



# BAT32A233 用户手册

**基于 ARM® Cortex®-M0+的超低功耗 32 位微控制器**

**内置 32K 字节 Flash，集成丰富的模拟功能，定时器，LIN2.2 等多种通讯接口**

**V0.1.4 Draft Version**

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](http://www.mcu.com.cn)

## 文档使用说明

本手册是BAT32A233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：

BAT32A233xx: BAT32A233\_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
<b>第1章 CPU .....</b>	<b>20</b>
1.1 概述 .....	20
1.2 Cortex-M0+内核特性.....	20
1.3 调试特性 .....	20
1.4 SWD接口引脚.....	22
1.5 ARM参考文档 .....	23
<b>第2章 引脚功能 .....</b>	<b>24</b>
2.1 端口功能 .....	24
2.2 端口复用功能 .....	24
2.3 控制端口功能的寄存器 .....	25
2.3.1 端口模式寄存器 (PMxx) .....	27
2.3.2 端口寄存器 (Pxx) .....	28
2.3.3 端口置位控制寄存器 (PSETxx) .....	29
2.3.4 端口清零控制寄存器 (PCLRxx) .....	30
2.3.5 上拉电阻选择寄存器 (PUxx) .....	31
2.3.6 端口输入模式寄存器 (PIMxx) .....	32
2.3.7 端口输出模式寄存器 (POMxx) .....	33
2.3.8 端口模式控制寄存器 (PMCxx) .....	34
2.3.9 端口回读寄存器 (PREADxx) .....	35
2.3.10 外围I/O重定向寄存器0 (PIOR0) .....	36
2.3.11 外围I/O重定向寄存器1 (PIOR1) .....	37
2.3.12 外围I/O重定向寄存器2 (PIOR2) .....	38
2.3.13 外围I/O重定向寄存器3 (PIOR3) .....	39
2.3.14 全局数字输入禁止寄存器 (GDIDIS) .....	40
2.4 未使用引脚的处理 .....	41
2.5 使用复用功能时的寄存器设定 .....	42
2.5.1 使用复用功能时的基本思想 .....	42
2.5.2 使用的端口功能和复用功能的寄存器设定例子 .....	43
<b>第3章 系统结构 .....</b>	<b>52</b>
3.1 概述 .....	52
3.2 系统地址划分 .....	53
<b>第4章 时钟发生电路 .....</b>	<b>55</b>
4.1 时钟发生电路的功能 .....	55
4.2 时钟发生电路的结构 .....	56
4.3 控制时钟发生电路的寄存器 .....	59
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、2(PER0、PER1、PER2) .....	65
4.3.7 副系统时钟提供模式控制寄存器 (OSMC) .....	69
4.3.8 副系统时钟选择寄存器 (SUBCKSEL) .....	70
4.3.9 高速内部振荡器的频率选择寄存器(HOCODIV) .....	71
4.3.10 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	高速内部振荡器 .....	76
4.4.3	低速内部振荡器 .....	76
4.5	时钟发生电路的运行 .....	76
4.6	时钟控制 .....	78
4.6.1	高速内部振荡器的设置例子 .....	78
4.6.2	X1振荡电路的设置例子 .....	80
4.6.3	CPU时钟的状态转移图 .....	81
4.6.4	CPU时钟转移前的条件和转移后的处理 .....	87
4.6.5	CPU时钟和主系统时钟的切换所需时间 .....	88
4.6.6	时钟振荡停止前的条件 .....	89
<b>第5章</b>	<b>硬件除法器 .....</b>	<b>90</b>
5.1	特点 .....	90
5.2	功能描述 .....	90
5.3	硬件除法器的寄存器 .....	90
5.3.1	被除数寄存器(DIVIDEND) .....	91
5.3.2	除数寄存器(DIVISOR) .....	91
5.3.3	商寄存器(QUOTIENT) .....	91
5.3.4	余数寄存器(REMAINDER) .....	91
5.3.5	状态寄存器(STATUS) .....	92
<b>第6章</b>	<b>通用定时器单元Timer4 .....</b>	<b>93</b>
6.1	通用定时器单元的功能 .....	94
6.1.1	独立通道运行功能 .....	94
6.1.2	多通道联动运行功能 .....	96
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) .....	97
6.1.4	LIN-bus支持功能(只限于单元0的通道3) .....	97
6.2	通用定时器单元的结构 .....	98
6.2.1	通用定时器单元寄存器列表 .....	100
6.2.2	定时器计数寄存器mn (TCRmn) .....	101
6.2.3	定时器数据寄存器mn(TDRmn) .....	103
6.3	控制通用定时器单元的寄存器 .....	104
6.3.1	外围允许寄存器0 (PER0) .....	105
6.3.2	定时器时钟选择寄存器m(TPSm) .....	106
6.3.3	定时器模式寄存器mn(TMRmn) .....	109
6.3.4	定时器状态寄存器mn(TSRmn) .....	113
6.3.5	定时器通道允许状态寄存器m(TEm) .....	114
6.3.6	定时器通道开始寄存器m(TSm) .....	115
6.3.7	定时器通道停止寄存器m(TTm) .....	116
6.3.8	定时器输入输出选择寄存器(TIOS0) .....	117
6.3.9	定时器输出允许寄存器m(TOEm) .....	118
6.3.10	定时器输出寄存器m(TOm) .....	118
6.3.11	定时器输出电平寄存器m(TOLm) .....	119
6.3.12	定时器输出模式寄存器m(TOMm) .....	120
6.3.13	输入切换控制寄存器(ISC) .....	121
6.3.14	噪声滤波器允许寄存器(NFEN1) .....	122
6.3.15	控制定时器输入/输出引脚端口功能的寄存器 .....	123
6.4	通用定时器单元的基本规则 .....	124

6.4.1	多通道联动运行功能的基本规则 .....	124
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).....	126
6.5	计数器的运行 .....	127
6.5.1	计数时钟( $F_{TCLK}$ ) .....	127
6.5.2	计数器的开始时序.....	129
6.5.3	计数器的运行.....	130
6.6	通道输出(TOmn引脚)的控制 .....	135
6.6.1	TOmn引脚输出电路的结构 .....	135
6.6.2	TOmn引脚的输出设置 .....	136
6.6.3	通道输出运行的注意事项 .....	137
6.6.4	TOmn位的一次性操作 .....	141
6.6.5	有关开始计数时的定时器中断和TOmn引脚输出 .....	142
6.7	定时器输入(TImn)的控制.....	143
6.7.1	TImn引脚输入电路的结构.....	143
6.7.2	噪声滤波器 .....	143
6.7.3	操作通道输入时的注意事项 .....	144
6.8	通用定时器单元的独立通道运行功能 .....	145
6.8.1	作为间隔定时器/方波输出的运行 .....	145
6.8.2	作为外部事件计数器的运行 .....	149
6.8.3	作为分频器的运行(只限于单元0 的通道0).....	152
6.8.4	作为输入脉冲间隔测量的运行.....	155
6.8.5	作为输入信号高低电平宽度测量的运行 .....	158
6.8.6	作为延迟计数器的运行 .....	162
6.9	通用定时器单元的多通道联动运行功能 .....	165
6.9.1	作为单触发脉冲输出功能的运行 .....	165
6.9.2	作为PWM功能的运行 .....	172
6.9.3	作为多重PWM输出功能的运行 .....	179
6.10	使用通用定时器单元时的注意事项 .....	187
6.10.1	使用定时器输出时的注意事项.....	187
<b>第7章</b>	<b>定时器A .....</b>	<b>188</b>
7.1	定时器A的功能.....	188
7.2	定时器A的结构.....	189
7.3	控制定时器A的寄存器 .....	190
7.3.1	外围允许寄存器1(PER1) .....	190
7.3.2	副系统时钟提供模式控制寄存器(OSMC) .....	191
7.3.3	定时器A计数寄存器0(TA0) .....	192
7.3.4	定时器A控制寄存器0(TACR0) .....	193
7.3.5	定时器AI/O控制寄存器0(TAIOC0) .....	194
7.3.6	定时器A控制寄存器0(TAMR0).....	196
7.3.7	定时器A事件引脚选择寄存器0(TAISR0) .....	197
7.3.8	端口模式寄存器x(PMx).....	198
7.4	定时器A的运行.....	199
7.4.1	重加载寄存器和计数器的改写.....	199
7.4.2	定时器模式 .....	200
7.4.3	脉冲输出模式.....	201
7.4.4	事件计数器模式 .....	202
7.4.5	脉宽测量模式.....	203
7.4.6	脉冲周期测量模式.....	204

7.4.7	与EVENTC的协作 .....	205
7.4.8	各模式的输出设置.....	205
7.5	使用定时器A时的注意事项 .....	206
7.5.1	计数的开始和停止控制 .....	206
7.5.2	标志的存取(TACR0寄存器的TEDGF位和TUNDF位) .....	206
7.5.3	计数寄存器的存取.....	206
7.5.4	模式的变更 .....	206
7.5.5	TAO引脚和TAIO引脚的设置步骤.....	207
7.5.6	不使用定时器A的情况.....	207
7.5.7	定时器A运行时钟的停止 .....	207
7.5.8	深度睡眠模式(事件计数器模式)的设置步骤.....	208
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 .....	208
7.5.10	通过TSTOP位进行强制的计数停止 .....	208
7.5.11	数字滤波器 .....	208
7.5.12	选择FIL作为计数源的情况 .....	208
<b>第8章</b>	<b>定时器B .....</b>	<b>209</b>
8.1	定时器B的功能.....	209
8.2	定时器B的结构.....	210
8.3	控制定时器B的寄存器 .....	211
8.3.1	外围允许寄存器1(PER1) .....	212
8.3.2	定时器B模式寄存器(TBMR).....	213
8.3.3	定时器B计数控制寄存器(TBCNTC) .....	214
8.3.4	定时器B控制寄存器(TBCR) .....	215
8.3.5	定时器B中断允许寄存器(TBIER) .....	216
8.3.6	定时器B状态寄存器(TBSR) .....	217
8.3.7	定时器B/O控制寄存器(TBIOR) .....	219
8.3.8	定时器B计数器(TB) .....	221
8.3.9	定时器B通用寄存器A、B、C、D.....	222
8.3.10	端口寄存器和端口模式寄存器.....	224
8.4	定时器B的运行.....	225
8.4.1	有关多个模式和功能的共同事项 .....	225
8.4.2	定时器模式(输入捕捉功能) .....	230
8.4.3	定时器模式(输出比较功能) .....	233
8.4.4	PWM模式 .....	237
8.4.5	相位计数模式.....	241
8.5	定时器B中断 .....	245
8.6	使用定时器B时的注意事项 .....	247
8.6.1	相位计数模式中的相位差、重叠和脉宽 .....	247
8.6.2	模式的切换 .....	247
8.6.3	计数源的切换.....	247
8.6.4	TBIO0引脚和TBIO1引脚的设置步骤.....	248
8.6.5	外部时钟TBCLK0和TBCLK1 .....	248
8.6.6	SFR的读写存取 .....	249
8.6.7	停止计数时的输入捕捉运行 .....	249
<b>第9章</b>	<b>定时器C .....</b>	<b>250</b>
9.1	定时器C的功能.....	250
9.2	定时器C的结构.....	251
9.3	控制定时器C的寄存器 .....	252

9.3.1	外围允许寄存器1(PER1) .....	252
9.3.2	定时器C计数寄存器(TC).....	253
9.3.3	定时器C计数缓冲寄存器(TCBUF).....	253
9.3.4	定时器C控制寄存器1(TCCR1).....	254
9.3.5	定时器C控制寄存器2(TCCR2).....	255
9.3.6	定时器C状态寄存器(TCSR).....	255
9.4	定时器C的运行.....	256
9.4.1	计数源 .....	256
9.4.2	定时器C开始计数的动作.....	256
9.4.2.1	选择Timer M的信号作为触发时的设置和动作 .....	257
9.4.2.2	选择软件触发时的设置和动作 .....	258
9.4.3	定时器C计数停止的动作.....	259
9.4.3.1	选择比较器1作为触发时的设置和动作.....	259
9.4.3.2	软件触发时的设置和动作 .....	259
9.4.4	输入捕捉动作.....	260
9.4.5	定时器C计数复位动作 .....	261
9.4.6	定时器C的中断 .....	263
9.5	定时器C使用时的注意事项.....	264
9.5.1	寄存器的读写.....	264
9.5.2	上溢中断.....	264
9.5.3	输入捕捉和定时器C计数复位动作.....	264
9.5.4	定时器C和定时器M, 比较器1联动时的步骤.....	264
<b>第10章</b>	<b>定时器M.....</b>	<b>265</b>
10.1	定时器M的功能 .....	265
10.2	定时器M的结构 .....	266
10.3	控制定时器M的寄存器 .....	267
10.3.1	外围允许寄存器1(PER1) .....	268
10.3.2	定时器M EVENTC 寄存器 (TMELC).....	269
10.3.3	定时器M启动寄存器(TMSTR).....	270
10.3.4	定时器M模式寄存器(TMMR) .....	271
10.3.5	定时器M PWM功能选择寄存器(TMPMR) .....	272
10.3.6	定时器M功能控制寄存器(TMFCR).....	273
10.3.7	定时器 M 输出主允许寄存器1(TMOER1).....	274
10.3.8	定时器M输出主允许寄存器2(TMOER2).....	275
10.3.9	定时器M输出控制寄存器(TMOCR).....	276
10.3.10	定时器M数字滤波器功能选择寄存器i(TMDFi)(i=0、1) .....	279
10.3.11	定时器M延时控制寄存器 (TMDR) .....	281
10.3.12	定时器M控制寄存器i(TMCRi)(i=0、1).....	282
10.3.13	定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1) .....	287
10.3.14	定时器MI/O控制寄存器Ci(TMIORCi)(i=0、1).....	289
10.3.15	定时器M状态寄存器0(TMSR0) .....	291
10.3.16	定时器M状态寄存器1(TMSR1).....	295
10.3.17	定时器M中断允许寄存器i(TMIERi)(i=0、1).....	301
10.3.18	定时器MPWM功能输出电平控制寄存器i(TMPOCRi)(i=0、1).....	302
10.3.19	定时器M计数器i(TMi)(i=0、1).....	303
10.3.20	定时器M通用寄存器Ai、Bi、Ci、Di(TMGR <i>Ai</i> 、TMGR <i>Bi</i> 、TMGR <i>Ci</i> 、TMGR <i>Di</i> )(i=0、1).....	305
10.3.21	端口模式寄存器(PMxx, PMCxx).....	313
10.4	有关多个模式的共同事项 .....	314

10.4.1	计数源 .....	314
10.4.2	缓冲器运行 .....	315
10.4.3	同步运行 .....	318
10.4.4	脉冲输出的强制截止 .....	319
10.4.5	从事件联动控制器(EVENTC)输入的事件 .....	321
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 .....	322
10.5	定时器M的运行 .....	323
10.5.1	输入捕捉功能 .....	323
10.5.2	输出比较功能 .....	327
10.5.3	PWM功能 .....	334
10.5.4	复位同步PWM模式 .....	338
10.5.5	互补PWM模式 .....	341
10.5.6	PWM3模式 .....	346
10.6	定时器M中断 .....	350
10.7	使用定时器M时的注意事项 .....	352
10.7.1	SFR的读写存取 .....	352
10.7.2	模式的切换 .....	352
10.7.3	计数源 .....	353
10.7.4	输入捕捉功能 .....	353
10.7.5	TMIOAi、TMIOBi、TMIOCi、TMIODi引脚的设置步骤(i=0、1) .....	353
10.7.6	外部时钟TMCLK .....	354
10.7.7	互补PWM模式 .....	354
10.8	PWMOP .....	358
10.8.1	PWMOP的功能 .....	359
10.8.2	PWMOP的寄存器 .....	359
10.8.3	PWMOP的运行 .....	365
10.8.3.1	输出强制截止 .....	365
10.8.3.2	硬件解除(HS_SEL=0) .....	366
10.8.3.3	软件解除(HS_SEL=1) .....	375
10.8.3.4	Hazard对策 .....	381
10.8.3.5	输出强制截止源检出状态和未检出状态 .....	382
10.8.3.6	定时器M的计数器的值到达0000H时的时序图 .....	383
10.8.3.7	设置步骤 .....	385
10.8.4	注意事项 .....	386
<b>第11章</b>	<b>时钟输出/蜂鸣器输出控制电路 .....</b>	<b>387</b>
11.1	时钟输出/蜂鸣器输出控制电路的功能 .....	387
11.2	时钟输出/蜂鸣器输出控制电路的结构 .....	388
11.3	控制时钟输出/蜂鸣器输出控制电路的寄存器 .....	388
11.3.1	时钟输出选择寄存器n(CKSn) .....	388
11.3.2	控制时钟输出/蜂鸣器输出引脚端口功能的寄存器 .....	390
11.4	时钟输出/蜂鸣器输出控制电路的运行 .....	391
11.4.1	输出引脚的运行 .....	391
11.5	时钟输出/蜂鸣器输出控制电路的注意事项 .....	391
<b>第12章</b>	<b>看门狗定时器 .....</b>	<b>392</b>
12.1	看门狗定时器的功能 .....	392
12.2	看门狗定时器的结构 .....	392
12.3	控制看门狗定时器的寄存器 .....	394
12.3.1	看门狗定时器的允许寄存器(WDTE) .....	394



12.3.2	LOCKUP控制寄存器(LOCKCTL)及其保护寄存器(PRCR)	395
12.3.3	WDTCFG配置寄存器(WDTCFG0/1/2/3)	396
12.4	看门狗定时器的运行	397
12.4.1	看门狗定时器的运行控制	397
12.4.2	看门狗定时器上溢时间的设置	398
12.4.3	看门狗定时器窗口打开期间的设置	399
12.4.4	看门狗定时器间隔中断的设置	400
12.4.5	WDTCFG未配置时看门狗定时器的运行	400
<b>第13章</b>	<b>A/D转换器</b>	<b>401</b>
13.1	A/D转换器的功能	401
13.2	控制A/D转换器的寄存器	403
13.2.1	外围允许寄存器0(PER0)	404
13.2.2	内部基准电压选择(VREFCTL)	405
13.2.3	A/D转换器的模式寄存器0(ADM0)	406
13.2.4	A/D转换器的模式寄存器1(ADM1)	409
13.2.5	A/D转换器的模式寄存器2(ADM2)	410
13.2.6	A/D转换器的触发模式寄存器(ADTRG)	411
13.2.7	模拟输入通道指定寄存器(ADS)	412
13.2.8	12位A/D转换结果寄存器(ADCR)	414
13.2.9	8位A/D转换结果寄存器(ADCRH)	415
13.2.10	转换结果比较上限值设置寄存器(ADUL)	416
13.2.11	转换结果比较下限值设置寄存器(ADLL)	416
13.2.12	A/D采样时间控制寄存器(ADNSMP)	417
13.2.13	A/D采样时间延长寄存器(ADSMPWAIT)	418
13.2.14	A/D测试寄存器(ADTES)	419
13.2.15	A/D充放电控制寄存器(ADNDIS)	419
13.2.16	控制模拟输入引脚端口功能的寄存器	420
13.3	输入电压和转换结果	421
13.4	A/D转换器的运行模式	422
13.4.1	软件触发模式(选择模式、连续转换模式)	422
13.4.2	软件触发模式(选择模式、单次转换模式)	423
13.4.3	软件触发模式(扫描模式、连续转换模式)	424
13.4.4	软件触发模式(扫描模式、单次转换模式)	425
13.4.5	硬件触发无等待模式(选择模式、连续转换模式)	426
13.4.6	硬件触发无等待模式(选择模式、单次转换模式)	427
13.4.7	硬件触发无等待模式(扫描模式、连续转换模式)	428
13.4.8	硬件触发无等待模式(扫描模式、单次转换模式)	429
13.4.9	硬件触发等待模式(选择模式、连续转换模式)	430
13.4.10	硬件触发等待模式(选择模式、单次转换模式)	431
13.4.11	硬件触发等待模式(扫描模式、连续转换模式)	432
13.4.12	硬件触发等待模式(扫描模式、单次转换模式)	433
13.5	A/D转换器的设置流程图	434
13.5.1	软件触发模式的设置	434
13.5.2	硬件触发无等待模式的设置	435
13.5.3	硬件触发等待模式的设置	436
13.5.4	选择温度传感器的输出电压/内部基准电压时的设置	437
13.5.5	测试模式的设置	438
<b>第14章</b>	<b>D/A转换器</b>	<b>439</b>

14.1	D/A转换器的功能 .....	439
14.2	D/A转换器的结构 .....	440
14.3	控制D/A转换器的寄存器.....	441
14.3.1	外围允许寄存器1(PER1) .....	441
14.3.2	内部基准控制寄存器(VREFCTL) .....	442
14.3.3	D/A转换器的模式寄存器(DAM).....	442
14.3.4	D/A转换值设置寄存器i(DACSi)(i=0、1).....	443
14.3.5	事件输出目标选择寄存器n(ELSELRn)、n=00~22 .....	444
14.3.6	控制模拟输入引脚端口功能的寄存器 .....	444
14.4	D/A转换器的运行 .....	445
14.4.1	通常模式的运行 .....	445
14.4.2	实时输出模式的运行 .....	446
14.4.3	D/A转换值的输出时序.....	447
14.5	使用D/A转换器时的注意事项 .....	448
<b>第15章</b>	<b>比较器.....</b>	<b>449</b>
15.1	比较器的功能 .....	449
15.2	比较器的结构 .....	450
15.3	控制比较器的寄存器 .....	451
15.3.1	外围允许寄存器1(PER1) .....	452
15.3.2	内部基准电压选择 (VREFCTL) .....	452
15.3.3	比较器模式设置寄存器(COMPMDR) .....	453
15.3.4	比较器滤波控制寄存器(COMPFR).....	454
15.3.5	比较器输出控制寄存器(COMPOCR) .....	456
15.3.6	比较器0的输入信号选择控制寄存器(CMPSEL0) .....	458
15.3.7	比较器1的输入信号选择控制寄存器(CMPSEL1) .....	459
15.3.8	比较器0的迟滞控制寄存器 (CMP0HY) .....	460
15.3.9	比较器1的迟滞控制寄存器 (CMP1HY) .....	461
15.3.10	控制模拟输入引脚端口功能的寄存器 .....	461
15.4	运行说明 .....	462
15.4.1	比较器i的数字滤波器(i=0、1) .....	464
15.4.2	比较器i中断(i=0、1).....	465
15.4.3	向联动控制器(EVENTC)输出的事件信号 .....	466
15.4.4	比较器i的输出(i=0、1).....	467
15.4.5	比较器时钟的停止和提供.....	467
<b>第16章</b>	<b>可编程增益放大器(PGA).....</b>	<b>468</b>
16.1	可编程增益放大器的功能 .....	468
16.2	可编程增益放大器的结构 .....	468
16.3	可编程增益放大器的寄存器.....	469
16.3.1	外围允许寄存器1(PER1) .....	469
16.3.2	内部基准电压选择 (VREFCTL) .....	469
16.3.3	可编程增益放大器控制寄存器(PGA0CTL) .....	470
16.3.4	控制模拟输入引脚端口功能的寄存器 .....	471
16.4	可编程增益放大器的运行 .....	472
16.4.1	可编程增益放大器的开始运行步骤.....	472
16.4.2	可编程增益放大器的停止运行步骤.....	473
<b>第17章</b>	<b>通用串行通信单元.....</b>	<b>474</b>
17.1	通用串行通信单元的功能 .....	475
17.1.1	3线串行I/O (SSPI00、SSPI20) .....	475

17.1.2	UART (UART0、UART2)	476
17.1.3	简易I <sup>2</sup> C (IIC00、IIC20)	477
17.2	通用串行通信单元的结构	478
17.2.1	移位寄存器	480
17.2.2	串行数据寄存器mn (SDRmn)	480
17.3	控制通用串行通信单元的寄存器	482
17.3.1	外围允许寄存器0 (PER0)	484
17.3.2	串行时钟选择寄存器m (SPSm)	485
17.3.3	串行模式寄存器mn (SMRmn)	486
17.3.4	串行通信运行设定寄存器mn (SCRmn)	488
17.3.5	串行数据寄存器mn (SDRmn)	491
17.3.6	串行标志清除触发寄存器mn (SIRmn)	492
17.3.7	串行状态寄存器mn (SSRmn)	493
17.3.8	串行通道开始寄存器m (SSm)	495
17.3.9	串行通道停止寄存器m (STm)	496
17.3.10	串行通道允许状态寄存器m (SEm)	497
17.3.11	串行输出允许寄存器m (SOEm)	498
17.3.12	串行输出寄存器m (SOM)	499
17.3.13	串行输出电平寄存器m (SOLm)	500
17.3.14	串行待机控制寄存器m (SSCm)	501
17.3.15	从属选择功能启用寄存器m (SSEm)	502
17.3.16	输入切换控制寄存器 (ISC)	503
17.3.17	噪声滤波器允许寄存器0 (NFEN0)	504
17.3.18	控制串行输入/输出引脚端口功能的寄存器	505
17.4	运行停止模式	506
17.4.1	以单元为单位停止运行的情况	506
17.4.2	按通道停止运行的情况	507
17.5	3线串行I/O (SSPI00、SSPI20) 通信的运行	508
17.5.1	主控发送	509
17.5.2	主控接收	517
17.5.3	主控的发送和接收	526
17.5.4	从属发送	535
17.5.5	从属接收	543
17.5.6	从属的发送和接收	549
17.5.7	传送时钟频率的计算	558
17.5.8	在3线串行I/O (SSPI00、SSPI20) 通信过程中发生错误时的处理步骤	560
17.6	从属选择输入功能的时钟同步串行通信的运行	561
17.6.1	从属发送	564
17.6.2	从属接收	574
17.6.3	从属的发送和接收	581
17.6.4	传送时钟频率的计算	591
17.6.5	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	592
17.7	UART (UART0、UART2) 通信的运行	593
17.7.1	UART发送	594
17.7.2	UART接收	602
17.7.3	低功耗UART模式功能	609
17.7.4	波特率的计算	615
17.7.5	在UART (UART0、UART2) 通信过程中发生错误时的处理步骤	619
17.8	LIN通信的运行	620

17.8.1	LIN发送 .....	620
17.8.2	LIN接收 .....	623
17.9	简易I <sup>2</sup> C (IIC00、IIC20) 通信的运行 .....	628
17.9.1	地址段发送 .....	629
17.9.2	数据发送 .....	634
17.9.3	数据接收 .....	637
17.9.4	停止条件的产生 .....	641
17.9.5	传送速率的计算 .....	642
17.9.6	在简易I <sup>2</sup> C (IIC00、IIC20) 通信过程中发生错误时的处理步骤 .....	644
<b>第18章</b>	<b>串行接口IICA .....</b>	<b>645</b>
18.1	串行接口IICA的功能 .....	645
18.1.1	运行停止模式 .....	645
18.1.2	I <sup>2</sup> C总线模式 (支持多主控) .....	645
18.1.3	唤醒模式 .....	645
18.2	串行接口IICA的结构 .....	648
18.2.1	IICA移位寄存器n (IICAn) .....	649
18.2.2	从属地址寄存器n (SVAn) .....	650
18.2.3	SO锁存器 .....	650
18.2.4	唤醒控制电路 .....	650
18.2.5	串行时钟计数器 .....	650
18.2.6	中断请求信号发生电路 .....	650
18.2.7	串行时钟控制电路 .....	650
18.2.8	串行时钟等待控制电路 .....	651
18.2.9	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 .....	651
18.2.10	数据保持时间校正电路 .....	651
18.2.11	开始条件生成电路 .....	651
18.2.12	停止条件生成电路 .....	651
18.2.13	总线状态检测电路 .....	651
18.3	控制串行接口IICA的寄存器 .....	652
18.3.1	外围允许寄存器0 (PER0) .....	653
18.3.2	IICA控制寄存器n0 (IICCTLn0) .....	653
18.3.3	IICA状态寄存器n (IICSn) .....	658
18.3.4	IICA标志寄存器n (IICFn) .....	661
18.3.5	IICA控制寄存器n1 (IICCTLn1) .....	663
18.3.6	IICA低电平宽度设定寄存器n (IICWLn) .....	665
18.3.7	IICA高电平宽度设定寄存器n (IICWHn) .....	665
18.3.8	端口模式寄存器x (PMx) .....	666
18.4	I <sup>2</sup> C总线模式的功能 .....	667
18.4.1	引脚结构 .....	667
18.4.2	通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法 .....	668
18.5	I <sup>2</sup> C总线的定义和控制方法 .....	670
18.5.1	开始条件 .....	671
18.5.2	地址 .....	672
18.5.3	传送方向的指定 .....	672
18.5.4	应答 (ACK) .....	673
18.5.5	停止条件 .....	674
18.5.6	等待 .....	675
18.5.7	等待的解除方法 .....	677

18.5.8	中断请求 (INTIICAn) 的产生时序和等待控制 .....	678
18.5.9	地址匹配的检测方法 .....	679
18.5.10	错误的检测 .....	679
18.5.11	扩展码 .....	680
18.5.12	仲裁 .....	681
18.5.13	唤醒功能 .....	683
18.5.14	通信预约 .....	686
18.5.15	其他注意事项 .....	689
18.5.16	通信运行 .....	690
18.5.17	I <sup>2</sup> C中断请求 (INTIICAn) 的产生时序 .....	698
18.6	时序图 .....	719
<b>第19章</b>	<b>串行接口SPI .....</b>	<b>734</b>
19.1	串行接口SPI的功能 .....	734
19.2	串行接口SPI的结构 .....	734
19.3	控制串行接口SPI的寄存器 .....	735
19.3.1	外围允许寄存器2 (PER2) .....	736
19.3.2	SPI操作模式寄存器 (SPIMn) .....	737
19.3.3	SPI时钟选择寄存器 (SPICn) .....	738
19.3.4	SPI状态寄存器 (SPISn) .....	739
19.3.5	发送缓冲寄存器 (SDROn) .....	740
19.3.6	接收缓冲寄存器 (SDRIIn) .....	740
19.3.7	SPI引脚的端口功能的控制寄存器 .....	741
19.4	串行接口SPI的操作 .....	742
19.4.1	主控的发送和接收 .....	743
19.4.2	主控的接收 .....	746
19.4.3	从属的发送和接收 .....	749
19.4.4	从属的接收 .....	752
<b>第20章</b>	<b>LIN/UART模块(LIN) .....</b>	<b>755</b>
20.1	概述 .....	755
20.2	寄存器描述 .....	760
20.2.1	主机模式的LIN寄存器 .....	762
20.2.1.1	输入切换控制寄存器 (ISC) .....	762
20.2.1.2	外围寄存器2 (PER2) .....	762
20.2.1.3	LIN时钟选择寄存器 (LINCKSEL) .....	763
20.2.1.4	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) .....	763
20.2.1.5	LIN唤醒波特率选择寄存器 (LWBRn) .....	764
20.2.1.6	LIN/UART波特率预分频器0寄存器 (LBRPn0) .....	765
20.2.1.7	LIN/UART波特率预分频器1寄存器 (LBRPn1) .....	765
20.2.1.8	LIN自测控制寄存器 (LSTCn) .....	766
20.2.1.9	LIN/UART模式寄存器 (LMDn) .....	767
20.2.1.10	LIN间隔场配置寄存器/UART配置寄存器 (LBFCn) .....	768
20.2.1.11	LIN/UART空间配置寄存器 (LSCn) .....	769
20.2.1.12	LIN唤醒配置寄存器 (LWUPn) .....	770
20.2.1.13	LIN中断允许寄存器 (LIEn) .....	771
20.2.1.14	LIN/UART错误检测寄存器 (LEDEn) .....	772
20.2.1.15	LIN/UART控制寄存器 (LCUCn) .....	774
20.2.1.16	LIN/UART传输控制寄存器 (LTRCn) .....	775
20.2.1.17	LIN/UART模式状态寄存器 (LMSTn) .....	776

20.2.1.18	LIN/UART状态寄存器 (LSTn)	777
20.2.1.19	LIN/UART错误状态寄存器 (LESTn)	779
20.2.1.20	LIN/UART数据场配置寄存器(LDFCn)	781
20.2.1.21	LIN/UARTID缓冲寄存器 (LIDBn)	783
20.2.1.22	LIN校验和缓冲寄存器 (LCBRn)	784
20.2.1.23	LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)	785
20.2.2	从机模式的LIN寄存器	786
20.2.2.1	输入切换控制寄存器 (ISC)	786
20.2.2.2	外围寄存器2 (PER2)	787
20.2.2.3	LIN时钟选择寄存器 (LINCKSEL)	788
20.2.2.4	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)	789
20.2.2.5	LIN唤醒波特率选择寄存器 (LWBRn)	789
20.2.2.6	LIN/UART波特率预分频器寄存器 (LBRPn)	790
20.2.2.7	LIN自测控制寄存器 (LSTCn)	791
20.2.2.8	LIN/UART模式寄存器 (LMDn)	792
20.2.2.9	LIN间隔场配置寄存器/UART配置寄存器 (LBFCn)	793
20.2.2.10	LIN/UART空间配置寄存器 (LSCn)	794
20.2.2.11	LIN唤醒配置寄存器 (LWUPn)	795
20.2.2.12	LIN中断允许寄存器 (LIEn)	796
20.2.2.13	LIN/UART错误检测寄存器 (LEDEn)	797
20.2.2.14	LIN/UART控制寄存器 (LCUCn)	799
20.2.2.15	LIN/UART传输控制寄存器 (LTRCn)	800
20.2.2.16	LIN/UART模式状态寄存器 (LMSTn)	801
20.2.2.17	LIN/UART状态寄存器 (LSTn)	802
20.2.2.18	LIN/UART错误状态寄存器 (LESTn)	804
20.2.2.19	LIN/UART数据场配置寄存器(LDFCn)	806
20.2.2.20	LIN/UARTID缓冲寄存器 (LIDBn)	808
20.2.2.21	LIN校验和缓冲寄存器 (LCBRn)	809
20.2.2.22	LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)	810
20.2.3	UART的寄存器	811
20.2.3.1	输入切换控制寄存器 (ISC)	811
20.2.3.2	外围寄存器2 (PER2)	812
20.2.3.3	LIN时钟选择寄存器 (LINCKSEL)	813
20.2.3.4	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)	814
20.2.3.5	LIN唤醒波特率选择寄存器 (LWBRn)	814
20.2.3.6	LIN/UART波特率预分频器寄存器 (LBRPn)	815
20.2.3.7	UART待机控制寄存器 (LUSCn)	816
20.2.3.8	LIN/UART模式寄存器 (LMDn)	817
20.2.3.9	LIN间隔场配置寄存器/UART配置寄存器 (LBFCn)	818
20.2.3.10	LIN/UART空间配置寄存器 (LSCn)	820
20.2.3.11	LIN/UART错误检测寄存器 (LEDEn)	821
20.2.3.12	LIN/UART控制寄存器 (LCUCn)	822
20.2.3.13	LIN/UART传输控制寄存器 (LTRCn)	823
20.2.3.14	LIN/UART模式状态寄存器 (LMSTn)	824
20.2.3.15	LIN/UART状态寄存器 (LSTn)	825
20.2.3.16	LIN/UART错误状态寄存器 (LESTn)	827
20.2.3.17	LIN/UART数据场配置寄存器(LDFCn)	829
20.2.3.18	LIN/UARTID缓冲寄存器 (LIDBn)	830
20.2.3.19	UART数据缓冲区0寄存器 (LUDBn0)	831

20.2.3.20	LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)	832
20.2.3.21	UART操作允许寄存器 (LUOERn)	833
20.2.3.22	UART选项寄存器1 (LUORn1)	834
20.2.3.23	UART传输数据寄存器 (LUTDRn)	836
20.2.3.24	UART接收数据寄存器 (LURDRn)	837
20.2.3.25	UART等待传输数据寄存器 (LUWTRn)	838
20.3	模式	839
20.3.1	LIN复位模式	841
20.3.2	LIN模式	842
20.3.2.1	LIN工作模式	844
20.3.2.2	LIN唤醒模式	844
20.3.3	UART模式	845
20.3.4	LIN自测模式	845
20.4	LIN模式	846
20.4.1	操作概述	846
20.4.1.1	LIN主机模式	846
20.4.1.2	LIN从机模式	849
20.4.2	数据传输/接收	854
20.4.2.1	数据传输	854
20.4.2.2	数据接收	855
20.4.3	传输/接收数据缓冲	856
20.4.3.1	LIN帧的传输	856
20.4.3.2	接收LIN帧	857
20.4.3.3	多字节响应传输/接收功能	858
20.4.4	唤醒传输/接收	859
20.4.4.1	唤醒传输	859
20.4.4.2	唤醒接收	860
20.4.5	状态	861
20.4.6	错误状态	863
20.4.6.1	LIN主机模式	863
20.4.6.2	LIN从机模式	866
20.5	UART模式	868
20.5.1	操作概述	868
20.5.1.1	传输	868
20.5.1.2	接收	872
20.5.1.3	扩展位	873
20.5.1.4	传输起始等待功能	876
20.5.1.5	SNOOZE模式的功能	877
20.5.2	数据传输/接收	883
20.5.2.1	数据传输	883
20.5.2.2	数据接收	884
20.5.3	传输数据的缓冲处理	885
20.5.3.1	UART缓冲传输	885
20.5.4	状态	886
20.5.5	错误状态	887
20.6	LIN自测模式	888
20.6.1	转换到LIN自测模式	889
20.6.2	LIN主机自测模式下的传输	890
20.6.3	LIN主机自测模式下的接收	891

20.6.4	LIN从机自测模式下的传输 .....	892
20.6.5	LIN从机自检模式下的接收 .....	893
20.6.6	终止LIN自测模式 .....	894
20.7	波特率发生器 .....	895
20.7.1	LIN主机模式 .....	895
20.7.2	LIN从机模式 .....	897
20.7.3	UART模式 .....	899
20.8	噪声滤波器 .....	900
20.9	中断 .....	902
<b>第21章</b>	<b>IrDA .....</b>	<b>903</b>
21.1	IrDA的功能 .....	903
21.2	控制IrDA的寄存器 .....	904
21.2.1	外围允许寄存器0(PER0) .....	904
21.2.2	IrDA控制寄存器(IRCR) .....	905
21.3	IrDA的运行 .....	906
21.3.1	IrDA通信的操作步骤 .....	906
21.3.2	发送 .....	907
21.3.3	接收 .....	907
21.3.4	高电平脉宽的选择 .....	908
21.4	使用IrDA时的注意事项 .....	908
<b>第22章</b>	<b>增强型DMA .....</b>	<b>909</b>
22.1	DMA的功能 .....	909
22.2	DMA的结构 .....	911
22.3	控制DMA的寄存器 .....	912
22.3.1	DMA控制数据区和DMA向量表区的分配 .....	913
22.3.2	控制数据的分配 .....	914
22.3.3	向量表 .....	916
22.3.4	外围允许寄存器1 (PER1) .....	918
22.3.5	DMA控制寄存器j (DMACRj) (j=0~39) .....	919
22.3.6	DMA块大小寄存器j (DMBSLj) (j=0~39) .....	920
22.3.7	DMA传送次数寄存器j (DMACTj) (j=0~39) .....	921
22.3.8	DMA传送次数重加载寄存器j (DMRLDj) (j=0~39) .....	922
22.3.9	DMA源地址寄存器j (DMSARj) (j=0~39) .....	922
22.3.10	DMA目标地址寄存器j (DMDARj) (j=0~39) .....	923
22.3.11	DMA启动允许寄存器i (DMAENi) (i=0~4) .....	924
22.3.12	DMAENi置位寄存器 (DMSETi) .....	926
22.3.13	DMAENi复位寄存器 (DMCLRi) .....	926
22.3.14	DMA基址寄存器 (DMABAR) .....	927
22.4	DMA的运行 .....	928
22.4.1	启动源 .....	928
22.4.2	正常模式 .....	929
22.4.3	重复模式 .....	932
22.4.4	链传送 .....	936
22.5	使用DMA时的注意事项 .....	938
22.5.1	DMA控制数据和向量表的设置 .....	938
22.5.2	DMA控制数据区和DMA向量表区的分配 .....	938
22.5.3	DMA的执行时钟数 .....	939
22.5.4	DMA的响应时间 .....	940



22.5.5	DMA的启动源 .....	940
22.5.6	待机模式中的运行 .....	941
<b>第23章</b>	<b>联动控制器(EVENTC) .....</b>	<b>942</b>
23.1	EVENTC的功能 .....	942
23.2	EVENTC的结构 .....	942
23.3	控制寄存器 .....	943
23.3.1	输出目标选择寄存器n (ELSELRn) (n=00~22) .....	944
23.4	EVENTC的运行 .....	947
<b>第24章</b>	<b>中断功能 .....</b>	<b>949</b>
24.1	中断功能的种类 .....	949
24.2	中断源和结构 .....	949
24.3	控制中断功能的寄存器 .....	955
24.3.1	中断请求标志寄存器 (IF00~IF31) .....	955
24.3.2	中断屏蔽标志寄存器 (MK00~MK31) .....	956
24.3.3	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) .....	959
24.4	中断处理的操作 .....	961
24.4.1	可屏蔽中断请求的接受 .....	961
24.4.2	不可屏蔽中断请求的接受 .....	961
<b>第25章</b>	<b>待机功能 .....</b>	<b>962</b>
25.1	待机功能 .....	962
25.2	睡眠模式 .....	963
25.2.1	睡眠模式的设置 .....	963
25.2.2	睡眠模式的解除 .....	966
25.3	深度睡眠模式 .....	966
25.3.1	深度睡眠模式的设置 .....	966
25.3.2	深度睡眠模式的解除 .....	968
<b>第26章</b>	<b>复位功能 .....</b>	<b>970</b>
26.1	复位时序 .....	972
26.2	确认复位源的寄存器 .....	975
26.2.1	复位控制标志寄存器 (RESF) .....	975
<b>第27章</b>	<b>上电复位电路 .....</b>	<b>978</b>
27.1	上电复位电路的功能 .....	978
27.2	上电复位电路的结构 .....	979
27.3	上电复位电路的运行 .....	980
<b>第28章</b>	<b>电压检测电路 .....</b>	<b>983</b>
28.1	电压检测电路的功能 .....	983
28.2	电压检测电路的结构 .....	984
28.3	控制电压检测电路的寄存器 .....	985
28.3.1	电压检测寄存器 (LVIM) .....	985
28.3.2	电压检测电平寄存器 (LVIS) .....	986
28.4	电压检测电路的运行 .....	989
28.4.1	用作复位模式时的设置 .....	989
28.4.2	用作中断模式时的设置 .....	991
28.4.3	用作中断&复位模式时的设置 .....	993
28.5	电压检测电路的注意事项 .....	999
<b>第29章</b>	<b>安全功能 .....</b>	<b>1001</b>
29.1	安全功能的概要 .....	1001

29.2	安全功能使用的寄存器.....	1002
29.3	安全功能的运行.....	1002
29.3.1	闪存CRC运算功能（高速CRC）.....	1002
29.3.1.1	闪存CRC控制寄存器（CRC0CTL）.....	1003
29.3.1.2	闪存CRC运算结果寄存器（PGCRCL）.....	1004
29.3.2	CRC运算功能（通用CRC）.....	1006
29.3.2.1	CRC输入寄存器（CRCIN）.....	1006
29.3.2.2	CRC数据寄存器（CRCD）.....	1007
29.3.3	RAM奇偶校验错误检测功能.....	1008
29.3.3.1	RAM奇偶校验错误控制寄存器（RPECTL）.....	1008
29.3.4	SFR保护功能.....	1010
29.3.4.1	SFR保护控制寄存器（SFRGD）.....	1010
29.3.5	频率检测功能.....	1011
29.3.5.1	定时器输入选择寄存器0（TIS0）.....	1011
29.3.6	A/D测试功能.....	1012
29.3.6.1	A/D测试寄存器（ADTES）.....	1014
29.3.6.2	模拟输入通道指定寄存器（ADS）.....	1014
29.3.7	输入/输出引脚的数字输出信号电平检测功能.....	1015
29.3.7.1	端口模式选择寄存器（PMS）.....	1015
29.3.8	产品唯一身份标识寄存器.....	1016
<b>第30章</b>	<b>温度传感器和内部参考电压.....</b>	<b>1017</b>
30.1	温度传感器.....	1017
30.2	温度传感器的寄存器.....	1017
30.2.1	温度传感器校准数据寄存器 TSN25.....	1017
30.2.2	温度传感器校准数据寄存器 TSN125.....	1017
30.3	温度传感器的使用说明.....	1018
30.3.1	温度传感器的使用原理.....	1018
30.3.2	温度传感器的使用方法.....	1019
30.4	内部参考电压.....	1020
30.4.1	VDD校准数据寄存器VDDCDR.....	1020
30.4.2	内部参考电压的使用说明.....	1020
<b>第31章</b>	<b>选项字节.....</b>	<b>1021</b>
31.1	选项字节的功能.....	1021
31.1.1	用户选项字节（000C0H~000C2H）.....	1021
31.1.2	闪存数据保护选项字节（000C3H, 000C4H）.....	1022
31.2	用户选项字节的格式.....	1023
31.3	闪存数据保护选项字节的格式.....	1029
<b>第32章</b>	<b>FLASH控制.....</b>	<b>1030</b>
32.1	FLASH控制功能描述.....	1030
32.2	FLASH存储器结构.....	1030
32.3	控制FLASH的寄存器.....	1031
32.3.1	Flash写保护寄存器（FLPROT）.....	1031
32.3.2	FLASH操作控制寄存器（FLOPMD1,FLOPMD2）.....	1032
32.3.3	Flash擦除控制寄存器（FLERMD）.....	1032
32.3.4	Flash状态寄存器（FLSTS）.....	1033
32.3.5	Flash全片擦除时间控制寄存器（FLCERCNT）.....	1033
32.3.6	Flash页擦除时间控制寄存器（FLSERCNT）.....	1034
32.3.7	Flash 写入时间控制寄存器（FLPROCNT）.....	1035

32.3.8	Flash擦写保护控制寄存器 (FLSECPR) .....	1036
32.4	FLASH操作方法.....	1037
32.4.1	页擦除 (sector erase) .....	1037
32.4.2	全片擦除 (chip erase) .....	1038
32.4.3	编程 (word program) .....	1038
32.5	闪存读取 .....	1038
32.6	FLASH操作的注意事项 .....	1038
<b>附录</b>	<b>修订记录 .....</b>	<b>1039</b>

Draft Version  
Cmsemicon Confidential

# 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。

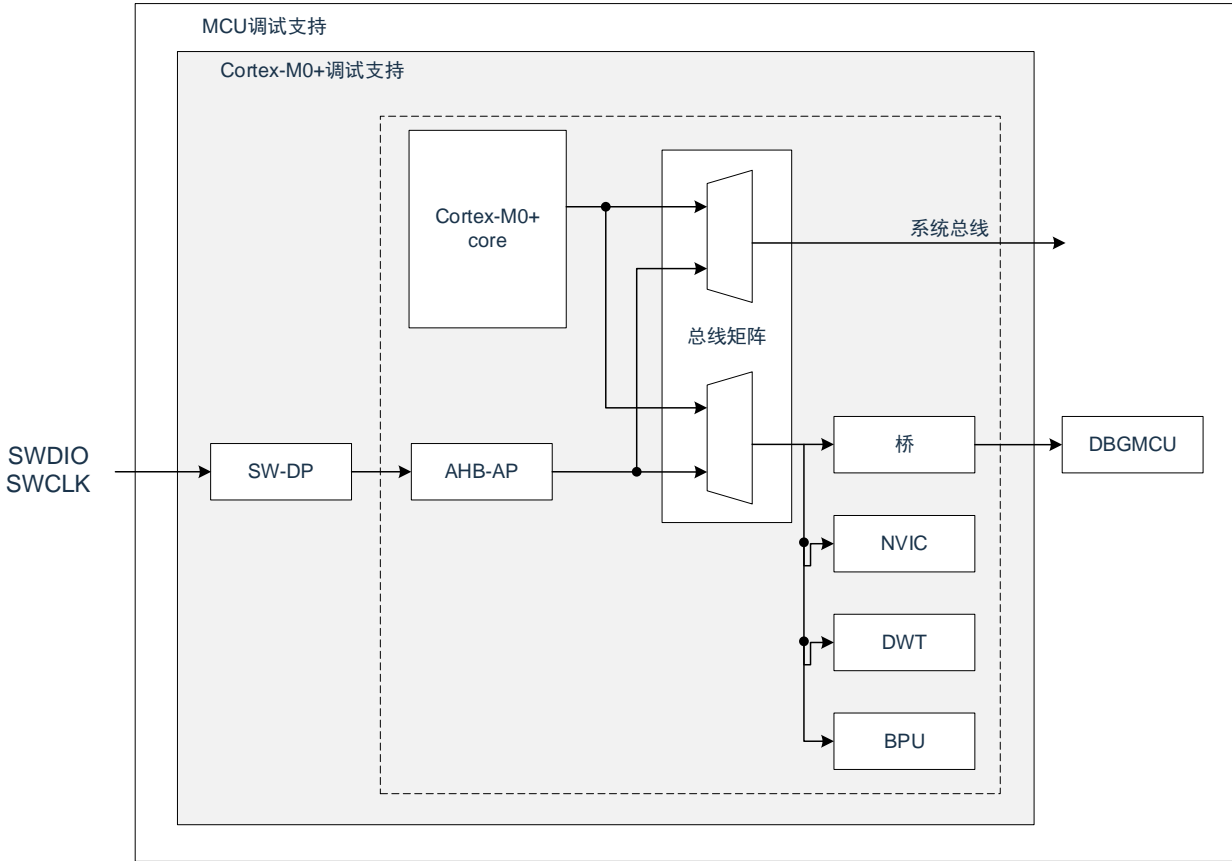
## 1.2 Cortex-M0+内核特性

- 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
- 单周期硬件乘法器
- 8周期硬件除法器
- 嵌套向量中断控制器(NVIC)
  - 1个不可屏蔽中断(NMI)
  - 支持28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<sub>CLK</sub>或F<sub>IL</sub>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

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	-	-	-	-	-	RESMSK	FRZEN1	FRZEN0
默认值	0	0	0	0	0	1	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

RESMSK	在调试器连接的状态下，屏蔽 WDT 复位和 RAM 校验错误复位
0	不屏蔽 WDT 复位和 RAM 校验错误复位
1	屏蔽 WDT 复位和 RAM 校验错误复位

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 <sup>1</sup>
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 <sup>2</sup>
0	周边动作
1	周边停止

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。
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

Draft Version  
Cmsemicon Confidential

## 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。

Draft Version  
Cmsemicon Confidential



## 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)
- 端口置位控制寄存器 (PSETxx)
- 端口清零控制寄存器 (PCLRxx)
- 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)
- 端口回读寄存器 (PREADxx)
- 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，请参照下表。必须给未分配的位设定初始值。

表 2-1 各产品分配的寄存器及其位(1/2)

端口		位名									32 引脚	24 引脚	20 引脚
		PMxx 寄存器	Pxx 寄存器	PSETxx 寄存器	PCLRxx 寄存器	PUxx 寄存器	PIMxx 寄存器	POMxx 寄存器	PMCxx 寄存器	PREADxx 寄存器			
端口0	0	PM00	P00	PSET00	PCLR00	PU00	PIM00	POM00	PMC00	PREAD00	○	○	○
	1	PM01	P01	PSET01	PCLR01	PU01	PIM01	POM01	PMC01	PREAD01	○	○	○
端口1	0	PM10	P10	PSET10	PCLR10	PU10	PIM10	POM10	PMC10	PREAD10	○	○	○
	1	PM11	P11	PSET11	PCLR11	PU11	—	POM11	PMC11	PREAD11	○	○	○
	2	PM12	P12	PSET12	PCLR12	PU12	—	—	PMC12	PREAD12	○	○	○
	3	PM13	P13	PSET13	PCLR13	PU13	—	POM13	—	PREAD13	○	○	○
	4	PM14	P14	PSET14	PCLR14	PU14	PIM14	POM14	—	PREAD14	○	○	—
	5	PM15	P15	PSET15	PCLR15	PU15	PIM15	POM15	—	PREAD15	○	○	—
	6	PM16	P16	PSET16	PCLR16	PU16	PIM16	—	—	PREAD16	○	—	—
端口2	0	PM17	P17	PSET17	PCLR17	PU17	PIM17	POM17	—	PREAD17	○	—	—
	0	PM20	P20	PSET20	PCLR20	—	—	—	PMC20	PREAD20	○	○	○
	1	PM21	P21	PSET21	PCLR21	—	—	—	PMC21	PREAD21	○	○	○
	2	PM22	P22	PSET22	PCLR22	—	—	—	PMC22	PREAD22	○	○	○
	3	PM23	P23	PSET23	PCLR23	—	—	—	PMC23	PREAD23	○	—	—

表2-1 各产品分配的寄存器及其位(2/2)

端口		位名									32 引脚	24 引脚	20 引脚
		PMxx 寄存器	Pxx 寄存器	PSETxx 寄存器	PCLRxx 寄存器	PUxx 寄存器	PIMxx 寄存器	POMxx 寄存器	PMCxx 寄存器	PREADxx 寄存器			
端口3	0	PM30	P30	PSET30	PCLR30	PU30	PIM30	POM30	—	PREAD30	○	—	—
	1	PM31	P31	PSET31	PCLR31	PU31	PIM31	POM31	—	PREAD31	○	—	—
端口4	0	PM40	P40	PSET40	PCLR40	PU40	—	—	—	PREAD40	○	○	○
端口5	0	PM50	P50	PSET50	PCLR50	PU50	PIM50	POM50	—	PREAD50	○	○	○
	1	PM51	P51	PSET51	PCLR51	PU51	—	POM51	—	PREAD51	○	○	○
端口7	0	PM70	P70	PSET70	PCLR70	PU70	—	—	—	PREAD70	○	—	—
	2	PM72	P72	PSET72	PCLR72	PU72	PIM72	POM72	—	PREAD72	○	○	○
	3	PM73	P73	PSET73	PCLR73	PU73	—	—	—	PREAD73	○	○	○
	4	PM74	P74	PSET74	PCLR74	PU74	PIM74	POM74	—	PREAD74	○	—	—
端口12	0	PM120	P120	PSET120	PCLR120	PU120	—	—	PMC120	PREAD120	○	—	—
	1	—	P121	—	—	—	—	—	—	PREAD121	○	○	○
	2	—	P122	—	—	—	—	—	—	PREAD122	○	○	○
端口13	6	PM136	P136	PSET136	PCLR136	PU136	—	—	PMC136	PREAD136	○	○	—
	7	PM137	P137	PSET137	PCLR137	PU137	—	—	—	PREAD137	○	○	○
端口14	7	PM147	P147	PSET147	PCLR147	PU147	—	—	PMC147	PREAD147	○	○	—

### 2.3.1 端口模式寄存器 (PMxx)

这是以位为单位设定端口输入/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定”进行设定。

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	1	1	1	1	1	PM01	PM00	0x320	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	0x321	FFH	R/W
PM2	1	1	1	1	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	1	1	1	PM40	0x324	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	0x325	FFH	R/W
PM7	1	1	1	PM74	PM73	PM72	1	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	1	0x32D	FFH	R/W
PM14	PM147	1	1	1	1	1	1	1	0x32E	FFH	R/W

PMmn	Pmn引脚的输入/输出模式的选择
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定初始值。

## 2.3.2 端口寄存器 (Pxx)

这是以位为单位设定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	0	0	0	0	0	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	0	0	0	0	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	0	0	0	P40	0x304	00H (输出锁存器)	R/W
P5	0	0	0	0	0	0	P51	P50	0x305	00H (输出锁存器)	R/W
P7	0	0	0	P74	P73	P72	0	P70	0x307	00H (输出锁存器)	R/W
P12	0	0	0	0	0	P122	P121	P120	0x30C	不定值	R/W 注1
P13	P137	P136	0	0	0	0	0	0	0x30D	00H (输出锁存器)	R/W
P14	P147	0	0	0	0	0	0	0	0x30E	00H (输出锁存器)	R/W

Pmn	Pmn引脚的输入/输出	
	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1: P121~P122是只读位。

注2: 必须给未分配的位设定初始值。

### 2.3.3 端口置位控制寄存器 (PSETxx)

这是以位为单位来置位端口输出锁存器的寄存器。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；端口置位控制寄存器的基址为0x40040000，偏址见下图。

图2-3 端口置位控制寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PSET0	0	0	0	0	0	0	PSET01	PSET00	0x080	00H	W
PSET1	PSET17	PSET16	PSET15	PSET14	PSET13	PSET12	PSET11	PSET10	0x081	00H	W
PSET2	0	0	0	0	PSET23	PSET22	PSET21	PSET20	0x082	00H	W
PSET3	0	0	0	0	0	0	PSET31	PSET30	0x083	00H	W
PSET4	0	0	0	0	0	0	0	PSET40	0x084	00H	W
PSET5	0	0	0	0	0	0	PSET51	PSET50	0x085	00H	W
PSET7	0	0	0	PSET74	PSET73	PSET72	0	PSET70	0x087	00H	W
PSET12	0	0	0	0	0	0	0	PSET120	0x08C	00H	W
PSET13	PSET137	PSET136	0	0	0	0	0	0	0x08D	00H	W
PSET14	PSET147	0	0	0	0	0	0	0	0x08E	00H	W

PSETmn	Pmn端口的置位控制
0	无操作
1	对应的Pmn置1

注意：必须给未分配的位设定初始值。

### 2.3.4 端口清零控制寄存器 (PCLRxx)

这是以位为单位来置位端口输出锁存器的寄存器。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；端口清零控制寄存器的基址为0x40040000，偏址见下图。

图2-4 端口清零控制寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PCLR0	0	0	0	0	0	0	PCLR01	PCLR00	0x090	00H	W
PCLR1	PCLR17	PCLR16	PCLR15	PCLR14	PCLR13	PCLR12	PCLR11	PCLR10	0x091	00H	W
PCLR2	0	0	0	0	PCLR23	PCLR22	PCLR21	PCLR20	0x092	00H	W
PCLR3	0	0	0	0	0	0	PCLR31	PCLR30	0x093	00H	W
PCLR4	0	0	0	0	0	0	0	PCLR40	0x094	00H	W
PCLR5	0	0	0	0	0	0	PCLR51	PCLR50	0x095	00H	W
PCLR7	0	0	0	PCLR74	PCLR73	PCLR72	0	PCLR70	0x097	00H	W
PCLR12	0	0	0	0	0	0	0	PCLR120	0x09C	00H	W
PCLR13	PCLR137	PCLR136	0	0	0	0	0	0	0x09D	00H	W
PCLR14	PCLR147	0	0	0	0	0	0	0	0x09E	00H	W

PCLRmn	Pmn端口的清零控制
0	无操作
1	对应的Pmn清零

注意：必须给未分配的位设定初始值。

### 2.3.5 上拉电阻选择寄存器 (PUxx)

内部上拉电阻的选择寄存器。只能对设定为输入模式 (PMmn=1) 并且POMmn位为“0”的引脚选择使用内部上拉电阻。对于设定为输出模式的位, 与上拉电阻选择寄存器的设定无关, 不连接内部上拉电阻。当用作复用功能的输出引脚或者设定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 2-5 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	0	0	0	0	0	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	0	0	0	PU40	0x034	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	0x035	00H	R/W
PU7	0	0	0	PU74	PU73	PU72	0	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	0	0	0	0	0	0	0	0x03E	00H	R/W

PUmn	Pmn引脚的内部上拉电阻的选择
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定初始值。

### 2.3.6 端口输入模式寄存器 (PIMxx)

这是以位为单位设定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 2-6 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	0	0	0	PIM01	PIM00	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	PIM30	0x043	00H	R/W
PIM5	0	0	0	0	0	0	0	PIM50	0x045	00H	R/W
PIM7	0	0	0	PIM74	0	PIM72	0		0x047	00H	R/W

PIMmn	Pmn引脚的输入缓冲器的选择
0	施密特输入缓冲器
1	TTL输入缓冲器

注：必须给未分配的位设定初始值。



### 2.3.7 端口输出模式寄存器 (POMxx)

这是以位为单位设定输出模式的寄存器。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I<sup>2</sup>C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。

图 2-7 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	0	0	0	POM01	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	POM30	0x053	00H	R/W
POM5	0	0	0	0	0	0	POM51	POM50	0x055	00H	R/W
POM7	0	0	0	POM74	0	POM72	0	0	0x057	00H	R/W

POMmn	Pmn引脚的输出模式的选择
0	通常的输出模式
1	N沟道漏极开路输出模式

注：必须给未分配的位设定初始值。

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

### 2.3.8 端口模式控制寄存器 (PMCxx)

PMC寄存器以位为单位设定的数字输入/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 2-8 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	1	1	1	PMC01	PMC00	0x060	FFH	R/W
PMC1	1	1	1	1	1	1	PMC11	PMC10	0x061	FFH	R/W
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20	0x062	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	0x06C	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	0x06E	FFH	R/W

PMCmn	Pmn引脚的数字输入/输出或者模拟输入的选择
0	数字输入/输出（模拟输入以外的复用功能）
1	模拟输入

注：必须给未分配的位设定初始值。

### 2.3.9 端口回读寄存器 (PREADxx)

这是一个只读寄存器，在端口作为普通数字GPIO时，可以通过读此寄存器得到端口电平。

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2-9 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
PREAD0	0	0	0	0	0	0	PREAD01	PREAD00	0x0A0	xxH	R
PREAD1	PREAD17 PREAD16 PREAD15 PREAD14 PREAD13 PREAD12 PREAD11 PREAD10								0x0A1	xxH	R
PREAD2	0	0	0	0	PREAD23 PREAD22 PREAD21			PREAD20	0x0A2	xxH	R
PREAD3	0	0	0	0	0	0	PREAD31	PREAD30	0x0A3	xxH	R
PREAD4	0	0	0	0	0	0	0	PREAD40	0x0A4	xxH	R
PREAD5	0	0	0	0	0	0	PREAD51	PREAD50	0x0A5	xxH	R
PREAD7	0	0	0	PREAD74 PREAD73		PREAD72	0	PREAD70	0x0A7	xxH	R
PREAD12	0	0	0	0	0	PREAD122 PREAD121		PREAD120	0x0AC	xxH	R
PREAD13	PREAD137 PREAD136		0	0	0	0	0	0	0x0AD	xxH	R
PREAD14	PREAD147		0	0	0	0	0	0	0x0AE	xxH	R

注：复位后该寄存器的读出值取决于各端口状态。

PREADmn	Pmn端口状态回读
	输出模式/输入模式
0	端口为低电平。
1	端口为高电平。

## 2.3.10 外围I/O重定向寄存器0 (PIOR0)

这是设定允许或者禁止外围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 2-10 外围I/O重定向寄存器0 (PIOR0) 的格式

地址: 0x40040877 复位后: 00H R/W

PIOR0	7	6	5	4	3	2	1	0
	0	PIOR06	PIOR05	0	0	PIOR02	PIOR01	0

PIOR0	功能	32pin/32pin_E		24pin/24pin_E		20pin	
		设定值		设定值		设定值	
		0	1	0	1	0	1
bit6 (PIOR06)	RxD2	P14	P14	P14	P14	-	-
	TxD2	P13	P10	P13	P10	-	-
	SCL20	P15	-	P15	-	-	-
	SDA20	P14	-	P14	-	-	-
	SDI20	P14	-	P14	-	-	-
	SDO20	P13	-	P13	-	-	-
	SCLK20	P15	-	P15	-	-	-
bit5 (PIOR05)	LRXD0	P01	P73	P01	P73	P01	P73
	LTXD0	P00	P72	P00	P72	P00	P72
bit2 (PIOR02)	SCLA0	P31	P14	P01	P14	P01	-
	SDAA0	P74	P15	P00	P15	P00	-
bit1 (PIOR01)	TxD0	P51	P17	P51	-	P51	-
	RxD0	P50	P16	P50	-	P50	-
	SCL00	P30	-	P72	-	P72	-
	SDA00	P50	-	P50	-	P50	-
	SDO00	P51	-	P51	-	P51	-
	SDI00	P50	-	P50	-	P50	-
	SCLK00	P30	-	P72	-	P72	-

### 2.3.11 外围I/O重定向寄存器1 (PIOR1)

这是设定允许或者禁止外围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 2-11 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A TAO引脚的选择
0	0	和P30复用
0	1	和P50复用
1	0	和P00复用
1	1	禁止设定

PIOR11	PIOR10	定时器A TAIO引脚的选择
0	0	和P01复用
0	1	和P31复用
1	0	禁止设定
1	1	禁止设定

## 2.3.12 外围I/O重定向寄存器2 (PIOR2)

这是设定允许或者禁止外围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 2-12 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	32pin		24pin		20pin	
		设定值		设定值		设定值	
		0	1	0	1	0	1
bit7(PIOR27)	TMIOC0	必须保持 0		P16	P13	必须保持 0	
bit6(PIOR26)	TMIOD0	P15	P17	必须保持 0			
bit5(PIOR25)	TMIOD1	P11	P51	P11	P51		
bit4(PIOR24)	TMIOC1	P13	P50	P13	P50		
bit3(PIOR23)	TMIOB1	P10	P30	必须保持 0			
bit2(PIOR22)	TMIOA1	P12	P16	必须保持 0			
bit1(PIOR21)	VCOUT1	P31	P70	P147	P72	P147	P72
bit0(PIOR20)	VCOUT0	P120	P73	P12	P73	P12	P73

### 2.3.13 外围I/O重定向寄存器3 (PIOR3)

这是设定允许或者禁止外围I/O重定向功能寄存器3。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 2-13 外围I/O重定向寄存器3 (PIOR3) 的格式

地址：0x4004087C 复位后：00H R/W

符号 7 6 5 4 3 2 1 0

PIOR3	PIOR37	PIOR36	PIOR35	PIOR34	0	PIOR32	PIOR31	PIOR30
-------	--------	--------	--------	--------	---	--------	--------	--------

PIOR3	功能	32pin		24pin		20pin	
		设定值		设定值		设定值	
		0	1	0	1	0	1
bit5(PIOR35)	TxD0	PIOR34 控制	P12	PIOR34 控制	P12	PIOR34 控制	P12
	RxD0	PIOR34 控制	P11	PIOR34 控制	P11	PIOR34 控制	P11
bit4(PIOR34)	TxD0	PIOR01 控制	P40	PIOR01 控制	P40	PIOR01 控制	P40
	RxD0	PIOR01 控制	P137	PIOR01 控制	P137	PIOR01 控制	P137
bit2(PIOR32)	VCOUT1	输出 L	PIOR21 控制	输出 L	PIOR21 控制	输出 L	PIOR21 控制
bit1(PIOR31)	VCOUT0	输出 L	PIOR20 控制	输出 L	PIOR20 控制	输出 L	PIOR20 控制
bit0(PIOR30)	IrRxD	必须保持 0		P14	P01	P14	P01
	IrTxD			P13	P00	P13	P00

PIOR37	PIOR36	TMIOA0	TMIOB0	TMIOC0	TMIOD0	TMIOA1	TMIOB1	TMIOC1	TMIOD1
0	0	P17	P14	P16	P15	P12	P10	P13	P11
0	1	P17	P12	P16	P15	P11	P10	P14	P13
1	0	P17	P15	P16	P14	P13	P12	P11	P10
1	1	P17	P15	P16	P14	P13	P11	P12	P10

注：x表示该位必须设置为初始值0。

### 2.3.14 全局数字输入禁止寄存器 (GDIDIS)

在切断EV<sub>DD</sub>的电源时，对于以EV<sub>DD</sub>为电源的输入端口，此寄存器防止输入缓冲器的贯通电流。当不使用以EV<sub>DD</sub>为电源的全部输入/输出端口时，能通过将GDIDIS寄存器置“1”来切断EV<sub>DD</sub>的电源，降低功耗。

通过将GDIDIS0位置“1”，禁止以EV<sub>DD</sub>为电源的全部输入缓冲器的输入，防止切断EV<sub>DD</sub>的电源时的贯通电流。

在产生复位信号后，此寄存器的值变为“00H”。

图 2-14 全局数字输入禁止寄存器 (GDIDIS)

地址: 0x4004087D 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
GDIDIS	0	0	0	0	0	0	0	GDIDIS0

GDIDIS0	EV <sub>DD</sub> 电源的输入缓冲器的设定
0	允许输入缓冲器的输入 (默认值)。
1	禁止输入缓冲器的输入。防止流向输入缓冲器的贯通电流。

要切断EV<sub>DD</sub>的电源时，必须按照以下步骤进行设定：

- ① 设定为禁止输入缓冲器的输入 (GDIDIS0=1)。
- ② 切断EV<sub>DD</sub>的电源。

要重新接通EV<sub>DD</sub>的电源时，必须按照以下步骤进行设定：

- ① 接通EV<sub>DD</sub>的电源。
- ② 设定为允许输入缓冲器的输入 (GDIDIS0=0)。

注意1.对于以EV<sub>DD</sub>为电源的输入端口，不能输入超过EV<sub>DD</sub>的电压。

2.如果设定为禁止输入缓冲器的输入 (GDIDIS0=1)，以EV<sub>DD</sub>为电源的端口的端口寄存器 (Pxx) 的读取值就为“1”。当端口输出模式寄存器 (POMxx) 为“1” (N沟道漏极开路输出 (EV<sub>DD</sub>耐压)) 时，端口寄存器 (Pxx) 的读取值为“0”。

备注1.即使设定为禁止输入缓冲器的输入 (GDIDIS0=1)，也能使用以EV<sub>DD</sub>为电源的未使用端口功能的外围功能。



## 2.4 未使用引脚的处理

各未使用引脚的处理如下表所示。

表 2-2 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~P01	输入/输出	输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
P10~P17		输出：置为开路。
P20~P23		输入：单独通过电阻连接V <sub>DD</sub> 或者V <sub>SS</sub> 。
		输出：置为开路。
P30、P31		输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
		输出：置为开路。
P40		输入：单独通过电阻连接EV <sub>DD</sub> 或者置为开路。
	输出：置为开路。	
P50~P51	输入/输出	输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
		输出：置为开路。
P70, P72~P74		输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
P120		输出：置为开路。
P121~P122	输入	单独通过电阻连接V <sub>DD</sub> 或者V <sub>SS</sub> 。
P136	输入/输出	输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
		输入：单独通过电阻连接V <sub>DD</sub> 或者V <sub>SS</sub> 。
		输出：置为开路。
P137	输入/输出	输入：单独通过电阻连接V <sub>DD</sub> 或者置为开路。
		输出：置为开路。
P147	输入/输出	输入：单独通过电阻连接EV <sub>DD</sub> 或者EV <sub>SS</sub> 。
		输出：置为开路。
RESETB	输入	直接或者通过电阻连接V <sub>DD</sub> 。

备注 对于没有EV<sub>DD</sub>、EV<sub>SS</sub>引脚的产品，必须将EV<sub>DD</sub>替换为V<sub>DD</sub>并且将EV<sub>SS</sub>替换为V<sub>SS</sub>。

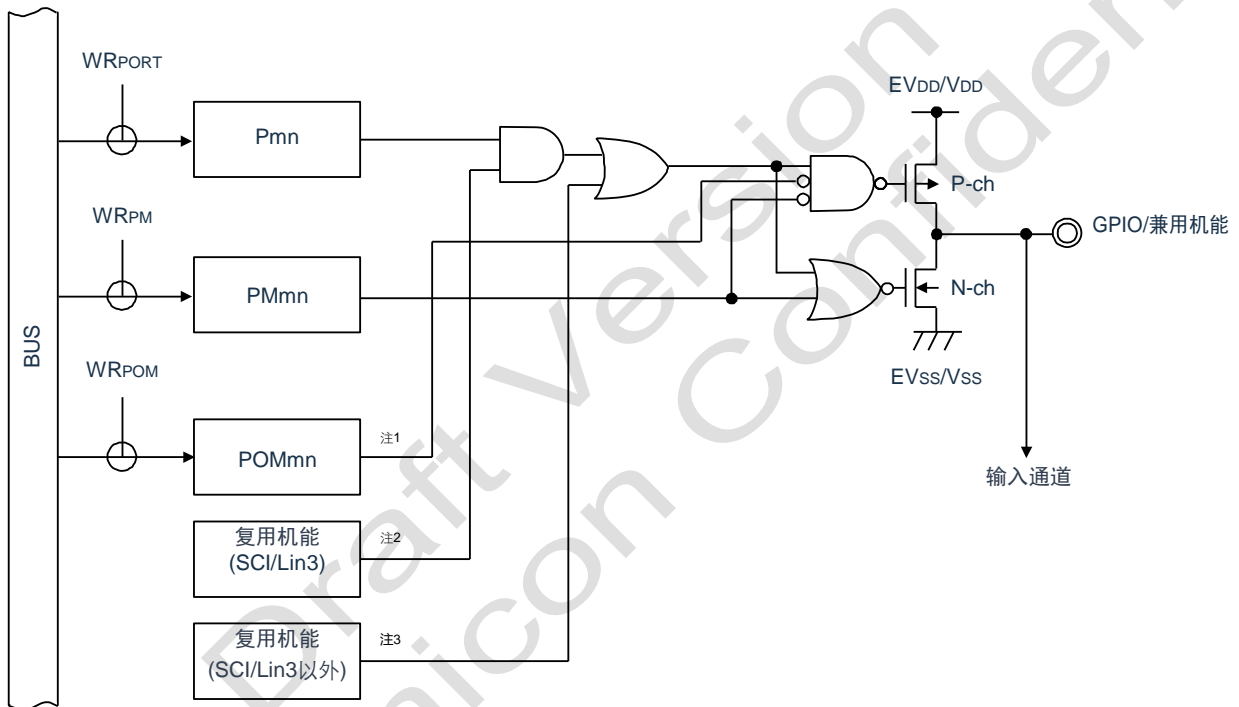
## 2.5 使用复用功能时的寄存器设定

### 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）设定是用作模拟功能还是用作数字输入/输出。

用作数字输入/输出的引脚输出电路的基本结构如图2-15所示。与端口的输出锁存器输出复用的SCI和Lin3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/Lin3功能（定时器、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。此时的设定基本思想如表 2-3 设定的基本思想所示。

图 2-15 引脚的输出电路的基本结构



- 注 1.当没有POM寄存器时，此信号为Low电平（0）。  
 2.当没有复用功能时，此信号为High电平（1）。  
 3.当没有复用功能时，此信号为Low电平（0）。

表 2-3 设定的基本思想

使用的引脚输出功能	不使用的复用功能的输出设定		
	端口功能	SCI/Lin3的输出功能	SCI/Lin3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/Lin3的输出功能	High（1）	—	Low电平输出（0）
SCI/Lin3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/Lin3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）。有关具体的设定方法，请参照“2.5.2 使用的端口功能和复用功能的寄存器设定例子”。

## 2.5.2 使用的端口功能和复用功能的寄存器设定例子

使用的端口功能和复用功能的寄存器设定例子（32引脚制品）如下表所示。

表 2-4 引脚功能寄存器设定例子

引脚名称	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/Lin3的输出功能	SCI/Lin3以外	
P00	P00	输入	—	×	0	1	×	—	—	
		输出	—	0	0	0	0/1	LTXD0=1	(TAO)=0	
		N沟道漏极开路输出		1	0	0	0/1			
	ANI11	模拟输入	—	×	1	1	×	—	—	
	VCIN10	模拟输入	—	×	1	1	×	—	—	
	TI00	输入	—	×	0	1	×	—	—	
	TBCLK0	输入	—	×	0	1	×	—	—	
	LTXD0	输出	PIOR05=0	0	0	0	1	—	(TAO)=0	
	(TAO)	输出	PIOR13、PIOR12=10B	0	0	0	0	LTXD0=1	—	
INTP8	输入	—	×	0	1	×	—	—		
P01	P01	输入	—	×	0	1	×	—	—	
		输出	—	0	0	0	0/1	—	TAIO=0, TO00=0	
		N沟道漏极开路输出		1	0	0	0/1			
	ANI10	模拟输入	—	×	1	1	×	—	—	
	VCIN11	模拟输入	—	×	1	1	×	—	—	
	(PGA1IN)	模拟输入	—	×	1	1	×	—	—	
	TO00	输出	—	0	0	0	0	—	TAIO=0	
	TBCLK1	输入	—	×	0	1	×	—	—	
	TAIO	输入	PIOR11、PIOR10=00B	×	0	0	1	×	—	—
		输出	PIOR11、PIOR10=00B	0	0	0	0	—	TO00=0	
LRXD0	输入	PIOR05=0	×	0	1	×	—	—		
INTP10	输入	—	×	0	1	×	—	—		

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P10	P10	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0/1	(TxD2/IrTxD)=1	TMIOB1/(TMIOD1)=0 SPICLK=0
		N沟道漏极开路输出	—	1	0	0	0/1		
	ANI9	模拟输入	—	×	1	1	×	—	—
	(TxD2)/ (IrTxD)	输出	PIOR30=0, PIOR06=1	0	0	0	1	—	TMIOB1/(TMIOD1)=0 SPICLK=0
	SPICLK	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0	(TxD2/IrTxD)=1	TMIOB1/(TMIOD1)=0
	TMIOB1	输入	PIOR23=0, PIOR37=0, PIOR36=0 或PIOR37=0, PIOR36=1	×	0	1	×	—	—
输出		PIOR23=0, PIOR37=0, PIOR36=0 或PIOR37=0, PIOR36=1	0	0	0	0	(TxD2/IrTxD)=1	(TMIOD1)=0, SPICLK=0	
(TMIOD1)	输入	PIOR37=1, PIOR36=0 或PIOR37=1, PIOR36=1	×	0	1	×	—	—	
	输出	PIOR37=1, PIOR36=0 或PIOR37=1, PIOR36=1	0	0	0	0	(TxD2/IrTxD)=1	TMIOB1=0, SPICLK=0	
P11	P11	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0/1	—	TMIOD1/(TMIOA1)/ (TMIOC1)/(TMIOB1)=0 SO=0
		N沟道漏极开路输出	—	1	0	0	0/1		
	ANI8	模拟输入	—	×	1	1	×	—	—
	SDI00	输入	PIOR35=1	×	0	1	×	—	—
	RxD0	输入	PIOR35=1	×	0	1	×	—	—
	MISO	输入	—	×	0	1	×	—	—
		输出	—	0	0	0	0	—	TMIOD1/(TMIOA1)/ (TMIOC1)/(TMIOB1)=0
	TMIOD1	输入	PIOR25=0, PIOR37=0, PIOR36=0	×	0	1	×	—	—
		输出	PIOR25=0, PIOR37=0, PIOR36=0	0	0	0	0	—	(TMIOA1)/(TMIOC1)/ (TMIOB1)=0 SO=0
	(TMIOA1)	输入	PIOR37=0, PIOR36=1	×	0	1	×	—	—
		输出	PIOR37=0, PIOR36=1	0	0	0	0	—	TMIOD1/(TMIOC1)/ (TMIOB1)=0 SO=0
(TMIOC1)	输入	PIOR37=1, PIOR36=0	×	0	1	×	—	—	
	输出	PIOR37=1, PIOR36=0	0	0	0	0	—	TMIOD1/(TMIOA1)/ (TMIOB1)=0 SO=0	
(TMIOB1)	输入	PIOR37=1, PIOR36=1	×	0	1	×	—	—	
	输出	PIOR37=1, PIOR36=1	0	0	0	0	—	TMIOD1/(TMIOA1)/ (TMIOC1)=0 SO=0	
P12	P12	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	TxD0/SDO00=1	TMIOA1/(TMIOB0)/ (TMIOB1)/(TMIOC1)=0 MO=0
	SDO00	输出	PIOR35=1	—	—	0	1	—	TMIOA1/(TMIOB0)/ (TMIOB1)/(TMIOC1)=0 MO=0
	TxD0	输出	PIOR35=1	—	—	0	1	—	TMIOA1/(TMIOB0)/ (TMIOB1)/(TMIOC1)=0 MO=0
	SIMO	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0	TxD0/SDO00=1	TMIOA1/(TMIOB0)/ (TMIOB1)/(TMIOC1)=0
	TMIOA1	输入	PIOR22=0, PIOR37=0, PIOR36=0	—	—	1	×	—	—
		输出	PIOR22=0, PIOR37=0, PIOR36=0	—	—	0	0	TxD0/SDO00=1	(TMIOB0)/ (TMIOB1)/(TMIOC1)=0 MO=0
	(TMIOB0)	输入	PIOR37=0, PIOR36=1	—	—	1	×	—	—
		输出	PIOR37=0, PIOR36=1	—	—	0	0	TxD0/SDO00=1	TMIOA1/ (TMIOB1)/(TMIOC1)=0 MO=0
(TMIOB1)	输入	PIOR37=1, PIOR36=0	—	—	1	×	—	—	
	输出	PIOR37=1, PIOR36=0	—	—	0	0	TxD0/SDO00=1	TMIOA1/(TMIOB0)/ (TMIOC1)=0 MO=0	
(TMIOC1)	输入	PIOR37=1, PIOR36=1	—	—	1	×	—	—	
	输出	PIOR37=1, PIOR36=1	—	—	0	0	TxD0/SDO00=1	TMIOA1/(TMIOB0)/ (TMIOB1)=0 MO=0	

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P13	P13	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	TxD2/IrTxD/SDO20=1	TMIOC1/(TMIOC0/TMIOD1/TMIOA1)=0
		N沟道漏极开路输出	—	1	—	0	0/1		
	TxD2/IrTxD	输出	PIOR06=0,PIOR30=0	0	—	0	1	—	TMIOC1/(TMIOC0/TMIOD1/TMIOA1)=0
	SDO20	输出	PIOR06=0,PIOR30=0	0	—	0	1	—	
	TMIOC1	输入	PIOR24=0, PIOR37=0, PIOR36=0	×	—	1	×	—	—
		输出	PIOR24=0, PIOR37=0, PIOR36=0	0	—	0	0	TxD2/IrTxD/SDO20=1	(TMIOC0/TMIOD1/TMIOA1)=0
	(TMIOC0)	输入	PIOR27=1, PIOR37=0, PIOR36=0	×	—	1	×	—	—
		输出	PIOR27=1, PIOR37=0, PIOR36=0	0	—	0	0	TxD2/IrTxD/SDO20=1	TMIOC1 / TMIOD1/TMIOA1)=0
	(TMIOD1)	输入	PIOR37=0, PIOR36=1	×	—	1	×	—	—
输出		PIOR37=0, PIOR36=1	0	—	0	0	TxD2/IrTxD/SDO20=1	TMIOC1/(TMIOC0/TMIOA1)=0	
(TMIOA1)	输入	PIOR37=1, PIOR36=0	×	—	1	×	—	—	
	输出	或PIOR37=1, PIOR36=1	0	—	0	0	TxD2/IrTxD/SDO20=1	TMIOC1/(TMIOC0/TMIOD1)=0	
P14	P14	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	SDA20=1	TMIOB0/(TMIOC1/TMIOD0)=0, (SCLA0)=0
		N沟道漏极开路输出	—	1	—	0	0/1		
	RxD2/IrRxD	输入	—	×	—	1	×	—	—
	SDI20	输入	—	×	—	1	×	—	—
	SDA20	输入/输出	—	1	—	0	1	—	TMIOB0/(TMIOC1/TMIOD0)=0, (SCLA0)=0
	TMIOB0	输入	PIOR37=0, PIOR36=0	×	—	1	×	—	—
		输出	PIOR37=0, PIOR36=0	1	—	0	0	SDA20=1	(TMIOC1/TMIOD0)=0 (SCLA0)=0
	TMIOC1	输入	PIOR37=0, PIOR36=1	×	—	1	×	—	—
		输出	PIOR37=0, PIOR36=1	1	—	0	0	SDA20=1	TMIOB0/(TMIOD0)=0 (SCLA0)=0
TMIOD0	输入	PIOR37=1, PIOR36=0	×	—	1	×	—	—	
	输出	或PIOR37=1, PIOR36=1	1	—	0	0	SDA20=1	TMIOB0/(TMIOC1)=0 (SCLA0)=0	
(SCLA0)	输入/输出	PIOR02=1	1	—	0	0	SDA20=1	TMIOB0/(TMIOC1/TMIOD0)=0	
P15	P15	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	SCLK20/SCL20=1	TMIOD0/(TMIOB0)=0 (SDAA0)=0, CLKBUZ1=0
		N沟道漏极开路输出	—	1	—	0	0/1		
	SCLK20	输入	PIOR06=0,PIOR30=0	×	—	1	×	—	—
		输出	PIOR06=0,PIOR30=0	0	—	0	1	—	TMIOD0/(TMIOB0)=0 (SDAA0)=0, CLKBUZ1=0
	SCL20	输出	PIOR06=0,PIOR30=0	1	—	0	1	—	—
	TMIOD0	输入	PIOR26=0, PIOR37=0, PIOR36=0	×	—	1	×	—	—
		输出	或PIOR37=0, PIOR36=1	0	—	0	0	SCLK20/SCL20=1	(TMIOB0)=0, (SDAA0)=0, CLKBUZ1=0
	(TMIOB0)	输入	PIOR37=1, PIOR36=0	×	—	1	×	—	—
		输出	或PIOR37=1, PIOR36=1	0	—	0	0	SCLK20/SCL20=1	TMIOD0 =0, (SDAA0)=0, CLKBUZ1=0
CLKBUZ1	输出	—	0	—	0	0	SCLK20/SCL20=1	(SDAA0)=0, TMIOD0/(TMIOB0)=0	
(SDAA0)	输入/输出	PIOR02=1	1	—	0	0	SCLK20/SCL20=1	TMIOD0/(TMIOB0)=0 CLKBUZ1=0	

引脚名称	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/Lin3的输出功能	SCI/Lin3以外	
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	
	(RxD0)	输入	PIOR01=1、 PIOR34=0,PIOR35=0	—	—	1	×	—	—	
	INTP5	输入	—	—	—	1	×	—	—	
	TMIOC0	输入	PIOR27=0	—	—	—	1	×	—	—
		输出		—	—	0	0	—	(TMIOA1)=0, TO01=0	
(TMIOA1)	输入	PIOR22=1	—	—	—	1	×	—	—	
	输出		—	—	0	0	—	TMIOC0=0, TO01=0		
P17	P17	输入	—	×	—	1	×	—	—	
		输出	—	0	—	0	0/1	(TxD0)=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)=1	TMIOA0=0, (TMIOD0)=0	
	TMIOA0	输入	—	×	—	1	×	—	—	
		输出		0	—	0	0	(TxD0)=1	TO02=0, (TMIOD0)=0	
	TMCLK	输入	—	×	—	1	×	—	—	
(TxD0)	输出	PIOR01=1、PIOR34=0	0	—	0	1	—	TO02=0, TMIOA0=0, (TMIOD0)=0		
(TMIOD0)	输入	PIOR26=1	×	—	—	1	×	—	—	
	输出		0	—	0	0	(TxD0)=1	TO02=0, TMIOA0=0		

引脚名称	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	×
	AV <sub>REFP</sub>	基准电压输入	—	1	1	×
	VCIN12	模拟输入	—	1	1	×
	INTP11	输入	—	0	1	1
P21	P21	输入	—	0	1	×
		输出	—	0	0	0/1
	ANI1	模拟输入	—	1	1	×
	AV <sub>REFM</sub>	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	×

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P30	P30	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	SCLK00/SCL00=1	TAO=0, (TMIOB1)=0
		N沟道漏极开路输出		1	—	0	0/1		
	INTP3	输入	—	×	—	1	×	—	—
	SCLK00	输入	PIOR01=0	×	—	1	×	—	—
		输出		0	—	0	1	—	TAO=0, (TMIOB1)=0
	SCL00	输出	PIOR01=0	1	—	0	1	—	TAO=0, (TMIOB1)=0
	TAO	输出	PIOR13, PIOR12=00B	0	—	0	0	SCLK00/SCL00=1	(TMIOB1)=0
(TMIOB1)	输入	PIOR23=1 PIOR37=0, PIOR36=0	×	—	1	×	—	—	
	输出		0	—	0	0	SCLK00/SCL00=1	TAO=0	
P31	P31	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	—	TO03=0, VCOU1=0, CLKBUZ0=0, (TAIO)=0 SCLA0=0
		N沟道漏极开路输出		1	—	0	0/1		
	TI03	输入	—	×	—	1	×	—	—
	TO03	输出	—	0	—	0	0	—	VCOU1=0, CLKBUZ0=0, (TAIO)=0, SCLA0=0
	INTP4	输入	—	×	—	1	×	—	—
	(TAIO)	输入	PIOR11, PIOR10=01B	×	—	1	×	—	—
		输出		0	—	0	0	—	TO03=0, VCOU1=0, CLKBUZ0=0, SCLA0=0
	CLKBUZ0	输出	—	0	—	0	0	—	TO03=0, (TAIO)=0, VCOU1=0, SCLA0=0
	SS00	输入	—	×	—	1	×	—	—
SCLA0	输入/输出	PIOR02=0	1	—	0	0	—	TO03=0, VCOU1=0 CLKBUZ0=0, (TAIO)=0	
VCOU1	输出	PIOR21=0	0	—	0	0	—	TO03=0, (TAIO)=0, CLKBUZ0=0, SCLA0=0	
P40	P40	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	(TxD0)/(SDO00) =1	—
	(TxD0)	输出	PIOR34=1	—	—	0	1	—	—
	(SDO00)	输出	PIOR34=1	—	—	0	1	—	—
	SWDIO	输入/输出	—	—	—	×	×	—	—



引脚名称	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/Lin3的输出功能	SCI/Lin3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	—	×	—	1	×	—	—
	SDI00	输入	PIOR01=0, PIOR34=0,PIOR35=0	×	—	1	×	—	—
	RxD0	输入	PIOR01=0 PIOR34=0,PIOR35=0	×	—	1	×	—	—
	SDA00	输入/输出	PIOR01=0	1	—	0	1	—	TBIO0=0、 (TAO)=0、 (TMIOC1)=0
	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 PIOR37=0, PIOR36=0	×	—	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	—	×	—	1	×	—	—
	SDO00	输出	PIOR01=0 PIOR34=0,PIOR35=0	0	—	0	1	—	TBIO1=0、 (TMIOD1)=0
	TxD0	输出	PIOR01=0 PIOR34=0,PIOR35=0	0	—	0	1	—	TBIO1=0、 (TMIOD1)=0
	TBIO1	输入	—	×	—	1	×	—	—
		输出		0	—	0	0	TxD0/SDO00=1	(TMIOD1)=0
(TMIOD1)	输入	PIOR25=1 PIOR37=0, PIOR36=0	×	—	1	×	—	—	
	输出		0	—	0	0	TxD0/SDO00=1	TBIO1=0	

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P70	P70	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	(VCOUT1)=0
	INTP6 (VCOUT1)	输入	—	—	—	1	×	—	—
		输出	PIOR21=1	—	—	0	0	—	—
P72	P72	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	LTXD0 =1	—
		N沟道漏极开路输出	—	1	—	0	0/1		
	INTP7	输入	—	×	—	1	×	—	—
LTXD0	输出	PIOR 05=1	0	—	0	1	—	—	
P73	P73	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	(VCOUT0)=0
	LRXD0 (VCOUT0)	输入 输出	PIOR 05=1 PIOR20=1	— —	— —	1 0	×	—	—
P74	P74	输入	—	×	—	1	×	—	—
		输出	—	0	—	0	0/1	—	SDAA0=0
		N沟道漏极开路输出		1	—	0	0/1		
	SDAA0	输入/输出	PIOR02=0	1	—	0	0	—	—

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P120	P120	输入	—	—	0	1	×	—	—
		输出	—	—	0	0	0/1	—	VCOUT0=0
	ANI14	模拟输入	—	—	1	1	×	—	—
	VCOUT0	输出	PIOR20=0	—	0	0	0	—	—

引脚名称	使用的功能		CMC		Pxx
	功能名称	输入/输出	(EXCLK、OSCSEL)		
P121	P121	输入	00/10/11		×
	X1	—	01		—
P122	P122	输入	00/10/11		×
	X2	—	01		—
	EXCLK	输入	11		—

引脚名称	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/Lin3的输出功能	SCI/Lin3以外
P136	P136	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
	INTP0	输入	—	—	—	1	×	—	—
P137	P137	输入	—	—	—	1	×	—	—
		输出	—	—	—	0	0/1	—	—
	SWCLK	输入	—	—	—	1	×	—	—
	(SDI00)	输入	PIOR34=1,PIOR35=0	—	—	1	×	—	—
	(RxD0)	输入	PIOR34=1,PIOR35=0	—	—	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	×	—	—
	PGA1GND	模拟输入	—	—	1	1	×	—	—
	NSS	输入	—	—	0	1	×	—	—

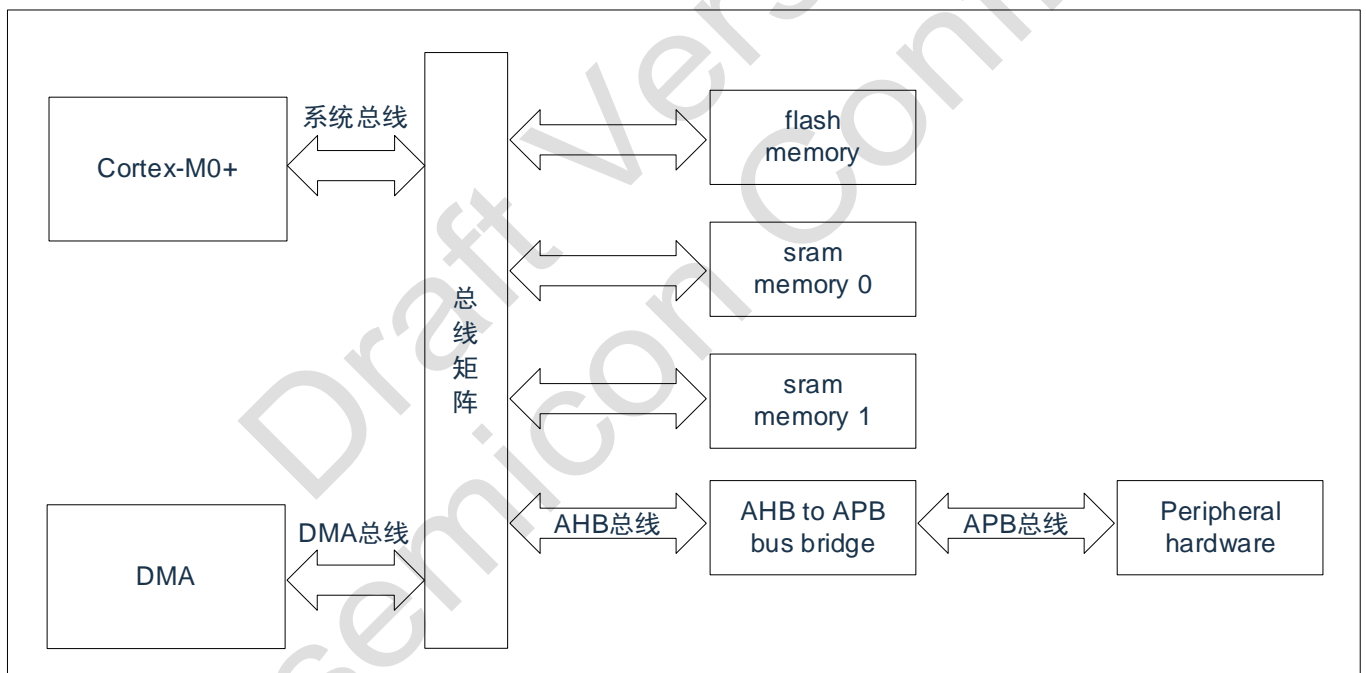
## 第3章 系统结构

### 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

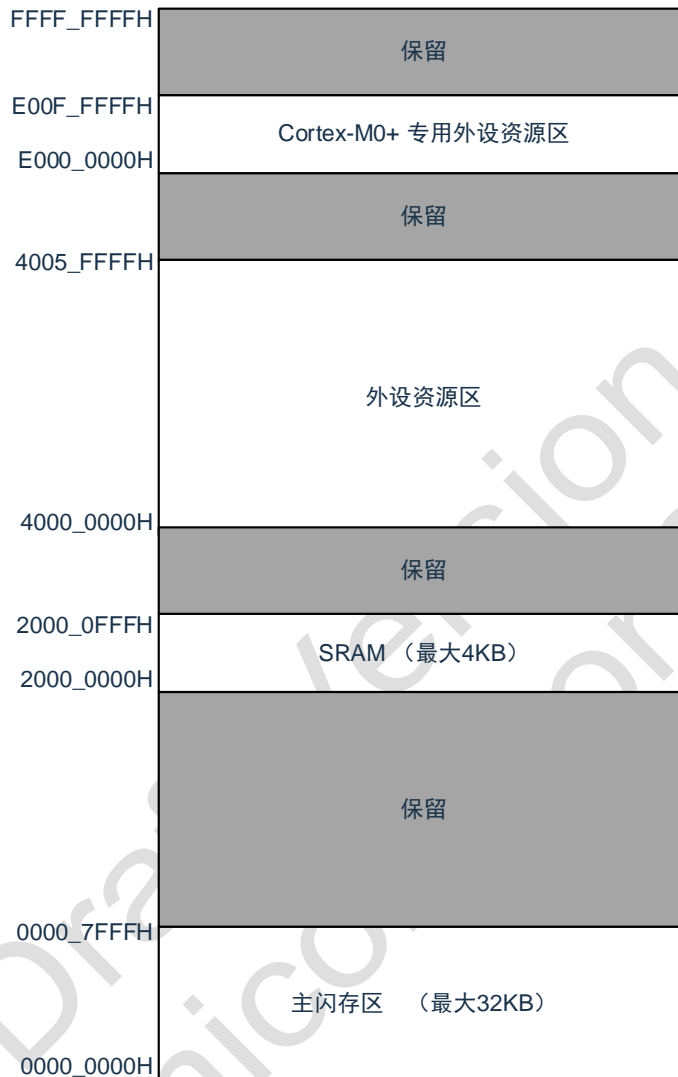
图3-1 系统结构示意图



- 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



外设地址分配如表3-1所示。

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_9FFF	保留	
0x4001_A000 - 0x4001_AFFF	硬件除法器	
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 29 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 29 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_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_5AFF	保留	
0x4004_5B00 - 0x4004_5BFF	中断控制	
0x4004_5C00 - 0x4004_5CFF	LIN 控制器	
0x4004_5D00 - 0x4004_5DFF	高速 SPI	
0x4004_5E00 - 0xDFFF_FFFF	保留	

## 第4章 时钟发生电路

### 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2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)对 $F_{HOCO}$ 进行分频，得到分频时钟 $F_{IH}=64\text{MHz}$ 、 $48\text{MHz}$ 、 $32\text{MHz}$ 、 $24\text{MHz}$ 、 $16\text{MHz}$ 、 $12\text{MHz}$ 、 $8\text{MHz}$ 、 $6\text{MHz}$ 、 $4\text{MHz}$ 、 $3\text{MHz}$ 、 $2\text{MHz}$ 和 $1\text{MHz}$ (TYP.)中选择频率进行振荡。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4-11 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( $F_{EX}=1\sim 20\text{MHz}$ )，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) 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。

SysTick定时器使用低速内部振荡器时钟作为外部参考时钟。

低速内部振荡器时钟能直接给看门狗定时器和定时器A提供时钟，也能用作系统时钟。

当选项字节(000C0H)的bit4(WDTON)为“1”时，低速内部振荡器振荡。

但是，在WDTON位为“1”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。

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)

$F_{IH}$ ： $F_{HOCO}$ 的分频时钟频率

$F_{EX}$ ：外部主系统时钟频率

$F_{IL}$ ：低速内部振荡器的时钟频率

## 4.2 时钟发生电路的结构

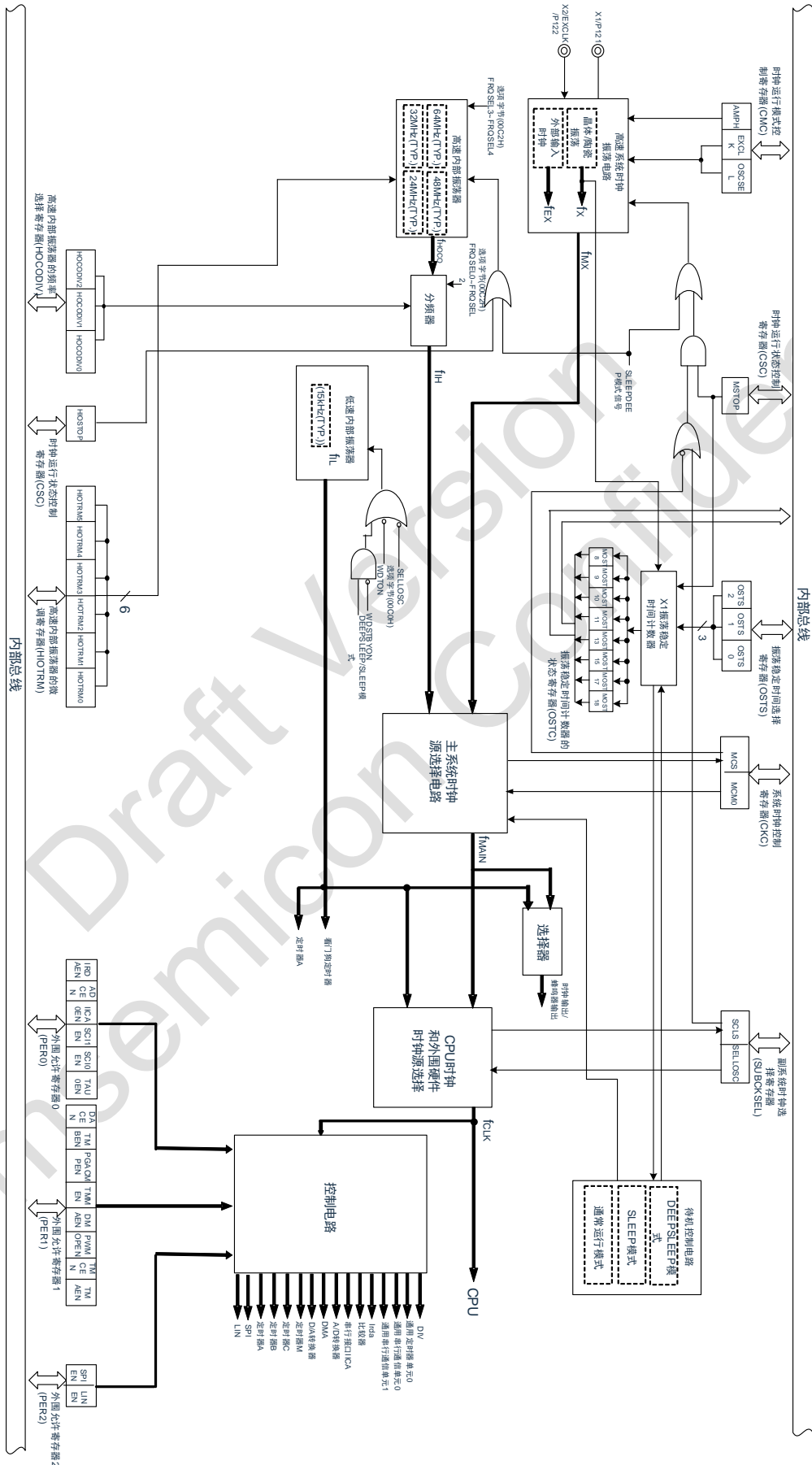
时钟发生电路由以下硬件构成。

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、2(PER0、PER1、PER2) 副系统时钟选择寄存器(SUBCKSEL) 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 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)

$F_{IH}$ :  $F_{HOCO}$ 的分频时钟

$F_{EX}$ : 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

$F_{CLK}$ : CPU/外围硬件的时钟频率

$F_{IL}$ : 低速内部振荡器的时钟频率

Draft Version  
Cmsemicon Confidential

## 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、2(PER0、PER1、PER2)
- 副系统时钟选择寄存器(SUBCKSEL)
- 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)

### 4.3.1 时钟运行模式控制寄存器(CMC)

这是设置X1/P121、X2/EXCLK/P122引脚运行模式的寄存器。

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	0	0	0	0	0	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

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 $F_{MX}$ 前的状态)下设置AMPH位。
5. 系统时钟的频率上限为64MHz，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<sup>3</sup>

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

MCS	主系统时钟(F <sub>MAIN</sub> )的状态
0	高速内部振荡器时钟(F <sub>IH</sub> )
1	高速系统时钟(F <sub>MX</sub> )

MCM0	主系统时钟(F <sub>MAIN</sub> )的运行控制
0	选择高速内部振荡器时钟(F <sub>IH</sub> )作为主系统时钟(F <sub>MAIN</sub> )。
1	选择高速系统时钟(F <sub>MX</sub> )作为主系统时钟(F <sub>MAIN</sub> )。

注: bit5是只读位

备注: F<sub>HOCO</sub>: 高速内部振荡器的时钟频率(最大64MHz)

F<sub>IH</sub>: F<sub>HOCO</sub>的分频时钟频率

F<sub>MX</sub>: 高速系统时钟频率

F<sub>MAIN</sub>: 主系统时钟频率

注意:

1. 必须将bit0~3置“0”。
2. 如果更改CPU时钟,就同时更改外围硬件的时钟(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。

### 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“80H”。

图4-4 时钟运行状态控制寄存器(CSC)的格式

地址: 40020401H 复位后: 80H R/W

符号	7	6	5	4	3	2	1	0
CSC	MSTOP	0	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	

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. 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。
5. 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。	MSTOP=1
外部主系统时钟	(MCS=0)	
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。	HIOSTOP=1
低速内部振荡器时钟	(MCS=1或SCLS=1)	
低速内部振荡器时钟	不使能WDT且不使用低速内部振荡器时钟做系统时钟时自动停止振荡。	-

#### 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

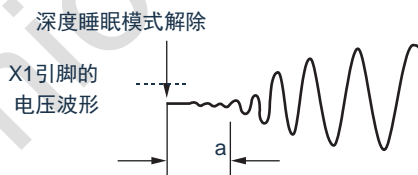
地址: 40020402H 复位后: 00H R

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 <sub>X</sub> =10MHz	F <sub>X</sub> =20MHz	
0	0	0	0	0	0	0	0	小于2 <sup>8</sup> /F <sub>X</sub>	小于25.6us	小于12.8us
1	0	0	0	0	0	0	0	至少2 <sup>8</sup> /F <sub>X</sub>	至少25.6us	至少12.8us
1	1	0	0	0	0	0	0	至少2 <sup>9</sup> /F <sub>X</sub>	至少51.2us	至少25.6us
1	1	1	0	0	0	0	0	至少2 <sup>10</sup> /F <sub>X</sub>	至少102us	至少51.2us
1	1	1	1	0	0	0	0	至少2 <sup>11</sup> /F <sub>X</sub>	至少204us	至少102us
1	1	1	1	1	0	0	0	至少2 <sup>13</sup> /F <sub>X</sub>	至少819us	至少409us
1	1	1	1	1	1	0	0	至少2 <sup>15</sup> /F <sub>X</sub>	至少3.27ms	至少1.63ms
1	1	1	1	1	1	1	0	至少2 <sup>17</sup> /F <sub>X</sub>	至少13.1ms	至少6.55ms
1	1	1	1	1	1	1	1	至少2 <sup>18</sup> /F <sub>X</sub>	至少26.2ms	至少13.1ms

注意:

- 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。
- 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)
- 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<sub>X</sub>: 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

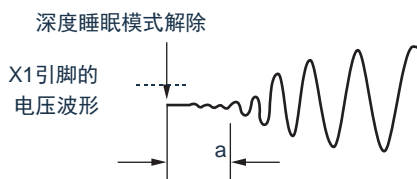
地址: 40020403H 复位后: 07H R/W

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 <sub>X</sub> =10MHz	F <sub>X</sub> =20MHz
0	0	0	2 <sup>8</sup> /F <sub>X</sub>	25.6us	12.8us
0	0	1	2 <sup>9</sup> /F <sub>X</sub>	51.2us	25.6us
0	1	0	2 <sup>10</sup> /F <sub>X</sub>	102us	51.2us
0	1	1	2 <sup>11</sup> /F <sub>X</sub>	204us	102us
1	0	0	2 <sup>13</sup> /F <sub>X</sub>	819us	409us
1	0	1	2 <sup>15</sup> /F <sub>X</sub>	3.27ms	1.63ms
1	1	0	2 <sup>17</sup> /F <sub>X</sub>	13.1ms	6.55ms
1	1	1	2 <sup>18</sup> /F <sub>X</sub>	26.2ms	13.1ms

注意:

- 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。
- 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)
- 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<sub>X</sub>: X1时钟振荡频率



### 4.3.6 外围允许寄存器0、1、2(PER0、PER1、PER2)

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。

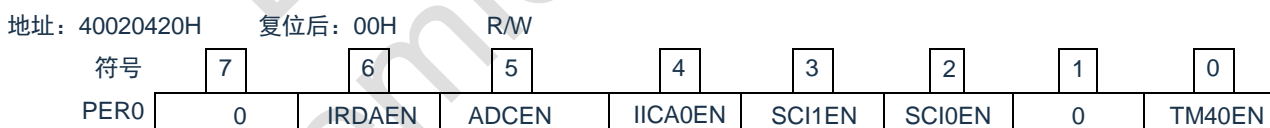
当使用由这些寄存器控制的以下外围功能时，必须在进行外围功能的初始设置前将对应位置“1”。

- 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
- 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
- SPI
- LIN

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



IRDAEN	提供串行接口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。

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。

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

地址: 4002081AH    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

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 <sup>注1</sup>	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。

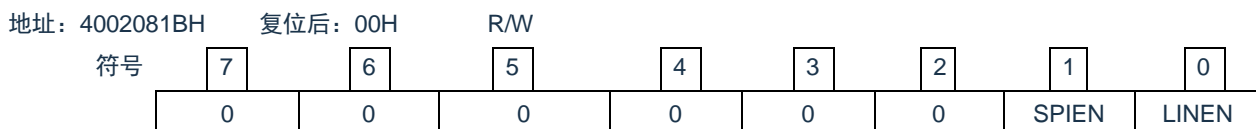
DMAEN	提供DMA的输入时钟的控制
0	停止提供输入时钟。 •DMA不能运行。
1	提供输入时钟。 •DMA能运行。

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 <sup>注2</sup>	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。

图4-9 外围允许寄存器2(PER2)的格式



SPIEN	SPI模块的输入时钟控制
0	停止提供输入时钟。 •不能写SPI模块使用的SFR。 •SPI模块处于复位状态。
1	提供输入时钟。 •能读写SPI模块使用的SFR。

LINEN	LIN模块的输入时钟控制
0	停止提供输入时钟。 •不能写LIN模块使用的SFR。 •LIN模块处于复位状态。
1	提供输入时钟。 •能读写LIN模块使用的SFR。

### 4.3.7 副系统时钟提供模式控制寄存器 (OSMC)

通过OSMC寄存器来选择蜂鸣器和定时器A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	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	蜂鸣器和定时器A的运行时钟的选择
0	<ul style="list-style-type: none"> <li>•不能选择低速内部振荡器作为蜂鸣器时钟。</li> <li>•不能选择低速内部振荡器作为定时器A的计数源。</li> </ul>
1	<ul style="list-style-type: none"> <li>•能选择低速内部振荡器作为蜂鸣器时钟。</li> <li>•能选择低速内部振荡器作为定时器A的计数源。</li> </ul>

### 4.3.8 副系统时钟选择寄存器 (SUBCKSEL)

SUBCKSEL寄存器是选择主系统时钟fMAIN和低速内部振荡器时钟FIL，以及选择低速内部振荡器时钟频率的寄存器。

通过8位存储器操作指令设置SUBCKSEL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-10 副系统时钟选择寄存器 (SUBCKSEL) 的格式

地址: 40020407H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	SCLS	LOCOSEL	SELLOSC

SCLS	系统时钟的状态
0	•主系统时钟
1	•低速内部振荡器时钟

LOCOSEL	低速内部振荡器时钟的频率选择
0	•低速内部振荡器时钟频率为 15K。
1	•低速内部振荡器时钟频率为 30K

SELLOSC	主系统时钟和低速内部振荡器时钟的选择
0	• 选择主系统时钟。
1	• 选择低速内部振荡器时钟。

### 4.3.9 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4-11 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

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	$F_{IH}=24\text{MHz}$	$F_{IH}=32\text{MHz}$	$F_{IH}=48\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=64\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	0	1	$F_{IH}=12\text{MHz}$	$F_{IH}=16\text{MHz}$	$F_{IH}=24\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=32\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	1	0	$F_{IH}=6\text{MHz}$	$F_{IH}=8\text{MHz}$	$F_{IH}=12\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=16\text{MHz}$ $F_{HOCO}=64\text{MHz}$
0	1	1	$F_{IH}=3\text{MHz}$	$F_{IH}=4\text{MHz}$	$F_{IH}=3\text{MHz}$ $F_{HOCO}=48\text{MHz}$	$F_{IH}=8\text{MHz}$ $F_{HOCO}=64\text{MHz}$
1	0	0	禁止设置。	$F_{IH}=2\text{MHz}$	禁止设置。	$F_{IH}=4\text{MHz}$ $F_{HOCO}=64\text{MHz}$
1	0	1	禁止设置。	$F_{IH}=1\text{MHz}$	禁止设置。	$F_{IH}=2\text{MHz}$ $F_{HOCO}=64\text{MHz}$
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。

### 4.3.10 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<sub>DD</sub>引脚的电压发生变化，频率就发生变化。

在温度和V<sub>DD</sub>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4-12 高速内部振荡器的微调寄存器(HIOTRM)的格式

地址：40021800H 复位后：<sup>注</sup> 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。

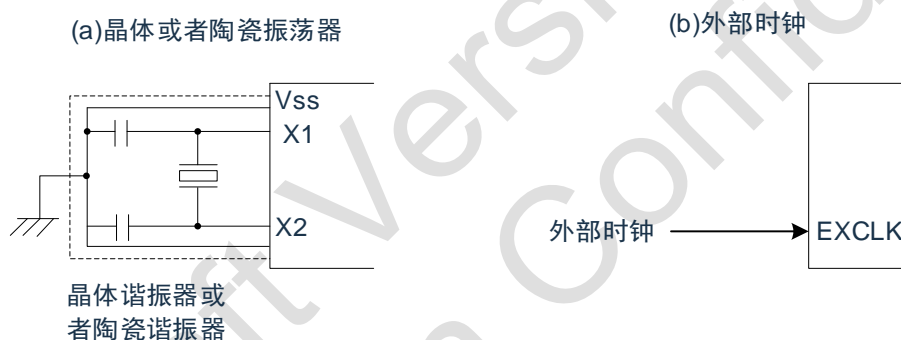
当使用X1振荡电路时,必须对时钟运行模式控制寄存器(CMC)的bit7和bit6(EXCLK、OSCSEL)进行以下的设置:

- 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 2-2 各未使用引脚的处理”。

X1振荡电路的外接电路例子如图4-13所示。

图4-13 X1振荡电路的外接电路例子



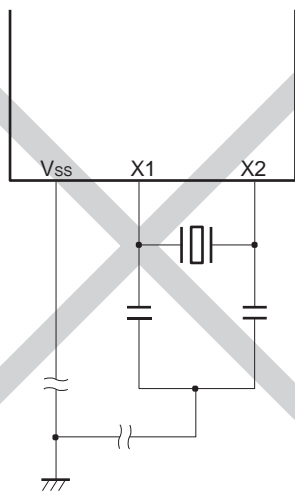
注意:当使用X1振荡电路时,为了避免布线电容等的影响,必须通过以下方法对图4-13中的虚线部分进行布线:

- 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VSS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。

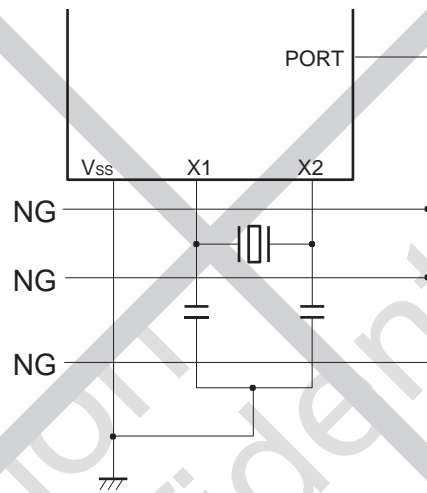
不正确的谐振器连接例子如图4-14所示。

图4-14 不正确的谐振器连接例子(1/2)

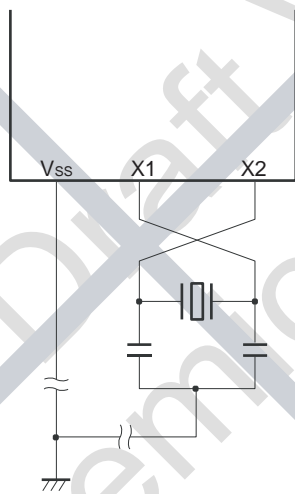
(a)连接电路的布线太长



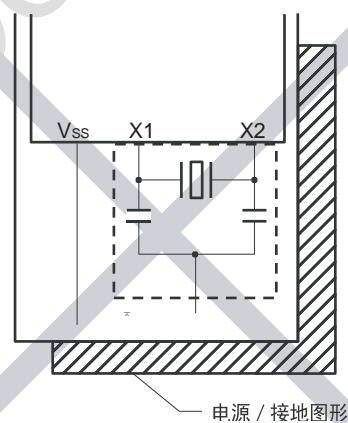
(b)信号线交叉



(c)X1和X2的信号线交叉布线



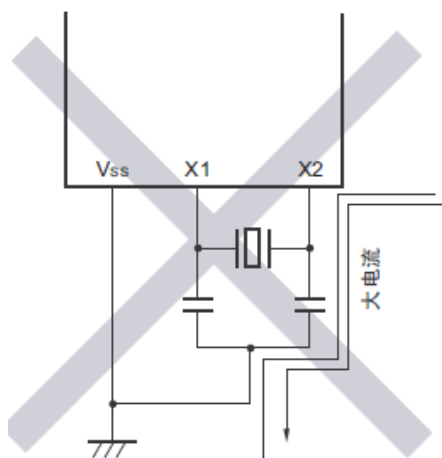
(d)X1和X2的布线下方有电源或者接地图形



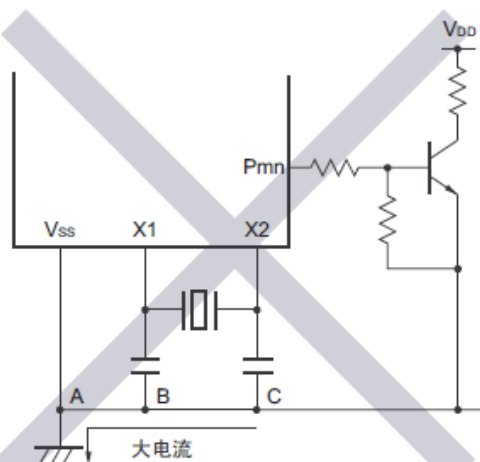
注：在多层板或者双面板中，不能在X1引脚、X2引脚和谐振器的布线区(图中虚线部分)下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

图4-14 不正确的谐振器连接例子(2/2)

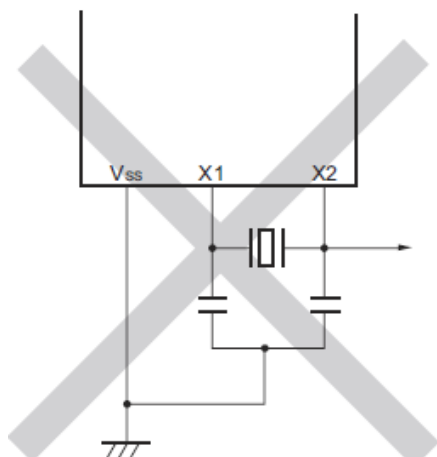
(e) 有变化的大电流接近信号线



(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X1并行布线时，X2的串扰噪声会叠加到X1而导致误动作。

## 4.4.2 高速内部振荡器

BAT32A233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。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。

## 4.4.3 低速内部振荡器

BAT32A233内置低速内部振荡器。

当选项字节(000C0H)的bit4(WDTON)，低速内部振荡器振荡。

当看门狗定时器停止运行并且SELLOSC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SELLOSC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<sub>MAIN</sub>

高速系统时钟：F<sub>MX</sub>

X1时钟：F<sub>X</sub>

外部主系统时钟：F<sub>EX</sub>

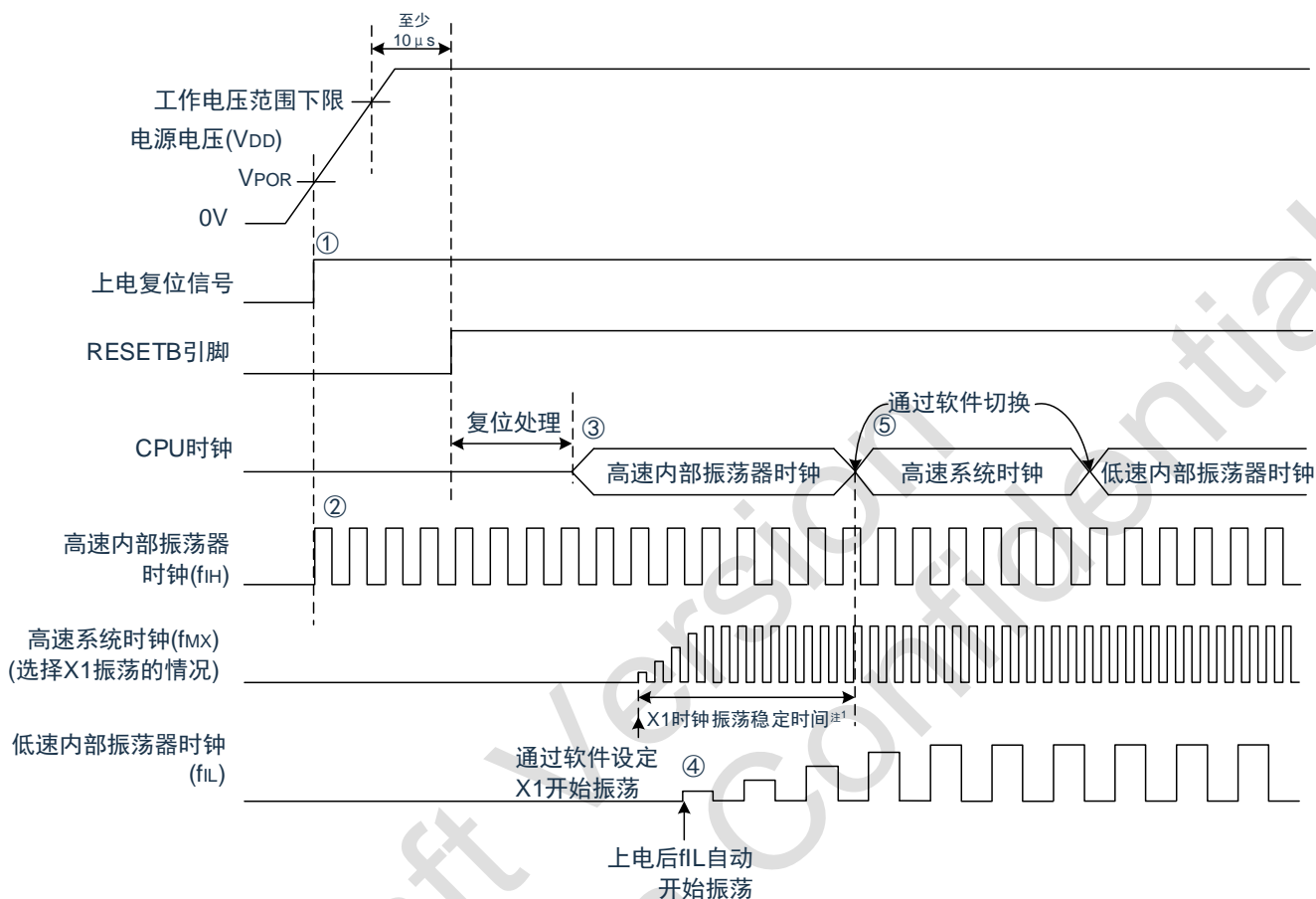
高速内部振荡器时钟：F<sub>IH</sub>

低速内部振荡器时钟：F<sub>IL</sub>

CPU/外围硬件时钟：F<sub>CLK</sub>

BAT32A233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时钟开始振荡(参照“4.6.2 X1振荡电路的设置例子”)。
- ⑤ 如果要将CPU时钟切换到X1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”)。

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	64MHz
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 <sub>IH</sub> =24MHz	F <sub>IH</sub> =32MHz	F <sub>IH</sub> =48MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =64MHz F <sub>HOCO</sub> =64MHz
0	0	1	F <sub>IH</sub> =12MHz	F <sub>IH</sub> =16MHz	F <sub>IH</sub> =24MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =32MHz F <sub>HOCO</sub> =64MHz
0	1	0	F <sub>IH</sub> =6MHz	F <sub>IH</sub> =8MHz	F <sub>IH</sub> =12MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =16MHz F <sub>HOCO</sub> =64MHz
0	1	1	F <sub>IH</sub> =3MHz	F <sub>IH</sub> =4MHz	F <sub>IH</sub> =6MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =8MHz F <sub>HOCO</sub> =64MHz
1	0	0	禁止设置。	F <sub>IH</sub> =2MHz	F <sub>IH</sub> =3MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =4MHz F <sub>HOCO</sub> =64MHz
1	0	1	禁止设置。	F <sub>IH</sub> =1MHz	禁止设置。	F <sub>IH</sub> =2MHz F <sub>HOCO</sub> =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OSCSEL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	EXCLK 0	OSCSEL 1	0	0	0	0	0	AMPH 0/1

- 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	OSTS2	OSTS1	OSTS0

- 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	MSTOP 0	0	0	0	0	0	0	HIOSTOP 0

- 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	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

- 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。

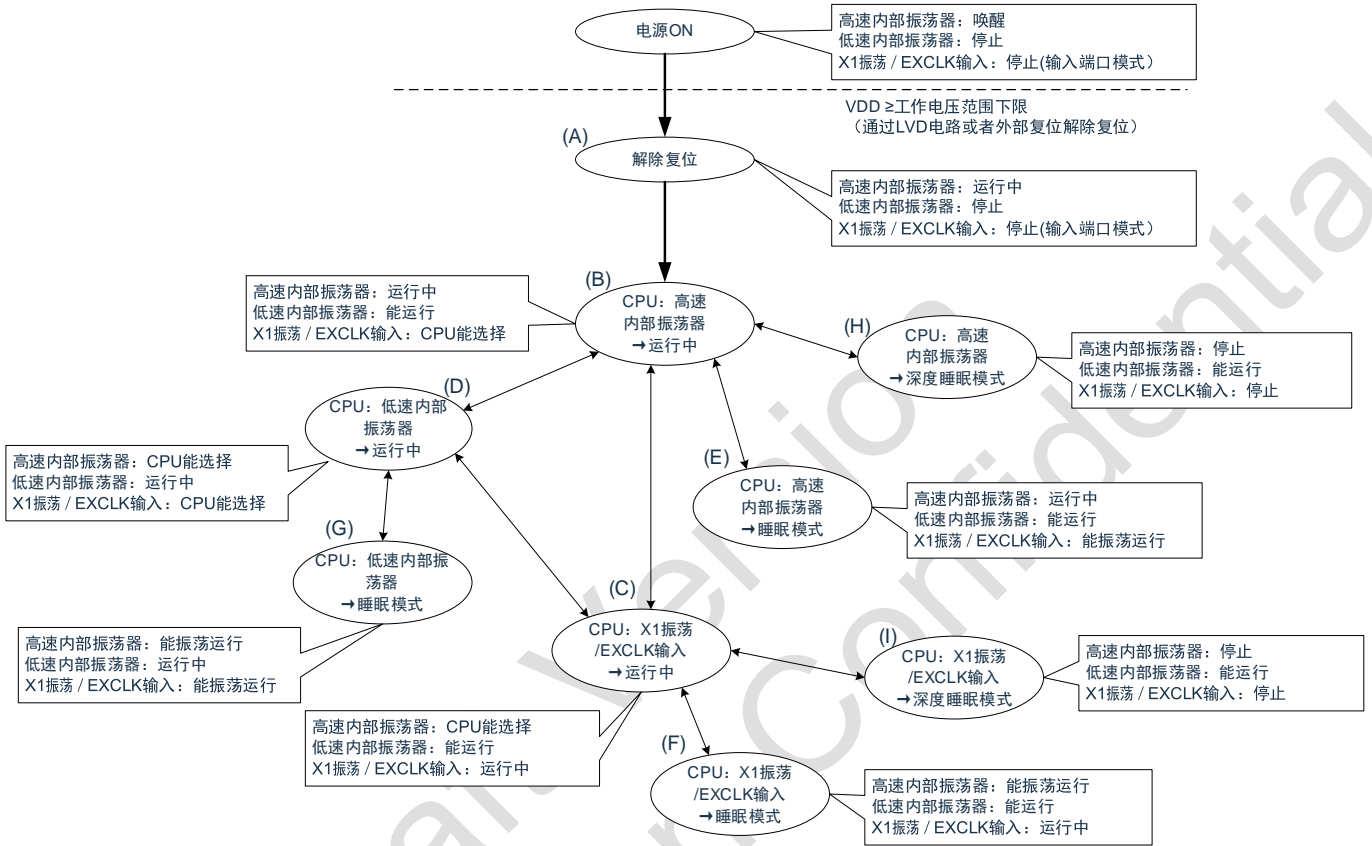
	7	6	5	4	3	2	1	0
CKC	0	0	MCS 0	MCM 0	0	0	0	0



### 4.6.3 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) →

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

(3) 在解除复位(A)后, CPU转移到内部低速系统时钟运行(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) →

SFR寄存器的设置标志 状态转移	SUBCKSEL寄存器	振荡稳定的 等待	SUBCKSE L寄存器
	SELLOSC		SCLS
(A)→(B)→(D) (内部低速振荡器时钟)	1	需要	1

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 <sup>注1</sup>			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	<sup>注2</sup>	0	需要确认	1
(B)→(C) (X1时钟: $10\text{MHz} < F_x \leq 20\text{MHz}$ )	0	1	1	<sup>注2</sup>	0	需要确认	1
(B)→(C) (外部主时钟)	1	1	×	<sup>注2</sup>	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  $\leq$  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。

(5) CPU从高速内部振荡器时钟运行(B)转移到内部低速系统时钟运行(D)。

(SFR寄存器的设置顺序)

SFR寄存器的设置标志 状态转移	SUBCKSEL寄存器	振荡稳定的等待	SUBCKSEL寄存器
	SELOSC		SCLS
(B)→(D) (内部低速振荡器时钟)	1	需要	1

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) →

SFR寄存器的设置标志 状态转移	CSC寄存器	振荡精度稳定的等待	CKC寄存器
	HIOSTOP		MCM0
(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)转移到内部低速系统时钟运行(D)。

(SFR寄存器的设置顺序) →

SFR寄存器的设置标志 状态转移	SUBCKSEL寄存器
	SELLSOC
(C)→(D)	1

(8) CPU从内部低速系统时钟运行(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) →

SFR寄存器的设置标志 状态转移	CSC寄存器	振荡精度稳定的等待	SUBCKSEL寄存器
	HIOSTOP		SELLOSC
(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)

SFR寄存器的设置标志 状态转移	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在低速内部系统时钟运行中(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)。

Draft Version  
 Cmsemicon Confidential

## 4.6.4 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

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	
	内部低速振荡器时钟	无特别条件。	
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。	—
	内部低速振荡器时钟	• 无特别条件。	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。	—
	内部低速振荡器时钟	• 无特别条件。	能将外部主系统时钟的输入置为无效(MSTOP=1)。
低速内部振荡器时钟	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	无需特别处理。
	X1时钟	X1振荡稳定。 OSCSEL=1, EXCLK=0, MSTOP=0 经过振荡稳定时间后	无需特别处理。
	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	无需特别处理。

### 4.6.5 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 <sub>IH</sub>	↔	F <sub>MX</sub>	参照表4-6。
F <sub>MAIN</sub>	↔	F <sub>IL</sub>	参照表4-7。

表4-6 F<sub>IH</sub>↔F<sub>MX</sub>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 <sub>MAIN</sub> =F <sub>IH</sub> )	1 (F <sub>MAIN</sub> =F <sub>MX</sub> )
0 (F <sub>MAIN</sub> =F <sub>IH</sub> )	F <sub>MX</sub> ≥ F <sub>IH</sub>		2个时钟
	F <sub>MX</sub> < F <sub>IH</sub>		2个F <sub>IH</sub> /F <sub>MX</sub> 时钟
1 (F <sub>MAIN</sub> =F <sub>MX</sub> )	F <sub>MX</sub> ≥ F <sub>IH</sub>	2个F <sub>MX</sub> /F <sub>IH</sub> 时钟	
	F <sub>MX</sub> < F <sub>IH</sub>	2个时钟	

表4-7 F<sub>MAIN</sub>↔F<sub>IL</sub>所需要的最大时钟数

切换前的设置值		切换后的设置值	
SELLOSC		SELLOSC	
		0 (F <sub>CLK</sub> =F <sub>MAIN</sub> )	1 (F <sub>CLK</sub> =F <sub>IL</sub> )
0 (F <sub>CLK</sub> =F <sub>MAIN</sub> )			1+2个F <sub>MAIN</sub> /F <sub>IL</sub> 时钟
1 (F <sub>CLK</sub> =F <sub>IL</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<sub>IH</sub>=8MHz、F<sub>MX</sub>=10MHz振荡的情况)

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



## 4.6.6 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 (CPU以高速系统时钟以外的时钟运行)	MSTOP=1
外部主系统时钟		
低速内部振荡器时钟	WDT不使能且不使用低速内部振荡器时钟做系统时钟时自动停止	-

## 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
- 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 = 4001\_A000H；

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

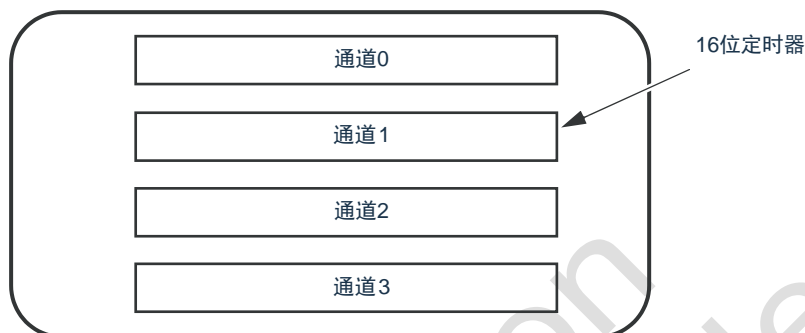
BUSY	用于指示除法运算的状态。
0	除法运算完成
1	除数运算进行中

## 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（即单元0），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"> <li>• 间隔定时器(→参照6.8.1)</li> <li>• 方波输出(→参照6.8.1)</li> <li>• 外部事件计数器(→参照6.8.2)</li> <li>• 分频器<sup>注</sup>(→参照6.8.3)</li> <li>• 输入脉冲间隔的测量(→参照6.8.4)</li> <li>• 输入信号的高低电平宽度的测量(→参照6.8.5)</li> <li>• 延迟计数器(→参照6.8.6)</li> </ul>	<ul style="list-style-type: none"> <li>• 单触发脉冲输出(→参照6.9.1)</li> <li>• PWM 输出(→参照6.9.2)</li> <li>• 多重PWM输出(→参照6.9.3)</li> </ul>

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

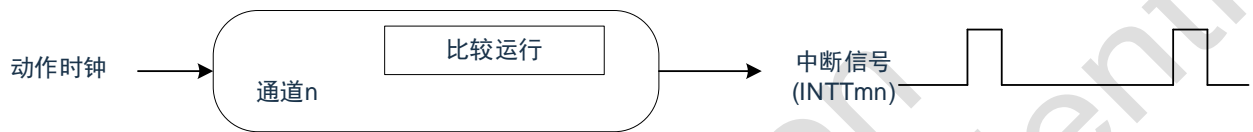
通用定时器单元有以下功能：

### 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。

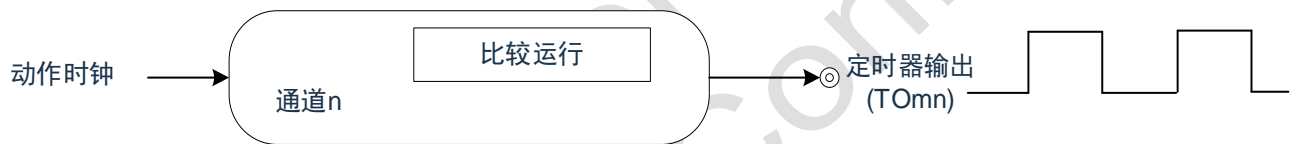
#### (1) 间隔定时器

能用作以固定间隔产生中断(INTTmn)的基准定时器。



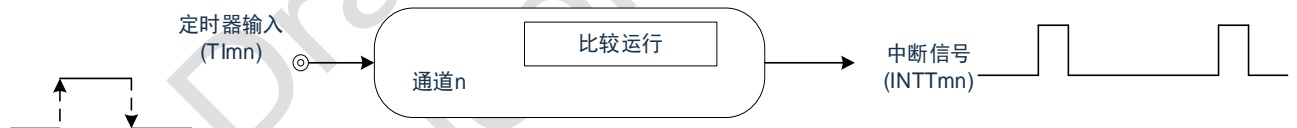
#### (2) 方波输出

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。



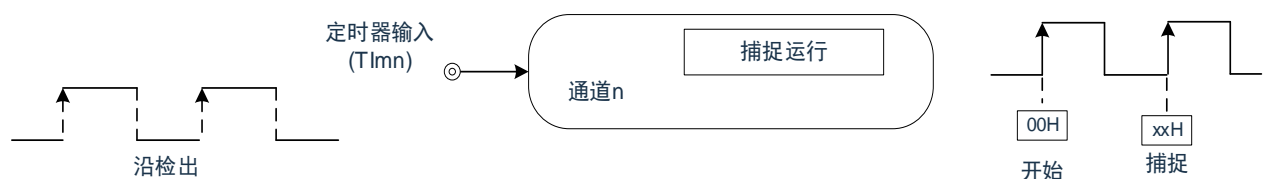
#### (4) 分频器功能(只限于单元0的通道0)

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。



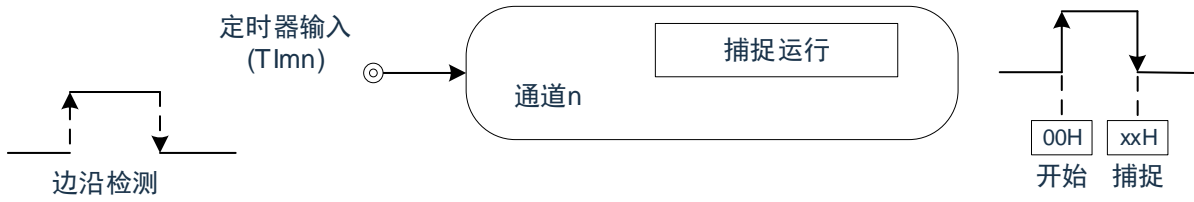
#### (5) 输入脉冲间隔的测量

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。

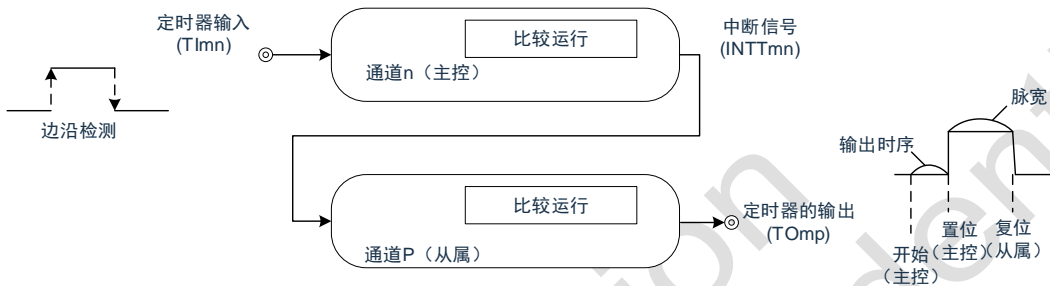
### 6.1.2 多通道联动运行功能

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。

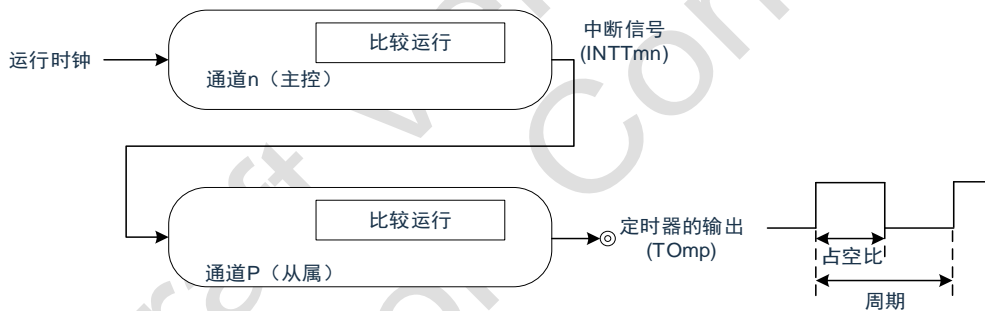
(1) 单触发脉冲输出

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。



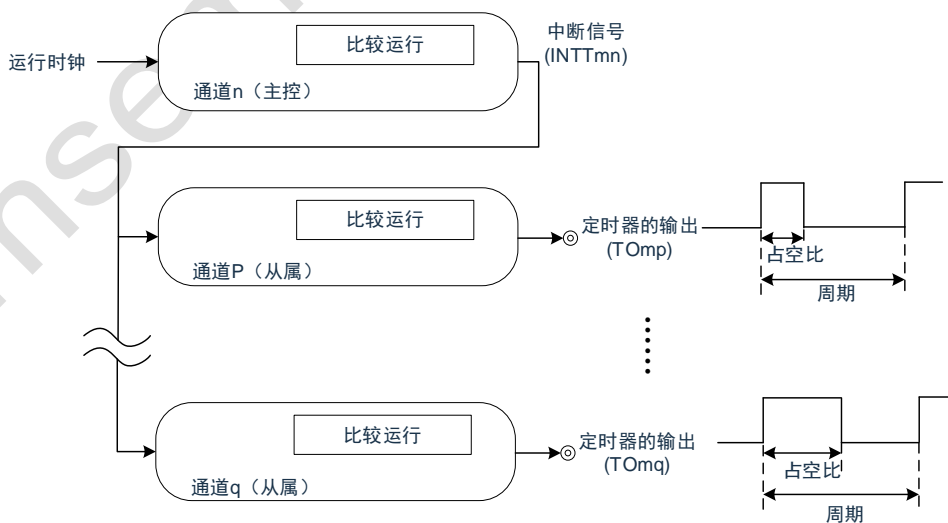
(2) PWM(Pulse Width Modulation)输出

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) 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6-1 通用定时器单元的结构

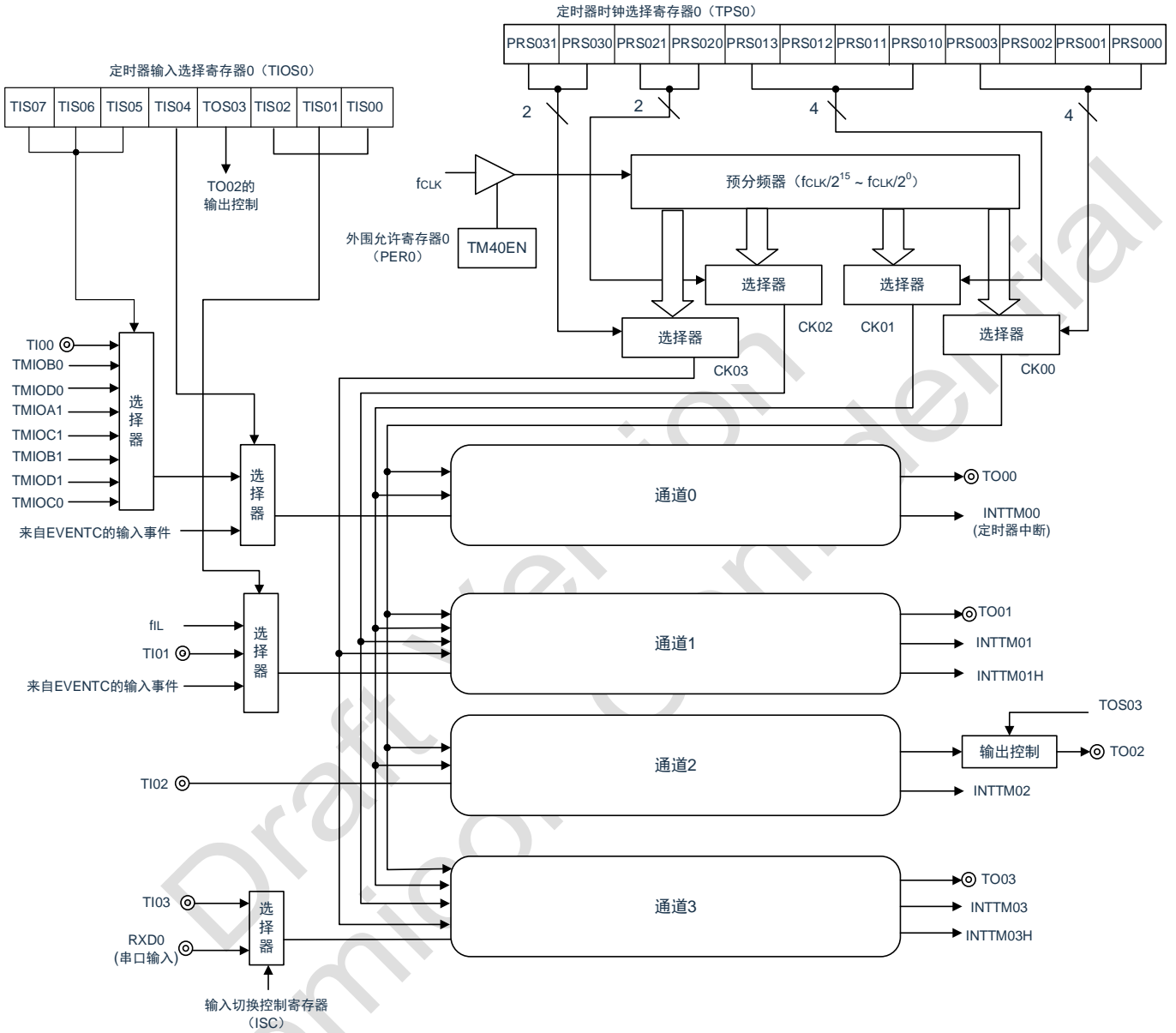
项目	结构
计数器	定时器计数寄存器mn(TCRmn)
寄存器	定时器数据寄存器mn(TDRmn)
定时器的输入	TI00~TI03、RxD0引脚(用于LIN-bus)
定时器的输出	TO00~TO03、输出控制电路
控制寄存器	<单元设置部的寄存器> <ul style="list-style-type: none"> <li>• 外围允许寄存器0(PER0)</li> <li>• 定时器时钟选择寄存器m(TPSm)</li> <li>• 定时器通道允许状态寄存器m(TEm)</li> <li>• 定时器通道开始寄存器m(TSm)</li> <li>• 定时器通道停止寄存器m(TTm)</li>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

注1：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定”。

备注：m：单元号(m=0) n：通道号(n=0~3)

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>IL</sub>：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
0x310	TDR00	R/W	16	0000H
0x312	TDR01	R/W	16	0000H
0x312	TDR01L	R/W	8	00H
0x313	TDR01H	R/W	8	00H
0x314	TDR02	R/W	16	0000H
0x316	TDR03	R/W	16	0000H
0x316	TDR03L	R/W	8	00H
0x317	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6-2 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 <sup>注</sup>			
		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)的表格

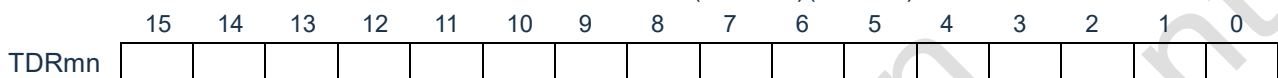
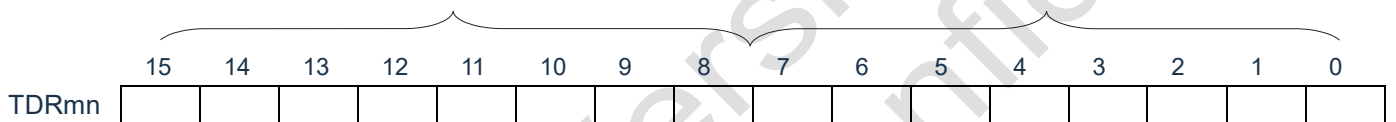


图6-4 定时器数据寄存器mn(TDRmn)(n=1、3)的表格  
(TDR0nH可支持8bit操作) (TDR0nL可支持8bit操作)



- (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

符号	7	6	5	4	3	2	1	0
PER0	0	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	0	TM40EN

TM40EN	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"> <li>不能写通用定时器单元0使用的SFR。</li> <li>通用定时器单元0处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写通用定时器单元0使用的SFR。</li> </ul>

**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 <sup>‡</sup> (k=0, 1)					
				F <sub>CLK</sub> =2MHz	F <sub>CLK</sub> =4MHz	F <sub>CLK</sub> =8MHz	F <sub>CLK</sub> =20MHz	F <sub>CLK</sub> =32MHz	
0	0	0	0	F <sub>CLK</sub>	2MHz	4MHz	8MHz	20MHz	32MHz
0	0	0	1	F <sub>CLK</sub> /2	1MHz	2MHz	4MHz	10MHz	16MHz
0	0	1	0	F <sub>CLK</sub> /2 <sup>2</sup>	500KHz	1MHz	2MHz	5MHz	8MHz
0	0	1	1	F <sub>CLK</sub> /2 <sup>3</sup>	250KHz	500KHz	1MHz	2.5MHz	4MHz
0	1	0	0	F <sub>CLK</sub> /2 <sup>4</sup>	125KHz	250KHz	500KHz	1.25MHz	2MHz
0	1	0	1	F <sub>CLK</sub> /2 <sup>5</sup>	62.5KHz	125KHz	250KHz	625KHz	1MHz
0	1	1	0	F <sub>CLK</sub> /2 <sup>6</sup>	31.3KHz	62.5KHz	125KHz	313KHz	500KHz
0	1	1	1	F <sub>CLK</sub> /2 <sup>7</sup>	15.6KHz	31.3KHz	62.5KHz	156KHz	250KHz
1	0	0	0	F <sub>CLK</sub> /2 <sup>8</sup>	7.81KHz	15.6KHz	31.3KHz	78.1KHz	125KHz
1	0	0	1	F <sub>CLK</sub> /2 <sup>9</sup>	3.91KHz	7.81KHz	15.6KHz	39.1KHz	62.5KHz
1	0	1	0	F <sub>CLK</sub> /2 <sup>10</sup>	1.95KHz	3.91KHz	7.81KHz	19.5KHz	31.25KHz
1	0	1	1	F <sub>CLK</sub> /2 <sup>11</sup>	977Hz	1.95KHz	3.91KHz	9.77KHz	15.6KHz
1	1	0	0	F <sub>CLK</sub> /2 <sup>12</sup>	488Hz	977Hz	1.95KHz	4.88KHz	7.81KHz
1	1	0	1	F <sub>CLK</sub> /2 <sup>13</sup>	244Hz	488Hz	977Hz	2.44KHz	3.91KHz
1	1	1	0	F <sub>CLK</sub> /2 <sup>14</sup>	122Hz	244Hz	488Hz	1.22KHz	1.95KHz
1	1	1	1	F <sub>CLK</sub> /2 <sup>15</sup>	61.0Hz	122Hz	244Hz	610Hz	977Hz

注：在更改选择为F<sub>CLK</sub>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<sub>MCK</sub>)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<sub>CLK</sub>(无分频)作为运行时钟(CKmk)并且将TDRmn置“0000H”(m=0, n=0~3)，就不能使用通用定时器单元的中断请求。

备注：

1. F<sub>CLK</sub>：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<sub>CLK</sub>周期为高电平(m=0)。详细内容请参照“6.5.1计数时钟(F<sub>TCLK</sub>)”。

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 <sup>注</sup>					
			F <sub>CLK</sub> =2MHz	F <sub>CLK</sub> =4MHz	F <sub>CLK</sub> =8MHz	F <sub>CLK</sub> =20MHz	F <sub>CLK</sub> =32MHz
0	0	F <sub>CLK</sub> /2	1MHz	2MHz	4MHz	10MHz	16MHz
0	1	F <sub>CLK</sub> /2 <sup>2</sup>	500KHz	1MHz	2MHz	5MHz	8MHz
1	0	F <sub>CLK</sub> /2 <sup>4</sup>	125KHz	250KHz	500KHz	1.25MHz	2MHz
1	1	F <sub>CLK</sub> /2 <sup>6</sup>	31.3KHz	62.5KHz	125KHz	313KHz	500KHz

PRSm31	PRSm30	运行时钟(CKm3)的选择 <sup>注</sup>					
			F <sub>CLK</sub> =2MHz	F <sub>CLK</sub> =4MHz	F <sub>CLK</sub> =8MHz	F <sub>CLK</sub> =20MHz	F <sub>CLK</sub> =32MHz
0	0	F <sub>CLK</sub> /2 <sup>8</sup>	7.81KHz	15.6KHz	31.3KHz	78.1KHz	125KHz
0	1	F <sub>CLK</sub> /2 <sup>10</sup>	1.95KHz	3.91KHz	7.81KHz	19.5KHz	31.3KHz
1	0	F <sub>CLK</sub> /2 <sup>12</sup>	488Hz	977Hz	1.95KHz	4.88KHz	7.81KHz
1	1	F <sub>CLK</sub> /2 <sup>14</sup>	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6-3所示的间隔时间。

表6-3 运行时钟CKSm2和CKSm3能设置的间隔时间

时钟		间隔时间 <sup>注</sup> (F <sub>CLK</sub> =32MHz)			
		10us	100us	1ms	10ms
CKm2	F <sub>CLK</sub> /2	○	—	—	—
	F <sub>CLK</sub> /2 <sup>2</sup>	○	—	—	—
	F <sub>CLK</sub> /2 <sup>4</sup>	○	○	—	—
	F <sub>CLK</sub> /2 <sup>6</sup>	○	○	—	—
CKm3	F <sub>CLK</sub> /2 <sup>8</sup>	—	○	○	—
	F <sub>CLK</sub> /2 <sup>10</sup>	—	○	○	—
	F <sub>CLK</sub> /2 <sup>12</sup>	—	—	○	○
	F <sub>CLK</sub> /2 <sup>14</sup>	—	—	○	○

注：○ 包含5%以内的误差。

备注：

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

### 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<sub>MCK</sub>)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<sub>mn</sub>=1)改写TMRmn寄存器。但是，能在一部分的功能运行中(TE<sub>mn</sub>=1)改写bit7和bit6(CIS<sub>mn1</sub>、CIS<sub>mn0</sub>)(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 <sup>注1</sup>	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS <sub>mn1</sub>	CKS <sub>mn0</sub>	通道n运行时钟(F <sub>MCK</sub> )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<sub>MCK</sub>)用于边沿检测电路。通过设置CCS<sub>mn</sub>位来产生采样时钟和计数时钟(F<sub>TCLK</sub>)。只有通道1和通道3才能选择运行时钟CKm2和CKm3。

CCS <sub>mn</sub>	通道n计数时钟(F <sub>TCLK</sub> )的选择
0	CKS <sub>mn0</sub> 位和CKS <sub>mn1</sub> 位指定的运行时钟(F <sub>MCK</sub> )
1	TI <sub>mn</sub> 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<sub>TCLK</sub>)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( $TTm=0A0FH$ )。

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 <sup>注1</sup>	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 <sup>注1</sup>	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 <sup>注1</sup>	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
• 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)。
• 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)。
• 单次计数模式 <sup>注2</sup> (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 <sup>注3</sup> 。此时不产生中断。
• 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6-4。

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) n：通道号(n=0~3)

表6-4 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	清除	— (不能使用)
•事件计数器模式	置位	
•单次计数模式		

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 <sub>m3</sub>	0	TEH <sub>m1</sub>	0	0	0	0	0	TEM <sub>3</sub>	TEM <sub>2</sub>	TEM <sub>1</sub>	TEM <sub>0</sub>

备注：m=0

TEH <sub>m3</sub>	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 <sub>m1</sub>	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

TE <sub>m</sub> <sub>n</sub>	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，TE<sub>m</sub><sub>1</sub>和TE<sub>m</sub><sub>3</sub>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6-5)。

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6-5)。

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6-5)。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	TTHm3	0	TTHm1	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)

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(FIL)
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 $1/F_{MCK}+10ns$ 。
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FCLK。

### 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6-17 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	TOEm3	TOEm2	TOEm1	TOEm0

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(TOmn)输出。

只有在禁止定时器输出 (TOEmn=0)时才能通过软件改写此寄存器的TOmn位。当允许定时器输出时 (TOEmn=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TO00、TO01、TO02、TO03 引脚用作端口功能时，必须将相应的TOmn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6-18：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

TOmn	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6-19 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6-20 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 <sub>m3</sub>	TOM <sub>m2</sub>	TOM <sub>m1</sub>	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。

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6-21 输入切换控制寄存器(ISC)的表格

地址: 40040473H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	ISC2	ISC1	ISC0
ISC2	外部中断 (INTP11) 的输入切换							
0	将INTP11引脚的输入信号用作外部中断的输入。							
1	将LRxD0引脚的输入信号用作外部中断的输入。							
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bit7~3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<sup>注</sup>。

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Tl<sub>mn</sub>引脚输入信号的有效边沿的情况(CCS<sub>mn</sub>=1)”、“6.5.2计数器的开始时序”和“6.7定时器输入(Tl<sub>mn</sub>)的控制”。

图6-22 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 <sup>注</sup>							
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。

### 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。

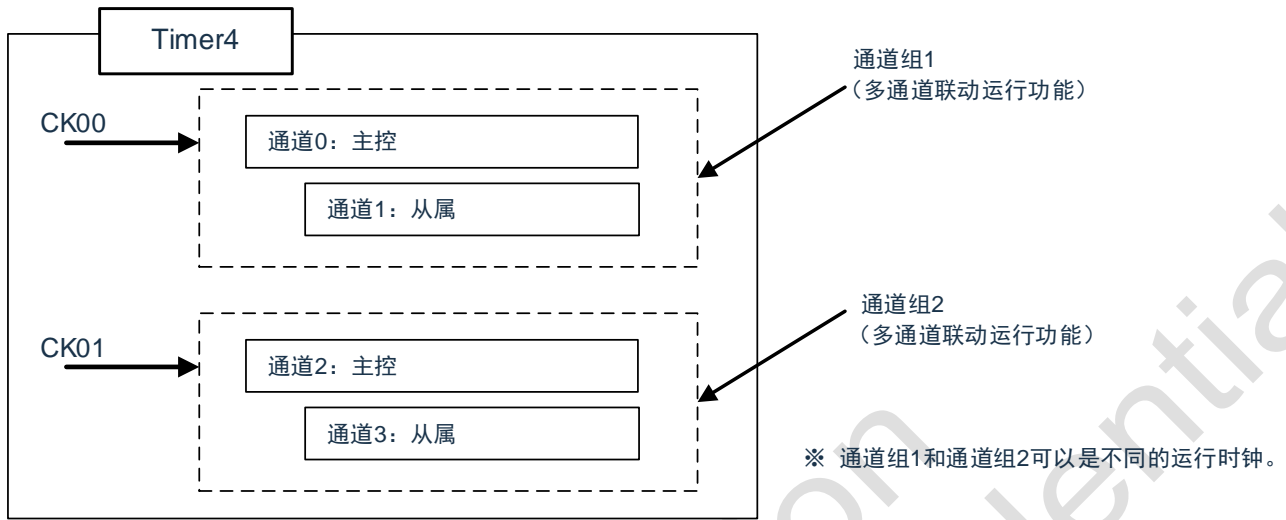
- 6) 和主控通道联动的从属通道需要设置相同的运行时钟。和主控通道联动的从属通道的CKSmn0位和CKSmn1位(定时器模式寄存器mn(TMRmn)的bit15和bit14)的值需要是相同的设置值。
- 7) 主控通道能将INTTMmn(中断)、开始软件触发和计数时钟传给低位通道。
- 8) 从属通道能将主控通道的INTTMmn(中断)、开始软件触发和计数时钟用作源时钟,但是不能将自己的INTTMmn(中断)、开始软件触发和计数时钟传给低位通道。
- 9) 主控通道不能将其他高位主控通道的INTTMmn(中断)、开始软件触发和计数时钟用作源时钟。
- 10) 为了同时启动要联动的通道,需要同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。

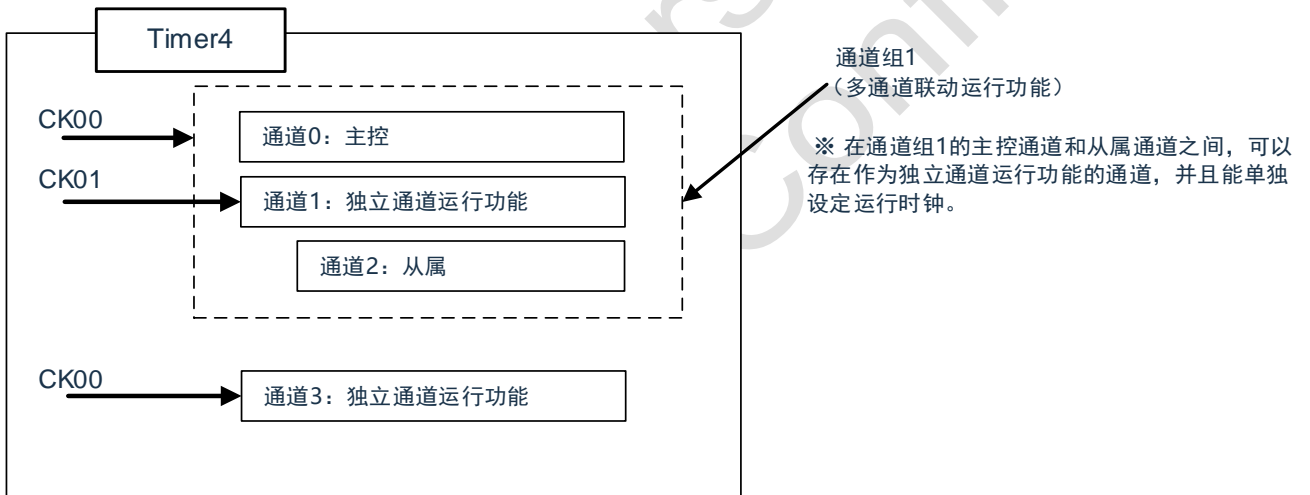
如果设置2个或者更多的相互不联动的通道群,通道群之间就不适用上述的基本规则。

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

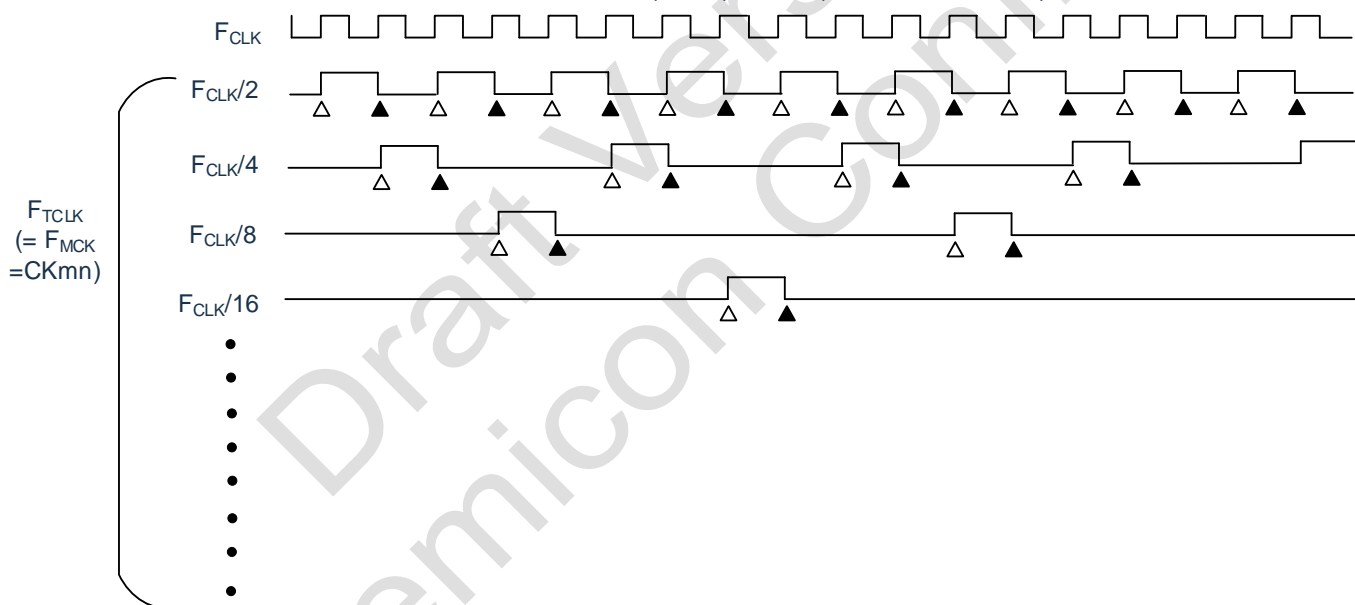
通用定时器单元被设计为与 $F_{CLK}$ 同步运行,因此计数时钟( $F_{TCLK}$ )的时序如下。

(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6-23  $F_{CLK}$ 和计数时钟( $F_{TCLK}$ )的时序(CCSmn=0的情况)



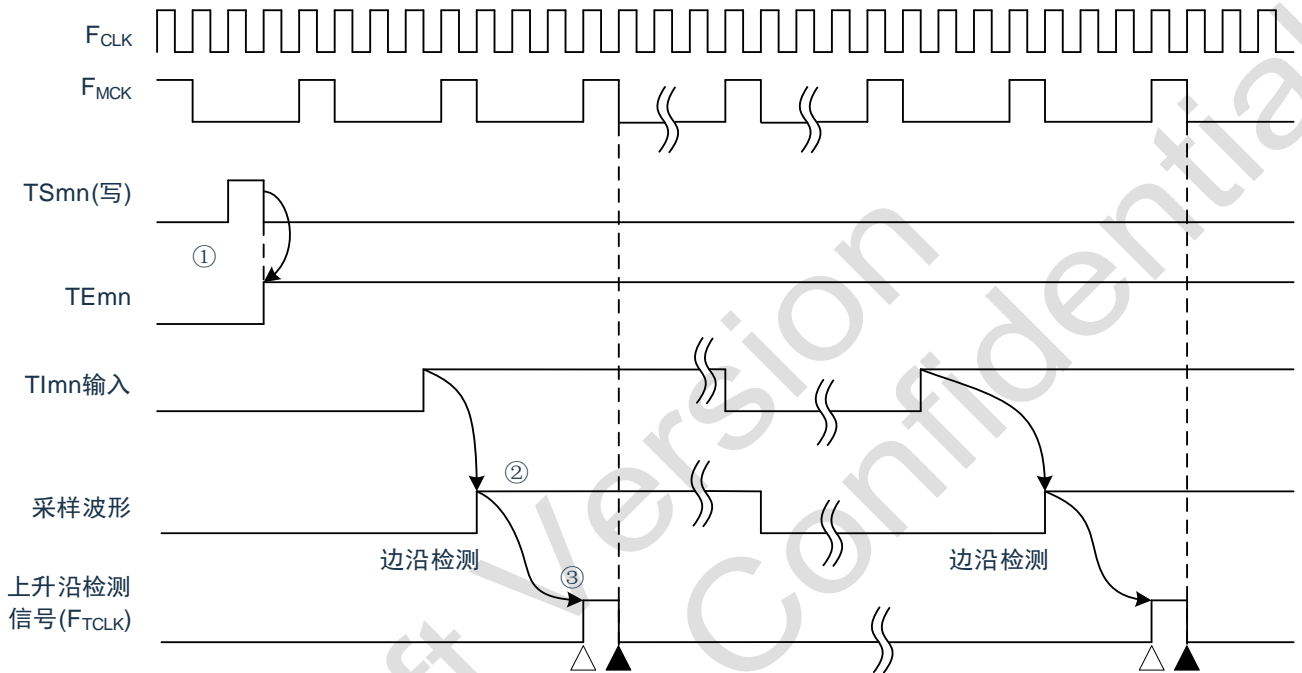
备注:

1.  $\Delta$ : 计数时钟的上升沿  
▲: 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 $F_{CLK}$ 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6-24 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6-5所示。

表6-5 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)”)。

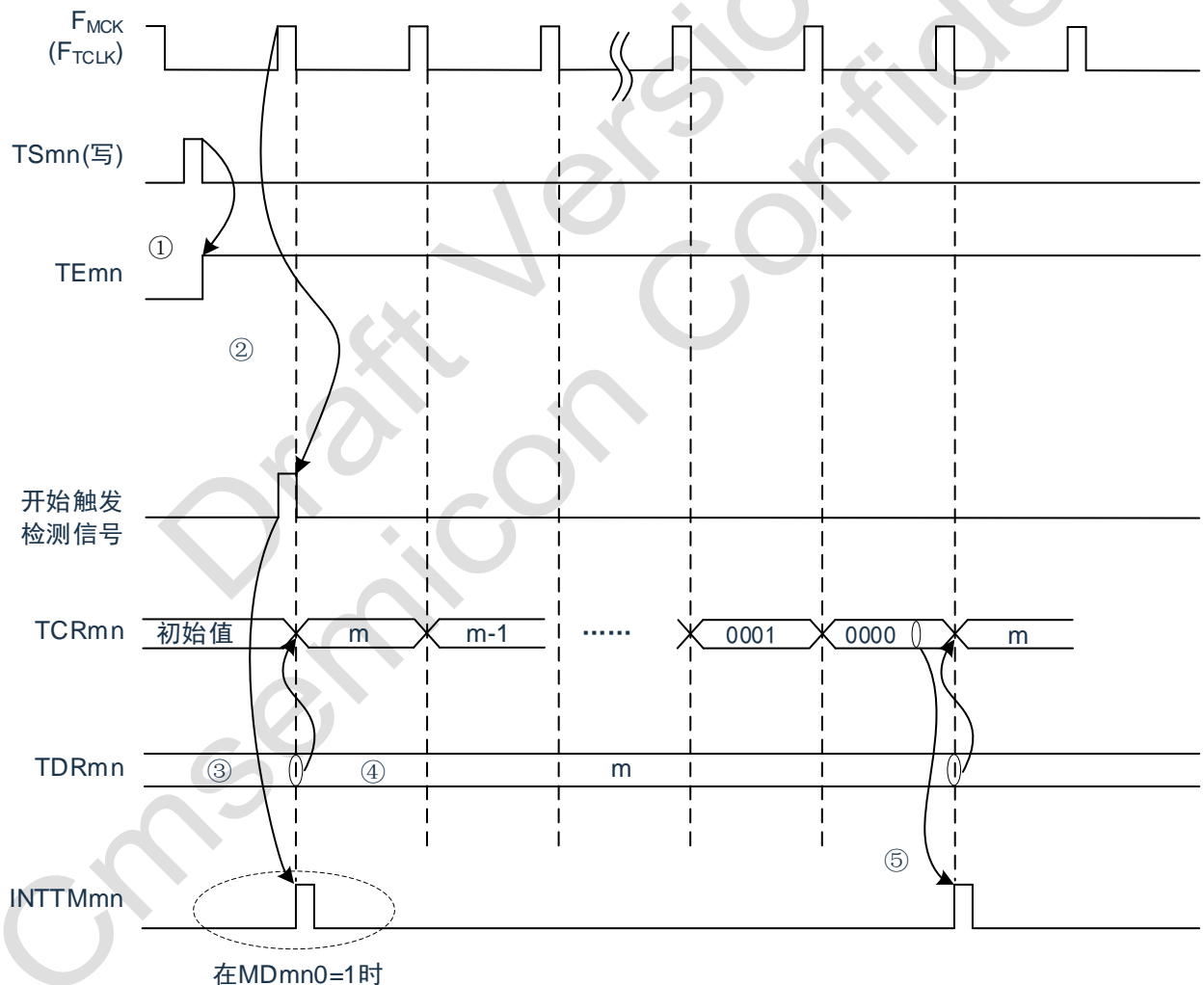
### 6.5.3 计数器的运行

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<sub>MCK</sub>)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<sub>MCK</sub>)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6-25 运行时序 (间隔定时器模式)



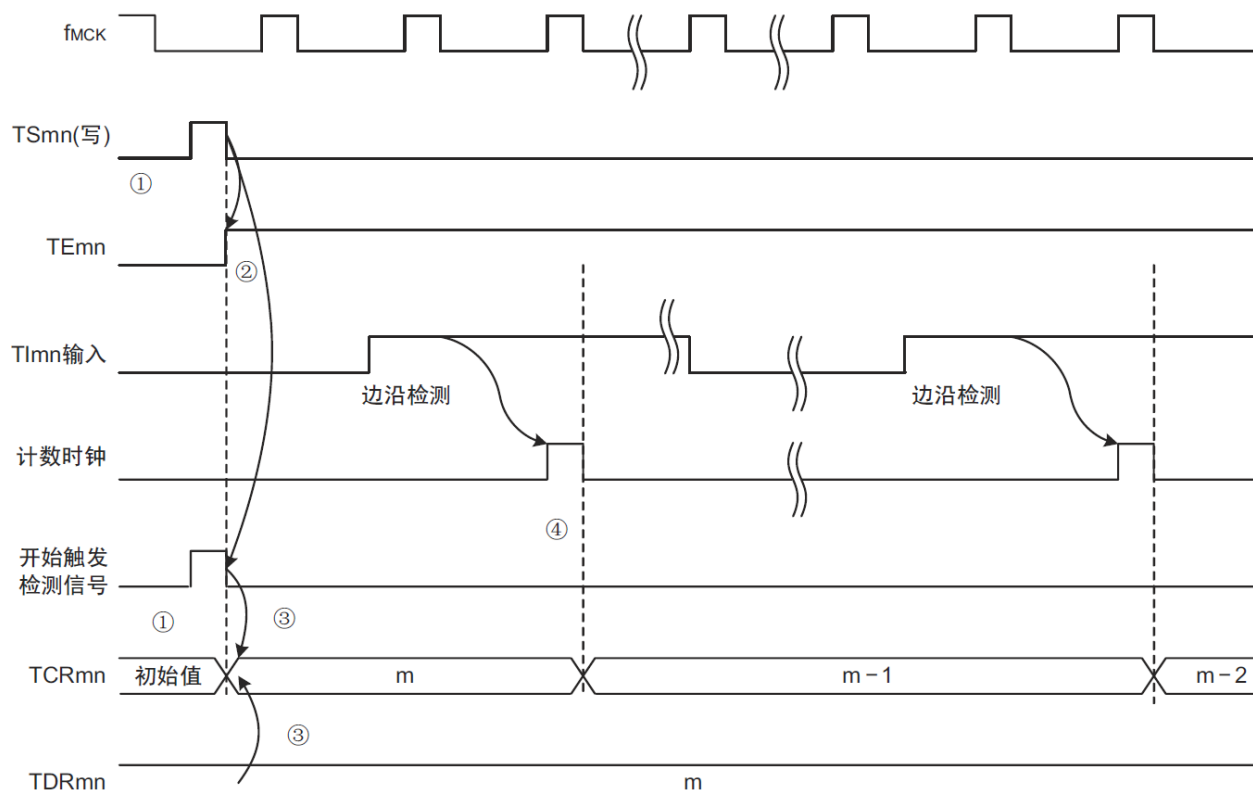
注意：因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

备注：F<sub>MCK</sub>、开始触发检测信号和INTTMmn与F<sub>CLK</sub>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<sub>mn</sub>=0)的期间, 定时器计数寄存器mn(TCR<sub>mn</sub>)保持初始值。
- ② 通过给TS<sub>mn</sub>位写“1”, 进入运行允许状态(TE<sub>mn</sub>=1)。
- ③ 在TS<sub>mn</sub>位和TE<sub>mn</sub>位都变为“1”的同时将定时器数据寄存器mn(TDR<sub>mn</sub>)的值装入TCR<sub>mn</sub>寄存器, 并且开始计数。
- ④ 此后, 在TI<sub>mn</sub>输入的有效边沿, 通过计数时钟对TCR<sub>mn</sub>寄存器的值进行递减计数。

图6-26 运行时序(事件计数器模式)

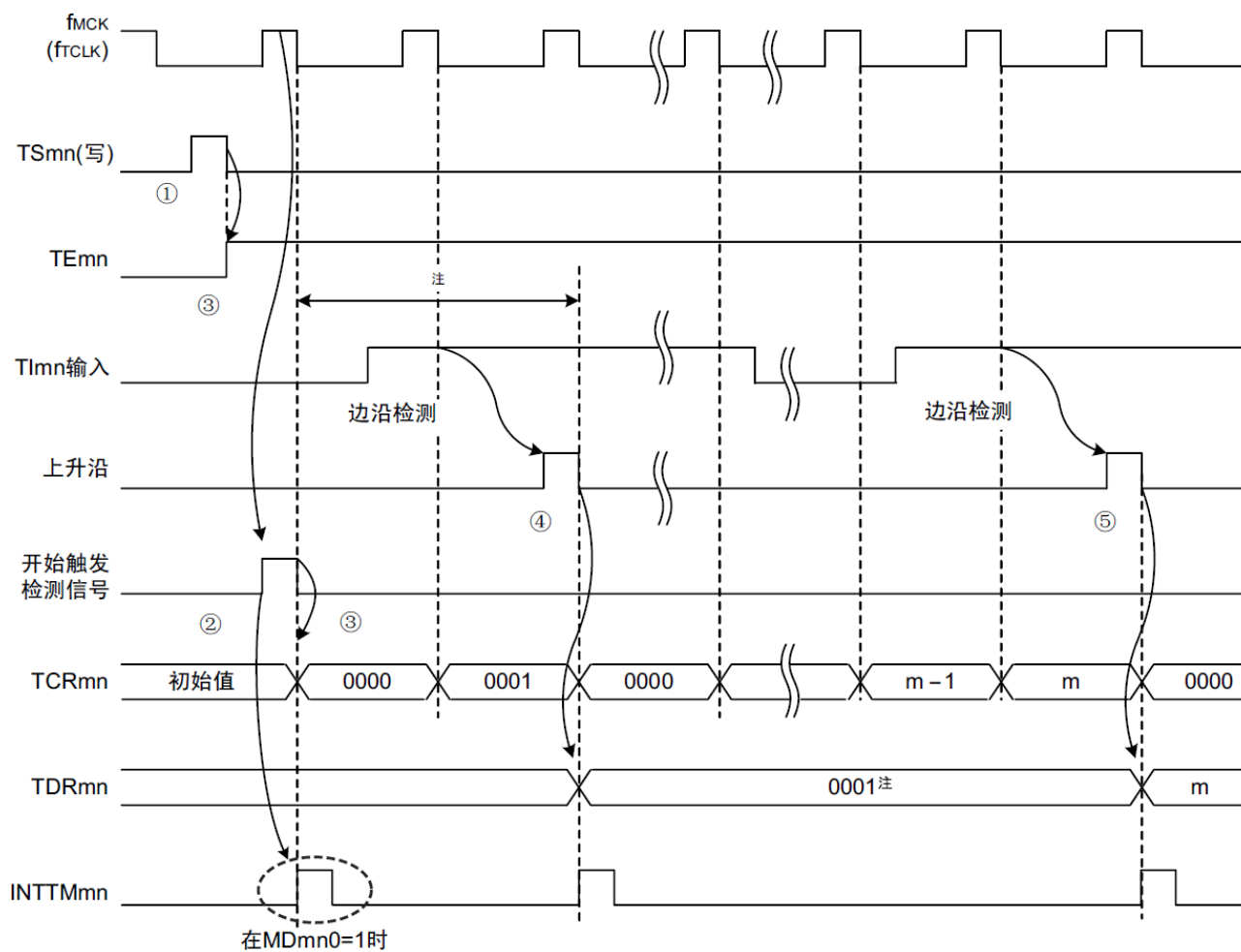


备注: 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从TI<sub>mn</sub>输入开始再延迟2个F<sub>MCK</sub>周期(合计3~4个周期)。1个周期的误差是因为TI<sub>mn</sub>输入与计数时钟(F<sub>MCK</sub>)不同步。

(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<sub>MCK</sub>)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6-27 运行时序(捕捉模式：输入脉冲的间隔测量)



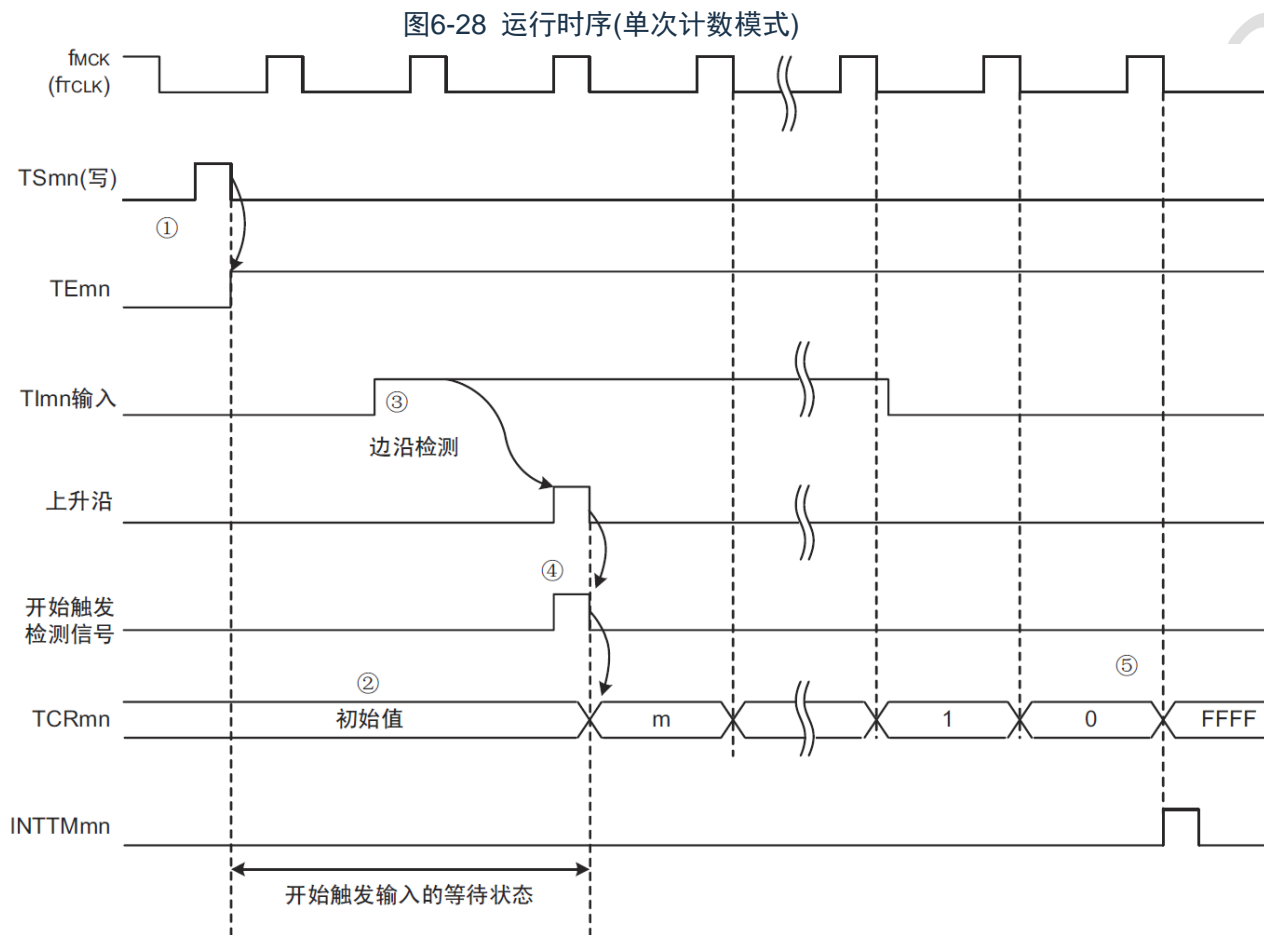
注：在开始前将时钟输入到TImn(有触发)时，即使没有检测到边沿也通过检测触发来开始计数，因此第1次捕捉时(④)的捕捉值不是脉冲间隔(在此例子中，0001：2个时钟间隔)，必须忽视。

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<sub>MCK</sub>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<sub>MCK</sub>)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“FFFF”，停止计数。

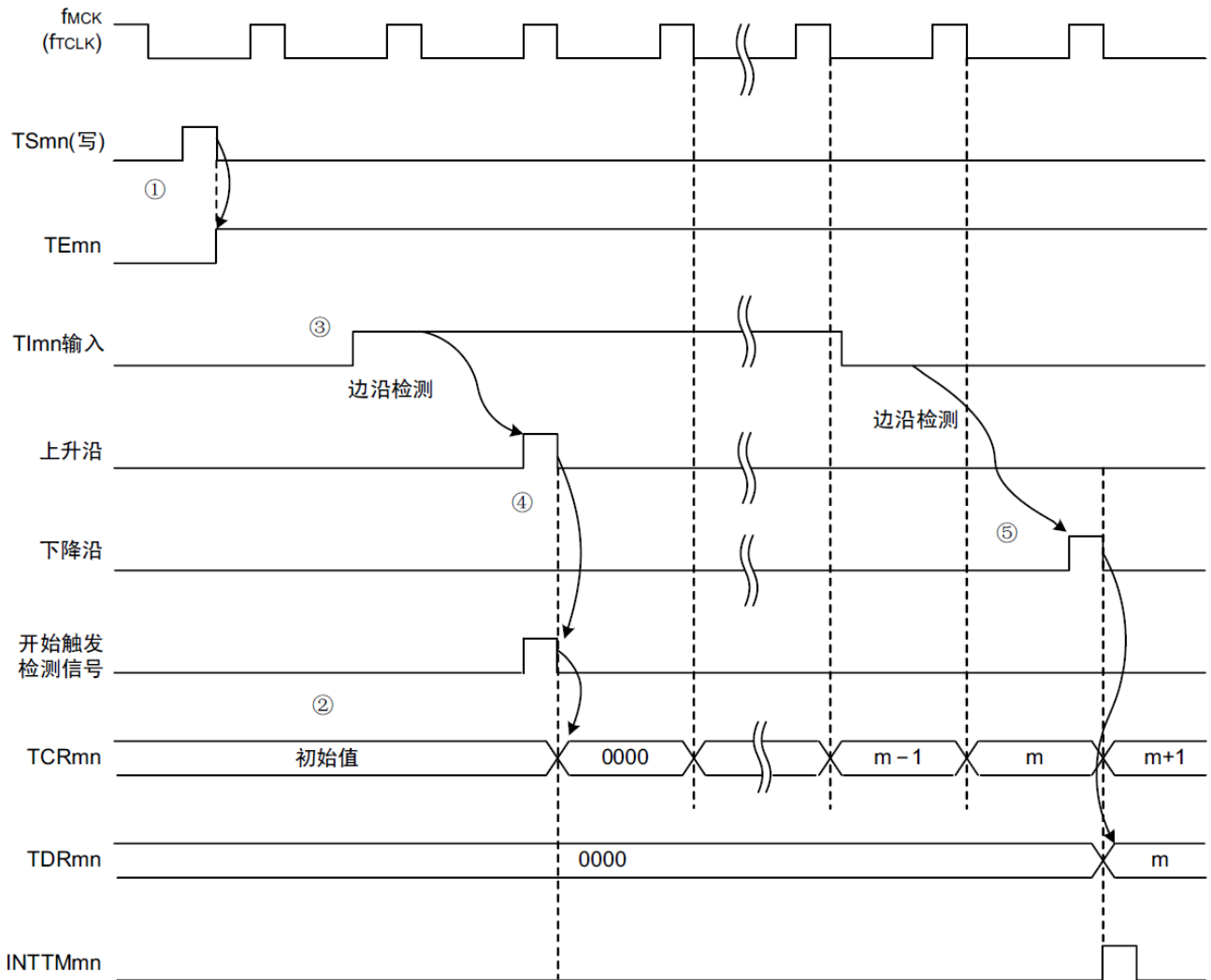


备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TIMn输入开始再延迟2个 $F_{MCK}$ 周期(合计3~4个周期)。1个周期的误差是因为TIMn输入与计数时钟( $F_{MCK}$ )不同步。

(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(TSmn)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6-29 运行时序(捕捉&单次计数模式：高电平宽度的测量)

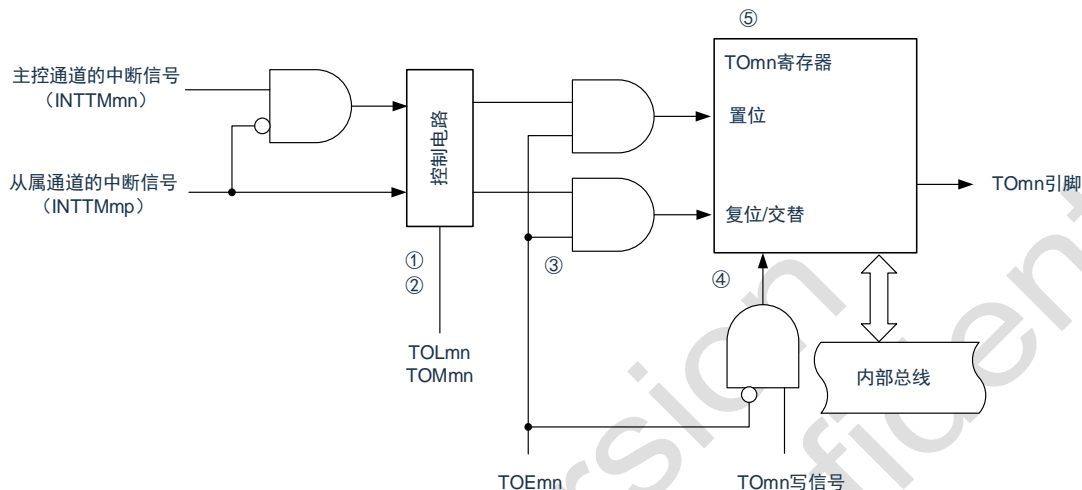


备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TImn输入开始再延迟2个F<sub>MCK</sub>周期(合计3~4个周期)。1个周期的误差是因为TImn输入和计数时钟(F<sub>MCK</sub>)不同步。

## 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6-30 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

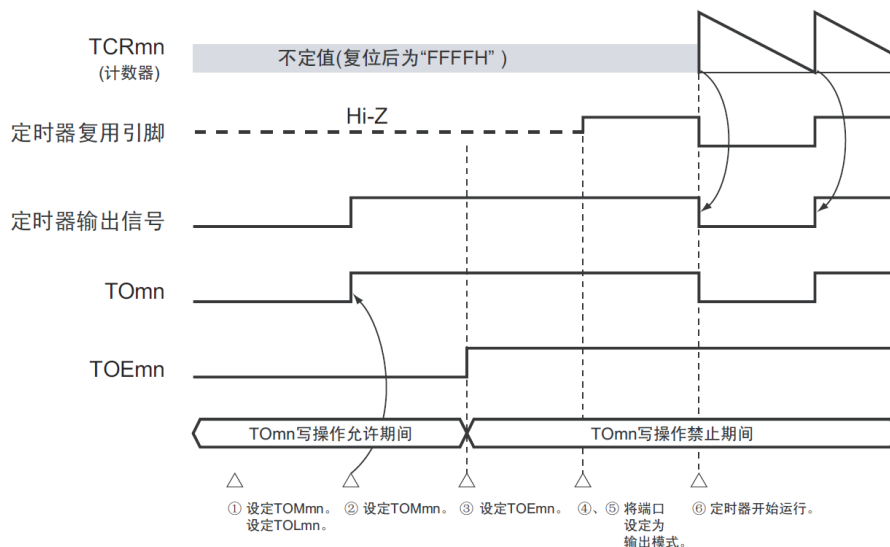
n=0：p=1、2、3

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6-31 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<sub>m</sub>、TOE<sub>m</sub>、TOL<sub>m</sub>、TOM<sub>m</sub>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<sub>mn</sub>)和定时器数据寄存器mn(TDR<sub>mn</sub>)的运行)和TOM<sub>n</sub>输出电路相互独立。因此,定时器输出寄存器m(TO<sub>m</sub>)、定时器输出允许寄存器m(TE<sub>m</sub>)和定时器输出电平寄存器m(TOL<sub>m</sub>)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<sub>n</sub>引脚输出期待的波形,必须参照6.8和6.9所示的寄存器设置的例子。

如果在产生各通道的定时器中断(INTT<sub>mn</sub>)信号前后更改除了TO<sub>m</sub>寄存器以外的TE<sub>m</sub>寄存器和TOL<sub>m</sub>寄存器的设置值,就根据是在产生定时器中断(INTT<sub>mn</sub>)信号前更改还是在产生后更改,TOM<sub>n</sub>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)

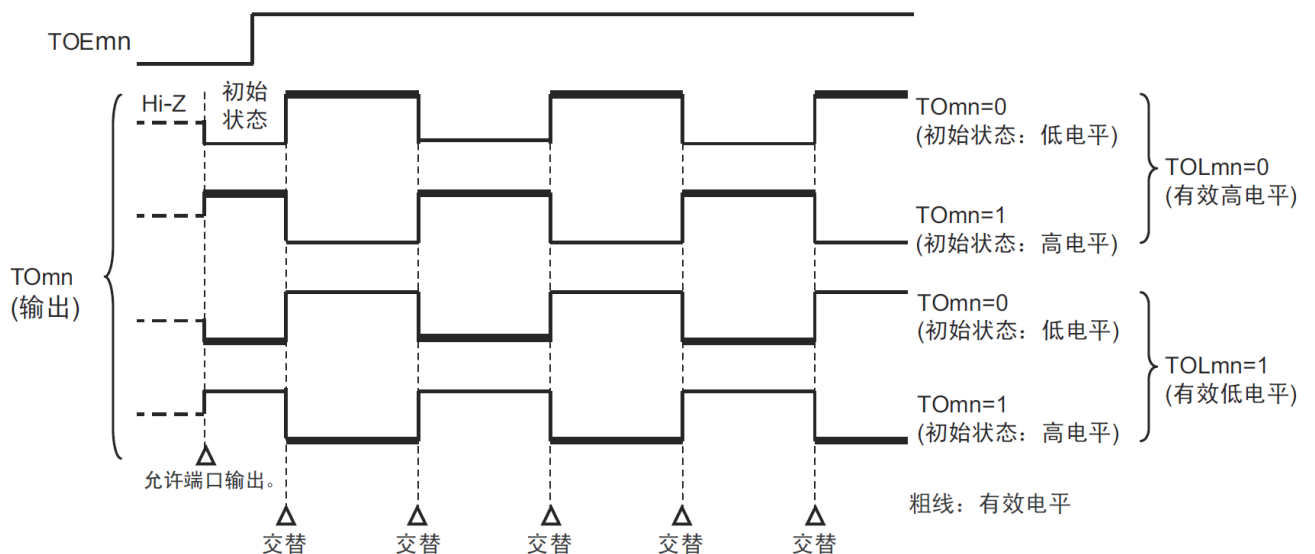
2) 有关TOM<sub>n</sub>引脚的初始电平和定时器开始运行后的输出电平

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

(a) 在主导通道输出模式(TOM<sub>mn</sub>=0)中开始运行的情况

在主导通道输出模式(TOM<sub>mn</sub>=0)中,定时器输出电平寄存器m(TOL<sub>m</sub>)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<sub>n</sub>引脚的输出电平。

图6-32 交替输出时(TOM<sub>mn</sub>=0)的TOM<sub>n</sub>引脚输出状态



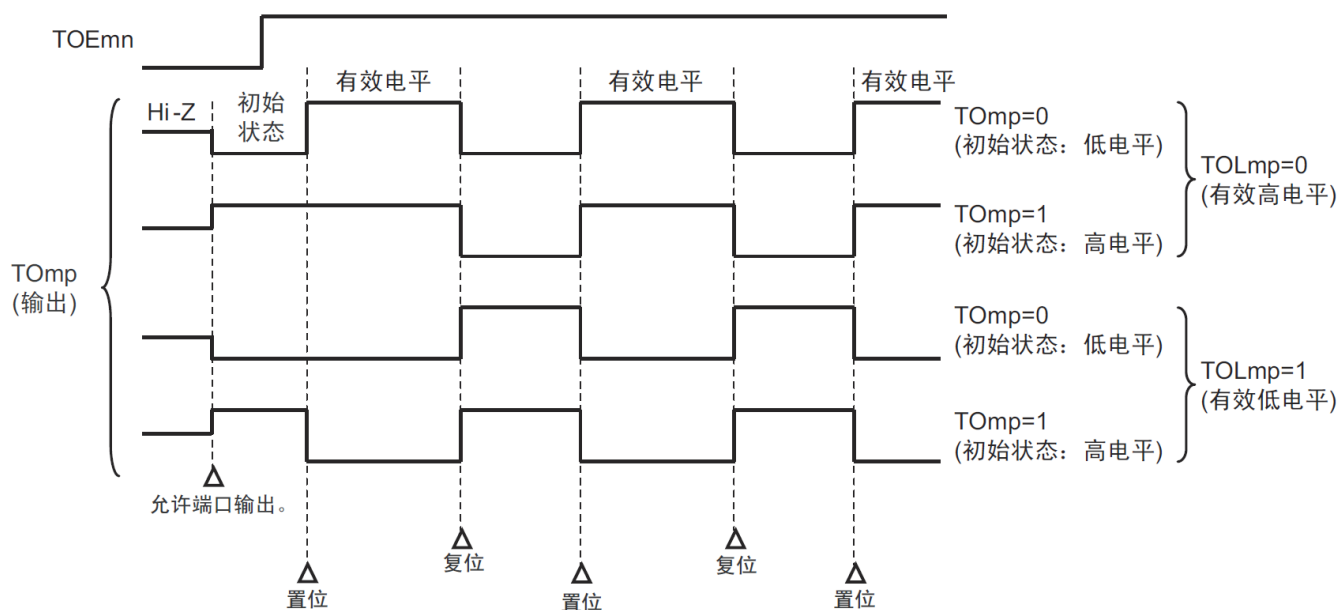
备注:

1. 交替:反相TOM<sub>n</sub>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6-33 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)

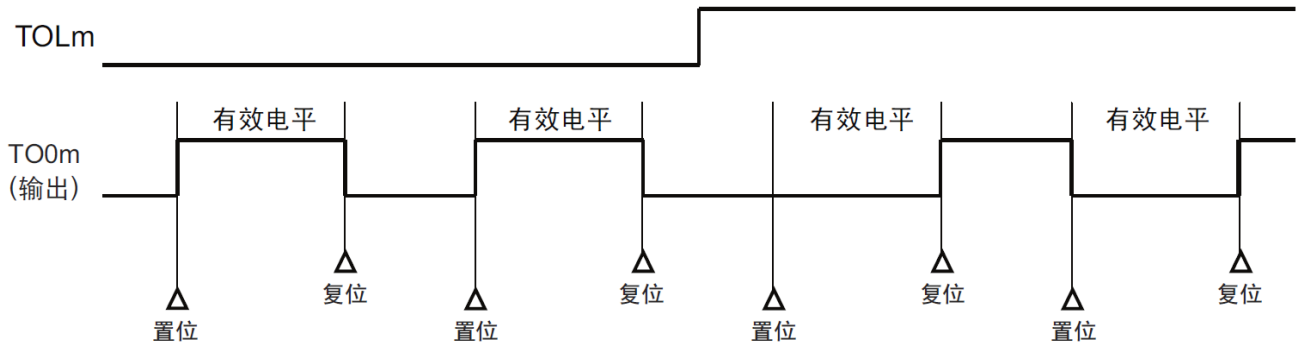
3) 有关从属通道输出模式(TOMmn=1)的TOMn引脚变化

(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图所示。

图6-34 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。

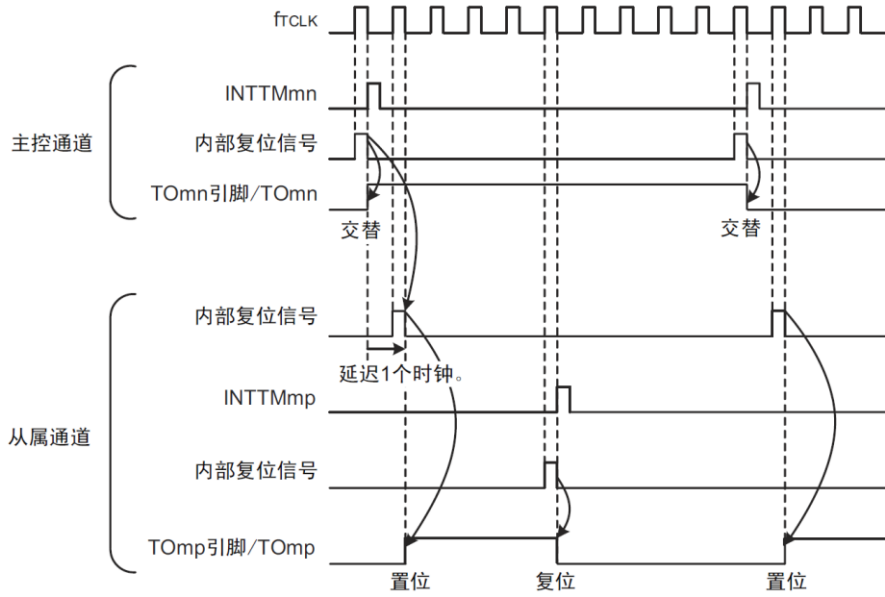
按照以下方法设置主控/从属通道时的置位/复位运行状态如图6-35所示。

主控通道: TOEmn=1、TOMmn=0、TOLmn=0

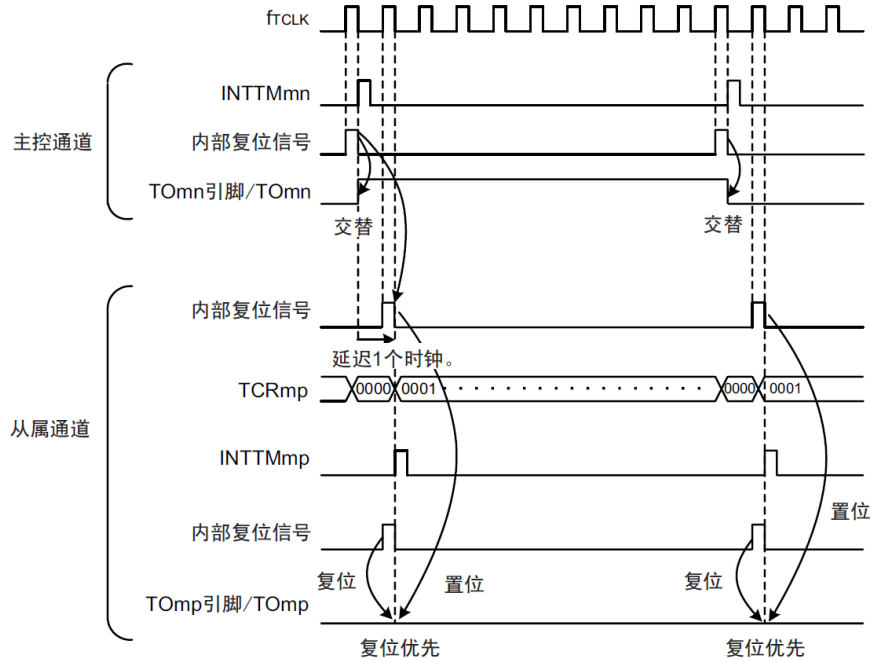
从属通道: TOEmp=1、TOMmp=1、TOLmp=0

图6-35 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



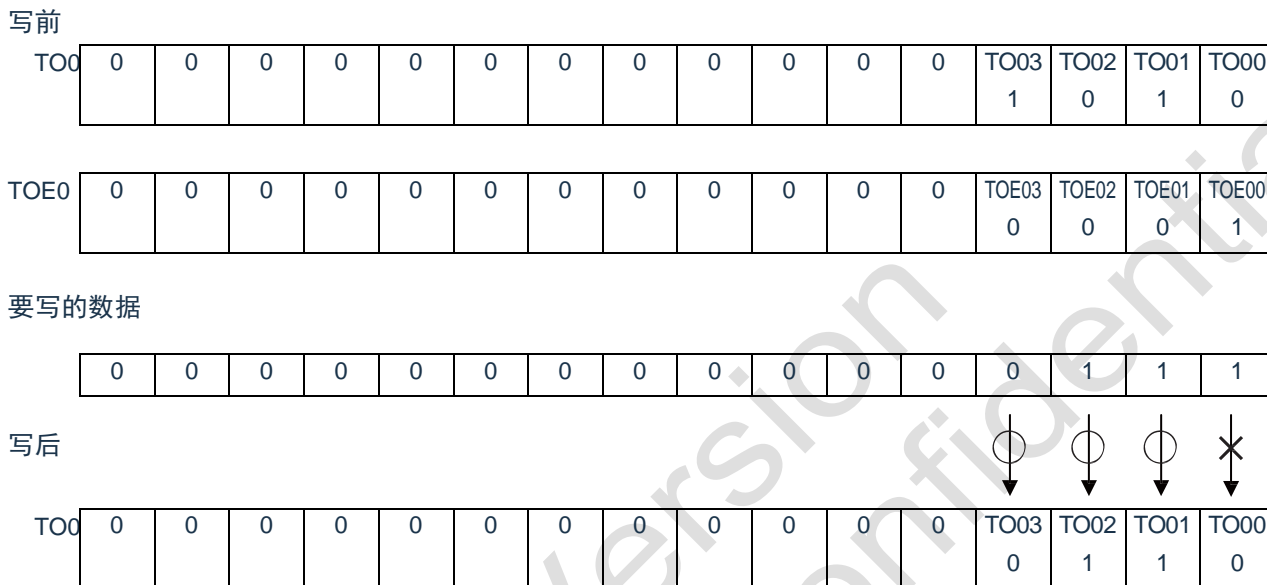
备注:

1. 内部复位信号: TOn引脚的复位/交替信号  
内部置位信号: TOn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。

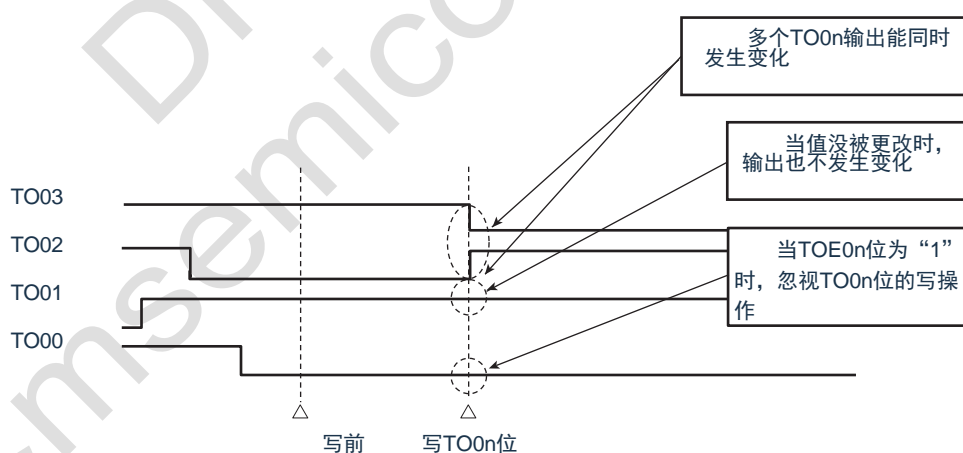
图6-36 TO0n位的一次性操作例子



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6-37 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)

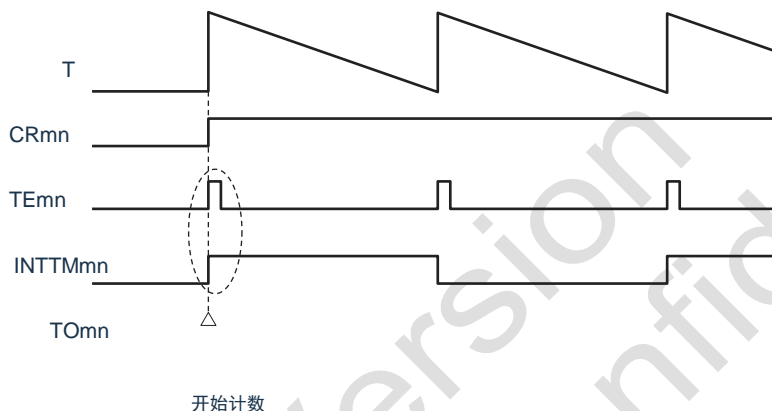
### 6.6.5 有关开始计数时的定时器中断和TOMn引脚输出

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位设置是否在开始计数时产生定时器中断。

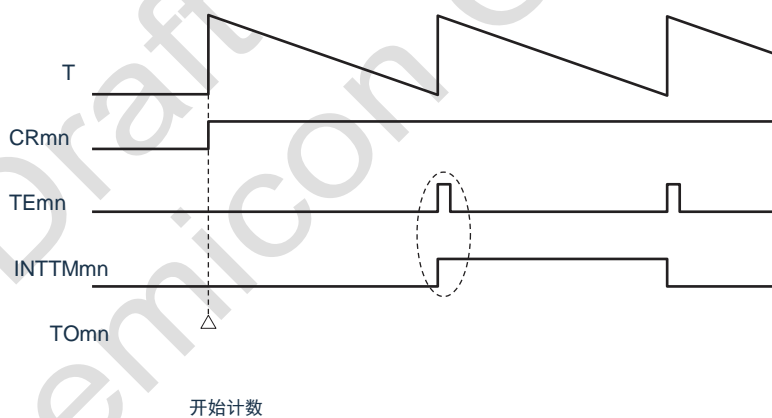
当MDmn0位为“1”时，能通过产生定时器中断(INTTMmn)得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和TOMn输出。设置为间隔定时器模式(TOEmn=1、TOMmn=0)时的运行例子如下所示。

图6-38 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。

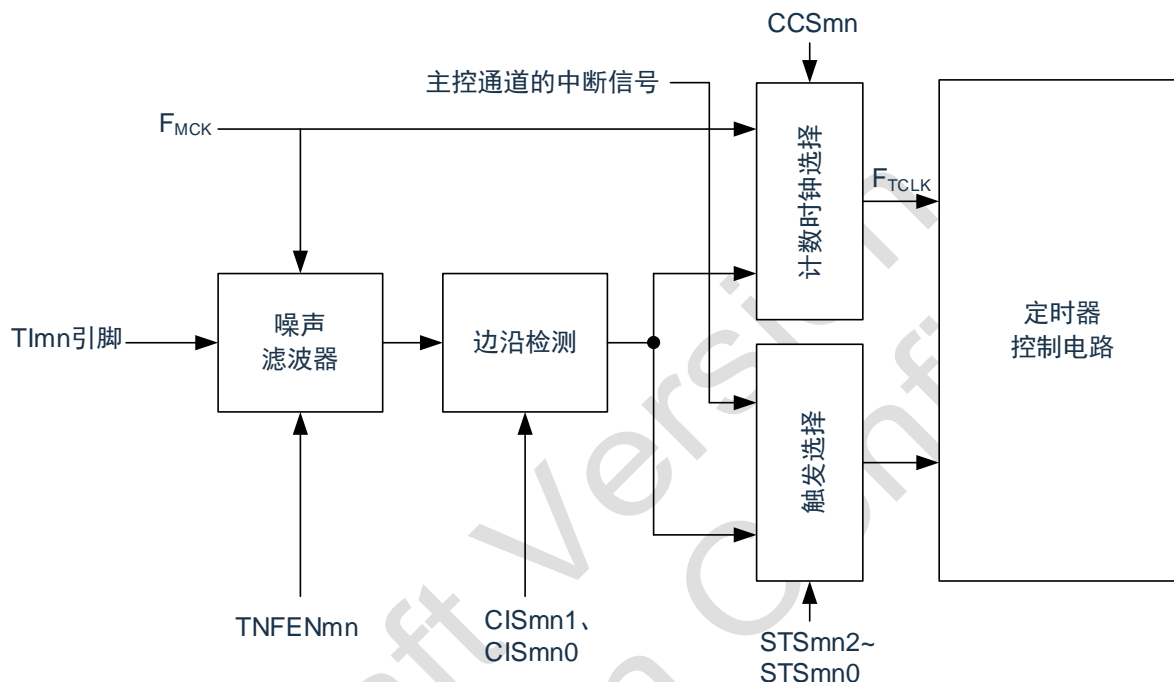
备注：m：单元号(m=0) n：通道号(n=0~3)

## 6.7 定时器输入(Tlmn)的控制

### 6.7.1 Tlmn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图所示。

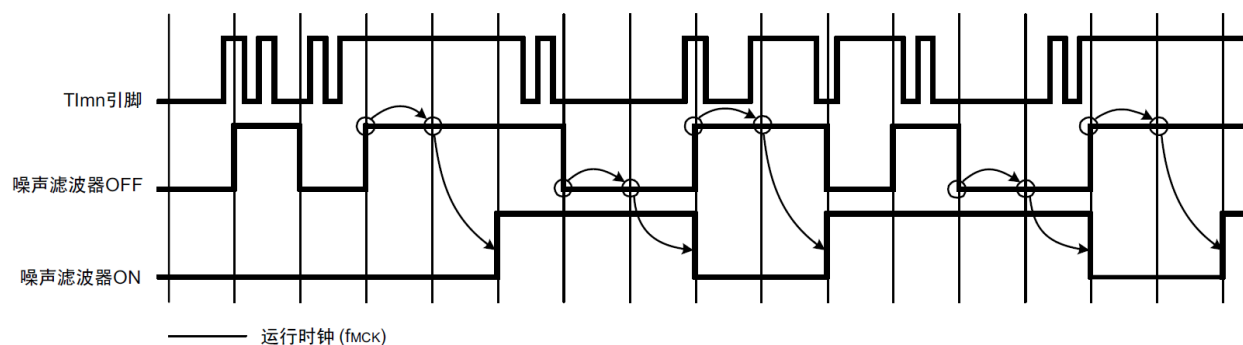
图6-39 输入电路的结构



### 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。Tlmn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6-40 Tlmn输入引脚在噪声滤波器ON或者OFF情况下的采样波形



注意：Tlmn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Tlmn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:

- $\text{TOmn输出的方波周期} = \text{计数时钟周期} \times (\text{TDRmn的设置值} + 1) \times 2$

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

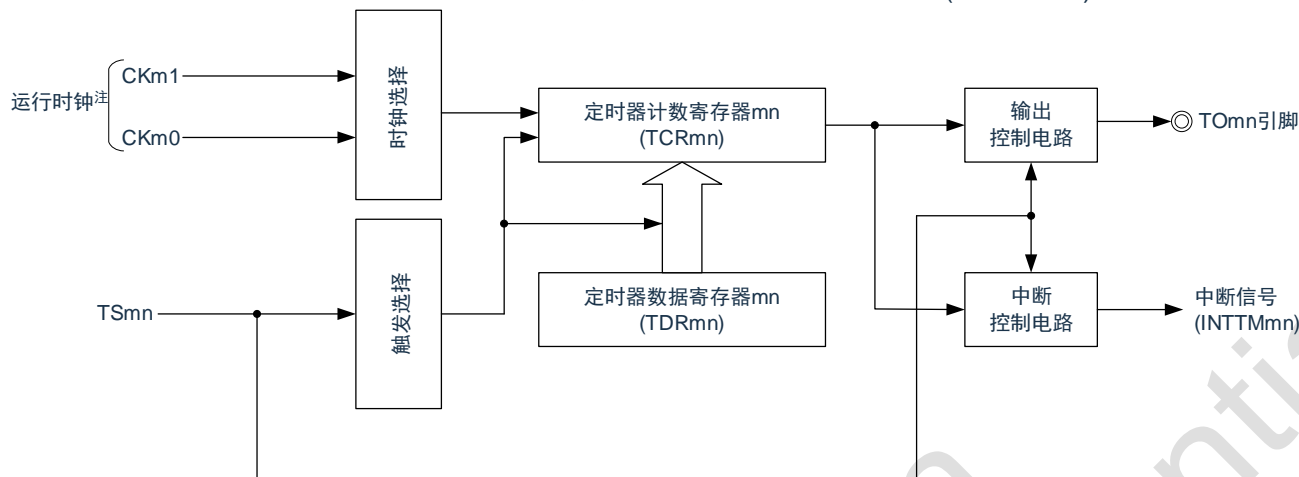
在间隔定时器模式中, 定时器计数寄存器mn(TCRmn)用作递减计数器。

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。

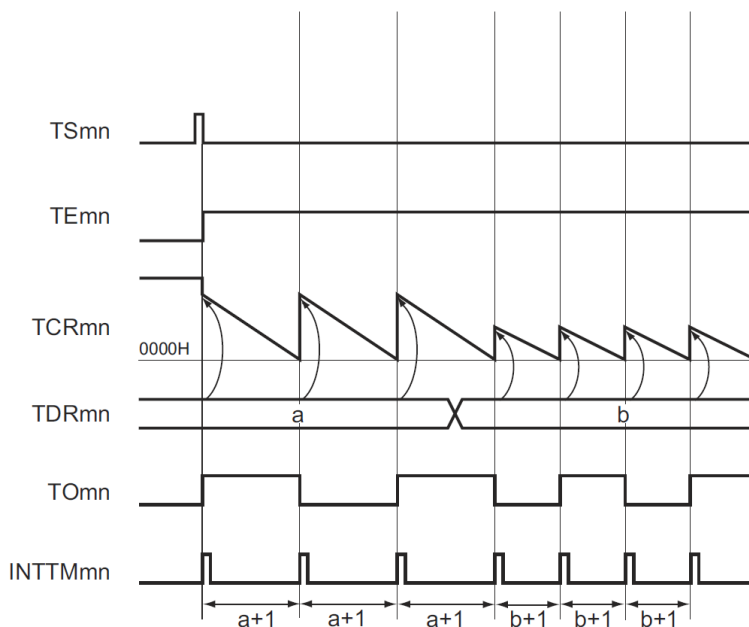
能随时改写TDRmn寄存器, 改写的TDRmn寄存器的值从下一个周期开始有效。

图6-41 作为间隔定时器/方波输出运行的基本时序例子(MDmn=1)



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

图6-42 作为间隔定时器/方波输出运行的基本时序例子(MDmn=0=1)

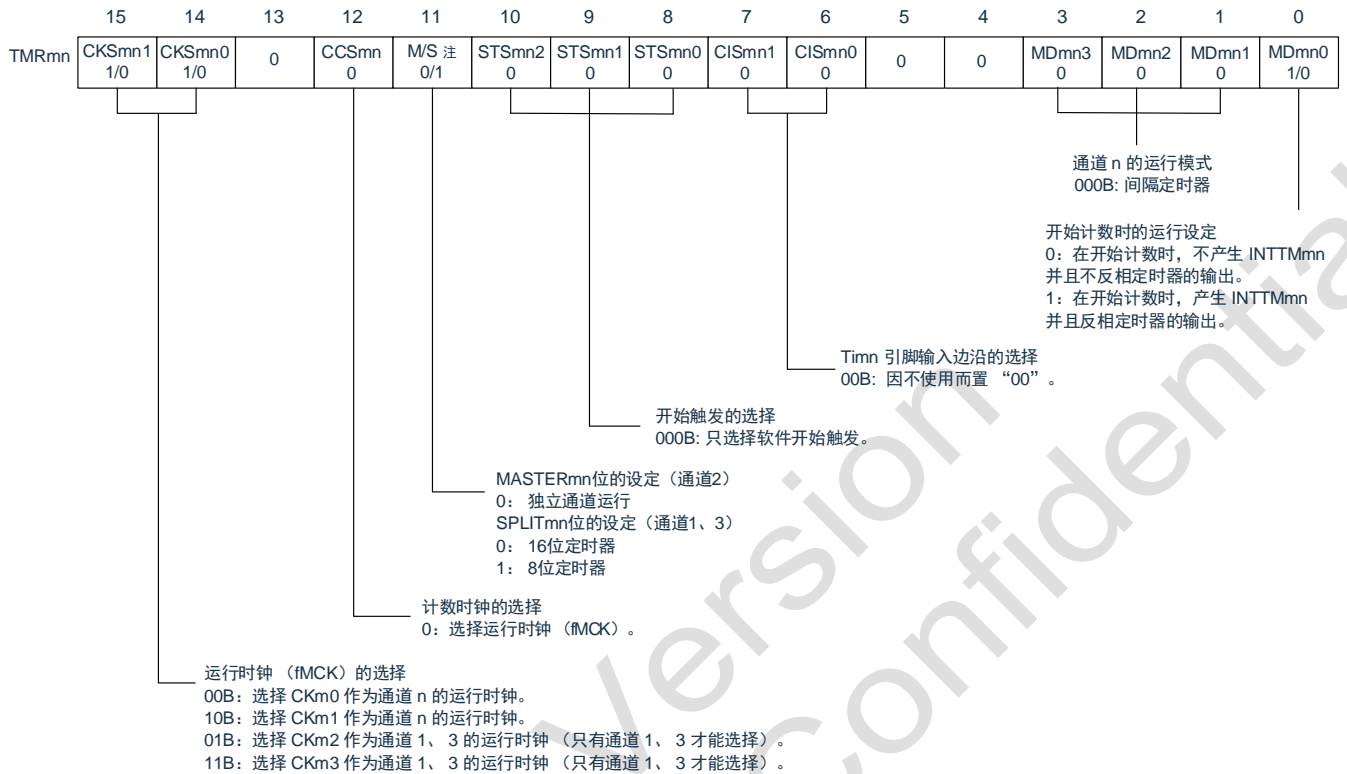


备注：

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6-43 间隔定时器/方波输出时的寄存器设置内容例子

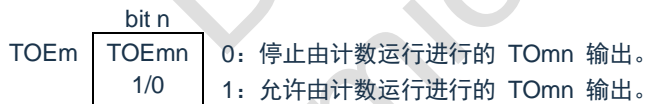
(a) 定时器模式寄存器mn(TMRmn)



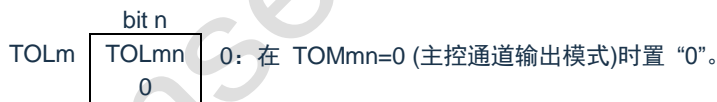
(b) 定时器输出寄存器 m (TOM)



(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6-44 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
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。 (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(TMRmm)(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。	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。
	将端口寄存器和端口模式寄存器置“0”。	
运行中	(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。
停止运行	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。
	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”。 将 TOEmn 位置“0”并且给 TOMn 位设定值。	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。 TOMn 引脚输出 TOMn 位设定的电平。
Timer4 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保持的值后将 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。

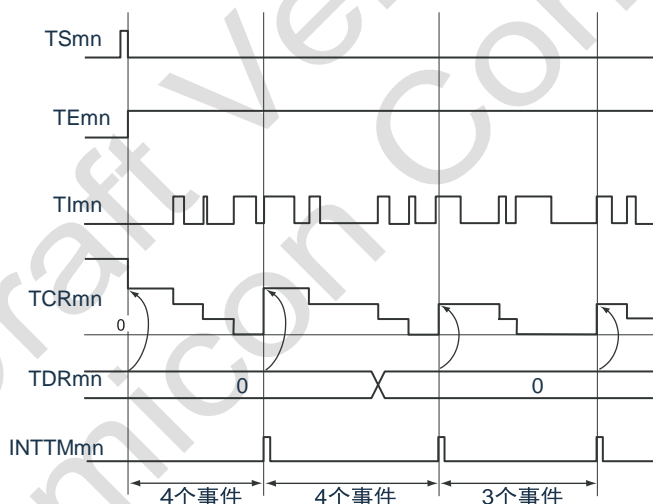
TCRmn寄存器在检测到TImn引脚输入的有效边沿的同时进行递减计数。如果TCRmn变为“0000H”，就再次装入TDRmn寄存器的值并且输出INTTMmn。

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6-45 作为外部事件计数器运行的基本时序例子

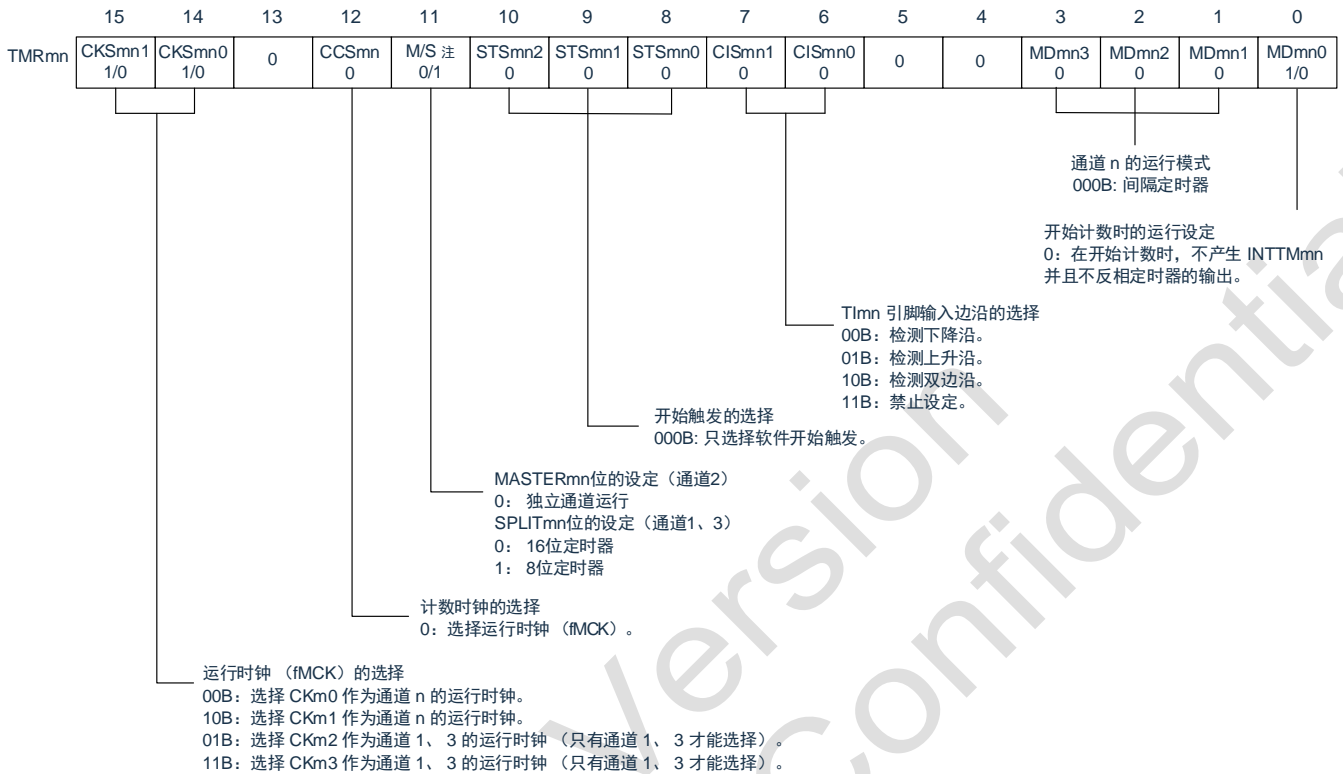


备注：

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6-46 外部事件计数器模式时的寄存器设置内容例子

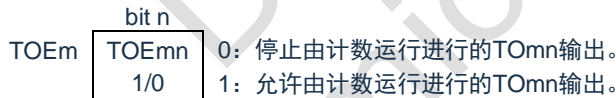
(a) 定时器模式寄存器mn(TMRmn)



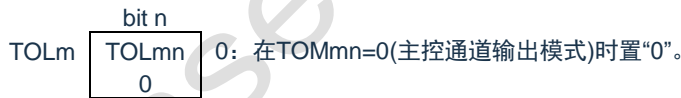
(b) 定时器输出寄存器 m(TOm)



(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6-47 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 INTTmnm。 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。

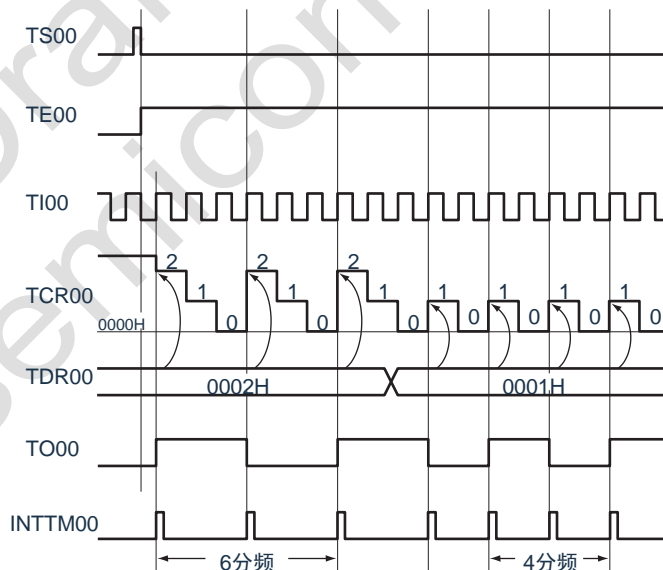
如果选择 TI00 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

$$T_{000} \text{输出的时钟周期} = \text{理想的} T_{000} \text{输出时钟周期} \pm \text{运行时钟周期 (误差)}$$

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6-48 作为分频器运行的基本时序例子(MD000=1)

