



# CMS32L032参考手册

基于ARM® Cortex®-M0+的低功耗32位微控制器

V1.0.3

请注意以下有关CMS知识产权政策

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

\* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

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

## 文档使用说明

本手册是CMS32L032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,并对每种功能模块都有专门的章节进行介绍。

技术参考手册是针对这一系列产品所有功能模块的说明，若要了解特定型号产品的特征说明（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：

CMS32L032xx: CMS32L032\_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>14</b>
1.1 概述 .....	14
1.2 Cortex-M0+内核特性 .....	14
1.3 调试特性 .....	14
1.4 SWD 接口引脚 .....	16
1.5 ARM 参考文档 .....	17
<b>第2章 引脚功能</b> .....	<b>18</b>
2.1 端口功能 .....	18
2.2 端口复用功能 .....	18
2.3 控制端口功能的寄存器 .....	19
2.3.1 端口模式寄存器 (PMxx) .....	22
2.3.2 端口寄存器 (Pxx) .....	23
2.3.3 端口置位控制寄存器 (PSETxx) .....	24
2.3.4 端口清零控制寄存器 (PCLRxx) .....	25
2.3.5 上拉电阻选择寄存器 (PUxx) .....	26
2.3.6 下拉电阻选择寄存器 (PDxx) .....	27
2.3.7 端口输出模式寄存器 (POMxx) .....	28
2.3.8 端口模式控制寄存器 (PMCxx) .....	29
2.3.9 端口回读寄存器 (PREADxx) .....	30
2.3.10 端口复用功能配置寄存器 (PxxCFG) .....	31
2.3.11 特殊功能端口 RESINB 说明 .....	32
<b>第3章 系统结构</b> .....	<b>33</b>
3.1 概述 .....	33
3.2 系统地址划分 .....	34
<b>第4章 时钟发生电路</b> .....	<b>36</b>
4.1 时钟发生电路的功能 .....	36
4.2 时钟发生电路的结构 .....	38
4.3 控制时钟发生电路的寄存器 .....	41
4.3.1 时钟运行模式控制寄存器 (CMC) .....	41
4.3.2 系统时钟控制寄存器 (CKC) .....	43
4.3.3 时钟运行状态控制寄存器 (CSC) .....	44
4.3.4 振荡稳定时间计数器的状态寄存器 (OSTC) .....	45
4.3.5 振荡稳定时间选择寄存器 (OSTS) .....	47
4.3.6 外围允许寄存器 0、1 (PER0、PER1) .....	48
4.3.7 副系统时钟提供模式控制寄存器 (OSMC) .....	52
4.3.8 高速内部振荡器的频率选择寄存器 (HOCODIV) .....	53
4.3.9 高速内部振荡器的微调寄存器 (HIOTRM) .....	54
4.3.10 副系统时钟选择寄存器 (SUBCKSEL) .....	55

4.3.11	供电模式控制保护寄存器 (PMUKEY) .....	56
4.3.12	供电模式控制寄存器 (PMUCTL) .....	56
4.4	系统时钟振荡电路 .....	57
4.4.1	X1 振荡电路 .....	57
4.4.2	XT1 振荡电路 .....	58
4.4.3	高速内部振荡器 .....	61
4.4.4	低速内部振荡器 .....	61
4.5	时钟发生电路的运行 .....	62
4.6	时钟控制 .....	64
4.6.1	高速内部振荡器的设置例子 .....	64
4.6.2	X1 振荡电路的设置例子 .....	66
4.6.3	XT1 振荡电路的设置例子 .....	67
4.6.4	CPU 时钟的状态转移图 .....	68
4.6.5	CPU 时钟转移前的条件和转移后的处理 .....	73
4.6.6	CPU 时钟和主系统时钟的切换所需时间 .....	75
4.6.7	时钟振荡停止前的条件 .....	76
4.7	高速内振校正功能 .....	77
4.7.1	高速内振自调整功能 .....	77
4.7.2	寄存器说明 .....	78
4.7.3	动作说明 .....	79
4.7.4	使用注意事项 .....	83
<b>第5章</b>	<b>通用定时器单元Timer4 .....</b>	<b>84</b>
5.1	通用定时器单元的功能 .....	85
5.1.1	独立通道运行功能 .....	85
5.1.2	多通道联动运行功能 .....	87
5.1.3	8 位定时器运行功能 (只限于单元 0 的通道 1 和通道 3) .....	88
5.1.4	LIN-bus 支持功能 (只限于单元 0 的通道 3) .....	88
5.2	通用定时器单元的结构 .....	89
5.2.1	通用定时器单元 0 寄存器列表 .....	92
5.2.2	通用定时器单元 1 寄存器列表 .....	93
5.2.3	定时器计数寄存器 mn (TCRmn) .....	94
5.2.4	定时器数据寄存器 mn (TDRmn) .....	95
5.3	控制通用定时器单元的寄存器 .....	96
5.3.1	外围允许寄存器 0 (PER0) .....	97
5.3.2	定时器时钟选择寄存器 m (TPSm) .....	98
5.3.3	定时器模式寄存器 mn (TMRmn) .....	101
5.3.4	定时器状态寄存器 mn (TSRmn) .....	106
5.3.5	定时器通道允许状态寄存器 m (TEm) .....	107
5.3.6	定时器通道开始寄存器 m (TSM) .....	108
5.3.7	定时器通道停止寄存器 m (TTm) .....	109
5.3.8	定时器输入输出选择寄存器 (TIOS0) .....	110
5.3.9	定时器输出允许寄存器 m (TOEm) .....	111

5.3.10	定时器输出寄存器 m (TOm)	112
5.3.11	定时器输出电平寄存器 m (TOLm)	113
5.3.12	定时器输出模式寄存器 m (TOMm)	114
5.3.13	噪声滤波器允许寄存器 1 (NFEN1)	115
5.3.14	噪声滤波器允许寄存器 2 (NFEN2)	116
5.3.15	控制定时器输入/输出引脚端口功能的寄存器	117
5.4	通用定时器单元的基本规则	118
5.4.1	多通道联动运行功能的基本规则	118
5.4.2	8 位定时器运行功能的基本规则 (只限于单元 0 的通道 1 和通道 3)	120
5.5	计数器的运行	121
5.5.1	计数时钟 (F <sub>TCLK</sub> )	121
5.5.2	计数器的开始时序	123
5.5.3	计数器的运行	124
5.6	通道输出 (TOmn 引脚) 的控制	129
5.6.1	TOmn 引脚输出电路的结构	129
5.6.2	TOmn 引脚的输出设定	130
5.6.3	通道输出运行的注意事项	131
5.6.4	TOmn 位的一次性操作	135
5.6.5	有关开始计数时的定时器中断和 TOmn 引脚输出	136
5.7	定时器输入 (TImn) 的控制	137
5.7.1	TImn 引脚输入电路的结构	137
5.7.2	噪声滤波器	138
5.7.3	操作通道输入时的注意事项	139
5.8	通用定时器单元的独立通道运行功能	140
5.8.1	作为间隔定时器/方波输出的运行	140
5.8.2	作为外部事件计数器的运行	144
5.8.3	作为分频器的运行	147
5.8.4	作为输入脉冲间隔测量的运行	150
5.8.5	作为输入信号高低电平宽度测量的运行	153
5.8.6	作为延迟计数器的运行	157
5.9	通用定时器单元的多通道联动运行功能	160
5.9.1	作为单触发脉冲输出功能的运行	160
5.9.2	作为 PWM 功能的运行	167
5.9.3	作为多重 PWM 输出功能的运行	174
<b>第6章</b>	<b>EPWM输出控制电路的功能</b>	<b>182</b>
6.1	输出控制电路的结构	182
6.2	EPWM 输出控制电路的控制寄存器	183
6.2.1	外围允许寄存器 1 (PER1)	183
6.2.2	EPWM 输入源选择寄存器(EPWMSRC)	183
6.2.3	EPWM 强制截断输入选择寄存器(EPWMSTC)	184
6.2.4	EPWM 输出控制寄存器(EPWMCTL)	185
6.2.5	EPWM 强制截断输出选择寄存器(EPWMSTL)	186

6.2.6	EPWM 状态寄存器(EPWMSTR).....	186
6.2.7	EPWM 死区控制寄存器(EPWMDTC) .....	187
6.2.8	EPWM 输出引脚的端口功能的控制寄存器 .....	187
6.3	EPWM 输出控制电路的运行.....	188
6.3.1	初始设定.....	188
6.3.2	通常运行.....	189
6.3.3	强制截断处理 .....	190
6.4	无刷直流电机的控制例子 .....	191
6.4.1	硬件连接例子 .....	191
6.4.2	三相无刷直流电机的控制时序.....	192
6.4.3	寄存器的设定例子 .....	193
6.5	步进电机的控制例子.....	194
6.5.1	硬件连接例子 .....	194
6.5.2	控制方法.....	195
6.5.3	寄存器的设定例子 .....	196
<b>第7章</b>	<b>实时时钟 .....</b>	<b>197</b>
7.1	实时时钟的功能 .....	197
7.2	实时时钟的结构.....	197
7.3	控制实时时钟的寄存器 .....	199
7.3.1	外围允许寄存器 0 (PER0) .....	200
7.3.2	实时时钟选择寄存器 (RTCCL) .....	201
7.3.3	实时时钟控制寄存器 0 (RTCC0) .....	202
7.3.4	实时时钟控制寄存器 1 (RTCC1) .....	203
7.3.5	时钟误差校正寄存器 (SUBCUD) .....	205
7.3.6	秒计数寄存器 (SEC) .....	206
7.3.7	分钟计数寄存器 (MIN) .....	206
7.3.8	小时计数寄存器 (HOUR) .....	207
7.3.9	日计数寄存器 (DAY) .....	209
7.3.10	星期计数寄存器 (WEEK) .....	210
7.3.11	月计数寄存器 (MONTH) .....	211
7.3.12	年计数寄存器 (YEAR) .....	211
7.3.13	闹钟分钟寄存器 (ALARMWM) .....	212
7.3.14	闹钟小时寄存器 (ALARMWH) .....	212
7.3.15	闹钟星期寄存器 (ALARMWW) .....	213
7.3.16	端口模式寄存器和端口寄存器.....	213
7.4	实时时钟的运行.....	214
7.4.1	实时时钟的运行开始 .....	214
7.4.2	开始运行后睡眠模式的转移.....	215
7.4.3	实时时钟计数器的读写 .....	216
7.4.4	实时时钟的闹钟设定 .....	218
7.4.5	实时时钟的 1Hz 输出.....	219
7.4.6	实时时钟的时钟误差校正例子.....	220

第8章	15位间隔定时器 .....	222
8.1	15位间隔定时器的功能 .....	222
8.2	15位间隔定时器的结构 .....	222
8.3	控制15位间隔定时器的寄存器 .....	223
8.3.1	外围允许寄存器0 (PER0) .....	223
8.3.2	实时时钟选择寄存器 (RTCCL) .....	224
8.3.3	15位间隔定时器的控制寄存器 (ITMC) .....	225
8.4	15位间隔定时器的运行 .....	226
8.4.1	15位间隔定时器的运行时序 .....	226
8.4.2	从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移 .....	227
第9章	时钟输出/蜂鸣器输出控制电路 .....	228
9.1	时钟输出/蜂鸣器输出控制电路的功能 .....	228
9.2	时钟输出/蜂鸣器输出控制电路的结构 .....	229
9.3	控制时钟输出/蜂鸣器输出控制电路的寄存器 .....	230
9.3.1	时钟输出选择寄存器 n (CKSn) .....	230
9.3.2	控制时钟输出/蜂鸣器输出引脚端口功能的寄存器 .....	231
9.4	时钟输出/蜂鸣器输出控制电路的运行 .....	232
9.4.1	输出引脚的运行 .....	232
9.5	时钟输出/蜂鸣器输出控制电路的注意事项 .....	232
第10章	看门狗定时器 .....	233
10.1	看门狗定时器的功能 .....	233
10.2	看门狗定时器的结构 .....	233
10.3	控制看门狗定时器的寄存器 .....	235
10.3.1	看门狗定时器的允许寄存器 (WDTE) .....	235
10.3.2	LOCKUP 控制寄存器 (LOCKCTL) 及其保护寄存器 (PRCR) .....	236
10.3.3	WDTCFG 配置寄存器 (WDTCFG0/1/2/3) .....	237
10.4	看门狗定时器的运行 .....	238
10.4.1	看门狗定时器的运行控制 .....	238
10.4.2	看门狗定时器上溢时间的设定 .....	239
10.4.3	看门狗定时器窗口打开期间的设定 .....	240
10.4.4	看门狗定时器间隔中断的设定 .....	241
10.4.5	LOCKUP 期间看门狗定时器的运行 .....	241
10.4.6	WDTCFG 未配置时看门狗定时器的运行 .....	241
第11章	A/D转换器 .....	242
11.1	A/D转换器的功能 .....	242
11.2	控制A/D转换器的寄存器 .....	244
11.2.1	外围允许寄存器0 (PER0) .....	245
11.2.2	A/D转换器的模式寄存器0 (ADM0) .....	246
11.2.3	A/D转换器的模式寄存器1 (ADM1) .....	251
11.2.4	A/D转换器的模式寄存器2 (ADM2) .....	252
11.2.5	A/D转换器的触发模式寄存器 (ADTRG) .....	253

11.2.6	模拟输入通道指定寄存器 (ADS)	254
11.2.7	12 位 A/D 转换结果寄存器 (ADCR)	256
11.2.8	8 位 A/D 转换结果寄存器 (ADCRH)	257
11.2.9	转换结果比较上限值设定寄存器 (ADUL)	257
11.2.10	转换结果比较下限值设定寄存器 (ADLL)	258
11.2.11	A/D 采样时间延长寄存器 (ADSMPWAIT)	258
11.2.12	控制模拟输入引脚端口功能的寄存器	258
11.3	输入电压和转换结果	259
11.4	A/D 转换器的运行模式	260
11.4.1	软件触发模式 (选择模式、连续转换模式)	260
11.4.2	软件触发模式 (选择模式、单次转换模式)	261
11.4.3	软件触发模式 (扫描模式、连续转换模式)	262
11.4.4	软件触发模式 (扫描模式、单次转换模式)	263
11.4.5	硬件触发无等待模式 (选择模式、连续转换模式)	264
11.4.6	硬件触发无等待模式 (选择模式、单次转换模式)	265
11.4.7	硬件触发无等待模式 (扫描模式、连续转换模式)	266
11.4.8	硬件触发无等待模式 (扫描模式、单次转换模式)	267
11.4.9	硬件触发等待模式 (选择模式、连续转换模式)	268
11.4.10	硬件触发等待模式 (选择模式、单次转换模式)	269
11.4.11	硬件触发等待模式 (扫描模式、连续转换模式)	270
11.4.12	硬件触发等待模式 (扫描模式、单次转换模式)	271
11.5	A/D 转换器的设定流程图	272
11.5.1	软件触发模式的设定	272
11.5.2	硬件触发无等待模式的设定	273
11.5.3	硬件触发等待模式的设定	274
<b>第12章 通用串行通信单元</b>		<b>275</b>
12.1	通用串行通信单元的功能	276
12.1.1	3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)	276
12.1.2	UART (UART0、UART1)	277
12.2	通用串行通信单元的结构	278
12.2.1	移位寄存器	280
12.2.2	串行数据寄存器 mn (SDRmn)	280
12.3	控制通用串行通信单元的寄存器	282
12.3.1	外围允许寄存器 0 (PER0)	284
12.3.2	串行时钟选择寄存器 m (SPSm)	285
12.3.3	串行模式寄存器 mn (SMRmn)	286
12.3.4	串行通信运行设定寄存器 mn (SCRmn)	288
12.3.5	串行数据寄存器 mn (SDRmn)	291
12.3.6	串行标志清除触发寄存器 mn (SIRmn)	292
12.3.7	串行状态寄存器 mn (SSRmn)	293
12.3.8	串行通道开始寄存器 m (SSm)	295
12.3.9	串行通道停止寄存器 m (STm)	296



12.3.10	串行通道允许状态寄存器 m (SEm)	297
12.3.11	串行输出允许寄存器 m (SOEm)	298
12.3.12	串行输出寄存器 m (SOM)	299
12.3.13	串行输出电平寄存器 m (SOLm)	300
12.3.14	输入切换控制寄存器 (ISC)	301
12.3.15	噪声滤波器允许寄存器 0 (NFEN0)	302
12.3.16	控制串行输入/输出引脚端口功能的寄存器	303
12.4	运行停止模式	304
12.4.1	以单元为单位停止运行的情况	304
12.4.2	按通道停止运行的情况	305
12.5	3线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11) 通信的运行	306
12.5.1	主控发送	307
12.5.2	主控接收	315
12.5.3	主控的发送和接收	324
12.5.4	从属发送	332
12.5.5	从属接收	340
12.5.6	从属的发送和接收	346
12.5.7	传送时钟频率的计算	355
12.5.8	在 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11) 通信过程中发生错误时的处理步骤	357
12.6	从属选择输入功能的时钟同步串行通信的运行	358
12.6.1	从属发送	361
12.6.2	从属接收	371
12.6.3	从属的发送和接收	378
12.6.4	传送时钟频率的计算	388
12.6.5	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	389
12.7	UART (UART0~UART1) 通信的运行	390
12.7.1	UART 发送	391
12.7.2	UART 接收	399
12.7.3	波特率的计算	406
12.7.4	在 UART (UART0~UART1) 通信过程中发生错误时的处理步骤	410
12.8	LIN 通信的运行	411
12.8.1	LIN 发送	411
12.8.2	LIN 接收	414
<b>第13章 串行接口SPI</b>		<b>419</b>
13.1	串行接口 SPI 的功能	419
13.2	串行接口 SPI 的结构	419
13.3	控制串行接口 SPI 的寄存器	420
13.3.1	外围允许寄存器 0 (PER0)	420
13.3.2	SPI 操作模式寄存器 (SPIM)	421
13.3.3	SPI 时钟选择寄存器 (SPIC)	422
13.3.4	发送缓冲寄存器 (SDRO)	423
13.3.5	接收缓冲寄存器 (SDRI)	423

13.3.6	SPI 引脚的端口功能的控制寄存器 .....	423
13.4	串行接口 SPI 的操作 .....	424
13.4.1	主控的发送和接收 .....	424
13.4.2	主控的接收 .....	427
13.4.3	从属的发送和接收 .....	430
13.4.4	从属的接收 .....	433
<b>第14章</b>	<b>串行接口 IICA .....</b>	<b>436</b>
14.1	串行接口 IICA 的功能 .....	436
14.2	串行接口 IICA 的结构 .....	439
14.3	控制串行接口 IICA 的寄存器 .....	442
14.3.1	外围允许寄存器 0 (PER0) .....	443
14.3.2	IICA 控制寄存器 n0 (IICCTLn0) .....	443
14.3.3	IICA 状态寄存器 n (IICSn) .....	448
14.3.4	IICA 标志寄存器 n (IICFn) .....	451
14.3.5	IICA 控制寄存器 n1 (IICCTLn1) .....	453
14.3.6	IICA 低电平宽度设定寄存器 n (IICWLn) .....	455
14.3.7	IICA 高电平宽度设定寄存器 n (IICWHn) .....	455
14.3.8	控制 IICA 引脚端口功能的寄存器 .....	455
14.4	I <sup>2</sup> C 总线模式的功能 .....	456
14.4.1	引脚结构 .....	456
14.4.2	通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法 .....	457
14.5	I <sup>2</sup> C 总线的定义和控制方法 .....	459
14.5.1	开始条件 .....	460
14.5.2	地址 .....	461
14.5.3	传送方向的指定 .....	461
14.5.4	应答 (ACK) .....	462
14.5.5	停止条件 .....	463
14.5.6	等待 .....	464
14.5.7	等待的解除方法 .....	466
14.5.8	中断请求 (INTIICAn) 的产生时序和等待控制 .....	467
14.5.9	地址匹配的检测方法 .....	469
14.5.10	错误的检测 .....	469
14.5.11	扩展码 .....	469
14.5.12	仲裁 .....	470
14.5.13	唤醒功能 .....	472
14.5.14	通信预约 .....	475
14.5.15	其他注意事项 .....	478
14.5.16	通信运行 .....	479
14.5.17	I <sup>2</sup> C 中断请求 (INTIICAn) 的产生时序 .....	487
14.6	时序图 .....	508
<b>第15章</b>	<b>联动控制器(EVENTC) .....</b>	<b>523</b>
15.1	EVENTC 的功能 .....	523

15.2	EVENTC 的结构 .....	523
15.3	控制寄存器 .....	524
15.3.1	输出目标选择寄存器 n (ELSELRn) (n=00~12) .....	525
15.4	EVENTC 的运行 .....	527
<b>第16章 中断功能 .....</b>		<b>528</b>
16.1	中断功能的种类 .....	528
16.2	中断源和结构 .....	528
16.3	控制中断功能的寄存器 .....	533
16.3.1	中断请求标志寄存器 (IF00~IF31) .....	533
16.3.2	中断屏蔽标志寄存器 (MK00~MK31) .....	534
16.3.3	外部中断上升沿允许寄存器 (EGP0)、外部中断下降沿允许寄存器 (EGN0) .....	537
16.4	中断处理的操作 .....	538
16.4.1	可屏蔽中断请求的接受 .....	538
16.4.2	不可屏蔽中断请求的接受 .....	538
<b>第17章 键中断功能 .....</b>		<b>539</b>
17.1	键中断的功能 .....	539
17.2	键中断的结构 .....	540
17.3	控制键中断的寄存器 .....	541
17.3.1	键返回模式寄存器 (KRM) .....	541
17.3.2	端口模式寄存器 (PMx) .....	541
<b>第18章 待机功能 .....</b>		<b>542</b>
18.1	待机功能 .....	542
18.2	睡眠模式 .....	543
18.2.1	睡眠模式的设定 .....	543
18.2.2	睡眠模式的解除 .....	546
18.3	深度睡眠模式 .....	547
18.3.1	深度睡眠模式的设定 .....	547
18.3.2	深度睡眠模式的解除 .....	549
18.4	部分掉电的深度睡眠模式 .....	550
18.4.1	部分掉电的深度睡眠模式的设定 .....	550
18.4.2	部分掉电的深度睡眠模式的解除 .....	552
<b>第19章 复位功能 .....</b>		<b>553</b>
19.1	确认复位源的寄存器 .....	557
19.1.1	复位控制标志寄存器 (RESF) .....	557
<b>第20章 上电复位电路 .....</b>		<b>559</b>
20.1	上电复位电路的功能 .....	559
20.2	上电复位电路的结构 .....	559
20.3	上电复位电路的运行 .....	560
<b>第21章 电压检测电路 .....</b>		<b>563</b>
21.1	电压检测电路的功能 .....	563

21.2	电压检测电路的结构.....	564
21.3	控制电压检测电路的寄存器.....	564
21.3.1	电压检测寄存器 (LVIM) .....	565
21.3.2	电压检测电平寄存器 (LVIS) .....	566
21.4	电压检测电路的运行.....	570
21.4.1	用作复位模式时的设定.....	570
21.4.2	用作中断模式时的设定.....	571
21.4.3	用作中断&复位模式时的设定.....	573
21.5	电压检测电路的注意事项 .....	578
<b>第22章 安全功能 .....</b>		<b>580</b>
22.1	安全功能的概要 .....	580
22.2	安全功能使用的寄存器 .....	580
22.3	安全功能的运行 .....	581
22.3.1	闪存 CRC 运算功能 (高速 CRC) .....	581
22.3.2	CRC 运算功能 (通用 CRC) .....	584
22.3.3	SFR 保护功能.....	587
22.3.4	频率检测功能 .....	588
22.3.5	A/D 测试功能.....	589
22.3.6	输入/输出引脚的数字输出信号电平检测功能.....	590
22.3.7	产品唯一身份标识寄存器 .....	591
<b>第23章 温度传感器.....</b>		<b>592</b>
23.1	温度传感器的功能 .....	592
23.2	温度传感器的寄存器.....	592
23.2.1	温度传感器校准数据寄存器 TSN25 .....	592
23.3	温度传感器的使用说明 .....	593
<b>第24章 选项字节 .....</b>		<b>594</b>
24.1	选项字节的功能 .....	594
24.1.1	用户选项字节 (000C0H~000C2H) .....	594
24.1.2	闪存数据保护选项字节 (000C3H, 500004H) .....	595
24.2	用户选项字节的格式.....	596
24.3	闪存数据保护选项字节的格式 .....	602
<b>第25章 FLASH控制.....</b>		<b>603</b>
25.1	FLASH 控制功能描述 .....	603
25.2	FLASH 存储器结构.....	603
25.3	控制 FLASH 的寄存器 .....	604
25.3.1	Flash 写保护寄存器 (FLPROT).....	604
25.3.2	FLASH 操作控制寄存器 (FLOPMD1, FLOPMD2).....	605
25.3.3	Flash 擦除控制寄存器 (FLERMD).....	605
25.3.4	Flash 状态寄存器 (FLSTS).....	606
25.3.5	Flash 全片擦除时间控制寄存器 (FLCERCNT).....	606
25.3.6	Flash 页擦除时间控制寄存器 (FLSERCNT) .....	607

---

25.3.7	Flash 写入时间控制寄存器 (FLPROCNT) .....	608
25.4	FLASH 操作方法 .....	609
25.4.1	页擦除 (sectorerase) .....	609
25.4.2	全片擦除 (chiperase) .....	610
25.4.3	编程 (word program) .....	610
25.5	闪存读取 .....	610
25.6	FLASH 操作的注意事项.....	610
附录	修订记录 .....	611

# 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。

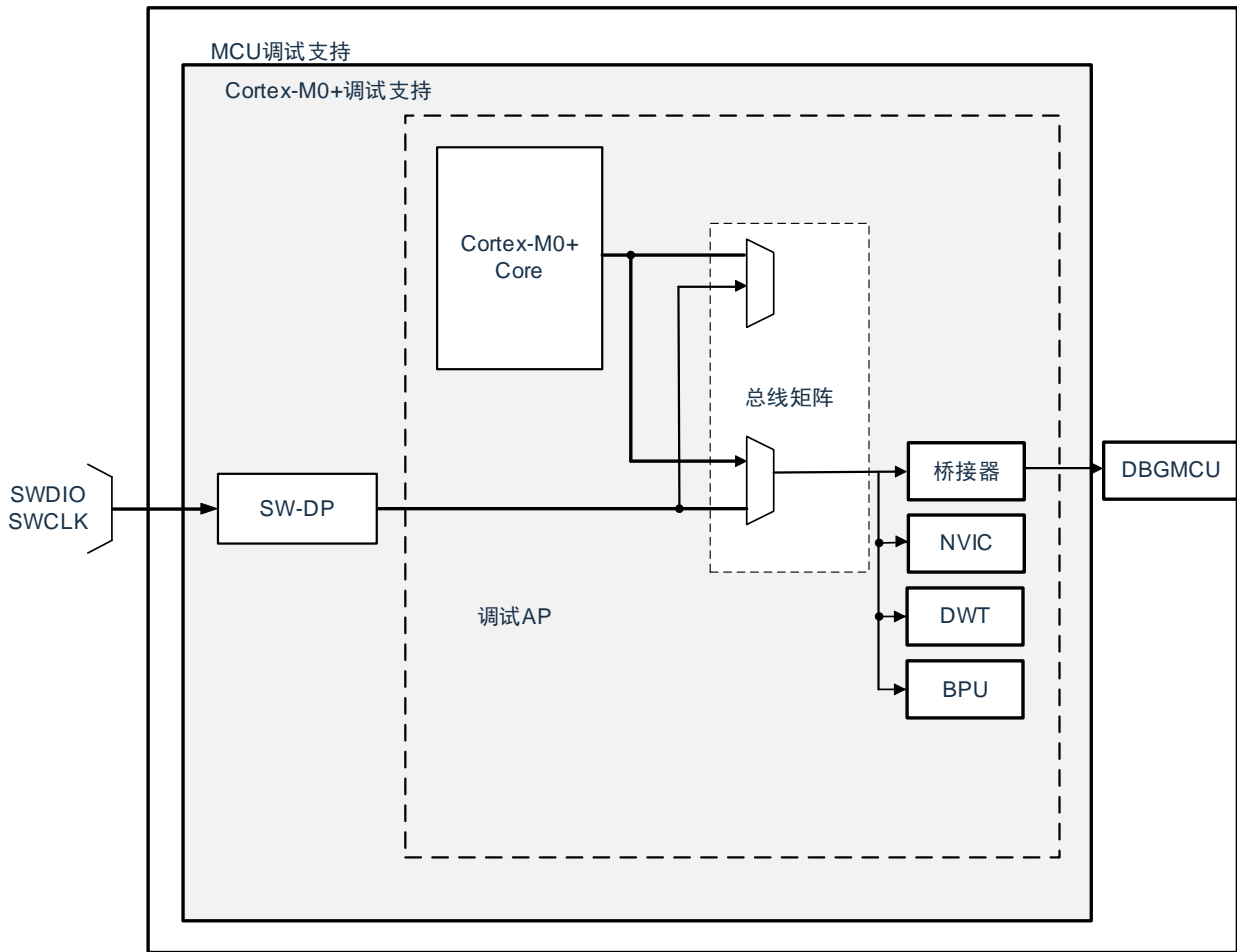
## 1.2 Cortex-M0+内核特性

- ARM Cortex-M0+处理器是32位RISC内核，采用2级流水线，仅支持特权模式
- 32周期硬件乘法器
- 嵌套向量中断控制器（NVIC）
  - 1个不可屏蔽中断(NMI)
  - 支持32个可屏蔽中断请求(IRQ)
  - 4个中断优先级
- 系统定时器SysTick是一个24位倒计时定时器，可选择F<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	P21/P37
SWDIO	串行数据输入 /输出	P20/P36

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。

地址：0x4001B004

复位后：00H

R/W

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

SWDIS	SWD 调试接口禁用
0	SWD 调试接口使能。在连接调试器的状态下，P20/P36 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。P20/P36 可用作 GPIO

FRZEN0	在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），定时器系周边模块动作/停止 <sup>注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>注2</sup>
0	周边动作
1	周边停止

注1：本制品的定时器系周边模块包括：通用定时器单元Timer4

注2：本制品的通信系周边模块包括：通信串行通信单元，串行IICA



## 1.5 ARM参考文档

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

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

## 第2章 引脚功能

### 2.1 端口功能

参见各产品系列的[数据手册](#)。

### 2.2 端口复用功能

参见各产品系列的[数据手册](#)。

## 2.3 控制端口功能的寄存器

通过以下寄存器控制端口：

- (1) 端口模式寄存器 (PMxx)
- (2) 端口寄存器 (Pxx)
- (3) 上拉电阻选择寄存器 (PUxx)
- (4) 下拉电阻选择寄存器 (PDxx)
- (5) 端口输出模式寄存器 (POMx)
- (6) 端口模式控制寄存器 (PMCxx)
- (7) 端口置位控制寄存器 (PSETxx)
- (8) 端口清零控制寄存器 (PCLRxx)
- (9) 端口回读寄存器 (PREADxx)
- (10) 端口输出复用功能配置寄存器 (PxxCFG)

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

端口		位名									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	PDxx 寄存器	POMxx 寄存器	PMCxx 寄存器	PxxCFG 寄存器		
端口 0	0	PM00	P00	PSET00	PCLR00	PU00	—	POM00	PMC00	P00CFG	○	○
	1	PM01	P01	PSET01	PCLR01	PU01	PD01	POM01	PMC01	P01CFG	○	○
	2	PM02	P02	PSET02	PCLR02	PU02	PD02	POM02	PMC02	P02CFG	○	○
端口 1	0	PM10	P10	PSET10	PCLR10	PU10	PD10	POM10	PMC10	P10CFG	○	—
	1	PM11	P11	PSET11	PCLR11	PU11	PD11	POM11	PMC11	P11CFG	○	○
	2	PM12	P12	PSET12	PCLR12	—	—	POM12	PMC12	P12CFG	○	○
	3	PM13	P13	PSET13	PCLR13	—	—	POM13	PMC13	P13CFG	○	○
端口 2	0	PM20	P20	PSET20	PCLR20	PU20	PD20	POM20	PMC20	P20CFG	○	○
	1	PM21	P21	PSET21	PCLR21	PU21	PD21	POM21	PMC21	P21CFG	○	○
	2	PM22	P22	PSET22	PCLR22	PU22	PD22	POM22	PMC22	P22CFG	○	—
	3	PM23	P23	PSET23	PCLR23	PU23	PD23	POM23	PMC23	P23CFG	○	—
	4	PM24	P24	PSET24	PCLR24	PU24	PD24	POM24	PMC24	P24CFG	○	○
	5	PM25	P25	PSET25	PCLR25	PU25	PD25	POM25	PMC25	P25CFG	○	○
	6	PM26	P26	PSET26	PCLR26	PU26	PD26	POM26	PMC26	P26CFG	○	○
端口 3	0	PM30	P30	PSET30	PCLR30	PU30	PD30	POM30	PMC30	P30CFG	○	○
	1	PM31	P31	PSET31	PCLR31	PU31	PD31	POM31	PMC31	P31CFG	○	○
	2	PM32	P32	PSET32	PCLR32	PU32	PD32	POM32	PMC32	P32CFG	○	○
	3	PM33	P33	PSET33	PCLR33	PU33	PD33	POM33	PMC33	P33CFG	○	—
	4	PM34	P34	PSET34	PCLR34	PU34	PD34	POM34	PMC34	P34CFG	○	—
	5	PM35	P35	PSET35	PCLR35	PU35	PD35	POM35	PMC35	P35CFG	○	○
	6	PM36	P36	PSET36	PCLR36	PU36	PD36	POM36	PMC36	P36CFG	○	○
	7	PM37	P37	PSET37	PCLR37	PU37	PD37	POM37	PMC37	P37CFG	○	○

表2-2: 管脚功能数字映射

功能名称	输入/输出	PxxCFG	PMCxx	PMxx	POMxx	Pxx	备注
模拟功能	输入/输出	6'h00	1	x	x	x	
GPIO	输入	6'h00	0	1	0	x	
	输出		0	0	0	x	
	开漏		0	0	1	x	
INTP0	输入	6'h02	0	1	0	x	
INTP1	输入	6'h03	0	1	0	x	
INTP2	输入	6'h04	0	1	0	x	
INTP3	输入	6'h05	0	1	0	x	
TI00	输入	6'h06	0	1	0	x	
TI01	输入	6'h07	0	1	0	x	
TI02	输入	6'h08	0	1	0	x	
TI03	输入	6'h09	0	1	0	x	
TI10	输入	6'h0a	0	1	0	x	
TI11	输入	6'h0b	0	1	0	x	
TI12	输入	6'h0c	0	1	0	x	
TI13	输入	6'h0d	0	1	0	x	
TO00	输出	6'h0e	0	0	0	x	
TO01	输出	6'h0f	0	0	0	x	
TO02	输出	6'h10	0	0	0	x	
TO03	输出	6'h11	0	0	0	x	
TO10	输出	6'h12	0	0	0	x	
TO11	输出	6'h13	0	0	0	x	
TO12	输出	6'h14	0	0	0	x	
TO13	输出	6'h15	0	0	0	x	
SCLA0	输入/输出	6'h16	0	0	1	x	
SDAA0	输入/输出	6'h17	0	0	1	x	
CLKBUZ0	输出	6'h18	0	0	0	x	
CLKBUZ1	输出	6'h19	0	0	0	x	
RTC1HZ	输出	6'h1a	0	0	0	x	
保留		6'h1b	0	x	x	x	禁止访问
SPI_SSI	输入	6'h1c	0	1	0	x	
SPI_MOSI	输出/输入	6'h1d	0	0/1	0	x	
SPI_MISO	输入/输出	6'h1e	0	1/0	0	x	
SPI_CLKOI	输出/输入	6'h1f	0	0/1	0	x	
保留		6'h20	0	x	x	x	禁止访问
保留		6'h21	0	x	x	x	禁止访问
保留		6'h22	0	x	x	x	禁止访问
保留		6'h23	0	x	x	x	禁止访问

保留		6'h24	0	x	x	x	禁止访问
保留		6'h25	0	x	x	x	禁止访问
SAU0_SS	输入	6'h26	0	1	0	x	
SAU1_SS	输入	6'h27	0	1	0	x	
SCLKOI00	输出/输入	6'h28	0	0	0	x	
SCLKOI01	输出/输入	6'h29	0	0	0	x	
SCLKOI10	输出/输入	6'h2a	0	0	0	x	
SCLKOI11	输出/输入	6'h2b	0	0	0	x	
SDI00/RxD0	输入	6'h2c	0	1	0	x	
SDI01	输入	6'h2d	0	1	0	x	
SDI10/RxD1	输入	6'h2e	0	1	0	x	
SDI11	输入	6'h2f	0	1	0	x	
SDO00/TxD0	输出	6'h30	0	0	0	x	
SDO01	输出	6'h31	0	0	0	x	
SDO10/TxD1	输出	6'h32	0	0	0	x	
SDO11	输出	6'h33	0	0	0	x	
保留	--	6'h34	0	0	0	x	禁止访问
保留	--	6'h35	0	0	0	x	禁止访问
保留	--	6'h36	0	0	0	x	禁止访问
保留	--	6'h37	0	0	0	x	禁止访问
EPWMO00	输出	6'h38	0	0	0	x	
EPWMO01	输出	6'h39	0	0	0	x	
EPWMO02	输出	6'h3a	0	0	0	x	
EPWMO03	输出	6'h3b	0	0	0	x	
EPWMO04	输出	6'h3c	0	0	0	x	
EPWMO05	输出	6'h3d	0	0	0	x	
EPWMO06	输出	6'h3e	0	0	0	x	
EPWMO07	输出	6'h3f	0	0	0	x	

注：由于本产品支持所有数字功能任意GPIO映射，故上表中所对应的功能和PxxCFG的值需要一一对应。

#### 配置说明：

- (1) 使用端口的复用功能时，端口必须配置成数字模式（PMCxx=0）。
- (2) 使用端口的复用功能时，端口必须配置成输出模式(推挽或开漏)（PMxx=0）。
- (3) 使用 P01, P02 端口的 GPIO 功能或者复用功能时，确认其 X1 震荡模式和外部时钟输入模式没有开启。参照 4.3 章节。
- (4) 使用 P12, P13 端口的 GPIO 功能或者复用功能时，确认其 XT1 震荡模式和外部时钟输入模式没有开启。参照 4.3 章节。

### 2.3.1 端口模式寄存器 (PMxx)

当端口作为数字通道使用时，这是以位为单位设定其输入/输出的寄存器。在产生复位信号后，所有端口默认为输入状态。当将端口引脚用作复用功能的引脚时，必须参照复用功能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，偏址见下表。

表2-3: 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	PM02	PM01	PM00	0x020	FFH	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	0x021	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	0x022	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	0x023	FFH	R/W

PMmn	Pmn引脚的输入/输出模式的选择 (m=0~3, n=0~7)
0	输出模式 (用作输出端口 (输出缓冲器ON))
1	输入模式 (用作输入端口 (输出缓冲器OFF))

## 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。在输入模式中读此寄存器可以得到引脚电平，而在输出模式中读可以得到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，偏址见下表。

表2-4：端口寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
P0	0	0	0	0	0	P02	P01	P00	0x000	00H	R/W
P1	0	0	0	0	P13	P12	P11	P10	0x001	00H	R/W
P2	0	P26	P25	P24	P23	P22	P21	P20	0x002	00H	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	0x003	00H	R/W

Pmn	m=0~3, n=0~7	
	输出数据的控制（输出模式）	输入数据的读取（输入模式）
0	输出“0”。	输入低电平。
1	输出“1”。	输入高电平。

### 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，偏址见下表。

表2-5：端口置位控制寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PSET0	0	0	0	0	0	PSET02	PSET01	PSET00	0x070	00H	W
PSET1	0	0	0	0	PSET13	PSET12	PSET11	PSET10	0x071	00H	W
PSET2	0	PSET26	PSET25	PSET24	PSET23	PSET22	PSET21	PSET20	0x072	00H	W
PSET3	PSET37	PSET36	PSET35	PSET34	PSET33	PSET32	PSET31	PSET30	0x073	00H	W

PSETmn	Pmn引脚的置位控制 (m=0~3, n=0~7)
0	无操作
1	对应的Pmn置1



### 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，偏址见下表。

表2-6：端口清零控制寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PCLR0	0	0	0	0	0	PCLR02	PCLR01	PCLR00	0x080	00H	W
PCLR1	0	0	0	0	PCLR13	PCLR12	PCLR11	PCLR10	0x081	00H	W
PCLR2	0	PCLR26	PCLR25	PCLR24	PCLR23	PCLR22	PCLR21	PCLR20	0x082	00H	W
PCLR3	PCLR37	PCLR36	PCLR35	PCLR34	PCLR33	PCLR32	PCLR31	PCLR30	0x083	00H	W

PCLRmn	Pmn引脚的清零控制 (m=0~3, n=0~7)
0	无操作
1	对应的Pmn清零

## 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。对于配置为数字功能的引脚，可以通过上拉电阻选择寄存器以位为单位使用内部上拉电阻；设定为模拟功能的引脚不连接内部上拉电阻。

在产生复位信号后，P00，P20，P21，P36，P37这四个端口的上拉功能默认打开（PU00，PU20，PU21，PU36，P37复位值为“1”），其他端口的上拉功能默认不打开。

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

表2-7：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	PU02	PU01	PU00	0x030	01H	R/W
PU1	0	0	0	0	0	0	PU11	PU10	0x031	00H	R/W
PU2	0	PU26	PU25	PU24	PU23	PU22	PU21	PU20	0x032	03H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	0x033	C0H	R/W

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

注意：本产品端口P12，P13不支持上拉功能，使用时必须外接上拉电阻。

## 2.3.6 下拉电阻选择寄存器 (PDxx)

内部下拉电阻的选择寄存器。对于配置为数字功能的引脚，可以通过下拉电阻选择寄存器以位为单位使用内部下拉电阻；设定为模拟功能的引脚不连接内部下拉电阻。

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；PD寄存器的基址为0x40040000，偏址见下表。

表2-81：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
PD0	0	0	0	0	0	PD02	PD01	0	0x040	00H	R/W
PD1	0	0	0	0	0	0	PD11	PD10	0x041	00H	R/W
PD2	0	PD26	PD25	PD24	PD23	PD22	PD21	PD20	0x042	00H	R/W
PD3	PD37	PD36	PD35	PD34	PD33	PD32	PD31	PD30	0x043	00H	R/W

PDmn	Pmn引脚的内部下拉电阻的选择 (m=0~3, n=0~7)
0	不连接内部下拉电阻。
1	连接内部下拉电阻。

注意：本产品端口P00，P12，P13不支持下拉功能，使用时必须外接下拉电阻。

### 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。在设备间进行通信时，引脚可选择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，偏址见下表。

表2-9：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	POM02	POM01	POM00	0x050	00H	R/W
POM1	0	0	0	0	POM13	POM12	POM11	POM10	0x051	00H	R/W
POM2	0	POM26	POM25	POM24	POM23	POM22	POM21	POM20	0x052	00H	R/W
POM3	POM37	POM36	POM35	POM34	POM33	POM32	POM31	POM30	0x053	00H	R/W

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

### 2.3.8 端口模式控制寄存器（PMCxx）

PMC寄存器以位为单位设定端口作为数字输入/输出使用或者作为模拟通道使用。

在产生复位信号后，P00，P01，P02，P20，P21，P36，P37默认作为数字通道使用（PMC00，PMC01，PMC02，PMC20，PMC21，PMC36，PMC37复位值为“0”），其他端口默认作为模拟通道使用。

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

表2-10：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	PMC02	PMC01	PMC00	0x060	F8H	R/W
PMC1	1	1	1	1	PMC13	PMC12	PMC11	PMC10	0x061	FFH	R/W
PMC2	1	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	0x062	FCH	R/W
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	0x063	3FH	R/W

PMCmn	Pmn引脚的数字输入/输出或者模拟输入的选择（m=0~3，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

### 2.3.9 端口回读寄存器 (PREADxx)

这是一个只读寄存器，在端口作为数字GPIO时，可以通过读此寄存器得到对应端口电平。

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

表2-11：端口回读寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位值	R/W
PREAD0	--	--	--	--	--	PREAD02	PREAD01	PREAD00	0x90	xxxxH	R
PREAD1	--	--	--	--	PREAD13	PREAD12	PREAD11	PREAD10	0x91	xxxxH	R
PREAD2	--	PREAD26	PREAD25	PREAD24	PREAD23	PREAD22	PREAD21	PREAD20	0x92	xxxxH	R
PREAD3	PREAD37	PREAD36	PREAD35	PREAD34	PREAD33	PREAD32	PREAD31	PREAD30	0x93	xxxxH	R

PREADmn	m=0~3, n=0~7	
	数字输出模式/输入模式	
0	端口为低电平	
1	端口为高电平	

备注：PREAD只用于读操作，不支持写操作。其复位后的读出值取决于对应端口电平。

### 2.3.10 端口复用功能配置寄存器 (PxxCFG)

端口复用功能配置寄存器可实现将一部分外围模块的输出功能映射到任意端口。端口复用功能配置寄存器的复位值为“00H”，此时端口为默认为GPIO功能。

寄存器地址=基址+偏址；PxxCFG寄存器的基址为0x40040800，偏址见下表。

基址地址：0x40040800

表2-12：端口复用功能配置寄存器的偏址

符号	7	6	5	4	3	2	1	0
P0xCFG	--	--	--	--	--	P02CFG	P01CFG	P00CFG
偏址	--	--	--	--	--	0x04	0x02	0x00
P1xCFG	--	--	--	--	P13CFG	P12CFG	P11CFG	P10CFG
偏址	--	--	--	--	0x16	0x14	0x12	0x10
P2xCFG	--	P26CFG	P25CFG	P24CFG	P23CFG	P22CFG	P21CFG	P20CFG
偏址	--	0x2c	0x2a	0x28	0x26	0x24	0x22	0x20
P3xCFG	P37CFG	P36CFG	P35CFG	P34CFG	P33CFG	P32CFG	P31CFG	P30CFG
偏址	0x3e	0x3c	0x3a	0x38	0x36	0x34	0x32	0x30

备注：具体功能映射参照“表 2-2：管脚功能数字映射”。

### 2.3.11 特殊功能端口RESINB说明

本产品上电默认 RESINB 有效，如果需要把该口做 GPIO 使用，那么需要通过寄存器关掉复位功能，寄存器描述如下

基地址：0x40020400 偏址：0x0B

符号	7	6	5	4	3	2	1	0	复位值	R/W
RSTM	0	0	0	0	0	0	0	RSTM	00H	R/W

RSTM	RESINB引脚的外部复位功能屏蔽
0	RESINB外部复位引脚
1	RESINB做GPIO引脚



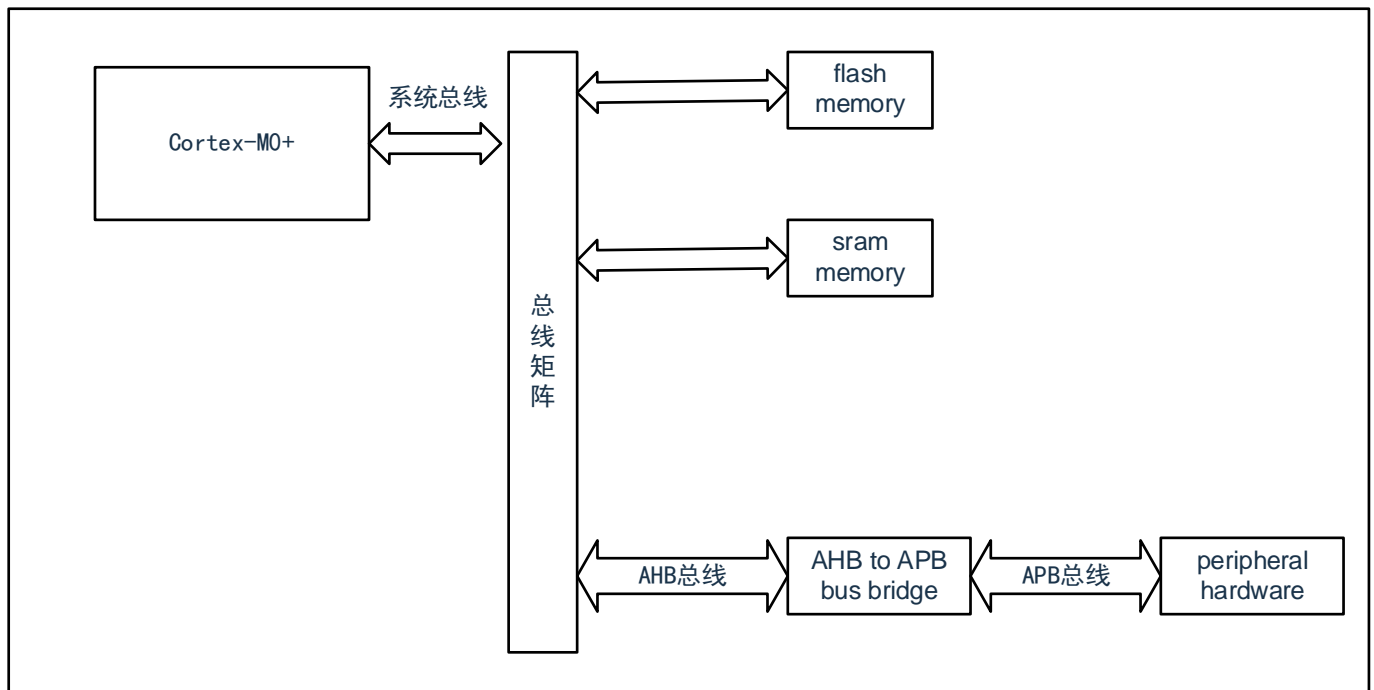
## 第3章 系统结构

### 3.1 概述

本产品系统由以下部分组成：

- 1个AHB总线Master：
  - Cortex-M0+
- 3个AHB总线Slaves：
  - FLASH存储器
  - SRAM存储器
  - AHB to APB Bridge，包含所有APB接口外设

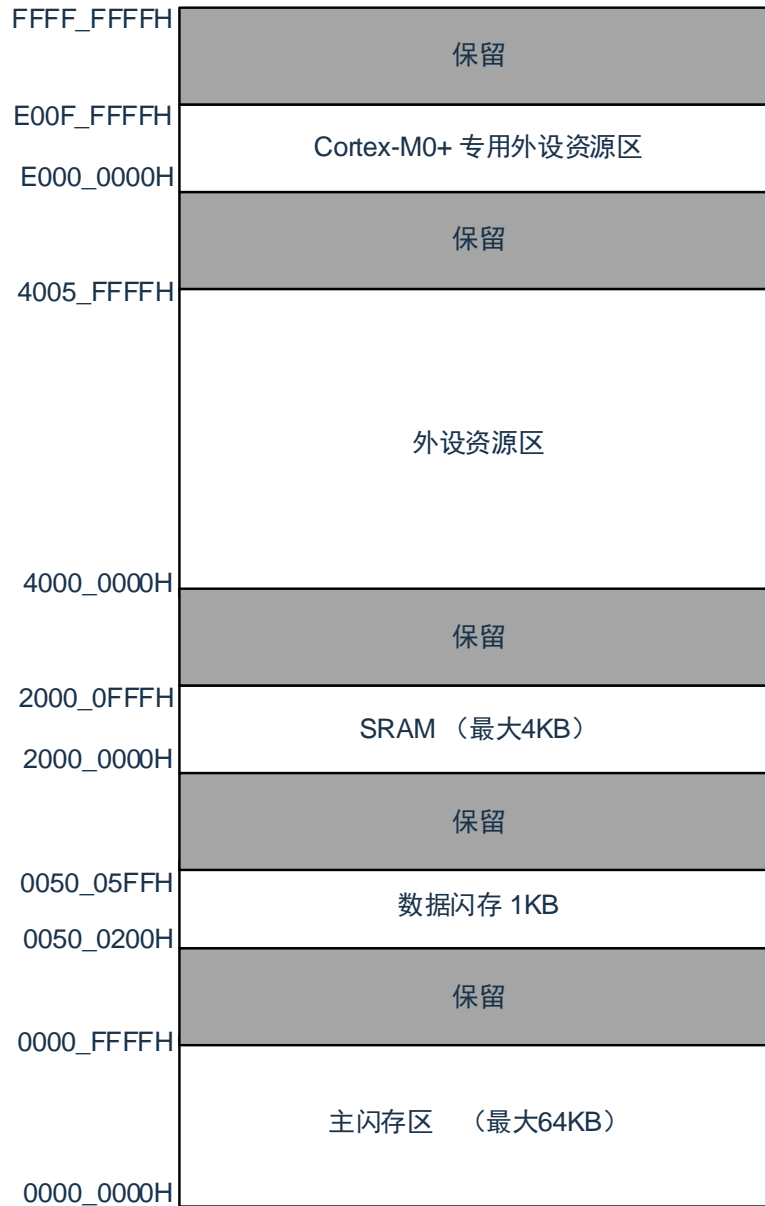
图3-1：系统结构示意图



- 1) 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。
- 2) 总线矩阵：总线矩阵协调内核系统总线其他总线的访问。
- 3) AHB to APB Bridge：AHB to APB Bridge 在AHB和APB总线间提供同步连接。有关连接到每个桥的不同外设的地址映射请参考表3-1。

### 3.2 系统地址划分

图3-2：地址区域划分示意图



## 外设地址分配

表3-1: 外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_4FFF	保留	
0x4000_5000 - 0x4000_5FFF	保留	
0x4000_6000 - 0x4000_6FFF	中断控制	
0x4000_7000 - 0x4001_8FFF	保留	
0x4001_9000 - 0x4001_9FFF	保留	
0x4001_A000 - 0x4001_FFFF	保留	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1000 - 0x4002_1001	看门狗定时器	
0x4002_1002 - 0x4002_1800	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见第 21 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 0	
0x4004_2000 - 0x4004_21FF	定时器阵列 1	
0x4004_2200 - 0x4004_23FF	保留	
0x4004_2400 - 0x4004_27FF	SPI	
0x4004_2800 - 0x4004_31FF	保留	
0x4004_3200 - 0x4004_32FF	通用 CRC	
0x4004_3300 - 0x4004_33FF	通用 CRC	
0x4004_3400 - 0x4004_37FF	联动控制器	
0x4004_3C00 - 0x4004_3FFF	保留	
0x4004_4000 - 0x4004_43FF	保留	
0x4004_4400 - 0x4004_47FF	EPWM	
0x4004_4800 - 0x4004_4EFF	保留	
0x4004_4F00 - 0x4004_4FFF	实时时钟	
0x4004_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_5AFF	保留	
0x4004_5B00 - 0x4004_5BFF	外部中断控制	
0x4008_0000 - 0x4008_01FF	保留	
0x4008_0200 - 0xDFFF_FFFF	保留	

## 第4章 时钟发生电路

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

### 4.1 时钟发生电路的功能

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

#### (1) 主系统时钟

##### ① X1振荡电路

能通过给X1引脚和X2引脚连接谐振器使 $F_X=4\sim 16\text{MHz}$ 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。

##### ② 高速内部振荡器（高速OCO）

能通过选项字节（000C2H）从 $F_{IH}=64\text{MHz}$ 、 $48\text{MHz}$ 、 $32\text{MHz}$ 、 $24\text{MHz}$ 、 $16\text{MHz}$ 、 $12\text{MHz}$ 、 $8\text{MHz}$ 、 $6\text{MHz}$ 、 $4\text{MHz}$ 、 $3\text{MHz}$ 和 $2\text{MHz}$ 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 $F_{IH}$ 开始运行。能通过进入深度睡眠模式或者设置HIOSSTOP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“表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引脚提供外部主系统时钟（ $F_{EX}=4\sim 16\text{MHz}$ ），并且能通过进入深度睡眠模式或者设置MSTOP位将外部主系统时钟的输入置为无效。

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

#### (2) 副系统时钟

##### XT1振荡电路

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

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

#### (3) 低速内部振荡器时钟（低速OCO）

能使 $F_{IL}=15\text{KHz}$ 的时钟振荡。

可以将低速内部振荡器时钟用作系统时钟。

当选项字节（000C0H）的bit4（WDTON）或者副系统时钟提供模式控制寄存器（OSMC）的bit4（WUTMMCK0）为“1”时，或者副系统时钟选择寄存器（SUBCKSEL）的bit0（SELLOSC）为“1”时，低速内部振荡器振荡。

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

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

备注： $F_X$ ：X1 时钟振荡频率

$F_{HOCO}$ ：高速内部振荡器的时钟频率

$F_{IH}$ ：高速内部振荡器的时钟频率

$F_{EX}$ ：外部主系统时钟频率

$F_{XT}$ ：XT1 时钟振荡频率

$F_{EXS}$ ：外部副系统时钟频率

$F_{IL}$ ：低速内部振荡器的时钟频率

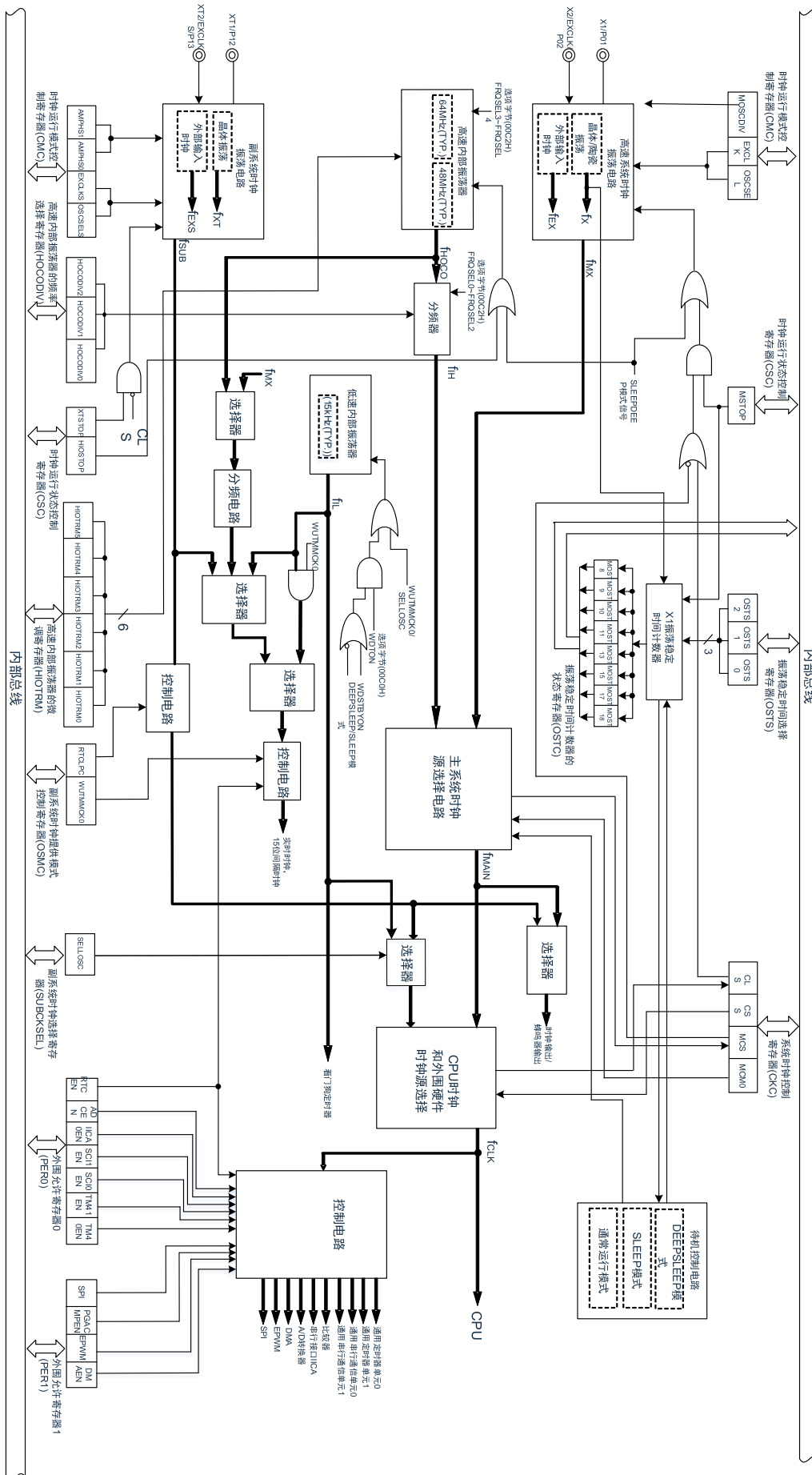
## 4.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表4-1：时钟发生电路的结构

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

图4-1：时钟发生电路的框图



备注:  $F_X$ : X1时钟振荡频率

$F_{HOCO}$ : 高速内部振荡器的时钟频率

$F_{IH}$ : 高速内部振荡器的时钟频率

$F_{EX}$ : 外部主系统时钟频率

$F_{MX}$ : 高速系统时钟频率

$F_{MAIN}$ : 主系统时钟频率

$F_{XT}$ : XT1时钟振荡频率

$F_{EXS}$ : 外部副系统时钟频率

$F_{SUB}$ : 副系统时钟频率

$F_{CLK}$ : CPU/外围硬件的时钟频率

$F_{IL}$ : 低速内部振荡器的时钟频率



## 4.3 控制时钟发生电路的寄存器

通过以下寄存器控制时钟发生电路。

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

注意：分配的寄存器和位因产品而不同。必须给未分配的位设置初始值。

### 4.3.1 时钟运行模式控制寄存器 (CMC)

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

在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。能通过 8 位存储器操作指令读此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表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 <sup>注1</sup>					
符号	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS <sup>注2</sup>	OSCSELS <sup>注2</sup>	0	AMPHS1 <sup>注2</sup>	AMPHS0 <sup>注2</sup>	0

EXCLK	OSCSEL	高速系统时钟引脚的运行模式	X1引脚	X2/EXCLK引脚
0	0	端口模式	输入/输出端口	
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/输出端口	
1	1	外部时钟输入模式	输入/输出端口	外部时钟输入

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

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

注1: bit3是只写位。

注2: EXCLKS位, OSCSELS位, AMPHS1位和AMPHS0位只在上电复位时被初始化, 而在其他复位时保持不变。

注3: 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”。

注4: 在解除复位后并且在通过设置时钟运行状态控制寄存器 (CSC) 开始X1或者XT1振荡前, 必须设置CMC寄存器。

注5: 必须在解除复位后并且在选择F<sub>IH</sub>作为F<sub>CLK</sub>的状态 (将F<sub>CLK</sub>切换为F<sub>MX</sub>或者F<sub>SUB</sub>前的状态) 下设置AMPHS1位和AMPHS0位。

注6: 必须通过软件对F<sub>XT</sub>的振荡稳定时间进行计数。

注7: 系统时钟的频率上限为64MHz, 但是X1振荡电路的频率上限为16MHz。

注8: F<sub>X</sub>: X1时钟振荡频率。

### 4.3.2 系统时钟控制寄存器（CKC）

这是选择 CPU/外围硬件时钟和主系统时钟的寄存器。

通过 8 位存储器操作指令设置 CKC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表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>注1</sup>					
符号	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0

CLS	CPU/外围硬件时钟（F <sub>CLK</sub> ）的状态
0	主系统时钟（F <sub>MAIN</sub> ）
1	副系统时钟（F <sub>SUB</sub> ）

CSS <sup>注1</sup>	CPU/外围硬件时钟（F <sub>CLK</sub> ）的选择
0	主系统时钟（F <sub>MAIN</sub> ）
1	副系统时钟（F <sub>SUB</sub> ）

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 <sup>注1</sup>	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1：bit7和bit5是只读位，必须将bit0~3置“0”。

注意：禁止在将CSS位置“1”的状态下更改MCM0位的值。

备注：

1. 给CPU和外围硬件提供CSS位设置的时钟。如果更改CPU时钟，就同时更改外围硬件的时钟（实时时钟、15位间隔定时器、时钟输出/蜂鸣器输出和看门狗定时器除外）。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。
2. 如果将副系统时钟用作外围硬件时钟，就无法保证A/D转换器和IICA的运行。有关外围硬件的运行特性，请参照各外围硬件的章节和数据手册。
3. F<sub>HOCO</sub>：高速内部振荡器的时钟频率  
 F<sub>IH</sub>：高速内部振荡器的时钟频率  
 F<sub>MX</sub>：高速系统时钟频率  
 F<sub>MAIN</sub>：主系统时钟频率  
 F<sub>SUB</sub>：副系统时钟频率

### 4.3.3 时钟运行状态控制寄存器（CSC）

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

在产生复位信号后，此寄存器的值变为“C0H”。

表4-4：时钟运行状态控制寄存器（CSC）的格式

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

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

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

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

注意：

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

备注：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4-5。

表4-5：时钟停止方法

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

### 4.3.4 振荡稳定时间计数器的状态寄存器（OSTC）

这是表示 X1 时钟的振荡稳定时间计数器计数状态的寄存器。能在以下情况下确认 X1 时钟的振荡稳定时间：  
当 CPU 时钟为高速内部振荡器时钟或者副系统时钟并且开始 X1 时钟的振荡时。

当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到深度睡眠模式后解除睡眠模式时。

能通过 8 位存储器操作指令读 OSTC 寄存器。

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

备注：在以下情况下，振荡稳定时间计数器开始计数：

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

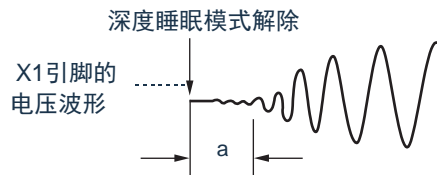
表4-6: 振荡稳定时间计数器的状态寄存器 (OSTC) 的格式

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> =4MHz	F <sub>X</sub> =16MHz
0	0	0	0	0	0	0	0	小于2 <sup>8</sup> /F <sub>X</sub>	小于64us
1	0	0	0	0	0	0	0	至少2 <sup>8</sup> /F <sub>X</sub>	至少64us
1	1	0	0	0	0	0	0	至少2 <sup>9</sup> /F <sub>X</sub>	至少128us
1	1	1	0	0	0	0	0	至少2 <sup>10</sup> /F <sub>X</sub>	至少256us
1	1	1	1	0	0	0	0	至少2 <sup>11</sup> /F <sub>X</sub>	至少512us
1	1	1	1	1	0	0	0	至少2 <sup>13</sup> /F <sub>X</sub>	至少2.048ms
1	1	1	1	1	1	0	0	至少2 <sup>15</sup> /F <sub>X</sub>	至少8.192ms
1	1	1	1	1	1	1	0	至少2 <sup>17</sup> /F <sub>X</sub>	至少32.768ms
1	1	1	1	1	1	1	1	至少2 <sup>18</sup> /F <sub>X</sub>	至少65.536ms

备注:

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



4. 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”。

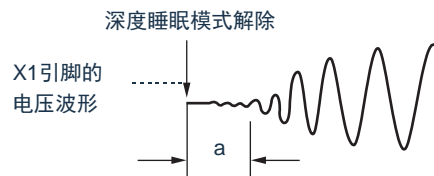
表4-7: 振荡稳定时间选择寄存器 (OSTS) 的格式

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> =4MHz	F <sub>X</sub> =16MHz
0	0	0	2 <sup>8</sup> /F <sub>X</sub>	64us
0	0	1	2 <sup>9</sup> /F <sub>X</sub>	128us
0	1	0	2 <sup>10</sup> /F <sub>X</sub>	256us
0	1	1	2 <sup>11</sup> /F <sub>X</sub>	512us
1	0	0	2 <sup>13</sup> /F <sub>X</sub>	2.048ms
1	0	1	2 <sup>15</sup> /F <sub>X</sub>	8.192ms
1	1	0	2 <sup>17</sup> /F <sub>X</sub>	32.768ms
1	1	1	2 <sup>18</sup> /F <sub>X</sub>	65.536ms

备注:

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



4. F<sub>X</sub>: X1时钟振荡频率。

### 4.3.6 外围允许寄存器0、1（PER0、PER1）

这是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。当使用由这些寄存器控制的以下外围功能时，必须在进行外围功能的初始设置前将对应位置“1”。

实时时钟、15位间隔定时器

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

通用定时器单元 1

通用定时器单元 0

EPWM

SPI

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”。

表4-8：外围允许寄存器0（PER0）的格式(1/3)

地址：	40020420H	复位后：00H	R/W					
符号	7	6	5	4	3	2	1	0
PER0	RTCEN <sup>注</sup>	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

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

注：RTCEN位只在上电复位时被初始化，而在其他复位时保持不变。



表4-8：外围允许寄存器0（PER0）的格式(2/3)

地址：	40020420H	复位后：00H	R/W					
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

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。

IICAEN	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。

表4-8：外围允许寄存器0（PER0）的格式(3/3)

地址:	40020420H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0	
PER0	RTCEN	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN	

TM41EN	提供通用定时器单元1的输入时钟的控制
0	停止提供输入时钟。 • 不能写通用定时器单元1使用的SFR。 • 通用定时器单元1处于复位状态。
1	提供输入时钟。 • 能读写通用定时器单元1使用的SFR。

TM40EN	提供通用定时器单元0的输入时钟的控制
0	停止提供输入时钟。 • 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。

表4-9: 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	SPIEN	0	-	0	0	EPWMEN	0	0

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

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

### 4.3.7 副系统时钟提供模式控制寄存器（OSMC）

OSMC 寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

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

另外，能通过 OSMC 寄存器选择实时时钟和 15 位间隔定时器的运行时钟。

通过 8 位存储器操作指令设置 OSMC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表4-10：副系统时钟提供模式控制寄存器（OSMC）的格式

地址：	40020423H	复位后：00H	R/W					
符号	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	深度睡眠模式和CPU以副系统时钟运行的睡眠模式中的设置
0	允许给外围功能提供副系统时钟 (有关允许运行的外围功能，请参照表18-1~表18-3)。
1	停止给实时时钟和15位间隔定时器以外的外围功能提供副系统时钟。

WUTMMCK0	实时时钟、15位间隔定时器的运行时钟的选择
0	副系统时钟为实时时钟和15位间隔定时器的运行时钟。
1	低速内部振荡器时钟为实时时钟和15位间隔定时器的运行时钟。

### 4.3.8 高速内部振荡器的频率选择寄存器（HOCODIV）

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

通过8位存储器操作指令设置HOCODIV寄存器。

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

表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	
			FRQSEL4,3=00	FRQSEL4,3=01
0	0	0	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> =24MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =32MHz F <sub>HOCO</sub> =64MHz
0	1	0	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> =6MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =8MHz F <sub>HOCO</sub> =64MHz
1	0	0	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> =2MHz F <sub>HOCO</sub> =64MHz
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<sub>IH</sub>）作为CPU/外围硬件时钟（F<sub>CLK</sub>）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：
  - 1) 以更改前的频率，最多进行3个时钟的运行。
  - 2) 以更改后的频率，最多等待3个CPU/外围硬件的时钟。

### 4.3.9 高速内部振荡器的微调寄存器（HIOTRM）

这是校正高速内部振荡器精度的寄存器。能使用高精度的外部时钟输入的定时器等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。

表4-12: 高速内部振荡器的微调寄存器（HIOTRM）的格式

地址:	40021C00H	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：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。

注意：

1. 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。
2. 复位值是发货时的调整值。
3. 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。

### 4.3.10 副系统时钟选择寄存器 (SUBCKSEL)

SUBCKSEL 寄存器是选择副系统时钟  $F_{SUB}$  和低速内部振荡器时钟  $F_{IL}$  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”。

表4-13：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	
SUBCKSEL	0	0	0	0	0	0	0	0	SELLOSC

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

### 4.3.11 供电模式控制保护寄存器（PMUKEY）

PMUKEY 寄存器是供电模式控制 PMUCTL 的保护的寄存器。

通过 16 位存储器操作指令设置 PMUKEY 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

表4-14：供电模式控制保护寄存器（PMUKEY）的格式

地址：	40020408H	复位后：	0000H	R/W <sup>注1</sup>						0
符号	15									
PMUKEY										

PMUKEY	供电模式控制保护寄存器的选择
先写192AH 再写3E4FH	<ul style="list-style-type: none"> <li>解除 PMUCTL写入保护。 通过先后写入192AH、3E4FH到PMUKEY使能PMUCTL的PVDNEN位的写入控制</li> </ul>
其他	<ul style="list-style-type: none"> <li>PMUCTL写入设置无效。</li> </ul>

### 4.3.12 供电模式控制寄存器（PMUCTL）

PMUCTL 寄存器是控制使能供电控制模式的寄存器。

通过 8 位存储器操作指令设置 PMUCTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”并且写入保护打开，通过 PMUKEY 解除写入控制。

表4-15：副系统时钟选择寄存器（PMUCTL）的格式

地址：	4002040AH	复位后：	00H	R/W							
符号	7	6	5	4	3	2	1	0			
PMUCTL											PVDNEN

PVDNEN	供电模式控制寄存器的选择
0	<ul style="list-style-type: none"> <li>部分掉电模式禁止。</li> </ul>
1	<ul style="list-style-type: none"> <li>部分掉电模式使能。</li> </ul>

注意：通过PMUKEY解除PMUCTL的写入保护。



## 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（4~16MHz）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，必须设置为端口模式（EXCLK、OSCSEL=0、0）。

X1 振荡电路的外接电路例子如图 4-2 所示。

图4-2：X1振荡电路的外接电路例子



注意事项如下页所示。

### 4.4.2 XT1振荡电路

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

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

晶体振荡: EXCLKS、OSCSELS=0、1

外部时钟输入: EXCLKS、OSCSELS=1、1

当不使用XT1振荡电路时, 必须设置为端口模式 (EXCLKS、OSCSELS=0、0)。XT1振荡电路的外接电路例子如图4-3所示。

图4-3: XT1振荡电路的外接电路例子



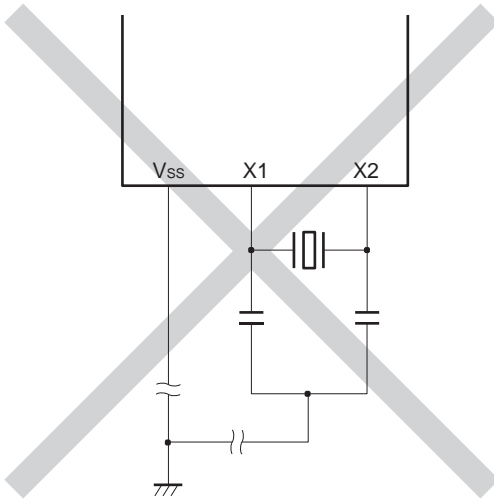
注意: 当使用X1振荡电路和XT1振荡电路时, 为了避免布线电容等的影响, 必须通过以下方法对图4-2(a)和图4-2(b)中的虚线部分进行布线:

- 1) 必须尽量缩短布线。
- 2) 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。
- 3) 必须始终保持振荡电路的电容器接地点和V<sub>SS</sub>同电位, 而且不能给大电流流过的接地图形接地。
- 4) 不能从振荡电路取出信号。

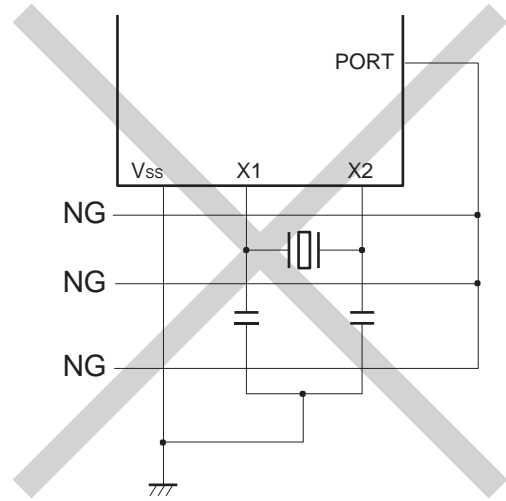
不正确的谐振器连接例子如图4-4所示。

图4-4：不正确的谐振器连接例子(1/2)

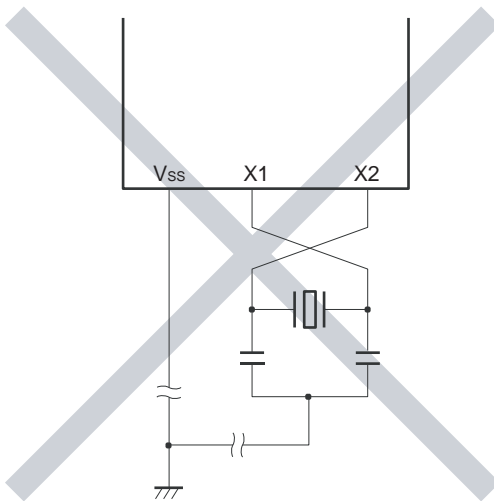
(a)连接电路的布线太长



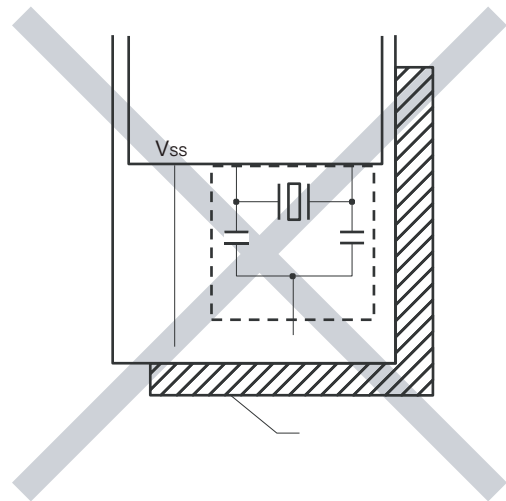
(b)信号线交叉



(c)X1和X2的信号线交叉布线



(d)X1和X2的布线下有电源或者接地图形

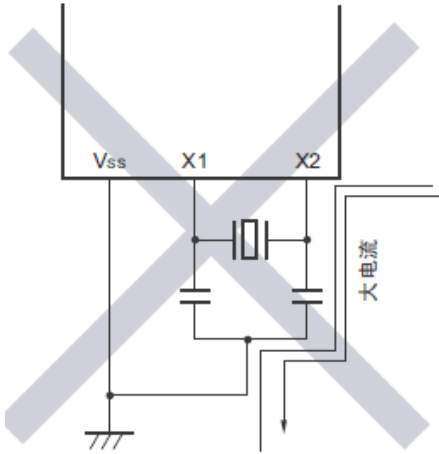


注意：

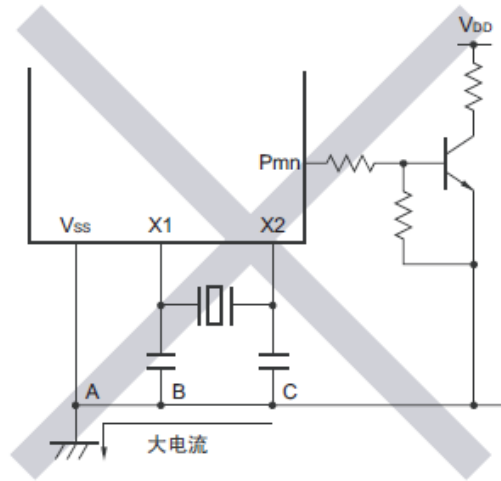
1. 在多层板或者双面板中，不能在X1引脚、X2引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。
2. 在使用副系统时钟的情况下，请阅读时分别用XT1和XT2代替X1和X2，并且在XT2侧插入串联电阻。

图4-4：不正确的谐振器连接例子(2/2)

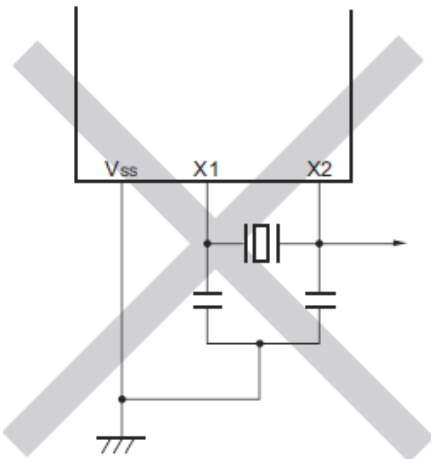
(e)有变化的大电流接近信号线



(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：

1. 当X2和XT1并行布线时，X2的串扰噪声会叠加到XT1而导致误动作。
2. 在使用副系统时钟的情况下，请阅读时分别用XT1和XT2代替X1和X2，并且在XT2侧插入串联电阻。

### 4.4.3 高速内部振荡器

CMS32L032 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 和 2MHz 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。

在解除上电复位后，高速内部振荡器自动开始振荡。

### 4.4.4 低速内部振荡器

CMS32L032 内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器、实时时钟、15 位间隔定时器的时钟，以及 SysTick 定时器的外部参考时钟，也可用作 CPU 时钟和外围模块时钟。

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

当看门狗定时器停止运行并且 WUTMMCK0 位不为“0”时，低速内部振荡器继续振荡。但是，如果看门狗定时器运行而 WUTMMCK0 位、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）。

$F_{\text{MAIN}}$ : 主系统时钟频率

$F_{\text{MX}}$ : 高速系统时钟频率

$F_{\text{X}}$ : X1时钟振荡频率

$F_{\text{EX}}$ : 外部主系统时钟频率

$F_{\text{IH}}$ : 高速内部振荡器的时钟频率

$F_{\text{SUB}}$ : 副系统时钟频率

$F_{\text{XT}}$ : XT1时钟振荡频率

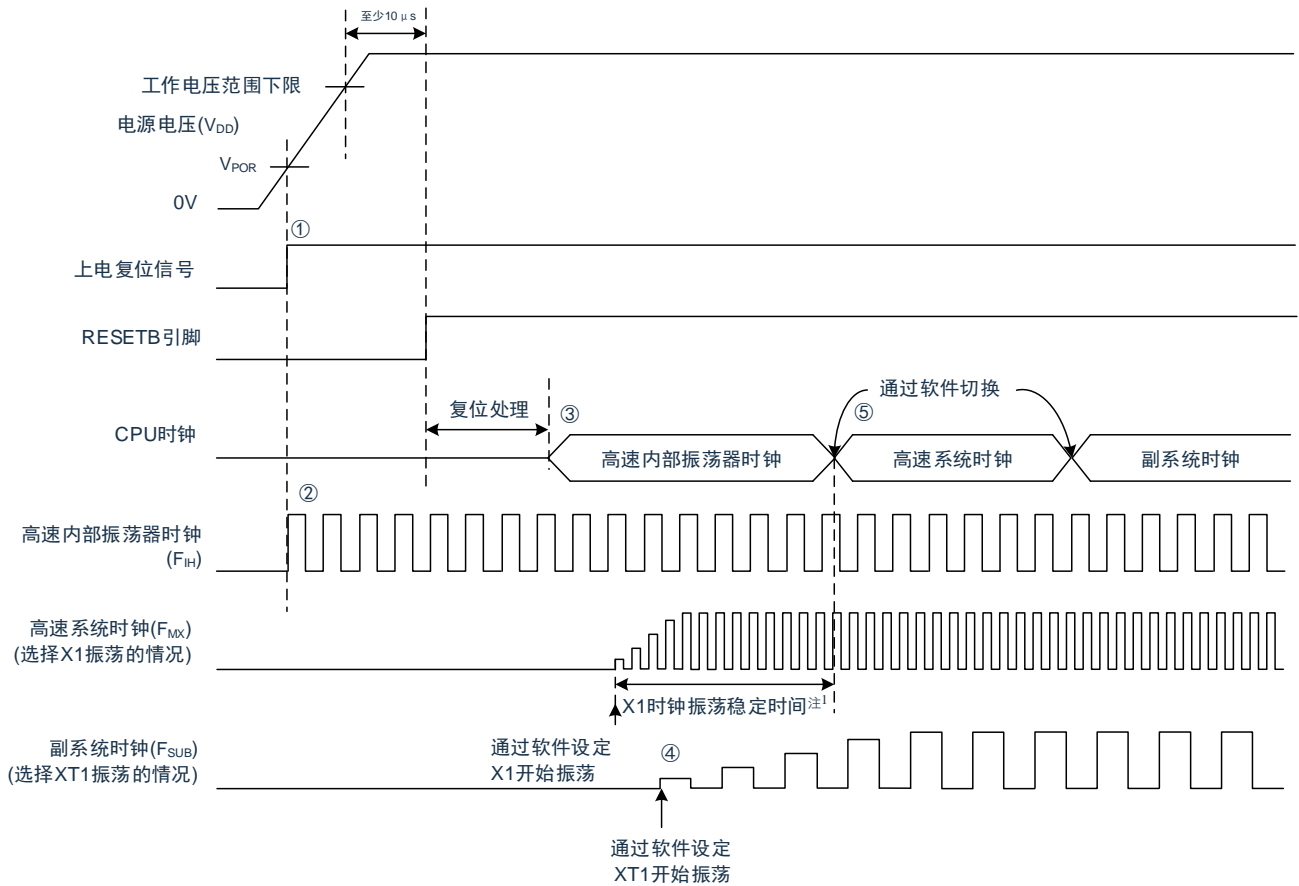
$F_{\text{EXS}}$ : 外部副系统时钟频率

$F_{\text{CLK}}$ : CPU/外围硬件的时钟频率

$F_{\text{IL}}$ : 低速内部振荡器的时钟频率

CMS32L032 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 4-5 所示。

图4-5: 接通电源时的时钟发生电路的运行



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

注1：当解除复位时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认X1时钟的振荡稳定时间。

备注：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和2MHz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	7	6	5	4	3	2	1	0
字节 (000C2H)	1	1	1	FRQSEL4 0	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	
					$F_{HOCO}$	$F_{IH}$
0	1	0	0	0	64MHz	64MHz
0	0	0	0	0	48MHz	48MHz
0	1	0	0	1	64MHz	32MHz
0	0	0	0	1	48MHz	24MHz
0	1	0	1	0	64MHz	16MHz
0	0	0	1	0	48MHz	12MHz
0	1	0	1	1	64MHz	8MHz
0	0	0	1	1	48MHz	6MHz
0	1	1	0	0	64MHz	4MHz
0	0	1	0	0	48MHz	3MHz
0	1	1	0	1	64MHz	2MHz
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	
			FRQSEL4,3=00	FRQSEL4,3=01
0	0	0	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> =24MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =32MHz F <sub>HOCO</sub> =64MHz
0	1	0	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> =6MHz F <sub>HOCO</sub> =48MHz	F <sub>IH</sub> =8MHz F <sub>HOCO</sub> =64MHz
1	0	0	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> =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<sub>CLK</sub>）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<sub>CLK</sub>。

【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”，通过 MOSCDIV 设置 X1 时钟分频选择，使 X1 振荡电路运行。

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL1	EXCLKS0	OSCELS0	MOSCDIV 0	AMPHS10	AMPHS00	0

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

例) 要通过 10MHz 谐振器至少等待 102us 时，必须设置为以下的值。

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

- ③ 将 CSC 寄存器的 MSTOP 位清“0”，使 X1 振荡电路开始振荡。

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

- ④ 通过 OSTC 寄存器等待 X1 振荡电路的振荡稳定。

例) 要通过 10MHz 谐振器至少等待 102us 时，必须等到各位变为以下的值。

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

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

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

### 4.6.3 XT1振荡电路的设置例子

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

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

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

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

- ② 将 CMC 寄存器的 OSCSELS 位置“1”，使 XT1 振荡电路运行。

	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL0	EXCLKS0	OSCSELS1	MOCDIV0	AMPHS1 0/1	AMPHS0 0/1	0

AMPHS0 位和 AMPHS1 位：设置 XT1 振荡电路的振荡模式。

- ③ 将 CSC 寄存器的 XTSTOP 位清“0”，使 XT1 振荡电路开始振荡。

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

- ④ 必须通过软件和定时器功能等，等待副系统时钟所需的振荡稳定时间。

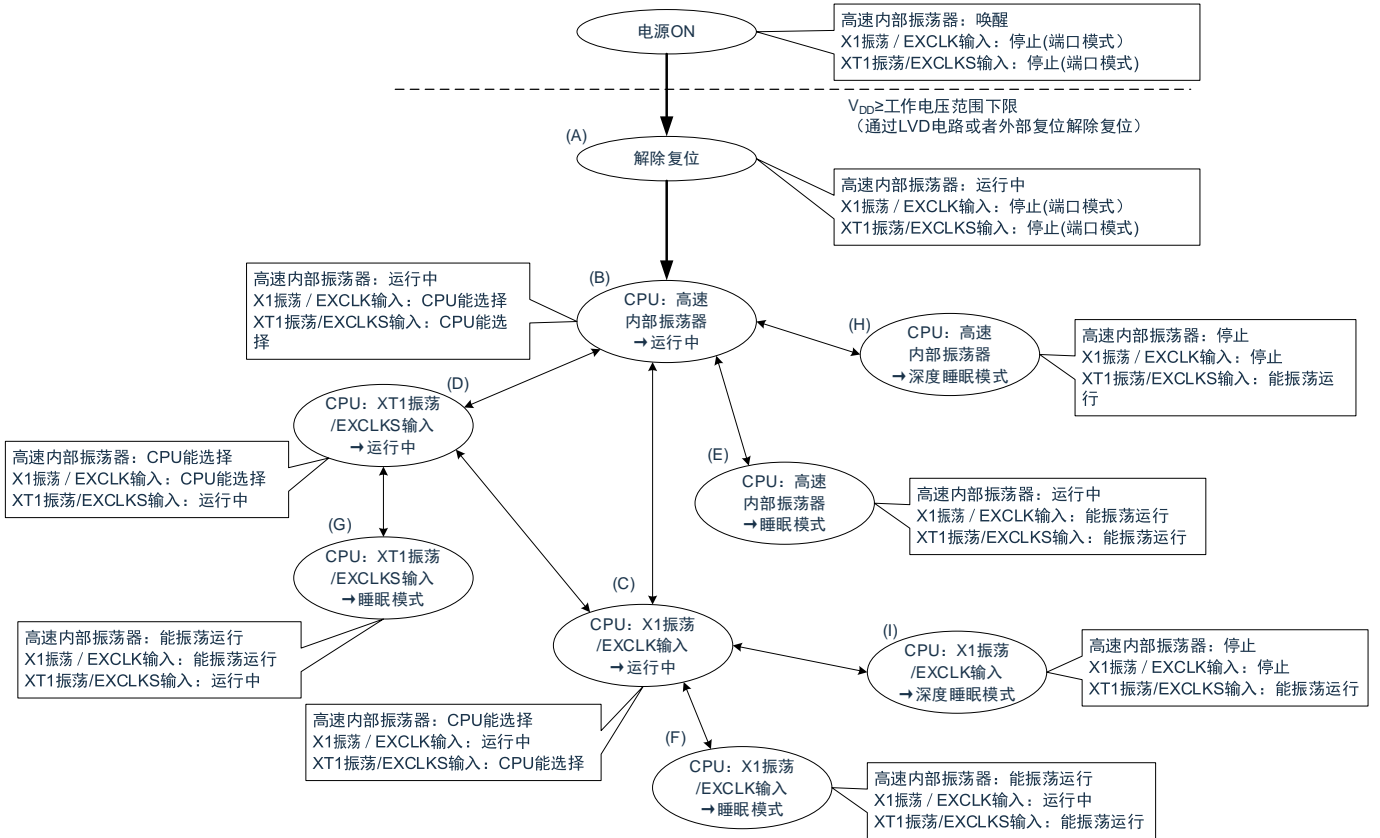
- ⑤ 通过 CKC 寄存器的 CSS 位将 XT1 振荡时钟设置为 CPU/外围硬件时钟。

	7	6	5	4	3	2	1	0
CKC	CLS0	CSS1	MCS0	MCM0	0	0	0	0

### 4.6.4 CPU时钟的状态转移图

本产品的 CPU 时钟状态转移图如图 4-6 所示。

图4-6: 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 4-16 所示。

表 4-16: CPU 时钟的转移和 SFR 寄存器的设置例子(1/4)

(1) 在解除复位(A)后, CPU 转移到高速内部振荡器时钟运行(B)。

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

(2) 在解除复位(A)后, CPU 转移到高速系统时钟运行(C)。（CPU 在解除复位后立即以高速内部振荡器时钟运行(B)）

(SFR寄存器的设置顺序) →

状态转移	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	OSCESEL		MSTOP		MCM0
(A)→(B)→(C) (X1时钟)	0	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

(3) 在解除复位(A)后, CPU 转移到副系统时钟运行(D)。（CPU 在解除复位后立即以高速内部振荡器时钟运行(B)）

(SFR寄存器的设置顺序) →

状态转移	CMC寄存器 <sup>注3</sup>				CSC 寄存器	振荡稳定的 等待	CKC 寄存器
	EXCLKS	OSCELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A)→(B)→(D) (XT1时钟)	0	1	0/1	0/1	0	需要	1
(A)→(B)→(D) (外部副时钟)	1	1	x	x	0	需要	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3: 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 (参照数据手册的电气特性) 后设置时钟。

备注:

1. x: 忽略。
2. 表4-16的(A)~(I)对应图4-6的(A)~(I)。

表4-16: CPU时钟的转移和SFR寄存器的设置例子(2/4)

(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	OSCSSEL		MSTOP		MCM0
(B)→(C) (X1时钟)	0	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。

(5) CPU 从高速内部振荡器时钟运行(B)转移到副系统时钟运行(D)。

(SFR寄存器的设置顺序) →

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

如果已设置就不需要。      在副系统时钟运行中不需要。

(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	1us	0

在高速内部振荡器时钟运行中不需要。

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 (参照数据手册的电气特性) 后设置时钟。

备注:

1. x: 忽略。
2. 表4-16的(A)~(I)对应图4-6的(A)~(I)。
3. 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4-16: CPU时钟的转移和SFR寄存器的设置例子(3/4)

(7) CPU 从高速系统时钟运行(C)转移到副系统时钟运行(D)。

(SFR寄存器的设置顺序) →

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

在副系统时钟运行中不需要。

(8) CPU 从副系统时钟运行(D)转移到高速内部振荡器时钟运行(B)。

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志	CSC寄存器	振荡精度稳定的等待	CKC寄存器
		HIOSTOP		CSS
(D)→(B)		0	1us	0

在高速内部振荡器时钟运行中不需要。

(9) CPU 从副系统时钟运行(D)转移到高速系统时钟运行(C)。

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志	OSTS寄存器	CSC寄存器	OSTC寄存器	CKC寄存器
			MSTOP		CSS
(D)→(C) (X1时钟)		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 ≤ OSTC寄存器设置的振荡稳定时间

注意:

1. 在解除复位后, 只能通过8位存储器操作指令写1次时钟运行模式控制寄存器 (CMC)。
2. 必须在电源电压达到设置的时钟可运行电压 (参照数据手册的电气特性) 后设置时钟。

备注:

1. x: 忽略。
2. 表4-16的(A)~(I)对应图4-6的(A)~(I)。

表4-16: CPU时钟的转移和SFR寄存器的设置例子(4/4)

(10) CPU 在高速内部振荡器时钟运行中(B)转移到睡眠模式(E)。

CPU 在高速系统时钟运行中(C)转移到睡眠模式(F)。

CPU 在副系统时钟运行中(D)转移到睡眠模式(G)。

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

(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		—	

注意:

1. 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
2. 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)。
3. 必须在电源电压达到设置的时钟可运行电压 (参照数据手册的电气特性) 后设置时钟。

备注:

4. x: 忽略。
5. 表4-16的(A)~(I)对应图4-6的(A)~(I)。



## 4.6.5 CPU时钟转移前的条件和转移后的处理

CPU时钟转移前的条件和转移后的处理如下所示。

表4-17：有关CPU时钟的转移(1/2)

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

表4-17: 有关CPU时钟的转移(2/2)

CPU时钟		转移前的条件	转移后的处理
转移前	转移后		
XT1时钟	高速内部振荡器	高速内部振荡器正在振荡并且选择高速内部振荡器时钟作为主系统时钟。	能停止XT1的振荡 (XTSTOP=1)。
	时钟	<ul style="list-style-type: none"> <li>• HIOSTOP=0, MCS=0</li> </ul>	
	X1时钟	X1振荡稳定并且选择高速系统时钟作为主系统时钟。	
		<ul style="list-style-type: none"> <li>• OSCSEL=1, EXCLK=0, MSTOP=0</li> <li>• 经过振荡稳定时间后</li> <li>• MCS=1</li> </ul>	
		外部主系统时钟	
外部副系统时钟	不能转移。	—	
外部副系统时钟	高速内部振荡器	高速内部振荡器正在振荡并且选择高速内部振荡器时钟作为主系统时钟。	(XTSTOP=1)。
	时钟	<ul style="list-style-type: none"> <li>• HIOSTOP=0, MCS=0</li> </ul>	
	X1时钟	X1振荡稳定并且选择高速系统时钟作为主系统时钟。	
		<ul style="list-style-type: none"> <li>• OSCSEL=1, EXCLK=0, MSTOP=0</li> <li>• 经过振荡稳定时间后</li> <li>• MCS=1</li> </ul>	
		外部主系统时钟	
XT1时钟	不能转移。	—	

### 4.6.6 CPU时钟和主系统时钟的切换所需时间

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

在改写 CKC 寄存器后不立即进行实际的切换,而是在更改 CKC 寄存器后仍然以切换前的时钟继续运行数个时钟(参照表 4-18~表 4-20)。

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4-18: 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4-18。
F <sub>MAIN</sub>	↔	F <sub>SUB</sub>	参照表4-19。

表4-19: 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4-20: F<sub>MAIN</sub> F<sub>SUB</sub>所需要的最大时钟数

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

备注:

1. 表4-19和表4-20中的时钟数是切换前的CPU时钟数。
2. 表4-19和表4-20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)

$$2F_{MX}/F_{IH}=2(10/8)=2.53\text{个时钟}$$

## 4.6.7 时钟振荡停止前的条件

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

表4-21：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		
XT1时钟	CLS=0 (CPU以副系统时钟以外的时钟运行)	XTSTOP=1
外部副系统时钟		

## 4.7 高速内振校正功能

### 4.7.1 高速内振自调整功能

本功能以副系统时钟  $F_{SUB}$  (32.768KHz) 为基准来测量高速内振的频率，并实时校正高速内振  $F_{HOCO}$  的频率精度。

表 4-22 是高速内振频率校正功能的操作规范，图 4-7 是高速内振频率校正功能的动作框图。

表4-22：高速内振频率校正功能的操作规范

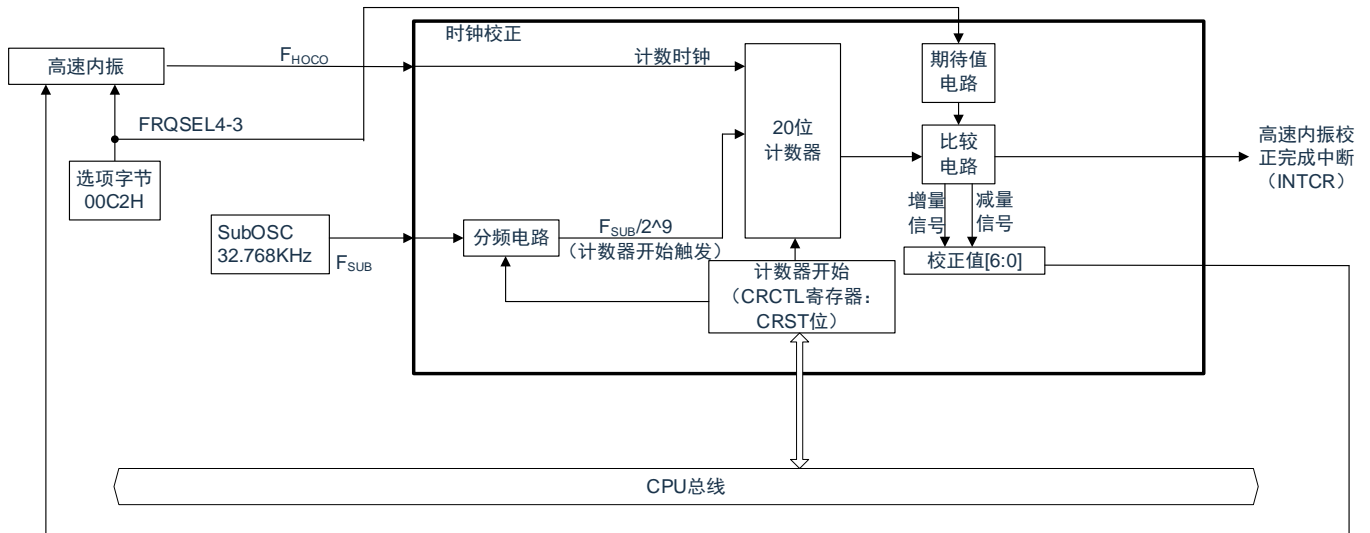
项目	内容
基准时钟	<ul style="list-style-type: none"> <li><math>F_{SUB}/2^9</math> (副系统时钟32.768KHz)</li> </ul>
校正对象时钟	<ul style="list-style-type: none"> <li><math>F_{HOCO}</math> (高速内振)</li> </ul>
动作模式	<ul style="list-style-type: none"> <li>连续动作模式 连续进行高速内振频率校正的模式</li> <li>间隔动作模式 使用定时器时钟端等，来间隔进行高速内振频率校正的模式</li> </ul>
时钟精度调整功能	<ul style="list-style-type: none"> <li>校正时间：校正周期 (31.2ms) × (校正次数-0.5) <sup>注</sup></li> </ul>
中断	<ul style="list-style-type: none"> <li>高速内振频率校正完成时产生中断 (中断许可打开时)</li> </ul>

注：校正时间：根据校正次数不同而变化。

校正周期：频率测定阶段和频率校正阶段的总时间。

校正次数：频率收束到期待值范围的校正次数。

图4-7：高速内振频率校正功能的动作框图



## 4.7.2 寄存器说明

表 4-23 是高速内振频率校正功能使用的寄存器一览。

表4-23：高速内振频率校正功能寄存器一览

项目	结构
控制寄存器	高速内振频率校正控制寄存器（HOCOFC）

### 4.7.2.1 高速内振频率校正控制寄存器（HOCOFC）

高速内振频率校正功能的控制寄存器。

通过 8 位存储器操作指令设定 HOCOFC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表4-24：高速内振频率校正控制寄存器（HOCOFC）的格式

地址：0x40022400

复位后：00H R/W

符号	7	6	5	4	3	2	1	0
HOCOFC	FCMD	FCIE	0	0	0	0	0	FCST

FCMD <sup>注1</sup>	高速内振频率校正功能动作模式
0	连续动作模式
1	间隔动作模式

FCIE	高速内振频率校正完成的中断控制
0	高速内振频率校正完成后不产生中断
1	高速内振频率校正完成后产生中断

FCST <sup>注2</sup>	高速内振频率校正电路动作控制/状态
0	高速内振频率校正电路动作停止/停止中
1	高速内振频率校正电路动作开始/动作中

连续动作模式时，软件写0来停止动作。

间隔动作模式时，校正完成后，硬件将FCST位清除。

注1：FCST位为1时，禁止改写FCMD位。

注2：给FCST位写1时，先确认当前FCST位的值为0再给它写1。由于硬件清除优先，间隔动作完成后立刻（高速内振频率校正完成中断产生时）给FCST位写1时，应在高速内振频率校正完成中断产生后至少经过 $F_{HOCO}$  1个周期再执行该操作。

注意：

1. 给FCST位写0（高速内振频率校正电路动作停止）后， $F_{HOCO}$  2个周期内禁止给FCST位写1（高速内振频率校正电路动作开始）。
2. bit5~1必须写0。

### 4.7.3 动作说明

#### 4.7.3.1 动作概要

高速内振频率校正功能以副系统时钟（F<sub>SUB</sub>）为基准，生成校正周期，测量高速内振的频率，并实时校正高速内振的频率精度。时钟调整重复频率测量阶段和频率校正阶段的操作。在频率测量阶段进行校正演算，在频率校正阶段将反映校正演算结果的校正值保存。

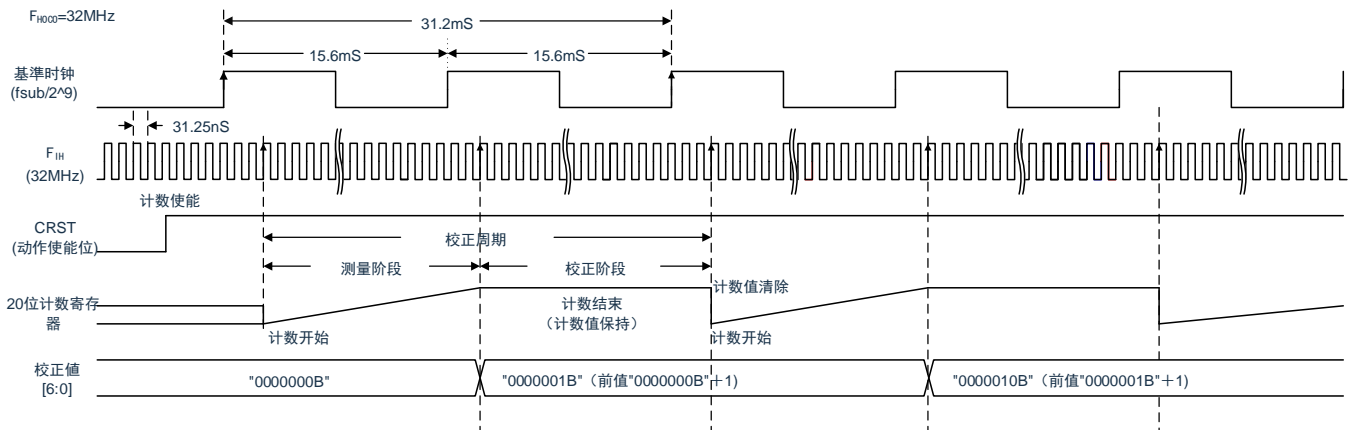
表4-25是高速内振输入频率和校正周期，图4-8是高速内振频率校正动作时序图（详细）。

表4-25：高速内振输入频率和校正周期

F <sub>HOCO</sub> (MHz)	FRQSEL4-FRQSEL3 <sup>注</sup>	校正周期 (ms)
64	11	31.2
48	10	(频率测量阶段+频率校正阶段)

在校正周期的频率测量阶段，使用高速内振计数，根据计数值和期待值的大小结果来校正高速内振的频率。

图 4-8：高速内振频率校正动作时序图（详细）



注：FRQSEL4-FRQSEL3为选项字节00C2H的bit4-bit3。

注意：连续动作模式和间隔动作模式的基本动作是相同的。不同点是FCST位的清除是由软件控制还是硬件控制。另，只有系统复位能清楚校正值。

(1) 连续动作模式

连续动作模式时，高速内振时钟频率校正动作一直进行。HOCOFC 寄存器的 FCMD 位设置为 0，即为连续动作模式。

HOCOFC 寄存器的 FCST 位设置为 1 时高速内振时钟频率校正动作开始。同样，FCST 位设置为 0 时高速内振时钟频率校正动作停止。

高速内振时钟频率校正动作后，在基准时钟 ( $F_{SUB}/2^9$ ) 的上升沿频率计数器开始计数，在下一个基准时钟 ( $F_{SUB}/2^9$ ) 的上升沿停止计数。（频率测量阶段）

然后，将计数值和期待值进行比较，根据以下所述进行校正值调整。（频率校正阶段）

计数值比期待值大时：校正值-1

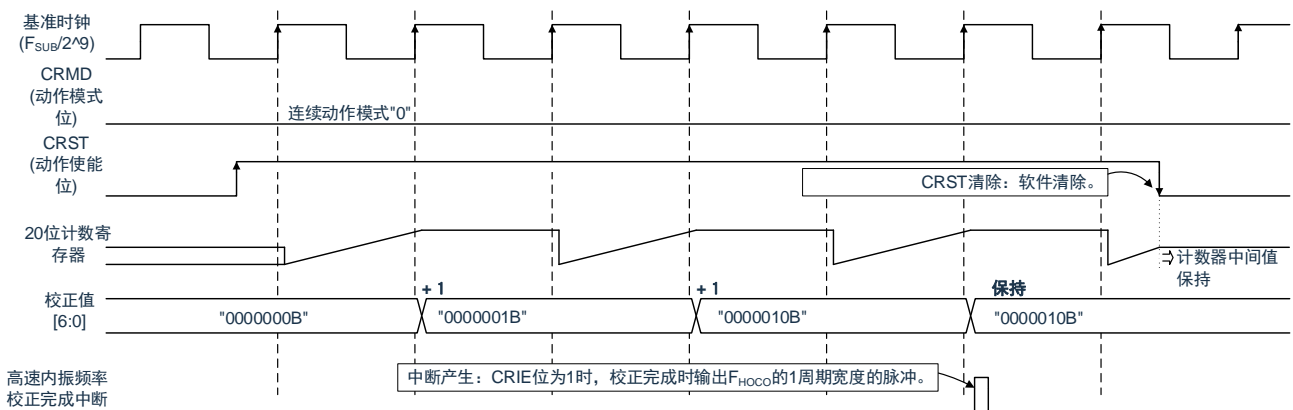
计数值比期待值小时：校正值+1

计数值在期待值范围内时：校正值保持（高速内振时钟频率校正结束）

HOCOFC 寄存器的 FCIE 位设定为 1 的情况下，高速内振时钟频率校正完成后产生高速内振时钟频率校正完成中断。连续动作模式时，高速内振时钟频率校正功能一直重复频率测量阶段和频率校正阶段，直到停止高速内振时钟频率校正功能。

图 4-9 是连续动作模式的时序图。

图4-9：连续动作模式时序图





(2) 间隔动作模式

间隔动作模式时，使用定时器中断等，间歇的进行高速内振时钟频率校正。HOCOFC 寄存器的 FCMD 位设置为 1，即为间隔动作模式。

HOCOFC 寄存器的 FCST 位设置为 1 时高速内振时钟频率校正动作开始。

高速内振时钟频率校正动作后，在基准时钟 ( $F_{SUB}/2^9$ ) 的上升沿频率计数器开始计数，在下一个基准时钟 ( $F_{SUB}/2^9$ ) 的上升沿停止计数。（频率测量阶段）

然后，将计数值和期待值进行比较，根据以下所述进行校正值调整。（频率校正阶段）

计数值比期待值大时：校正值-1

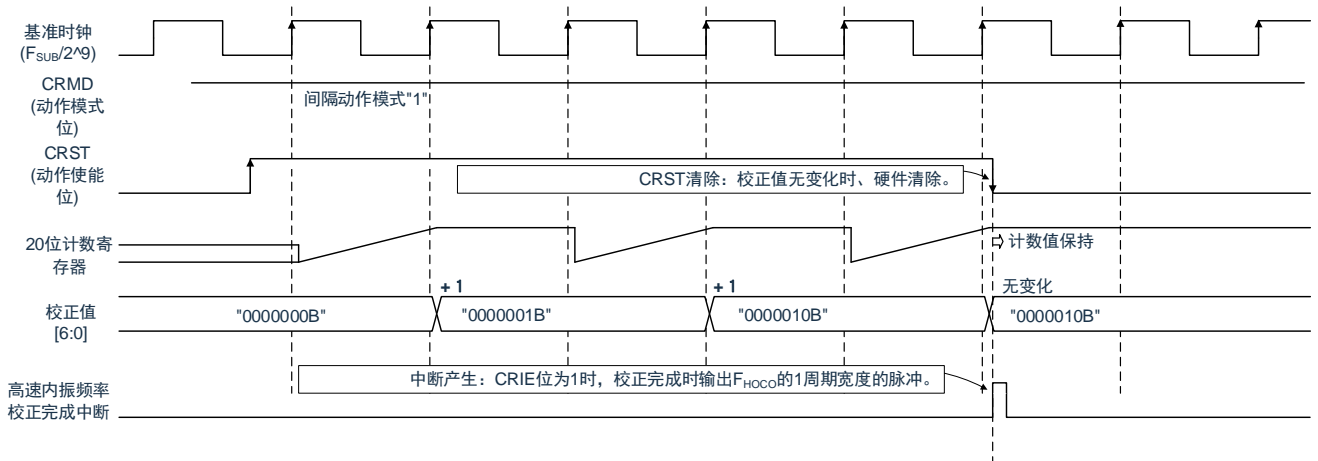
计数值比期待值小时：校正值+1

计数值在期待值范围内时：校正值保持（高速内振时钟频率校正结束）

HOCOFC 寄存器的 FCIE 位设定为 1 的情况下，高速内振时钟频率校正完成后产生高速内振时钟频率校正完成中断。间隔动作模式时，高速内振时钟频率校正功能重复频率测量阶段和频率校正阶段，到高速内振时钟频率校正完成后停止高速内振时钟频率校正功能。

图4-10是间隔动作模式的时序图。

图4-10：间隔动作模式时序图



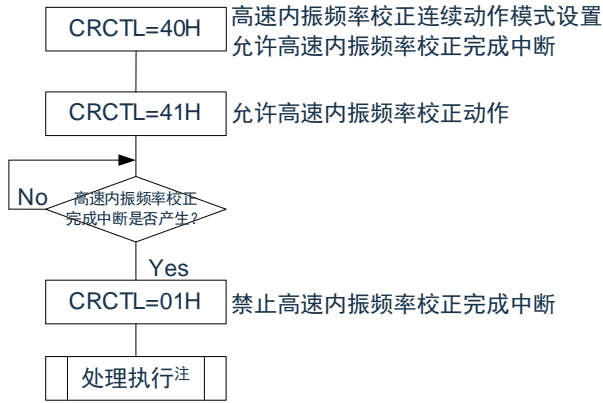
### 4.7.3.2 动作设置流程

高速内振时钟频率校正功能使用时的动作开始/停止流程如下图所示。

图4-11：动作模式设定流程（例）

#### <连续动作模式>

##### ■ 动作开始流程

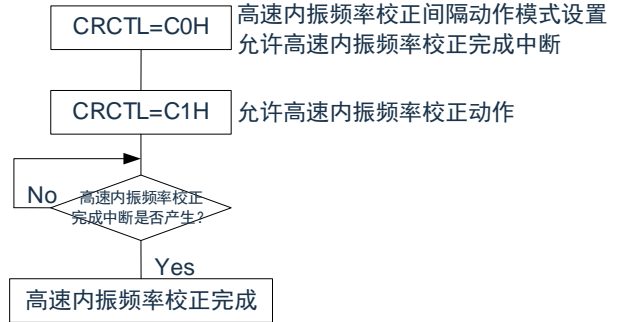


##### ■ 动作停止流程

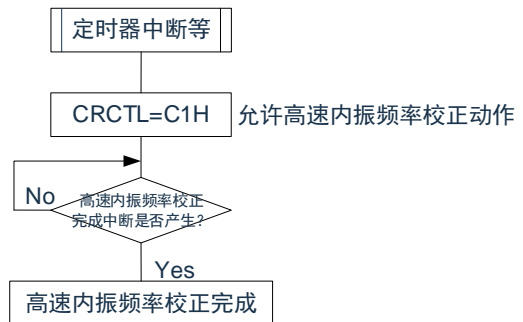


#### <间隔动作模式>

##### ■ 动作开始流程（1）



##### ■ 动作开始流程（2）



注：在停止高速内振时钟频率校正功能前，高速内振时钟频率校正动作反复执行。

## 4.7.4 使用注意事项

### 4.7.4.1 SFR访问

关于间隔动作模式时的 FCST 位的控制，给 FCST 位写 1 时，必须先确认当前 FCST 位的值为 0 再给它写 1。由于硬件清除优先，间隔动作完成后立刻（高速内振频率校正完成中断产生时）给 FCST 位写 1 时，应在高速内振频率校正完成中断产生后至少经过  $F_{HOCO}$  1 个周期再执行该操作。

### 4.7.4.2 复位时动作

进入深度睡眠之前，必须停止高速内振时钟频率校正功能。

## 第5章 通用定时器单元Timer4

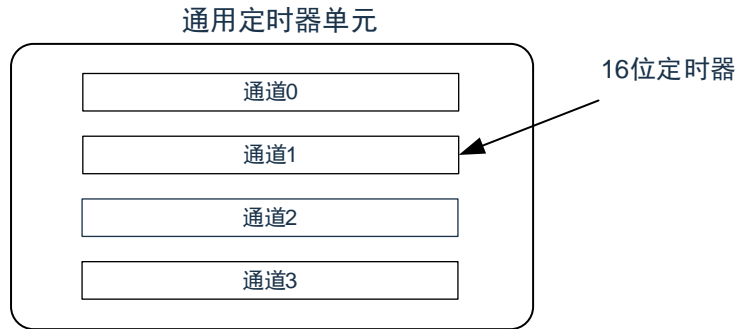
本制品搭载两个通用定时器单元,每个单元含有 4 个通道。

说明:

- 1、本章下文中的标号“m”代表单元号，本产品搭载了两个通用定时器 Timer4，因此 m=0, 1。
- 2、本章下文中的标号“n”代表通道号（在本章中 n=0~3）。

每个通用定时器单元有 4 个 16 位定时器。

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



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5.8.1）</li> <li>• 方波输出（参照5.8.1）</li> <li>• 外部事件计数器（参照5.8.2）</li> <li>• 分频器（参照5.8.3）</li> <li>• 输入脉冲间隔的测量（参照5.8.4）</li> <li>• 输入信号的高低电平宽度的测量（参照5.8.5）</li> <li>• 延迟计数器（参照5.8.6）</li> </ul>	<ul style="list-style-type: none"> <li>• 单触发脉冲输出（参照5.9.1）</li> <li>• PWM输出（参照5.9.2）</li> <li>• 多重PWM输出（参照5.9.3）</li> </ul>

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）

## 5.1 通用定时器单元的功能

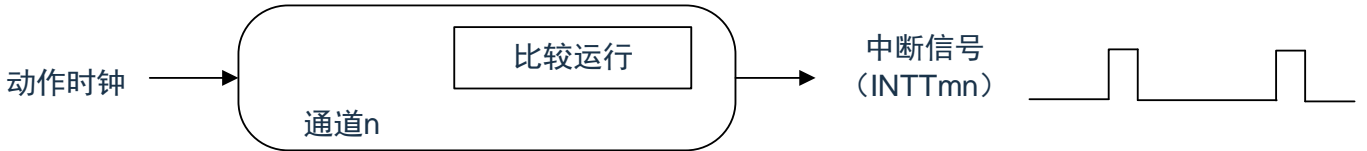
通用定时器单元有以下功能：

### 5.1.1 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。

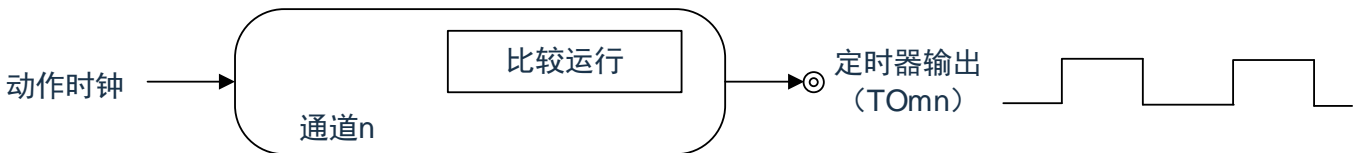
#### (1) 间隔定时器

能用作以固定间隔产生中断（INTTMmn）的基准定时器。



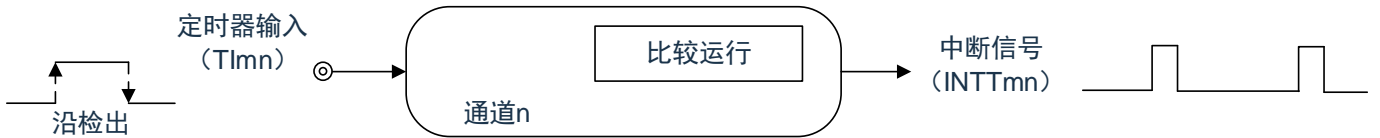
#### (2) 方波输出

每当产生INTTMmn 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。



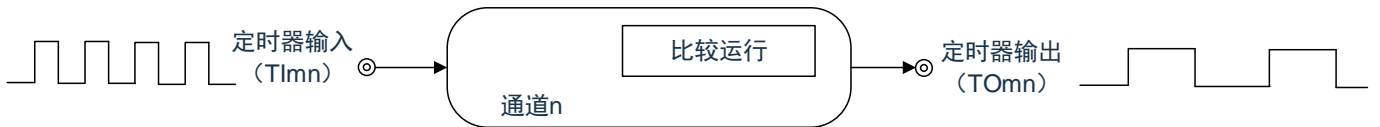
#### (3) 外部事件计数器

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。



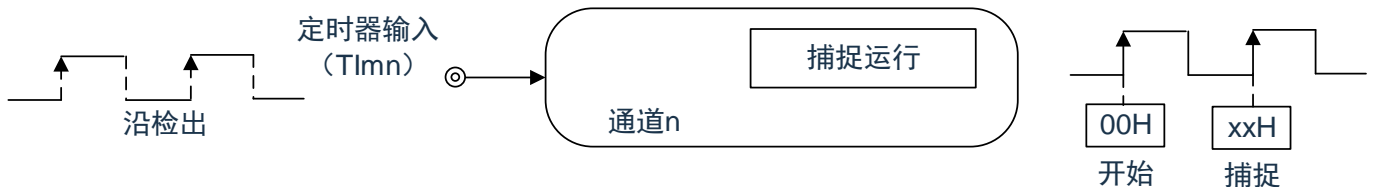
#### (4) 分频器功能（只限于单元0的通道0）

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。



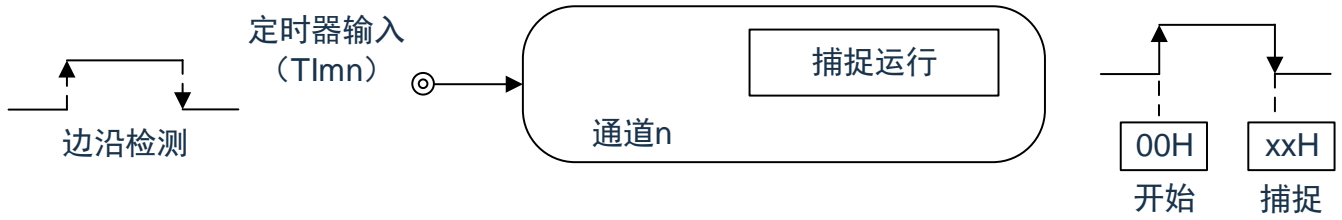
#### (5) 输入脉冲间隔的测量

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。



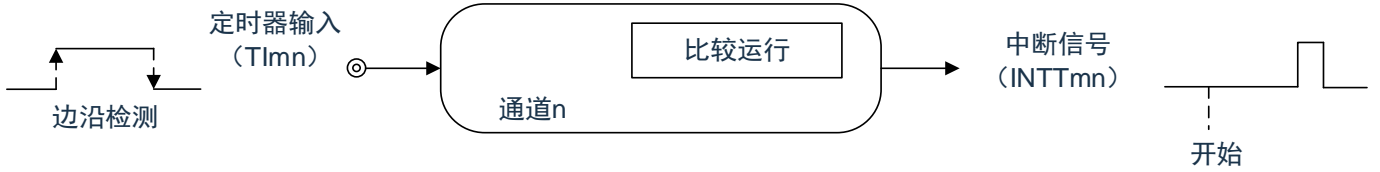
(6) 输入信号的高低电平宽度的测量

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, 1) n: 通道号 (n=0~3)
2. 通道0~3的定时器输入/输出引脚可配置请参照“第2章 引脚功能”。

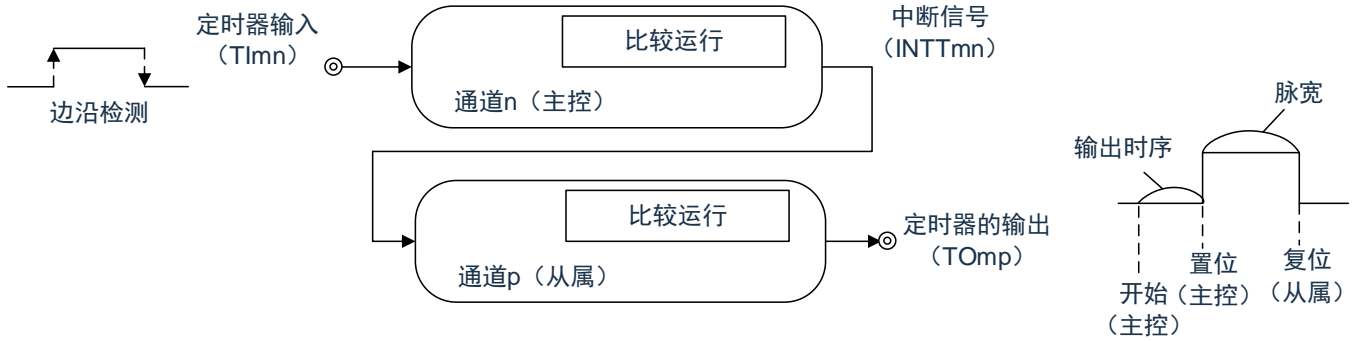
### 5.1.2 多通道联动运行功能

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。

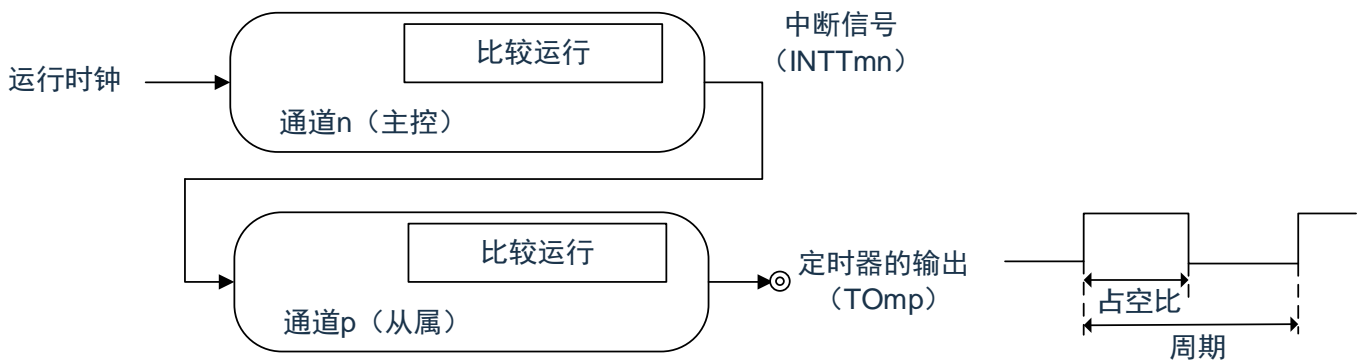
#### (1) 单触发脉冲输出

将2个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



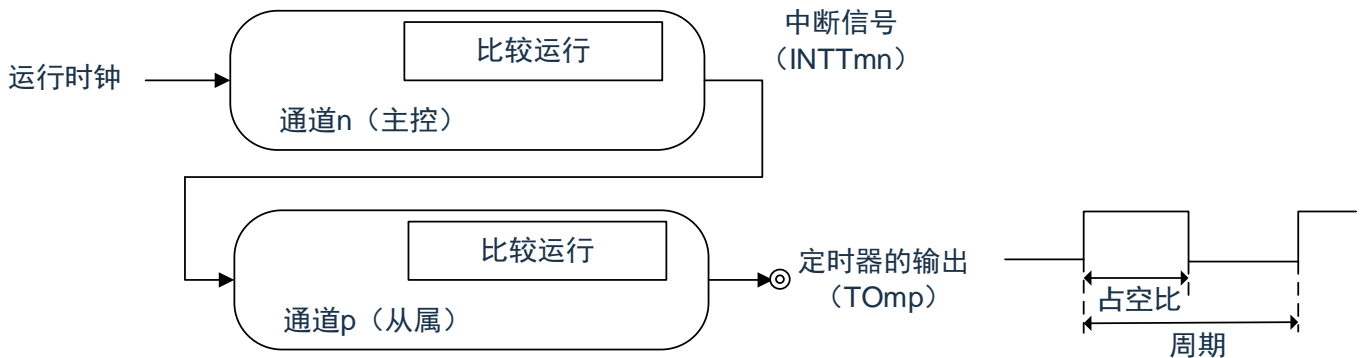
#### (2) PWM (Pulse Width Modulation) 输出

将2个通道成对使用，生成能任意设定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。



备注：

1. 有关多通道联动运行功能规则的详细内容，请参照“5.4.1 多通道联动运行功能的基本规则”。
2. m: 单元号 (m=0,1) n: 通道号 (n=0~3) p、q: 从属通道号 (n<p<q≤3)

### 5.1.3 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。只能使用通道1和通道3。

备注：

1. 在使用8位定时器运行功能时，有几个规则。
2. 详细内容请参照“5.4.2 8位定时器运行功能的基本规则（只限于通道1和通道3）”。

### 5.1.4 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支持功能的运行设定，请参照“13.3.14 输入切换控制寄存器（ISC）”和“5.8.5 作为输入信号高低电平宽度测量的运行”。



## 5.2 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5-1：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 <sup>注1</sup> 、TI10~TI13 <sup>注1</sup>
定时器的输出	TO00~TO03 <sup>注1</sup> 、TO10~TO13 <sup>注1</sup> 、输出控制电路
控制寄存器	<单元设定的寄存器> <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 (TIOS0)<sup>注2</sup></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>• 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3</sup></li> <li>• 端口模式寄存器 (PMxx)<sup>注3</sup></li> <li>• 端口输出复用功能配置寄存器 (PxxCFG)<sup>注3</sup></li> <li>• 端口输入复用功能配置寄存器 (TI1XPCFG)<sup>注3</sup></li> </ul>

注1：通用定时器单元0的输入/输出引脚复用到固定端口，通用定时器单元1的通道0~3的定时器输入/输出引脚可任意配置到除RESETB以外的各端口。详细内容请参照“第2章 引脚功能”。

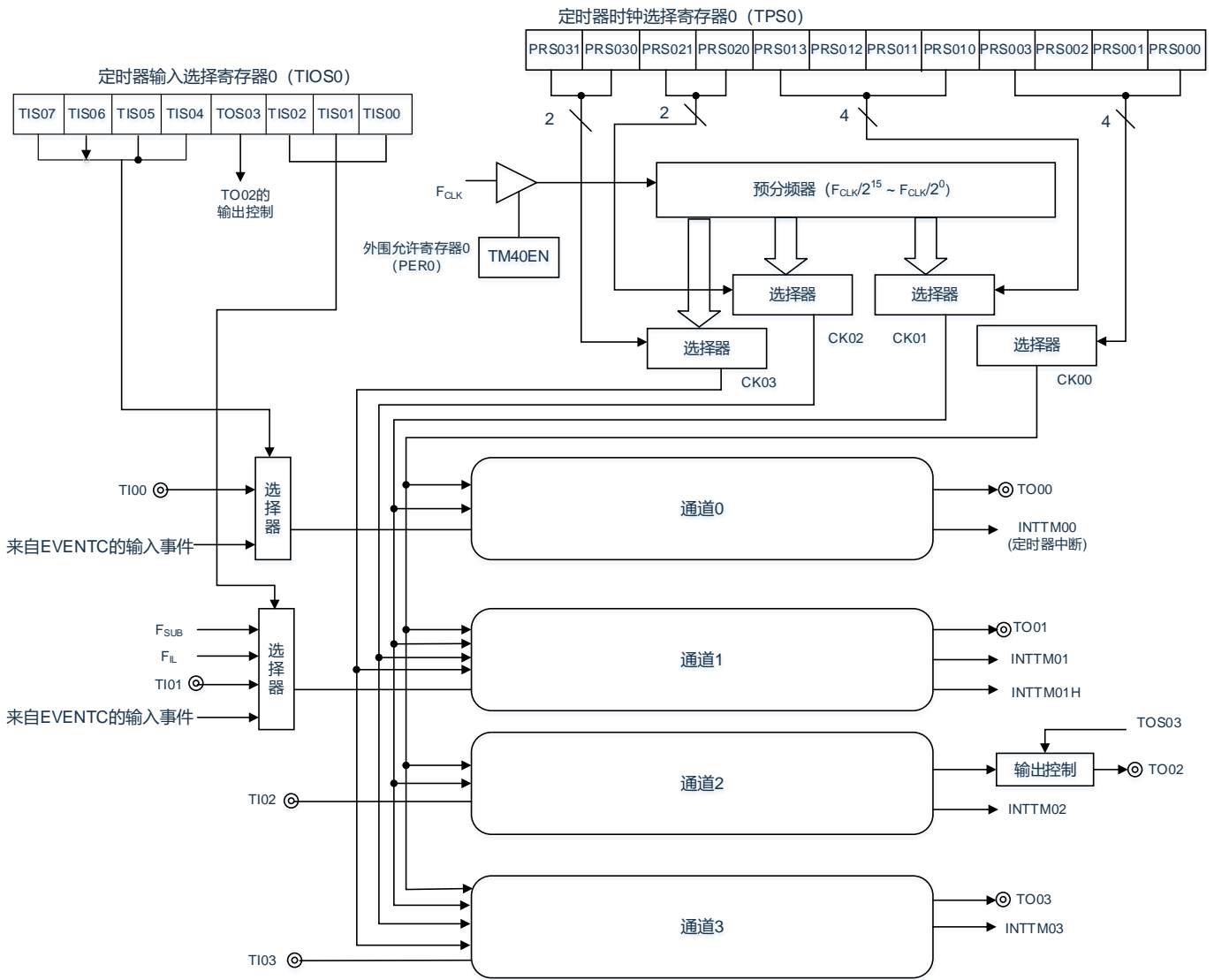
注2：仅用于单元0的通道选择。

注3：用于通道0~3的定时器输入/输出引脚配置。详细内容请参照“第2章 引脚功能”。

备注：m：单元号 (m=0, 1) n：通道号 (n=0~3)

通用定时器单元的框图如图5-1所示。

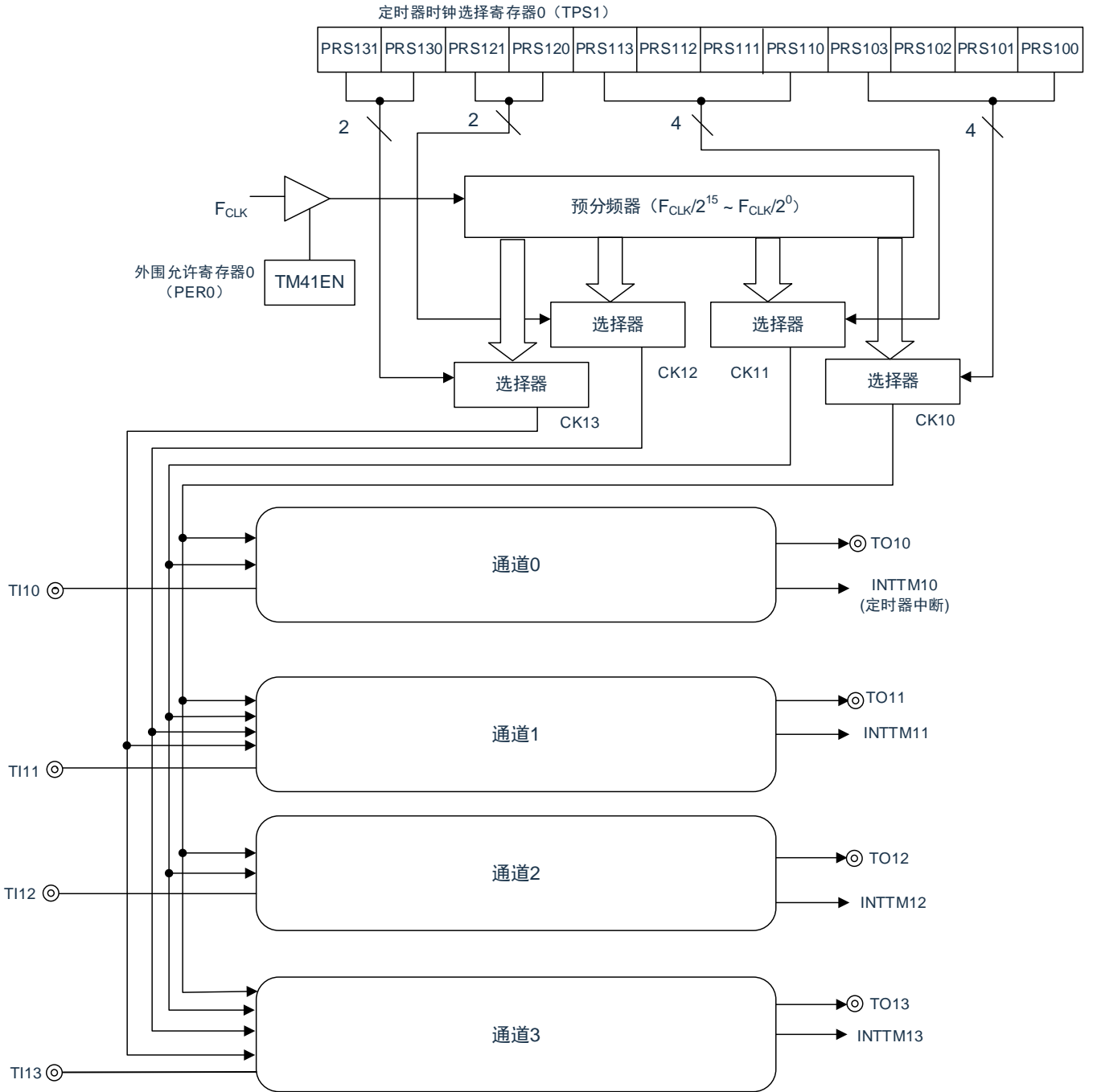
图5-1：通用定时器单元0的整体框图



备注：F<sub>SUB</sub>：副系统时钟频率

F<sub>IL</sub>：低速内部振荡器时钟频率

图5-2: 通用定时器单元1的整体框图



## 5.2.1 通用定时器单元0寄存器列表

单元0的寄存器基址：0x40041C00

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

## 5.2.2 通用定时器单元1寄存器列表

单元1的寄存器基址：0x40042000

偏移地址	寄存器名	读写属性	位宽	复位值
0x180	TCR10	R	16	FFFFH
0x182	TCR11	R	16	FFFFH
0x184	TCR12	R	16	FFFFH
0x186	TCR13	R	16	FFFFH
0x190	TMR10	R/W	16	0000H
0x192	TMR11	R/W	16	0000H
0x194	TMR12	R/W	16	0000H
0x196	TMR13	R/W	16	0000H
0x1A0	TSR10	R	16	0000H
0x1A0	TSR10L	R	8	00H
0x1A2	TSR11	R	16	0000H
0x1A2	TSR11L	R	8	00H
0x1A4	TSR12	R	16	0000H
0x1A4	TSR12L	R	8	00H
0x1A6	TSR13	R	16	0000H
0x1A6	TSR13L	R	8	00H
0x1B0	TE1	R	16	0000H
0x1B0	TE1L	R	8	00H
0x1B2	TS1	R/W	16	0000H
0x1B2	TS1L	R/W	8	00H
0x1B4	TT1	R/W	16	0000H
0x1B4	TT1L	R/W	8	00H
0x1B6	TPS1	R/W	16	0000H
0x1B8	TO1	R/W	16	0000H
0x1B8	TO1L	R/W	8	00H
0x1BA	TOE1	R/W	16	0000H
0x1BA	TOE1L	R/W	8	00H
0x1BC	TOL1	R/W	16	0000H
0x1BC	TOL1L	R/W	8	00H
0x1BE	TOM1	R/W	16	0000H
0x1BE	TOM1L	R/W	8	00H
0x318	TDR10	R/W	16	0000H
0x31A	TDR11	R/W	16	0000H
0x31A	TDR11L	R/W	8	00H
0x31B	TDR11H	R/W	8	00H
0x364	TDR12	R/W	16	0000H
0x366	TDR13	R/W	16	0000H
0x366	TDR13L	R/W	8	00H
0x367	TDR13H	R/W	8	00H

### 5.2.3 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“5.3.3 定时器模式寄存器 mn (TMRmn)”）。

表5-2: 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																

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”。

- (1) 当产生复位信号时
- (2) 当清除外围允许寄存器 0 (PER0) 的 TM4mEN 位时
- (3) 在 PWM 输出模式中从属通道的计数结束时
- (4) 在延迟计数模式中从属通道的计数结束时
- (5) 在单触发脉冲输出模式中主控/从属通道的计数结束时
- (6) 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”。

- (1) 在捕捉模式中输入开始触发时
- (2) 在捕捉模式中捕捉结束时

备注:

1. 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)。
2. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

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5-3: 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1</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1: 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, 1) n: 通道号 (n=0~3)

### 5.2.4 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”。

表5-4: 定时器数据寄存器mn (TDRmn) (n=0、2) 的表格

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

表5-5: 定时器数据寄存器mn (TDRmn) (n=1、3) 的表格  
(TDR01H 可支持8bit操作) (TDR01L 可支持8bit操作)

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

(1) 定时器数据寄存器 mn (TDRmn) 用作比较寄存器的情况

从 TDRmn 寄存器的设定值开始递减计数，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。

(2) 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 TI<sub>mn</sub> 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) 来设定捕捉触发的选择。

备注:

1. 即使输入捕捉触发信号，设定为比较功能的TDRmn寄存器也不进行捕捉运行。
2. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

## 5.3 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)  
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)  
噪声滤波器允许寄存器 1 (NFEN1)  
噪声滤波器允许寄存器 2 (NFEN2)  
端口模式控制寄存器 (PMCxx)  
端口模式寄存器 (PMxx)  
端口复用功能配置寄存器 (PxxCFG)

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定初始值。

备注：单元号 (m=0, 1) n: 通道号 (n=0~3)



### 5.3.1 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定 PER0 寄存器。在产生复位信号后，PER0 寄存器的值变为“00H”。

表5-6: 外围允许寄存器0 (PER0) 的表格

地址: 0x40020420		复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0	
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	TM41EN	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。 • 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。

TM41EN	通用定时器单元1的输入时钟的控制
0	停止提供输入时钟。 • 不能写通用定时器单元1使用的SFR。 • 通用定时器单元1处于复位状态。
1	提供输入时钟。 • 能读写通用定时器单元1使用的SFR。

注意: 要设定通用定时器单元时, 必须先在TM4mEN位为“1”的状态下设定以下的寄存器。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)、噪声滤波器允许寄存器1 (NFEN1)、噪声滤波器允许寄存器2 (NFEN2)、端口模式控制寄存器PMCx、端口模式寄存器PMx和端口复用功能配置寄存器PxxCFG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)

### 5.3.2 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。

- 1) 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)。
- 2) 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)。
- 3) 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)。
- 4) 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定 TPSm 寄存器。在产生复位信号后，TPSm 寄存器的值变为“0000H”。

表5-7：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			PRS m31	PRS m31			PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	运行时钟（CKmk）的选择 <sup>注</sup> （k=0、1）
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：必须将bit15、14、11、10置“0”。

注意：

1. 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。
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、1，n=0~3），就不能使用通用定时器单元的中断请求。
3. 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。详细内容请参照“5.5.1 计数时钟（F<sub>TCLK</sub>）”。
4. F<sub>CLK</sub>：CPU/外围硬件的时钟频率。

表5-7: 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			PRS m31	PRS m31			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>
0	0	$F_{CLK}/2$
0	1	$F_{CLK}/2^2$
1	0	$F_{CLK}/2^4$
1	1	$F_{CLK}/2^6$

PRSm31	PRSm30	运行时钟 (CKm3) 的选择 <sup>注</sup>
0	0	$F_{CLK}/2^8$
0	1	$F_{CLK}/2^{10}$
1	0	$F_{CLK}/2^{12}$
1	1	$F_{CLK}/2^{14}$

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下表所示的间隔时间。

表5-8: 运行时钟CKSm2和CKSm3能设定的间隔时间

时钟		间隔时间 <sup>注</sup> ( $F_{CLK}=32\text{MHz}$ )			
		10us	100us	1ms	10ms
CKm2	$F_{CLK}/2$	○	—	—	—
	$F_{CLK}/2^2$	○	—	—	—
	$F_{CLK}/2^4$	○	○	—	—
	$F_{CLK}/2^6$	○	○	—	—
CKm3	$F_{CLK}/2^8$	—	○	○	—
	$F_{CLK}/2^{10}$	—	○	○	—
	$F_{CLK}/2^{12}$	—	—	○	○
	$F_{CLK}/2^{14}$	—	—	○	○

注: 必须将bit15、14、11、10置“0”。

注意:

1. 在更改选择为 $F_{CLK}$ 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止通用定时器单元 (TTm=000FH)。即使在选择运行时钟 ( $F_{MCK}$ ) 或者TIMn引脚输入信号的有效边沿时, 也需要停止通用定时器单元。
2. ○包含5%以内的误差。
3.  $F_{CLK}$ : CPU/外围硬件的时钟频率。
4. 有关TPSm寄存器所选 $F_{CLK}/2^i$ 波形的详细内容, 请参照“5.5.1 计数时钟 ( $F_{TCLK}$ )”。

### 5.3.3 定时器模式寄存器mn (TMRmn)

TMRmn 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、计数时钟的选择、主控/从属的选择、16 位/8 位定时器的选择（只限于单元 0 的通道 1 和通道 3）、开始触发和捕捉触发的设定、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）的设定。

禁止在运行中 ( $TE_{mn}=1$ ) 改写 TMRmn 寄存器。但是，能在一部分的功能运行中 ( $TE_{mn}=1$ ) 改写 bit7 和 bit6 ( $CIS_{mn1}$ 、 $CIS_{mn0}$ )（详细内容请参照“5.8 通用定时器单元的独立通道运行功能”和“5.9 通用定时器单的多通道联动运行功能”）。

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

备注：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”。

表5-9: 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	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	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	MD mn3	MD mn2	MD mn1	MD mn0

CKSmn1	CKSmn0	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) 设定的运行时钟CKm0
0	1	定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm2
1	0	定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm1
1	1	定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm3

运行时钟 (F<sub>MCK</sub>) 用于边沿检测电路。通过设定CCSmn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。

CCSmn	通道n计数时钟 (F <sub>TCLK</sub> ) 的选择
0	CKSmn0位和CKSmn1位指定的运行时钟 (F <sub>MCK</sub> )
1	TImn引脚输入信号的有效边沿 单元0的情况: 通道0: TIOS0选择的输入信号的有效边沿 通道1: TIOS0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: 必须将bit13、5、4置“0”。

备注:

- 要更改选择为F<sub>TCLK</sub>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<sub>MCK</sub>) 或者TImn引脚输入信号的有效边沿作为计数时钟 (F<sub>TCLK</sub>), 也必须停止定时器阵列单元 (TTm=00FFH)。
- m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

