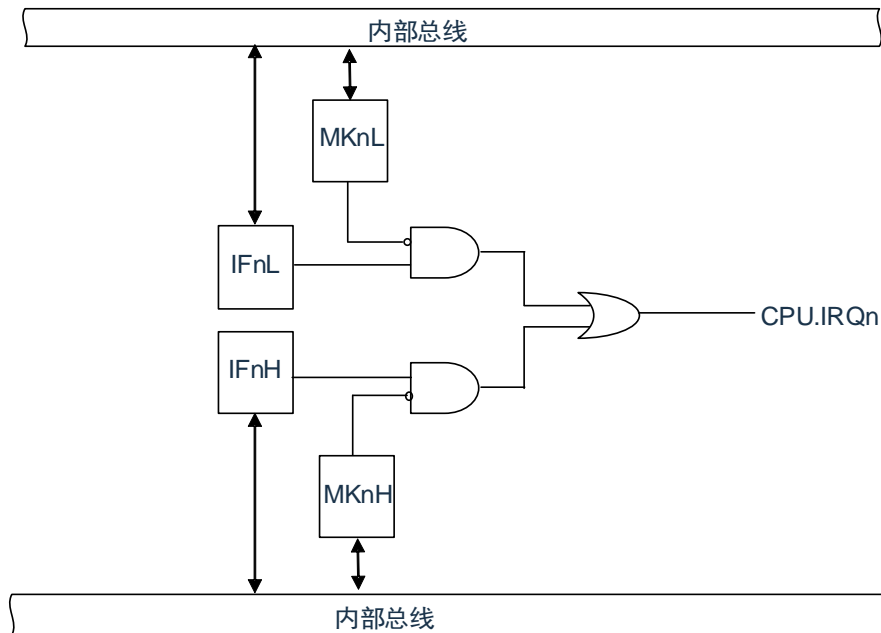
注1: 中断源与中断请求屏蔽寄存器的对应关系见表25-2.

注2: 中断请求屏蔽寄存器与CPU.IRQ的对应关系见图25-4.

表25-2: 中断源和各标志寄存器的对应关系

编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器	编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器
0	INTLVI	IF00.IFL	MK00.MKL	32	保留	IF00.IFH	MK00.MKH
1	INTP0	IF01.IFL	MK01.MKL	33	INTP6	IF01.IFH	MK01.MKH
2	INTP1	IF02.IFL	MK02.MKL	34	INTP7	IF02.IFH	MK02.MKH
3	INTP2	IF03.IFL	MK03.MKL	35	INTP8	IF03.IFH	MK03.MKH
4	INTP3	IF04.IFL	MK04.MKL	36	INTP9	IF04.IFH	MK04.MKH
5	INTP4	IF05.IFL	MK05.MKL	37	INTP10	IF05.IFH	MK05.MKH
6	INTP5	IF06.IFL	MK06.MKL	38	INTP11	IF06.IFH	MK06.MKH
7	INTST2/INTSSPI20/ INTIIC20	IF07.IFL	MK07.MKL	39	保留	IF07.IFH	MK07.MKH
8	INTSR2/INTSSPI 21/INTIIC21	IF08.IFL	MK08.MKL	40	保留	IF08.IFH	MK08.MKH
9	INTSRE2	IF09.IFL	MK09.MKL	41	INTCOERR	IF09.IFH	MK09.MKH
10	INTST0/INTSSPI00/ INTIIC00	IF10.IFL	MK10.MKL	42	保留	IF10.IFH	MK10.MKH
11	INTSR0/INTSSPI 01/INTIIC01	IF11.IFL	MK11.MKL	43	保留	IF11.IFH	MK11.MKH
12	INTSRE0	IF12.IFL	MK12.MKL	44	INTTM01H	IF12.IFH	MK12.MKH
13	INTST1/INTSSPI 10/INTIIC10	IF13.IFL	MK13.MKL	45	保留	IF13.IFH	MK13.MKH
14	INTSR1/INTSSPI 11/INTIIC11	IF14.IFL	MK14.MKL	46	保留	IF14.IFH	MK14.MKH
15	INTSRE1	IF15.IFL	MK15.MKL	47	INTTM03H	IF15.IFH	MK15.MKH
16	INTIICA0	IF16.IFL	MK16.MKL	48	INTDIV	IF16.IFH	MK16.MKH
17	INTTM00	IF17.IFL	MK17.MKL	49	保留	IF17.IFH	MK17.MKH
18	INTTM01	IF18.IFL	MK18.MKL	50	保留	IF18.IFH	MK18.MKH
19	INTTM02	IF19.IFL	MK19.MKL	51	保留	IF19.IFH	MK19.MKH
20	INTTM03	IF20.IFL	MK20.MKL	52	保留	IF20.IFH	MK20.MKH
21	INTAD	IF21.IFL	MK21.MKL	53	保留	IF21.IFH	MK21.MKH
22	INTRTC	IF22.IFL	MK22.MKL	54	INTIT	IF22.IFH	MK22.MKH
23	INTKR	IF23.IFL	MK23.MKL	55	INTCOREC	IF23.IFH	MK23.MKH
24	INTCMP0	IF24.IFL	MK24.MKL	56	INTCOWUP	IF24.IFH	MK24.MKH
25	INTCMP1	IF25.IFL	MK25.MKL	57	INTCOTRX	IF25.IFH	MK25.MKH
26	INTTMA	IF26.IFL	MK26.MKL	58	保留	IF26.IFH	MK26.MKH
27	INTTMM0	IF27.IFL	MK27.MKL	59	保留	IF27.IFH	MK27.MKH
28	INTTMM1	IF28.IFL	MK28.MKL	60	保留	IF28.IFH	MK28.MKH
29	INTTMB	IF29.IFL	MK29.MKL	61	保留	IF29.IFH	MK29.MKH
30	INTTMC	IF30.IFL	MK30.MKL	62	保留	IF30.IFH	MK30.MKH
31	INTFL	IF31.IFL	MK31.MKL	63	保留	IF31.IFH	MK31.MKH

图25-4: 各标志寄存器与CPU.IRQ的关系



25.3.3 外部中断上升沿允许寄存器(EGP0、EGP1)、外部中断下降沿允许寄存器(EGN0、EGN1)

这些寄存器设置INTP0~INTP11的有效边沿。

通过8位存储器操作指令设置EGP0、EGP1、EGN0、EGN1寄存器。在产生复位信号后，这些寄存器的值变为“00H”。

图25-5：外部中断上升沿允许寄存器(EGP0、EGP1)和外部中断下降沿允许寄存器(EGN0、EGN1)的格式

地址：40045B38H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址：40045B39H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

地址：40045B3AH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

地址：40045B3BH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn引脚的有效边沿选择(n=0~11)
0	0	禁止检测边沿。
0	1	下降沿
1	0	上升沿
1	1	上升和下降的双边沿

对应EGPn位和EGNn位的端口如表25-3所示。

表25-3: 对应EGPn位和EGNn位的中断请求信号

检测允许位		中断请求信号	64引脚	52、48、40引脚	36引脚	32引脚
EGP0	EGN0	INTP0	○	○	○	○
EGP1	EGN1	INTP1	○	○	○	○
EGP2	EGN2	INTP2	○	○	○	○
EGP3	EGN3	INTP3	○	○	○	○
EGP4	EGN4	INTP4	○	○	○	○
EGP5	EGN5	INTP5	○	○	○	○
EGP6	EGN6	INTP6	○	○	○	○
EGP7	EGN7	INTP7	○	—	—	○
EGP8	EGN8	INTP8	○	○	○	○
EGP9	EGN9	INTP9	○	○	—	—
EGP10	EGN10	INTP10	○	○	○	○
EGP11	EGN11	INTP11	○	○	○	○

注意：如果将外部中断功能使用的输入端口切换到输出模式，就可能检测到有效边沿而产生INTPn中断。当切换到输出模式时，必须在禁止检测边沿后(EGPn、EGNn=0、0)将端口模式寄存器(PMxx)置“0”。

备注：

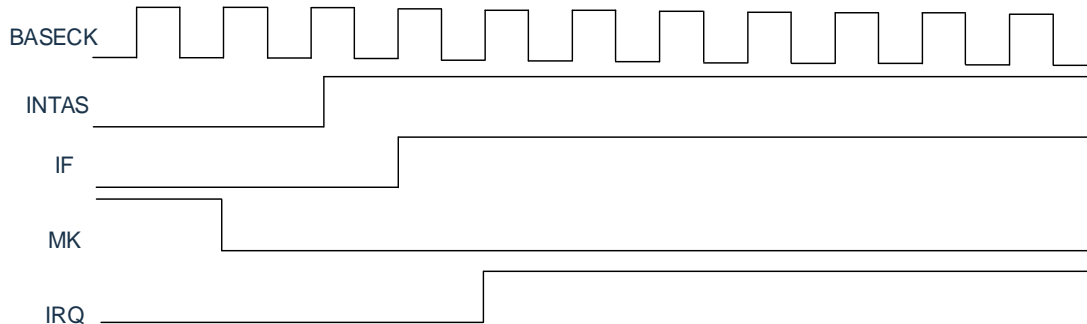
1. 有关边沿检测的端口，请参照“2.1端口功能”。
2. n=0~11

25.4 中断处理的操作

25.4.1 可屏蔽中断请求的接受

如果中断请求标志被置“1”并且该中断请求的屏蔽(MK)标志已被清“0”，就进入能接受可屏蔽中断请求的状态，可以将中断请求传递给NVIC。

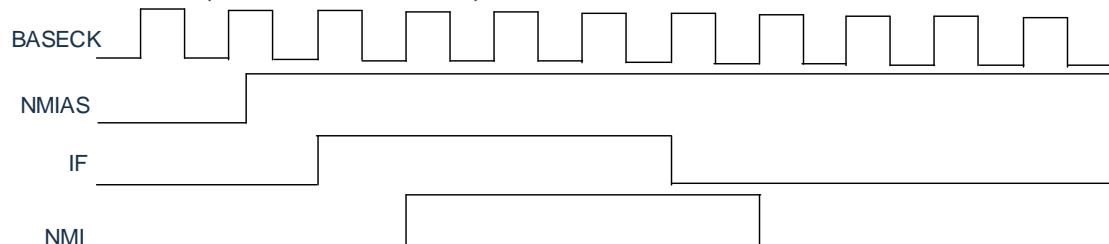
从中断请求标志被置1，到CPU的IRQ被置1，只需要1个时钟。



25.4.2 不可屏蔽中断请求的接受

如果产生不可屏蔽中断请求，中断请求标志将会被置“1”，并直接传递给NVIC。

从中断请求标志被置1，到CPU的NMI被置1，只需要1个时钟。



第26章 键中断功能

键中断输入的通道数因产品而不同。

26.1 键中断的功能

能通过给键中断输入引脚(KR0~KR7)输入下降沿, 产生键中断(INTKR)。

表26-1: 键中断检测引脚的分配

键中断引脚	键返回模式寄存器(KRM)
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

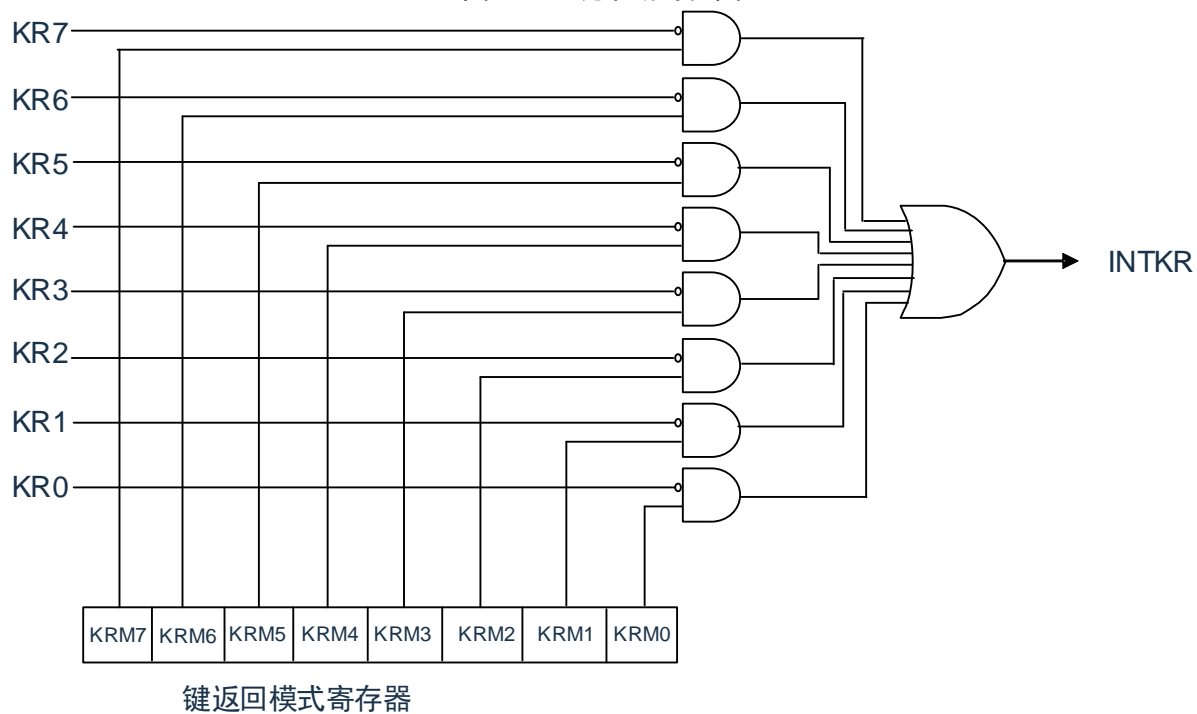
26.2 键中断的结构

键中断由以下硬件构成。

表26-2: 键中断的结构

项目	控制寄存器
控制寄存器	键返回模式寄存器(KRM) 端口模式寄存器(PMx)

图26-1: 键中断的框图



26.3 控制键中断的寄存器

通过以下寄存器控制键中断功能。

- 键返回模式寄存器(KRM)
- 端口模式寄存器(PMx)

26.3.1 键返回模式寄存器(KRM)

KRM0~KRM7位控制KR0~KR7信号。

通过8位存储器操作指令设置KRM寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图26-2：键返回模式寄存器(KRM)的格式

地址：40044B37H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	键中断模式的控制
0	不检测键中断信号。
1	检测键中断信号。

注意：

1. 能通过将键中断输入引脚的上拉电阻寄存器(PUx)的对象位置“1”，使用内部上拉电阻。
2. 如果在给键中断输入引脚输入低电平的状态下将KRM寄存器的对象位置位，就产生中断。要想忽视此中断时，必须在通过中断屏蔽标志禁止中断处理后设置KRM寄存器。然后，必须在等待键中断输入的低电平宽度(tKR)(请参照数据手册)后清除中断请求标志，允许中断处理。
3. 在键中断模式中未使用的引脚能用作通常的端口。

备注：n=0~7

26.3.2 端口模式寄存器(PMx)

当用作键中断输入引脚(KR0~KR7)时，必须将PMxn位分别置“1”。此时，Pxn的输出锁存器可以是“0”或者“1”。

通过8位存储器操作指令设置PMx寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

能通过上拉电阻选择寄存器(PUx)以位为单位使用内部上拉电阻。

端口模式寄存器的格式请参考“2.3.1 端口模式寄存器(PMxx)”。

第27章 待机功能

27.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下2种模式。

(1) 睡眠模式

睡眠模式是停止CPU运行时钟的模式。在设置睡眠模式前，如果高速系统时钟振荡电路、高速内部振荡器或者副系统时钟振荡电路正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到深度睡眠模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

(2) 深度睡眠模式

深度睡眠模式是停止高速系统时钟振荡电路和高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小CPU的工作电流。

因为深度睡眠模式能通过中断请求来解除，所以也能进行间歇运行。但是，在X1时钟的情况下，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设置为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注意：

1. 只有在CPU以主系统时钟运行时才能使用深度睡眠模式。当CPU以副系统时钟运行时，不能设置为深度睡眠模式。无论CPU是以主系统时钟还是以副系统时钟运行，都能使用睡眠模式。
2. 在转移到深度睡眠模式时，必须在停止以主系统时钟运行的外围硬件后执行WFI指令。
3. 为了减小A/D转换器的工作电流，必须将A/D转换器模式寄存器0(ADM0)的bit7(ADCS)和bit0(ADCE)清“0”，在停止A/D转换运行后执行WFI指令。
4. 能通过选项字节选择在睡眠模式或者深度睡眠模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“第33章选项字节”。

27.2 睡眠模式

27.2.1 睡眠模式的设置

在SCR寄存器的SLEEPDEEP位为0时，执行WFI指令，就进入了睡眠模式。在睡眠模式，CPU停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，发振器等在睡眠模式下的状态见表27-1。

无论设置前的CPU时钟是高速系统时钟还是高速内部振荡器时钟或者副系统时钟，都能设置睡眠模式。

注意：当中断屏蔽标志为“0”(允许中断处理)并且中断请求标志为“1”(产生中断请求信号)时，中断请求信号用于解除睡眠模式。因此，即使在此情况下执行WFI令，也不转移到睡眠模式。

表27-1: 睡眠模式中的运行状态(1/2)

睡眠模式的设置		在CPU以主系统时钟运行的过程中执行WFI指令的情况		
		CPU以高速内部振荡器时钟(F _{IH})运行	CPU以X1时钟(F _X)运行	CPU以外部主系统时钟(F _{EX})运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	F _{IH}	继续运行(不能停止)。	禁止运行。	
	F _X	禁止运行。	继续运行(不能停止)。	不能运行。
	F _{EX}		不能运行。	继续运行(不能停止)。
副系统时钟	F _{XT}	保持设置为睡眠模式前的状态。		
	F _{EXS}			
低速内部振荡器时钟	F _{IL}	通过选项字节(000C0H)的bit0(WDSTBYON)和bit4(WDTON)以及副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位进行设置。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存		停止运行。		
RAM		停止运行(在执行DMA时, 能运行)。		
端口(锁存器)		保持设置为睡眠模式前的状态。		
DIV		能运行。		
Timer4				
实时时钟(RTC)				
15位间隔定时器				
看门狗定时器				
定时器A		能运行。		
定时器M				
定时器B				
定时器C				
时钟输出/蜂鸣器输出				
A/D转换器				
D/A转换器				
比较器				
通用串行通信单元(SCI)				
串行接口(IICA)				
aFCAN				
数据传送控制器(DMA)				
联动控制器				
上电复位功能		能运行。		
电压检测功能				
外部中断				
键中断功能				
CRC运算功能	高速CRC	在RAM区的运算中执行DMA时, 能运行。		
	通用CRC			
RAM奇偶校验功能		在执行DMA时, 能运行。		
SFR保护功能				

备注: 停止运行: 在转移到睡眠模式时自动停止运行。

禁止运行: 在转移到睡眠模式前停止运行。

F_{IH}: 高速内部振荡器时钟 F_{IL}: 低速内部振荡器时钟

F_X: X1时钟 F_{EX}: 外部主系统时

F_{XT}: XT1时钟 F_{EXS}: 外部副系统

表27-1: 睡眠模式中的运行状态(2/2)

睡眠模式的设置		在CPU以副系统时钟运行的过程中执行WFI指令的情况	
		CPU以XT1时钟(F _{XT})运行	CPU以外部副系统时钟(F _{EXS})运行
系统时钟		停止给CPU提供时钟。	
主系统时钟	F _{IH}	禁止运行。	
	F _X		
	F _{EX}		
副系统时钟	F _{XT}	继续运行(不能停止)。	不能运行。
	F _{EXS}	不能运行。	继续运行(不能停止)。
低速内部振荡器时钟	F _{IL}	通过选项字节(000C0H)的bit0(WDSTBYON)和bit4(WDTON)以及副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位进行设置。 <ul style="list-style-type: none"> • WUTMMCK0=1: 振荡 • WUTMMCK0=0并且WDTON=0: 停止 • WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 • WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止 	
CPU		停止运行。	
代码闪存		停止运行。	
RAM		停止运行(在执行DMA时, 能运行)。	
端口(锁存器)		保持设置为睡眠模式前的状态。	
DIV		当RTCLPC=0时, 能运行(否则禁止运行)。	
Time4		当RTCLPC=0时, 能运行(否则禁止运行)。	
实时时钟(RTC)		能运行。	
15位间隔定时器		能运行。	
看门狗定时器		参照“第14章看门狗定时器”。	
定时器A		当RTCLPC=0时, 能运行(否则禁止运行)。	
定时器M			
定时器B			
定时器C			
时钟输出/蜂鸣器输出			
A/D转换器		禁止运行。	
D/A转换器		当RTCLPC=0时, 能运行(否则禁止运行)。	
比较器		能运行。	
通用串行通信单元(SCI)		当RTCLPC=0时, 能运行(否则禁止运行)。	
串行接口(IICA)		禁止运行。	
aFCAN		当RTCLPC=0时, 能运行(否则禁止运行)。	
数据传送控制器(DMA)		当RTCLPC=0时, 能运行(否则禁止运行)。	
联动控制器		能在可运行的功能块之间进行链接。	
上电复位功能		能运行。	
电压检测功能			
外部中断			
键中断功能			
CRC运算功能	高速CRC	禁止运行。	
	通用CRC	在RAM区的运算中执行DMA时, 能运行。	
RAM奇偶校验错误检测功能		在执行DMA时, 能运行。	
SFR保护功能			

