



BAT32G133用户手册

基于ARM® Cortex®-M0+的超低功耗32位微控制器

V1.0.6

请注意以下有关CMS知识产权政策

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

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

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

文档使用说明

本手册是BAT32G133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，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。

BAT32G133系列产品的数据手册信息如下：

BAT32G133xx: BAT32G133_datasheet_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+核心和NVIC的详细说明，请参照对应ARM的文档。

目录

文档使用说明	2
第1章 CPU	14
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
第2章 引脚功能	18
2.1 端口通用功能	18
2.2 端口复用功能	19
2.3 控制端口功能的寄存器	20
2.3.1 端口模式寄存器 (PMxx)	21
2.3.2 端口寄存器 (Pxx)	22
2.3.3 上拉电阻选择寄存器 (PUxx)	23
2.3.4 下拉电阻选择寄存器 (PDxx)	24
2.3.5 端口输出模式寄存器 (POMxx)	25
2.3.6 端口模式控制寄存器 (PMCxx)	26
2.3.7 端口复用功能配置寄存器 (PxxCFG)	27
2.4 未使用引脚的处理	29
2.5 特殊引脚的设置	29
2.5.1 P00端口 (默认为RESETB)	29
第3章 系统结构	30
3.1 概述	30
3.2 系统地址划分	31
第4章 时钟发生电路	33
4.1 时钟发生电路的功能	33
4.2 时钟发生电路的结构	35
4.3 控制时钟发生电路的寄存器	38
4.3.1 时钟运行模式控制寄存器 (CMC)	38
4.3.2 系统时钟控制寄存器 (CKC)	40
4.3.3 时钟运行状态控制寄存器 (CSC)	41
4.3.4 振荡稳定时间计数器的状态寄存器 (OSTC)	42
4.3.5 振荡稳定时间选择寄存器 (OSTS)	44
4.3.6 外围允许寄存器0、1 (PER0、PER1)	45
4.3.7 副系统时钟提供模式控制寄存器 (OSMC)	49
4.3.8 高速内部振荡器的频率选择寄存器 (HOCODIV)	50
4.3.9 低速内部振荡器的频率选择寄存器 (LOSCSEL) 及其保护寄存器 (PRCR)	51
4.3.10 外部复位引脚的复位功能屏蔽寄存器 (RSTM)	52

4.4	系统时钟振荡电路	53
4.4.1	X1振荡电路	53
4.4.2	XT1振荡电路	54
4.4.3	高速内部振荡器	57
4.4.4	低速内部振荡器	57
4.5	时钟发生电路的运行	58
4.6	时钟控制	60
4.6.1	高速内部振荡器的设置例子	60
4.6.2	X1振荡电路的设置例子	62
4.6.3	XT1振荡电路的设置例子	63
4.6.4	CPU时钟的状态转移图	64
4.6.5	CPU时钟转移前的条件和转移后的处理	70
4.6.6	CPU时钟和主系统时钟的切换所需时间	72
4.6.7	时钟振荡停止前的条件	73
第5章	通用定时器单元Timer4	74
5.1	通用定时器单元的功能	75
5.1.1	独立通道运行功能	75
5.1.2	多通道联动运行功能	77
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）	78
5.2	通用定时器单元的结构	79
5.2.1	通用定时器单元0寄存器列表	82
5.2.2	通用定时器单元1寄存器列表	83
5.2.3	定时器计数寄存器mn (TCRmn)	84
5.2.4	定时器数据寄存器mn (TDRmn)	85
5.3	控制通用定时器单元的寄存器	86
5.3.1	外围允许寄存器0 (PER0)	87
5.3.2	定时器时钟选择寄存器m (TPSm)	88
5.3.3	定时器模式寄存器mn (TMRmn)	91
5.3.4	定时器状态寄存器mn (TSRmn)	96
5.3.5	定时器通道允许状态寄存器m (TEm)	97
5.3.6	定时器通道开始寄存器m (TSM)	98
5.3.7	定时器通道停止寄存器m (TTm)	99
5.3.8	定时器输入输出选择寄存器 (TIOS0)	100
5.3.9	定时器输出允许寄存器m (TOEm)	101
5.3.10	定时器输出寄存器m (TOM)	102
5.3.11	定时器输出电平寄存器m (TOLm)	103
5.3.12	定时器输出模式寄存器m (TOMm)	104
5.3.13	噪声滤波器允许寄存器1 (NFEN1)	105
5.3.14	噪声滤波器允许寄存器2 (NFEN2)	106
5.3.15	控制定时器输入/输出引脚端口功能的寄存器	107
5.4	通用定时器单元的基本规则	108
5.4.1	多通道联动运行功能的基本规则	108
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）	110