表5-9: 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	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	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	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设定为主控通道 (MASTERmn=1)。  
通道0固定为“0” (因为通道0为最高位的通道, 所以与此位的设定无关, 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定
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			禁止设定。

注1: bit11是只读位, 固定为“0”, 忽视写操作。

备注: m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

表5-9: 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	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	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	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, 1) n: 通道号 (n=0~3)



表5-9: 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	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	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	MD mn3	MD mn2	MD mn1	MD mn0

MD mn3	MD mn2	MD mn1	通道n运行模式的设定	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			禁止设定。		
各模式的运行因MDmn0位而变(参照下表)。					

运行模式(MDmn3~MDmn1位的设定 (参照上表))	MDmn0	开始计数和中断的设定
• 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, 1) n: 通道号(n=0~3)

### 5.3.4 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 5-11。

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”。

表5-10: 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)。

表5-11: 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:

1. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)
2. 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。

### 5.3.5 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”。

表5-12: 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	TEm3	TEm2	TEm1	TEm0

TEHm3	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

TEHm1	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

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

在通道1和通道3为8位定时器模式时，TEm1和TEm3表示低8位定时器的运行允许或者停止状态。

备注：m：单元号 (m=0, 1) n：通道号 (n=0~3)

### 5.3.6 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) 进行初始化并且设定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定 TSm 寄存器。

能用 TSmL 并且通过 8 位存储器操作指令设定 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”。

表5-13: 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	TSm3	TSm2	TSm1	TSm0

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“5.5.2 计数器的开始时序”的表5-22）。

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“5.5.2 计数器的开始时序”的表5-22）。

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“5.5.2 计数器的开始时序”的表5-22）。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，从设定定时器模式寄存器 mn (TMRmn) 到将TSmn (TSHm1、TSHm3) 位置“1”为止，需要以下期间的等待：
  - ① TImn引脚噪声滤波器有效时 (TNFENmn=1)：4个运行时钟 (F<sub>MCK</sub>)
  - ② TImn引脚噪声滤波器无效时 (TNFENmn=0)：2个运行时钟 (F<sub>MCK</sub>)
3. 必须将bit15~12、10、8~4置“0”。

备注：

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

### 5.3.7 定时器通道停止寄存器m (TTm)

TTm 寄存器是设定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定 TTm 寄存器。

能用 TTmL 并且通过 8 位存储器操作指令设定 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”。

表5-14：定时器通道停止寄存器m (TTm) 的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTH <sub>m3</sub>	0	TTH <sub>m1</sub>	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, 1) n：通道号 (n=0~3)

### 5.3.8 定时器输入输出选择寄存器 (TIOS0)

TIOS0 寄存器用于对单元 0 的输入输出进行选择。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定 TIOS0 寄存器。在产生复位信号后，TIOS0 寄存器的值变为“00H”。

表5-15: 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
其他01			设定禁止

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	ELC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)
1	0	1	副系统时钟 (F <sub>SUB</sub> )
上述以外			禁止设定。

注意:

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

### 5.3.9 定时器输出允许寄存器m (TOEm)

TOEm寄存器是设定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定TOEm寄存器。

能用TOEmL并且通过8位存储器操作指令设定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”。

表5-16: 定时器输出允许寄存器m (TOEm) 的表格

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

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位设定的电平。
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, 1) n：通道号 (n=0~3)

### 5.3.10 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 TI00/TO00、TI01/TO01、TI02/TO02、TI03/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定 TOM 寄存器。

能用 TOML 并且通过 8 位存储器操作指令设定 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”。

表5-17: 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	TOM <sub>3</sub>	TOM <sub>2</sub>	TOM <sub>1</sub>	TOM <sub>0</sub>

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, 1) n：通道号 (n=0~3)



### 5.3.11 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 的反相设定。在主导通道输出模式 (TOMmn=0) 中, 此寄存器的设定无效。

通过 16 位存储器操作指令设定 TOLm 寄存器。

能用 TOLmL 并且通过 8 位存储器操作指令设定 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”。

表5-18: 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	TOL m3	TOL m2	TOL m1	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:

1. 必须将bit15~4和bit0置“0”。
2. 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。

备注: m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

### 5.3.12 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定 TOMm 寄存器。

能用 TOMmL 并且通过 8 位存储器操作指令设定 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”。

表5-19: 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, 1） 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）（有关主控通道和从属通道关系的详细内容，请参照“5.4.1 多通道联动运行功能的基本规则”）

### 5.3.13 噪声滤波器允许寄存器1 (NFEN1)

NFEN1 寄存器设定噪声滤波器是否用于单元 0 各通道定时器输入引脚的输入信号。对于需要消除噪声的引脚，必须将对应位置“1”，使噪声滤波器有效。当噪声滤波器有效时，在通过对象通道的运行时钟（ $F_{MCK}$ ）进行同步后检测 2 个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟（ $F_{MCK}$ ）进行同步注。

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

表5-20：噪声滤波器允许寄存器1 (NFEN1) 的表格

地址：0x40040471

符号	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	TI03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

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

备注：

1. 详细内容请参照“5.5.1(2) 选择TIMn引脚输入信号的有效边沿的情况（CCSmn=1）”、“5.5.2 计数器的开始时序”和“5.7 定时器输入（TIMn）的控制”。
2. 通道0~3的定时器输入/输出引脚的配置参照“第2章 引脚功能”。

### 5.3.14 噪声滤波器允许寄存器2 (NFEN2)

NFEN2 寄存器设定噪声滤波器是否用于单元 1 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<sub>MCK</sub>）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<sub>MCK</sub>）进行同步注。

通过 8 位存储器操作指令设定 NFEN2 寄存器。在产生复位信号后，NFEN2 寄存器的值变为“00H”。

表5-21：噪声滤波器允许寄存器2 (NFEN2) 的表格

地址：0x40040472

符号	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN13	TI13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

TNFEN12	TI12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

TNFEN11	TI11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

TNFEN10	TI10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

备注：

1. 详细内容请参照“5.5.1(2) 选择TIMn引脚输入信号的有效边沿的情况（CCSmn=1）”、“5.5.2 计数器的开始时序”和“5.7 定时器输入（TIMn）的控制”。
2. 通道0~3的定时器输入/输出引脚的配置参照“第2章 引脚功能”。

### 5.3.15 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，定时器 0/定时器 1 的输入/输出引脚可任意配置到除 RESETB 以外的各端口。详细内容请参照“第 2 章 引脚功能”。

在将定时器 0/定时器 1 的输出引脚复用到某端口时，必须将该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）的位置“0”。并设置端口复用功能配置寄存器（PxxCFG）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

- （例）将 P21 配置为 TO10 用作定时器输出的情况
  - 将端口模式控制寄存器 2 的 PMC21 位置“0”。
  - 将端口模式寄存器 2 的 PM21 位置“0”。
  - 将端口输出复用功能配置寄存器 P21CFG 置“0x12”。

在将定时器 0/定时器 1 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”。并设置端口复用功能配置寄存器（PxxCFG）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

- （例）将 P20 配置为 TI10 用作定时器输入的情况
  - 将端口模式控制寄存器 2 的 PMC20 位置“0”。
  - 将端口模式寄存器 2 的 PM20 位置“1”。
  - 将端口输入复用功能配置寄存器 P20CFG 置“0x0a”。

## 5.4 通用定时器单元的基本规则

### 5.4.1 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）设定为主控通道。
- 2) 通道 0 以外的任何通道都能设定为从属通道。
- 3) 只能将主控通道的低位通道设定为从属通道。

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

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

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

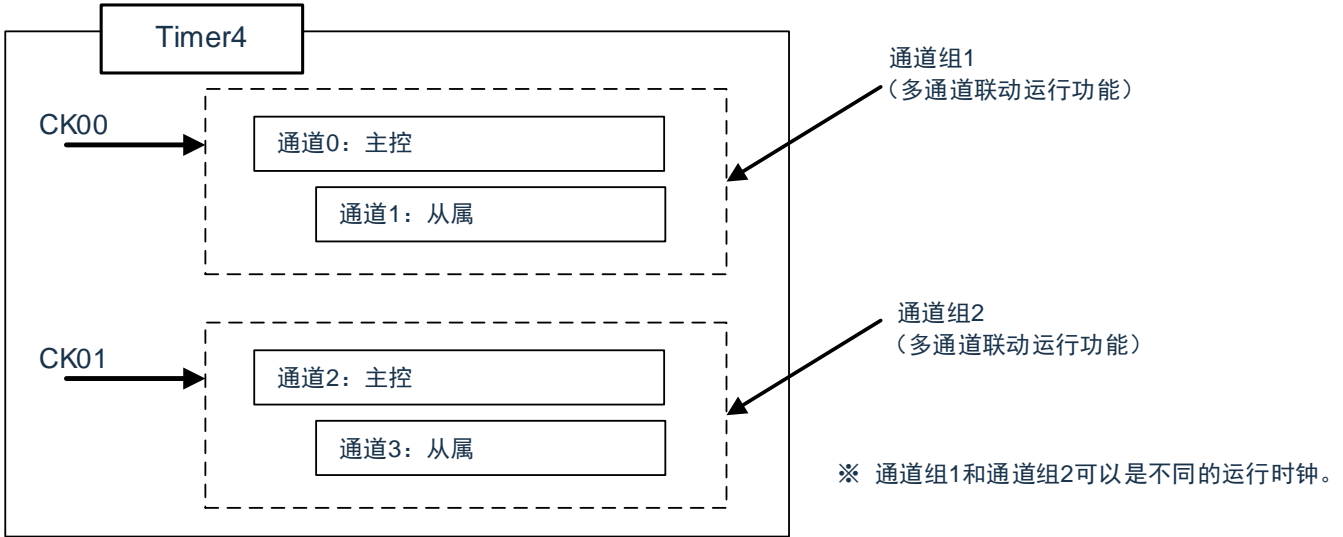
- 6) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的 CKSmn0 位和 CKSmn1 位（定时器模式寄存器 mn (TMRmn) 的 bit15 和 bit14) 的值需要是相同的设定值。
- 7) 主控通道能将 INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
- 8) 从属通道能将主控通道的 INTTMmn（中断）、开始软件触发和计数时钟用作源时钟，但是不能将自己的 INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
- 9) 主控通道不能将其他高位主控通道的 INTTMmn（中断）、开始软件触发和计数时钟用作源时钟。
- 10) 为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

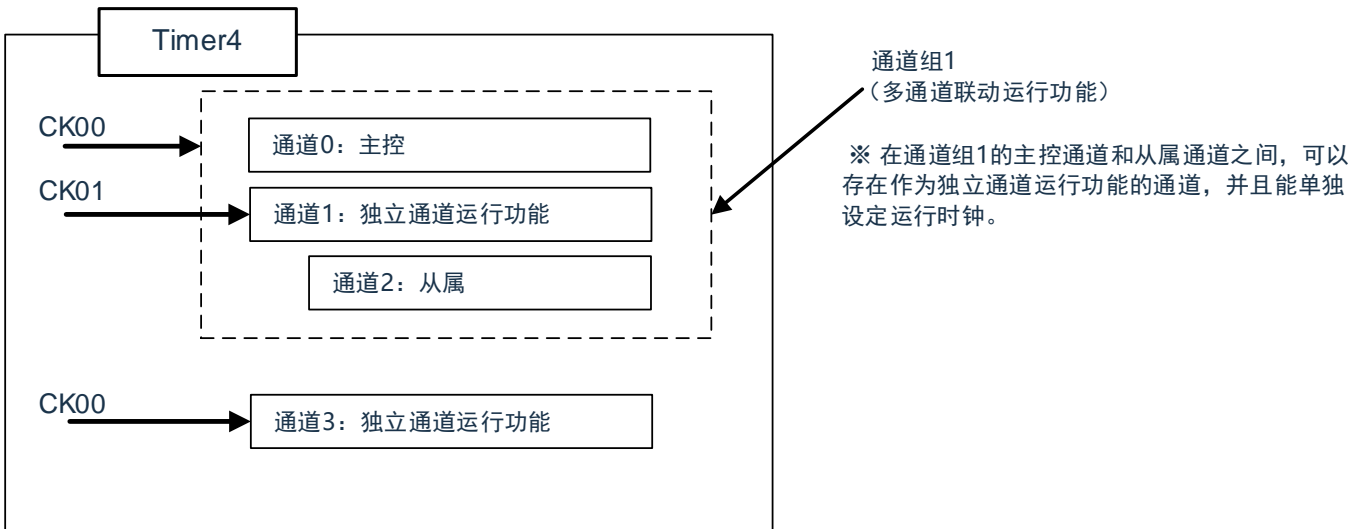
如果设定 2 个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

备注：m：单元号 (m=0, 1) n：通道号 (n=0~3)

例1:



例2:



## 5.4.2 8 位定时器运行功能的基本规则（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 INTTMm1H（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 位的设定。
- 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 寄存器的设定，有以下 3 种支持低 8 位定时器运行的功能：
  - ① 间隔定时器功能
  - ② 外部事件计数器功能
  - ③ 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：单元号（m=0） n：通道号（n=1、3）



## 5.5 计数器的运行

### 5.5.1 计数时钟 (F<sub>TCLK</sub>)

通用定时器单元的计数时钟 (F<sub>TCLK</sub>) 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:

- ① CKSmn0 位和 CKSmn1 位指定的运行时钟 (F<sub>MCK</sub>)
- ② TImn 引脚输入信号的有效边沿

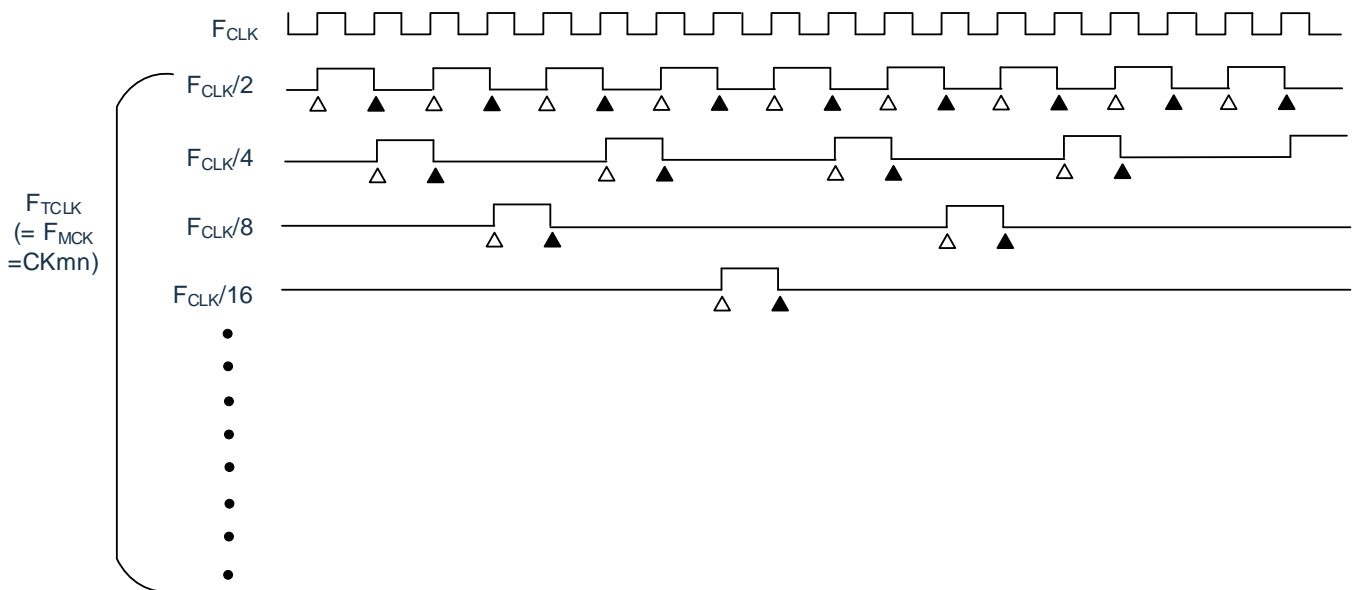
通用定时器单元被设计为与 F<sub>CLK</sub> 同步运行, 因此计数时钟 (F<sub>TCLK</sub>) 的时序如下。

(1) 选择 CKSmn0 位和 CKSmn1 位指定的运行时钟 (F<sub>MCK</sub>) 的情况 (CCSmn=0)

根据定时器时钟选择寄存器 m (TPSm) 的设定, 计数时钟 (F<sub>TCLK</sub>) 为 F<sub>CLK</sub>~F<sub>CLK</sub>/2<sup>15</sup>。但是, 当选择 F<sub>CLK</sub> 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<sub>CLK</sub> 周期为高电平的信号。当选择 F<sub>CLK</sub> 时, 固定为高电平。

为了取得与 F<sub>CLK</sub> 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<sub>CLK</sub> 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5-3: F<sub>CLK</sub>和计数时钟 (F<sub>TCLK</sub>) 的时序 (CCSmn=0的情况)



备注: Δ: 计数时钟的上升沿

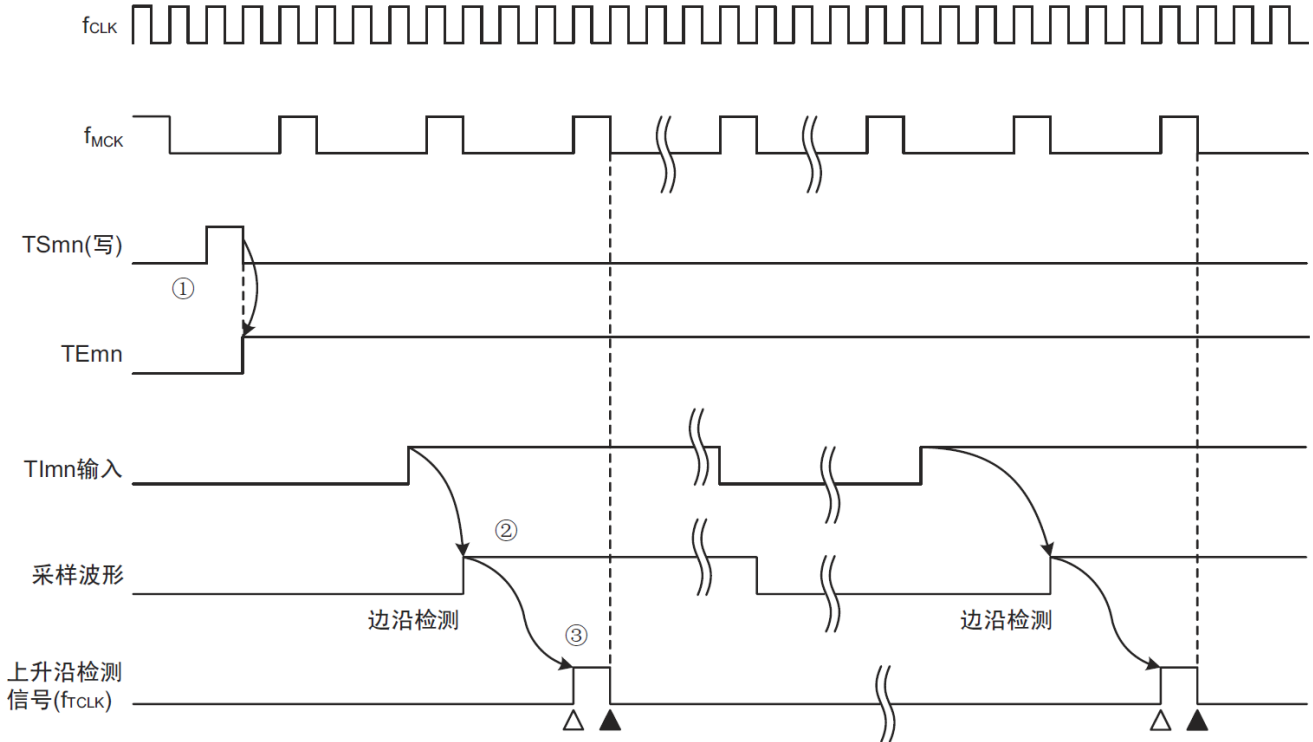
▲: 同步、计数器的递增/递减

F<sub>CLK</sub>: 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<sub>TCLK</sub>) 是检测 TImn 引脚输入信号的有效边沿并且与下一个 F<sub>MCK</sub> 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<sub>MCK</sub> 时钟的信号 (在使用噪声滤波器时, 延迟 3~4 个 F<sub>MCK</sub> 时钟)。为了取得与 F<sub>CLK</sub> 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<sub>CLK</sub> 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5-4: 计数时钟 (F<sub>TCLK</sub>) 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<sub>MCK</sub> 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. Δ: 计数时钟的上升沿  
▲: 同步、计数器的递增/递减  
F<sub>CLK</sub>: CPU外围硬件时钟  
F<sub>MCK</sub>: 通道n的运行时钟
- 2. 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

## 5.5.2 计数器的开始时序

通过将定时器通道开始寄存器 m (TSmn) 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 5-22 所示。

表5-22: 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“5.5.3(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“5.5.3(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“5.5.3(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“5.5.3(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“5.5.3(5) 捕捉&单次计数模式的运行 (高电平宽度的测量)”)。

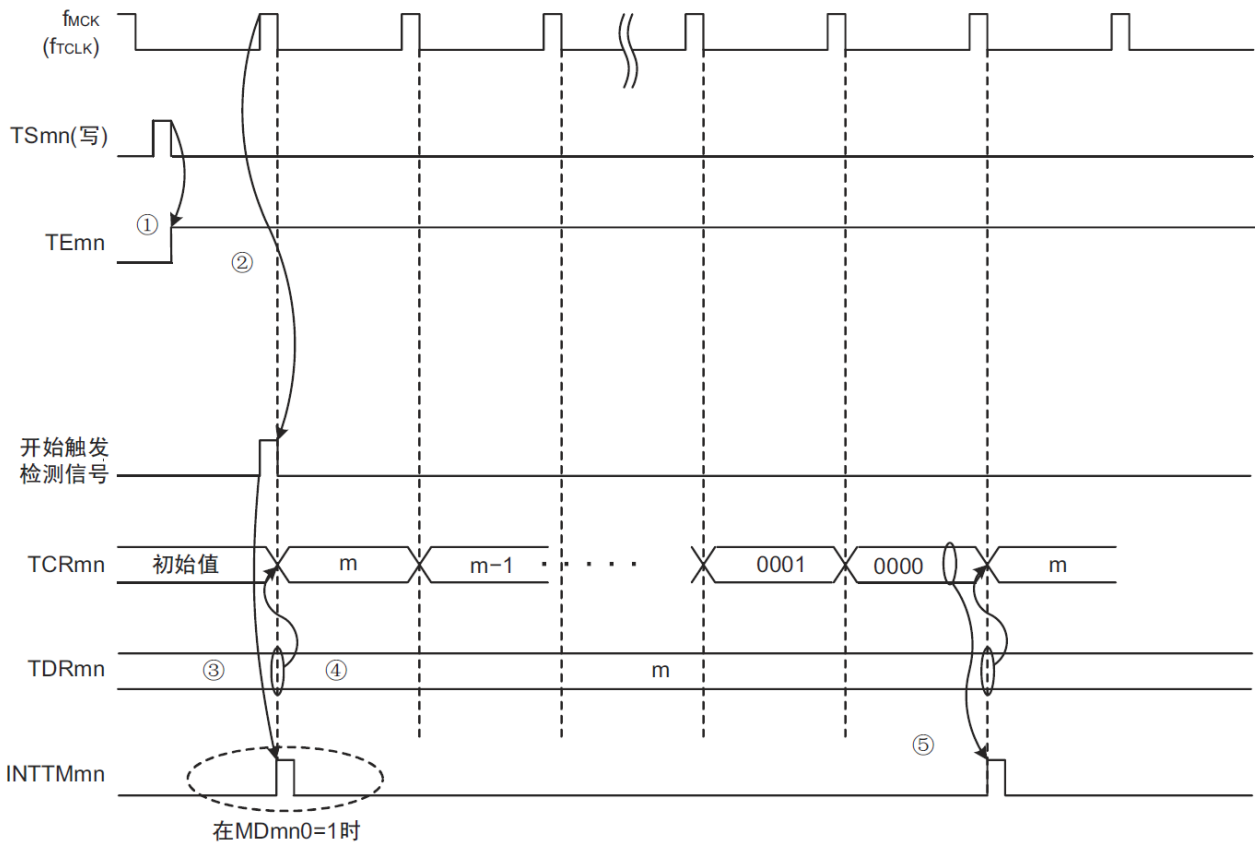
### 5.5.3 计数器的运行

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 (FMCK) 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 (FMCK) 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5-5：运行时序（间隔定时器模式）



#### 备注：

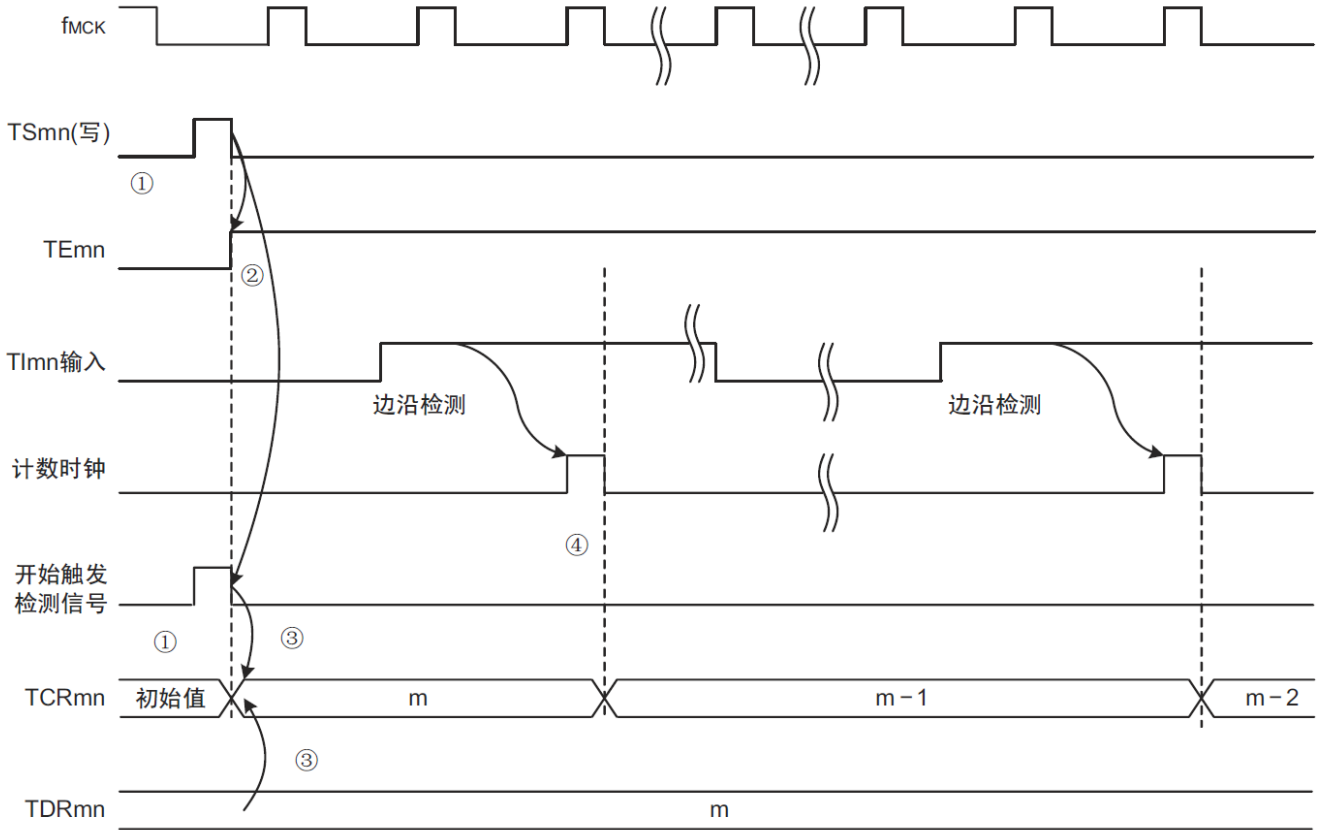
1. 因为第1个计数时钟周期的运行在写 TSmn 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MDmn0 位置“1”，以便能在开始计数时产生中断。
2.  $F_{MCK}$ 、开始触发检测信号和 INTTMmn 与  $F_{CLK}$  同步并且在1个时钟内有效。

(2) 事件计数器模式的运行

- ① 在运行停止状态 (TE<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。

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5-6: 运行时序 (事件计数器模式)

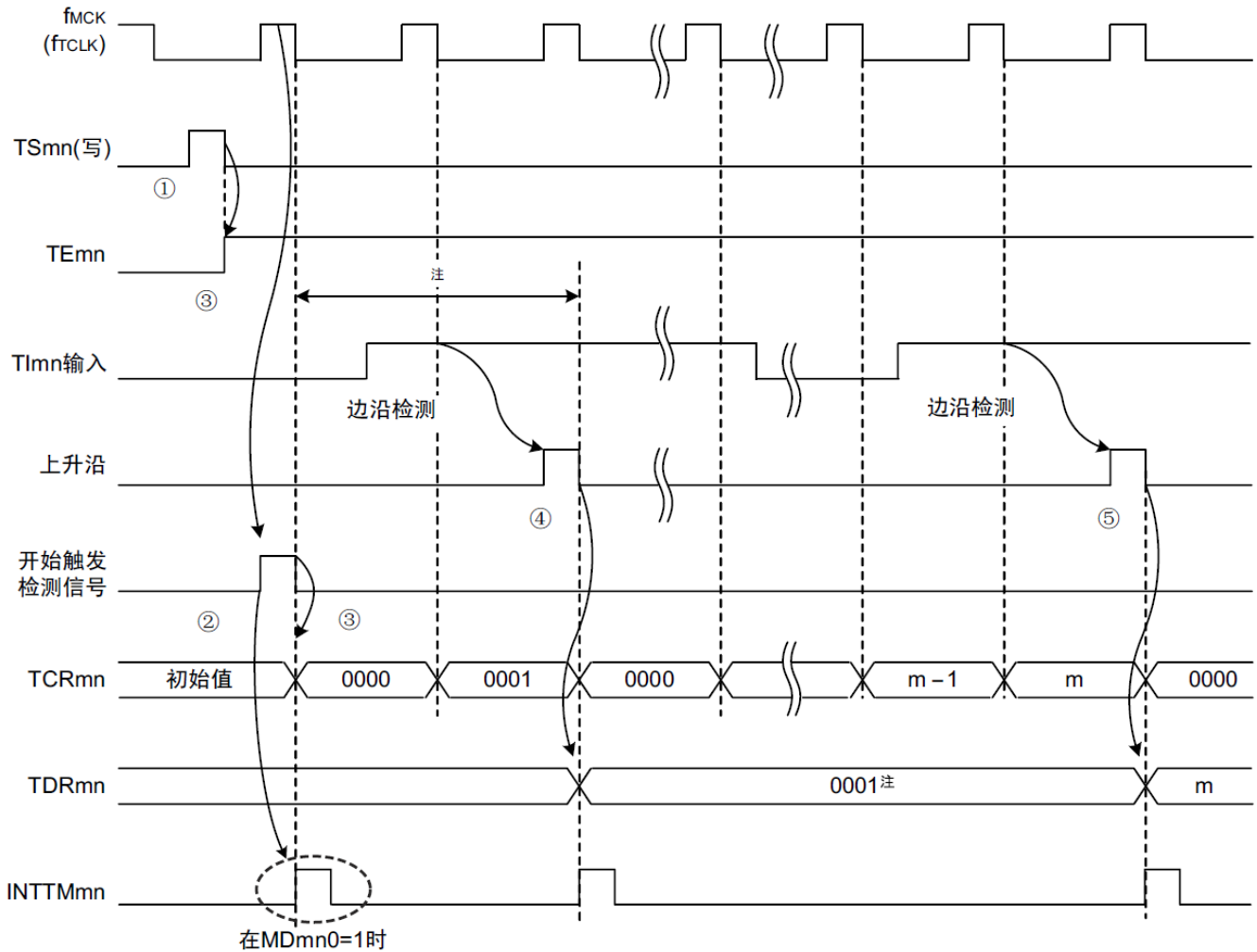


备注: 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从TI<sub>mn</sub>输入开始再延迟2个 F<sub>MCK</sub>周期 (合计3~4个周期)。1个周期的误差是因为TI<sub>mn</sub>输入与计数时钟 (F<sub>MCK</sub>) 不同步。

(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5-7：运行时序（捕捉模式：输入脉冲的间隔测量）



注1：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。

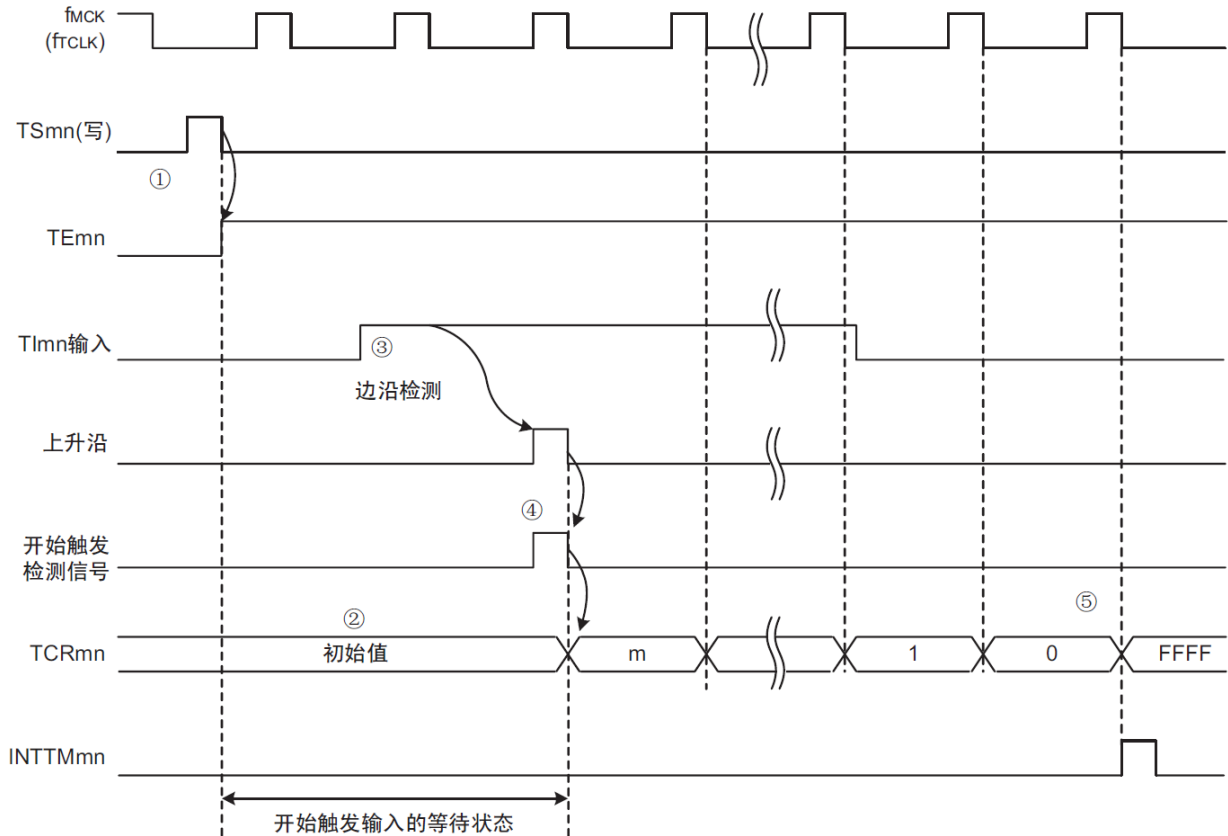
注2：因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

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“FFFFH”，停止计数。

图5-8: 运行时序 (单次计数模式)

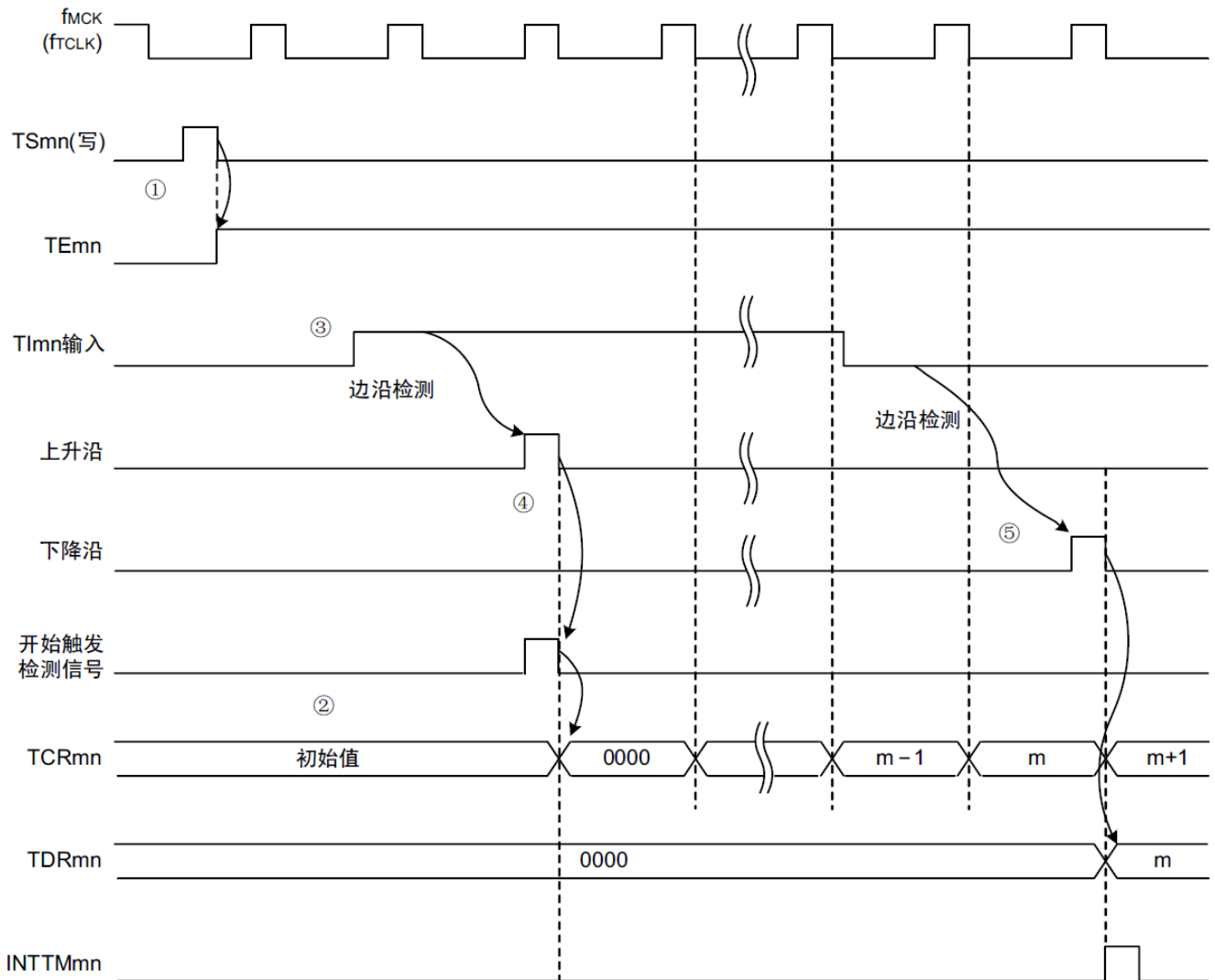


备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TImn输入开始再延迟2个F<sub>MCK</sub>周期（合计3~4个周期）。1个周期的误差是因为TImn输入与计数时钟（F<sub>MCK</sub>）不同步。

(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5-9：运行时序（捕捉&单次计数模式：高电平宽度的测量）



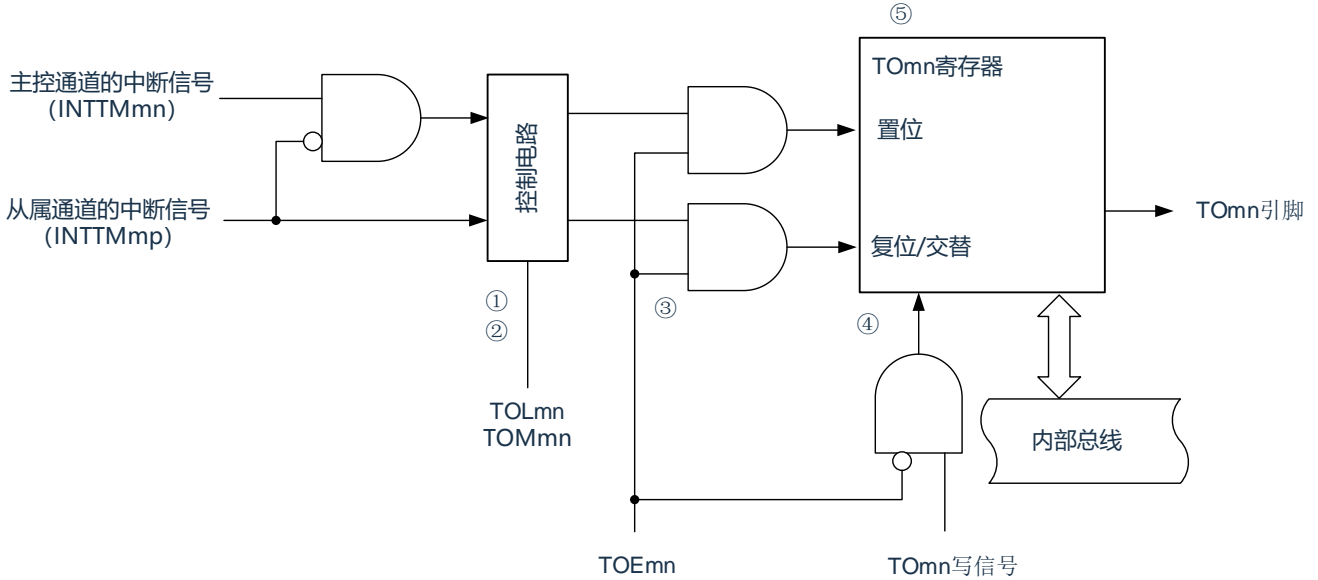
备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TImn输入开始再延迟2个F<sub>MCK</sub>周期（合计3~4个周期）。1个周期的误差是因为TImn输入和计数时钟（F<sub>MCK</sub>）不同步。



## 5.6 通道输出（TOmn引脚）的控制

### 5.6.1 TOmn引脚输出电路的结构

图5-10：输出电路的结构



以下说明 TOmn 引脚的输出电路。

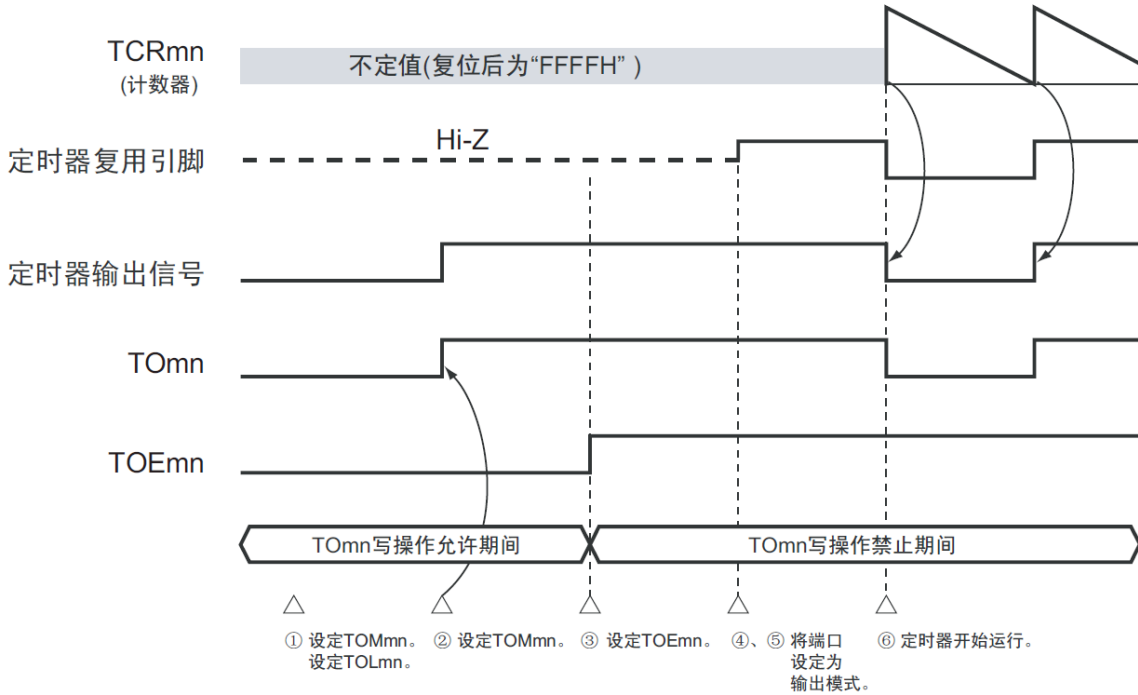
- ① 当 TOMmn 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 m（TOLm）的设定值，只将 INTTMmp（从属通道定时器中断）传给定时器输出寄存器 m（TOM）。
- ② 当 TOMmn 位为“1”（从属通道输出模式）时，将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。  
此时，TOLm 寄存器有效并且进行以下信号的控制：  
TOLmn=0 时：正相运行（INTTMmn→置位、INTTMmp→复位）  
TOLmn=1 时：反相运行（INTTMmn→复位、INTTMmp→置位）  
当同时产生 INTTMmn 和 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，需要在设定为禁止定时器输出（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, 1） 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）

## 5.6.2 T0mn引脚的输出设定

从 T0mn 输出引脚的初始设定到定时器开始运行的步骤和状态变化如下所示。

图5-11：从设定定时器的输出到开始运行的状态变化



① 设定定时器输出的运行模式。

T0Mmn 位 (0: 主控通道输出模式、1: 从属通道输出模式)

T0Lmn 位 (0: 正逻辑输出、1: 负逻辑输出)

② 通过设定定时器输出寄存器 m (T0M)，将定时器输出信号设定为初始状态。

给 TOEmn 位写“1”，允许定时器输出 (禁止写 T0M 寄存器)。

③ 通过端口模式控制寄存器 (PMCxx) 将端口设定为数字输入/输出。

④ 将端口的输入/输出设定为输出。

⑤ 允许定时器运行 (TSmn=1)。

备注：m：单元号 (m=0, 1) n：通道号 (n=0~3)

### 5.6.3 通道输出运行的注意事项

(1) 有关定时器运行中的  $TOm$ 、 $TOEm$ 、 $TOLm$ 、 $TOMm$  寄存器的设定值变更

定时器的运行（定时器计数寄存器  $mn$  ( $TCRmn$ ) 和定时器数据寄存器  $mn$  ( $TDRmn$ ) 的运行) 和  $TOmn$  输出电路相互独立。因此，定时器输出寄存器  $m$  ( $TOm$ )、定时器输出允许寄存器  $m$  ( $TOEm$ ) 和定时器输出电平寄存器  $m$  ( $TOLm$ ) 的设定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定值。但是，为了在各定时器的运行中从  $TOmn$  引脚输出期待的波形，必须设定为 5.8 和 5.9 所示的各运行的寄存器设定内容例子的值。

如果在产生各通道的定时器中断 ( $INTTMmn$ ) 信号前后更改除了  $TOm$  寄存器以外的  $TOEm$  寄存器和  $TOLm$  寄存器的设定值，就根据是在产生定时器中断 ( $INTTMmn$ ) 信号前更改还是在产生后更改， $TOmn$  引脚输出的波形可能不同。

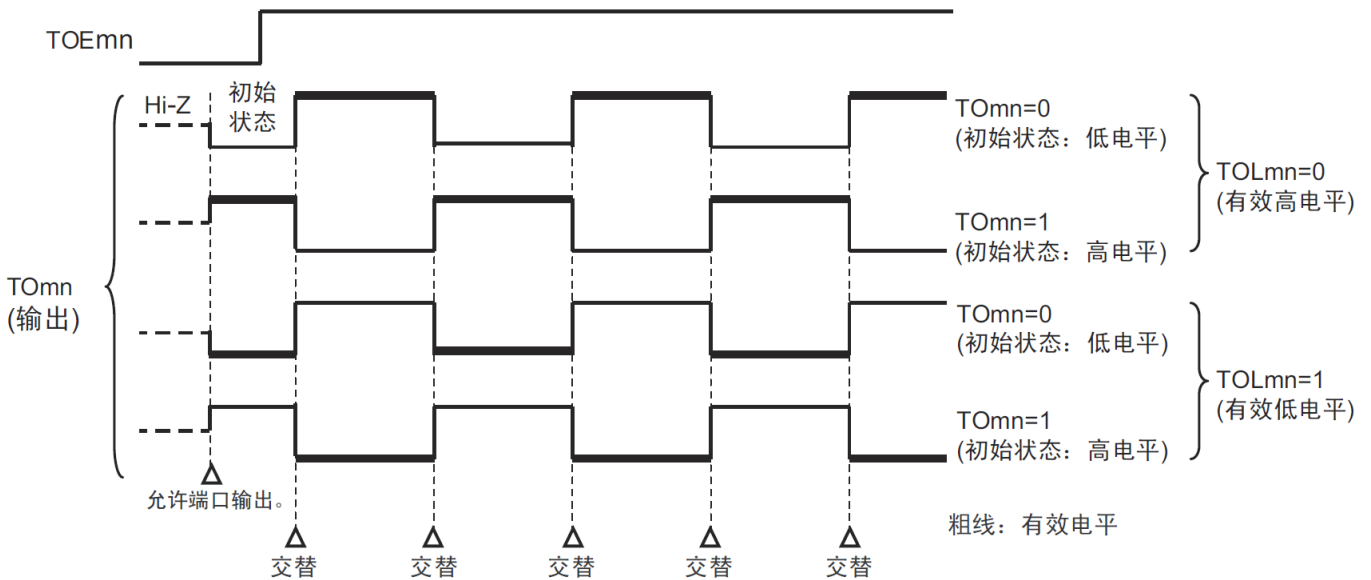
(2) 有关  $TOmn$  引脚的初始电平和定时器开始运行后的输出电平

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

① 在主导通道输出模式 ( $TOMmn=0$ ) 中开始运行的情况

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

图5-12: 交替输出时 ( $TOMmn=0$ ) 的  $TOmn$  引脚输出状态



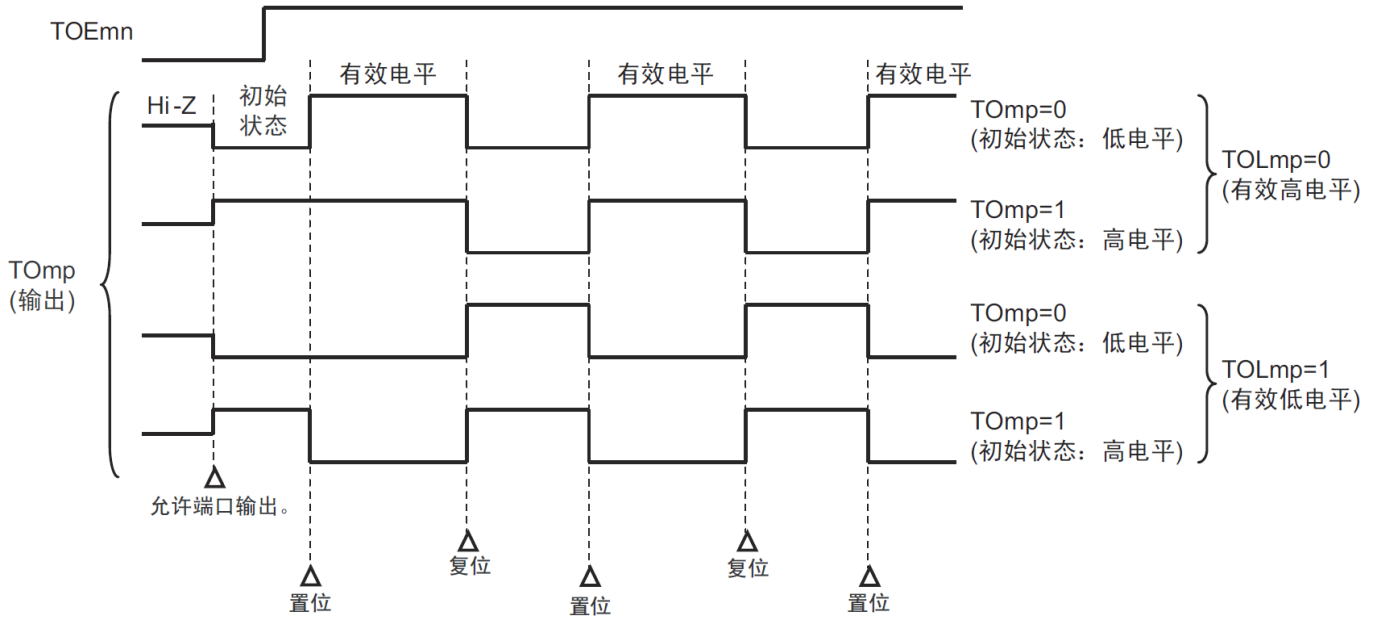
备注:

1. 交替: 反相  $TOmn$  引脚的输出状态。
2.  $m$ : 单元号 ( $m=0, 1$ )  $n$ : 通道号 ( $n=0\sim 3$ )

② 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) 的设定。

图5-13: 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。
2. 复位: TOmp引脚的输出信号从有效电平变为无效电平。
3. m: 单元号 (m=0, 1) 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)

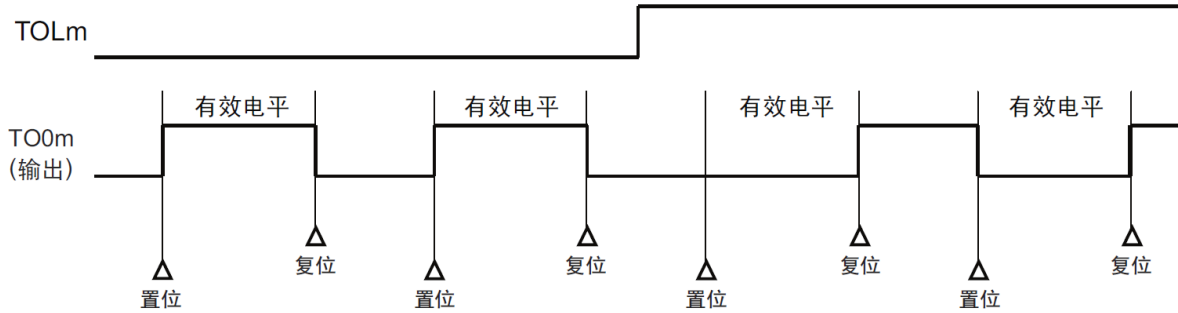
(3) 有关从属通道输出模式 (TOMmn=1) 的 TOMn 引脚变化

① 在定时器运行中更改定时器输出电平寄存器 m (TOLm) 的设定的情况

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

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

图5-14: 在定时器运行中更改TOLm寄存器的内容时的运行



② 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。

按照以下方法设定主控/从属通道时的置位/复位运行状态如图 5-15 所示。

主控通道: TOEmn=1、TOMmn=0、TOLmn=0

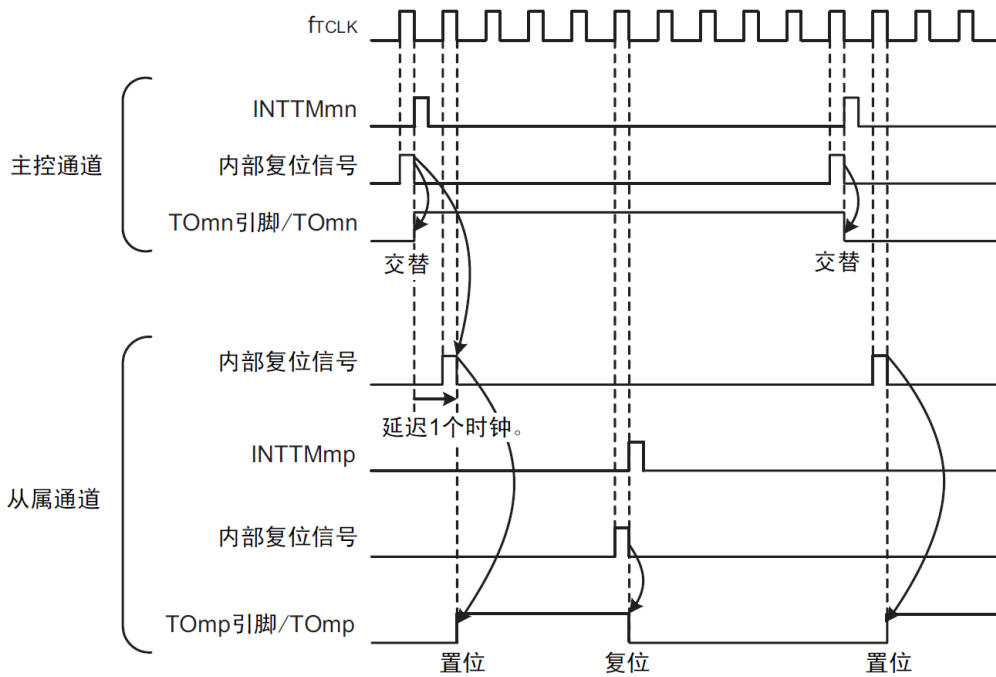
从属通道: TOEmp=1、TOMmp=1、TOLmp=0

备注:

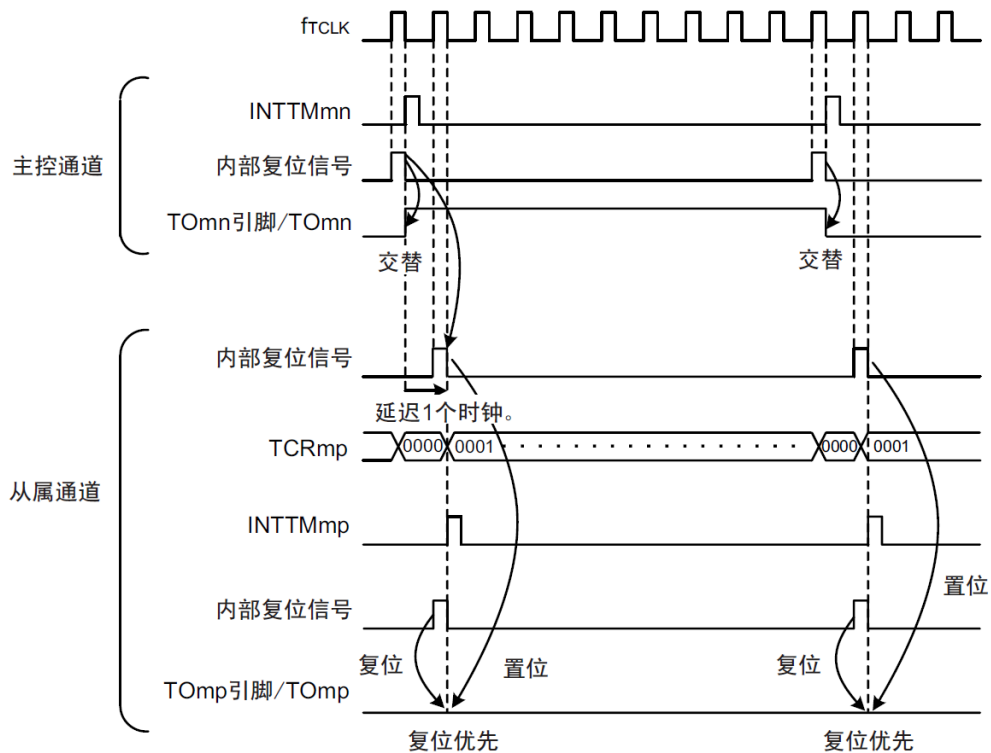
1. 置位: TOMn引脚的输出信号从无效电平变为有效电平。
2. 复位: TOMn引脚的输出信号从有效电平变为无效电平。
3. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-15: 置位/复位时序运行状态

(1) 基本运行时序



(2) 0%占空比的运行时序



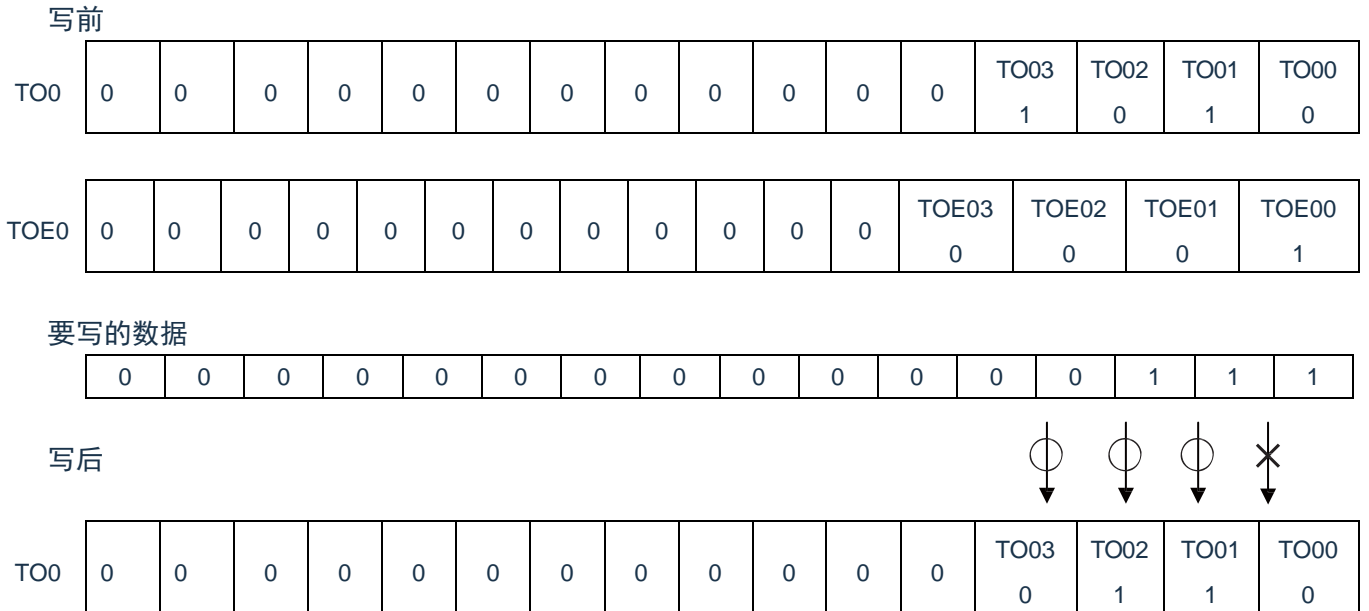
备注:

1. 内部复位信号:  $TOmn$ 引脚的复位/交替信号
2. 内部置位信号:  $TOmn$ 引脚的置位信号
3. m: 单元号 (m=0, 1) 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)

### 5.6.4 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) 有全部通道的设定位 (TOmn)，因此能一次性地操作全部通道的 TOmn 位。

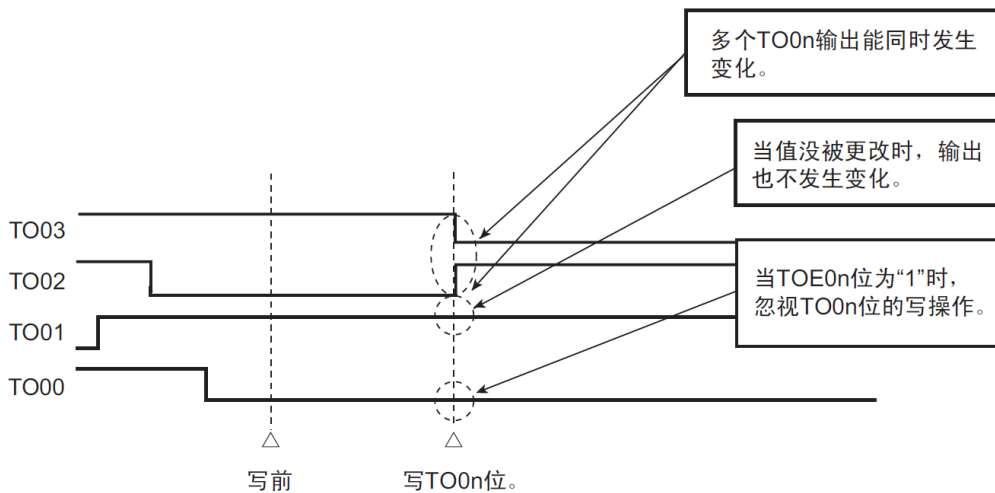
表5-23: TO0n位的一次性操作例子



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5-16: 一次性操作TO0n位时的TO0n引脚状态



备注: m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

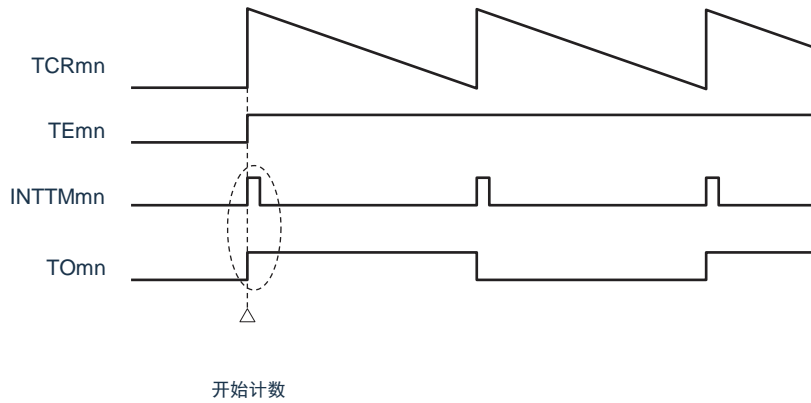
### 5.6.5 有关开始计数时的定时器中断和TOMn引脚输出

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

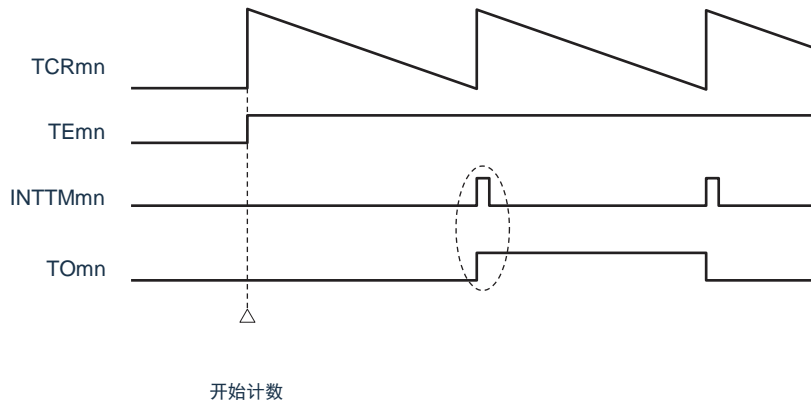
当 MDmn0 位为“1”时，能通过产生定时器中断 (INTTMmn) 得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和 TOMn 输出。设定为间隔定时器模式 (TOEmn=1、TOMmn=0) 时的运行例子如下所示。

图5-17：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, 1) n：通道号 (n=0~3)

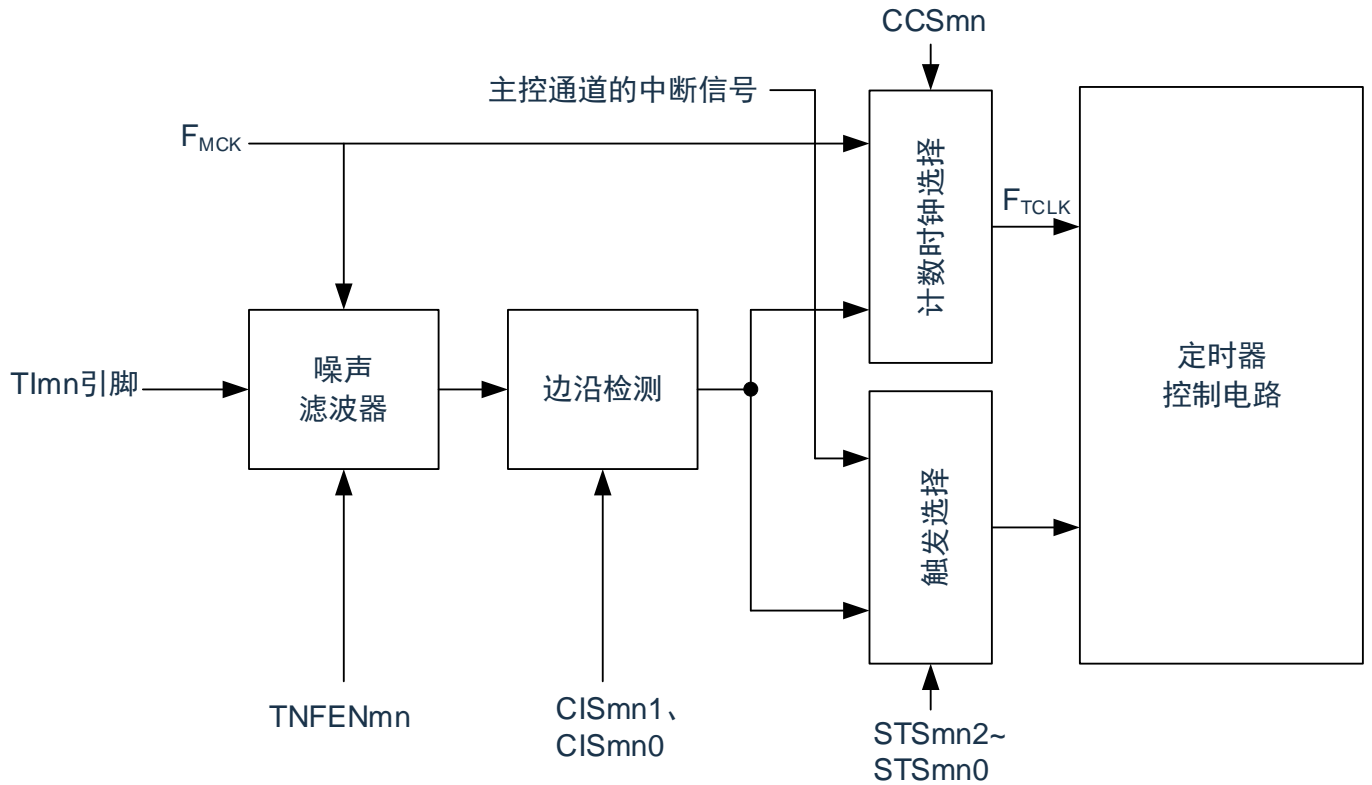


## 5.7 定时器输入 (TImn) 的控制

### 5.7.1 TImn引脚输入电路的结构

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

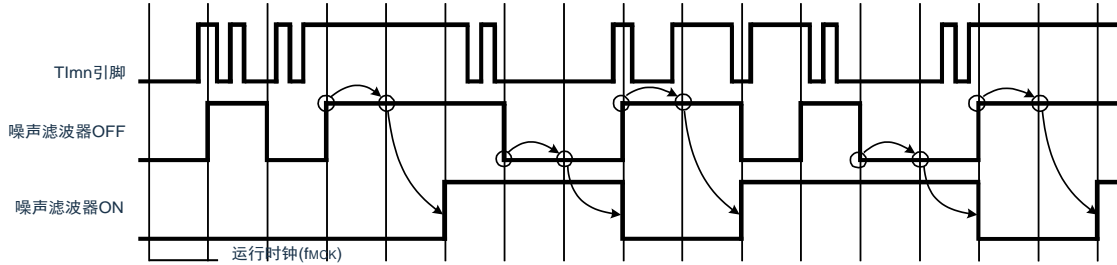
图5-18: 输入电路的结构



### 5.7.2 噪声滤波器

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

图5-19：TImn输入引脚在噪声滤波器ON或者OFF情况下的采样波形



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

### 5.7.3 操作通道输入时的注意事项

在设定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，从设定为使用定时器输入引脚到设定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。

## 5.8 通用定时器单元的独立通道运行功能

### 5.8.1 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：

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

#### (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：

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

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

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5-20: 作为间隔定时器/方波输出运行的基本时序例子 (MDmn0=1)

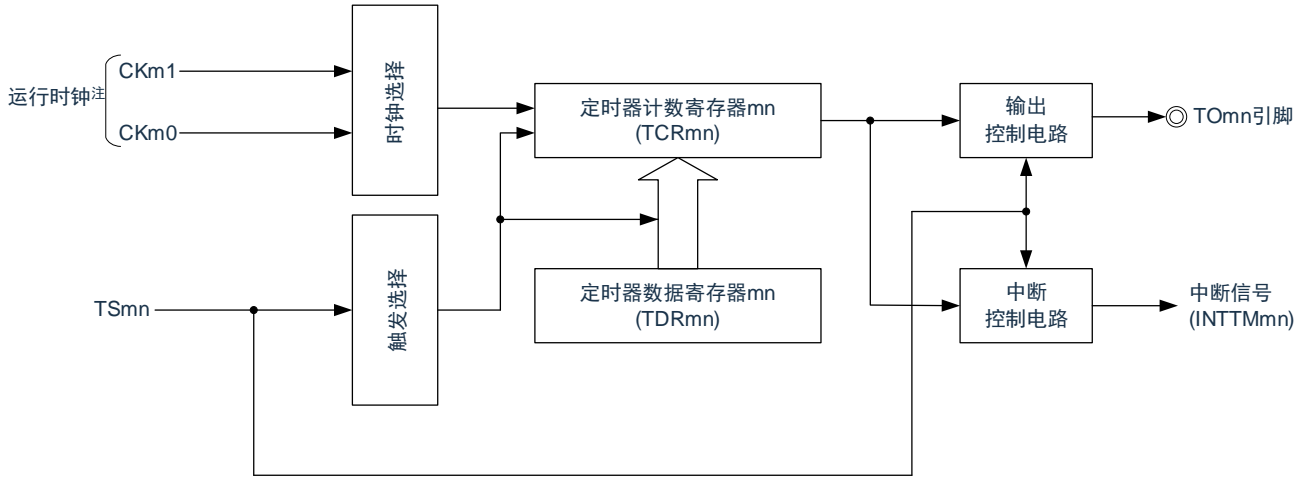
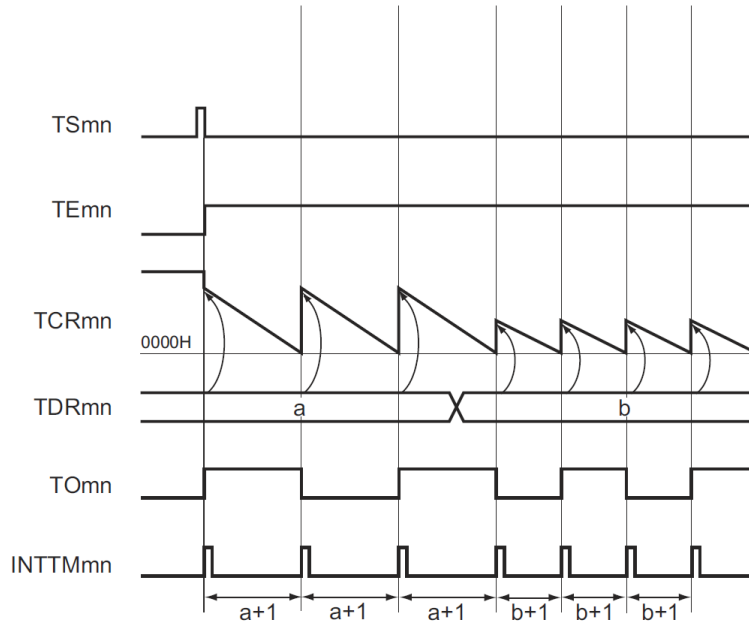


图5-21: 作为间隔定时器/方波输出运行的基本时序例子 (MDmn0=1)



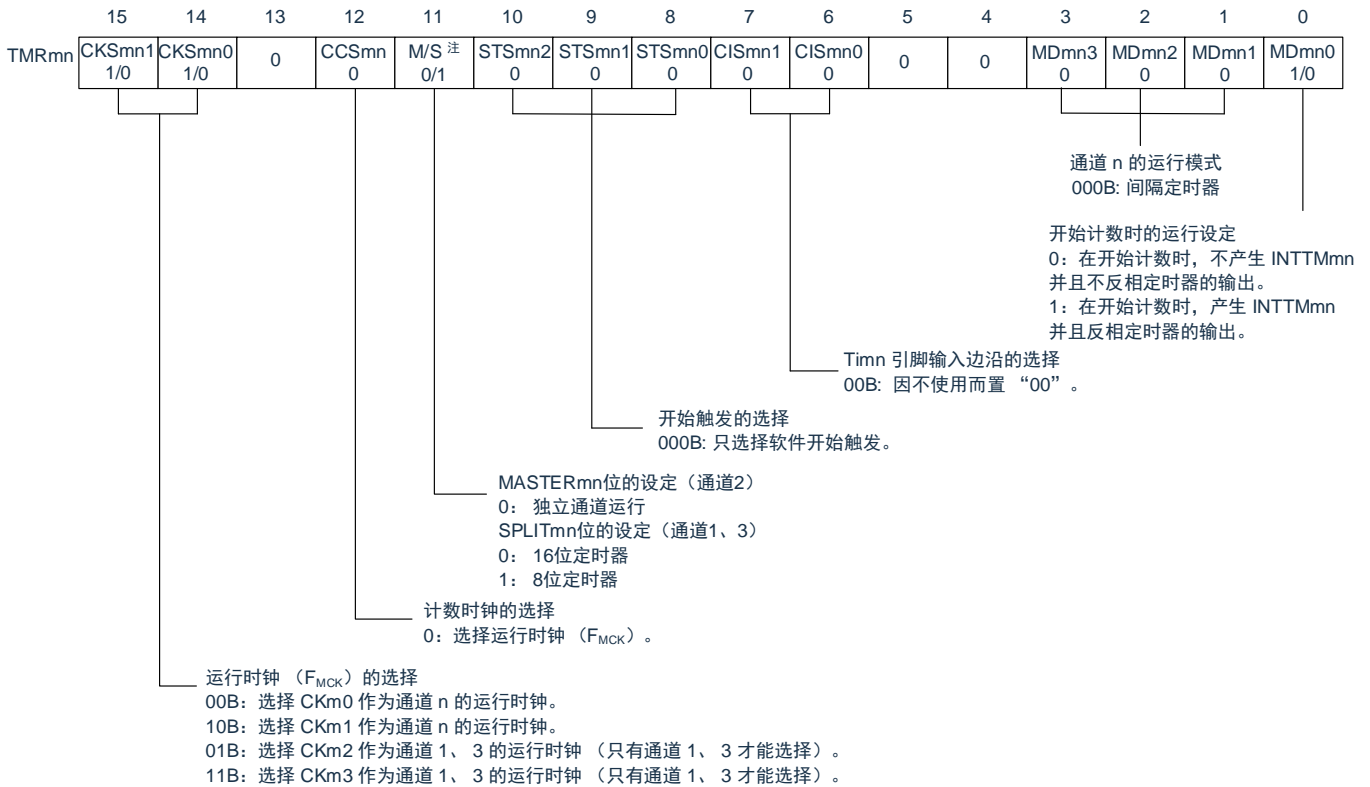
注: 在通道1和通道3时, 能从CKm0、CKm1、CKm2和CKm3中选择时钟。

备注:

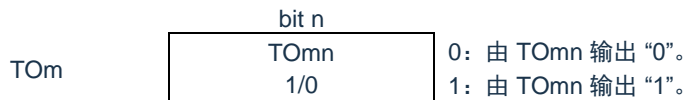
1. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)
2. TSmn: 定时器通道开始寄存器m (TSm) 的bit n  
 TEMn: 定时器通道允许状态寄存器m (TEm) 的bit n  
 TCRmn: 定时器计数寄存器mn (TCRmn)  
 TDRmn: 定时器数据寄存器mn (TDRmn)  
 TOMn: TOMn引脚输出信号

图5-22: 间隔定时器/ 方波输出时的寄存器设定内容例子

(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)



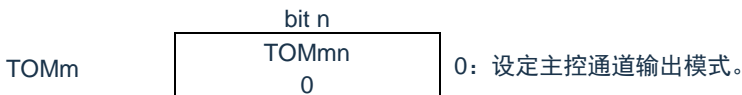
(c) 定时器输出允许寄存器 m (TOEm)



(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

备注:

1. TMRm1、TMRm3: SPLITmn位
2. TMRm0: 固定为“0”。
3. m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~ 3)

表5-24: 间隔定时器/ 方波输出功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定	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。 (开始提供时钟, 能写各寄存器)
通道初 始设定	设定定时器模式寄存器 mn (TMRmn) (确定通道的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	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”。 设定 TOmn 位, 确定 TOmn 输出的初始电平。	TOmn 引脚处于 Hi-Z 输出状态。
	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。
	将端口寄存器和端口模式寄存器置“0”。	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。
开始 运行	(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
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。 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。
TAU 停止	要保持 TOmn 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 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)

备注: m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~ 3)

## 5.8.2 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：

$$\text{规定的计数值} = \text{TDRmn 的设定值} + 1$$

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。

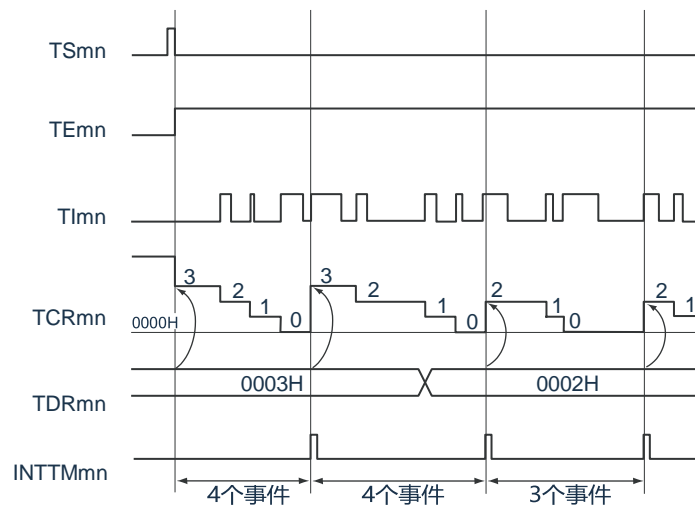
TCRmn 寄存器在检测到 TImn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入 TDRmn 寄存器的值并且输出 INTTMmn。

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5-23：作为外部事件计数器运行的基本时序例子

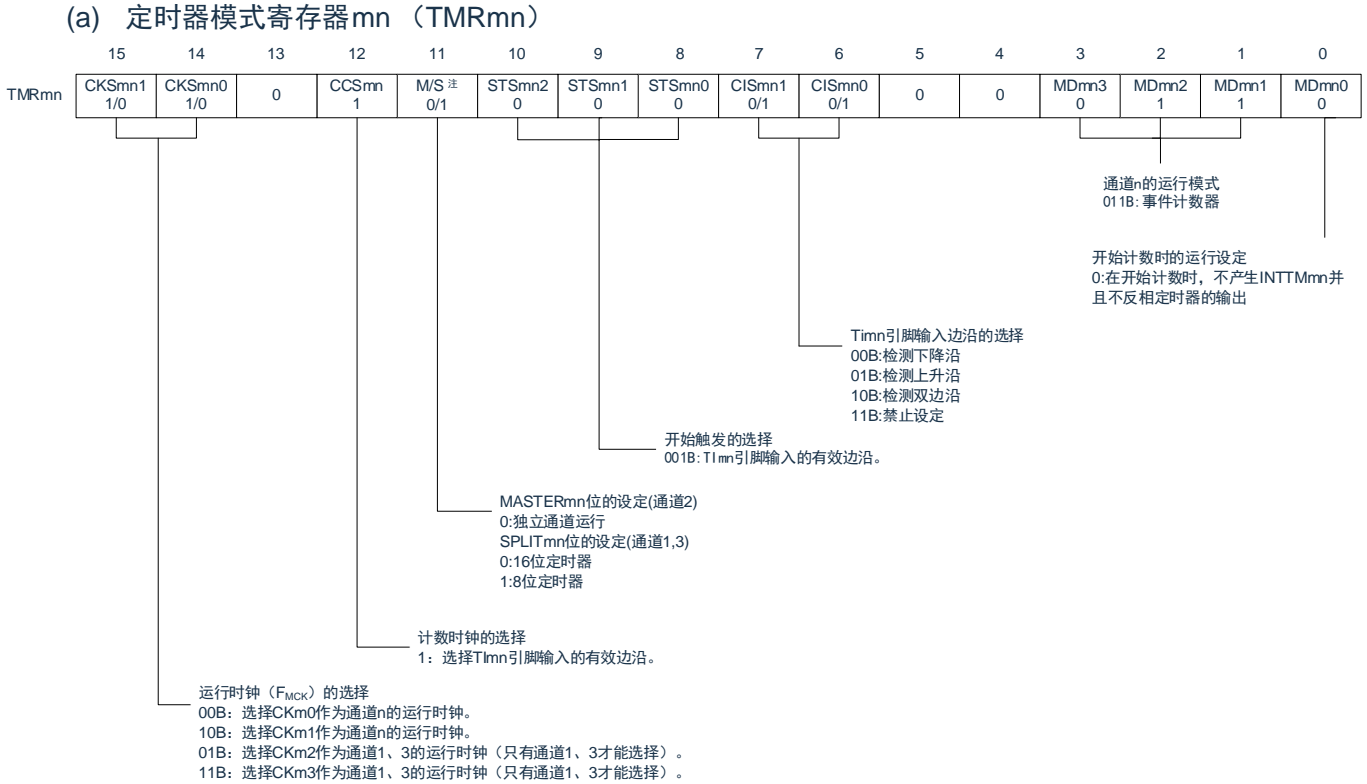


备注：

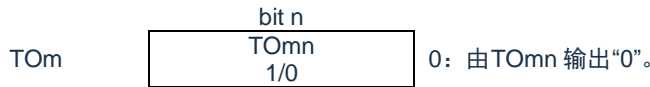
1. m：单元号（m=0, 1）n：通道号（n=0~3）
2. TSMn：定时器通道开始寄存器m（TSM）的bit n  
 TEMn：定时器通道允许状态寄存器m（TEM）的bit n  
 TImn：TImn 引脚输入信号  
 TCRmn：定时器计数寄存器 mn（TCRmn）  
 TDRmn：定时器数据寄存器 mn（TDRmn）



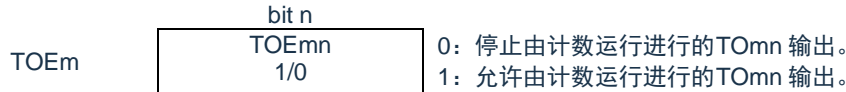
图5-24：外部事件计数器模式时的寄存器设定内容例子



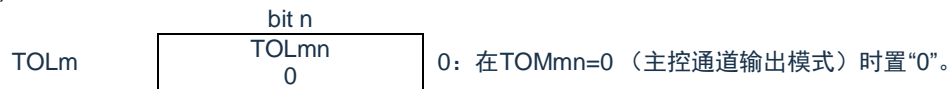
(b) 定时器输出寄存器m (TOM)



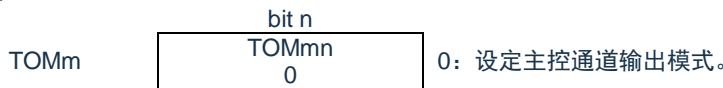
(c) 定时器输出允许寄存器m (TOEm)



(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

备注:

1. TMRm1、TMRm3: SPLITmn位
2. TMRm0: 固定为“0”。
3. m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~ 3)

表5-25: 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"。  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 (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)
重新 开始 运行 → 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。
	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 INTTmn。 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 TA4mEN 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。

### 5.8.3 作为分频器的运行

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 的设定值+1) × 2}
- 选择双边沿的情况：  
分频时钟频率≈输入时钟频率/ (TDR00 的设定值+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。

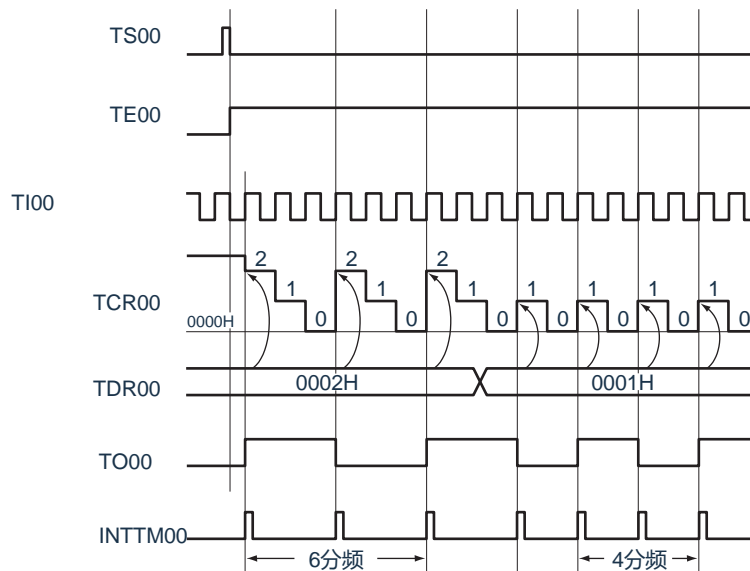
如果选择 TI00 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

$$\text{TO00 输出的时钟周期} = \text{理想的 TO00 输出时钟周期} \pm \text{运行时钟周期 (误差)}$$

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。

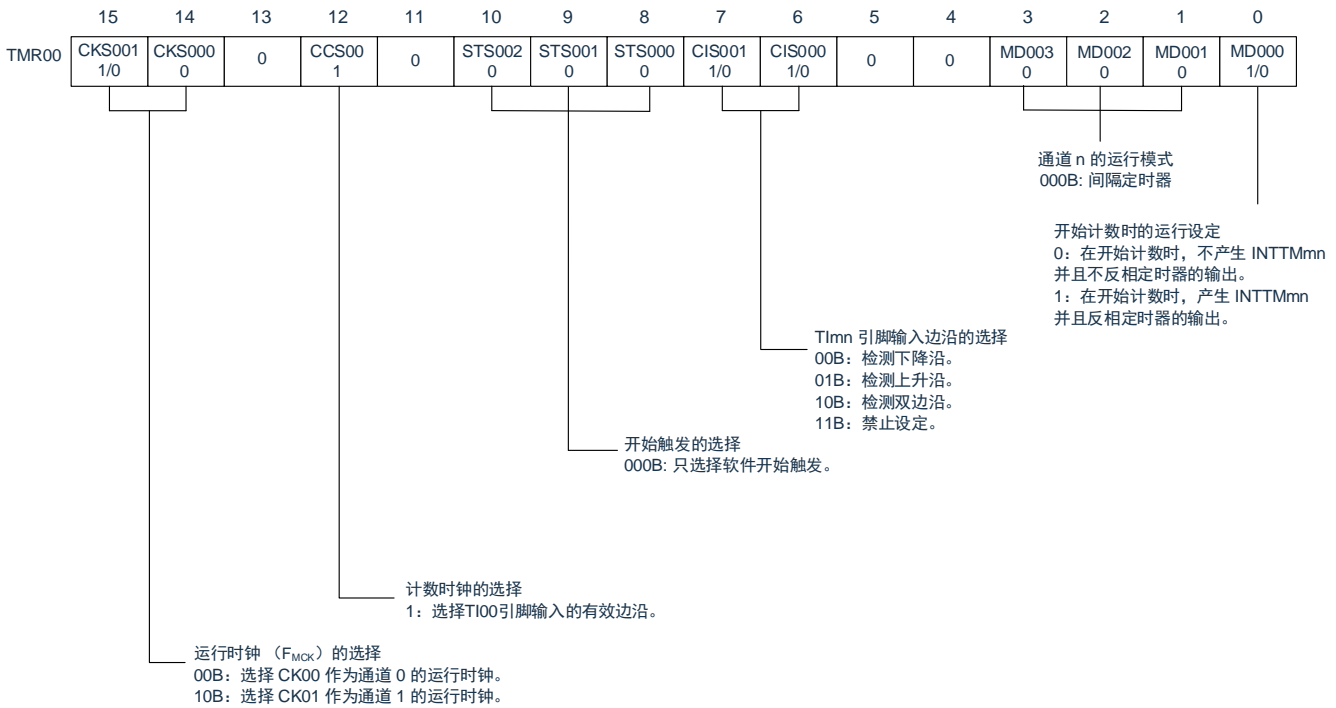
图5-25：作为分频器运行的基本时序例子（MD000=1）



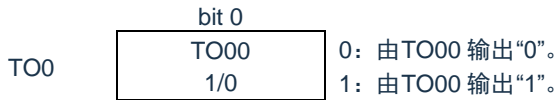
- 备注：TS00：定时器通道开始寄存器0（TS0）的bit0  
 TE00：定时器通道允许状态寄存器0（TE0）的bit0  
 TI00：TI00 引脚输入信号  
 TCR00：定时器计数寄存器00（TCR00）  
 TDR00：定时器数据寄存器00（TDR00）  
 TO00：TO00 引脚输出信号

图5-26：作为分频器运行时的寄存器设定内容例子

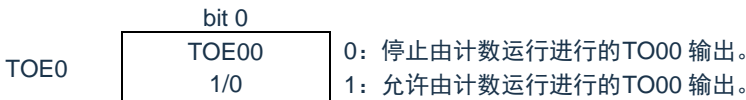
(a) 定时器模式寄存器00 (TMR00)



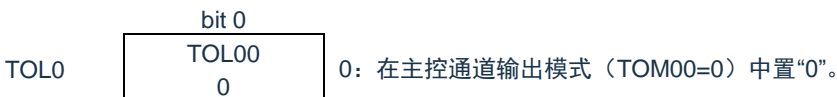
(b) 定时器输出寄存器0 (TO0)



(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 (TOL0)



(e) 定时器输出模式寄存器0 (TOM0)



表5-26: 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”。	TO00 引脚处于 Hi-Z 输出状态。
	设定 TO00 位并且确定 TO00 输出的初始电平。————> 将 TOE00 位置 “1”, 允许 TO00 输出。————> 将端口寄存器和端口模式寄存器置 “0”————>	当端口模式寄存器为输出模式并且端口寄存器为 “0” 时, 输 出 TO00 初始设定的电平。 因为通道处于运行停止状态, 所以 TO00 不变。 TO00 引脚输出 TO00 设定的电平。
重新 开始 运行 →	开始运行	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”。
	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 。
	停止运行	将 TT00 位置 “1”。————> 因为 TT00 位是触发位, 所以自动返回到 “0”。
Timer4 停止	将 TOE00 位置 “0” 并且给 TO00 位设定值。————>	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。 TO00 引脚输出 TO00 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”。————> 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”。————>	通过端口功能保持 TO00 引脚的输出电平。 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)

### 5.8.4 作为输入脉冲间隔测量的运行

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

脉冲间隔能用以下计算式进行计算：

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

在捕捉模式中，定时器计数寄存器mn (TCRmn) 用作递增计数器。

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

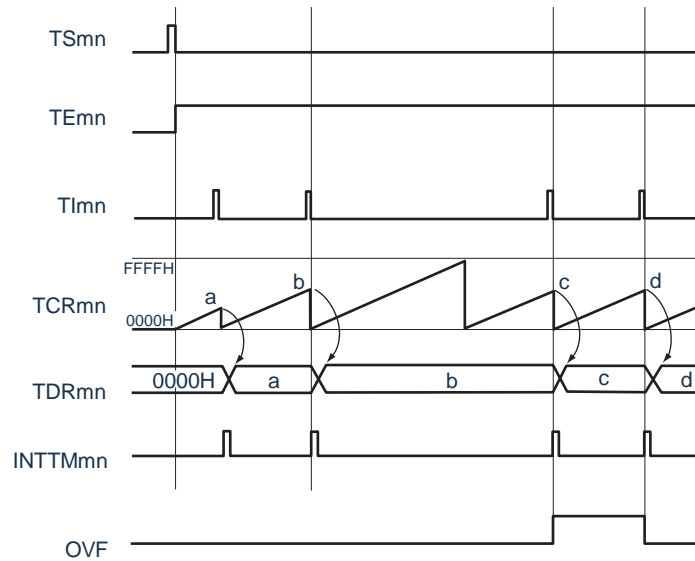
如果检测到 TImn 引脚输入的有效边沿，就将 TCRmn 寄存器的计数值传送 (捕捉) 到定时器数据寄存器 mn (TDRmn)，同时将 TCRmn 寄存器清“0000H”，然后输出 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器mn (TSRmn) 的OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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

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

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

图5-27：作为输入脉冲间隔测量的运行基本时序例子 (MDmn0=0)

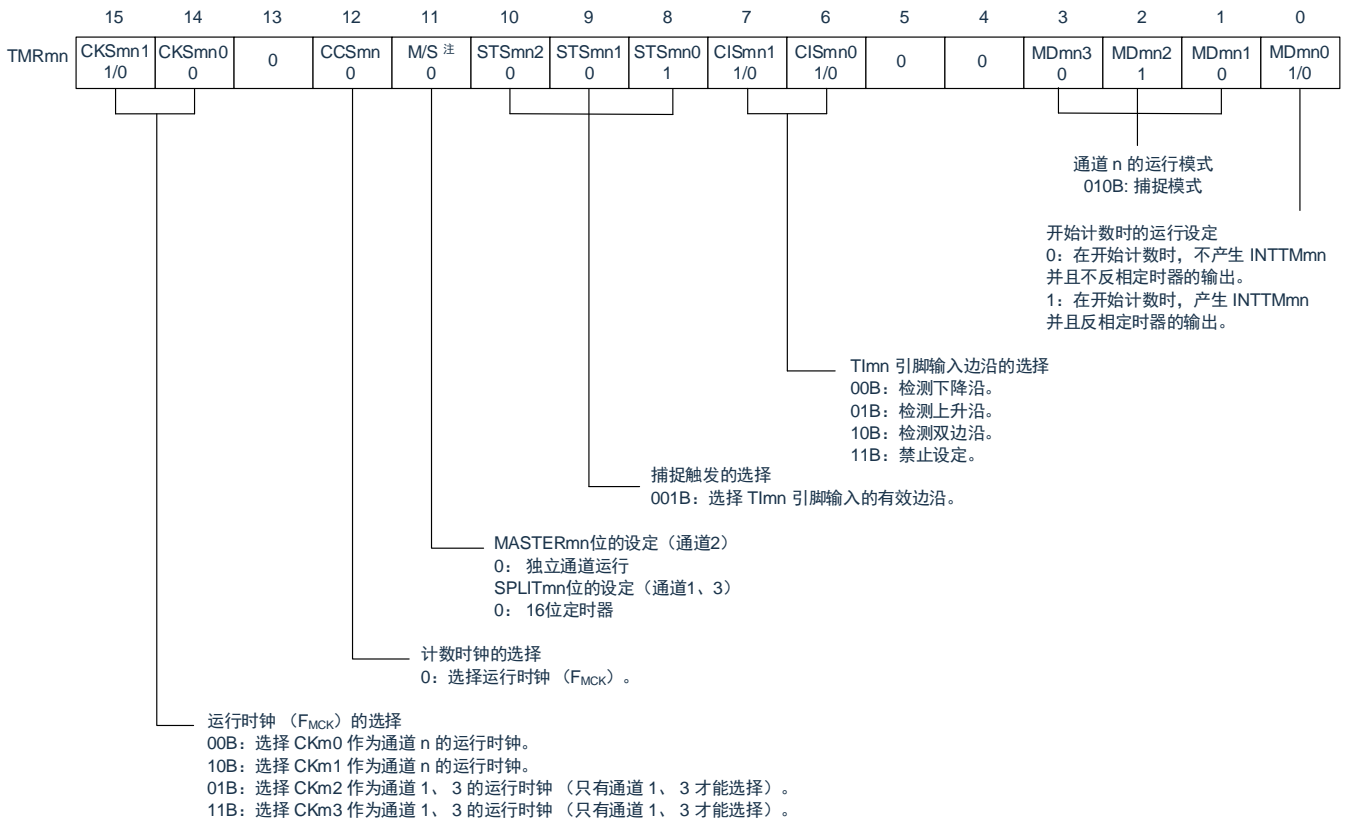


备注：

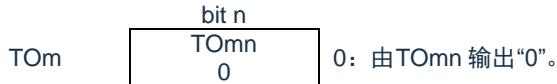
1. 因为通过定时器模式寄存器mn (TMRmn) 的CKSmn 位选择的运行时钟对TImn 引脚输入进行采样，所以产生1 个运行时钟的误差。
2. m：单元号 (m= 0) n：通道号 (n=0 ~3)
3. TSmn：定时器通道开始寄存器m (TSM) 的bit n  
 TEMn：定时器通道允许状态寄存器m (TEM) 的bit n  
 TImn：TImn引脚输入信号  
 TCRmn：定时器计数寄存器mn (TCRmn)  
 TDRmn：定时器数据寄存器mn (TDRmn)  
 OVF：定时器状态寄存器mn (TSRmn) 的bit0

图5-28：测量输入脉冲间隔时的寄存器设定内容例子

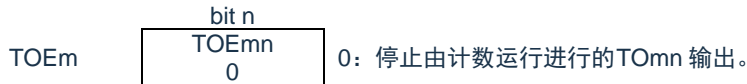
(a) 定时器模式寄存器 mn (TMRmn)



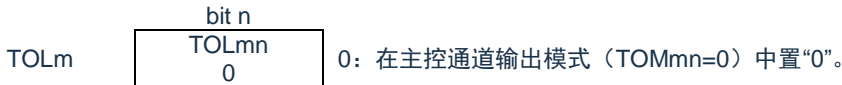
(b) 定时器输出寄存器 m (TOM)



(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

备注:

1. TMRm1、TMRm3: SPLITmn 位
2. TMRm0: 固定为“0”。
3. m: 单元号 (m=0) n: 通道号 (n=0~3)

表5-27：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
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, 1) n: 通道号 (n=0~3)



## 5.8.5 作为输入信号高低电平宽度测量的运行

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

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

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（TSMn）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 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置位。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 位来设定是测量 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”。

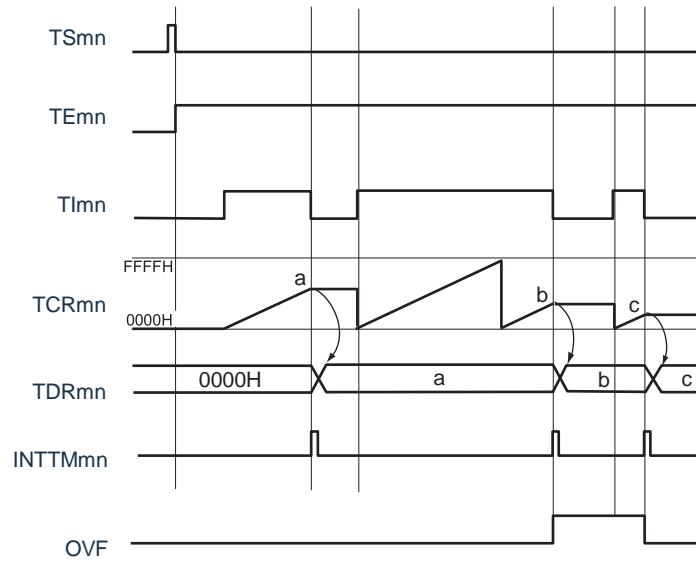
TMRmn 寄存器的 CISmn1、CISmn0=10B：测量低电平宽度。

TMRmn 寄存器的 CISmn1、CISmn0=11B：测量高电平宽度。

注意：

1. 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。
2. 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。

图5-29: 作为输入信号高低电平宽度测量的运行基本时序例子

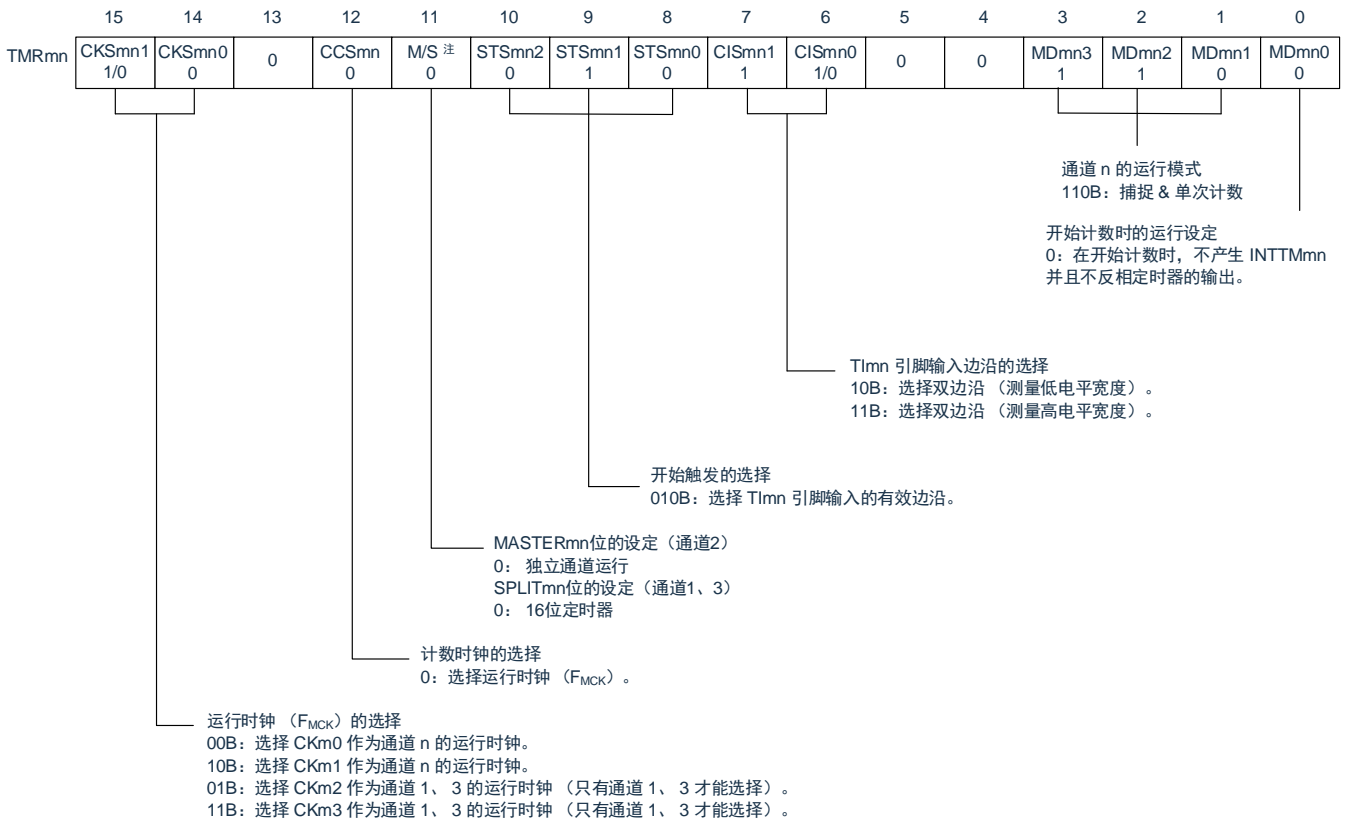


备注:

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  
TEmn: 定时器通道允许状态寄存器m (TEm) 的bit n  
TImn: TImn引脚输入信号  
TCRmn: 定时器计数寄存器mn (TCRmn)  
TDRmn: 定时器数据寄存器mn (TDRmn)  
OVF: 定时器状态寄存器mn (TSRmn) 的bit0

图5-30：测量输入信号的高低电平宽度时的寄存器设定内容例子

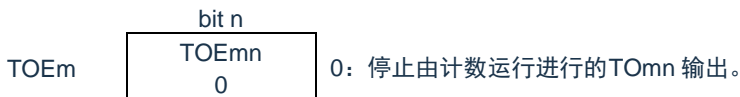
(a) 定时器模式寄存器mn (TMRmn)



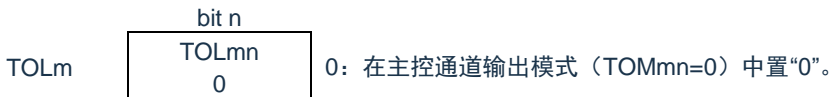
(b) 定时器输出寄存器 m (TOM)



(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

备注:

1. TMRm1、TMRm3: SPLITmn位
2. TMRm0: 固定为“0”。
3. m: 单元号 (m=0) n: 通道号 (n=0~3)

表5-28：输入信号高低电平宽度测量功能时的操作步骤

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

重新  
开始  
运行

备注: m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~3)

### 5.8.6 作为延迟计数器的运行

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

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

中断产生周期能用以下计算式进行计算：

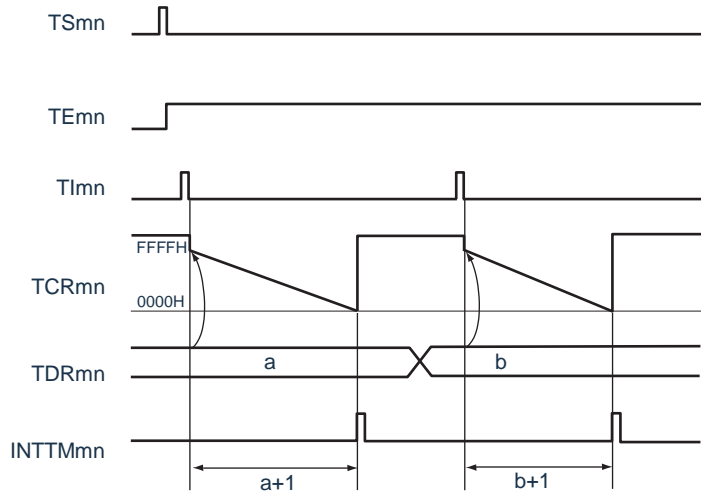
$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDRmn 的设定值} + 1)$$

在单次计数模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

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

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

图5-31：作为延迟计数器的运行基本时序例子

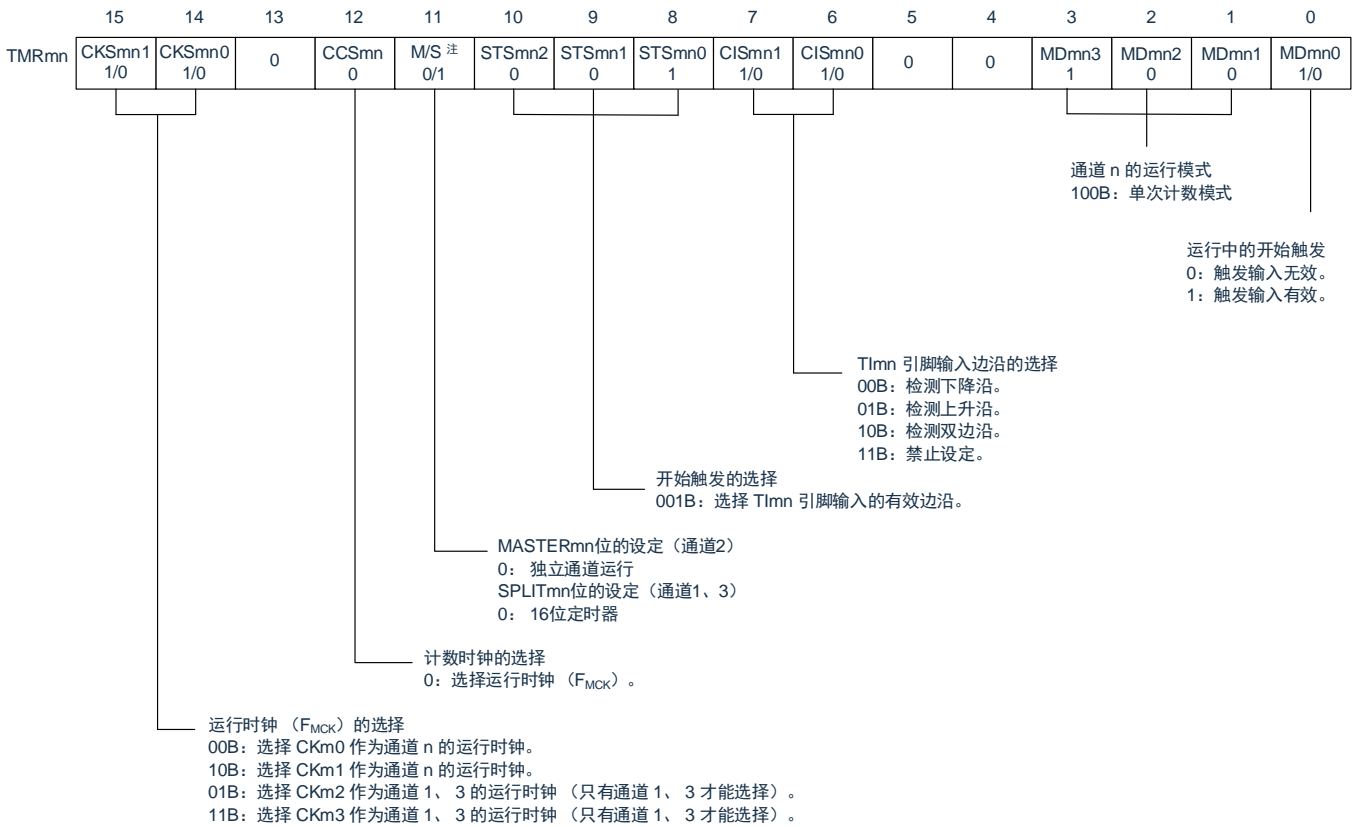


备注：

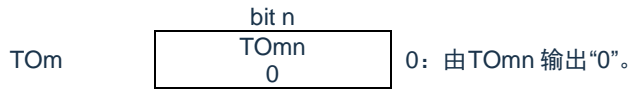
1. m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~ 3)
2. TSmn: 定时器通道开始寄存器m (TSm) 的 bit n  
 TEmn: 定时器通道允许状态寄存器m (TEm) 的 bit n  
 TImn: TImn引脚输入信号  
 TCRmn: 定时器计数寄存器mn (TCRmn)  
 TDRmn: 定时器数据寄存器mn (TDRmn)

图5-32：延迟计数器功能时的寄存器设定内容例子

(a) 定时器模式寄存器 mn (TMRmn)



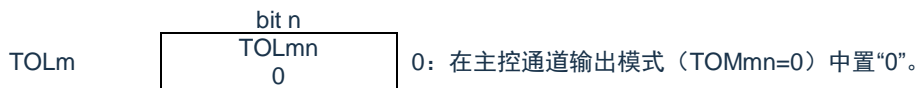
(b) 定时器输出寄存器 m (TOM)



(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

备注:

1. TMRm1、TMRm3: SPLITmn位
2. TMRm0: 固定为“0”。
3. m: 单元号 (m=0) n: 通道号 (n=0 ~3)

表5-29: 延迟计数器功能时的操作步骤

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

备注: m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~3)

## 5.9 通用定时器单元的多通道联动运行功能

### 5.9.1 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：

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

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。

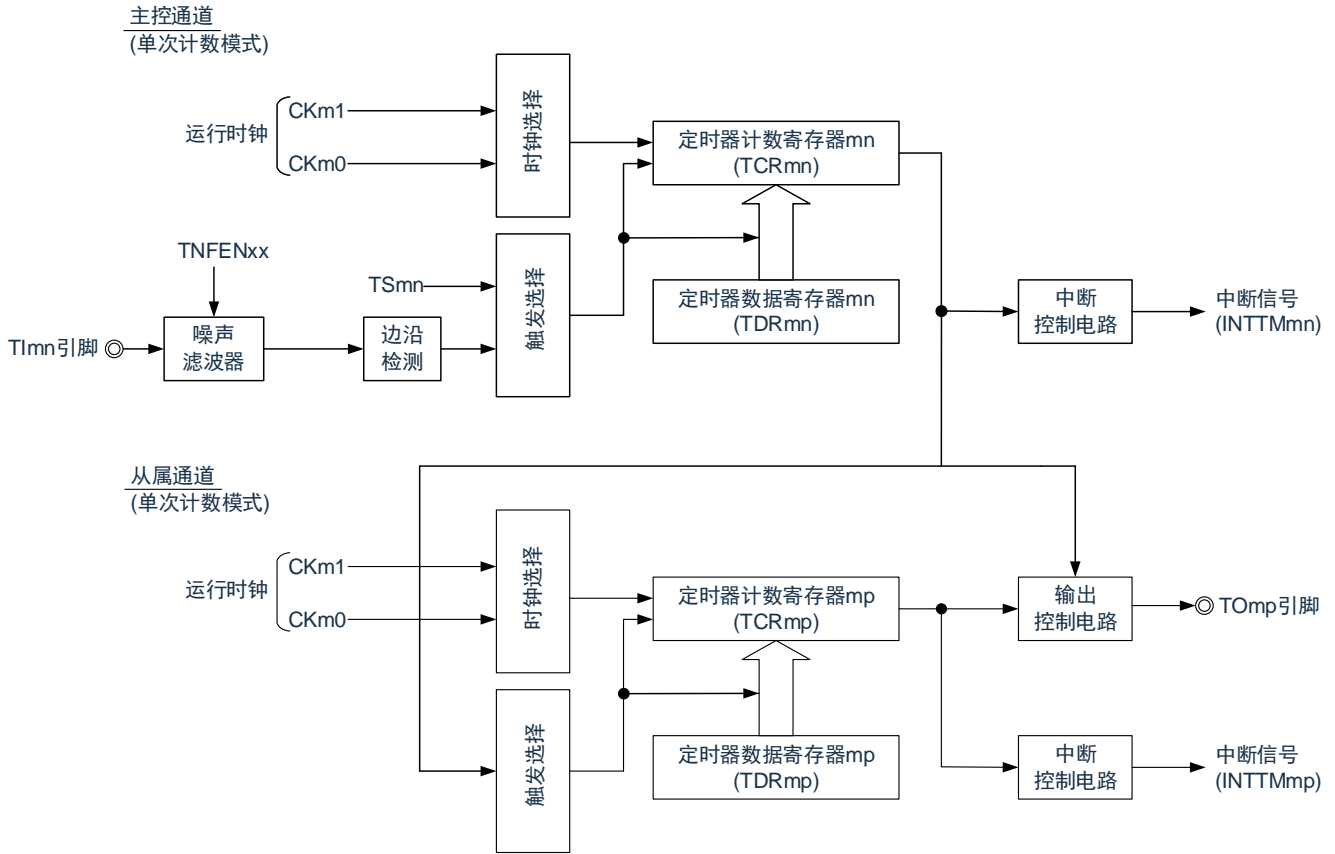
不使用TImn 引脚输入也能将软件操作（TSmn=1）作为开始触发来输出单触发脉冲。

**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, 1） 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）

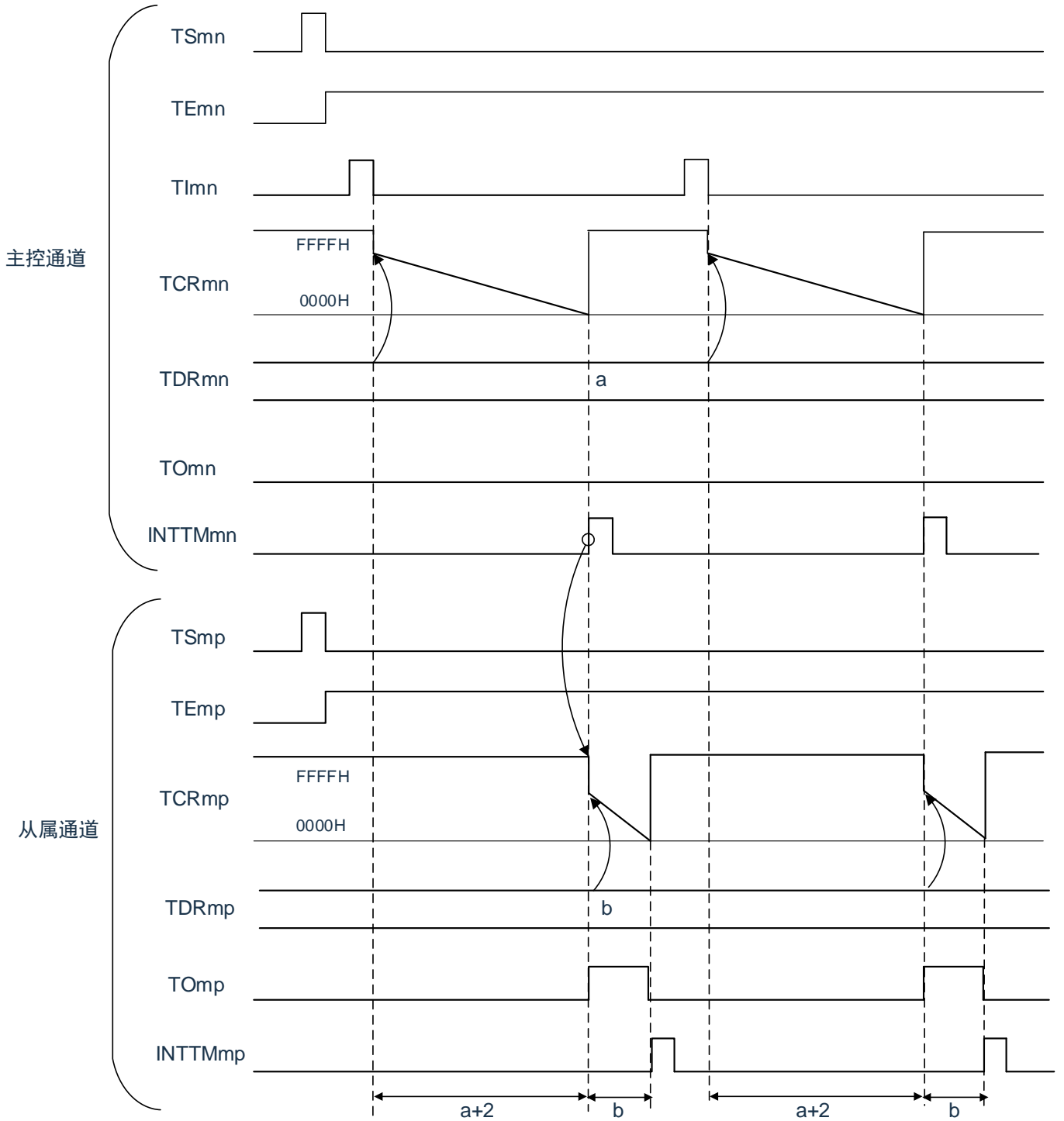


图5-33：作为单触发脉冲输出功能运行的框图



备注：m：单元号 (m=0, 1) 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5-34：作为单触发脉冲输出功能的运行基本时序例子

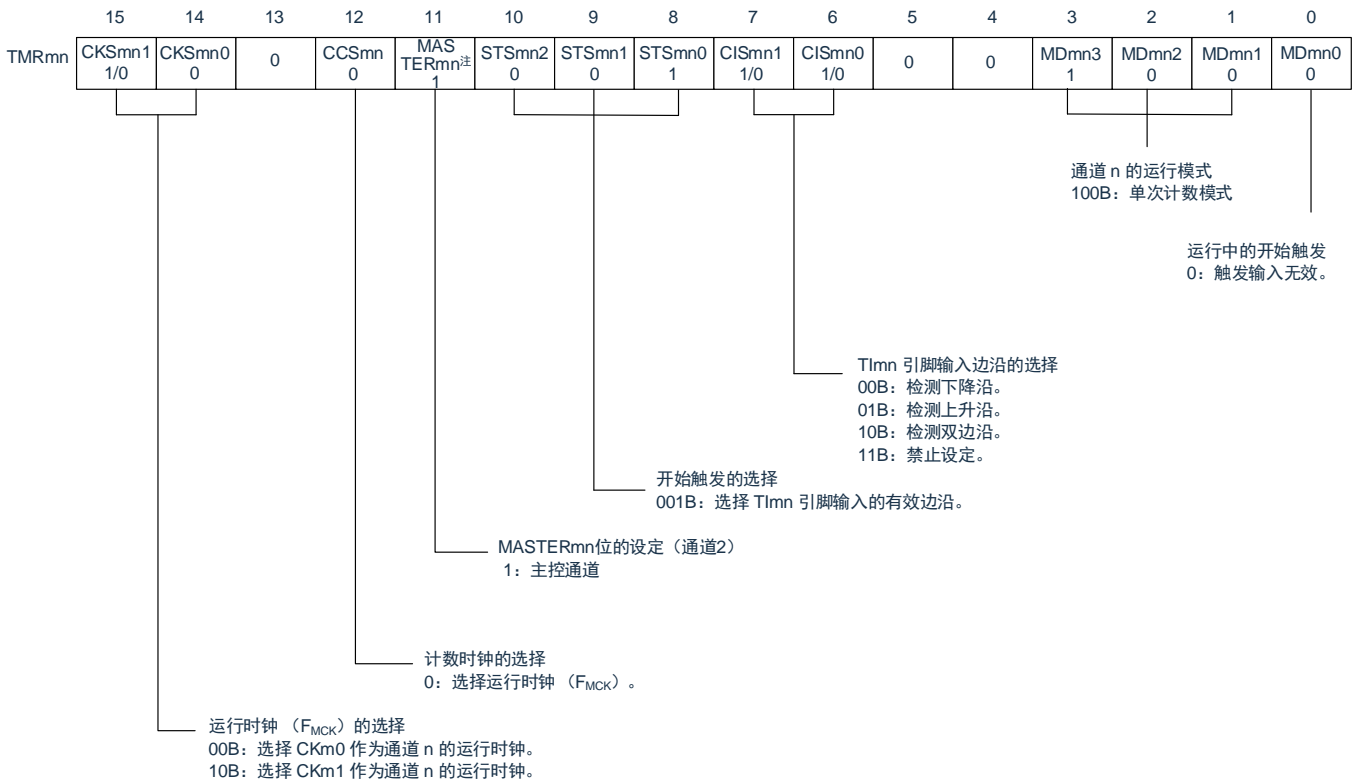


备注:

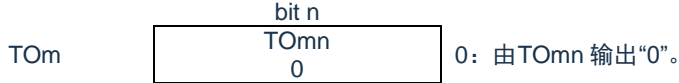
1. m: 单元号 (m=0, 1) n: 主控通道号 (n=0, 2) p: 从属通道号 (n=0: p=1, 2, 3, n=2: p=3)
2. TSmn、TSmp: 定时器通道开始寄存器m (TSM) 的bit n、p  
 TEmn、TEmp: 定时器通道允许状态寄存器m (TEM) 的bit n、p  
 TImn、TImp: TImn 引脚和TImp 引脚的输入信号  
 TCRmn、TCRmp: 定时器计数寄存器mn、mp (TCRmn、TCRmp)  
 TDRmn、TDRmp: 定时器数据寄存器mn、mp (TDRmn、TDRmp)  
 TOmn、TOmp: TOmn 引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

图5-35：单触发脉冲输出功能时（主控通道）的寄存器设定内容例子

(a) 定时器模式寄存器mn (TMRmn)



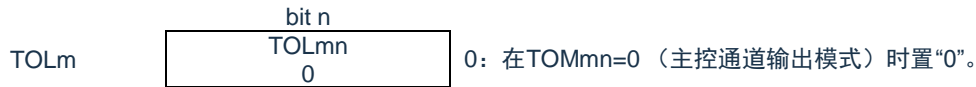
(b) 定时器输出寄存器 m (TOm)



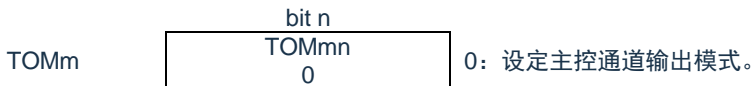
(c) 定时器输出允许寄存器m (TOEm)



(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)



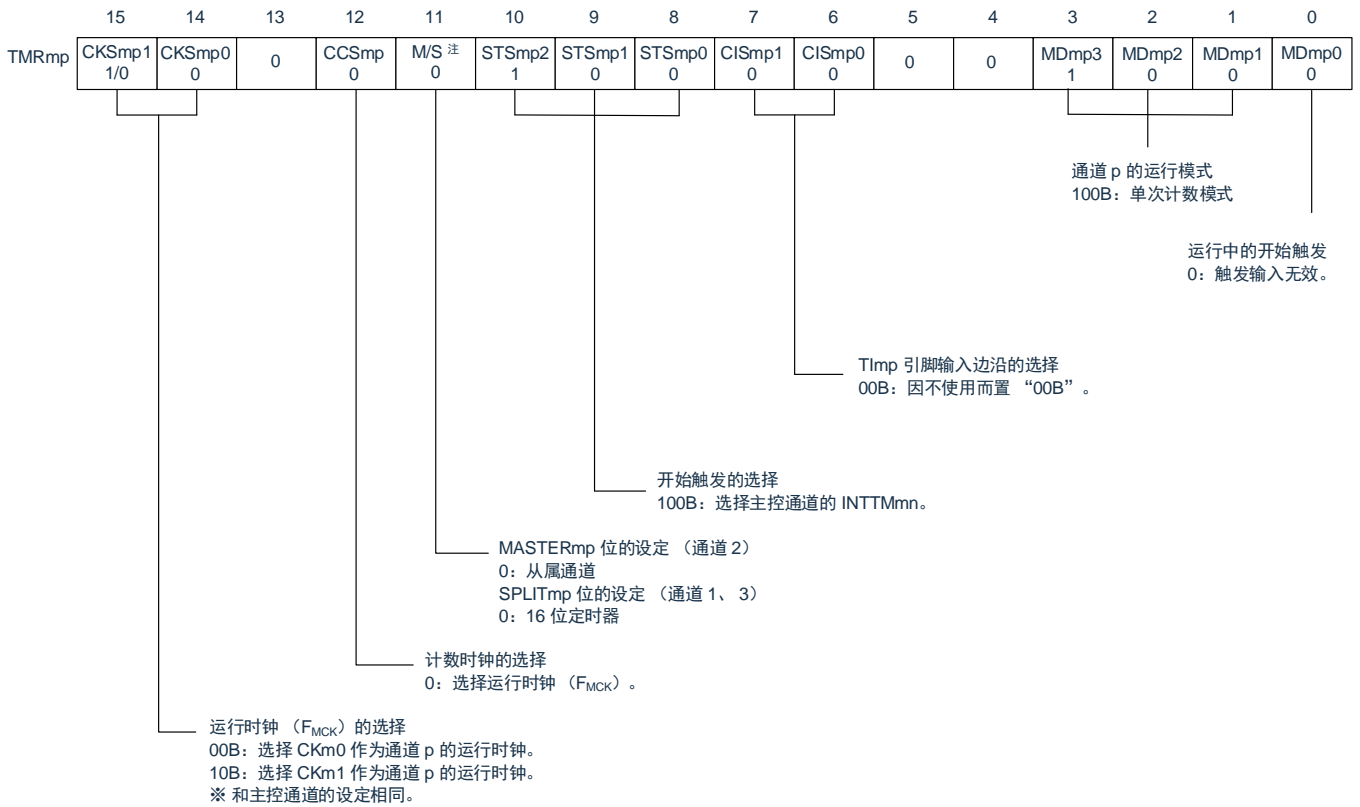
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

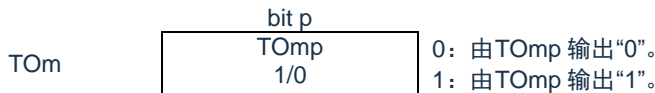
备注: m: 单元号 (m=0, 1) n: 主控通道号 (n=0、2)

图5-36: 单触发脉冲输出功能时(从属通道)的寄存器设定内容例子

(a) 定时器模式寄存器 mp (TMRmp)



(b) 定时器输出寄存器 m (TOM)



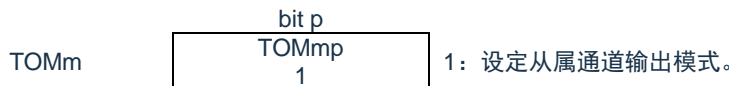
(c) 定时器输出允许寄存器 m (TOEm)



(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, 1) 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)

表5-30: 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。

表5-30: 单触发脉冲输出功能时的操作步骤(2/2)

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

重新开始运行

注: 不能将从属通道的TSmn位置“1”。

备注: m: 单元号 (m= 0, 1) 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)

## 5.9.2 作为 PWM 功能的运行

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

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

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) 的设定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。

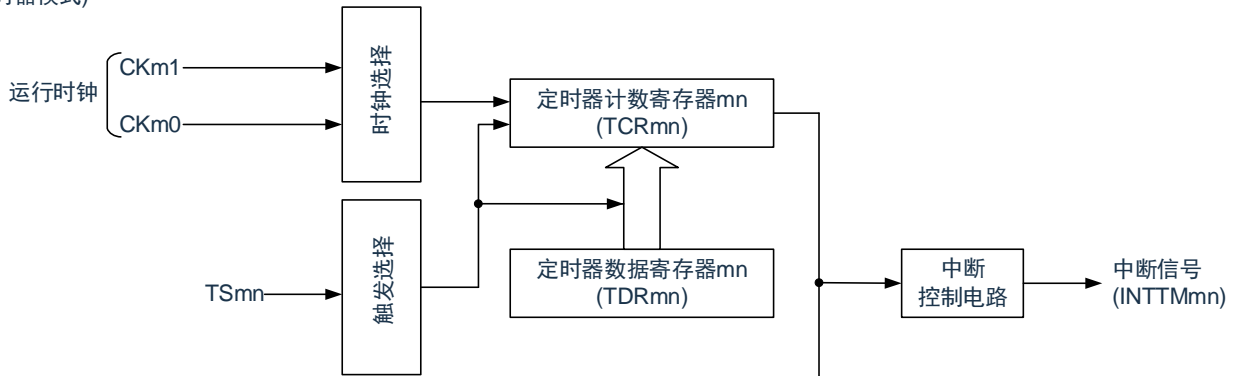
注意：

1. 当 TDRmp (从属) 的设定值  $> \{TDRmn \text{ (主控) 的设定值} + 1\}$  时，占空比超过 100%，但是为 100% 输出。
2. 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。

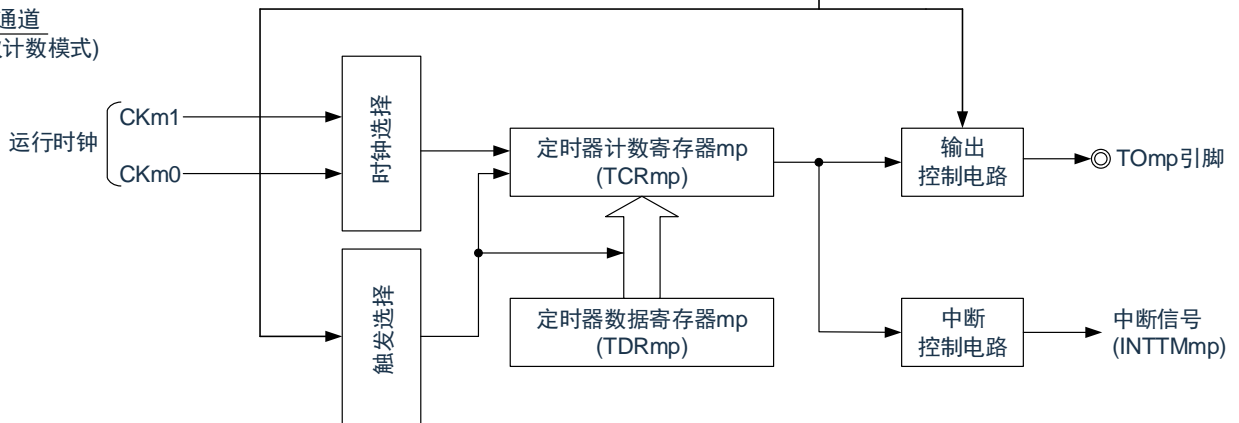
备注： $m$ ：单元号 ( $m=0, 1$ )  $n$ ：主控通道号 ( $n=0, 2$ )  $p$ ：从属通道号 ( $n=0$ :  $p=1, 2, 3$ ,  $n=2$ :  $p=3$ )

图5-37: 作为PWM 功能运行的框图

主控通道  
(间隔定时器模式)



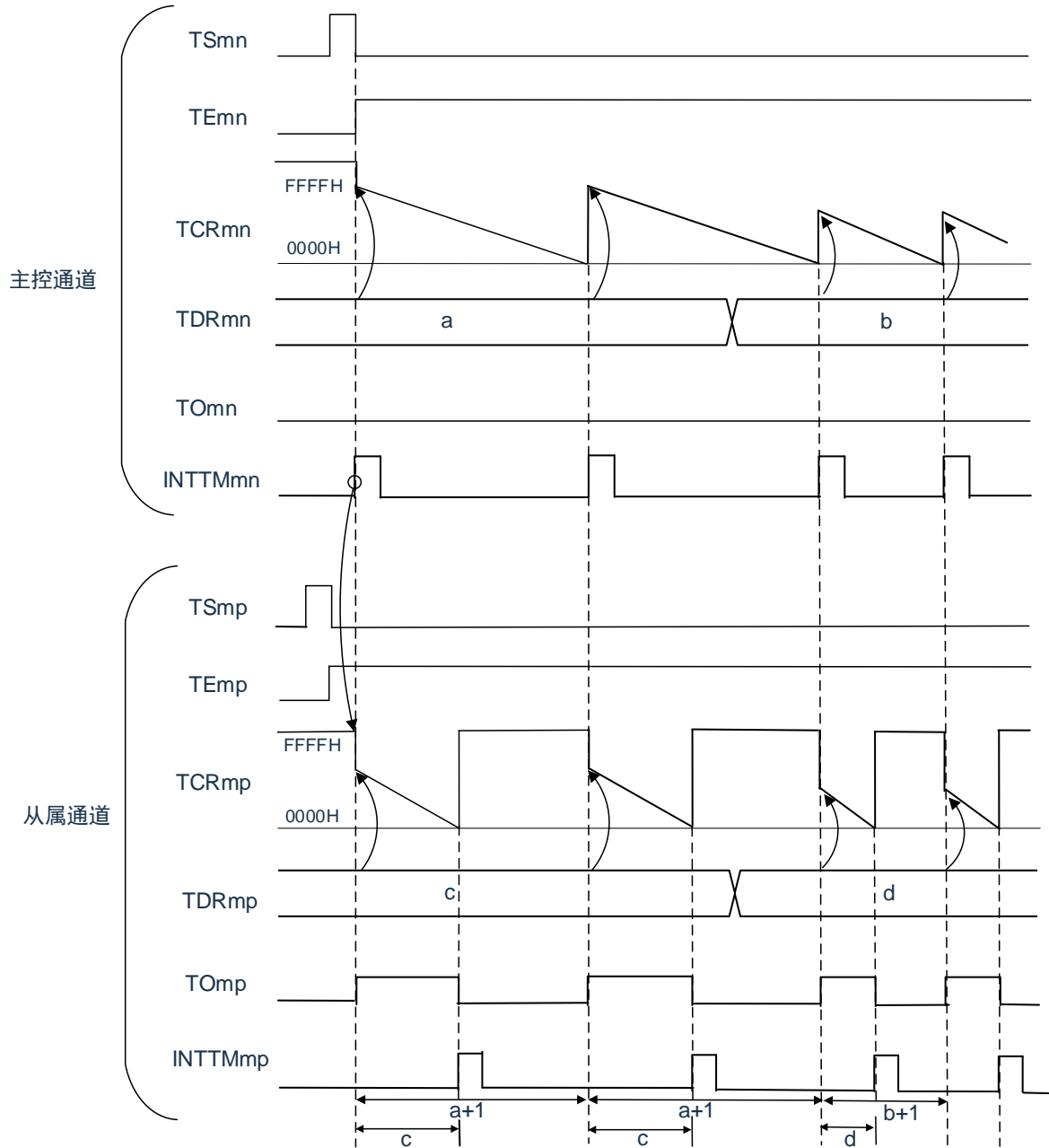
从属通道  
(单次计数模式)



备注: m: 单元号 (m= 0, 1) 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5-38: 作为PWM功能的运行基本时序例子

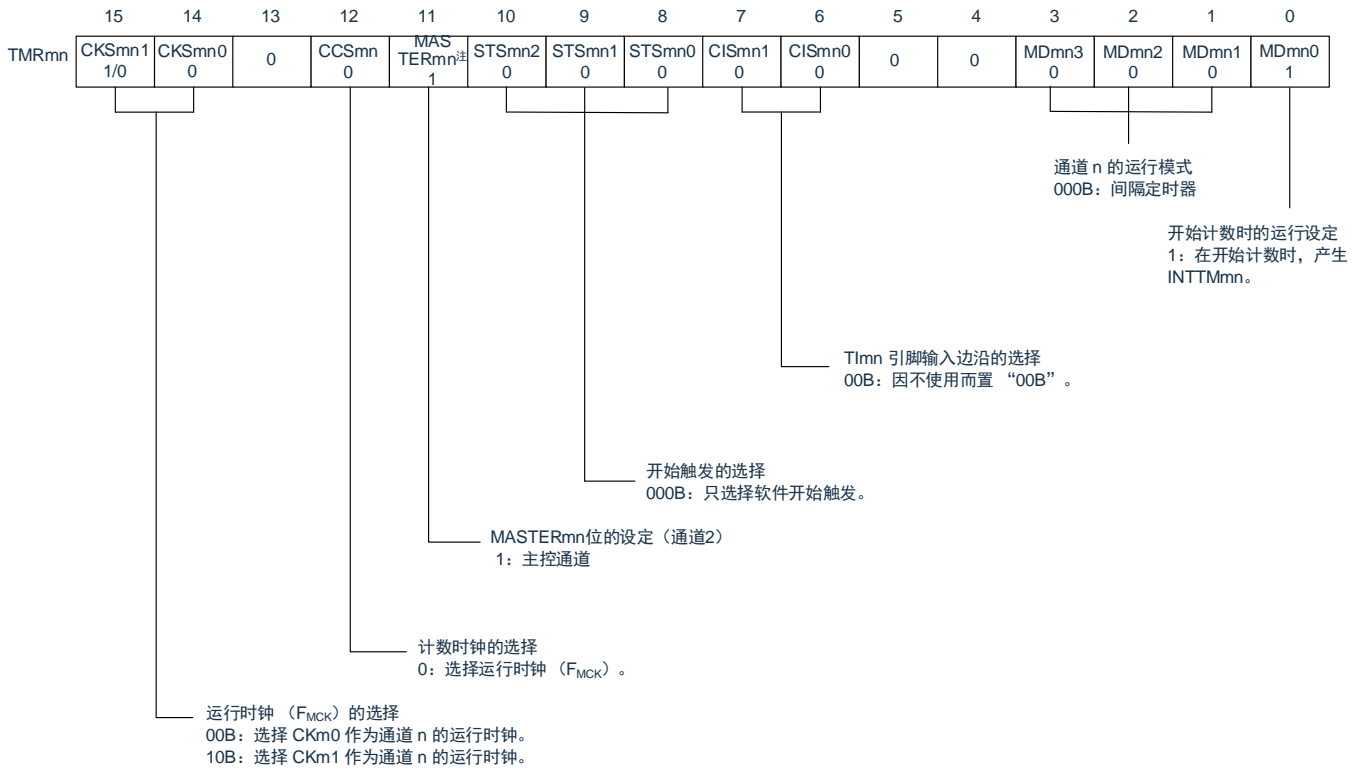


备注:

1. m: 单元号 (m=0, 1) 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  
 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5-39: PWM 功能时（主控通道）的寄存器设定内容例子

(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)



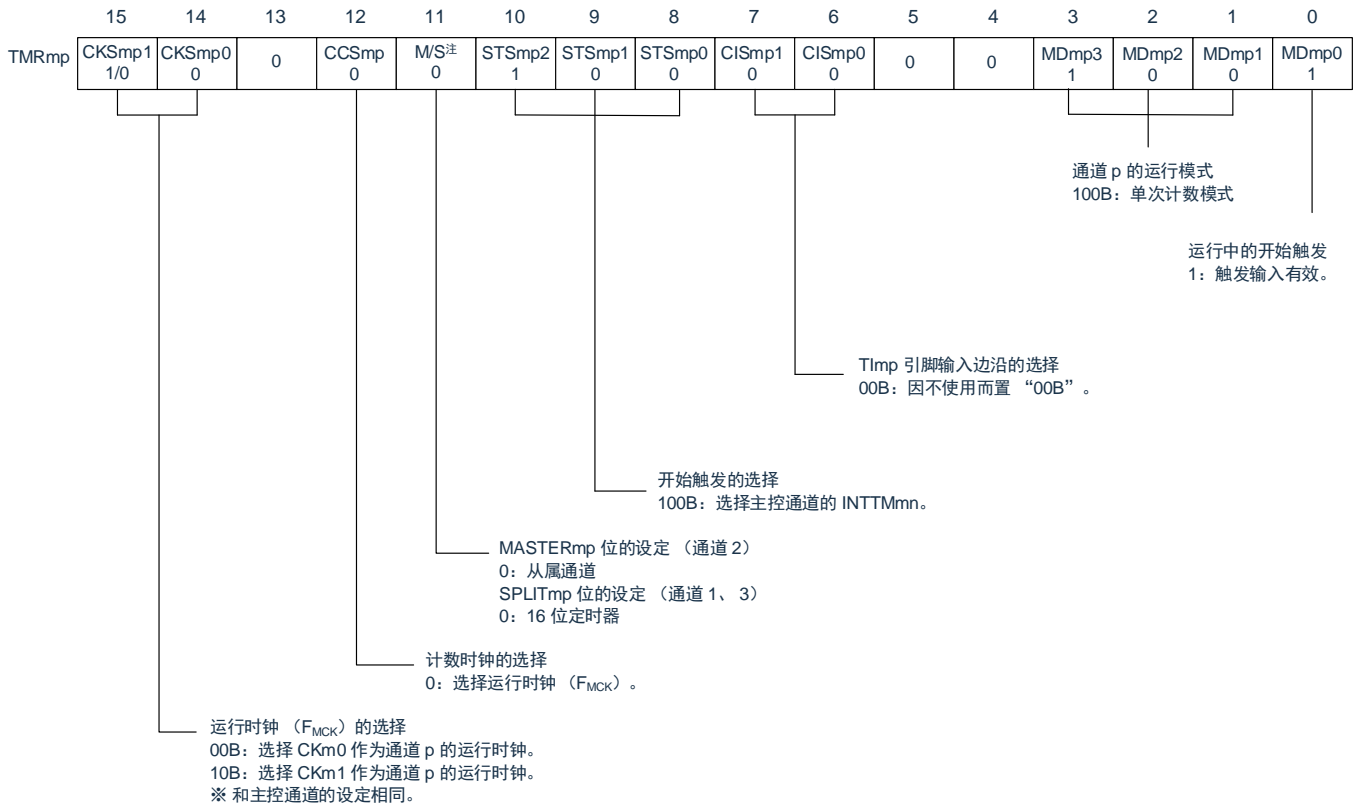
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