备注: 停止运行: 在转移到睡眠模式时自动停止运行。

禁止运行: 在转移到睡眠模式前停止运行。

F_{IH}: 高速内部振荡器时钟 F_{IL}: 低速内部振荡器时钟

F_X: X1时钟 F_{EX}: 外部主系统时

F_{XT}: XT1时钟 F_{EXS}: 外部副系统

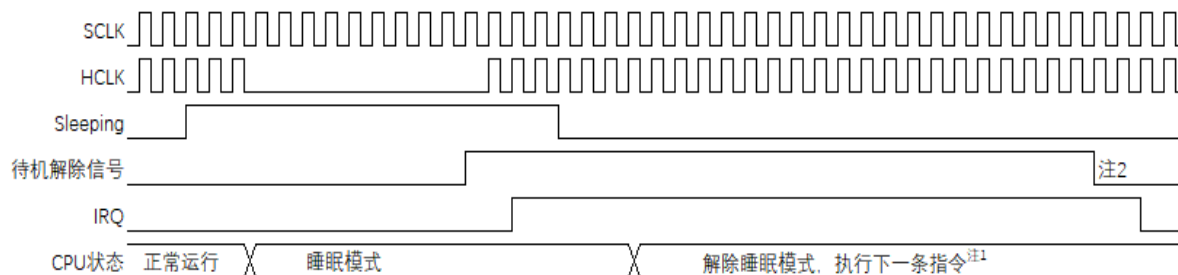
27.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位端子，POR复位，低电压检测复位，RAM奇偶校验错误复位，WDT复位，软件复位解除。

(1) 通过中断解除

当产生一个未屏蔽的中断，且处于允许接受中断的状态时，睡眠模式就被解除，CPU开始处理中断服务程序。

图27-1：通过中断请求解除睡眠模式



注1：从待机解除信号产生到睡眠模式解除，开始中断服务程序，需要16个时钟。

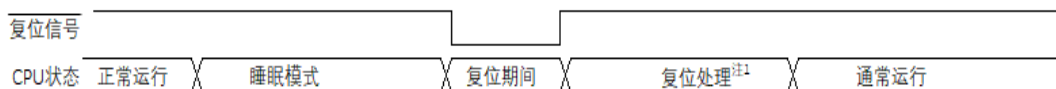
注2：待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

注意：进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。

(2) 通过复位解除

当有复位信号产生时，CPU处于复位状态，睡眠模式被解除。和通常的复位相同，在转移到复位向量地址后执行程序。

图27-2：通过复位解除睡眠模式



注1：有关复位处理时间，请参照“第28章复位功能”。有关上电复位(POR)电路和电压检测(LVD)电路的复位处理时间，请参照“第29章上电复位电路”。

27.3 深度睡眠模式

27.3.1 深度睡眠模式的设置

在SCR寄存器的SLEEPDEEP位为1时，执行WFI指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及发振器都停止运行。但是，CPU内部寄存器的值，RAM数据，周边模块，I/O的状态被保持。周边模块，发振器在深度睡眠模式的运行状态见表27-2。

只有在设置前的CPU时钟为主系统时钟的情况下才能设置深度睡眠模式。

注意：当中断屏蔽标志为“0”(允许中断处理)并且中断请求标志为“1”(产生中断请求信号)时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行WFI指令，就在一旦进入深度睡眠模式后立即被解除。在执行WFI指令并且经过深度睡眠模式解除时间后返回到运行模式。

表27-2: 深度睡眠模式中的运行状态

深度睡眠模式的设置		在CPU以主系统时钟运行的过程中执行WFI指令的情况		
		CPU以高速内部振荡器时钟(F _{IH})运行	CPU以X1时钟(F _X)运行	CPU以外部主系统时钟(F _{EX})运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	F _{IH}	停止		
	F _X			
F _{EX}				
副系统时钟	F _{XT}	保持设置为深度睡眠模式前的状态。		
	F _{EXS}			
F _{IL}		通过选项字节(000C0H)的bit0(WDSTBYON)和bit4(WDTON)以及副系统时钟提供模式控制寄存器(OSMC)的WUTMMCK0位进行设置。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存		停止运行。		
RAM		保持设置为深度睡眠模式前的状态。		
端口(锁存器)		保持设置为深度睡眠模式前的状态。		
DIV		禁止运行。		
定时器阵列单元		禁止运行。		
实时时钟(RTC)		能运行。		
15位间隔定时器		能运行。		
看门狗定时器		参照“第14章看门狗定时器”。		
定时器A		<ul style="list-style-type: none"> 在选择没有TAIO输入滤波器时的事件计数模式中, 能运行。 在选择副系统时钟作为计数源并且OSMC寄存器的RTCLPC位为“0”时, 能运行。 在选择低速内部振荡器作为计数源时, 能运行。 上述以外: 禁止运行。 		
定时器M		禁止运行。		
定时器B		禁止运行。		
定时器C		禁止运行。		
时钟输出/蜂鸣器输出		在选择副系统时钟作为计数时钟并且RTCLPC位为“0”时, 能运行(否则禁止运行)。		
A/D转换器		能进行唤醒。		
D/A转换器		能运行(保持设置为深度睡眠模式前的状态)。		
比较器		能运行(只限于不使用数字滤波器的情况)。		
通用串行通信单元(SCI)		只有SSPIp和UARTq才能唤醒。除了SSPIp和UARTq以外, 禁止运行。		
串行阵列单元(IICA)		能通过地址匹配进行唤醒。		
aFCAN		禁止运行。		
数据传送控制器(DMA)		能接受DMA启动源。		
联动控制器		能在可运行的功能块之间进行链接。		
上电复位功能		能运行。		
电压检测功能				
外部中断				
键中断功能				
CRC运算功能	高速CRC	停止运行。		
	通用CRC			
RAM奇偶校验功能		停止运行。		
SFR保护功能		停止运行。		

备注：停止运行：在转移到深度睡眠模式时自动停止运行。

禁止运行：在转移到深度睡眠模式前停止运行。

F_{IH}：高速内部振荡器时钟 F_{IL}：低速内部振荡器时钟

F_X：X1时钟 F_{EX}：外部主系统时钟

F_{XT}：XT1时钟 F_{EXS}：外部副系统时钟

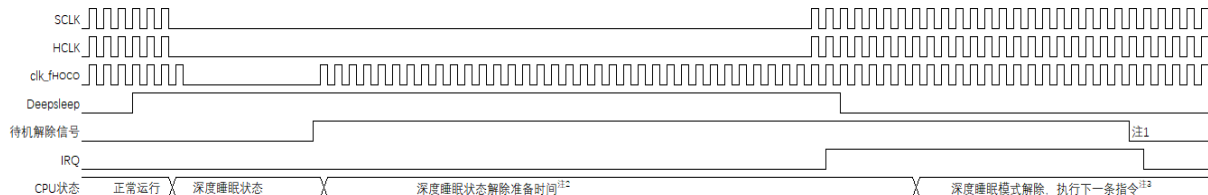
27.3.2 深度睡眠模式的解除

能通过以下2种方法解除深度睡眠模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除深度睡眠模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图27-3：通过中断请求解除深度睡眠模式



注1：待机解除信号：有关待机解除信号的详细内容，请参照“图25-1：中断功能的基本结构”。

注2：深度睡眠状态解除准备时间：

进入深度睡眠模式前CPU时钟为高速内部振荡时钟或外部时钟输入时：

至少20us

进入深度睡眠模式前CPU时钟为高速系统时钟(X1振荡)时：

至少20us与振荡稳定时间(通过OSTS进行设定)中较长的时间

注3：等待：从CPU.IRQ有效到开始执行中断服务程序，需要14个时钟。

注意：

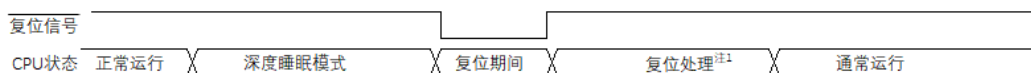
1. 进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。
2. 在CPU以高速系统时钟(X1振荡)运行并且要缩短深度睡眠模式解除后的振荡稳定时间时，必须在执行WFI指令前暂时将CPU时钟切换为高速内部振荡器时钟。

备注：高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

(b) 通过产生复位信号进行的解除

通过产生复位信号来解除深度睡眠模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图27-4：通过复位解除深度睡眠模式



注：有关复位处理时间，请参照“第28章 复位功能”。有关上电复位(POR)电路和电压检测(LVD)电路的复位处理时间，请参照“第29章 上电复位电路”。

第28章 复位功能

以下7种方法产生复位信号。

- (1) 通过RESETB引脚输入外部复位。
- (2) 通过看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5) 因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为1而产生内部复位。
- (6) 因RAM奇偶校验错误而产生内部复位。
- (7) 因存取非法存储器而产生内部复位。

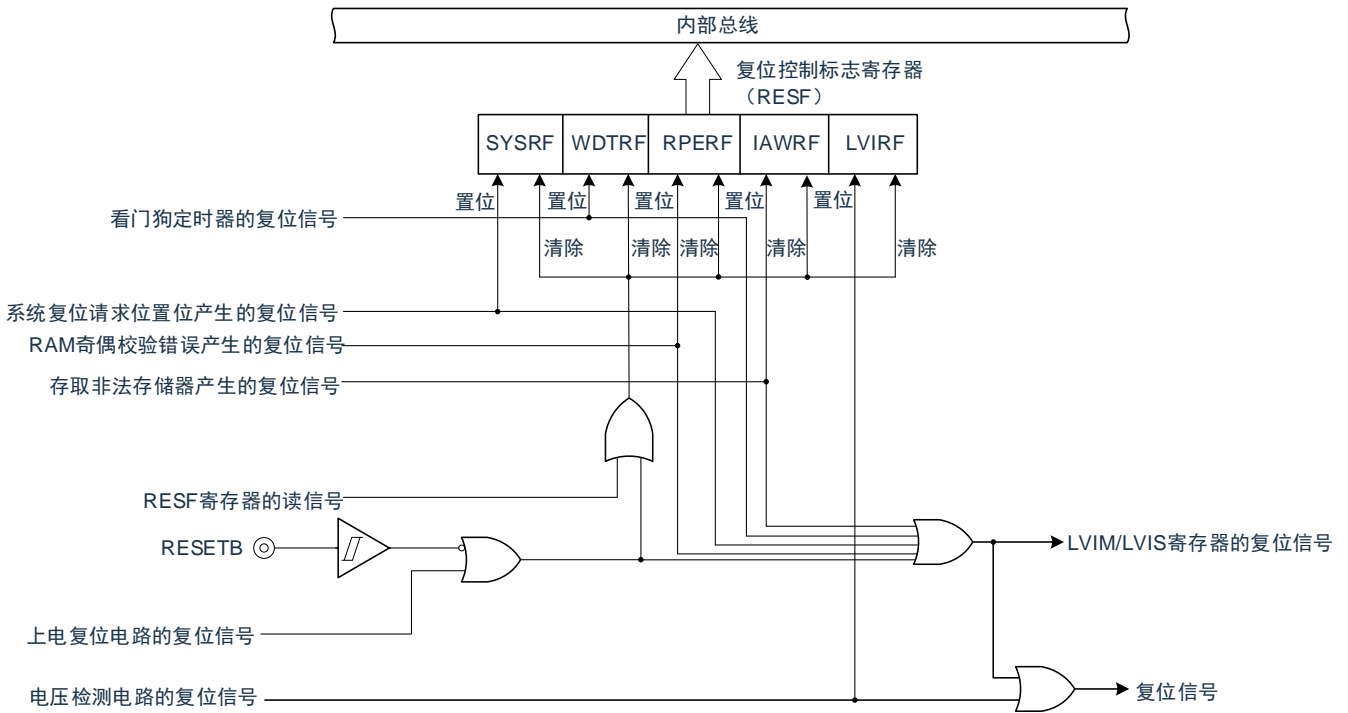
内部复位和外部复位相同，在产生复位信号后，从用户自定义的程序起始地址开始执行程序。

当给RESETB引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到POR电路和LVD电路的电压，或者系统复位请求位被置位，或者发生RAM奇偶检验错误，或者存取非法存储器时，产生复位并且各硬件变为如表28-1所示的状态。

注意：

1. 在进行外部复位时，必须至少给RESETB引脚输入10us的低电平。如果在电源电压上升时进行外部复位，就必须在给RESETB引脚输入低电平后接通电源，而且在datasheet的AC特性所示的工作电压范围内至少保持10us的低电平，然后输入高电平。
2. 在复位信号发生期间，停止X1时钟、XT1时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟和外部副系统时钟的输入无效。
3. 如果发生复位，就对各SFR进行初始化，因此端口引脚变为以下状态：
 - P40、P137：在外部复位或者POR复位期间为高阻态，复位解除后为高电平(连接内部上拉电阻)。
 - P130：在复位期间以及解除复位后输出低电平。
 - P40、P130及P137以外的端口：在复位期间以及解除复位后为高阻态。

图28-1：复位功能的框图



注意：LVD电路的内部复位不会对LVD电路进行复位。

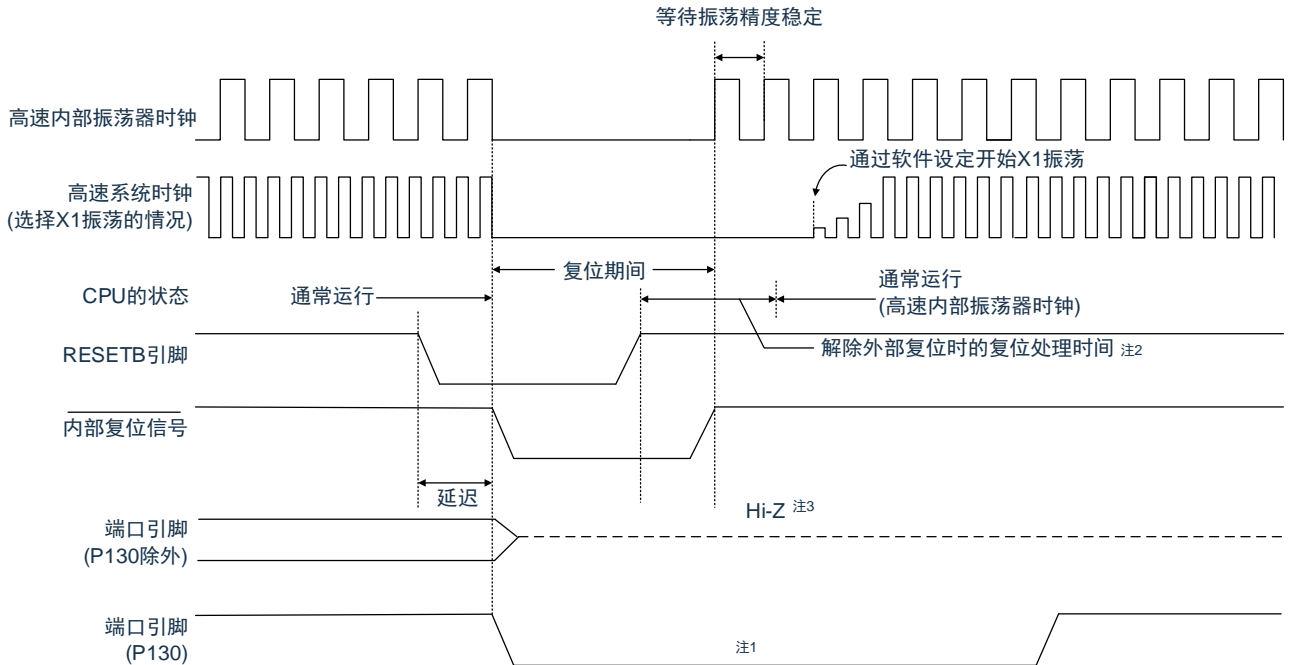
备注：

1. LVIM：电压检测寄存器
2. LVIS：电压检测电平寄存器

复位时序

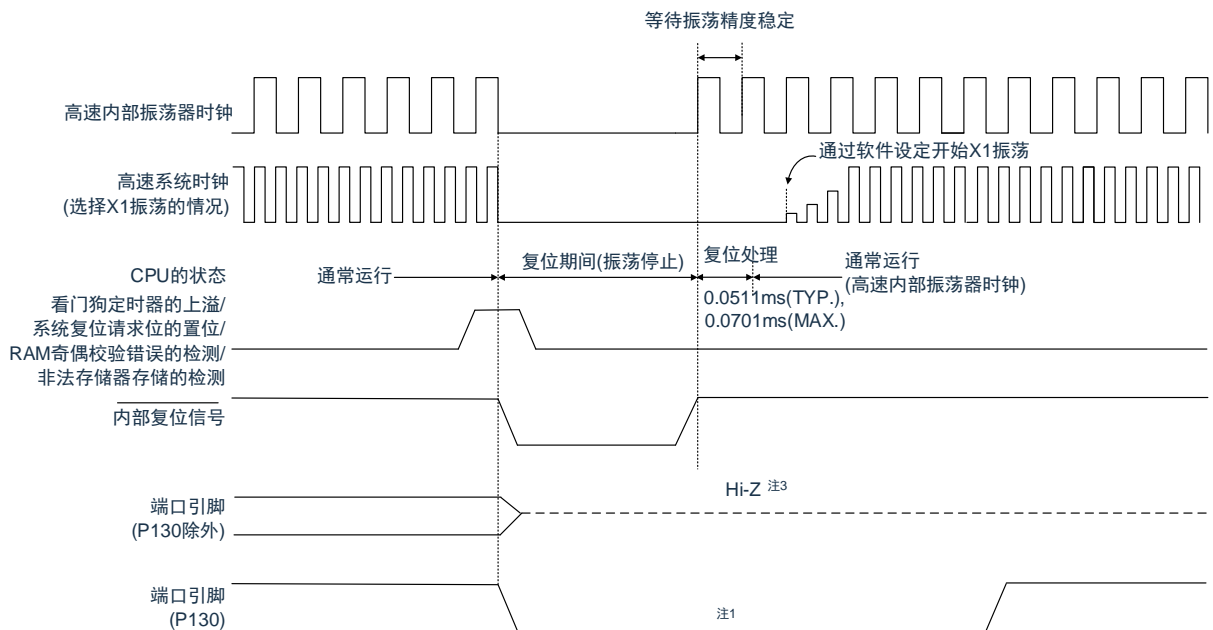
当给RESETB引脚输入低电平时，产生复位。然后，如果给RESETB引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图28-2: RESETB输入的复位时序



对于因看门狗定时器的上溢、系统复位请求位的置位、RAM奇偶校验错误的检测或者非法存储器存取检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图28-3: 因看门狗定时器的上溢、系统复位请求位的置位、RAM奇偶校验错误的检测或者非法存储器存取的检测而产生的复位时序



注1: 如果发生复位, P130就输出低电平。因此, 如果在发生复位前将P130设置为高电平输出, 就能将P130的输出作为外部器件的复位信号进行虚拟输出。要解除外部器件的复位信号时, 必须通过软件将P130设置为高电平输出。

注2: 解除外部复位时的复位处理时间:

解除POR后的第1次: 0.672ms(TYP.)、0.832ms(MAX.)(使用LVD的情况)

0.399ms(TYP.)、0.519ms(MAX.)(不使用LVD的情况)解除POR后的第2次以后: 0.531ms(TYP.)、0.675ms(MAX.)(使用LVD的情况)

0.259ms(TYP.)、0.362ms(MAX.)(不使用LVD的情况)

当电源电压上升时, 在解除外部复位时的复位处理时间之前需要电压稳定等待时间0.99ms(TYP.)、2.30ms(MAX.)。

注3: 端口引脚P40变为以下状态:

- 在外部复位或者POR复位的期间为高阻抗。
- 在其他复位期间以及接受复位后为高电平(连接内部上拉电阻)。

注意: 看门狗定时器也不例外, 在发生内部复位时进行复位。

对于由POR电路和LVD电路的电压检测产生的复位, 如果在复位后满足 $V_{DD} \geq V_{POR}$ 或者 $V_{DD} \geq V_{LVD}$, 就解除复位状态, 并且在复位处理后以高速内部振荡器时钟开始执行程序。详细内容请参照“第29章 上电复位电路”和“第30章 电压检测电路”。

备注: V_{POR} : POR电源电压上升检测电压

V_{LVD} : LVD检测电压

表28-1：复位期间的运行状态

项目		复位期间
系统时钟		停止给CPU提供时钟。
主系统时钟	F _{IH}	停止运行。
	F _X	停止运行(X1引脚和X2引脚处于输入端口模式)。
	F _{EX}	时钟输入无效(引脚处于输入端口模式)。
副系统时钟	F _{XT}	能运行。
	F _{EXS}	时钟输入无效(引脚处于输入端口模式)。
F _{IL}		停止运行。
CPU		
代码闪存		停止运行。
RAM		停止运行。
端口(锁存器)		高阻抗 ^{注1}
DIV		停止运行。
定时器阵列单元		
定时器A		
定时器M		
定时器B		
定时器C		
实时时钟(RTC)		在POR复位时，停止运行。在其他复位时，能运行。
15位间隔定时器		停止运行。
看门狗定时器		
时钟输出/蜂鸣器输出		
A/D转换器		
D/A转换器 ^{注1}		
比较器 ^{注1}		
通用串行通信单元(SCI)		
串行接口(IICA)		
aFCAN		
数据传送控制器(DMA)		
上电复位功能		能进行检测运行。
电压检测功能		在LVD复位时，能运行。在其他复位时，停止运行。
外部中断		停止运行。
键中断功能		
CRC运算功能	高速CRC	
	通用CRC	
RAM奇偶校验功能		
SFR保护功能		

注1：端口引脚P40、P130变为以下状态：

- P40：在外部复位或者POR复位的期间为高阻抗。在其他复位期间为高电平(连接内部上拉电阻)。
- P130：在复位期间输出低电平。

备注：F_{IH}：高速内部振荡器时钟 F_X：X1振荡时钟

F_{EX}：外部主系统时钟 F_{XT}：XT1振荡时钟

F_{EXS}：外部副系统时钟 F_{IL}：低速内部振荡器时钟

28.1 确认复位源的寄存器

28.1.1 复位控制标志寄存器(RESF)

BAT32A237微控制器存在多种内部复位发生源。复位控制标志寄存器(RESF)保存发生复位请求的复位源。能通过8位存储器操作指令读RESF寄存器。

通过RESETB的输入、上电复位(POR)电路的复位和RESF寄存器的读取，清除SYSRF、WDTRF、RPERF、IAWRF、LVIRF标志。要判断复位源时，必须将RESF寄存器的值保存到任意的RAM，然后通过其RAM值进行判断。