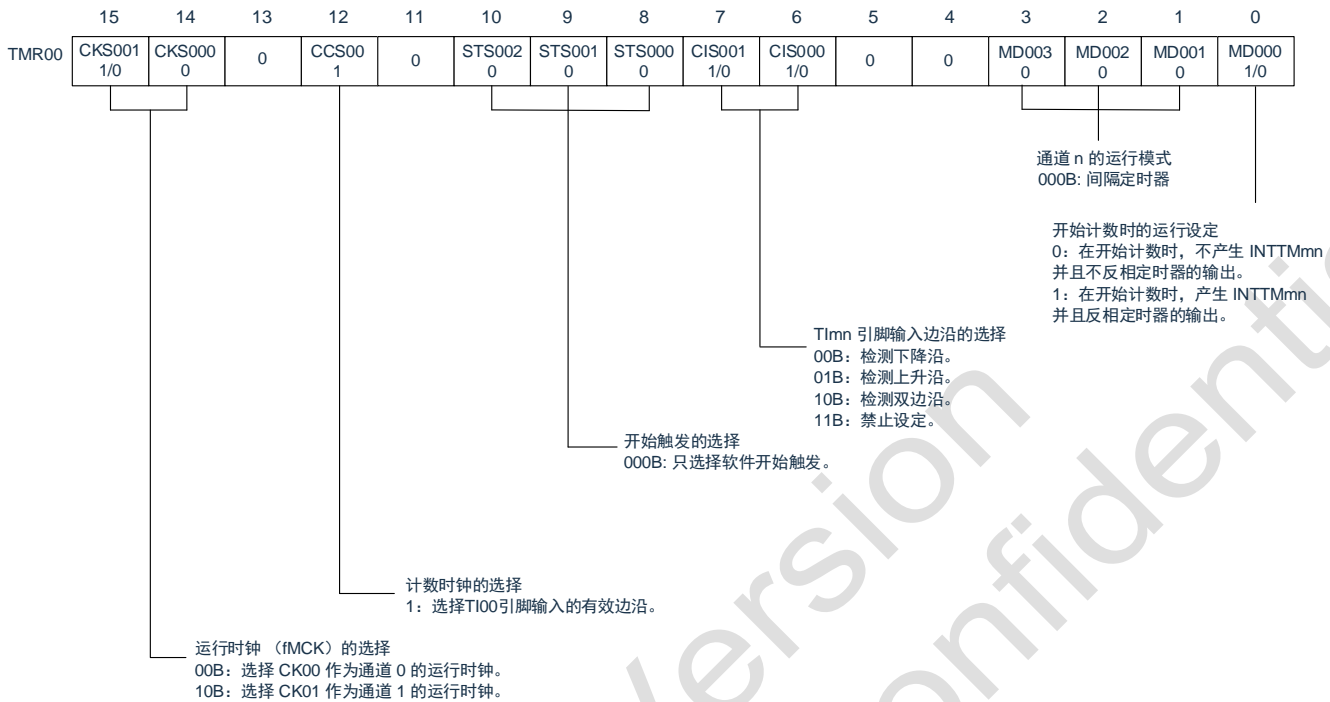


- 备注：TS00：定时器通道开始寄存器0(TS0)的 bit 0  
 TE00：定时器通道允许状态寄存器0(TE0)的 bit 0  
 TI00：TI00引脚输入信号  
 TCR00：定时器计数寄存器00(TCR00)  
 TDR00：定时器数据寄存器00(TDR00)  
 TO00：TO00引脚输出信号



图6-49 作为分频器运行时的寄存器设置内容例子

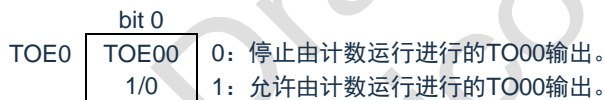
(a) 定时器模式寄存器00(TMR00)



(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6-50 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。 (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)
	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"。 设定TO00 位并且确定TO00 输出的初始电平。 →	TO00 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TO00 初始设定的电平。
开始 运行	将TOE00 位置 "1", 允许TO00 输出。 →	因为通道处于运行停止状态, 所以TO00 不变。
	将TOE00 位置 "1" (只限于重新开始运行)。 将TS00 位置 "1"。 因为TS00 位是触发位, 所以自动返回到 "0"。 →	TO00 引脚输出TO00 设定的电平。  TE00位变为 "1" 并且开始计数。 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。
运行中	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"。 因为TT00位是触发位, 所以自动返回到 "0"。 →	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。
停止 运行	将TOE00 位置 "0" 并且给TO00 位设定值。 →	TO00 引脚输出TO00 位设定的电平。
	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。
Timer4 停止	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)

重新开始运行

### 6.8.4 作为输入脉冲间隔测量的运行

能在Tl<sub>mn</sub>有效边沿捕捉计数值，测量Tl<sub>mn</sub>输入脉冲的间隔。在TE<sub>mn</sub>位为“1”的期间，也能将软件操作(TS<sub>mn</sub>=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：

$$Tl_{mn} \text{输入脉冲间隔} = \text{计数时钟的周期} \times ((10000H \times \text{TSR}_{mn}:\text{OVF}) + (\text{TDR}_{mn} \text{的捕捉值} + 1))$$

注意：因为通过定时器模式寄存器mn(TMR<sub>mn</sub>)的CKS<sub>mn</sub>位选择的运行时钟对Tl<sub>mn</sub>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(TCR<sub>mn</sub>)用作递增计数器。

如果将定时器通道开始寄存器m(TS<sub>m</sub>)的通道开始触发位(TS<sub>mn</sub>)置“1”，TCR<sub>mn</sub>寄存器就通过计数时钟从“0000H”开始递增计数。

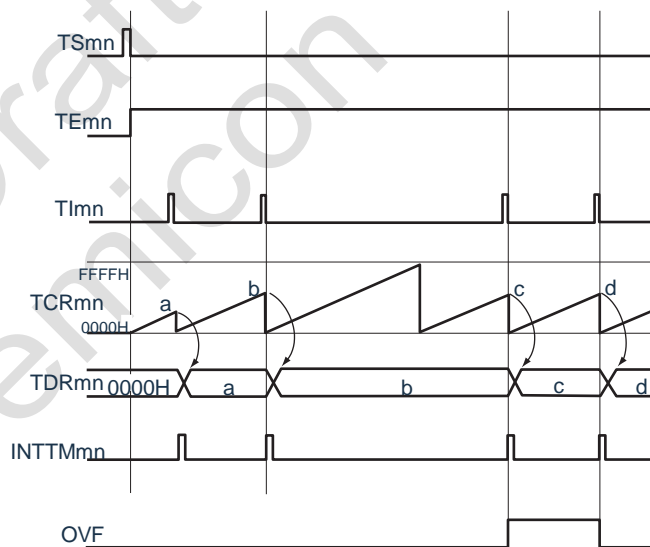
如果检测到Tl<sub>mn</sub>引脚输入的有效边沿，就将TCR<sub>mn</sub>寄存器的计数值传送(捕捉)到定时器数据寄存器mn(TDR<sub>mn</sub>)，同时将TCR<sub>mn</sub>寄存器清“0000H”，然后输出INTT<sub>Mmn</sub>。此时，如果计数器发生上溢，就将定时器状态寄存器mn(TSR<sub>mn</sub>)的OVF位置“1”。如果计数器没有发生上溢，就清除OVF位。此后，继续同样的运行。

在将计数值捕捉到TDR<sub>mn</sub>寄存器的同时，根据在测量期间是否发生上溢，更新TSR<sub>mn</sub>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TSR<sub>mn</sub>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TMR<sub>mn</sub>寄存器的STS<sub>mn2</sub>~STS<sub>mn0</sub>位置“001B”，并且将Tl<sub>mn</sub>的有效边沿用于开始触发和捕捉触发。

图6-51 作为输入脉冲间隔测量的运行基本时序例子(MD<sub>mn0</sub>=0)

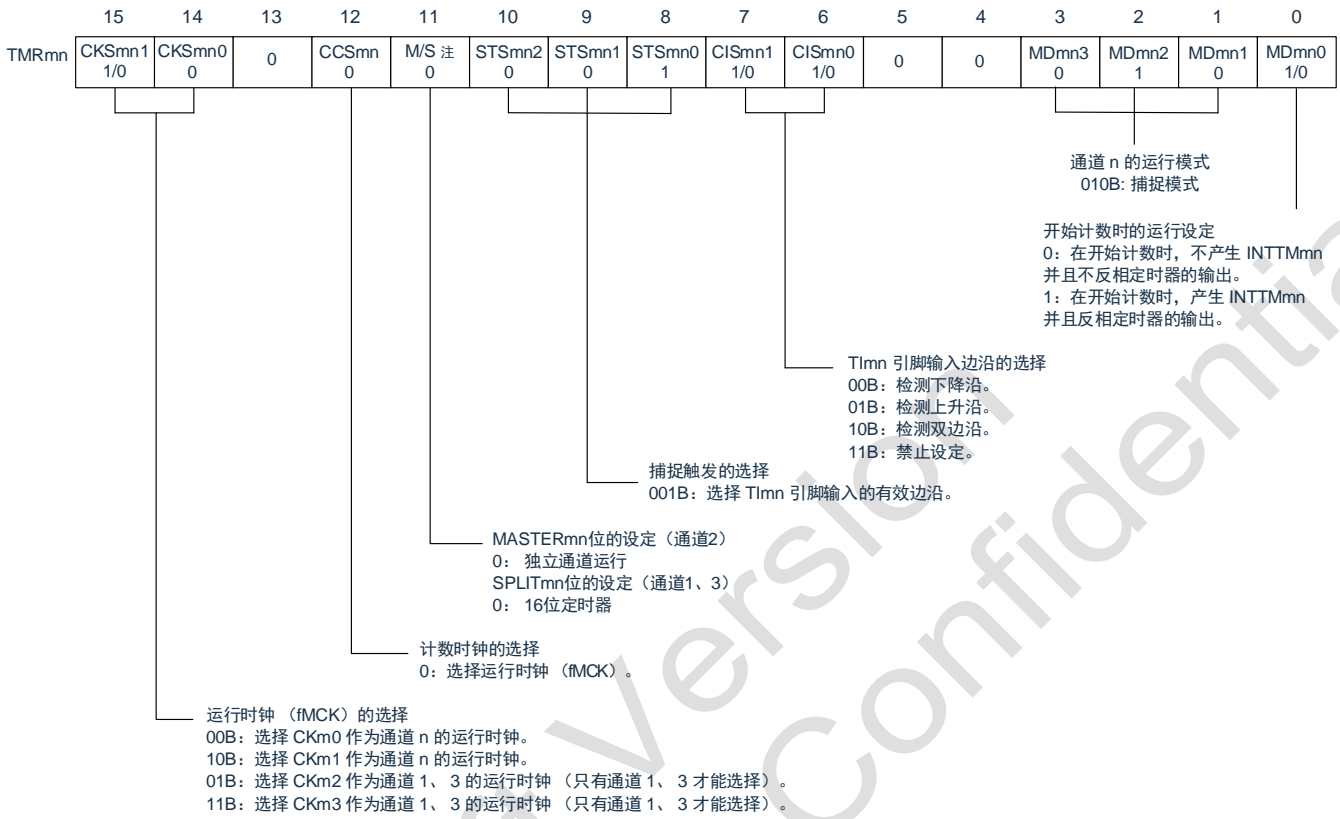


备注：

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. TS<sub>mn</sub>: 定时器通道开始寄存器m(TS<sub>m</sub>)的bit n  
 TE<sub>mn</sub>: 定时器通道允许状态寄存器m(TE<sub>m</sub>)的bit n  
 Tl<sub>mn</sub>: Tl<sub>mn</sub>引脚输入信号  
 TCR<sub>mn</sub>: 定时器计数寄存器mn(TCR<sub>mn</sub>)  
 TDR<sub>mn</sub>: 定时器数据寄存器mn(TDR<sub>mn</sub>)  
 OVF: 定时器状态寄存器mn(TSR<sub>mn</sub>)的bit 0

图6-52 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)



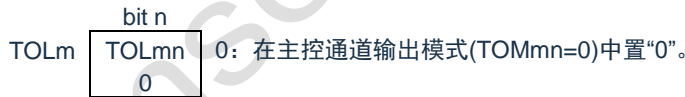
(b) 定时器输出寄存器m(TOm)



(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6-53 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n 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。

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

能通过通过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：1. 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。

2. 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。

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INTTm。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。

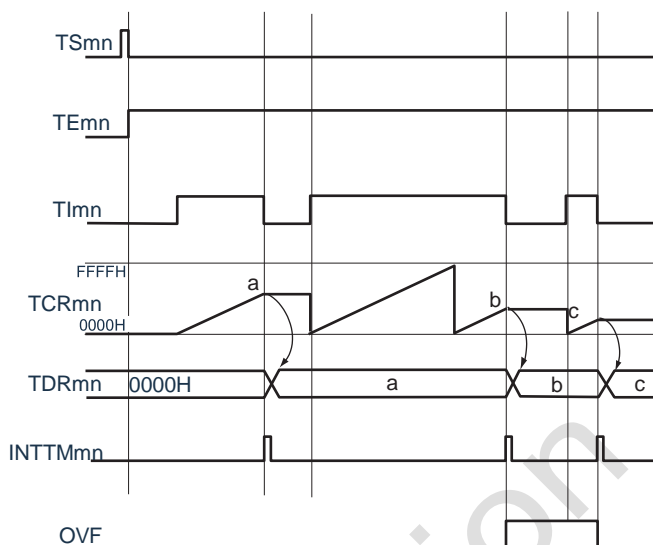
即使计数器进行了2个周期或者2个周期以上的完整计数，也认为发生上溢而将TSRmn寄存器的OVF位置“1”。但是，在发生2次或者2次以上的上溢时，无法通过OVF位正常测量间隔值。

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6-54 作为输入信号高低电平宽度测量的运行基本时序例子

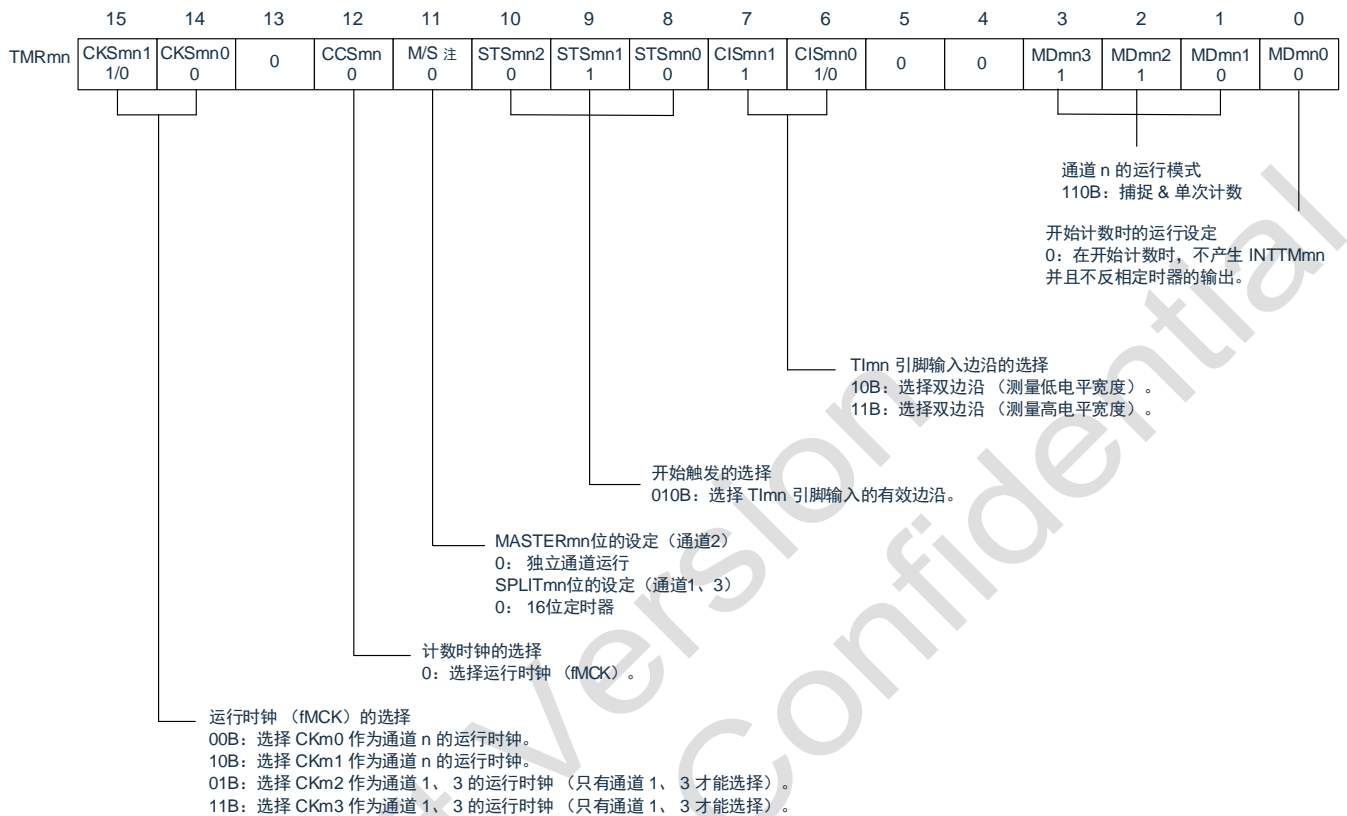


备注:

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6-55 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)



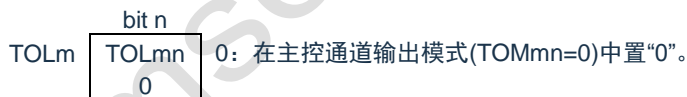
(b) 定时器输出寄存器m(TOm)



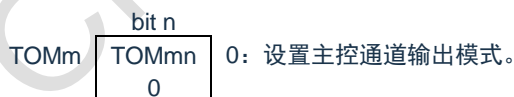
(c) 定时器输出允许寄存器m(TOEm)



(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6-56 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	将外围允许寄存器0 (PER0) 的TM4mEN 位置 "1" 。 →	定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm) 。 确定CKm0 ~ CKm3 的时钟频率。	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	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Tlmn 引脚输入的有效边沿或者将TSmn 位置 "1" ) 的检测等待状态。
	检测 Tlmn 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。
重新 开始 运行	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 Tlmn 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 Tlmn 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 Tlmn 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" 。
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)

### 6.8.6 作为延迟计数器的运行

能通过Tl<sub>mn</sub>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INTTM<sub>mn</sub>。

在TE<sub>mn</sub>位为“1”的期间, 能通过软件将TS<sub>mn</sub>位置“1”, 开始递减计数, 并且以任意的设置间隔产生INTTM<sub>mn</sub>(定时器中断)。

中断产生周期能用以下计算式进行计算:

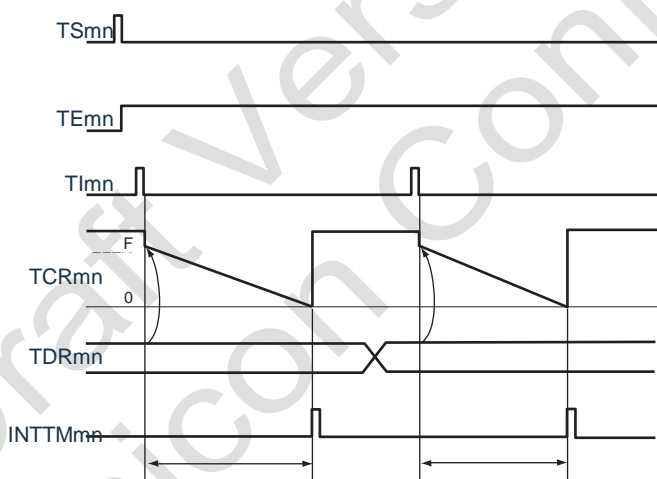
$$\text{INTTM}_{mn} \text{ (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR}_{mn} \text{的设置值} + 1)$$

在单次计数模式中, 定时器计数寄存器mn(TCR<sub>mn</sub>)用作递减计数器。

如果将定时器通道开始寄存器m(TS<sub>m</sub>)的通道开始触发位(TS<sub>mn</sub>、TSH<sub>m1</sub>、TSH<sub>m3</sub>)置“1”, TE<sub>mn</sub>位、TEH<sub>m1</sub>位和TEH<sub>m3</sub>位就变为“1”, 并且进入Tl<sub>mn</sub>引脚的有效边沿检测等待状态。通过Tl<sub>mn</sub>引脚输入的有效边沿检测, 开始TCR<sub>mn</sub>寄存器的运行, 并且装入定时器数据寄存器mn(TDR<sub>mn</sub>)的值。TCR<sub>mn</sub>寄存器通过计数时钟, 从装入的TDR<sub>mn</sub>寄存器的值开始递减计数。如果TCR<sub>mn</sub>变为“0000H”, 就输出INTTM<sub>mn</sub>, 并且在检测到下一个Tl<sub>mn</sub>引脚输入的有效边沿前停止计数。

能随时改写TDR<sub>mn</sub>寄存器, 改写的TDR<sub>mn</sub>寄存器的值从下一个周期开始有效。

图6-57 作为延迟计数器的运行基本时序例子

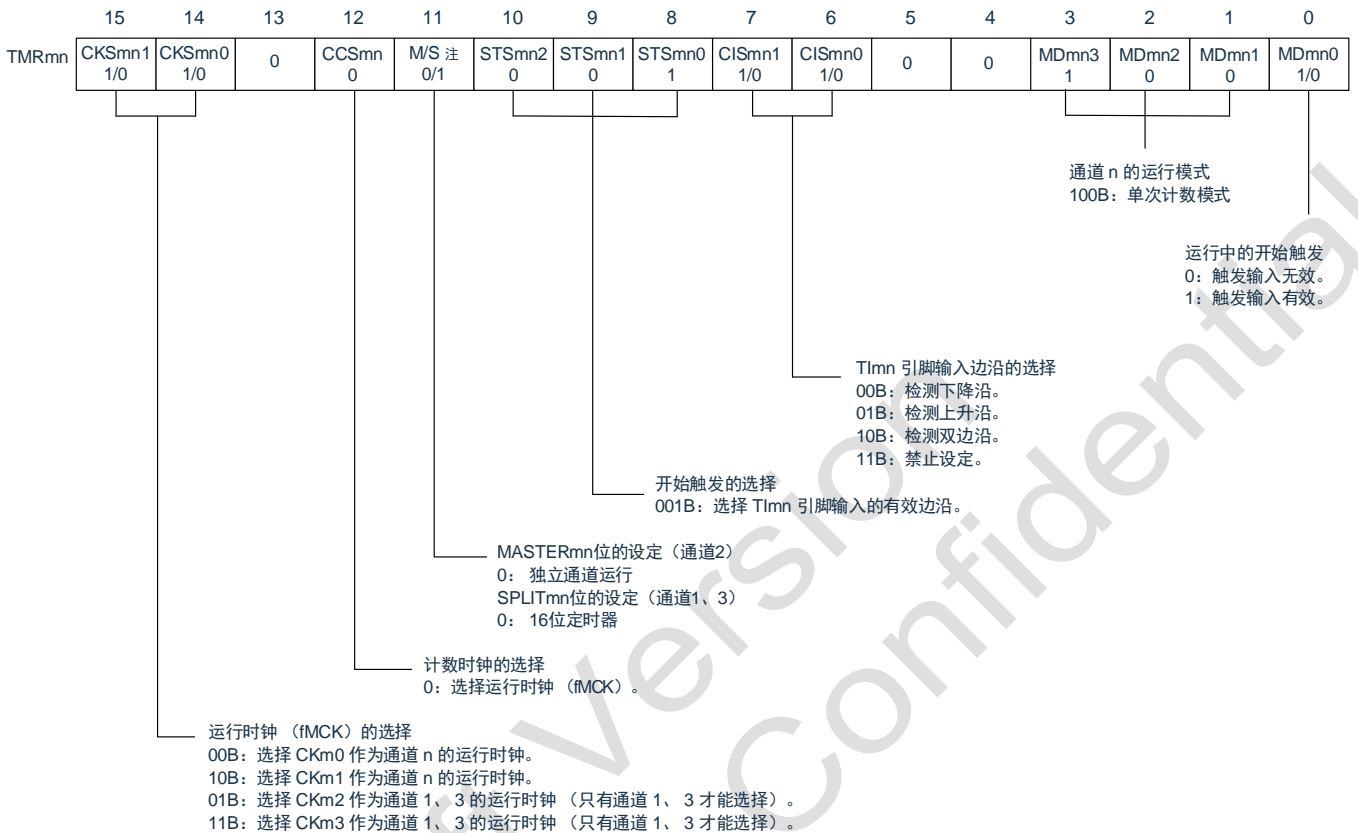


备注:

1. m: 单元号(m=0) n: 通道号(n=0~3)
2. TS<sub>mn</sub>: 定时器通道开始寄存器m(TS<sub>m</sub>)的bit n  
 TE<sub>mn</sub>: 定时器通道允许状态寄存器m(TE<sub>m</sub>)的bit n  
 Tl<sub>mn</sub>: Tl<sub>mn</sub>引脚输入信号  
 TCR<sub>mn</sub>: 定时器计数寄存器mn(TCR<sub>mn</sub>)  
 TDR<sub>mn</sub>: 定时器数据寄存器mn(TDR<sub>mn</sub>)

图6-58 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)



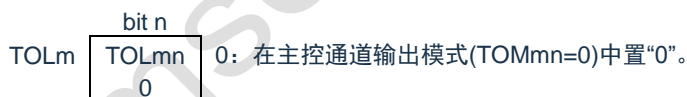
(b) 定时器输出寄存器m(TOm)



(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6-59 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	将外围允许寄存器0 (PER0) 的TM4mEN 位置 “1”。	定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0 ~ CKm3 的时钟频率。	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	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”。 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TImn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。

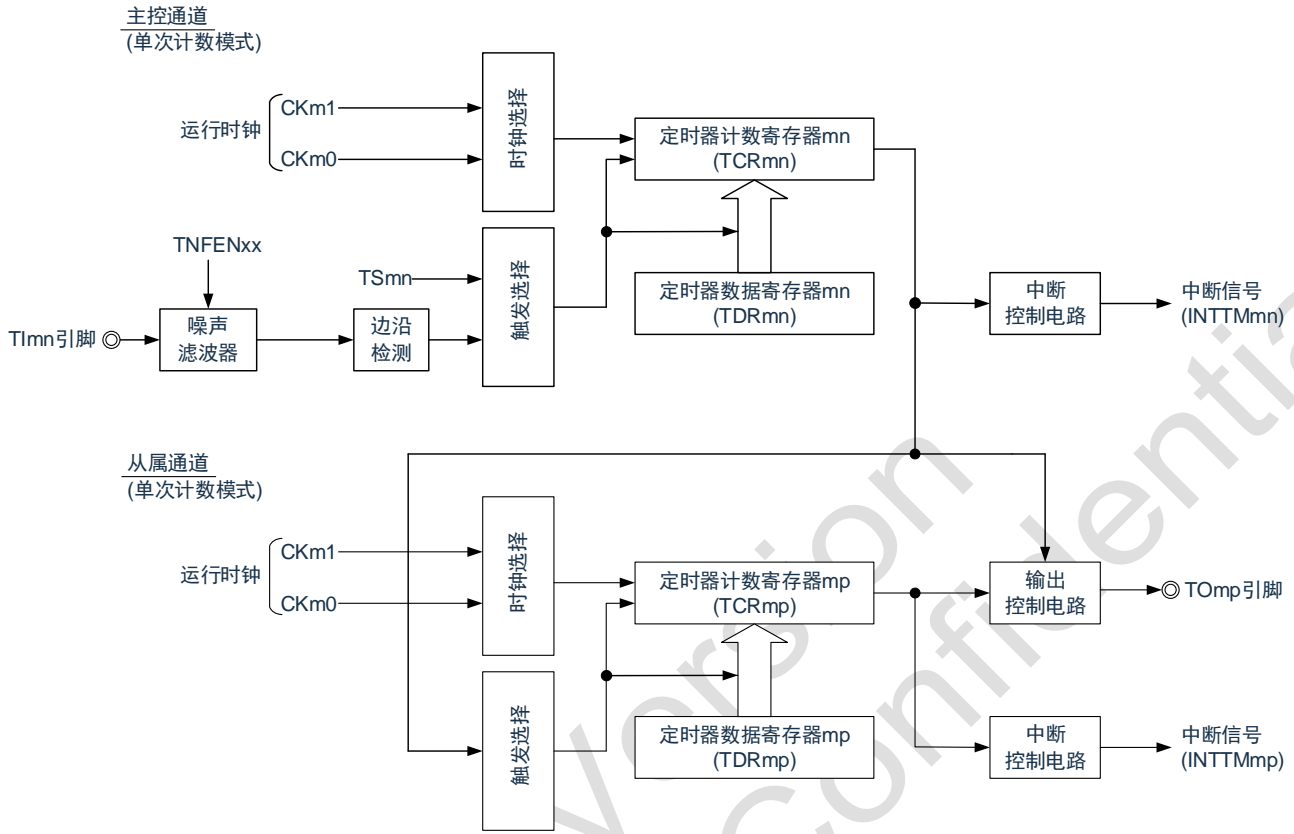
在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的INTTMmn作为开始触发，从属通道的TCRmp寄存器开始运行并且装入TDRmp寄存器的值。TCRmp寄存器通过计数时钟，从装入的TDRmp寄存器值开始递减计数。如果计数值变为“0000H”，就输出INTTMmp，并且在检测到下一个开始触发(主控通道的INTTMmn)前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOMP的输出电平变为有效电平，如果TCRmp变为“0000H”，就变为无效电平。

不使用TImn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。

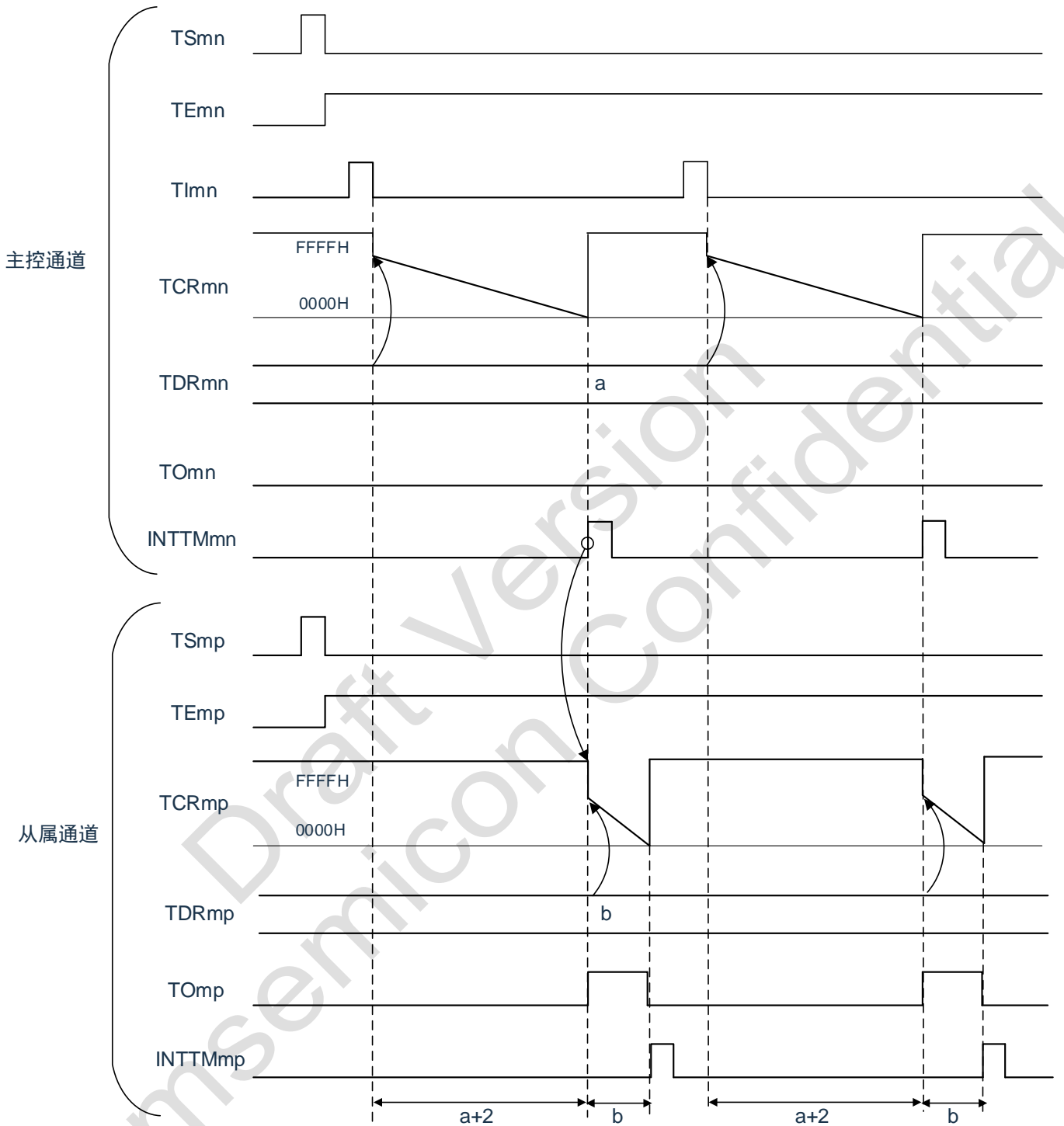
**备注：** m：单元号(m=0) n：主控通道号(n=0、2)  
 p：从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-60 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6-61 作为单触发脉冲输出功能的运行基本时序例子

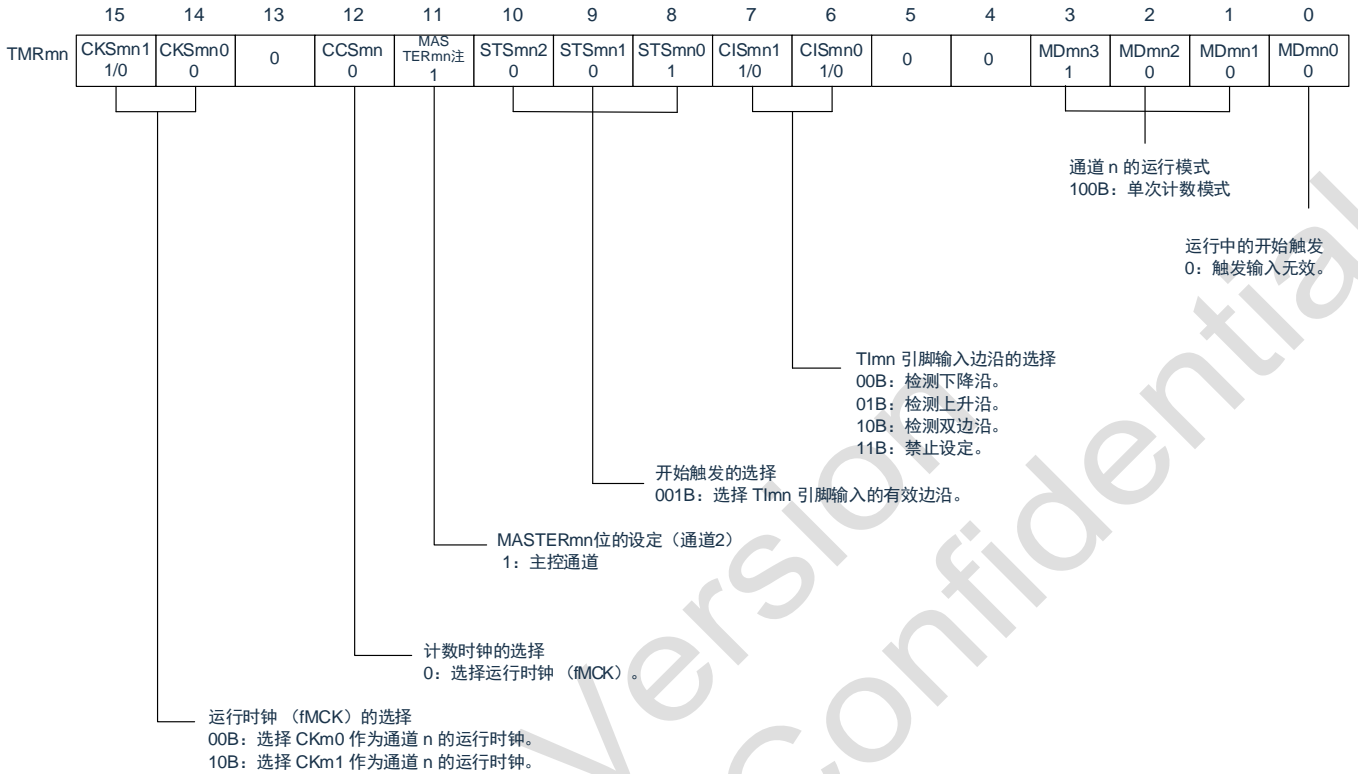


备注:

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6-62 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

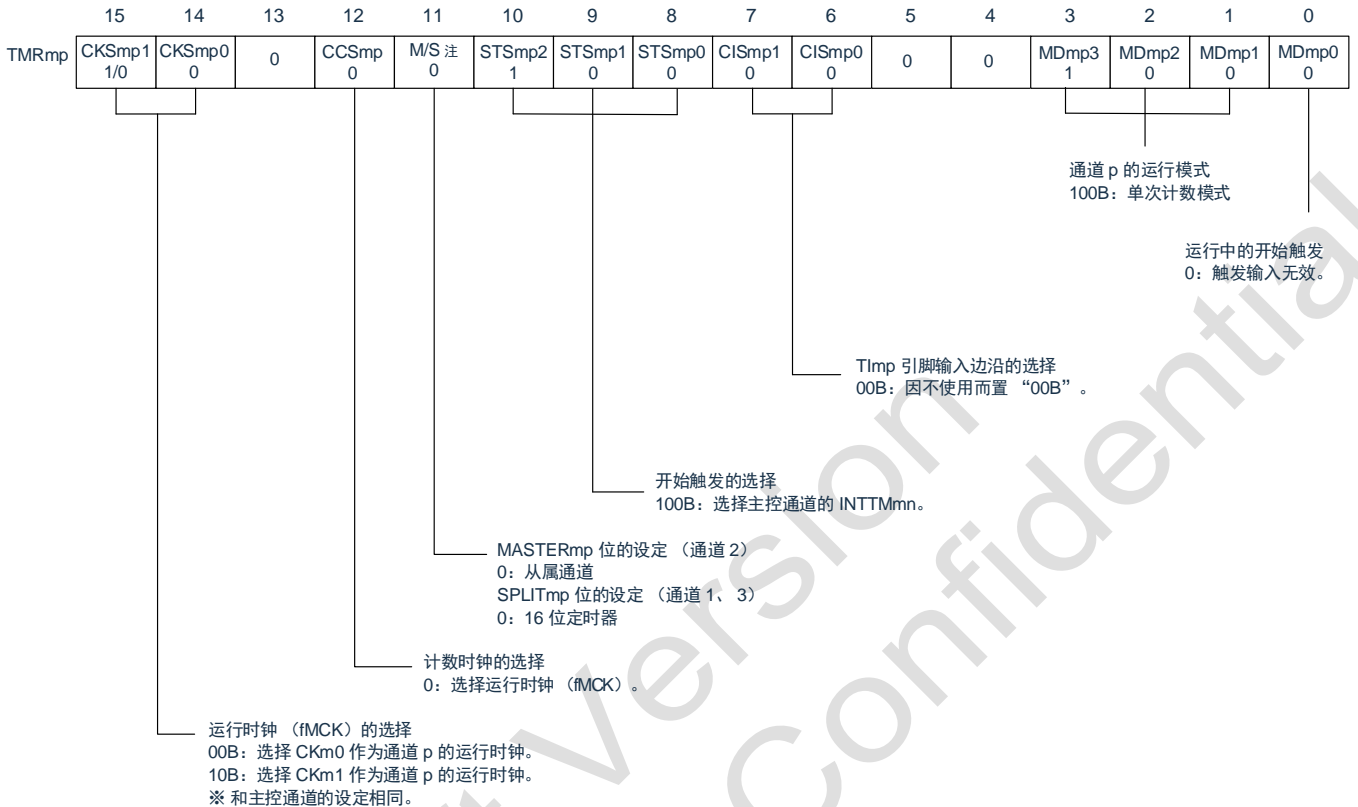
TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 主控通道号(n=0、2)



图6-63 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6-64 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6-65 单触发脉冲输出功能时的操作步骤 (2/2)

重新开始运行	<p>将 TOEmp 位 (从属) 置 “1” (只限于重新开始运行)。</p> <p>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”。</p> <p>因为 TSMn 位和 TSmp 位是触发位, 所以自动返回 “0”。</p>	<p>TEmn 位和 TEmP 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。</p>
	<p>通过检测主控通道的开始触发, 开始主控通道的计数。</p> <ul style="list-style-type: none"> <li>检测 TImn 引脚输入的有效边沿。</li> <li>通过软件将主控通道的 TSMn 位置 “1” 注。</li> </ul>	<p>主控通道开始计数。</p>
运行中	<p>只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。</p> <p>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。</p> <p>能随时读 TCRmn 寄存器和 TCRmp 寄存器。</p> <p>不使用 TSRmn 寄存器和 TSRmp 寄存器。</p> <p>能更改从属通道的 TOM 寄存器和 TOEm 寄存器的设定值。</p>	<p>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。</p> <p>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。</p>
停止运行	<p>将 TTmn 位 (主控) 和 TTmp 位 (从属) 同时置 “1”。</p> <p>因为 TTmn 位和 TTmp 位是触发位, 所以自动返回 “0”。</p>	<p>TEmn 位和 TEmP 位都变为 “0” 并且停止计数。</p> <p>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。</p>
	<p>将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设定值。</p>	<p>TOmp 引脚输出 TOmp 设定的电平。</p>
Timer4 停止	<p>要保持 TOmp 引脚输出电平的情况:</p> <p>在给端口寄存器设定要持有的值后将 TOmp 位置 “0”。</p> <p>不需要保持 TOmp 引脚输出电平的情况:</p> <p>不需要设定。</p>	<p>通过端口功能保持 TOmp 引脚的输出电平。</p>
	<p>将 PER0 寄存器的 TM4mEN 位置 “0”。</p>	<p>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。</p>

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：

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

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。

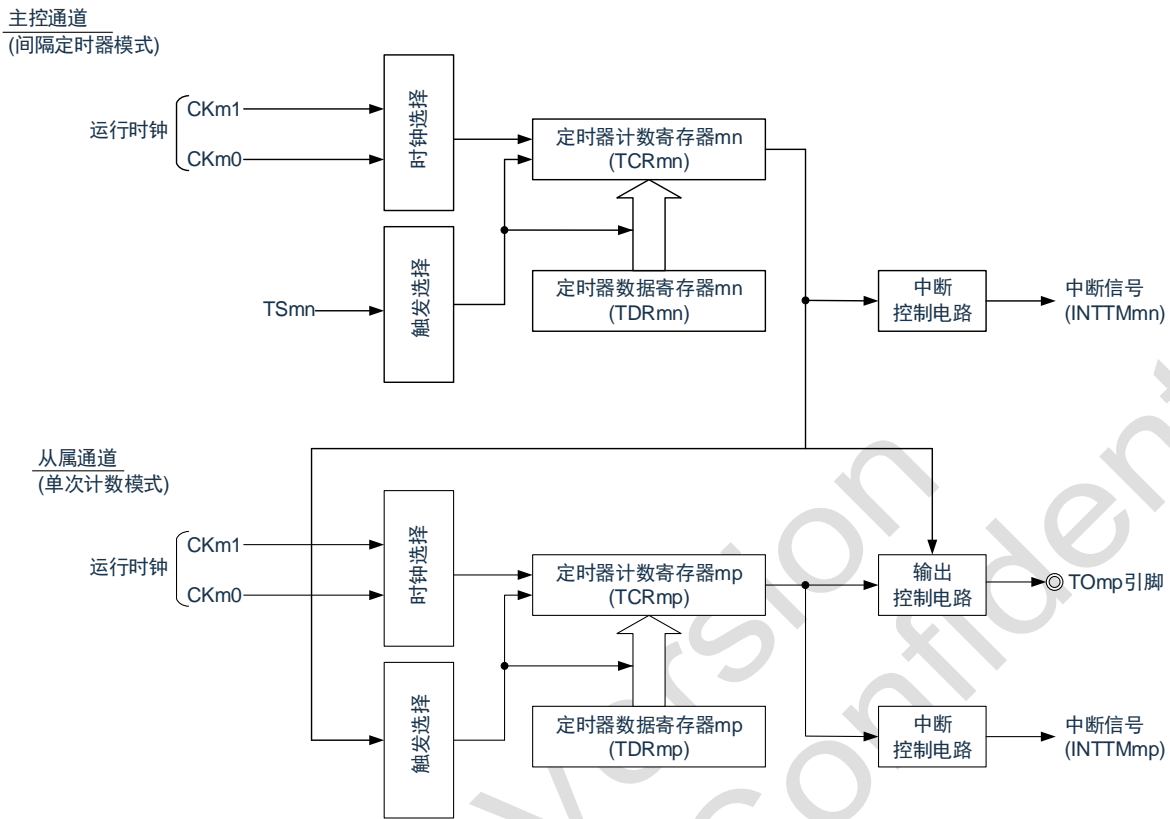
注意：要同时改写主控通道的定时器数据寄存器mn(TDRmn)和从属通道的TDRmp寄存器时，需要2次写存取。

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)

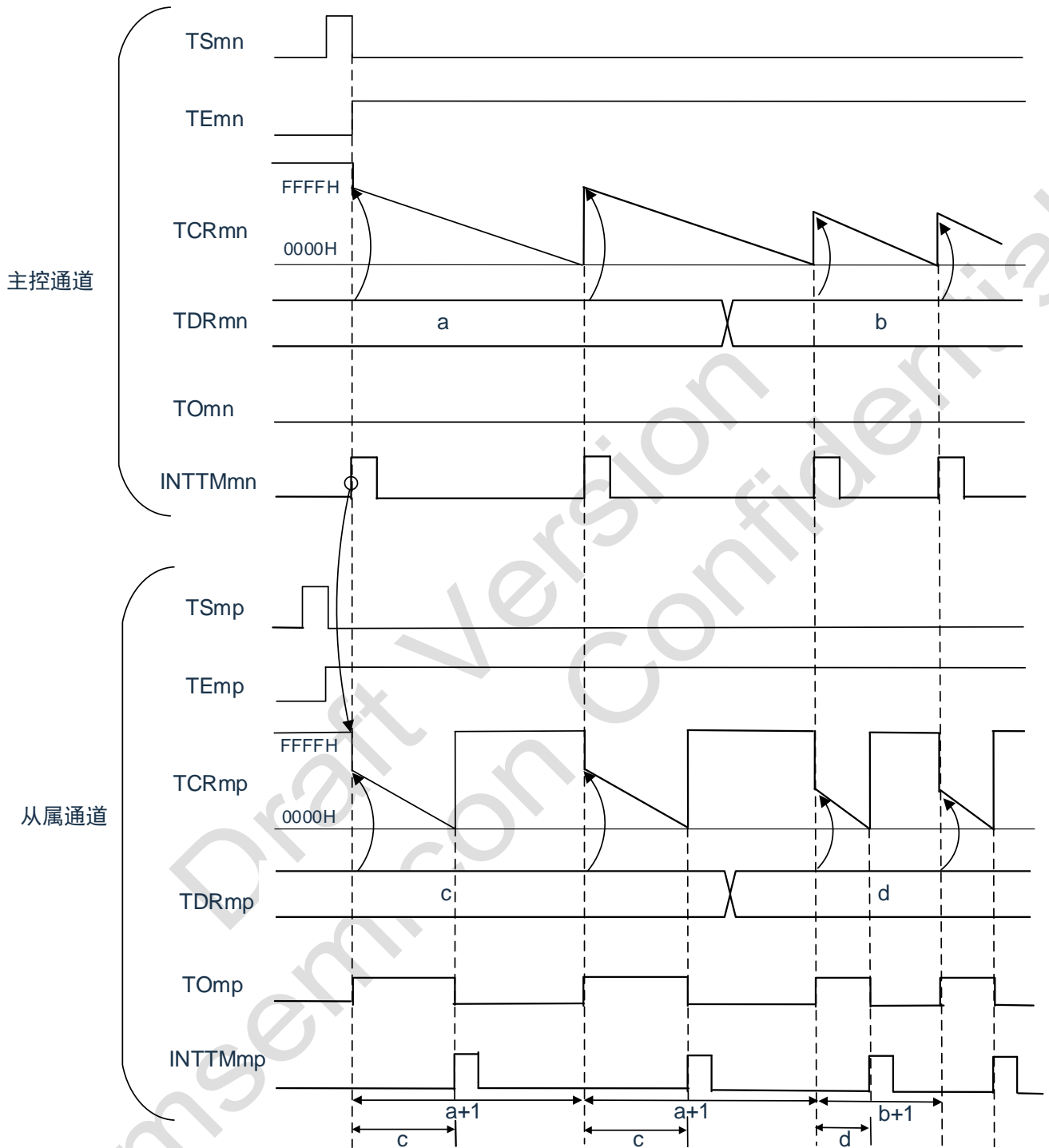
p：从属通道号(n=0: p=1、2、3, n=2: p=3)

图6-66 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6-67 作为PWM功能的运行基本时序例子

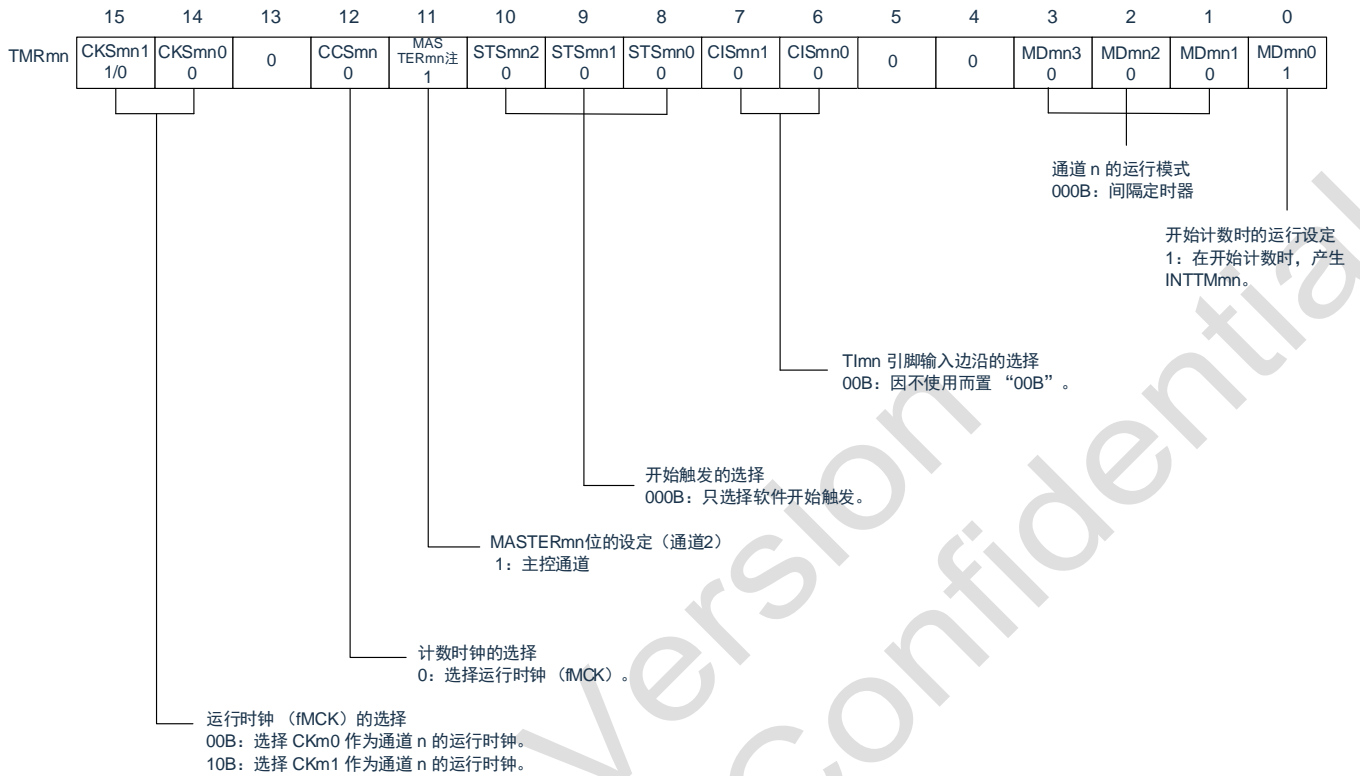


备注:

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6-68 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)



(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)



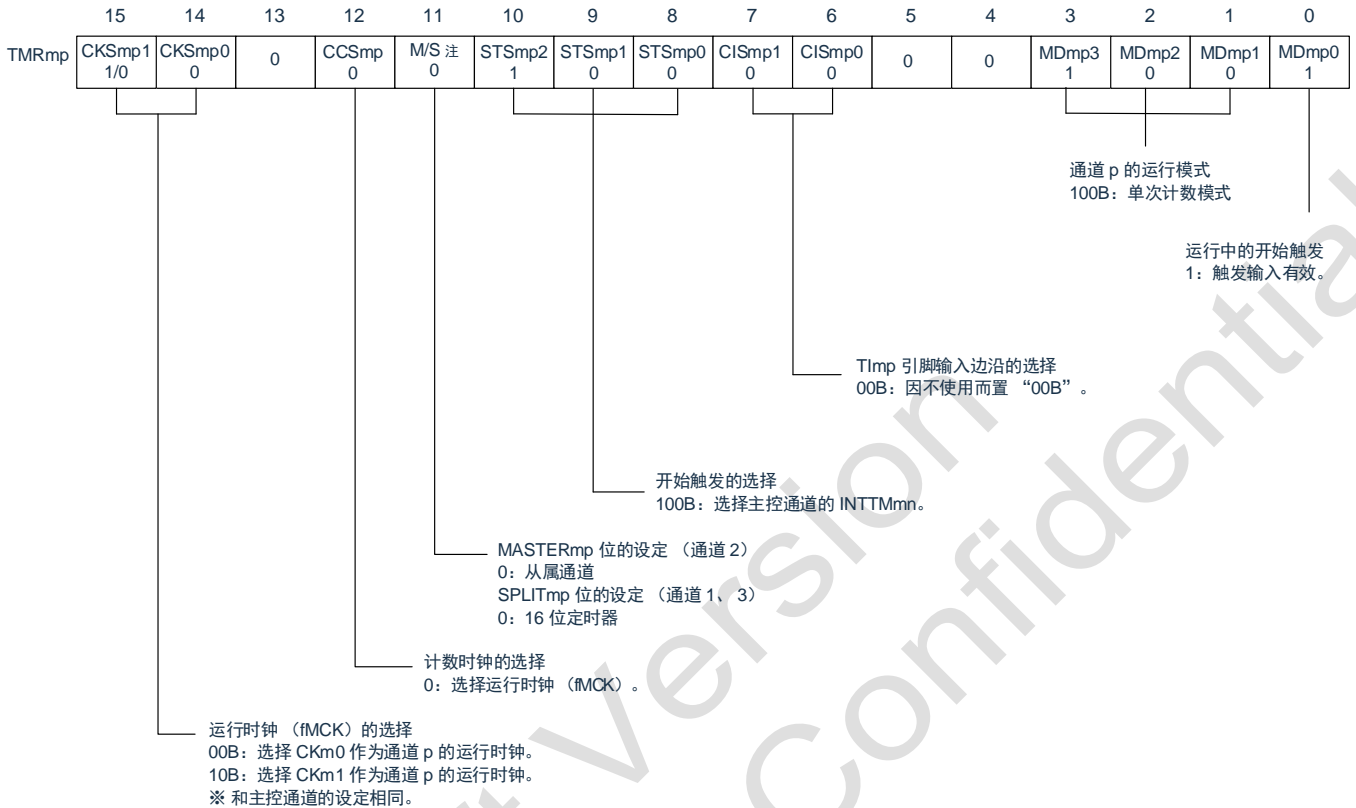
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

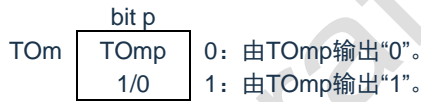
备注: m: 单元号(m=0) n: 主控通道号(n=0、2)

图6-69 PWM功能时(从属通道)的寄存器设置内容例子

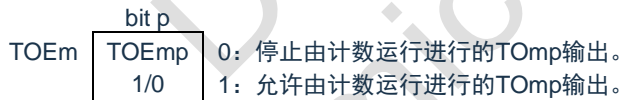
(a) 定时器模式寄存器 mp(TMRmp)



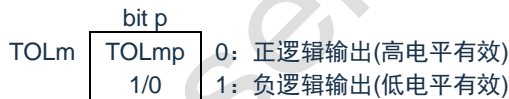
(b) 定时器输出寄存器 m(TOm)



(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6-70 PWM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	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。	TOmp 引脚处于 Hi-Z 输出状态。
	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" 。	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6-71 PWM功能时的操作步骤 (2/2)

重新开始运行	开始运行	<p>将 TOEmp 位 (从属) 置 "1" (只限于重新开始运行)。</p> <p>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"。</p> <p>因为 TSmn 位和 TSmp 位是触发位, 所以自动返回到 "0"。</p>	<p>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。</p>
	运行中	<p>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。</p> <p>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。</p> <p>能随时读 TCRmn 寄存器和 TCRmp 寄存器。</p> <p>不使用 TSRmn 寄存器和 TSRmp 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,</p> <p>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。</p>
	停止运行	<p>将 TTmn 位 (主控) 和 TTmp 位 (从属) 同时置 "1"。</p> <p>因为 TTmn 位和 TTmp 位是触发位, 所以自动返回到 "0"。</p>	<p>TEmn 位和 TEmP 位都变为 "0" 并且停止计数。</p> <p>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。</p>
	Timer4 停止	<p>将从属通道的 TOEmp 位置 "0" 并且给 TOmp 位设定值。</p> <p>要保持 TOmp 引脚输出电平的情况: 在端口寄存器设定要保持的值后将 TOmp 位置 "0"。</p> <p>不需要保持 TOmp 引脚输出电平的情况: 不需要设定。</p> <p>将 PER0 寄存器的 TM4mEN 位置 "0"。</p>	<p>TOmp 引脚输出 TOmp 设定的电平。</p> <p>通过端口功能保持 TOmp 引脚的输出电平。</p> <p>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)</p>

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)

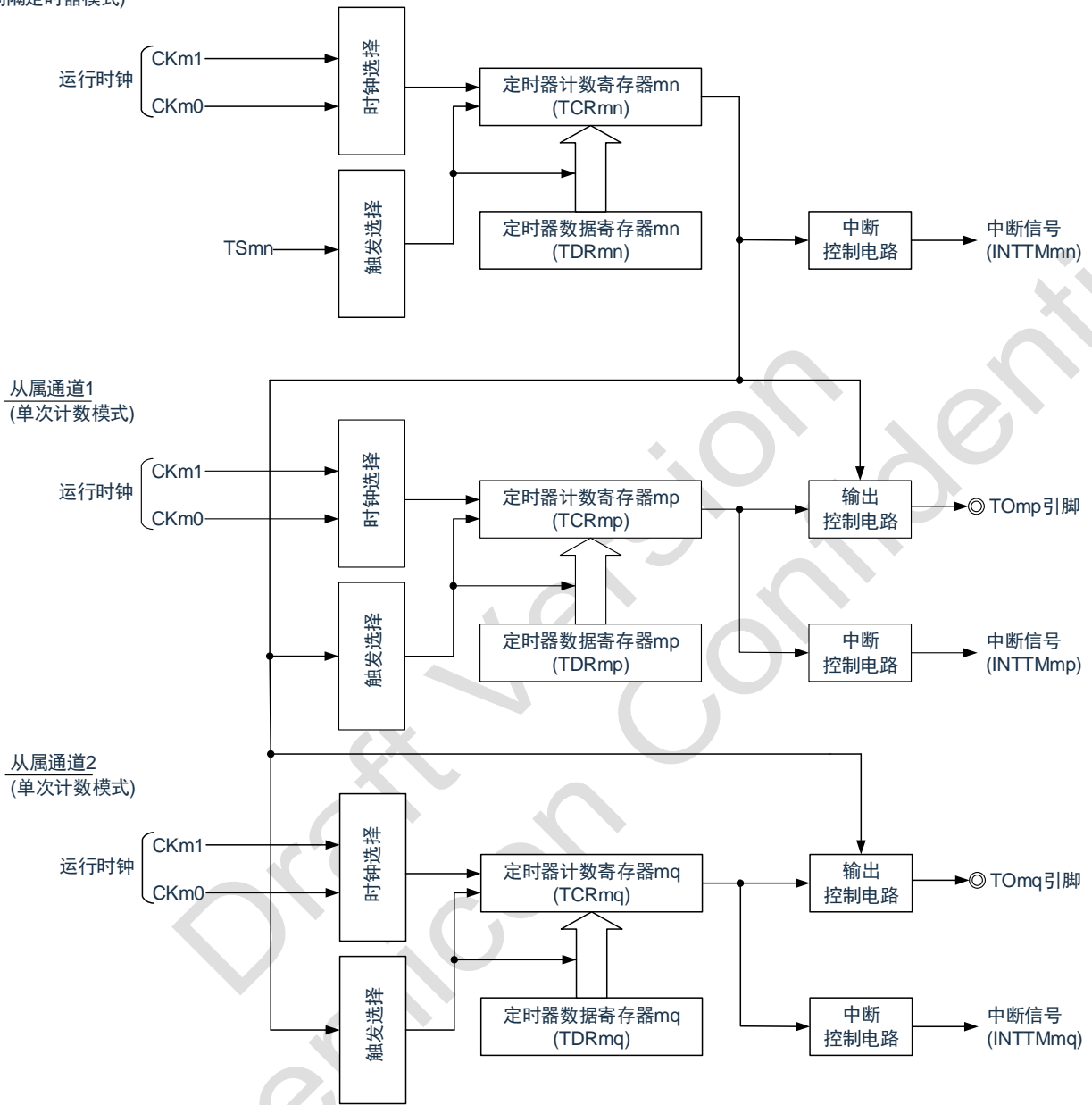
p：从属通道号

q：从属通道号  $n < p < q \leq 3$  (p和q是大于n的整数)

图6-71 作为多重PWM输出功能运行的框图(输出2种PWM的情况)

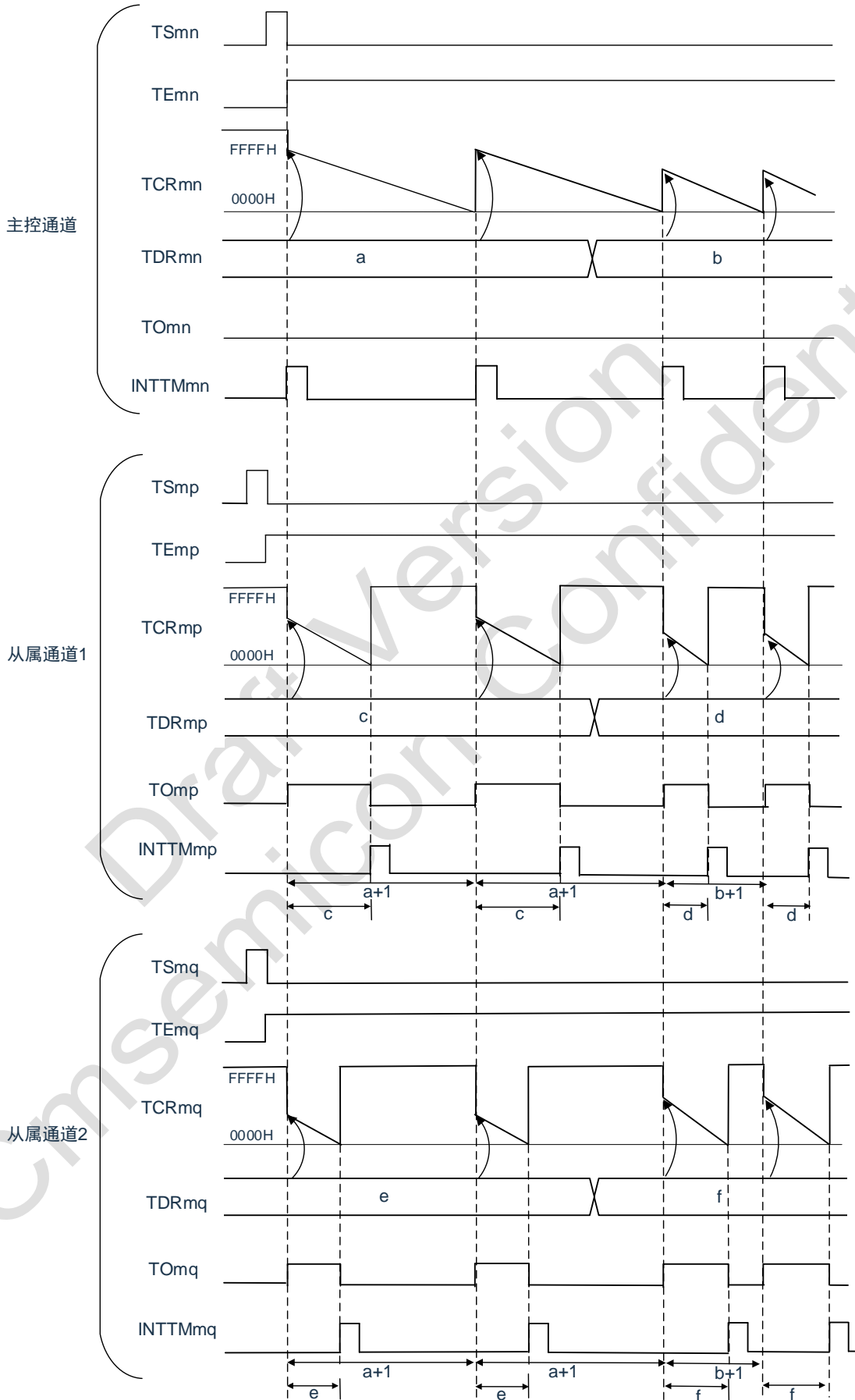
主控通道

(间隔定时器模式)



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6-72 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)



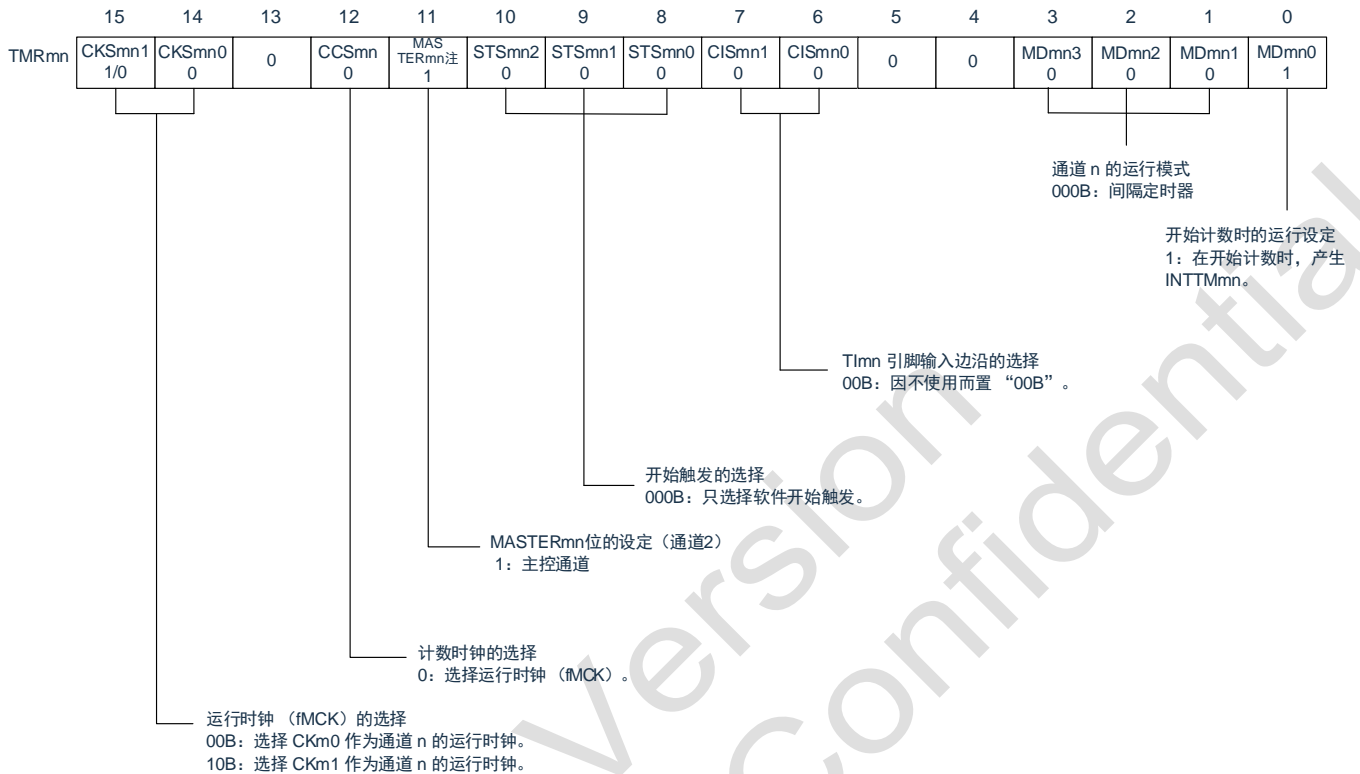
备注:

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  
 $n < p < q \leq 3$ (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

Draft Version  
Cmsemicon Confidential

图6-73 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)



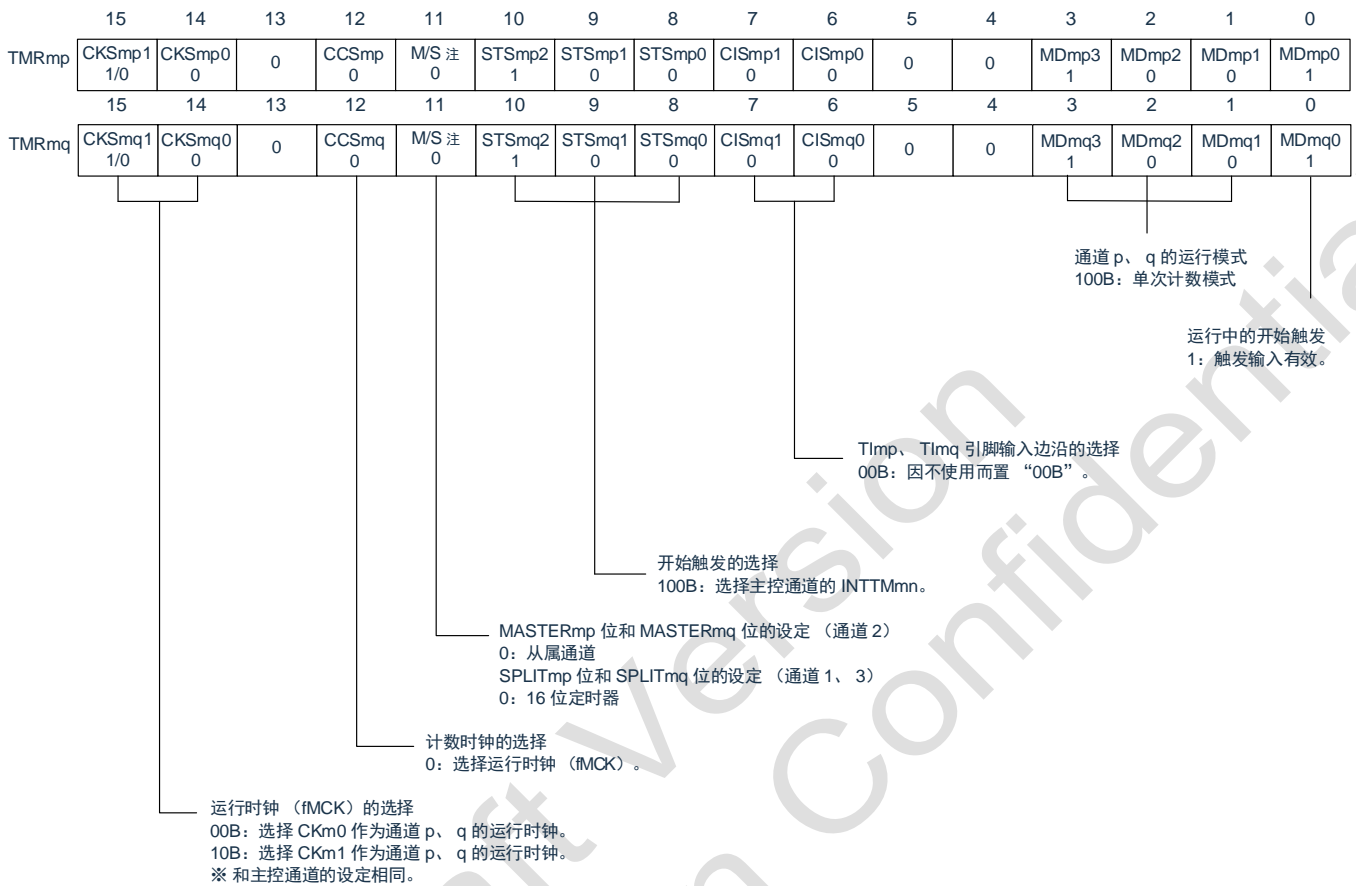
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

备注: m: 单元号(m=0) n: 主控通道号(n=0)

图6-74 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)

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

(c) 定时器输出允许寄存器m(TOEm)

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

(d) 定时器输出电平寄存器m(TOLm)

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

(e) 定时器输出模式寄存器m(TOMm)

	bit q	bit p	
TOMm	TOMLq	TOMLp	1: 设置从属通道输出模式。
	1	1	

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



$n < p < q \leq 3$  (p和q是大于n的整数)

图6-75 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”。因为TSmn 位、TSmp 位和TSmq 位是触发位,所以自动返回到“0”。</p>	<p>TEmn 位、TEmp 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。</p>
	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。</p> <p>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。</p> <p>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、TEmp 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>
		<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>	<p>TOmp 引脚和TOMq 引脚输出TOmp 和TOMq 设定的电平。</p>
	Timer4 停止	<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持的值后将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>	<p>通过端口功能保持TOmp 引脚和TOMq 引脚的输出电平。</p>
	<p>将PERO 寄存器的TM4mEN 位置“0”。</p>	<p>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  
 $n < p < q \leq 3$  (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定”。

Draft Version  
Cmsemicon Confidential

## 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。

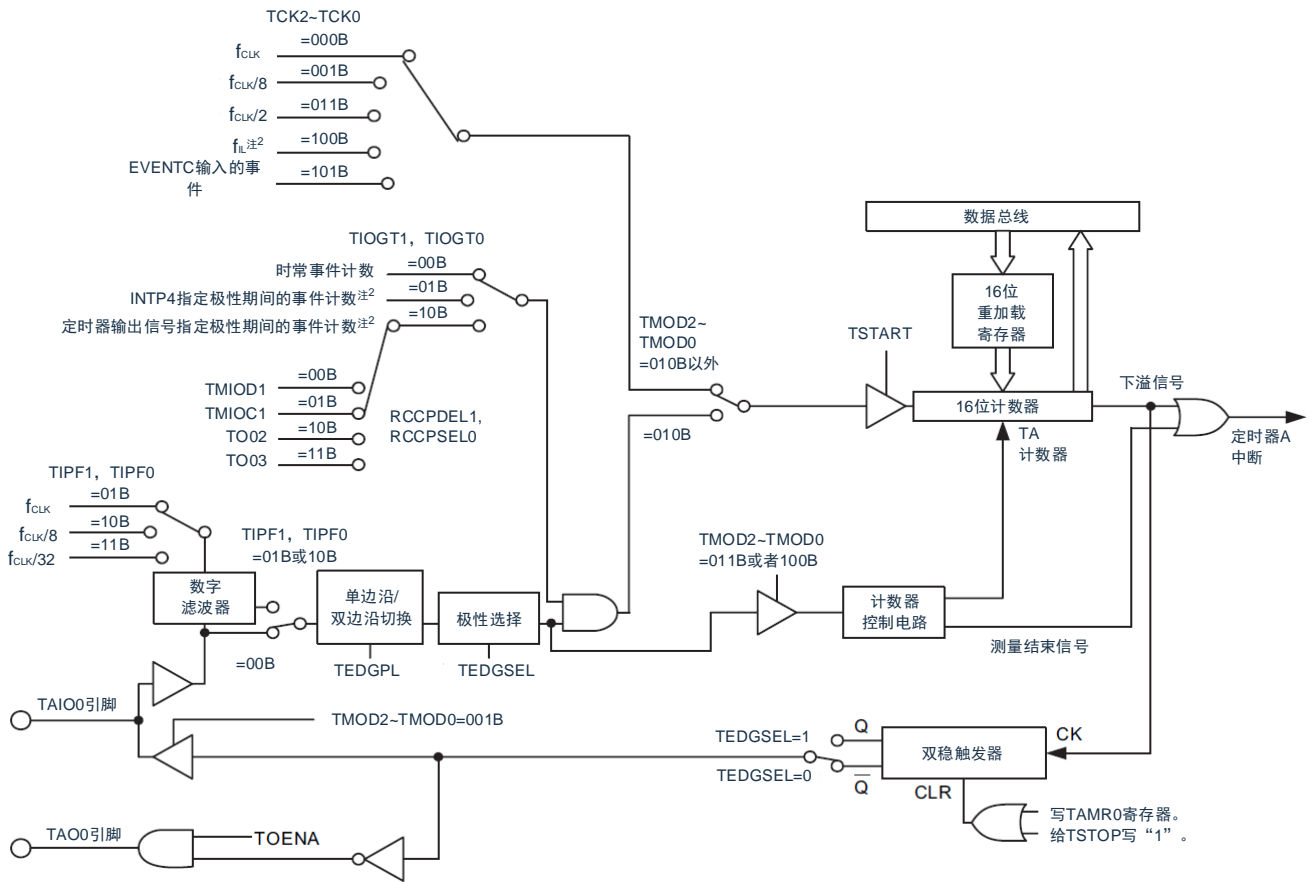
表7-1 定时器A的规格

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 $F_{CLK}$ 、 $F_{CLK}/2$ 、 $F_{CLK}/8$ 、 $F_{IL}$ 或者 EVENTC 输入的事件。
中断		<ul style="list-style-type: none"> <li>•当计数器发生下溢时</li> <li>•在脉宽测量模式中外部输入(TAIO)的有效宽度测量结束时</li> <li>•在脉冲周期测量模式中外部输入(TAIO)的设置边沿时</li> </ul>
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”。

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 <sup>注</sup>	输入/输出	定时器A的外部事件输入和脉冲输出
TAO <sup>注</sup>	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)

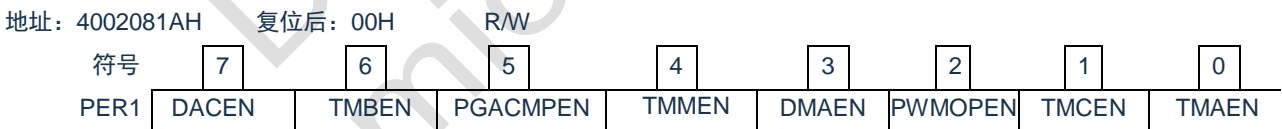
PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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



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。

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	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	蜂鸣器和定时器A的运行时钟的选择
0	<ul style="list-style-type: none"> <li>不能选择低速内部振荡器作为蜂鸣器时钟。</li> <li>不能选择低速内部振荡器作为定时器A的计数源。</li> </ul>
1	<ul style="list-style-type: none"> <li>能选择低速内部振荡器作为蜂鸣器时钟。</li> <li>能选择低速内部振荡器作为定时器A的计数源。</li> </ul>

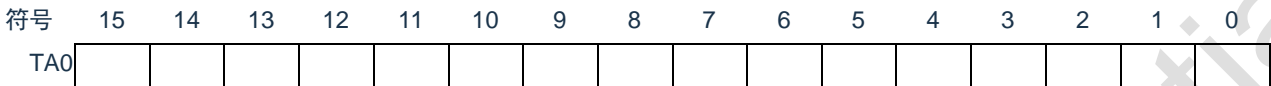
### 7.3.3 定时器A计数寄存器0(TA0)

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7-3 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 <sup>注1,2</sup>	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<sub>CLK</sub>/8)和“011B”(F<sub>CLK</sub>/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7-4 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 <sup>注1</sup>
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 <sup>注2</sup>
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 <sup>注2</sup>
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7-5 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 <sup>注1,2</sup>
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 <sub>CLK</sub> 进行采样。
1	0	有滤波器，通过F <sub>CLK</sub> /8进行采样。
1	1	有滤波器，通过F <sub>CLK</sub> /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7-6 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 <sup>注1,2</sup>
0	0	0	F <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /8
0	1	1	F <sub>CLK</sub> /2
1	0	0	F <sub>IL</sub>
1	0	1	EVENTC输入的事件
上述以外			禁止设置。

TEDGPL	TAIO边沿极性的选择 <sup>注5</sup>
0	单边沿
1	双边沿

TMOD2	TMOD1	TMOD0	定时器A运行模式的选择 <sup>注3</sup>
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<sub>IL</sub>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”。

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7-5 定时器A/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7-7 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 <sup>注</sup>	RCCPSEL1 <sup>注</sup>	RCCPSEL0 <sup>注</sup>

RCCPSEL2 <sup>注</sup>	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 <sup>注</sup>	RCCPSEL0 <sup>注</sup>	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 各产品分配的寄存器及其位(1/2)”。

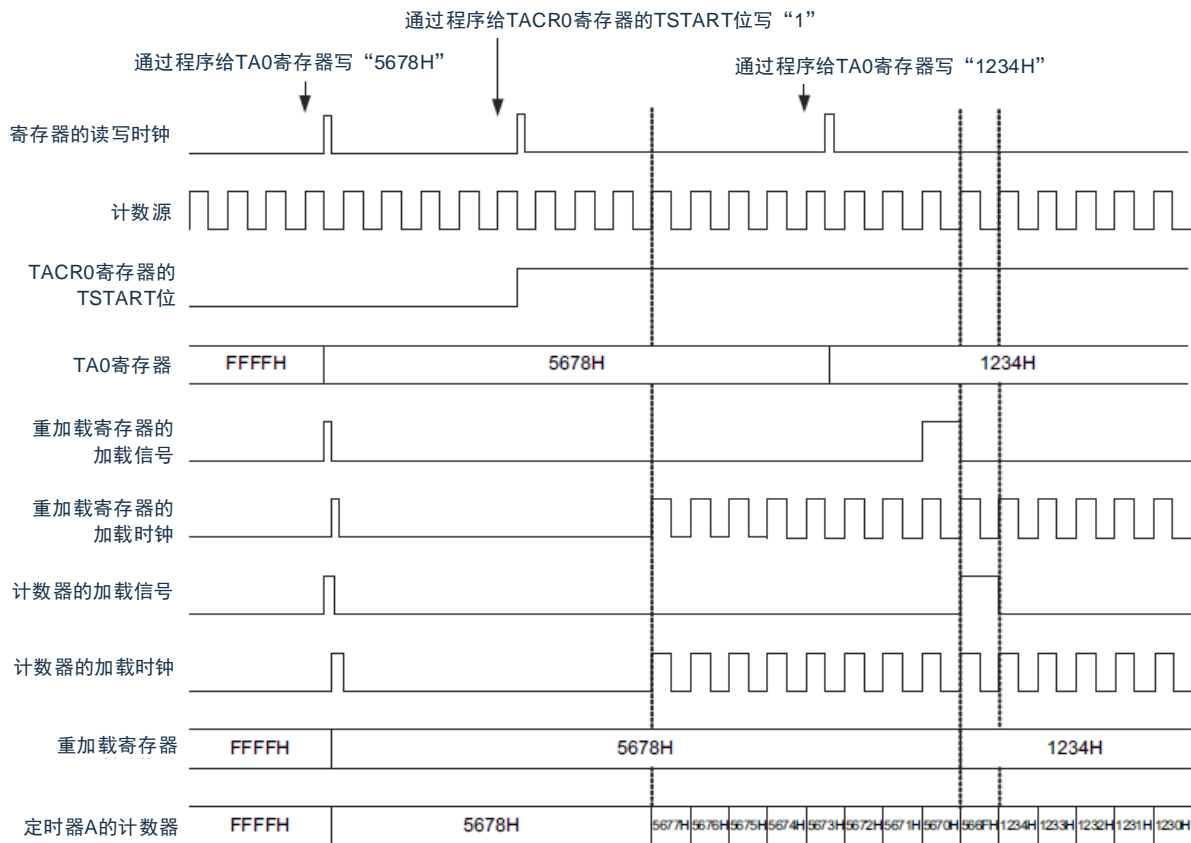
## 7.4 定时器A的运行

### 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7-8所示。

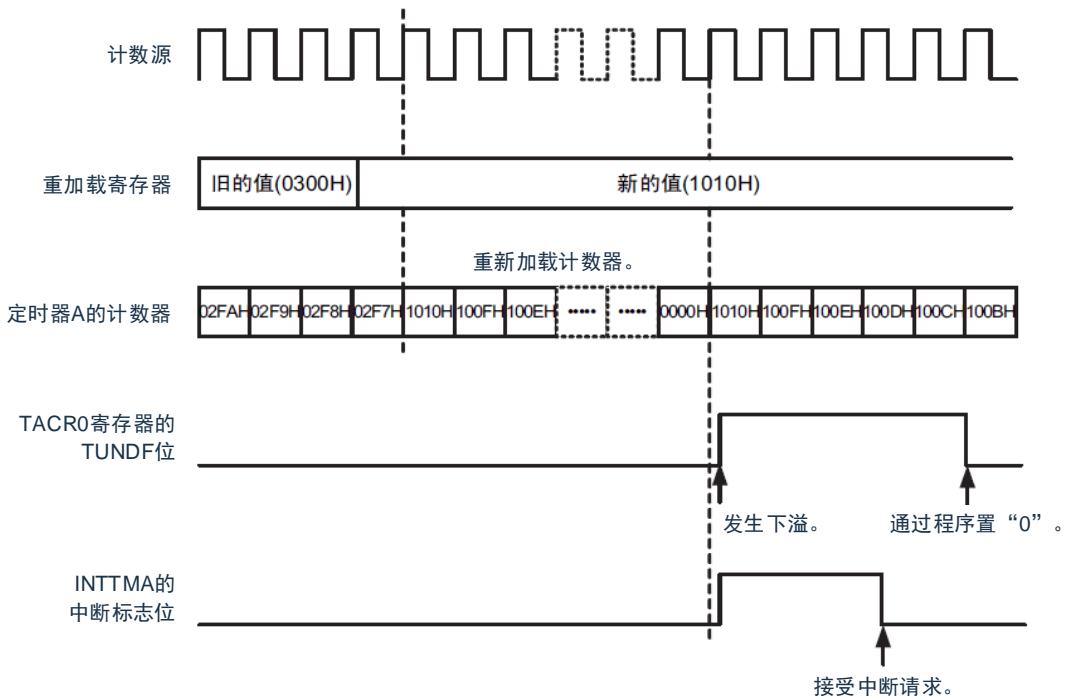
图7-8 由TSTART位的值决定的改写时序图



## 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7-9所示。

图7-9 定时器模式的运行例子





### 7.4.3 脉冲输出模式

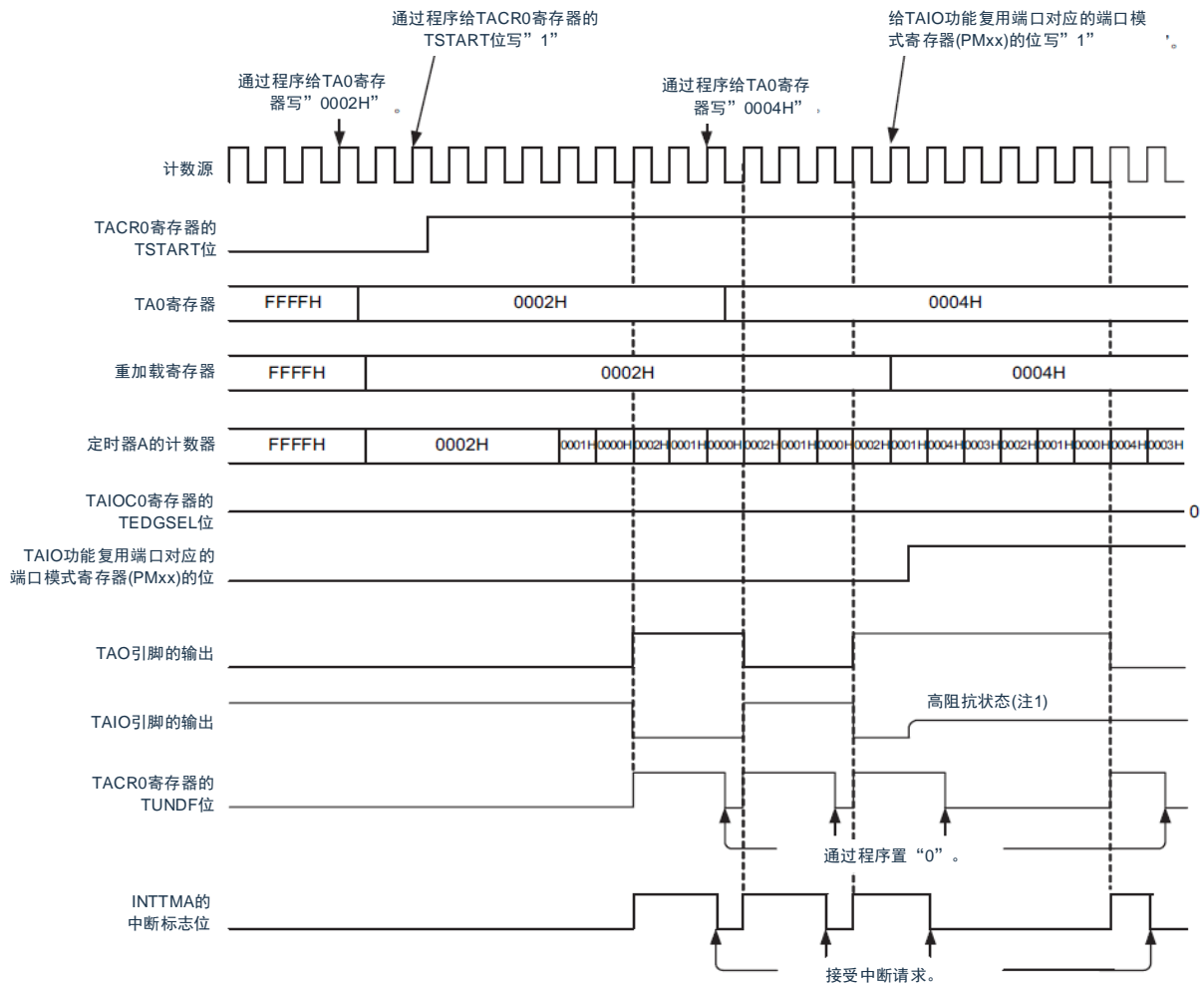
在此模式中，通过TAMR0寄存器的TCK0~TCK2位选择的计数源进行递减计数，每当发生下溢时，就将TAIO引脚和TAO引脚的输出电平反相输出。

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7-10所示。

图7-10 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。

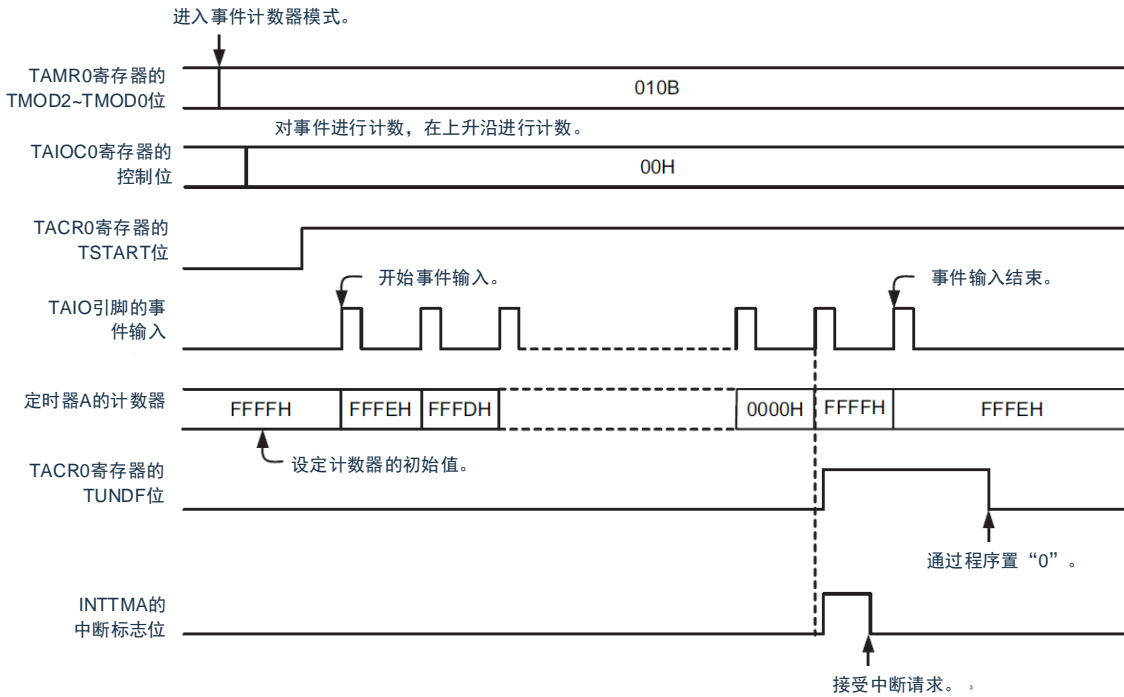
### 7.4.4 事件计数器模式

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7-11所示。

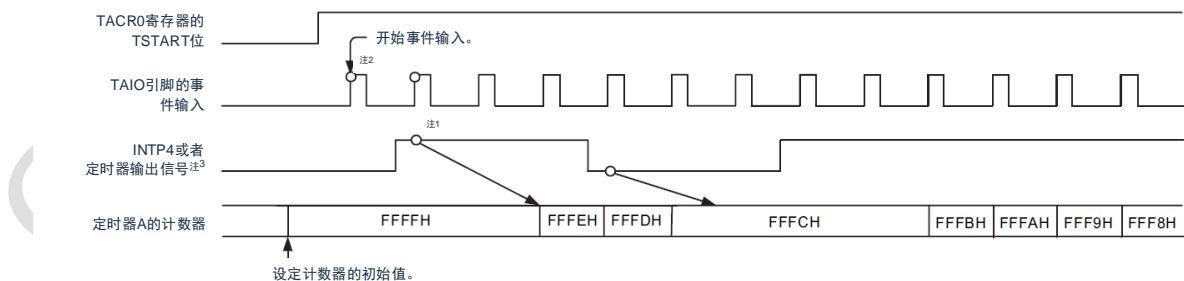
图7-11 事件计数器模式的运行例子1



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7-12所示。

图7-12 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  
 TAMRO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为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。
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TACRO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。
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。

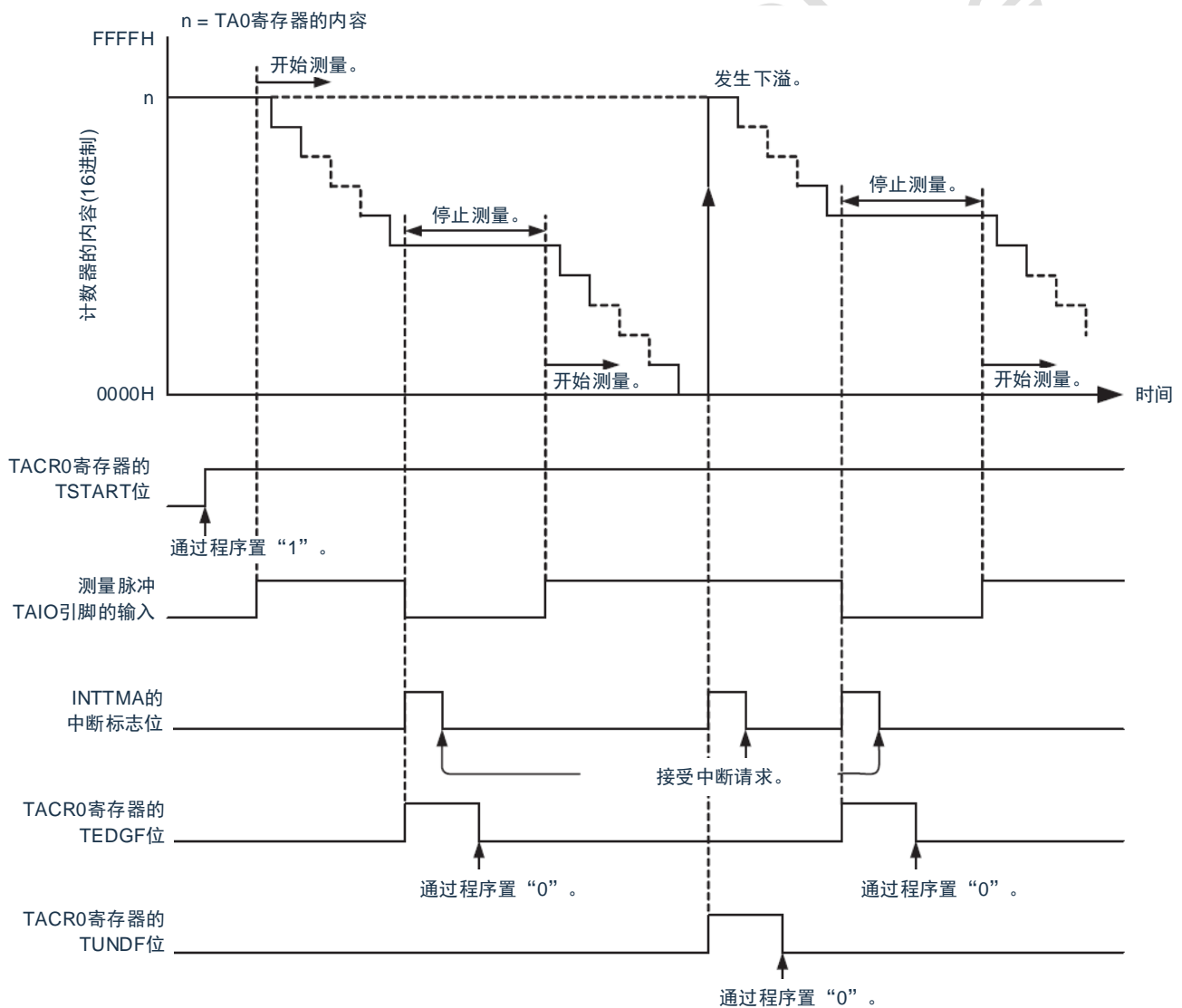
在脉宽测量模式中，如果给TAIO引脚输入TAIOC0寄存器的TEDGSEL位指定的电平，就通过选择的计数源开始递减计数。如果TAIO引脚输入的指定电平结束，计数器就停止计数，TACR0寄存器的TEDGF位变为“1”(有有效边沿)并且产生中断请求。通过在计数器停止计数时读计数值进行脉宽数据的测量。如果在测量过程中计数器发生下溢，TACR0寄存器的TUNDF位就变为“1”(发生下溢)并且产生中断请求。

脉宽测量模式的运行例子如图7-13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7-13 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)。



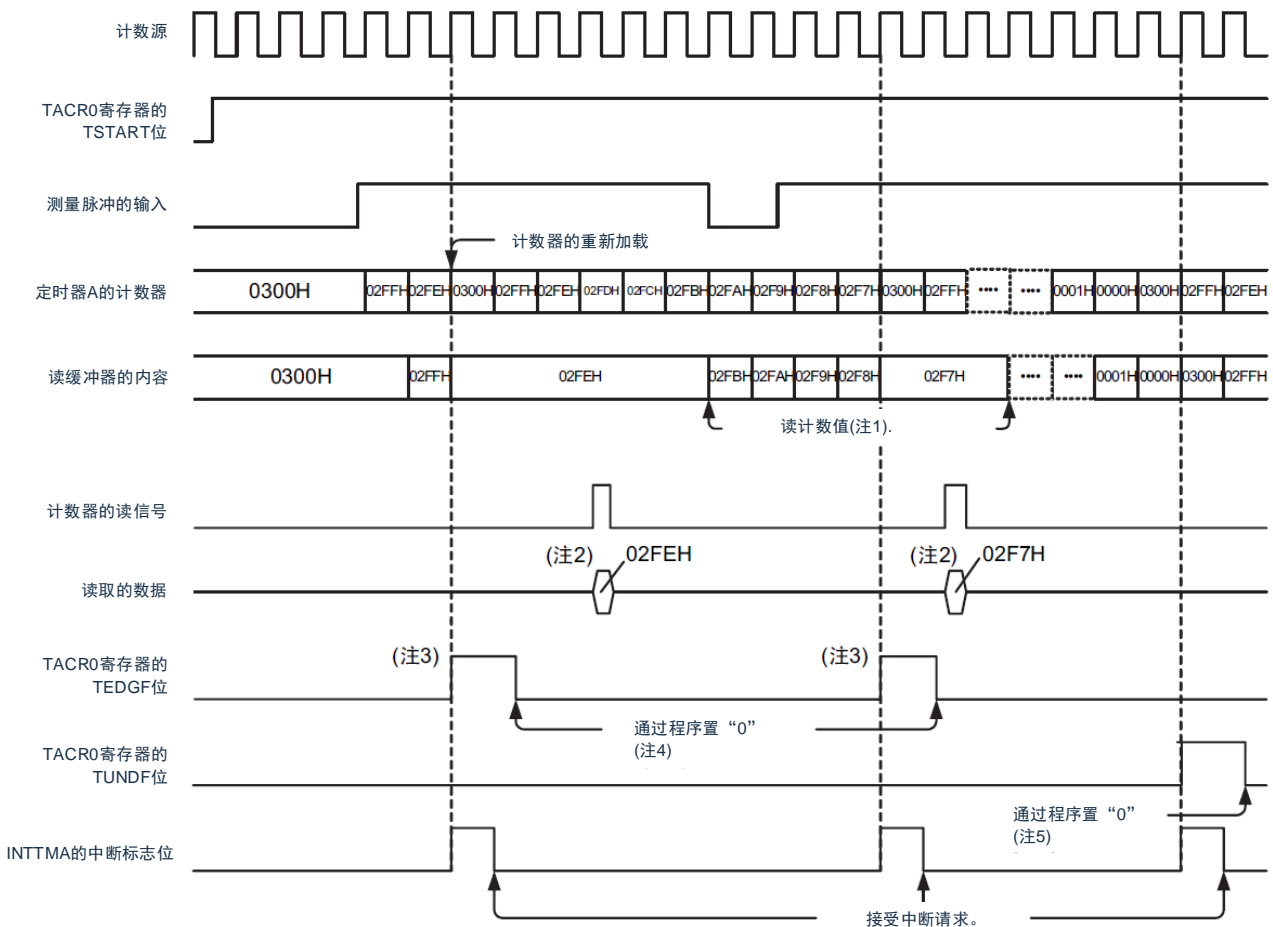
### 7.4.6 脉冲周期测量模式

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TA0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7-14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7-14 脉冲周期测量模式的运行例子



这是TA0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. 必须在从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TA0寄存器。读缓冲器的内容被保持到读TA0寄存器为止，因此，如果在输入有效边沿前不读TA0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TA0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	1	0或者1	禁止输出(Hi-Z输出)
		0	1
		0	反相输出
事件计数器模式	1	0或者1	输入
脉宽测量模式			
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。TA0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<sub>IL</sub>作为计数源的情况

要选择F<sub>IL</sub>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”。



## 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。

Draft Version  
Cmsemicon Confidential

## 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

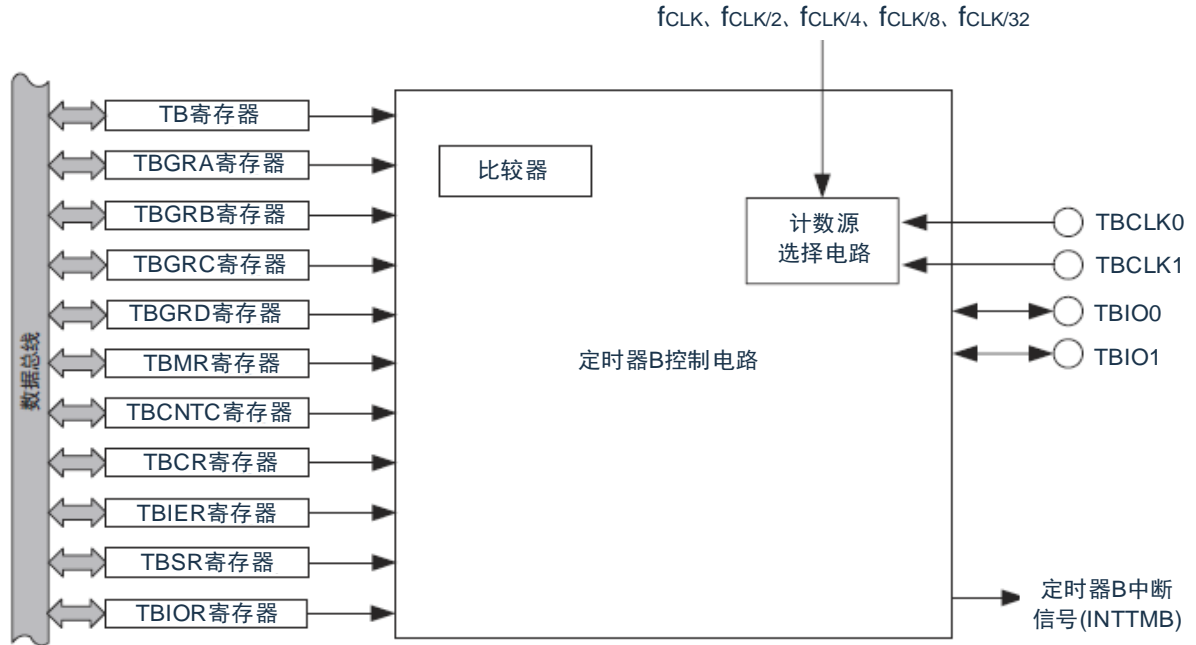


表8-1 定时器B的引脚结构

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

## 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BI/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	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 <sup>注1,2</sup>							
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 <sup>注1</sup>						
0	0	F <sub>CLK</sub> /32						
0	1	F <sub>CLK</sub> /8						
1	0	F <sub>CLK</sub>						
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TBIO12位置“1”, 并且将TBIO11位和TBIO10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 <sup>注1,2</sup>
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 <sup>注1</sup>
0	0	0	F <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /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 <sup>注1</sup>
[为“0”的条件]读后写“0” <sup>注2</sup> 。[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” <sup>注2</sup> 。[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” <sup>注2,3</sup> 。[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” <sup>注2,3</sup> 。[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)。

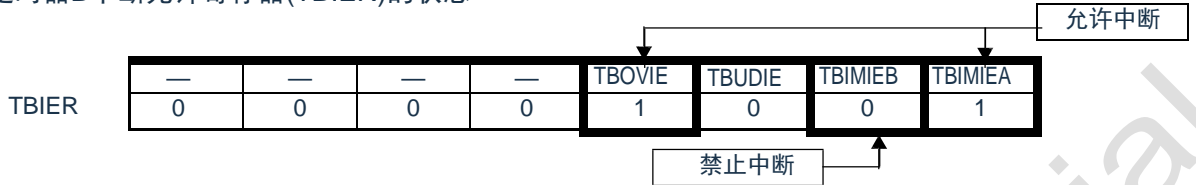
在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

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



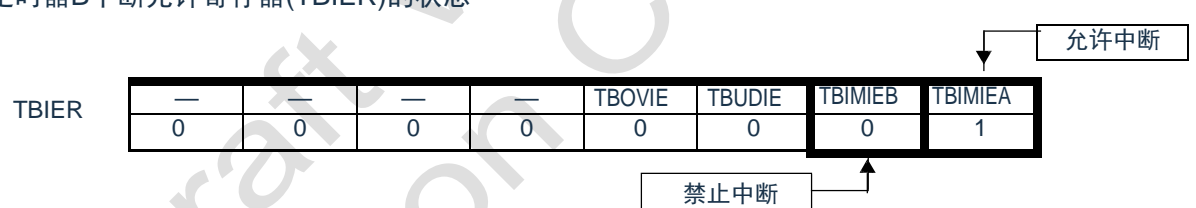
- 定时器B状态寄存器(TBSR)的状态



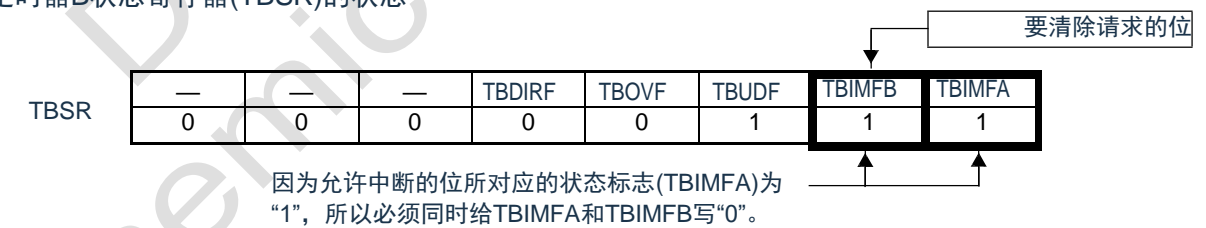
- (c) 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(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 <sup>注1</sup>		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 <sup>注2</sup>	当TB和TBGRB的值相同时	
TBIMFA	TBIO0引脚的输入边沿 <sup>注2</sup>	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 <sup>注1,2</sup>
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	TBGRC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 <sup>注1,2</sup>
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

地址: 40042656H	复位后: 0000H	R/W																
符号	15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0																	
TB	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%; height: 20px;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> </tr> </table>																	

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

—	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)	TBGRA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<sub>CLK</sub>)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端口输入模式寄存器 (PIMxx)”。

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定”。



## 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

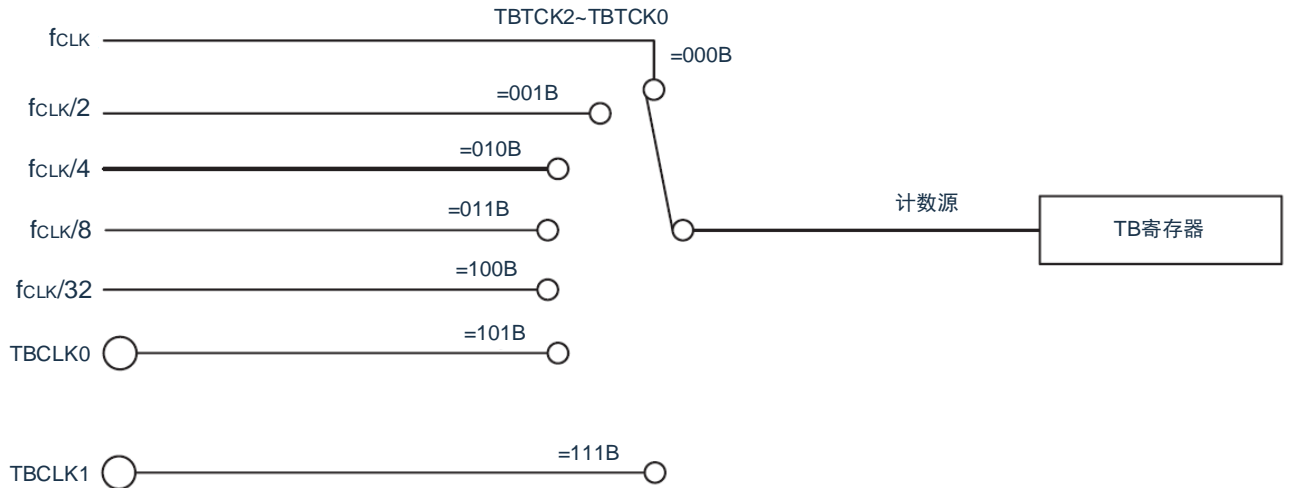
计数源的选择和框图分别如表8-5和图8-11所示。

在选择相位计数模式时，TBCR寄存器的TBTK0~TBTK2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TBTK0~TBTK2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TBTK2~TBTK0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 TBTK2~TBTK0：TBCR寄存器的位

TBCLK<sub>j</sub>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。

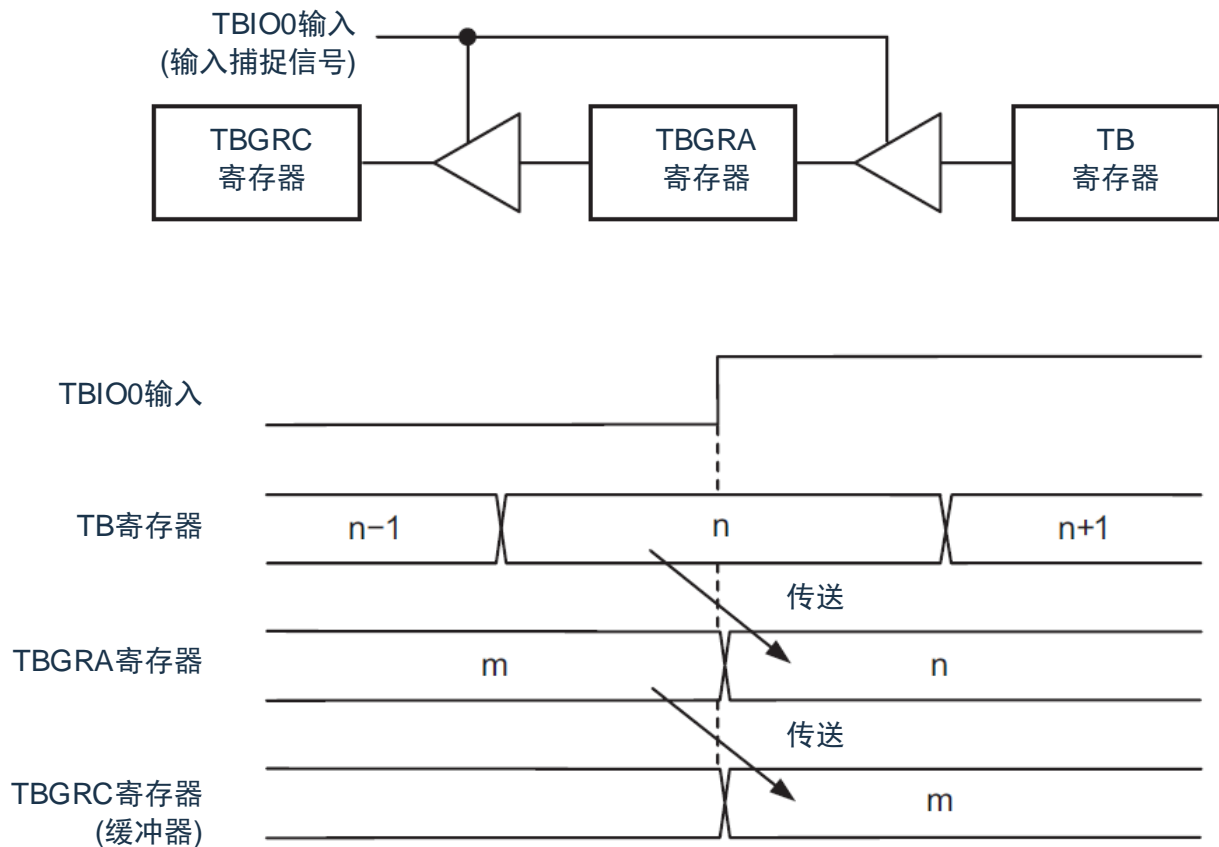
- TBGRA的缓冲寄存器：TBGRC寄存器
- TBGRB的缓冲寄存器：TBGRD寄存器缓冲器运行因定时器模式而不同。

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		

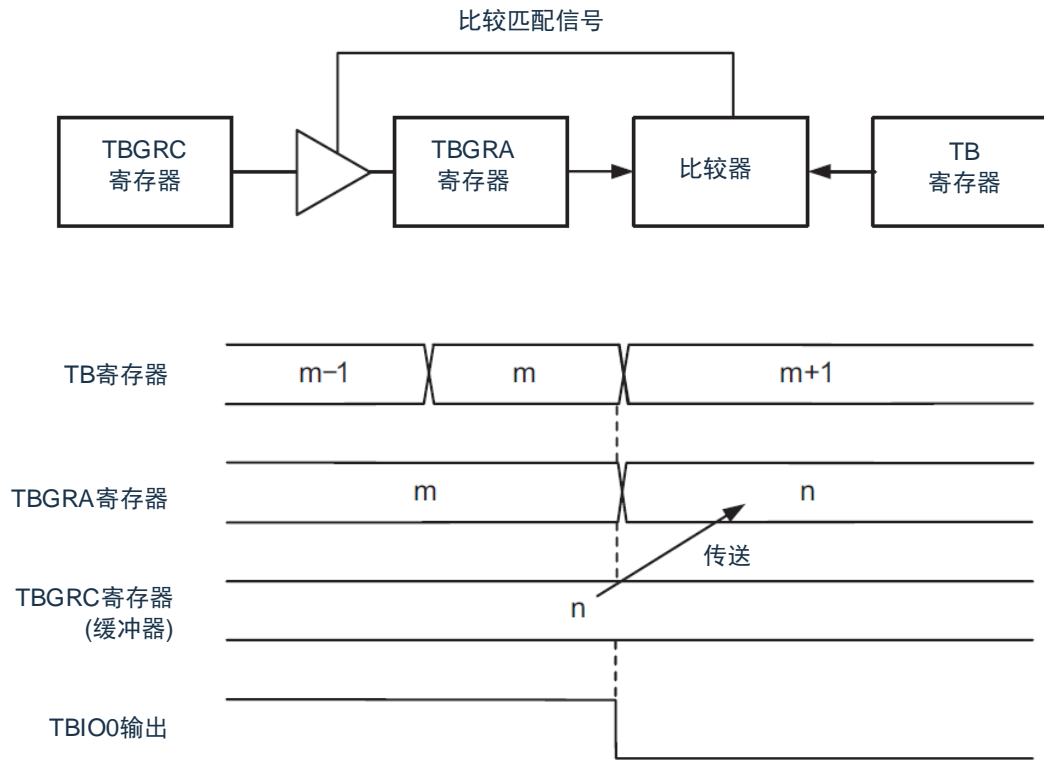
图8-12 输入捕捉功能的缓冲器运行



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:

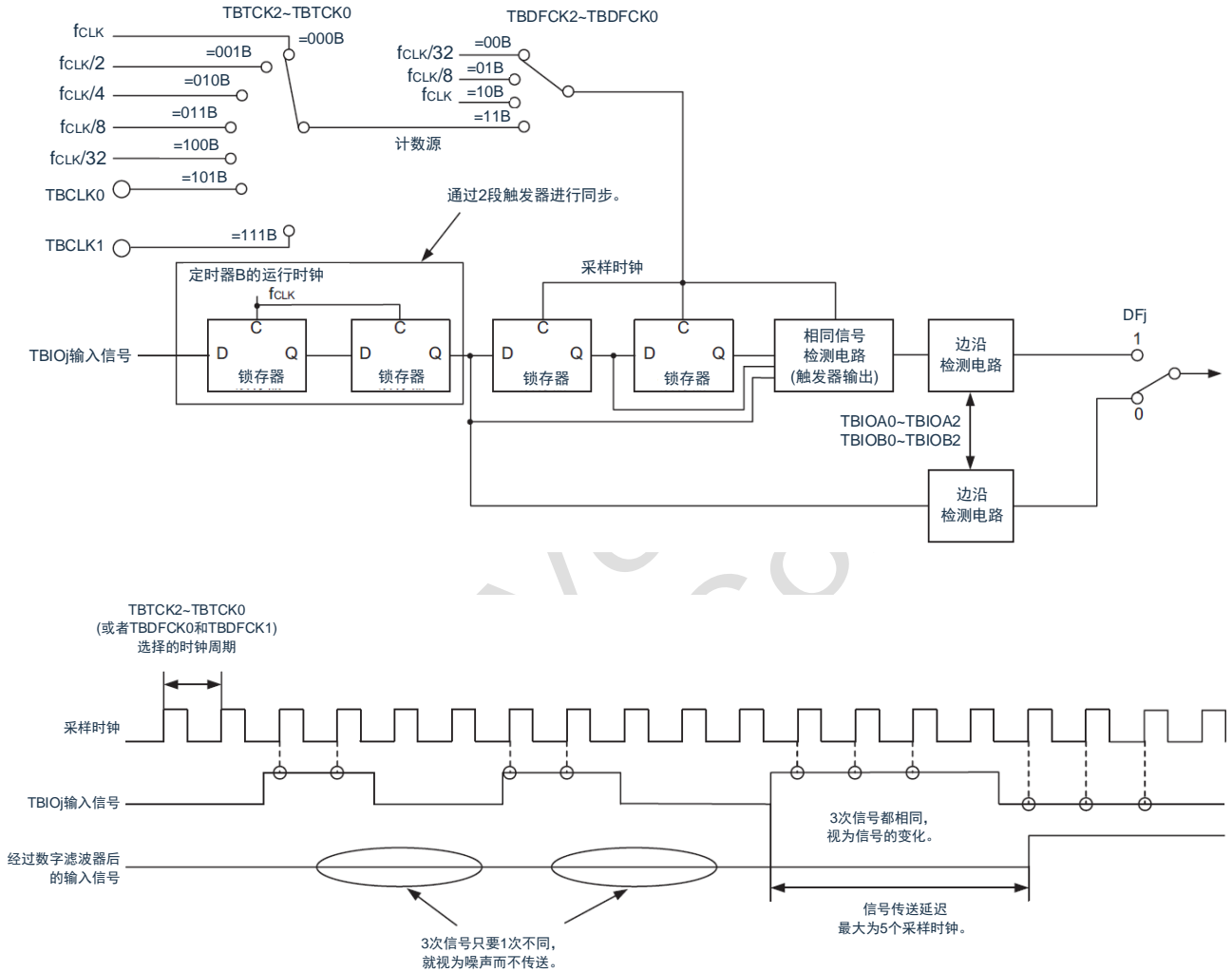
- TBIOR寄存器的TBBUFA位为“1” (TBGRC寄存器为TBGRA寄存器的缓冲寄存器)。
- TBIOR寄存器的TBIOA2~TBIOA0位为“001B” (在比较匹配时, 输出“L”电平)。

(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

TBTK2-TBTK0: TBCR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。

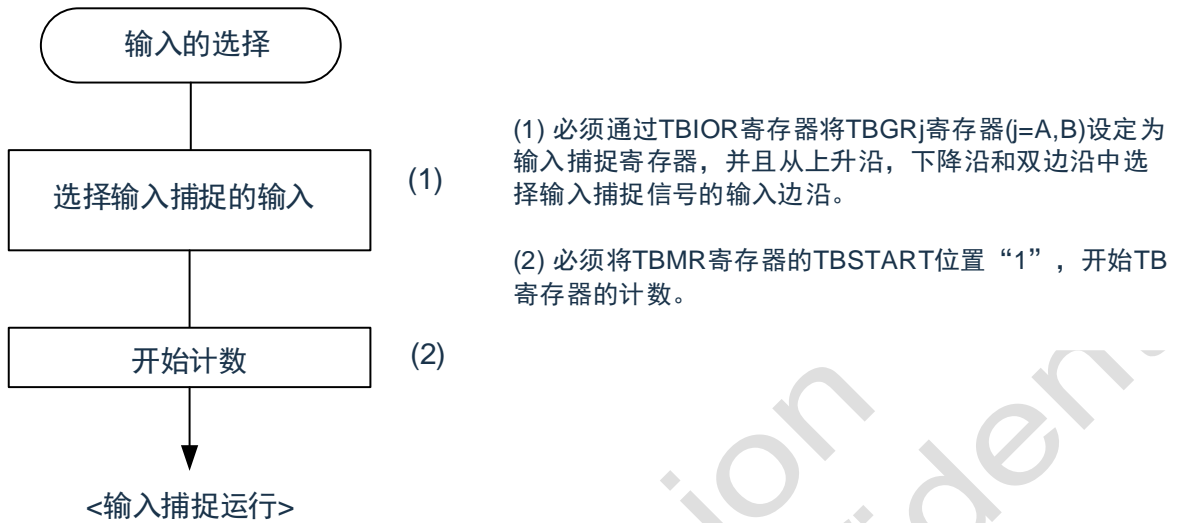
表8-9 输入捕捉功能的规格

项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /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"> <li>•输入捕捉(TBIO0引脚和TBIO1引脚输入的有效边沿)</li> <li>•TB寄存器的上溢</li> </ul>
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"> <li>•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</li> <li>•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</li> <li>•将TB寄存器置“0000H”的时序 上溢或者输入捕捉</li> <li>•缓冲器运行(参照“8.4.1(2)缓冲器运行”)</li> <li>•数字滤波器(参照“8.4.1(3)数字滤波器”)</li> <li>•通过EVENTC的事件输入信号(输入捕捉)进行的输入捕捉运行</li> </ul>

(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

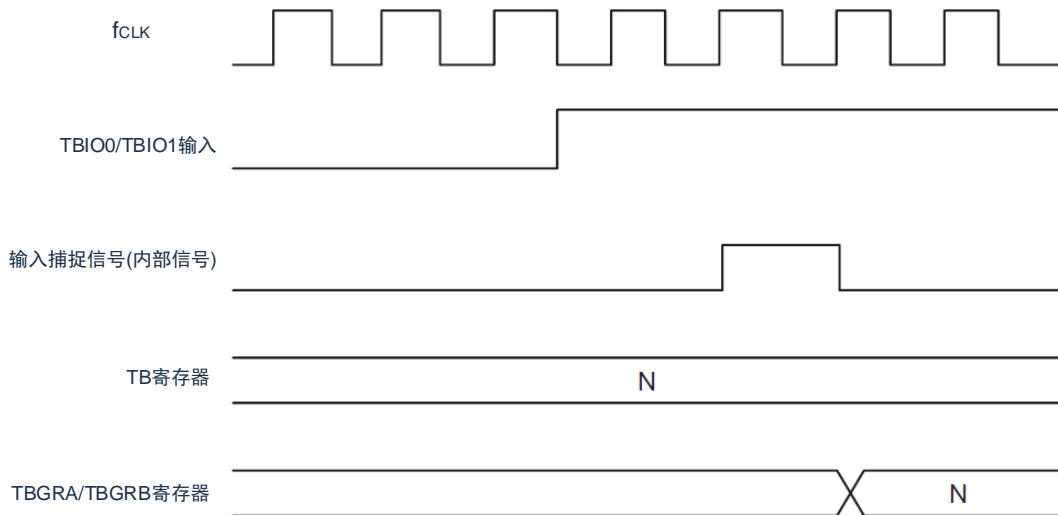


(2) 输入捕捉信号的时序

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

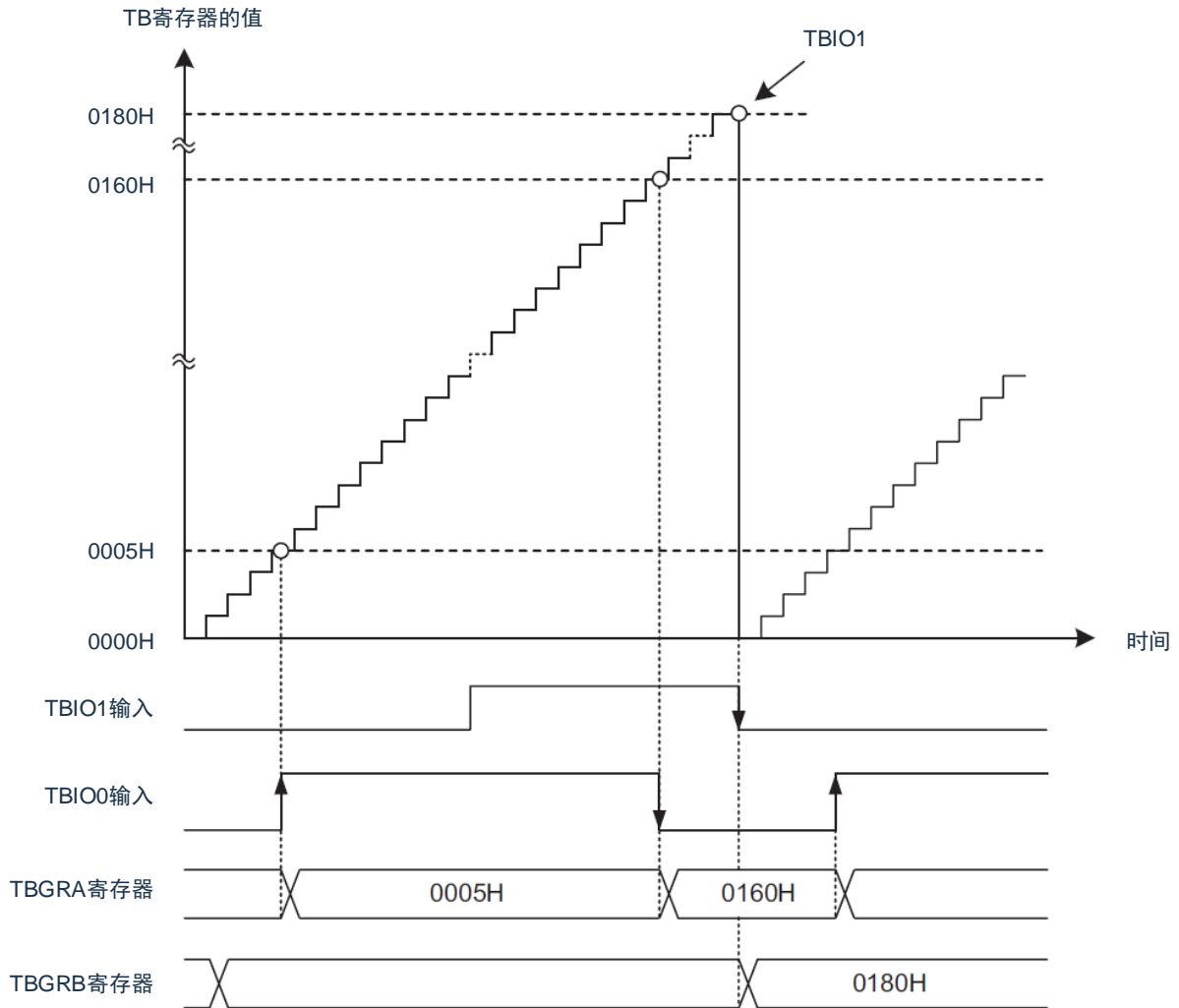
输入捕捉的运行例子如图8-17所示。

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

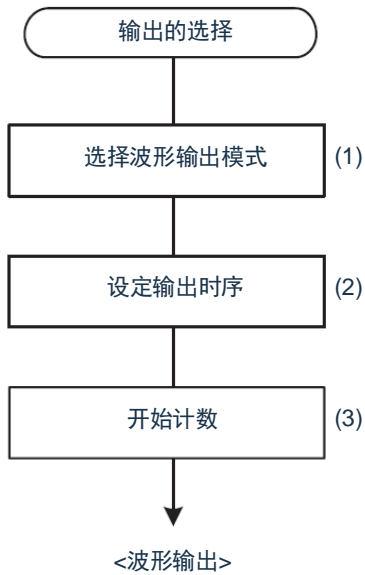
项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /32 TBCLKj引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	<ul style="list-style-type: none"> <li>当TBCR寄存器的TBCCLR1位和TBCCLR0位为“00B”(自由运行)时 1/fk 65536fk: 计数源的频率</li> <li>当TBCR寄存器的TBCCLR1位和TBCCLR0位为“01B”或者“10B”(在TBGRj比较匹配时将TB置“0000H”)时1/fk (n+1)n: TBGRj寄存器的设置值</li> </ul>
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"> <li>比较匹配(TB寄存器和TBGRj寄存器的内容相同)</li> <li>TB寄存器的上溢</li> </ul>
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"> <li>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</li> <li>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</li> <li>将TB寄存器置“0000H”的时序 上溢或者TBGRj寄存器的比较匹配</li> <li>缓冲器运行(参照“8.4.1(2)缓冲器运行”)</li> </ul>

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



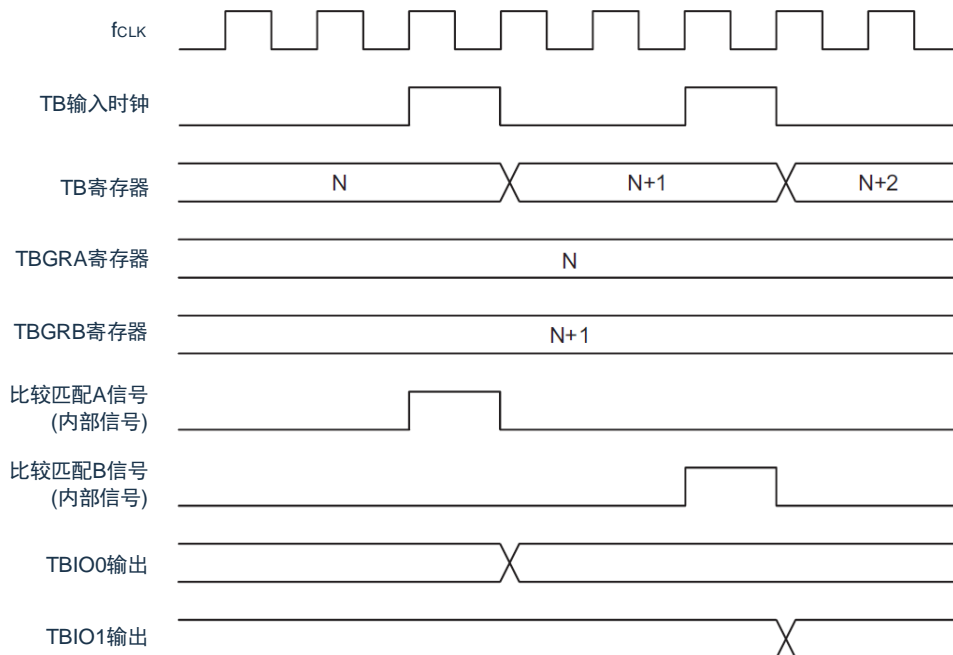
- (1) 必须通过TBIOR寄存器从“L”电平输出，“H”电平输出和交替输出中选择比较匹配输出。如果设定为波形输出模式，端口就为比较匹配的输出引脚(TBIO0,TBIO1)。在发生第一次比较匹配前，比较匹配输出引脚的输出电平取决于TBIOR寄存器的TBIOA0和TBIOA1，TBIOB0和TBIOB1位的设定。
- (2) 必须给TBGRA寄存器和TBGRB寄存器设定比较匹配的发生时序。
- (3) 必须将TBMR寄存器的TBSTART位置“1”，开始TB寄存器的计数。

(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

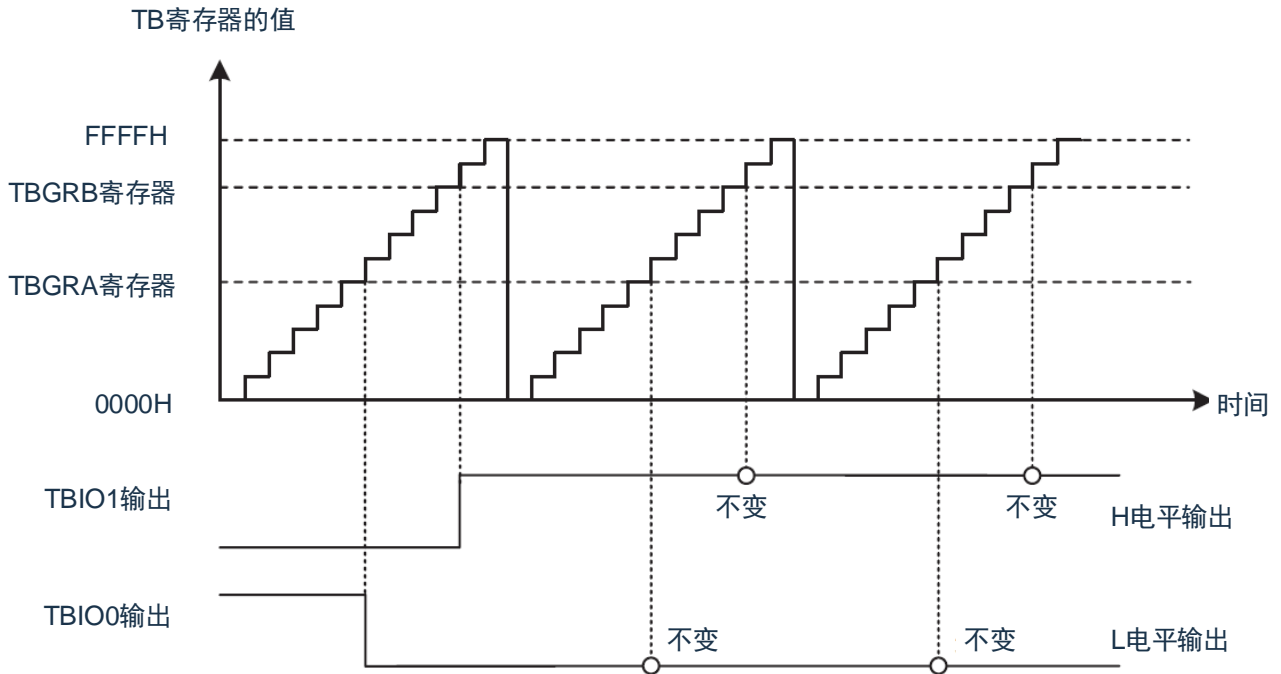


(3) 运行例子

“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

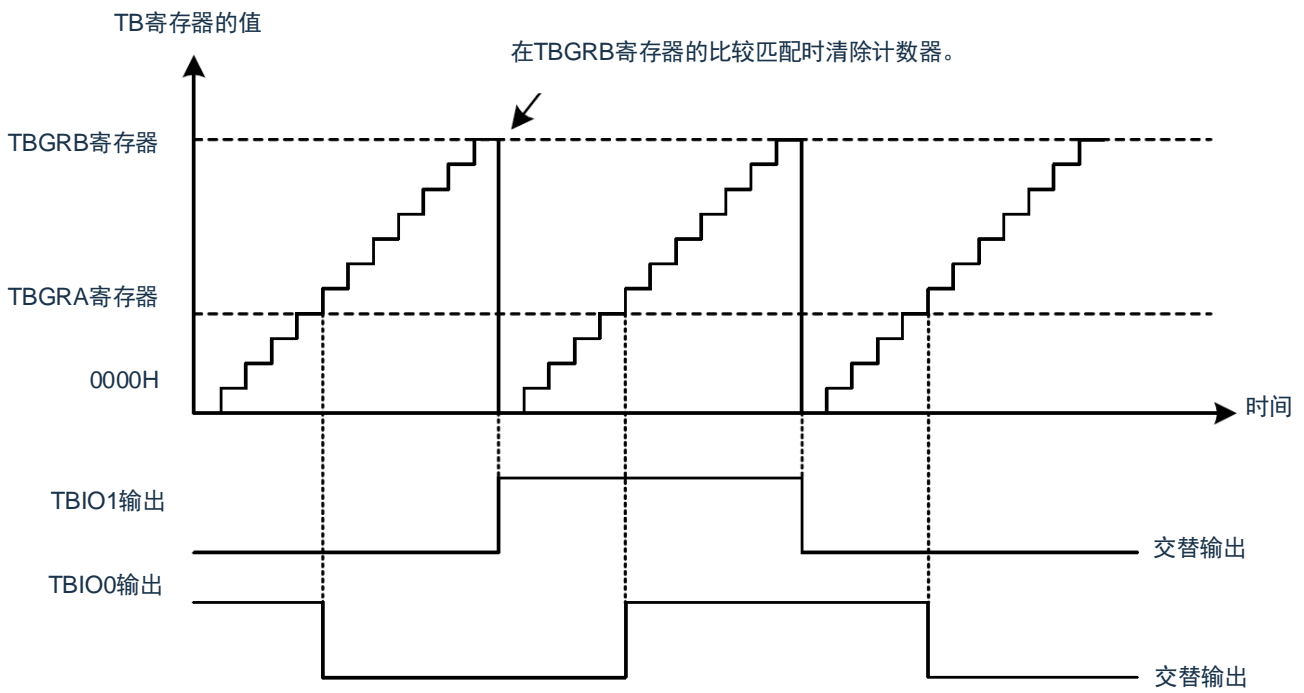


交替输出的运行例子如图8-21所示。在此例子中，将TB寄存器设置为周期计数运行(在比较匹配B时清除计数器)，并且在发生比较匹配A或者比较匹配B时都进行交替输出。

- (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TBIO00位、TBIO01位、TBIO10位和TBIO11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



Draft Version  
Cmsemicon Corporation

## 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 <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /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"> <li>给TBGRA寄存器设置PWM波形的“H”电平输出时序。</li> <li>给TBGRB寄存器设置PWM波形的“L”电平输出时序。</li> </ul>
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"> <li>比较匹配(TB寄存器和TBGRj寄存器的内容相同)</li> <li>TB寄存器的上溢</li> </ul>
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"> <li>将TB寄存器置“0000H”的时序上溢或者TBGRj寄存器的比较匹配</li> <li>缓冲器运行(参照“8.4.1(2)缓冲器运行”)</li> </ul>

备注：j=A、B

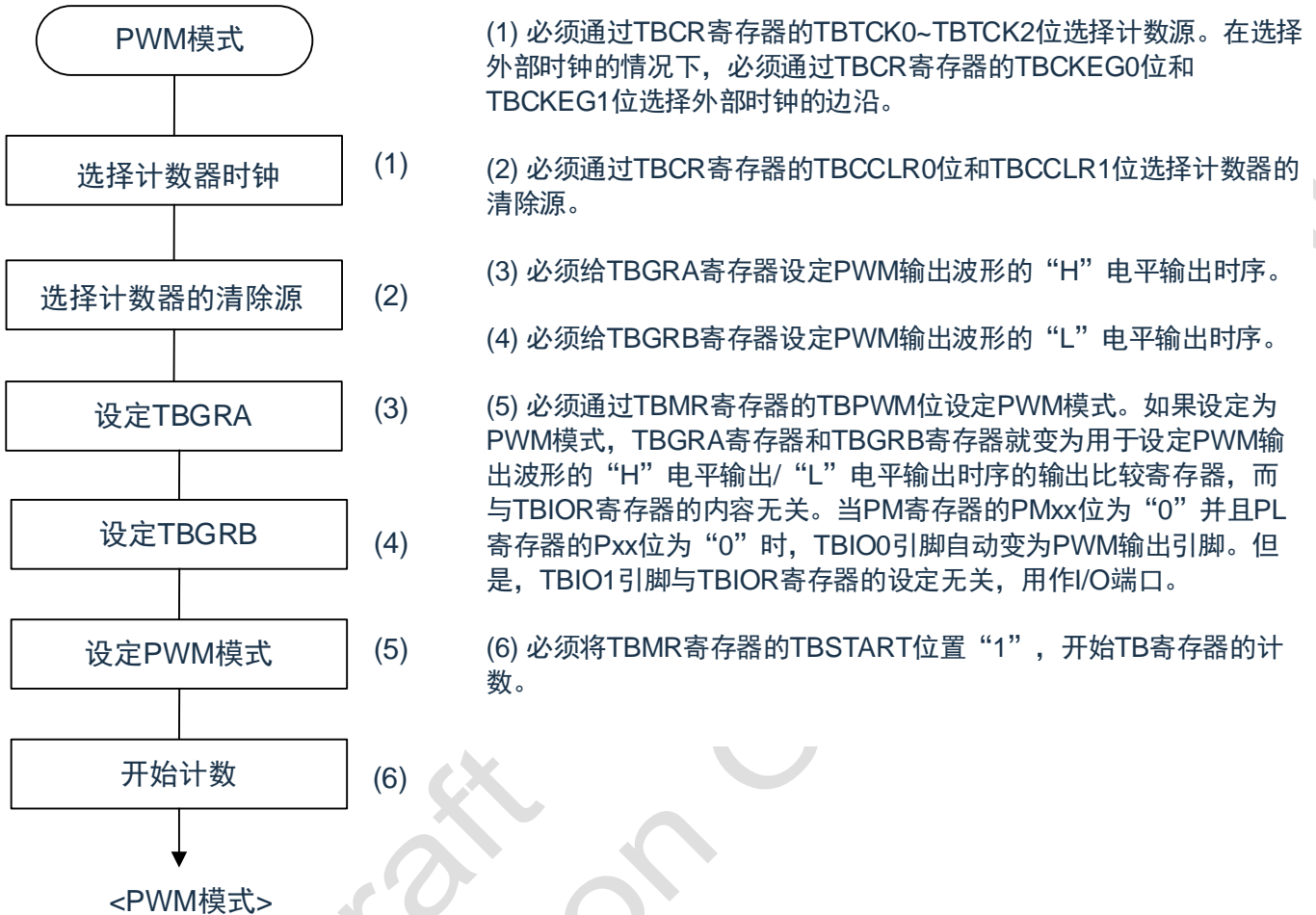
表8-12: PWM输出引脚和寄存器的组合

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，此对应关系如表8-13所示。

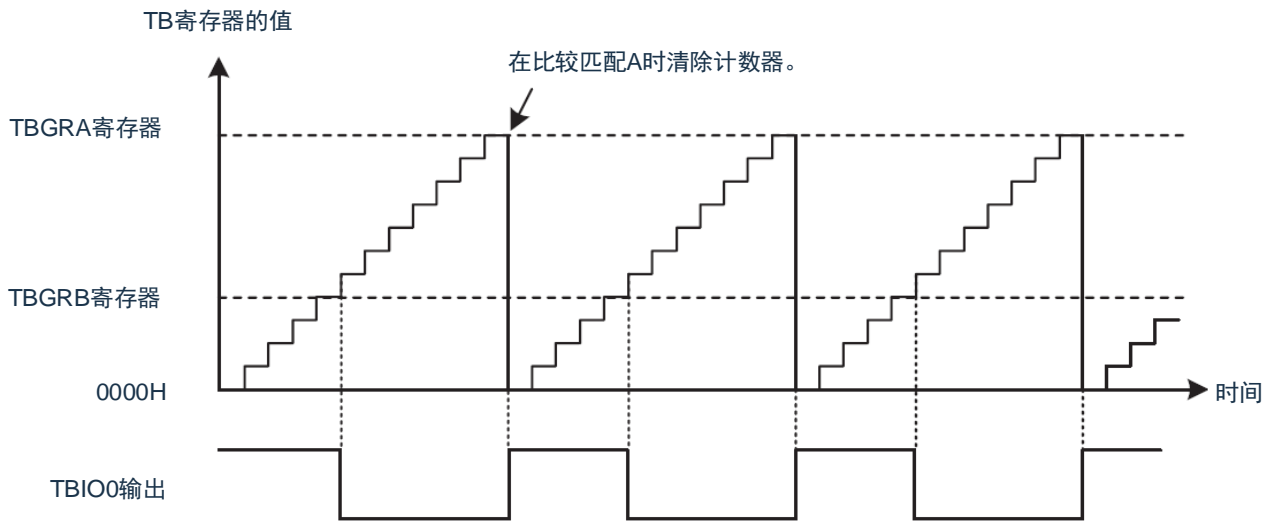
在TBMR寄存器的TBSTART位为“0”(停止计数)时进行此初始化。

表8-13 TBIO0引脚的初始状态和计数器清除源的对应关系

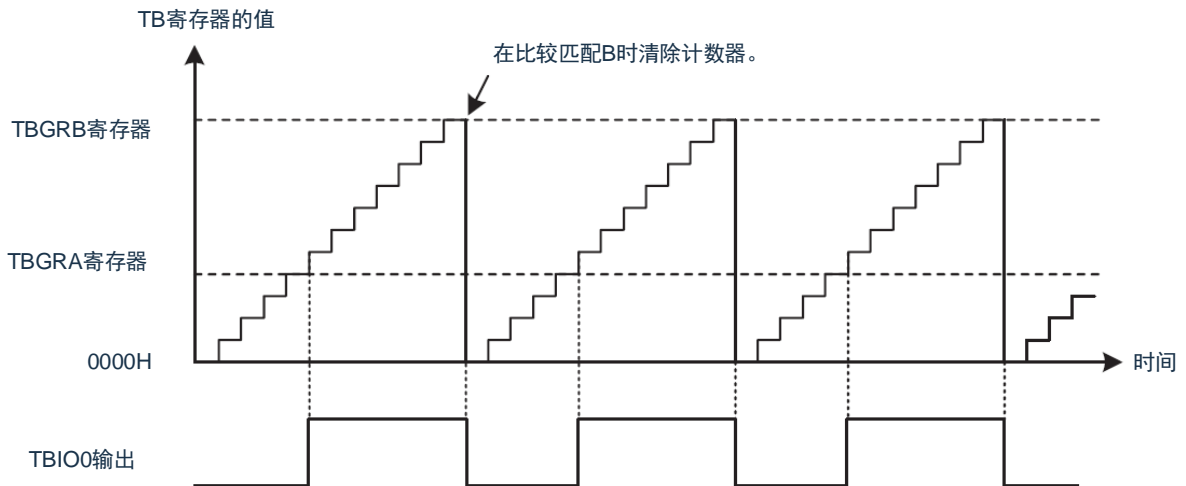
计数器清除源	TBIO0引脚的初始状态
TBGRA寄存器的比较匹配	“H”电平
TBGRB寄存器的比较匹配	“L”电平

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。

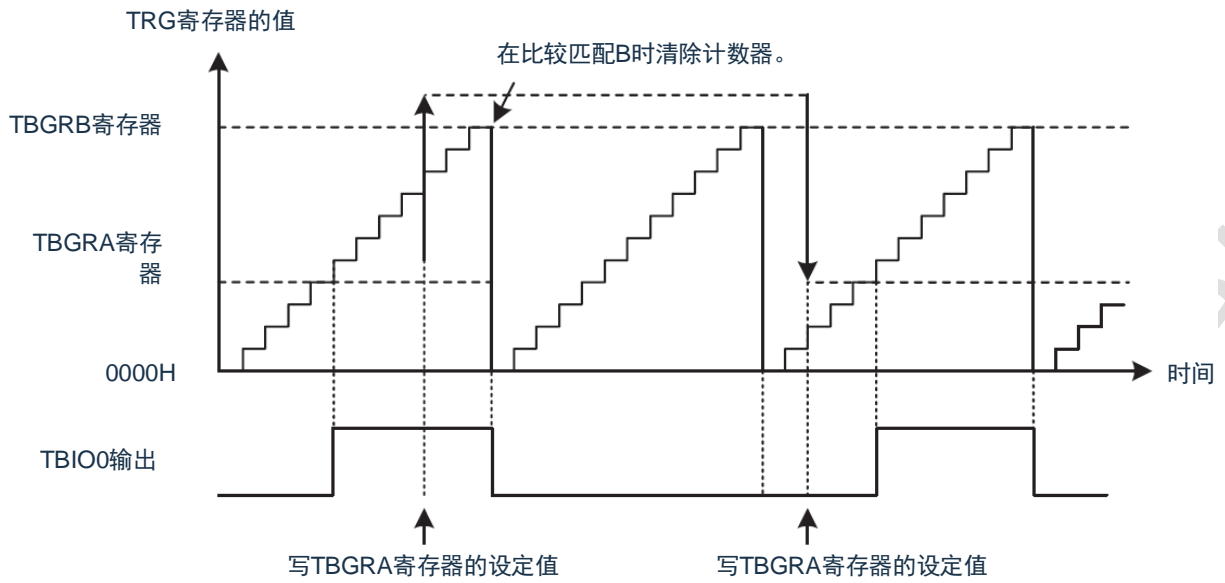
当将TBGRB寄存器的比较匹配设置为计数器的清除源并且满足以下条件时，PWM波形的占空比为0%。

- TBGRA寄存器的设置值 > TBGRB寄存器的设置值

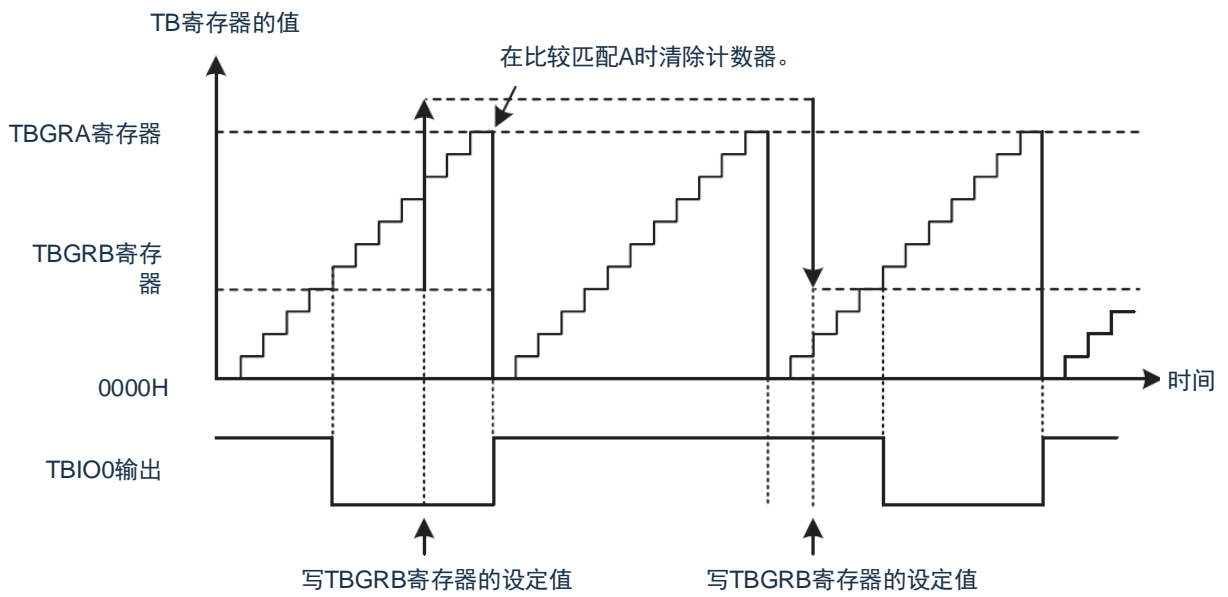
当将TBGRA寄存器的比较匹配设置为计数器的清除源并且满足以下条件时，PWM波形的占空比为100%。

- 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"> <li>•输入捕捉(TBIO0/TBIO1输入的有效边沿)</li> <li>•比较匹配(TB寄存器和TBGRA/TBGRB寄存器的内容相同)</li> <li>•TB寄存器的上溢</li> <li>•TB寄存器的下溢</li> </ul>
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"> <li>•计数器的加减条件的选择</li> <li>通过TBCNTC寄存器的CNTEN0~CNTEN7位进行选择。</li> <li>•能使用输入捕捉/输出比较功能和PWM功能。</li> </ul>

表8-15 TB寄存器的加减条件

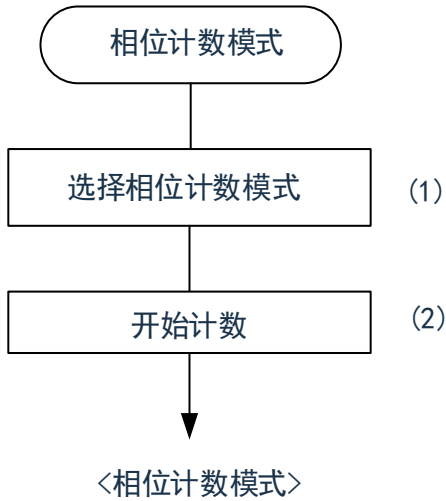
TBCLK1引脚	↑	“H”	↓	“L”	“H”	↓	“L”	↑
TBCLK0引脚	“L”	↑	“H”	↓	↓	“L”	↑	“H”
TBCNTC寄存器的CNTEN0~CNTEN7位	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0
计数方向 <sup>注</sup>	+1	+1	+1	+1	-1	-1	-1	-1

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

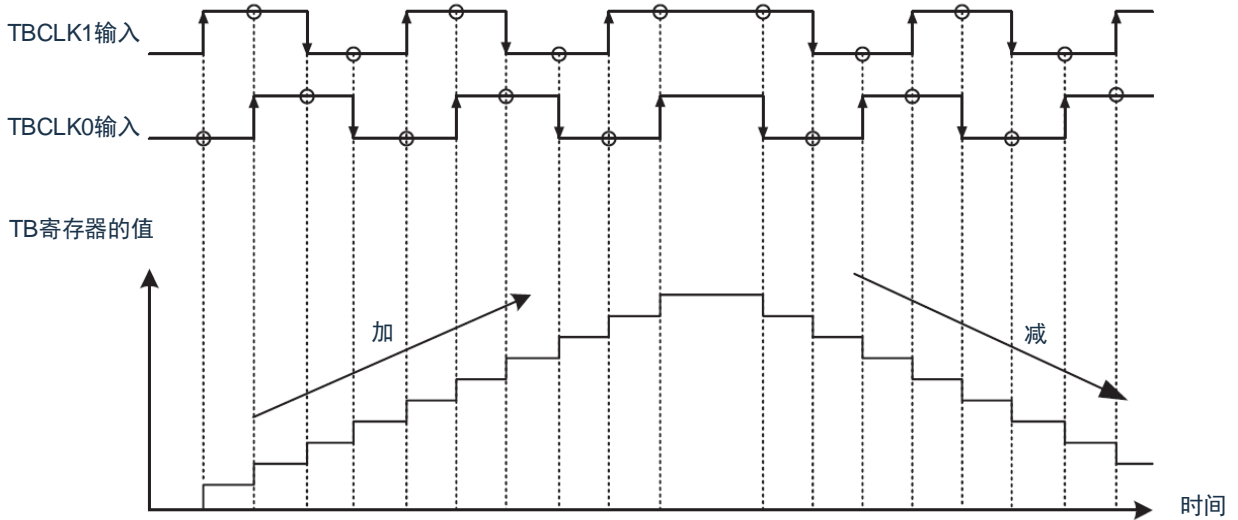


图8-27 相位计数模式的运行例子2

· TBCNTC寄存器的值为“24H”的情况

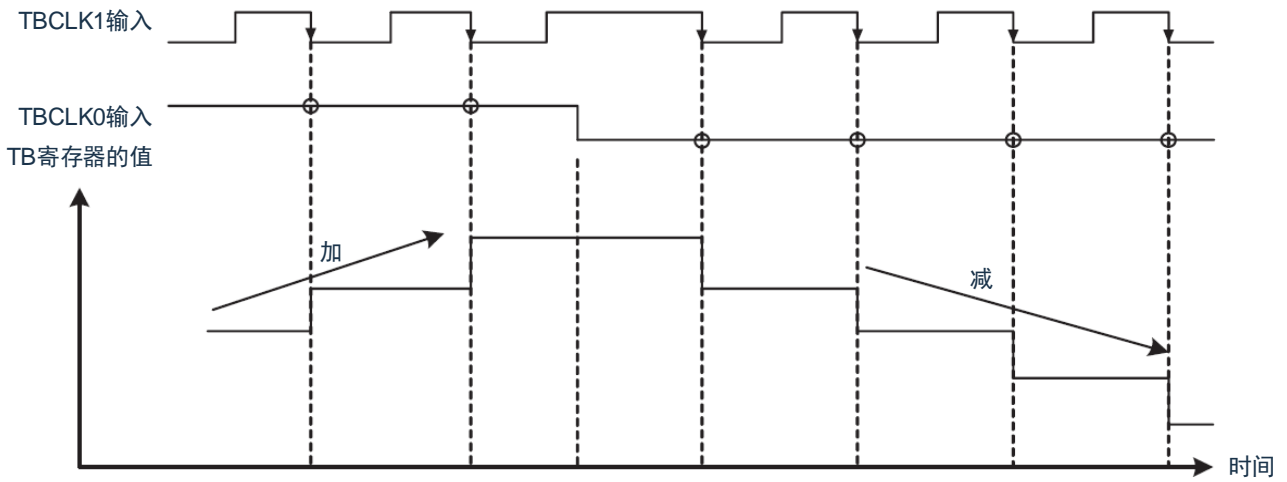


图8-28 相位计数模式的运行例子3

· TBCNTC寄存器的值为“28H”的情况

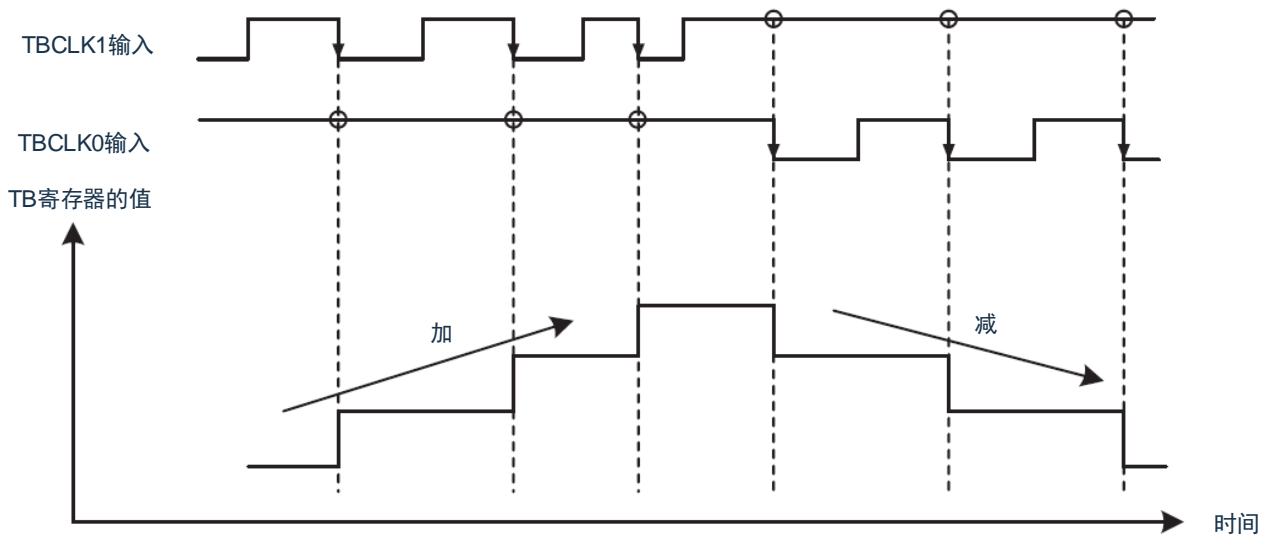
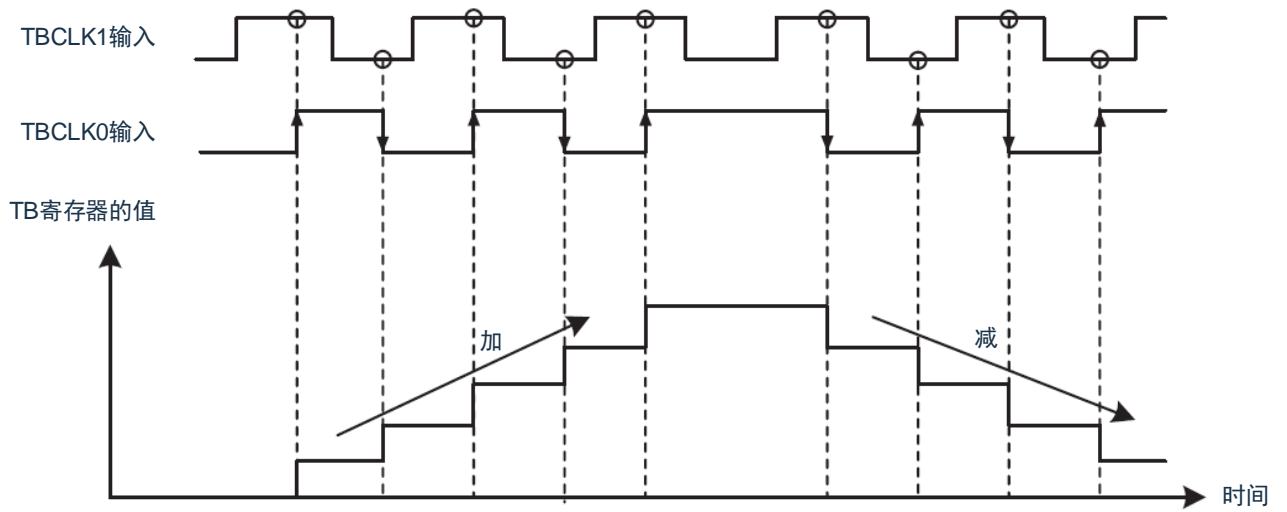


图8-29 相位计数模式的运行例子4

· TBCNTC寄存器的值为“5AH”的情况



Draft Version  
Cmsemicon Confidential

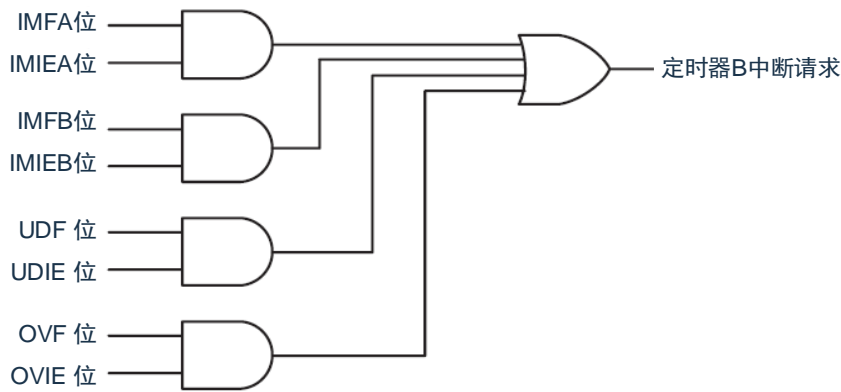
## 8.5 定时器B中断

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



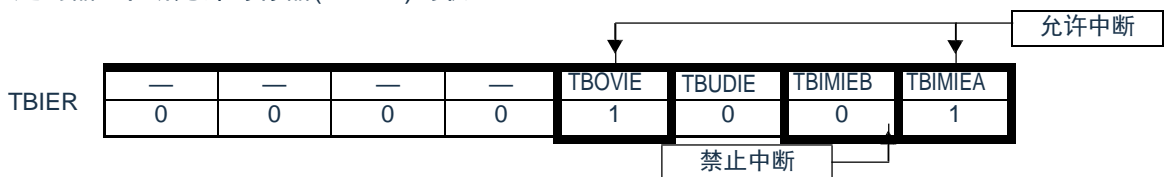
IMFA、IMFB、UDF、OVF: TBSR寄存器的位  
IMIEA、IMIEB、UDIE、OVIE: TBIER寄存器的位

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:

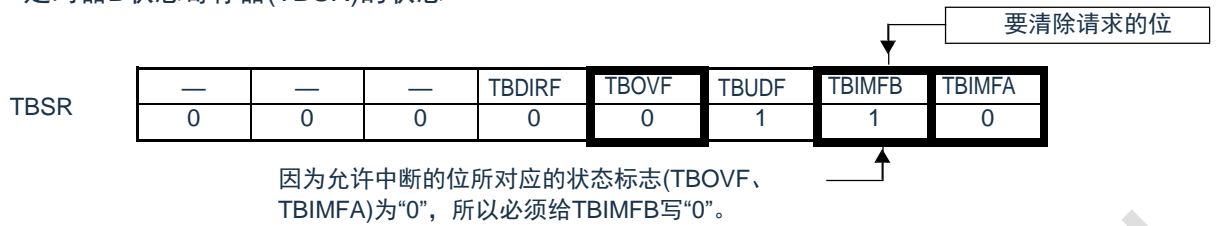
- 当TBSR寄存器的位为“1”并且对应的TBIER寄存器的位为“1”(允许中断)时, IF1D寄存器的Bit0位就变为“1”(有中断请求)。
- 当TBIER寄存器的多个位为“1”时, 必须通过TBSR寄存器判断是哪个请求源产生的中断。
- 因为即使接受中断, TBSR寄存器的各位也不会自动变为“0”, 所以必须在中断程序中将这位置“0”。
- 要将定时器B的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时, 如果该中断被定时器B中断允许寄存器(TBIER)设置为禁止中断, 就必须用以下(a)~(c)的任意一种方法置“0”。
  - 必须在将定时器B中断允许寄存器(TBIER)置“00H”(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



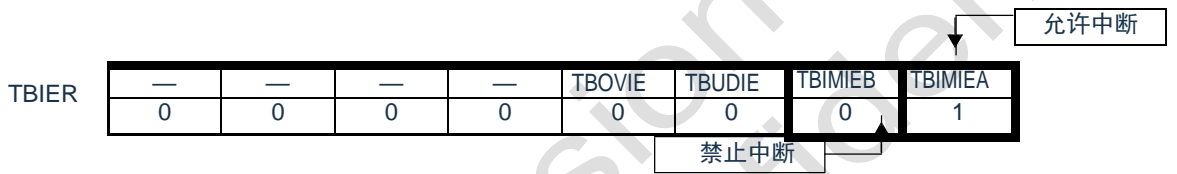
- 定时器B状态寄存器(TBSR)的状态



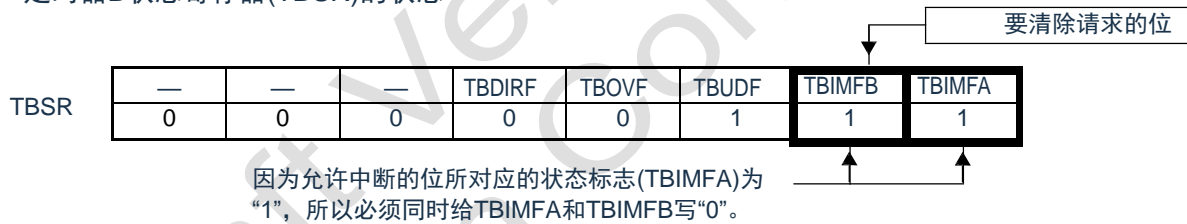
- (c) 当定时器B中断允许寄存器(TBIER)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(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

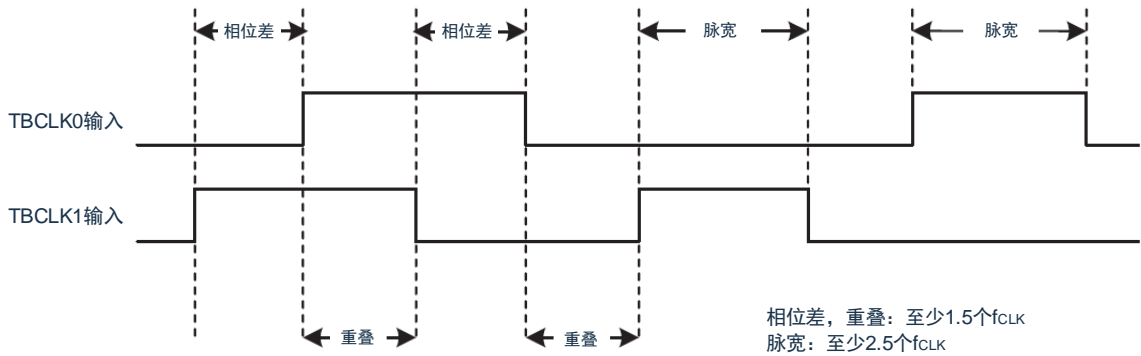


## 8.6 使用定时器B时的注意事项

### 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<sup>注</sup>。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TBIO10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<sub>j</sub>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

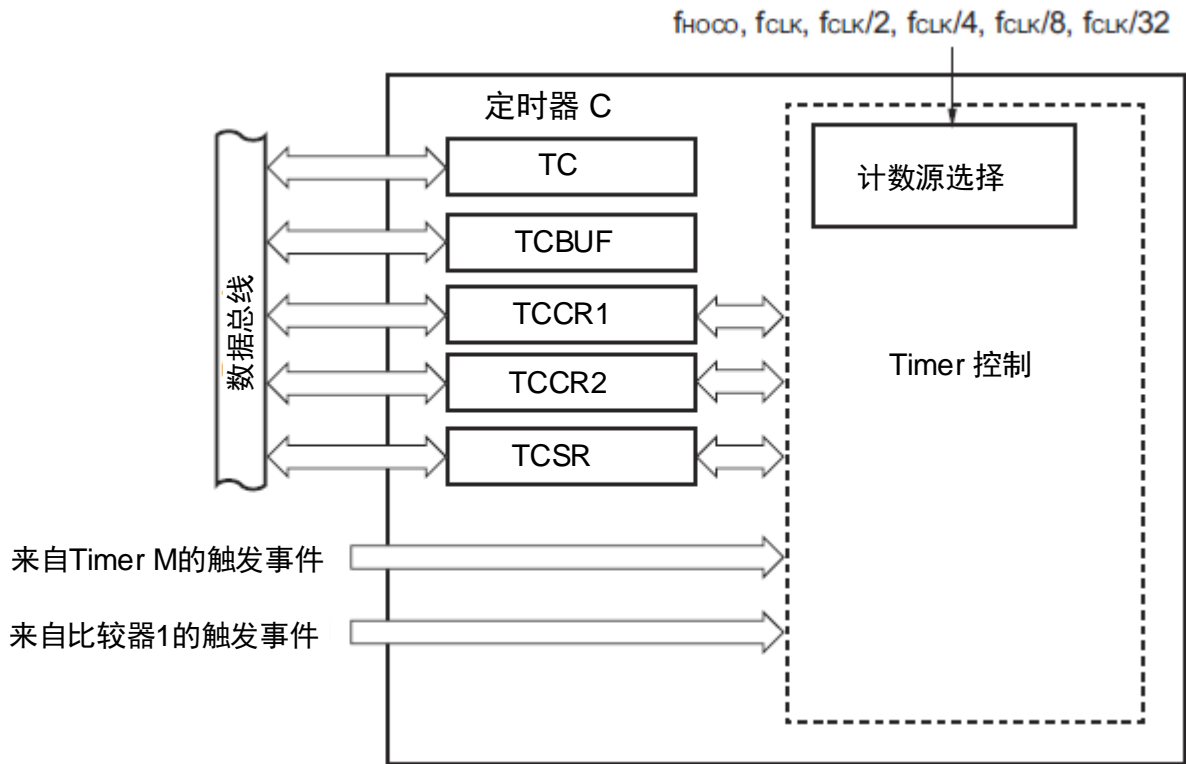
计数复位：由定时器M或者比较器1触发计数复位

Draft Version  
Cmsemicon Confidential

## 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	TCBUF0
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

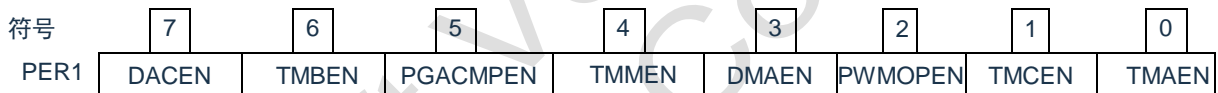
### 9.3.1 外围允许寄存器1(PER1)

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



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。

### 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

### 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

### 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 <sup>注</sup>
0	0	0	F <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /32
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。

### 9.3.5 定时器C控制寄存器2(TCCR2)

图9-6 定时器C控制寄存器2(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 <sup>注</sup>
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。

### 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 <sup>注1</sup>
0	计数停止
1	计数中

TCOVF	Timer C计数器的上溢状态标志位 <sup>注2,3</sup>
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。

## 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

由选项字节（000C2H）决定。

(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

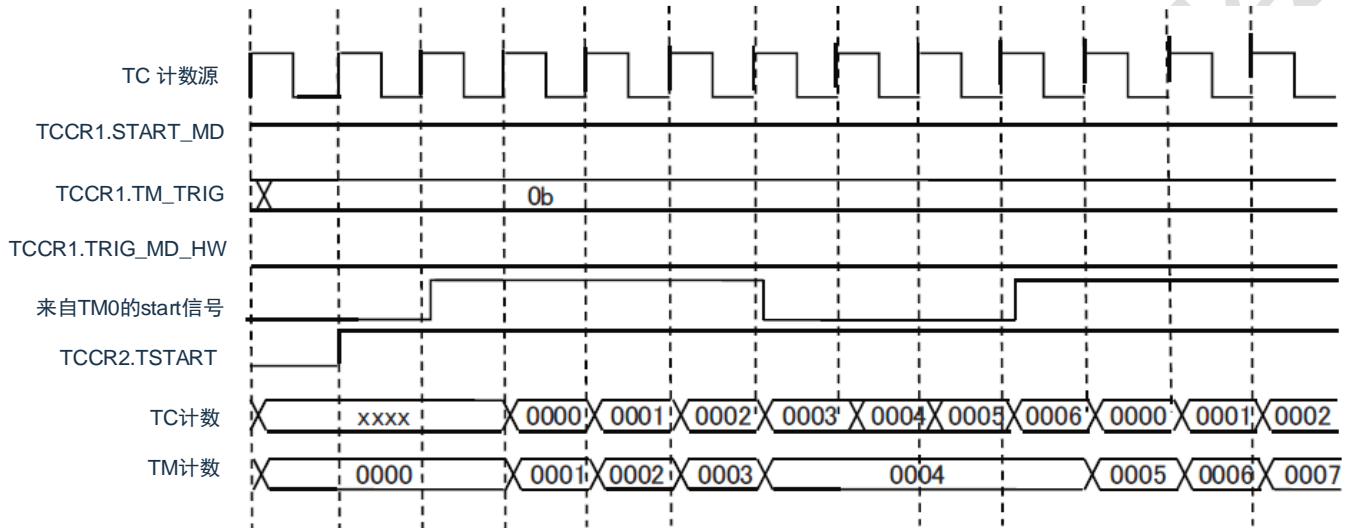
通过Timer M或者软件设置，开始Timer C的计数动作。



### 9.4.2.1 选择Timer M的信号作为触发时的设置和动作

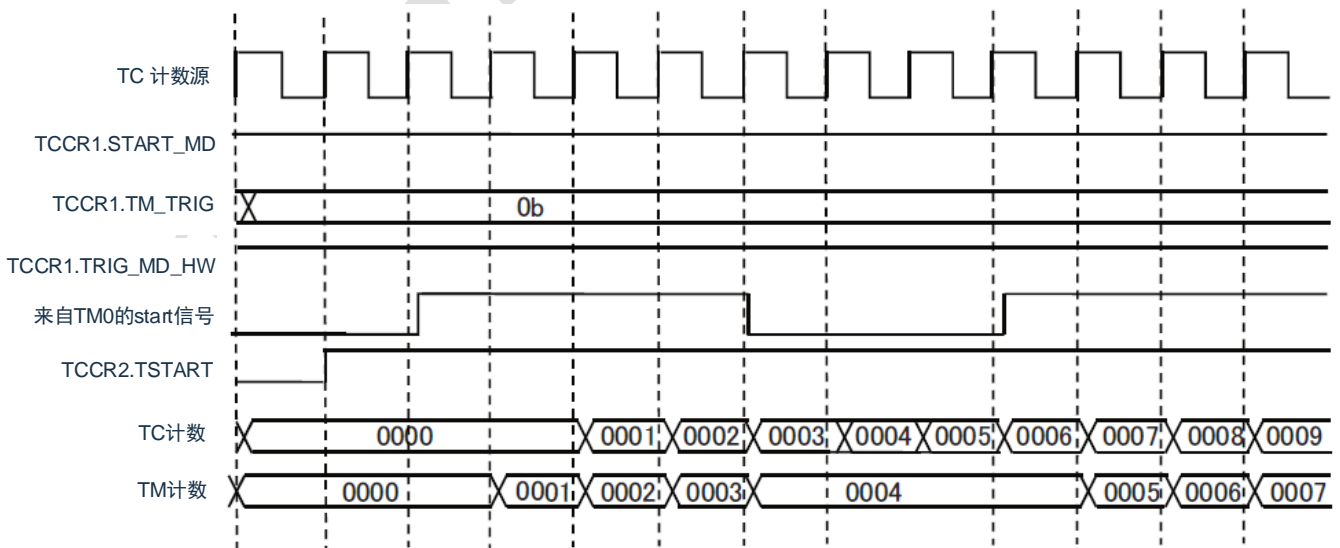
- (1) TRIG\_MD\_HW=0时, Timer C计数的复位和开始设置步骤:
  - (a) 选择Timer M输出信号作为计数开始的触发源: TCCR1.START\_MD=1
  - (b) 选择Timer C的触发功能: TCCR1.TRIG\_MD\_HW=0
  - (c) 选择Timer M\_0/1的触发信号: TCCR1.TC\_TRIG=1/0
  - (d) Timer C计数开始: TCCR2.TSTART=1

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

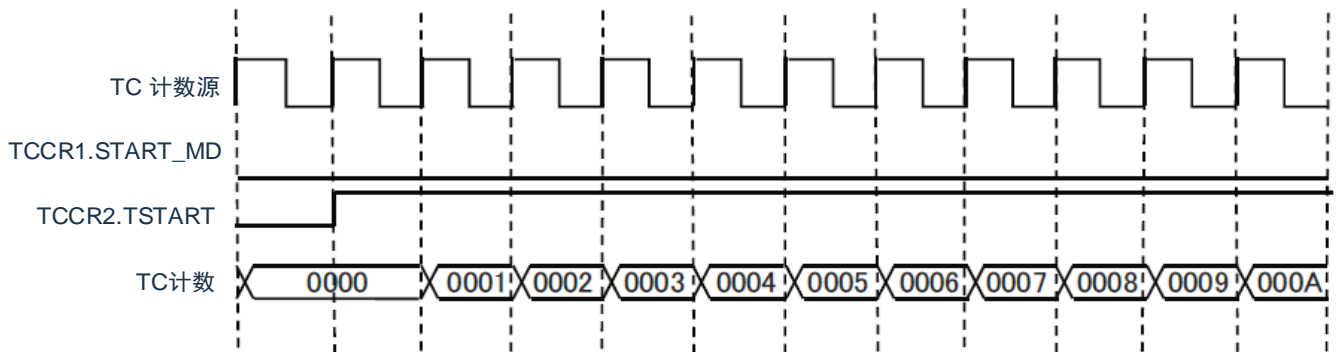
图9-9 Timer C计数开始的例子(TRIG\_MD\_HW=1)



### 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



Draft Version  
Cmsemicon Confidential

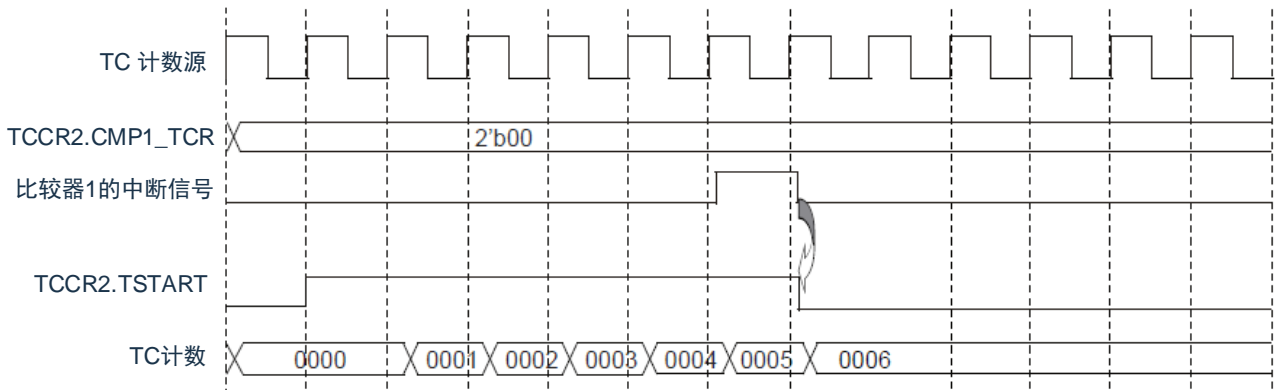
### 9.4.3 定时器C计数停止的动作

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

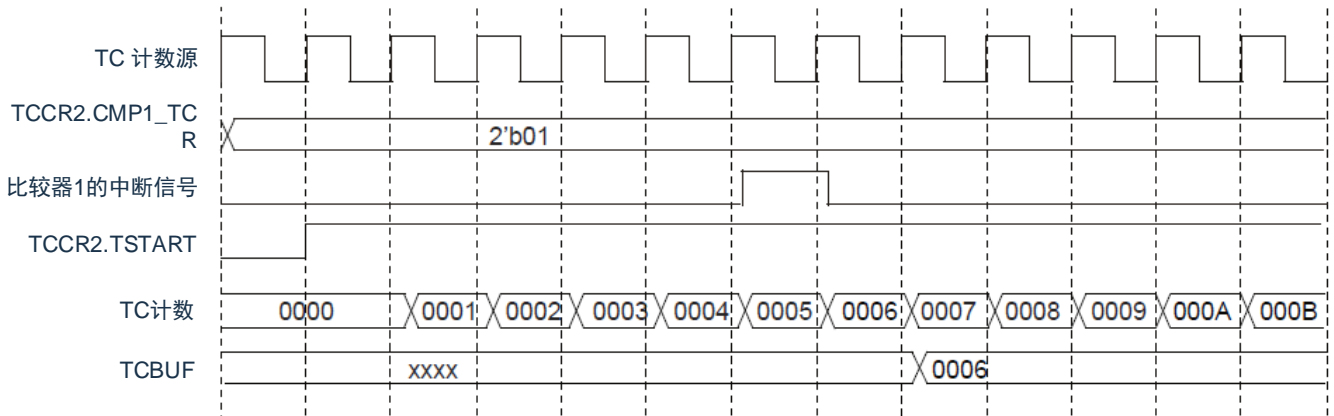
如果在Timer C动作过程中，比较器1产生了中断，Timer C的动作会发生变化。

(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

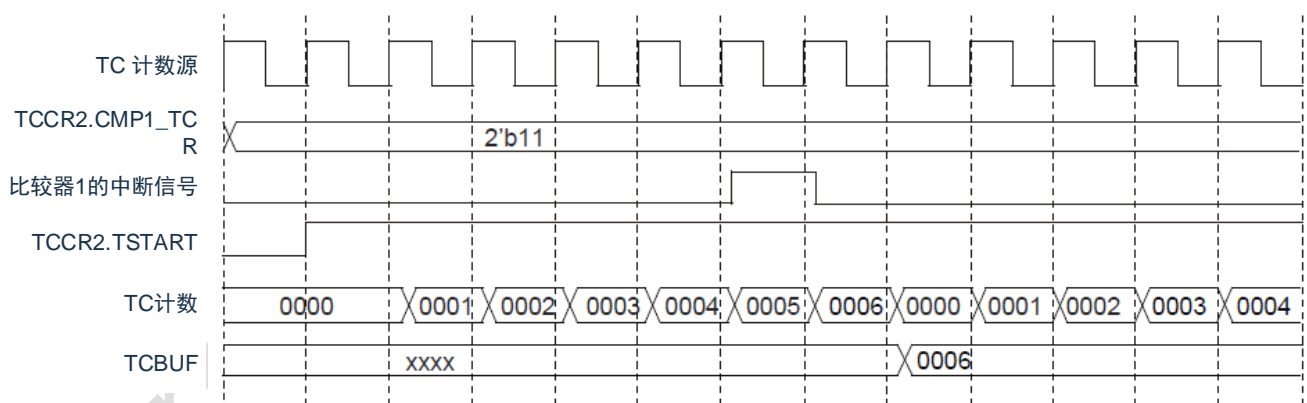


(2) Case 2:

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)

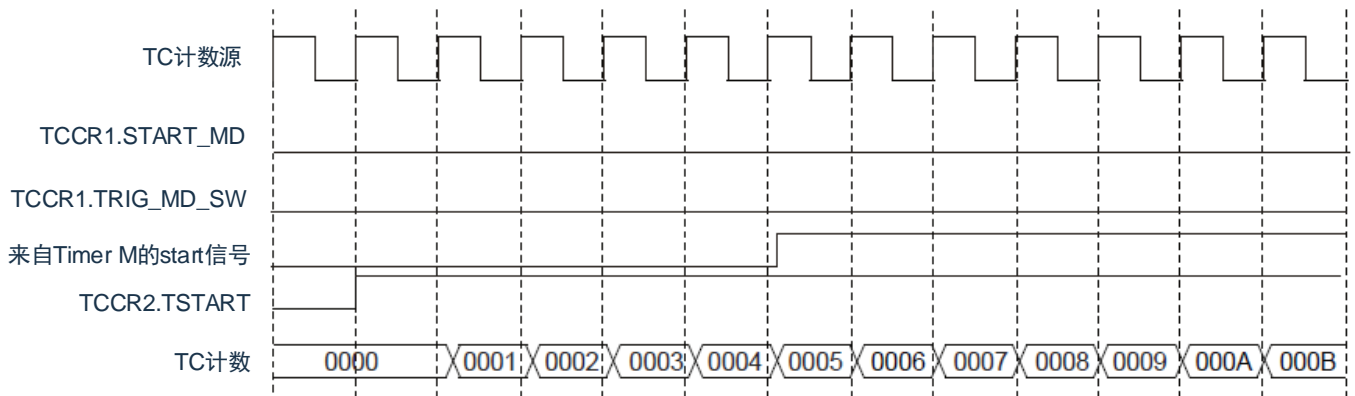


### 9.4.5 定时器C计数复位动作

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。

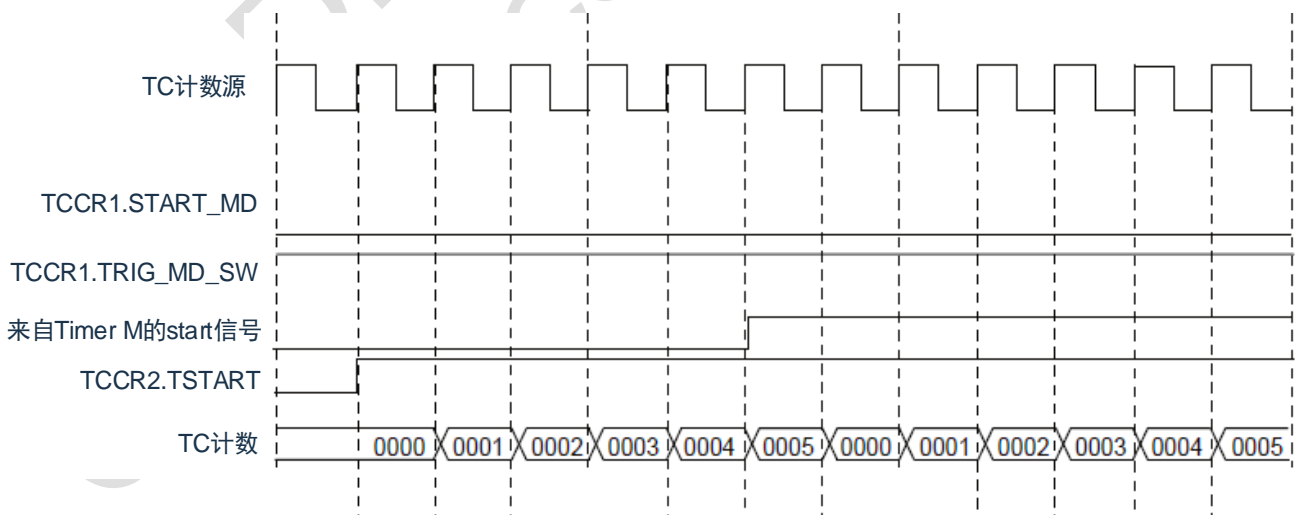
- (1) TRIG\_MD\_SW=0，软件触发计数开始时，Timer M的输出信号不能对计数值进行复位。
  - 软件触发计数开始：TCCR1.START\_MD=0
  - 允许软件复位计数器：TCCR1.TRIG\_MD\_SW=0
  - Timer C计数开始：TCCR2.TSTART=1

图9-14 Timer M触发Timer C计数复位的例子



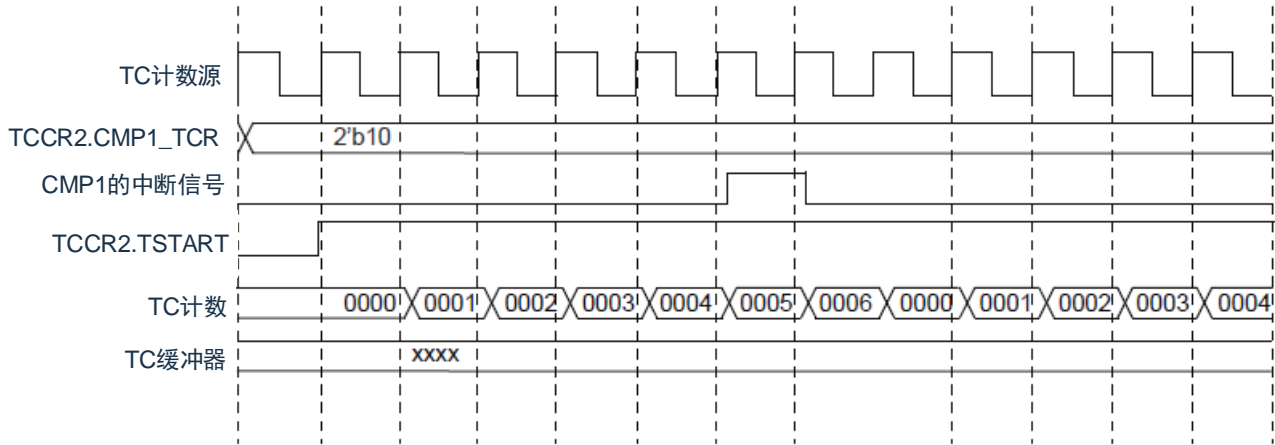
- (2) TRIG\_MD\_SW=1，软件触发计数开始时，Timer M的输出信号对计数值进行复位。
  - 软件触发计数开始：TCCR1.START\_MD=0
  - 禁止软件复位计数器：TCCR1.TRIG\_MD\_SW=1
  - Timer C计数开始：TCCR2.TSTART=1

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

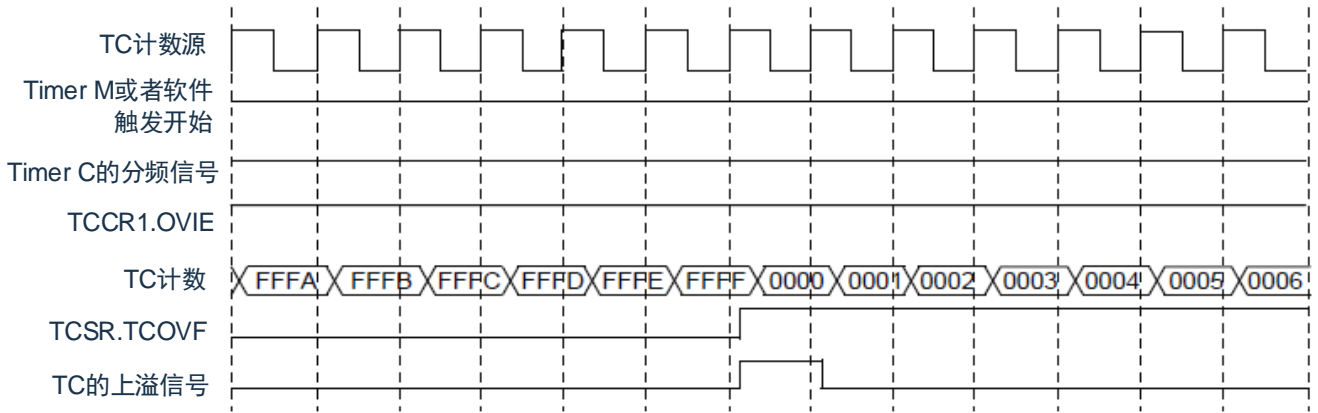


Draft Version  
Cmsemicon Confidential

### 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



Draft Version  
Cmsemicon Confidential

## 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15章 比较器”。
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。

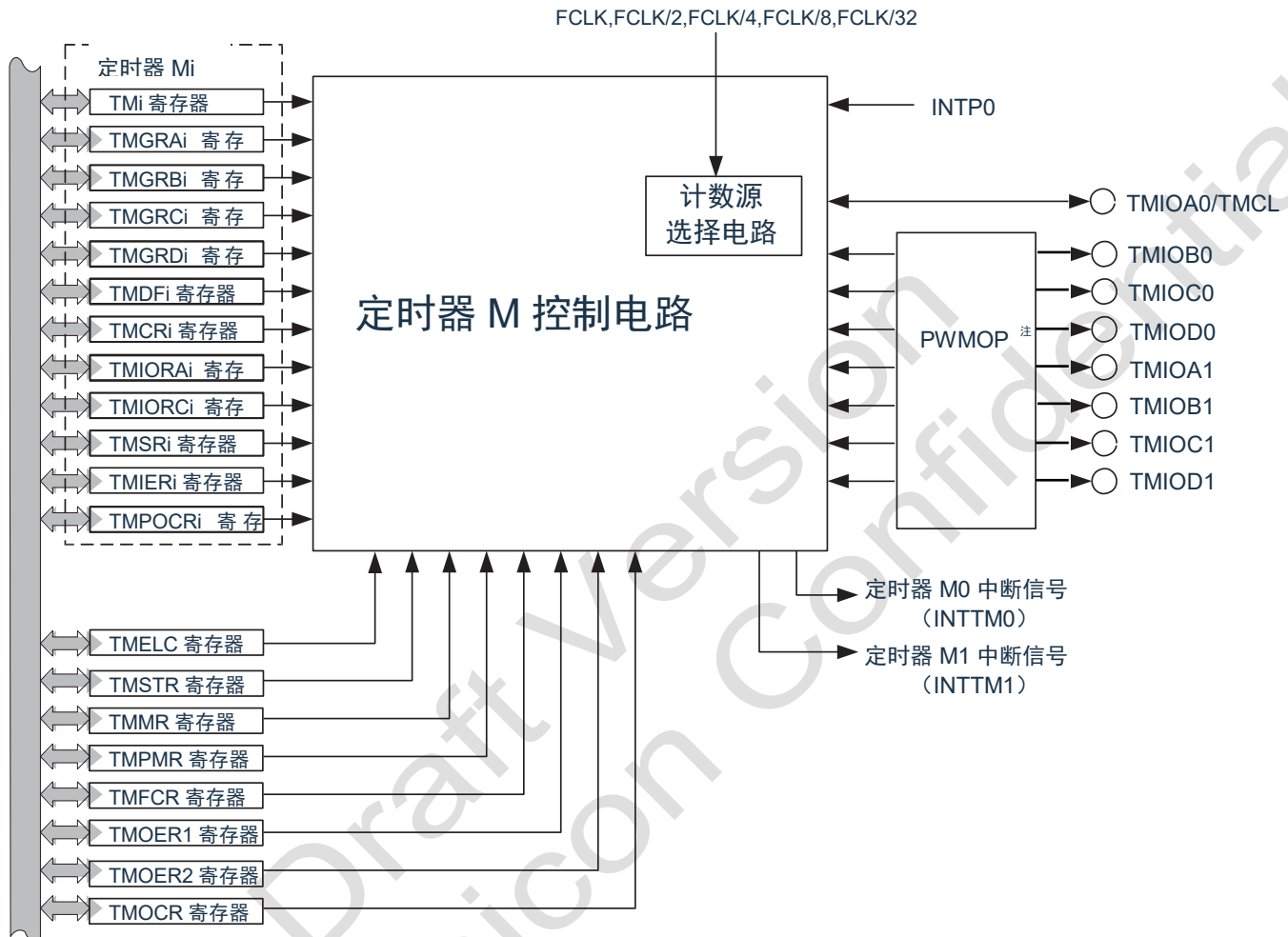
复位同步PWM模式、互补PWM模式和PWM3模式通过定时器M0和定时器M1的计数器和寄存器的组合来输出波形，引脚功能取决于运行模式。

定时器M有8个输入/输出引脚。

## 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	TMDF0
定时器M数字滤波器功能选择寄存器1	TMDF1
定时器M延时控制寄存器	TMDR
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
定时器M功能控制寄存器的镜像寄存器	TMFCRM
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。 <ul style="list-style-type: none"> <li>不能写定时器M使用的SFR。</li> <li>定时器M处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写定时器M使用的SFR。</li> </ul>

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。 <ul style="list-style-type: none"> <li>不能写PWM截止电路使用的SFR。</li> <li>PWM截止电路处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写PWM截止电路使用的SFR。</li> </ul>

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<sub>IH</sub>。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						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。

### 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						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 <sup>注1</sup>
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 <sup>注2</sup> 。

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 <sup>注2</sup> 。

TSTART1	TM1计数的开始标志 <sup>注3,4</sup>
0	停止计数。
1	开始计数。

TSTART0	TM0计数的开始标志 <sup>注5,6</sup>
0	停止计数。
1	开始计数。

注1: 不能用PWM3模式。

注2: 在使用输入捕捉功能时, 必须将此位置“1”。

注3: 当CSEL1位为“1”时, 必须给TSTART1位写“0”。

注4: 当CSEL1位为“0”并且产生比较匹配信号(即TM1与TMGRA1匹配)时, 此标志为“0”(停止计数)。

注5: 当CSEL0位为“1”时, 必须给TSTART0位写“0”。

注6: 当CSEL0位为“0”并且产生比较匹配信号(即TM0与TMGRA0匹配)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 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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1,2</sup>
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 <sup>注3</sup>
0	TM0和TM1独立运行。
1	TM0和TM1同步运行。

注1: 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2: 在互补PWM模式中, 必须将此位置“0”(通用寄存器)。

注3: 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 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	



## 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							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 <sup>注1</sup>
<ul style="list-style-type: none"> <li>在定时器模式中, 必须置“1”(非PWM3模式)。</li> <li>在PWM3模式中, 必须置“0”(PWM3模式)。</li> <li>在复位同步PWM模式和互补PWM模式中无效。</li> </ul>	

STCLK	外部时钟输入的选择
<ul style="list-style-type: none"> <li>定时器模式、复位同步PWM模式、互补PWM模式</li> <li>0: 外部时钟输入无效</li> <li>1: 外部时钟输入有效</li> <li>在PWM3模式中, 必须置“0”(外部时钟输入无效)。</li> </ul>	

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"> <li>复位同步PWM模式、互补PWM模式</li> <li>0: 初始输出“H”电平, “L”电平有效。</li> <li>1: 初始输出“L”电平, “H”电平有效。</li> <li>在定时器模式和PWM3模式中无效。</li> </ul>	

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"> <li>复位同步PWM模式、互补PWM模式</li> <li>0: 初始输出“H”电平, “L”电平有效。</li> <li>1: 初始输出“L”电平, “H”电平有效。</li> <li>在定时器模式和PWM3模式中无效。</li> </ul>	

CMD1	CMD0	组合模式的选择 <sup>注2、3</sup>
<ul style="list-style-type: none"> <li>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)。</li> <li>在复位同步PWM模式中, 必须置“01B”(复位同步PWM模式)。</li> <li>互补PWM模式</li> </ul>		
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: 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2: 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3: 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 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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1,2</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注2,3</sup>
0	允许输出。
1	禁止输出(TMIOA0引脚为I/O端口)。

注1: 在PWM3模式中, 必须将此位置“1”。

注2: 当使用PWM功能时, 必须将此位置“1”。

注3: 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	R/W					
符号	7	6	5	4	3	2	1	0	
TMOER2	TMPTO	0	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 <sup>注1</sup>
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: 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 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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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: 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 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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
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: 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 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 <sup>注1</sup>
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: 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 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 <sup>注1</sup>
0	0	F <sub>CLK</sub> /32
0	1	F <sub>CLK</sub> /8
1	0	F <sub>CLK</sub>
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：必须在设置DFCK0位和DFCK1位后开始计数。

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 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。



### 10.3.11 定时器M延时控制寄存器 (TMDR)

TMDR寄存器仅在互补PWM模式时使用，控制该模式下TMGRB0, TMGRA1, TMGRB1的比较匹配事件(IMFB0, IMFA1, IMFB1)的延时。

图10-15 定时器M延时控制寄存器(TMDR)的格式

地址: 0x40042A6C	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMDR	DSIDE[1:0]		0	0	0	0	0	DNUM[8:0]								

DSIDE[1:0]		延时边沿选择
0	0	匹配事件不延时，正常输出。
0	1	上升计数时延时，屏蔽下降计数时的匹配事件。
1	0	下降计数时延时，屏蔽上升计数时的匹配事件。
1	1	双边（上升和下降计数时）延时，匹配事件延时后输出。

DNUM[8:0]	延时时间设置 <sup>注1</sup>
1~511	事件延时 DNUM+1 个计数时钟。

注1：延时时间必须大于0。

### 10.3.12 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10-16 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 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) <sup>注1</sup>
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 <sup>注2</sup>
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 <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /3
1	0	1	TMCLK的输入 <sup>注3</sup>
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: 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i(TMCRi)(i=0、1)的格式[PWM功能]

地址: 0x40042A70(TMCR0)、0x40042A80(TMCR1) 复位后: 00H 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 <sup>注2</sup>
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 <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /32
1	0	1	TMCLK的输入 <sup>注2</sup>
上述以外			禁止设置。

注1: 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2: 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[复位同步PWM模式]

地址:	0x40042A70	复位后: 00H	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 <sup>注1</sup>
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 <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /32
1	0	1	TMCLK的输入 <sup>注2</sup>
上述以外			禁止设置。

注1: 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2: 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i(TMCRi) (i=0、1) 的格式[互补PWM模式]

 地址: 0x40042A70(TMCR0)、0x40042A80(TMCR1<sup>注1</sup>) 复位后: 00H 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“000B”(禁止清除(自由运行))。			

CKEG1	CKEG0	外部时钟沿的选择 <sup>注2,3</sup>
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 <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /32
1	0	1	TMCLK的输入 <sup>注4</sup>
上述以外			禁止设置。

注1: 必须给TMCR1寄存器的TCK0~TCK2位和TMCR0寄存器的TCK0~TCK2位设置相同的值(相同的计数源)。

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: 在TMFCR寄存器的STCLK位为“1”(外部时钟输入有效)时有效。

图10-20 定时器M控制寄存器0(TMCR0)的格式[PWM3模式]

地址:	0x40042A70	复位后: 00H	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 <sub>CLK</sub>
0	0	1	F <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /32
上述以外			禁止设置。

### 10.3.13 定时器MI/O控制寄存器Ai(TMIORAi)(i=0、1)

图10-21 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(TMIORA0)、0x40042A81H(TMIORA1)	复位后: 00H	R/W					
符号	7	6	5	4	3	2	1	0
TMIORAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TMGRBi模式的选择 <sup>注1</sup>
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOB1	IOB0	TMGRBi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	TMGRAi模式的选择 <sup>注2</sup>
在使用输入捕捉功能时, 必须置“1”(输入捕捉)。	

IOA1	IOA0	TMGRAi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: 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2: 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10-22 定时器MI/O控制寄存器Ai(TMIOAi)(i=0、1)的格式[输出比较功能]

地址:	0x40042A71(TMIOA0)、0x40042A81H(TMIOA1)		复位后: 00H		R/W			
符号	7	6	5	4	3	2	1	0
TMIOAi	0	IOB2	IOB1	IOB0	0	IOA2	IOA1	IOA0

IOB2	TMGRB模式的选择 <sup>注1</sup>
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 <sup>注2</sup>
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: 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TMIOAi寄存器的IOB2位和TMIORCi寄存器的IOD2位设置相同的值。

注2: 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TMIOAi寄存器的IOA2位和TMIORCi寄存器的IOC2位设置相同的值。



### 10.3.14 定时器MI/O控制寄存器Ci(TMIORCi)(i=0、1)

图10-23 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 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 <sup>注1</sup>
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 <sup>注2</sup>
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: 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2: 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10-24 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	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 <sup>注1</sup>
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	TMGRC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	TMGRC模式的选择 <sup>注2</sup>
在使用输出比较功能时，必须置“0”(输出比较)。	

IOC1	IOC0	TMGRC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：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<sub>i</sub>寄存器的IOB2位和TMIORC<sub>i</sub>寄存器的IOD2位设置相同的值。

注2：如果通过TMMR寄存器的TMBFCi位选择“1”(TMGRA<sub>i</sub>寄存器的缓冲寄存器)，就必须给TMIOA<sub>i</sub>寄存器的IOA2位和TMIORC<sub>i</sub>寄存器的IOC2位设置相同的值。

### 10.3.15 定时器M状态寄存器0(TMSR0)

图10-25 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	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 <sup>注1</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] 当TM0发生上溢时	

IMFD	输入捕捉/比较匹配标志D <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOD0引脚的输入边沿 <sup>注4</sup>	

IMFC	输入捕捉/比较匹配标志C <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOC0引脚的输入边沿 <sup>注3</sup>	

IMFB	输入捕捉/比较匹配标志B <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOB0引脚的输入边沿 <sup>注4</sup>	

IMFA	输入捕捉/比较匹配标志A <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOA0引脚的输入边沿 <sup>注4</sup>	

注1: 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”。

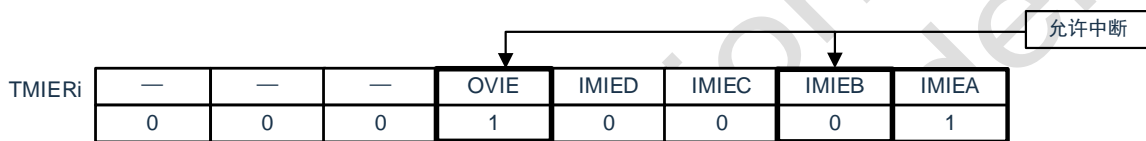
注2: 写的结果如下:

- 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(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”。

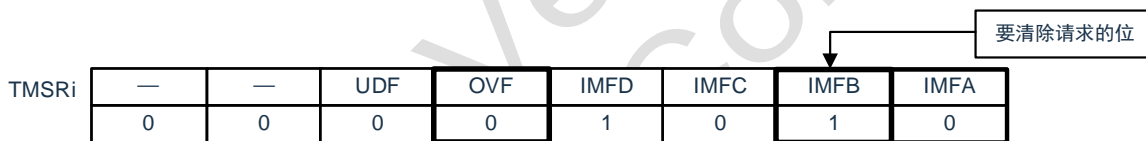
- 必须在将定时器M中断允许寄存器i(TMIERi)置“00H”(禁止全部中断)后给对象状态标志写“0”。
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“0”时, 必须给对象状态标志写“0”。

(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(TMIERi)的状态



- 定时器M状态寄存器(TMSRi)的状态

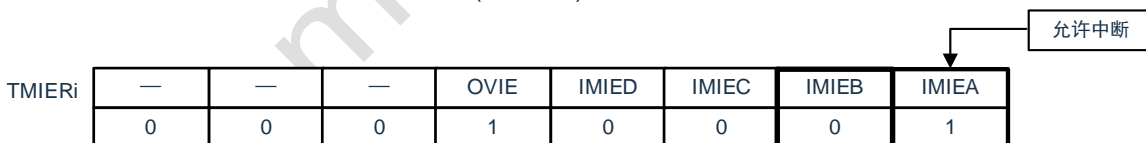


因为允许中断的位所对应的状态标志 (OVF、IMFA) 为“0”, 所以必须给 IMFB 写“0”。

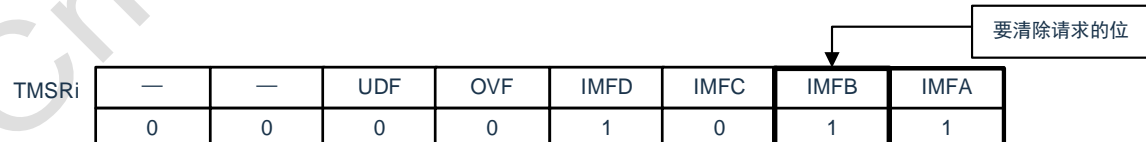
- 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时, 必须同时给此状态标志和对象状态标志写“0”。

(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(TMIERi)的状态



- 定时器M状态寄存器(TMSRi)的状态



因为允许中断的位所对应的状态标志 (IMFA) 为“1”, 所以必须同时给 IMFA 和 IMFB 写“0”。

注3: 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4: 这是TMIORA0寄存器的IOj1位和IOj0位(j=A或者B)选择的边沿。

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”。

图10-26 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						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 <sup>注2</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。	
[为“1”的条件] 当TM0发生上溢时	

IMFD	输入捕捉/比较匹配标志D <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。	
[为“1”的条件] 当TM0和TMGRD0的值相同时 <sup>注4</sup>	

IMFC	输入捕捉/比较匹配标志C <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。	
[为“1”的条件] 当TM0和TMGRC0的值相同时 <sup>注3</sup>	

IMFB	输入捕捉/比较匹配标志B <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。	
[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 <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。	
[为“1”的条件] 当TM0和TMGRA0的值相同时	

注1: 写的结果如下:

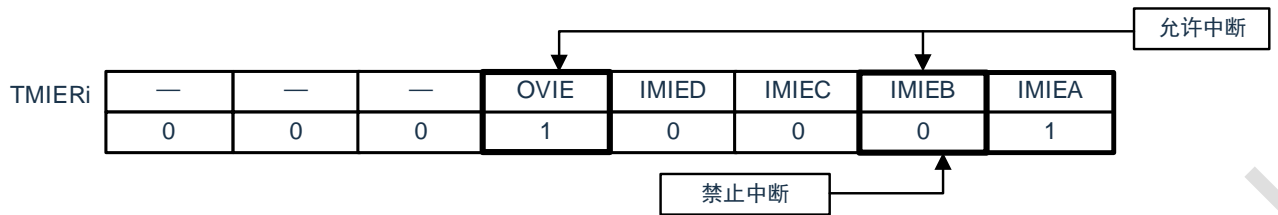
- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

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”。

- a) 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”。
- b) 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(TMIERi)的状态



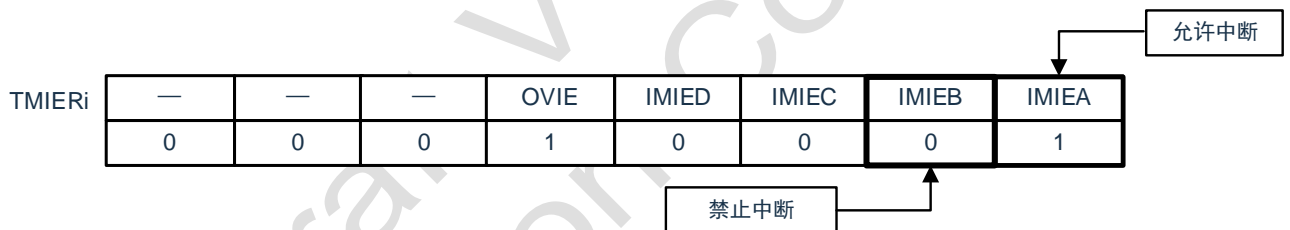
- 定时器M状态寄存器i(TMSRi)的状态



- 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(TMIERi)的状态



- 定时器M状态寄存器i(TMSRi)的状态



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: 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4: 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状态寄存器1(TMSR1)

图10-27 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	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 <sup>注1</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] 当TM1发生上溢时	

IMFD	输入捕捉/比较匹配标志D <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOD1引脚的输入边沿 <sup>注3</sup>	

IMFC	输入捕捉/比较匹配标志C <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOC1引脚的输入边沿 <sup>注3</sup>	

IMFB	输入捕捉/比较匹配标志B <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOB1引脚的输入边沿 <sup>注4</sup>	

IMFA	输入捕捉/比较匹配标志A <sup>注5</sup>
[为“0”的条件] 读后写“0” <sup>注2</sup> 。 [为“1”的条件] TMIOA1引脚的输入边沿 <sup>注4</sup>	

注1: 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:

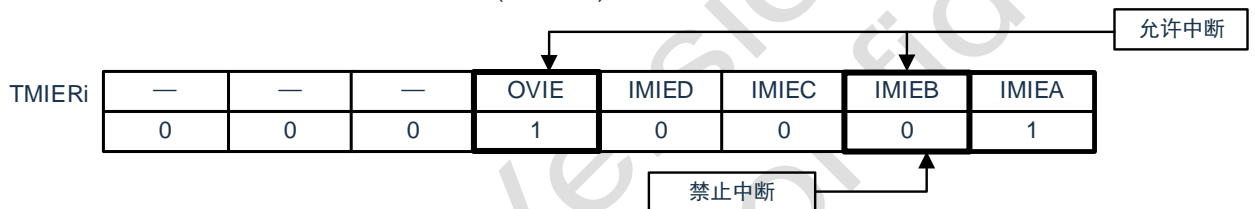
- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

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



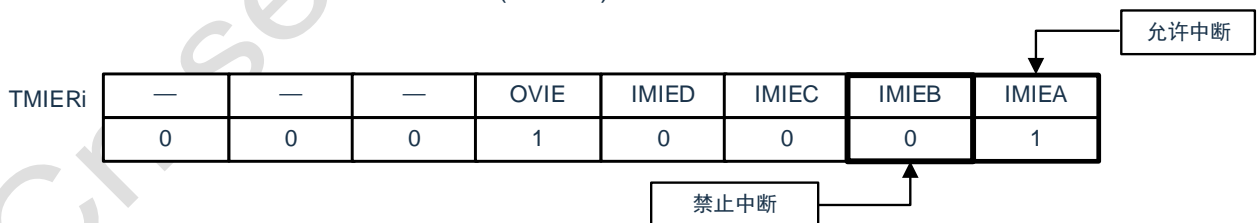
- 定时器M状态寄存器i(TMSRi)的状态



- 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i(TMSRi)的状态





注3: 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4: 这是TMIORA1寄存器的IOj1位和IOj0位(j=A或者B)选择的边沿。

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”。

Draft Version  
Cmsemicon Confidential

图10-28 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	RW					
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” <sup>注1</sup> 。 [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 <sup>注2</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。 [为“1”的条件] 当TM1发生上溢时	

IMFD	输入捕捉/比较匹配标志D <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。 [为“1”的条件] 当TM1和TMGRD1的值相同时 <sup>注3</sup>	

IMFC	输入捕捉/比较匹配标志C <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。 [为“1”的条件] 当TM1和TMGRC1的值相同时 <sup>注3</sup>	

IMFB	输入捕捉/比较匹配标志B <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。 [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 <sup>注4</sup>
[为“0”的条件] 读后写“0” <sup>注1</sup> 。 [为“1”的条件] 当TM1和TMGRA1的值相同时	

注1: 写的结果如下:

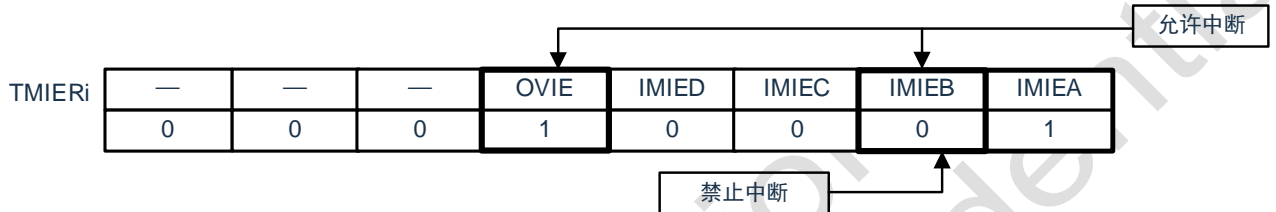
- 当写“1”时, 此位不变。
- 在读取值为“0”的情况下, 即使给相同的位写“0”也不变(在读后从“0”变为“1”的情况下, 即使写“0”也保持“1”的状态)。
- 在读取值为“1”的情况下, 如果给相同的位写“0”, 此位就变为“0”。

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”。

- a) 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”。
- b) 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



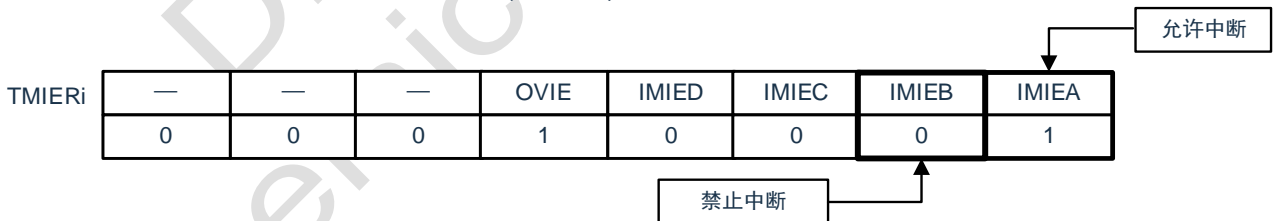
- 定时器M状态寄存器i(TMSRi)的状态



- 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: 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4: 当使用DMA时, IMFA位、IMFB位、IMFC位和IMFD位在DMA传送结束后变为“1”。

Draft Version  
Cmsemicon Confidential

### 10.3.17 定时器M中断允许寄存器i(TMIERi)(i=0、1)

图10-29 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	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)。

### 10.3.18 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10-30 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	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。

### 10.3.19 定时器M计数器i(TMi)(i=0、1)

[定时器模式]

必须以16位而不能以8位为单位存取TMi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10-31 定时器M计数器i(TMi)(i=0、1)的格式[定时器模式]

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	对计数源进行递增计数。 如果发生上溢，TMSRi寄存器的OVF位就变为“1”。	0000H~FFFFH

图10-32 定时器M计数器i(TMi)(i=0、1)的格式[复位同步PWM模式和PWM3模式]

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	对计数源进行递增计数。 如果发生上溢，TMSR0寄存器的OVF位就变为“1”。	0000H~FFFFH

**图10-33 定时器M计数器i(TM<sub>i</sub>)(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

TM<sub>i</sub>

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

—	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

**图10-34 定时器M计数器i(TM<sub>i</sub>)(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

TM<sub>i</sub>

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

—	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



### 10.3.20 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<sup>注</sup>、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<sup>注</sup>、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10-35 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 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10-3 使用输入捕捉功能时的TMGRji寄存器功能”。

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10-36 定时器M通用寄存器Ai、Bi、Ci、Di

(TMGRAi、TMGRBi、TMGRCi、TMGRDi)(i=0、1)的格式[输出比较功能]

地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH 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10-4 使用输出比较功能时的TMGRji寄存器功能”。

表10-4 使用输出比较功能时的TMGRji寄存器功能

寄存器	设置		寄存器功能	输出比较的输出引脚
	TMBFki	IOj3		
TMGRAi	—	—	通用寄存器，必须写比较值。	TMIOAi
TMGRBi				TMIOBi
TMGRCi	0	1	通用寄存器，必须写比较值。	TMIOCi
TMGRDi				TMIODi
TMGRCi	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”)。	TMIOAi
TMGRDi				TMIOBi
TMGRCi	0	0	TMIOAi输出控制	(参照“10.5.2(2)TMGRCi寄存器和 TMGRDi寄存器(i=0、1)的输出引脚的 变更”) TMIOAi
TMGRDi			TMIOBi输出控制	

注意: 如果将TMCRCi寄存器的TCK2~TCK0位置“000B”(F<sub>CLK</sub>)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

TMBFki: TMMR寄存器的位, IOj3: TMIORCi寄存器的位

图10-37 定时器M通用寄存器Ai、Bi、Ci、Di

 (TMGRA<sub>i</sub>、TMGRB<sub>i</sub>、TMGRC<sub>i</sub>、TMGRD<sub>i</sub>)(i=0、1)的格式[PWM功能]

地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH 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 <sub>i</sub>																
TMGRB <sub>i</sub>																
TMGRC <sub>i</sub>																
TMGRD <sub>i</sub>																

—	功能
bit15~0	参照“表10-5 使用PWM功能时的TMGR <sub>ji</sub> 寄存器功能”。

 表10-5 使用PWM功能时的TMGR<sub>ji</sub>寄存器功能

寄存器	设置	寄存器功能	PWM输出引脚
TMGRA <sub>i</sub>	—	通用寄存器，必须设置PWM周期。	—
TMGRB <sub>i</sub>	—	通用寄存器，必须设置PWM输出的变化点。	TMIOB <sub>i</sub>
TMGRC <sub>i</sub>	TMBFC <sub>i</sub> =0	通用寄存器，必须设置PWM输出的变化点。	TMIOC <sub>i</sub>
TMGRD <sub>i</sub>	TMBFD <sub>i</sub> =0		TMIOD <sub>i</sub>
TMGRC <sub>i</sub>	TMBFC <sub>i</sub> =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”)。	—
TMGRD <sub>i</sub>	TMBFD <sub>i</sub> =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”)。	TMIOB <sub>i</sub>

注意：如果将TMCRC<sub>i</sub>寄存器的TCK2~TCK0位置“000B”(F<sub>CLK</sub>)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

TMBFC<sub>i</sub>、TMBFD<sub>i</sub>：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[复位同步PWM模式]

 地址: 0x40042A78(TMGRA0)、0x40042A7A(TMGRB0)、 复位后: FFFFH 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”。

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。	TMIOB0、TMIOD0
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。	TMIOA1、TMIOC1
TMGRB1	—	通用寄存器，必须设置PWM3输出的变化点。	TMIOB1、TMIOD1
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”)。	TMIOB0、TMIOD0
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”)。	TMIOB1、TMIOD1

注意：如果将TMCRO寄存器的TCK2~TCK0位置“000B”(F<sub>CLK</sub>)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10-39 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 R/W  
 0x40042B58(TMGRD0)、0x40042B5A(TMGRD1)、  
 0x40042A88(TMGRA1)、0x40042A8A(TMGRB1)、  
 0x40042B5C(TMGRD0)、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10-7 互补PWM模式中的TMGRji寄存器功能”。

表10-7 互补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。	TMIOB0、TMIOD0
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。	TMIOA1、TMIOC1
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。	TMIOB1、TMIOD1
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。	TMIOB0、TMIOD0
TMGRD1	—	(在互补PWM模式中不使用)	—

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。	TMIOB1、TMIOD1

注意：如果将TMCRi寄存器的TCK2~TCK0位置“000B”(F<sub>CLK</sub>)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10-40：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 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”。

表10-8 PWM3模式中的TMGR<sub>j</sub>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TMCRO寄存器的TCK2~TCK0位置“000B”(F<sub>CLK</sub>)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



### 10.3.21 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10-41 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

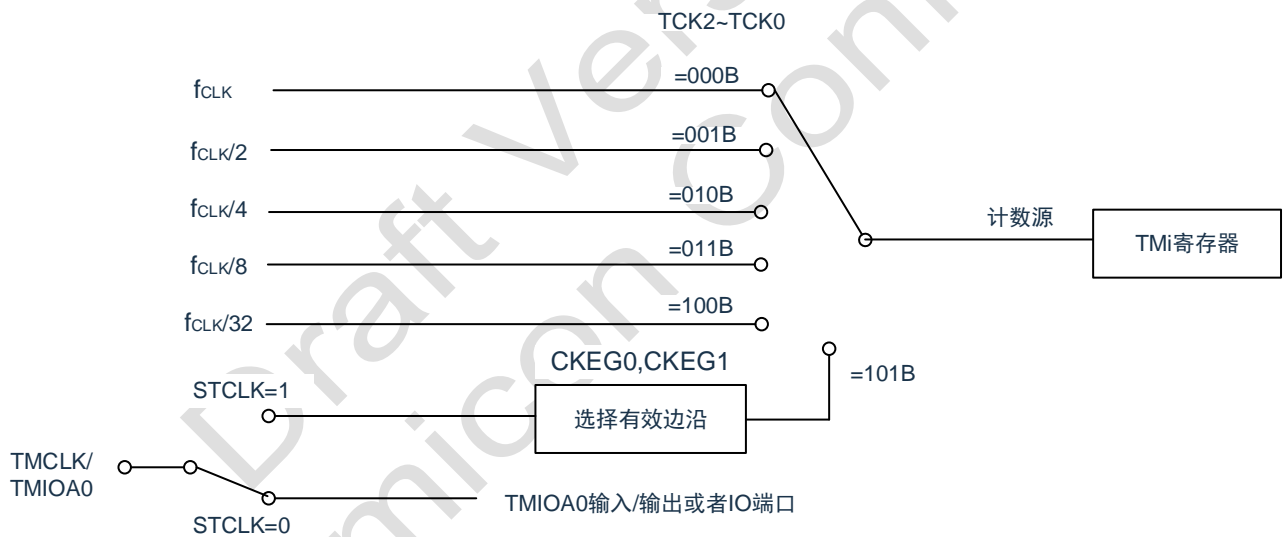
所有模式的计数源的选择方法都一样。但是，在PWM3模式中，不能选择外部时钟。

表10-9 计数源的选择

计数源	选择方法
$F_{CLK}$ 、 $F_{CLK}/2$ 、 $F_{CLK}/4$ 、 $F_{CLK}/8$ 、 $F_{CLK}/32$	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

图10-42 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

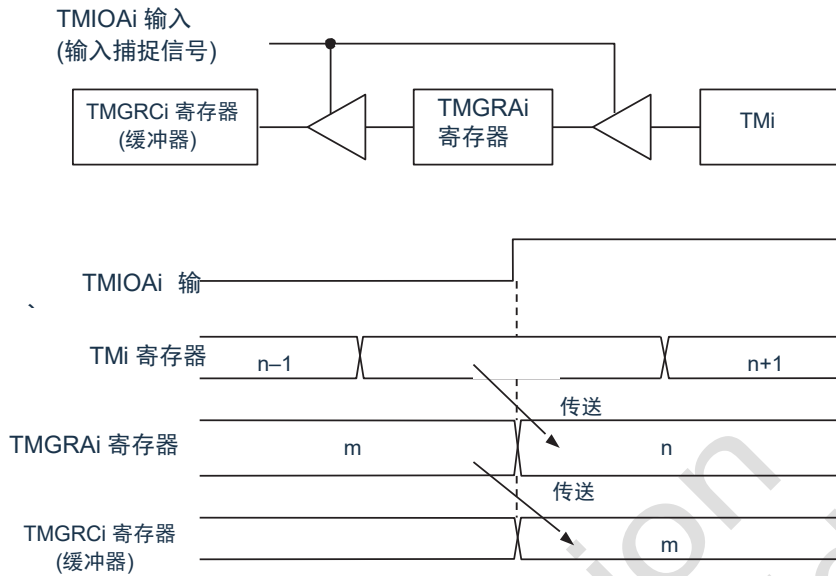
缓冲器运行因定时器模式而不同，各模式的缓冲器运行如表10-10所示。

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"> <li>• 当TMFCR寄存器的CMD1位和CMD0位是“11B”时，为TM1寄存器的下溢。</li> <li>• 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。</li> </ul>	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10-43 输入捕捉功能的缓冲器运行