备注: m: 单元号 (m=0, 1) n: 主控通道号 (n=0、2)

图5-40: PWM 功能时（从属通道）的寄存器设定内容例子

(a) 定时器模式寄存器mp (TMRmp)



(b) 定时器输出寄存器 m (TOMm)



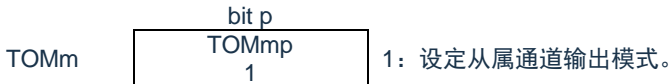
(c) 定时器输出允许寄存器 m (TOEm)



(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1: TMRm2: MASTERmp位

备注:

1. TMRm1、TMRm3: SPLITmp位
2. m: 单元号 (m=0, 1) 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)

表5-31: 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。  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。

表5-31: PWM 功能时的操作步骤(2/2)

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

备注：m：单元号（m= 0, 1） n：主控通道号（n=0）  
p：从属通道号 q：从属通道号 n<p<q≤3（p和q是大于n的整数）

### 5.9.3 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：

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

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。

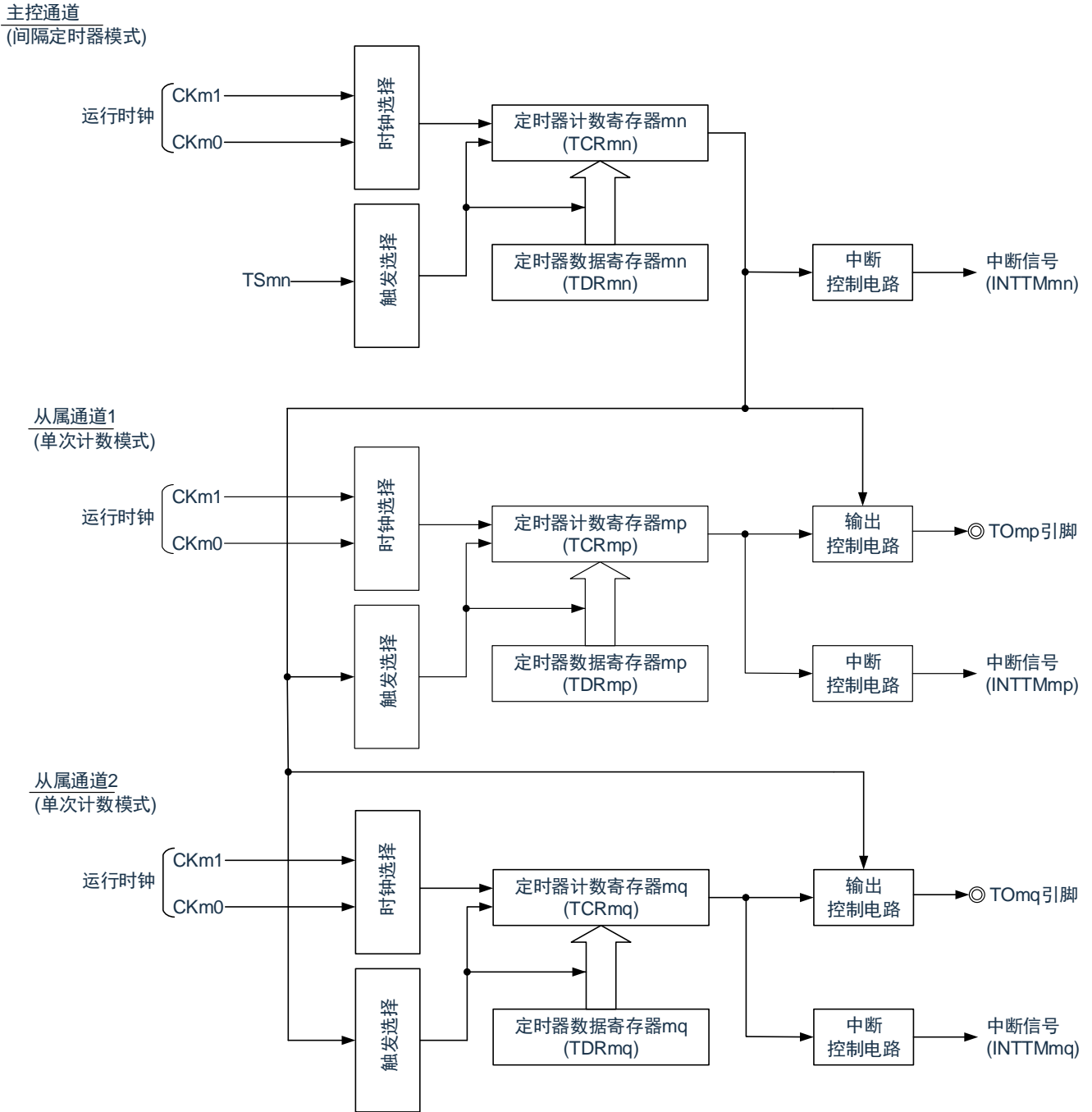
注意：

1. 当TDRmp (从属1) 的设定值 > {TDRmn (主控) 的设定值+1} 或者 {TDRmq (从属2) 的设定值} > {TDRm (主控) 的设定值+1} 时，占空比超过100%，但是为100%输出。
2. 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, 1) 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)

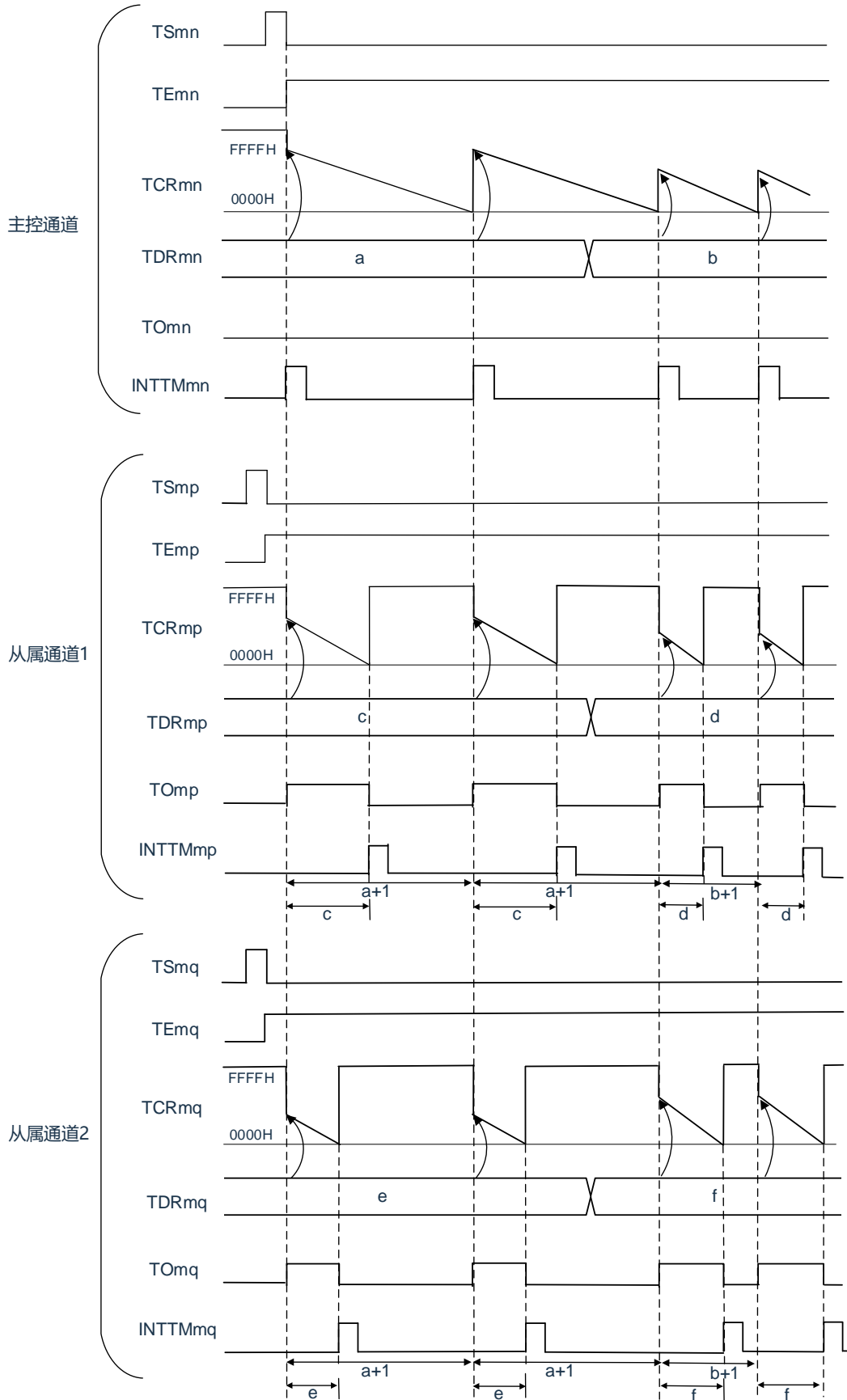
图5-41：作为多重PWM 输出功能运行的框图（输出2 种PWM 的情况）



备注：m：单元号（m= 0, 1） n：主控通道号（n=0）

p：从属通道号 q：从属通道号 n<p<q≤3（p和q是大于n的整数）

图5-42: 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)



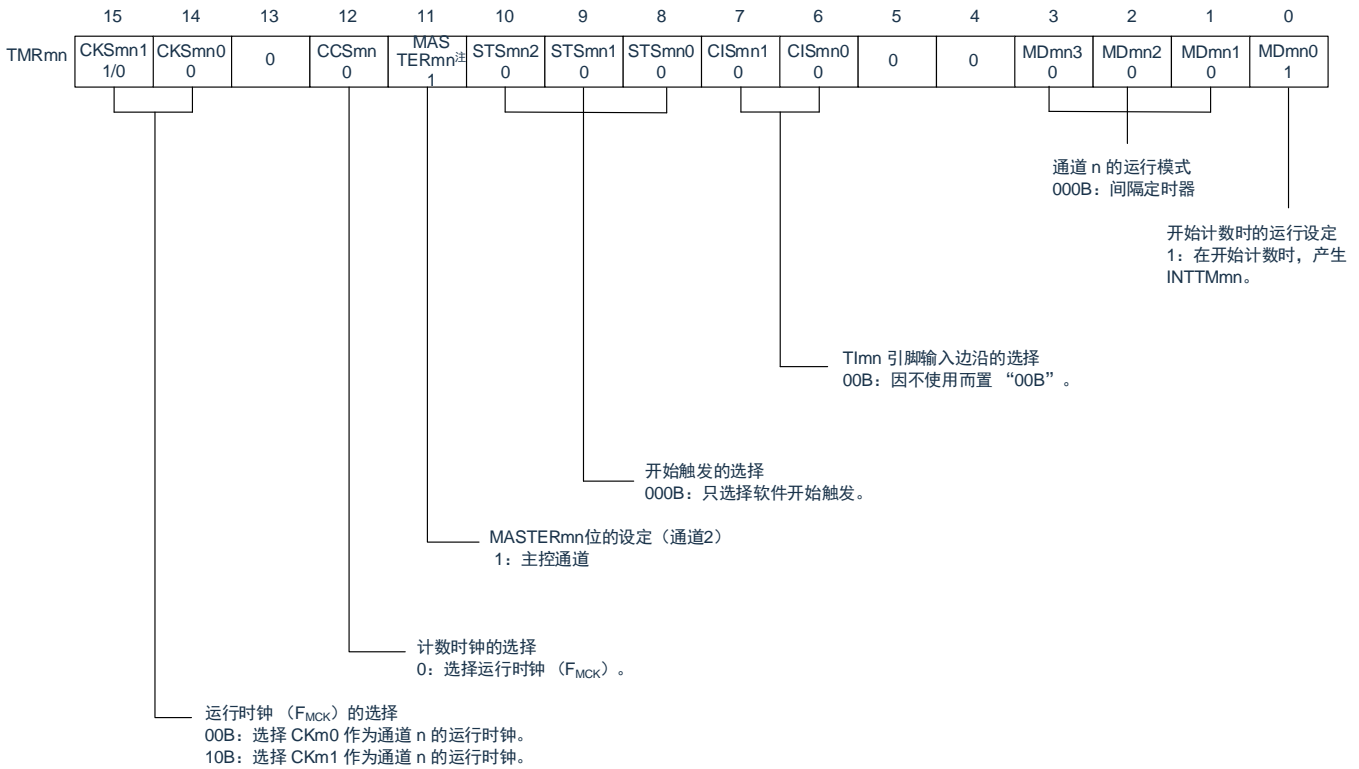


## 备注:

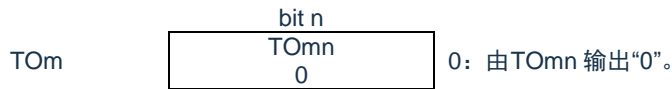
1. m: 单元号 (m= 0, 1) n: 主控通道号 (n=0)  
p: 从属通道号 q: 从属通道号  $n < p < q \leq 3$  (p 和q 是大于n 的整数)
2. TS<sub>mn</sub>、TS<sub>mp</sub>、TS<sub>mq</sub>: 定时器通道开始寄存器m (TS<sub>m</sub>) 的bit n、p、q  
TE<sub>mn</sub>、TE<sub>mp</sub>、TE<sub>mq</sub>: 定时器通道允许状态寄存器m (TE<sub>m</sub>) 的bit n、p、q  
TCR<sub>mn</sub>、TCR<sub>mp</sub>、TCR<sub>mq</sub>: 定时器计数寄存器mn、mp、mq (TCR<sub>mn</sub>、TCR<sub>mp</sub>、TCR<sub>mq</sub>)  
TDR<sub>mn</sub>、TDR<sub>mp</sub>、TDR<sub>mq</sub>: 定时器数据寄存器mn、mp、mq (TDR<sub>mn</sub>、TDR<sub>mp</sub>、TDR<sub>mq</sub>)  
TO<sub>mn</sub>、TO<sub>mp</sub>、TO<sub>mq</sub>: TO<sub>mn</sub>、TO<sub>mp</sub>、TO<sub>mq</sub> 引脚的输出信号

图5-43: 多重PWM 输出功能时（主控通道）的寄存器设定内容例子

(a) 定时器模式寄存器 mn (TMRmn)



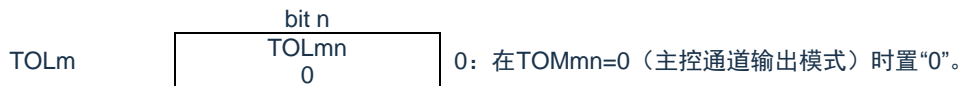
(b) 定时器输出寄存器 m (TOM)



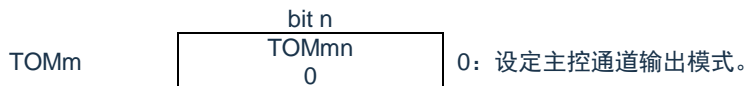
(c) 定时器输出允许寄存器 m (TOEm)



(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)



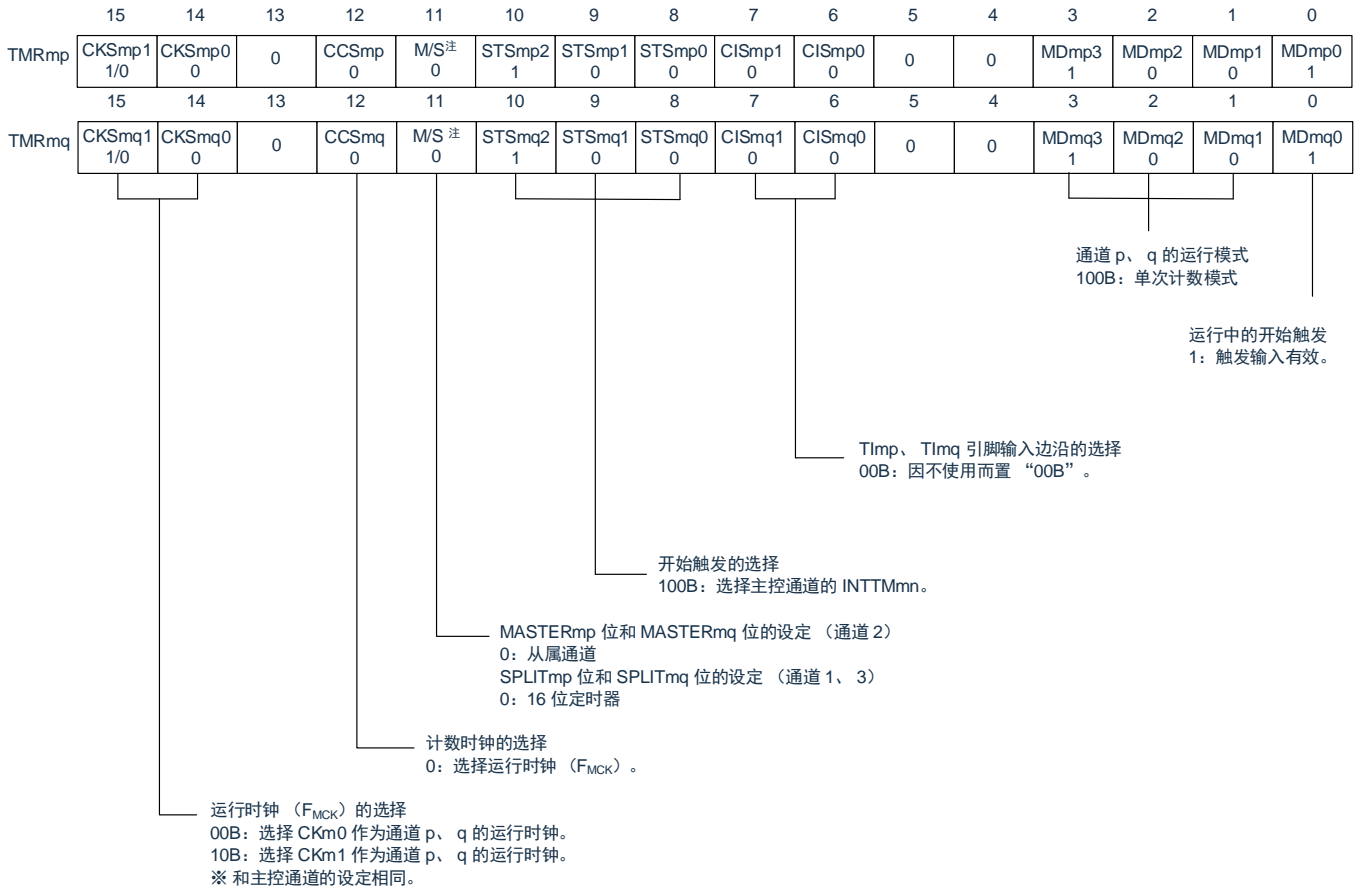
注: TMRm2: MASTERmn=1

TMRm0: 固定为“0”。

备注: m: 单元号 (m= 0, 1) n: 主控通道号 (n=0)

图5-44: 多重PWM 输出功能时（从属通道）的寄存器设定内容例子（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（TOMm）

	bit q	bit p	
TOMm	TOMq 1/0	TOMp 1/0	0: 由TOMp 和TOMq 输出“0”。 1: 由TOMp 和TOMq 输出“1”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit q	bit p	
TOEm	TOEq 1/0	TOEp 1/0	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。

(d) 定时器输出电平寄存器 m（TOLm）

	bit q	bit p	
TOLm	TOLmq 1/0	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 q	bit p	
TOMm	TOMmq 1	TOMmp 1	1: 设定从属通道输出模式。

注: TMRm2: MASTERmp位

备注:

1. TMRm1、TMRm3: SPLITmp位
2. m: 单元号 (m=0, 1) 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)

表5-32: 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”。	TOMP引脚处于 Hi-Z 输出状态。
	设定 TOMP 位和 TOMq 位, 并且确定 TOMP 和 TOMq 输出的初始电平。→	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。
	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”。→	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。

表5-32: 多重PWM 输出功能时的操作步骤 (输出2 种PWM 的情况) (2/2)

	软件操作	硬件状态
重新开始运行	<p><b>开始运行</b></p> <p>(只在重新开始运行时将 TOEmp 位和 TOEmq 位 (从属) 置“1”) 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 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>
	<p><b>运行中</b></p> <p>禁止更改 TMRmn、TMRmp、TMRmq 寄存器以及 TOMmn 位、TOMmp 位、TOMmq 位、TOLmn 位、TOLmp、TOLmq 位的设定值。 能在主控通道产生 INTTMmn 后更改 TDRmn、TDRmp、TDRmq 寄存器的设定值。 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。 从属通道 1 以主控通道的 INTTMmn 信号为触发, 将 TDRmp 寄存器的值传送到 TCRmp 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将 TOmp 的输出电平置为有效电平。然后, 如果计数到“0000H”, 就在将 TOmp 的输出电平置为无效电平后停止计数。 从属通道 2 以主控通道的 INTTMmn 信号为触发, 将 TDRmq 寄存器的值传送到 TCRmq 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将 TOMq 的输出电平置为有效电平。然后, 如果计数到“0000H”, 就在将 TOMq 的输出电平置为无效电平后停止计数。此后, 重复此运行。</p>
	<p><b>停止运行</b></p> <p>将 TTmn 位 (主控)、TTmp 位和 TTmq 位 (从属) 位同时置“1”。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。 TCRmn、TCRmp、TCRmq 寄存器保持计数值而停止计数。 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>
	<p><b>Timer4 停止</b></p> <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”。 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。 将 PER0 寄存器的 TM4mEN 位置“0”。</p>	<p>通过端口功能保持 Tomp 引脚和 Tomq 引脚的输出电平。 定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 (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, 1) 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)

# 第6章 EPWM输出控制电路的功能

使用Timer的PWM输出功能，实现一个直流电机或者两个步进电机的控制。通过截断源INTP0输入以及EVENTC事件，能截断输出。通过软件的设定，能从强制截断时进行Hi-Z输出、低电平输出、高电平输出以及禁止截断输出的4种输出中进行选择。

## 6.1 输出控制电路的结构

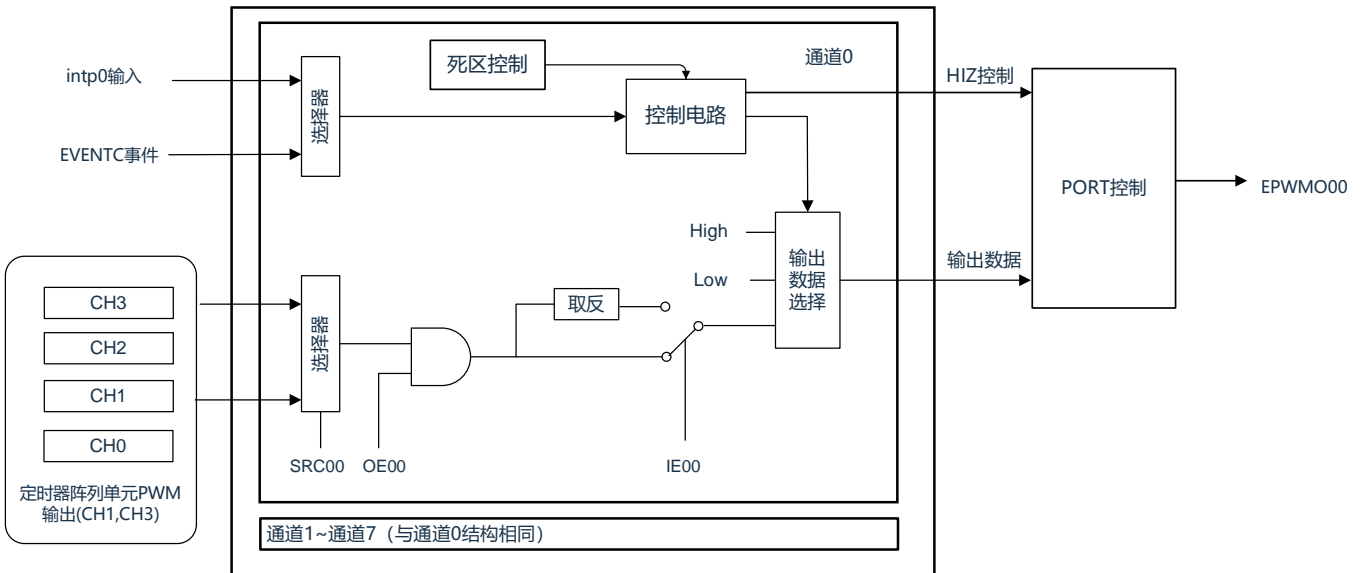
EPWM输出控制电路由以下硬件构成。

表6-1：EPWM的输出控制电路的结构

项目	结构
控制寄存器	EPWM输入源选择寄存器(EPWMSRC)
	EPWM输出控制寄存器(EPWMCTL)
	EPWM强制截断输入选择寄存器(EPWMSTC)
	EPWM强制截断输出选择寄存器(EPWMSTL)
	EPWM状态寄存器(EPWMSTR)
	EPWM死区控制寄存器(EPWMDTC)
输出	EPWM输出(EPWMO00~EPWMO07)

EPWM输出控制电路的框图如图6-1所示。

图6-1：EPWM输出控制电路的框图



## 6.2 EPWM输出控制电路的控制寄存器

通过以下寄存器控制实时输出控制电路。

- 外围允许寄存器0 (PER1)
- EPWM输入源选择寄存器(EPWMSRC)
- EPWM输出控制寄存器(EPWMCTL)
- EPWM强制截断输入选择寄存器(EPWMSTC)
- EPWM强制截断输出选择寄存器(EPWMSTL)
- EPWM状态寄存器(EPWMSTR)
- EPWM死区控制寄存器(EPWMDTC)
- 端口模式寄存器 (PMxx)
- 端口模式控制寄存器 (PMCxx)
- 端口寄存器 (Pxx)

### 6.2.1 外围允许寄存器1 (PER1)

PER1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EPWM功能时，必须将EPWMEN置“1”。

详细请参见“4.3.6外围允许寄存器0、1 (PER0、PER1)”

### 6.2.2 EPWM输入源选择寄存器(EPWMSRC)

EPWMSRC寄存器选择实时输出电路的输入时钟的源时钟。选择Timer 的定时器输出TO01或者TO03作为源时钟，并且输入到EPWM。

通过8位存储器操作指令设定EPWMSRC 寄存器。

通过产生复位信号，此寄存器的值变为“00H”。

表6-2: EPWM输入源选择寄存器的格式

地址: 0x40044400	复位后: 00H							R/W
符号	7	6	5	4	3	2	1	0
EPWMSRC	SRC07	SRC06	SRC05	SRC04	SRC03	SRC02	SRC01	SRC00

SRC0n	选择EPWM0n输出的源时钟
0	选择TO01
1	选择TO03

备注: n: 通道号 (n=0~7)

## 6.2.3 EPWM强制截断输入选择寄存器(EPWMSTC)

EPWMSTC寄存器进行强制截断输入源的选择。

通过8位存储器操作指令设定EPWMSTC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表6-3：EPWM强制截断输入选择寄存器(EPWMSTC)的格式

地址：0x40044404	复位后：00H			R/W				
符号	7	6	5	4	3	2	1	0
EPWMSTC	0	0	0	REL_SEL	HS_SEL	IN_EG	SC_SEL1	SC_SELO

SC_SEL1	SC_SELO	截断源的选择 <sup>注1,3,4</sup>
0	0	不选择
0	1	不选择
1	0	INTP0端子输入
1	1	来自EVENTC的事件输入

IN_EG	输出强制截断的源/输出强制截断解除的源的沿的选择 <sup>注1,2</sup>
0	上升沿：输出强制截断 下降沿：输出强制截断解除
1	上升沿：输出强制截断解除 下降沿：输出强制截断

HS_SEL	输出强制截断的模式选择
0	软件解除
1	硬件解除

REL_SEL	输出强制截断的解除时序选择
0	通过硬件或软件产生的解除信号发生后，截断立即解除，脉冲输出恢复。
1	通过硬件或软件产生的解除信号发生后，等待下列时序： 选择TO01为源时钟的通道：在下一个TO01的上升沿时截断解除，脉冲输出恢复 选择TO03为源时钟的通道：在下一个TO03的上升沿时截断解除，脉冲输出恢复

注意：

1. 在IN\_EG设置后至少间隔三个时钟，再设置SC\_SEL1和SC\_SELO。
2. 只在选择INTP0输入时有效。
3. 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INTP0输入时没有限制。
4. 选择INTP0输入的有效宽度必须大于一个时钟周期。



## 6.2.4 EPWM输出控制寄存器(EPWMCTL)

EPWMCTL寄存器进行EPWMO00~EPWMO07的波形输出的允许控制和取反控制。

通过16位存储器操作指令设定EPWMCTL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表6-4：EPWM输出控制寄存器(EPWMCTL)的格式

地址：0x40044408

复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWMCTL	IE07	IE06	IE05	IE04	IE03	IE02	IE01	IE00	OE07	OE06	OE05	OE04	OE03	OE02	OE01	OE00

OE0n	EPWMO0n输出的控制
0	禁止输出
1	允许输出

备注：n：通道号（n=0~7）

IE0n	EPWMO0n输出的取反控制
0	不取反
1	取反

备注：n：通道号（n=0~7）

## 6.2.5 EPWM强制截断输出选择寄存器(EPWMSTL)

EPWMSTL寄存器进行强制截断时EPWMO端子的输出状态。

通过16位存储器操作指令设定EPWMSTL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表6-5：EPWM强制截断输出选择寄存器(EPWMSTL)的格式

地址：0x4004440C	复位后：0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWMSTL	IO71	IO70	IO61	IO60	IO51	IO50	IO41	IO40	IO31	IO30	IO21	IO20	IO11	IO10	IO01	IO00

IO <sub>n</sub> 1	IO <sub>n</sub> 0	截断时端输出的选择
0	0	禁止截断
0	1	HI-Z输出
1	0	低电平输出
1	1	高电平输出

备注：n：通道号（n=0~7）

## 6.2.6 EPWM状态寄存器(EPWMSTR)

EPWMSTR寄存器清除强制截断信号以及显示截断状态。如果将清除触发位HZCLR置“1”，就解除截断状态。当截断状态标志SHTFLG的信号为高电平时，就进入强制截断状态。bit0为只写位，读取值总是为“0”。bit7~1为只读位。

通过8位存储器操作指令设定EPWMSTR寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表6-6：EPWM状态寄存器(EPWMSTR)的格式

地址：0x4004410	复位后：0000H	R/W						
符号	7	6	5	4	3	2	1	0
EPWMSTR	0	0	0	0	0	0	SHTFLG	HZCLR

SHTFLG	强制截断状态标志
0	通常输出状态
1	强制截断状态

HZCLR	强制截断信号的软件清除
0	-
1	软件解除截断状态

备注：通过强制截断输出选择寄存器（EPWMSTL）设定为禁止截断时，虽然因为发生外部截止源的输入而将SHTFLG置“1”，但是不进行截断处理。

## 6.2.7 EPWM死区控制寄存器(EPWMDTC)

EPWMDTC寄存器进行EPWMO00~EPWMO07的波形输出的死区使能，和死区数值的控制。

通过16位存储器操作指令设定EPWMDTC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表6-7: EPWM死区控制寄存器(EPWMDTC)的格式

地址: 0x40044414	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPWM DTC	DTCVAL [7:0]							EN 7	EN 6	EN 5	EN 4	EN 3	EN 2	EN 1	EN 0	

DTCVAL[7:0]	死区时间（8个通道共用）
-------------	--------------

ENn	EPWMO0n输出的死区使能
1	使能
0	禁止

备注：n：通道号（n=0~7）

## 6.2.8 EPWM输出引脚的端口功能的控制寄存器

使用EPWM输出时，必须设定与EPWM输出引脚（EPWMO<sub>n</sub>引脚）复用的端口功能的控制寄存器（端口模式寄存器（PM<sub>xx</sub>，PMC<sub>xx</sub>）。详细内容请参照“2.3.1 端口模式寄存器（PM<sub>xx</sub>）”。

在将EPWM引脚的复用端口用作EPWMO的输出时，必须将各端口对应的端口模式寄存器（PM<sub>xx</sub>，PMC<sub>xx</sub>）的位置“0”。此时，端口寄存器（P<sub>xx</sub>）的位可以是“0”或者“1”。

详细内容请参照“表2-2 管脚功能数字映射”。

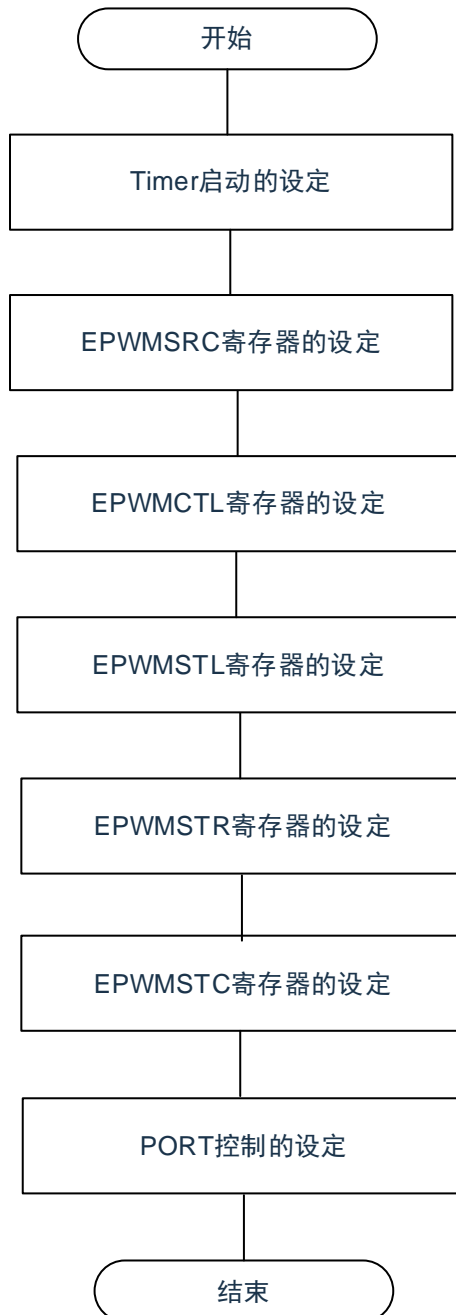
## 6.3 EPWM输出控制电路的运行

### 6.3.1 初始设定

定时器波形通过EPWSRC寄存器选择TAU的输出（TO01、TO03）作为源时钟。能通过设定EPWMCTL寄存器，设定定时器波形的正相或者反相，固定低电平或者高电平。

发生强制截断时，能通过EPWMSTL寄存器的设定选择Hi-Z输出、低电平输出、高电平输出或者禁止截断输出。

图6-2: 寄存器的初始设定流程

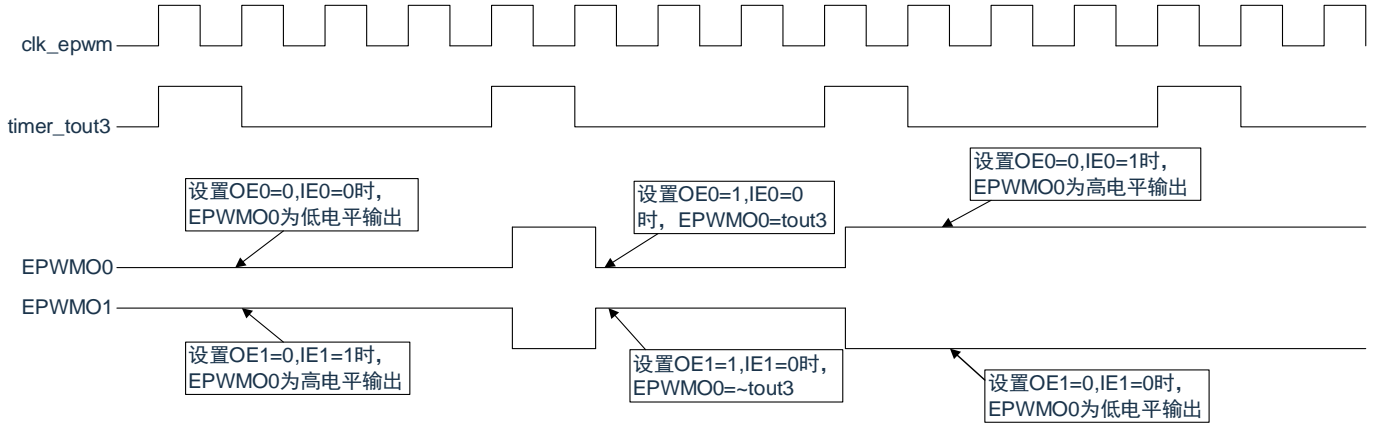


### 6.3.2 通常运行

根据寄存器的设定，可选择4种输出数据，分别是正转波形输出、反转波形输出、低电平输出和高电平输出。运行时可更改EPWMCTL寄存器。必须同时写OE0n位和IE0n位。

详细内容请参照“表6-8 截断信号的运行说明表”。

图6-3：输出时序图



### 6.3.3 强制截断处理

EPWM能通过EPWMSTC的寄存器的bit1, 0选择INTP0输入以及EVENTC事件, 使EPWMO输出进入强制截断状态。

(1) 强制截断的发生

通过INTP0输入以及EVENTC事件进入截断状态。通过EPWMSTC 的寄存器的bit2(IN\_EG), 能选择上升沿或者下降沿, 并且在1~2个时钟后进入截断状态。详细内容请参照图6-4。

(2) 强制截断的解除

a) 软件解除: EPWMSTC的寄存器bit3 (HS\_SEL) 为0时, 使用软件解除模式。EPWMSTR寄存器的bit0 (HZCLR) 是截断状态的清除位。当截断状态标志SHTFLG为高电平时, 如果将HZCLR位置“1”, 截断状态标志SHTFLG就变为低电平, 并且解除强制截断状态。

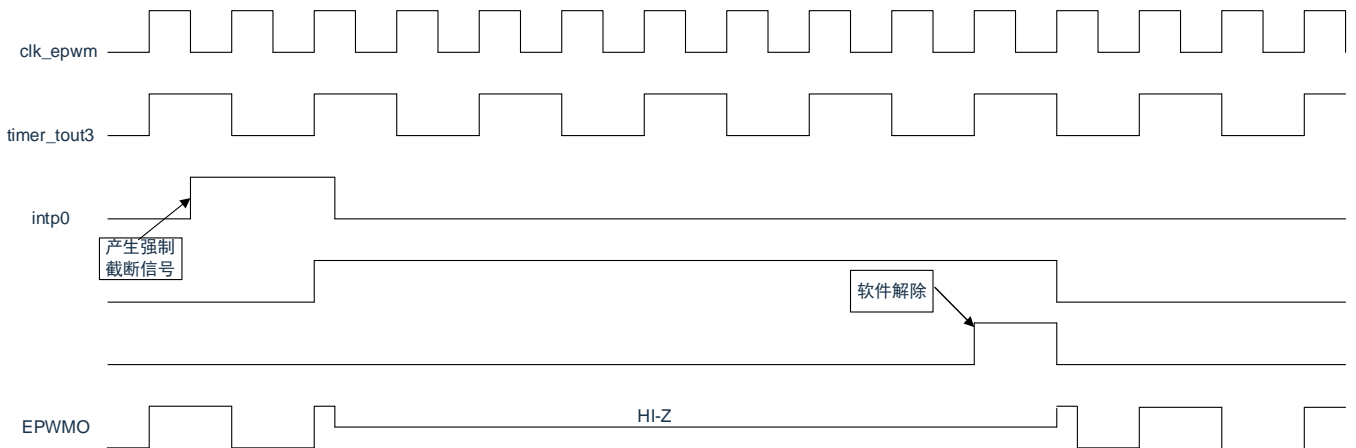
b) 硬件解除: EPWMSTC的寄存器bit3 (HS\_SEL) 为1时, 使用硬件解除模式。通过INTP0输入的边沿解除强制截断状态。

表6-8: 截断信号的运行说明表

位	IOn1-0	OE0n	IE0n	SHTFLG	EPWM输出引脚
设定值	00	1	0	*	正转波形
	00	1	1	*	反转波形
	01	*	*	*	低电平输出
	10	*	*	*	高电平输出
	11	*	*	1	HI-Z输出

备注: n=0~7

图6-4: INTP0截断的产生和解除时序图(HS\_SEL=0,REL\_SEL=0)



备注: 从截断信号INTP0引起的强制截断时的“通常运行”向“Hi-Z”、“固定低电平”或者“固定高电平”切换时, 以及通过立即解除返回强制截断状态时, 都有可能产生短脉冲。

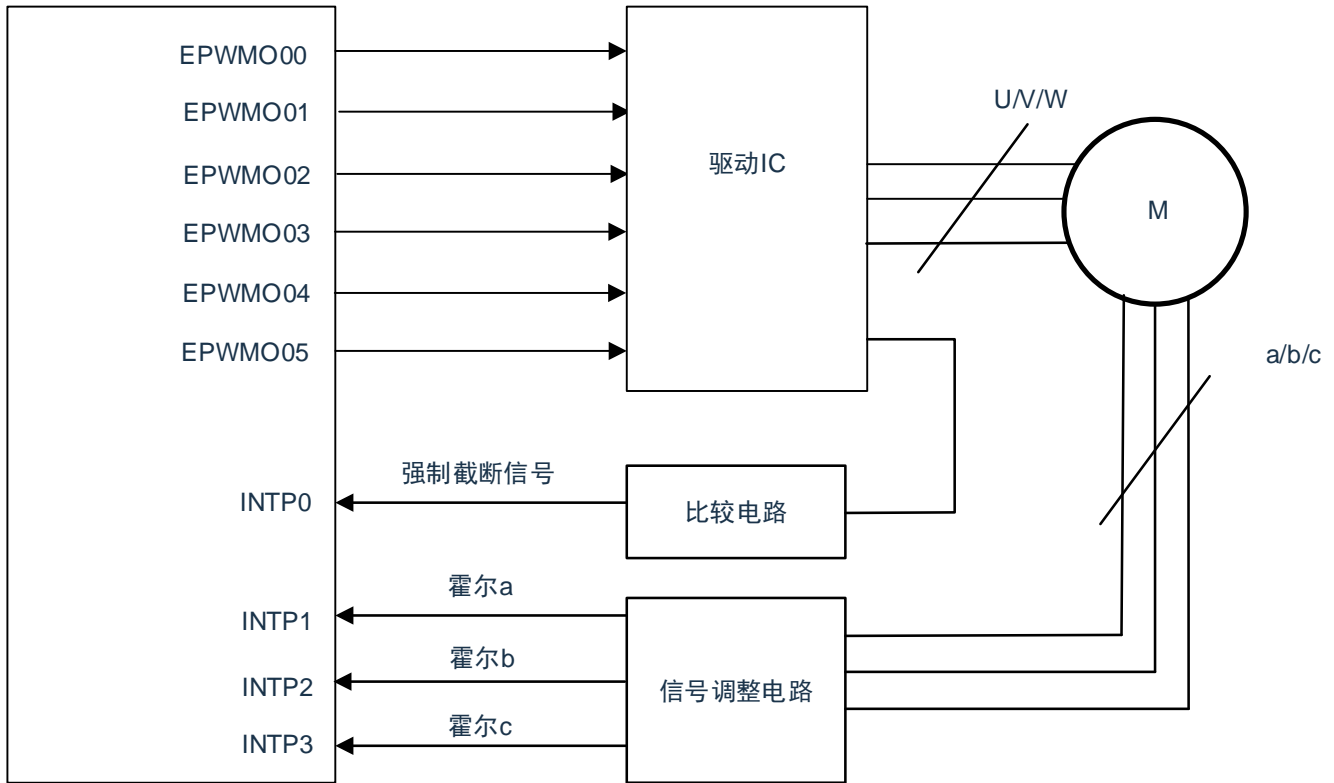
## 6.4 无刷直流电机的控制例子

以下对使用EPWM控制功能控制无刷直流电机（以下简称BLDC电机）的例子进行说明。

### 6.4.1 硬件连接例子

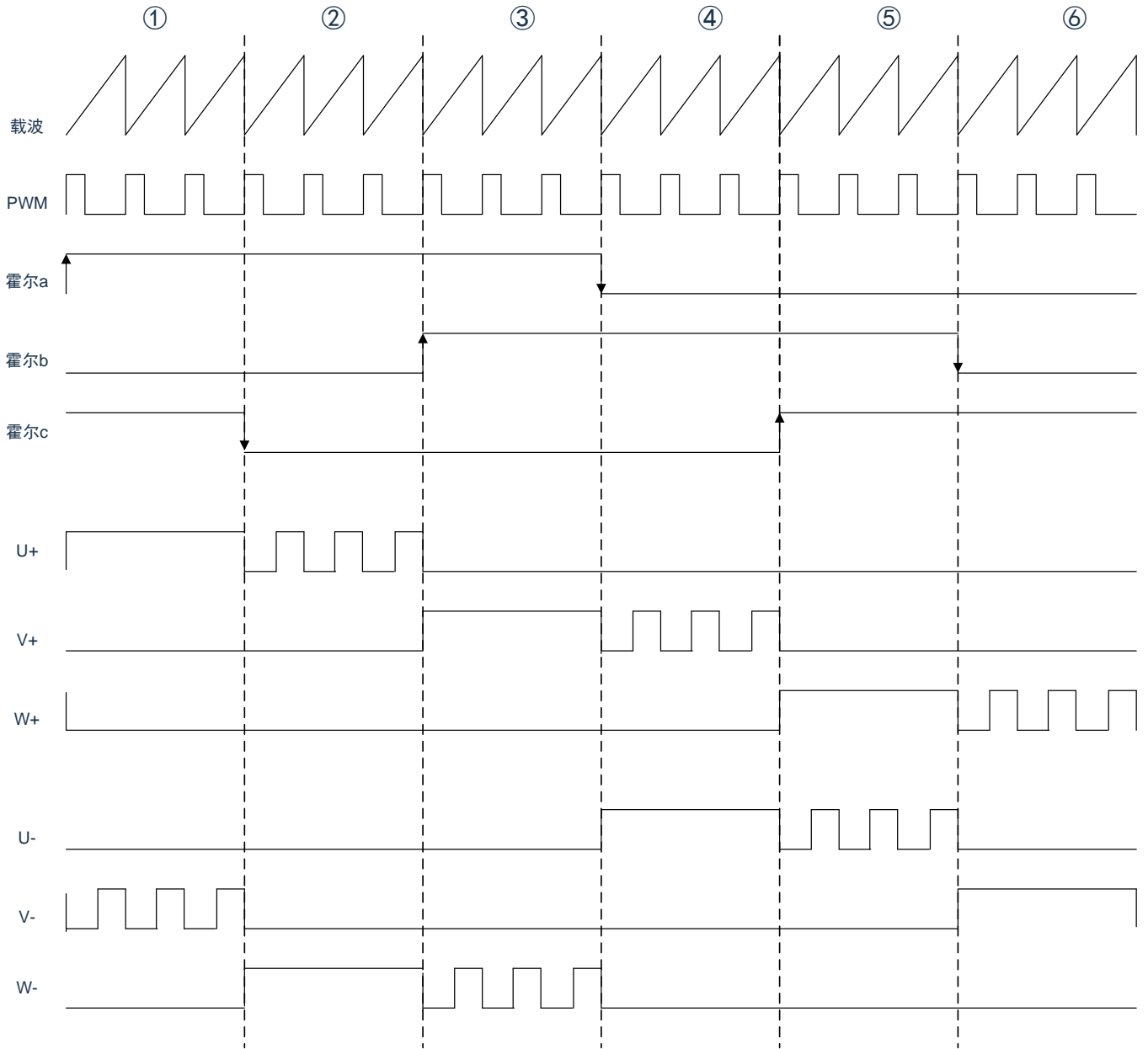
无刷直流电机的硬件连接例子如图6-5所示。在此例中，EPWMO00~EPWMO05（输出）用于BLDC电机的输出控制，INTP1~INTP3（输入）用于霍尔传感器的输出信号，INTP0（输入）用于强制截断信号。

图6-5：硬件连接例子



### 6.4.2 三相无刷直流电机的控制时序

图6-6: 三相无刷直流电机的控制时序





### 6.4.3 寄存器的设定例子

此例中，通过对EPWM源选择寄存器（EPWMSRC）、EPWM控制寄存器（EPWMCTL）进行初始化，同时从EPWM00~EPWM05向BLDC电机输出正向旋转的波形。

1. 将EPWMSRC寄存器的EPWMSRC5~EPWMSRC0位置“0”、将Timer的通道1设定为EPWMO00~EPWMO05的输入源。
2. 将EPWMCTL寄存器的EPWMOE3~EPWMOE0位置“1”，允许EPWMO03~EPWMO00输出。将EPWMCTL寄存器的EPWMIE3~EPWMIE0位置“0”，EPWMO00~EPWMO03正向输出。
3. 将EPWMCTL寄存器的EPWMOE5~EPWMOE4位置“1”，允许EPWMO05~EPWMO04输出。将EPWMCTL寄存器的EPWMIE5~EPWMIE4位置“1”，EPWMO04~EPWMO05反向输出。

表6-9: EPWMCTL0寄存器的设定例子

说明	EPWMCTL的设定值
状态①：霍尔a上升沿 禁止U+、U+反向输出，允许V-、V-正向输出。	0x0110
状态②：霍尔c下降沿 允许U+、U+正向输出，禁止W-、W-反向输出。	0x2001
状态③：霍尔b上升沿 禁止V+、V+反向输出，允许W-、W-正向输出。	0x0220
状态④：霍尔a下降沿 允许V+、V+正向输出，禁止U-、U-反向输出。	0x0802
状态⑤：霍尔c上升沿 禁止W+、W+反向输出，允许U-、U-正向输出。	0x0408
状态⑥：霍尔b下降沿 允许W+、W+正向输出，禁止V-、V-反向输出。	0x1004

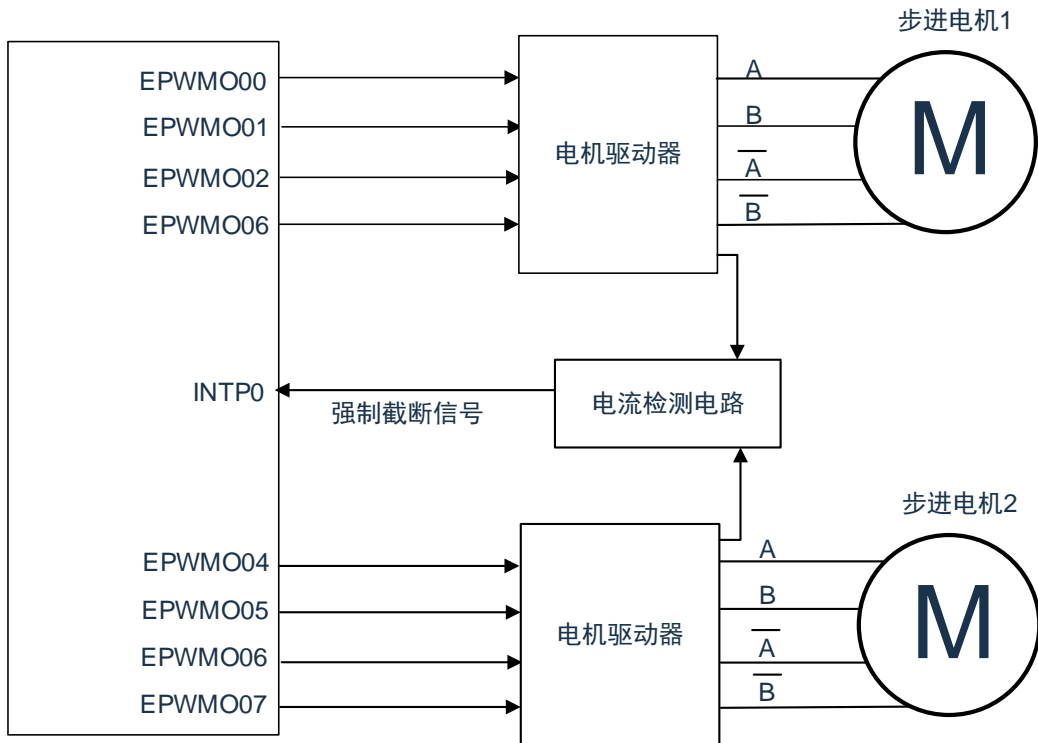
## 6.5 步进电机的控制例子

以下对使用8个实时输出控制2台2相步进电机的例子进行说明。

### 6.5.1 硬件连接例子

控制2台步进电机的硬件连接例子如图6-7所示。

图6-7：硬件连接例子



## 6.5.2 控制方法

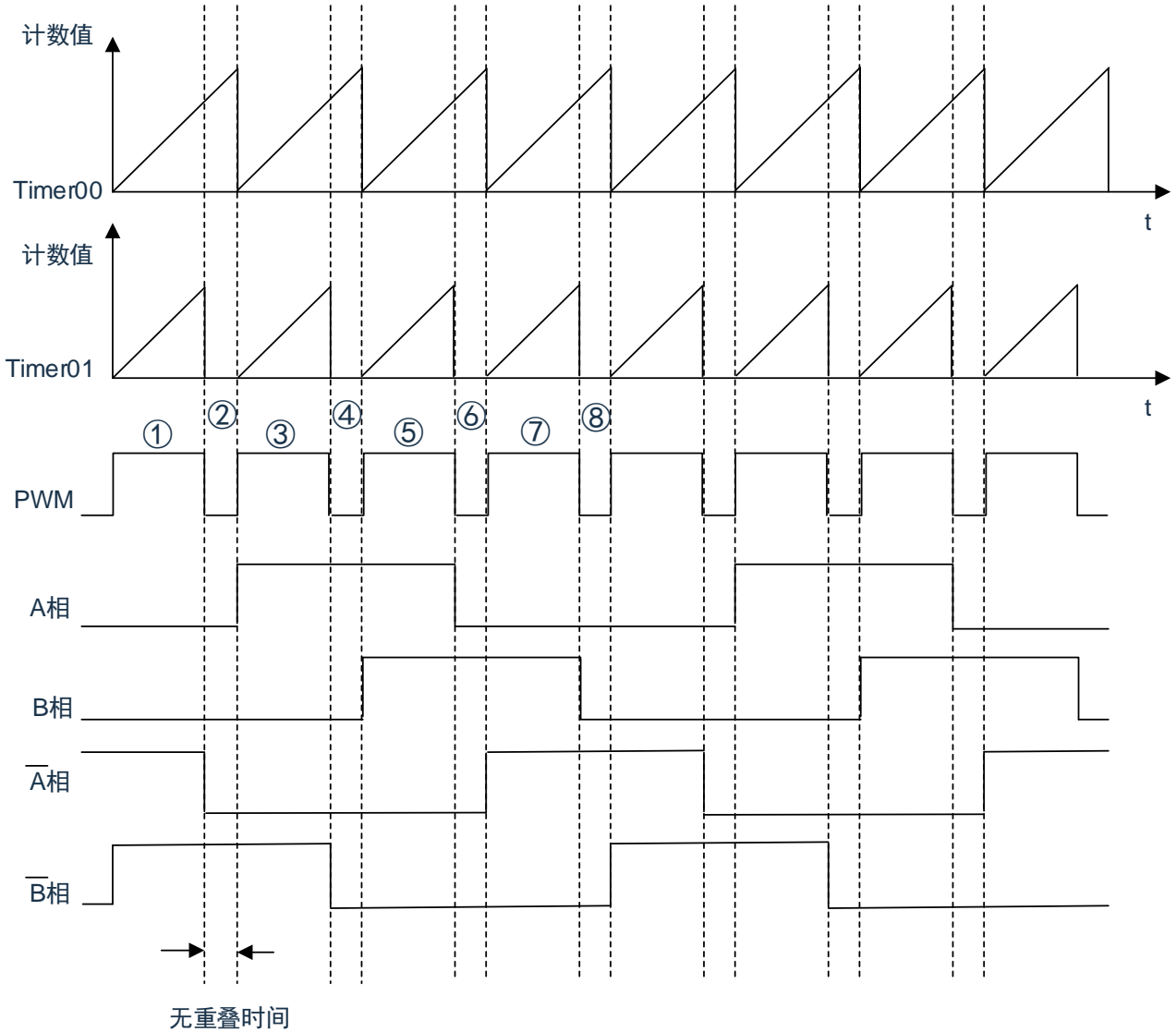
通过8个EPWMO实现步进电机以二相激励方式正转、反转或者停止运行。通过Timer的PWM模式控制旋转速度。

此例中，Timer的CH0和CH1用于步进电机1的控制，CH2和CH3用于步进电机2的控制。如果组合2个Timer的通道，就能生成任意周期和占空比的脉冲。CH0和CH2为主控通道，作为间隔定时器模式运行。CH1和CH3为从属通道，作为单次计数模式运行。

另外，在进行输出类型的切换时插入穿越性电流防止时间（无重叠时间）。

步进电机控制的波形例子如图6-8所示。

图6-8：步进电机控制的波形例子



### 6.5.3 寄存器的设定例子

表6-10: 控制步进电机的寄存器的设定例子

状态		EPWMSRC的设定值	EPWMCTL的设定值
	①	0x00	0x4400
	②	0x00	0x4000
	③	0x00	0x4100
	④	0x00	0x0100
	⑤	0x00	0x0300
	⑥	0x00	0x0200
	⑦	0x00	0x0600
	⑧	0x00	0x0400

## 第7章 实时时钟

### 7.1 实时时钟的功能

实时时钟有以下功能：

拥有年、月、星期、日、小时、分钟和秒的计数器，最长能计数到 99 年。

固定周期中断功能（周期：0.5 秒、1 秒、1 分钟、1 小时、1 日、1 个月）。

闹钟中断功能（闹钟：星期、小时、分钟）。

1Hz 的引脚输出功能。

### 7.2 实时时钟的结构

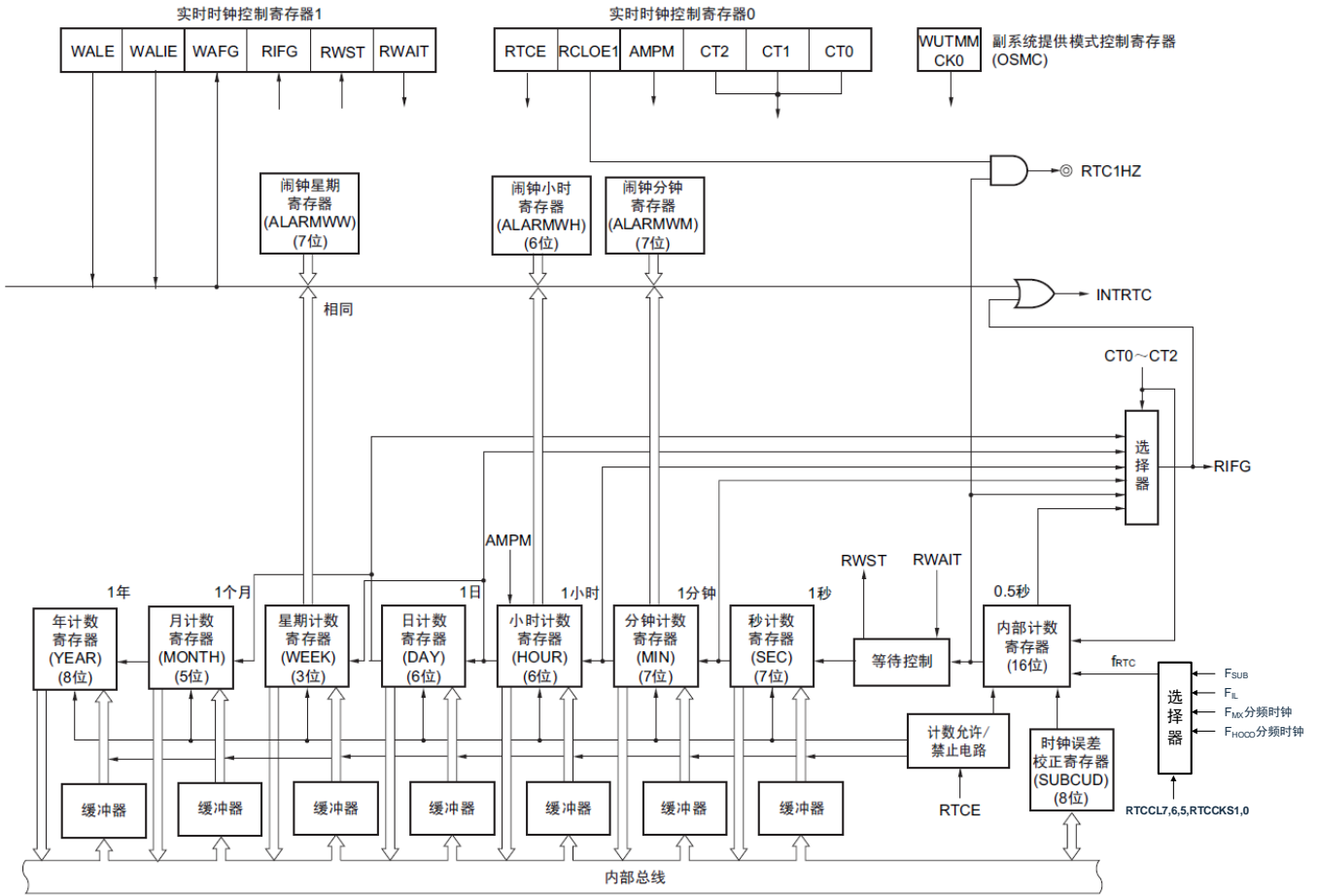
实时时钟由以下硬件构成。

表7-1：实时时钟的结构

项目	结构
计数器	内部计数器（16位）
控制寄存器	外围允许寄存器0（PER0）
	实时时钟选择寄存器（RTCCL）
	实时时钟控制寄存器0（RTCC0）
	实时时钟控制寄存器1（RTCC1）
	秒计数寄存器（SEC）
	分钟计数寄存器（MIN）
	小时计数寄存器（HOUR）
	日计数寄存器（DAY）
	星期计数寄存器（WEEK）
	月计数寄存器（MONTH）
	年计数寄存器（YEAR）
	时钟误差校正寄存器（SUBCUD）
	闹钟分钟寄存器（ALARMWM）
	闹钟小时寄存器（ALARMWH）
闹钟星期寄存器（ALARMWW）	

注意：以上RTC控制寄存器的复位只受POR复位控制。

图7-1：实时时钟的框图



注意：只有在选择F<sub>MX</sub>/F<sub>HOCO</sub>的分频时钟（分频后≈32.768KHZ）或者副系统时钟（F<sub>SUB</sub>=32.768KHZ）作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟（F<sub>IL</sub>=15KHz）时，只能使用固定周期中断功能。选择F<sub>IL</sub>时的固定周期中断间隔用以下计算式进行计算：固定周期(RTCC0寄存器选择的值) × F<sub>SUB</sub>/F<sub>IL</sub>

## 7.3 控制实时时钟的寄存器

通过以下寄存器控制实时时钟：

外围允许寄存器 0 (PER0)

实时时钟选择寄存器 (RTCCL)

实时时钟控制寄存器 0 (RTCC0)

实时时钟控制寄存器 1 (RTCC1)

秒计数寄存器 (SEC)

分钟计数寄存器 (MIN)

小时计数寄存器 (HOUR)

日计数寄存器 (DAY)

星期计数寄存器 (WEEK)

月计数寄存器 (MONTH)

年计数寄存器 (YEAR)

时钟误差校正寄存器 (SUBCUD)

闹钟分钟寄存器 (ALARMWM)

闹钟小时寄存器 (ALARMWH)

闹钟星期寄存器 (ALARMWW)

端口模式寄存器 (PMxx)

端口模式控制寄存器 (PMCxx)

端口复用功能配置寄存器 (PxxCFG)

### 7.3.1 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。

要使用实时时钟时，必须将 bit7 (RTCEN) 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”。

表7-2: 外围允许寄存器0 (PER0) 的格式

地址: 0x40020420	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	TM41EN	TM40EN

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

注意:

- 如果要使用实时时钟，必须先将在计数时钟 ( $F_{RTC}$ ) 振荡稳定的状态下将RTCEN位置“1”，然后设定以下的寄存器。当RTCEN位为“0”时，忽视实时时钟控制寄存器的写操作，而且读取值为初始值（实时时钟选择寄存器 (RTCCCL)、端口模式寄存器和端口寄存器除外）。
  - 实时时钟控制寄存器0 (RTCC0)
  - 实时时钟控制寄存器1 (RTCC1)
  - 秒计数寄存器 (SEC)
  - 分钟计数寄存器 (MIN)
  - 小时计数寄存器 (HOUR)
  - 日计数寄存器 (DAY)
  - 星期计数寄存器 (WEEK)
  - 月计数寄存器 (MONTH)
  - 年计数寄存器 (YEAR)
  - 时钟误差校正寄存器 (SUBCUD)
  - 闹钟分钟寄存器 (ALARMWM)
  - 闹钟小时寄存器 (ALARMWH)
  - 闹钟星期寄存器 (ALARMWW)
- 能通过将副系统时钟提供模式控制寄存器 (OSMC) 的RTCLPC位置“1”，在深度睡眠模式或者以副系统时钟运行的睡眠模式中停止给实时时钟和15位间隔定时器以外的外围功能提供副系统时钟。



## 7.3.2 实时时钟选择寄存器 (RTCCL)

能通过 RTCCL 选择实时时钟和 15 位间隔定时器的计数时钟 (F<sub>RTC</sub>)。

表7-3: 实时时钟选择寄存器 (RTCCL) 的格式

地址: 0x4004047C      复位后: 00H      R/W

符号	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCKS1	RTCKS0

RTCCL7	实时时钟、15位间隔定时器的计数时钟的时钟源的选择
0	选择高速系统时钟(F <sub>MX</sub> )
1	选择高速内部振荡器(F <sub>HOCO</sub> )

RTCKS1	RTCKS0	RTCCL6	RTCCL5	实时时钟、15位间隔定时器的计数时钟的运行时钟的选择
0	0	x	x	副系统时钟 (F <sub>SUB</sub> )
0	1			低速内部振荡器时钟 (F <sub>IL</sub> ) (必须设置WUTMMCK0为1)
1	0	0	1	主时钟F <sub>MAX</sub> /F <sub>HOCO</sub> (通过RTCCL7选择)/1952
1	0	0	0	主时钟F <sub>MAX</sub> /F <sub>HOCO</sub> (通过RTCCL7选择)/1464
1	0	1	0	主时钟F <sub>MAX</sub> /F <sub>HOCO</sub> (通过RTCCL7选择)/976
1	1	0	0	主时钟F <sub>MAX</sub> /F <sub>HOCO</sub> (通过RTCCL7选择)/488
1	1	1	0	主时钟F <sub>MAX</sub> /F <sub>HOCO</sub> (通过RTCCL7选择)/244

### 7.3.3 实时时钟控制寄存器0 (RTCC0)

这是设定实时时钟的运行开始或者停止、RTC1HZ 引脚的控制、12/24 小时系统和固定周期中断功能的 8 位寄存器。

通过 8 位存储器操作指令设定 RTCC0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表7-4：实时时钟控制寄存器0 (RTCC0) 的格式

地址: 0x40044F5D	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1 <sup>注</sup>	0	AMPM	CT2	CT1	CT0

RTCE	实时时钟的运行控制
0	停止计数器的运行。
1	开始计数器的运行。

RCLOE1	RTC1HZ引脚的输出控制
0	禁止RTC1HZ引脚的输出 (1Hz)。
1	允许RTC1HZ引脚的输出 (1Hz)。

AMPM	12小时系统/24小时系统的选择
0	12小时系统 (表示上午或者下午)
1	24小时系统

要更改AMPM位的值时，必须在将RWAIT位 (实时时钟控制寄存器1 (RTCC1) 的bit0) 置“1”后进行改写。如果更改AMPM位的值，小时计数寄存器 (HOUR) 的值就变为所设时间系统的对应值。时间位的表示如表7-11所示。

CT2	CT1	CT0	固定周期中断 (INTRTC) 的选择
0	0	0	不使用固定周期中断功能。
0	0	1	0.5秒一次 (与秒累加同步)
0	1	0	1秒一次 (与秒累加同时)
0	1	1	1分钟一次 (每分钟的00秒)
1	0	0	1小时一次 (每小时的00分00秒)
1	0	1	1日一次 (每日的00点00分00秒)
1	1	x	1个月一次 (每月的1日上午00点00分00秒)

要在计数器运行中 (RTCE=1) 更改CT2~CT0位的值时，必须在通过中断屏蔽标志寄存器将INTRTC设定为禁止中断处理后进行改写，并且必须在改写后清除RIFG标志和RTCIF标志，然后再设定为允许中断处理。

注1：在RTCE位为“1”时，不能更改RCLOE1位。

注2：在RTCE位为“0”时，即使将RCLOE1位置“1”也不输出1Hz。

备注：x：忽略

### 7.3.4 实时时钟控制寄存器1 (RTCC1)

这是控制闹钟中断功能和计数器等待的 8 位寄存器。通过 8 位存储器操作指令设定 RTCC1 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表7-5: 实时时钟控制寄存器1 (RTCC1) 的格式(1/2)

地址: 0x40044F5E	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	闹钟的运行控制
0	一致运行无效。
1	一致运行有效。

要在计数器运行中 (RTCE=1) 并且WALIE位为“1”的情况下设定WALE位时，必须在通过中断屏蔽标志寄存器将INTRTC设定为禁止中断处理后进行改写，并且必须在改写后清除WAFG标志和RTCIF标志。要设定各闹钟寄存器 (RTCC1寄存器的WALIE标志、闹钟分钟寄存器 (ALARMWM)、闹钟小时寄存器 (ALARMWH) 和闹钟星期寄存器 (ALARMWW)) 时，必须将WALE位置“0” (一致运行无效)。

WALIE	闹钟中断 (INTRTC) 功能的运行控制
0	不产生闹钟一致中断。
1	产生闹钟一致中断。

WAFG	闹钟检测状态标志
0	闹钟不一致。
1	检测到闹钟一致。

这是表示检测到闹钟一致的状态标志。只在WALE位为“1”时有效，在检测到闹钟一致并且经过1个f<sub>RTC</sub>时钟后变为“1”。通过给此标志写“0”来清除此标志。写“1”的操作无效。

表7-6: 实时时钟控制寄存器1 (RTCC1) 的格式(2/2)

RIFG	固定周期中断状态标志
0	没有产生固定周期中断。
1	产生固定周期中断。

这是表示产生固定周期中断的状态标志。当产生固定周期中断时，此标志为“1”。  
通过给此标志写“0”来清除此标志。写“1”的操作无效。

RWST	实时时钟的等待状态标志
0	计数器正在运行。
1	正处于计数器的读写模式。

这是表示RWAIT位的设定是否有效的状态。必须在确认此标志为“1”后读写计数值。

RWAIT	实时时钟的等待控制
0	设定为计数器运行。
1	设定为SEC~YEAR计数器停止运行，进入计数器的读写模式。

此位控制计数器的运行。要读写计数值时，必须给此位写“1”。  
因为内部计数器（16位）继续运行，所以必须在1秒内结束读写，然后返回到“0”。  
从将RWAIT位置“1”到能读写计数值（RWST=1）为止，最多需要1个FRTC时钟的时间。  
如果在RWAIT位为“1”时发生内部计数器（16位）上溢，就保持发生上溢的状态，在RWAIT位变为“0”后进行递增计数。  
但是，当写秒计数寄存器时，不保持发生上溢的状态。

**注意：**

1. 固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这2个中断的情况下，能在发生INTRTC时通过确认固定周期中断状态标志（RIFG）和闹钟检测状态标志（WAFG）来判断发生的是哪个中断。
2. 如果写秒计数寄存器（SEC），就清除内部计数器（16位）。

### 7.3.5 时钟误差校正寄存器 (SUBCUD)

这是能通过改变从内部计数器（16 位）到秒计数寄存器（SEC）的上溢值（基准值：7FFFH）来高精度地校正时钟快慢的寄存器。

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

表7-7：时钟误差校正寄存器（SUBCUD）的格式

地址：0x40044F34H	复位后：0000H	R/W						
符号	15	14	13	12	11	10	9	8
SUBCUD	DEV	0	0	F12	F11	F10	F9	F8
	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

DEV	时钟误差的校正时序的设定
0	在秒位为“00”、“20”、“40”时（每20秒）进行时钟误差的校正。
1	只在秒位为“00”时（每60秒）进行时钟误差的校正。

在以下所示期间禁止写SUBCUD寄存器：  
 DVE=0：SEC=00H、20H、40H的期间  
 DVE=1：SEC=00H的期间

F12	时钟误差校正值的设定
0	$\{((F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) - 1) \times 2\}$ 增加
1	$\{((F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) + 1) \times 2\}$ 减少

当 (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) 或者 (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1) 时, 不进行时钟误差的校正。  
 校正值的范围： (F12=0) 2、4、6、8、.....、8186、8188  
 (F12=1) -2、-4、-6、-8、.....、-8186、-8188

备注：“/”表示各位取反后的值。

能通过时钟误差校正寄存器（SUBCUD）进行校正的范围如下所示。

	DEV=0（每20秒的校正）	DEV=1（每60秒的校正）
可校正的范围	-12496.9ppm~12496.9ppm	-4165.6ppmto4165.6ppm
最大量化误差	±1.53ppm	±0.51ppm
最小分辨率	±3.05ppm	±1.02ppm

注意：当校正范围超出-4165.6ppm~4165.6ppm的范围时，必须将DEV位置“0”。

### 7.3.6 秒计数寄存器 (SEC)

这是用 0~59（十进制）表示秒计数值的 8 位寄存器。通过内部计数器（16 位）的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多 2 个 F<sub>RTC</sub> 时钟后被写到计数器。必须以 BCD 码设定十进制的 00~59。

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

表7-8：秒计数寄存器（SEC）的格式

地址：0x40044F52	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

注意：

1. 要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。
2. 如果写秒计数寄存器（SEC），就清除内部计数器（16位）。

### 7.3.7 分钟计数寄存器 (MIN)

这是用 0~59（十进制）表示分钟计数值的 8 位寄存器。通过秒计数器的上溢进行递增计数。

在写时，数据先被写到缓冲器，在经过最多 2 个 F<sub>RTC</sub> 时钟后被写到计数器。在写操作过程中忽视秒计数寄存器的上溢并且设定为写入值。必须以 BCD 码设定十进制的 00~59。

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

表7-9：分钟计数寄存器（MIN）的格式

地址：0x40044F53	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

### 7.3.8 小时计数寄存器 (HOUR)

这是用 00~23 或者 01~12、21~32 (十进制) 表示小时计数值的 8 位寄存器。通过分钟计数器的上溢进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个  $F_{RTC}$  时钟后被写到计数器。在写操作过程中忽视分钟计数寄存器的上溢并且设定为写入值。

必须根据实时时钟控制寄存器 0 (RTCC0) 的 bit3 (AMPM) 设定的时间系统, 以 BCD 码设定十进制的 00~23 或者 01~12、21~32。

如果更改 AMPM 位的值, HOUR 寄存器的值就变为所设时间系统的对应值。通过 8 位存储器操作指令设定 HOUR 寄存器。在产生复位信号后, 此寄存器的值变为“12H”。

但是, 如果在复位后将 AMPM 位置“1”, 此寄存器的值就变为“00H”。

表7-10: 小时计数寄存器 (HOUR) 的格式

地址: 0x40044F54	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意:

1. 当选择AMPM位为“0” (12小时系统) 时, HOUR寄存器的bit5 (HOUR20) 表示AM (0) /PM (1)。
2. 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

AMPM 位的设定值、小时计数寄存器（HOUR）的值和时间的关系如表 7-11 所示。

表7-11：时间位的表示

24小时表示 (AMPM=1)		12小时表示 (AMPM=0)	
时间	HOUR 寄存器	时间	HOUR 寄存器
0 时	00H	AM 12 时	12H
1 时	01H	AM 1 时	01H
2 时	02H	AM 2 时	02H
3 时	03H	AM 3 时	03H
4 时	04H	AM 4 时	04H
5 时	05H	AM 5 时	05H
6 时	06H	AM 6 时	06H
7 时	07H	AM 7 时	07H
8 时	08H	AM 8 时	08H
9 时	09H	AM 9 时	09H
10 时	10H	AM 10 时	10H
11 时	11H	AM 11 时	11H
12 时	12H	PM 12 时	32H
13 时	13H	PM 1 时	21H
14 时	14H	PM 2 时	22H
15 时	15H	PM 3 时	23H
16 时	16H	PM 4 时	24H
17 时	17H	PM 5 时	25H
18 时	18H	PM 6 时	26H
19 时	19H	PM 7 时	27H
20 时	20H	PM 8 时	28H
21 时	21H	PM 9 时	29H
22 时	22H	PM 10 时	30H
23 时	23H	PM 11 时	31H

当 AMPM 位为“0”时，HOUR 寄存器的值为 12 小时表示；当 AMPM 位为“1”时，HOUR 寄存器的值为 24 小时表示。在 12 小时表示时，HOUR 寄存器的 bit5 表示上午/下午。上午（AM）为“0”，下午（PM）为“1”。



### 7.3.9 日计数寄存器 (DAY)

这是用 1~31 (十进制) 表示日计数值的 8 位寄存器。通过小时计数器的上溢进行递增计数。计数器进行以下的计数:

01~31 (1、3、5、7、8、10、12月)

01~30 (4、6、9、11月)

01~29 (2月闰年)

01~28 (2月平常年)

在写时, 数据先被写到缓冲器, 在经过最多 2 个 F<sub>RTC</sub> 时钟后被写到计数器。在写操作过程中忽视小时计数寄存器的上溢并且设定为写入值。必须以 BCD 码设定十进制的 01~31。

通过 8 位存储器操作指令设定 DAY 寄存器。在产生复位信号后, 此寄存器的值变为“01H”。

表7-12: 日计数寄存器 (DAY) 的格式

地址: 0x40044F56H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

注意: 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

### 7.3.10 星期计数寄存器 (WEEK)

这是用 0~6 (十进制) 表示星期计数值的 8 位寄存器。与日计数器同步进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个  $F_{RTC}$  时钟后被写到计数器。必须以 BCD 码设定十进制的 00~06。

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

表7-13: 星期计数寄存器 (WEEK) 的格式

地址: 0x40044F55H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

注意: 月计数寄存器 (MONTH) 和日计数寄存器 (DAY) 的对应值不自动保存到星期寄存器 (WEEK)。

必须在解除复位后进行以下的设定:

星期	WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

注意: 要在计数器运行中 ( $RTCE=1$ ) 读写此寄存器时, 必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

### 7.3.11 月计数寄存器 (MONTH)

这是用 1~12 (十进制) 表示月计数值的 8 位寄存器。通过日计数器的上溢进行递增计数。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 F<sub>RTC</sub> 时钟后被写到计数器。在写操作过程中忽视日计数寄存器的上溢并且设定为写入值。必须以 BCD 码设定十进制的 01~12。

通过 8 位存储器操作指令设定 MONTH 寄存器。在产生复位信号后, 此寄存器的值变为“01H”。

表7-14: 月计数寄存器 (MONTH) 的格式

地址: 0x40044F57H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

注意: 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

### 7.3.12 年计数寄存器 (YEAR)

这是用 0~99 (十进制) 表示年计数值的 8 位寄存器。通过月计数器 (MONTH) 的上溢进行递增计数。

00、04、08、……、92、96 是闰年。

在写时, 数据先被写到缓冲器, 在经过最多 2 个 F<sub>RTC</sub> 时钟后被写到计数器。在写操作过程中忽视 MONTH 寄存器的上溢并且设定为写入值。必须以 BCD 码设定十进制的 00~99。通过 8 位存储器操作指令设定 YEAR 寄存器。在产生复位信号后, 此寄存器的值变为“00H”。

表7-15: 年计数寄存器 (YEAR) 的格式

地址: 0x40044F58H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

注意: 要在计数器运行中 (RTCE=1) 读写此寄存器时, 必须按照“7.4.3 实时时钟计数器的读写”记载的步骤进行。

### 7.3.13 闹钟分钟寄存器 (ALARMWM)

这是设定闹钟分钟的寄存器。

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

注：必须以BCD码设定十进制的00~59。如果设定范围以外的值，就不检测闹钟。

表7-16：闹钟分钟寄存器 (ALARMWM) 的格式

地址: 0x40044F5AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

### 7.3.14 闹钟小时寄存器 (ALARMWH)

这是设定闹钟小时的寄存器。

通过 8 位存储器操作指令设定 ALARMWH 寄存器。在产生复位信号后，此寄存器的值变为“12H”。

但是，如果在复位后将 AMPM 位置“1”，此寄存器的值就变为“00H”。

注意：必须以BCD码设定十进制的00~23或者01~12、21~32。如果设定范围以外的值，就不检测闹钟。

表7-17：闹钟小时寄存器 (ALARMWH) 的格式

地址: 0x40044F5BH	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

注意：当选择AMPM位为“0”（12小时系统）时，ALARMWH寄存器的bit5（WH20）表示AM（0）/PM（1）。

### 7.3.15 闹钟星期寄存器 (ALARMWW)

这是设定闹钟星期的寄存器。

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

表7-18: 闹钟星期寄存器 (ALARMWW) 的格式

地址: 0x40044F5CH	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

闹钟时间的设定例子如下所示。

闹钟设定时间	星期							12小时表示				24小时表示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	10时	1时	10分	1分	10时	1时	10分	1分
	W	W	W	W	W	W	W								
每天 上午0时00分	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天 上午1时30分	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 上午11时59分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一~星期五 下午0时00分	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 下午1时30分	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 下午11时59分	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

### 7.3.16 端口模式寄存器和端口寄存器

在将 RTC1HZ 输出引脚的复用端口用 1Hz 输出时，必须将各端口对应的端口模式控制寄存器 (PMCxx) 的位、端口模式寄存器 (PMxx) 的位和端口寄存器 (Pxx) 的位置“0”。

设置的端口模式寄存器 (PMxx)、端口寄存器 (Pxx) 和端口模式控制寄存器 (PMCxx) 因产品而不同。详细内容请参照“2.3 控制端口功能的寄存器”。

## 7.4 实时时钟的运行

### 7.4.1 实时时钟的运行开始

图7-2: 实时时钟的运行开始步骤



注1: 必须先在计数时钟 ( $F_{RTC}$ ) 振荡稳定的状态下将RTCEN位置“1”。

注2: 这只是需要校正时钟误差的情况。有关校正值的计算方法, 请参照“7.4.6实时时钟的时钟误差校正例子”。

注3: 在RTCE位为“1”后不等待INTRTC位变为“1”而转移到睡眠模式的情况下, 请确认“7.4.2开始运行后睡眠模式的转移”的步骤。

## 7.4.2 开始运行后睡眠模式的转移

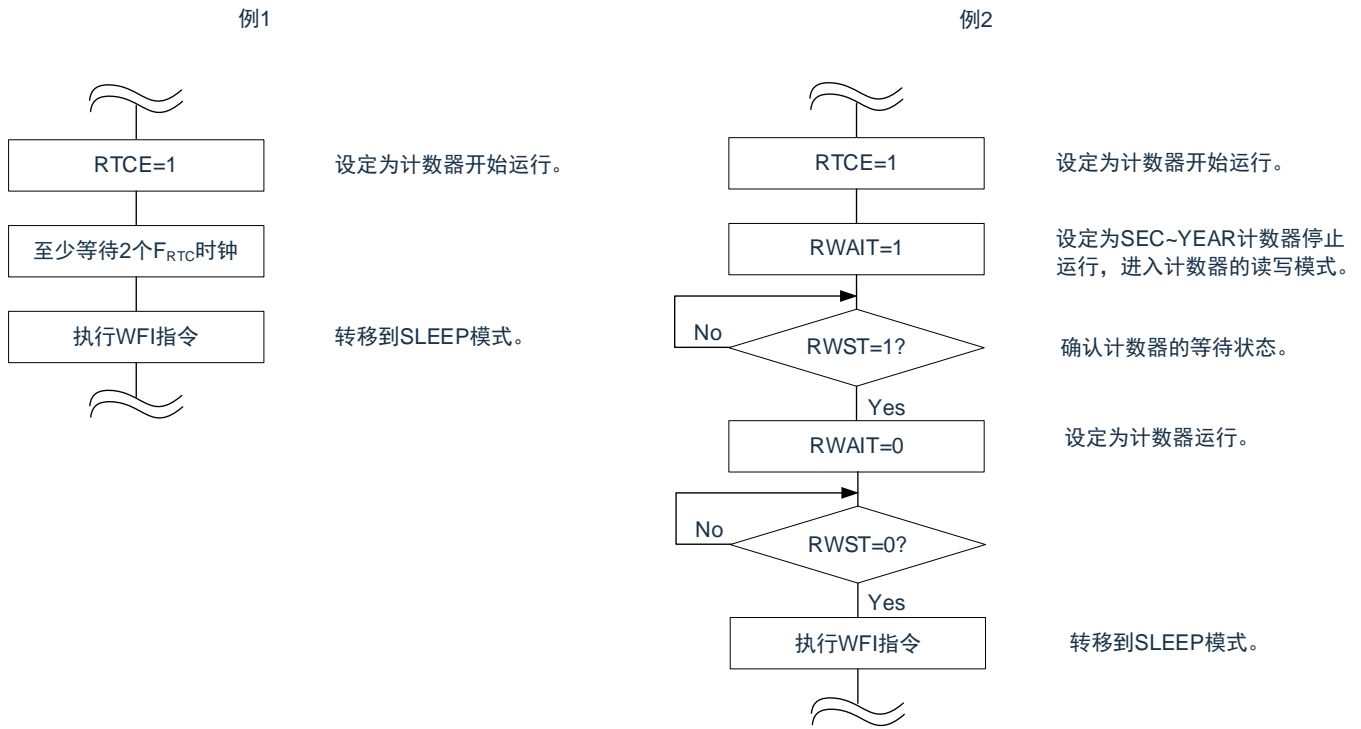
要在将 RTCE 位置“1”后立即转移到睡眠(包括深度睡眠)模式时, 必须进行以下某种处理。

但是, 在将 RTCE 位置“1”后, 如果要在发生 INTRTC 中断后转移到睡眠模式, 就不需要这些处理。

在将 RTCE 位置“1”后至少经过 2 个计数时钟 ( $F_{RTC}$ ) 的时间之后转移到睡眠模式 (参照图 7-3 的例 1)。

在将 RTCE 位置“1”后将 RWAIT 位置“1”, 通过轮询确认 RWST 位变为“1”。然后, 将 RWAIT 位置“0”并且再次通过轮询确认 RWST 位变为“0”, 然后转移到睡眠模式 (参照图 7-3 的例 2)。

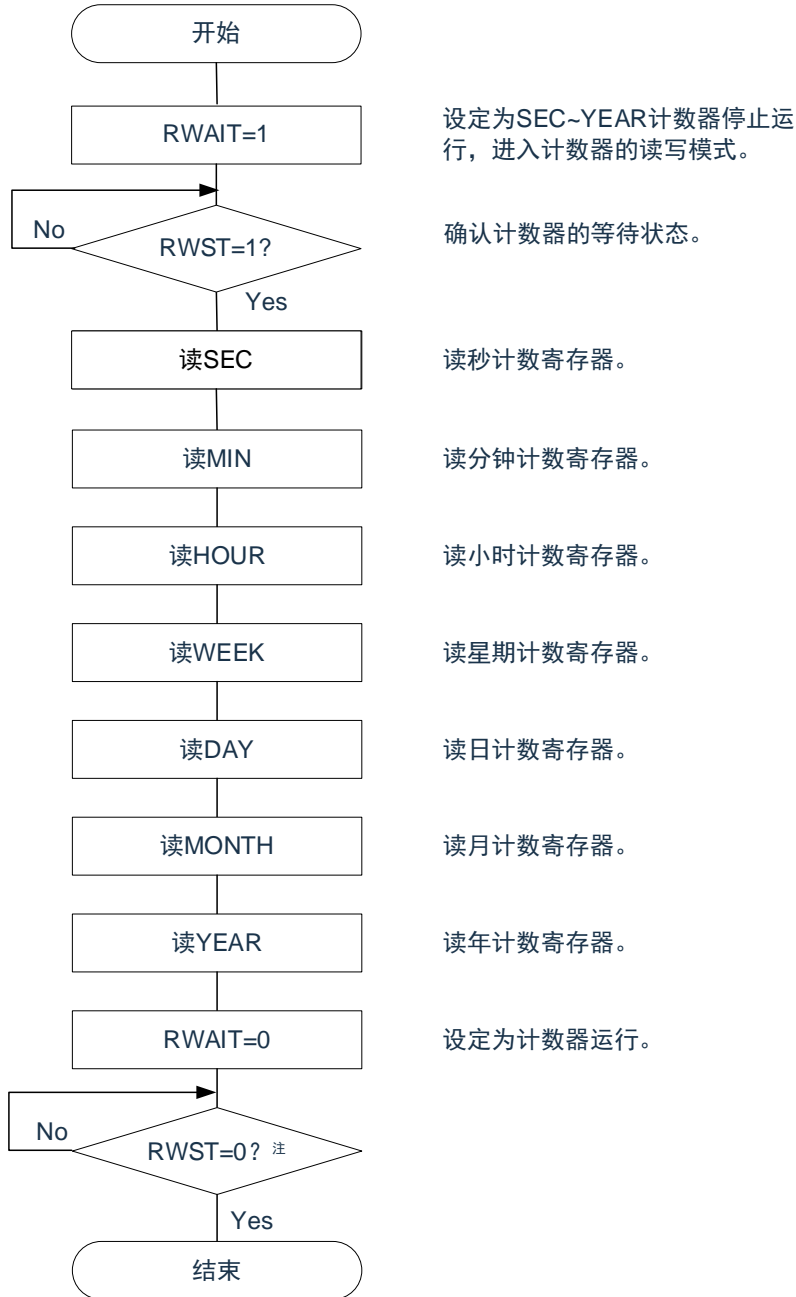
图7-3: 将RTCE位置“1”后的睡眠/深度睡眠模式的转移步骤



### 7.4.3 实时时钟计数器的读写

必须先将 RWAIT 位置“1”，然后读写计数器。必须在读写计数器后将 RWAIT 位置“0”。

图7-4：实时时钟计数器的读操作步骤



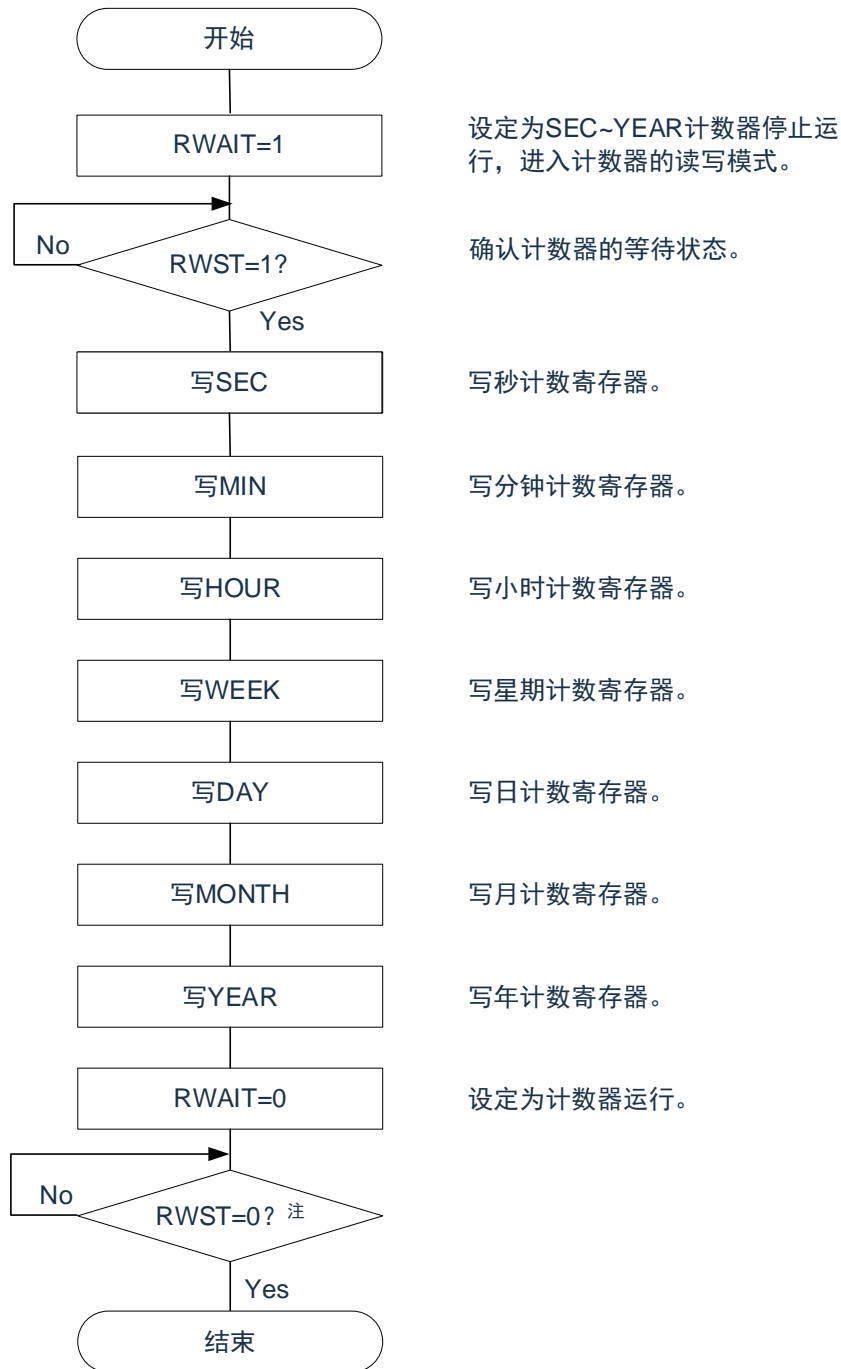
注：必须在转移到睡眠模式前，确认RWST位为“0”。

注意：

1. 必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。
2. 不限制秒/分钟/小时/星期/日/月/和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。



图7-5: 实时时钟计数器的读操作步骤



注1: 必须在转移到SLEEP模式前，确认RWST位为“0”。

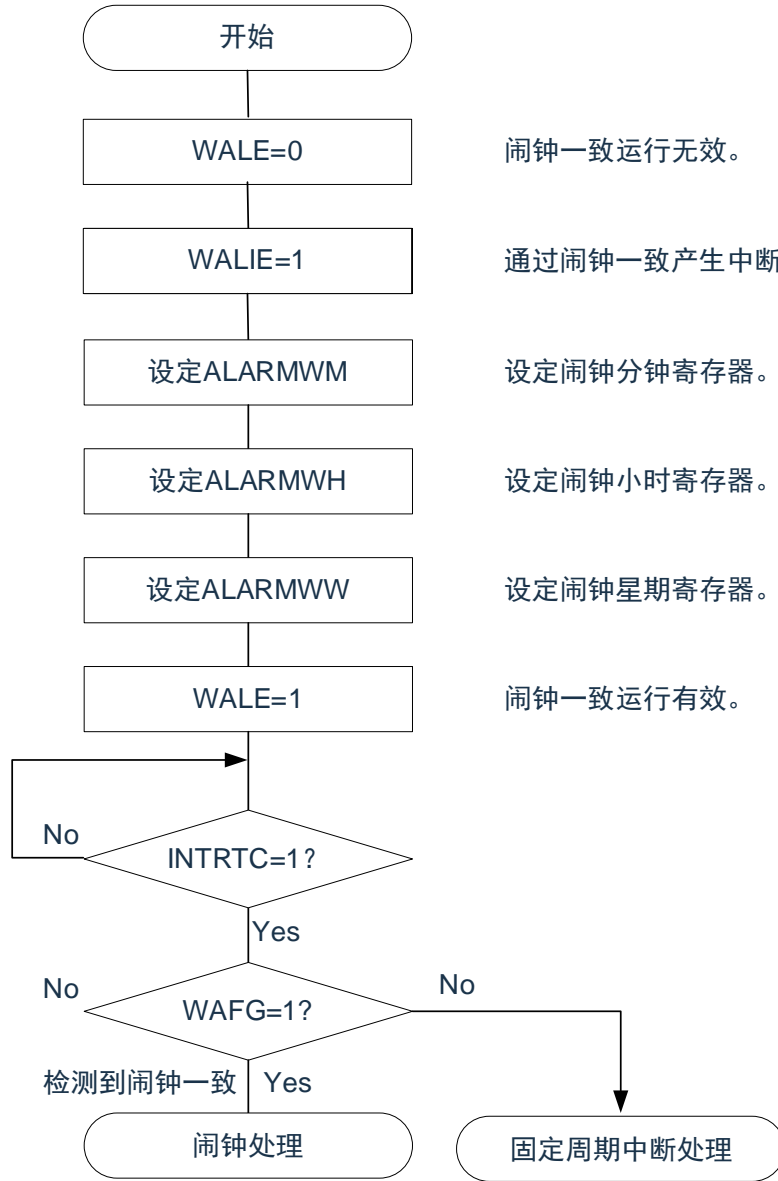
注意:

1. 必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。
2. 要在计数器运行中 (RTCE=1) 改写SEC、MIN、HOUR、WEEK、DAY、MONTH、YEAR寄存器时，必须在通过中断屏蔽标志寄存器将INTRTC设定为禁止中断处理后进行改写，并且必须在改写后清除WAFG标志、RIFG标志和RTCIF标志。
3. 不限制秒/分钟/小时/星期/日/月/和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。

### 7.4.4 实时时钟的闹钟设定

必须先将 WALE 位置“0”（闹钟运行无效），然后设定闹钟时间。

图7-6: 闹钟设定步骤



注意:

1. 不限制闹钟分钟寄存器（ALARMWWM）、闹钟小时寄存器（ALARMWH）和闹钟星期寄存器（ALARMWW）的写操作顺序。
2. 固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这2个中断的情况下，能在发生INTRTC时通过确认固定周期中断状态标志（RIFG）和闹钟检测状态标志（WAFG）来判断发生的是哪个中断。

### 7.4.5 实时时钟的1Hz输出

图7-7: 1Hz输出的设定步骤



注意：必须先在此计数时钟（ $F_{SUB}$ ）振荡稳定的状态下将RTCEEN位置“1”。

## 7.4.6 实时时钟的时钟误差校正例子

能通过给时钟误差校正寄存器设定值进行高精度的时钟快慢校正。

### 校正值的计算方法的例子

校正内部计数器（16位）的计数值时的校正值能用以下计算式进行计算。当校正范围超出-4165.6ppm~4165.6ppm的范围时，必须将DEV位置“0”。

（DEV=0的情况）

校正值<sup>注</sup>=1分钟的校正计数值÷3=(振荡频率÷目标频率-1)×32768×60÷3

（DEV=1的情况）

校正值<sup>注</sup>=1分钟的校正计数值=(振荡频率÷目标频率-1)×32768×60

注：校正值是根据时钟误差校正寄存器（SUBCUD）的bit12~0的值计算的时钟误差校正值。

（F12=0的情况）校正值={ (F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) - 1 } × 2

（F12=1的情况）校正值= -{ ( /F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0 ) + 1 } × 2

当(F12~F0)=(\*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \*)时，不进行时钟误差的校正。\*是“0”或者“1”。

/F12~/F0是各位取反后的值（“000000000011”时，为“111111111100”）。

备注：

1. 校正值为2、4、6、8、……、8186、8188或者-2、-4、-6、-8、……、-8186、-8188。
2. 振荡频率是计数时钟（F<sub>RTC</sub>）的值，能用以下计算式进行计算：时钟误差校正寄存器为初始值（“00H”）时的RTC1HZ引脚的输出频率×32768
3. 目标频率是使用时钟误差校正寄存器进行校正后的频率。

校正例子

从 32767.4Hz 校正到 32768Hz (32767.4Hz+18.3ppm) 的例子

**【振荡频率的测量】**

在时钟误差校正寄存器 (SUBCUD) 为初始值 (“0000H”) 时, 通过从 RTC1HZ 引脚输出大约 1Hz 的信号来测量各产品的振荡频率<sup>注</sup>。

注: 有关RTC1Hz输出的设定步骤, 请参照“7.4.5实时时钟的1Hz输出”。

**【校正值的计算】**

(RTC1HZ 引脚的输出频率为 0.9999817Hz 的情况)

振荡频率=32768×0.9999817≈32767.4Hz

假设目标频率为 32768Hz (32767.4Hz+18.3ppm) 并且 DEV=1。适用 DEV 位为“1”时的校正值的计算式。

校正值=1 分钟的校正计数值=(振荡频率÷目标频率-1)×32768×60=(32767.4÷32768-1)× 32768×60= -36

**【(F12~F0) 的设定值的计算】**

(校正值= -36 的情况)

因为校正值小于 0 (变快的情况), 所以 F12=1。根据校正值计算(F11~F0)。

$$-\{(/F11~/F0)-1\} \times 2 = -36$$

$$(/F11~/F0) = 17$$

$$(/F11~/F0) = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)$$

$$(F11~/F0) = (1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)$$

因此, 从 32767.4Hz 校正到 32768Hz (32767.4Hz+18.3ppm) 的情况如下:

如果通过 DEV=1 和校正值= -36 (SUBCUD 寄存器的 bit12~0: 1,1,1,1,1,1,1,1,0,1,1,1,0) 来设定校正寄存器, 就能校正到 32768Hz (0ppm)。

# 第8章 15位间隔定时器

## 8.1 15位间隔定时器的功能

以事先设定的任意时间间隔产生中断（INTIT），能用于从深度睡眠模式的唤醒。

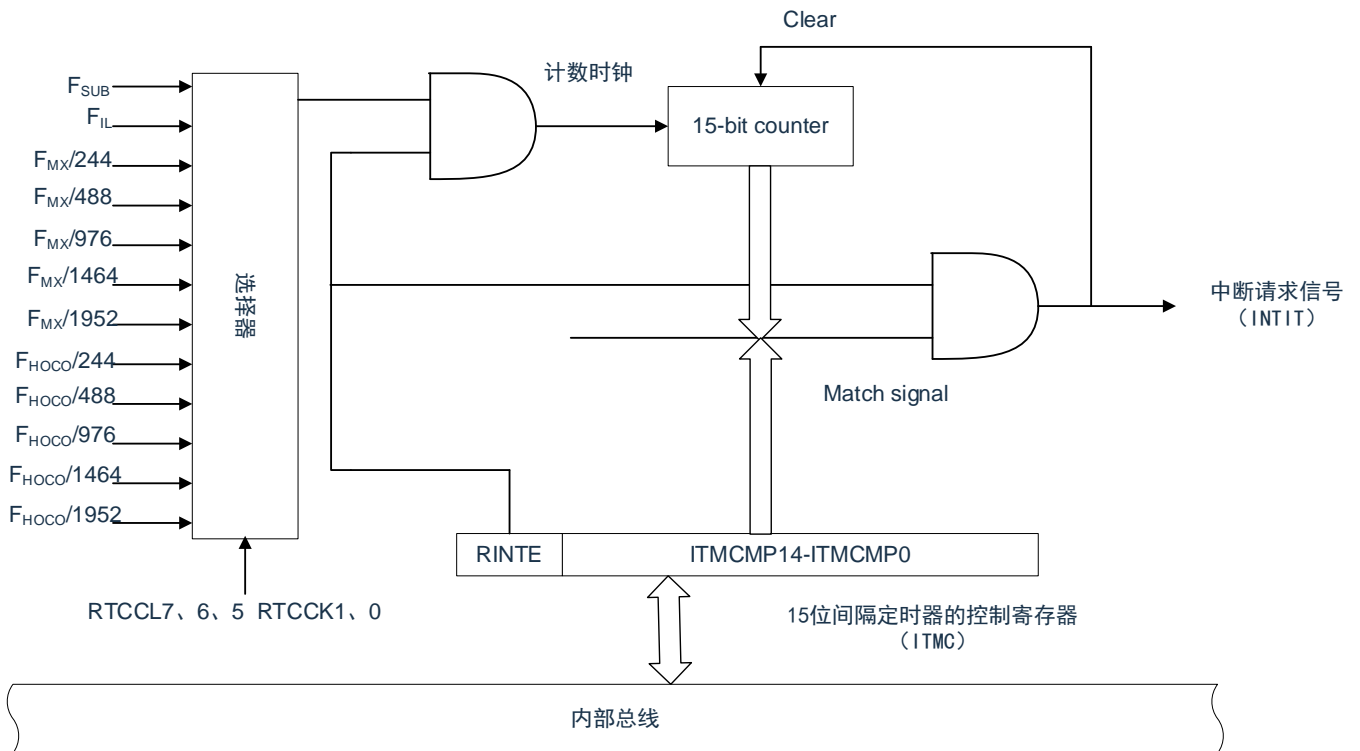
## 8.2 15位间隔定时器的结构

15 位间隔定时器由以下硬件构成。

表8-1：15位间隔定时器的结构

项目	结构
计数器	15位计数器
控制寄存器	外围允许寄存器0（PER0）
	实时时钟选择寄存器（RTCCL）
	15位间隔定时器的控制寄存器（ITMC）

图8-1：15位间隔定时器的框图



## 8.3 控制15位间隔定时器的寄存器

通过以下寄存器控制 15 位间隔定时器：

外围允许寄存器 0（PER0）

实时时钟选择寄存器（RTCCL）

15 位间隔定时器的控制寄存器（ITMC）

### 8.3.1 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 15 位间隔定时器时，必须将 bit7（RTCEN）置“1”。通过 8 位存储器操作指令设定 PER0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表8-2：外围允许寄存器0（PER0）的格式

地址：0x40020420    复位后：00H    RW

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICA0EN	SCI1EN	SCI0EN	TM41EN	TM40EN

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

### 8.3.2 实时时钟选择寄存器 (RTCCL)

能通过 RTCCL 选择实时时钟和 15 位间隔定时器的计数时钟 ( $F_{RTC}$ )。

表8-3: 实时时钟选择寄存器 (RTCCL) 的格式

地址: 0x4004047C      复位后: 00H      R/W

符号	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCKS1	RTCKS0

RTCCL7	实时时钟、15位间隔定时器的计数时钟的时钟源的选择
0	选择高速系统时钟( $F_{MX}$ )
1	选择高速内部振荡器( $F_{HOCO}$ )

RTCKS1	RTCKS0	RTCCL6	RTCCL5	实时时钟、15位间隔定时器的计数时钟的运行时钟的选择
0	0	x	x	副系统时钟 ( $F_{SUB}$ )
0	1			低速内部振荡器时钟 ( $F_{IL}$ ) (必须设置WUTMMCK0为1)
1	0	0	1	主时钟 $F_{MAX}/F_{HOCO}$ (通过RTCCL7选择)/1952
1	0	0	0	主时钟 $F_{MAX}/F_{HOCO}$ (通过RTCCL7选择)/1464
1	0	1	0	主时钟 $F_{MAX}/F_{HOCO}$ (通过RTCCL7选择)/976
1	1	0	0	主时钟 $F_{MAX}/F_{HOCO}$ (通过RTCCL7选择)/488
1	1	1	0	主时钟 $F_{MAX}/F_{HOCO}$ (通过RTCCL7选择)/244





## 8.4 15位间隔定时器的运行

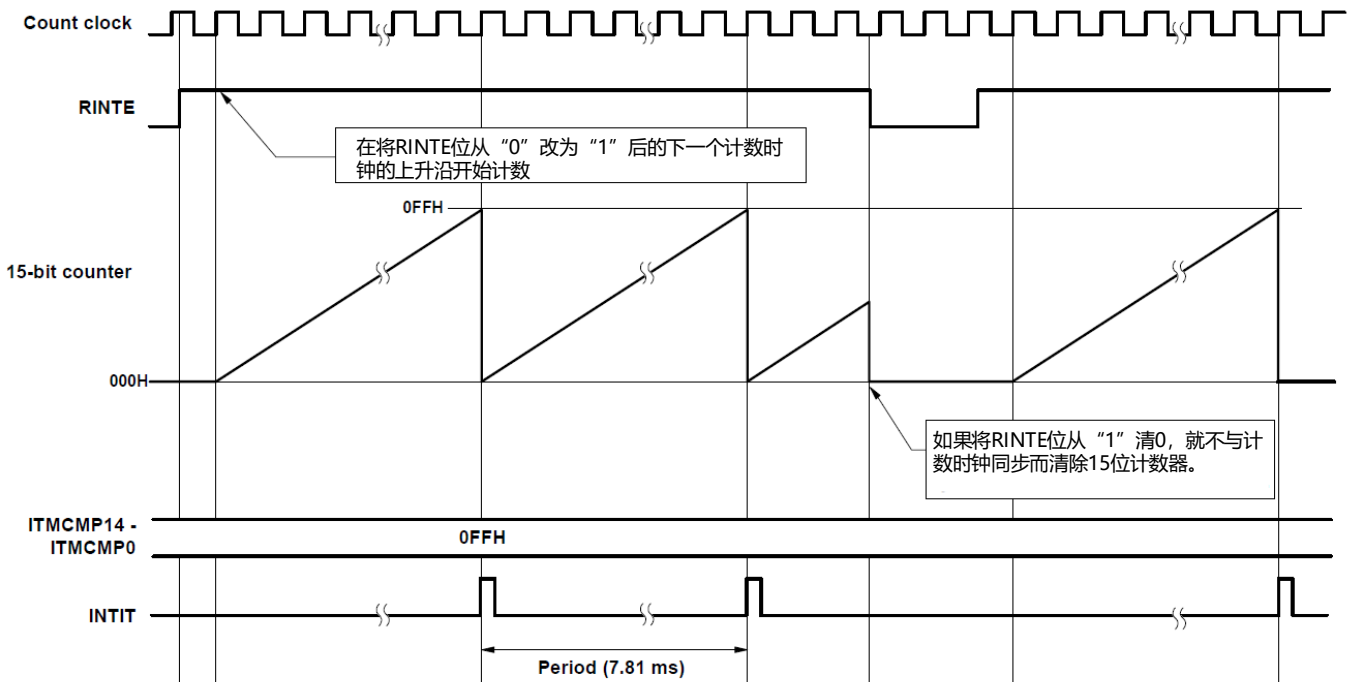
### 8.4.1 15位间隔定时器的运行时序

以 ITCMP14~ITCMP0 位设定的计数值为间隔，作为重复产生中断请求（INTIT）的 15 位间隔定时器运行。如果将 RINTE 位置“1”，15 位计数器就开始计数。

当 15 位计数值和 ITCMP14~ITCMP0 位的设定值相同时，将 15 位计数值清“0”并且继续计数，同时产生中断请求信号（INTIT）。

15 位间隔定时器的基本运行如图 8-2 所示。

图8-2：15位间隔定时器的运行时序  
(ITCMP14~ITCMP0=0FFH, 计数时钟:  $F_{SUB}=32.768\text{KHz}$ )

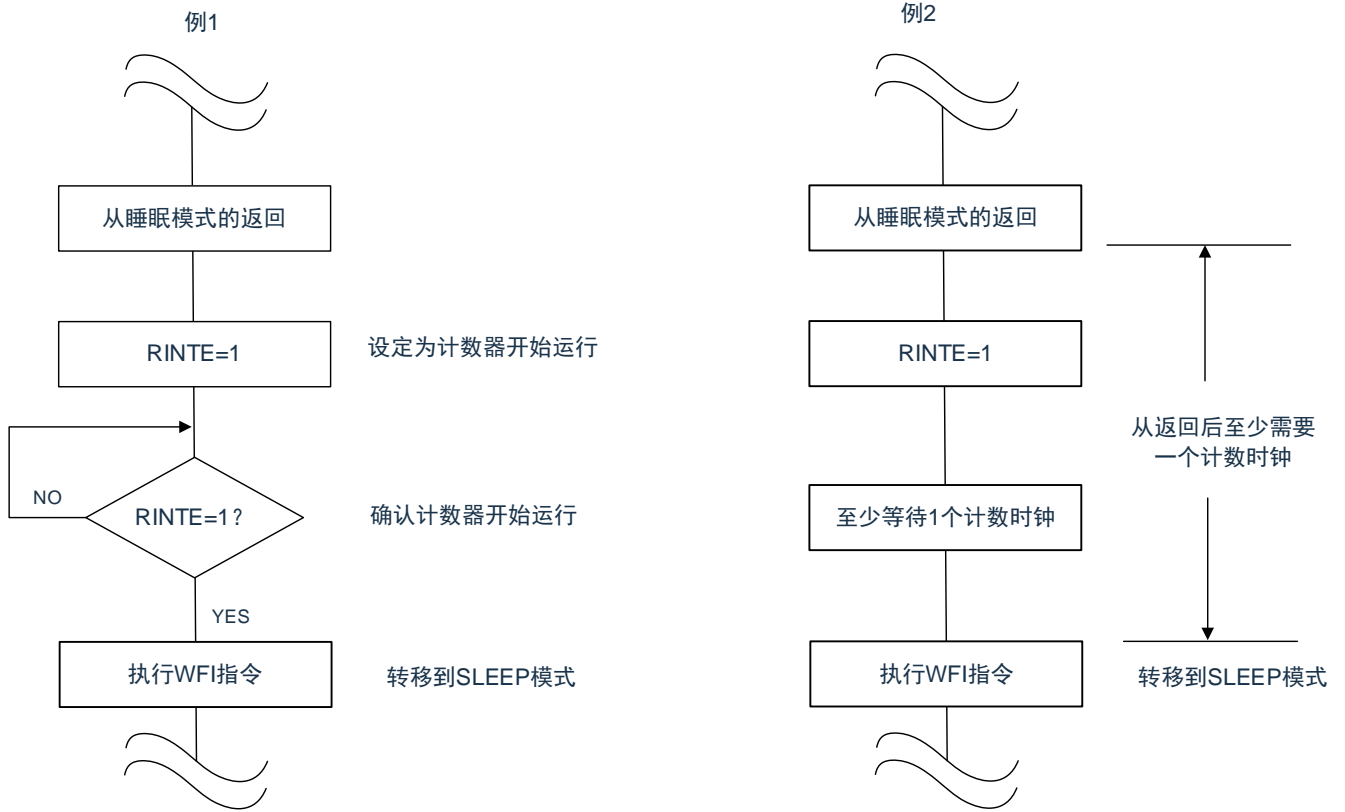


### 8.4.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

在从睡眠模式返回后，如果要将 RINTE 位置“1”并且再次转移到睡眠模式，就必须在将 RINTE 位置“1”后确认 RINTE 位的写入值被反映，或者在返回后至少经过 1 个计数时钟的时间之后再转移到睡眠模式。

在将 RINTE 位置“1”后，通过轮询确认 RINTE 位变为“1”，然后转移到睡眠模式（参照下图的例 1）。

在将 RINTE 位置“1”后至少经过 1 个计数时钟的时间之后转移到睡眠模式（参照下图的例 2）。



# 第9章 时钟输出/蜂鸣器输出控制电路

## 9.1 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。

本产品有两个时钟输出/蜂鸣器输出引脚，能选择从RESETB以外的任意引脚用作时钟输出或者蜂鸣器输出。

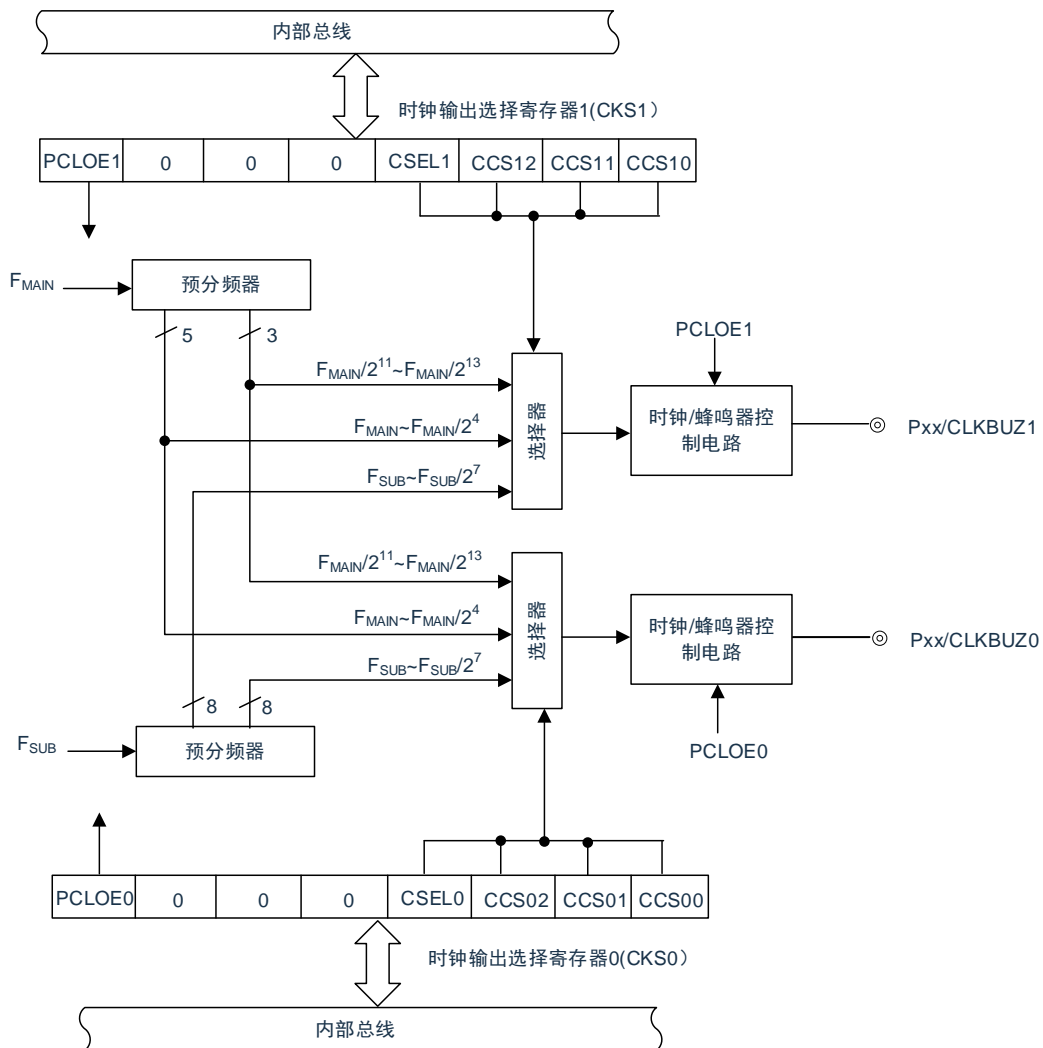
CLKBUZn引脚输出由时钟输出选择寄存器n（CKSn）选择的时钟。

时钟输出/蜂鸣器输出控制电路的框图如图9-1所示。

注意：在副系统时钟提供模式控制寄存器（OSMC）的RTCLPC位为“1”时并且在CPU以副系统时钟（F<sub>SUB</sub>）运行的SLEEP模式中，不能从CLKBUZn引脚输出副系统时钟（F<sub>SUB</sub>）。

备注：n=0、1。

图9-1：时钟输出/蜂鸣器输出控制电路的框图



备注：有关能从CLKBUZ0引脚和CLKBUZ1引脚输出的频率，请参照“数据手册的AC特性”。

## 9.2 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9-1：时钟输出/蜂鸣器输出控制电路的结构

项目	结构
控制寄存器	时钟输出选择寄存器n (CKSn) 端口模式控制寄存器 (PMCxx)、端口模式寄存器 (PMxx)、 端口复用控制寄存器 (PxxCFG)

## 9.3 控制时钟输出/蜂鸣器输出控制电路的寄存器

### 9.3.1 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 (CLKBUZn) 的输出以及设定输出时钟的寄存器。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定CKSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表9-2: 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
0	0	0	0	F <sub>MAIN</sub>
0	0	0	1	F <sub>MAIN</sub> /2
0	0	1	0	F <sub>MAIN</sub> /2 <sup>2</sup>
0	0	1	1	F <sub>MAIN</sub> /2 <sup>3</sup>
0	1	0	0	F <sub>MAIN</sub> /2 <sup>4</sup>
0	1	0	1	F <sub>MAIN</sub> /2 <sup>11</sup>
0	1	1	0	F <sub>MAIN</sub> /2 <sup>12</sup>
0	1	1	1	F <sub>MAIN</sub> /2 <sup>13</sup>
1	0	0	0	F <sub>SUB</sub>
1	0	0	1	F <sub>SUB</sub> /2
1	0	1	0	F <sub>SUB</sub> /2 <sup>2</sup>
1	0	1	1	F <sub>SUB</sub> /2 <sup>3</sup>
1	1	0	0	F <sub>SUB</sub> /2 <sup>4</sup>
1	1	0	1	F <sub>SUB</sub> /2 <sup>5</sup>
1	1	1	0	F <sub>SUB</sub> /2 <sup>6</sup>
1	1	1	1	F <sub>SUB</sub> /2 <sup>7</sup>

注：必须在16MHz以内的范围内使用输出时钟。详细内容请参照“数据手册的AC特性”。

注意：

1. 输出时钟的切换必须在设定为禁止输出 (PCLOEn=0)后进行。
2. 在选择主系统时钟 (CSELn=0) 时，如果要转移到深度睡眠模式，就必须在执行WFI指令前将PCLOEn置“0”；在选择副系统时钟 (CSELn=1) 时，因为能在副系统时钟提供模式控制寄存器 (OSMC) 的RTCLPC位为“0”时并且在深度睡眠模式中输出时钟，所以能将PCLOEn置“1”。
3. 在副系统时钟提供模式控制寄存器 (OSMC) 的RTCLPC位为“1”时并且在CPU以副系统时钟 (F<sub>SUB</sub>) 运行的睡眠模式中，不能从CLKBUZn引脚输出副系统时钟 (F<sub>SUB</sub>)。

备注：

1. n=0、1。
2. F<sub>MAIN</sub>：主系统时钟频率。  
F<sub>SUB</sub>：副系统时钟频率。

### 9.3.2 控制时钟输出/蜂鸣器输出引脚端口功能的寄存器

本产品可以将时钟输出/蜂鸣器输出功能CLKBUZ0/ CLKBUZ1复用到除RESETB以外的任意端口。使用时钟输出/蜂鸣器输出功能时，必须设定端口复用功能配置寄存器（PxxCFG），端口寄存器（Pxx），端口模式寄存器（PMxx）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”。

被配置为时钟输出/蜂鸣器输出引脚的复用端口，其对应的端口寄存器（Pxx），端口模式寄存器（PMxx）的位和端口模式控制寄存器（PMCxx）的位必须置“0”。

（例）将P20用作时钟输出/蜂鸣器输出（CLKBUZ0）的情况：

- (1) 将端口寄存器2的P20位置“0”。
- (2) 将端口模式寄存器2的PM20位置“0”。
- (3) 将端口模式控制寄存器2的PMC20位置“0”。
- (4) 将端口复用功能配置寄存器P20CFG置“0x18”。

（例）将P15用作时钟输出/蜂鸣器输出（CLKBUZ1）的情况：

- (1) 将端口寄存器1的P15位置“0”。
- (2) 将端口模式寄存器1的PM15位置“0”。
- (3) 将端口模式控制寄存器1的PMC15位置“0”。
- (4) 将端口复用功能配置寄存器P15CFG置“0x19”。

## 9.4 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。

### 9.4.1 输出引脚的运行

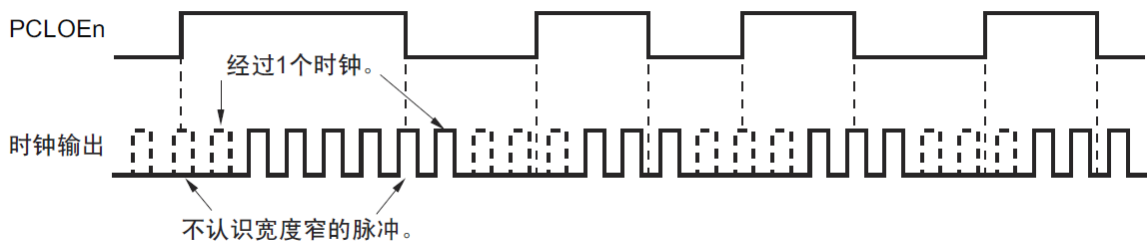
CLKBUZn引脚按照以下步骤进行输出：

- ① 设定端口复用功能配置寄存器 (PxxCFG),将用作CLKBUZ0引脚的端口对应的端口寄存器 (Pxx), 端口模式寄存器 (PMxx) 和端口模式控制寄存器 (PMCxx) 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9-2所示。
2. n=0、1。

图9-2: CLKBUZn引脚的时钟输出时序



## 9.5 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, 如果在设定停止输出 (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10章 看门狗定时器

## 10.1 看门狗定时器的功能

看门狗定时器通过选项字节（000C0H）设定计数运行。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。

- (1) 当看门狗定时器的计数器发生上溢时
- (2) 当给WDTE寄存器写“ACH”以外的数据时
- (3) 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19章 复位功能”。当达到上溢时间的75%+1/2  $F_{IL}$ 时，能产生间隔中断。

## 10.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表10-1：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）

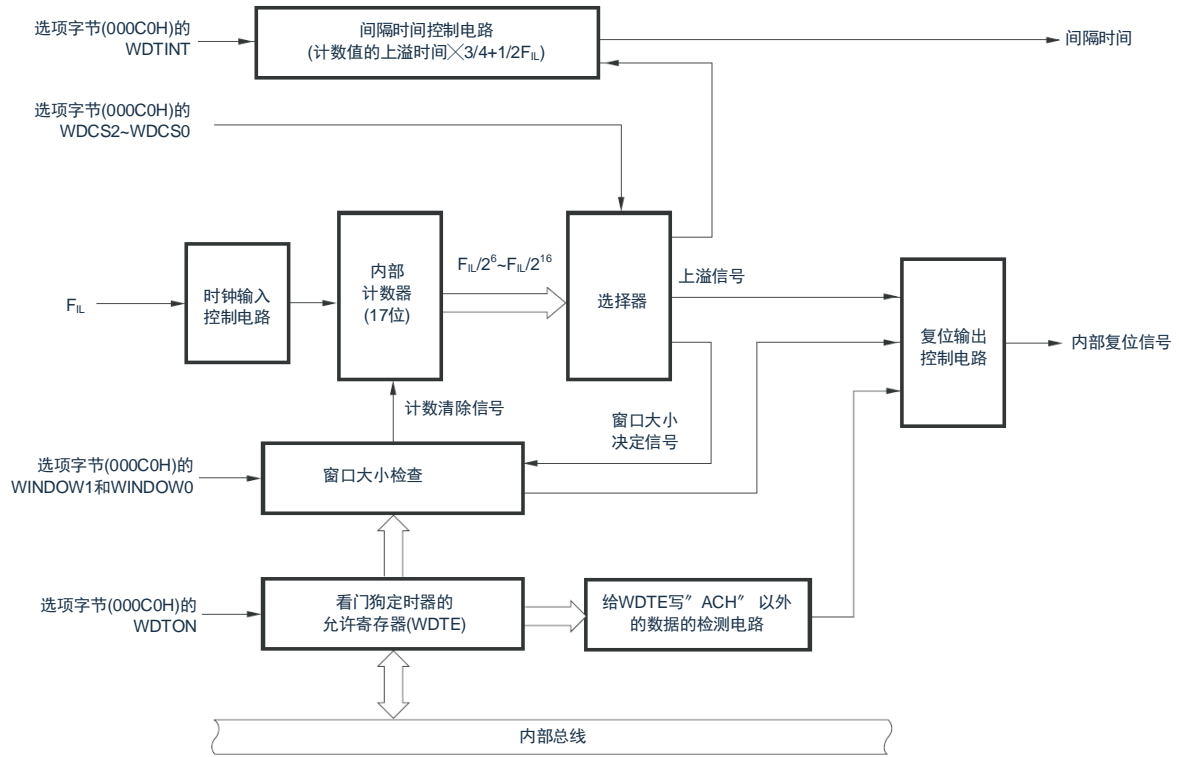
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表10-2：选项字节和看门狗定时器的设定内容

看门狗定时器的设定内容	选项字节（000C0H）
看门狗定时器的间隔中断的设定	bit7（WDTINT）
窗口打开期间的设定	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定	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24章 选项字节”。

图10-1：看门狗定时器的框图



备注： $F_{IL}$ ：低速内部振荡器的时钟频率

## 10.3 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。

### 10.3.1 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定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>。

表10-3：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					
符号	7	6	5	4	3	2	1 0
WDTE							

注：WDTE寄存器的复位值因选项字节（000C0H）的WDTON位的设定值而不同。要使看门狗定时器运行时，必须将WDTON位置“1”。

WDTON位的设定值	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

备注：

1. 当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。
2. WDTE寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

## 10.3.2 LOCKUP控制寄存器（LOCKCTL）及其保护寄存器（PRCR）

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，PRCR是其写保护寄存器。

通过8位存储器操作指令设置LOCKCTL，PRCR寄存器。

在产生复位信号后，LOCKCTL寄存器的值变为“01H”，PRCR寄存器的值变为“00H”。

表10-4：LOCKUP控制寄存器（LOCKCTL）及其保护寄存器（PRCR）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

表10-4：LOCKUP控制寄存器（LOCKCTL）及其保护寄存器（PRCR）的格式（2/2）

地址：40020406H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
PRCR	PRTKEY[7:1]							PRCR

PRCR	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]	PRCR的写保护
78H	• PRCR可写
其它	• PRCR不可写

### 10.3.3 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”。

表10-5: WDTCFG配置寄存器 (WDTCFG0/1/2/3)

地址: 4002040CH	复位后: 00H	WO						
符号	7	6	5	4	3	2	1	0
WDTCFG0	WDTCFG0							

地址: 4002040DH	复位后: 00H	WO						
符号	7	6	5	4	3	2	1	0
WDTCFG1	WDTCFG1							

地址: 4002040EH	复位后: 00H	WO						
符号	7	6	5	4	3	2	1	0
WDTCFG2	WDTCFG2							

地址: 4002040FH	复位后: 00H	WO						
符号	7	6	5	4	3	2	1	0
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</sup>
其他				复位后强制运行看门狗定时器

注1: 详细配置参考24.2用户选项字节的格式。

## 10.4 看门狗定时器的运行

### 10.4.1 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）设定以下内容：

- (1) 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24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）。

- (2) 必须通过选项字节（000C0H）的bit3~1（WDCS2~WDCS0）设定上溢时间（详细内容请参照10.4.2和第24章）。

- (3) 必须通过选项字节（000C0H）的bit6和bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照10.4.2和第24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：

- (1) 当给WDTE寄存器写“ACH”以外的数据时

备注：

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<sub>IL</sub>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）的设定值而不同。

	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		

5. 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。

6. 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。

7. 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，就需要立刻清除看门狗定时器。因为在经过振荡稳定时间后才清除看门狗定时器，很有可能发生看门狗定时器上溢而产生复位的情况。所以必须考虑这种情况进行上溢时间的设定。

## 10.4.2 看门狗定时器上溢时间的设定

通过选项字节（000C0H）的bit3~1（WDCS2~WDCS0）设定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。能设定的上溢时间如下所示。

表10-6：看门狗定时器上溢时间的设定

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	$2^6/F_{IL}$ (3.2ms)
0	0	1	$2^7/F_{IL}$ (6.4ms)
0	1	0	$2^8/F_{IL}$ (12.8ms)
0	1	1	$2^9/F_{IL}$ (25.6ms)
1	0	0	$2^{11}/F_{IL}$ (102.4ms)
1	0	1	$2^{13}/F_{IL}$ (409.6ms)
1	1	0	$2^{14}/F_{IL}$ (819.2ms)
1	1	1	$2^{16}/F_{IL}$ (3276.8ms)

备注： $F_{IL}$ ：低速内部振荡器的时钟频率

### 10.4.3 看门狗定时器窗口打开期间的设定

通过选项字节（000C0H）的bit6和bit5（WINDOW1、WINDOW0）设定看门狗定时器的窗口打开期间。窗口概要如下：

- (1) 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。
- (2) 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。

备注：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。

能设定的窗口打开期间如下所示。

表10-7：看门狗定时器窗口打开期间的设定

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	-	禁止设定
1	0	75%
1	1	100%

备注：

1. 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%。
2. 当将上溢时间设定为 $2^9/F_{IL}$ 的情况时，窗口关闭时间和打开时间如下所示。

	窗口打开期间的设定	
	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>

(1) 上溢时间：

$$2^9/F_{IL} (\text{Max.}) = 2^9/20\text{KHz} (\text{Max.}) = 25.6\text{ms}$$

(2) 窗口关闭时间：

$$0 \sim 2^9/F_{IL} (\text{Min.}) \times (1-0.75) = 0 \sim 2^9/10\text{KHz} \times 0.25 = 0 \sim 12.8\text{ms}$$

(3) 窗口打开时间：

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



## 10.4.4 看门狗定时器间隔中断的设定

能通过设定选项字节（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10-8：看门狗定时器间隔中断的设定

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。

备注：

1. 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，就需要立刻清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，很有可能发生看门狗定时器上溢而产生复位的情况。所以必须考虑这种情况进行上溢时间的设定。
2. 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。

## 10.4.5 LOCKUP期间看门狗定时器的运行

当LOCKUP控制寄存器LOCKCTL的lockup\_rst位设置为1时，一旦内核进入LOCKUP状态，低速内部振荡器开始发振，看门狗定时器的计时器自动开始运行，并将上溢时间的控制位（WDCS2~WDCS0）设置为3'b010，即设置上溢时间为12.8ms。

## 10.4.6 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11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，详细管脚参考对应产品数据手册。

## 11.1 A/D转换器的功能

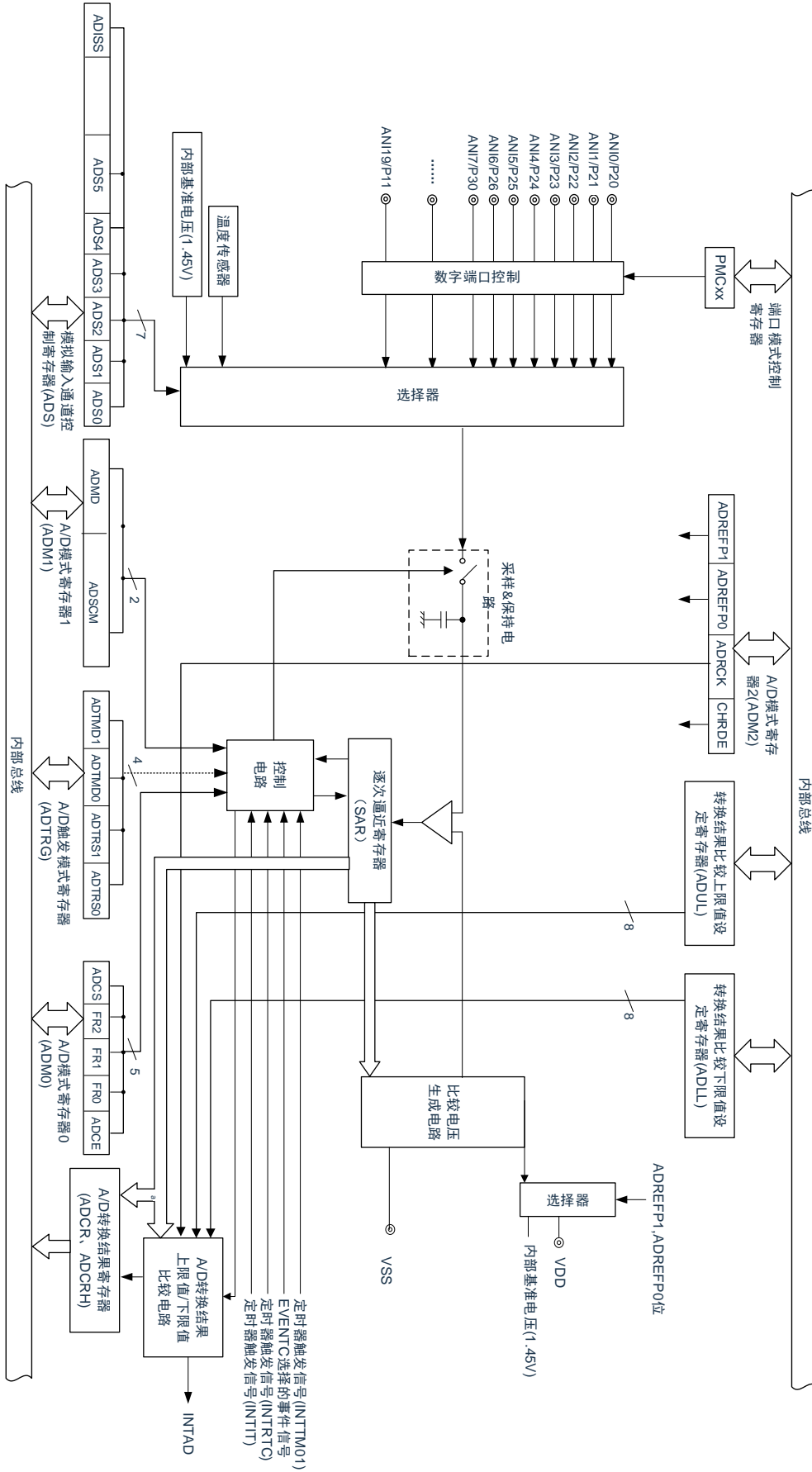
A/D转换器是将模拟输入转换为数字值的转换器，A/D转换器有以下的功能。

- (1) 12位分辨率的A/D转换
- (2) 从ANI0~ANI19、V内部基准电压和温度传感器中选择1个通道的模拟输入，重复进行12位分辨率的A/D转换。每结束1次A/D转换，就产生中断请求（INTAD）（选择模式的情况）。

能通过下述的模式组合设定各种 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 A/D 电源的转换待机状态下，通过检测硬件触发来接通电源，在经过 A/D 电源稳定等待时间后自动开始转换。
通道选择模式	选择模式	选择 1 个通道的模拟输入，进行 A/D 转换。
	扫描模式	按顺序对 4 个通道的模拟输入进行 A/D 转换。能选择 ANI0~ANI15 中连续的 4 个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行 1 次 A/D 转换。
	连续转换模式	对所选通道进行连续的 A/D 转换，直到被软件停止为止。
采样时间	采样时钟 4/8 个 ADCLK	采样时间可以通过 ADSMPWAIT 寄存器来选择，默认使用 4 个转换时（F <sub>AD</sub> ）。

图11-1: A/D转换器的框图



备注：模拟输入通道ANx的选择请参考11.2.6模拟输入通道指定寄存器（ADS）

## 11.2 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定寄存器	R/W	00H	ADC_BASE+0AH
ADUL	转换结果比较上限值设定寄存器	R/W	00H	ADC_BASE+0BH
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
ADSMPWAIT	A/D转换器的采样时间延长控制寄存器	R/W	00H	ADC_BASE+15H
PMCn	端口模式控制寄存器	R/W	注1	PORT_BASE+注1

R: readonly, W: writeonly, R/W: both read and write

备注：通过ADS寄存器选择某通道时，需要配置该通道引脚的PMC寄存器，将其配置为模拟通道。

## 11.2.1 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定PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-1：外围允许寄存器0 (PER0) 的格式

复位值：00H		R/W						
	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	TM41EN	TM40EN

ADCEN	A/D转换器的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> <li>不能写A/D转换器使用的SFR。</li> <li>A/D转换器处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写A/D转换器使用的SFR。</li> </ul>

备注：要设定A/D转换器时，必须先在ADCEN位为“1”的状态下读写以下的寄存器。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）。

## 11.2.2 A/D转换器的模式寄存器0 (ADM0)

用于设定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定ADM0寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

表11-2: A/D转换器的模式寄存器0 (ADM0) 的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADM0	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D转换运行的控制
0	停止转换运行。 [读时] 停止转换运行/待机状态
1	允许转换运行。 [读时] 软件触发模式时: 转换运行状态 硬件触发等待模式时: A/D电源等待稳定状态+转换运行状态

ADCE	A/D电压比较器的运行控制
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。

备注:

1. 有关FR2~FR0位和A/D转换的详细内容，请参照“表11-5 A/D转换时间的选择”。
2. A/D转换器开始运行需要2us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2us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2us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2us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的设定。
3. 禁止通过8位操作指令将ADCS=0、ADCE=0的状态设定为ADCS=1、ADCE=1。必须按照“11.5 A/D转换器的设定流程图”的步骤进行设定。

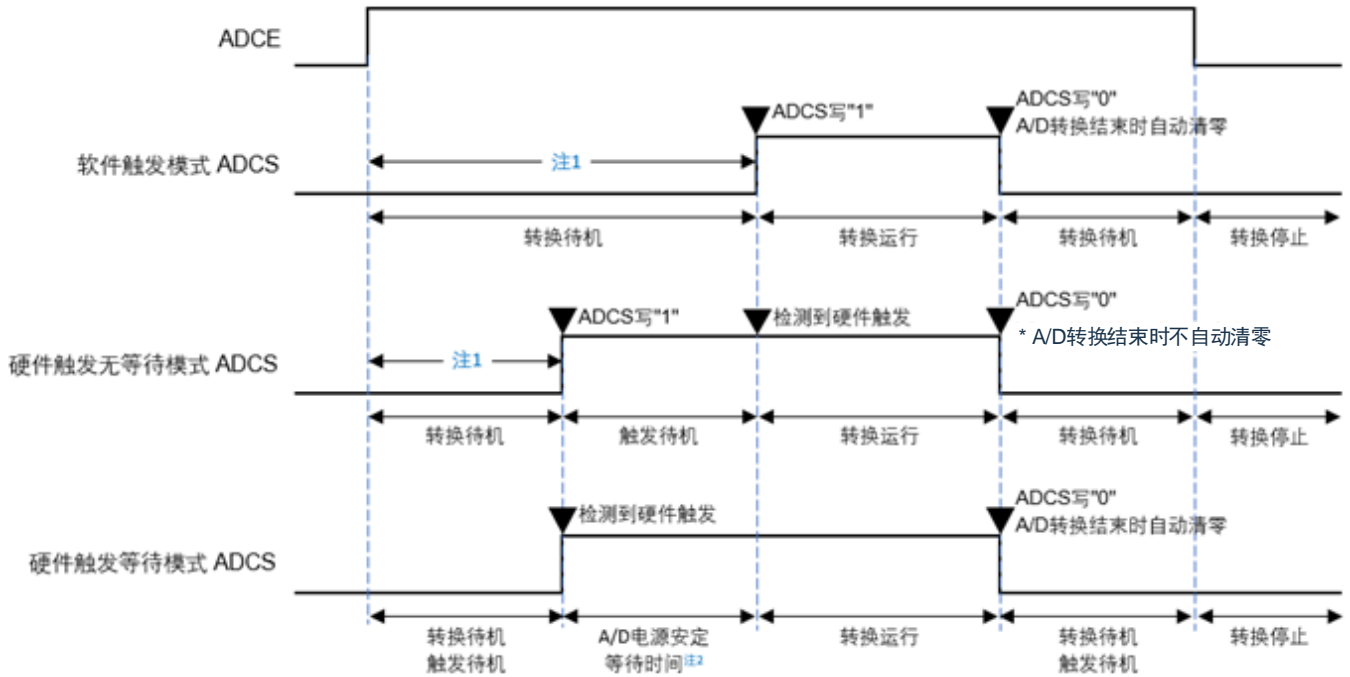
表11-3: ADCS位和ADCE位的设定

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	禁止设定。
1	1	转换运行状态

表11-4: 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“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 •当设定的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“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 •当设定的4个通道转换结束时，自动清“0”。

图11-2: 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2us (TBD)。

注2: 在硬件触发等待模式, A/D电源稳定时间1us由设计保证。

注意:

1. 要使用硬件触发等待模式时, 禁止将ADCS位置“1” (当检测到硬件触发信号时, 自动切换为“1”)。但是, 为了设定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定为以下时间:

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

(F<sub>CLK</sub>: CPU/外围硬件的时钟频率)



表11-5: 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采样时间延长寄存器 (ADSMPWAIT)	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	ADSMPWAIT		ADC转换时钟数	ADC转换时间
0	0	0	0	F <sub>CLK</sub> /32	16个ADCLK (采样时钟数4个+逐次比较时钟数12个)	16/F <sub>AD</sub>
0	0	1		F <sub>CLK</sub> /16		-
0	1	0		F <sub>CLK</sub> /8		-
0	1	1		F <sub>CLK</sub> /4		-
1	0	0		F <sub>CLK</sub> /2		-
1	0	1		F <sub>CLK</sub> /1		-
0	0	0	1	F <sub>CLK</sub> /32	20个ADCLK (采样时钟数8个+逐次比较时钟数12个)	20/F <sub>AD</sub>
0	0	1		F <sub>CLK</sub> /16		-
0	1	0		F <sub>CLK</sub> /8		-
0	1	1		F <sub>CLK</sub> /4		-
1	0	0		F <sub>CLK</sub> /2		-
1	0	1		F <sub>CLK</sub> /1		-

注意: 要将FR2~FR0位、ADSMPWAIT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: 进行一次ADC转换需要的时间 = (采样时钟数 + 逐次比较时钟数) / F<sub>AD</sub>, 其中采样时钟个数可通过ADSMPWAIT寄存器进行调整, 默认为4个ADCLK。ADCLK支持的最快时钟为8MHz。

F<sub>CLK</sub>: CPU/外围硬件的时钟频率

F<sub>AD</sub>: ADC转换时钟频率最快为8MHz。

表11-6: 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）

A/D转换器的模式寄存器0 (ADM0)			A/D采样时间延长寄存器 (ADSMPWAIT)	转换时钟ADCLK的频率 (F <sub>AD</sub> )	A/D电源稳定等待时间	ADC转换时钟数	A/D电源稳定等待时间+ADC转换时间
FR2	FR1	FR0	ADSMPWAIT				
0	0	0	0	F <sub>CLK</sub> /32	2us	16个ADCLK (采样时钟数4个+逐次比较时钟数12个)	2us + 16/F <sub>AD</sub>
0	0	1		F <sub>CLK</sub> /16			
0	1	0		F <sub>CLK</sub> /8			
0	1	1		F <sub>CLK</sub> /4			
1	0	0		F <sub>CLK</sub> /2			
1	0	1		F <sub>CLK</sub> /1			
0	0	0	1	F <sub>CLK</sub> /32	2us	20个ADCLK (采样时钟数8个+逐次比较时钟数16个)	2us + 20/F <sub>AD</sub>
0	0	1		F <sub>CLK</sub> /16			
0	1	0		F <sub>CLK</sub> /8			
0	1	1		F <sub>CLK</sub> /4			
1	0	0		F <sub>CLK</sub> /2			
1	0	1		F <sub>CLK</sub> /1			