图28-4：复位控制标志寄存器(RESF)的格式

地址：40020440H 复位后：不定值^{注1} R

符号	7	6	5	4	3	2	1	0
RESF	SYSRF	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

SYSRF	系统复位请求位被置位而产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

WDTRF	看门狗定时器(WDT)产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

RPERF	RAM奇偶校验错误产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

IAWRF	存取非法存储器产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

LVIRF	电压检测电路(LVD)产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

注1：因复位源而不同。请参照表28-2。

注意：在允许产生RAM奇偶校验错误复位(RPERDIS=0)的情况下，当存取数据时，必须对“所用RAM区”进行初始化；当从RAM区执行指令时，必须对“所用RAM区+10字节”的区域进行初始化。通过产生复位，进入允许产生RAM奇偶校验错误复位(RPERDIS=0)的状态。详细内容请参照“31.3.3 RAM奇偶校验错误检测功能”。

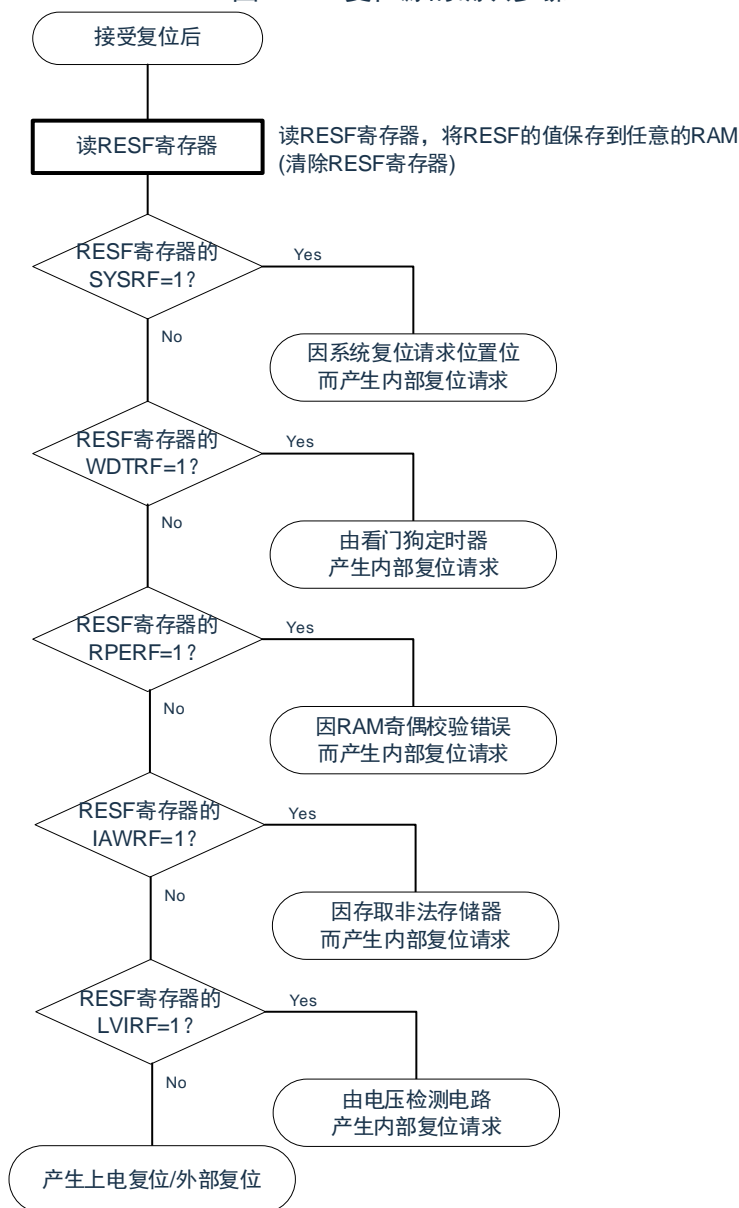
发生复位请求时的RESF寄存器状态如表28-2所示。

表28-2：发生复位请求时的RESF寄存器状态

标志 \ 复位源	RESETB 输入	POR产生的复位	系统复位请求位置位产生的复位	WDT产生的复位	RAM奇偶校验错误产生的复位	存取非法存储器产生的复位	LVD产生的复位
SYSRF	清“0”	清“0”	置“1”	保持	保持	保持	保持
WDTRF			保持	置“1”	保持		
RPERF				置“1”	置“1”		
IAWRF				保持		置“1”	
LVIRF			保持	保持	置“1”		

复位源的确认步骤如图28-5所示。

图28-5：复位源的确认步骤



注意：上述流程是确认步骤的例子。

第29章 上电复位电路

29.1 上电复位电路的功能

上电复位电路(POR)有以下功能。

- 在接通电源时产生内部复位信号。
如果电源电压(V_{DD})超过检测电压(V_{POR}), 就解除复位。但是, 必须在电源电压达到数据手册的AC特性所示的工作电压范围前, 通过电压检测电路或者外部复位保持复位状态。
- 将电源电压(V_{DD})和检测电压(V_{PDR})进行比较。当 $V_{DD} < V_{PDR}$ 时, 产生内部复位信号。但是, 当电源电压下降时, 必须在电源电压低于数据手册的AC特性所示的工作电压范围前, 通过深度睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

注意: 当上电复位电路产生内部复位信号时, 将复位控制标志寄存器(RESF)清“00H”。

备注:

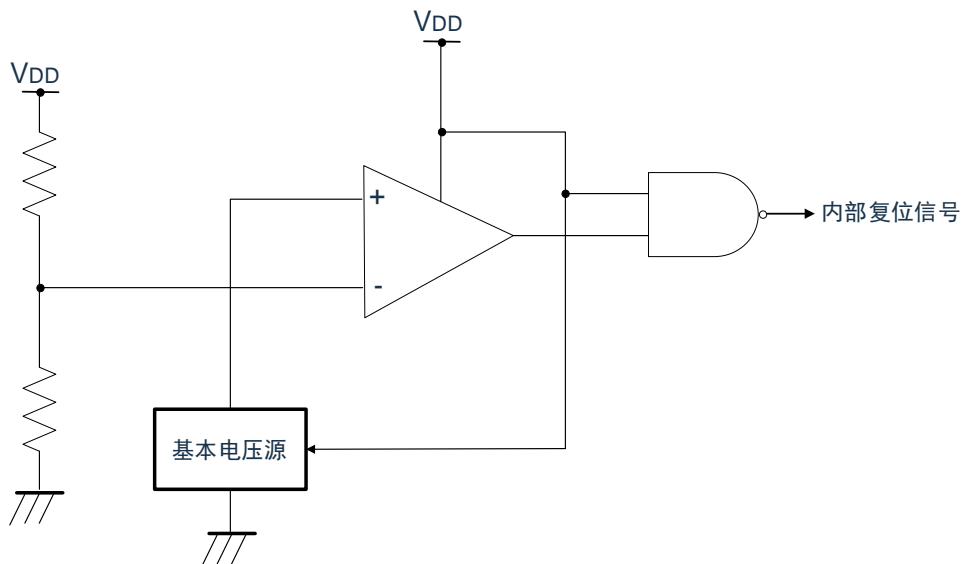
1. BAT32A237内置多个产生内部复位信号的硬件。当由看门狗定时器(WDT)、电压检测(LVD)电路、系统复位请求位置位、RAM奇偶校验错误或者非法存储器的存取而产生内部复位信号时, 用于表示复位源的标志分配在RESF寄存器; 当由WDT、LVD、系统复位请求位的置位、RAM奇偶校验错误或者非法存储器的存取而产生内部复位信号时, 不将RESF寄存器清“00H”而将标志置“1”。有关RESF寄存器的详细内容, 请参照“第28章 复位功能”。
2. V_{POR} : POR电源电压上升检测电压
 V_{PDR} : POR电源电压下降检测电压

详细内容请参照数据手册的POR电路特性。

29.2 上电复位电路的结构

上电复位电路的框图如图29-1所示。

图29-1：上电复位电路的框图

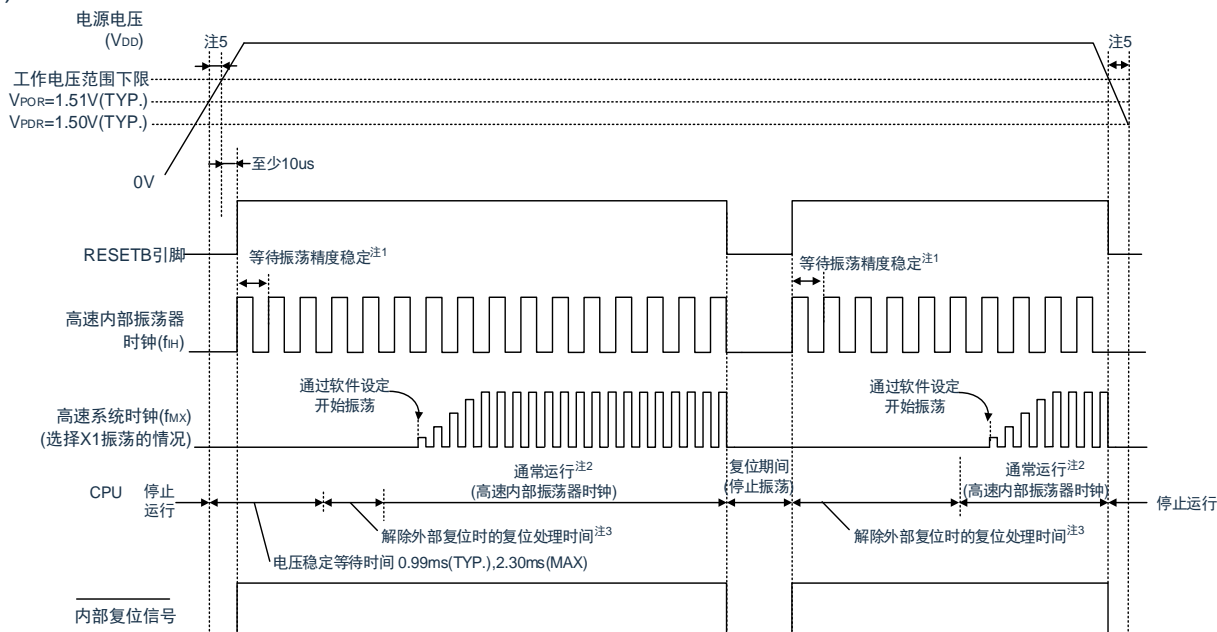


29.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图29-2：上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用RESETB引脚的外部复位输入的情况



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器(OSTC)确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注3：到开始通常运行为止的时间除了达到V_{POR}(1.51V(TYP.))后的“电压稳定等待时间”以外，在将RESETB信号置为高电平(“1”)后还需要以下的“解除外部复位时的复位处理时间(解除POR后的第1次)”。解除外部复位时的复位处理时间如下所示：

解除POR后的第1次 : 0.672ms(TYP.)、0.832ms(MAX.)(使用LVD的情况)
0.399ms(TYP.)、0.519ms(MAX.)(不使用LVD的情况)

注4：解除POR后第2次以后的解除外部复位时的复位处理时间如下所示：

解除POR后的第2次以后：0.531ms(TYP.)、0.675ms(MAX.)(使用LVD的情况)
0.259ms(TYP.)、0.362ms(MAX.)(不使用LVD的情况)

注5：当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

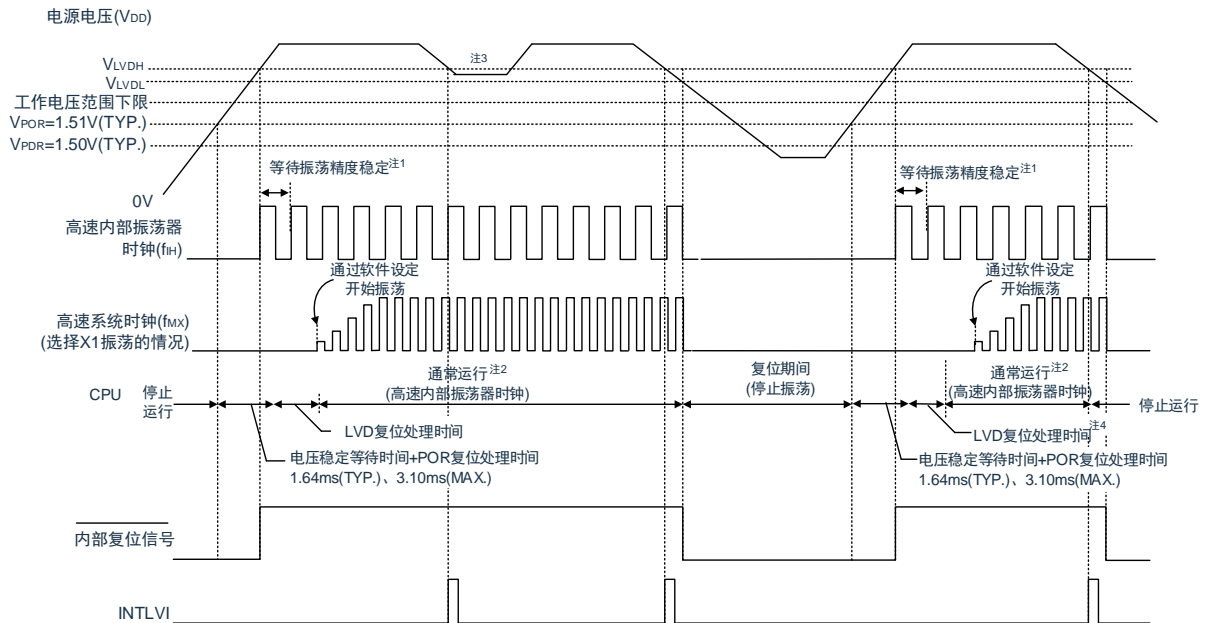
备注：V_{POR}：POR电源电压上升检测电压

V_{PDR}：POR电源电压下降检测电压

注意：在LVD为OFF时，必须使用RESETB引脚的外部复位。详细内容请参照“第30章 电压检测电路”。

图29-2：上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD为中断&复位模式的情况(选项字节000C1H的LVIMDS1、LVIMDS0=1、0)

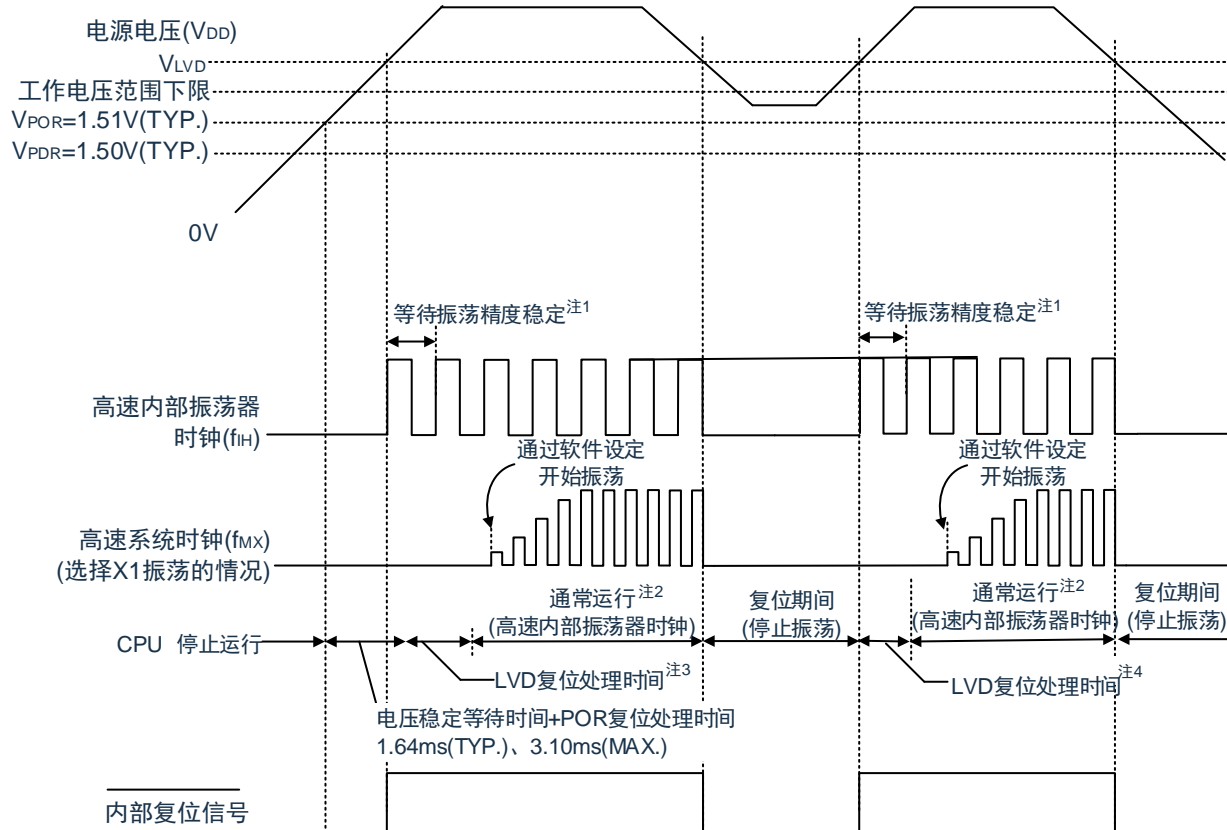


- 注1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
 - 注2: 能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下, 必须在通过振荡稳定时间计数器的状态寄存器(OSTC)确认振荡稳定时间后进行切换; 在使用XT1时钟的情况下, 必须在利用定时器功能等确认振荡稳定时间后进行切换。
 - 注3: 在产生中断请求信号(INTLVI)后, 自动将电压检测电平寄存器(LVIS)的LVILV位和LVIMD位置“1”。因此, 必须考虑可能出现电源电压在不低于低电压检测电压(V_{LVDL})的状态下恢复到高电压检测电压(V_{LVDH})或者更高的情况, 在产生INTLVI后按照“图30-7: 工作电压的确认/复位的设置步骤”和“图30-8: 中断&复位模式的初始设置步骤”进行设置。
 - 注4: 到开始通常运行为止的时间除了达到V_{POR}(1.51V(TYP.))后的“电压稳定等待时间+POR复位处理时间”以外, 在达到LVD检测电平(V_{LVDH})后还需要以下的“LVD复位处理时间”。
- LVD复位处理时间: 0ms~0.0701ms(MAX.)

备注: V_{LVDH}、V_{LVDL}: LVD检测电压
 V_{POR}: POR电源电压上升检测电压
 V_{PDR}: POR电源电压下降检测电压

图29-2：上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3) LVD复位模式的情况(选项字节000C1H的LVIMDS1、LVIMDS0=1、1)



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器(OSTC)确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注3：到开始通常运行为止的时间除了达到V_{POR}(1.51V(TYP.))后的“电压稳定等待时间+POR复位处理时间”以外，在达到LVD检测电平(V_{LVD})后还需要以下的“LVD复位处理时间”。

LVD复位处理时间：0ms~0.0701ms(MAX.)

注4：在电源电压下降时，如果只在发生电压检测电路(LVD)的内部复位后恢复电源电压，就在达到LVD检测电平(V_{LVD})后需要以下的“LVD复位处理时间”。

LVD复位处理时间：0.0511ms(TYP.)、0.0701ms(MAX.)