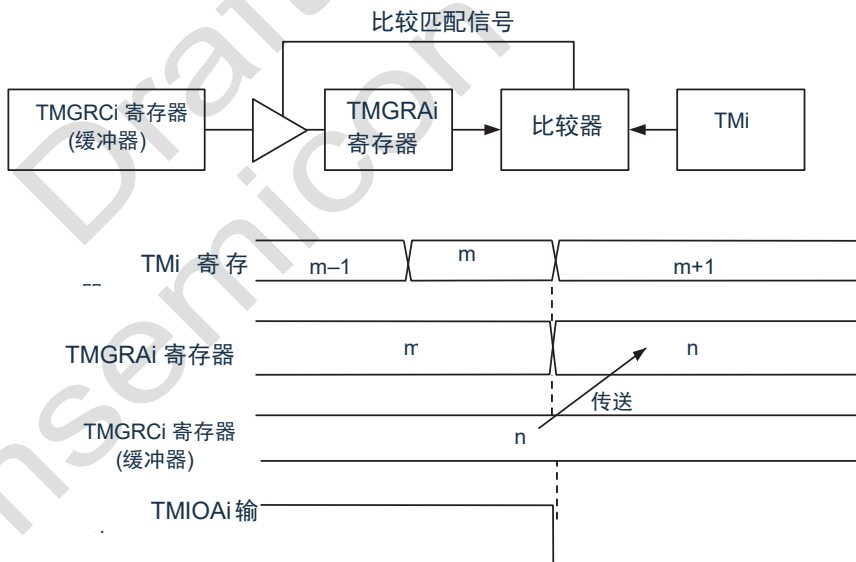


备注:  $i=0, 1$

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10-44 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”。

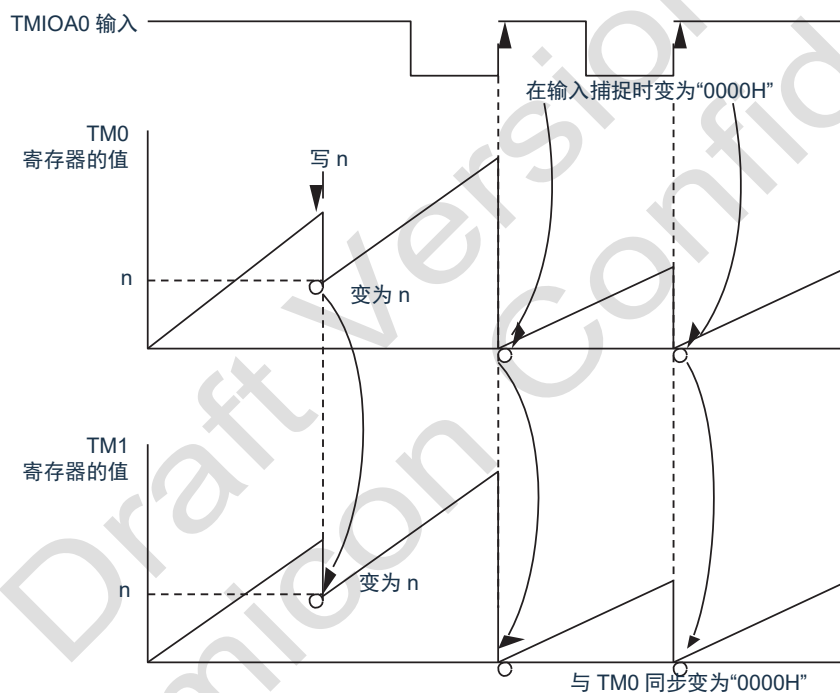
### 10.4.3 同步运行

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TMi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10-45 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

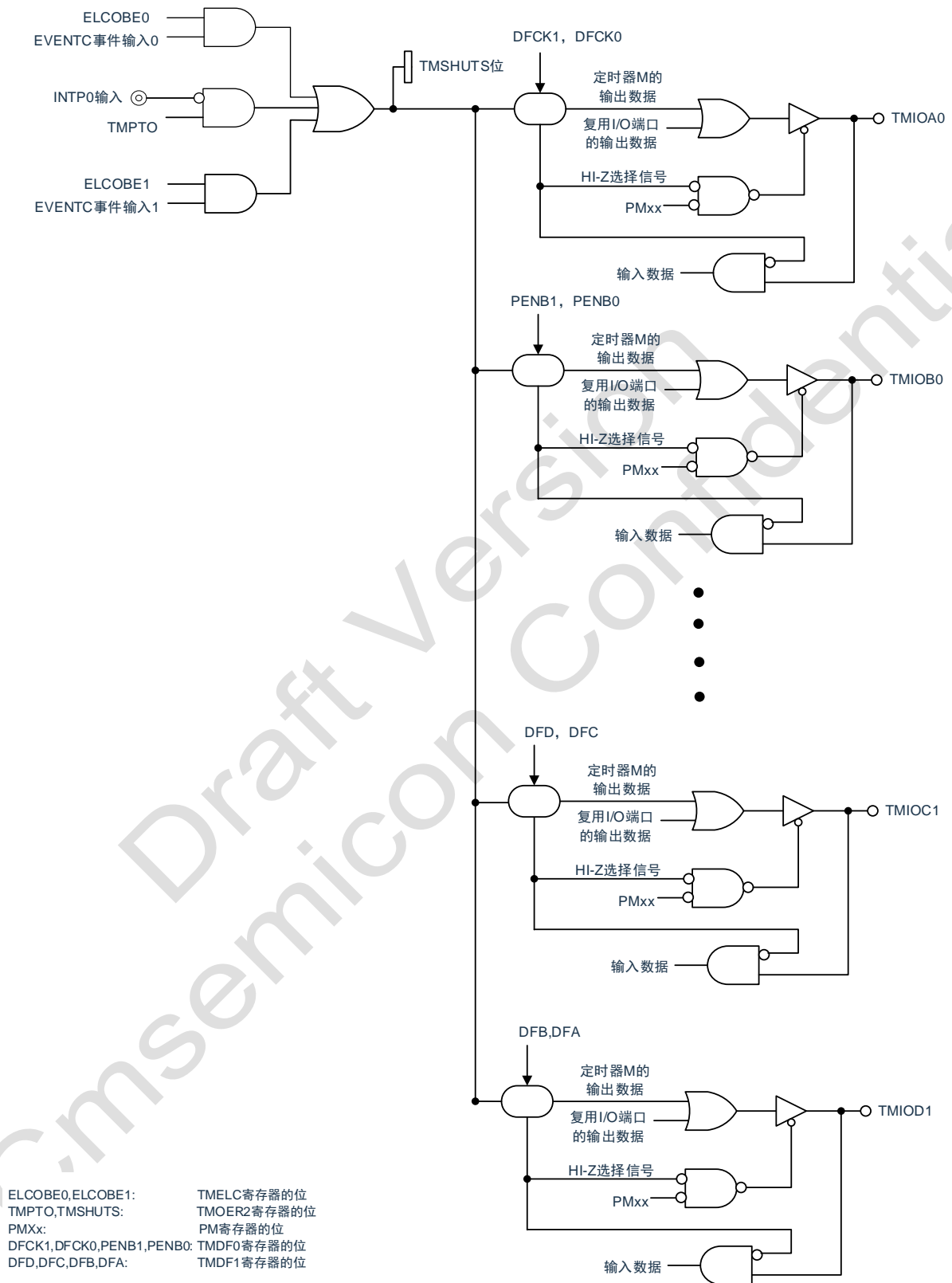
当使用PWM功能时或者在复位同步PWM模式、互补PWM模式和PWM3模式中，能通过INTP0引脚的输入来截止TMIO<sub>j</sub>输出引脚( $i=0, 1, j=A, B, C, D$ )的脉冲输出。

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<sub>i</sub>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10-46 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<sup>注</sup>

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<sub>ji</sub>引脚(i=0、1, j=A、B、C、D)的外部信号为触发，将TMI<sub>i</sub>寄存器(计数器)的内容传送到TMGR<sub>ji</sub>寄存器(输入捕捉)。因为TMIO<sub>ji</sub>引脚和TMGR<sub>ji</sub>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10-47和图10-48所示，输入捕捉功能的规格如表10-12所示。

图10-47：输入捕捉功能的框图

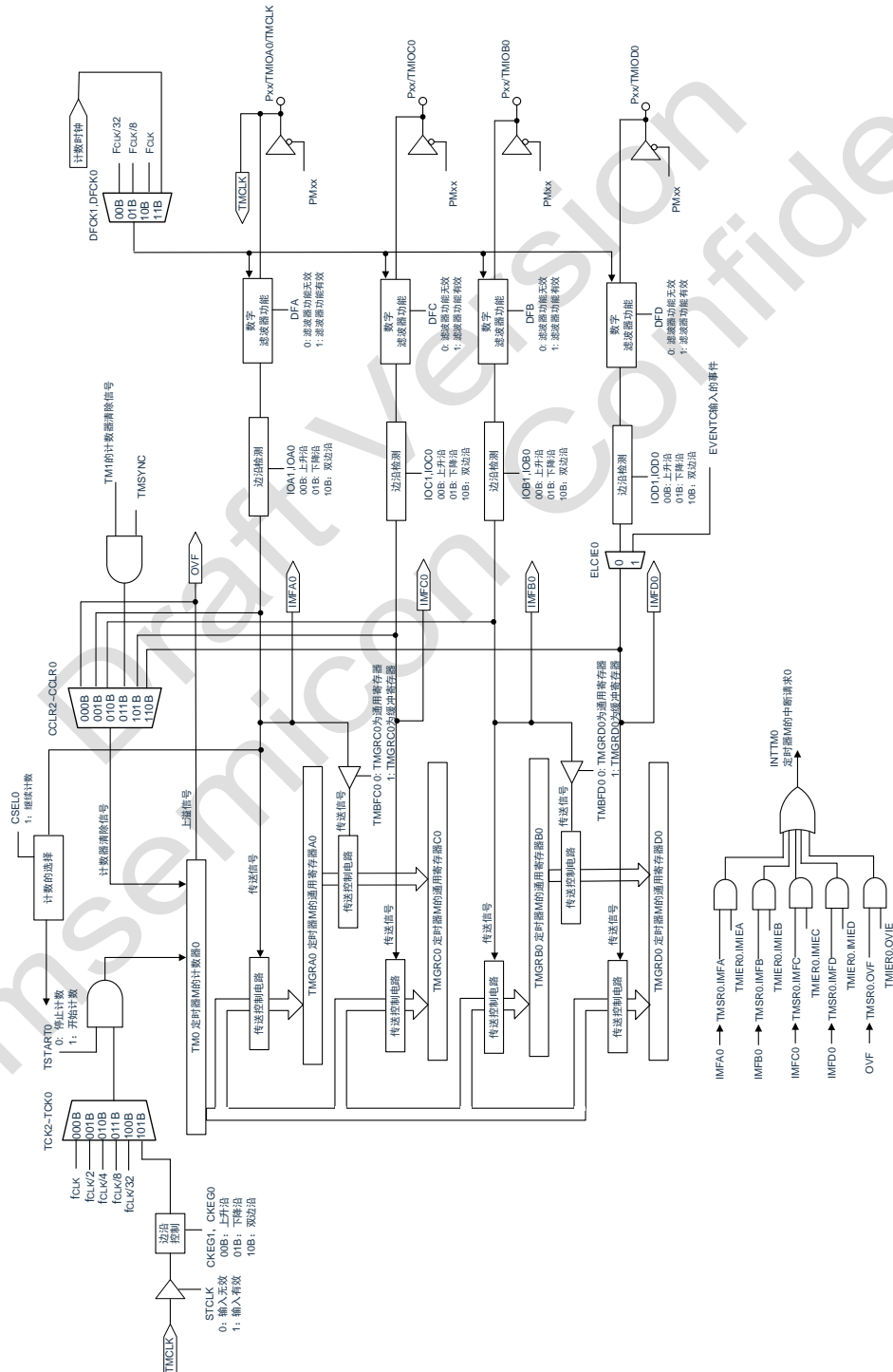


表10-12: 输入捕捉功能的规格

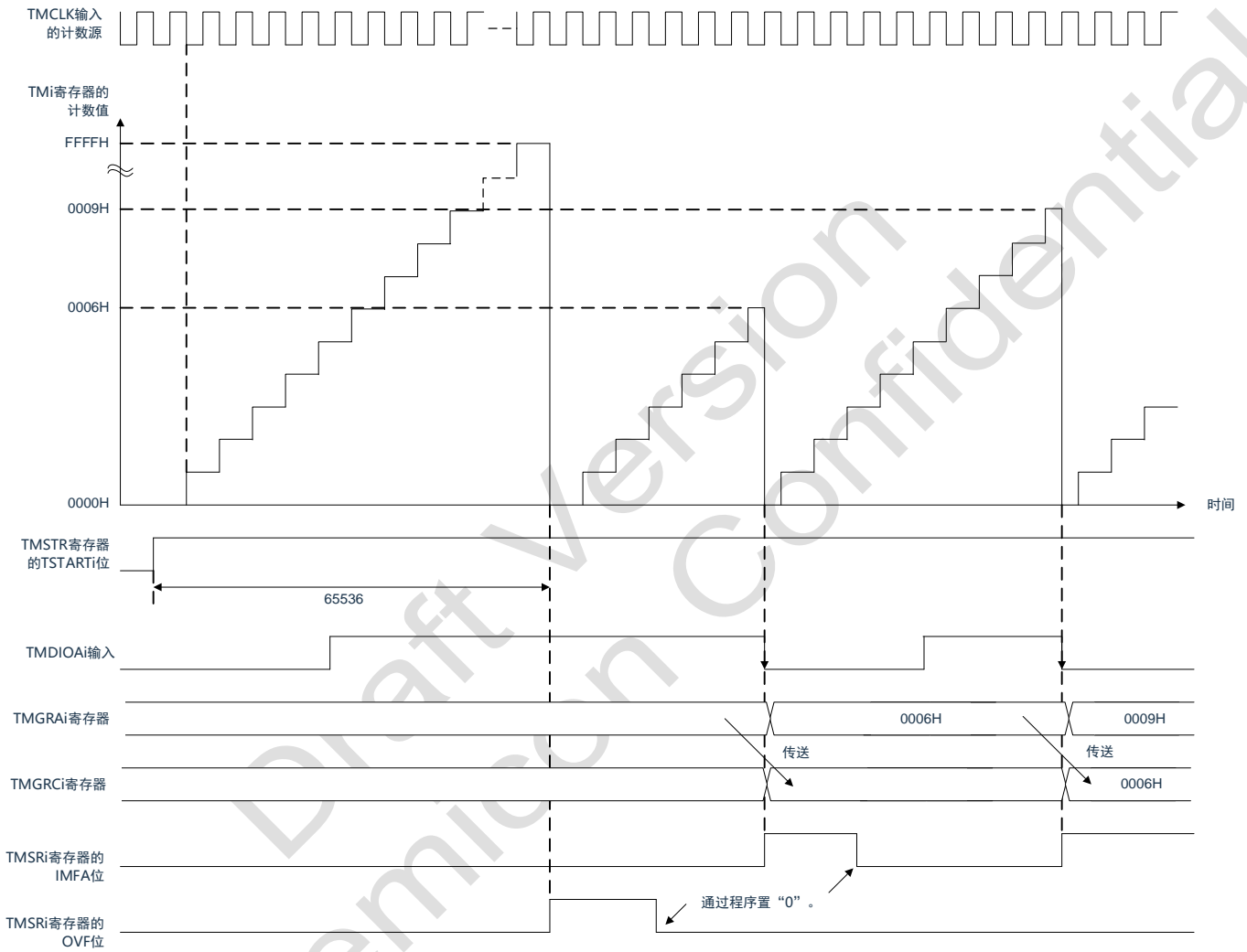
项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /32 TMCLK引脚的外部输入信号(通过程序选择有效边沿)
计数	递增计数
计数周期	当TMCRI寄存器的CCLR2~CCLR0位为“000B”(自由运行)时 1/fk 65536 fk: 计数源的频率
计数开始条件	给TMSTR寄存器的TSTARTi位写“1”(开始计数)。
计数停止条件	当TMSTR寄存器的CSELi位为“1”时, 给TSTARTi位写“0”(停止计数)。
中断请求的产生时序	<ul style="list-style-type: none"> <li>• 输入捕捉(TMIOji输入的有效边沿)</li> <li>• Tmi的上溢</li> </ul>
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"> <li>• 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。</li> <li>• 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。</li> </ul>
选择功能	<ul style="list-style-type: none"> <li>• 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</li> <li>• 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</li> <li>• 将Tmi置“0000H”的时序上溢或者输入捕捉</li> <li>• 缓冲器运行(参照“10.4.2缓冲器运行”)</li> <li>• 同步运行(参照“10.4.3同步运行”)</li> <li>• 数字滤波器 对TMIOji输入进行采样, 如果信号3次相同, 就视为电平已确定。</li> <li>• 通过ELC输入的事件进行的输入捕捉运行</li> </ul>

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10-48 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)。  
 TMIORAi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)。

(2) 数字滤波器

对TMIOj输入(i=0、1, j=A、B、C、D)进行采样, 如果信号3次相同, 就视为电平已确定。必须通过TMDFi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10-49所示。

图10-49 数字滤波器的框图

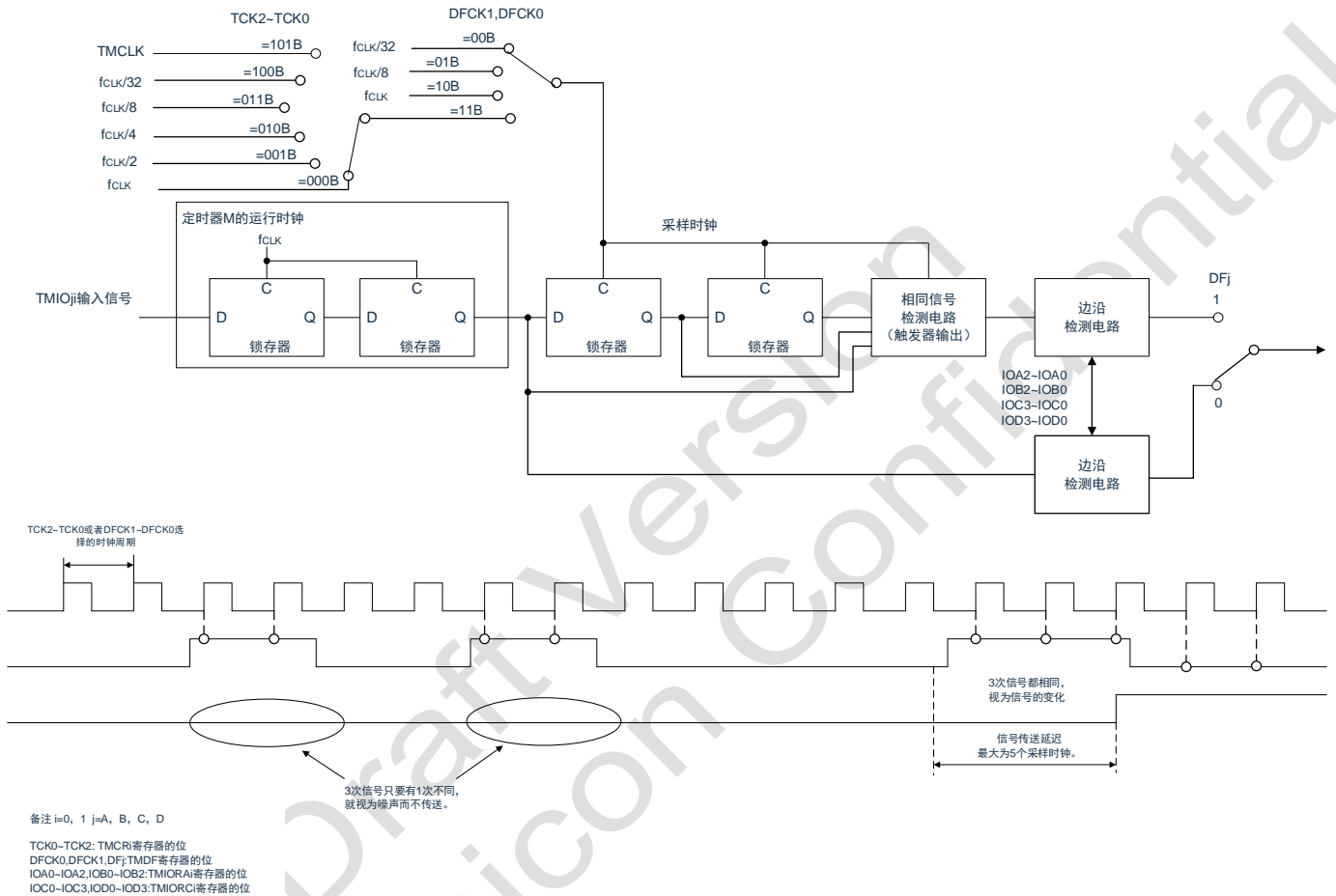




表10-13 输出比较功能的规格

项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /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"> <li>当TMCRi寄存器的CCLR2~CCLR0位为“000B”(自由运行)时 <math>1/fk \times 65536</math> fk: 计数源的频率</li> <li>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 <math>1/fk \times (n+1)</math> n: TMGRji寄存器的设置值</li> </ul>
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"> <li>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。</li> <li>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。</li> </ul>
中断请求的产生时序	<ul style="list-style-type: none"> <li>比较匹配(Tmi寄存器和TMGRji寄存器的内容相同)</li> <li>Tmi的上溢</li> </ul>
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"> <li>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。</li> <li>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。</li> </ul>
选择功能	<ul style="list-style-type: none"> <li>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</li> <li>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</li> <li>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。</li> <li>将Tmi置“0000H”的时序 上溢或者TMGRAi寄存器的比较匹配</li> <li>缓冲器运行(参照“10.4.2 缓冲器运行”)</li> <li>同步运行(参照“10.4.3 同步运行”)</li> <li>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。</li> <li>能将定时器M用作内部定时器而不进行输出。</li> </ul>

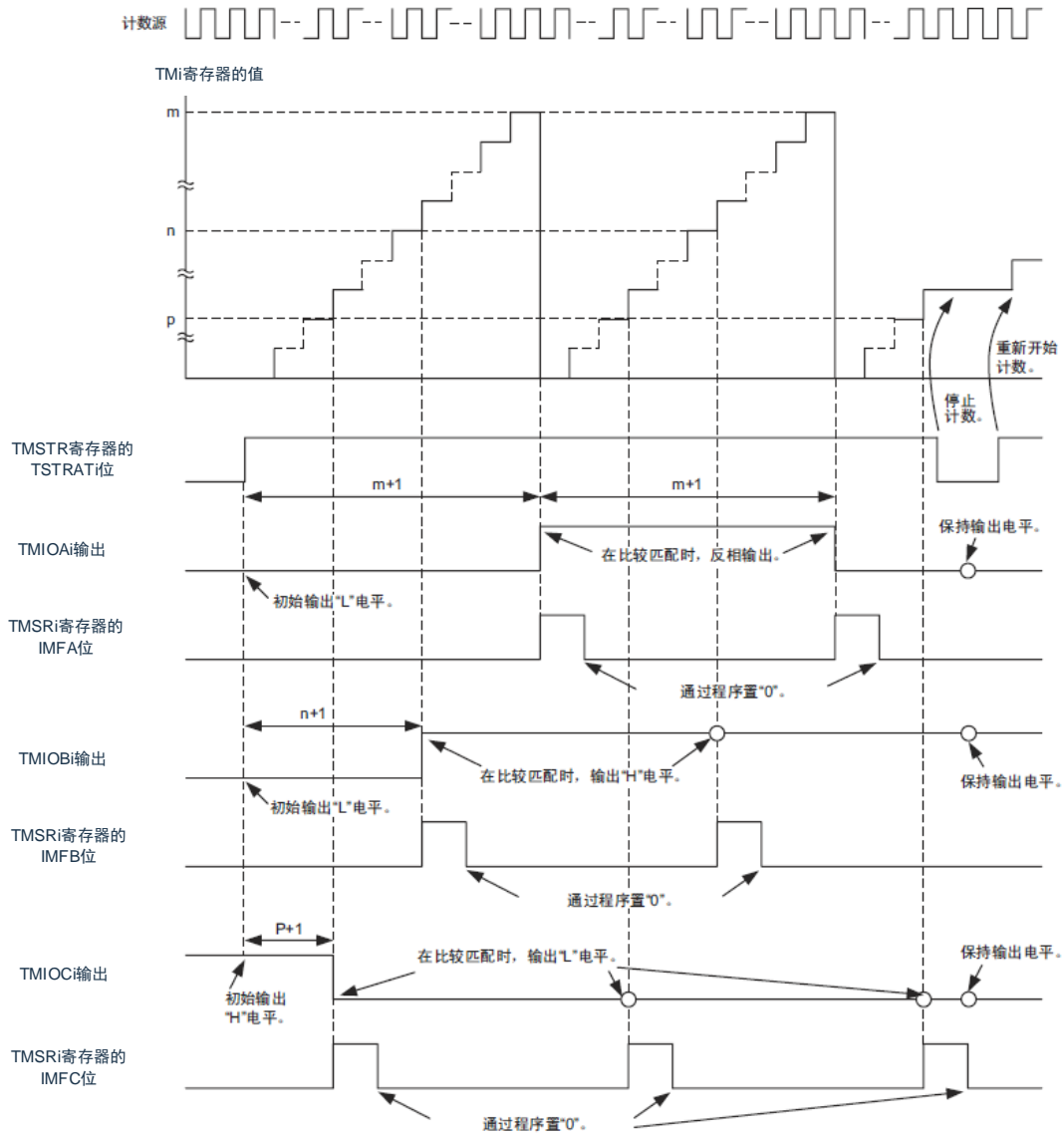
备注: i=0、1, j=A、B、C、D



(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10-51 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）。

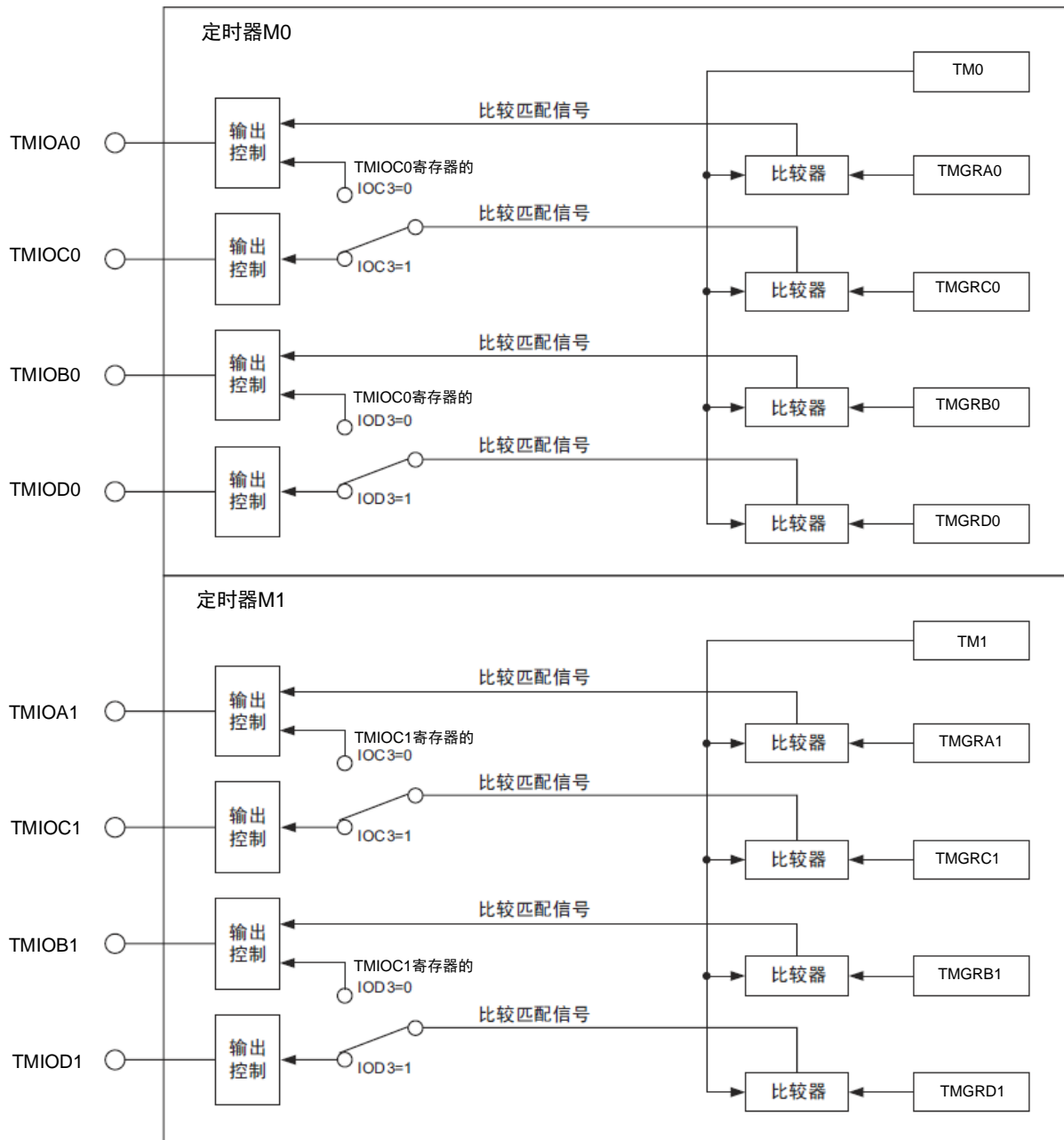
(2) TMGR*C*寄存器和TMGRD*i*寄存器(*i*=0、1)的输出引脚的变更

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10-52: TMGR*C* 和 TMGRD*i* 的输出引脚的变更

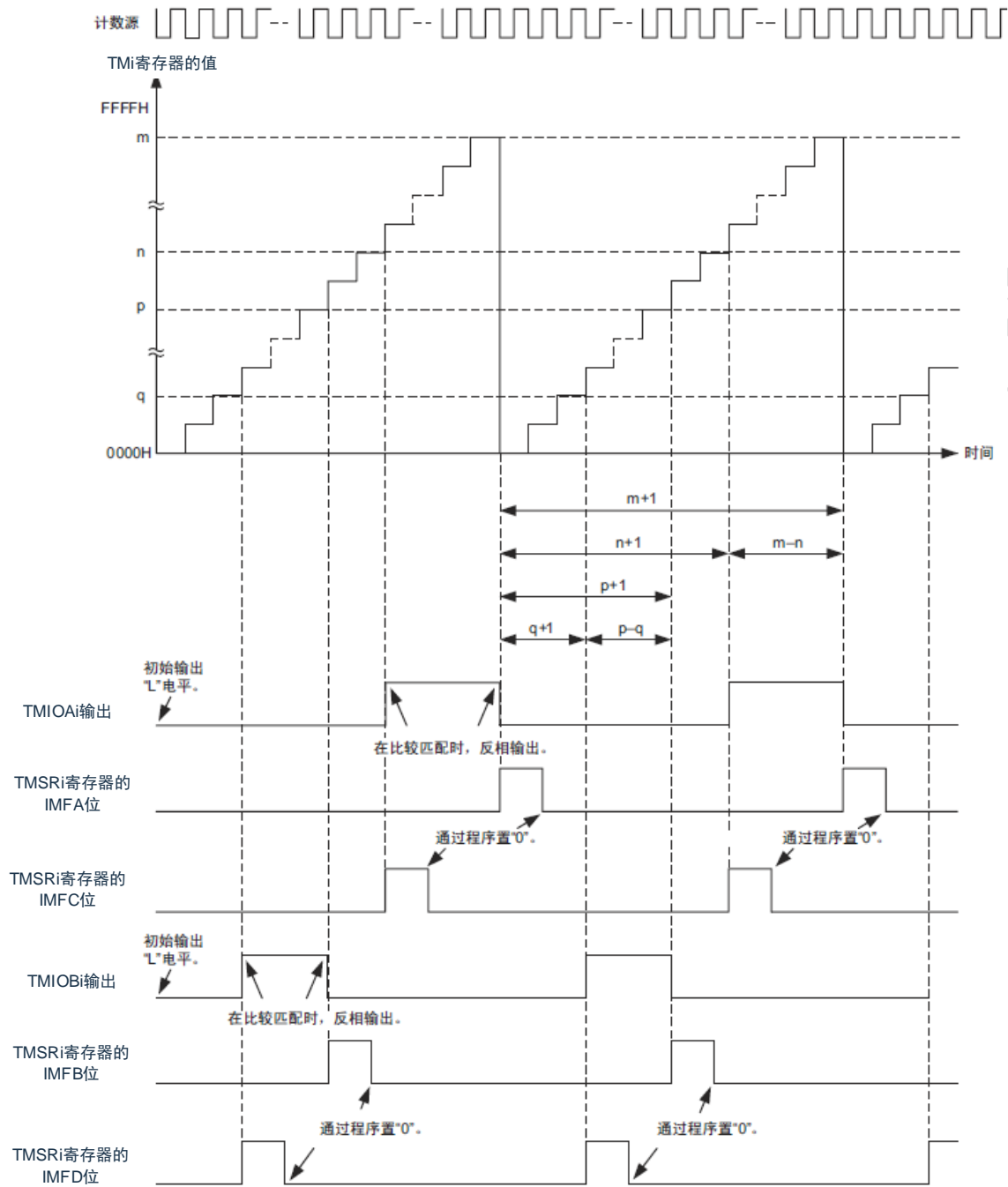


要更改TMGR*C*寄存器和TMGRD*i*寄存器的输出引脚时，必须进行以下的设置：

- 通过TMIORC*i*寄存器的IOj3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TMBFj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*<sub>i</sub>和TMGR*D*<sub>i</sub>分别用于TMIOA*i*引脚和TMIOB*i*引脚的输出控制时的运行例子如图10-53所示。

图10-53: 将TMGR*C*<sub>i</sub>和TMGR*D*<sub>i</sub>分别用于TMIOA*i*引脚和TMIOB*i*引脚的输出控制时的运行例子



备注:  $i=0, 1$

m: TMGR*A*<sub>i</sub>寄存器的设置值

n: TMGR*B*<sub>i</sub>寄存器的设置值

p: TMGR*C*<sub>i</sub>寄存器的设置值

q: TMGR*D*<sub>i</sub>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)。

Draft Version  
Cmsemicon Confidential

### 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<sub>i</sub>引脚(j=B、C、D)和TMGRj<sub>i</sub>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<sub>i</sub>寄存器，因此TMGRA<sub>i</sub>寄存器不能用于其他模式)。

PWM功能的框图和规格分别如图10-55和表10-14所示，PWM功能的运行例子如图10-56和图10-54所示。

图10-55 PWM功能的框图

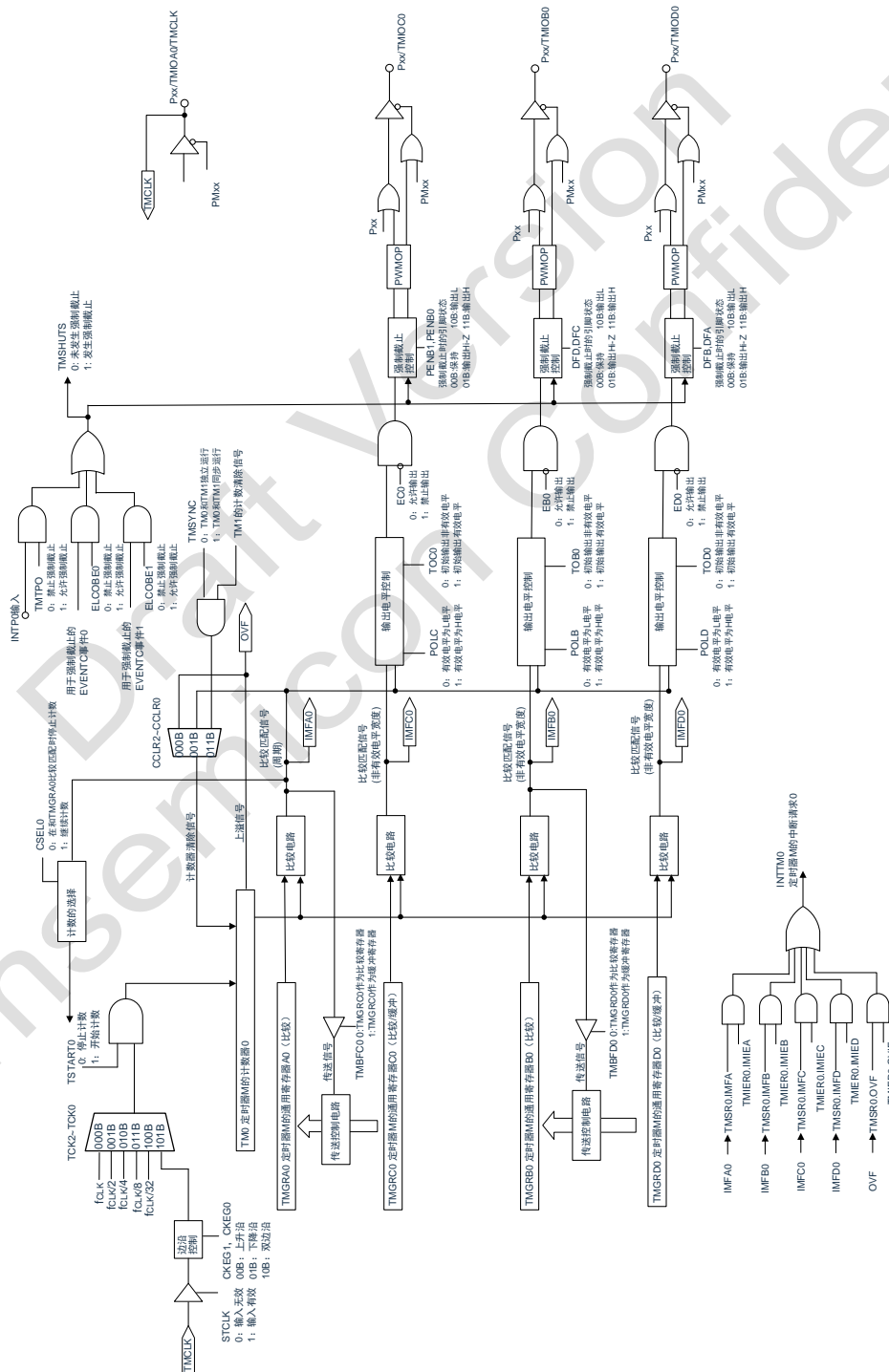


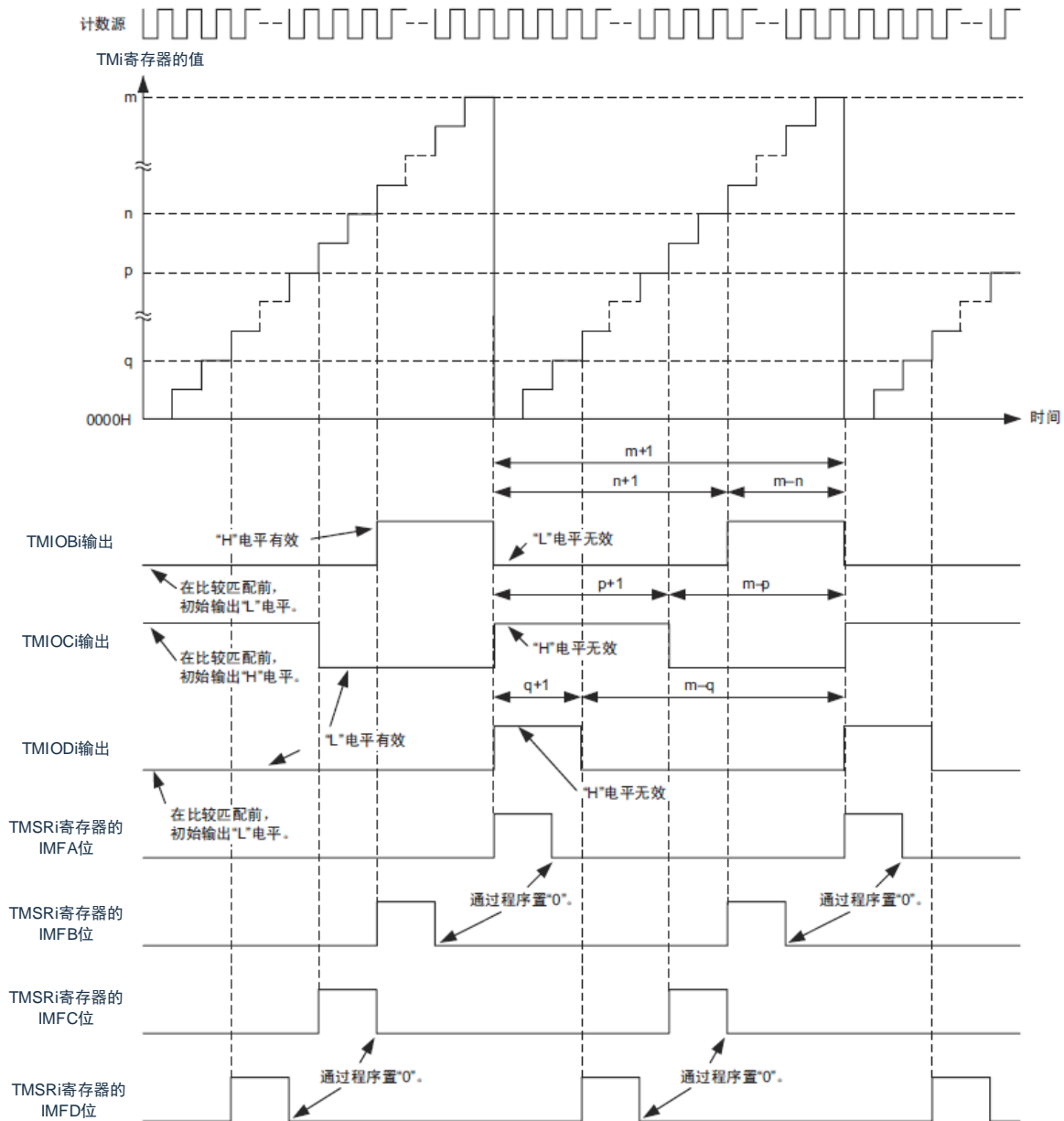
表10-14 PWM功能的规格

项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /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: $1/f_k \times (m+1)$ 有效电平宽度: $1/f_k \times (m-n)$ 无效电平宽度: $1/f_k \times (n+1)$ f <sub>k</sub> : 计数源的频率 m: TMGRA <sub>i</sub> 寄存器的设置值 n: TMGR <sub>j</sub> 寄存器的设置值 
计数开始条件	给TMSTR寄存器的TSTART <sub>i</sub> 位写“1”(开始计数)。
计数停止条件	<ul style="list-style-type: none"> <li>当TMSTR寄存器的CSEL<sub>i</sub>位为“1”时, 给TSTART<sub>i</sub>位写“0”(停止计数)。PWM输出引脚保持停止计数前的输出电平。</li> <li>在TMSTR寄存器的CSEL<sub>i</sub>位为“0”并且发生TMGRA<sub>i</sub>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。</li> </ul>
中断请求的产生时序	<ul style="list-style-type: none"> <li>比较匹配(TM<sub>i</sub>寄存器和TMGR<sub>h</sub>寄存器的内容相同)</li> <li>TM<sub>i</sub>的上溢</li> </ul>
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 <sub>i</sub> 寄存器, 就能读到计数值。
写定时器	能写TM <sub>i</sub> 寄存器。
选择功能	<ul style="list-style-type: none"> <li>通过定时器M<sub>i</sub>进行的1~3个PWM输出引脚的选择 TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚中的1个或者多个引脚</li> <li>各引脚有效电平的选择</li> <li>各引脚初始输出电平的选择</li> <li>同步运行(参照“10.4.3同步运行”)</li> <li>缓冲器运行(参照“10.4.2缓冲器运行”)</li> <li>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”)</li> </ul>

备注: i=0、1, j=B、C、D, h=A、B、C、D

(1) 运行例子

图10-56 PWM 功能的运行例子



备注:  $i=0, 1$

- m: TMGRA<sub>i</sub>寄存器的设置值
- n: TMGRB<sub>i</sub>寄存器的设置值
- p: TMGRCl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TMBFC<sub>i</sub>位和TMBFD<sub>i</sub>位为“0”(TMGRCl和TMGRDi不作为缓冲器运行)。

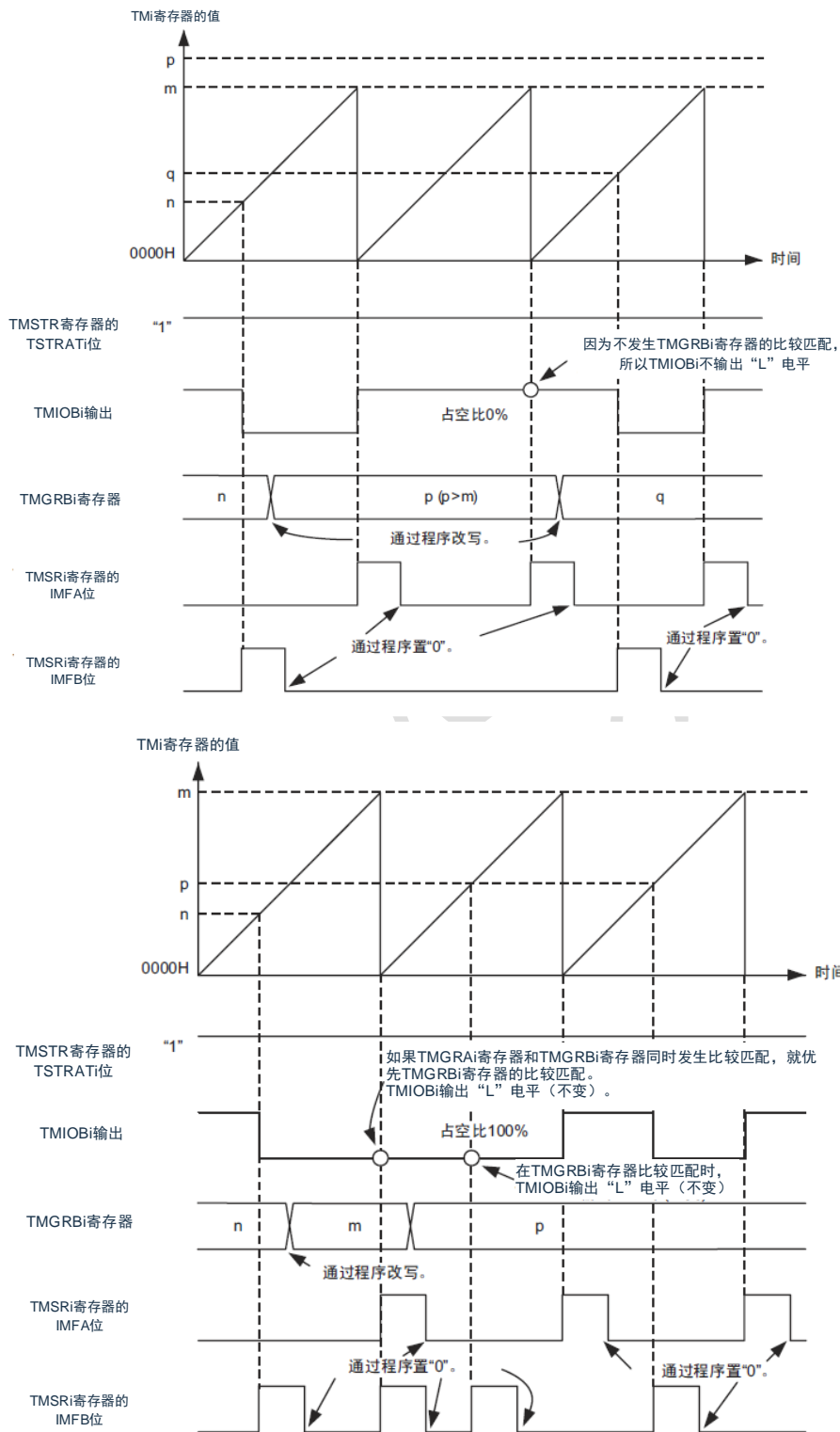
TMOER1寄存器的EB<sub>i</sub>位和EC<sub>i</sub>位为“0”(允许TMIOD<sub>i</sub>和TMIOCI<sub>i</sub>的输出)。

TMOCR寄存器TOB<sub>i</sub>位和TOC<sub>i</sub>位为“0”(无效电平), TOD<sub>i</sub>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10-57 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10-58和图10-59所示，复位同步PWM模式的规格如表10-15所示。

图10-58 复位同步PWM模式的框图

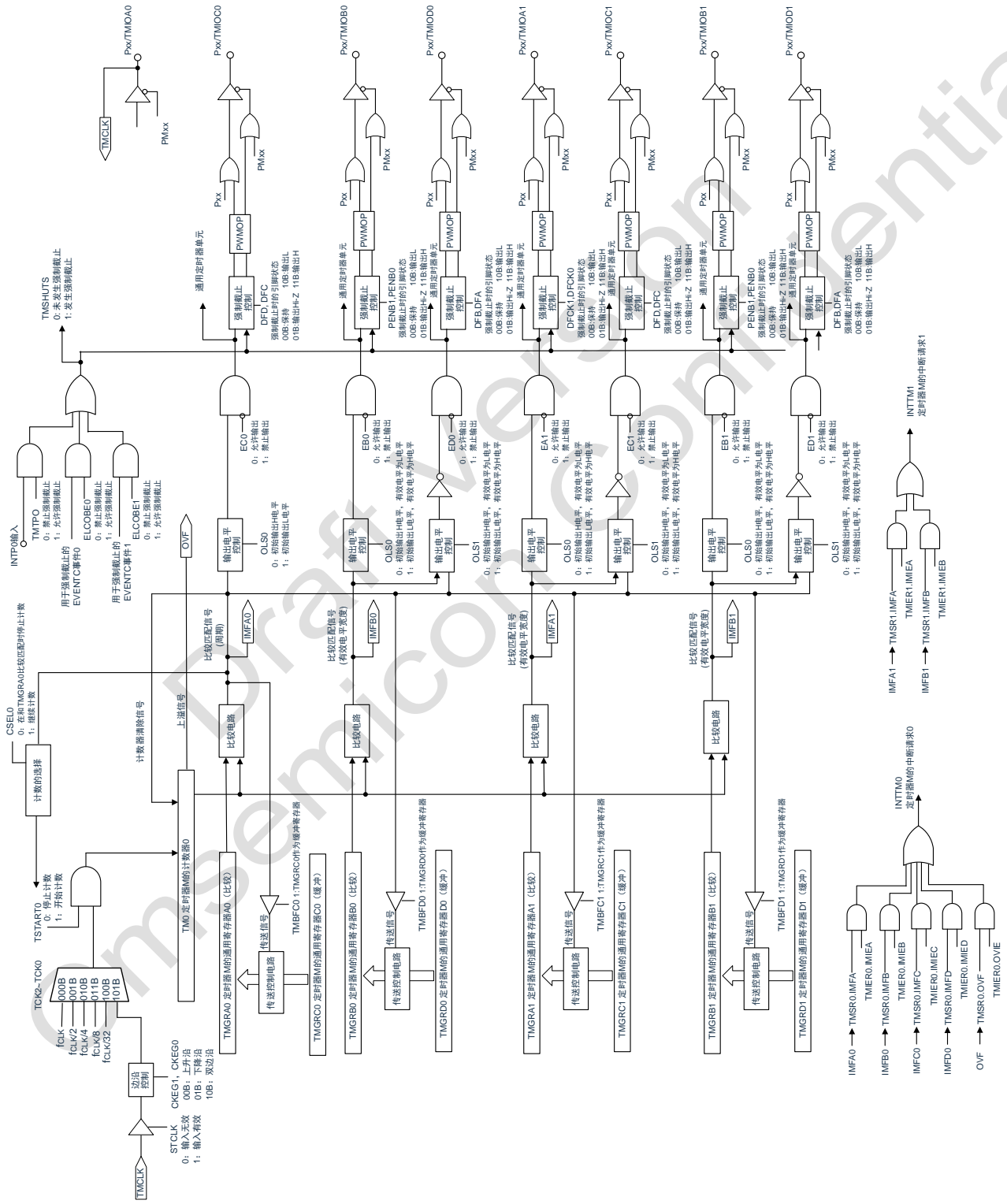


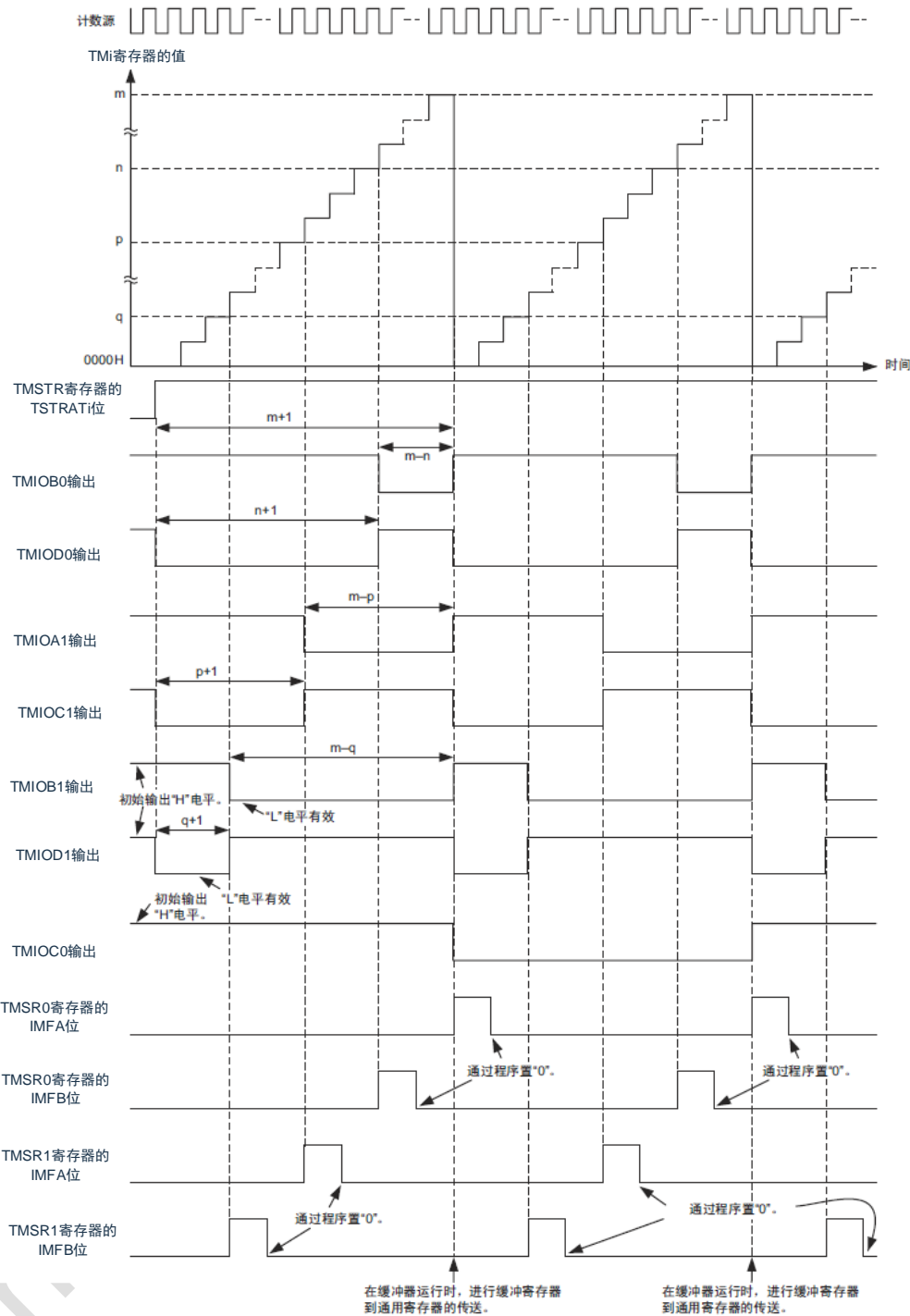
表10-15 复位同步 PWM 模式的规格

项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /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: $1/fk \times (m+1)$ 正相有效电平宽度: $1/fk \times (m-n)$ 反相有效电平宽度: $1/fk \times (n+1)$ 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) 
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"> <li>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。</li> <li>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。</li> </ul>
中断请求的产生时序	<ul style="list-style-type: none"> <li>比较匹配(TM0寄存器和TMGRj0、TMGRA1、TMGRB1寄存器的内容相同)</li> <li>TM0的上溢</li> </ul>
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"> <li>正反相有效电平和初始输出电平的选择</li> <li>缓冲器运行(参照“10.4.2缓冲器运行”)</li> <li>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”)</li> </ul>

备注: j=A、B、C、D

(1) 运行例子

图10-59 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10-60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10-61和图10-63所示。

图10-60 互补PWM模式的框图

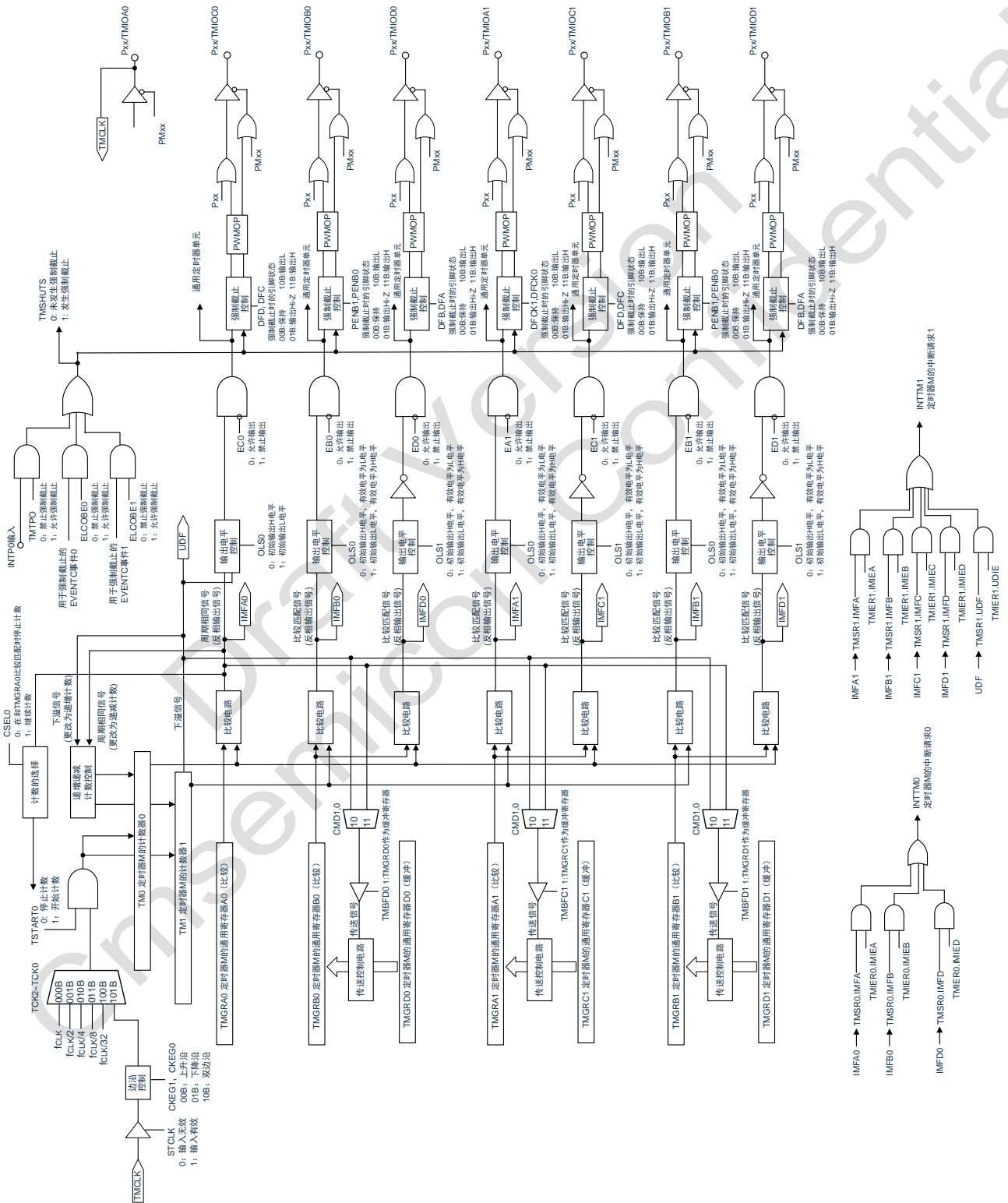


表10-16 互补PWM模式的规格

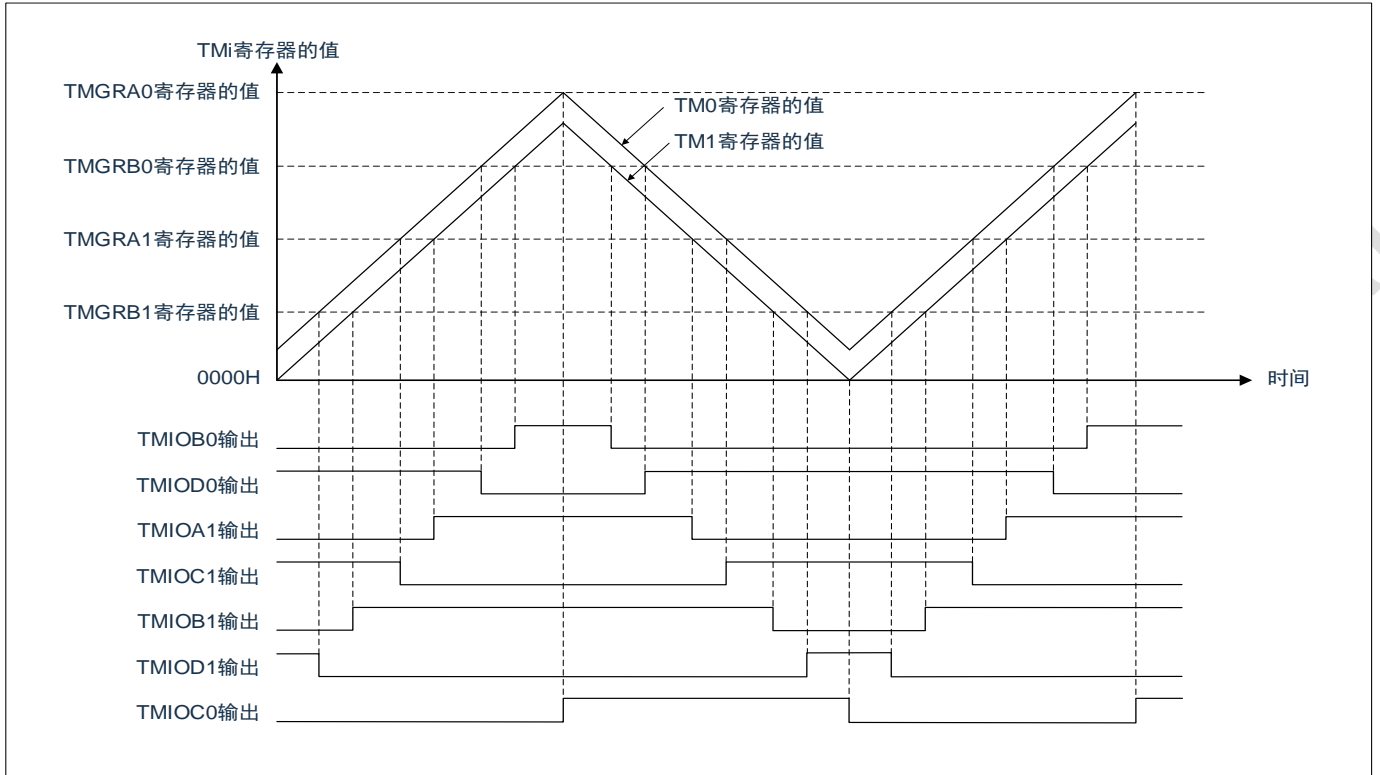
项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /32 TMCLK引脚的外部输入信号(能通过程序选择有效边沿) 必须给TMCR0寄存器的TCK0~TCK2位和TMCR1寄存器的TCK0~TCK2位设置相同的值(相同的计数源)。
计数	递增计数或者递减计数 如果在递增计数的过程中TM0寄存器和TMGRA0寄存器比较匹配, TM0和TM1就都变为递减计数;如果在递减计数的过程中TM1寄存器从“0000H”变为“FFFFH”, TM0和TM1就都变为递增计数。
PWM波形	PWM周期: $1/fk \times (m+2-p) \times 2^z$ 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  (有效电平为“L”的情况)
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"> <li>•比较匹配(TMi寄存器和TMGRji寄存器的内容相同)</li> <li>•TM1的下溢</li> </ul>
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"> <li>•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”)</li> <li>•正反相有效电平和初始输出电平的选择</li> <li>•缓冲寄存器的传送时序的选择</li> </ul>
匹配事件延时功能	可以设置TMGRB0, TMGRA1, TMGRB1寄存器比较匹配事件的单边/双边延时输出

注: 在开始计数后, PWM周期是固定的。

备注: i=0、1, j=A、B、C、D

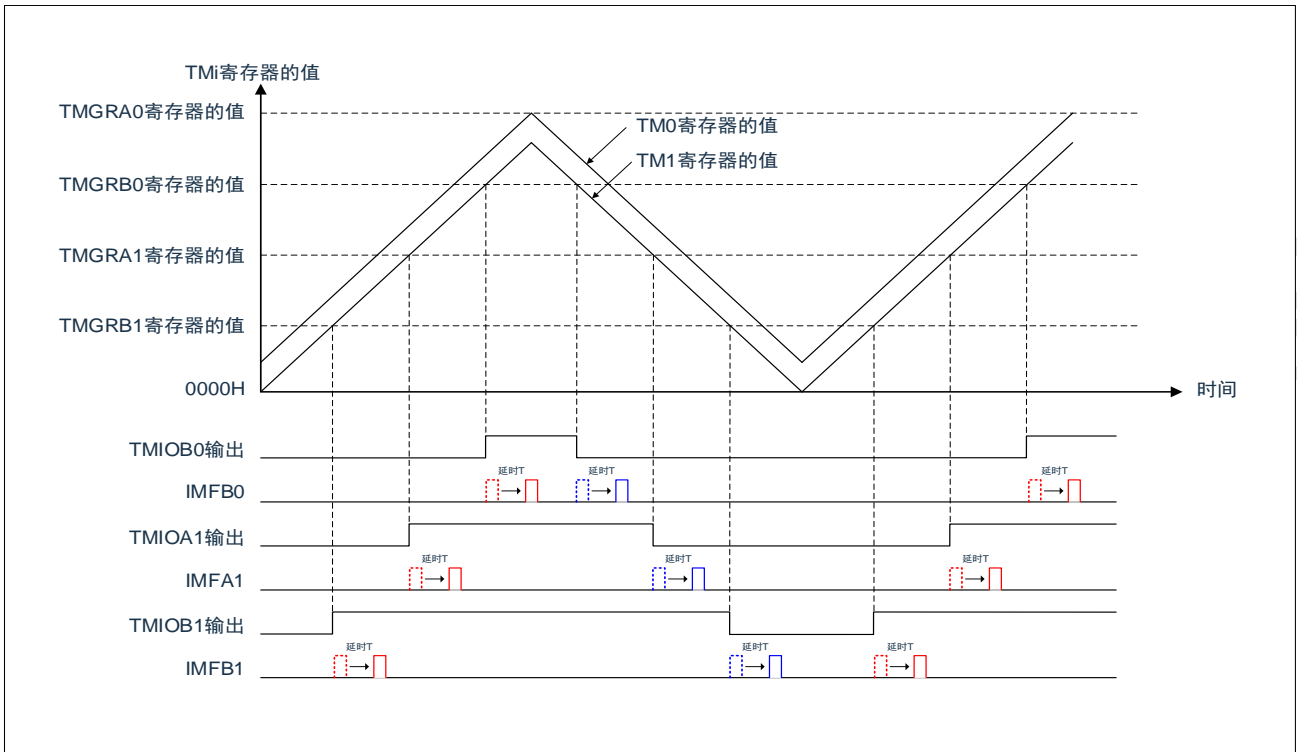
(1) 运行例子

图10-61 互补PWM模式的输出模型



备注: i=0, 1

图10-62 互补PWM模式比较匹配事件（IMFxx）延时时序图

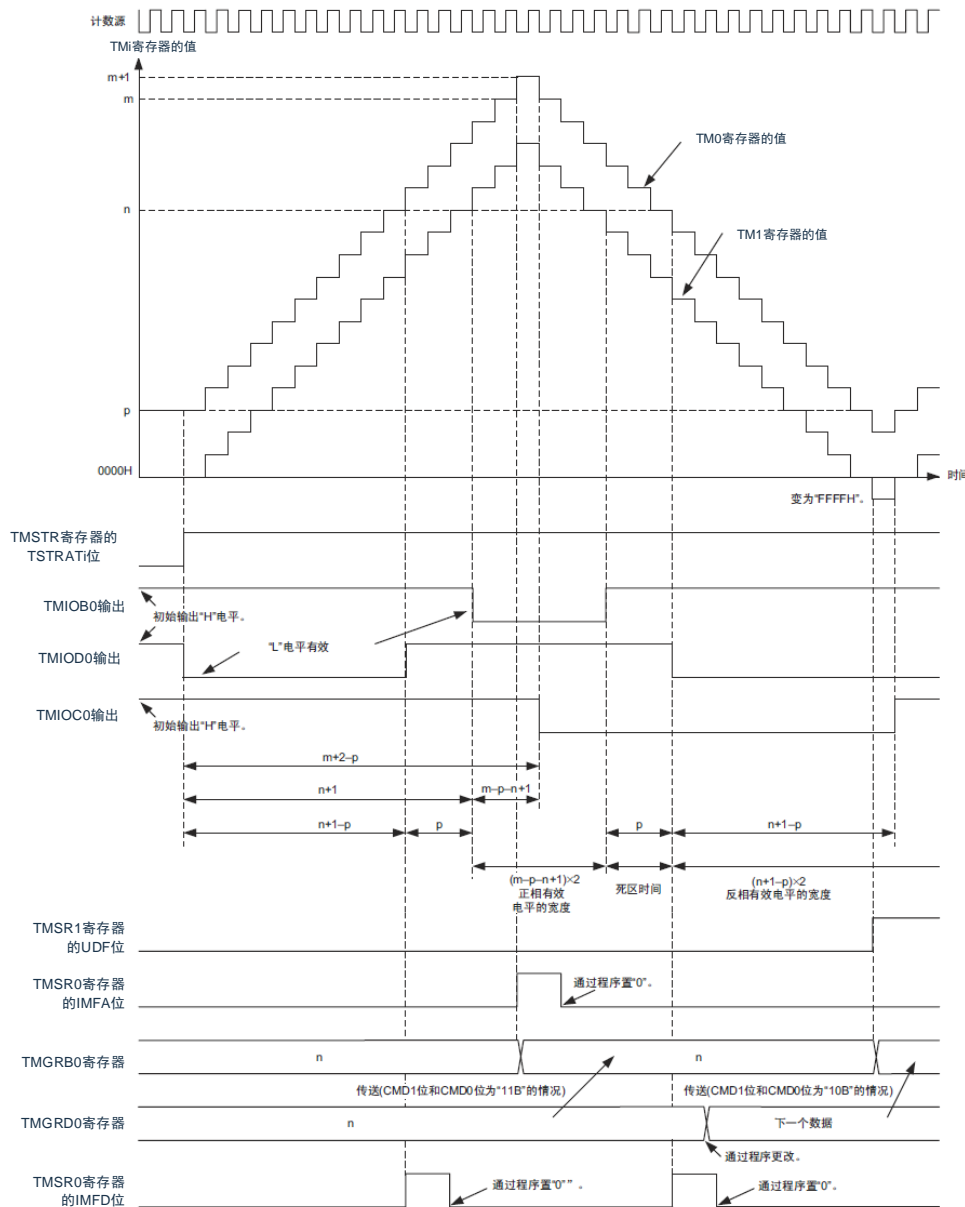


备注：图中实线：红色部分为DSIDE[0]=1的情况，蓝色部分为DSIDE[1]=1的情况；  
 图中虚线：为DSIDE[1: 0]=0的情况；  
 图中延时 $T = DNUM + 1$ 个计数时钟。

上图为比较匹配事件延时时序图，用户可通过设置TMDR寄存器来控制匹配事件延时输出，设置的延时时间必须大于等于1个计数时钟。



图10-63 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、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。

### 10.5.6 PWM3模式

输出2个同周期的PWM波形。

PWM3模式的框图和运行例子分别如图10-64和图10-65所示，PWM3模式的规格如表10-17所示。

图10-64 PWM3模式的框图

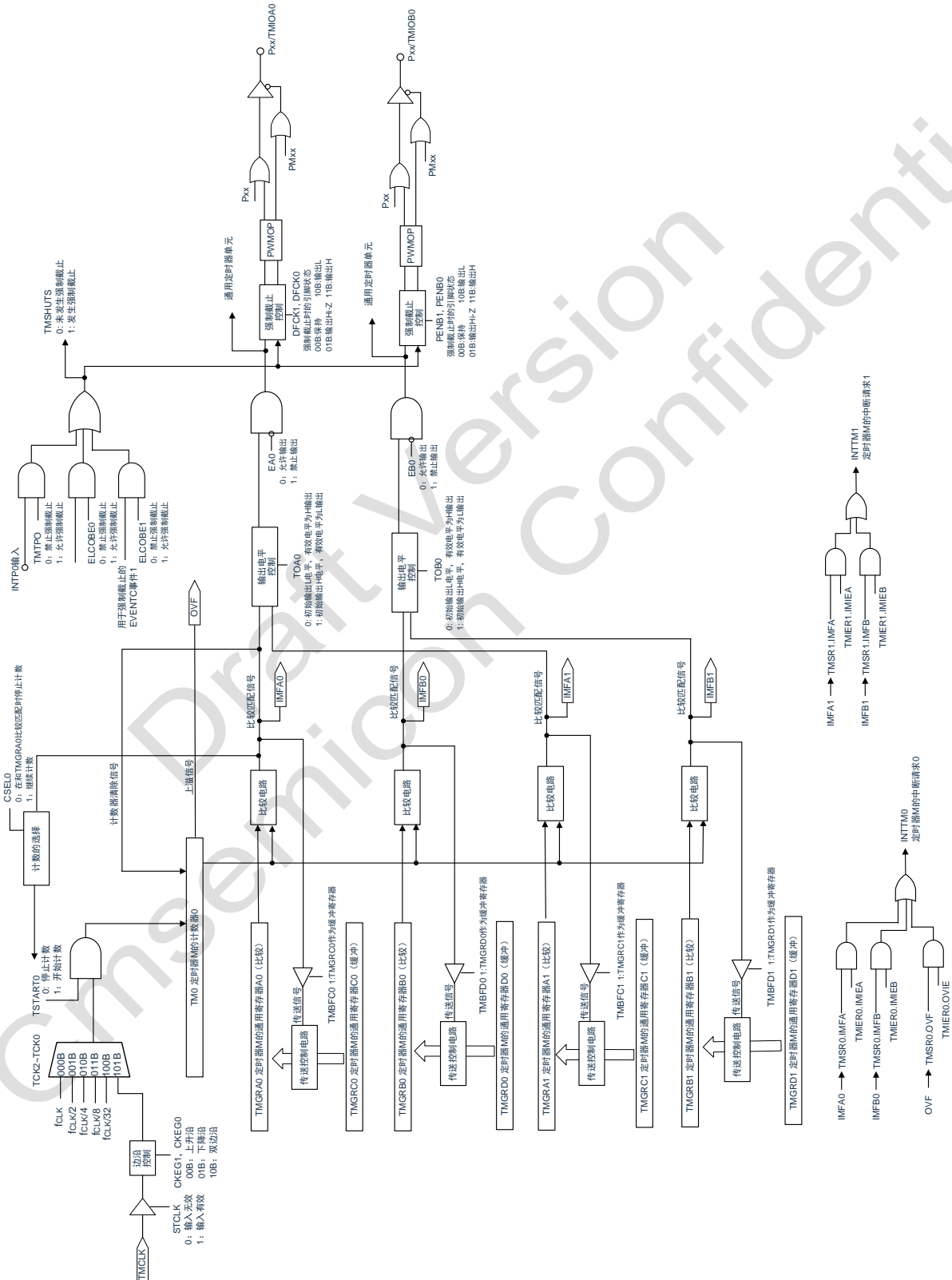
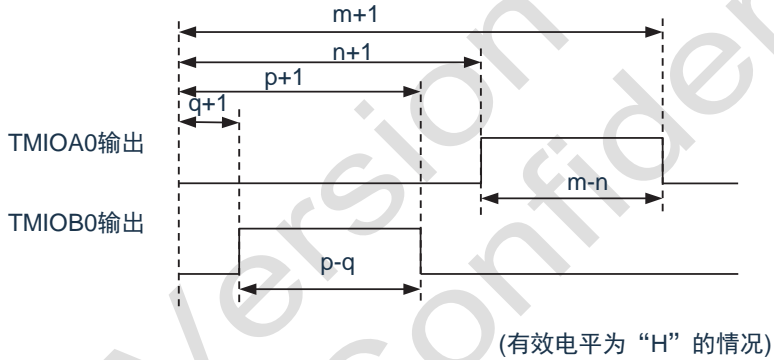


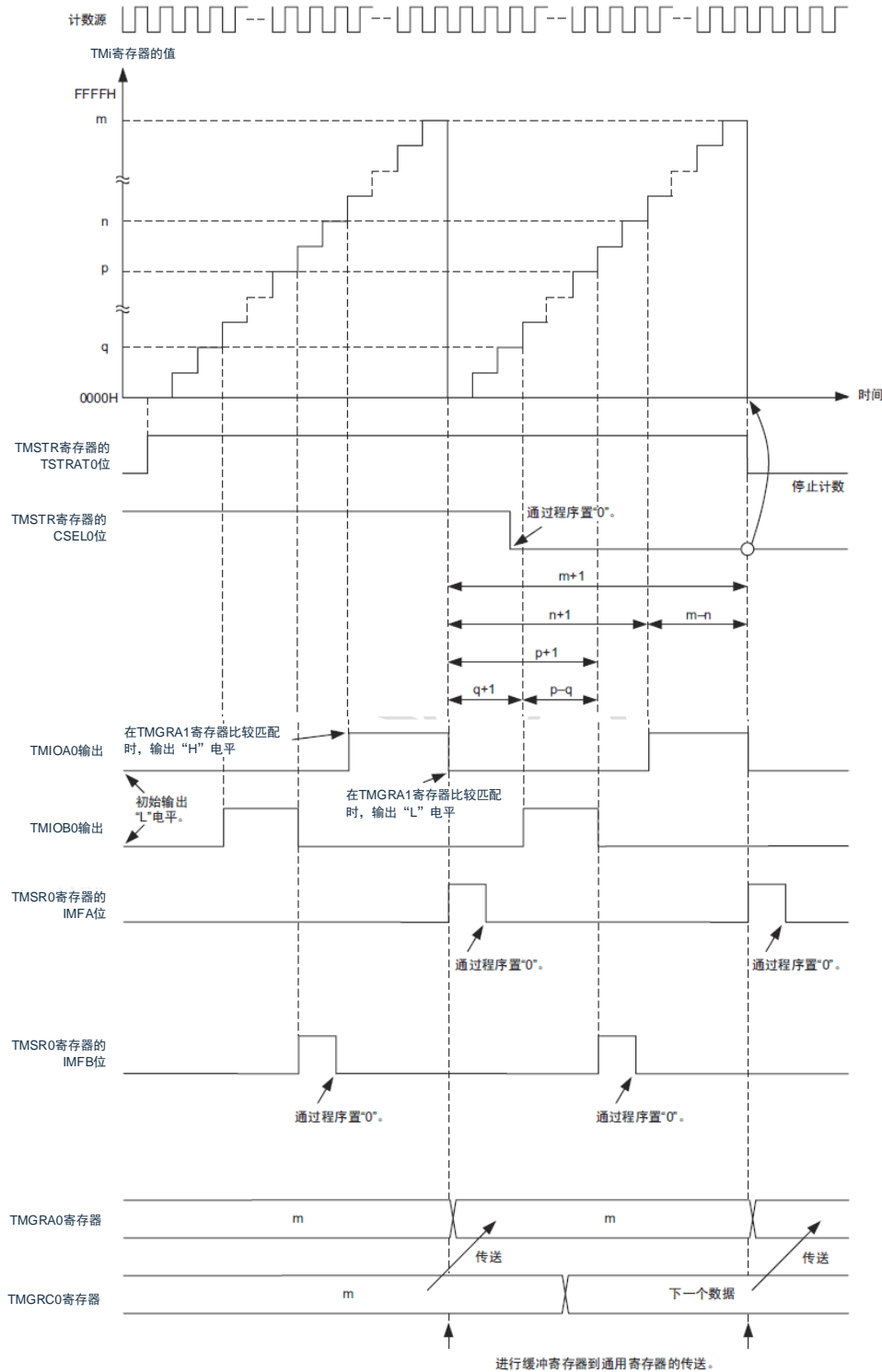
表10-17 PWM3模式的规格

项目	规格
计数源	F <sub>CLK</sub> 、F <sub>CLK</sub> /2、F <sub>CLK</sub> /4、F <sub>CLK</sub> /8、F <sub>CLK</sub> /32
计数	TM0为递增计数(不使用TM1)。
PWM波形	<p>                     PWM周期: <math>1/f_k \times (m+1)</math>                      TMIOA0输出的有效电平宽度: <math>1/f_k \times (m-n)</math>                      TMIOB0输出的有效电平宽度: <math>1/f_k \times (p-q)</math>                      f<sub>k</sub>: 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>  <p>(有效电平为“H”的情况)</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"> <li>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。</li> <li>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。</li> </ul>
中断请求的产生时序	<ul style="list-style-type: none"> <li>比较匹配(TM<sub>i</sub>寄存器和TMGR<sub>j</sub>寄存器的内容相同)</li> <li>TM0的上溢</li> </ul>
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"> <li>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”)</li> <li>各引脚有效电平的选择</li> <li>缓冲器运行(参照“10.4.2缓冲器运行”)</li> </ul>

备注: i=0、1, j=A、B、C、D

(1) 运行例子

图10-65 PWM3模式的运行例子



备注:  $j=A, B$

- 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

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）

Draft Version  
Cmsemicon Confidential

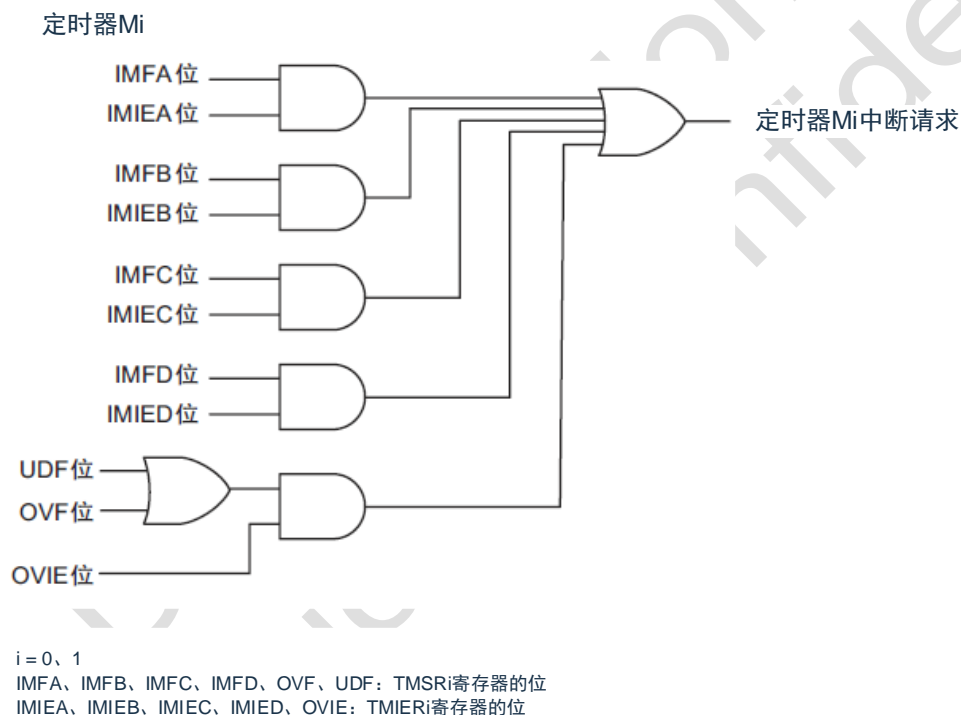
## 10.6 定时器M中断

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10-66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(IF27L)	TMMK0(MK27L)
定时器M1	TMSR1	TMIER1	TMIF1(IF28L)	TMMK1(MK28L)

图10-66 定时器M中断的框图



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)，所以和其他可屏蔽中断有以下的不同：

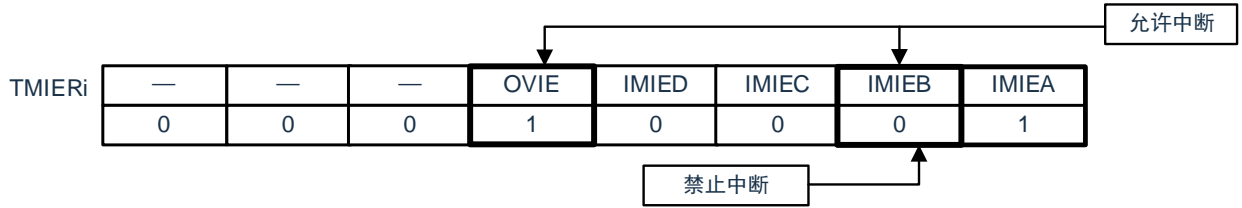
- 当TMSRi寄存器的位为“1”并且对应的TMIERi寄存器的位为“1”(允许中断)时，IF27L、IF28L寄存器的TMIFi位就变为“1”(有中断请求)。
- 当TMIERi寄存器的多个位为“1”时，必须通过TMSRi寄存器判断是哪个请求源产生的中断。
- 因为即使接受中断，TMSRi寄存器的各位也不会自动变为“0”，所以必须在中断程序中将这些位置“0”。
- 要将定时器M的某个中断源的状态标志(以下称为“对象状态标志”)置“0”时，如果该中断被定时器M中断允许寄存器i(TMIE Ri)设置为禁止中断，就必须用以下(a)~(c)的任意一种方法置“0”。

(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(TMIE Ri)的状态



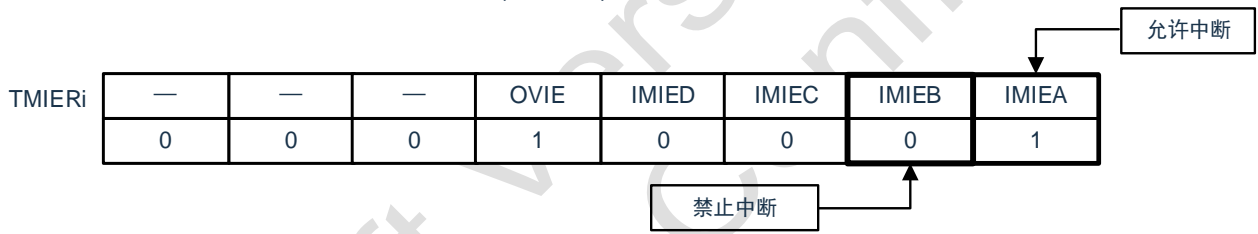
- 定时器M状态寄存器i(TMSRi)的状态



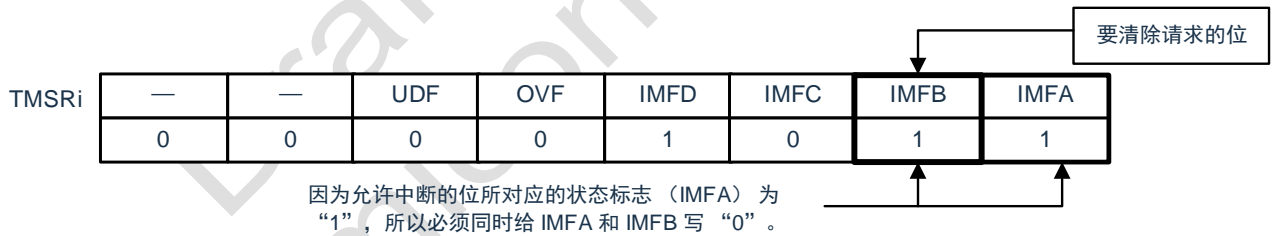
(c) 当定时器M中断允许寄存器i(TMIERi)中有被置“1”(允许)的位并且该位允许的中断源状态标志为“1”时，必须同时给此状态标志和对象状态标志写“0”。

(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TMIOj<sub>i</sub>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TMIOj<sub>i</sub>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TMIOj <sub>i</sub> 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24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TSTART<sub>i</sub>位( $i=0、1$ )置“0”(停止计数)。
  - 更改TMCR<sub>i</sub>寄存器的TCK0~TCK2位。

### 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TMIO<sub>j</sub>引脚( $j=A、B、C、D$ )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TMGR<sub>j</sub>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TMTSTART<sub>i</sub>位为“0”(停止计数)时给TMIO<sub>j</sub>引脚输入TMIOR<sub>k</sub>寄存器的IO<sub>j0</sub>位和IO<sub>j1</sub>位选择的边沿，就在TMIO<sub>j</sub>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 TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚的设置步骤( $i=0、1$ )

- 在复位后，TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚的复用I/O端口用作输入端口。
- 要从TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>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TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚置为允许输出(TMOER1寄存器)。
  - 将TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚对应的端口寄存器的位置“0”。
  - 将TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚对应的端口模式寄存器的位设置为输出模式(从TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚开始输出)。
  - 开始计数(将TSTART0位和TSTART1位置“1”)。
- 要将TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>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TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>引脚对应的端口模式寄存器的位设置为输入模式(从TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>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TMIOA<sub>i</sub>、TMIOB<sub>i</sub>、TMIOC<sub>i</sub>、TMIOD<sub>i</sub>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]

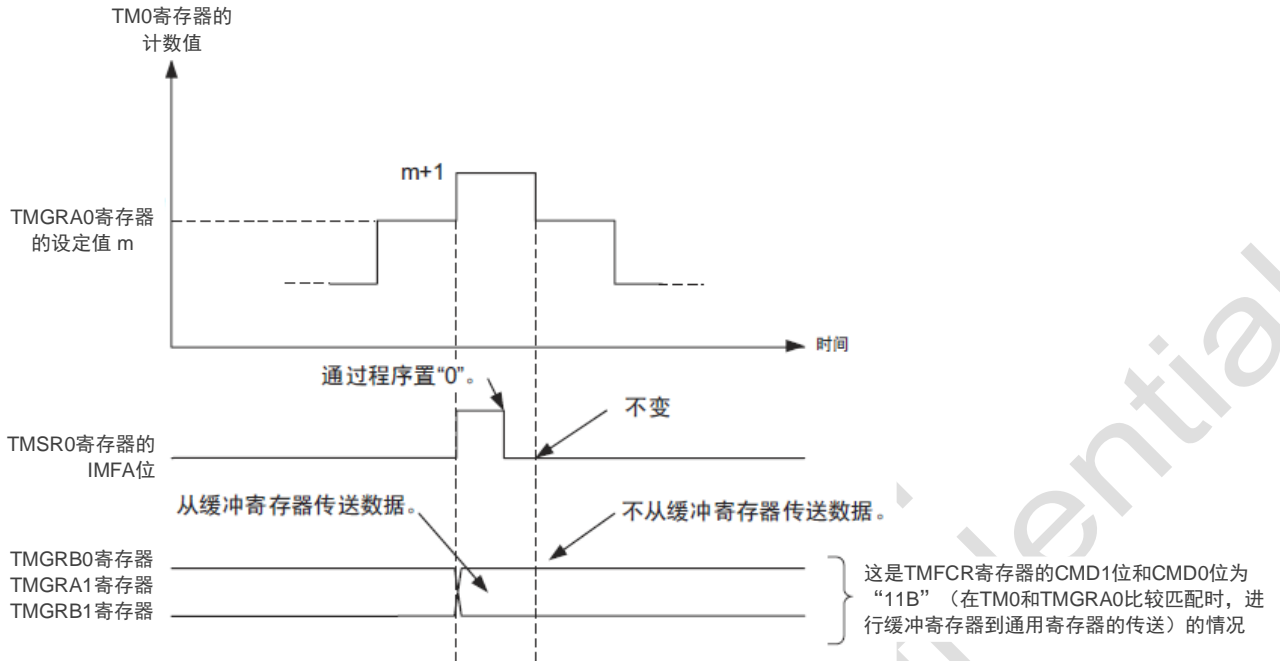
- 1) 将TMSTR寄存器的TSTART0位和TSTART1位都置“0”(停止计数)。
- 2) 将TMFCR寄存器的CMD1位和CMD0位置“00B”(定时器模式、PWM模式和PWM3模式)。
- 3) 将CMD1位和CMD0位置“10B”或者“11B”(互补PWM模式)。
- 4) 重新设置定时器M的其他相关寄存器。

[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)

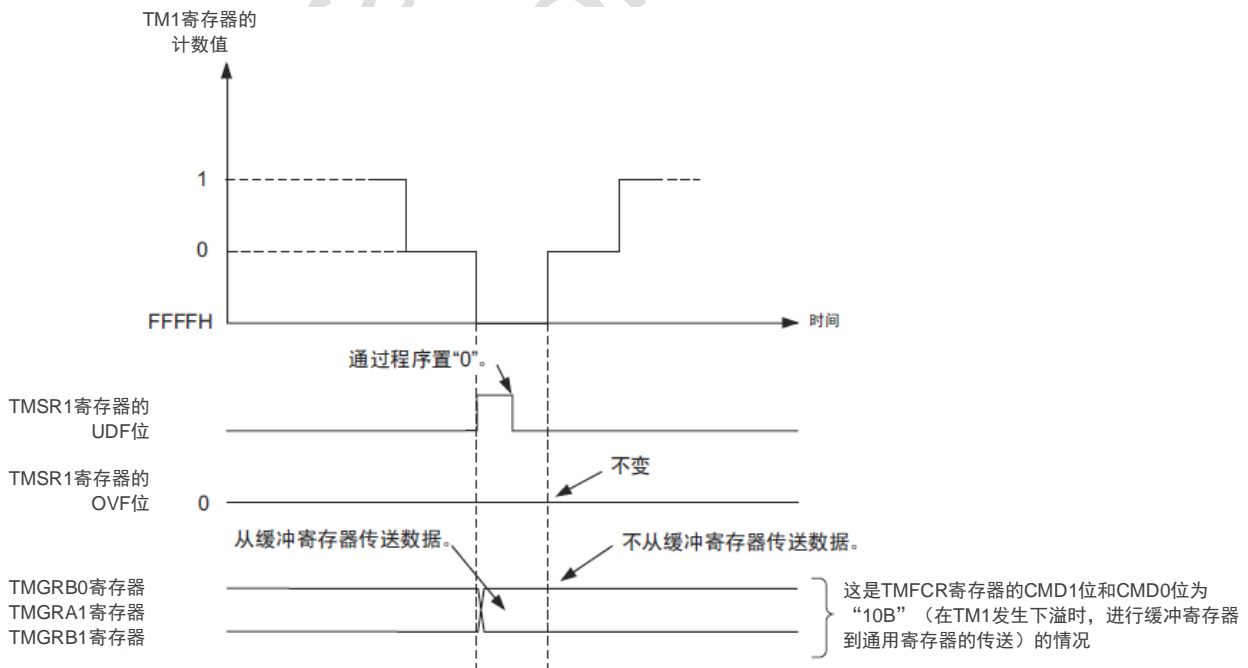
- 不能在运行过程中写TMGRA0、TMGRB0、TMGRA1、TMGRB1寄存器。
- 要更改PWM波形时，必须通过缓冲器运行将TMGRD0、TMGRC1、TMGRD1寄存器的写入值传送到TMGRB0、TMGRA1、TMGRB1寄存器。但是，在写TMGRD0、TMGRC1和TMGRD1时，必须在先将TMBFD0位、TMBFC1位和TMBFD1位置“0”(通用寄存器)，然后给这些寄存器写数据。此后，能将TMBFD0位、TMBFC1位和TMBFD1位置“1”(缓冲寄存器)。不能更改PWM周期。
- 假设TMGRA0寄存器的设置值为m，则TM0寄存器在从递增计数变为递减计数时以m-1, m, m+1, m, m-1的顺序计数。
- 在进行m到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10-67 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10-68 互补PWM模式中的TM1发生下溢时的运行例子



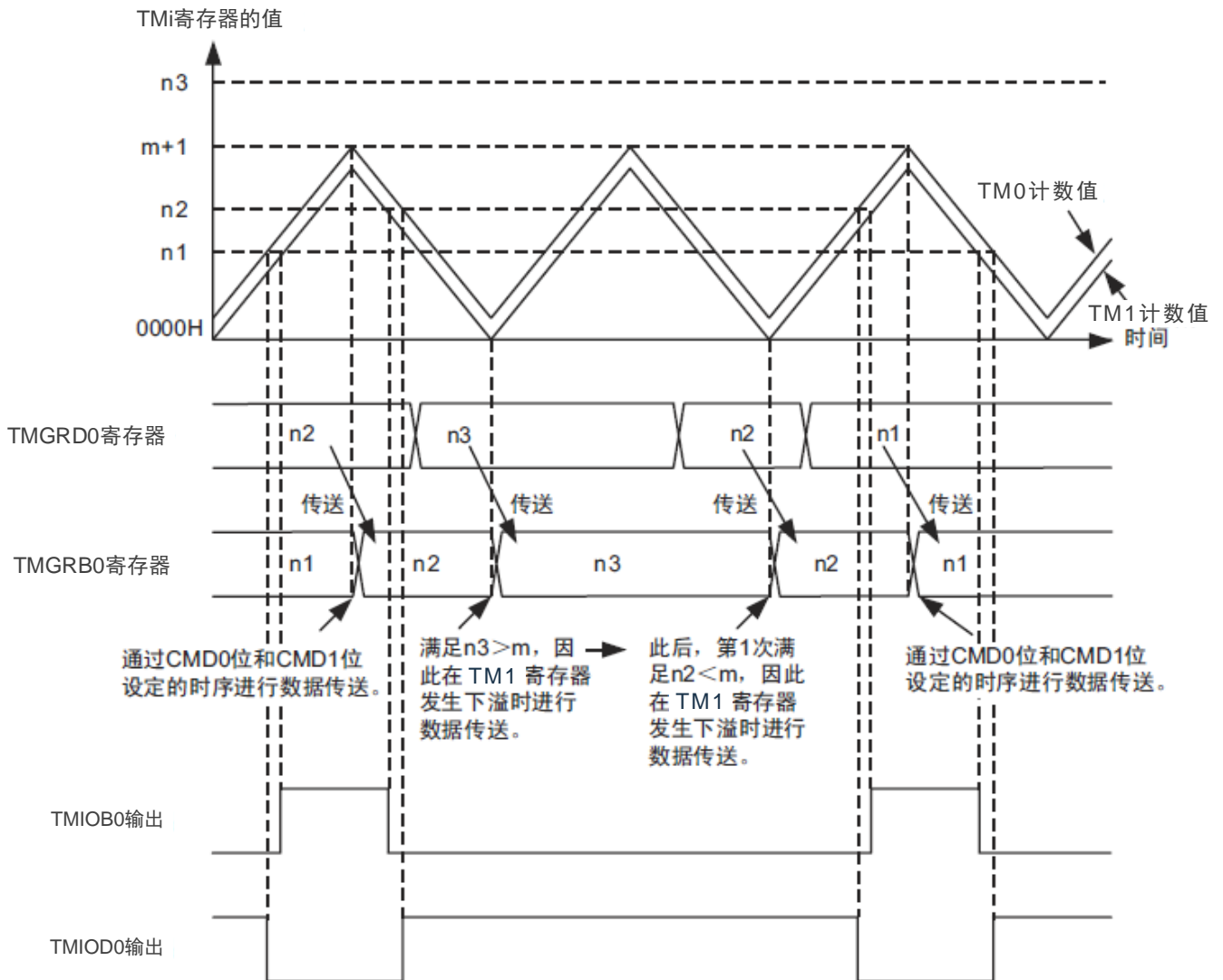
- 必须通过TMFCR寄存器的CMD0位和CMD1位选择缓冲寄存器到通用寄存器的数据传送时序。但是，在0%占空比和100%占空比的情况下，与CMD0位和CMD1位的值无关，为以下的传送时序。

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10-69 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。

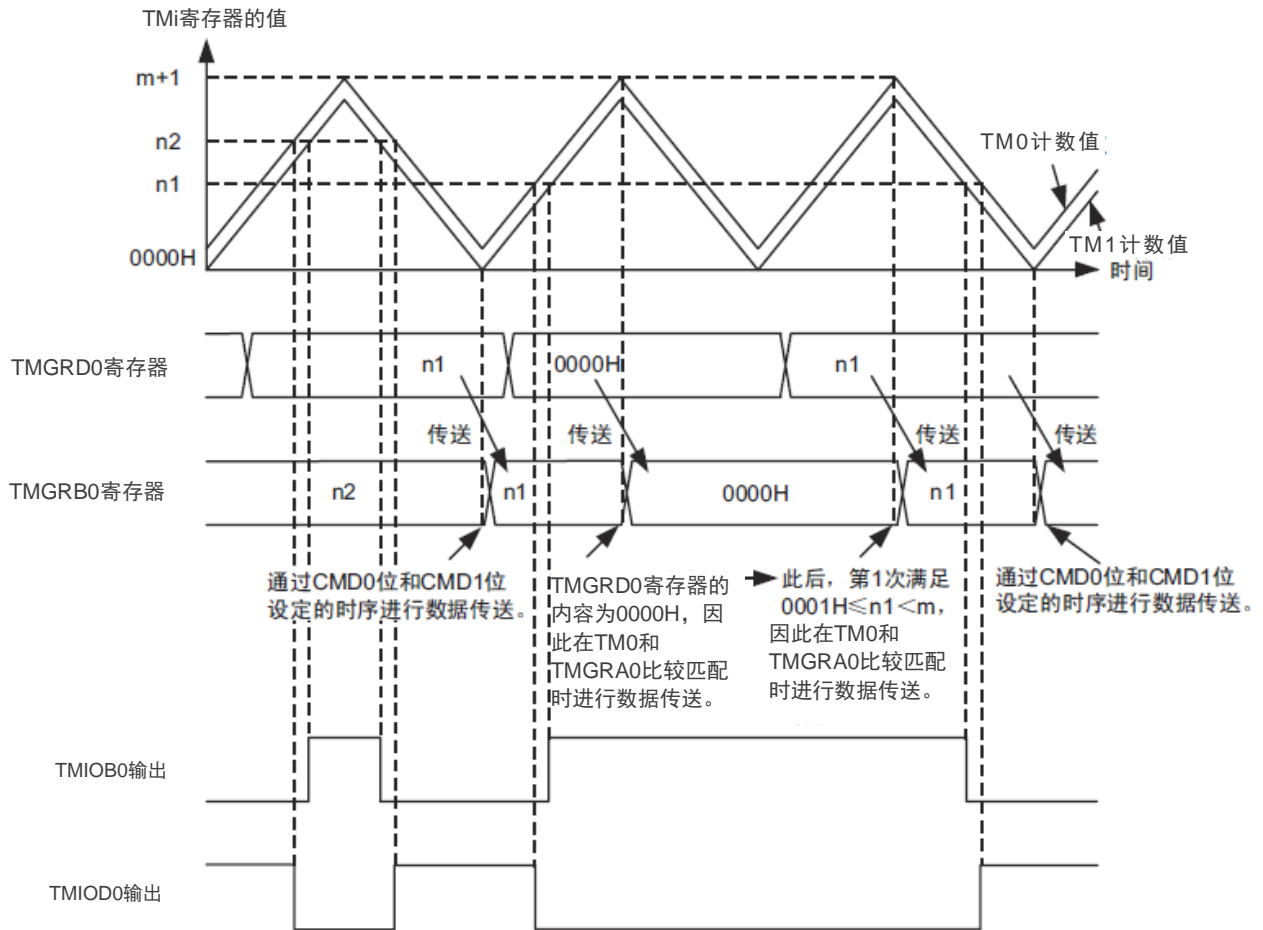
但是，不能用缓冲寄存器的初始值“FFFFH”设置正相100%占空比和反相0%占空比的输出。也不能直接将正相100%占空比和反相0%占空比的输出改为正相0%占空比和反相100%占空比的输出。

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( $0001H \leq \text{设置值} < \text{TMGRA0寄存器的值}$ ), 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10-70 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( $\text{TM0寄存器的值} \leq \text{设置值} \leq (\text{TMGRA0的值} - \text{TM0寄存器的值})$ )。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, INTP0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 INTP0 输入	EVENTC 的事件输入 INTP0 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：

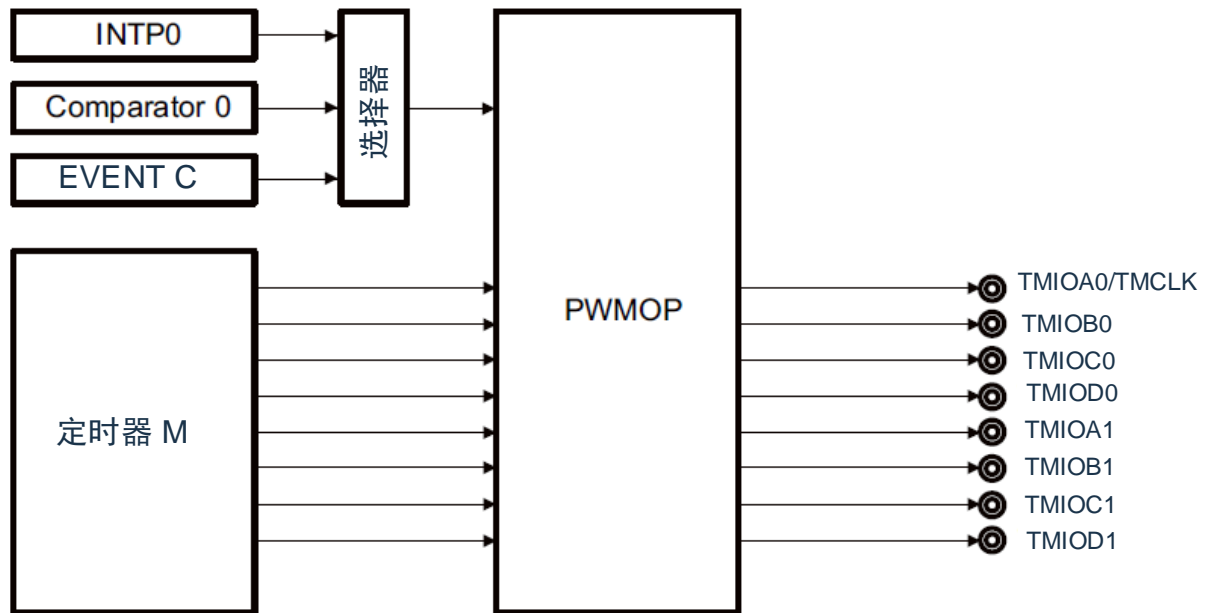
可以选择比较器0的输出，INTP0输入以及EVENTC的事件输入作为输出强制截止源。

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10-71 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10-72 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 <sup>注1</sup>
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 <sup>注2, 3</sup>
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 <sup>注2, 4, 5</sup>
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 <sup>注6</sup>

HZ_REL	软件解除时: 输出强制截止解除控制 <sup>注7</sup>
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 <sup>注8</sup>
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 <span style="float: right;">注9</span>
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10-73 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, 仅为同一引脚设置需要的TMIOji(j=A,B,C,D;i=0,1)功能作为输出。

例: 使用PIOR2选择P17作为TMIOD0引脚, 并且通过设置TMOER1允许TMIOD0输出时, 必须根据需要设置DFD0n(n=0,1), 同时将DFA0n(n=0,1)设置为0(禁止TMIOA0强制截止功能)。

## (3) PWMOP强制截止控制寄存器1(OPDF1)

图10-74 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, 仅为同一引脚设置需要的TMIOji(j=A,B,C,D;i=0,1)功能作为输出。

例: 使用PIOR3选择P10作为TMIOD1引脚, 并且通过设置TMOER1允许TMIOD1输出时, 必须根据需要设置DFD1n(n=0,1), 同时将DFB1n(n=0,1)设置为0(禁止TMIOB1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10-75 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10-76: 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 <sup>注1, 2</sup>
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<sub>ji</sub>(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10-78。

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。

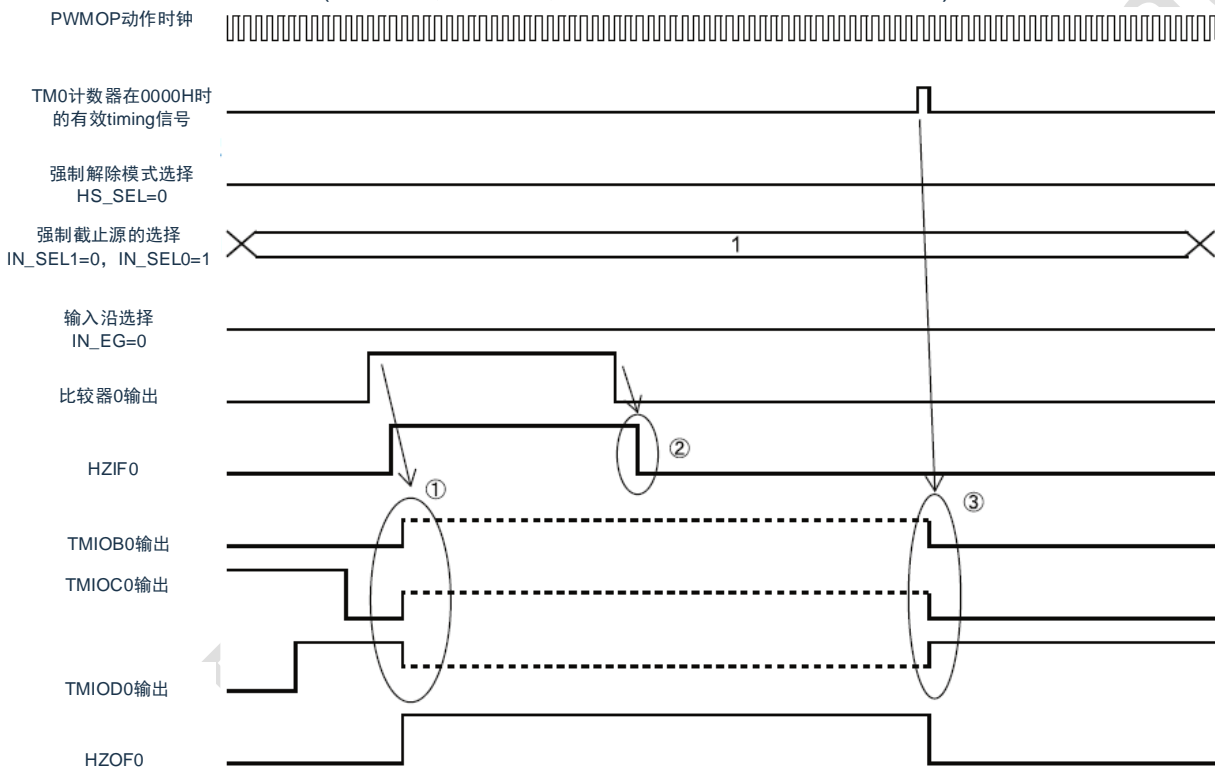
### 10.8.3.2 硬件解除(HS\_SEL=0)

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

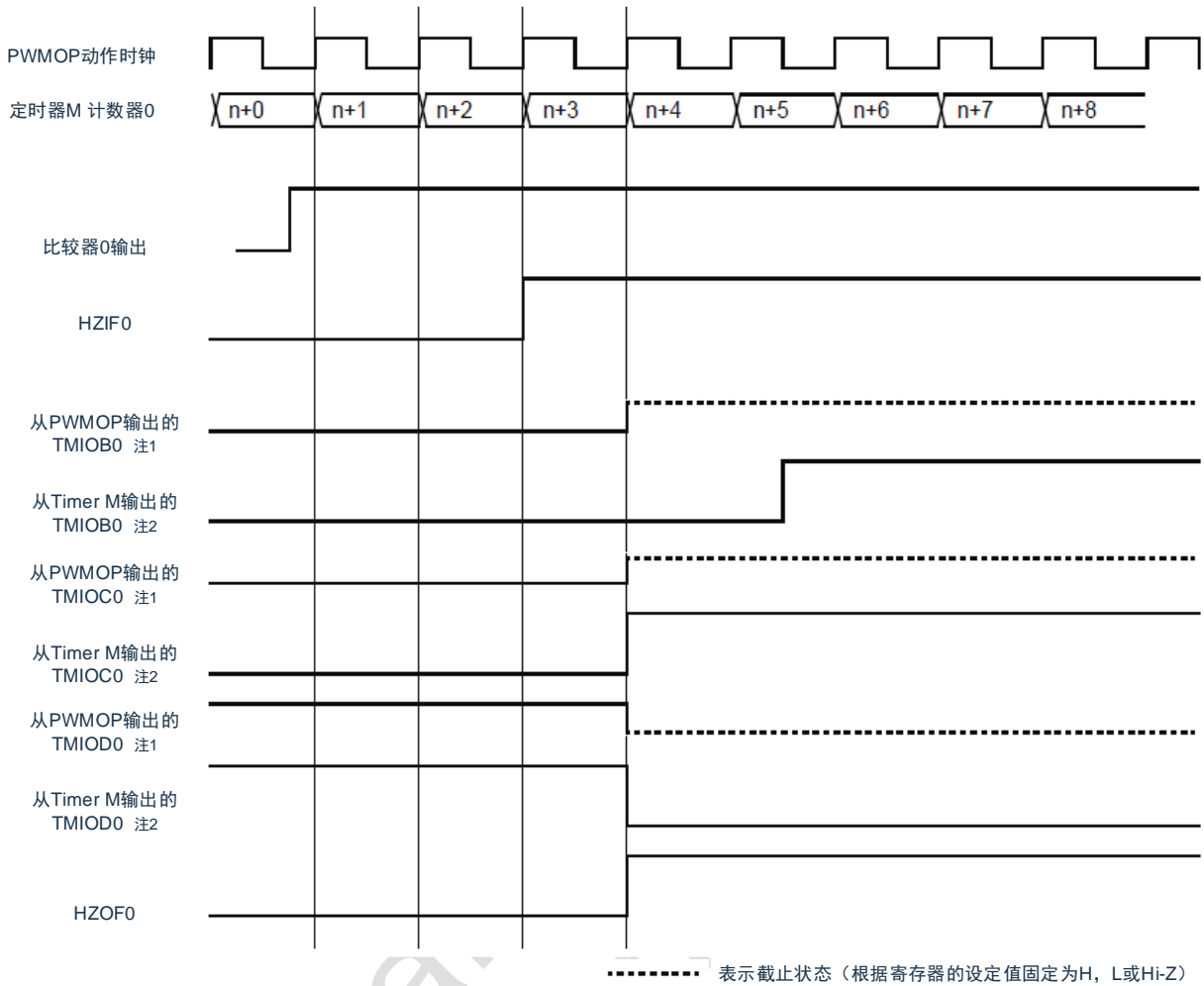
- 定时器M工作在输出比较功能，PWM功能或者PWM3模式下时，解除事件发生后，当TM0的计数值为0000H时，TMIOj0(j=A,B,C,D)的强制截止解除。当TM0的计数值为0000H时，TMIOj1(j=A,B,C,D)的强制截止解除。
- 定时器M工作复位同期PWM模式下时，解除事件发生后，当TM0的计数值为0000H时，所有的TMIO引脚的强制截止全部解除。

图10-77 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。

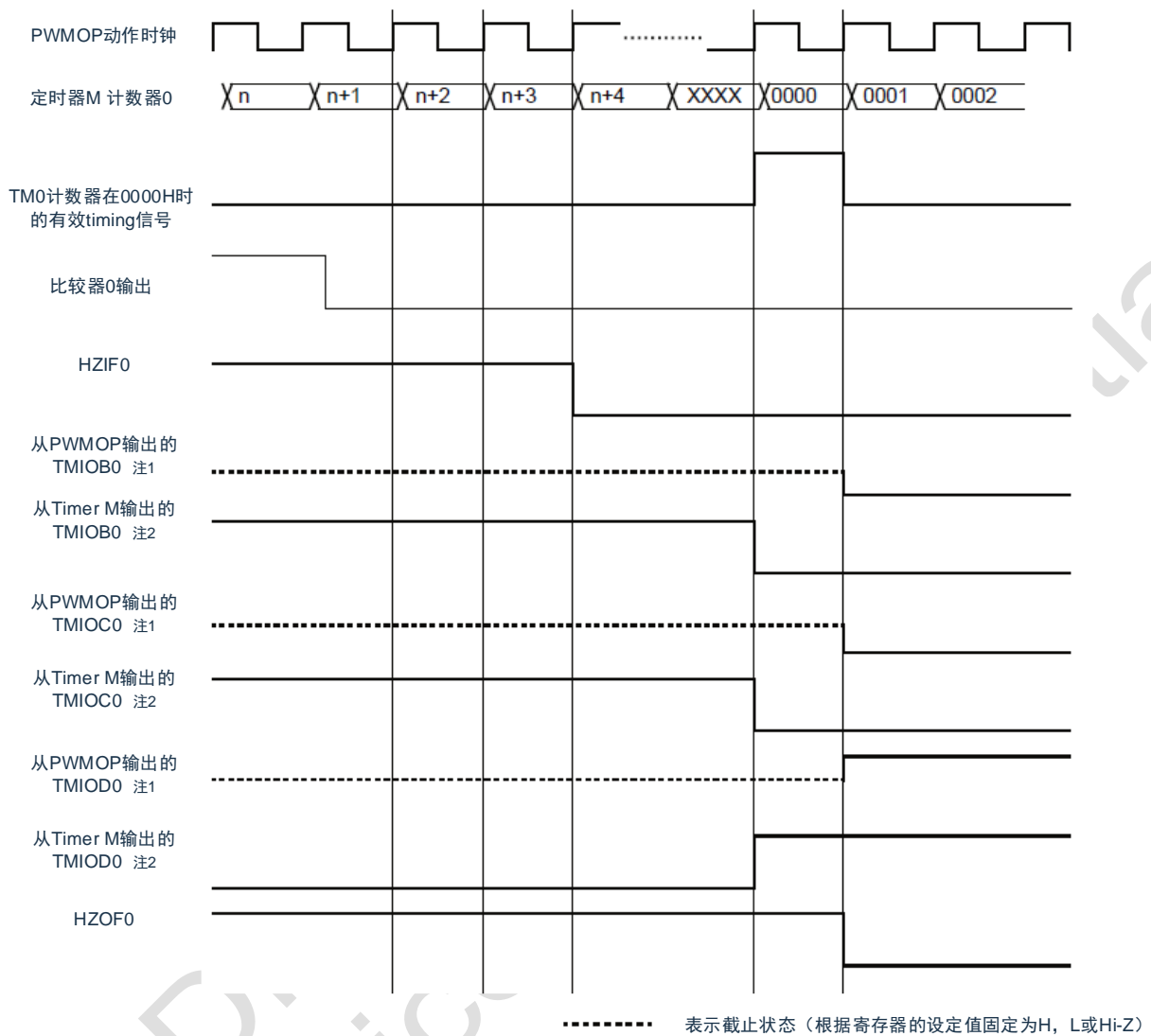
图10-78 强制截止的详细时序图



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10-79 强制截止解除的详细时序图(定时器M的计数源=Fclk)

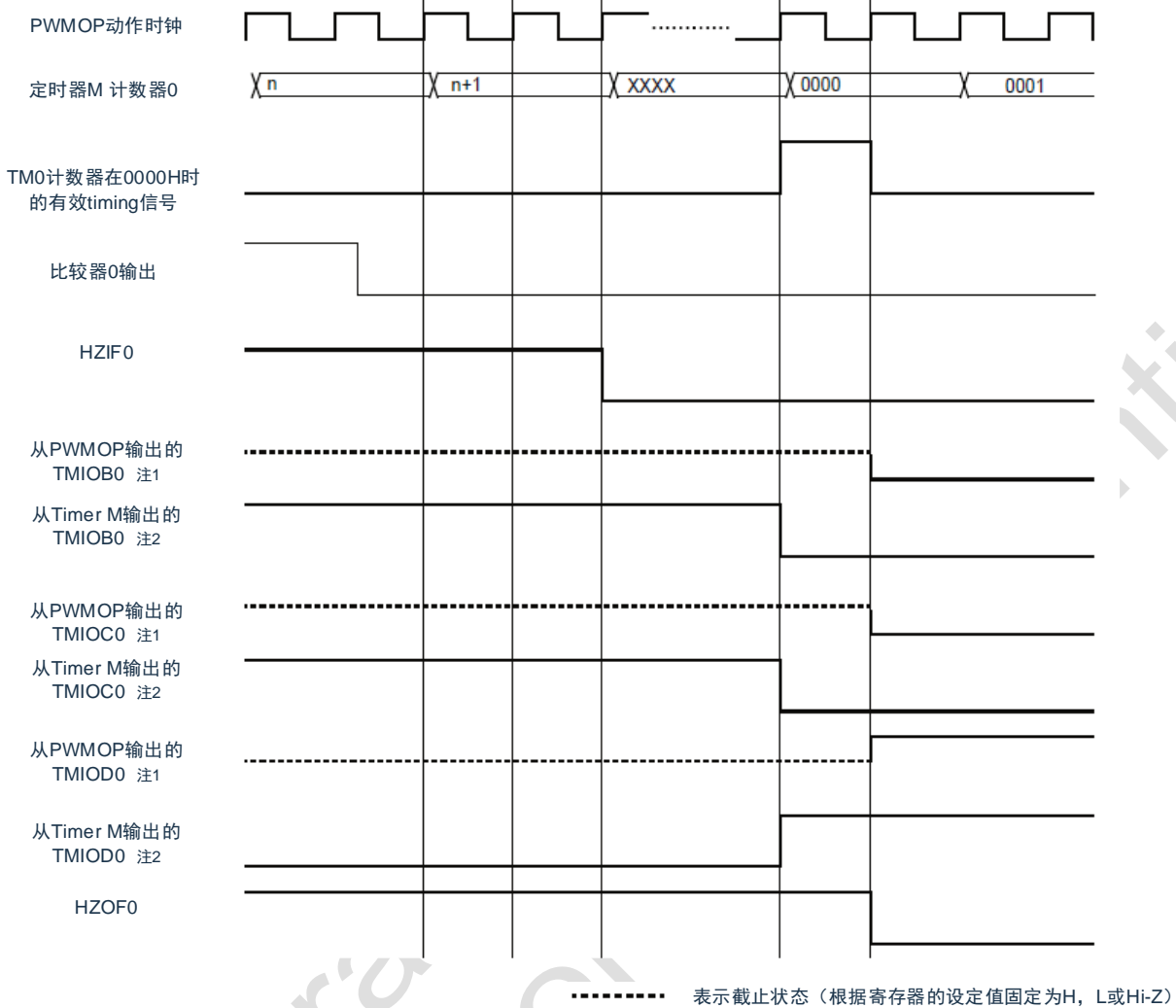


注1：从PWMOP输出的TMIO\*(\*=B~D)表示复用定时器M功能的引脚的状态。

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10-80 强制截止解除的详细时序图(定时器M的计数源=Fclk/2)



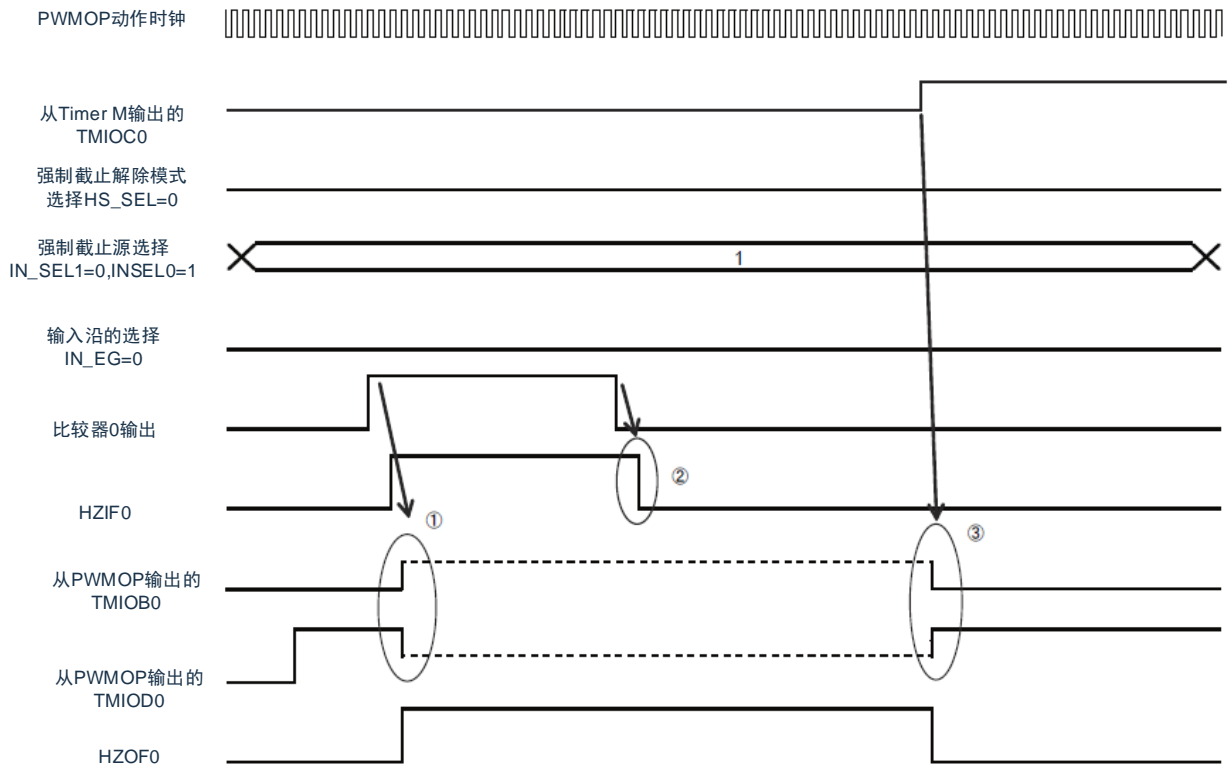
注1: 从PWMOP输出的TMIO\*(\*=B~D)表示复用定时器M功能的引脚的状态。

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。

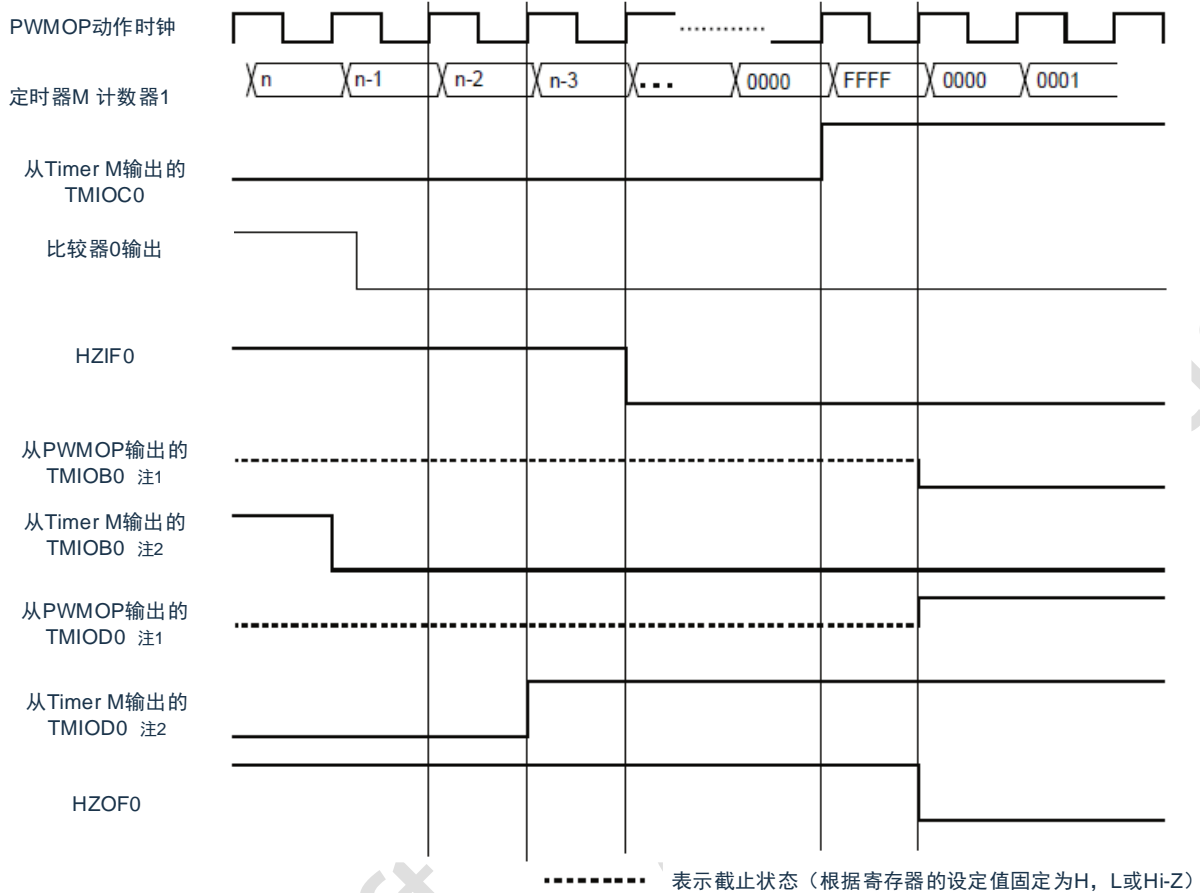
图10-81 硬件解除强制截止的例子(以TMIOB0, TMIOD0为例)



- 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。

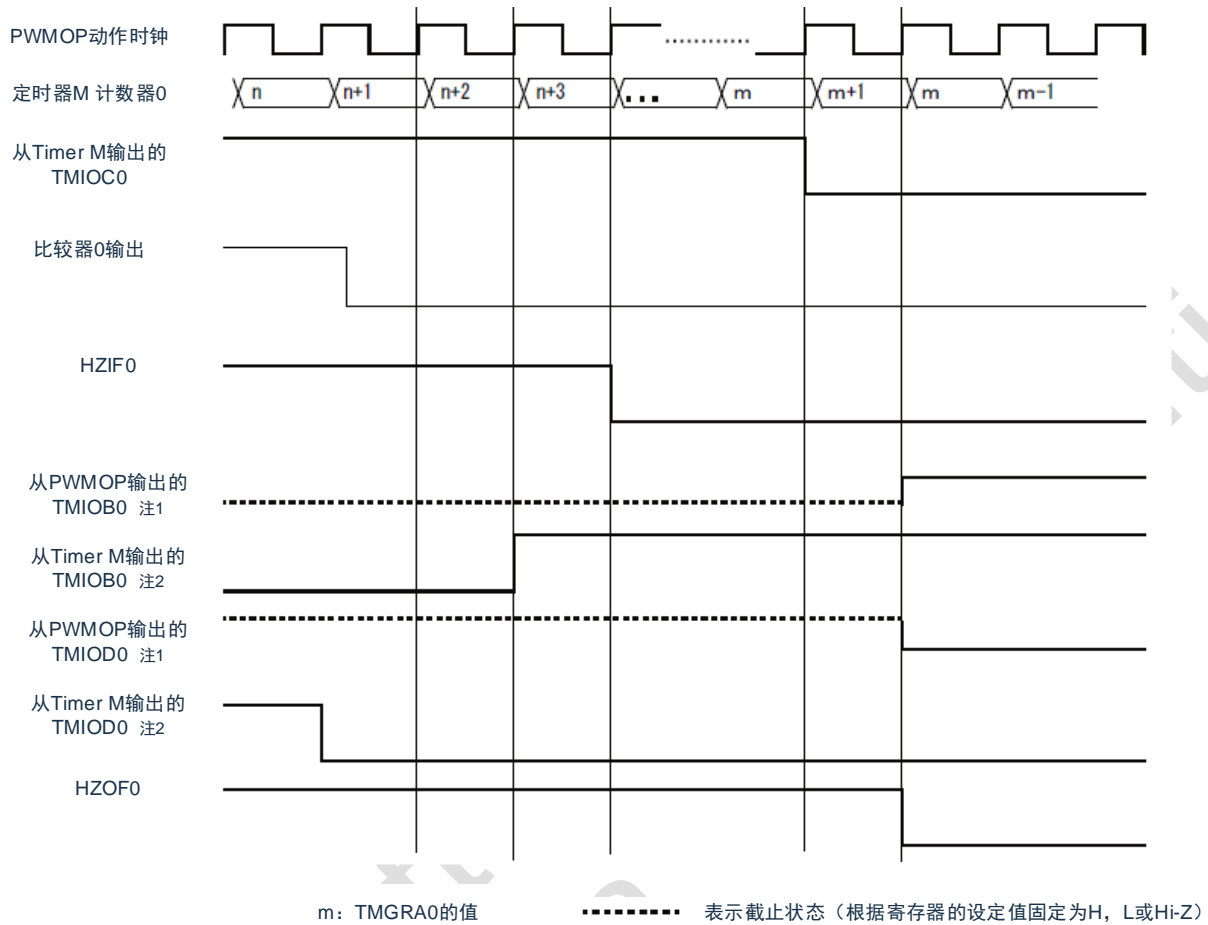
图10-82 强制截止解除的详细时序图(定时器M的计数源= $F_{CLK}$ , 递减计数)



注1: 从PWMOP输出的TMIOD0(\*=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TMIOD0(\*=B~D)表示从定时器M输出到PWMOP的输入信号状态。

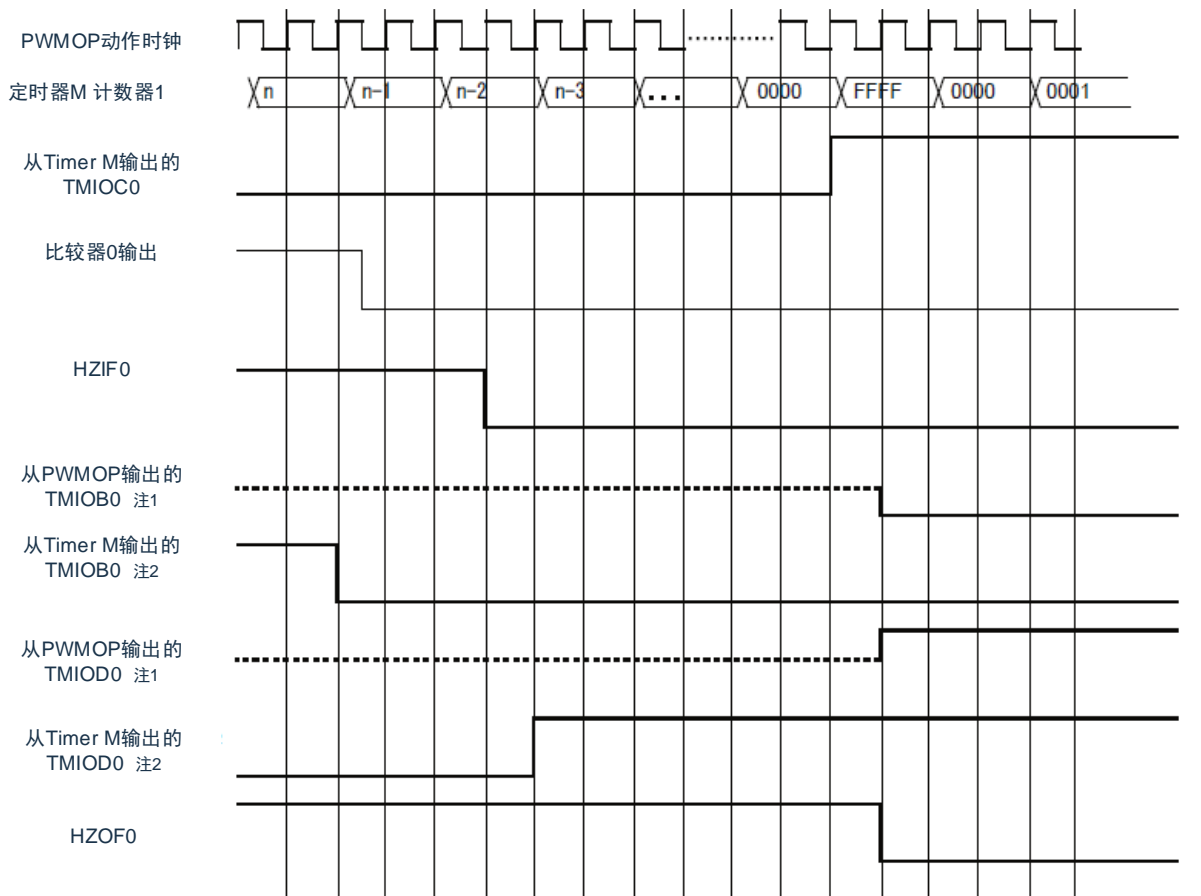
图10-83 强制截止解除的详细时序图(定时器M的计数源= $F_{CLK}$ ，计数器=TMGRA0)



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10-84 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)

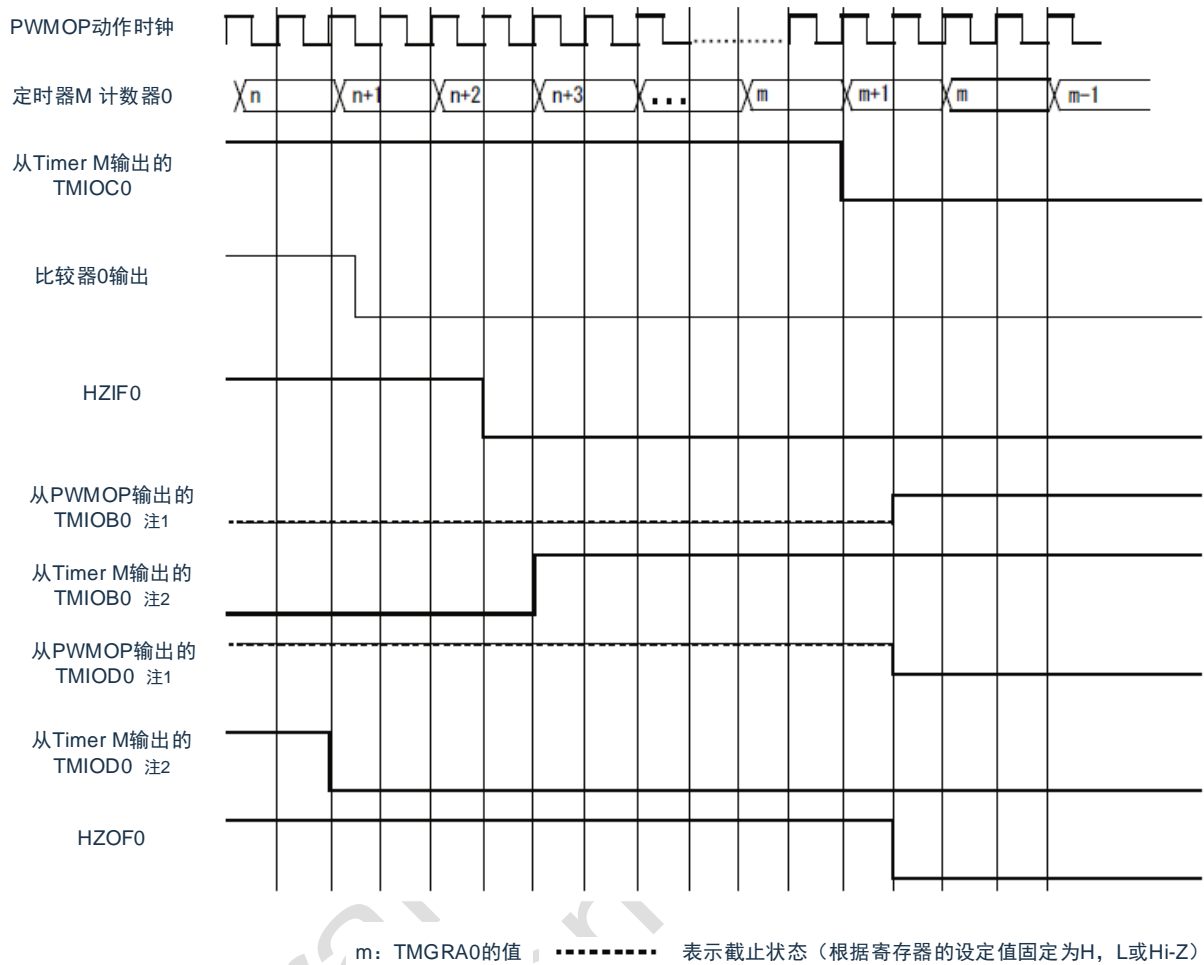


----- 表示截止状态 (根据寄存器的设定值固定为H, L或Hi-Z)

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10-85 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。

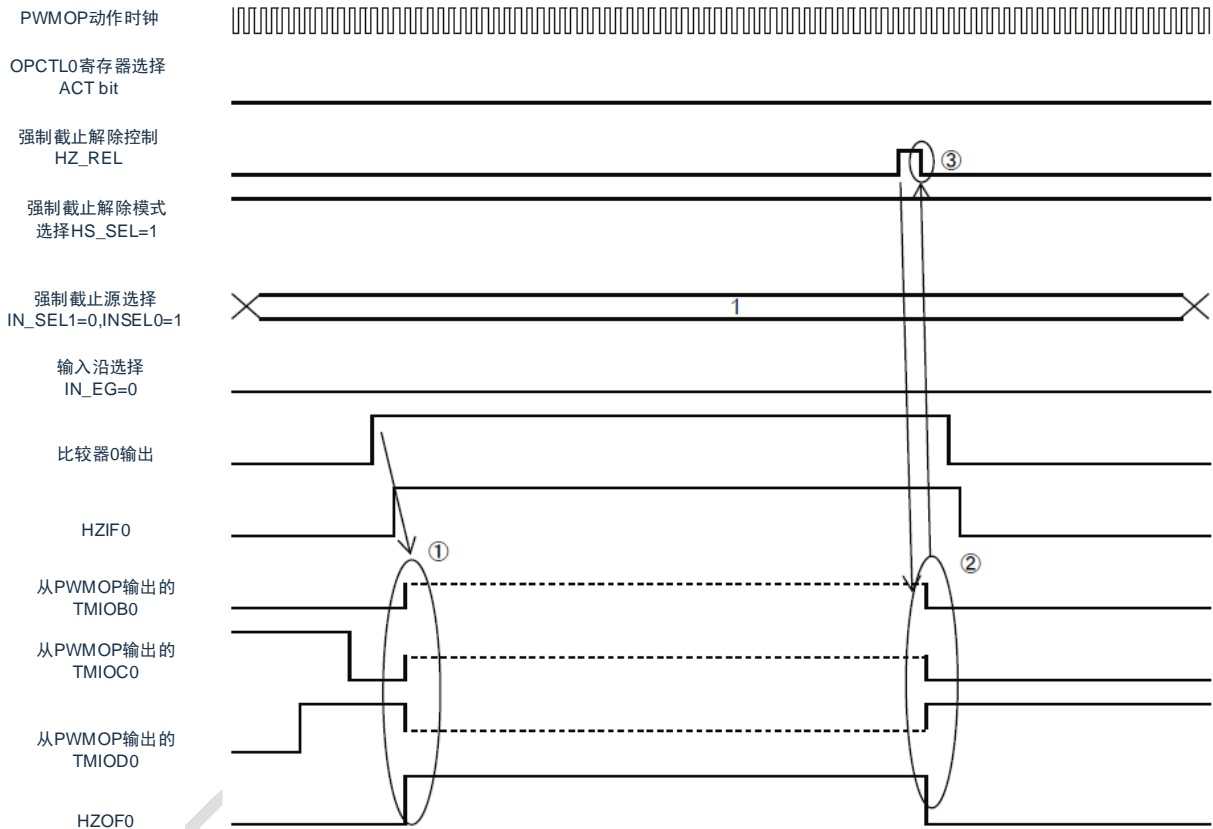
### 10.8.3.3 软件解除(HS\_SEL=1)

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。

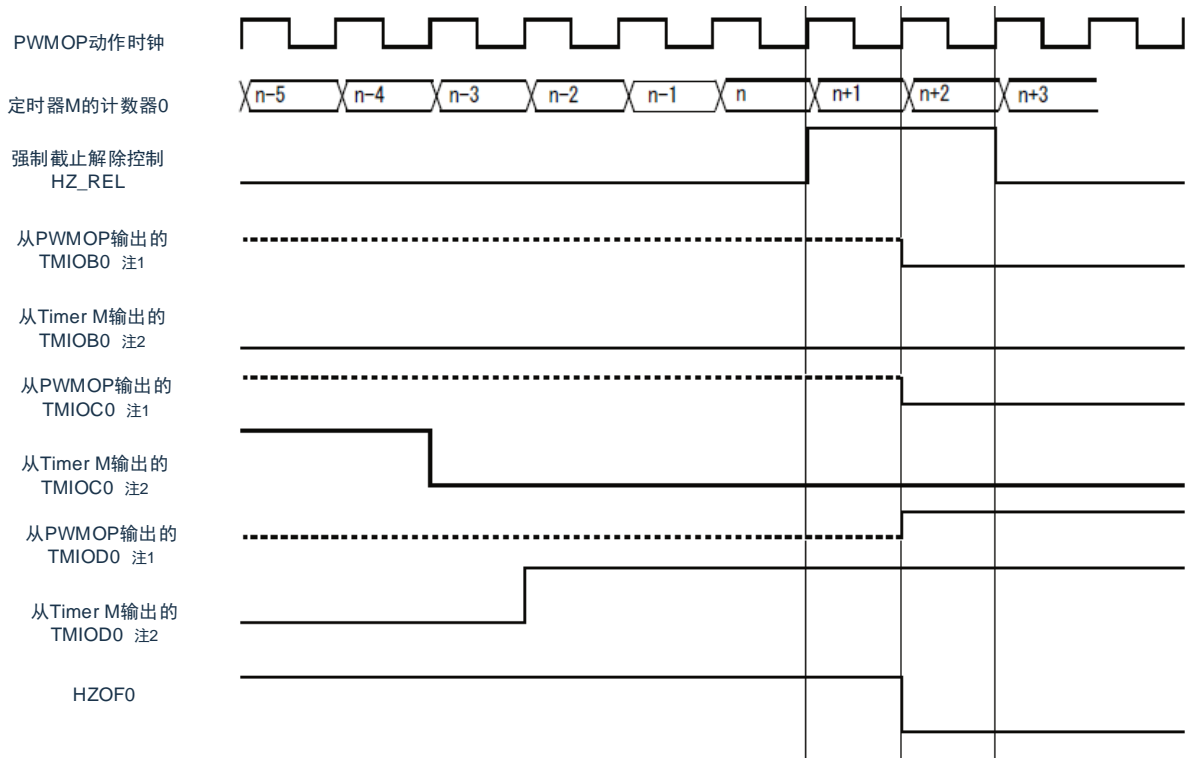
图10-86 软件解除强制截止的例子(以TMIOB0, TMIOC0, TMIOD0为例)



- 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10-78。

图10-87 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。

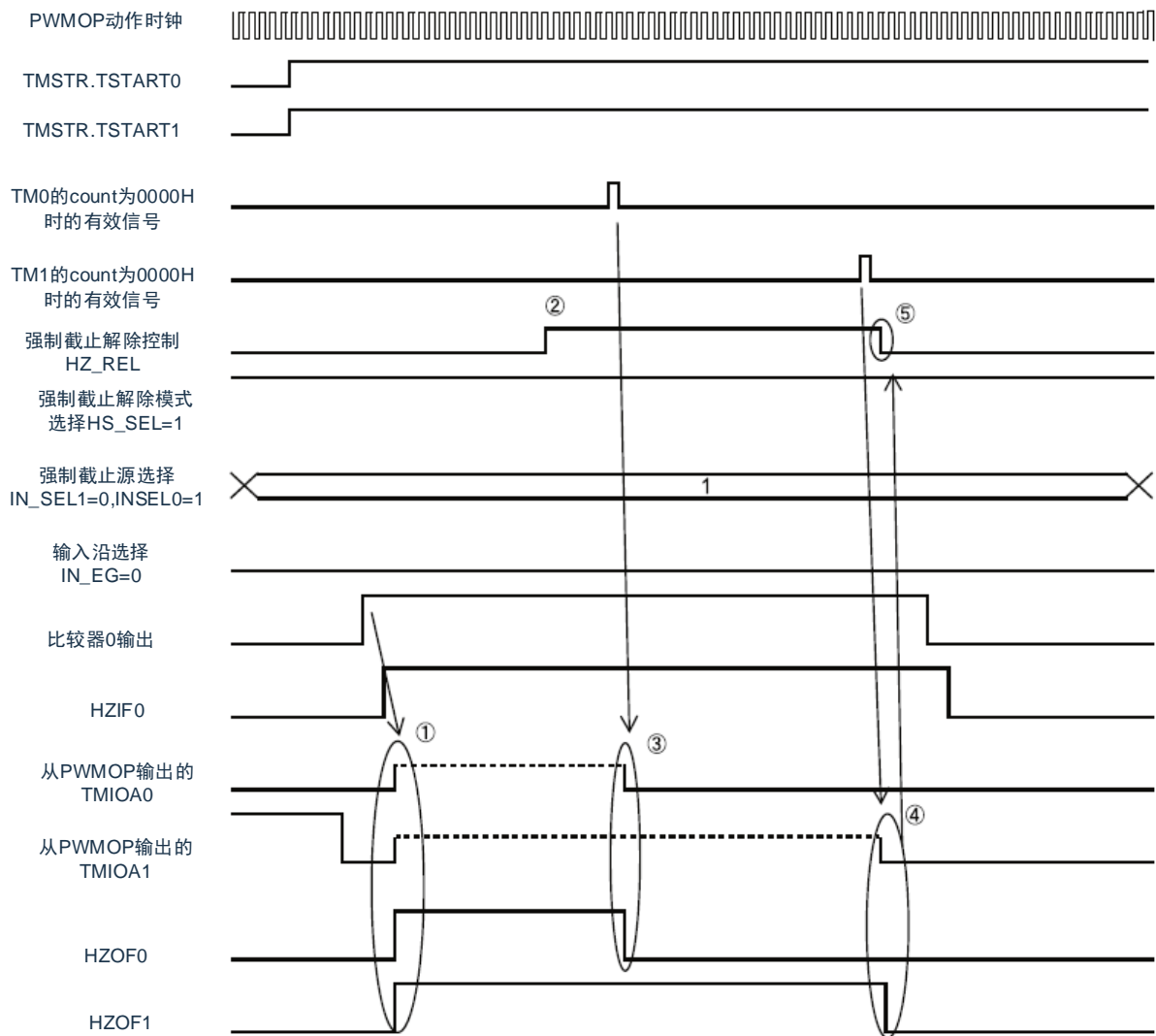
硬件解除截止时，在解除事件被检测到后，通过定时器M的信号来触发解除并且恢复输出。软件解除时，设置HZ\_REL为1后通过定时器M的信号来触发解除并且恢复输出。解除时序是相同的。

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。

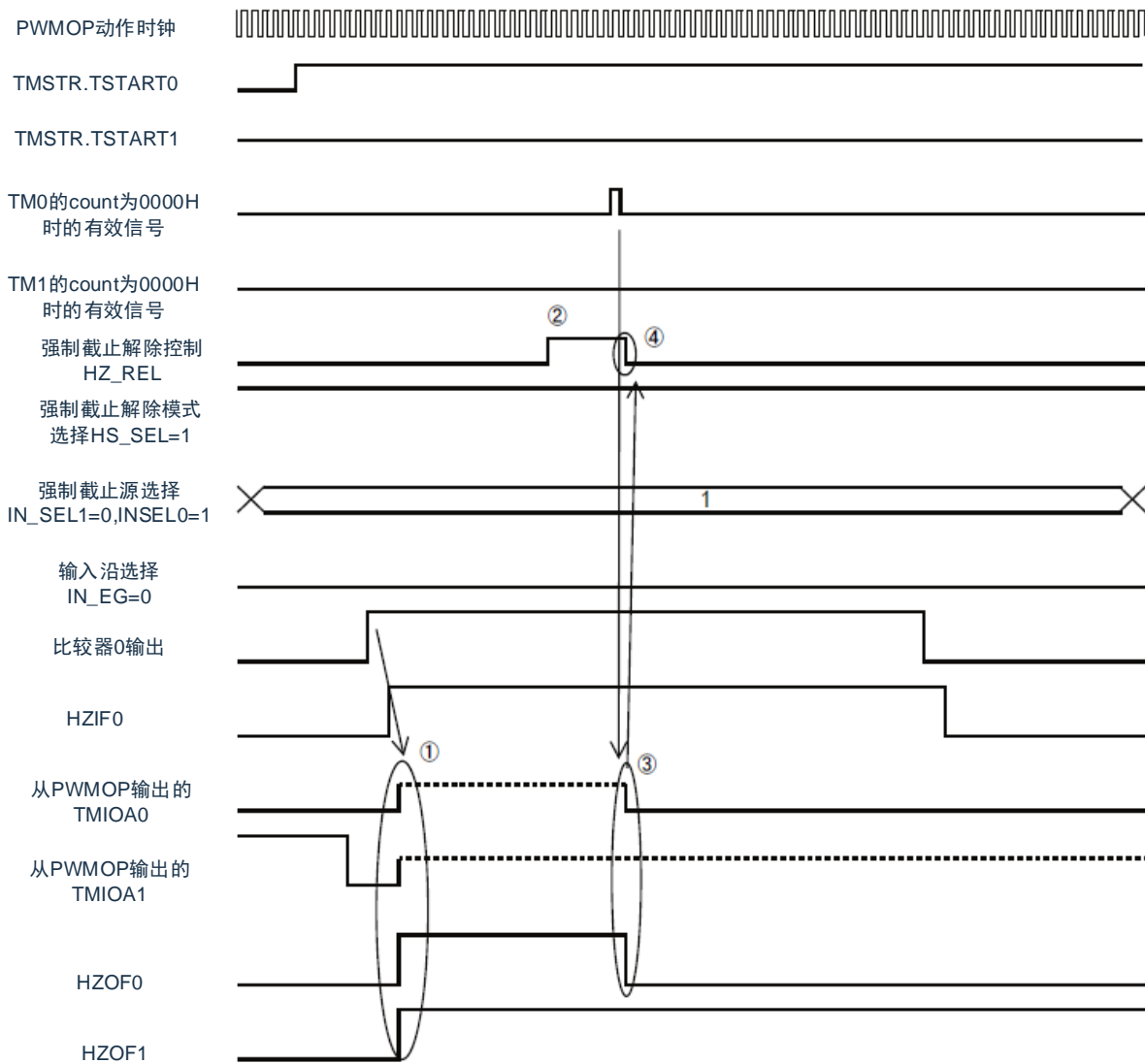


图10-88 软件解除强制截止的例子(定时器M, 2通道计数)



- 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10-89 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) TM0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10-78。

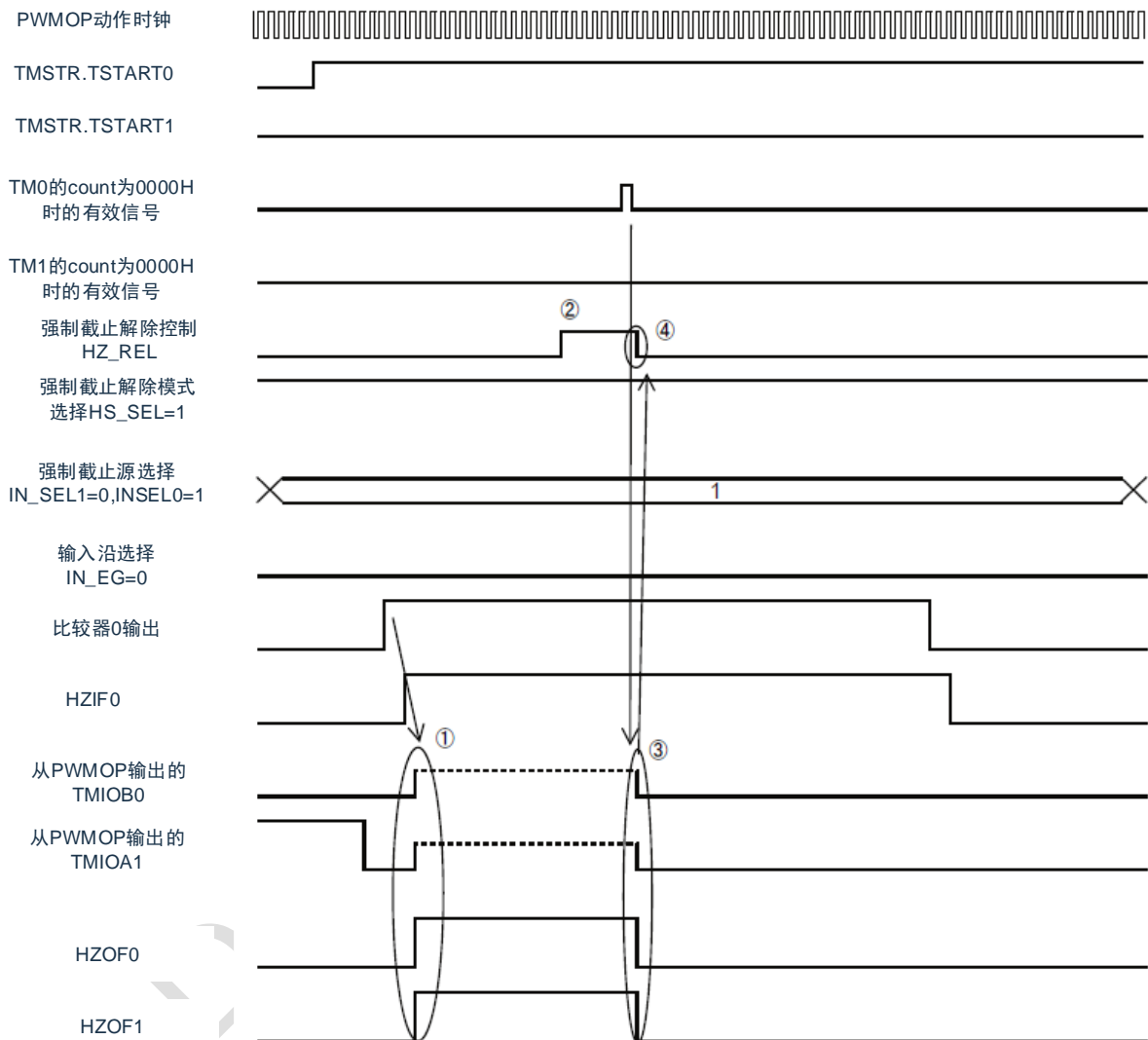
强制截止解除的详细时序请参考图10-79, 图10-80。

HZ\_REL自动清零的时序请参考图10-87。

(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10-90 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10-78。

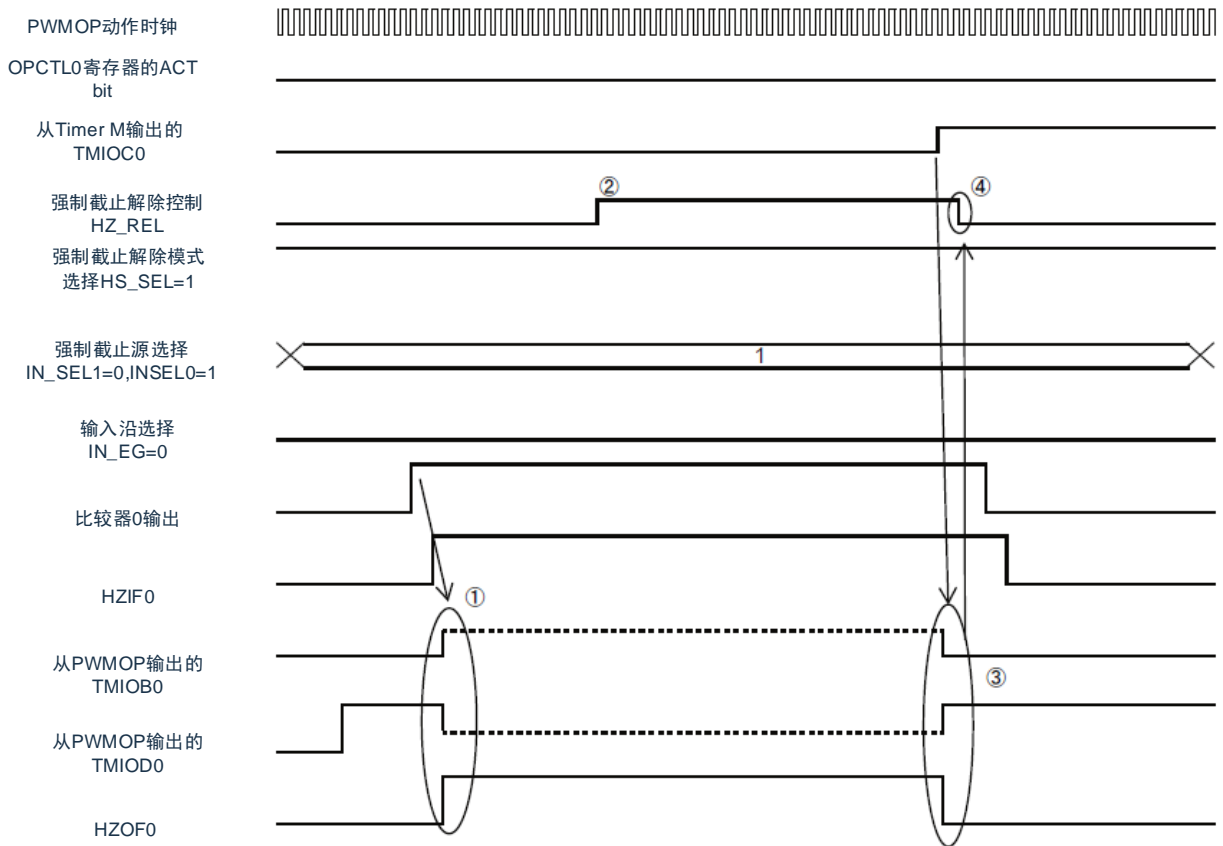
强制截止解除的详细时序请参考图10-79，图10-80。

HZ\_REL自动清零的时序请参考图10-87。

(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10-91 软件解除强制截止的例子(以TMIOB0, TMIOD0为例)



- 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。
- 设置HZ\_REL为1，等待TMIOC0的上升沿。
- TMIOC0的上升沿检出后，强制截止状态解除。
- 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10-78。

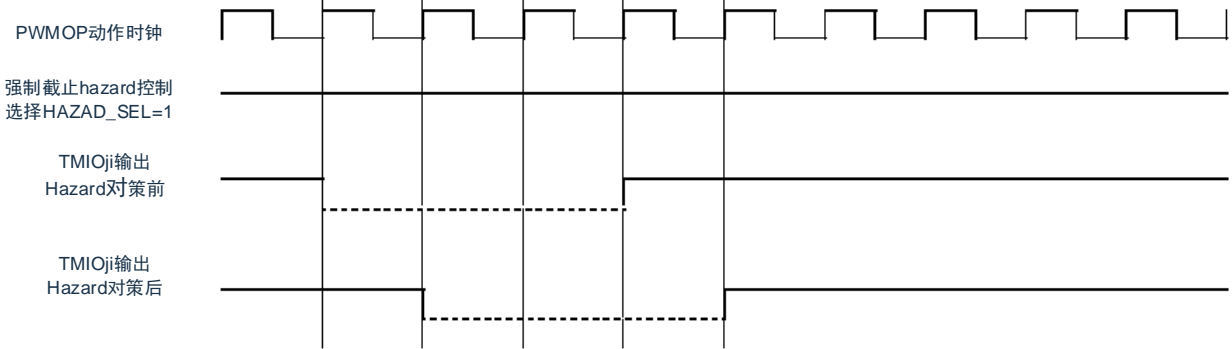
强制截止解除的详细时序请参考图10-82，图10-83，图10-84，图10-85。

HZ\_REL自动清零的时序请参考图10-87。

### 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10-92 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10-93 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)

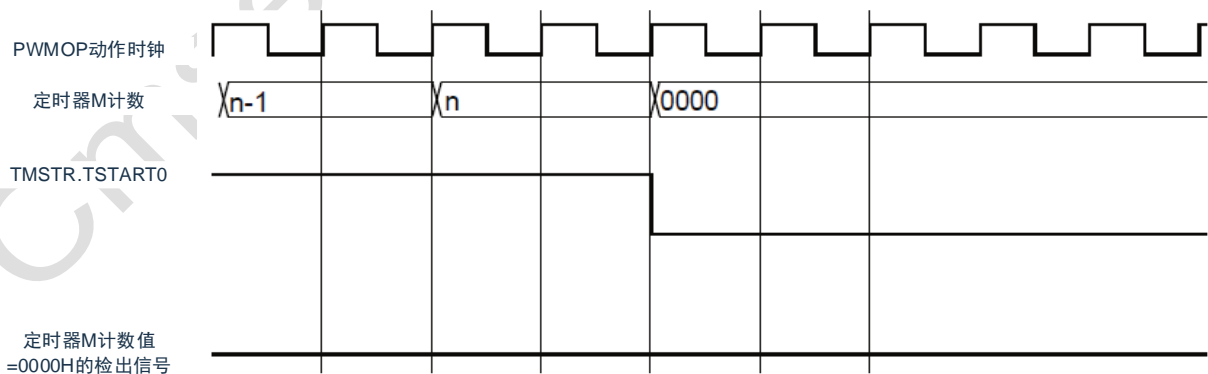
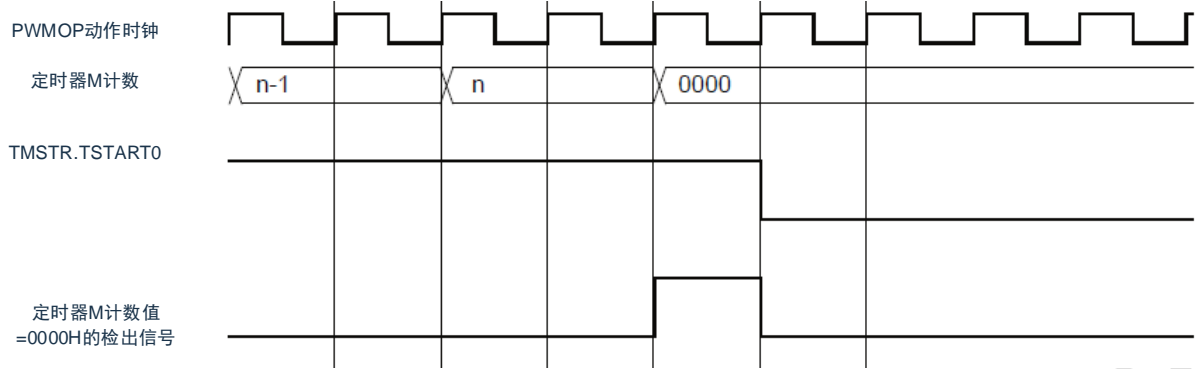


图10-94 计数值=0000H时的判定时序

(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)



Draft Version  
Cmsemicon Confidential



### 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的时钟和模式设置之后

- (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的状态，可能会对PWMOP产生解除截止的沿。
- (3) 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)
- (4) 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。
- (5) 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复用功能和PORT功能。
- (6) 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 时钟输出/蜂鸣器输出控制电路

## 11.1 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。

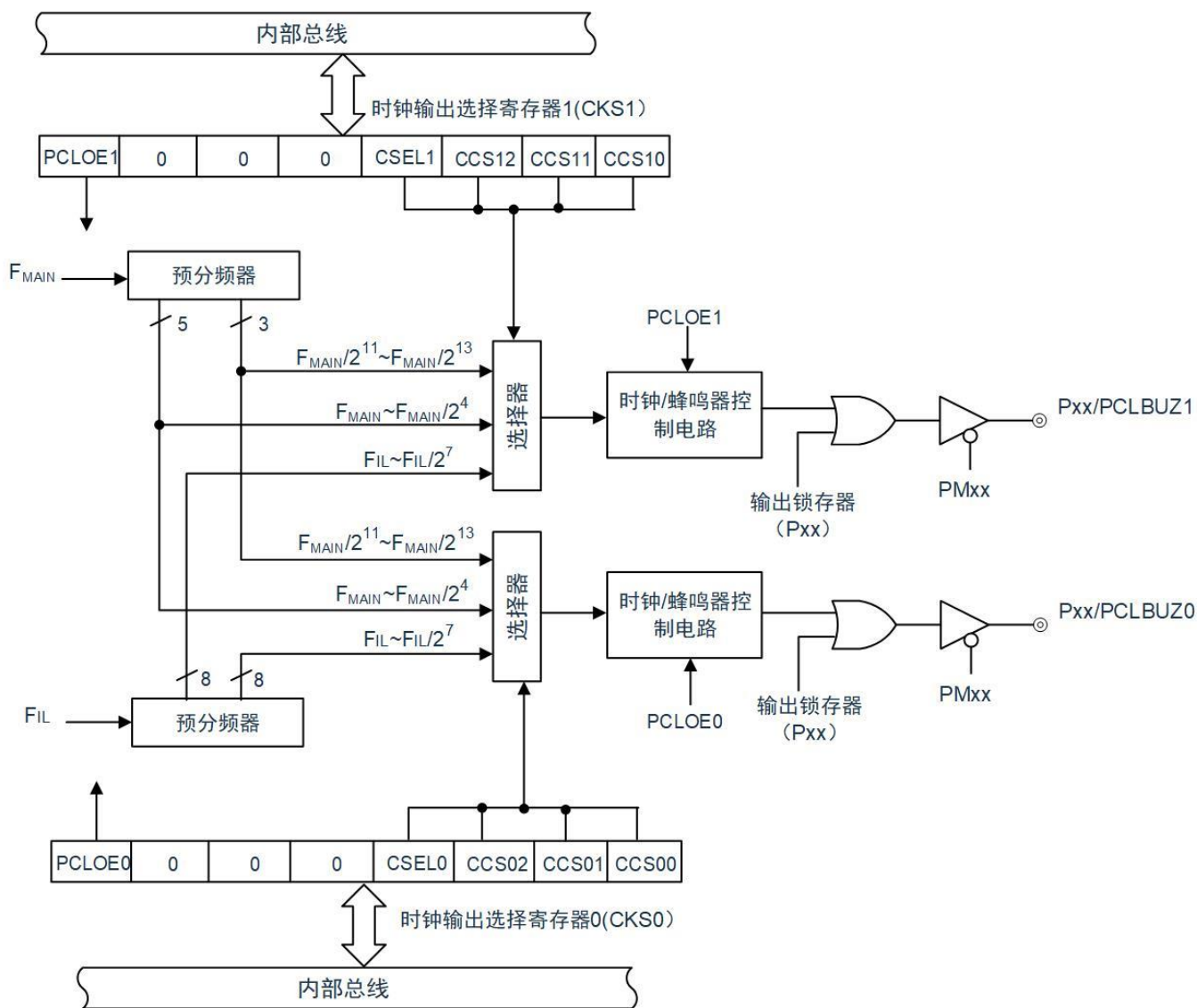
能用1个引脚选择用作时钟输出或者蜂鸣器输出。

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11-1所示。

备注：n=0、1。

图11-1 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”。

## 11.2 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11-1 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)

## 11.3 控制时钟输出/蜂鸣器输出控制电路的寄存器

### 11.3.1 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(CLKBUN)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11-2 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	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 <sub>MAIN</sub> =10MHz	F <sub>MAIN</sub> =20MHz	F <sub>MAIN</sub> =32MHz	F <sub>MAIN</sub> =48MHz
0	0	0	0	F <sub>MAIN</sub>	10MHz <sup>注</sup>	禁止设置 <sup>注</sup>	禁止设置 <sup>注</sup>	禁止设置 <sup>注</sup>
0	0	0	1	F <sub>MAIN</sub> /2	5MHz	10MHz <sup>注</sup>	16MHz <sup>注</sup>	禁止设置 <sup>注</sup>
0	0	1	0	F <sub>MAIN</sub> /2 <sup>2</sup>	2.5MHz	5MHz	8MHz	12MHz
0	0	1	1	F <sub>MAIN</sub> /2 <sup>3</sup>	1.25MHz	2.5MHz	4MHz	6MHz
0	1	0	0	F <sub>MAIN</sub> /2 <sup>4</sup>	625KHz	1.25MHz	2MHz	3MHz
0	1	0	1	F <sub>MAIN</sub> /2 <sup>11</sup>	4.88KHz	9.77KHz	15.63KHz	23.44KHz
0	1	1	0	F <sub>MAIN</sub> /2 <sup>12</sup>	2.44KHz	4.88KHz	7.81KHz	11.72KHz
0	1	1	1	F <sub>MAIN</sub> /2 <sup>13</sup>	1.22KHz	2.44KHz	3.91KHz	5.86KHz
1	0	0	0	F <sub>IL</sub>	15KHz			
1	0	0	1	F <sub>IL</sub> /2	7.5KHz			
1	0	1	0	F <sub>IL</sub> /2 <sup>2</sup>	3.75KHz			
1	0	1	1	F <sub>IL</sub> /2 <sup>3</sup>	1.875KHz			
1	1	0	0	F <sub>IL</sub> /2 <sup>4</sup>	937.5Hz			
1	1	0	1	F <sub>IL</sub> 2 <sup>5</sup>	468.75Hz			
1	1	1	0	F <sub>IL</sub> /2 <sup>6</sup>	234.375Hz			
1	1	1	1	F <sub>IL</sub> /2 <sup>7</sup>	117.1875Hz			

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：输出时钟的切换必须在设置为禁止输出(PCLOEn=0)后进行。

备注：n=0、1

F<sub>MAIN</sub>：主系统时钟频率

F<sub>IL</sub>：低速内部振荡器时钟频率

Draft Version  
Cmsemicon Confidential

## 11.3.2 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P31/CLKBUZ0用作时钟输出/蜂鸣器输出的情况

将端口模式寄存器3的PM31位置“0”。

将端口寄存器3的P31位置“0”。

## 11.4 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。

### 11.4.1 输出引脚的运行

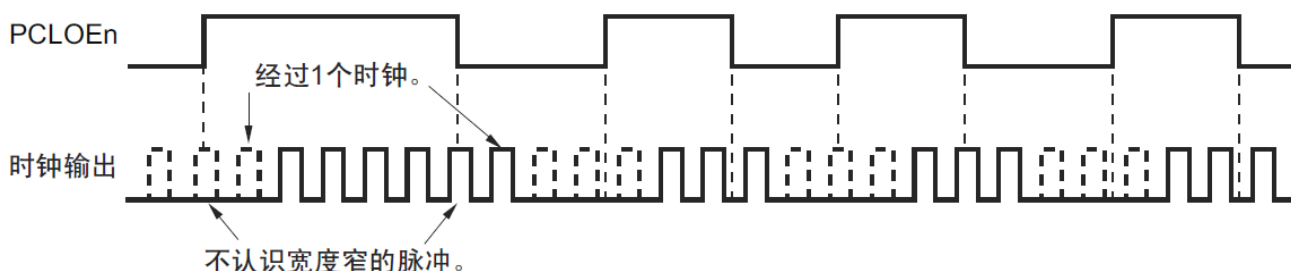
CLKBUZn引脚按照以下步骤进行输出：

- 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11-3所示。
- ② n=0、1

图11-3 CLKBUZn引脚的时钟输出时序



## 11.5 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12章 看门狗定时器

### 12.1 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 $75\%+1/2F_{IL}$ 时，能产生间隔中断。

### 12.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表12-1 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。

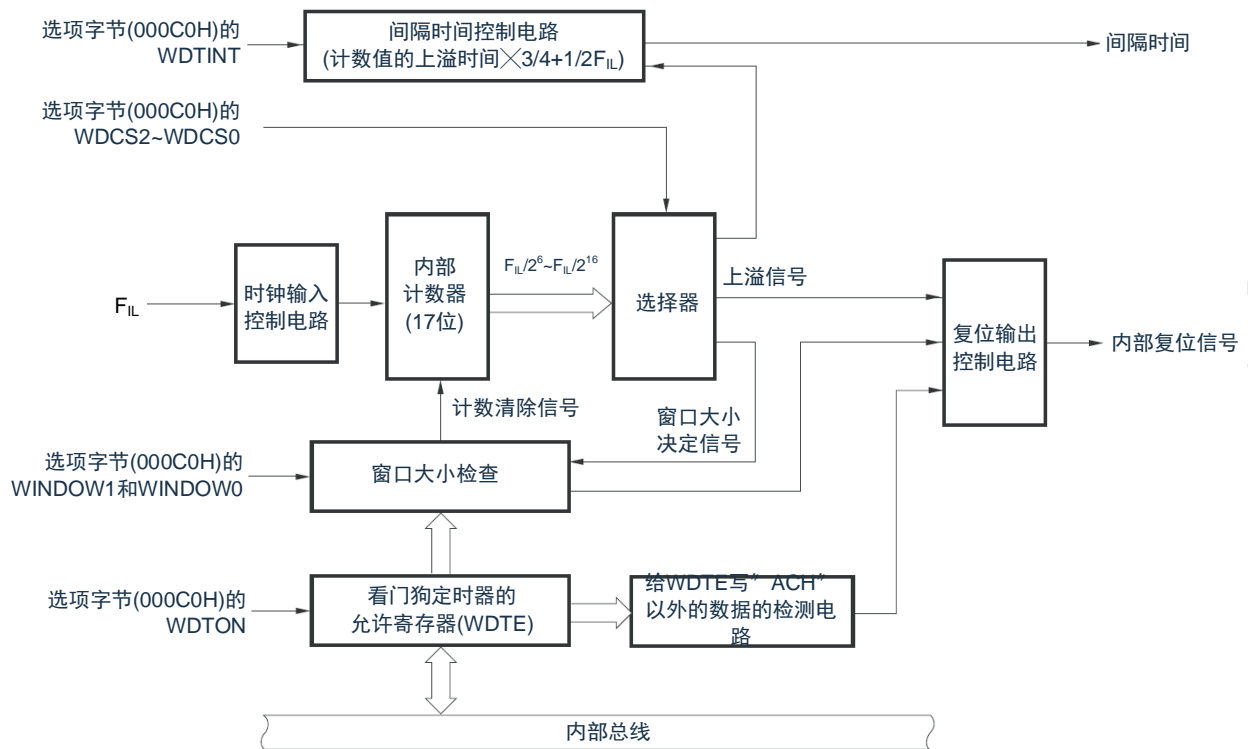
表12-2 选项字节和看门狗定时器的设置内容

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31章 选项字节”。



图12-1 看门狗定时器的框图



备注: F<sub>IL</sub>: 低速内部振荡器的时钟频率

## 12.3 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。

### 12.3.1 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”<sup>注</sup>。

图12-2 看门狗定时器的允许寄存器(WDTE)的格式

地址：0x40021001      复位后：9AH/1AH<sup>注</sup>      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)。

## 12.3.2 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12-3 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12-3 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

### 12.3.3 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12-4 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 <sup>注1注2</sup>
其他				•复位后强制运行看门狗定时器

注意：详细配置参考用户选项字节章节。

## 12.4 看门狗定时器的运行

### 12.4.1 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31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12.4.2和第31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12.4.2和第31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。

## 12.4.2 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12-3 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。

## 12.4.3 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12-4: 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:

$$2^9/F_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/F_{IL}(\text{MAX.})=12.8\sim 25.6\text{ms}$$

## 12.4.4 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12-5 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。

## 12.4.5 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13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	24引脚
模拟输入通道	10ch	8ch
	(ANI0~ANI3、ANI8~ANI12、ANI14)	(ANI0~ANI2、ANI8~ANI12)

### 13.1 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13个模拟通道的A/D转换(10个引脚输入通道和3个内部通道(PGA0、温度传感器和内部基准电压(1.45V))。

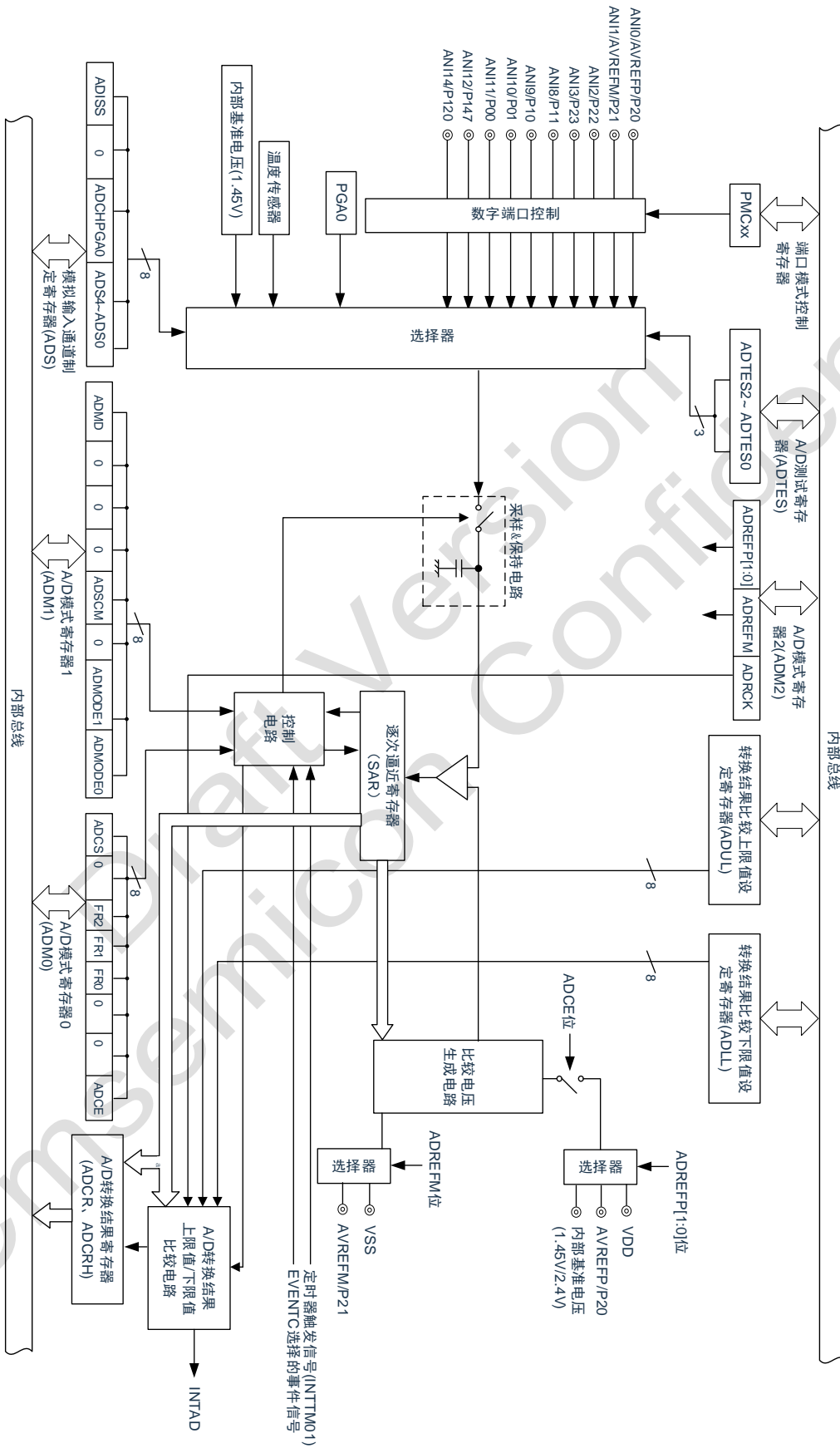
A/D转换器具有以下功能：

- 12位分辨率的A/D转换

从ANI0~ANI3, ANI8~ANI12、ANI14, PGA0, 温度传感器或内部基准电压(1.45V)中选择1个模拟输入, 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~ANI3, ANI8~ANI12、ANI14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 <sub>AD</sub> )。

图13-1 A/D转换器的框图



## 13.2 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
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:只读, W:只写, R/W:读和写

## 13.2.1 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13-2 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	0	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	0	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)

## 13.2.2 内部基准电压选择 (VREFCTL)

当A/D转换器的正(+)基准电压源选择使用内部基准电压时，通过VREFCTL寄存器可以选择内部基准电压是1.45V或2.4V。

地址：0x4004047A

复位值：00H      R/W

	7	6	5	4	3	2	1	0
VREFCTL	0	0	0	0	0	0	VREFSEL	0

VREFSEL	内部基准电压选择
0	内部基准电压选择为 1.45V
1	内部基准电压选择为 2.4V

注：A/D 转换内部基准电压不受此寄存器的影响，转换的内部基准电压固定为 1.45V。

### 13.2.3 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13-3 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 <sup>注1</sup>	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 <sup>注2</sup>
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位和A/D转换的详细内容，请参照“表13-3：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“13.5 A/D转换器的设置流程图”的步骤进行设置。

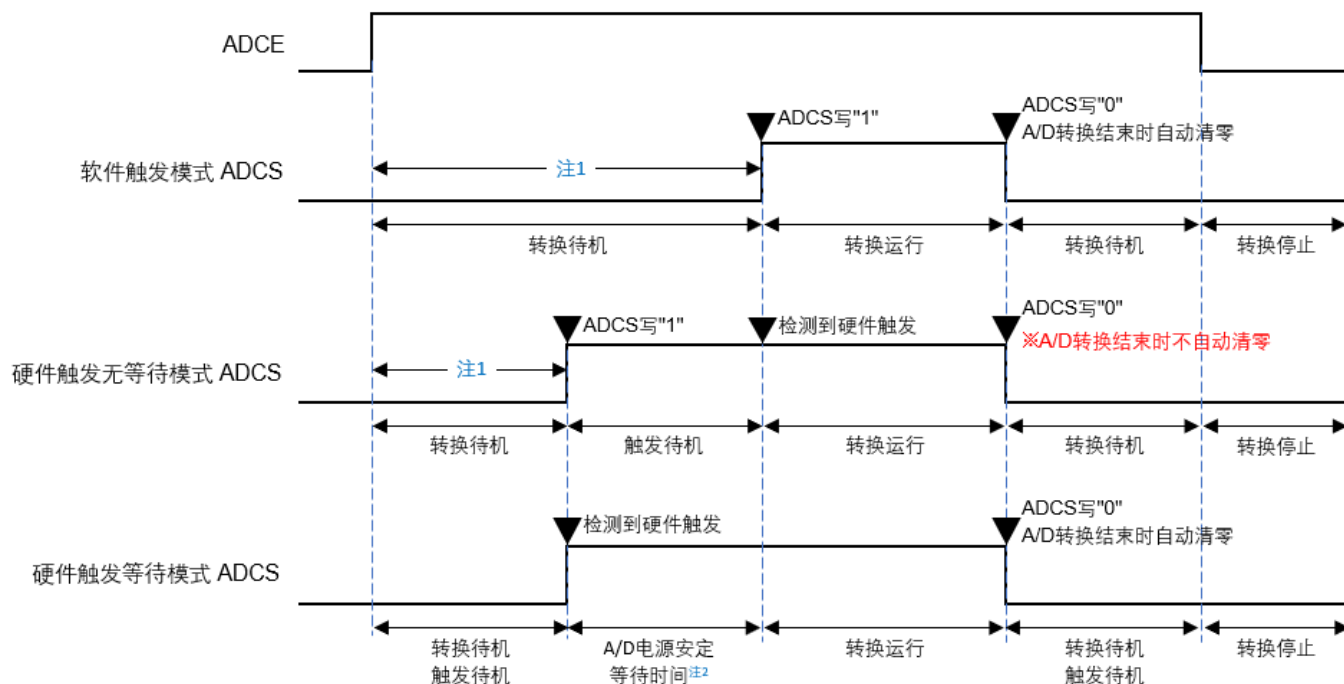
表13-1 ADCS位和ADCE位的设置

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13-2 ADCS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		•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		•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		•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		•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13-4 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1us。

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. 在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<sub>CLK</sub>时钟+A/D转换时间

硬件触发等待模式时：2个F<sub>CLK</sub>时钟+A/D电源稳定等待时间+A/D转换时间

备注：F<sub>CLK</sub>：CPU/外围硬件的时钟频率

表13-3 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 <sub>AD</sub> )	12位分辨率的转换时间	
FR2	FR1	FR0	ADM0[1]	ADM0[0]			转换时钟数	转换时间
0	0	0	0	0	高速变 换模式	F <sub>CLK</sub> /32	45个ADCLK (采样时钟数：13.5个 ADCLK)	1440/F <sub>CLK</sub>
0	0	1				F <sub>CLK</sub> /16		720/F <sub>CLK</sub>
0	1	0				F <sub>CLK</sub> /8		360/F <sub>CLK</sub>
0	1	1				F <sub>CLK</sub> /4		180/F <sub>CLK</sub>
1	0	0				F <sub>CLK</sub> /2		90/F <sub>CLK</sub>
1	0	1				F <sub>CLK</sub> /1		45/F <sub>CLK</sub>
0	0	0	1	1	低电流 模式	F <sub>CLK</sub> /32	54个ADCLK (采样时钟数：13.5个 ADCLK)	1728/F <sub>CLK</sub>
0	0	1				F <sub>CLK</sub> /16		864/F <sub>CLK</sub>
0	1	0				F <sub>CLK</sub> /8		432/F <sub>CLK</sub>
0	1	1				F <sub>CLK</sub> /4		216/F <sub>CLK</sub>
1	0	0				F <sub>CLK</sub> /2		108/F <sub>CLK</sub>
1	0	1				F <sub>CLK</sub> /1		54/F <sub>CLK</sub>

注意：要将FR2~FR0位、ADM0[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<sub>CLK</sub>：CPU/外围硬件的时钟频率

表13-3 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<sup>注1</sup>)

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 <sub>AD</sub> )	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	ADM0[1]	ADM0[0]					
0	0	0	0	0	高速变 换模式	F <sub>CLK</sub> /32	1us	45个ADCLK (采样时钟数：13.5 个ADCLK)	1us +1440/F <sub>CLK</sub>
0	0	1				F <sub>CLK</sub> /16			1us +720/F <sub>CLK</sub>
0	1	0				F <sub>CLK</sub> /8			1us +360/F <sub>CLK</sub>
0	1	1				F <sub>CLK</sub> /4			1us +180/F <sub>CLK</sub>
1	0	0				F <sub>CLK</sub> /2			1us +90/F <sub>CLK</sub>
1	0	1				F <sub>CLK</sub> /1			1us +45/F <sub>CLK</sub>
0	0	0	1	1	低电流 模式	F <sub>CLK</sub> /32	1us	54个ADCLK (采样时钟数：13.5 个ADCLK)	1us +1728/F <sub>CLK</sub>
0	0	1				F <sub>CLK</sub> /16			1us +864/F <sub>CLK</sub>
0	1	0				F <sub>CLK</sub> /8			1us +432/F <sub>CLK</sub>
0	1	1				F <sub>CLK</sub> /4			1us +216/F <sub>CLK</sub>
1	0	0				F <sub>CLK</sub> /2			1us +108/F <sub>CLK</sub>
1	0	1				F <sub>CLK</sub> /1			1us +54/F <sub>CLK</sub>

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13-3)。

注意：

1. 要将FR2~FR0位、ADM0[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<sub>CLK</sub>：CPU/外围硬件的时钟频率。



## 13.2.4 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13-5 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<sub>CLK</sub>时钟+A/D转换时间  
 硬件触发等待模式时：2个F<sub>CLK</sub>时钟+A/D电源稳定等待时间+A/D转换时间

备注：F<sub>CLK</sub>：CPU/外围硬件的时钟频率

### 13.2.5 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13-6 A/D转换器的模式寄存器2(ADM2)的格式

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 <sub>DD</sub> 提供。
0	1	由P20/AV <sub>REFP</sub> /ANI0引脚提供。
1	0	由内部基准电压（1.45V/2.4V）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 <sub>SS</sub> 提供。
1	由P21/AV <sub>REFM</sub> /ANI1引脚提供。

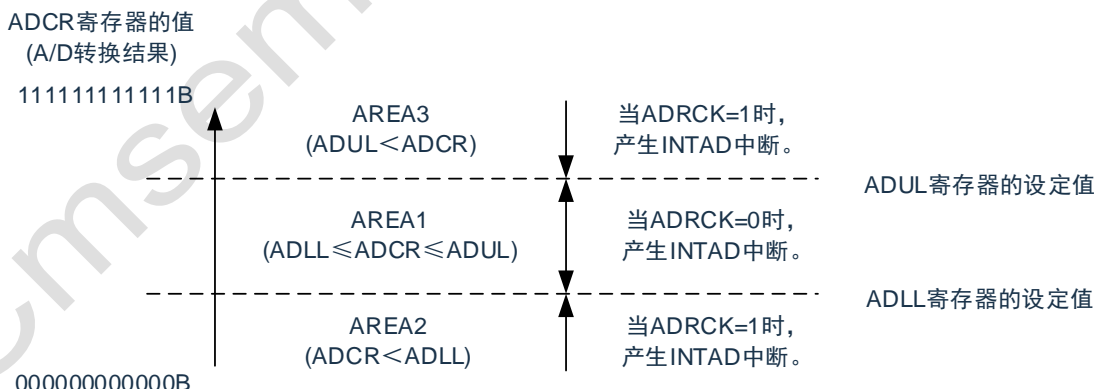
ADRCK	转换结果上限值和下限值的检查
0	当ADLL寄存器 ≤ ADCR寄存器 ≤ ADUL寄存器 (AREA1)时，产生中断信号(INTAD)。
1	当ADCR寄存器 < ADLL寄存器 (AREA2)或者ADUL寄存器 < ADCR寄存器 (AREA3)时，产生中断信号(INTAD)。

AREA1~AREA3的中断信号(INTAD)的产生范围如图13-7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13-7 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。

## 13.2.6 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13-8 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	禁止设定
1	1	禁止设定

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

## 13.2.7 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13-9 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	0	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0

### ◆ 选择模式(ADM1.ADMD=0)

ADS 寄存器设置值							CH 选择
ADISS	ADCHPGA0	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	ANI0(P20)
0	0	0	0	0	0	1	ANI1(P21)
0	0	0	0	0	1	0	ANI2(P22)
0	0	0	0	0	1	1	ANI3(P23)
0	0	0	0	1	0	0	禁止设定
0	0	0	0	1	0	1	禁止设定
0	0	0	0	1	1	0	禁止设定
0	0	0	0	1	1	1	禁止设定
0	0	0	1	0	0	0	ANI8(P11)
0	0	0	1	0	0	1	ANI9(P10)
0	0	0	1	0	1	0	ANI10(P03)
0	0	0	1	0	1	1	ANI11(P02)
0	0	0	1	1	0	0	ANI12(P147)
0	0	0	1	1	0	1	禁止设定
0	0	0	1	1	1	0	ANI14(P120)
0	0	0	1	1	1	1	禁止设定
0	0	1	1	1	1	1	ANI0~ANI3、ANI8~ANI2 和 ANI14 通道全关
1	0	0	0	0	0	0	温度传感器的输出电压
1	0	0	0	0	0	1	内部基准电压(1.45V)
0	1	0	0	0	0	0	PGA0
其他禁止设置							

## ◆ 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	禁止设定			
0	0	0	1	0				
0	0	0	1	1				
0	0	1	0	0				
0	0	1	0	1				
0	0	1	1	0				
0	0	1	1	1				
0	1	0	0	0				
0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	1	0	1	0	禁止设定			
0	1	0	1	1				
0	1	1	0	0				
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，才可以通过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<sub>REFP</sub>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<sub>REFM</sub>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“13.5.5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以低速内部振荡器时钟运行中转移到睡眠模式时，不能将ADISS位置“1”。

## 13.2.8 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<sup>注</sup>。

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13-7))的设置值范围内，就不保存A/D转换结果。

图13-10 12位A/D转换结果寄存器(ADCR)的格式

复位值：0000H R



注意：

1. 如果仅需要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开始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)
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	1	0	ANI14(P120)

### 13.2.9 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位分辨率转换结果的高8位<sup>注</sup>。

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13-7))的设置值范围内，就不保存A/D转换结果。

图13-11 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。

## 13.2.10 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13-7)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13-12 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

## 13.2.11 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13-7)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13-13 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。



## 13.2.12 A/D采样时间控制寄存器(ADNSMP)

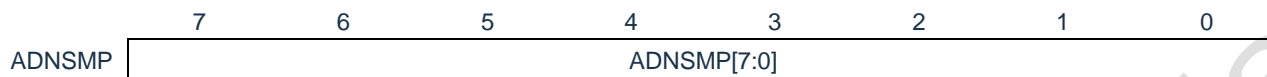
此寄存器可控制A/D采样时间。

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13-14 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 [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

### 13.2.13 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13-15 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”。

### 13.2.14 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13-16 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	0	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。

### 13.2.15 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13-17 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

## 13.2.16 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~ANI3、ANI8~ANI12与ANI14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”。

Draft Version  
Cmsemicon Confidential

### 13.3 输入电压和转换结果

模拟输入引脚(ANI0~ANI3、ANI8~ANI12与ANI14)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

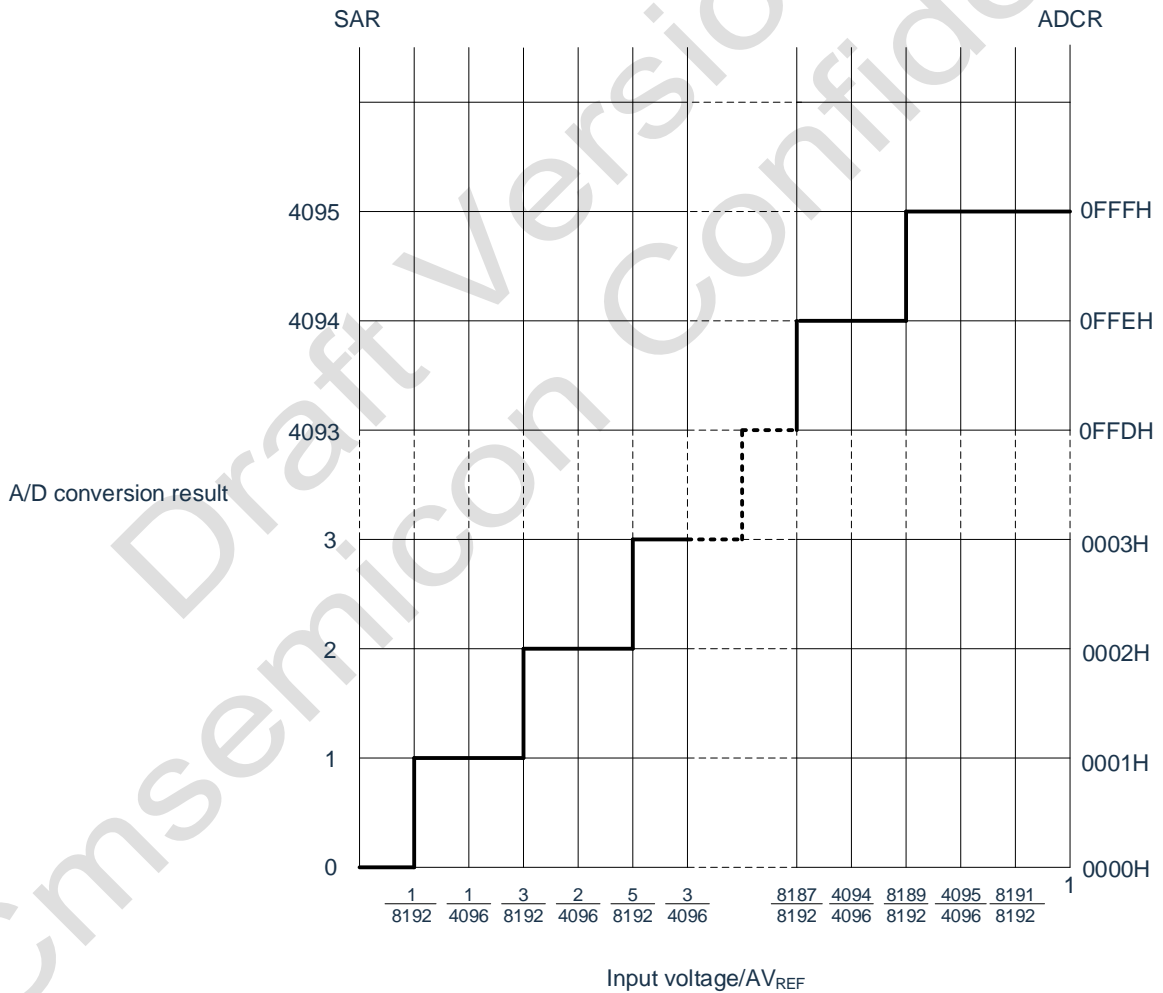
$V_{AIN}$ : 模拟输入电压

$AV_{REF}$ : AVREF引脚电压

ADCR: A/D转换结果寄存器(ADCR)的值

模拟输入电压和A/D转换结果的关系如图13-18所示。

图13-18 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 $AV_{REFP}$ 、 $V_{DD}$ 或者内部基准电压(1.45V/2.4V)。

SAR: 逐次逼近寄存器

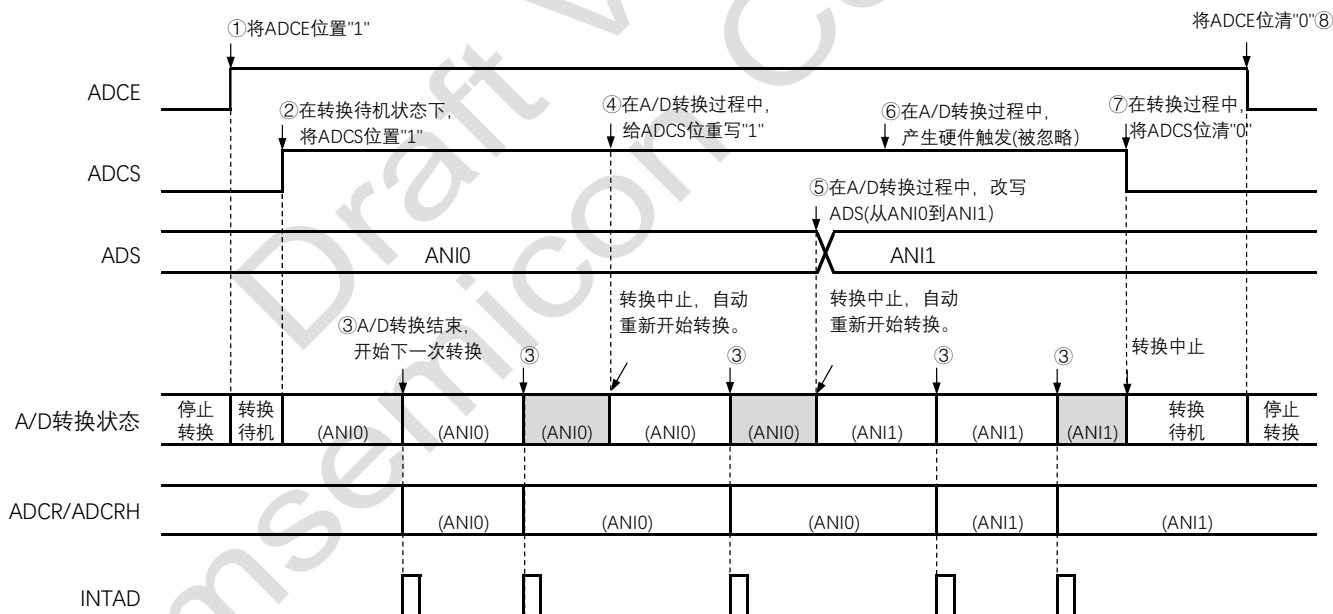
## 13.4 A/D转换器的运行模式

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“13.5A/D转换器的设置流程图”。

### 13.4.1 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。

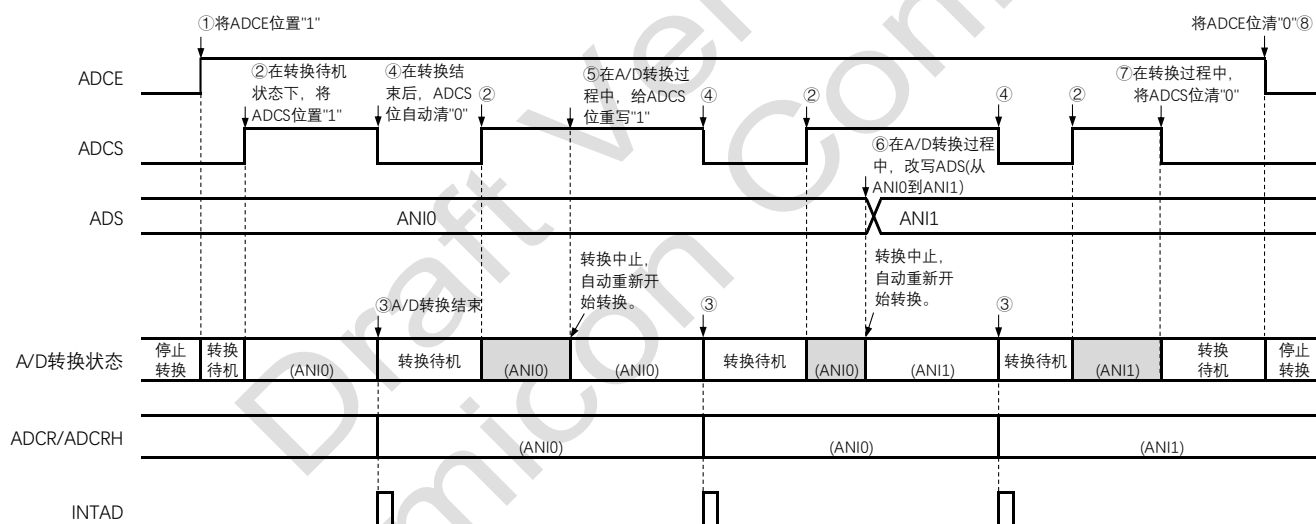
图13-19 软件触发模式(选择模式、连续转换模式)的运行时序例子



### 13.4.2 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。

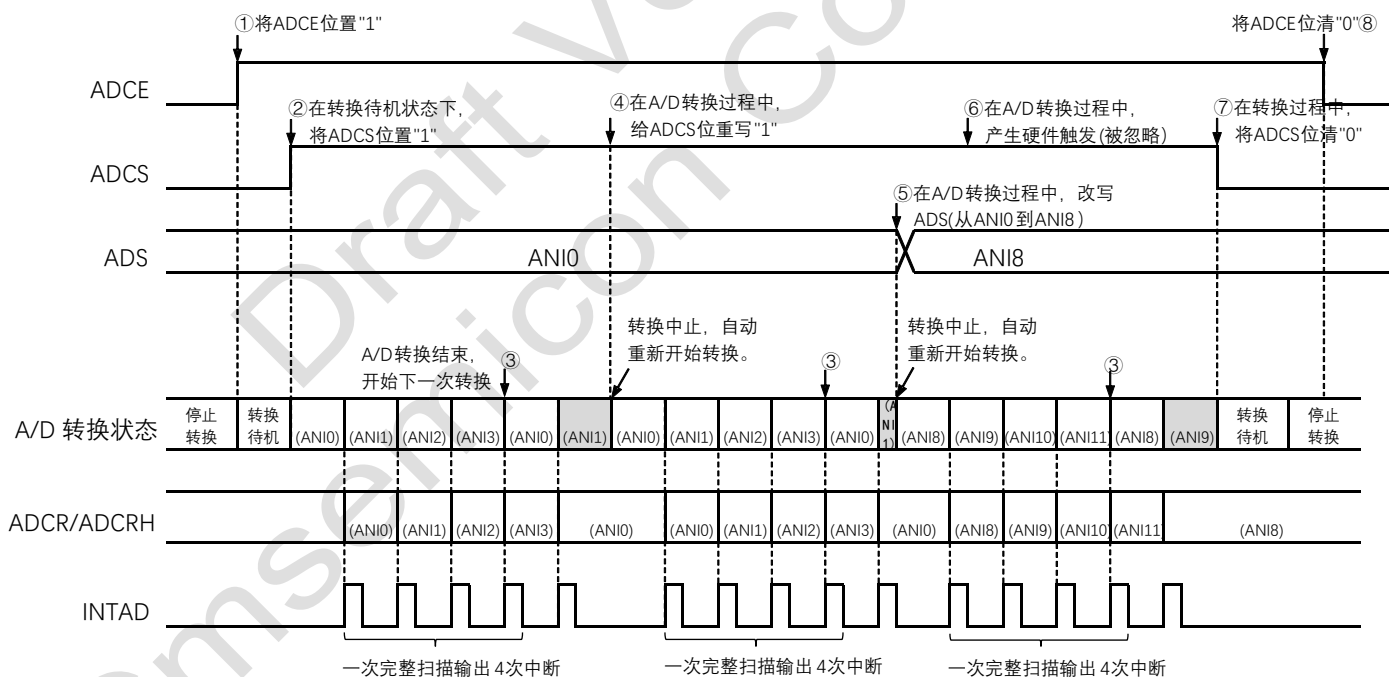
图13-20 软件触发模式(选择模式、单次转换模式)的运行时序例子



### 13.4.3 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13-21 软件触发模式(扫描模式、连续转换模式)的运行时序例子

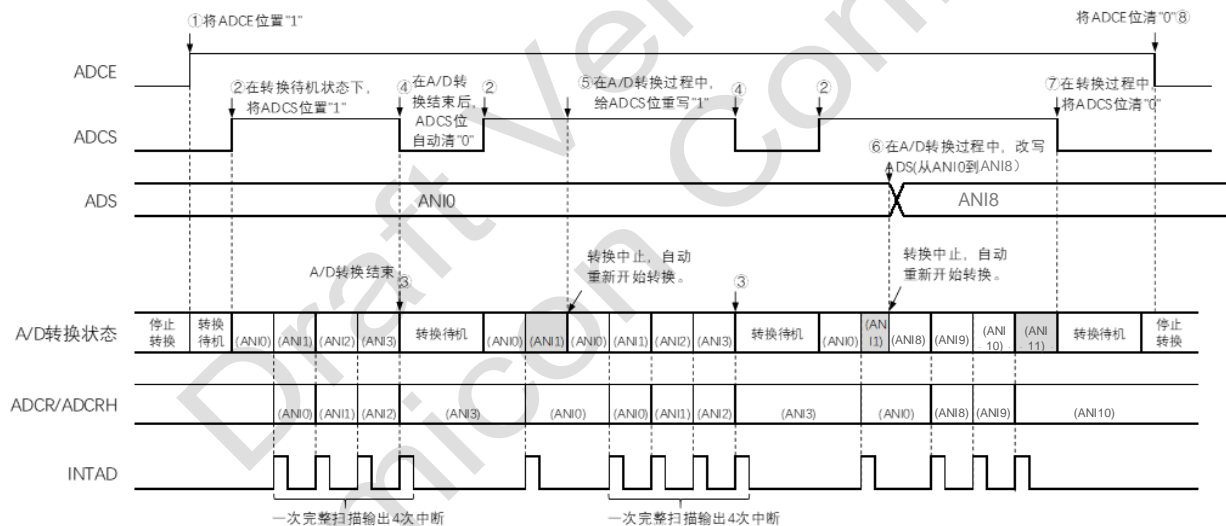




### 13.4.4 软件触发模式(扫描模式、单次转换模式)

- 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。

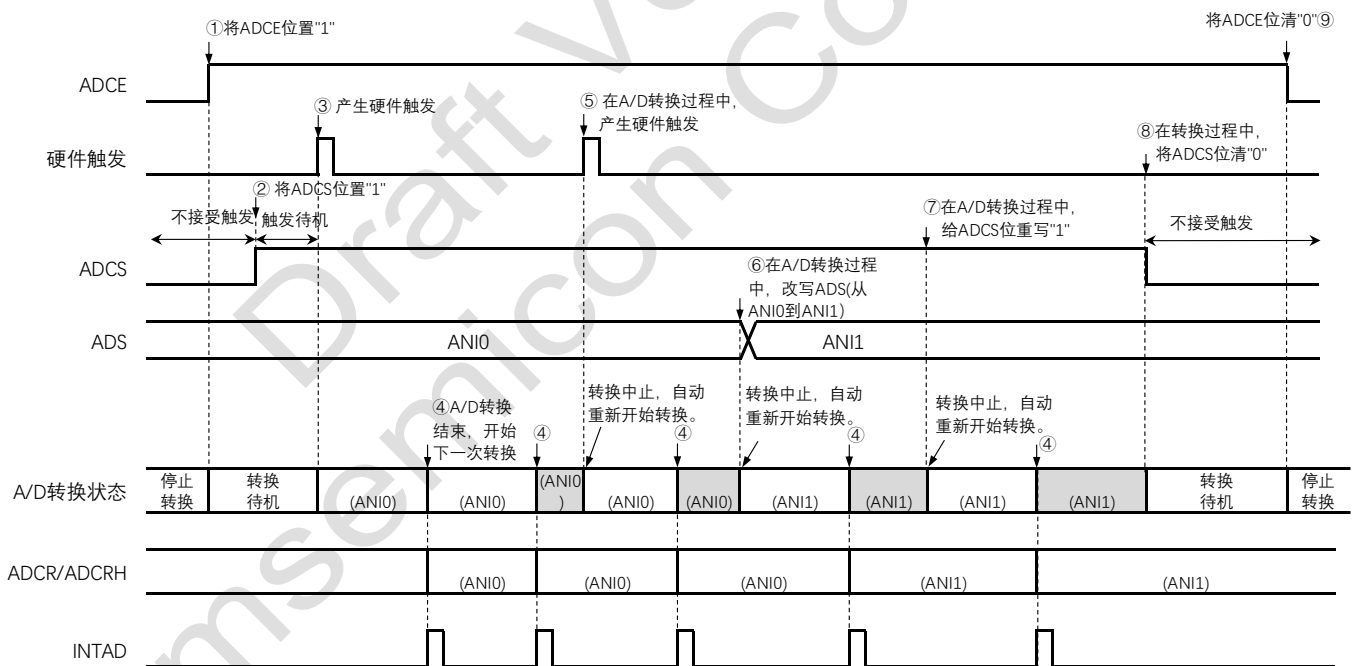
图13-22 软件触发模式(扫描模式、单次转换模式)的运行时序例子



### 13.4.5 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。

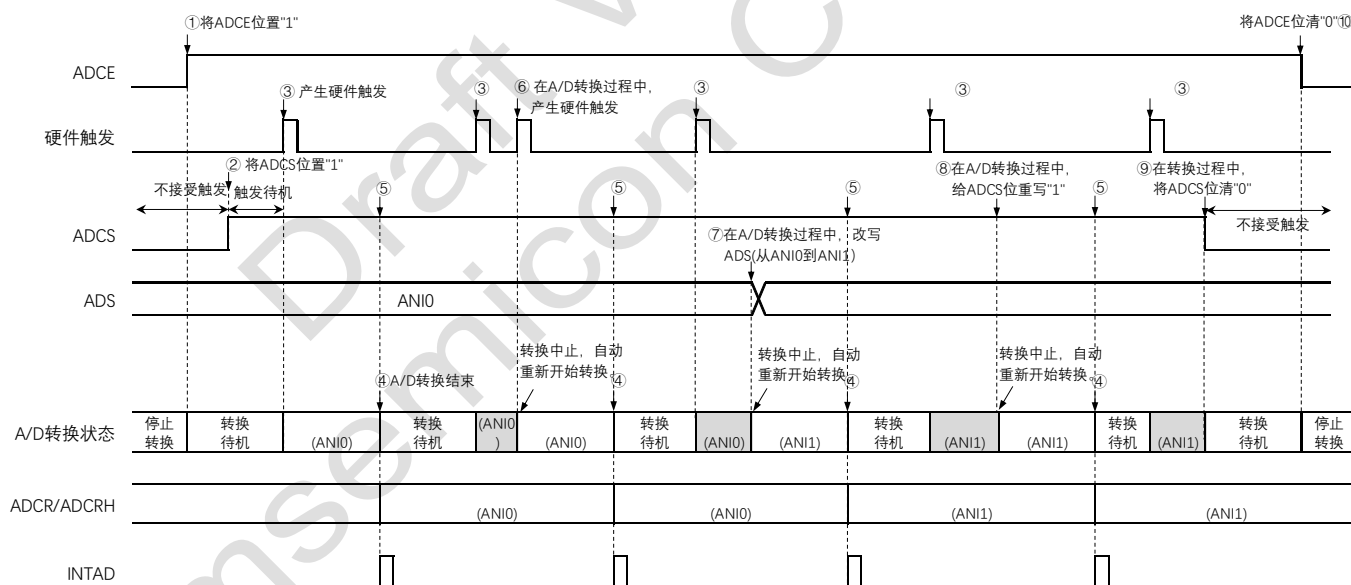
图13-23 硬件触发无等待模式(选择模式、连续转换模式)的运行时序例子



### 13.4.6 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。

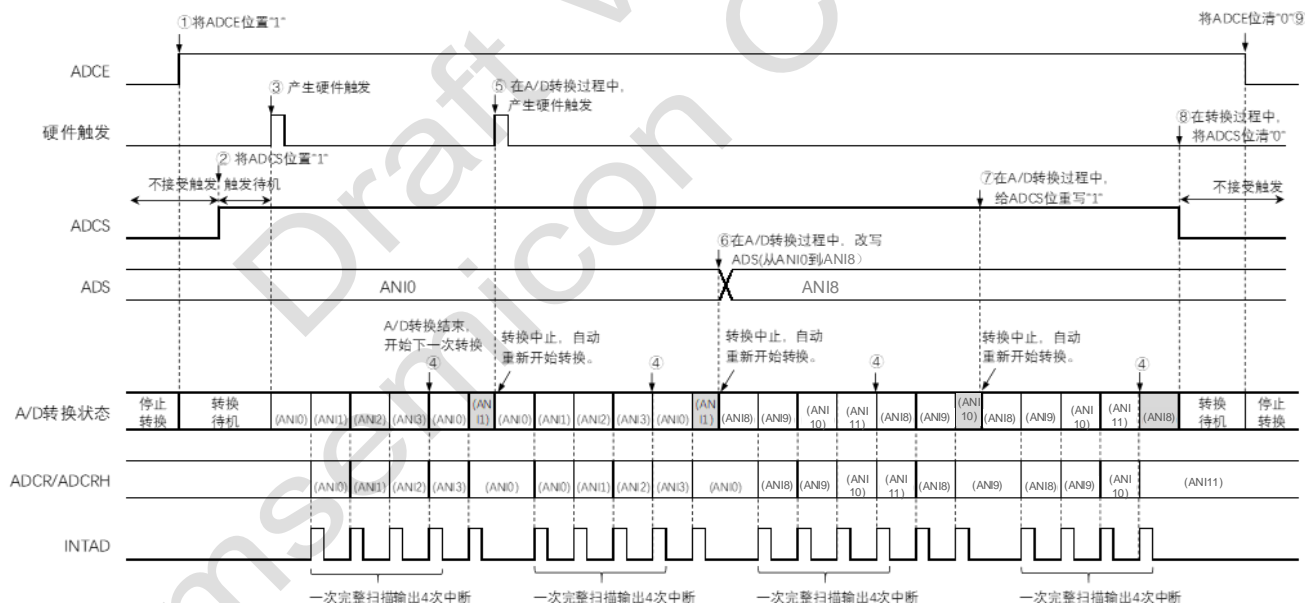
图13-24 硬件触发无等待模式(选择模式、单次转换模式)的运行时序例子



### 13.4.7 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。

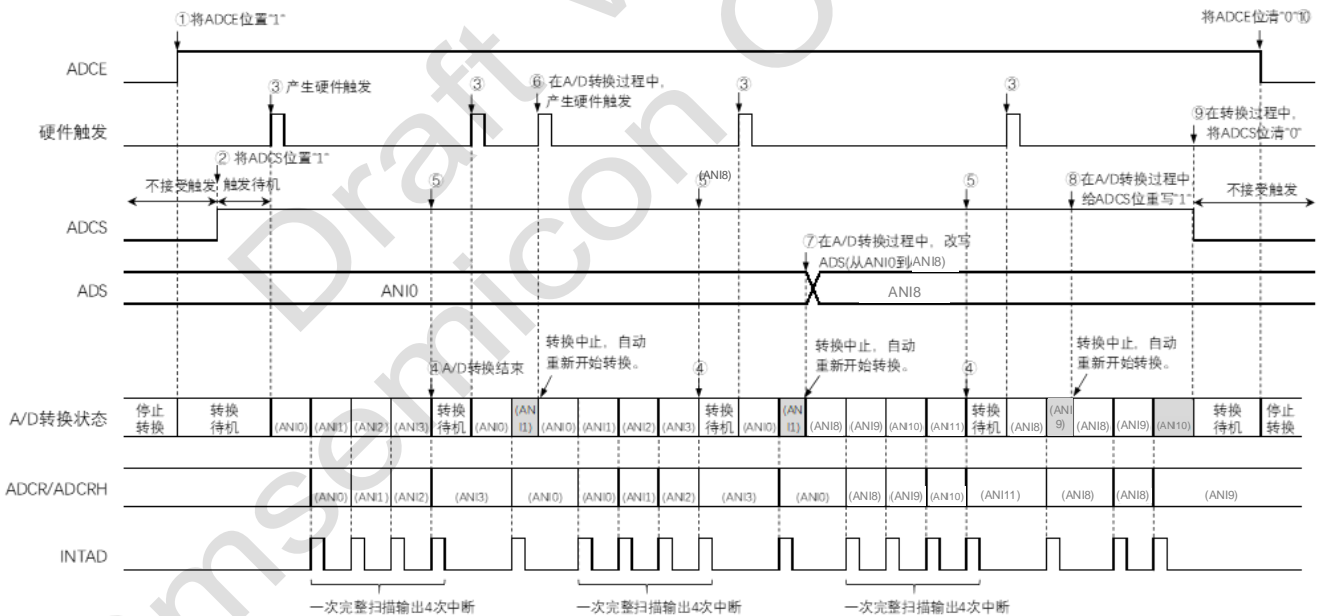
图13-25 硬件触发无等待模式(扫描模式、连续转换模式)的运行时序例子



### 13.4.8 硬件触发无等待模式(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)。
- ⑤ 在4个通道的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，然后从最初的通道重新开始转换。
- 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，然后从由ADS寄存器重新指定的最初通道进行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，然后重新从最初的通道开始转换。
- 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。
- 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。

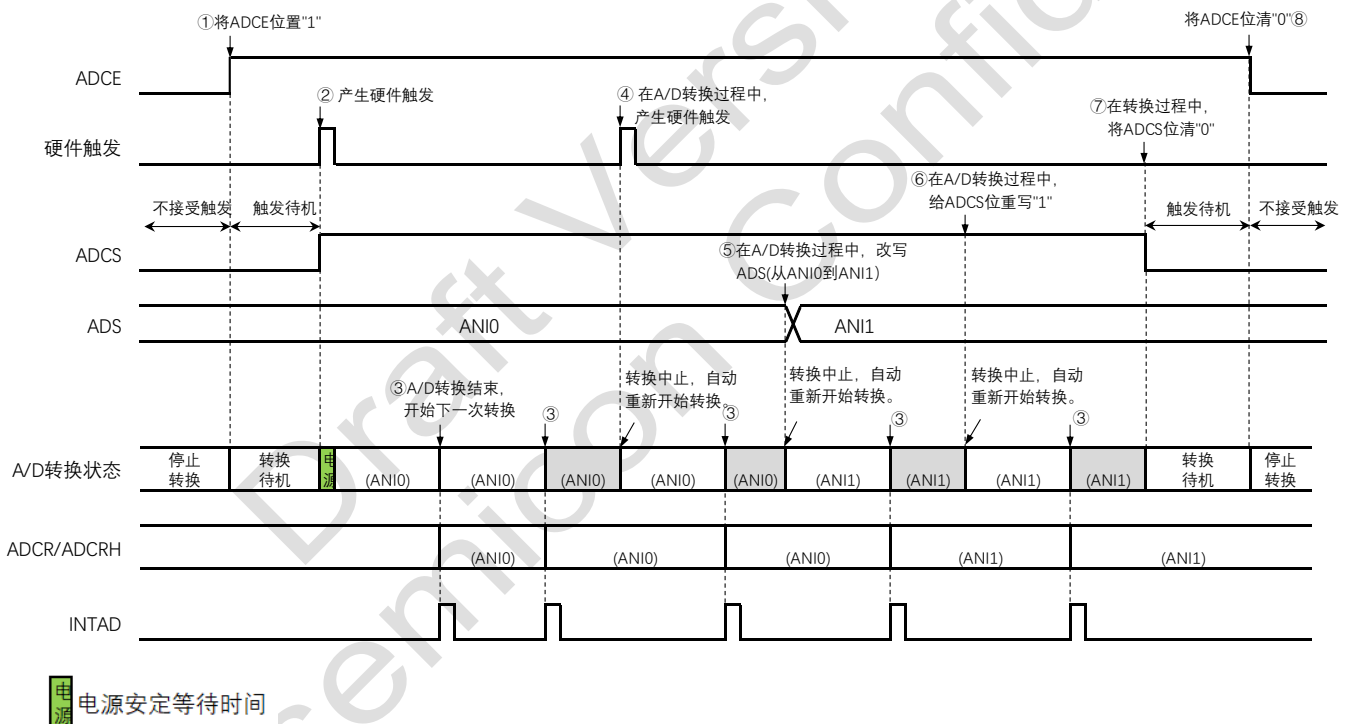
图13-26 硬件触发无等待模式(扫描模式、单次转换模式)的运行时序例子



### 13.4.9 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。

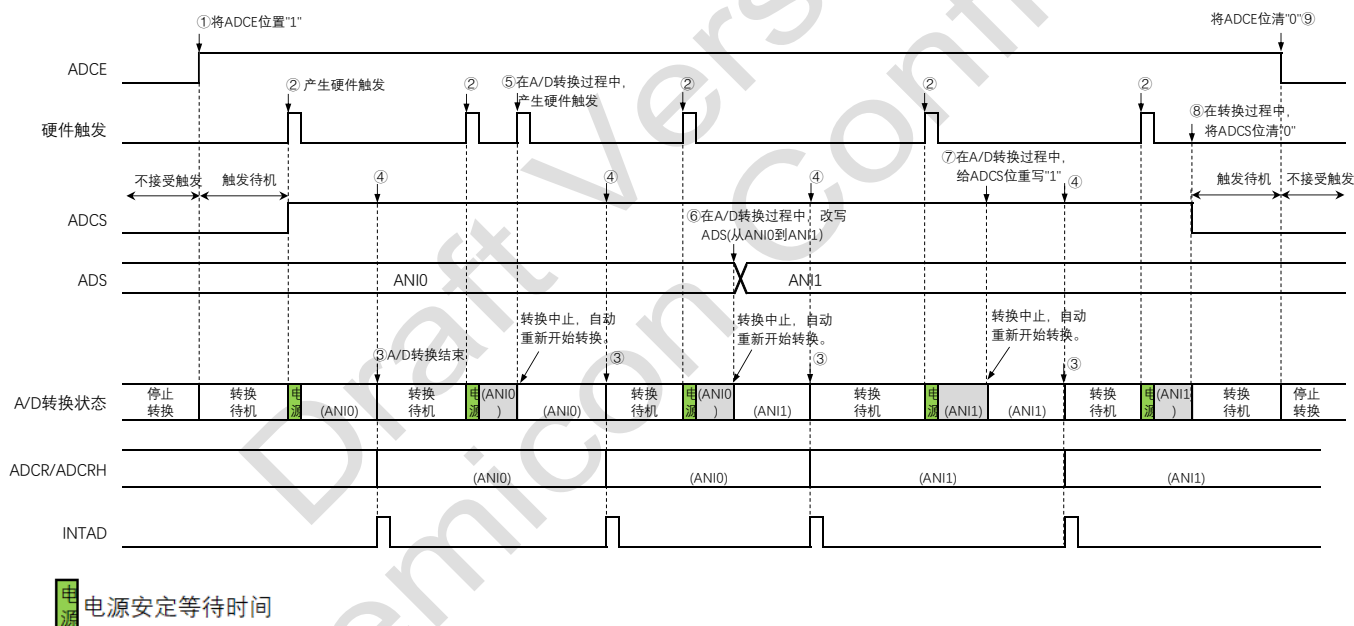
图13-27 硬件触发等待模式(选择模式、连续转换模式)的运行时序例子



### 13.4.10 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13-28 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