## 备注:

1. 硬件触发等待模式时，电源稳定时间由硬件设计保证，不需要设置。且在连续转换模式时，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。
2. 硬件触发后ADC转换需要的时间 = 2us + (采样时钟数 + 逐次比较时钟数) / F<sub>AD</sub>，其中采样时钟个数可通过ADSMPWAIT寄存器进行调整，默认为4个ADCLK，ADCLK支持的最快时钟为8MHz。
3. 要将FR2~FR0位、ADSMPWAIT位改写为不同数据时，必须在转换停止状态 (ADCS=0) 下进行。
4. 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。  
(F<sub>CLK</sub>: CPU/外围硬件的时钟频率)

## 11.2.3 A/D转换器的模式寄存器1 (ADM1)

这是设定A/D转换模式的寄存器。

通过8位存储器操作指令设定ADM1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-7: A/D转换器的模式寄存器1 (ADM1) 的格式

复位值: 00H								
	7	6	5	4	3	2	1	0
ADM1	ADMD	0	0	0	ADSCM	0	0	0

ADMD	A/D转换通道选择模式的设定
0	选择模式
1	扫描模式

ADSCM	A/D转换模式的设定
0	连续转换模式
1	单次转换模式

注意:

1. 必须将bit6~4, 2置“0”。
2. 要改写ADM1寄存器时, 必须在转换停止状态 (ADCS=0) 下进行。
3. 为了正常结束A/D转换, 必须至少将硬件触发间隔设定为以下时间:  
 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  
 (F<sub>CLK</sub>: CPU/外围硬件的时钟频率)

### 11.2.4 A/D转换器的模式寄存器2 (ADM2)

通过8位存储器操作指令设定ADM2寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-8: A/D转换器的模式寄存器2 (ADM2) 的格式(1/3)

复位值: 00H

R/W

	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	0	0	ADRCK	0	CHRDE	0

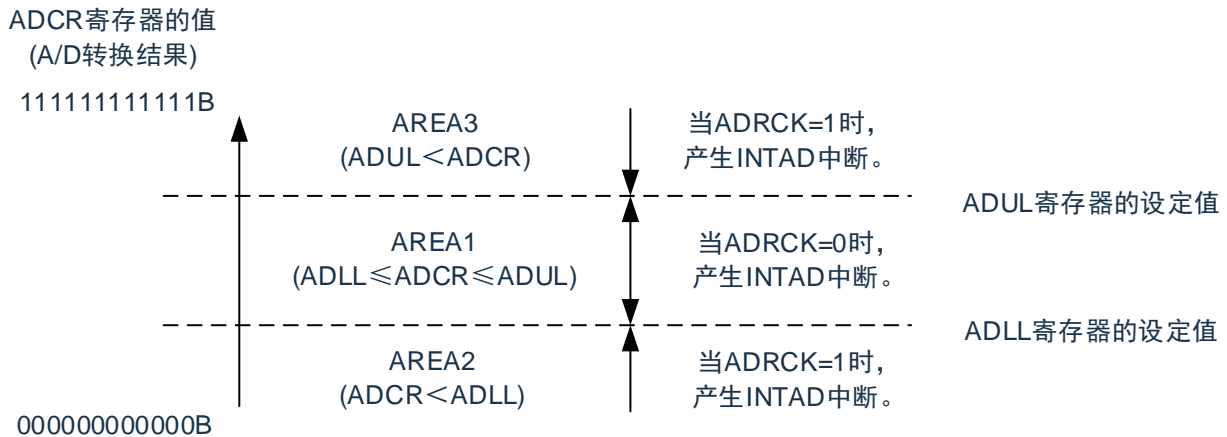
ADREFP1	ADREFP0	A/D转换器的正 (+) 基准电压源的选择
0	0	由V <sub>DD</sub> 提供。
1	0	由内部基准电压提供 (1.45V)。
其他		设定禁止

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11-3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11-3: 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. 当不发生INTAD时, A/D转换结果不保存到ADCR寄存器和ADCRH寄存器。

## 11.2.5 A/D转换器的触发模式寄存器（ADTRG）

这是设定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定ADTRG寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-9: A/D转换器的触发模式寄存器（ADTRG）的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADTRG	ADTMD1	ADTMD0	0	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	A/D转换触发模式的选择
0	0	软件触发模式
0	1	
1	0	硬件触发无等待模式
1	1	硬件触发等待模式

ADTRS1	ADTRS0	硬件触发信号的选择
0	0	定时器通道1的计数结束或者捕捉结束中断信号（INTTM01）
0	1	ELC选择的事件信号
1	0	实时时钟中断信号（INTRTC）
1	1	间隔定时器中断信号（INTIT）

注意：

1. 要改写ADTRG寄存器时，必须在转换停止状态（ADCS=0，ADCE=0）下进行。
2. 为了正常结束A/D转换，必须至少将硬件触发间隔设定为以下时间：  
 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  
 （F<sub>CLK</sub>：CPU/外围硬件的时钟频率）

## 11.2.6 模拟输入通道指定寄存器（ADS）

这是指定要进行A/D转换的模拟电压输入通道的寄存器。

通过8位存储器操作指令设定ADS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-10：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	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0

- 选择模式（ADM1.ADMD=0）

ADS寄存器设定值		CH选择
ADISS	ADC[5:0]	
0	6'h00	ANI0(P20)
0	6'h01	ANI1(P21)
0	6'h02	ANI2(P22)
0	6'h03	ANI3(P23)
0	6'h04	ANI4(P24)
0	6'h05	ANI5(P25)
0	6'h06	ANI6(P26)
0	6'h07	ANI7(P30)
0	6'h08	ANI8(P31)
0	6'h09	ANI9(P32)
0	6'h0a	ANI10(P33)
0	6'h0b	ANI11(P34)
0	6'h0c	ANI12(P35)
0	6'h0d	ANI13(P36)
0	6'h0e	ANI14(P37)
0	6'h0f	ANI15(P00)
0	6'h10	ANI16(P01)
0	6'h11	ANI17(P02)
0	6'h12	ANI18(P10)
0	6'h13	ANI19(P11)
1	6'h00	BGR(温度sensor0)
1	6'h01	BGR
其他值禁止设定		

备注：A/D转换器的模拟输入通道因产品而不同。详细的通道分配信息请参考数据手册。

## • 扫描模式 (ADM1.ADMD=1)

ADISS	ADS3	ADS2	ADS1	ADS0	模拟输入通道			
					扫描0	扫描1	扫描2	扫描3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
0	1	1	0	0	ANI12	ANI13	ANI14	ANI15
上述以外					禁止设定。			

## 注意:

1. 扫描模式时必须将bit4,bit5和bit6置“0”。
2. 对于由PMCx寄存器设定为模拟输入的端口, 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) 设定为数字输入/输出的引脚, 不能通过ADS寄存器进行设定。
4. 要改写ADISS位时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
5. 在将ADISS位置“1”后, 不能使用第1次的转换结果。
6. 要转移到深睡眠模式时或者要在CPU以副系统时钟运行中转移到睡眠模式时, 不能将ADISS位置“1”。

## 11.2.7 12位A/D转换结果寄存器（ADCR）

这是保存A/D转换结果的16位寄存器，此寄存器只可读。每当A/D转换结束时，就从逐次逼近寄存器（SAR）装入转换结果<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寄存器进行设定）的设定值范围内，就不保存A/D转换结果。

表11-11：12位A/D转换结果寄存器（ADCR）的格式

复位值：0000H R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	ADCH3	ADCH2	ADCH1	ADCH0	ADCR[11:0]											

备注：

1. 如果在仅仅需要8位分辨率A/D转换结果，可以通过ADCRH寄存器读取转换结果的高8位。
2. 当对ADCR寄存器进行16位存取时，能从bit11依次读转换结果的高12位。

- 选择模式（ADM1.ADMD=0）

ADCH0~3的读出值固定为4 ‘b0000

- 扫描模式（ADM1.ADMD=1）且ADM2.CHRDE=1，ADCH0~3的读出值与转换通道的关系如下：

ADCH3	ADCH2	ADCH1	ADCH0	转换通道标识
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	0	1	1	ANI11
1	1	0	0	ANI12
1	1	0	1	ANI13
1	1	1	0	ANI14
1	1	1	1	ANI15



## 11.2.8 8位A/D转换结果寄存器（ADCRH）

这是保存A/D转换结果的8位寄存器，保存12位分辨率的高8位<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寄存器进行设定）的设定值范围内，就不保存A/D转换结果。

表11-12：8位A/D转换结果寄存器（ADCRH）的格式

复位值：00H R

	7	6	5	4	3	2	1	0
ADCRH								

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。

## 11.2.9 转换结果比较上限值设定寄存器（ADUL）

这是用于检查A/D转换结果上限值的设定寄存器。

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位的设定范围内（参照图11-3：ADRCK位的中断信号产生范围）控制中断信号（INTAD）的产生。

通过8位存储器操作指令设定ADUL寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

注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。

注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。

注3：在设定ADUL寄存器和ADLL寄存器时，必须使ADUL>ADLL。

表11-13：转换结果比较上限值设定寄存器（ADUL）的格式

复位值：FFH RW

	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

### 11.2.10 转换结果比较下限值设定寄存器 (ADLL)

这是用于检查A/D转换结果下限值的设定寄存器。

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位の設定范围内 (参照图11-3: ADRCK位的中断信号产生范围) 控制中断信号 (INTAD) 的产生。

通过8位存储器操作指令设定ADLL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-14: 转换结果比较下限值设定寄存器 (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. 在设定ADUL寄存器和ADLL寄存器时，必须使ADUL > ADLL。

### 11.2.11 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定ADSMPWAIT寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表11-15: 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采样时间为4个ADCLK
1	为“1”时，A/D采样时间为8个ADCLK

备注: 转换停止状态 (ADCS=0) 下设定ADSMPWAIT。

### 11.2.12 控制模拟输入引脚端口功能的寄存器

在将ANix引脚用作A/D转换器的模拟输入时，必须将该端口配置为模拟通道，即将对应的端口模式控制寄存器 (PMCxx) 的位置“1”。详细内容请参照“第二章引脚功能”。

### 11.3 输入电压和转换结果

模拟输入引脚（ANIX）的模拟输入电压和理论上的A/D转换结果（12位A/D转换结果寄存器（ADCR））有以下表达式的关系。

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right) \text{ or } (ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

INT(): 将括号中的数值的整数部分返回的函数

V<sub>AIN</sub>: 模拟输入电压

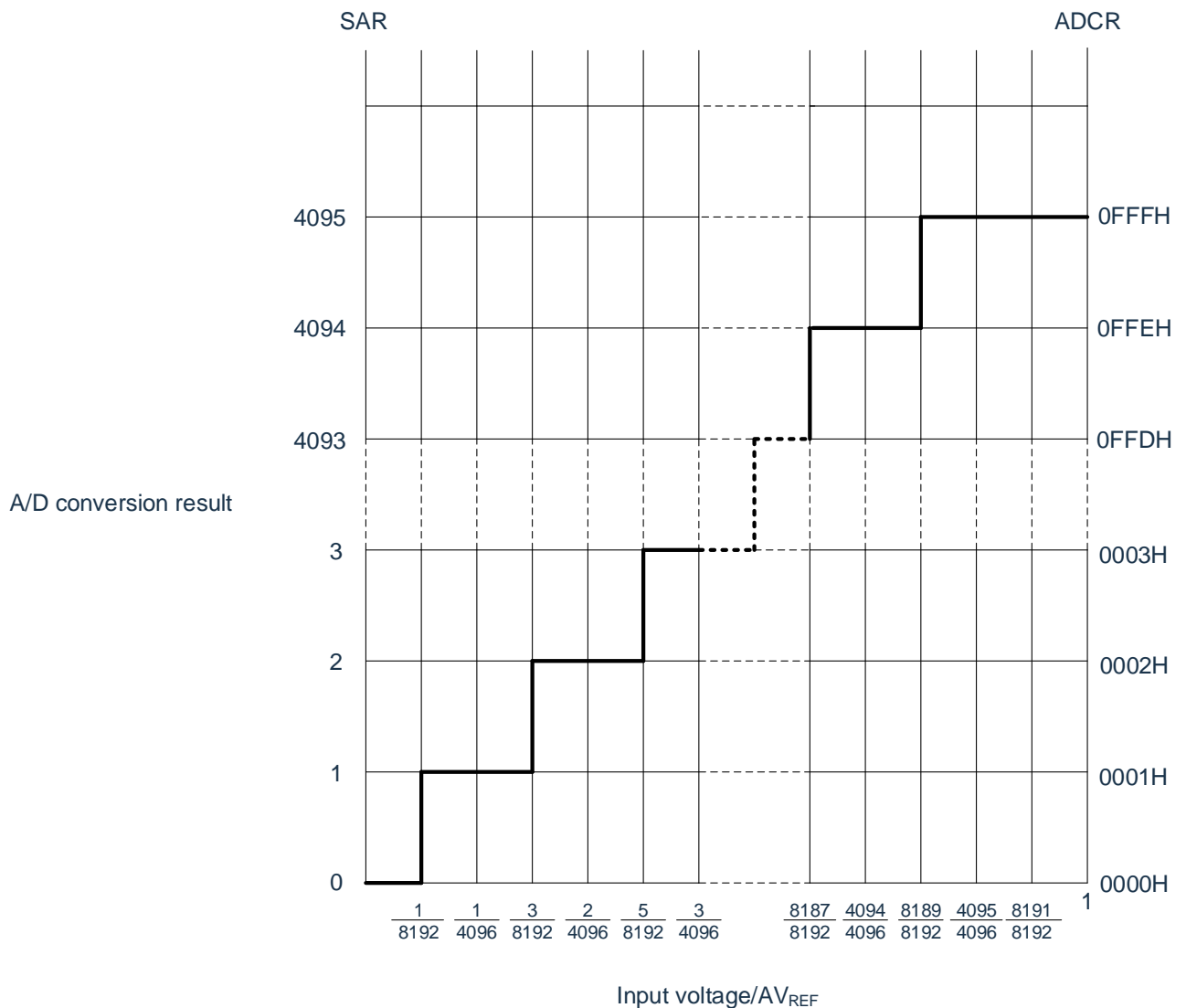
AV<sub>REF</sub>: AV<sub>REF</sub>引脚电压

ADCR: A/D转换结果寄存器（ADCR）的值

SAR: 逐次逼近寄存器

模拟输入电压和A/D转换结果的关系如下图所示。

图11-4: 模拟输入电压和A/D转换结果的关系



备注: AV<sub>REF</sub>是A/D转换器的正 (+) 基准电压。

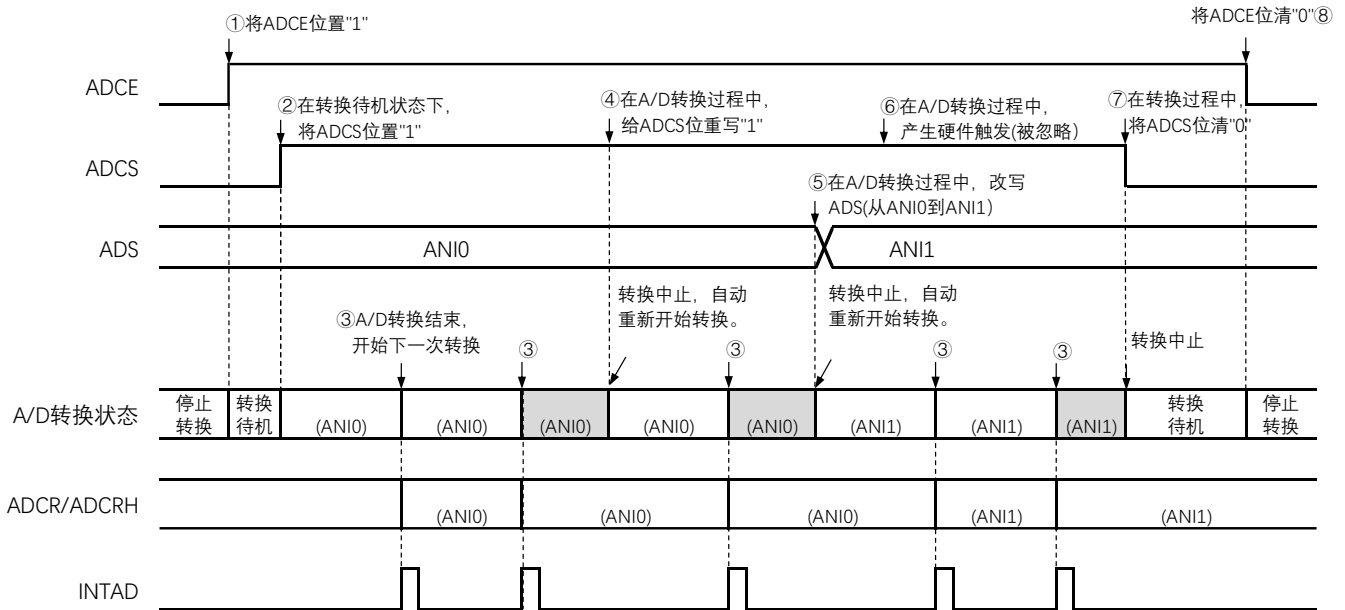
## 11.4 A/D转换器的运行模式

A/D转换器的各模式的运行如下所示。有关各模式的设定步骤，请参照“11.5A/D转换器的设定流程图”。

### 11.4.1 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。

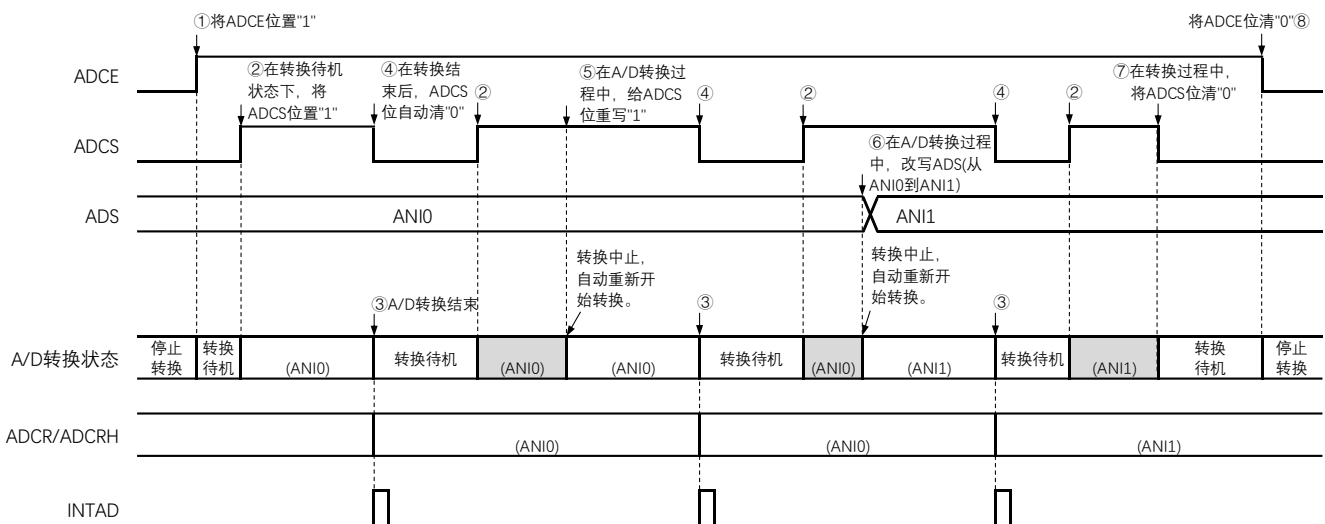
图11-5：软件触发模式（选择模式、连续转换模式）的运行时序例子



### 11.4.2 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。

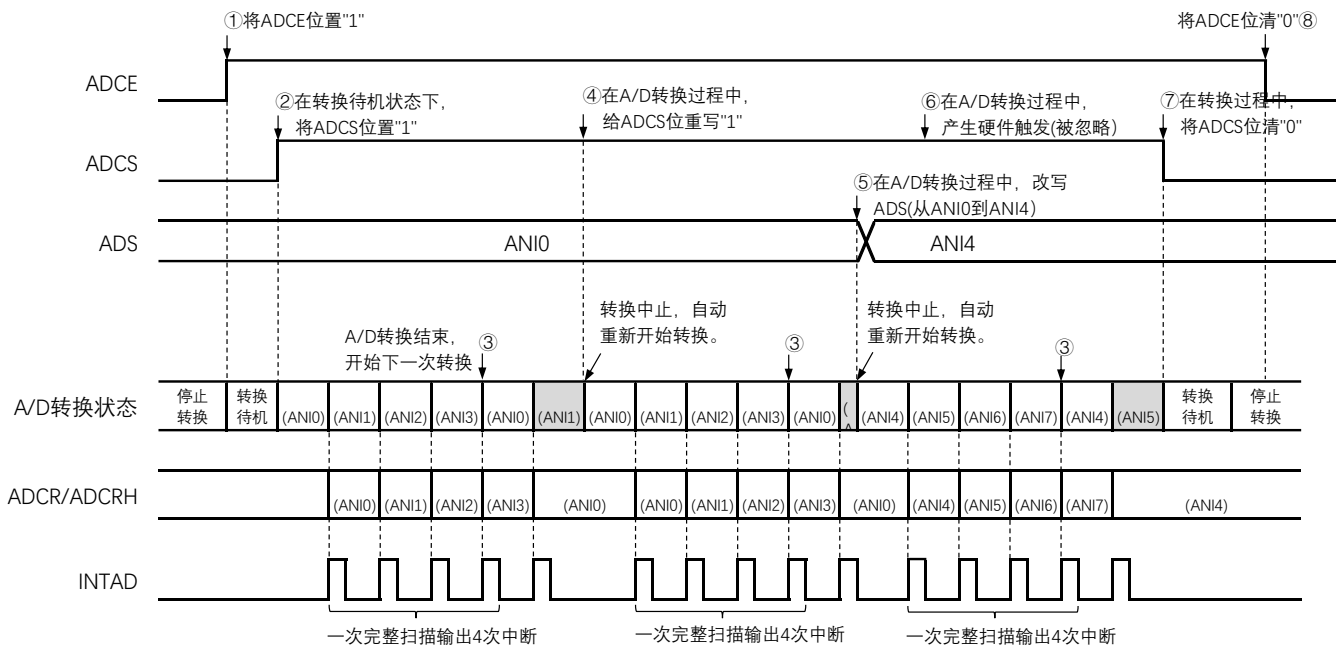
图11-6：软件触发模式（选择模式、单次转换模式）的运行时序例子



### 11.4.3 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。

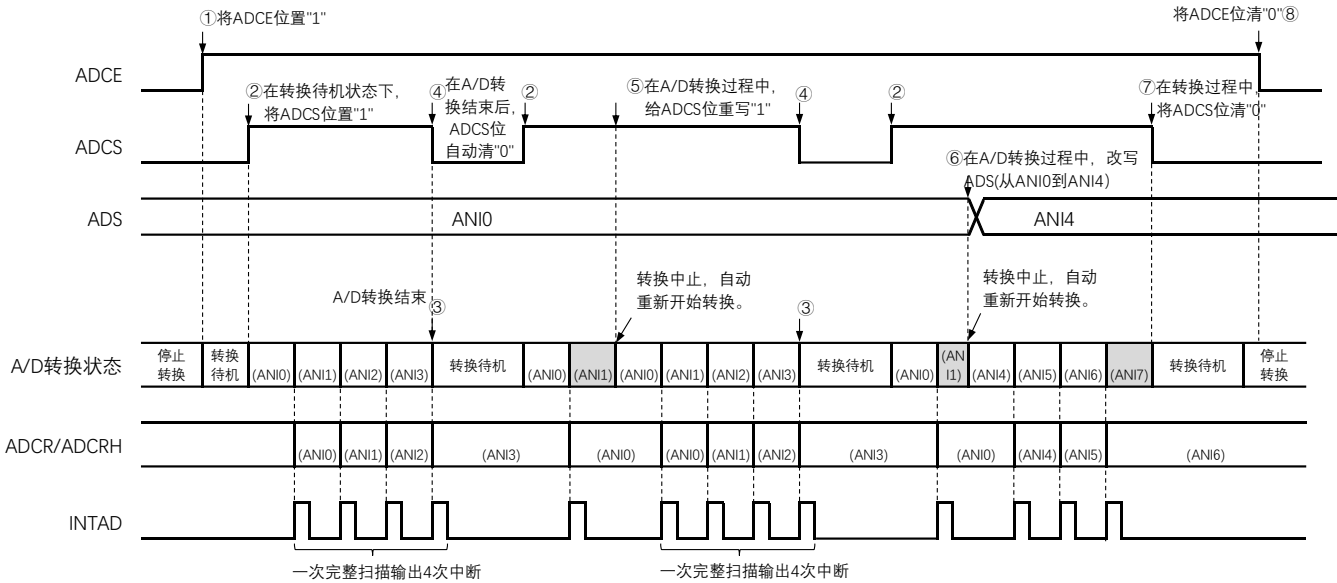
图11-7：软件触发模式（扫描模式、连续转换模式）的运行时序例子



### 11.4.4 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。

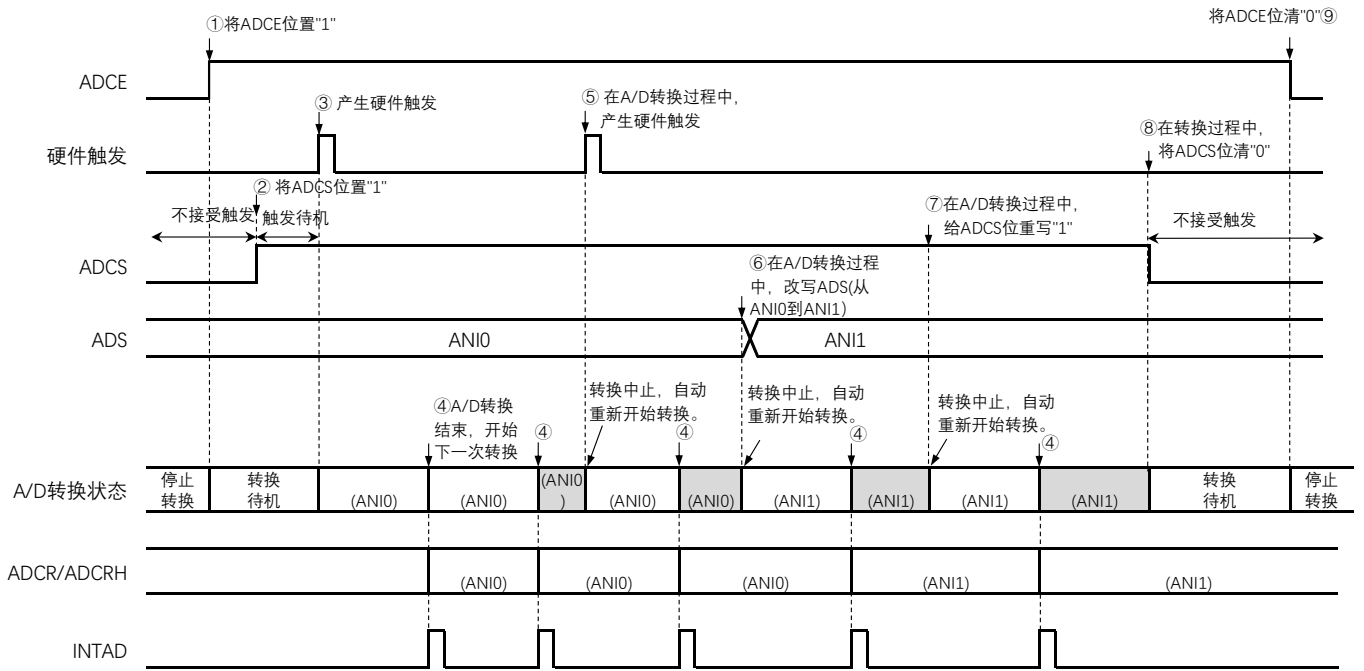
图11-8：软件触发模式（扫描模式、单次转换模式）的运行时序例子



### 11.4.5 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11-9：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

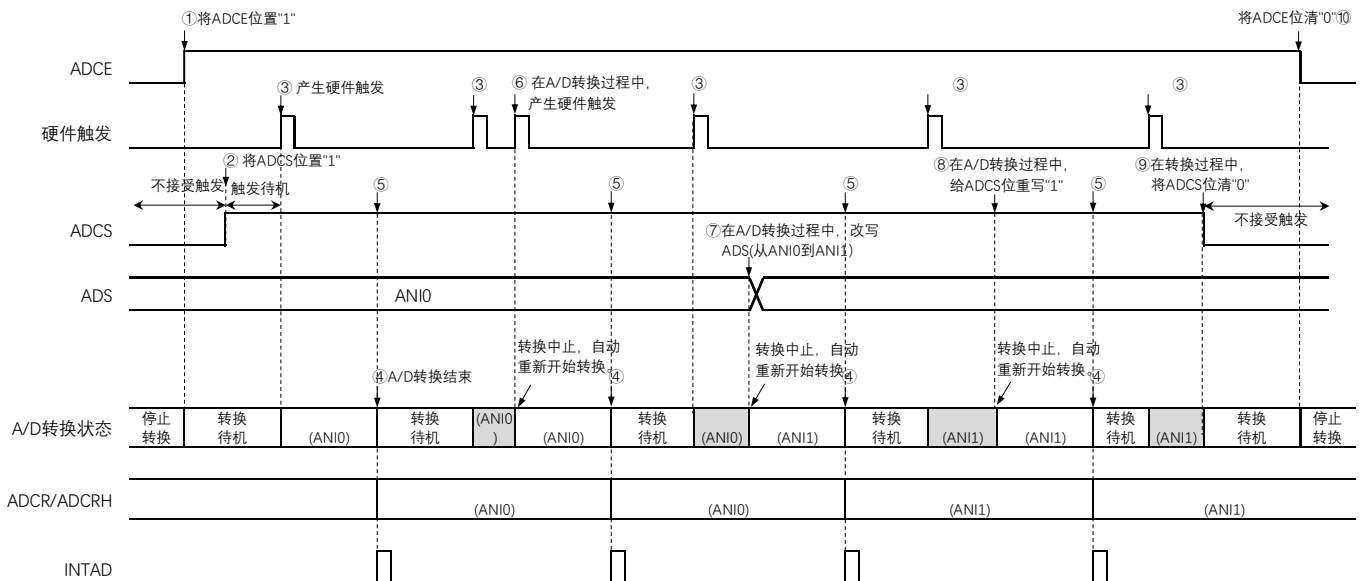




### 11.4.6 硬件触发无等待模式（选择模式、单次转换模式）

- 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。

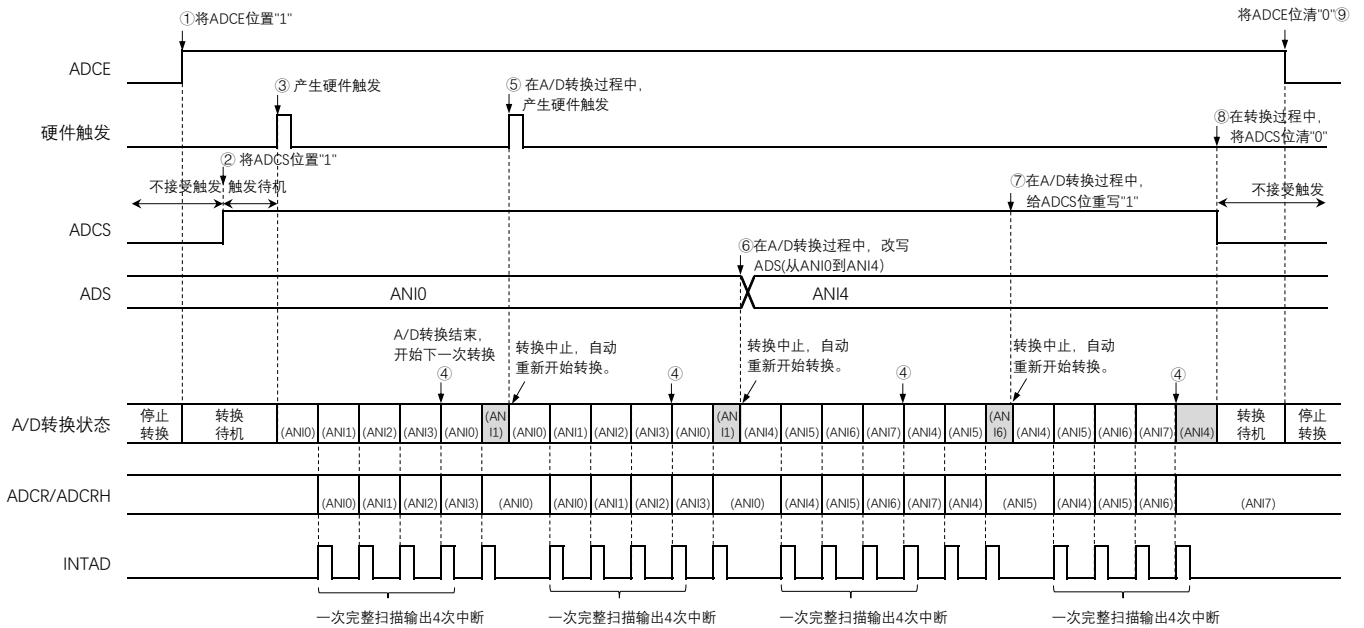
图11-10：硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



### 11.4.7 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。

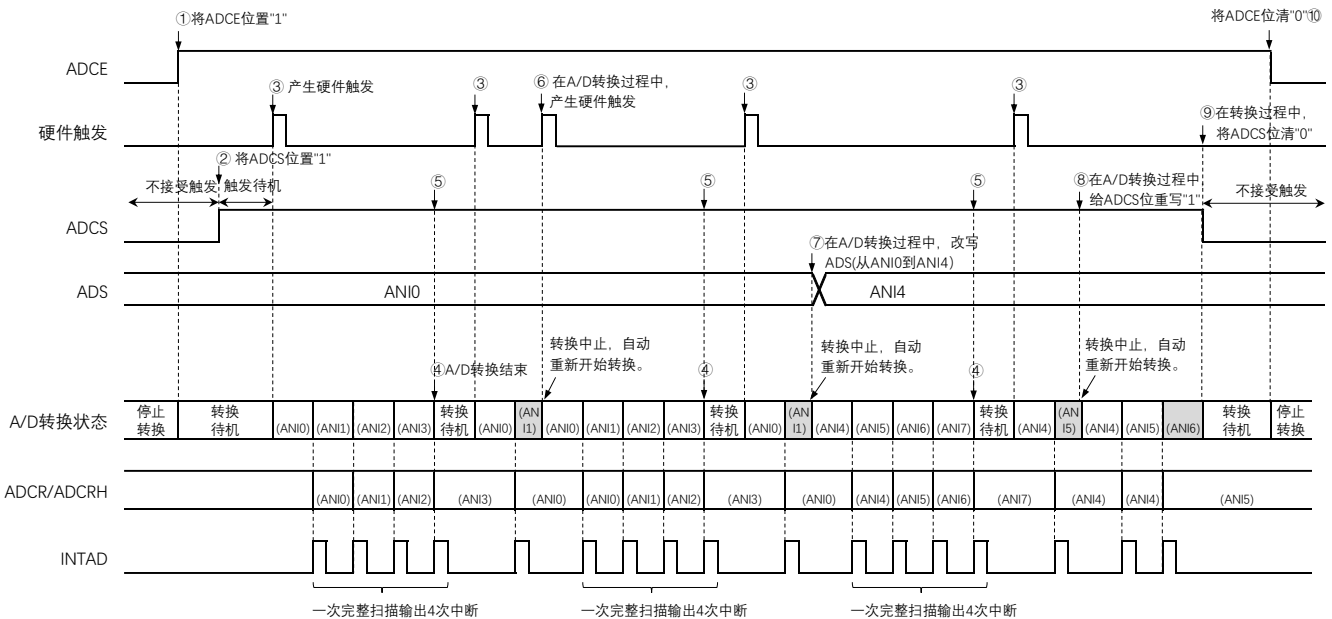
图11-11：硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



### 11.4.8 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。

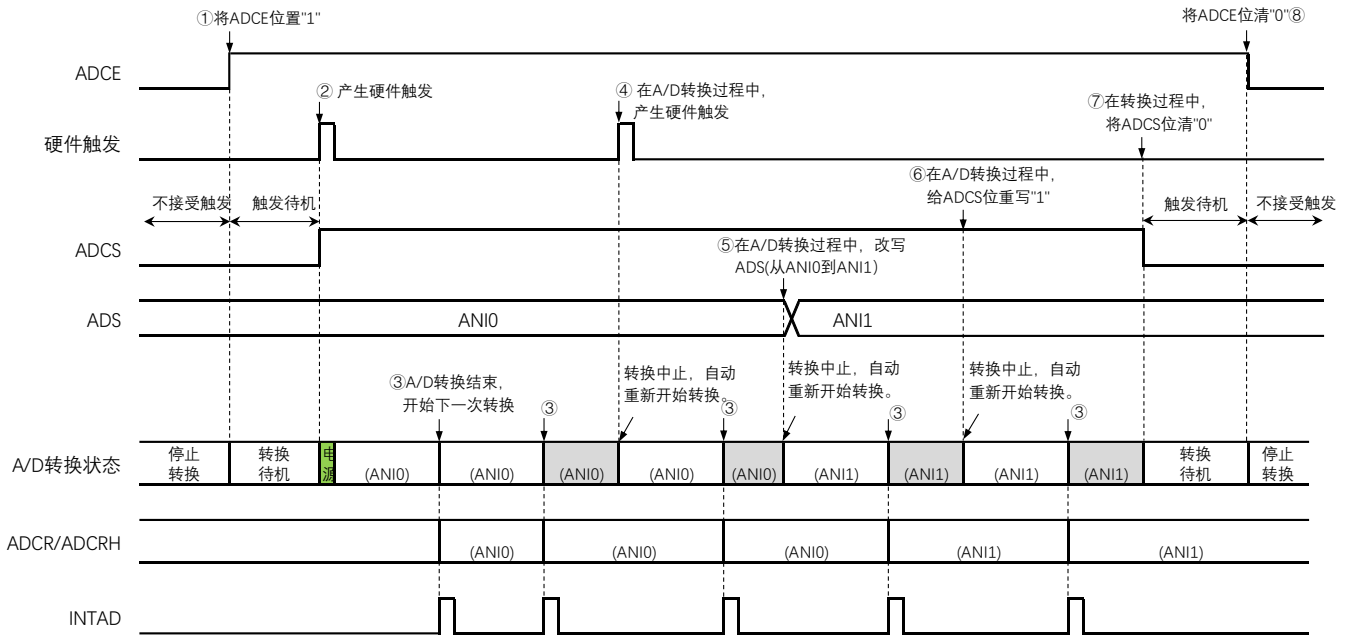
图11-12：硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子




### 11.4.9 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11-13：硬件触发等待模式（选择模式、连续转换模式）的运行时序例子

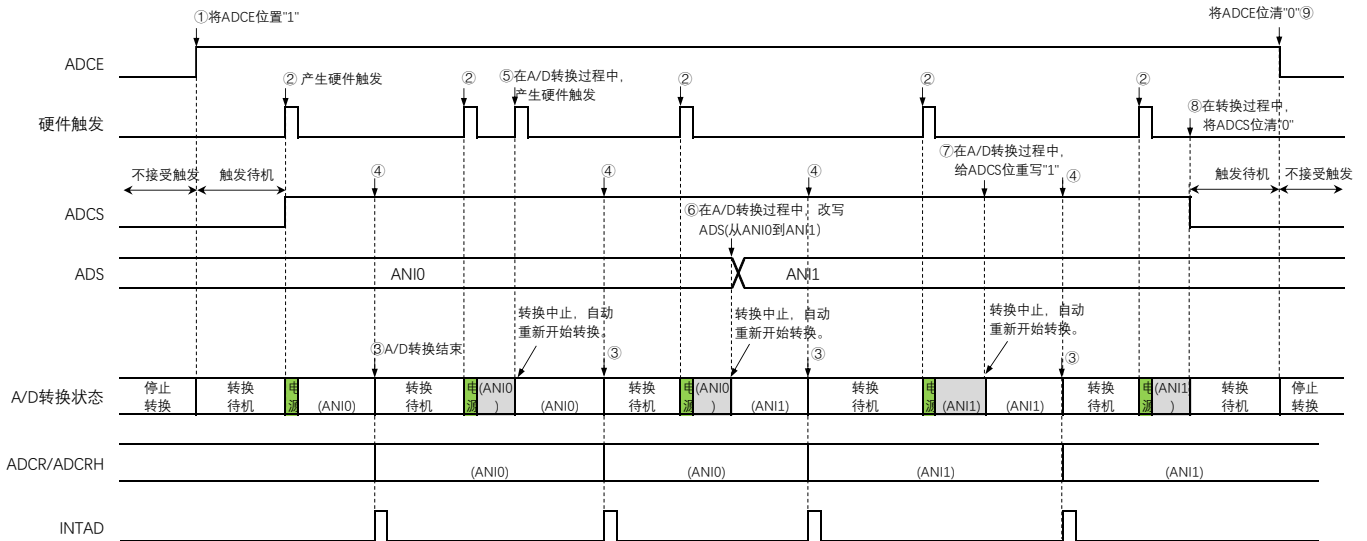


 电源安定等待时间

### 11.4.10 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11-14：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

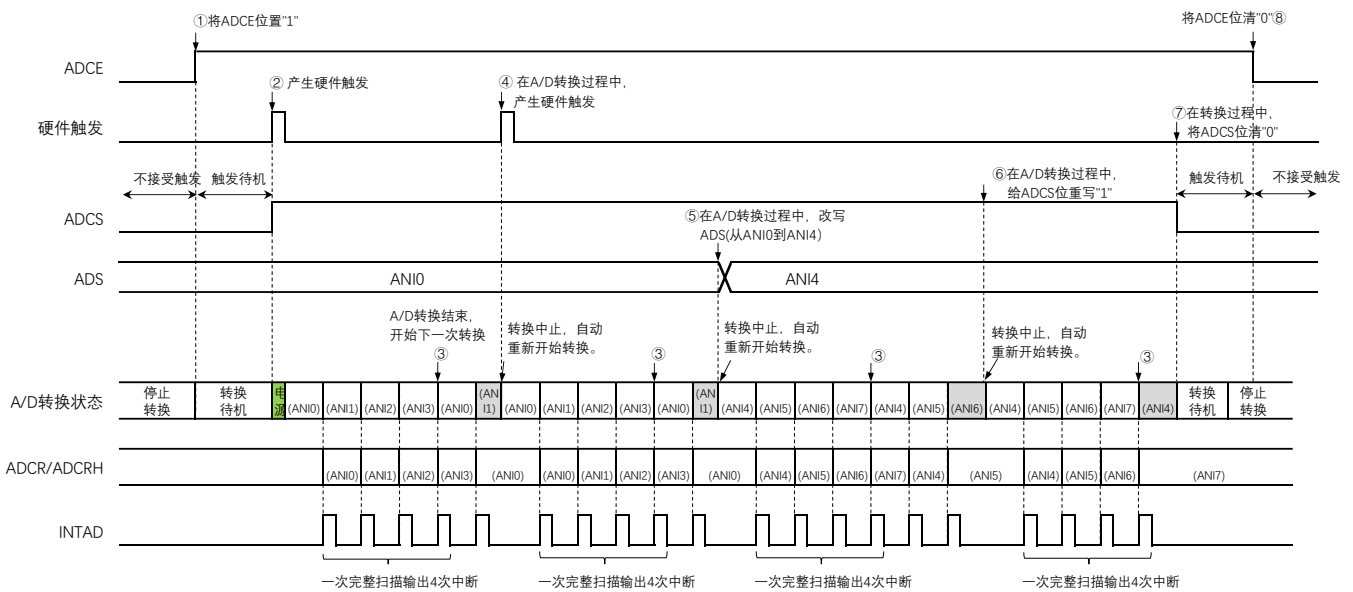


电源安定等待时间

### 11.4.11 硬件触发等待模式（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。在输入硬件触发的同时自动将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）。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 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。
- 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寄存器重新指定的通道开始进行扫描转换。
- 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，然后从最初的通道重新开始转换。
- 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11-15：硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子

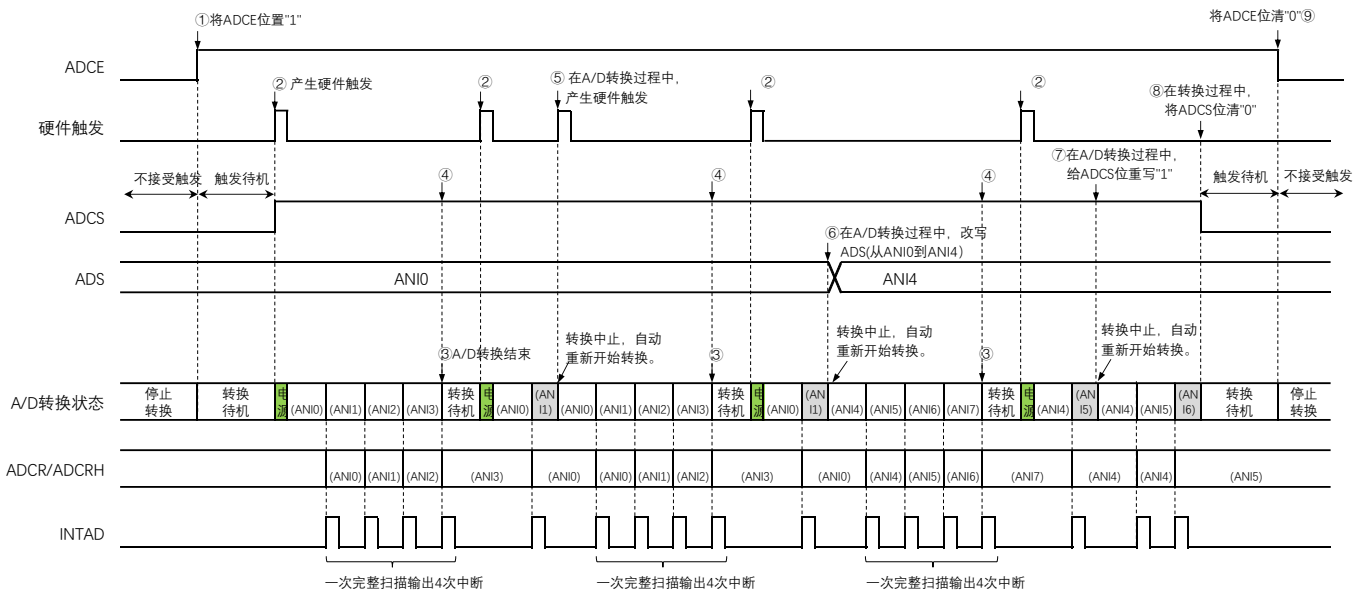


电源安定等待时间

### 11.4.12 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11-16：硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子



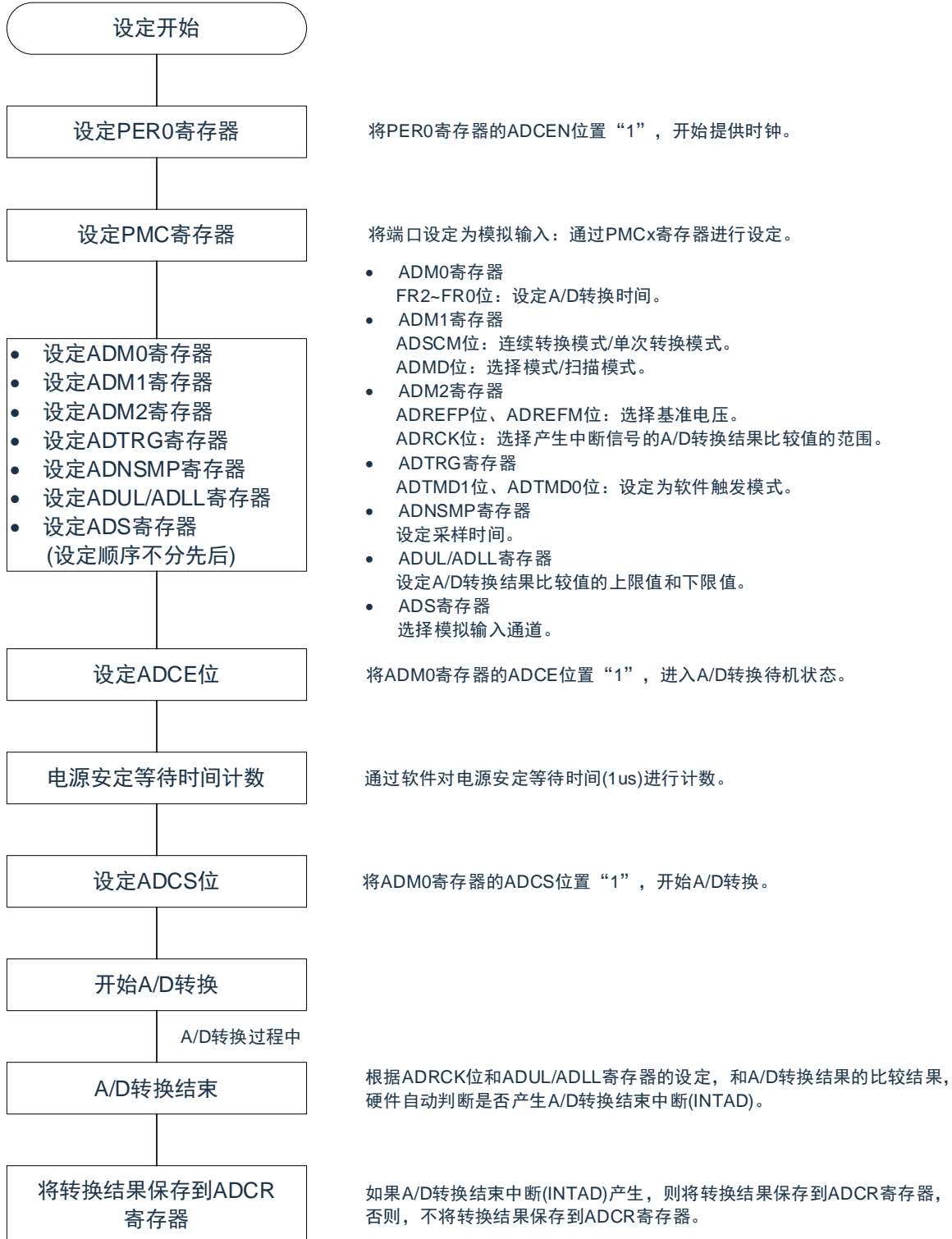
**电源** 电源安定等待时间

## 11.5 A/D转换器的设定流程图

各运行模式的A/D转换器的设定流程图如下所示。

### 11.5.1 软件触发模式的设定

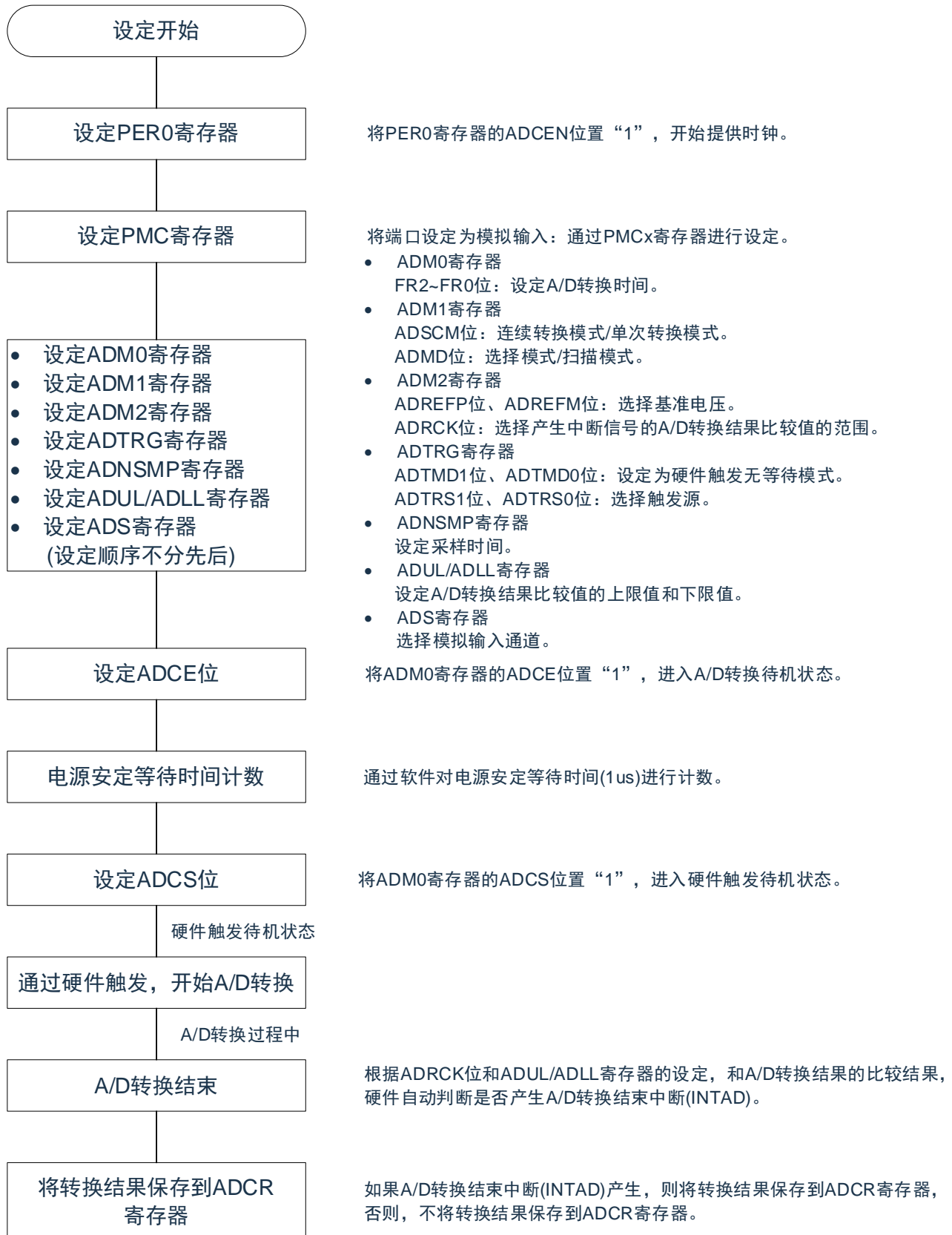
图11-17：软件触发模式的设定





## 11.5.2 硬件触发无等待模式的设定

图11-18：硬件触发无等待模式的设定



### 11.5.3 硬件触发等待模式的设定

图11-19：硬件触发等待模式的设定



## 第12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 的通信功能。其通道功能分配如下：

表 12-1：通用串行通道单元通道功能分配

单元	通道	用作 SSPI	用作 UART
0	0	SSPI00	UART0（支持 LIN-bus）
	1	SSPI01	
1	0	SSPI10	UART1
	1	SSPI11	

注意：

1. “-”表示在此型号产品中不支持；
2. 在单元 0 的通道 0 和通道 1 使用 UART0 时，不能使用 SSPI00 和 SSPI01；
3. 在单元 1 的通道 0 和通道 1 使用 UART1 时，不能使用 SSPI10 和 SSPI11。

## 12.1 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：

### 12.1.1 3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11）

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“12.5 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11）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]

- (1) 主控或者从属的选择
- (2) 输入/输出时钟的相位控制
- (3) 设定由预分频器和通道内部计数器产生的传送周期。
- (4) 最大传送速率

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

注意：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。

## 12.1.2 UART (UART0、UART1)

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 00) 和接收专用 (奇数通道 01) 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“12.7 UART (UART0~UART1) 通信的运行”。

[数据的发送和接收]

- (1) 7 位、8 位、9 位或者 16 位的数据长度
- (2) MSB/LSB 优先的选择
- (3) 发送和接收数据的电平设定、反相的选择
- (4) 奇偶校验位的附加、奇偶校验功能
- (5) 停止位的附加

[中断功能]

- (1) 传送结束中断、缓冲器空中断
- (2) 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

## 12.2 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 12-2: 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) <sup>注</sup>
串行时钟输入/输出	SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11引脚 (用于3线串行I/O)
串行数据输入	SDI00、SDI01、SDI10、SDI11引脚 (用于3线串行I/O)、 RxD0、RxD1引脚 (用于UART)
串行数据输出	SDO00、SDO01、SDO10、SDO11引脚 (用于3线串行I/O)、 TxD0、TxD1引脚 (用于UART)
从属选择输入	SAU0_SS引脚 (用于串行通信单元0做SPI从属选择输入功能) SAU1_SS引脚 (用于串行通信单元1做SPI从属选择输入功能)
控制寄存器	<单元设定部的寄存器> 外围允许寄存器0 (PER0) 串行时钟选择寄存器m (SPSm) 串行通道允许状态寄存m (SEm) 串行通道开始寄存器m (SSm) 串行通道停止寄存器m (STm) 串行输出允许寄存器m (SOEm) 串行输出寄存器m (SOM) 串行输出电平寄存器m (SOLm) 输入切换控制寄存器 (ISC) 噪声滤波器允许寄存器0 (NFEN0)
	<各通道部的寄存器> 串行数据寄存器mn (SDRmn) 串行模式寄存器mn (SMRmn) 串行通信运行设定寄存器mn (SCRmn) 串行状态寄存器mn (SSRmn) 串行标志清除触发寄存器mn (SIRmn)
	端口复用功能配置寄存器 (PxxCFG) 端口输出模式寄存器 (POMxx) 端口模式寄存器 (PMxx) 端口寄存器 (Pxx)

注: SE<sub>mn</sub>=1 的动作期间;

备注: m: 单元号 (m=0、1);

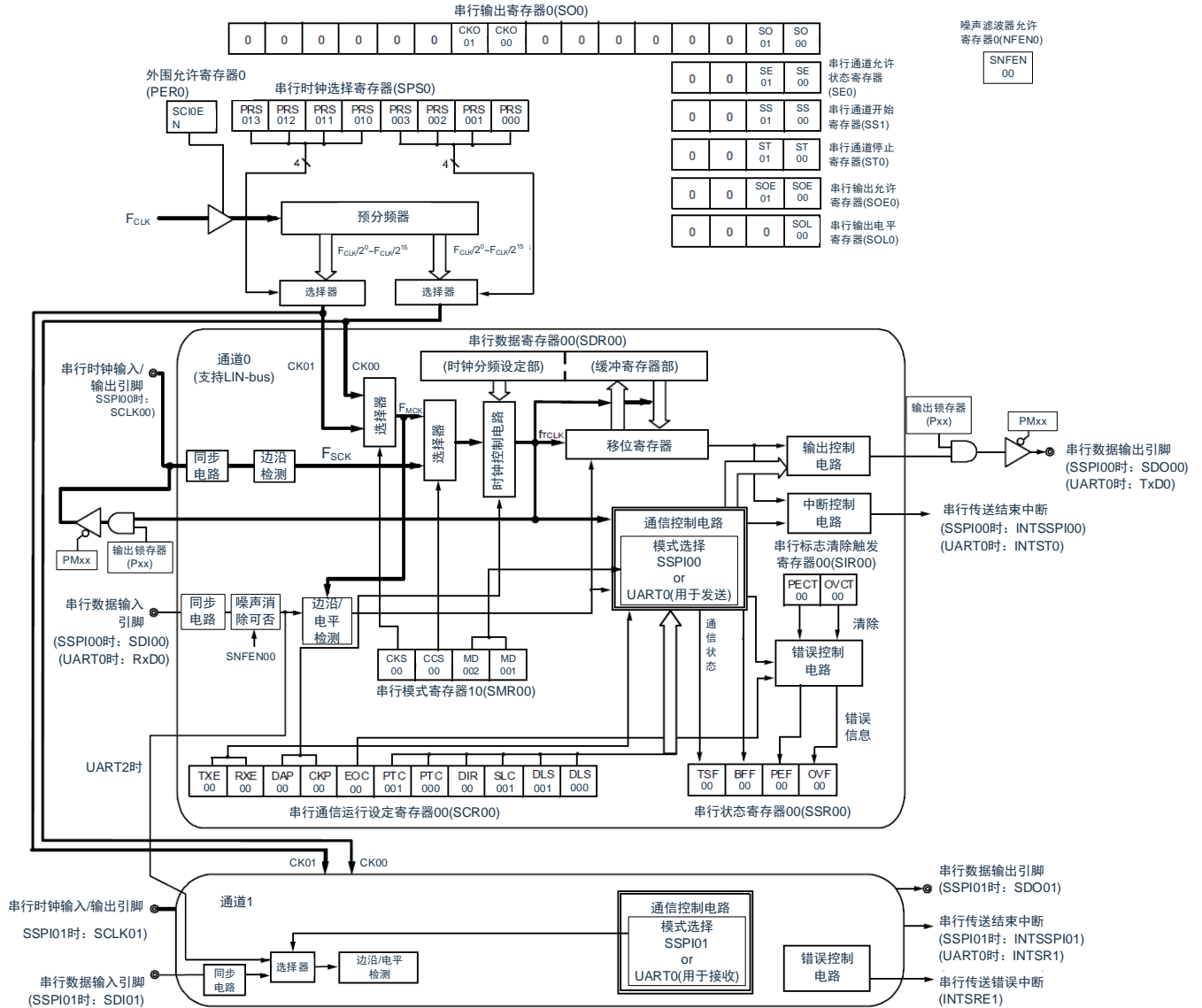
n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

q: UART 号 (q=0、1);

通用串行通信单元的框图如图 12-1 所示：（以单元 0 为例）

图 12-1：通用串行通信单元 0 的框图



备注：单元 0, 1 的结构相同

## 12.2.1 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)。

表 12-3: 移位寄存器的格式

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
移位寄存器																

## 12.2.2 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：

- (1) 7 位数据长度（保存在 SDRmn 寄存器的 bit0~6）；
- (2) 8 位数据长度（保存在 SDRmn 寄存器的 bit0~7）；
- (3) 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。

- (1) SSPIp 通信.....SDIOpL
- (2) UARTq 接收.....RXDq (UARTq 接收数据寄存器)
- (3) UARTq 发送.....TXDq (UARTq 发送数据寄存器)

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）；

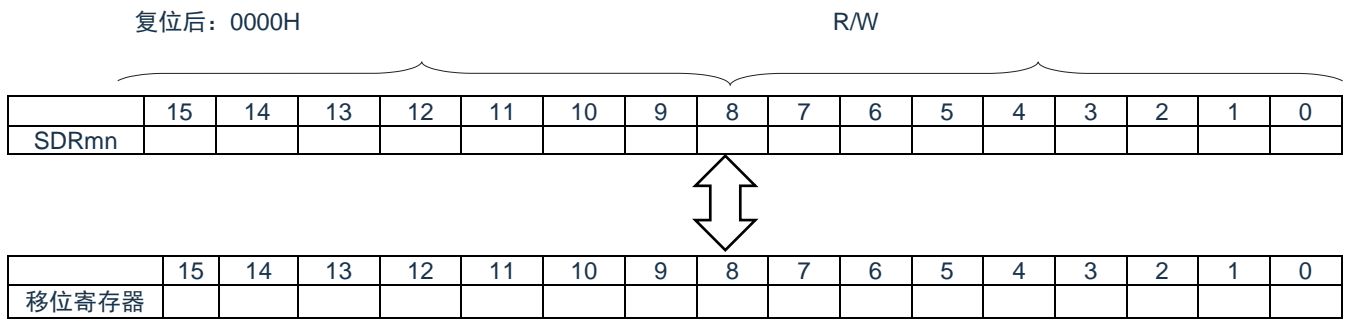
n：通道号（n=0、1）；

p：SSPI 号（p=00、01、10、11）；

q：UART 号（q=0、1）；



表 12-4: 串行数据寄存器 mn (SDRmn) 的格式



备注:

1. 有关 SDRmn 寄存器的高 7 位的功能, 请参照“12.3 控制通用串行通信单元的寄存器”;
2. m: 单元号 (m=0、1);  
n: 通道号 (n=0、1)。

## 12.3 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：

- (1) 外围允许寄存器 0 (PER0)
- (2) 串行时钟选择寄存器 m (SPSm)
- (3) 串行模式寄存器 mn (SMRmn)
- (4) 串行通信运行设定寄存器 mn (SCRmn)
- (5) 串行数据寄存器 mn (SDRmn)
- (6) 串行标志清除触发寄存器 mn (SDIRmn)
- (7) 串行状态寄存器 mn (SSRmn)
- (8) 串行通道开始寄存器 m (SSm)
- (9) 串行通道停止寄存器 m (STm)
- (10) 串行通道允许状态寄存器 m (SEm)
- (11) 串行输出允许寄存器 m (SOEm)
- (12) 串行输出电平寄存器 m (SOLm)
- (13) 串行输出寄存器 m (SOM)
- (14) 输入切换控制寄存器 (ISC)
- (15) 噪声滤波器允许寄存器 0 (NFEN0)
- (16) 端口复用功能配置寄存器 (PxxCFG)
- (17) 端口输出模式寄存器 (POMx)
- (18) 端口模式寄存器 (PMx)
- (19) 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: 0x40041500

表 12-5: 寄存器列表

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/W	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
0x040	SDRm0	R/W	0000H
0x042	SDRm1	R/W	0000H

备注: 单元号 m=0, 1。

### 12.3.1 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”。

表 12-6: 围允许寄存器 0 (PER0) 的格式

地址: 0x40020420	复位后: 00H							R/W
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	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, 必须先在 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)、端口复用功能配置寄存器 (PxxCFG)、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):

- (1) 串行时钟选择寄存器 m (SPSm);
- (2) 串行模式寄存器 mn (SMRmn);
- (3) 串行通信运行设定寄存器 mn (SCRmn);
- (4) 串行数据寄存器 mn (SDRmn);
- (5) 串行标志清除触发寄存器 mn (SIRmn);
- (6) 串行状态寄存器 mn (SSRmn);
- (7) 串行通道开始寄存器 m (SSm);
- (8) 串行通道停止寄存器 m (STm);
- (9) 串行通道允许状态寄存器 m (SEm);
- (10) 串行输出允许寄存器 m (SOEm);
- (11) 串行输出电平寄存器 m (SOLm);
- (12) 串行输出寄存器 m (SOM)。

## 12.3.2 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”。

表 12-7: 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	PRSm	PRSm	PRSm	PRSm	PRSm	PRSm	PRSm	PRSm

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>
其他				禁止设定

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。

### 12.3.3 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) 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”。

表 12-8: 串行模式寄存器 mn (SMRmn) 的格式(1/2)

复位后: 0020H															R/W		
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	STSmn	0	SISmn0	0	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 寄存器为 bit13~6、4、3) 置“0”，并且将 bit5 置 1”。

备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

q: UART 号 (q=0、1);

表 12-8: 串行模式寄存器 mn (SMRmn) 的格式(2/2)

复位后:	0020H													R/W		
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	STSmn	0	SISmn0	0	0	0	0	MDmn2	MDmn1	MDmn0

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	禁止设定。
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 寄存器为 bit13~6、4、3) 置“0”, 并且将 bit5 置 1”。

备注: m: 单元号 (m=0、1) ;

n: 通道号 (n=0、1) ;

p: SSPI 号 (p=00、01、10、11) ;

q: UART 号 (q=0、1) ;

### 12.3.4 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”。

表 12-9: 串行通信运行设定寄存器 mn (SCRmn) 的格式(1/3)

复位后: 0087H																
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0

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 模式中的数据和时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4

在UART模式中，必须将DAPmn位和CKPmn位都置0。

EOCmn	错误中断信号 (INTSREx (x=0~1)) 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 INTSREn;

注意: 必须将 bit6、10、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、01、10、11)。

表 12-9: 串行通信运行设定寄存器 mn (SCRmn) 的格式(2/3)

复位后: 0087H																R/W	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SCR mn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	DLS mn2	DLS mn1	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3</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。

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, 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)。

注 1: 只限于 SCR00、SCR10 寄存器。

注意:

1. 与数据的内容无关, 总是附加“0”;
2. 必须将 bit6、10、11 置“0”。

备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11)。

表 12-9: 串行通信运行设定寄存器 mn (SCRmn) 的格式(3/3)

复位后: 0087H

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR mn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	DLS mn2	DLS mn1	DLS mn0

DLSmn3	DLSmn2	DLSmn1	DLSmn0	SSPI模式中的数据长度的设定	串行功能对应	
					SSPI	UART
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。		

注 1: 只限于 SCR00、SCR10 寄存器。

注意: 必须将 bit6、10、11 置“0”。

备注: m: 单元号 (m=0、1) ;

n: 通道号 (n=0、1) ;

p: SSPI 号 (p=00、01、10、11) 。

### 12.3.5 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 (FMCK) 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 SDRmn 寄存器。

在产生复位信号后, SDRmn 寄存器的值变为“0000H”。

表 12-10: 串行数据寄存器 mn (SDRmn) 的格式

复位后: 0000H								RW								
SDRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SDRmn[15:9]							运行时钟分频的传送时钟设定	
0	0	0	0	0	0	0	FMCK	
0	0	0	0	0	0	1	FMCK/2	
0	0	0	0	0	1	0	FMCK/3	
0	0	0	0	0	1	1	FMCK/4	
•	•	•	•	•	•	•	•	
1	1	1	1	1	1	0	FMCK/127	
1	1	1	1	1	1	1	FMCK/128	

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“12.2 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)。

### 12.3.6 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”。

表 12-11: 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
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn	PECTmn	OVCTmn

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溢出错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的OVFmn位清“0”。

注 1: 只限于 SIR01、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”。

### 12.3.7 串行状态寄存器mn (SSRmn)

SSRmn 寄存器表示通道 n 的通信状态和发生错误的情况。表示的错误为帧错误、奇偶校验错误和溢出错误。通过 16 位存储器操作指令读取 SSRmn 寄存器。

能用 SSRmnL 并且通过 8 位存储器操作指令读取 SSRmn 寄存器的低 8 位。

在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

表 12-12: 串行状态寄存器 mn (SSRmn) 的格式(1/2)

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	TSFmn	BFFmn	0	0	FEFmn	PEFmn	OVFmn

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STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信待机状态）时</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STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信允许状态）时</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。

备注: 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)。

m: 单元号 (m=0、1) n: 通道号 (n=0、1)

表 12-12: 串行状态寄存器 mn (SSRmn) 的格式(2/2)

符号	复位后				0000H				R							
	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	TSFmn	BFFmn	0	0	FEFmn	PEFmn	OVFmn

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”层面上名称时 [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)。
[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	

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;

m: 单元号 (m=0、1);

n: 通道号 (n=0、1)。

### 12.3.8 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”。

表 12-13: 串行通道开始寄存器 m (SSm) 的格式

符号	15	复位后		12	11	10	0000H								R/W		1	0
SSm	0	0	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 <sup>注</sup> 。

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”。

### 12.3.9 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”。

表 12-14: 串行通道停止寄存器 m (STm) 的格式

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST01	ST00

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 <sup>注</sup> 。

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”。



### 12.3.10 串行通道允许状态寄存器m (SEm)

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

如果给串行开始允许寄存器 m (SSm) 的各位写“1”，就将其对应位置“1”。如果给串行通道停止寄存器 m (STm) 的各位写“1”，就将其对应位清“0”。

对于允许运行的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOM) 的 CKOmn 位（通道 n 的串行时钟输出）的值，而从串行时钟引脚输出由通信运行反映的值。

对于停止运行的通道 n，能通过软件设定 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”。

表 12-15: 串行通道允许状态寄存器 m (SEm) 的格式

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

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）。

### 12.3.11 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”。

表 12-16: 串行输出允许寄存器 m (SOEm) 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		
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1	SOEm0		

SOEmn	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。

备注：SOEm 寄存器的 bit15~2 置“0”；

m: 单元号 (m=0、1)；

n: 通道号 (n=0、1)。

### 12.3.12 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”。

表 12-17: 串行输出寄存器 m (SOm) 的格式

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	CKOm1	CKOm0	0	0	0	0	0	0	SOm1	SOm0

CKOmn	通道n的串行时钟输出
0	串行时钟的输出值为“0”。
1	串行时钟的输出值为“1”。

SOmn	通道n的串行数据输出
0	串行数据的输出值为“0”。
1	串行数据的输出值为“1”。

注意: 必须将 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)。

### 12.3.13 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”。

表 12-18: 串行输出电平寄存器 m (SOLm) 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
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、SOL2 寄存器的 bit15~1 置“0”；

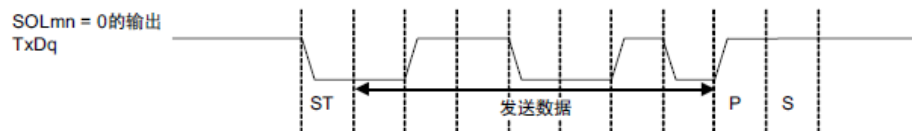
备注：m：单元号 (m=0、1)；

n：通道号 (n=0)。

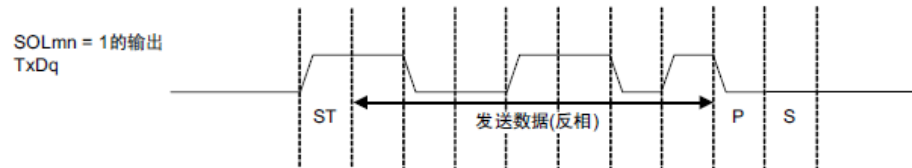
当进行 UART 发送时，发送数据的电平反相例子如图 12-2 所示：

图 12-2: 发送数据的电平反相例子

(a) 正相输出 (SOLmn = 0)



(b) 反相输出 (SOLmn = 1)



备注：m：单元号 (m=0、1)；

n：通道号 (n=0)。

### 12.3.14 输入切换控制寄存器 (ISC)

在通过 UART0 实现 LIN-bus 通信时，ISC 寄存器的 ISC1 位和 ISC0 位用于外部中断和定时器阵列单元的协调。如果将 bit0 置“1”，就选择串行数据输入 (RxD0) 引脚的输入信号作为外部中断的输入 (INTP0)，因此能通过 INTP0 中断检测唤醒信号。

如果将 bit1 置“1”，就选择串行数据输入 (RxD0) 引脚的输入信号作为定时器的输入，因此能通过定时器检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽。

SSIE00 位在 SSPI00 通信的从属模式中控制通道 0 的 SS00 引脚输入。在给 SS00 引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收；在给 SS00 引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设定进行发送和接收。

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”。

表 12-19: 输入切换控制寄存器 (ISC) 的格式

地址:	40040473H	复位后:	00H	R/W					
符号	7	6	5	4	3	2	1	0	
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0	
SSIE00	SSPI00通信的从属模式中通道0的SS00输入的设定								
0	SS00引脚输入无效。								
1	SS00引脚输入有效。								
ISC1	定时器Timer4的通道3的输入切换								
0	将TI03引脚的输入信号用作定时器的输入（通常运行）。								
1	将RxD0引脚的输入信号用作定时器的输入（检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽）。								
ISC0	外部中断 (INTP0) 的输入切换								
0	将INTP0引脚的输入信号用作外部中断的输入（通常运行）。								
1	将RxD0引脚的输入信号用作外部中断的输入（检测唤醒信号）。								

注意：必须将 bit6~2 置“0”。

### 12.3.15 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 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”。

表 12-20: 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	0	0	SNFEN10	0	SNFEN00

SNFEN10	RxD1 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON

当用作 RxD1 引脚时，必须将 SNFEN10 位置“1”。  
当用作 RxD1 引脚以外的功能时，必须将 SNFEN10 位置“0”。

SNFEN00	RxD0 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON

当用作 RxD0 引脚时，必须将 SNFEN00 位置“1”。  
当用作 RxD0 引脚以外的功能时，必须将 SNFEN00 位置“0”。

注意：必须将 bit7~5、3、1 置“0”。

### 12.3.16 控制串行输入/输出引脚端口功能的寄存器

在使用通用串行通信单元时，必须设定复用端口功能的控制寄存器（端口模式寄存器（PMxx）、端口复用功能配置寄存器（PxxCFG）、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”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

另外，当用于N沟道漏极开路输出模式时，必须将各端口对应的端口输出模式寄存器（POMxx）的位置“1”。

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”。

## 12.4 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。

### 12.4.1 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”。

外围允许寄存器 0 (PER0) .....只将要停止 SCIm 的对应位置“0”。

表 12-21: 以单元为单位停止运行时的外围允许寄存器 0 (PER0) 的设定

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

SCImEN	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。

备注:

- 当 SCImEN 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：
  - ① 输入切换控制寄存器 (ISC)
  - ② 噪声滤波器允许寄存器 0 (NFEN0)
  - ③ 端口复用功能配置寄存器 (PxxCFG)
  - ④ 端口输出模式寄存器 (POMx)
  - ⑤ 端口模式寄存器 (PMx)
  - ⑥ 端口寄存器 (Px)
- x: 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”。



## 12.4.2 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。

(1) 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。

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	STm1	STm0

- ① 将 SE<sub>m</sub>n 位清“0”并且停止通信运行
- ② 因为 ST<sub>m</sub>n 位是触发位，所以如果 SE<sub>m</sub>n 位为“0”就立即清除 ST<sub>m</sub>n 位。

(2) 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。

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

- ① 0: 运行停止状态
- ② SE<sub>m</sub> 寄存器是只读状态寄存器，通过 ST<sub>m</sub> 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 SO<sub>m</sub> 寄存器的 CKO<sub>m</sub>n 位的值。

(3) 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。

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	SOE01	SOE00

- ① 0: 通过串行通信运行来停止输出
- ② 对于已经停止串行输出的通道，能通过软件设定 SO<sub>m</sub> 寄存器的 SO<sub>m</sub>n 位的值。

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

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	CKOm1	CKOm0	0	0	0	0	0	0	SOM1	SOM0

- ① CKOm1/0: 1: 串行时钟的输出值为“1”; 0: 串行时钟的输出值为“0”
- ② SOM1/0: 1: 串行数据的输出值为“1”; 0: 串行数据的输出值为“0”
- ③ 当将各通道对应的引脚用作端口功能时，必须将相应的 CKO<sub>m</sub>n 位和 SO<sub>m</sub>n 位置“1”。

备注：只限于通用串行通信单元 0；

m: 单元号 (m=0、1)；

n: 通道号 (n=0、1)；

0/1: 根据用户的用途置“0”或者“1”。

## 12.5 3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11）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]

- (1) 7 位~16 位的数据长度
- (2) 发送和接收数据的相位控制
- (3) MSB/LSB 优先的选择

[时钟控制]

- (1) 主控或者从属的选择
- (2) 输入/输出时钟的相位控制
- (3) 设定由预分频器和通道内部计数器产生的传送周期。
- (4) 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

SCI0 的通道 0~1, SCI1 的通道 0~1 是支持 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11）的通道。

3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11）有以下 6 种通信运行：

- (1) 主控发送（参照 12.5.1）
- (2) 主控接收（参照 12.5.2）
- (3) 主控的发送和接收（参照 12.5.3）
- (4) 从属发送（参照 12.5.4）
- (5) 从属接收（参照 12.5.5）
- (6) 从属的发送和接收（参照 12.5.6）

注意：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。

## 12.5.1 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。

表 12-22：主控发送

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0的通道0	SCI0的通道1	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、 SDO00	SCLKOI01、 SDO01	SCLKOI10、 SDO10	SCLKOI11、 SDO11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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 <sup>注</sup>	Max.F <sub>CLK</sub> /2[Hz] Min.F <sub>CLK</sub> /(2×2 <sup>11</sup> ×128)[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。

m: 单元号 (m=0、1) ;

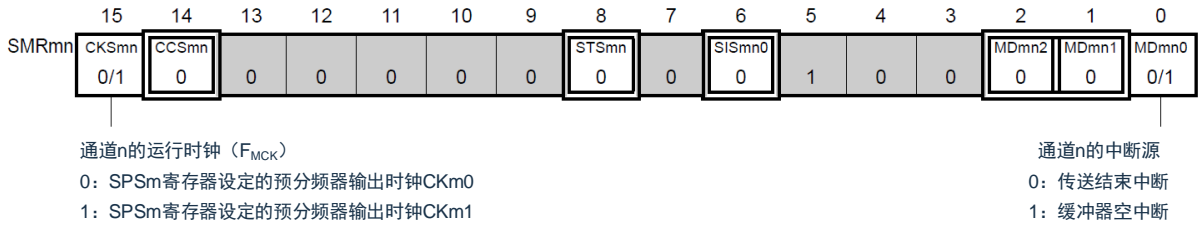
n: 通道号 (n=0、1) ;

mn=00~01、10~11。

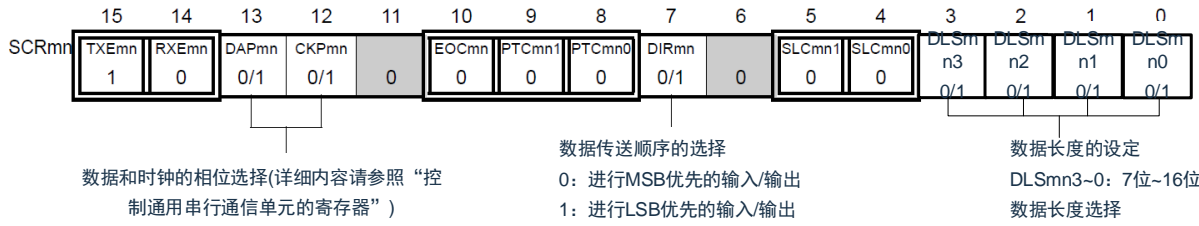
(1) 寄存器的设定

图 12-3: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)  
主控发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

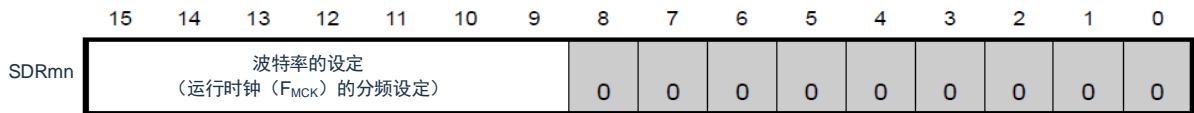


(b) 串行通信运行设定寄存器mn(SCRmn)

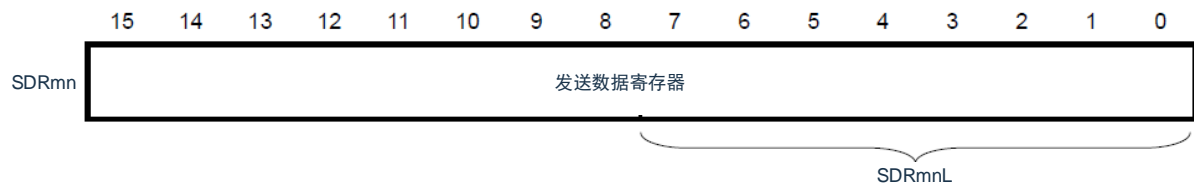


(c) 串行数据寄存器mn(SDRmn)

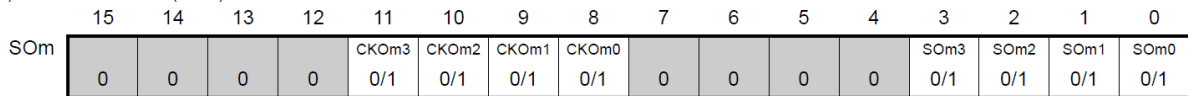
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)

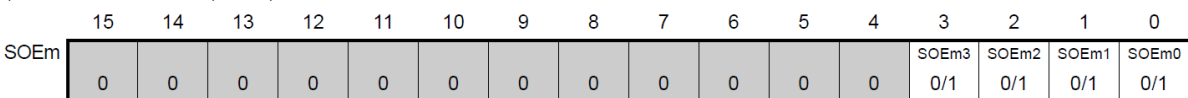


(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。

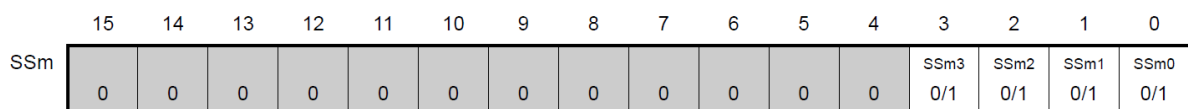


当时钟相位为“正相”(SCRmn寄存器的CKPmn=0)时,“1”为开始通信;当时钟相位为“反相”(CKPmn=1)时,“0”为开始通信。

(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1) mn=00~01、10~11

■: 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 12-4: 主控发送的初始设置步骤

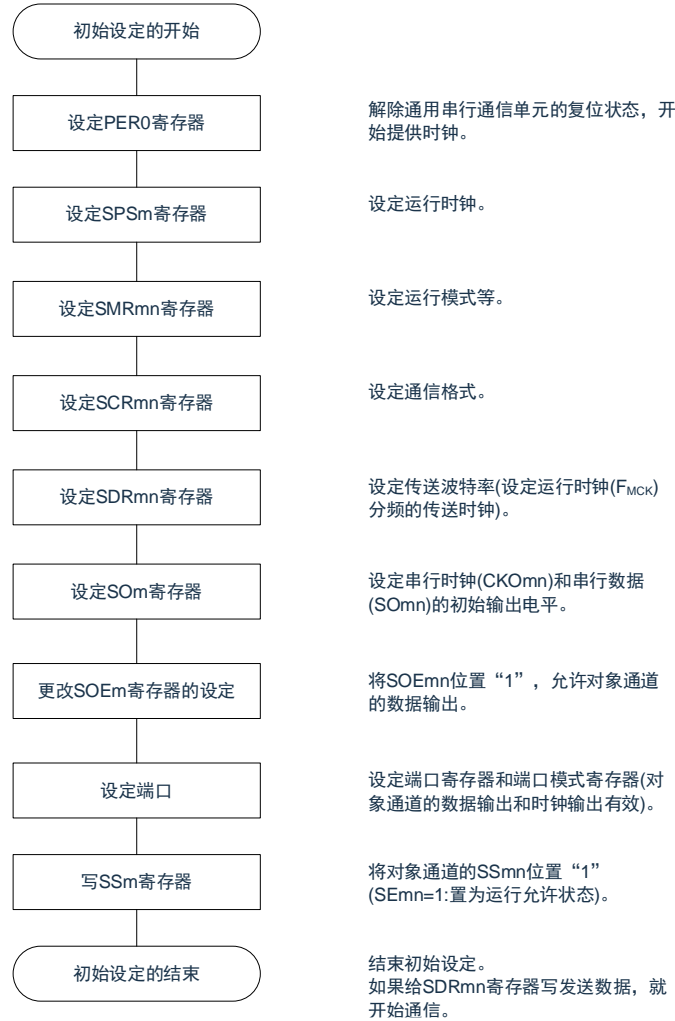


图 12-5: 主控发送的中止步骤

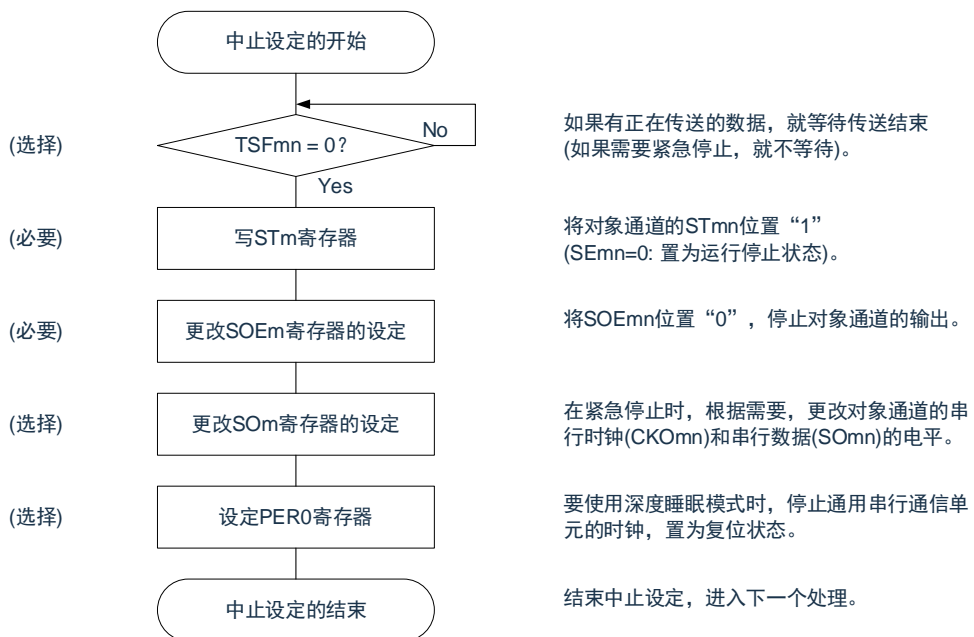


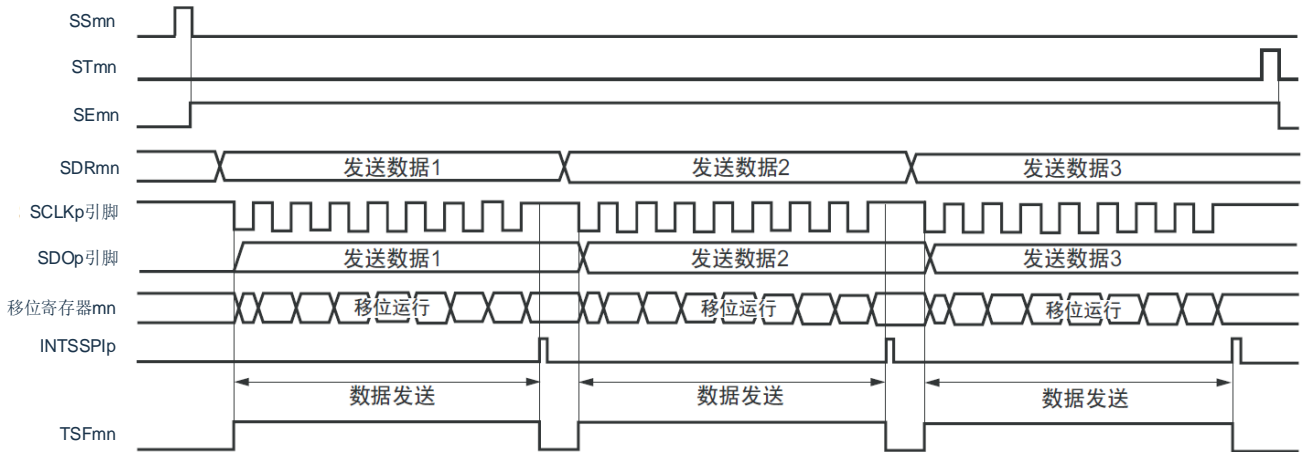
图 12-6: 重新开始主控发送的设定步骤



注意: 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。

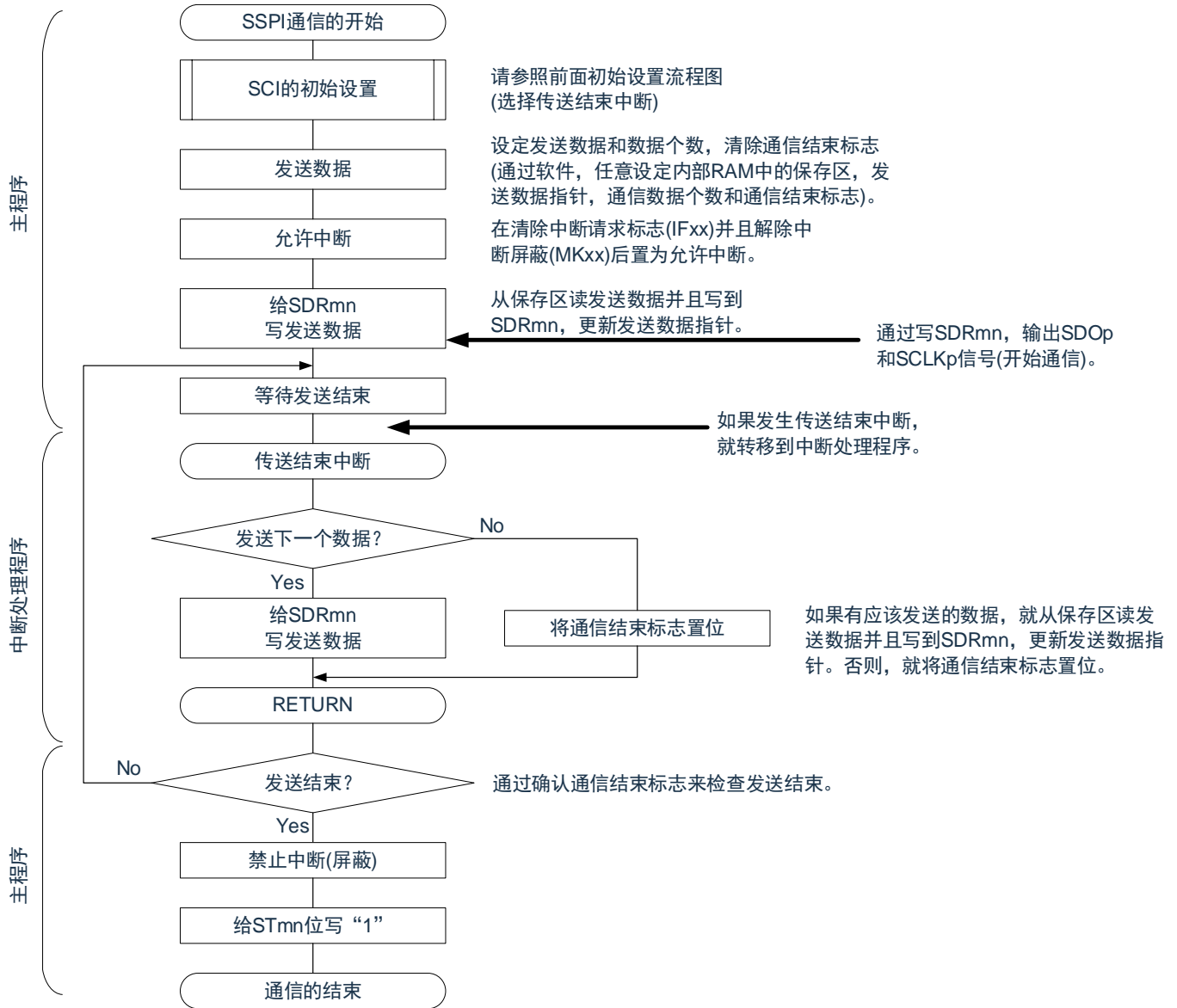
(3) 处理流程（单次发送模式）

图 12-7：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、1）；  
 p：SSPI号（p=00、01、10、11）。

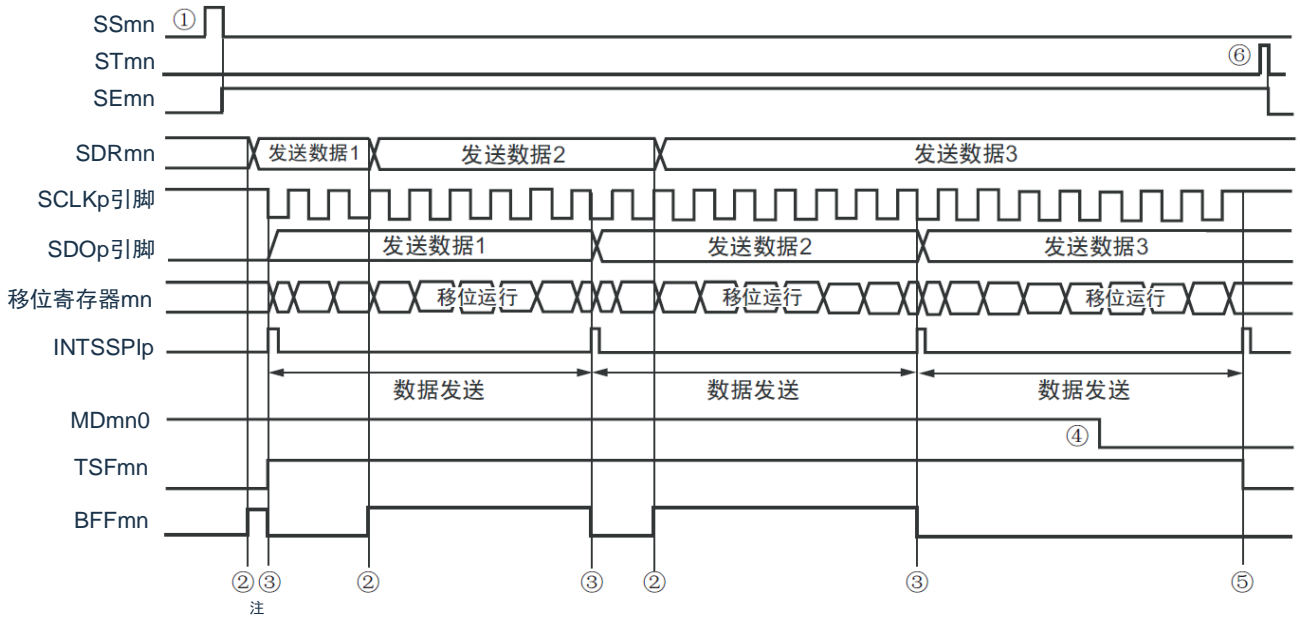
图 12-8: 主控发送（单次发送模式）的流程图





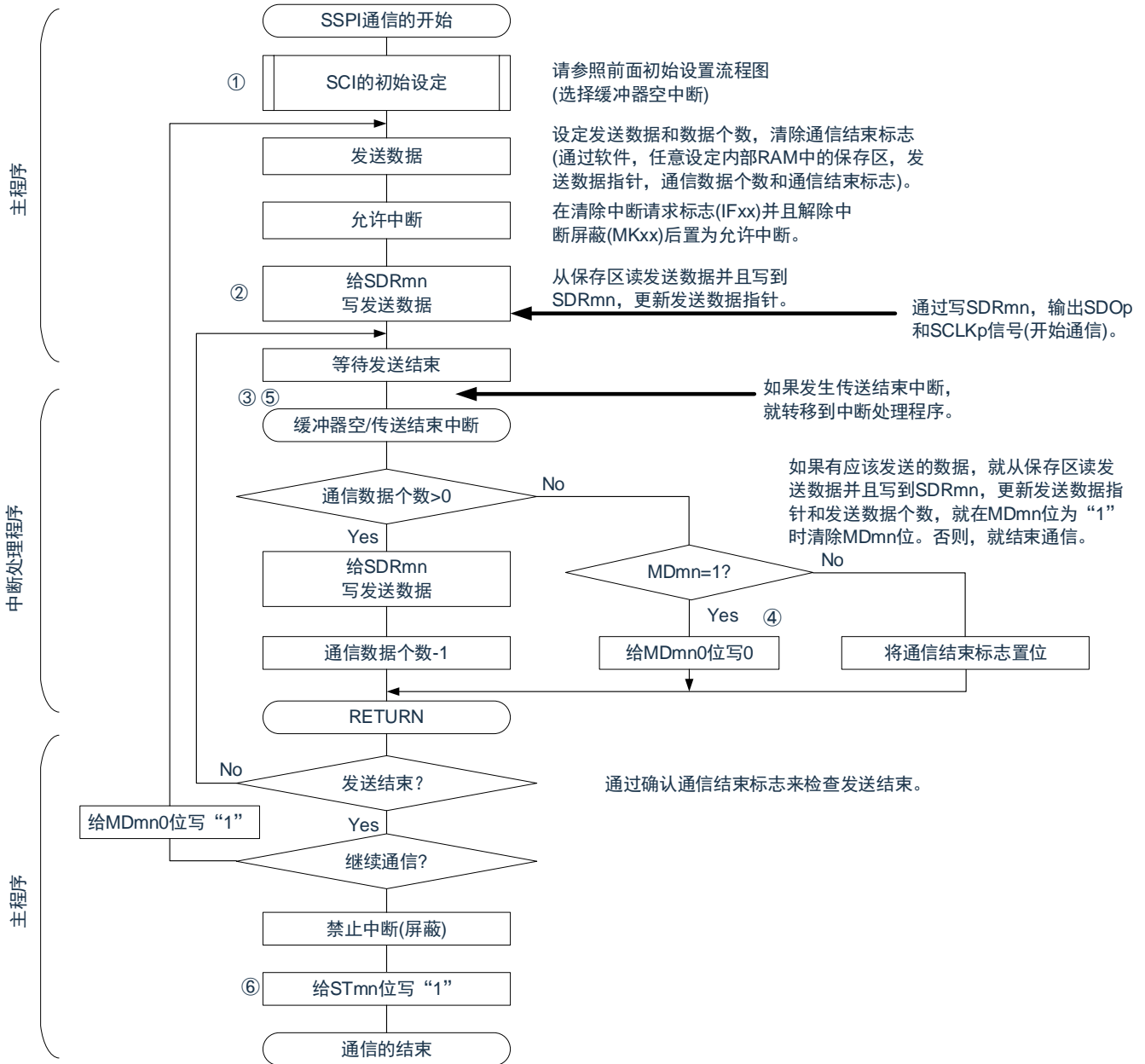
(4) 处理流程（连续发送模式）

图 12-9：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、1）；
- p：SSPI 号（p=00、01、10、11）。

图 12-10: 主控发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 12-9 主控发送（连续发送模式）的时序图”中的①~⑥。

## 12.5.2 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。

表 12-23: 主控接收

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1
使用的引脚	SCLKOI00、 SDI00	SCLKOI01、 SDI01	SCLKOI10、 SDI10	SCLKOI11、 SDI11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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 <sup>注</sup>	Max.F <sub>CLK</sub> /2[Hz] Min.F <sub>CLK</sub> /(2×2 <sup>11</sup> ×128)[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			

注意: 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);

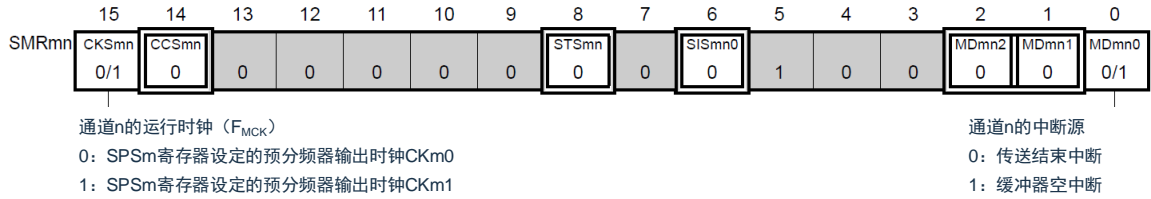
n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11)。

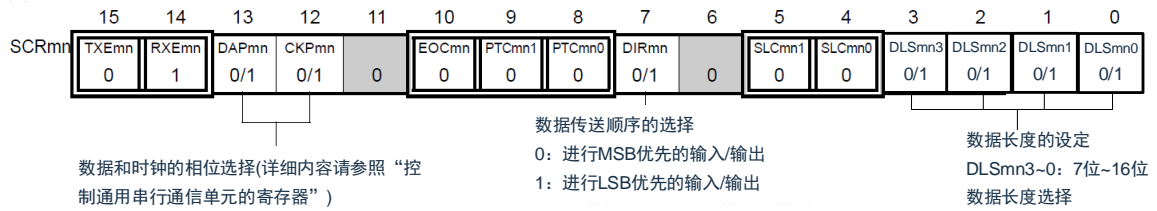
(1) 寄存器的设定

图 12-11: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)  
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)

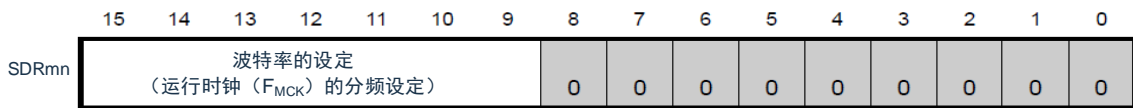


(b) 串行通信运行设定寄存器mn(SCRmn)

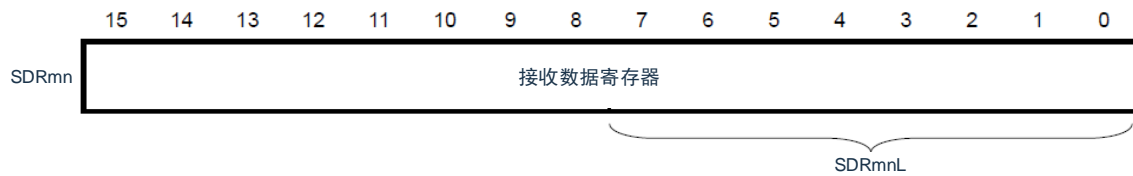


(c) 串行数据寄存器mn(SDRmn)

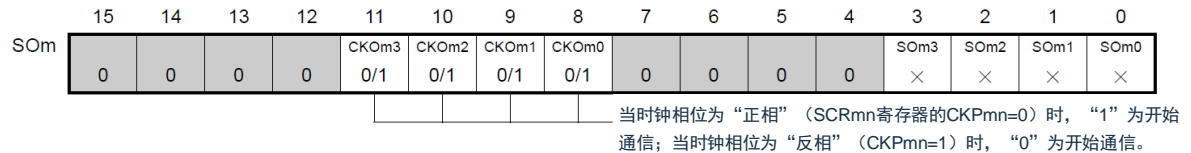
(1) 运行停止 (SEmn=0) 时



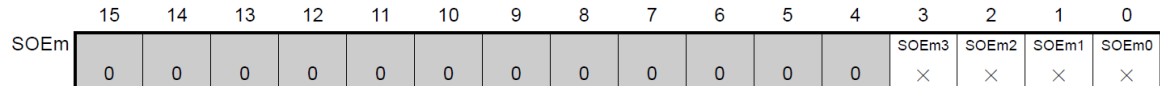
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



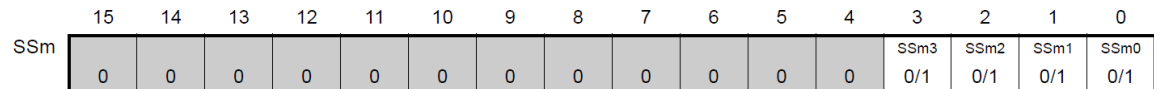
(d) 串行输出寄存器m(SOm) .....只设定对象通道的位。



(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: m: 单元号 (m=0、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

□: 在 SSPI 主控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”。

(2) 操作步骤

图 12-12: 主控接收的初始设定步骤



图 12-13: 主控接收的中止步骤

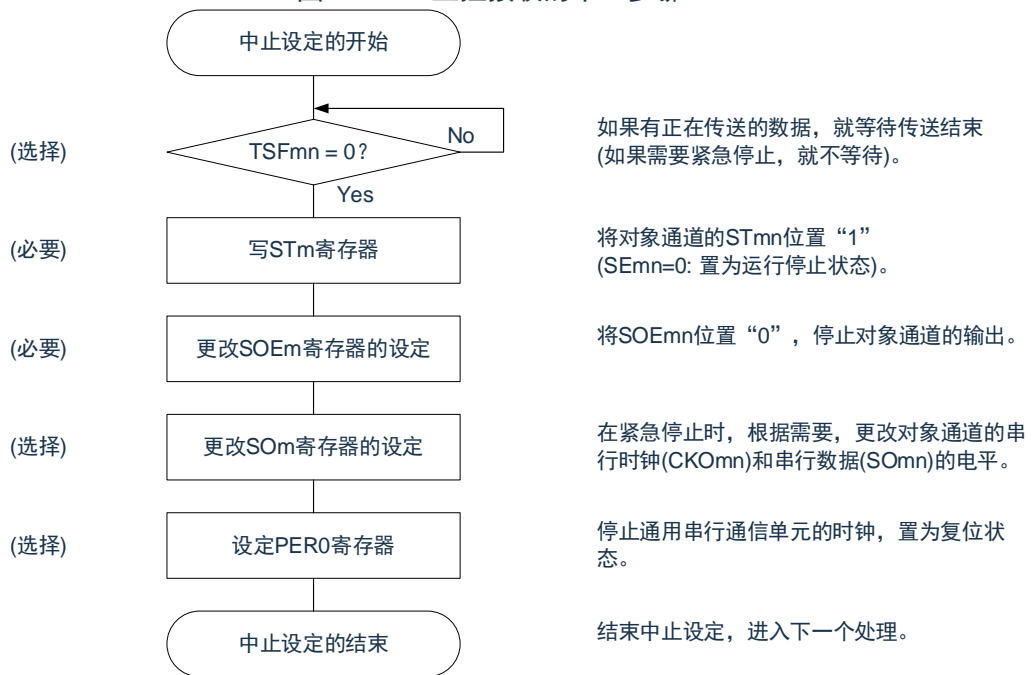


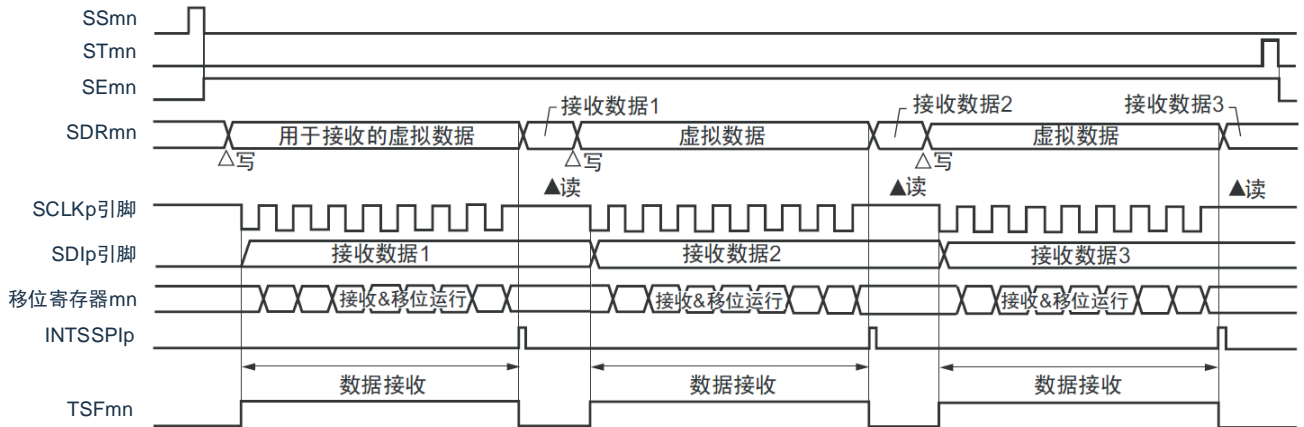
图 12-14: 重新开始主控接收的设定步骤



注意: 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）

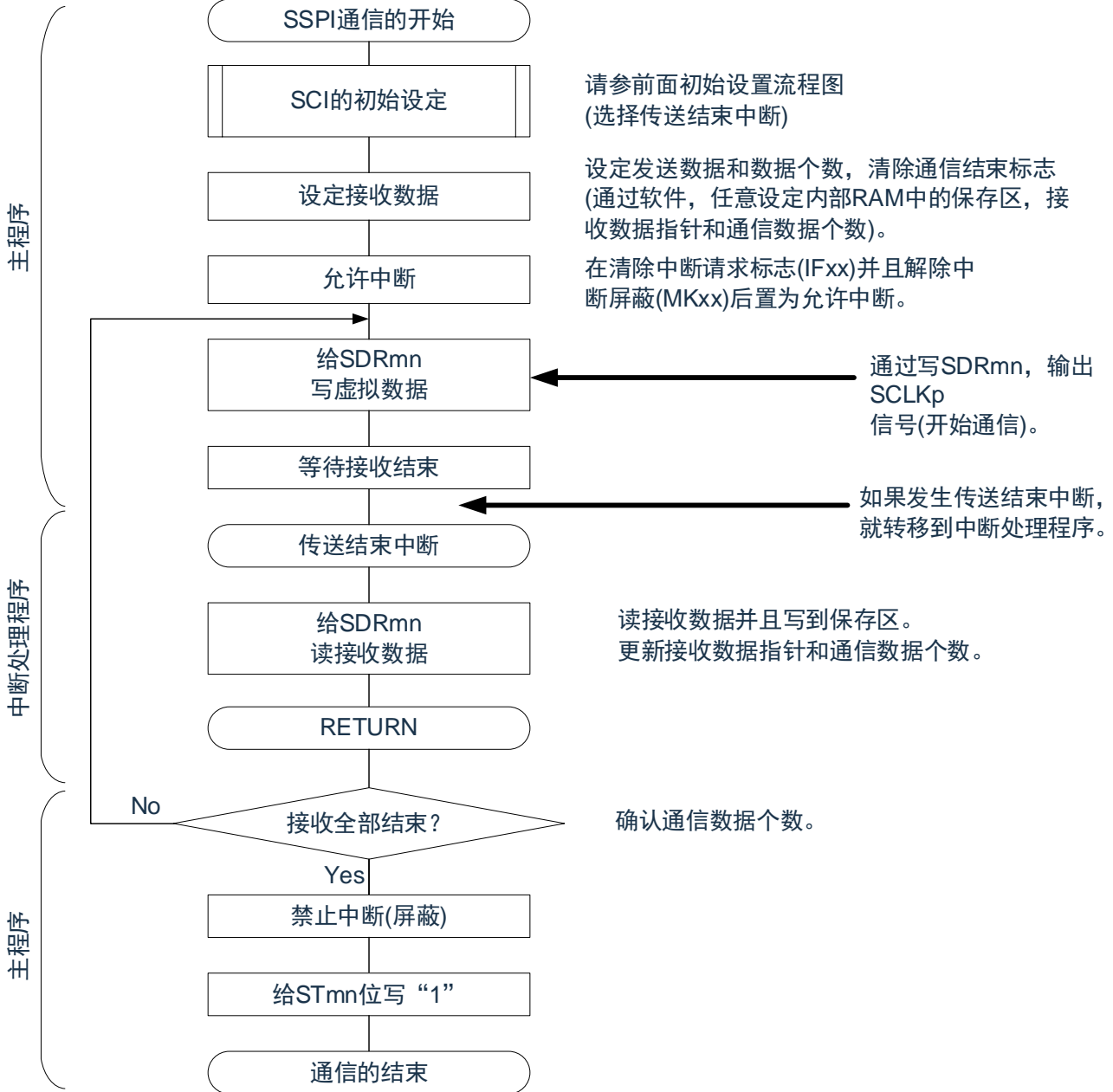
图 12-15: 主控接收（单次接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）；  
 n：通道号（n=0、1）；  
 p：SSPI号（p=00、01、10、11）。

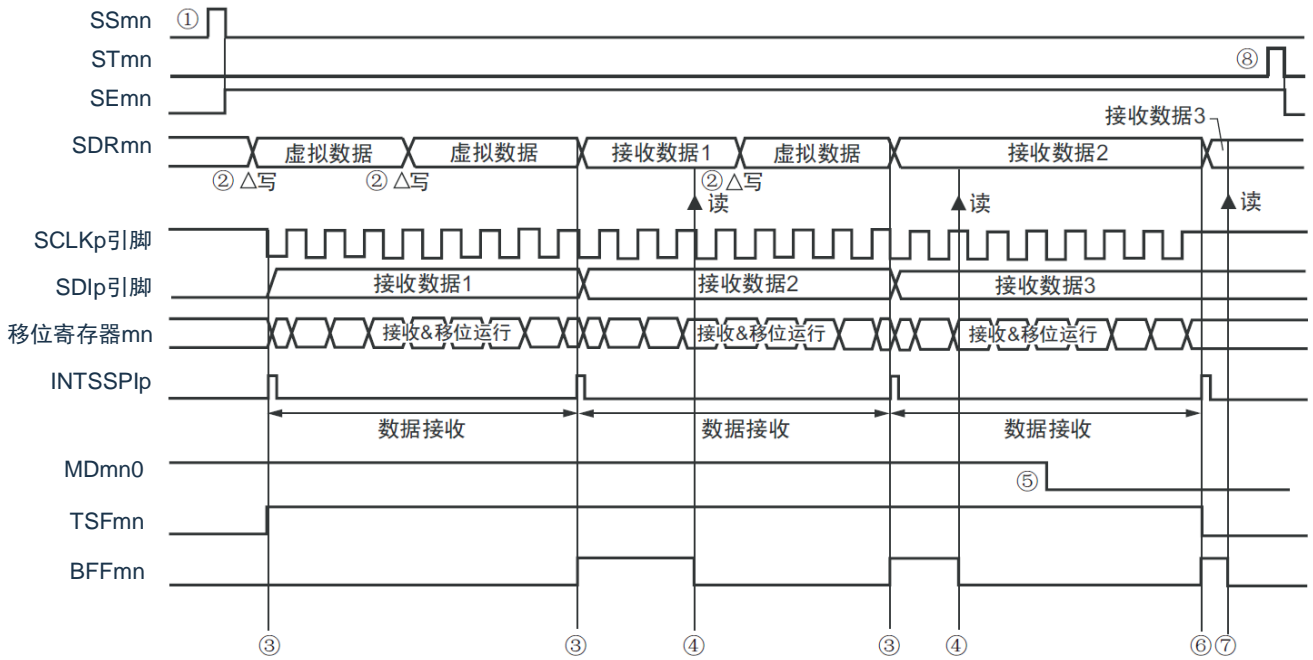


图 12-16: 主控接收 (单次接收模式) 的流程图



(4) 处理流程（连续接收模式）

图 12-17：主控接收（连续接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

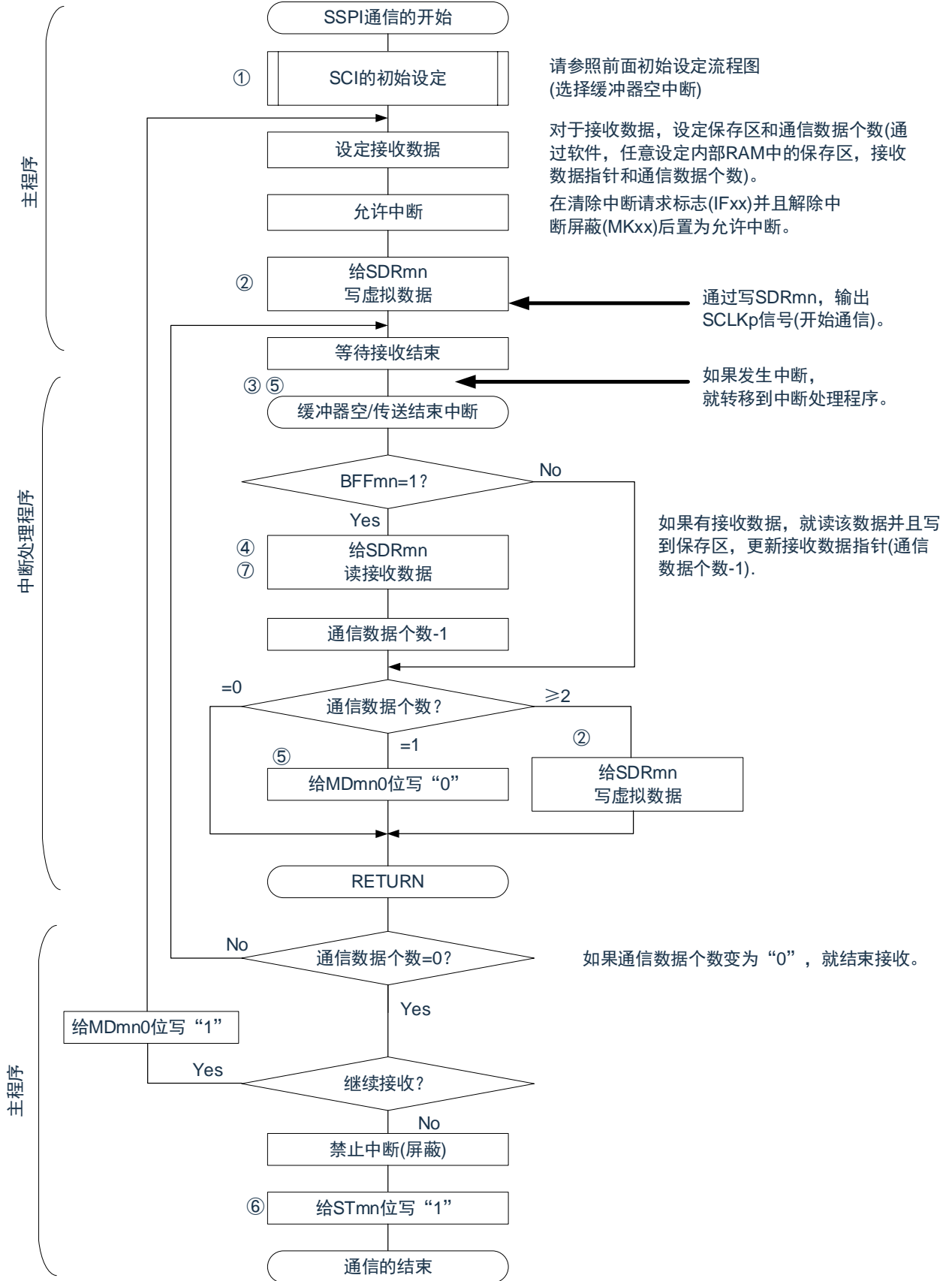


注意：即使在运行过程中也能改写 MDmn0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写；

备注：

1. 图中的①~⑧对应“图 12-18 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、1)；  
p: SSPI 号 (p=00、01、10、11)。

图 12-18: 主控接收（连续接收模式）的流程图



备注: 图中的①~⑧对应“图 12-17 主控接收（连续接收模式）的时序图”中的①~⑧。

### 12.5.3 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。

表 12-24：主控的发送和接收

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1
使用的引脚	SCLKOI00、SDI00、SDO00	SCLKOI01、SDI01、SDO01	SCLKOI10、SDI10、SDO10	SCLKOI11、SDI11、SDO11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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 <sup>注</sup>	Max.F <sub>CLK</sub> /2[Hz] Min.F <sub>CLK</sub> /(2×2 <sup>11</sup> ×128)[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			

注意：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）；

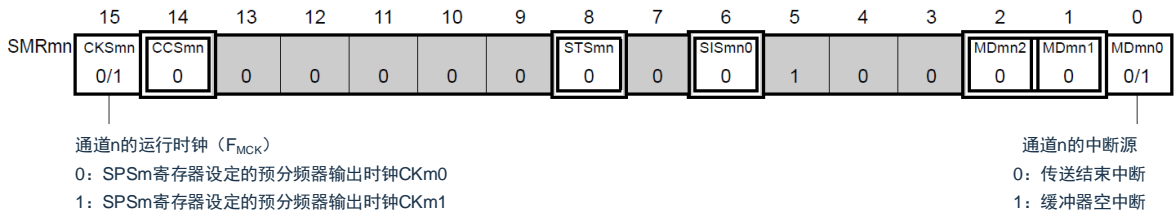
n：通道号（n=0、1）；

p：SSPI 号（p=00、01、10、11）。

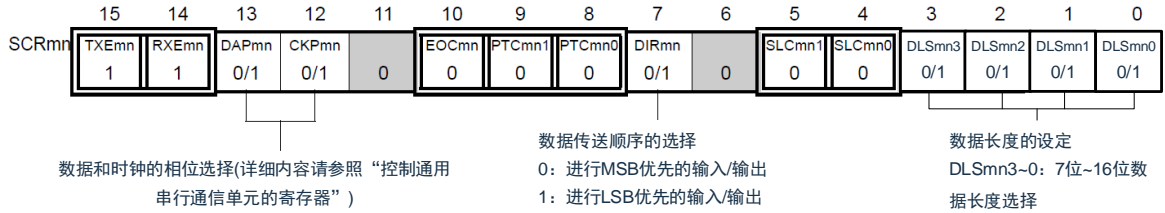
(1) 寄存器的设定

图 12-19: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)  
主控发送和接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

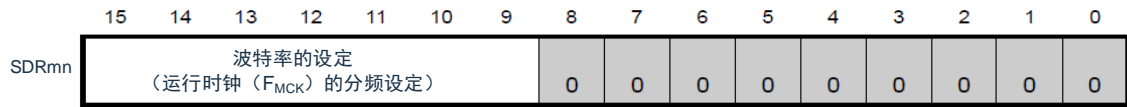


(b) 串行通信运行设定寄存器mn(SCRmn)

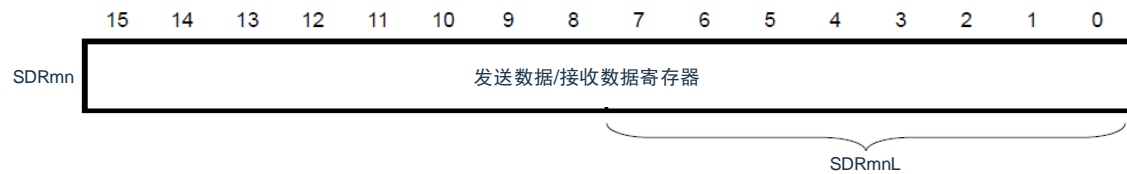


(c) 串行数据寄存器mn(SDRmn)

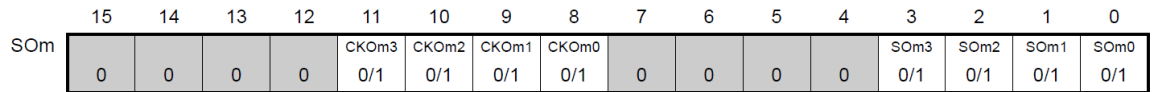
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)

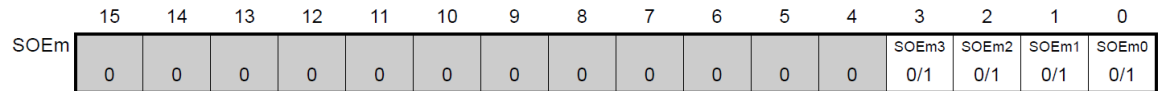


(d) 串行输出寄存器m(SOm) .....只设定对象通道的位。

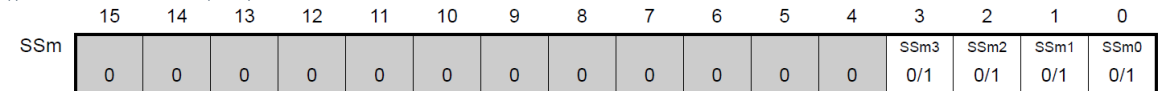


当时钟相位为“正相”(SCRmn寄存器的CKPmn=0)时,“1”为开始通信;当时钟相位为“反相”(CKPmn=1)时,“0”为开始通信。

(e) 串行输出允许寄存器m(SOEm) .....只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) .....只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1) ;

n: 通道号 (n=0、1) ;

p: SSPI 号 (p=00、01、10、11) ;

□ : 在 SSPI 主控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 12-20: 主控发送和接收的初始设定步骤

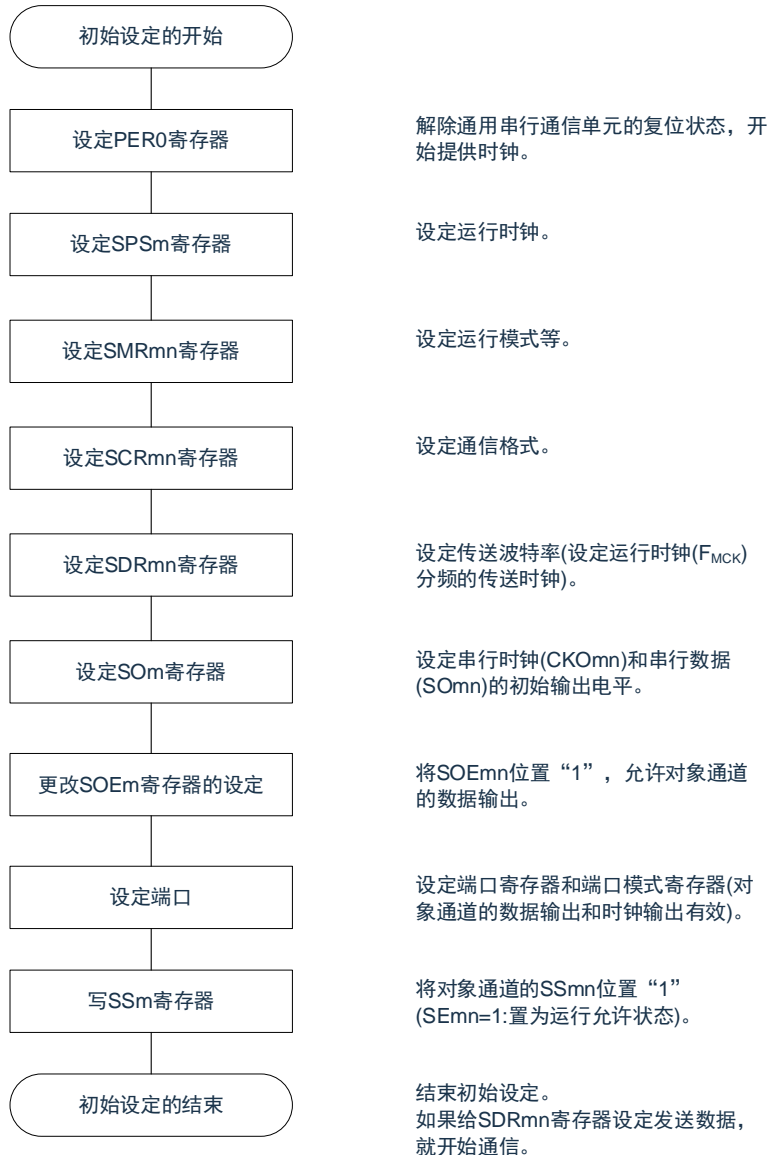


图 12-21: 主控发送和接收的中止步骤

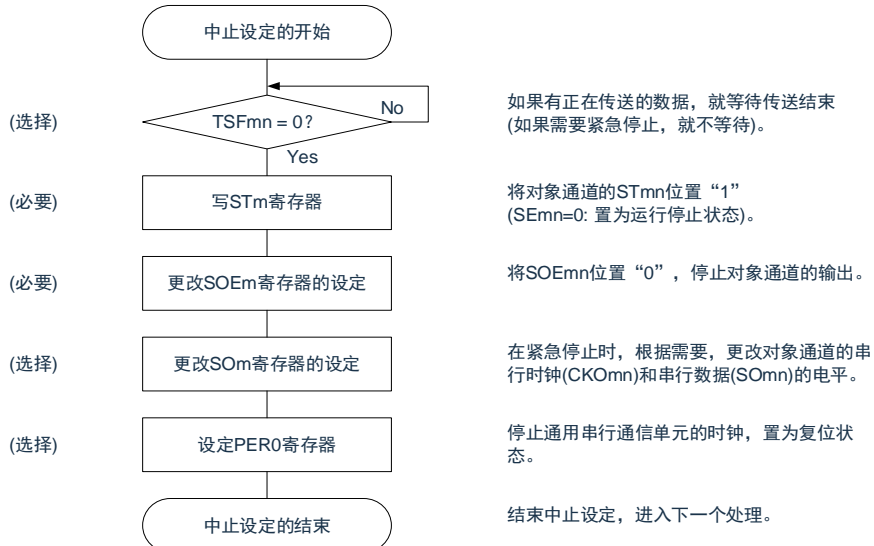
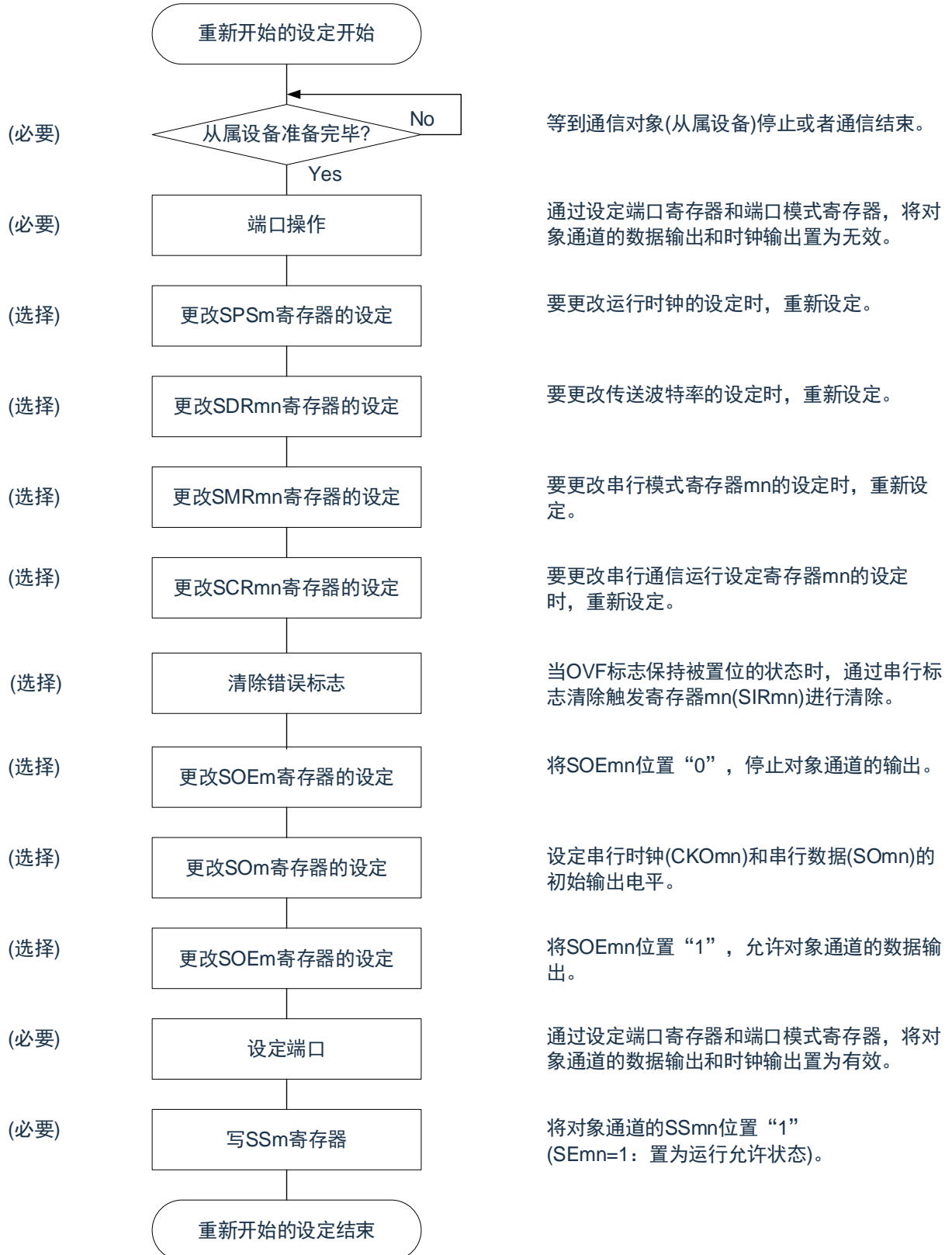
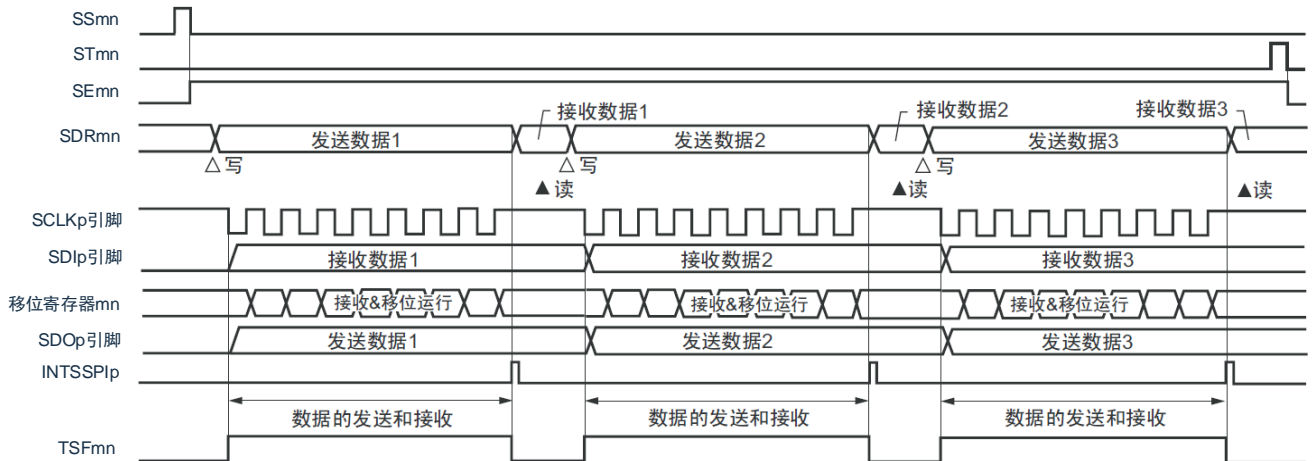


图 12-22：重新开始主控发送和接收的设定步骤



(3) 处理流程（单次发送和接收模式）

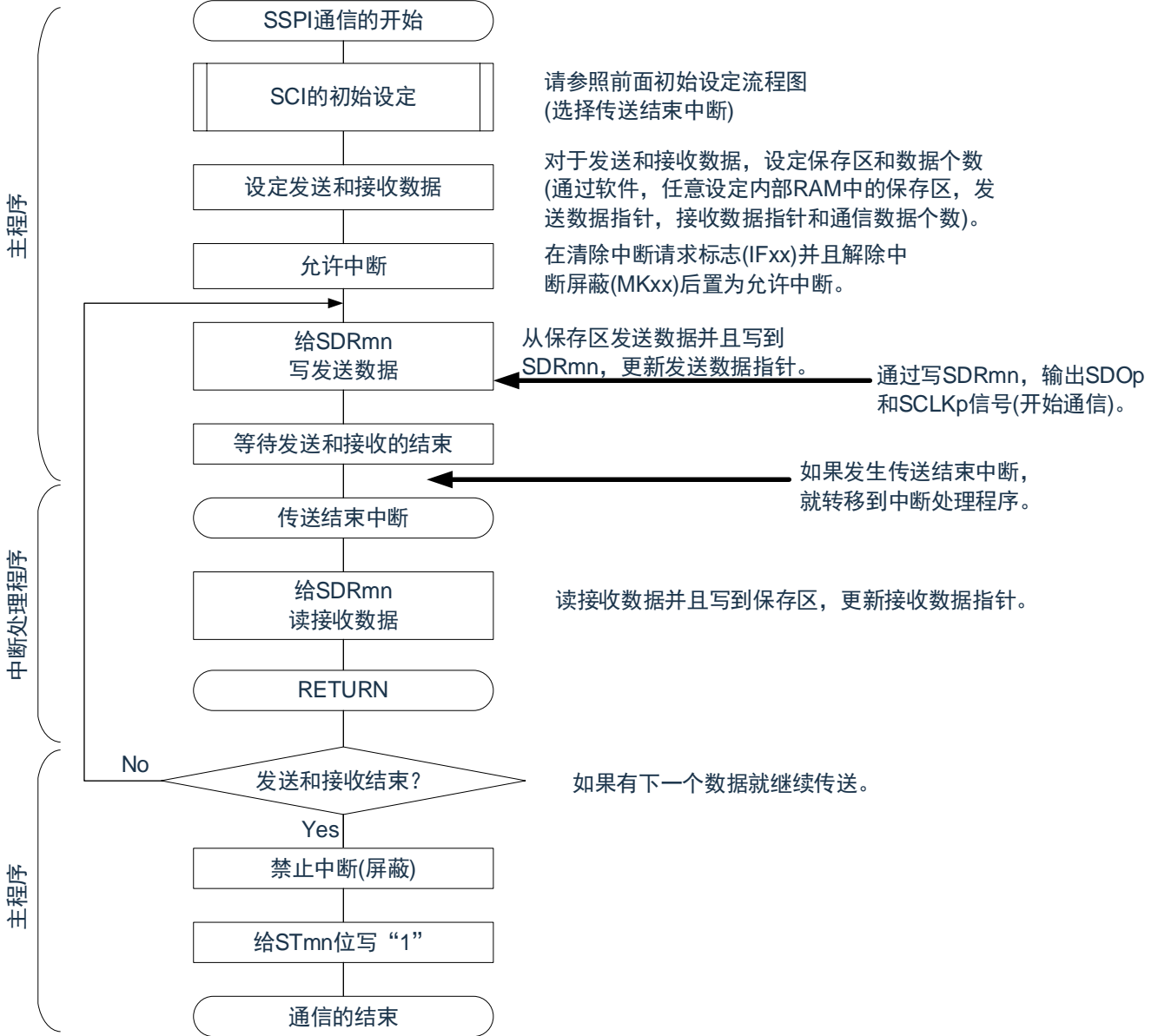
图 12-23: 主控发送和接收（单次发送和接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）



备注：m: 单元号 (m=0、1) ;  
 n: 通道号 (n=0、1) ;  
 p: SSPI 号 (p=00、01、10、11) 。

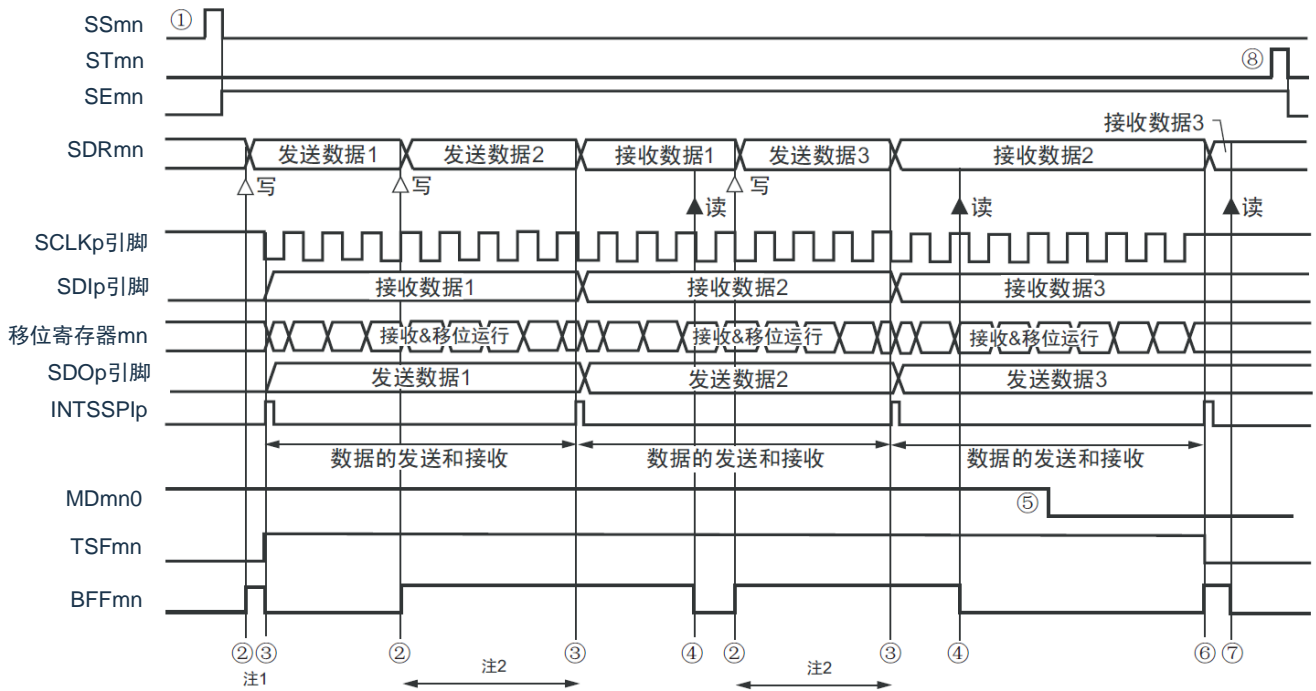


图 12-24: 主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图 12-25: 主控发送和接收（连续发送和接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）



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

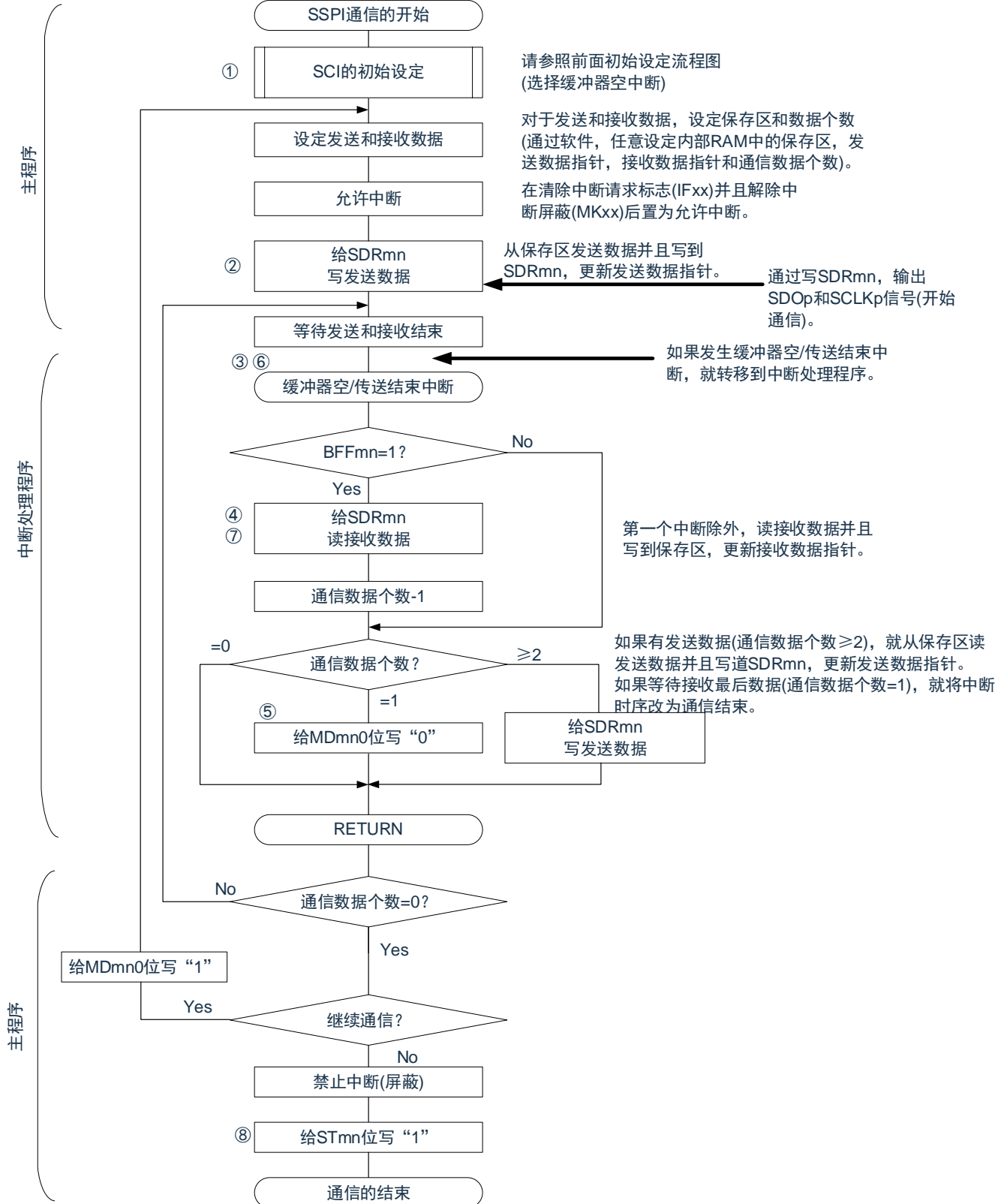
注 2: 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