备注：

1. V_{LVDH}、V_{LVDL}：LVD检测电压
V_{POR}：POR电源上升检测电压
V_{PDR}：POR电源下降检测电压
2. 当选择LVD中断模式(选项字节000C1H的LVIMD1、LVIMD0=0、1)时，从接通电源到开始通常运行的时间和“图29-2(3)LVD位模式的情况”的“注3”的时间相同。

第30章 电压检测电路

30.1 电压检测电路的功能

电压检测电路通过选项字节(000C1H)设置运行模式和检测电压(V_{LVDH} 、 V_{LVDL} 、 V_{LVD})。电压检测(LVD)电路有以下功能。

- 将电源电压(V_{DD})和检测电压(V_{LVDH} 、 V_{LVDL} 、 V_{LVD})进行比较,产生内部复位或者内部中断信号。
- 电源电压的检测电压(V_{LVDH} 、 V_{LVDL})能通过选项字节选择12种检测电平(参照“第33章 选项字节”)。
- 也能在深度睡眠模式中运行。
- 当电源电压上升时,必须在电源电压达到数据手册的AC特性所示的工作电压范围前,通过电压检测电路或者外部复位保持复位状态;当电源电压下降时,必须在电源电压低于工作电压范围前,通过深度睡眠模式的转移、电压检测电路或者外部复位,置为复位状态。工作电压范围取决于用户选项字节(000C2H/010C2H)的设置。

(a) 中断&复位模式(选项字节的LVIMDS1、LVIMDS0=1、0)

通过选项字节000C1H选择2个检测电压(V_{LVDH} 、 V_{LVDL}),高电压检测电平(V_{LVDH})用于解除复位或者产生中断,低电压检测电平(V_{LVDL})用于产生复位。

(b) 复位模式(选项字节的LVIMDS1、LVIMDS0=1、1)

将选项字节000C1H选择的1个检测电压(V_{LVD})用于产生或者解除复位。

(c) 中断模式(选项字节的LVIMDS1、LVIMDS0=0、1)

将选项字节000C1H选择的1个检测电压(V_{LVD})用于产生中断或者解除复位。在各模式中,产生以下的中断信号和内部复位信号。

中断&复位模式 (LVIMDS1、LVIMDS0=1、0)	复位模式 (LVIMDS1、LVIMDS0=1、1)	中断模式 (LVIMDS1、LVIMDS0=0、1)
在工作电压下降时,当检测到 $V_{DD} < V_{LVDH}$ 时,产生中断请求信号;当检测到 $V_{DD} < V_{LVDL}$ 时,产生内部复位;当检测到 $V_{DD} \geq V_{LVDH}$ 时,解除内部复位。	当检测到 $V_{DD} \geq V_{LVD}$ 时,解除内部复位;当检测到 $V_{DD} < V_{LVD}$ 时,产生内部复位。	在发生复位后,LVD的内部复位状态继续保持到 $V_{DD} \geq V_{LVD}$ 为止。当检测到 $V_{DD} \geq V_{LVD}$ 时,解除LVD的内部复位。在解除LVD的内部复位后,如果检测到 $V_{DD} < V_{LVD}$ 或者 $V_{DD} \geq V_{LVD}$ 时,就产生中断请求信号(INTLVI)。

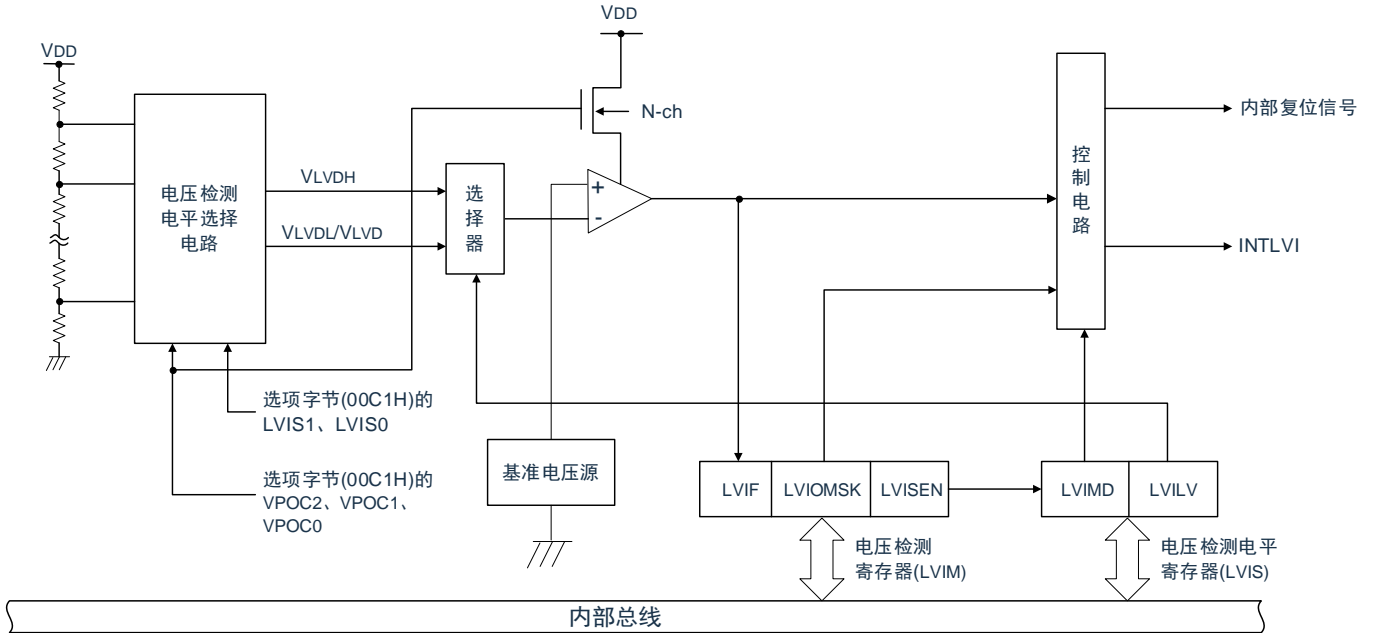
在电压检测电路运行时,能通过读电压检测标志(LVIF:电压检测寄存器(LVIM)的bit0)来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位,就将复位控制标志寄存器(RESF)的bit0(LVIRF)置“1”。有关RESF寄存器的详细内容,请参照“第28章 复位功能”。

30.2 电压检测电路的结构

电压检测电路的框图如图30-1所示。

图30-1：电压检测电路的框图



30.3 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器(LVIM)
- 电压检测电平寄存器(LVIS)

30.3.1 电压检测寄存器(LVIM)

此寄存器设置允许或者禁止改写电压检测电平寄存器(LVIS)，并且确认LVD输出的屏蔽状态。通过8位存储器操作指令设置LVIM寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图30-2: 电压检测寄存器(LVIM)的格式

地址: 40020441H	复位后: 00H ^{注1}	R/W ^{注2}						
	7	6	5	4	3	2	1	0
LVIM	LVISEN ^{注3}	0	0	0	0	0	LVIOMSK	LVIF

LVISEN ^{注3}	电压检测电平寄存器(LVIS)的允许/禁止改写的设置
0	禁止改写LVIS寄存器(LVIOMSK=0(LVD输出屏蔽无效))。
1	允许改写LVIS寄存器(LVIOMSK=1(LVD输出屏蔽有效))。

LVIOMSK	LVD输出的屏蔽状态标志
0	LVD输出屏蔽无效。
1	LVD输出屏蔽有效 ^{注4} 。

LVIF	电压检测标志
0	电源电压(V _{DD}) ≥ 检测电压(V _{LVD})或者LVD为OFF。
1	电源电压(V _{DD}) < 检测电压(V _{LVD})

注1: 复位值因复位源而变。

在LVD发生复位时，不对LVIM寄存器的值进行复位而保持原来的值；在其他复位时，将LVISEN清“0”。

注2: bit0和bit1是只读位。

注3: 只有在选择中断&复位模式(选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”)时才能设置，在其他模式中不能更改初始值。

注4: 只有在选择中断&复位模式(选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”)时，LVIOMSK位才在以下期间自动变为“1”，屏蔽LVD产生的复位或者中断。

- LVISEN=1的期间
- 从发生LVD中断开始到LVD检测电压稳定为止的等待时间
- 从更改LVILV位的值到LVD检测电压稳定为止的等待时间

30.3.2 电压检测电平寄存器(LVIS)

这是设置电压检测电平的寄存器。

通过8位存储器操作指令设置LVIS寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”^{注1}。

图30-3: 电压检测电平寄存器(LVIS)的格式

地址: 40020442H 复位后: 00H/01H/81H^{注1} R/W

	7	6	5	4	3	2	1	0
LVIS	LVIMD ^{注2}	0	0	0	0	0	0	LVILV ^{注2}

LVIMD ^{注2}	电压检测的运行模式
0	中断模式
1	复位模式

LVILV ^{注2}	LVD检测电平
0	高电压检测电平(VLVDH)
1	低电压检测电平(VLVDL或者VLVD)

注1: 复位值因复位源和选项字节的设置而变。在发生LVD复位时，不将此寄存器清“00H”。

在发生LVD以外的复位时，此寄存器的值如下：

- 选项字节的LVIMDS1、LVIMDS0=1、0时：00H
- 选项字节的LVIMDS1、LVIMDS0=1、1时：81H
- 选项字节的LVIMDS1、LVIMDS0=0、1时：01H

注2: 只有在选择中断&复位模式(选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”)时才能写“0”。在其他情况下不能设置。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注意:

1. 要改写LVIS寄存器时，必须按照图30-7和图30-8的步骤进行。
2. 通过选项字节000C1H选择LVD的运行模式和各模式的检测电压(VLVDH、VLVDL、VLVD)。用户选项字节(000C1H/010C1H)的格式如表30-1所示。有关选项字节的详细内容，请参照“第33章 选项字节”。

表30-1: 用户选项字节(000C1H/010C1H)的格式(1/2)

 地址: 000C1H/010C1H^注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置(中断&复位模式)

检测电压			选项字节的设置值						
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降	下降						LVIMDS1	LVIMDS0
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V		0	1	1	0		
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V		1	0	1	0		
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V		1	1	1	0		
3.02V	2.96V		0			1			
4.06V	3.98V		0			0			
—			禁止设置上述以外的值。						

• LVD的设置(复位模式)

检测电压		选项字节的设置值									
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置				
上升	下降						LVIMDS1	LVIMDS0			
1.67V	1.63V	0	0	0	1	1	1	1			
1.77V	1.73V		0	0	1	0					
1.88V	1.84V		0	1	1	1					
1.98V	1.94V		0	1	1	0					
2.09V	2.04V		0	1	0	1					
2.50V	2.45V		1	0	1	1					
2.61V	2.55V		1	0	1	0					
2.71V	2.65V		1	0	0	1					
2.81V	2.75V		1	1	1	1					
2.92V	2.86V		1	1	1	0					
3.02V	2.96V		1	1	0	1					
3.13V	3.06V		0	1	0	0					
3.75V	3.67V		1	0	0	0					
4.06V	3.98V		1	1	0	0					
—			禁止设置上述以外的值。								

备注:

1. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
2. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

表30-1: 用户选项字节(000C1H)的格式(2/2)

地址: 000C1H

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置(中断模式)

检测电压		选项字节的设置值								
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置			
上升	下降						LVIMDS1	LVIMDS0		
1.67V	1.63V	0	0	0	1	1	0	1		
1.77V	1.73V		0	0	1	0				
1.88V	1.84V		0	1	1	1				
1.98V	1.94V		0	1	1	0				
2.09V	2.04V		0	1	0	1				
2.50V	2.45V		1	0	1	1				
2.61V	2.55V		1	0	1	0				
2.71V	2.65V		1	0	0	1				
2.81V	2.75V		1	1	1	1				
2.92V	2.86V		1	1	1	0				
3.02V	2.96V		1	1	0	1				
3.13V	3.06V		0	1	0	0				
3.75V	3.67V		1	0	0	0				
4.06V	3.98V		1	1	0	0				
—			禁止设置上述以外的值。							

• LVD为OFF(使用RESETB引脚的外部复位)

检测电压		选项字节的设置值						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设置上述以外的值。						

注意:

1. 必须给bit4写“1”。
2. 当电源电压上升时, 必须在电源电压达到数据手册的AC特性所示的工作电压范围前, 通过电压检测电路或者外部复位保持复位状态; 当电源电压下降时, 必须在电源电压低于工作电压范围前, 通过深度睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。
工作电压范围取决于用户选项字节(000C2H)的设置。

备注:

1. ×: 忽略
2. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
3. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

30.4 电压检测电路的运行

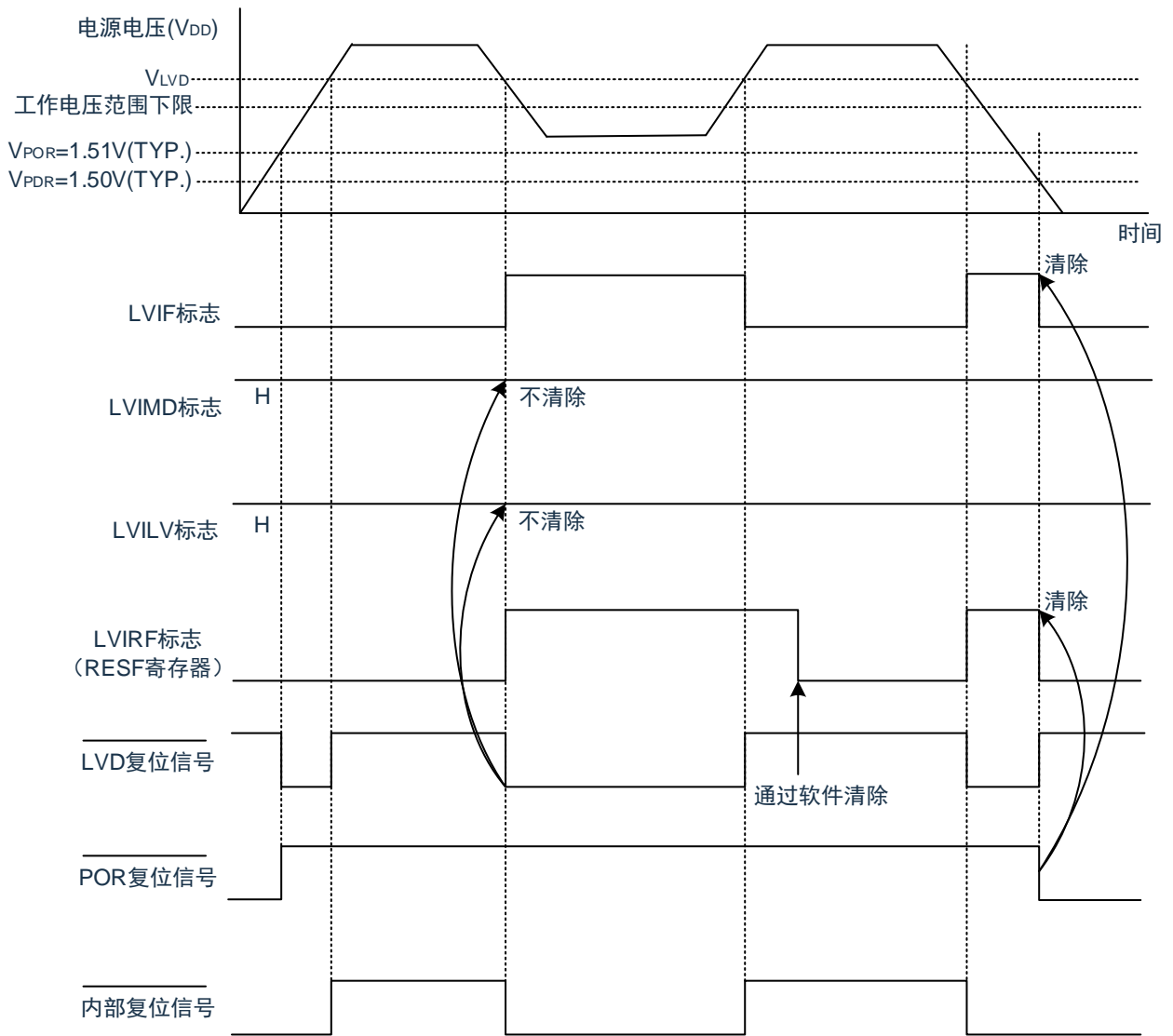
30.4.1 用作复位模式时的设置

通过选项字节000C1H设置运行模式(复位模式(LVIMDS1、LVIMDS0=1、1))和检测电压(V_{LVD})。如果设置复位模式,就在以下初始设置的状态下开始运行。

- 将电压检测寄存器(LVIM)的bit7(LVISEN)置“0”(禁止改写电压检测电平寄存器(LVIS))。
- 将电压检测电平寄存器(LVIS)的初始值置“81H”。
bit7(LVIMD)为“1”(复位模式)。
bit0(LVILV)为“1”(电压检测电平: V_{LVD})。
- LVD复位模式的运行
当接通电源时,复位模式(选项字节的LVIMDS1、LVIMDS0=1、1)在电源电压(V_{DD})超过电压检测电平(V_{LVD})前保持LVD的内部复位状态。如果电源电压(V_{DD})超过电压检测电平(V_{LVD}),就解除内部复位。
当工作电压下降时,如果电源电压(V_{DD})低于电压检测电平(V_{LVD}),就产生LVD的内部复位。

LVD复位模式的内部复位信号的产生时序如图30-4所示。

图30-4: 内部复位信号的产生时序(选项字节的LVIMDS1、LVIMDS0=1、1)



备注: V_{POR}: POR电源电压上升检测电压
 V_{PDR}: POR电源电压下降检测电压

30.4.2 用作中断模式时的设置

通过选项字节000C1H设置运行模式(中断模式(LVIMDS1、LVIMDS0=0、1))和检测电压(V_{LVD})。如果设置中断模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器(LVIM)的bit7(LVISEN)置“0”(禁止改写电压检测电平寄存器(LVIS))。
- 将电压检测电平寄存器(LVIS)的初始值置“01H”。

bit7(LVIMD)为“0”(中断模式)。

bit0(LVILV)为“1”(电压检测电平： V_{LVD})。

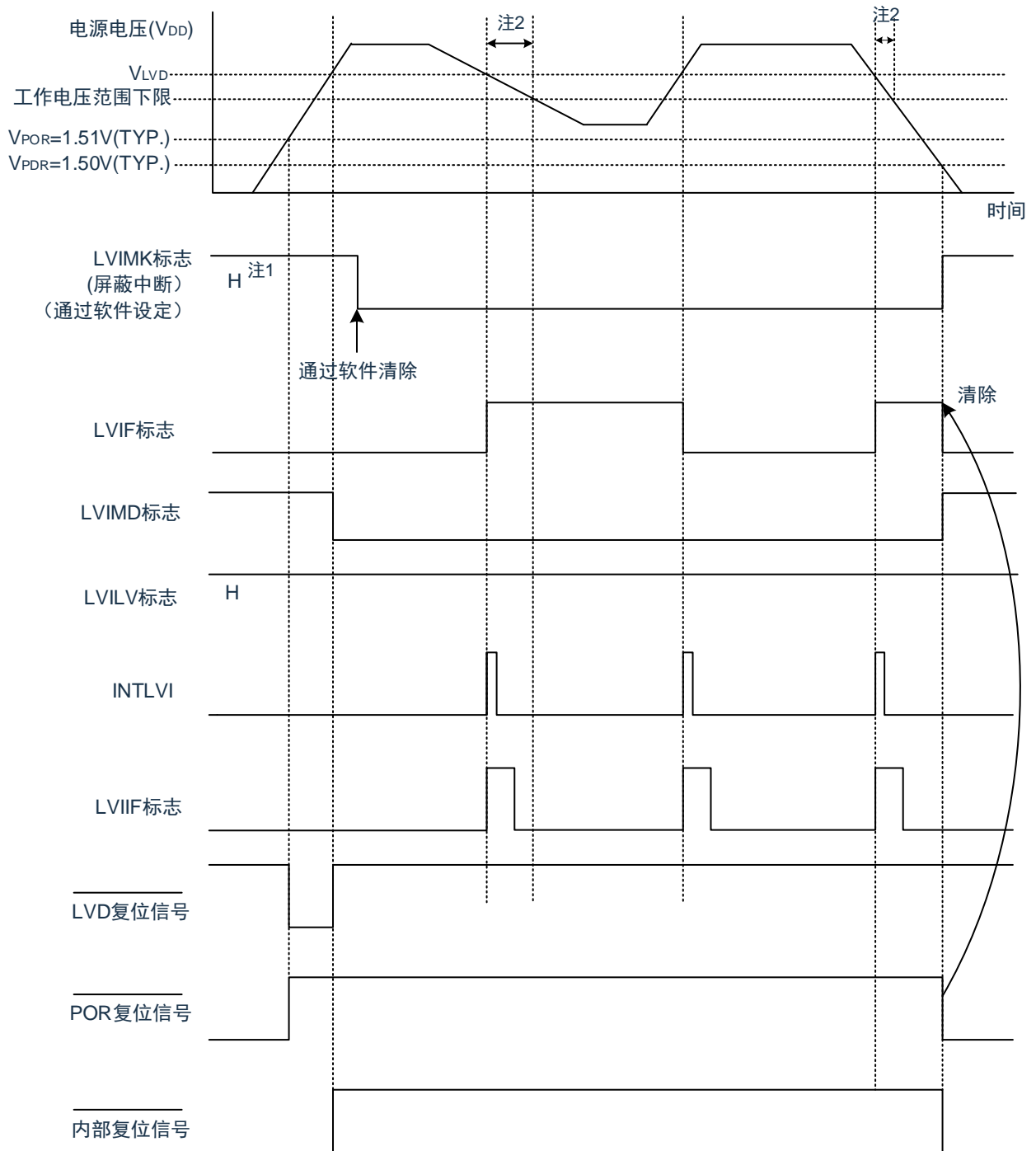
- LVD中断模式的运行

在产生复位后，中断模式(选项字节的LVIMDS1、LVIMDS0=0、1)在电源电压(V_{DD})超过电压检测电平(V_{LVD})前保持LVD的内部复位状态。如果电源电压(V_{DD})超过电压检测电平(V_{LVD})，就解除LVD的内部复位。

在解除LVD的内部复位后，如果电源电压(V_{DD})超过电压检测电平(V_{LVD})，就产生LVD的中断请求信号(INTLVI)。当工作电压下降时，必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD中断模式的中断请求信号的产生时序如图30-5所示。

图30-5：中断信号的产生时序(选项字节的LVIMDS1、LVIMDS0=0、1)



注1：在产生复位信号后，LVIMK标志变为“1”。

注2：当工作电压下降时，必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注：V_{POR}：POR电源电压上升检测电压

V_{PDR}：POR电源电压下降检测电压

30.4.3 用作中断&复位模式时的设置

通过选项字节000C1H设置运行模式(中断&复位模式(LVIMDS1、LVIMDS0=1、0))和检测电压(V_{LVDH} 、 V_{LVDL})。

如果设置中断&复位模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器(LVIM)的bit7(LVISEN)置“0”(禁止改写电压检测电平寄存器(LVIS))。
- 将电压检测电平寄存器(LVIS)的初始值置“00H”。
 - bit7(LVIMD)为“0”(中断模式)。
 - bit0(LVILV)为“0”(高电压检测电平： V_{LVDH})。