5.5	计数器的运行	111
5.5.1	计数时钟 (f_{TCLK})	111
5.5.2	计数器的开始时序	113
5.5.3	计数器的运行	114
5.6	通道输出 (TOmn引脚) 的控制	119
5.6.1	TOmn引脚输出电路的结构	119
5.6.2	TOmn引脚的输出设置	120
5.6.3	通道输出运行的注意事项	121
5.6.4	TOmn位的一次性操作	125
5.6.5	有关开始计数时的定时器中断和TOmn引脚输出	126
5.7	定时器输入 (TImn) 的控制	127
5.7.1	TImn引脚输入电路的结构	127
5.7.2	噪声滤波器	127
5.7.3	操作通道输入时的注意事项	128
5.8	通用定时器单元的独立通道运行功能	129
5.8.1	作为间隔定时器/方波输出的运行	129
5.8.2	作为外部事件计数器的运行	133
5.8.3	作为分频器的运行	136
5.8.4	作为输入脉冲间隔测量的运行	139
5.8.5	作为输入信号高低电平宽度测量的运行	142
5.8.6	作为延迟计数器的运行	146
5.9	通用定时器单元的多通道联动运行功能	149
5.9.1	作为单触发脉冲输出功能的运行	149
5.9.2	作为PWM功能的运行	156
5.9.3	作为多重 PWM 输出功能的运行	163
第6章	实时时钟	171
6.1	实时时钟的功能	171
6.2	实时时钟的结构	171
6.3	控制实时时钟的寄存器	173
6.3.1	外围允许寄存器0 (PER0)	174
6.3.2	实时时钟选择寄存器 (RTCCCL)	175
6.3.3	实时时钟控制寄存器0 (RTCC0)	176
6.3.4	实时时钟控制寄存器1 (RTCC1)	177
6.3.5	时钟误差校正寄存器 (SUBCUD)	179
6.3.6	秒计数寄存器 (SEC)	180
6.3.7	分钟计数寄存器 (MIN)	180
6.3.8	小时计数寄存器 (HOUR)	181
6.3.9	日计数寄存器 (DAY)	183
6.3.10	星期计数寄存器 (WEEK)	184
6.3.11	月计数寄存器 (MONTH)	185
6.3.12	年计数寄存器 (YEAR)	185
6.3.13	闹钟分钟寄存器 (ALARMWM)	186