注意: 即使在运行中也能改写串行模式寄存器 mn (SMRmn) 的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注:

1. 图中的①~⑧对应“图 12-26 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、1) ;  
p: SSPI 号 (p=00、01、10、11) 。

图 12-26: 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 12-25 主控发送和接收 (连续发送和接收模式) 的时序图”中的①~⑧。

## 12.5.4 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微控制器将数据发送到其他设备的运行。

表 12-25: 从属发送

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0的通道0	SCI0的通道1	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、SDO00	SCLKOI01、SDO01	SCLKOI10、SDO10	SCLKOI11、SDO11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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	$\text{Max.F}_{\text{MCK}}/6[\text{Hz}]^{\text{注}1,2}$			
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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11 引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为  $F_{\text{MCK}}/6[\text{Hz}]$ ;

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

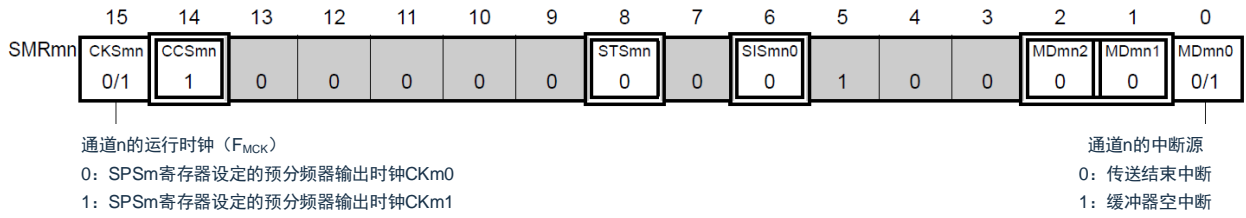
备注:

1.  $F_{\text{MCK}}$ : 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)。

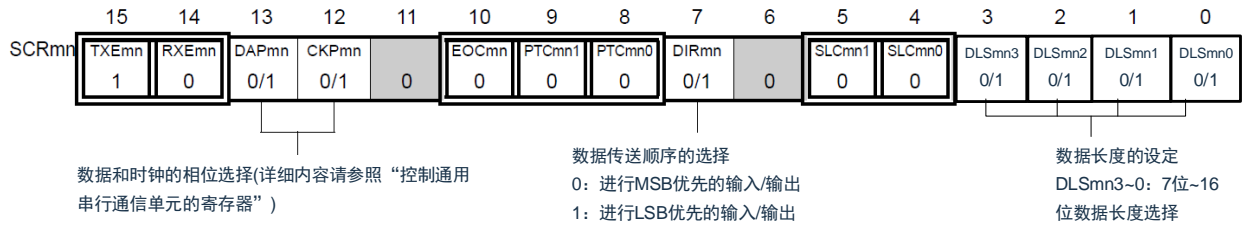
(1) 寄存器的设定

图 12-27: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)  
从属发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

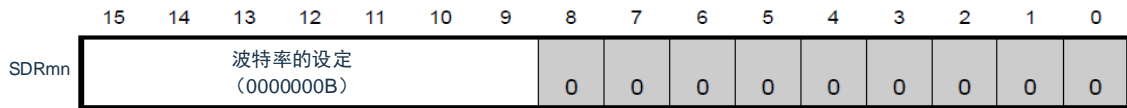


(b) 串行通信运行设定寄存器mn(SCRmn)

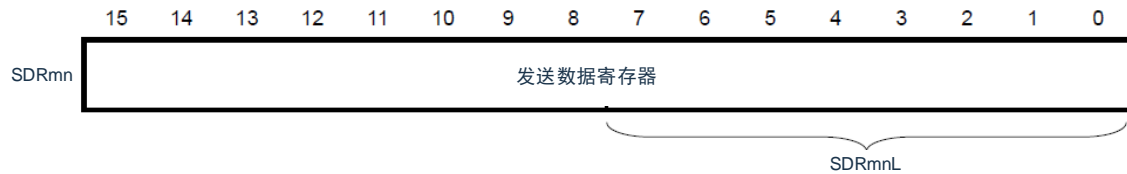


(c) 串行数据寄存器mn(SDRmn)

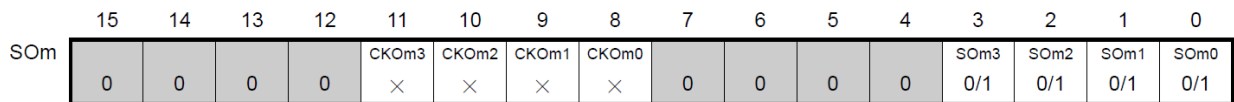
(1) 运行停止 (SEmn=0) 时



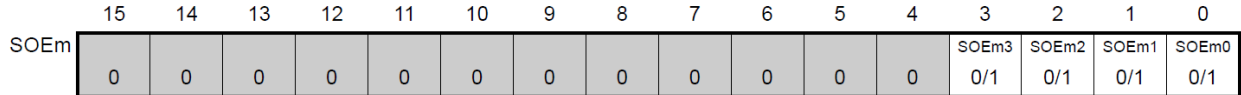
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



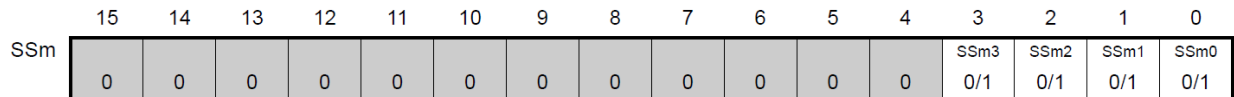
(d) 串行输出寄存器m(SOm) ...只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ...只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ...只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1) ;

n: 通道号 (n=0、1) ;

p: SSPI 号 (p=00、01、10、11) ;

□: 在 SSPI 主控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”。

(2) 操作步骤

图 12-28: 从属发送的初始设定步骤



图 12-29: 从属发送的中止步骤

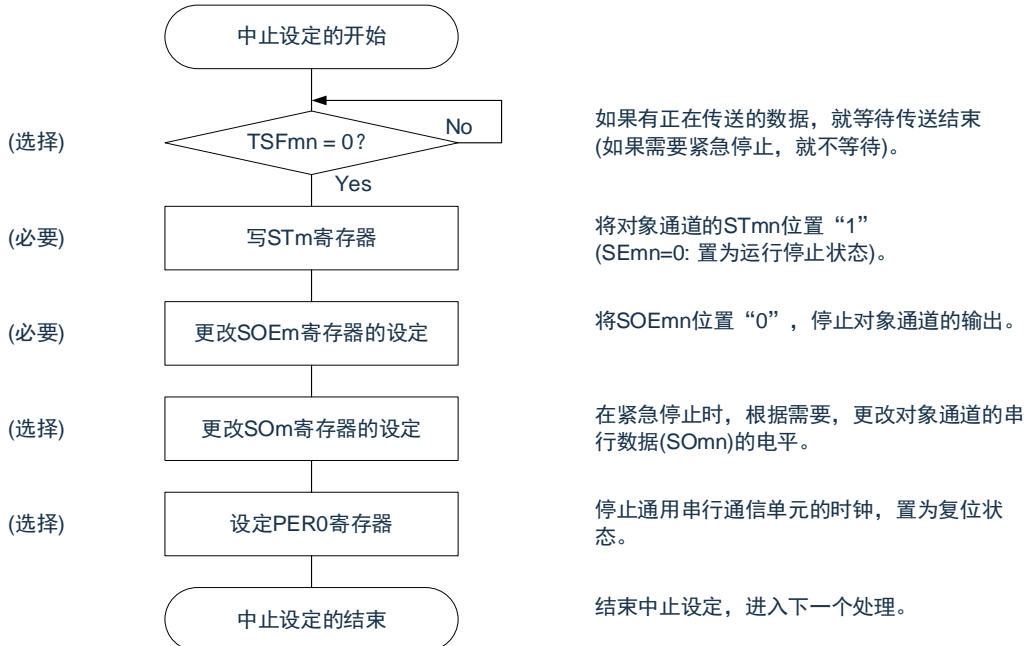
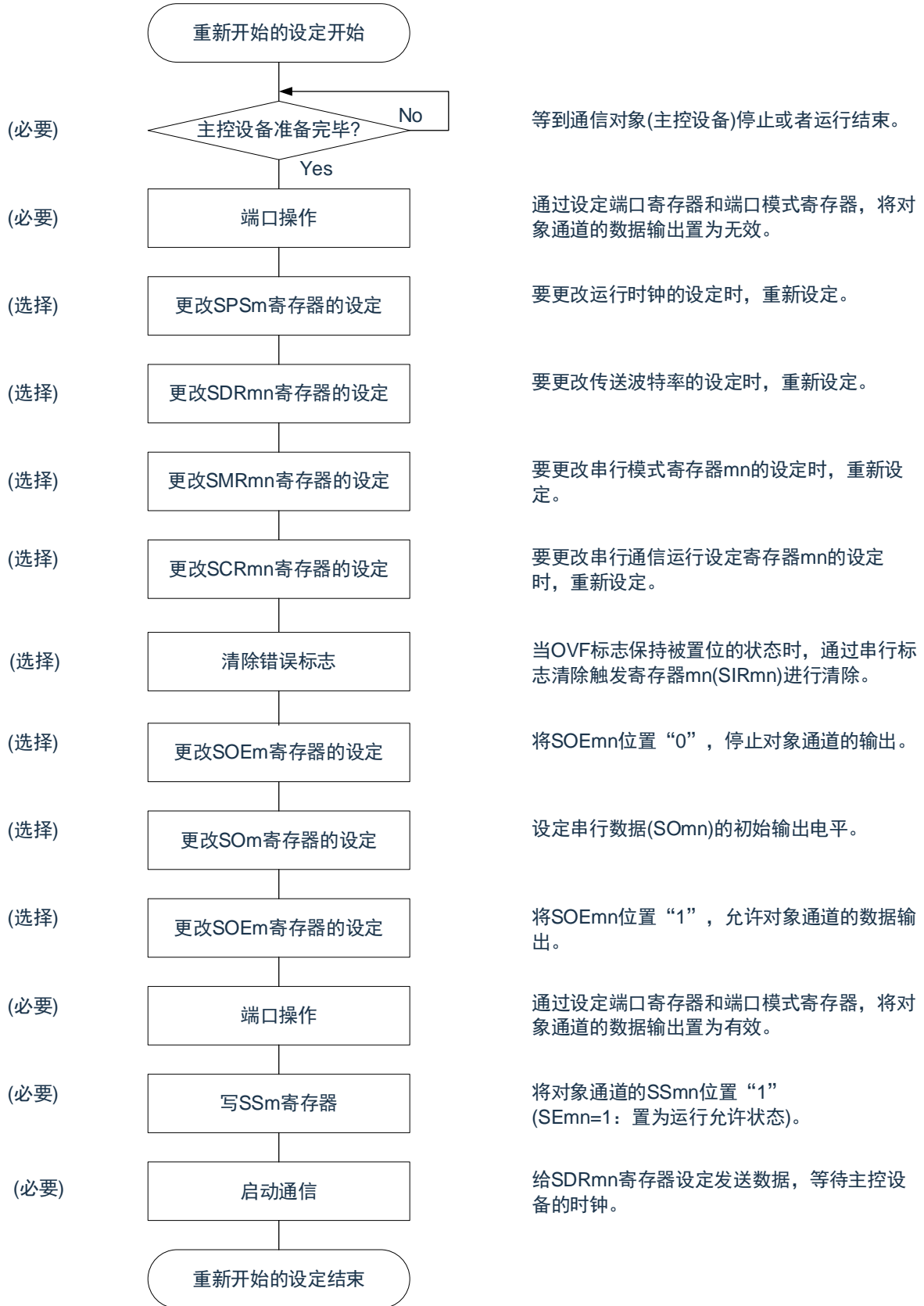


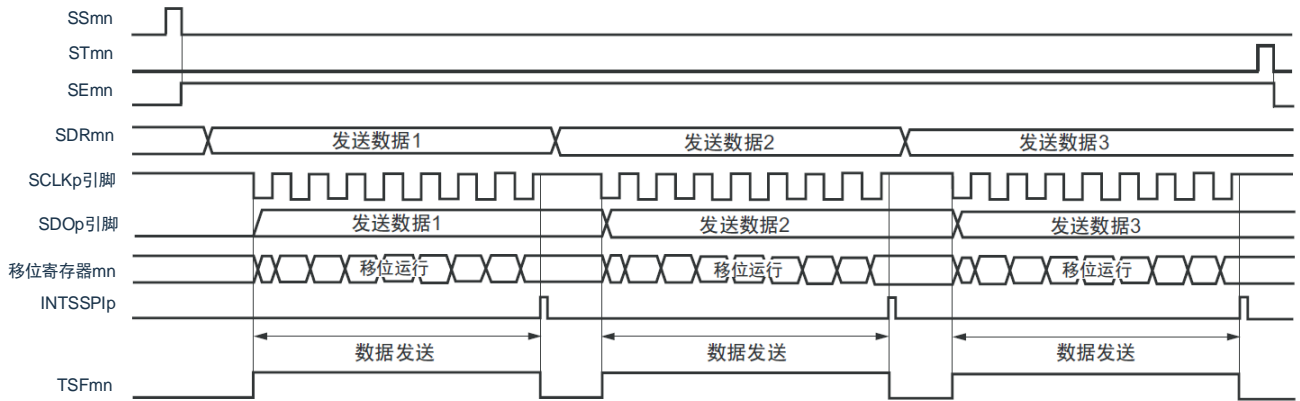
图 12-30: 重新开始从属发送的设定步骤



注意: 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）

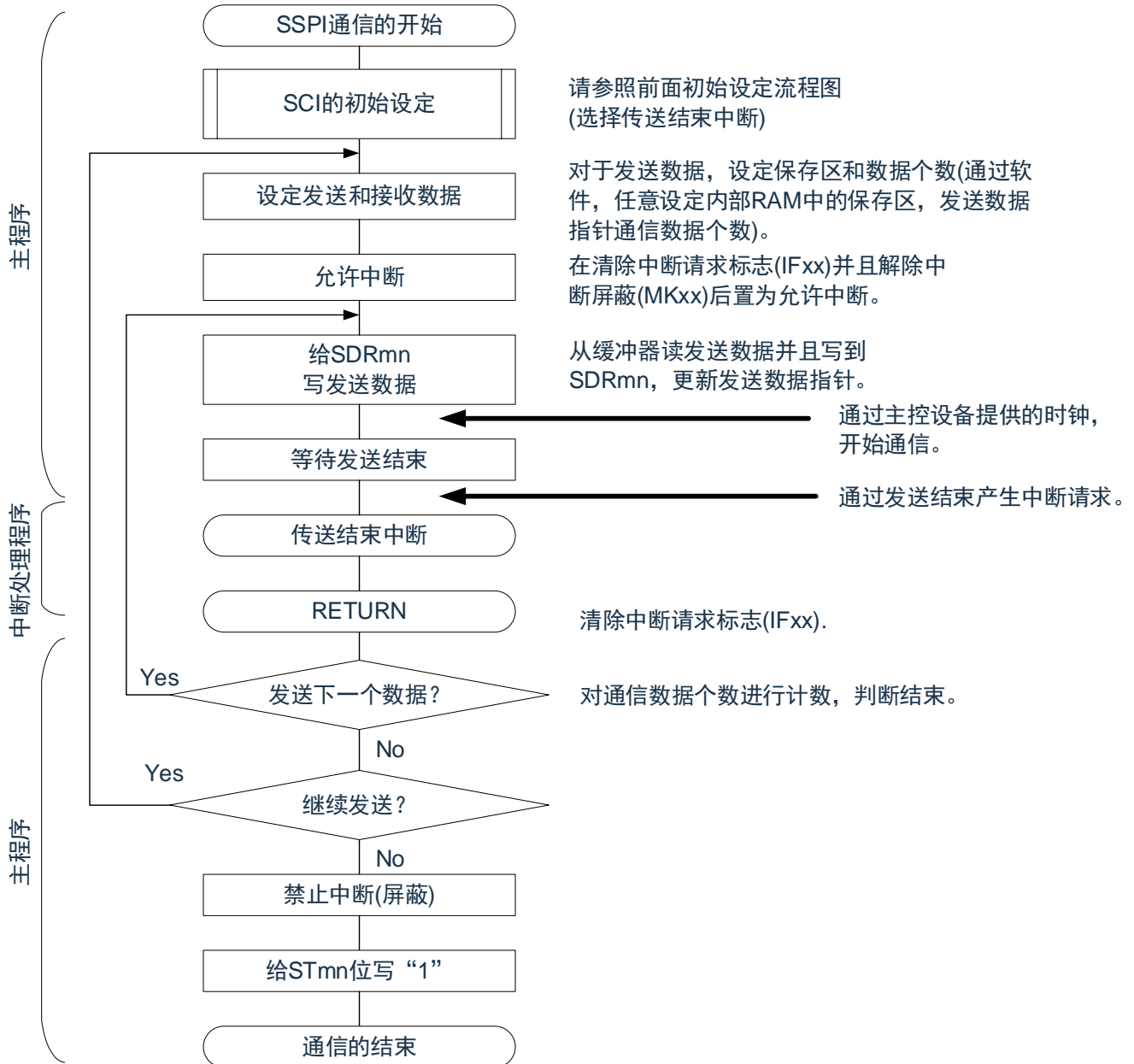
图 12-31：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）；  
 n：通道号（n=0、1）；  
 p：SSPI号（p=00、01、10、11）。

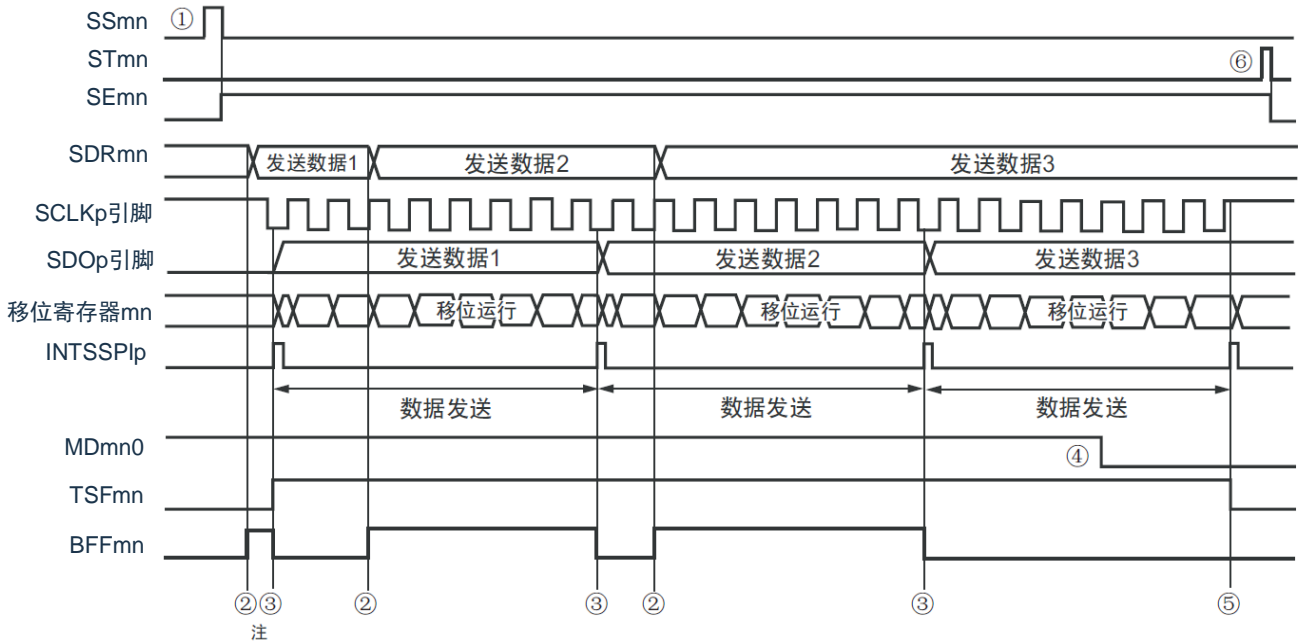


图 12-32: 从属发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 12-33：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。

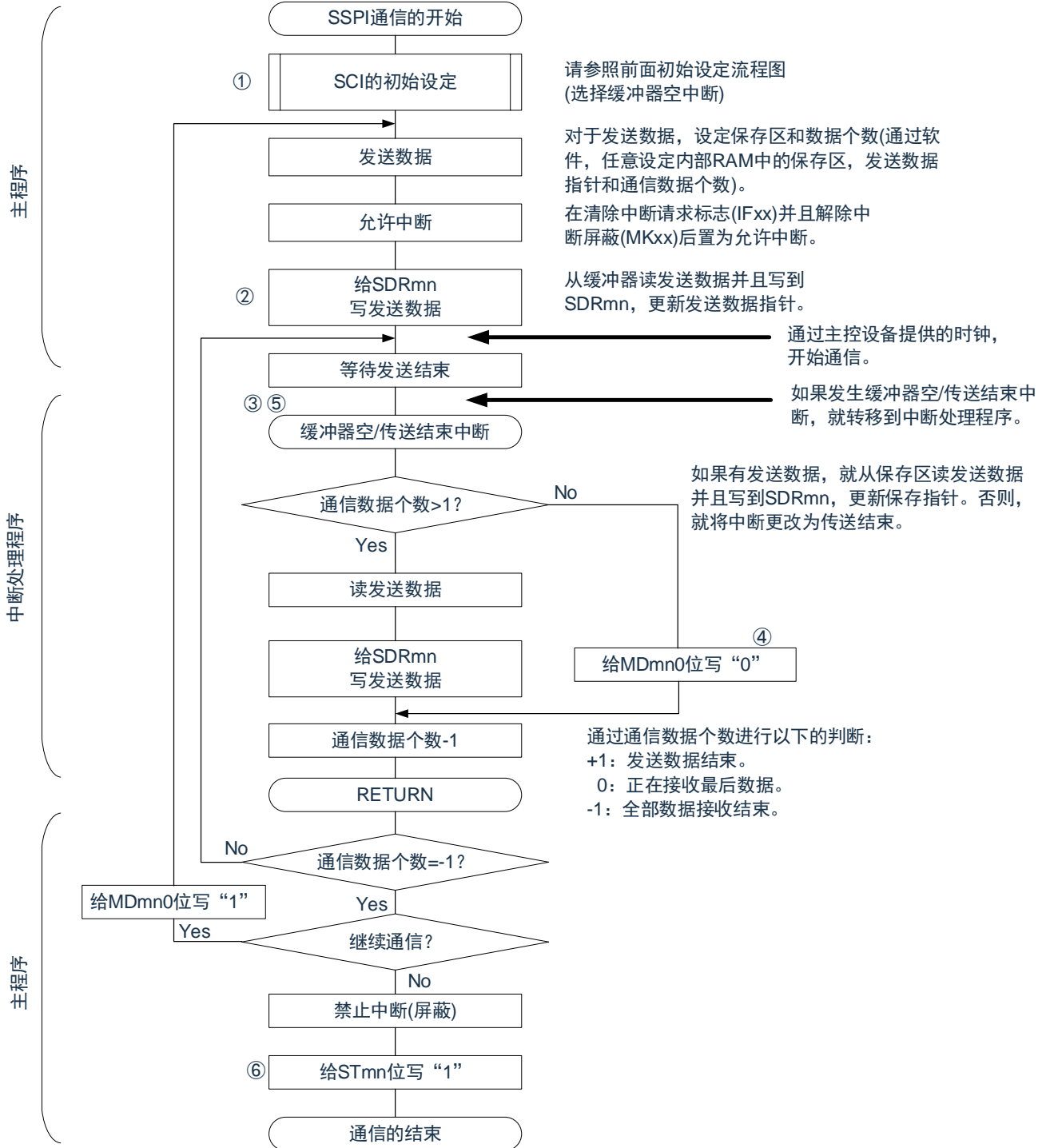
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0、1）；

n：通道号（n=0、1）；

p：SSPI 号（p=00、01、10、11）。

图 12-34: 从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图12-33 从属发送（连续发送模式）的时序图”中的①~⑥。

## 12.5.5 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。

表 12-26：从属接收

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0的通道0	SCI0的通道1	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、SDI00	SCLKOI01、SDI01	SCLKOI10、SDI10	SCLKOI11、SDI11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $F_{\text{MCK}}/6[\text{Hz}]$ ；

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

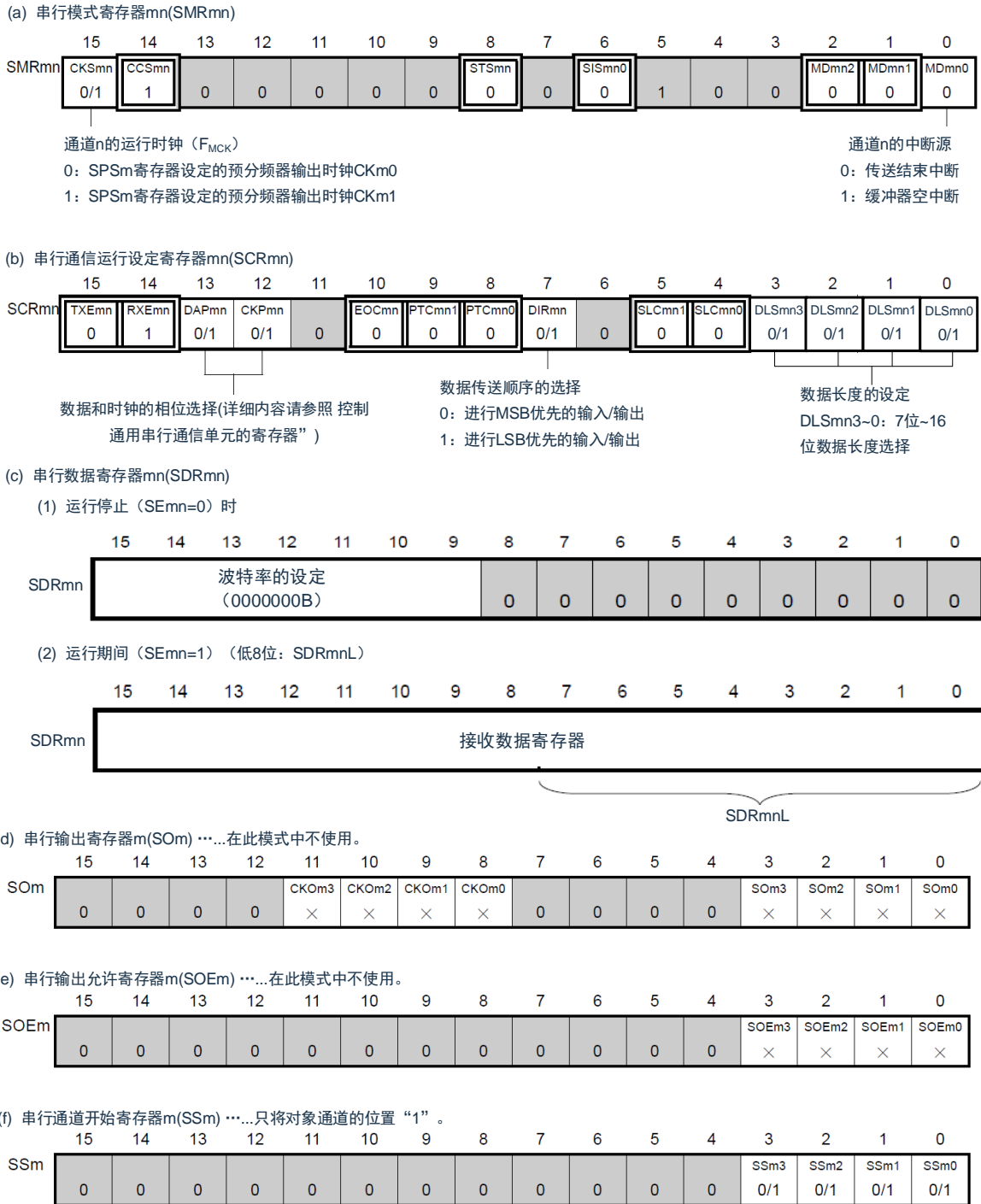
$F_{\text{MCK}}$ ：对象通道的运行时钟频率；

m：单元号（m=0、1）；

n：通道号（n=0、1）。

(1) 寄存器的设定

图 12-35: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11)  
从属接收时的寄存器设定内容例子



备注: m: 单元号 (m=0、1) ;  
 n: 通道号 (n=0、1) ;  
 p: SSPI 号 (p=00、01、10、11) ;  
 □ : 在 SSPI 主控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”。

(2) 操作步骤

图 12-36: 从属接收的初始设定步骤

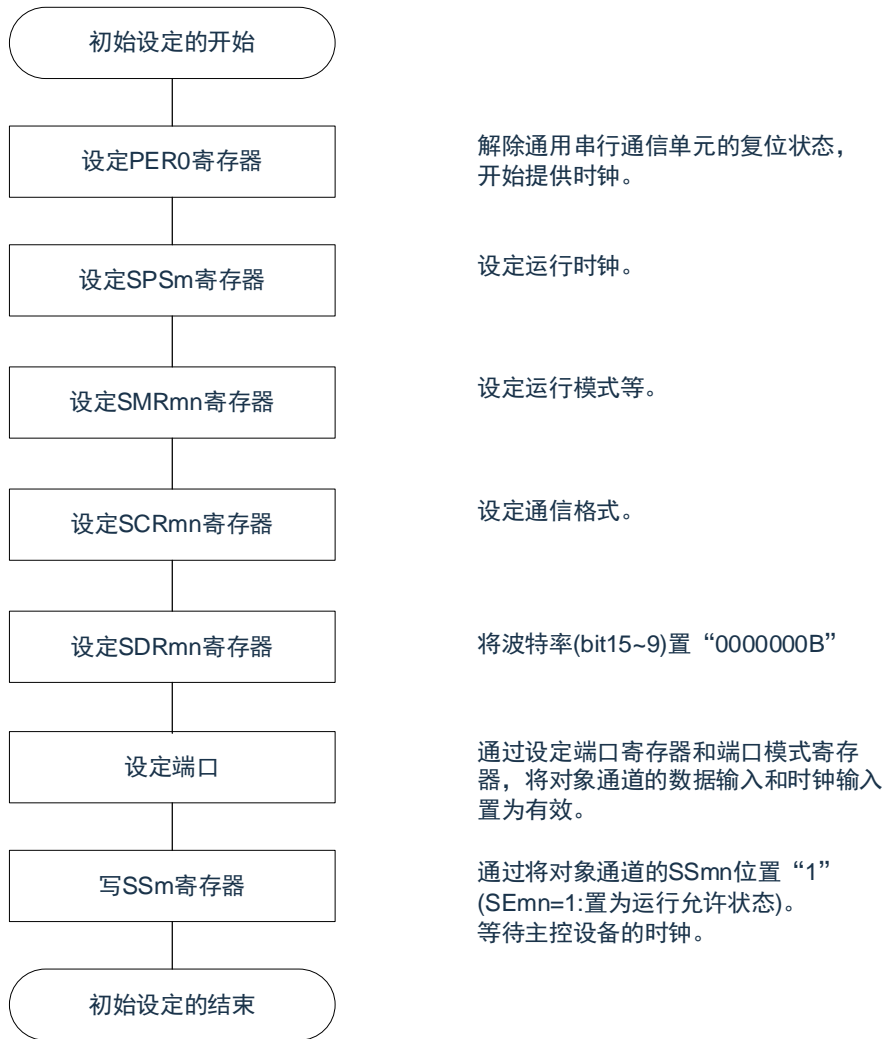


图 12-37: 从属接收的中止步骤

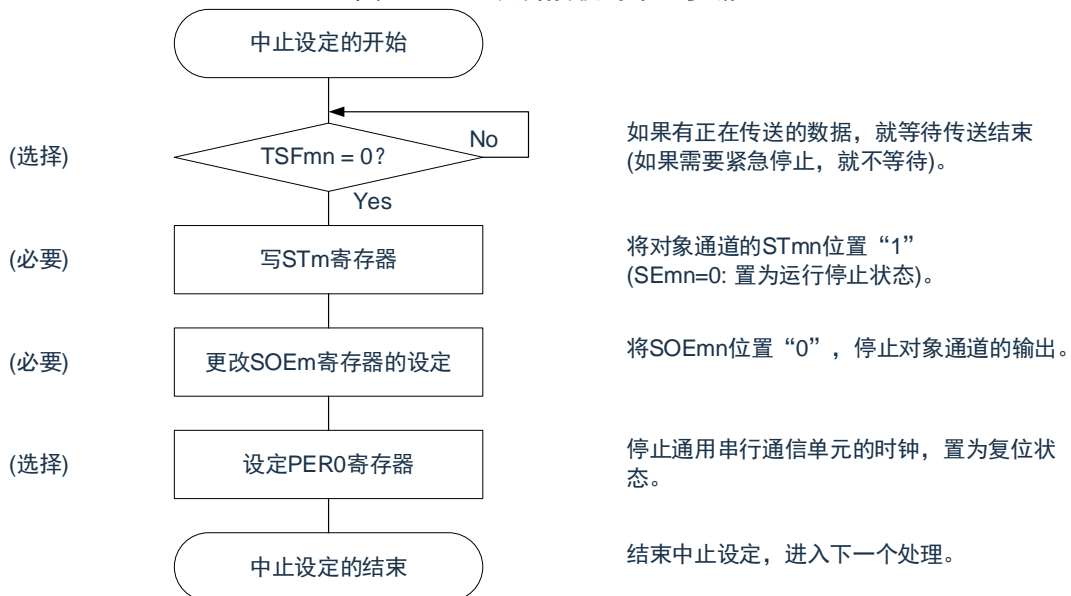


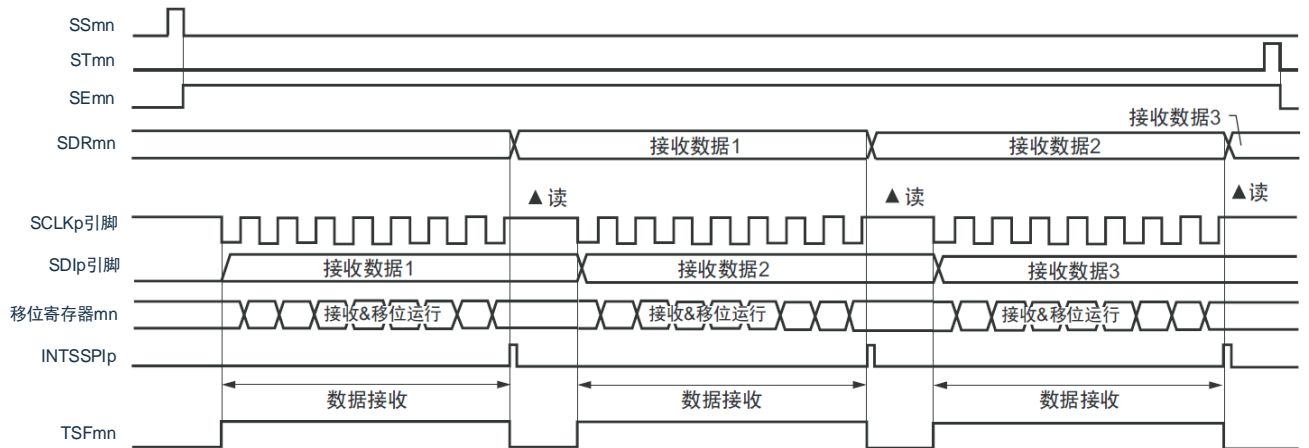
图 12-38: 重新开始从属接收的设定步骤



注意: 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）

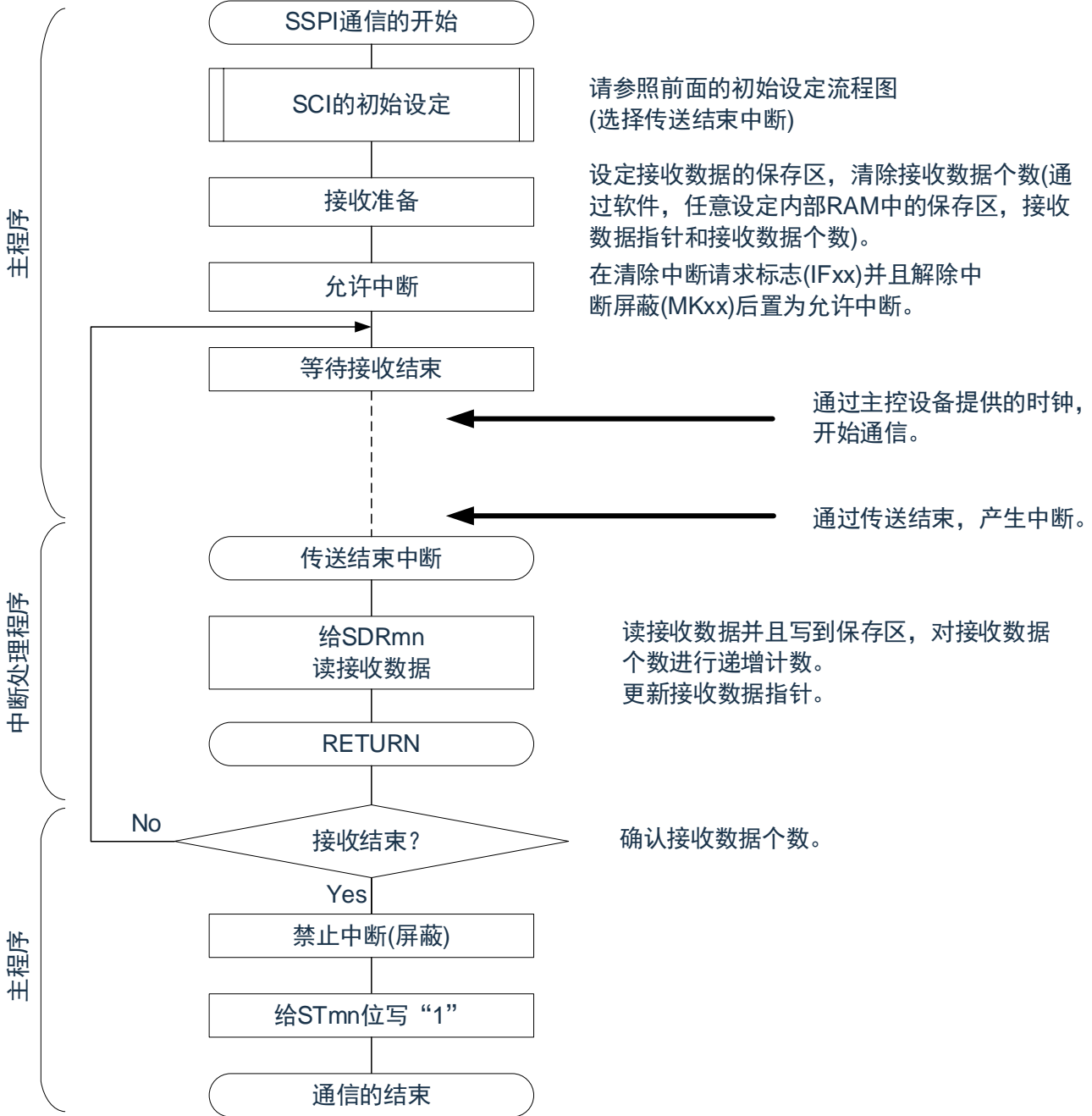
图 12-39：从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1）；  
 n：通道号（n=0、1）；  
 p：SSPI号（p=00、01、10、11）。



图 12-40: 从属接收 (单次接收模式) 的流程图



## 12.5.6 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。

表 12-27：从属的发送和接收

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11
对象通道	SCI0的通道0	SCI0的通道1	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、SDI00、SDO00	SCLKOI01、SDI01、SDO01	SCLKOI10、SDI10、SDO10	SCLKOI11、SDI11、SDO11
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11
	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 <sub>MCK</sub> /6[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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11 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  $F_{MCK}/6[Hz]$ ；

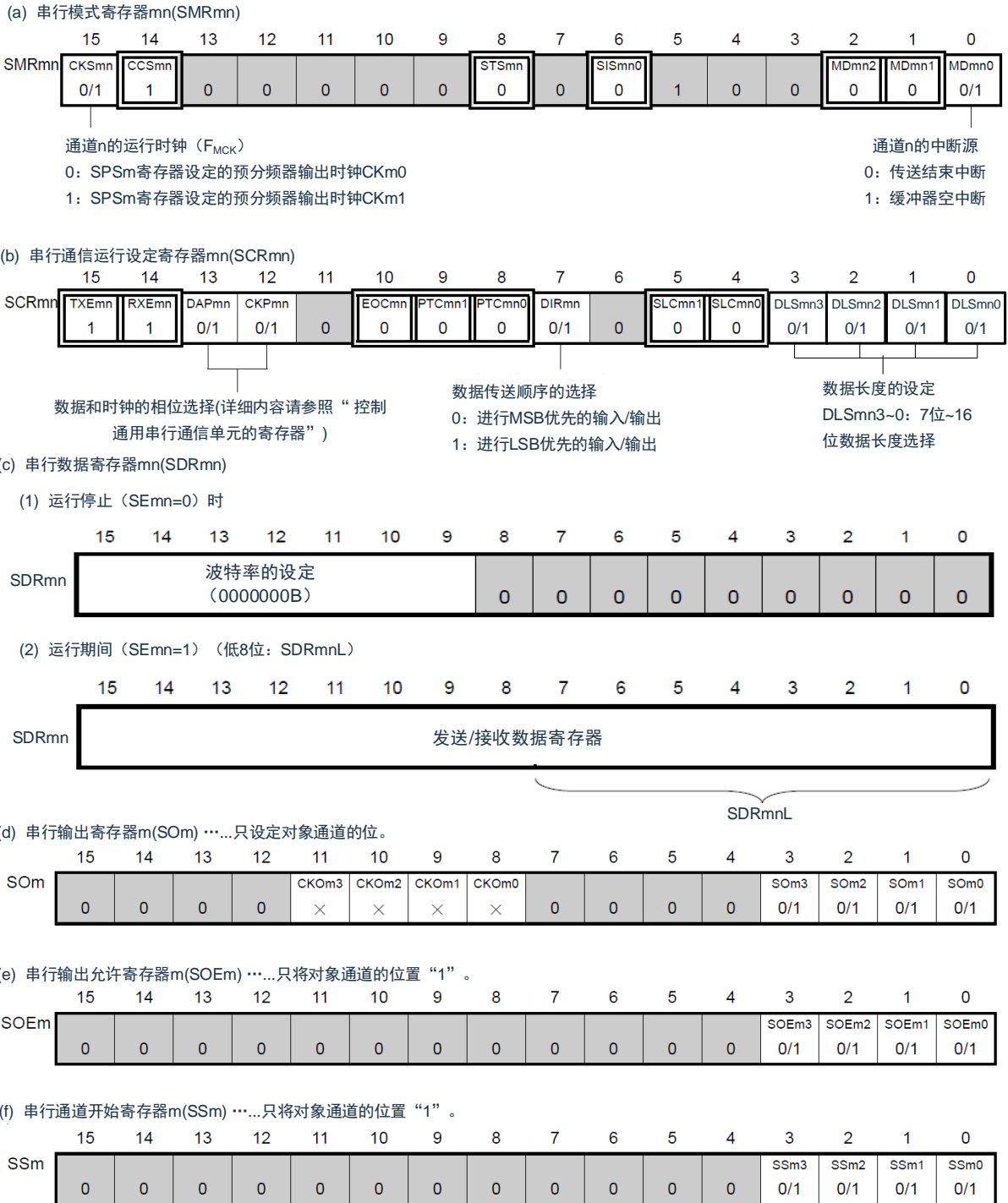
注 2：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用；

备注：

1.  $F_{MCK}$ ：对象通道的运行时钟频率；
2. m：单元号（m=0、1）；  
n：通道号（n=0、1）。

(1) 寄存器的设定

图 12-41: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11) 从属发送和接收时的寄存器设定内容例子



注意: 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;

备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

□: 在 SSPI 主控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”。

(2) 操作步骤

图 12-42: 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 12-43: 从属发送和接收的中止步骤

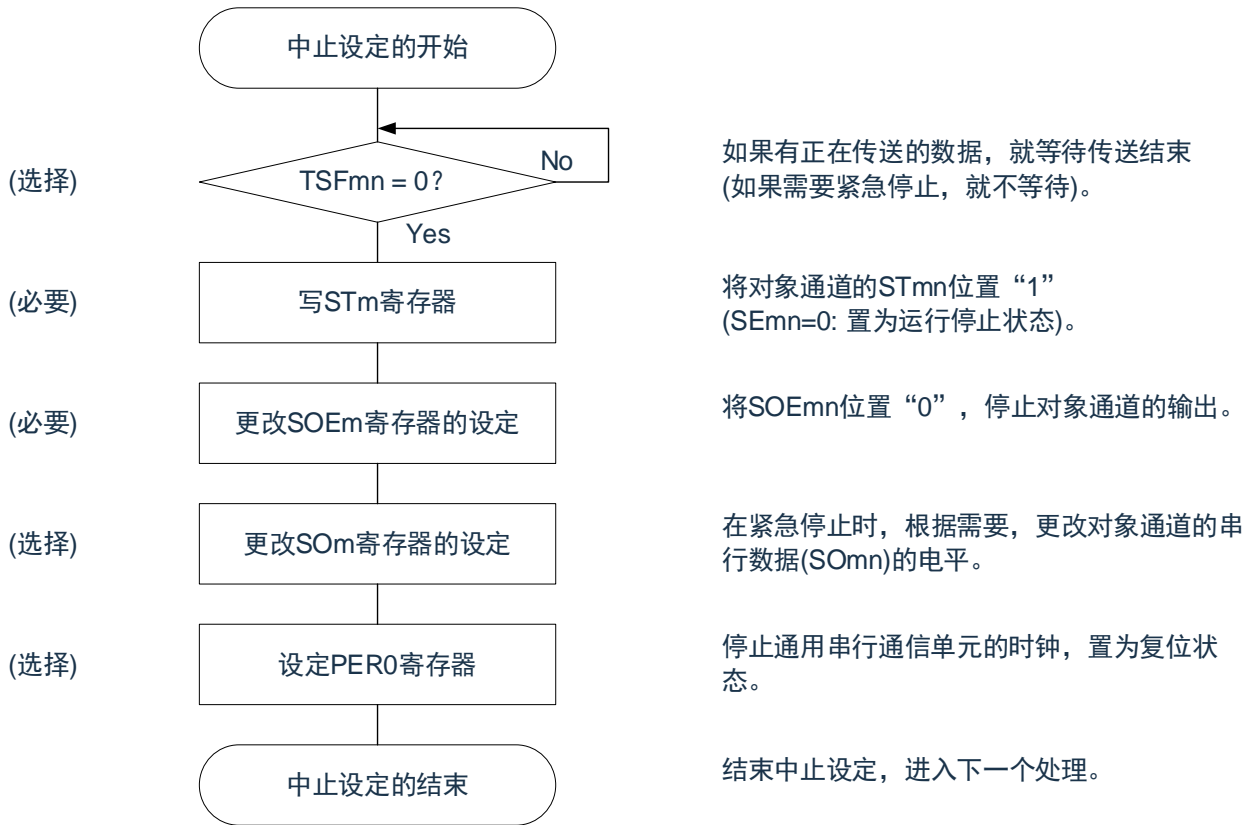


图 12-44：重新开始从属发送和接收的设定步骤

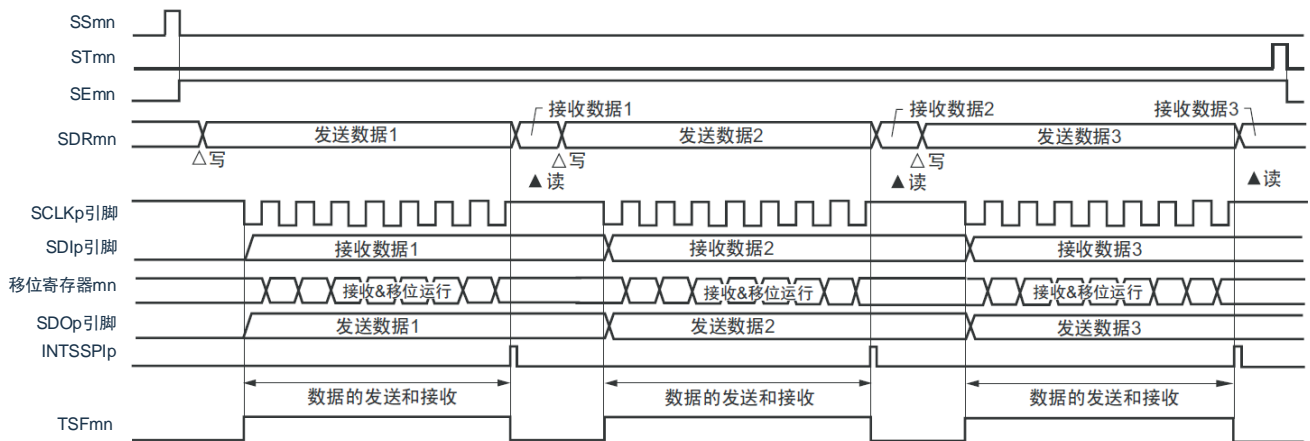


注意：

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的设定。

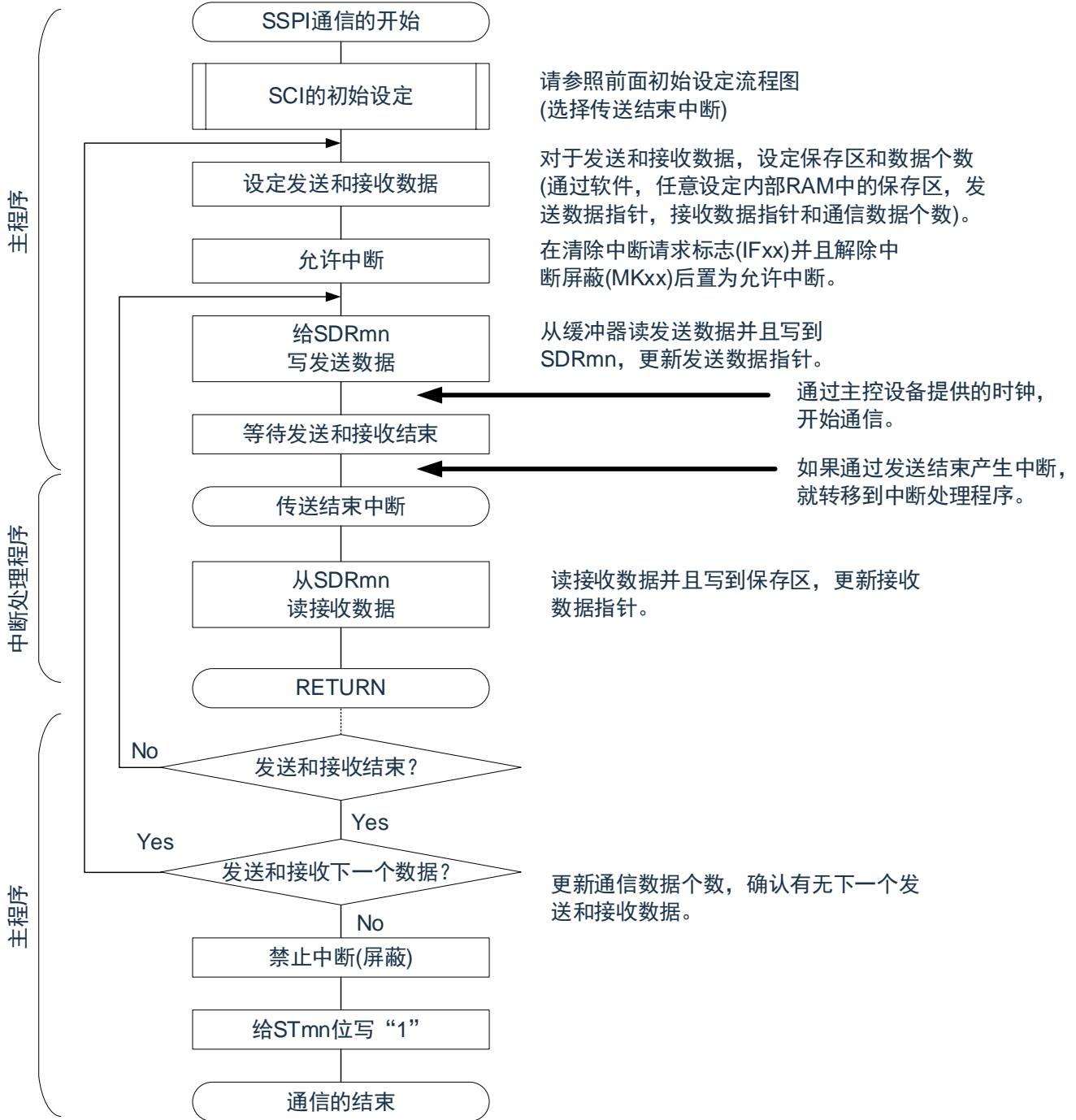
(3) 处理流程（单次发送和接收模式）

图 12-45：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、1）；  
 p：SSPI号（p=00、01、10、11）。

图 12-46: 从属发送和接收（单次发送和接收模式）的流程图

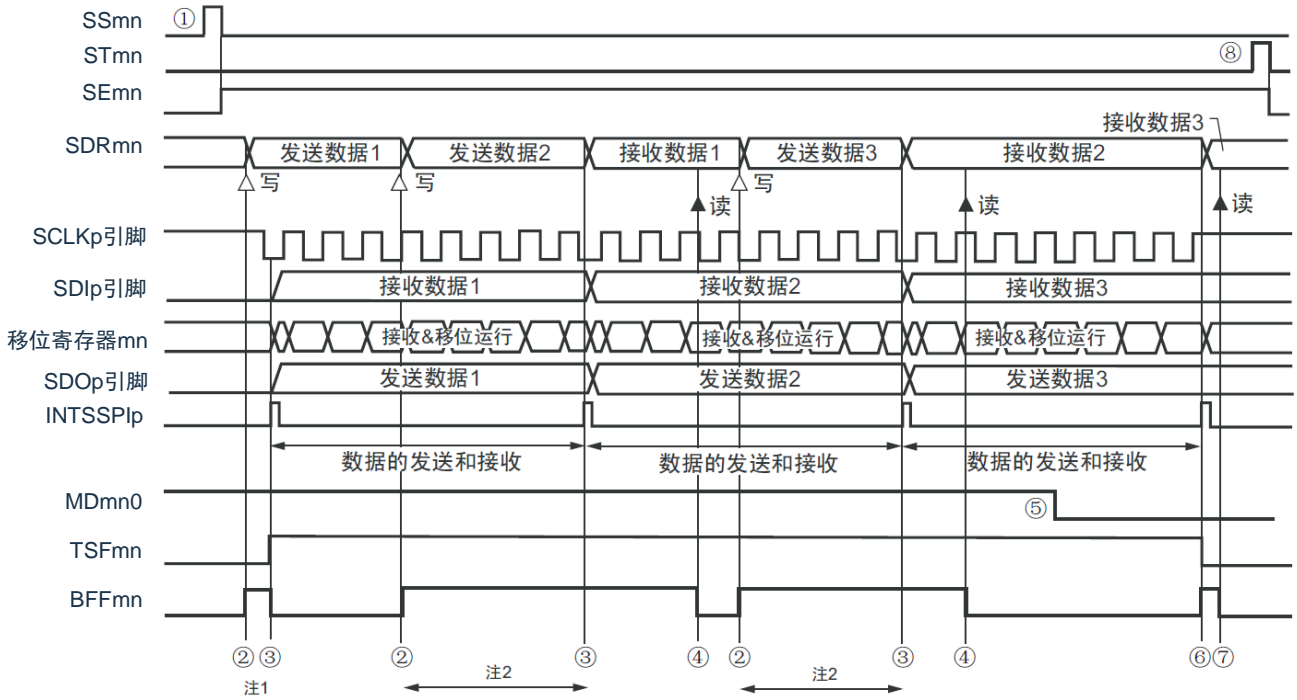


注意: 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。



(4) 处理流程（连续发送和接收模式）

图 12-47：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

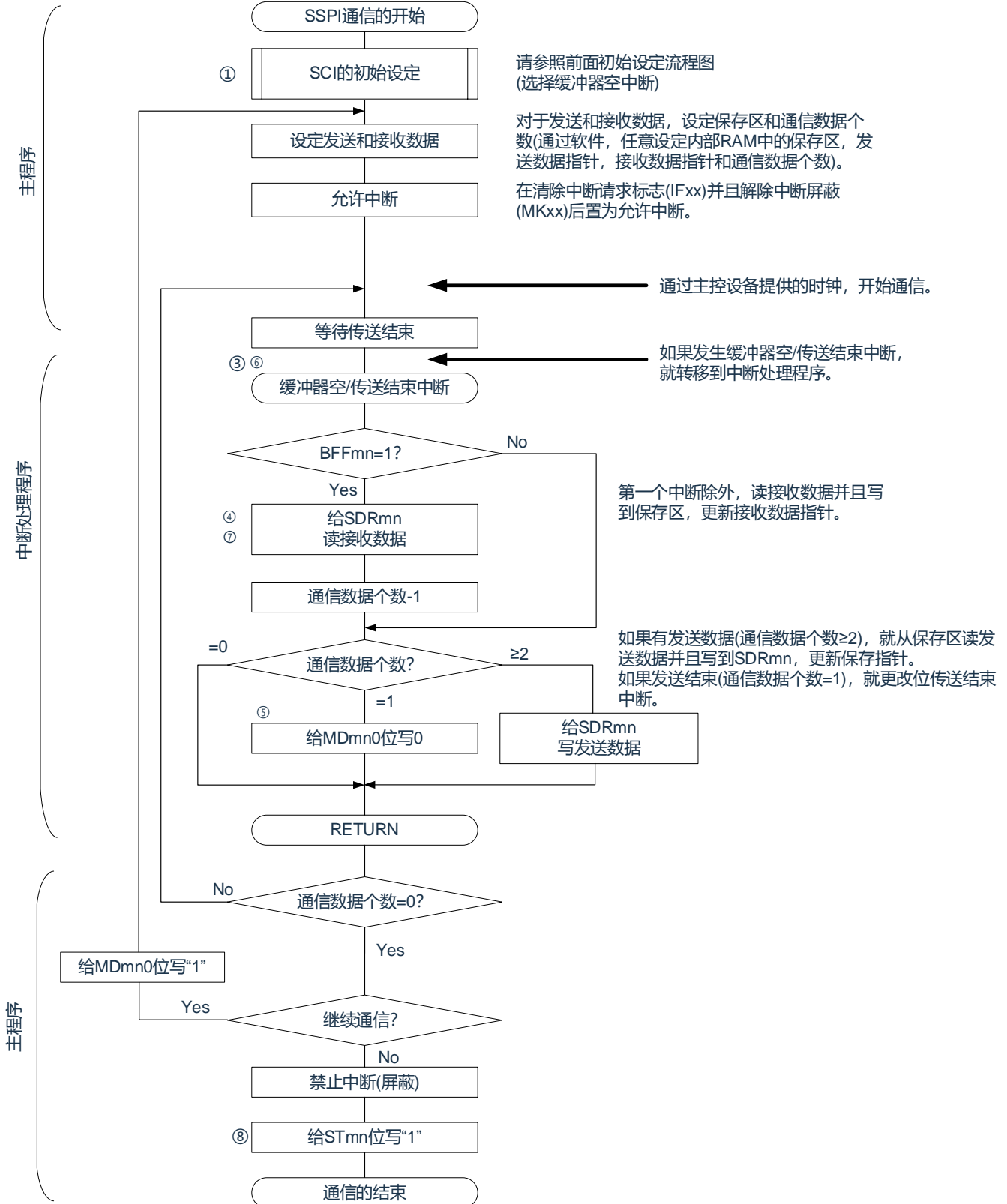


- 注 1：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据；
- 注 2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行；
- 注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写；

备注：

1. 图中的①~⑧对应“图 12-48 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、1）；
- p：SSPI 号（p=00、01、10、11）。

图 12-48: 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 12-47 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

## 12.5.7 传送时钟频率的计算

3线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11) 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

$$\text{(传送时钟频率)} = \{\text{对象通道的运行时钟 (F}_{MCK}\text{) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) 从属设备

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

备注：

1. 容许的最大传送时钟频率为  $F_{MCK}/6$ ；
2. 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；
3. 运行时钟 ( $F_{MCK}$ ) 取决于串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的 bit15 (CKSmn)。

表 12-28: 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 <sub>MCK</sub> ) <sup>注</sup>	
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	F <sub>CLK</sub>	32MHz
	X	X	X	X	0	0	0	1	F <sub>CLK</sub> /2	16MHz
	X	X	X	X	0	0	1	0	F <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	X	X	X	X	0	0	1	1	F <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	X	X	X	X	0	1	0	0	F <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	X	X	X	X	0	1	0	1	F <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	X	X	X	X	0	1	1	0	F <sub>CLK</sub> /2 <sup>6</sup>	500KHz
	X	X	X	X	0	1	1	1	F <sub>CLK</sub> /2 <sup>7</sup>	250KHz
	X	X	X	X	1	0	0	0	F <sub>CLK</sub> /2 <sup>8</sup>	125KHz
	X	X	X	X	1	0	0	1	F <sub>CLK</sub> /2 <sup>9</sup>	62.5KHz
	X	X	X	X	1	0	1	0	F <sub>CLK</sub> /2 <sup>10</sup>	31.25KHz
	X	X	X	X	1	0	1	1	F <sub>CLK</sub> /2 <sup>11</sup>	15.63KHz
	X	X	X	X	1	1	0	0	F <sub>CLK</sub> /2 <sup>12</sup>	7.81KHz
	X	X	X	X	1	1	0	1	F <sub>CLK</sub> /2 <sup>13</sup>	3.91KHz
	X	X	X	X	1	1	1	0	F <sub>CLK</sub> /2 <sup>14</sup>	1.95KHz
X	X	X	X	1	1	1	1	F <sub>CLK</sub> /2 <sup>15</sup>	977Hz	
1	0	0	0	0	X	X	X	X	F <sub>CLK</sub>	32MHz
	0	0	0	1	X	X	X	X	F <sub>CLK</sub> /2	16MHz
	0	0	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	0	0	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	0	1	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	0	1	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	0	1	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>6</sup>	500KHz
	0	1	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>7</sup>	250KHz
	1	0	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>8</sup>	125KHz
	1	0	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>9</sup>	62.5KHz
	1	0	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>10</sup>	31.25KHz
	1	0	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>11</sup>	15.63KHz
	1	1	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>12</sup>	7.81KHz
	1	1	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>13</sup>	3.91KHz
	1	1	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>14</sup>	1.95KHz
1	1	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>15</sup>	977Hz	

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;

X: 忽略;

m: 单元号 (m=0、1);

n: 通道号 (n=0、1)。

## 12.5.8 在3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11）通信过程中发生错误时的处理步骤如表 12-29 所示：

表 12-29：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。→	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn（SDIRmn）写“1”。→	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0、1）；  
n：通道号（n=0、1）。

## 12.6 从属选择输入功能的时钟同步串行通信的运行

SCI0 的通道 0 是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- (1) 7 位~16 位的数据长度
- (2) 发送和接收数据的相位控制
- (3) MSB/LSB 优先的选择
- (4) 发送和接收数据的电平设定

[时钟控制]

- (1) 输入/输出时钟的相位控制
- (2) 设定由预分频器和通道内部计数器产生的传送周期。
- (3) 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

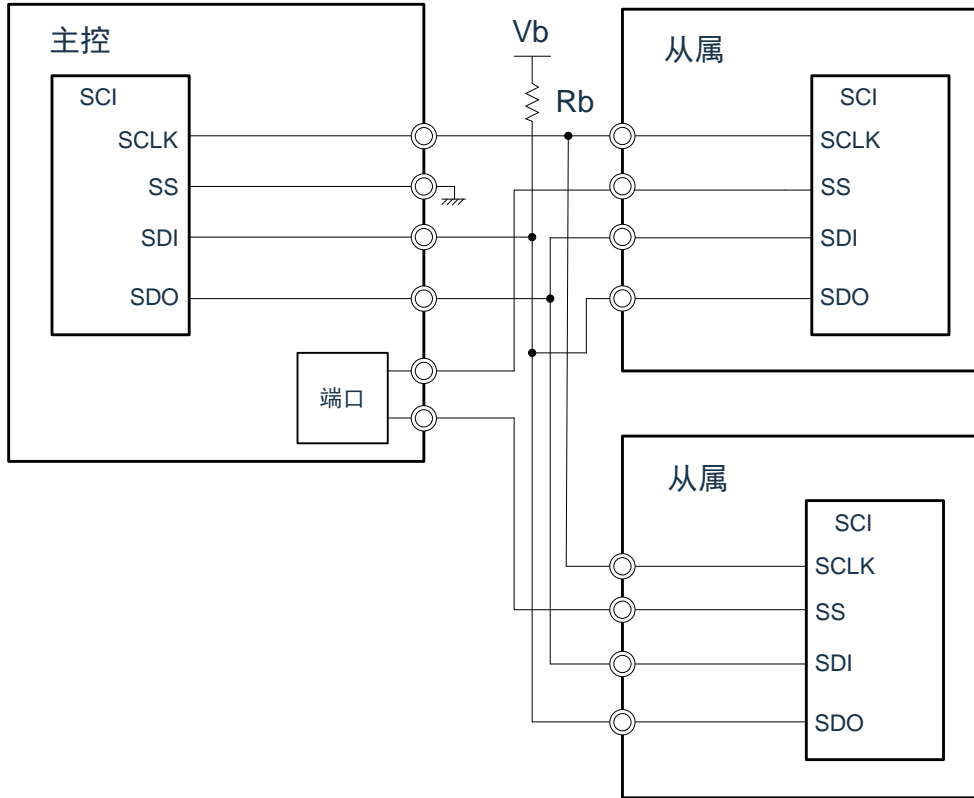
从属选择输入功能有以下 3 种通信运行:

- (1) 从属发送 (参照 12.6.1)
- (2) 从属接收 (参照 12.6.2)
- (3) 从属的发送和接收 (参照 12.6.3)

注意: 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。

能通过使用从属选择输入功能，使 1 个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1 个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SDO 引脚的输出。当被选择为通信对象的从属设备时，SDO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO 引脚变为高电平输出，因此在连接多个从属设备的环境下需要将 SDO 引脚设定为 Nch-O.D 并且将该节点上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

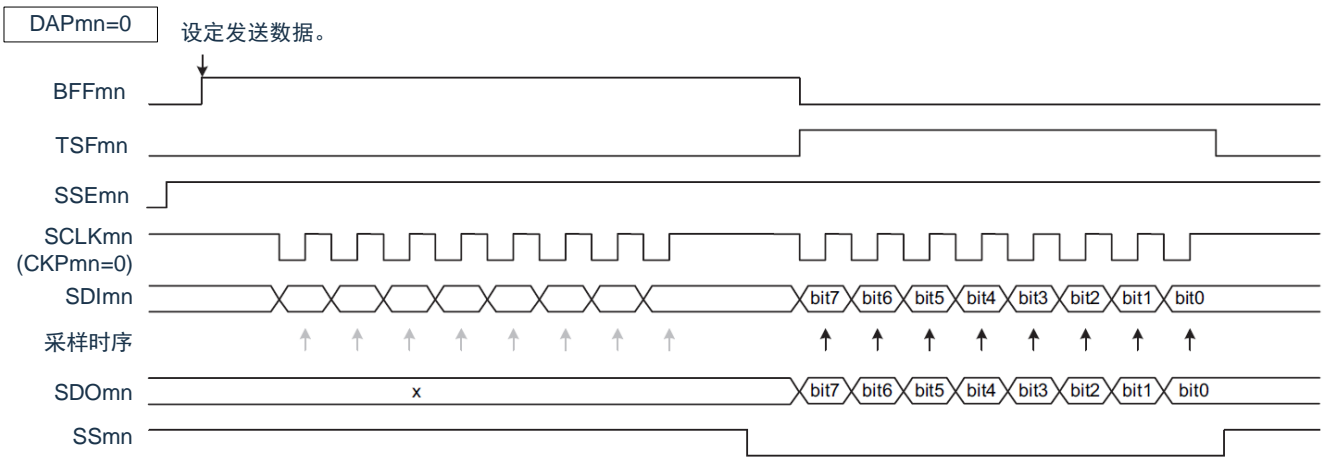
图 12-49：从属选择输入功能的结构例子



注意：

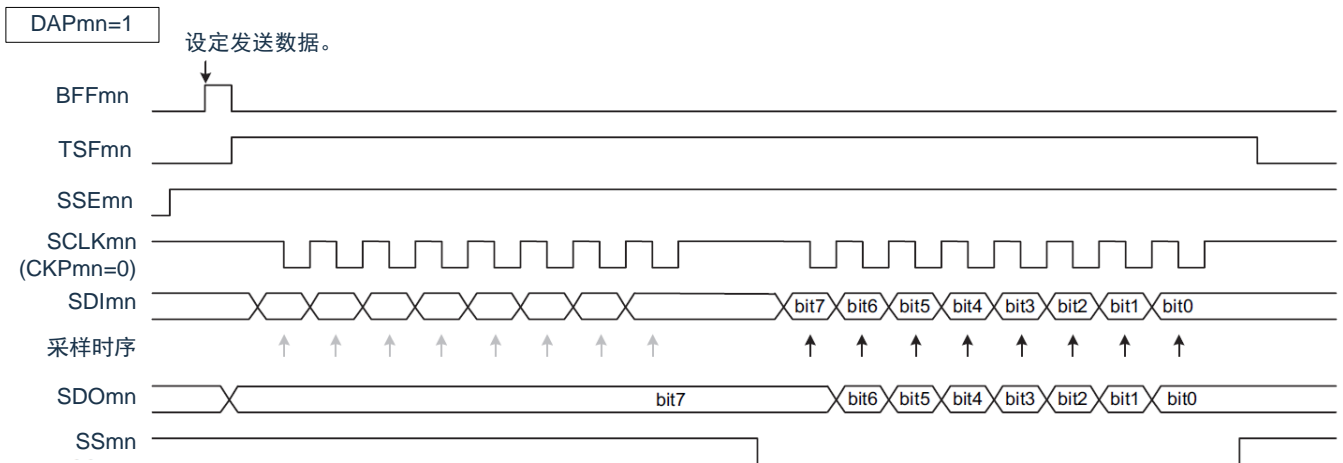
1. 将 SDO00 引脚选择为 N 沟道漏极开路输出模式；
2. 必须通过端口的操作输出从属选择信号。

图 12-50: 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) ；  
n：通道号 (n=0) 。



## 12.6.1 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。

表 12-30：从属发送

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKO100、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 <sub>MCK</sub> /6[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
从属选择输入功能	可选择从属选择功能的运行。

注 1：因为在内部对 SCLK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F<sub>MCK</sub>/6[Hz]。

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

F<sub>MCK</sub>：对象通道的运行时钟频率

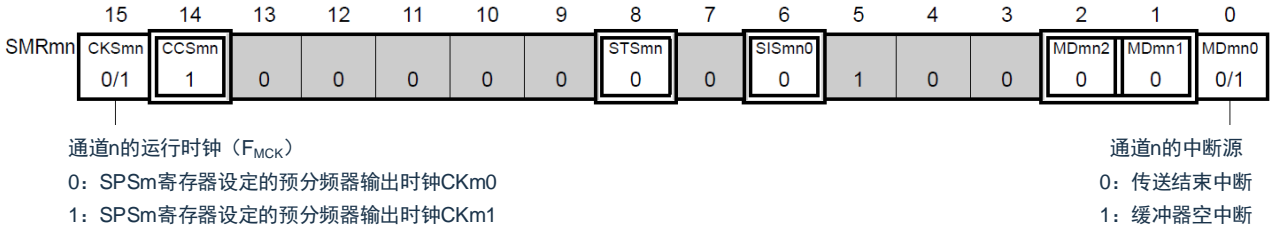
m：单元号（m=0）；

n：通道号（n=0）。

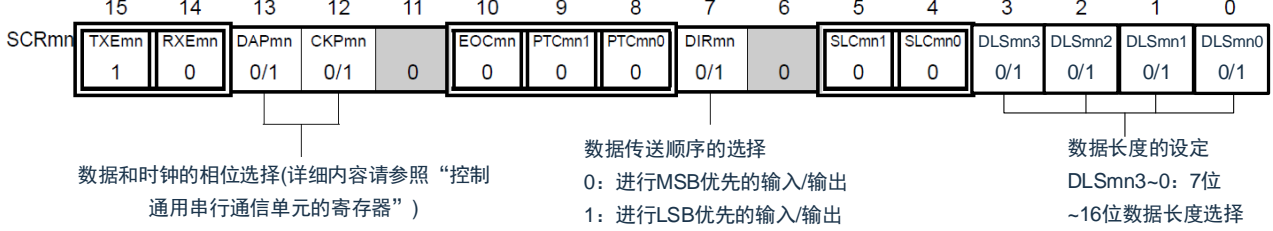
(1) 寄存器的设定

图 12-51：从属选择输入功能（SSPI00）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)

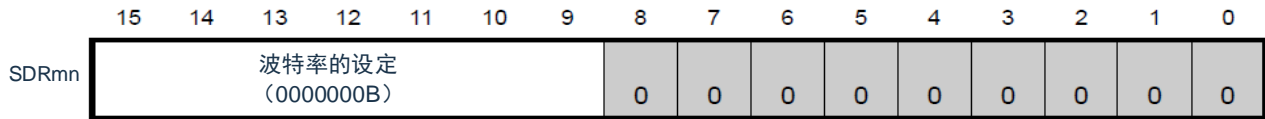


(b) 串行通信运行设定寄存器mn(SCRmn)

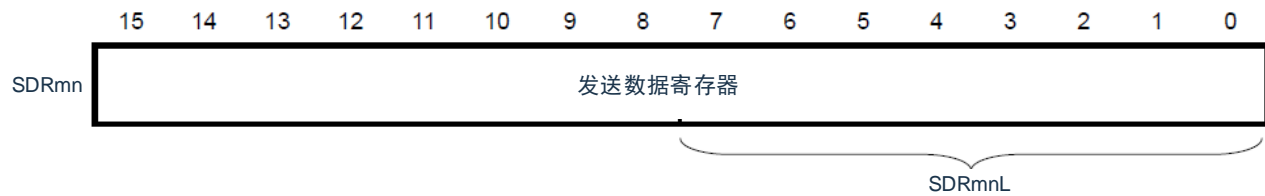


(c) 串行数据寄存器mn(SDRmn)

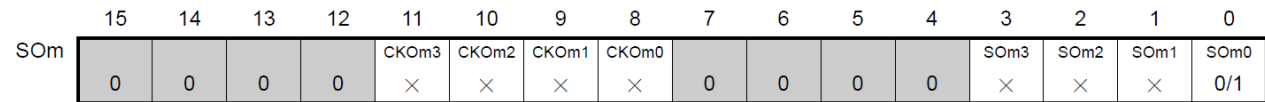
(1) 运行停止 (SEmn=0) 时



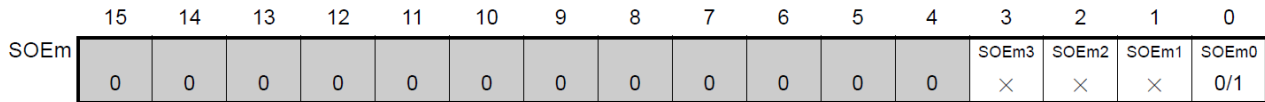
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(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: m: 单元号 (m=0) ;

n: 通道号 (n=0) ;

p: SSPI 号 (p=00) ;

□: 在 SSPI 主控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 12-51：从属选择输入功能（SSPI00）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 ×	SSm0 0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效  
1: SS00引脚的输入值有效

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI号（p=00）；

☐：在 SSPI 主控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”。

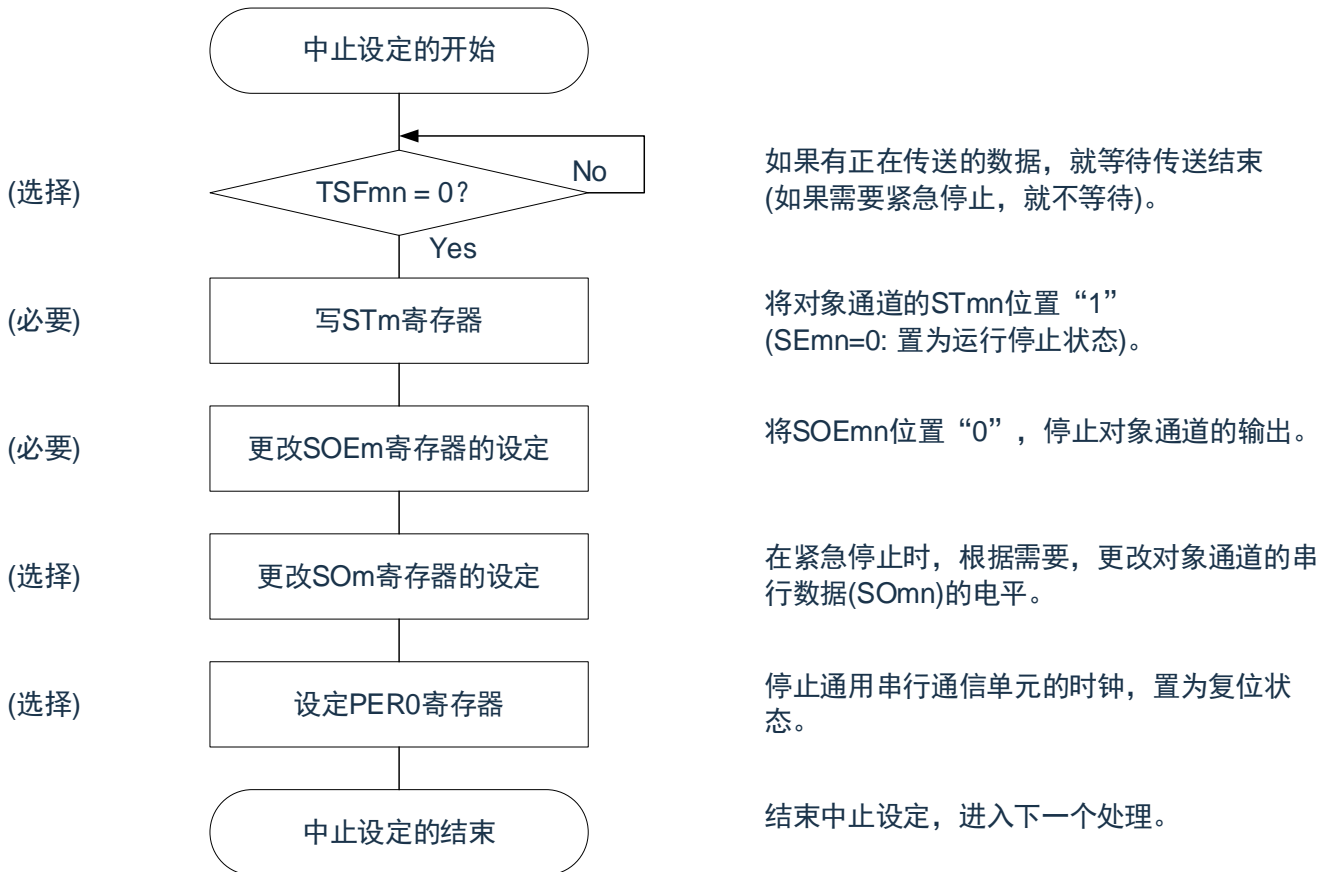
(2) 操作步骤

图 12-52: 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 12-53: 从属发送的中止步骤



注意：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；

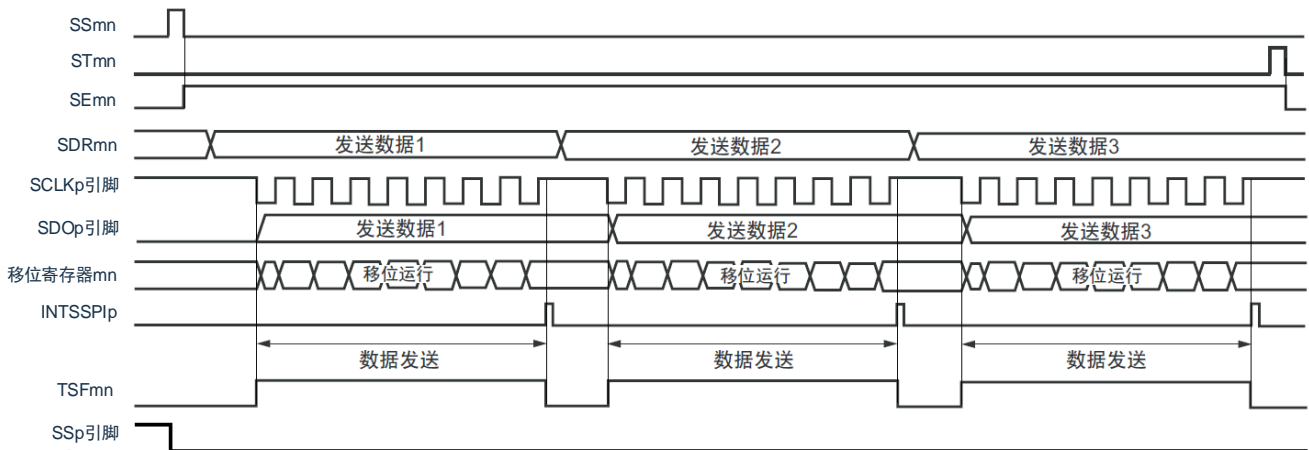
备注：m: 单元号 (m=0)； n: 通道号 (n=0)； p: SSPI 号 (p=00)；

图 12-54: 重新开始从属发送的设定步骤



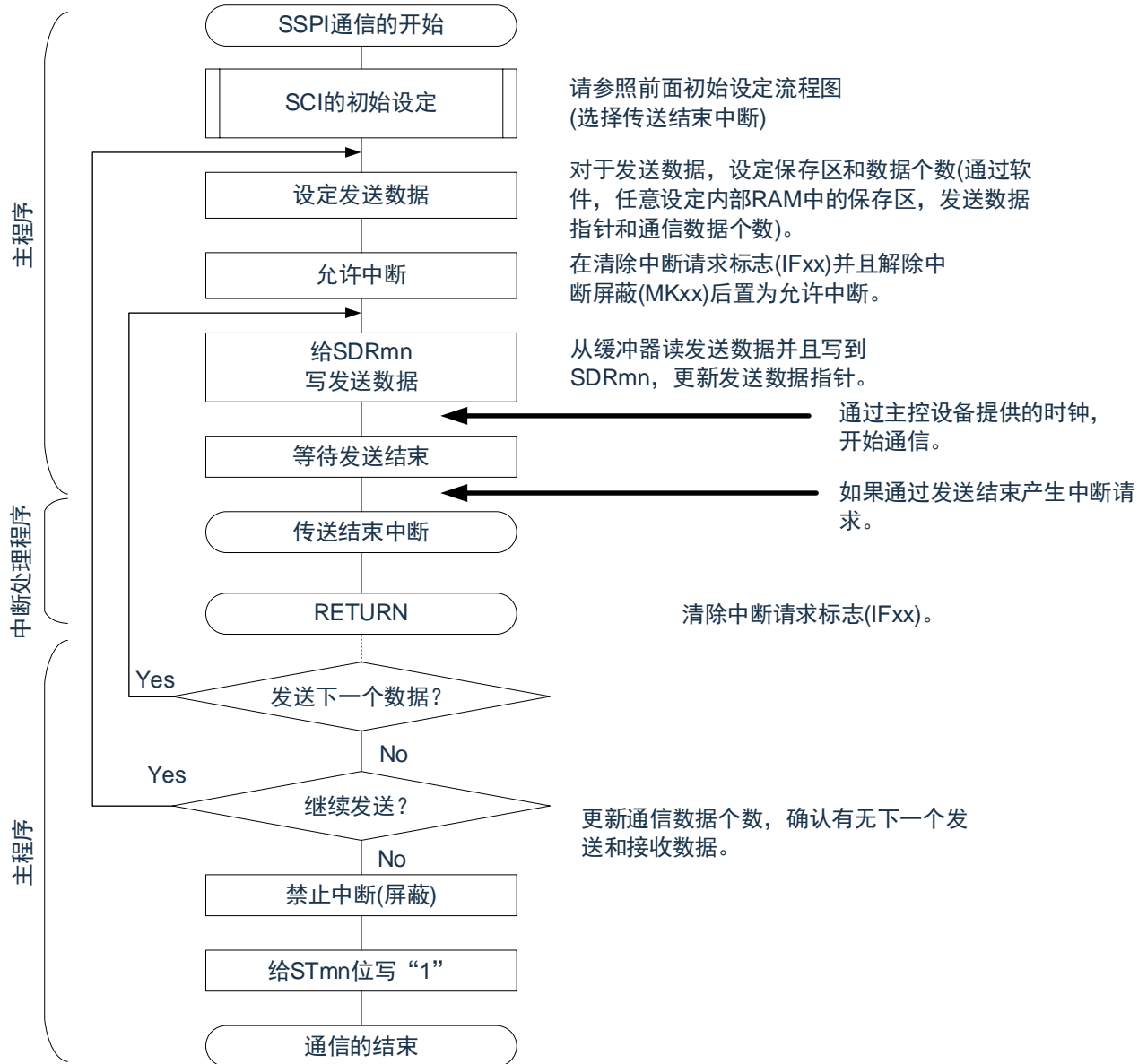
(3) 处理流程（单次发送模式）

图 12-55：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

图 12-56: 从属发送（单次发送模式）的流程图

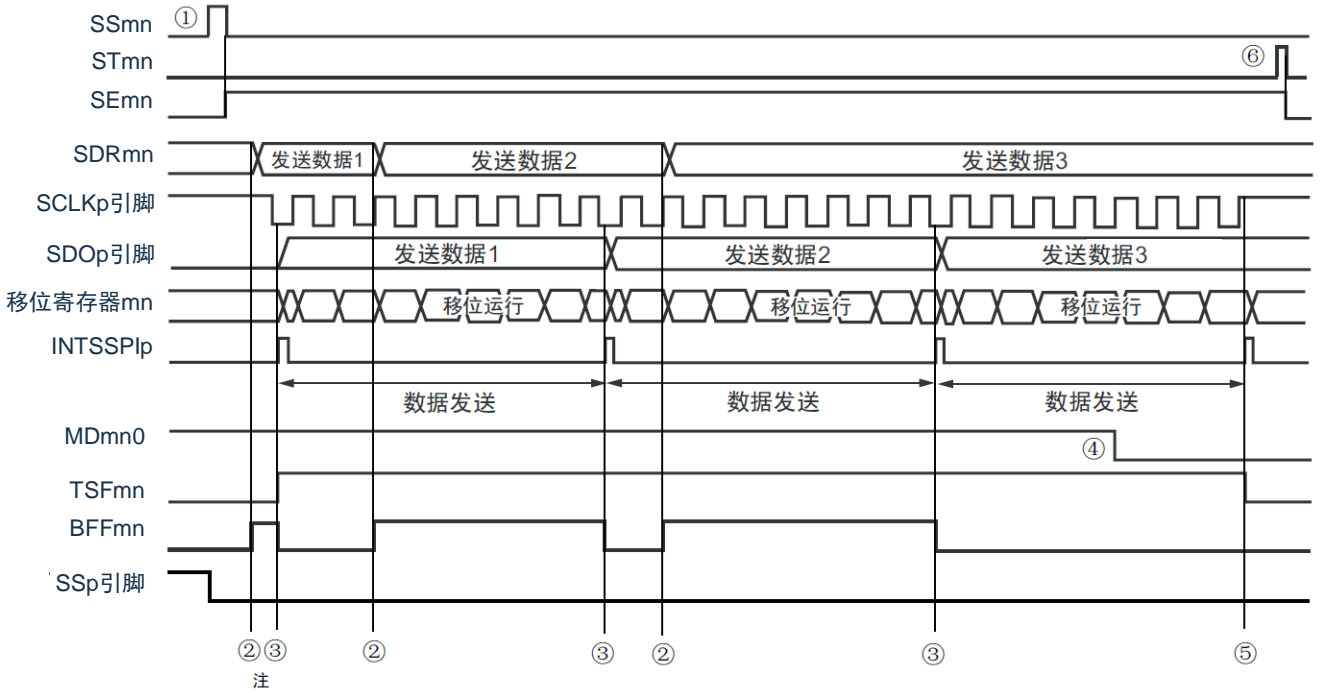


备注: m: 单元号 (m=0) ;  
n: 通道号 (n=0) ;  
p: SSPI 号 (p=00) 。



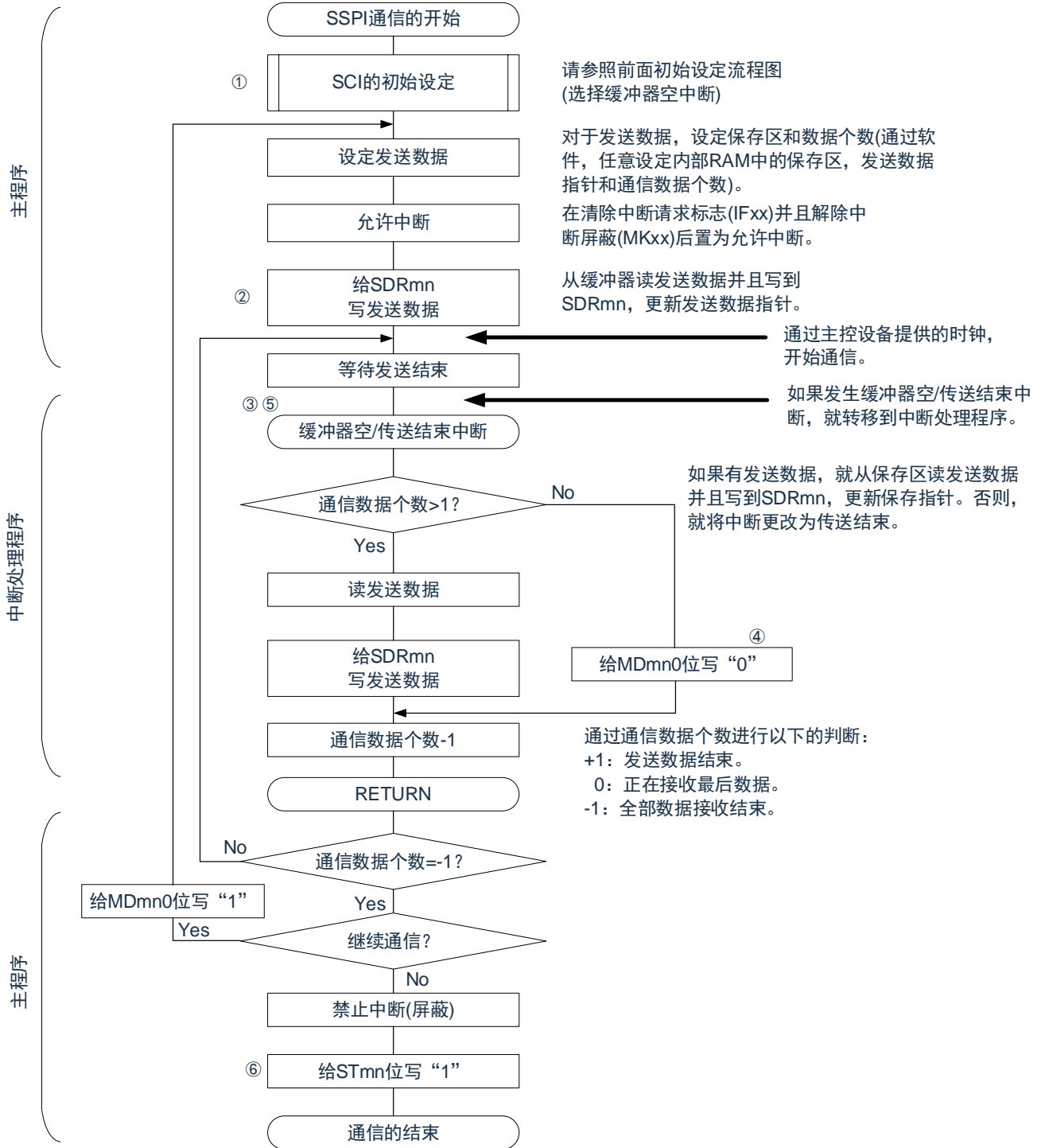
(4) 处理流程（连续发送模式）

图 12-57：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 12-58: 从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 12-57 从属发送（连续发送模式）的时序图”中的①~⑥。

- m: 单元号 (m=0) ;
- n: 通道号 (n=0) ;
- p: SSPI 号 (p=00) 。

## 12.6.2 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。

表 12-31：从属接收

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、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	Max.F <sub>MCK</sub> /6[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
从属选择输入功能	可选择从属选择输入功能的运行。

注 1：因为在内部对 SCLK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $F_{MCK}/6[Hz]$ 。

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

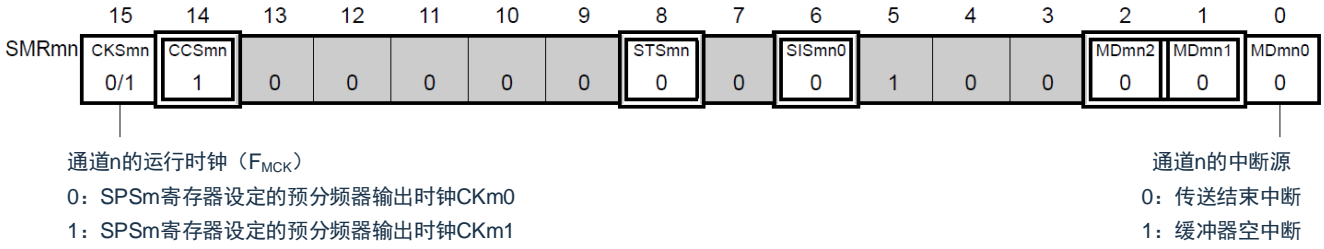
F<sub>MCK</sub>：对象通道的运行时钟频率

m：单元号（m=0） n：通道号（n=0）。

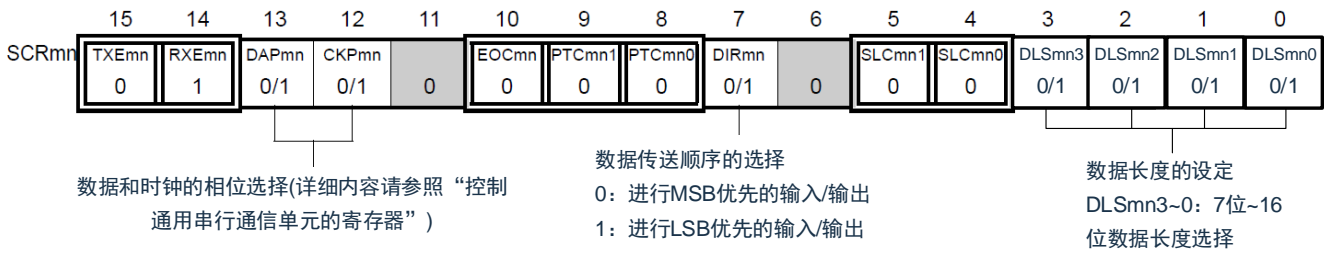
(1) 寄存器的设定

图 12-59: 从属选择输入功能 (SSPI00) 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)

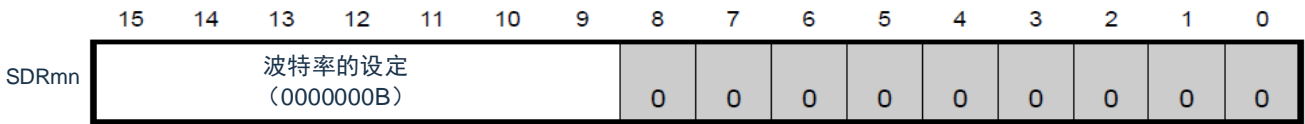


(b) 串行通信运行设定寄存器mn(SCRmn)

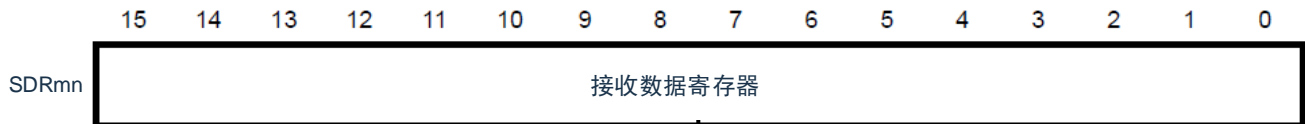


(c) 串行数据寄存器mn(SDRmn)

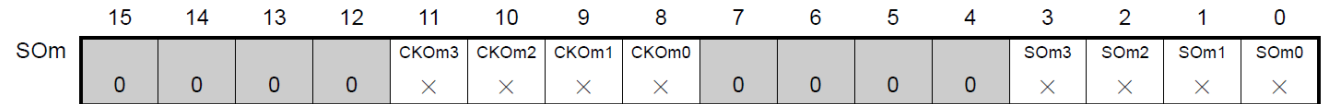
(1) 运行停止 (SEmn=0) 时



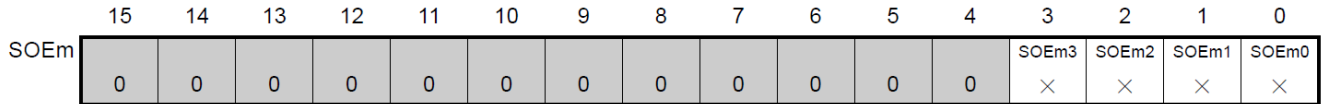
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(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: m: 单元号 (m=0) ;  
 n: 通道号 (n=0) ;  
 p: SSPI 号 (p=00) ;  
 □: 在 SSPI 主控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 12-59: 从属选择输入功能 (SSPI00) 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

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效  
1: SS00引脚的输入值有效

备注: m: 单元号 (m=0) ;

n: 通道号 (n=0) ;

p: SSPI 号 (p=00) ;

□ : 在 SSPI 主控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) ;

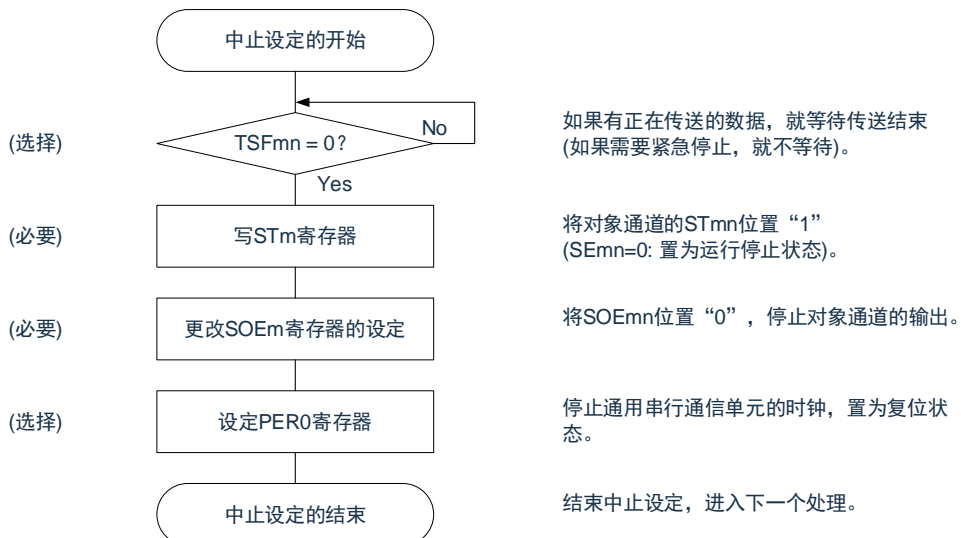
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 12-60: 从属接收的初始设定步骤



图 12-61: 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) 。

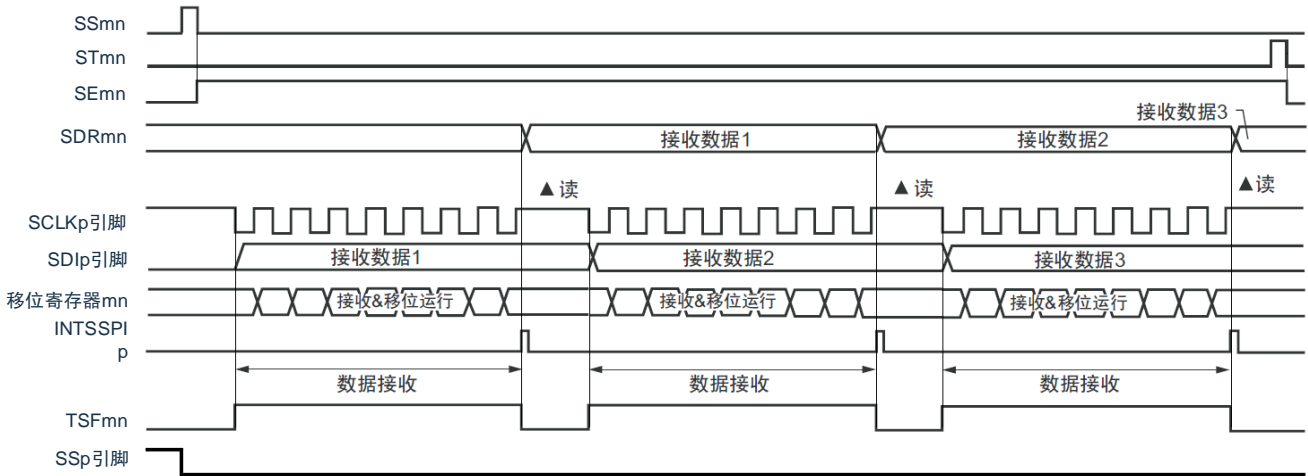
图 12-62: 重新开始从属接收的设定步骤



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）

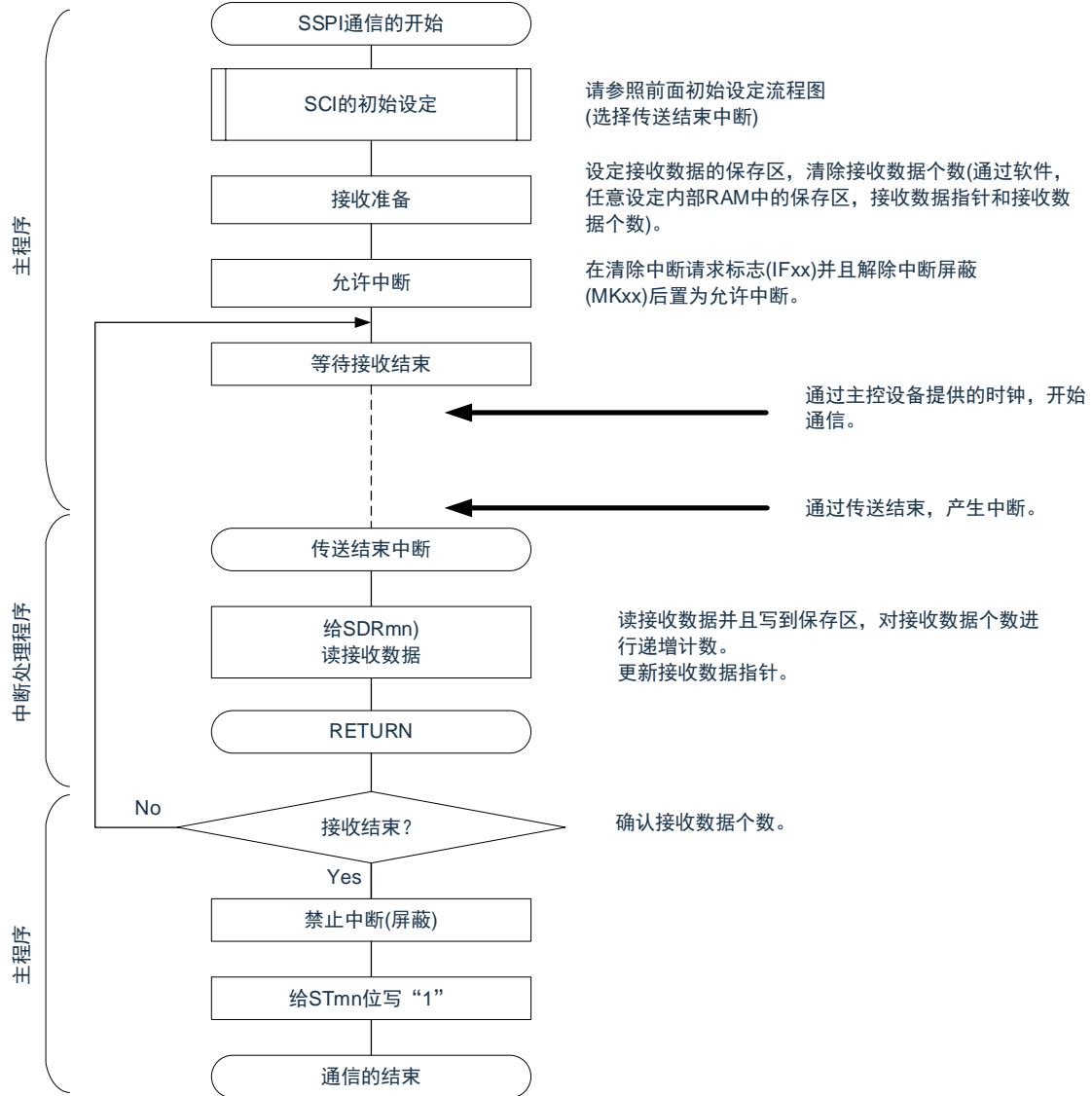
图 12-63：从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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



图 12-64: 从属接收（单次接收模式）的流程图



### 12.6.3 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。

表 12-32：从属的发送和接收

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位~16位
传送速率	Max.F <sub>MCK</sub> /6[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
从属选择输入功能	可选择从属选择输入功能的运行。

注 1：因为在内部对 SCLK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $F_{MCK}/6[Hz]$ 。

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

$F_{MCK}$ ：对象通道的运行时钟频率

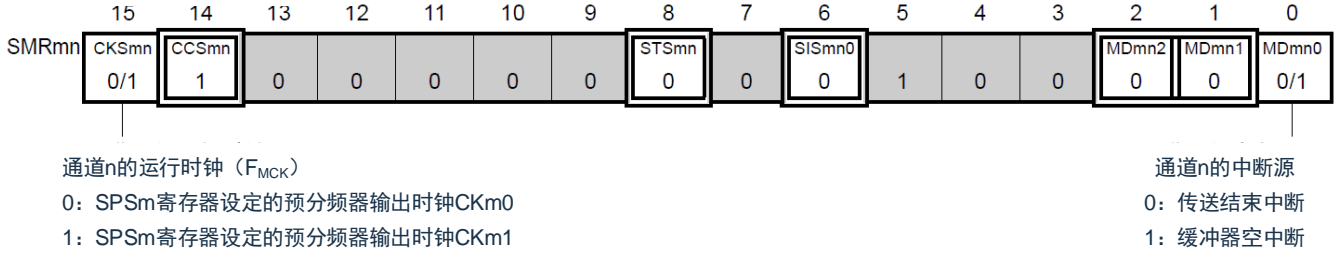
m：单元号（m=0）；

n：通道号（n=0）。

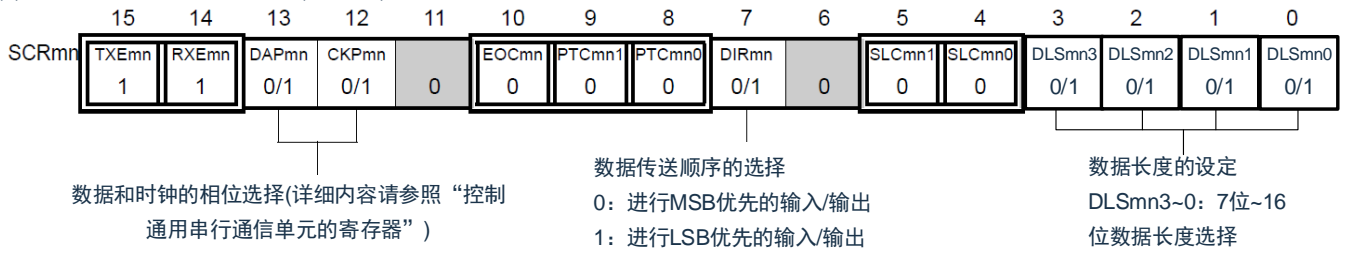
(1) 寄存器的设定

图 12-65: 从属选择输入功能 (SSPI00) 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(SMRmn)

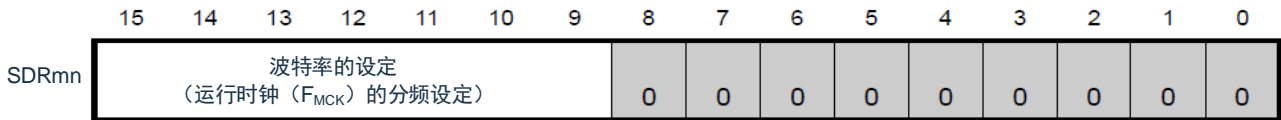


(b) 串行通信运行设定寄存器mn(SCRmn)

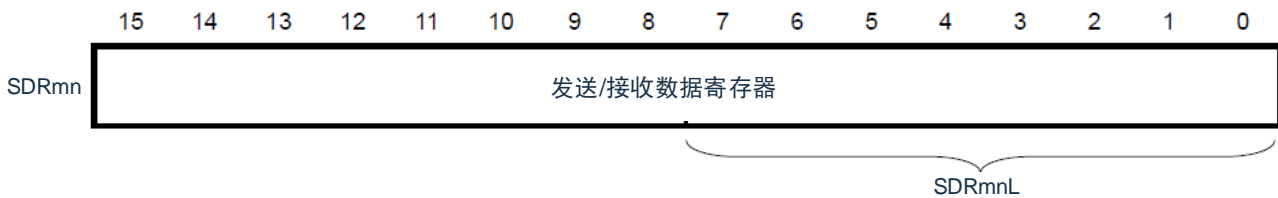


(c) 串行数据寄存器mn(SDRmn)

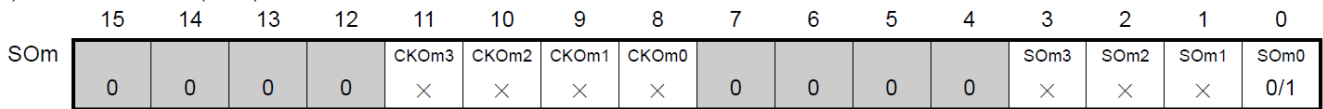
(1) 运行停止 (SEmn=0) 时



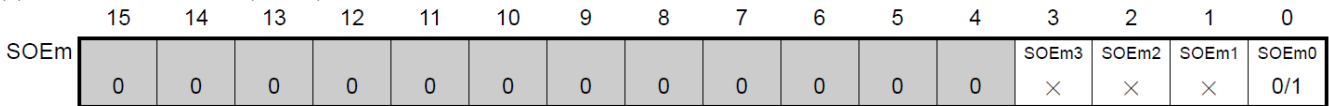
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(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”。



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、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

□: 在 SSPI 主控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 12-65: 从属选择输入功能 (SSPI00) 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	×	0/1

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效  
1: SS00引脚的输入值有效

注意: 在主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: m: 单元号 (m=0、1);

n: 通道号 (n=0、1);

p: SSPI 号 (p=00、01、10、11);

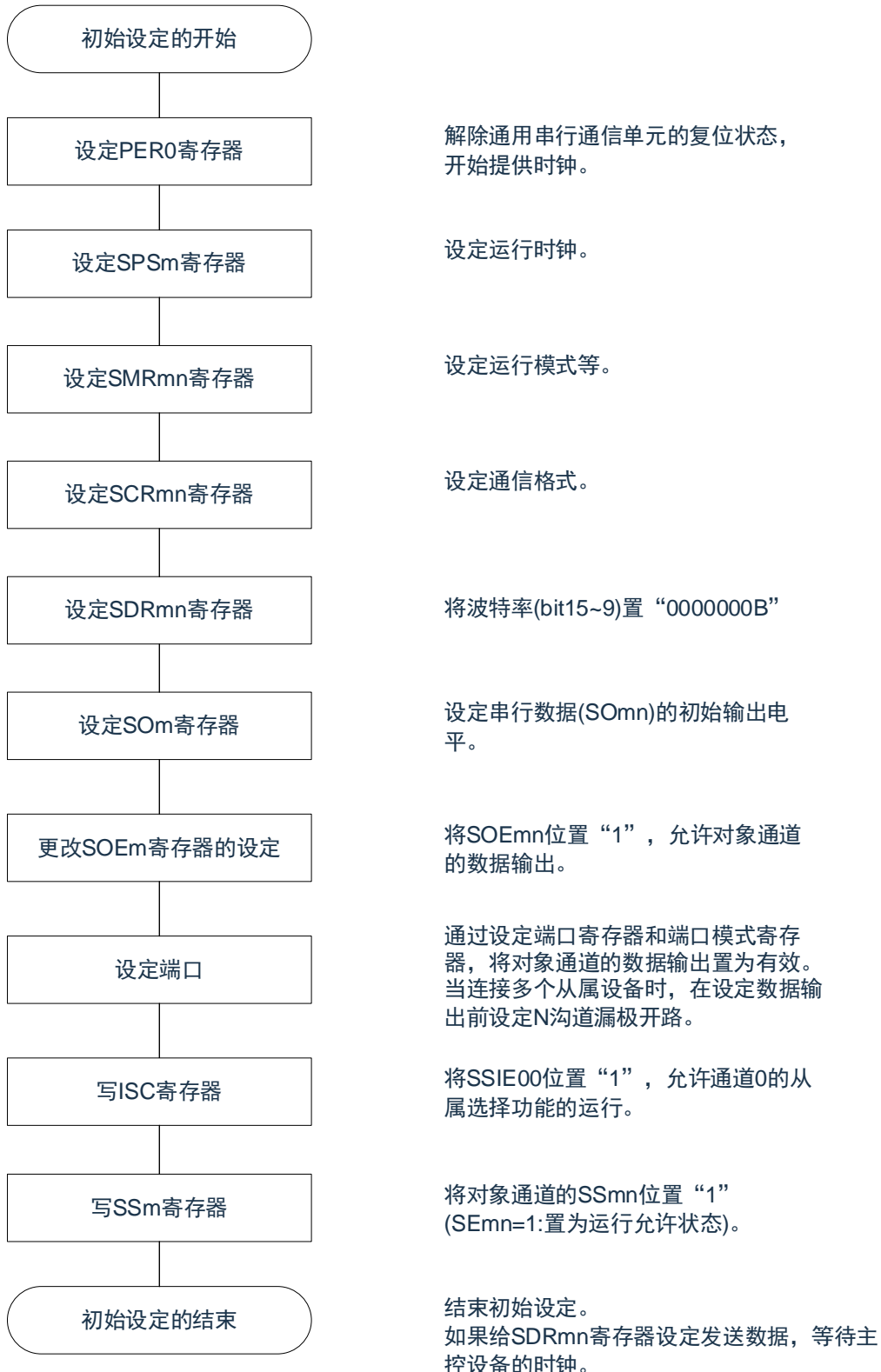
□: 在 SSPI 主导接收模式中为固定设定; ■: 不能设定 (设定初始值);

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”。

(2) 操作步骤

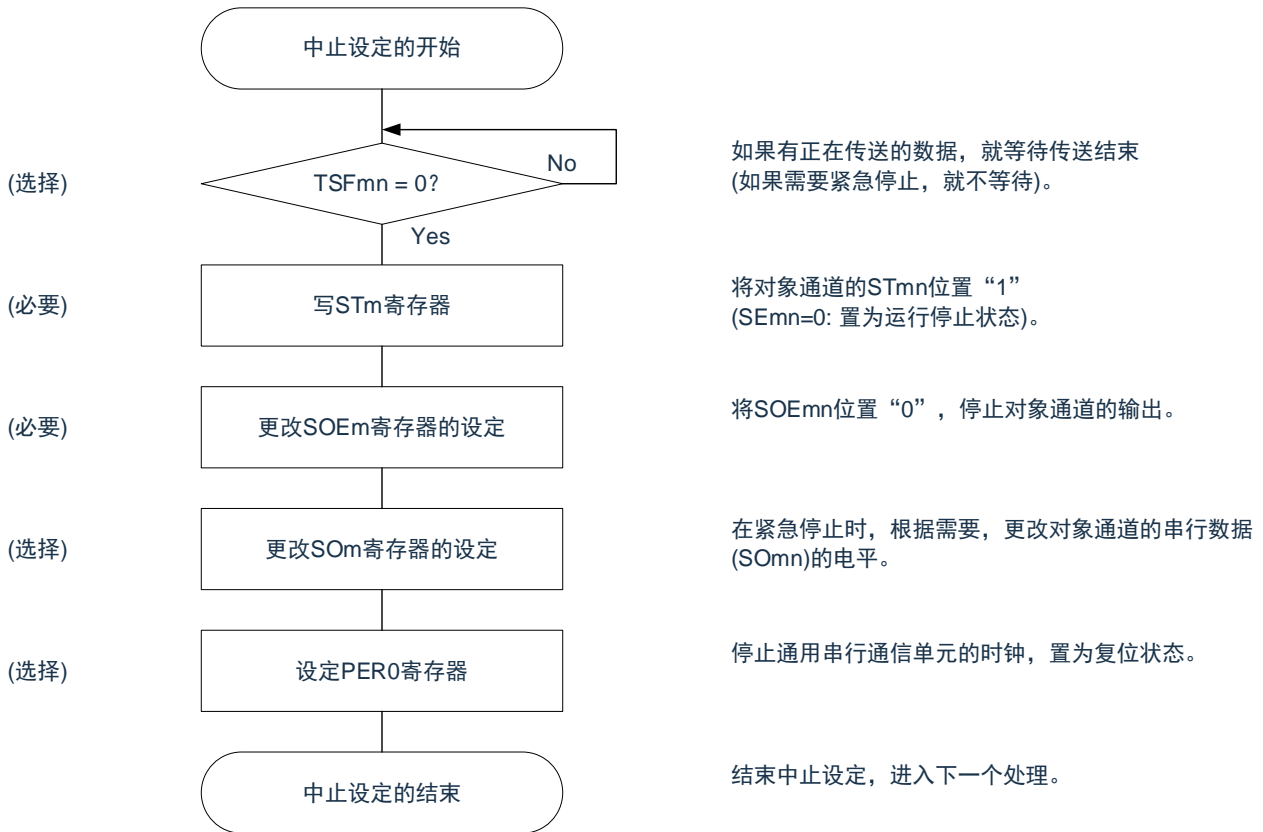
图 12-66: 从属发送和接收的初始设定步骤



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。

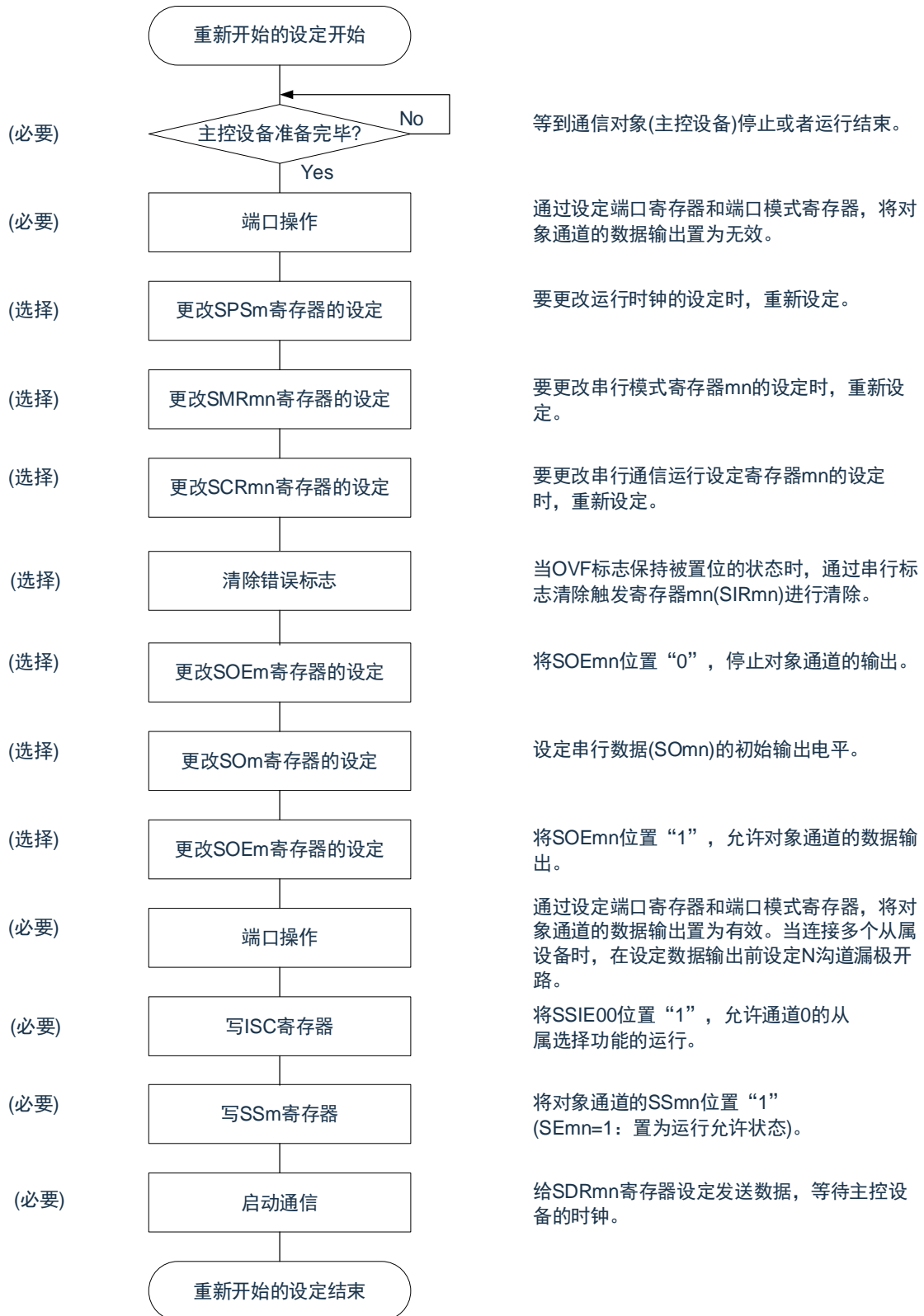
备注: m: 单元号 (m=0);  
 n: 通道号 (n=0);  
 p: SSPI 号 (p=00)。

图 12-67：从属发送和接收的中止步骤



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

图 12-68: 重新开始从属发送和接收的设定步骤

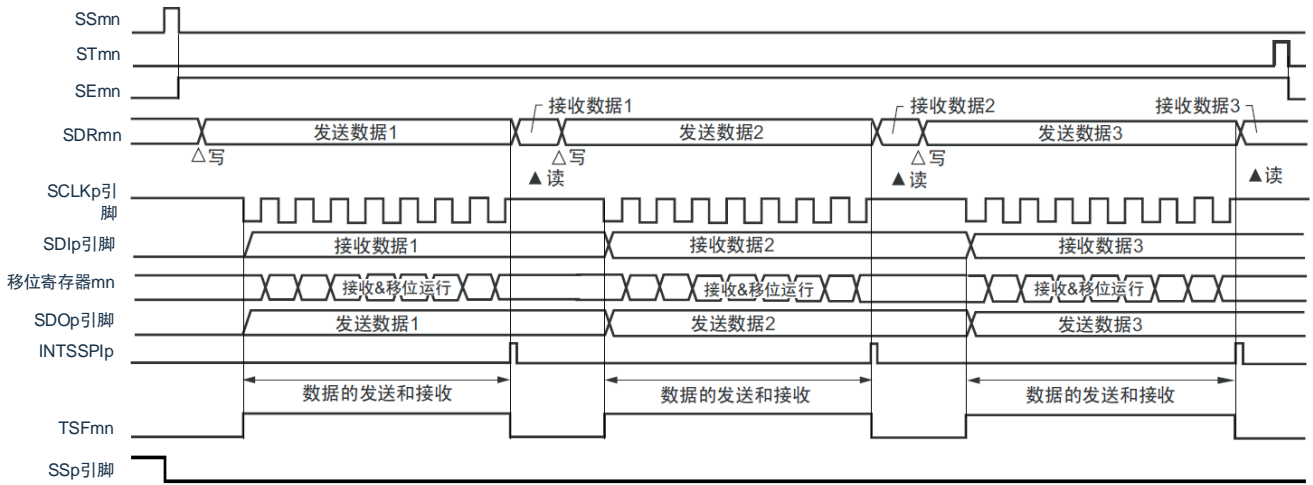


注意:

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）

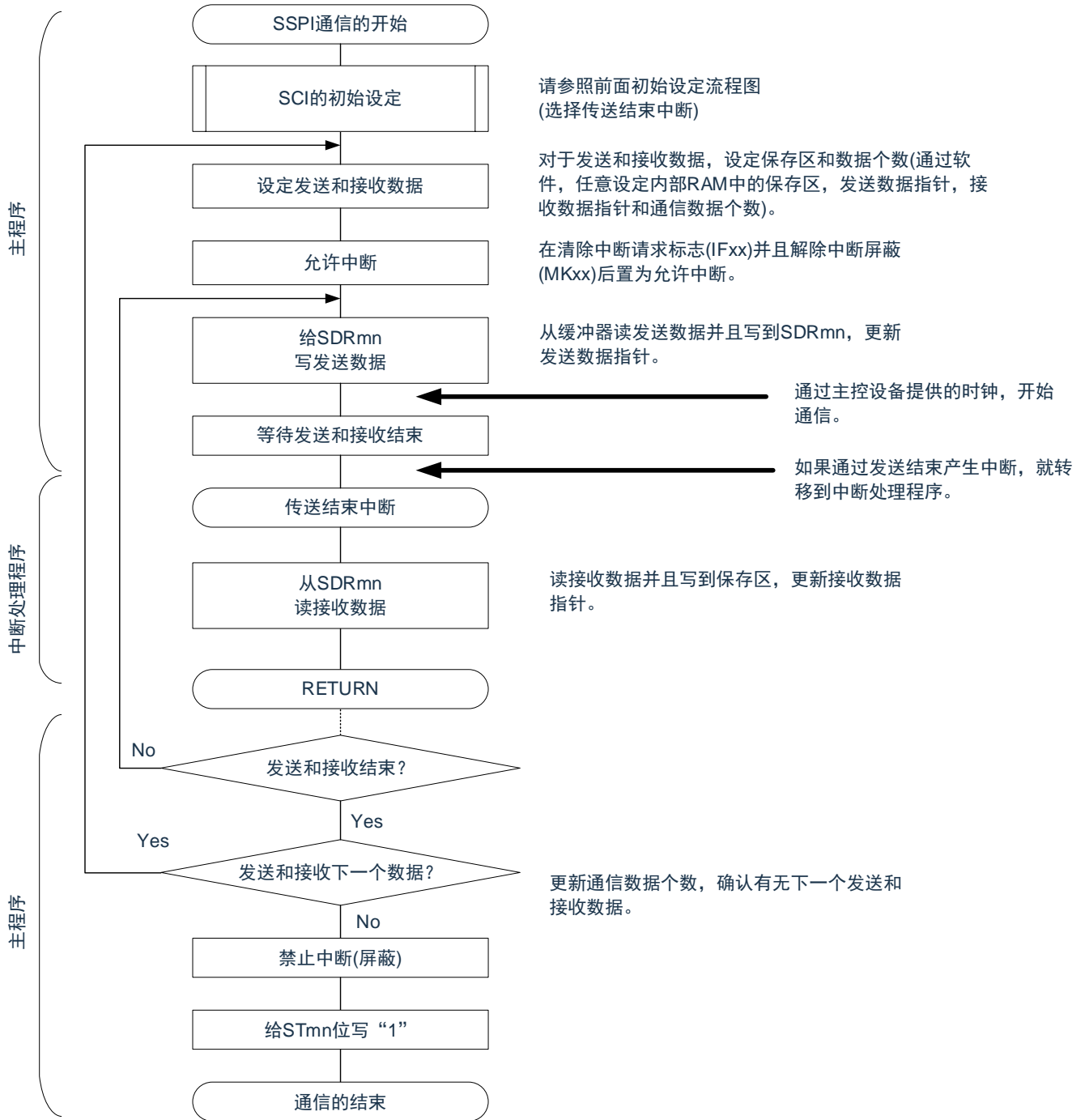
图 12-69：从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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



图 12-70: 从属发送和接收（单次发送和接收模式）的流程图

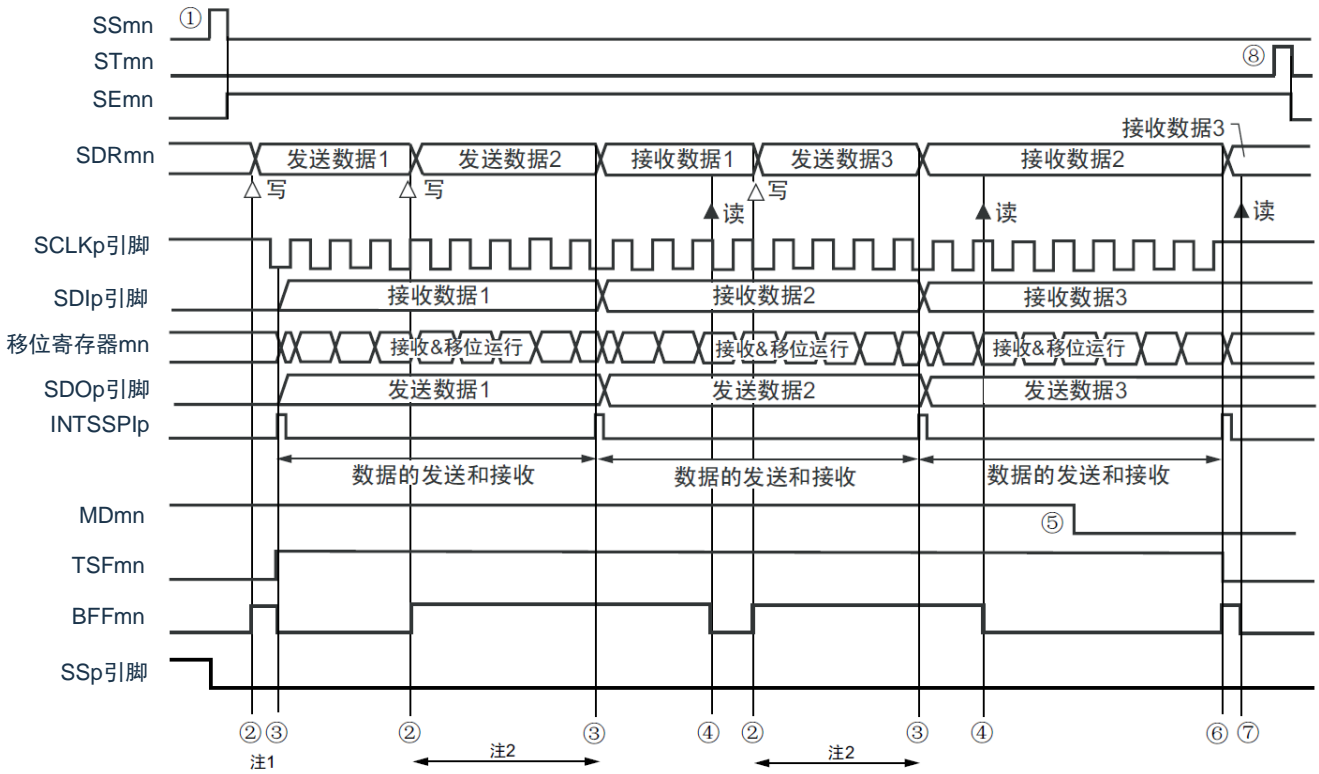


注意: 在主控设备开始输出时钟前, 必须给 SDRmn 寄存器设定发送数据。

- 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 12-71：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

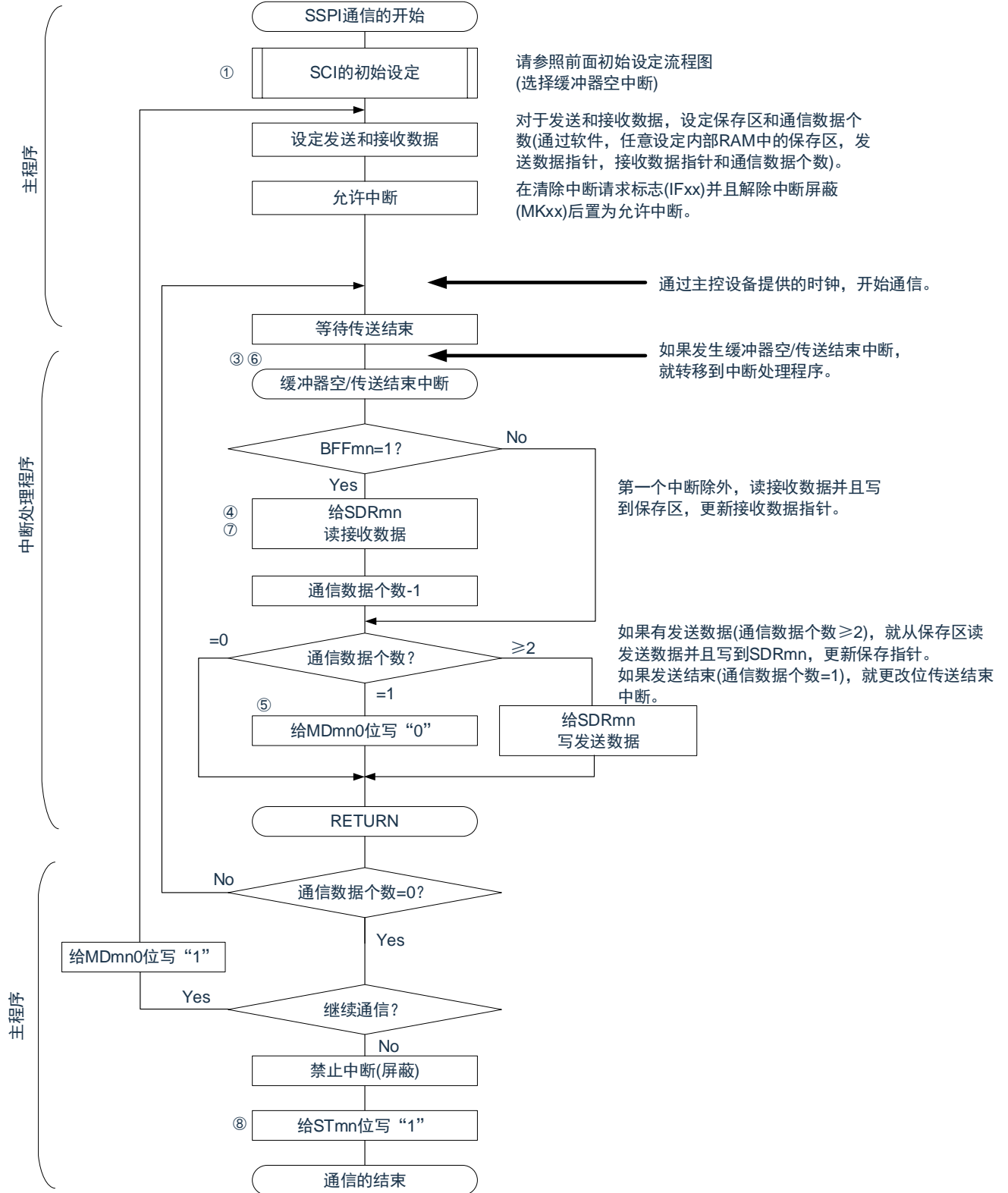
注 2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：

1. 图中的①~⑧对应“图 12-72 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 12-72: 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 12-71 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)。

## 12.6.4 传送时钟频率的计算

从属选择输入功能（SSPI00）通信的传送时钟频率能用以下计算式进行计算。

(1) 从属设备

$$(\text{传送时钟频率}) = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\}^{2^1} [\text{Hz}]$$

注 1：容许的最大传送时钟频率为  $F_{MCK}/6$ 。

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

表 12-33: 从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 ( $F_{MCK}$ ) <sup>注</sup>	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$F_{CLK}=32\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	$F_{CLK}$	32MHz
	X	X	X	X	0	0	0	1	$F_{CLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$F_{CLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$F_{CLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$F_{CLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$F_{CLK}/2^5$	1KHz
	X	X	X	X	0	1	1	0	$F_{CLK}/2^6$	500KHz
	X	X	X	X	0	1	1	1	$F_{CLK}/2^7$	250KHz
	X	X	X	X	1	0	0	0	$F_{CLK}/2^8$	125KHz
	X	X	X	X	1	0	0	1	$F_{CLK}/2^9$	62.5KHz
	X	X	X	X	1	0	1	0	$F_{CLK}/2^{10}$	31.25KHz
	X	X	X	X	1	0	1	1	$F_{CLK}/2^{11}$	15.63KHz
	X	X	X	X	1	1	0	0	$F_{CLK}/2^{12}$	7.81KHz
	X	X	X	X	1	1	0	1	$F_{CLK}/2^{13}$	3.91KHz
	X	X	X	X	1	1	1	0	$F_{CLK}/2^{14}$	1.95KHz
X	X	X	X	1	1	1	1	$F_{CLK}/2^{15}$	977Hz	

注：要更改被选择为  $F_{CLK}$  的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m (STm) = 000FH）后进行更改。

X: 忽略

m: 单元号 (m=0) n: 通道号 (n=0)。

## 12.6.5 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表 12-34 所示：

表 12-34：溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。 →	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。	-	判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn (SDIRmn)写“1”。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号 (m=0) n：通道号 (n=0)。

## 12.7 UART (UART0~UART1) 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 00/10) 和接收专用 (奇数通道 01/11) 共 2 个通道来实现全双工异步 UART 通信。

### [数据的发送和接收]

- (1) 7 位、8 位、9 位或者 16 位的数据长度<sup>注</sup>
- (2) MSB/LSB 优先的选择
- (3) 发送和接收数据的电平设定 (选择电平是否反相)
- (4) 奇偶校验位的附加、奇偶校验功能
- (5) 停止位的附加、停止位的检测功能

### [中断功能]

- (1) 传送结束中断、缓冲器空中断
- (2) 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。

UART1 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。

UART 有以下 2 种通信运行:

- (1) UART 发送 (参照 12.7.1)
- (2) UART 接收 (参照 12.7.2)

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。

## 12.7.1 UART发送

UART 发送是本产品微控制器将数据异步发送到其他设备的运行。

UART 使用的 2 个通道中的偶数通道用于 UART 发送。

表 12-35: UART 发送

UART	UART0	UART1
对象通道	SCI0的通道0	SCI1的通道0
使用的引脚	TxD0	TxD1
中断	INTST0	INTST1
	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 <sub>MCK</sub> /6[bps]（SDR <sub>mn</sub> [15:9]≥3）、Min.F <sub>CLK</sub> /(2×2 <sup>11</sup> ×128)[bps] <sup>注</sup>	
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	

注意：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。

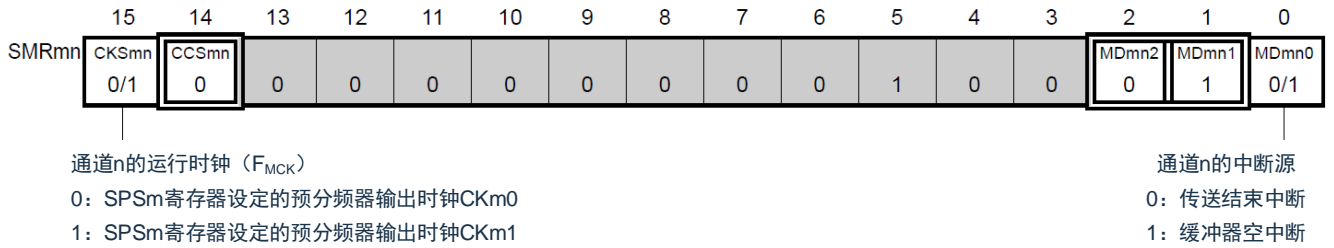
备注：

1. F<sub>MCK</sub>：对象通道的运行时钟频率
2. F<sub>CLK</sub>：系统时钟频率
3. m：单元号（m=0、1） n：通道号（n=0）。

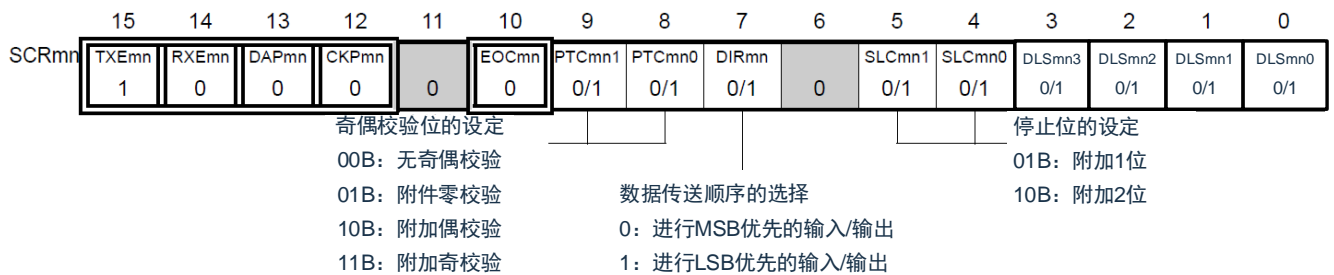
(1) 寄存器的设定

图 12-73: UART (UART0、UART1) 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)

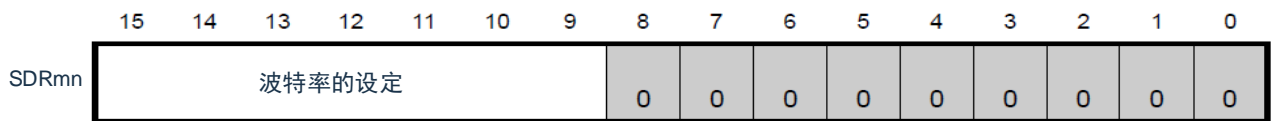


(b) 串行通信运行设定寄存器mn(SCRmn)

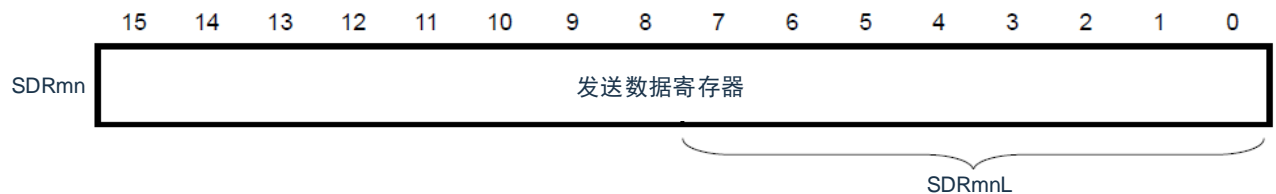


(c) 串行数据寄存器mn(SDRmn)

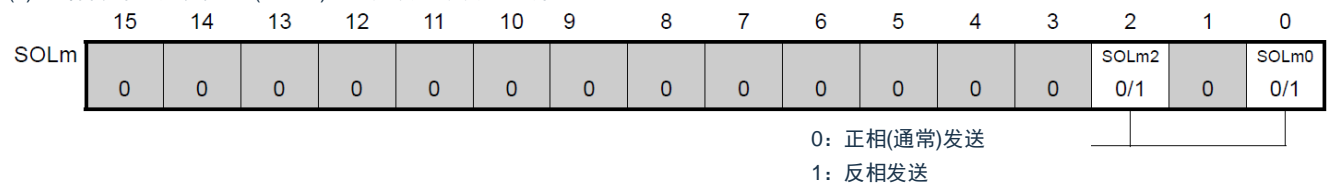
(1) 运行停止 (SEmn=0) 时



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

□: 在 SSPI 主控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 12-73: UART (UART0~UART1) 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