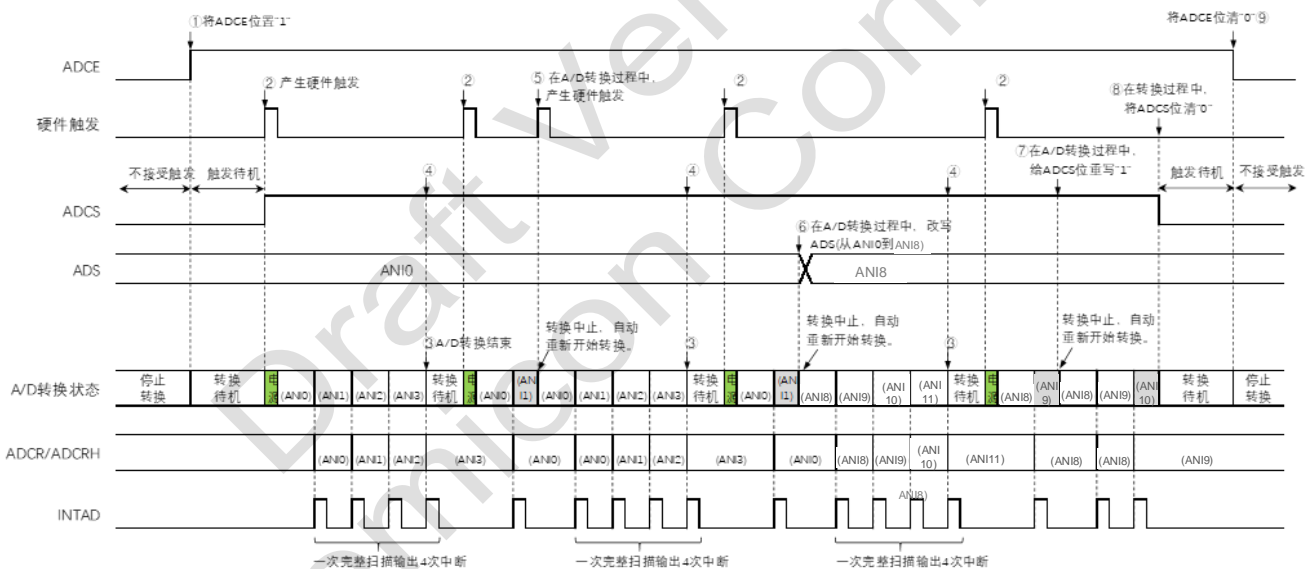




### 13.4.12 硬件触发等待模式(扫描模式、单次转换模式)

- 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13-30 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



电源安定等待时间

## 13.5 A/D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。

### 13.5.1 软件触发模式的设置

图13-31 软件触发模式的设置



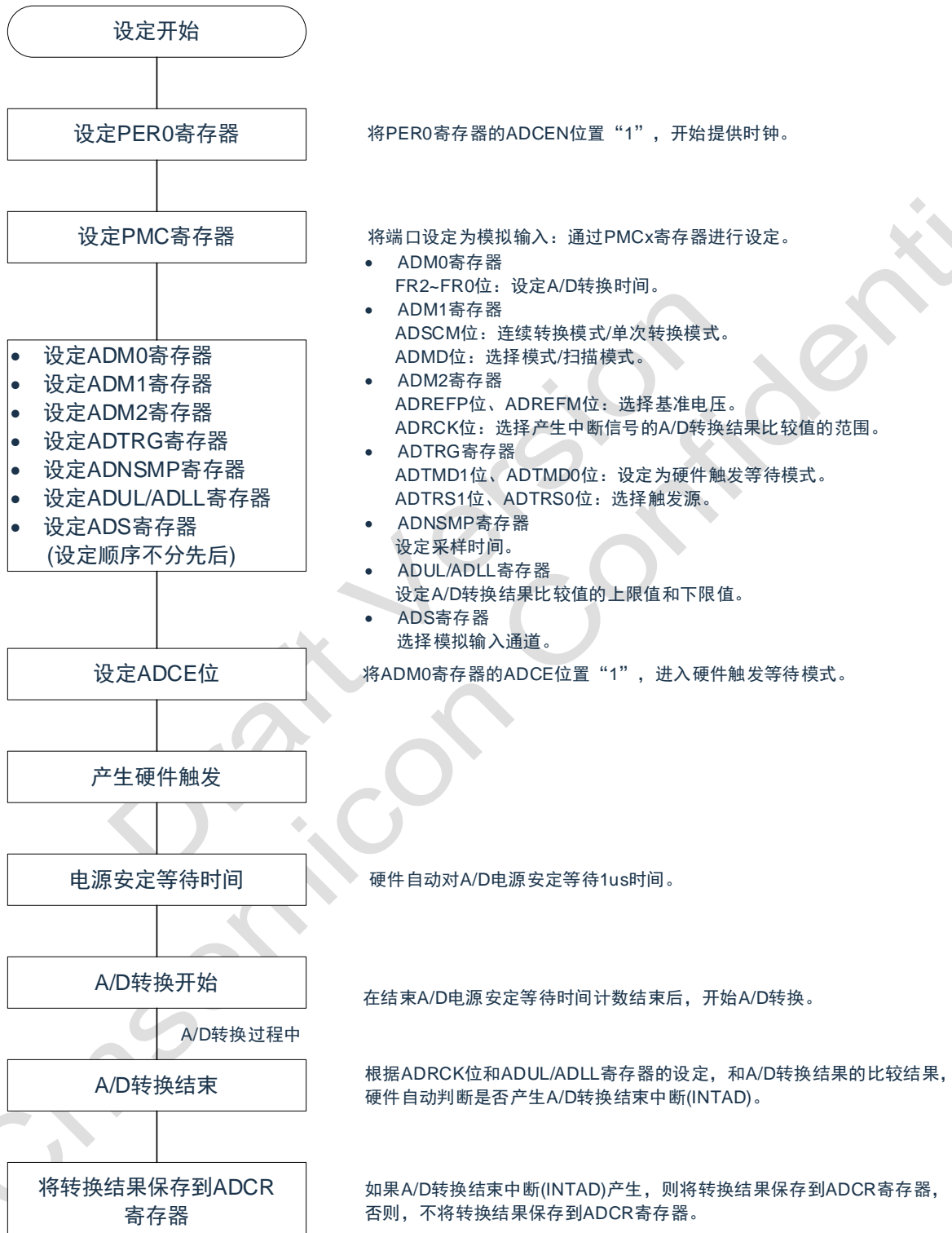
### 13.5.2 硬件触发无等待模式的设置

图13-32 硬件触发无等待模式的设置



### 13.5.3 硬件触发等待模式的设置

图13-33 硬件触发等待模式的设置



### 13.5.4 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13-34 选择温度传感器的输出电压/内部基准电压时的设置



### 13.5.5 测试模式的设置

图13-35 测试模式的设置(VSS/half\_VDD/VDD作为转换对象)



## 第14章 D/A转换器

### 14.1 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12位分辨率的转换器，能控制2个通道(ANO0、ANO1)的模拟输出。

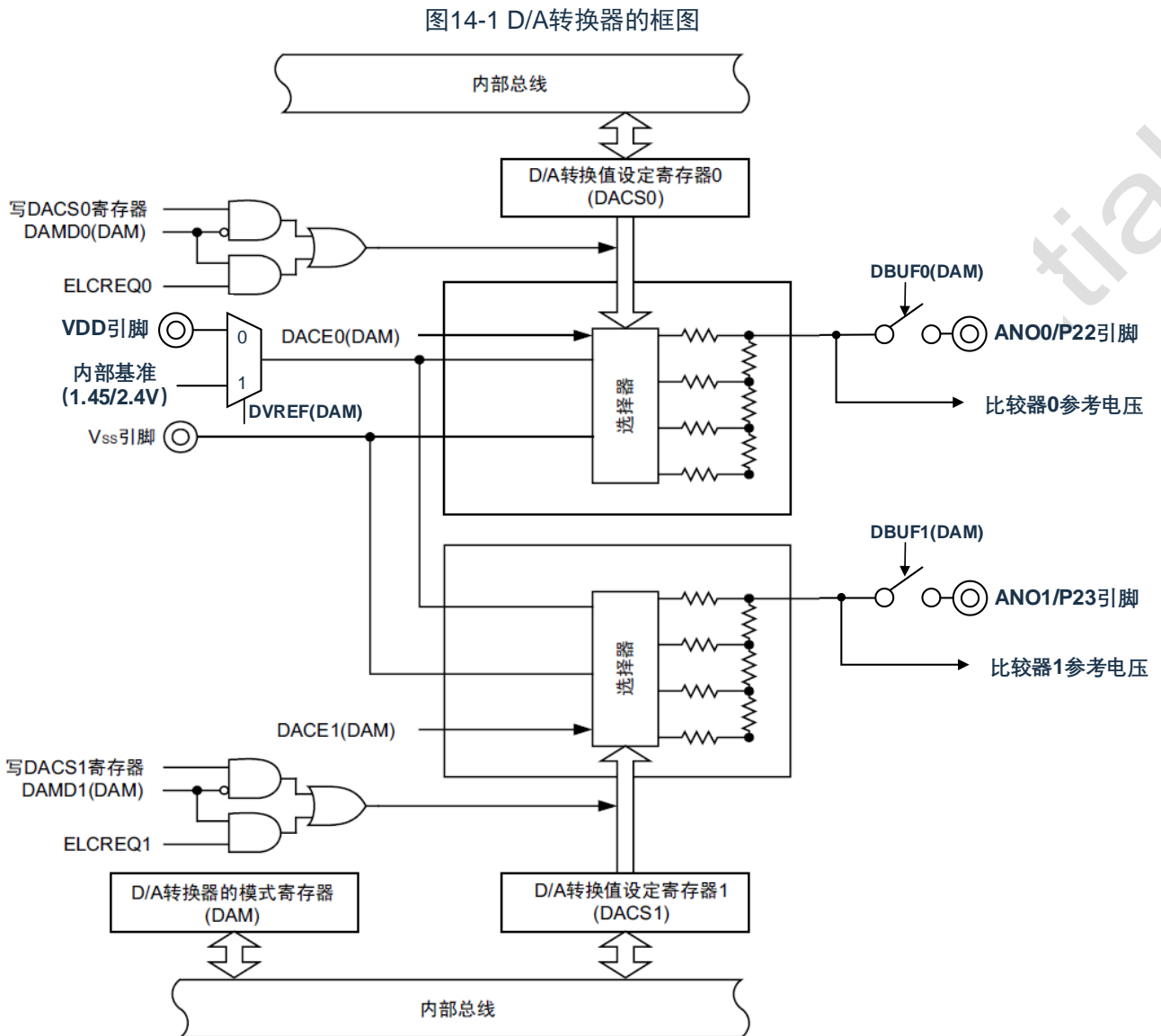
D/A转换器有以下功能：

- 12位分辨率×2ch
- R-2R梯形方式
- 模拟输出电压
  - 12位分辨率：参考电压×m12/4096(m12：给DACSi寄存器设置的值)
- 运行模式
  - 通常模式
  - 实时输出模式
- 复位保持功能
  - DRSTMDi = 1时，除上电复位以外，其它复位不能复位DA的寄存器，输出保持

备注：i=0、1

## 14.2 D/A转换器的结构

D/A转换器的框图如图14-1所示。



备注：ELCREQ0和ELCREQ1是用于实时输出模式的触发信号(EVENTC的事件信号)。



## 14.3 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
- 内部基准控制寄存器(VREFCTL)

### 14.3.1 外围允许寄存器1(PER1)

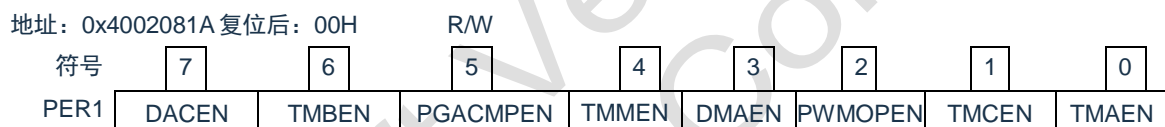
PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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14-2 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)。

### 14.3.2 内部基准控制寄存器(VREFCTL)

当D/A转换器选择内部基准作为参考电压时，也就是DVREF=1时，通过VREFCTL寄存器可以选择内部基准电压是1.45V或2.4V。

图14-3 内部基准控制寄存器的格式

地址：0x4004047A

复位值：00H R/W

	7	6	5	4	3	2	1	0
VREFCTL	0	0	0	0	0	0	VREFSEL	0

VREFSEL	内部基准电压选择
0	内部基准电压选择为 1.45V
1	内部基准电压选择为 2.4V

### 14.3.3 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16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“0000H”。

图14-4 D/A转换器的模式寄存器(DAM)的格式

地址：40044738H 复位后：0000HR/W

符号	15~9	8	7	6	5	4	3	2	1	0
DAM	—	DVREF	DRSTMD1	DRSTMD0	DBUF1	DBUF0	DACE1	DACE0	DAMD1	DAMD0

DAMDi	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

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。

DBUFi	D/A转换器端口输出buf的使能控制
0	D/A转换器端口输出缓冲器不使能，转换结果无法从端口输出
1	D/A转换器端口输出缓冲器使能，转换结果可以从端口输出

DRSTMDi	D/A转换器的复位模式选择
0	普通复位模式，任意复位都有效
1	除上电复位外，其它形式的复位对D/A转换器的输出数据无效

DVREF	参考电压选择
0	选择VDD作为D/A转换的参考电压
1	选择内部基准电压（1.45V/2.4V）作为D/A转换的参考电压

备注：i=0、1

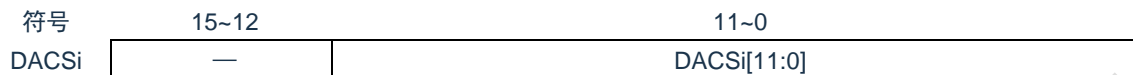
### 14.3.4 D/A转换值设置寄存器i(DACSi)(i=0、1)

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16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“0000H”。

图14-5 D/A转换值设置寄存器i(DACSi)(i=0、1)的格式

地址：40044734H(DACS0)、40044736H(DACS1) 复位后：0000H R/W



备注：D/A转换器的模拟输出电压(VANOi)如下：

$$VANOi = V_{REF} \times (DACSi) / 4096$$

$V_{REF}$ 由寄存器DAM的DVREF位确定。

当不使用D/A转换器时，为了减小不必要的电流消耗，必须将DACEi位置“0”(禁止输出)并且将DACSi寄存器置“0000H”，使R-2R的电阻没有电流流过。

### 14.3.5 事件输出目标选择寄存器n(ELSELRn)、n=00~22

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“23.3.1 事件输出目标选择寄存器n(ELSELRn)(n=00~22)”。

### 14.3.6 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。

## 14.4 D/A转换器的运行

### 14.4.1 通常模式的运行

以DACSi寄存器的写操作为启动触发，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 设置DAM寄存器(D/A转换器的模式寄存器)的DVREF和DRSTMDi位来分别选择参考电压和复位模式。
- ④ 将DAM寄存器(D/A转换器的模式寄存器)的DAMD<sub>i</sub>位置“0”(通常模式)。
- ⑤ 将DAM寄存器(D/A转换器的模式寄存器)的DBUF<sub>i</sub>位置“1”(允许转换结果从端口输出)。
- ⑥ 给DACSi寄存器(D/A转换值设置寄存器i)设置ANO<sub>i</sub>引脚输出的模拟电压值。

以上①~⑥为初始设置。

- ⑦ 将DAM寄存器的DACE<sub>i</sub>位置“1”(允许D/A转换)。  
开始D/A转换，在经过稳定时间后将⑥设置的模拟电压输出到ANO<sub>i</sub>引脚。
- ⑧ 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DACE<sub>i</sub>位置“0”(停止D/A转换)，就停止D/A转换。

注意：

1. 即使将DACE<sub>i</sub>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ANO<sub>i</sub>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

## 14.4.2 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 设置DAM寄存器(D/A转换器的模式寄存器)的DVREF和DRSTMDi位来分别选择参考电压和复位模式。
- ④ 将DAM寄存器(D/A转换器的模式寄存器)的DAMD<sub>i</sub>位置“0”(通常模式)。
- ⑤ 将DAM寄存器(D/A转换器的模式寄存器)的DBUF<sub>i</sub>位置“1”(允许转换结果从端口输出)。
- ⑥ 给DACSi寄存器(D/A转换值设置寄存器i)设置ANO<sub>i</sub>引脚输出的模拟电压值。
- ⑦ 将DAM寄存器的DACE<sub>i</sub>位置“1”(允许D/A转换)。  
开始D/A转换，在经过稳定时间后将⑥设置的模拟电压输出到ANO<sub>i</sub>引脚。
- ⑧ 通过事件输出目标选择寄存器n(ELSELR<sub>n</sub>, n=00~22)设置用于实时输出模式的触发信号。
- ⑨ 将DAM寄存器的DAMD<sub>i</sub>位置“1”(实时输出模式)。
- ⑩ 开始事件发生源的运行。

以上①~⑩为初始设置。

- ⑪ 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ANO<sub>i</sub>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ANO<sub>i</sub>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ANO<sub>i</sub>引脚输出的模拟电压值。

如果将DAM寄存器的DACE<sub>i</sub>位置“0”(停止D/A转换)，就停止D/A转换。

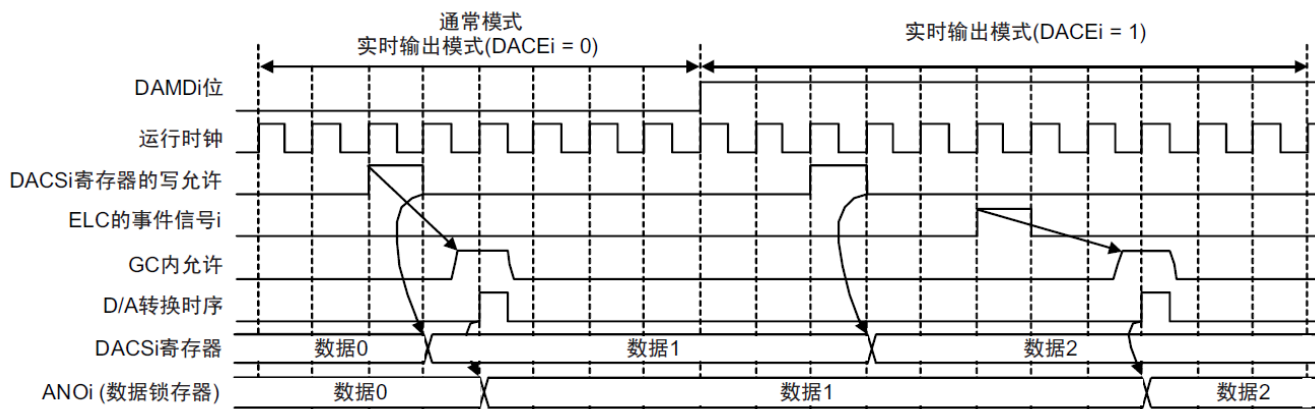
注意：

1. 即使将DACE<sub>i</sub>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ANO<sub>i</sub>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<sub>CLK</sub>时钟。如果以小于等于3个F<sub>CLK</sub>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。

### 14.4.3 D/A转换值的输出时序

D/A转换值的输出时序如图14-6所示。

图14-6 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)。

## 14.5 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PMC2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PMC2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DACS<sub>i</sub>寄存器的值。  
在触发信号有效期间, 不能更改DACS<sub>i</sub>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。
- (8) 当设置DAM寄存器中DRSTMD<sub>i</sub> ( $i=0, 1$ ) 位为1时, 对应通道的D/A转换结果只会受到上电复位的影响, 其它复位均不能使输出结果复位。



# 第15章 比较器

本产品内置2个通道的比较器。

## 15.1 比较器的功能

比较器有以下功能：

- CMP的输入引脚可选择外部端口，内部基准电压（1.45V/2.4V）以及DAC0/1提供的电压。
- 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15-1 比较器的功能概要

项目	内容
CMP	<ul style="list-style-type: none"> <li>• 2通道比较器(CMP0和CMP1)</li> <li>• 比较器负端可以选择DAC0/1电压或者内部基准电压（1.45V/2.4V） CMP0的负端可选外部引脚输入，DAC0的输出电压或内部基准电压 CMP1的负端可选外部引脚输入(4个)，DAC1的输出电压或内部基准电压</li> <li>• CMP0的正端可选择外部引脚输入(1个)或者PGA0的输出</li> <li>• CMP1的正端可以选择外部引脚输入(4个)</li> <li>• 正端输入电压&gt;负端输入电压时，输出高电平 正端输入电压&lt;负端输入电压时，输出低电平</li> <li>• 数字滤波器的滤波宽度可选</li> <li>• 输出反转功能</li> <li>• 比较结果可以从引脚(VCOUT0, VCOUT1)输出</li> <li>• 能检测比较器输出的有效边沿并且产生中断信号</li> <li>• 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</li> <li>• 与Timer4组合可输出TIMER WINDOW</li> <li>• 支持比较器正迟滞，负迟滞，和双边迟滞，迟滞电压可选20mV，40mV，60mV</li> </ul>

## 15.2 比较器的结构

比较器的框图如图15-1所示。

图15-1 比较器0的框图

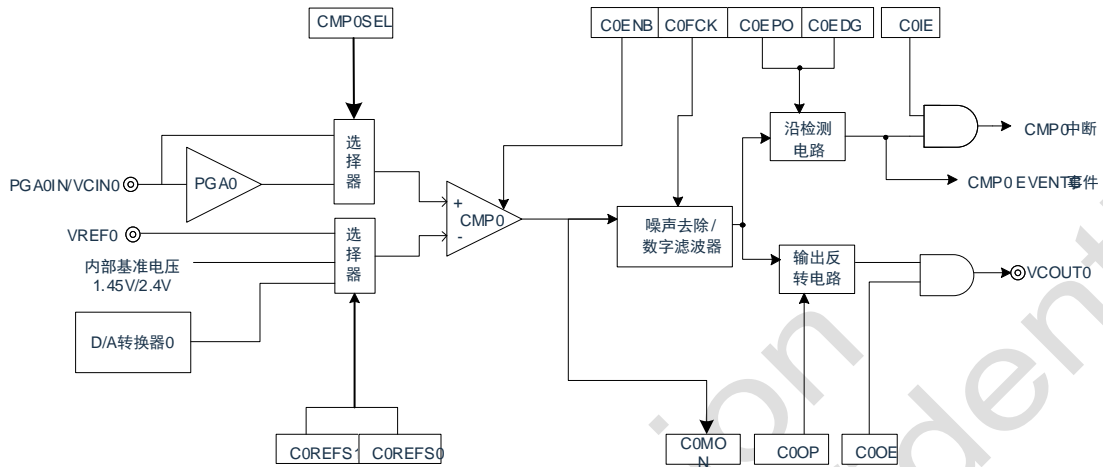
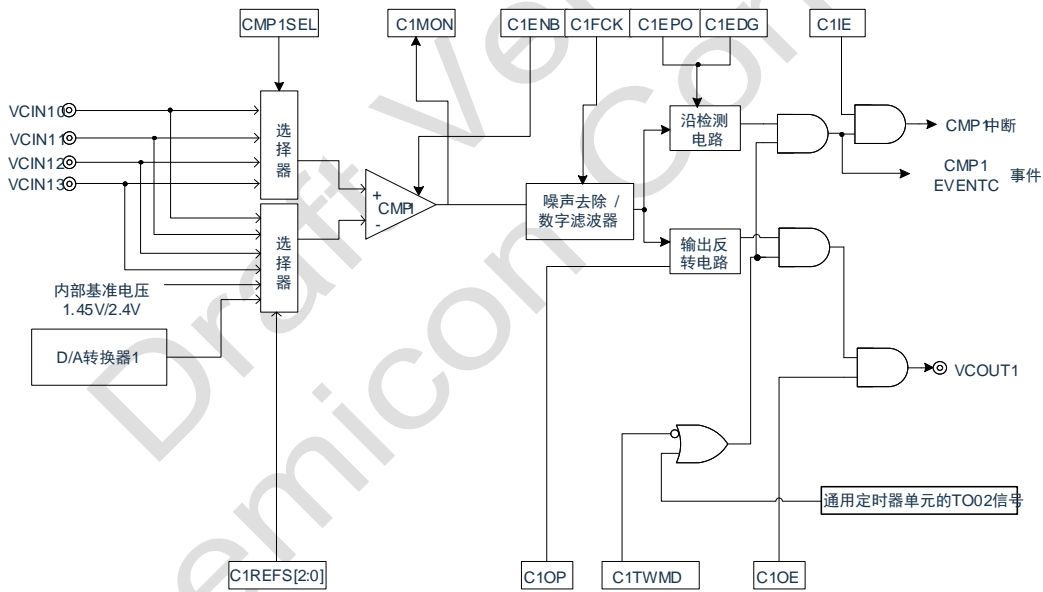


图15-2 比较器1的框图



## 15.3 控制比较器的寄存器

控制比较器的寄存器如表15-2所示。

表15-2 控制比较器的寄存器

寄存器名	符号
外围允许寄存器1	PER1
内部基准电压选择	VREFCTL
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
比较器0迟滞控制寄存器	CMP0HY
比较器1迟滞控制寄存器	CMP1HY
端口模式控制寄存器	PMCxx
端口模式寄存器	PMxx
端口寄存器	Pxx

### 15.3.1 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15-3 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)。

### 15.3.2 内部基准电压选择 (VREFCTL)

当比较器CMP0/1选择内部基准作为负端参考电压时，通过VREFCTL寄存器可以选择内部基准电压是1.45V或2.4V。

地址: 0x4004047A    复位值: 00H    R/W

	7	6	5	4	3	2	1	0
VREFCTL	0	0	0	0	0	0	VREFSEL	0

VREFSEL	内部基准电压选择
0	内部基准电压选择为 1.45V
1	内部基准电压选择为 2.4V

### 15.3.3 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DAC的输出电压，而DAC动作未使能时
- CMP0的输入选择PGA的输出，而PGA动作未使能时

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15-4 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 <sup>注1,2</sup>
0	比较器1的正端电压 < 比较器1的负端电压，或者比较器1停止运行。
1	比较器1的正端电压 > 比较器1的负端电压

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 <sup>注1,2</sup>
0	比较器0的正端电压 < 比较器0的负端电压，或者比较器0停止运行。
1	比较器0的正端电压 > 比较器0的负端电压

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。

### 15.3.4 比较器滤波控制寄存器(COMPFI<sub>R</sub>)

COMPFI<sub>R</sub>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COMPFI<sub>R</sub>寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图15-5 比较器滤波控制寄存器(COMPFI<sub>R</sub>)的格式

地址：40043841H      复位后：00H      R/W

符号	7	6	5	4	3	2	1	0
COMPFI <sub>R</sub>	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	比较器1边沿检测的选择 <sup>注1</sup>
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 <sup>注1</sup>
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 <sup>注1</sup>
0	0	比较器1没有滤波器。
0	1	比较器1有滤波器，通过F <sub>CLK</sub> 进行采样。
1	0	比较器1有滤波器，通过F <sub>CLK</sub> /8进行采样。
1	1	比较器1有滤波器，通过F <sub>CLK</sub> /32进行采样。

C0EDG	比较器0边沿检测的选择 <sup>注2</sup>
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 <sup>注2</sup>
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 <sup>注2</sup>
0	0	比较器0没有滤波器。
0	1	比较器0有滤波器，通过F <sub>CLK</sub> 进行采样。
1	0	比较器0有滤波器，通过F <sub>CLK</sub> /8进行采样。
1	1	比较器0有滤波器，通过F <sub>CLK</sub> /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。

Draft Version  
Cmsemicon Confidential

## 15.3.5 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DAC的输出电压，而DAC未使能时
- CMP0的输入选择PGA的输出，而PGA动作未使能时

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15-6 比较器输出控制寄存器(COMPOCR)的格式

地址：40043842H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
COMPOCR	C1OTWMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE
C1OTWMD	比较器1的TIMER WINDOW输出模式控制位 <sup>注1</sup>							
	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 <sup>注2、注3</sup>							
	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 <sup>注4</sup>							
	0	禁止比较器1的中断请求。						
	1	允许比较器1的中断请求。						
C0OP	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。						
C0OE	VCOUT0引脚输出的允许 <sup>注5、注6、注7</sup>							
	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。						
C0IE	比较器0中断请求的允许 <sup>注8</sup>							
	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: 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5: 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6: 设置C0OE位, C0OP位, 将比较器0的结果输入到PWM选项单元, 来控制PWM输出的强制切断。
- 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。

### 15.3.6 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15-7 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PGA0输出信号

C0REFS1	C0REFS0	比较器0的负端输入信号选择位
0	0	选择DAC0的输出
0	1	选择内部基准电压 (1.45V/2.4V)
1	0	选择外部引脚(VREF0引脚)
1	1	禁止设置

### 15.3.7 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15-8 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	CMP1SELO	0	0	0	C0REFS1	C0REFS1	C0REFS0

CMP1SEL1	CMP1SELO	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DAC1的输出
0	0	1	选择内部基准电压 (1.45V/2.4V)
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。

### 15.3.8 比较器0的迟滞控制寄存器 (CMP0HY)

CMP0HY寄存器是比较器0的迟滞功能控制寄存器。

在比较器0停止动作(COENB=0)时, 改写CMP0HY寄存器。

通过8位存储器操作指令设定CMP0HY寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图15-9 比较器0的迟滞控制寄存器 (CMP0HY) 的格式

地址: 4004384EH 复位后: 00H R/W

	符号	7	6	5	4	3	2	1	0
CMP0HY		0	0	C0HYSVS1	C0HYSVS0	0	0	C0HYSLS1	C0HYSLS0

C0HYSVS1	C0HYSVS0	比较器0的迟滞电压选择位
0	0	无迟滞
0	1	20mV
1	0	40mV
1	1	60mV

C0HYSLS1	C0HYSLS0	比较器0的迟滞方式选择位
0	0	无迟滞
0	1	正迟滞
1	0	负迟滞
1	1	双边迟滞

### 15.3.9 比较器1的迟滞控制寄存器 (CMP1HY)

CMP1HY寄存器是比较器1的迟滞功能控制寄存器。

在比较器1停止动作(C1ENB=0)时, 改写CMP1HY寄存器。

通过8位存储器操作指令设定CMP1HY寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图15-10 比较器1的迟滞控制寄存器 (CMP1HY) 的格式

地址: 4004384FH

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CMP1HY	0	0	C1HYSVS1	C1HYSVS0	0	0	C1HYSLS1	C1HYSLS0

C1HYSVS1	C1HYSVS0	比较器1的迟滞电压选择位
0	0	无迟滞
0	1	20mV
1	0	40mV
1	1	60mV

C1HYSLS1	C1HYSLS0	比较器1的迟滞方式选择位
0	0	无迟滞
0	1	正迟滞
1	0	负迟滞
1	1	双边迟滞

### 15.3.10 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“第二章 引脚功能”

## 15.4 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, DAC0/1可以组合起来联动。

比较器联动运行的设置步骤举例, 如表15-3所示。

表15-3 比较器相关寄存器的设置步骤

步骤	寄存器	位	设置值
1	PGA0CTL	PGAVG0/1/2	选择增益 <sup>注3</sup>
2	PGA0CTL	PGAMODE	0(Vss引脚选择) <sup>注3</sup>
3	PGA0CTL	PGAEN	1(允许运行) <sup>注3</sup>
4	等待PGA的稳定时间(最小10us)		
5	COMPSELi	CMP0SEL/ CMP1SELi	比较器i正端输入选择
6	COMPSELi	CIREFS	比较器i负端输入选择
7	DACSi	DACSi[11:0]	DACi模块输出电压设置
8	DAM	DAM	DACi模式设置
9	DAM	DACEi	DACi允许运行
10	等待DAC输出稳定时间		
11	将VCIN0、VCIN1x、VREF0引脚(输入), PGAI(输入) <sup>注3</sup> 设置为模拟输入功能。即PMCxx位置“1”(模拟输入)。		
12	COMPMDR	CIENB	1(允许运行)
13	等待比较器的稳定时间(最小3us)		
14	COMPFIR	CI FCK	使用或者不使用数字滤波器, 选择采样时钟。
		CI EOP、CI EDG	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	CI OP、CI OE	设置VCOUTi的输出(选择极性, 设置允许或者禁止输出)。
		CI IE	设置允许或者禁止中断请求的输出。
		C1 OTWMD	设置比较器1的TIMER WINDOW输出许可/禁止
16	MKxx <sup>注1</sup>	MKL	使用中断时: 选择屏蔽中断。
17	IFxx <sup>注1</sup>	IFL	使用中断时: 0(无中断请求: 初始化) <sup>注2</sup>

注1: MKxx, IFxx是比较器的中断控制寄存器, 详细内容请参考“第24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

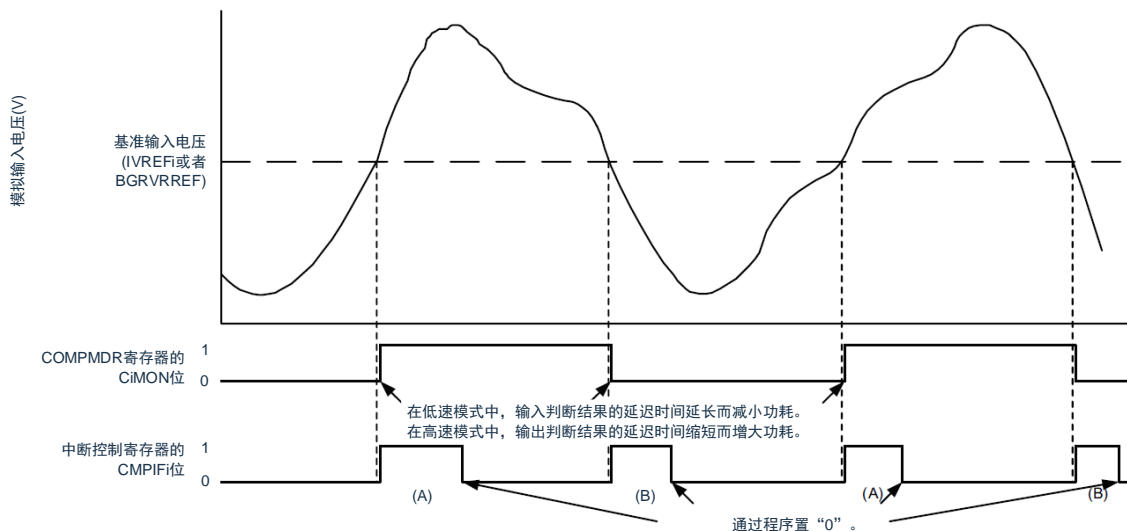
备注: i=0、1, x=0-3

比较器*i*(*i*=0、1)的运行例子如图15-11所示。在基本模式中，当正端模拟输入电压高于负端模拟输入电压时，COMPMDR寄存器的CiMON位为“1”；当正端模拟输入电压低于负端模拟输入电压时，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“15.4.2 比较器*i*中断(*i*=0、1)”。

图15-11 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COMPDIR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)。

### 15.4.1 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15-12是比较器i的数字滤波器的结果，图15-13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15-12 比较器i(i=0、1)的数字滤波器和沿检出结构

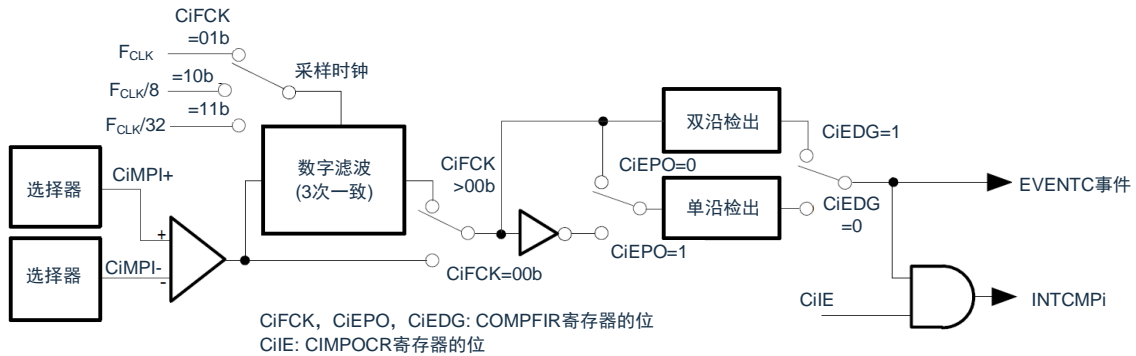
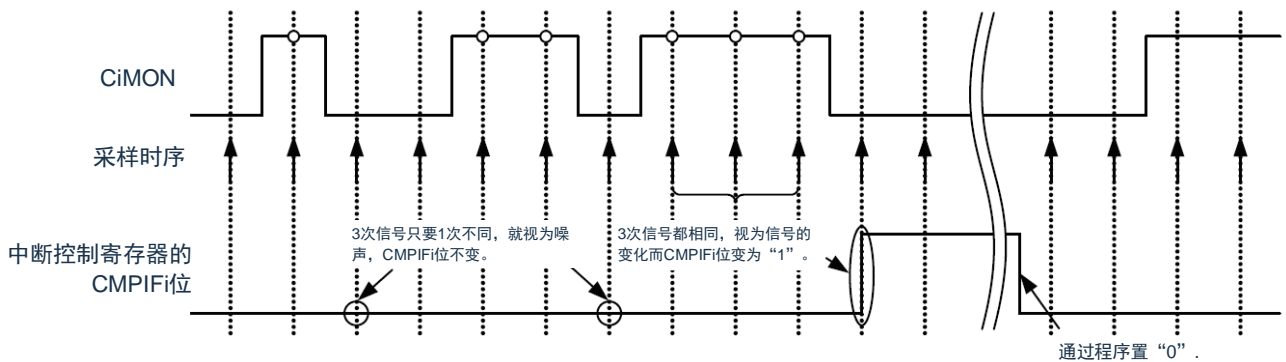


图15-13: 比较器i(i=0、1)的数字滤波器和中断运行例子



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。



## 15.4.2 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。

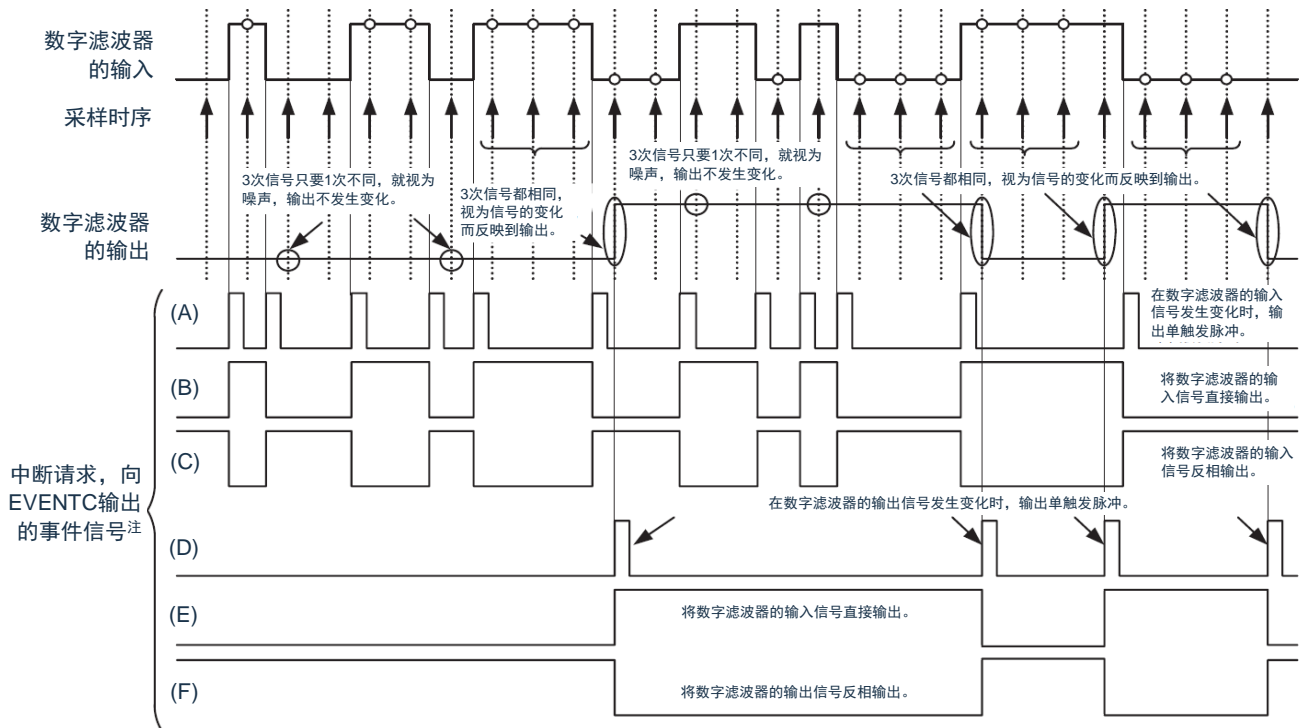
要使用比较器i中断时，必须将COMPOCR寄存器的CiIE位置“1”(允许中断请求的输出)。通过COMPFIR寄存器设置中断请求的产生条件，还能给比较器的输出附加数字滤波器。数字滤波器可选择3种采样时钟。有关寄存器的设置和产生中断请求的对应，请参照“15.3.4 比较器滤波控制寄存器(COMPFIR)”和“15.3.5 比较器输出控制寄存器(COMPOCR)”。

Draft Version  
Cmsemicon Confidential

### 15.4.3 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15-14 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。

## 15.4.4 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“15.3.5 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15-3 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)。

## 15.4.5 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15-3的步骤设置寄存器。

注意：

1. 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)而且比较器的正端电压 > 比较器的负端电压。
    - 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)而且比较器的正端电压 < 比较器的负端电压。
- (n=0、1)

## 第16章 可编程增益放大器(PGA)

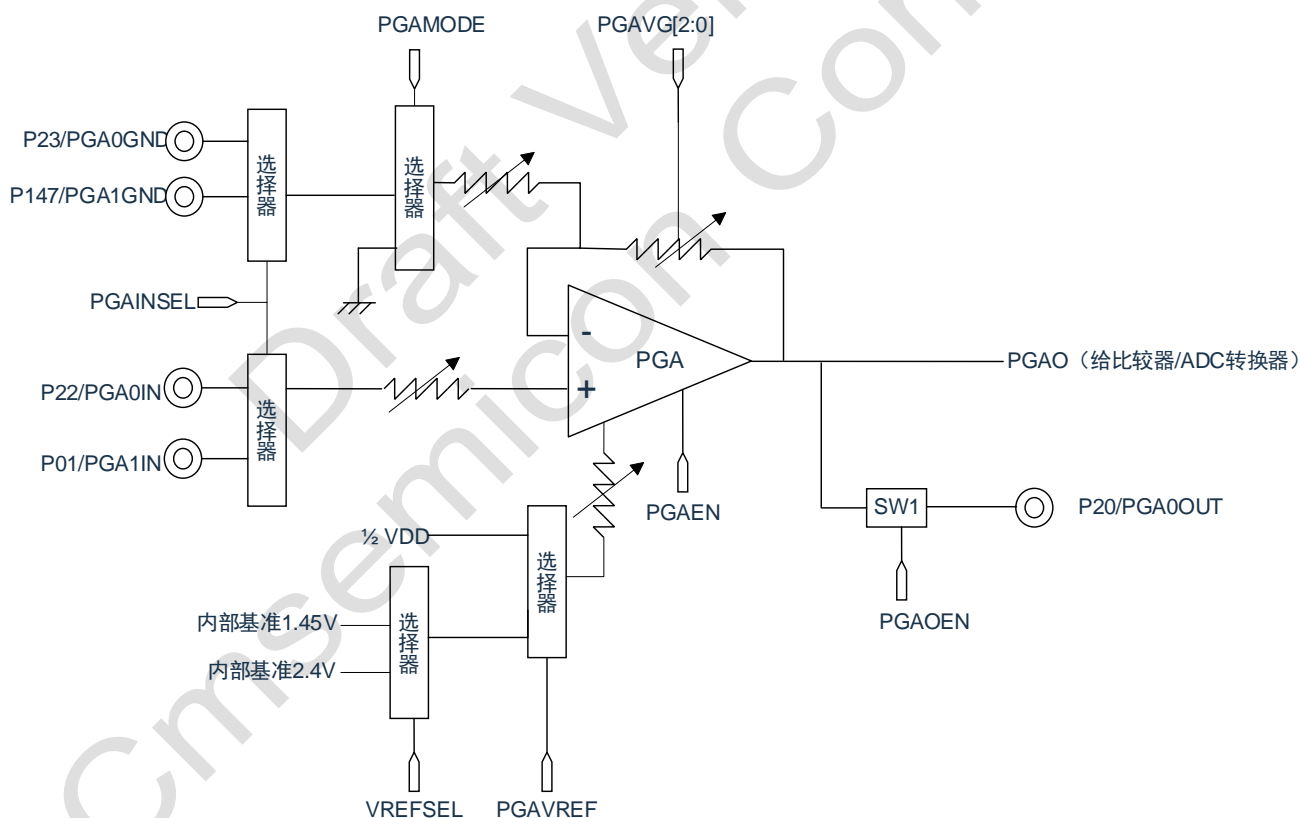
### 16.1 可编程增益放大器的功能

本产品内置一路可编程增益放大器(PGA0)，具有如下功能：

- 每路PGA的放大增益有7种选择：1倍，2.5倍，4倍，8倍，10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(CMP0)正端的模拟输入
- 参考电压可选1/2 VDD或者内部基准电压（1.45V/2.4V）
- $PGA输出 = 输入 * 增益 + 参考电压$

### 16.2 可编程增益放大器的结构

图16-1 可编程增益放大器的框图



## 16.3 可编程增益放大器的寄存器

表16-1 控制可编程增益放大器的寄存器

外围允许寄存器1	PER1
内部基准电压选择	VREFCTL
可编程增益放大器控制寄存器	PGA0CTL
端口模式控制寄存器2	PMC2
端口模式寄存器2	PM2

### 16.3.1 外围允许寄存器1(PER1)

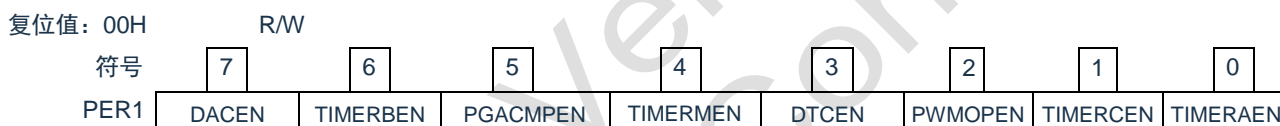
PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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16-2 外围允许寄存器1(PER1)的格式



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)

### 16.3.2 内部基准电压选择 (VREFCTL)

当PGA选择内部基准作为参考电压时，通过VREFCTL寄存器可以选择内部基准电压是1.45V或2.4V。

地址: 0x4004047A

复位值: 00H      R/W

7	6	5	4	3	2	1	0
0	0	0	0	0	0	VREFSEL	0

VREFSEL	内部基准电压选择
0	内部基准电压选择为 1.45V
1	内部基准电压选择为 2.4V

### 16.3.3 可编程增益放大器控制寄存器(PGA0CTL)

PGA0CTL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寄存器。在产生复位信号后，此寄存器复位值为00H。

图16-3 PGA控制寄存器(PGA0CTL)的格式

复位值：00H      R/W

	7	6	5	4	3	2	1	0
PGA0CTL	PGAEN	PGA0EN	PGAINSEL	PGAVREF	PGAMODE	PGAVG2	PGAVG1	PGAVG0

n=0,1

PGAEN	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

PGA0EN	PGAOUT 从引脚输出使能
0	不允许 PGAOUT 从引脚 P20 输出
1	允许 PGAOUT 从 P20 引脚输出

PGAINSEL	可编程增益放大器输入选择
0	选择 PGA0IN (和 PGA0GND) 作为放大器的输入
1	选择 PGA1IN (和 PGA1GND) 作为放大器的输入

PGAMODE	PGA 模式选择
0	单端模式：选择 Vss 作为反馈电阻地
1	差分模式：选择 PGAnGND 引脚作为反馈电阻地

PGAVREF	参考电压选择
0	选择 1/2 VDD 作为 PGA 的参考电压
1	选择内部基准电压 (1.45V/2.4V) 作为 PGA 的参考电压

PGAVG2	PGAVG1	PGAVG0	PGA0 增益
0	0	0	1 倍
0	0	1	2.5 倍
0	1	0	4 倍
0	1	1	8 倍
1	0	0	10 倍
1	0	1	16 倍
1	1	0	32 倍
其他			禁止设置

注意：PGAEN设置为1后，可编程增益放大器运行需要10us的稳定时间。

$$\text{PGA输出电压} = \text{PGA输入电压} * \text{增益} + \text{参考电压}$$

### 16.3.4 控制模拟输入引脚端口功能的寄存器

在将 PGA0IN 、PGA0GND 引脚或者PGA1IN 引脚、PGA1GND 引脚或PGAOUT引脚用作可编程增益放大器的模拟输入/输出时，必须将各端口对应的端口模式寄存器 (PMxx)的位和端口数字/模拟控制寄存器 (PMCxx)的位置“1”。

Draft Version  
Cmsemicon Confidential

## 16.4 可编程增益放大器的运行

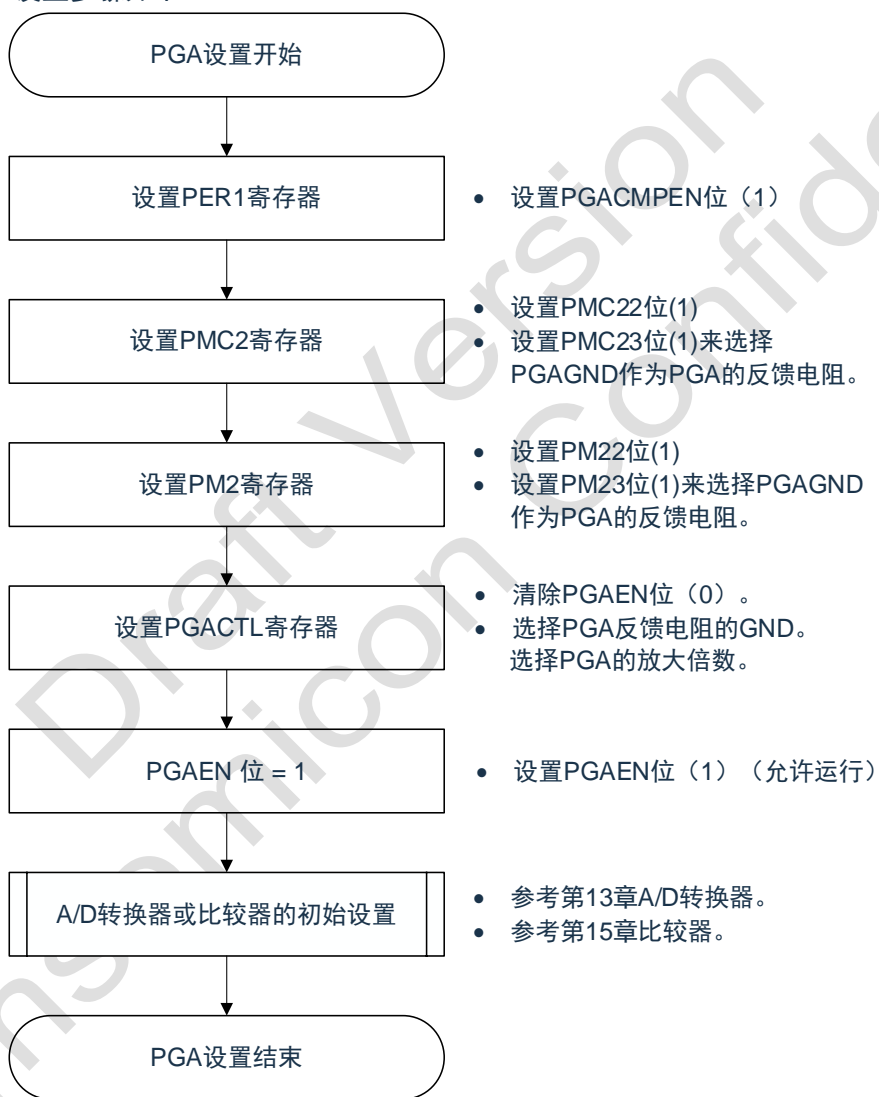
对PGA<sub>IN</sub>引脚输入的模拟电压进行放大，放大增益有7种选择：1倍，2.5倍，4倍，8倍，10倍，16倍，32倍。

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。

### 16.4.1 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：

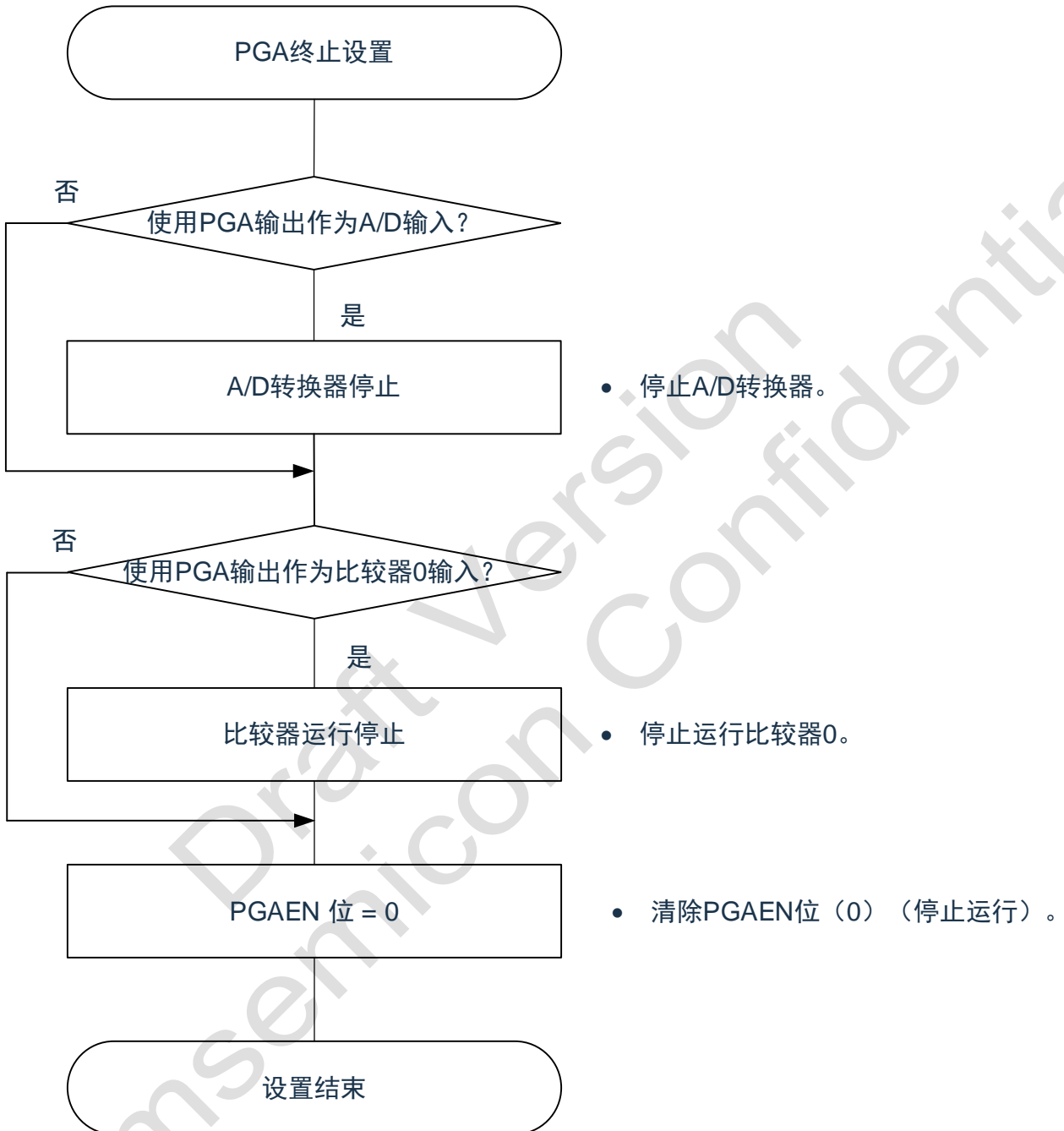


注意：在设置PGAEN位为1后需要10us的PGA稳定时间。之后再启动A/D转换。



## 16.4.2 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17章 通用串行通信单元

本产品搭载2个通用串行通信单元，每个单元有2个串行通道，各通道能实现3线串行（SSPI）、UART和简易I<sup>2</sup>C的通信功能。

32pin产品支持的通道功能分配如下：

单元	通道	用作 SSPI	用作 UART	用作简易 I <sup>2</sup> C
0	0	SSPI00	UART0 (支持 LIN-bus)	IIC00
	1	-		-
1	0	SSPI20	UART2	IIC20
	1	-		-

在单元0的通道0和通道1使用UART0时，不能使用SSPI00和IIC00。

在单元1的通道0和通道1使用UART2时，不能使用SSPI20和IIC20。

## 17.1 通用串行通信单元的功能

本产品支持的各串行接口的特征如下所示。

### 17.1.1 3线串行I/O (SSPI00、SSPI20)

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。

有关具体的设定例子, 请参照“17.5 3线串行I/O (SSPI00、SSPI20) 通信的运行”。

[数据的发送和接收]

- 7位~16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定

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设定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  
主控通信:  $\text{Max.F}_{\text{CLK}}/2$   
从属通信:  $\text{Max.F}_{\text{MCK}}/6$

[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 ( $T_{\text{KCY}}$ ) 特性的范围内使用。详细内容请参照数据手册。

## 17.1.2 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。

有关具体的设定例子, 请参照“17.7 UART (UART0、UART2) 通信的运行”。

[数据的发送和接收]

- 7位、8位、9位或者16位的数据长度
- MSB/LSB优先的选择
- 发送和接收数据的电平设定、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

### 17.1.3 简易I<sup>2</sup>C (IIC00、IIC20)

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。因此简易I<sup>2</sup>C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。有关具体的设定例子，请参照“17.9简易I<sup>2</sup>C (IIC00、IIC20) 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<sup>注</sup>、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RW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I<sup>2</sup>C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“17.9.3(2) 处理流程”。

备注：在使用全功能的I<sup>2</sup>C总线时，请参照“第18章 串行接口IICA”。

## 17.2 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17-1 通用串行通信单元的结构

项目	结构
移位寄存器	16位
缓冲寄存器	串行数据寄存器mn (SDRmn) 注
串行时钟输入/输出	SCLK00、SCLK20引脚（用于3线串行I/O）、 SCL00、SCL20引脚（用于简易I <sup>2</sup> C）
串行数据输入	SDI00、SDI20引脚（用于3线串行I/O）、 RxD0、RxD2引脚（用于UART）
串行数据输出	SDO00、SDO20引脚（用于3线串行I/O）、 TxD0、TxD2引脚（用于UART）
串行数据输入/输出	SDA00、SDA20引脚（用于简易I <sup>2</sup> 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	<单元设定部的寄存器> <ul style="list-style-type: none"> <li>•外围允许寄存器0 (PER0)</li> <li>•串行时钟选择寄存器m (SPSm)</li> <li>•串行通道允许状态寄存m (SEm)</li> <li>•串行通道开始寄存器m (SSm)</li> <li>•串行通道停止寄存器m (STm)</li> <li>•串行输出允许寄存器m (SOEm)</li> <li>•串行输出寄存器m (SOM)</li> <li>•串行输出电平寄存器m (SOLm)</li> <li>•串行待机控制寄存器m (SSCm)</li> <li>•从属选择功能启用寄存器m (SSEm)</li> <li>•输入切换控制寄存器 (ISC)</li> <li>•噪声滤波器允许寄存器0 (NFEN0)</li> </ul>
	<各通道部的寄存器> <ul style="list-style-type: none"> <li>•串行数据寄存器mn (SDRmn)</li> <li>•串行模式寄存器mn (SMRmn)</li> <li>•串行通信运行设定寄存器mn (SCRmn)</li> <li>•串行状态寄存器mn (SSRmn)</li> <li>•串行标志清除触发寄存器mn (SIRmn)</li> </ul>
	<ul style="list-style-type: none"> <li>•端口输出模式寄存器 (POMxx)</li> <li>•端口模式寄存器 (PMxx)</li> <li>•端口寄存器 (Pxx)</li> </ul>

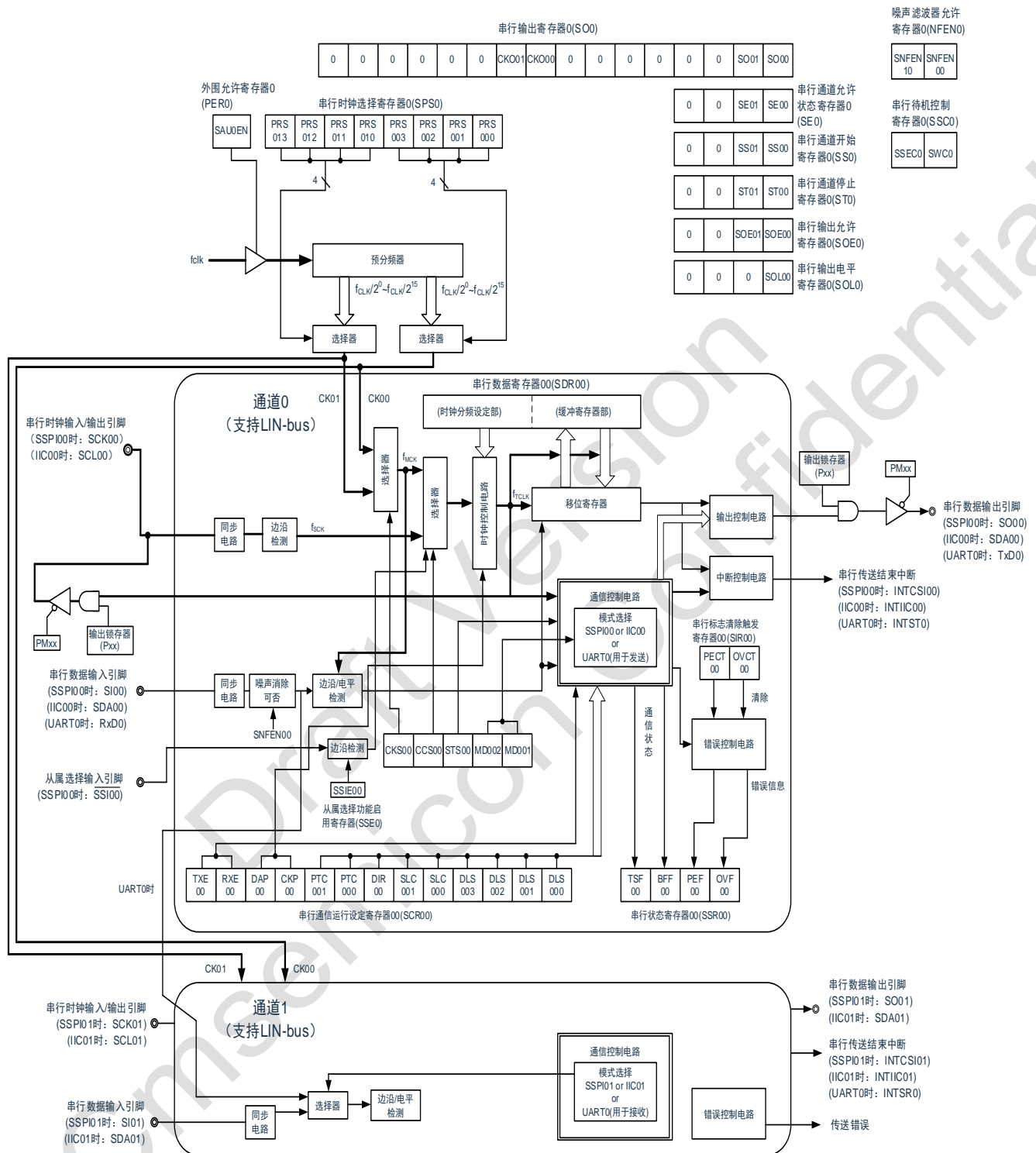
注：SEmn=1的动作期间。

备注：m：单元号 (m=0、1) n：通道号 (n=0、1) p：SSPI号 (p=00、20)

q：UART号 (q=0、2) r：IIC号 (r=00、20)

通用串行通信单元的框图如图17-1所示。(以单元0为例)

图17-1 通用串行通信单元0的框图



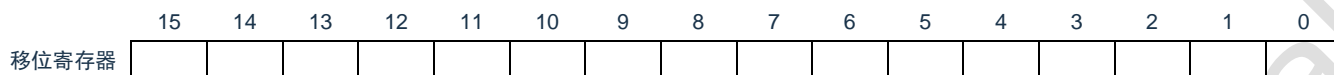
注：单元0, 1的结构相同

## 17.2.1 移位寄存器

这是进行并行和串行相互转换的16位寄存器。

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。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，在运行期间（SEmn=1）使用串行数据寄存器mn（SDRmn）。



## 17.2.2 串行数据寄存器mn（SDRmn）

SDRmn寄存器是通道n的发送和接收数据寄存器（16位）。

当运行停止（SEmn=0）时，bit15~9用作运行时钟（F<sub>MCK</sub>）的分频设定寄存器。在运行期间（SEmn=1）bit15~9用作发送接收缓冲寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到串行数据寄存器SDRmn；在发送数据时，将被传送到移位寄存器的发送数据设置到串行数据寄存器SDRmn。

与数据的输出顺序无关，根据串行通信运行设定寄存器mn（SCRmn）的bit3~bit0（DLSmn3~DLSmn0）的设定，保存到SDRmn寄存器的数据如下所示：

- 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）
- 16位数据长度（保存在SDRmn寄存器的bit0~15）

能以16位为单位读写SDRmn寄存器。

SEmn=1时，SDRmn寄存器的低8位可以作为SDRmnL以8位为单位进行读写<sup>注</sup>。

根据通信模式，能用以下SFR名称，读写SDRmnL寄存器。

- SSPIp通信.....SDIOpL
- 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”。

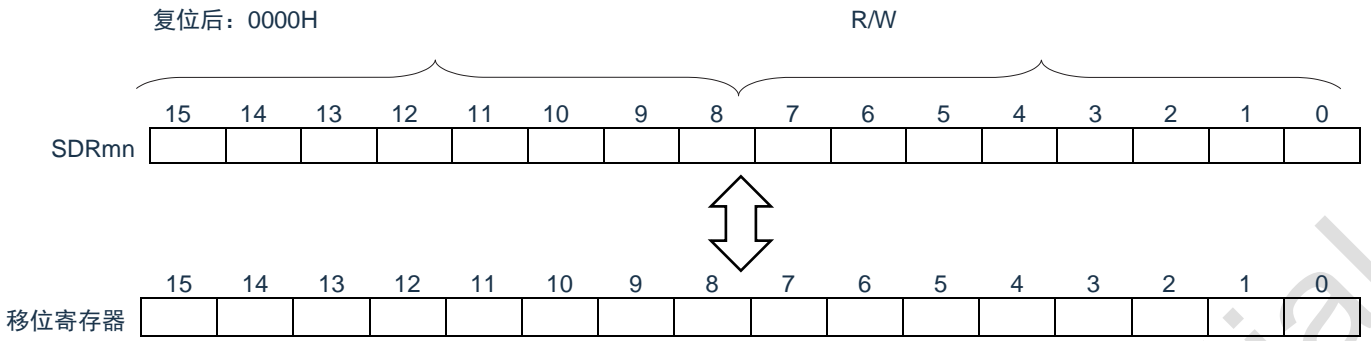
注：在运行停止（SEmn=0）时，禁止通过8位存储器操作指令改写SDRmn[7:0]（否则，SDRmn[15:9]全部被清“0”）。

备注：m：单元号（m=0、1） n：通道号（n=0、1） p：SSPI号（p=00、20）

q：UART号（q=0、2） r：IIC号（r=00、20）



图17-2 串行数据寄存器mn (SDRmn) 的格式



备注：有关SDRmn寄存器的高7位的功能，请参照“17.3 控制通用串行通信单元的寄存器”。

m: 单元号 (m=0、1) n: 通道号 (n=0、1)

## 17.3 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)
- 串行通信运行设定寄存器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)
- 从属选择功能启用寄存器m (SSEm)
- 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、1)

## 串行通信单元寄存器列表

单元 0 寄存器基地址：0x40041100

单元 1 寄存器基地址：0x40041300

偏移地址	寄存器名称	R/W	复位值
0x000	SSRm0	R	0000H
0x002	SSRm1	R	0000H
0x004	SIRm0	R/W	0000H
0x006	SIRm1	R/W	0000H
0x008	SMRm0	R/W	0020H
0x00A	SMRm1	R/W	0020H
0x00C	SCRm0	R/W	0087H
0x00E	SCRm1	R/W	0087H
0x010	SEm	R	0000H
0x012	SSm	R/W	0000H
0x014	STm	R/W	0000H
0x016	SPSm	R/W	0000H
0x018	SOm	R/W	0303H
0x01A	SOEm	R/W	0000H
0x020	SOLm	R/W	0000H
0x022	SSEm	R/W	0000H
0x024	SSCm	R/W	0000H
0x040	SDRm0	R/W	0000H
0x040	SI0m0	R/W	00H
0x040	TXDm	R/W	00H
0x042	SDRm1	R/W	0000H
0x042	RXDm	R/W	00H
0x042	SI0m1	R/W	00H

备注：单元号 m=0, 1

### 17.3.1 外围允许寄存器0 (PER0)

PER0寄存器是设定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定PER0寄存器。

在产生复位信号后，PER0寄存器的值变为“00H”。

图17-3 外围允许寄存器0 (PER0) 的格式

地址: 40020420H	复位后: 00000000H R/W							
符号	7	6	5	4	3	2	1	0
PER0	0	IRDAEN	ADCEN	IICAEN	SCI1EN	SCI0EN	-	TM40EN

SCI <sub>m</sub> 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：要设定通用串行通信单元m时，必须先使SCI<sub>m</sub>EN位为“1”的状态下设定以下的寄存器。当SCI<sub>m</sub>EN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)
- 串行通信运行设定寄存器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)

## 17.3.2 串行时钟选择寄存器m (SPSm)

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定SPSm寄存器。

能用SPSmL并且通过8位存储器操作指令设定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17-4 串行时钟选择寄存器m (SPSm) 的格式

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	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	运行时钟 (CKmk) 的选择 <sup>注</sup>
0	0	0	0	F <sub>CLK</sub>
0	0	0	1	F <sub>CLK</sub> /2
0	0	1	0	F <sub>CLK</sub> /2 <sup>2</sup>
0	0	1	1	F <sub>CLK</sub> /2 <sup>3</sup>
0	1	0	0	F <sub>CLK</sub> /2 <sup>4</sup>
0	1	0	1	F <sub>CLK</sub> /2 <sup>5</sup>
0	1	1	0	F <sub>CLK</sub> /2 <sup>6</sup>
0	1	1	1	F <sub>CLK</sub> /2 <sup>7</sup>
1	0	0	0	F <sub>CLK</sub> /2 <sup>8</sup>
1	0	0	1	F <sub>CLK</sub> /2 <sup>9</sup>
1	0	1	0	F <sub>CLK</sub> /2 <sup>10</sup>
1	0	1	1	F <sub>CLK</sub> /2 <sup>11</sup>
1	1	0	0	F <sub>CLK</sub> /2 <sup>12</sup>
1	1	0	1	F <sub>CLK</sub> /2 <sup>13</sup>
1	1	1	0	F <sub>CLK</sub> /2 <sup>14</sup>
1	1	1	1	F <sub>CLK</sub> /2 <sup>15</sup>

注：要在通用串行通信单元（SCI）运行过程中更改被选择为F<sub>CLK</sub>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<sub>CLK</sub>: CPU/外围硬件的时钟频率
2. m: 单元号 (m=0、1)
3. k=0、1

### 17.3.3 串行模式寄存器mn (SMRmn)

SMRmn寄存器是设定通道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、开始触发的设定、运行模式（SSPI、UART、简易I<sup>2</sup>C）的设定以及中断源的选择。另外，只在UART模式中设定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定SMRmn寄存器。

在产生复位信号后，SMRmn寄存器的值变为“0020H”。

图17-5 串行模式寄存器mn (SMRmn) 的格式(1/2)

	复位后: 0020H													RW		
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 <sub>n</sub> <sup>注1</sup>	0	SISmn <sub>0</sub> <sup>注1</sup>	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	通道n运行时钟（ $F_{MCK}$ ）的选择
0	SPSm寄存器设定的运行时钟CKm0
1	SPSm寄存器设定的运行时钟CKm1
运行时钟（ $F_{MCK}$ ）用于边沿检测电路。通过设定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（ $F_{MCK}$ ）的分频设定。	

STSmn <sup>注1</sup>	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 <sup>2</sup> 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、SMR11寄存器。

注意：必须将bit13~9、7、4、3（SMR00、SMR10寄存器的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、1） p：SSPI号（p=00、20）

q：UART号（q=0、2） r：IIC号（r=00、20）

图17-7 串行模式寄存器mn (SMRmn) 的格式(2/2)

		R/W															
复位后: 0020H		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
符号	SMRmn	CKS mn	CCS mn	0	0	0	0	0	STSm <sub>n</sub> <sup>注1</sup>	0	SISmn <sub>0</sub> <sup>注1</sup>	1	0	0	MD mn2	MD mn1	MD mn0

SISmn0 <sup>注1</sup>	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运行模式的设定
0	0	SSPI模式
0	1	UART模式
1	0	简易I <sup>2</sup> C模式
1	1	禁止设定。

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、SMR11寄存器。

注意：必须将bit13~9、7、4、3 (SMR00、SMR10寄存器的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、1) p：SSPI号 (p=00、20)

q：UART号 (q=0、2) r：IIC号 (r=00、20)

### 17.3.4 串行通信运行设定寄存器mn (SCRmn)

SCRmn寄存器是通道n的通信运行设定寄存器，设定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定SCRmn寄存器。

在产生复位信号后，SCRmn寄存器的值变为“0087H”。

图17-6 串行通信运行设定寄存器mn (SCRmn) 的格式(1/2)

复位后: 0087H

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 <sup>注1</sup>	SLC mn0	DLS mn3	DLS mn2	DLSm n1	DLS mn0

TXEmn	RXEmn	通道n运行模式的设定
0	0	禁止通信。
0	1	只进行接收。
1	0	只进行发送。
1	1	使能发送和接收。

DAPmn	CKPmn	SSPI 模式中的数据与时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4

在 UART 模式和简易 I<sup>2</sup>C 模式中，必须将 DAPmn 位和 CKPmn 位都置“0”。

EOCmn	错误中断信号(INTSREx(x=0、2))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I<sup>2</sup>C模式中或者在UART发送时，必须将EOCmn位置“0”<sup>注2</sup>。

注1：只限于SCR00、SCR10寄存器。

注2：在EOCmn位为“0”并且不使用SSPImn时，有可能产生错误中断INTSREx。

注意：必须将bit6、11置“0”（也必须将SCR01、SCR11寄存器的bit5置“0”）。

备注：m：单元号 (m=0、1) n：通道号 (n=0、1) p：SSPI号 (p=00、20)



图17-8 串行通信运行设定寄存器mn (SCRmn) 的格式(2/2)

	复位后: 0087H															R/W
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 <sup>注1</sup>	SLC mn0	DLS mn3	DLS mn2	DLSm n1	DLS mn0

PTCmn1	PTCmn0	UART模式中的奇偶校验位的设定	
		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 <sup>注2</sup> 。	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I<sup>2</sup>C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I<sup>2</sup>C模式中，必须将DIRmn位置“0”。

SLCmn1 <sup>注1</sup>	SLCmn0	UART模式中的停止位的设定
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、10、20）
1	1	禁止设定。

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I<sup>2</sup>C模式中，必须设定为1个停止位（SLCmn1、SLCmn0=0、1）。  
 在SSPI模式中，必须设定为无停止位（SLCmn1、SLCmn0=0、0）。  
 在UART发送时，必须设定为1位（SLCmn1、SLCmn0=0、1）或者2位（SLCmn1、SLCmn0=1、0）。

DLS mn3	DLS mn2	DLS mn1	DLS mn0	SSPI模式中的数据长度的设定	串行功能对应		
					SSPI	UART	IIC
0	1	1	0	7位数据长度（保存在SDRmn寄存器的bit0~6）	○	○	×
0	1	1	1	8位数据长度（保存在SDRmn寄存器的bit0~7）	○	○	○
1	0	0	0	9位数据长度（保存在SDRmn寄存器的bit0~8）	○	○	×
1	0	0	1	10位数据长度（保存在SDRmn寄存器的bit0~9）	○	×	×
1	0	1	0	11位数据长度（保存在SDRmn寄存器的bit0~10）	○	×	×
1	0	1	1	12位数据长度（保存在SDRmn寄存器的bit0~11）	○	×	×
1	1	0	0	13位数据长度（保存在SDRmn寄存器的bit0~12）	○	×	×
1	1	0	1	14位数据长度（保存在SDRmn寄存器的bit0~13）	○	×	×
1	1	1	0	15位数据长度（保存在SDRmn寄存器的bit0~14）	○	×	×
1	1	1	1	16位数据长度（保存在SDRmn寄存器的bit0~15）	○	○	×
其他				禁止设置。			

在简易I<sup>2</sup>C模式中，必须设置DLSmn3~DLSmn0=0111B。

注1：只限于SCR00、SCR10寄存器。

注2: 与数据的内容无关, 总是附加“0”。

注意: 必须将bit6、11置“0”。

备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1) p: SSPI号 (p=00、20)

Draft Version  
Cmsemicon Confidential

## 17.3.5 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）。

当运行停止（SEmn=0）时，bit15~9用作运行时钟（F<sub>MCK</sub>）的分频设定寄存器。在运行期间（SEmn=1）bit15~9用作发送接收缓冲寄存器。

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）设定的运行时钟的分频时钟就用作传送时钟。

通过16位存储器操作指令设定SIRmn寄存器。

在产生复位信号后，SDRmn寄存器的值变为“0000H”。

图17-7 串行数据寄存器mn (SDRmn) 的格式



注意：

1. 运行停止（SEmn=0）时，必须将bit8~0清零。
2. 在使用UART时，禁止将SDRmn[15:9]设定为“0000000B”和“0000001B”。
3. 在使用简易I2C时，禁止将SDRmn[15:9]设定为“0000000B”，SDRmn[15:9]的设定值必须大于等于“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. SDRmn寄存器在运行期间的功能，请参照“17.2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、1）

### 17.3.6 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SIRmn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SIRmn寄存器。

通过16位存储器操作指令设定SIRmn寄存器。

能用SIRmnL并且通过8位存储器操作指令设定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17-8 串行标志清除触发寄存器mn (SIRmn) 的格式

符号	复位后: 0000H															R/W			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	FECTmn <sup>注1</sup>	PECTmn	OVC Tmn
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn <sup>注1</sup>	PECTmn	OVC Tmn

FECTmn <sup>注1</sup>	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、SIR11寄存器。

注意: 必须将bit15~3 (SIR00、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、1)
2. SIRmn寄存器的读取值总是“0000H”。

## 17.3.7 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。指示的错误为帧错误、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17-9 串行状态寄存器mn (SSRmn) 的格式(1/2)

	复位后: 0000H								R							
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 <sup>注1</sup>	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]	
<ul style="list-style-type: none"> <li>•当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信待机状态）时</li> <li>•当通信结束时</li> </ul>	
[置位条件]	
<ul style="list-style-type: none"> <li>•当开始通信时</li> </ul>	

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]	
<ul style="list-style-type: none"> <li>•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</li> <li>•在接收过程中从SDRmn寄存器读完接收数据时</li> <li>•当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信允许状态）时</li> </ul>	
[置位条件]	
<ul style="list-style-type: none"> <li>•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</li> <li>•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</li> <li>•当发生接收错误时</li> </ul>	

注1: 只限于SSR01、SSR11寄存器。

注意: 如果在BFFmn位为“1”时写SDRmn寄存器, 就会丢弃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、1)

图17-11 串行状态寄存器mn (SSRmn) 的格式(2/2)

复位后: 0000H														R		
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 <sup>注1</sup>	PEF mn	OVF mn

FEFmn <sup>注1</sup>	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 <sup>2</sup> 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 <sup>2</sup> 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、SSR11寄存器。

备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1)

### 17.3.8 串行通道开始寄存器m (SSm)

SSm寄存器是设定允许各通道的通信/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定SSm寄存器。

能用SSmL并且通过8位存储器操作指令设定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17-10 串行通道开始寄存器m (SSm) 的格式

复位后: 0000H	R/W															
符号	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	0	0	SSm1	SSm0

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：必须将SSm寄存器的bit15~2置“0”。

备注：

1. m: 单元号 (m=0、1) n: 通道号 (n=0、1)
2. SSm寄存器的读取值总是“0000H”。

### 17.3.9 串行通道停止寄存器m (STm)

STm寄存器是设定允许各通道的通信/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定STm寄存器。

能用STmL并且通过8位存储器操作指令设定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17-11 串行通道停止寄存器m (STm) 的格式

复位后: 0000H		R/W															
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

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STm寄存器的bit15~2置“0”。

备注：

1. m：单元号 (m=0、1) n：通道号 (n=0、1)
2. STm寄存器的读取值总是“0000H”。



### 17.3.10 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，能通过软件设定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17-12 串行通道允许状态寄存器m (SEm) 的格式

复位后: 0000H																R		
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1	SEm0		

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、1)

### 17.3.11 串行输出允许寄存器m (SOEm)

SOEm寄存器设定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，能通过软件设定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定SOEm寄存器。

能用SOEmL并且通过8位存储器操作指令设定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17-13 串行输出允许寄存器m (SOEm) 的格式

复位后: 0000H		R/W															
符号	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	0	0	SOE <sub>m1</sub>	SOE <sub>m0</sub>	

SOE <sub>mn</sub>	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：SOEm寄存器的bit15~2置“0”。

备注：m：单元号 (m=0、1) n：通道号 (n=0、1)

## 17.3.12 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定SOm寄存器。

在产生复位信号后, SOm寄存器的值变为“0303H”。

图17-14 串行输出寄存器m (SOm) 的格式

复位后: 0303H		R/W															
符号		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm		0	0	0	0	0	0	CKO m1	CKO m0	0	0	0	0	0	0	SO m1	SO m0

CKO mn	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”。															

SO mn	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SOm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、1)

### 17.3.13 串行输出电平寄存器m (SOLm)

SOLm寄存器是设定各通道的数据输出电平反相的寄存器。

只有在UART模式中才能设定此寄存器。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反相设定反映到引脚输出。在禁止串行输出时 (SOEmn=0)，将SOLmn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定SOLm寄存器。

能用SOLmL并且通过8位存储器操作指令设定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17-15 串行输出电平寄存器m (SOLm) 的格式

复位后: 0000H	R/W															
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm0

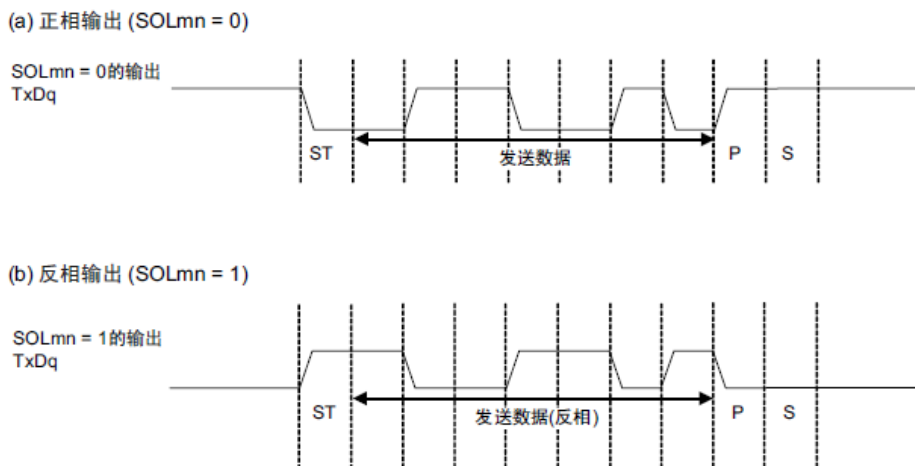
SOLmn	UART模式中的通道n发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

注意：必须将SOL0、SOL1寄存器的bit15~1置“0”。

备注：m：单元号 (m=0、1) n：通道号 (n=0)

当进行UART发送时，发送数据的电平反相例子如图17-16所示。

图17-16 发送数据的电平反相例子



备注：m：单元号 (m=0、1) n：通道号 (n=0)

### 17.3.14 串行待机控制寄存器m (SSCm)

SSC0寄存器是控制在SSPI00或者UART0的串行数据接收时从深度睡眠模式启动接收运行的寄存器。

SSC2寄存器是控制在SSPI20或者UART2的串行数据接收时从深度睡眠模式启动接收运行的寄存器。

通过16位存储器操作指令设定SSCm寄存器。

能用SSCmL并且通过8位存储器操作指令设定SSCm寄存器的低 8 位。

在产生复位信号后，SSCm寄存器的值变为“0000H”。

图 17-17 串行输出电平寄存器m (SSCm) 的格式

复位后: 0000H		R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSE Cm	SWC m

SSE Cm	深度睡眠唤醒模式中的通信错误中断产生的允许或者停止的选择
0	允许产生错误中断。
1	禁止产生错误中断。
<ul style="list-style-type: none"> <li>在 UART 低功耗模式中进行 UART 接收的情况下，只有在 SWCm 位和 EOCm 位为“1”时才能将 SSECm 位置“1”或者“0”，否则必须将 SSECm 位置“0”。</li> <li>禁止将 SSECm 位和 SWCm 位分别置“1”和“0”。</li> </ul>	

SWC m	低功耗 UART 模式的设定
0	不使用深度睡眠唤醒模式功能。
1	用深度睡眠唤醒模式功能。
<ul style="list-style-type: none"> <li>在深度睡眠模式中，通过硬件触发信号解除深度睡眠模式，并且在 CPU 不运行的状态下进行 UART 的接收（低功耗 UART 模式）。</li> <li>只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟 (fCLK) 时才能设定低功耗 UART 模式功能，而在选择其他时钟的情况下禁止设定。</li> <li>即使使用低功耗 UART 模式，也必须在通常运行模式中将 SWCm 位置“0”并且在即将要转移到深度睡眠模式前将 SWCm 位改为“1”。</li> </ul> 另外，必须在从深度睡眠模式返回到通常运行模式后将 SWCm 位改为“0”。	

### 17.3.15 从属选择功能启用寄存器m (SSEm)

SSEm寄存器是控制在SSPI<sub>m</sub>n通信中作为从属功能时SS<sub>m</sub>n端子的输入是否有效。

通过16位存储器操作指令设定SSEm寄存器。

能用SSEmL并且通过8位存储器操作指令设定SSEm寄存器的低8位。

在产生复位信号后，SSEm寄存器的值变为“0000H”。

图17-18 从属选择功能启用寄存器m (SSEm) 的格式

复位后: 0000H		R/W															
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIE <sub>m0</sub>

SSIE <sub>m</sub>	SSPI <sub>m</sub> n 通信的从属模式中通道 n 的 SS <sub>m</sub> n 输入的设置
0	SS <sub>m</sub> n引脚输入无效。
1	SS <sub>m</sub> n引脚输入有效

备注: m=0, n=0

## 17.3.16 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。

通过8位存储器操作指令设定ISC寄存器。

在产生复位信号后，ISC寄存器的值变为“00H”。

图17-19 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	0	0	0	0	0	ISC2	ISC1	ISC0

ISC2	外部中断 (INTP11) 的输入切换
0	将INTP11引脚的输入信号用作外部中断的输入。
1	将LRxD0引脚的输入信号用作外部中断的输入。

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bit7~3置“0”。

### 17.3.17 噪声滤波器允许寄存器0 (NFEN0)

NFEN0寄存器设定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I<sup>2</sup>C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定NFEN0寄存器。

在产生复位信号后，NFEN0寄存器的值变为“00H”。

图17-20 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	0	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”。	

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”。



### 17.3.18 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，必须设定复用端口功能的控制寄存器（端口模式寄存器（PMxx）、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“第2章 引脚功能”。

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”。并将端口寄存器Pxx的位置“1”。

另外，当使用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”。

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”。

## 17.4 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。

### 17.4.1 以单元为单位停止运行的情况

通过外围允许寄存器0（PER0）设定以单元为单位的停止运行。

PER0寄存器是设定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17-21 以单元为单位停止运行时的外围允许寄存器0（PER0）的设定

(a)外围允许寄存器0（PER0）……只将要停止SCI<sub>m</sub>的对应位置“0”。

	7	6	5	4	3	2	1	0
PER0	XX	XX	XX	XX	SCI1EN	SCI0EN	XX	XX

SCI<sub>m</sub>输入时钟的控制  
 0: 停止提供输入时钟  
 1: 提供输入时钟

注意：1.当SCI<sub>m</sub>EN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定）。

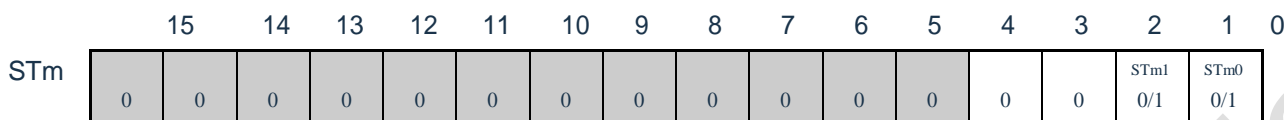
0/1：根据用户的用途置“0”或者“1”。

## 17.4.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

图17-22 按通道停止运行时的各寄存器的设定

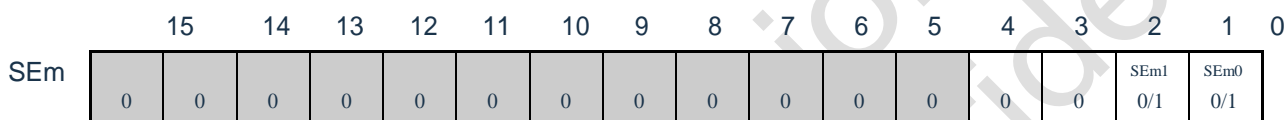
(a) 串行通道停止寄存器m (STm) .....这是设定允许各通道的通信/停止计数的寄存器。



1: 将SEmn位清“0”并且停止通信运行

※因为STmn位是触发位，所以如果SEmn位为“0”就立即清除STmn位。

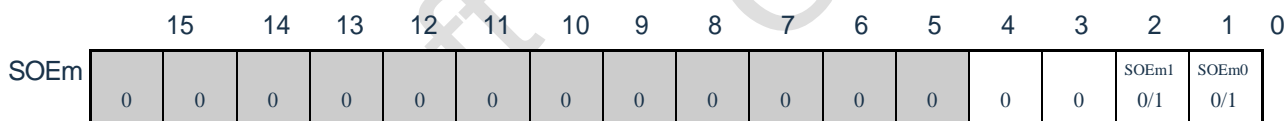
(b) 串行通道允许状态寄存器m (SEm) .....此寄存器表示各通道的数据发送和接收的运行或者停止状态。



0: 运行停止状态

※SEm寄存器是只读状态寄存器，通过STm寄存器来停止运行。对于已经停止运行的通道，能通过软件设定SOM寄存器的CKOmn位的值。

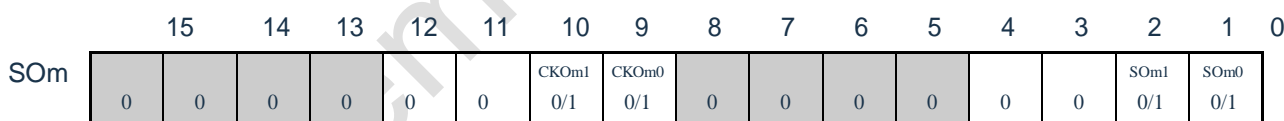
(c) 串行输出允许寄存器m (SOEm) .....这是设定允许或者停止各通道串行通信输出的寄存器。



0: 通过串行通信运行来停止输出

※对于已经停止串行输出的通道，能通过软件设定SOM寄存器的SOMn位的值。

(d) 串行输出寄存器m (SOM) .....这是各通道串行输出的缓冲寄存器。



1: 串行时钟的输出值为“1”

1: 串行数据的输出值为“1”

※当将各通道对应的引脚用作端口功能时，必须将相应的CKOmn位和SOMn位置“1”。

注：只限于通用串行通信单元0。

备注：1.m: 单元号 (m=0、1) n: 通道号 (n=0、1)

2. □ : 不能设定 (设定初始值)。0/1: 根据用户的用途置“0”或者“1”。

## 17.5 3线串行I/O（SSPI00、SSPI20）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~16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
- 设定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：Max.f<sub>CLK</sub>/2

从属通信：Max.f<sub>MCK</sub>/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<sub>KCY</sub>）特性的范围内使用。详细内容请参照数据手册。

SCI0的通道0，SCI1的通道0是支持3线串行I/O（SSPI00、SSPI20）的通道。

3线串行I/O（SSPI00、SSPI20）有以下6种通信运行：

- 主控发送（参照17.5.1）
- 主控接收（参照17.5.2）
- 主控的发送和接收（参照17.5.3）
- 从属发送（参照17.5.4）
- 从属接收（参照17.5.5）
- 从属的发送和接收（参照17.5.6）

## 17.5.1 主控发送

主控发送是指本产品输出传送时钟并且将数据发送到其他设备的运行。

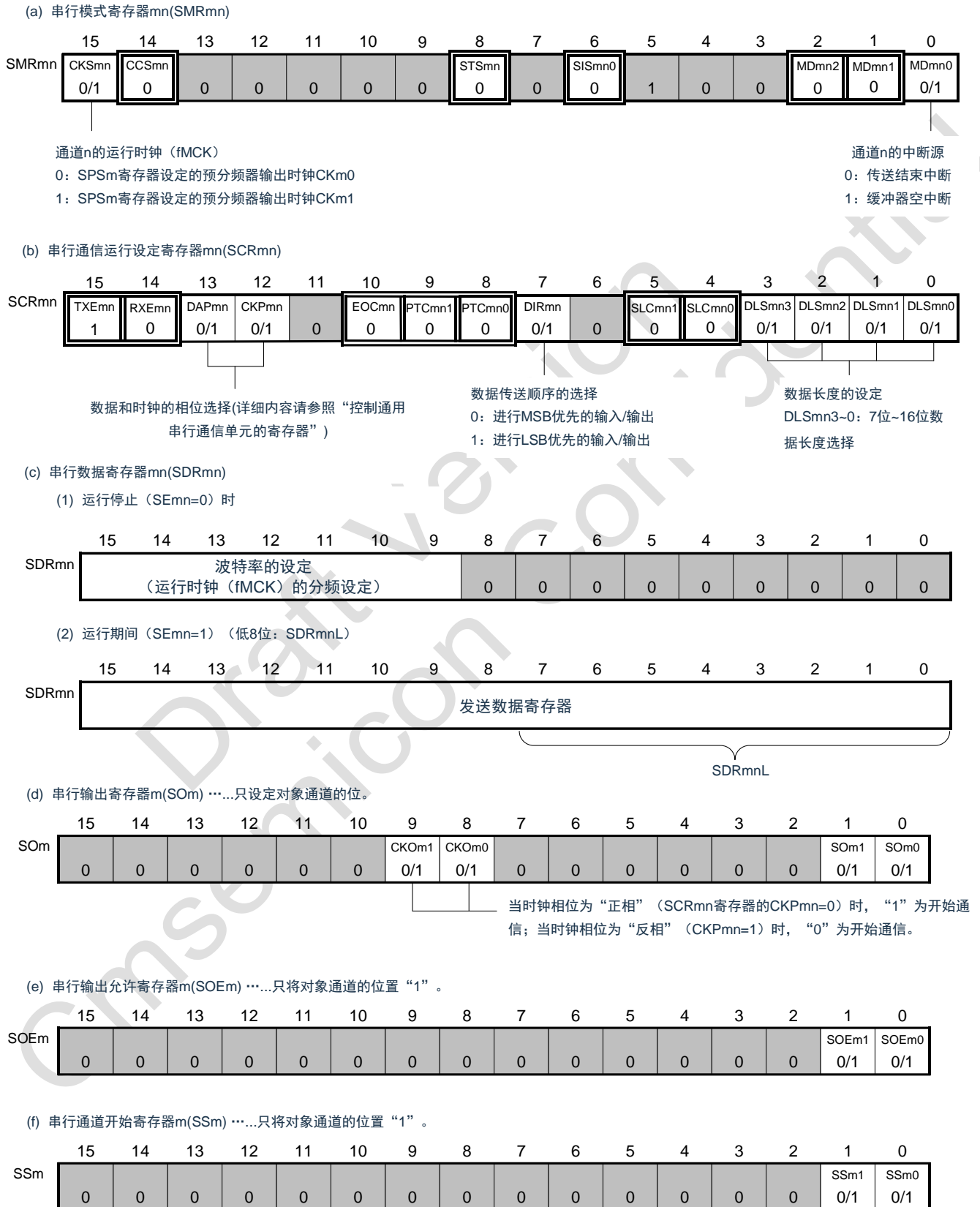
3线串行 I/O	SSPI00	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00	SCLK20
	SDO00	SDO20
中断	INTSSPI00	INTSSPI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。	
错误检测标志	无	
传送数据长度	7 位~16 位	
传送速率注	Max.f <sub>CLK</sub> /2[Hz] Min.f <sub>CLK</sub> /(2x2 <sup>11</sup> x128) [Hz]    f <sub>CLK</sub> : 系统时钟频率	
数据相位	能通过 SCR <sub>mn</sub> 寄存器的 DAP <sub>mn</sub> 位进行选择。 • DAP <sub>mn</sub> =0: 在串行时钟开始运行时, 开始数据输出。 • DAP <sub>mn</sub> =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 <sub>mn</sub> 寄存器的 CKP <sub>mn</sub> 位进行选择。 • CKP <sub>mn</sub> =0: 正相 • CKP <sub>mn</sub> =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) mn=00、10

(1) 寄存器的设定

图17-23 3线串行I/O (SSPI00、SSPI20)  
主控发送时的寄存器设定内容例子



备注: 1.m: 单元号 (m=0、1) n: 通道号 (n=0) mn=00、10  
2. □: 不能设定 (设定初始值)。0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-24 主控发送的初始设置步骤

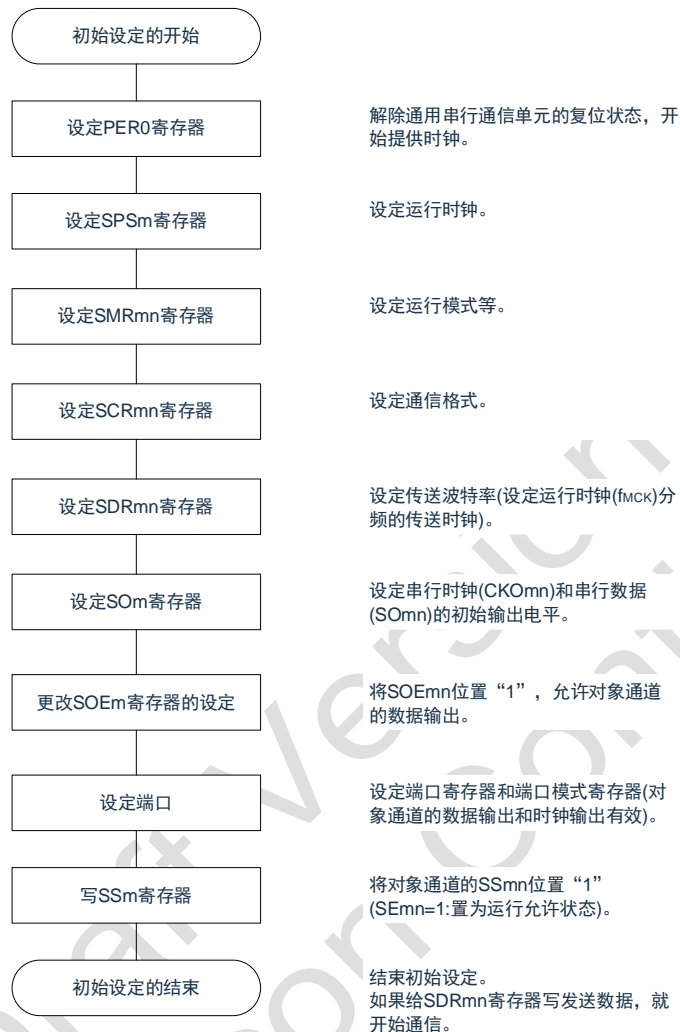


图17-25 主控发送的中止步骤

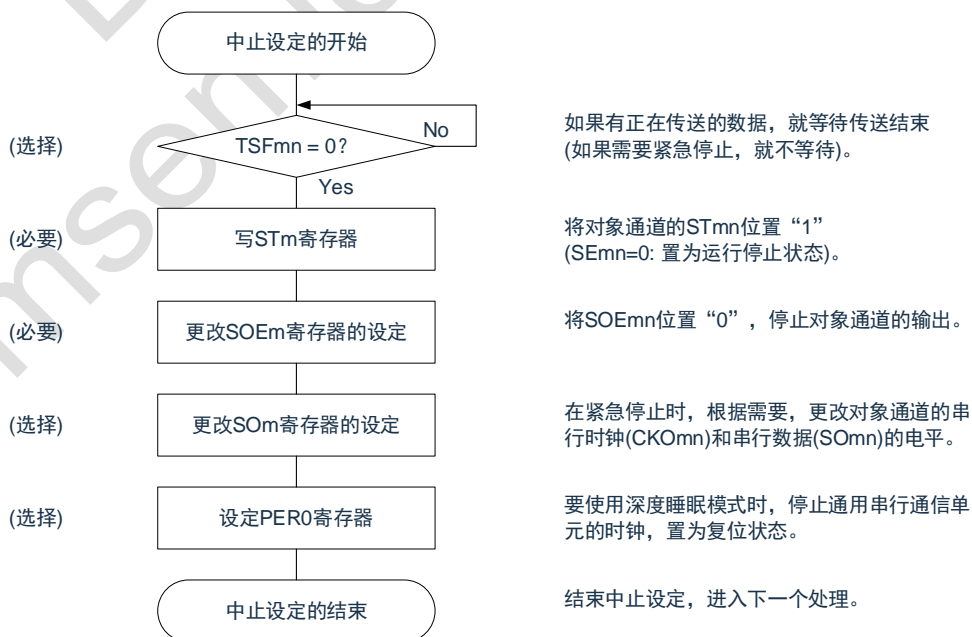
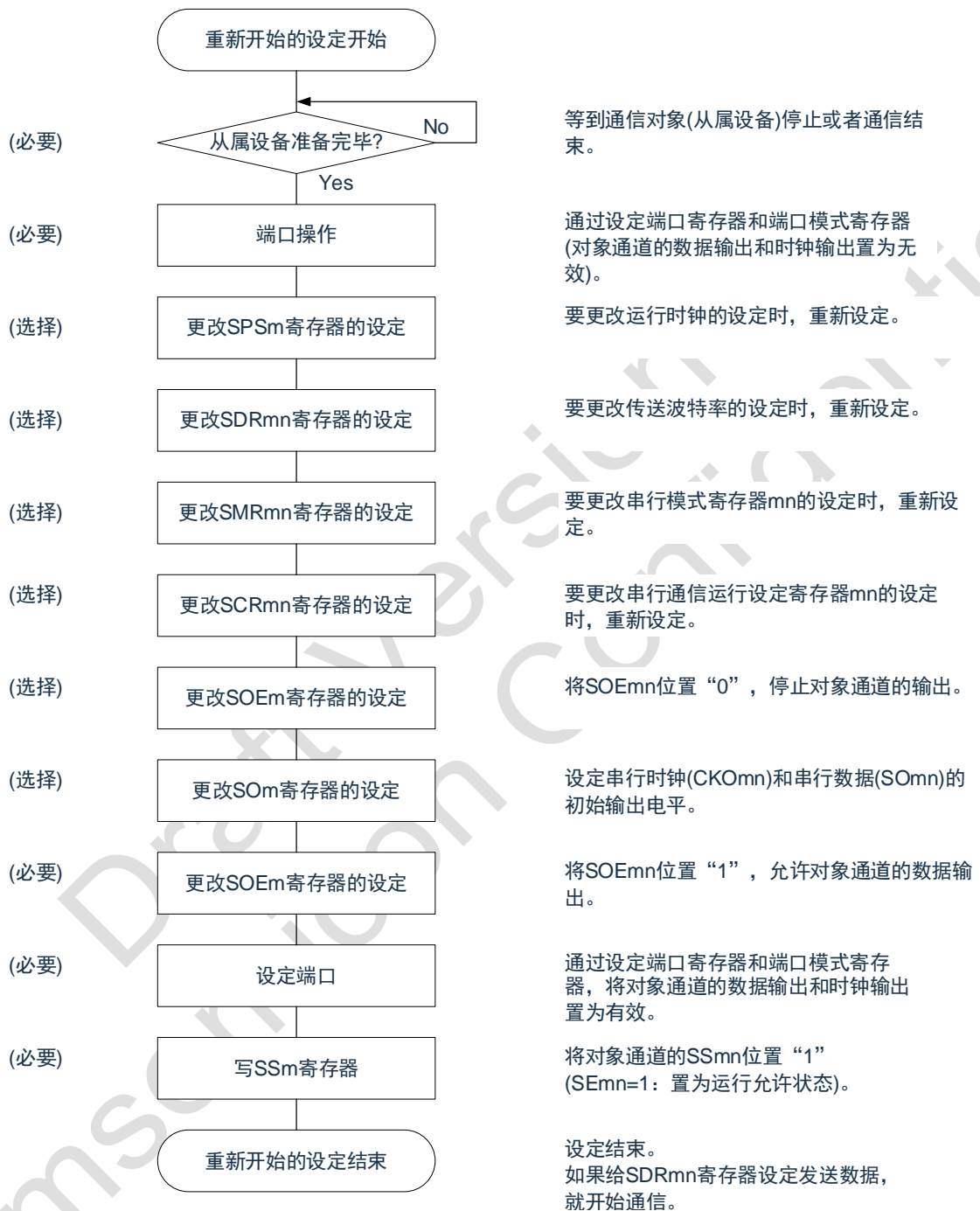


图17-26 重新开始主控发送的设定步骤

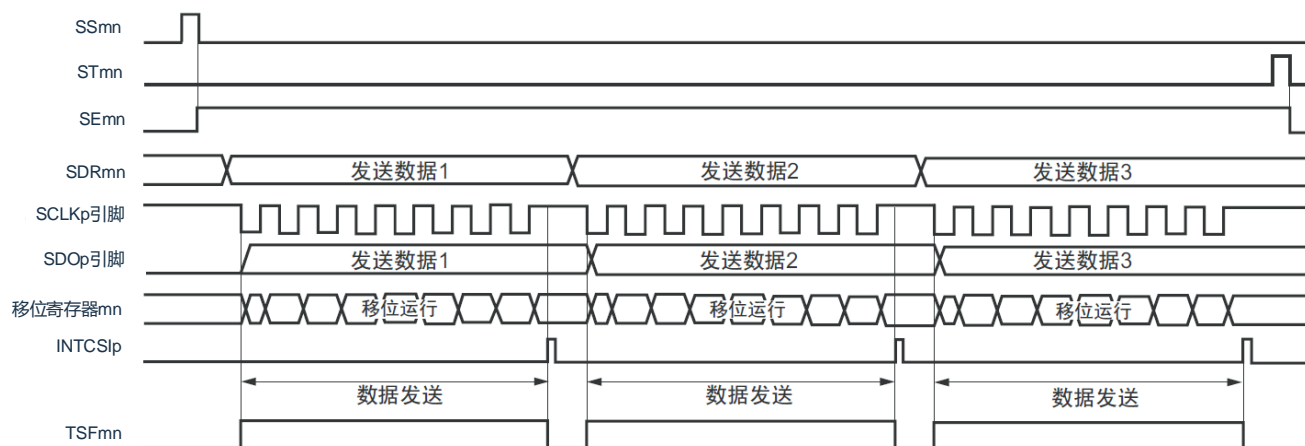


备注: 如果在中止设定中改写PER0来停止提供时钟, 就必须在等到通信对象(从属设备)停止或者通信结束后进行初始设定而不是进行重新开始设定。



(3) 处理流程（单次发送模式）

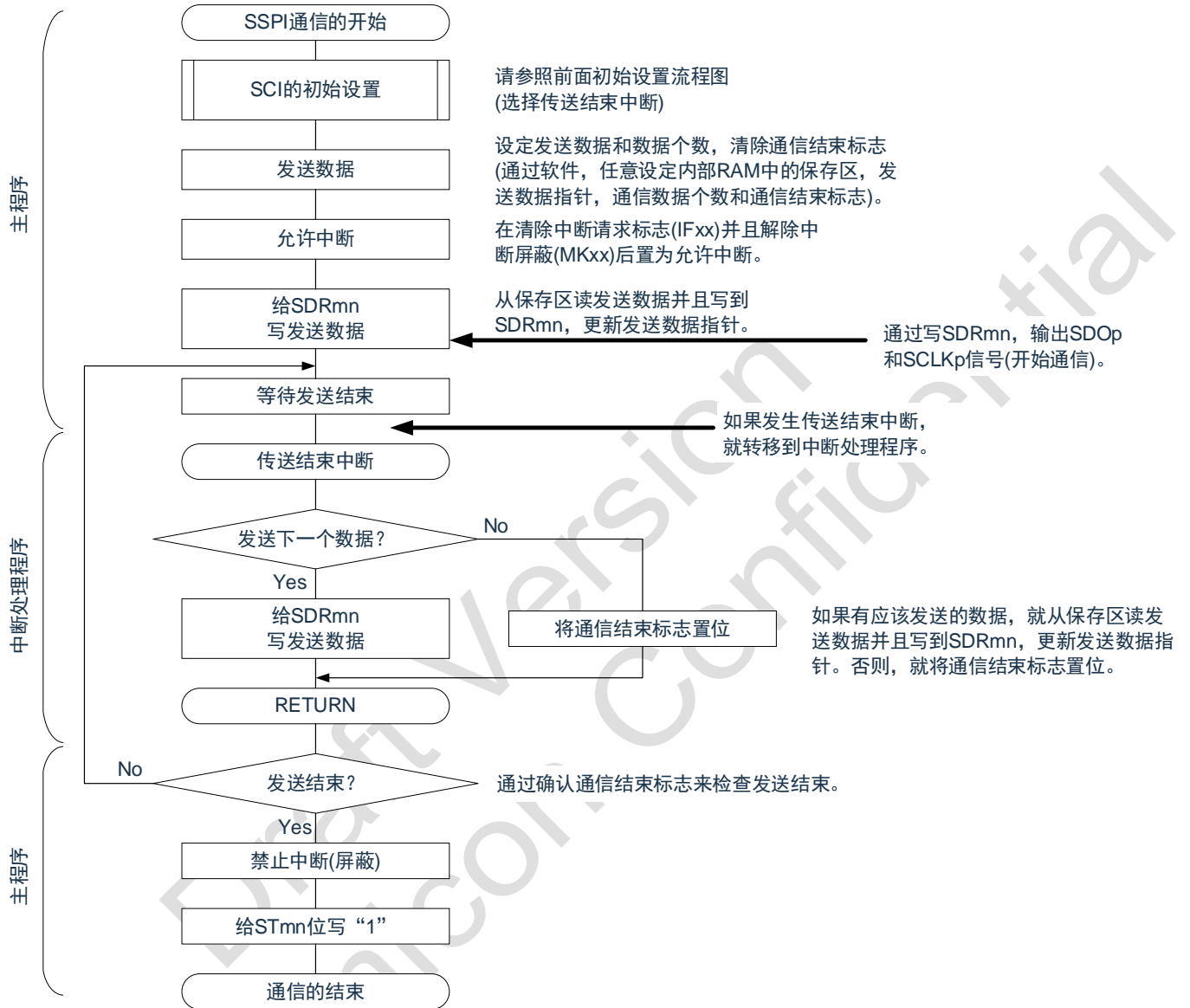
图17-27 主控发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1） n：通道号（n=0） p：SSPI号（p=00、20）

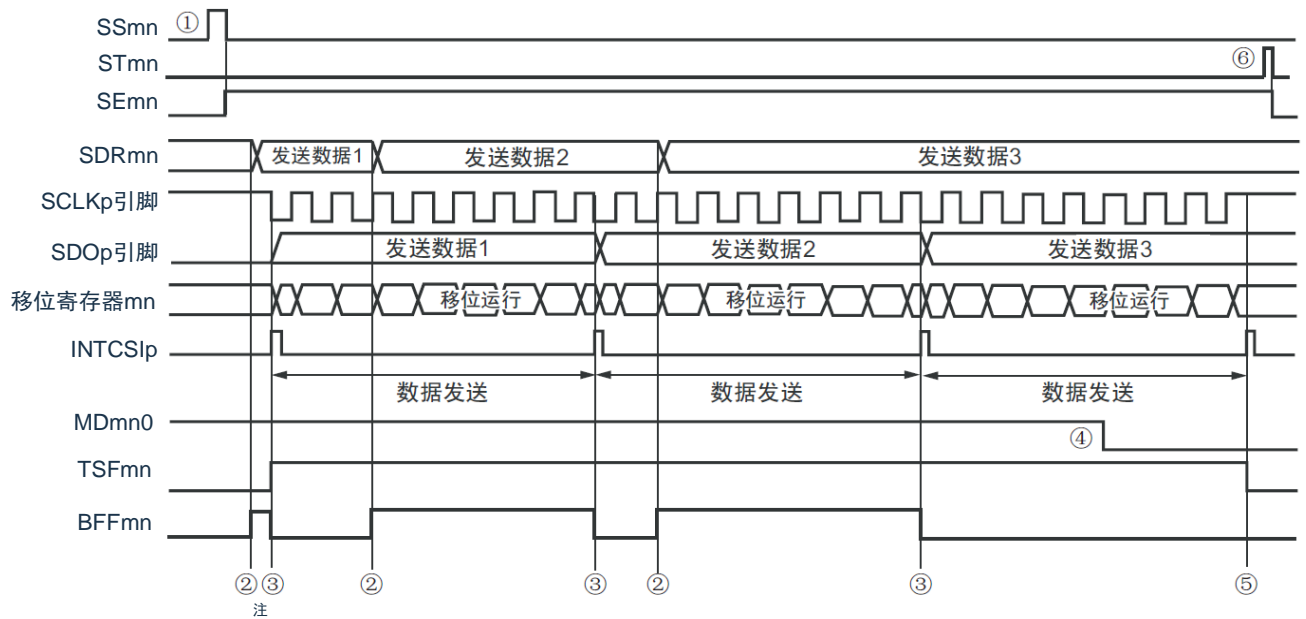
Draft Version  
Cmsemicon Confidential

图17-28 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17-29 主控发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

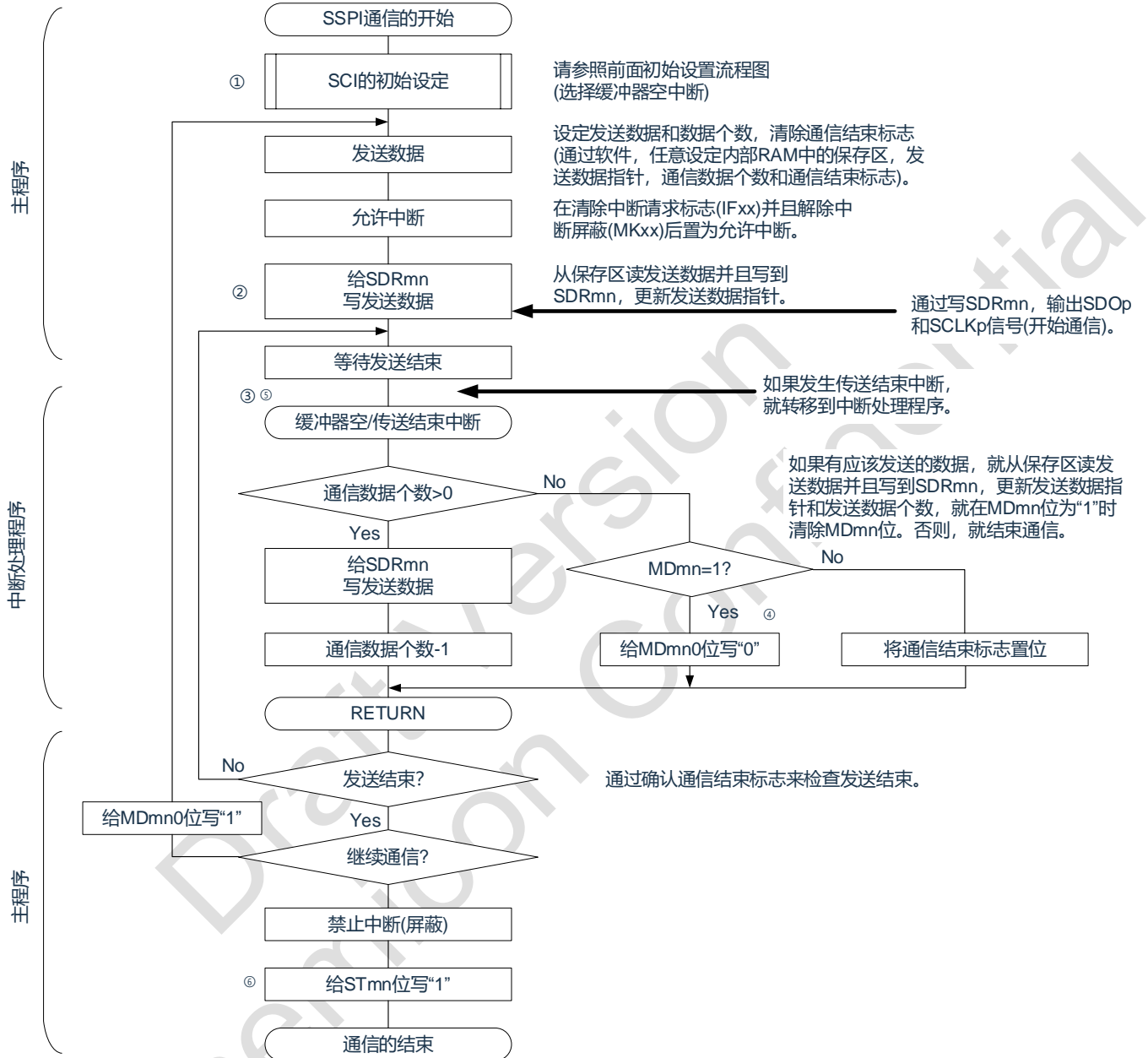


注：如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。

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） p：SSPI号（p=00、20）

图17-30 主控发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图17-29 主控发送（连续发送模式）的时序图”中的①~⑥。

## 17.5.2 主控接收

主控接收是指本产品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	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00	SCLK20
	SDO00	SDO20
中断	INTSSPI00	INTSSPI20
	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~16 位	
传送速率注	Max.f <sub>CLK</sub> /2[Hz]	
	Min.f <sub>CLK</sub> /(2x2 <sup>11</sup> x128) [Hz]     f <sub>CLK</sub> : 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	

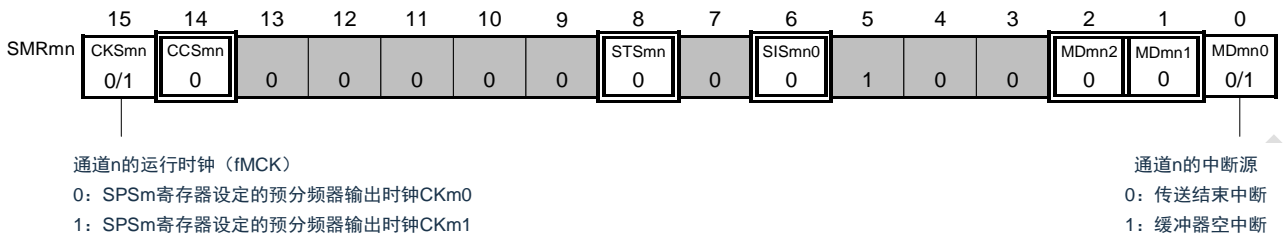
注：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

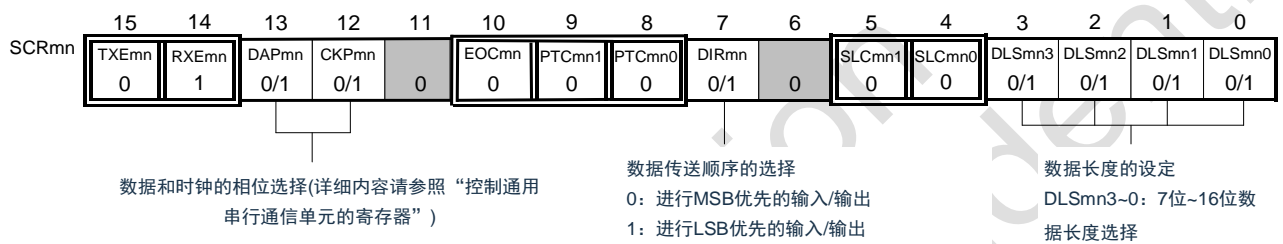
(1) 寄存器的设定

图17-31 3线串行I/O (SSPI00、SSPI20)  
主控接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

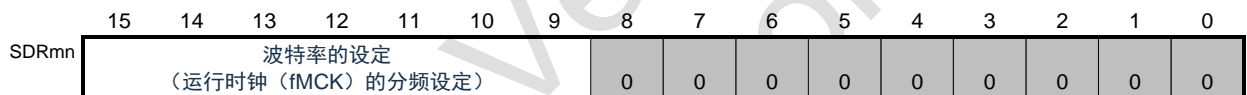


(b) 串行通信运行设定寄存器mn(SCRmn)

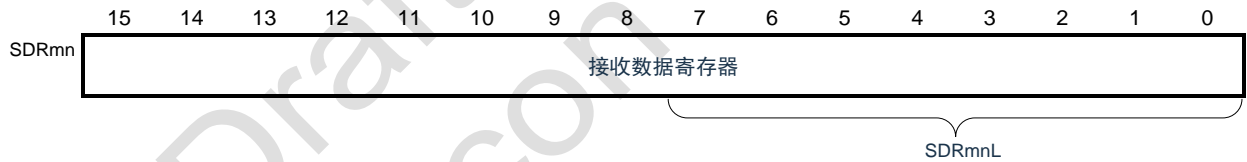


(c) 串行数据寄存器mn(SDRmn)

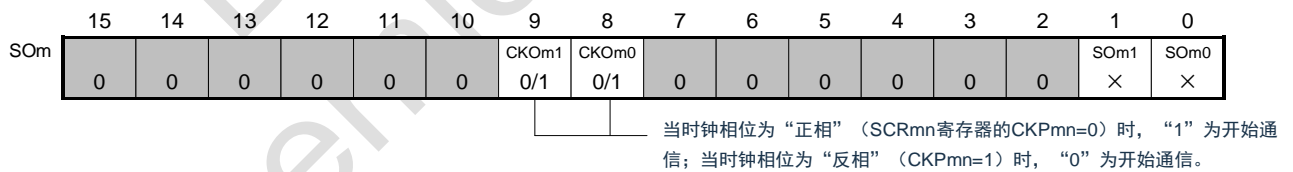
(1) 运行停止 (SEmn=0) 时



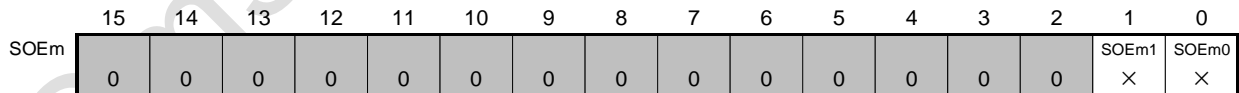
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



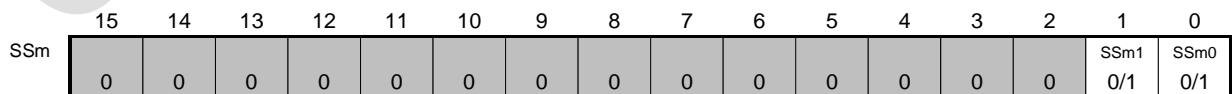
(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(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”。



备注: 1.m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

2.    : 在SSPI主控接收模式中为固定设定    : 不能设定(设定初始值)。  
 ×: 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, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-32 主控接收的初始设定步骤

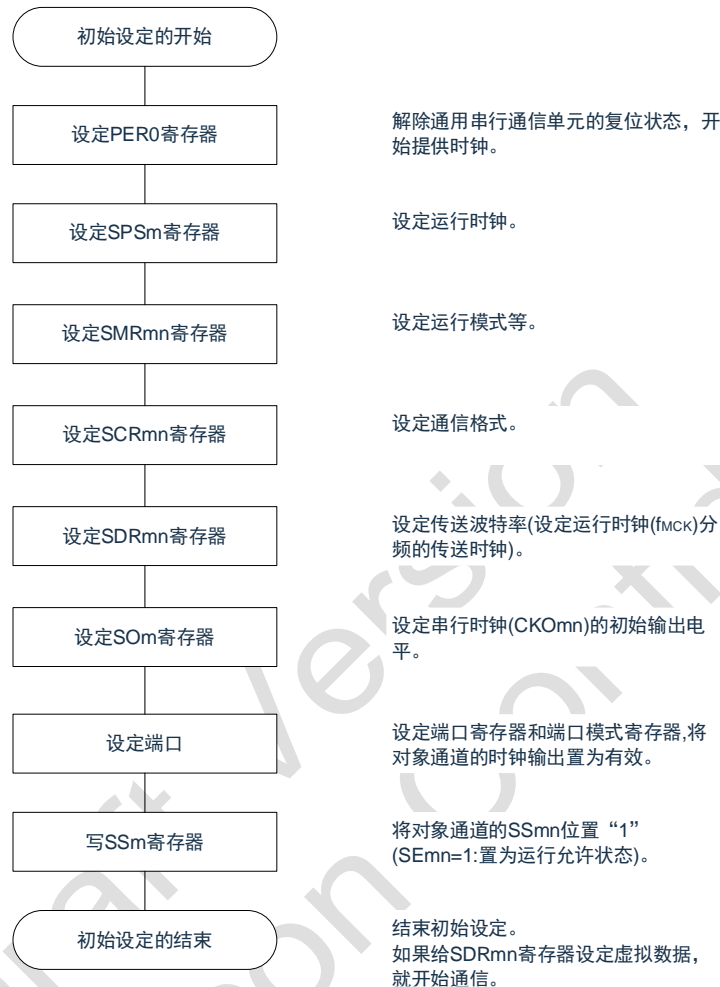


图17-33 主控接收的中止步骤

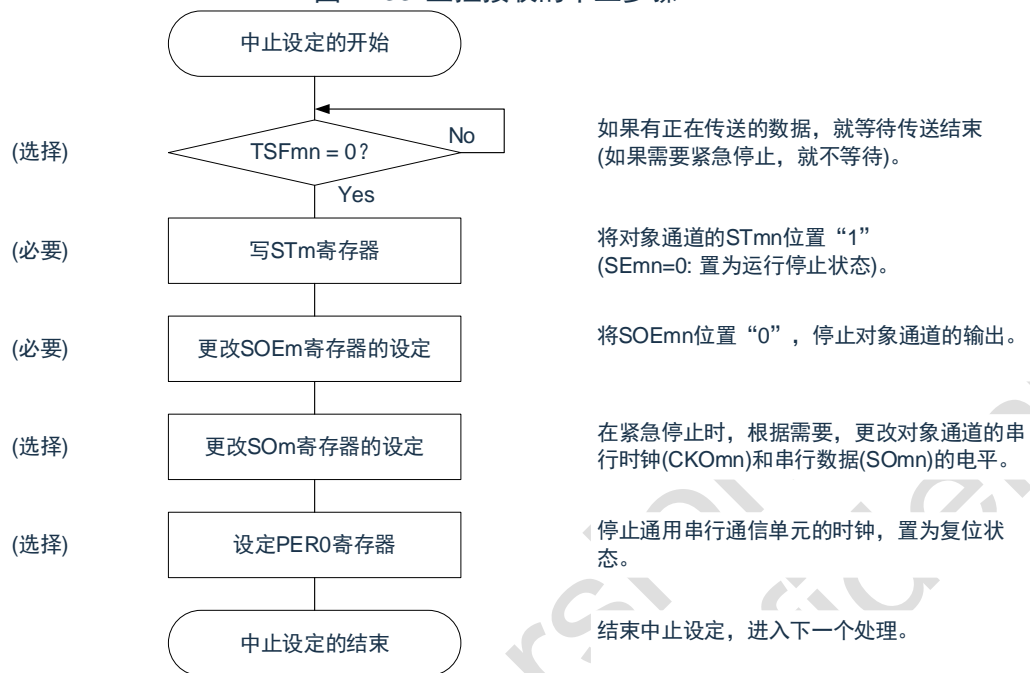
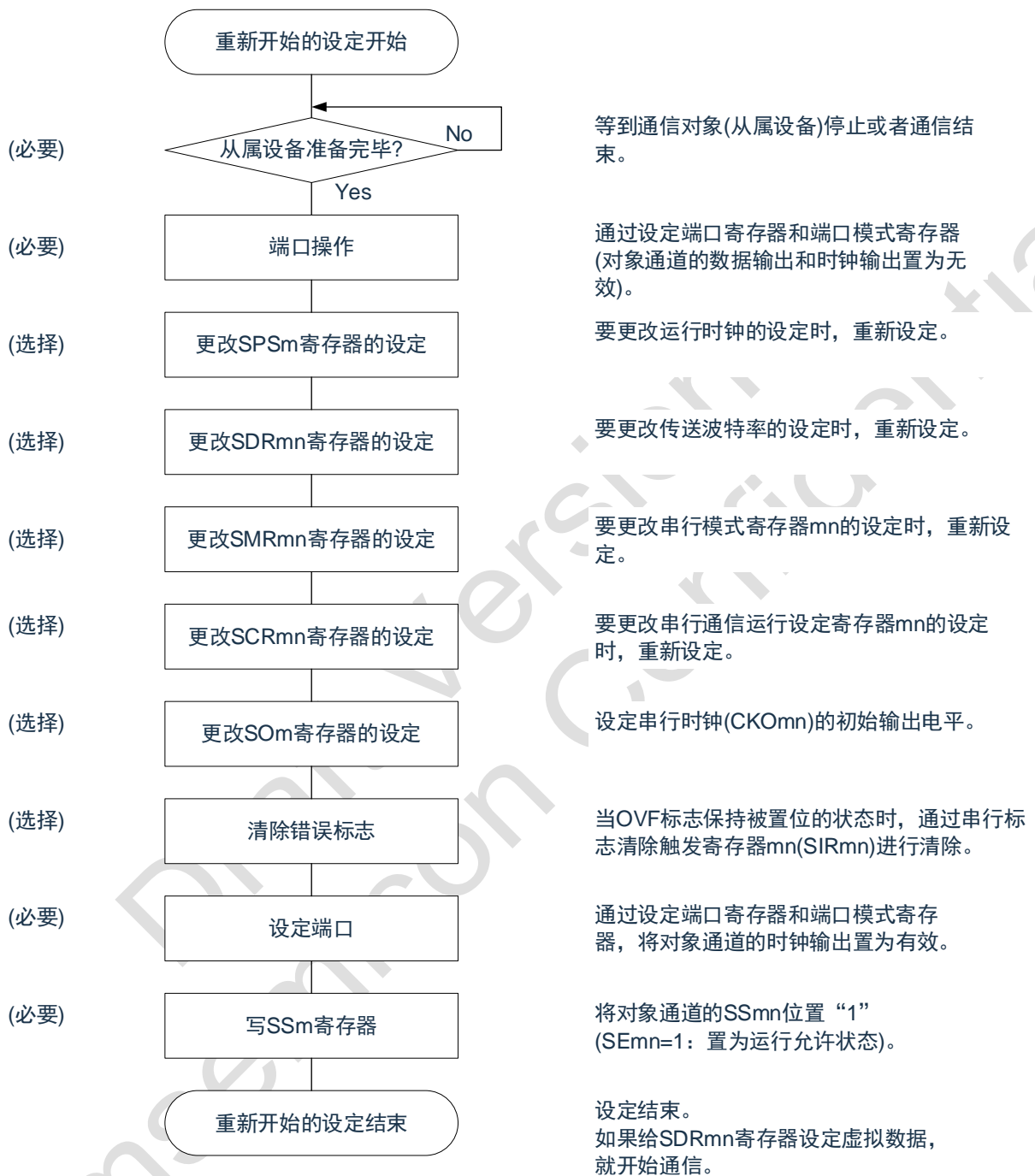




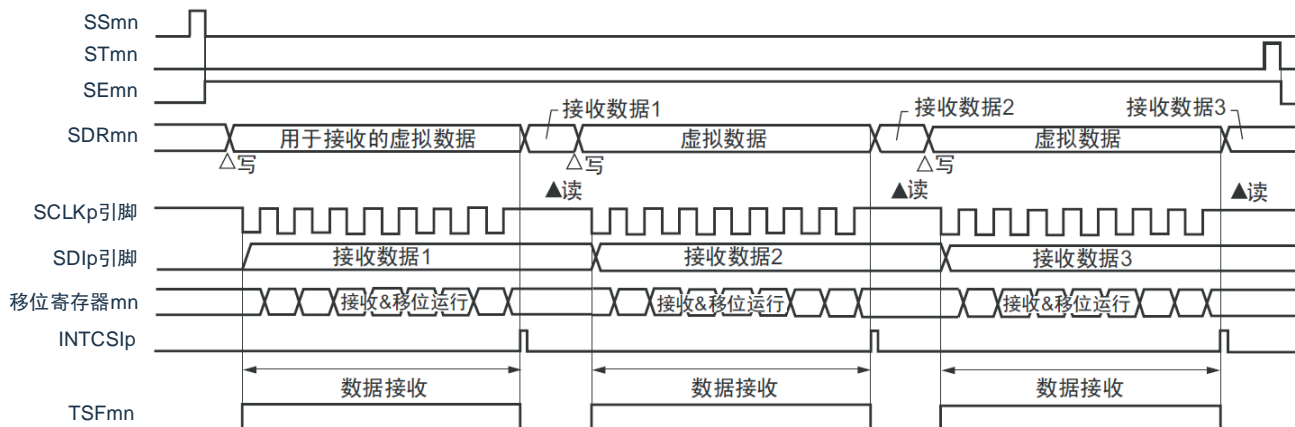
图17-34 重新开始主控接收的设定步骤



备注: 如果在中止设定中改写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定而不是进行重新开始设定。

(3) 处理流程（单次接收模式）

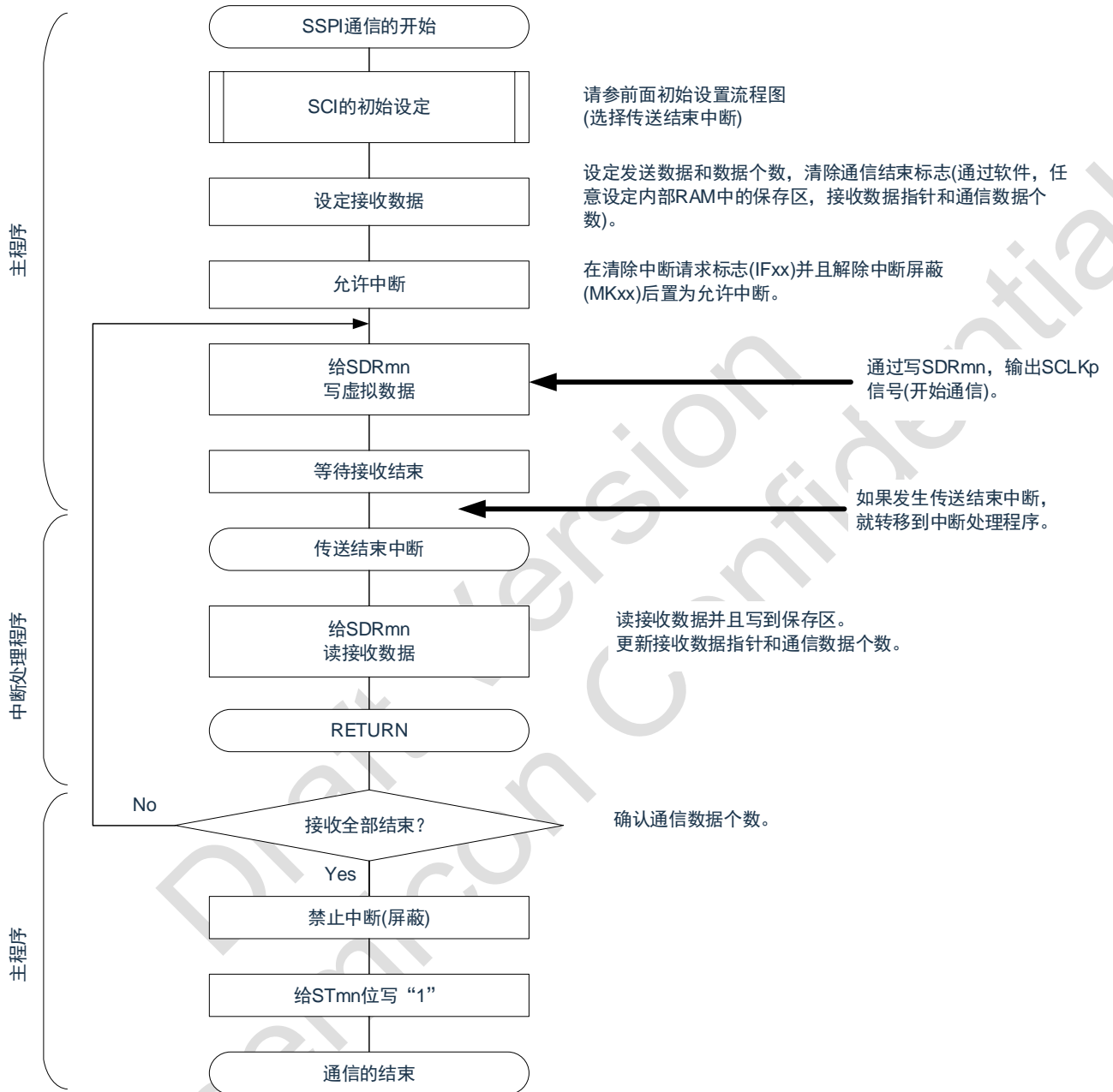
图17-35 主控接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）n：通道号（n=0）p：SSPI号（p=00、20）

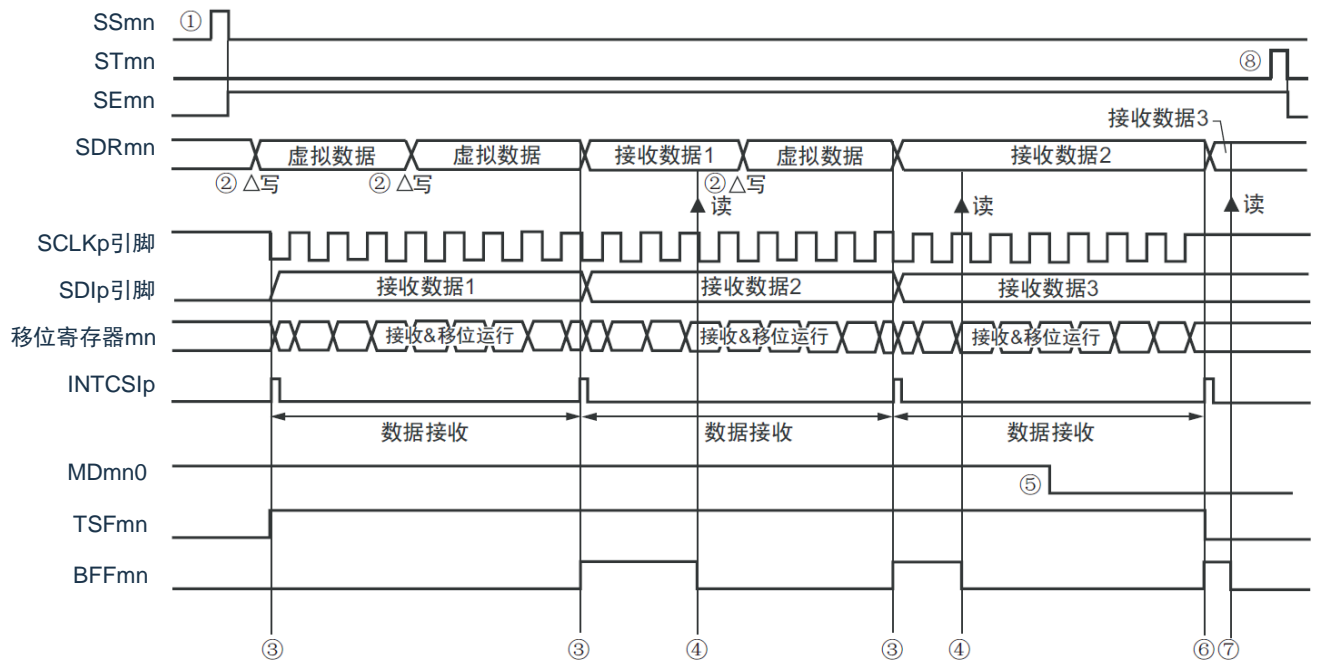
Draft Version  
Cmsemicon Confidential

图17-36 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17-37 主控接收（连续接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

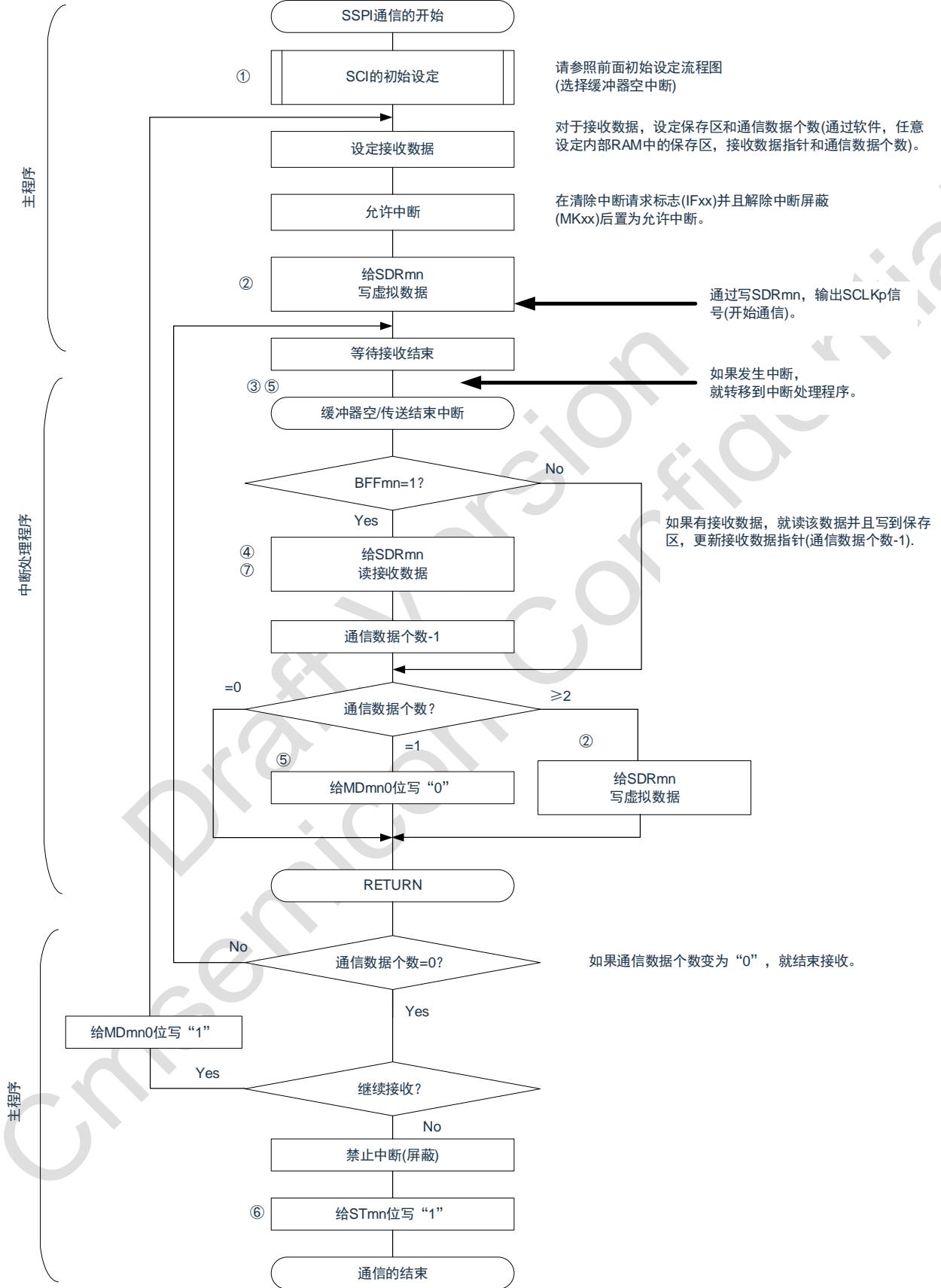


注意：即使在运行过程中也能改写MDmn0位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注：1.图中的①~⑧对应“图17-38 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) p: SSPI号 (p=00, 20)

图17-38 主控接收（连续接收模式）的流程图



备注： 图中的①~⑧对应“图17-37 主控接收（连续接收模式）的时序图”中的①~⑧。

### 17.5.3 主控的发送和接收

主控的发送和接收是指本产品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。

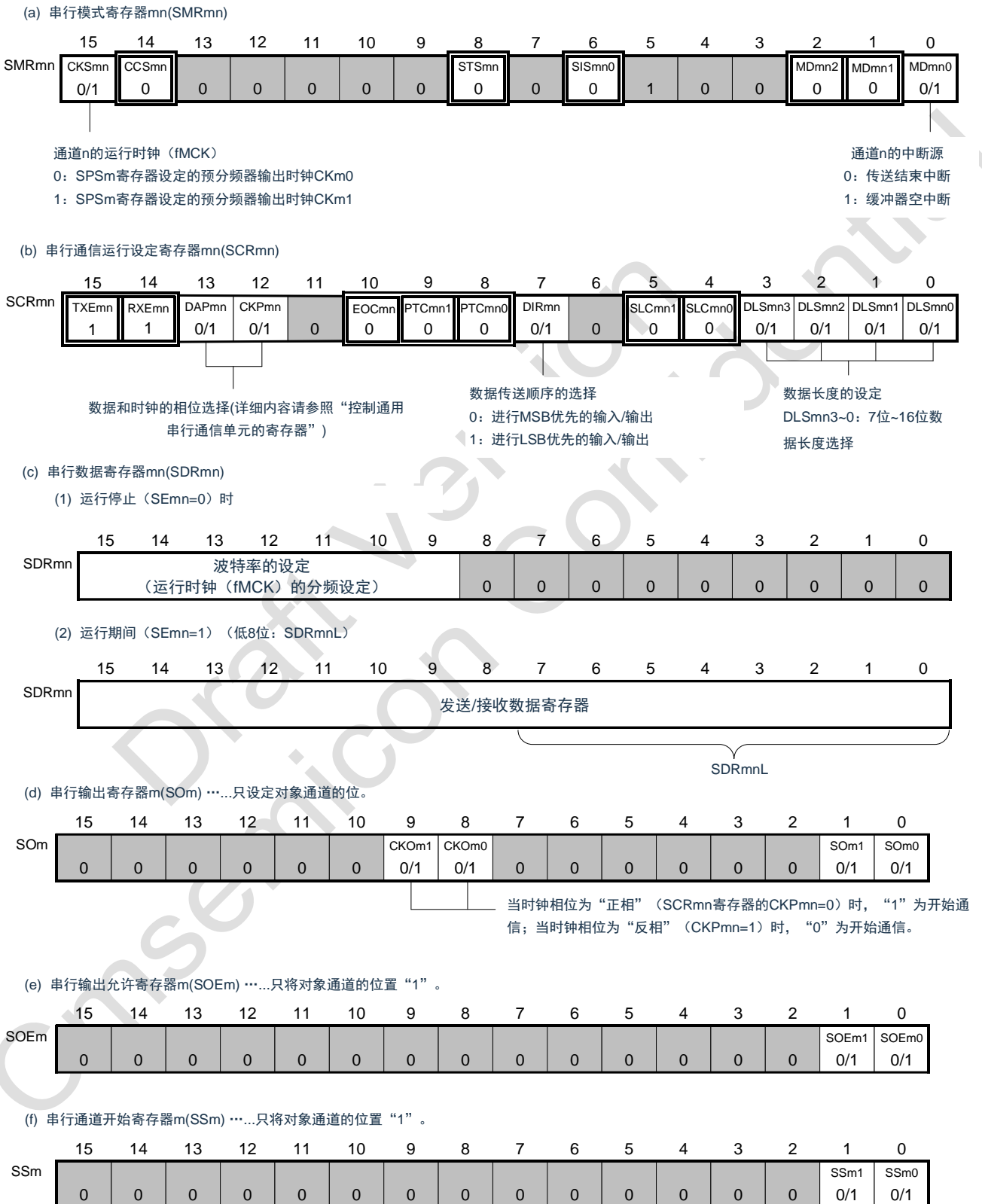
3 线串行 I/O	SSPI00	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00 SDI00 SDO00	SCLK20 SDI20 SDO20
中断	INTSSPI00	INTSSPI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。	
错误检测标志	只有溢出错检测标志（OVFmn）。	
传送数据长度	7 位~16 位	
传送速率注	Max.f <sub>CLK</sub> /2[Hz]	
	Min.f <sub>CLK</sub> /(2x2 <sup>11</sup> x128) [Hz]    f <sub>CLK</sub> : 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） p：SSPI号（p=00、20）

(1) 寄存器的设定

图17-39 3线串行I/O (SSPI00、SSPI20)  
主控发送和接收时的寄存器设定内容例子



备注1.m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

2. □ : 在SSPI主控发送和接收模式中为固定设定。 ■ : 不能设定(设定初始值)。  
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-40 主控发送和接收的初始设定步骤

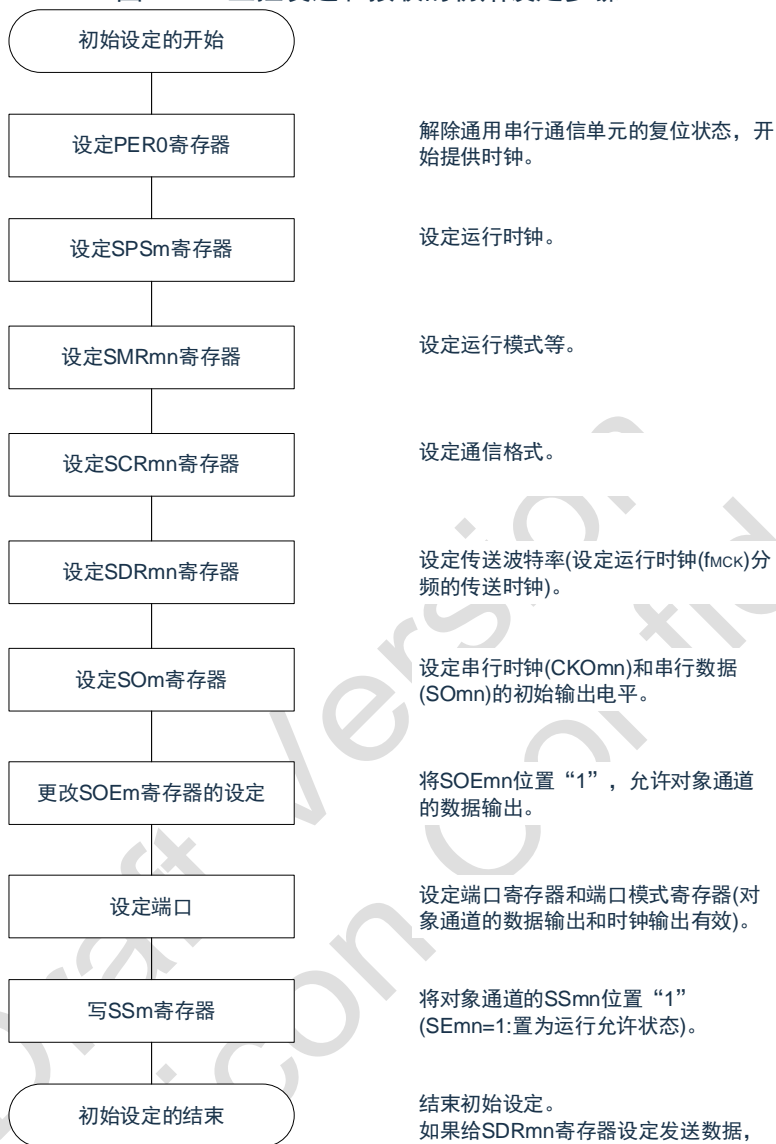




图17-41 主控发送和接收的中止步骤

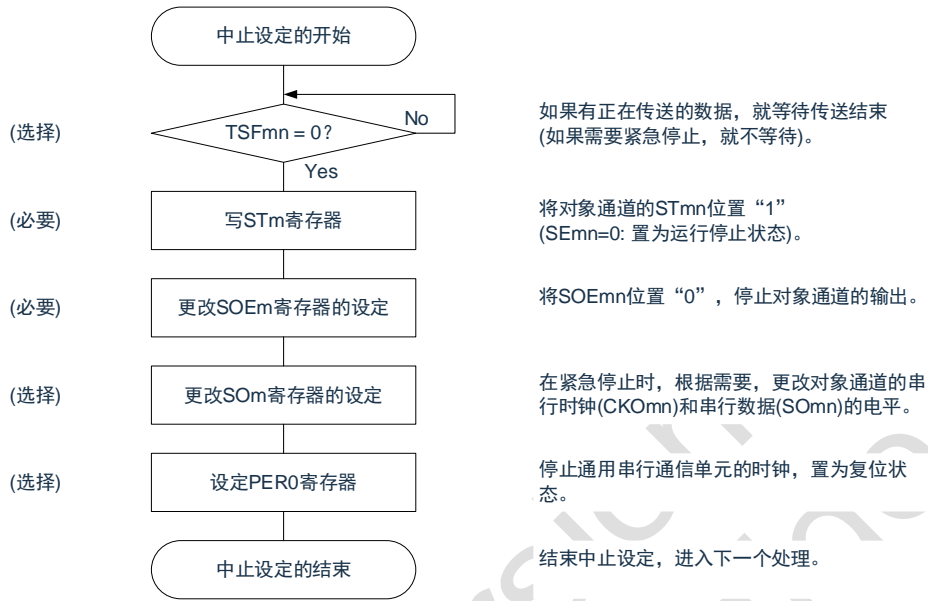
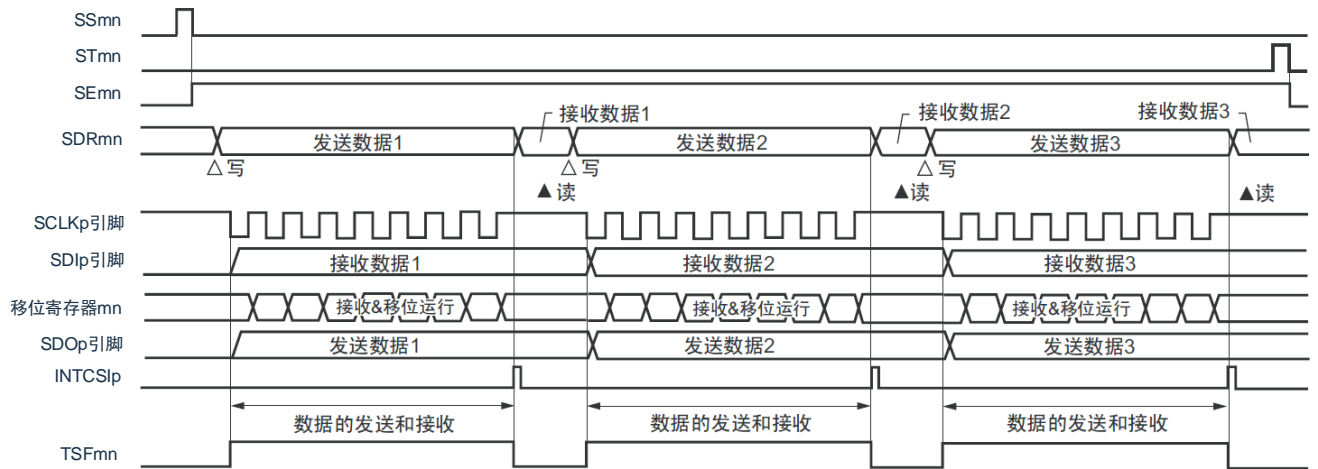


图17-42 重新开始主控发送和接收的设定步骤



(3) 处理流程（单次发送和接收模式）

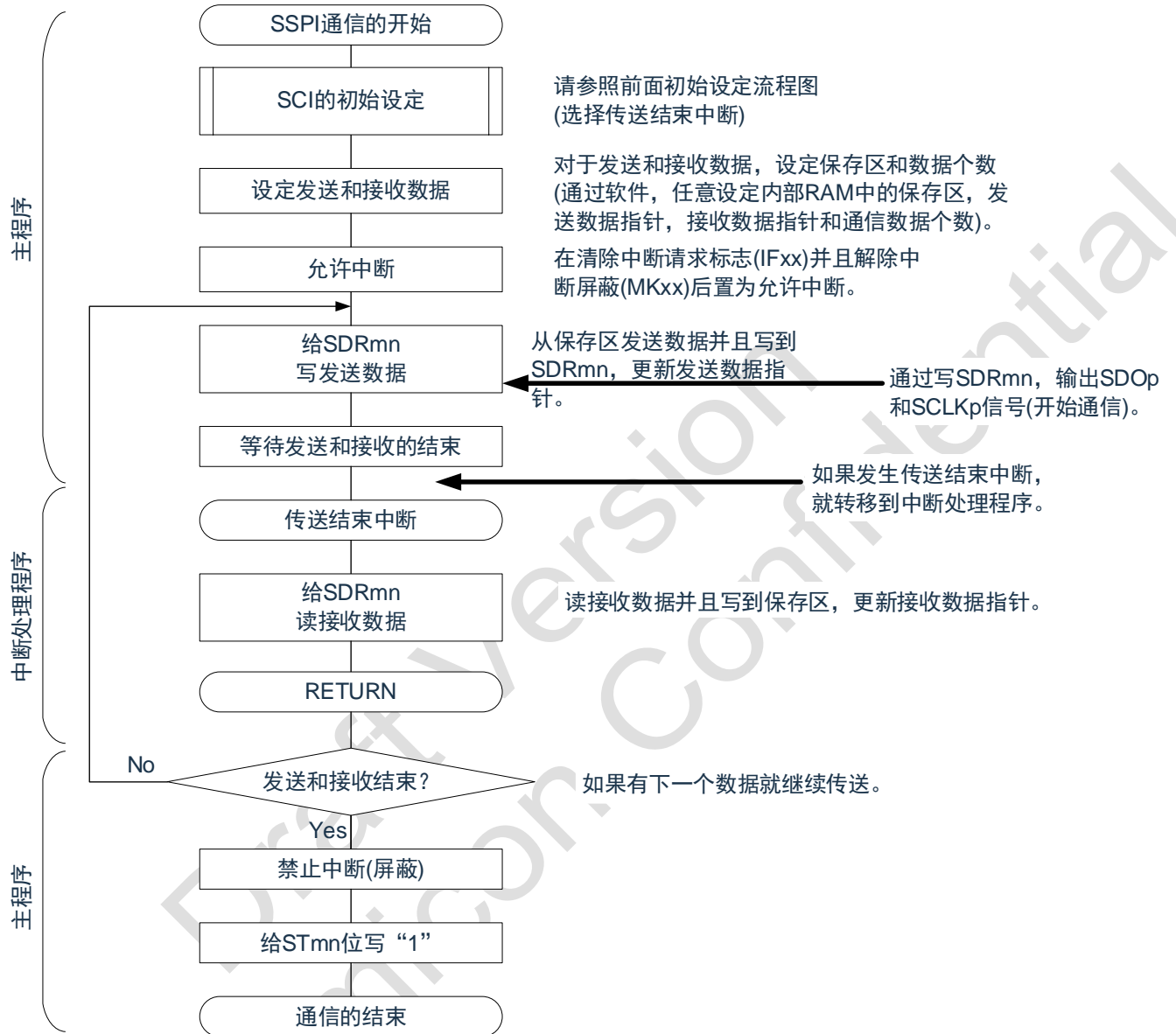
图17-43 主控发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）n：通道号（n=0）p：SSPI号（p=00、20）

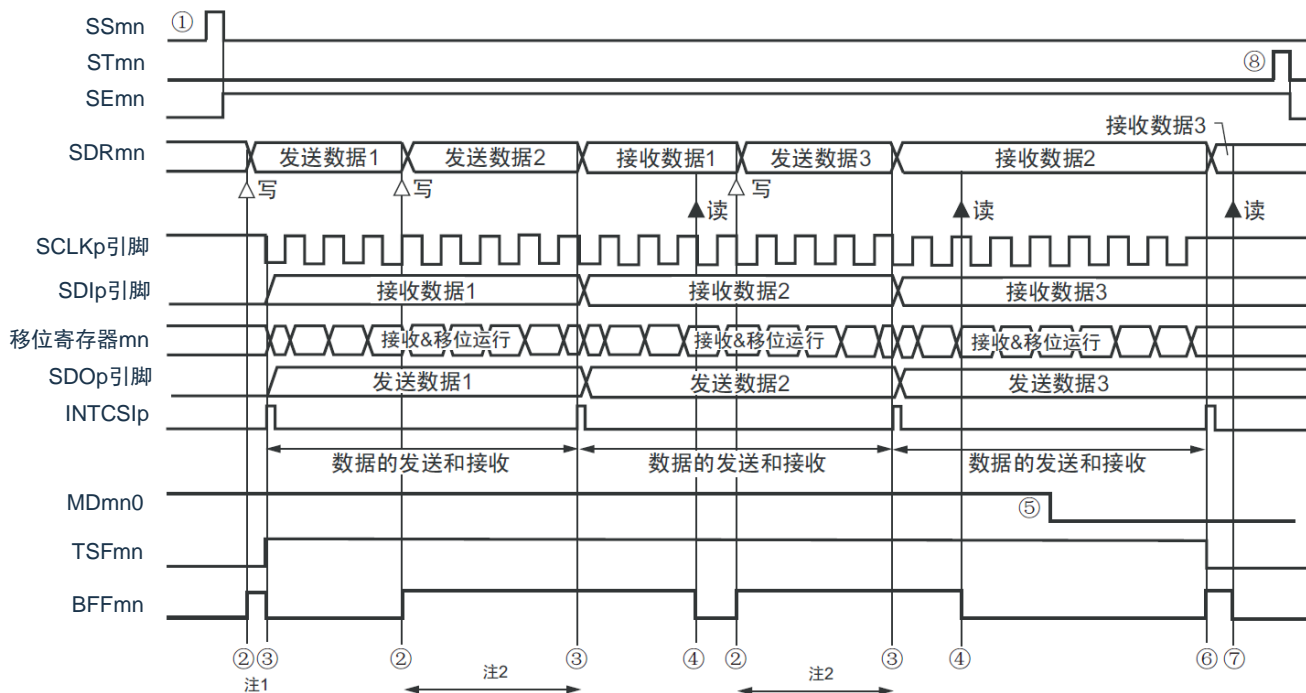
Draft Version  
Cmsemicon Confidential

图17-44 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17-45 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）

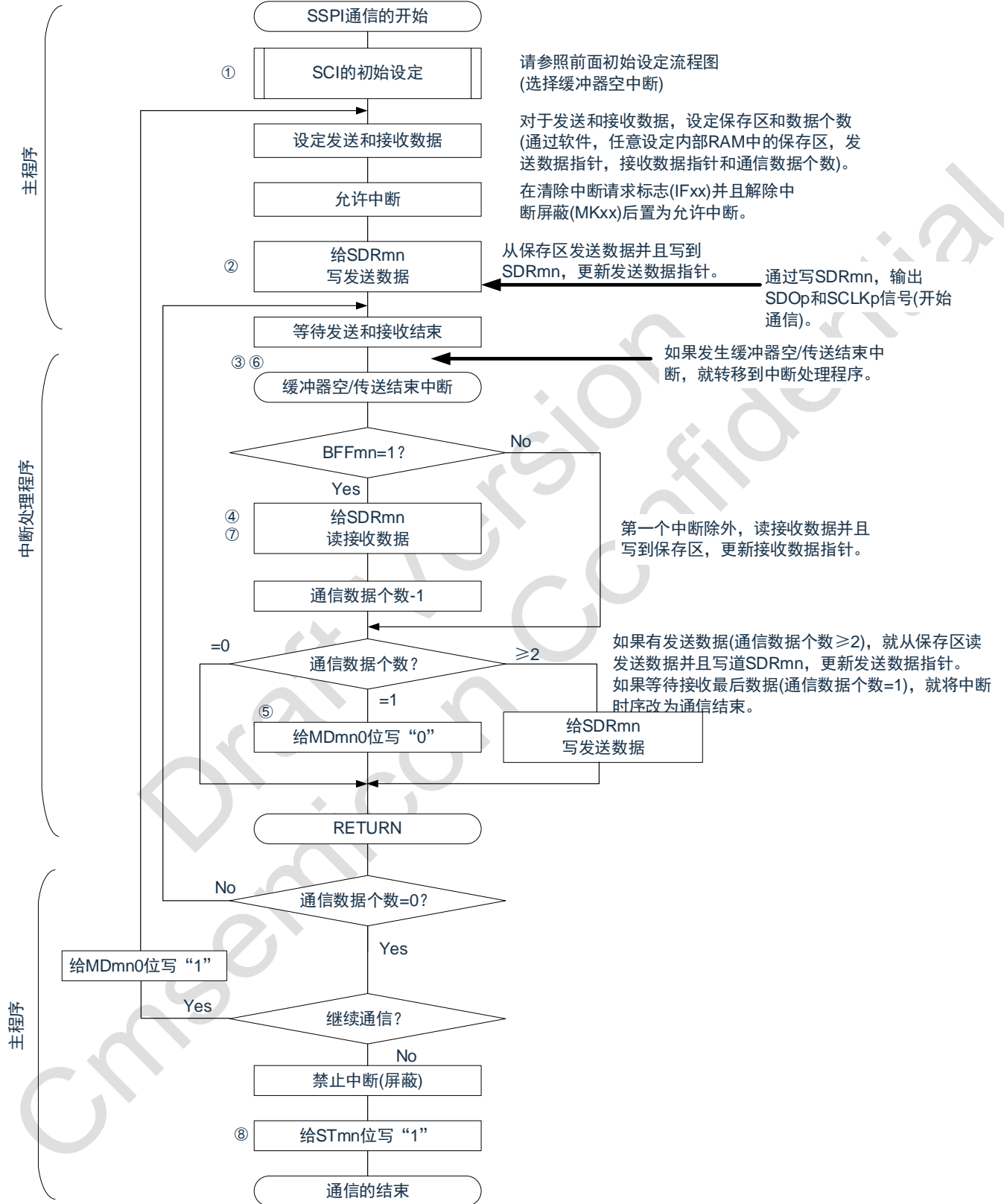


注：1.如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。  
 2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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17-46 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) p: SSPI号 (p=00、20)

图17-46 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17-45 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

## 17.5.4 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本产品将数据发送到其他设备的运行。

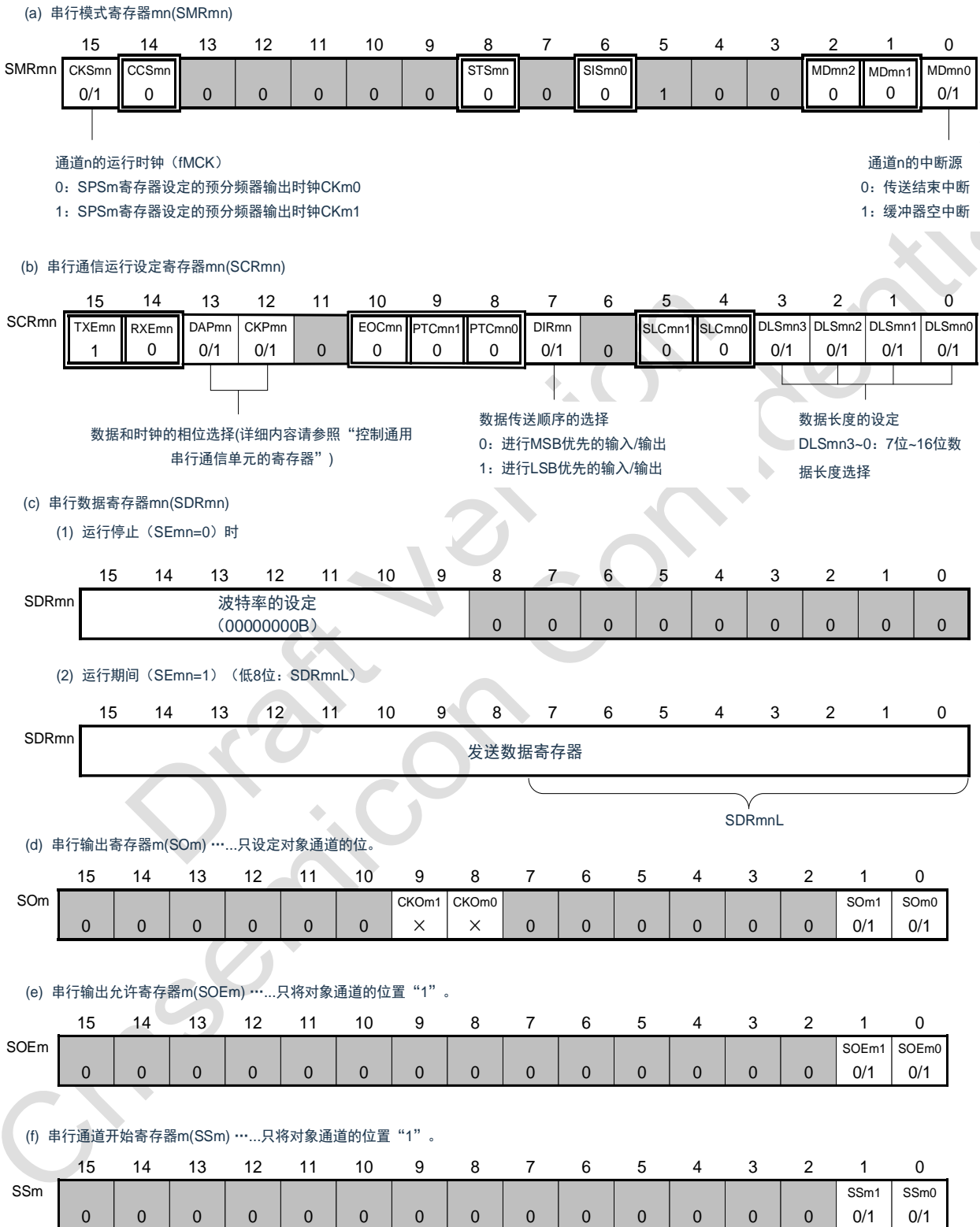
3线串行 I/O	SSPI00	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00	SCLK20
	SDO00	SDO20
中断	INTSSPI00	INTSSPI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。	
错误检测标志	只有溢出错误检测标志（OVFmn）。	
传送数据长度	7 位~16 位	
传送速率	$f_{MCK}/6[\text{Hz}]$ <sup>注1, 2</sup>	
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 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、SCLK20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 $f_{MCK}/6[\text{Hz}]$ 。  
 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)

(1) 寄存器的设定

图17-47 3线串行I/O (SSPI00、SSPI20)  
从属发送时的寄存器设定内容例子



备注1.m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

2.  : 在SSPI从属发送模式中为固定设定。  : 不能设定(设定初始值)。  
 ×: 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, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。



(2) 操作步骤

图17-48 从属发送的初始设定步骤

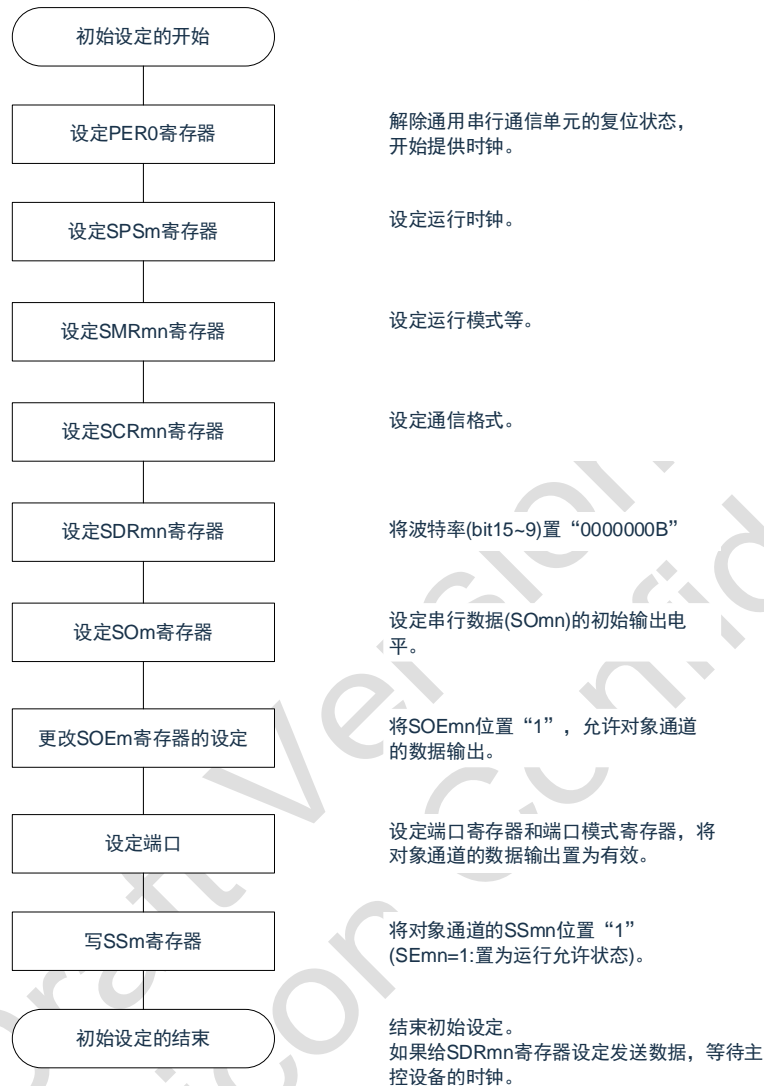


图17-49 从属发送的中止步骤

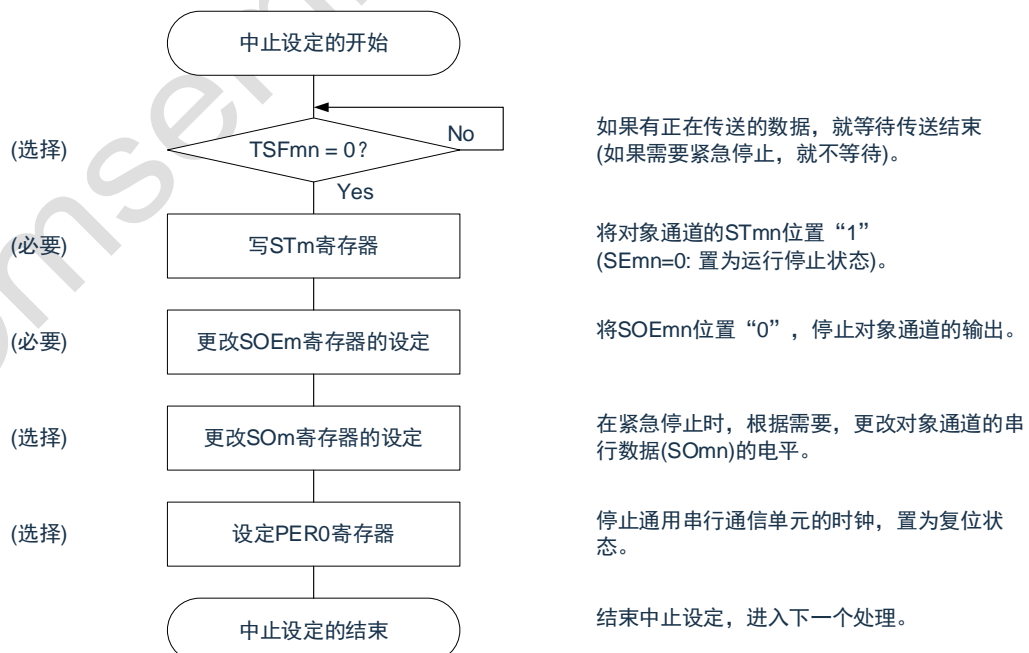


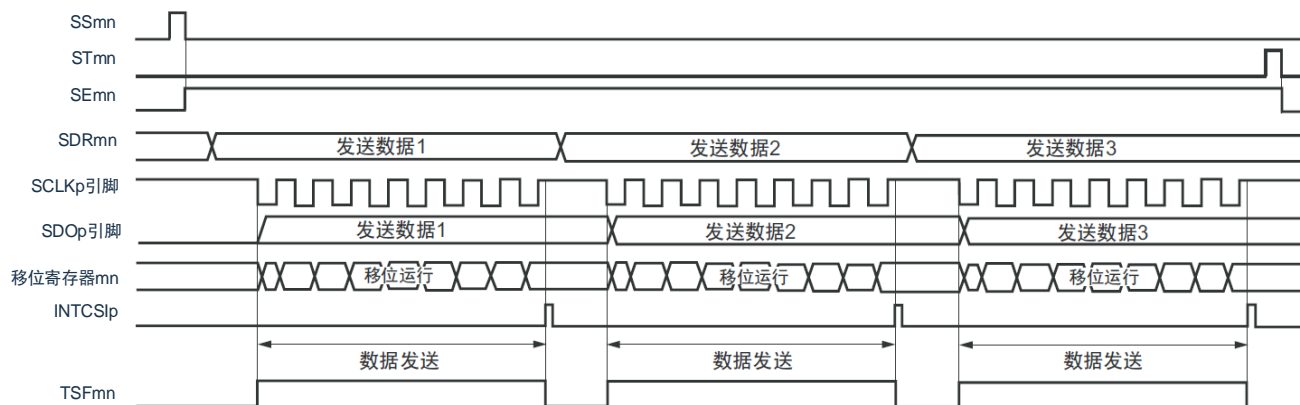
图17-50 重新开始从属发送的设定步骤



备注: 如果在中止设定中改写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定而不是进行重新开始的设定。

(3) 处理流程（单次发送模式）

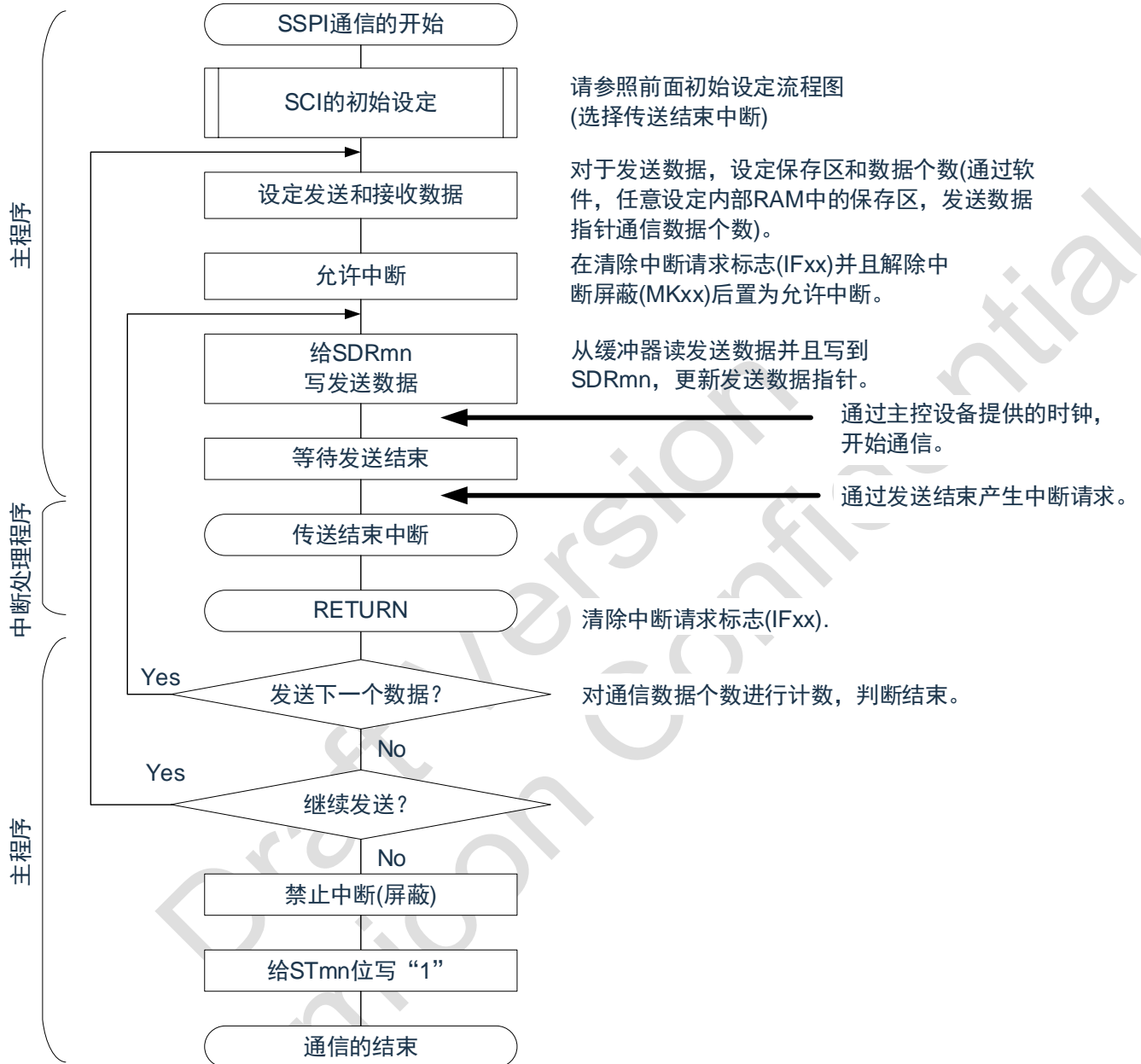
图17-51 从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注： m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

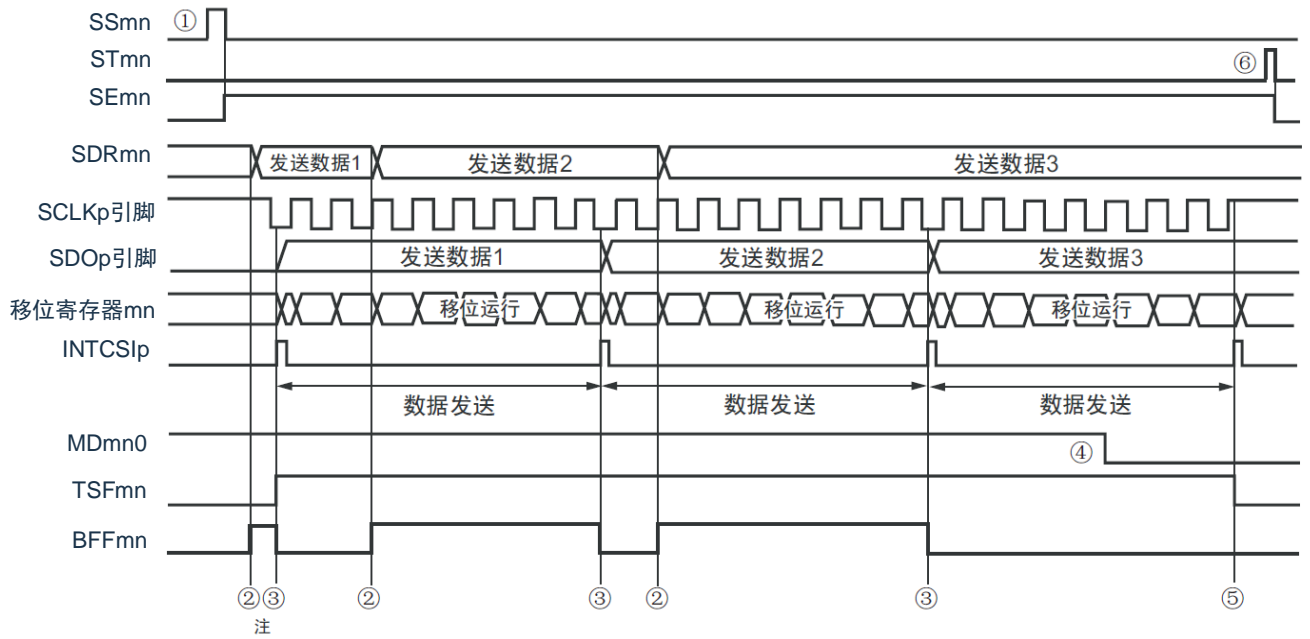
Draft Version  
Cmsemicon Confidential

图17-52 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17-53 从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

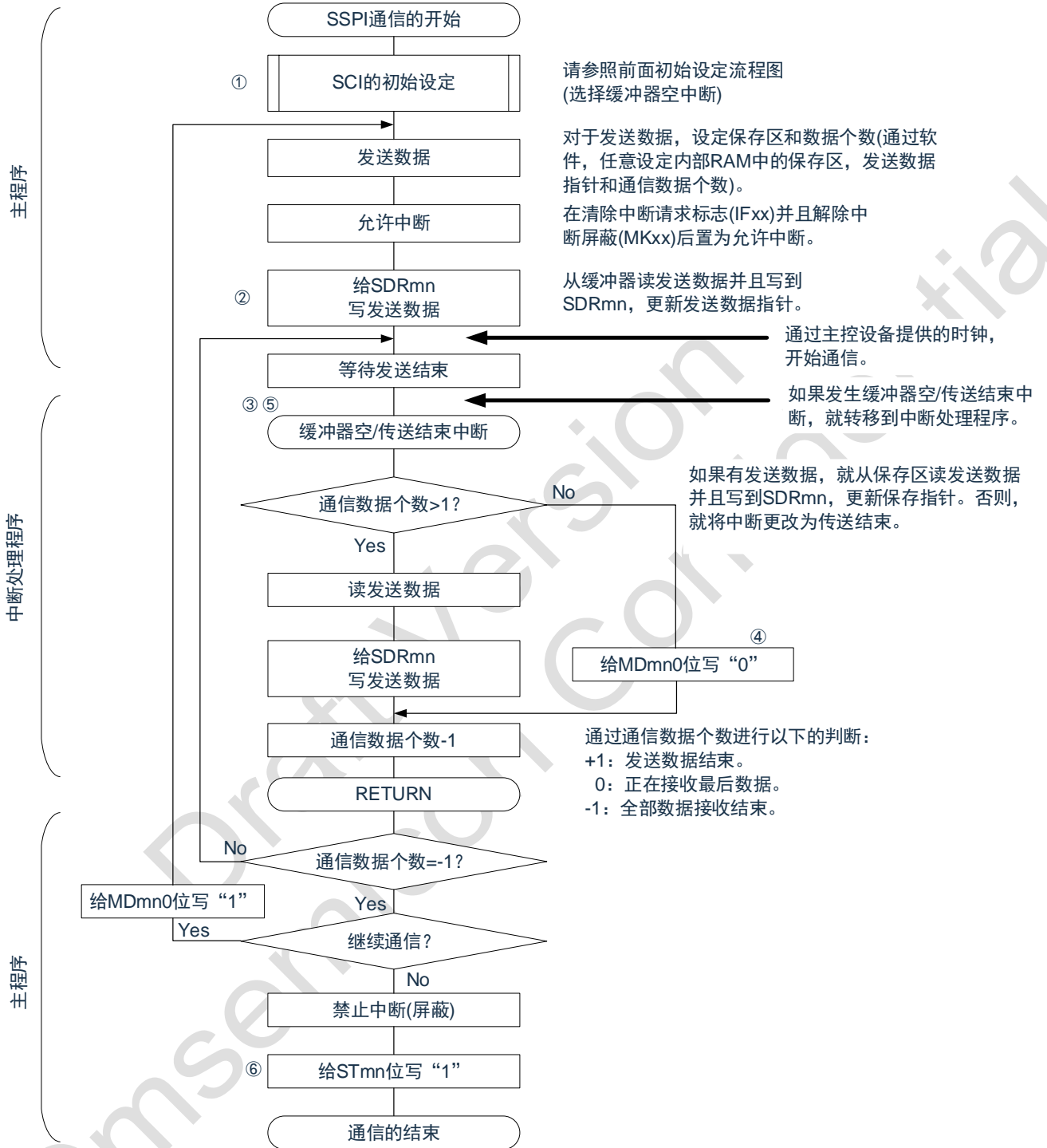


注：如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。

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） p：SSPI号（p=00、20）

图17-54 从属发送（连续发送模式）的流程图



备注： 图中的①~⑥对应“图17-53 从属发送（连续发送模式）的时序图”中的①~⑥。

## 17.5.5 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本产品从其他设备接收数据的运行。

3 线串行 I/O	SSPI00	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00 SDI00	SCLK20 SDI20
中断	INTSSPI00	INTSSPI20
	只限于传送结束中断（禁止设定缓冲器空中断）。	
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~16 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、SCLK20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  $f_{\text{MCK}}/6[\text{Hz}]$ 。

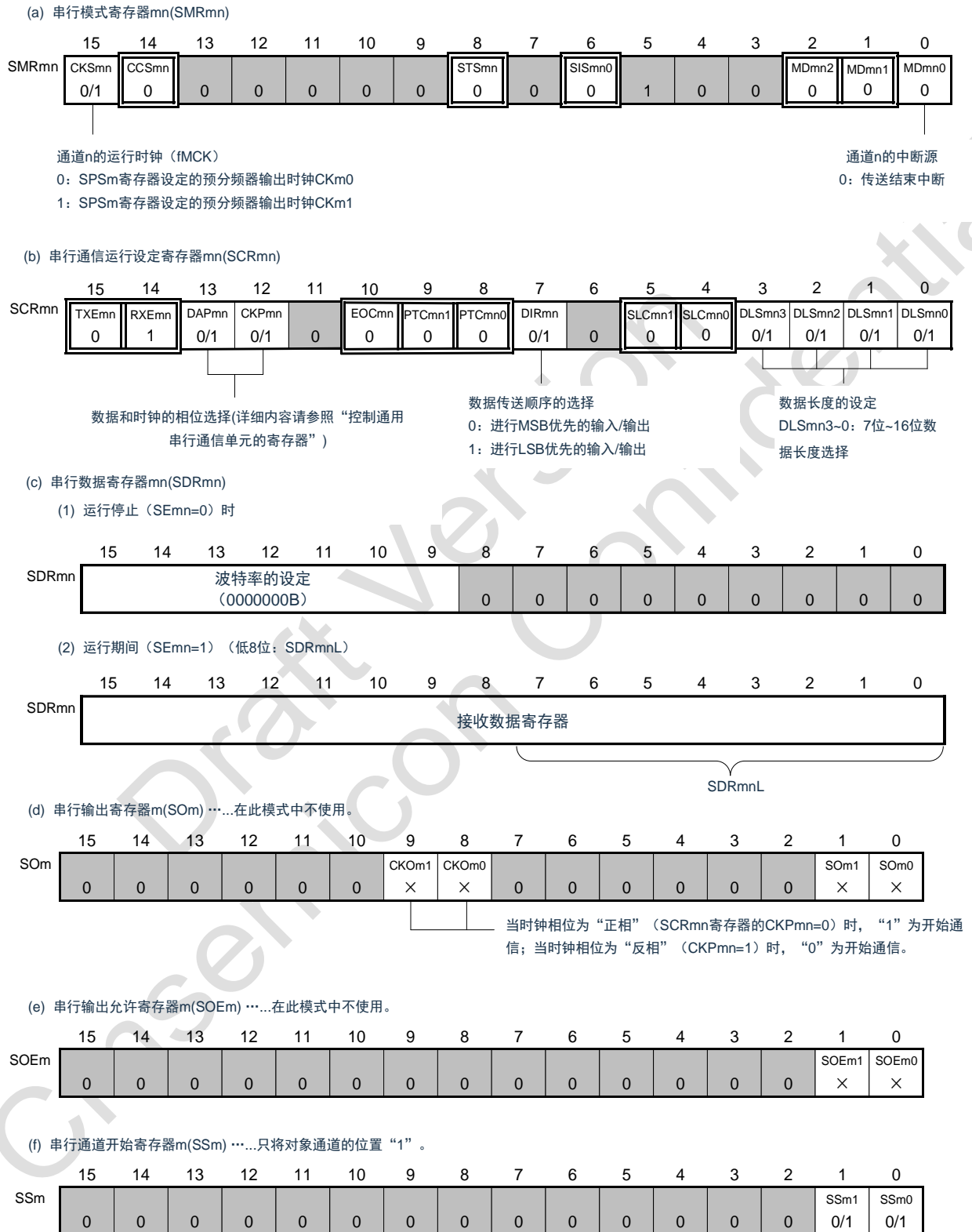
2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注: 1. $f_{\text{MCK}}$ : 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)

(1) 寄存器的设定

图17-55 3线串行I/O (SSPI00、SSPI20)  
从属接收时的寄存器设定内容例子



备注1.m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

2. □: 在从属接收模式中为固定设定。■: 不能设定(设定初始值)。  
 ×: 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, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。



2) 操作步骤

图17-56 从属接收的初始设定步骤

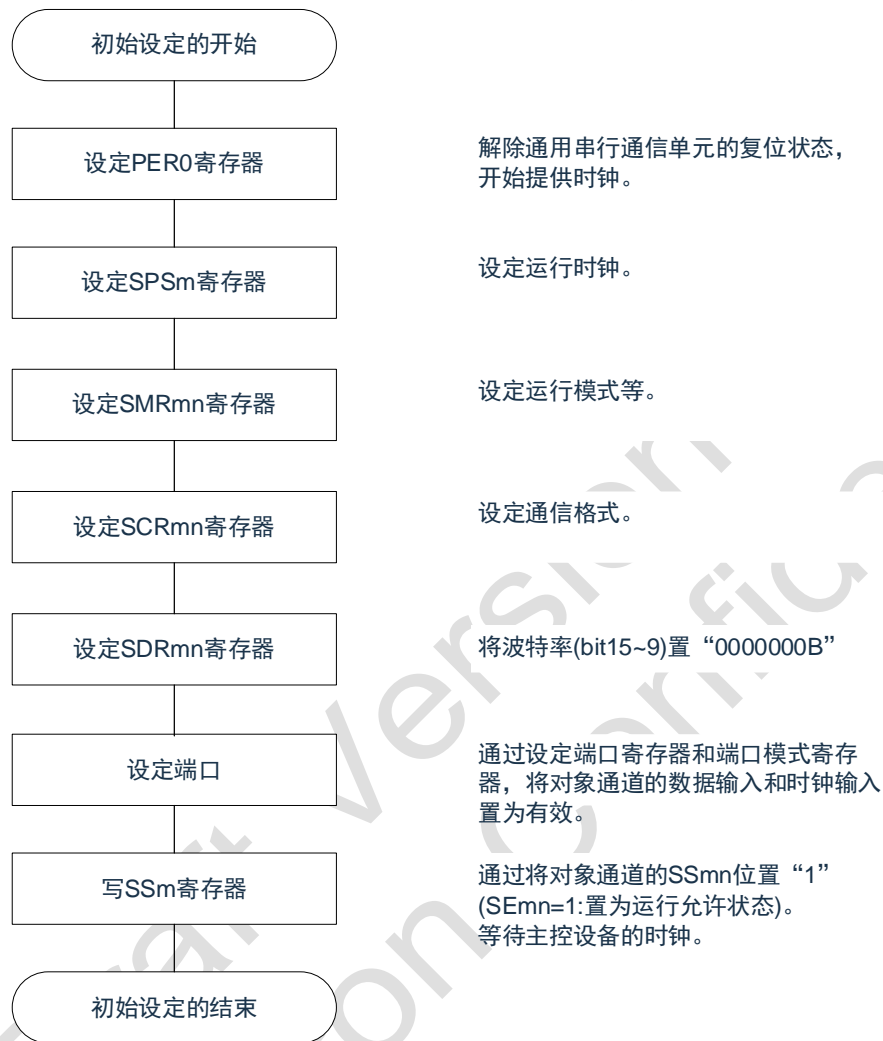


图17-57 从属接收的中止步骤

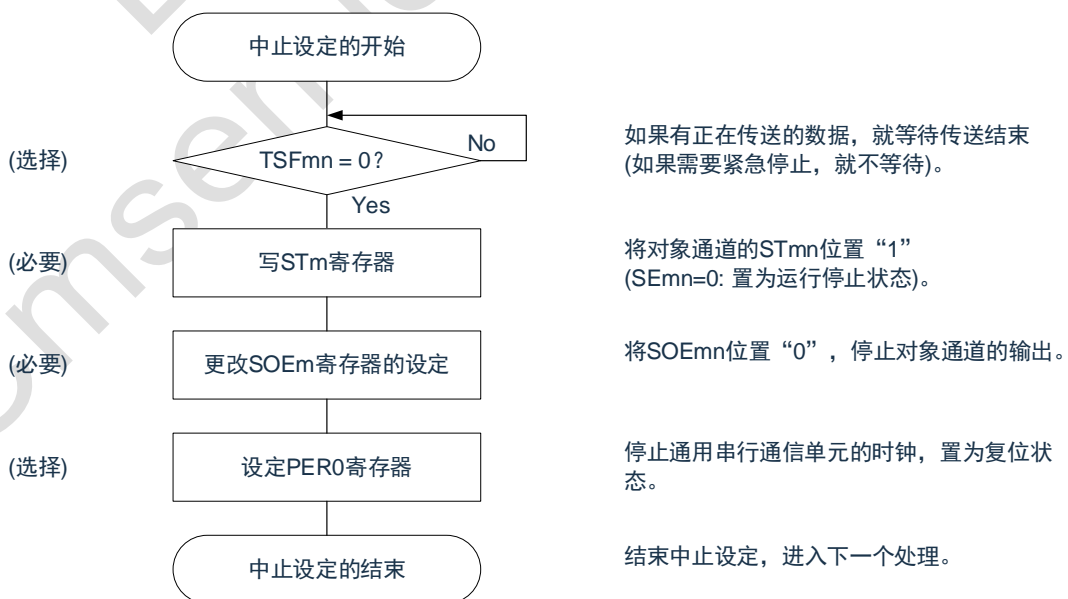


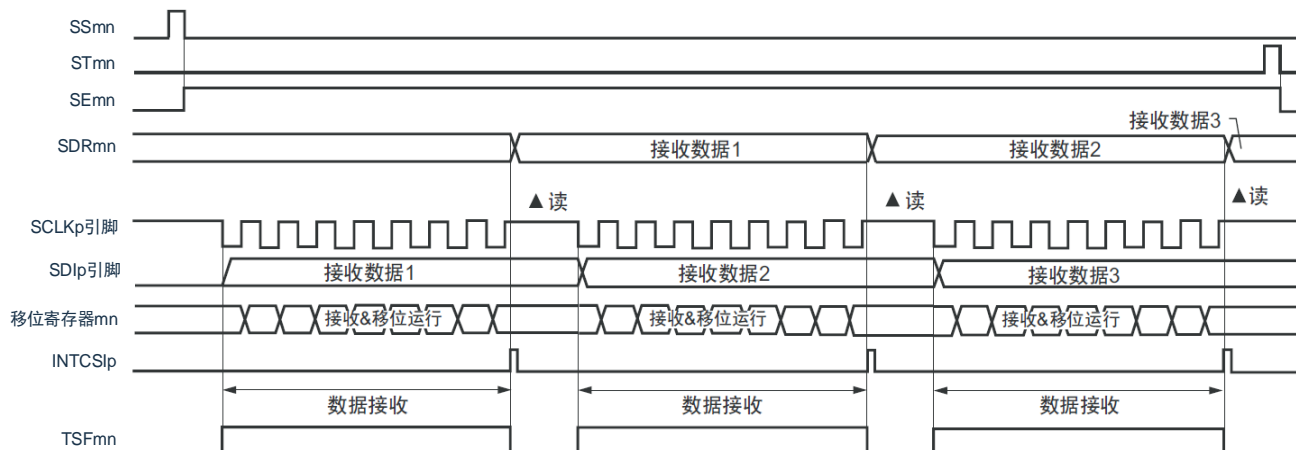
图17-58 重新开始从属接收的设定步骤



备注: 如果在中止设定中改写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定而不是进行重新开始的设定。

(3) 处理流程（单次接收模式）

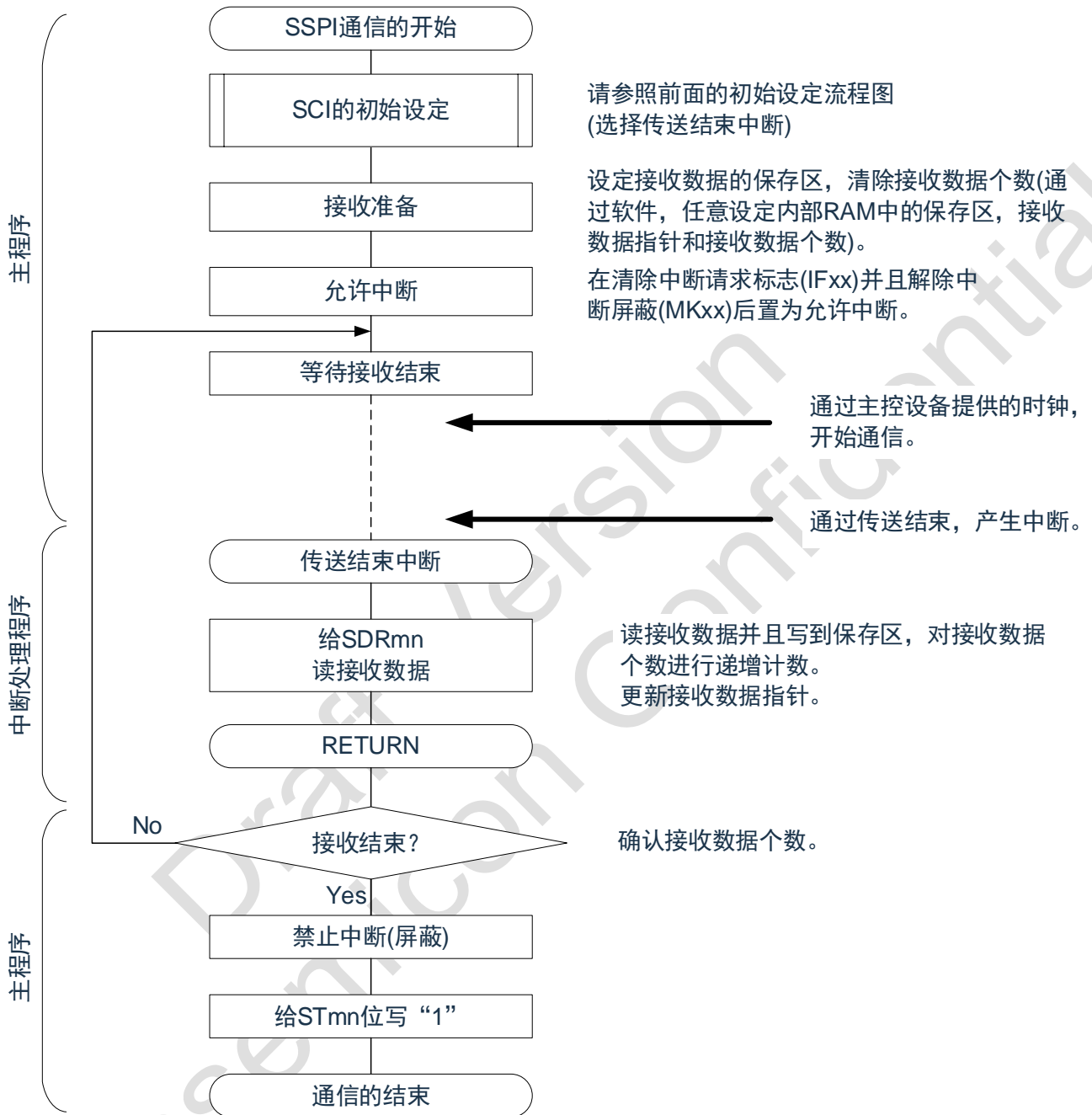
图17-59 从属接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）n：通道号（n=0）p：SSPI号（p=00、20）

Draft Version  
Cmsemicon Confidential

图17-60 从属接收（单次接收模式）的流程图



## 17.5.6 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本产品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	SSPI20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCLK00、SDI00、SDO00	SCLK20、SDI20、SDO20
中断	INTSSPI00	INTSSPI20
	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~16 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、SCLK20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 $f_{MCK}/6$ [Hz]。

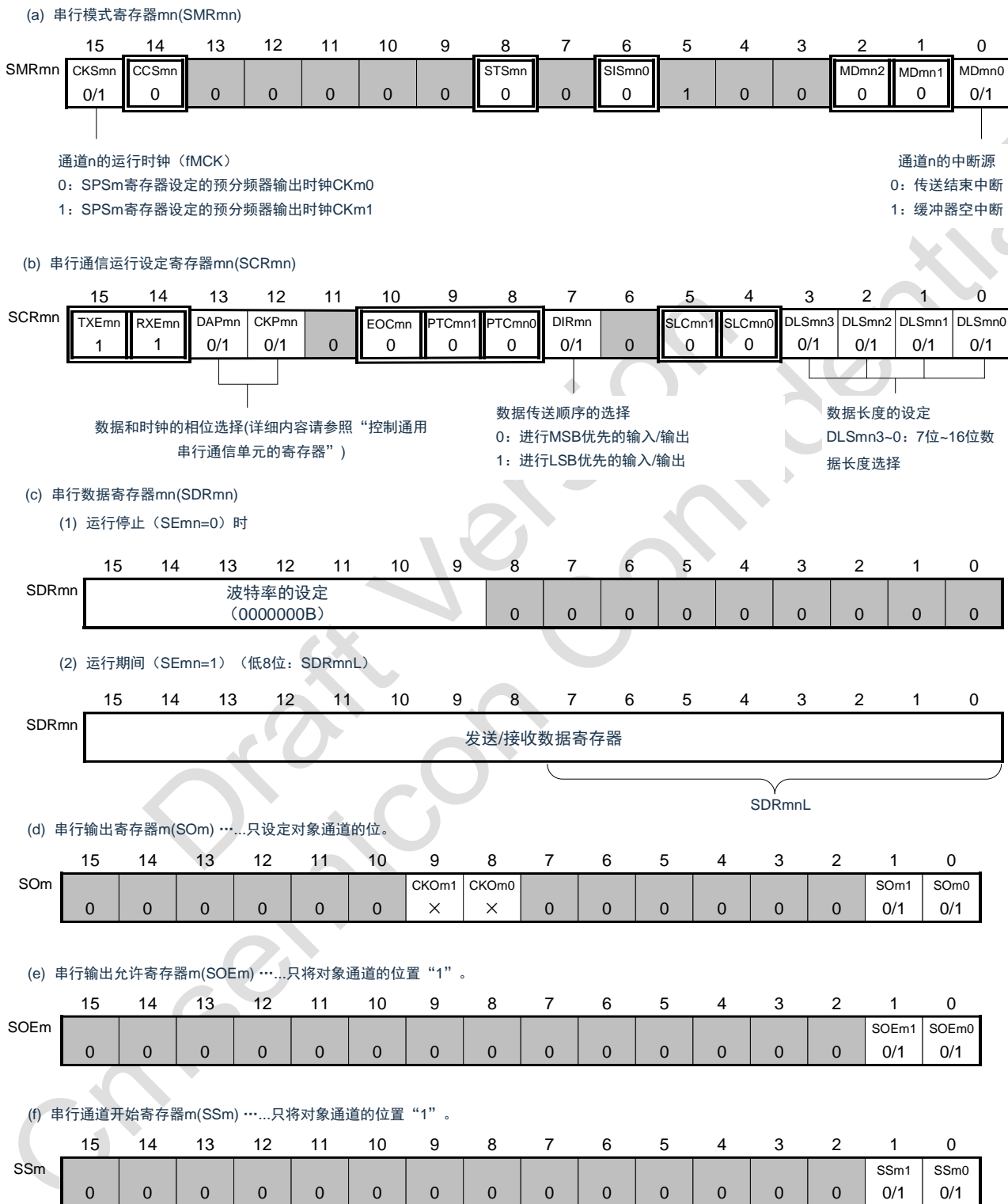
2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

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）

(1) 寄存器的设定

图17-61 3线串行I/O (SSPI00、SSPI20)  
从属发送和接收时的寄存器设定内容例子



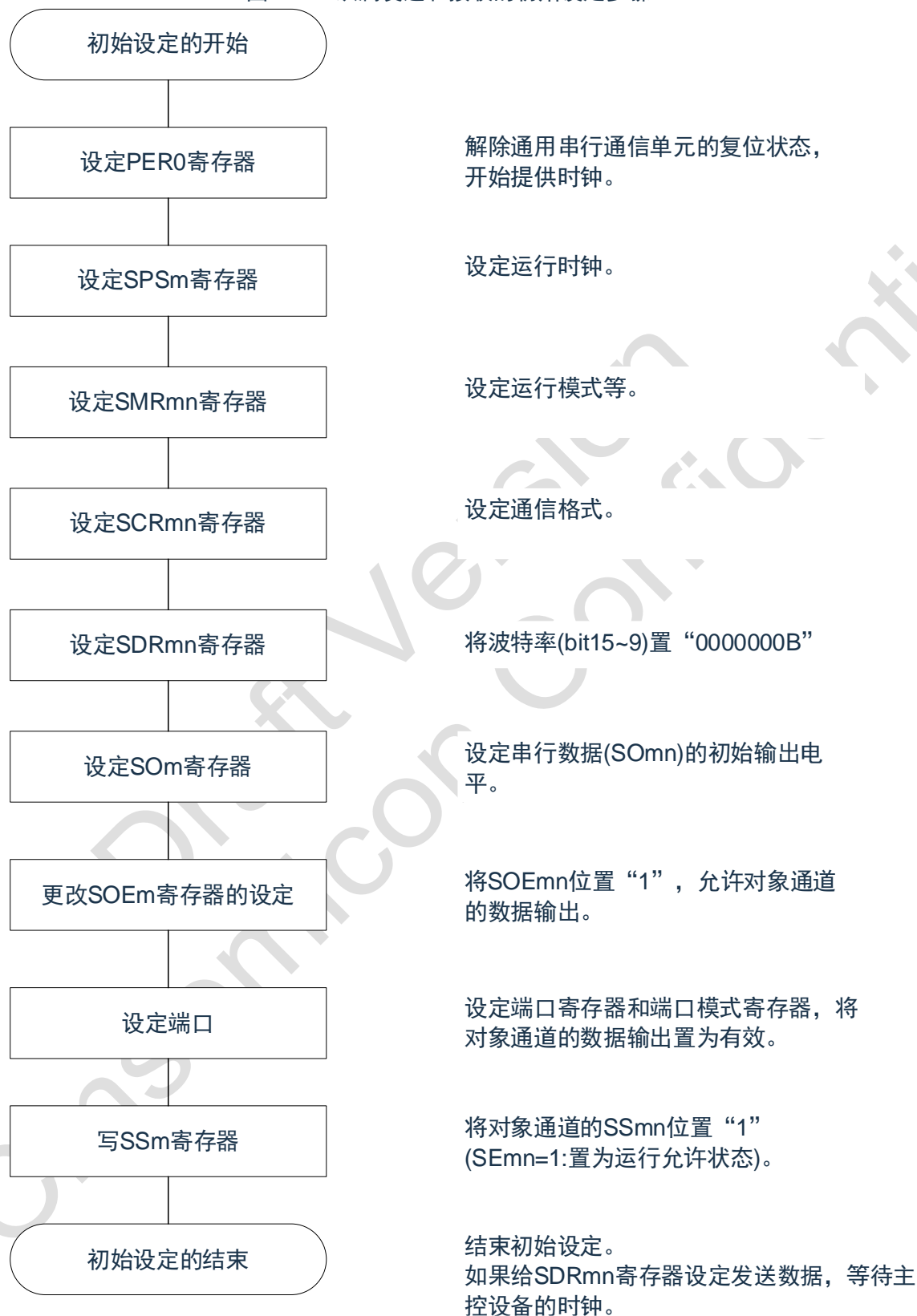
注意: 在主导设备开始输出时钟前, 必须给SDRmn寄存器设定发送数据。

备注: 1.m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

2.  : 在SSPI从属发送和接收模式中为固定设定。  : 不能设定(设定初始值)。  
 ×: 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, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-62 从属发送和接收的初始设定步骤



注意：在主控设备开始输出时钟前，必须给SDRmn寄存器设定发送数据。

图17-63 从属发送和接收的中止步骤

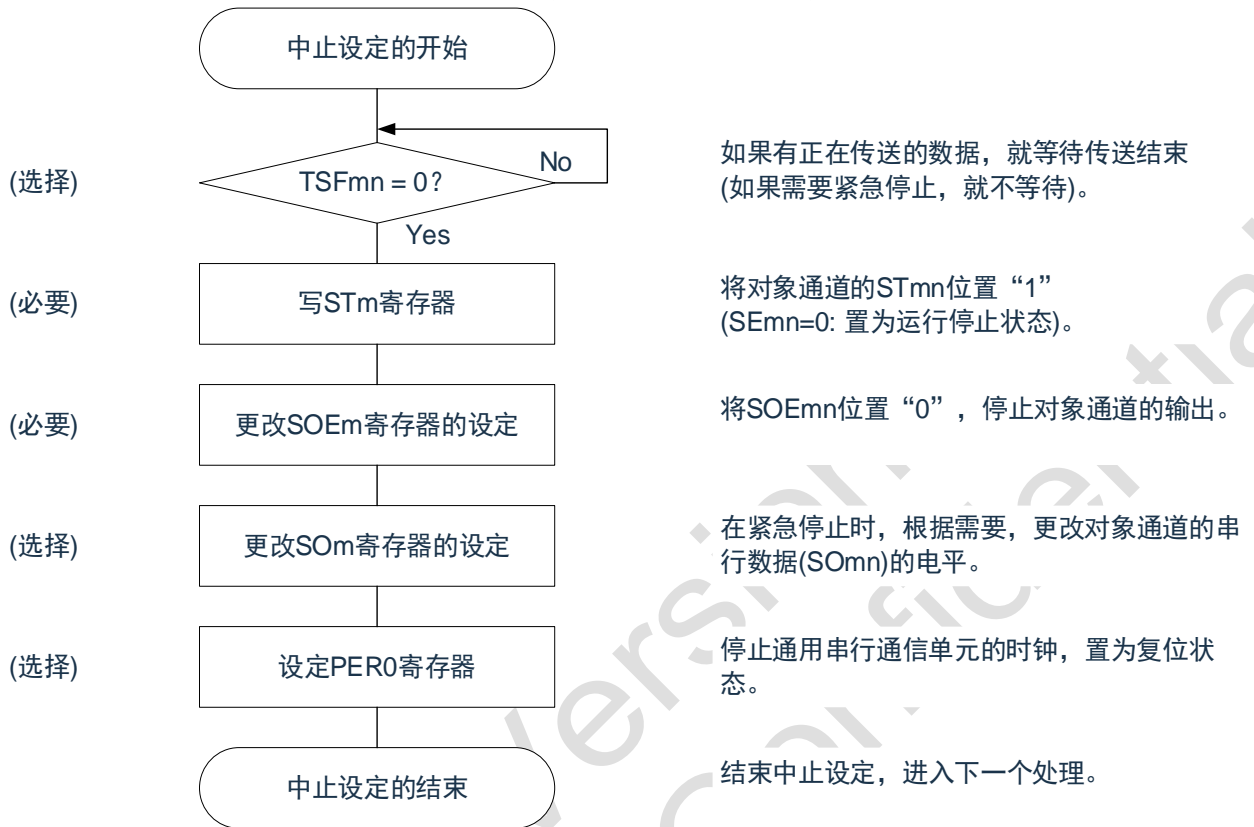




图17-64 重新开始从属发送和接收的设定步骤

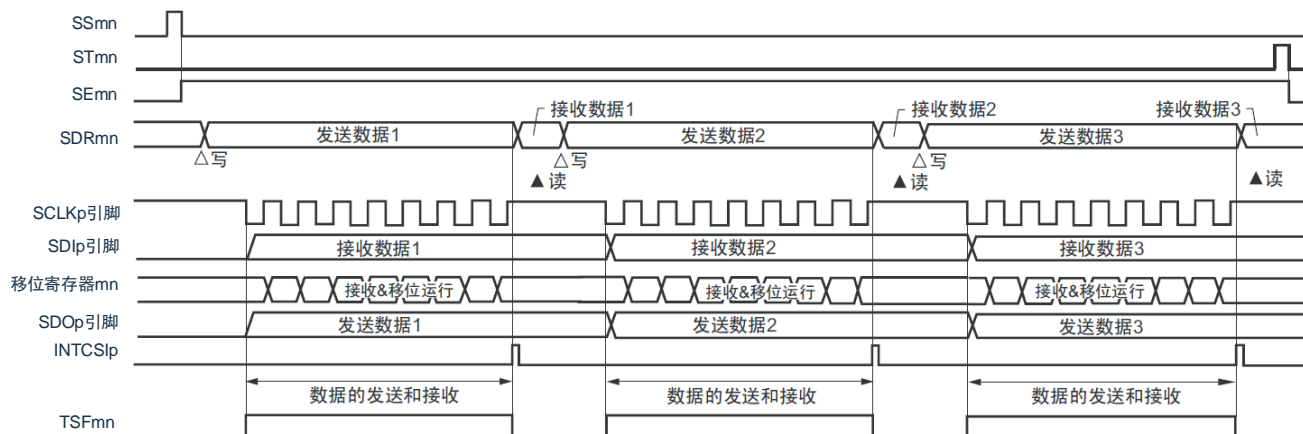


注意1. 在主机设备开始输出时钟前, 必须给SDRmn寄存器设定发送数据。

2. 如果在中止设定中改写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定而不是进行重新开始的设定。

(3) 处理流程（单次发送和接收模式）

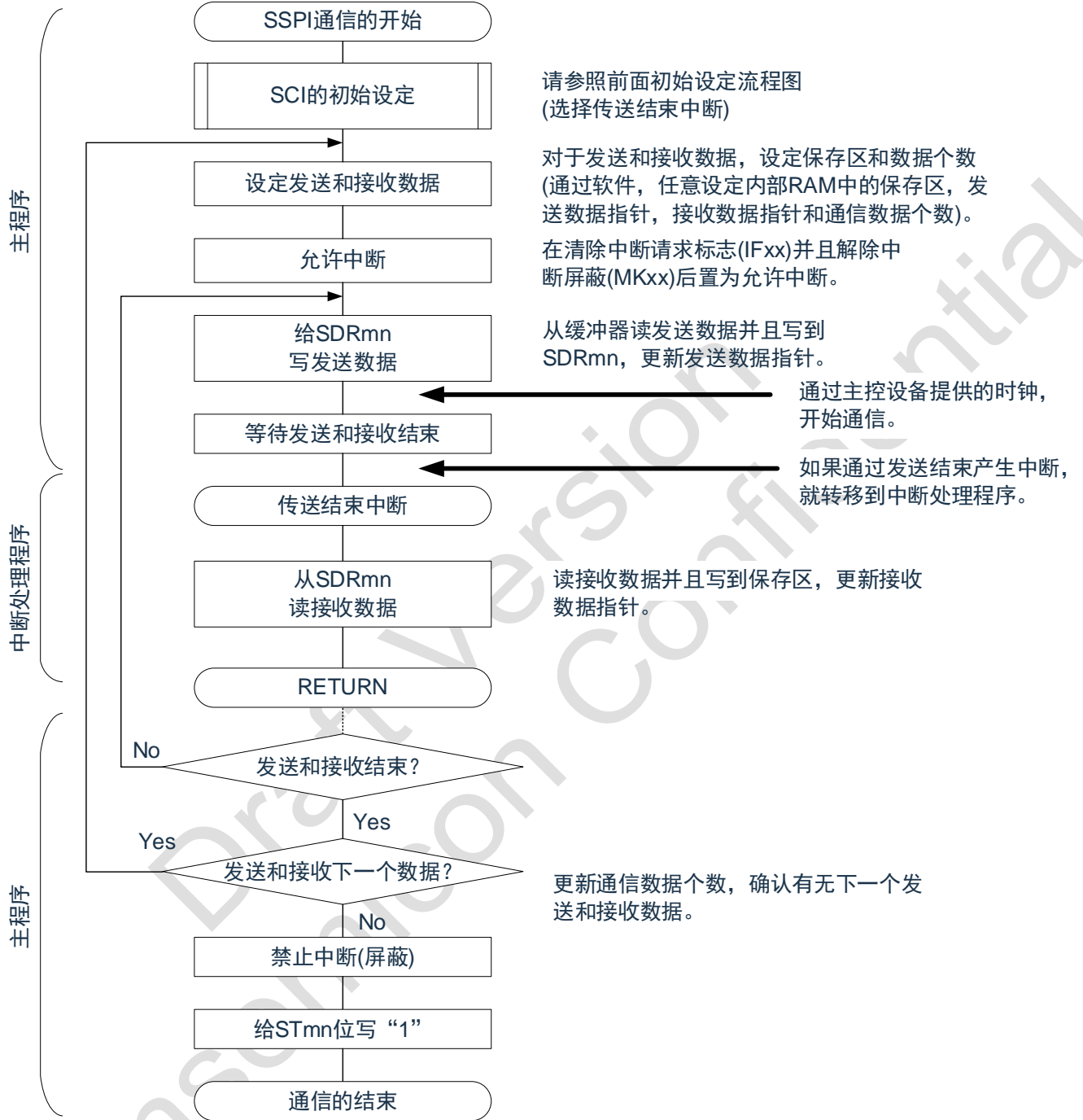
图17-65 从属发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注： m: 单元号 (m=0、1) n: 通道号 (n=0) p: SSPI号 (p=00、20)

Draft Version  
Cmsemicon Confidential

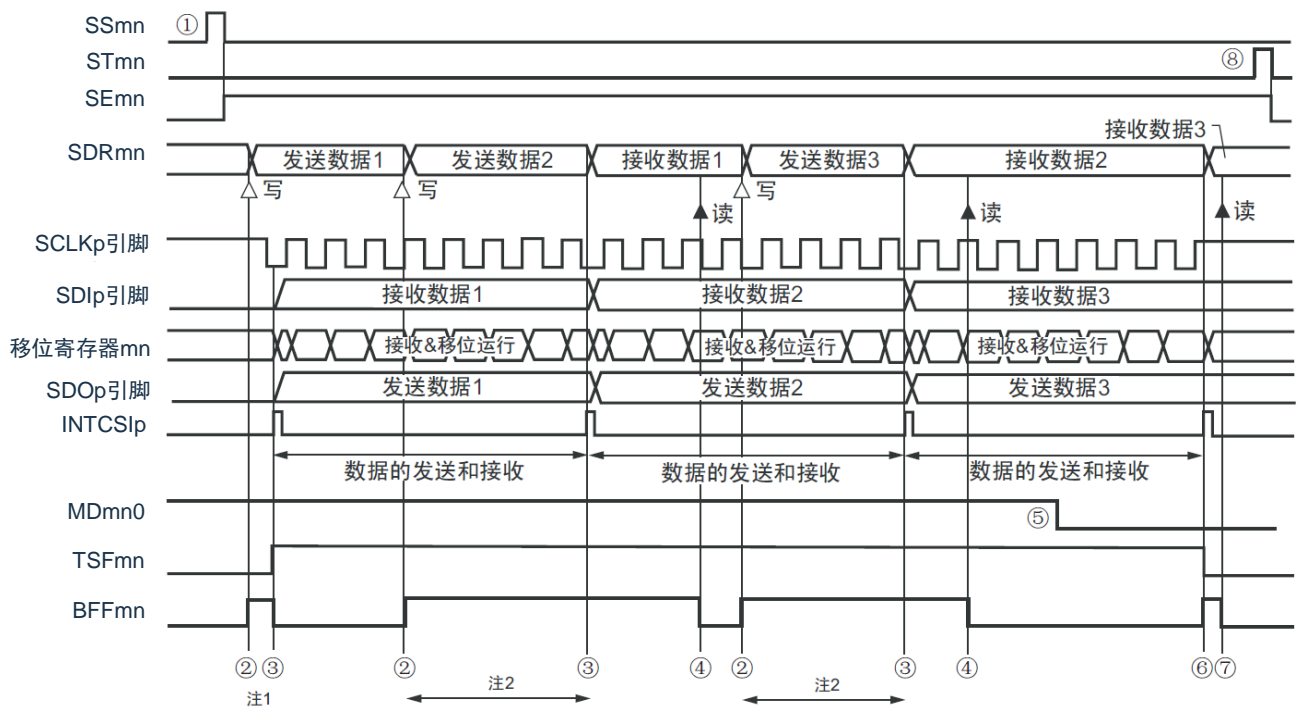
图17-66 从属发送和接收（单次发送和接收模式）的流程图



注意： 在主控设备开始输出时钟前，必须给SIOp寄存器设定发送数据。

(4) 处理流程（连续发送和接收模式）

图17-67 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。

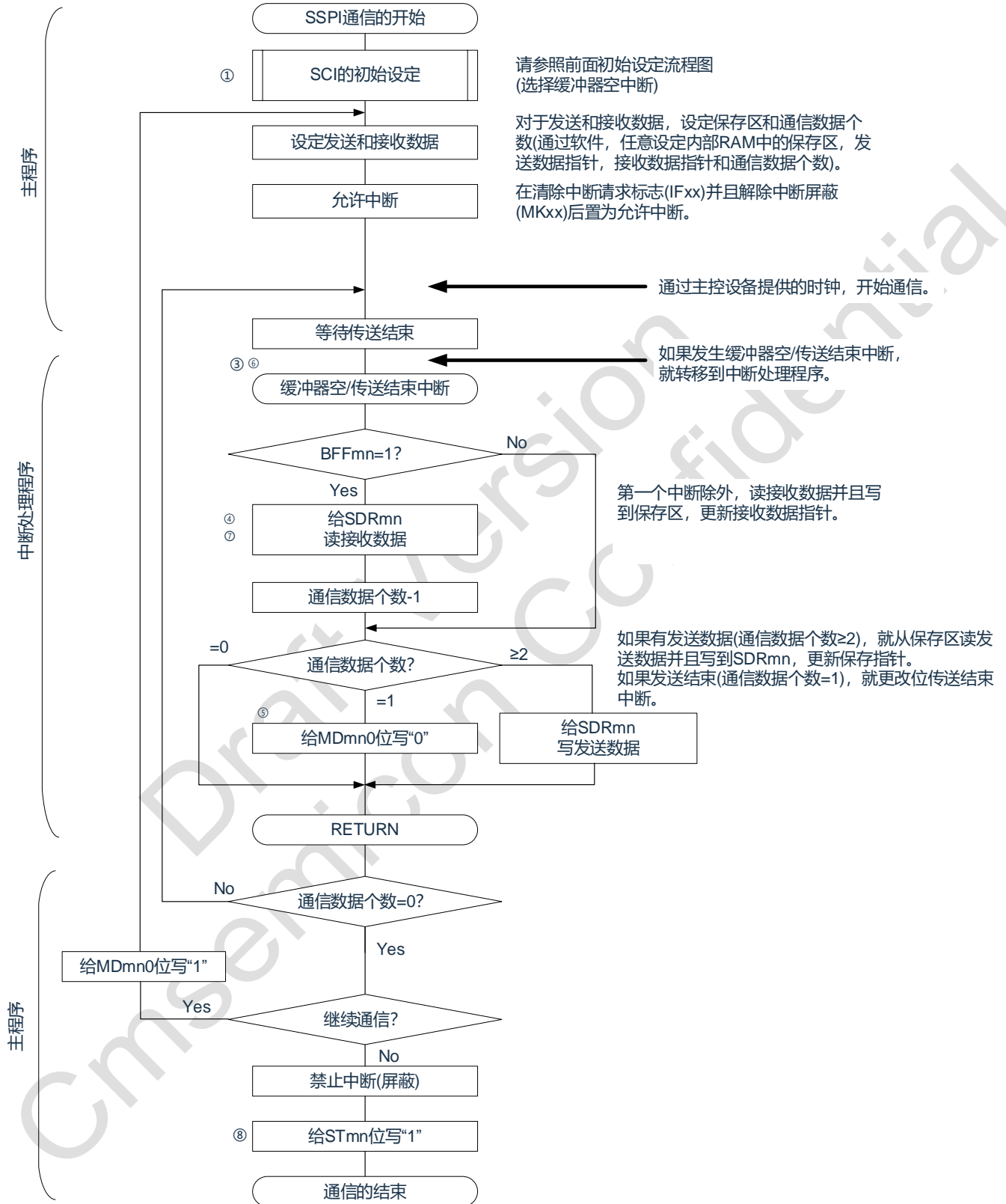
2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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17-68 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) p: SSPI号 (p=00、20)

图17-68 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SDRmn寄存器设定发送数据。

备注： 图中的①~⑧对应“图17-67 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

## 17.5.7 传送时钟频率的计算

3线串行I/O（SSPI00、SSPI20）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

$$\text{(传送时钟频率)} = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\} \text{注}[\text{Hz}]$$

注：容许的最大传送时钟频率为f<sub>MCK</sub>/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<sub>MCK</sub>）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17-2 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 ( $f_{MCK}$ ) 注		
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	$f_{CLK}$	$f_{CLK}=32\text{MHz}$ 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	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	0	$f_{CLK}/2^5$	1MHz
	X	X	X	X	0	1	1	0	0	$f_{CLK}/2^6$	500kHz
	X	X	X	X	0	1	1	1	0	$f_{CLK}/2^7$	250kHz
	X	X	X	X	1	0	0	0	0	$f_{CLK}/2^8$	125kHz
	X	X	X	X	1	0	0	1	0	$f_{CLK}/2^9$	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	0	$f_{CLK}/2^{11}$	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	0	$f_{CLK}/2^{13}$	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	0	$f_{CLK}/2^{15}$	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）