● LVD中断&复位模式的运行

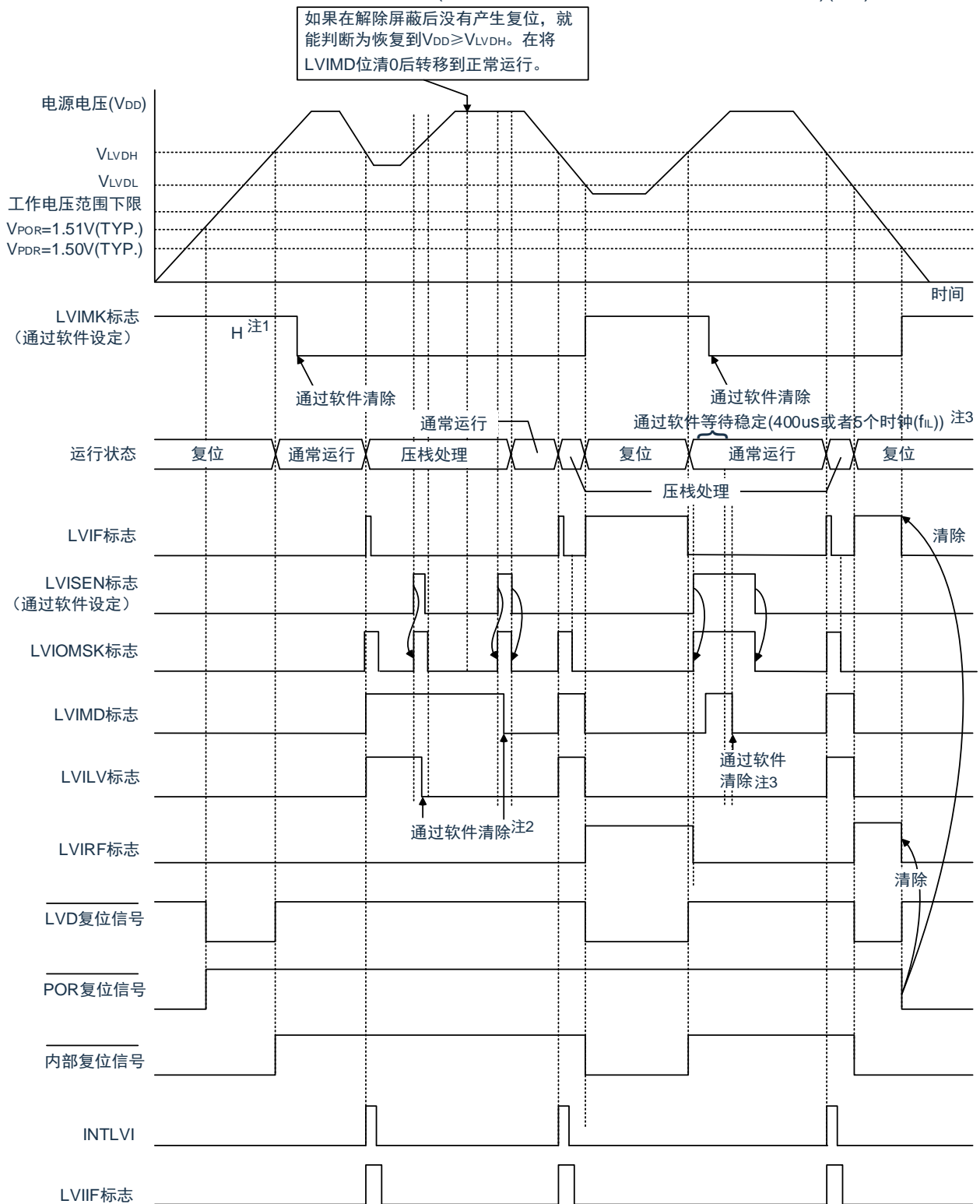
当接通电源时，中断&复位模式(选项字节的LVIMDS1、LVIMDS0=1、0)在电源电压(V_{DD})超过高电压检测电平(V_{LVDH})前保持LVD的内部复位状态。如果电源电压(V_{DD})超过高电压检测电平(V_{LVDH})，就解除内部复位。

当工作电压下降时，如果电源电压(V_{DD})低于高电压检测电平(V_{LVDH})，就产生LVD的中断请求信号(INTLVI)并且能进行任意的压栈处理。此后，如果电源电压(V_{DD})低于低电压检测电平(V_{LVDL})，就产生LVD的内部复位。但是，在发生INTLVI后，即使电源电压(V_{DD})在不低于低电压检测电压(V_{LVDL})的状态下恢复到高电压检测电压(V_{LVDH})或者更高，也不产生中断请求信号。

当使用LVD中断&复位模式时，必须按照“图30-7：工作电压的确认/复位的设置步骤”和“图30-8：中断&复位模式的初始设置步骤”所示的流程图的步骤进行设置。

LVD中断&复位模式的内部复位信号和中断信号的产生时序如图30-6所示。

图30-6: 复位&中断信号的产生时序(选项字节的LVIMDS1、LVIMDS0=1、0)(1/2)



注1: 在产生复位信号后, LVIMK标志变为“1”。

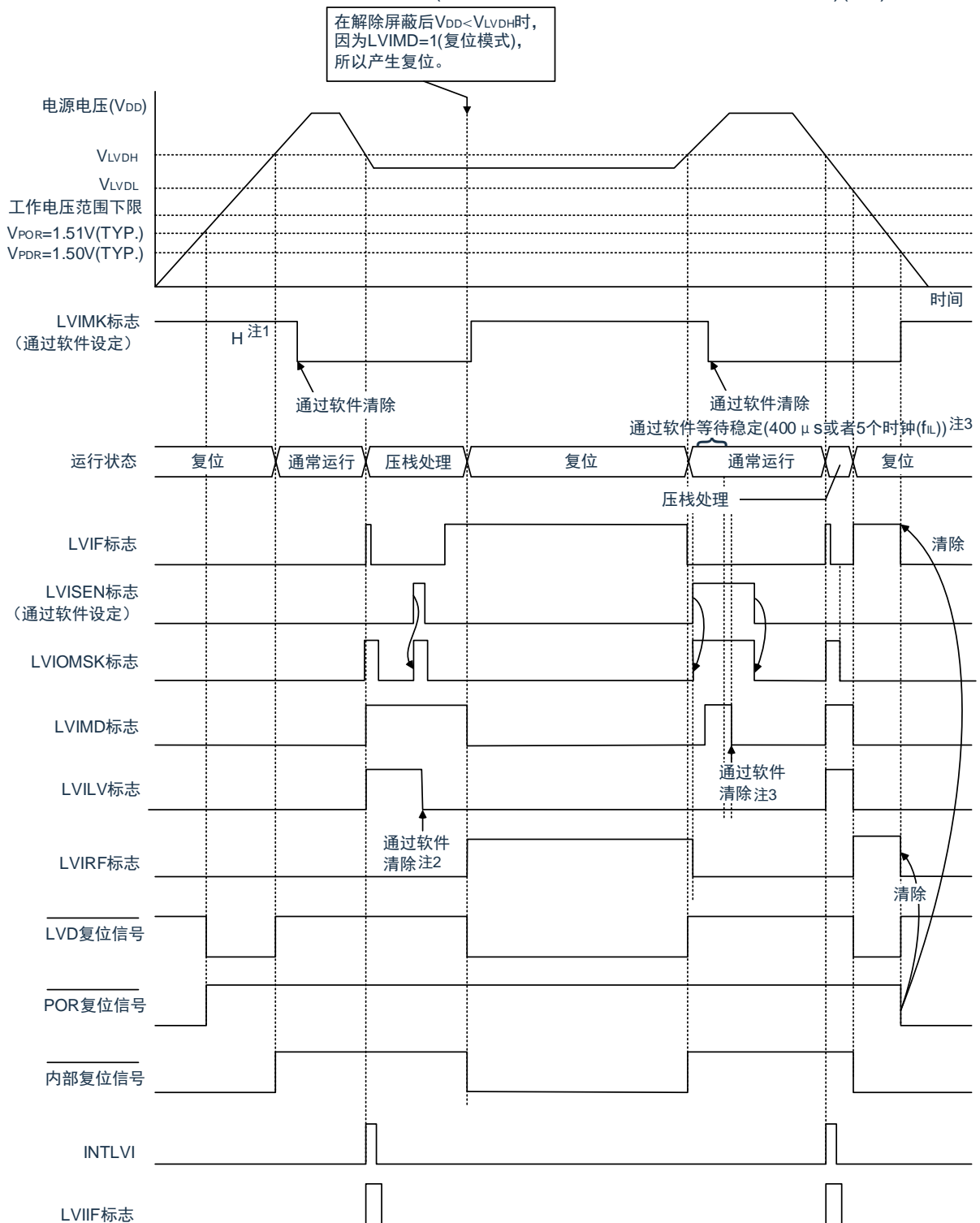
注2: 当使用中断&复位模式时, 必须在发生中断后按照“图30-7: 工作电压的确认/复位的设置步骤”进行设置。

注3: 当使用中&复位模式时, 必须在解除复位后按照“图30-8: 中断&复位模式的初始设置步骤”进行设置。

备注: V_{POR} : POR电源电压上升检测电压

V_{PDR} : POR电源电压下降检测电压

图30-6: 中断&复位信号的产生时序(选项字节的LVIMDS1、LVIMDS0=1、0)(2/2)



注1: 在产生复位信号后, LVIMK标志变为“1”。

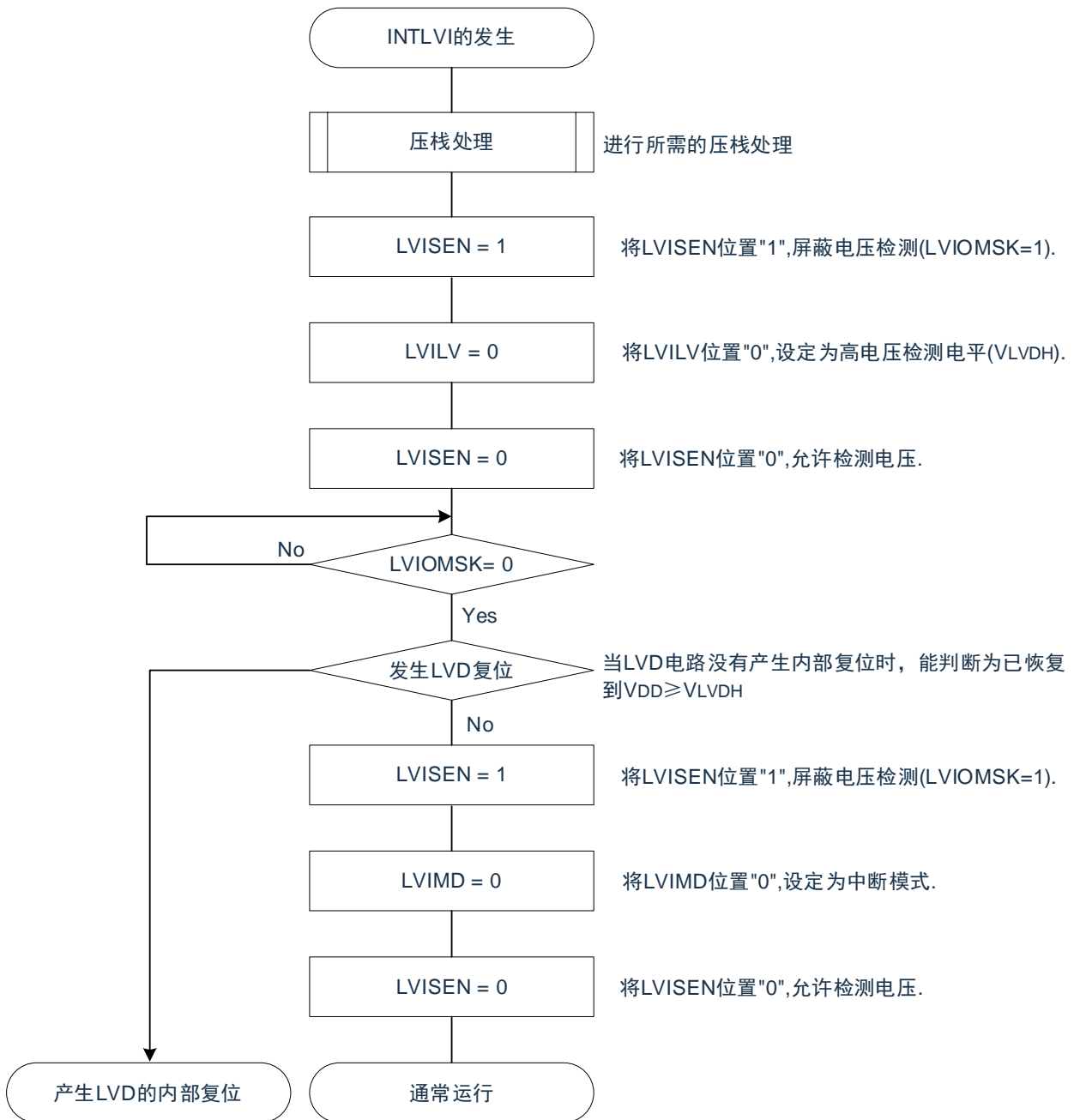
注2: 当使用中断&复位模式时, 必须在发生中断后按照“图30-7: 工作电压的确认/复位的设置步骤”进行设置。

注3: 当使用中断&复位模式时, 必须在解除复位后按照“图30-8: 中断&复位模式的初始设置步骤”进行设置。

备注: V_{POR} : POR电源电压上升检测电压

V_{PDR} : POR电源电压下降检测电压

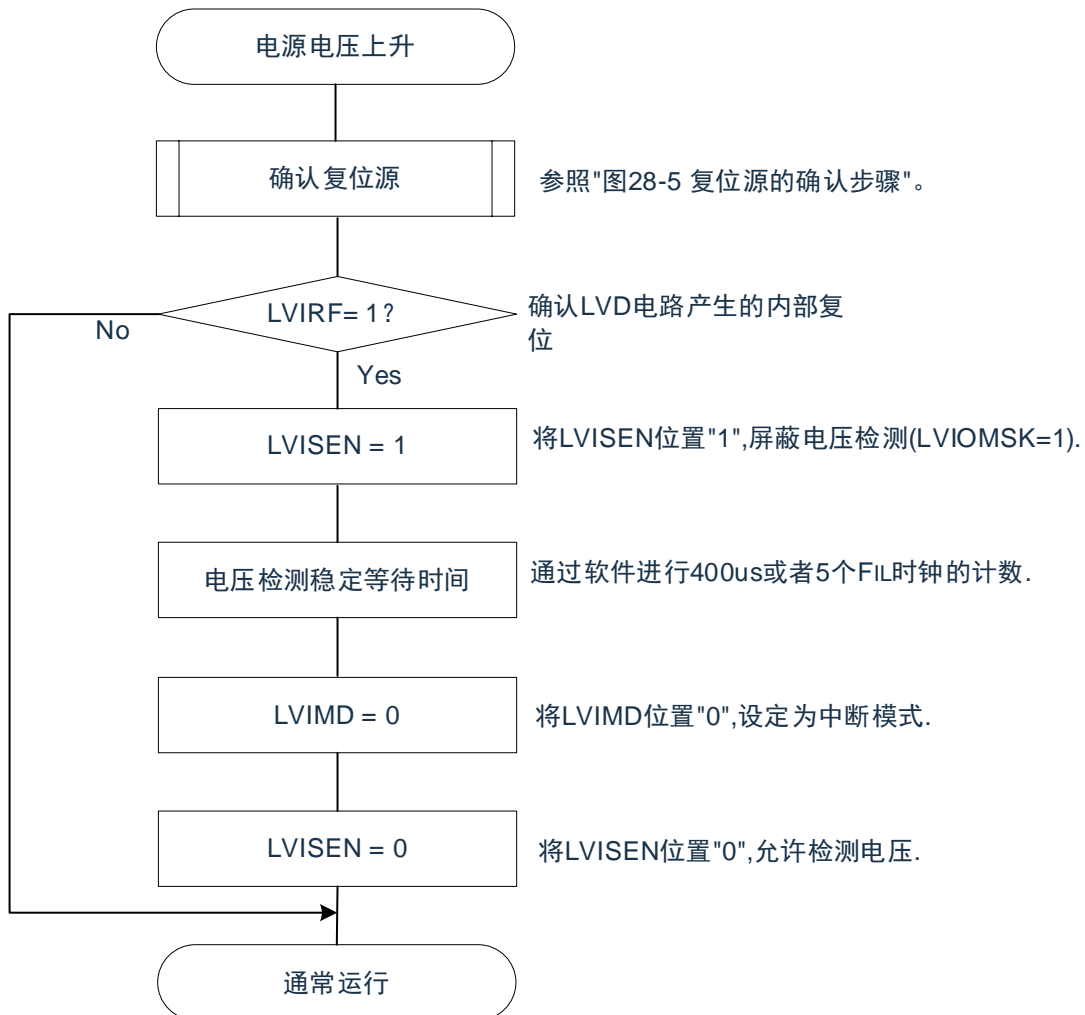
图30-7: 工作电压的确认/复位的设置步骤



如果设置中断&复位模式(LVIMDS1、LVIMDS0=1、0)，就在解除LVD复位(LVIRF=1)后需要400us或者5个FIL时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将LVIMD位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写LVIMD位时，必须将LVISEN位置“1”，屏蔽LVD产生的复位或者中断的产生。

中断&复位模式的初始设置步骤如图30-8所示。

图30-8：中断&复位模式的初始设置步骤



备注：FIL：低速内部振荡器时钟频率

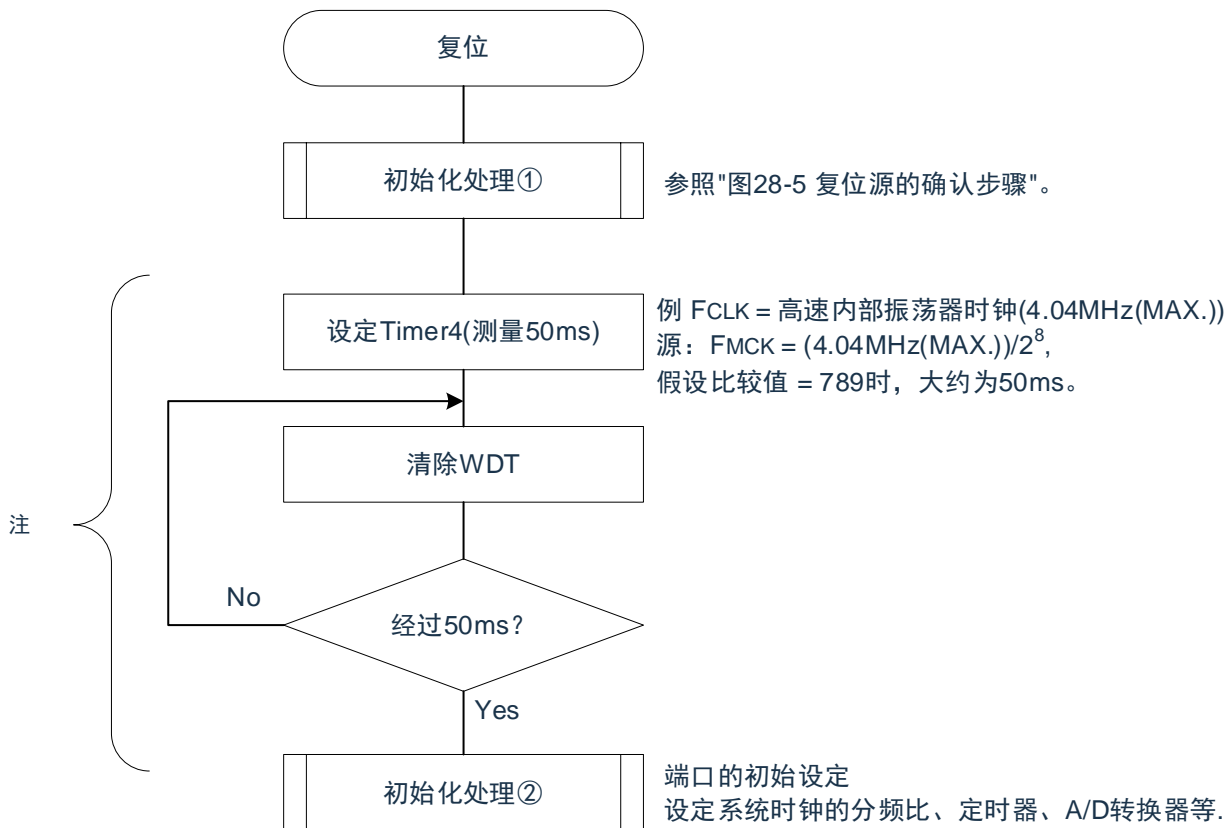
30.5 电压检测电路的注意事项

(1) 有关接通电源时的电压波动

对于电源电压(V_{DD})在LVD检测电压附近发生一定时间波动的系统，有可能重复进入复位状态和复位解除状态。能通过以下的处理，任意设置解除复位到单片机开始运行的时间。

<处理> 在解除复位后，必须通过使用定时器的软件计数器，在等待各系统不同的电源电压波动时间后进行端口等的初始设置。

图30-9: LVD检测电压附近的电源电压波动不超过50ms时的软件处理例子

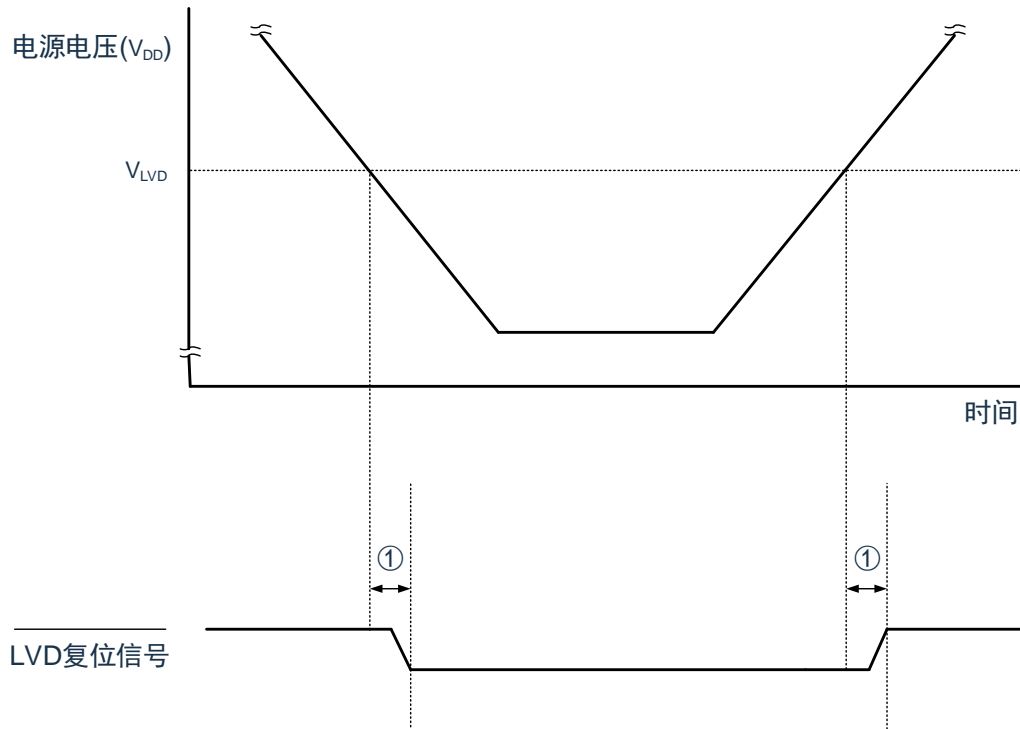


注: 如果在此期间再次发生复位, 就不转移到初始化处理②。

(2) 从产生LVD复位源到产生或者解除LVD复位的延迟

从满足电源电压(V_{DD}) < LVD检测电压(V_{LVD})到产生LVD复位为止会发生延迟。同样，从LVD检测电压(V_{LVD}) ≤ 电源电压(V_{DD})到解除LVD复位为止也会发生延迟(参照图30-10)。

图30-10: 从产生LVD复位源到产生或者解除LVD复位的延迟



①: 检测延迟(300us(MAX.))