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

□：在 SSPI 主控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”。

(2) 操作步骤

图 12-74: UART 发送的初始设定步骤



图 12-75: UART 发送的中止步骤

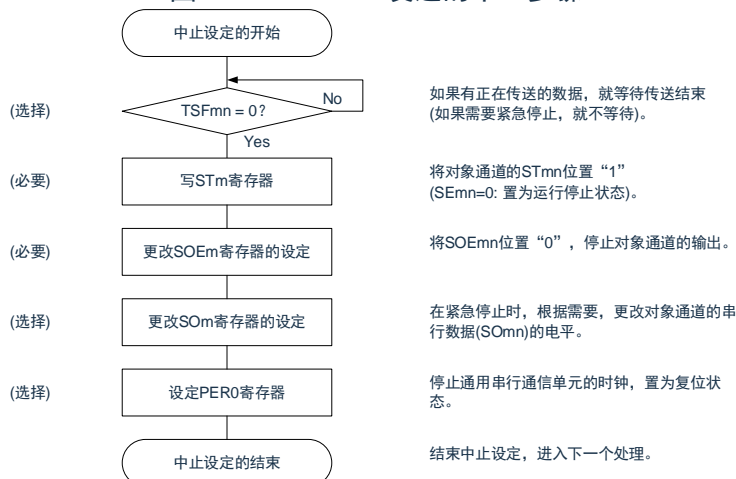
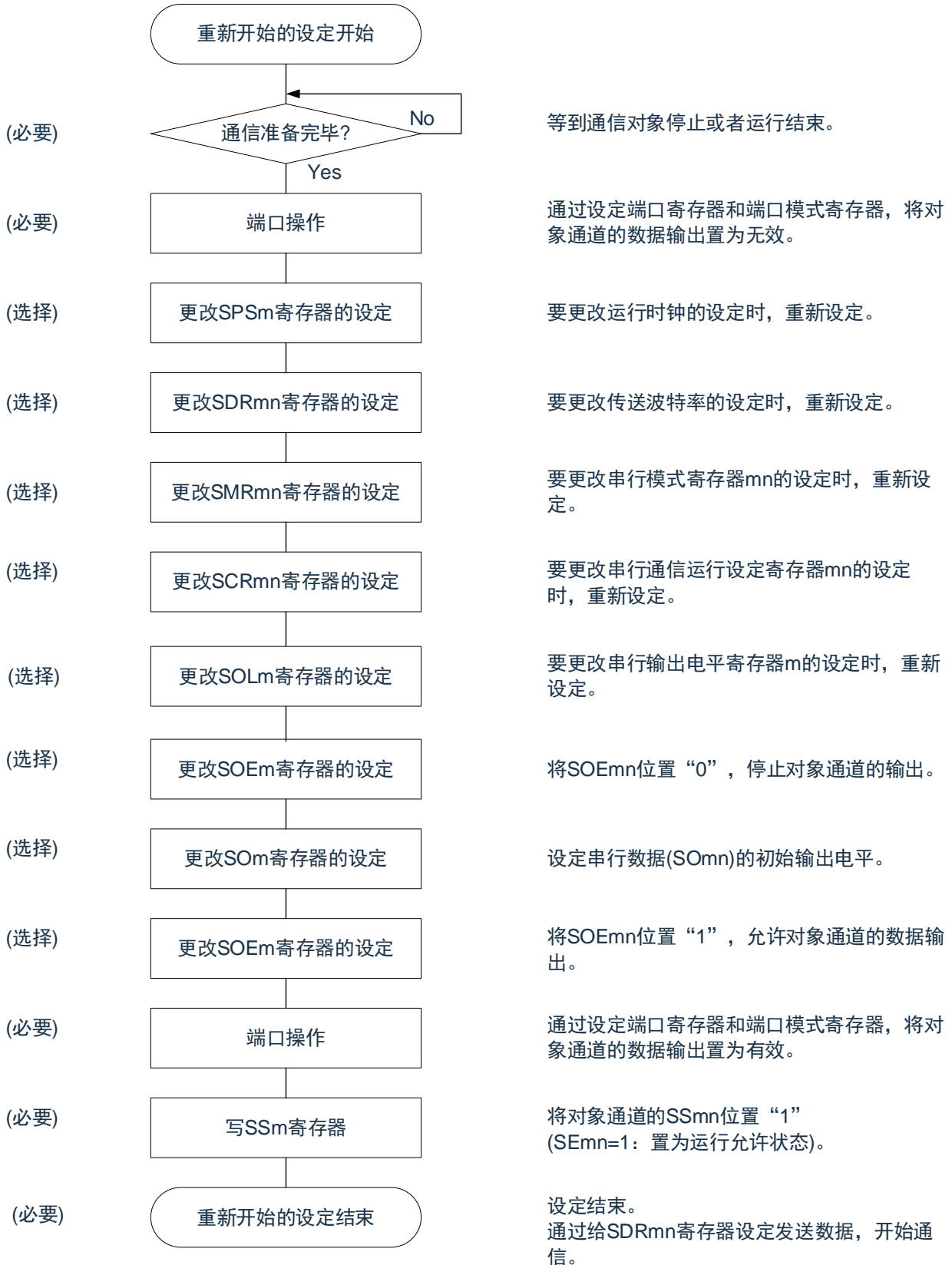


图 12-76: 重新开始 UART 发送的设定步骤



注意: 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 12-77: UART 发送（单次发送模式）的时序图

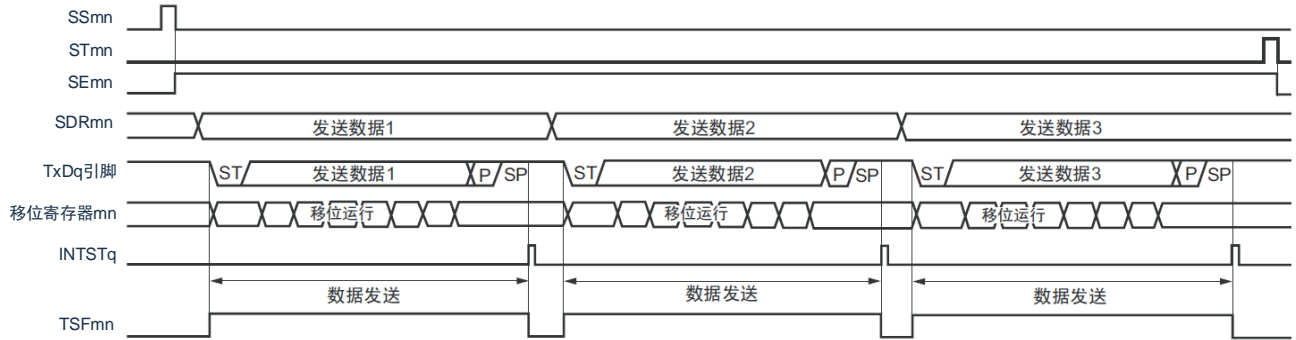
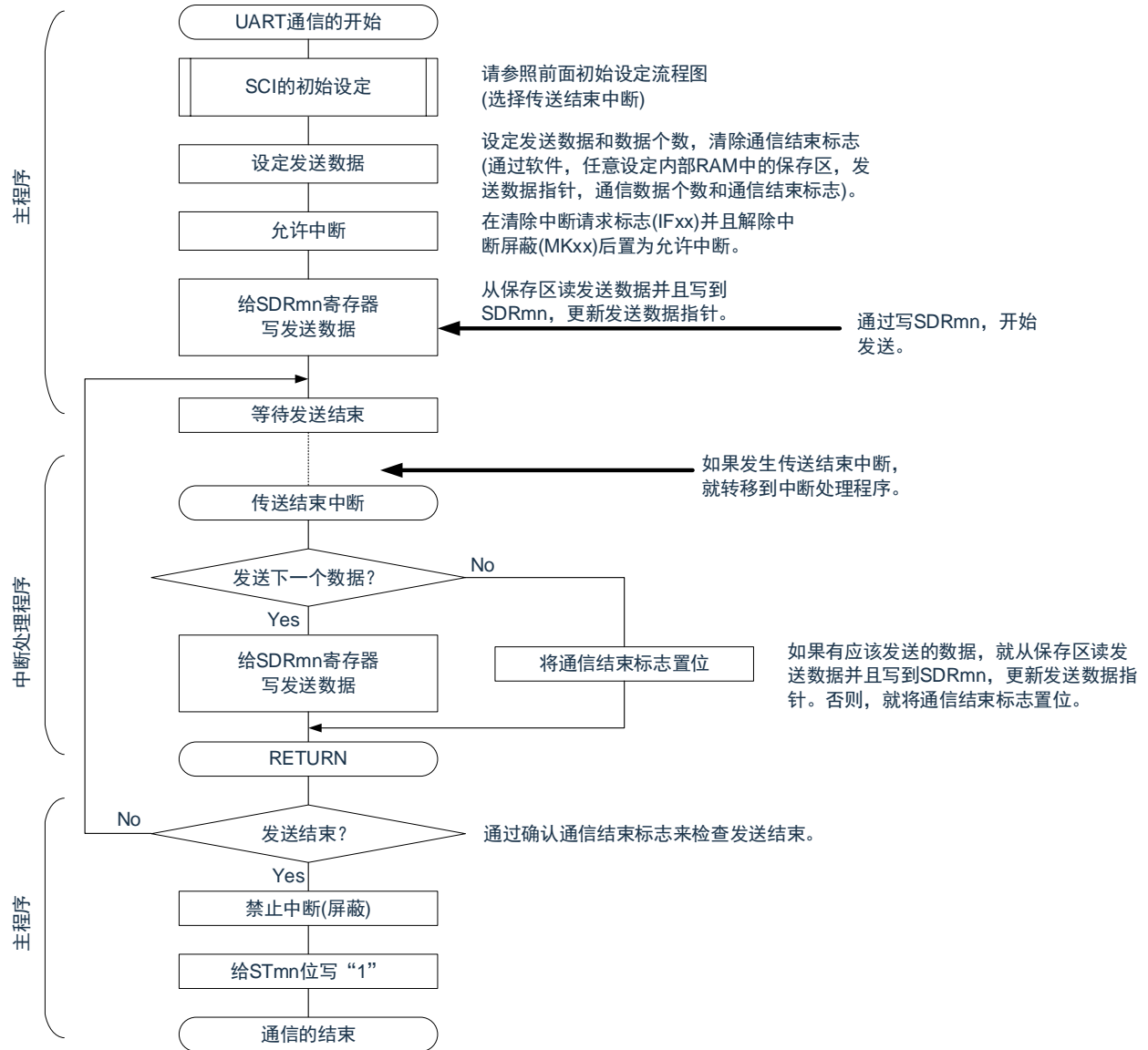


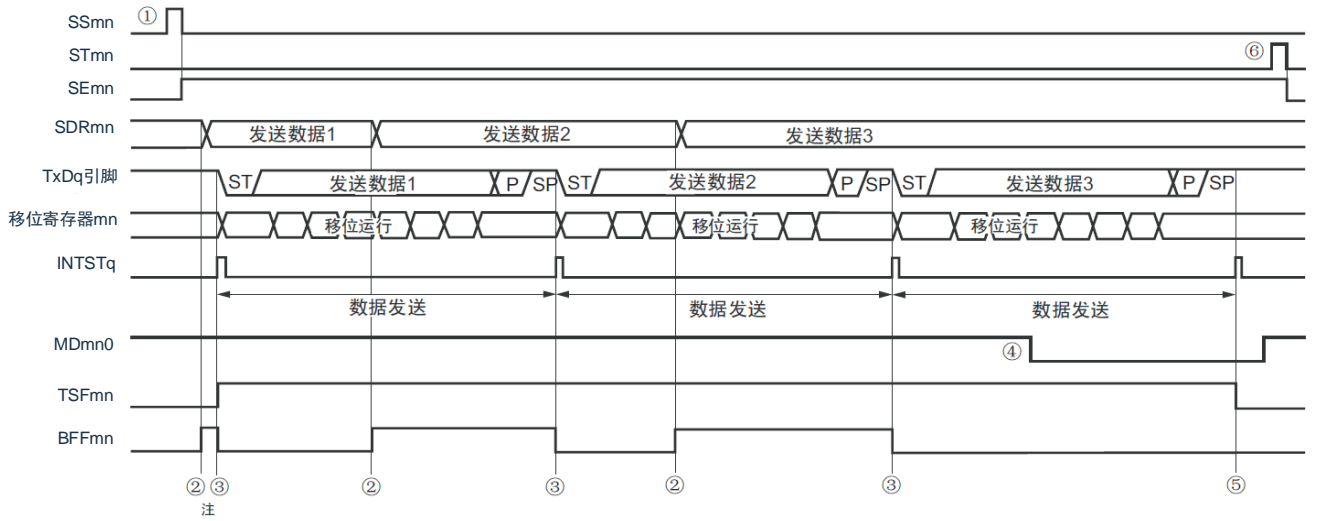
图 12-78: UART 发送（单次发送模式）的流程图



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

(4) 处理流程（连续发送模式）

图 12-79：UART 发送（连续发送模式）的时序图

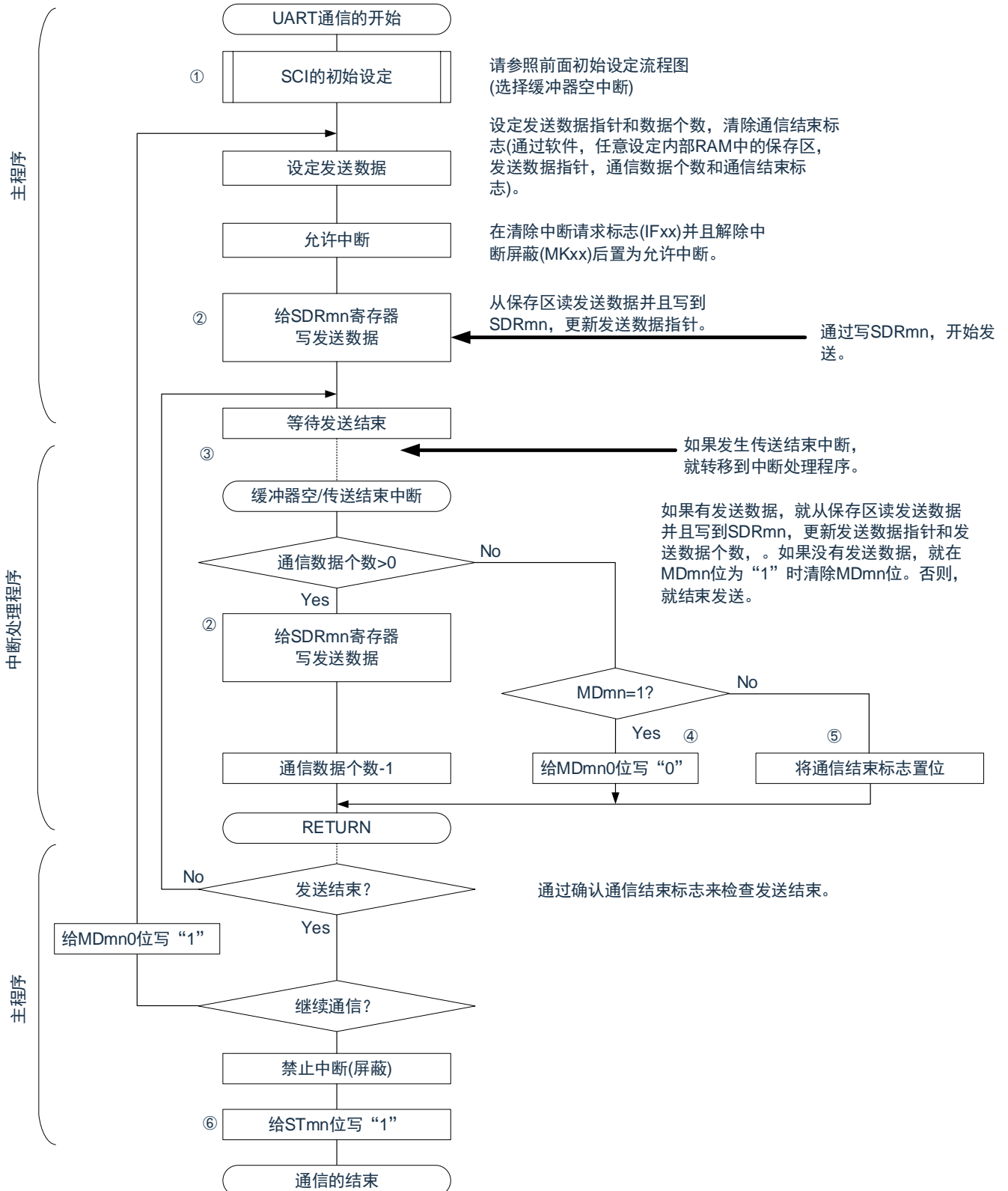


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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、1）q：UART 号（q=0、1）。

图 12-80: UART 发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 12-79 UART 发送（连续发送模式）的时序图”中的①~⑥。

## 12.7.2 UART接收

UART 接收是本产品微控制器其他设备异步接收数据的运行。

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

表 12-36: UART 接收

UART	UART0	UART1
对象通道	SCI0的通道1	SCI1的通道1
使用的引脚	RxD0	RxD1
中断	INTSR0	INTSR1
	只限于传送结束中断（禁止设定缓冲器空中断）。	
错误中断	INTSRE0	INTSRE1
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	$\text{Max.F}_{\text{MCK}}/6[\text{bps}]$ ( $\text{SDRmn}[15:9] \geq 2$ )、 $\text{Min.F}_{\text{CLK}}/(2 \times 2^{15} \times 128)[\text{bps}]$	
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	

注意：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。

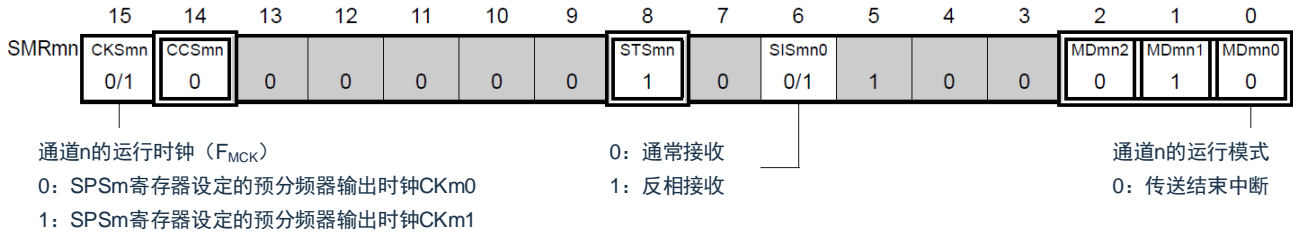
备注：

1.  $F_{\text{MCK}}$ : 对象通道的运行时钟频率
2.  $F_{\text{CLK}}$ : 系统时钟频率
3. m: 单元号 (m=0、1) n: 通道号 (n=1)。

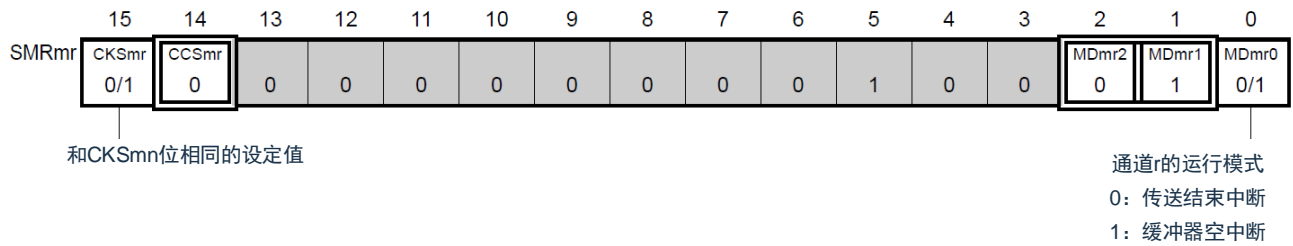
(1) 寄存器的设定

图 12-81: UART (UART0~UART1) 的 UART 接收时的寄存器设定内容例子(1/2)

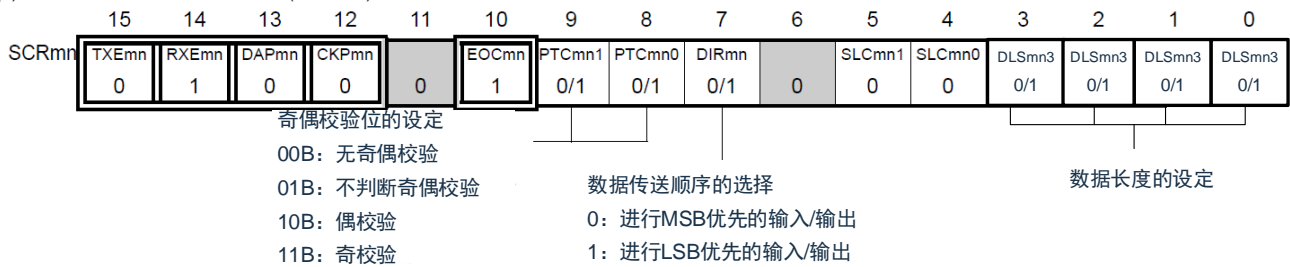
(a) 串行模式寄存器mn(SMRmn)



(b) 串行模式寄存器mr(SMRmr)

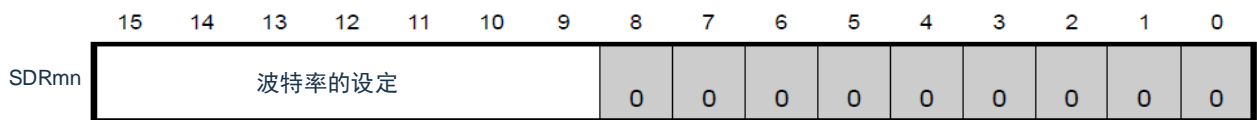


(c) 串行通信运行设定寄存器mn(SCRmn)

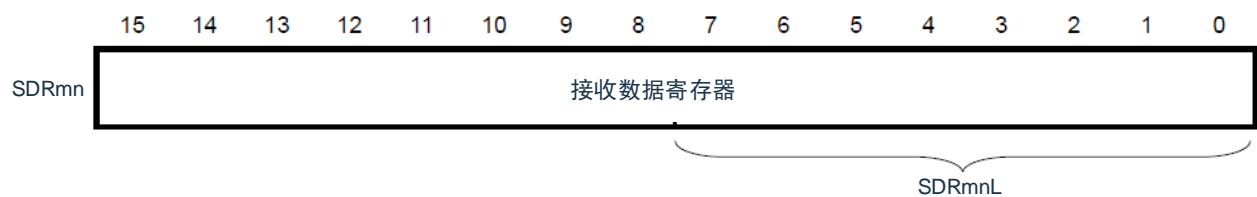


(d) 串行数据寄存器mn(SDRmn)

(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: m: 单元号 (m=0、1); n: 通道号 (n=1);

r: 通道号 (r=n-1); q: UART 号 (q=0~1)。

□: 在 SSPI 主控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 12-81: UART (UART0~UART1) 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							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	×

(f) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×

(g) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

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：m：单元号 (m=0、1)；n：通道号 (n=1)；

r：通道号 (r=n-1)；q：UART 号 (q=0~1)。

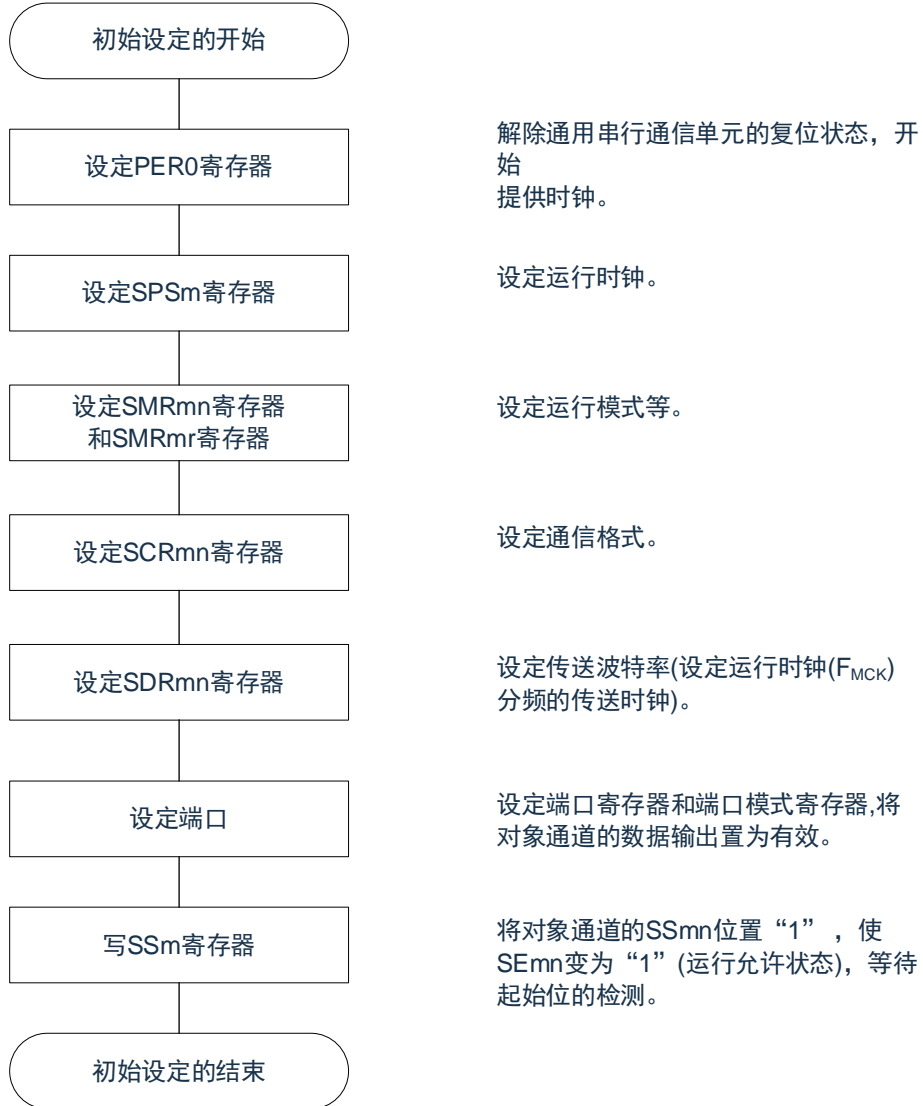
□：在 SSPI 主控接收模式中为固定设定；■：不能设定（设定初始值）；

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”。

(2) 操作步骤

图 12-82: 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  $F_{MCK}$  时钟, 然后将 SSmn 位置“1”。

图 12-83: UART 接收的中止步骤

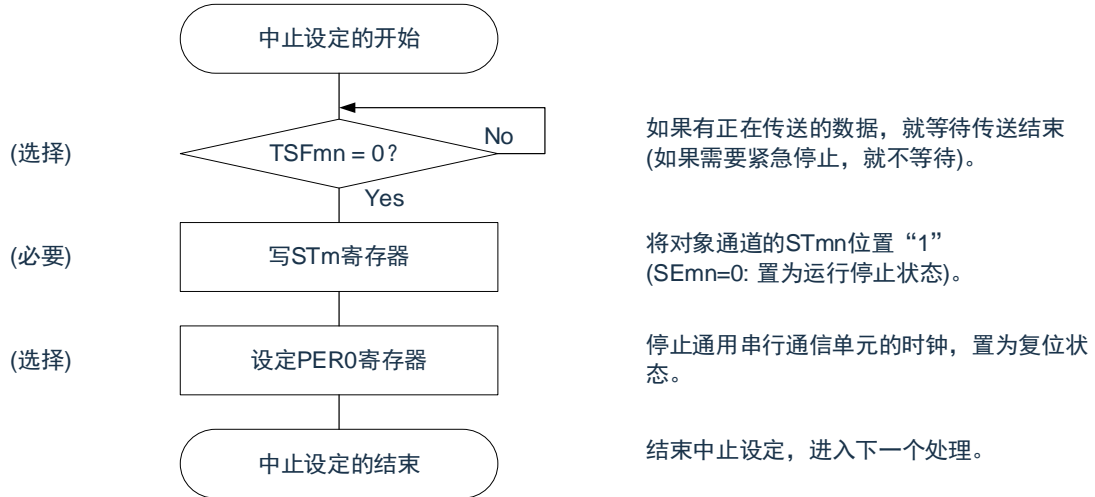


图 12-84: 重新开始 UART 接收的设定步骤

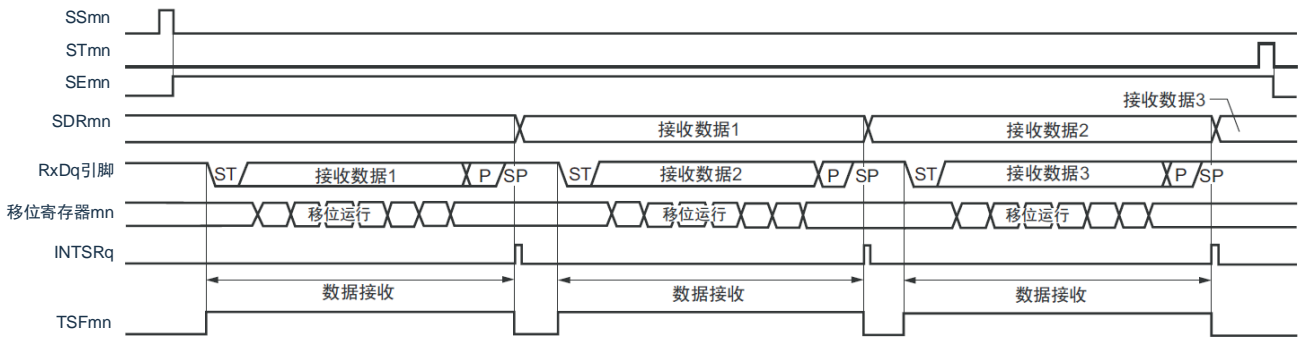


注意:

1. 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”。
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。

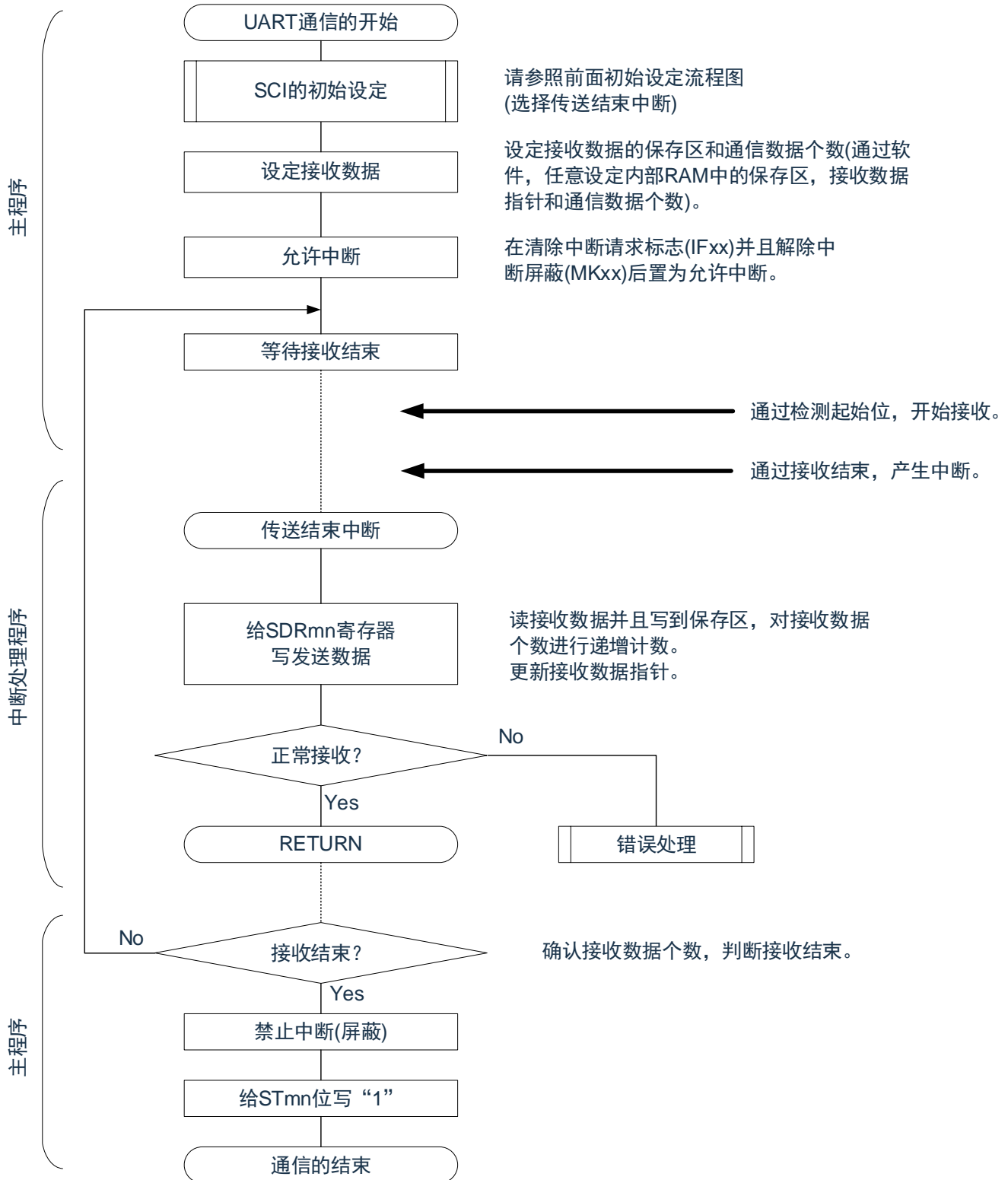
(3) 处理流程

图 12-85: 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、1) 。

图 12-86: UART 接收的流程图



### 12.7.3 波特率的计算

#### (1) 波特率的计算式

UART (UART0~UART1) 通信的波特率能用以下计算式进行计算:

$$\text{(波特率)} = \{\text{对象通道的运行时钟 (F}_{MCK}\text{) 频率}\} \div (\text{SDRmn}[15:9]+1) \div 2[\text{bps}]$$

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)。

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: 因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15~9 的值 (0000010B~1111111B), 所以为 2~127。

m: 单元号 (m=0、1) n: 通道号 (n=0、1)。

表 12-37: UART 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 (F <sub>CLK</sub> ) <sup>注</sup>	
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	F <sub>CLK</sub>	32MHz
	X	X	X	X	0	0	0	1	F <sub>CLK</sub> /2	16MHz
	X	X	X	X	0	0	1	0	F <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	X	X	X	X	0	0	1	1	F <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	X	X	X	X	0	1	0	0	F <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	X	X	X	X	0	1	0	1	F <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	X	X	X	X	0	1	1	0	F <sub>CLK</sub> /2 <sup>6</sup>	500KHz
	X	X	X	X	0	1	1	1	F <sub>CLK</sub> /2 <sup>7</sup>	250KHz
	X	X	X	X	1	0	0	0	F <sub>CLK</sub> /2 <sup>8</sup>	125KHz
	X	X	X	X	1	0	0	1	F <sub>CLK</sub> /2 <sup>9</sup>	62.5KHz
	X	X	X	X	1	0	1	0	F <sub>CLK</sub> /2 <sup>10</sup>	31.25KHz
	X	X	X	X	1	0	1	1	F <sub>CLK</sub> /2 <sup>11</sup>	15.63KHz
	X	X	X	X	1	1	0	0	F <sub>CLK</sub> /2 <sup>12</sup>	7.81KHz
	X	X	X	X	1	1	0	1	F <sub>CLK</sub> /2 <sup>13</sup>	3.91KHz
	X	X	X	X	1	1	1	0	F <sub>CLK</sub> /2 <sup>14</sup>	1.95KHz
X	X	X	X	1	1	1	1	F <sub>CLK</sub> /2 <sup>15</sup>	977Hz	
1	0	0	0	0	X	X	X	X	F <sub>CLK</sub>	32MHz
	0	0	0	1	X	X	X	X	F <sub>CLK</sub> /2	16MHz
	0	0	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	0	0	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	0	1	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	0	1	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	0	1	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>6</sup>	500KHz
	0	1	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>7</sup>	250KHz
	1	0	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>8</sup>	125KHz
	1	0	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>9</sup>	62.5KHz
	1	0	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>10</sup>	31.25KHz
	1	0	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>11</sup>	15.63KHz
	1	1	0	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>12</sup>	7.81KHz
	1	1	0	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>13</sup>	3.91KHz
	1	1	1	0	X	X	X	X	F <sub>CLK</sub> /2 <sup>14</sup>	1.95KHz
1	1	1	1	X	X	X	X	F <sub>CLK</sub> /2 <sup>15</sup>	977Hz	

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。

X：忽略

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~UART1) 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。

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

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

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

备注：m：单元号 (m=0、1) n：通道号 (n=0)。



(3) 接收时的波特率容许范围

UART (UART0~UART1) 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。

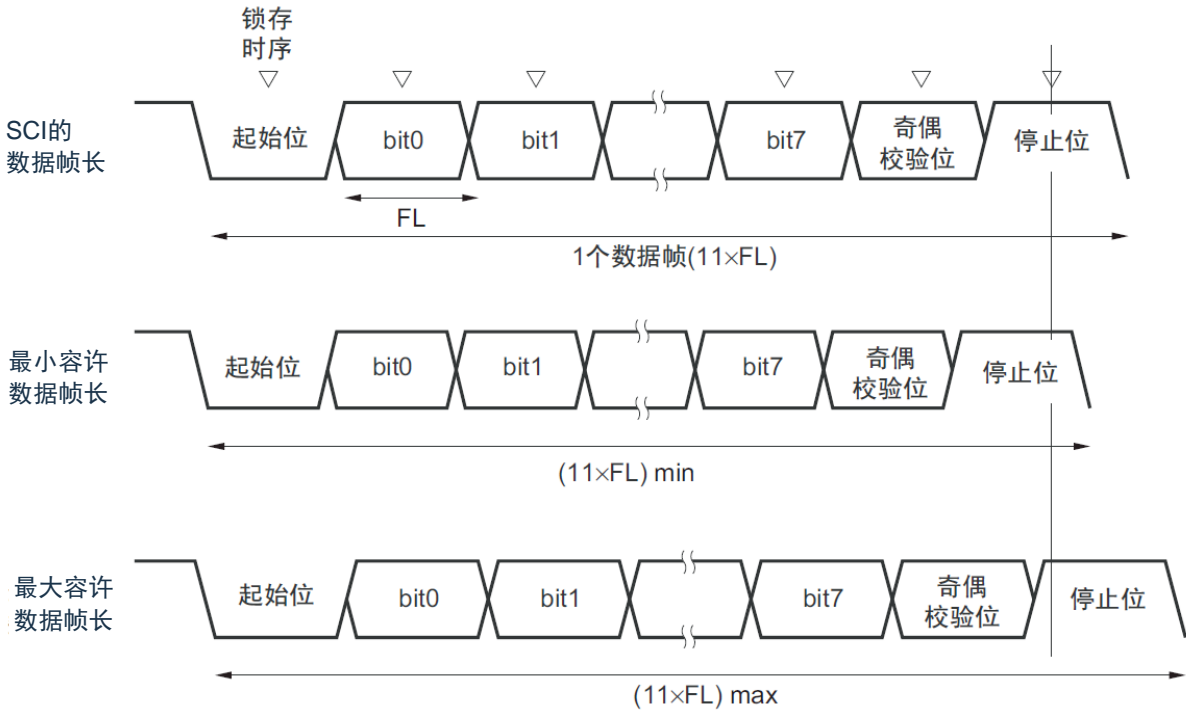
$\text{(可接收的最大波特率)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$
$\text{(可接收的最小波特率)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$

Brate: 接收方的波特率的计算值 (参照“12.7.4 (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)。

图 12-87: 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 12-87 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 (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。

备注: m: 单元号 (m=0、1) ; n: 通道号 (n=1)。

## 12.7.4 在UART (UART0~UART1) 通信过程中发生错误时的处理步骤

在 UART (UART0~UART1) 通信过程中发生错误时的处理步骤如表 12-38 和表 12-39 所示：

表 12-38：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)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SDIRmn) 写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

表 12-39：发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn (SIRmn)。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m (STm) 的STmn位置“1”。	串行通道允许状态寄存器m (SEm) 的SEmn位为“0”并且通道n为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存m (SSm) 的SSmn位置“1”。	串行通道允许状态寄存器m (SEm) 的SEmn位为“1”并且通道n为可运行状态。	

备注：m：单元号 (m=0、1) n：通道号 (n=0、1)。

## 12.8 LIN通信的运行

### 12.8.1 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。

表 12-40：LIN 发送

UART	UART0	UART1
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 <sup>注</sup>	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	

注意：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：

1. F<sub>MCK</sub>：对象通道的运行
2. 时钟频率 F<sub>CLK</sub>：系统时钟频率。

LIN 是 Local Interconnect Network 的简称，通信速率为 1~20Kbps。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。

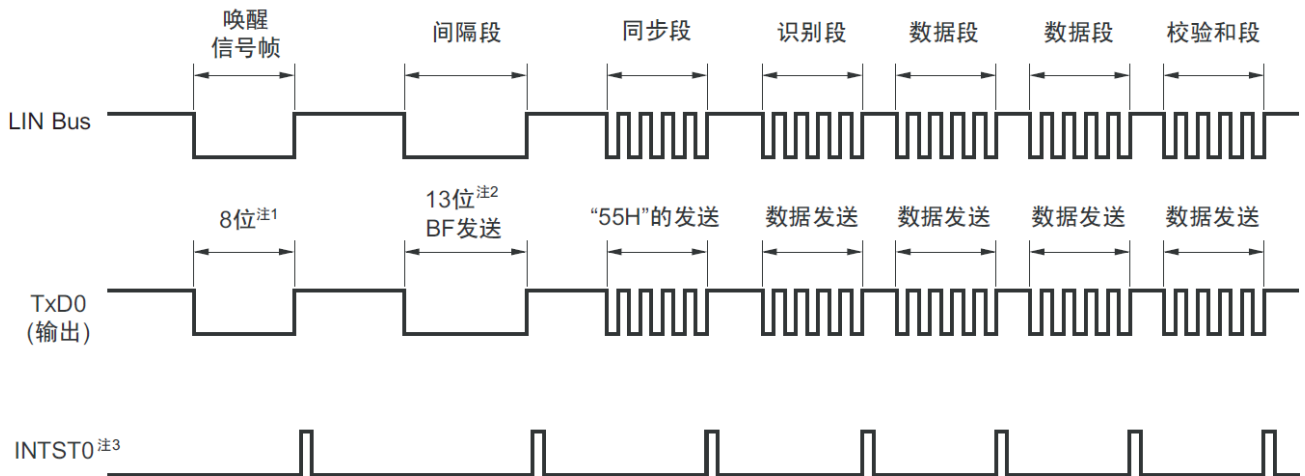
LIN 主控一般连接 CAN（Controller Area Network）等的网络。

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 12-88 所示：

图 12-88：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。

注 2：间隔段规定为 13 位宽的低电平输出，因此假设主传送使用的波特率为 N[bps]，间隔段使用的波特率如下：

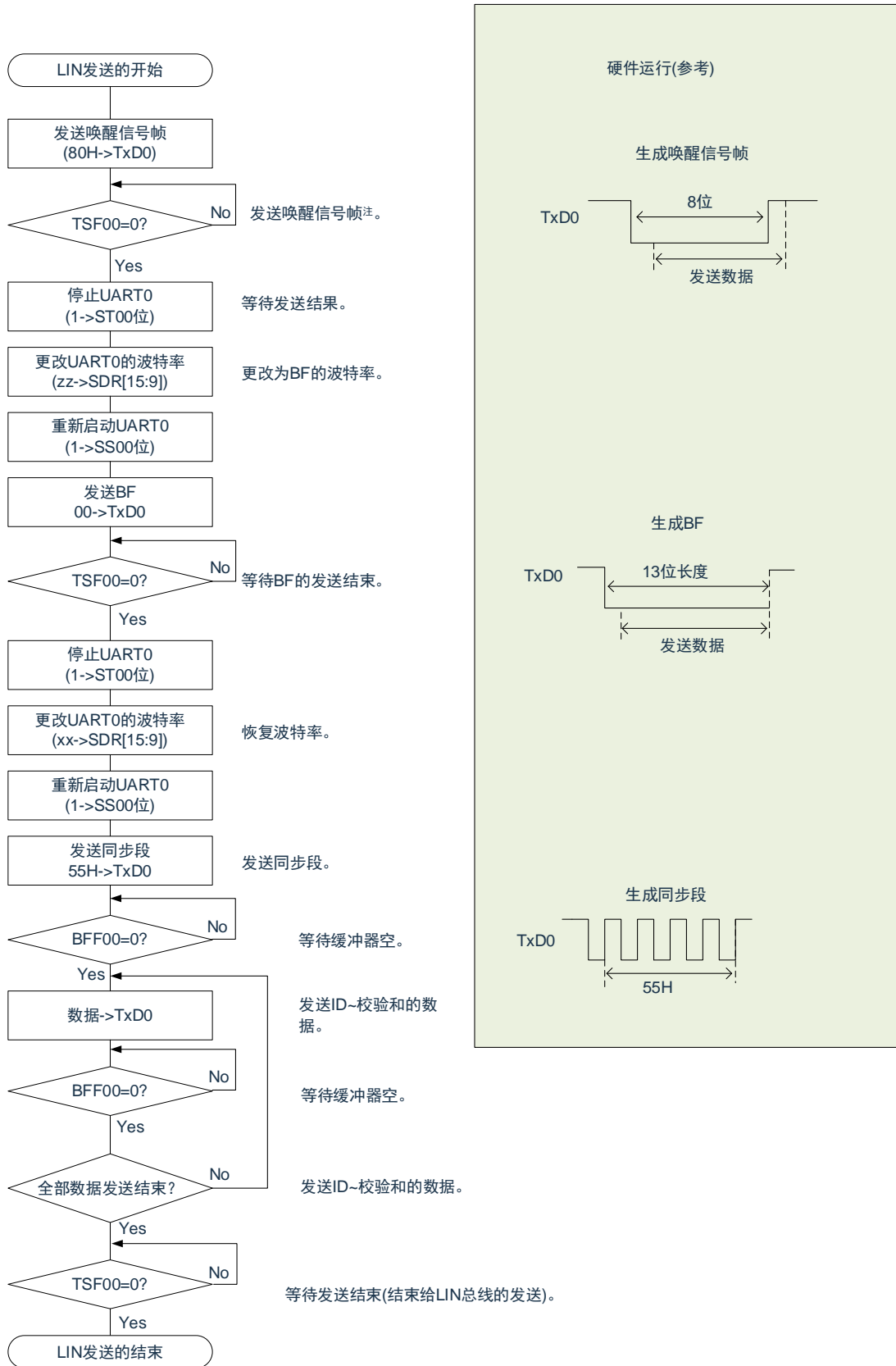
$$(间隔段的波特率)=9/13 \times N$$

通过此波特率发送“00H”的数据，生成间隔段。

注 3：在各数据发送结束时输出 INTST0，而且在 BF 发送时也输出 INTST0。

备注：由软件控制各段间的间隔。

图 12-89: LIN 发送的流程图



备注:

1. 只限于从 LIN-bus 睡眠状态启动的情况。
2. 这是从结束 UART 的初始设定并且允许从属发送开始的流程。

## 12.8.2 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。

表 12-41: LIN 接收

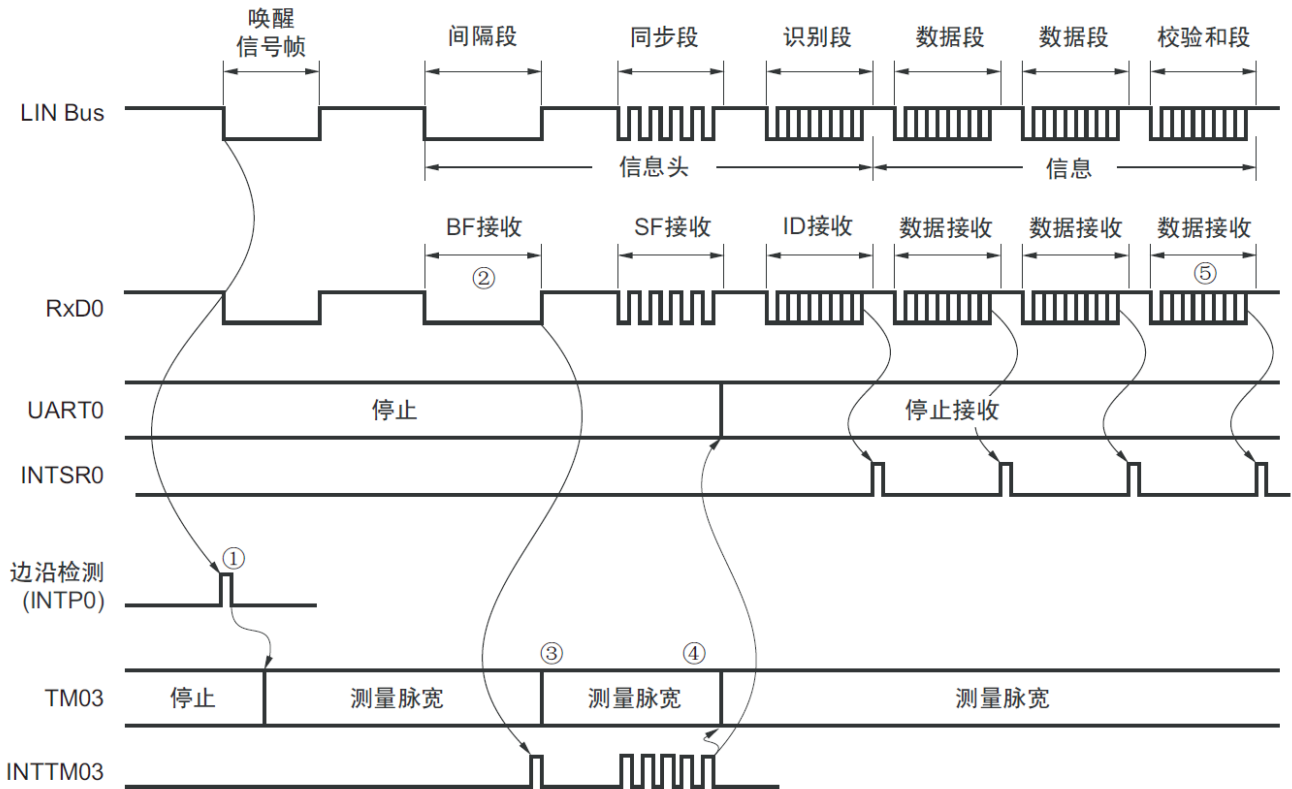
UART	UART0	UART1
LIN通信的支持	能	不能
对象通道	SCI0的通道1	—
使用的引脚	RxD0	—
中断	INTSR0	—
	只限于传送结束中断（禁止设定缓冲器空中断）。	
错误中断	INTSRE0	—
错误检测标志	帧错误检测标志（FEF01） 溢出错误检测标志（OVF01）	
传送数据长度	8位	
传送速率 <sup>注</sup>	Max.F <sub>MCK</sub> /6[bps]（SDR01[15:9]≥2）、Min.F <sub>CLK</sub> /(2×2 <sup>15</sup> ×128)[bps]	
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。	
奇偶校验位	无奇偶校验位（不进行奇偶校验）。	
停止位	附加1位。	
数据方向	LSB优先	

注意：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。

F<sub>MCK</sub>：对象通道的运行 3、时钟频率 F<sub>CLK</sub>：系统时钟频率。

LIN 接收操作的概要如图 12-90 所示：

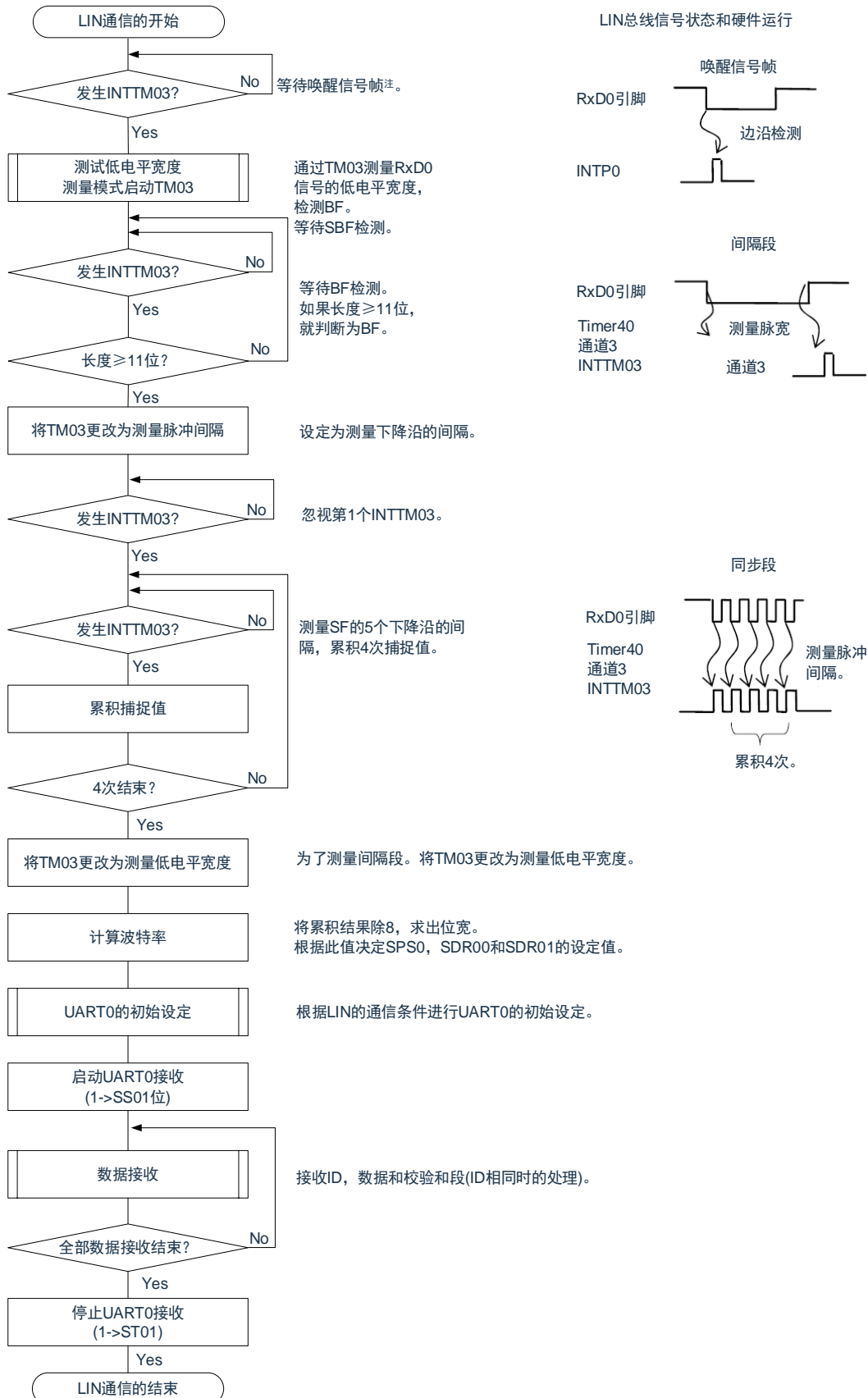
图 12-90：LIN 的接收操作



信号处理的流程如下：

- (1) 通过检测引脚的中断边沿 (INTP0) 来检测唤醒信号。当检测到唤醒信号时，为了测量 BF 的低电平宽度，将 TM03 设定为测量脉宽，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 12-91：LIN 接收的流程图



注意：只在睡眠状态下才需要。

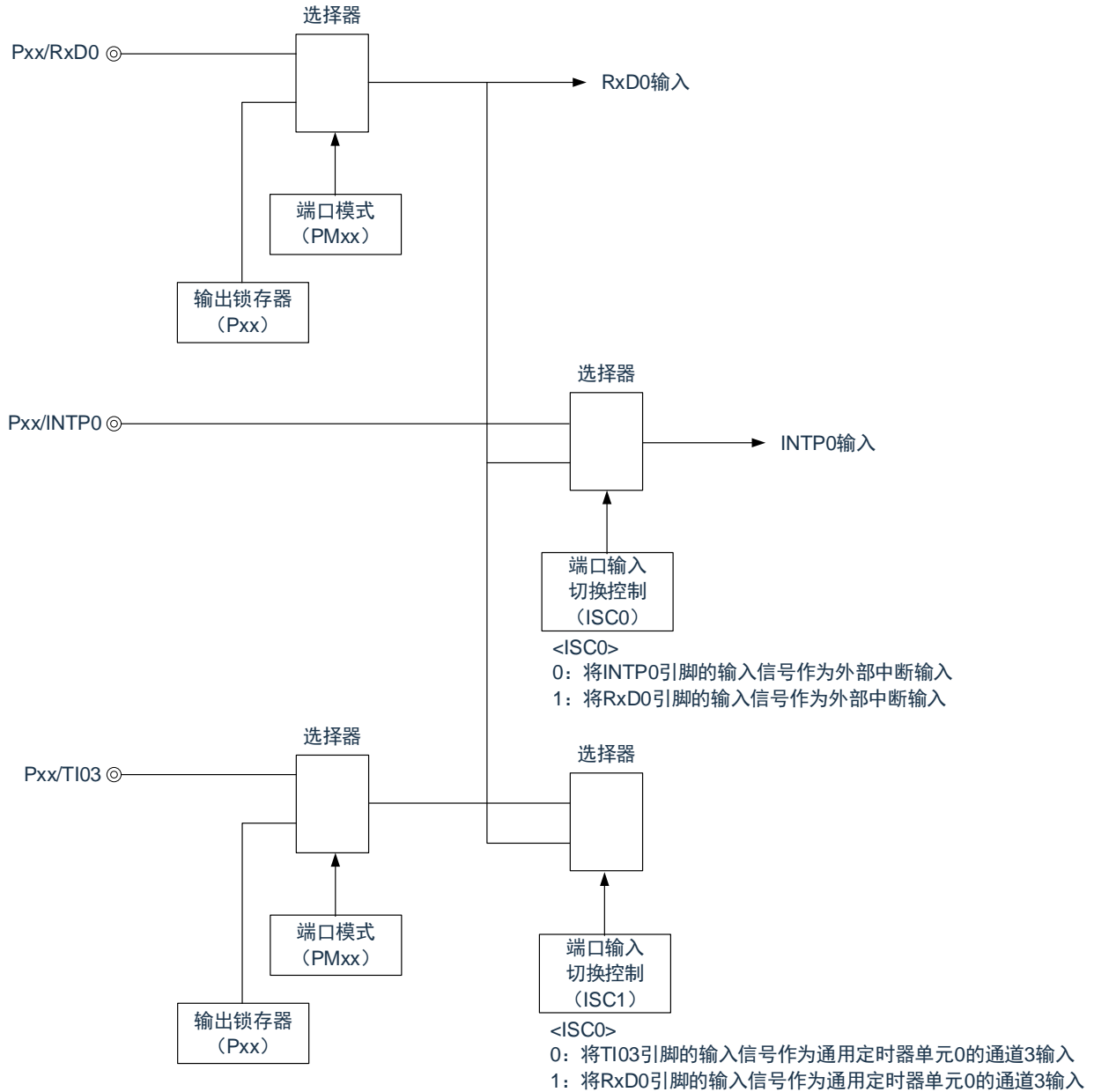


用于 LIN 接收操作的端口结构图如图 12-92 所示。

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 12-92：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>

(1) 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。

(2) 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)。

(3) 通用串行通信单元 0 (SCI0) 的通道 0 和通道 1 (UART0)

# 第13章 串行接口SPI

## 13.1 串行接口SPI的功能

串行接口 SPI 有以下 2 种模式。

(1) 运行停止模式

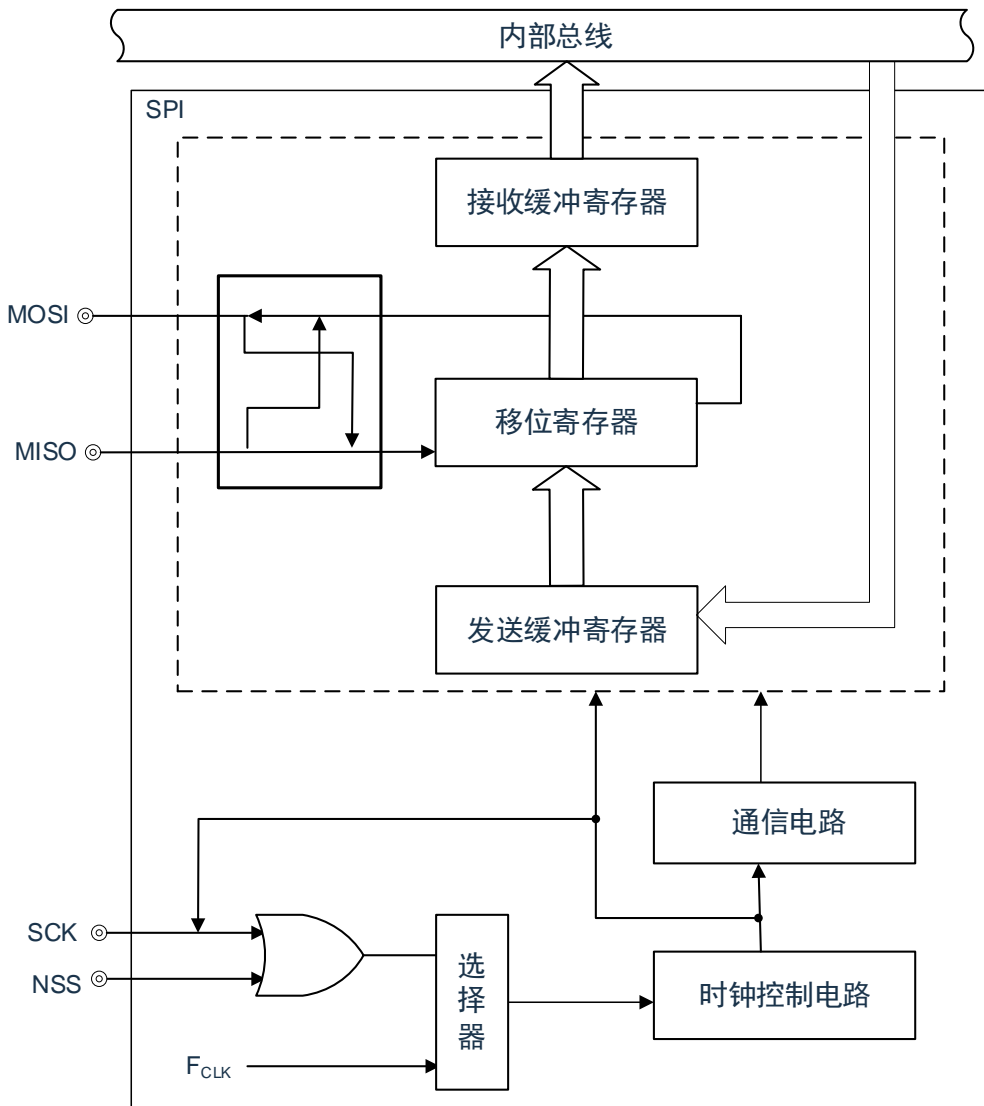
这是用于不进行串行传送时的模式，能降低功耗。

(2) 3-wire 串行 I/O 模式

此模式通过串行时钟（SCK）和串行数据总线（MISO 和 MOSI）的 3 条线，与多个设备进行 8 位或 16 位数据传送。

## 13.2 串行接口SPI的结构

图 13-1：串行接口 SPI 的框图



## 13.3 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。

- (1) 外围允许寄存器 0 (PER0)
- (2) 串行操作模式寄存器 (SPIM)
- (3) 串行时钟选择寄存器 (SPIC)
- (4) 发送缓冲寄存器 (SDRO)
- (5) 接收缓冲寄存器 (SDRI)
- (6) 端口模式寄存器 (PMxx)
- (7) 端口模式控制寄存器 (PMCxx)
- (8) 端口寄存器 (Pxx)

### 13.3.1 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 SPI 功能时，必须将 SPIEN 置“1”。

详细请参见“4.3.6 外围允许寄存器 0、1 (PER0、PER1)”

### 13.3.2 SPI操作模式寄存器（SPIM）

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 SPIM。

产生复位信号将该寄存器清除为 00H。

表 13-1：模式控制寄存器（SPIM）的格式

地址： 符号	0x40042400 7	复位后： 6	00H 5	4	3	2	1	0
						R/W <sup>注1</sup>		
SPIM	SPIE	TRMD	NSSE	DIR	INTMD	DLS	SDRIF	SPTF
SPIE		SPI运行的允许						
0		停止运行。						
1		允许运行。						
TRMD <sup>注3</sup>		发送/接收模式控制						
0		接收模式						
1		发送/接收模式						
NSSE <sup>注4</sup>		NSS引脚使用选择						
0		未使用NSS引脚						
1		使用NSS引脚						
DIR		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。						
INTMD		中断源选择						
0		传送结束中断						
1		发送缓冲器空中断						
DLS		数据长度的设定						
0		8位数据长度						
1		16位数据长度						
SDRIF		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						
SPTF <sup>注2</sup>		通信状态标志位						
0		通信停止						
1		通信正在进行中						

注 1：位 0 和位 1 为只读位。

注 2：SPTF=1（串行通信期间）时，禁止重写 TRMD,DIR,NSSE。

注 3：TRMD 为 0 时 MO 或 SO 输出固定为低电平。

注 4：将该位置为 1 之前，将 NSS 引脚输入电平固定为 0 或者 1。

### 13.3.3 SPI时钟选择寄存器（SPIC）

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 00H。

表 13-2: 时钟选择寄存器（SPIC）的格式

地址:	0x40042404	复位后:	00H	R/W				
符号	7	6	5	4	3	2	1	0
SPIC	0	0	0	CKP	DAP	CKS2	CKS1	CKS0

CKP	DAP	数据发送/接收时序的指定	类型
0	0		1
0	1		2
1	0		3
1	1		4

CKS2	CKS1	CKS0	SPI串行时钟选择	模式
0	0	0	$F_{CLK}$	主机模式
0	0	1	$F_{CLK}/2$	
0	1	0	$F_{CLK}/2^2$	
0	1	1	$F_{CLK}/2^3$	
1	0	0	$F_{CLK}/2^4$	
1	0	1	$F_{CLK}/2^5$	
1	1	0	$F_{CLK}/2^6$	
1	1	1	从SCK输入的外部时钟	从机模式

注意:

1. SPIE=1（操作使能）时禁止写入 SPIC。
2. 复位后数据时钟的相位类型为类型 1。

### 13.3.4 发送缓冲寄存器 (SDRO)

该寄存器设置发送数据。

当将串行操作模式寄存器 (SPIM) 的位 7 (SPIE) 和位 6 (TRMD) 置为 1 时, 通过将数据写入 SDRO 开始发送/接收。

串行 I/O 移位寄存器把 SDRO 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 SDRO。

产生复位信号将该寄存器清除为 0000H。

表 13-3: 发送缓冲寄存器 (SDRO) 的格式

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R/W
SDRO	SDRO																

### 13.3.5 接收缓冲寄存器 (SDRI)

该寄存器存储接收到的数据。

如果将串行操作模式寄存器 (SPIM) 的位 6 (TRMD) 置为 0, 则通过从 SDRI 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 SDRI 中。

可用 8 位或 16 位存储操作指令读取 SDRI。

产生复位信号将该寄存器清除为 0000H。

表 13-4: 接收缓冲寄存器 (SDRI) 的格式

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

### 13.3.6 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”。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“2.3 控制端口功能的寄存器”。

## 13.4 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，SPIM 的位 0 (SPTF) 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 (SPIIF)，并将 SPTF 清除为 0。然后使能下一次通信。

注意：

1. SPTF=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 ( $T_{KCY}$ ) 特性的范围内使用。详细内容请参照数据手册。

### 13.4.1 主控的发送和接收

如果串行操作模式寄存器 (SPIM) 的位 6 (TRMD) 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 (SDRO) 时，开始发送/接收。

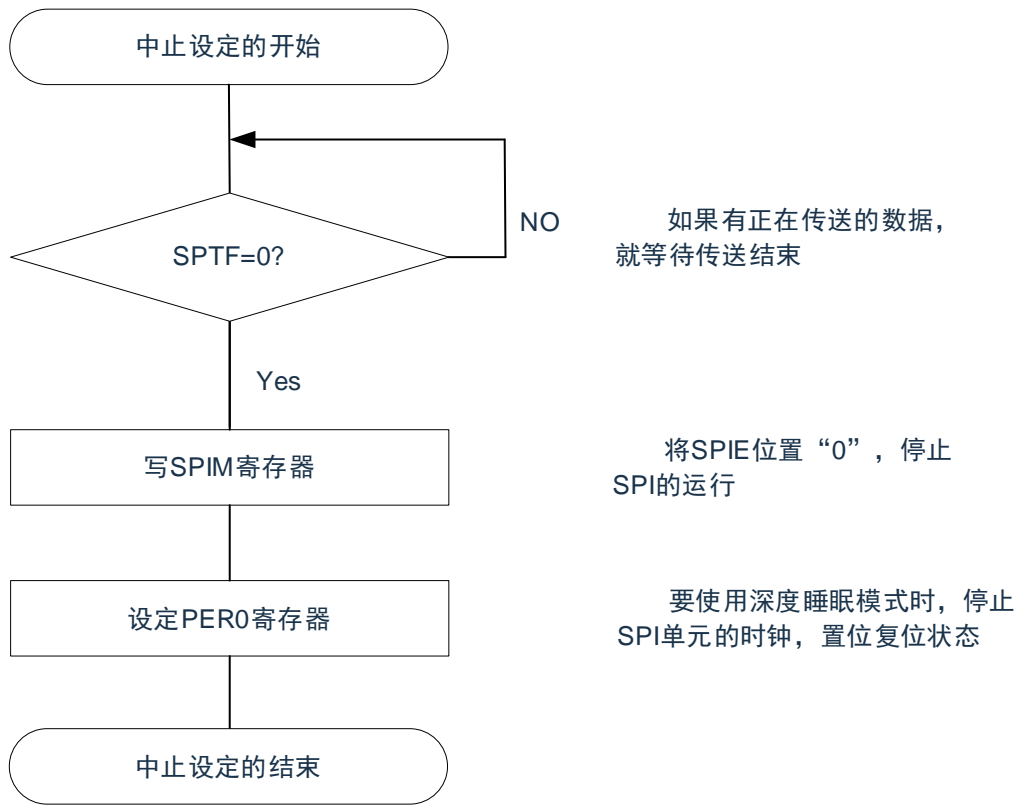
#### (1) 操作步骤

图 13-2: 主控发送/接收的初始设置步骤





图 13-3: 主控发送/接收的中止步骤



(2) 处理流程

图 13-4: 发送/接收时序(单次发送模式)的时序图 (INTMD=0,DAP=0、CKPmn=0)

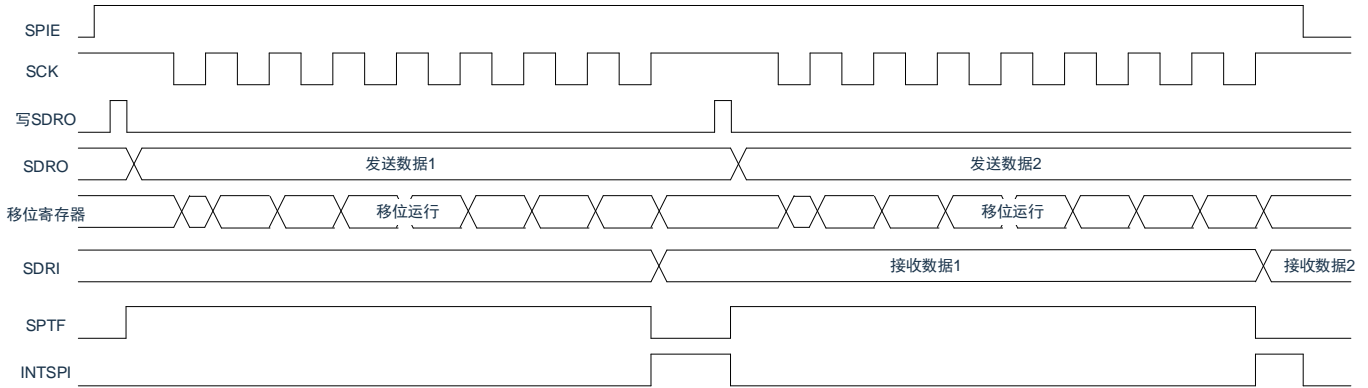
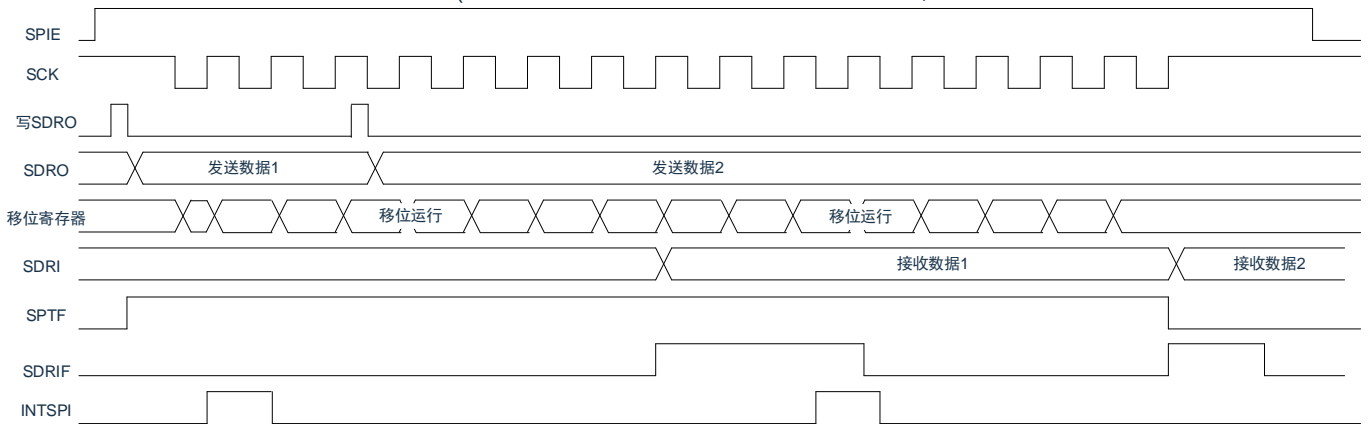


图 13-5: 发送/接收时序(连续发送模式)的时序图 (INTMD=1,DAP=0、CKPmn=0)



### 13.4.2 主控的接收

如果串行操作模式寄存器（SPIM）的位 6（TRMD）为 0，则可以只接收数据。当从接收缓冲寄存器（SDRI）中读取数据时，开始接收。

(1) 操作步骤

图 13-6：主控接收的初始设置步骤

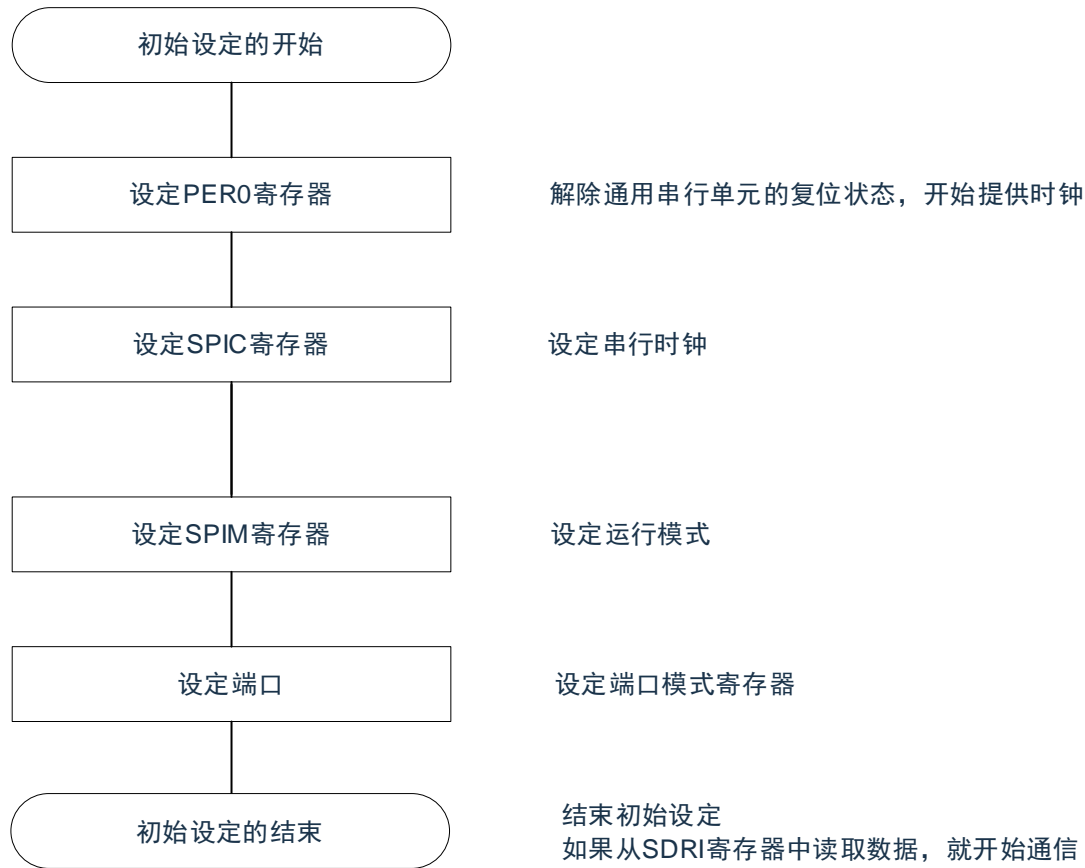
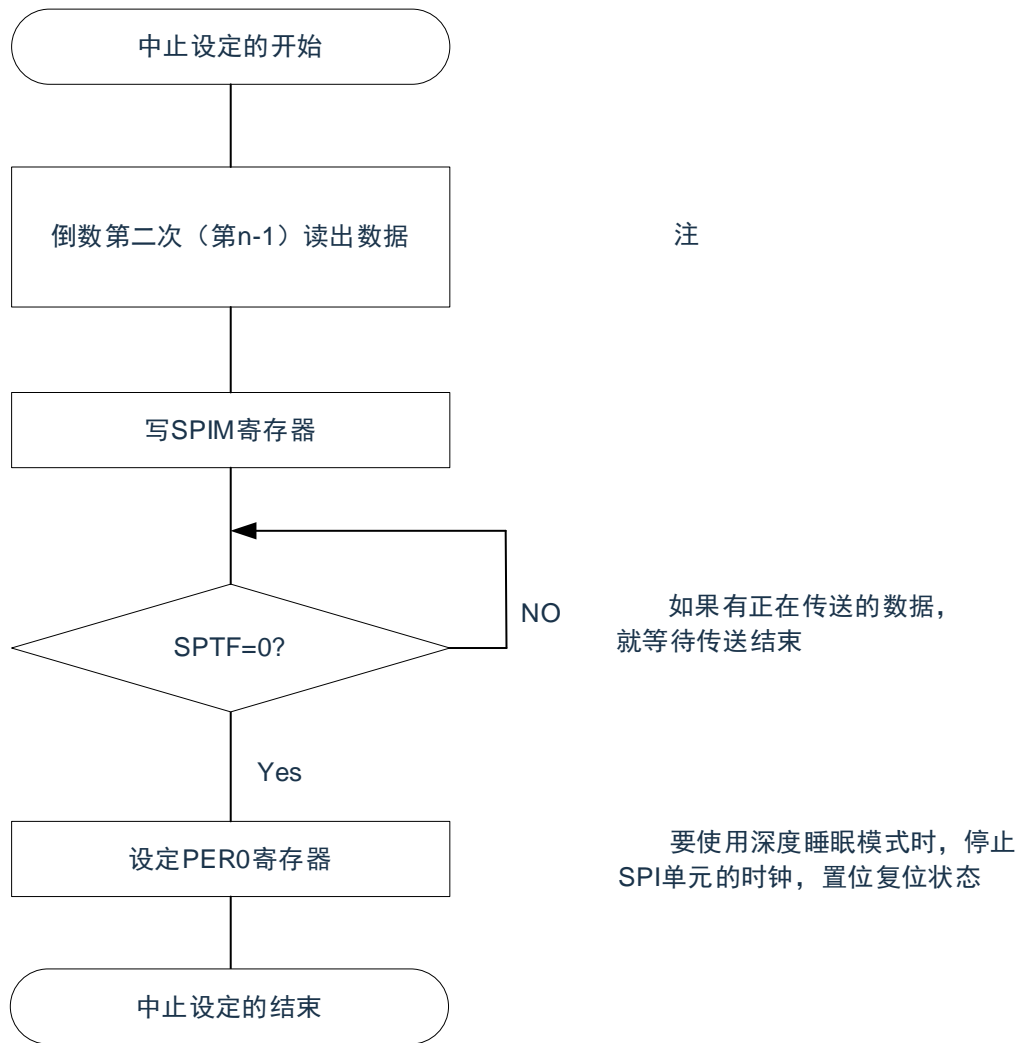


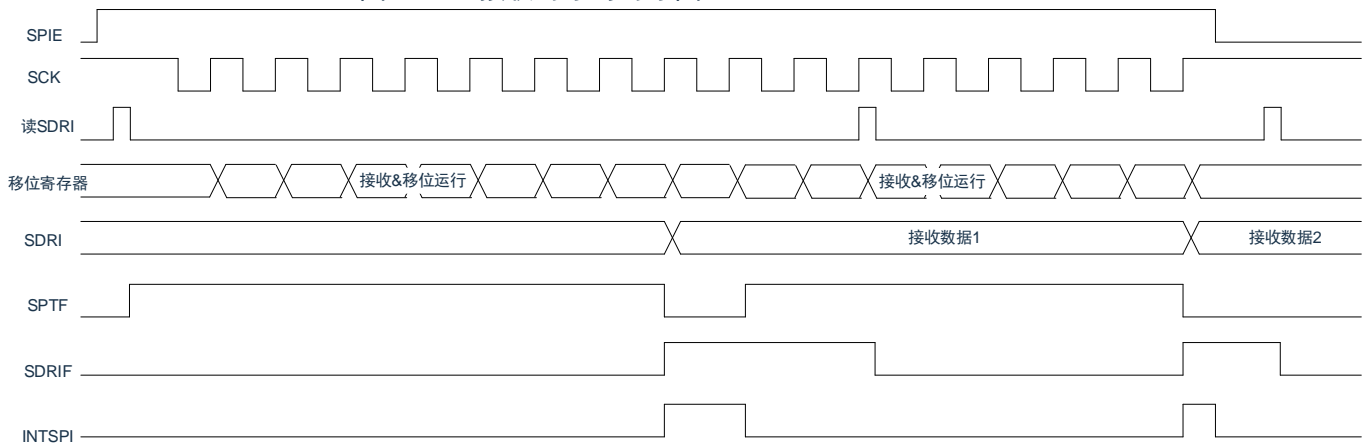
图 13-7: 主控接收的中止步骤



注：只接收模式下，SPI 传输由读取 SDRI 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 SDRI 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 SPIE。SPI 的传输将在最后一个数据传输完成后中止。

(2) 处理流程

图 13-8: 接收时序的时序图 (DAP=0、CKPmn=0)



### 13.4.3 从属的发送和接收

如果串行时钟选择寄存器（SPIC）的 CKS2-0 位选择从机模式，串行操作模式寄存器（SPIM）的位 6（TRMD）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（SDRO）后，等待主控设备的时钟，开始发送/接收。

(1) 操作步骤

图 13-9：从属发送/接收的初始设置步骤

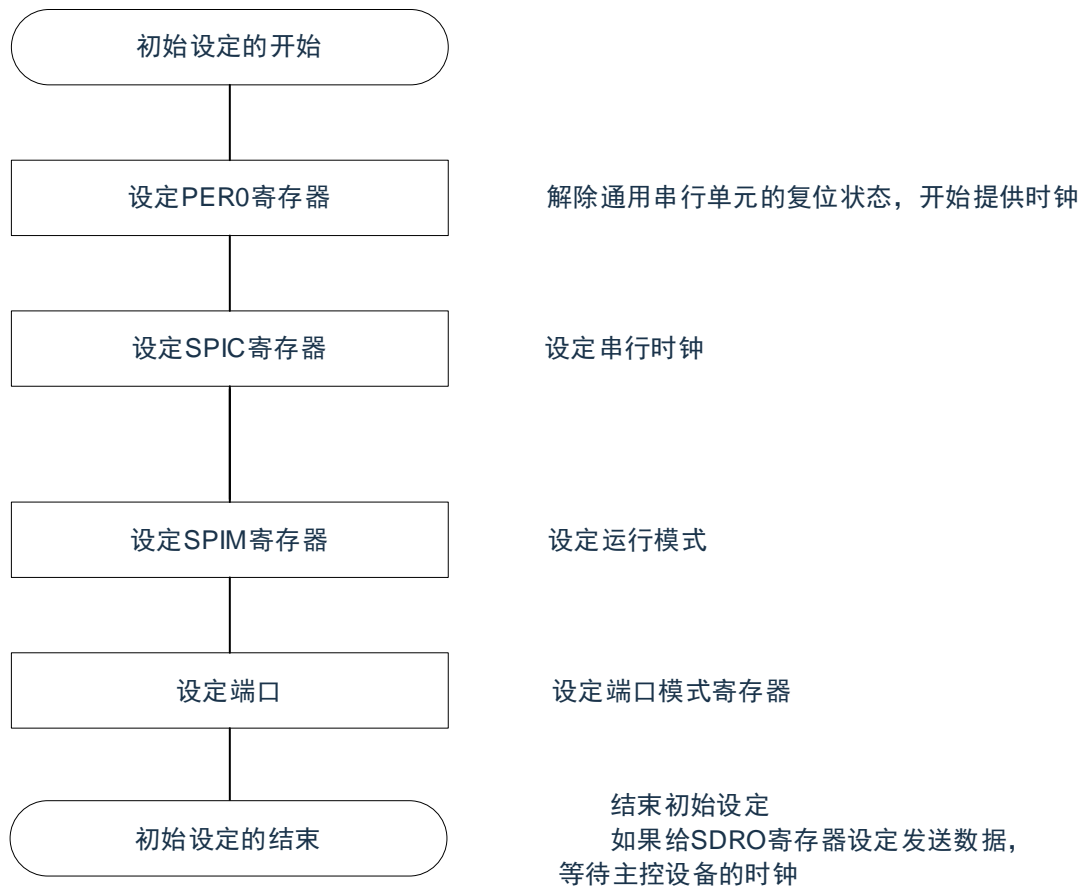
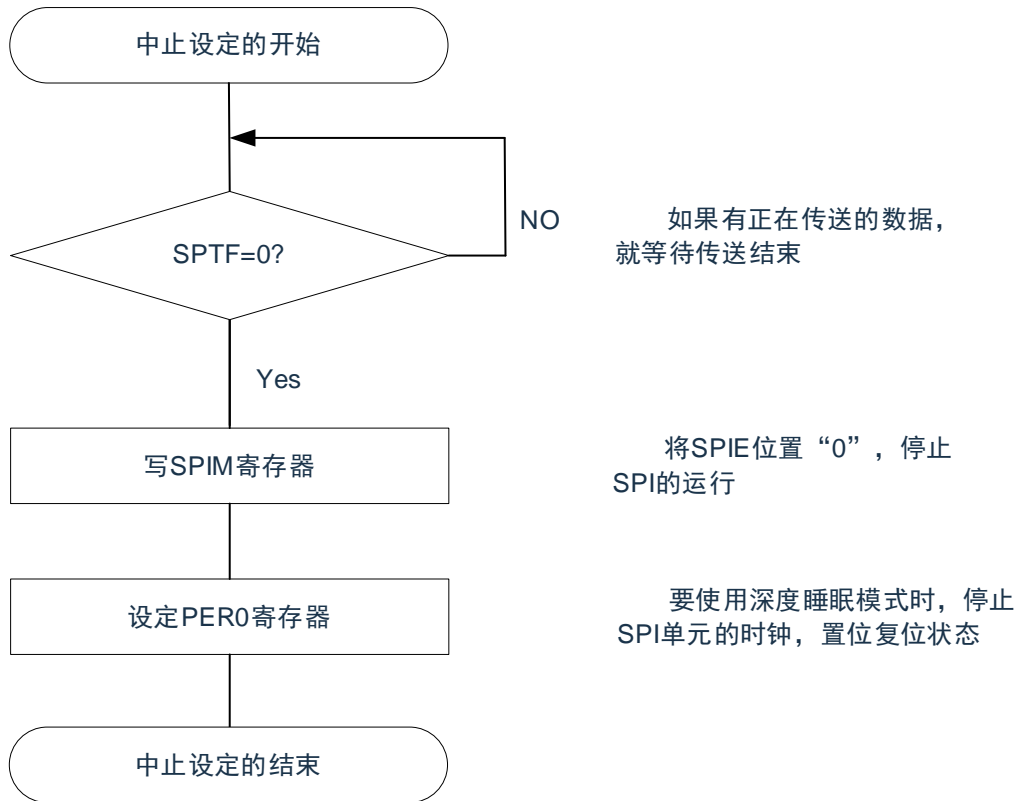


图 13-10: 从属发送/接收的中止步骤



(2) 处理流程

图 13-11: 发送/接收时序(单次发送模式)的时序图 (INTMD=0,DAP=0、CKPmn=0)

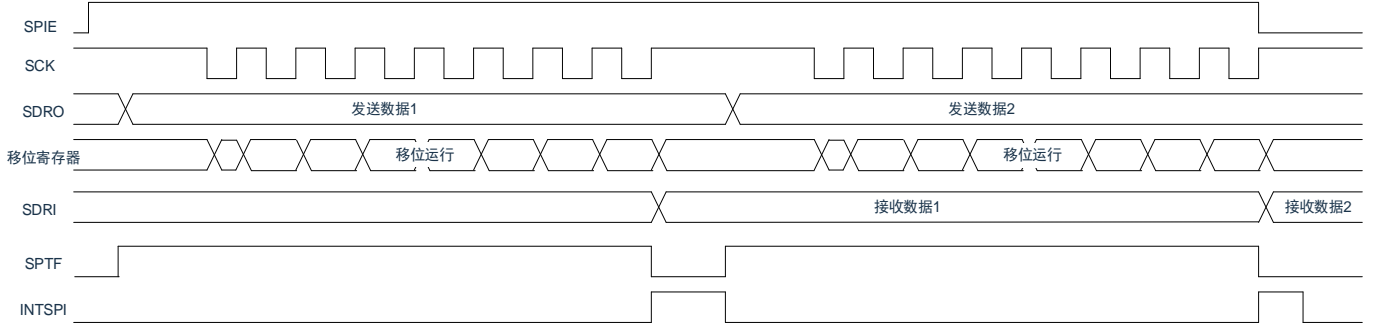
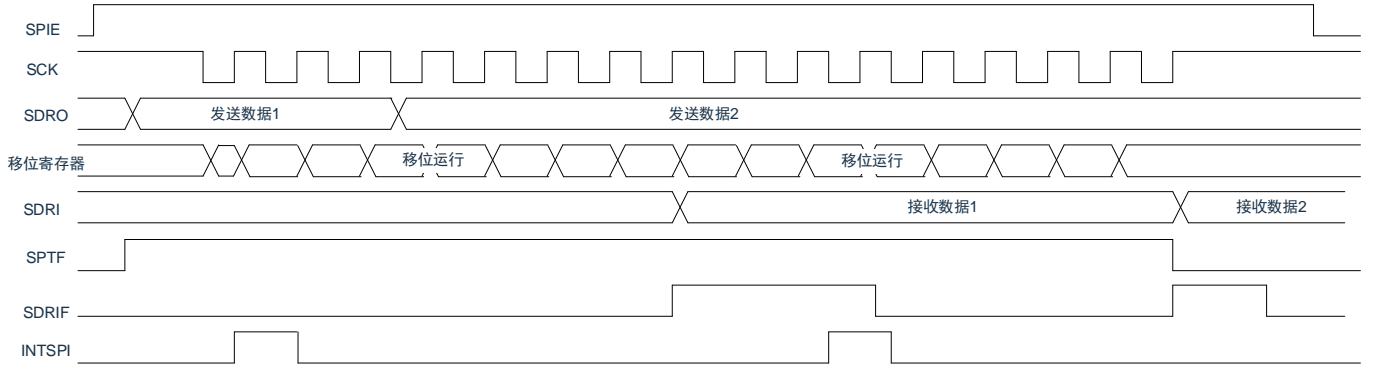


图 13-12: 发送/接收时序(连续发送模式)的时序图 (INTMD=1,DAP=0、CKPmn=0)





### 13.4.4 从属的接收

如果串行时钟选择寄存器（SPIC）的 CKS2-0 位选择从机模式，串行操作模式寄存器（SPIM）的位 6（TRMD）为 0，则进入从机接收模式。当从接收缓冲寄存器（SDRI）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 13-13: 从属接收的初始设置步骤

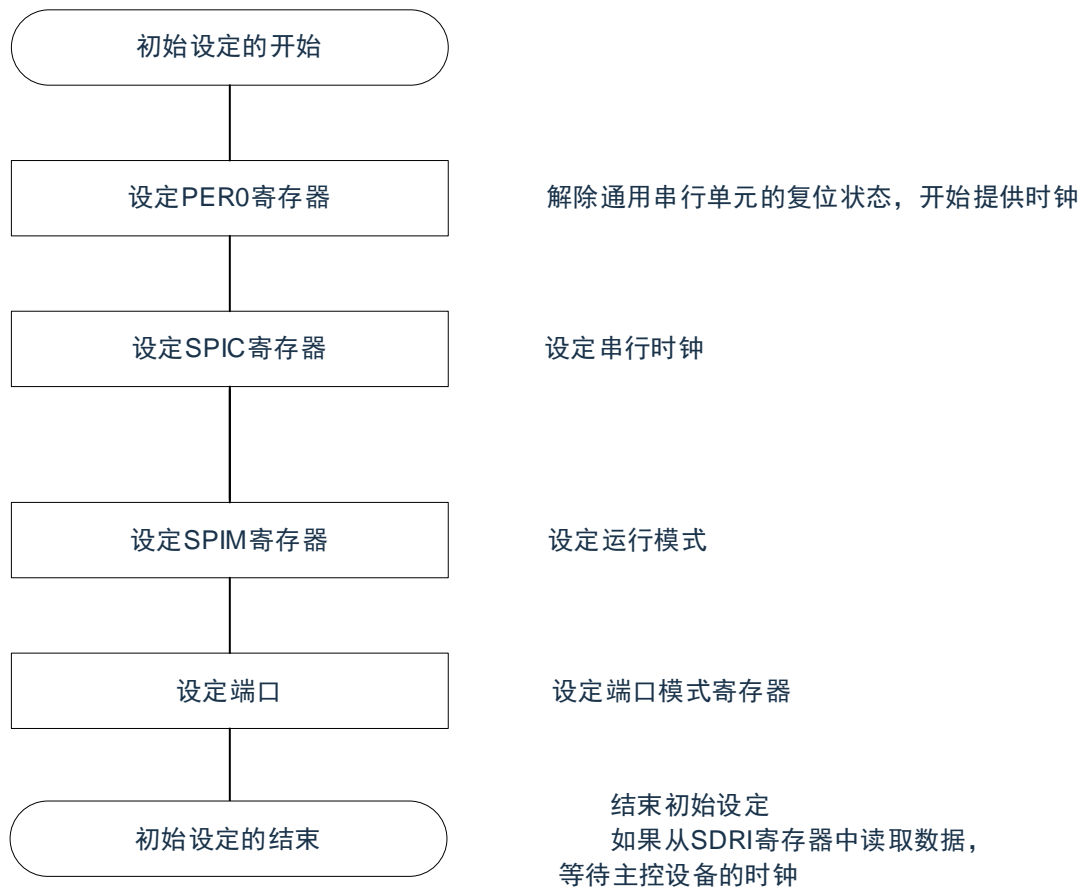
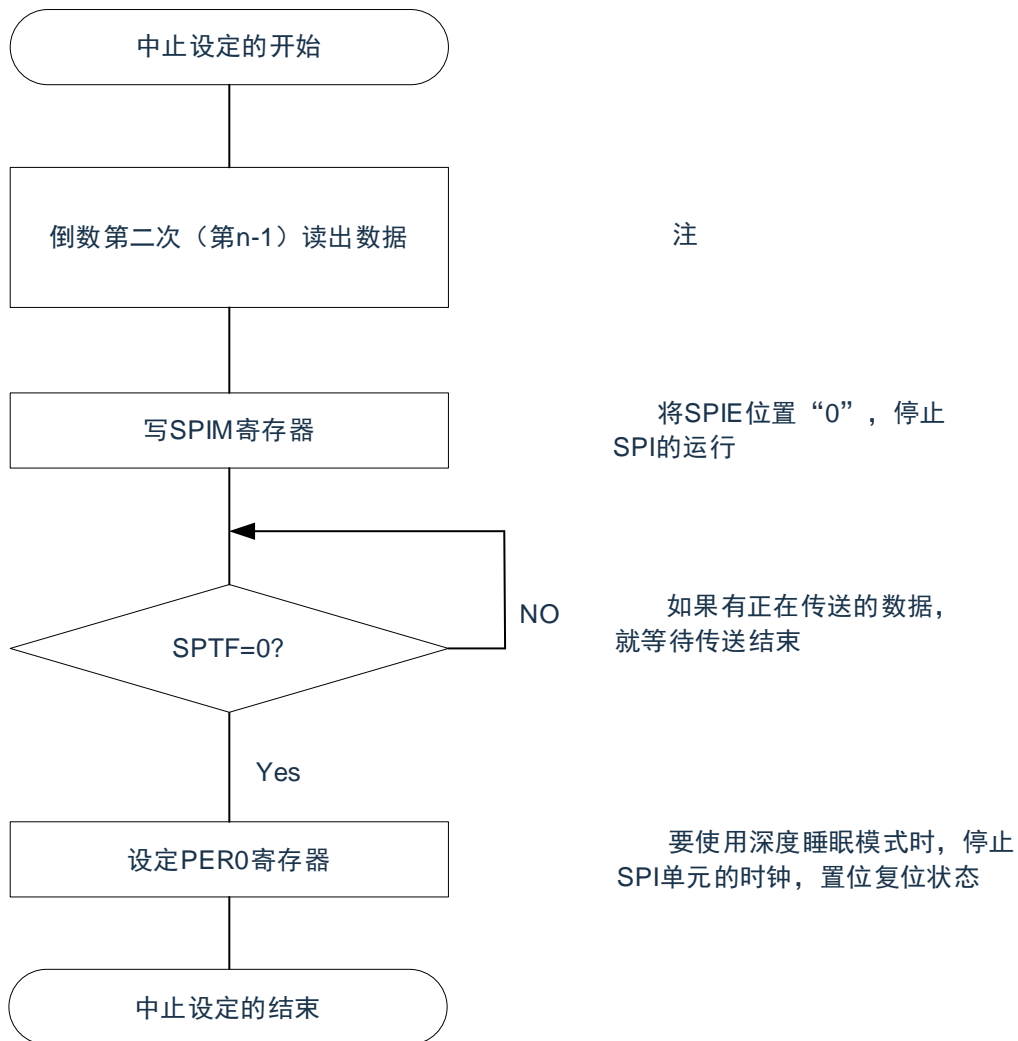


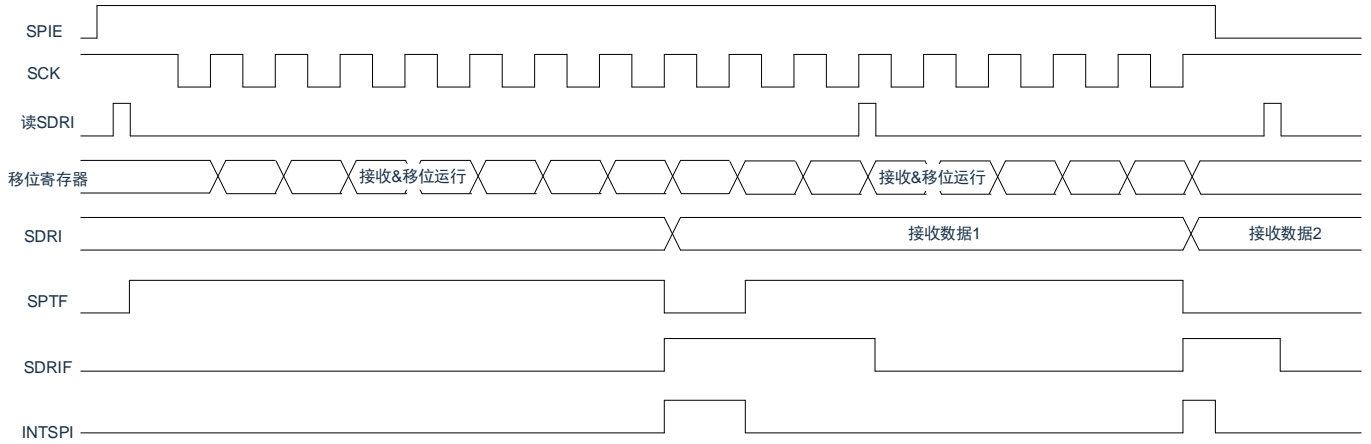
图 13-14：从属接收的中止步骤



注：只接收模式下，SPI传输由读取SDRI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SDRI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SPIE。SPI的传输将在最后一个数据传输完成后中止。

(2) 处理流程

图 13-15: 接收时序的时序图 (DAP=0、CKPmn=0)



## 第14章 串行接口IICA

### 14.1 串行接口IICA的功能

串行接口 IICA 有以下 3 种模式。

(1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

(2) 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。

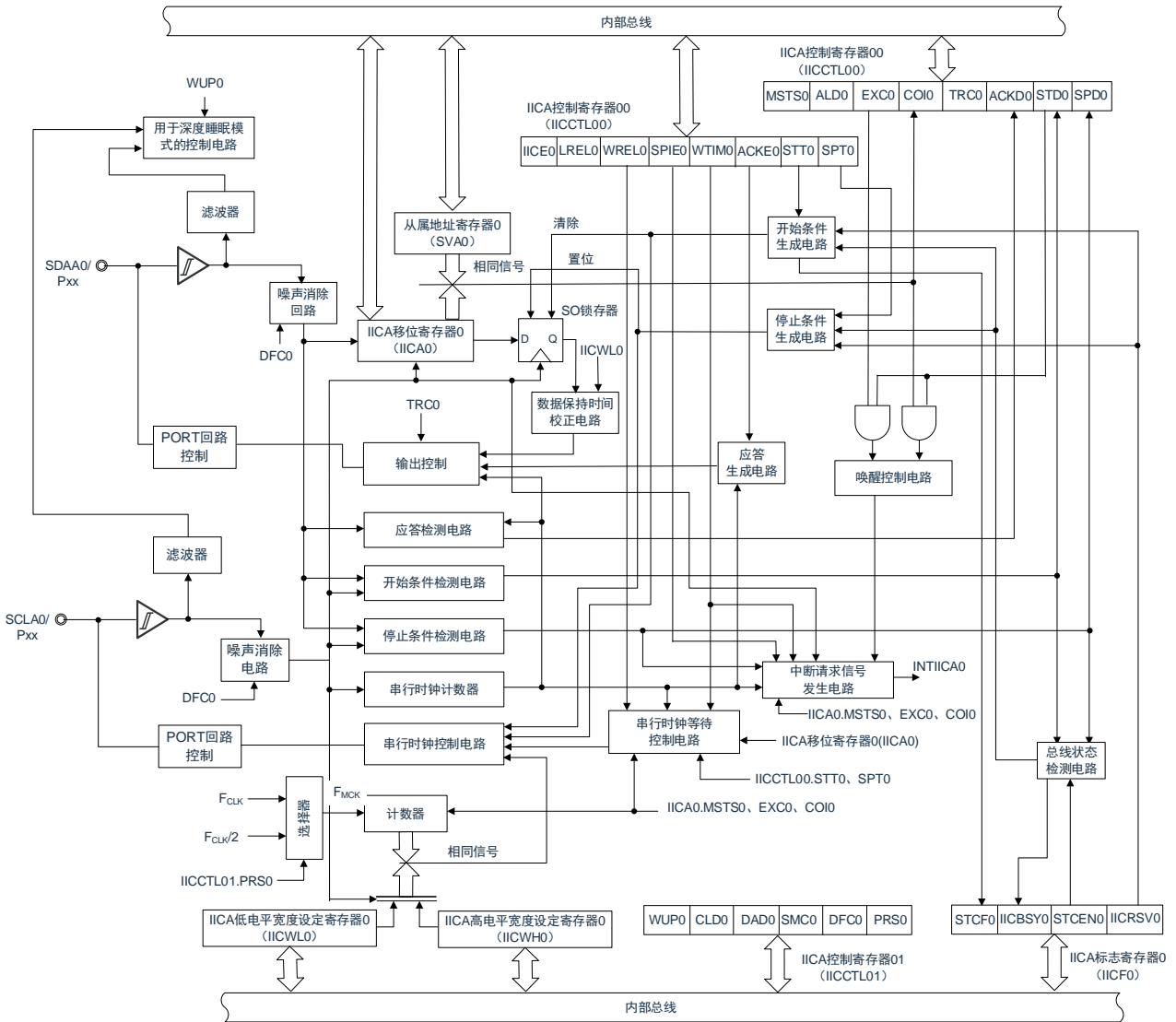
(3) 唤醒模式

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 位进行设定。

串行接口 IICA 的框图如图 14-1 所示。

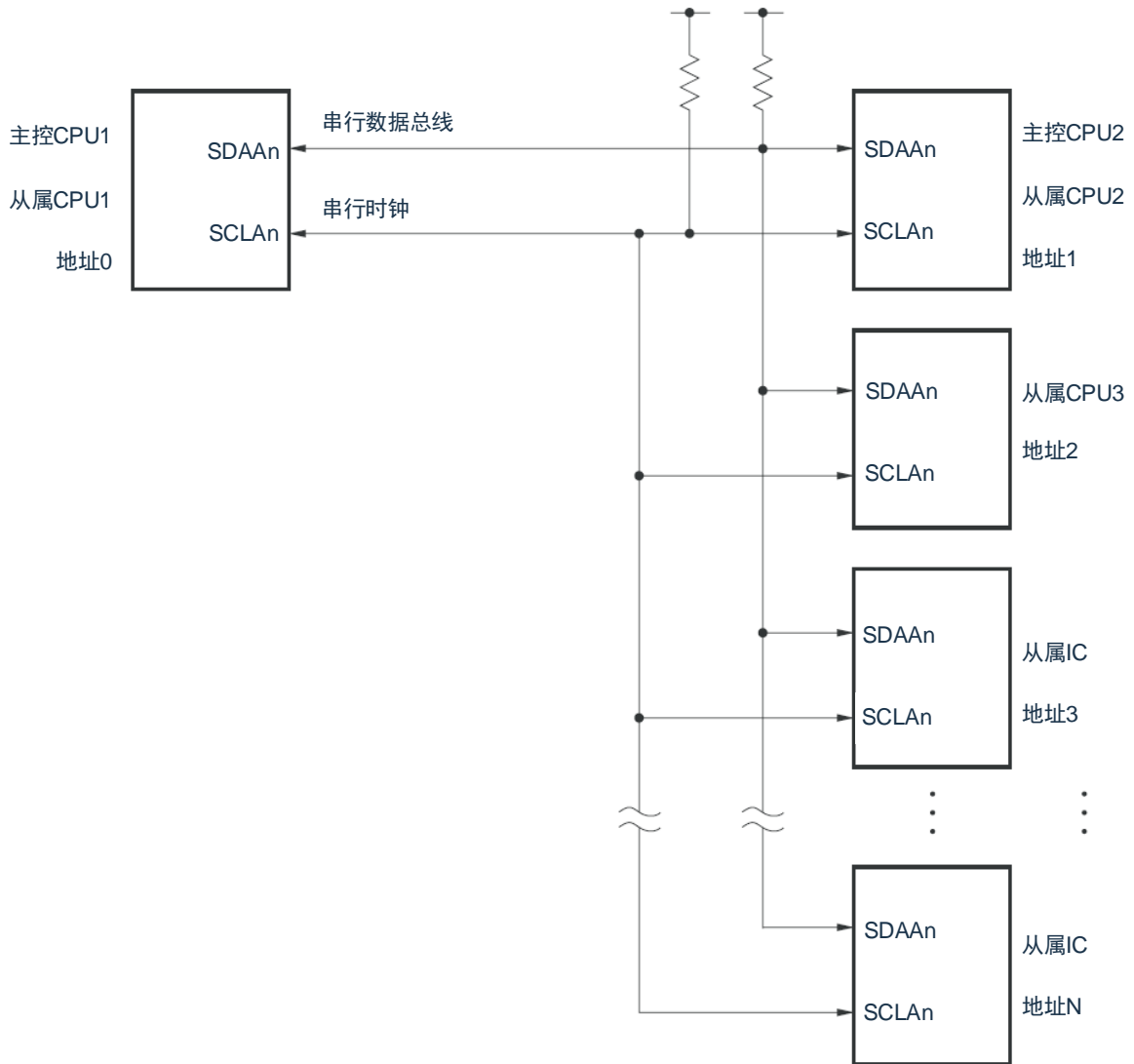
备注：n=0

图 14-1: 串行接口 IICA 的框图



串行总线的结构例子如图 14-2 所示。

图 14-2: I<sup>2</sup>C 总线的串行总线结构例子



备注: n=0



(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”。

表 14-3: 从属地址寄存器 n (SVAn) 的格式

地址: 符号	0x40041A34 7	复位后: 6	00H 5	4	3	2	R/W	1	0
SVAn	A6	A5	A4	A3	A2	A1		A0	0注

注: bit0 固定为“0”。

(3) SO 锁存器

SO 锁存器保持 SDAAn 引脚的输出电平。

(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)。

(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。

(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:

1. WTIMn 位: IICA 控制寄存器 n0 (IICCTLn0) 的 bit3
2. SPIEn 位: IICA 控制寄存器 n0 (IICCTLn0) 的 bit4

(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。

(8) 串行时钟等待控制电路

此电路控制等待时序。

(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。

(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。

(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”。

(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。



### (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
2. SPTn 位：IICA 控制寄存器 n0 (IICCTLn0) 的 bit0
3. IICRSVn 位：IICA 标志寄存器 n (IICFn) 的 bit0
4. IICBSYn 位：IICA 标志寄存器 n (IICFn) 的 bit6
5. STCFn 位：IICA 标志寄存器 n (IICFn) 的 bit7
6. STCENn 位：IICA 标志寄存器 n (IICFn) 的 bit1
7. n=0

## 14.3 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)

注：n=0

### 14.3.1 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 IICAn 时，必须将 bit4 (IICAEN) 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”。

表 14-4：外围允许寄存器 0 (PER0) 的格式

地址：0x40020420	复位后：00H							R/W
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	0	ADCEN	IICAEN	SCI1EN	SCI0EN	TM41EN	TM40EN

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

注意：要设定串行接口 IICA 时，必须先在 IICAEN 位为“1”的状态下设定以下的寄存器。当 IICAEN 位为“0”时，串行接口 IICA 的控制寄存器的值为初始值，忽视写操作（端口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)：

- (1) IICA 控制寄存器 n0 (IICCTLn0)
- (2) IICA 标志寄存器 n (IICFn)
- (3) IICA 状态寄存器 n (IICSn)
- (4) IICA 控制寄存器 n1 (IICCTLn1)
- (5) IICA 低电平宽度设定寄存器 n (IICWLn)
- (6) IICA 高电平宽度设定寄存器 n (IICWHn)

备注：n=0

### 14.3.2 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

表 14-5: IICA 控制寄存器 n0 (IICCTLn0) 的格式(1/4)

地址:	0x40041A30	复位后:	00H	R/W				
符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I <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。

注 2: 在 IICEn 位为“0”的状态下, 此位的信号无效。

注 3: LRELn 位和 WRELn 位的读取值总是“0”。

注意: 如果在 SCLAn 线为高电平、SDAAn 线为低电平并且数字滤波器为 ON (IICCTLn1 寄存器的 DFCn=1) 时允许 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

表 14-5: 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)
通过指令清除。 当复位时		通过指令置位。

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)
通过指令清除。 当复位时		通过指令置位。

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)
通过指令清除。 当复位时		通过指令置位。

注 1: 在 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。

注 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

表 14-5: 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： 允许通信预约功能的情况（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除STTn位并且将STTn清除标志（STCFn）置“1”，不生成开始条件。等待状态（主控设备）： 在解除等待后生成重新开始条件。
有关置位时序的注意事项： 主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。 禁止与停止条件的触发（SPTn）同时置“1”。 在将STTn位置“1”后，禁止在满足清除条件前再次将此位“1”。	
清除条件（STTn=0）	置位条件（STTn=1）
在禁止通信预约的状态下将STTn位置“1”。 在仲裁失败时 主控设备生成开始条件。 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位为“0”（停止运行）时 当复位时	通过指令置位。

注 1：在 IICEn 位为“0”的状态下，此位的信号无效。

注 2：STTn 位的读取值总是“0”。

备注：

1. 如果在设定数据后读 bit1（STTn），此位就变为“0”。
2. IICRSVn: IICA 标志寄存器 n (IICFn) 的 bit0
3. STCFn: IICA 标志寄存器 n (IICFn) 的 bit7
4. n=0

表 14-5: IICA 控制寄存器 n0 (IICCTLn0) 的格式(4/4)

SPTn <sup>注1</sup>	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）。
<p>有关置位时序的注意事项：                      主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。                      主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第9个时钟后的等待期间将此位置“1”。                      禁止与开始条件的触发（STTn）同时置“1”。                      只有在主控设备的情况下才能将SPTn位置“1”。                      在WTIMn位为“0”时，必须注意：如果在输出8个时钟后的等待期间将SPTn位置“1”，就在解除等待后的第9个时钟的高电平期间生成停止条件。必须在输出8个时钟后的等待期间将WTIMn位从“0”置为“1”并且在输出第9个时钟后的等待期间将SPTn位置“1”。                      在将SPTn位置“1”后，禁止在满足清除条件前再次将此位置“1”。</p>	
清除条件（SPTn=0）	置位条件（SPTn=1）
当仲裁失败时 在检测到停止条件后自动清除。 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位为“0”（停止运行）时 当复位时	通过指令置位。

注 1：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

### 14.3.3 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：

1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1
2. WUPn: IICA 控制寄存器 n1 (IICCTLn1) 的 bit7

表 14-6: IICA 状态寄存器 n (IICSn) 的格式(1/3)

地址:	0x40041B51	复位后:	00H						
符号	7	6	5	4	3	2	1	0	R
IICSn	MSTS <sub>n</sub>	ALD <sub>n</sub>	EXC <sub>n</sub>	COI <sub>n</sub>	TRC <sub>n</sub>	ACKD <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)
当检测到停止条件时 当ALD <sub>n</sub> 位为“1”（仲裁失败）时 因LREL <sub>n</sub> 位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时	当生成开始条件时

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)
在读IICS <sub>n</sub> 寄存器后自动清除 <sup>注1</sup> 。 当IICEn位从“1”变为“0”（停止运行）时 当复位时	当仲裁失败时

注 1：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 ALDn 位时，必须在读其他位前先读 ALDn 位的数据。

备注：

1. LREL<sub>n</sub>: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6
2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7
3. n=0



表 14-6: 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)	
当检测到开始条件时 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时	
置位条件 (EXCn=1)	
当接收的地址数据的高4位为“0000”或者“1111”时 （在第8个时钟的上升沿置位）	

COIn	地址匹配的检测
0	地址不同。
1	地址相同。
清除条件 (COIn=0)	
当检测到开始条件时 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时	
置位条件 (COIn=1)	
当接收地址和本地站地址（从属地址寄存器n (SVAn) ) 相同时（在第8个时钟的上升沿置位）	

TRCn	发送/接收的状态检测
0	处于接收状态（发送状态除外）。将SDAAn线置为高阻抗。
1	处于发送状态。设定为能将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)	
<主控设备和从属设备> 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 因WRELn位为“1”（解除等待）而进行的清除 <sup>注1</sup> 当ALDn位从“0”变为“1”（仲裁失败）时 当复位时 不参加通信的情况 (MSTSn、EXCn、COIn=0)	
置位条件 (TRCn=1)	
<主控设备> 当生成开始条件时 当第1字节（地址传送）的LSB（传送方向指示位） 输出“0”（主控发送）时 <从属设备> 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	
<主控设备> 当第1字节的LSB（传送方向指示位）输出“1”时	
<从属设备> 当检测到开始条件时 当第1字节的LSB（传送方向指示位）输入“0”时	

注 1: 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
2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7
3. n=0

表 14-6: 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)
当检测到停止条件时 当下一个字节的第1个时钟上升时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时		在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时

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)
当检测到停止条件时 在地址传送后的下一个字节的第1个时钟上升时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时		当检测到开始条件时

SPDn	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件，主控设备结束通信并且已释放总线。	
清除条件 (SPDn=0)		置位条件 (SPDn=1)
在将此位置位后，在检测到开始条件后的地址传送字节的第1个时钟上升时 当WUPn位从“1”变为“0”时 当IICEn位从“1”变为“0”（停止运行）时 当复位时		当检测到停止条件时

**备注：**

1. LRELn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6
2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7
3. n=0

### 14.3.4 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”。

表 14-7: IICA 标志寄存器 n (IICFn) 的格式

地址:	0x40041B52	复位后:	00H					RW <sup>注</sup>	
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)
因STTn位为“1”而进行的清除 当IICEn位为“0” (停止运行) 时 当复位时	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

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)
当检测到停止条件时 当IICEn位为“0” (停止运行) 时 当复位时	当检测到开始条件时 STCENn位为“0”时的IICEn位的置位

STCENn	初始开始允许触发
0	在允许运行 (IICEn=1) 后，通过检测停止条件来允许生成开始条件。
1	在允许运行 (IICEn=1) 后，不检测停止条件而允许生成开始条件。
清除条件 (STCENn=0)	置位条件 (STCENn=1)
通过指令清除。 当检测到开始条件时 当复位时	通过指令置位。

IICRSVn	通信预约功能禁止位
0	允许通信预约。
1	禁止通信预约。
清除条件 (IICRSVn=0)	置位条件 (IICRSVn=1)
通过指令清除。 当复位时	通过指令置位。

注: bit6 和 bit7 是只读位。

**注意:**

1. 只有在停止运行 (IICEn=0) 时才能写 STCENn 位。
2. 如果 STCENn 位为“1”，就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0)，因此为了避免在发行第 1 个开始条件 (STTn=1) 时破坏其他通信，需要确认没有正在通信的第三方。
3. 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

**备注:**

1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1
2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

### 14.3.5 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”。

表 14-8: IICA 控制寄存器 n1 (IICCTLn1) 的格式(1/2)

地址:	0x40041A31	复位后:	00H						
符号	7	6	5	4	3	2	1	0	
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn	

WUPn	地址匹配唤醒的控制
0	在深度睡眠模式中，停止地址匹配唤醒功能的运行。
1	在深度睡眠模式中，允许地址匹配唤醒功能的运行。

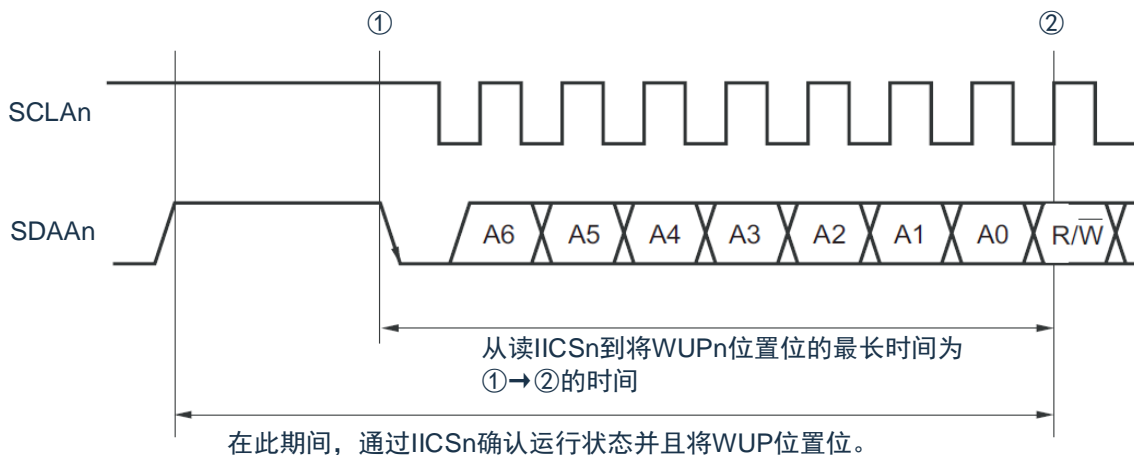
要通过将WUPn位置“1”来转移到深度睡眠模式时，必须在将WUPn位置“1”后至少经过3个F<sub>MCK</sub>时钟，然后执行深度睡眠指令（参照“图14-13将WUPn位置“1”时的流程”）。在地址匹配或者接收到扩展码后，必须将WUPn位清“0”。能通过将WUPn位清“0”来参加后续的通信（需要在将WUPn位清“0”后解除等待以及写发送数据）。

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。

注 2: 在以下所示的期间，需要确认 IICA 状态寄存器 n (IICSn) 的状态并且将其置位。



备注: n=0

表 14-8: 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)	
当SCLAn引脚为低电平时 当IICEn位为“0” (停止运行) 时 当复位时	
置位条件 (CLDn=1)	
当SCLAn引脚为高电平时	

DADn	SDAAn引脚的电平检测 (只在IICEn位为“1”时有效)
0	检测到SDAAn引脚为低电平。
1	检测到SDAAn引脚为高电平。
清除条件 (DADn=0)	
当SDAAn引脚为低电平时 当IICEn位为“0” (停止运行) 时 当复位时	
置位条件 (DADn=1)	
当SDAAn引脚为高电平时	

SMCn	运行模式的切换
0	在标准模式中运行 (最大传送速率: 100kbps)。
1	在快速模式 (最大传送速率: 400kbps) 或者增强型快速模式 (最大传送速率: 1Mbps) 中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器OFF
1	数字滤波器ON
必须在快速模式或者增强型快速模式中使用数字滤波器。数字滤波器用于消除噪声。 无论是将DFCn位置“1”还是清“0”，传送时钟都不变。	

PRSn	运行时钟 (F <sub>MCK</sub> ) 的控制
0	选择F <sub>CLK</sub> (4MHz ≤ F <sub>CLK</sub> ≤ 16MHz)。
1	选择F <sub>CLK</sub> /2 (16MHz < F <sub>CLK</sub> )。

**注意:**

1. IICA 运行时钟 (F<sub>MCK</sub>) 的最大工作频率为 16MHz(Max.)。只有在 F<sub>CLK</sub> 超过 16MHz 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

### 14.3.6 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“14.4.2 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。

表 14-9 IICA 低电平宽度设定寄存器 n (IICWLn) 的格式

地址:	0x40041A32		复位后:		FFH		R/W	
符号	7	6	5	4	3	2	1	0
IICWLn	1	1	1	1	1	1	1	1

### 14.3.7 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”。

表 14-10 IICA 高电平宽度设定寄存器 n (IICWHn) 的格式

地址:	0x40041A33		复位后:		FFH		R/W	
符号	7	6	5	4	3	2	1	0
IICWHn	1	1	1	1	1	1	1	1

注 1: 有关主控方传送时钟的设定方法, 请参照 14.4.2(1); 有关从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法, 请参照 14.4.2(2)。

注 2: n=0

### 14.3.8 控制IICA引脚端口功能的寄存器

本产品可将 IICA 的引脚功能复用到除 RESETB 以外的任意端口。

通过设置端口复用功能配置寄存器 (SCLA0PCFG 和 SDAA0PCFG) 可将 SCALn 引脚和 SDAAn 引脚分别配置到两个端口。

将这两个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”。

这两个端口被配置为 IICA 引脚的复用功能后, 端口的 N 沟道漏极开路输出 (VDD/EVDD 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。

详细的设置方法参见“第 2 章引脚功能”

## 14.4 I<sup>2</sup>C总线模式的功能

### 14.4.1 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

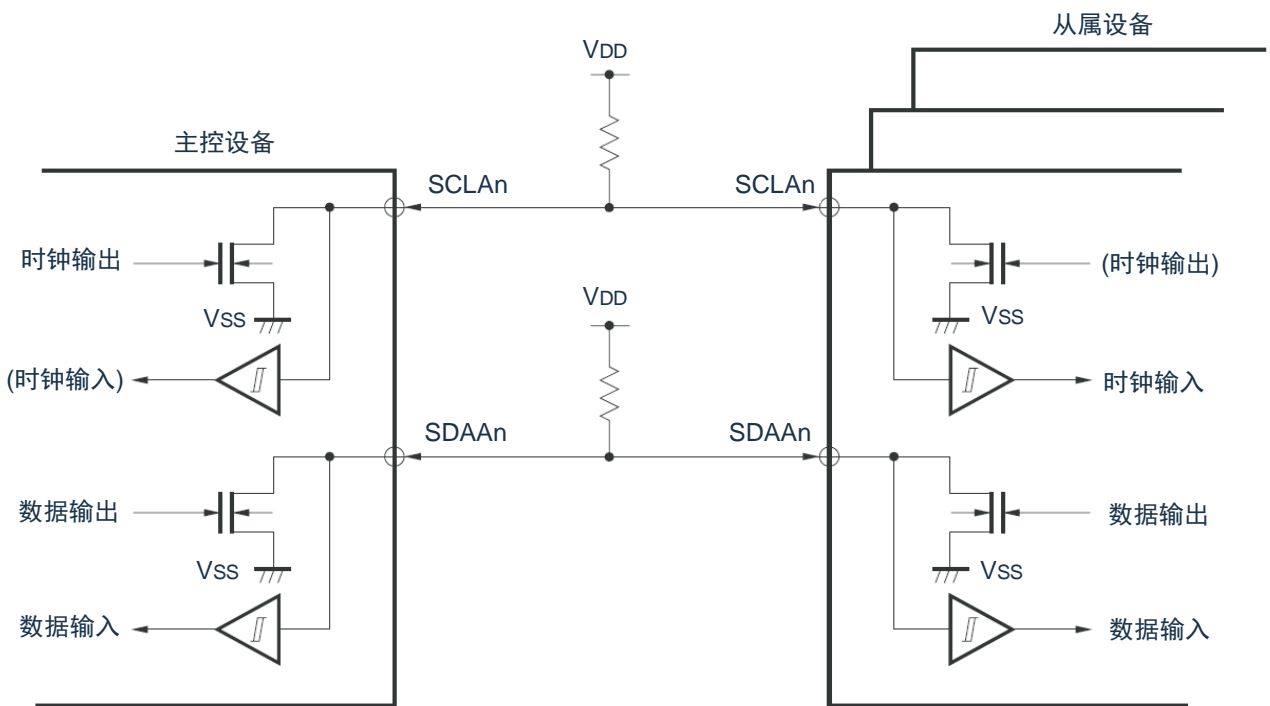
主控设备和从属设备的输出都为 N 沟道漏极开路输出，输入都为施密特输入。

(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 14-3：引脚结构图



备注：n=0



## 14.4.2 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

$$\text{传送时钟} = \frac{F_{MCK}}{IICWL + IICWH + F_{MCK} (T_R + T_F)}$$

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）

#### ① 快速模式

$$IICWLn = \frac{0.52}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left( \frac{0.48}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

#### ② 标准模式

$$IICWLn = \frac{0.47}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left( \frac{0.53}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

#### ③ 增强型快速模式

$$IICWLn = \frac{0.50}{\text{传送时钟}} \times F_{MCK}$$

$$IICWHn = \left( \frac{0.50}{\text{传送时钟}} - T_R - T_F \right) \times F_{MCK}$$

### (2) 从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法

（全部设定值的小数部分都舍入）

#### ① 快速模式

$$IICWLn = 1.3\mu s \times F_{MCK}$$

$$IICWHn = (1.2\mu s - T_R - T_F) \times F_{MCK}$$

#### ② 标准模式

$$IICWLn = 4.7\mu s \times F_{MCK}$$

$$IICWHn = (5.3\mu s - T_R - T_F) \times F_{MCK}$$

#### ③ 增强型快速模式

$$IICWLn = 0.50\mu s \times F_{MCK}$$

$$IICWHn = (0.50\mu s - T_R - T_F) \times F_{MCK}$$

注意：

- IICA 运行时钟（ $F_{MCK}$ ）的最大工作频率为 16MHz(Max.)。只有在  $F_{CLK}$  超过 16MHz 时才必须将 IICA 控制寄存器 n1（IICCTLn1）的 bit0（PRSn）置“1”。
- 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： $F_{CLK}=1\text{MHz}(\text{Min.})$
- 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。

备注:

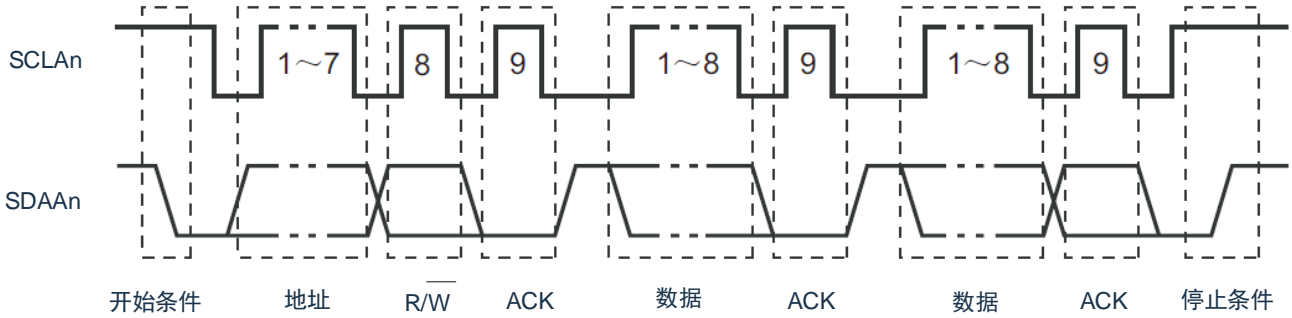
1. IICWLn: IICA 低电平宽度设定寄存器 n; IICWHn : IICA 高电平宽度设定寄存器 n  
T<sub>F</sub>: SDAAn 信号和 SCLAn 信号的下降时间; T<sub>R</sub>: SDAAn 信号和 SCLAn 信号的上升时间  
F<sub>МCK</sub>: IICA 运行时钟频率;
2. n=0

## 14.5 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 14-4: 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。

## 14.5.1 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。



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”。

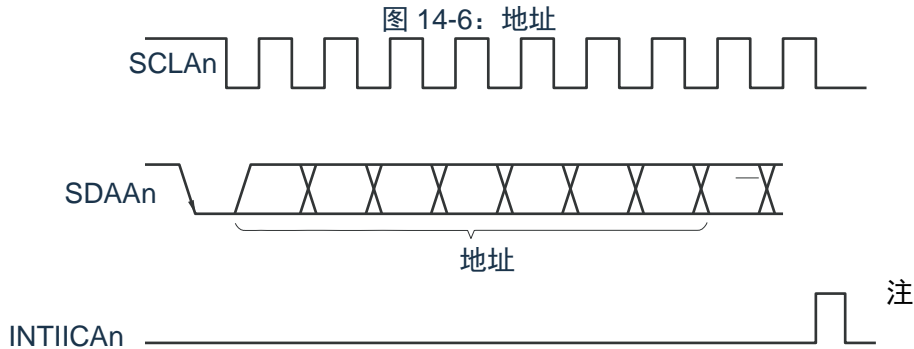
备注：n=0

### 14.5.2 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 7 位数据生成开始条件或者停止条件前，与主控设备进行通信。



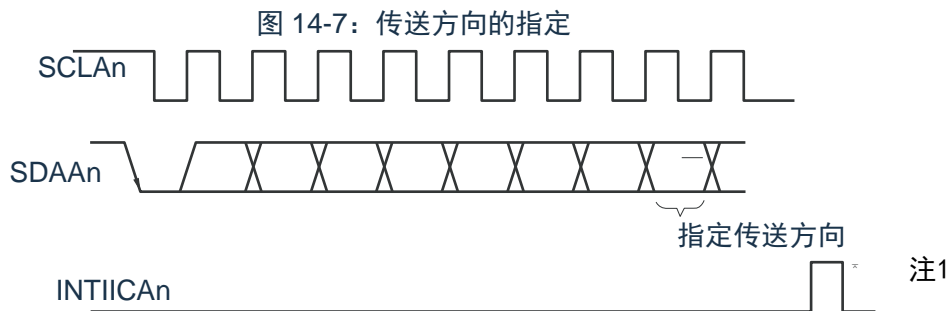
注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

如果将从属地址和“14.5.3 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。

### 14.5.3 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。

当此传送方向指定位为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定位为“1”时，表示主控设备从从属设备接收数据。



注 1：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

### 14.5.4 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。

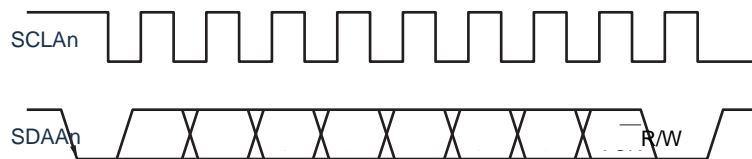
接收方在第 9 个时钟将 SDAAn 线置为低电平，生成应答 (正常接收)。

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。

在从属接收运行过程中 (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 14-8: 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：

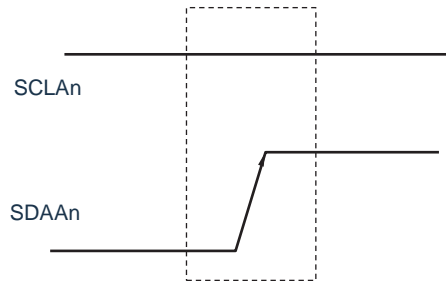
- (1) 当选择 8 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =0)：通过在解除等待前将 ACKEn 位置“1”，与 SCLAn 引脚的第 8 个时钟下降沿同步生成应答。
- (2) 当选择 9 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =1)：通过事先将 ACKEn 位置“1”，生成应答。

备注：n=0

## 14.5.5 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 14-9：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

### 14.5.6 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 14-10：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）

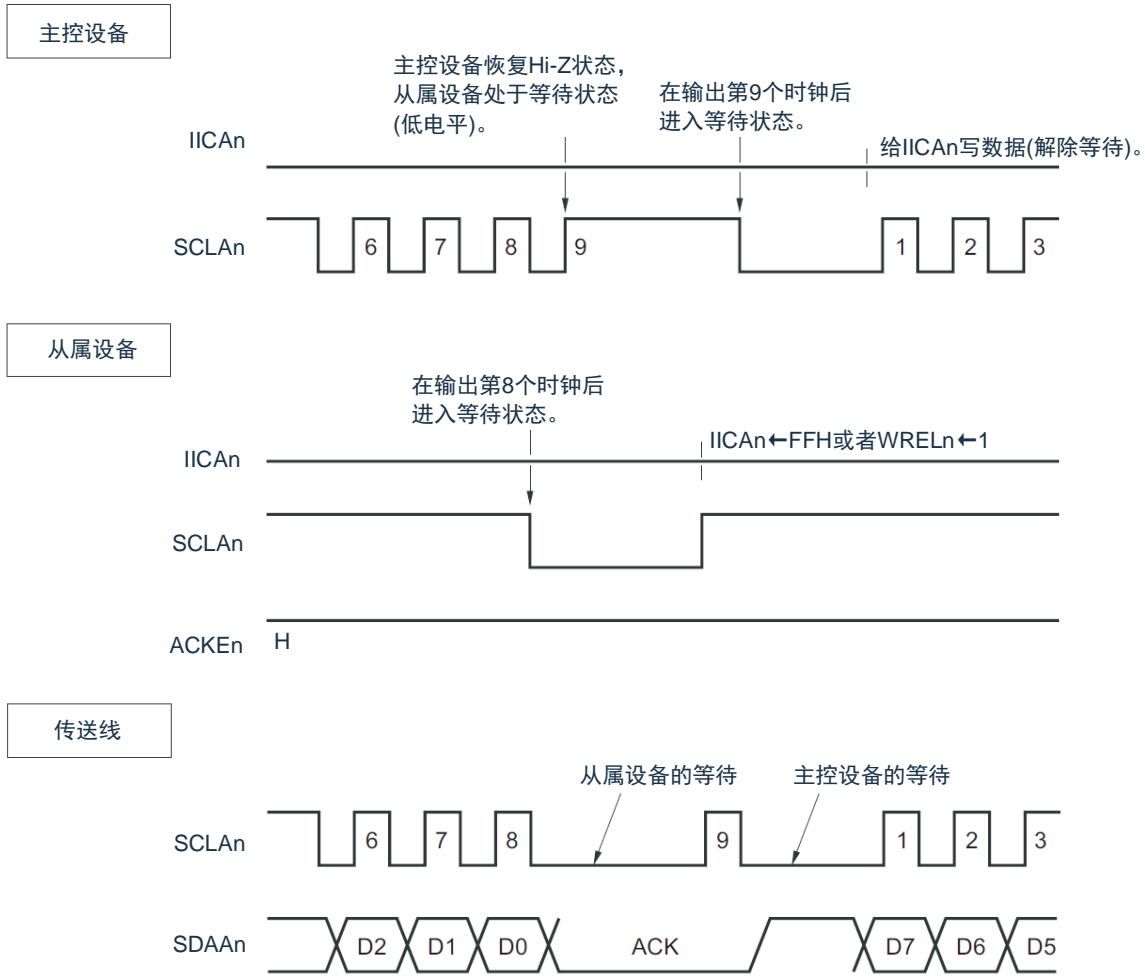
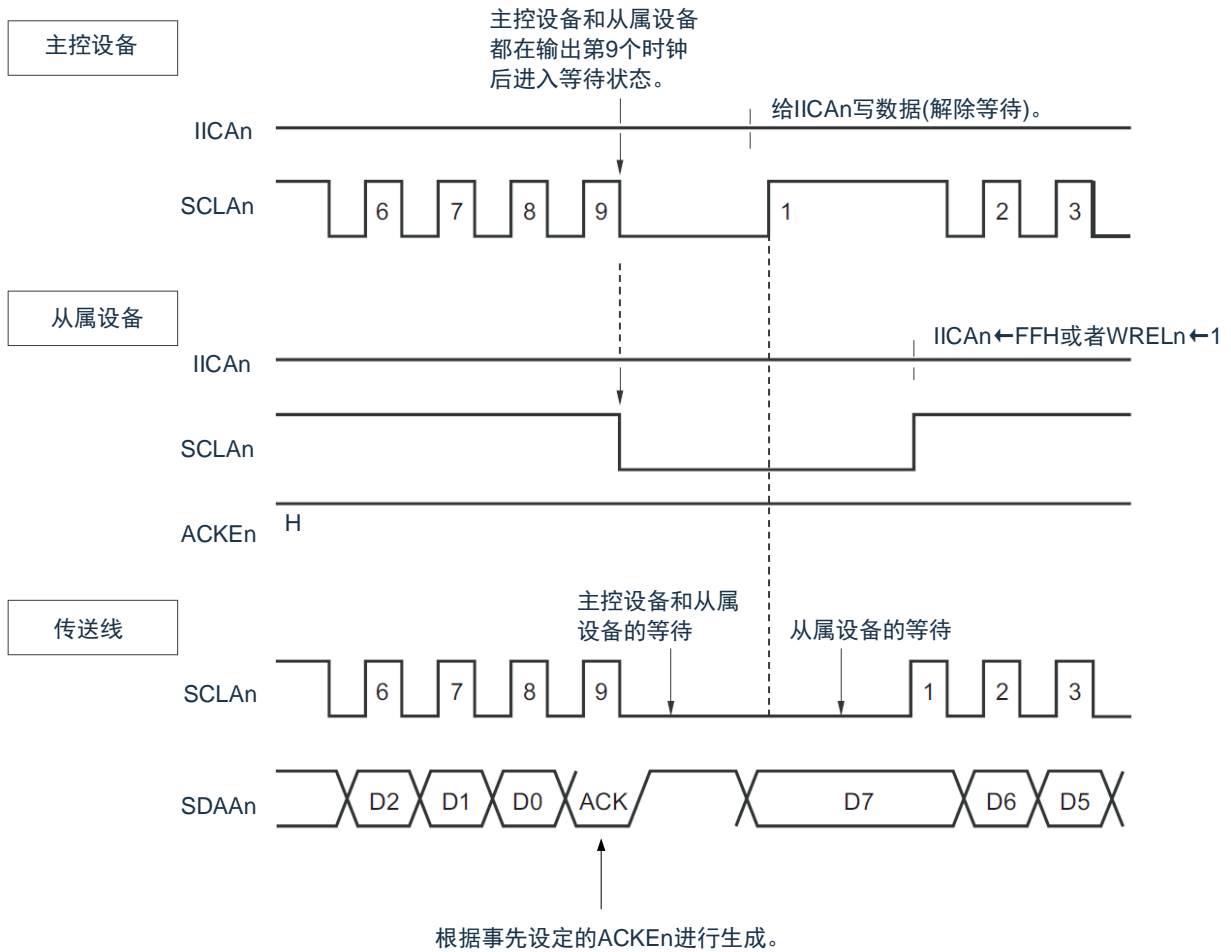




图 14-11：等待(2/2)

(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

## 14.5.7 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：

- (1) 给 IICA 移位寄存器 n (IICAn) 写数据。
- (2) 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- (3) 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件)<sup>注</sup>。
- (4) 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件)<sup>注</sup>。

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。

备注：

1. 如果在 WUPn 位为“1”时执行等待的解除处理，就不解除等待。
2. n=0

## 14.5.8 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 14-11 所示的时序产生 INTIICAn 并且进行等待控制。

表 14-11：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。

注 2：如果接收的地址和从属地址寄存器 n（SVAn）的内容不同并且未接收到扩展码，就不产生 INTIICAn 并且也不进入等待状态。

备注：表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

### (1) 地址的发送和接收

- ① 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。
- ② 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：

- ① 给 IICA 移位寄存器 n (IICAn) 写数据。
- ② 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- ③ 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件)<sup>注</sup>。
- ④ 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件)<sup>注</sup>。

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)。

## 14.5.9 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。

## 14.5.10 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

## 14.5.11 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)。
  - ① 高 4 位数据相同：EXCn=1
  - ② 7 位数据相同：COIn=1

备注：

1. EXCn: IICA 状态寄存器 n (IICSn) 的 bit5
  2. 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 14-12: 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

### 14.5.12 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。

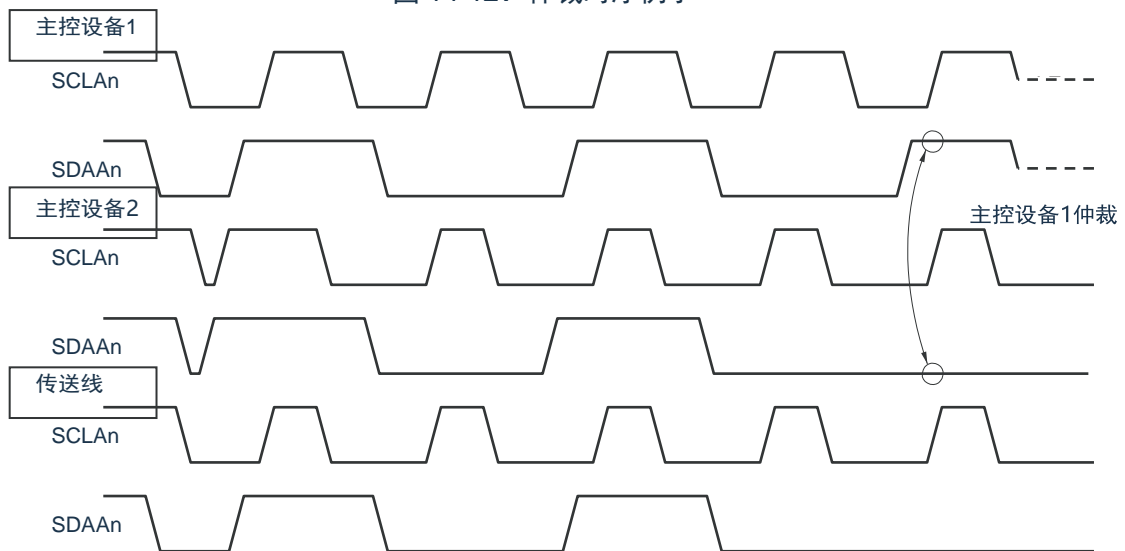
在发生下一次中断请求时（例如：在第 8 或者第 9 个时钟检测到停止条件），用软件通过 ALDn 位为“1”来检测仲裁的失败。

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：

1. STDn: IICA 状态寄存器 n（IICSn）的 bit1
2. STTn: IICA 控制寄存器 n0（IICCTLn0）的 bit1

图 14-12: 仲裁时序例子



备注：n=0

表 14-13: 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。	在生成停止条件时 (SPIEn=1) <sup>注2</sup>
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。

注 2: 当有可能发生仲裁时, 必须在主控运行时将 SPIEn 位置“1”。

备注:

1. SPIEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit4
2. n=0

### 14.5.13 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 14-13 所示，通过地址匹配将 WUPn 位置“0”时的流程如图 14-14 所示。

图 14-13: 将 WUPn 位置“1”时的流程

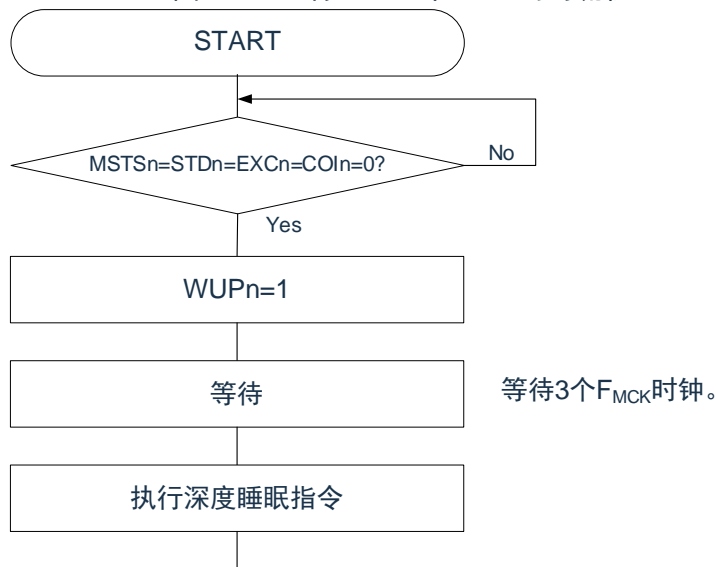
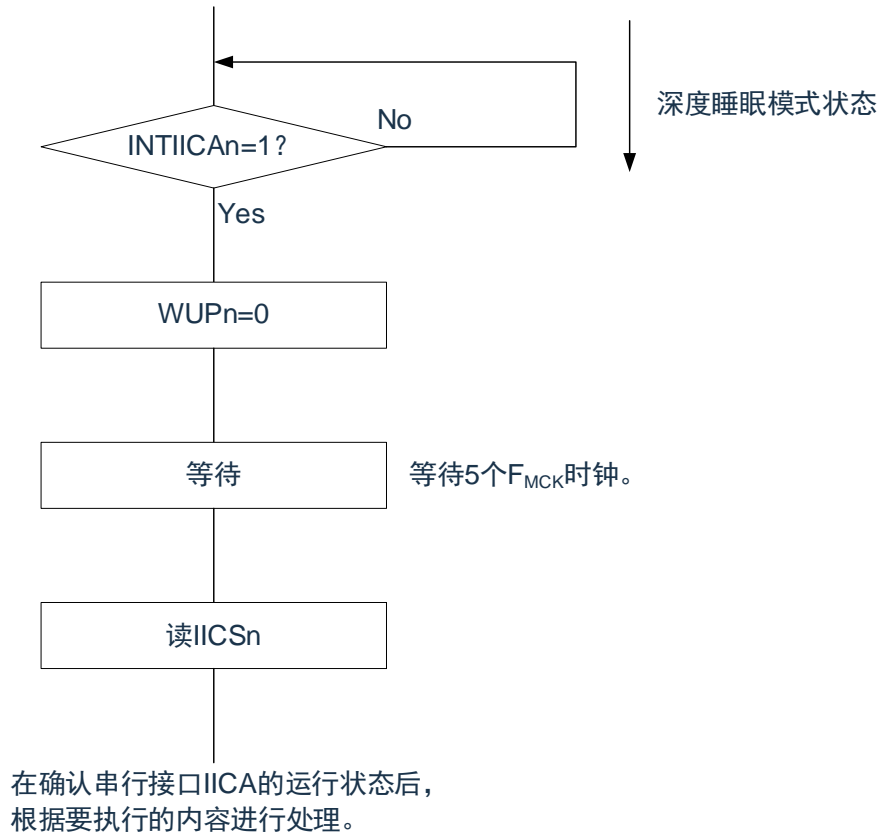