## 17.5.8 在3线串行I/O（SSPI00、SSPI20）通信过程中发生错误时的处理步骤

在3线串行I/O（SSPI00、SSPI20）通信过程中发生错误时的处理步骤如图17-69所示。

图17-69 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（SIRmn）写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SIRmn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）



## 17.6 从属选择输入功能的时钟同步串行通信的运行

每个通道都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~16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定

[时钟控制]

- 输入/输出时钟的相位控制
- 设定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： $\text{Max.f}_{\text{MCK}}/6$

[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 ( $t_{\text{KCY}}$ ) 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17.6.1)
- 从属接收 (参照17.6.2)
- 从属的发送和接收 (参照17.6.3)

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引脚变为高阻。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17-70 从属选择输入功能的结构例子

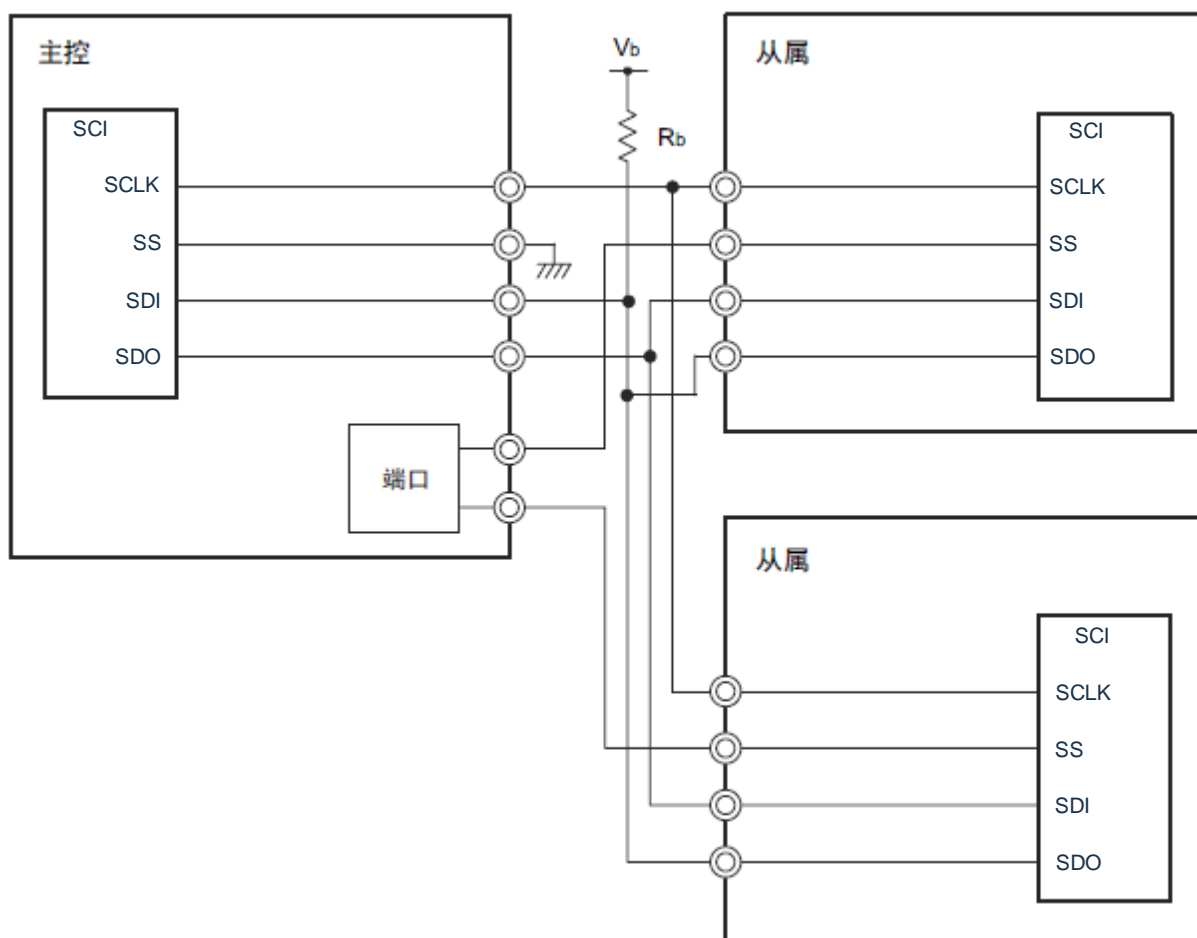
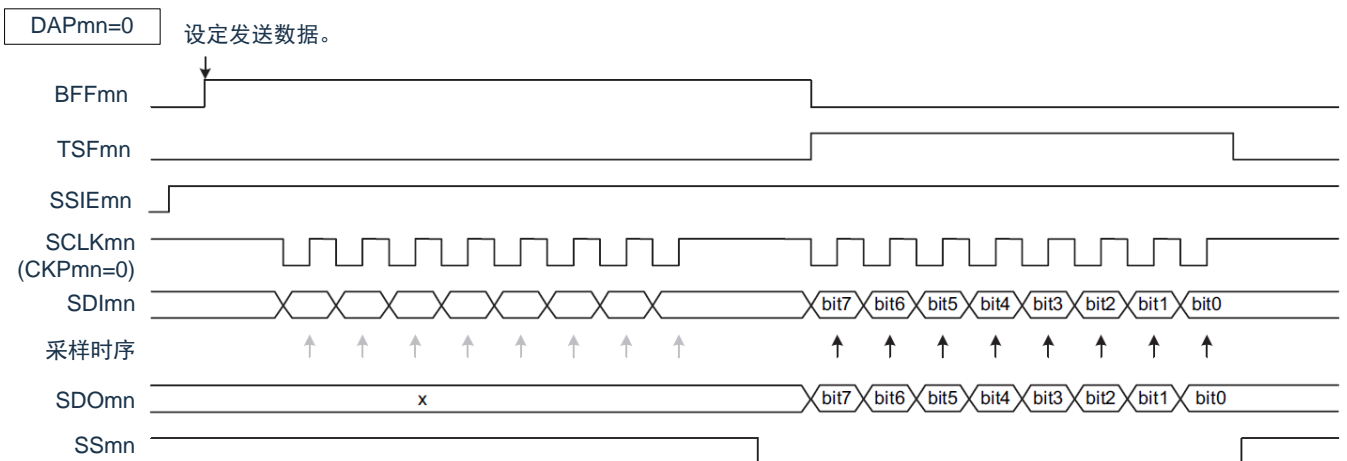
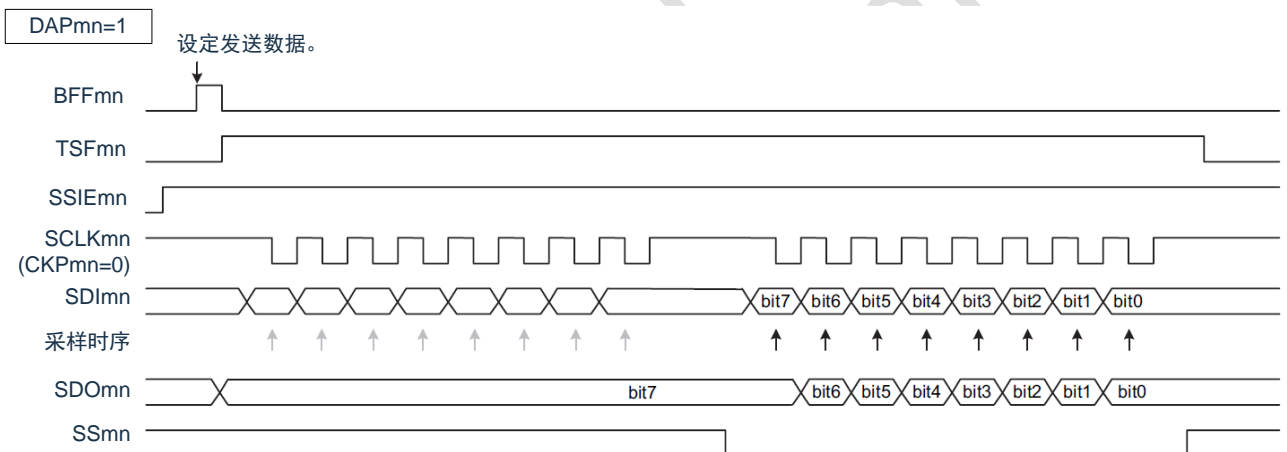


图17-71 从属选择输入功能的时序图



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为高电平期间设定发送数据，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、1） n：通道号（n=0）

## 17.6.1 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本产品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~16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SCLKmn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]。

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。

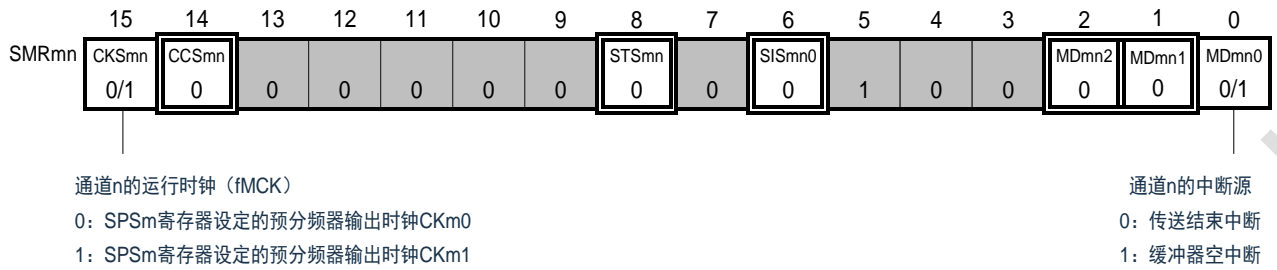
备注： 1. $f_{MCK}$ ：对象通道的运行时钟频率

2.m：单元号（m=0） n：通道号（n=0）

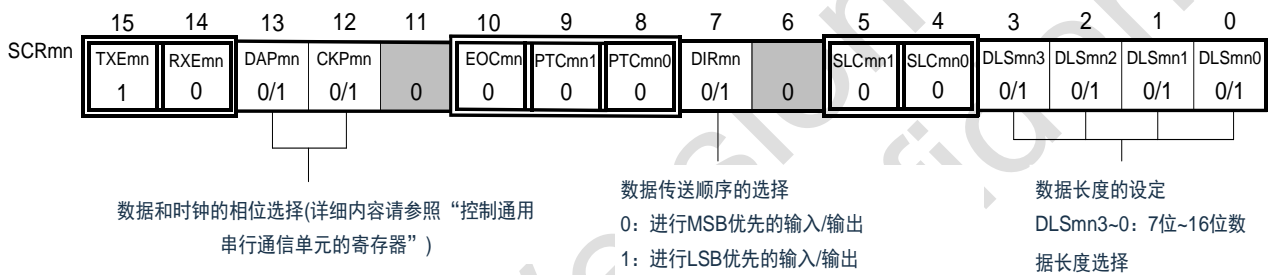
(1) 寄存器的设定

图17-72 从属选择输入功能 (SSPImn) 从属发送时的寄存器设定内容例子(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)

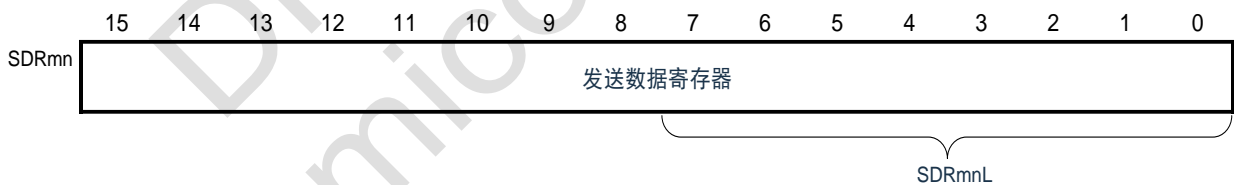


(c) 串行数据寄存器mn(SDRmn)

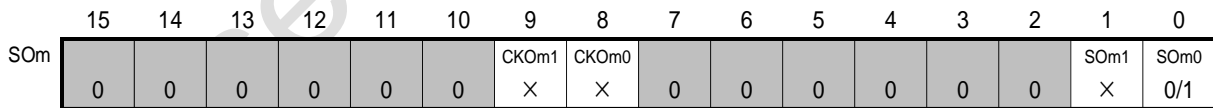
(1) 运行停止 (SEmn=0) 时



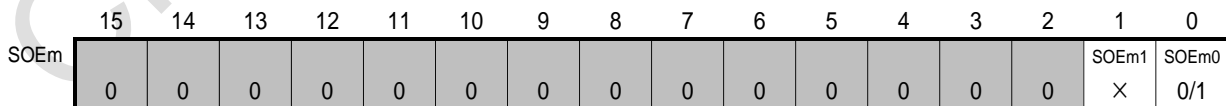
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(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”。



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定。■: 不能设定(设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图17-72 从属选择输入功能 (SSPImn) 从属发送时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器(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	0	0	SSm1 X	SSm0 0/1

(g) 从属选择功能启用寄存器 (SSE).....这是SSPImn从属通道(单元m的通道n)的SSPImn引脚的控制。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIm0 0/1

0: SSPIm0引脚的输入值无效  
1: SSPIm0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定。 : 不能设定(设定初始值)。

X: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

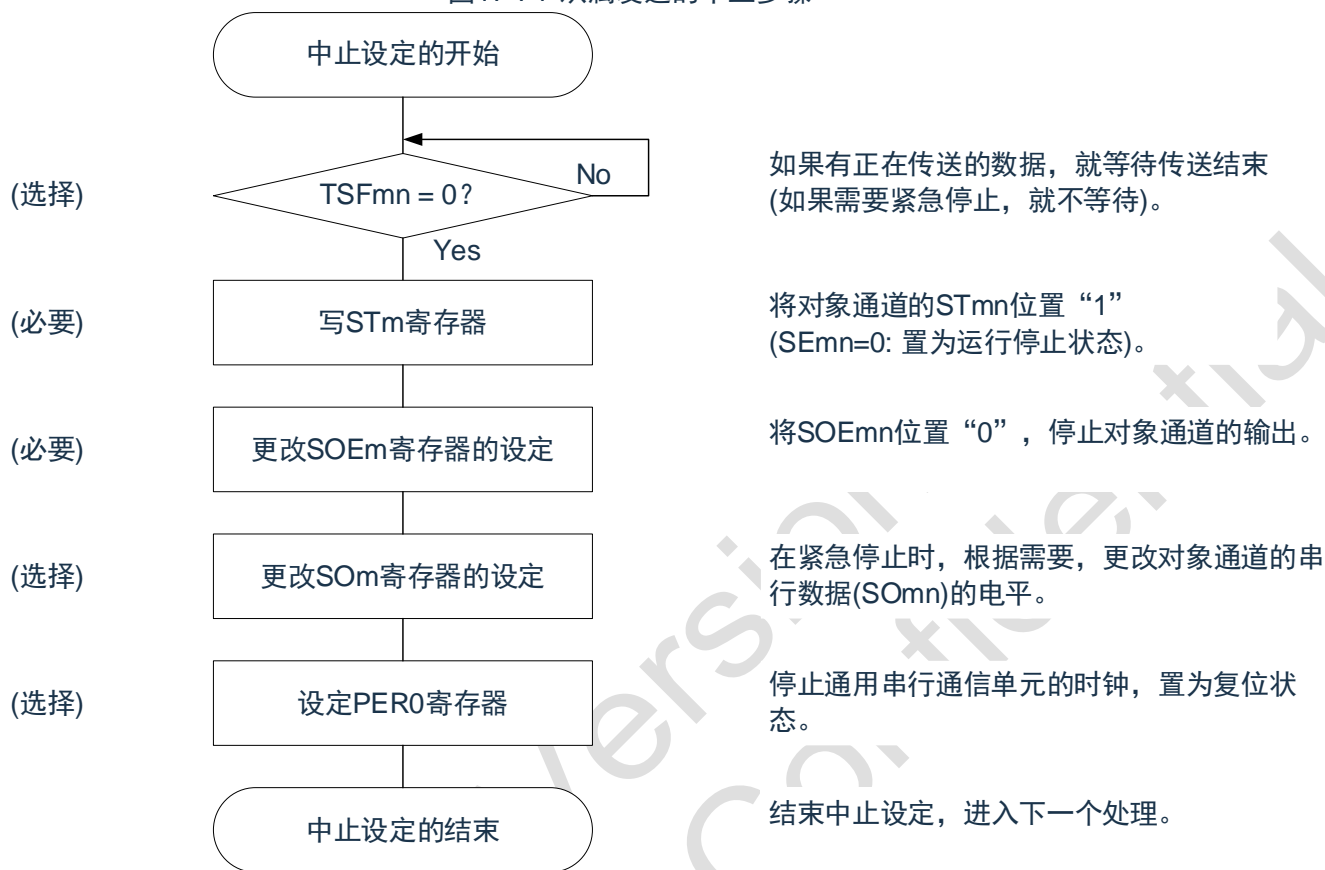
(2) 操作步骤

图17-73 从属发送的初始设定步骤



备注：m：单元号 (m=0) n：通道号 (n=0) p：SSPI号 (p=00)

图17-74 从属发送的中止步骤



备注: 1.如果在中止设定中改写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定而不是进行重新开始设定。

2.m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

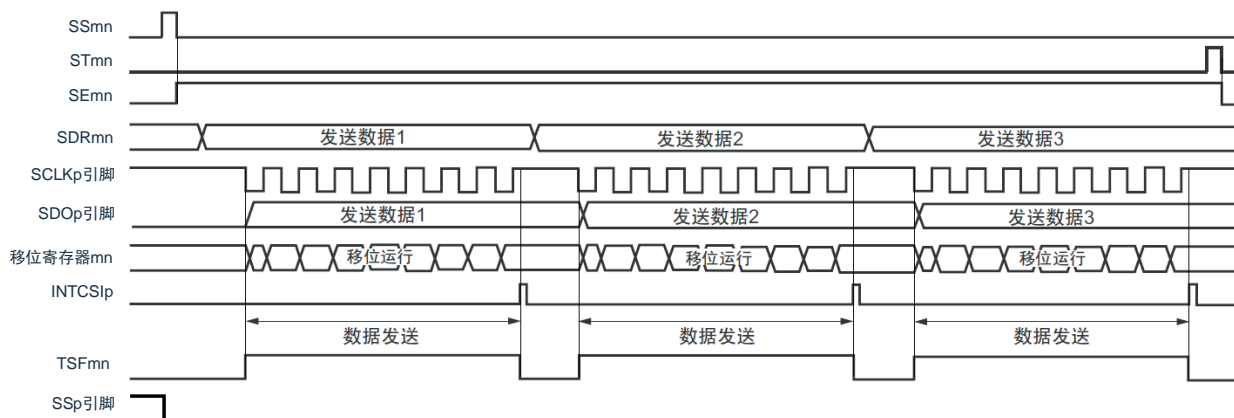


图17-75 重新开始从属发送的设定步骤



(3) 处理流程（单次发送模式）

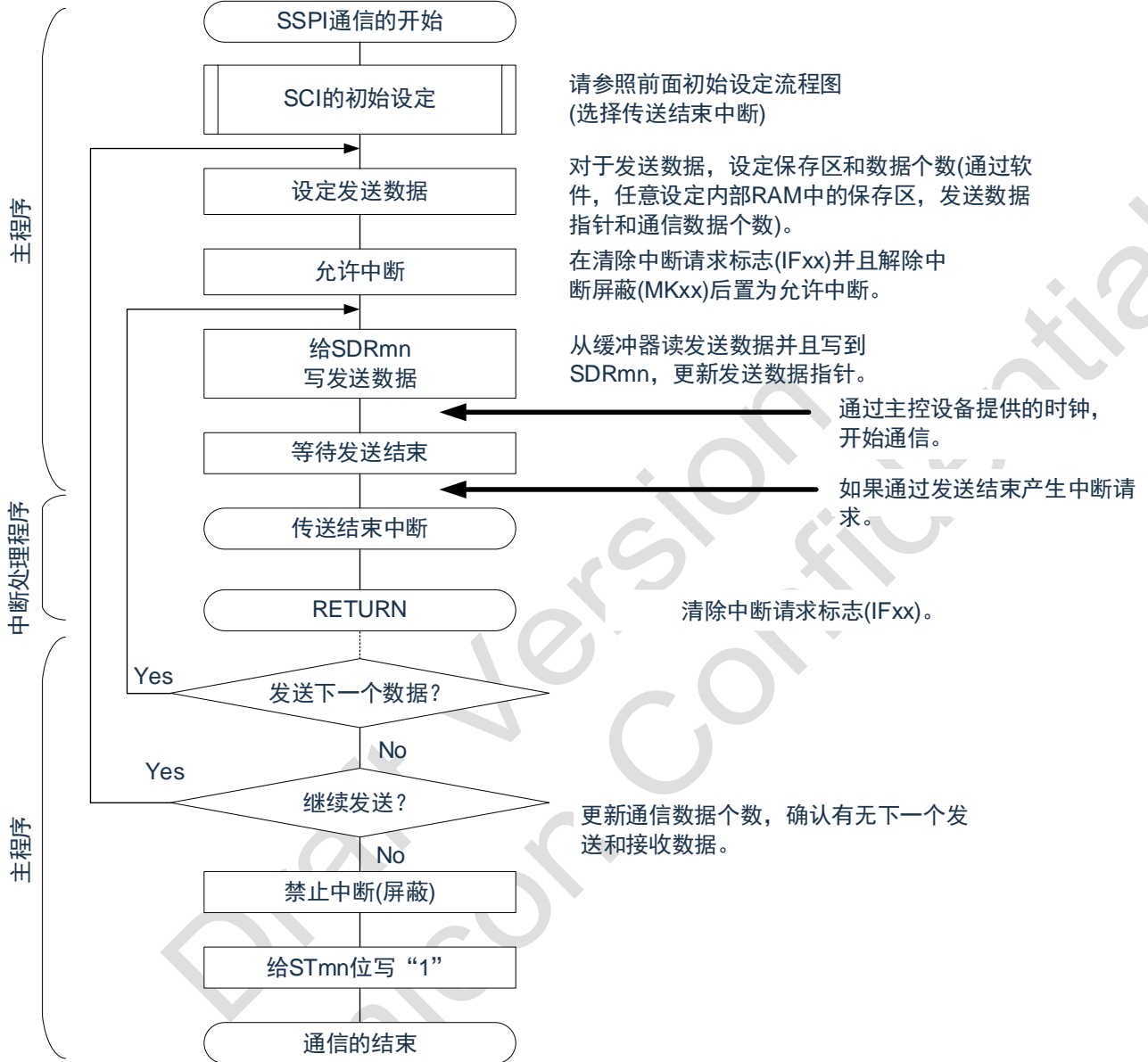
图17-76 从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0）n：通道号（n=0）p：SSPI号（p=00）

Draft Version  
Cmsemicon Confidential

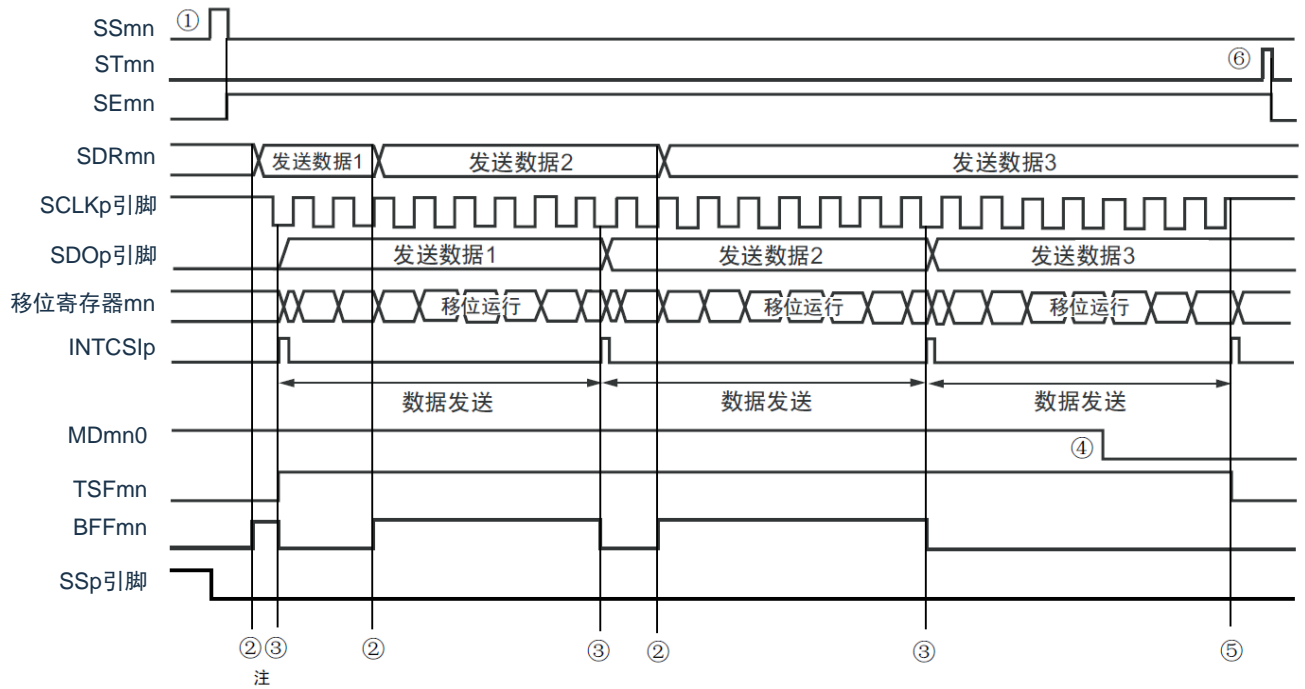
图17-77 从属发送（单次发送模式）的流程图



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17-78 从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

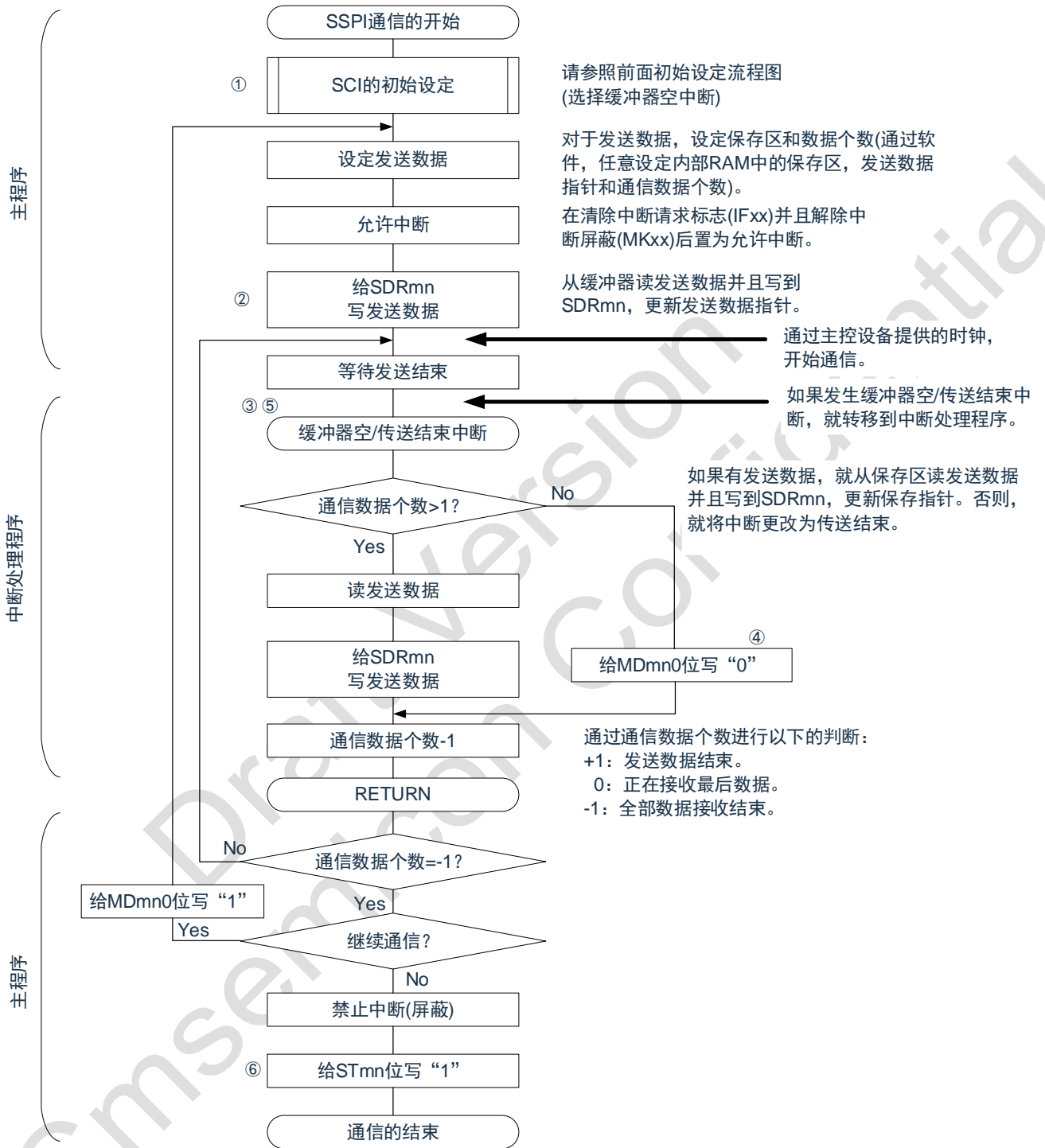


注：如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。

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17-79 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17-78 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)

## 17.6.2 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本产品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（禁止设定缓冲器空中断）。
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~16位
传送速率	$\text{Max}f_{\text{MCK}}/6[\text{Hz}]$ 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 $f_{\text{MCK}}/6[\text{Hz}]$ 。

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。

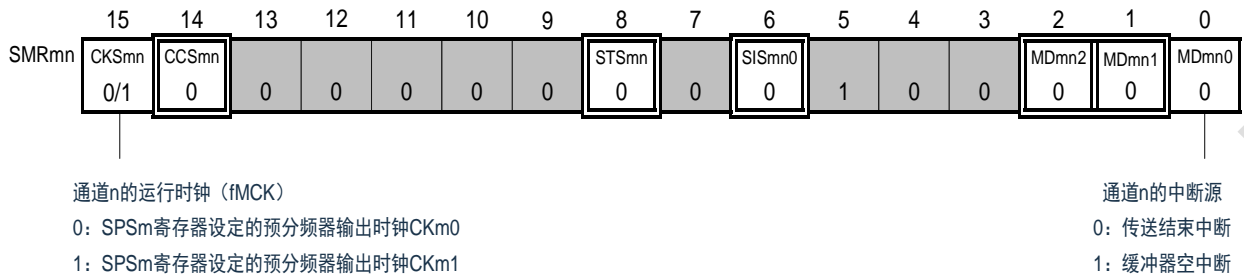
备注：1. $f_{\text{MCK}}$ ：对象通道的运行时钟频率

2.m：单元号（m=0） n：通道号（n=0）

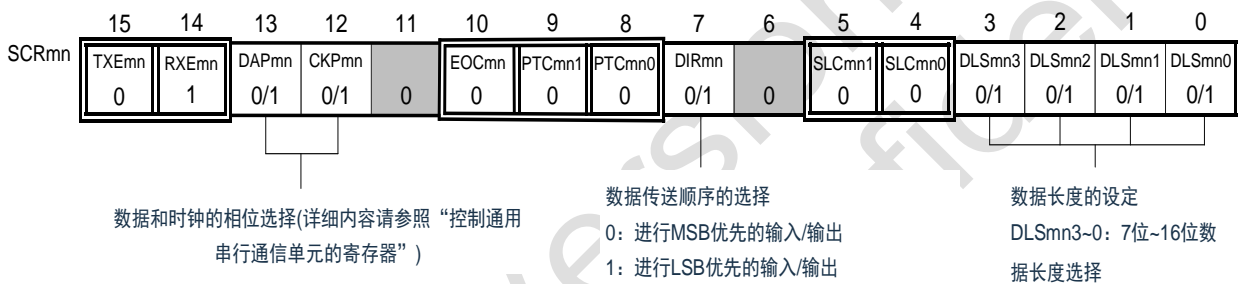
(1) 寄存器的设定

图17-80 从属选择输入功能 (SSPImn) 从属接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

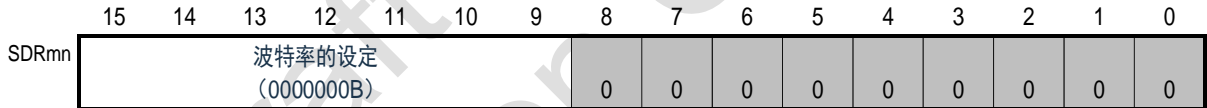


(b) 串行通信运行设定寄存器mn(SCRmn)

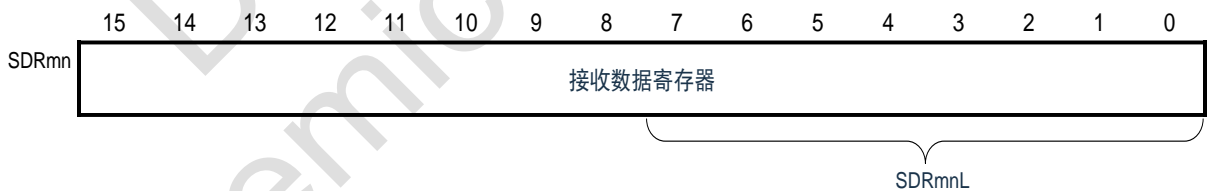


(c) 串行数据寄存器mn(SDRmn)

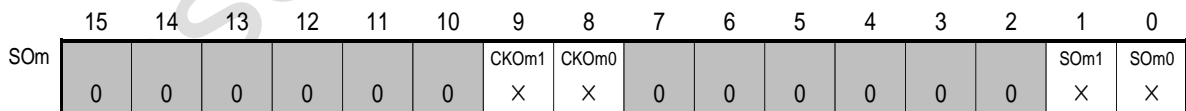
(1) 运行停止 (SEmn=0) 时



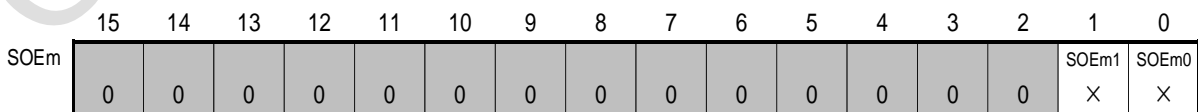
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(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。



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定。■: 不能设定(设定初始值)。

X: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图17-81 从属选择输入功能 (SSPImn) 从属接收时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器(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	0	0	SSm1 X	SSm0 0/1

(g) 从属选择功能启用寄存器(SSE)……这是SSPImn从属通道(单元m的通道n)的SSPImn引脚的控制。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIEm0 0/1

0: SSPImn引脚的输入值无效  
1: SSPImn引脚的输入值有效

备注: 1.m: 单元号 (m=0) n: 通道号 (n=0) p: SSPId (p=00)

 2.  : 在从属接收模式中为固定设定。  : 不能设定 (设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。



(2) 操作步骤

图17-82 从属接收的初始设定步骤

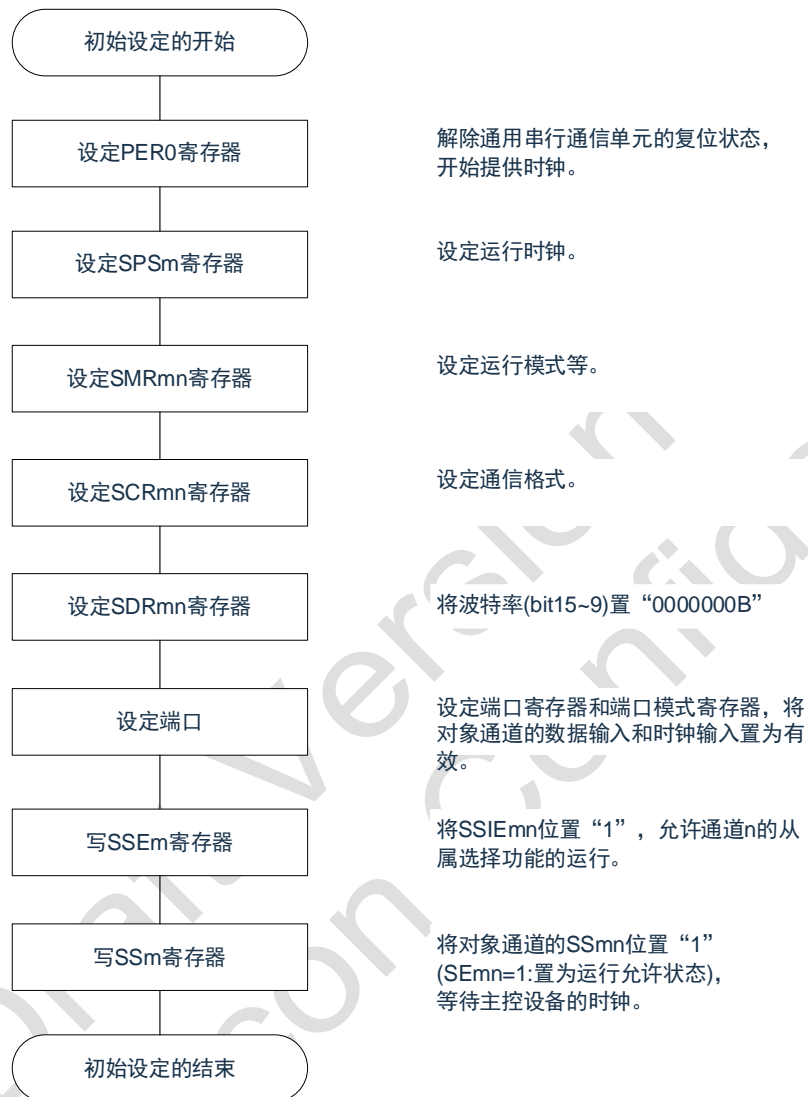
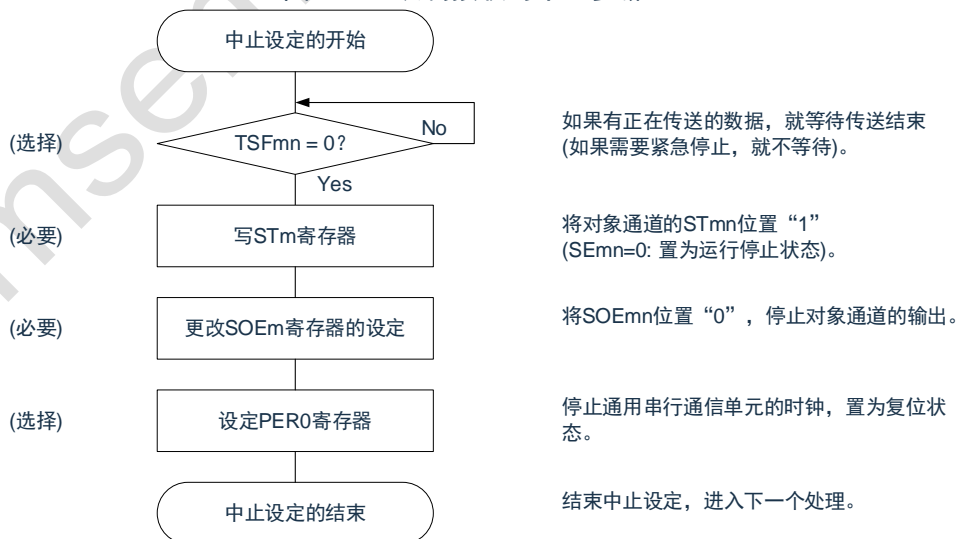


图17-83 从属接收的中止步骤



备注： m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

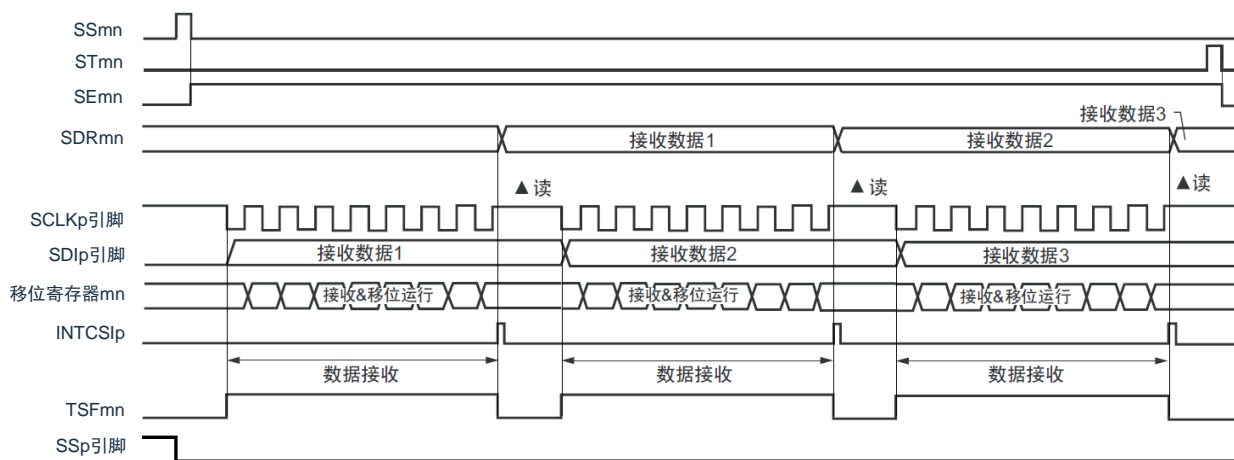
图17-84 重新开始从属接收的设定步骤



备注: m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

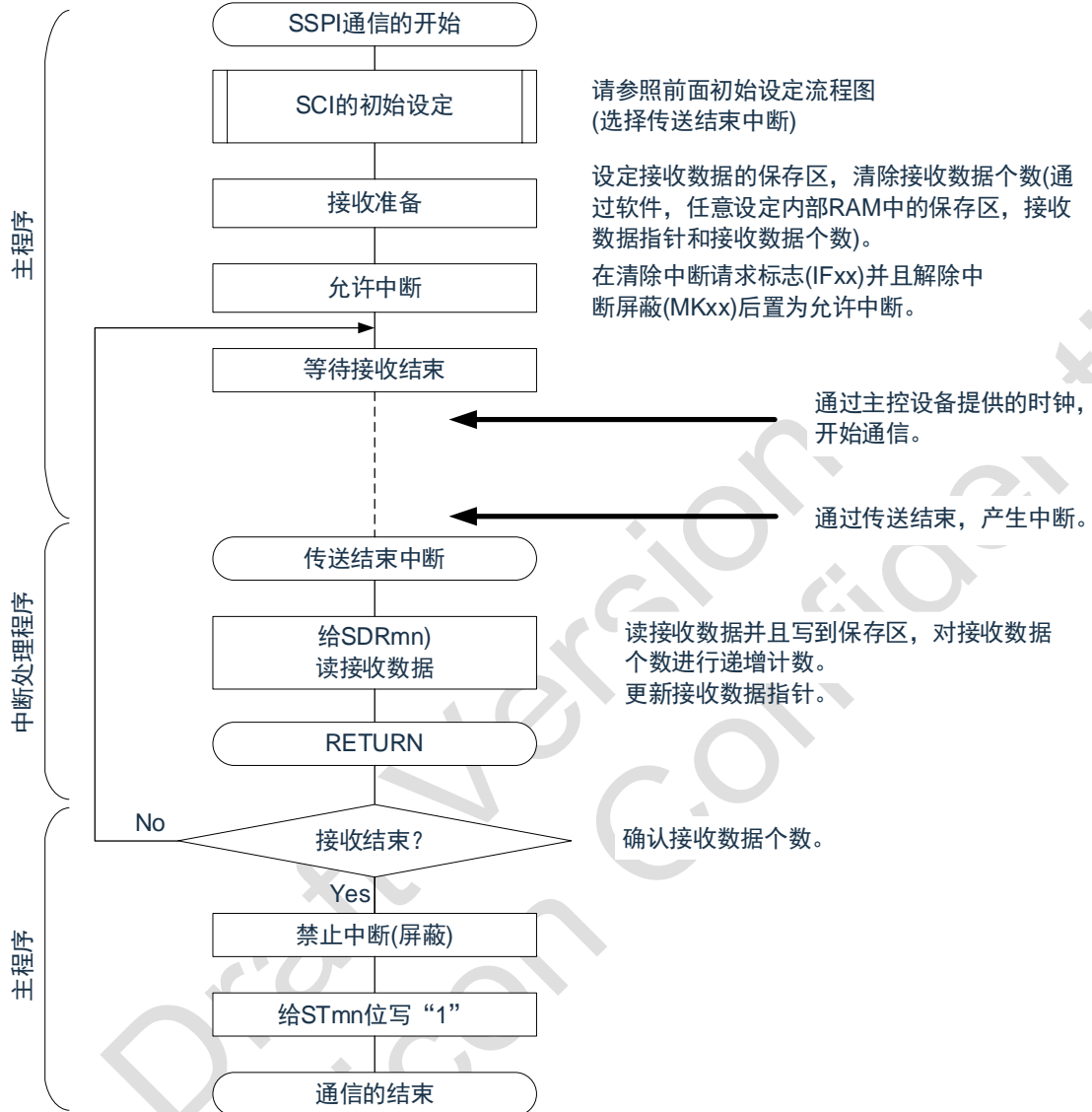
(3) 处理流程（单次接收模式）

图17-85 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17-86 从属接收（单次接收模式）的流程图



### 17.6.3 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本产品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、SDO00、SDI00、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	只有溢出错检测标志（OVFmn）。
传送数据长度	7位~16位
传送速率	$\text{Max.}f_{\text{MCK}}/6[\text{Hz}]$ 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SCLKmn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 $f_{\text{MCK}}/6[\text{Hz}]$ 。
  - 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。

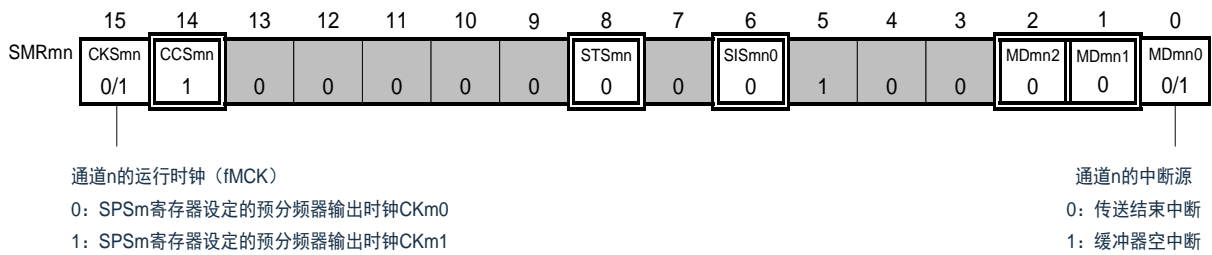
备注：1. $f_{\text{MCK}}$ ：对象通道的运行时钟频率

2.m：单元号（m=0） n：通道号（n=0）

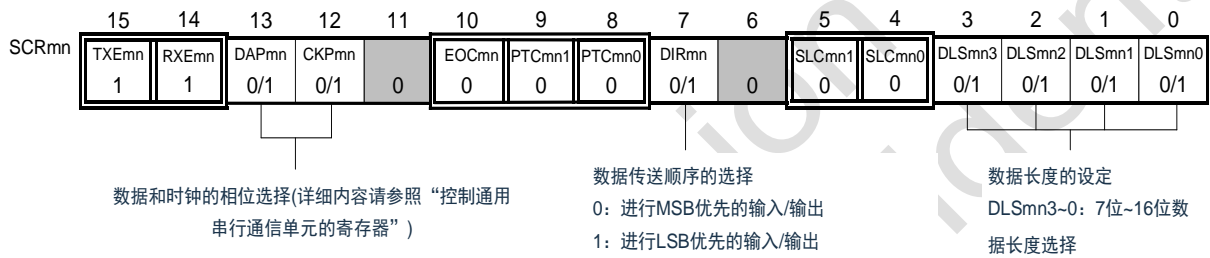
(1) 寄存器的设定

图17-87 从属选择输入功能 (SSPI<sub>mn</sub>) 从属发送和接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMR<sub>mn</sub>)

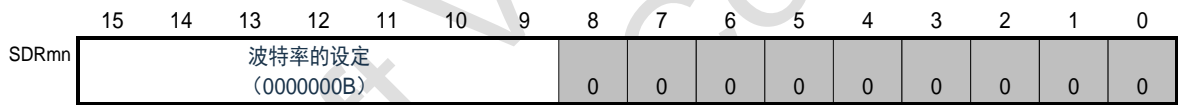


(b) 串行通信运行设定寄存器mn(SCR<sub>mn</sub>)

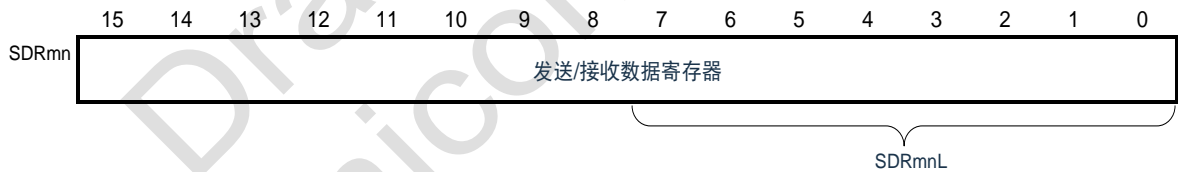


(c) 串行数据寄存器mn(SDR<sub>mn</sub>)

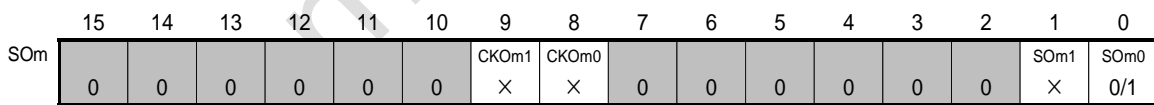
(1) 运行停止 (SEmn=0) 时



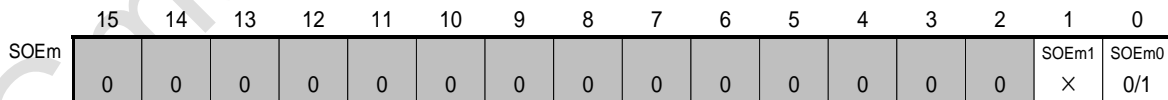
(2) 运行期间 (SEmn=1) (低8位: SDR<sub>mnL</sub>)



(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SDR<sub>mn</sub>寄存器设定发送数据。

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定。  : 不能设定 (设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

**图17-87 从属选择输入功能 (SSPImn) 从属发送和接收时的寄存器设定内容例子(2/2)**

(f) 串行通道开始寄存器(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	0	0	SSm1 X	SSm0 0/1

(g) 从属选择功能启用寄存器 (SSE)……这是SSPImn从属通道(单元m的通道n)的SSImn引脚的控制。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIm0 0/1

0: SSIm0引脚的输入值无效  
1: SSIm0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SDRmn寄存器设定发送数据。

备注: 1.m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

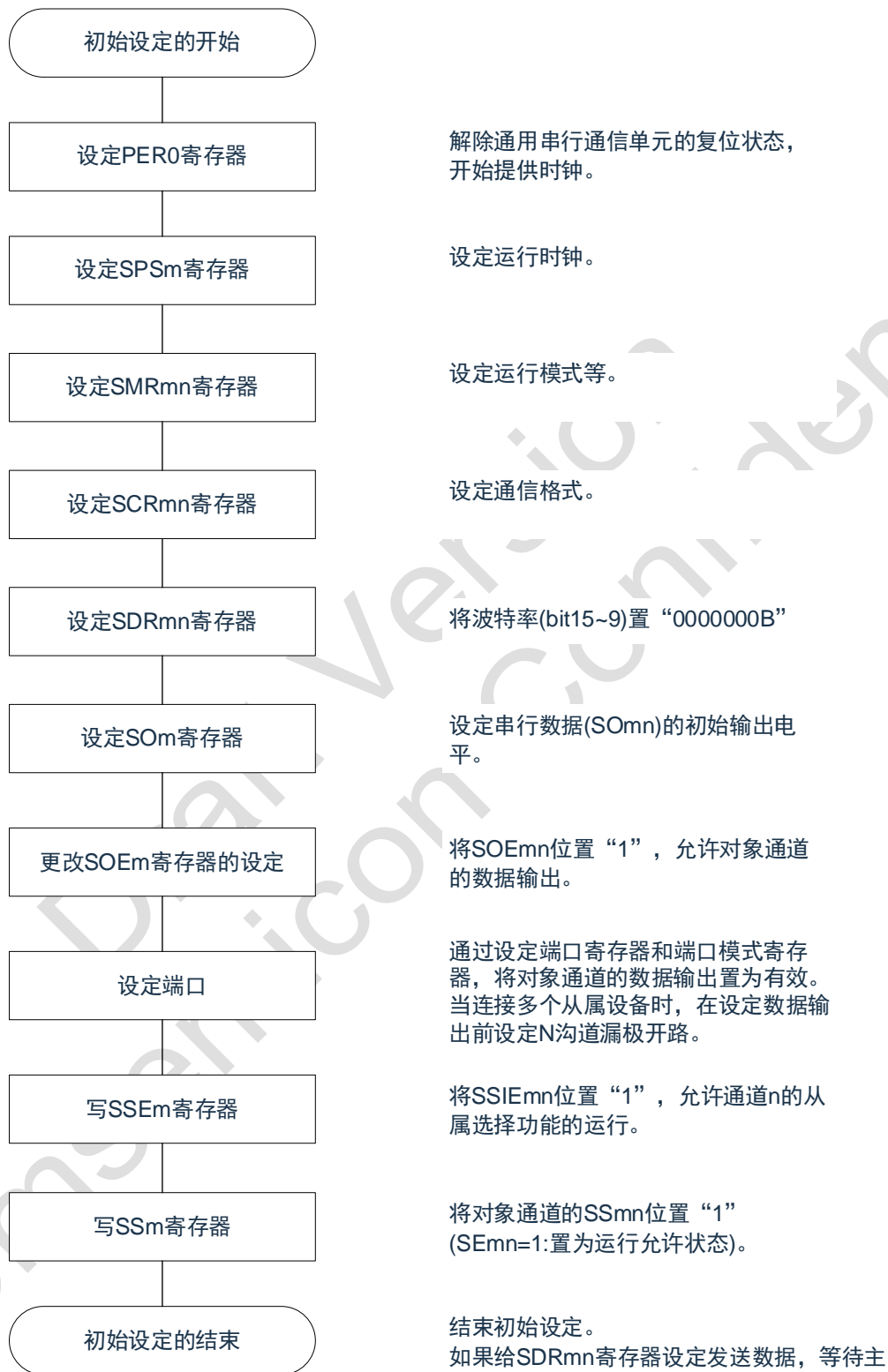
2. : 在从属接收模式中为固定设定。 : 不能设定(设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-88 从属发送和接收的初始设定步骤

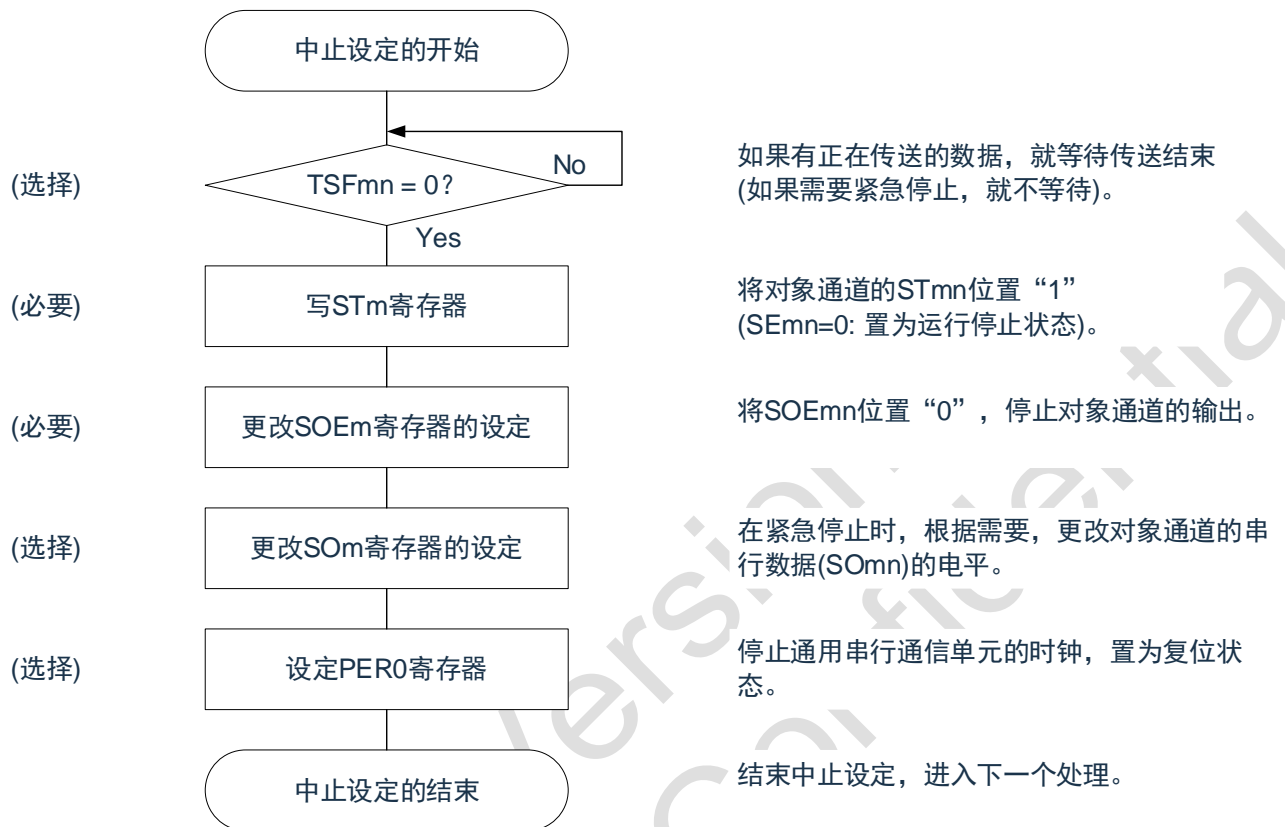


注意：在主控设备开始输出时钟前，必须给SDRmn寄存器设定发送数据。

备注：m：单元号 (m=0) n：通道号 (n=0) p：SSPI号 (p=00)



图17-89 从属发送和接收的中止步骤



备注: 1.m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

图17-90 重新开始从属发送和接收的设定步骤

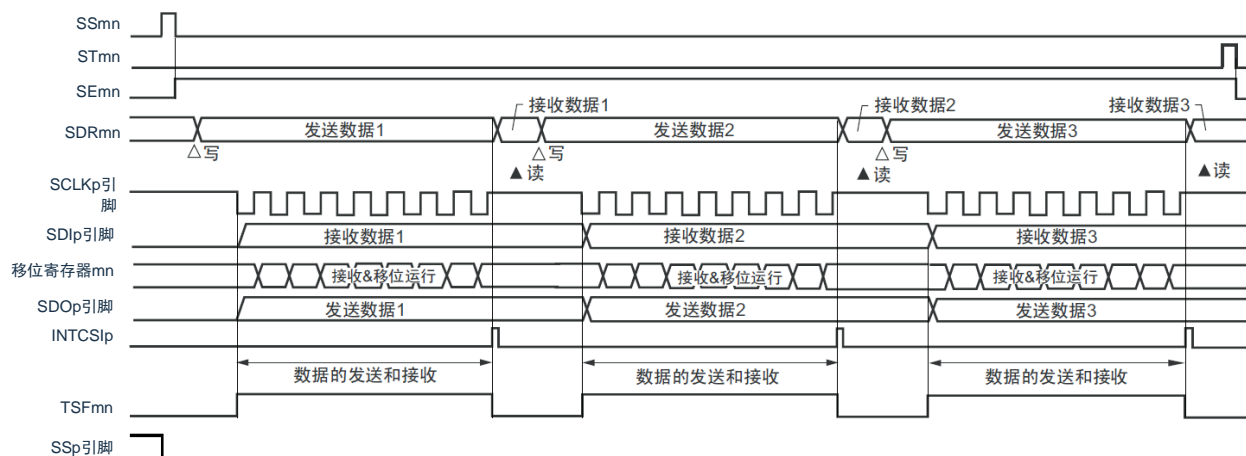


注意: 1.在主控设备开始输出时钟前, 必须给SDRmn寄存器设定发送数据。

2.如果在中止设定中改写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定而不是进行重新开始设定。

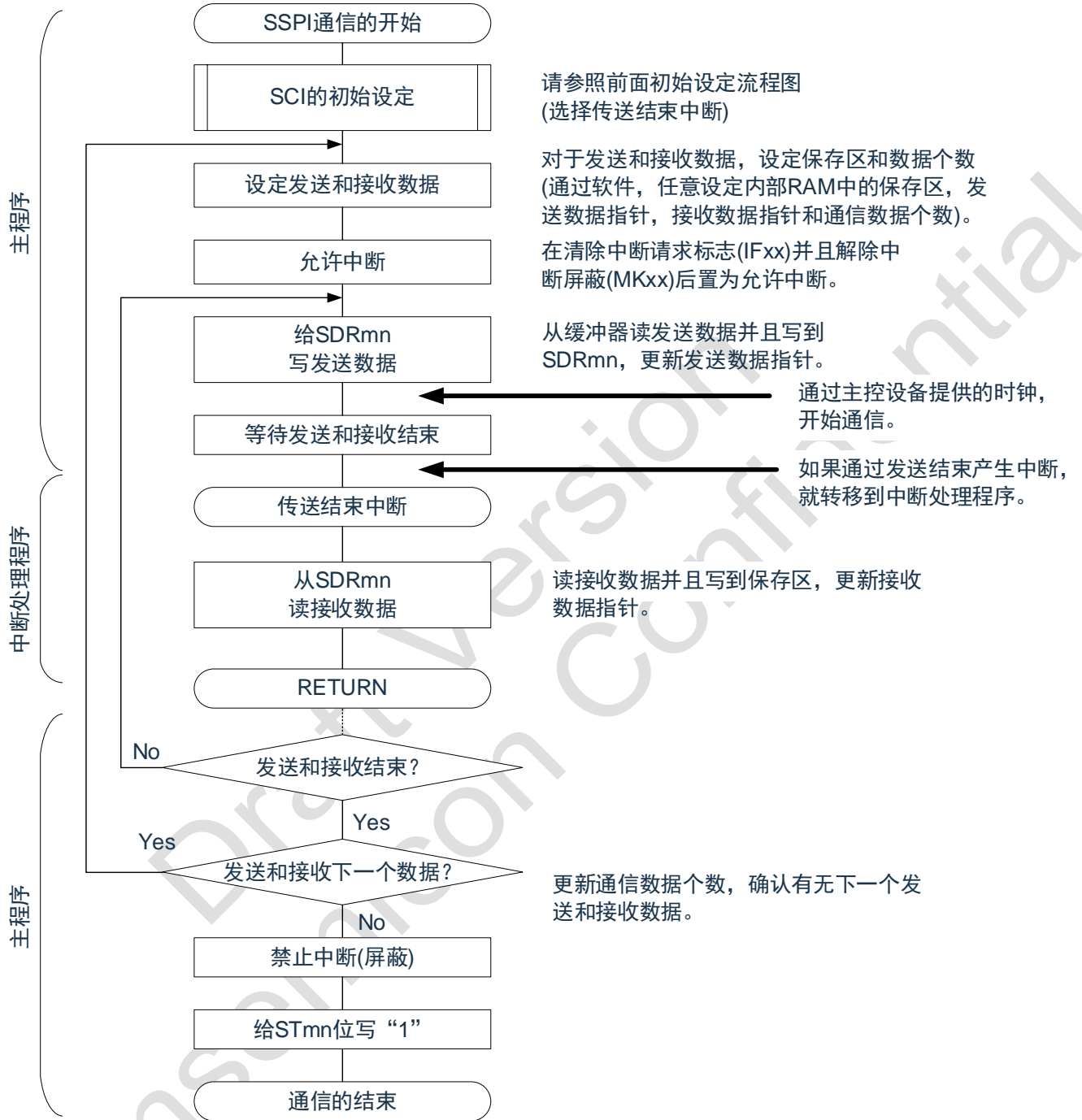
(3) 处理流程（单次发送和接收模式）

图17-91 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17-92 从属发送和接收（单次发送和接收模式）的流程图

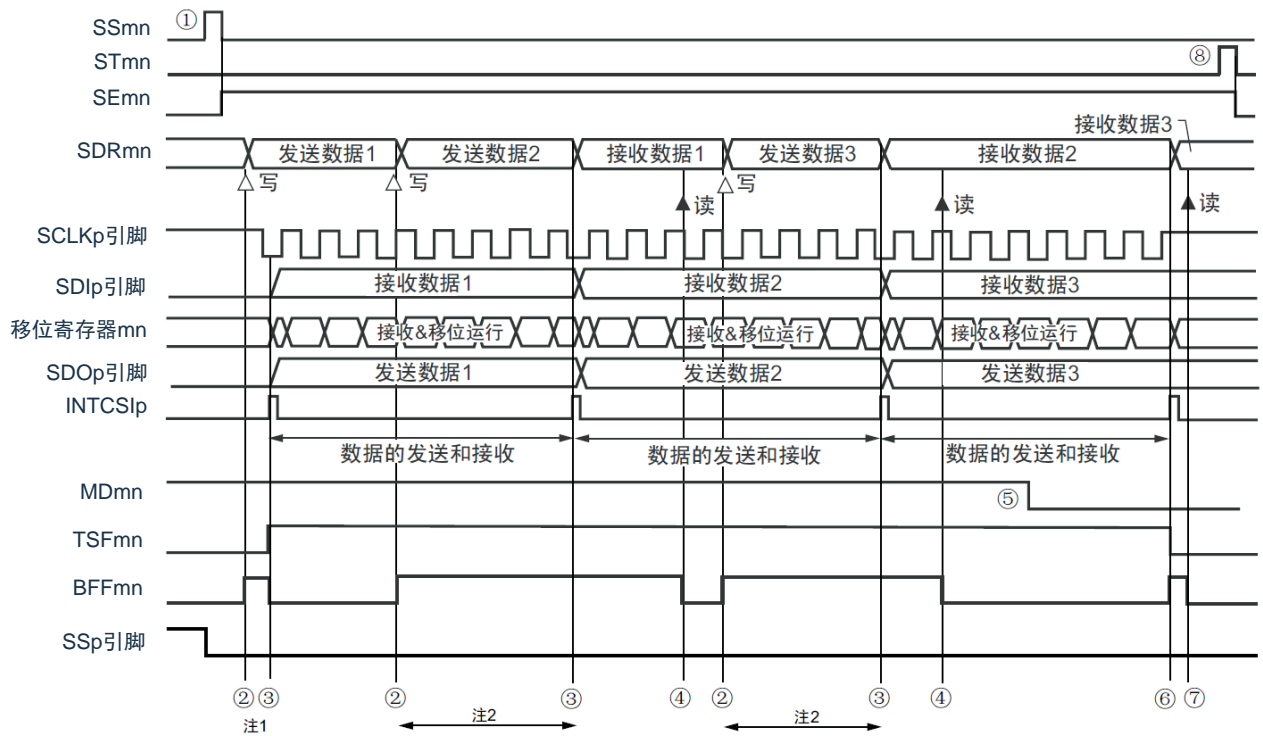


注意：在主控设备开始输出时钟前，必须给SDRmn寄存器设定发送数据。

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17-93 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）



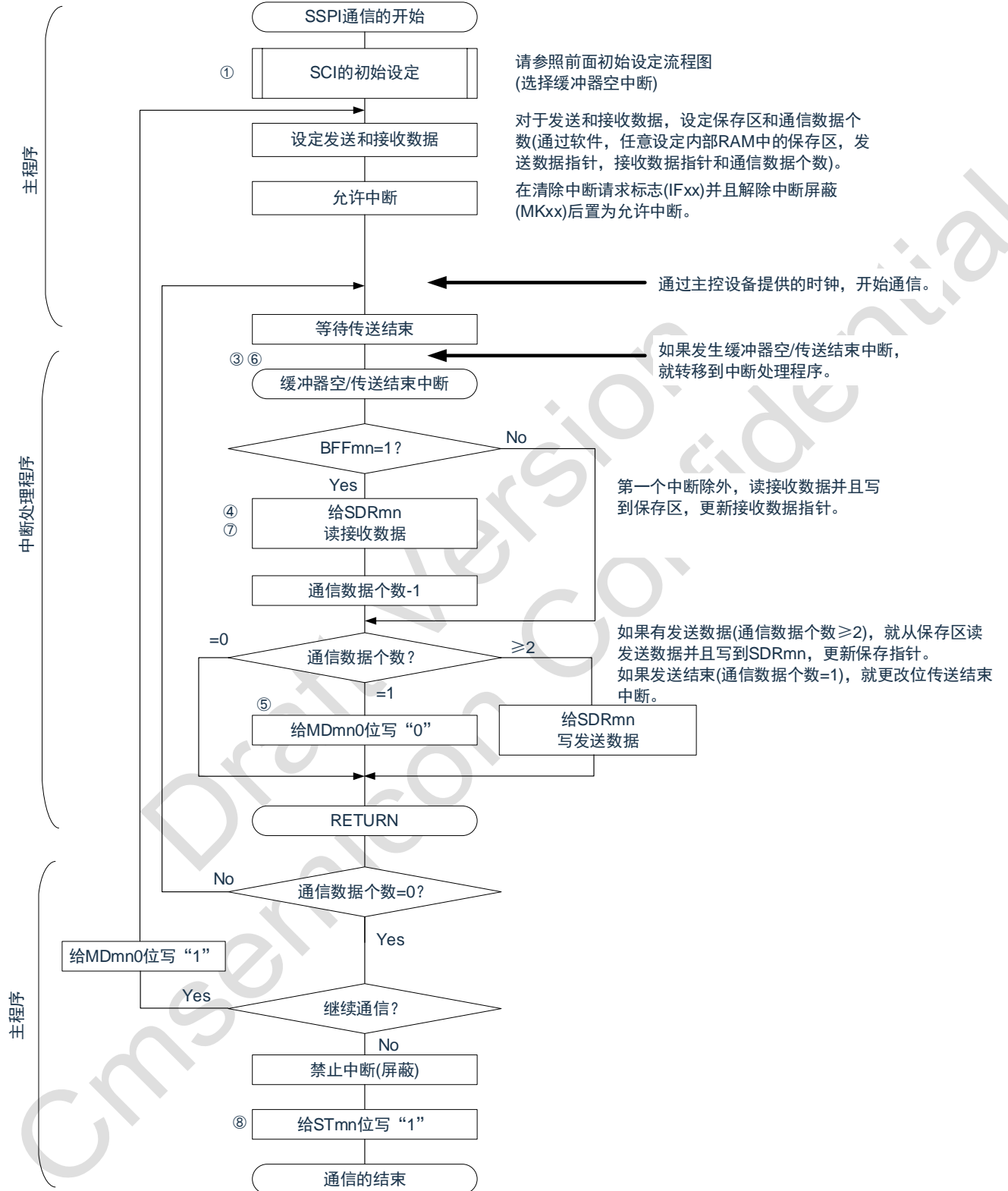
注：1.如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。  
 2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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17-94 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17-94 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SDRmn寄存器设定发送数据。

备注：1.图中的①~⑧对应“图17-93 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)

## 17.6.4 传送时钟频率的计算

从属选择输入功能（SSPImn）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 $f_{\text{MCK}}/6$ 。

表17-3 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 ( $f_{\text{MCK}}$ ) 注		
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$f_{\text{CLK}}=32\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	0	$f_{\text{CLK}}$	32MHz
	X	X	X	X	0	0	0	1	1	$f_{\text{CLK}}/2$	16MHz
	X	X	X	X	0	0	1	0	0	$f_{\text{CLK}}/2^2$	8MHz
	X	X	X	X	0	0	1	1	1	$f_{\text{CLK}}/2^3$	4MHz
	X	X	X	X	0	1	0	0	0	$f_{\text{CLK}}/2^4$	2MHz
	X	X	X	X	0	1	0	1	1	$f_{\text{CLK}}/2^5$	1MHz
	X	X	X	X	0	1	1	1	0	$f_{\text{CLK}}/2^6$	500kHz
	X	X	X	X	0	1	1	1	1	$f_{\text{CLK}}/2^7$	250kHz
	X	X	X	X	1	0	0	0	0	$f_{\text{CLK}}/2^8$	125kHz
	X	X	X	X	1	0	0	1	1	$f_{\text{CLK}}/2^9$	62.5kHz
	X	X	X	X	1	0	1	0	0	$f_{\text{CLK}}/2^{10}$	31.25kHz
	X	X	X	X	1	0	1	1	1	$f_{\text{CLK}}/2^{11}$	15.63kHz
	X	X	X	X	1	1	0	0	0	$f_{\text{CLK}}/2^{12}$	7.81kHz
	X	X	X	X	1	1	0	1	1	$f_{\text{CLK}}/2^{13}$	3.91kHz
	X	X	X	X	1	1	1	1	0	$f_{\text{CLK}}/2^{14}$	1.95kHz
X	X	X	X	1	1	1	1	1	$f_{\text{CLK}}/2^{15}$	977Hz	

注：要更改被选择为 $f_{\text{CLK}}$ 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）

## 17.6.5 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17-95所示。

图17-95 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 (SIRmn) 写“1”。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SIRmn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)



## 17.7 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。

### [数据的发送和接收]

- 7位、8位、9位或者16位的数据长度<sup>注</sup>
- MSB/LSB优先的选择
- 发送和接收数据的电平设定 (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使用SCI0的通道0和通道1。

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m的通道0和通道1使用UART0时, 不能使用SSPI00和IIC00。

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2种通信运行:

- UART发送 (参照17.7.1)
- UART接收 (参照17.7.2)

## 17.7.1 UART发送

UART发送是本产品将数据异步发送到其他设备。

UART使用2个通道中的偶数通道用于UART发送。

UART	UART0	UART2
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	TxD0	TxD2
中断	INTST0	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、9 位或者 16 位	
传送速率	Max. $f_{MCK}/6$ [bps] ( $SDRmn[15:9] \geq 3$ )、Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]注	
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：	
	•无奇偶校验位。	
	•附加零校验。	
	•附加偶校验。	
停止位	可选择以下内容：	
	•附加 1 位。	
	•附加 2 位。	
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：1. $f_{MCK}$ ：对象通道的运行时钟频率

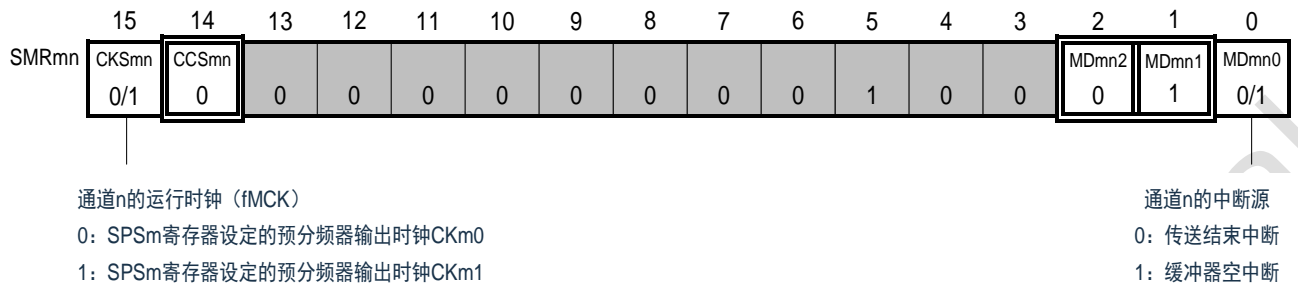
$f_{CLK}$ ：系统时钟频率

2.m：单元号（m=0、1） n：通道号（n=0）

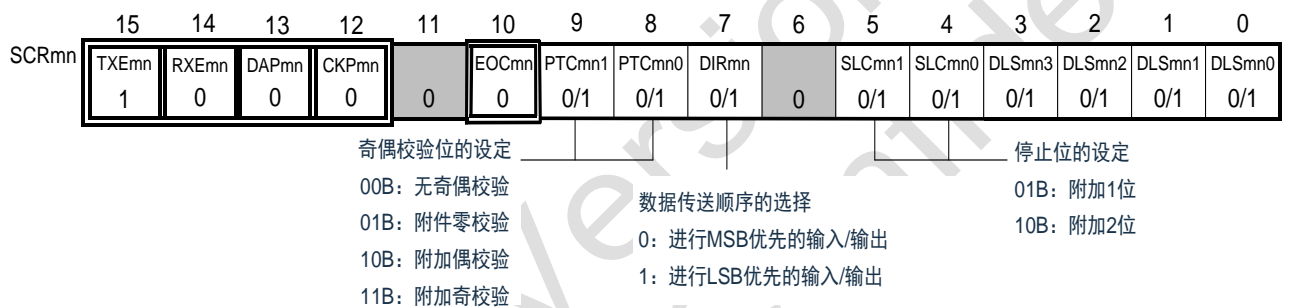
(1) 寄存器的设定

图17-96 UART (UART0、UART2)的UART发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

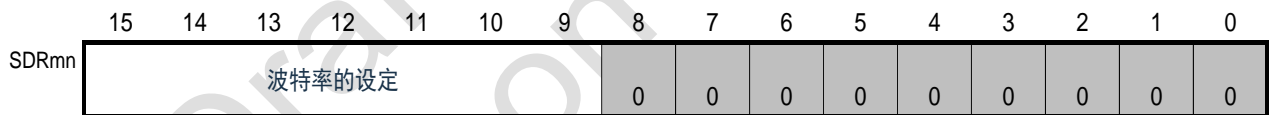


(b) 串行通信运行设定寄存器mn(SCRmn)

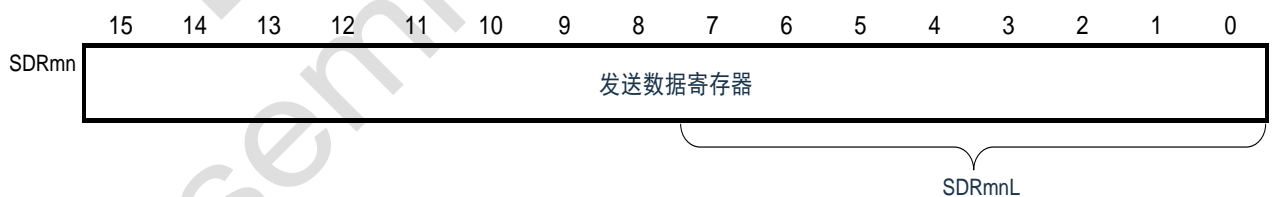


(c) 串行数据寄存器mn(SDRmn)

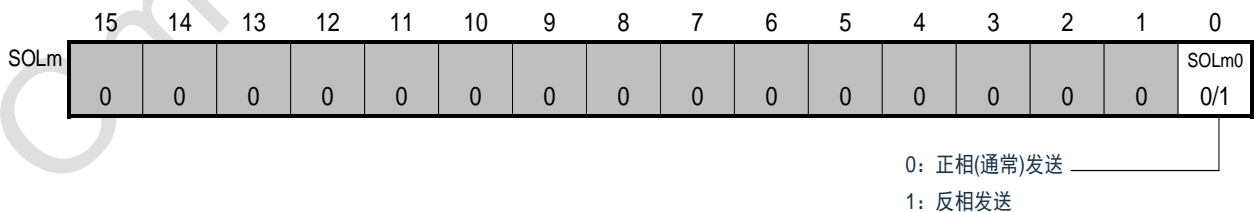
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(d) 串行输出电平寄存器m(SOLm) ...只设定对象通道的位。



备注: 1.m: 单元号 (m=0、1) n: 通道号 (n=0) q: UART号 (q=0、2)

2. □: 在UART发送模式中为固定设定。 ■: 不能设定(设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图17-96 UART (UART0~UART2) 的UART发送时的寄存器设定内容例子(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	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOm1	SOm0
							×	×							×	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	0	0	SOEm1	SOEm0
															×	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	0	0	SSm1	SSm0
															×	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) q: UART号 (q=0、2)

2.  : 在UART发送模式中为固定设定。  : 不能设定 (设定初始值)。

×: 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，设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-97 UART发送的初始设定步骤

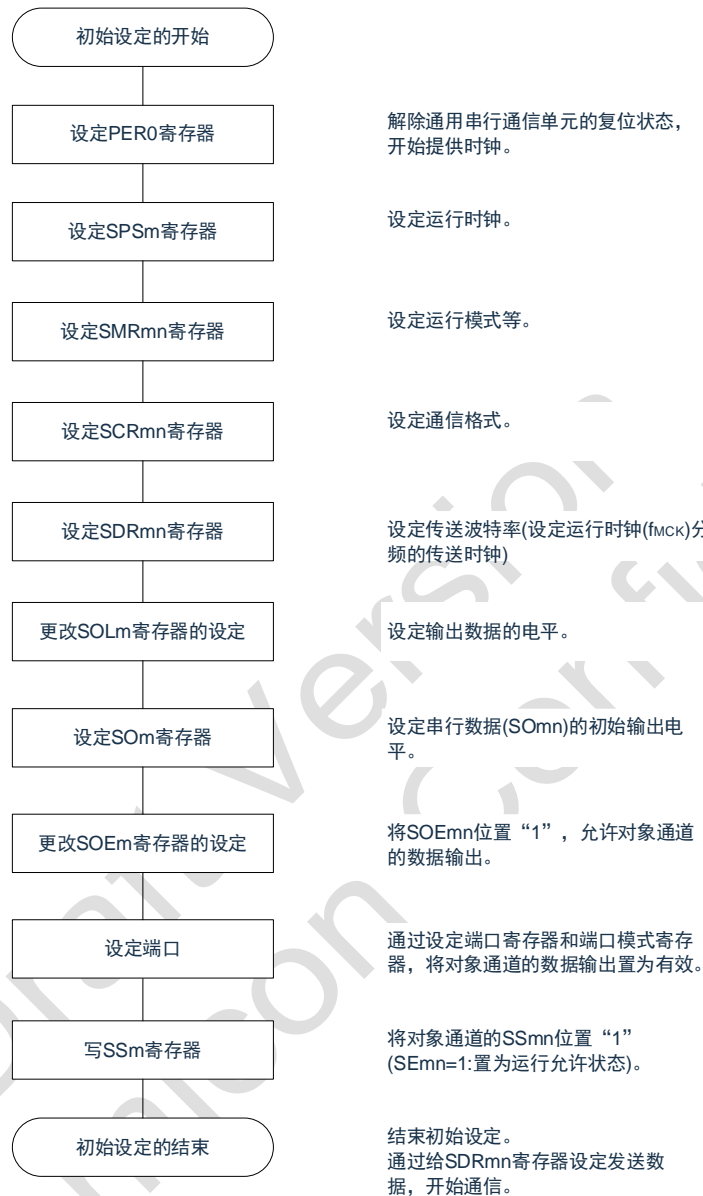


图17-98 UART发送的中止步骤

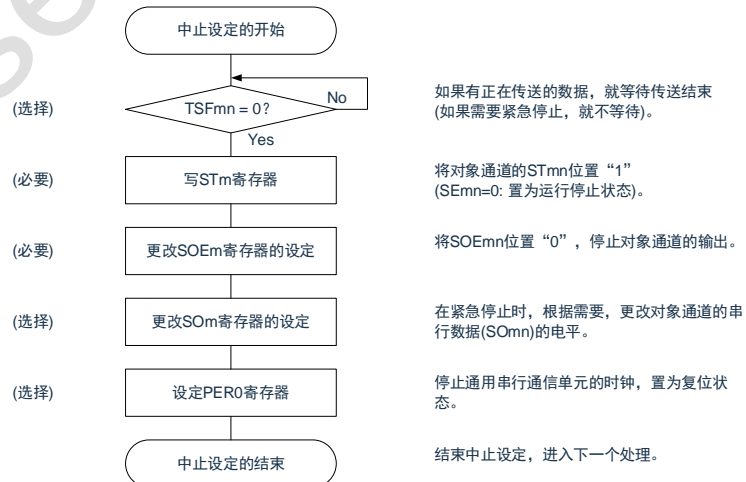


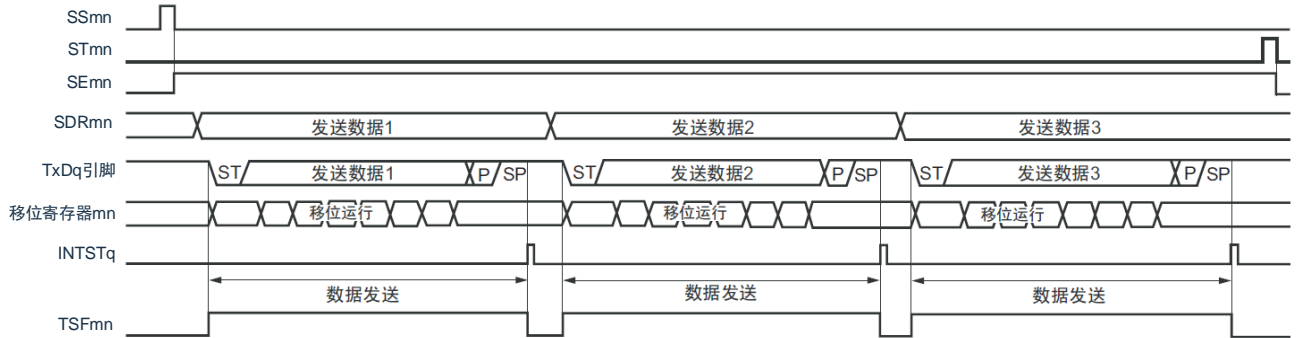
图17-99 重新开始UART发送的设定步骤



备注：如果在中止设定中改写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定而不是进行重新开始设定。

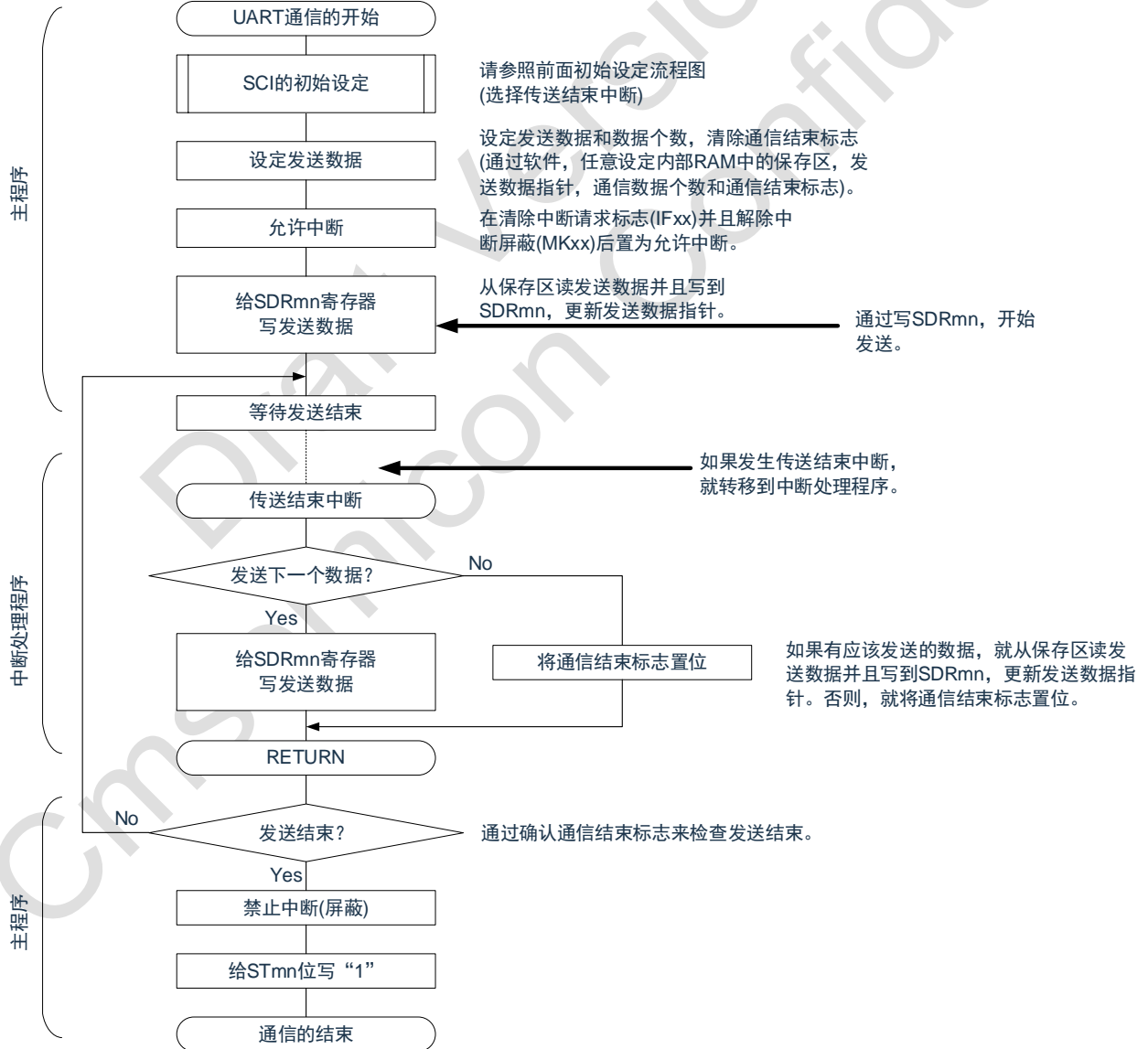
(3) 处理流程（单次发送模式）

图17-100 UART发送（单次发送模式）的时序图



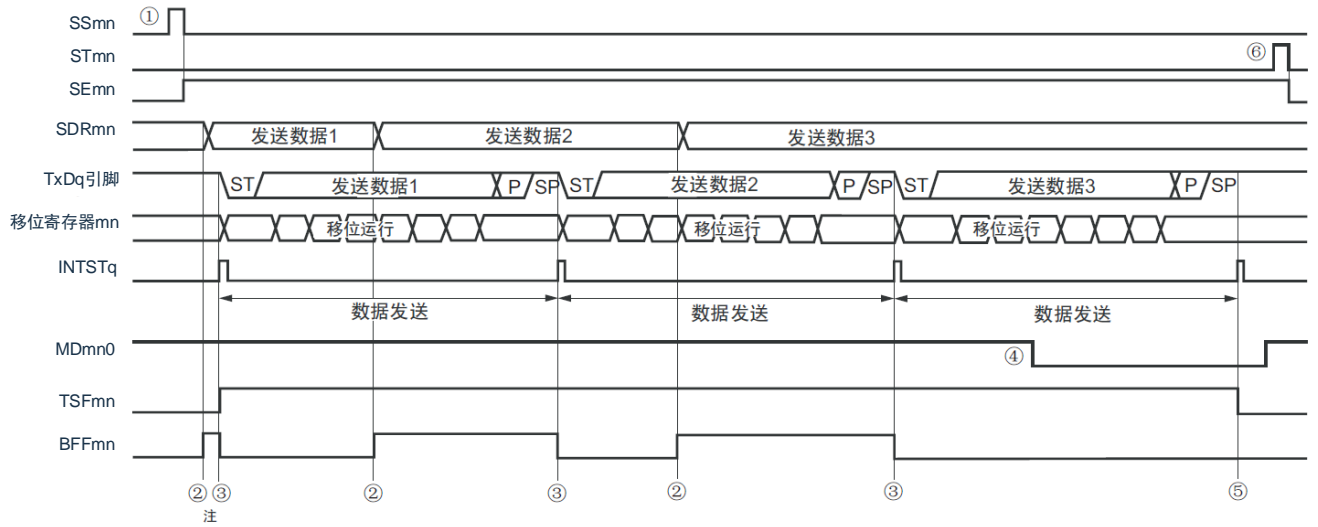
备注： m：单元号（m=0、1） n：通道号（n=0） q：UART号（q=0、2）

图17-101 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17-102 UART发送（连续发送模式）的时序图



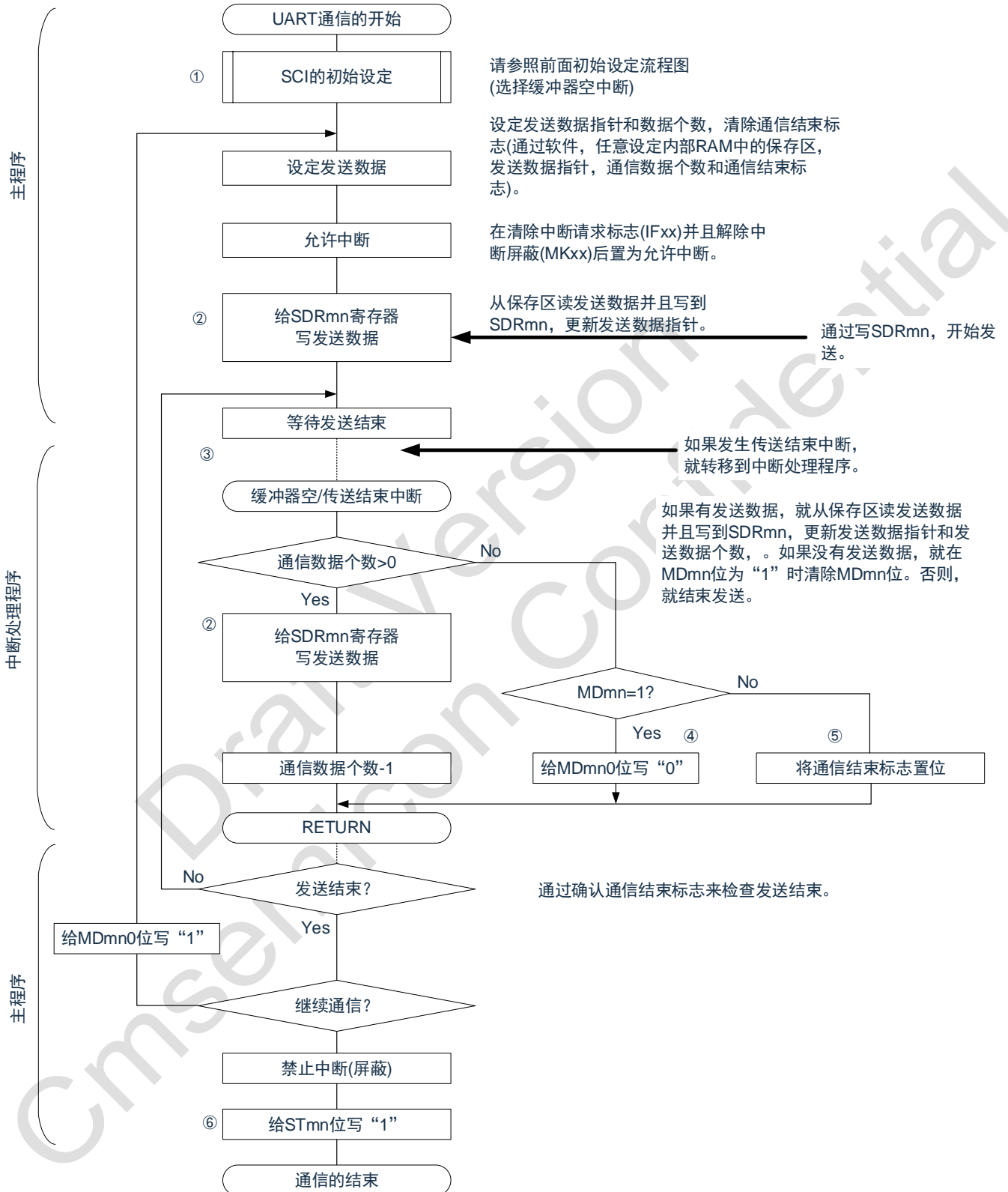
注：如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。

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） q：UART号（q=0、2）



图17-103 UART发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图17-102 UART发送（连续发送模式）的时序图”中的①~⑥。

## 17.7.2 UART接收

UART接收是本产品异步接收其他设备发送的数据。

UART使用2个通道中的奇数通道用于UART接收。但是，需要设定奇数通道和偶数通道的SMR寄存器。

UART	UART0	UART2
对象通道	SCI0 的通道 1	SCI1 的通道 1
使用的引脚	RxD0	RxD2
中断	INTSR0	INTSR2
	只限于传送结束中断（禁止设定缓冲器空中断）。	
错误检测标志	• 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、9 位或者 16 位	
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：	
	• 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：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

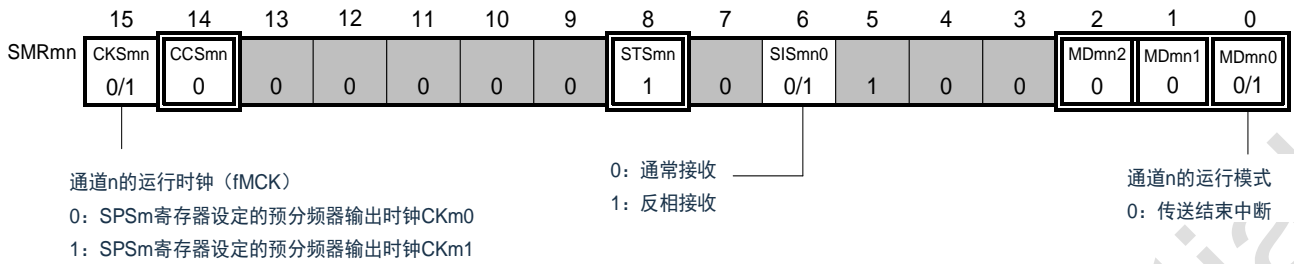
$f_{CLK}$ ：系统时钟频率

2. m：单元号（m=0、1） n：通道号（n=1）

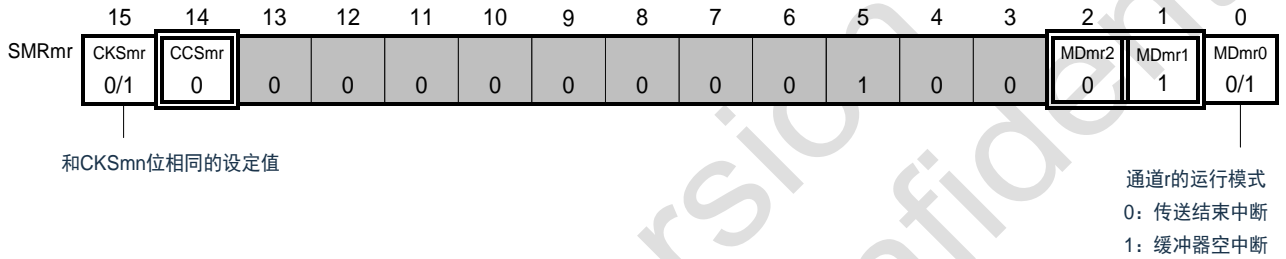
(1) 寄存器的设定

图17-104 UART (UART0、UART2) 的UART接收时的寄存器设定内容例子(1/2)

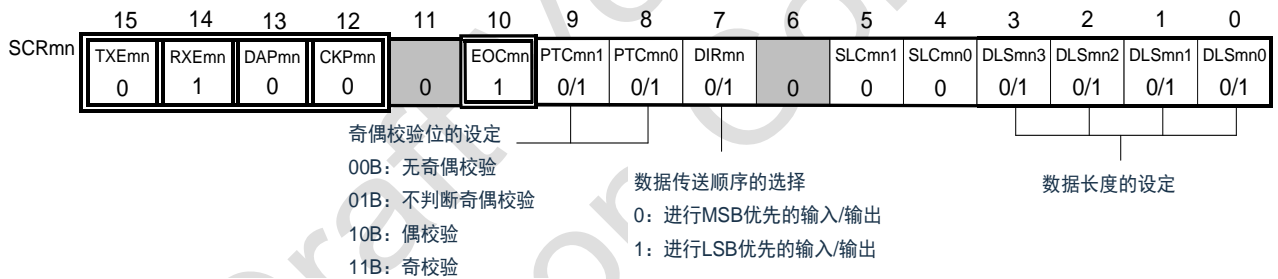
(a) 串行模式寄存器mn(SMRmn)



(b) 串行模式寄存器mn(SMRmr)

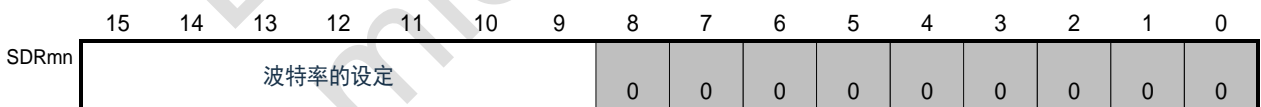


(c) 串行通信运行设定寄存器mn(SCRmn)

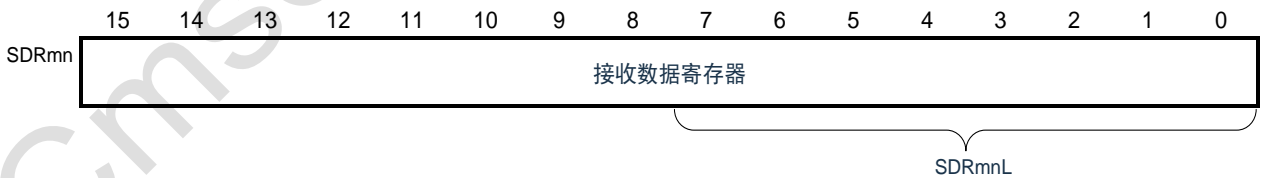


(d) 串行数据寄存器mn(SDRmn)

(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



注意: 在UART接收时, 还必须设定与通道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)

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定。  : 不能设定 (设定初始值)。

×: 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, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图17-104 UART (UART0~UART2) 的UART接收时的寄存器设定内容例子(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	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOm1	SOm0
							×	×							×	×

(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	0	0	SOEm1	SOEm0
															×	×

(g) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。

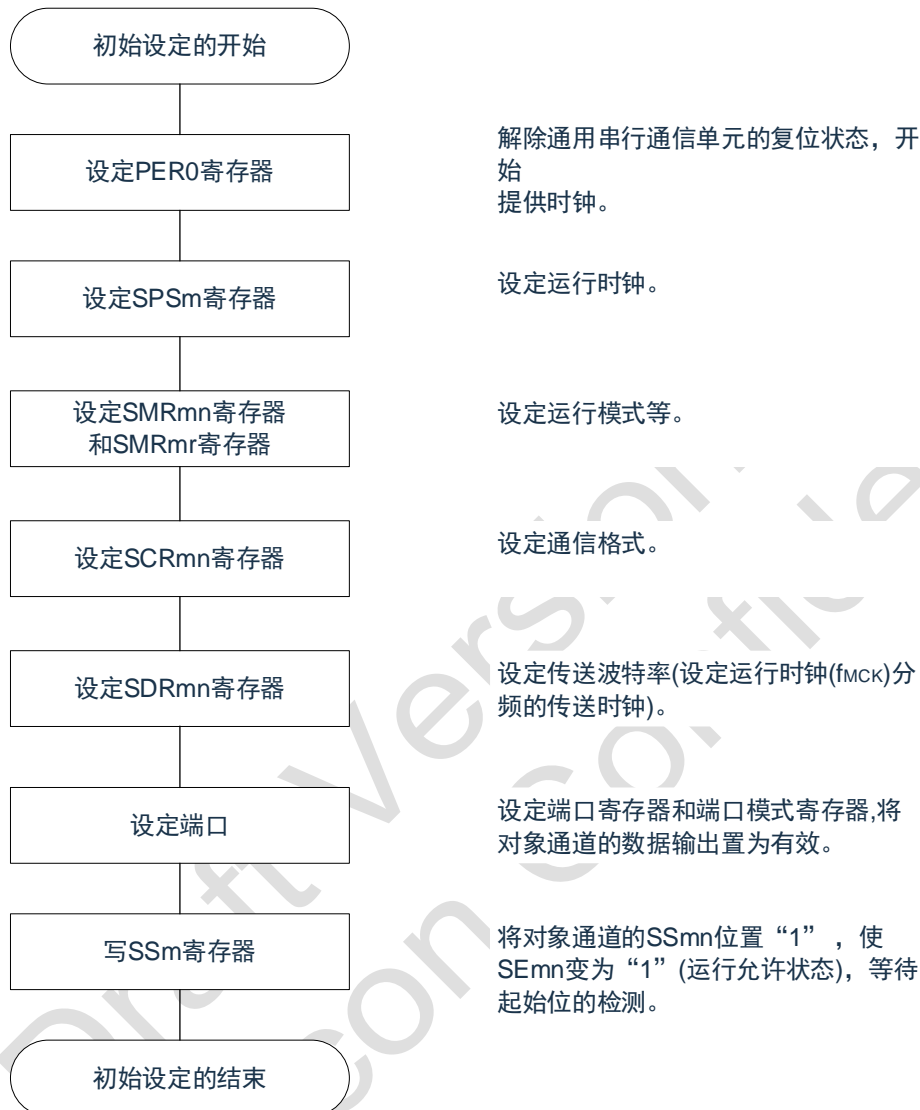
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1	SSm0
															0/1	×

备注：1.m: 单元号 (m=0、1)

2.  : 在UART接收模式中为固定设定。  : 不能设定 (设定初始值)。  
 × : 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, 设定初始值)。  
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图17-105 UART接收的初始设定步骤



注意：必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个f<sub>MCK</sub>时钟，然后将SSmn位置“1”。

图17-106 UART接收的中止步骤

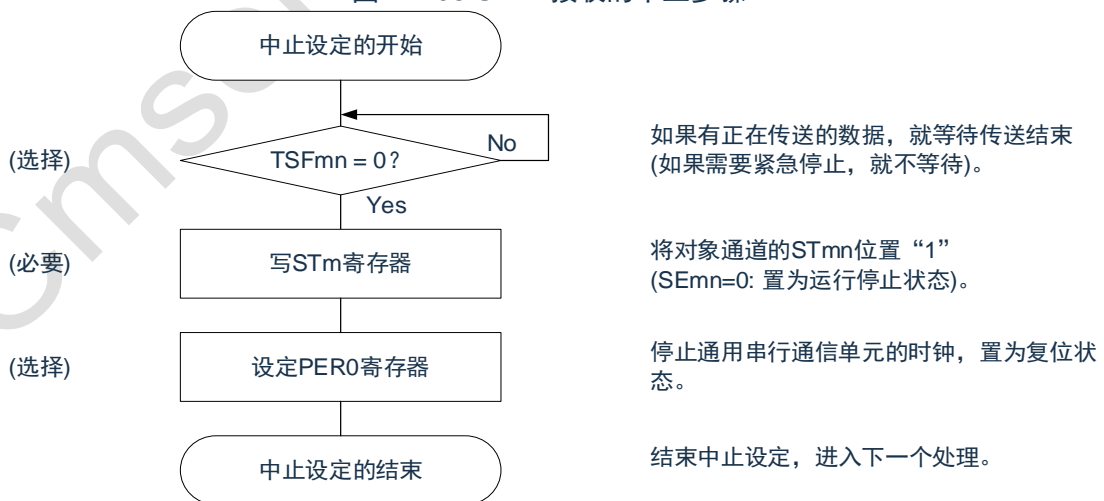
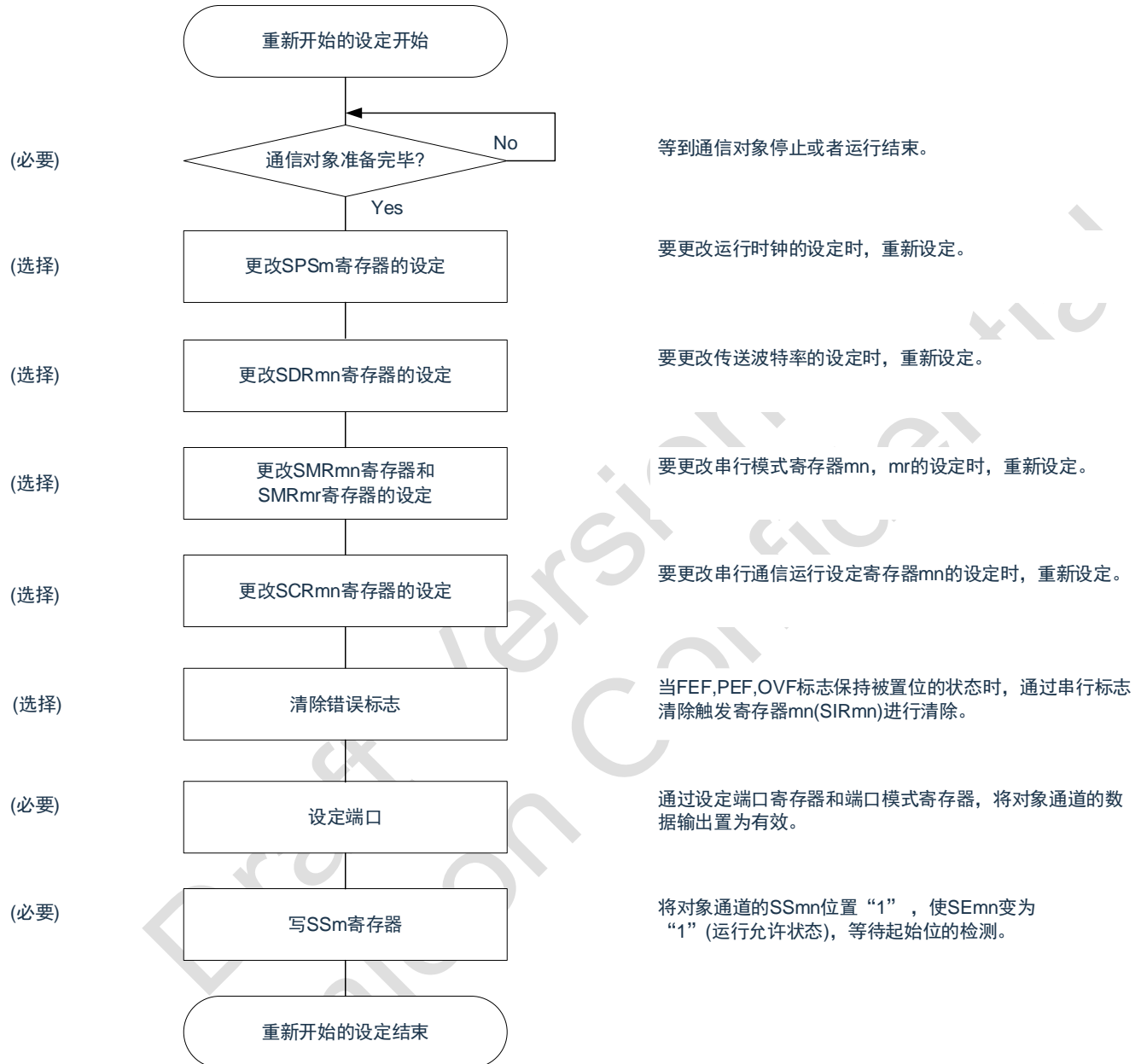


图17-107 重新开始UART接收的设定步骤

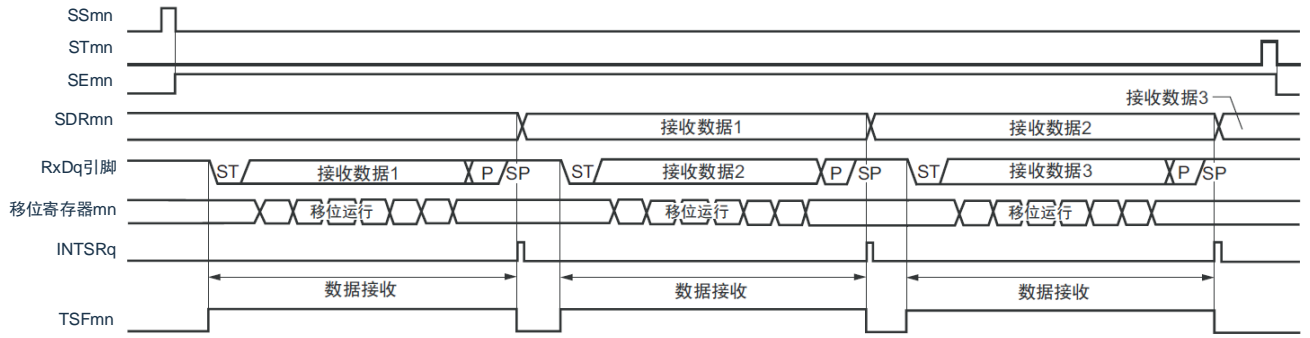


注意: 必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个 $f_{MCK}$ 时钟, 然后将SSmn位置“1”。

备注: 如果在中止设定中改写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定而不是进行重新开始设定。

(3) 处理流程

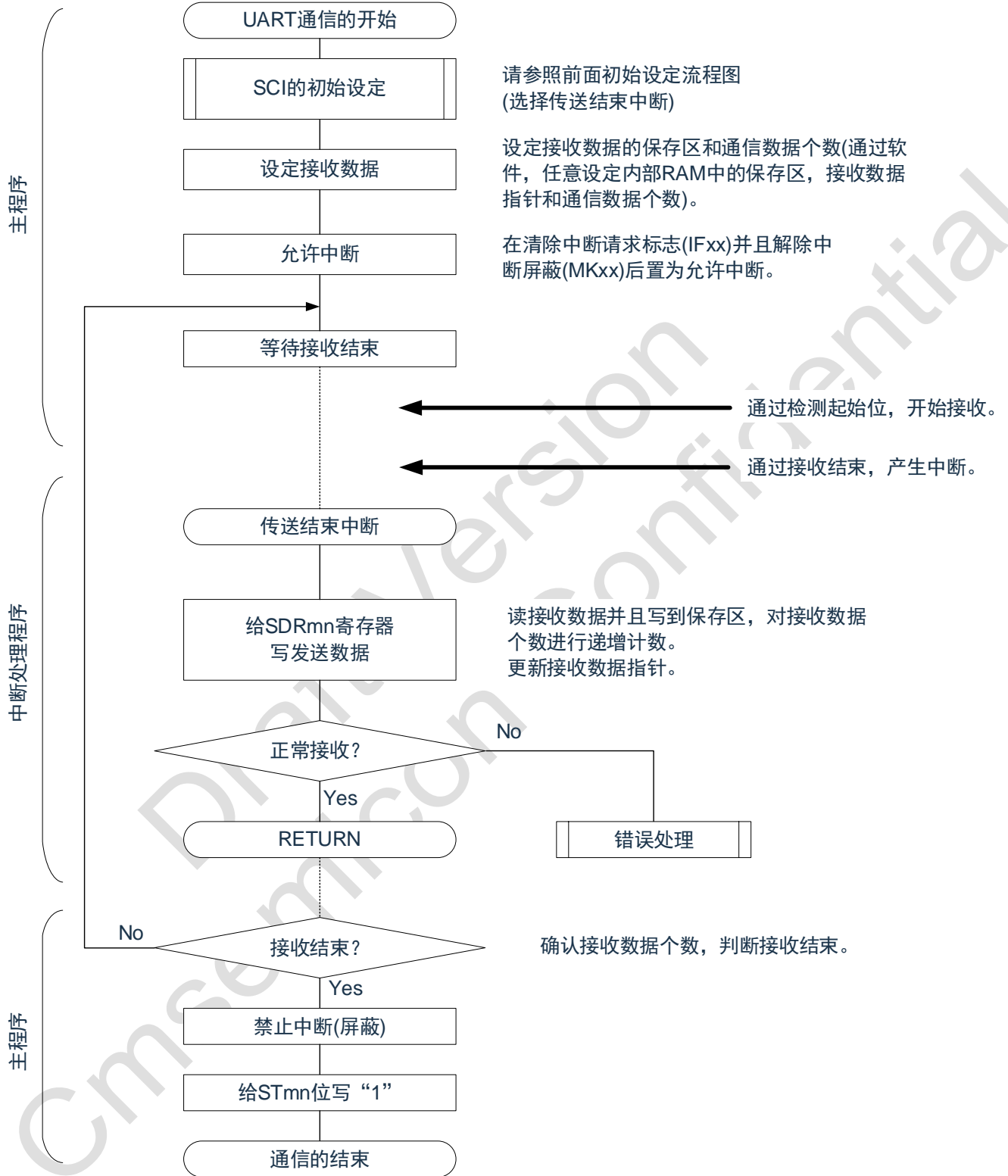
图17-108 UART接收的时序图



备注： m: 单元号 (m=0、1) n: 通道号 (n=1)  
 r: 通道号 (r=n-1) q: UART号 (q=0、2)

Draft Version  
 Cmsemicon Confidential

图17-109 UART接收的流程图





### 17.7.3 低功耗UART模式功能

这是在深度睡眠模式中通过检测 RxDq 引脚的输入使 UART 进行接收的模式。通常，在深度睡眠模式中 UART 停止通信，但是如果使用低功耗 UART 模式，就能在 CPU 不运行的状态下进行 UART 的接收。

要在低功耗 UART 模式中使用 UARTq 时，必须在转移到深度睡眠模式前进行以下的设定（参照“图 17-111 和图 17-113 低功耗 UART 模式运行的流程图”）。

- 在低功耗 UART 模式中，需要更改 UART 接收波特率的设定（更改为和通常运行时不同的值）。必须参照表 17-4，设定 SPSm 寄存器和 SDRmn 寄存器[15:9]。
- 设定 SSECMn 位。能设定允许或者停止在发生通信错误时产生错误中断。
- 必须在即将要转移到深度睡眠模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。在初始设定结束后，将串行通道开始寄存器 m（SSm）的 SSm1 位置“1”。
- 如果在转移到深度睡眠模式后检测到 RxDq 输入的起始位，UARTq 就开始接收。

注意 1. 只有在选择高速内部振荡器时钟（f<sub>IH</sub>）作为 f<sub>CLK</sub> 时才能使用低功耗 UART 模式。

2. 低功耗 UART 模式中的传送速率只为 4800bps。
3. 如果将 SWCm 位置“1”，就只能在深度睡眠模式中开始接收时使用 UARTq。当其他低功耗 UART 模式的功能和中断同时使用并且在以下非深度睡眠模式的状态下开始接收时，可能无法正常接收数据而发生帧错误或者奇偶校验错误。
  - 在将 SWCm 位置“1”后并且在转移到深度睡眠模式前开始接收的情况
  - 在其他低功耗 UART 模式中开始接收的情况
  - 在通过中断等从深度睡眠模式返回到通常运行后并且在将 SWCm 位置“0”前开始接收的情况
4. 如果将 SSECMn 位置“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFmn、FEFmn、OVFmn 标志置位。因此，在 SSECMn 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFmn、FEFmn、OVFmn 标志并且读 SDRm1 寄存器的 bit7 ~ 0（RxDq）。
5. 通过检测 RxDq 引脚的有效边沿，转移到低功耗 UART 模式。  
如果接受到无法检测到输入起始位的短脉冲，就可能不开始 UART 接收而继续保持低功耗 UART 模式。此时，可能在下一轮的 UART 接收过程中无法正常接收数据而发生帧错误或者奇偶校验错误。

表 17-4 低功耗UART模式中UART接收波特率的设定

高速内部振荡器 (fIH)	低功耗 UART 模式中的 UART 接收波特率			
	波特率 4800bps			
	运行时钟 (fMCK)	SDRmn[15:9]	最大容许值	最小容许值
32MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>5</sup>	105	2.27%	- 1.53%
24MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>5</sup>	79	1.60%	- 2.18%
16MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>4</sup>	105	2.27%	- 1.53%
12MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>4</sup>	79	1.60%	- 2.19%
8MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>3</sup>	105	2.27%	- 1.53%
6MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>3</sup>	79	1.60%	- 2.19%
4MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>2</sup>	105	2.27%	- 1.53%
3MHz ± 1.0% <sup>注</sup>	fCLK/2 <sup>2</sup>	79	1.60%	- 2.19%
2MHz ± 1.0% <sup>注</sup>	fCLK/2	105	2.27%	- 1.54%
1MHz ± 1.0% <sup>注</sup>	fCLK	105	2.27%	- 1.57%

注 当高速内部振荡器的频率精度为 ±1.5%、±2.0% 时，以下的容许范围会变窄。

- 在 fIH±1.5% 的情况下，必须将上表的最大容许值设定为-0.5%，并且将最小容许值设定为+0.5%。
- 在 fIH±2.0% 的情况下，必须将上表的最大容许值设定为-1.0%，并且将最小容许值设定为+1.0%。

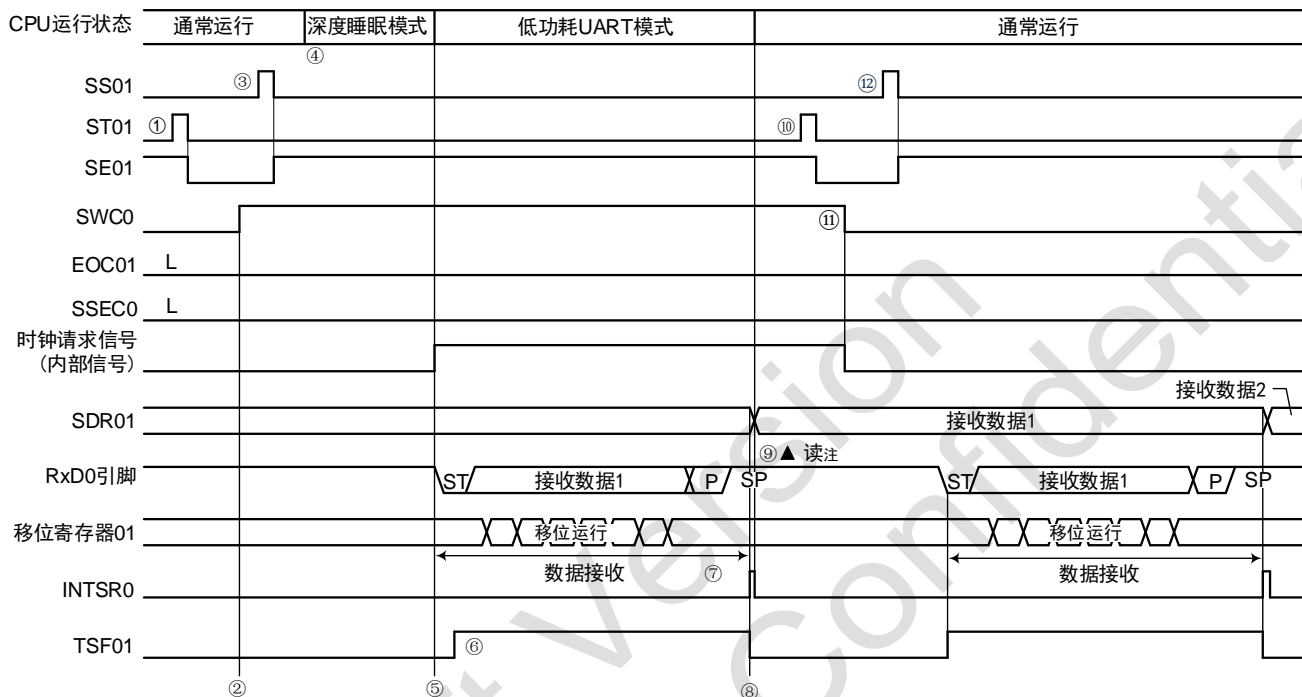
备注 最大容许值和最小容许值是 UART 接收时的波特率的容许值。

必须将发送方的波特率设定在此范围内。

(1) 低功耗 UART 模式运行 (EOCm1=0、SSECm=0/1)

因为 EOCm1 位为“0”，所以与 SSECm 位的设定无关，即使发生通信错误也不产生错误中断。但是，产生传送结束中断 (INTSRq)。

图 17-110 低功耗UART模式运行 (EOCm1=0、SSECm=0/1) 时序图

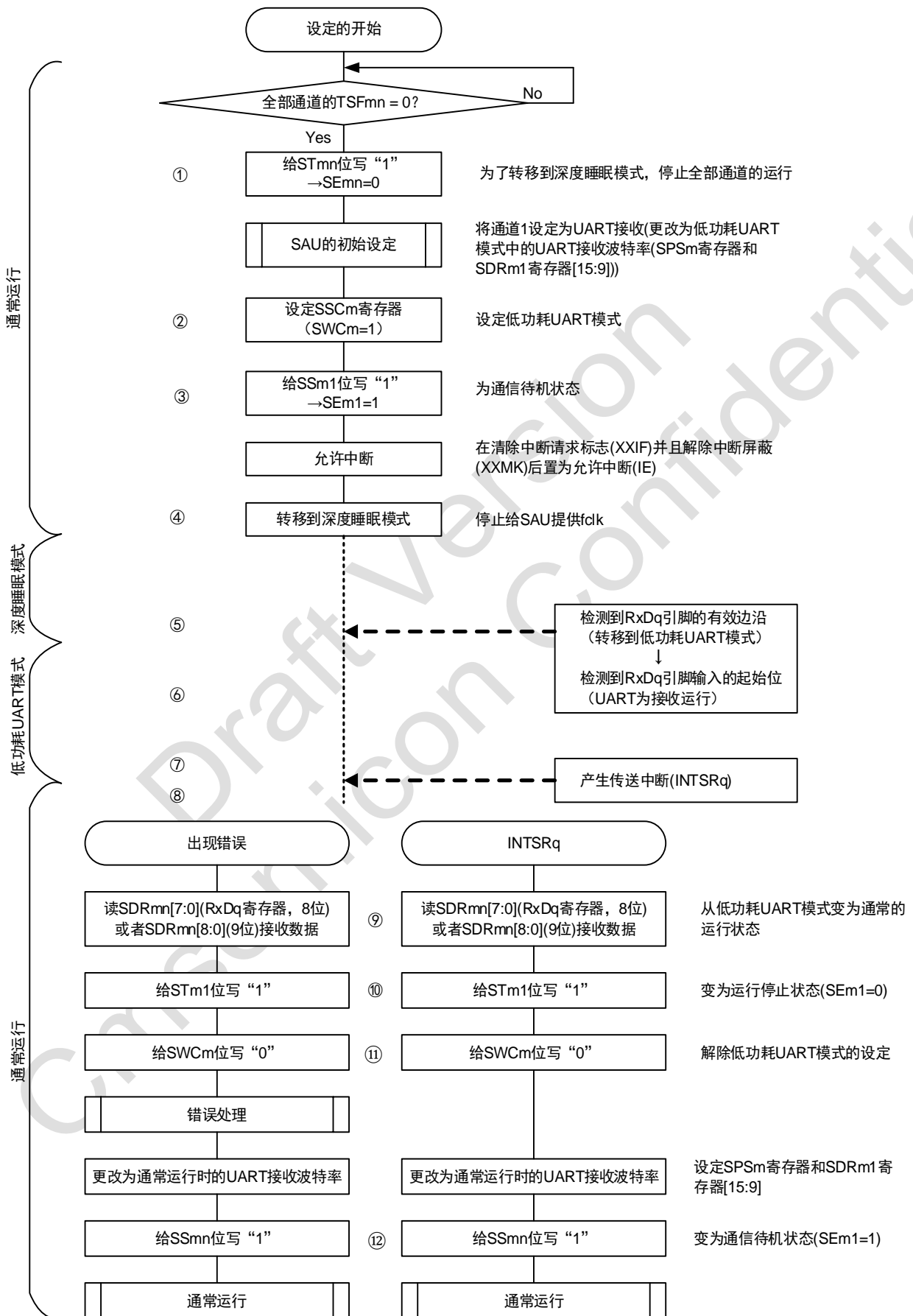


注 必须在 SWCm 位为“1”的状态下读接收数据。

注意 在向低功耗 UART 模式转移前或者在低功耗 UART 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除低功耗 UART 模式）。

备注 1. 图中的①~⑫对应“图 17-111 低功耗 UART 模式运行 (EOCm1=0、SSECm=0/1) 的流程图”中的①~⑫。

图 17-111 低功耗UART模式运行 (EOCm1=0、SSECm=0/1) 的流程图

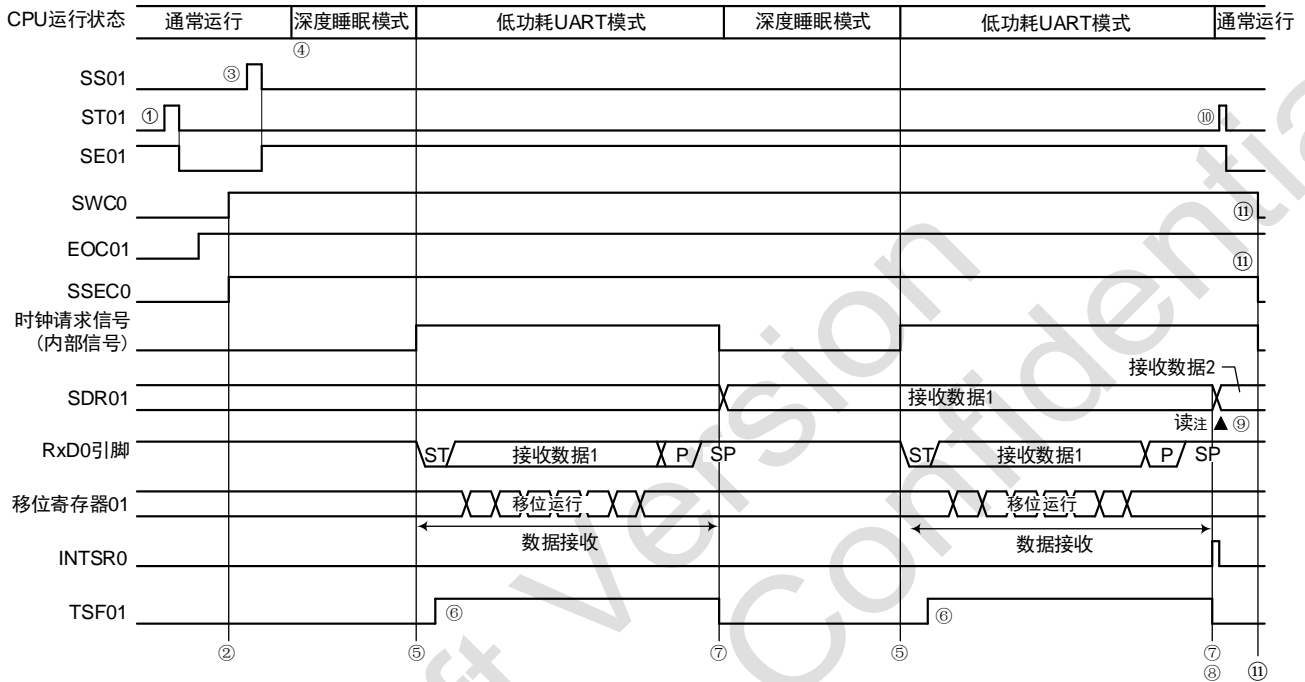


备注 图中的①~⑫ 对应“图 17-110 低功耗 UART 模式运行 (EOCm1=0、SSECm=0/1) 时序图”中的①~⑫。

(2) 低功耗 UART 模式运行 (EOCm1=1、SSECm=1)

因为 EOCm1 位为 “1”，所以发生通信错误不产生错误中断。

图 17-112 低功耗UART模式运行 (EOCm1=1、SSECm=1) 的时序图



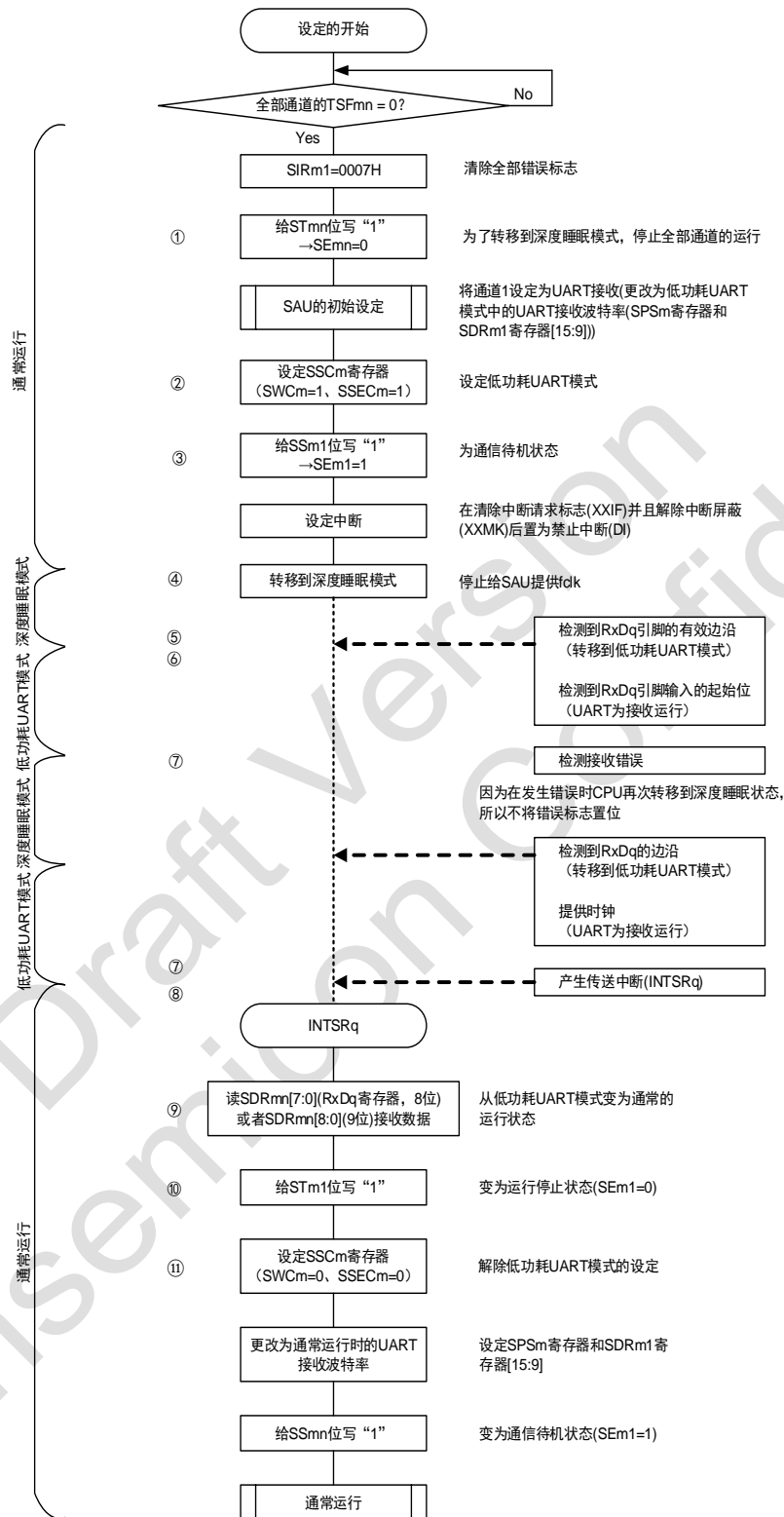
注 必须在 SWCm 位为 “1” 的状态下读接收数据。

注意 1. 在向低功耗 UART 模式转移前或者在低功耗 UART 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除低功耗 UART 模式）。

2. 如果 SSECm 位为 “1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位。因此，在 SSECm 位为 “1” 的情况下使用时，必须在将 SWCm 位置 “1” 前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0] (RXDq 寄存器，8 位) 或者 SDRm1[8:0] (9 位)。

备注 1. 图中的①~⑪对应“图 17-113 低功耗 UART 模式运行 (EOCm1=1、SSECm=1) 的流程图”中的①~⑪。

图 17-113 低功耗UART模式运行 (EOCm1=1、SSECm=1) 的流程图



注意 如果 SSECm 位为 “1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位，因此在 SSECm 位为 “1” 的情况下使用时，必须在将 SWC0 位置 “1” 前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0] (RXDq 寄存器, 8 位) 或者 SDRm1[8:0] (9 位)。

备注 1. 图中的①~⑪对应“图 17-112 低功耗 UART 模式运行 (EOCm1=1、SSECm=1) 的时序图”中的①~⑪。

## 17.7.4 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]设定为“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 (000010B~111111B)，所以为2~127。

2.m: 单元号 (m=0、1) n: 通道号 (n=0、1)

运行时钟 (f<sub>MCK</sub>) 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17-5 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	$f_{CLK}$	$f_{CLK}=32\text{MHz}$ 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	$f_{CLK}/2^5$	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、1）



## (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定在接收方的波特率容许范围内。

$$(\text{波特率误差}) = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100[\%]$$

$f_{\text{CLK}}=32\text{MHz}$ 时的UART波特率的设定例子如下所示。

UART波特率 (目标波特率)	$f_{\text{CLK}}=32\text{MHz}$			
	运行时钟 ( $f_{\text{MCK}}$ )	SDRmn[15:9]	波特率的计算值	与目标波特率的误差
300bps	$f_{\text{CLK}}/2^9$	103	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	63	31250.0bps	±0.0%
38400bps	$f_{\text{CLK}}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	103	76923.1bps	+0.16%
153600bps	$f_{\text{CLK}}$	103	153846bps	+0.16%
312500bps	$f_{\text{CLK}}$	50	313725bps	±0.39%

备注： m：单元号 (m=0、1) n：通道号 (n=0)

(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定在接收方的波特率容许范围内。

$$\text{(可接收的最大波特率)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - 1} \times \text{Brate}$$

$$\text{(可接收的最小波特率)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: 接收方的波特率的计算值 (参照“17.7.5 (1)波特率的计算式”)

)

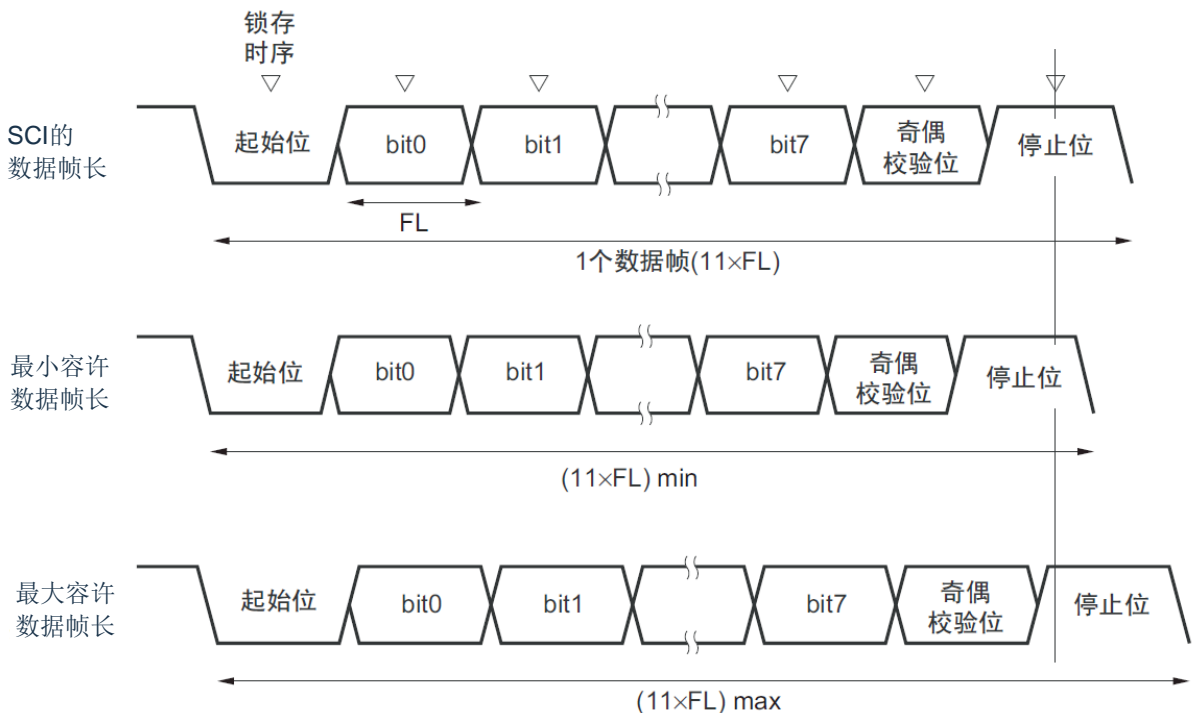
k : SDRmn[15:9]+1

Nfr : 1个数据的帧长[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)

图17-114 接收时的波特率容许范围 (1个数据的帧长=11位的情况)



如图17-114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设定的分频比。如果最后的数据 (停止位) 能赶上此锁存时序，就能正常接收。

## 17.7.5 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17-115和图17-116所示。

图17-115 发生奇偶校验错误或者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 (SIRmn) 写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SIRmn寄存器，只能清除读操作时的错误。

图17-116 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SIRmn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、1)

## 17.8 LIN通信的运行

### 17.8.1 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	UART2
LIN 通信的支持	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 <sub>MCK</sub> /6[bps] (SDR00[15:9]≥2)、Min.f <sub>CLK</sub> /(2 × 2 <sup>15</sup> × 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<sub>MCK</sub>：对象通道的运行时钟频率 f<sub>CLK</sub>：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。

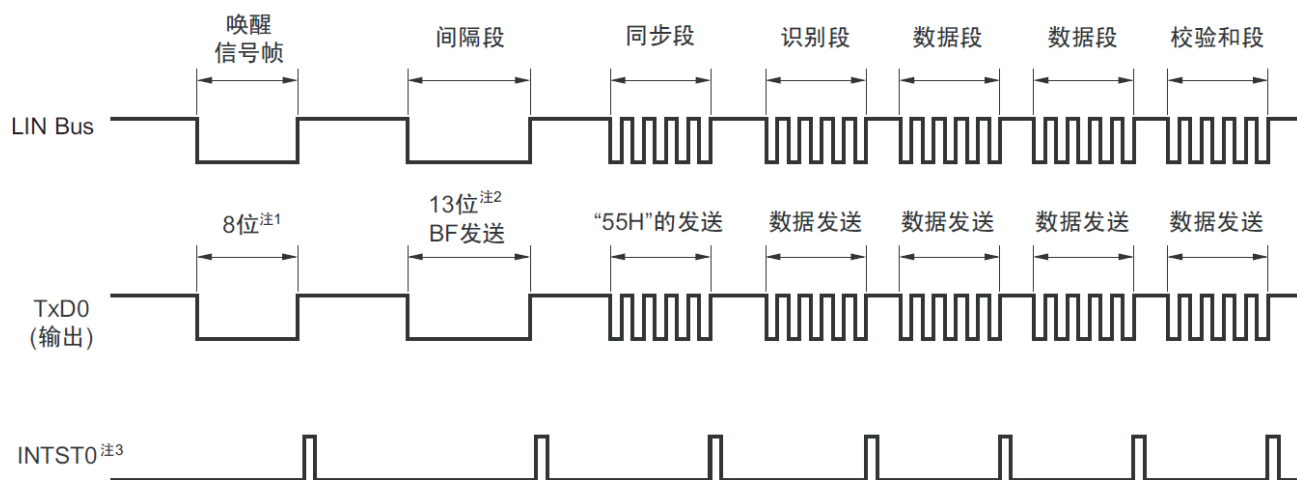
LIN主控一般连接CAN（Controller Area Network）等的网络。

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17-117所示。

图17-118 LIN的发送操作



- 注：
1. 为了满足唤醒信号的规定，设定波特率并且通过发送“80H”的数据进行对应。
  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：

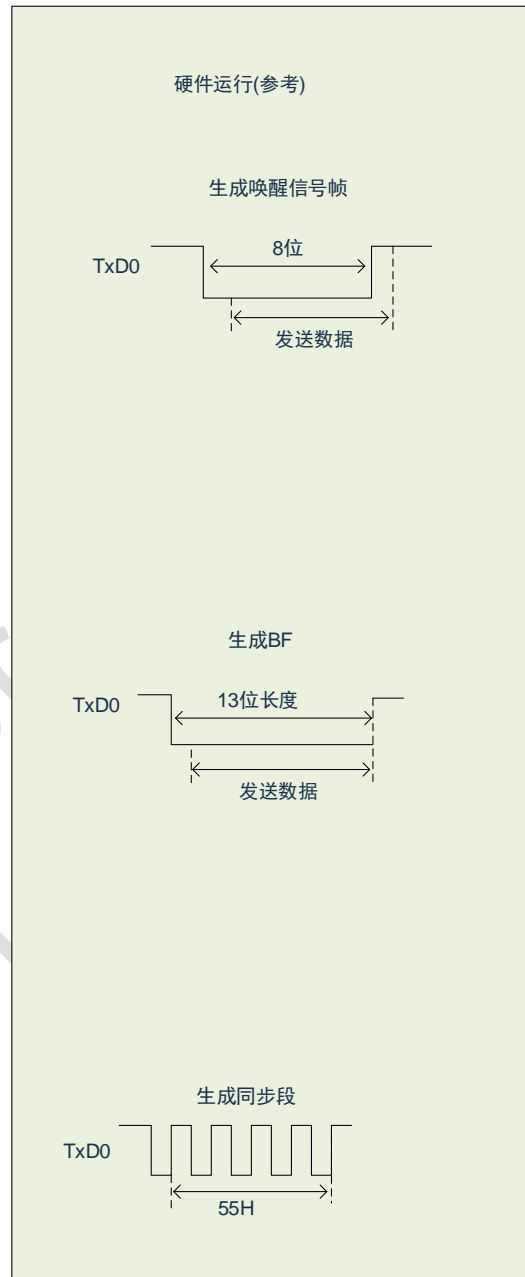
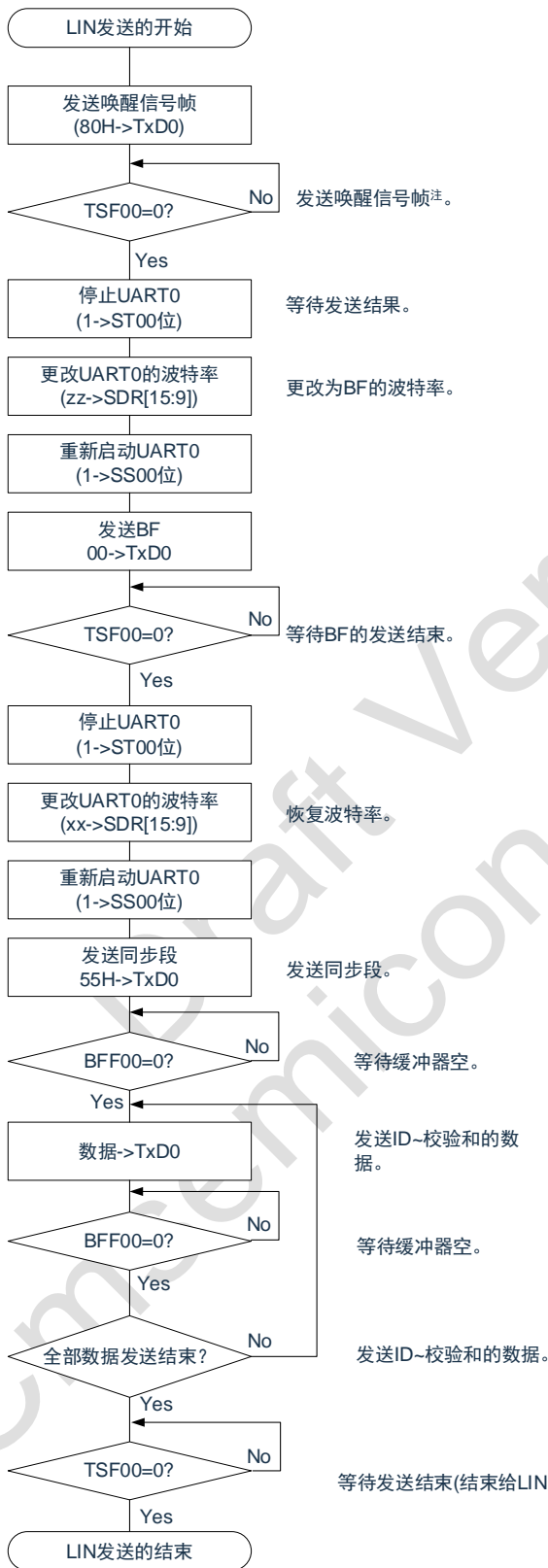
(间隔段的波特率)

通过此波特率发送“00H”的数据，生成间隔段。

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17-119 LIN发送的流程图



## 17.8.2 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。

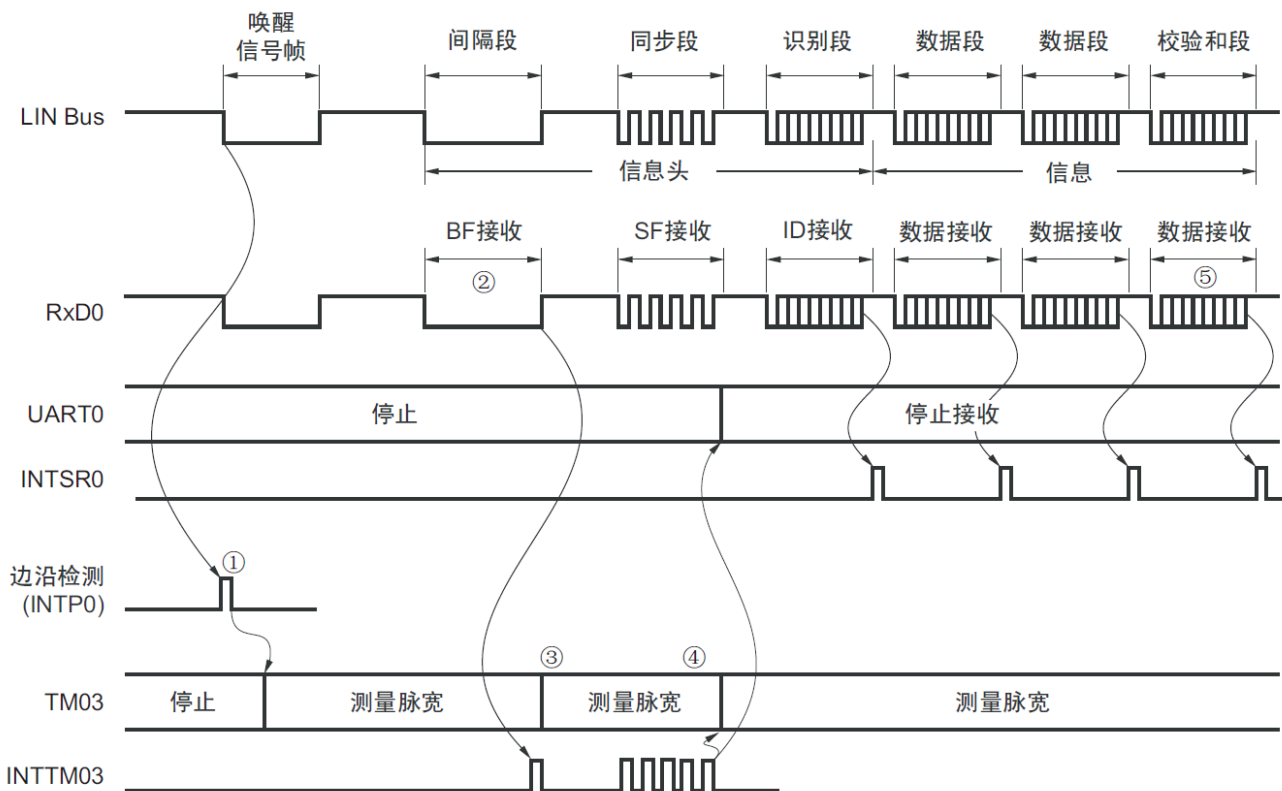
UART	UART0	UART2
LIN通信的支持	能	不能
对象通道	SCI0的通道1	—
使用的引脚	RxD0	—
中断	INTSR0	—
	只限于传送结束中断（禁止设定缓冲器空中断）。	
错误检测标志	<ul style="list-style-type: none"> <li>•帧错误检测标志（FEF01）</li> <li>•溢错误检测标志（OVF01）</li> </ul>	
传送数据长度	8位	
传送速率 <sup>注</sup>	Max.f <sub>MCK</sub> /6[bps]（SDR01[15:9]≥2）、Min.f <sub>CLK</sub> /(2 x 2 <sup>15</sup> x 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<sub>MCK</sub>：对象通道的运行时钟频率      f<sub>CLK</sub>：系统时钟频率

LIN接收操作的概要如图17-120所示。

图17-121 LIN的接收操作

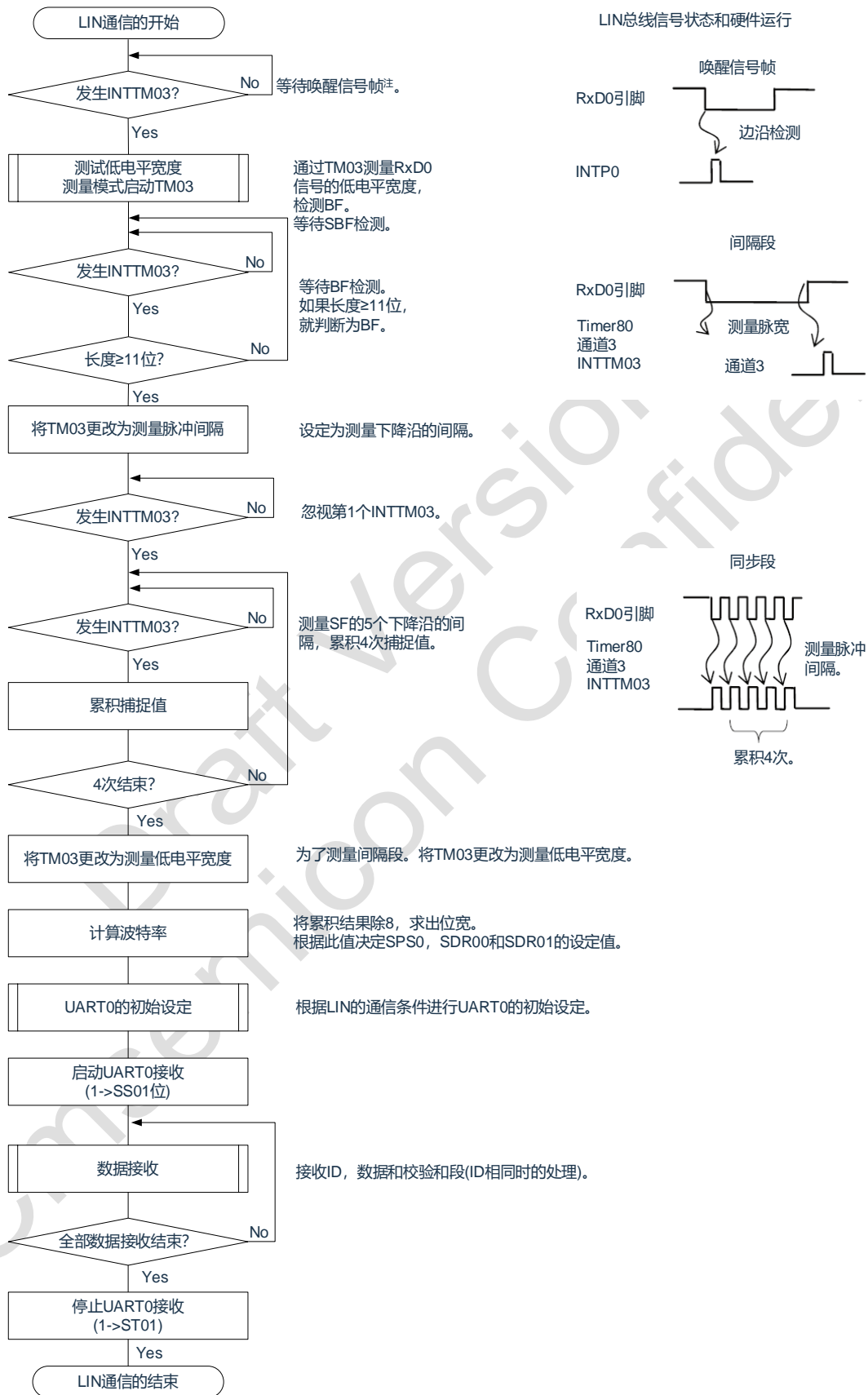


信号处理的流程如下：

- ① 通过检测引脚的中断边沿 (INTPO) 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设定为测量脉宽，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设定为测量脉冲间隔，并且测量4次同步段的RxD0信号下降沿的间隔（参照“5.8.4 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（重新设定）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定为BF接收等待状态。



图17-122 LIN接收的流程图



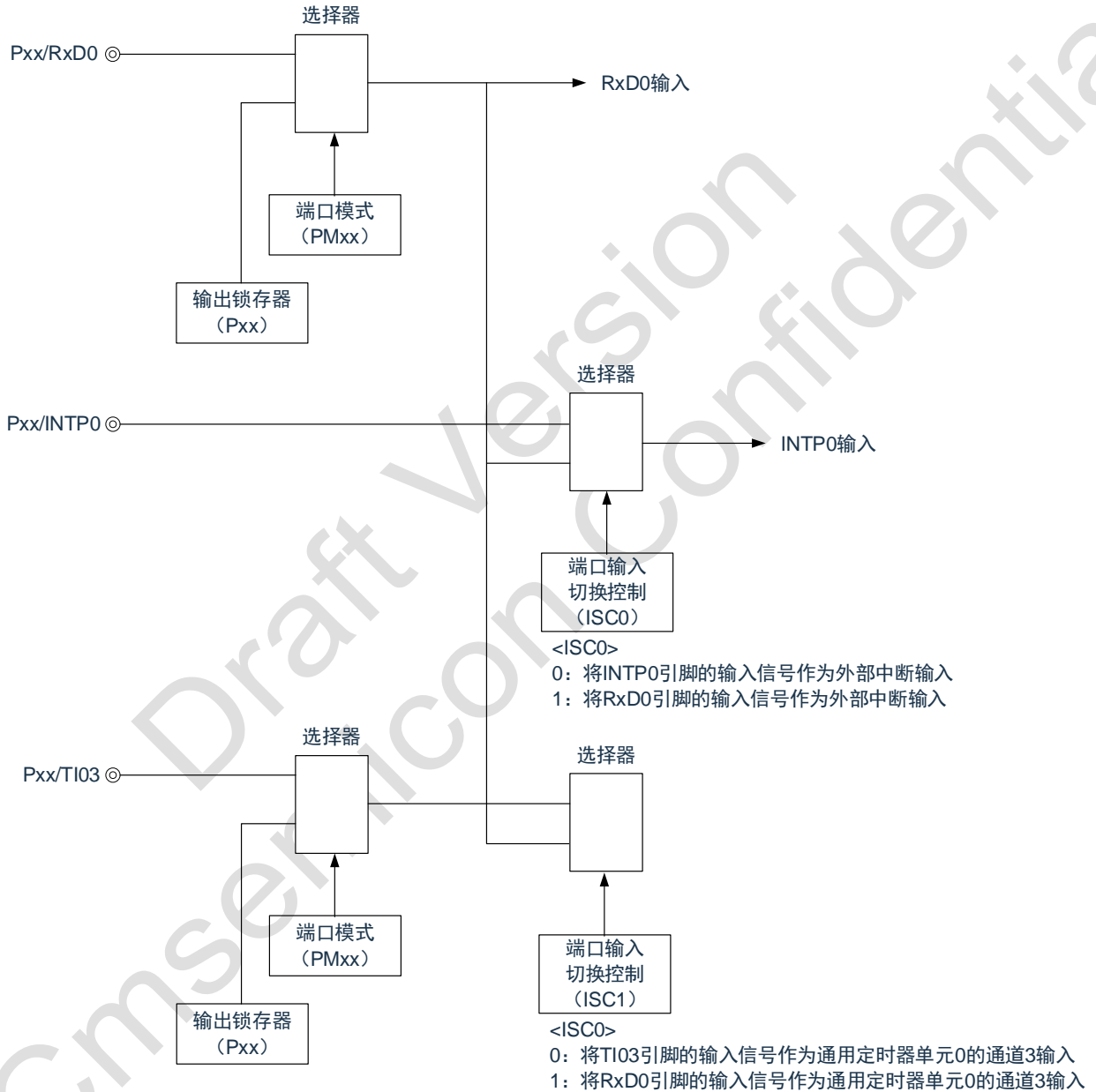
注：只在睡眠状态下才需要。

用于LIN接收操作的端口结构图如图17-123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17-123 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

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（INTP0）：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（SCI0）的通道0和通道1（UART0）

Draft Version  
Cmsemicon Confidential

## 17.9 简易I<sup>2</sup>C (IIC00、IIC20) 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I<sup>2</sup>C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<sup>注</sup>、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I<sup>2</sup>C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“17.9.3 (2)处理流程”。

备注: m: 单元号 (m=0、1) n: 通道号 (n=0)

SCI0的通道0和SCI1的通道0是支持简易I<sup>2</sup>C(IIC00、IIC20)的通道。

简易I<sup>2</sup>C (IIC00、IIC20) 有以下4种通信运行:

- 地址段发送 (参照17.9.1)
- 数据发送 (参照17.9.2)
- 数据接收 (参照17.9.3)
- 停止条件的产生 (参照17.9.4)

## 17.9.1 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I<sup>2</sup>C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 <sup>2</sup> C	IIC00	IIC20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC20
	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] ( $SDRmn[15:9] \geq 1$ ) $f_{MCK}$ : 对象通道的运行时钟频率但是，必须在 I <sup>2</sup> C 的各模式中满足以下条件： <ul style="list-style-type: none"> <li>•Max.1MHz（增强型快速模式）</li> <li>•Max.400kHz（快速模式）</li> <li>•Max.100kHz（标准模式）</li> </ul>	
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	

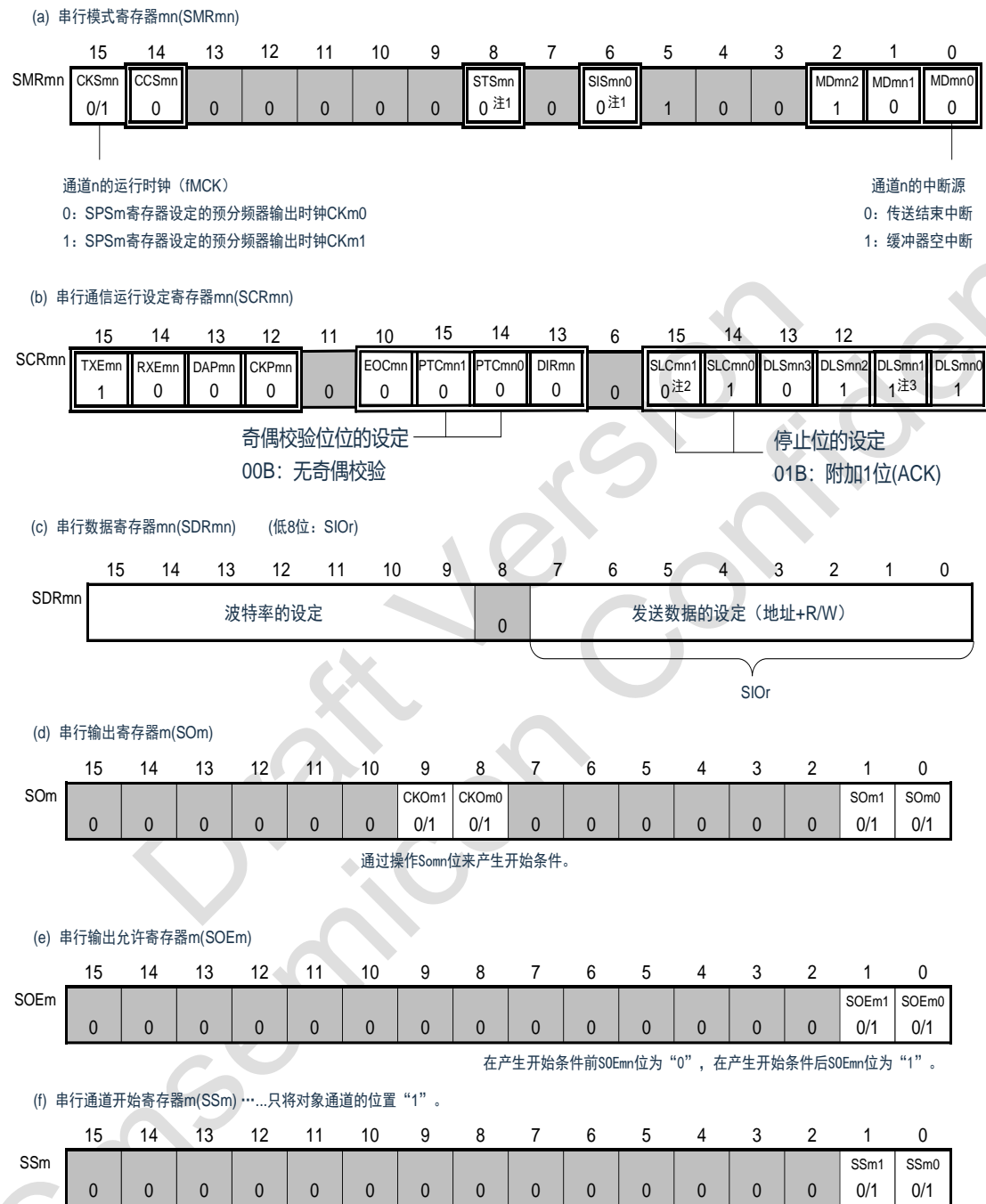
注：1.要通过简易I<sup>2</sup>C进行通信时，必须通过端口输出模式寄存器（POMxx）设定N沟道漏极开路输出模式（POMxx=1）。详细内容请参照“第2章 引脚功能”

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）

(1) 寄存器的设定

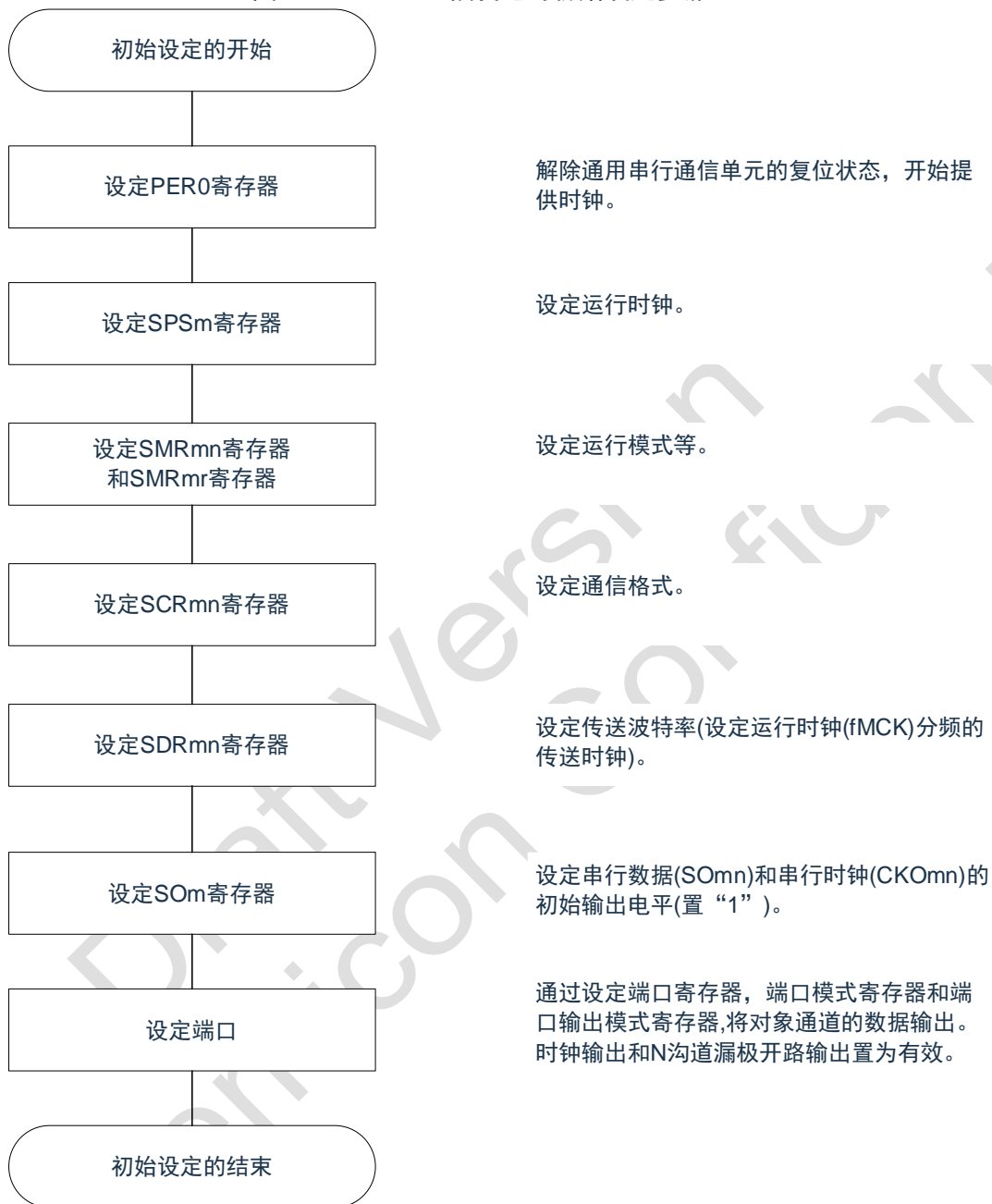
图17-124 简易I<sup>2</sup>C (IIC00、IIC20) 地址段发送时的寄存器设定内容例子



- 注:
- 只限于SMR01、SMR11。
  - 只限于SCR00、SCR10。
  - 只限于SCR00寄存器和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) r: IIC号 (r=00、20)
2. □: 在IIC模式中为固定设定。 ■: 不能设定 (设定初始值)。
- ×: 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，设定初始值)。
- 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

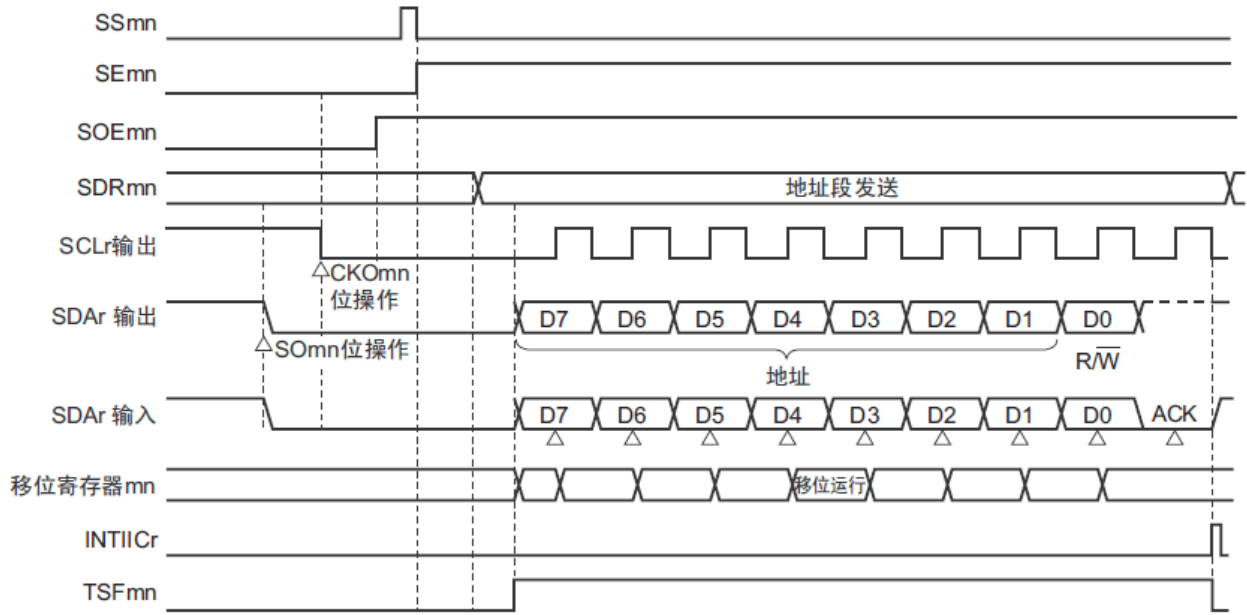
图17-125 地址段发送的初始设定步骤



备注： 在初始设定结束时，简易I<sup>2</sup>C（IIC00、IIC20）为禁止输出并且处于运行停止状态。

(3) 处理流程

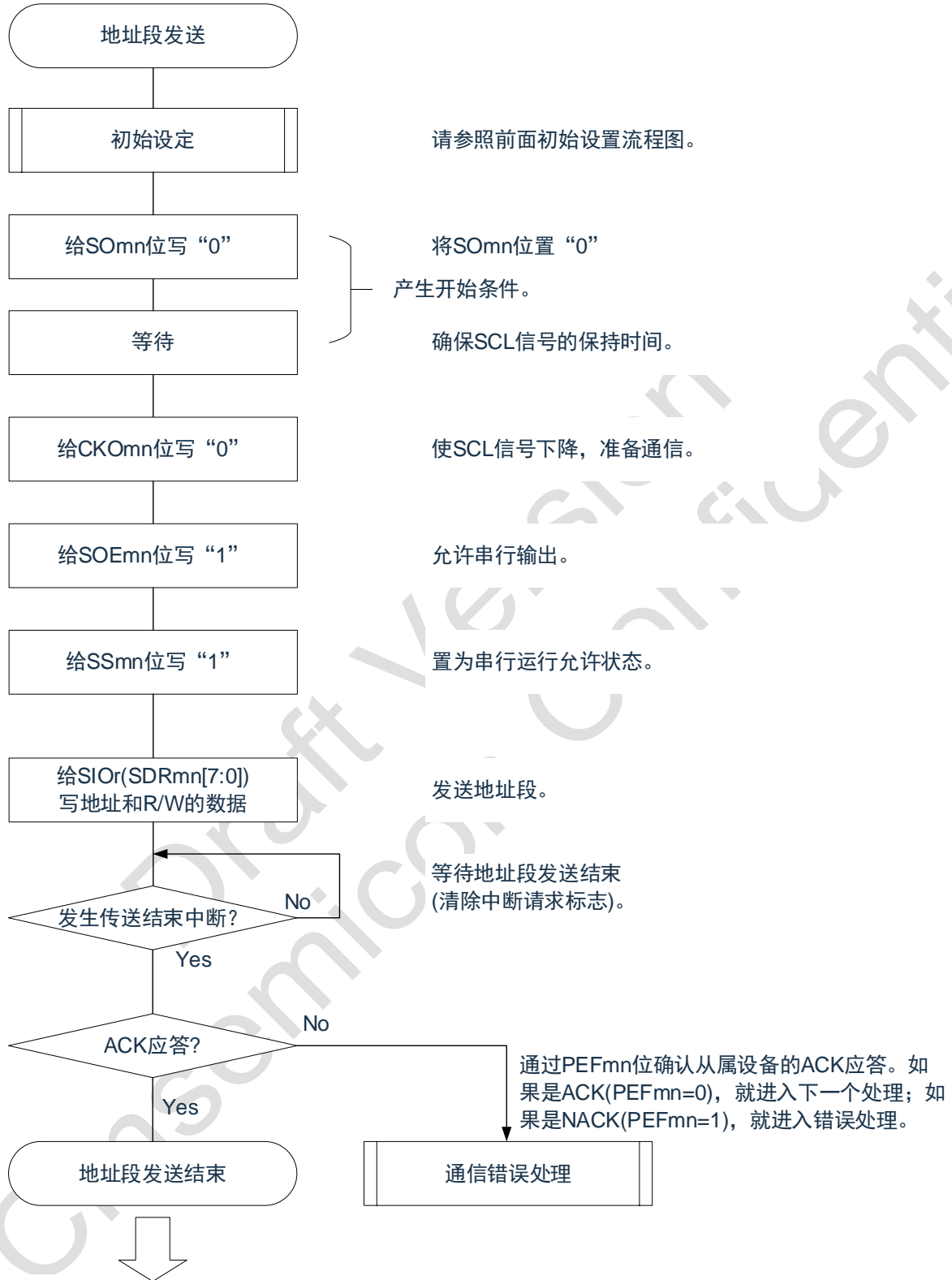
图17-126 地址段发送的时序图



备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1) r: IIC号 (r=00、20)



图17-127 地址段发送的流程图



## 17.9.2 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 I2C	IIC00	IIC20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC20
	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.fMCK/4[Hz] (SDRmn[15:9]≥1) fMCK：对象通道的运行时钟频率 但是，必须在 I2C 的各模式中满足以下条件： <ul style="list-style-type: none"> <li>•Max.1MHz（增强型快速模式）</li> <li>•Max.400kHz（快速模式）</li> <li>•Max.100kHz（标准模式）</li> </ul>	
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）设定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定”。

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）

(1) 寄存器的设定

图17-128 简易I<sup>2</sup>C (IIC00、IIC20) 数据发送时的寄存器设定内容例子

(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此寄存器。

	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	DLSmn3	DLSmn2	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<sub>r</sub>)……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
SDRmn	波特率的设定注4								0	发送数据的设定						
										SIO <sub>r</sub>						

(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							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1注5	0/1注5	0	0	0	0	0	0	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															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

- 注:
- 1.只限于SMR01、SMR11寄存器。
  - 2.只限于SCR00、SCR10寄存器。
  - 3.只限于SCR00、SCR01、SCR10、SCR11寄存器,其他固定为“1”。
  - 4.因为已经在发送地址段时设定,所以不需要设定。
  - 5.在通信运行过程中,值因通信数据而变。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0) r: IIC号 (r=00、20)

2.  : 在IIC模式中为固定设定。  : 不能设定(设定初始值)。

×: 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,设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2) 处理流程

图17-129 数据发送的时序图

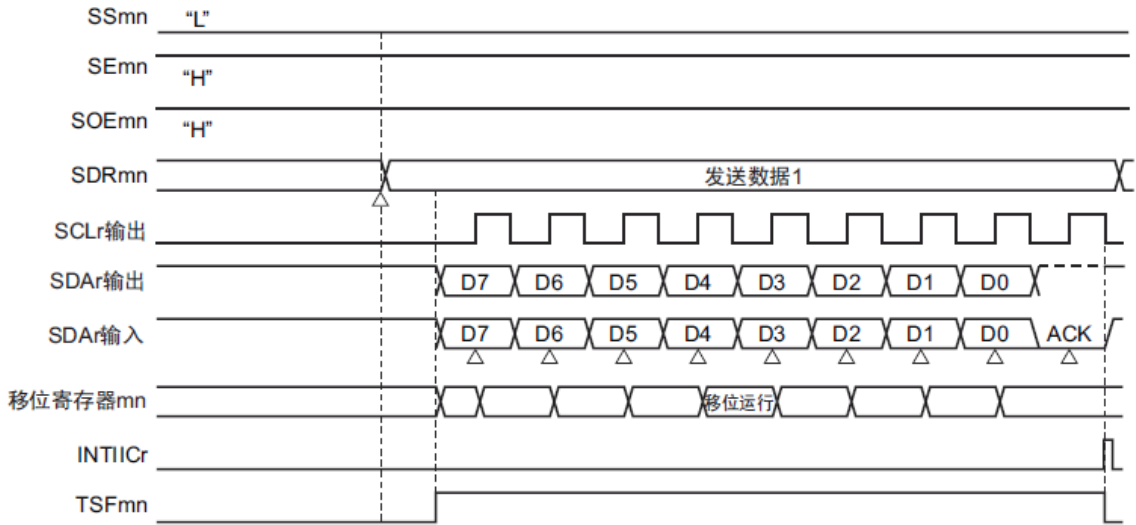
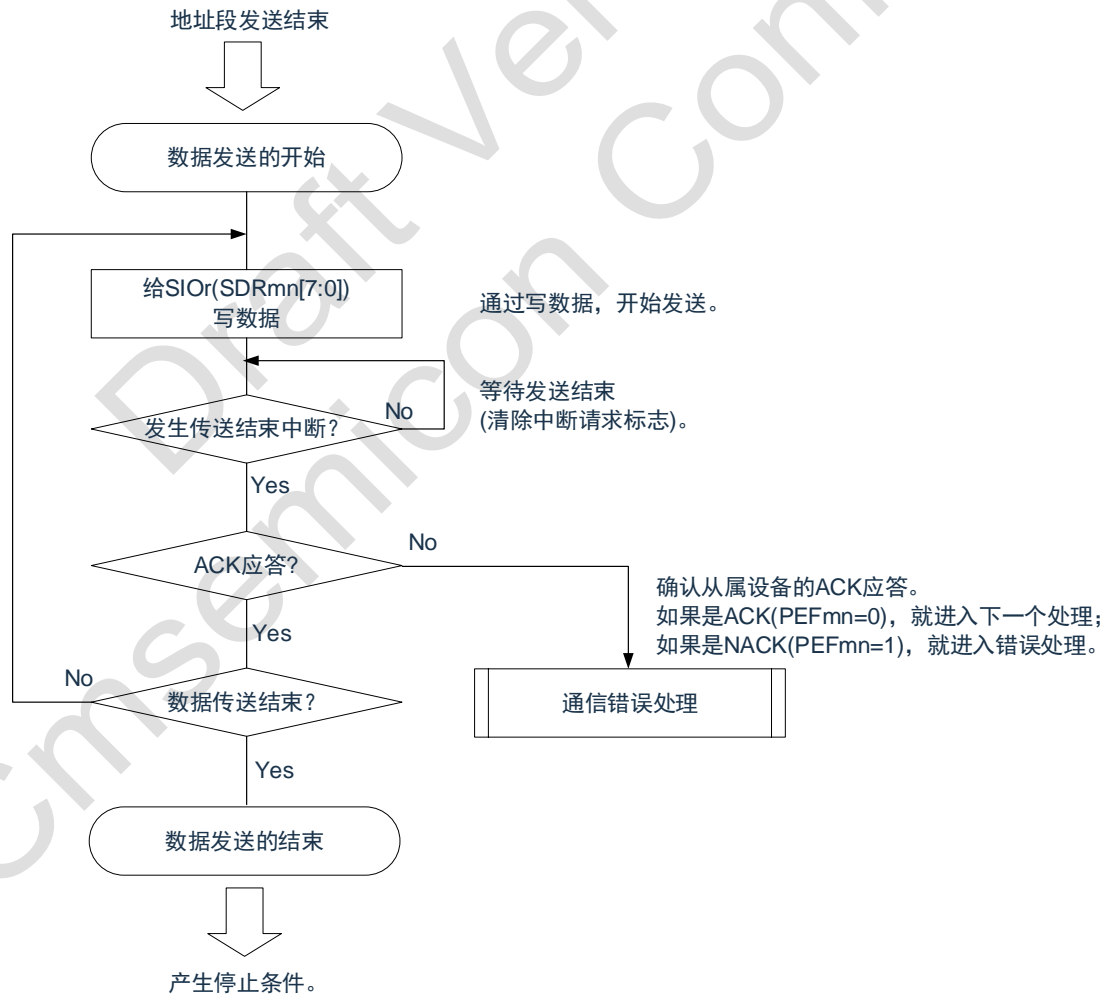


图17-130 数据发送的流程图



### 17.9.3 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 I2C	IIC00	IIC20
对象通道	SCI0 的通道 0	SCI1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC20
	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.fMCK/4[Hz] (SDRmn[15:9]≥1) fMCK：对象通道的运行时钟频率但是，必须在 I2C 的各模式中满足以下条件： <ul style="list-style-type: none"> <li>•Max.1MHz（增强型快速模式）</li> <li>•Max.400kHz（快速模式）</li> <li>•Max.100kHz（标准模式）</li> </ul>	
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）设定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定”。

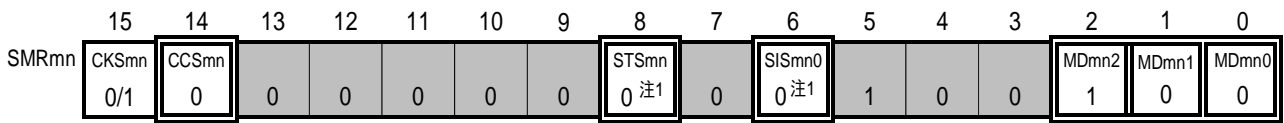
2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：m：单元号（m=0、1） n：通道号（n=0）

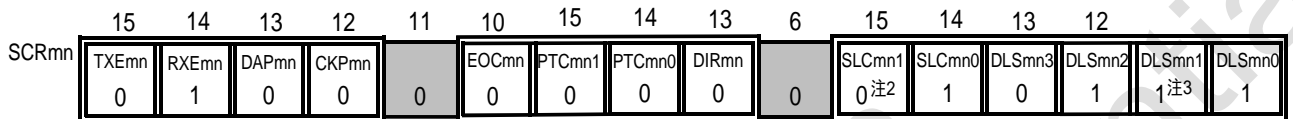
(1) 寄存器的设定

图17-131 简易I<sup>2</sup>C (IIC00、IIC20) 数据接收时的寄存器设定内容例子

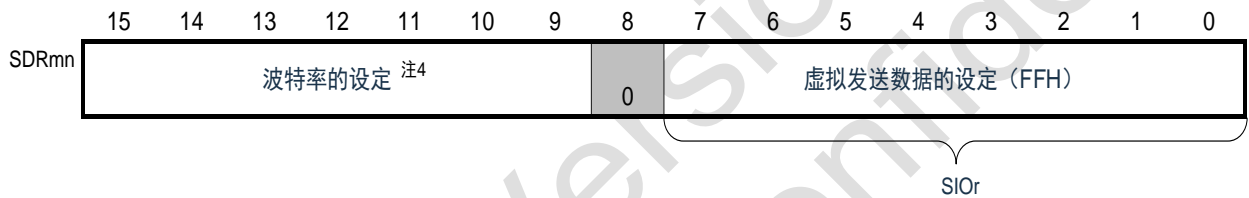
(a) 串行模式寄存器mn(SMRmn)....在数据发送和接收的过程中不操作此寄存器。



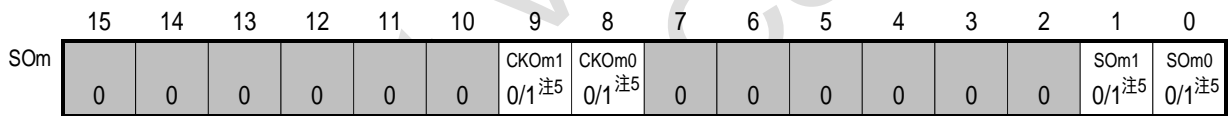
(b) 串行通信运行设定寄存器mn(SCRmn)....在数据发送和接收的过程中不操作TXEmn位和RXEmn位以外的位



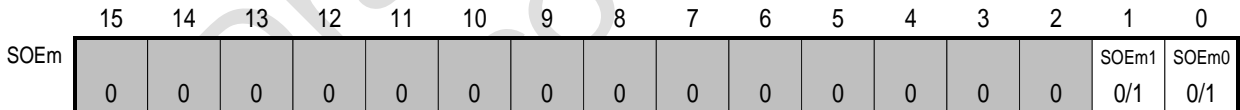
(c) 串行数据寄存器mn(SDRmn) (低8位: SIO<sub>r</sub>)



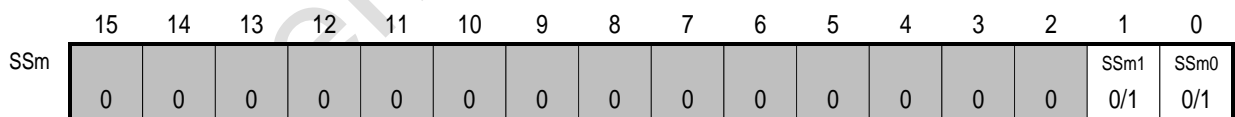
(d) 串行输出寄存器m(SOm) ....在数据发送和接收的过程中不操作此寄存器。



(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。



(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。



- 注:
- 1.只限于SMR01、SMR11寄存器。
  - 2.只限于SCR00、SCR10寄存器。
  - 3.只限于SCR00、SCR01、SCR10、SCR11寄存器，其他固定为“1”。
  - 4.因为已经在发送地址段时设定，所以不需要设定。
  - 5.在通信运行过程中，值因通信数据而变。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0) r: IIC号 (r=00、20)

2.  : 在IIC模式中为固定设定。  : 不能设定 (设定初始值)。

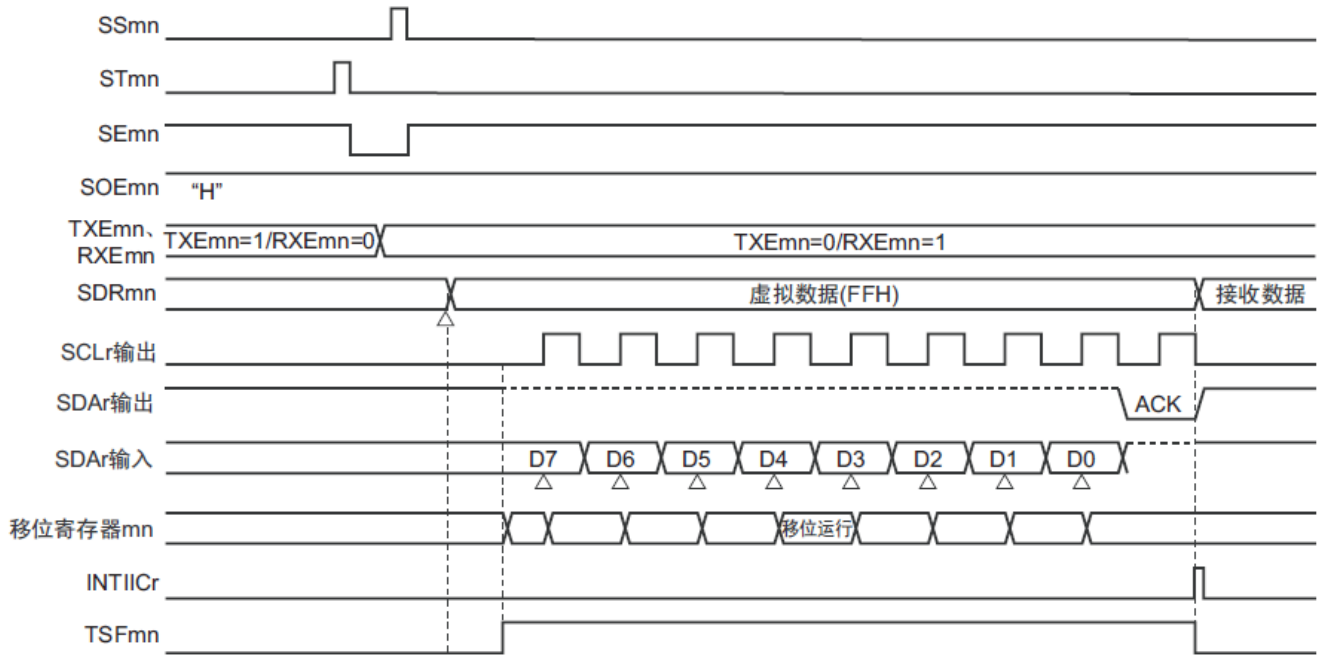
×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下，设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

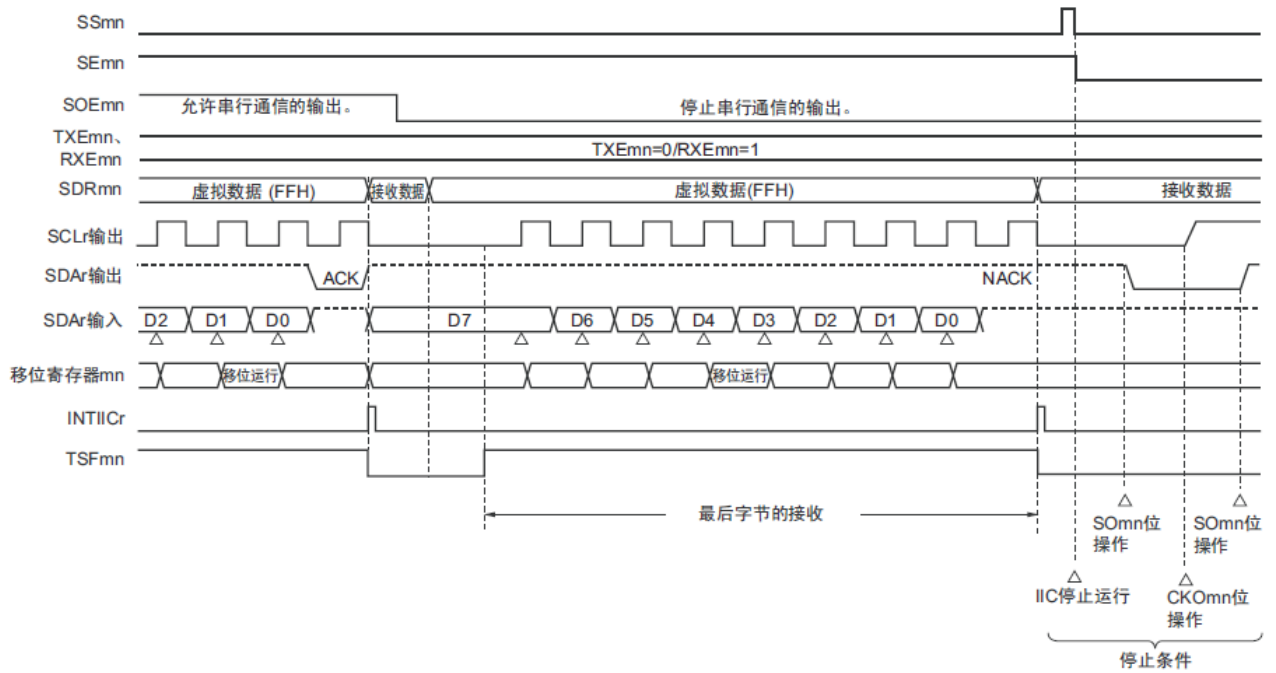
(2) 处理流程

图17-132 数据接收的时序图

(a)开始接收数据的情况

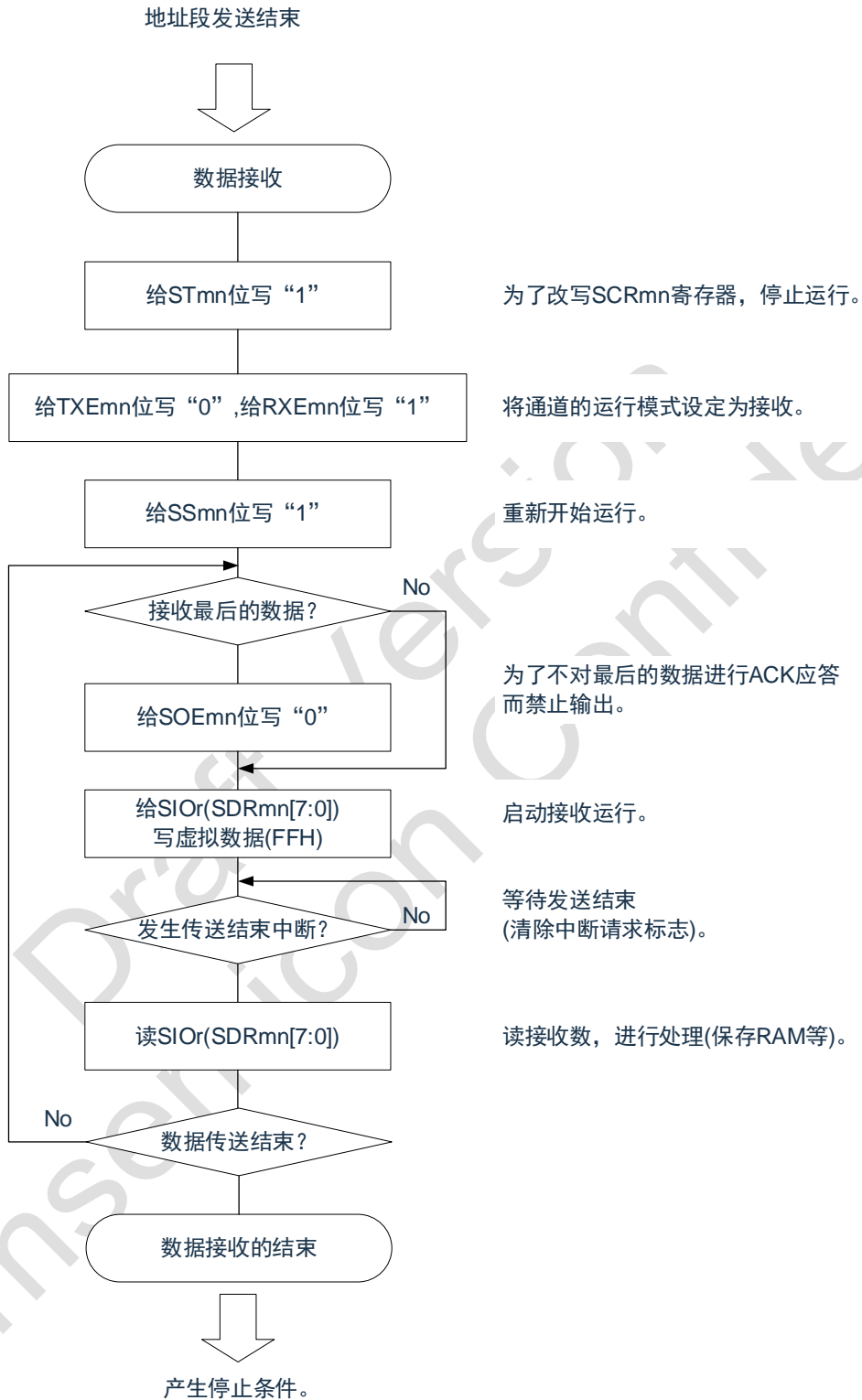


(b)接收最后数据的情况



备注： m: 单元号 (m=0、1) n: 通道号 (n=0) r: IIC号 (r=00、20)

图17-133 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。

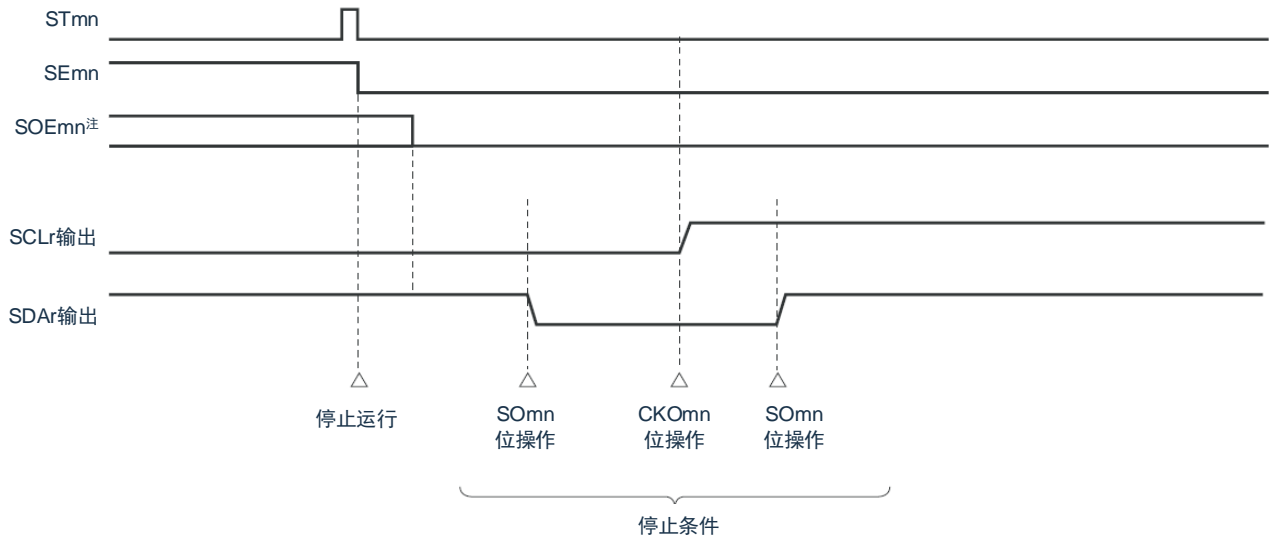


### 17.9.4 停止条件的产生

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17-134 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17-135 产生停止条件的流程图

数据发送结束/数据接收结束



## 17.9.5 传送速率的计算

简易I<sup>2</sup>C（IIC00、IIC20）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]设定为“0000000B”，SDRmn[15:9]的设定值必须大于等于“0000001B”。简易I<sup>2</sup>C输出的SCL信号的占空比为50%。在I<sup>2</sup>C总线规格中，SCL信号的低电平宽度大于高电平宽度。因此，如果设定为快速模式的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I<sup>2</sup>C总线的规格值。必须给SDRmn[15:9]设定能满足I<sup>2</sup>C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）

运行时钟（f<sub>MCK</sub>）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17-6 简易I<sup>2</sup>C运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 (f <sub>MCK</sub> ) 注		
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f <sub>CLK</sub> =32MHz运行时	
0	X	X	X	X	0	0	0	0	0	f <sub>CLK</sub>	32MHz
	X	X	X	X	0	0	0	1	1	f <sub>CLK</sub> /2	16MHz
	X	X	X	X	0	0	1	0	0	f <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	X	X	X	X	0	0	1	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	X	X	X	X	0	1	0	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	X	X	X	X	0	1	0	1	1	f <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	X	X	X	X	0	1	1	0	0	f <sub>CLK</sub> /2 <sup>6</sup>	500kHz
	X	X	X	X	0	1	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	250kHz
	X	X	X	X	1	0	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	125kHz
	X	X	X	X	1	0	0	1	1	f <sub>CLK</sub> /2 <sup>9</sup>	62.5kHz
	X	X	X	X	1	0	1	0	0	f <sub>CLK</sub> /2 <sup>10</sup>	31.25kHz
	X	X	X	X	1	0	1	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	15.63kHz
1	0	0	0	0	X	X	X	X	X	f <sub>CLK</sub>	32MHz
	0	0	0	1	X	X	X	X	X	f <sub>CLK</sub> /2	16MHz
	0	0	1	0	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	0	0	1	1	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	0	1	0	0	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	0	1	0	1	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	0	1	1	0	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	500kHz
	0	1	1	1	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	250kHz
	1	0	0	0	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	125kHz
	1	0	0	1	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	62.5kHz
	1	0	1	0	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	31.25kHz
	1	0	1	1	X	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	15.63kHz
上述以外									禁止设定。		

注：要更改被选择为f<sub>CLK</sub>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）

f<sub>MCK</sub>=f<sub>CLK</sub>=32MHz时的I<sup>2</sup>C传送速率的设定例子如下所示。

I <sup>2</sup> C传送模式 (期待的传送速率)	f <sub>CLK</sub> =32MHz			
	运行时钟 (f <sub>MCK</sub> )	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100kHz	f <sub>CLK</sub> /2	79	100kHz	0.0%
400kHz	f <sub>CLK</sub>	41	380kHz	5.0%注
1MHz	f <sub>CLK</sub>	18	0.84MHz	16.0%注

注：因为SCL信号的占空比为50%，所以不能将误差设定为“0%”左右。

## 17.9.6 在简易I<sup>2</sup>C（IIC00、IIC20）通信过程中发生错误时的处理步骤

在简易I<sup>2</sup>C（IIC00、IIC20）通信过程中发生错误时的处理步骤如图17-136和图17-137所示。

图17-136 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（SIRmn）写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SIRmn寄存器，只能清除读操作时的错误。

图17-137 在简易I<sup>2</sup>C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（SIRmn）。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SIRmn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） r：IIC号（r=00、20）

## 第18章 串行接口IICA

### 18.1 串行接口IICA的功能

本产品搭载两个串行接口IICA0，有以下3种模式。

#### 18.1.1 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。

#### 18.1.2 I<sup>2</sup>C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I<sup>2</sup>C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I<sup>2</sup>C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。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位进行设定。

串行接口IICA的框图如图18-1所示。

备注 n=0

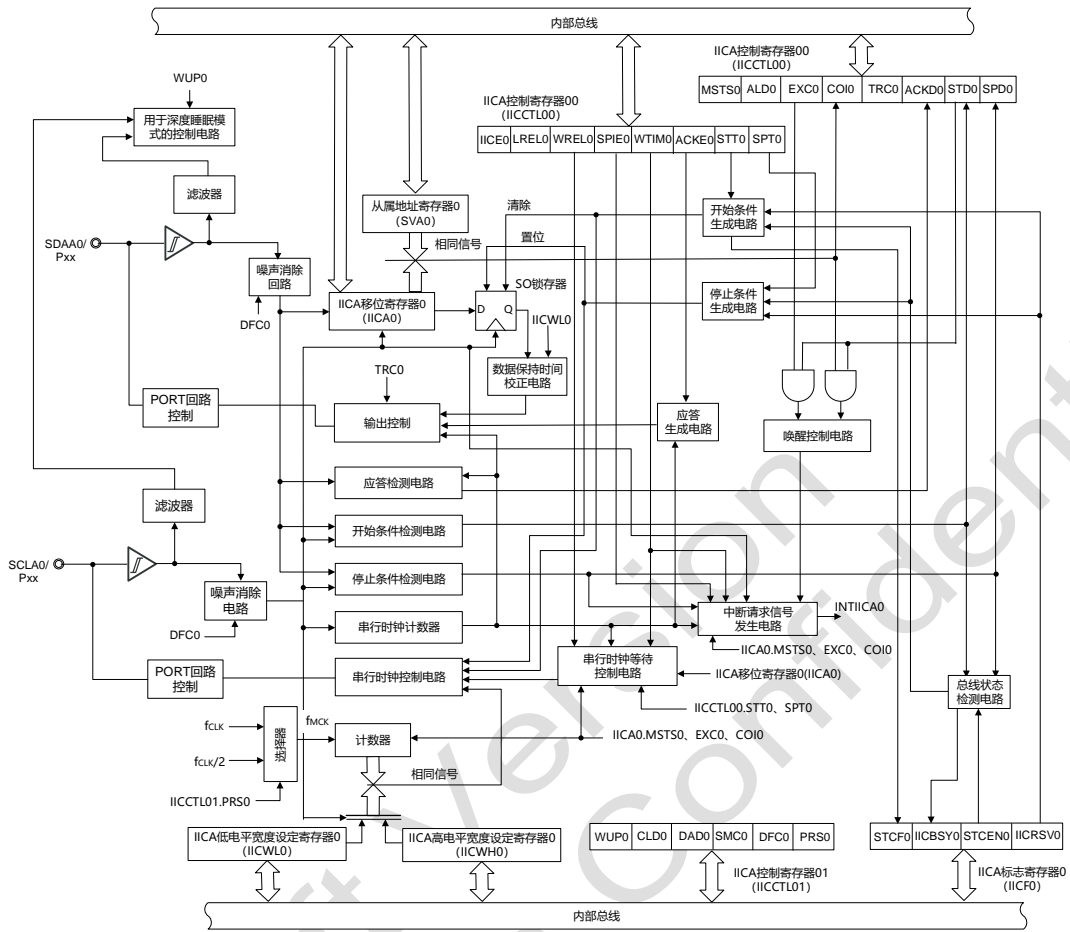
#### 18.1.3 唤醒模式

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18-1所示。

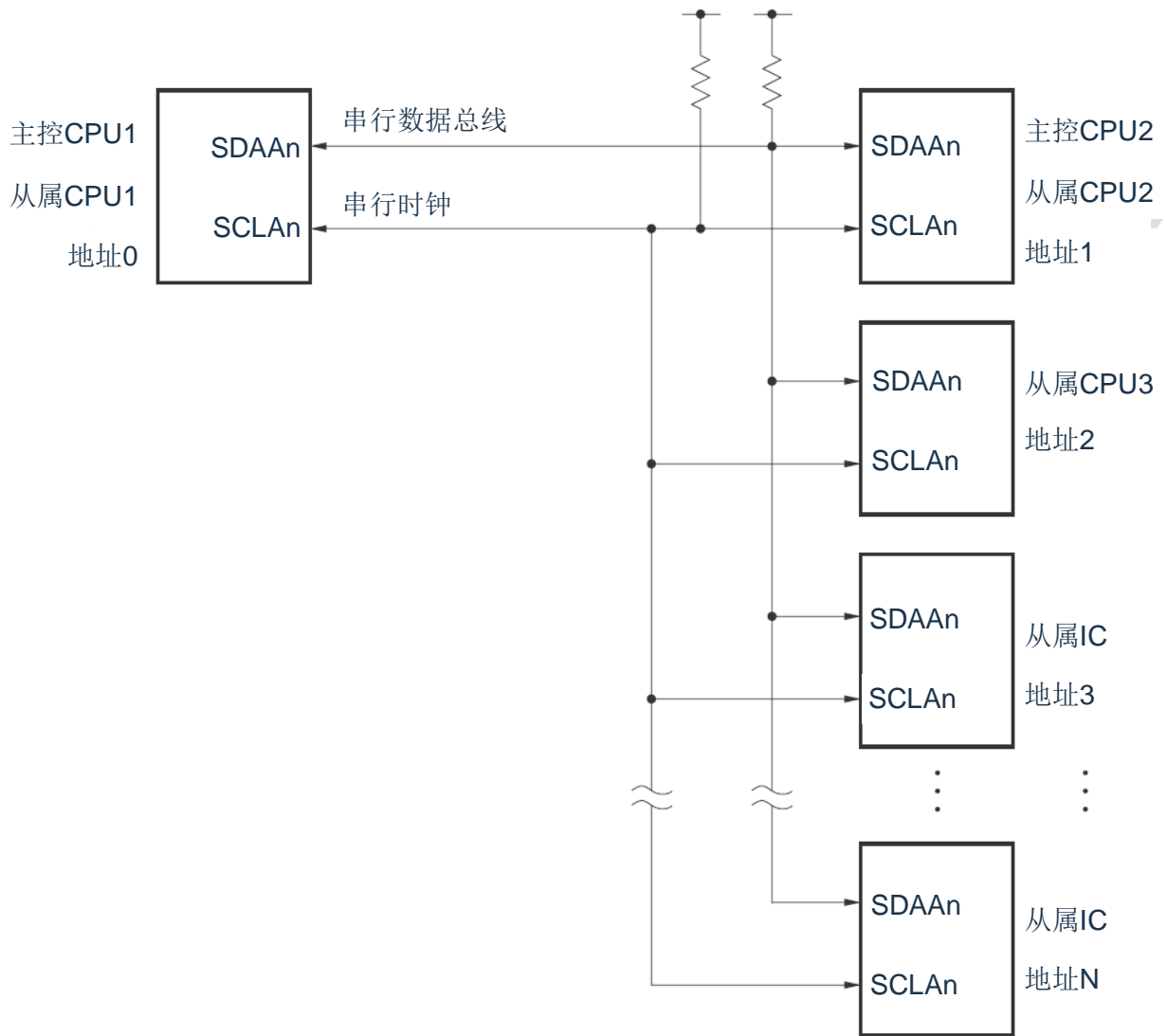
备注 n=0

图18-1 串行接口IICA的框图



串行总线的结构例子如图18-2所示。

图18-2 I<sup>2</sup>C总线的串行总线结构例子



备注： n=0

## 18.2 串行接口IICA的结构

串行接口IICA由以下硬件构成。

表18-1 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低电平宽度设定寄存器n (IICWLn) IICA高电平宽度设定寄存器n (IICWHn) 端 口模式寄存器 (PMxx) 端口模式控制寄存器 (PMCxx) 端口复用功能配置寄存器 (PxxCFG)

备注：1.n=0

- 2.本产品可将IICA输入/输出引脚功能复用到多个端口。当某端口被配置为IICA引脚的复用功能后，该端口的N沟道漏极开路输出 ( $V_{DD}/EV_{DD}$ 耐压) 模式由设计保证自动打开，即POMxx寄存器不需要用户设置。

寄存器列表：

基地址	偏移地址	寄存器名称	R/W	复位值
0x40041A30	0x120	IICA0	R/W	00H
	0x121	IICS0	R	00H
	0x122	IICF0	R/W	00H
	0x000	IICCTL00	R/W	00H
	0x001	IICCTL01	R/W	00H
	0x002	IICWL0	R/W	FFH
	0x003	IICWH0	R/W	FFH
	0x004	SVA00	R/W	00H
	0x005	SVA01	R/W	00H



## 18.2.1 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定IICAn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图18-3 IICAn移位寄存器n (IICAn) 的格式

复位后: 00H, R/W



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

## 18.2.2 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定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”。如下图所示，本产品支持两个从机地址。本文其它地方出现SVAn代表SVAn0或SVAn1。

图18-4 从属地址寄存器n (SVAn) 的格式

复位后: 00H, R/W

符号	7	6	5	4	3	2	1	0
SVAn0	A6	A5	A4	A3	A2	A1	A0	0注

复位后: 00H, R/W

符号	7	6	5	4	3	2	1	0
SVAn1	A6	A5	A4	A3	A2	A1	A0	0注

注: bit0固定为“0”。

## 18.2.3 SO锁存器

SO锁存器保持SDAAn引脚的输出电平。

## 18.2.4 唤醒控制电路

当设定在从属地址寄存器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)。

## 18.2.5 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。

## 18.2.6 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I<sup>2</sup>C中断请求。

- 第8个或者第9个串行时钟的下降（通过WTIMn位进行设定）
- 因检测到停止条件而产生中断请求（通过SPIEn位进行设定）。

备注: WTIMn位: IICA控制寄存器n0 (IICCTLn0) 的bit3

SPIEn位: IICA控制寄存器n0 (IICCTLn0) 的bit4

## 18.2.7 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。

## 18.2.8 串行时钟等待控制电路

此电路控制等待时序。

## 18.2.9 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。

## 18.2.10 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。

## 18.2.11 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”。

## 18.2.12 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。

## 18.2.13 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位设定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

## 18.3 控制串行接口IICA的寄存器

通过以下几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低电平宽度设定寄存器n (IICWLn)
- IICA高电平宽度设定寄存器n (IICWHn)
- 端口模式寄存器 (PMxx)
- 端口模式控制寄存器 (PMCxx)
- 端口复用功能配置寄存器 (PxxCFG)

备注: n=0

### 18.3.1 外围允许寄存器0 (PER0)

PER0寄存器是设定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IICA0时，必须将PER0.bit4 (IICA0EN) 置“1”。

通过8位存储器操作指令设定PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

寄存器详情请参考“4.3.6 外围允许寄存器0、1、2 (PER0、PER1、PER2)”。

注意1.要设定串行接口IICA时，必须先IICAnEN位为“1”的状态下设定以下的寄存器。当IICAnEN位为“0”时，串行接口IICAn的控制寄存器的值为初始值，忽视写操作（端口复用功能配置寄存器 (PxxCFG)、端口模式寄存器 (PMxx) 和端口模式控制寄存器 (PMCxx) 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低电平宽度设定寄存器n (IICWLn)
- IICA高电平宽度设定寄存器n (IICWHn)

备注： n=0

### 18.3.2 IICA控制寄存器n0 (IICCTLn0)

这是允许或者停止I<sup>2</sup>C运行、设定等待时序以及设定其他I<sup>2</sup>C运行的寄存器。

通过8位存储器操作指令设定IICCTLn0寄存器。但是，必须在IICEn位为“0”时或者在等待期间设定SPIEn位、WTIMn位和ACKEn位，而且在将IICEn位从“0”置为“1”时能同时设定这些位。

在产生复位信号后，此寄存器的值变为“00H”。

备注： n=0

图18-5 IICA控制寄存器n0 (IICCTLn0) 的格式(1/4)

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 <sup>2</sup> C运行的允许
0	停止运行。对IICA状态寄存器n (IICSn) 进行复位 <sup>注1</sup> , 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 <sup>注2, 3</sup>	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 <sup>注2, 3</sup>	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。

2.在IICEn位为“0”的状态下, 此位的信号无效。

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 (即IICCTLn1寄存器的DFCn=1) 时, 允许I<sup>2</sup>C运行 (IICEn=1), 就立即检测开始条件。此时, 必须在允许I<sup>2</sup>C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18-6 IICA控制寄存器n0 (IICCTLn0) 的格式(2/4)

SPIEn <sup>注1</sup>	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"> <li>•通过指令清除。</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

WTIMn <sup>注1</sup>	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, 与此位的设定无关, 在第9个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。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"> <li>•通过指令清除。</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

ACKEn <sup>注1、2</sup>	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"> <li>•通过指令清除。</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

- 注:
- 1.在IICEn位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。
  - 2.在地址传送过程中不是扩展码时, 设定值无效。当为从属设备并且地址匹配时, 与设定值无关而生成应答。

备注: n=0

图18-6 IICA控制寄存器n0 (IICCTLn0) 的格式(3/4)

STTn <sup>注1、2</sup>	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"> <li>•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。</li> <li>•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。</li> </ul>	
有关置位时序的注意事项： <ul style="list-style-type: none"> <li>•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”。</li> <li>•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”。</li> <li>•禁止与停止条件的触发（SPTn）同时置“1”。</li> <li>•在将STTn位置“1”后，禁止在满足清除条件前再次将此位“1”。</li> </ul>		
清除条件（STTn=0）		置位条件（STTn=1）
<ul style="list-style-type: none"> <li>•在禁止通信预约的状态下将STTn位置“1”。</li> <li>•在仲裁失败时</li> <li>•主控设备生成开始条件。</li> <li>•因LRELn位为“1”（退出通信）而进行的清除</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

注 1. 在IICEn位为“0”的状态下，此位的信号无效。

2. STTn位的读取值总是“0”。

备注：1. 如果在设定数据后读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18-6 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"> <li>•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”。</li> <li>•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”。</li> <li>•禁止与开始条件的触发（STTn）同时置“1”。</li> <li>•只有在主控设备的情况下才能将SPTn位置“1”。</li> <li>•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”。</li> <li>•在将SPTn位置“1”后，禁止在满足清除条件前再次将此位置“1”。</li> </ul>		
清除条件（SPTn=0）		置位条件（SPTn=1）
<ul style="list-style-type: none"> <li>•当仲裁失败时</li> <li>•在检测到停止条件后自动清除。</li> <li>•因LRELn位为“1”（退出通信）而进行的清除</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

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

### 18.3.3 IICA状态寄存器n (IICSn)

这是表示I<sup>2</sup>C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18-6 IICA状态寄存器n (IICSn) 的格式(1/3)

复位后: 00H

符号	7	6	5	4	3	2	1	0
IICSn	MSTS <sub>n</sub>	ALD <sub>n</sub>	EXC <sub>n</sub>	COI <sub>n</sub>	TRC <sub>n</sub>	ACK <sub>n</sub>	STD <sub>n</sub>	SPD <sub>n</sub>

MSTS <sub>n</sub>	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 <sub>n</sub> =0)	置位条件 (MSTS <sub>n</sub> =1)
<ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>当ALD<sub>n</sub>位为“1”（仲裁失败）时</li> <li>因LREL<sub>n</sub>位为“1”（退出通信）而进行的清除</li> <li>当IICEn位从“1”变为“0”（停止运行）时</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>当生成开始条件时</li> </ul>

ALD <sub>n</sub>	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 <sub>n</sub> 位。
清除条件 (ALD <sub>n</sub> =0)	置位条件 (ALD <sub>n</sub> =1)
<ul style="list-style-type: none"> <li>在读IICSn寄存器后自动清除注。</li> <li>当IICEn位从“1”变为“0”（停止运行）时</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>当仲裁失败时</li> </ul>

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<sub>n</sub>位时，必须在读其他位前先读ALD<sub>n</sub>位的数据。

备注：1. LREL<sub>n</sub>: 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18-7 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"> <li>当检测到开始条件时</li> <li>当检测到停止条件时</li> <li>因LRELn位为“1”（退出通信）而进行的清除</li> <li>当IICEn位从“1”变为“0”（停止运行）时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>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）</li> </ul>
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"> <li>当检测到开始条件时</li> <li>当检测到停止条件时</li> <li>因LRELn位为“1”（退出通信）而进行的清除</li> <li>当IICEn位从“1”变为“0”（停止运行）时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>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）</li> </ul>
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。设定为能将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"> <li>当检测到停止条件时</li> <li>因LRELn位为“1”（退出通信）而进行的清除</li> <li>当IICEn位从“1”变为“0”（停止运行）时</li> <li>因WRELn位为“1”（解除等待）而进行的清除<sup>注</sup></li> <li>当ALDn位从“0”变为“1”（仲裁失败）时</li> <li>当复位时</li> <li>不参加通信的情况 (MSTSn、EXCn、COIn=0)</li> </ul> <主控设备> <ul style="list-style-type: none"> <li>当第1字节的LSB（传送方向指示位）输出“1”时</li> </ul> <从属设备> <ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当第1字节的LSB（传送方向指示位）输入“0”时</li> </ul>		<主控设备> <ul style="list-style-type: none"> <li>当生成开始条件时</li> <li>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</li> </ul> <从属设备> <ul style="list-style-type: none"> <li>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</li> </ul>

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18-7 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"> <li>•当检测到停止条件时</li> <li>•当下一个字节的第1个时钟上升时</li> <li>•因LRELn位为“1” (退出通信) 而进行的清除</li> <li>•当IICEn位从“1”变为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时</li> </ul>
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"> <li>•当检测到停止条件时</li> <li>•在地址传送后的下一个字节的第1个时钟上升时</li> <li>•因LRELn位为“1” (退出通信) 而进行的清除</li> <li>•当IICEn位从“1”变为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•当检测到开始条件时</li> </ul>
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"> <li>•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</li> <li>•当WUPn位从“1”变为“0”时</li> <li>•当IICEn位从“1”变为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•当检测到停止条件时</li> </ul>

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

### 18.3.4 IICA标志寄存器n (IICFn)

这是设定I<sup>2</sup>C运行模式以及表示I<sup>2</sup>C总线状态的寄存器。

通过8位存储器操作指令设定IICFn寄存器。但是，只能读STTn清除标志 (STCFn) 和I<sup>2</sup>C总线状态标志 (IICBSYn)

。通过IICRSVn位设定允许或者禁止通信预约功能，并且通过STCENn位设定IICBSYn位的初始值。只有在禁止I<sup>2</sup>C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”。

Draft Version  
Cmsemicon Confidential

图18-7 IICA标志寄存器n (IICFn) 的格式

复位后: 00H

RW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"> <li>•因STTn位为“1”而进行的清除</li> <li>•当IICEn位为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>	<ul style="list-style-type: none"> <li>•在设定为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将STTn位清“0”时</li> </ul>

IICBSYn	I <sup>2</sup> C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"> <li>•当检测到停止条件时</li> <li>•当IICEn位为“0” (停止运行) 时</li> <li>•当复位时</li> </ul>	<ul style="list-style-type: none"> <li>•当检测到开始条件时</li> <li>•STCENn位为“0”时的IICEn位的置位</li> </ul>

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"> <li>•通过指令清除。</li> <li>•当检测到开始条件时</li> <li>•当复位时</li> </ul>	<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

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"> <li>•通过指令清除。</li> <li>•当复位时</li> </ul>	<ul style="list-style-type: none"> <li>•通过指令置位。</li> </ul>

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

### 18.3.5 IICA控制寄存器n1 (IICCTLn1)

这是用于设定I<sup>2</sup>C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定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I<sup>2</sup>C运行（IICA控制寄存器n0（IICCTLn0）的bit7（IICEn）=0）时设定IICCTLn1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图18-8 IICA控制寄存器n1 (IICCTLn1) 的格式(1/2)

复位后: 00H

符号	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

R/W<sup>注1</sup>

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。

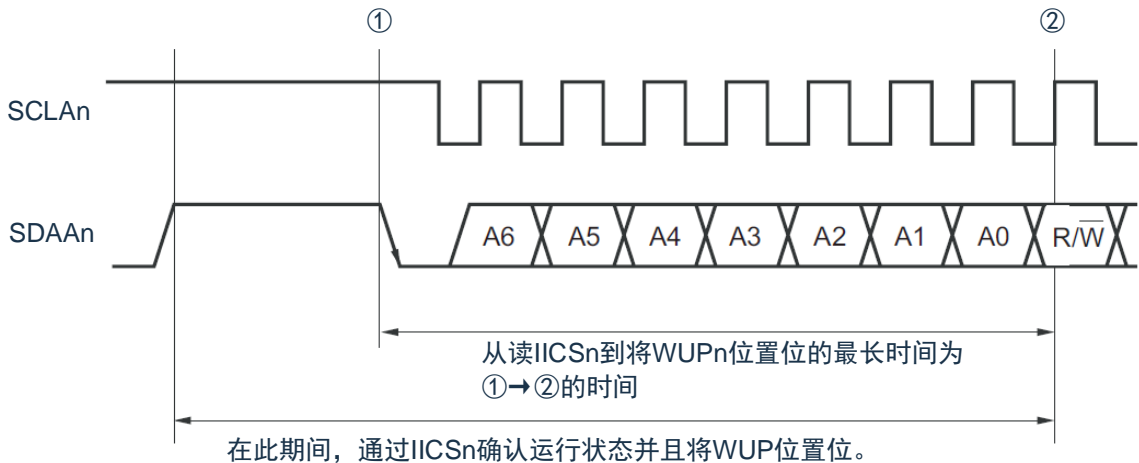
要通过将WUPn位置“1”来转移到深度睡眠模式时，必须在将WUPn位置“1”后至少经过3个f<sub>MCK</sub>时钟，然后执行深度睡眠指令（参照“图18-20将WUPn位置“1”时的流程”）。在地址匹配或者接收到扩展码后，必须将WUPn位清“0”。能通过将WUPn位清“0”来参加后续的通信（需要在将WUPn位清“0”后解除等待以及写发送数据）。

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。

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）） <sup>注2</sup> 。

注： 1.bit4和bit5是只读位。

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18-8 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）
<ul style="list-style-type: none"> <li>•当SCLAn引脚为低电平时</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•当SCLAn引脚为高电平时</li> </ul>
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）
<ul style="list-style-type: none"> <li>•当SDAAn引脚为低电平时</li> <li>•当IICEn位为“0”（停止运行）时</li> <li>•当复位时</li> </ul>		<ul style="list-style-type: none"> <li>•当SDAAn引脚为高电平时</li> </ul>
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 <sub>MCK</sub> ）的控制	
0	选择f <sub>CLK</sub> （1MHz≤f <sub>CLK</sub> ≤20MHz）。	
1	选择f <sub>CLK</sub> /2（20MHz<f <sub>CLK</sub> ）。	

注意：1.IICA运行时钟（f<sub>MCK</sub>）的最大工作频率为20MHz(Max.)。只有在f<sub>CLK</sub>超过20MHz时才必须将IICA控制寄存器n1（IICCTLn1）的bit0（PRSn）置“1”。

2.在设定传送时钟的情况下，必须注意f<sub>CLK</sub>的最小工作频率。串行接口IICA的f<sub>CLK</sub>最小工作频率取决于运行模式。

快速模式： f<sub>CLK</sub>=3.5MHz(Min.)

增强型快速模式： f<sub>CLK</sub>=10MHz(Min.)

标准模式： f<sub>CLK</sub>=1MHz(Min.)