(3) 有关将LVD置为OFF时接通电源的情况

当将LVD置为OFF时，必须使用RESETB引脚的外部复位。

在进行外部复位时，必须至少给RESETB引脚输入10us的低电平。如果在电源电压上升时进行外部复位，就必须在给RESETB引脚输入低电平后接通电源，而且在数据手册的AC特性所示的工作电压范围内至少保持10us的低电平，然后输入高电平。

(4) 有关将LVD置为OFF并且设置为LVD中断模式时工作电压下降的情况

在将LVD置为OFF并且设置为LVD中断模式的情况下，如果工作电压下降，就必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

第31章 安全功能

31.1 安全功能的概要

为了对应IEC60730和EC61508安全标准，BAT32A237内置以下安全功能。

此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

(1) 闪存CRC运算功能(高速CRC、通用CRC)

通过CRC运算检测闪存的数据错误。能根据不同的用途和使用条件，分别使用以下2个CRC。

- “高速CRC”... 在初始化程序中，能停止CPU的运行并且高速检查整个代码闪存区。
- “通用CRC”... 在CPU运行中，不限于代码闪存区而能用于多用途的检查。

(2) RAM 奇偶校验错误检测功能

在读RAM数据时，检测奇偶校验错误。

(3) SFR保护功能

防止因CPU失控而改写SFR。

(4) 频率检测功能

能使用通用定时器单元进行CPU/外围硬件时钟频率的自检。

(5) A/D 测试功能

能通过A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(ANI)、温度传感器输出和内部基准电压输出的A/D转换进行A/D转换器的自检。

(6) 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

(7) 产品唯一身份标识寄存器(128位)

31.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
<ul style="list-style-type: none"> •闪存CRC控制寄存器(CRCOCTL) •闪存CRC运算结果寄存器(PGCRCL) 	闪存CRC运算功能 (高速CRC)
<ul style="list-style-type: none"> •CRC输入寄存器(CRCIN) •CRC数据寄存器(CRCD) 	CRC运算功能 (通用CRC)
•RAM奇偶校验错误控制寄存器(RPECTL)	RAM奇偶校验错误检测功能
•特殊SFR保护控制寄存器(SFRGD)	SFR保护功能
•定时器输入选择寄存器0(TIS0)	频率检测功能
•A/D测试寄存器(ADTES)	A/D测试功能
•端口模式选择寄存器(PMS)	输入/输出引脚的数字输出信号电平检测功能

有关各寄存器的内容，在“31.3 安全功能的运行”中进行说明。

31.3 安全功能的运行

31.3.1 闪存CRC运算功能(高速CRC)

IEC60730标准要求确认闪存中的数据，并且建议CRC为确认手段。此高速CRC能在初始设置(初始化)程序中检查整个代码闪存区。

高速CRC停止CPU的运行并且通过1个时钟从闪存读32位数据进行运算。因此，其特点是完成检查的时间较短(例如，64KB闪存：512us@32MHz)。

CRC生成多项式对应CRC-16-CCITT的“ $X^{16}+X^{12}+X^5+1$ ”。

以bit31→bit0的MSB优先进行运算。

备注：因为通用CRC为LSB优先，所以运算结果不同。

31.3.1.1 闪存CRC控制寄存器(CRC0CTL)

这是设置高速CRC运算器的运行控制和运算范围的寄存器。通过8位存储器操作指令设置CRC0CTL寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图31-1：闪存CRC控制寄存器(CRC0CTL)的格式

地址：40021810H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	高速CRC运算器的运行控制
0	停止运行。
1	通过执行HALT指令开始运算。

FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	高速CRC的演算范围
0	0	0	0	0	0	0	00000H~1FFBH(8K-4byte)
0	0	0	0	0	0	1	00000H~3FFBH(16K-4byte)
0	0	0	0	0	1	0	00000H~5FFBH(24K-4byte)
0	0	0	0	0	1	1	00000H~7FFBH(32K-4byte)
0	0	0	0	1	0	0	00000H~9FFBH(40K-4byte)
0	0	0	0	1	0	1	00000H~BFFBH(48K-4byte)
0	0	0	0	1	1	0	00000H~DFFBH(56K-4byte)
0	0	0	0	1	1	1	00000H~FFFBH(64K-4byte)
0	0	0	1	0	0	0	00000H~11FFBH(72K-4byte)
0	0	0	1	0	0	1	00000H~13FFBH(80K-4byte)
0	0	0	1	0	1	0	00000H~15FFBH(88K-4byte)
0	0	0	1	0	1	1	00000H~17FFBH(96K-4byte)
0	0	0	1	1	0	0	00000H~19FFBH(104K-4byte)
0	0	0	1	1	0	1	00000H~1BFFBH(112K-4byte)
0	0	0	1	1	1	0	00000H~1DFFBH(120K-4byte)
0	0	0	1	1	1	1	00000H~1FFFBH(128K-4byte)
1	x	x	x	x	x	x	00000H~1EFFBH(124K-4byte)

备注：必须事先将用于比较的CRC运算结果期待值存入闪存的最后4字节，因此运算范围为减去4字节的范围。

31.3.1.2 闪存CRC运算结果寄存器(PGCRCL)

这是保存高速CRC运算结果的寄存器。
 通过16位存储器操作指令设置PGCRCL寄存器。
 在产生复位信号后，此寄存器的值变为“0000H”。

图31-2：闪存CRC运算结果寄存器(PGCRCL)的格式

地址：0x40021812 复位后：0000H R/W

	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

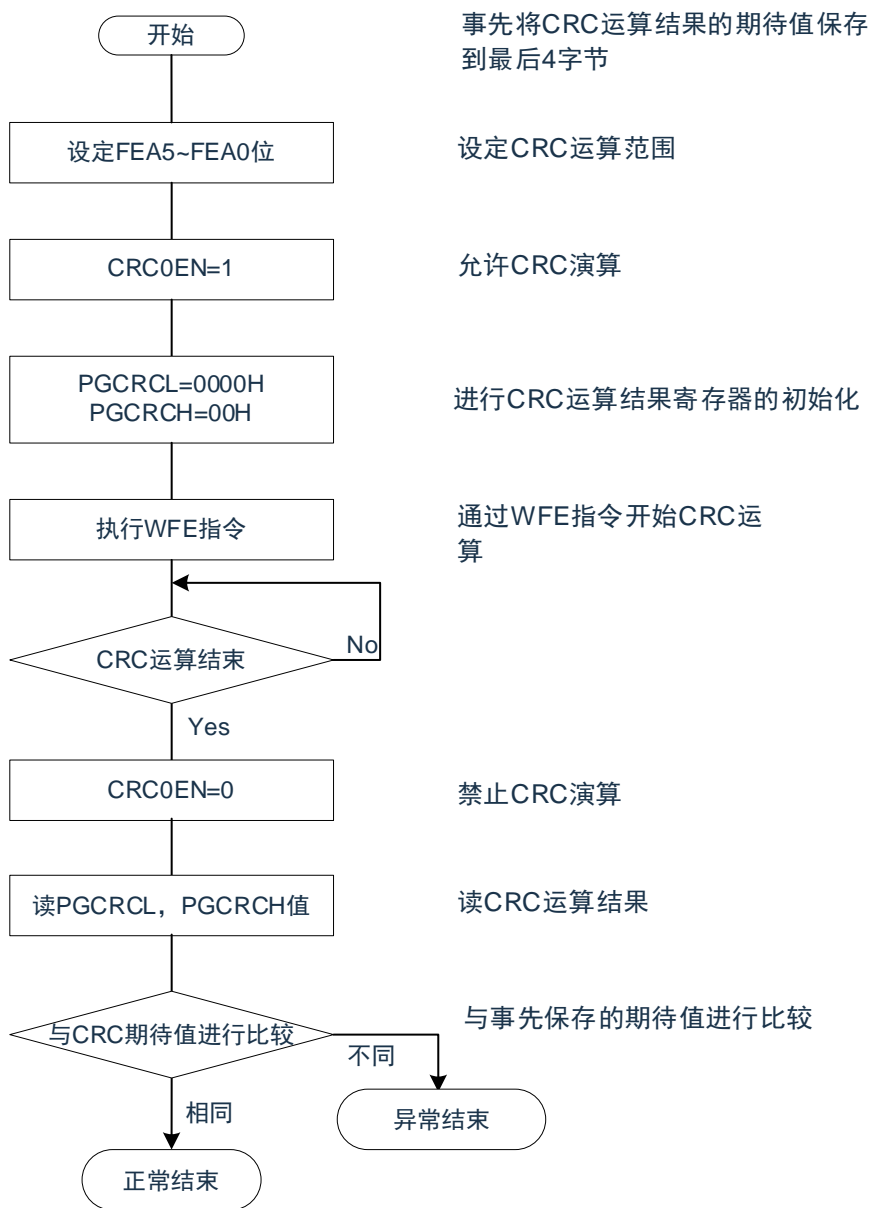
PGCRC15~0	高速CRC的运算结果
0000H~FFFFH	保存高速CRC的运算结果。

注意：只有在CRC0EN(CRC0CTL寄存器的bit7)位为“1”时才能写PGCRCL寄存器。

闪存CRC运算功能(高速CRC)的流程图如图31-3所示。

<操作流程>

图31-3: 闪存CRC运算功能(高速CRC)的流程图



注意:

1. 只以代码闪存为CRC运算的对象。
2. 必须将CRC运算的期待值保存在代码闪存中的运算范围后的区域。

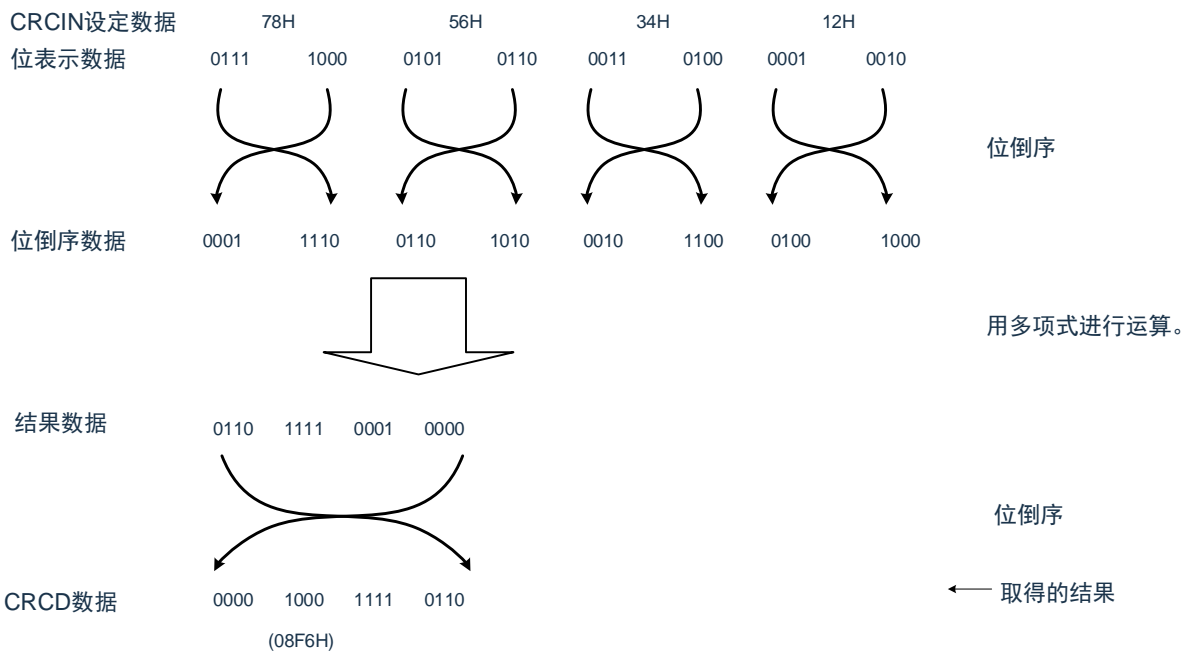
31.3.2 CRC运算功能(通用CRC)

为了必须保证运行过程中的安全，IEC61508标准要求即使在CPU运行中也需要确认数据。

此通用CRC能在CPU运行中作为外围功能进行CRC运算。通用CRC不限于代码闪存区而能用于多用途的检查。通过软件(用户程序)指定要确认的数据。睡眠模式中的CRC运算功能只能在DMA传送过程中使用。

在主系统时钟运行模式或者副系统时钟运行模式中，都能使用CRC运算功能。

CRC生成多项式使用CRC-16-CCITT的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以LSB优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从LSB发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给CRCIN寄存器写值，从CRCD寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行CRC运算的结果。



注意：在执行程序的过程中，因为调式程序将软件断点的设置行改写为断点指令，所以如果在CRC运算的对象区设置软件断点，CRC的运算结果就不同。

31.3.2.1 CRC输入寄存器(CRCIN)

这是设置通用CRC的CRC计算数据的8位寄存器。能设置的范围为“00H~FFH”。

通过8位存储器操作指令设置CRCIN寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图31-4：CRC输入寄存器(CRCIN)的格式

地址：400433ACH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CRCIN								

bit7~0	功能
00H~FFH	数据输入

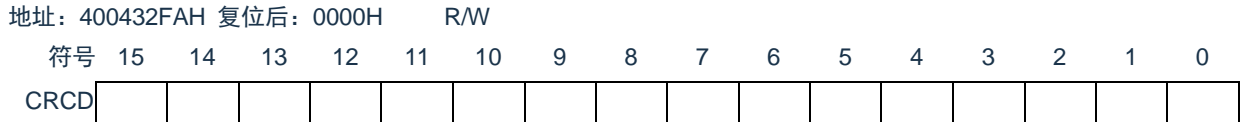
31.3.2.2 CRC数据寄存器(CRCD)

这是保存通用CRC运算结果的寄存器。能设置的范围为“0000H~FFFFH”。

在写CRCIN寄存器后经过1个CPU/外围硬件时钟(FCLK)，将CRC运算结果保存到CRCD寄存器。通过16位存储器操作指令设置CRCD寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图31-5: CRC数据寄存器(CRCD)的格式

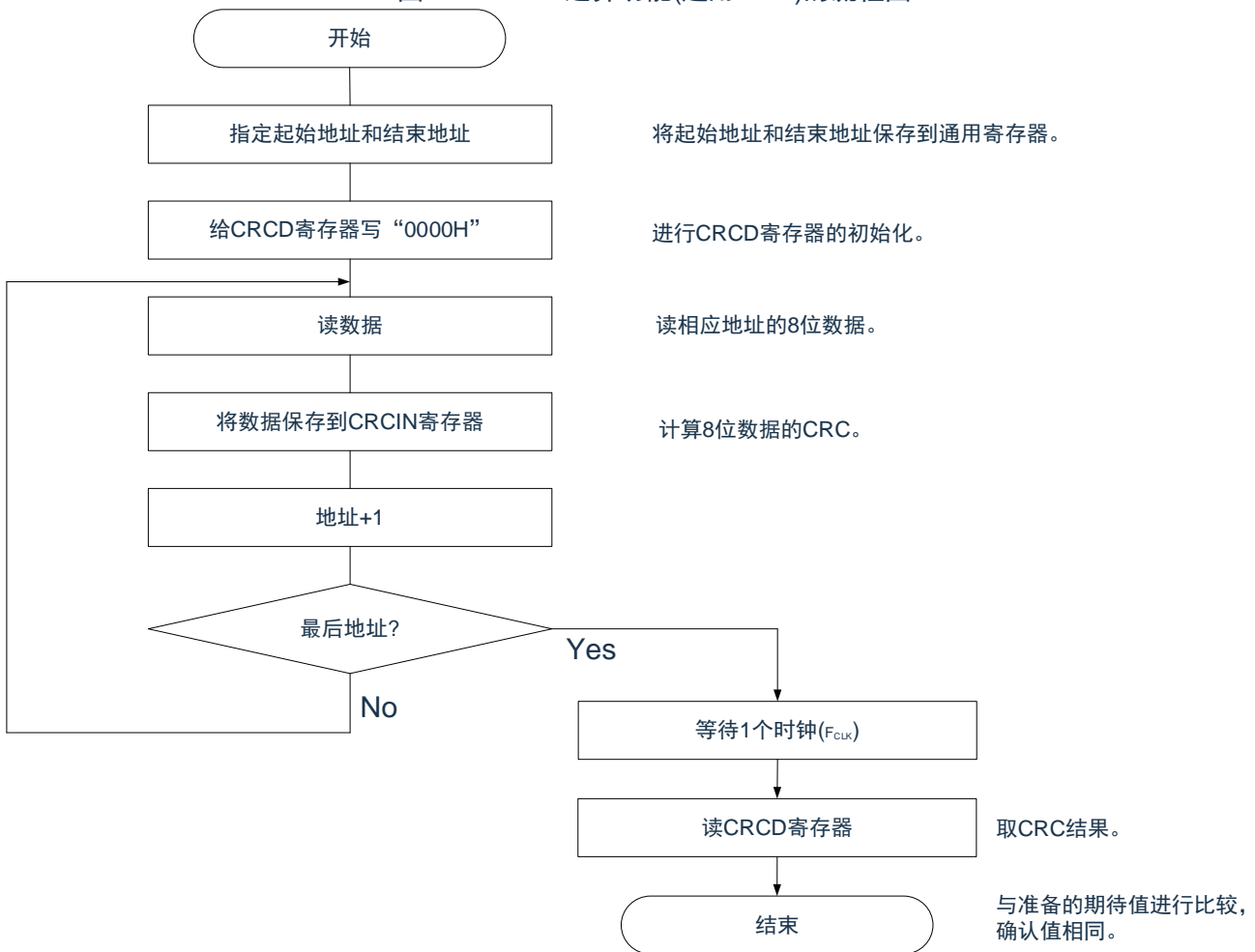


注意:

1. 要读CRCD寄存器的写入值时，必须在写CRCIN寄存器前读CRCD寄存器。
2. 如果CRCD寄存器的写操作与运算结果的保存发生竞争，就忽视写操作。

<操作流程>

图31-6: CRC运算功能(通用CRC)的流程图



31.3.3 RAM奇偶校验错误检测功能

IEC60730标准要求确认RAM数据。因此，BAT32A237的RAM每8位附加1位奇偶校验位。RAM奇偶校验错误检测功能在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位，并且能在发生奇偶校验错误时产生复位。

31.3.3.1 RAM奇偶校验错误控制寄存器(RPECTL)

此寄存器控制奇偶校验的错误确认位和因奇偶校验错误而产生复位。通过8位存储器操作指令设置RPECTL寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图31-7：RAM奇偶校验错误控制寄存器(RPECTL)的格式

地址：40020425H 复位后：00H R/W

	7	6	5	4	3	2	1	0
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	奇偶校验错误复位的屏蔽标志
0	允许产生奇偶校验错误复位。
1	禁止产生奇偶校验错误复位。

RPEF	奇偶校验错误状态标志
0	没有发生奇偶校验错误。
1	发生奇偶校验错误。

注意：在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位。

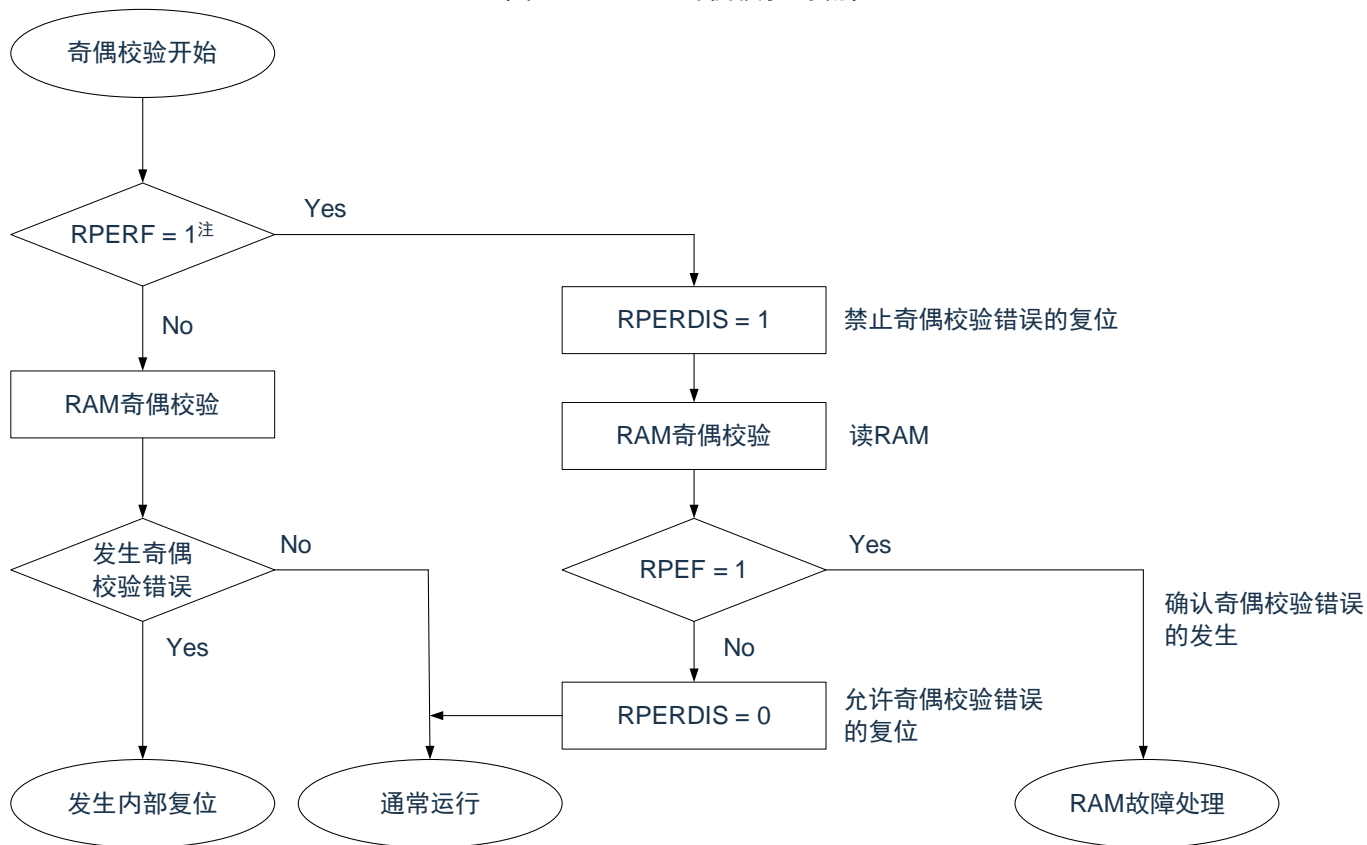
因此，要允许产生RAM奇偶校验错误复位(RPERDIS=0)时，必须在存取数据时并且在读数据前对“所用RAM区”进行初始化。