6.3.14	闹钟小时寄存器 (ALARMWH)	186
6.3.15	闹钟星期寄存器 (ALARMWW)	187
6.3.16	端口模式寄存器和端口寄存器	187
6.4	实时时钟的运行	188
6.4.1	实时时钟的运行开始	188
6.4.2	开始运行后睡眠模式的转移	189
6.4.3	实时时钟计数器的读写	190
6.4.4	实时时钟的闹钟设置	192
6.4.5	实时时钟的1Hz输出	193
6.4.6	实时时钟的时钟误差校正例子	194
第7章	15位间隔定时器	196
7.1	15位间隔定时器的功能	196
7.2	15位间隔定时器的结构	196
7.3	控制15位间隔定时器的寄存器	197
7.3.1	外围允许寄存器0 (PER0)	197
7.3.2	实时时钟选择寄存器 (RTCCL)	198
7.3.3	15位间隔定时器的控制寄存器 (ITMC)	199
7.4	15位间隔定时器的运行	200
7.4.1	15位间隔定时器的运行时序	200
7.4.2	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	201
第8章	时钟输出/蜂鸣器输出控制电路	202
8.1	时钟输出/蜂鸣器输出控制电路的功能	202
8.2	时钟输出/蜂鸣器输出控制电路的结构	204
8.3	控制时钟输出/蜂鸣器输出控制电路的寄存器	204
8.3.1	时钟输出选择寄存器n (CKSn)	204
8.3.2	控制时钟输出/蜂鸣器输出引脚端口功能的寄存器	206
8.4	时钟输出/蜂鸣器输出控制电路的运行	207
8.4.1	输出引脚的运行	207
8.5	时钟输出/蜂鸣器输出控制电路的注意事项	207
第9章	看门狗定时器	208
9.1	看门狗定时器的功能	208
9.2	看门狗定时器的结构	208
9.3	控制看门狗定时器的寄存器	210
9.3.1	看门狗定时器的允许寄存器 (WDTE)	210
9.4	看门狗定时器的运行	211
9.4.1	看门狗定时器的运行控制	211
9.4.2	看门狗定时器上溢时间的设置	212
9.4.3	看门狗定时器窗口打开期间的设置	213
9.4.4	看门狗定时器间隔中断的设置	214
第10章	A/D转换器	215
10.1	A/D转换器的功能	215

10.2	控制A/D转换器的寄存器	217
10.2.1	外围允许寄存器0 (PER0)	218
10.2.2	A/D转换器的模式寄存器0 (ADM0)	219
10.2.3	A/D转换器的模式寄存器1 (ADM1)	224
10.2.4	A/D转换器的模式寄存器2 (ADM2)	225
10.2.5	A/D转换器的触发模式寄存器 (ADTRG)	226
10.2.6	模拟输入通道指定寄存器 (ADS)	227
10.2.7	12位A/D转换结果寄存器 (ADCR)	229
10.2.8	8位A/D转换结果寄存器 (ADCRH)	230
10.2.9	转换结果比较上限值设置寄存器 (ADUL)	231
10.2.10	转换结果比较下限值设置寄存器 (ADLL)	231
10.2.11	A/D采样时间控制寄存器 (ADNSMP)	232
10.2.12	A/D采样时间延长寄存器 (ADSMPWAIT)	234
10.2.13	A/D测试寄存器 (ADTES)	235
10.2.14	A/D状态寄存器 (ADFLG)	236
10.2.15	A/D充放电控制寄存器 (ADNDIS)	237
10.2.16	控制模拟输入引脚端口功能的寄存器	237
10.3	输入电压和转换结果	238
10.4	A/D转换器的运行模式	239
10.4.1	软件触发模式 (选择模式、连续转换模式)	239
10.4.2	软件触发模式 (选择模式、单次转换模式)	240
10.4.3	软件触发模式 (扫描模式、连续转换模式)	241
10.4.4	软件触发模式 (扫描模式、单次转换模式)	242
10.4.5	硬件触发无等待模式 (选择模式、连续转换模式)	243
10.4.6	硬件触发无等待模式 (选择模式、单次转换模式)	244
10.4.7	硬件触发无等待模式 (扫描模式、连续转换模式)	245
10.4.8	硬件触发无等待模式 (扫描模式、单次转换模式)	246
10.4.9	硬件触发等待模式 (选择模式、连续转换模式)	247
10.4.10	硬件触发等待模式 (选择模式、单次转换模式)	248
10.4.11	硬件触发等待模式 (扫描模式、连续转换模式)	249
10.4.12	硬件触发等待模式 (扫描模式、单次转换模式)	250
10.5	A/D转换器的设置流程图	251
10.5.1	软件触发模式的设置	251
10.5.2	硬件触发无等待模式的设置	252
10.5.3	硬件触发等待模式的设置	253
10.5.4	选择温度传感器的输出电压/内部基准电压时的设置	254
10.5.5	测试模式的设置	255
第11章	比较器	256
11.1	比较器的功能	256
11.2	比较器的结构	257
11.3	控制比较器的寄存器	259
11.3.1	外围允许寄存器1 (PER1)	260
11.3.2	比较器模式设置寄存器 (COMPMDR)	261