备注：1.IICEn：IICA控制寄存器n0（IICCTLn0）的bit7

2.n=0



### 18.3.6 IICA低电平宽度设定寄存器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 ( $t_{LOW}$ ) 和SDAAn引脚信号。

通过8位存储器操作指令设定IICWLn寄存器。

必须在禁止I<sup>2</sup>C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设定IICWLn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

有关IICWLn寄存器的设定方法, 请参照“18.4.2通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法”。

数据保持时间为IICWLn所设时间的1/4。

图18-9 IICA低电平宽度设定寄存器n (IICWLn) 的格式

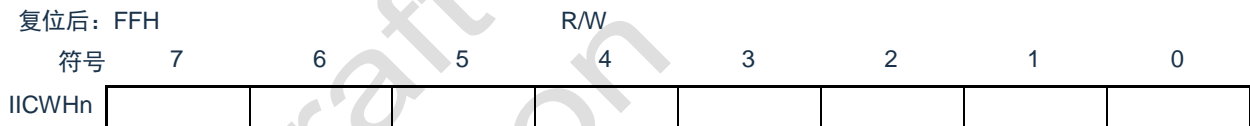


### 18.3.7 IICA高电平宽度设定寄存器n (IICWHn)

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定IICWHn寄存器。

必须在禁止I<sup>2</sup>C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设定IICWHn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

图18-10 IICA高电平宽度设定寄存器n (IICWHn) 的格式



备注: 1.有关主控方传送时钟的设定方法, 请参照18.4.2(1); 有关从属方IICWLn寄存器和IICWHn寄存器的设定方法, 请参照18.4.2(2)。

2.n=0

### 18.3.8 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/SCLAn引脚用作时钟输入/输出并且将Pxx/SDAAn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/SCLAn引脚和Pxx/SDAAn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）”。

## 18.4 I<sup>2</sup>C总线模式的功能

### 18.4.1 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

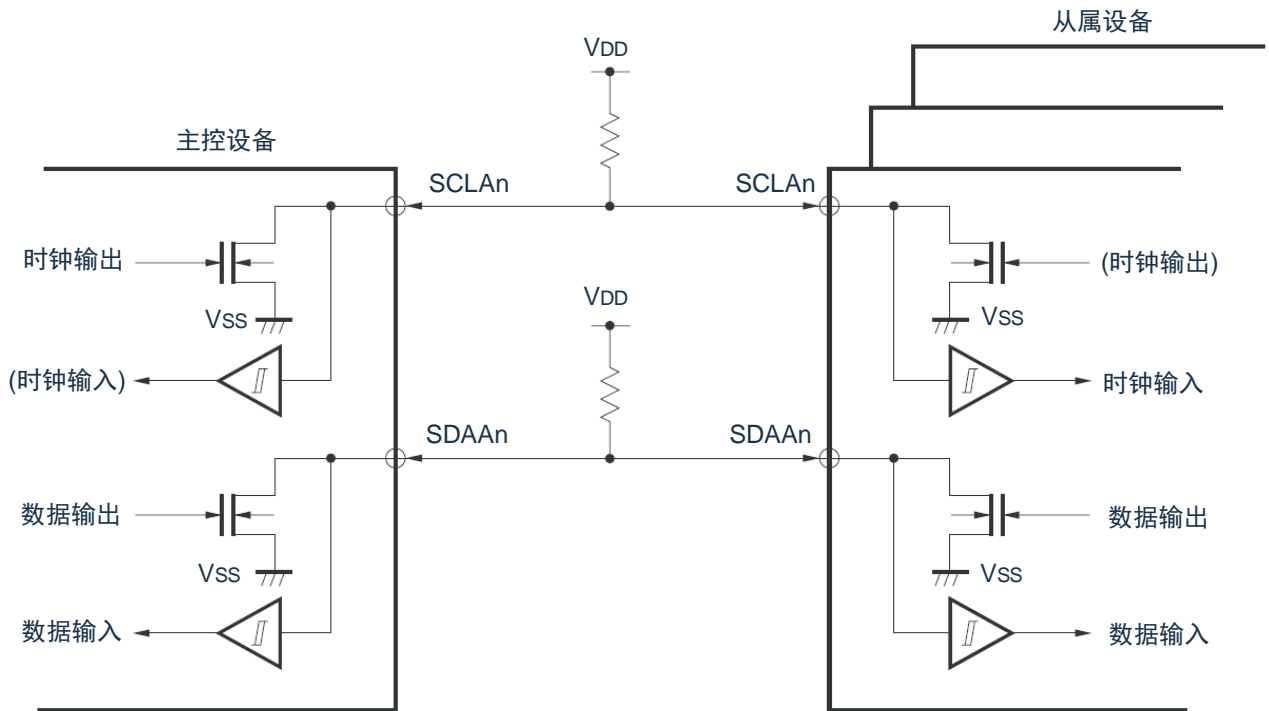
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

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18-11 引脚结构图



备注：n=0

## 18.4.2 通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法

### (1) 主控方传送时钟的设定方法

$$\text{传送时钟} = \frac{f_{MCK}}{IICWLn + IICWHn + f_{MCK}(t_R + t_F)}$$

此时，IICWLn寄存器和IICWHn寄存器的最佳设定值如下：  
(全部设定值的小数部分都舍入)

- 快速模式

$$IICWLn = \frac{0.52}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.48}{\text{传送时钟}} \times t_R - t_F \right) \times f_{MCK}$$

- 标准模式

$$IICWLn = \frac{0.47}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.53}{\text{传送时钟}} \times t_R - t_F \right) \times f_{MCK}$$

- 增强型快速模式

$$IICWLn = \frac{0.50}{\text{传送时钟}} \times f_{MCK}$$

$$IICWHn = \left( \frac{0.50}{\text{传送时钟}} \times t_R - t_F \right) \times f_{MCK}$$

### (2) 从属方IICWLn寄存器和IICWHn寄存器的设定方法

(全部设定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：1. 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”。

2. 在设定传送时钟的情况下，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低电平宽度设定寄存器n

IICWHn：IICA高电平宽度设定寄存器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.  $n=0$

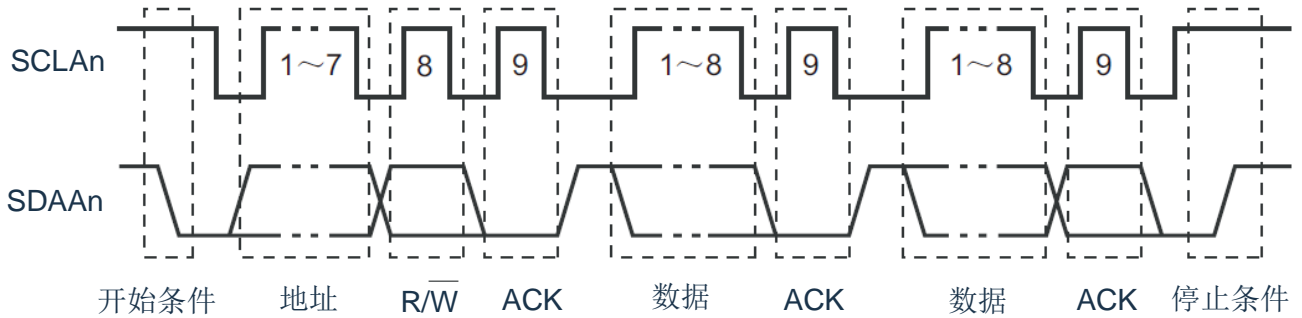
Draft Version  
Cmsemicon Confidential

## 18.5 I<sup>2</sup>C总线的定义和控制方法

以下说明I<sup>2</sup>C总线的串行数据通信格式和使用的信号。

I<sup>2</sup>C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”的各传送时序如下图所示。

图18-12 I<sup>2</sup>C总线的串行数据传送时序



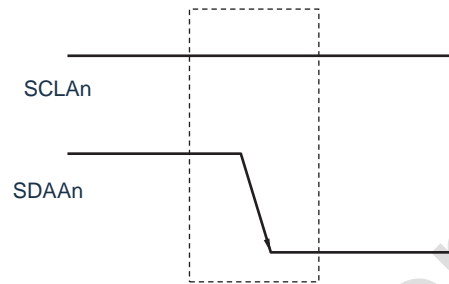
主控设备生成开始条件、从属地址和停止条件。

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。

## 18.5.1 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18-13 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

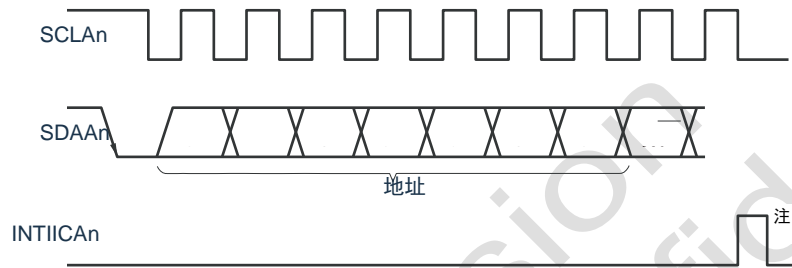
## 18.5.2 地址

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定完全不同的地址。

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，在主控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18-14 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。

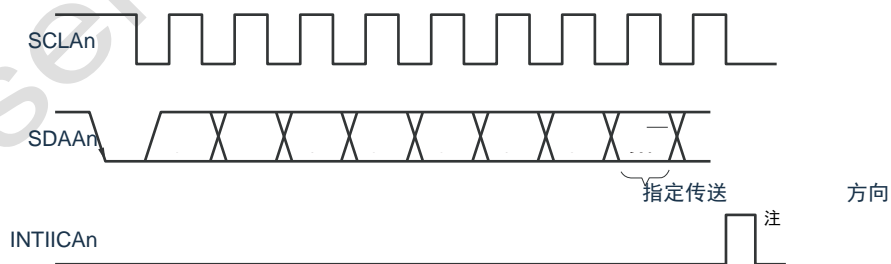
如果将从属地址和“18.5.3 传送方向的指定”中说明的传送方向构成的8位数据写到IICA移位寄存器n（IICAn），就输出地址。接收到的地址被写到IICAn寄存器。从属地址分配在IICAn寄存器的高7位。

## 18.5.3 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18-15 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



## 18.5.4 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。在主控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，主控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。

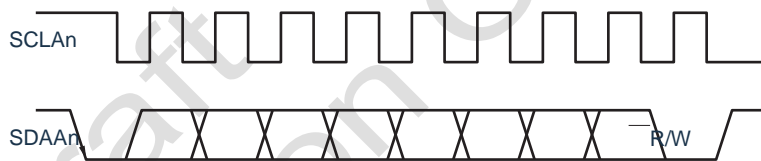
接收方在第9个时钟将SDAAn线置为低电平，生成应答 (正常接收)。

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位数据设定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18-16 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定而不同，如下所示。

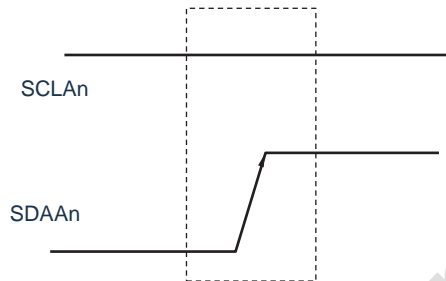
- 当选择8个时钟的等待时 (IICCTLn0寄存器的bit3 (WTIMn) =0)：通过在解除等待前将ACKEn位置“1”，与SCLAn引脚的第8个时钟下降沿同步生成应答。
- 当选择9个时钟的等待时 (IICCTLn0寄存器的bit3 (WTIMn) =1)：通过事先将ACKEn位置“1”，生成应答。

备注：n=0

## 18.5.5 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18-17 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

## 18.5.6 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18-18 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)

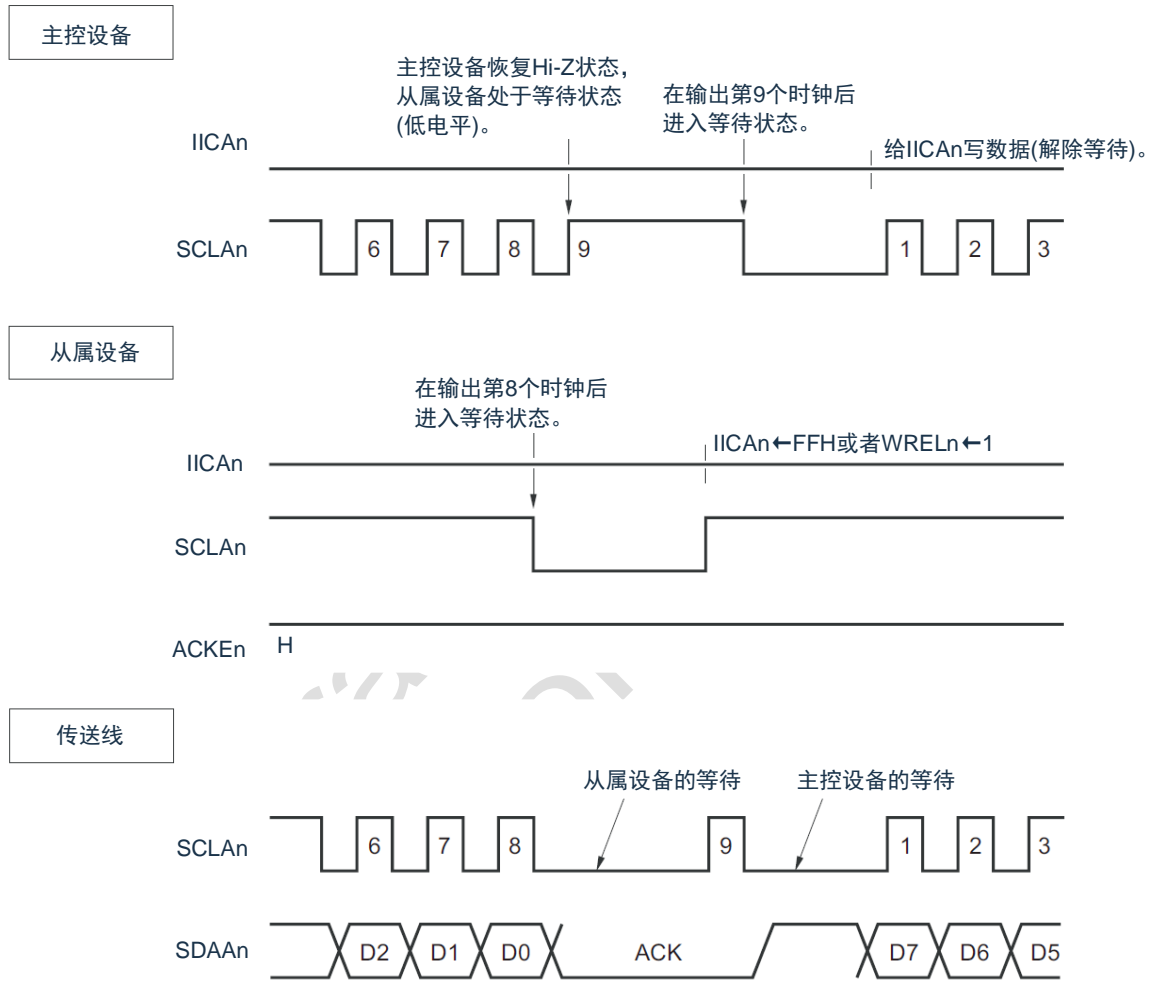
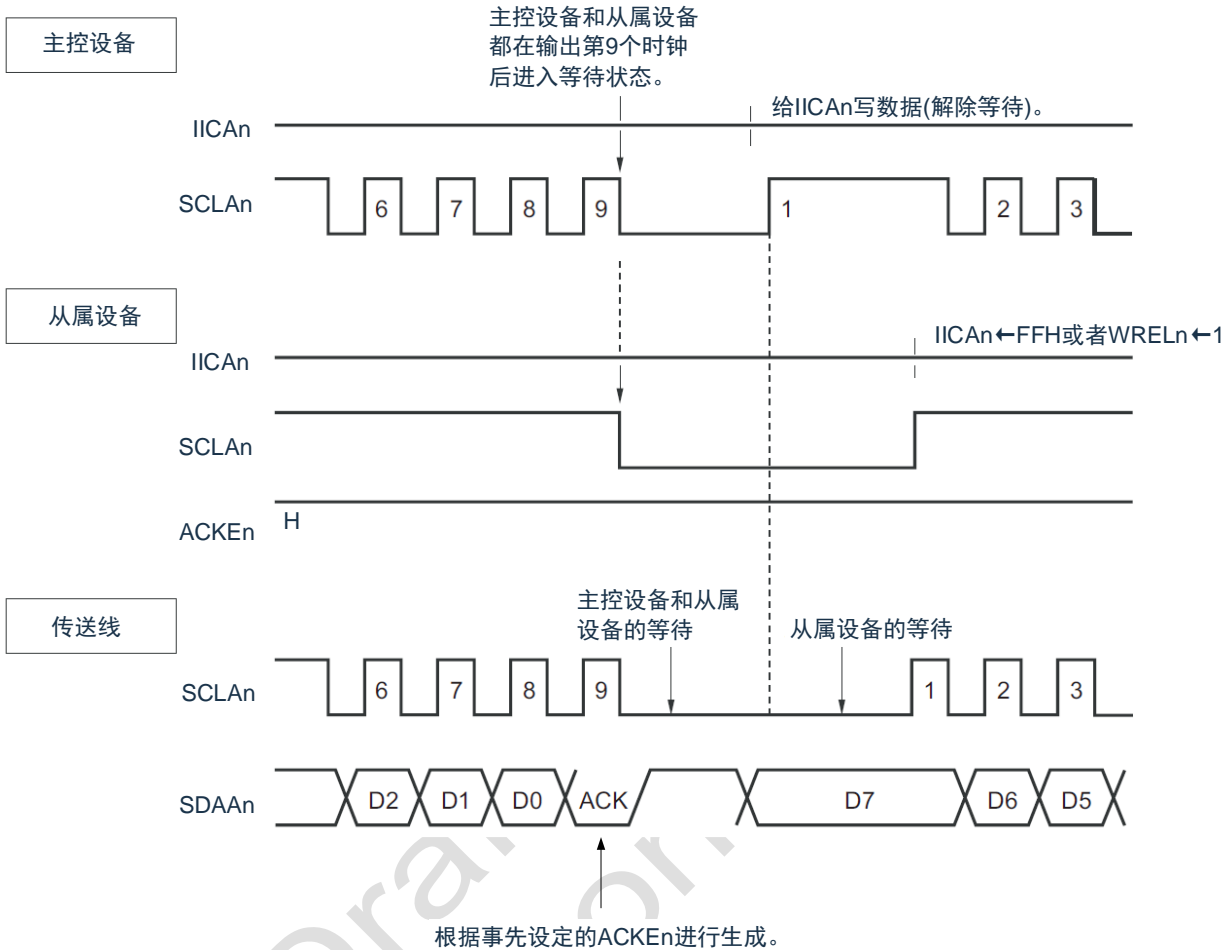


图18-19 等待(2/2)

(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

通过设定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

## 18.5.7 等待的解除方法

在一般情况下，I<sup>2</sup>C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，I<sup>2</sup>C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I<sup>2</sup>C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

## 18.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设定IICA控制寄存器n0（IICCTLn0）的bit3（WTIMn），在表18-2所示的时序产生INTIICAn并且进行等待控制。

表18-2 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 <sup>注1、2</sup>	8 <sup>注2</sup>	8 <sup>注2</sup>	9	8	8
1	9 <sup>注1、2</sup>	9 <sup>注2</sup>	9 <sup>注2</sup>	9	9	9

注：1.只有在接收的地址和从属地址寄存器n（SVAn）的设定地址相同时，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）的设定无关，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。

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)。

## 18.5.9 地址匹配的检测方法

在I<sup>2</sup>C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) 的设定地址相同或者只接收到扩展码时，产生INTIICAn中断请求。

### 18.5.10 错误的检测

在I<sup>2</sup>C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

## 18.5.11 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寄存器的设定值为“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18-3

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I<sup>2</sup>C总线规格书。

2.n=0



## 18.5.12 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。

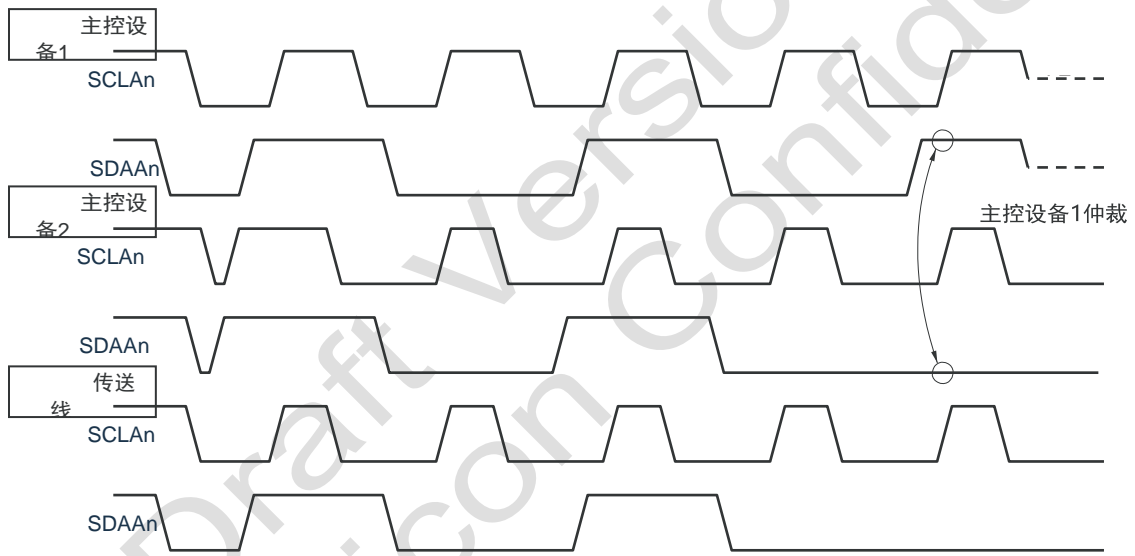
在发生下一次中断请求时（例如：在第8或者第9个时钟检测到停止条件），用软件通过ALDn位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“14.5.8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18-19 仲裁时序例子



备注：n=0

表18-4 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 <sup>注1</sup>
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) <sup>注2</sup>
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 <sup>注1</sup>
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) <sup>注2</sup>
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 <sup>注1</sup>
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。

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

### 18.5.13 唤醒功能

这是I<sup>2</sup>C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。

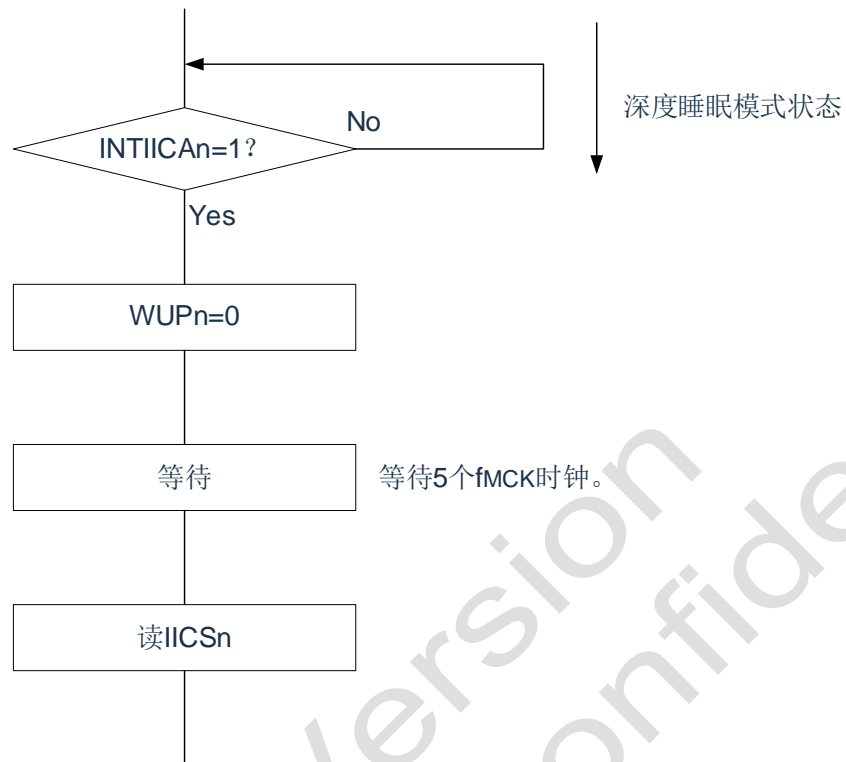
要在深度睡眠模式中使用唤醒功能时，必须将WUPn位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）。在产生此中断后，通过指令将WUPn位清“0”，返回到通常运行。

将WUPn位置“1”时的流程如图18-20所示，通过地址匹配将WUPn位置“0”时的流程如图18-21所示。

图18-20 将WUPn位置“1”时的流程



图18-21 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。

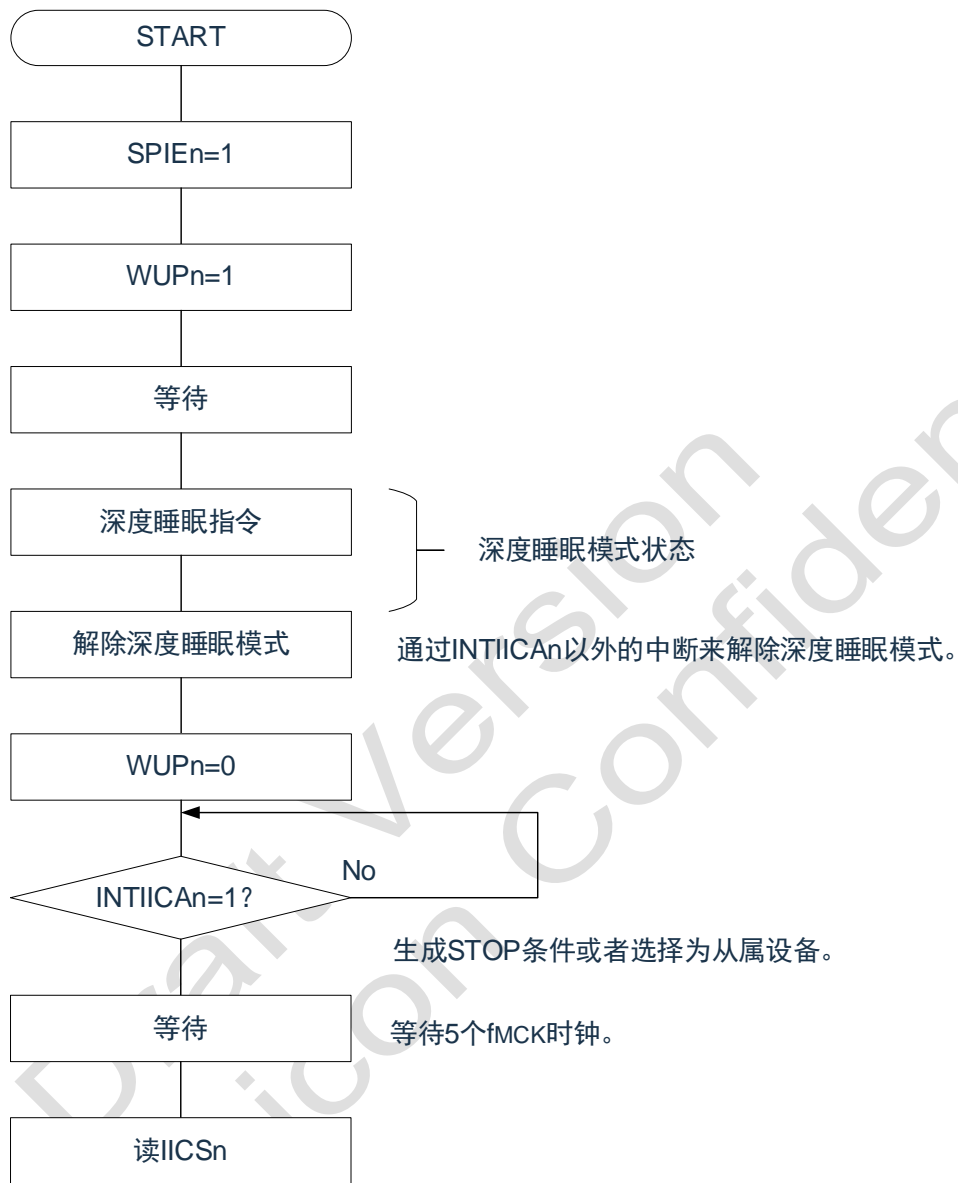
- 下一次IIC通信为主控设备运行的情况：图18-22的流程
- 下一次IIC通信为从属设备运行的情况：

通过INTIICAn中断返回的情况：和图18-21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18-22 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

## 18.5.14 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MSTS<sub>n</sub>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:

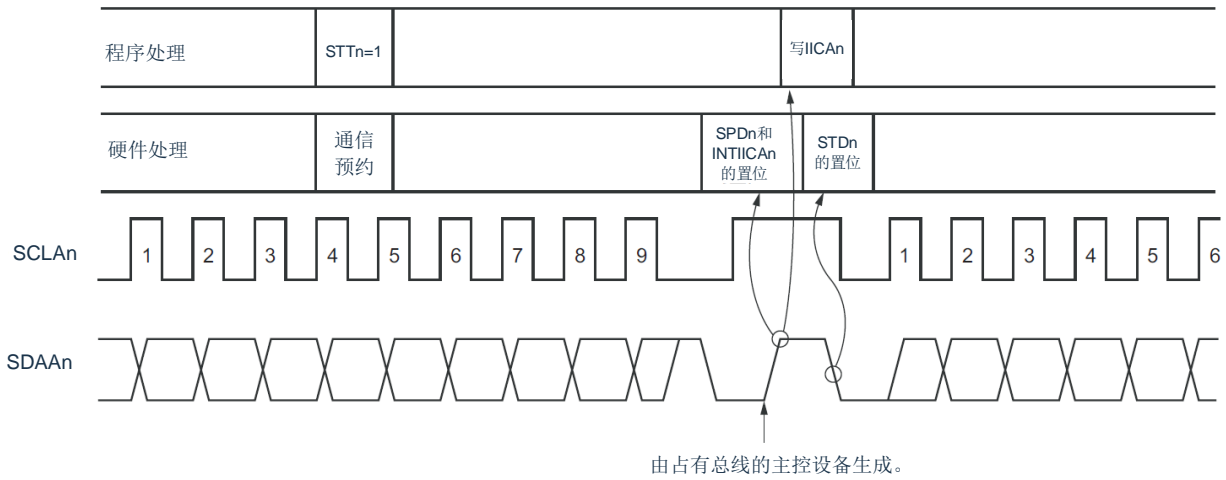
将从STTn 位置“1” 到确认MSTS<sub>n</sub> 标志为止的等待时间:  

$$(\text{IICWLn 的设定值} + \text{IICWHn 的设定值} + 4) / f_{\text{MCK}} + t_{\text{F}} \times 2$$

- 备注: 1. IICWLn: IICA低电平宽度设定寄存器n  
 IICWHn: IICA高电平宽度设定寄存器n  
 $t_{\text{F}}$ : SDAAn信号和SCLAn信号的下降时间  
 $f_{\text{MCK}}$ : IICA运行时钟频率  
 2. n=0

通信预约的时序如下图所示。

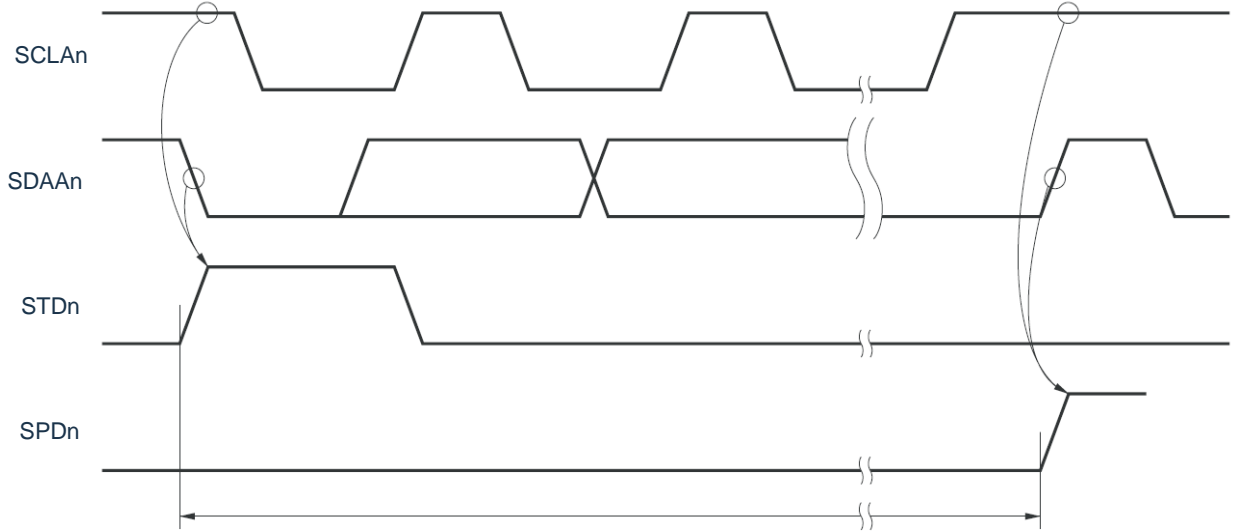
图18-23 通信预约的时序



- 备注：
- IICAn : IICA移位寄存器n
  - 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18-24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18-24 通信预约的接受时序

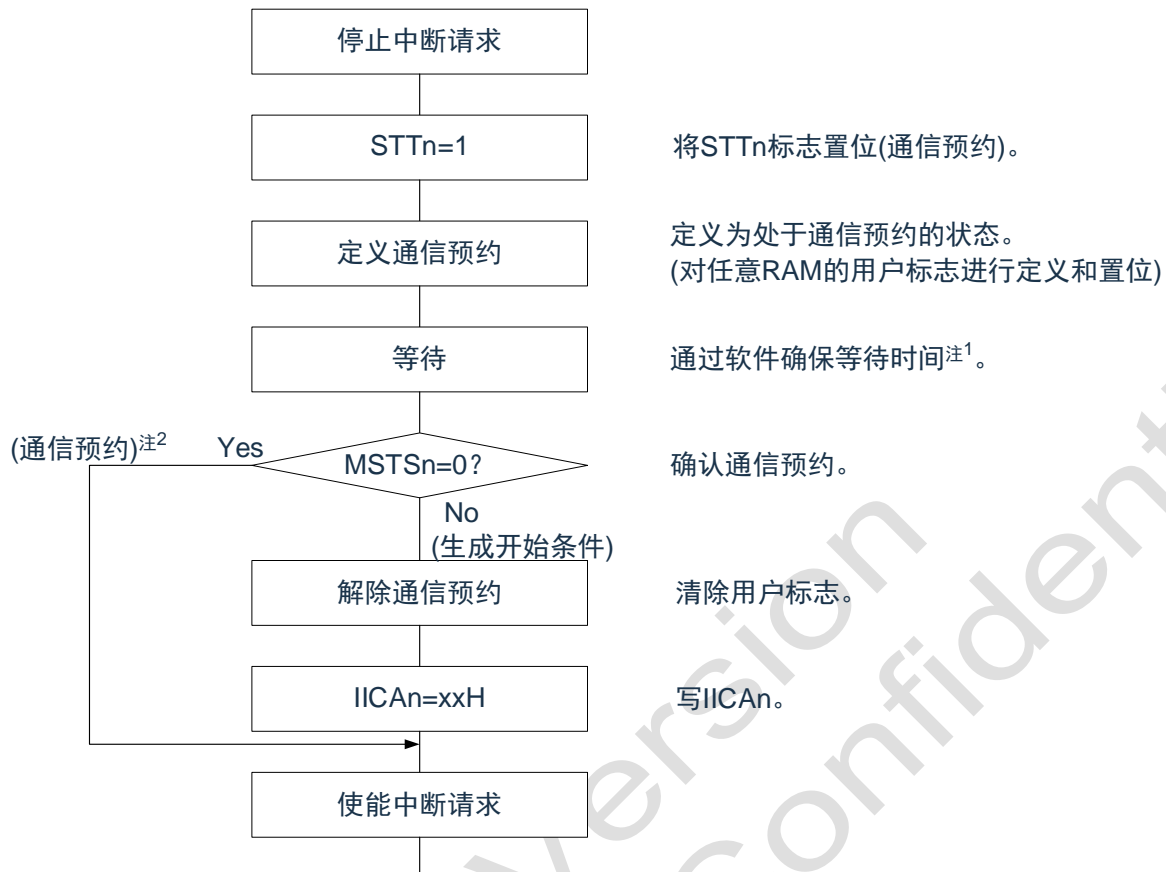


待机状态(在此期间，能通过将STTn位置“1”进行通信预约)

通信预约的步骤如图18-25所示。

备注：n=0

图18-25 通信预约的步骤



注：1. 等待时间如下： $(IICWLn的设定值+IICWHn的设定值+4)/f_{MCK}+t_F$  2

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低电平宽度设定寄存器n

IICWHn: IICA高电平宽度设定寄存器n

$t_F$  : SDAAn信号和SCLAn信号的下降时间

$f_{MC}$ : 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



## 18.5.15 其他注意事项

### (1) STCENn位为“0”的情况

在刚允许I<sup>2</sup>C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 设定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I<sup>2</sup>C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I<sup>2</sup>C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I<sup>2</sup>C运行而且中途参加通信, I<sup>2</sup>C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I<sup>2</sup>C通信。为了避免这种情况, 必须按照以下顺序启动I<sup>2</sup>C:

- 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I<sup>2</sup>C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

## 18.5.16 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“初始设定”和“通信处理”。在启动时执行“初始设定”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“初始设定”、“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“初始设定”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。

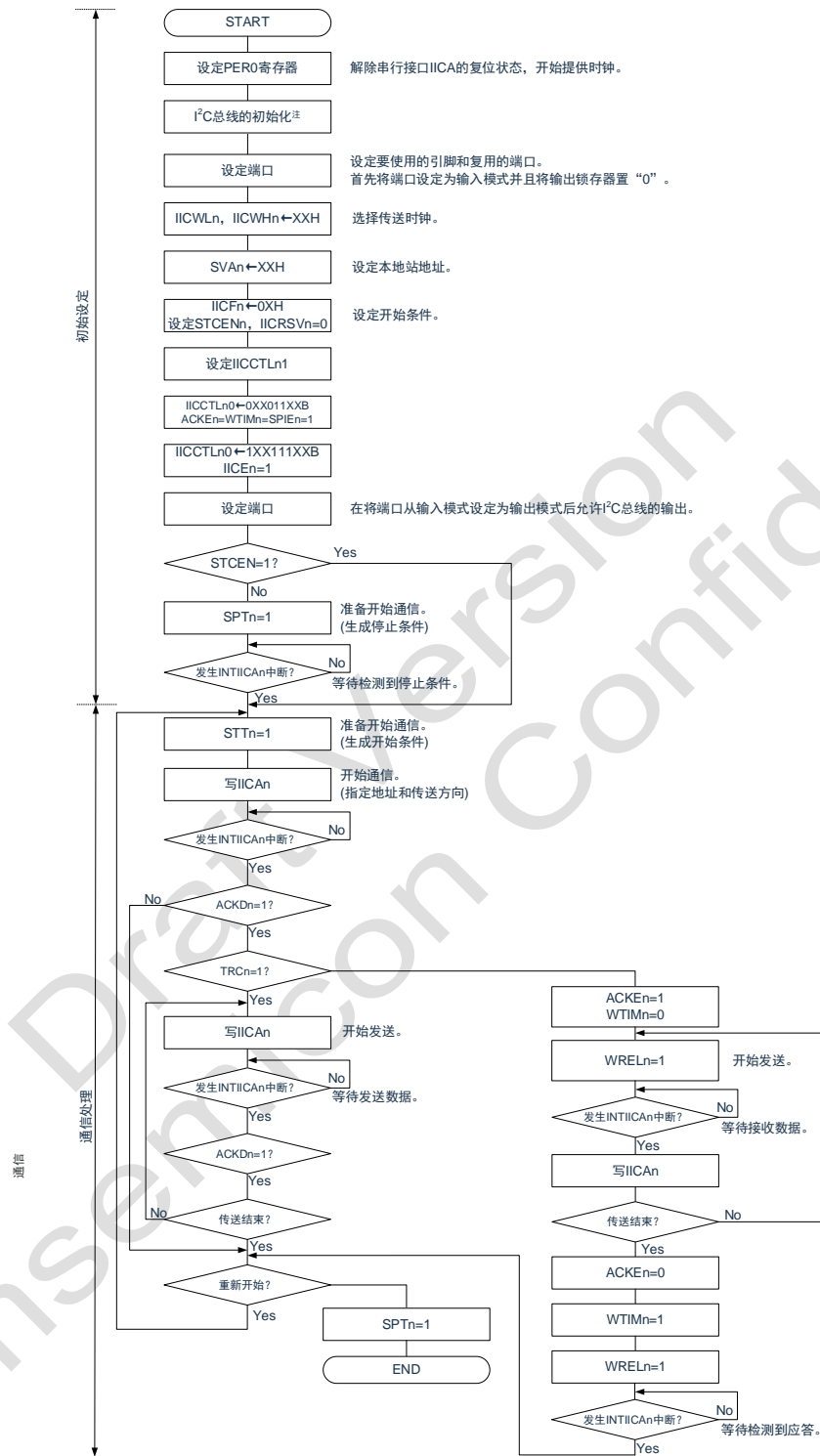
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待INTIICAn中断的发生。如果发生INTIICAn中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注 n=0

(1) 单主控系统的主控运行

图18-26 单主控系统的主控运行



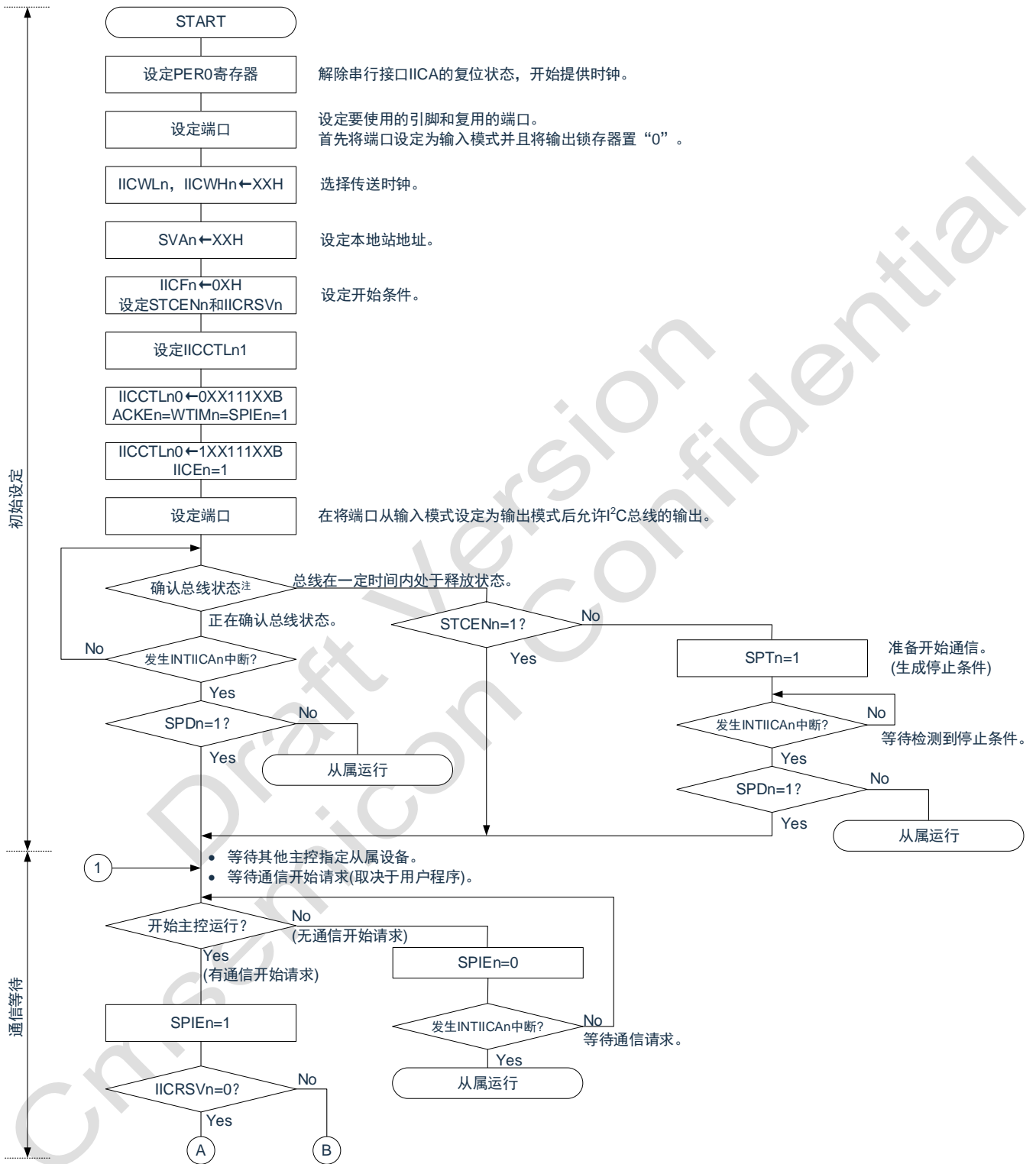
注： 必须根据通信中的产品的规格，释放I<sup>2</sup>C总线（SCLAn引脚和SDAAn引脚为高电平）。例如，如果EEPROM处于给SDAAn引脚输出低电平的状态，就必须将SCLAn引脚设定为输出端口，并且在SDAAn引脚固定为高电平前从输出端口输出时钟脉冲。

备注： 1.发送和接收的格式必须符合通信中的产品的规格。

2.n=0

(2) 多主控系统的主控运行

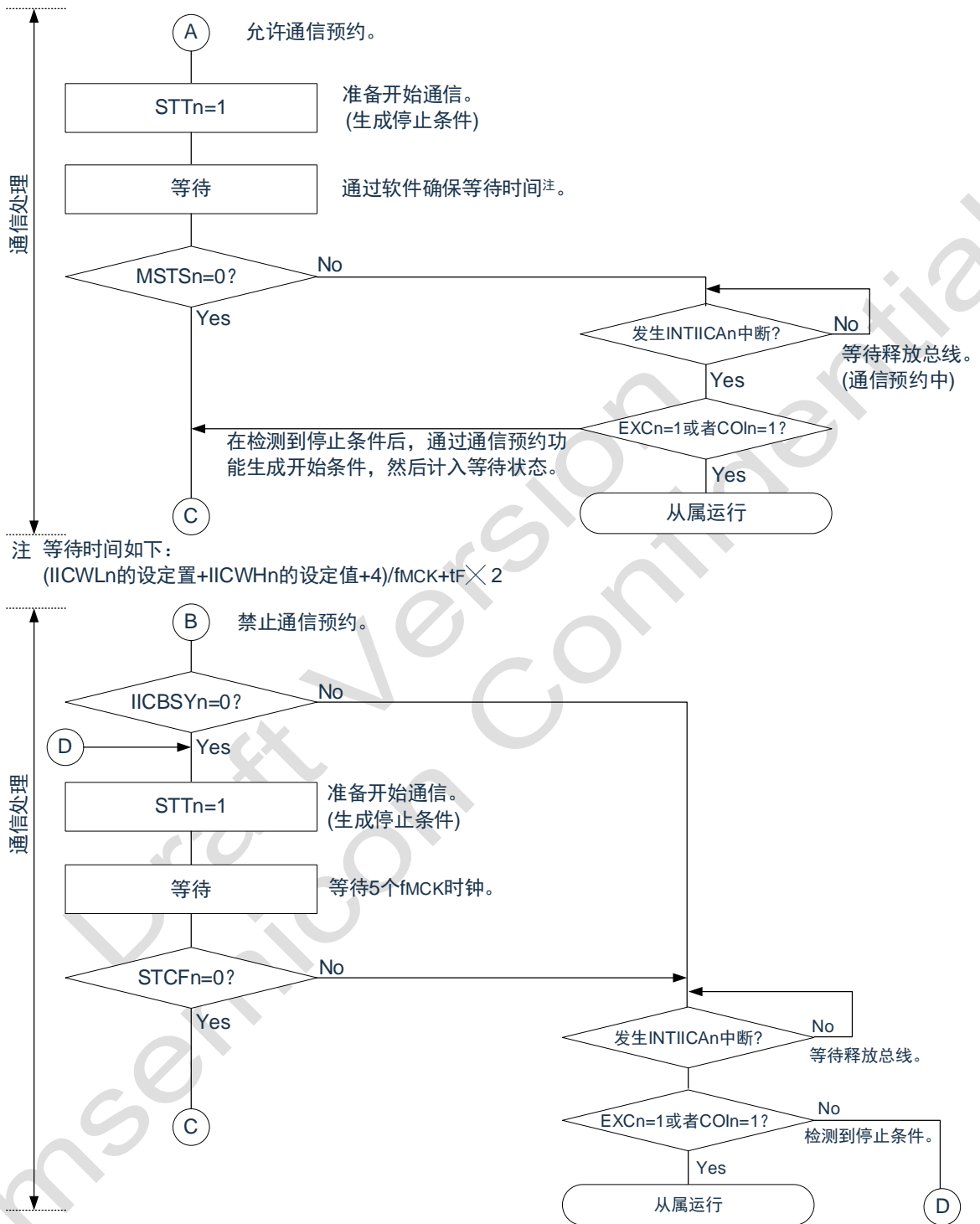
图18-27 多主控系统的主控运行(1/3)



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I<sup>2</sup>C总线(SCLAn引脚和SDAAn引脚为高电平)。

备注: n=0

图18-28 多主控系统的主控运行(2/3)



备注: 1. IICWLn : IICA低电平宽度设定寄存器n

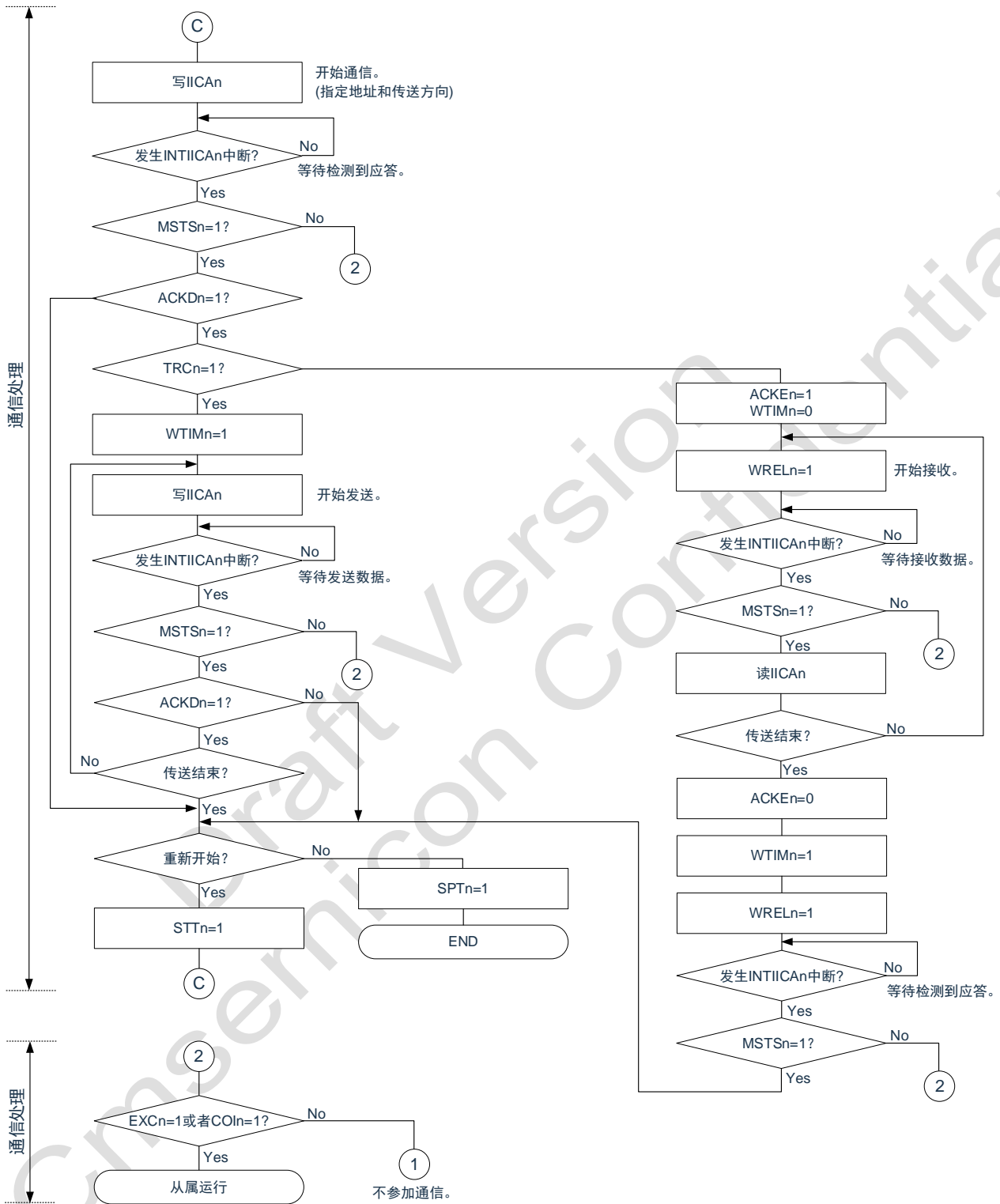
IICWHn : IICA高电平宽度设定寄存器n

t<sub>F</sub> : SDAAn信号和SCLAn信号的下降时间

f<sub>MCK</sub> : IICA运行时钟频率

2. n=0

图18-28 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。

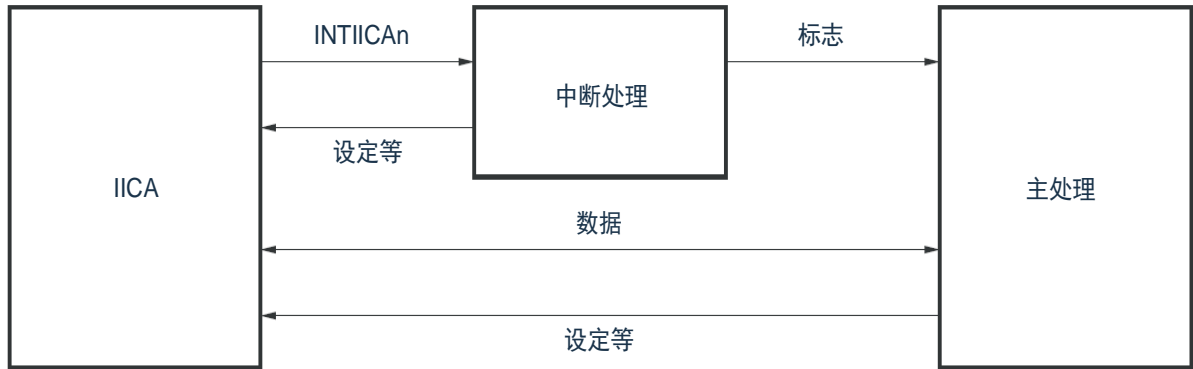
4. n=0

(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。

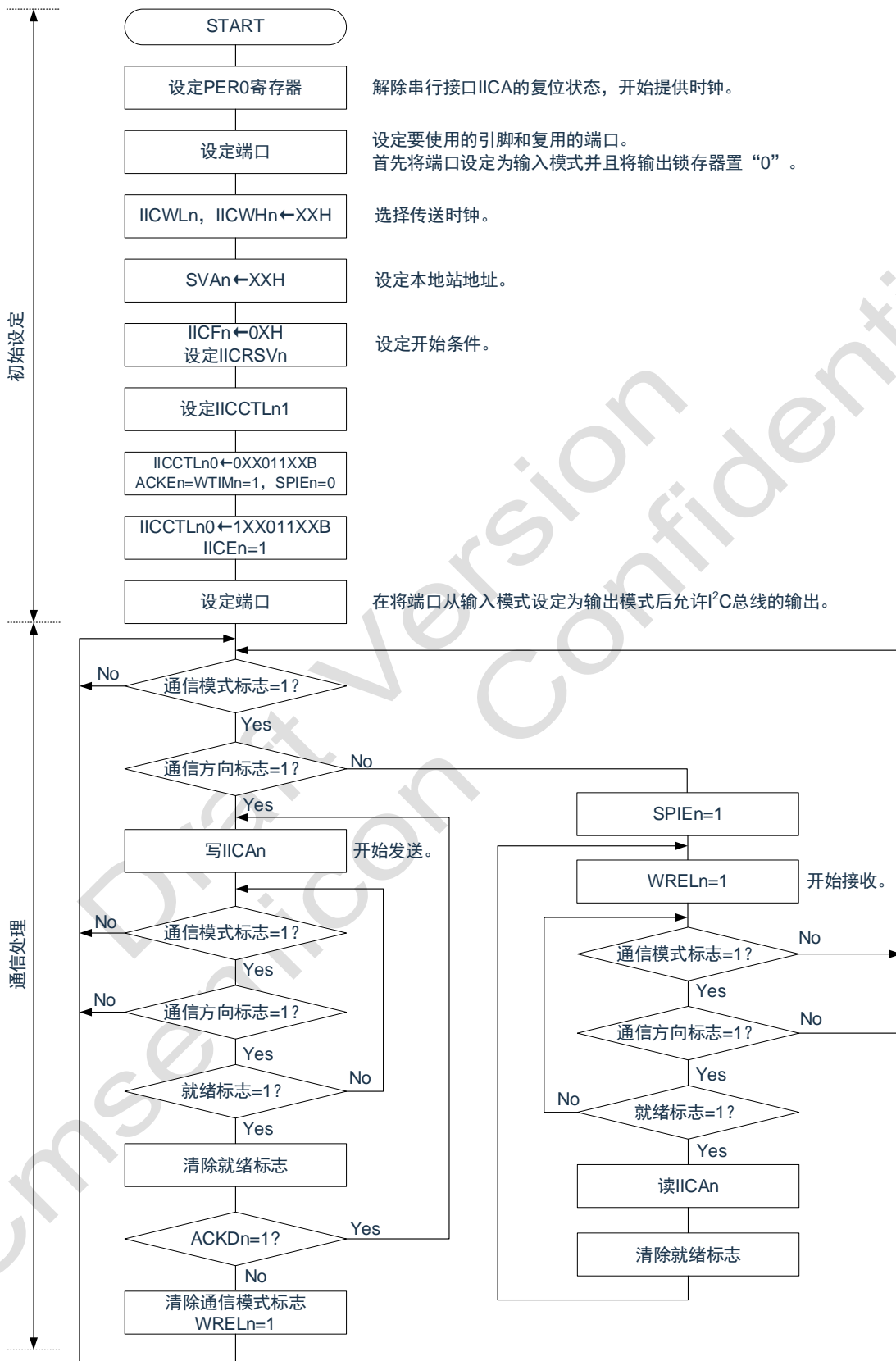
备注 n=0

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18-28 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

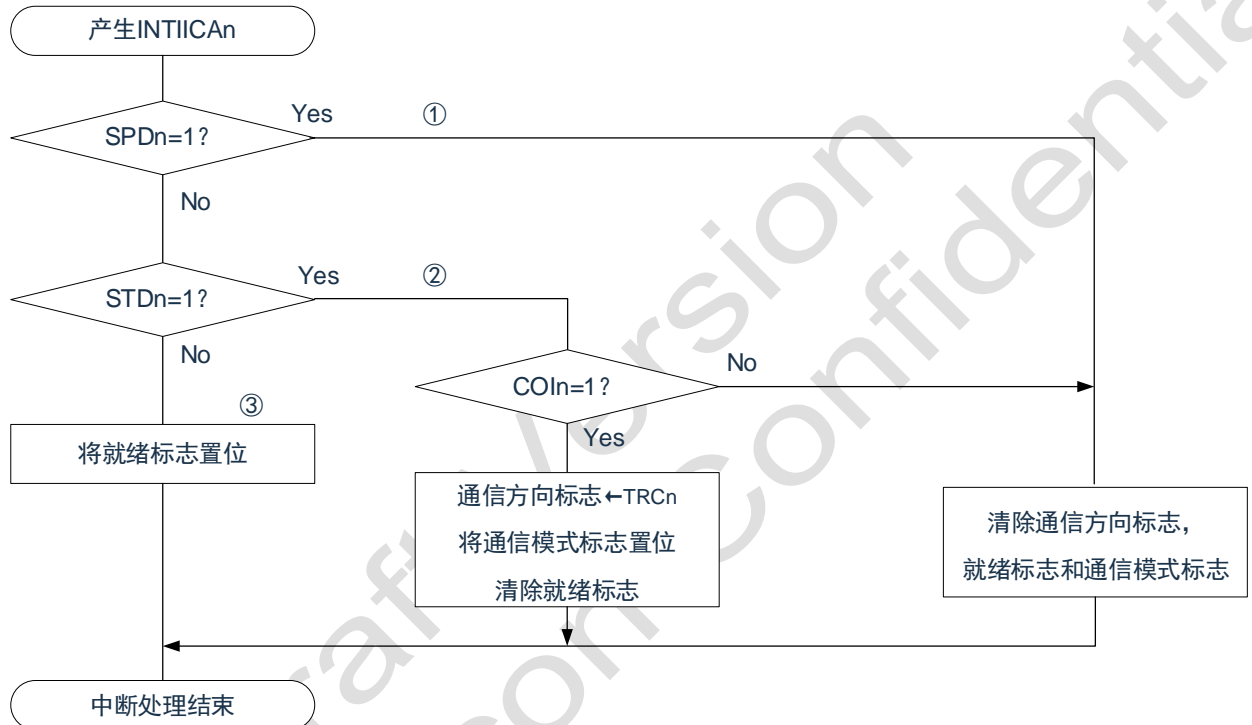


从属设备通过INTIICAn中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过INTIICAn中断确认状态并且进行以下处理。

- 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，就设定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，I<sup>2</sup>C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18-29 从属运行步骤(2)”的①~③。

图18-29 从属运行步骤(2)



备注：n=0

## 18.5.17 I<sup>2</sup>C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

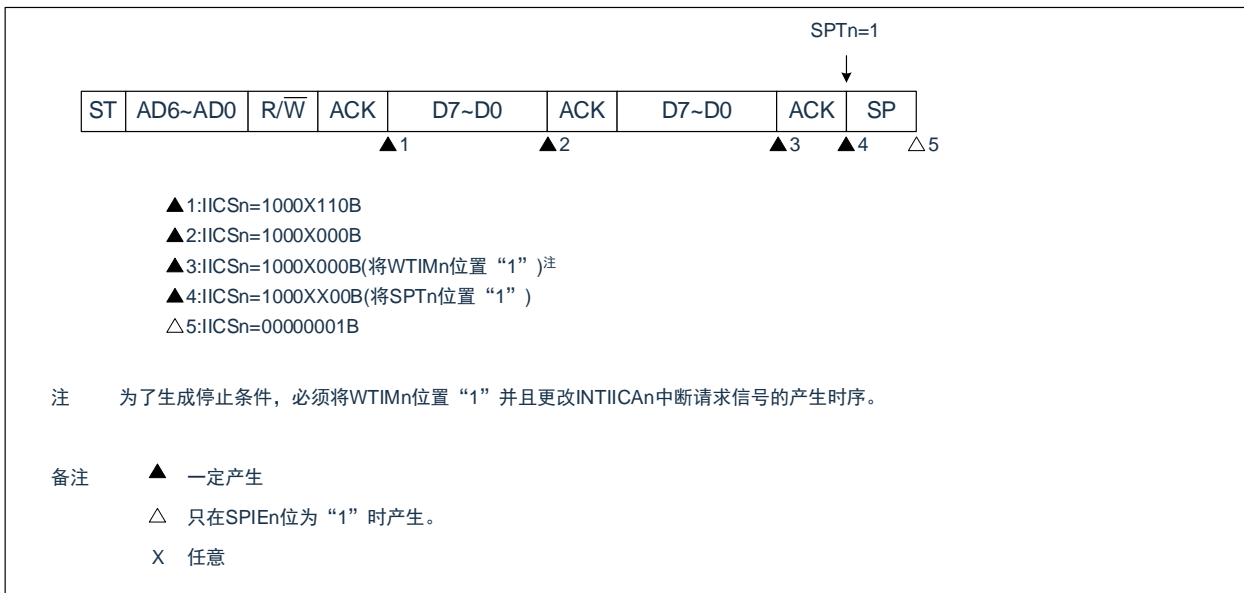
2. n=0

Draft Version  
Cmsemicon Confidential

(1) 主控运行

(a) Start~Address~Data~Data~Stop (发送和接收)

(i) WTIMn=0的情况



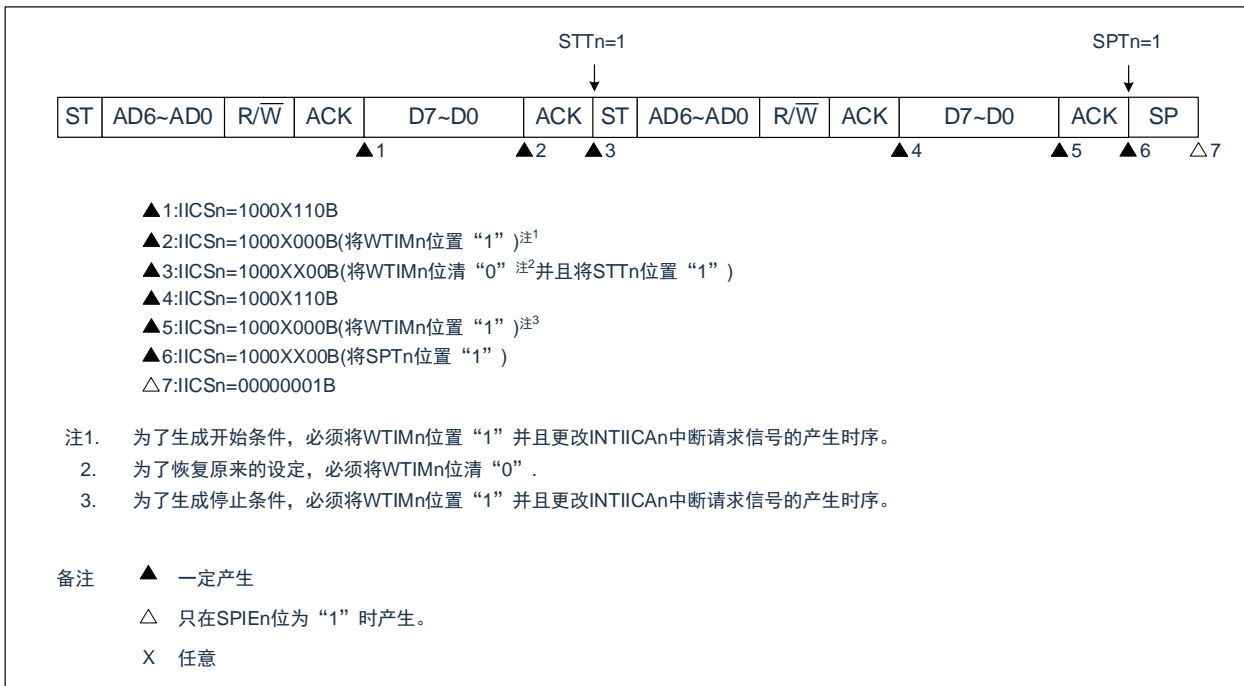
(ii) WTIMn=1的情况



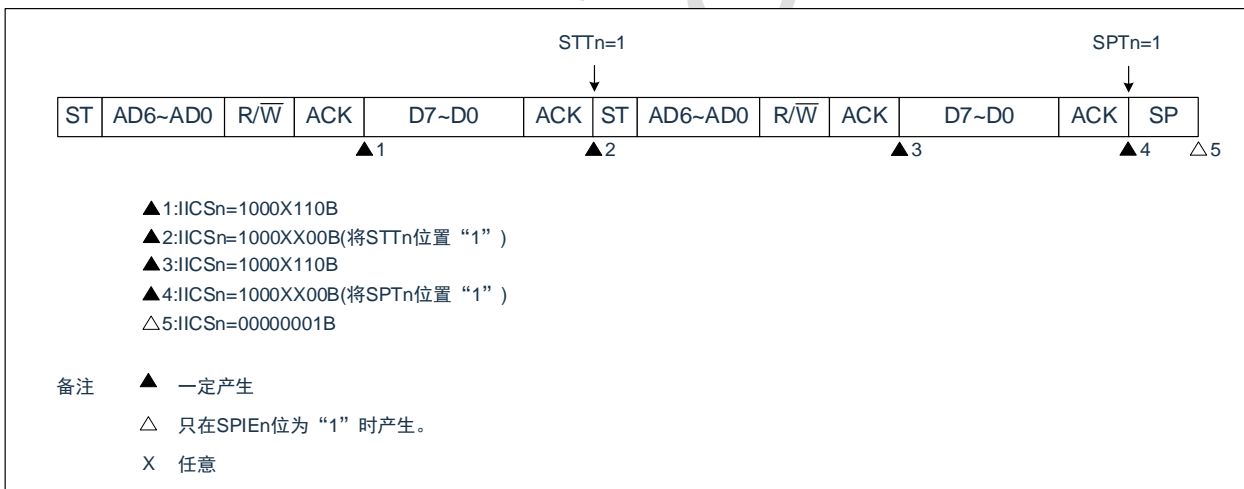
备注: n=0

(b) Start~Address~Data~Start~Address~Data~Stop (重新开始)

(i) WTIMn=0的情况



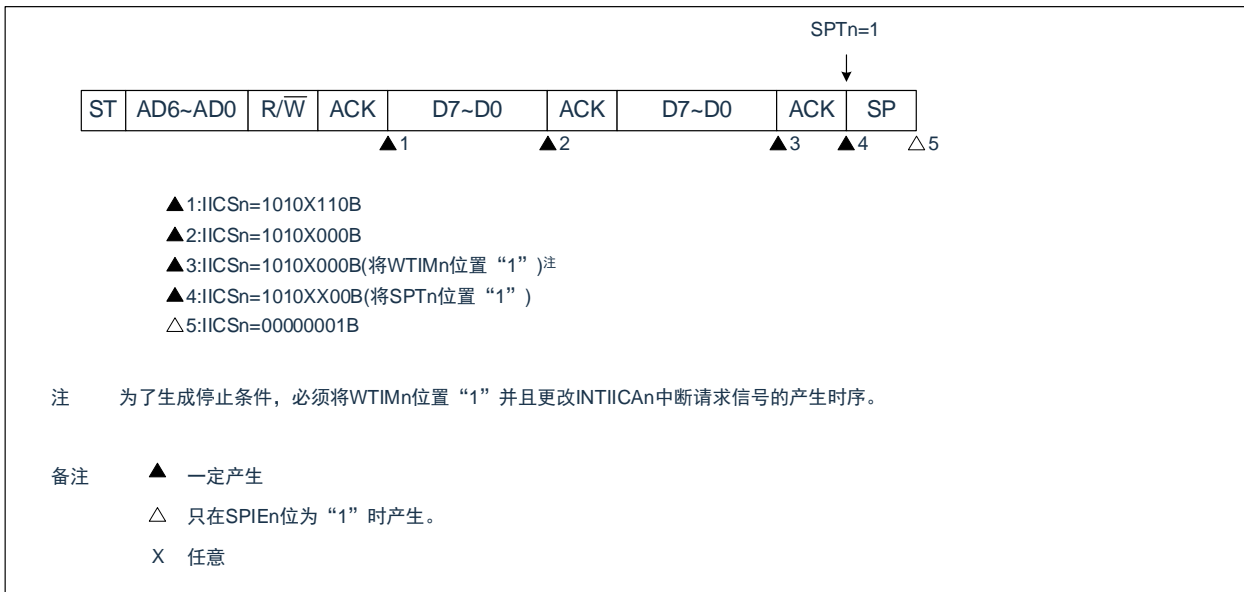
(ii) WTIMn=1的情况



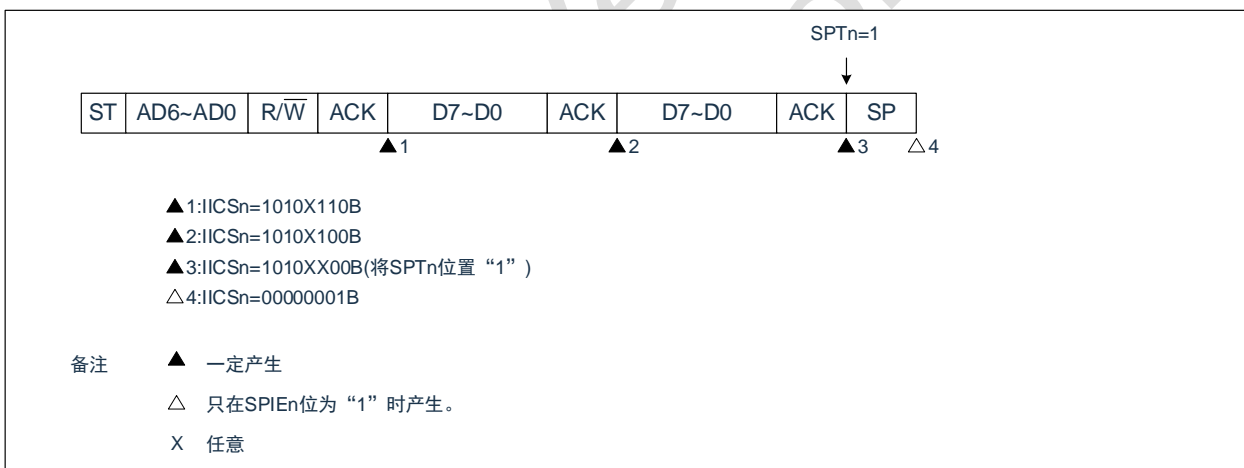
备注: n=0

(c) Start~Code~Data~Data~Stop (发送扩展码)

(i) WTIMn=0的情况



(ii) WTIMn=1的情况

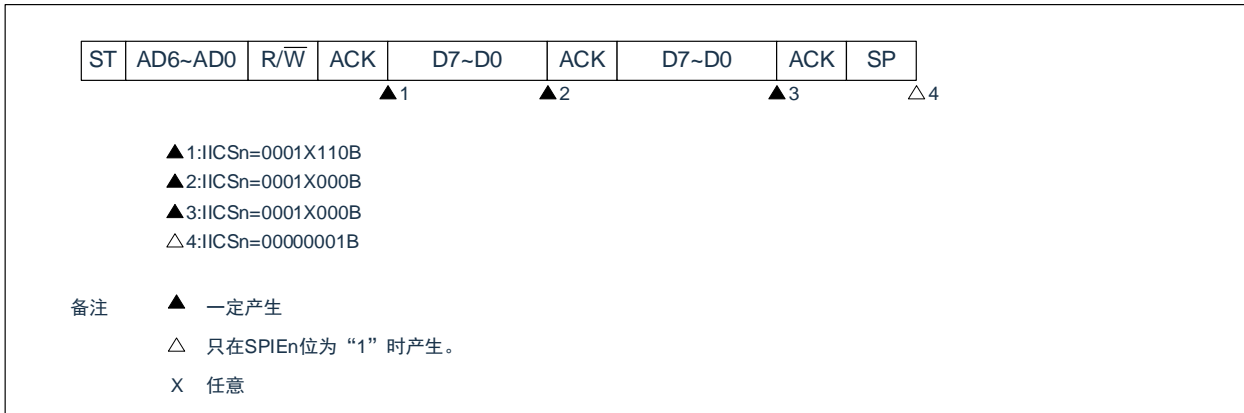


备注: n=0

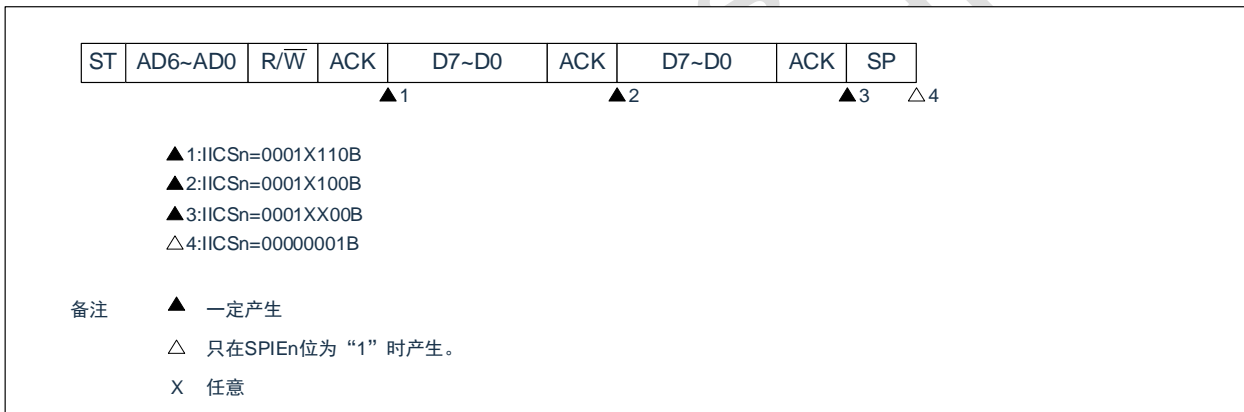
(2) 从属运行（接收从属地址的情况）

(a) Start~Address~Data~Data~Stop

(i) WTIMn=0的情况



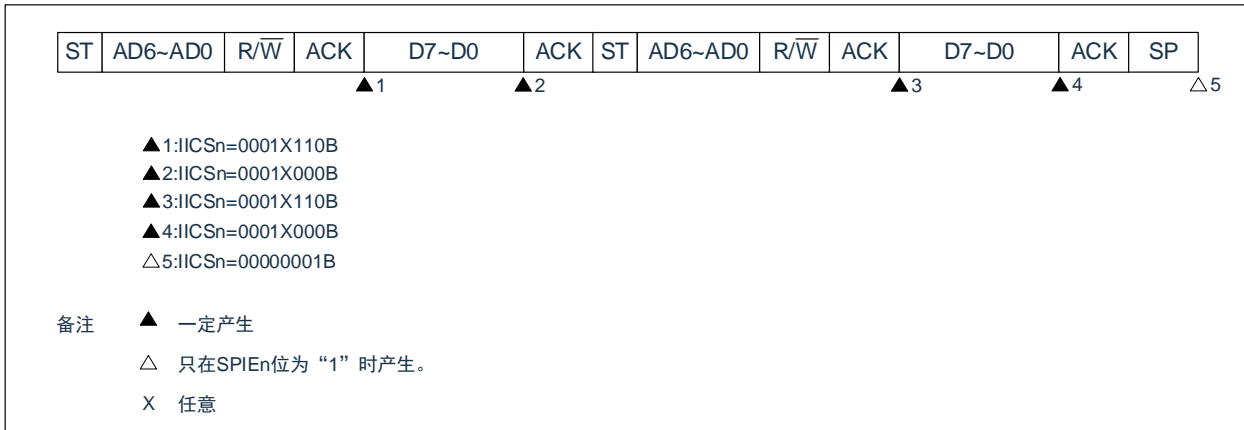
(ii) WTIMn=1的情况



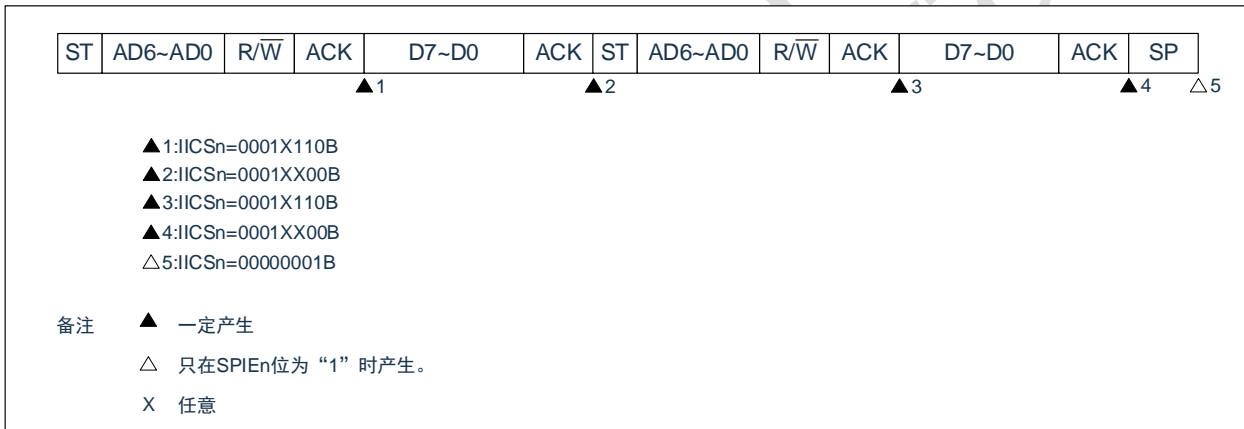
备注: n=0

(b) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）



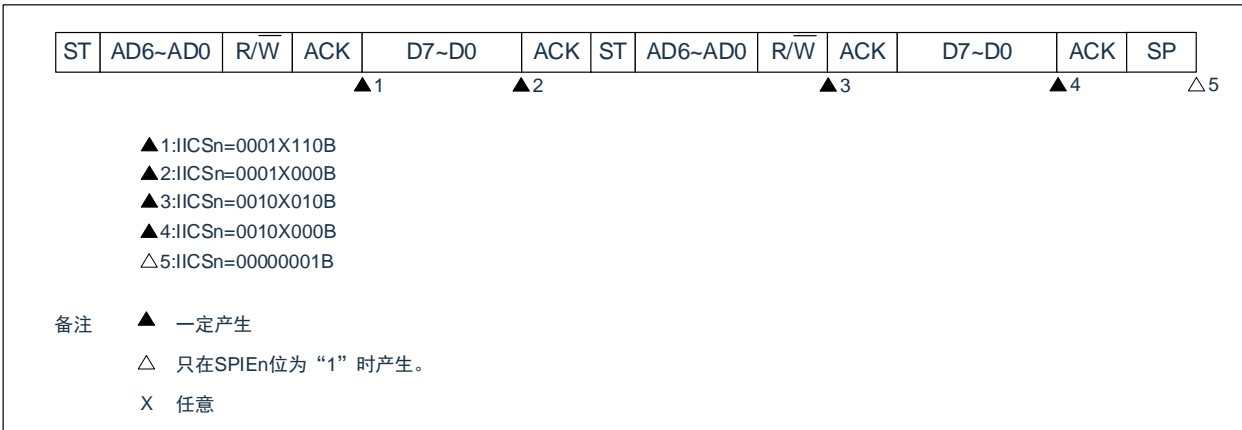
(ii) WTIMn=1的情况（在重新开始后SVAn相同）



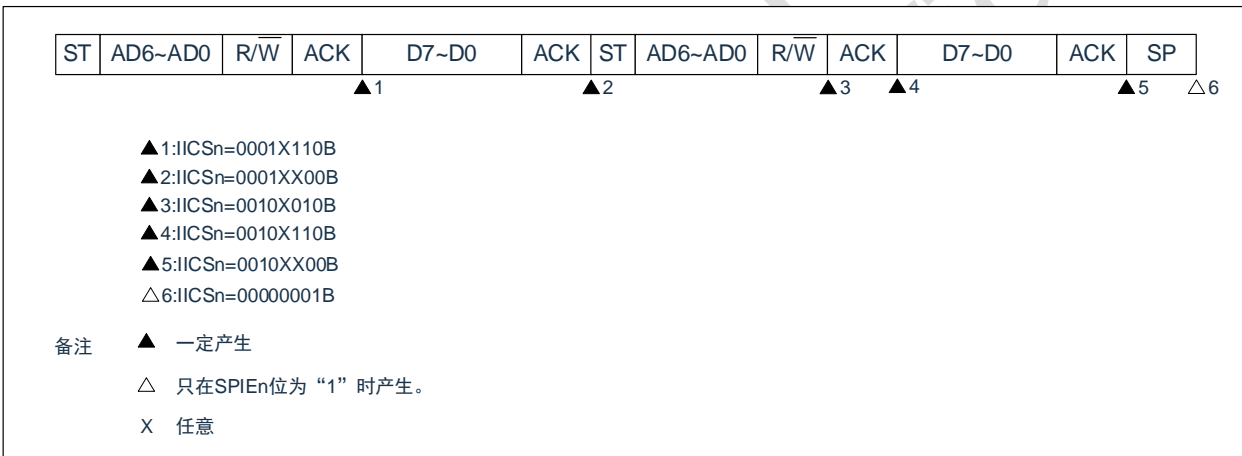
备注: n=0

(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））

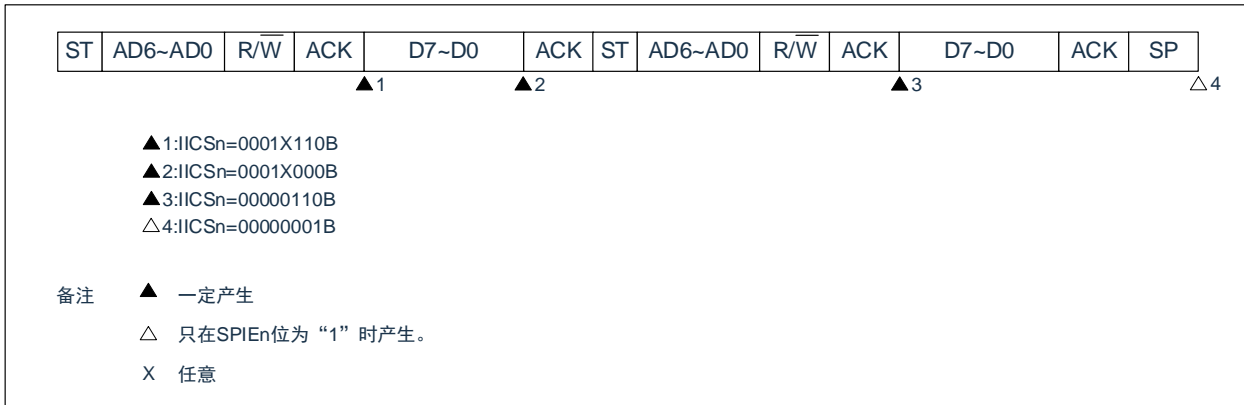


备注：n=0

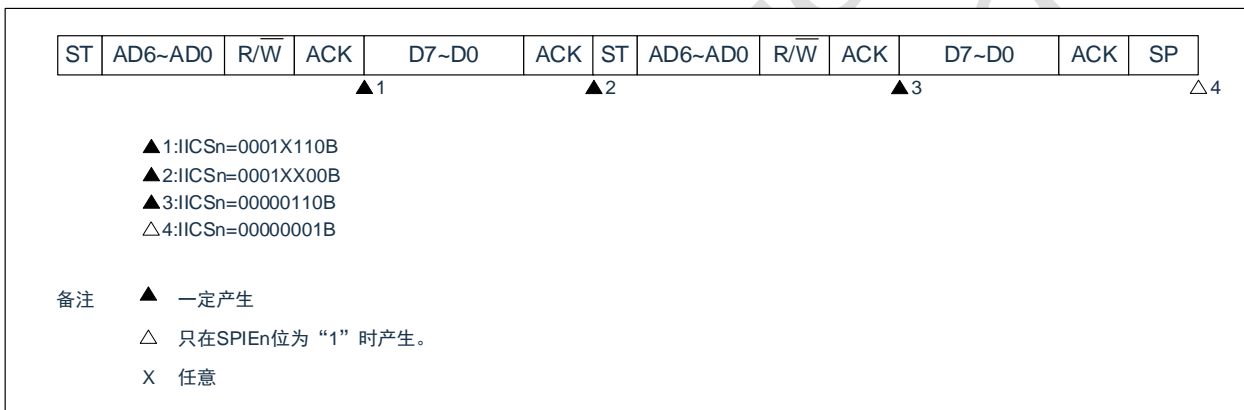


(d) Start~Address~Data~Start~Address~Data~Stop

(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））



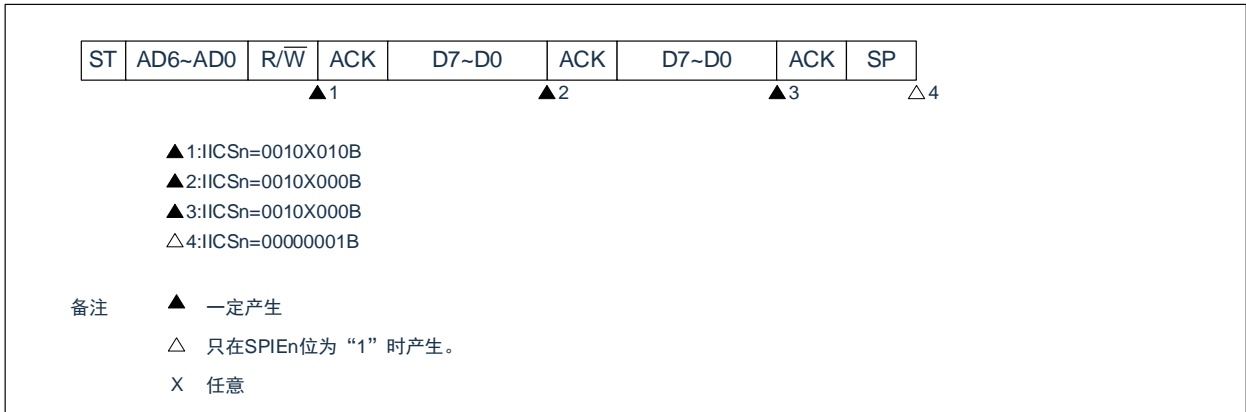
备注: n=0

(3) 从属运行（接收扩展码的情况）

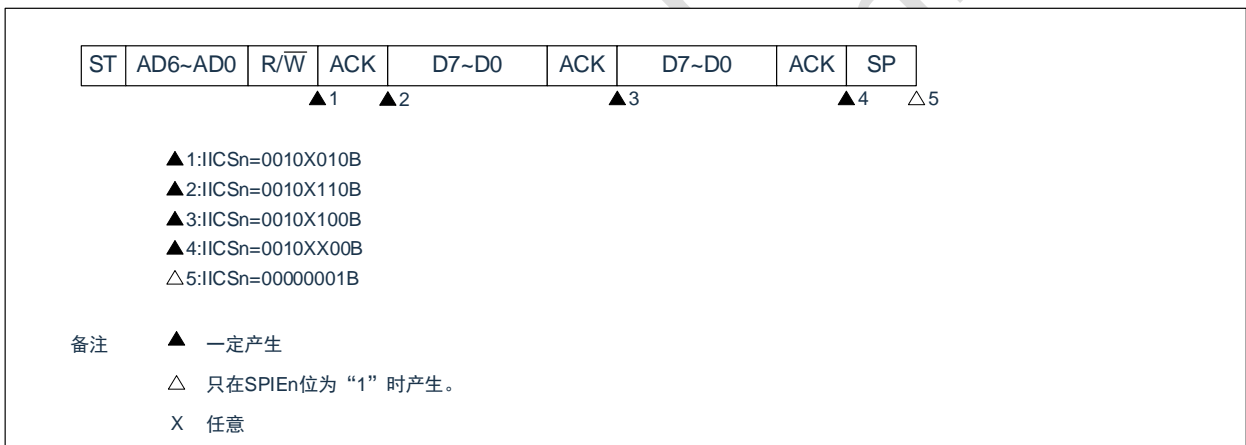
在接收扩展码时，始终参加通信。

(a) Start~Code~Data~Data~Stop

(i) WTIMn=0的情况



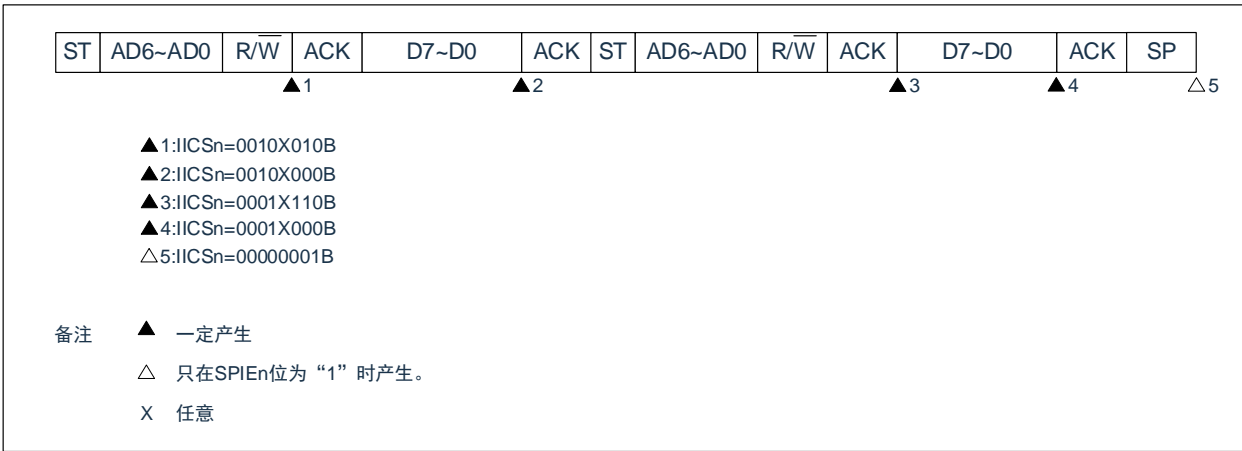
(ii) WTIMn=1的情况



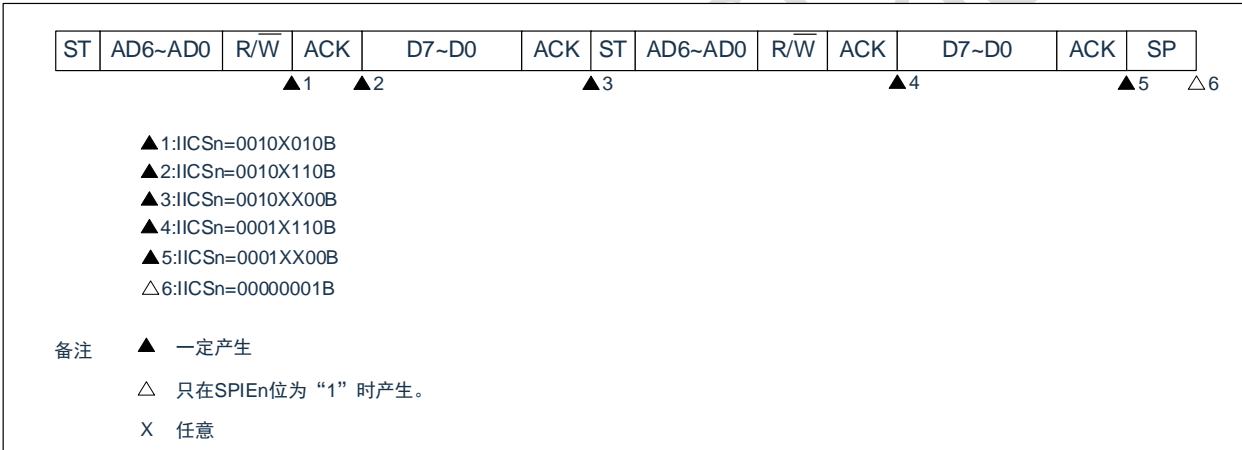
备注：n=0

(b) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）



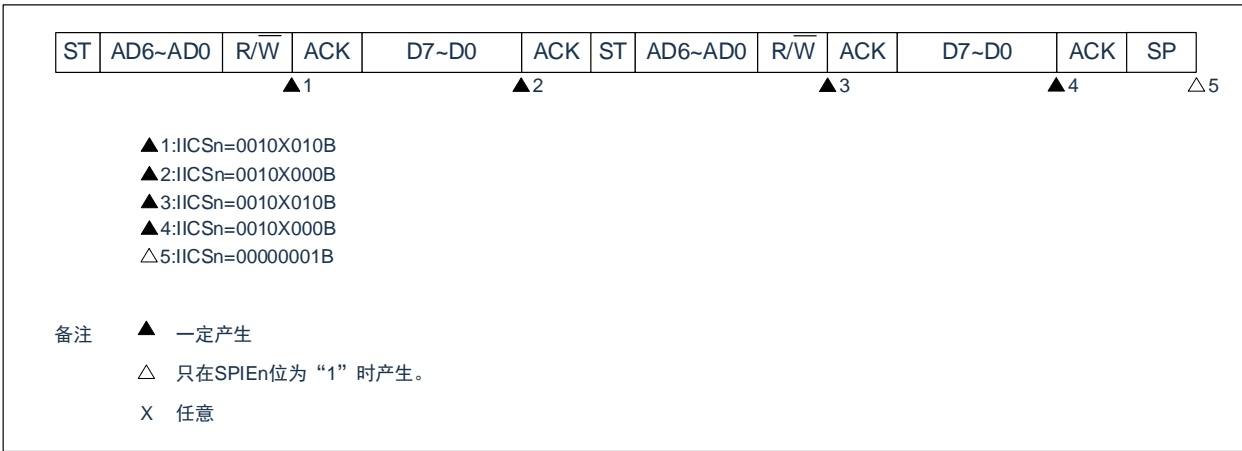
(ii) WTIMn=1的情况（在重新开始后SVAn相同）



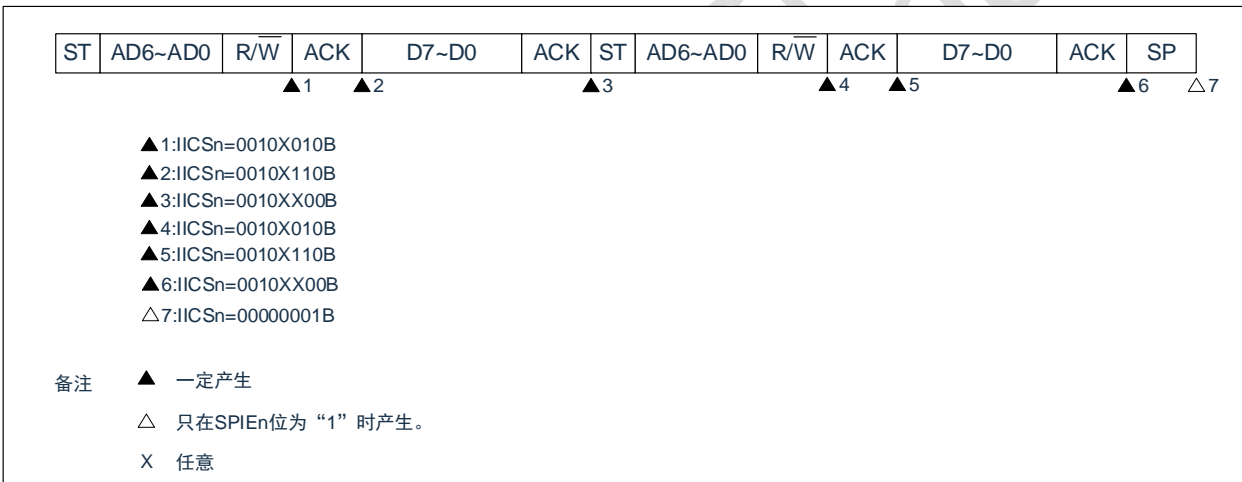
备注: n=0

(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）



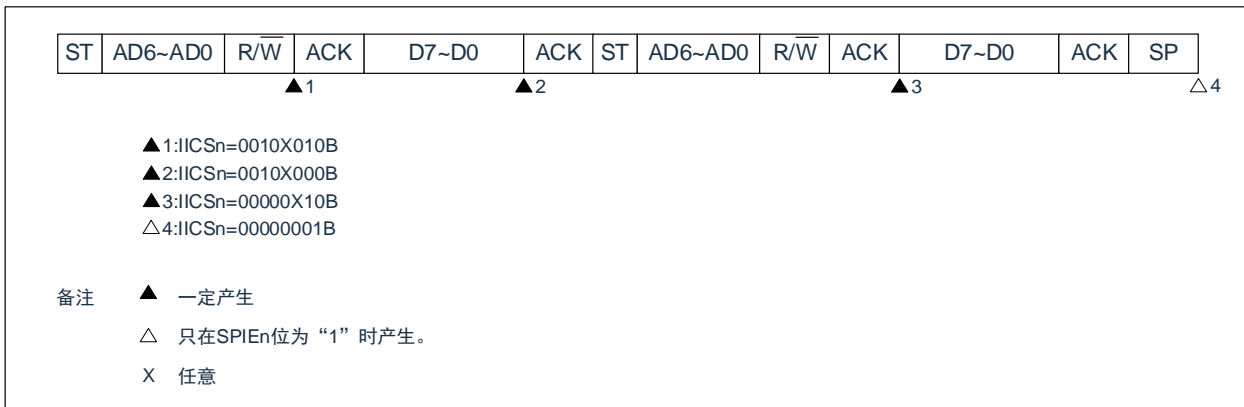
(ii) WTIMn=1的情况（在重新开始后接收扩展码）



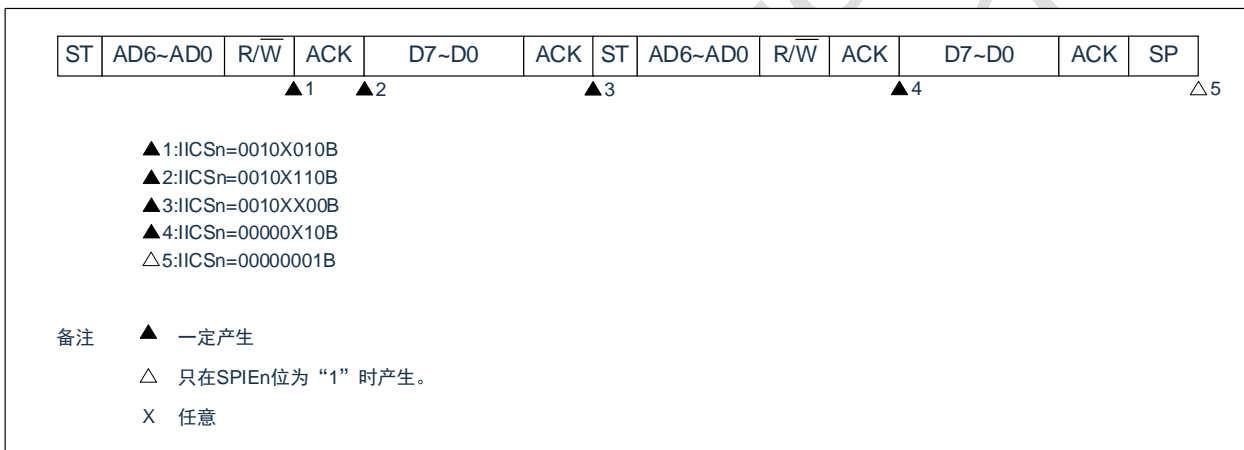
备注：n=0

(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））



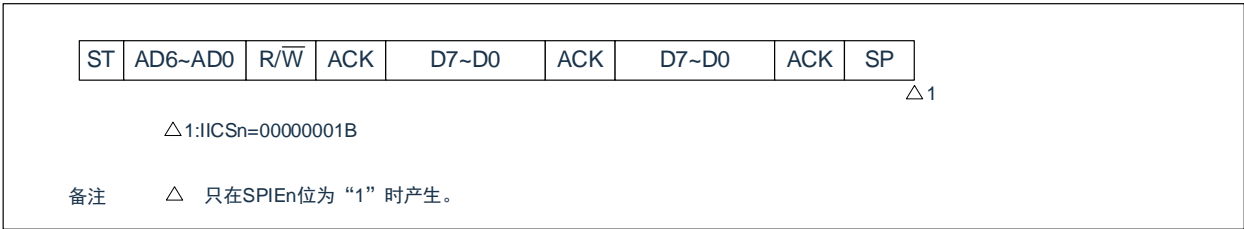
(ii) WTIMn=1的情况（在重新开始后地址不同（非扩展码））



备注: n=0

(4) 不参加通信的运行

(a) Start~Code~Data~Data~Stop

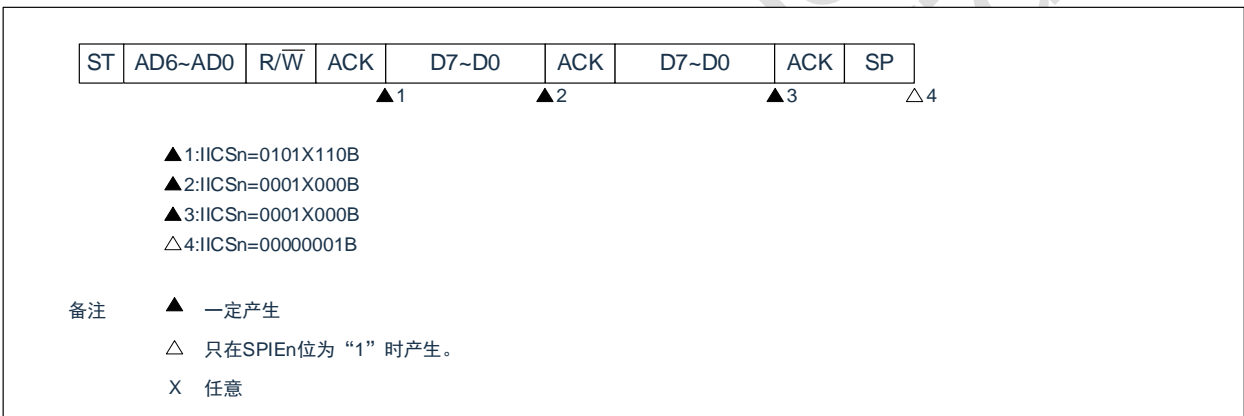


(5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）

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<sub>n</sub>位，确认仲裁结果。

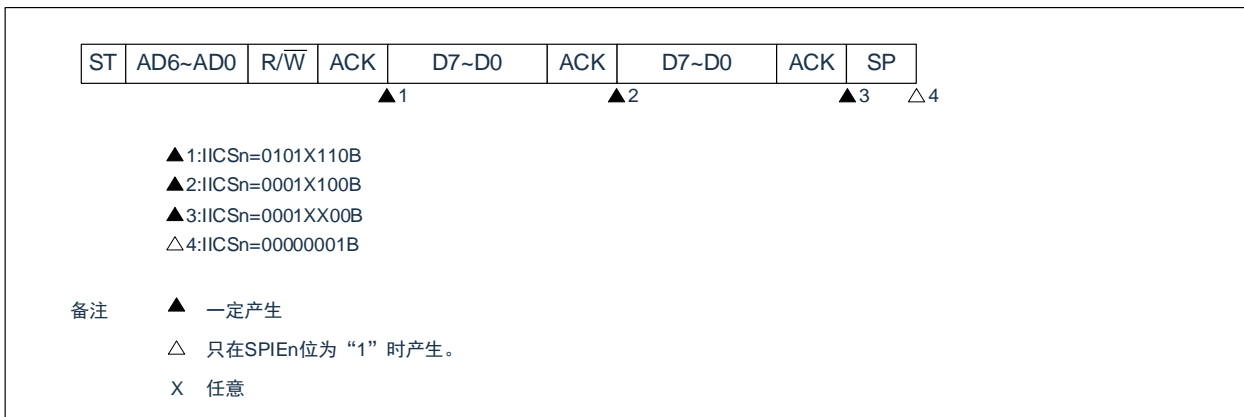
(a) 在发送从属地址数据的过程中仲裁失败的情况

(i) WTIM<sub>n</sub>=0的情况



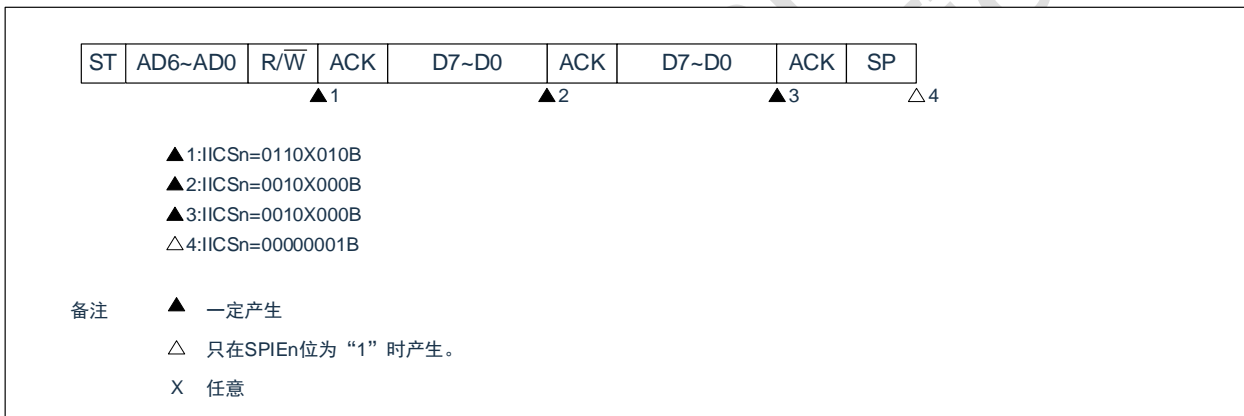
备注: n=0

(ii) WTIMn=1的情况



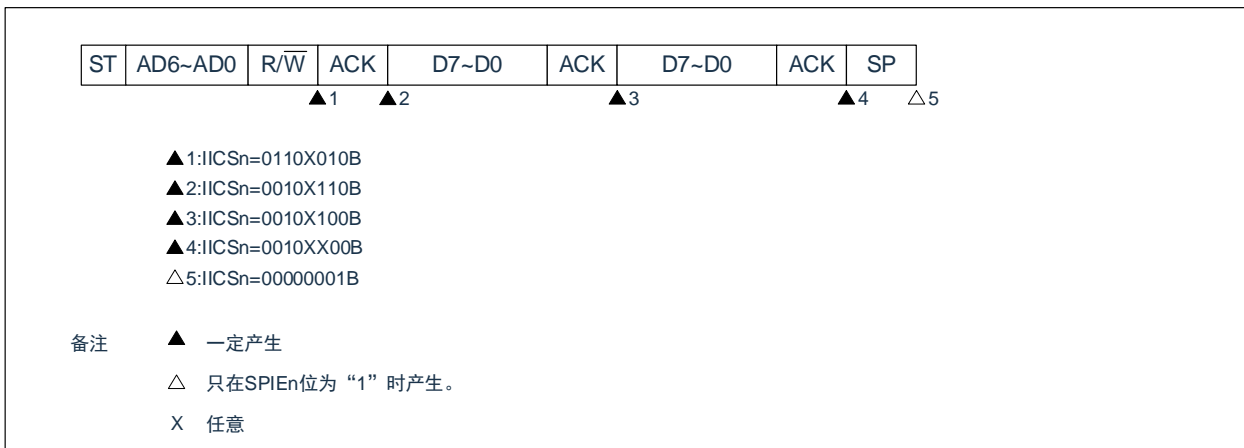
(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIMn=0的情况



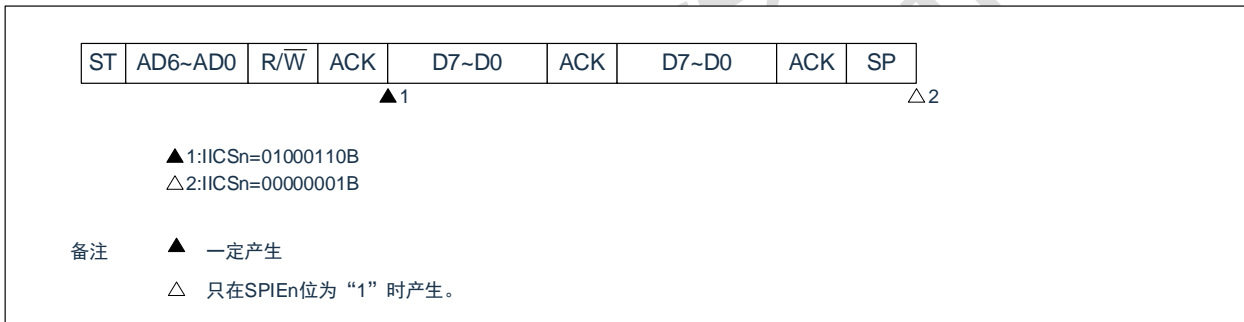
备注: n=0

(ii) WTIMn=1的情况



(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MSTSn位，确认仲裁结果。

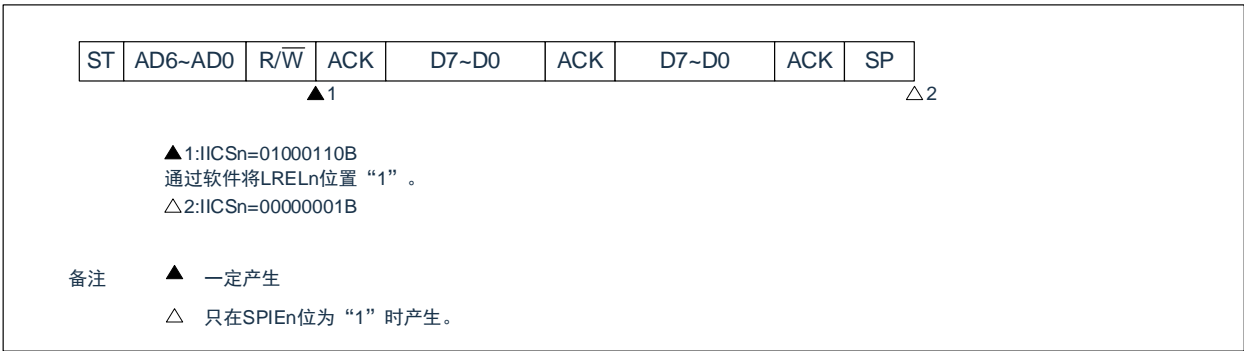
(a) 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）



备注：n=0

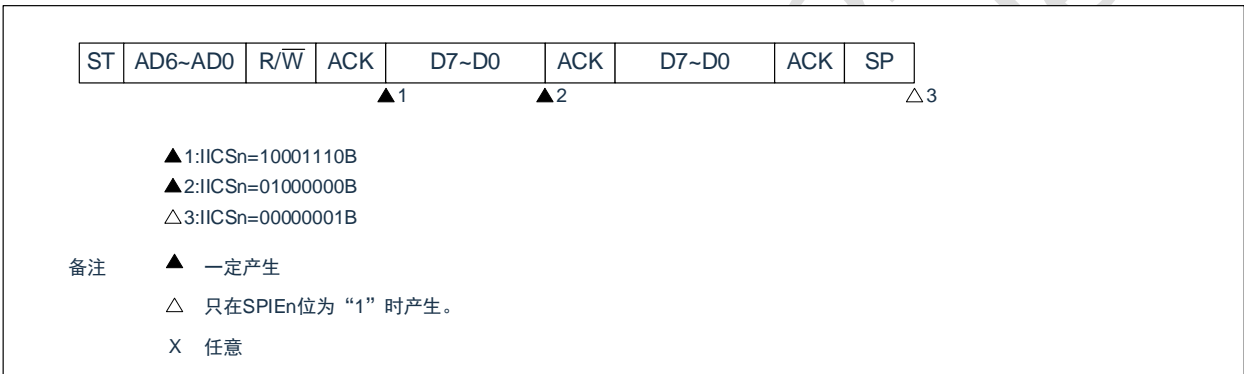


(b) 在发送扩展码的过程中仲裁失败的情况



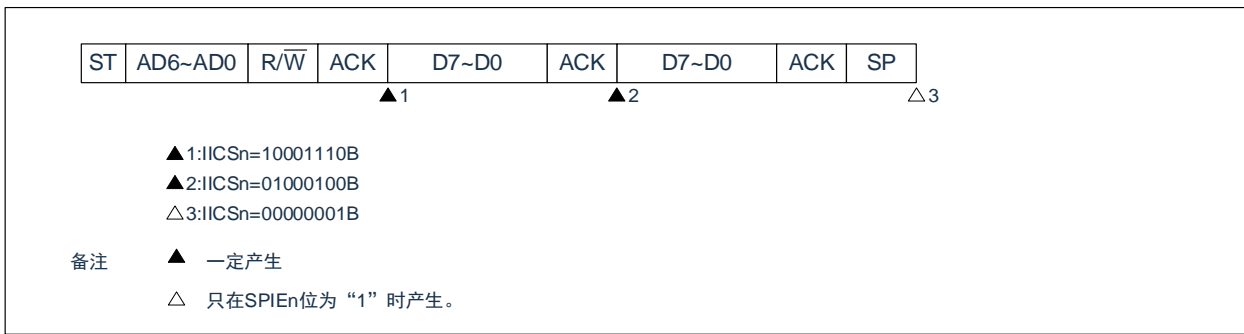
(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0的情况



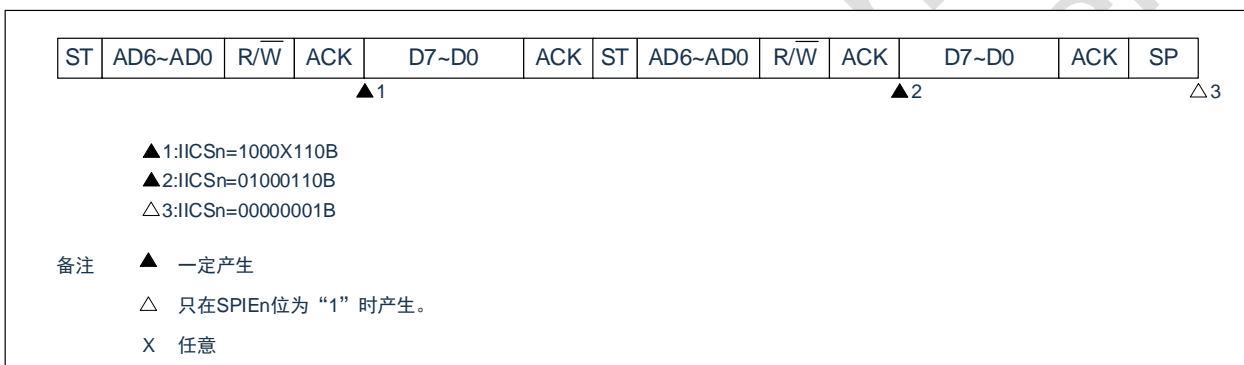
备注: n=0

(ii) WTIMn=1的情况



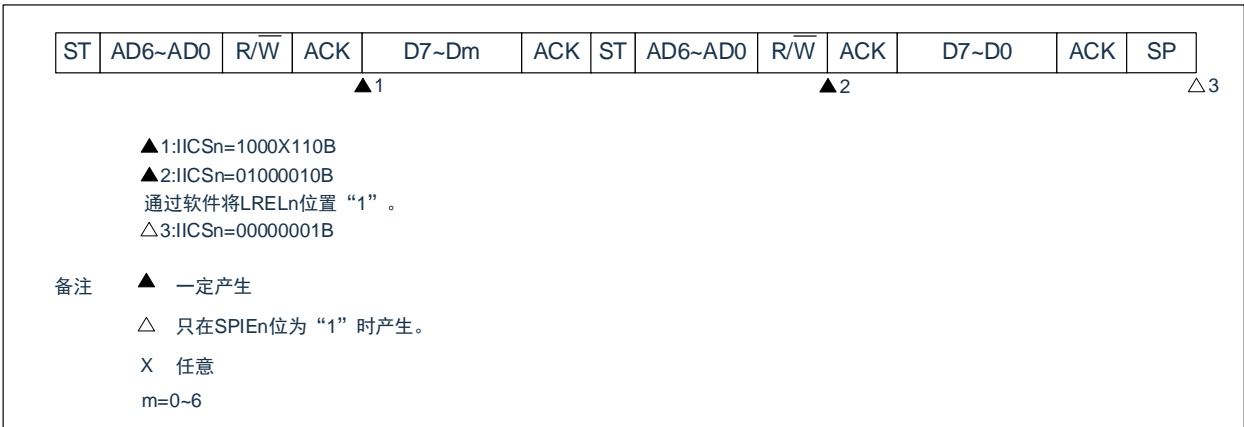
(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码（例如，SVAn不同）

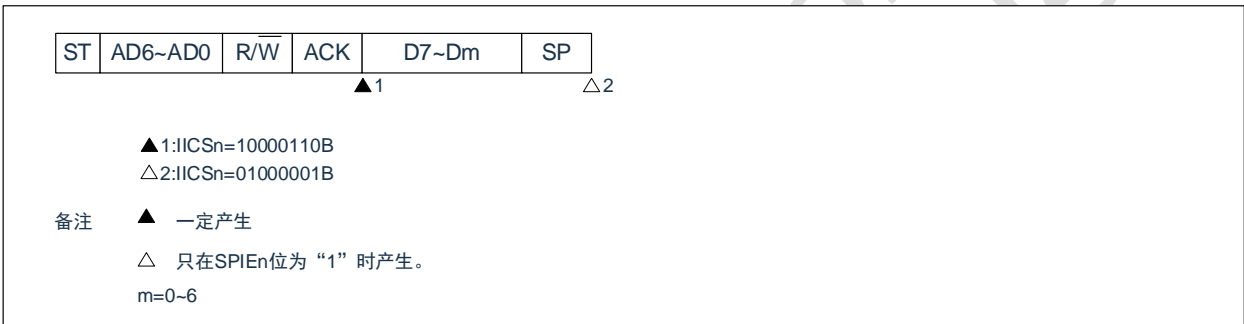


备注: n=0

(ii)扩展码



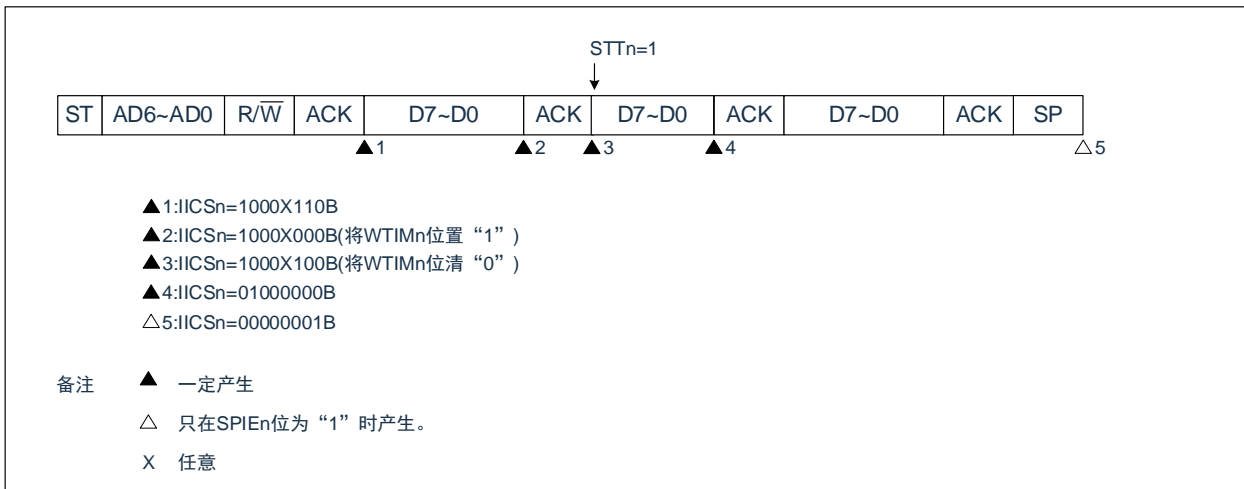
(e) 在传送数据时因停止条件而仲裁失败的情况



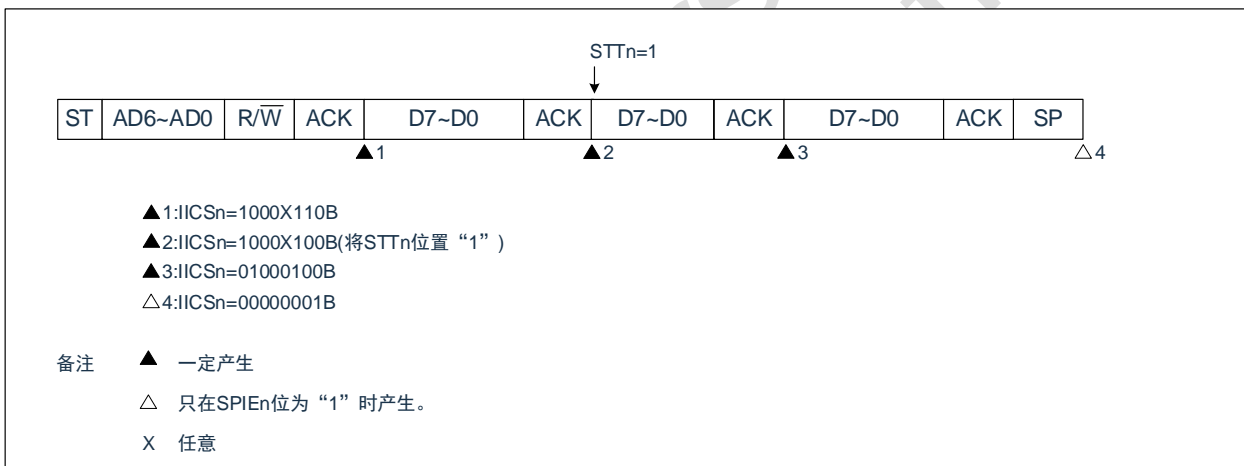
备注: n=0

(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



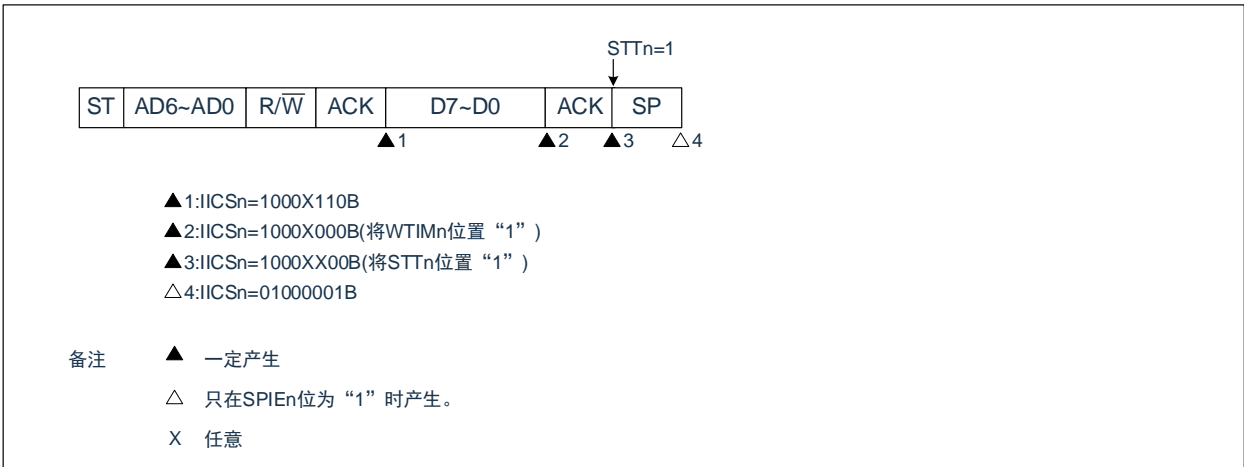
(ii) WTIMn=1的情况



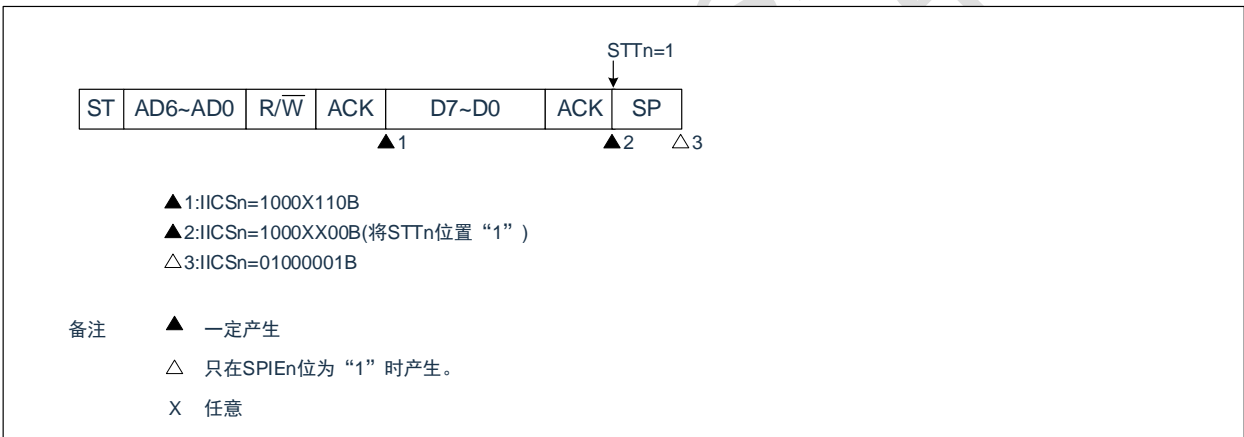
备注: n=0

(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



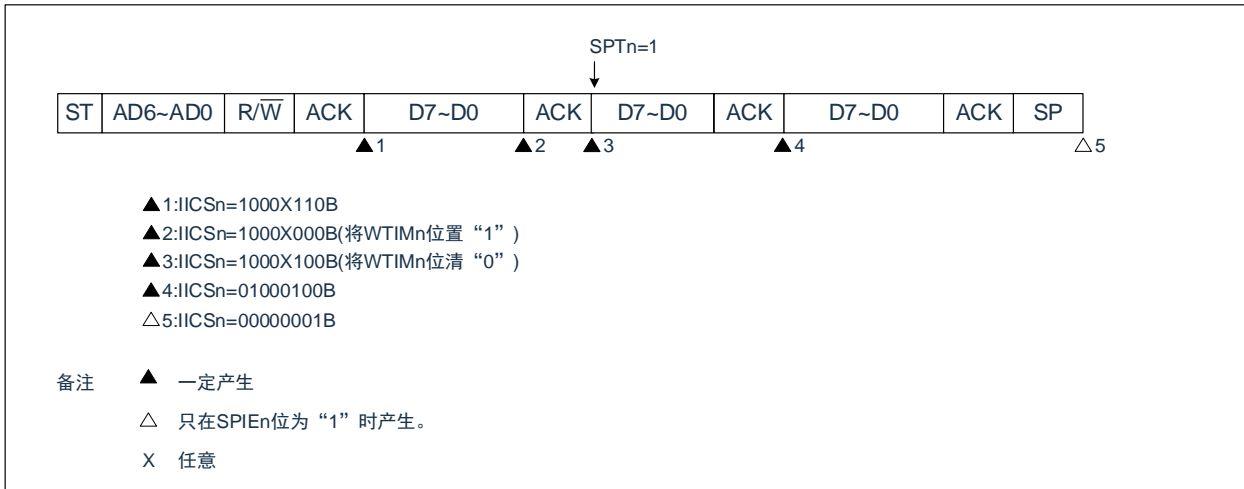
(ii) WTIMn=1的情况



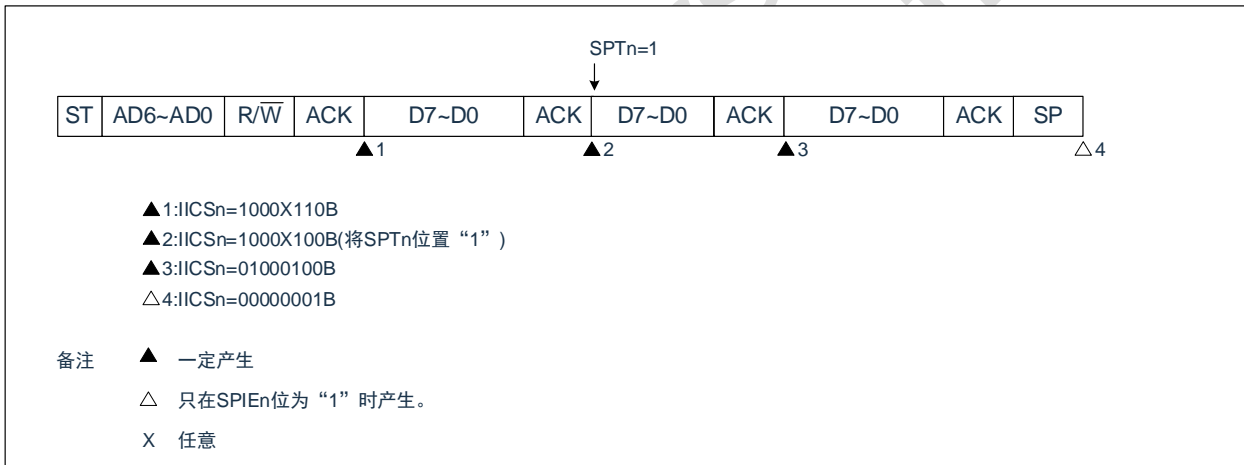
备注: n=0

(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

## 18.6 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18-30和图18-31所示。

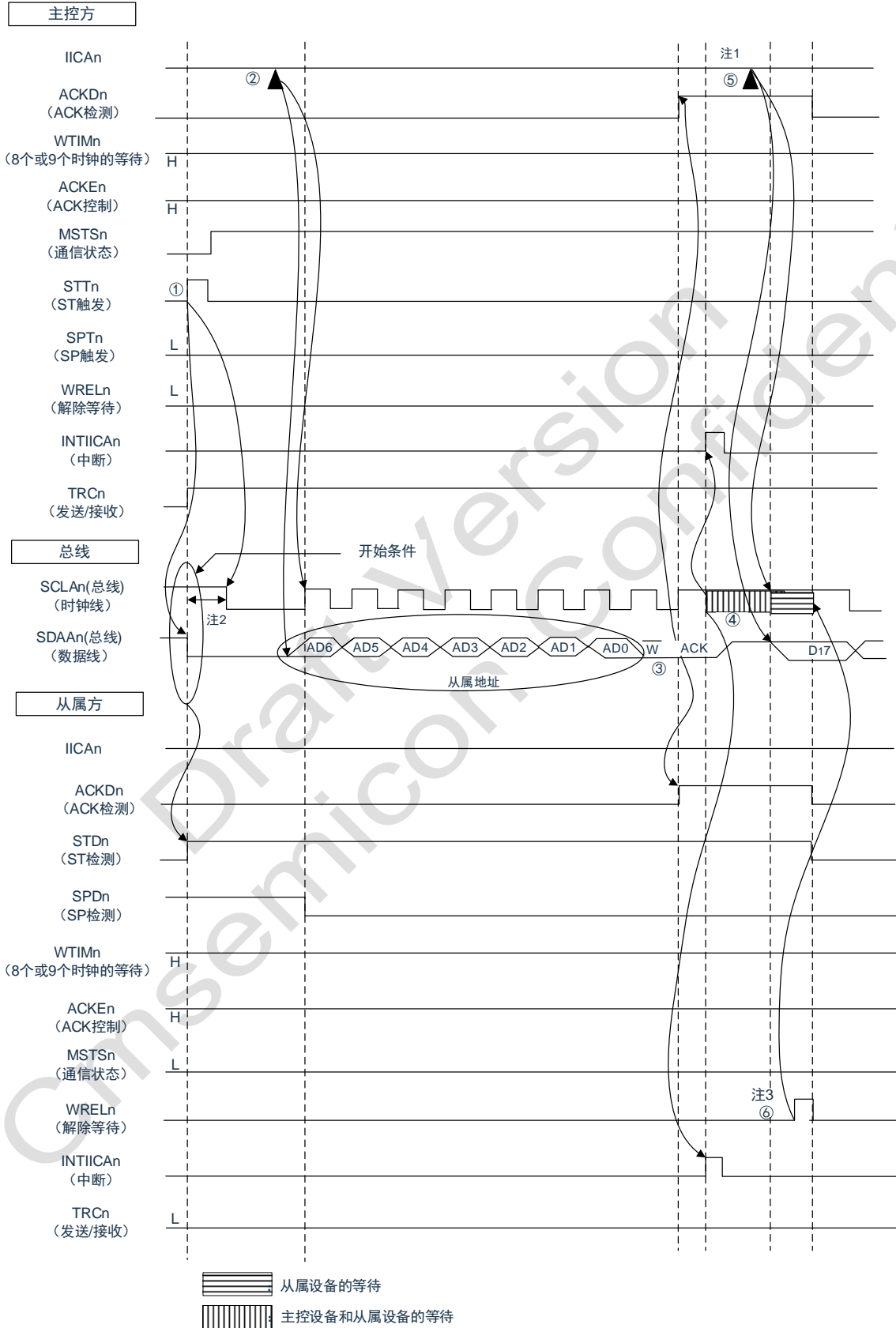
与串行时钟（SCLAn）的下降沿同步进行IICA移位寄存器n（IICAn）的移位，并且将发送数据传送到SO锁存器，以MSB优先从SDAAn引脚输出数据。

在SCLAn的上升沿将SDAAn引脚输入的数据取到IICAn。

备注：n=0

图18-30 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。  
2.从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设定为标准模式时至少为4.0 $\mu$ s，在设定为快速模式时至少为0.6 $\mu$ s。

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18-30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（MSTSn=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18-30的①~⑯是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

图18-30的“(1)开始条件~地址~数据”说明步骤①~⑥。

图18-30的“(2)地址~数据~数据”说明步骤③~⑩。

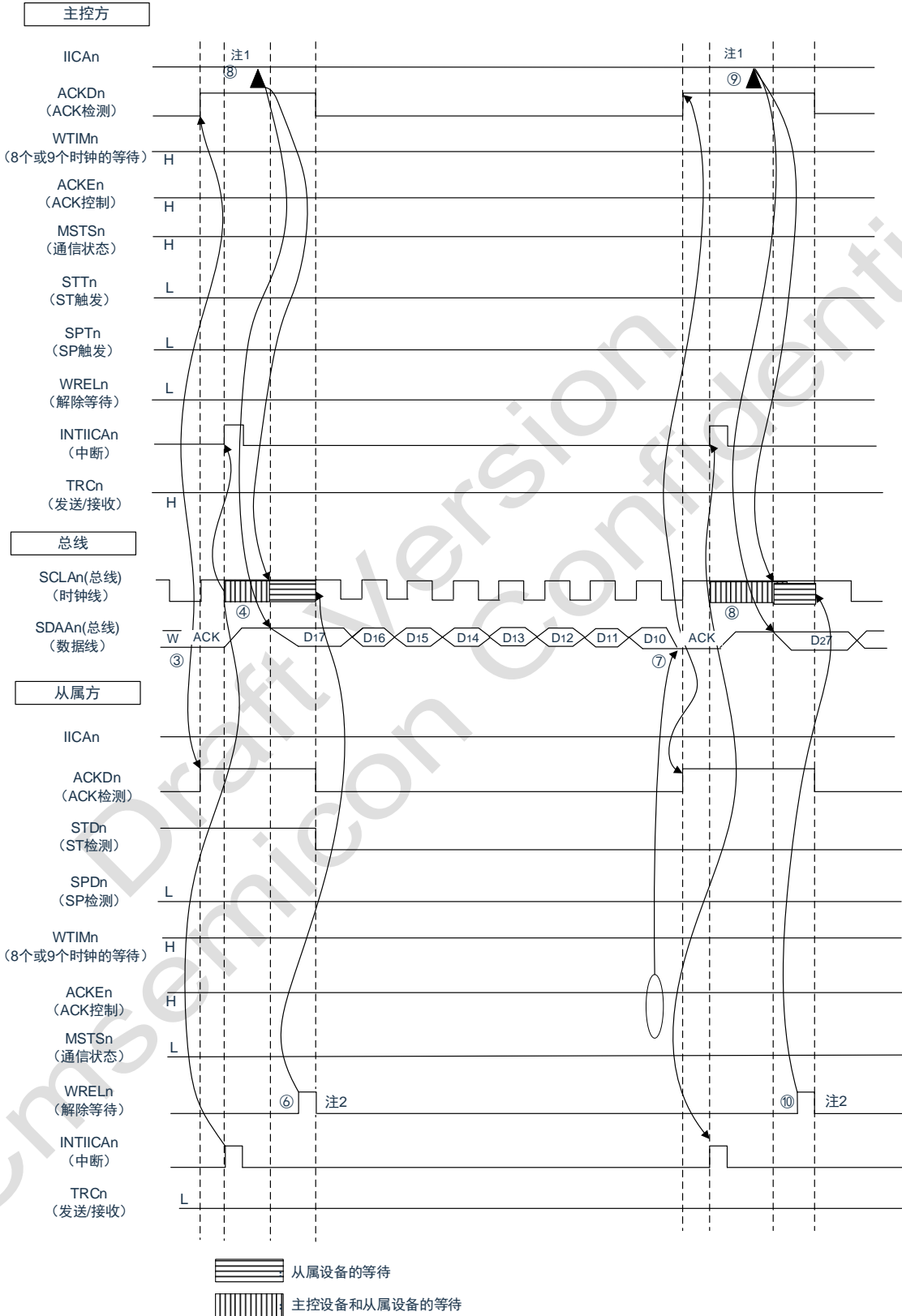
图18-30的“(3)数据~数据~停止条件”说明步骤⑦~⑯。

2.n=0

图18-30 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。

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18-30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18-30的①~⑬是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

图18-30的“(1)开始条件~地址~数据”说明步骤①~⑥。

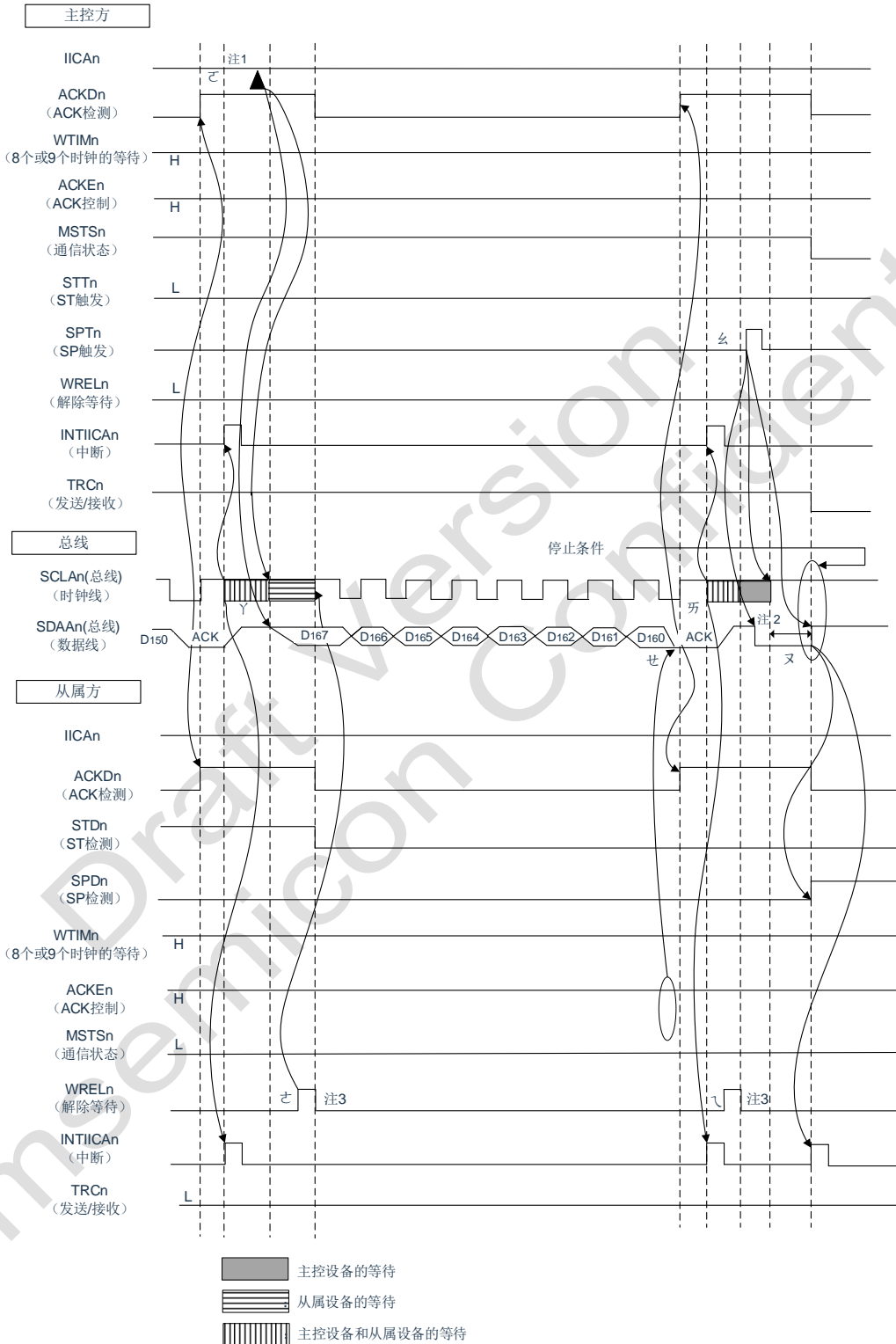
图18-30的“(2)地址~数据~数据”说明步骤③~⑩。

图18-30的“(3)数据~数据~停止条件”说明步骤⑦~⑬。

2.n=0

图18-30 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)

(2) 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。

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，在设定为标准模式时至少为4.0µs，在设定为快速模式时至少为0.6µs。

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18-30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18-30的①~⑮是通过I2C总线进行数据通信的一系列运行步骤。

图18-30的“(1)开始条件~地址~数据”说明步骤①~⑥。

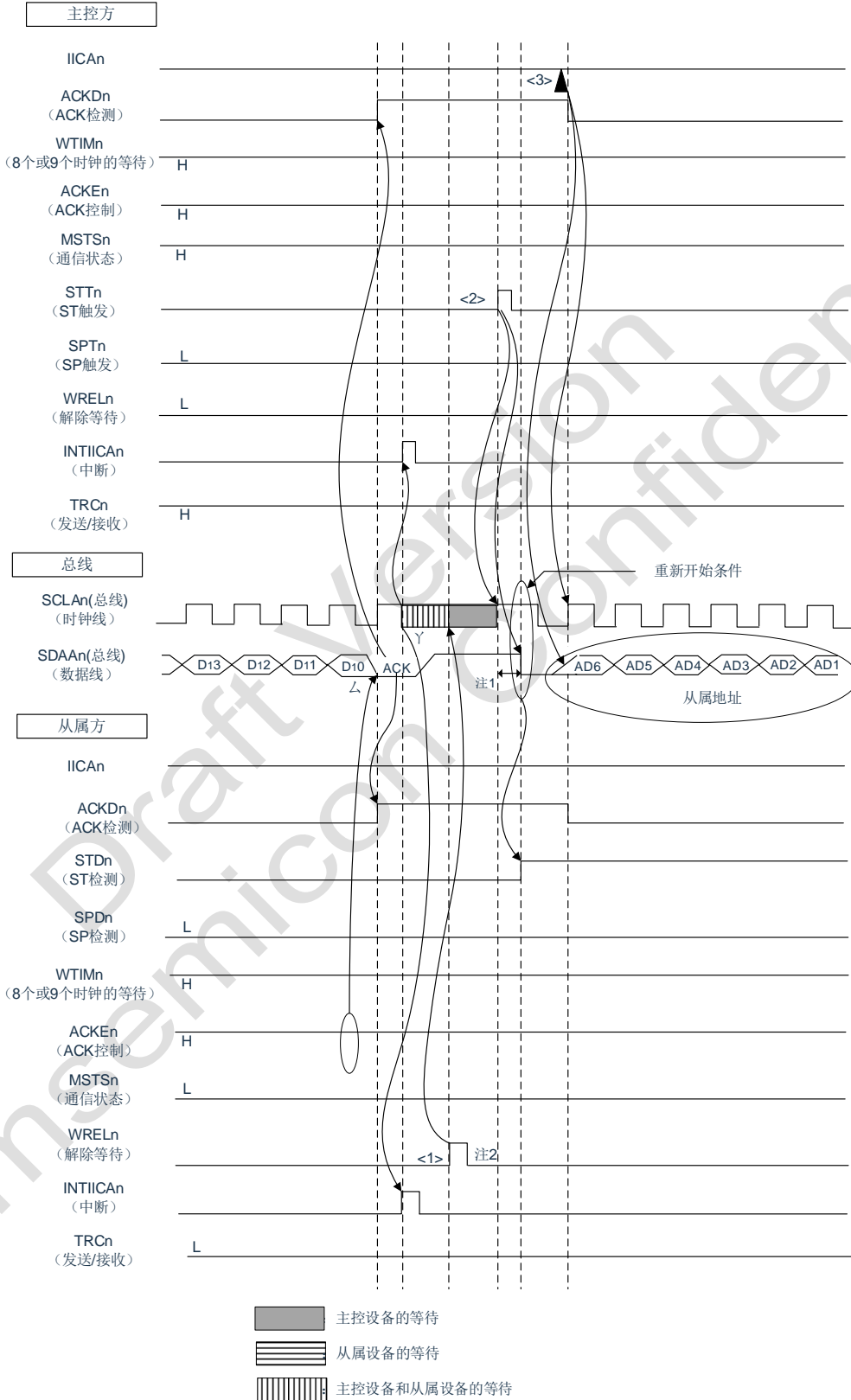
图18-30的“(2)地址~数据~数据”说明步骤③~⑩。

图18-30的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

2.n=0

图18-30 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, 在设定为标准模式时至少为4.7μs, 在设定为快速模式时至少为0.6μs。

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18-30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。

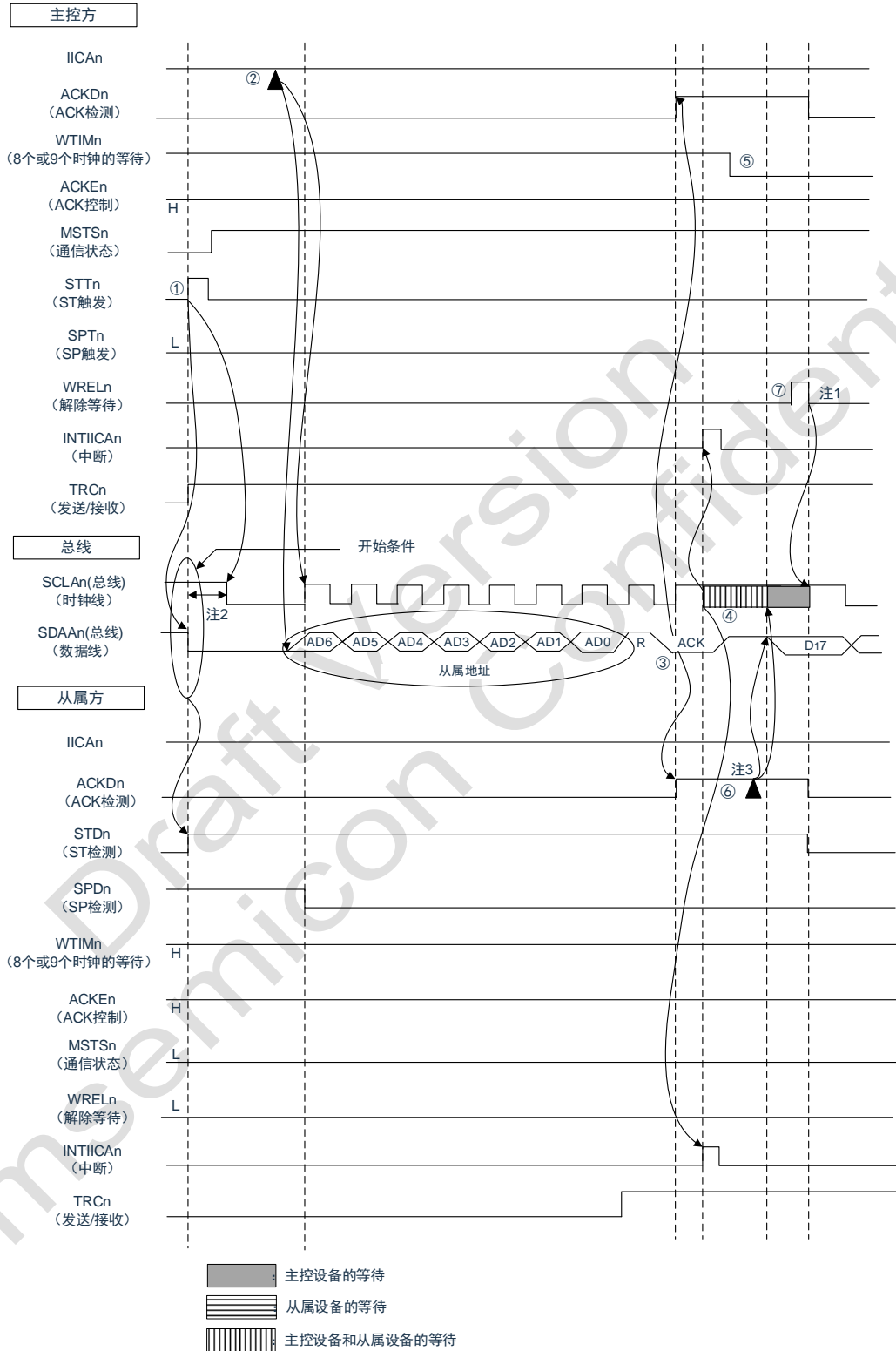
<3>如果主控方给IICA移位寄存器n（IICAn）写地址+RW（发送），就发送从属地址。

备注：n=0

图18-31 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。

2.从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设定为标准模式时至少为4.0us，在设定为快速模式时至少为0.6μs。

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18-31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（MSTSn=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18-31的①~⑱是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

图18-31的“(1)开始条件~地址~数据”说明步骤①~⑦。

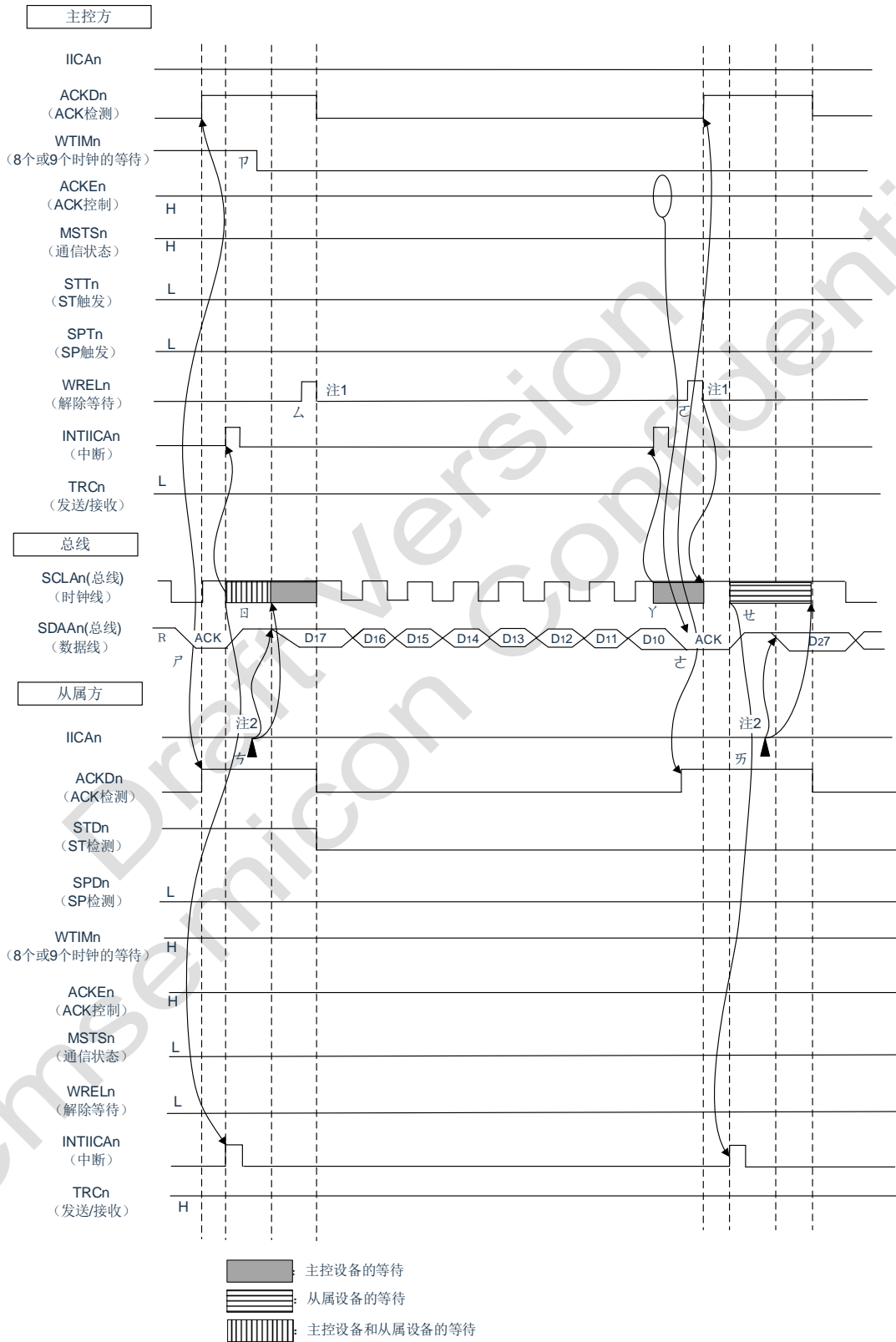
图18-31的“(2)地址~数据~数据”说明步骤③~⑫。

图18-31的“(3)数据~数据~停止条件”说明步骤⑧~⑱。

2.n=0

图18-31 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。

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18-31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：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18-31的①~⑭是通过I<sup>2</sup>C总线进行数据通信的一系列运行步骤。

图18-31的“(1)开始条件~地址~数据”说明步骤①~⑦。

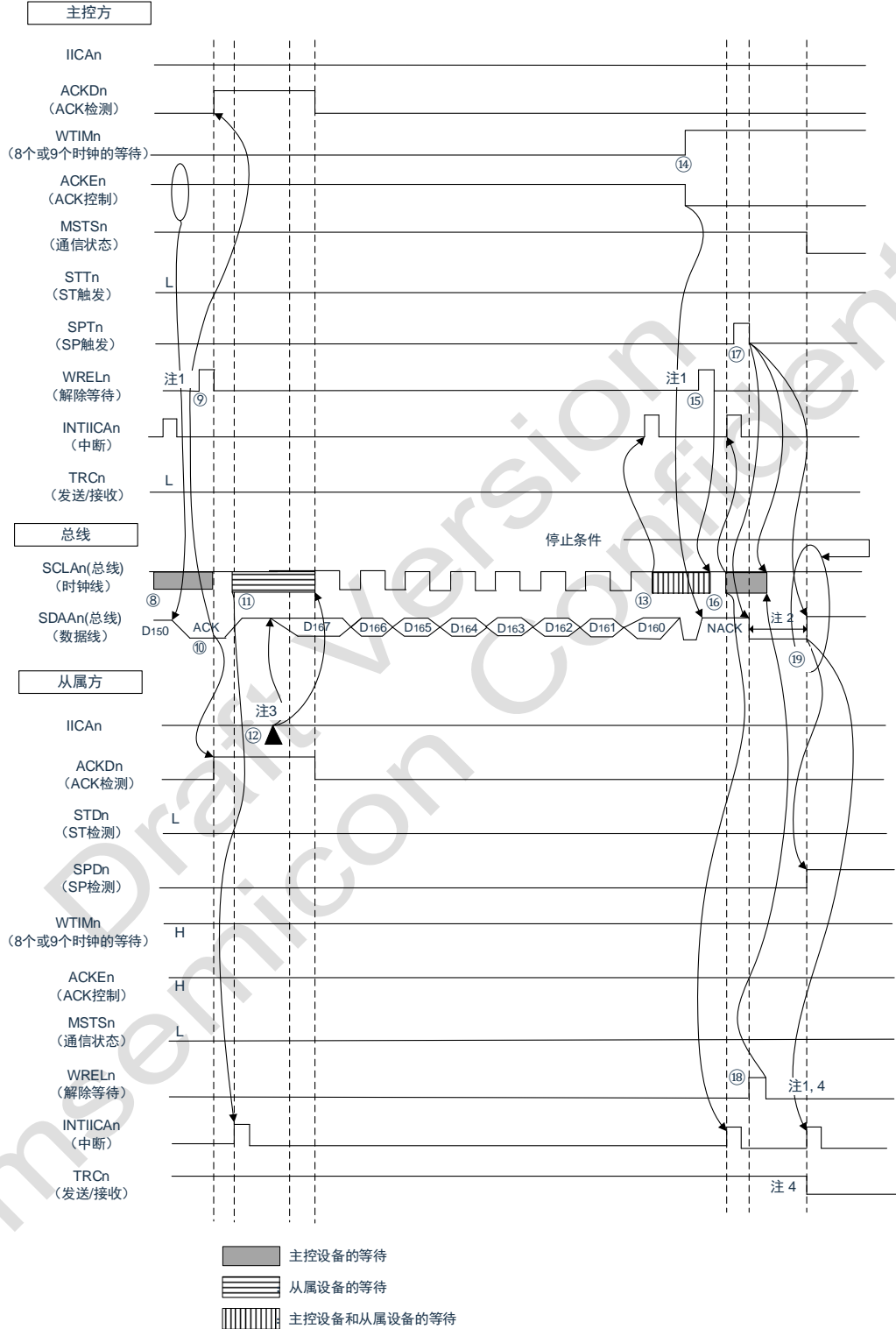
图18-31的“(2)地址~数据~数据”说明步骤③~⑫。

图18-31的“(3)数据~数据~停止条件”说明步骤⑧~⑭。

2.n=0

图18-31 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。  
 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，在设定为标准模式时至少为4.0us，在设定为快速模式时至少为0.6us。  
 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。  
 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18-31的“(3)数据~数据~停止条件”的⑧~⑲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. 主控方设定为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19章 串行接口SPI

### 19.1 串行接口SPI的功能

本产品搭载一个串行接口SPI0，有以下2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。

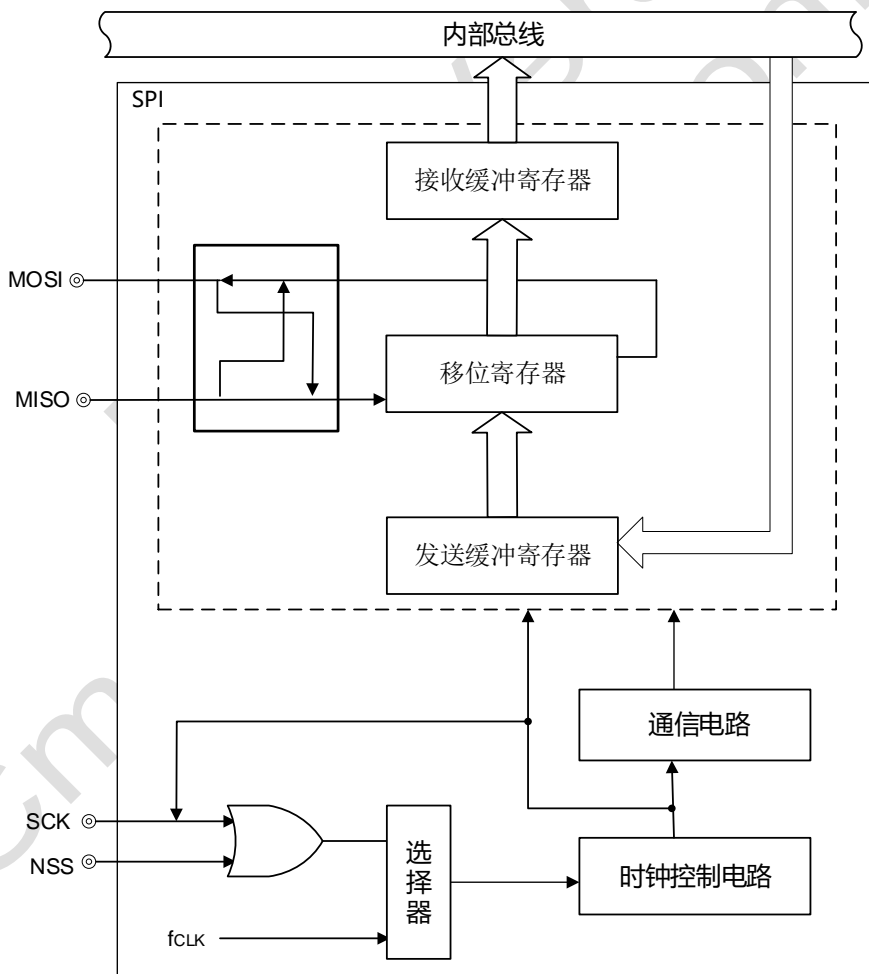
(2) 3-wire串行I/O模式

此模式通过串行时钟（SCKn）和串行数据总线（MISO<sub>n</sub>和MOSI<sub>n</sub>）的3条线，与多个设备进行8位或16位数  
据传送。

备注：n=0

### 19.2 串行接口SPI的结构

图19-1 串行接口SPI的框图



## 19.3控制串行接口SPI的寄存器

通过以下寄存器控制串行接口SPI。

外围允许寄存器2 (PER2)

串行操作模式寄存器 (SPIMn)

串行时钟选择寄存器 (SPICn)

发送缓冲寄存器 (SDROn)

接收缓冲寄存器 (SDRI n)

端口模式寄存器 (PMxx)

端口模式控制寄存器 (PMCxx)

端口寄存器 (Pxx)

备注: n=0

寄存器列表:

基地址	偏移地址	寄存器名称	R/W	复位值
0x4004_5D00	0x000	SPIMn	R/W	00H
	0x004	SPICn	R/W	01H
	0x008	SDRO	R/W	0000H
	0x00C	SDRI	R	0000H
	0x010	SPI Sn	R	00H

### 19.3.1 外围允许寄存器2 (PER2)

PER2寄存器是设定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SPI功能时，必须将SPInEN置“1”。

详细请参见“4.3.6 外围允许寄存器0、1、2 (PER0、PER1、PER2)”

备注：n=0

Draft Version  
Cmsemicon Confidential



## 19.3.2 SPI操作模式寄存器 (SPIMn)

SPIM用于选择操作模式并控制操作的允许或禁止。

可由8位存储操作指令设置SPIMn。

产生复位信号将该寄存器清除为00H。

图19-2 模式控制寄存器 (SPIMn) 的格式

复位后: 00H								
符号	7	6	5	4	3	2	1	0
SPIMn	SPIEn	TRMDn	NSSEn	DIRn	INTMDn	DLSn	RECMDn	-

SPIEn	SPI运行的允许
0	停止运行。
1	允许运行。

TRMDn注3	发送/接收模式控制
0	接收模式
1	发送/接收模式

NSSEn	NSS引脚使用选择
0	未使用NSS引脚
1	使用NSS引脚

DIRn	数据传送顺序选择
0	进行MSB优先的输入/输出。
1	进行LSB优先的输入/输出。

INTMDn	中断源选择
0	传送结束中断
1	发送缓冲器空中断

DLSn	数据长度的设定
0	8位数据长度
1	16位数据长度

RECMDn	接收模式的模式选择
0	单次接收
1	连续接收

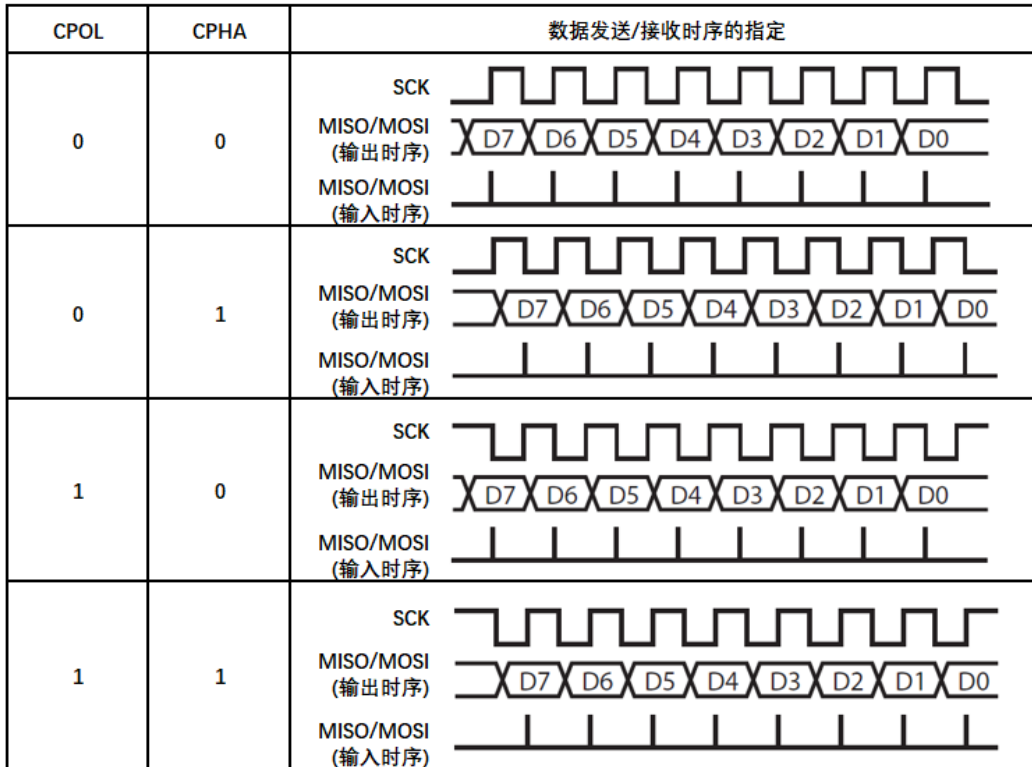
- 注
- 1.SPTF=1 (串行通信期间) 时, 禁止重写 TRMD,DIR,NSSE。
  - 2.TRMD 为 0 时 MO 或 SO 输出固定为低电平。
  - 3.将该位置为 1 之前, 将 NSS 引脚输入电平固定为 0 或者 1。

### 19.3.3 SPI时钟选择寄存器 (SPICn)

该寄存器指定数据发送/接收的时序，并设置串行时钟。  
 SPIC可由8位存储操作指令设置。  
 产生复位信号将该寄存器清除为01H。

图19-3 时钟选择寄存器 (SPICn) 的格式

复位后: 01H				R/W				
符号	7	6	5	4	3	2	1	0
SPICn	0	0	0	CPOLn	CPHAn	CKS2n	CKS1n	CKS0n



CKS2n	CKS1n	CKS0n	SPI 串行时钟选择	模式
0	0	0	fCLK	主机模式
0	0	1	fCLK/2	
0	1	0	fCLK/2 <sup>2</sup>	
0	1	1	fCLK/2 <sup>3</sup>	
1	0	0	fCLK/2 <sup>4</sup>	
1	0	1	fCLK/2 <sup>5</sup>	
1	1	0	fCLK/2 <sup>6</sup>	
1	1	1	从 SCK 输入的外部时钟	从机模式

注意事项1.SPIEn=1（操作使能）时禁止写入SPICn。  
 2.复位后数据时钟的相位类型为类型1。

备注: n=0

### 19.3.4 SPI状态寄存器 (SPISn)

SPIS寄存器用于确认SPI的通信状态。  
 可由8位存储操作指令读取SPISn。  
 产生复位信号将该寄存器清除为00H。

图19-4 状态寄存器 (SPISn) 的格式

复位后: 00H

R

符号	7	6	5	4	3	2	1	0
SPISn	-	-	-	-	-	-	SDRIFn	SPTFn

SDRIFn	接收缓冲器非空标志位
0	接收缓存里没有新接收到的有效数据
1	接收缓存里有接收到的有效数据。在读取寄存器SDRIF时, 该位清0

SPTFn注1	通信状态标志位
0	通信停止
1	通信正在进行中

注 1.SPTF=1 (串行通信期间) 时, 禁止重写 TRMD,DIR,NSSE。  
 2.n=0

### 19.3.5 发送缓冲寄存器 (SDR0n)

该寄存器设置发送数据。

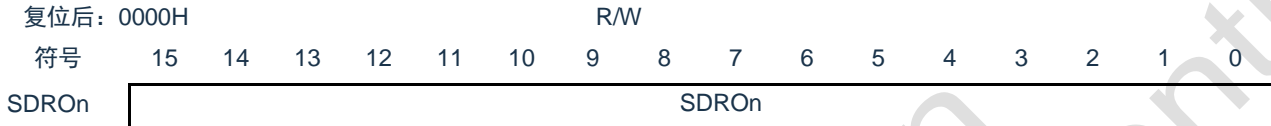
当将串行操作模式寄存器 (SPIMn) 的位7 (SPIEn) 和位6 (TRMDn) 置为1时, 通过将数据写入SDR0n 开始发送/接收。

串行I/O移位寄存器把SDR0n中的数据从并行数据转换成串行数据, 并输出到串行输出引脚。

可用8位或16位存储操作指令写入或读取SDR0n。

产生复位信号将该寄存器清除为0000H。

图19-5 发送缓冲寄存器 (SDR0n) 的格式



### 19.3.6 接收缓冲寄存器 (SDR1n)

该寄存器存储接收到的数据。

如果将串行操作模式寄存器 (SPIMn) 的位6 (TRMDn) 置为0, 则通过从SDR1中读取数据开始接收。

接收期间, 将数据从串行输入引脚读入SDR1n中。

可用8位或16位存储操作指令读取SDR1n。

产生复位信号将该寄存器清除为0000H。

图19-6 接收缓冲寄存器 (SDR1n) 的格式



备注: n=0

### 19.3.7 SPI引脚的端口功能的控制寄存器

使用SPI时，必须设定与SPI输入输出引脚复用的端口功能的控制寄存器（端口模式寄存器（PMxx，PMCxx）。详细内容请参照“2.3.1 端口模式寄存器（PMxx）”。

在将SPI引脚的复用端口用作SCK/SO/MO的输出时，必须将各端口对应的端口模式寄存器（PMxx，PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位置“0”。在将SPI引脚的复用端口用作SCK/SI/MI的输入时，必须将各端口对应的端口模式寄存器（PMxx）的位“1”，PMCxx的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。详细内容请参照“2.5 使用复用功能时的寄存器设定”。

Draft Version  
Cmsemicon Confidential

## 19.4 串行接口SPI的操作

3线串行I/O模式中，数据以8位或16位为单位发送或接收。数据各位的发送或接收与串行时钟同步。

开始通信后，SPISn的位0（SPTFn）被置为1。当数据的通信已完成时，设置通信完成中断请求标志（SPIIFn），并将SPTFn清除为0。然后使能下一次通信。

注意事项

- 1.SPTFn=1（串行通信期间）时，禁止访问控制寄存器和数据寄存器。
- 2.必须在满足SCLK周期时间（tKCY）特性的范围内使用。详细内容请参照数据手册。

备注：n=0

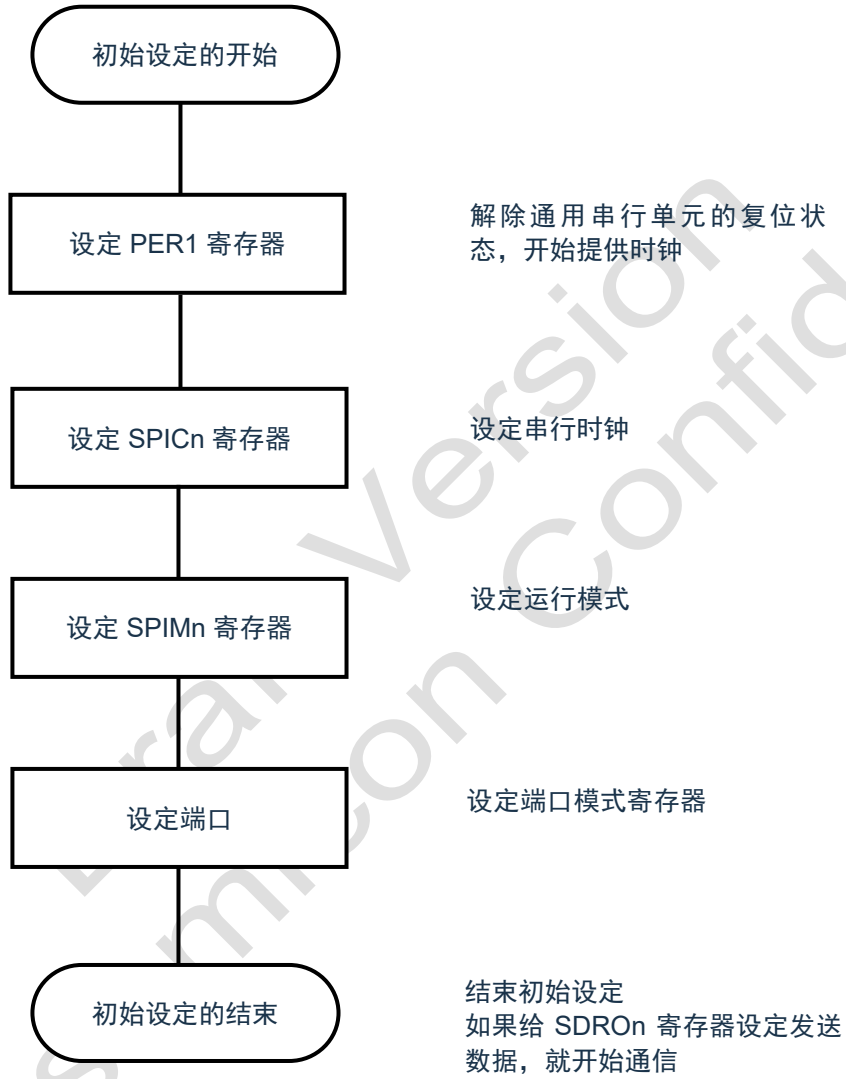
Draft Version  
Cmsemicon Confidential

### 19.4.1 主控的发送和接收

如果串行操作模式寄存器（SPIMn）的位6（TRMDn）为1，则可以发送或接收数据。当将某个值写入发送缓冲寄存器（SDR0n）时，开始发送/接收。

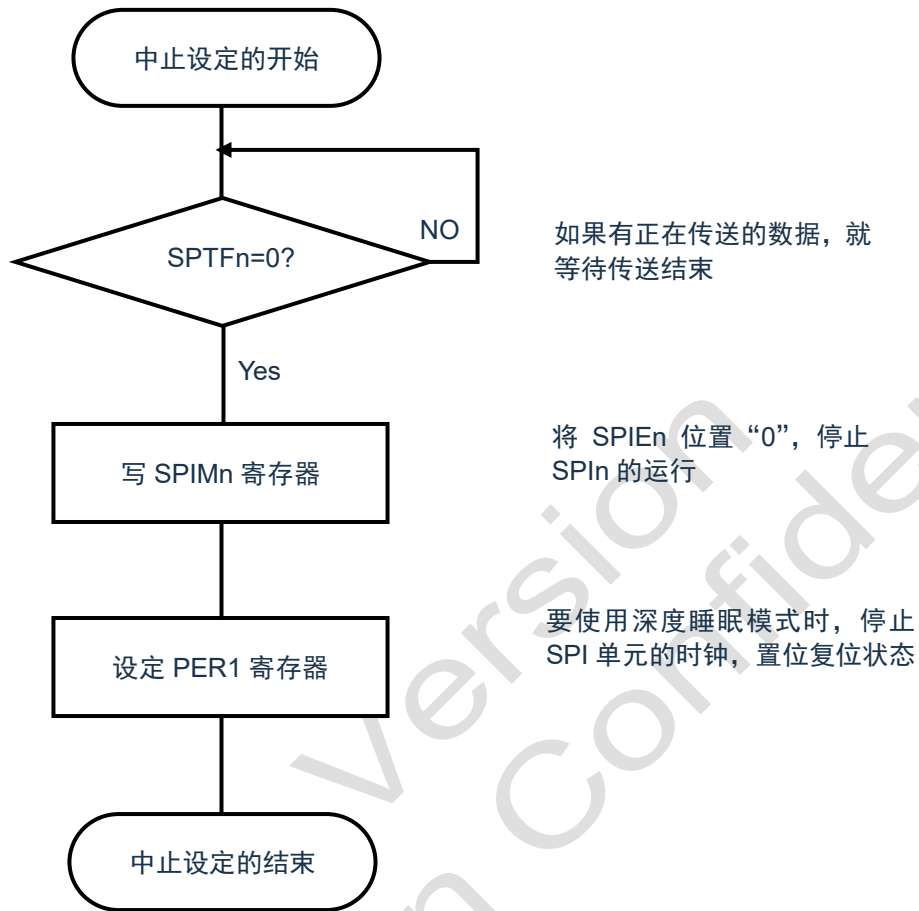
(1) 操作步骤

图19-7 主控发送/接收的初始设置步骤



备注：n=0

图19-8 主控发送/接收的中止步骤



备注：n=0



(2) 处理流程

图19-9 发送/接收时序(单次发送模式)的时序图 (INTMD=0,CPHA=1、CPOL=1)

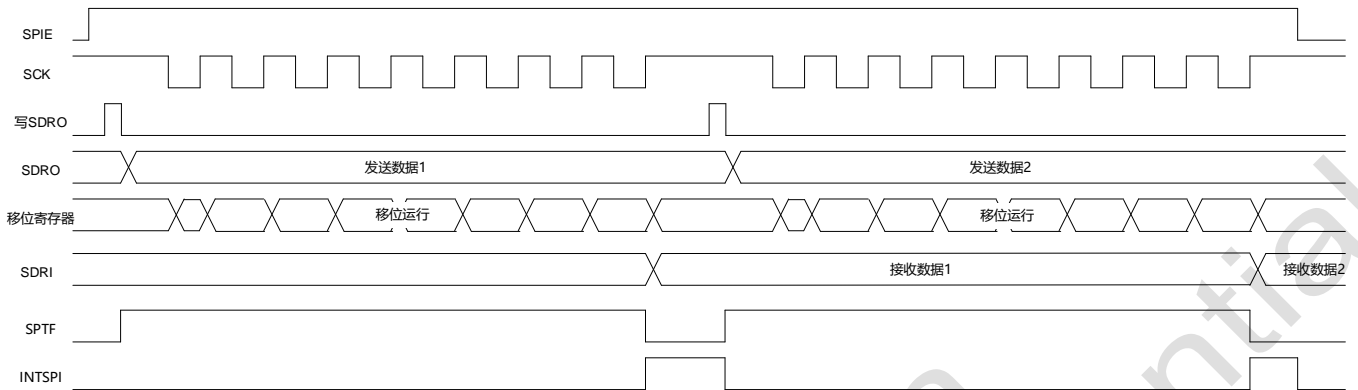
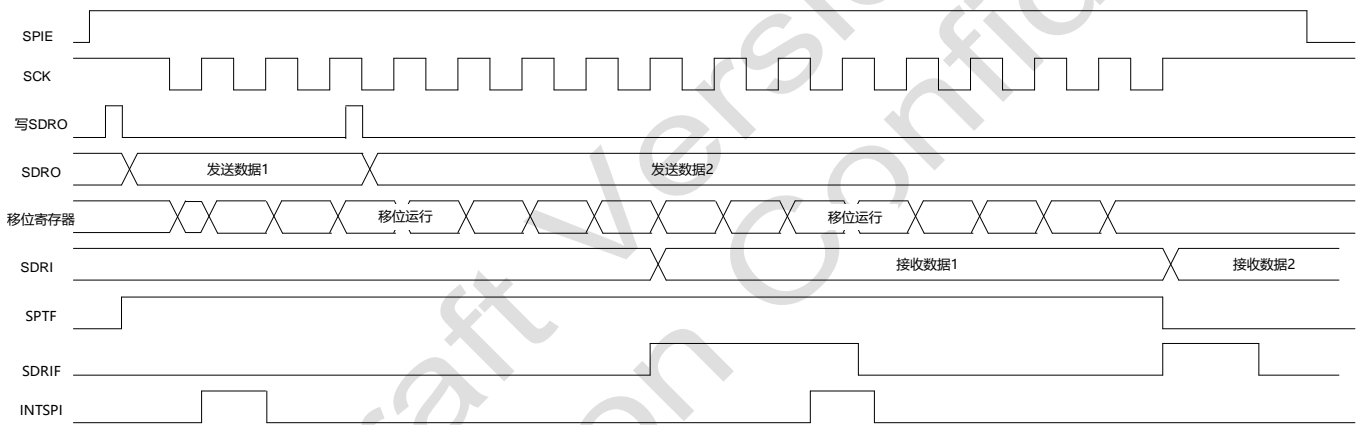


图19-10 发送/接收时序(连续发送模式)的时序图 (INTMD=1,CPHA=1、CPOL=1)

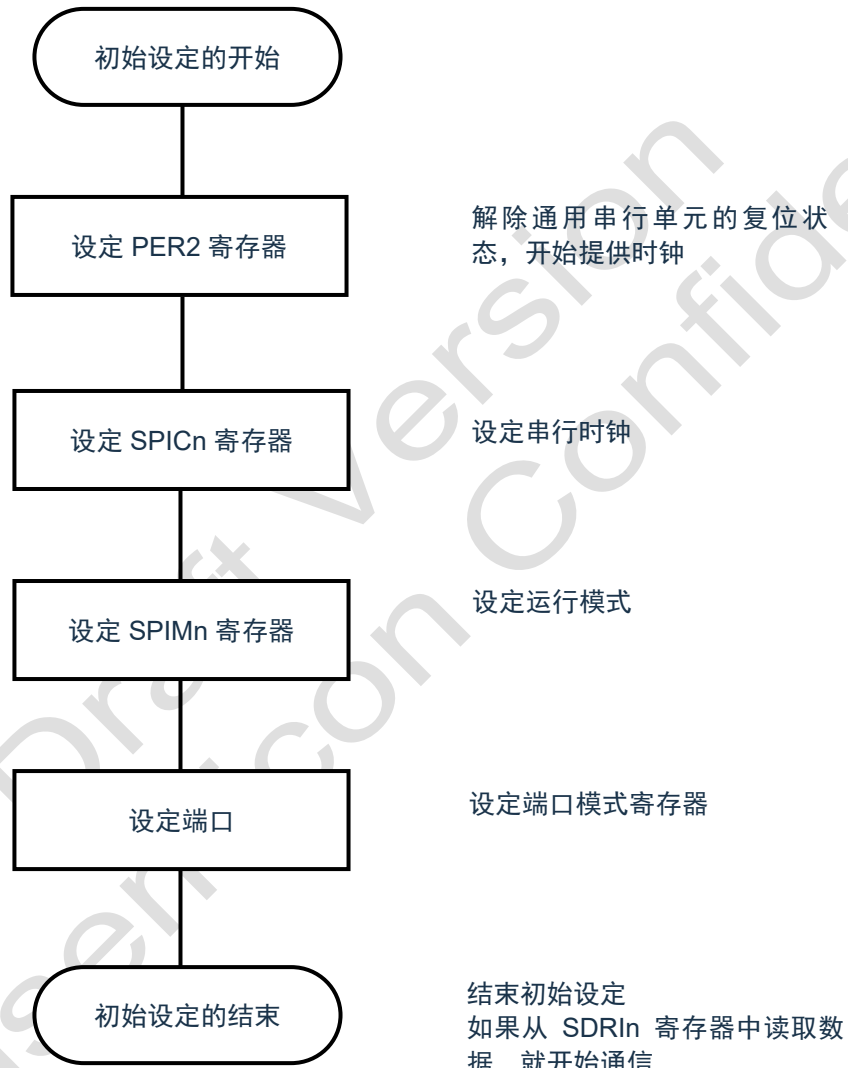


## 19.4.2 主控的接收

如果串行操作模式寄存器 (SPIMn) 的位6 (TRMDn) 为0, 则可以只接收数据。当从接收缓冲寄存器 (SDRIn) 中读取数据时, 开始接收。

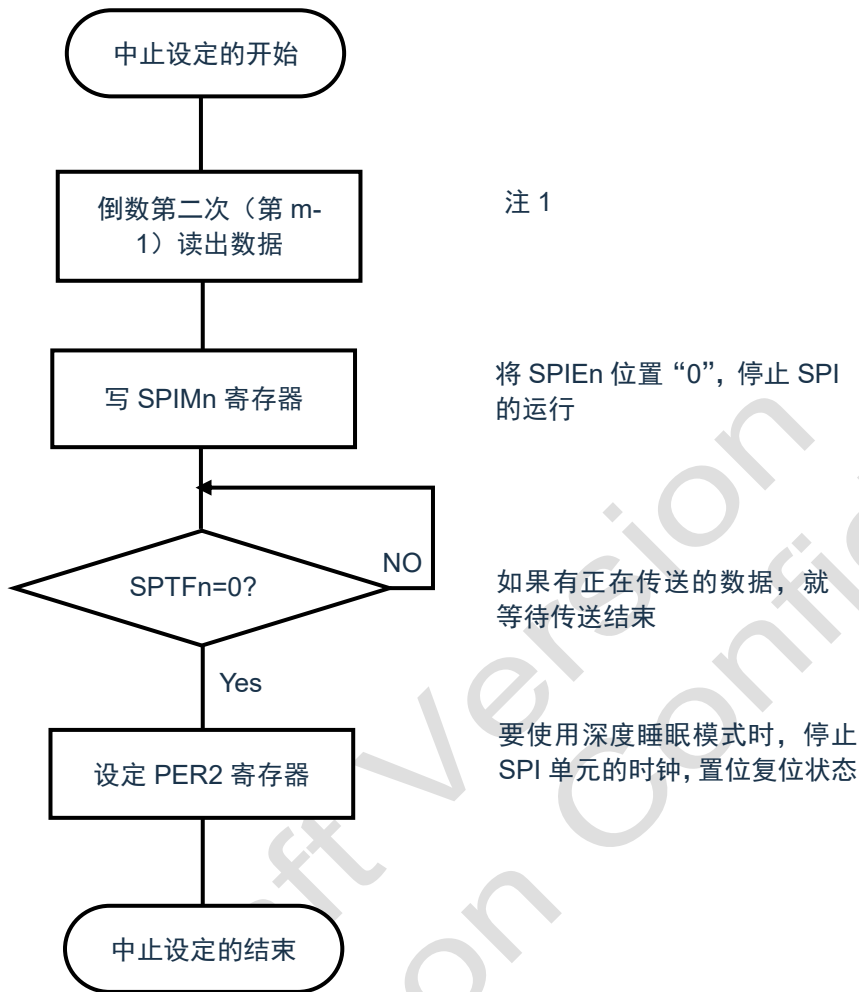
### (1) 操作步骤

图19-11 主控接收的初始设置步骤



备注: n=0

图19-12 主控接收的中止步骤

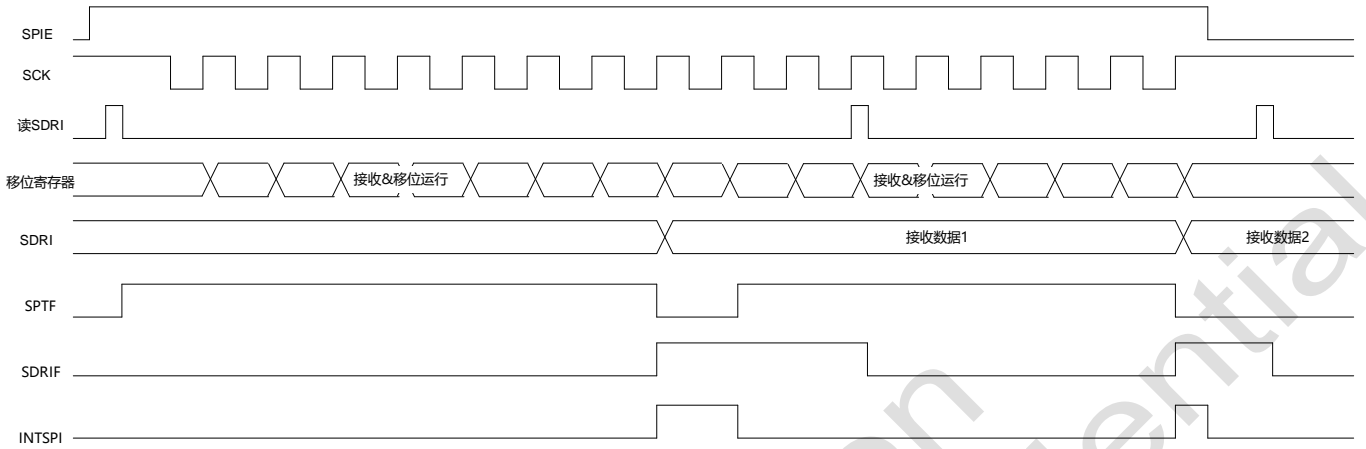


注 1：只接收模式下，SPI 传输由读取 SDRIn 寄存器的值触发。如果不及时中止 SPI 的动作，可能会在最后一次读取 SDRIn 之后有一次冗余的传输。如果想避免最后一次冗余的传输，可以在倒数第二次读出数据之后，等待一个 SCK 周期后关闭 SPIEn。SPI 的传输将在最后一个数据传输完成后中止。

备注：n=0

(2) 处理流程

图19-13 接收时序的时序图 (CPHA=1、CPOL=1)



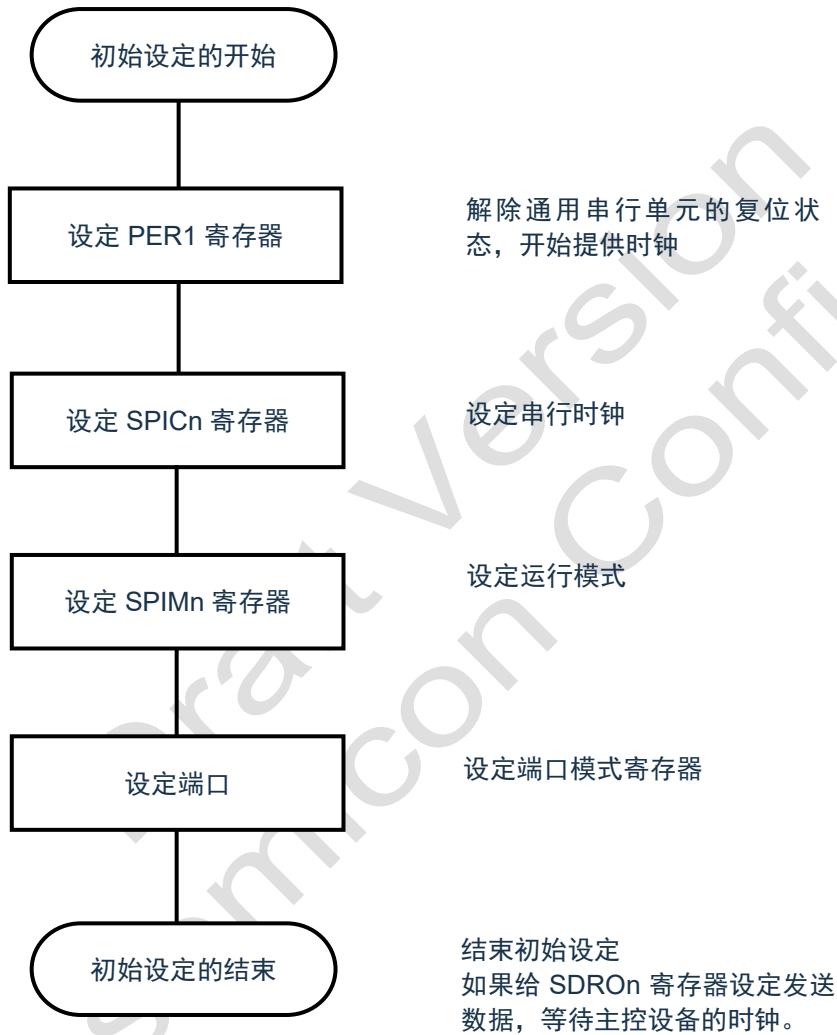
Draft Version  
Cmsemicon Confidential

### 19.4.3 从属的发送和接收

如果串行时钟选择寄存器（SPICn）的CKS2-0位选择从机模式，串行操作模式寄存器（SPIMn）的位6（TRMDn）为1，则进入从机发送/接收模式。当将某个值写入发送缓冲寄存器（SDRON）后，等待主控设备的时钟，开始发送/接收。

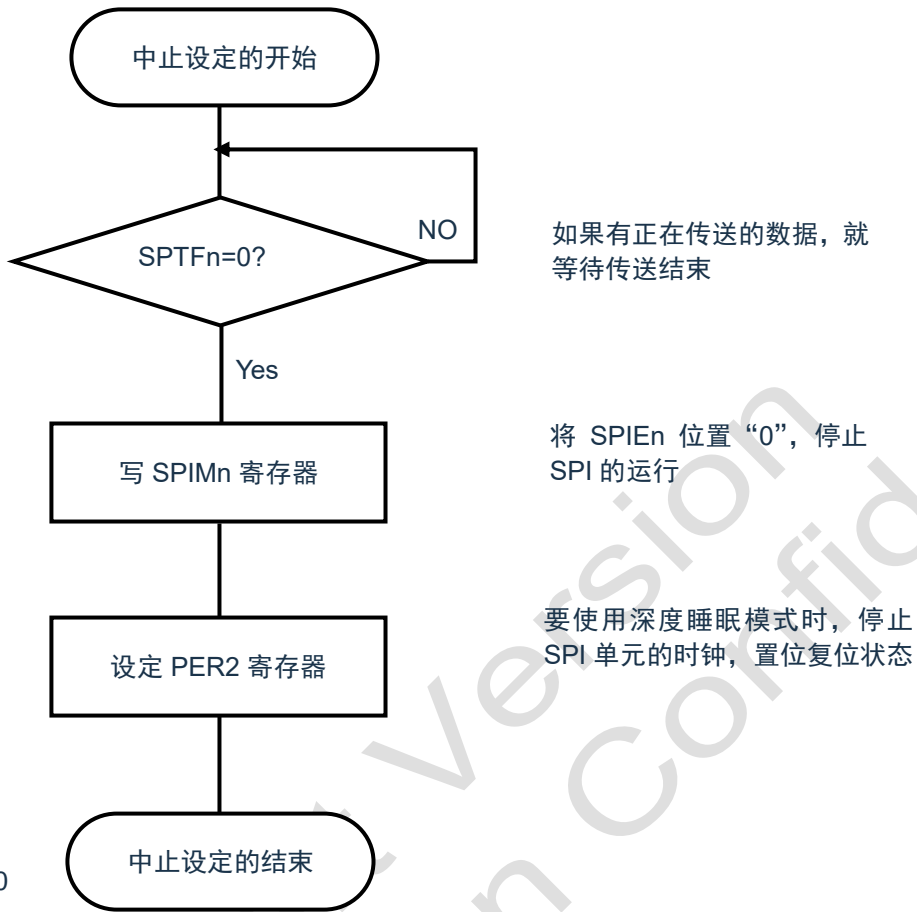
(1) 操作步骤

图19-14 从属发送/接收的初始设置步骤



备注：n=0

图19-15 从属发送/接收的中止步骤



(2) 处理流程

图19-16 发送/接收时序(单次发送模式)的时序图 (INTMD=0,CPHA=1、CPOL=1)

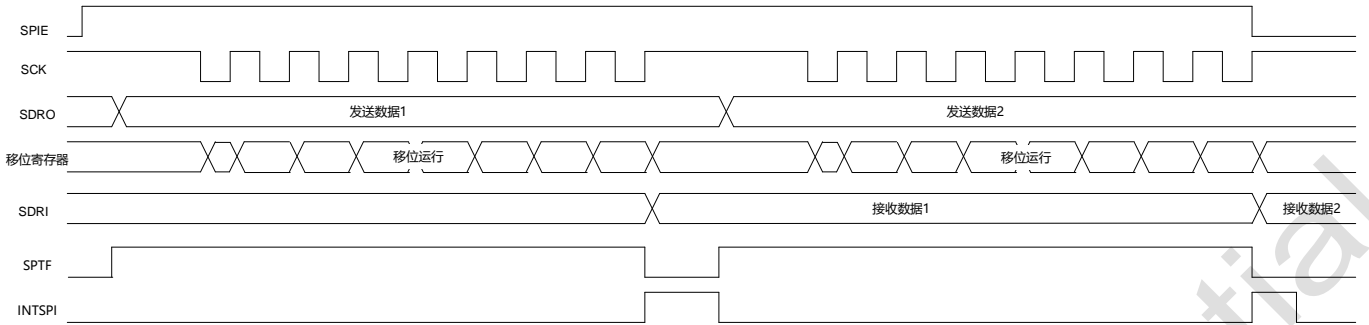
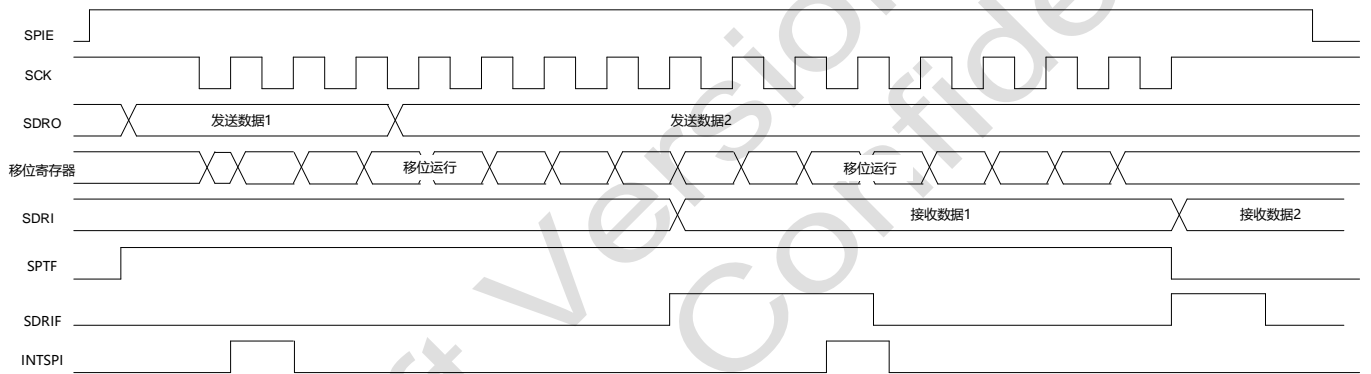


图19-17 发送/接收时序(连续发送模式)的时序图 (INTMD=1, CPHA=1、CPOL=1)

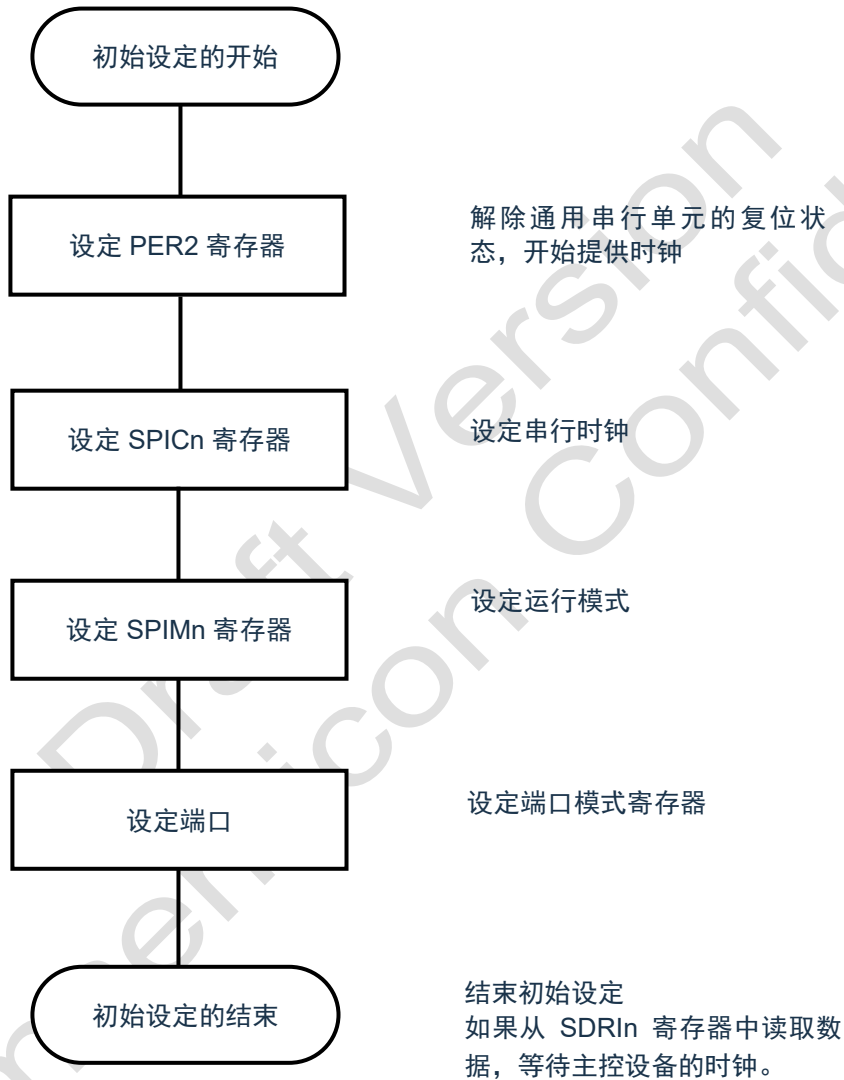


### 19.4.4 从属接收

如果串行时钟选择寄存器（SPICn）的CKS2-0n位选择从机模式，串行操作模式寄存器（SPIMn）的位6（TRMDn）为0，则进入从机接收模式。当从接收缓冲寄存器（SDRIn）中读取数据时，等待主控设备的时钟，开始接收。

(1) 操作步骤

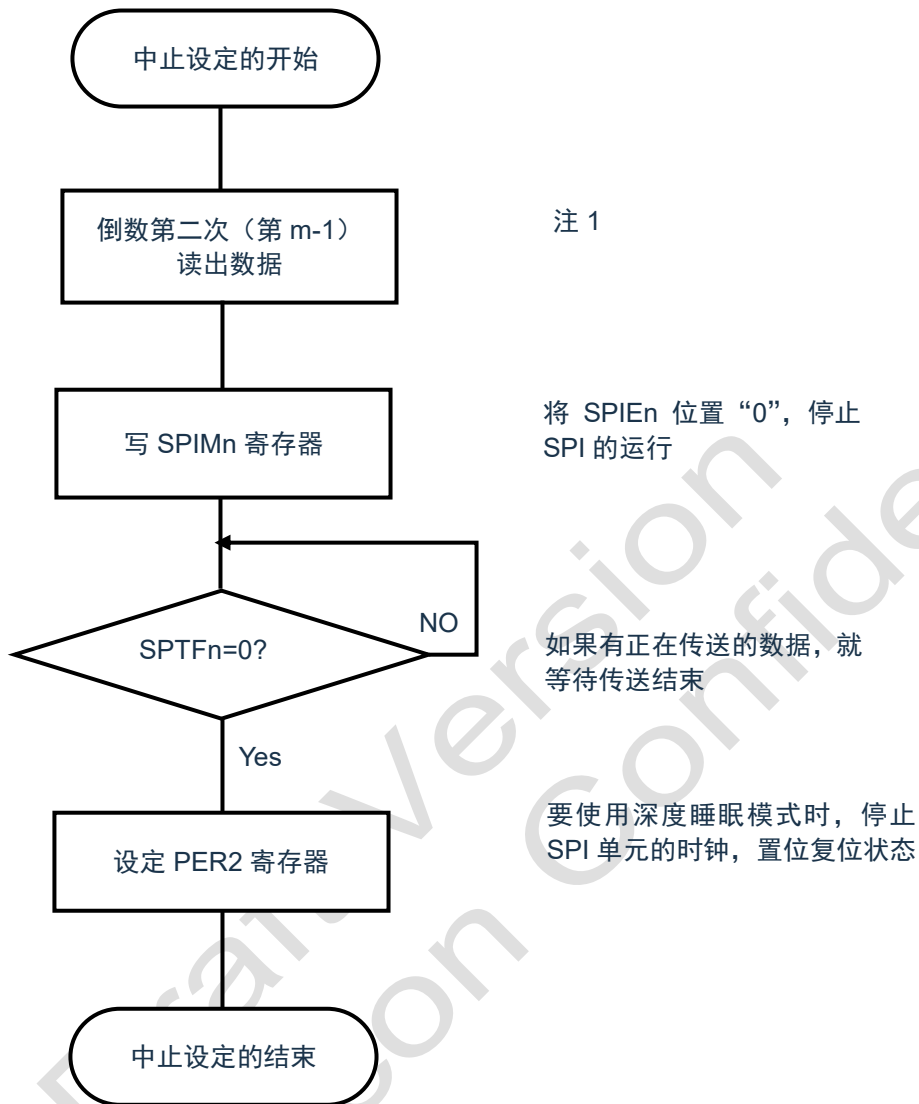
图19-18 从属接收的初始设置步骤



备注：n=0



图19-19 从属接收的中止步骤



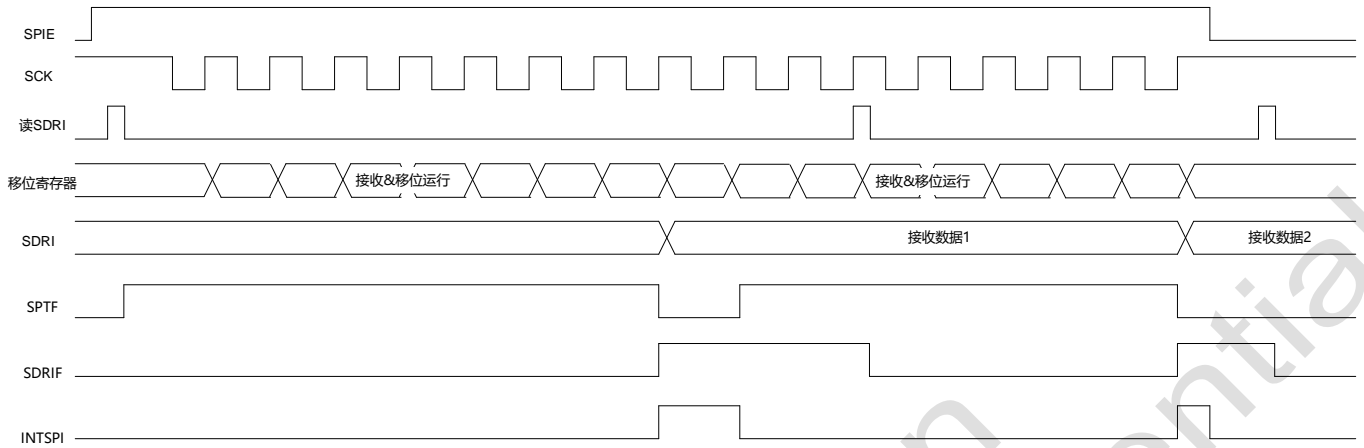
注 1: 只接收模式下，SPI 传输由读取 SDRIn 寄存器的值触发。如果不及时中止 SPI 的动作，可能会在最后一次读取 SDRIn 之后有一次冗余的传输。

如果想避免最后一次冗余的传输，可以在倒数第二次读出数据之后，等待一个 SCK 周期后关闭 SPIEn。SPI 的传输将在最后一个数据传输完成后中止。

备注：n=0

(2) 处理流程

图19-20 接收时序的时序图 (CPHA=1、CPOL=1)



Draft Version  
Cmsemicon Confidential

## 第20章 LIN/UART模块(LIN)

### 20.1 概述

LIN/UART模块是一个硬件类LIN通信控制器，支持LIN规范协议包括1.3、2.0、2.1、2.2和SAEJ 2602版本，并具有自动进行帧间通信和错误判断的功能。

LIN/UART模块提供了UART模式，也可以当作单个的UART来使用。

表20-1是LIN/UART模块的规格说明，图 20-1和图 20-2是LIN/UART模块的结构图。

表20-1.LIN/UART模块规格

项目	规格		
通道数	1 个通道 (本产品搭载 1 个通道, 故本章中 n=0)		
Lin 通讯功能	协议	LIN 规范协议包含修订版 1.3、2.0、2.1、2.2 和 SAEJ 2602	
	可变的帧结构	主机	<ul style="list-style-type: none"> <li>•间隔 (低) 传输宽度: 13 至 28 Tbits</li> <li>•间隔符的传输宽度: 1 至 4 Tbits</li> <li>•字节间隔 (帧头): 0 到 7 Tbits (同步场和 ID 场之间的间隔) <sup>※1</sup></li> <li>•响应间隔: 0 至 7 Tbits <sup>※1</sup></li> <li>•字节间隔: 0 至 3 Tbits (响应区数据字节间的间隔)</li> <li>•唤醒: 1 至 16 Tbits</li> </ul>
		从机	<ul style="list-style-type: none"> <li>•间隔接收宽度: 9.5 或 10.5Tbits[用于固定波特率] : 10 或 11Tbits[用于自动波特率]</li> <li>•响应间隔: 0 至 7Tbits</li> <li>•字节间隔: 0 至 3Tbits (响应区数据字节间的间隔)</li> <li>•唤醒: 1 至 16Tbits</li> </ul>
	校验和	<ul style="list-style-type: none"> <li>•发送和接收都是自动操作的</li> <li>•可选择经典型或增强型 (对每一帧)</li> </ul>	
	响应场数据字节数	0 到 8 字节之间 也可以进行多字节 (9 个或更多字节) 的响应传输和接收	
	帧通信模式	主机	<ul style="list-style-type: none"> <li>•通过单个传输启动请求开始帧头传输和响应传输或接收的模式</li> <li>•通过单个传输启动请求开始帧头传输和响应传输的模式 (帧分离模式)。</li> </ul>
		从机	<ul style="list-style-type: none"> <li>•以固定的波特率自动接收帧头的模式</li> <li>•根据检测到的同步场和间隔场的测量结果设置波特率, 自动接收帧头的模式。</li> </ul>
	唤醒传输和接收	提供 LIN 唤醒模式 <ul style="list-style-type: none"> <li>•唤醒传输 (1 至 16T 位)</li> <li>•唤醒接收</li> </ul> 测量的输入信号的低宽度	
状态	主机	<ul style="list-style-type: none"> <li>•成功的帧/唤醒传输</li> <li>•成功的帧头传输</li> <li>•成功的帧/唤醒接收注 2</li> <li>•成功接收数据 1</li> <li>•错误检测</li> <li>•操作模式</li> </ul> (LIN 复位模式、LIN 唤醒模式、LIN 工作模式、LIN 自测模式)	

项目		规格	
Lin 通讯功能	状态	从机	<ul style="list-style-type: none"> <li>•成功的帧/唤醒传输</li> <li>•成功的帧/唤醒接收注 2</li> <li>•成功接收帧头文件</li> <li>•成功接收数据 1</li> <li>•错误检测</li> <li>•操作模式 (LIN 复位模式、LIN 唤醒模式、LIN 工作模式、LIN 自测模式)</li> </ul>
	错误状态	主机	<ul style="list-style-type: none"> <li>•位错误</li> <li>•校验和错误</li> <li>•帧超时错误/响应超时错误</li> <li>•物理总线错误</li> <li>•帧错误</li> <li>•响应准备错误</li> </ul>
		从机	<ul style="list-style-type: none"> <li>•位错误</li> <li>•校验和错误</li> <li>•帧超时错误/响应超时错误</li> <li>•同步场错误</li> <li>•ID 奇偶校验错误</li> <li>•帧错误</li> <li>•响应准备错误</li> </ul>
	波特率选择	使用波特率发生器生成符合 LIN 规范的波特率	
	测试模式	用于用户评估的自测模式	
	中断功能	主机	<ul style="list-style-type: none"> <li>•成功的帧头/帧/唤醒传输</li> <li>•成功的帧/唤醒接收注 2</li> <li>•错误检测</li> </ul>
		从机	<ul style="list-style-type: none"> <li>•成功的帧/唤醒传输</li> <li>•帧头/帧/唤醒接收注 2</li> <li>•错误检测</li> </ul>

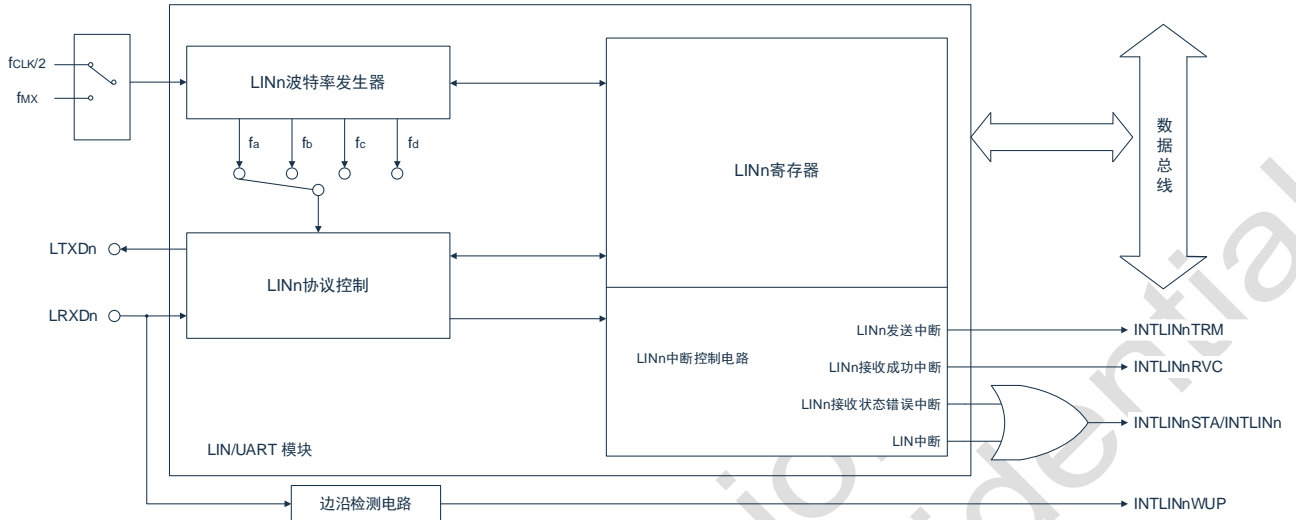
项目	规格	
UART 通讯功能	数据缓冲	<ul style="list-style-type: none"> <li>•传输数据缓冲/用于等待的传输数据缓冲（专门用于传输；支持数据长度为 1、7、8 和 9 位）</li> <li>•UART 缓冲（专门用于传输；数据长度从 1 到 9 位不等；支持数据长度为 7 位和 8 位）</li> <li>•接收数据缓冲（专门用于接收；支持数据长度为 1、7、8 和 9 位）</li> </ul>
	数据格式	<ul style="list-style-type: none"> <li>•字符长度：7 或 8 位 支持 9 位，包括扩展位。</li> <li>•传输停止位：1 或 2 位</li> <li>•奇偶校验功能：奇数、偶数、0 或无</li> <li>•可选择 LSB-或 MSB-优先传输</li> <li>•发送/接收数据的反向输入/输出</li> </ul>
	状态	<ul style="list-style-type: none"> <li>•传送状态</li> <li>•接收状态</li> <li>•成功的 UART 缓冲传输</li> <li>•误差总和</li> <li>•扩展位检测</li> <li>•ID 匹配</li> <li>•复位模式状态</li> </ul>
	错误状态	<ul style="list-style-type: none"> <li>•位错误</li> <li>•帧错误</li> <li>•奇偶校验错误</li> <li>•过载运行错误</li> </ul>
	波特率选择	有了内置的波特率发生器，可设置不同的波特率。
		如果检测到任何一个扩展位的预期电平，则可将收到的 8 位数据与预设的寄存器数据进行比较。接收到的停止位受到保证（在接收停止位期间试图开始传输时，开始传输可能会推迟）。
	中断功能	<ul style="list-style-type: none"> <li>•传送开始/传送成功</li> <li>•成功接收</li> <li>•状态检测</li> </ul>

**注：1.**由于设置时使用的是同一个寄存器，所以字节间隔（帧头）=响应间隔。

**2.**对于唤醒接收，指示输入信号的低电平宽度。

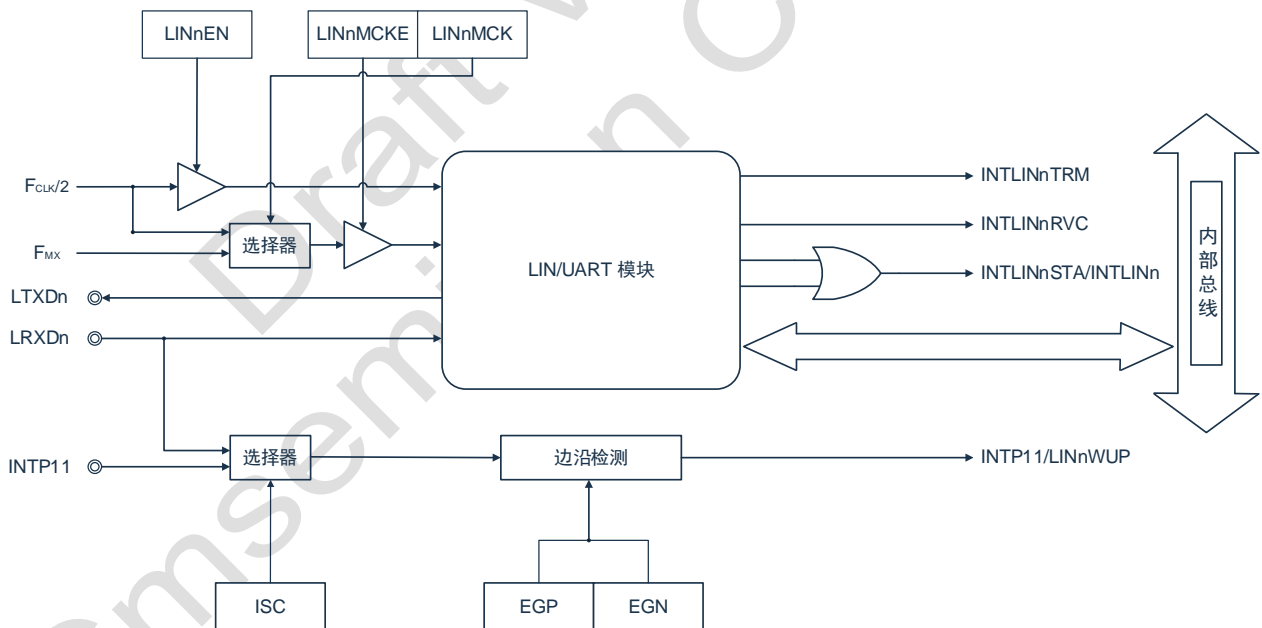
注意：本产品搭载了1通道LIN，故本章中所有表述中n=0。

图 20-1.LIN/UART结构图(1)



- LTXDn,LRXDn: LIN/UART模块I/O引脚
- LINn波特率发生器: 生成LIN/UART模块通信时钟信号。
- LINn寄存器: LIN/UART模块寄存器
- LINn中断控制器: 由LIN/UART模块生成的控制中断请求。(n=0)

图 20-2.LIN/UART结构图(2)



注： n=0

表20-2是LIN/UART模块中使用的I/O引脚：

表20-2.LIN/UART模块的I/O引脚

模块符号	引脚名称	输入/输出	功能
LINn	LRXDn	输入	LIN 通信功能 UART 通信功能的输入引脚
	LTXDn	输出	LIN 通信功能 UART 通信功能的输出引脚

(n=0)

应根据应用情况为LIN/UART模块选择运行适当的模式：LIN主机、LIN从机或UART。

#### LIN主机

- LIN复位模式
- LIN模式（LIN主机模式）
- LIN唤醒模式
- LIN工作模式
  - LIN自测模式

#### LIN从机

- LIN复位模式
- LIN模式（LIN从机模式[自动波特率]或LIN从机模式[固定波特率]）
- LIN唤醒模式
- LIN工作模式
  - LIN自测模式

#### UART

- LIN复位模式
- UART模式

## 20.2 寄存器描述

表20-3列出与LIN/UART模块相关的寄存器，BASE = 0x40045C00。

表20-3.与LIN/UART模块相关的寄存器列表

寄存器名称	符号	寄存器地址	LIN 主机	LIN 从机	UART
外围寄存器 2	PER2	4002081BH	√	√	√
输入切换控制寄存器	ISC	40020473H	√	√	√
LIN 时钟选择寄存器	LINCKSEL	4002081CH	√	√	√
外部中断上升沿允许寄存器 1	EGP1	40045B3AH	√	√	√
外部中断下降沿允许寄存器 1	EGN1	40045B3BH	√	√	√
LIN 唤醒波特率选择寄存器	LWBRn	BASE+01H	√	√	√
LIN/UART 波特率预分频器寄存器	LBRPn	BASE+02H	-	√	√
LIN/UART 波特率预分频器 0 寄存器	LBRPn0	BASE+02H	√	√	√
LIN/UART 波特率预分频器 1 寄存器	LBRPn1	BASE+03H	√	√	√
LIN 自测控制寄存器	LSTCn	BASE+04H	√	√	-
UART 待机控制寄存器	LUSCn	BASE+05H	-	-	√
LIN/UART 模式寄存器	LMDn	BASE+08H	√	√	√
LIN 间隔场配置寄存器/UART 配置寄存器	LBFCn	BASE+09H	√	√	√
LIN/UART 空间配置寄存器	LSCn	BASE+0AH	√	√	√
LIN 唤醒配置寄存器	LWUPn	BASE+0BH	√	√	-
LIN 中断允许寄存器	LIE n	BASE+0CH	√	√	-
LIN/UART 错误检测允许寄存器	LEDE n	BASE+0DH	√	√	√
LIN/UART 控制寄存器	LCUCn	BASE+0EH	√	√	√
LIN/UART 传输控制寄存器	LTRCn	BASE+10H	√	√	√
LIN/UART 模式状态寄存器	LMSTn	BASE+11H	√	√	√
LIN/UART 状态寄存器	LSTn	BASE+12H	√	√	√
LIN/UART 错误状态寄存器	LESTn	BASE+13H	√	√	√
LIN/UART 数据场配置寄存器	LDFCn	BASE+14H	√	√	√
LIN/UARTID 缓冲寄存器	LIDBn	BASE+15H	√	√	√
LIN 校验和缓冲寄存器	LICBRn	BASE+16H	√	√	-
UART 数据缓冲区 0 寄存器	LUDBn0	BASE+17H	-	-	√
LIN/UART 数据缓冲区 1 寄存器	LDBn1	BASE+18H	√	√	√
LIN/UART 数据缓冲区 2 寄存器	LDBn2	BASE+19H	√	√	√
LIN/UART 数据缓冲区 3 寄存器	LDBn3	BASE+1AH	√	√	√
LIN/UART 数据缓冲区 4 寄存器	LDBn4	BASE+1BH	√	√	√
LIN/UART 数据缓冲区 5 寄存器	LDBn5	BASE+1CH	√	√	√
LIN/UART 数据缓冲区 6 寄存器	LDBn6	BASE+1DH	√	√	√
LIN/UART 数据缓冲区 7 寄存器	LDBn7	BASE+1EH	√	√	√
LIN/UART 数据缓冲区 8 寄存器	LDBn8	BASE+1FH	√	√	√
UART 工作允许寄存器	LUOERn	BASE+20H	-	-	√
UART 选项寄存器 1	LUORn1	BASE+21H	-	-	√



寄存器名称	符号	寄存器地址	LIN 主机	LIN 从机	UART
UART 传输数据寄存器	LUTDRn	BASE+24H	-	-	√
	LUTDRnL	BASE+24H	-	-	√
	LUTDRnH	BASE+25H	-	-	√
UART 接收数据寄存器	LURDRn	BASE+26H	-	-	√
	LURDRnL	BASE+26H	-	-	√
	LURDRnH	BASE+27H	-	-	√
UART 等待传输数据寄存器	LUWTDRn	BASE+28H	-	-	√
	LUWTDRnL	BASE+28H	-	-	√
	LUWTDRnH	BASE+29H	-	-	√

√: 使用过

-: 未使用

当写到未使用的寄存器时, 写入00H。

## 20.2.1 主机模式的LIN寄存器

### 20.2.1.1 输入切换控制寄存器 (ISC)

ISC寄存器中的ISC2位用于LIN/UART模块 (LIN)。

位2设置为1，选择LIN/UART模块的串行数据输入引脚的输入信号作为外部中断输入。

这个寄存器可以通过8位的内存操作指令来设置。

复位信号产生将此寄存器设置为00H。

符号	复位值: 00H								R/W
	7	6	5	4	3	2	1	0	
ISC	0	0	0	0	0	ISC2	ISC1	ISC0	

ISC2	外部中断 INTP11 的输入切换
0	将 INTP11 引脚的输入信号用作外部中断的输入。
1	将 LRXD0 引脚的输入信号用作外部中断的输入。

**注意事项** 第7至3位应始终设置为0。

### 20.2.1.2 外围寄存器2 (PER2)

PER2寄存器用于允许或禁止对外围硬件的时钟供应。停止向未使用的硬件提供时钟，以降低功耗和噪声。要使用此寄存器控制的外围功能，在指定外围功能的初始设置之前，请为每个功能的对应位设置“1”。

这个寄存器可以由8位的内存操作指令来设置。

复位信号产生将此寄存器设置为00H。

符号	复位值: 00H								R/W
	7	6	5	4	3	2	1	0	
PER2	0	0	0	0	0	0	SPIEN	LINEN	

LINEN	LIN模块的输入时钟控制
0	停止提供输入时钟。 •不能写LIN模块使用的SFR。 •LIN模块处于复位状态。
1	提供输入时钟。 •能读写LIN模块使用的SFR。

### 20.2.1.3 LIN时钟选择寄存器 (LINCKSEL)

该寄存器用于控制提供给LIN的通信时钟源。

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LINCKSEL	0	0	0	LIN0MCKE	0	0	0	LIN0MCK

LINnMCKE	供应或停止 LIN 通信时钟源的控制
0	停止 LIN 通信时钟源供应。
1	允许 LIN 通信时钟源供应。

LINnMCK	选择 LIN 通信时钟源的控制
0	选择 fCLK 的 2 分频时钟。
1	选择 fMX 时钟。

- 注意事项
- 1.在将LINnMCKE (n=0) 位设置为1 (提供工作时钟) 之前, 用LINnMCK位选择LINn工作时钟。
  - 2.当在SNOOZE模式下操作LINn时, 将LINnMCK位设为0。
  - 3.如果LINnMCK设置为1, 则不使用超时错误检测。  
在这种情况下, 将至少1.2倍的LIN通信时钟源的频率设置成fCLK时钟。

### 20.2.1.4 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)

更多细节, 详见24.3.3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)。

### 20.2.1.5 LIN唤醒波特率选择寄存器 (LWBRn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LWBRn	NSPB[3:0]			LPRS[2:0]			LWBR0	

位	符号	位名称	功能	R/W
0	LWBR0	唤醒波特率选择	0:使用 LIN1.3 时 1:使用 LIN2.x 时	RW
3 至 1	LPRS[2:0]	预分频器时钟选择	b3 b1 000:1/1 001:1/2 010:1/4 011:1/8 100:1/16 101:1/32 110:1/64 111:1/128	RW
7 至 4	NSPB[3:0]	位采样计数选择	b7 b4 0000: 16 采样 1111: 16 采样 禁止使用上述以外的设置。	RW

当LMSTn寄存器中的OMM0位为0（LIN复位模式）时，设置LWBRn寄存器。

#### LWBR0位（唤醒波特率选择位）

使用LIN规范包1.3修订版时，将LWBRn寄存器中的LWBR0位设置为0。这允许测量2.5-Tbit或更长的输入信号的低电平宽度。使用LIN规范包2.x修订版时，将LWBR0位设置为1。当LWBR0位设置为1时，无论LMDn寄存器中LCKS位的设置如何（LCKS位的设置不受影响），在LIN唤醒模式下始终选择fa作为LIN系统时钟（fLIN）。这允许2.5-Tbit或更长输入信号的低电平宽度得到测量。

无论LMDn寄存器中LCKS位的设置如何，将波特率设置为19200bps并选择fa，即允许在LIN唤醒模式下检测130 μs或更长的输入信号的低电平宽度。

#### LPRS位（预分频器时钟选择位）

LPRS位选择预分频器的分频比。

LIN通信时钟源的频率是根据这个预分频器划分的。

#### NSPB位（位采样计数选择位）

NSPB位选择单个Tbit采样次数（波特率的倒数）。

在LIN主机模式下（LIN/UART模式寄存器中的LIN/UART模式选择位=00b），设置NSPB位为0000b或1111b（16采样）。

### 20.2.1.6 LIN/UART波特率预分频器0寄存器 (LBRPn0)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LBRPn0								

位	功能	设置范围	R/W
7 至 0	假设该寄存器中设置的值为 N (0~255)，波特率预分频器 0 将预分频器时钟的频率除以 N + 1。	00H 至 FFH	R/W

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时，设置LBRP0寄存器。该寄存器中设置的值用于控制波特率时钟源fa、fb和fc的频率。

假设该寄存器中设置的值为N，波特率预分频器0将LPRS位选择的时钟频率除以N+1。

### 20.2.1.7 LIN/UART波特率预分频器1寄存器 (LBRPn1)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LBRPn1								

位	功能	设置范围	R/W
7 至 0	假设该寄存器中设置的值为 M (0~255)，波特率预分频器 1 将预分频器时钟的频率除以 M + 1。	00H 至 FFH	R/W