因为是流水线运行，所以CPU进行预读，有可能因读所用RAM区前的未初始化RAM区而发生RAM奇偶校验错误。因此，要允许产生RAM奇偶校验错误复位(RPERDIS=0)时，必须在从RAM区执行指令时对“所用RAM区+10字节”的区域进行初始化。

备注：

1. 初始状态为允许产生奇偶校验错误复位(RPERDIS=0)。
2. 即使设置为禁止产生奇偶校验错误复位(RPERDIS=1)，也在发生奇偶校验错误时将RPEF标志置“1”。如果在RPEF位为“1”的状态下设置为允许产生奇偶校验错误复位(RPERDIS=0)，就在将RPERDIS清“0”时产生奇偶校验错误复位。
2. 因RAM奇偶校验错误而将RPECTL寄存器的RPEF标志置“1”，通过写“0”或者全部复位源将RPEF标志清“0”。当RPEF标志为“1”时，即使读未发生奇偶校验错误的RAM，RPEF标志也保持“1”的状态。
3. RAM奇偶校验检测的范围不包括通用寄存器。

图31-8: RAM奇偶校验的流程



注：有关RAM奇偶错误的内部复位的确认，请参照“第28章 复位功能”。

31.3.4 SFR保护功能

为了必须保证运行过程中的安全，IEC61508标准要求即使CPU失控也需要保护重要的SFR，使其免遭改写。SFR保护功能用于保护比较器功能、端口功能、中断功能、时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的数据。

如果设置为SFR保护功能，被保护的SFR的写操作就无效，但是能正常读取。

31.3.4.1 SFR保护控制寄存器(SFRGD)

此寄存器控制SFR保护功能是否有效。

SFR保护功能使用GCOMP位、GPORT位、GINT位和GCSC位。

通过8位存储器操作指令设置SFRGD寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图31-9: SFR保护控制寄存器(SFRGD)的格式

地址: 40040478H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SFRGD	0	0	0	0	GCOMP	GPORT	GINT	GCSC
GCOMP	比较器功能的控制寄存器的保护							
0	无效。能读写比较器功能的控制寄存器。							
1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的SFR]COMPMDR、COMPFIR、COMPOCR、CVRCTL、CxRVM、PGAxCTL、PGASHMD、CMPSELx							
GPORT	端口功能的控制寄存器的保护							
0	无效。能读写端口功能的控制寄存器。							
1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的SFR]PMxx、PUxx、PIMxx、POMxx、PMCxx、ADPC、PIORx ^注							
GINT	中断功能的寄存器的保护							
0	无效。能读写中断功能的控制寄存器。							
1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的SFR]IFxx、MKxx、PRxx、EGPx、EGNx							
GCSC	时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的保护							
0	无效。能读写时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器。							
1	有效。时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的写操作无效，能读。 [被保护的SFR]CMC、CSC、OSTS、CKC、PERx、OSMC、LVIM、LVIS、RPECTL							

注：不保护Pxx(端口寄存器)。

31.3.5 频率检测功能

IEC60730标准要求确认振荡频率是否正常。

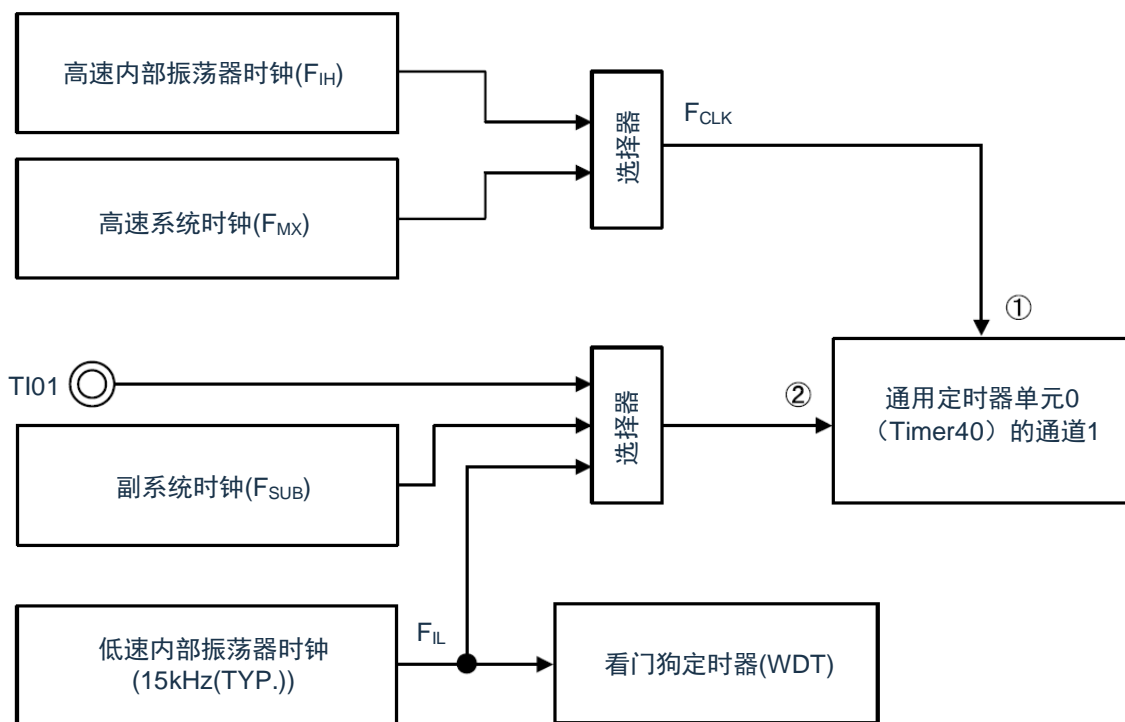
频率检测功能可使用CPU/外围硬件的时钟频率(F_{CLK}), 并且能通过测量Timer4的通道1输入脉冲, 判断2个时钟的比率关系是否正确。

但是, 如果某1个时钟或者2个时钟停止振荡, 就不能判断2个时钟的比率关系。

<要比较的时钟>

- ① CPU/外围硬件的时钟频率(F_{CLK}):
 - 高速内部振荡器时钟(F_{IH})
 - 高速系统时钟(F_{MX})
- ② Timer4的通道1输入:
 - 通道1的定时器输入(TI01)
 - 低速内部振荡器时钟(F_{IL} : 15KHz(TYP.))
 - 副系统时钟(F_{SUB})^注

图31-10: 频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时, 能判断为“时钟频率异常”。有关输入脉冲间隔的测量方法, 请参照“6.8.4作为输入脉冲间隔测量的运行”。

注: 只有内置副系统时钟的产品才能选择。

31.3.5.1 定时器输入选择寄存器0(TIS0)

寄存器说明请参考6.3.8节。

31.3.6 A/D测试功能

IEC60730标准要求进行A/D转换器的测试。此A/D测试功能通过对A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(ANi)、温度传感器的输出电压和内部基准电压进行A/D转换，确认A/D转换器是否正常运行。

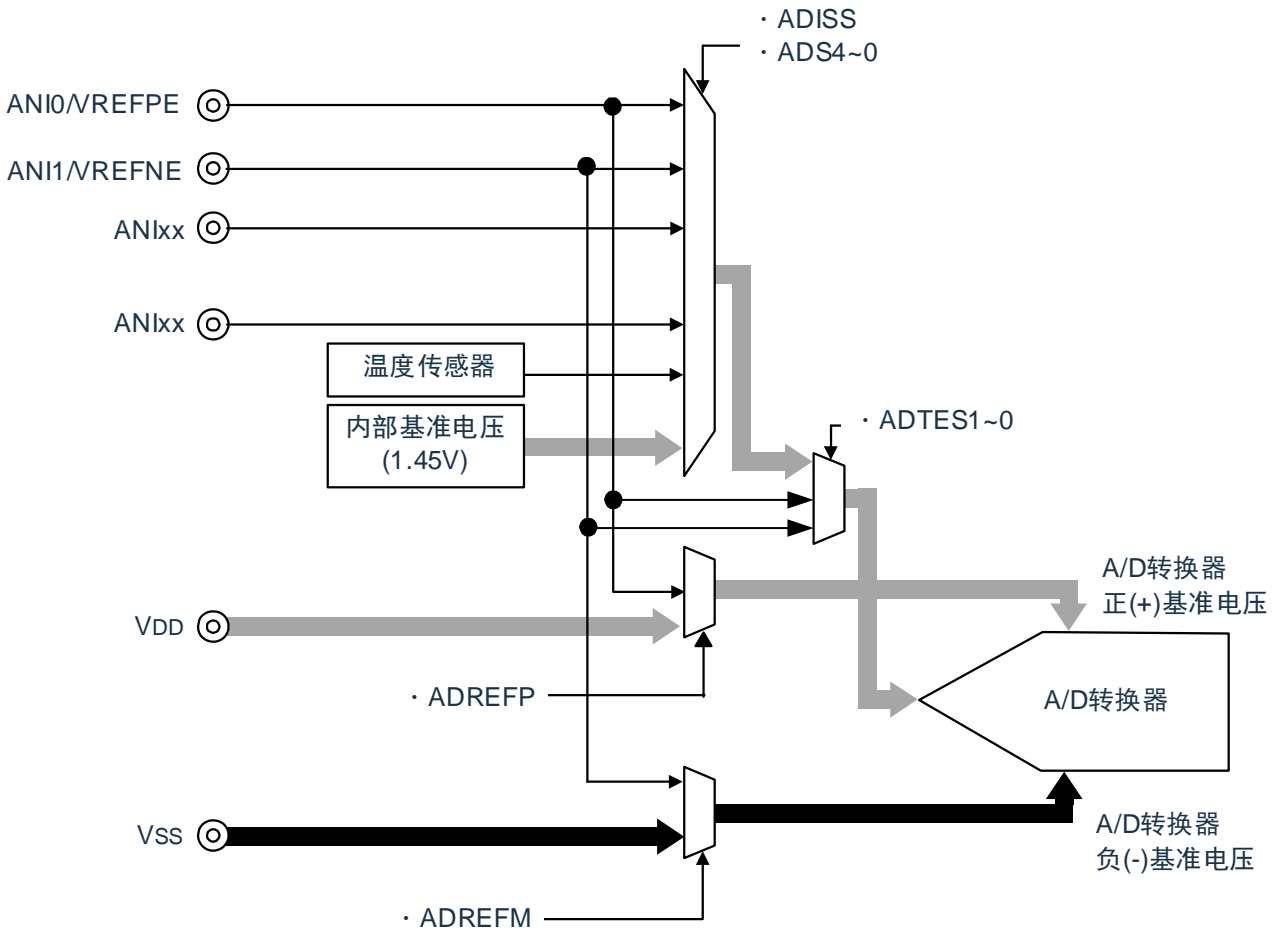
能通过以下步骤确认模拟多路转换器：

- ① 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ② 对ANix引脚进行A/D转换(转换结果1-1)。
- ③ 通过ADTES寄存器选择A/D转换器的负(-)基准电压作为A/D转换对象(ADTES1、ADTES0=1、0)。
- ④ 对A/D转换器的负(-)基准电压进行A/D转换(转换结果2-1)。
- ⑤ 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ⑥ 对ANix引脚进行A/D转换(转换结果1-2)。
- ⑦ 通过ADTES寄存器选择A/D转换器的正(+)基准电压作为A/D转换对象(ADTES1、ADTES0=1、1)。
- ⑧ 对A/D转换器的正(+)基准电压进行A/D转换(转换结果2-2)。
- ⑨ 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ⑩ 对ANix引脚进行A/D转换(转换结果1-3)。
- ⑪ 确认“转换结果1-1”、“转换结果1-2”和“转换结果1-3”相同。
- ⑫ 确认“转换结果2-1”的A/D转换结果全部为“0”并且“转换结果2-2”的A/D转换结果全部为“1”。通过以上步骤，能选择模拟多路转换器以及确认布线没有断线。

备注：

1. 在①~⑩的转换过程中，如果模拟输入电压可变，就必须采用其他方法来确认模拟多路转换器。
2. 转换结果含有误差，因此必须在比较转换结果时要适当考虑误差。

图31-11: A/D测试功能的结构



31.3.6.1 A/D测试寄存器(ADTES)

此寄存器选择A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(ANLxx)、温度传感器的输出电压和内部基准电压(1.45V)作为A/D转换对象。

当用作A/D测试功能时，进行以下的设置：

- 在测量零刻度时，选择负(-)基准电压作为A/D转换对象。
- 在测量满刻度时，选择正(+)基准电压作为A/D转换对象。

寄存器说明请参考15.2.10。

31.3.6.2 模拟输入通道指定寄存器(ADS)

此寄存器指定A/D转换的模拟电压的输入通道。

要通过A/D测试功能测量ANLxx、温度传感器输出或者内部基准电压(1.45V)时，必须将A/D测试寄存器(ADTES)置“00H”。

寄存器说明请参考15.2.7。

31.3.7 输入/输出引脚的数字输出信号电平检测功能

IEC60730标准要求确认I/O功能是否正常。

输入/输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

31.3.7.1 端口模式选择寄存器(PMS)

此寄存器选择在引脚为输出模式(端口模式寄存器(PMm)的PMmn位为“0”)时是读端口的输出锁存器的值还是读引脚的输出电平。

通过8位存储器操作指令设置PMS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图31-12: 端口模式选择寄存器(PMS)的格式

地址: 4004087BH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	在引脚为输出模式时读数据的选择
0	读Pmn寄存器的值。
1	读引脚的数字输出电平。

注意: 对于使用定时器M的脉冲输出强制截止功能而使引脚变为高阻抗状态的引脚，如果读引脚的数字输出电平，读取值就为“0”。

备注: m=0~7、12~14

n=0~7

31.3.8 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

- 用来作为序列号(例如USB字符序列号或者其他的终端应用)
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器内的安全性。
- 用来激活带安全机制的自举过程

128位的产品唯一身份标识所提供的参考号码对任意一个BAT32微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

基地址：0x0050_084C

地址偏移：0x00

只读，其值在出厂时编写

U_ID[31:0]

地址偏移：0x04

只读，其值在出厂时编写

U_ID[63:32]

地址偏移：0x08

只读，其值在出厂时编写

U_ID[95:64]

地址偏移：0x0C

只读，其值在出厂时编写

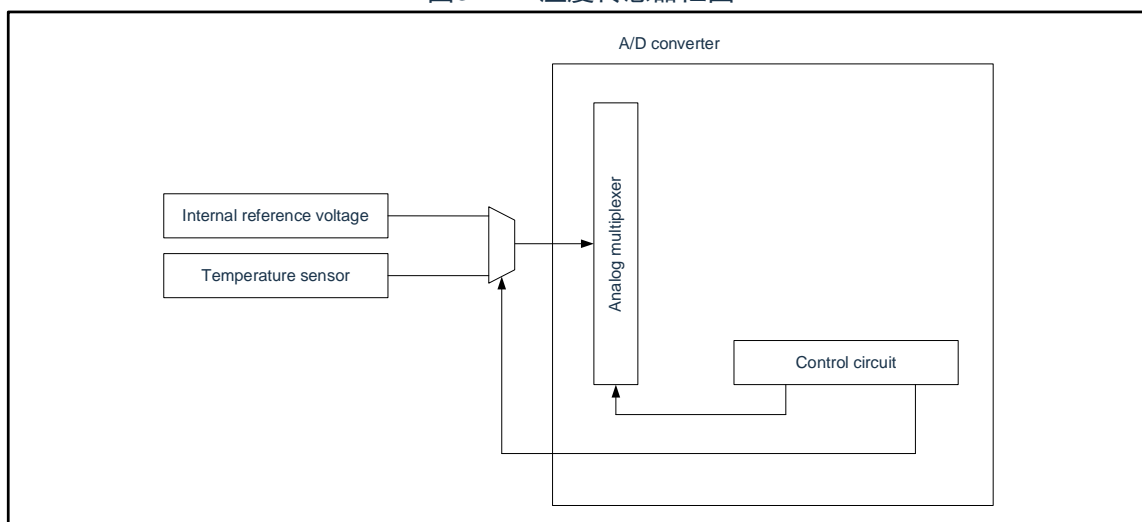
U_ID[127:96]

第32章 温度传感器

32.1 温度传感器的功能

片上的温度传感器可以对产品的核心温度进行测量和监控，从而保证产品的可靠运行。温度传感器输出的电压与核心温度成正比，并且电压和温度之间是线性关系。其输出电压提供给ADC进行转换。图32-1显示了温度传感器框图。

图32-1：温度传感器框图



32.2 温度传感器的寄存器

32.2.1 温度传感器校准数据寄存器 TSN25

地址：0x0050_066C

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
TSN25					TSN25[11:0]											-	R	

只读寄存器，用于记录温度传感器的25度校准数据，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

32.2.2 温度传感器校准数据寄存器 TSN125

地址：0x0050_0668

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
TSN125					TSN125[11:0]											-	R	

只读寄存器，用于记录温度传感器的125度校准数据，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

32.3 温度传感器的使用说明

32.3.1 温度传感器的使用原理

温度(T)与传感器电压输出(Vs)成正比，因此温度的计算公式如下：

$$T = (V_s - V_1) / \text{slope} + T_1$$

slope: 温度传感器的温度斜率(V/°C), $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$

T: 测量的温度(°C)

Vs: 温度传感器在温度测量时的输出电压(V)

T1: 在第一个点进行实验测量的温度(°C)

V1: 温度传感器测量T1时的电压输出(V)

T2: 在第二个点进行实验测量的温度(°C)

V2: 温度传感器测量T2时的电压输出(V)

不同传感器的特性不同，因此我们建议测量以下两种不同的样品温度：

使用A/D转换器测量温度传感器在温度T1时输出的电压V1，

使用A/D转换器测量温度传感器在第二个温度T2时输出的电压V2，

从两次结果中计算获得温度斜率， $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$

随后，通过将斜率代入温度特性的公式来获得温度， $T = (V_s - V_1) / \text{slope} + T_1$ 。

32.3.2 温度传感器的使用方法

方法一：在本产品中，TSN25寄存器存储了在 $T_A=T_J=25^{\circ}\text{C}$ 和 $V_{DD}=3.0\text{V}$ 条件下测量的温度传感器的电压转换值(CAL25)。TSN125寄存器存储了在 $T_A=T_J=125^{\circ}\text{C}$ 和 $V_{DD}=3.0\text{V}$ 条件下测量的温度传感器的电压转换值(CAL125)。利用这两组值可以计算出温度斜率：

$$\text{slope} = (V2 - V1) / (125 - 25)$$

$$V1 = 3.0 \times \text{CAL25} / 256 \text{ [V]}$$

$$V2 = 3.0 \times \text{CAL125} / 256 \text{ [V]}$$

利用以上结果，可根据以下公式计算温度：

$$T = (V_s - V1) / \text{slope} + 25 \text{ [}^{\circ}\text{C]}$$

T： 测量的温度($^{\circ}\text{C}$)

V_s ： 使用A/D转换器得到的温度传感器在T温度时的输出电压(V)

方法二：如果使用“电气特性”中给出的温度斜率，可直接使用以下公式计算测量温度：

$$T = (V_s - V1) / \text{slope} + 25 \text{ [}^{\circ}\text{C]}$$

备注：这种方法产生的温度比方法一测量的精度要低。

第33章 选项字节

33.1 选项字节的功能

BAT32A237的闪存000C0H~000C3H, 500004H~500005H为选项字节区。

选项字节由用户选项字节(000C0H~000C2H)和闪存数据保护选项字节(000C3H, 500004H~500005H)构成。在接通电源或者复位启动时, 自动参照选项字节进行指定功能的设置。在使用本产品时, 必须通过选项字节进行以下功能的设置。对于没有配置功能的位, 不能更改初始值。要在自编程过程中使用引导交换功能时, 因为000C0H~000C3H被010C0H~010C3H替换, 所以必须给010C0H~010C3H设置和000C0H~000C3H相同的值。

注意: 与是否使用各功能无关, 必须设置选项字节。

33.1.1 用户选项字节(000C0H~000C2H/010C0H~010C2H)

(1) 000C0H/010C0H

- 看门狗定时器的运行
 - 允许或者禁止计数器的运行。
 - 在睡眠/深度睡眠模式中允许或者停止计数器的运行。
- 看门狗定时器的上溢时间的设置
- 看门狗定时器的窗口打开期间的设置
- 看门狗定时器的间隔中断的设置
 - 使用或者不使用间隔中断。