11.3.3	比较器滤波控制寄存器 (COMPFIR)	262
11.3.4	比较器输出控制寄存器 (COMPOCR)	263
11.3.5	比较器内置基准电压控制寄存器 (CVRCTL)	265
11.3.6	比较器内置基准电压选择寄存器 (CiRVM)	266
11.3.7	比较器0的输入信号选择控制寄存器 (CMPSEL0)	267
11.3.8	比较器1的输入信号选择控制寄存器 (CMPSEL1)	268
11.3.9	控制模拟输入引脚端口功能的寄存器	268
11.4	运行说明	269
11.4.1	比较器i的数字滤波器 (i=0、1)	271
11.4.2	比较器i中断 (i=0、1)	271
11.4.3	向联动控制器 (EVENTC) 输出的事件信号	272
11.4.4	比较器i的输出 (i=0、1)	273
11.4.5	比较器时钟的停止和提供	273
第12章	可编程增益放大器 (PGA)	274
12.1	可编程增益放大器的功能	274
12.2	可编程增益放大器的结构	275
12.3	可编程增益放大器的寄存器	276
12.3.1	外围允许寄存器1(PER1)	276
12.3.2	可编程增益放大器控制寄存器 (PGAnCTL)	277
12.3.3	控制模拟输入引脚端口功能的寄存器	277
12.4	可编程增益放大器的运行	278
12.4.1	可编程增益放大器的开始运行步骤	278
12.4.2	可编程增益放大器的停止运行步骤	279
第13章	通用串行通信单元	280
13.1	通用串行通信单元的功能	281
13.1.1	3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)	281
13.1.2	UART (UART0~UART2)	282
13.2	通用串行通信单元的结构	283
13.2.1	移位寄存器	286
13.2.2	串行数据寄存器mn (SDRmn) 的低8位或者低9位	286
13.3	控制通用串行通信单元的寄存器	288
13.3.1	外围允许寄存器0 (PER0)	289
13.3.2	串行时钟选择寄存器m (SPSm)	290
13.3.3	串行模式寄存器mn (SMRmn)	291
13.3.4	串行通信运行设置寄存器mn (SCRmn)	293
13.3.5	串行数据寄存器mn (SDRmn)	295
13.3.6	串行标志清除触发寄存器mn (SIRmn)	296
13.3.7	串行状态寄存器mn (SSRmn)	297
13.3.8	串行通道开始寄存器m (SSm)	299
13.3.9	串行通道停止寄存器m (STm)	300
13.3.10	串行通道允许状态寄存器m (SEm)	301
13.3.11	串行输出允许寄存器m (SOEm)	302