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时，设置LBRP1寄存器。

该寄存器中设置的值用于控制波特率时钟源fd的频率。

假设该寄存器中设置的值为M，波特率预分频器1将LPRS位 (预分频器时钟选择位) 选择的时钟频率除以M+1。

### 20.2.1.8 LIN自测控制寄存器 (LSTCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSTCn								LSTM

位	符号	位名称	功能	RW
7 至 0			将 A7H、58H 和 01H 连续写入该位，使 LIN/UART 模块进入 LIN 自测模式。	RW
0	LSTM	LIN 自测模式	0: 模块不在 LIN 自测模式中 1: 模块在 LIN 自测模式中	RW

LSTCn寄存器取消了对LIN自测模式的保护。

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LSTCn寄存器。

连续向LSTCn寄存器写入A7H、58H和01H，使模块进入LIN自测模式。当连续写入完成，从而进入LIN自测模式时，LSTM位设置为1。在连续写的过程中，不要写任何其他值。

关于转换到LIN自测模式，请参考20.6 LIN自测模式。

读取第6至1位返回000000b，读取第7位返回未定义值。

#### LSTM位（LIN自测模式位）

当转换到LIN自测模式完成后，LSTM位设置为1。

退出LIN自测模式，请参考20.6 LIN自测模式。

如果不是在连续写入A7H、58H和01H过程中，向该位写1不会影响LSTCn寄存器的值。

### 20.2.1.9 LIN/UART模式寄存器 (LMDn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LMDn	-	-	LRDNFS	LIOS	LCKS[1:0]	LMD[1:0]		

位	符号	位名称	功能	RW
1,0	LMD[1:0]	LIN/UART 模式选择	b1b0 00: LIN 主机模式	RW
3,2	LCKS[1:0]	LIN 系统时钟选择	b3b2 00:fa (由波特率预分频器 0 产生的时钟) 01:fb (由波特率预分频器 0 产生的 1/2 时钟) 10:fc (由波特率预分频器 0 产生的 1/8 时钟) 11:fd (由波特率预分频器 1 产生的 1/2 时钟)	RW
4	LIOS	LIN 中断输出选择	0: 使用 LIN 中断。 1: 使用传输中断、成功接收中断和接收状态中	RW
5	LRDNFS	禁止 LIN 接收数据过滤噪声	0: 允许噪声滤波器。 1: 禁止噪声滤波器。	RW
6	-	保留	该位总是读为 0, 写入值应始终为 0。	RW
7	-	保留	该位总是读为 0。写入值应始终为 0。	RW

当LMSTn寄存器中的OMM0位为0时 (LIN复位模式), 设置LMDn寄存器。

#### LMD[1:0]位 (LIN/UART模式选择位)

LMD位选择LIN/UART模块模式。

要将LIN/UART模块作为LIN主机使用, 需将该位设置为00b。

设置为00b时, LIN/UART模块在LIN主机模式下运行。

#### LCKS[1:0]位 (LIN系统时钟选择位)

LCKS位选择将要输入到协议控制器的时钟。

设置为00b时, 协议控制器提供fa (由波特率预分频器0产生的时钟)。

设置为01b时, 协议控制器提供fb (由波特率预分频器0产生的1/2时钟)。

设置为10b时, 协议控制器提供fc (由波特率预分频器0产生的1/8时钟)。

设置为11b时, 协议控制器提供fd (由波特率预分频器1产生的1/2时钟)。

当LWBRn寄存器中的LWBR0位为1 (使用LIN2.x) 且LMSTn寄存器为01h (LIN唤醒模式) 时, 无论LCKS位的设置如何 (LCKS位的设置不受影响), fa总是输入到协议控制器。

#### LIOS位 (LIN中断输出选择位)

LIOS位选择LIN/UART模块中断输出的数量。

设置为0时, LIN中断由LIN/UART模块产生。

设置为1时, LIN/UART模块产生传输中断、成功接收中断和接收状态中断。

对于每一个中断源, 请参考20.9中断。

#### LRDNFS位 (LIN接收数据噪声过滤禁止位)

LRDNFS位在接收数据时允许或禁止噪声滤波器。

设置为0时, 接收数据时允许噪声滤波器。

设置为1时, 接收数据时禁止噪声滤波器。

### 20.2.1.10 LIN间隔场配置寄存器/UART配置寄存器 (LBFCn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LBFCn	-	-	BDT[1:0]		BLT[3:0]			

位	符号	位名称	功能	RW
3至0	BLT[3:0]	传输间隔(低)宽度选择	b3b0 0000:13Tbits 0001:14Tbits 0010:15Tbits : 1110:27Tbits 1111:28Tbits	R/W
5,4	BDT[1:0]	传输间隔符(高)宽度选择	b5b4 00:1Tbit 01:2Tbits 10:3Tbits 11:4Tbits	R/W
6	-	保留	该位总是读为0, 写入值应始终为0。	R/W
7	-	保留	该位总是读为0, 写入值应始终为0。	R/W

当LMSTn寄存器中的OMM0位为0时(LIN复位模式), 设置LBFCn寄存器。

某些设置值的组合会导致帧的长度超过帧超时时间, 因此需在这个寄存器中设置适当的值。

#### BLT[3:0]位(传输间隔(低)宽度选择位)

BLT位设置传输帧头的间隔(低)宽度。

可以设置13Tbits到28Tbits。

#### BDT位(传输间隔符(高)宽度选择位)

BDT位设置传输帧头场的间隔符(高)宽度。可以设置1Tbit到4Tbit。



### 20.2.1.11 LIN/UART空间配置寄存器 (LSCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSCn	-	-	IBS[1:0]		-	IBSH[2:0]		

位	符号	位名称	功能	R/W
2至0	IBSH[2:0]	字节间隔（帧头）/ 响应间隔选择	b2 b0 000:0 Tbit 001:1 Tbit 010:2 Tbits 011:3 Tbits 100:4 Tbits 101:5 Tbits 110:6 Tbits 111:7 Tbits	RW
3	-	保留	该位总是读为 0，写入值应始终为 0。	RW
5,4	IBS[1:0]	字节间隔选择	b5b4 00:0Tbit 01:1Tbit 10:2Tbits 11:3Tbits	RW
7,6	-	保留	该位总是读为 0，写入值应始终为 0。	RW

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LSCn寄存器。

某些设定值的组合会导致帧的长度或响应超过超时时间，因此需在这个寄存器中设置适当的值。

#### IBSH[2:0]位（字节间隔（帧头）/响应间隔选择位）

IBSH位设置传输帧头场的字节间隔（帧头）和响应间隔的宽度。

可以设置0Tbit到7Tbit。

响应间隔设置仅在响应传输过程中允许；在响应接收过程中禁止。

字节间隔（帧头）与响应间隔相等。

#### IBS[1:0]位（字节间隔选择位）

IBS位设置传输帧响应场字节间隔的宽度。

可以设置0Tbit到3Tbit。

该位仅在响应传输过程中允许；在响应接收过程中禁止。

### 20.2.1.12 LIN唤醒配置寄存器 (LWUPn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LWUPn	WUTL[3:0]				-	-	-	-

位	符号	位名称	功能	RW
3至0	-	保留	该位总是读为0, 写入值应始终为0。	RW
7至4	WUTL[3:0]	唤醒传输低宽度选择	b7 b4 0000:1 Tbit 0001:2 Tbits 0010:3 Tbits 0011:4 Tbits : 1100:13 Tbits 1101:14 Tbits 1110:15 Tbits 1111:16 Tbits	RW

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 设置LWUPn寄存器。

#### WUTL[3:0]位 (唤醒传输低宽度选择位)

WUTL位设置唤醒信号传输的低宽度。

可以设置1Tbit到16Tbit。

当LWBRn寄存器中的LWBR0位为1时 (使用LIN2.x), 无论LMDn寄存器中的LCKS位如何设置 (LCKS位的设置不受影响), fa总是选为LIN系统时钟 (fLIN)。

### 20.2.1.13 LIN中断允许寄存器 (LIEn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LIEn	-	-	-	-	SHIE	ERRIE	FRCIE	FTCIE

位	符号	位名称	功能	RW
0	FTCIE	成功的帧/唤醒传输中断允许	0: 禁止成功的帧/唤醒传输中断。 1: 允许成功的帧/唤醒传输中断。	RW
1	FRCIE	成功的帧/唤醒接收中断允许	0: 禁止成功的帧/唤醒接收中断。 1: 允许成功的帧/唤醒接收中断。	RW
2	ERRIE	错误检测中断允许	0: 禁止错误检测中断。 1: 允许错误检测中断。	RW
3	SHIE	成功的帧头传输中断允许	0: 禁止成功的帧头传输中断。 1: 允许成功的帧头传输中断。	RW
7至4	-	保留	该位总是读为0, 写入值应始终为0。	RW

当LMSTn寄存器中的OMM0位为0时 (LIN复位模式), 设置LIEn寄存器。

#### FTCIE位 (成功的帧/唤醒传输中断允许位)

FTCIE位在成功的帧传输或唤醒信号时允许或禁止中断生成。

设置为0时, 当LSTn寄存器中的FTC标志设置为1时, 不产生中断。

设置为1时, 当LSTn寄存器中的FTC标志设置为1时, 产生中断。

#### FRCIE位 (成功的帧/唤醒接收中断允许位)

FRCIE位在成功接收到帧或唤醒信号 (计算输入信号的低宽度) 时允许或禁止中断产生。

设置为0时, 当LSTn寄存器中的FRC标志设置为1时, 不产生中断。

设置为1时, 当LSTn寄存器中的FRC标志设置为1时, 产生中断。

#### ERRIE位 (错误检测中断允许位)

ERRIE位在检测到错误时允许或禁止中断生成。

设置为0时, 当LSTn寄存器中的ERR标志设置为1时, 不产生中断。

设置为1时, 当LSTn寄存器中的ERR标志设置为1时, 产生中断。

中断源可以是位错误、物理总线错误、帧/响应超时错误、帧错误、校验和错误和响应准备错误。

位错误、物理总线错误、帧/响应超时错误和帧错误的检测可以通过LEDEn寄存器允许或禁止。

#### SHIE位 (成功的帧头传输中断允许位)

SHIE位在成功的帧头传输时允许或禁止中断生成。

设置为0时, 当LSTn寄存器中的HTRC标志设置为1时, 不产生中断。

设置为1时, 当LSTn寄存器中的HTRC标志设置为1时, 产生中断。

### 20.2.1.14 LIN/UART错误检测寄存器 (LEDEn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LEDEn	LTES	-	-	-	FERE	FTERE	PBERE	BERE

位	符号	位名称	功能	R/W
0	BERE	位错误检测允许	0: 禁止位错误检测。 1: 允许位错误检测。	R/W
1	PBERE	物理总线错误检测允许	0: 禁止物理总线错误检测。 1: 允许物理总线错误检测。	R/W
2	FTERE	超时错误检测允许	0: 禁止帧/响应超时错误检测。 1: 允许帧/响应超时错误检测。	R/W
3	FERE	帧错误检测允许	0: 禁止帧错误检测。 1: 允许帧错误检测。	R/W
4	-	保留	该位总是读为 0，写入值应始终为 0。	R/W
5	-	保留	该位总是读为 0，写入值应始终为 0。	R/W
6	-	保留	该位总是读为 0，写入值应始终为 0。	R/W
7	LTES	超时错误选择	0: 帧超时错误 1: 响应超时错误	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LEDEn寄存器。

#### BERE位（位错误检测允许位）

BERE位允许或禁止位错误的检测。

将此位设为1（位错误检测允许）。

位错误的检测结果在LESTn寄存器的BER标志中显示。

有关位错误的详情，请参考20.4.6错误状态。

#### PBERE位（物理总线错误检测允许位）

PBERE位允许或禁止对物理总线错误的检测。

设置为0时，不会检测物理总线错误。

设置为1时，检测物理总线错误。

当该位设置为1时，检测结果在LESTn寄存器的PBER标志中显示。

关于物理总线错误的详情，请参考20.4.6错误状态。

**FTERE位（超时错误检测允许位）**

FTERE位允许或禁止帧超时错误或响应超时错误的检测。

设置为0时，不会检测帧超时错误或响应超时错误。

设置为1时，检测帧超时错误或响应超时错误。

当该位设置为1时，检测结果在LESTn寄存器的FTER标志中显示。

通过LTES位，可以选择帧超时错误或响应超时错误。

如果要传输或接收9字节或更多的响应数据，请不要使用超时错误。

关于超时错误的详情，请参考20.4.6错误状态。

**FERE位（帧错误检测允许位）**

FERE位允许或禁止帧错误的检测。

将此位设为1（帧错误检测允许）。

帧错误检测结果在LESTn寄存器的FER标志中显示。

关于帧错误的详情，请参考20.4.6错误状态。

**LTES位（超时错误选择位）**

LTES位选择要使用的特定超时功能。

设置为0时，超时功能适用于帧超时。

设置为1时，超时功能适用于响应超时。

关于超时错误的详情，请参考20.4.6错误状态。

### 20.2.1.15 LIN/UART控制寄存器 (LCUCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LCUCn	-	-	-	-	-	-	OM1	OM0

位	符号	位名称	功能	R/W
0	OM0	LIN 复位	0: 开启 LIN 复位模式。 1: 取消 LIN 复位模式。	R/W
1	OM1	LIN 模式选择	0: 开启 LIN 唤醒模式。 1: 开启 LIN 工作模式。	R/W
7 至 2	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

将LCUCn寄存器设置为01H, 使其在取消LIN复位模式后转换到LIN唤醒模式。并将LCUCn寄存器设置为03H, 转换到LIN工作模式。

在LIN自测模式下, 将LCUCn寄存器在LIN自测模式转换完成后设置为03H。

在向该寄存器写入一个值后, 在写入另一个值之前, 应确认所写入的值确实在LMSTn寄存器中已显示。

#### OM0位 (LIN复位位)

OM0位可选择转换成LIN复位模式的或取消LIN复位模式。

设置为0时, 开启LIN复位模式。

设置为1时, 取消LIN复位模式。

#### OM1位 (LIN模式选择位)

OM1位在取消LIN复位模式后选择具体的操作模式 (LIN唤醒模式或LIN工作模式)。设置为0时, 开启LIN唤醒模式。

设置为1时, 开启LIN工作模式。

只有当LMSTn寄存器中的OMM0位为1时, 该寄存器才有效。当LTRCn寄存器中的FTS位为1时, 向该位写值是无效的。

### 20.2.1.16 LIN/UART传输控制寄存器 (LTRCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LTRCn	-	-	-	-	-	-	RTS	FTS

位	符号	位名称	功能	R/W
0	FTS	帧传输或唤醒传输/接收开始	0: 帧传输或唤醒传输/接收停止。 1: 帧传输或唤醒传输/接收开始。	R/W
1	RTS	响应传输/接收开始	0: 在帧分离模式下停止响应传输/接收。 1: 在帧分离模式下开始响应传输/接收。	R/W
2	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
7 至 3	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

#### FTS位 (帧传输或唤醒传输/接收起始位)

将FTS位设置为1, 开始帧或唤醒传输。

同时将该位设置为1, 允许唤醒接收 (对输入信号的低宽度进行计数)。

这个位只能写1, 不能写0。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 向该位写值是无效的。在完成帧或唤醒通信并转换到LIN复位模式时, 该位设置为0。

#### RTS位 (响应传输/接收起始位)

在帧分离模式下, 在帧头传输开始 (FTS位为1) 且响应传输数据准备好后, 将RTS位设置为1。一旦设置, 在完成帧通信或转换到LIN复位模式, 该位会自动清除为0。

这个位只能写1, 不能写0。

要向该位写1, 请使用8位数据传输指令向LTRCn寄存器写02H。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 向该位写值是无效的。

当FTS位为0时 (帧传输或唤醒传输/接收停止), 向该位写值是无效的。

当要传输或接收9字节或更多的响应数据时, 每次传输或接收一个数据组 (0到8字节) 时, 将该位设置为1。一旦设置, 在完成数据组通信或转换到LIN复位模式时, 该位会自动清除为0。

### 20.2.1.17 LIN/UART模式状态寄存器 (LMSTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LMSTn	-	-	-	-	-	-	OMM1	OMM0

位	符号	位名称	功能	R/W
0	OMM0	LIN 复位状态监控	0: 模块处于 LIN 复位模式。 1: 模块不处于 LIN 复位模式。	R
1	OMM1	LIN 模式状态监控	0: 模块处于 LIN 唤醒模式。 1: 该模块处于 LIN 工作模式。	R
7 至 2	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

OMM0位 (LIN复位状态监控)

OMM1位 (LIN模式状态监控)

OMM0和OMM1位表示当前的工作模式。



## 20.2.1.18 LIN/UART状态寄存器 (LSTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSTn	HTRC	D1RC	-	-	ERR	-	FRC	FTC

位	符号	位名称	功能	RW
0	FTC	成功的帧/唤醒传输标志	0: 帧或唤醒传输尚未完成。 1: 帧或唤醒传输已经完成。	RW
1	FRC	成功的帧/唤醒接收标志	0: 帧或唤醒接收尚未完成。 1: 帧或唤醒接收已经完成。	RW
2	-	保留	该位总是读为 0, 写入值应始终为 0。	RW
3	ERR	错误检测标志	0: 没有检测到错误。 1: 检测到错误。	R
4,5	-	保留	该位总是读为 0, 写入值应始终为 0。	RW
6	D1RC	成功的数据 1 接收标志	0: 数据 1 的接收尚未完成。 1: 数据 1 的接收已经完成。	RW
7	HTRC	成功的帧头传输标志	0: 帧头的传输尚未完成。 1: 帧头的传输已经完成。	RW

在转换到LIN复位模式和开始下一次通信时, LSTn寄存器会自动清零到00H (LTRCn寄存器中的FTS位为1)。在LIN复位模式下, 禁止对该寄存器的写入。在LIN复位模式下, 该寄存器保留为00H。

要清除寄存器中的特定位, 通过使用8位数据传输指令把要清除的位写成0, 把其他位写成1。

### FTC标志 (成功的帧/唤醒传输标志)

只有0可以写入FTC标志; 当1写入时, 该位保留了1写入前的数值。响应或唤醒传输完成后, FTC标志设置为1。如果LIEn寄存器中的FTCIE位为1 (允许中断), 就会产生一个中断。要在下一次通信前将该位清零 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式或LIN唤醒模式下向该位写0。

当要传输9字节或更多的响应数据时, 每次传输一个数据组 (0到8字节) 向该位设置为1。在下一个数据组开始传输之前写0。

### FRC标志 (成功的帧/唤醒接收标志)

只有0可以写入FRC标志; 当1写入时, 该位保留了1写入前的数值。响应或唤醒接收完成后, FRC标志设置为1。如果LIEn寄存器中的FRCIE位为1 (允许中断), 就会产生一个中断。要在下一次通信前将该位清零 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式或LIN唤醒模式下向该位写0。

当要接收9字节或更多的响应数据时, 每次接收数据组 (0到8字节) 时, 该位设置为1。在开始接收下一个数据组之前, 向该位写0。

### ERR标志 (错误检测标志)

ERR标志在检测到错误时设置为1 (LSTn寄存器的任何一个标志为1)。如果LIEn寄存器中的ERRIE位为1 (允许中断), 则产生一个中断。要在下一次通信前将该位清除为0 (LTRCn寄存器中的FTS位为1), 在LIN工作模式或LIN唤醒模式下, 向LSTn寄存器中的RPER、CSER、FER、TER、PBER和BER标志写0, 则ERR标志清除为0。

**D1RC标志（成功的数据1接收标志）**

只有0可以写入D1RC标志；当1写入时，该位保留了1写入前的数值。在完成数据1的接收后，D1RC标志设置为1。这里，不产生中断。要在下一次通信前将该位清除为0（LTRCn寄存器中的FTS位为1），请在LIN工作模式下将该位写入0。

当要接收9字节或更多的响应数据时，每次接收数据组（0到8字节）的数据1，该位设置为1。在开始接收下一个数据组之前写0。

**HTRC标志（成功的帧头传输标志）**

只有0可以写入HTRC标志；当1写入时，该位保留了1写入前的值。HTRC标志在帧头传输完成后设置为1。如果LIEn寄存器中的SHIE位为1（允许中断），就会产生一个中断。要在下一次通信前将该位清除为0（LTRCn寄存器中的FTS位为1），需在LIN工作模式下将该位写入0。

## 20.2.1.19 LIN/UART错误状态寄存器 (LESTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LESTn	RPER	-	CSER	-	FER	FTER	PBER	BER

位	符号	位名称	功能	RW
0	BER	位错误标志	0: 未检测到位错误。 1: 已检测到位错误。	RW
1	PBER	物理总线错误标志	0: 未检测到物理总线错误。 1: 已检测到物理总线错误。	RW
2	FTER	超时错误标志	0: 未检测到帧/响应超时错误。 1: 已检测到帧/响应超时错误。	RW
3	FER	帧错误标志	0: 未检测到帧错误。 1: 已检测到帧错误。	RW
4	-	保留	该位总是读为0, 写入值应始终为0。	RW
5	CSEr	校验和错误标志	0: 未检测到校验和错误。 1: 已检测到校验和错误。	RW
6	-	保留	该位总是读为0, 写入值应始终为0。	RW
7	RPER	响应准备错误标志	0: 未检测到响应准备错误。 1: 已检测到响应准备错误。	RW

在切换到LIN复位模式和开始下一次通信时, LESTn寄存器会自动清零到00H (LTRCn寄存器中的FTS位为1)。在LIN复位模式下, 禁止对该寄存器的写入。在LIN复位模式下, 该寄存器保留00H。

当LTRCn寄存器中的FTS位为1时 (帧传输或唤醒传输/接收开始), 不要向该寄存器写值。

要清除寄存器中特定的位, 通过使用8位数据传输指令, 把要清除的位写成0, 把其他位写成1。

### BER标志 (位错误标志)

只有0可以写入BER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的BERE位为1 (允许位错误检测), BER标志在位错误检测时设置为1。要在下一次通信前将该位清除为0 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式或LIN唤醒模式下向该位写入0。

### PBER标志 (物理总线错误标志)

只有0可以写入PBER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的PBERE位为1 (允许物理总线错误检测), PBER标志在物理总线错误检测时设置为1。要在下一次通信前将该位清除为0 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式或LIN唤醒模式下向该位写入0。

### FTER标志 (超时错误标志)

只有0可以写入FTER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的FTERE位为1 (允许帧/响应超时错误检测), 帧超时错误或响应超时错误检测时FTER标志设置为1。要在下一次通信前将该位清零 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式下将该位写入0。

### FER标志 (帧错误标志)

只有0可以写入FER标志; 当1写入时, 该位保留1写入前的值。如果LEDEn寄存器中的FERE位为1 (允许帧错误检测), 帧错误检测时FER标志设置为1。要在下一次通信前将该位清除为0 (LTRCn寄存器中的FTS位为1), 请在LIN工作模式下将该位写入0。

**CSER标志（校验和错误标志）**

只有0可以写入CSER标志；当1写入时，该位保留了1写入前的值。CSER标志在校验和错误检测时设置为1。要在下一次通信前将该位清除为0（LTRCn寄存器中的FTS位为1），请在LIN工作模式下向该位写入0。

**RPER标志（响应准备错误标志）**

只有0可以写入RPER标志；当1写入时，该位保留在1写入之前的值。响应准备错误检测时，RPER标志设置为1。要在下一次通信前将该位清除为0（LTRCn寄存器中的FTS位为1），请在LIN工作模式下向该位写入0。

Draft Version  
Cmsemicon Confidential

### 20.2.1.20 LIN/UART数据场配置寄存器(LDFCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LDFCn	LSS	FSM	CSM	RFT	RFDL[3:0]			

位	符号	位名称	功能	RW
3至0	RFDL[3:0]	响应场长度选择	b3 b0 0000: 0 个字节 (+校验和) 0001: 1 个字节 (+校验和) 0010: 2 个字节 (+校验和) : 0111: 7 个字节 (+校验和) 1000: 8 个字节 (+校验和) 禁止使用上述以外的设置。	RW
4	RFT	响应场通信方向选择	0:接收 1:传输	RW
5	CSM	校验和选择	0:经典校验和模式 1:增强校验和模式	RW
6	FSM	帧分离模式选择	0:帧分离模式未设置。 1:帧分离模式已设置。	RW
7	LSS	传输/接收继续选择	0: 接下来要传输/接收的数据组是最后一个。 1: 接下来要传输/接收的数据组不是最后一个。 (不包括校验和)。	RW

**RFDL[3:0]位（响应场长度选择位）** RFDL位设置响应场数据的长度。

数据长度可以是0到8个字节，不包括校验和大小。

要在FSM位设置为0的情况下传输响应数据（不是帧分离模式），在帧头传输前设置RFDL位（LTRCn寄存器中的FTS位为0）。要在FSM位设置为1（帧分离模式）的情况下传输响应数据，在响应传输前设置RFDL位（LTRCn寄存器中的RTS位为0）。

如要接收响应数据，请在帧头传输前设置RFDL位（LTRCn寄存器中的FTS位为0）。

当要传输或接收9字节或更多的响应数据时，在数据组传输/接收前设置RFDL位（LTRCn寄存器的RTS位为0）。

在9字节或更多的响应数据通信期间，只有最后一个数据组（LSS位为0）包括校验和，其他组（LSS位为1）不包括校验和。

**RFT位（响应场通信方向选择位）**

RFT位设置响应场/唤醒信号的通信方向。

设置为0时，在响应场进行接收。在LIN唤醒模式下，执行唤醒接收（计算输入信号的低电平宽度）。

设置为1时，在响应场进行传输。在LIN唤醒模式下，执行唤醒传输。当LTRCn寄存器中的FTS位为0时，设置该位（帧传输或唤醒传输/接收停止）。

当要传输或接收9个字节或更多的响应数据时，在第一个数据组到最后一个数据组之后不要改变RFT位的设置。

**CSM位（校验和选择位）** CSM位设置校验和模式。

设置为0时，选择经典型校验和模式。

设置为1时，则选择增强型校验模式。

当使用超时错误时（LEDEn寄存器中的FTERE位为1），具体超时时间取决于该位的设置。有关该位错误的详情，请参考20.4.6错误状态。

当LTRCn寄存器中的FTS位为0（帧传输或唤醒传输/接收停止）时，设置该位。

当要传输或接收9字节或更多的响应数据时，在第一个数据组到最后一个数据组之后不要改变CSM位的设置。

在9字节或更多的响应数据通信期间，只有最后一个数据组（LSS位为0）包括校验和，其他组（LSS位为1）不包括校验和。

**FSM位（帧分离模式选择位）** FSM位设置响应通信模式。

设置为0时，不选择帧分离模式。在这种情况下，帧头传输开始后（LTRCn寄存器中的FTS位为1），响应已经传输/接收，且不设置LTRCn寄存器中的RTS位。

设为1时，选择帧分离模式。如果在帧头传输过程中LTRCn寄存器中的RTS位设置为1，则在成功的帧头传输后响应已经传输。

对于响应接收（RFT位为0），FSM位设为0。

当转换到LIN自测模式时，在转换前将该位设置为0。

关于帧分离模式的详情，请参考20.4.3(1)(a)帧分离模式。

当LTRCn寄存器中的FTS位为0（帧传输或唤醒传输/接收停止）时，设置该位。

当要传输或接收9字节或更多的响应数据时，将FSM位设为1。

**LSS位（传送/接收持续选择位）**

LSS位显示，当要传输或接收9字节或更多的响应数据时，下一个要传输或接收的数据组不是最后一组。设置为0时，数据和校验和将传送或接收，因为下一个要传送或接收的数据组是最后一组。

设置为1时，只传输或接收数据，不包括校验和，因为要传输或接收的下一个数据组不是最后一组。只有当FSM位为1（帧分离模式）并且要传输或接收9字节或更多的响应数据时，才设置LSS位。

当LTRCn寄存器中的RTS位为0（响应传输/接收停止）时，设置该位。

### 20.2.1.21 LIN/UARTID缓冲寄存器 (LIDBn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LIDBn	IDP[1:0]		ID[5:0]					

位	符号	位名称	功能	R/W
5至0	ID[5:0]	ID 设置	设置要在 ID 场中传输的 6 位 ID 值。	R/W
7,6	IDP[1:0]	奇偶校验设置	设置要在 ID 场中传输的奇偶校验位 (P)。	R/W

当LTRCn寄存器中的FTS位为0时，设置LIDBn寄存器（帧传输或唤醒传输/接收停止）。

在LIN自测模式下，该寄存器的操作如下：

在通信前写入要传输的值。在帧传输/接收完成后，可以从寄存器中读取接收值的反转值（回环后）。关于LIN自测模式的详情，请参考20.6LIN自测模式。

#### ID位 (ID设置位)

ID位设置6位在LIN帧的ID场中传输的ID值。

#### IDP位 (奇偶校验设置位)

IDP位设定了将在LIN帧的ID场中传输的奇偶校验位 (P0和P1)。

IDP0位是P0，IDP1位是P1。

由于奇偶校验不是自动计算的，所以要设置计算结果。注意，如果设置了错误的结果，就会按原样传输。

### 20.2.1.22 LIN校验和缓冲寄存器 (LCBRn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LCBRn								

位	功能	R/W
7至0	保存传输或接收的校验和值。	R/W

在LIN模式下，该寄存器的操作如下：

- 当LDFCn寄存器中的RFT位为1（传输）：
  - 可以从寄存器中读取传输的数值。在传输完成后读取该值。
  - 对该寄存器的写入是无效的。
- 当LDFCn寄存器中的RFT位为0（接收）：
  - 收到的值可以从寄存器中读取。在接收完成后读取该值。
  - 对该寄存器的写入是无效的。

在LIN自测模式下，该寄存器的操作如下：

- 当LDFCn寄存器中的RFT位为1（传输）：
  - 在帧传输完成后，可以从寄存器中读取传输值的反转值（回环后）。
- 当LDFCn寄存器中的RFT位为0时（接收）：
  - 写入通信前要接收的值。
  - 在帧传输/接收完成后，可以从寄存器中读取接收值的反转值（回环后）。

关于LIN自测模式的详情，请参考20.6LIN自测模式。

当LTRCn寄存器中的FTS位为0时（帧传输或唤醒传输/接收停止），设置LCBRn寄存器。

当要传输或接收9字节或更多的响应数据时，校验和只附加到最后一个数据组；该寄存器不对其他数据组进行更新。



### 20.2.1.23 LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LCBRnm								

位	功能	设置范围	R/W
7至0	设置要传输的数据, 或允许读取收到的数据。	00H至FFH	R/W

用于响应传输:

寄存器设置要在响应场中传输的数据。

使用寄存器的设置如下。

- LDFCn寄存器中的RFT为1 (传输)
  - LDFCn寄存器中的FSM为0 (非帧分离模式)
  - LTRCn寄存器中的FTS位为0 (帧传输或唤醒传输/接收停止)
- 或
- LDFCn寄存器中的RFT为1 (传输)
  - LDFCn寄存器中的FSM为1 (帧分离模式)
  - LTRCn寄存器中的RTS为0 (响应传输/接收停止)

用于响应接收:

寄存器保存在响应场中收到的数据。

收到的数据被覆盖。如果检测到错误, 接收中断前的数据会保存在寄存器中。

当FTS位为1 (帧传输或唤醒传输/接收开始) 时, 不要读这些寄存器。

用于传输9字节或以上的响应数据: 使用寄存器的设置如下。

- LDFCn寄存器中的RFT为1 (传输)
- LDFCn寄存器中的FSM为1 (帧分离模式)
- LTRCn寄存器中的RTS为0 (响应传输/接收停止)

用于接收9字节或以上的响应数据:

当RTS位为1时 (响应传输/接收开始), 不要读这些寄存器。

在LIN自测模式下, 这些寄存器的操作如下:

写入通信前要传输的数值。

在帧传输/接收完成后, 可以从寄存器中读取接收值的反转值 (回环后)。

关于LIN自测模式的详情, 请参考20.6LIN自测模式。

## 20.2.2 从机模式的LIN寄存器

### 20.2.2.1 输入切换控制寄存器 (ISC)

ISC寄存器中的ISC2位用于LIN/UART模块 (LIN)。

位2设置为1，即选择LIN/UART模块的串行数据输入引脚的输入信号作为外部中断输入。

这个寄存器可以由8位的内存操作指令来设置。

复位信号的产生将该寄存器设置为00H。

40020473H      复位值 00H    R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	ISC2	ISC1	ISC0

ISC2	外部中断 INTP11 的输入切换
0	将 INTP11 引脚的输入信号用作外部中断的输入。
1	将 LRXD0 引脚的输入信号用作外部中断的输入。

**注意事项** 第7至3位应始终设置为0。

### 20.2.2.2 外围寄存器2 (PER2)

PER2寄存器用于允许或禁止对外围硬件的时钟供应。停止向未使用的硬件提供时钟，以降低功耗和噪声。要使用此寄存器控制的外围功能，在指定外围功能的初始设置之前，请设置（1）与每个功能相对应的位。

这个寄存器可以由8位的内存操作指令来设置。

复位信号产生将此寄存器设置为00H。

4002081BH      复位值 00H    R/W

符号	7	6	5	4	3	2	1	0
PER2	0	0	0	0	0	0	SPIE N	LINE N

LINEN	LIN模块的输入时钟控制
0	停止提供输入时钟。 •不能写LIN模块使用的SFR。 •LIN模块处于复位状态。
1	提供输入时钟。 •能读写LIN模块使用的SFR。

### 20.2.2.3 LIN时钟选择寄存器 (LINCKSEL)

该寄存器用于控制提供给LIN的通信时钟源。

4002081CH      复位值 00H    R/W

符号	7	6	5	4	3	2	1	0
LINCKSEL	0	0	0	LINMCKE	0	0	0	LINMCK

LINMCKE	供应或停止 LIN 通信时钟源的控制
0	停止 LIN 通信时钟源供应。
1	允许 LIN 通信时钟源供应。

LINMCK	选择 LIN 通信时钟源的控制
0	选择 fCLK 的 2 分频时钟。
1	选择 fMX 时钟。

注意事项1.在将LINnMCKE (n=0) 位设置为1 (提供工作时钟) 之前, 用LINnMCK位选择LINn工作时钟。

2.当在SNOOZE模式下操作LINn时, 将LINnMCK位设为0。

3.如果LINnMCK设置为1, 则不使用超时错误检测。

在这种情况下, 将至少1.2倍的LIN通信时钟源的频率设置成fCLK时钟。

### 20.2.2.4 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)

更多细节，详见24.3.3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)。

### 20.2.2.5 LIN唤醒波特率选择寄存器 (LWBRn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LWBRn	NSPB[3:0]			LPRS[2:0]			LWBR 0	

位	符号	位名称	功能	RW
0	-	保留	该位总是读为0，写入值应始终为0。	RW
3至1	LPRS [2:0]	预分频器时钟选择	b3 b1 000:1/1 001:1/2 010:1/4 011:1/8 100:1/16 101:1/32 110:1/64 111:1/128	RW
7至4	NSPB [3:0]	位采样计数选择	b7 b4 0000:16 采样 0011:4 采样 0111:8 采样 1111:16 采样 禁止使用上述以外的设置。	RW

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时，设置LWBRn寄存器。

#### LPRS[2:0]位 (预分频器时钟选择位)

LPRS位选择预分频器的分频比。

LIN通信时钟源的频率是根据这个预分频器来划分的。

在自动波特率的LIN从机模式下 (LIN/UART模式寄存器中的LIN/UART模式选择位=10b)，根据目标波特率设置位，使预分频器时钟的频率达到列表中的相应值。

[目标波特率][预分频器时钟频率]

1kbps到20kbps: 4MHz<sup>注</sup>

1kbps到小于2.4kbps: 4MHz

2.4kbps至20kbps: 8MHz至12MHz

**注:**将NSPB位设置为0011b (4采样)。

#### NSPB[3:0]位(位采样计数选择位)

NSPB位从单个Tbit中选择采样次数 (波特率的倒数)。

在自动波特率的LIN从机模式下 (LIN/UART模式寄存器中的LIN/UART模式选择位=10b)，NSPB位设置为0011b (4采样) 或0111b (8采样)。

在固定波特率的LIN从机模式下 (LIN/UART模式寄存器中的LIN/UART模式选择位=11b)，NSPB位设置为0000b或1111b (16采样)。

### 20.2.2.6 LIN/UART波特率预分频器寄存器 (LBRPn)

符号	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
LBR Pn	LBRPn1								LBRPn0							

位	功能	设置范围	R/W
15至0	假设该寄存器中设置的值为L (0~65535)，波特率预分频器将预分频器时钟的频率除以L+1。	0000H至FFFFH	R/W

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时，设置LBRPn寄存器。

假设该寄存器中设置的值为L，波特率预分频器将LWBRn寄存器中LPRS位 (预分频器时钟选择位) 选择的时钟频率除以L+1。

LBRPn寄存器可以通过以下寄存器以8位访问。

- 低8位：LIN/UART波特率预分频器0寄存器 (LBRPn0) ；
- 高8位：LIN/UART波特率预分频器1寄存器 (LBRPn1) ；

**注意事项** 在LIN从机模式[自动波特率]下，当同步场接收成功后，波特率校正结果会自动设置到LBRPn寄存器中。

### 20.2.2.7 LIN自测控制寄存器 (LSTCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSTCn								LSTM

位	符号	位名称	功能	RW
7至0			将 A7H、58H 和 01H 连续写入该位，使模块进入 LIN 自测模式。	RW
0	LSTM	LIN 自测模式	0: 模块不在 LIN 自测模式中 1: 模块在 LIN 自测模式中。	RW

LSTCn寄存器取消了对LIN自测模式的保护。

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LSTCn寄存器。

连续向LSTCn寄存器写入A7H、58H和01H，使模块进入LIN自测模式。

当连续写入完成后进入LIN自测模式，LSTM位设置为1。

在连续写的过程中，不要写任何其他值。

关于转换到LIN自测模式，请参考20.6**LIN自测模式**。

读取第6至1位返回000000b，读取第7位返回未定义值。

#### LSTM位（LIN自测模式位）

当LIN自测模式转换完成后，LSTM位设置为1。

关于退出LIN自测模式，请参考20.6**LIN自测模式**。

如果不是连续写入A7H、58H和01H，向该位写1不会影响LSTCn寄存器的值。

### 20.2.2.8 LIN/UART模式寄存器 (LMDn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LMDn	-	-	LRDN FS	LIOS	-	-	LMD[1:0]	

位	符号	位名称	功能	RW
1,0	LMD[1:0]	LIN/UART 模式选择	b1b0 10: LIN 从机模式 (自动波特率)。 11: LIN 从机模式 (固定波特率)。	RW
3,2	-	保留	该位总是读为 0, 写入值应始终为 0。	RW
4	LIOS	LIN 中断输出选择	0: 使用 LIN 中断。 1: 使用传输中断、成功的接收中断和接收状态中断。	RW
5	LRDNFS	LIN 接收数据噪声过滤禁止	0: 噪声滤波器允许。 1: 噪声滤波器禁止。	RW
6	-	保留	该位总是读为 0, 写入值应始终为 0。	RW
7	-	保留	该位总是读为 0, 写入值应始终为 0。	RW

当LMSTn寄存器中的OMM0位为0时 (LIN复位模式), 设置LMDn寄存器。

**LMD[1:0]位 (LIN/UART模式选择位)** LMD位选择LIN/UART模块模式。

如将LIN/UART模块作为LIN从机使用, 需将该位设置为10b或11b。

设置为10b时, LIN/UART模块以自动波特率的LIN从机模式运行。

设置为11b时, LIN/UART模块在固定波特率的LIN从机模式运行。

**LIOS位 (LIN中断输出选择位)**

LIOS位选择LIN/UART模块中断输出的数量。

设置为0时, LIN中断由LIN/UART模块产生。

设置为1时, LIN/UART模块产生传输中断、成功接收中断和接收状态中断。

对于每一个中断源, 请参考20.9 中断。

**LRDNFS位 (LIN接收数据噪声过滤禁止位)**

LRDNFS位在接收数据时允许或禁止噪声滤波器。

设置为0时, 接收数据时允许噪声滤波器。

设置为1时, 接收数据时禁止噪声滤波器。



### 20.2.2.9 LIN间隔场配置寄存器/UART配置寄存器 (LBFCn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LBFCn	-	-	-	-	-	-	-	BLT

位	符号	位名称	功能	R/W
0	BLT	接收间隔(低)宽度选择	0: 检测到 9.5/10 或更多 Tbits 的接收间隔(低宽度)。 1: 检测到 10.5/11 或更多 Tbits 的接收间隔(低宽度)。	R/W
6至1	-	保留	该位总是读为0, 写入值应始终为0。	R/W
7	-	保留	该位总是读为0, 写入值应始终为0。	R/W

当LMSTn寄存器中的OMM0位为0时(LIN复位模式), 设置LBFCn寄存器。

#### BLT位(接收间隔(低)宽度选择位)

BLT位设置接收数据的低电平宽度临界点, 以确定为间隔断点。

在自动波特率的LIN从机模式下(LMDn寄存器中的LMD位为10b):

设置为0时, 检测到10或更多Tbits的低宽度。

设置为1时, 检测到11或更多Tbits的低宽度。

在固定波特率的LIN从机模式下(LMDn寄存器中的LMD位为11b)。

设置为0时, 检测到9.5或更多Tbits的低宽度。

设置为1时, 检测到10.5或更多Tbits的低宽度。

### 20.2.2.10 LIN/UART空间配置寄存器 (LSCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSCn	-	-	IBS[1:0]		-	RS[2:0]		

位	符号	位名称	功能	R/W
2至0	RS[2:0]	响应间隔选择	b2 b0 000:0 Tbit 001:1 Tbit 010:2 Tbits 011:3 Tbits 100:4 Tbits 101:5 Tbits 110:6 Tbits 111:7 Tbits	R/W
3	-	保留	该位总是读为 0，写入值应始终为 0。	R/W
5,4	IBS[1:0]	字节间隔选择	b5b4 00:0 Tbit 01:1 Tbit 10:2 Tbits 11:3 Tbits	R/W
7,6	-	保留	该位总是读为 0，写入值应始终为 0。	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LSCn寄存器。设置仅在响应传输期间允许；设置在响应接收期间禁止。

某些设定值的组合会导致帧的长度或响应超过超时时间，所以请在这个寄存器中设置合适的值。

#### RS[2:0]位（响应间隔选择位）

RS位设置响应传输的响应间隔宽度。

可以设置0 Tbit到7 Tbit。

#### IBS[1:0]位（字节间隔选择位）

IBS位设置响应传输的字节间隔宽度。

可以设置0 Tbit到3 Tbit。

### 20.2.2.11 LIN唤醒配置寄存器 (LWUPn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LWUPn	WUTL[3:0]				-	-	-	-

位	符号	位名称	功能	R/W
3至0	-	保留	该位总是读为0，写入值应始终为0。	R/W
7至4	WUTL[3:0]	唤醒传输低宽度选择	b7 b4 0000:1 Tbit 0001:2 Tbits 0010:3 Tbits 0011:4 Tbits : 1100:13 Tbits 1101:14 Tbits 1110:15 Tbits 1111:16 Tbits	R/W

当LMSTn寄存器中的OMM0位为0（LIN复位模式）时，设置LWUPn寄存器。

#### WUTL[3:0]位（唤醒传输低宽度选择位）

WUTL位设置唤醒帧传输的低宽度。

可以设置1Tbit到16Tbit。

## 20.2.2.12 LIN中断允许寄存器 (LIEn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LIEn	-	-	-	-	SHIE	ERRIE	FRCIE	FTCIE

位	符号	位名称	功能	R/W
0	FTCIE	成功的帧/唤醒传输中断允许	0:禁止成功响应/唤醒传输中断。 1:允许成功响应/唤醒传输中断。	R/W
1	FRCIE	成功的帧/唤醒接收中断允许	0:禁止成功响应/唤醒接收中断。 1:允许成功响应/唤醒接收中断。	R/W
2	ERRIE	错误检测中断允许	0:禁止错误检测中断。 1:允许错误检测中断。	R/W
3	SHIE	成功的帧头接收中断允许	0:禁止成功帧头接收中断。 1:允许成功接收帧头的中断。	R/W
7至4	-	保留	该位总是读为0，写入值应始终为0。	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LIEn寄存器。

### FTCIE位（成功的帧/唤醒传输中断允许位）

FTCIE位在成功的传输响应或唤醒信号时允许或禁止中断生成。

设置为0时，当LSTn寄存器中的FTC标志设置为1时，不产生中断。

设置为1时，当LSTn寄存器中的FTC标志设置为1时，产生中断。

### FRCIE位（成功的帧/唤醒接收中断允许位）

FRCIE位在成功接收到响应或唤醒信号（计算输入信号的低宽度）时，允许或禁止中断产生。设置为0时，当LSTn寄存器中的FRC标志设置为1时，不产生中断。

设置为1时，当LSTn寄存器中的FRC标志设置为1时，产生中断。

### ERRIE位（错误检测中断允许位）

ERRIE位在检测到错误时允许或禁止中断生成。

设置为0时，当LSTn寄存器中的ERR标志设置为1时，不产生中断。

设置为1时，当LSTn寄存器中的ERR标志设置为1时，将产生中断。

中断源可以是位错误、帧/响应超时错误、帧错误、同步场错误、校验和错误、ID奇偶校验错误和响应准备错误。位错误、帧/响应超时错误、帧错误、同步场错误和ID奇偶校验错误的检测可以通过LEDEn寄存器允许或禁止。

### SHIE位（成功的接收帧头中断允许位）

SHIE位在成功传输头时允许或禁止中断生成。

设置为0时，当LSTn寄存器中的HTRC标志设置为1时，不产生中断。

设置为1时，当LSTn寄存器中的HTRC标志设置为1时，产生中断。

### 20.2.2.13 LIN/UART错误检测寄存器 (LEDEn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LEDEn	LTES	IPERE	-	SFERE	FERE	TERE	-	BERE

位	符号	位名称	功能	R/W
0	BERE	位错误检测允许	0:禁止位错误检测。 1:允许位错误检测。	R/W
1	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
2	TERE	超时错误检测允许	0:禁止帧/响应超时错误检测。 1:允许帧/响应超时错误检测。	R/W
3	FERE	帧错误检测允许	0:禁止帧错误检测。 1:允许帧错误检测。	R/W
4	SFERE	同步场错误检测允许	0:禁止同步场错误检测。 1:允许同步场错误检测。	R/W
5	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
6	IPERE	ID 奇偶校验错误检测允许	0:禁止 ID 奇偶校验错误检测。 1:允许 ID 奇偶校验错误检测。	R/W
7	LTES	超时错误选择	0:帧超时错误 1:响应超时错误	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LEDEn寄存器。

#### BERE位（位错误检测允许位）

BERE位允许或禁止位错误的检测。将此位设为1（位错误检测允许）。

位错误的检测结果在LESTn寄存器的BER标志中显示。有关位错误的详情，请参考20.4.6**错误状态**。

#### TERE位（超时错误检测允许位）

TERE位允许或禁止对帧超时错误或响应超时错误的检测。

设置为0时，不检测帧超时错误或响应超时错误。

设置为1时，检测到帧超时错误或响应超时错误。

当该位设置为1时，检测结果在LESTn寄存器的TER标志中显示。

通过LTES位，可以选择帧超时错误或响应超时错误。

在自动波特率的LIN从机模式下，不要使用超时错误（LIN/UART模式寄存器中的LIN/UART模式选择位=10b）。

如果要传输或接收9字节或更多的响应数据，请不要使用超时错误。关于超时错误的详情，请参考20.4.6**错误状态**。

#### FERE位(帧错误检测允许位)

FERE位允许或禁止帧错误的检测。

将该位设为1（帧错误检测允许）。

帧错误检测结果在LESTn寄存器的FER标志中显示。

关于帧错误的详细信息，请参考20.4.6**错误状态**。

**SFERE位（同步场错误检测允许位）**

SFERE位允许或禁止对同步场错误的检测。

设置为0时，不检测同步场错误。

设置为1时，检测到同步场错误。

一旦检测到同步场错误，系统就会进入下一个帧头的等待状态，与该位的设置无关。

当该位设置为1时，检测结果在LESTn寄存器的SFER标志中显示。

关于同步场错误的详情，请参考20.4.6**错误状态**。

**IPEERE位（ID奇偶校验错误检测允许位）**

IPEERE位允许或禁止对ID奇偶校验错误的检测。

设置为0时，不检测ID奇偶校验错误。

设置为1时，检测ID奇偶校验错误。

当该位设置为1时，检测结果在LESTn寄存器的IPER标志中显示。

关于ID奇偶校验错误的详情，请参考20.4.6**错误状态**。

**LTES位（超时错误选择位）**

LTES位选择要使用的特定超时功能。

设置为0时，超时功能适用于帧超时。

设置为1时，超时功能适用于响应超时。

关于超时错误的详情，请参考20.4.6**错误状态**。

## 20.2.2.14 LIN/UART控制寄存器 (LCUCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LCUCn	-	-	-	-	-	-	OM1	OM0

位	符号	位名称	功能	R/W
0	OM0	LIN 复位	0: 开启 LIN 复位模式。 1: 取消 LIN 复位模式。	R/W
1	OM1	LIN 模式选择	0: 开启 LIN 唤醒模式。 1: 开启 LIN 工作模式。	R/W
7 至 2	-	保留	该位总是读为 0，写入值应始终为 0。	R/W

将LCUCn寄存器设置为01H，即取消LIN复位模式后转换到LIN唤醒模式。将LCUCn寄存器设置为03H，即转换到LIN工作模式。

在LIN自测模式下，完成LIN自测模式的转换后，将LCUCn寄存器设置为03H。

当LIN/UART模块在作为LIN从机工作时（以固定的波特率）从LIN工作模式转换到LIN复位模式时，将给定的位清除为0后，向PER2寄存器中的LIN0EN位（或LIN1EN位）写1。

向该寄存器写入一个值后，在写入另一个值之前，应确认所写入的值确实在LMSTn寄存器中已显示。

### OM0位 (LIN复位位)

OM0位可以选择向LIN复位模式的转换或取消LIN复位模式。

设置为0时，开启LIN复位模式。

设置为1时，取消LIN复位模式。

### OM1位 (LIN模式选择位)

OM1位在取消LIN复位模式后选择具体的工作模式（LIN唤醒模式或LIN工作模式）。设置为0时，开启LIN唤醒模式。

设置为1时，开启LIN工作模式。

只有当LMSTn寄存器中的OMM0位为1时，该寄存器才有效。LTRCn寄存器中的FTS位为1时，向该位写值是无效的。

## 20.2.2.15 LIN/UART传输控制寄存器 (LTRCn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LTRCn	-	-	-	-	-	LNRR	RTS	FTS

位	符号	位名称	功能	R/W
0	FTS	LIN 通信启动	0: 帧头接收或唤醒传输/接收停止。 1: 帧头接收或唤醒传输/接收开始。	R/W
1	RTS	响应传输/接收开始	0: 响应传输/接收停止。 1: 响应传输/接收开始。	R/W
2	LNRR	无响应请求	0: 接收/发送了对接收 ID 的响应。 1: 未接收/未发送对接收 ID 的响应。	R/W
7 至 3	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

### FTS位 (LIN通信起始位)

将FTS位设置为1, 开始接收帧头或唤醒信号 (对输入信号的低宽度进行计数)。

同时将该位设置为1, 允许唤醒传输。

这个位只能写1, 不能写0。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 向该位写值是无效的。

在完成唤醒传输/接收和转换到LIN复位模式时, 该位设置为0。

### RTS位 (响应传输/接收起始位)

在接收到帧头信息并检查了收到的ID后开始响应传输/接收时, 将RTS位设置为1。

一旦设置, 在完成响应通信或转换到LIN复位模式时, 该位会自动清空为0。

这个位只能写1, 不能写0。

要将该位写入1, 请使用8位数据传输指令将02H写入LTRCn寄存器。

不要同时将该位和LNRR位设置为1。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 向该位写值是无效的。当FTS位为0时, 向该位写值是无效的 (帧头接收或唤醒传输/接收停止)。

当要传输或接收9字节或更多的响应数据时, 在每次传输或接收数据组 (从0到8字节) 时将该位设置为1。

一旦设置, 在完成数据组传输/接收或转换到LIN复位模式时, 该位将自动清除为0。

### LNRR位 (无响应请求位)

在收到帧头且检查收到的ID后, 既不开始响应传输, 也不开始接收, 将LNRR位设为1。一旦设置, 在检测到新的同步场或转换到LIN复位模式时, 该位会自动清除为0。

这个位只能写1, 不能写0。

要将该位写入1, 请使用8位数据传输指令将04H写入LTRCn寄存器。

不要同时将该位和RTS位设置为1。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 向该位写值是无效的。

当FTS位为0时, 向该位写值是无效的 (帧头接收或唤醒传输/接收停止)。

当要传输或接收一个9字节或更长的响应时, 仅在完成帧头的接收时使用该位 (在第二次和后续数据组接收完成时不要使用该位)。



### 20.2.2.16 LIN/UART模式状态寄存器 (LMSTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LMSTn	-	-	-	-	-	-	OMM1	OMM0

位	符号	位名称	功能	RW
0	OMM0	LIN 复位状态监控	0: 模块处于 LIN 复位模式。 1: 模块不处于 LIN 复位模式。	R
1	OMM1	LIN 模式状态监控	0: 模块处于 LIN 唤醒模式。 1: 模块处于 LIN 工作模式。	R
7 至 2	-	保留	该位总是读为 0, 写入值应始终为 0。	RW

**OMM0位 (LIN复位状态监控)**

**OMM1位 (LIN模式状态监控)**

OMM0和OMM1位指当前的操作模式。

## 20.2.2.17 LIN/UART状态寄存器 (LSTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LSTn	HTR C	D1R C	-	-	ERR	-	FRC	FTC

位	符号	位名称	功能	R/W
0	FTC	成功的帧/唤醒传输标志	0: 响应或唤醒传输尚未完成。 1: 响应或唤醒传输已经完成。	R/W
1	FRC	成功的帧/唤醒接收标志	0: 响应或唤醒接收尚未完成。 1: 响应或唤醒接收已经完成。	R/W
2	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
3	ERR	错误检测标志	0: 没有检测到错误。 1: 检测到错误。	R
4,5	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
6	D1RC	成功的数据 1 接收标志	0: 数据 1 的接收尚未完成。 1: 数据 1 的接收已经完成。	R/W
7	HTRC	成功的帧头接收标志	0: 帧头接收尚未完成。 1: 帧头接收已经完成。	R/W

在转换到LIN复位模式时, LSTn寄存器会自动清零到00H。

在LIN复位模式下, 禁止对该寄存器的写入。在LIN复位模式下, 该寄存器保留为00H。

要清除寄存器中的特定位, 通过使用8位数据传输指令, 把要清除的位写成0, 把其他位写成1。

### FTC标志 (成功帧/唤醒传输标志)

只有0可以写入FTC标志; 当1写入时, 该位保留了1写入前的数值。响应或唤醒传输完成后, FTC标志设置为1。如果LIEn寄存器中的FTCIE位为1 (允许中断), 就会产生一个中断。请注意, 当响应或唤醒传输完成时, FTC标志设置为1, 不会产生中断。如要将该位清除为0, 请向该位写0。

当要传输9字节或更多的响应数据时, 每次传输一个数据组 (从0到8字节) 时, 该位设置为1。在开始传输下一个数据组之前写0。

### FRC标志 (成功帧/唤醒接收标志)

只有0可以写入FRC标志; 当1写入时, 该位保留了1写入前的数值。响应或唤醒接收完成后, FRC标志设置为1。如果LIEn寄存器中的FRCIE位为1 (允许中断), 就会产生一个中断。请注意, 当响应或唤醒接收完成时, FRC标志设置为1, 不会产生中断。如要将该位清除为0, 请向该位写0。

当要接收9字节或更多的响应数据时, 每次接收一个数据组 (从0到8字节) 时, 该位设置为1。在开始接收下一个数据组之前写0。

### ERR标志 (错误检测标志)

在检测到错误时, ERR标志设置为1 (LSTn寄存器的任何标志为1)。如果LIEn寄存器中的ERRIE位为1 (允许中断), 就会产生一个中断。注意, 当检测到错误时, ERR标志设置为1, 则不产生中断。如要将该位清除为0, 请向LSTn寄存器中的RPER、IPER、CSER、SFER、FER、TER和BER标志写0。这样就把ERR标志清除到0。

**D1RC标志（成功的数据1接收标志）**

只有0可以写入D1RC标志；当1写入时，该位保留了1写入前的数值。在完成数据1的接收后，D1RC标志设置为1。这里，不会产生中断。如要将该位清除为0，请向该位写入0。

当要接收9字节或更多的响应数据时，每次接收数据组（从0到8字节）的数据1时，该位设置为1。在开始接收下一个数据组之前写0。

**HTRC标志（成功的帧头传输标志）**

只有0可以写入HTRC标志；当1写入时，该位保留了1写入前的值。HTRC标志在帧头传输完成后设置为1。这里，如果LIEn寄存器中的SHIE位为1（允许中断），则产生一个中断。请注意，当HTRC标志设置为1时，帧头的接收完成，不会产生中断。如要将该位清除为0，请向该位写0。

在接收到一个帧头后，将该位读为1后清空，这样就可以检测到新的帧头。

## 20.2.2.18 LIN/UART错误状态寄存器 (LESTn)

复位值 00H R/W

符号	7	6	5	4	3	2	1	0
LESTn	RPE R	-	CSE R	-	FER	FTER	PBER	BER

位	符号	位名称	功能	R/W
0	BER	位错误标志	0: 未检测到位错误。 1: 检测到位错误。	R/W
1	-	保留	该位总是读为0, 写入值应始终为0。	R/W
2	TER	超时错误标志	0: 未检测到帧/响应超时错误。 1: 检测到帧/响应超时错误。	R/W
3	FER	帧错误标志	0: 未检测到帧错误。 1: 检测到帧错误。	R/W
4	SFER	同步场错误标志	0: 未检测到同步场错误。 1: 已检测到同步场错误。	R/W
5	CSER	校验和错误标志	0: 未检测到校验和错误。 1: 检测到校验和错误。	R/W
6	IPER	ID 奇偶校验错误标志	0: 未检测到ID奇偶性错误。 1: 检测到ID奇偶校验错误。	R/W
7	RPER	响应准备错误标志	0: 未检测到响应准备错误。 1: 检测到响应准备错误。	R/W

在转换到LIN复位模式时, LESTn寄存器会自动清零到00H。

在LIN复位模式下, 禁止对该寄存器的写入。在LIN复位模式下, 该寄存器保留00H。

要清除寄存器中的特定位, 通过使用8位数据传输指令, 把要清除的位写成0, 把其他位写成1。

### BER标志 (位错误标志)

只有0可以写入BER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的BERE位为1 (允许位错误检测), BER标志在位错误检测时设置为1。如要将该位清除为0, 请向该位写入0。

### TER标志 (超时错误标志)

只有0可以写入TER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的TERE位为1 (允许帧/响应超时错误检测), 在检测到帧超时错误或响应超时错误时, TER标志设置为1。如要将该位清除为0, 请向该位写入0。

### FER标志 (帧错误标志)

只有0可以写入FER标志; 当1写入时, 该位保留1写入前的值。如果LEDEn寄存器中的FERE位为1 (允许帧错误检测), 帧错误检测时FER标志设置为1。如要将该位清除为0, 请向该位写入0。

### SFER标志 (同步场错误标志)

只有0可以写入SFER标志; 当1写入时, 该位保留了1写入前的值。如果LEDEn寄存器中的SFERE位为1 (允许同步场错误检测), SFER标志在同步场错误检测时设置为1。如要将该位清除为0, 请向该位写入0。

### CSER标志 (校验和错误标志)

只有0可以写入CSER标志; 当1写入时, 该位保留了1写入前的值。CSER标志在校验和错误检测时设置为1。如要将该位清除为0, 请向该位写入0。

**IPER标志（ID奇偶校验错误标志）**

只有0可以写入IPER标志；当1写入时，该位保留了1写入前的值。如果LEDEn寄存器中的IPERE位为1（允许ID奇偶校验错误检测），IPER标志在ID奇偶校验错误检测时设置为1。如要将该位清除为0，请向该位写入0。

**RPER标志（响应准备错误标志）**

只有0可以写入RPER标志；当1写入时，该位保留在1写入之前的值。响应准备错误检测时，RPER标志设置为1。如要将该位清除为0，请向该位写入0。

Draft Version  
Cmsemicon Confidential

### 20.2.2.19 LIN/UART数据场配置寄存器(LDFCn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LDFCn	LSS	FSM	CS M	RFT	RFDL[3:0]			

位	符号	位名称	功能	R/W
3至0	RFDL[3:0]	响应场长度选择	b3 b0 0000: 0字节 (+校验和) 0001: 1字节 (+校验和) 0010: 2字节 (+校验和) : 0111: 7字节 (+校验和) 1000: 8字节 (+校验和) 禁止使用上述以外的设置。	R/W
4	RCDS	响应场通信方向选择	0:接收 1:传输	R/W
5	LCS	校验和选择	0:经典校验和模式 1:增强型校验和模式	R/W
6	-	保留	该位总是读为0, 写入值应始终为0。	R/W
7	LSS	传输/接收继续选择	0: 接下来要传输/接收的数据组是最后一个。 1: 接下来要传输/接收的数据组不是最后一个。 (继续数据传输/接收, 无需等待下一个帧头的接收)	R/W

**RFDL[3:0]位 (响应场长度选择位)** RFDL位设置响应场数据的长度。

数据长度可以是0到8个字节, 不包括校验和大小。

当RTS位为0时, 设置该位 (响应传输/接收停止)。

<R> 当要传送和接收9字节或更多的响应数据时, 只有最后一个数据组 (LSS位为0) 包括校验和, 其他组 (LSS位为1) 不包括校验和。

**RCDS位 (响应场通信方向选择位)**

RCDS位设置响应场/唤醒信号的通信方向。

设置为0时, 在响应场进行接收。在LIN唤醒模式下, 执行唤醒接收 (计算输入信号的低电平宽度)。

设置为1时, 在响应场进行传输。在LIN唤醒模式下, 执行唤醒传输。当模块处于LIN工作模式时, 在LTRCn寄存器中的RTS位为0时设置该位 (响应传输/接收停止)。当模块处于LIN唤醒模式时, 在LTRCn寄存器中的RTS位为0时设置该位 (帧头接收或唤醒传输/接收停止)。

当要传输或接收9个字节或更多的响应数据时, 在第一个数据组到最后一个数据组之后不要改变RCDS位的设置。

**LCS位（校验和选择位）** LCS位设置校验和模式。

设置为0时，选择经典型校验和模式。

设置为1时，选择增强型校验和模式。

当使用超时错误时（LEDEn寄存器中的TERE位为1），具体超时时间取决于该位的设置。详情请参考20.4.6错误状态。当响应场长度为0字节时（RFDL位为0），不要将该位设为1（增强型模式）。

当要传输或接收9字节或更多的响应时，在第一个数据组到最后一个数据组之后不要改变LCS位的设置。

在9字节或更多的响应数据通信期间，只有最后一个数据组（LSS位为0）包括校验和，其他组（LSS位为1）不包括校验和。

当LTRCn寄存器中的RTS位为0（响应传输/接收停止）时，设置该位。

**LSS位（发送/接收持续选择位）**

LSS位显示下一个要传输或接收的数据组不是最后一个。

设置为0时，数据和校验和将传送或接收，因为下一个要传送或接收的数据组是最后一个。

设置为1时，只传输或接收数据，不包括校验和，因为要传输或接收的下一个数据组不是最后一个。

在LIN通信期间，不要将此位设为1。

当LTRCn寄存器中的RTS位为0（响应传输/接收停止）时，设置该位。

### 20.2.2.20 LIN/UARTID缓冲寄存器 (LIDBn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LIDBn	IDP[1:0]		ID[5:0]					

位	符号	位名称	功能	RW
5至0	ID[5:0]	ID	保存在ID场收到的6位ID值。	RW
7,6	IDP[1:0]	奇偶校验	保存ID场收到的奇偶校验位(P)。	RW

允许在帧头接收完成后对LIDBn寄存器的写入。

在LIN模式下(LIN工作模式或LIN唤醒模式)，禁止写入。

在LIN自测模式下，该寄存器的操作如下：

在通信前写入要传输的值。在帧传输/接收完成后(回环后)，可以从寄存器中读取接收值的反转值。关于LIN自测模式的详情，请参考20.6LIN自测模式。

#### ID[5:0]位 (ID位)

ID位保存LIN帧ID场收到的6位ID值。

#### IDP[1:0]位 (奇偶校验位)

IDP位保存LIN帧ID场收到的奇偶校验位(P0和P1)。

IDP0位是P0，IDP1位是P1。

当LEDEn寄存器中的IPERE位为1时(允许ID奇偶校验)，接收到的值与内部预计算的值进行核对，如果它们不一致，设置IPER位(ID奇偶校验错误标志)。



### 20.2.2.21 LIN校验和缓冲寄存器 (LCBRn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LCBRn								

位	功能	RW
7至0	保存传输或接收的校验和值。	RW

在LIN模式下，该寄存器的操作如下：

- 当LDFCn寄存器中的RCDS位为1（传输）：

传输的值可以从寄存器中读取。对该寄存器的写入是无效的。

- 当LDFCn寄存器中的RCDS位为0（接收）：

收到的值可以从寄存器中读取。对该寄存器的写入是无效的。

当要传输或接收9字节或更多的响应数据时，校验和只附加到最后一个数据组；该寄存器不对其他数据组进行更新。

在LIN自测模式下，该寄存器的操作如下：

- 当LDFCn寄存器中的RCDS位为1（传输）：

在帧传输完成后（回环后），可以从寄存器中读取收到的反转值。

- 当LDFCn寄存器中的RCDS位为0（接收）：

在通信前写入要接收的值。在帧传输/接收完成后（回环后），可以从寄存器中读取接收值的反转值。

关于LIN自测模式的详情，请参考20.6LIN自测模式。

当LTRCn寄存器中的FTS位为0时，设置LCBRn寄存器（停止帧传输或唤醒传输/接收）。

### 20.2.2.22 LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LDBnm								

位	功能	设置范围	R/W
7至0	设置要传输的数据, 或允许读取收到的数据。	00H至FFH	R/W

用于响应传输:

这些寄存器设置了要在响应场中传输的数据。

当RTS位为0时, 设置这些寄存器(停止响应的接收/传送)。

用于响应接收:

这些寄存器持有在响应场中收到的数据。

收到的数据被覆盖。如果检测到错误, 中断前接收的数据将保存在寄存器中。

当RTS位为1时, 不要读取这些寄存器(响应传输/接收开始)。

在LIN自测模式下, 该寄存器的操作如下:

在通信前写入要传输的值。在帧传输/接收完成后(回环后), 可以从寄存器中读取接收值的反转值。关于LIN自测模式的详情, 请参考20.6LIN自测模式。

## 20.2.3 UART的寄存器

### 20.2.3.1 输入切换控制寄存器 (ISC)

ISC寄存器中的ISC2位用于LIN/UART模块 (LIN)。

位2设置为1，即选择LIN/UART模块的串行数据输入引脚的输入信号作为外部中断输入。

这个寄存器可以由8位的内存操作指令来设置。

复位信号的产生将该寄存器设置为00H。

40020473H      复位值:      R/W  
00H

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	ISC2	ISC1	ISC0

ISC2	外部中断 INTP11 的输入切换
0	将 INTP11 引脚的输入信号用作外部中断的输入。
1	将 LRXD0 引脚的输入信号用作外部中断的输入。

**注意事项** 第7至3位应始终设置为0。

### 20.2.3.2 外围寄存器2 (PER2)

PER2寄存器用于允许或禁止对外围硬件的时钟供应。停止向未使用的硬件提供时钟，以降低功耗和噪声。要使用此寄存器控制的外围功能，在指定外围功能的初始设置之前，请设置（1）与每个功能相对应的位。

这个寄存器可以由1位或8位的内存操作指令来设置。

复位信号产生将此寄存器设置为00H。

4002081BH      复位值：      R/W  
00H

符号	7	6	5	4	3	2	1	0
PER2	0	0	0	0	0	0	SPIE N	LINE N

LINEN	LIN模块的输入时钟控制
0	停止提供输入时钟。 •不能写LIN模块使用的SFR。 •LIN模块处于复位状态。
1	提供输入时钟。 •能读写LIN模块使用的SFR。

### 20.2.3.3 LIN时钟选择寄存器 (LINCKSEL)

该寄存器用于控制提供给LIN的通信时钟源。

4002081CH      复位值:      R/W  
00H

符号	7	6	5	4	3	2	1	0
LINCKSEL	0	0	0	LINMCKE	0	0	0	LINMCK

LINMCKE	供应或停止 LIN 通信时钟源的控制
0	停止 LIN 通信时钟源供应。
1	允许 LIN 通信时钟源供应。

LINMCK	选择 LIN 通信时钟源的控制
0	选择 fCLK 的 2 分频时钟。
1	选择 fMX 时钟。

注意事项1.在将LINnMCKE (n=0) 位设置为1 (提供工作时钟) 之前, 用LINnMCK位选择LINn工作时钟。

2.当在SNOOZE模式下操作LINn时, 将LINnMCK位设为0。

3.如果LINnMCK设置为1, 则不使用超时错误检测。

在这种情况下, 将至少1.2倍的LIN通信时钟源的频率设置成fCLK时钟。

### 20.2.3.4 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)

更多细节, 详见24.3.3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)。

### 20.2.3.5 LIN唤醒波特率选择寄存器 (LWBRn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LWBRn	NSPB[3:0]			LPRS[2:0]			-	

位	符号	位名称	功能	R/W
0	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
3 至 1	LPRS [2:0]	预分频器时钟选择	b3 b1 000:1/1 001:1/2 010:1/4 011:1/8 100:1/16 101:1/32 110:1/64 111:1/128	R/W
7 至 4	NSPB [3:0]	位采样计数选择	b7 b4 0000:16 采样 0101:6 采样 0110:7 采样 0111:8 采样 1000:9 采样 1001:10 采样 1010:11 采样 1011:12 采样 1100:13 采样 1101:14 采样 1110:15 采样 1111:16 采样 禁止使用上述以外的设置。	R/W

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 设置LWBRn寄存器。

**LPRS[2:0]位 (预分频器时钟选择位)** LPRS位选择预分频器的分频比。

LIN通信时钟源的频率是根据这个预分频器划分的。

**NSPB[3:0]位 (位采样计数选择位)**

NSPB位选择单个Tbit (波特率的倒数) 中的采样次数。

在UART模式下 (LIN/UART模式寄存器中的LIN/UART模式选择位=01b), NSPB位可设置为6到16采样。

### 20.2.3.6 LIN/UART波特率预分频器寄存器 (LBRPn)

符号	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
LBR Pn	LBRP1								LBRP0							

位	功能	设置范围	R/W
15 至 0	假设该寄存器中设置的值为 L (0~65535)，波特率预分频器将预分频器时钟的频率除以 L+1。	0000H 至 FFFFH	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LBRPn寄存器。

假设该寄存器中设置的值为L，波特率预分频器将LWBRn寄存器中LPRS位（预分频器时钟选择位）选择的时钟频率除以L+1。

LBRPn寄存器可以通过以下寄存器以8位访问。

- 低8位：LIN/UART波特率预分频器0寄存器（LBRPn0）；
- 高8位：LIN/UART波特率预分频器1寄存器（LBRPn1）；

### 20.2.3.7 UART待机控制寄存器 (LUSCn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LUSCn	0	0	0	0	0	URD CC	USEC	UWC

位	符号	位名称	功能	RW
0	UWC	UART 待机唤醒控制	0:禁止从 STOP 模式开始接收。 1:允许从 STOP 模式开始接收。	RW
1	USEC	UART 待机错误控制	0:允许错误检测中断生成。 1:禁止错误检测中断生成。	RW
2	URDCC	UART 待机接收数据比较控制	0:禁止在 SNOOZE 模式下比较接收的数据和 LIDBn 寄存器的值。 1:允许在 SNOOZE 模式下比较接收的数据和 LIDBn 寄存器的值。	RW
7 至 3	-	保留	该位总是读为 0, 写入值应始终为 0。	RW

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LUSCn寄存器。

#### UWC位 (UART待机唤醒控制位)

在STOP模式下，当检测到接收引脚的下降沿时，UWC位允许或禁止转换到SNOOZE模式。

设置为0时，在STOP模式下检测到接收引脚的下降沿不会导致转换到SNOOZE模式，因此不会启动接收。

设置为1时，在STOP模式下检测到接收引脚的下降沿会导致转换到SNOOZE模式，从而启动接收。

#### USEC位 (UART待机错误控制位)

USEC位在SNOOZE模式下检测到错误或状态变化时允许或禁止中断生成。

设置为0时，如果在SNOOZE模式下检测到错误（帧错误或奇偶校验错误）或状态变化（检测扩展位），相应的标志设置为1，从而生成错误检测中断。

设置为1时，如果在SNOOZE模式下检测到错误（帧错误或奇偶校验错误）或状态变化（检测扩展位），相应的标志不设置为1，从而不生成错误检测中断，模块转换到STOP模式。

当UWC位为0（禁止从STOP模式开始接收）时，不要将此位设置为1（禁止生成错误检测中断）。

当UWC位设置为1时，该位生效（允许从STOP模式开始接收）。

#### URDCC位 (UART待机接收数据比较控制位)

URDCC位允许或禁止在SNOOZE模式下比较收到的数据与LIDBn寄存器的值。

设置为0时，在SNOOZE模式下收到的数据不会与LIDBn寄存器的值进行比较，并产生相应的中断。

设置为1时，在SNOOZE模式下收到的数据与LIDBn寄存器的值进行比较，如果它们一致，就会产生成功接收中断。如果它们不一致，则不产生中断，但模块会转换到STOP模式。

当UWC位为0（禁止从STOP模式开始接收）时，不要将该位设为1（允许比较SNOOZE模式下接收的数据和LIDBn寄存器的值）。

当该位应设置为1时（允许比较在SNOOZE模式下接收的数据和LIDBn寄存器的值），请务必将位长设置为8位（LBFCn寄

寄存器中的UBLS位为0；8位UART通信），并将LUORn1寄存器中的UEBE位设置为0；禁止扩展位工作）。

当UWC位设置为1时，该位生效（允许从STOP模式开始接收）。



### 20.2.3.8 LIN/UART模式寄存器 (LMDn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LMDn	-	-	LRDN FS	-	-	-	LMD[1:0]	

位	符号	位名称	功能	RW
1,0	LMD[1:0]	LIN/UART 模式选择	b1b0 01: UART 模式	RW
4至2	-	保留	该位总是读为0, 写入值应始终为0。	RW
5	LRDNFS	禁止 LIN 接收数据噪声过滤	0: 允许噪声滤波器。 1: 禁止噪声滤波器。	RW
6	-	保留	该位总是读为0, 写入值应始终为0。	RW
7	-	保留	该位总是读为0, 写入值应始终为0。	RW

当LMSTn寄存器中的OMM0位为0时 (LIN复位模式), 设置LMDn寄存器。

#### LMD[1:0]位 (LIN/UART模式选择位)

LMD位选择LIN/UART模块模式。

LIN/UART模块作为UART使用, 需将该位设置为01b。

设置为01b时, LIN/UART模块作为UART运行。

#### LRDNFS位 (LIN接收数据噪声过滤禁止位)

LRDNFS位在接收数据时允许或禁止噪声滤波器。

设置为0时, 接收数据时允许噪声滤波器。

设置为1时, 接收数据时禁止噪声滤波器。

### 20.2.3.9 LIN间隔场配置寄存器/UART配置寄存器 (LBFCn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LBFCn	-	UTP S	URP S	UPS[1:0]		USBL S	UBO S	UBLS

位	符号	位名称	功能	R/W
0	UBLS	UART 字符长度选择	0: 8 位 UART 通信 1: 7 位 UART 通信	R/W
1	UBOS	UART 传输格式选择	0: LSB 优先 1: MSB 优先	R/W
2	USBLS	UART 停止位长度选择	0: 一个停止位 1: 两个停止位	R/W
4,3	UPS[1:0]	UART 奇偶校验选择	b4b3 00: 无校验 01: 偶校验 10: 0 校验 11: 奇校验	R/W
5	URPS	UART 输入极性选择	0: 接收数据按原样输入。 1: 接收数据在输入前反转。	R/W
6	UTPS	UART 输出极性选择	0: 传输数据按原样输出。 1: 传输数据在输出前反转。	R/W
7	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LBFCn寄存器。

#### UBLS位 (UART字符长度选择位)

UBLS位设置UART通信帧的一个字符的长度。

设置为0时，通信帧中的一个字符长为8位。

设置为1时，通信帧中的一个字符长为7位。

当一个用于通信的UART帧的字符长度为9位时，设置该位无效（LUORn1寄存器的UEBE位为1）。

#### UBOS位 (UART传输格式选择位)

UBOS位设置UART通信数据的位序。

设置为0时，数据的传输以LSB优先。

设置为1时，数据的传输以MSB优先。

#### USBLS位 (UART停止位长度选择位)

USBLS位设置UART通信中的停止位长度。

设置为0时，以1个停止位进行传输。

设置为1时，以2个停止位进行传输。

**UPS[1:0]位（UART奇偶校验选择位）**

UPS位设置UART通信的奇偶校验。

设置为00b时，通信中不使用奇偶校验。

- 传输  
奇偶校验位不附加在传输数据上。
- 接收  
奇偶校验位不附加在接收数据上，因此不会造成奇偶校验错误。

设置为01b时，在通信中使用偶校验。

- 传输  
当传输数据中1的数量为奇数时，附加1作为奇偶校验位，而当传输数据中1的数量为偶数时，附加0作为奇偶校验位。
- 接收  
当收到的数据中包括奇偶校验位的1的数量为奇数时，就会发生奇偶校验错误。

设置为10b时，在通信中使用0校验。

- 传输  
无论传输数据中有多少个1，都会附加一个0作为奇偶校验位。
- 接收  
由于不检查奇偶校验位的值，所以不会造成奇偶校验错误。

在11b设置下，通信中使用奇校验。

- 传输  
当传输数据中1的数量为奇数时，附加0作为奇偶校验位，而当传输数据中1的数量为偶数时，附加1作为奇偶校验位。
- 接收  
当收到的数据中包括奇偶校验位的1的数量为偶数时，就会发生奇偶校验错误。

**URPS位（UART输入极性选择位）**

URPS位设置UART通信中的输入极性。

设置为0时，接收数据按原样输入。

设置为1时，接收数据在输入前反转。

该位的设置适用于UART帧的所有位。

在半双工通信中，将该位和UTPS位设置为同一数值。

**UTPS位（UART输出极性选择位）**

UTPS位设置UART通信中的输出极性。

设置为0时，传输数据按原样输出。

设置为1时，传输数据在输出前反转。

该位的设置适用于UART帧的所有位。

在半双工通信中，将该位和URPS位设置为同一数值。

### 20.2.3.10 LIN/UART空间配置寄存器 (LSCn)

复位值: R/W  
00H

符号	7	6	5	4	3	2	1	0
LSCn	-	-	IBS[1:0]		-			

位	符号	位名称	功能	R/W
2至0	-	保留	该位总是读为0，写入值应始终为0。	R/W
3	-	保留	该位总是读为0，写入值应始终为0。	R/W
5,4	IBS[1:0]	字节间隔选择	b5b4 00:0 Tbit 01:1 Tbit 10:2 Tbits 11:3 Tbits	R/W
7,6	-	保留	该位总是读为0，写入值应始终为0。	R/W

当LMSTn寄存器中的OMM0位为0时（LIN复位模式），设置LSCn寄存器。

#### IBS[1:0]位（字节间隔选择位）

IBS位设置使用UART缓冲传输时UART帧之间的间隔宽度。

可以设置0Tbit到3Tbit。

当传输缓冲（LUTDRn寄存器）和等待传输缓冲（LUWTDRn寄存器）传输执行时，将IBS[1:0]位设置为00b。

### 20.2.3.11 LIN/UART错误检测寄存器 (LEDEn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LEDEn	-	-	-	-	FERE	OERE	-	BERE

位	符号	位名称	功能	R/W
0	BERE	位错误检测允许	0:禁止位错误检测。 1:允许位错误检测。	R/W
1	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
2	OERE	过载运行错误检测允许	0:禁止过载运行错误检测。 1:允许过载运行错误检测。	R/W
3	FERE	帧错误检测允许	0:禁止帧错误检测。 1:允许帧错误检测。	R/W
4	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
5	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
6	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
7	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

当LMSTn寄存器中的OMM0位为0时 (LIN复位模式), 设置LEDEn寄存器。

#### BERE位 (位错误检测允许位)

BERE位允许或禁止对位错误的检测。

设置为0时, 不检测位错误。

设置为1时, 检测到位错误。

当该位设置为1时, 检测结果在LESTn寄存器的BER标志中显示。

当LIN/UART模块在全双工模式下使用时, 不要将该位设置为1。

有关位错误的详情, 请参考20.5.5错误状态。

当LWBRn寄存器中的NSPB位为0101b (6采样), LMDn寄存器中的LRDNFS位为0 (噪声滤波器正在使用) 时, 不要设置该位。

#### OERE位 (过载运行错误检测允许位)

OERE位允许或禁止对过载运行错误的检测。

设置为0时, 不检测过载运行错误。

设置为1时, 则检测到过载运行错误。

当该位设置为1时, 检测结果在LESTn寄存器的OER标志中显示。

关于过载运行错误的详情, 请参考20.5.5错误状态。

#### FERE位 (帧错误检测允许位)

FERE位允许或禁止对帧错误的检测。

设置为0时, 不检测帧错误。

设置为1时, 检测到帧错误。

该位设置为1时, 检测结果在LESTn寄存器的FER标志中显示。

关于帧错误的详情, 请参考20.5.5错误状态。

### 20.2.3.12 LIN/UART控制寄存器 (LCUCn)

复位值：  
00H R/W

符号	7	6	5	4	3	2	1	0
LCUCn	-	-	-	-	-	-	-	OM0

位	符号	位名称	功能	RW
0	OM0	LIN 复位	0:LIN 复位模式开启。 1:LIN 复位模式取消。	RW
1	-	保留	该位总是读为 0，写入值应始终为 0。	RW
7 至 2	-	保留	该位总是读为 0，写入值应始终为 0。	RW

在向该寄存器写入一个值后，在写入另一个值之前，应确认所写入的值确实已在LMSTn寄存器中已显示。

#### OM0位 (LIN复位位)

OM0位可以选择向LIN复位模式的转换或取消LIN复位模式。

设置为0时，开启LIN复位模式。

设置为1时，取消LIN复位模式。

### 20.2.3.13 LIN/UART传输控制寄存器 (LTRCn)

复位值:  
00H R/W

符号	7	6	5	4	3	2	1	0
LTRCn	-	-	-	-	-	-	RTS	-

位	符号	位名称	功能	R/W
0	-	保留	该位总是读为0，写入值应始终为0。	R/W
1	RTS	UART 缓冲传输开始	0:UART 缓冲传输禁止。 1:UART 缓冲传输允许。	R/W
2	-	保留	该位总是读为0，写入值应始终为0。	R/W
7至3	-	保留	该位总是读为0，写入值应始终为0。	R/W

#### RTS位 (UART缓冲传输起始位)

将RTS位设置为1，从UART缓冲传输数据。

该位只能写入1，不能写入0。

当LUOERn寄存器中的UTOE位为1（允许传输），LSTn寄存器中的UTS位为0（传输未进行）时，向该位写入一个值。

一旦设置，无论是否发生错误，该位在与数据单元数设置相对应的数据量传输完成后（由LDFCn寄存器中的MDL位决定）自动清零。

转换到LIN复位模式时，该位会自动清除为0。

LMSTn寄存器中的OMM0位为0（LIN复位模式）时，向该位写值是无效的。

LDFCn寄存器中的UTSW位设置为1时（在等待停止位接收完成后请求从UART缓冲中传输），只有在接收到停止位后才能将1写入该位。

### 20.2.3.14 LIN/UART模式状态寄存器 (LMSTn)

复位值：  
00H

R/W

符号	7	6	5	4	3	2	1	0
LMSTn	-	-	-	-	-	-	OMM1	OMM0

位	符号	位名称	功能	RW
0	OMM0	LIN 复位状态监控	0: 模块处于 LIN 复位模式。 1: 模块不处于 LIN 复位模式。	R
1	-	保留	该位总是读为 0, 写入值应始终为 0。	R
7 至 2	-	保留	该位总是读为 0, 写入值应始终为 0。	RW

#### OMM0位 (LIN复位状态监控位)

OMM0位指示当前是否设置了LIN复位模式。

设置为0时, LIN/UART模块处于LIN复位模式。

设置为1时, LIN/UART模块不处于LIN复位模式。



### 20.2.3.15 LIN/UART状态寄存器 (LSTn)

复位值：  
00H R/W

符号	7	6	5	4	3	2	1	0
LSTn	-	-	URS	UTS	ERR	-	-	FTC

位	符号	位名称	功能	R/W
0	FTC	成功的帧/唤醒传输标志	0: 响应或唤醒传输尚未完成。 1: 响应或唤醒传输已经完成。	R/W
1	-	保留	该位总是读为0, 写入值应始终为0。	R/W
2	-	保留	该位总是读为0, 写入值应始终为0。	R/W
3	ERR	错误检测标志	0: 没有检测到错误。 1: 检测到错误。	R
4	UTS	传输状态标志	0: LIN/UART 模块不在传输数据。 1: LIN/UART 模块正在传输数据。	R
5	URS	接收状态标志	0: LIN/UART 模块不在接收数据。 1: LIN/UART 模块正在接收数据。	R
6,7	-	保留	该位总是读为0, 写入值应始终为0。	R/W

在转换到LIN复位模式时, LSTn寄存器会自动清零到00H。

在LIN复位模式下, 禁止对该寄存器的写入。在LIN复位模式下, 该寄存器保留为00H。

要清除寄存器中的特定位, 通过使用8位数据传输指令, 把要清除的位写成0, 把其他位写成1。

#### FTC标志 (成功的缓冲传输标志)

只有0可以写入FTC标志; 当1写入时, 该位保留了1写入前的值。无论是否发生错误, UART缓冲数据传输完成后该位设置为1, 该数据等于LDFCn寄存器中用MDL位设置的数据单元数。此处产生一个中断。

如要将该位清除为0, 请向该位写0。

#### ERR标志 (错误检测标志)

检测到错误时, ERR标志设置为1 (LSTn寄存器的任一标志为1)。此处产生一个中断。请注意, 当检测到错误、扩展位或ID匹配时, ERR标志设置为1, 不会产生中断。如需将该位清除为0, 请向LSTn寄存器中的UPER、IDMT、EXBT、FER、OER和BER标志写0。这样就把ERR标志清除为0。

#### UTS标志 (传输状态标志)

UTS标志在传输开始时设置为1。在传输过程中, 该标志保持为1。

在以下情况下开始传输:

- 传输数据设置在LUTDRn或LUWTDn寄存器中。
- LTRCn寄存器中的RTS位设置为1。

UTS标志在传输结束时清空为0。当传输停止时, 该标志保持为0。

在以下情况下结束传输:

- LUTDRn或LUWTDn寄存器中设置的数据的传输已经完成, 并且下一个传输数据未设置。
- UART缓冲的数据传输已经完成 (LTRCn寄存器中的RTS位设置为0)。

**URS标志（接收状态标志）**

URS标志在接收开始时设置为1。在接收期间，该标志保持为1。

在以下情况下开始接收：

- 检测到起始位。

URS标志在接收结束时清空为0。当接收停止时，该标志保持为0。

在以下情况下，接收结束：

- 采样了第一个停止位。

Draft Version  
Cmsemicon Confidential

### 20.2.3.16 LIN/UART错误状态寄存器 (LESTn)

复位值：  
00H R/W

符号	7	6	5	4	3	2	1	0
LESTn	-	UPER	IDMT	EXBT	FER	OER	-	BER

位	符号	位名称	功能	R/W
0	BER	位错误标志	0: 未检测到位错误。 1: 检测到位错误。	R/W
1	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
2	OER	过载运行错误标志	0: 未检测到过载运行错误。 1: 已检测到过载运行错误。	R/W
3	FER	帧错误标志	0: 未检测到帧错误。 1: 检测到帧错误。	R/W
4	EXBT	扩展位检测标志	0: 未检测到扩展位。 1: 已检测到扩展位。	R/W
5	IDMT	ID 匹配标志	0: 收到的数据与 ID 值不一致。 1: 收到的数据与 ID 值一致。	R/W
6	UPER	奇偶校验错误标志	0: 未检测到奇偶校验错误。 1: 检测到了奇偶校验错误。	R/W
7	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

在转换到LIN复位模式时，LESTn寄存器会自动清零到00H。

在LIN复位模式下，禁止对该寄存器的写入。在LIN复位模式下，该寄存器保留00H。

要清除寄存器中的特定位，通过使用8位数据传输指令，把要清除的位写成0，把其他位写成1。

#### BER标志（误码标志）

只有0可以写入BER标志；当1写入时，该位保留了1写入前的值。如果LESTn寄存器中的BERE位为1（允许位错误检测），BER标志在位错误检测时设置为1。如要将该位清除为0，请向该位写入0。

#### OER标志（过载运行错误标志）

只有0可以写入OER标志；当1写入时，该位保留了1写入前的值。如果LESTn寄存器中的OERE位为1，OER标志在过载运行错误检测时设置为1（允许过载运行错误检测）。如要将该位清除为0，请向该位写入0。

#### FER标志（帧错误标志）

只有0可以写入FER标志；当1写入时，该位保留1写入前的值。如果LESTn寄存器中的FERE位为1（允许帧错误检测），帧错误检测时FER标志设置为1。

在SNOOZE模式下，要把这个标志设置为1，还应该满足以下条件。

- LUSCn寄存器中的UWC位为1（允许从STOP模式开始接收）。
- LUSCn寄存器中的USEC位为0（允许错误检测中断生成）。

如要将该位清除为0，请向该位写0。

### EXBT标志（扩展位检测标志）

EXBT标志只能写入0；当写入1时，该位保留了写入1之前的值。当LUORn1寄存器中的UEBE位为1（允许扩展位），且接收的扩展位与LUORn1寄存器中的UEDL位值一致时，该标志设置为1。

在SNOOZE模式下，要把这个标志设置为1，还应该满足以下条件。

- LUSCn寄存器中的UWC位为1（允许从STOP模式开始接收）。
- LUSCn寄存器中的USEC位为0（允许错误检测中断生成）。
- LUORn1寄存器中的UECD位为0（允许扩展位的比较）。

如要将该位清除为0，请向该位写0。

### IDMT标志（ID匹配标志）

只有0可以写入IDMT标志；当1写入时，该位保留了1写入前的值。当满足以下所有条件时，IDMT标志设置为1。

- LUORn1寄存器中的UEBE位为1（允许扩展位操作）。
- LUORn1寄存器中的UECD位为0（允许扩展位比较）。
- LUORn1寄存器中的UEBDCE位为1（允许扩展位检测后的数据比较）。
- 收到的扩展位与LUORn1寄存器中的UEBDL位的值一致。
- 8位接收数据不包括扩展位与LIDBn寄存器一致的值。

如要将该位清除为0，请向该位写0。

### UPER标志（奇偶校验错误标志）

IDMT标志只能写入0；当写入1时，该位保留了写入1之前的数值。

UPER标志在检测到奇偶校验错误时设置为1。

在SNOOZE模式下，要把这个标志设置为1，还应该满足以下条件。

- LUSCn寄存器中的UWC位为1（允许从STOP模式开始接收）。
- LUSCn寄存器中的USEC位为0（允许错误检测中断生成）。

如要将该位清除为0，请向该位写0。

### 20.2.3.17 LIN/UART数据场配置寄存器(LDFCn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LDFCn	-	-	UTSW	-	MDL[3:0]			

位	符号	位名称	功能	R/W
3至0	MDL[3:0]	UART 缓冲数据长度选择	b3 b0 0000: 9 数据 0001: 1 数据 0010: 2 数据 0011: 3 数据 0100: 4 数据 0101: 5 数据 0110: 6 数据 0111: 7 数据 1000: 8 数据 1001: 9 数据 禁止使用上述以外的设置。	R/W
4	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W
5	UTSW	传送开始等待	0: 当请求 UART 缓冲传输时, 立即开始传输。 1: 当请求 UART 缓冲传输时, 在等待停止位接收完成后开始传输。	R/W
7,6	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

**MDL[3:0]位 (UART缓冲数据长度选择位)** MDL位设置UART缓冲的数据长度。当RTS位为1 (允许UART缓冲传输) 时, 向该位写值是无效的。

#### UTSW位 (传输开始等待位)

UTSW位控制UART缓冲传输的开始时间。

设置为0时, 当请求UART缓冲传输时, 立即开始传输。

设置为1时, 当请求UART缓冲传输时, 在等待停止位的接收完成后开始传输。

请注意, 即使停止位长度经LDFCn寄存器中的USBLS位设置为2位, 等待时间也只有1位。

当LDFCn寄存器中的RTS位设置为1时, 该位生效。

当RTS位为1时, 向该位写值是无效的 (允许UART缓冲传输)。

除了在半双工通信中从接收切换到传输外, 不要将此位设置为1。

### 20.2.3.18 LIN/UARTID缓冲寄存器 (LIDBn)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LIDBn								

位	功能	R/W
7 至 0	设置 ID 值, 用于扩展位/数据比较或 SNOOZE 模式下的数据比较的参考。	R/W

当LUORn1寄存器中的UEBE位为1（允许扩展位）和UEBDCE位为1（允许扩展位检测后的数据比较）时，将与接收数据进行比较的值设置到LIDBn寄存器中。

当LUSCn寄存器中的UWC位为1（允许从STOP模式开始接收）和URDCC位为1（在SNOOZE模式下允许比较接收数据和LIDBn寄存器的值）时，将与接收数据比较的值设置到LIDBn寄存器中。

当LSTn寄存器中的URS位为0（接收未进行）时，设置LIDBn寄存器。

### 20.2.3.19 UART数据缓冲区0寄存器 (LUDBn0)

复位值: 00H  
R/W

符号	7	6	5	4	3	2	1	0
LUDBn0								

位	功能	设置范围	RW
7 至 0	设置要从 UART 缓冲区传输的数据。	00H 至 FFH	RW

LUDBn0寄存器设置从UART缓冲区中首先传输的数据，用于9位数长的数据传输（LDFCn寄存器中的MDL位为0H或9H）。

在RTS位为0（禁止UART缓冲传输）时，向LUDBn0寄存器写入。  
关于UART缓冲的详情，请参考20.5.3传输数据的缓冲处理。

### 20.2.3.20 LIN/UART数据缓冲区m寄存器 (LDBnm) (m=1至8)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LDBnm								

位	功能	设置范围	R/W
7至0	设置要从UART缓冲区传输的数据。	00H至FFH	R/W

这些寄存器设置了要从UART缓冲传输的数据。

当RTS位为0时，写入这些寄存器（禁止UART缓冲传输）。

关于UART缓冲的详情，请参考20.5.3传输数据的缓冲处理。



### 20.2.3.21 UART操作允许寄存器 (LUOERn)

复位值：  
00H R/W

符号	7	6	5	4	3	2	1	0
LUOERn	-	-	-	-	-	-	UROE	UTOE

位	符号	位名称	功能	R/W
0	UTOE	允许传输功能	0:禁止传输。 1:允许传输。	R/W
1	UROE	允许接收功能	0:禁止接收。 1:允许接收。	R/W
7至2	-	保留	该位总是读为0，写入值应始终为0。	R/W

在转换到LIN复位模式时，LUOERn寄存器自动清除到00H。

在LIN复位模式下，禁止对该寄存器的写入。在LIN复位模式下，该寄存器保留00H。

#### UTOE位 (传输允许位)

UTOE位允许或禁止传输。

设置为0时，禁止传输。

设置为1时，允许传输。

在传输过程中不要清除该位。要在传输过程中取消传输，通过将LCUCn寄存器中的OM0位设置为0（LIN复位模式），将模块置于LIN复位模式。注意，这个操作也会取消接收。

#### UROE位 (接收允许位)

UROE位允许或禁止接收功能。

设置为0时，禁止接收。

设置为1时，允许接收。

在接收过程中不要清除该位。要在接收过程中取消传输，通过将LCUCn寄存器中的OM0位设置为0（LIN复位模式），将模块置于LIN复位模式。注意，这个操作也会取消传输。

从UART缓冲传输数据时，该位必须为0。

### 20.2.3.22 UART选项寄存器1 (LUORn1)

复位值: 00H R/W

符号	7	6	5	4	3	2	1	0
LUOR1n	-	-	-	UECD	UTIGTS	UEBDCE	UEBDR	UEBE

位	符号	位名称	功能	R/W
0	UEBE	扩展位允许	0: 禁止扩展位操作。 1: 允许扩展位操作。	R/W
1	UEBDL	扩展位检测电平选择	0: 选择扩展位的值 0 作为扩展位检测电平。 1: 选择扩展位的值 1 作为扩展位检测电平。	R/W
2	UEBDCE	扩展位/数据比较允许	0: 在检测到扩展位后禁止数据比较。 1: 在检测到扩展位后允许数据比较。	R/W
3	UTIGTS	传输中断生成时序选择	0: 传输开始时产生传输中断。 1: 传输完成时产生传输中断。	R/W
4	UECD	扩展位比较禁止	0: 允许比较接收的扩展位和 UEBDL 位的值。 1: 禁止比较接收的扩展位和 UEBDL 位的值。	R/W
7至5	-	保留	该位总是读为 0, 写入值应始终为 0。	R/W

#### UEBE位 (扩展位允许位)

UEBE位允许或禁止扩展位操作。

设置为0时, 禁止扩展位操作。

设置为1时, 允许扩展位操作。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 设置该位。

当UART缓冲正在使用时, 不要将该位设置为1。

#### UEBDL位 (扩展位检测电平选择位)

当UEBE位为1 (允许扩展位操作) 和UECD位为0 (允许扩展位的比较) 时, UEDL位设置检测为扩展位的电平。

设置为0时, 扩展位的值0是作为扩展位检测的电平。

设置为1时, 扩展位的值1是作为扩展位检测的电平。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 设置该位。

当UART缓冲正在使用时, 不要将该位设置为1。

#### UEBDCE位 (扩展位/数据比较允许位)

UEBDCE位在检测到扩展位后, 允许或禁止不包括扩展位的8位接收数据与LIDBn寄存器的值进行比较。

设置为0时, 在检测到由UEDL位选择的扩展位值作为扩展位后, 禁止比较LURDRn寄存器中的接收数据与LIDBn寄存器的值。

设置为1时, 在检测到由UEBDL位选择的扩展位值作为扩展位后, 允许比较LURDRn寄存器中的接收数据与LIDBn寄存器的值。

当LMSTn寄存器中的OMM0位为0 (LIN复位模式) 时, 设置该位。

当UEBE位设置为0时, 不要将此位设置为1 (禁止扩展位操作)。

当UECD位设置为1时, 不要将该位设置为1 (禁止扩展位的比较)。

当UART缓冲正在使用时, 不要将该位设置为1。

当LUSCn寄存器中的UWC位设置为1时, 不要将此位设置为1 (允许从STOP模式开始接收)。

**UTIGTS位（传输中断生成时序选择位）** UTIGTS位设置传输中断生成时序。

设置为0时，传输中断在传输开始时产生。

设置为1时，传输完成时产生传输中断。

当从UART缓冲进行传输时，设置为0，传输中断只在LDFCn寄存器中用MDL位设置的数据长度的最后一个数据开始传输时产生。

当从UART缓冲进行传输时，设置为1时，只有在LDFCn寄存器中用MDL位设置的数据长度的最后一个数据传输完成后才会产生传输中断。

**UECD位（扩展位比较禁止位）**

当UEBE位为1（扩展位操作允许）时，UECD位允许或禁止接收的扩展位和UEBDL位值之间的比较。

设置为0时，当收到扩展位时，允许比较接收的扩展位和UEBDL位的值。

设置为1时，当收到扩展位时，禁止比较接收的扩展位和UEBDL位的值。

当LMSTn寄存器中的OMM0位为0（LIN复位模式）时，设置该位。

当UART缓冲正在使用时，不要将该位设置为1。

当UEBDCE位设置为1时，不要将此位设置为1（允许扩展位检测后的数据比较）。

### 20.2.3.23 UART传输数据寄存器 (LUTDRn)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUTDRn	-	-	-	-	-	-	-	[8:0]								

位	功能	设置范围	RW
8 至 0	设置要从传输缓冲传输的数据。	000H 至 1FFH	RW
15 至 9	保留。 该位总是读为 0，写入值应始终为 0。	-	RW

LUTDRn寄存器设置要从传输数据寄存器中传输的数据。

在LUOERn寄存器的UTOE位设置为1时，向该寄存器写入数据开始传输。

这个寄存器可以用8位来访问。

在9位通信模式下，不要尝试8位访问。

当UART缓冲的数据传输正在进行时，不要向该寄存器写入数据。

另外，由于LUWTDn寄存器的写入访问，在产生传输请求时，不要向该寄存器写入数据。当连续传输多组数据时，在产生传输中断之前，不要向该寄存器写入其他数据项。

下表是根据传输格式设置的位排列。

项目	LUTDRn								
	8	7	6	5	4	3	2	1	0
7 位; LSB 优先	-	-	第6位	第5位	第4位	第3位	第2位	第1位	第0位
7 位; MSB 优先	-	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位
8 位; LSB 优先	-	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
8 位; MSB 优先	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位
9 位; LSB 优先	第8位	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
9 位; MSB 优先	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位	第8位

### 20.2.3.24 UART接收数据寄存器 (LURDRn)

符号	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
LURDRn	-	-	-	-	-	-	-	[8:0]								

位	功能	设置范围	R/W
8至0	允许从接收缓冲读取接收数据。	000H至1FFH	R
15至9	保留。 该位总是读为0，写入值应始终为0。	-	R/W

LURDRn寄存器允许从接收数据寄存器中读取接收数据。

当LUOERn寄存器中的UROE位为1时，接收数据存储在寄存器中，且数据可以读取。

该寄存器更新接收数据中的停止位。

当奇偶校验或停止位生成错误时，该寄存器也会更新。

如果LEDEn寄存器中的OERE位为1（允许过载运行检测），发生过载运行错误时，该寄存器不更新。如果OERE位为0（禁止过载运行检测），该寄存器在发生过载运行错误时更新。

如果LEDEn寄存器中的OERE位为1（允许过载运行错误检测），在发生接收错误时（过载运行错误、帧错误或奇偶校验错误）读取该寄存器。读取下一个数据而不读取该寄存器会导致过载运行错误。

这个寄存器可以8位为单位进行访问。但是，当使用扩展位时，不要以8位单位访问该寄存器（LUORn1寄存器中的UEBE位为1（允许扩展位操作））。

下表是根据传输格式设置的位排列。

项目	LURDRn									
	8	7	6	5	4	3	2	1	0	
7位；LSB 优先	-	-	第6位	第5位	第4位	第3位	第2位	第1位	第0位	
7位；MSB 优先	-	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位	
8位；LSB 优先	-	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位	
8位；MSB 优先	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位	
9位；LSB 优先	第8位	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位	
9位；MSB 优先	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位	第8位	

### 20.2.3.25 UART等待传输数据寄存器 (LUWTDn)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUWTDn	-	-	-	-	-	-	-	[8:0]								

位	功能	设置范围	RW
8至0	在等待停止位接收完成后, 设置从UART等待传输数据寄存器传输的数据。	000H 至 1FFH	RW
15至9	保留。 该位总是读为0, 写入值应始终为0。	-	RW

LUWTDn寄存器设置UART等待传输数据寄存器中要传输的数据。

在LUOERn寄存器中的UTOE位设置为1时, 向该寄存器写入数据开始传输。

在半双工通信中, 仅在从接收切换到发送时使用该寄存器。

用户只应在接收停止位时向该寄存器写入。

注意, 即使在LBFcn寄存器中的USBLS位将停止位长度设置为2位, 等待时间也只有1位。

读取该寄存器时, 实际上是读取LUTDRn寄存器的值。

在9位通信模式下, 不要尝试8位访问。

当UART缓冲的数据正在传输时, 不要向该寄存器写入数据。

下表是根据传输格式设置的位排列。

项目	LUWTDn								
	8	7	6	5	4	3	2	1	0
7位; LSB 优先	-	-	第6位	第5位	第4位	第3位	第2位	第1位	第0位
7位; MSB 优先	-	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位
8位; LSB 优先	-	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
8位; MSB 优先	-	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位
9位; LSB 优先	第8位	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
9位; MSB 优先	第0位	第1位	第2位	第3位	第4位	第5位	第6位	第7位	第8位

## 20.3 模式

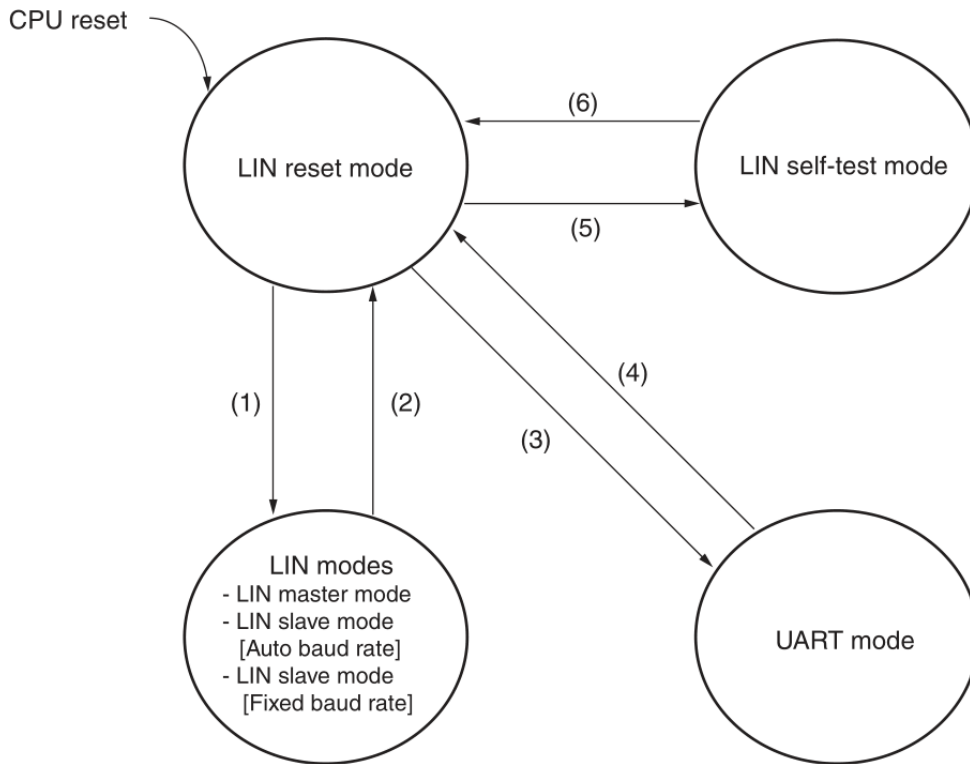
LIN/UART模块提供以下四种模式，模式的选择取决于要执行的具体功能。

- LIN复位模式
- LIN模式  
(LIN主机模式/LIN从机模式[自动波特率]/LIN从机模式[固定波特率])
- UART模式
- LIN自测模式

在LIN复位模式下，对LIN/UART停止供应模块的时钟，从而降低了功耗。

图20-3是模式转换图。表20-4描述了模式转换条件。表20-5列出了每种模式下适用的操作。

图20-3.模式转换



**表20-4.模式转换条件**

步骤	模式转换	转换条件
(1)	LIN 复位模式 (LIN 主机模式) → LIN 模式	LMDn 寄存器中的 LMD 位=00b 和 LCUCn 寄存器中的 OM1 和 OM0 位=01b 或 11b
	LIN 复位模式 (LIN 从机模式[自动波特率]) → LIN 模式	LMDn 寄存器中的 LMD 位=10b 和 LCUCn 寄存器中的 OM1 和 OM0 位=01b 或 11b
	LIN 复位模式 (LIN 从机模式[固定波特率]) → LIN 模式	LMDn 寄存器中的 LMD 位=11b 和 LCUCn 寄存器中的 OM1 和 OM0 位=01b 或 11b
(2)	LIN 模式 → LIN 复位模式	LCUCn 寄存器中的 OM0 位=0b
(3)	LIN 复位模式 → UART 模式	LMDn 寄存器中的 LMD 位=01b 和 LCUCn 寄存器中的 OM0 位=1b
(4)	UART 模式 → LIN 复位模式	LCUCn 寄存器中的 OM0 位=0b
(5)	LIN 复位模式 → LIN 自测模式	见 20.6 LIN 自测模式。
(6)	LIN 自测模式 → LIN 复位模式	见 20.6 LIN 自测模式。

**表20-5.每种模式下适用的操作**

LIN 模式		UART 模式	LIN 自测模式
LIN 主机模式	LIN 从机模式[自动波特率]/LIN 从机模式[固定波特率]。		
帧头传输	帧头接收	UART 传输 UART 接收 错误检测	自测试
响应传输	响应接收		
响应接收	响应接收		
唤醒传输	唤醒接收		
唤醒接收	唤醒接收		
错误检测	错误检测		

通过读取LMDn寄存器中的LMD位或LMSTn寄存器中的OMM0位，可以验证是否已经转换到LIN复位模式、LIN模式或UART模式。

最长模式转换时间（从LSUCn寄存器设置数值到LMSTn寄存器显示数值的最长时间）是三个CPU时钟（fCLK）周期和四个LIN通信时钟源（由LINnMCK选择的LIN/UART模块的输入时钟）周期之和。

关于LIN自测模式的描述，请参考20.6 LIN自测模式。



## 20.3.1 LIN复位模式

LCUCn寄存器中的OM0位设置为0b（LIN复位模式）会转换到LIN复位模式。验证LIN复位模式的转换可以通过确定LMSTn寄存器中的OMM0位已设置为0b（LIN复位模式）。在这种模式下，LIN通信和UART通信功能全部停止，fLIN也停止。

从LIN复位模式，可以转换到LIN模式、UART模式和LIN自测模式。

模式变为LIN复位模式，以下寄存器初始化为它们的复位值，且只要LIN复位模式有效，就会保留它们的初始值。

- LTRCn寄存器
- LSTn寄存器
- LESTn寄存器
- LUOERn寄存器

即使转换到LIN复位模式，以下寄存器也会保留它们之前的值。

- LCHSEL寄存器
- LWBRn寄存器
- LBRPn0寄存器
- LBRPn1寄存器
- LUSCn寄存器
- LMDn寄存器
- LBFCn寄存器
- LSCn寄存器
- LWUPn寄存器
- LIEn寄存器
- LEDEn寄存器
- LDFCn寄存器
- LIDBn寄存器
- LCBRn寄存器
- LUDBn0寄存器
- LDBnm寄存器（m=1至8）
- LUORn1寄存器
- LUTDRn寄存器
- LURDRn寄存器
- LUWTDRn寄存器

## 20.3.2 LIN模式

LIN模式可以在以下子模式下运行：LIN主机模式、LIN从机模式[自动波特率]和LIN从机模式[固定波特率]。

在LIN主机模式下，可以进行以下操作：帧头传输、响应传输、响应接收、唤醒传输、唤醒接收和错误检测。在LIN复位模式下，通过将LMDn寄存器中的LMD位设置为00b（LIN主机模式），LCUCn寄存器中的OM1和OM0位设置为01b或11b设置LIN主机模式，使得LMSTn寄存器中的OMM1和OMM0位转为01b至11b。

在LIN从机模式[自动波特率]和LIN从机模式[固定波特率]下，可以进行帧头接收、响应传输、响应接收、唤醒传输、唤醒接收和错误检测。

LIN从机模式[自动波特率]允许自动检测间隔场和同步场，并根据同步场的测量结果设置1kbps至20kbps区间的波特率。根据目标波特率设置LWBRn寄存器中的LPRS[2:0]位，以便通过LIN通信时钟源频率除以预分频器得到的时钟（预分频器时钟）的频率为列表中的相应值。

[目标波特率]	[预分频器的时钟频率]
1kbps到20kbps:	4MHz <sup>注</sup>
1kbps到小于2.4kbps:	4MHz
2.4kbps至20kbps:	8MHz至12MHz

注 LWBRn寄存器中的NSPB[3:0]位设置为0011b（4采样）。

LIN从机模式下，[固定波特率]允许自动检测间隔场、同步场，并通过波特率发生器预先设定的波特率设置ID场。

LIN复位模式下，通过LMDn寄存器中的LMD位设置为10b（LIN从机模式[自动波特率]），LCUCn寄存器中的OM1和OM0位设置为01b或11b从而设置LIN从机模式[自动波特率]。通过LMDn寄存器中的LMD位设置为11b（LIN从机模式[固定波特率]），和LCUCn寄存器中的OM1和OM0位设置为01b或11b，从而设置LIN从机模式[固定波特率]，使得LMSTn寄存器中的OMM1和OMM0位转为01b或11b。

当在LIN模式内一个子模式变为另一个子模式时，应首先转换到LIN复位模式并改变LMDn寄存器中的LMD位。

LIN模式提供以下两种操作模式：

- LIN工作模式
- LIN唤醒模式

图20-4操作模式的转换。表20-6描述了操作模式的转换条件。

图20-4.操作模式的转换

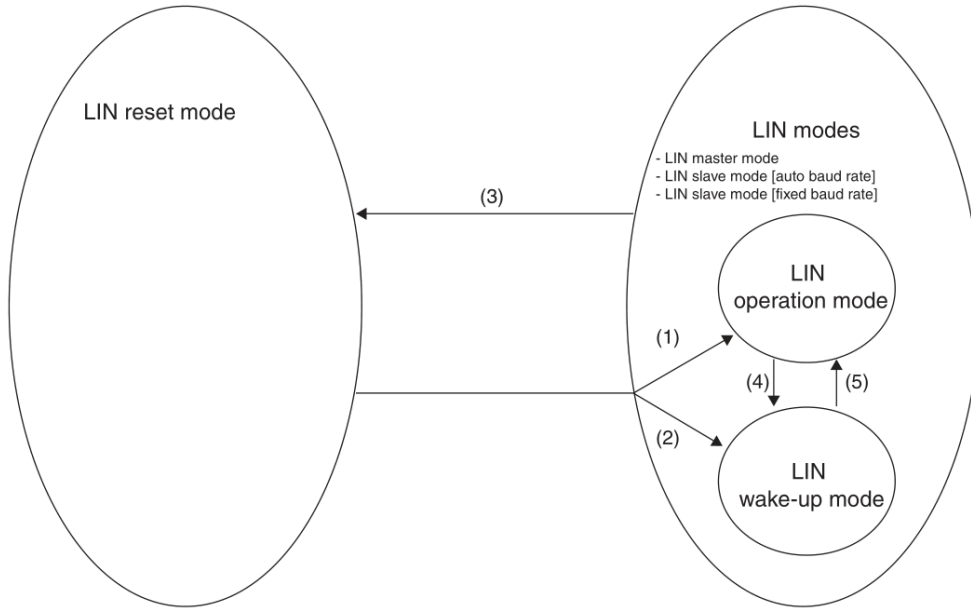


表20-6.操作模式的转换条件

步骤	操作模式转换	转换条件
(1)	LIN 复位模式 → LIN 模式 -LIN 工作模式	LMDn 寄存器中的 LMD 位=00b 或 10b 或 11b 和 LCUCn 寄存器中的 OM1 和 OM0 位=11b
(2)	LIN 复位模式 → LINmode -LIN 唤醒模式	LMDn 寄存器中的 LMD 位=00b 或 10b 或 11b 和 LCUCn 寄存器中的 OM1 和 OM0 位=01b
(3) <sup>注1</sup>	LIN 模式 → LIN 复位模式 -LIN 工作模式 -LIN 唤醒模式	LCUCn 寄存器中的 OM0 位=0b
(4) <sup>注2</sup>	LIN 模式 → LIN 模式 -LIN 工作模式 -LIN 唤醒模式	LCUCn 寄存器中的 OM1 和 OM0 位=01b
(5) <sup>注2</sup>	LIN 模式 → LIN 模式 -LIN 唤醒模式 -LIN 工作模式	LCUCn 寄存器中的 OM1 和 OM0 位=11b

注

1.当LIN/UART模块作为LIN从机工作时（固定波特率），且LIN工作模式转换到LIN复位模式，在PER2寄存器中的LIN0EN位（或LIN1EN位）清除为0后写1。

2.正在通信时（当LTRCn寄存器中的FTS位为1时），不能在LIN工作模式和LIN唤醒模式之间进行转换。

### 20.3.2.1 LIN工作模式

在LIN工作模式下，可以进行帧处理（帧头传输、帧头接收、响应传输、响应接收和错误检测）。

在LIN复位模式转换到LIN模式期间，通过LCUCn寄存器中的OM1和OM0位设置为11b，使模式变为LIN工作模式，从而LMSTn寄存器中的OMM1和OMM0位变为11b。通信设置应在LMSTn寄存器变为11b后进行。

### 20.3.2.2 LIN唤醒模式

在LIN唤醒模式下，可以进行唤醒信号处理（唤醒传输、唤醒接收和错误检测）。

在LIN复位模式转换到LIN模式期间，通过LCUCn寄存器中的OM1和OM0位设置为01b，使模式变为LIN唤醒模式，从而LMSTn寄存器中的OMM1和OMM0位变为01b。通信设置应在LMSTn寄存器变为01b后进行。

Draft Version  
Cmsemicon Confidential

### 20.3.3 UART模式

在LIN复位模式下，通过LMDn寄存器中的LMD位设置为01b（UART模式），LCUCn寄存器中的OM0位设置为1b，使模式变为UART模式，从而LMSTn寄存器中的OMM0位变为1b。通信设置应在LMSTn寄存器变为01b后进行。

### 20.3.4 LIN自测模式

写入LSTCn寄存器可将模式改为LIN自测模式。LSTCn寄存器中的LSTM位为1，表示模式已经转换到LIN自测模式。

更多的操作细节，请参考20.6 LIN自测模式。

Draft Version  
Cmsemicon Confidential

## 20.4 LIN模式

### 20.4.1 操作概述

#### 20.4.1.1 LIN主机模式

##### (a) 帧头传输

图20-5是LIN/UART模块（LIN主机模式）在帧头传输中的操作。表20-7是帧头传输中的处理。

图20-5.帧头传输中的操作

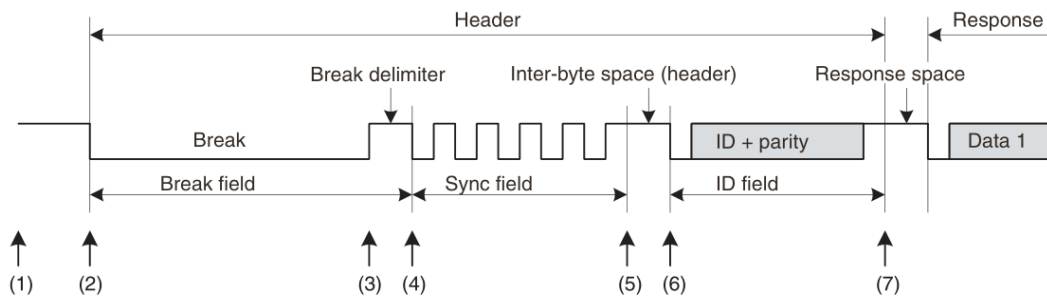


表20-7.帧头传输中的处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>设置波特率</li> <li>设定噪音滤波器的开/关</li> <li>允许中断</li> <li>允许错误检测</li> <li>设置帧配置参数</li> <li>LIN/UART 模块改为 LIN 主机模式；LIN 工作模式</li> <li>设置要传输的帧的信息（ID、奇偶校验、数据长度、响应方向、校验和方法 and 传输数据）。</li> </ul>	等待软件对 LTRCn 寄存器中 FTS 位的设置（空闲）。
(2)	LTRCn 寄存器中的 FTS 位设置为 1（开始帧传输或唤醒传输/接收）。	传送间隔。
(3)	等待中断请求。	传送间隔分隔符。
(4)		传送同步场（55H）。
(5)		传送字节间隔（帧头）。
(6)		传送 ID 场。
(7)		设置成功的帧头传输标志。

有关错误检测的信息，请参考20.4.6 错误状态。

(b) 响应传输

图20-6是LIN/UART模块（LIN主机模式）在响应传输中的操作。

表20-8是响应传输中的处理。

图20-6.响应传输中的操作

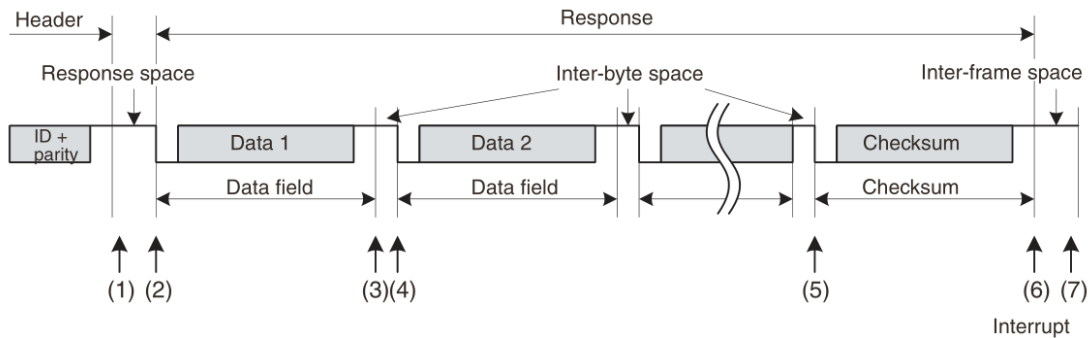


表20-8.响应传输中的处理

步骤	软件处理	LIN/UART 模块处理
(1)	(在帧分离模式下) • LTRCn 寄存器中的 RTS 位设置为 1(响应传输/接收开始)。 (不在帧分离模式下) • 等待中断请求。	(当在帧分离模式下) • 等待软件将 LTRCn 寄存器中的 RTS 位设置为 1。 • 当该位设置为 1 时, 发送响应间隔。(当不在帧分离模式时) • 发送响应间隔。
(2)	等待中断请求。	传送数据 1。
(3)		传送单个字节间隔。
(4)		<ul style="list-style-type: none"> <li>• 传送数据 2。</li> <li>• 传送单个字节间隔</li> <li>• 传送数据 3。</li> <li>• 传送单个字节间隔</li> </ul> (重复传输字节间隔, 次数与 LDFCn 寄存器中 RFDL[3:0]位指定的数据长度相同。当 LESTn 寄存器中 BER 标志为 1 (检测到位错误) 时, 停止传输。如果发生错误, 不执行第 (5) 项中的校验和传输。)
(5)		传递校验和。
(6)		<ul style="list-style-type: none"> <li>• 设置成功的帧/唤醒传输标志。</li> <li>• LTRCn 寄存器中的 FTS 位设置为 0 (帧传输或唤醒传输/接收停止)。(在帧分离模式下)</li> <li>• LTRCn 寄存器中的 RTS 位设置为 0 (响应传输/接收停止)。</li> </ul>
(7)	<ul style="list-style-type: none"> <li>• 通信后的处理</li> <li>• 检查 LSTn 寄存器并清除标志。</li> </ul>	空闲

有关错误检测的信息, 请参考20.4.6 错误状态。

(c) 响应接收

图20-7是LIN/UART模块（LIN主机模式）在响应接收中的操作。表20-9提供了响应接收中的处理。

图20-7.响应接收中的操作

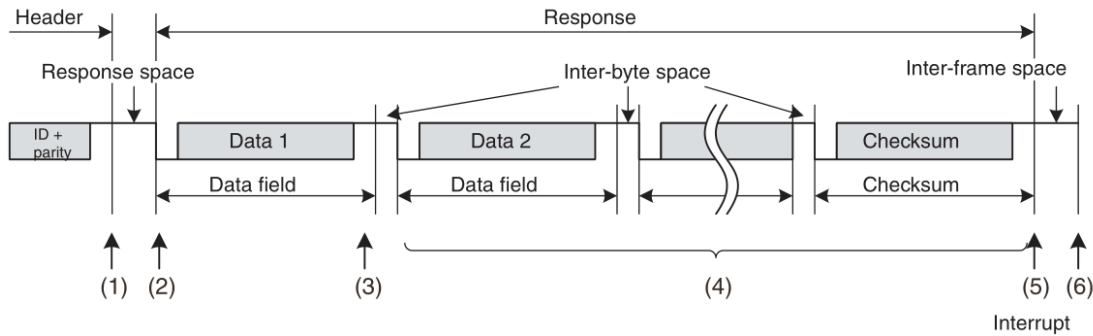


表20-9.响应接收中的处理

步骤	软件处理	LIN/UART 模块处理
(1)		等待检测到起始位。
(2)		检测到起始位时，接收数据 1。
(3)		设置成功的数据 1 的接收标志。
(4)	等待中断请求（不处理）。	<ul style="list-style-type: none"> <li>检测到起始位时，接收数据 2。</li> <li>检测到起始位时，接收数据 3。</li> </ul> (重复传输字节间隔，其次数与 LDFCn 寄存器中 RFDL[3:0]位指定的数据长度相同，当 LESTn 寄存器中的任何位为 1 时（检测到位错误），停止传输。如果发生错误，则不执行第（5）项中的校验和判断）。
(5)		<ul style="list-style-type: none"> <li>检测到起始位时，接收校验和。</li> <li>确定校验和。</li> <li>设置成功的帧/唤醒接收标志。</li> <li>将 LTRCn 寄存器中的 FTS 位设置为 0（帧传输或唤醒传输/接收停止）。</li> </ul>
(6)	<ul style="list-style-type: none"> <li>通信后处理</li> <li>读取收到的数据。</li> <li>检查LSTn寄存器并清除标志。</li> </ul>	空闲

有关错误检测的信息，请参考20.4.6 错误状态。



### 20.4.1.2 LIN从机模式

#### (a) 帧头接收

图20-8是LIN/UART模块（LIN从机模式）在帧头接收中的操作。表20-10是帧头接收中的处理。

图20-8.帧头接收中的操作

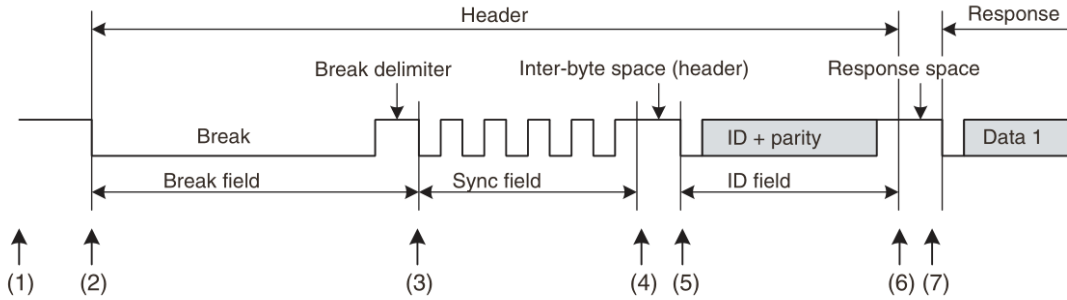


表20-10.帧头接收中的处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>•设置波特率。</li> <li>•设置噪声滤波器的开/关。</li> <li>•允许中断。</li> <li>•允许错误检测。</li> <li>•设置帧配置参数。</li> <li>•LIN/UART 模块改为 LIN 从机模式：LIN 工作模式。</li> </ul>	等待软件对 LTRCn 寄存器中 FTS 位的设置。
(2)	等待中断请求。	等待检测到间隔场。
(3)		检测间隔场（在 LIN 从机模式[固定波特率]下；在 LIN 从机模式[自动波特率]下，关于间隔场的检测时间，见（1）自动波特率校正功能）。
(4)		<ul style="list-style-type: none"> <li>•检测同步场（55H）</li> <li>•设置波特率生成器（在 LIN 从机模式下[自动波特率]）。</li> <li>•清除无响应请求位（LNRR 位）。</li> </ul>
(5)		•接收 ID 场。
(6)		<ul style="list-style-type: none"> <li>•检查 ID 奇偶校验位</li> <li>设置帧头接收完成标志。</li> </ul>
(7)	<ul style="list-style-type: none"> <li>•检查 LSTn 寄存器并清除标志。</li> <li>•检查 LIDBn 寄存器并准备响应。</li> </ul>	<ul style="list-style-type: none"> <li>•完成帧头的接收过程。</li> <li>•等待响应请求。</li> </ul>

n=0

LIN/UART模块可以在传输或接收帧的过程中接收间隔场。在此处，可能在检测到帧错误、位错误等时产生接收状态中断，位置是在接收间隔场之前帧的停止位。

有关错误检测的信息，请参考20.4.6 错误状态。

(1)自动波特率校正功能

在LIN从机模式[自动波特率]下，系统经常测量收到的低电平宽度。如果第一个“低电平”宽度是10倍（如果LBFCn寄存器中的BLT位为"0"）或11倍（如果LBFCn寄存器中的BLT位为"1"）或比同步场的起始2位（从同步场开始的连续下降沿的周期）的平均值计算出来更大的宽度，系统即得出结论，间隔场检测成功，并确认同步场的数据为55H。当确认数据为55H且同步场的接收成功后，系统自动在LBRPn1和LBRPn0寄存器中设置波特率校正结果。

如果ID场接收到数据且没有错误，在停止位位置产生一个成功的帧头接收中断。

如果同步场数据不是55H，系统就会断定同步场的检测失败，并设置一个同步场错误标志，产生一个错误检测中断。

在这种情况下，LIN/UART模块等待检测到另一个间隔场（"低"）而不进行波特率校正。

图20-9.LIN从机模式下的帧头接收[自动波特率]（正常运行）

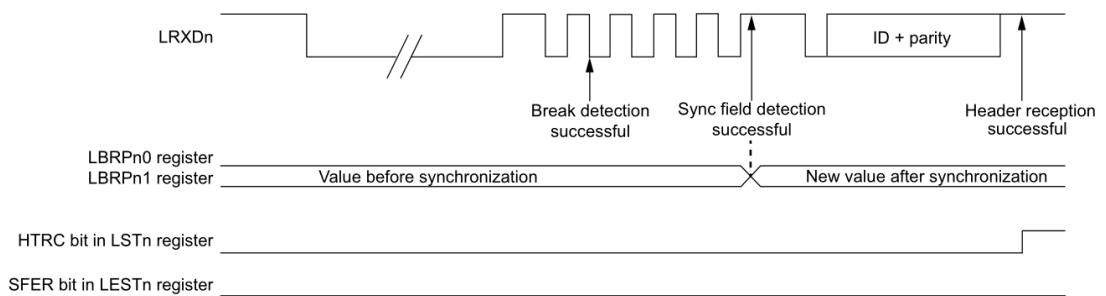
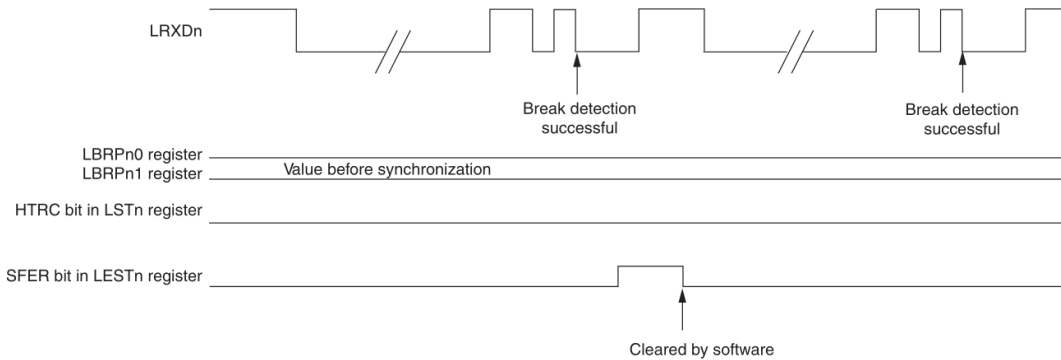


图20-10.LIN从机模式下的帧头接收[自动波特率]（同步场错误）



(b) 响应传输

图20-11是LIN/UART模块（在LIN从机模式下）在响应传输中的操作。表20-11是响应传输中的处理。

图20-11.响应传输中的操作

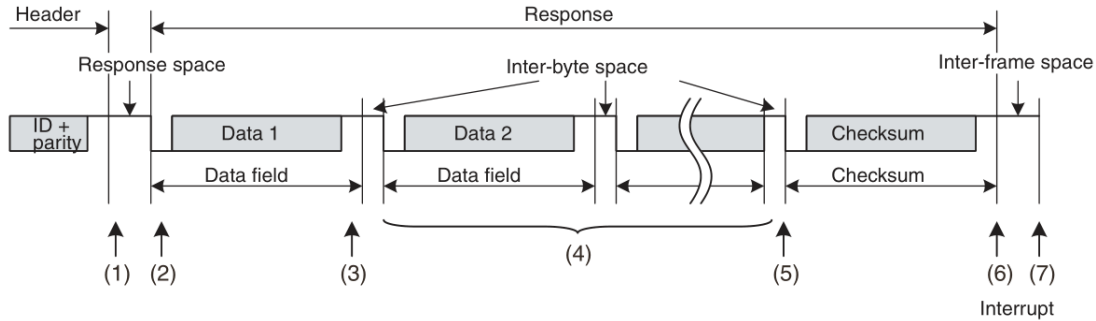


表20-11.响应传输中的处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>•设置 LDFCn 寄存器。</li> <li>•设置 LDBnm 寄存器。</li> <li>•将 LTRCn 寄存器中的 RTS 位设置为 1（响应传输/接收开始）。</li> </ul>	<ul style="list-style-type: none"> <li>•等待软件对 LTRCn 寄存器中 RTS 位或 LNRR 位的设置。</li> <li>•LTRCn 寄存器中的 RTS 位设置为 1 后，发送响应间隔。</li> </ul>
(2)	等待中断请求。	传送数据 1。
(3)		传送字节间隔。
(4)		<ul style="list-style-type: none"> <li>•传送数据 2。</li> <li>•传送单个字节间隔</li> <li>•传送数据 3。</li> <li>•传递单个字节间隔 (重复传输字节间隔, 次数与 LDFCn 寄存器中 RFDL[3:0]位指定的数据长度相同, 当 LESTn 寄存器中 BER 位为 1 (检测到位错误) 时, 停止传输。如果发生错误, 则不执行第 (5) 项中的校验和传输)。</li> </ul>
(5)		传递校验和。
(6)		<ul style="list-style-type: none"> <li>•设置一个成功的帧/唤醒传输标志或一个错误标志。</li> <li>•将 LTRCn 寄存器中的 RTS 位设置为 0 (停止响应传输/接收)。</li> </ul>
(7)	<ul style="list-style-type: none"> <li>•通信后的处理</li> <li>检查 LSTn 寄存器并清除标志。</li> </ul>	<ul style="list-style-type: none"> <li>•完成响应传输过程。</li> <li>•等待一个新的间隔。</li> </ul>

LIN/UART模块可以在传输或接收单个帧的过程中接收一个间隔场。在此处检测到帧错误、位错误等时，且在间隔场接收前帧的停止位上可能会产生一个接收状态中断。

有关错误检测的信息，请参考20.4.6 错误状态。

(c) 响应接收

图20-12是LIN/UART模块（LIN从机模式）在响应接收中的操作。表20-12是响应接收中的处理。

图20-12.响应接收中的操作

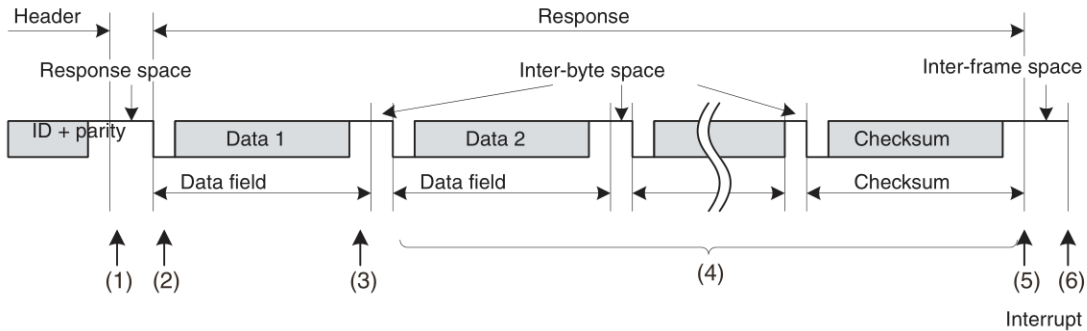


表20-12.响应接收中的处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>设置 LDFCn 寄存器。</li> <li>将 LTRCn 寄存器中的 RTS 位设置为 1（响应传输/接收开始）。</li> </ul>	<ul style="list-style-type: none"> <li>等待软件对 LTRCn 寄存器中 RTS 位或 LNRR 位的设置。</li> <li>等待起始位检测。</li> </ul>
(2)	等待中断请求。	检测到起始位时，接收数据 1。
(3)		设置成功接收数据 1 的标志。
(4)		<ul style="list-style-type: none"> <li>检测到起始位时，接收数据 2。</li> <li>检测到起始位时，接收数据 3。</li> <li>重复传输字节间隔，其次数与 LDFCn 寄存器中 RFDL[3:0]位指定的数据长度相同，当 LESTn 寄存器中的任何位为 1 时（检测到位错误），停止传输。如果发生错误，则不执行第（5）项中的校验和判断）。</li> <li>检测到起始位时，接收校验和。</li> </ul>
(5)		<ul style="list-style-type: none"> <li>确定校验和。</li> <li>设置一个成功的帧/唤醒接收标志或一个错误标志。</li> <li>将 LTRCn 寄存器中的 RTS 位设置为 0（停止响应传输/接收）。</li> </ul>
(6)	<ul style="list-style-type: none"> <li>通信后处理</li> <li>读取收到的数据。</li> <li>检查LSTn寄存器并清除标志。</li> </ul>	<ul style="list-style-type: none"> <li>完成响应过程。</li> <li>等待一个新的间隔。</li> </ul>

LIN/UART模块可以在传输或接收帧的过程中接收一个间隔场。在此处，当检测到帧错误、位错误等时，在间隔场接收前的帧的停止位上可能会产生一个接收状态中断。

有关错误检测的信息，请参考20.4.6 错误状态。

(d) 无响应请求

图20-13是在没有响应请求时LIN/UART模块（LIN从机模式）的操作。表20-13是无响应请求时的处理。

图20-13.无响应请求时的操作

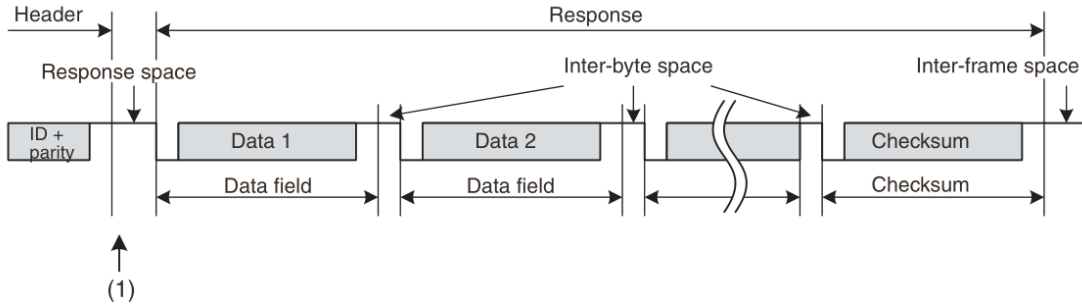


表20-13.无响应请求时的处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>•将无响应请求位（LNRR 位）设为 1。</li> </ul>	<ul style="list-style-type: none"> <li>•等待软件对无响应请求位（LNRR 位）的设置。</li> <li>•完成帧接收过程。</li> <li>•等待一个新的间隔。</li> </ul>

## 20.4.2 数据传输/接收

### 20.4.2.1 数据传输

每1Tbit传输1位数据。

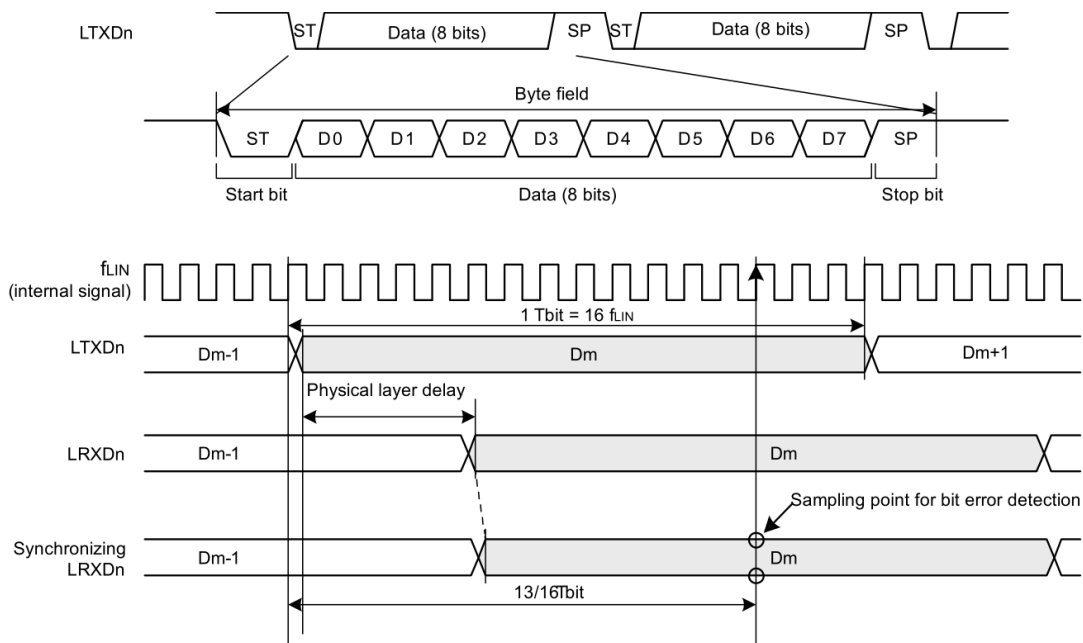
传输的数据通过LIN收发器返回到接收数据输入引脚。逐位比较接收的数据和传输的数据，把结果存储在LESTn寄存器中的位错误标志中（见20.4.6 错误状态。）。

在LIN主机模式和LIN从机模式[固定波特率]下，1Tbit生成16f<sub>LIN</sub>，因此接收数据的采样点在第13个时钟周期（81.25%的位置）。

在LIN从机模式[自动波特率]下，如果1Tbit生成成为4f<sub>LIN</sub>，则接收数据的采样点在第三个时钟周期（75%的位置）。如果1Tbit生成成为8f<sub>LIN</sub>，则接收数据的采样点在第七个时钟周期（87.5%的位置）。

图20-14举例单个数据传输的时序。

图20-14.数据传输时序示例（LIN主机模式，LIN从机模式[固定波特率]）



### 20.4.2.2 数据接收

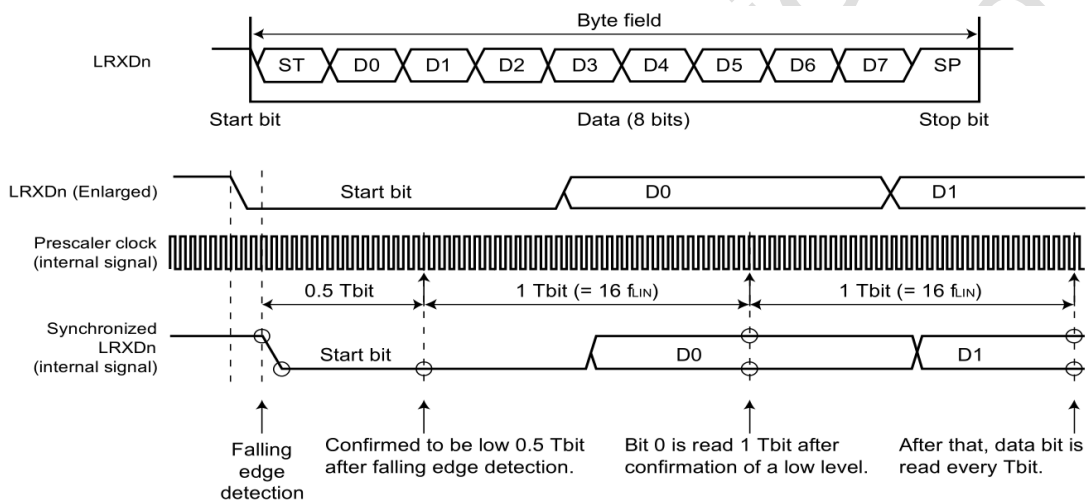
数据接收是通过使用同步的LRXDn信号（一个内部信号）进行，该信号是来自LRXDn引脚的输入与预分频器时钟同步。

字节场在同步LRXDn信号的起始位的下降沿进行同步。在检测到下降沿后，0.5Tbit后再次进行采样，如果同步LRXDn信号为低电平，则下降沿会识别为起始位。如果清除复位后的LRXDn信号是低固定的，或者在重新采样时检测到高电平，则下降沿不会识别为起始位。

在检测到起始位后，系统每Tbit采样1位。LIN/UART模块在接收数据方面有一个噪声滤波器功能。如果LMDn寄存器中的LRDNFS位为0，则LIN/UART模块使用噪声滤波器。对于采样值，则使用由预分频器时钟上的3次采样根据多数决定原则(3-samplingmajorityrule)来确定值。如果LMDn寄存器中的LRDNFS位为1，则LIN/UART模块不使用噪声滤波器，对于采样值来说，采样位同步LRXDn的值按原样使用。

图20-15举例数据接收的时序。

图20-15.数据接收时序示例（LIN主机模式，LIN从机模式[固定波特率]



## 20.4.3 传输/接收数据缓冲

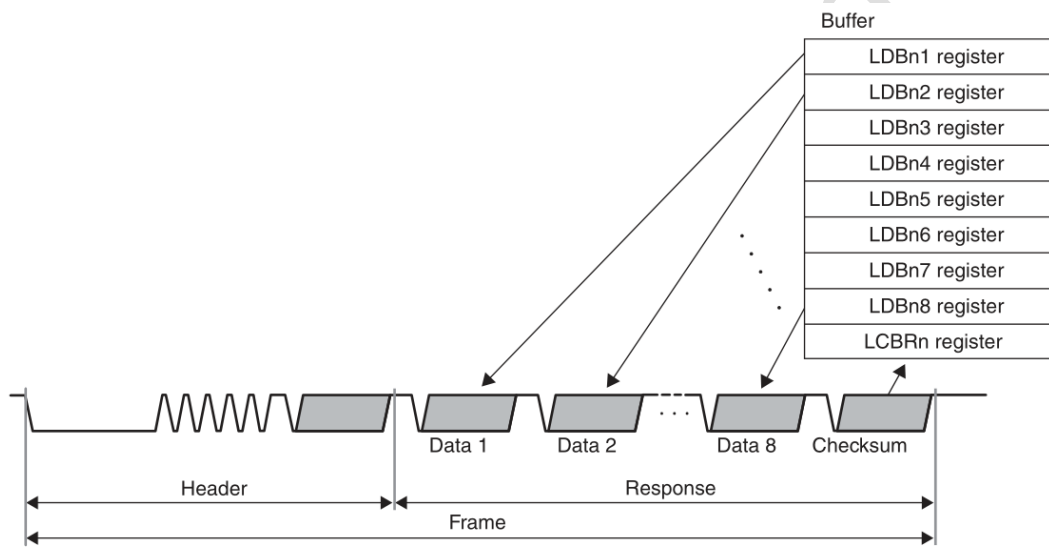
本节说明在LIN/UART模块连续发送或接收数据时的缓冲处理。

### 20.4.3.1 LIN帧的传输

对于8字节的传输，存储在寄存器LDBn1至LDBn8中的内容依次传输到LIN帧的数据区1至8。在4字节传输的情况下，存储在寄存器LDBn1至LDBn4中的内容传输到LIN帧的数据区1至4，但寄存器LDBn5至LDBn8的内容不传输。传输的校验和数据存储在LCBRn寄存器中。

图20-16描述了LIN传输处理和所需的缓冲。

图20-16.LIN传输处理和所需的缓冲



#### (a) 帧分离模式

将LDFCn寄存器中的FSM位设置为1，开启帧分离模式。

在帧分离模式下，在单独的传输开始请求提示后，会传输一个帧头和一个响应。当帧头的传输完成后，LSTn寄存器中的HTRC标志变为1（帧头传输成功）。

在LIN主机模式下发送或接收9字节或更多的响应数据时，使用帧分离模式。

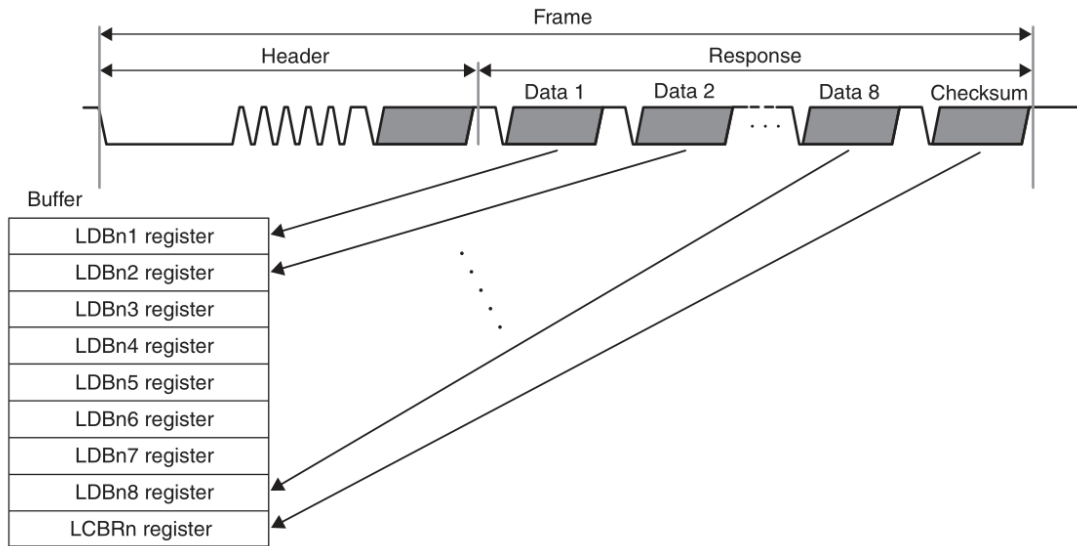


### 20.4.3.2 接收LIN帧

对于一个8字节的接收，在收到一个停止位后，LIN帧的数据区域1到8的内容将分别存储在寄存器LDBn1到LDBn8中。而对于4字节接收，LIN帧的数据区域1至4的内容分别存储在寄存器LDBn1至LDBn4中；但是，在寄存器LDBn5至LDBn8中不存储数据。另外，接收到的校验和数据存储在LCBRn寄存器中。

图20-17描述了LIN接收处理和所需的缓冲区。

图20-17.LIN接收处理和所需的缓冲区



#### (a)接收数据1

当第一个字节的数据接收完成后，LSTn寄存器中的D1RC标志变为1（数据1接收成功）。

### 20.4.3.3 多字节响应传输/接收功能

在LIN通信中，响应一般是9个字节或更少字节（包括一个校验和场）；但是，也可以发送和接收10个字节或更多的响应。在这种情况下，允许位错误、帧错误、响应准备错误检测和自动校验功能。

如果数据长度大于8字节，在发送或接收数据组之前，应将第一个数据组（0至8字节）中的LSS位设置为1（这表示要发送或接收的下一个数据组不是最后一组）。发送或接收后，用户应确定下一个数据组是否为最后一组。如果是最后一组，LSS位应设置为0（表示要发送或接收的下一个数据组是最后一组），并在最后一组数据上附加一个校验和。

在RTS位为0时通过改变RFDL位的设置，用户可以改变每个数据组的数据长度。

在LIN主机模式下执行多字节响应传输/接收时，将LDFCn寄存器中的FSM位设置为1（帧分离模式）。

**注意事项** 在LIN从机模式下，LIN/UART模块可以在传输或接收响应过程中检测到一个新的间隔场。

## 20.4.4 唤醒传输/接收

唤醒传输/接收可以在LIN唤醒模式下使用。

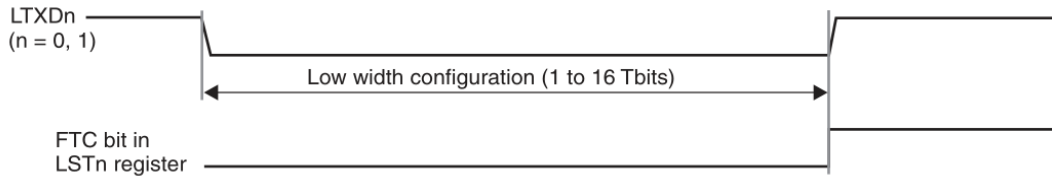
### 20.4.4.1 唤醒传输

在LIN唤醒模式下，将LDFCn寄存器中的RCDS位设置为1（传输），将LTRCn寄存器中的FTS位设置为1（帧头接收或唤醒传输/接收开始），会从输出引脚输出一个唤醒信号。唤醒信号的低电平宽度应该用LWUPn寄存器中的WUTL[3:0]位来设置。然而，当LWBRn寄存器中的LWBR0位在LIN主机模式下为1时，无论LMDn寄存器中的LCKS位如何设置，低电平宽度都是基于fa作为LIN系统时钟（fLIN）来定义的。在选择fa的情况下，将波特率设置为19200bps且LWUPn寄存器中的WUTL[3:0]位设置为0100b(5Tbits)，允许在LIN唤醒模式下输出260 μs的低电平宽度信号，而不管LMDn寄存器中LCKS位的设置。

如果在没有任何位错误的情况下输出唤醒低电平，LSTn寄存器中的FTC标志变为1（成功响应或唤醒传输）；当LIEn寄存器中的FTCIE位为1（成功响应/唤醒传输中断允许），将产生一个中断请求。

如果检测到一个位错误，唤醒传输取消，LESTn寄存器中的BER标志设置为1（位错误检测）。图20-18是唤醒传输时序。

图20-18.唤醒传输时序



### 20.4.4.2 唤醒接收

唤醒信号的检测涉及到输入信号低宽计数功能的使用。

输入信号低宽计数功能可测量到LRXDn引脚输入信号的低宽，并使用与数据接收相同的采样点。该功能可以测量fLIN的2.5Tbits或更大的输入信号低宽。

在LIN主机模式下，适当设置LWBRn寄存器的LWBR0位可以在LIN工作模式和LIN唤醒模式之间切换，而不需要改变波特率发生器的设置。

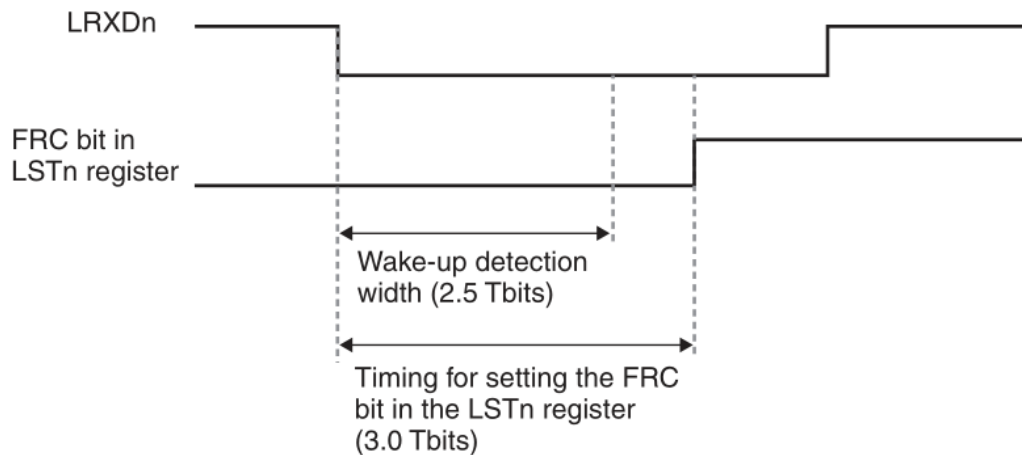
当使用1.3版LIN协议时，LWBRn寄存器中的LWBR0位设置为0。当使用2.x版LIN协议时，设置为1。当LWBR0位设置为1时，无论LMDn寄存器中LCKS位的设置如何，fa总是选为LIN系统时钟（fLIN）（LCKS位的设置不受影响）。

在选择fa的情况下，无论LMDn寄存器中LCKS位如何设置,波特率设置为19200bps，并允许检测130s或更长的输入信号的低电平宽度。

在LIN唤醒模式下使用该功能时，将LDFCn寄存器中的RFT位设为0（LIN主机模式：接收），或RCDS位设为0（LIN从机模式：接收），将LTRCn寄存器中的FTS位设为1（LIN主机模式：帧传输或唤醒传输/接收开始；LIN从机模式：帧头接收或唤醒传输/

接收开始）。当达到要测量的低宽度时，LSTn寄存器中的FRC标志变为1（成功响应/唤醒接收）。如果LIEn寄存器中的FRCIE位为1（成功响应或唤醒接收中断允许），就会产生一个中断请求。

图20-19.输入信号低计数功能



## 20.4.5 状态

在LIN模式运行期间，LIN/UART模块可以检测出七种状态。

帧/唤醒传输成功、帧/唤醒接收成功、错误检测、帧头传输/帧头接收成功，这四种状态可以产生中断请求。

表20-14列出了在LIN主机模式下的状态类型。表20-15列出了在LIN从机模式[自动波特率]和LIN从机模式[固定波特率]下的状态类型。

表20-14.LIN主机模式下的状态类型

状态	状态设置条件	状态清除条件	能够进行状态检测的操作模式	相应的位	中断
复位	如果实际上 LIN/UART 模块从 LIN 复位模式中清除，LCUCn 寄存器中的 OM0 位设置为非 LIN 复位模式后。	如果实际上 LIN/UART 模块进入了 LIN 复位模式，LCUCn 寄存器中的 OM0 位设置为 LIN 复位模式后。	所有模式	LMSTn 寄存器中的 OMM0 位	不适用
操作模式	如果实际上 LIN/UART 模块进入了 LIN 工作模式，LCUCn 寄存器中的 OM1 位设置为 LIN 工作模式后。	如果实际上 LIN/UART 模块进入了 LIN 唤醒模式，LCUCn 寄存器中的 OM1 位设置为 LIN 唤醒模式后。	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LMSTn 寄存器中的 OMM1 位注册	不适用
帧/唤醒传输结束	成功传输一个帧(帧头传输+响应传输)、一个唤醒信号或一个数据组。	<ul style="list-style-type: none"> <li>•当另一个通信开始时</li> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 FTC 标志	适用
帧/唤醒接收结束	成功收到一个帧(帧头传输+响应接收)、一个唤醒信号或一个数据组。	<ul style="list-style-type: none"> <li>•当另一个通信开始时</li> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 FRC 标志	适用
错误检测	如果 LESTn 寄存器中的 PRER 标志、CSER 标志、FER 标志、FTER 标志、PBER 标志和 BER 标志中的任何一个变为 1 (检测到错误)。	<ul style="list-style-type: none"> <li>•当另一个通信开始时</li> <li>•当软件清除的时候<sup>注1</sup></li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 ERR 标志	适用
数据 1 接收结束	LDFCn 寄存器中的 RFT 位为 0 (接收)，接收到响应场的第一个字节 <sup>注2</sup> 。	<ul style="list-style-type: none"> <li>•当另一个通信开始时</li> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	LIN 工作模式	LSTn 寄存器中的 D1RC 标志	不适用
帧头接收结束	当成功收到帧头场时。	<ul style="list-style-type: none"> <li>•当另一个通信开始时</li> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	LIN 工作模式	LSTn 寄存器中的 HTRC 标志	适用

注1.在LIN唤醒模式和LIN工作模式下，LSTn寄存器中的ERR标志通过向LESTn寄存器中的PRER标志、CSER标志、FER标志、FTER标志、PBER标志或BER标志写0而清零。

2. 当LDFCn寄存器中的RFDL[3:0]位为0000b (0字节+校验和) 时，不检测。

表20-15.LIN从机模式下的状态类型

状态	状态设置条件	状态清除条件	能够进行状态检测的操作模式	相应的位	中断
复位	如果实际上 LIN/UART 模块从 LIN 复位模式中清除, LCUCn 寄存器中的 OM0 位设置为非 LIN 复位模式后。	如果实际上 LIN/UART 模块进入了 LIN 复位模式, LCUCn 寄存器中的 OM0 位设置为 LIN 复位模式后,。	所有模式	LMSTn 寄存器中的 OMM0 位	不适用
操作模式	如果实际上 LIN/UART 模块进入了 LIN 工作模式, LCUCn 寄存器中的 OM1 位设置为 LIN 工作模式后。	如果实际上 LIN/UART 模块进入了 LIN 唤醒模式, LCUCn 寄存器中的 OM1 位设置为 LIN 唤醒模式后。	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LMSTn 寄存器中的 OMM1 位	不适用
帧/唤醒传输结束	成功传输一个响应场、一个唤醒信号或一个数据组。	<ul style="list-style-type: none"> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 FTC 标志	适用
帧/唤醒接收结束	成功收到一个响应场、一个唤醒信号或一个数据组。	<ul style="list-style-type: none"> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 FRC 标志	适用
错误检测	如果 LESTn 寄存器中的 PRER 标志、IPER 标志、CSER 标志、SFER 标志、FER 标志、TER 标志和 BER 标志中的任何一个变为 1(检测到错误)。	<ul style="list-style-type: none"> <li>•当软件清除的时候<sup>注1</sup></li> <li>•转换到 LIN 复位模式后</li> </ul>	<ul style="list-style-type: none"> <li>•LIN 工作模式</li> <li>•LIN 唤醒模式</li> </ul>	LSTn 寄存器中的 ERR 标志	适用
数据 1 接收结束	LDFCn 寄存器中的 RCDS 位为 0(接收), 收到响应场的第一个字节 <sup>注2</sup> 。	<ul style="list-style-type: none"> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	LIN 工作模式	LSTn 寄存器中的 D1RC 标志	不适用
帧头接收结束	当成功收到帧头场时。	<ul style="list-style-type: none"> <li>•当软件清除的时候</li> <li>•转换到 LIN 复位模式后</li> </ul>	LIN 工作模式	LSTn 寄存器中的 HTRC 标志	适用

注1.在LIN唤醒模式和LIN工作模式下, LSTn寄存器中的ERR标志通过向LESTn寄存器中的PRER标志、IPER标志、CSER标志、SFER标志、FER标志、TER标志或BER标志写0而清零。

2. 当LDFCn寄存器中的RFDL[3:0]位为0000b(0字节+校验和)时, 不检测。

## 20.4.6 错误状态

### 20.4.6.1 LIN主机模式

#### (a) 错误状态的类型

LIN/UART模块可以在LIN主机模式下检测到六种错误状态。这些错误状态的状态可以通过LESTn寄存器中的相应的位来检查。

所有的错误状态都代表中断事件。表20-16列出了错误状态的类型。

表20-16.LIN主机模式下的错误状态类型

状态	错误检测条件	能够检测错误的操作模式	通信	允许/禁止检测	相应的位
位错误	传输的数据和接收引脚监测到LIN总线上的数据不匹配 <sup>注1,2</sup> 。	•LIN 工作模式 •LIN 唤醒模式	取消	○	LESTn 寄存器中的 BER 标志
物理总线错误	•发送中断时, LIN 总线检测为高电平 •发送断裂分隔符时, LIN 总线检测为低电平 •当唤醒信号发送时, LIN 总线检测到为高电平。	•LIN 工作模式 •LIN 唤醒模式	取消	○	LESTn 寄存器中的 PBER 标志
超时错误	帧或响应传输/接收没有在给定的时间内终止 <sup>注3</sup>	LIN 工作模式	取消	○	LESTn 寄存器中的 FTER 标志
帧错误	在响应场接收中, 每个数据字节的停止位为低电平。	LIN 工作模式	取消	○	LESTn 寄存器中的 FER 标志
校验和错误	在响应场接收中, 校验测试的结果出现错误	LIN 工作模式	-	×	LESTn 寄存器中的 CSER 标志
响应准备错误	在帧分离模式下会出现以下情况: •收到的第一个接收数据字节是在帧头传输完成后, 但在响应传输/接收请求设置之前。 •收到第一个接收数据字节在前一个数据组的完成接收后, 但在另一个数据组的发送/接收请求设置之前	LIN 工作模式	取消	×	LESTn 寄存器中的 RPER 标志

**注1.**如果检测到位错误, 在发送一个停止位后, 该过程取消。如果在非数据区域, 如字节间隔检测到位错误, 则在发送错误位后立即取消传输。如果在发送唤醒信号的过程中检测到一个位错误, 则在致错位发送之后取消唤醒信号的传送。

**2.**在多字节响应传输中, 数据组之间可以检测到位错误。

**3.**超时时间取决于响应场的长度 (LDFCn寄存器中的RFDL[3:0]位) 和校验和的选择 (LDFCn寄存器中的CSM位), 可以根据以下公式计算:

超时时间保持为8个数据字节, 直到在帧分离模式下设置了寄存器LTRCn (LDFCn寄存器的FSM位设置为1) 发生改变。

## [帧超时]

- 经典型模式（当LDFCn寄存器的CSM位为0时）。  
超时时间=49+（数据字节数+1）×14[Tbit]。
- 增强型模式（当LDFCn寄存器中的CSM位为1时）。  
超时时间=48+（数据字节数+1）×14[Tbit]

上述的超时时间是大于 LIN 协议 1.3 版经典型规定的或 2.x 版增强型规定的 TFRAME\_MAX 时间。

## [响应超时]

超时时间=（数据字节数+1）×14[Tbit]

**注意事项** 当另一个通信开始时，或在软件清除时，或转换到LIN复位模式后，错误状态清除。

Draft Version  
Cmsemicon Confidential

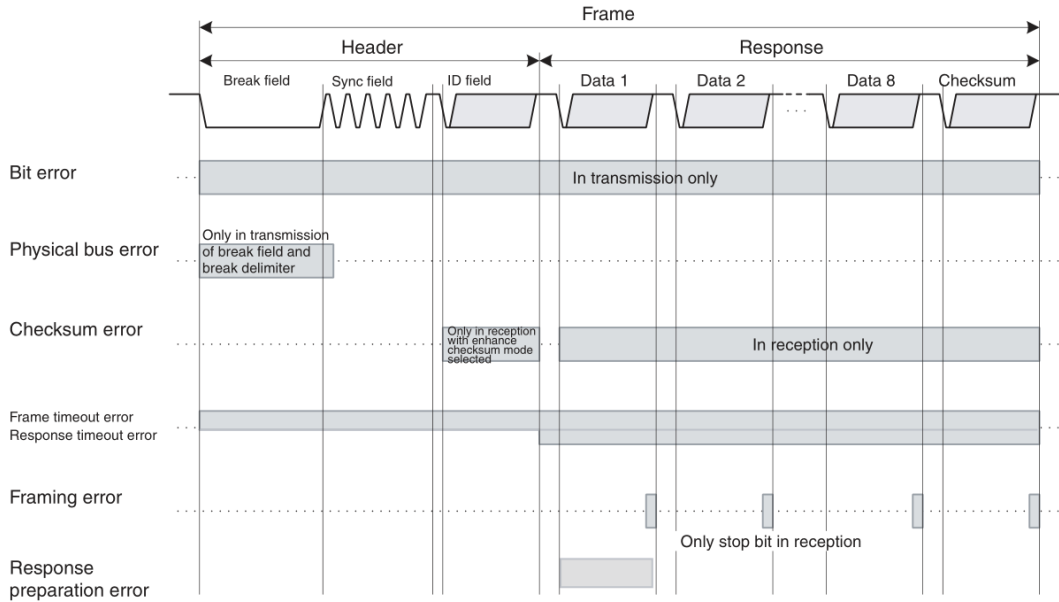


(b) LIN错误检测的目标时间区域

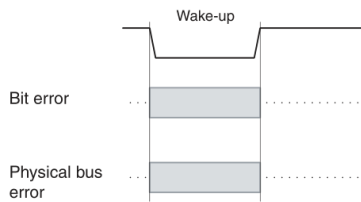
图20-20是主机模式下LIN/UART模块执行错误检测监控的时域。

图20-20.LIN错误检测的目标时间区域（LIN主机模式）

<帧发送/接收>



<唤醒发送>



## 20.4.6.2 LIN从机模式

### (c) 错误状态的类型

LIN/UART模块可以在LIN从机模式[自动波特率]或LIN从机模式[固定波特率]下检测出七种错误状态。这些错误状态可以通过检查LESTn寄存器中的相应的位进行验证。

表20-17展示了错误状态的类型。

表20-17.LIN从机模式下的错误状态类型

状态	错误检测条件	能够检测错误的操作模式	通信	允许/禁止检测	相应的位
位错误	传输的数据和接收引脚监控的 LIN 总线上的数据不匹配 <sup>注1, 2</sup>	•LIN 工作模式 •LIN 唤醒模式	取消	○	LESTn 寄存器中的 BER 标志
超时错误	帧或响应传输/接收没有在给定的时间内终止 <sup>注3</sup>	LIN 工作模式	取消	○	LESTn 寄存器中的 TER 标志
帧错误	在帧接收中, 每个数据字节的停止位为低电平。	LIN 工作模式	取消	○	LESTn 寄存器中的 FER 标志
同步场错误	如果中断低电平的宽度大于 LBFCn 寄存器中 BLT 位设置的宽度, 并且同步场不是 55H	LIN 工作模式	取消	○ <sup>注4</sup>	LESTn 寄存器中的 SFER 标志
校验和错误	在响应帧接收中, 校验和测试的结果检出错误。	LIN 工作模式	- <sup>注5</sup>	×	LESTn 寄存器中的 CSER 标志
ID 奇偶性错误	如果收到的 ID 奇偶校验位与 LIN/UART 模块自动计算的值不一	LIN 工作模式	取消	○	LESTn 寄存器中的 IPER 标志
响应准备错误	•在接收到帧头后, 在收到第一个接收数据字节之前, 没有及时进行响应准备。 •在一个多字节传输/接收响应过程中, 在收到另一个数据组的第一个接收数据字节之前, 没有及时为另一个数据组做准备。	LIN 工作模式	取消	×	LESTn 寄存器中的 RPER 标志

注1. 如果检测到一个位错误, 在发送一个停止位后, 该过程会取消。如果在非数据区域, 如字节间隔检测到位错误, 则在发送错误位后立即取消传输。如果在发送唤醒信号的过程中检测到一个位错误, 则在致错位发送之后取消唤醒信号的传送。

2. 在多字节响应传输中, 数据组之间可以检测到位错误

3. 超时时间取决于响应场的数据长度 (LDFCn寄存器中的RFDL[3:0]位) 和校验和选择 (LDFCn寄存器中的LCS位)。检查和选择 (LDFCn寄存器中的LCS位), 这可以根据以下公式来计算以下公式计算:

在LTRCn寄存器中的RTS或LNRR位设置之前, 超时时间是根据8个字节数据来设置。一旦设置RTS位, 超时时间将根据响应场数据长度 (LDFCn寄存器中RFDL[3:0]位) 重新设置。设置LNRR位时, 超时功能停止。

[帧超时]

- 经典模式 (当LDFCn寄存器中CSM位为0时)  
超时时间=49+ (数据字节数+1) ×14[Tbit]
- 增强型模式(当LDFCn寄存器中的CSM位为1时)  
超时时间=48+ (数据字节数+1) ×14[Tbit]

上述的超时时间是大于LIN协议1.3版经典型规定的或2.x版增强型规定的TFRAME\_MAX时间

[响应超时]

超时时间= (数据字节数+1) ×14[Tbit]

4. 只有SFER标志中的指示可以允许或禁止; 错误检测没有允许或禁止的功能。

5.校验和验证是在响应帧接收完成后进行的。如果结果确认有误，成功的接收标志不设置为1。

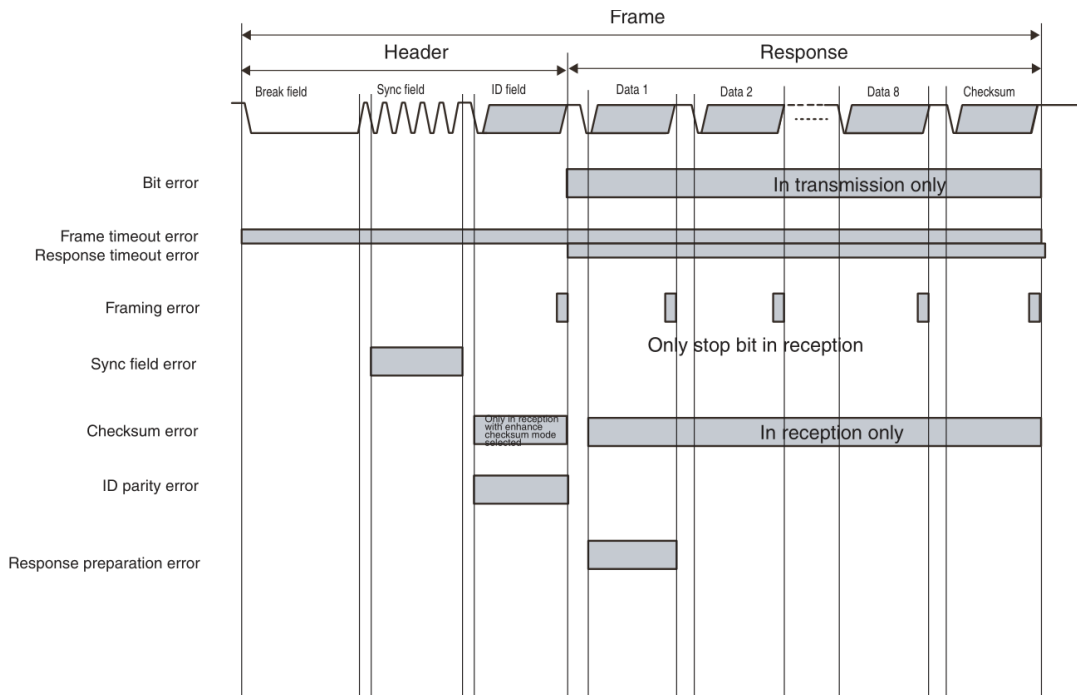
**注意事项** 错误状态的清除在软件清除或转换到LIN复位模式后。

(d) LIN错误检测的目标时间区域

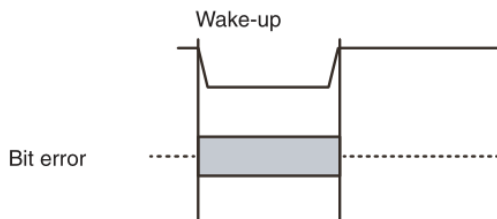
图20-21是在从机模式下LIN/UART模块执行错误检测监控的时域。

图20-21.LIN错误检测的目标时间区域（LIN从机模式）

<帧发送/接收>



<唤醒发送>



## 20.5 UART模式

### 20.5.1 操作概述

#### 20.5.1.1 传输

图20-22是LIN/UART模块（UART模式下）的传输操作；表20-18是LIN/UART模块（UART模式下）的传输处理。

图20-22.LIN/UART模块（在UART模式下）传输操作

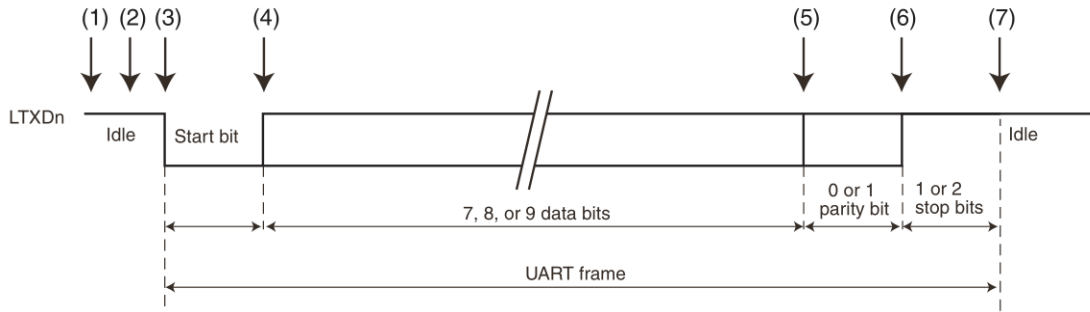


表20-18.LIN/UART模块（UART模式）传输处理

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>设置波特率。</li> <li>设置噪声滤波器的开/关。</li> <li>设置错误检测允许。</li> <li>设置数据格式</li> <li>设置中断产生的时序。</li> <li>LIN/UART 模块从 LIN 复位模式中清除</li> <li>发送允许位 (UTOE 位) 设为 1</li> </ul>	<ul style="list-style-type: none"> <li>等待软件的传输触发 (LUTDRn 寄存器)。</li> </ul>
(2)	<ul style="list-style-type: none"> <li>设置 UART 传输数据寄存器 (LUTDRn 寄存器) 或 UART 等待传输数据 (LUWTDn) 中的传输数据。</li> </ul>	<ul style="list-style-type: none"> <li>设置传输状态标志。</li> </ul>
(3)	<ul style="list-style-type: none"> <li>等待中断请求。</li> </ul> <p>[当 UTIGTS 位为 0 时 (传输开始时输出一个传输中断请求)]</p> <ul style="list-style-type: none"> <li>当连续传输数据时, 在 UART 传输数据寄存器 (LUTDRn 寄存器) 中设置另一组传输数据, 等待中断请求的生成。</li> </ul>	<ul style="list-style-type: none"> <li>发送一个起始位 (为了在半双工通信中切换发送和接收, 收到 1 个停止位后发送一个起始位。此功能详细内容请参考 20.5.1.4 传输起始等待功能)。</li> <li>[当 UTIGTS 位为 0 时 (传输开始时输出一个传输中断请求)]。</li> <li>输出传输中断。</li> </ul>
(4)		发送 UART (等待) 传输数据寄存器中设置的数据。
(5)		使用奇偶校验时, 传送一个奇偶校验位。
(6)		传送 1 或 2 个停止位。
(7)	<p>[当 UTIGTS 位为 0 时 (传输开始时输出一个传输中断请求)]。</p> <ul style="list-style-type: none"> <li>如果设置了另一组传输数据, 则进入步骤 (3)。</li> </ul> <p>[当 UTIGTS 位为 1 时 (传输结束后输出一个传输中断请求)]。</p> <ul style="list-style-type: none"> <li>当连续传输数据时, 转到步骤 (2)。</li> </ul>	<p>[当 UTIGTS 位为 0 时 (传输开始时输出一个传输中断请求)]。</p> <ul style="list-style-type: none"> <li>如果设置了另一组传输数据, 则进入步骤 (3)。</li> <li>如果另一组传输数据没有设置, 则清除传输状态标志。</li> </ul> <p>[当 UTIGTS 位为 1 时 (传输结束后输出一个传输中断请求)]。</p> <ul style="list-style-type: none"> <li>输出传输中断。</li> <li>清除传输状态标志</li> </ul>

(a) 连续传输

LIN/UART模块（在UART模式下）可以通过使用LUTDRn寄存器连续传输多组数据。图20-23是一个操作实例，其中传输中断的生成点是传输的开始。图20-24展示了一个传输中断生成成为传输结束的操作实例。

图20-23.LIN/UART模块（UART模式）连续传输操作  
（当LUORn1寄存器的UTIGTS位为0时）

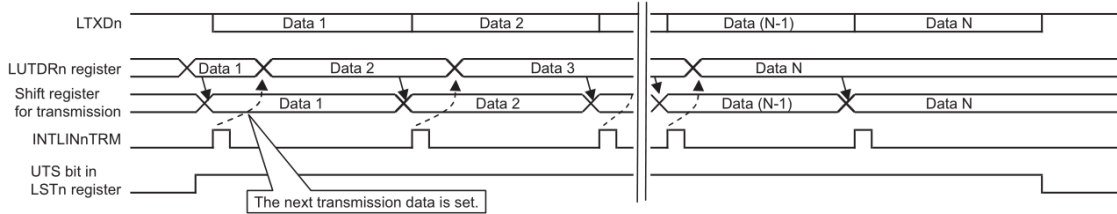
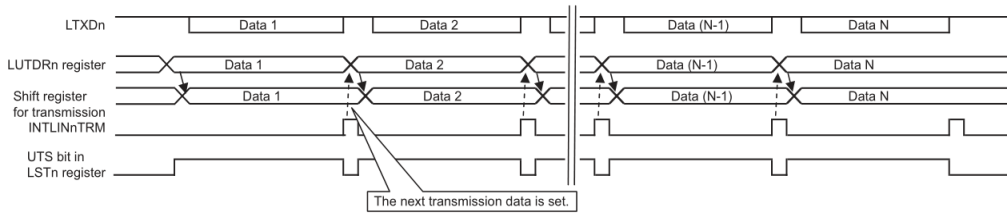


图20-24.LIN/UART模块（UART模式）连续传输操作  
（当LUORn1寄存器的UTIGTS位为1时）



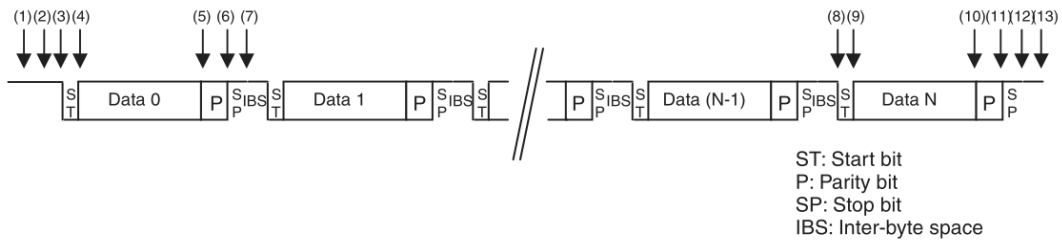
在最后一组数据开始传输后，通过将LUORn1寄存器中的UTIGTS位从0改为1，可以在传输结束时产生一个中断，前提是需要知道传输中断产生的时间是在传输的开始，并在最后一组数据的传输结束点。

(b) UART缓冲传输

LIN/UART模块（在UART模式下）拥有最多9个字节的UART缓冲区，因此它能够通过使用UART缓冲区进行连续传输。

图20-25展示了LIN/UART模块中的UART缓冲传输操作（在UART模式下）。表20-19是UART缓冲传输处理。

图20-25.LIN/UART模块中的UART缓冲传输（在UART模式下）



**表20-19.LIN/UART模块中的UART缓冲传输处理（在UART模式下）**

步骤	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>•设置波特率。</li> <li>•设置噪声滤波器的开/关。</li> <li>•设置允许错误检测。</li> <li>•设置数据格式。</li> <li>•设置在传输结束时中断产生的时序。</li> <li>•LIN/UART 模块从 LIN 复位模式中清除。</li> <li>•发送允许位（UTOE 位）设为 1</li> </ul>	<ul style="list-style-type: none"> <li>•等待软件的传输触发（RTS 位）。</li> </ul>
(2)	<ul style="list-style-type: none"> <li>•设置 UART 缓冲区的数据长度以及系统是否必须等待传输的开始。</li> <li>•设置 UART 数据缓冲区 0 寄存器（LUDBn0）和 LIN/UART 数据缓冲区 m 寄存器（LDBnm）中的传输数据。</li> </ul>	<ul style="list-style-type: none"> <li>•设置传输状态标志。</li> </ul>
(3)	<ul style="list-style-type: none"> <li>•等待中断请求。</li> </ul>	<ul style="list-style-type: none"> <li>•发送一个起始位（为了在半双工通信中切换发送和接收，收到 1 个停止位后发送一个起始位。此功能详细内容请参考 20.5.1.4 传输起始等待功能。</li> </ul>
(4)		传输 UART 数据缓冲 0 寄存器(LUDBn0)和 LIN/UART 数据缓冲区 m 寄存器(LDBnm)中设置的
(5)		使用奇偶校验时，传送一个奇偶校验位。
(6)		传输 1 或 2 个停止位
(7)		传送一个字节间隔（空闲）。
		重复步骤(3)至(7)，直到达到 UART 缓冲数据长度选择位中设置的帧计数-1。
(8)		传送一个起始位。
(9)		传送 LIN/UART 数据缓冲区 m 寄存器（LDBnm）中设置的数据。
(10)		使用奇偶校验时，传送一个奇偶校验位。
(11)		传送 1 或 2 个停止位。
(12)		<ul style="list-style-type: none"> <li>•设置缓冲区传输结束标志。</li> <li>•清除 UART 缓冲发送起始（RTS）位。</li> <li>•输出一个传输中断。</li> </ul>
(13)	<ul style="list-style-type: none"> <li>•检查 LSTn 寄存器并清除标志。</li> <li>•当连续传输数据时，转到步骤（2）。</li> </ul>	

n=0

### 20.5.1.2 接收

图20-26是LIN/UART模块（在UART模式下）的接收操作。表20-20是LIN/UART模块（在UART模式下）的接收处理。

图20-26.LIN/UART模块（UART模式下）接收操作

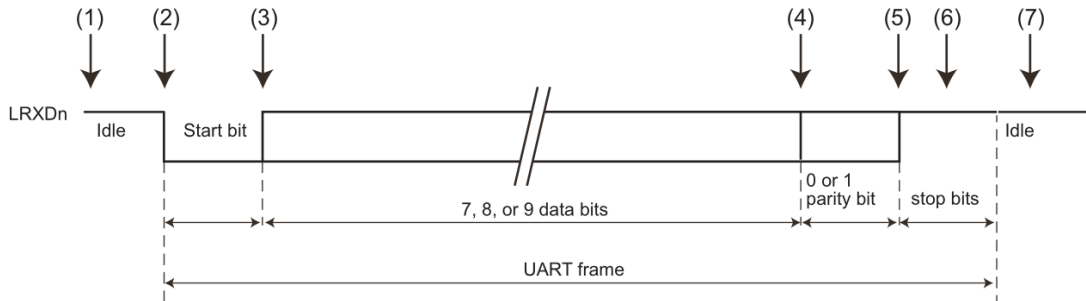


表20-20.LIN/UART模块（UART模式下）接收处理

Step	软件处理	LIN/UART 模块处理
(1)	<ul style="list-style-type: none"> <li>•设置波特率。</li> <li>•设置噪声滤波器的开/关。</li> <li>•设置允许错误检测。</li> <li>•设置数据格式。</li> <li>•LIN/UART 模块从 LIN 复位模式中清除。</li> <li>•接收允许位（UROE 位）设为 1。</li> </ul>	<ul style="list-style-type: none"> <li>•等待软件接收允许状态切换。</li> <li>•等待检测到一个起始位。</li> </ul>
(2)	等待一个中断请求。	<ul style="list-style-type: none"> <li>•等待来自接收引脚的下降沿，并检测一个起始位。</li> <li>•设置接收状态标志。</li> </ul>
(3)		接收数据。
(4)		使用奇偶校验时，接收一个奇偶校验位。
(5)		只接收 1 个停止位。
(6)		<ul style="list-style-type: none"> <li>•输出一个成功的接收中断。</li> <li>•清除接收状态标志。</li> </ul>
(7)	检查 LSTn 寄存器并清除标志。	等待接收引脚的下降沿。

n=0,



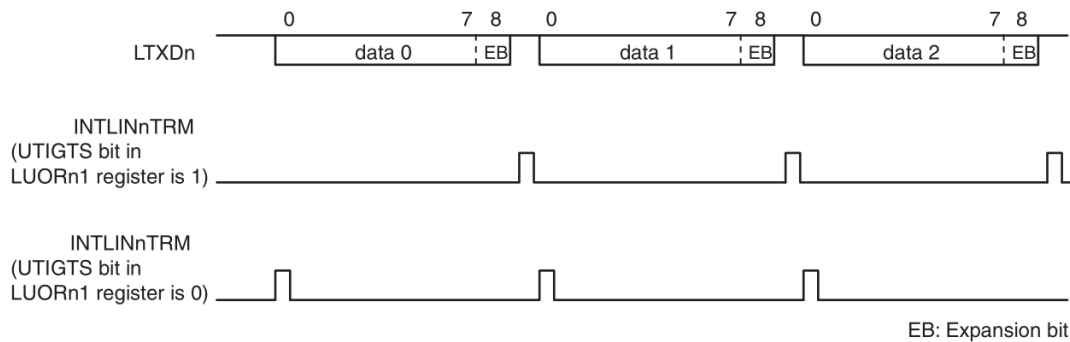
### 20.5.1.3 扩展位

LIN/UART模块（在UART模式下）通过将LUORn1寄存器中的UEBE位设置为1，可以发送和接收9位长的数据。

#### (a) 扩展位传输

当UART选项寄存器1（LUORn1）中的扩展位使能位（UEBE）为1时，通过将9位数据写入UART传输数据寄存器（LUTDRn）或UART等待传输数据寄存器（LUWTDRn），LIN/UART模块（在UART模式下）可以传输9位长数据。

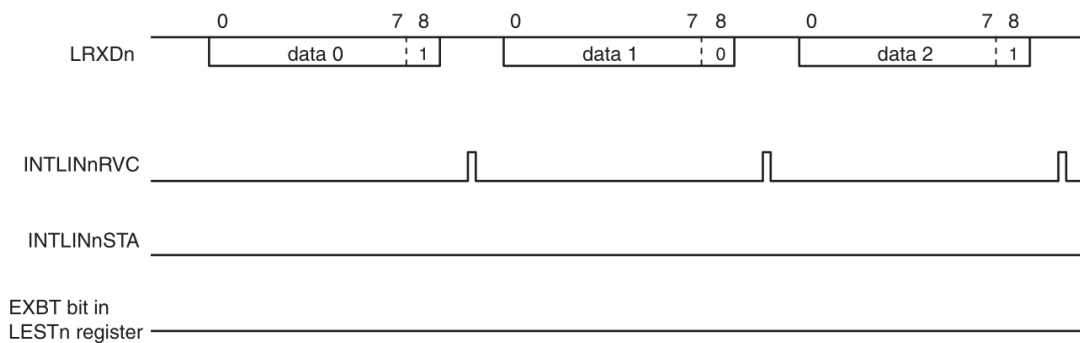
图20-27.允许扩展位的传输示例（LSB优先）



#### (b) 扩展位接收

LIN/UART模块（UART模式）下，只要UART选项寄存器1（LUORn1）中的扩展位允许位（UEBE）为1，扩展位比较禁止位（UECD）为1，扩展位/数据比较允许位（UEBDCE）为0，就可以始终接收9位数据而不需要比较扩展位。无论UART选项寄存器1（LUORn1）中的扩展位检测电平选择位（UEBDL）的具体设置如何，当接收到9位数据时，会产生一个成功的LINn接收中断（n=0, 1）。

图20-28.扩展位接收示例（LSB优先）

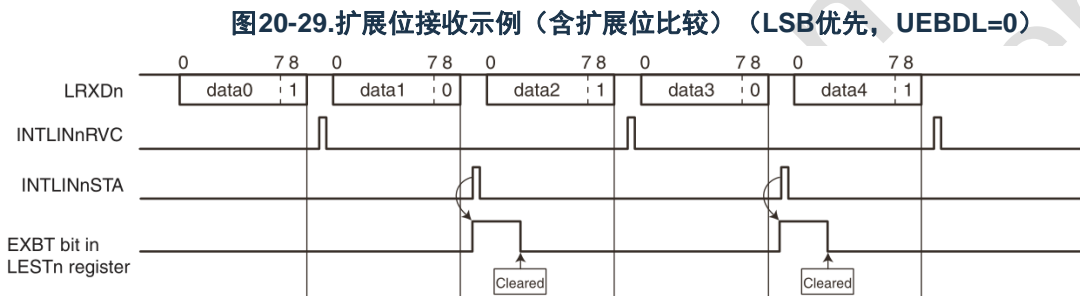


(c) 扩展位接收 (含扩展位比较)

当UART选项寄存器1 (LUORn1) 中的扩展位允许位 (UEBE) 为1, 扩展位比较禁止位 (UECD) 为0, 扩展位/数据比较允许位 (UEBDCE) 为0时, LIN/UART模块 (在UART模式下) 可以比较收到的扩展位和UEDL位。

如果检测到在扩展位检测电平选择位 (UEBDL) 中设置的电平, 则在完成数据接收后产生一个LINn接收状态中断, 并且设置LIN/UART错误状态寄存器 (LESTn) 中的扩展位检测标志 (EXBT)。如果检测到扩展位检测电平的反转值, 则会产生一个成功的LINn接收中断。在这两种情况下, 除非出现过载运行错误, 否则接收的数据将会存储在UART接收数据寄存器 (LURDRn) 中。

图20-29是当扩展位检测水平选择位 (UEDL) 设置为0时的一个例子。



注意1.如果在接收的数据0、2或4 (如果检测到一个扩展位检测电平的反向值) 中发生接收错误 (奇偶校验错误、帧错误或过载运行错误), 将产生一个LINn接收状态中断, 并且更新错误标志。在这种情况下, 不会产生一个成功的LINn接收中断。

2. 如果在接收的数据1或3 (如果检测到扩展位检测电平) 中发生接收错误 (奇偶校验错误、帧错误或过载运行错误), 将产生一个LINn接收状态中断, 并且更新错误标志。在发生过载运行错误的情况下, 设置扩展位检测标志 (EXBT)。

(d) 扩展位接收 (含数据比较)

当UART选项寄存器1 (LUORn1) 中的扩展位允许位 (UEBE) 为1, 扩展位比较禁止位 (UECD) 为0, 扩展位/数据比较允许位 (UEBDCE) 为1时, LIN/UART模块 (在UART模式下) 将不包含扩展位的8位接收数据与预设的LIDBn寄存器值进行比较。如果比较的两个值一致, 将执行以下操作。

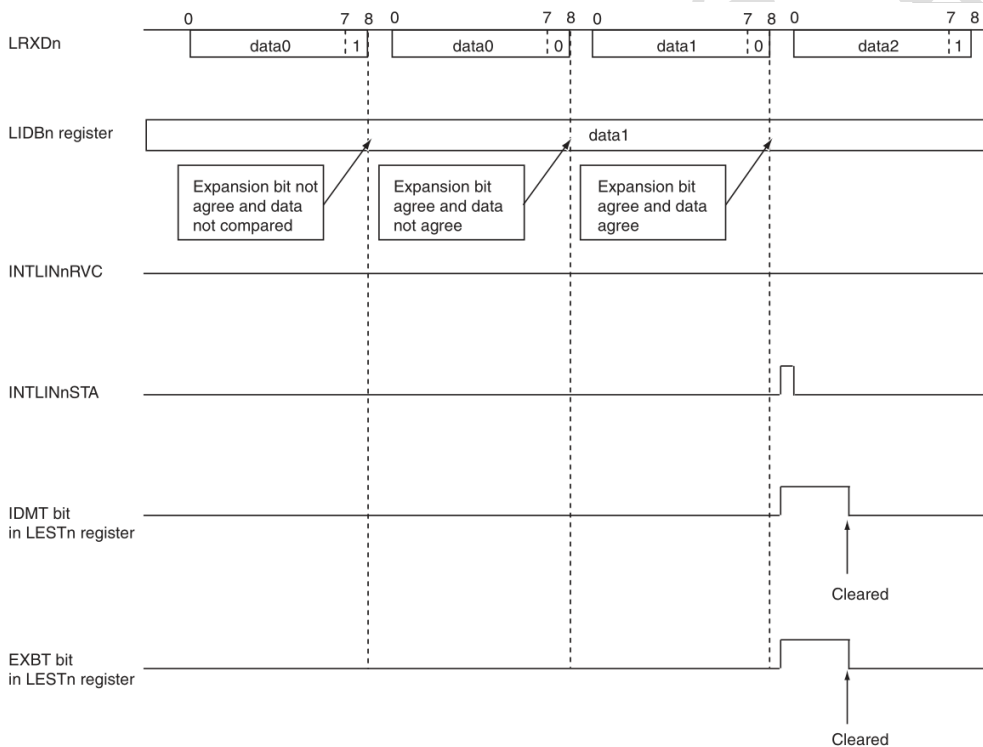
- 产生一个LINn接收状态中断 (n=0或1)。
- 设置扩展位检测标志 (EXBT)。
- 设置ID匹配标志 (IDMT)。
- 在UART接收数据寄存器 (LURDRn) 中存储收到的数据。即使比较的两个值一致, 也不会产生一个成功的LINn接收中断。

如果比较的两个值不一致, 既不产生成功的LINn接收中断, 也不产生LINn接收状态中断, 因此不设置EXBT或IDMT标志为1。接收的数据不存储在UART接收数据寄存器 (LURDRn) 中。

UEBDCE位改为0时, 需确认在收到完整的下一个数据之前完成。

图20-30是当扩展位检测水平选择位 (UEDL) 设置为0时的一个例子。

图20-30.扩展位接收示例 (含数据比较) (LSB优先, UEBDL=0)



**注意** 如果发生接收错误 (奇偶校验错误、帧错误或过载运行错误), 会产生一个LINn接收状态中断, 并更新错误标志。在过载运行错误的情况下, 如果有匹配的比较结果, EXBT和IDMT标志设置为1。

### 20.5.1.4 传输起始等待功能

为了执行半双工通信，LIN/UART模块（在UART模式下）设置了从接收切换到发送这段期间确保停止位接收的功能。

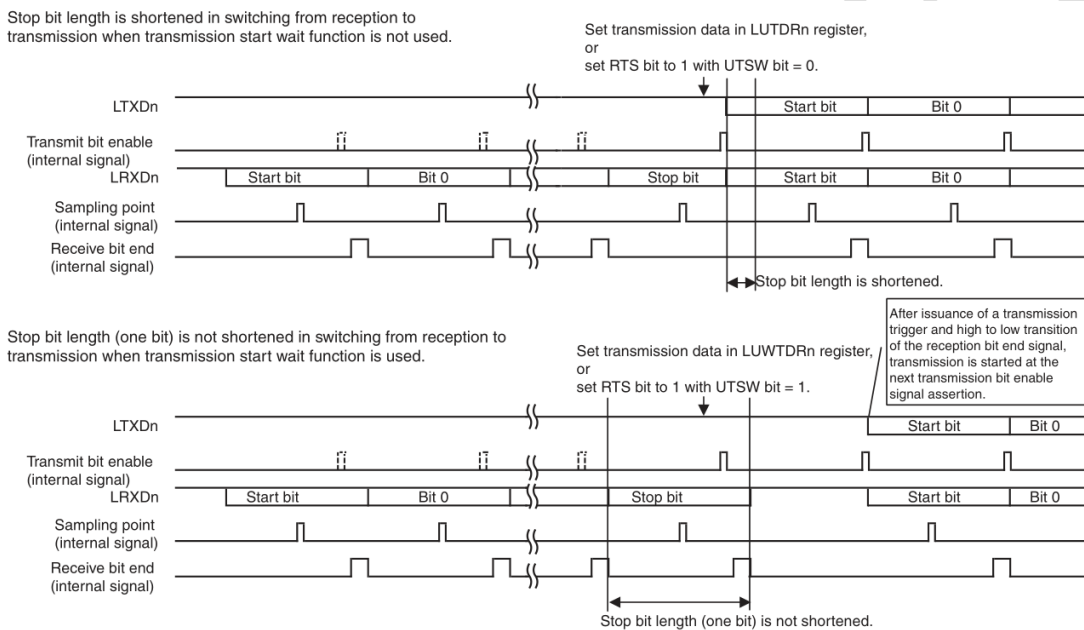
如果将传输的开始时间推迟到接收的停止位完成后，则需要在LUWTDn寄存器中设置数据用于等待，而不是在LUTDRn寄存器中设置传输数据用于请求传输开始。UART缓冲传输时，在LTRCn寄存器的RST位上设置1（UART缓冲传输允许），在LDFCn寄存器的UTSW位上设置1。

在这种情况下，直到接收数据的停止位完成后，LIN/UART模块传输开始。

应该注意的是，即使UART停止位长度选择位（USBLS）为1（停止位=2位），也只对1位进行延迟。

图20-31描述了传输等待功能的操作。

图20-31.传输等待功能（如果在接收数据的停止位期间设置传输数据）



### 20.5.1.5 SNOOZE模式的功能

LIN/UART模块（UART模式）在接收期间提供SNOOZE（瞌睡）模式。在TOP（暂停）模式下检测到LRXDn引脚输入，SNOOZE模式允许在没有CPU操作的情况下接收数据。

如在SNOOZE模式使用LIN/UART模块（UART模式），需在进入STOP模式前要做以下设置。

- 在SNOOZE模式下，设置与正常运行时不同的UART接收波特率。参考表20-21至20-24，设置LBRPn寄存器和LWBRn寄存器中的LPRS[2:0]位和NSPB[3:0]位。
- 设置UART待机控制寄存器（LUSCn）中的UWC位。同时设置LUSCn寄存器中的USEC和URDCC位，以便在发生通信错误和比较接收数据与LIDBn寄存器值时分别允许或禁止错误中断生成。
- 在进入STOP模式之前，将UART操作允许寄存器（LUOERn）中的UROE位设置为1。

进入STOP模式后，检测到LRXDn边沿（起始位输入）后开始UART的接收。

注意事项1.只有当LINCKSEL寄存器中的LINnMCK位为0（选择f<sub>CLK</sub>的2分频）并且高速片上振荡器时钟（f<sub>IH</sub>）选择为f<sub>CLK</sub>，才能设置SNOOZE模式。

2. 当UWC=1时，只有在STOP模式下开始接收时才能使用UART。

如果同时使用另一个SNOOZE功能或中断，并在下述STOP模式以外的任何状态下开始接收，则不能正确接收数据，可能会发生帧错误或奇偶校验错误。

- 将UWC设置为1后，在进入STOP模式前开始接收。
- 在另一个SNOOZE模式运行中开始接收。
- 因中断或其他原因从STOP模式返回到正常操作后，UWC设置为0之前开始接收。

3. 在USEC=1的情况下，如果SNOOZE模式下检测到错误（奇偶校验错误或帧错误）或状态变化（检测到扩展位），该标志不设置为1，从而不产生错误中断。

**表20-21.SNOOZE模式下UART接收的波特率设置  
(LIN通信时钟源=32MHz2%，FRQSEL4=0)**

UART 波特率 (目标)	预分频器 LPRS[2:0]	波特率生成器 0 和 1 LBRP0 和 LBRP1	最大允许值	最小允许值
1200bps	1/2	828	2.45%	-2.50%
2400bps	1/2	412	2.07%	-2.30%
4800bps	1/2	203	1.81%	-1.42%

**表20-22.SNOOZE模式下UART接收的波特率设置  
(LIN通信时钟源=24MHz2%，FRQSEL4=0)**

UART 波特率 (目标)	预分频器 LPRS[2:0]	波特率生成器 0 和 1 LBRP0 和 LBRP1	最大允许值	最小允许值
1200bps	1/2	621	2.41%	-2.55%
2400bps	1/2	308	2.31%	-2.08%
4800bps	1/2	152	1.81%	-1.45%

**表20-23.SNOOZE模式下UART接收的波特率设置  
(LIN通信时钟源=32MHz2%，FRQSEL4=1)**

UART 波特率 (目标)	预分频器 LPRS[2:0]	波特率生成器 0 和 1 LBRP0 和 LBRP1	最大允许值	最小允许值
1200bps	1/2	826	2.23%	-2.26%
2400bps	1/2	410	1.65%	-1.83%

**表20-24.SNOOZE模式下UART接收的波特率设置  
(LIN通信时钟源=24MHz2%，FRQSEL4=1)**

UART 波特率 (目标)	预分频器 LPRS[2:0]	波特率生成器 0 和 1 LBRP0 和 LBRP1	最大允许值	最小允许值
1200bps	1/2	619	2.28%	-2.23%
2400bps	1/2	307	1.73%	-1.76%

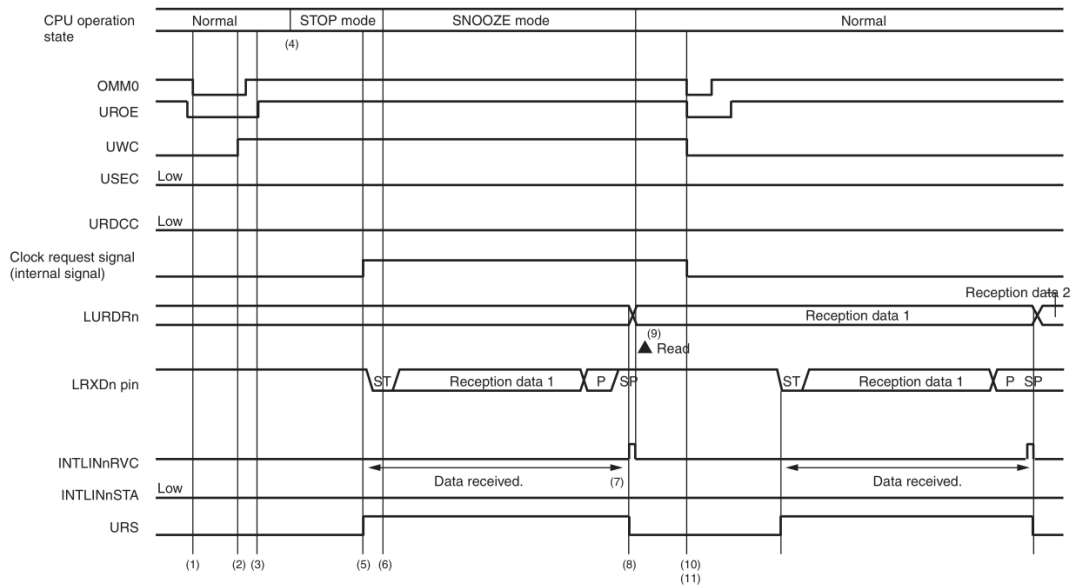
**备注1.**最大和最小允许值适用于UART接收的波特率。这些参数的设置保证了UART传输的波特率也应在允许的范围内。

2.接收数据长度为8位+一个奇偶校验位。

3.表中的数字是针对每一位的十六个样本（即当NSPB[3:0]=0000b或1111b时）。

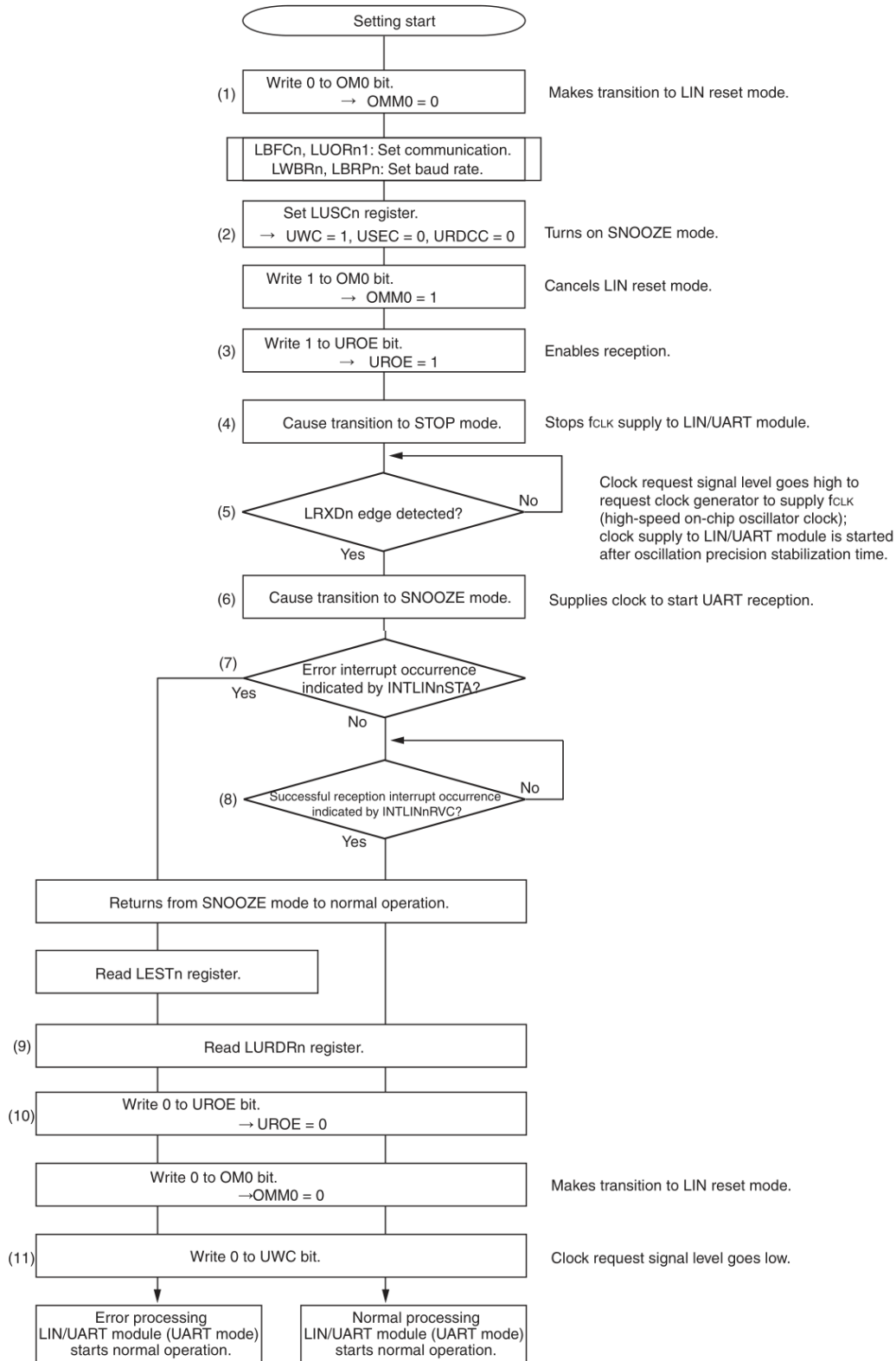
(a) SNOOZE模式操作（成功接收后返回到正常操作；UWC=1, URDCC=0）

图20-32.SNOOZE模式操作的时序图(成功接收后返回正常操作；UWC=1, URDCC=0)



备注 图20-32中的（1）至（11）对应于图20-33中的（1）至（11）

**图20-33.SNOOZE模式的操作流程图**  
(成功接收后返回正常操作; UWC=1, URDCC=0)

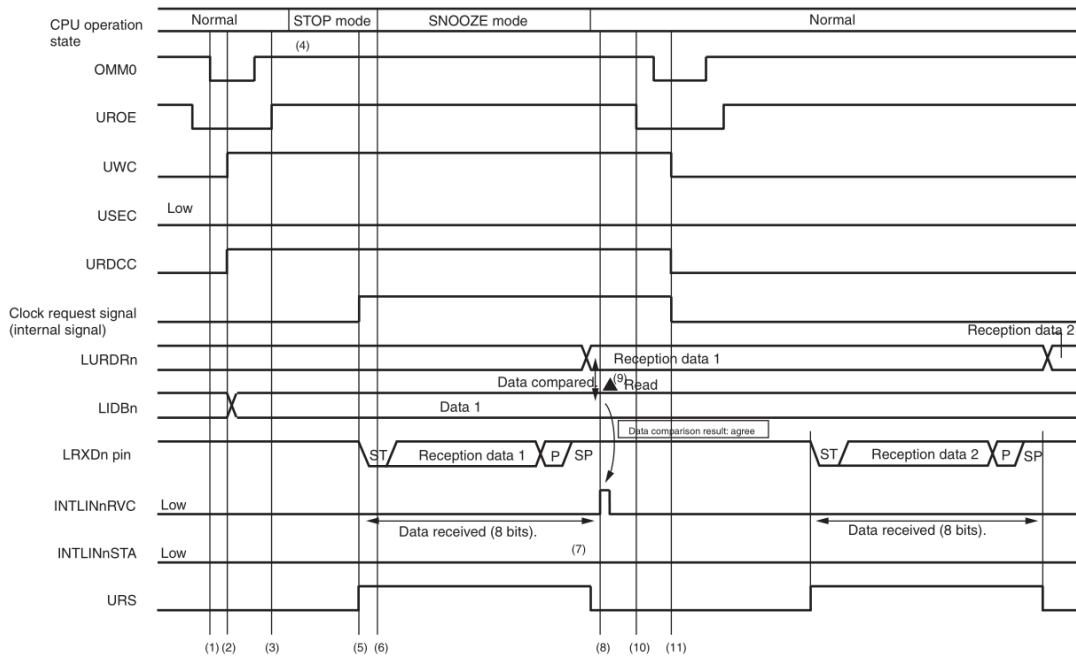


备注 图20-33中的(1)至(11)对应图20-32的(1)至(11)。

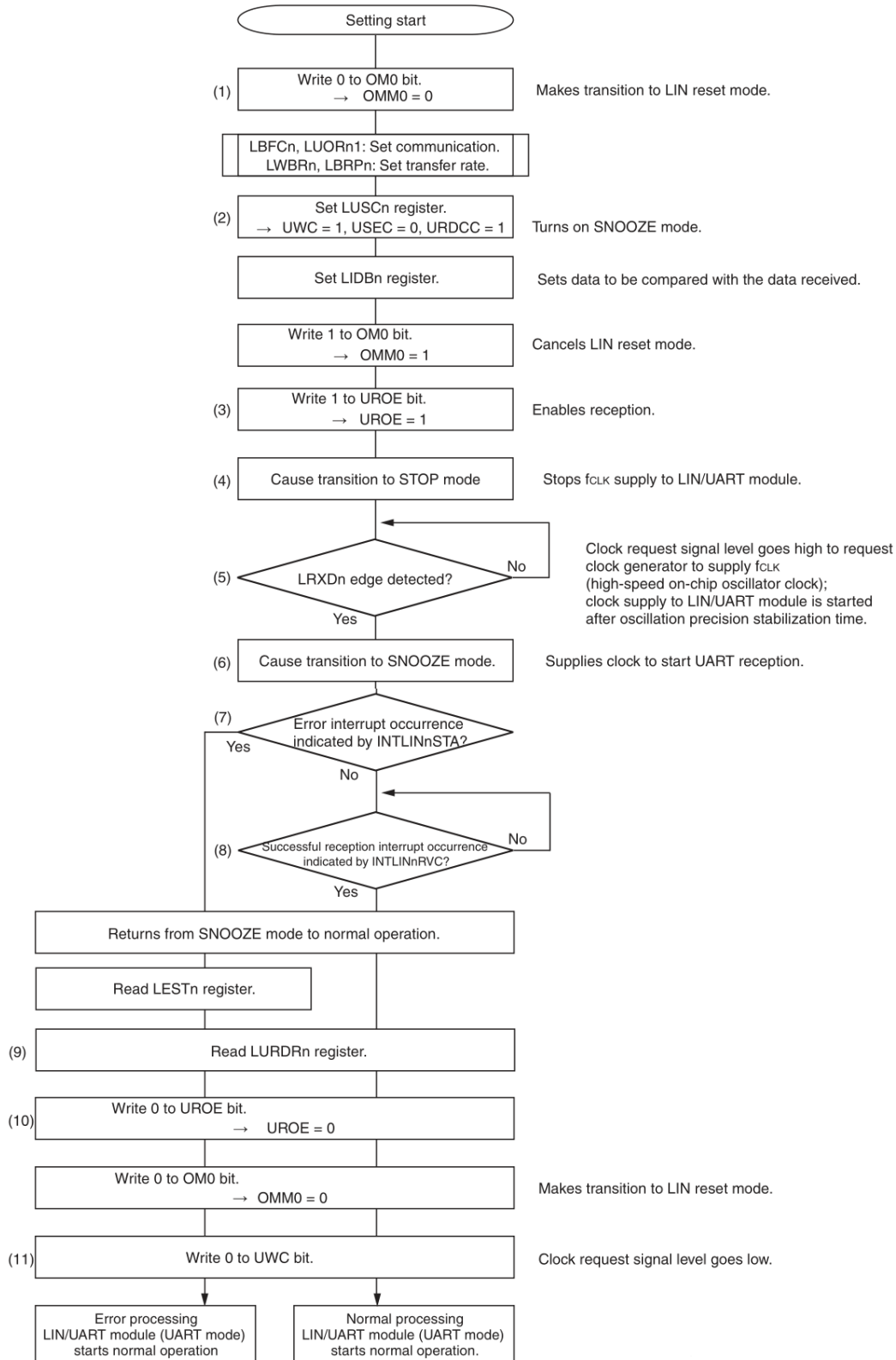


(b) SNOOZE模式操作 (收到的数据比较结果一致时返回到正常操作; UWC=1, URDCC=1)

图20-34.SNOOZE模式操作的时序图  
(收到的数据比较结果一致时返回到正常操作; UWC=1, URDCC=1)



**图20-35.SNOOZE模式的操作流程图**  
(在收到的数据比较结果一致时返回到正常操作; UWC=1, URDCC=1。)



如果要将LUSCn寄存器中的URDCC位设置为1（在SNOOZE模式下允许接收的数据和LIDBn寄存器的数据比较），则只能在LBFCn寄存器中的UBLS位设置为0（UART8位字符通信）和LUORn1寄存器中的UEBE位设置为0（禁止扩展位操作）的情况下使用SNOOZE模式。

## 20.5.2 数据传输/接收

### 20.5.2.1 数据传输

每Tbit传输一位数据。

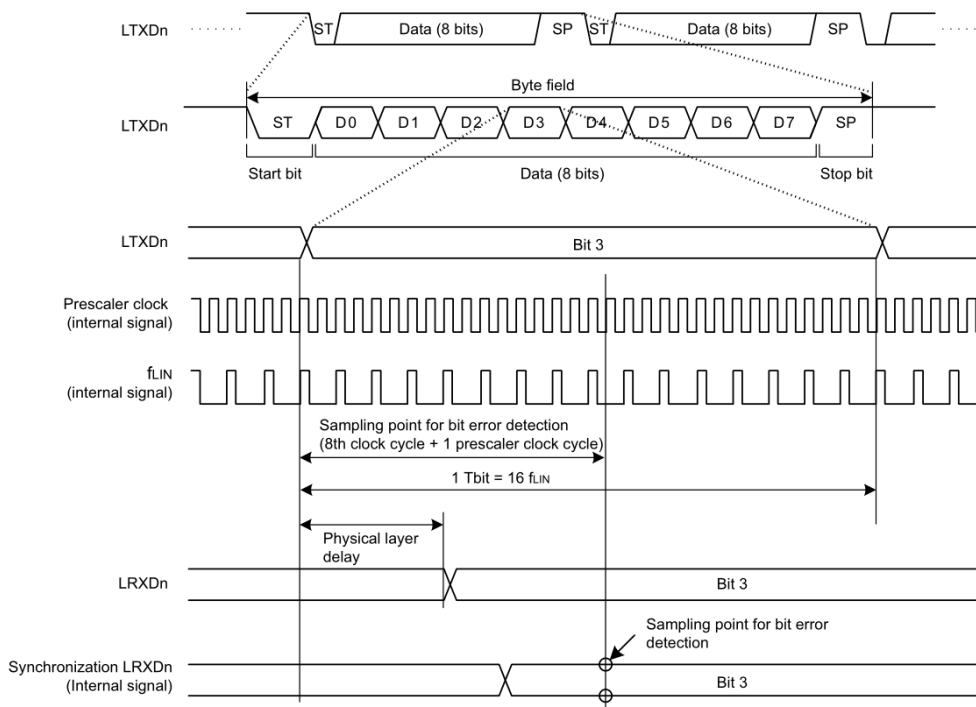
在半双工通信中，如果LEDEn寄存器中的BERE位为1（允许位错误检测），则在数据传输过程中逐位比较传输数据和输入引脚的电平，并将结果保存在LESTn寄存器的BER标志中（参考20.5.5错误状态）。在数据传输过程中，输入引脚采样的时间可以根据LWBRn寄存器中LPRS[2:0]和NSPB[3:0]位的设置而变化。

UART模式下的位错误检测时序如表20-25所示。

表20-25.UART模式下的错误检测时序

每位的采样次数	位错误检测时间
6 个样本	第 3 个时钟周期+预分频器时钟的一个周期
7 个样本	第 4 个时钟周期+预分频器时钟的一个周期
8 个样本	第 4 个时钟周期+预分频器时钟的一个周期
9 个样本	第 5 个时钟周期+预分频器时钟的一个周期
10 个样本	第 5 个时钟周期+预分频器时钟的一个周期
11 个样本	第 6 个时钟周期+预分频器时钟的一个周期
12 个样本	第 6 个时钟周期+预分频器时钟的一个周期
13 个样本	第 7 个时钟周期+预分频器时钟的一个周期
14 个样本	第 7 个时钟周期+预分频器时钟的一个周期
15 个样本	第 8 个时钟周期+预分频器时钟的一个周期
16 个样本	第 8 个时钟周期+预分频器时钟的一个周期

图20-36.数据传输时序示例（当1Tbit中采样计数为16时）



### 20.5.2.2 数据接收

数据通过同步的LRXDn（一个内部信号）进行接收，该信号是与预分频器时钟同步的LRXDn引脚输入。

字节场在LRXDn信号的起始位的下降沿进行同步。在检测到下降沿后，如果每1Tbit的采样计数为偶数，则在0.5Tbit后进行再次采样；如果为奇数，则在{(采样计数+1)/2}/(采样计数) Tbits后进行重采样。如果同步的LRXDn信号为低电平，则该位识别为起始位。如果复位后LRXDn信号固定在低电平，或者在再次采样期间检测到高电平，则该位不会识别为起始位。

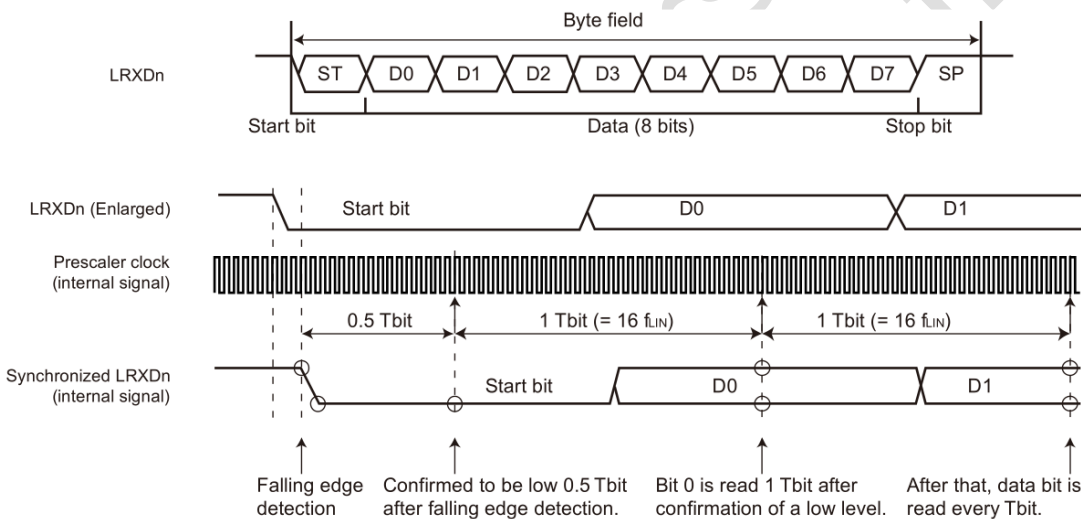
检测到起始位后，每Tbit采样1位。

注意，当LEDEn寄存器中的BERE位设置为1时，采样与检测到的位错误同时进行。

LIN/UART模块对接收的数据会进行噪声过滤。如果LMDn寄存器中的LRDNFS位为0，则使用噪声过滤。对于采样值，使用预分频器时钟的3次采样多数规则（3-samplingmajorityrule）确定的值。如果LMDn寄存器中的LRDNFS位为1，则不使用噪声过滤。在这种情况下，对于一个采样值，采样位置的LRXDn值会按照原样使用。

图20-37是一个数据接收时序的示例。

图20-37.数据接收时序示例（当1Tbit中采样计数为16时）

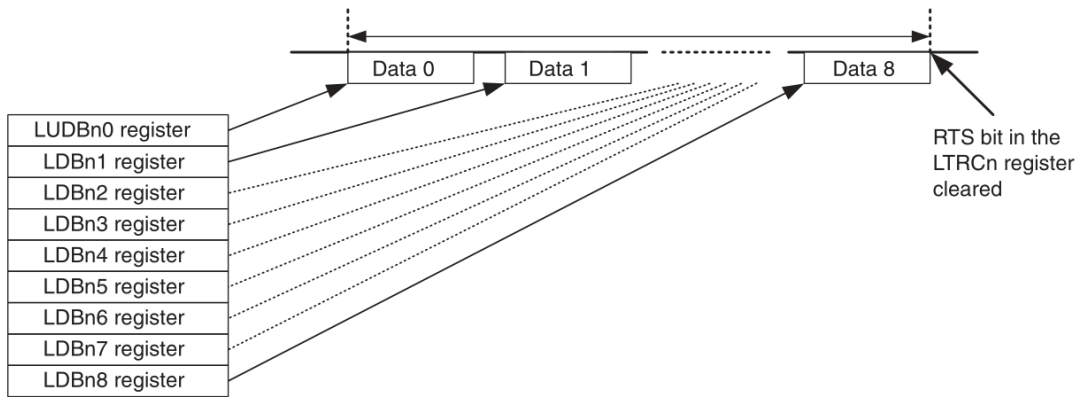


## 20.5.3 传输数据的缓冲处理

### 20.5.3.1 UART缓冲传输

对于9字节的传输，存储在LUDBn0和LDBn1至LDBn8寄存器中的内容传输到数据区0至8。LUDBn0寄存器只有在设置了9字节传输时才会使用。在其他情况下，根据所涉及的数据长度，选择LDBn1至LDBn8寄存器。对于4字节的传输，存储在LDBn1至LDBn4寄存器中的内容传输到数据区1至4，但LDBn5至LDBn8寄存器的内容不传输。在传输完LDFCn寄存器中MDL[3:0]位设置的数据后，会产生一个LINn传输中断（n=0）。传输数据项之间的间隔可以在LSCn寄存器的IBS位中设置。图20-38描述了一个9字节的UART缓冲和传输处理。

图20-38.UART缓冲和传输处理（用于9字节的传输）



## 20.5.4 状态

在UART模式下，LIN/UART模块可以检测出五种状态。  
成功的UART缓冲区传输和错误检测，这两种状态可以产生中断请求。  
表20-26描述了UART模式下适用的状态类型。

表20-26.UART模式下的状态类型

状态	状态设置条件	状态清除条件	相应的位	中断
复位	如果实际上LIN/UART模块从LIN复位模式清除，在LCUCn寄存器中的OM0位设置为非LIN复位模式后。	如果实际上LIN/UART模块进入了LIN复位模式，在LCUCn寄存器中的OM0位设置为LIN复位模式后。	LMSTn寄存器中的OMM0位	不适用
成功的UART缓冲区传输	<ul style="list-style-type: none"> <li>•LUORn1寄存器中的UTIGTS位为0时，开始传输相当于LDFCn寄存器中MDL位所设定长度的最后一个数据（传输开始时产生传输中断）。</li> <li>•传输的数据长度等于LDFCn寄存器中的MDL位完成设置的长度，且LUORn1寄存器中的UTIGTS位为1（传输完成时产生传输中断）。</li> </ul>	<ul style="list-style-type: none"> <li>•当软件清除的时候</li> <li>•转换到LIN复位模式后</li> </ul>	LSTn寄存器中的FTC标志	适用
错误检测	如果LESTn寄存器中的UPER标志、IDMT标志、EXBT标志、FER标志、OER标志和BER标志中的任何一个变为1（错误检测）。	<ul style="list-style-type: none"> <li>•软件清除的时候</li> <li>•转换到LIN复位模式后</li> </ul>	LSTn寄存器中的ERR标志	适用
传送状态	<ul style="list-style-type: none"> <li>•数据写入LUTDRn或LUWDRn寄存器时。</li> <li>•在LTRCn寄存器RTS位写入1时。</li> </ul>	<ul style="list-style-type: none"> <li>•LUTDRn或LUWDRn寄存器中设置的数据传输完成，但未设置另一个传输数据项。</li> <li>•UART缓冲区中的数据传送完成后，且LTRCn寄存器中的RTS位清零。</li> <li>•转换到LIN复位后模式</li> </ul>	LSTn寄存器中的UTS标志	适用
接收状态	<ul style="list-style-type: none"> <li>•检测到一个起始位时。</li> </ul>	<ul style="list-style-type: none"> <li>•检测到一个停止位的采样点时</li> <li>•转换到LIN复位模式后</li> </ul>	LSTn寄存器中的URS标志	不适用

注：取消LIN复位模式时，向LESTn寄存器中的UPER、IDMT、EXBT、FET、OER和BER标志写0，会使LSTn寄存器中的ERR标志变为0。

## 20.5.5 错误状态

在UART模式下，LIN/UART模块可以检测出四种类型的错误和两种类型的状态。这些错误状态的状态可以通过LESTn寄存器中的相应位来检查。

表20-27列出了适用的错误状态类型。

表20-27.UART模式下的错误状态类型

状态	错误检测条件	通信	允许/禁止检测	相应的位
位错误	传输的数据和接收引脚上监测的数据不匹配 <sup>注1</sup>	持续到设定的传输数据传输完毕。	○	LESTn 寄存器中的 BER 标志
过载运行错误	在收到的数据存储于 LURDRn 寄存器中后, 读取数据之前会收到另一个数据项。 (在这种情况下, 没有数据存储在 LURDRn 寄存器中)。	- (检测到此错误时, 接收已经结束)	○	LESTn 寄存器中的 OER 标志
帧错误	在接收处理中, 当第一位的第一个停止位为低电平时。	- (检测到这个错误时, 接收已经结束)	○	LESTn 寄存器中的 FER 标志
奇偶校验错误	接收到的奇偶校验值与从收到的数据计算出的奇偶校验值不一致	持续到数据接收结束。	× <sup>注2</sup>	LESTn 寄存器中的 UPER 标志
扩展位检测	收到的扩展位的值与 LUORn1 寄存器中的 UEEDL 位的值一致。	-	○	LESTn 寄存器中的 EXBT 标志
ID 匹配	接收到的扩展位的值与 LUORn1 寄存器中的 UEEDL 位的值一致, 除扩展位外的 8 位接收数据与 LIDBn 寄存器的值一致。	-	○	LESTn 寄存器中的 IDMT 标志