注意: 在引导交换时, 000C0H被010C0H替换, 因此必须给010C0H设置和000C0H相同的值。

(2) 000C1H/010C1H

- LVD运行模式的设置
 - 中断&复位模式
 - 复位模式
 - 中断模式
 - LVD为OFF(使用RESETB引脚的外部复位输入)。
- LVD检测电平(V_{LVDH} 、 V_{LVDL} 、 V_{LVD})的设置

注意:

1. 当电源电压上升时, 必须在电源电压达到数据手册的AC特性所示的工作电压范围前, 通过电压检测电路或者外部复位保持复位状态; 当电源电压下降时, 必须在电源电压低于工作电压范围前, 通过深度睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。
工作电压范围取决于用户选项字节(000C2H/010C2H)的设置。
2. 在引导交换时, 000C1H被010C1H替换, 因此必须给010C1H设置和000C1H相同的值。

(3) 000C2H/010C2H

- 高速内部振荡器的频率设置
 - 从1MHz~32MHz、48MHz、64MHz中选择。

注意：在引导交换时，000C2H被010C2H替换，因此必须给010C2H设置和000C2H相同的值。

33.1.2 闪存数据保护选项字节(000C3H/010C3H, 500004H~500005H)

- 片上调试时闪存数据保护的控制在
 - Level0: 允许通过debugger对闪存数据进行读出/写入/擦除操作
 - Level1: 允许通过debugger对闪存数据进行chip全擦除操作, 不允许进行读写操作。
 - Level2: 不允许通过debugger对闪存数据进行操作。

- 引导交换功能的控制在
 - 禁止或者允许使用引导交换功能。

注意: 在引导交换时, 000C3H被010C3H替换, 因此必须给010C3H设置和000C3H相同的值。

33.2 用户选项字节的格式

图33-1: 用户选项字节(000C0H/010C0H)的格式

 地址: 000C0H/010C0H^{注1}

符号	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	看门狗定时器的间隔中断的使用/不使用
0	不使用间隔中断。
1	当达到上溢时间的75%+1/2F _{IL} 时, 产生间隔中断。

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间 ^{注2}
0	-	禁止设置。
1	0	75%
1	1	100%

WDTON	看门狗定时器的计数器运行控制
0	禁止计数器的运行(解除复位后停止计数)。
1	允许计数器的运行(解除复位后开始计数)。

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 (F _{IL} =20KHz(MAX.))
0	0	0	2 ⁶ /F _{IL} (3.2ms)
0	0	1	2 ⁷ /F _{IL} (6.4ms)
0	1	0	2 ⁸ /F _{IL} (12.8ms)
0	1	1	2 ⁹ /F _{IL} (25.6ms)
1	0	0	2 ¹¹ /F _{IL} (102.4ms)
1	0	1	2 ¹³ /F _{IL} (409.6ms)
1	1	0	2 ¹⁴ /F _{IL} (819.2ms)
1	1	1	2 ¹⁶ /F _{IL} (3276.8ms)

WDSTBYON	看门狗定时器的计数器运行控制(睡眠模式)
0	在睡眠模式中, 停止计数器的运行 ^{注2} 。
1	在睡眠模式中, 允许计数器的运行。

注1: 在引导交换时, 000C0H被010C0H替换, 因此必须给010C0H设置和000C0H相同的值。

注2: 当WDSTBYON位为“0”时, 与WINDOW1位和WINDOW0位的值无关, 窗口打开期间为100%。

备注: F_{IL}: 低速内部振荡器的时钟频率

图33-2: 用户选项字节(000C1H/010C1H)的格式(1/4)

 地址: 000C1H/010C1H^注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置 (中断&复位模式)

检测电压			选项字节的设置值						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降	下降						LVIMDS1	LVIMDS0
2.09V	2.04V	1.84V	0	0	1	0	1	1	0
2.71V	2.65V	2.45V		1	0	0	1		
3.75V	3.67V	2.75V		1	1	0	1		
3.02V	2.96V					0	0		
4.06V	3.98V					0	0		
—			禁止设置上述以外的值。						

注: 在引导交换时, 000C1H被010C1H替换, 因此必须给010C1H设置和000C1H相同的值。

注意: 必须给bit4写“1”。

备注:

1. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
2. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

图33-2: 用户选项字节(000C1H/010C1H)的格式(2/4)

 地址: 000C1H/010C1H^注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置 (复位模式)

检测电压		选项字节的设置值						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
2.09V	2.04V	0	0	1	0	1	1	1
2.71V	2.65V		1	0	0	1		
3.02V	2.96V		1	1	0	1		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设置上述以外的值。						

注: 在引导交换时, 000C1H被010C1H替换, 因此必须给010C1H设置和000C1H相同的值。

注意: 必须给bit4写“1”。

备注:

1. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
2. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

图33-2: 用户选项字节(000C1H/010C1H)的格式(3/4)

 地址: 000C1H/010C1H^注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置 (中断模式)

检测电压		选项字节的设置值						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
2.09V	2.04V	0	0	1	0	1	0	1
2.71V	2.65V		1	0	0	1		
3.02V	2.96V		1	1	0	1		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设置上述以外的值。						

注: 在引导交换时, 000C1H被010C1H替换, 因此必须给010C1H设置和000C1H相同的值。

注意: 必须给bit4写“1”。

备注:

1. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
2. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

图33-2: 用户选项字节(000C1H/010C1H)的格式(4/4)

 地址: 000C1H/010C1H^注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD为OFF时的设置(使用RESETB引脚的外部复位输入)

检测电压		选项字节的设置值						
V _{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设置上述以外的值。						

注: 在引导交换时, 000C1H被010C1H替换, 因此必须给010C1H设置和000C1H相同的值。

注意:

1. 必须给bit4写“1”。
2. 当电源电压上升时, 必须在电源电压达到数据手册的AC特性所示的工作电压范围前, 通过电压检测电路或者外部复位保持复位状态; 当电源电压下降时, 必须在电源电压低于工作电压范围前, 通过睡眠模式的转移、电压检测电路或者外部复位, 置为复位状态。
3. 工作电压范围取决于用户选项字节(000C2H/010C2H)的设置。

备注:

1. ×: 忽略
2. 有关LVD电路的详细内容, 请参照“第30章 电压检测电路”。
3. 检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

图33-3: 用户选项字节(000C2H/010C2H)的格式

 地址: 000C2H/010C2H^注

7	6	5	4	3	2	1	0
1	1	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
					F _{HOCO}	F _{IH}
1	1	0	0	0	64MHz	32MHz
1	0	0	0	0	48MHz	48MHz
0	1	0	0	0	32MHz	32MHz
0	0	0	0	0	24MHz	24MHz
0	1	0	0	1	32MHz	16MHz
0	0	0	0	1	24MHz	12MHz
0	1	0	1	0	32MHz	8MHz
0	0	0	1	0	24MHz	6MHz
0	1	0	1	1	32MHz	4MHz
0	0	0	1	1	24MHz	3MHz
0	1	1	0	0	32MHz	2MHz
0	1	1	0	1	32MHz	1MHz
上述以外					禁止设置。	

注: 在引导交换时, 000C2H被010C2H替换, 因此必须给010C2H设置和000C2H相同的值。

注意:

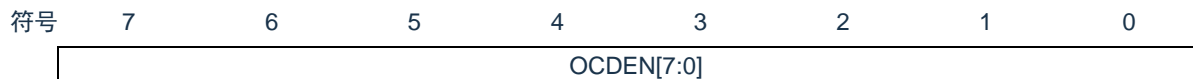
1. 必须给bit7~5写“1”。
2. 工作频率范围和工作电压范围因闪存的不同运行模式而不同。详细内容请参照数据手册的AC特性。
3. 改变HOCO频率需要振荡稳定时间, 不允许MCU运行时修改FRQSEL4、3, 修改后需要上电复位生效。

33.3 闪存数据保护选项字节的格式

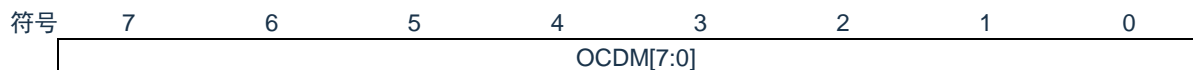
闪存数据保护选项字节的格式如下所示。

图33-4: 闪存数据保护选项字节(000C3H/010C3H)的格式

地址: 000C3H/010C3H^注



地址: 500004H



OCDM	OCDEN	闪存数据保护的
3C	C3	不允许通过debugger对闪存数据进行操作。
3C以外的值	C3	允许通过debugger对闪存数据进行chip全擦除操作，不允许进行读写操作。
上記以外		允许通过debugger对闪存数据进行读出/写入/擦除操作

注: 在引导交换时, 000C3H被010C3H替换, 因此必须给010C3H设置和000C3H相同的值。

注意: 50_0004H, 50_0005H地址属于数据闪存区, 如果使用该地址做数据存储用, 需先确定数值不会引起保护选项的误设。

地址: 500005H



BTEN	引导交换功能的
0	使用引导交换功能, 0000H~0FFFH和1000H~1FFFH内容交换
1	禁止交换功能使能

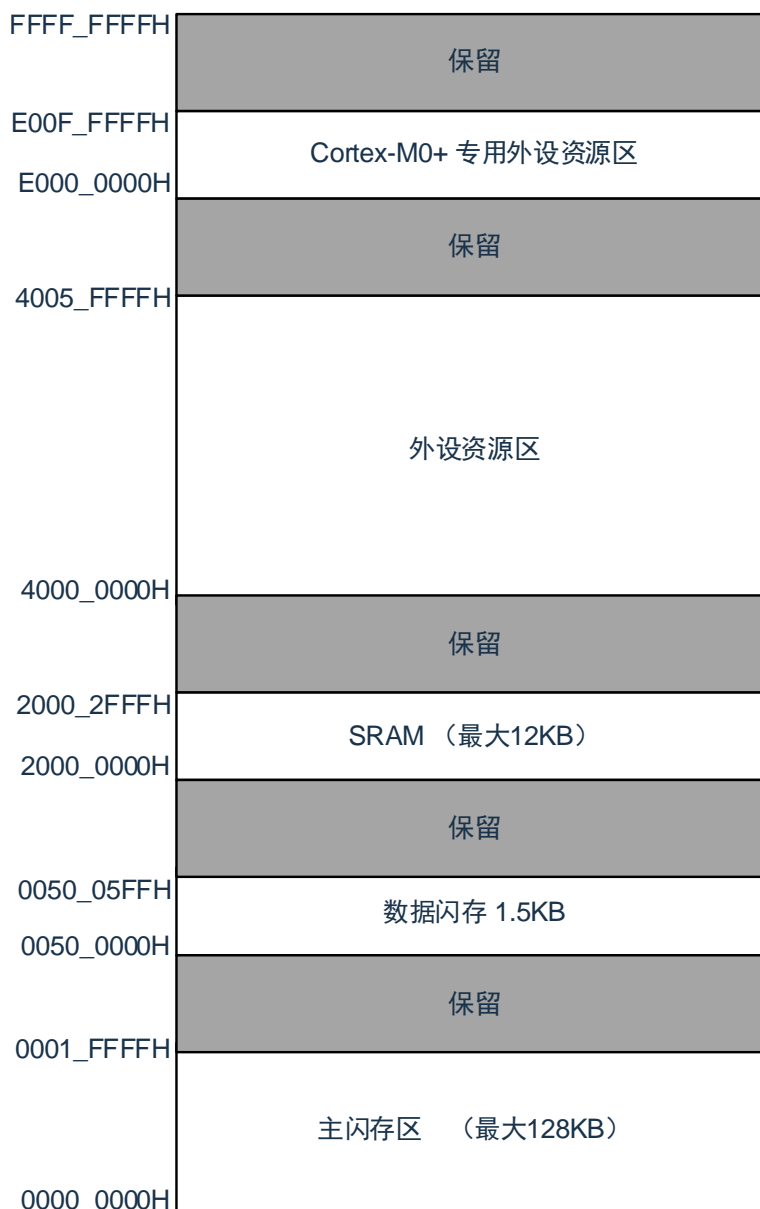
注意: 50_0004H, 50_0005H地址属于数据闪存区, 如果使用该地址做数据存储用, 需先确定数值不会引起保护选项的误设。

第34章 FLASH控制

34.1 FLASH控制功能描述

本制品包含一颗128KByte容量的FLASH 存储器，共划分为256个Sector，每个Sector容量为512Byte。可做为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

34.2 FLASH存储器结构



34.3 控制FLASH的寄存器

控制FLASH的寄存器如下所示：

- Flash 写保护寄存器(FLPROT)
- Flash 操作控制寄存器(FLOPMD1,FLOPMD2)
- Flash 擦除模式控制寄存器(FLERMD)
- Flash 状态寄存器(FLSTS)
- Flash 全片擦除时间控制寄存器(FLCERCNT)
- Flash 页擦除时间控制寄存器(FLSERCNT)
- Flash 写入时间控制寄存器(FLPROCNT)

34.3.1 Flash写保护寄存器(FLPROT)

Flash保护寄存器用于对FLASH操作控制寄存器进行保护。

地址：0x40020020 复位后：0000000H R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FLPROT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	-	PRKEY[7:1]							WRP

WRP	操作寄存器(FLOPMD1/FLOPMD2)写保护
0	不允许改写 FLOPMD1/ FLOPMD2
1	允许改写 FLOPMD1/ FLOPMD2

PRKEY[7:1]	WRP写保护
78h	允许改写 WRP
78h以外的值	不允许改写 WRP

34.3.2 FLASH 操作控制寄存器(FLOPMD1,FLOPMD2)

Flash操作控制寄存器，用于设置FLASH的擦除和写入操作。

地址：0x40020004 复位后：00000000H R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	FLOPMD1[7:0]							

地址：0x40020008 复位后：00H R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	FLOPMD2[7:0]							

FLOPMD1	FLOPMD2	操作
55	AA	擦除
AA	55	写入
00	00	读出
上記以外		禁止设置

34.3.3 Flash擦除控制寄存器(FLERMD)

Flash擦除控制寄存器，用于设置FLASH擦除操作的类型。

地址：0x4002000C 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
FLERMD	0	0	0	ERMD1	ERMD0	0	0	0

ERMD1	ERMD0	操作
0	0	sector 擦除，擦除后不进行硬件校验
1	0	sector 擦除，擦除后进行硬件校验
0	1	chip 擦除 ^注
1	1	禁止设置

注：chip擦除只擦除代码闪存区域，不擦除数据闪存区域。且chip擦除不支持硬件校验。

34.3.4 Flash 状态寄存器(FLSTS)

通过状态寄存器可以查询FLASH控制器的状态。

地址: 0x40020000	复位后: 00H R/W							
符号	7	6	5	4	3	2	1	0
FLSTS	0	0	0	0	0	EVF [‡]	0	OVF [‡]

OVF	FLASH 擦写操作完了标志
0	FLASH 擦写操作未完成
1	FLASH 擦写操作完成

注: OVF需要软件写“1”进行清除。若不清除, 不能进行下一次擦写操作。

EVF	FLASH 擦除硬件校验错误标志
0	FLASH 擦除后, 硬件校验没有发生错误
1	FLASH 擦除后, 硬件校验发生了错误

注: EVF需要软件写“1”进行清除。

34.3.5 Flash 全片擦除时间控制寄存器(FLCERCNT)

通过FLCERCNT寄存器可以设置FLASH全片擦除的时间。

地址: 0x40020010 复位后: 不定 R/W

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

Load	擦除时间设置的选择 [‡]
0	使用硬件设置的擦除时间
1	使用软件设置的擦除时间(FLCERCNT[9:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLCERCNT。

FLCERCNT[9:0]	软件擦除时间设置
Chip 擦除时间=(CERCNT*2048*T _{FCLK}), 需满足>20ms 的硬件要求	

34.3.6 Flash 页擦除时间控制寄存器(FLSERCNT)

通过FLSERCNT寄存器可以设置FLASH全片擦除的时间。

地址: 0x40020014

复位后: 不定 R/W

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

Load	擦除时间设置的选择 ^注
0	使用硬件设置的擦除时间
1	使用软件设置的擦除时间(FLSERCNT[9:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLSERCNT。

FLSERCNT [9:0]	软件擦除时间设置
sector 擦除时间 $= (\text{SERCNT} * 256 * T_{\text{FCLK}})$, 需满足 $> 4\text{ms}$ 的硬件要求	

34.3.7 Flash 写入时间控制寄存器(FLPROCNT)

通过FLPROCNT寄存器可以设置FLASH WORD写入的时间。

地址: 0x4002001C

复位后: 不定 R/W

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLPROCNT	Load1	-	-	-	-	-	-	FLPGSCNT[8:0]								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Load0	-	-	-	-	-	-	FLPROCNT[8:0]								

Load0	写入时间(Tprog)设置 ^注
0	使用硬件设置的写入时间
1	使用软件设置的擦除时间(FLPROCNT[9:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLPROCNT。

FLPROCNT[8:0]	软件擦除时间设置
写入时间=(PROCNT*4* T_{FCLK}), 需满足 $>24\mu\text{s}$ 的硬件要求	

Load1	写入动作建立时间(Tpgs)设置 ^注
0	使用硬件设置的写入动作建立时间
1	使用软件设置的擦除时间(FLPGSCNT8:0)

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLPGSCNT。

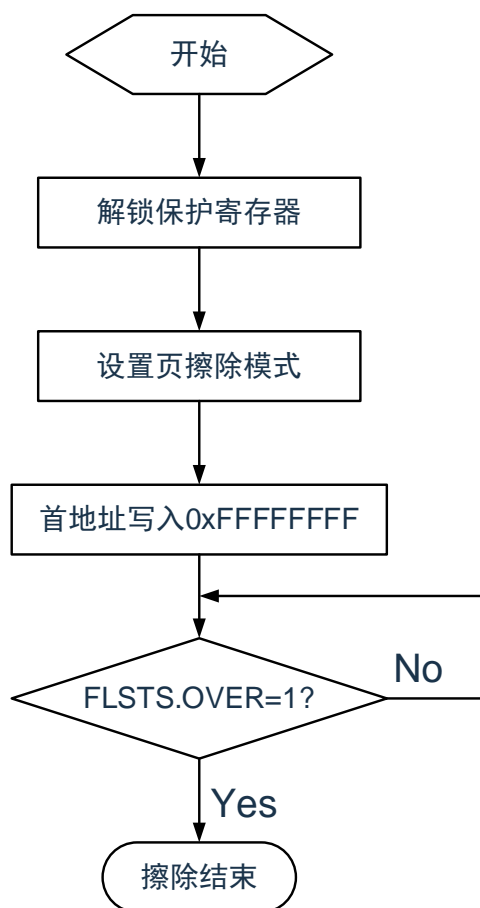
FLPGSCNT[8:0]	软件擦除时间设置
写入动作建立时间=(PGSCNT* T_{FCLK}), 需满足 $>5\mu\text{s}$ 的硬件要求	

34.4 FLASH操作方法

34.4.1 页擦除(sector erase)

sector擦除，擦除时间由硬件实现，也可以通过FLSERCNT配置。操作流程如下：

- 1) 设置FLERMD.ERMD0为1'b0，选择sector擦除模式,根据是否需要硬件校验选择设置ERMD1的值；
- 2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，
- 3) 向擦除目标sector的首地址写入任意数据。例：`*((unsigned long *)0x00000200)= 0xffffffff`。
- 4) 软件查询状态寄存器FLSTS.OVF, OVF=1时，表示擦除操作完成。
- 5) 如果设置了擦除后进行硬件校验(ERMD1=1)，可以通过软件判断FLSTS.EVF，查询是否校验正确。
- 6) 进行下次操作前，软件置“1”来清除FLSTS。



34.4.2 全片擦除(chip erase)

chip擦除，擦除时间由硬件实现，也可以通过FLCERCNT配置。操作流程如下：

- 1) 设置FLERMD.ERMD0为1'b1，选择chip擦除模式；
- 2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，
- 3) 向代码闪存区域的任意地址写入任意数据。
- 4) 软件查询状态寄存器FLSTS.OVF, OVF=1时，表示擦除操作完成。
- 5) 进行下次操作前，软件置"1"来清除FLSTS。

34.4.3 编程(word program)

word编程，写入时间由硬件实现，也可以通过PROCNT配置。操作流程如下：

- 1) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0xAA，FLOPMD2设置为0x55，
- 2) 向目标地址写入相应的数据。
- 3) 软件查询状态寄存器FLSTS.OVF, OVF=1时，表示写入操作完成。
- 4) 进行下次操作前，软件置"1"来清除FLSTS。

34.5 闪存读取

本设备内置的FLASH支持的最快取指频率为32MHz。当HCLK频率超过32MHz时，硬件会在CPU访问FLASH时插入1等待周期。

34.6 FLASH操作的注意事项

- FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。擦写参数的设置可以由硬件实现，也可通过修改参数寄存器进行软件修改；在使用内部高速OCO，MAINOSC/外部输入时钟=20M时，推荐使用硬件设置的擦写参数，无需设置参数寄存器。
- 如果擦写操作从FLASH内执行，则CPU会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从RAM里执行，CPU不会停止取指，当前可以继续下一条指令。
- 在FLASH处于编程操作中时，如果CPU执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

附录 修订记录

版本	日期	修订内容
V1.00	2022年3月	初版发行
V1.0.1	2023年5月	1) 优化格式 2) 19.5.2 更正使用的引脚 3) 更正8.3.2、8.4.3、8.6.4描述的错误内容
V1.0.2	2023年11月	1) 寄存器TSN85变更为TSN125 2) 32.3.2节 公式 $slope = (V2 - V1) / (85 - 25)$ 变更为 3) $slope = (V2 - V1) / (125 - 25)$ 4) 32.3.2节 “CAL85” 变更为 “CAL125” 5) 第32章中, “85度” 均变更为 “125度” 6) 第28章标题下 “注意3” 端口引脚复位状态的描述变更。 7) 修改图30-1和图30-6中的错误 8) 修改图21-62中的错误 9) 章节32.2 温度传感器的寄存器中TSN25和TSN125寄存器由8位改为12位
V1.0.3	2024年3月	1) 修改11.3.4章节中寄存器RWAIT位的描述 2) 2.3.8章节增加24引脚产品的相关描述