13.3.12	串行输出寄存器m (SOm)	303
13.3.13	串行输出电平寄存器m (SOLm)	304
13.3.14	输入切换控制寄存器 (ISC)	305
13.3.15	噪声滤波器允许寄存器0 (NFEN0)	306
13.3.16	控制串行输入/输出引脚端口功能的寄存器	307
13.4	运行停止模式	308
13.4.1	以单元为单位停止运行的情况	308
13.4.2	按通道停止运行的情况	309
13.5	3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信的运行	310
13.5.1	主控发送	311
13.5.2	主控接收	320
13.5.3	主控的发送和接收	329
13.5.4	从属发送	338
13.5.5	从属接收	347
13.5.6	从属的发送和接收	354
13.5.7	传送时钟频率的计算	363
13.5.8	在3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信过程中发生错误时的处理步骤	365
13.6	从属选择输入功能的时钟同步串行通信的运行	366
13.6.1	从属发送	369
13.6.2	从属接收	379
13.6.3	从属的发送和接收	386
13.6.4	传送时钟频率的计算	396
13.6.5	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	397
13.7	UART (UART0~UART2) 通信的运行	398
13.7.1	UART发送	399
13.7.2	UART接收	408
13.7.3	波特率的计算	416
13.7.4	在UART (UART0~UART2) 通信过程中发生错误时的处理步骤	420
第14章	串行接口IICA	421
14.1	串行接口IICA的功能	421
14.2	串行接口IICA的结构	424
14.3	控制串行接口IICA的寄存器	427
14.3.1	外围允许寄存器0 (PER0)	428
14.3.2	IICA控制寄存器n0 (IICCTLn0)	428
14.3.3	IICA状态寄存器n (IICSn)	433
14.3.4	IICA标志寄存器n (IICFn)	436
14.3.5	IICA控制寄存器n1 (IICCTLn1)	438
14.3.6	IICA低电平宽度设置寄存器n (IICWLn)	440
14.3.7	IICA高电平宽度设置寄存器n (IICWHn)	440
14.3.8	控制IICA引脚端口功能的寄存器	441
14.4	I ² C总线模式的功能	442
14.4.1	引脚结构	442

14.4.2	通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法	443
14.5	I ² C总线的定义和控制方法	445
14.5.1	开始条件	446
14.5.2	地址	447
14.5.3	传送方向的指定	447
14.5.4	应答 (ACK)	448
14.5.5	停止条件	449
14.5.6	等待	450
14.5.7	等待的解除方法	452
14.5.8	中断请求 (INTIICAn) 的产生时序和等待控制	453
14.5.9	地址匹配的检测方法	455
14.5.10	错误的检测	455
14.5.11	扩展码	456
14.5.12	仲裁	457
14.5.13	唤醒功能	459
14.5.14	通信预约	462
14.5.15	其他注意事项	466
14.5.16	通信运行	467
14.5.17	I ² C中断请求 (INTIICAn) 的产生时序	475
14.6	时序图	496
第15章	IrDA	511
15.1	IrDA的功能	511
15.2	控制IrDA的寄存器	512
15.2.1	外围允许寄存器0 (PER0)	512
15.2.2	IrDA控制寄存器 (IRCR)	513
15.3	IrDA的运行	514
15.3.1	IrDA通信的操作步骤	514
15.3.2	发送	515
15.3.3	接收	515
15.3.4	高电平脉宽的选择	516
15.4	使用IrDA时的注意事项	517
第16章	增强型DMA	518
16.1	DMA的功能	518
16.2	DMA的结构	520
16.3	控制DMA的寄存器	521
16.3.1	DMA控制数据区和DMA向量表的分配	522
16.3.2	控制数据的分配	523
16.3.3	向量表	525
16.3.4	外围允许寄存器1 (PER1)	527
16.3.5	DMA控制寄存器j (DMACRj) (j=0~23)	527
16.3.6	DMA块大小寄存器j (DMBSLj) (j=0~23)	529
16.3.7	DMA传送次数寄存器j (DMACTj) (j=0~23)	530