图 14-14：通过地址匹配将 WUPn 位置“0”时的流程（包括接收扩展码）



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：

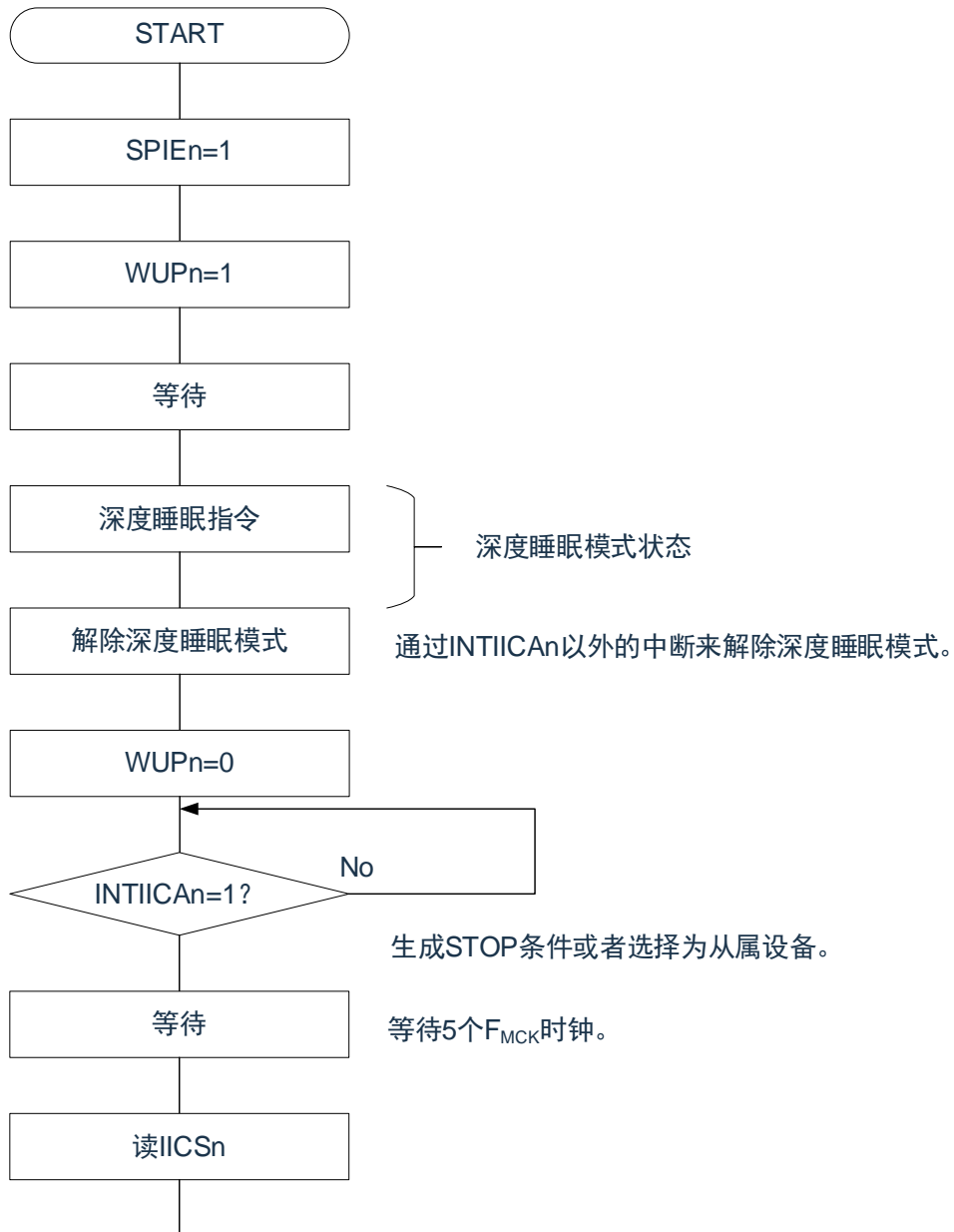
- (1) 下一次 IIC 通信为主控设备运行的情况：图 14-15 的流程
- (2) 下一次 IIC 通信为从属设备运行的情况：

通过 INTIICAn 中断返回的情况：和图 14-14 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 14-15: 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

### 14.5.14 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:

- ① 在仲裁结果既不是主控设备也不是从属设备时
- ② 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。

- ① 总线处于释放状态时生成开始条件
- ② 总线未处于释放状态 (待机状态) 时通信预约

在将 STTn 位置“1”并且经过等待时间后, 通过 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 确认是否作为通信预约运行。

必须通过软件确保以下计算式计算的等待时间:

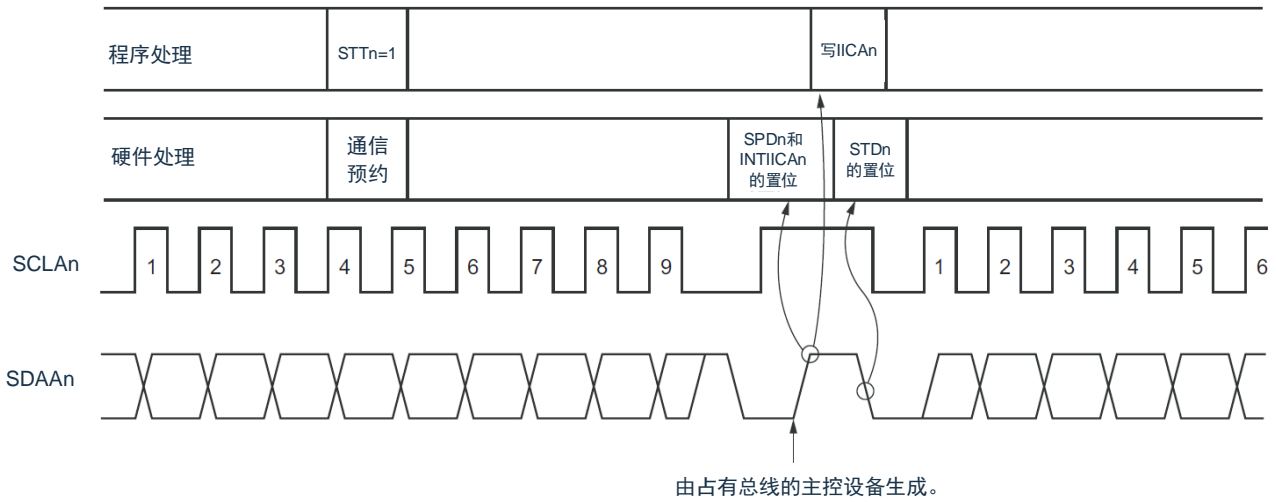
$$\text{从将 STTn 位置“1”到确认 MSTSn 标志为止的等待时间:} \\ (\text{IICWLn 的设定值} + \text{IICWHn 的设定值} + 4) / F_{\text{MCK}} + T_{\text{F}} \times 2$$

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<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

通信预约的时序如下图所示：

图 14-16：通信预约的时序



备注：IICAn：IICA 移位寄存器 n

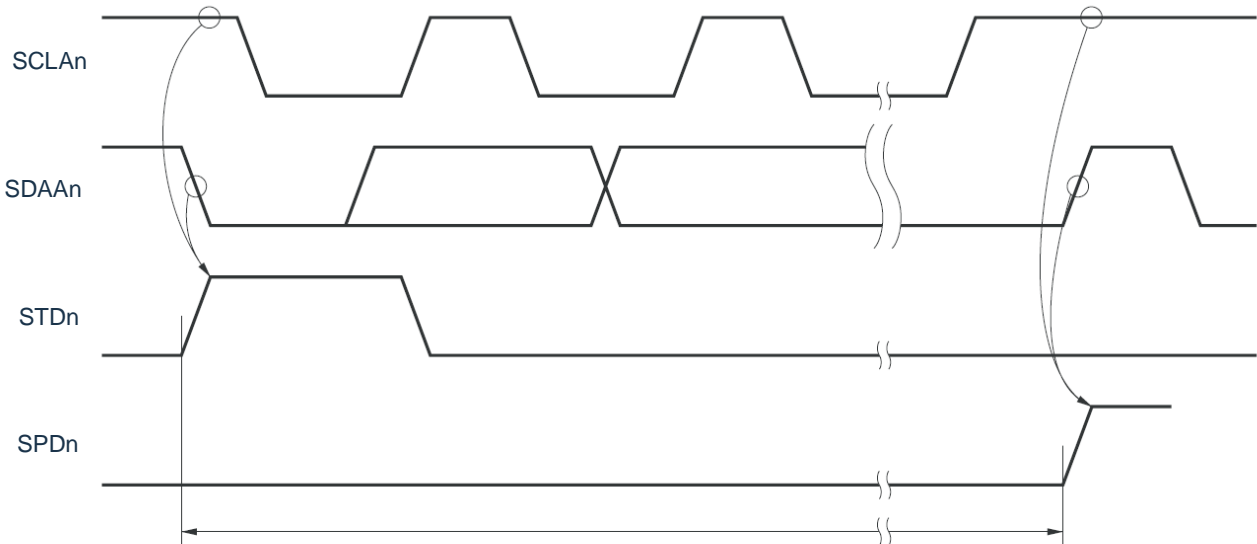
STTn：IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

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 14-17 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。

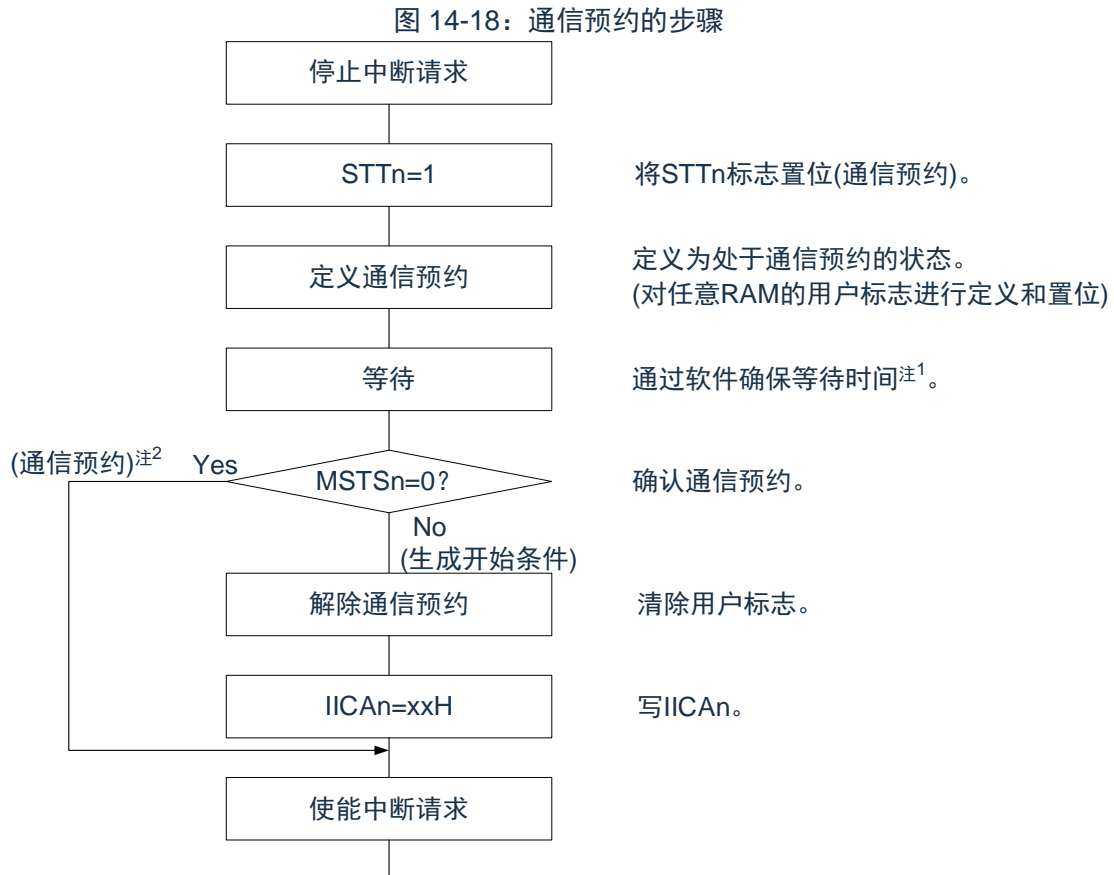
图 14-17：通信预约的接受时序



待机状态(在此期间，能通过将STTn位置“1”进行通信预约)

备注：n=0

通信预约的步骤如图 14-18 所示：



注 1：等待时间如下： $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) / F_{MCK} + T_F \times 2$

注 2：在通信预约运行时，通过停止条件中断请求来写 IICA 移位寄存器 n (IICAn)。

备注：

1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1  
 MSTSn: IICA 状态寄存器 n (IICSn) 的 bit7  
 IICAn: IICA 移位寄存器 n  
 IICWLn: IICA 低电平宽度设定寄存器 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  
 $F_{MCK}$ : IICA 运行时钟频率
2. n=0

(2) 禁止通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =1)

在总线通信过程中，如果不参加此通信的状态下将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”，就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下 2 种状态：

- ① 在仲裁结果既不是主控设备也不是从属设备时
- ② 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

## 14.5.15 其他注意事项

### 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”。

### (1) 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。

### (2) 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<sub>MCK</sub> 时钟内) 将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 强制将检测置为无效。

### (3) 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。

### (4) 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

## 14.5.16 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 I<sup>2</sup>C 总线的多主控系统中，只根据 I<sup>2</sup>C 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 I<sup>2</sup>C 总线从属设备的例子如下所示。

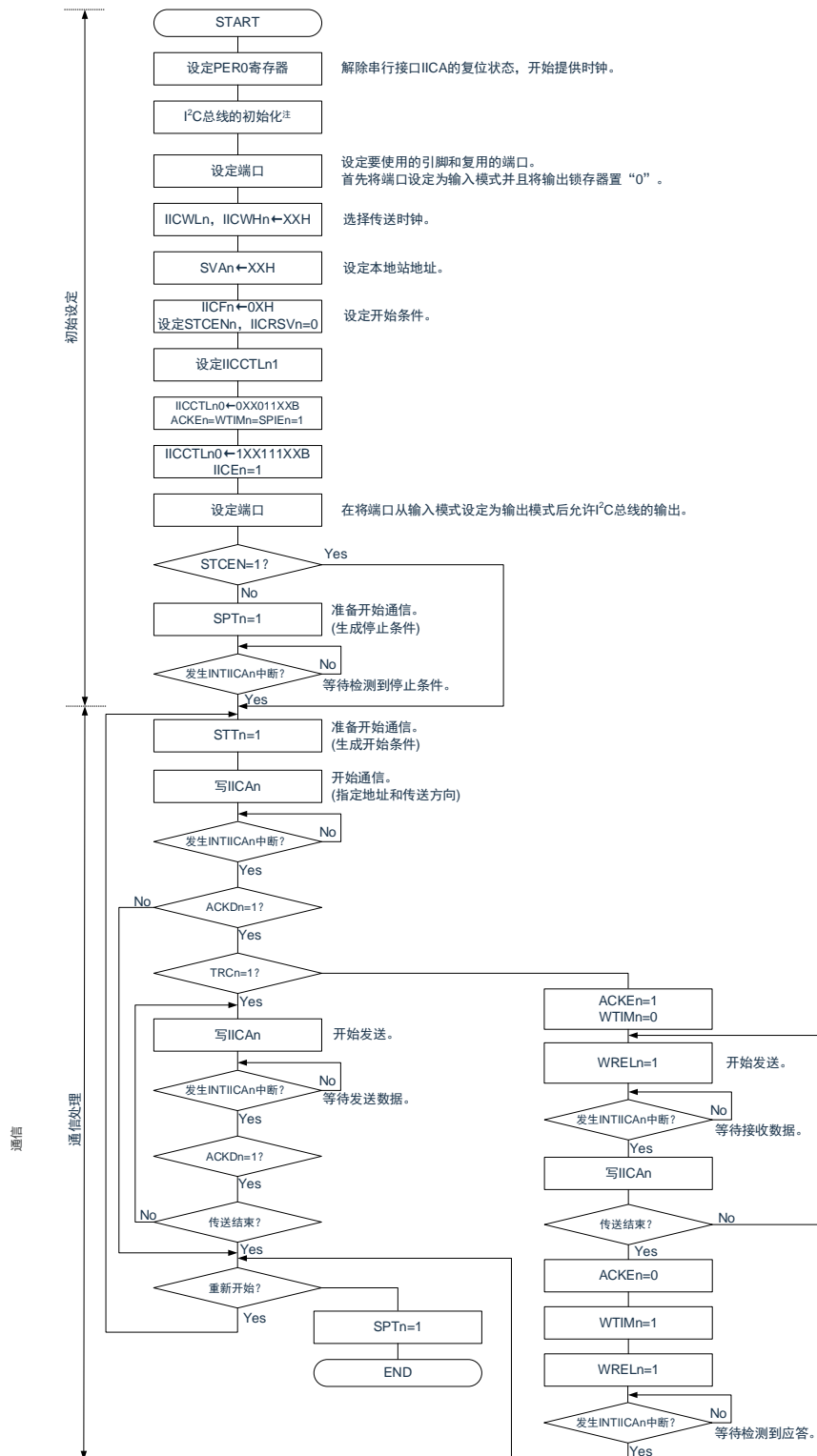
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

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 14-19: 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。

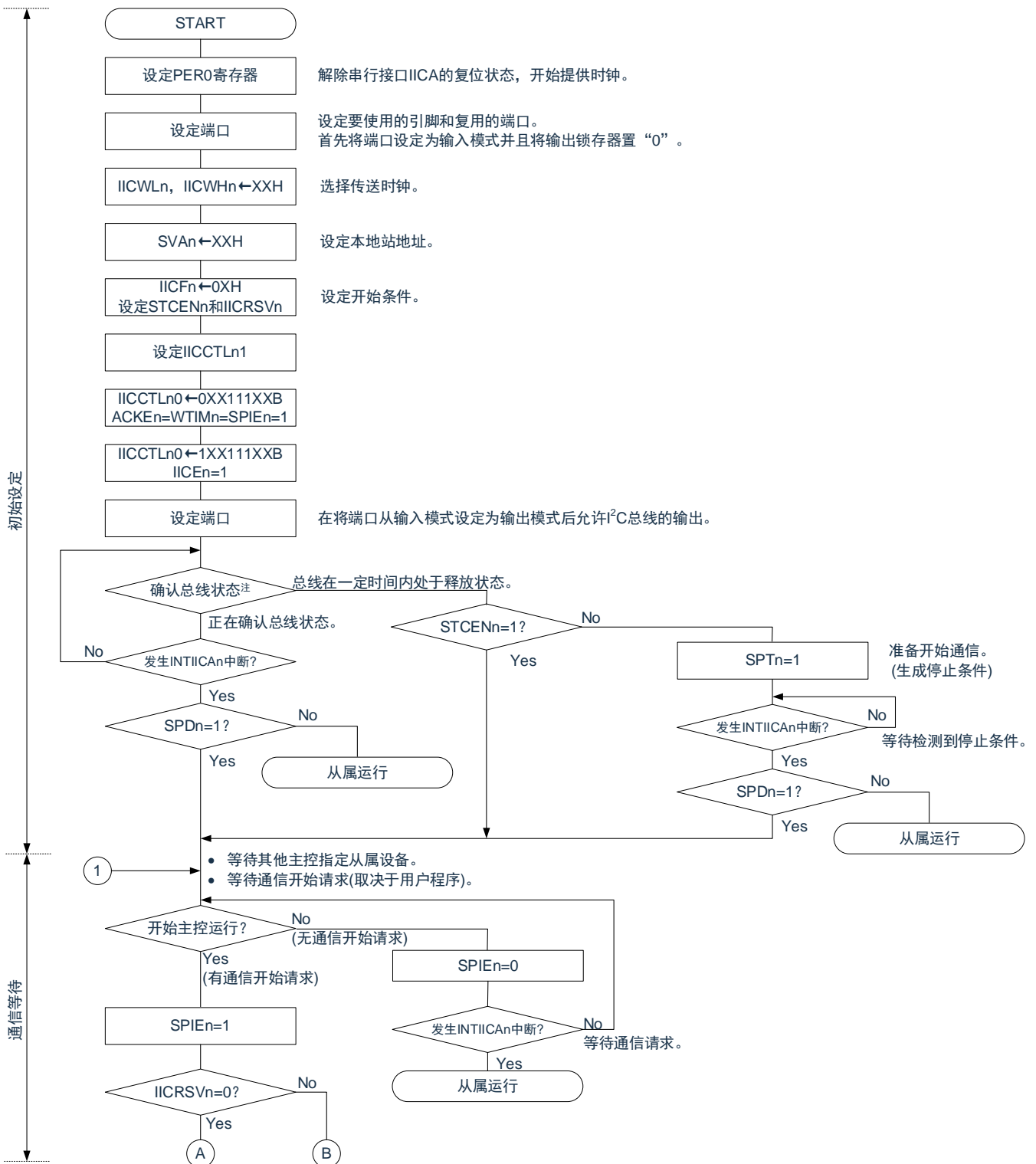
注意：发送和接收的格式必须符合通信中的产品的规格。

备注：n=0



(2) 多主控系统的主控运行

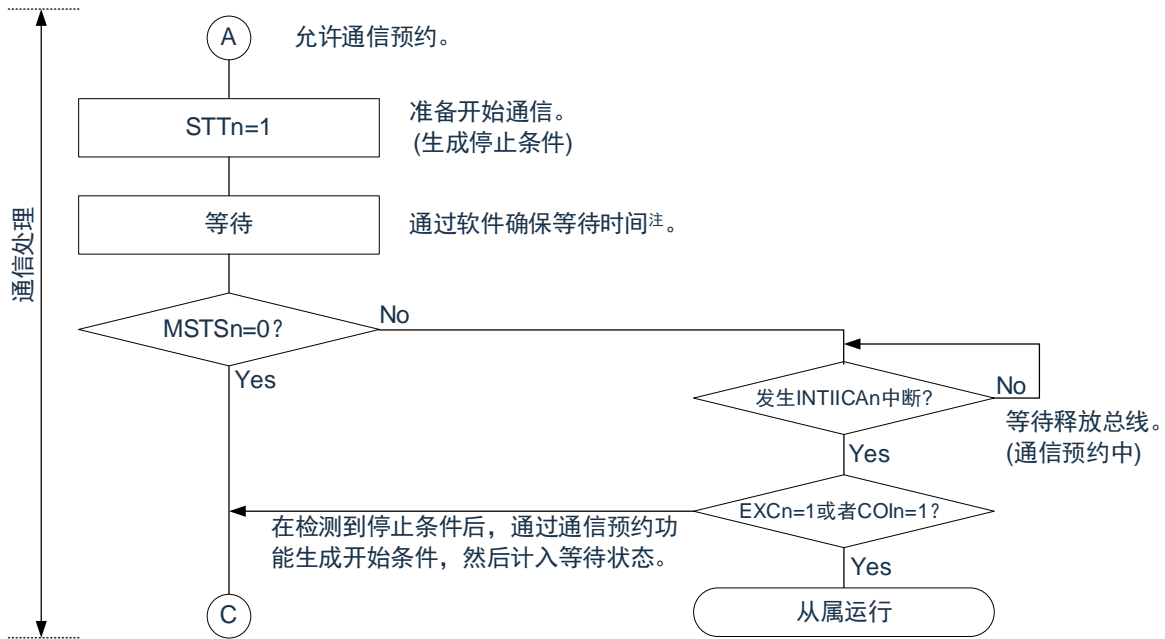
图 14-20: 多主控系统的主控运行(1/3)



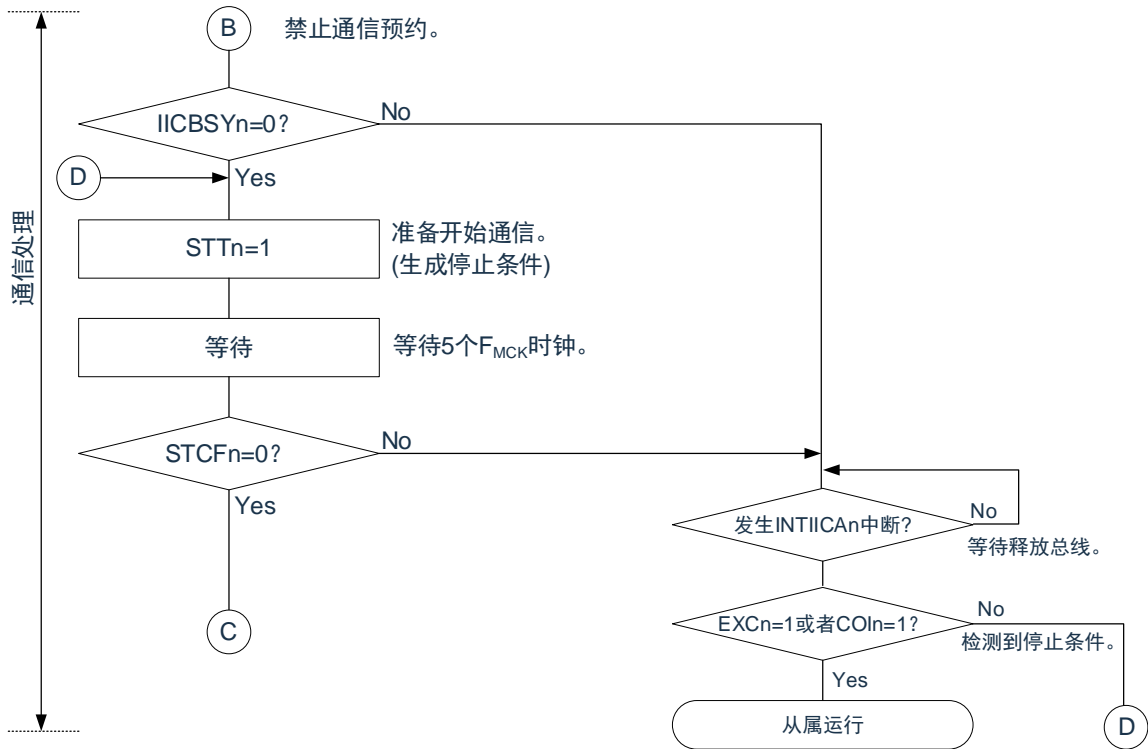
注: 必须确认总线在一定时间内(例如, 1帧)处于释放状态(CLDn位=1、DADn位=1)。当SDAAn引脚固定为低电平时, 必须根据通信中的产品的规格, 判断是否释放I<sup>2</sup>C总线(SCLAn引脚和SDAAn引脚为高电平)。

备注: n=0

图 14-20: 多主控系统的主控运行(2/3)



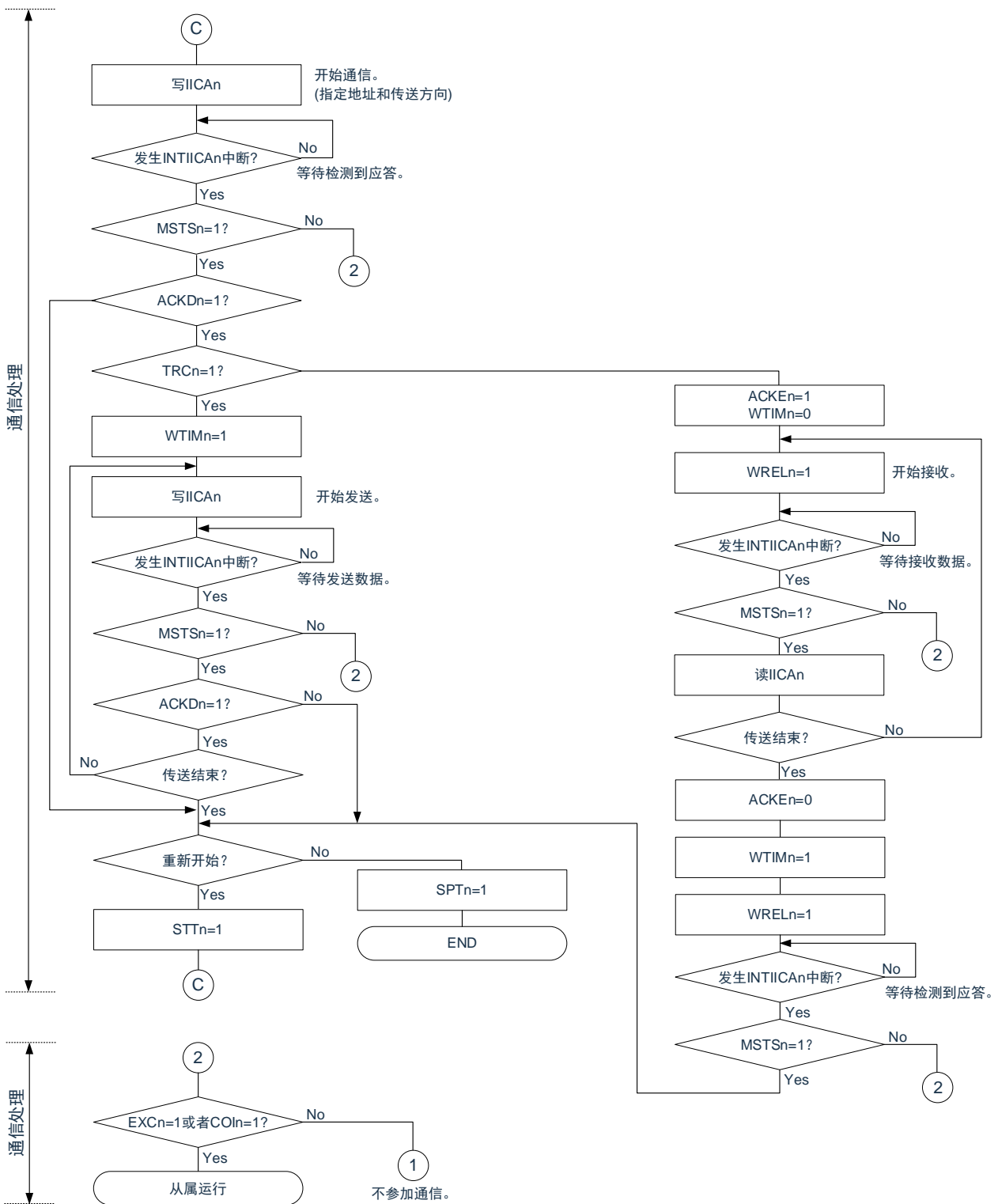
注 等待时间如下:  
 $(IICWLn的设定值+IICWHn的设定值+4)/F_{MCK}+T_F \times 2$



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<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 14-20: 多主控系统的主控运行(3/3)



注意:

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。

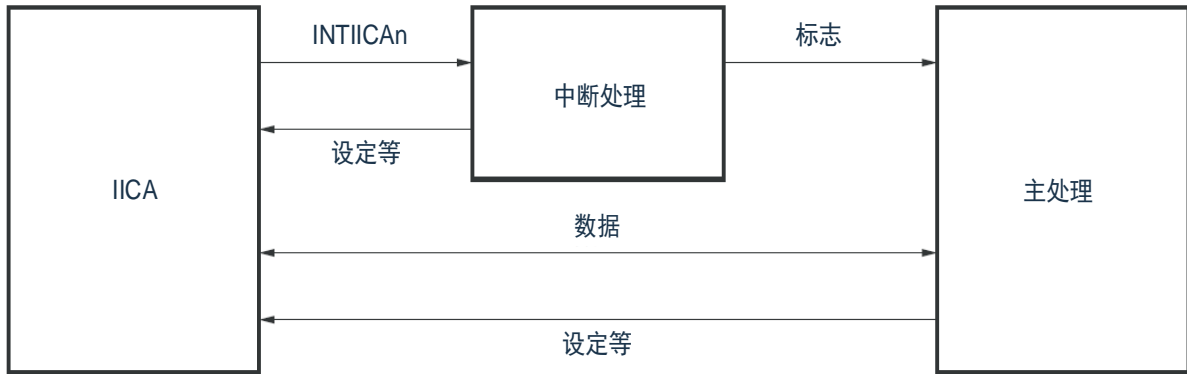
备注: n=0

(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。

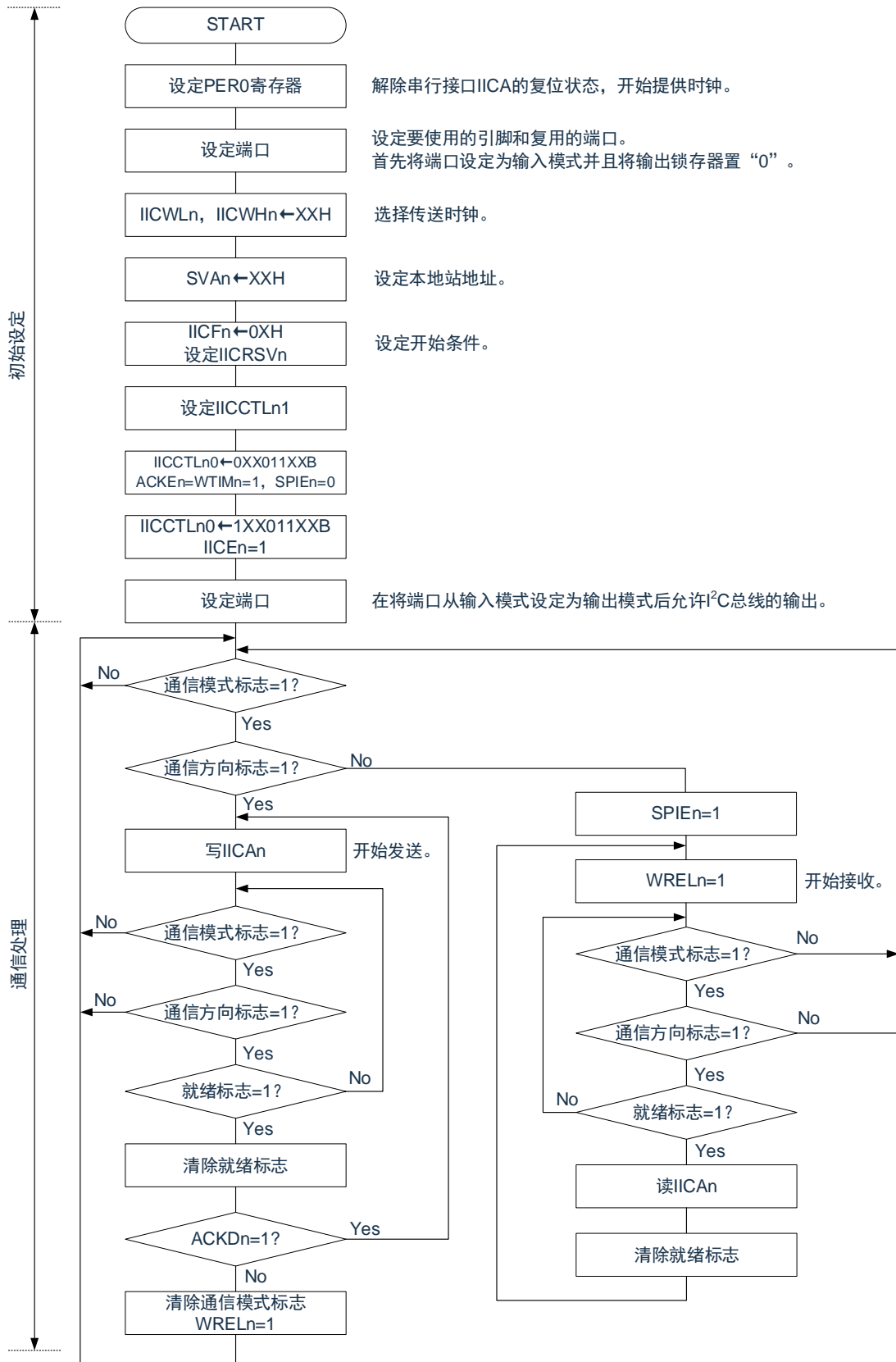
备注：n=0

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 14-21：从属运行步骤(1)



注意：传送和接收的格式必须符合通信中的产品的规格。

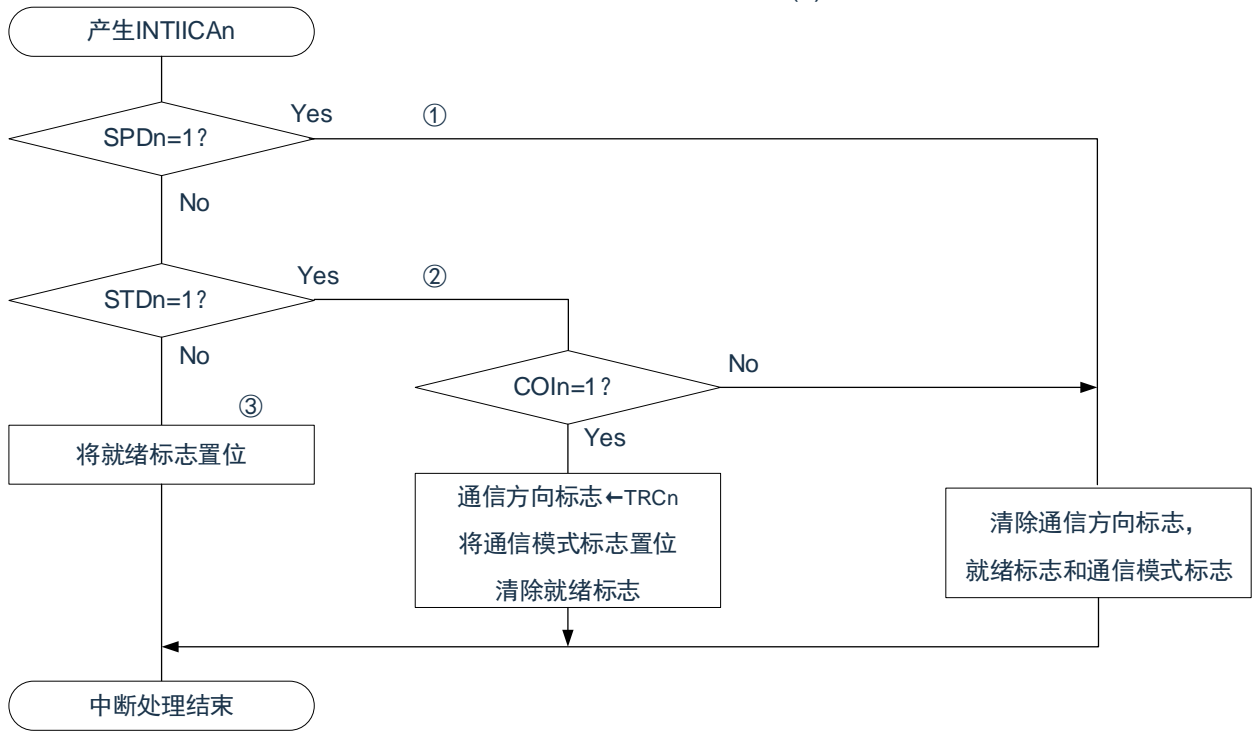
备注：n=0

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 14-21：从属运行步骤(2)”的①~③。

图 14-21：从属运行步骤(2)



备注：n=0

## 14.5.17 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。

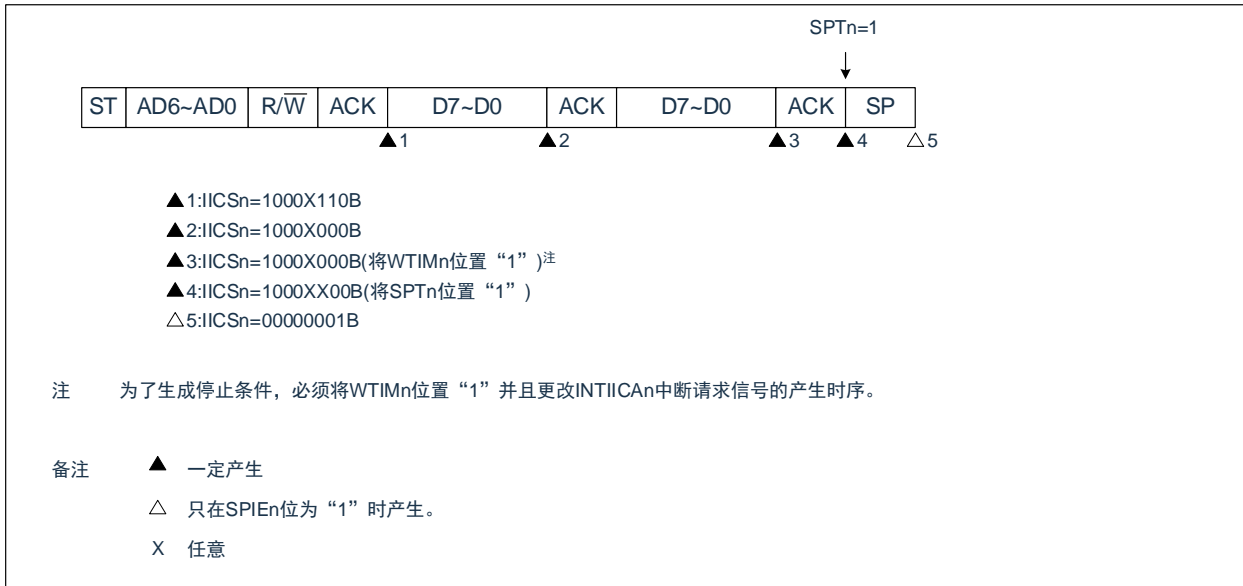
备注：

1. ST：开始条件  
AD6~AD0：地址  
R/W：传送方向的指定  
ACK：应答  
D7~D0：数据  
SP：停止条件
2. n=0

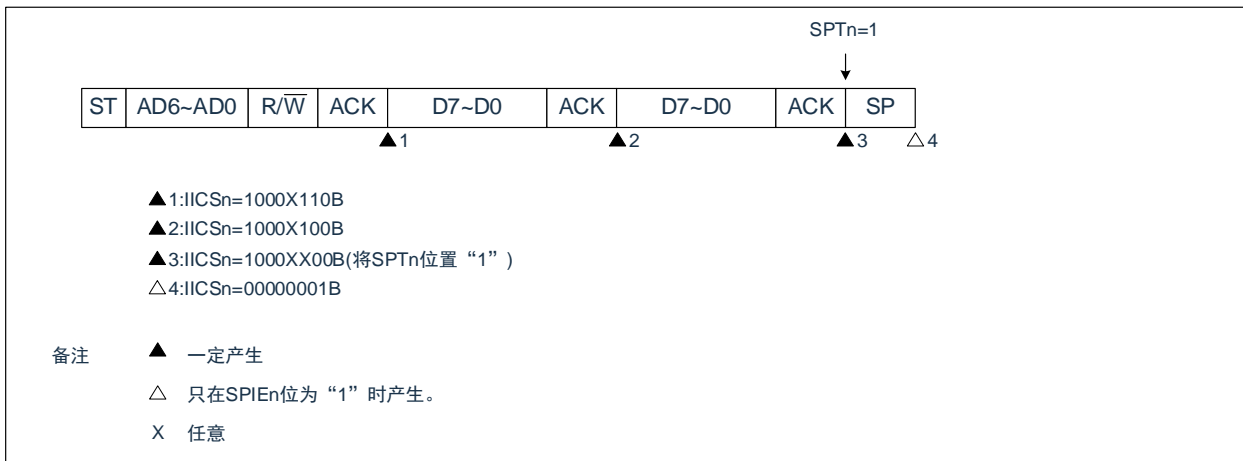
(1) 主控运行

① Start~Address~Data~Data~Stop (发送和接收)

a) WTIMn=0 的情况



b) WTIMn=1 的情况

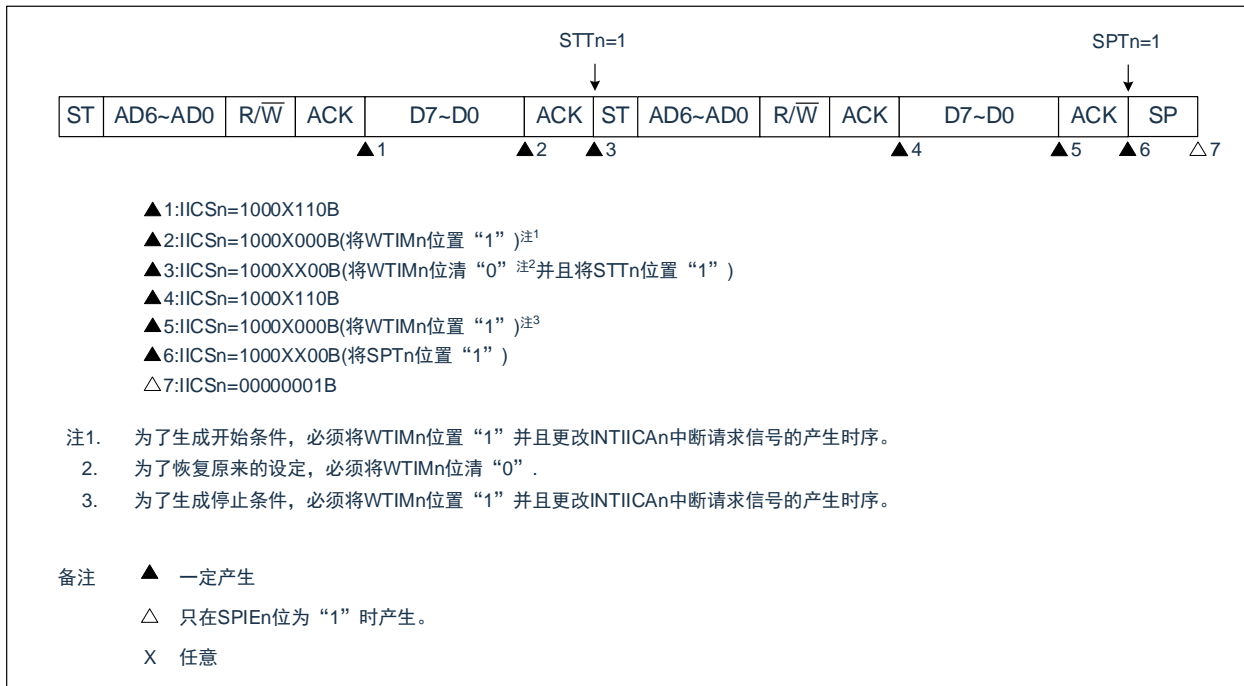


备注: n=0

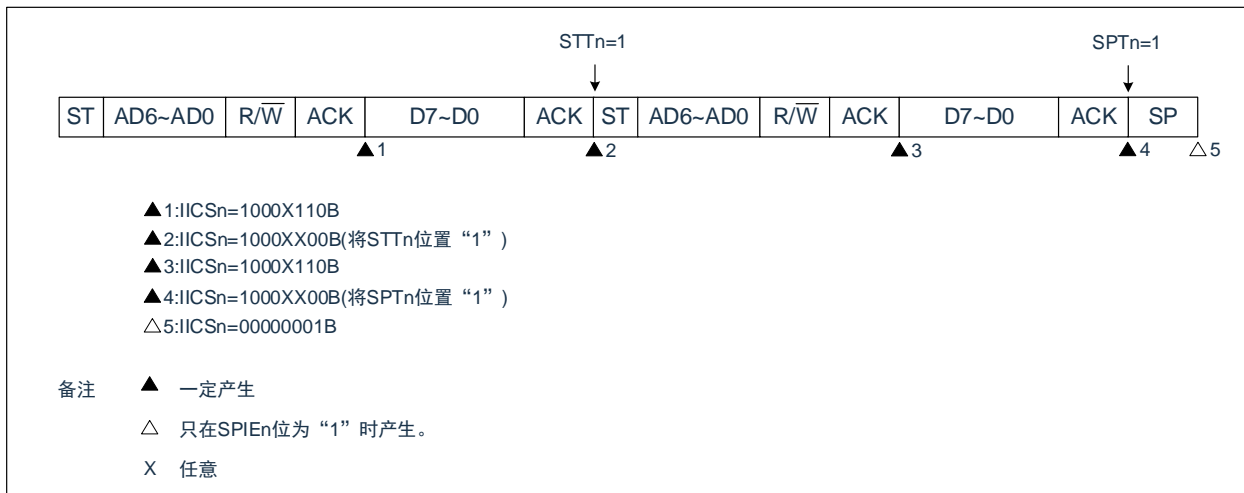


② Start~Address~Data~Start~Address~Data~Stop (重新开始)

a) WTIMn=0 的情况



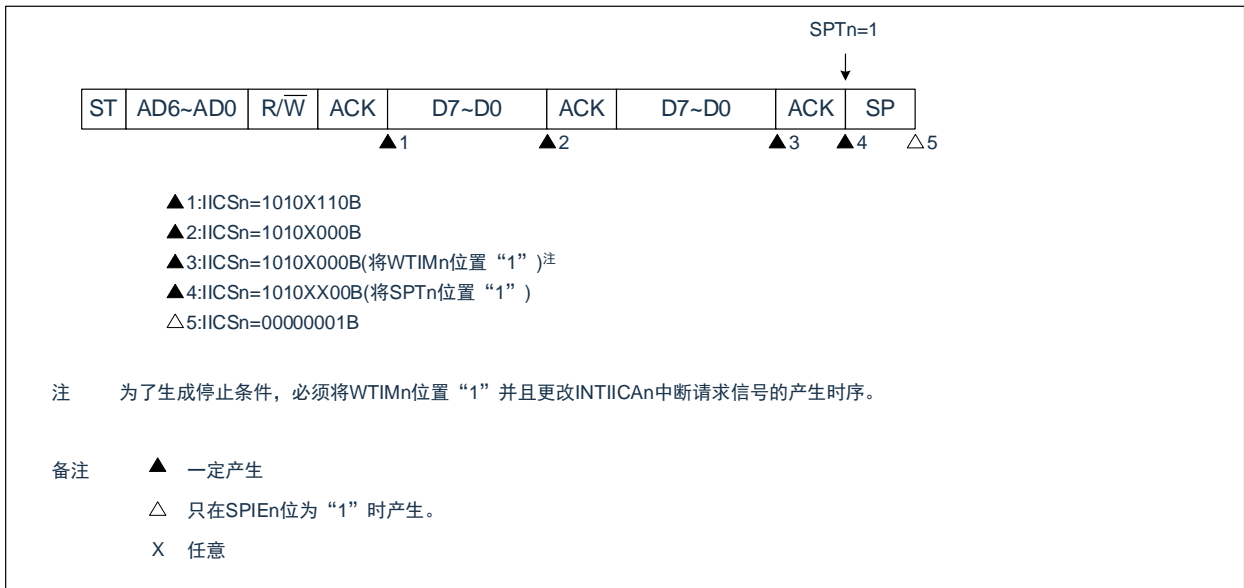
b) WTIMn=1 的情况



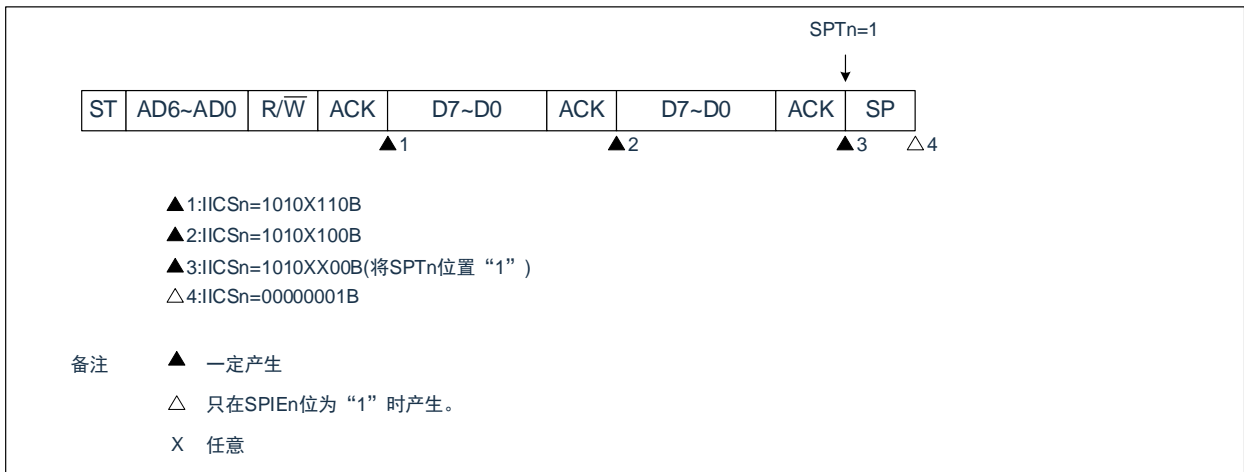
备注: n=0

③ Start~Code~Data~Data~Stop (发送扩展码)

a) WTIMn=0 的情况



b) WTIMn=1 的情况

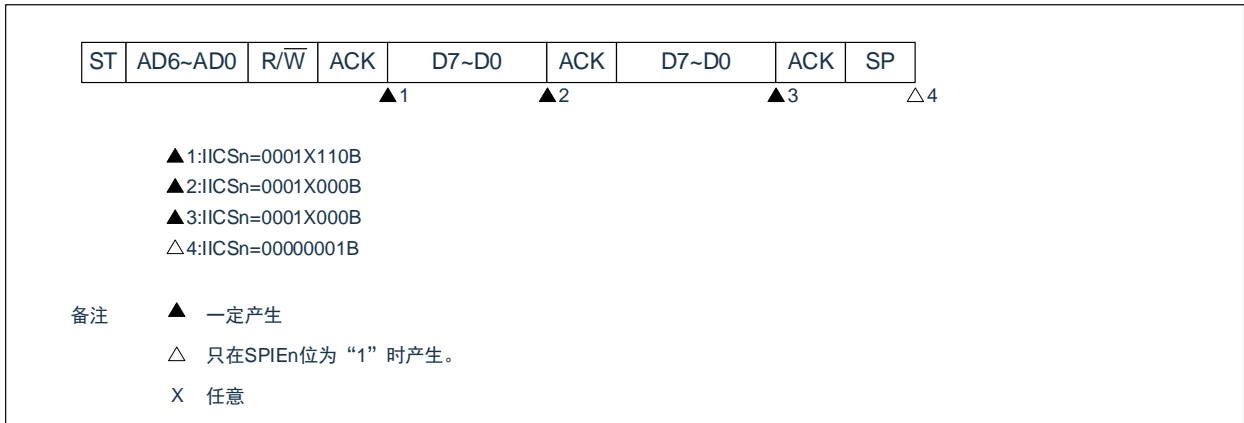


备注: n=0

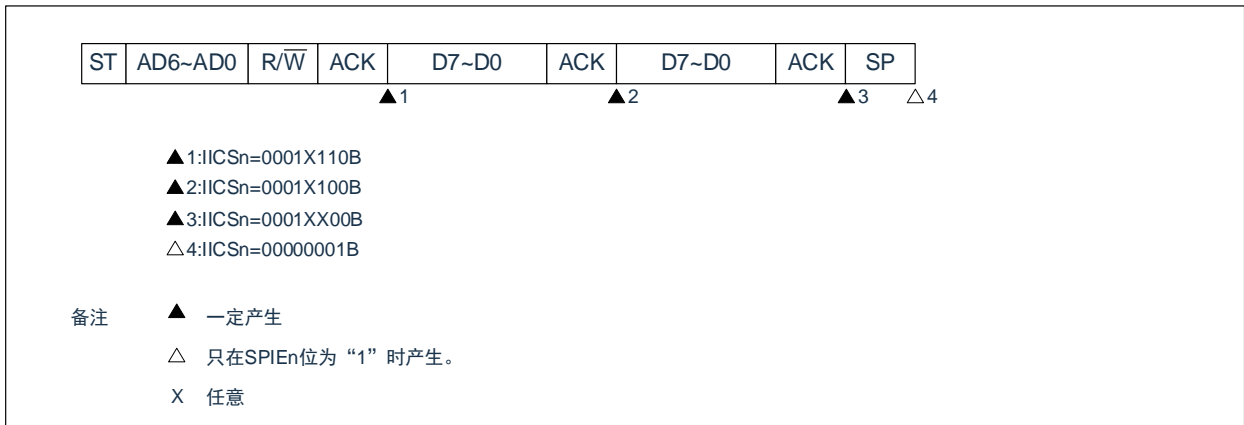
(2) 从属运行（接收从属地址的情况）

① Start~Address~Data~Data~Stop

a) WTIMn=0 的情况



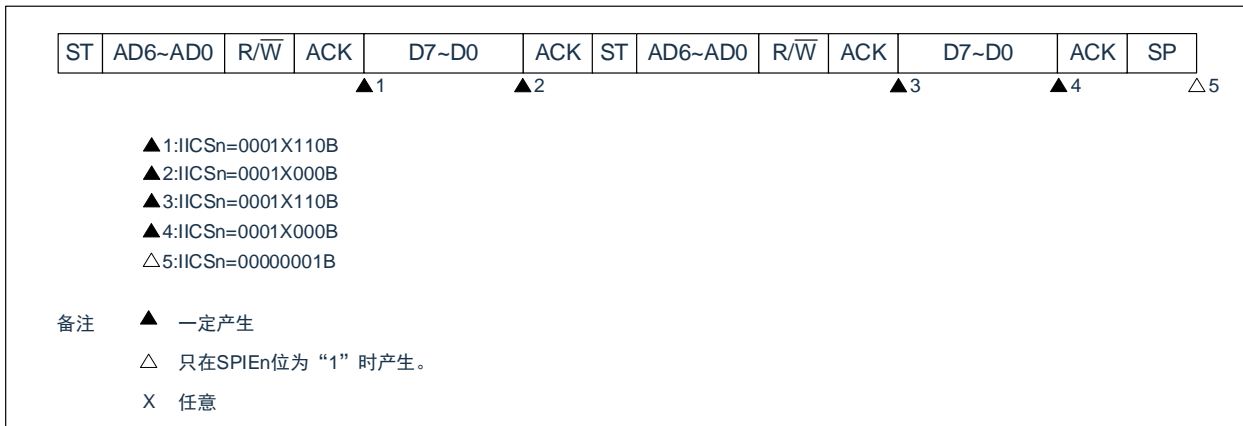
b) WTIMn=1 的情况



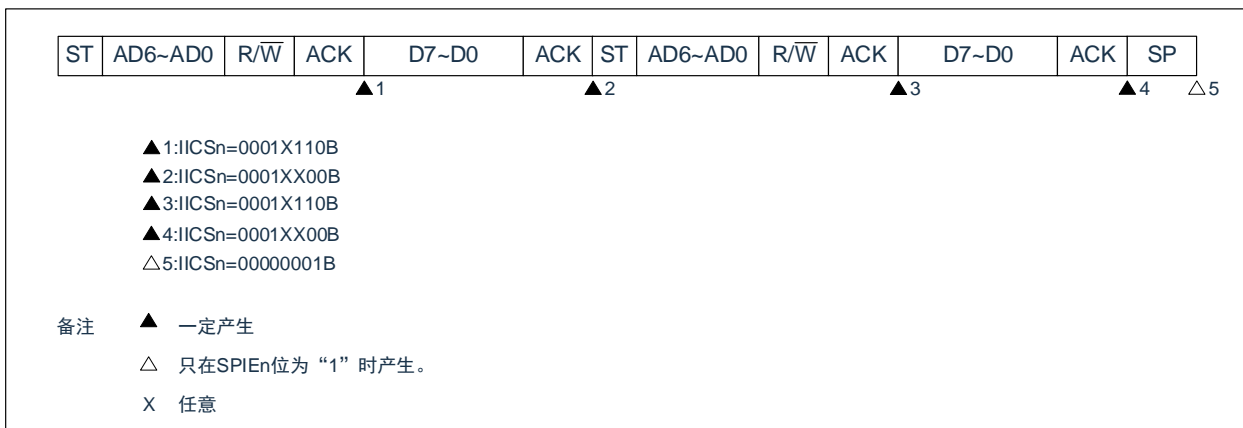
备注：n=0

② Start~Address~Data~Start~Address~Data~Stop

a) WTIMn=0 的情况（在重新开始后 SVAn 相同）



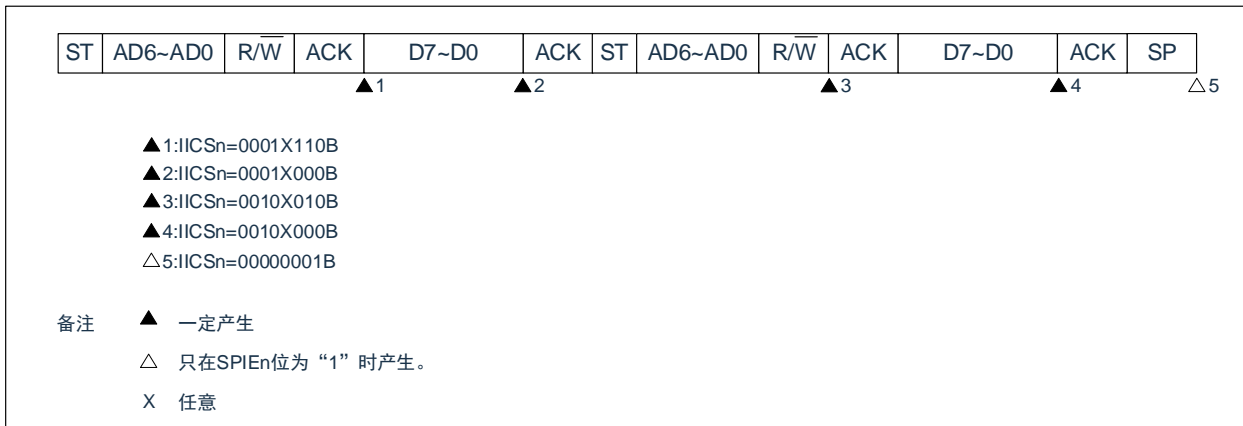
b) WTIMn=1 的情况（在重新开始后 SVAn 相同）



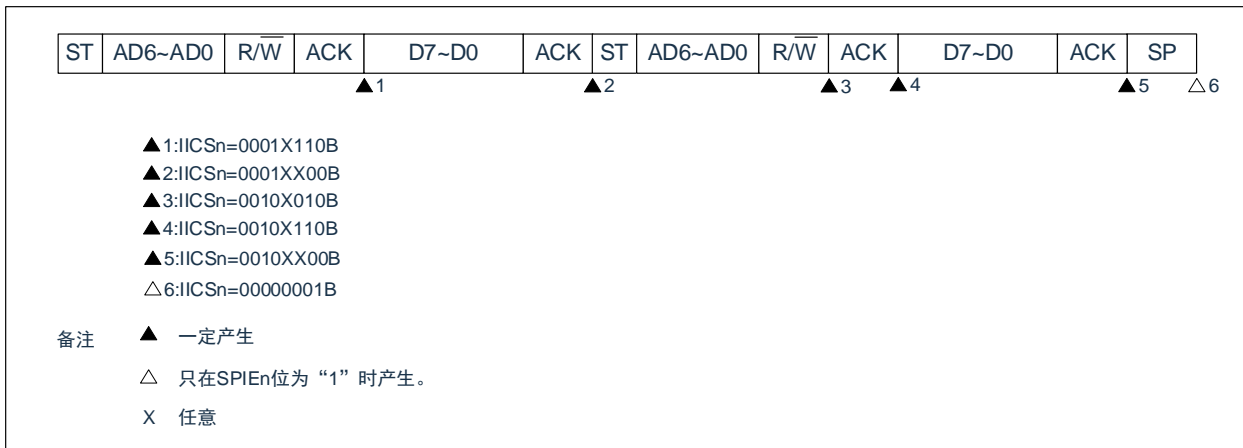
备注：n=0

③ Start~Address~Data~Start~Code~Data~Stop

a) WTIMn=0 的情况（在重新开始后地址不同（扩展码））



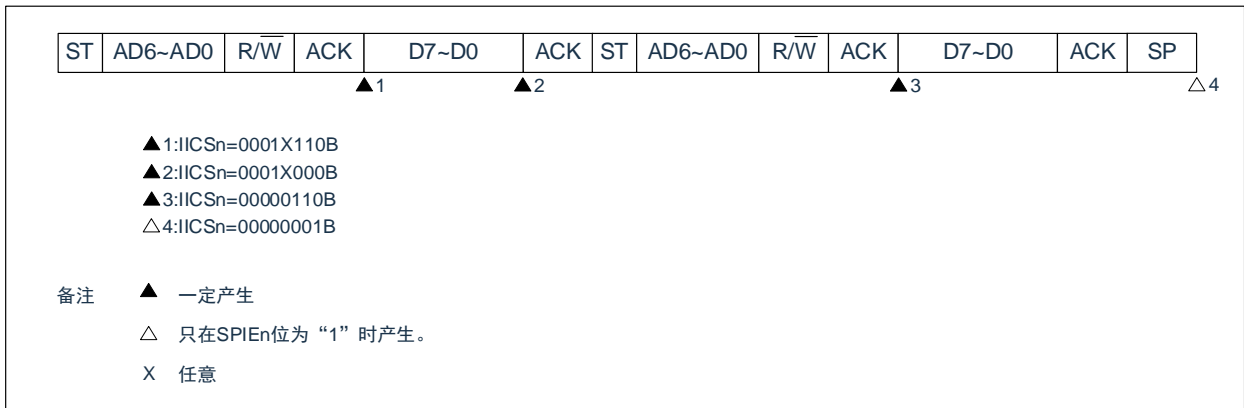
b) WTIMn=1 的情况（在重新开始后地址不同（扩展码））



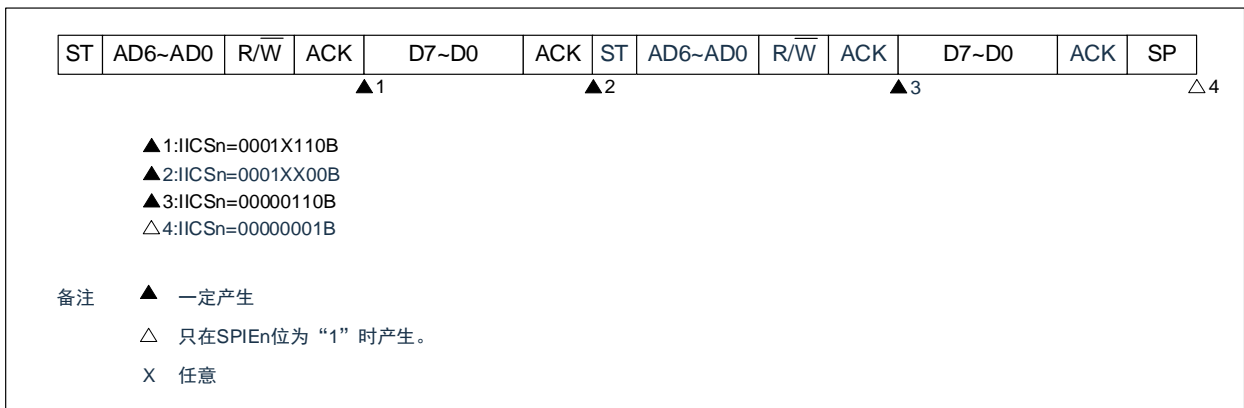
备注：n=0

④ Start~Address~Data~Start~Address~Data~Stop

a) WTIMn=0 的情况（在重新开始后地址不同（非扩展码））



b) WTIMn=1 的情况（在重新开始后地址不同（非扩展码））



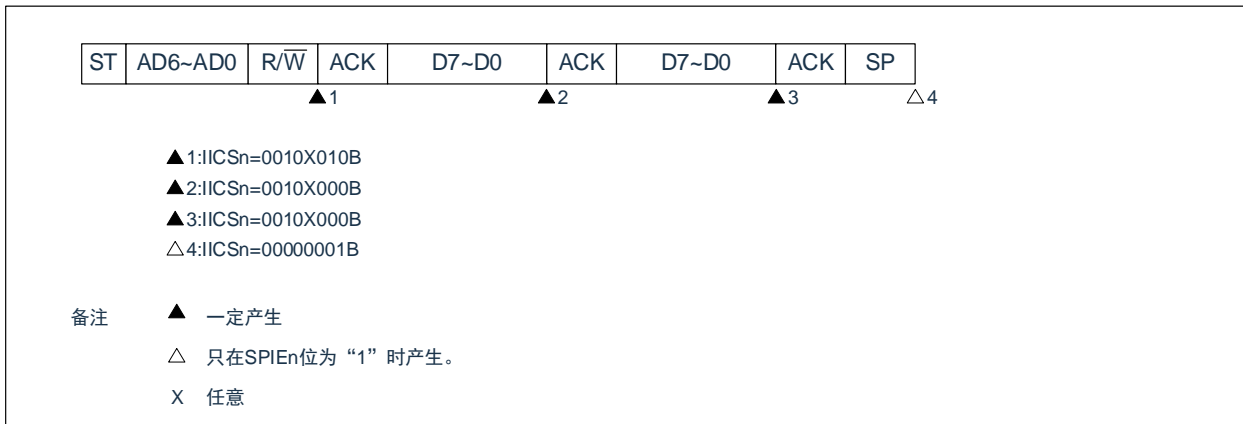
备注：n=0

(3) 从属运行（接收扩展码的情况）

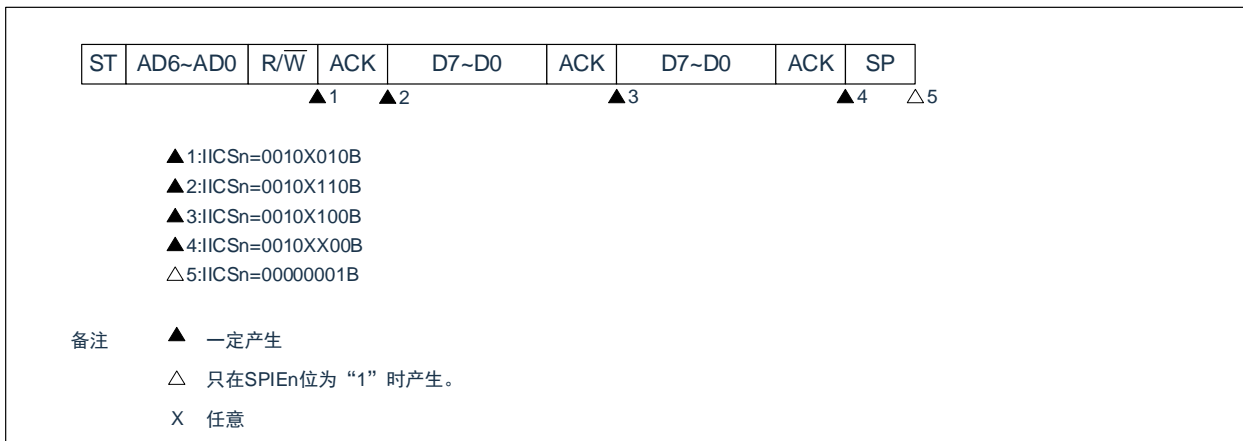
在接收扩展码时，始终参加通信。

① Start~Code~Data~Data~Stop

a) WTIMn=250 的情况



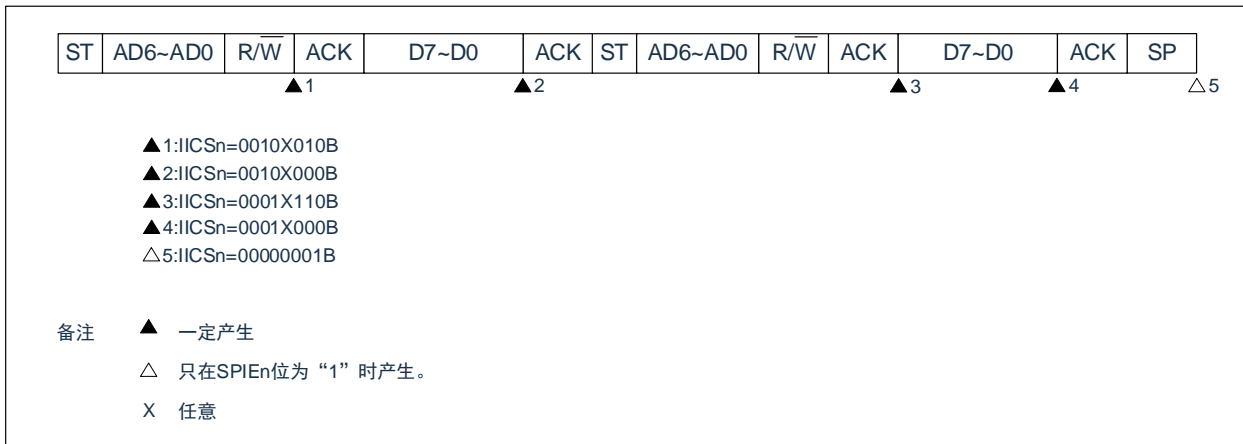
b) WTIMn=1 的情况



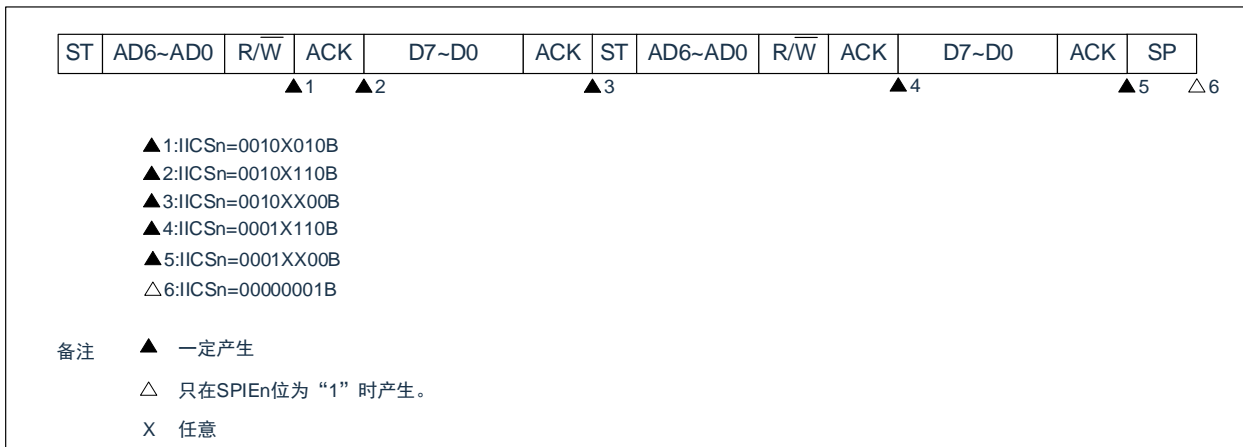
备注：n=0

② Start~Code~Data~Start~Address~Data~Stop

a) WTIMn=0 的情况（在重新开始后 SVAn 相同）



b) WTIMn=1 的情况（在重新开始后 SVAn 相同）

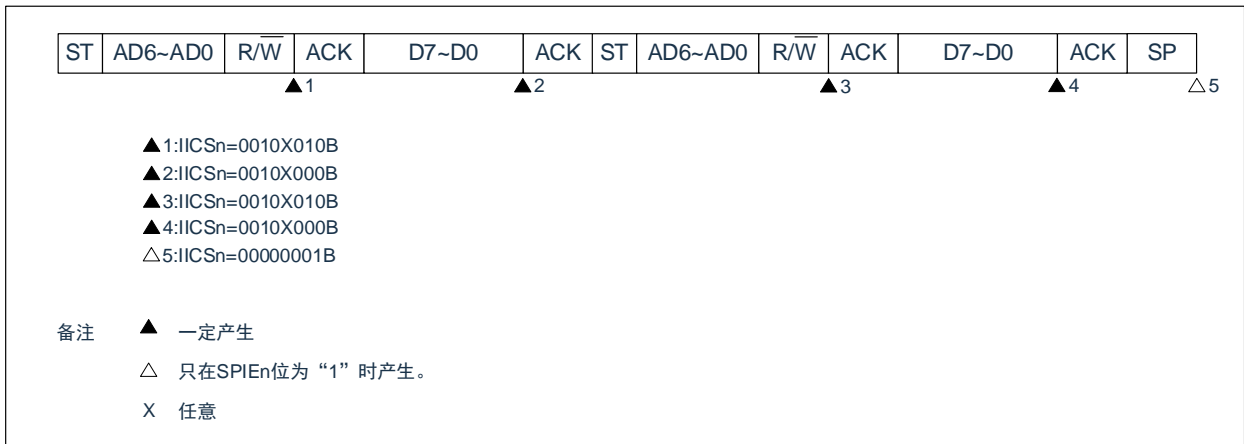


备注：n=0

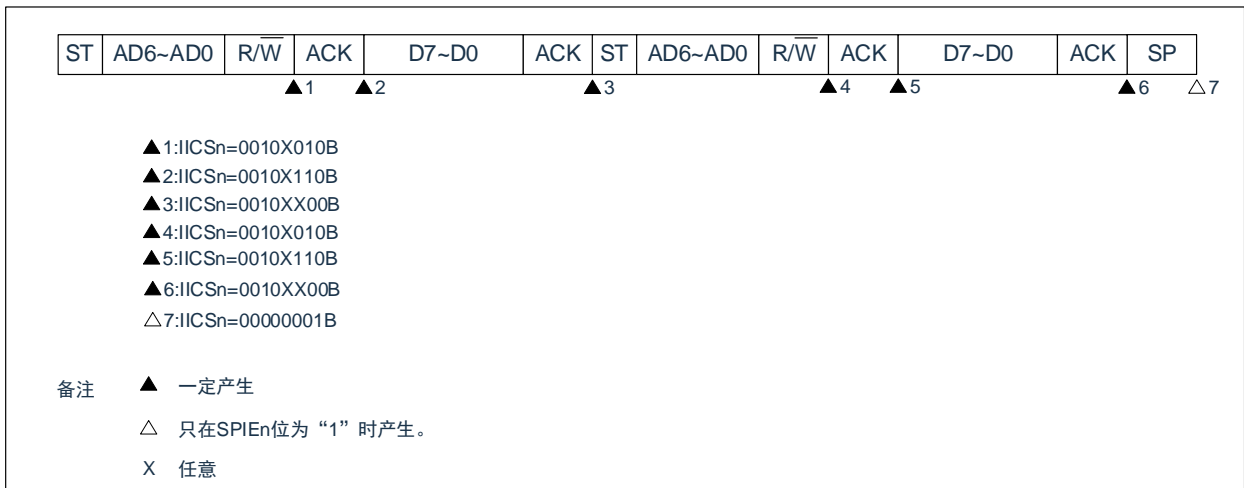


③ Start~Code~Data~Start~Code~Data~Stop

a) WTIMn=0 的情况（在重新开始后接收扩展码）



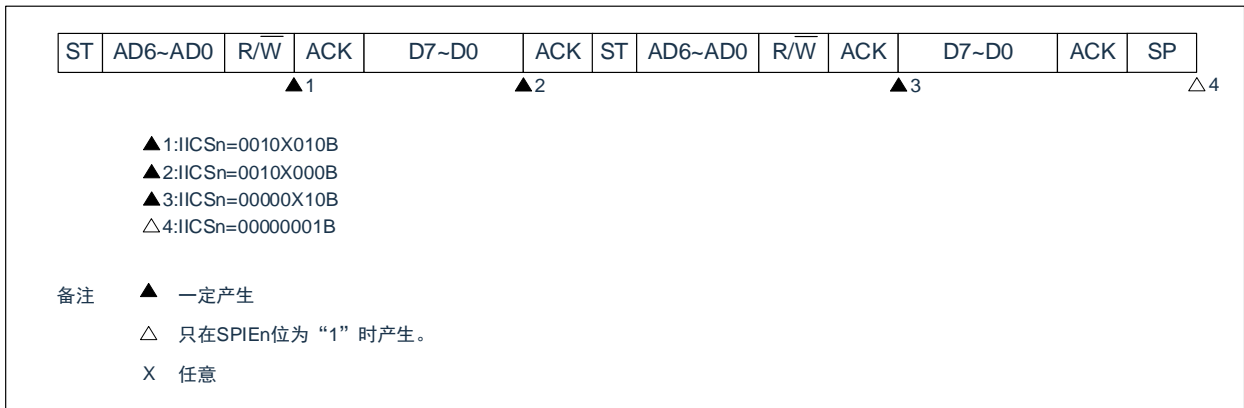
b) WTIMn=1 的情况（在重新开始后接收扩展码）



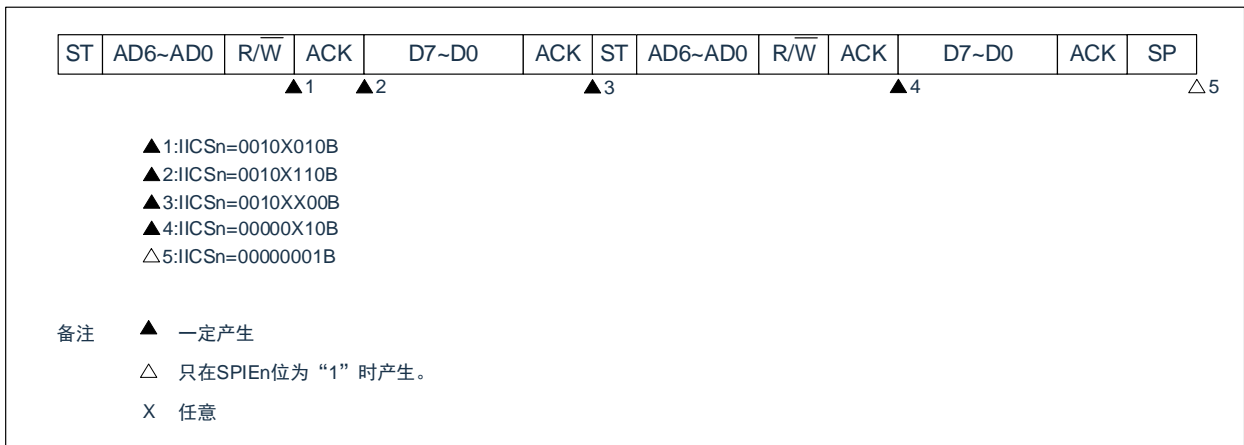
备注：n=0

④ Start~Code~Data~Start~Address~Data~Stop

a) WTIMn=0 的情况（在重新开始后地址不同（非扩展码））



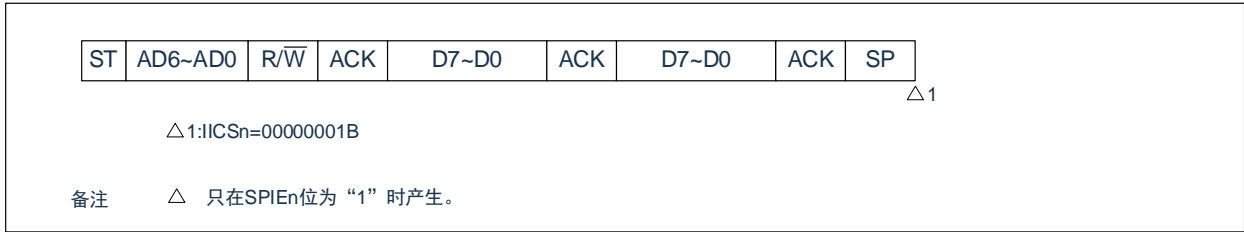
b) WTIMn=1 的情况（在重新开始后地址不同（非扩展码））



备注：n=0

(4) 不参加通信的运行

Start~Code~Data~Data~Stop

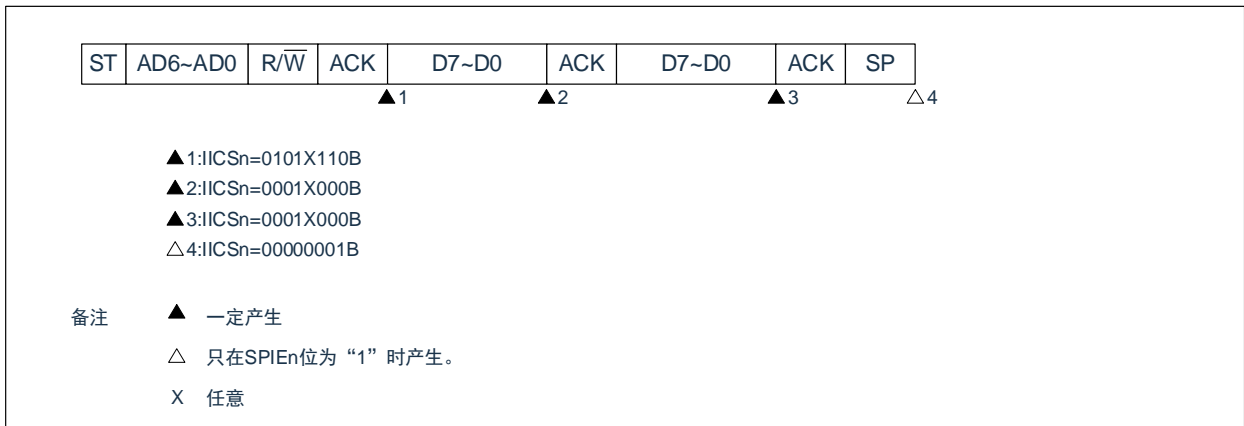


(5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）

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 MSTSn 位，确认仲裁结果。

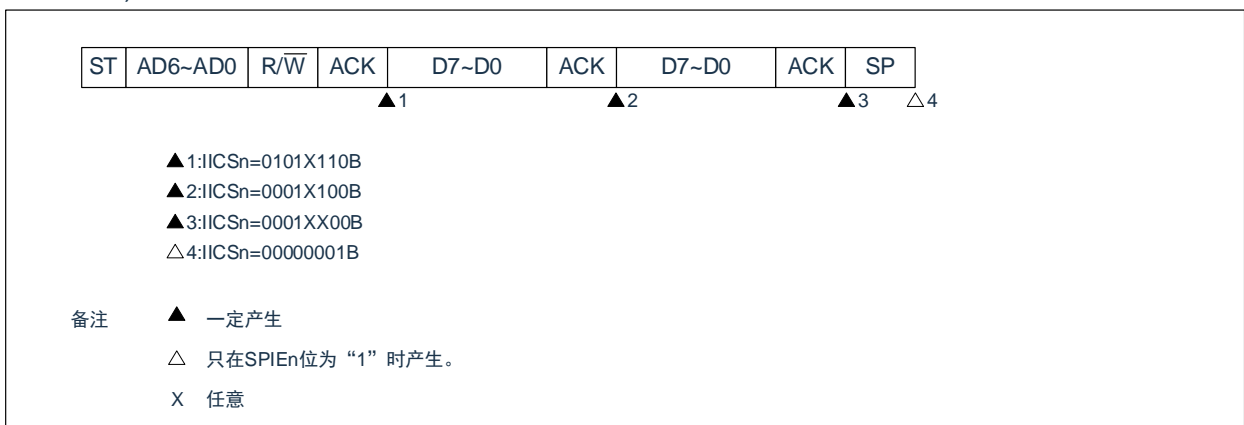
① 在发送从属地址数据的过程中仲裁失败的情况

a) WTIMn=0 的情况



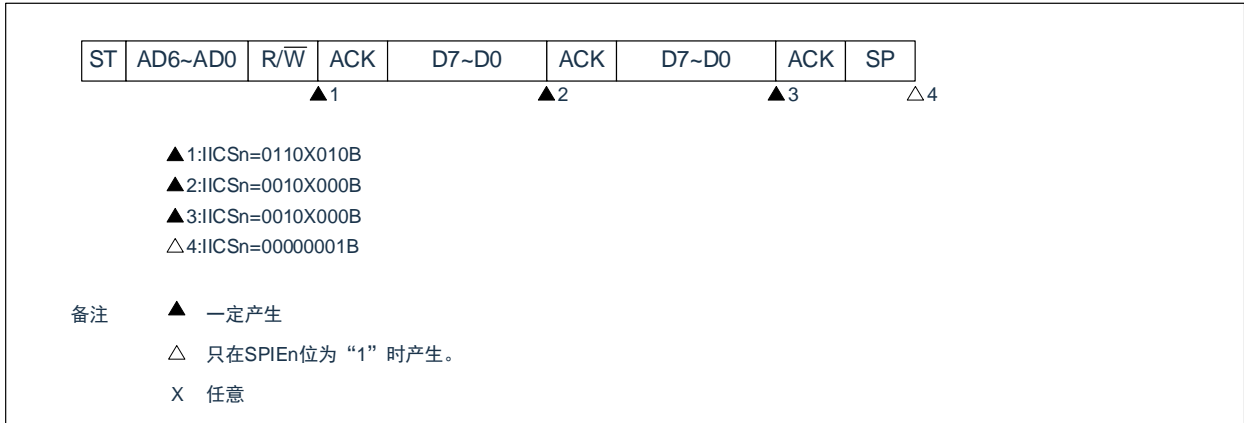
备注：n=0

b) WTIMn=1 的情况



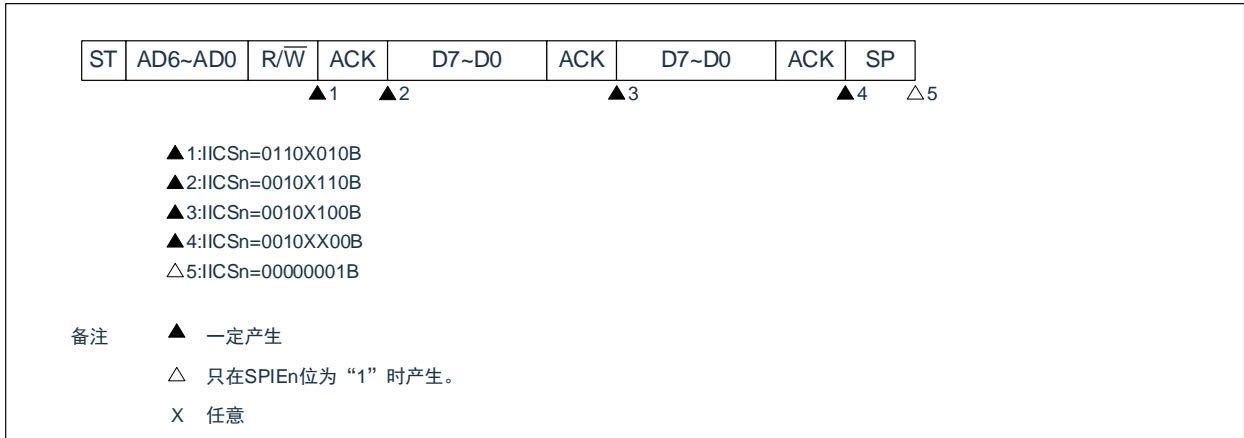
② 在发送扩展码的过程中仲裁失败的情况

a) WTIMn=0 的情况



备注：n=0

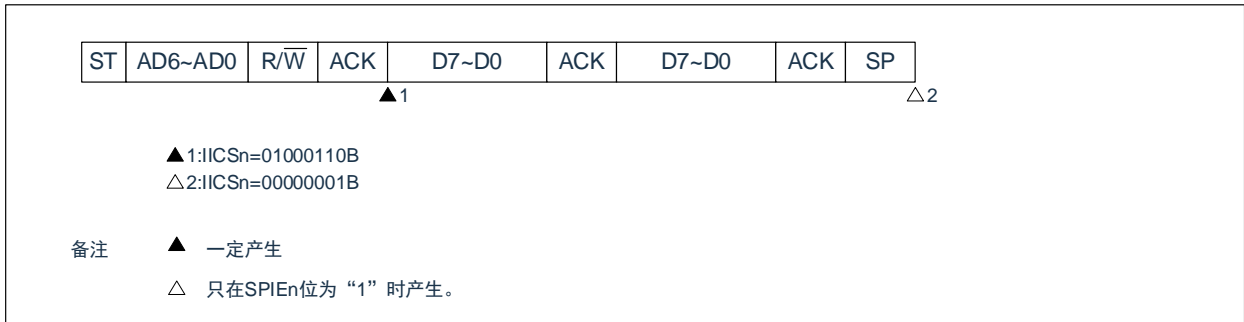
b) WTIMn=1 的情况



(6) 仲裁失败的运行（在仲裁失败后不参加通信）

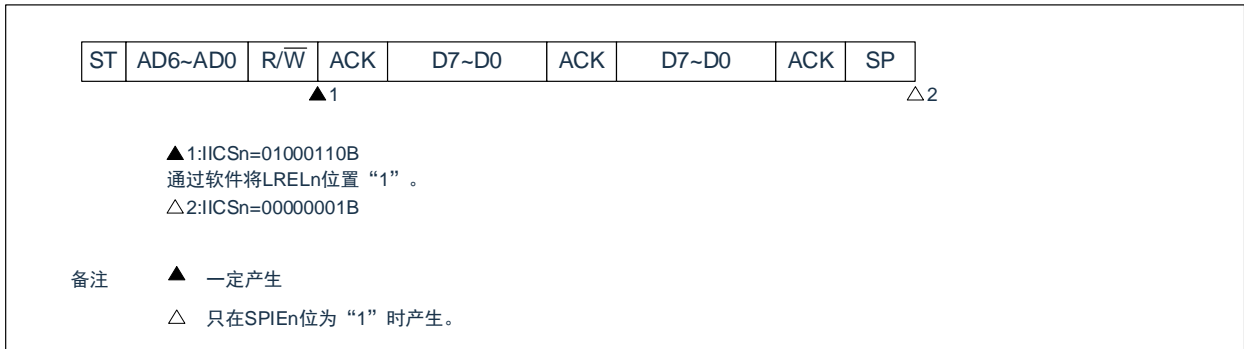
在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

① 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）



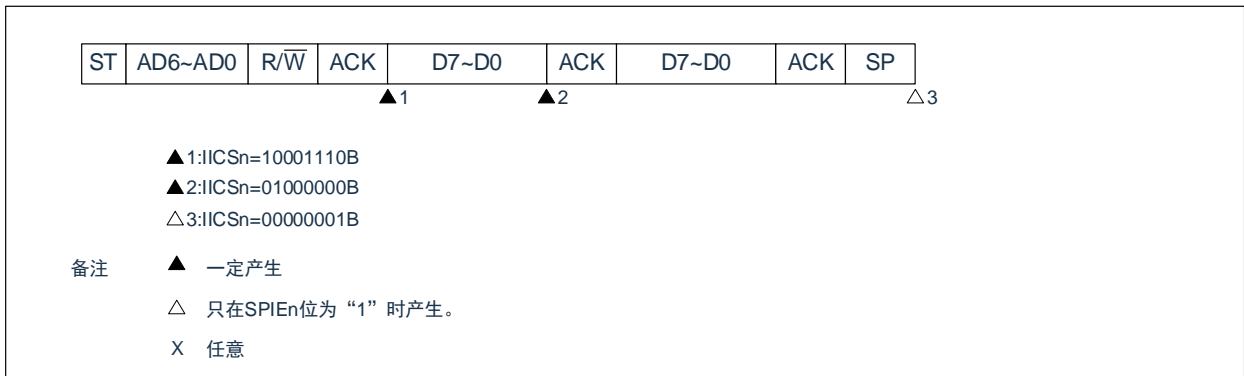
备注：n=0

② 在发送扩展码的过程中仲裁失败的情况



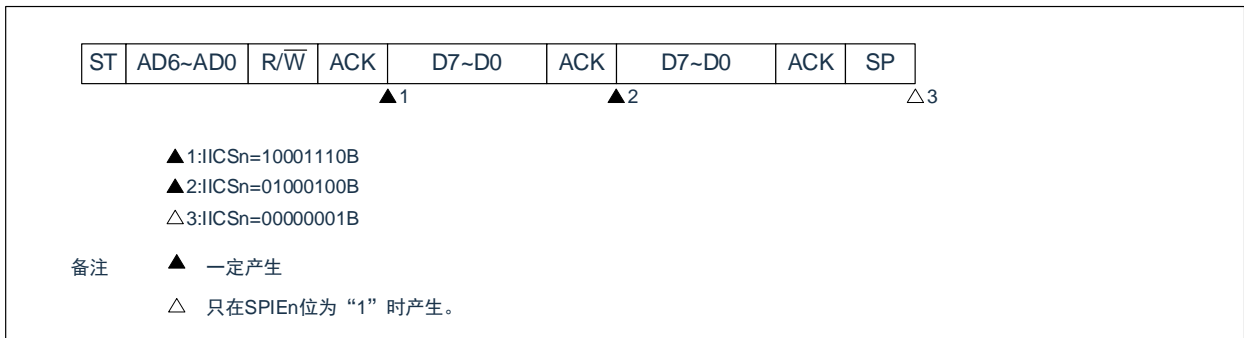
③ 在传送数据时仲裁失败的情况

a) WTIMn=0 的情况



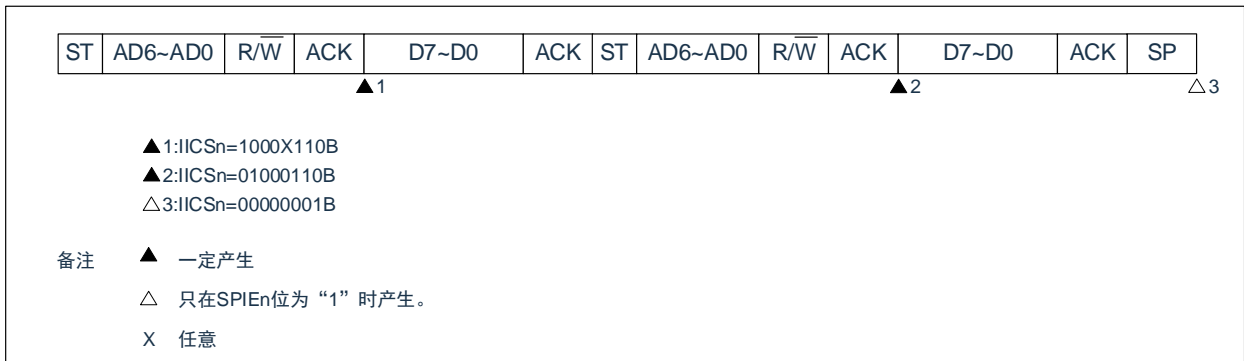
备注: n=0

b) WTIMn=1 的情况



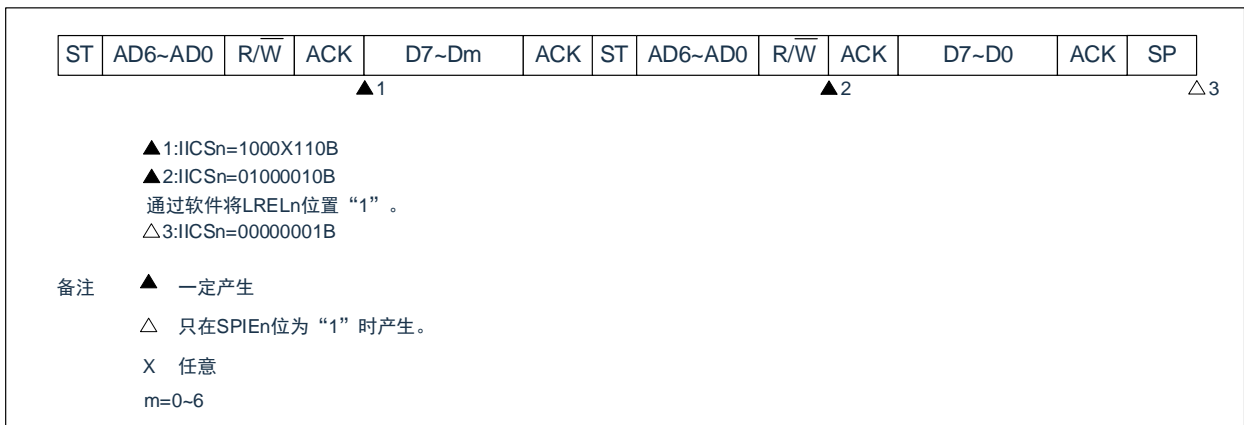
④ 在传送数据时因重新开始条件而仲裁失败的情况

a) 非扩展码（例如，SVAn不同）

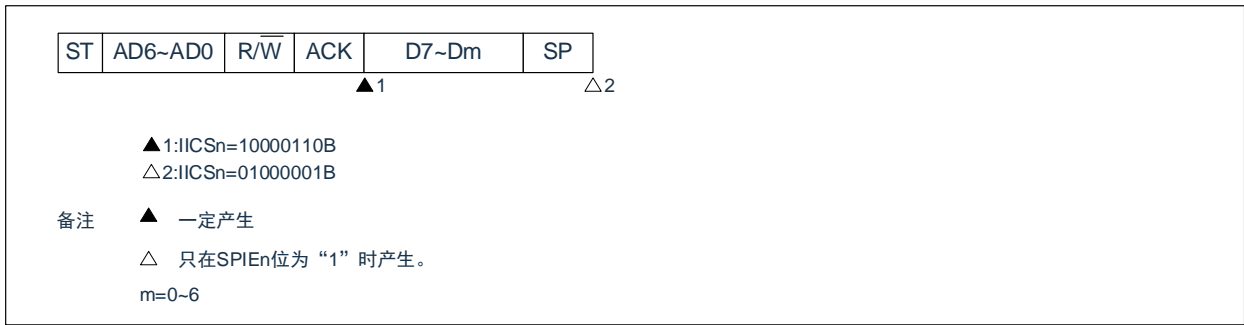


备注：n=0

b) 扩展码



⑤ 在传送数据时因停止条件而仲裁失败的情况

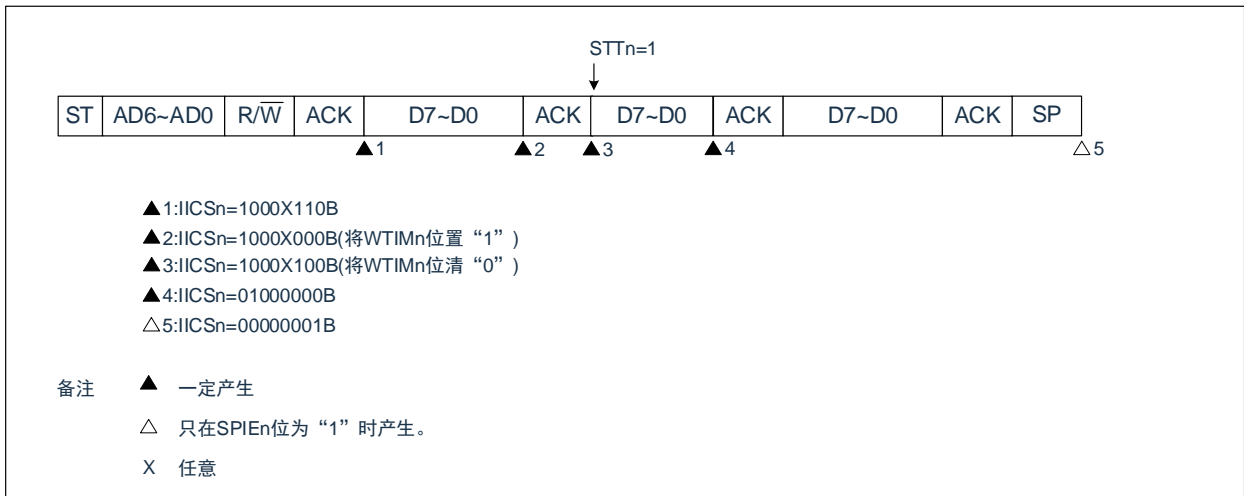


备注：n=0

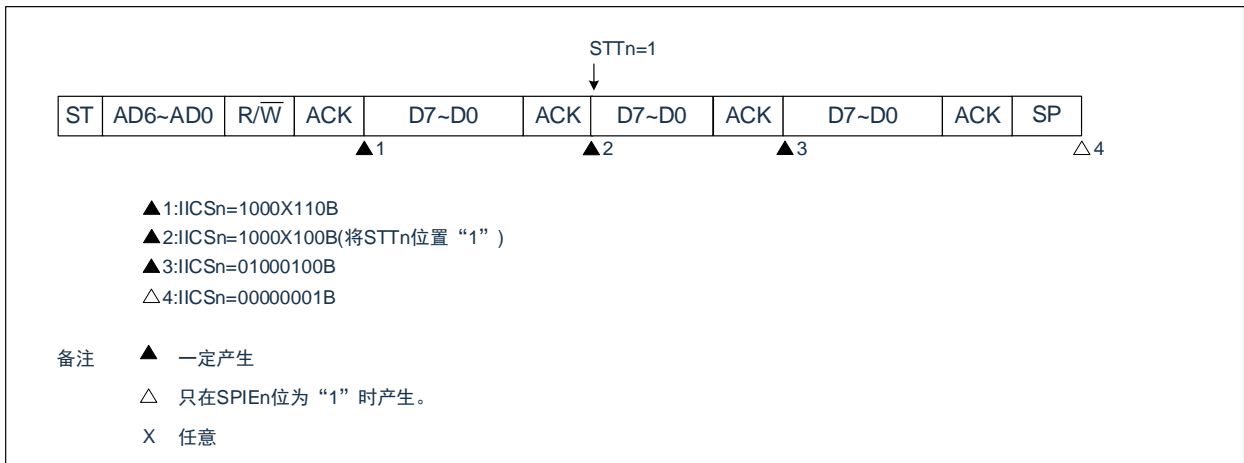


⑥ 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

a) WTIMn=0 的情况



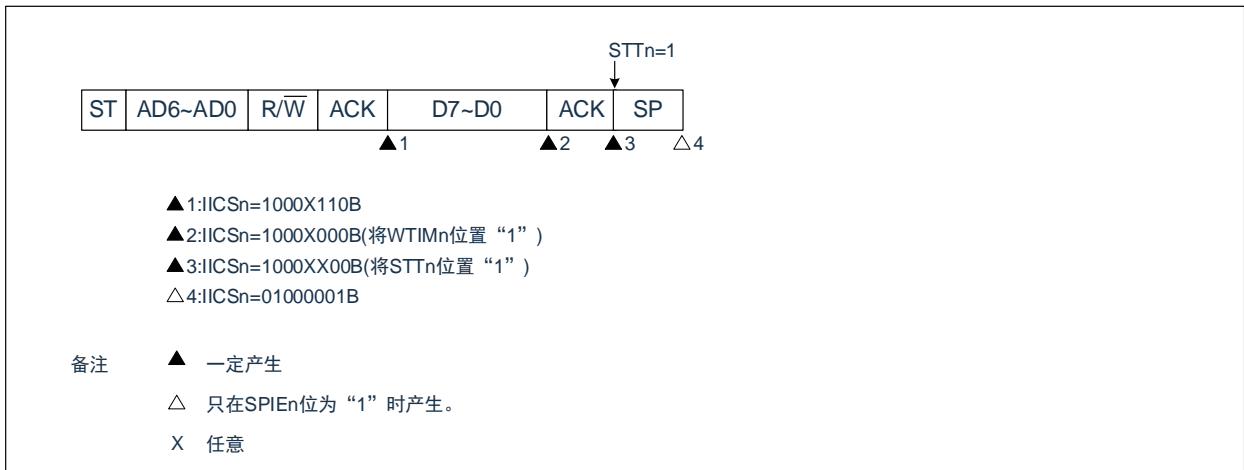
b) WTIMn=1 的情况



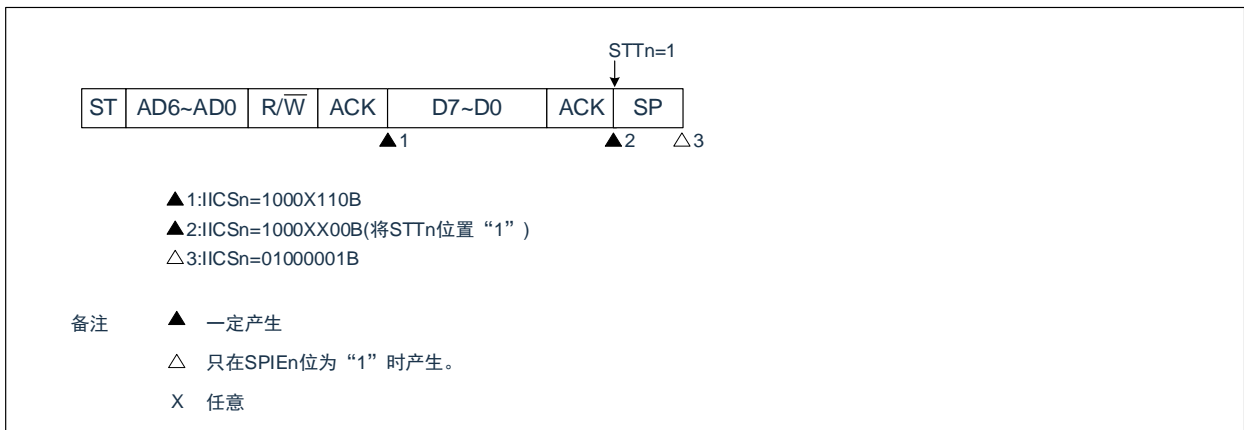
备注: n=0

⑦ 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

a) WTIMn=0 的情况



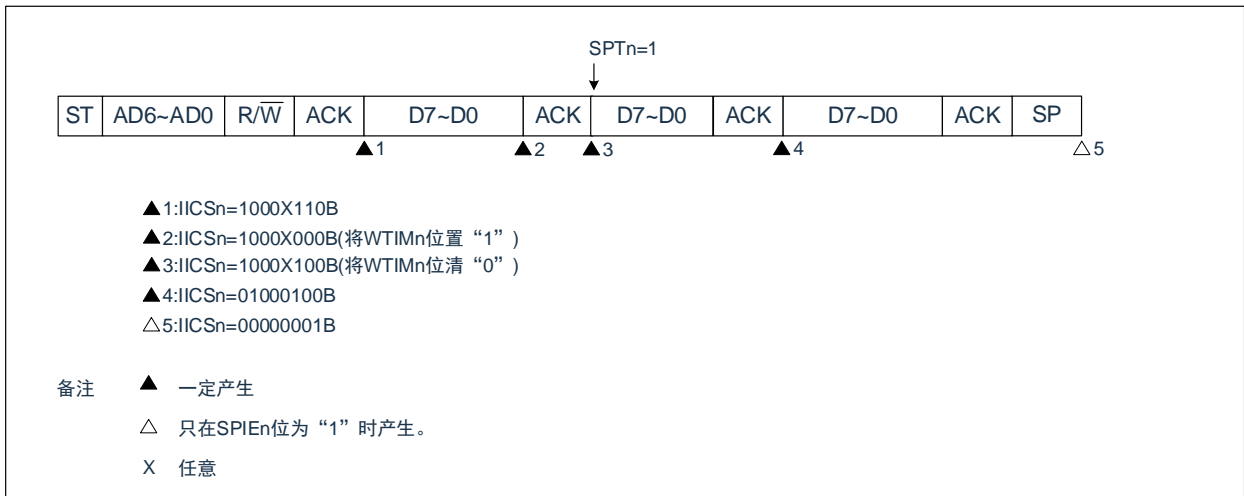
b) WTIMn=1 的情况



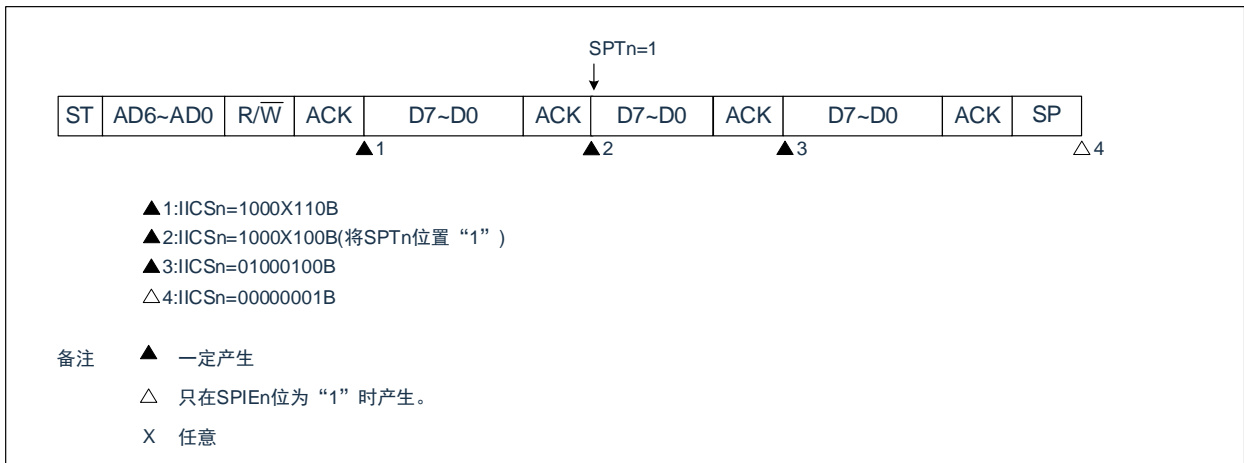
备注: n=0

⑧ 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

a) WTIMn=0 的情况



b) WTIMn=1 的情况



备注: n=0

## 14.6 时序图

在 I<sup>2</sup>C 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 14-22 和图 14-23 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。

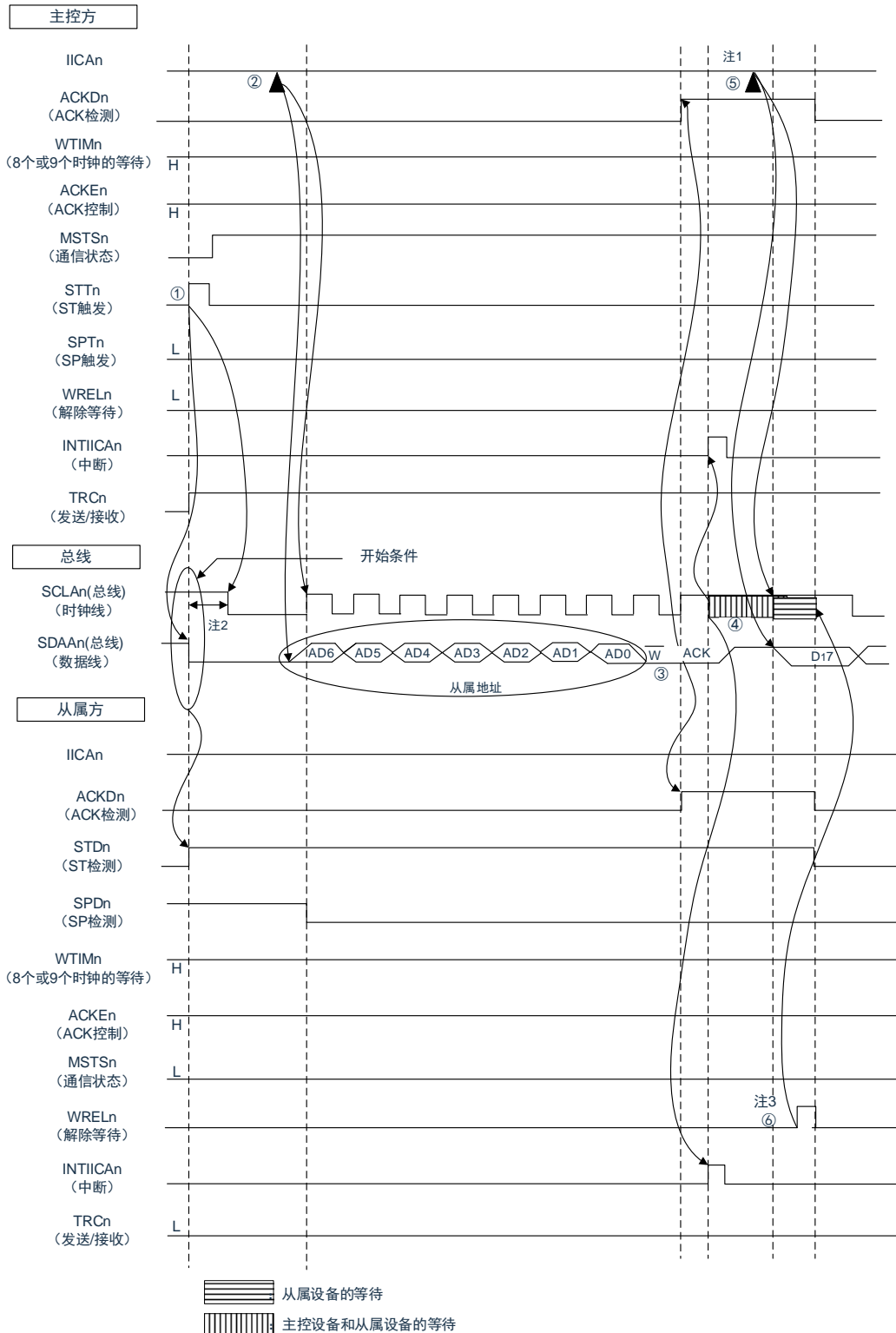
在 SCLAn 的上升沿将 SDAAn 引脚输入的数据取到 IICAn。

备注：n=0

图 14-22: 主控设备&从属设备的通信例子

(主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (1/4)

(1) 开始条件~地址~数据



注 1: 要解除主控方发送期间的等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2: 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间, 在设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。

注 3: 要解除从属方接收期间的等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。

图 14-22 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<sup>注</sup>，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）<sup>注</sup>。

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）。

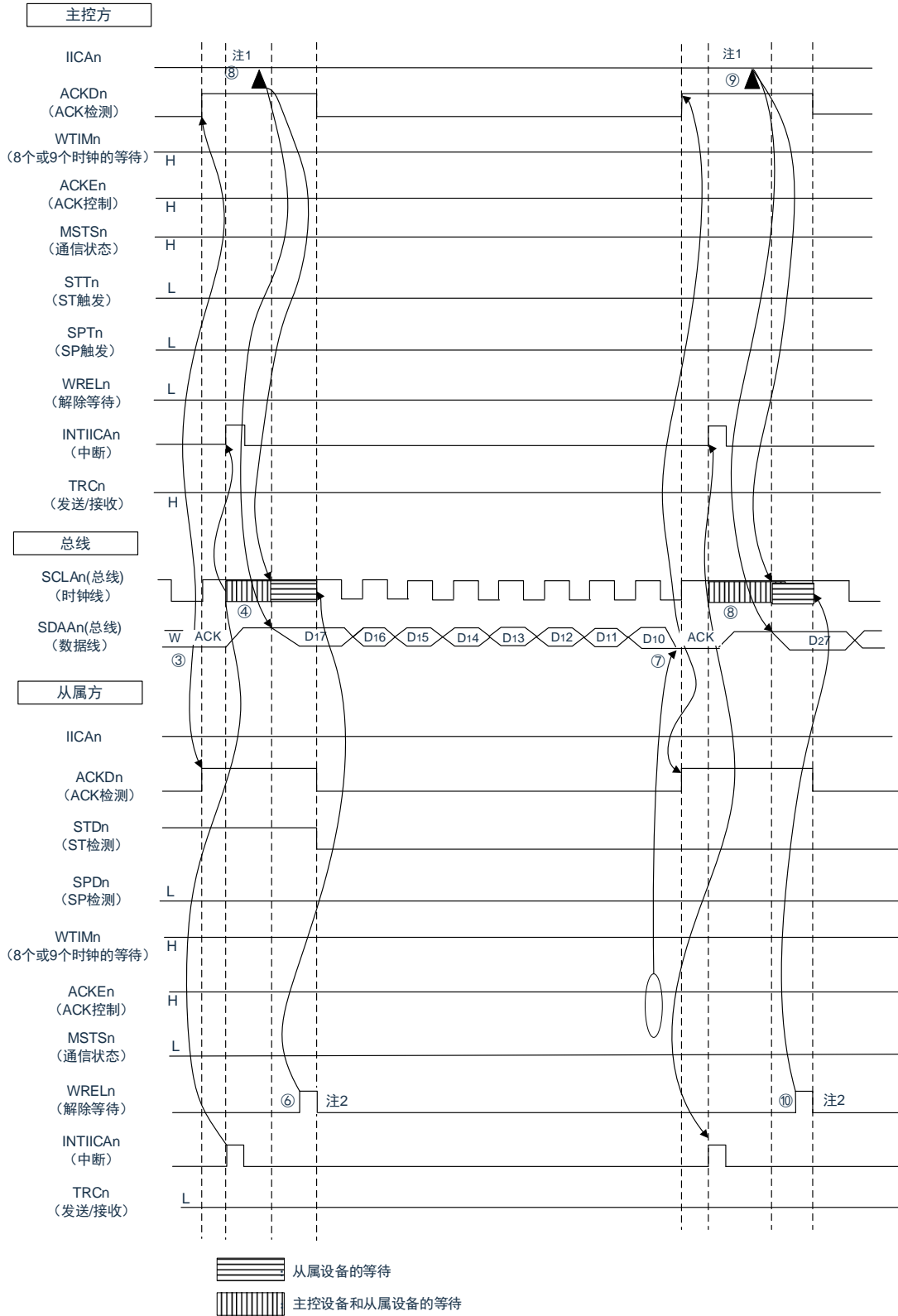
备注：

1. 图 14-22 的①~⑮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 14-22 的“(1)开始条件~地址~数据”说明步骤①~⑥。  
 图 14-22 的“(2)地址~数据~数据”说明步骤③~⑩。  
 图 14-22 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图 14-22：主控设备&从属设备的通信例子

(主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/4)

(2) 地址~数据~数据



注 1：要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2：要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注：n=0

图 14-22 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<sup>注</sup>，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）<sup>注</sup>。

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：

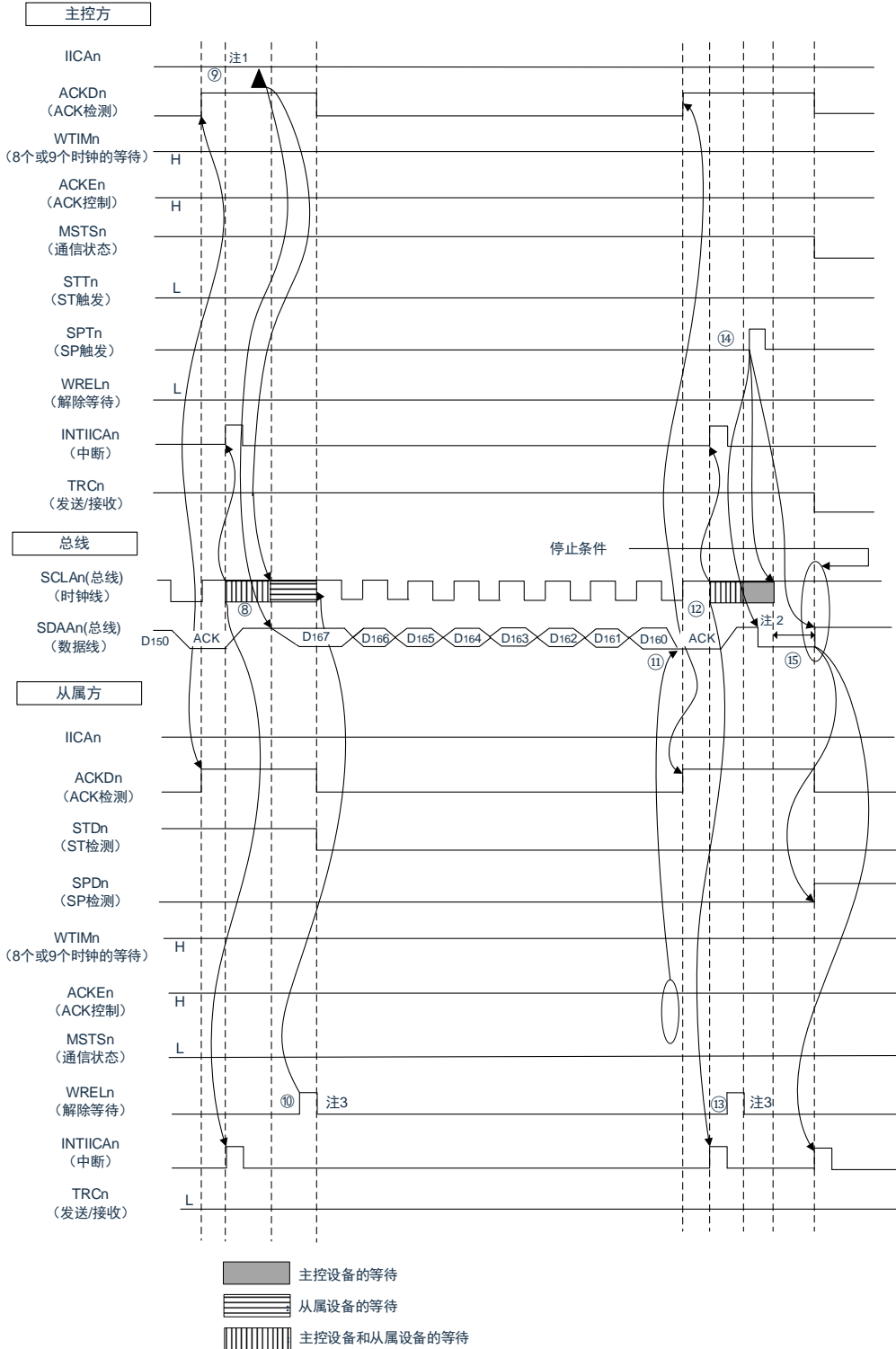
1. 图 14-22 的①~⑮是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。  
 图 14-22 的“(1)开始条件~地址~数据”说明步骤①~⑥。  
 图 14-22 的“(2)地址~数据~数据”说明步骤③~⑩。  
 图 14-22 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0



图 14-22：主控设备&从属设备的通信例子

(主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (3/4)

(3) 数据~数据~停止条件



注 1：要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2：在发行停止条件后，从 SCLAn 引脚信号上升到生成停止条件的的时间，在设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。

注 3：要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注：n=0

图 14-22 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)。

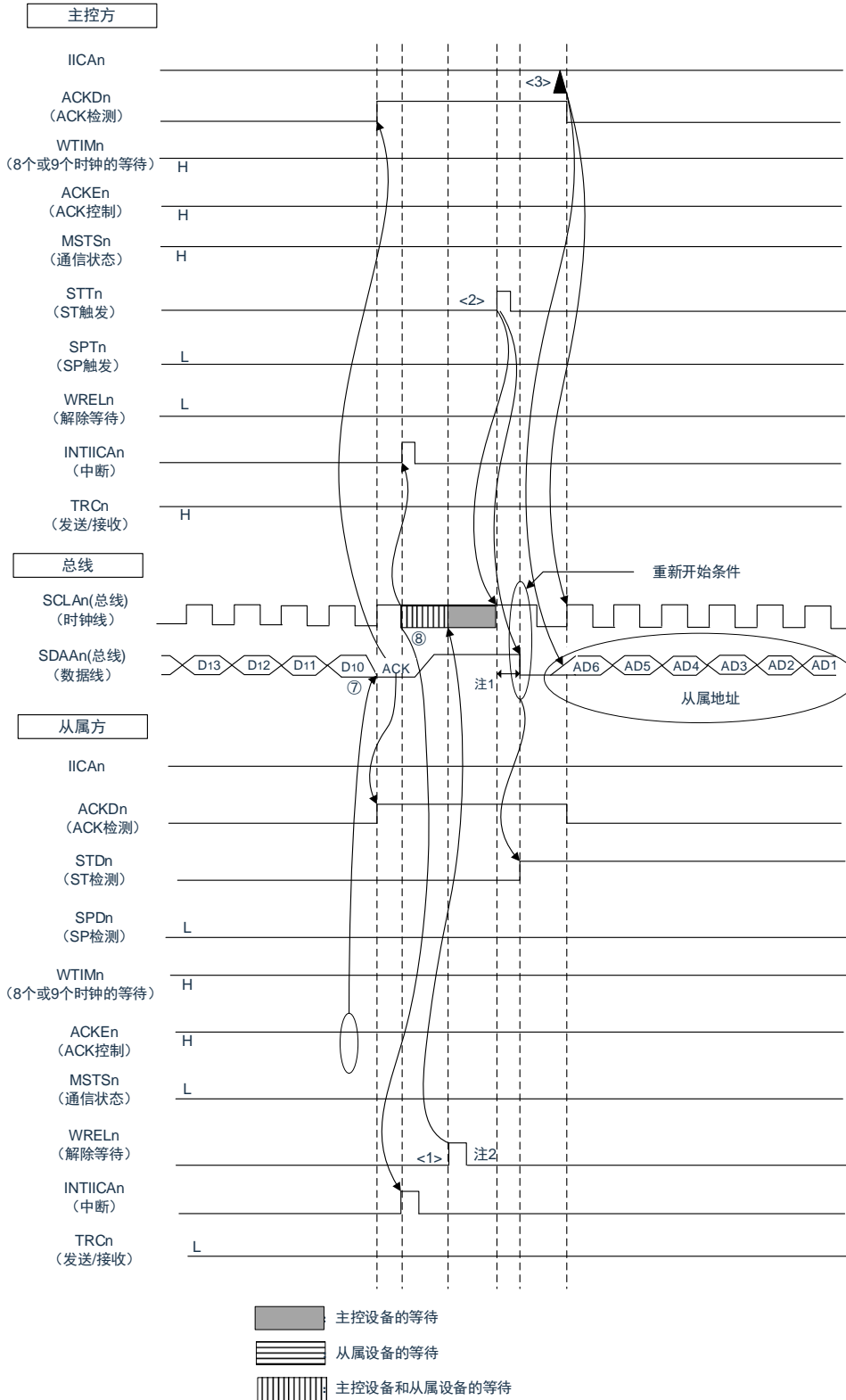
备注：

1. 图 14-22 的①~⑮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 14-22 的“(1)开始条件~地址~数据”说明步骤①~⑥。  
 图 14-22 的“(2)地址~数据~数据”说明步骤③~⑩。  
 图 14-22 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图 14-22：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的时间，在设定为标准模式时至少为 4.7us，在设定为快速模式时至少为 0.6us。

注 2：要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注：n=0

图 14-22 的“(4)数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行<1>~<3>, 从而返回到步骤③的数据发送步骤。

⑦在数据传送结束后, 因为从属方的 ACKEn 位为“1”, 所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。

⑧主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且都产生中断 (INTIICAn: 传送结束中断)。

<1>从属方读接收数据, 解除等待 (WRELn=1)。

<2>如果在主控方再次将开始条件触发置位 (STTn=1), 总线时钟线就上升 (SCLAn=1), 而且在经过重新开始条件的准备时间后总线数据线下降 (SDAAn=0), 生成开始条件 (通过 SCLAn=1 使 SDAAn 从“1”变为“0”)。然后, 如果检测到开始条件, 就在经过保持时间后总线时钟线下降 (SCLAn=0), 结束通信准备。

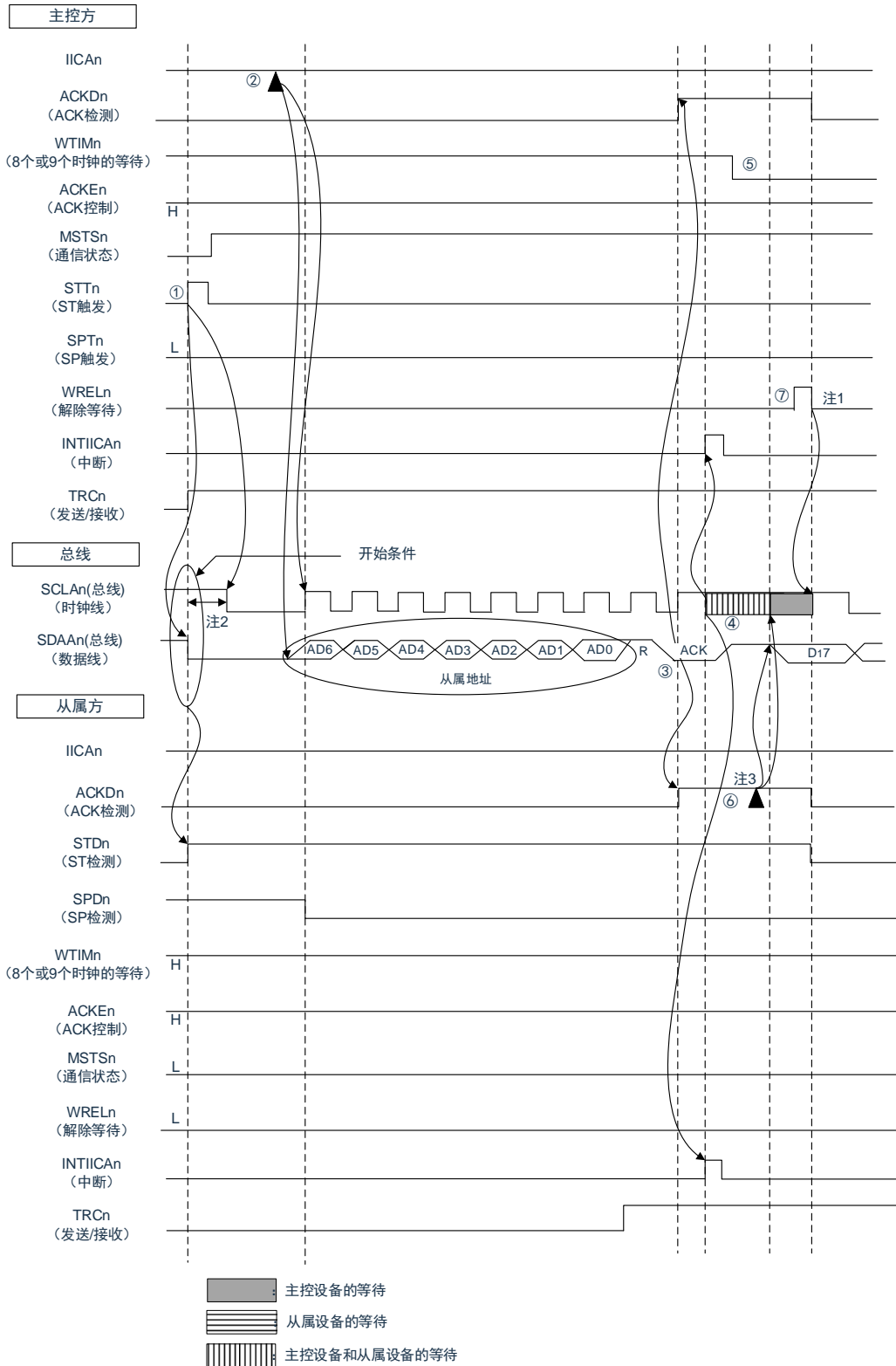
<3>如果主控方给 IICA 移位寄存器 n (IICAn) 写地址+R/W (发送), 就发送从属地址。

备注: n=0

图 14-23: 从属设备&主控设备的通信例子

(主控设备: 选择 8 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (1/3)

(1) 开始条件~地址~数据



注 1: 要解除主控方接收期间的等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2: 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间, 在设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。

注 3: 要解除从属方发送期间的等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

图 14-23 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）<sup>注</sup>。

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）。

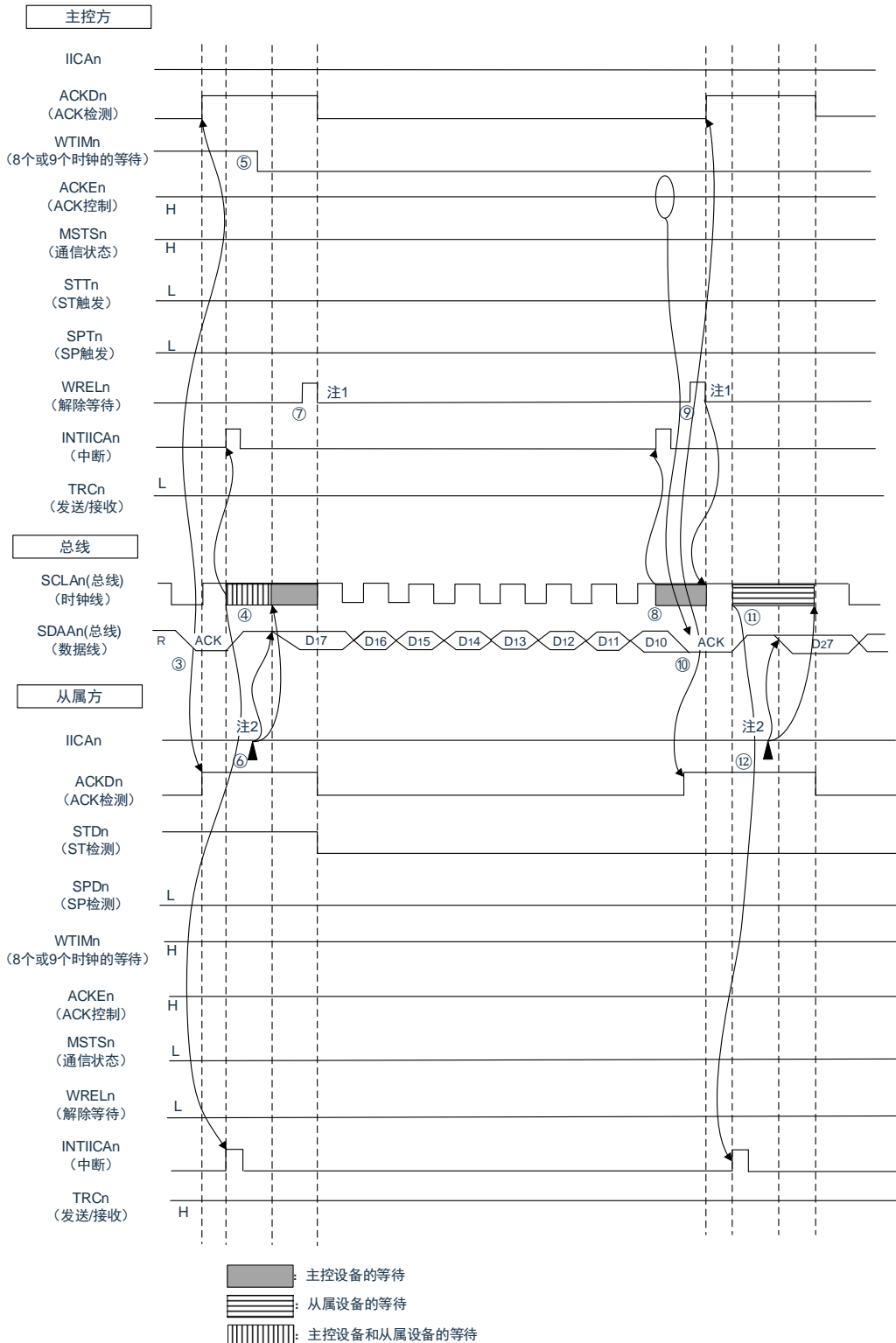
备注：

1. 图 14-23 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 14-23 的“(1)开始条件~地址~数据”说明步骤①~⑦。  
 图 14-23 的“(2)地址~数据~数据”说明步骤③~⑫。  
 图 14-23 的“(3)数据~数据~停止条件”说明步骤⑧~⑱。
2. n=0

图 14-23: 从属设备&主控设备的通信例子

(主控设备: 选择 8 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (2/3)

(2) 地址~数据~数据



注 1: 要解除主控方接收期间的等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2: 要解除从属方发送期间的等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

图 14-23 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<sup>注</sup>，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）<sup>注</sup>。

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 14-24 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 14-24 的“(1)开始条件~地址~数据”说明步骤①~⑦。

图 14-24 的“(2)地址~数据~数据”说明步骤③~⑫。

图 14-24 的“(3)数据~数据~停止条件”说明步骤⑧~⑱。

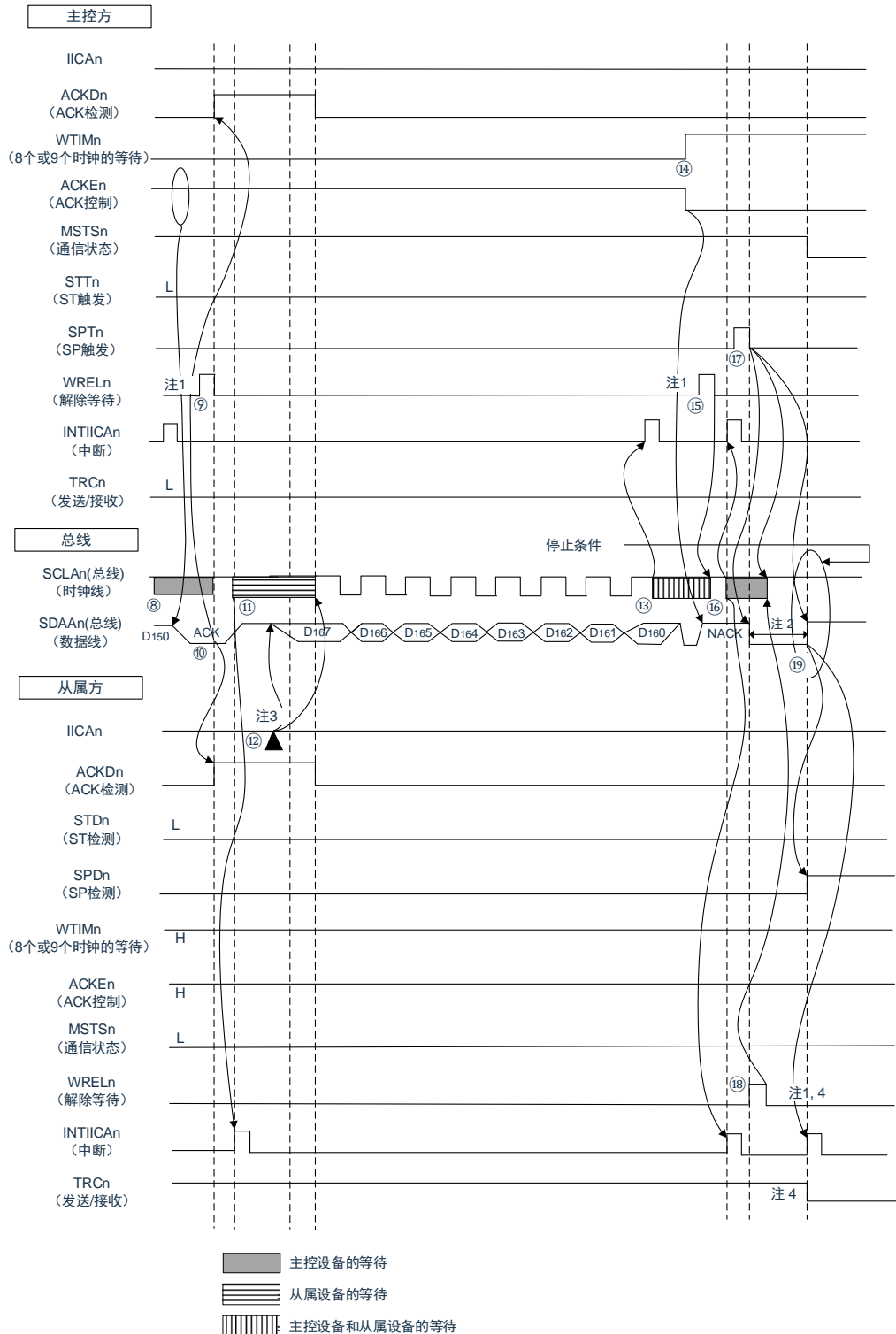
2. n=0



图 14-23：从属设备&主控设备的通信例子

(主控设备：选择 8 个/9 个时钟的等待，从属设备：选择 9 个时钟的等待) (3/3)

(3) 数据~数据~停止条件



注 1: 要解除等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2: 在发行停止条件后, 从 SCLAn 引脚信号上升到生成停止条件的时, 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。

注 3: 要解除从属方发送期间的等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

注 4: 在从属方的发送期间, 如果通过 WRELn 位的置位来解除等待, 就清除 TRCn 位。

图 14-23 的“(3)数据~数据~停止条件”的⑧~⑱的说明如下：

⑧主控方在第 8 个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）。因为主控方的 ACKEn 位为“0”，所以通过硬件给从属方发送 ACK。

⑨主控方读接收数据，解除等待（WRELn=1）。

⑩从属方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

⑪从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）。

⑫如果从属方给 IICA 移位寄存器 n（IICAn）写发送数据，就解除从属方的等待，开始从属方到主控方的数据传送。

⑬主控方在第 8 个时钟的下降沿产生中断（INTIICAn：传送结束中断），并且进入等待状态（SCLAn=0）。因为进行 ACK 控制（ACKEn=1），所以此阶段的总线数据线变为低电平（SDAAn=0）。

⑭主控方设定为 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

⑲（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15章 联动控制器(EVENTC)

## 15.1 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：

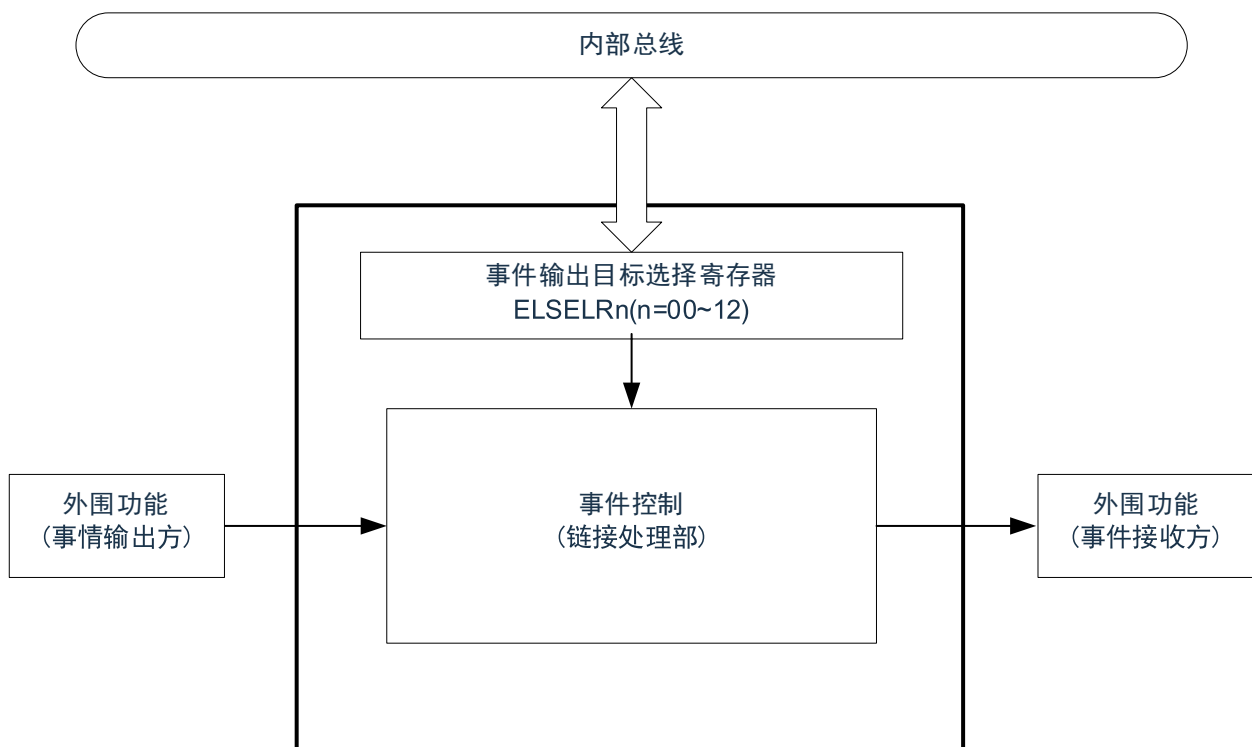
根据产品，能将 13 种外围功能的事件信号直接链接到指定的外围功能。

根据产品，能将事件信号用作 4 种外围功能中的 1 种外围功能运行的启动源。

## 15.2 EVENTC的结构

EVENTC 的框图如图 15-1 所示。

图15-1：EVENTC的框图



## 15.3 控制寄存器

控制器寄存器如表 15-1 所示。

表15-1：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

### 15.3.1 输出目标选择寄存器n (ELSELRn) (n=00~12)

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定为相同的功能。

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定 ELSELRn 寄存器。

ELSELRn 寄存器 (n=00~12) 和外围功能的对应如表 15-3 所示，ELSELRn 寄存器 (n=00~12) 的设定值和链接目标外围功能接受事件时的运行的对应如表 15-4 所示。