注1. 如果数据从UART缓冲区传输, 在UART帧之间的空间(字节间隔)也会检测到位错误。

2. 将LBFCn寄存器中的UPS[1:0]位设置为10b(0校验), 可禁止奇偶校验位值的检查。在这种情况下, 不会产生奇偶校验错误。

**注意** 软件清除或转换到LIN复位模式后, 错误状态清除。

## 20.6 LIN自测模式

LIN/UART模块有一个LIN自测模式。当LIN自测试开启时，LTXDn和LRXDn信号从外部引脚断开，并在LIN/UART模块内连接。从内部LTXDn terminal传输的帧循环回到内部LRXDn terminal。LIN自测模式是专门用于测试LIN模式的操作。适用以下四种类型的测试：

- LIN主机自检模式（传输）：帧头传输和响应传输
- LIN主机自检模式（接收）：帧头传输和响应接收
- LIN从机自检模式（传输）：帧头接收和响应传输
- LIN从机自检模式（接收）：帧头接收和响应接收

在LIN自测模式下，无论波特率发生器的设置如何，LIN/UART模块都以最高波特率运行。无论波特率相关寄存器的设置如何，波特率都是 $\langle \text{LIN通信时钟源的频率} \rangle / 16\text{bps}$ （LWBRn寄存器中的NSPB位必须设置为0000b或1111b）。

在LIN自测模式下，不支持以下功能。请不要使用这些功能。

- LIN唤醒模式
- 帧分离模式
- 多字节响应的接收和传输
- LIN从机模式[自动波特率]
- 帧/响应超时错误

图20-39.在LIN复位模式、LIN模式和UART模式下的连接

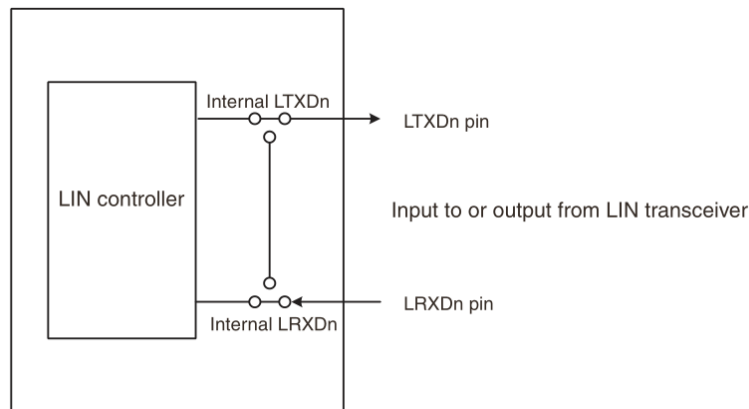
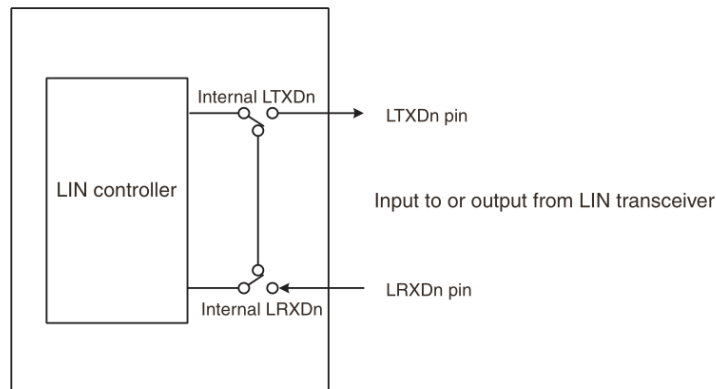


图20-40.LIN自测模式的连接





## 20.6.1 转换到LIN自测模式

通过写入LSTCn寄存器进入LIN自测模式。

当LSTCn寄存器中的LSTM位变为1时，可以确认转换到LIN自测模式。

当转换到LIN自检模式时，一定要执行一个特定的程序。在该程序中，必须向LIN自检控制寄存器连续写入三次信息，如下所示：

- 转换为LIN复位模式  
将LCUCn寄存器中的OM0位设为0（LIN复位模式）。  
读取LMSTn寄存器中的OMM0位；确认其为0（LIN复位模式）。
- 选择LIN模式  
LMDn寄存器中的LMD位=00b（LIN主机模式）或11b（LIN从机模式[固定波特率]）。
- 第一次写入：LSTCn寄存器=10100111b(A7H)
- 第二次写入：LSTCn寄存器=01011000b(58H)
- 第三次写入：LSTCn寄存器=00000001b(01H)
- 验证已转换到LIN自测模式

读取LSTCn寄存器中的LSTM位；确认其为1（LIN自测模式）。

如果第一次写的键（A7H）误写了两次，就会取消向LIN自检模式的转换。应从上述顺序第一次写入的步骤开始重试。此外，如果在到LIN自检模式期间对另一个与LIN有关的寄存器进行了写入操作（对LSTCn寄存器连续进行了三次写操作），转换也会取消。

## 20.6.2 LIN主机自测模式下的传输

要在LIN主机自测模式运行传输，请执行以下程序：

- 设置与波特率、噪声滤波器和中断输出有关的寄存器
  - LWBRn寄存器=0000xxxxb<sup>注1</sup>
  - LBRPn0寄存器=xxxxxxx<sup>注1</sup>
  - LBRPn1寄存器=xxxxxxx<sup>注1</sup>
  - LMDn寄存器=00xxx00b<sup>注1,3</sup>
- 设置与中断允许和错误允许有关的寄存器。
  - LIEn寄存器=0000xxxxb<sup>注2,3</sup>
  - LEDEn寄存器=x000x0xb
- 设置与间隔场和空间有关的寄存器。
  - LBFCn寄存器=00xxxxxb
  - LSCn寄存器=00xx0xxx
- 取消复位。
  - 将11b写入LCUCn寄存器中的OM1和OM0位，并验证LMSTn寄存器中的OMM1和OMM0位变为11b。
- 设置与传输帧有关的寄存器。
  - LDFCn寄存器=00x1xxxxb
  - LIDBn寄存器=xxxxxxx<sup>注1</sup>
  - LDBn1至LDBn8寄存器=xxxxxxx<sup>注1</sup>
- 开始帧头传输，然后响应传输
  - LDFCn寄存器=00x1xxxxb
  - 将LTRCn寄存器中的FTS位设置为1（帧传输或唤醒传输/接收开始）。

执行LIN主机自检模式（传输）。在这种模式下，会产生中断，状态和错误状态也会适当地更新。LIN/UART模块自动计算校验和。

当LIN主机自检模式（传输）的执行终止时，LCUCn寄存器的OM0位设置为0（LIN复位模式）。

- 当传输完成后，环回帧数据的反转值存储在LIDBn,LDBnm(m=1~8)和LCBRn寄存器中（因为传输值应与环回值比较，数据在存储前反转）。LTRCn寄存器中的FTS位清零。
- 如果传输由于错误而未能完成，设置适用的错误标志。LTRCn寄存器中的FTS位清零。

**注1.** 下列寄存器的设置不影响LIN自测模式下的操作。因此，对它们的设置不是强制性的。

LWBRn寄存器、LBRPn0寄存器、LBRPn1寄存器中的LPRS位和LMDn寄存器中的LCKS位。

2. 根据需要，可参考第24章中断功能设置相关的寄存器。

3. 当成功帧头传输中断和成功帧传输中断在同一个中断时，如果在成功帧传输中断产生之前，成功帧头传输中断的软件处理没有完成，那么LIEn寄存器中的SHIE位不应该设置为1（成功帧头传输中断允许）。

从成功帧头传输标志设置开始，到成功的帧/唤醒传输标志设置为止，这段时间可以按以下方式计算：

$$10 \times (\text{数据字节数} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = \text{LIN通信时钟源} \times 16$$

**备注x:**任何期望值

## 20.6.3 LIN主机自测模式下的接收

要在LIN主机自测模式下运行接收，请执行以下程序：

- 设置与波特率、噪声滤波器和中断输出有关的寄存器。
  - LWBRn寄存器=0000xxxxb<sup>注1</sup>
  - LBRPn0寄存器=xxxxxxx<sup>注1</sup>
  - LBRPn1寄存器=xxxxxxx<sup>注1</sup>
  - LMDn寄存器=00xxx00b<sup>注1, 3</sup>
- 设置与中断允许和错误允许有关的寄存器。
  - LIE n寄存器=0000xxxxb<sup>注2, 3</sup>
  - LEDEn寄存器=x000x0xb
- 设置与间隔场和空间有关的寄存器。
  - LBFCn寄存器=00xxxxxb
  - LSCn寄存器=00xx0xxx<sup>注1</sup>
- 取消复位。
  - 将11b写入LCUCn寄存器中的OM1和OM0位，并验证LMSTn寄存器中的OMM1和OMM0位变为11b。
- 设置与接收帧有关的寄存器。
  - LDFCn寄存器=00x0xxxxb
  - LIDBn寄存器=xxxxxxx<sup>注1</sup>
  - LDBn1至LDBn8寄存器=xxxxxxx<sup>注1</sup>
  - LCBRn寄存器=xxxxxxx<sup>注1</sup>

由于校验和不是自动计算的，所以要存储一个计算值。通过故意设置一个不正确的计算结果作为校验和，可以对校验和错误进行测试。

- 开始帧头传输，然后响应接收
  - 将LTRCn寄存器中的FTS位设置为1（帧传输或唤醒传输/接收开始）。

执行LIN主机自检模式（接收）。在这种模式下，会产生中断，状态和错误状态也会适当地更新。当LIN主机自检模式（接收）的执行终止时，将LCUCn寄存器的OM0位设置为0（LIN复位模式）。

- 当接收完成后，环回帧数据的反转值存储在LIDBn、LDBnm（m=1~8）和LCBRn寄存器中（因为设定值应与回环值和接收值进行比较，数据在存储前反转）。LTRCn寄存器中的FTS位清零。
- 如果由于错误而不能完成接收，设置适用的错误标志。LTRCn寄存器中的FTS位清零。

**注意1.** 下列寄存器的设置不影响LIN自测模式下的操作。因此，对它们的设置不是强制性的。

LWBRn寄存器、LBRPn0寄存器、LBRPn1寄存器中的LPRS位，LMDn寄存器中的LCKS位和LSCn寄存器中的IBS位。

2. 根据需要，可参考第24章中断功能设置相关的寄存器。

3. 当成功帧头传输中断和成功帧传输中断在同一个中断时，如果在成功帧传输中断产生之前，成功帧头传输中断的软件处理没有完成，那么LIE n寄存器中的SHIE位不应该设置为1（成功帧头传输中断允许）。

从成功帧头传输标志设置开始，到成功的帧/唤醒传输标志设置为止，这段时间可以按以下方式计算：

$$10 \times (\text{数据字节数} + 1) [\text{Tbit}]$$

$$1 \text{Tbit} = \text{LIN通信时钟源} \times 16$$

**备注x:**任何期望值

## 20.6.4 LIN从机自测模式下的传输

要在LIN从机自测模式下运行传输，请执行以下程序：

- 设置与波特率、噪声滤波器和中断输出有关的寄存器。
  - LWBRn寄存器=0000xxx0b<sup>注1</sup>
  - LBRPn0寄存器=xxxxxxx0b<sup>注1</sup>
  - LBRPn1寄存器=xxxxxxx0b<sup>注1</sup>
  - LMDn寄存器=00xx0011b<sup>注4</sup>
- 设置与中断允许和错误允许有关的寄存器。
  - LIE n寄存器=0000xxx0b<sup>注2、4</sup>
  - LEDEn寄存器=xx0xx00xb
- 设置与间隔场和空间有关的寄存器。
  - LBFCn寄存器=0000000xb<sup>注3</sup>
  - LSCn寄存器=00xx0001b
- 取消复位。
  - 将11b写入LCUCn寄存器中的OM1和OM0位，并验证LMSTn寄存器中的OMM1和OMM0位变为11b。
- 设置与传输帧有关的寄存器。
  - LDFCn寄存器=00x1xxxxb
  - LIDBn寄存器=xxxxxxx0b
  - LDBn1至LDBn8寄存器=xxxxxxx0b
- 开始帧头接收，然后响应传输
  - 将LTRCn寄存器中的FTS位设置为1（帧头接收或唤醒传输/接收开始）。
  - （在不涉及LTRCn寄存器中RTS位任何操作的情况下，按指定的顺序执行帧头接收和响应传输）。

执行LIN从机自检模式（传输）。在这种模式下，会产生中断，状态和错误状态也会适当地更新。LIN/UART模块自动计算校验和。

当LIN主机自检模式（传输）执行中止时，将LCUCn寄存器的OM0位设置为0（LIN复位模式）。

- 当传输完成后，回环帧数据的反转值存储在LIDBn、LDBnm（m=1~8）和LCBRn寄存器中（因为传输值应与回环值进行比较，数据在存储前反转）。LTRCn寄存器中的FTS位清零。
- 如果传输由于错误而未能完成，设置适用的错误标志。LTRCn寄存器中的FTS位清零。

**注意1.** 下列寄存器的设置不影响LIN自测模式下的操作。因此，对它们的设置不是强制性的。

LWBRn寄存器、LBRPn0寄存器和LBRPn1寄存器中的LPRS位。

2. 根据需要，可参考第24章中断功能设置相关的寄存器。

3. 根据该寄存器的设置，将从内部LTXDn引脚输出一个宽度为9.5或10.5Tbits的间隔。

4. 当成功帧头接收中断和成功响应传输中断在同一个中断时，如果在成功响应传输中断产生之前，成功帧头接收中断的软件处理没有完成，那么LIE n寄存器中的SHIE位不应该设置为1（成功帧头接收中断允许）。

从成功帧头接收标志设置开始，到成功的响应/唤醒传输标志设置为止，这段时间可以按以下方式计算：

$$10 \times (\text{数据字节数} + 1) [\text{Tbit}]$$

$$1 \text{Tbit} = \text{LIN通信时钟源} \times 16$$

**备注x:**任何期望值

## 20.6.5 LIN从机自检模式下的接收

要在LIN从机自测模式下运行接收，请执行以下程序：

- 设置与波特率、噪声滤波器和中断输出有关的寄存器。
  - LWBRn寄存器=0000xxx0b<sup>注1</sup>
  - LBRPn0寄存器=xxxxxxx0b<sup>注1</sup>
  - LBRPn1寄存器=xxxxxxx0b<sup>注1</sup>
  - LMDn寄存器=00xx0011b<sup>注4</sup>
- 设置与中断允许和错误允许有关的寄存器。
  - LIE n寄存器=0000xxx0b<sup>注2、4</sup>
  - LEDEn寄存器=xx0xx00xb
- 设置与间隔场和空间有关的寄存器。
  - LBFCn寄存器=0000000xb<sup>注3</sup>
  - LSCn寄存器=00xx0001b<sup>注1</sup>
- 取消复位。
  - 将11b写入LCUCn寄存器中的OM1和OM0位，并验证LMSTn寄存器中的OMM1和OMM0位变为11b。
- 设置与接收帧有关的寄存器。
  - LDFCn寄存器=00x0xxxxb
  - LIDBn寄存器=xxxxxxx0b
  - LDBn1至LDBn8寄存器=xxxxxxx0b
  - LCBRn寄存器=xxxxxxx0b

由于校验和不是自动计算的，所以要存储一个计算值。通过故意设置一个不正确的计算结果作为校验和，可以对校验和错误进行测试。

- 开始帧头接收，然后响应接收。
  - 将LTRCn寄存器中的FTS位设置为1（帧头接收或唤醒传输/接收开始）。
  - （在不涉及LTRCn寄存器中RTS位任何操作的情况下，按指定的顺序执行帧头接收和响应接收）。

LIN执行从机自检模式（接收）。在这种模式下，会产生中断，状态和错误状态也会适当地更新。当LIN主机自检模式（接收）的执行中止时，将LCUCn寄存器的OM0位设置为0（LIN复位模式）。

- 当接收完成后，环回帧数据的反转值存储在LIDBn、LDBnm（m=1~8）和LCBRn寄存器中（因为设定值应与回环值和接收值进行比较，数据在存储前反转）。LTRCn寄存器中的FTS位清零。
- 如果由于错误而不能完成接收，设置适用的错误标志。LTRCn寄存器中的FTS位清零。

**注意1.** 下列寄存器的设置不影响LIN自检模式下的操作。因此，对它们的设置不是强制性的。

LWBRn寄存器、LBRPn0寄存器、LBRPn1寄存器中的LPRS位和LSCn寄存器中的IBS位。

2. 根据需要，可参考**第24章 中断功能**设置相关的寄存器。

3. 根据该寄存器的设置，将从内部LTXDn引脚输出一个宽度为9.5或10.5Tbits的间隔。

4. 当成功帧头接收中断和成功响应接收中断在同一个中断时，如果在成功响应接收中断产生之前，成功帧头接收中断的软件处理没有完成，那么LIE n寄存器中的SHIE位不应该设置为1（成功帧头接收中断允许）。

从成功的帧头接收标志设置开始，到成功的响应/唤醒接收标志设置为止，这段时间可以按以下方式计算：

$$10 \times (\text{数据字节数} + 1) [\text{Tbit}]$$

$$1 \text{Tbit} = \text{LIN通信时钟源} \times 16$$

**备注x:**任何期望值

## 20.6.6 终止LIN自测模式

要终止LIN自测模式，请执行以下程序。

- 向LCUCn寄存器中的OM0位写0（LIN复位模式）。  
如果LMSTn寄存器中的OMM1和OMM0位不是11b，则向LCUCn寄存器中的OM1和OM0位写入11b。在确认LMSTn寄存器中的OMM1和OMM0位变成11b后，转换为LIN复位模式。
- 验证LIN自测模式的取消。  
读取LSTCn寄存器中的LSTM位；确认其不为0（非LIN自检模式中）。
- 验证是否转换到LIN复位模式。  
读取LMSTn寄存器中的OMM0位；确认其为0（LIN复位模式）。

Draft Version  
Cmsemicon Confidential

## 20.7 波特率发生器

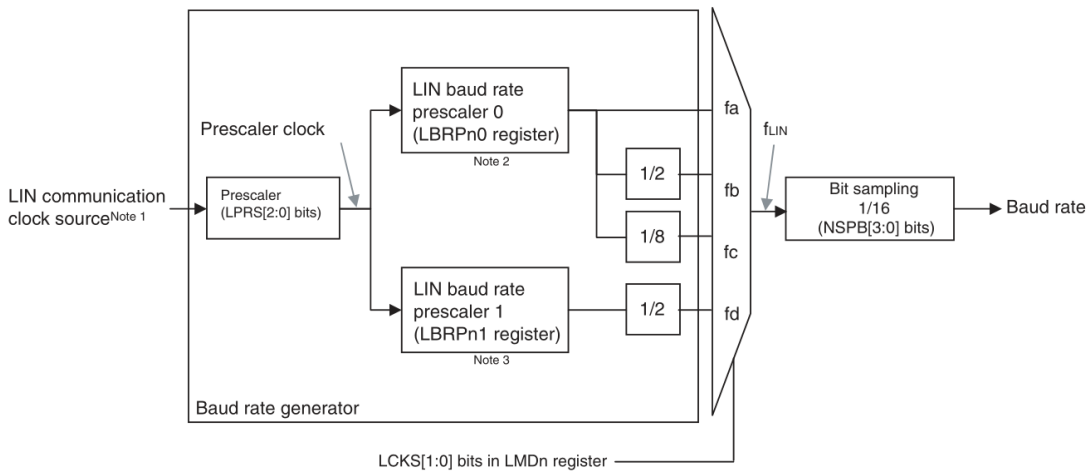
预分频器时钟通过预分频器对LIN通信时钟源进行分频而获得，LIN系统时钟 ( $f_{LIN}$ ) 是通过波特率发生器对预分频器时钟进行分频而获得。通过采样数对LIN系统时钟 ( $f_{LIN}$ ) 分频得到的时钟就是波特率。该波特率的倒数称为位时间 (Tbit)。

LIN/UART模块有两种波特率生成器。波特率发生器根据使用的模式进行切换。

### 20.7.1 LIN主机模式

图20-41是LIN主机模式下波特率的结构图。

图20-41.LIN主机模式下波特率生成结构图



- 备注1.关于LIN通讯的时钟源，请参考第5章时钟发生器。  
 2.LBRPn0寄存器中的值为N (N=0~255) 时，时钟频率除以N+1。  
 3. LBRPn1寄存器中的值为M (M=0~255) 时，时钟频率除以M+1。

设置LIN通信时钟源如下：

- LIN通信时钟源= $f_{CLK}/2^{\text{注1}}$
- 在4MHz至32MHz的范围内

**注1.**当不使用超时错误检测时，可选择 $f_{MX}$ 时钟作为LIN通信时钟源。在这种情况下，将LIN通信时钟源的频率至少设置为CPU/外围硬件时钟 ( $f_{CLK}$ ) 的1.2倍。

通过设置LBRPn0寄存器，fa为307200Hz (=19200×16)，得到的比特率为fa=19200×16，fb=9600×16和fc=2400×16。这些比特率在比特时序发生器中除以16的频率，可以产生19200bps、9600bps和2400bps的比特率。另外，通过设置LBRPn1寄存器，使fd为166672Hz (=10417×16)，产生的比特率为fd=10417×16。这个比特率在比特时序发生器中频率除以16，就可以生成10417bps。

**表20-28**是每个LIN通信时钟源频率的波特率 (19200、10417、9600和2400bps) 产生的例子，及相对应的误差。

**表20-28.LIN主机模式下波特率（19200、10417、9600和2400bps）生成示例**

LIN 通信 时钟源	预分频器	波特率发生器 0 N+1 分频器	波特率发生器 1 M+1 分频器	LIN 系统时钟	波特率	误差
32MHz	1/1	104	-	fa	19230.77	+0.16%
		-	96	fd	10416.67	-0.003%
		104	-	fb	9615.38	+0.16%
		104	-	fc	2403.85	+0.16%
24MHz	1/1	78	-	fa	19230.77	+0.16%
		-	72	fd	10416.67	-0.003%
		78	-	fb	9615.38	+0.16%
		78	-	fc	2403.85	+0.16%
16MHz	1/1	52	-	fa	19230.77	+0.16%
		-	48	fd	10416.67	-0.003%
		52	-	fb	9615.38	+0.16%
		52	-	fc	2403.85	+0.16%
12MHz	1/1	39	-	fa	19230.77	+0.16%
		-	36	fd	10416.67	-0.003%
		39	-	fb	9615.38	+0.16%
		39	-	fc	2403.85	+0.16%
8MHz	1/1	26	-	fa	19230.77	+0.16%
		-	24	fd	10416.67	-0.003%
		26	-	fb	9615.38	+0.16%
		26	-	fc	2403.85	+0.16%

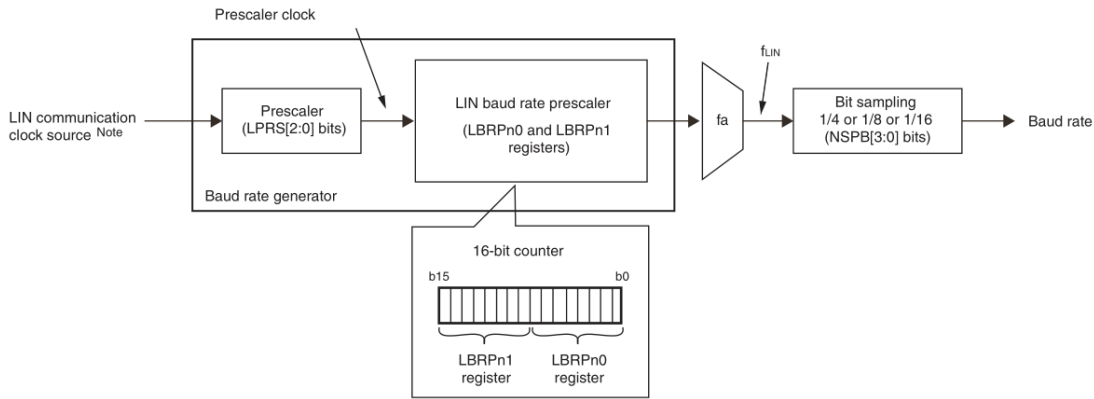
**备注** 表中的数字是针对每一位的16个采样（例如当NSPB[3:0]=0000b或1111b时）。



## 20.7.2 LIN从机模式

图20-42是LIN从机模式下的波特率生成结构图。

图20-42.LIN从机模式下波特率生成结构图



注 关于LIN通信的时钟源，请参考第4章 时钟发生电路。

设置LIN通信时钟源如下：

- LIN通信时钟源= $f_{CLK}/2^{\text{注}1}$
- 在4MHz至32MHz的范围内

注1 当不使用超时错误检测时，可选择 $f_{MX}$ 时钟作为LIN通信时钟源。在这种情况下，将LIN通信时钟源的频率至少设置为CPU/外围硬件时钟（ $f_{CLK}$ ）的1.2倍。

在LIN从机模式[自动波特率]下，可以用1kbps到20kbps的波特率进行操作。根据目标波特率设置预分频器时钟，使其频率对应列表中的相应值。

[目标波特率]	[预分频器的时钟频率]
1kbps到20kbps:	4MHz <sup>注</sup>
1kbps到小于2.4kbps:	4MHz
2.4kbps至20kbps:	8MHz至12MHz

注 将LWBRn寄存器中的NSPB[3:0]位设置为0011b（4次采样）。

表20-29是LIN从机模式[固定波特率]中每个LIN通信时钟源频率的波特率（19200、10417、9600和2400bps）和误差的生成示例。

**表20-29.LIN从机模式[固定波特率] 下波特率生成示例  
(19200bps、10417bps、9600bps和2400bps)**

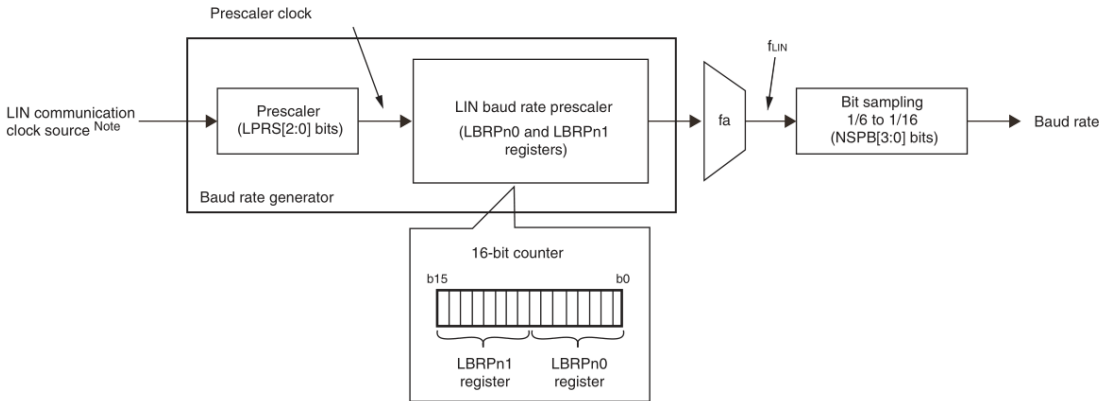
LIN 通信时钟源	预分频器	波特率发生器 0-1 N+1 分频器	波特率	误差
32MHz	1/1	104	19230.77	+0.16%
		192	10416.67	-0.003%
		208	9615.38	+0.16%
		833	2400.96	+0.04%
24MHz	1/1	78	19230.77	+0.16%
		144	10416.67	-0.003%
		156	9615.38	+0.16%
		625	2400	0%
16MHz	1/1	52	19230.77	+0.16%
		96	10416.67	-0.003%
		104	9615.38	+0.16%
		417	2398.08	-0.08%
12MHz	1/1	39	19230.77	+0.16%
		72	10416.67	-0.003%
		78	9615.38	+0.16%
		313	2396.17	-0.16%
8MHz	1/1	26	19230.77	+0.16%
		48	10416.67	-0.003%
		52	9615.38	+0.16%
		208	2403.85	+0.16%

备注 表中的数字是针对每一位的16个采样（例如当NSPB[3:0]=0000b或1111b时）。

### 20.7.3 UART模式

图20-43是UART模式下的波特率生成结构图。

图20-43.UART模式下波特率生成结构图



注释 关于LIN通信的时钟源，请参考第4章 时钟发生电路。

设置LIN通信时钟源如下。

- LIN通信时钟源= $f_{CLK}/2$ <sup>注1</sup>
- 在4MHz至32MHz的范围内

注1 可以将 $f_{MX}$ 选择为LIN通信的时钟源。在这种情况下，至少要将LIN通信时钟源的频率设置为CPU/外围硬件时钟 ( $f_{CLK}$ ) 的1.2倍。

表20-30.UART波特率设置示例（当LIN通信时钟源=32MHz时）

UART 波特率 (目标波特率)	预分频器	波特率发生器 0-1 N+1 分频器	波特率	误差
1200bps	1/2	833	1200.48	+0.04%
2400bps	1/2	417	2398.08	-0.08%
4800bps	1/2	208	4807.69	+0.16%
9600bps	1/2	104	9615.38	+0.16%
19200bps	1/2	52	19230.77	+0.16%
31250bps	1/2	32	31250.00	0.00%
38400bps	1/2	26	38461.54	+0.16%

注1.这些例子是针对每一位的16个采样（例如当NSPB[3:0]=0000b或1111b时）。

2.波特率可以通过以下表达式来计算。

$$\text{波特率} = (\text{LIN通信时钟源} (f_{CLK}/2 \text{ 或 } f_{MX}: \text{ 由LINnMCK位选择}) \text{ 频率}) \times (\text{由 LPRS[2:0]选择的分频器} + (\text{LBRPn0} + (0 \times 100 \times \text{LBRPn1}) + 1)) + \text{由 NSPB[3:0]bps 选择的数}$$

## 20.8 噪声滤波器

LIN/UART模块有一个噪声滤波器，用于减少由于噪声造成的数据错误接收。通过将LMDn寄存器中的LRDNFS位设置为0（使用噪声滤波器），激活噪声滤波器。噪声滤波器基于预分频器时钟对同步的LRXDn（n=0）电平进行采样，并根据三个采样电平中的多数输出。收到的数据中每个位的值由噪声滤波器的输出决定。

图20-44是噪声滤波器的配置，图20-45是噪声滤波器电路的示例，图20-46使用噪声滤波器时对接收数据的确认

图20-44.噪声滤波器的配置

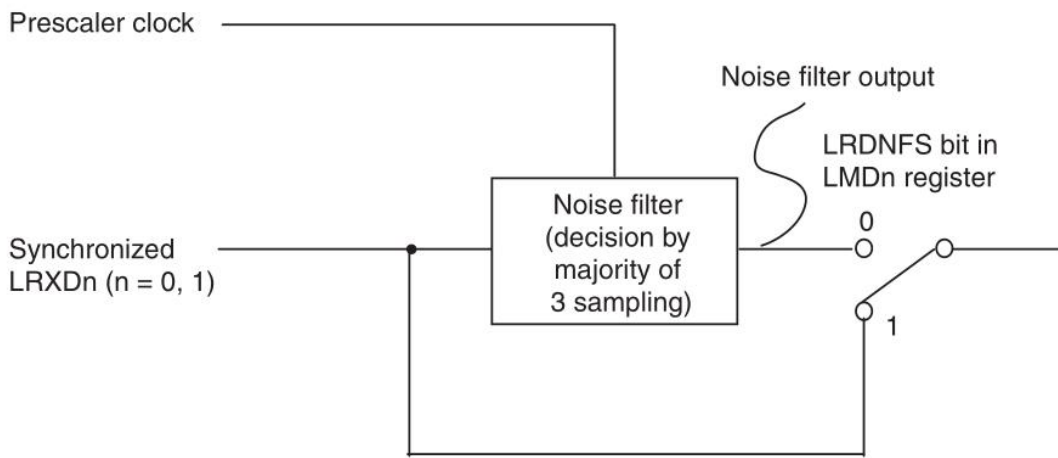


图20-45.噪声滤波器的电路

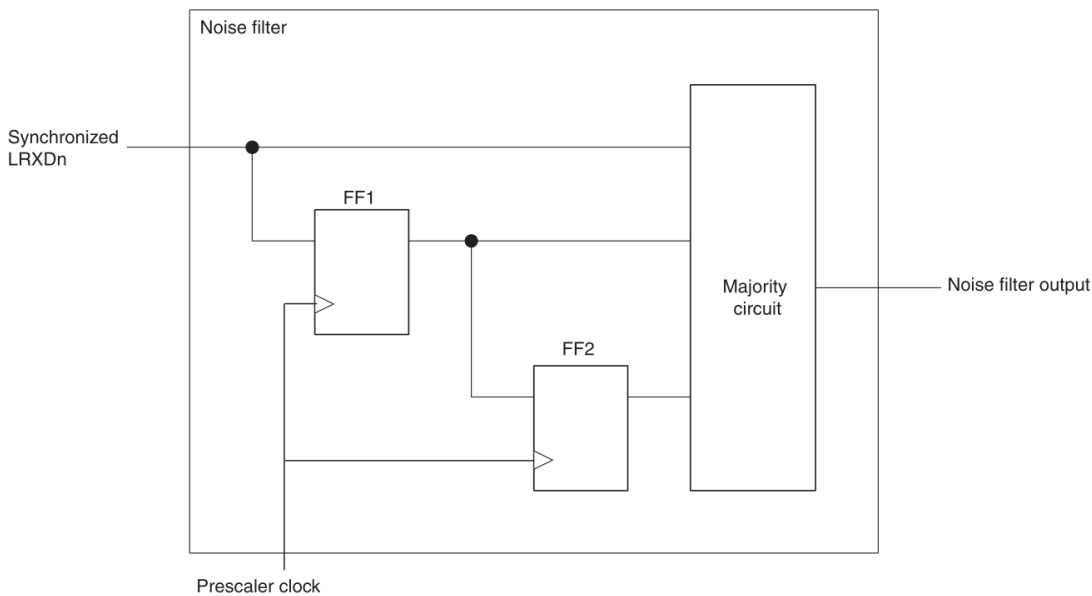
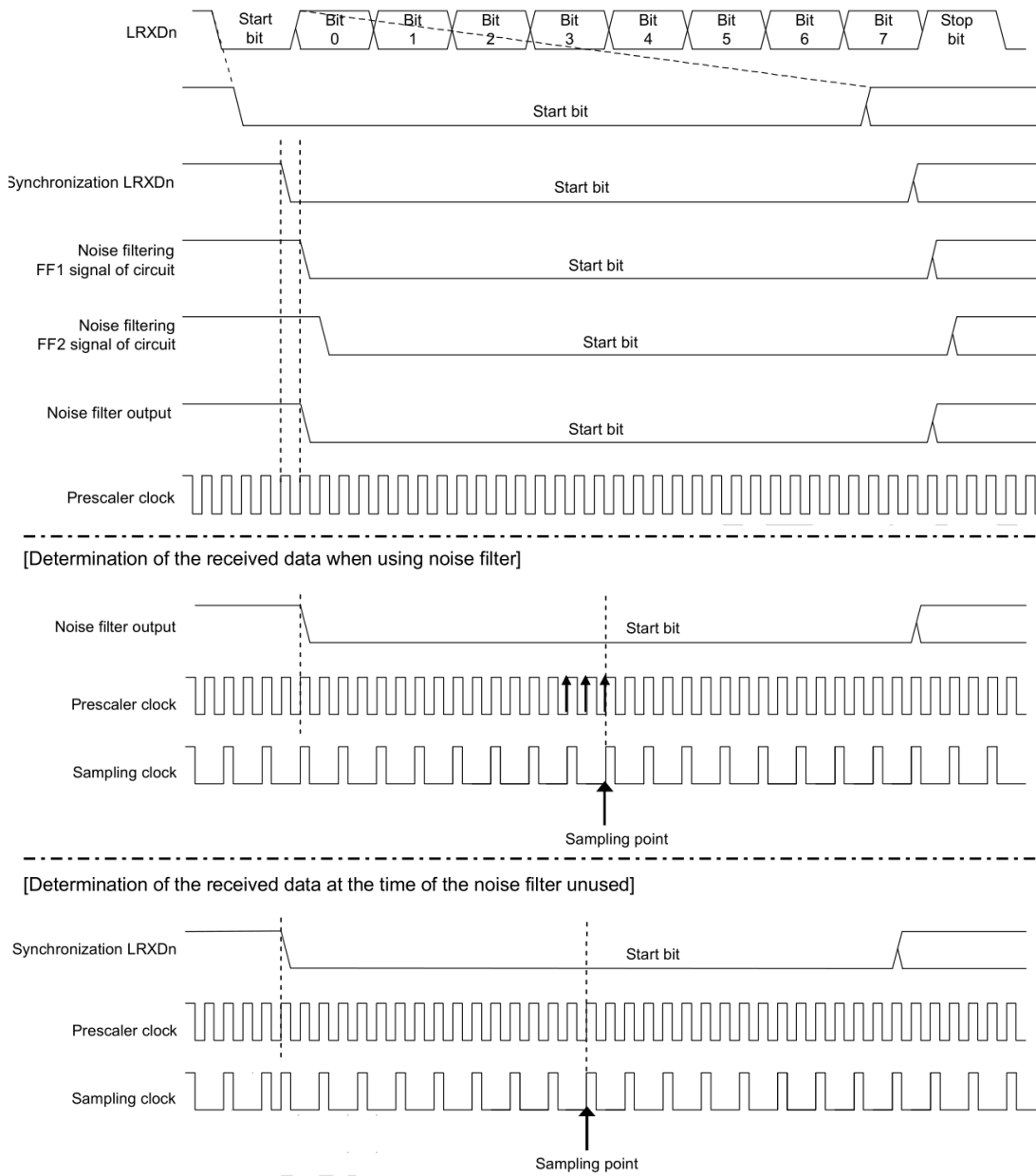


图20-46.使用噪声滤波器时接收数据的确定



## 20.9 中断

LIN/UART模块会产生四种类型的中断请求。

- LINn成功传输中断
- LINn成功接收中断
- LINn接收状态中断
- LINn中断

将LMDn寄存器中的LIOS位设置为0，可以对所有的中断源进行逻辑OR操作，输出LINn中断的中断请求。

将LMDn寄存器中的LIOS位设置为1，可以根据中断源，输出LINn成功传输中断、LINn成功接收中断或LINn接收状态中断。表20-31列出了每个中断的来源。

表20-31.中断源

模式		LMDn 寄存器中的 LIOS 位为 0	LMDn 寄存器中的 LIOS 位为 1 <sup>注</sup>		
		LINn 中断	LINn 成功的传输中断	LINn 成功的接收中断	LINn 接收状态中断
LIN 模式	LIN 主机模式	<ul style="list-style-type: none"> <li>•成功的帧传输</li> <li>•成功的帧接收</li> <li>•成功的唤醒传输</li> <li>•成功的唤醒接收</li> <li>•成功的帧头传输</li> <li>•位错误</li> <li>•物理总线错误</li> <li>•帧/响应超时错误</li> <li>•帧错误</li> <li>•校验和错误</li> <li>•响应准备错误</li> </ul>	<ul style="list-style-type: none"> <li>•成功的帧传输</li> <li>•成功的唤醒传输</li> <li>•成功的帧头传输</li> </ul>	<ul style="list-style-type: none"> <li>•成功的帧接收</li> <li>•成功的唤醒接收</li> </ul>	<ul style="list-style-type: none"> <li>•位错误</li> <li>•物理总线错误</li> <li>•帧/响应超时错误</li> <li>•帧错误</li> <li>•校验和错误</li> <li>•响应准备错误</li> </ul>
	LIN 从机模式	<ul style="list-style-type: none"> <li>•成功的响应传输</li> <li>•成功的响应接收</li> <li>•成功的唤醒传输</li> <li>•成功的唤醒接收</li> <li>•成功的帧头接收</li> <li>•位错误</li> <li>•帧/响应超时错误</li> <li>•帧错误</li> <li>•同步场错误</li> <li>•校验码错误</li> <li>•ID 奇偶性错误</li> <li>•响应准备错误</li> </ul>	<ul style="list-style-type: none"> <li>•成功的响应传输</li> <li>•成功唤醒传输</li> </ul>	<ul style="list-style-type: none"> <li>•成功的响应接收</li> <li>•成功的唤醒接收</li> <li>•成功的帧头接收</li> </ul>	<ul style="list-style-type: none"> <li>•位错误</li> <li>•帧/响应超时错误</li> <li>•帧错误</li> <li>•同步场错误</li> <li>•校验和错误</li> <li>•ID 奇偶性错误</li> <li>•响应准备错误</li> </ul>
UART 模式		-	<ul style="list-style-type: none"> <li>•传输开始/成功传输</li> </ul>	<ul style="list-style-type: none"> <li>•成功的接收</li> <li>•扩展位不匹配</li> </ul>	<ul style="list-style-type: none"> <li>•位错误</li> <li>•过载运行错误</li> <li>•帧错误</li> <li>•扩展位检测</li> <li>•ID 匹配</li> <li>•奇偶校验错误</li> </ul>

注 LIOS位的设置在LIN模式下允许。在UART模式下不需要设置LIOS位。

当LIE n寄存器中的相应的位为1（中断允许）且LSTn寄存器中的相应的标志为1时，将输出每个中断请求。

# 第21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。

## 21.1 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21-1所示。

图 21-1 IrDA与SCI的合作示意框图

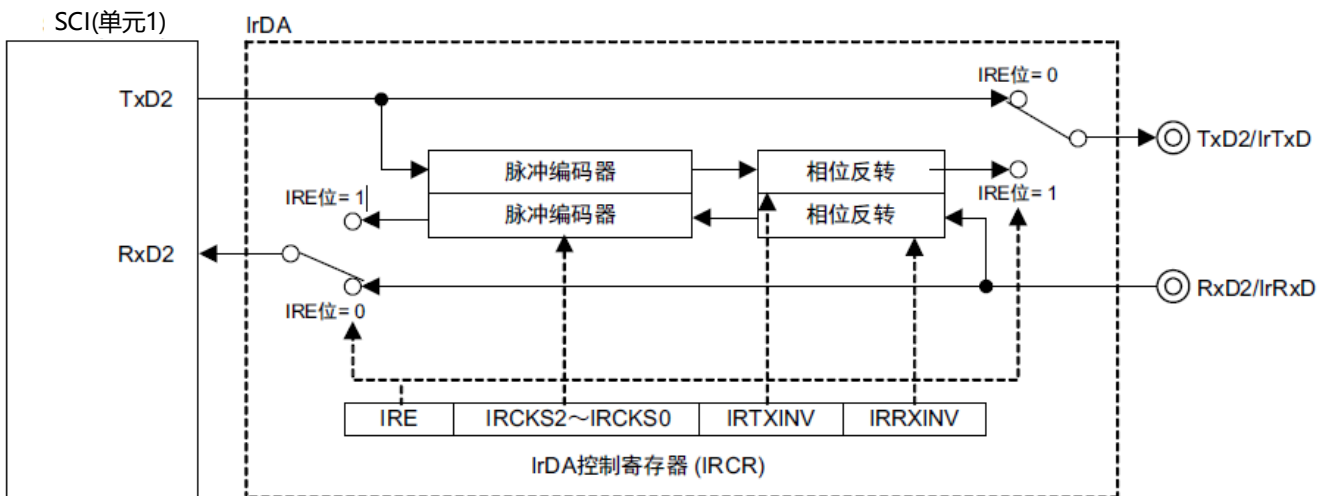


表 21-1 IrDA的引脚结构

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

## 21.2 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)

### 21.2.1 外围允许寄存器0(PER0)

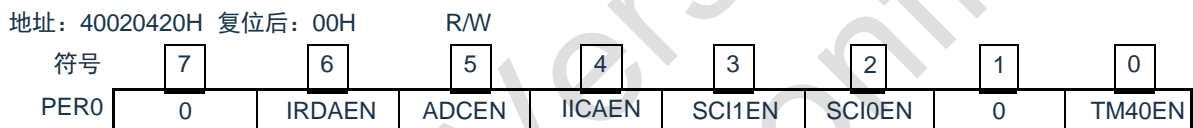
PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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21-2 外围允许寄存器0(PER0)的格式



IRDAEN	IrDA的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> <li>• 不能写 IrDA 使用的 SFR。</li> <li>• IrDA处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>• 能读写 IrDA 使用的 SFR。</li> </ul>

注意 1. 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。



## 21.2.2 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21-3 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 <sub>CLK</sub> /2
0	1	0	F <sub>CLK</sub> /4
0	1	1	F <sub>CLK</sub> /8
1	0	0	F <sub>CLK</sub> /16
1	0	1	F <sub>CLK</sub> /32
1	1	0	F <sub>CLK</sub> /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。

## 21.3 IrDA的运行

### 21.3.1 IrDA通信的操作步骤

#### 1. (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) 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”。

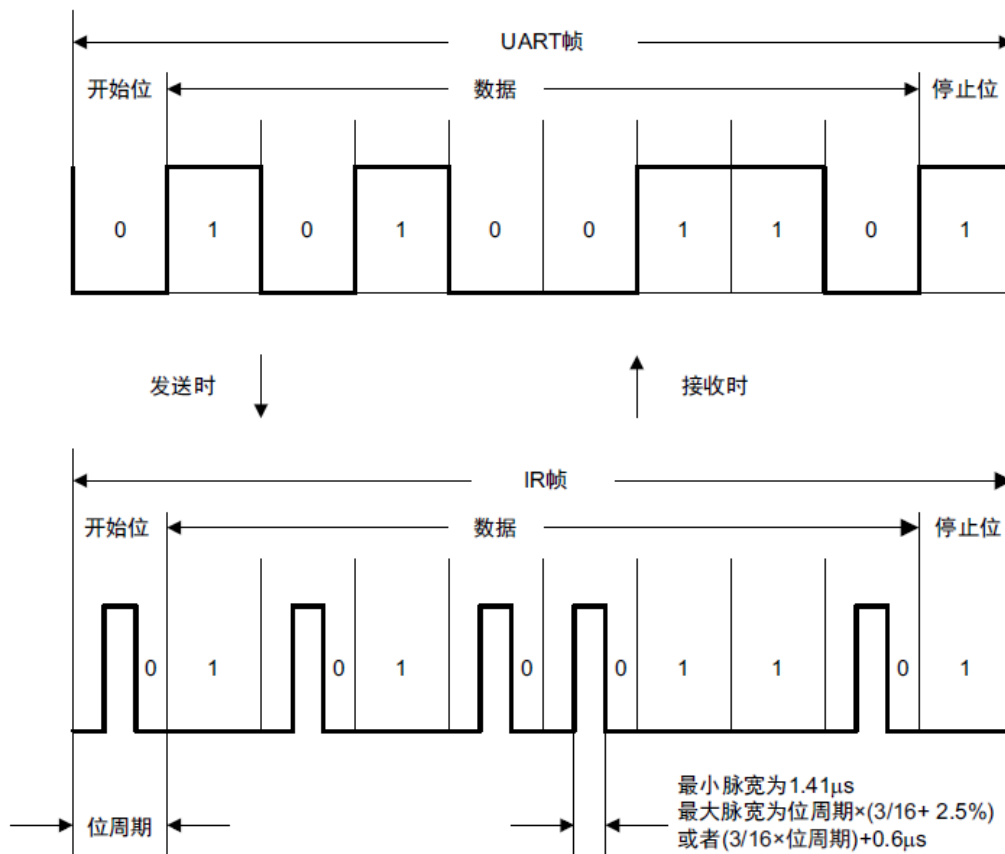
### 21.3.2 发送

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) $\times 3/16$ 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 $(3/16+2.5\%) \times$ 位周期，或者 $(3/16 \times \text{位周期}) + 0.6\mu\text{s}$ 。

在CPU或者外围硬件时钟( $F_{\text{CLK}}$ )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21-4 IrDA的发送/接收运行图



### 21.3.3 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。

## 21.3.4 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21-2IRCKS2~IRCKS0位的设置值

F <sub>CLK</sub> [MHz]	项目	<上段>位速率[kbps]					
		<下段>位速率 $\times 3/16$ [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。

2.脉宽不能小于位速率  $\times 3/16$ 。

## 21.4 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22章 增强型DMA

### 22.1 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22-1所示。

表22-1 DMA的规格(1/2)

项目		规格
启动源		最大38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22-5 DMA启动源和向量地址”。

表22-1 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"> <li>•将DMAENi0~DMAENi7位置“0”（禁止启动）。</li> <li>•当DMACTj寄存器从“1”变为“0”的数据传送结束时</li> </ul>
	重复模式	<ul style="list-style-type: none"> <li>•将DMAENi0~DMAENi7位置“0”（禁止启动）。</li> <li>•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</li> </ul>

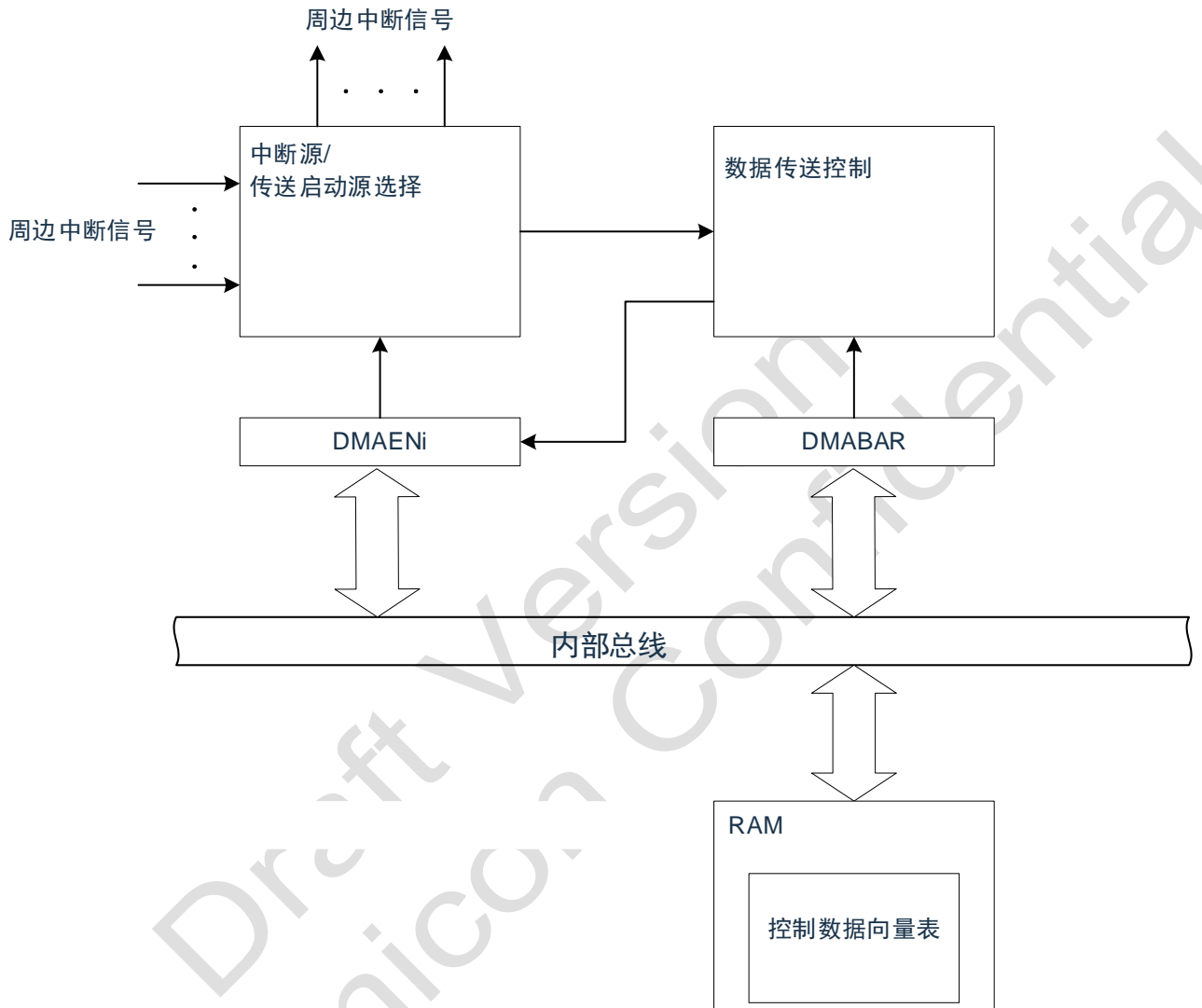
注 在深度睡眠模式，因为闪存停止运行，所以不能作为DMA传送源。

备注 i=0~4、j=0~39

## 22.2 DMA的结构

DMA的框图如图22-1所示

图22-1 DMA的框图



## 22.3 控制DMA的寄存器

控制DMA的寄存器如表22-2所示。

表22-2 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22-3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22-3 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

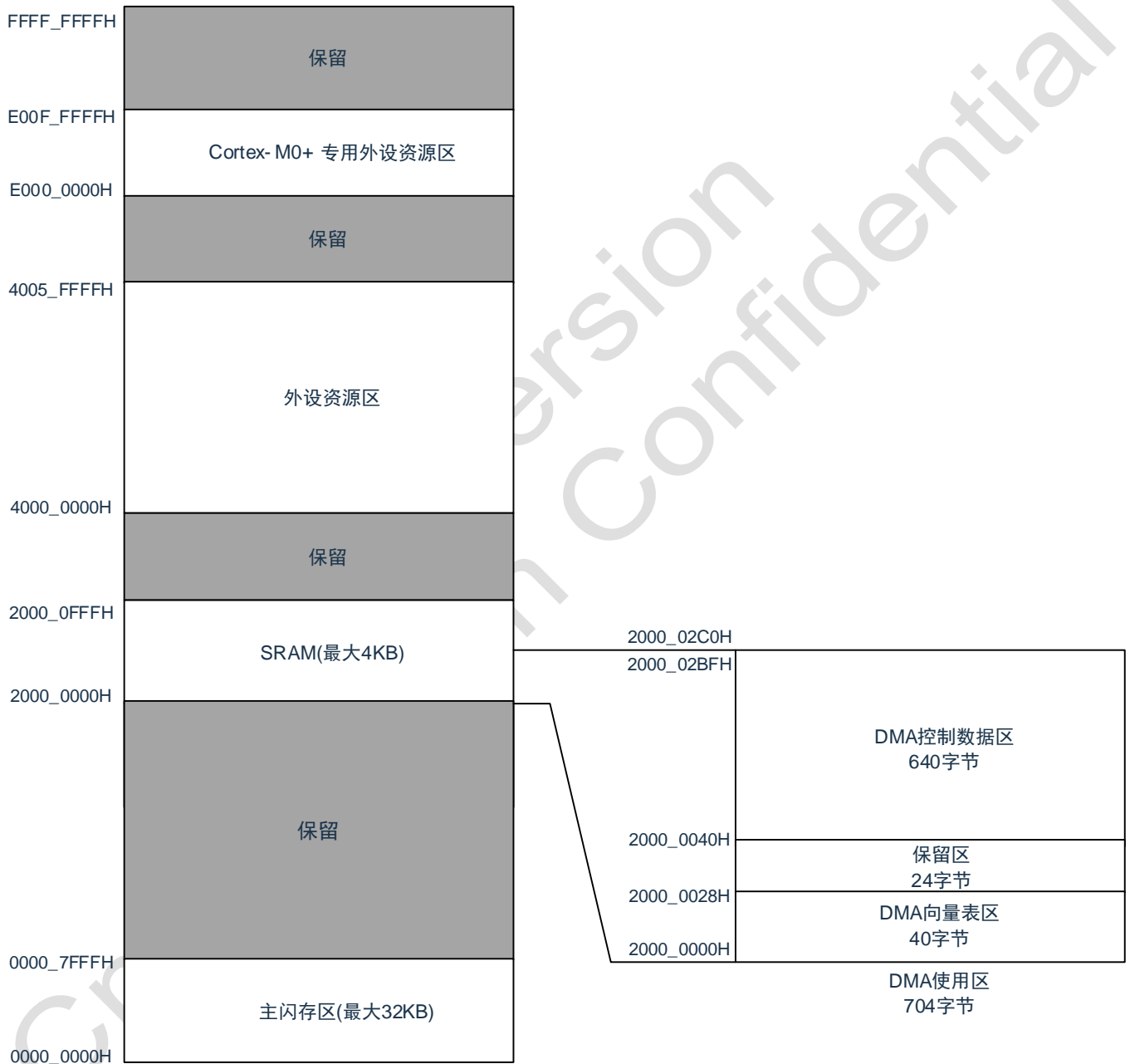
备注 j=0~39



### 22.3.1 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22-2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22-2 DMABAR寄存器的设置值为“20000000H”时的存储器映像例子



### 22.3.2 控制数据的分配

从起始地址开始，按照DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>、DMDAR<sub>j</sub> (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22-3所示。

注意：1. 必须在对应的DMAEN<sub>i</sub> (i=0~4) 的DMAEN<sub>i0</sub>~DMAEN<sub>i7</sub>位为“0”（禁止启动）时更改DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>、DMDAR<sub>j</sub>寄存器的数据。

2. 不能通过DMA传送进行DMACR<sub>j</sub>、DMBLS<sub>j</sub>、DMACT<sub>j</sub>、DMRLD<sub>j</sub>、DMSAR<sub>j</sub>和DMDAR<sub>j</sub>的存取。

图22-3 控制数据的分配(DMABAR设置为2000000H)

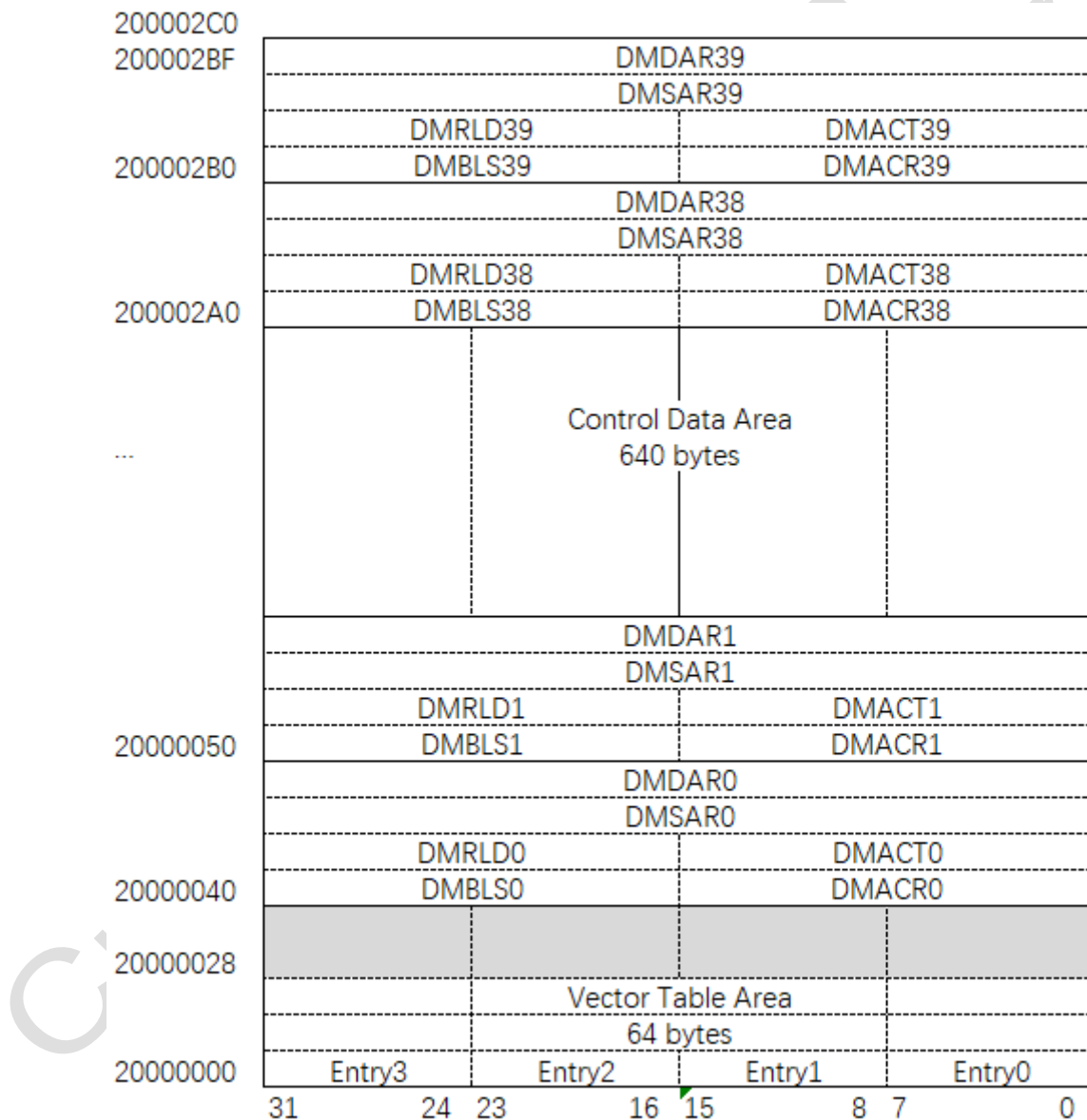


表22-4 控制数据的起始地址

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

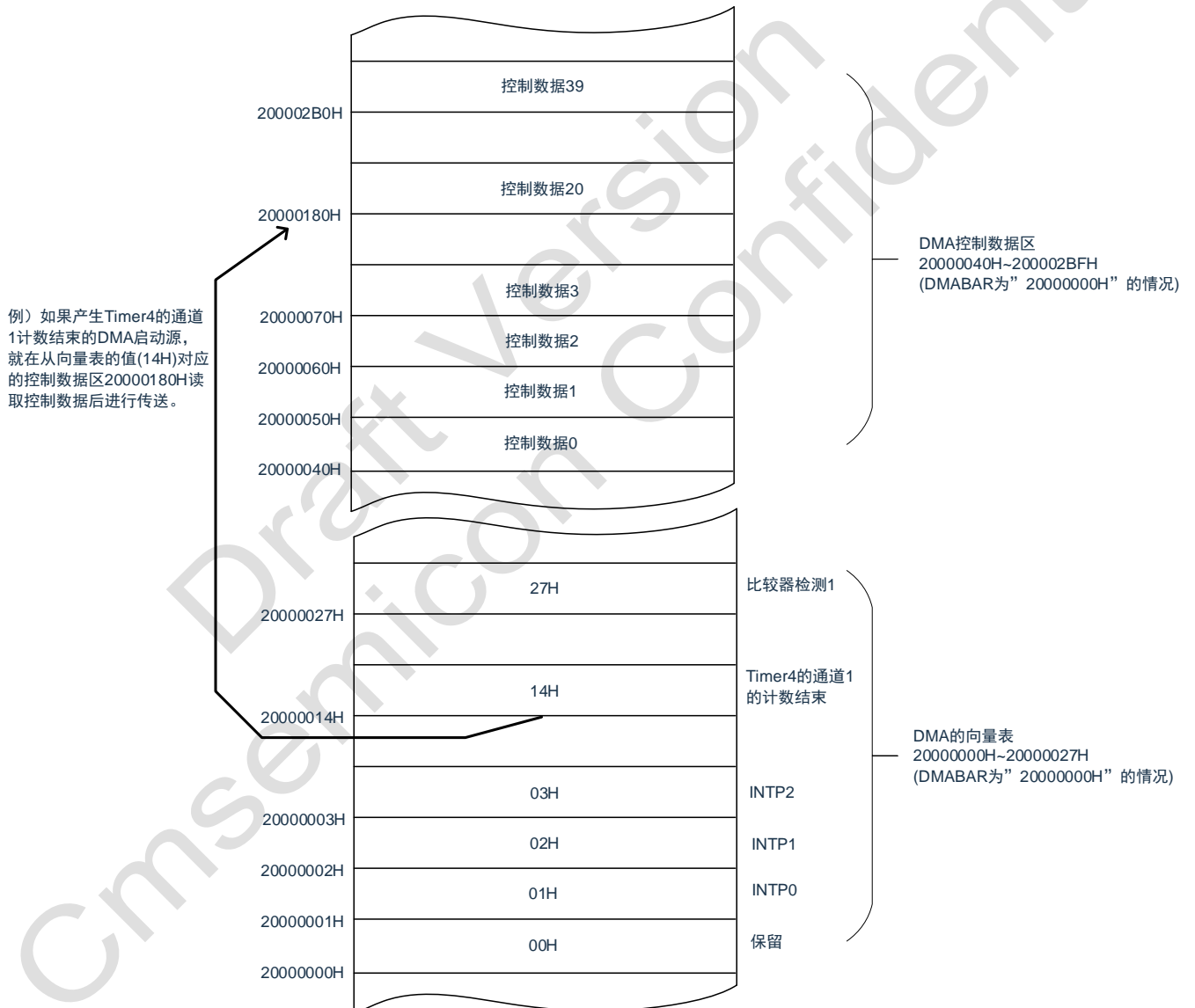
### 22.3.3 向量表

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22-5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22-4 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）





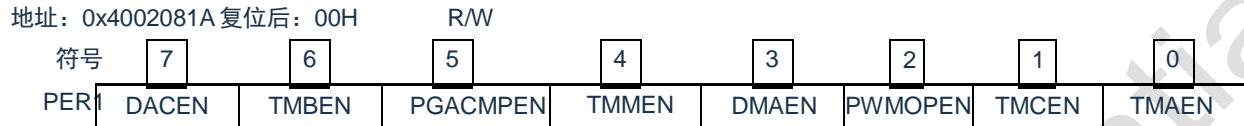
### 22.3.4 外围允许寄存器1 (PER1)

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22-5 外围允许寄存器1 (PER1) 的格式



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。

## 22.3.5 DMA控制寄存器j (DMACRj) (j=0~39)

DMACRj寄存器控制DMA的运行模式。

图22-6 DMA控制寄存器j (DMACRj) 的格式

地址：参照“22.3.2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	FIFO
	7	6	5	4	3	2	1	0
	SZ		RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

FIFO	FIFO块传输控制
0	不是FIFO块传输
1	是FIFO块传输，源地址 (SAMOD=0) 或目的地址 (DAMOD=0) 绝对固定

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。

## 22.3.6 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22-7 DMA块大小寄存器j (DMBLSj) 的格式

地址：参照“22.3.2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 1.不能通过DMA传送进行DMBLSj寄存器的存取。



### 22.3.7 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22-8DMA传送次数寄存器j (DMACTj) 的格式

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

地址: 参照控制数据的分配”。

复位后: 不定值

R/W

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 1.不能通过DMA传送进行DMACTj寄存器的存取。

### 22.3.8 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22-9 DMA传送次数重加载寄存器j (DMRLDj) 的格式

地址：参照“22.3.2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 1.不能通过DMA传送进行DMRLDj寄存器的存取。

### 22.3.9 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22-10 DMA源地址寄存器j (DMSARj) 的格式

地址：参照“22.3.2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1.不能通过DMA传送进行DMSARj寄存器的存取。

## 22.3.10 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22-11 DMA目标地址寄存器j (DMDARj) 的格式

地址：参照“22.3.2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	DMDARj3 1	DMDARj3 0	DMDARj2 9	DMDARj2 8	DMDARj2 7	DMDARj2 6	DMDARj2 5	DMDARj2 4
	23	22	21	20	19	18	17	16
	DMDARj2 3	DMDARj2 2	DMDARj2 1	DMDARj2 0	DMDARj1 9	DMDARj1 8	DMDARj1 7	DMDARj1 6
	15	14	13	12	11	10	9	8
	DMDARj1 5	DMDARj1 4	DMDARj1 3	DMDARj1 2	DMDARj1 1	DMDARj1 0	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。

### 22.3.11 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22-6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。

3.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22-12 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	DMAENi7	DMAENi6	DMAENi5	DMAENi4	DMAENi3	DMAENi2	DMAENi1	DMAENi0

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22-6 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	LIN发送完成	LIN接收完成	UART0发送的传送结束/SSPI00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	A/D转换结束	保留	INTP7
DMAEN2	保留	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	高速SPI0通信结束	IICA0通信结束	UART2发送的传送结束/SSPI20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	TimerM的计数器1下溢	Flash擦除/写入结束
DMAEN4	比较器检测1	比较器检测0	TimerA的下溢	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

备注 i=0~4

### 22.3.12 DMAENi置位寄存器 (DMSETi)

这是 DMA 启动允许寄存器 DMAENi 的置位寄存器，设置相应的位为 1 则可以将 DMAENi 的相应位置位为 1。

图22-13 DMAENi置位寄存器 (DMSETi) (i=0~4) 的格式

地址: 40005018H (DMSET0)、40005019H (DMSET1)、  
4000501AH (DMSET2)、4000501BH (DMSET3)、  
4000501CH (DMSET4)

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DMSETi	DMSETi7	DMSETi6	DMSETi5	DMSETi4	DMSETi3	DMSETi2	DMSETi1	DMSETi0

DMSETin	DMAENin的置位寄存器
0	不产生置位。
1	将DMCENi的位n置位为1

注: i=0~4, n=0~7

### 22.3.13 DMAENi复位寄存器 (DMCLRi)

这是 DMA 启动允许寄存器 DMAENi 的复位寄存器，设置相应的位为 1 则可以将 DMAENi 的相应位复位为 0。

图22-14 DMAENi复位寄存器 (DMCLRi) (i=0~4) 的格式

地址: 40005020H (DMCLR0)、40005021H (DMCLR1)、  
40005022H (DMCLR2)、40005023H (DMCLR3)、  
40005024H (DMCLR4)

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DMCLRi	DMCLRi7	DMCLRi6	DMCLRi5	DMCLRi4	DMCLRi3	DMCLRi2	DMCLRi1	DMCLRi0

DMSETin	DMAENin的复位寄存器
0	不产生复位。
1	将DMCENi的位n复位为0

注: i=0~4, n=0~7

### 22.3.14 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。
- 3.不能通过DMA传送进行DMABAR寄存器的存取。
- 4.有关DMA控制数据区和DMA向量表区的分配，请参照“22.3.1DMA控制数据区和DMA向量表区的分配”的注意。
- 5.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。
- 6.该寄存器只能WORD访问，BYTE和HALFWORD访问忽略。

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

## 22.4 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。

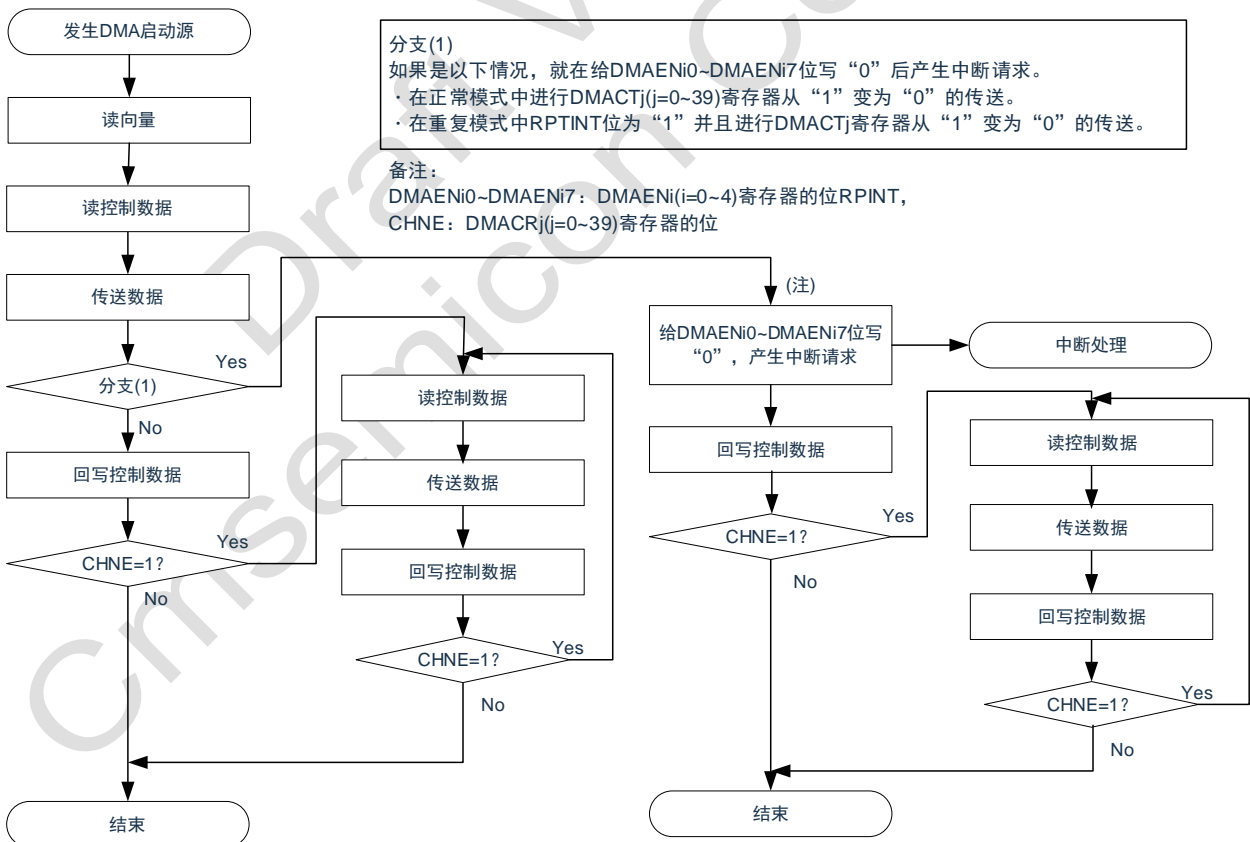
### 22.4.1 启动源

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）的设置为下述两种情况时，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22-16所示。

图22-16 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。



## 22.4.2 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22-7和图22-17所示。

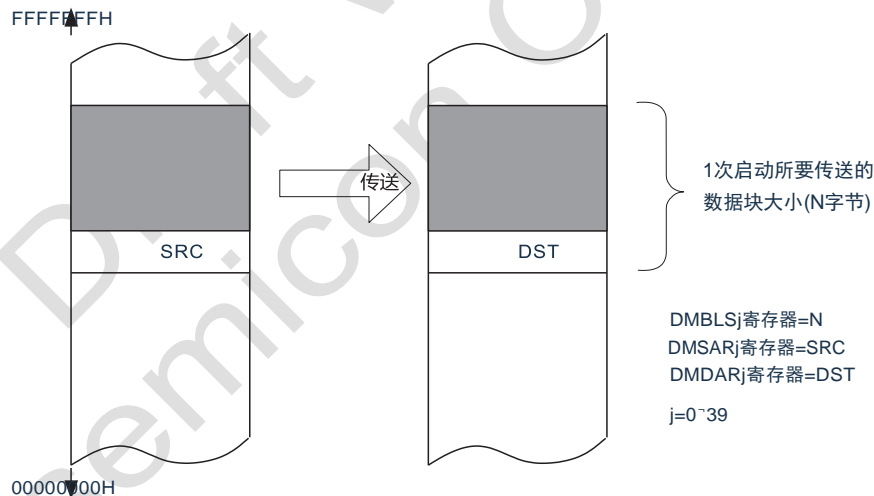
表22-7 正常模式的寄存器功能

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22-17 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

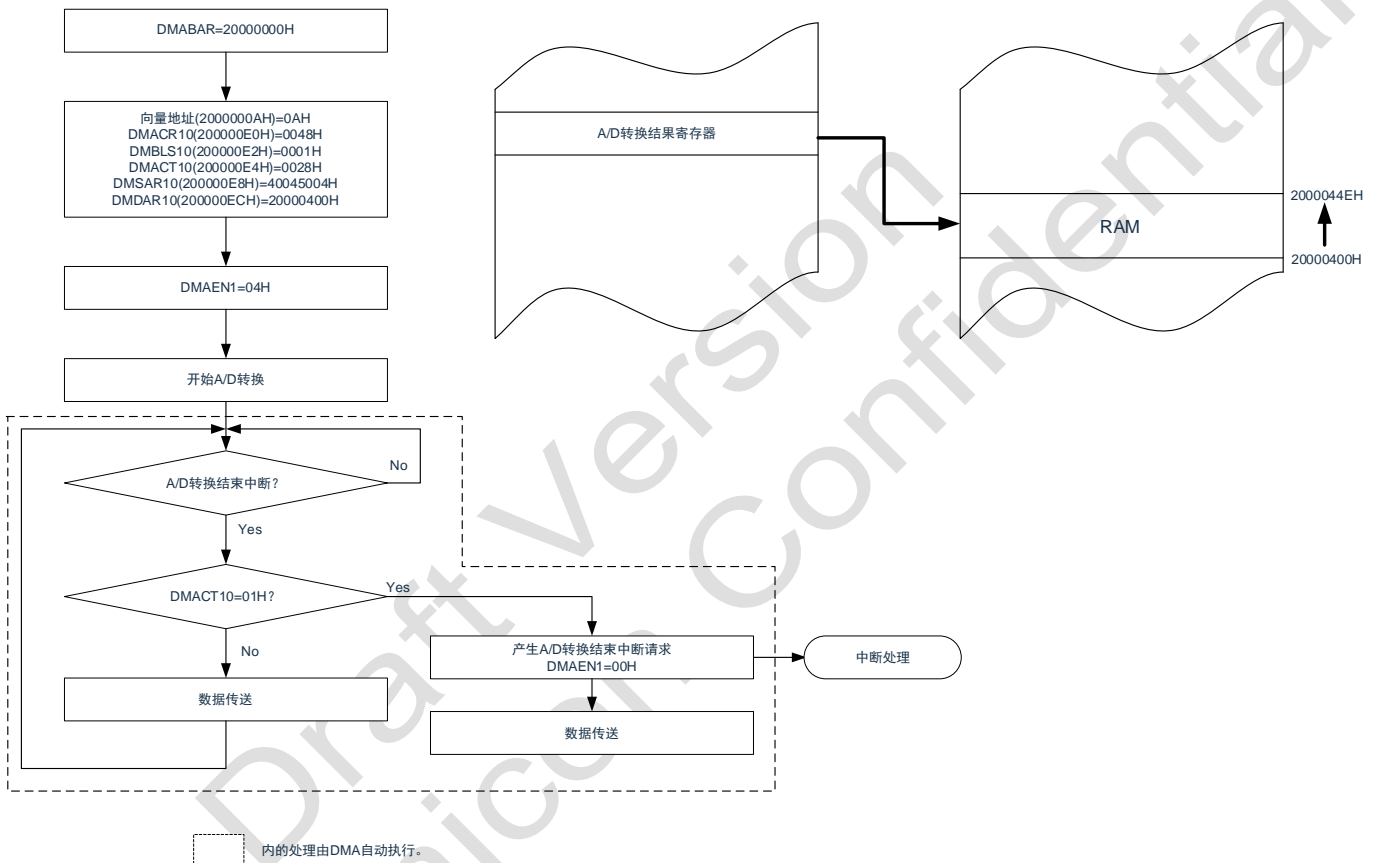
X: “0”或者“1”

(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22-18 正常模式的使用例子1：连续取A/D转换结果



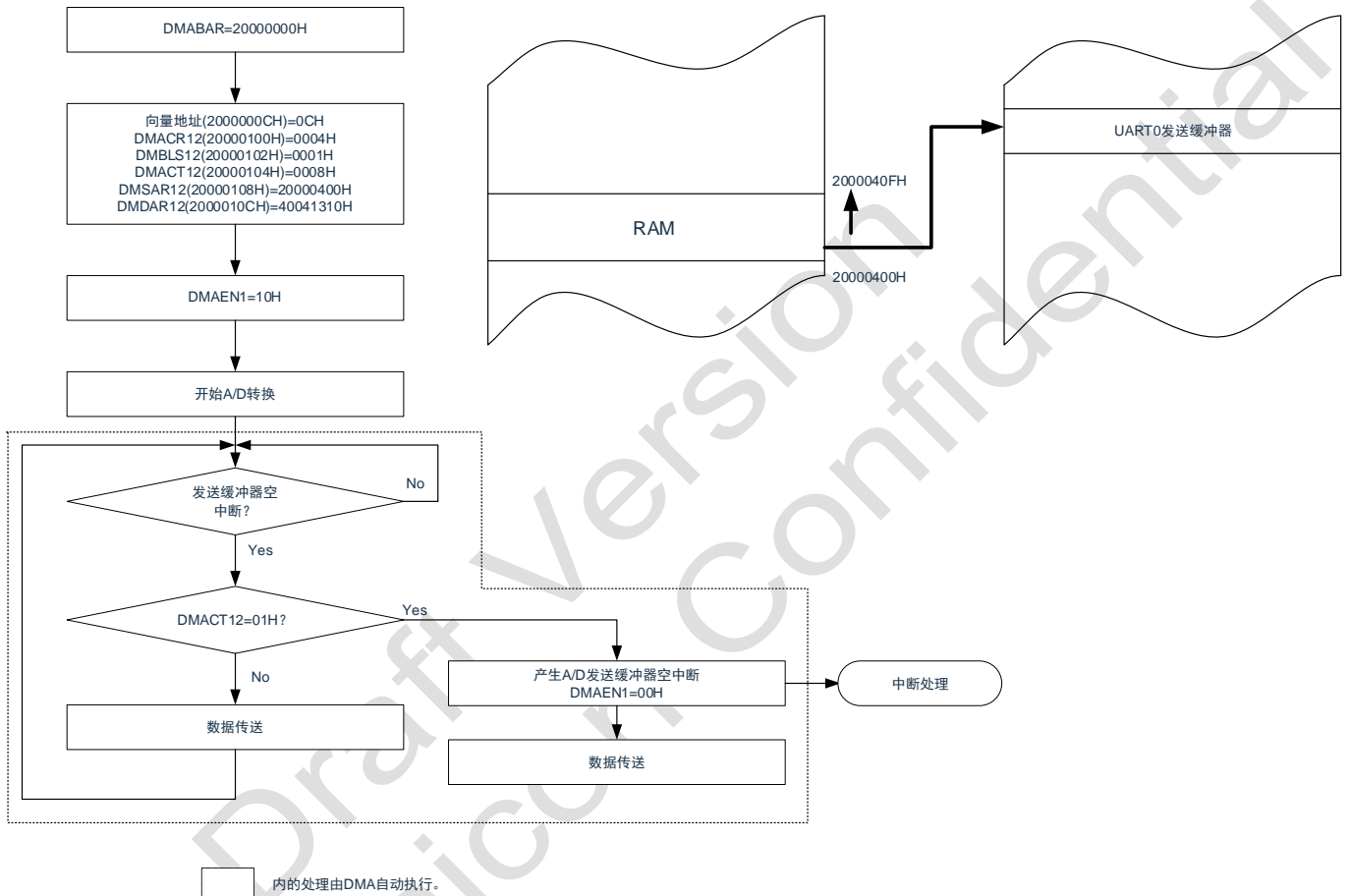
因为是正常模式，所以不使用DMRLD10寄存器的值。但是，当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须对DMRLD10寄存器进行初始化（0000H）。

(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22-19 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。

### 22.4.3 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”。

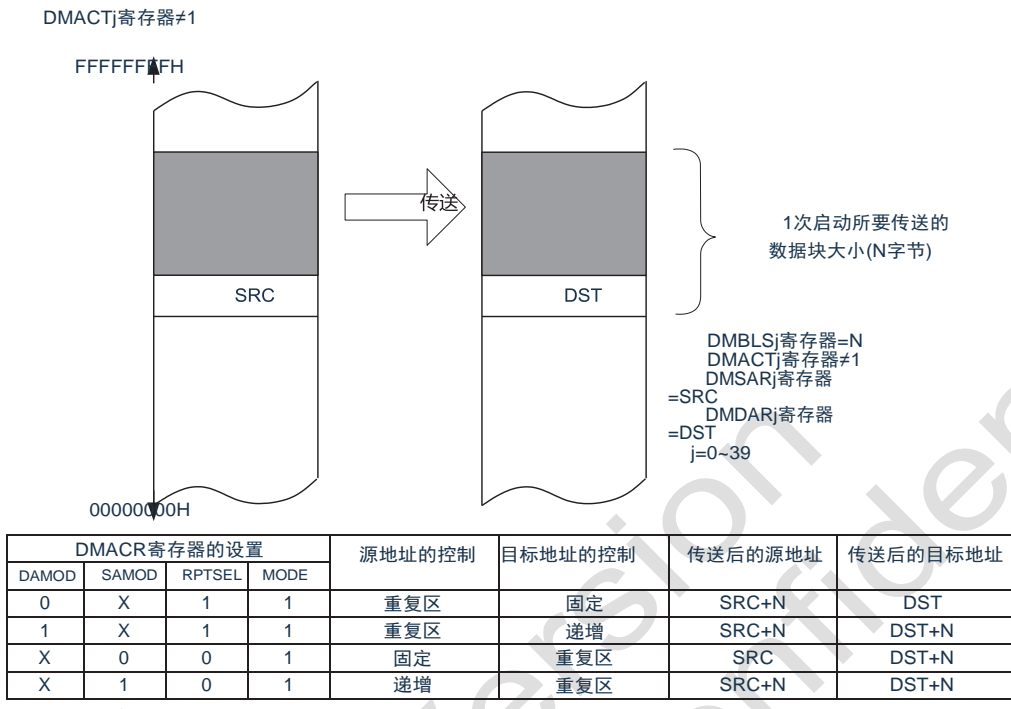
重复模式的寄存器功能和数据传送分别如表22-8和图22-20所示。

表22-8 重复模式的寄存器功能

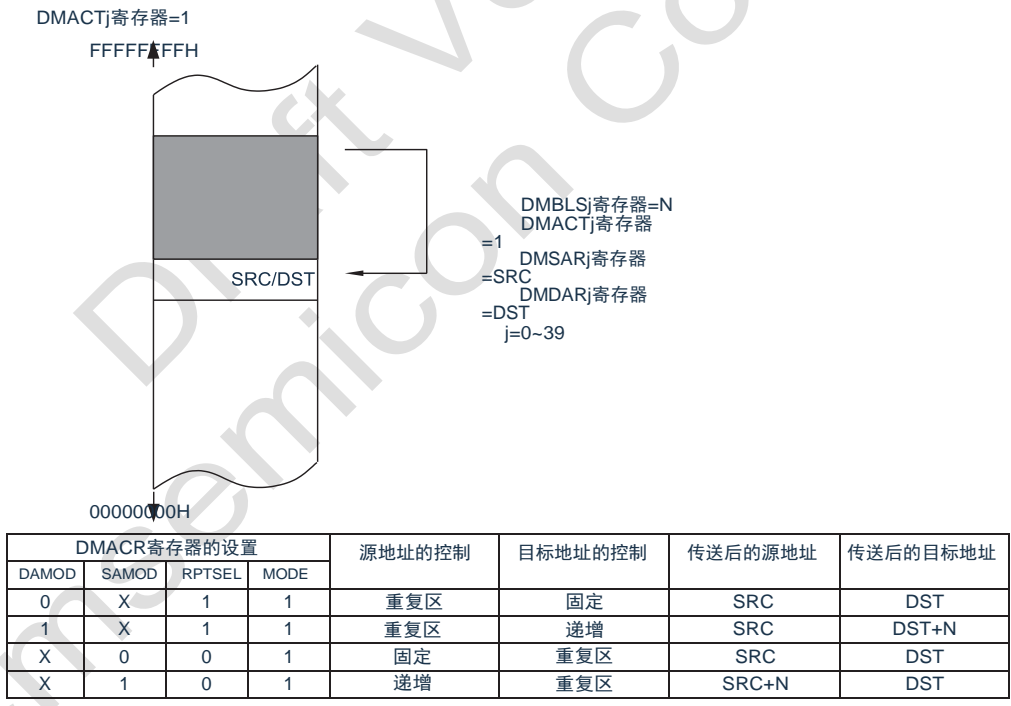
寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	将此寄存器的值重新加载到DMACT寄存器。 (对数据的传送次数进行初始化)
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

备注 j=0~39

图22-20 重复模式的数据传送



X: "0"或者"1"



SRC0: 源地址的初始值  
DST0: 目标地址的初始值  
X: "0"或者"1"

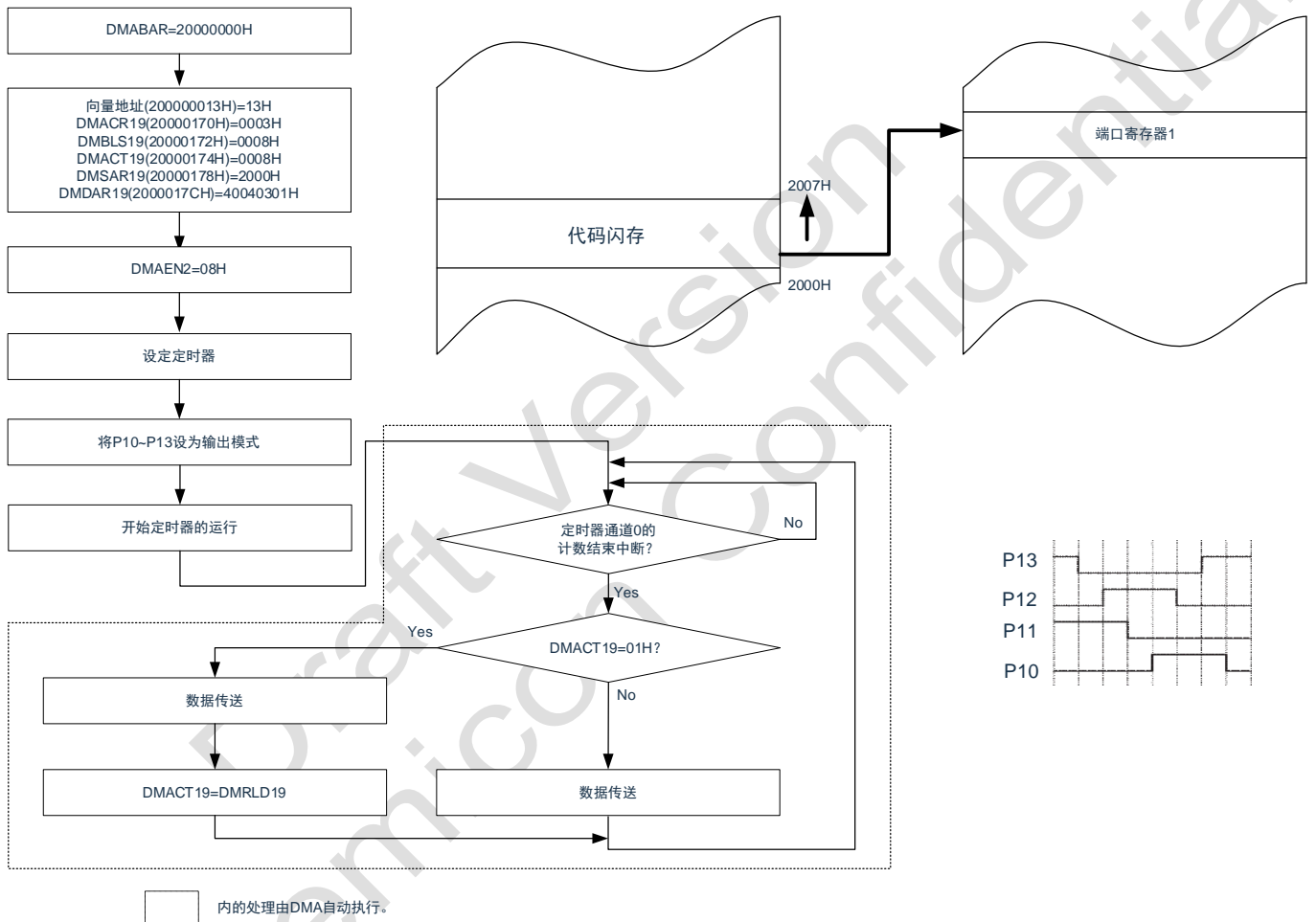
注意1.在使用重复模式时, 必须将重复区的数据长度设置在65535字节以内。

(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22-21 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。

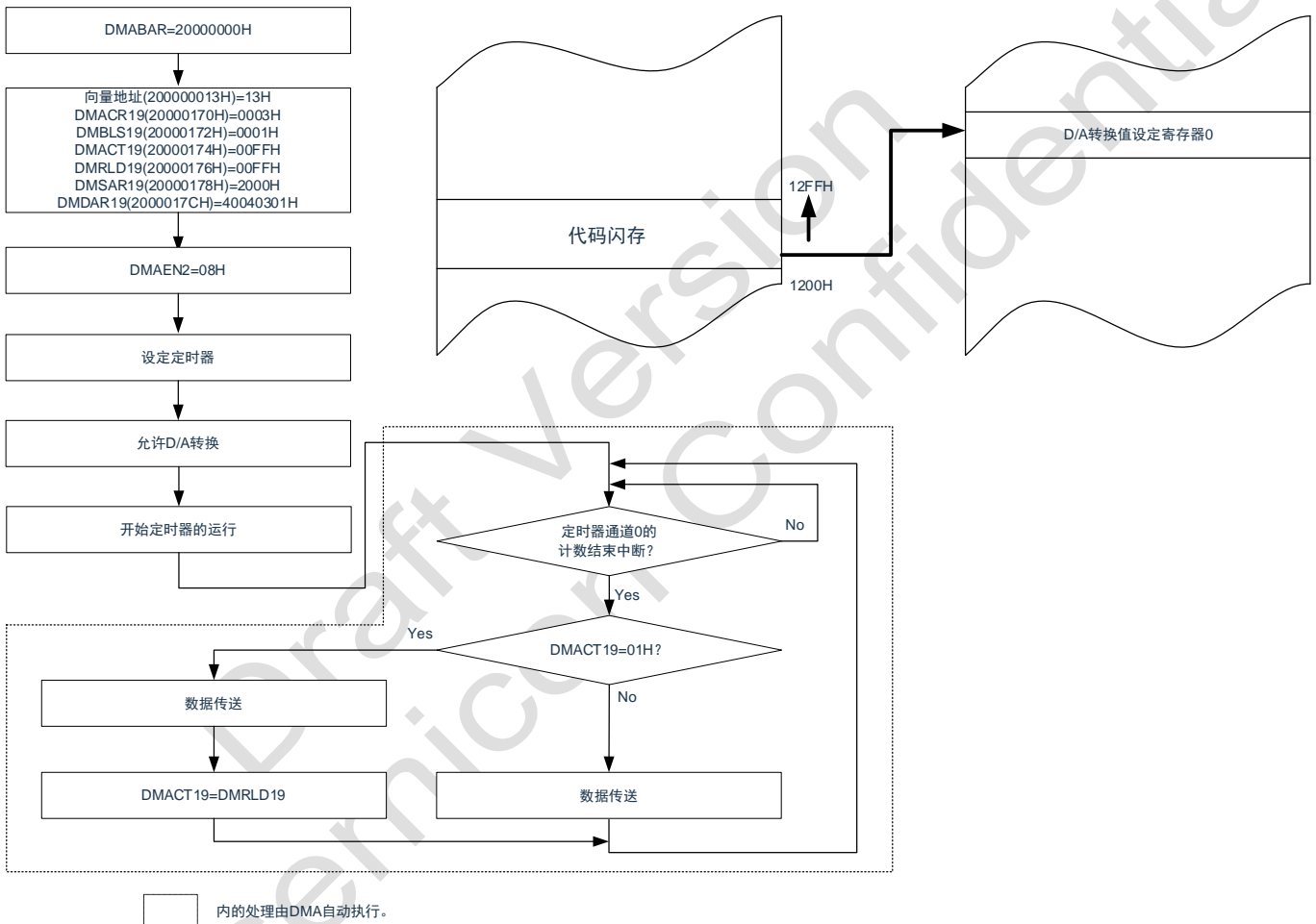
(2) 重复模式的使用例子2：使用12位D/A转换器的正弦波输出

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12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22-22 重复模式的使用例子2：使用12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。

### 22.4.4 链传送

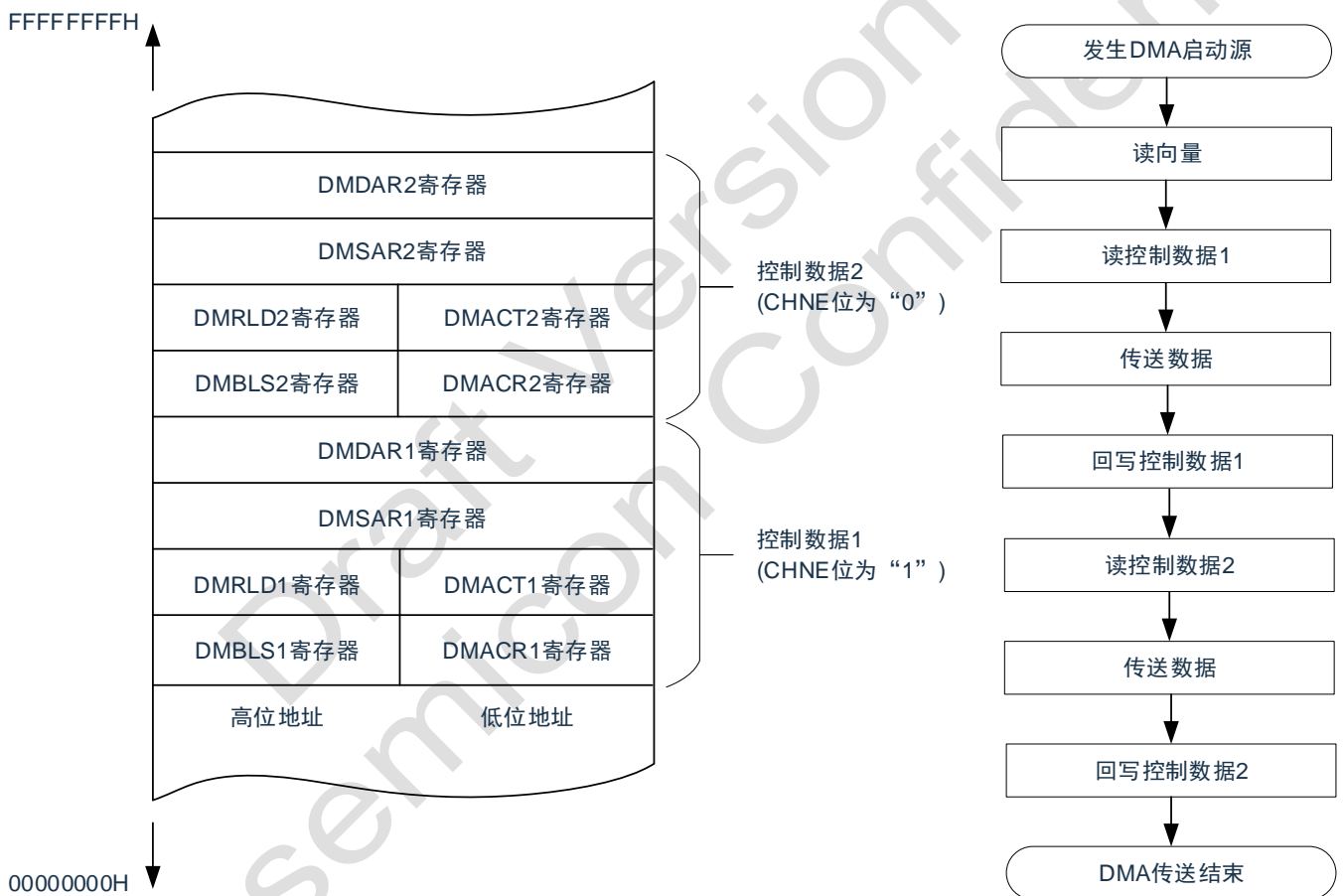
当DMACRj (j=0~39) 寄存器的CHNE位为“1” (允许链传送) 时, 能通过1个启动源连续进行多个数据的传送。

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22-23所示。

图22-23 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。

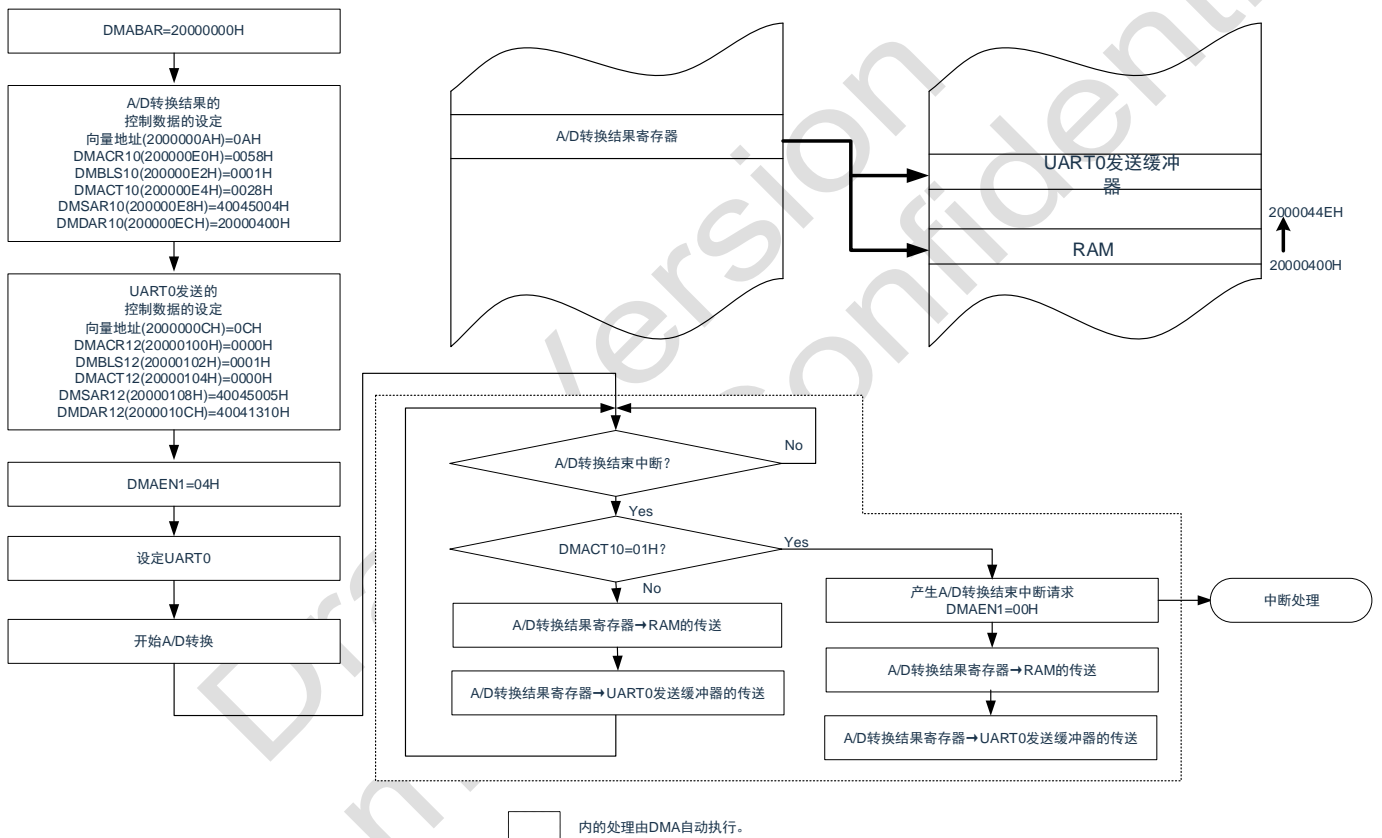


(1) 链传送的使用例子：连续取A/D转换结果进行UART0发送

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22-24 链传送的使用例子：连续取A/D转换结果进行UART0发送



## 22.5 使用DMA时的注意事项

### 22.5.1 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。

### 22.5.2 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）。

## 22.5.3 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22-9所示。

表22-9 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22-10 回写控制数据所需的时钟数”。  
 2.有关读写数据所需的时钟数，请参照“表22-11 读写数据所需的时钟数”。

表22-10 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	固定	重复区	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=0~39, X: “0”或者“1”

表22-11 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+等待数注
写数据	1	—	—	1	1	1+等待数注

## 22.5.4 DMA的响应时间

DMA响应时间如表22-12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22-12 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)

## 22.5.5 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“21.3.3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)

## 22.5.6 待机模式中的运行

状态	DMA运行
睡眠模式	能运行（禁止在低功耗模式中运行）。
深度睡眠模式	能接受DMA启动源，并进行DMA传送 <sup>注1</sup>

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。

Draft Version  
Cmsemicon Confidential

## 第23章 联动控制器(EVENTC)

### 23.1 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。

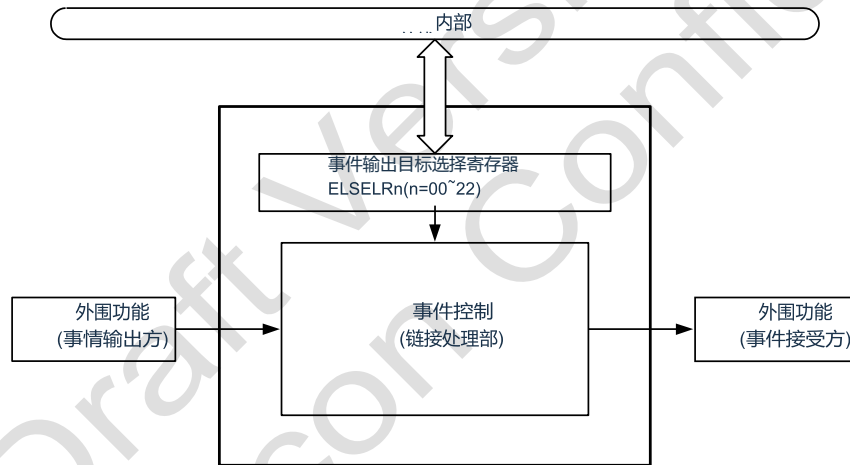
EVENTC有以下功能：

- 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。

### 23.2 EVENTC的结构

EVENTC的框图如图23-1所示。

图23-1 EVENTC的框图



## 23.3 控制寄存器

控制器寄存器如表23-1所示。

表23-1 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
事件输出目标选择寄存器22	ELSELR22

### 23.3.1 输出目标选择寄存器n (ELSELRn) (n=00~22)

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=00~22）和外围功能的对应如表23-2所示，ELSELRn寄存器（n=00~22）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23-3所示。

图23-2 事件输出目标选择寄存器n (ELSELRn) 的格式

地址：40043400H (ELSELR00) ~ 40043416H (ELSELR22) 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 <sup>注1</sup>	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 <sup>注1</sup> 。
0	0	1	0	选择所链接的外围功能2的运行 <sup>注1</sup> 。
0	0	1	1	选择所链接的外围功能3的运行 <sup>注1</sup> 。
0	1	0	0	选择所链接的外围功能4的运行 <sup>注1</sup> 。
0	1	0	1	选择所链接的外围功能5的运行 <sup>注1</sup> 。
0	1	1	0	选择所链接的外围功能6的运行 <sup>注1</sup> 。
0	1	1	1	选择所链接的外围功能7的运行 <sup>注1</sup> 。
1	0	0	0	选择所链接的外围功能8的运行 <sup>注1</sup> 。
1	0	0	1	选择所链接的外围功能9的运行 <sup>注1</sup> 。
上述以外				禁止设置。

注1.请参照“表23-3ELSELRn寄存器（n=00~22）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23-2 ELSELRn寄存器 (n=00~22) 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	保留	-
ELSELR07	保留	-
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
ELSELR22	保留	-

表23-3 ELSELRn寄存器 (n=00~22) 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 注1	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的定时器输入注2	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注3	实时输出
1001B	9	DA1注3	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 $f_{CLK}$ , 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。

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 $f_{CLK}$ , 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。

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。

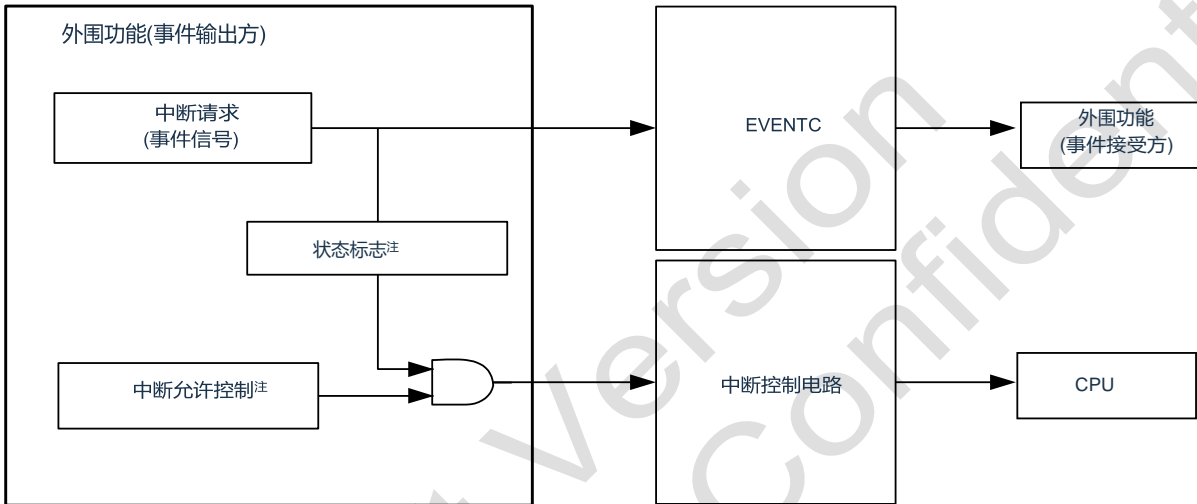
## 23.4 EVENTC的运行

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23-3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23-3 ELSELRn寄存器（n=00~22）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23-3 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23-4所示。

表23-4 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24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96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。

### 24.1 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。

### 24.2 中断源和结构

中断源列表参照表24-1。

表24-1 中断源一览表(1/3)

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		
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	UART2接收的传送结束		
	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	UART0接收的传送结束		
	12	INTSRE0	发生UART0接收的通信错误		
	13	INTLIN_t	LIN发送完成		
	14	INTLIN_r	LIN接收完成		
	15	INTLIN_s	LIN状态		
	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			

注 1.基本构成类型(A)~(D)分别对应图24-1的(A)~(D)。

2.这是将电压检测电平寄存器 (LVIS) 的bit7 (LVIMD) 置“0”的情况。

表24-1 中断源一览表(2/3)

中断处理	中断源 编号	中断源		内部/外部	基本结构 类型 <sup>注1</sup>		
		名称	触发				
可屏蔽	21	INTAD	A/D转换结束	内部	(A)		
	22	保留	-	-	-		
	23	保留	-	-	-		
	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				
	32	保留	-				
	33	INTP6	引脚输入边沿的检测			外部	(B)
	34	INTP7	引脚输入边沿的检测				
	35	INTP8	引脚输入边沿的检测				
	36	INTP9	引脚输入边沿的检测				
	37	INTP10	引脚输入边沿的检测				
	38	INTP11	引脚输入边沿的检测				
	39	保留	-	-	-		
	40	保留	-	-	-		
	41	INTSSPI0	高速SPIHS0传输结束中断	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	INTLIN_m	LIN综合中断	内部	(A)		
	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）				
	48	INTDIV	除法器计算结束				

注 1.基本构成类型(A)~(D)分别对应图24-1的(A)~(D)。

表24-1 中断源一览表(3/3)

中断处理	中断源 编号	中断源		内部/外部	基本结构 类型注
		名称	触发		
可屏蔽	49	保留	-	-	-
	50	保留	-	-	-
	51	保留	-	-	-
	52	保留	-	-	-
	53	保留	-	-	-
	54	保留	-	-	-
	55	保留	-	-	-
	56	保留	-	-	-
	57	保留	-	-	-
	58	保留	-	-	-
	59	保留	-	-	-
	60	保留	-	-	-
	61	保留	-	-	-
	62	保留	-	-	-
	63	保留	-	-	-
不可屏蔽	—	INTWDT	看门狗定时器间隔中断注2	内部	(D)

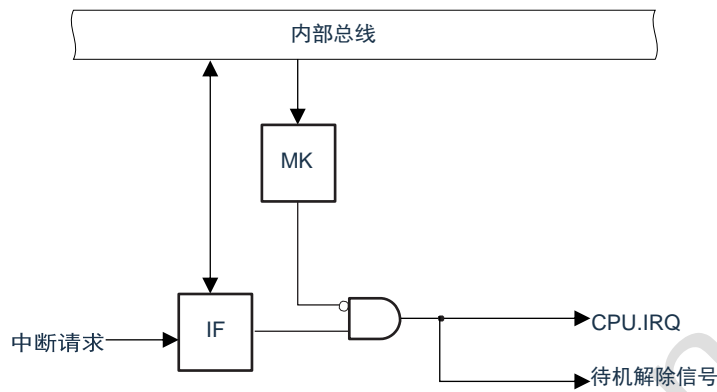
注 1.基本构成类型(A)~(D)分别对应图24-1的(A)~(D)。

2.这是将选项字节(000C0H)的bit7(WDTINT)置“1”的情况。

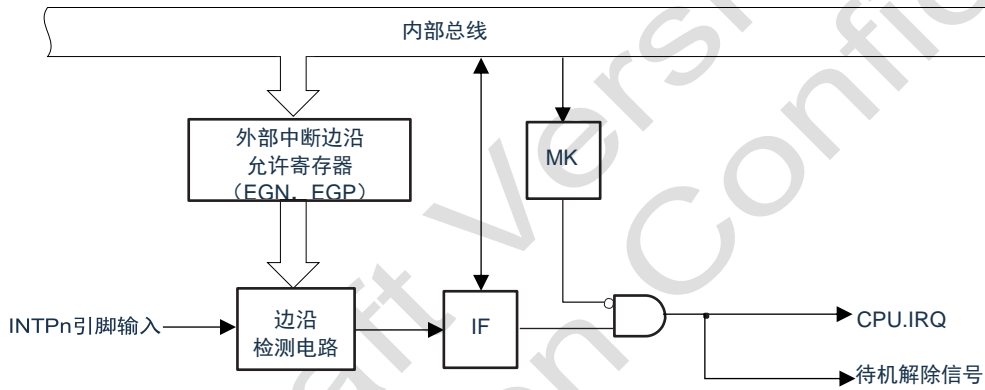


图24-1 中断功能的基本结构

(A)内部可屏蔽中断

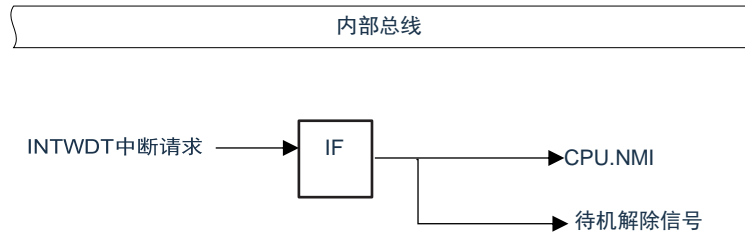


(B)外部可屏蔽中断 (INTPn)



注  $0 \leq n \leq 11$

(C)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。

Draft Version  
Cmsemicon Confidential

## 24.3 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)

### 24.3.1 中断请求标志寄存器 (IF00~IF31)

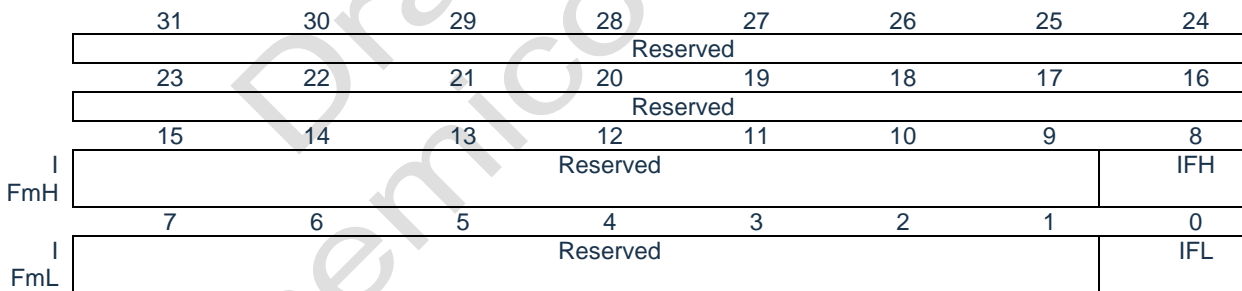
通过发生对应的中断请求或者执行指令，将中断请求标志置“1”。通过产生复位信号或者执行指令，将中断请求标志清“0”。

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24-2 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



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 24-2。  
 2.中断请求标志寄存器与 CPU.IRQ 的对应关系见图 24-4。  
 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。

### 24.3.2 中断屏蔽标志寄存器（MK00~MK31）

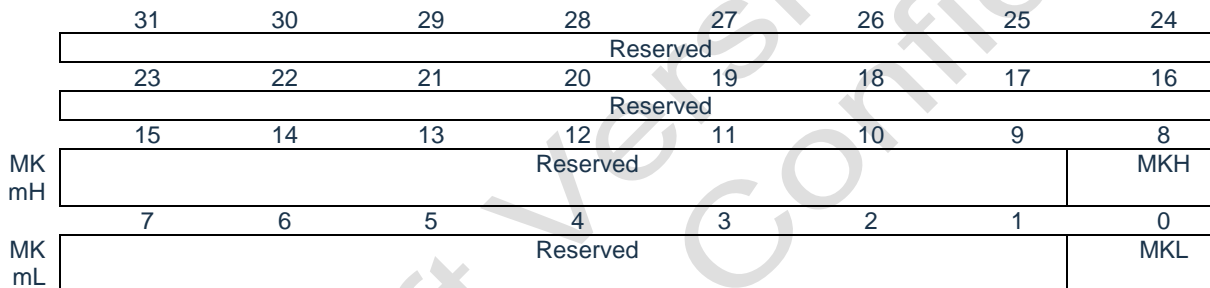
中断屏蔽标志设置允许或者禁止对应的可屏蔽中断处理。

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24-3 中断请求屏蔽寄存器（MKm）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



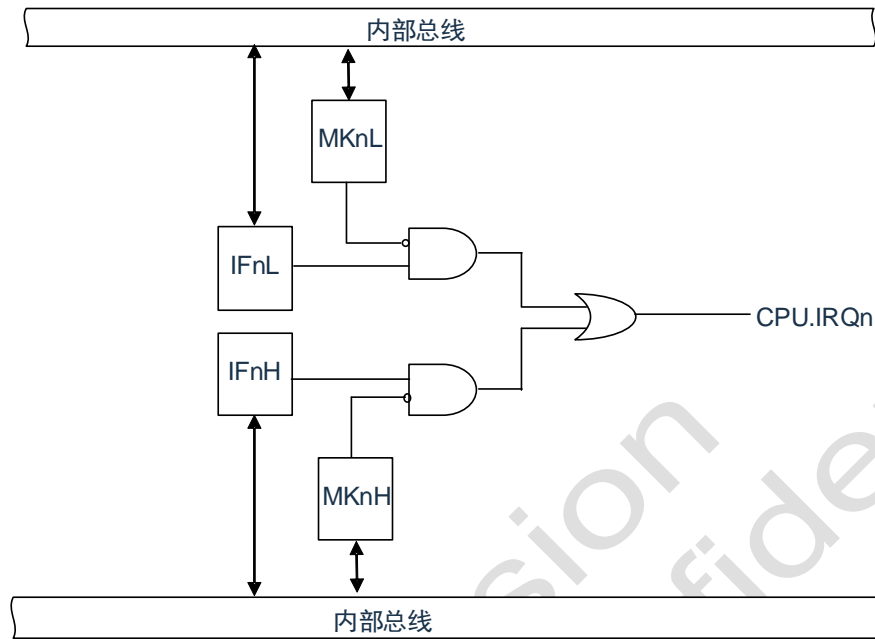
MK mL	编号0~31的中断源的中断处理控制注1
0	允许中断处理。
1	禁止中断处理。
MK mH	编号32~63的中断源的中断处理控制注2
0	允许中断处理。
1	禁止中断处理。

注：1.中断源与中断请求屏蔽寄存器的对应关系见表 24-2。  
 2.中断请求屏蔽寄存器与 CPU.IRQ 的对应关系见图 24-4.

表24-2 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/INTS SPI20/INTIIC2 0	IF07.IFL	MK07.MKL	39	保留	IF07.IFH	MK07.MKH
8	INTSR2	IF08.IFL	MK08.MKL	40	保留	IF08.IFH	MK08.MKH
9	INTSRE2	IF09.IFL	MK09.MKL	41	INTSSPI0	IF09.IFH	MK09.MKH
10	INTST0/INTS SPI00/INTIIC0 0	IF10.IFL	MK10.MKL	42	保留	IF10.IFH	MK10.MKH
11	INTSR0	IF11.IFL	MK11.MKL	43	保留	IF11.IFH	MK11.MKH
12	INTSRE0	IF12.IFL	MK12.MKL	44	INTTM01H	IF12.IFH	MK12.MKH
13	INTLIN_t	IF13.IFL	MK13.MKL	45	保留	IF13.IFH	MK13.MKH
14	INTLIN_r	IF14.IFL	MK14.MKL	46	INTLIN_m	IF14.IFH	MK14.MKH
15	INTLIN_s	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	保留	IF22.IFL	MK22.MKL	54	保留	IF22.IFH	MK22.MKH
23	保留	IF23.IFL	MK23.MKL	55	保留	IF23.IFH	MK23.MKH
24	INTCMP0	IF24.IFL	MK24.MKL	56	保留	IF24.IFH	MK24.MKH
25	INTCMP1	IF25.IFL	MK25.MKL	57	保留	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24-4 各标志寄存器与CPU.IRQ的关系



### 24.3.3 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24-5 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24-3所示。

表24-3 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	32引脚	24引脚	20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

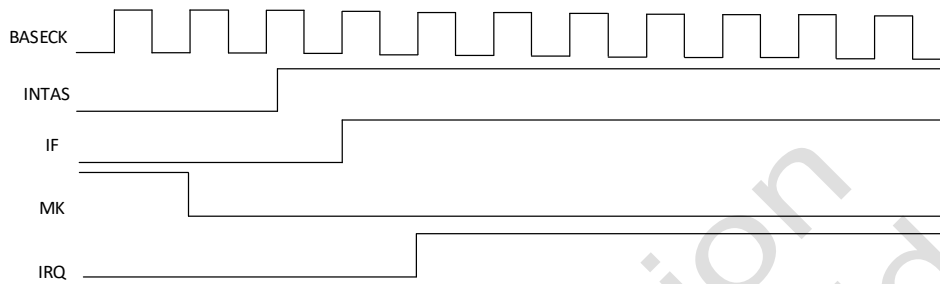


## 24.4 中断处理的操作

### 24.4.1 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。

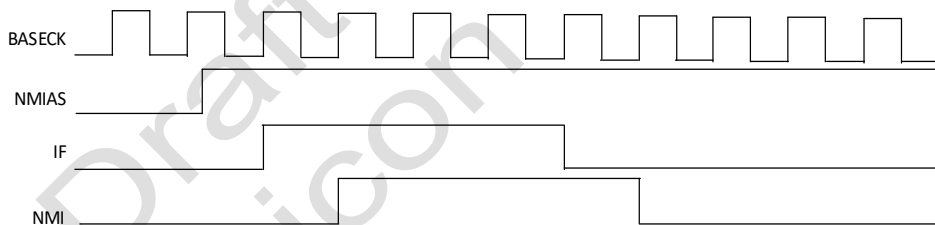
从中断请求标志被置1，到CPU的IRQ被置1，只需要1个时钟。



### 24.4.2 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25章 待机功能

### 25.1 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31章选项字节”。

## 25.2 睡眠模式

### 25.2.1 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25-1。

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指令，也不转移到睡眠模式。

Draft Version  
Cmsemicon Confidential

表25-1 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)。
低速内部振荡器时钟	$f_{IL}$	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟控制寄存器 (SUBCKSEL) 的SELLOSC位进行设置。 SELLOSC=1: 振荡 SELLOSC=0并且WDTON=0: 停止 SELLOSC=0、WDTON=1并且WDSTBYON=1: 振荡 SELLOSC=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存				
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				
看门狗定时器		参照“第12章看门狗定时器”。		
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)				
LIN				
SPI				
数据传送控制器 (DMA)				
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				
CRC运算功能	高速CRC			
	通用CRC	在RAM区的运算中执行DMA时, 能运行。		
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

表25-1 睡眠模式中的运行状态(2/2)

睡眠模式的设置		在CPU以低速内部振荡器时钟运行的过程中执行WFI指令的情况	
项目		CPU以低速内部振荡器时钟 ( $f_{IL}$ ) 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。	
	$f_X$		
	$f_{EX}$		
低速内部振荡器时钟	$f_{IL}$	继续运行	
CPU		停止运行。	
代码闪存			
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。	
Time4		能运行。	
看门狗定时器		参照“第12章看门狗定时器”。	
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		禁止运行。	
D/A转换器		能运行。	
比较器		能运行。	
通用串行通信单元（SCI）		能运行。	
串行接口（IICA）		禁止运行。	
LIN		禁止运行。	
SPI		能运行。	
数据传送控制器（DMA）		能运行。	
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			
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

$f_{IL}$ ：低速内部振荡器时钟

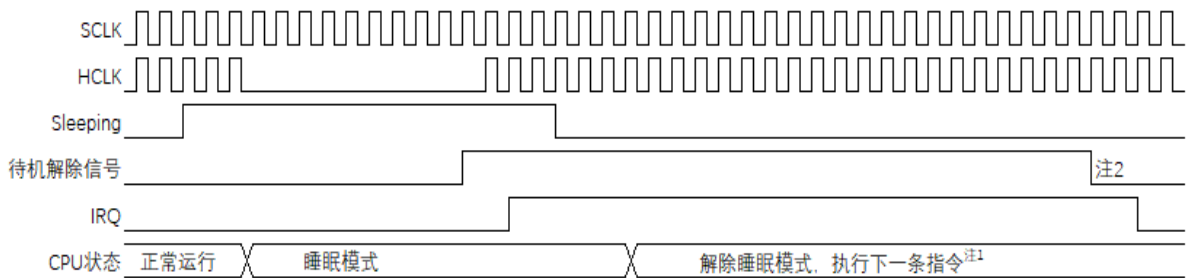
## 25.2.2 睡眠模式的解除

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25-1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，开始执行中断服务程序，需要16个时钟。

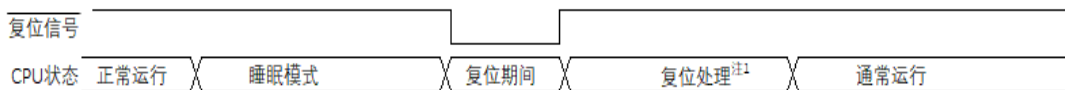
2. 待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

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25-2通过复位解除睡眠模式



注 1：有关复位处理时间，请参照“第 26 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 27 章上电复位电路”。

## 25.3 深度睡眠模式

### 25.3.1 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25-2。

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25-2 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以外系统时钟 ( $f_{EX}$ ) 运行
项目				
系统时钟		停止给CPU提供时钟。		
主系统时钟	$f_{IH}$	停止		
	$f_X$			
	$f_{EX}$			
	$f_{IL}$	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟控制寄存器 (SUBCKSEL) 的SELLOSC位进行设置。 SELLOSC=1: 振荡 SELLOSC=0并且WDTON=0: 停止 SELLOSC=0、WDTON=1并且WDSTBYON=1: 振荡 SELLOSC=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存				
RAM				
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。		
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"> <li>在选择没有TAIO输入滤波器时的事件计数模式中, 能运行。</li> <li>在选择低速内部振荡器作为计数源时, 能运行。</li> <li>上述以外: 禁止运行。</li> </ul>		
定时器M		禁止运行。		
定时器B				
定时器C				
时钟输出/蜂鸣器输出		在选择选择低速内部振荡器时钟作为计数时钟时, 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。		
LIN		禁止运行。		
SPI		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				
CRC运算功能	高速CRC	停止运行。		
	通用CRC			
RAM奇偶校验功能				
SFR保护功能				

备注1. 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

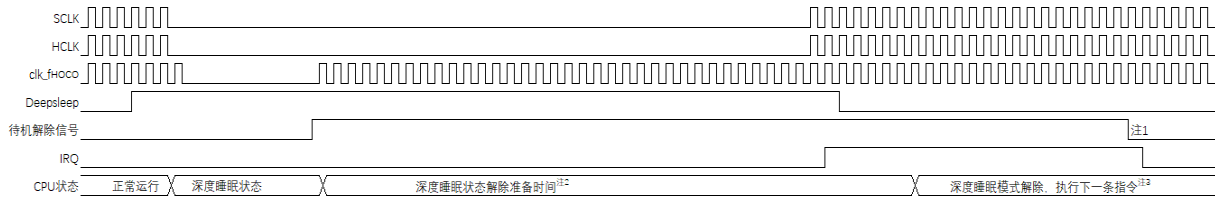
## 25.3.2 深度睡眠模式的解除

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25-3 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”。

#### 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和振荡稳定时间(通过OSTS进行设置)中较长的时间

进入深度睡眠模式前CPU时钟为PLL时钟时需要额外的LOCKUP时间。

#### 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。**

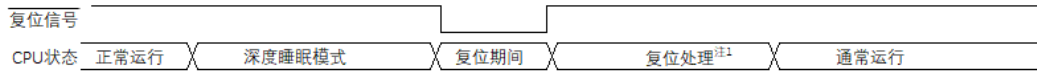
备注 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。



(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25-4 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26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27章 上电复位电路”。

Draft Version  
Cmsemicon Confidential

## 第26章 复位功能

以下7种方法产生复位信号。

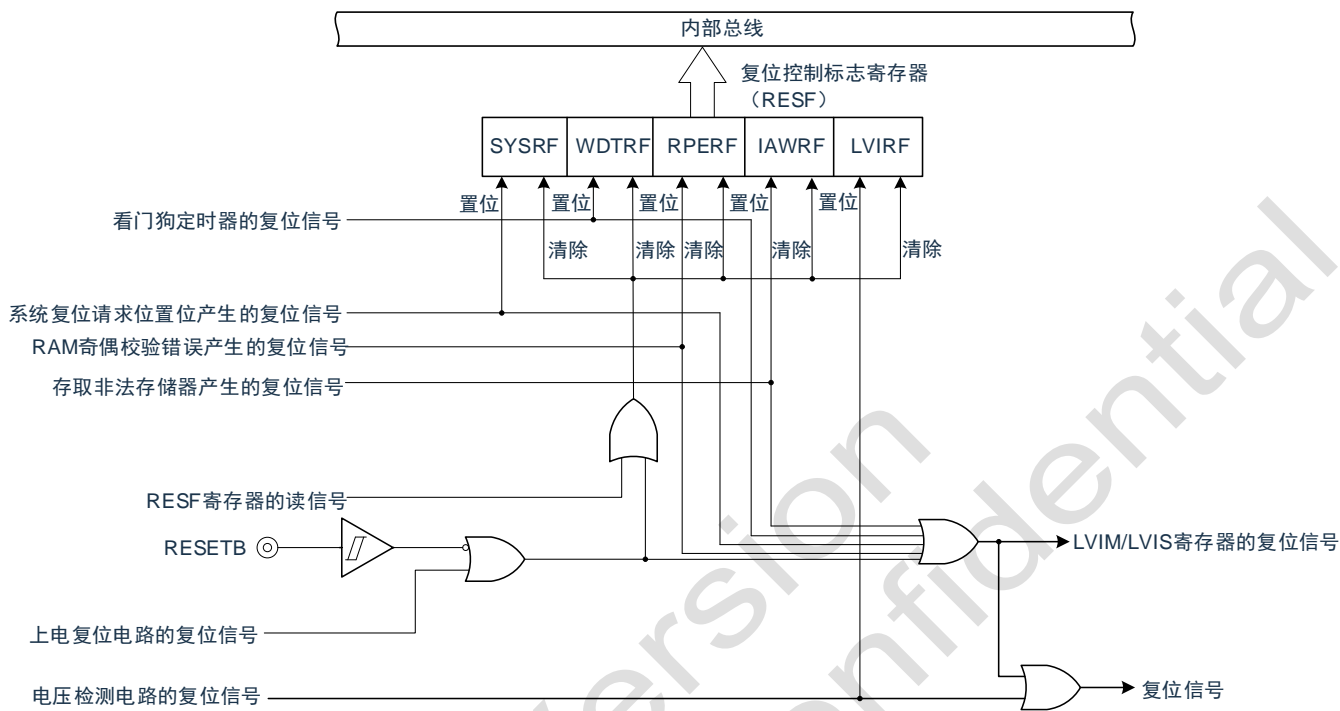
- (1)通过RESETB引脚输入外部复位。
- (2)通过看门狗定时器的程序失控检测产生内部复位。
- (3)通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4)通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5)因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为1而产生内部复位。
- (6)因RAM奇偶校验错误而产生内部复位。
- (7)因存取非法存储器而产生内部复位。

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，或者检测到停振时，产生复位并且各硬件变为如表26-1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、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：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解除复位后为高电平(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以外的端口：在复位期间以及解除复位后为高阻抗。

图26-1 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。

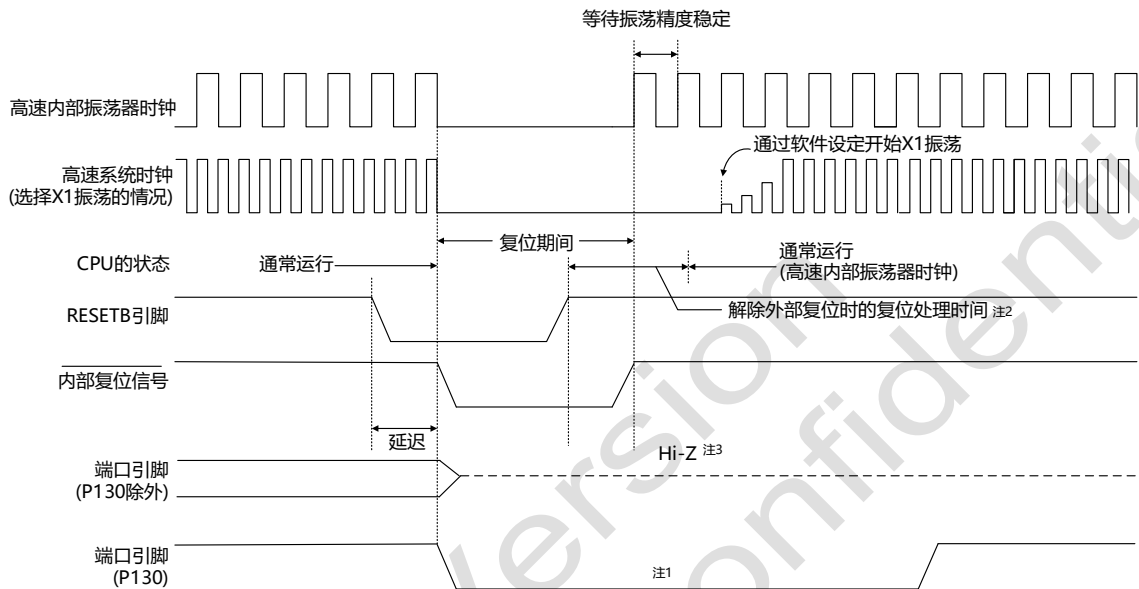
备注1.LVIM: 电压检测寄存器

2.LVIS: 电压检测电平寄存器

## 26.1 复位时序

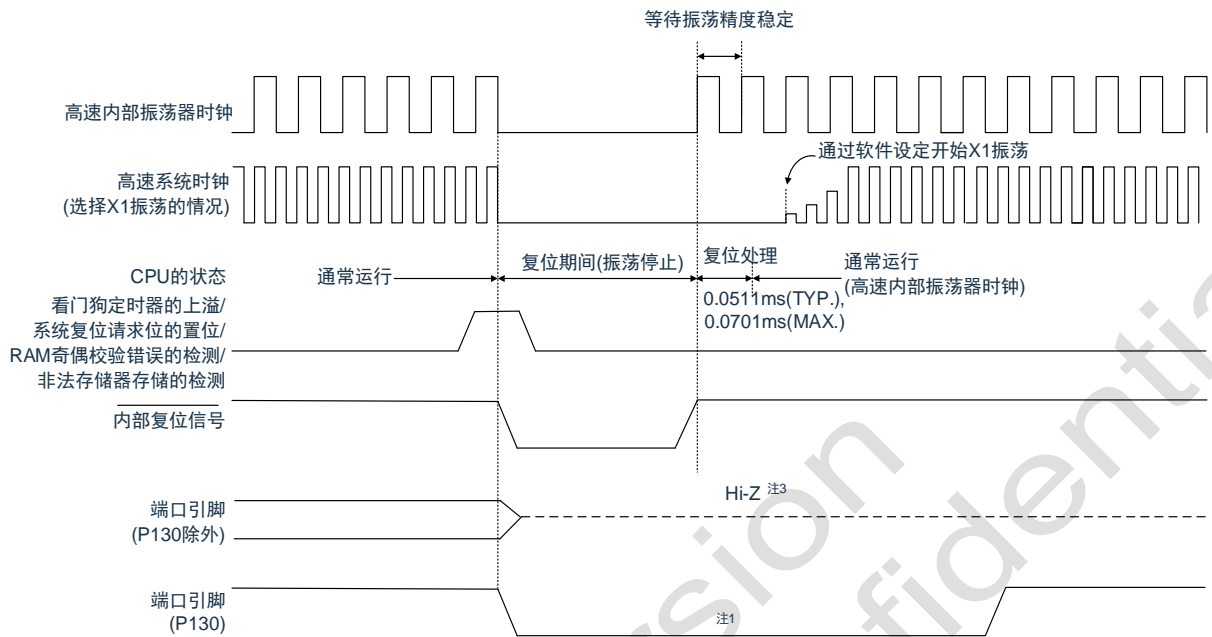
当给RESETB引脚输入低电平时，产生复位。然后，如果给RESETB引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图26-2 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或者停振检测而产生的复位，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26-3 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。

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.)。

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30章上电复位电路”和“第31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26-1 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）。	
	$f_{IL}$	停止运行。	
CPU		停止运行。	
代码闪存		停止运行。	
RAM		停止运行。	
端口（锁存器）		高阻抗 <sup>注1</sup>	
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		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 <sup>注1</sup>			
比较器 <sup>注1</sup>			
通用串行通信单元（SCI）			
串行接口（IICA）			
SPI			
LIN			
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。	
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  
 $f_{EX}$  : 外部主系统时钟

$f_X$  : X1振荡时钟  
 $f_{IL}$  : 低速内部振荡器时钟

## 26.2 确认复位源的寄存器

### 26.2.1 复位控制标志寄存器（RESF）

BAT32A233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26-4 复位控制标志寄存器（RESF）的格式

地址：40020440H 复位后：不定值<sup>注1</sup> 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26-2。

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26-2所示。

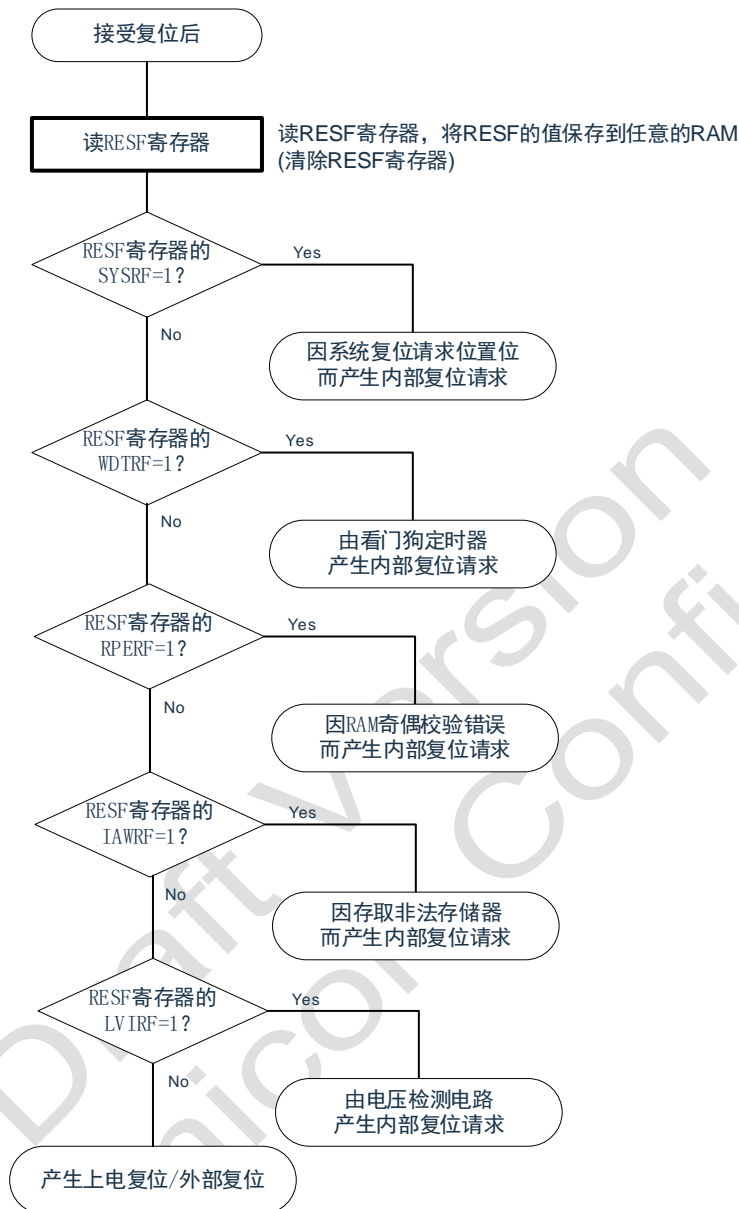
表26-2 发生复位请求时的RESF寄存器状态

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”			
RPERF			保持	置“1”			
IAWRF			保持	置“1”			
LVIRF			保持	置“1”			

复位源的确认步骤如图26-5所示。



图26-5 复位源的确认步骤



注意 上述流程是确认步骤的例子。

## 第27章 上电复位电路

### 27.1 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. BAT32G113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

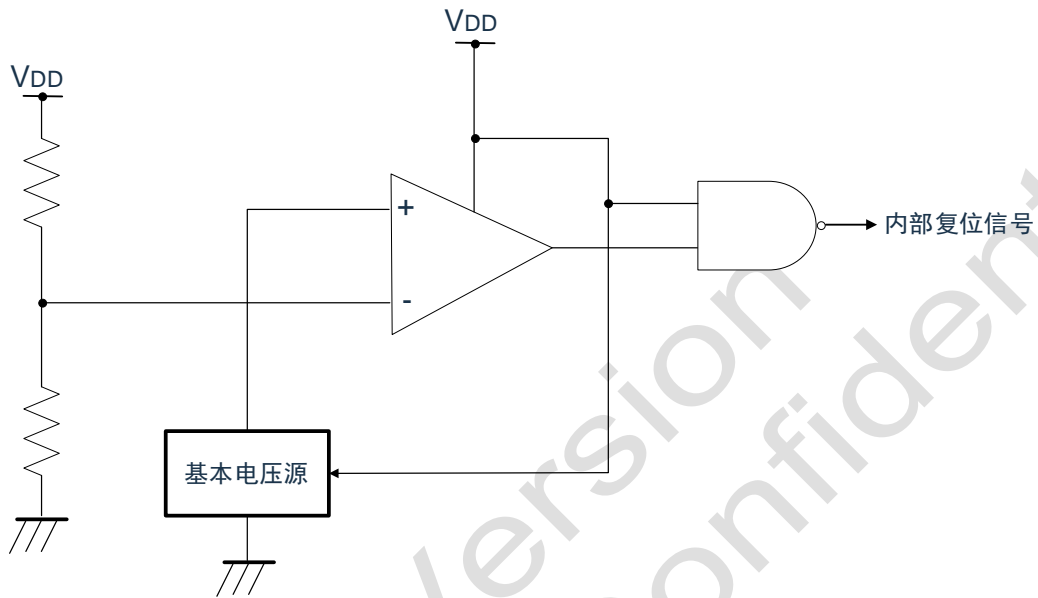
$V_{PDR}$ : POR电源电压下降检测电压

详细内容请参照数据手册的POR电路特性。

## 27.2 上电复位电路的结构

上电复位电路的框图如图27-1所示。

图27-1 上电复位电路的框图

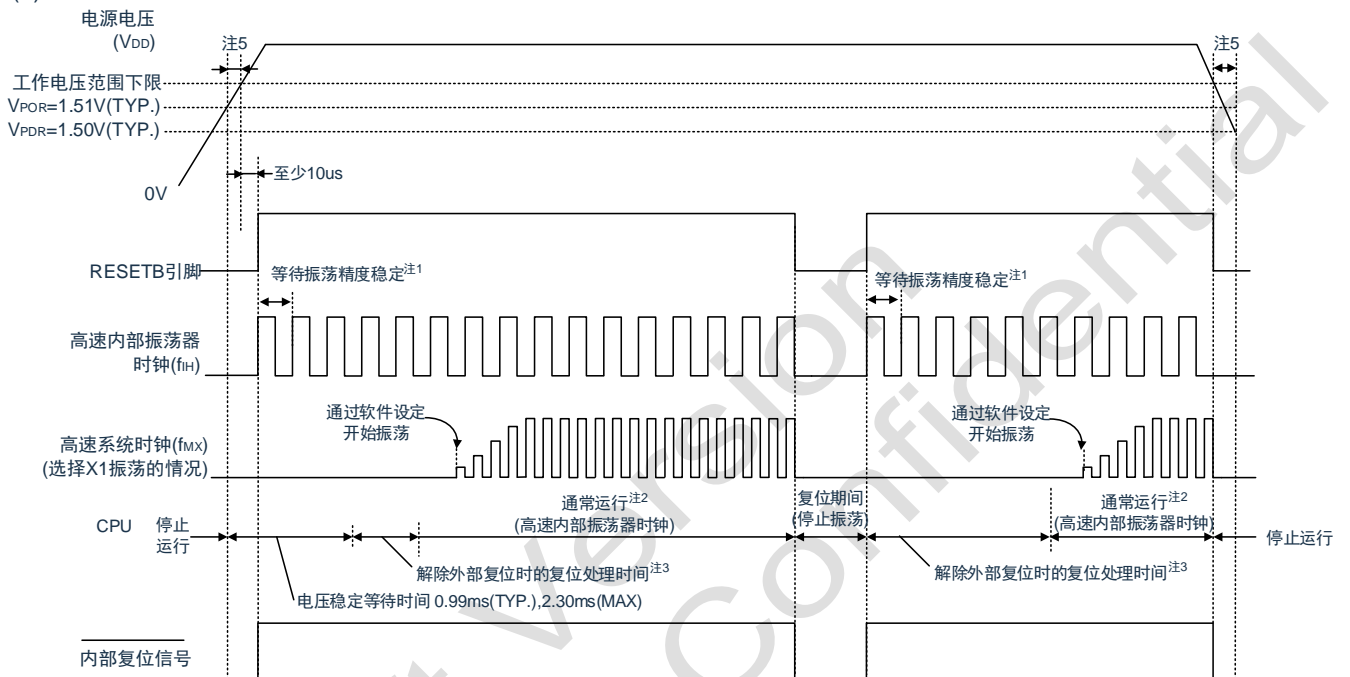


## 27.3 上电复位电路的运行

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27-2 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。

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。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。

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）

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）

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。

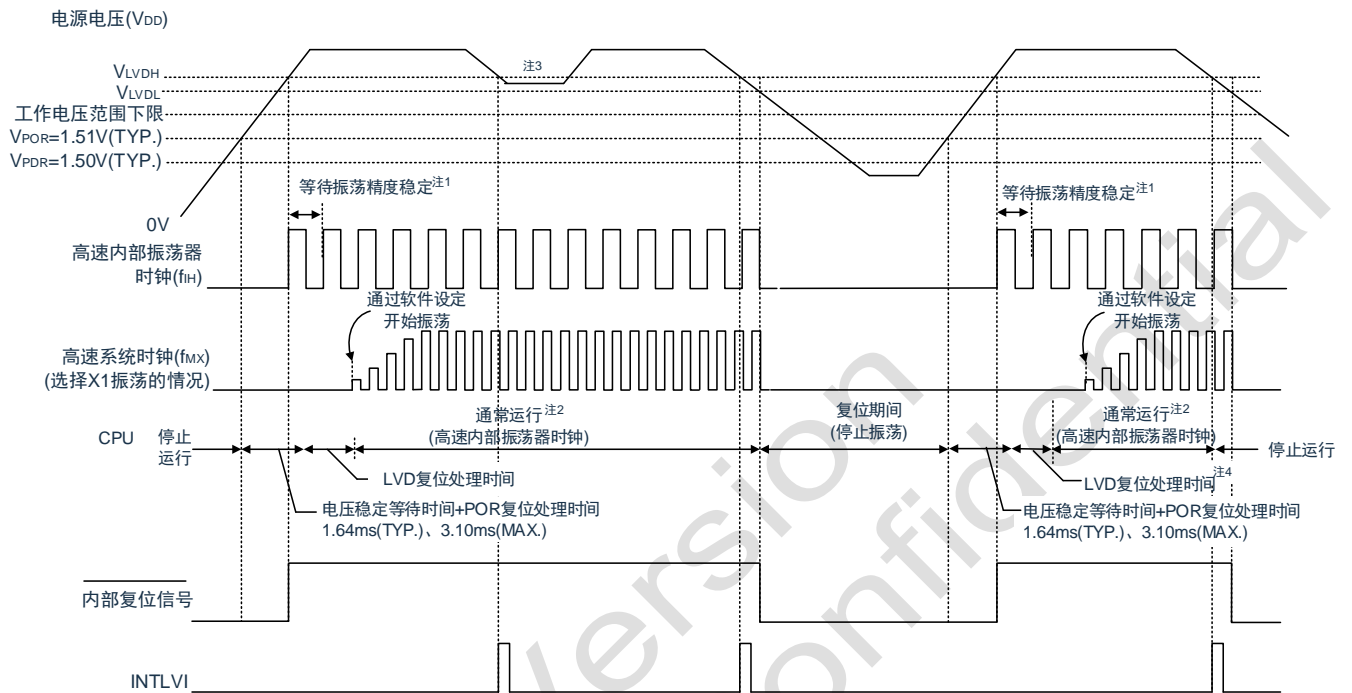
备注  $V_{POR}$ ：POR电源电压上升检测电压

$V_{PDR}$ ：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27-2 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。

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。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。

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<sub>LVDL</sub>）的状态下恢复到高电压检测电压（V<sub>LVDH</sub>）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。

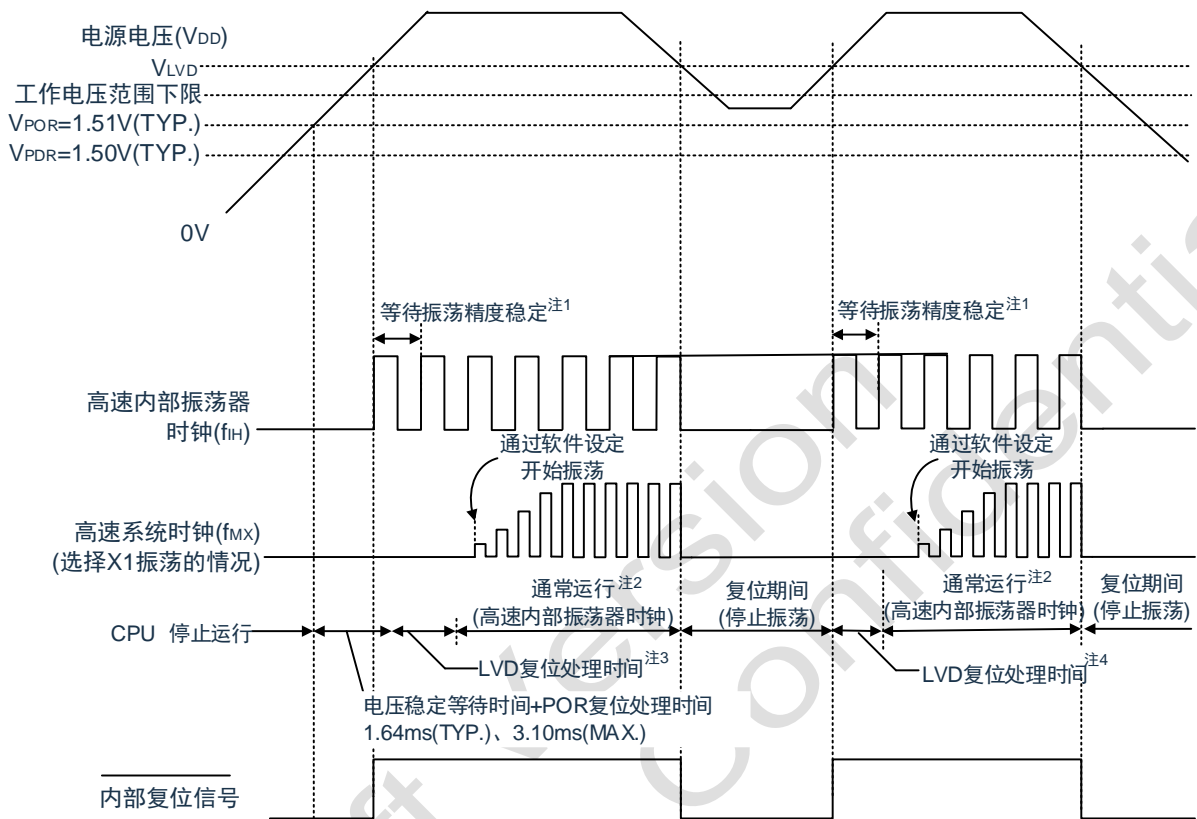
4.到开始通常运行为止的时间除了达到V<sub>POR</sub>（1.51V(TYP.))后的“电压稳定等待时间+POR复位处理时间”以外，在达到LVD检测电平（V<sub>LVDH</sub>）后还需要以下的“LVD复位处理时间”。

LVD复位处理时间：0ms~0.0701ms(MAX.)

备注 V<sub>LVDH</sub>、V<sub>LVDL</sub> : LVD检测电压  
 V<sub>POR</sub> : POR电源电压上升检测电压  
 V<sub>PDR</sub> : POR电源电压下降检测电压

图27-2 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。

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。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。

3.到开始通常运行为止的时间除了达到V<sub>POR</sub> (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<sub>LVD</sub>) 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.)

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<sub>LVD</sub>) 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<sub>LVDH</sub>、V<sub>LVDL</sub> : LVD检测电压

V<sub>POR</sub> : POR电源上升检测电压

V<sub>PDR</sub> : 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28章 电压检测电路

### 28.1 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34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）。

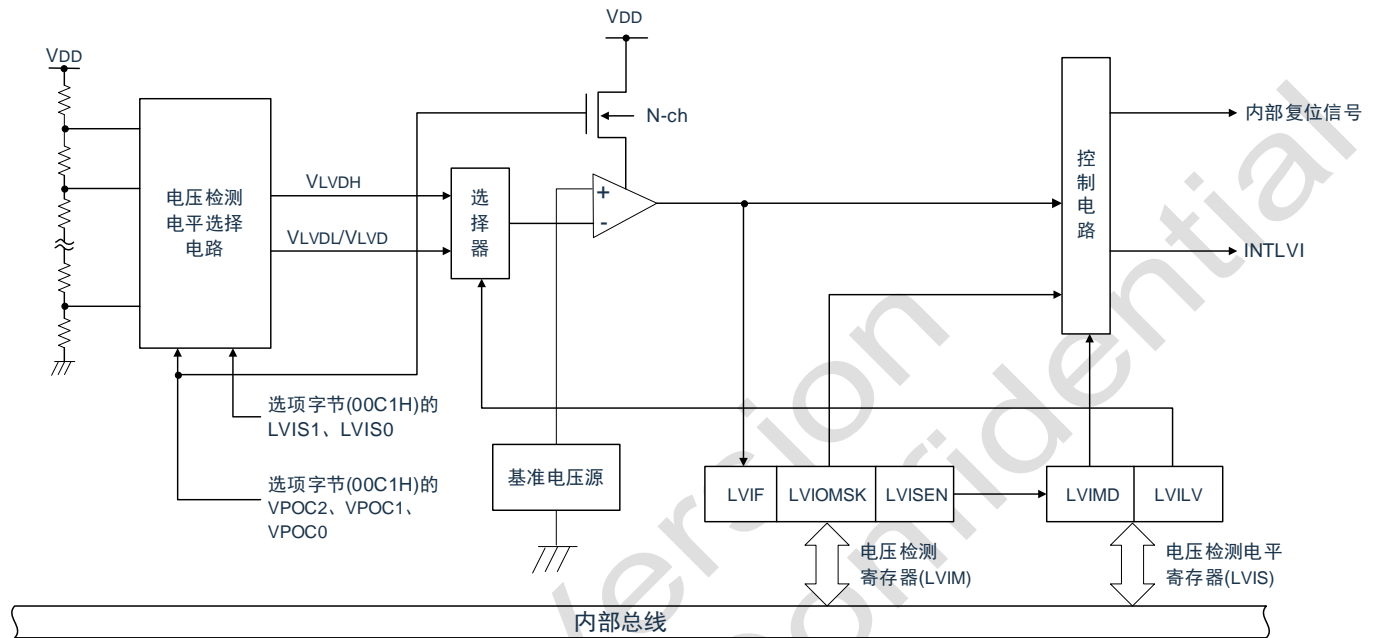
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

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29章 复位功能”。

## 28.2 电压检测电路的结构

电压检测电路的框图如图28-1所示。

图28-1 电压检测电路的框图





## 28.3 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)

### 28.3.1 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28-2电压检测寄存器 (LVIM) 的格式

地址：40020441H 复位后：00H<sup>注1</sup> R/W<sup>注2</sup>

符号	7	6	5	4	3	2	1	0
LVIM	LVISEN <sup>注3</sup>	0	0	0	0	0	LVIOMSK	LVIF

LVISEN <sup>注3</sup>	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 <sup>注4</sup> 。

LVIF	电压检测标志
0	电源电压 ( $V_{DD}$ ) $\geq$ 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”。

2. bit0和bit1是只读位。

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。

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

## 28.3.2 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”<sup>注1</sup>。

图28-3 电压检测电平寄存器 (LVIS) 的格式

地址: 40020442H 复位后: 00H/01H/81H<sup>注1</sup>

R/W

符号	7	6	5	4	3	2	1	0
LVIS	LVIMD <sup>注2</sup>	0	0	0	0	0	0	LVILV <sup>注2</sup>

LVIMD <sup>注2</sup>	电压检测的运行模式
0	中断模式
1	复位模式

LVILV <sup>注2</sup>	LVD检测电平
0	高电压检测电平 ( $V_{LVDH}$ )
1	低电压检测电平 ( $V_{LVDL}$ 或者 $V_{LVD}$ )

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

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 ( $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ )。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28-1 用户选项字节（000C1H/010C1H）的格式(1/2)

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&amp;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28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28-1 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。

## 28.4 电压检测电路的运行

### 28.4.1 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： $V_{LVD}$ ）。

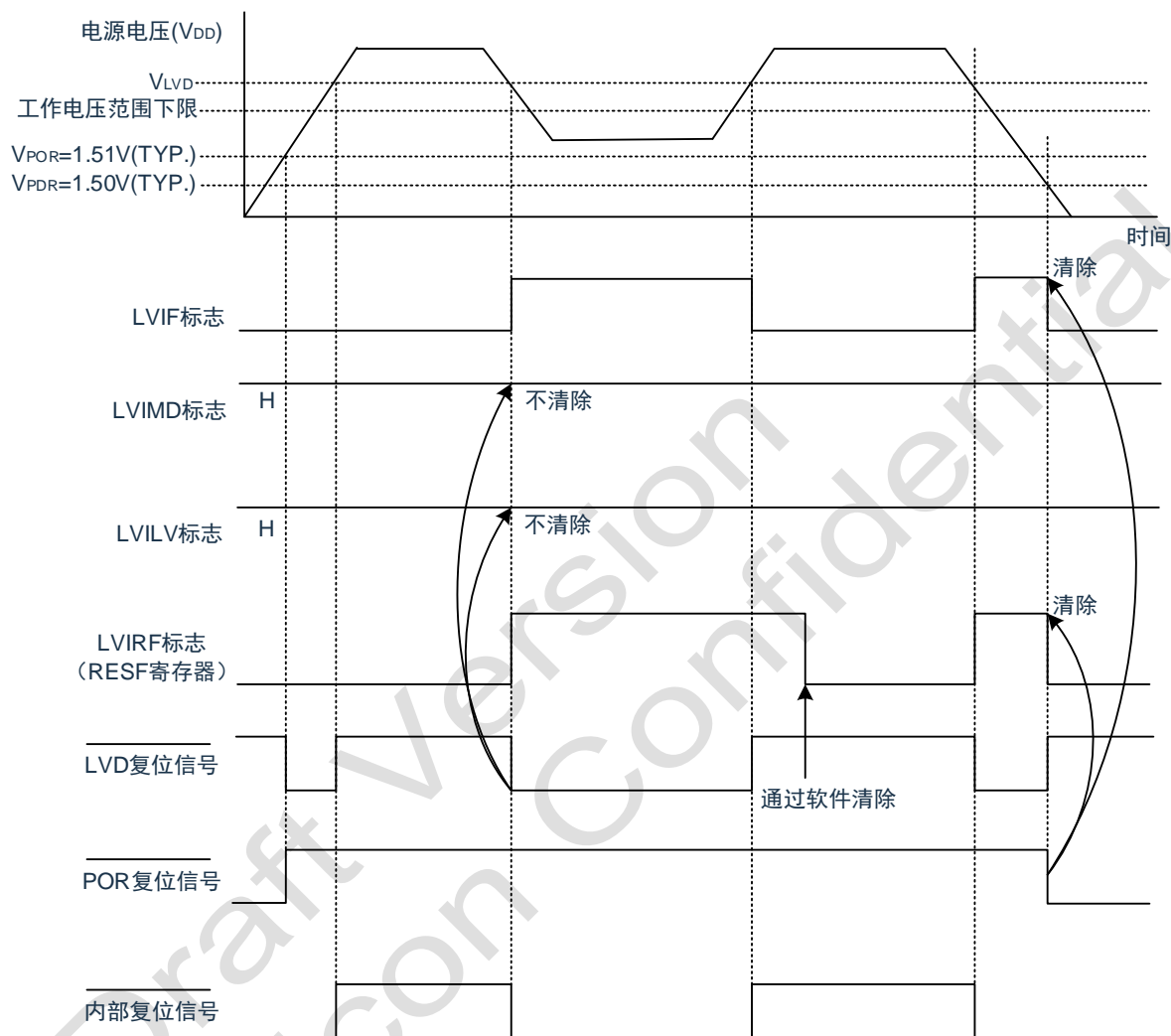
#### ●LVD复位模式的运行

当接通电源时，复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）在电源电压（VDD）超过电压检测电平（VLVD）前保持LVD的内部复位状态。如果电源电压（VDD）超过电压检测电平（VLVD），就解除内部复位。

当工作电压下降时，如果电源电压（VDD）低于电压检测电平（VLVD），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28-4所示。

图28-4 内部复位信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、1）



备注 V<sub>POR</sub>: POR电源电压上升检测电压  
 V<sub>PDR</sub>: POR电源电压下降检测电压

## 28.4.2 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： $V_{LVD}$ ）。

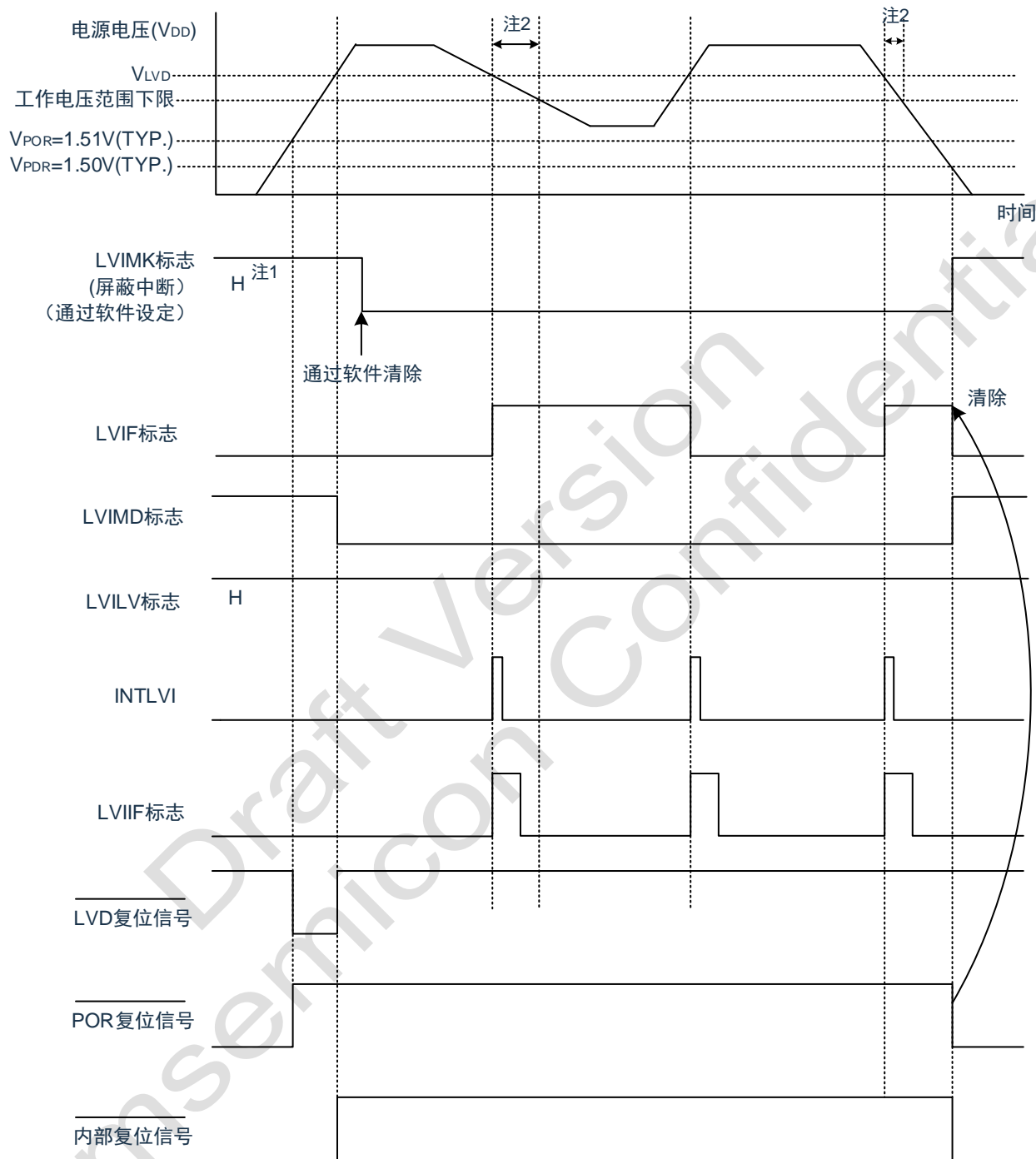
### ●LVD中断模式的运行

在产生复位后，中断模式（选项字节的LVIMDS1、LVIMDS0=0、1）在电源电压（VDD）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（VDD）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（VDD）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28-5所示。

图28-5 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”。

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<sub>POR</sub>: POR电源电压上升检测电压

V<sub>PDR</sub>: POR电源电压下降检测电压



### 28.4.3 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（VDD）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（VDD）超过高电压检测电平（ $V_{LVDH}$ ），就解除内部复位。

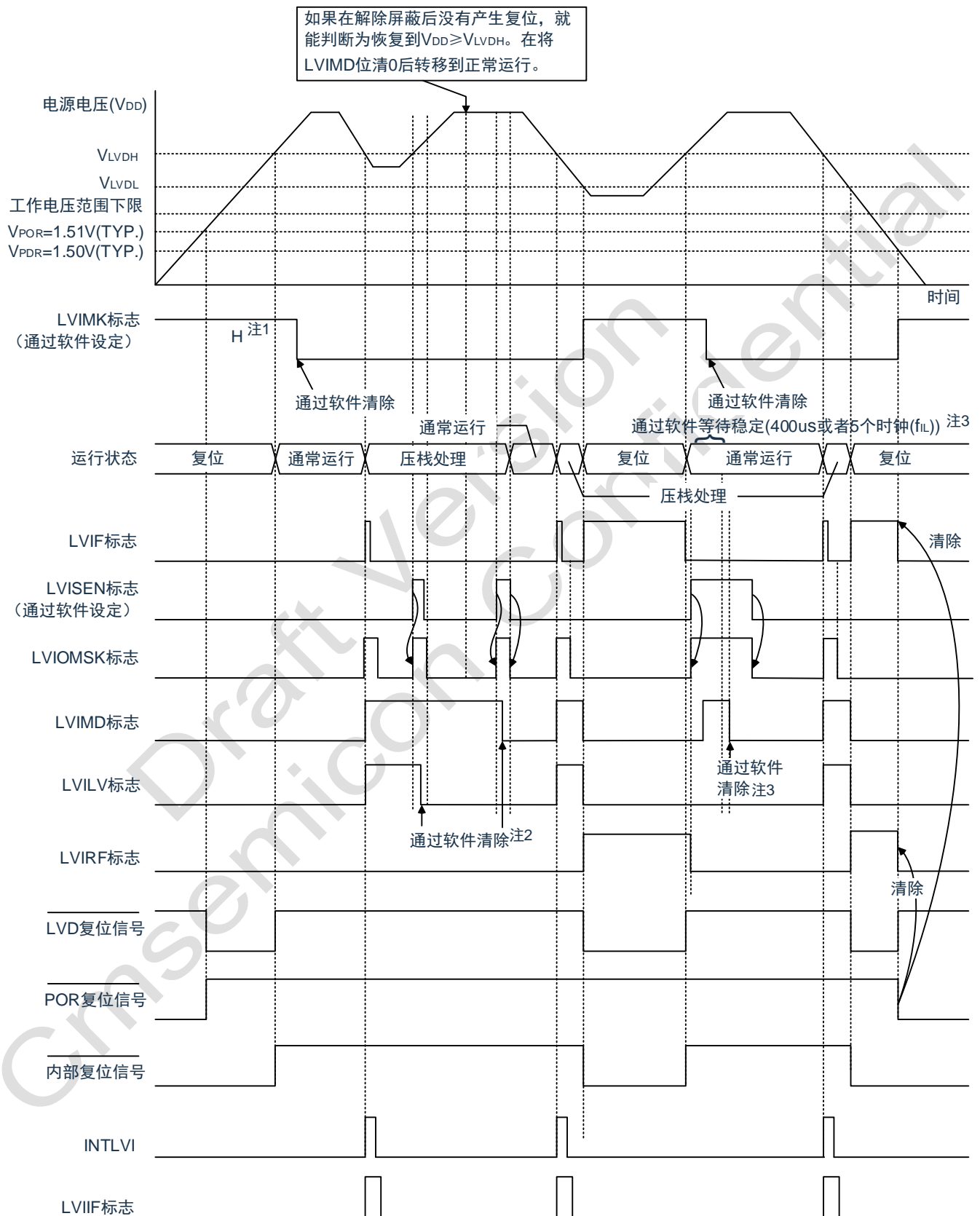
当工作电压下降时，如果电源电压（VDD）低于高电压检测电平（ $V_{LVDH}$ ），就产生LVD的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（VDD）低于低电压检测电平

（ $V_{LVDL}$ ），就产生LVD的内部复位。但是，在发生INTLVI后，即使电源电压（VDD）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28-7 工作电压的确认/复位的设置步骤”和“图28-8 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28-6所示。

图28-6 复位&中断信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、0）(1/2)

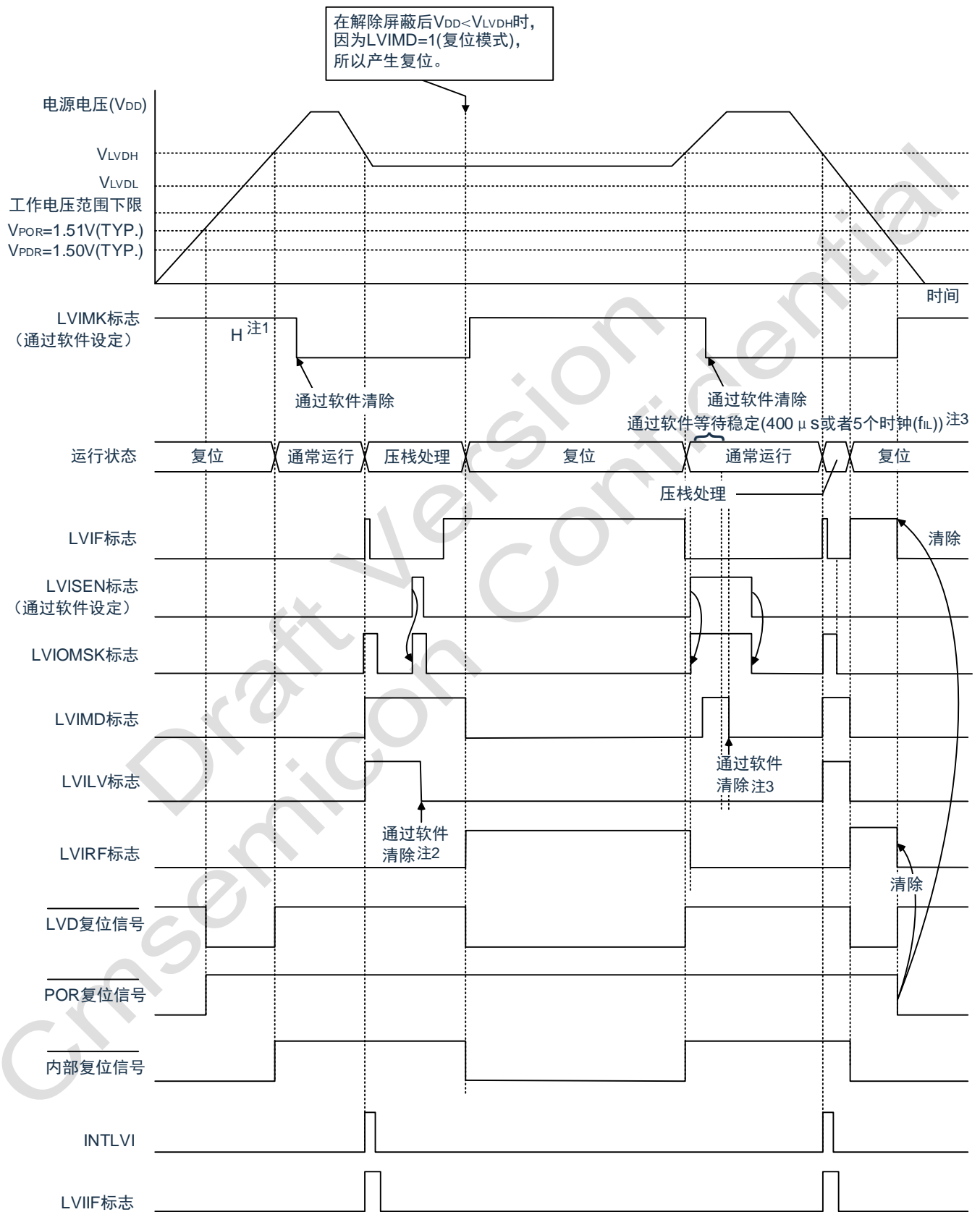


- 注 1.在产生复位信号后，LVIMK标志变为“1”。
- 2.当使用中断&复位模式时，必须在发生中断后按照“图27-7 工作电压的确认/复位的设置步骤”进行设置。
- 3.当使用中&复位模式时，必须在解除复位后按照“图27-8 中断&复位模式的初始设置步骤”进行设置。

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

Draft Version  
Cmsemicon Confidential

图28-7 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”。

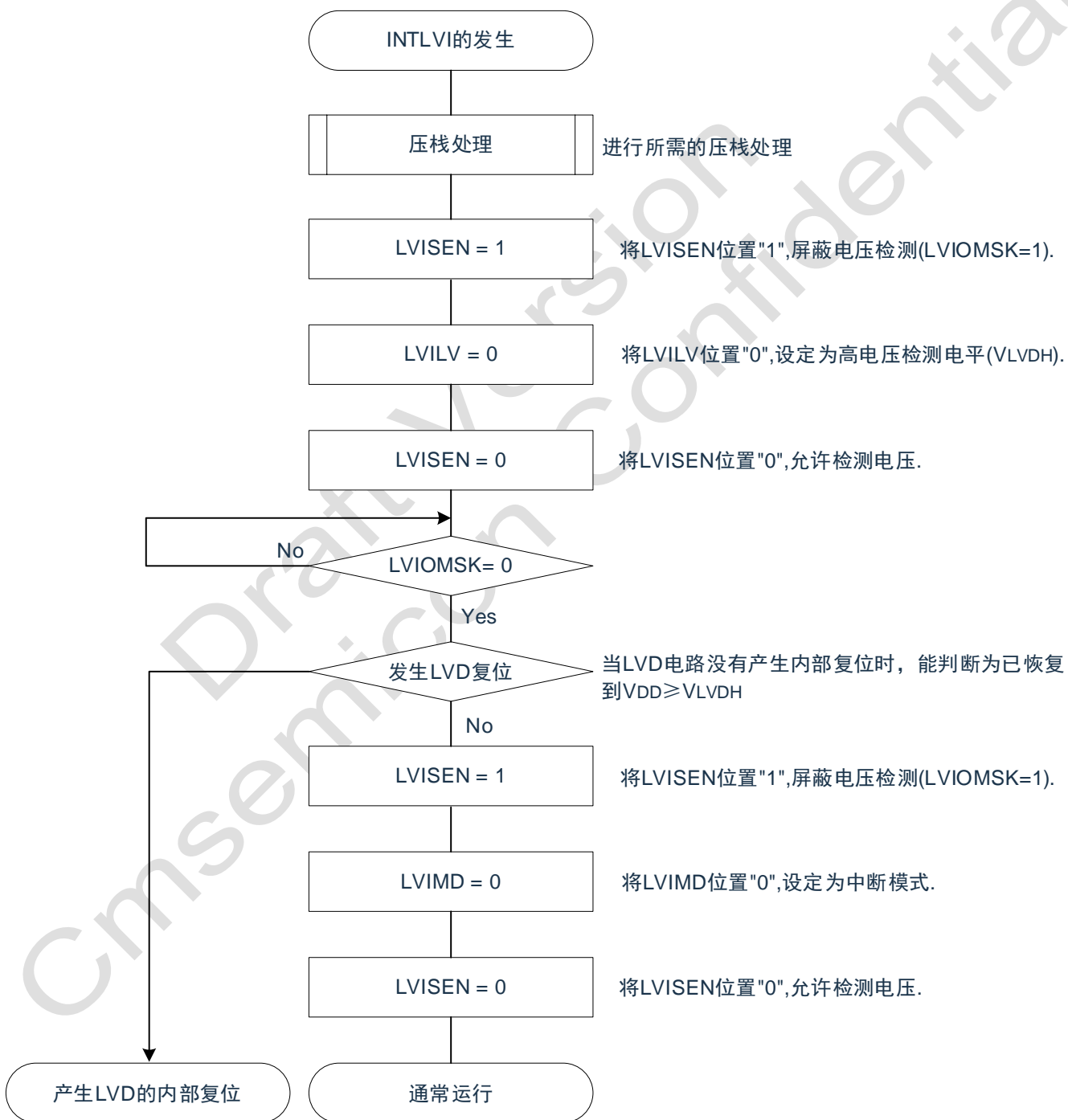
2.当使用中断&复位模式时, 必须在发生中断后按照“图28-8 工作电压的确认/复位的设置步骤”进行设置。

3.当使用中断&复位模式时, 必须在解除复位后按照“图28-9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

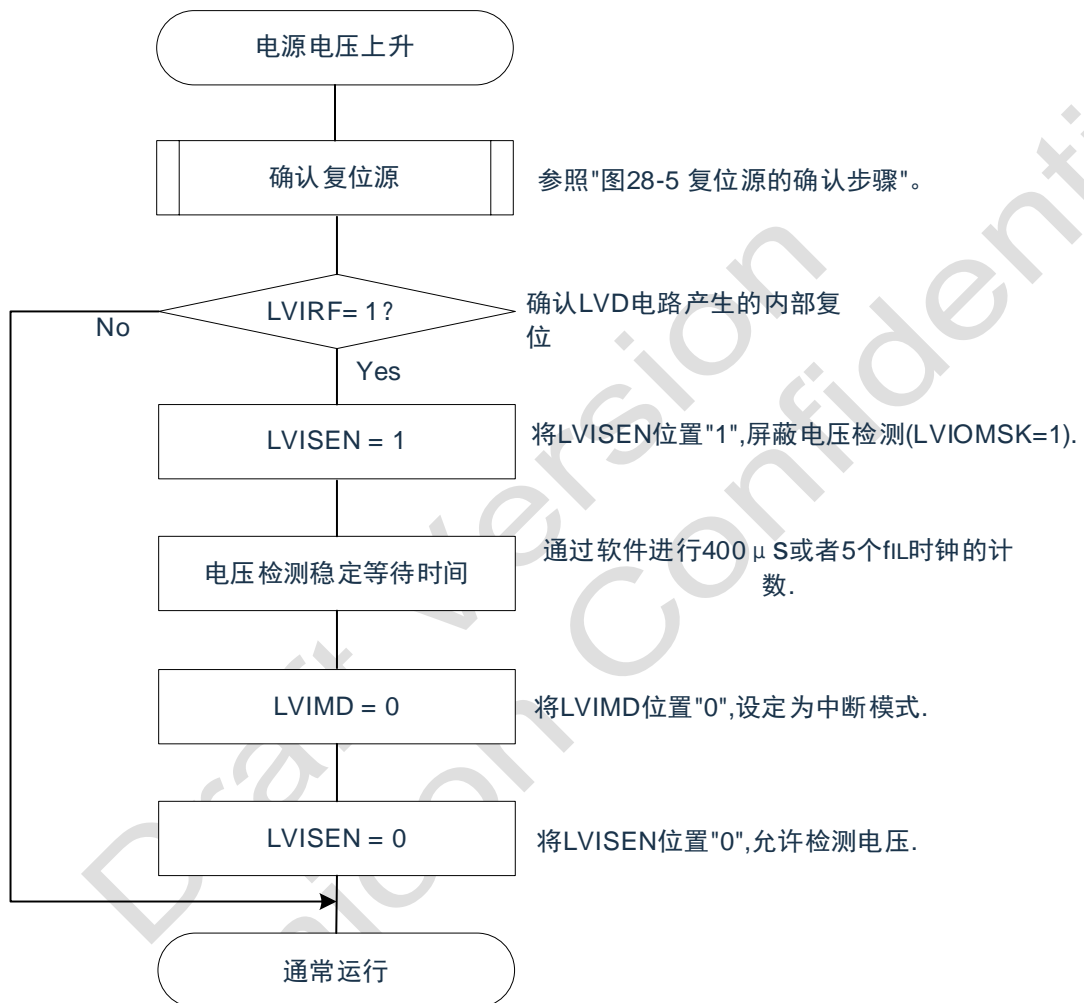
图28-8 工作电压的确认/复位的设置步骤



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 $f_{IL}$ 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28-9所示。

图28-9 中断&复位模式的初始设置步骤



备注  $f_{IL}$ : 低速内部振荡器时钟频率

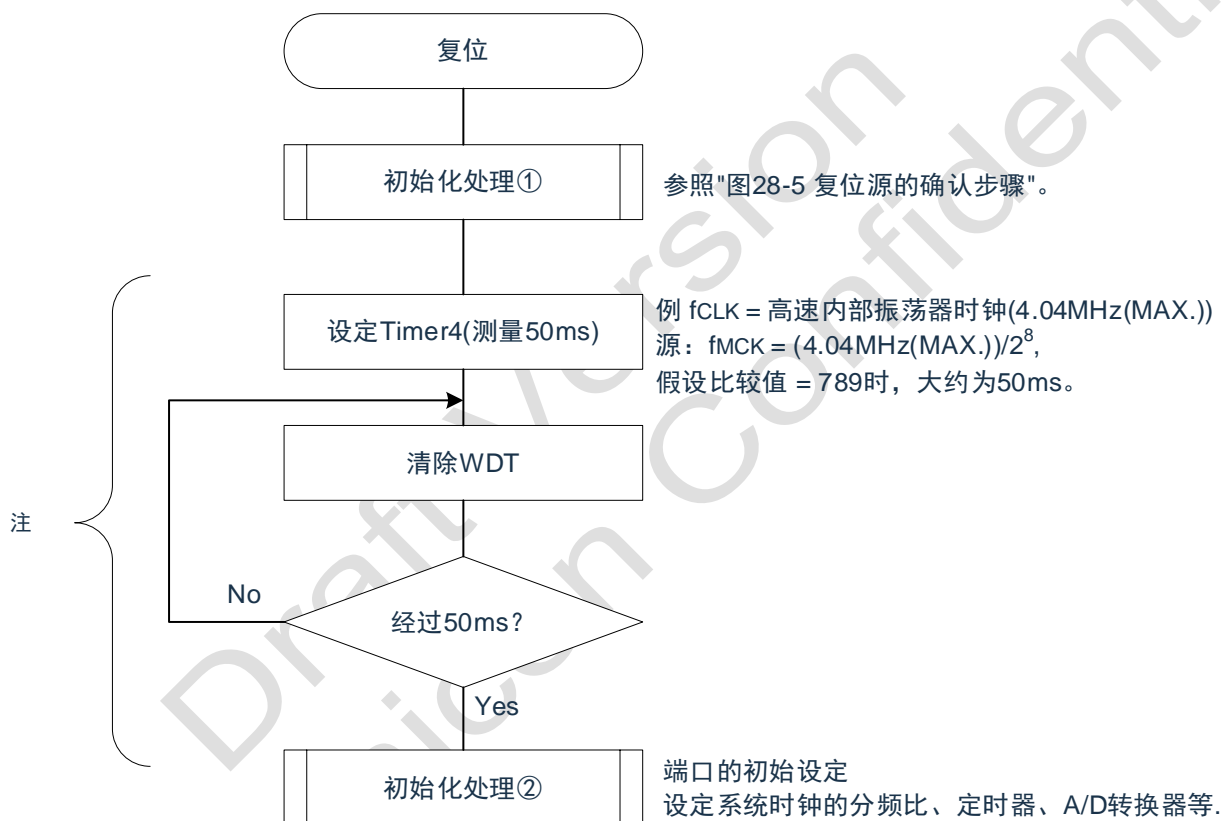
## 28.5 电压检测电路的注意事项

### (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28-10 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

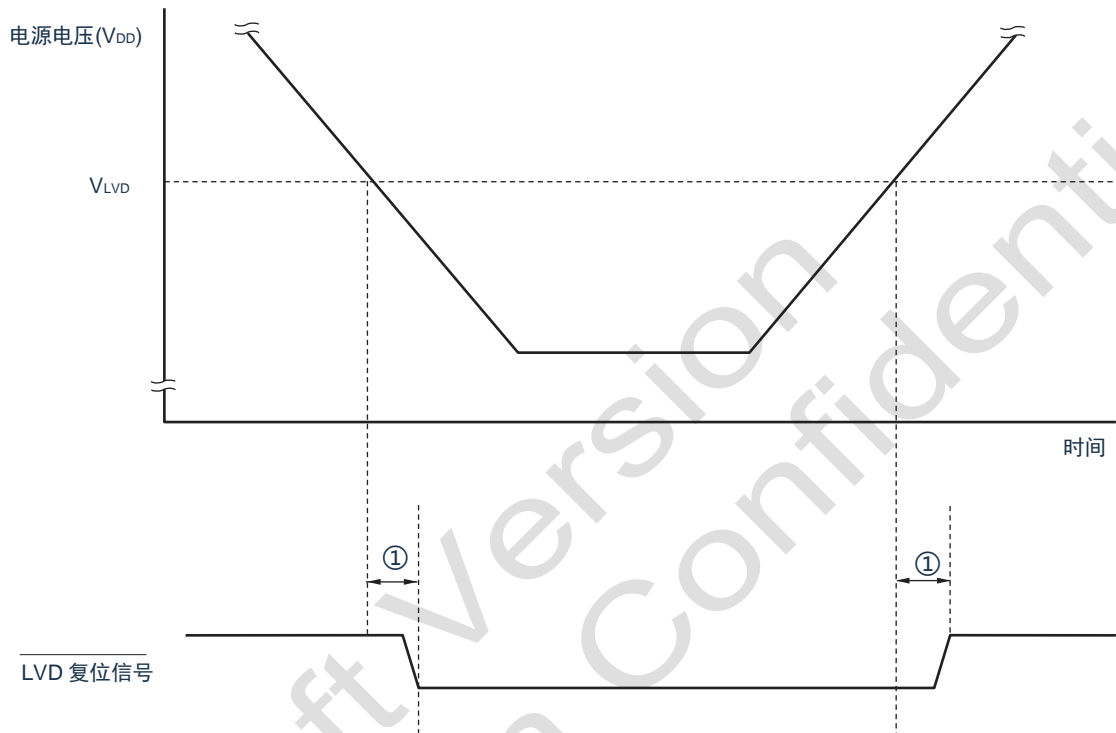


注 如果在此期间再次发生复位, 就不转移到初始化处理②。

(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 (图28-11)。

图28-11 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29章 安全功能

### 29.1 安全功能的概要

为了对应IEC60730和EC61508安全标准，BAT32A233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）

## 29.2 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"> <li>•闪存CRC控制寄存器 (CRC0CTL)</li> <li>•闪存CRC运算结果寄存器 (PGCRCL)</li> </ul>	闪存CRC运算功能 (高速CRC)
<ul style="list-style-type: none"> <li>•CRC输入寄存器 (CRCIN)</li> <li>•CRC数据寄存器 (CRCD)</li> </ul>	CRC运算功能 (通用CRC)
<ul style="list-style-type: none"> <li>•RAM奇偶校验错误控制寄存器 (RPECTL)</li> </ul>	RAM奇偶校验错误检测功能
<ul style="list-style-type: none"> <li>•特殊SFR保护控制寄存器 (SFRGD)</li> </ul>	SFR保护功能
<ul style="list-style-type: none"> <li>•定时器输入选择寄存器0 (TIS0)</li> </ul>	频率检测功能
<ul style="list-style-type: none"> <li>•A/D测试寄存器 (ADTES)</li> </ul>	A/D测试功能
<ul style="list-style-type: none"> <li>•端口模式选择寄存器 (PMS)</li> </ul>	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“29.3安全功能的运行”中进行说明。

## 29.3 安全功能的运行

### 29.3.1 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。

### 29.3.1.1 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29-1 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	0	0	0	0	0	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。

FEA1	FEA0	高速CRC的演算范围
0	0	0000H~1FFBH(8K-4byte)
0	1	0000H~3FFBH(16K-4byte)
1	0	0000H~5FFBH(24K-4byte)
1	1	0000H~7FFBH(32K-4byte)

注：bit6~2必须写0。

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。

### 29.3.1.2 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29-2 闪存CRC运算结果寄存器 (PGCRCL) 的格式

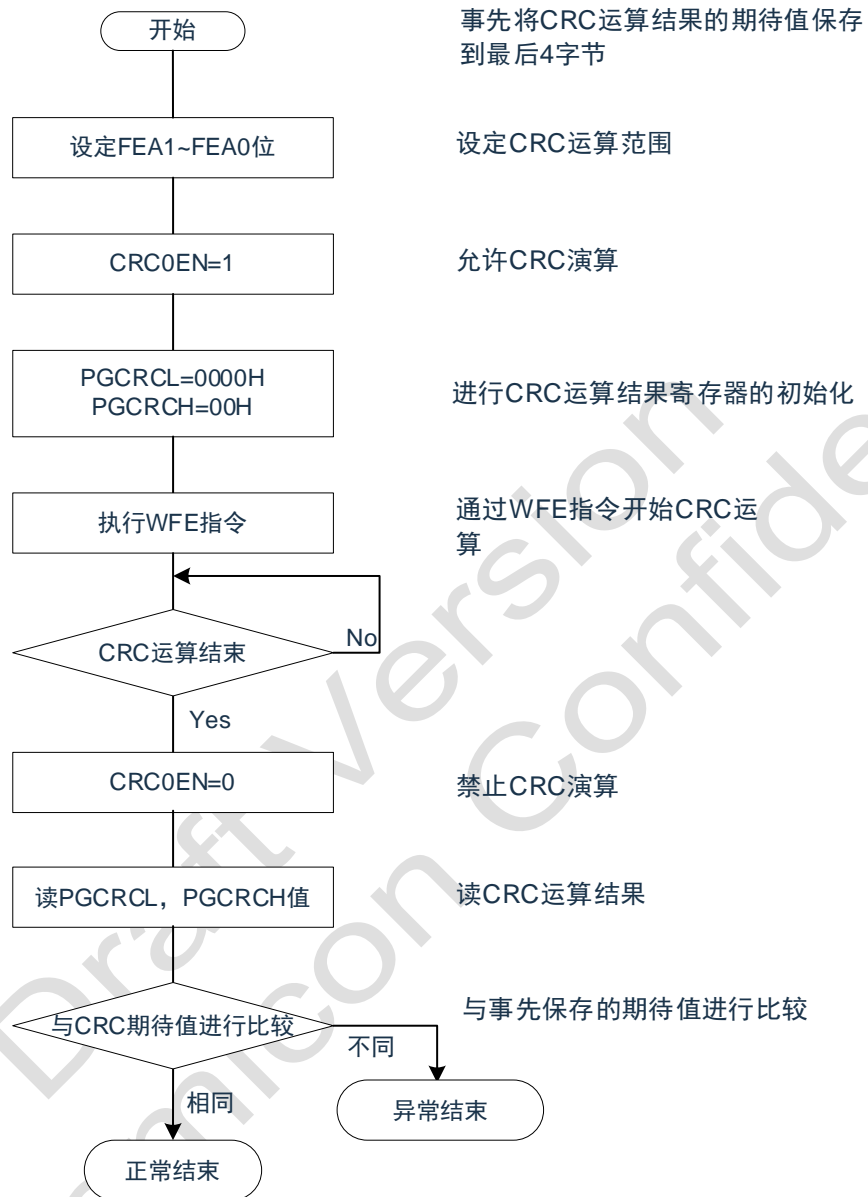
地址:	0x40021812	复位后: 0000H	R/W					
符号	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0
	PGCRC15~0		高速CRC的运算结果					
	0000H~FFFFH		保存高速CRC的运算结果。					

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29-3所示。

<操作流程>

图29-3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。

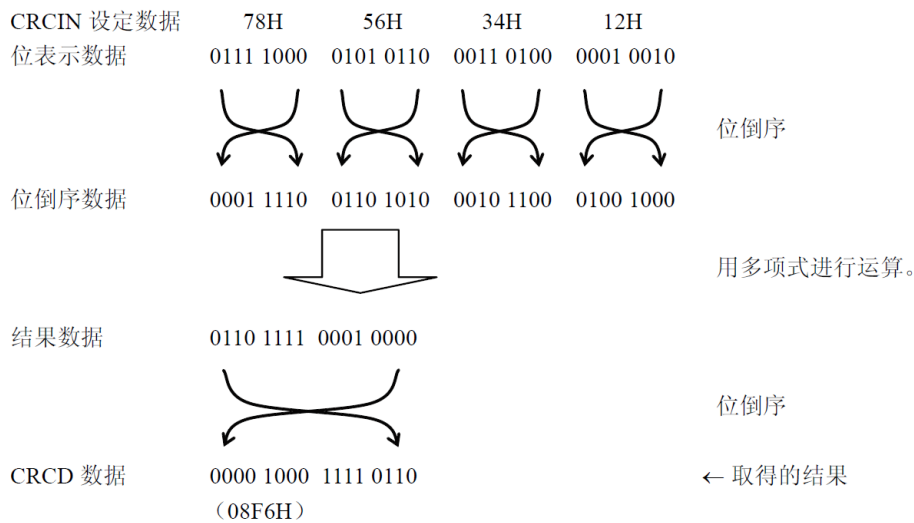
### 29.3.2 CRC运算功能（通用CRC）

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。

#### 29.3.2.1 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29-4 CRC输入寄存器（CRCIN）的格式

地址：400433ACH 复位后：00H

R/W



bit7~0	功能
00H~FFH	数据输入

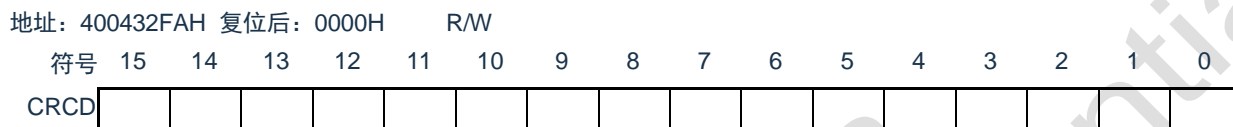
### 29.3.2.2 CRC数据寄存器 (CRCD)

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 ( $f_{CLK}$ )，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29-5 CRC数据寄存器 (CRCD) 的格式

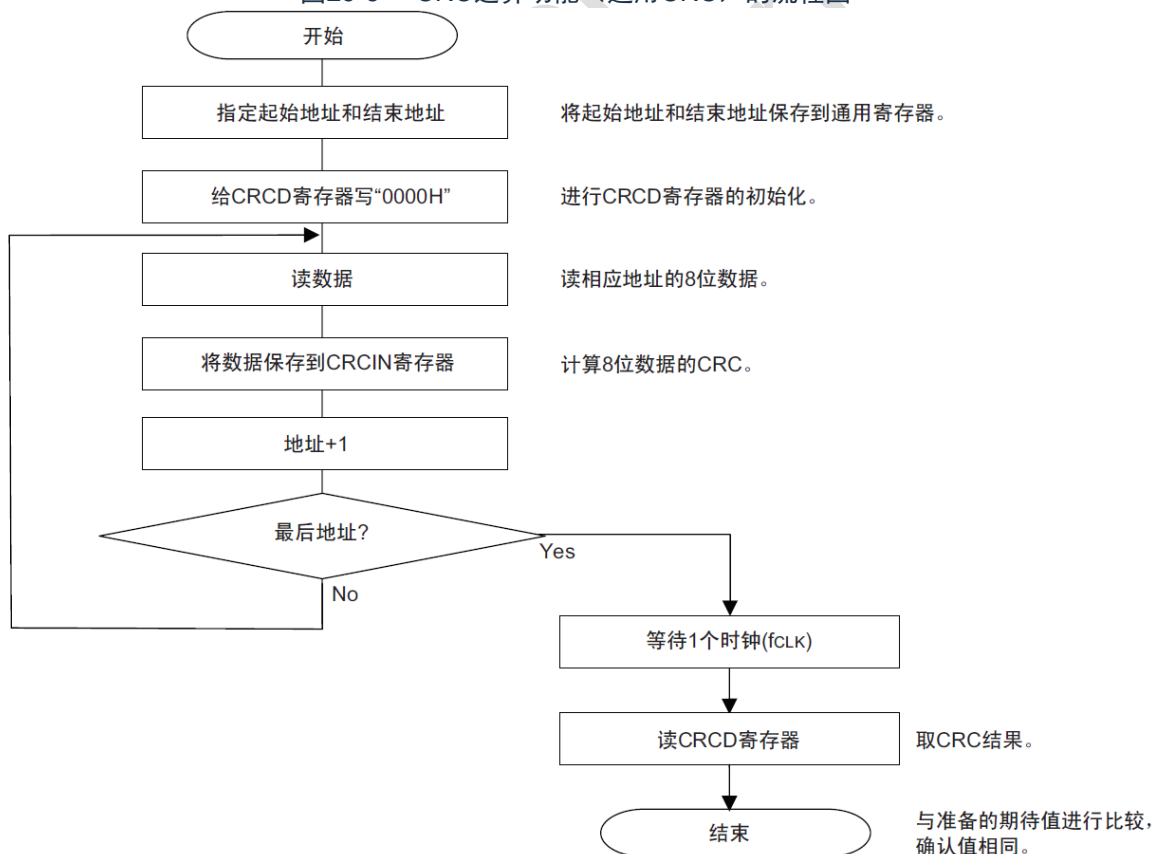


注意1.要读CRCD寄存器的写入值时，必须在写CRCIN寄存器前读CRCD寄存器。

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29-6 CRC运算功能 (通用CRC) 的流程图



### 29.3.3 RAM奇偶校验错误检测功能

IEC60730标准要求确认RAM数据。因此，BAT32A233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。

#### 29.3.3.1 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29-7 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

符号	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）。

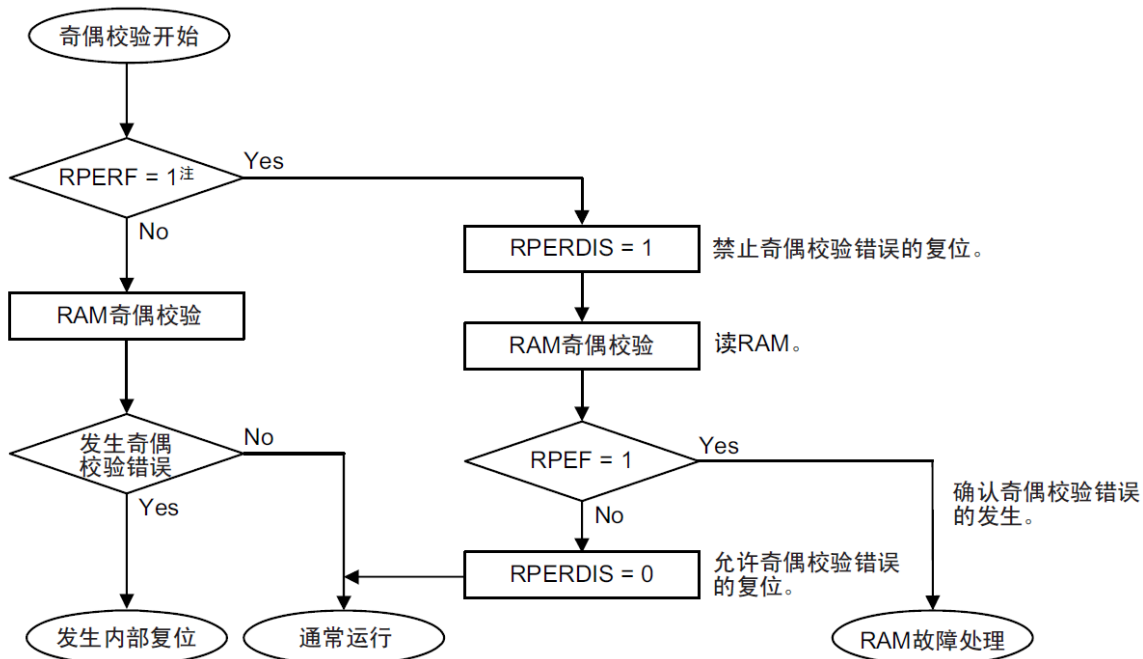
2.即使设置为禁止产生奇偶校验错误复位（RPERDIS=1），也在发生奇偶校验错误时将RPEF标志置“1”。如果在RPEF位为“1”的状态下设置为允许产生奇偶校验错误复位（RPERDIS=0），就在将RPERDIS清“0”时产生奇偶校验错误复位。

3.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。

4.RAM奇偶校验检测的范围不包括通用寄存器。



图29-8 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”。

## 29.3.4 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。

### 29.3.4.1 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29-9 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、CMPxHY、PGAxCTL、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、PSETxx、PCLRxx、PREADxx、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、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、OSTC、OSMC、SUBCKSEL、HOCODIV、HIOTRM、LVIM、LVIS							

注 不保护Pxx（端口寄存器）。

### 29.3.5 频率检测功能

IEC60730标准要求确认振荡频率是否正常。

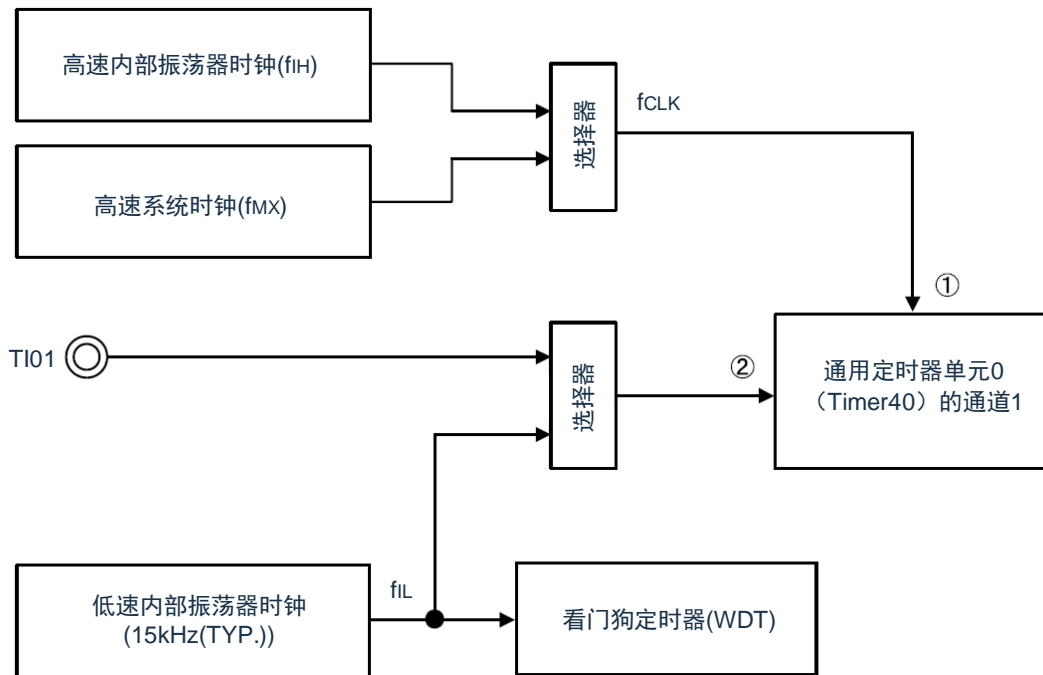
频率检测功能可使用CPU/外围硬件的时钟频率 ( $f_{CLK}$ )，并且能通过测量Timer4的通道1输入脉冲，判断2个时钟的比率关系是否正确。

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>

- ①CPU/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 ( $f_{MX}$ )
- ②Timer4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29-10 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。

#### 29.3.5.1 定时器输入选择寄存器0 (TIS0)

寄存器说明请参考 6.3.8 节。

## 29.3.6 A/D测试功能

IEC60730标准要求对A/D转换器进行测试。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(AN1)、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。

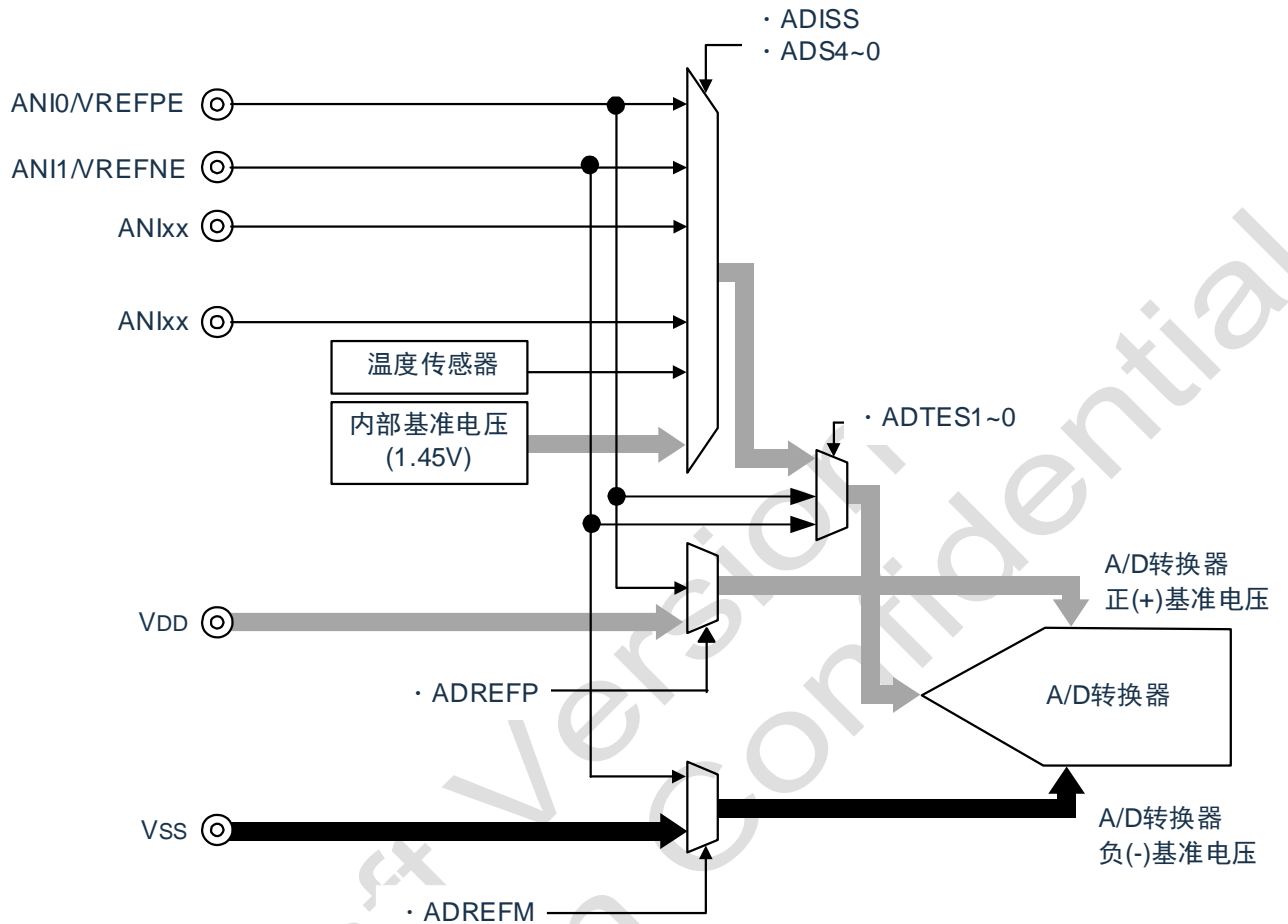
能通过以下步骤确认模拟多路转换器:

- ① 通过ADTES寄存器选择AN1x引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ② 对AN1x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AN1x引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ⑥ 对AN1x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AN1x引脚作为A/D转换对象(ADTES1、ADTES0=0、0)。
- ⑩ 对AN1x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29-11 A/D测试功能的结构



### 29.3.6.1 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13.2.10。

### 29.3.6.2 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13.2.7。

## 29.3.7 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。

### 29.3.7.1 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29-12 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1.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

## 29.3.8 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\_0E4C

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]
--------------

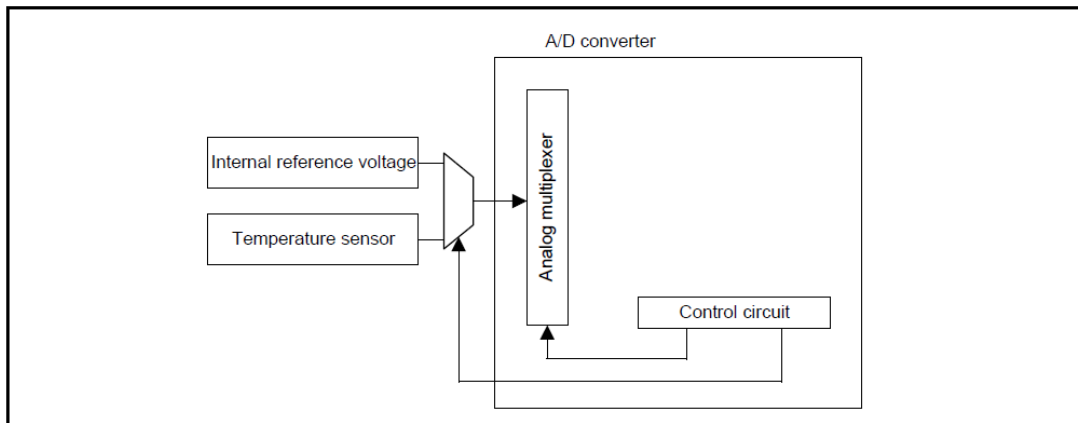


## 第30章 温度传感器和内部参考电压

### 30.1 温度传感器

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 30-1 显示了温度传感器框图。

图 30-1温度传感器框图



### 30.2 温度传感器的寄存器

#### 30.2.1 温度传感器校准数据寄存器 TSN25

地址：0x50016C

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校准数据 1，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。

#### 30.2.2 温度传感器校准数据寄存器 TSN125

地址：0x500168

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
TSN85	-	-	-	-	TSN125[11:0]											-	R	

只读寄存器，用于记录温度传感器的校准数据 2，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。

## 30.3 温度传感器的使用说明

### 30.3.1 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:

$$T = (V_s - V_1) / \text{slope} + T_1$$

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)

Slope: 温度传感器的温度斜率(V/°C),  $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$ .

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:

- 1、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。
- 2、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。
- 3、从两次结果中计算获得温度斜率( $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$ )
- 4、随后, 通过将斜率代入温度特性的公式来获得温度( $T = (V_s - V_1) / \text{slope} + T_1$ ).

### 30.3.2 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  $T_a=25^{\circ}\text{C}$  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)。TSN85 寄存器存储了在  $T_a=125^{\circ}\text{C}$  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：

$$\text{slope} = (V_2 - V_1) / (125 - 25).$$

$$V_1 = 3.0 \times \text{CAL25} / 4096 [\text{V}]$$

$$V_2 = 3.0 \times \text{CAL125} / 4096 [\text{V}]$$

利用以上结果，可根据以下公式计算温度：

$$T = (V_s - V_1) / \text{slope} + 25 [^{\circ}\text{C}]$$

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：

$$T = (V_s - V_1) / \text{slope} + 25 [^{\circ}\text{C}]$$

注：这种方法产生的温度比方法一测量的精度要低。

## 30.4 内部参考电压

Band-gap 参考电压(VBG)是一个内部的固定参考电压，不受外部供电的影响。VBG 输出可以通过内部连接到 ADC 进行转换，因此 VDD 可以通过 VBG 的 ADC 转换结果计算出来。

由于制程的原因每个芯片的 VBG 稍有不同，因此 VDD 的计算会有一点偏差。本产品出厂时内置了 VDD=3.0v 时 VBG 的 A/D 转换结果，用户可以通过这个值及当前 VBG 的 A/D 转换结果精准计算出 VDD 的电压值。

### 30.4.1 VDD校准数据寄存器VDDCDR

地址：0x500C64

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
VDDCDR	-	-	-	-	VDDCDR[11:0]											-	R	

只读寄存器，用于记录 VDD=3.0v 时 VBG 的 A/D 转换结果，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。

### 30.4.2 内部参考电压的使用说明

在本产品中，VDDCDR 寄存器存储了在  $T_a=T_j=25^\circ\text{C}$  和 VDD=3.0v 条件下测量的内部参考电压转换值。利用该结果，用户可根据以下公式计算当前 VDD 电压：

$$VDD = 3.0V * VDDCDR / CALVBG$$

CALVBG: 使用 A/D 转换器，将 VBG 作为 ADC 输入通道，得到的当前 VBG 转换结果。

## 第31章 选项字节

### 31.1 选项字节的功能

BAT32A233的闪存000C0H~000C4H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，000C4H）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定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

注意 与是否使用各功能无关，必须设定选项字节。

#### 31.1.1 用户选项字节（000C0H~000C2H）

##### (1) 000C0H

- 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定
- 看门狗定时器的窗口打开期间的设定
- 看门狗定时器的间隔中断的设定
  - 使用或者不使用间隔中断。

##### (2) 000C1H

- LVD运行模式的设定
  - 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（VLVDH、VLVDL、VLVD）的设定

注意 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）的设定。

##### (3) 000C2H

- 高速内部振荡器的频率设定
  - 从1MHz~32MHz、48MHz、64MHz中选择。

## 31.1.2 闪存数据保护选项字节（000C3H, 000C4H）

○片上调试时闪存数据保护的 control

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。

Draft Version  
Cmsemicon Confidential

## 31.2 用户选项字节的格式

图31-1 用户选项字节（000C0H）的格式

地址：000C0H

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 <sub>IL</sub> 时，产生间隔中断。		

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间注1	
0	-	禁止设定。	
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 <sub>IL</sub> =20kHz(MAX.))
0	0	0	2 <sup>6</sup> /f <sub>IL</sub> (3.2ms)
0	0	1	2 <sup>7</sup> /f <sub>IL</sub> (6.4ms)
0	1	0	2 <sup>8</sup> /f <sub>IL</sub> (12.8ms)
0	1	1	2 <sup>9</sup> /f <sub>IL</sub> (25.6ms)
1	0	0	2 <sup>11</sup> /f <sub>IL</sub> (102.4ms)
1	0	1	2 <sup>13</sup> /f <sub>IL</sub> (409.6ms)
1	1	0	2 <sup>14</sup> /f <sub>IL</sub> (819.2ms)
1	1	1	2 <sup>16</sup> /f <sub>IL</sub> (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.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<sub>IL</sub>：低速内部振荡器的时钟频率

图31-2 用户选项字节（000C1H）的格式(1/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

## • LVD的设定（中断&amp;复位模式）

检测电压			选项字节的设置值						
V <sub>LVDH</sub>		V <sub>LVDL</sub>	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。						

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32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31-2 用户选项字节（000C1H）的格式(2/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设定（复位模式）

检测电压		选项字节的设定值								
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
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				
—			禁止设定上述以外的值。							

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32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31-2 用户选项字节（000C1H）的格式(3/4)

地址：000C1H

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

## • LVD的设定（中断模式）

检测电压		选项字节的设定值									
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定				
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					
—			禁止设定上述以外的值。								

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32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31-2 用户选项字节（000C1H）的格式(4/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD为OFF时的设定（使用RESETB引脚的外部复位输入）

检测电压		选项字节的设定值						
$V_{LVDH}$		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

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）的设定。

备注1.×：忽略

2.有关LVD电路的详细内容，请参照“第31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31-3 用户选项字节（000C2H）的格式

地址：000C2H

	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	64MHz
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					禁止设定。	

注意1. 必须给bit7~5写“1”。

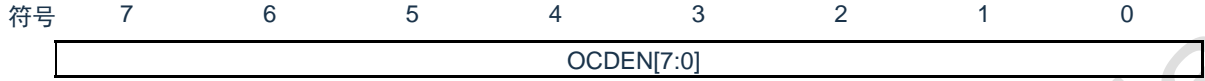
2. 工作频率范围和工作电压范围因闪存的各运行模式而不同。详细内容请参照数据手册的AC特性。

### 31.3 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31-4 闪存数据保护选项字节（000C3H）的格式

地址：000C3H



地址：000C4H



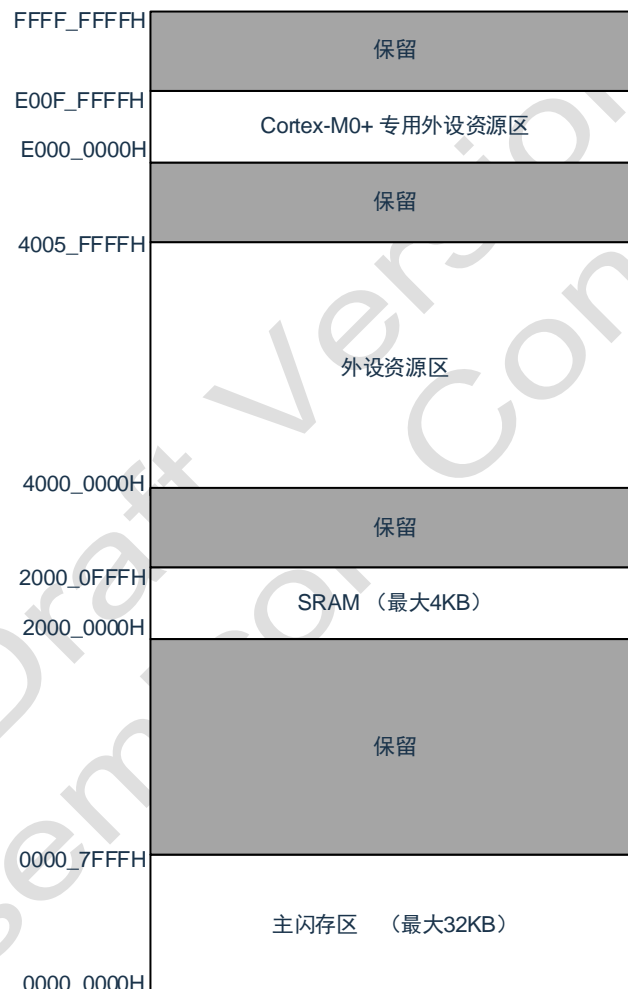
OCDM	OCDEN	闪存数据保护的 control
3C	C3	不允许通过debugger对闪存数据进行操作。
3C以外的值	C3	允许通过debugger对闪存数据进行chip全擦除操作，不允许进行读写操作。
上記以外		允许通过debugger对闪存数据进行读出/写入/擦除操作

## 第32章 FLASH控制

### 32.1 FLASH控制功能描述

BAT32A233制品包含一颗32KByte容量的FLASH存储器，共划分为64个Sector，每个Sector容量为512Byte。可作为程序存储器，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。此外，本模块支持对FLASH 存储器擦写的保护，以及控制寄存器的写保护。

### 32.2 FLASH存储器结构



### 32.3 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)
- Flash模式时间控制寄存器 (FLNVSCNT/FLPRVCNT/FLERVCNT)
- Flash擦写保护控制寄存器 (FLSECPR)

#### 32.3.1 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：00000000H RW

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
上記以外	不允许改写 WRP

### 32.3.2 FLASH操作控制寄存器（FLOPMD1,FLOPMD2）

Flash操作控制寄存器，用于设定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		设定禁止

### 32.3.3 Flash擦除控制寄存器(FLERMD)

Flash擦除控制寄存器，用于设定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
1	0	设定禁止
0	1	chip 擦除 <sup>注</sup>
1	1	设定禁止

注： chip擦除只擦除代码闪存区域，不擦除数据闪存区域。



### 32.3.4 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	0	0	OVF注

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。

### 32.3.5 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	擦除时间设定的选择 <sup>注</sup>
0	使用硬件设定的擦除时间
1	使用软件设定的擦除时间 (FLCERCNT[9:0])

注: 在主时钟为内部高速 OCO,或者外部输入时钟<=20M 时, 可以使用硬件设定时间, 不设定 FLCERCNT。

FLCERCNT[9:0]	软件擦除时间设定
Chip 擦除时间= (CERCNT*2048*Tfclk), 需满足>20ms 的硬件要求	

### 32.3.6 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

FLSERCNT

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

Load	擦除时间设定的选择 注
0	使用硬件设定的擦除时间
1	使用软件设定的擦除时间（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，可以使用硬件设定时间，不设定 FLSERCNT。

FLSERCNT[9:0]	软件擦除时间设定
sector 擦除时间= (SERCNT*256*Tfclk)，需满足 $>4\text{ms}$ 的硬件要求	

### 32.3.7 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

符号	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) 设定 注
0	使用硬件设定的写入时间
1	使用软件设定的擦除时间 (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, 可以使用硬件设定时间, 不设定 FLPROCNT。

FLPROCNT[8:0]	软件擦除时间设定
写入时间= (PROCNT*4*Tfclk), 需满足 $>24\mu\text{s}$ 的硬件要求	

Load1	写入动作建立时间 (Tpgs) 设定 注
0	使用硬件设定的写入动作建立时间
1	使用软件设定的擦除时间 (FLPGSCNT8:0))

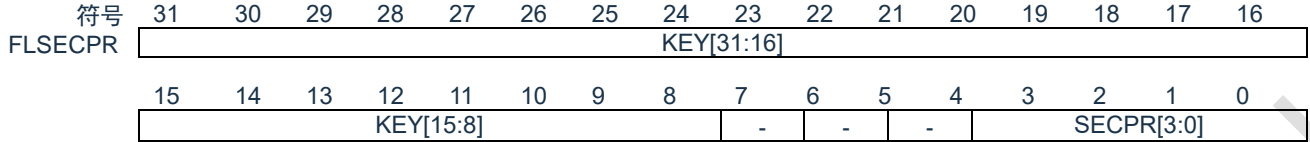
注: 在主时钟为内部高速 OCO, 或者外部输入时钟 $\leq 20\text{M}$  时, 可以使用硬件设定时间, 不设定 FLPGSCNT。

FLPGSCNT[8:0]	软件擦除时间设定
写入动作建立时间= (PGSCNT*Tfclk), 需满足 $>5\mu\text{s}$ 的硬件要求	

### 32.3.8 Flash擦写保护控制寄存器（FLSECPR）

当Sector被保护时，对该Sector进行的擦写操作均无效。

地址：0x40020210 复位后：00000000H R/W



KEY	寄存器 SECPR 写保护
5AA5F1	允许改写 SECPR[3:0]
上述以外	不允许改写 SECPR[3:0]

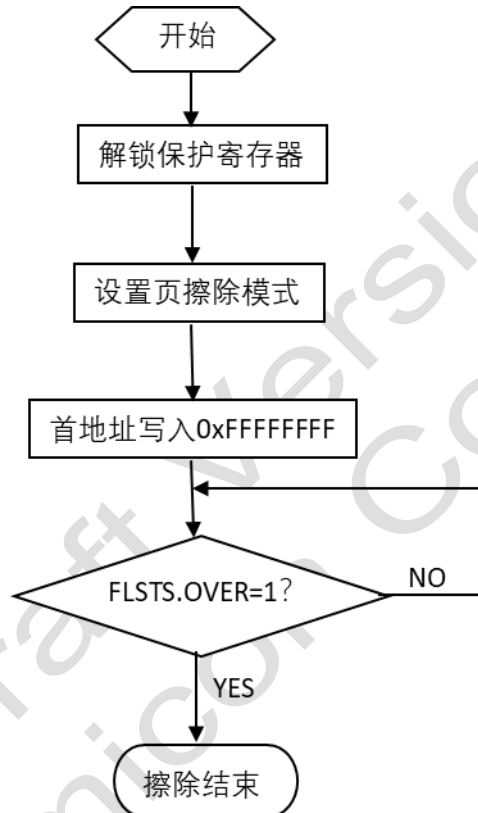
SECPR	寄存器 SECPR 写保护
0001	00_0000H~00_0FFFH 不能擦写
0010	00_0000H~00_1FFFH 不能擦写
0011	00_0000H~00_3FFFH 不能擦写
0100	00_0000H~00_7FFFH 不能擦写
0101	00_0000H~00_FFFFH 不能擦写
0110	00_0000H~01_FFFFH 不能擦写
0111	00_0000H~03_FFFFH 不能擦写
1000	00_0000H~05_FFFFH 不能擦写
1001	00_0000H~07_FFFFH 不能擦写 (全部 512 个 sector 不允许擦写)
上述以外	sector 没有保护，允许擦写

## 32.4 FLASH操作方法

### 32.4.1 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；
- 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) 进行下次操作前，软件置“1”来清除FLSTS。



## 32.4.2 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。

## 32.4.3 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。

## 32.5 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。

## 32.6 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
V0.1.0	2022年4月	初始版本
V0.1.1	2022年5月	(1) 寄存器PIOR1, 当bit[1:0]=0x10时, 由“和P41复用”变更为“禁止设定”。 (2) 寄存器TAMR0的TCK位禁止设置为0x110。 (3) 寄存器TCCR1中删除计数源选择“FHOCO”, 并删除相关注释1和2。 (4) 变更9.4.1节“定时器C的时钟源”部分的表述。 (5) 第10章相关图表中删除对FHOCO 用作计数源的选择。 (6) 10.3.12节图10-19, 互补PWM模式的TMCR寄存器由TMCR0变更为TMCR1, 同时增加“注1: 必须给TMCR1寄存器的TCK0~TCK2位和TMCR0寄存器的TCK0~TCK2位设置相同的值(相同的计数源)。” (7) 表10-18中, 定时器M中断请求标志和中断屏蔽标志寄存器变更。 (8) 14.3.4节D/A转换器的模拟输出电压计算公式中的“VDD”变更为“VREF”, 并增加说明“VREF由寄存器DAM的DVREF位确定”。 (9) 17.3.4节寄存器SCR中增加bit10“EOCmn”位及位说明。 (10) 17.3.15节SSEm寄存器中删除bit1“SSEm1”位, 并增加“备注: m=0, n=0”。 (11) SSPI片选引脚名称由“SSmn”和“SSImn”统一为“SSmn” (12) 图17-72、图17-81、图17-87中删除bit1位“SSIEm1”。 (13) 17.5与17.6章节SCR寄存器设定实例图中增加bit10“EOCmn”位的设置。 (14) 第20章ISC寄存器的名称由“输入开关寄存器”变更为“输入切换控制寄存器”, 与第6章和第17章一致。 (15) 其他: 文字表述修改、图表序号修改、格式调整等。
V0.1.2	2023年9月	(1) 修改32.3.3 Flash擦除控制寄存器(FLERMD)章节描述 (2) 修改32.3.4 Flash状态寄存器(FLSTS)章节中寄存器值 (3) 修改32.4.1 页擦除 (sector erase) 章节内容 (4) 优化20章节中标题格式 (5) 修改第28章电压检测电路中图28-1和图28-6的错误 (6) 更正2.3.1章节中PM13寄存器的复位值
V0.1.3	2023年10月	(1) 更正2.3.1章节中PM13寄存器的复位值 (2) 更正24.2章节中中断源一览表内容描述 (3) 修改第28章电压检测电路中图28-1和图28-6中的错误以及LVIS寄存器地址的错误 (4) 将图10-63中的IMFB位更正为IMFD位
V0.1.4	2023年10月	更正内容中的引用链接