表 15-2: 事件输出目标选择寄存器 n (ELSELRn) 的格式

地址: 40043400H (ELSELR00) ~4004340CH (ELSELR12) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	事件链接的选择
0	0	0	禁止事件链接。
0	0	1	选择所链接的外围功能1的运行 <sup>注</sup> 。
0	1	0	选择所链接的外围功能2的运行 <sup>注</sup> 。
0	1	1	选择所链接的外围功能3的运行 <sup>注</sup> 。
1	0	0	选择所链接的外围功能4的运行 <sup>注</sup> 。
其他			设定禁止

注: 请参照“表15-4: ELSELRn寄存器 (n=00~12) 的设定值和链接目标外围功能接受事件时的运行的对应

表15-3: ELSELRn寄存器 (n=00~12) 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	RTC固定周期/闹钟一致检测	INTRTC
ELSELR05	Timer40通道00的计数结束/捕捉结束	INTTM00
ELSELR06	Timer40通道01的计数结束/捕捉结束	INTTM01
ELSELR07	Timer40通道02的计数结束/捕捉结束	INTTM02
ELSELR08	Timer40通道03的计数结束/捕捉结束	INTTM03
ELSELR09	Timer41通道00的计数结束/捕捉结束	INTTM10
ELSELR10	Timer41通道01的计数结束/捕捉结束	INTTM11
ELSELR11	Timer41通道02的计数结束/捕捉结束	INTTM12
ELSELR12	Timer41通道03的计数结束/捕捉结束	INTTM13

表15-4: ELSELRn寄存器 (n=00~12) 的设定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 ELSELn2~ELSELn0位	链接目标 No.	链接目标外围功能	接受事件时的运行
001B	1	A/D转换器	开始A/D转换。
010B	2	Timer40通道0的 定时器输入 <sup>注1</sup>	延迟计数器、输入脉冲间隔的测量、外部事件计数器
011B	3	Timer40通道1的 定时器输入 <sup>注2</sup>	延迟计数器、输入脉冲间隔的测量、外部事件计数器
100B	4	EPWM输出控制的截断源	脉冲输出的强制截止

注1: 要选择Timer40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的运行时钟设定为 $F_{CLK}$ , 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定为联动控制器的事件输入信号。

注2: 要选择Timer40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的运行时钟设定为 $F_{CLK}$ , 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定为EVENTC的事件输入信号。

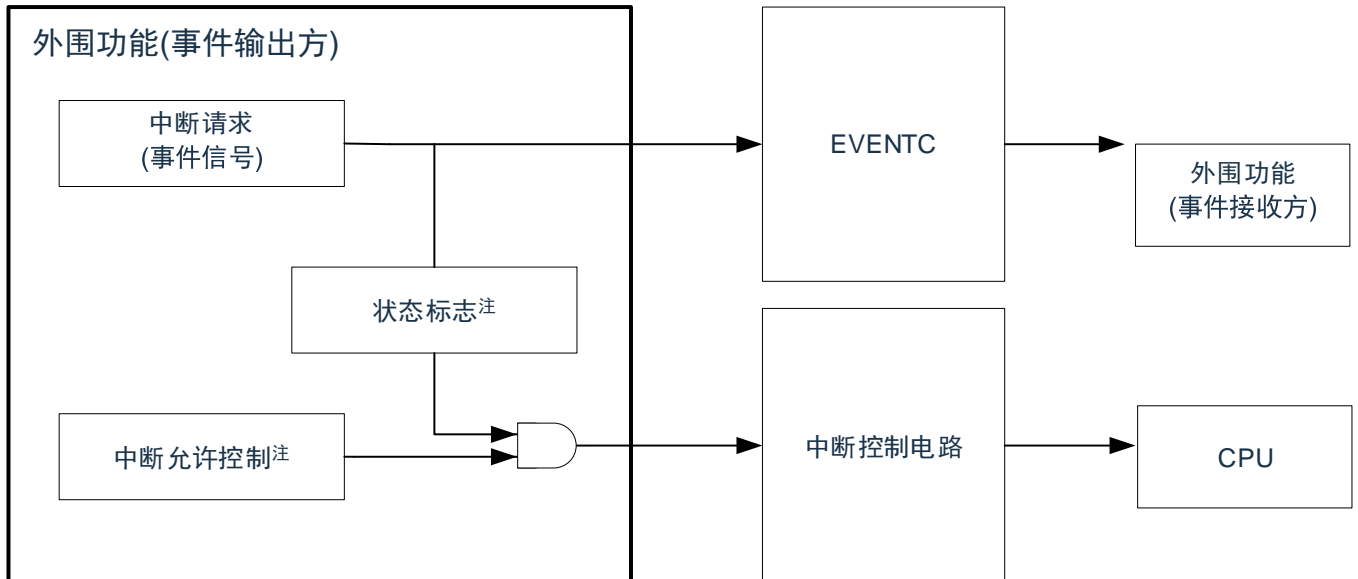
## 15.4 EVENTC的运行

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 15-2 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“（参照“表 15-4：ELSELRn 寄存器（n=00~12）的设定值和链接目标外围功能接受事件时的运行的对应

图 15-2：中断处理和 EVENTC 的关系



接受事件的外围功能的响应如表 15-5 所示。

表15-5：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	Timer40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 <sub>CLK</sub> 周期后进行边沿的检测。
3	Timer40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 <sub>CLK</sub> 周期后进行边沿的检测。
4	EPWM输出控制的截断源	脉冲输出的强制截止	在从发生EVENTC事件经过2个或者3个EPWM的运行时钟周期后变为强制截止状态。

注：有些外围功能没有此功能。

## 第16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。

本系统对32个中断请求(IRQ)输入和1个不可屏蔽中断(NMI)输入的中断源进行了处理。本用户手册只对本系统中的处理进行了说明, Cortex-M0+处理器内置NVIC的功能, 请参考Cortex-M0+处理器的用户手册。

### 16.1 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

这是不受屏蔽控制的中断, 中断请求一旦产生, CPU必须响应。

### 16.2 中断源和结构

中断源列表参照表16-1。



表16-1：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 <sup>注1</sup>
		名称	触发		
可屏蔽	0	INTLVI	电压检测 <sup>注2</sup>	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	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)
	6	INTKR	键中断		
	7	INTST1/ INTSSPI10/ INTIIC10 INTIIC10	UART1发送的传送结束或者缓冲器空中断/SSPI10的传送结束或者缓冲器空中断/IIC10的传送结束		
	8	INTSR1/ INTSSPI11/INTIIC11 INTIIC11	UART1接收的传送结束/SSPI11的传送结束或者缓冲器空中断/IIC11的传送结束		
	9	INTSRE1	发生UART1接收的通信错误		
	10	INTST0/ INTSSPI00/ INTIIC00 INTIIC00	UART0发送的传送结束或者缓冲器空中断/SSPI00的传送结束或者缓冲器空中断/IIC00的传送结束		
	11	INTSR0/ INTSSPI01/INTIIC01	UART0接收的传送结束/SSPI01的传送结束或者缓冲器空中断/IIC01的传送结束		
12	INTSRE0	发生UART0接收的通信错误			

注1：基本结构类型(A)~(C) 分别对应图16-1的(A)~(C)。

注2：这是将电压检测电平寄存器（LVIS）的bit7（LVIMD）置“0”的情况。

表16-1：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	13	INTSPI	串行接口SPI的传送结束中断	内部	(A)
	14	保留			
	15	保留			
	16	INTIICA0	IICA0通信结束		
	17	INTTM00	定时器通道00的计数结束或者捕捉结束		
	18	INTTM01	定时器通道01的计数结束或者捕捉结束		
	19	INTTM02	定时器通道02的计数结束或者捕捉结束		
	20	INTTM03	定时器通道03的计数结束或者捕捉结束		
	21	INTAD	A/D转换结束		
	22	INTRTC	实时时钟的固定周期/ 闹钟一致检测		
	23	INTIT	间隔信号的检测		
	24	INTOCR	高速内部振荡器校准中断		
	25	保留			
	26	保留			
	27	INTTM10	定时器通道10的计数结束或者捕捉结束		
	28	INTTM11	定时器通道11的计数结束或者捕捉结束		
	29	INTTM12	定时器通道12的计数结束或者捕捉结束		
	30	INTTM13	定时器通道13的计数结束或者捕捉结束		
	31	INTFL	FLASH擦除、编程操作完成		

注1：基本构成类型(A)~(C) 分别对应图16-1的(A)~(C)。

表16-1：中断源一览表(3/3)

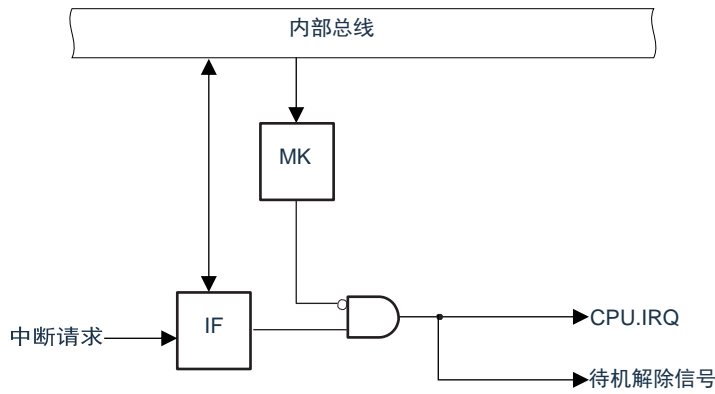
中断处理	中断源编号	中断源		内部/外部	基本结构类型 <sup>注1</sup>
		名称	触发		
不可屏蔽	—	INTWDT	看门狗定时器间隔中断 <sup>注2</sup>	内部	(C)

注1：基本构成类型(A)~(C) 分别对应图16-1的(A)~(C)。

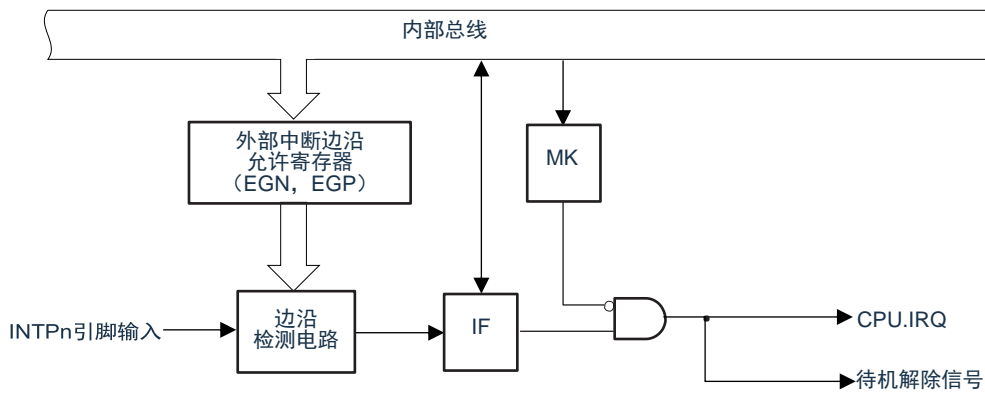
注2：这是将选项字节（000C0H）的bit7（WDTINT）置“1”的情况。

图16-1：中断功能的基本结构

(A)内部可屏蔽中断

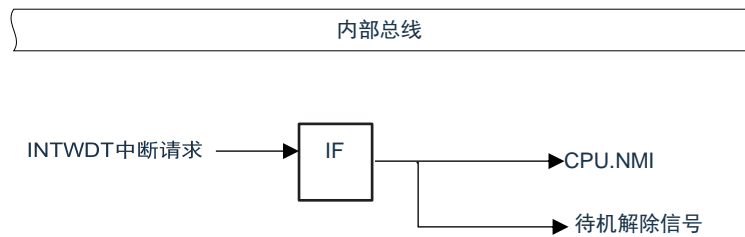


(B)外部可屏蔽中断 (INTPn)



备注：n=0~3。

(C)不可屏蔽中断



备注：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。

## 16.3 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：

- (1) 中断请求标志寄存器（IF00~IF31）
- (2) 中断屏蔽标志寄存器（MK00~MK31）
- (3) 外部中断上升沿允许寄存器（EGP0）
- (4) 外部中断下降沿允许寄存器（EGN0）

### 16.3.1 中断请求标志寄存器（IF00~IF31）

- (1) 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”。
- (2) 通过产生复位信号或者执行指令，将中断请求标志清“0”。
- (3) 可通过 8 位存储器操作指令设定 IF00L~IF31L 寄存器
- (4) 可通过 32 位存储器操作指令设定 IF00~IF31 寄存器。
- (5) 在产生复位信号后，这些寄存器的值变为“0000\_0000H”。

表16-2：中断请求标志寄存器（IFm）的格式（m=0~31）

地址：IF00: 40006000H, IF01: 40006004H, IF02: 40006008H, IF03: 4000600CH  
 IF04: 40006010H, IF05: 40006014H, IF06: 40006018H, IF07: 4000601CH  
 IF08: 40006020H, IF09: 40006024H, IF10: 40006028H, IF11: 4000602CH  
 IF12: 40006030H, IF13: 40006034H, IF14: 40006038H, IF15: 4000603CH  
 IF16: 40006040H, IF17: 40006044H, IF18: 40006048H, IF19: 4000604CH  
 IF20: 40006050H, IF21: 40006054H, IF22: 40006058H, IF23: 4000605CH  
 IF24: 40006060H, IF25: 40006064H, IF26: 40006068H, IF27: 4000606CH  
 IF28: 40006070H, IF29: 40006074H, IF30: 40006078H, IF31: 4000607CH

复位值：0000\_0000H R/W

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

IFmL	编号0~31的中断源的中断请求标志 <sup>注1</sup>
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16-4。

备注：中断请求标志寄存器与CPU.IRQ的对应关系见图16-2。

### 16.3.2 中断屏蔽标志寄存器（MK00~MK31）

- (1) 中断屏蔽标志设定允许或者禁止对应的可屏蔽中断处理。
- (2) 可通过 8 位存储器操作指令设定 MK00L~MK31L 寄存器
- (3) 可通过 32 位存储器操作指令设定 MK00~MK31 寄存器。
- (4) 在产生复位信号后，这些寄存器的值变为“FFFF\_FFFF”。

表16-3：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

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

MKmL	编号0~31的中断源的中断处理控制 <sup>注1</sup>
0	允许中断处理。
1	禁止中断处理。

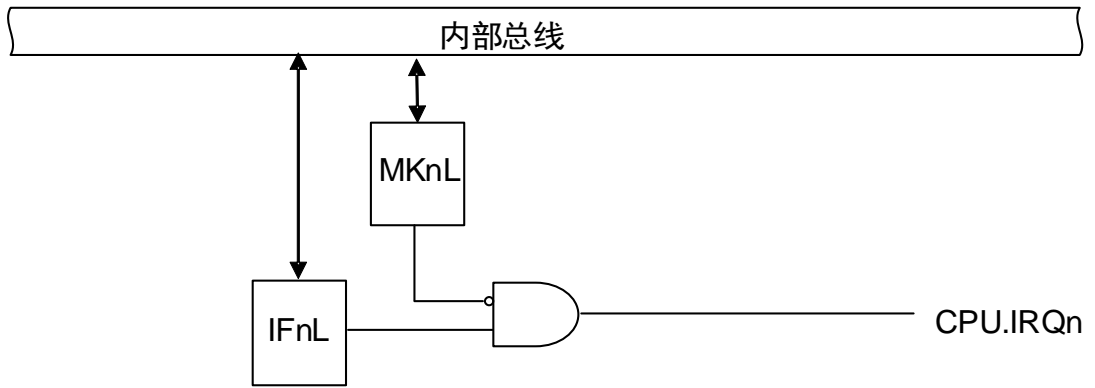
注1：中断源与中断请求屏蔽寄存器的对应关系见表16-4。

备注：中断请求屏蔽寄存器与CPU.IRQ的对应关系见图16-2。

表16-4：中断源与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
0	INTLVI	IF00.IFL	MK00.MKL
1	INTP0	IF01.IFL	MK01.MKL
2	INTP1	IF02.IFL	MK02.MKL
3	INTP2	IF03.IFL	MK03.MKL
4	INTP3	IF04.IFL	MK04.MKL
5	INTTM01H	IF05.IFL	MK05.MKL
6	INTKR	IF06.IFL	MK06.MKL
7	INTST1/INTSSPI10/INTIIC10	IF07.IFL	MK07.MKL
8	INTSR1/INTSSPI11/INTIIC11	IF08.IFL	MK08.MKL
9	INTSRE1	IF09.IFL	MK09.MKL
10	INTST0/INTSSPI00/INTIIC00	IF10.IFL	MK10.MKL
11	INTSR0/INTSSPI01/INTIIC01	IF11.IFL	MK11.MKL
12	INTSRE0	IF12.IFL	MK12.MKL
13	INTSPI	IF13.IFL	MK13.MKL
14		IF14.IFL	MK14.MKL
15		IF15.IFL	MK15.MKL
16	INTIICA0	IF16.IFL	MK16.MKL
17	INTTM00	IF17.IFL	MK17.MKL
18	INTTM01	IF18.IFL	MK18.MKL
19	INTTM02	IF19.IFL	MK19.MKL
20	INTTM03	IF20.IFL	MK20.MKL
21	INTAD	IF21.IFL	MK21.MKL
22	INTRTC	IF22.IFL	MK22.MKL
23	INTIT	IF23.IFL	MK23.MKL
24	INTOCR V	IF24.IFL	MK24.MKL
25		IF25.IFL	MK25.MKL
26		IF26.IFL	MK26.MKL
27	INTTM10	IF27.IFL	MK27.MKL
28	INTTM11	IF28.IFL	MK28.MKL
29	INTTM12	IF29.IFL	MK29.MKL
30	INTTM13	IF30.IFL	MK30.MKL
31	INTFL	IF31.IFL	MK31.MKL

图16-2: 各标志寄存器与CPU.IRQ的关系





### 16.3.3 外部中断上升沿允许寄存器 (EGP0)、外部中断下降沿允许寄存器 (EGN0)

- (1) 这些寄存器设定 INTP0~INTP3 的有效边沿。
- (2) 通过 8 位存储器操作指令设定 EGP0、EGN0 寄存器。
- (3) 在产生复位信号后，这些寄存器的值变为“00H”。

表16-5: 外部中断上升沿允许寄存器 (EGP0) 和外部中断下降沿允许寄存器 (EGN0) 的格式

地址: 40045B38H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	EGP3	EGP2	EGP1	EGP0

地址: 40045B39H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	EGN3	EGN2	EGN1	EGN0

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16-6所示。

表16-6: 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
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3

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=0~3。

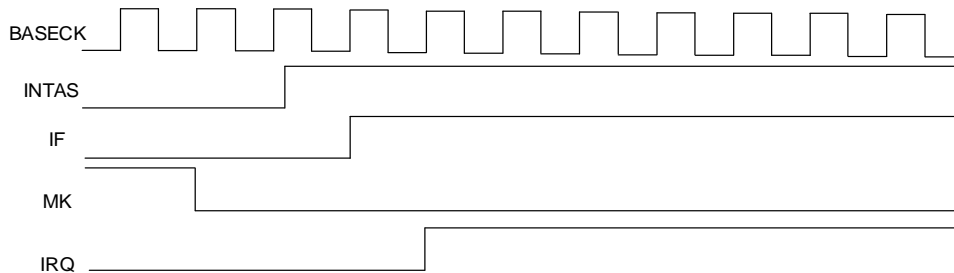
## 16.4 中断处理的操作

### 16.4.1 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。

图16-3：可屏蔽中断请求标志传输至CPU时钟图

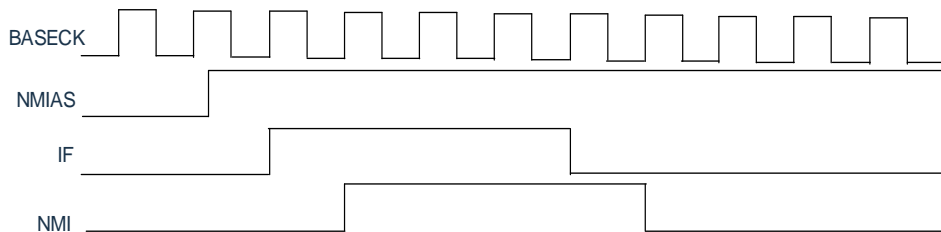


### 16.4.2 不可屏蔽中断请求的接受

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。

图16-4：不可屏蔽中断请求标志传输至CPU时钟图



## 第17章 键中断功能

键中断输入的通道数因产品而不同。

### 17.1 键中断的功能

能通过给键中断输入引脚（KR0~KR7）输入下降沿，产生键中断（INTKR）。

表17-1：键中断检测引脚的分配

键中断引脚	键返回模式寄存器（KRM）
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

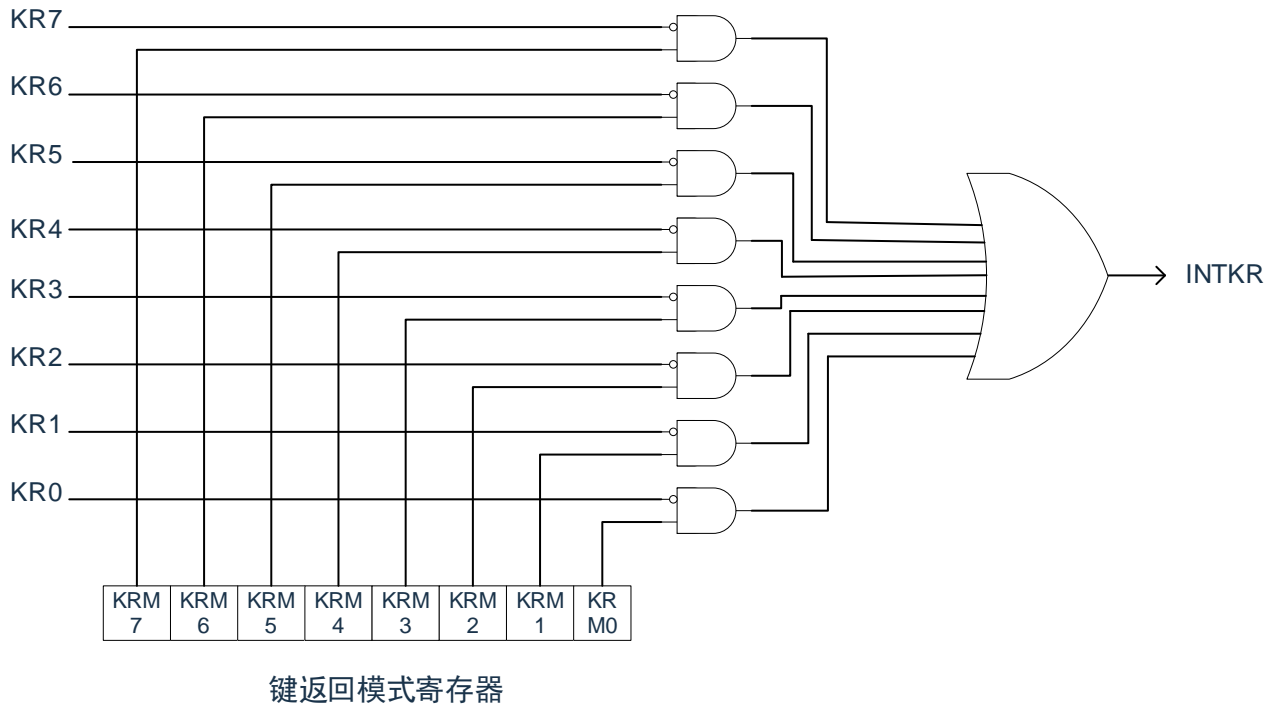
## 17.2 键中断的结构

键中断由以下硬件构成。

表17-2：键中断的结构

项目	控制寄存器
控制寄存器	键返回模式寄存器 (KRM) 端口模式寄存器 (PMx) 端口模式控制寄存器 (PMCx)

图17-1：键中断的框图



## 17.3 控制键中断的寄存器

通过以下寄存器控制键中断功能。

- (1) 键返回模式寄存器 (KRM)
- (2) 端口模式寄存器 (PMx)

### 17.3.1 键返回模式寄存器 (KRM)

- (1) KRM0~KRM7 位控制 KR0~KR7 信号。
- (2) 通过 8 位存储器操作指令设定 KRM 寄存器。
- (3) 在产生复位信号后，此寄存器的值变为“00H”。

表17-3: 键返回模式寄存器 (KRM) 的格式

地址: 40044B37H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	键中断模式的控制
0	不检测键中断信号。
1	检测键中断信号。

注意:

1. 能通过将键中断输入引脚的上拉电阻寄存器 (PUx) 的对象位置“1”，使用内部上拉电阻。
2. 如果在给键中断输入引脚输入低电平的状态下将KRM寄存器的对象位置位，就产生中断。要想忽视此中断时，必须在通过中断屏蔽标志禁止中断处理后设定KRM寄存器。然后，必须在等待键中断输入的低电平宽度 ( $T_{KR}$ ) (请参照数据手册) 后清除中断请求标志，允许中断处理。

备注:

1. 在键中断模式中未使用的引脚能用作通常的端口。
2. n=0~7。

### 17.3.2 端口模式寄存器 (PMx)

- (1) 当用作键中断输入引脚 (KR0~KR7) 时，必须将 PMCxn 位分别置“0”，PMxn 位分别置“1”。此时，Pxn 的输出锁存器可以是“0”或者“1”。
- (2) 通过 8 位存储器操作指令设定 PMx 寄存器。
- (3) 在产生复位信号后，此寄存器的值变为“FFH”。
- (4) 能通过上拉电阻选择寄存器 (PUx) 以位为单位使用内部上拉电阻。
- (5) 端口模式寄存器的格式请参考“2.3.1 端口模式寄存器 (PMxx)”。

# 第18章 待机功能

## 18.1 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。在设定睡眠模式前，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定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，不能设定为深度睡眠模式。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24章 选项字节”。

## 18.2 睡眠模式

### 18.2.1 睡眠模式的设定

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，振荡器等在睡眠模式下的状态见表18-1.

无论设定前的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，都能设定睡眠模式。

备注：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。

表18-1: 睡眠模式中的运行状态(1/2)

睡眠模式的设定		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 <sub>IH</sub> ) 运行	CPU以X1时钟 (F <sub>X</sub> ) 运行	CPU以外部主系统时钟 (F <sub>EX</sub> ) 运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	F <sub>IH</sub>	继续运行 (不能停止)。	禁止运行。	
	F <sub>X</sub>	禁止运行。	继续运行 (不能停止)。	不能运行。
	F <sub>EX</sub>	禁止运行。	不能运行。	继续运行 (不能停止)。
副系统时钟	F <sub>XT</sub>	保持设定为睡眠模式前的状态。		
	F <sub>EXS</sub>			
低速内部振荡器时钟	F <sub>IL</sub>	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设定。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存		停止运行。		
RAM		停止运行。		
端口 (锁存器)		保持设定为睡眠模式前的状态。		
通用定时器单元		能运行。		
实时时钟 (RTC)				
15位间隔定时器				
看门狗定时器		参照“第10章 看门狗定时器”。		
时钟输出/蜂鸣器输出		能运行。		
A/D转换器				
通用串行通信单元 (SCI)				
串行接口 (SPI)				
串行接口 (IICA)				
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			
SFR保护功能		停止运行。		

备注:

1. 停止运行: 在转移到睡眠模式时自动停止运行。
2. 禁止运行: 在转移到睡眠模式前停止运行。
3. F<sub>IH</sub>: 高速内部振荡器时钟 F<sub>IL</sub>: 低速内部振荡器时钟  
 F<sub>X</sub>: X1时钟 F<sub>EX</sub>: 外部主系统时钟  
 F<sub>XT</sub>: XT1时钟 F<sub>EXS</sub>: 外部副系统时钟



表18-1: 睡眠模式中的运行状态(2/2)

睡眠模式的设定		在CPU以副系统时钟运行的过程中执行WFI指令的情况	
		CPU以XT1时钟 (F <sub>XT</sub> ) 运行	CPU以外部副系统时钟 (F <sub>EXS</sub> ) 运行
系统时钟		停止给CPU提供时钟。	
	主系统时钟	F <sub>IH</sub>	禁止运行。
		F <sub>X</sub>	
		F <sub>EX</sub>	
	副系统时钟	F <sub>XT</sub>	继续运行 (不能停止)。
F <sub>EXS</sub>		不能运行。	继续运行 (不能停止)。
低速内部振荡器时钟	F <sub>IL</sub>	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设定。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止	
CPU		停止运行。	
代码闪存		停止运行。	
RAM		停止运行。	
端口 (锁存器)		保持设定为睡眠模式前的状态。	
通用定时器单元		当RTCLPC=0时, 能运行 (否则禁止运行)。	
实时时钟 (RTC)		能运行。	
15位间隔定时器		能运行。	
看门狗定时器		参照“第10章 看门狗定时器”。	
时钟输出/蜂鸣器输出		当RTCLPC=0时, 能运行 (否则禁止运行)。	
A/D转换器		禁止运行。	
通用串行通信单元 (SCI)		当RTCLPC=0时, 能运行 (否则禁止运行)。	
串行接口 (SPI)		当RTCLPC=0时, 能运行 (否则禁止运行)。	
串行接口 (IICA)		当RTCLPC=0时, 能运行 (否则禁止运行)。	
联动控制器		能在可运行的功能块之间进行链接。	
上电复位功能		能运行。	
电压检测功能		能运行。	
外部中断		能运行。	
CRC运算功能	高速CRC	禁止运行。	
	通用CRC	禁止运行。	
SFR保护功能		禁止运行。	

注1: 停止运行: 在转移到睡眠模式时自动停止运行。

注2: 禁止运行: 在转移到睡眠模式前停止运行。

注3: F<sub>IH</sub>: 高速内部振荡器时钟 F<sub>IL</sub>: 低速内部振荡器时钟

F<sub>X</sub>: X1时钟 F<sub>EX</sub>: 外部主系统时钟

F<sub>XT</sub>: XT1时钟 F<sub>EXS</sub>: 外部副系统时钟

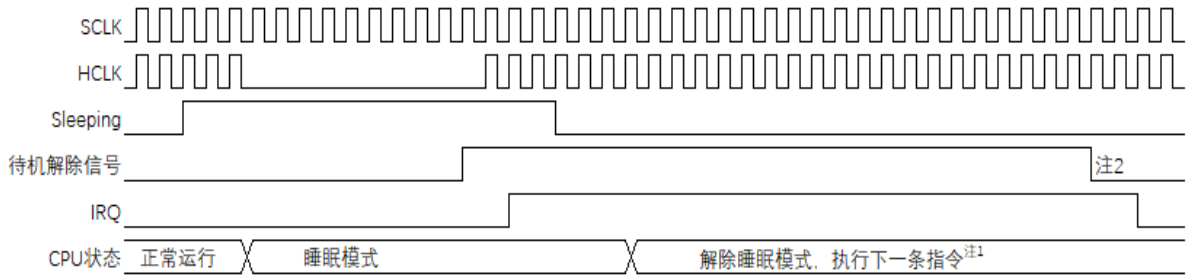
## 18.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位，POR复位，低电压检测复位，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18-1：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。

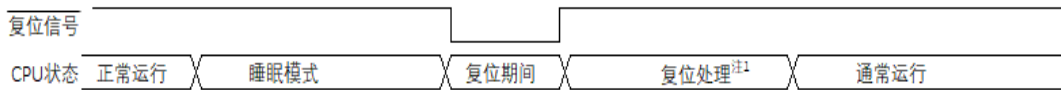
注2：待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

备注：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18-2：通过复位解除睡眠模式



注1：有关复位处理，请参照“第19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，请参照“第20章 上电复位电路”。

## 18.3 深度睡眠模式

### 18.3.1 深度睡眠模式的设定

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，以及振荡器都停止运行。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，振荡器在深度睡眠模式的运行状态见表18-2。

只有在设定前的CPU时钟为主系统时钟的情况下才能设定深度睡眠模式。

备注：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18-2: 深度睡眠模式中的运行状态

深度睡眠模式的设定		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 <sub>IH</sub> ) 运行	CPU以X1时钟 (F <sub>X</sub> ) 运行	CPU以外部主系统时钟 (F <sub>EX</sub> ) 运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	F <sub>IH</sub>	停止		
	F <sub>X</sub>			
	F <sub>EX</sub>			
副系统时钟	F <sub>XT</sub>	保持设定为深度睡眠模式前的状态。		
	F <sub>EXS</sub>			
低速内部振荡器时钟	F <sub>IL</sub>	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设定。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存				
RAM				
端口 (锁存器)		保持设定为深度睡眠模式前的状态。		
通用定时器单元		禁止运行。		
实时时钟 (RTC)		能运行。		
15位间隔定时器				
看门狗定时器		参照“第10章 看门狗定时器”。		
时钟输出/蜂鸣器输出		在选择副系统时钟作为计数时钟并且RTCLPC位为“0”时, 能运行 (否则禁止运行)。		
A/D转换器		能进行唤醒。		
通用串行通信单元 (SCI)		禁止运行。		
串行接口 (SPI)		禁止运行。		
串行接口 (IICA)		能通过地址匹配进行唤醒。		
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		停止运行。		
CRC运算功能	高速CRC			
	通用CRC			
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。
2. 禁止运行: 在转移到深度睡眠模式前停止运行。
3. F<sub>IH</sub>: 高速内部振荡器时钟 F<sub>IL</sub>: 低速内部振荡器时钟  
 F<sub>X</sub>: X1时钟 F<sub>EX</sub>: 外部主系统时钟  
 F<sub>XT</sub>: XT1时钟 F<sub>EXS</sub>: 外部副系统时钟

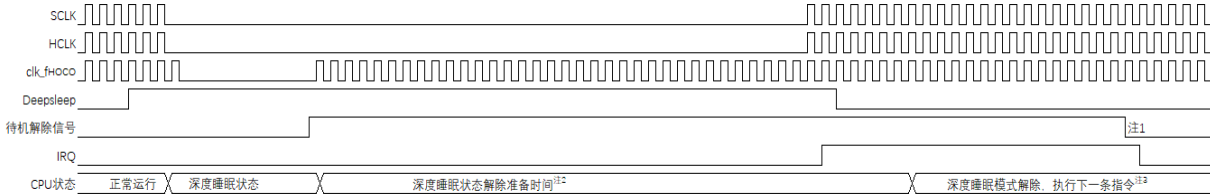
### 18.3.2 深度睡眠模式的解除

能通过以下2种方法解除深度睡眠模式。

(1) 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18-3：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16-1 中断功能的基本结构”。

注2：深度睡眠状态解除准备时间：

- ① 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。
- ② 进入深度睡眠模式前 CPU 时钟为高速系统时钟(X1 振荡)时：至少 20us 与振荡稳定时间(通过 OSTS 进行设定)中较长的时间”。

注3：等待：从CPU.IRQ有效到开始执行中断服务程序，需要14个时钟。

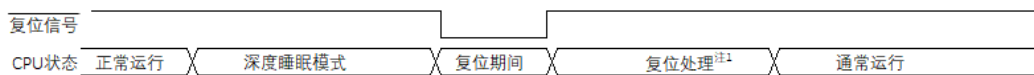
注意：

- 1. 进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。
- 2. 在CPU以高速系统时钟（X1振荡）运行并且要缩短深度睡眠模式解除后的振荡稳定时间时，必须在执行WFI指令前暂时将CPU时钟切换为高速内部振荡器时钟。
- 3. 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。

(2) 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18-4：通过复位解除深度睡眠模式



注1：有关复位处理，请参照“第19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，请参照“第20章 上电复位电路”和“第21章 电压检测电路”。

## 18.4 部分掉电的深度睡眠模式

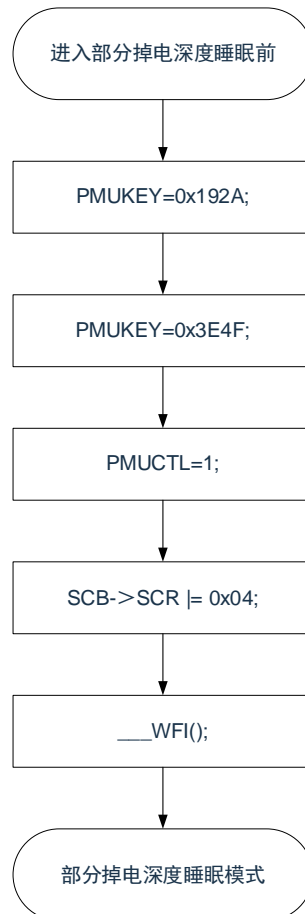
### 18.4.1 部分掉电的深度睡眠模式的设定

部分掉电的深度睡眠模式是在深度睡眠模式的基础上关闭部分周边的电源供电而进一步节省电源消耗的深度睡眠模式。进入部分掉电的深度睡眠模式需要配置PMUCTL寄存器的PVDNEN位，该控制位的写入受供电模式控制保护寄存器（PMUKEY）的保护，解除部分掉电的深度睡眠模式时要求重新初始化掉电的周边后才能重新正常运行，需要重新初始化的周边模块详情请参考表23-3 部分掉电的深度睡眠模式中的运行状态。

在SCR寄存器的SLEEPDEEP位为1并且PMUCTL寄存器的PVDNEN位也为1时，执行WFI指令就可进入部分掉电的深度睡眠模式。在这个模式，CPU，以及发振器都停止运行，大多数的周边模块会被关闭电源供给。但是，CPU内部寄存器的值，RAM数据，I/O的状态会被保持。周边模块，发振器在部分掉电的深度睡眠模式的运行状态见表23-3。

只有在设定前的CPU时钟为主系统时钟的情况下才能设定部分掉电的深度睡眠模式。PMUCTL寄存器的PVDNEN位的控制参考4.3.11供电模式控制保护寄存器（PMUKEY）和4.3.12 供电模式控制寄存器（PMUCTL）部分。

图18-5：进入部分掉电的深度睡眠模式流程图



备注：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，此种情况下不会进入部分掉电模式。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18-3: 部分掉电的深度睡眠模式中的运行状态

项目		部分掉电深度睡眠模式的设定	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 <sub>IH</sub> ) 运行	CPU以X1时钟 (F <sub>X</sub> ) 运行	CPU以外部主系统时钟 (F <sub>EX</sub> ) 运行
系统时钟			停止给CPU提供时钟、保持供电。		
	主系统时钟	F <sub>IH</sub>	停止运行		
		F <sub>X</sub>			
		F <sub>EX</sub>			
	副系统时钟	F <sub>XT</sub>	保持设定为部分掉电深度睡眠模式前的状态。		
		F <sub>EXS</sub>			
	低速内部振荡器时钟	F <sub>IL</sub>	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设定。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU			停止运行、保持供电。		
代码闪存			RAM保持供电。		
RAM			禁止运行、停止供电。		
端口 (锁存器)			能运行, 保持供电。		
通用定时器单元			参照“第10章 看门狗定时器”, 保持供电。		
实时时钟 (RTC)			禁止运行、停止供电。		
15位间隔定时器			禁止运行、停止供电。		
看门狗定时器			禁止运行、停止供电。		
时钟输出/蜂鸣器输出			禁止运行、停止供电。		
A/D转换器			禁止运行、停止供电。		
通用串行通信单元 (SCI)			禁止运行、停止供电。		
串行接口 (SPI)			禁止运行、停止供电。		
串行接口 (IICA)			禁止运行、停止供电。		
联动控制器			禁止运行、停止供电。		
上电复位功能			能运行、保持供电。		
电压检测功能					
外部中断					
CRC运算功能	高速CRC		停止运行、停止供电。		
	通用CRC				
SFR保护功能					

## 备注:

1. 停止运行: 在转移到部分掉电的深度睡眠模式时自动停止运行。
2. 禁止运行: 在转移到部分掉电的深度睡眠模式前停止运行。
3. 保持供电: 转移到部分掉电的深度睡眠模式后维持模块电源供给。
4. 停止供电: 转移到部分掉电的深度睡眠模式后停止模块的电源供应, 解除模式后需要重新初始化模块。
5. F<sub>IH</sub>: 高速内部振荡器时钟 F<sub>IL</sub>: 低速内部振荡器时钟  
 F<sub>X</sub>: X1时钟 F<sub>EX</sub>: 外部主系统时钟  
 F<sub>XT</sub>: XT1时钟 F<sub>EXS</sub>: 外部副系统时钟

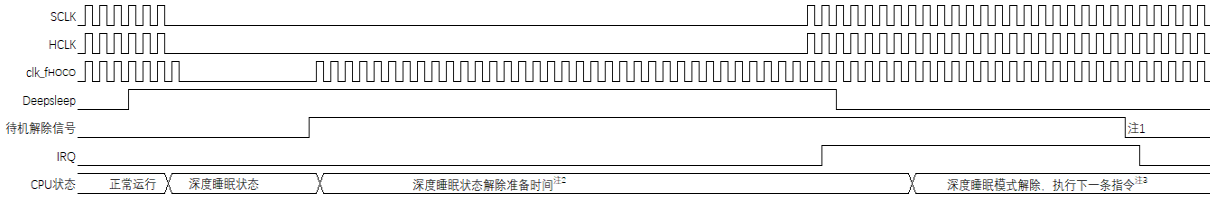
### 18.4.2 部分掉电的深度睡眠模式的解除

能通过以下2种方法解除部分掉电的深度睡眠模式。

(1) 通过中断请求进行部分掉电的深度睡眠的解除

如果产生INTP0-3、INTKR、INTRTC、INTIT、INTLVI及INTWDT中断请求，可解除部分掉电的深度睡眠模式。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18-6：通过中断请求解除深度睡眠模式

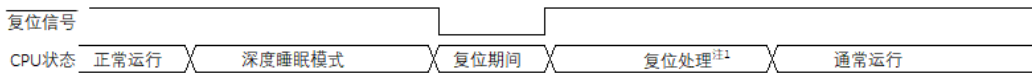


- 注1：待机解除信号：INTP0-3、INTKR、INTRTC、INTIT、INTLVI以及INTWDT的中断请求信号。
- 注2：部分掉电的深度睡眠状态解除准备，需要重新初始化周边功能，才能保证程序继续正常运行。
- 注3：进入部分掉电的深度睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。

(2) 通过产生复位信号进行的解除

通过产生复位信号来解除部分掉电的深度睡眠模式。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18-7：通过复位解除部分掉电的深度睡眠模式



- 注1：有关复位处理，请参照“第19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，请参照“第章 20上电复位电路”和“第21章 电压检测电路”



## 第19章 复位功能

以下6种方法产生复位信号。

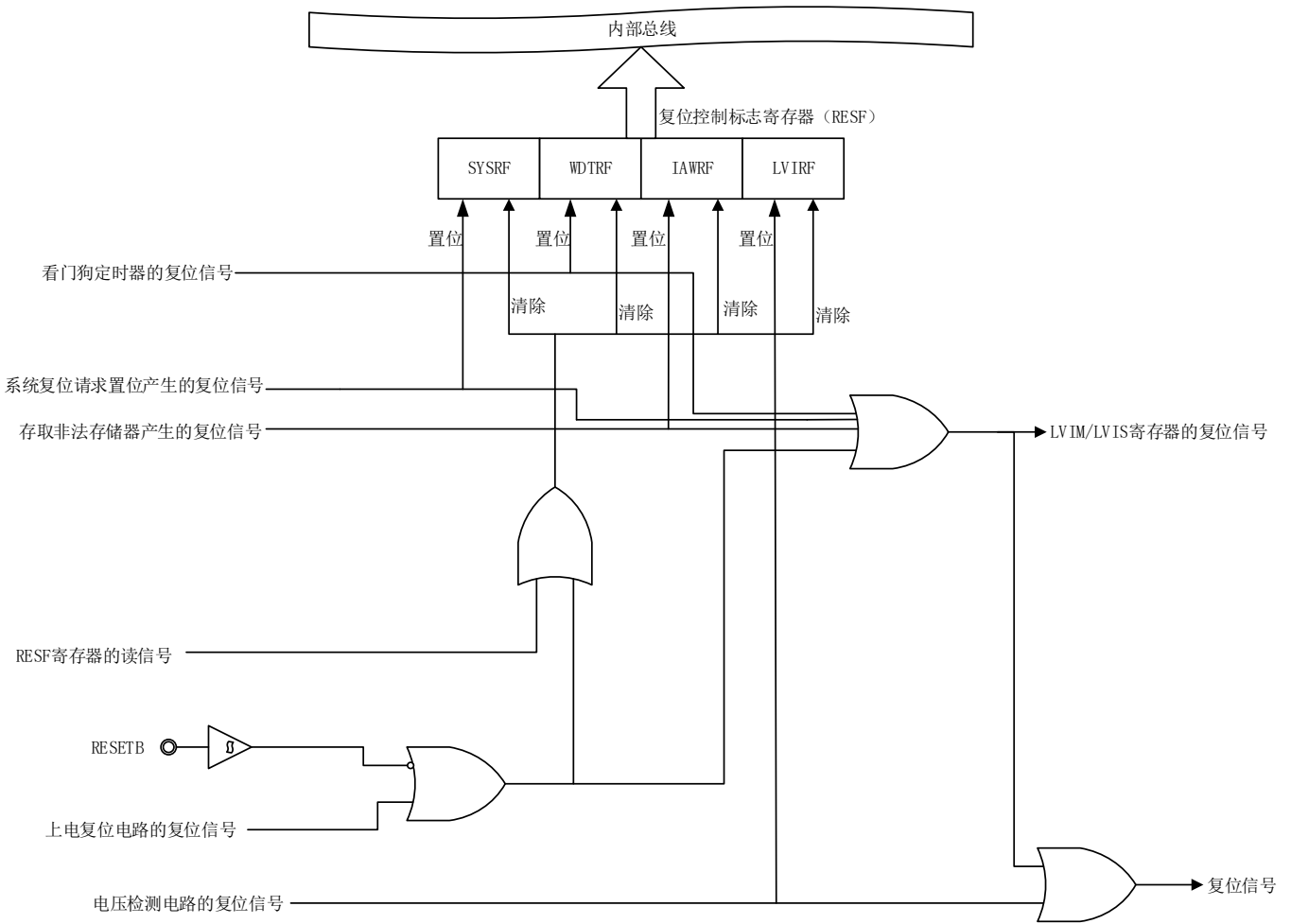
- (1) 通过 RESETB 引脚输入外部复位。
- (2) 通过看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5) 因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为 1 而产生内部复位。
- (6) 因存取非法存储器而产生内部复位。

内部复位和外部复位相同，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，或者存取非法存储器时，产生复位并且各硬件变为如表19-1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，而且在用户手册的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。
2. 在复位信号发生期间，停止X1时钟、XT1时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟和外部副系统时钟的输入无效。
3. 如果发生复位，就对各SFR进行初始化，因此端口引脚变为以下状态：
  - ① P00：在外部复位或者 POR 复位的期间为低电平,其他复位期间为和正常运行时为高电平。
  - ② P20,P21,P36,P37：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)。
  - ③ 其他端口：在复位期间以及接受复位后为高阻抗。

图19-1：复位功能的框图

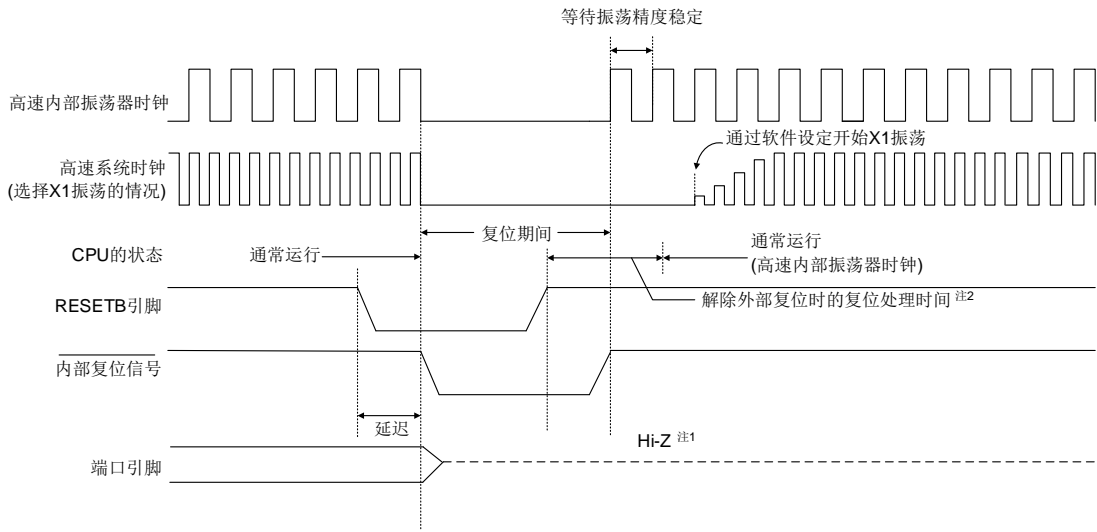


备注：

1. LVD电路的内部复位不会对LVD电路进行复位。
2. LVIM：电压检测寄存器。
3. LVIS：电压检测电平寄存器。

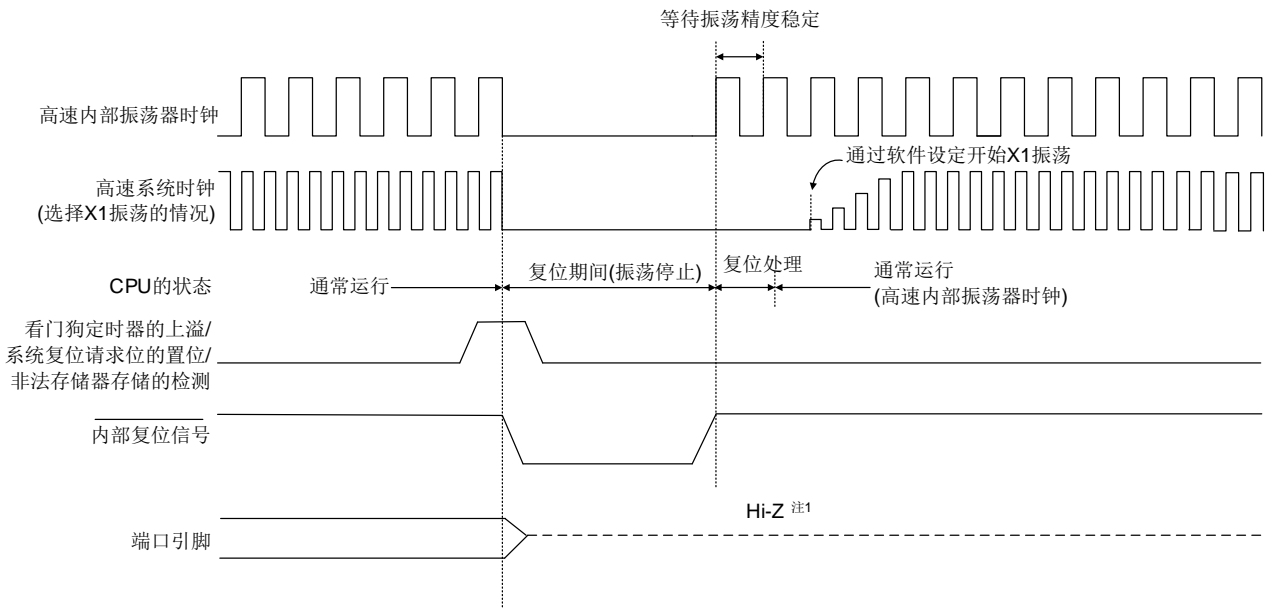
当给RESETB引脚输入低电平时，产生复位。然后，如果给RESETB引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图19-2: 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或者非法存储器存取检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图19-3: 因看门狗定时器的上溢、系统复位请求位的置位或者非法存储器存取的检测而产生的复位时序



注1: 端口引脚P00,P20,P20,P36,P37变为以下状态:

- ① P00: 在外部复位或者 POR 复位的期间为低电平,其他复位期间为和正常运行时为高电平。
- ② P20,P21,P36,P37: 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)。

备注: 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20章 上电复位电路”和“第21章 电压检测电路”。

备注：

1.  $V_{POR}$ ：POR电源电压上升检测电压。
2.  $V_{LVD}$ ：LVD检测电压。

表19-1：复位期间的运行状态

项目		复位期间
系统时钟		停止给CPU提供时钟。
主系统时钟	$F_{IH}$	停止运行。
	$F_X$	停止运行（X1引脚和X2引脚处于输入端口模式）。
	$F_{EX}$	时钟输入无效（引脚处于输入端口模式）。
副系统时钟	$F_{XT}$	能运行。
	$F_{EXS}$	时钟输入无效（引脚处于输入端口模式）。
低速内部振荡器时钟	$F_{IL}$	停止运行。
CPU		
代码闪存		停止运行。
RAM		停止运行。
端口（锁存器）		高阻抗 <sup>注1</sup>
通用定时器单元		停止运行。
实时时钟（RTC）		
15位间隔定时器		
看门狗定时器		
时钟输出/蜂鸣器输出		
A/D转换器		
通用串行通信单元（SCI）		
串行接口（IICA）		
上电复位功能		能进行检测运行。
电压检测功能		在LVD复位时，能运行。在其他复位时，停止运行。
外部中断		停止运行。
键中断功能		
CRC 运算功能	高速CRC	
	通用CRC	
SFR保护功能		

注1：端口引脚P00,P20,P20,P36,P37变为以下状态：

- ① P00：在外部复位或者POR复位的期间为低电平,其他复位期间为和正常运行时为高电平。
- ② P20,P21,P36,P37：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。

备注： $F_{IH}$ ：高速内部振荡器时钟  $F_{IL}$ ：低速内部振荡器时钟

$F_X$ ：X1时钟  $F_{EX}$ ：外部主系统时钟

$F_{XT}$ ：XT1时钟  $F_{EXS}$ ：外部副系统时钟

## 19.1 确认复位源的寄存器

### 19.1.1 复位控制标志寄存器（RESF）

CMS32L032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、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RAM，然后通过其RAM值进行判断。

表19-2：复位控制标志寄存器（RESF）的格式

地址：40020440H	复位后：不定值 <sup>注</sup>							R
符号	7	6	5	4	3	2	1	0
RESF	SYSRF	0	0	WDTRF	0	0	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。

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：因复位源而不同。请参照表19-3。

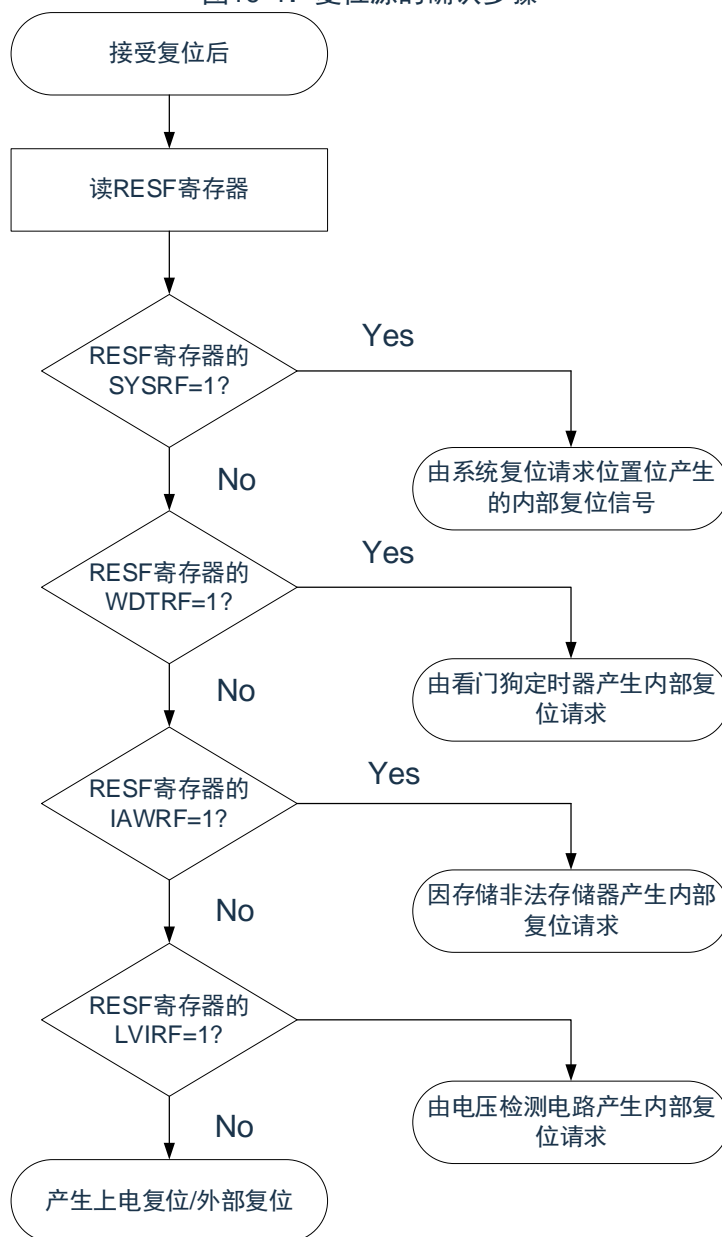
发生复位请求时的RESF寄存器状态如表19-3所示。

表19-3：发生复位请求时的RESF寄存器状态

标志 \ 复位源	RESETB输入	POR产生复位	系统复位请求位置位产生的复位	WDT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
WDTRF			保持	置“1”		
IAWRF				置“1”		
LVIRF			保持			

复位源的确认步骤如图19-4所示。

图19-4：复位源的确认步骤



备注：上述流程是确认步骤的例子。

## 第20章 上电复位电路

### 20.1 上电复位电路的功能

上电复位电路（POR）有以下功能：

- (1) 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。
- (2) 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。

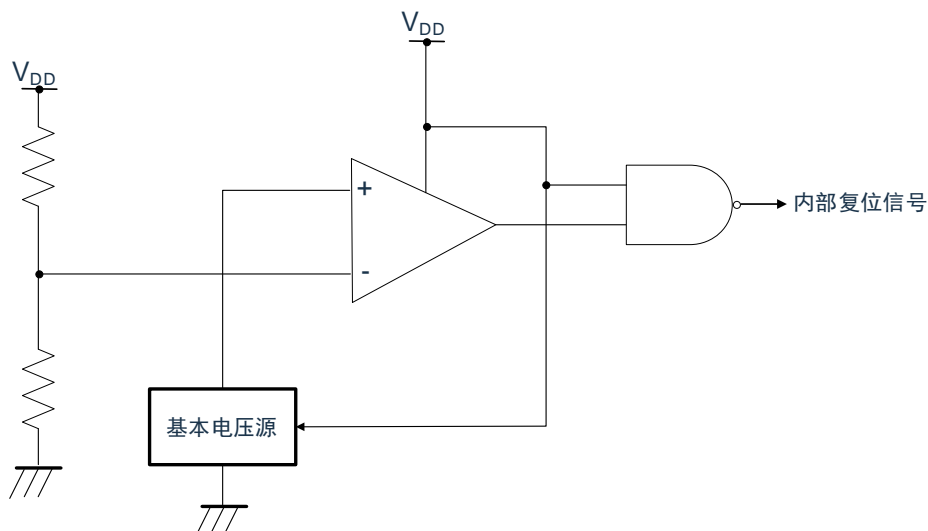
备注：

1. 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”。
2. CMS32L032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”。RESF寄存器的详细内容，请参照“第19章 复位功能”。
3.  $V_{POR}$ ：POR电源电压上升检测电压； $V_{PDR}$ ：POR电源电压下降检测电压。
4. 详细内容请参照数据手册的POR电路特性。

### 20.2 上电复位电路的结构

上电复位电路的结构如图20-1所示。

图20-1：上电复位电路的结构

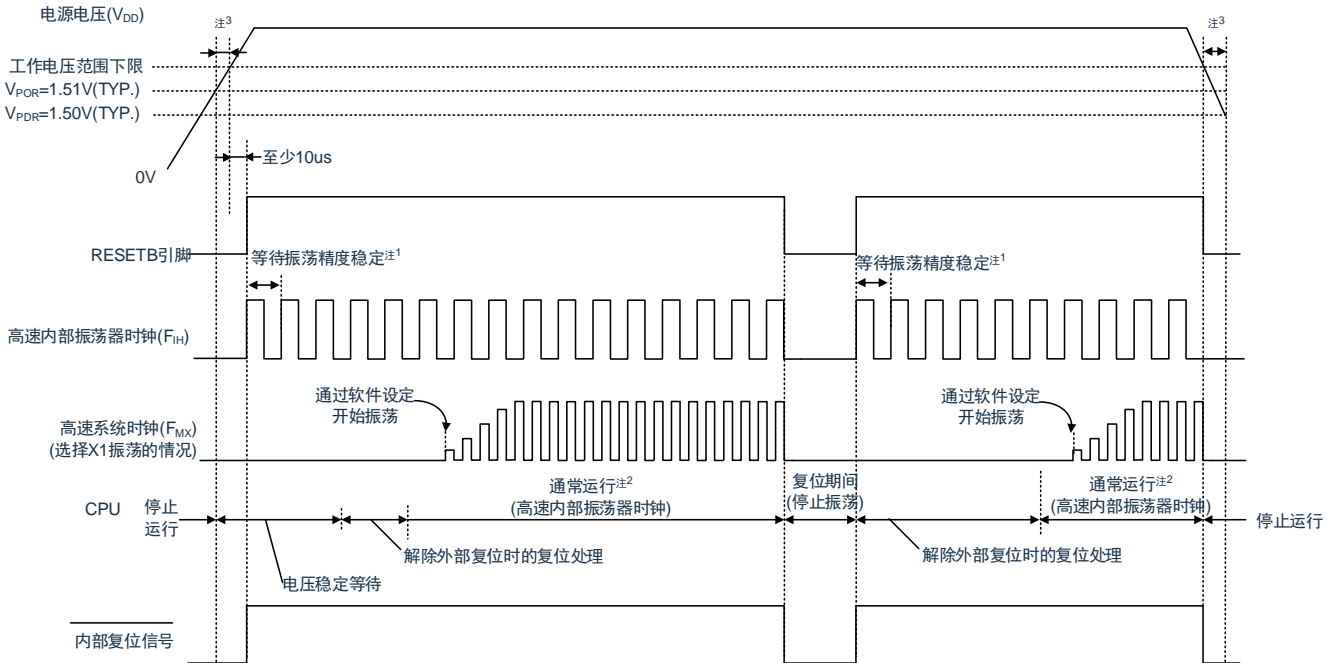


## 20.3 上电复位电路的运行

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20-2：上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用 RESETB 引脚的外部复位输入的情况



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注3：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21章 电压检测电路”。

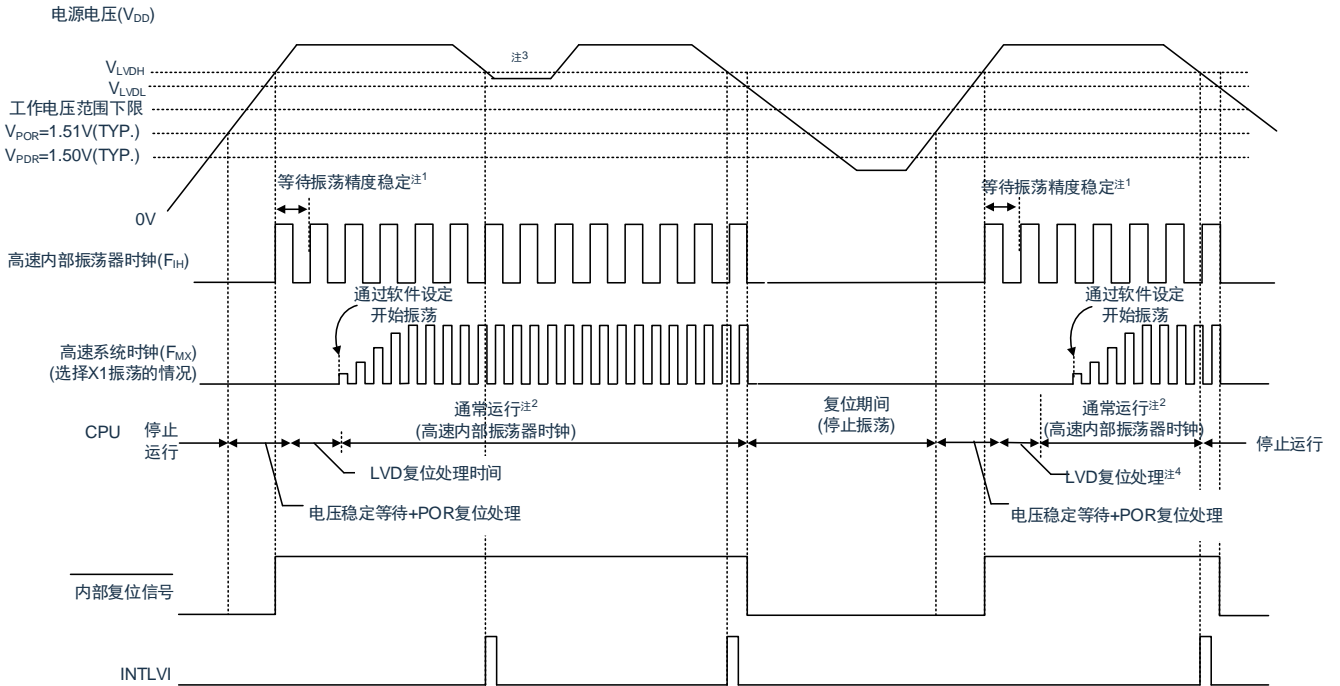
注意：当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注：V<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20-2：上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD 为中断&复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、0）



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注3：在产生中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的LVILV位和LVIMD位置“1”。因此，必须考虑可能出现电源电压在不低于低电压检测电压（ $V_{LVDL}$ ）的状态下恢复到高电压检测电压（ $V_{LVDH}$ ）或者更高的情况，在产生INTLVI后按照“图21-5 工作电压的确认/复位的设定步骤”和“图21-6 中断&复位模式的初始设定步骤”进行设定。

注4：到开始通常运行为止的时间除了达到VPOR（1.51V(典型值)）后的“电压稳定等待+POR复位处理”以外，在达到LVD检测电平（ $V_{LVD}$ ）后还需要“LVD复位处理”。

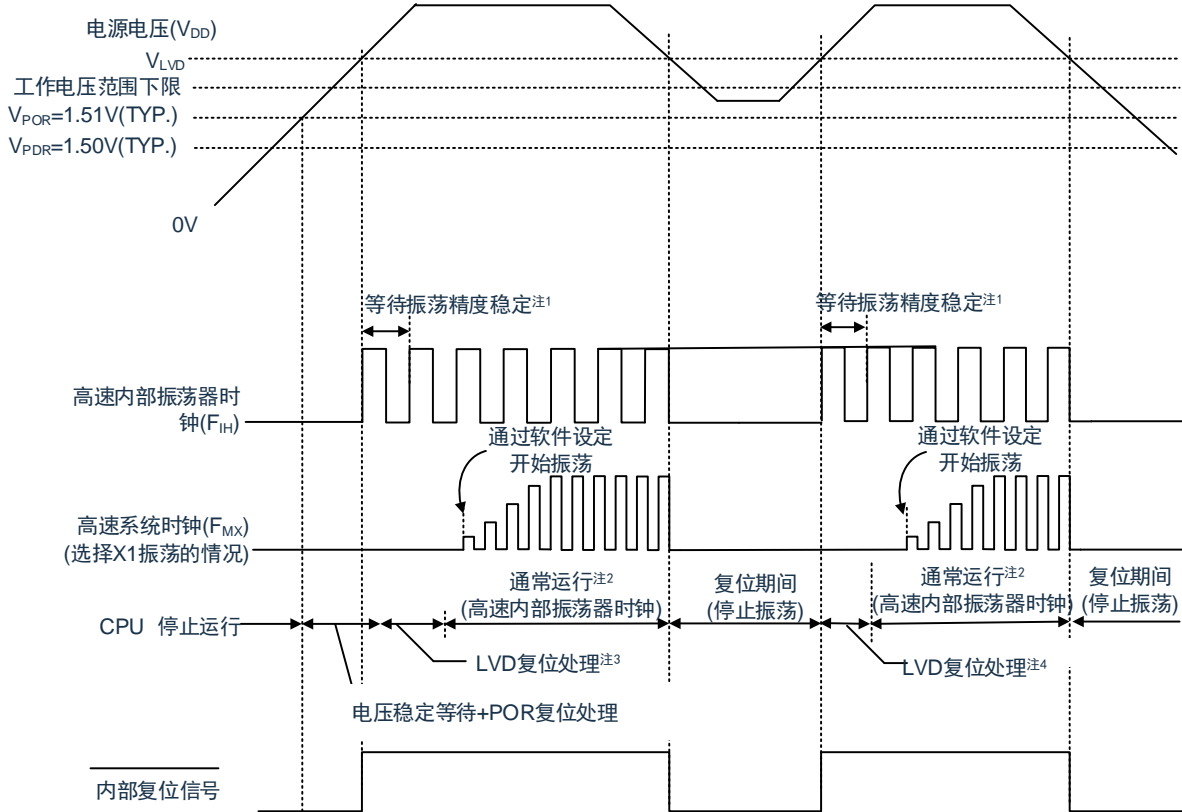
备注： $V_{LVDH}$ 、 $V_{LVDL}$ ：LVD检测电压。

$V_{POR}$ ：POR电源电压上升检测电压。

$V_{PDR}$ ：POR电源电压下降检测电压。

图20-2：上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3) LVD 复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、1）



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注3：到开始通常运行为止的时间除了达到 $V_{POR}$ （1.51V(典型值)）后的“电压稳定等待+POR复位处理”以外，在达到LVD检测电平（ $V_{LVD}$ ）后还需要“LVD复位处理”。

注4：在电源电压下降时，如果只在发生电压检测电路（LVD）的内部复位后恢复电源电压，就在达到LVD检测电平（ $V_{LVD}$ ）后需要“LVD复位处理”。

备注：

1.  $V_{LVDH}$ 、 $V_{LVDL}$ ：LVD检测电压  
 $V_{POR}$ ：POR电源上升检测电压  
 $V_{PDR}$ ：POR电源下降检测电压
2. 当选择LVD中断模式（选项字节000C1H的LVIMD1、LVIMD0=0、1）时，从接通电源到开始通常运行的时间和“图20-2(3/3)LVD复位模式的情况”的“注3”的时间相同。

# 第21章 电压检测电路

## 21.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设定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：

- (1) 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。
- (2) 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 24 章 选项字节”）。
- (3) 能在深度睡眠模式中运行
- (4) 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）的设定。

电压检测电路有以下3种运行模式设定：

- (1) 中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）  
选项字节000C1H可选择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。
- (2) 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。
- (3) 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。

表21-1：电压检测电路的模式

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）。

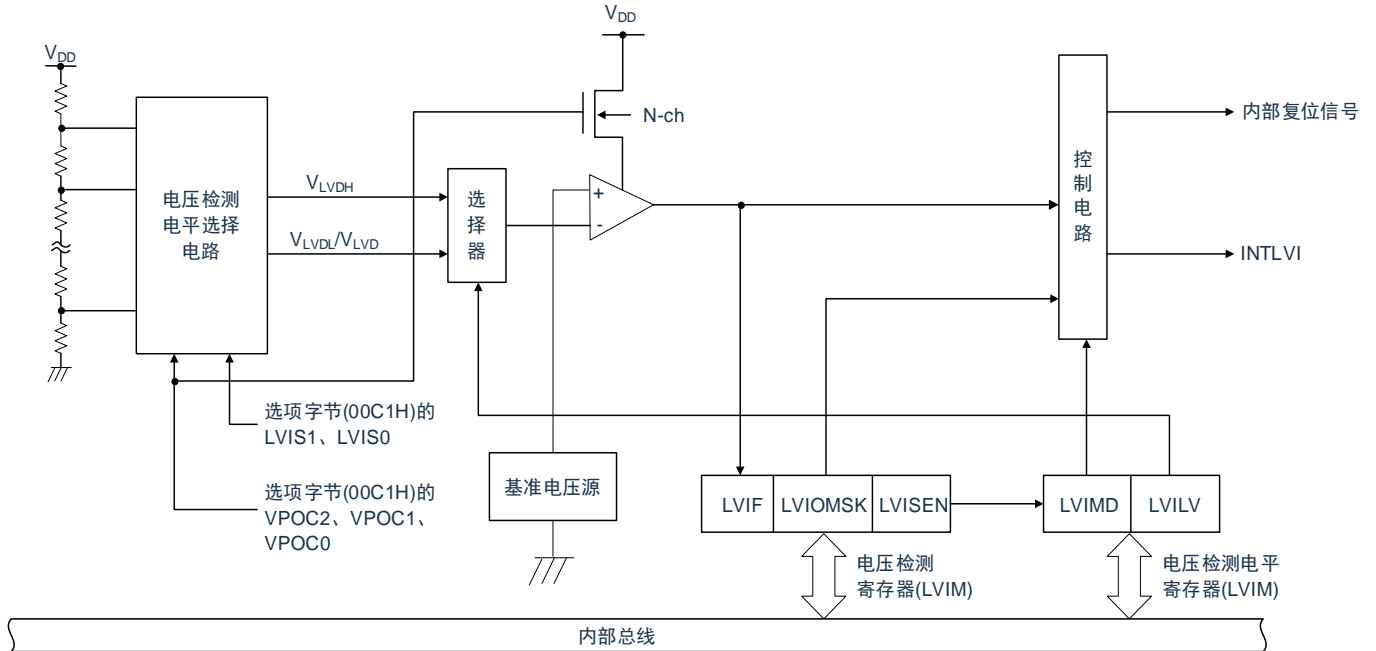
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

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19章 复位功能”。

## 21.2 电压检测电路的结构

电压检测电路的框图如图21-2所示。

图21-2：电压检测电路的框图



## 21.3 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：

- (1) 电压检测寄存器（LVIM）
- (2) 电压检测电平寄存器（LVIS）

### 21.3.1 电压检测寄存器 (LVIM)

此寄存器设定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定LVIM寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表21-3: 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>						LVIOMSK	LVIF

LVISEN <sup>注3</sup>	电压检测电平寄存器 (LVIS) 的允许/禁止改写的设定
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 <sub>DD</sub> ) ≥ 检测电压 (V <sub>LVD</sub> ) 或者LVD为OFF。
1	电源电压 (V <sub>DD</sub> ) < 检测电压 (V <sub>LVD</sub> )。

注1: 复位值会因复位源而改变。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；发生其他复位时，将LVISEN清“0”。

注2: LVIM寄存器的bit0和bit1是只读位。

注3: 只有在选择中断&复位模式 (选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”) 时才能设定，在其他模式中不能更改初始值。

注4: 只有在选择中断&复位模式 (选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”) 时，LVIOMSK位才在以下期间自动变为“1”，屏蔽LVD产生的复位或者中断。

- ① LVISEN=1 的期间。
- ② 从发生 LVD 中断开始到 LVD 检测电压稳定为止的等待时间。
- ③ 从更改 LVILV 位 (LVIS 寄存器的 bit0) 的值到 LVD 检测电压稳定为止的等待时间。

## 21.3.2 电压检测电平寄存器 (LVIS)

这是设定电压检测电平的寄存器。通过8位存储器操作指令设定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>。

表21-4：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	LVILV

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 (VLVDH)
1	低电压检测电平 (VLVDL或者VLVD)

注1：复位值因复位源和选项字节的设定而变。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：

- ① 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
- ② 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
- ③ 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H

注2：只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能写“0”。在其他情况下不能设定。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：要改写LVIS寄存器时，必须按照图21-5和图21-6的步骤进行。

备注：通过选项字节000C1H选择LVD的运行模式和各模式的检测电压 (VLVDH、VLVDL、VLVD)。用户选项字节 (000C1H/010C1H) 的格式如表21-4所示。有关选项字节的详细内容，请参照“第24章 选项字节”。

表21-4：用户选项字节（000C1H/010C1H）的格式(1/2)

 地址：000C1H/010C1H<sup>注</sup>

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

## (1) LVD 的设定（中断&amp;复位模式）

检测电压			选项字节的设定值						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
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		
—			禁止设定上述以外的值。						

## (2) LVD 的设定 (复位模式)

检测电压		选项字节的设定值						
V <sub>LVD</sub>		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：检测电压是典型值。详细内容请参照数据手册的LVD电路特性。



表21-4: 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

## (1) LVD 的设定 (中断模式)

检测电压		选项字节的设定值						
$V_{LVD}$		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。						

## (2) LVD 为 OFF (使用 RESETB 引脚的外部复位)

检测电压		选项字节的设定值						
$V_{LVD}$	$V_{LVD}$	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	上升						LVIMDS1	LVIMDS1
—	—	1	×	×	×	×	×	×
—		禁止设定上述以外的值。						

## 注意:

1. 中断模式必须给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) 的设定。

## 备注:

3. ×: 忽略。
4. 检测电压是典型值。详细内容请参照数据手册的LVD电路特性。

## 21.4 电压检测电路的运行

### 21.4.1 用作复位模式时的设定

通过选项字节000C1H设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1））和检测电压（ $V_{LVD}$ ）。如果设定复位模式，就在以下初始设定的状态下开始运行：

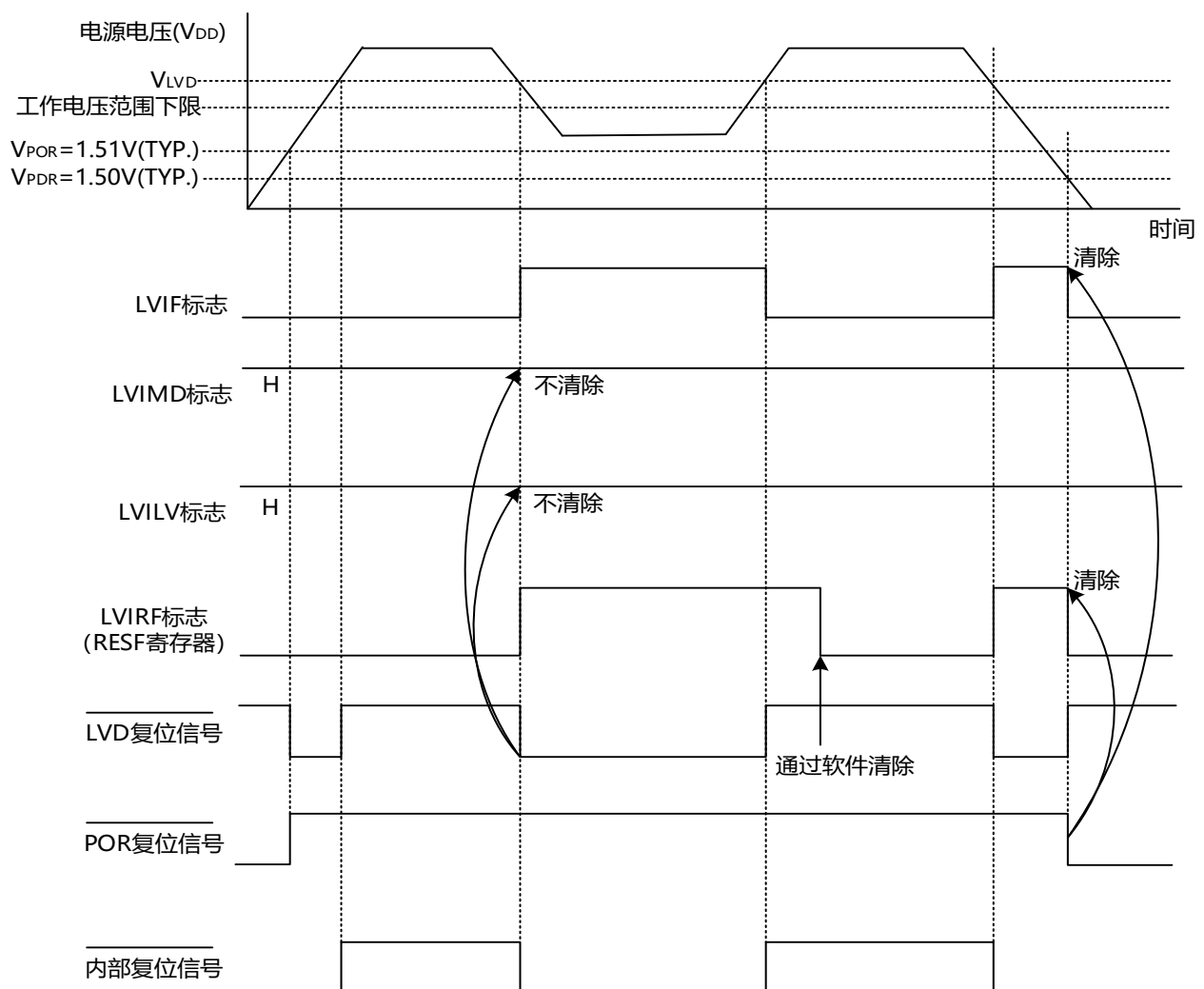
- (1) 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- (2) 将电压检测电平寄存器（LVIS）的初始值置“81H”。设置 bit7（LVIMD）为“1”（复位模式）。bit0（LVILV）为“1”（电压检测电平： $V_{LVD}$ ）。

LVD复位模式的运行：

当接通电源时，复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持LVD的内部复位状态。若电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），解除内部复位。当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于电压检测电平（ $V_{LVD}$ ），就产生LVD的内部复位。

LVD复位模式的内部复位信号的产生时序如图21-2所示。

图21-2：内部复位信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、1）



备注： $V_{POR}$ ：POR电源电压上升检测电压； $V_{PDR}$ ：POR电源电压下降检测电压。

## 21.4.2 用作中断模式时的设定

通过选项字节000C1H设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1））和检测电压（ $V_{LVD}$ ）。如果设定中断模式，就在以下初始设定的状态下开始运行：

- (1) 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））。
- (2) 将电压检测电平寄存器（LVIS）的初始值置“01H”。bit7（LVIMD）为“0”（中断模式）bit0（LVILV）为“1”（电压检测电平： $V_{LVD}$ ）。

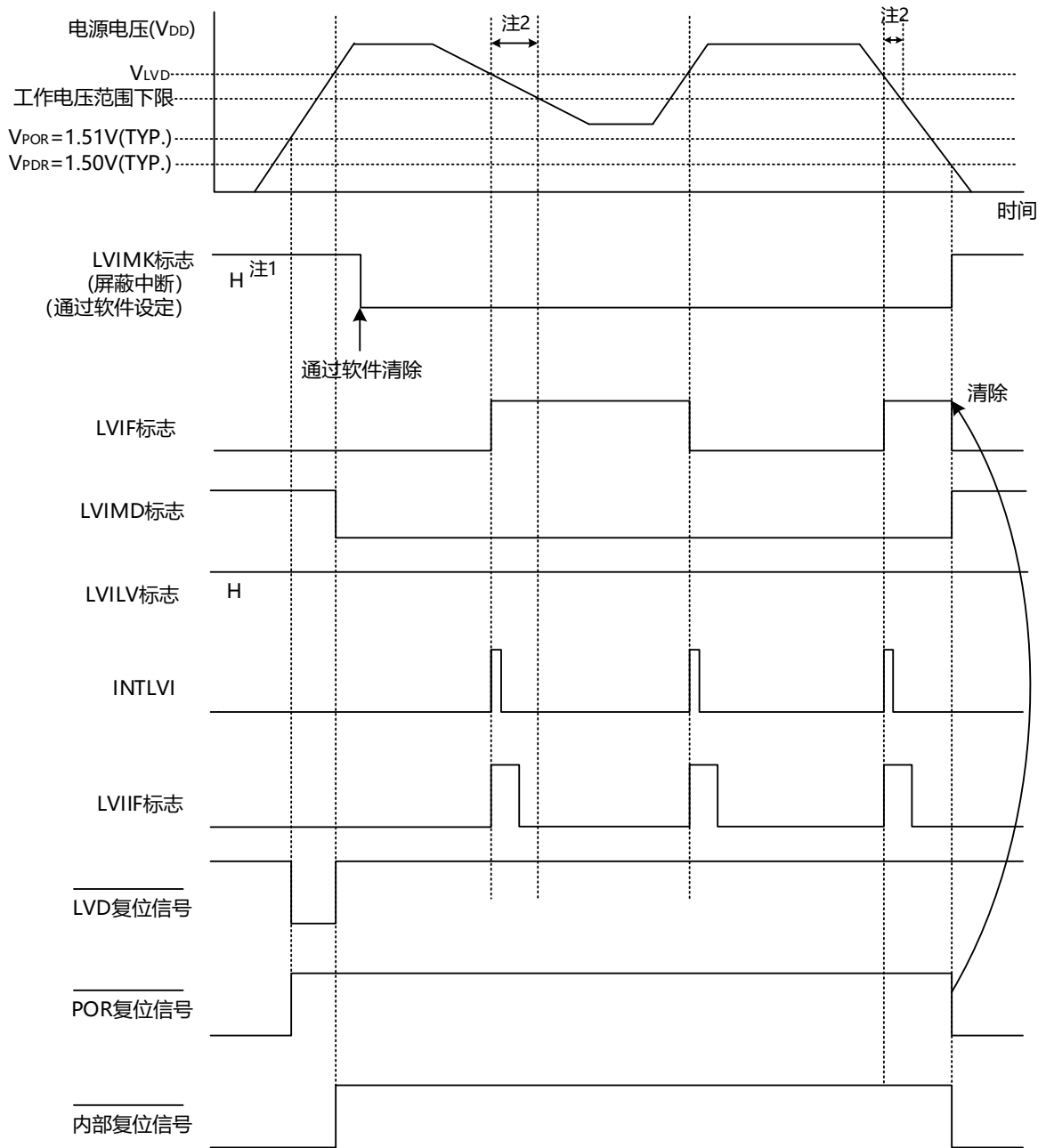
LVD中断模式的运行：

在产生复位后，中断模式（选项字节的LVIMDS1、LVIMDS0=0、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持LVD的内部复位状态。如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就解除LVD的内部复位。

在解除LVD的内部复位后，如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就产生LVD的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD中断模式的中断请求信号的产生时序如图21-3所示。

图21-3: 中断信号的产生时序 (选项字节的LVIMDS1、LVIMDS0=0、1)



注1: 在产生复位信号后, LVIMK标志变为“1”。

注2: 当工作电压下降时, 须在工作电压低于数据手册的AC特性所示的工作电压范围前, 通过深度睡眠模式的转移或者外部复位, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

备注:  $V_{POR}$ : POR电源电压上升检测电压;  $V_{PDR}$ : POR电源电压下降检测电压。

### 21.4.3 用作中断&复位模式时的设定

通过选项字节000C1H设定运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0））和检测电压值（VLVDH、VLVDL）。如果设定中断&复位模式，就在以下初始设定的状态下开始运行：

- (1) 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））。
- (2) 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：VLVDH）。

LVD中断&复位模式的运行：

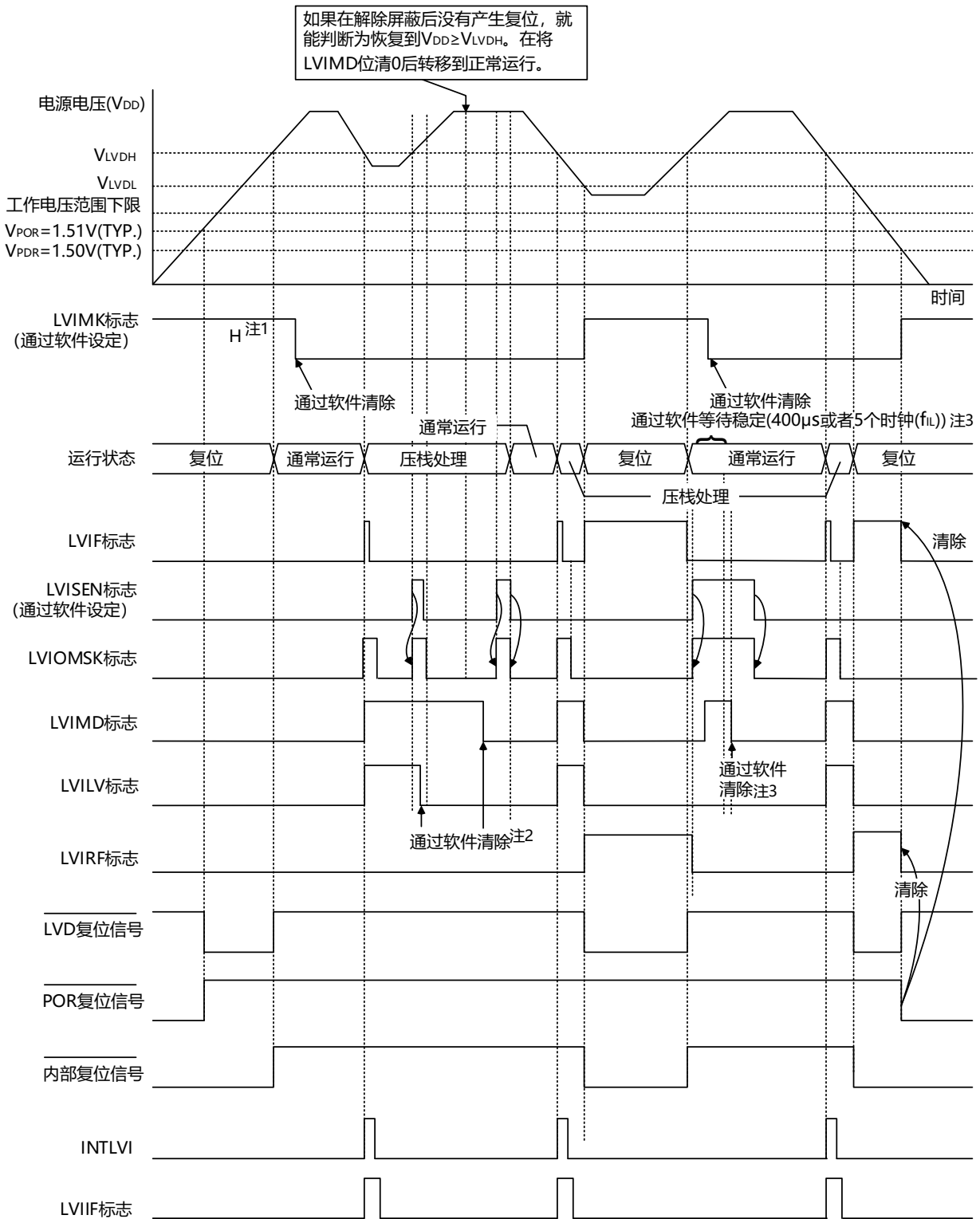
当接通电源时，设定的中断&复位模式（选项字节的LVIMDS1、LVIMDS0=1、0）在电源电压（V<sub>DD</sub>）低于低电压检测电平（VLVDL）前保持LVD的内部复位状态。如果电源电压（V<sub>DD</sub>）超过高电压检测电平（VLVDH），就解除内部复位。

当工作电压下降时，如果电源电压（V<sub>DD</sub>）低于高电压检测电平（VLVDH），就产生LVD的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（V<sub>DD</sub>）低于低电压检测电平（VLVDL），就产生LVD的内部复位。但是，在发生INTLVI后，即使电源电压（V<sub>DD</sub>）在不低于低电压检测电压（VLVDL）的状态下恢复到高电压检测电压（VLVDH）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21-5：工作电压的确认/复位的设定步骤和图21-6：中断&复位模式的初始设定步骤”所示的流程图的步骤进行设定。

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21-4示。

图21-4：复位&中断信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、0）(1/2)



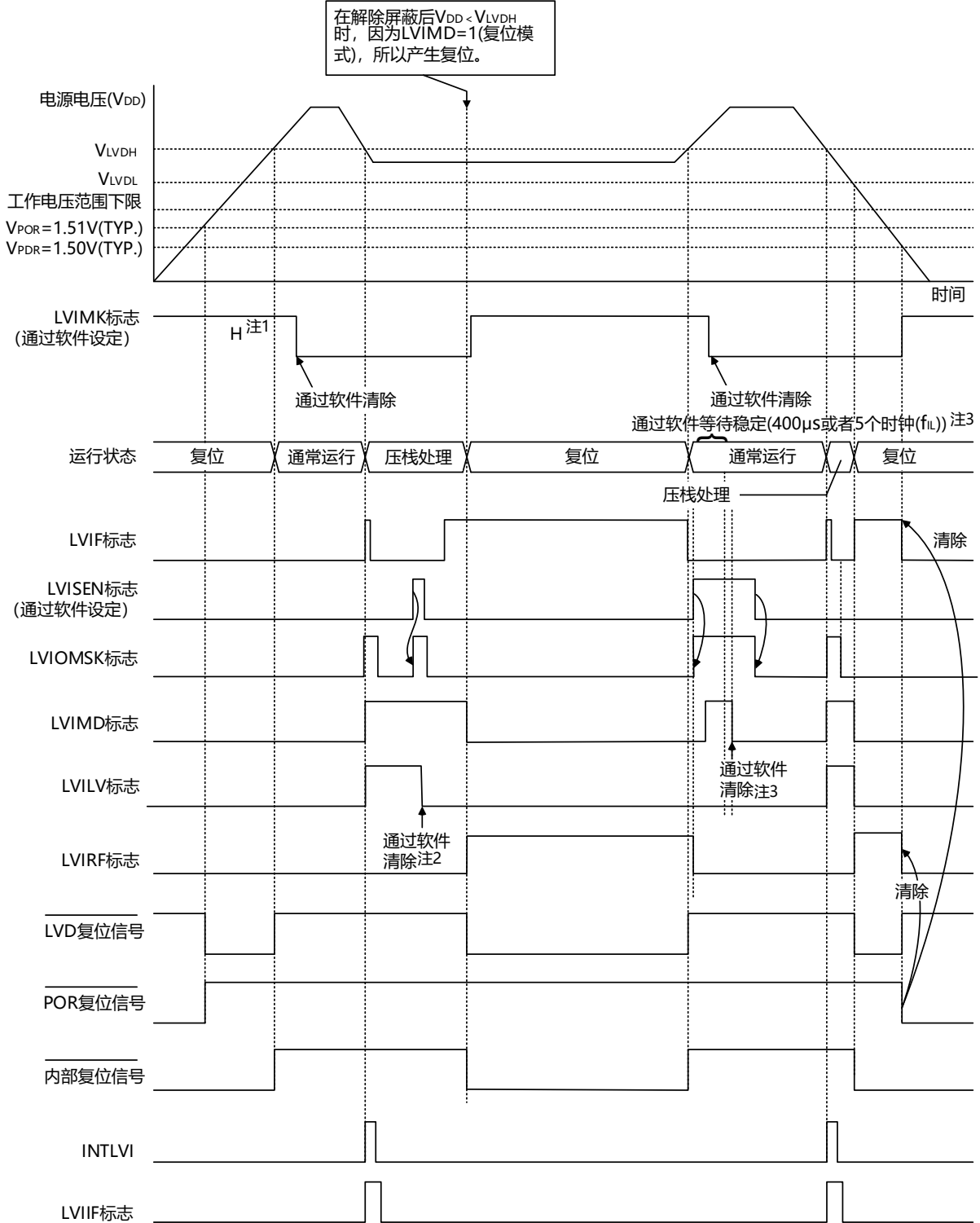
注1：在产生复位信号后，LVIMK标志变为“1”。

注2：当使用中断&复位模式时，必须在发生中断后按照“图21-5：工作电压的确认/复位的设定步骤”进行设定。

注3：当使用中断&复位模式时，必须在解除复位后按照“图21-6：中断&复位模式的初始设定步骤”进行设定。

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21-4: 中断&复位信号的产生时序 (选项字节的LVIMDS1、LVIMDS0=1、0) (2/2)



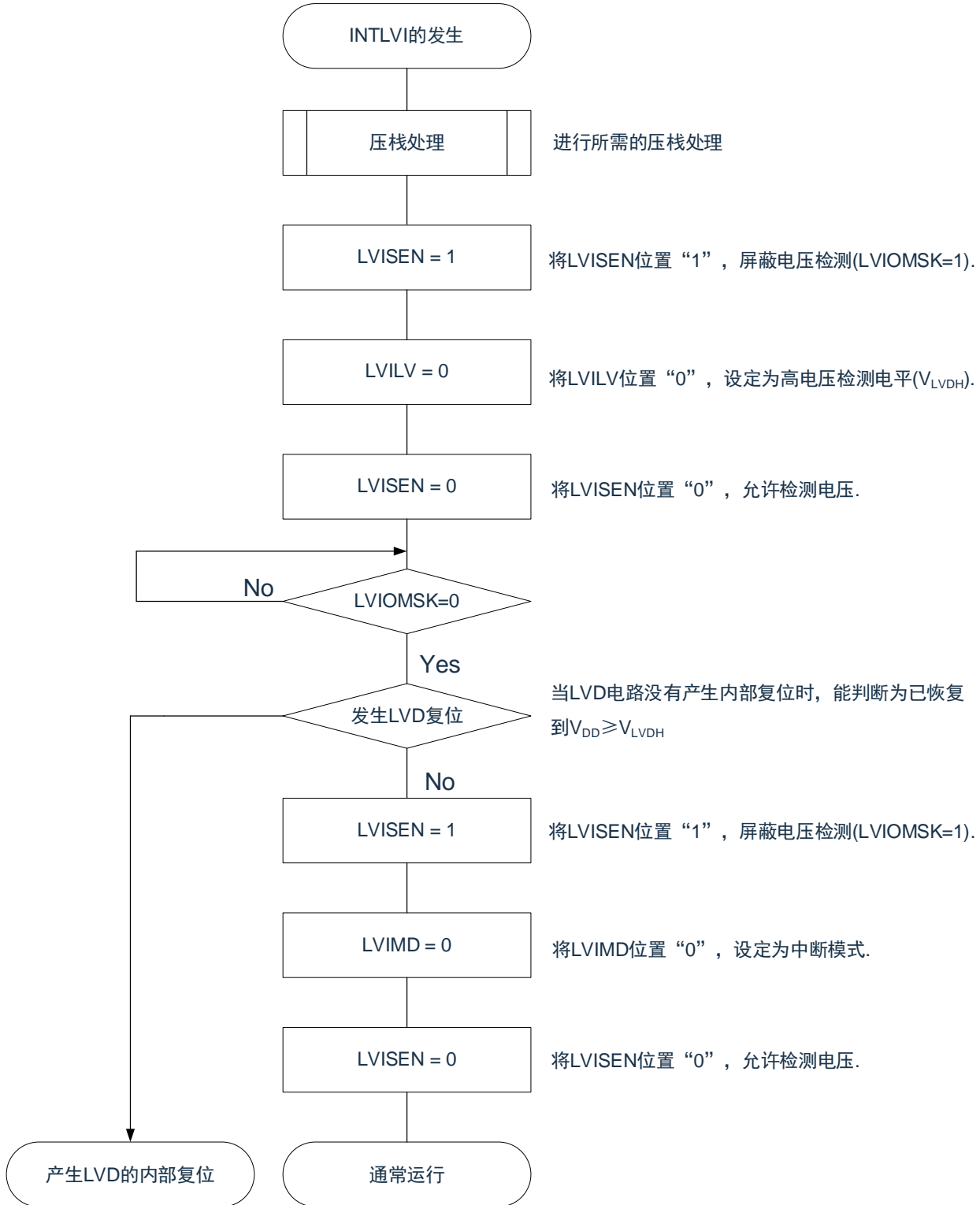
注1: 在产生复位信号后, LVIMK标志变为“1”。

注2: 当使用中断&复位模式时, 必须在发生中断后按照“图21-5: 工作电压的确认/复位的设定步骤”进行设定。

注3: 当使用中断&复位模式时, 必须在解除复位后按照“图21-6: 中断&复位模式的初始设定步骤”进行设定。

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21-5: 工作电压的确认/复位的设定步骤

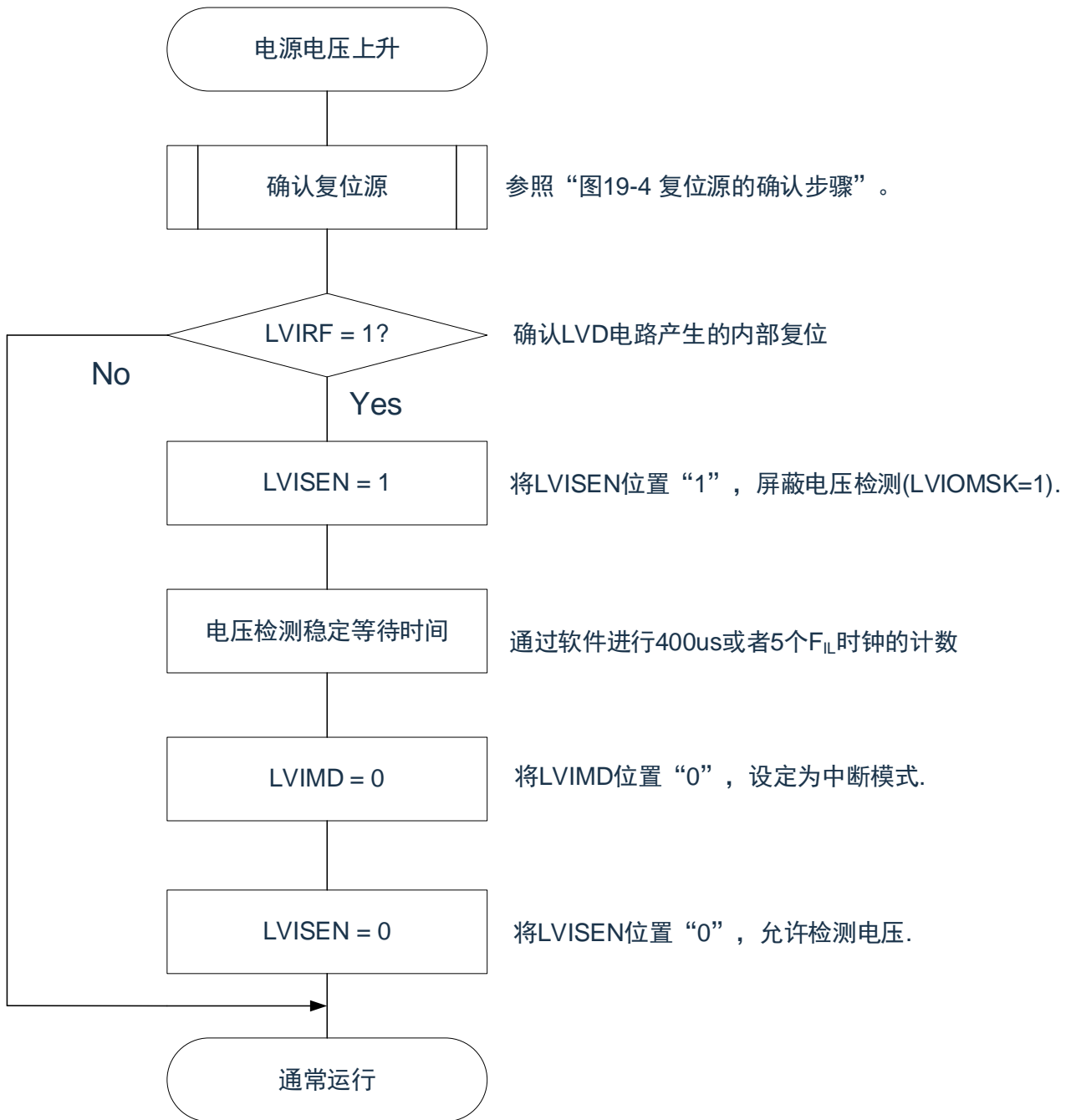


如果设定中断&复位模式（LVIMDS1、LVIMDS0=1、0），就在解除LVD复位（LVIRF=1）后需要400us或者5个F<sub>IL</sub>时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将LVIMD位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写LVIMD位时，必须将LVISEN位置“1”，屏蔽LVD产生的复位或者中断的产生。

中断&复位模式的初始设定步骤如图21-6所示。



图21-6: 中断&复位模式的初始设定步骤



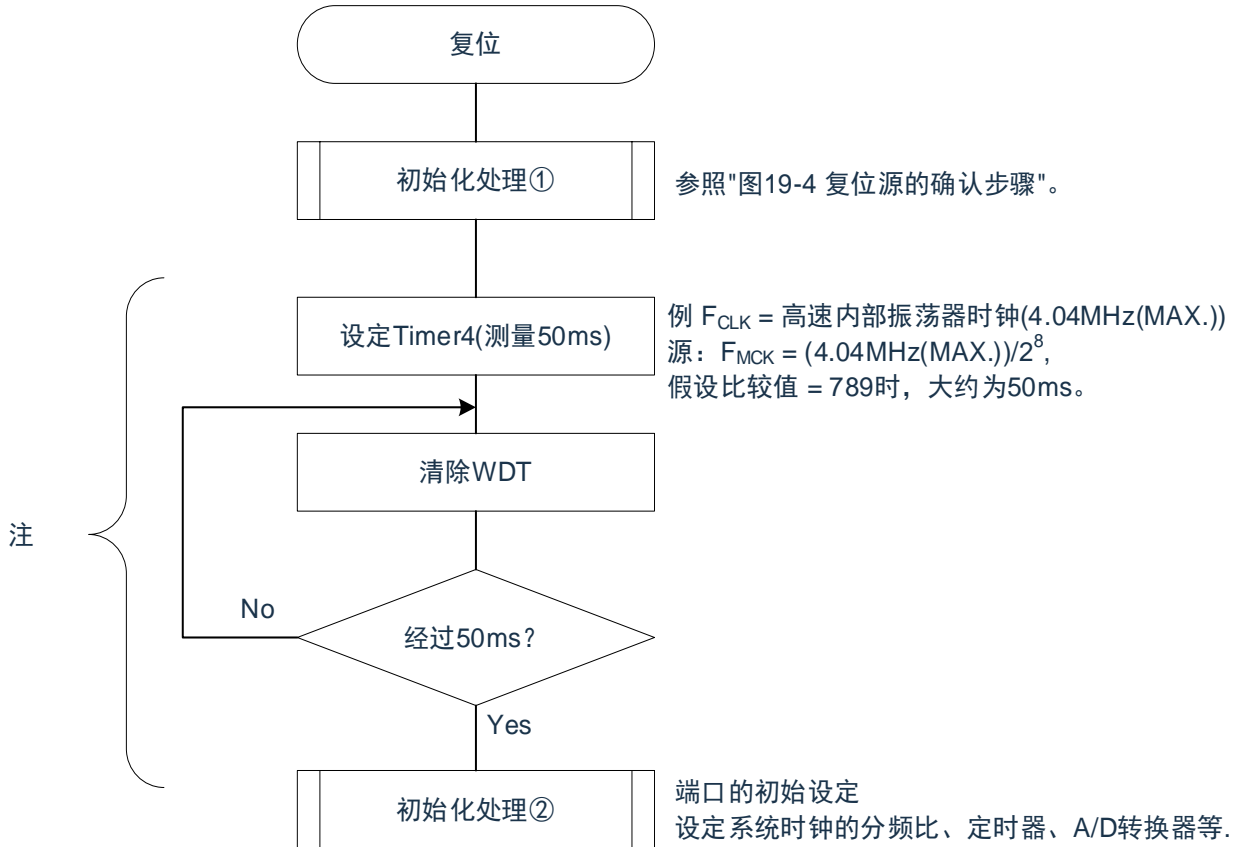
备注：F<sub>IL</sub>：低速内部振荡器时钟频率。

## 21.5 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, 任意设定解除复位到单片机开始运行的时间。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定。

图21-7: LVD检测电压附近的电源电压波动不超过50ms时的软件处理例子

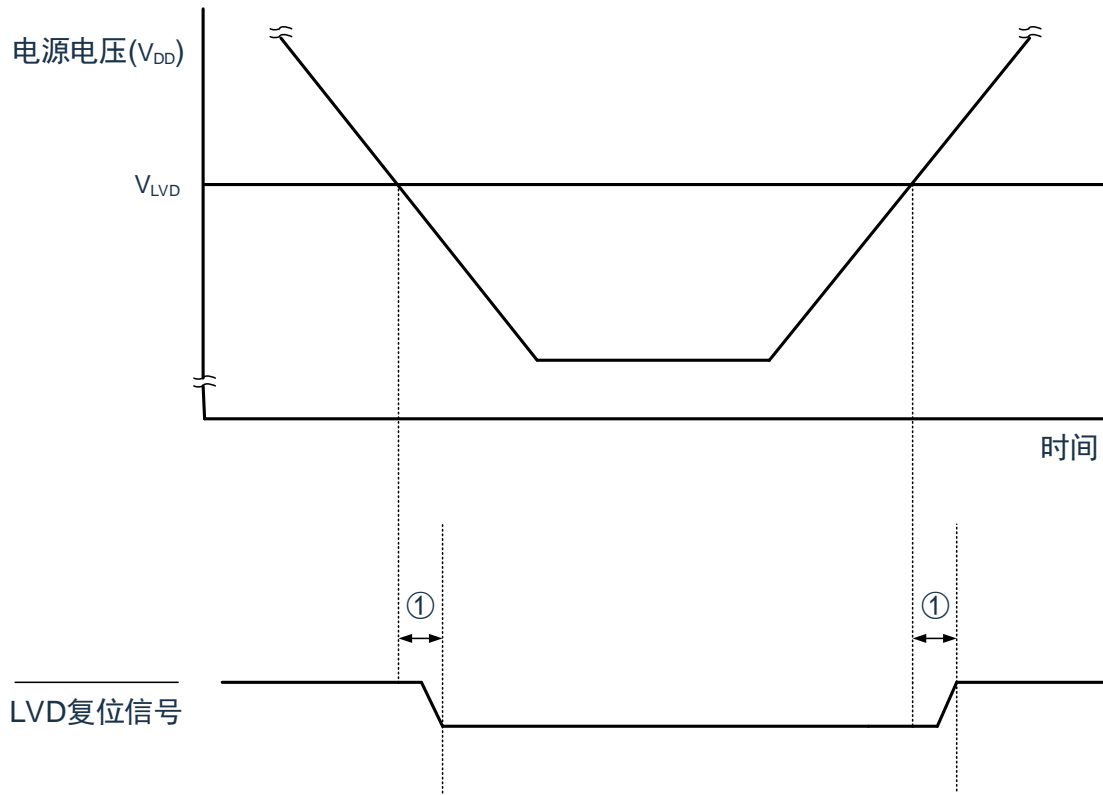


注: 如果在此期间再次发生复位, 就不转移到初始化处理②。

(2) 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压 ( $V_{DD}$ ) < LVD检测电压 ( $V_{LVD}$ ) 到产生LVD复位为止会发生延迟。同样，从LVD检测电压 ( $V_{LVD}$ ) ≤ 电源电压 ( $V_{DD}$ ) 到解除LVD复位为止也会发生延迟（参照图21-8）。

图21-8：从产生LVD复位源到产生或者解除LVD复位的延迟



备注：①：检测延迟（300us(MAX.)）

(1) 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。

(2) 有关将 LVD 置为 OFF 并且设定为 LVD 中断模式时工作电压下降的情况

在将LVD置为OFF并且设定为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22章 安全功能

### 22.1 安全功能的概要

为了对应 IEC60730 和 EC61508 安全标准，CMS32L032 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。

① “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。

② “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) SFR 保护功能

防止因 CPU 失控而改写 SFR。

(3) 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。

(4) A/D 测试功能

能通过 A/D 转换器的正 (+) 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。

(5) 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。

### 22.2 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>• 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>• 模拟输入通道指定寄存器 (ADS)</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“22.3 安全功能的运行”中进行说明。

## 22.3 安全功能的运行

### 22.3.1 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 能在初始设定（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。

#### 22.3.1.1 闪存CRC控制寄存器（CRC0CTL）

这是设定高速 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定 CRC0CTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表22-1：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	CRCCHK60	0	0	0	FEA2	FEA1	FEA0

CRC0EN	高速CRC运算器的运行控制
0	停止运行。
1	通过执行WFE指令开始运算。

CRCCHK60	FEA2	FEA1	FEA0	高速 CRC 的演算范围
0	0	0	0	00000H ~ 1FFBH(8K-4byte)
0	0	0	1	00000H ~ 3FFBH(16K-4byte)
0	0	1	0	00000H ~ 5FFBH(24K-4byte)
0	0	1	1	00000H ~ 7FFBH(32K-4byte)
0	1	0	0	00000H ~ 9FFBH(40K-4byte)
0	1	0	1	00000H ~ BFFBH(48K-4byte)
0	1	1	0	00000H ~ DFFBH(56K-4byte)
0	1	1	1	00000H ~ FFFBH(64K-4byte)
1	0	0	0	00000H ~ EFFFH(60K-4byte)

注意：

1. bit3~5必须设置为0.
2. 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。

### 22.3.1.2 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定 PGCRCL 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

表22-2：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

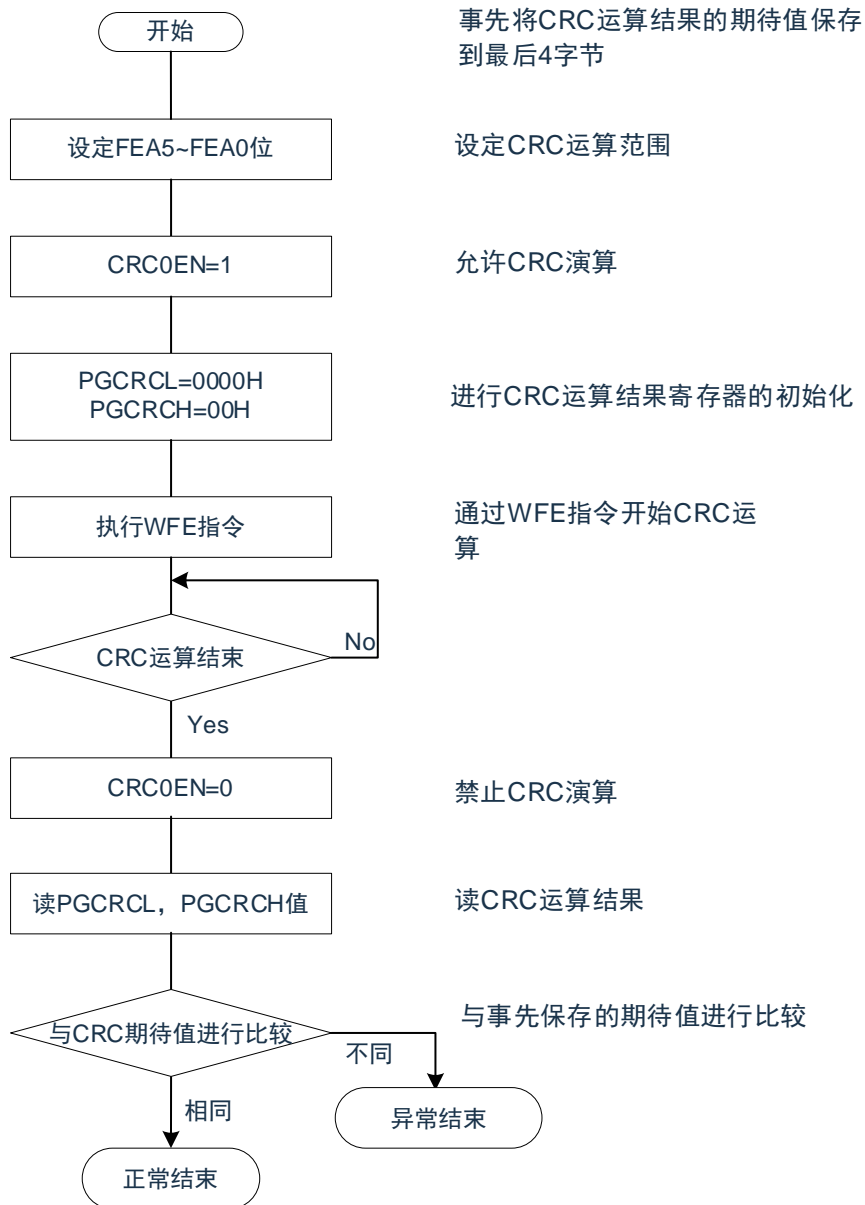
PGCRC15~0	高速CRC的运算结果
0000H~FFFFH	保存高速CRC的运算结果。

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 22-1 所示。

<操作流程>

图22-1：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。

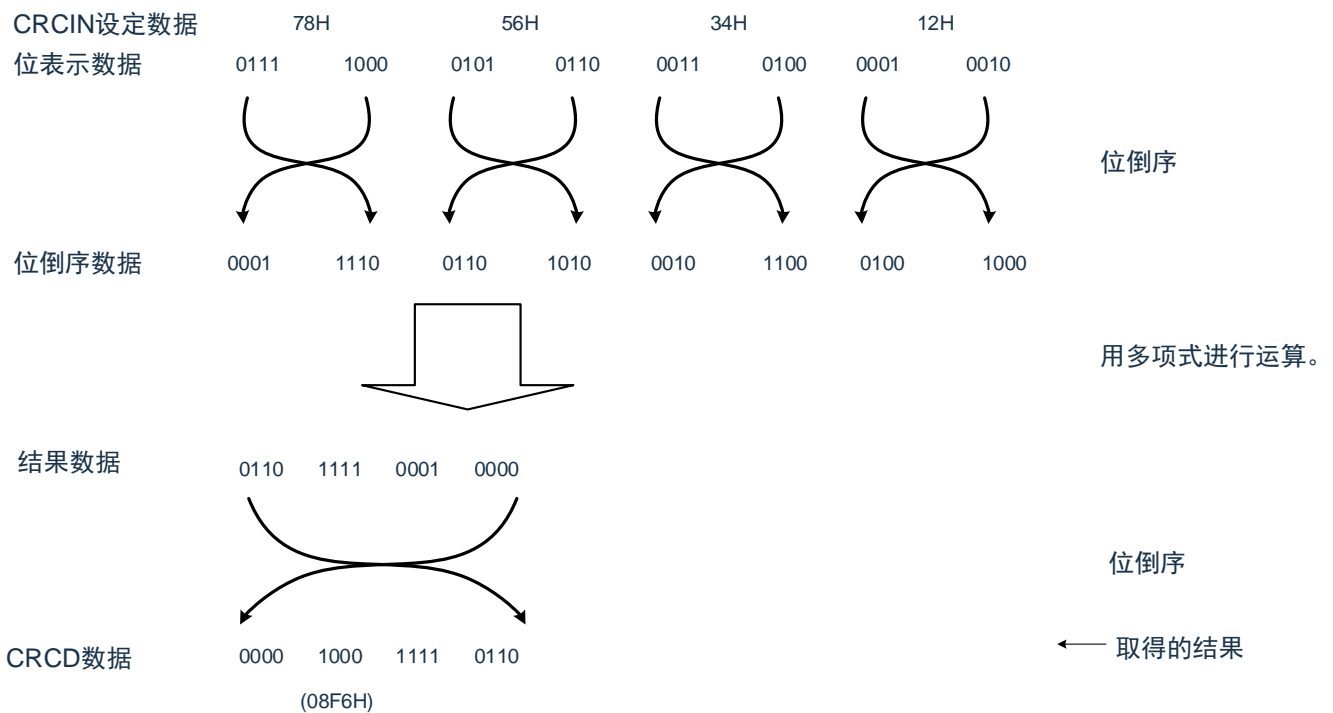
### 22.3.2 CRC运算功能（通用CRC）

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。

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。



备注：在执行程序的过程中，因为调式程序将软件断点的设定行改写为断点指令，所以如果在CRC运算的对象区设定软件断点，CRC的运算结果就不同。



### 22.3.2.1 CRC输入寄存器（CRCIN）

这是设定通用 CRC 的 CRC 计算数据的 8 位寄存器。能设定的范围为“00H~FFH”。

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

表22-3: CRC输入寄存器（CRCIN）的格式

地址: 400433ACH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
CRCIN								

bit7~0	功能
00H~FFH	数据输入

### 22.3.2.2 CRC数据寄存器 (CRCD)

这是保存通用 CRC 运算结果的寄存器。能设定的范围为“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定 CRCD 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

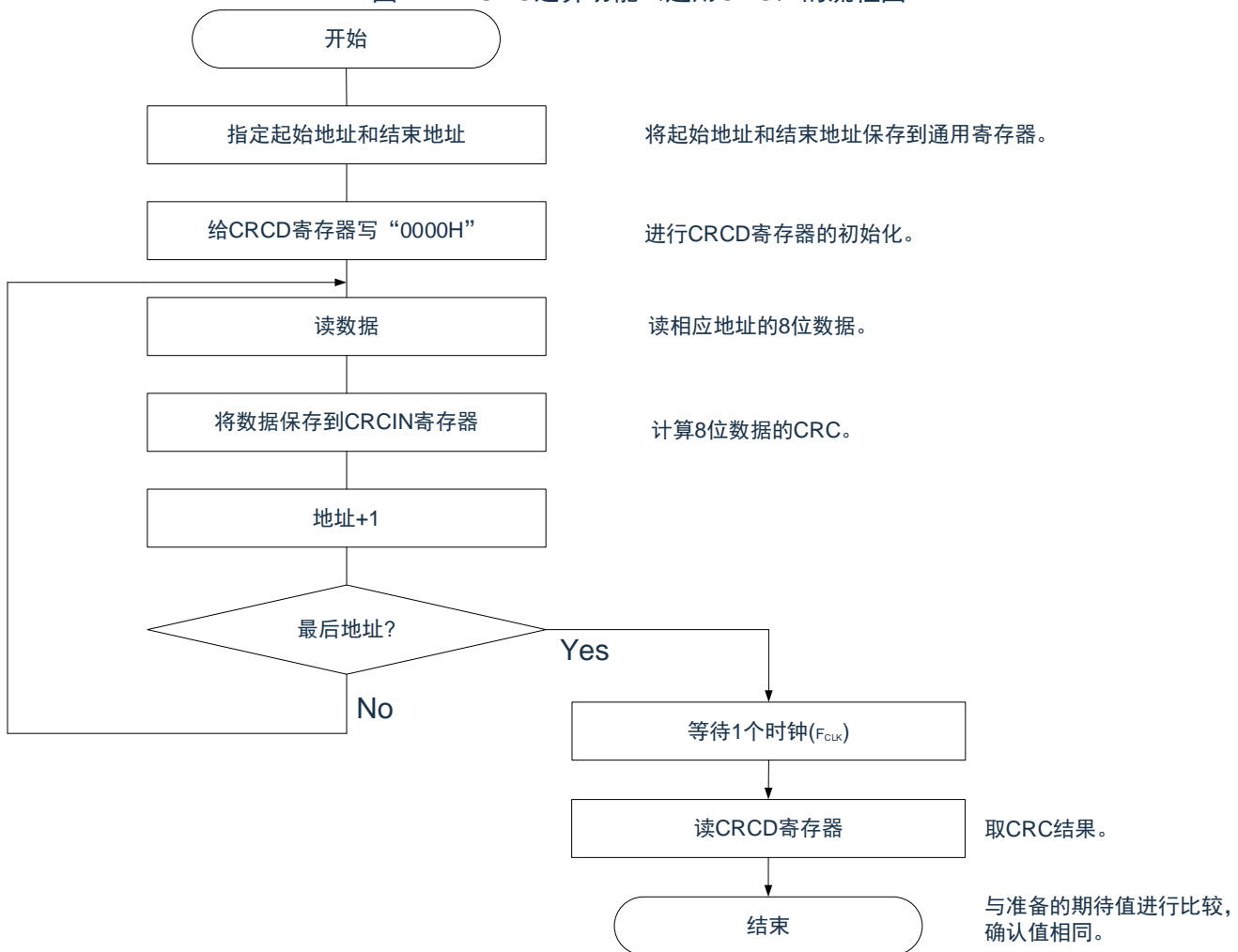
表22-4: CRC数据寄存器 (CRCD) 的格式

地址: 400432FAH	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

CRC 运算功能的流程图如图 22-2 所示

<操作流程>

图22-2: CRC运算功能 (通用CRC) 的流程图



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。

### 22.3.3 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 保护功能用于保护端口功能、中断功能、时钟控制功能、电压检测电路的控制寄存器的数据。

如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

#### 22.3.3.1 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 GPORT 位、GINT 位和 GCSC 位。

通过 8 位存储器操作指令设定 SFRGD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表22-5：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	0	GPORT	GINT	GCSC

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、PDxx、POMxx、PMCxx、PxxCFG <sup>注</sup>

GINT	中断功能的寄存器的保护
0	无效。能读写中断功能的控制寄存器。
1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的SFR]IFxx、MKxx、PRxx、EGPx、EGNx

GCSC	时钟控制功能、电压检测电路的控制寄存器的保护
0	无效。能读写时钟控制功能、电压检测电路的控制寄存器。
1	有效。时钟控制功能、电压检测电路的控制寄存器的写操作无效，能读。 [被保护的SFR]CMC、CSC、OSTS、CKC、PERx、OSMC、LVIM、LVIS

备注：不保护Pxx（端口寄存器）。

### 22.3.4 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

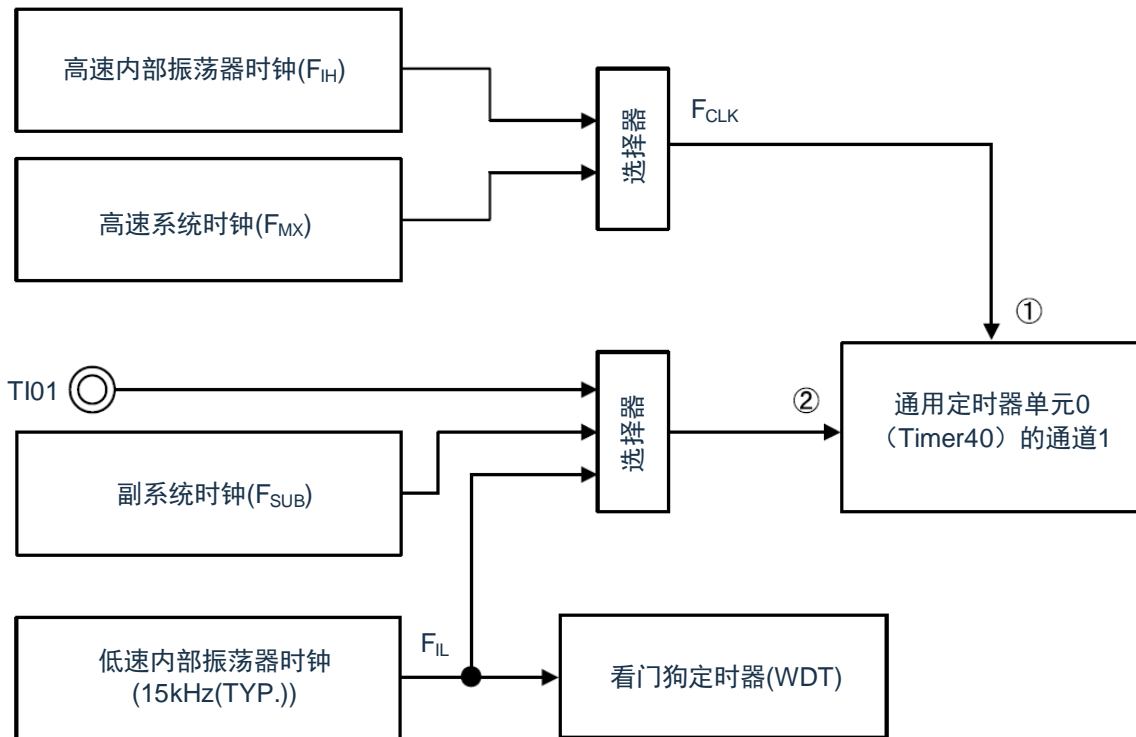
频率检测功能可使用 CPU/外围硬件的时钟频率 ( $F_{CLK}$ )，并且能通过测量 Timer40 通道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟停止振荡，就不能判断 2 个时钟的比率关系。

<要比较的时钟>

- (1) CPU/外围硬件的时钟频率 ( $F_{CLK}$ ) :
  - ① 高速内部振荡器时钟 ( $F_{IH}$ )
  - ② 高速系统时钟 ( $F_{MX}$ )
- (2) Timer40 通道 1 输入:
  - ① 通道 1 的定时器输入 (TI01)
  - ② 低速内部振荡器时钟 ( $F_{IL}$ : 15KHz(典型值))
  - ③ 副系统时钟 ( $F_{SUB}$ )<sup>注</sup>

图22-3: 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“5.8.4 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。

#### 22.3.4.1 定时器输入选择寄存器0 (TIS0)

寄存器说明请参考 5.3.8 节。

## 22.3.5 A/D测试功能

IEC60730 标准要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正 (+) 基准电压、模拟输入通道 (ANI)、温度传感器的输出电压和内部基准电压进行 A/D 转换，确认 A/D 转换器是否正常运行。

能通过以下步骤确认模拟多路转换器：

- ① 通过 ADS 寄存器选择 ANIx 引脚作为 A/D 转换对象。
- ② 对 ANIx 引脚进行 A/D 转换（转换结果 1-1）。
- ③ 通过 ADS 寄存器选择 A/D 转换器的正 (+) 基准电压作为 A/D 转换对象。
- ④ 对 A/D 转换器的正 (+) 基准电压进行 A/D 转换（转换结果 2-1）。
- ⑤ 通过 ADS 寄存器选择 ANIx 引脚作为 A/D 转换对象
- ⑥ 对 ANIx 引脚进行 A/D 转换（转换结果 1-2）。
- ⑦ 确认“转换结果 1-1”、“转换结果 1-2”相同。
- ⑧ 确认“转换结果 2-1”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。

注意：

1. 在①-⑧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。

### 22.3.5.1 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。

可通过 A/D 测试功能测量 ANIxx、温度传感器输出、内部基准电压 (1.45V) 或者正 (+) 基准电压。

寄存器说明请参考 11.2.6。

## 22.3.6 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。

### 22.3.6.1 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定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表22-6：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。

备注：

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”。
2. m=0~3 n=0~7

## 22.3.7 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

- (1) 用来作为序列号(例如 USB 字符序列号或者其他的终端应用)
- (2) 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。
- (3) 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\_0894

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

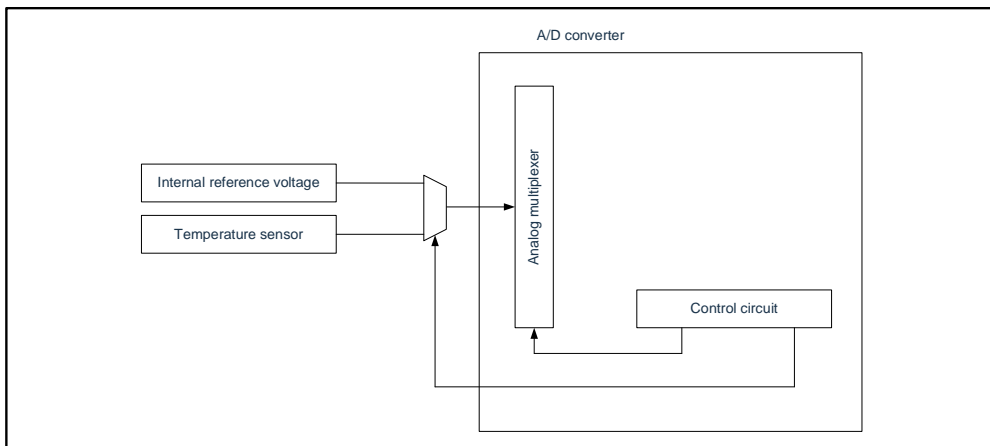
U ID[127:96]

# 第23章 温度传感器

## 23.1 温度传感器的功能

片上的温度传感器可以对产品的核心温度进行测量和监控，从而保证产品的可靠运行。温度传感器输出的电压与核心温度成正比，并且电压和温度之间是线性关系。其输出电压提供给ADC进行转换。图23-1显示了温度传感器框图。

图23-1：温度传感器框图



## 23.2 温度传感器的寄存器

### 23.2.1 温度传感器校准数据寄存器TSN25

地址：0x0050\_066C

符号	15	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。



## 23.3 温度传感器的使用说明

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:

$$T = (Vs - V1) / \text{slope} + 25^{\circ}\text{C}$$

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)

V1: 温度传感器测量25°C时的电压输出 (V)

Slope: 温度传感器的温度斜率(V/°C), slope = -3.5 mV/°C

备注: 温度传感器的精度较低, 不建议使用在精度要求高的场合。

## 第24章 选项字节

### 24.1 选项字节的功能

CMS32L032的闪存000C0H~000C3H, 500004H为选项字节区。

选项字节由用户选项字节(000C0H~000C2H)和闪存数据保护选项字节(000C3H, 500004H)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。

#### 24.1.1 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( $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ )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2MHz~32MHz、48MHz、64MHz中选择。

## 24.1.2 闪存数据保护选项字节（000C3H, 500004H）

- 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。

## 24.2 用户选项字节的格式

表24-1：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
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。
1	在睡眠模式中，允许计数器的运行。

备注：

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%。
2. F<sub>IL</sub>：低速内部振荡器的时钟频率

表24-2: 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			选项字节的设定值						
V <sub>LVDH</sub>		V <sub>LVDL</sub>	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
1.98V	1.94V	1.84V	0	0	1	1	0	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21章电压检测电路”。
2. 检测电压是典型值。详细内容请参照数据手册的LVD电路特性。

表24-3: 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								
V <sub>LVD</sub>		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.88V	1.84V	0	0	1	1	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21章电压检测电路”。
2. 检测电压是典型值。详细内容请参照数据手册的LVD电路特性。

表24-4：用户选项字节（000C1H）的格式(3/4)

 地址：000C1H<sup>注</sup>

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								
V <sub>LVD</sub>		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.88V	1.84V	0	0	1	1	1	0	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：必须给bit4写“1”。

备注：

1. 有关LVD电路的详细内容，请参照“第21章电压检测电路”。
2. 检测电压是典型值。详细内容请参照数据手册的LVD电路特性。

表24-5: 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 <sub>LVDH</sub>		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) 的设定。

## 备注:

1. ×: 忽略
2. 有关LVD电路的详细内容, 请参照“第21章电压检测电路”。
3. 检测电压是典型值。详细内容请参照数据手册的LVD电路特性。



表24-6: 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 <sub>HOCO</sub>	F <sub>IH</sub>
0	1	0	0	0	64MHz	64MHz
0	0	0	0	0	48MHz	48MHz
0	1	0	0	1	64MHz	32MHz
0	0	0	0	1	48MHz	24MHz
0	1	0	1	0	64MHz	16MHz
0	0	0	1	0	48MHz	12MHz
0	1	0	1	1	64MHz	8MHz
0	0	0	1	1	48MHz	6MHz
0	1	1	0	0	64MHz	4MHz
0	0	1	0	0	48MHz	3MHz
0	1	1	0	1	64MHz	2MHz
上述以外					禁止设定。	

注意: 必须给bit7~5写“1”。

备注: 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。

## 24.3 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。

表24-7：闪存数据保护选项字节（000C3H）的格式

地址：000C3H



地址：500004H



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

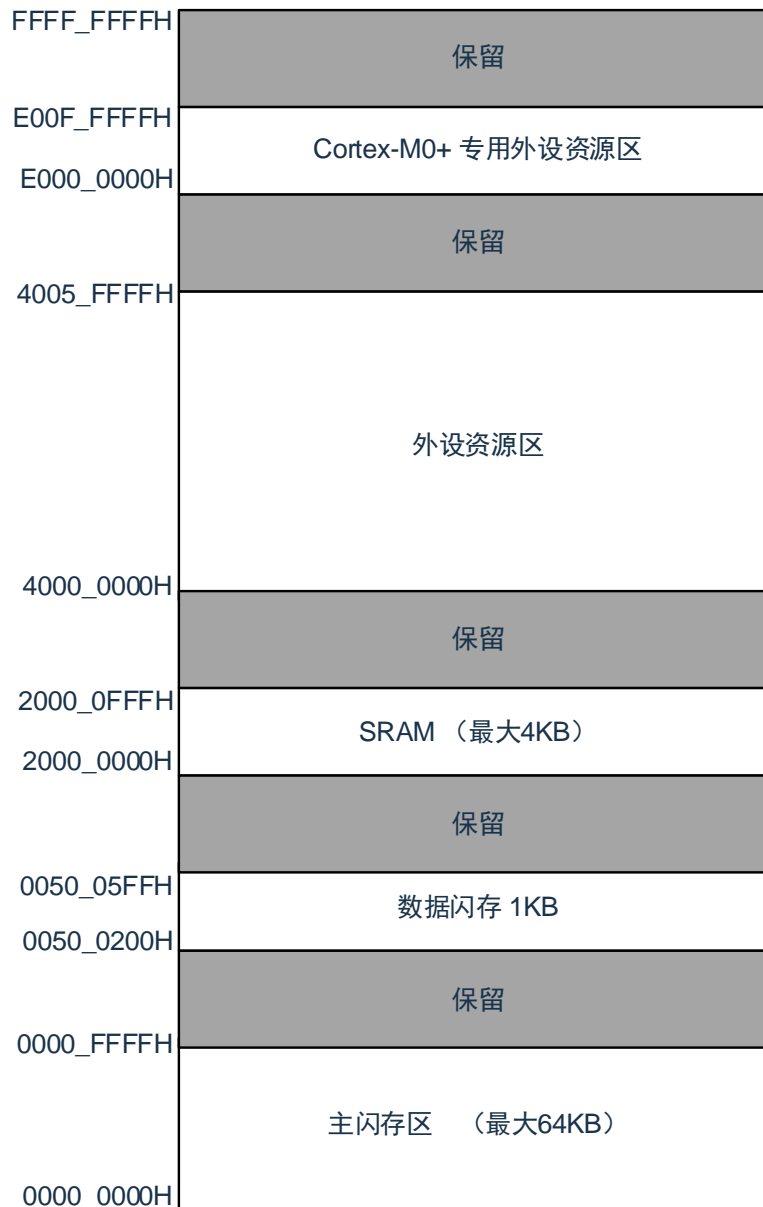
注意：50\_0004H地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

# 第25章 FLASH控制

## 25.1 FLASH控制功能描述

本制品包含一颗64KByte容量的FLASH存储器，共划分为128个Sector，每个Sector容量为512Byte。可做为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

## 25.2 FLASH存储器结构



## 25.3 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)

### 25.3.1 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 R/W

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

WRP	操作寄存器 (FLOPMD1/FLOPMD2) 写保护
0	不允许改写FLOPMD1/ FLOPMD2
1	允许改写FLOPMD1/ FLOPMD2

PRKEY[7:1]	WRP写保护
78h	允许改写WRP
上記以外	不允许改写WRP

## 25.3.2 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

## 25.3.3 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，擦除后不进行硬件校验
1	0	sector擦除，擦除后进行硬件校验
0	1	chip擦除 <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。且chip擦除不支持硬件校验。

### 25.3.4 Flash状态寄存器(FLSTS)

通过状态寄存器可以查询FLASH控制器的状态。

地址: 0x40020000	复位后: 00H	R/W																
符号	7	6	5	4	3	2	1	0										
FLSTS	0	0	0	0	0	EVF <sup>注</sup>	0	OVF <sup>注</sup>										

OVF	FLASH擦写操作完了标志
0	FLASH擦写操作未完成
1	FLASH擦写操作完成

注: OVF需要软件写“1”进行清除。若不清除, 不能进行下一次擦写操作。

EVF	FLASH擦除硬件校验错误标志
0	FLASH擦除后, 硬件校验没有发生错误
1	FLASH擦除后, 硬件校验发生了错误

注: EVF需要软件写“1”进行清除。

### 25.3.5 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																			
FLCERCNT	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			
	-	-	-	-	-	-	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≤16M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*T <sub>FCLK</sub> ), 需满足>30ms的硬件要求	

## 25.3.6 Flash页擦除时间控制寄存器（FLSERCNT）

通过FLSERCNT寄存器可以设置FLASH全片擦除的时间。

地址：0x40020014      复位后：不定      R/W

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

Load	擦除时间设定的选择 <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（FLSERCNT[9:0]）

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 16\text{M}$ 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* $T_{\text{FCLK}}$ )，需满足 $> 2\text{ms}$ 的硬件要求	

## 25.3.7 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[12:0]												
	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（ $T_{PROG}$ ）设定 <sup>注</sup>
0	使用硬件设定的写入时间
1	使用软件设定的写入时间（FLPROCNT[8:0]）

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 16M$ 时，可以使用硬件设定时间，不设定FLPROCNT。

FLPROCNT[8:0]	软件写入时间设定
写入时间=（PROCNT* $T_{FCLK}$ ），需满足 $>7\mu s$ 的硬件要求	

Load1	写入动作建立时间（ $T_{PGS}$ ）设定 <sup>注</sup>
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（FLPGSCNT12:0]）

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 16M$ 时，可以使用硬件设定时间，不设定FLPGSCNT。

FLPGSCNT[12:0]	软件写入动作建立时间设定
写入动作建立时间=（PGSCNT* $T_{FCLK}$ ），写字节需满足 $>70\mu s$ 的硬件要求，写半字需满足 $>40\mu s$ 的硬件要求	

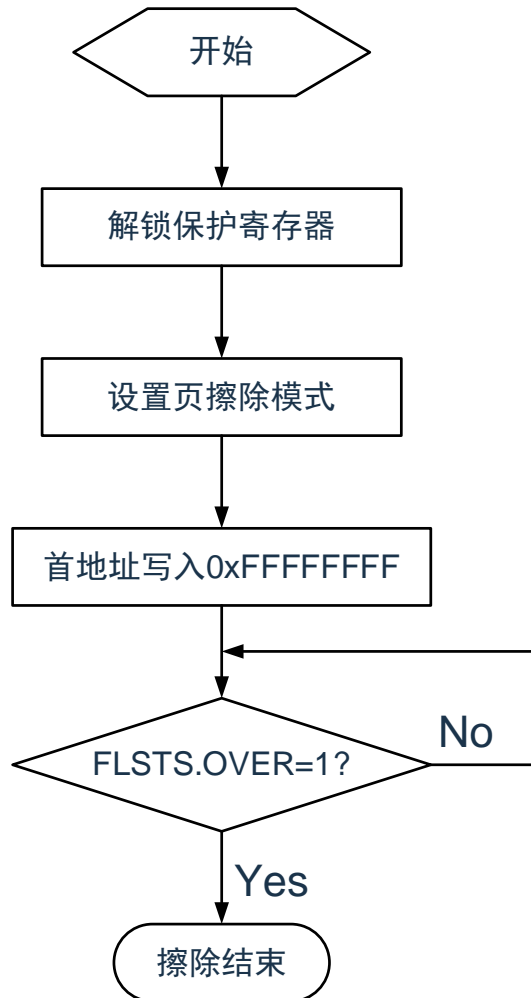


## 25.4 FLASH操作方法

### 25.4.1 页擦除 (sector erase)

sector擦除，擦除时间由硬件实现，也可以通过FLSERCNT配置。操作流程如下：

- 1) 设置FLERMD.ERMD0为1'b0，选择sector擦除模式,根据是否需要硬件校验选择设置ERMD1的值；
- 2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，
- 3) 向擦除目标sector的首地址写入任意数据。例： $*((\text{unsigned long } *)0x00000200) = 0xffffffff$ 。
- 4) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示擦除操作完成。
- 5) 如果设置了擦除后进行硬件校验（ERMD1=1），可以通过软件判断FLSTS.EVF，查询是否校验正确。
- 6) 进行下次操作前，软件置"1"来清除FLSTS。



## 25.4.2 全片擦除 (chiperase)

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。

## 25.4.3 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。

## 25.5 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。

## 25.6 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=16M时，推荐使用硬件设置的擦写参数，无需设置参数寄存器。
- 如果擦写操作从FLASH内执行，则CPU会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从RAM里执行，CPU不会停止取指，当前可以继续下一条指令。
- 在FLASH处于编程操作中时，如果CPU执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

## 附录 修订记录

版本	日期	修订内容
V1.0.0	2023年6月	初始版本
V1.0.1	2023年7月	更正21.5章节文字错误
V1.0.2	2023年8月	<ol style="list-style-type: none"> <li>1. 删除10.1,10.3.1,10.4.1中对WDTE寄存器执行位操作指令功能的描述</li> <li>2. 删除第13章通用串行通信单元有关简易I<sup>2</sup>C部分</li> <li>3. 修改图13-1 通用串行通信单元0的框图</li> <li>4. 删除表2 2 管脚功能数字映射中简易I<sup>2</sup>C的部分</li> <li>5. 更正图5 44: 多重 PWM 输出功能时 (从属通道) 的寄存器设定内容例子的内容</li> <li>6. 更正25.3.5章节和25.3.6章节的软件擦除时间内容</li> <li>7. 更正25.3.7章节的软件写入时间设定和软件写入动作建立时间设定内容</li> <li>8. 更正图表标识顺序; 更新目录</li> <li>9. 调整中断功能和待机功能章节位置</li> </ol>
V1.0.3	2023年12月	<ol style="list-style-type: none"> <li>1. 更正第6章及章节6.3.3内容</li> <li>2. 修改表12-7内容</li> <li>3. 修改22.3.7内容</li> <li>4. 增加WDTCFG配置寄存器说明</li> <li>5. 更正表2-2 管脚功能数字映射</li> </ol>