16.3.8	DMA传送次数重加载寄存器j (DMRLDj) (j=0~23)	531
16.3.9	DMA源地址寄存器j (DMSARj) (j=0~23)	532
16.3.10	DMA目标地址寄存器j (DMDARj) (j=0~23)	532
16.3.11	DMA启动允许寄存器i (DMAENi) (i=0~2)	533
16.3.12	DMA基址寄存器 (DMABAR)	535
16.4	DMA的运行	536
16.4.1	启动源	536
16.4.2	正常模式	537
16.4.3	重复模式	540
16.4.4	链传送	543
16.5	使用DMA时的注意事项	545
16.5.1	DMA控制数据和向量表	545
16.5.2	DMA控制数据区和DMA向量表区的分配	545
16.5.3	DMA的执行时钟数	546
16.5.4	DMA的响应时间	547
16.5.5	DMA的启动源	547
16.5.6	待机模式中的运行	548
第17章	联动控制器(EVENTC)	549
17.1	EVENTC的功能	549
17.2	EVENTC的结构	549
17.3	控制寄存器	550
17.3.1	输出目标选择寄存器n (ELSELRn) (n=00~14)	551
17.4	EVENTC的运行	553
第18章	中断功能	554
18.1	中断功能的种类	554
18.2	中断源和结构	554
18.3	控制中断功能的寄存器	559
18.3.1	中断请求标志寄存器 (IF00~IF31)	559
18.3.2	中断屏蔽标志寄存器 (MK00~MK31)	560
18.3.3	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)	563
18.4	中断处理的操作	564
18.4.1	可屏蔽中断请求的接受	564
18.4.2	不可屏蔽中断请求的接受	564
第19章	待机功能	565
19.1	待机功能	565
19.2	睡眠模式	566
19.2.1	睡眠模式的设置	566
19.2.2	睡眠模式的解除	569
19.3	深度睡眠模式	569
19.3.1	深度睡眠模式的设置	569
19.3.2	深度睡眠模式的解除	571

第20章	复位功能.....	573
20.1	确认复位源的寄存器	577
20.1.1	复位控制标志寄存器 (RESF)	577
第21章	上电复位电路	580
21.1	上电复位电路的功能	580
21.2	上电复位电路的结构	581
21.3	上电复位电路的运行	581
第22章	电压检测电路	585
22.1	电压检测电路的功能	585
22.2	电压检测电路的结构	586
22.3	控制电压检测电路的寄存器	587
22.3.1	电压检测寄存器 (LVIM)	587
22.3.2	电压检测电平寄存器 (LVIS)	588
22.4	电压检测电路的运行	591
22.4.1	用作复位模式时的设置	591
22.4.2	用作中断模式时的设置	593
22.4.3	用作中断&复位模式时的设置.....	595
22.5	电压检测电路的注意事项	601
第23章	安全功能.....	603
23.1	安全功能的概要.....	603
23.2	安全功能使用的寄存器.....	604
23.3	安全功能的运行.....	604
23.3.1	闪存CRC运算功能 (高速CRC)	604
23.3.1.1	闪存CRC控制寄存器 (CRC0CTL)	605
23.3.1.2	闪存CRC运算结果寄存器 (PGCRCL)	606
23.3.2	CRC运算功能 (通用CRC)	608
23.3.2.1	CRC输入寄存器 (CRCIN)	609
23.3.2.2	CRC数据寄存器 (CRCD)	610
23.3.3	RAM奇偶校验错误检测功能	611
23.3.4	SFR保护功能	613
23.3.4.1	SFR保护控制寄存器 (SFRGD)	613
23.3.5	频率检测功能	614
23.3.5.1	定时器输入选择寄存器0 (TIS0)	614
23.3.6	A/D测试功能	615
23.3.6.1	A/D测试寄存器 (ADTES)	616
23.3.6.2	模拟输入通道指定寄存器 (ADS)	616
23.3.7	输入/输出引脚的数字输出信号电平检测功能.....	617
23.3.7.1	端口模式选择寄存器 (PMS)	617
23.3.8	产品唯一身份标识寄存器.....	618
第24章	温度传感器	619
24.1	温度传感器的功能	619

24.2	温度传感器的寄存器	619
24.2.1	温度传感器校准数据寄存器 TSCDR1	619
24.2.2	温度传感器校准数据寄存器 TSCDR2	619
24.3	温度传感器的使用说明	620
24.3.1	温度传感器的使用原理	620
24.3.2	温度传感器的使用方法	621
第25章	选项字节	622
25.1	选项字节的功能	622
25.1.1	用户选项字节 (000C0H~000C2H/010C0H~010C2H)	622
25.1.2	闪存数据保护选项字节 (000C3H/010C3H, 500004H~500005H)	623
25.2	用户选项字节的格式	624
25.3	闪存数据保护选项字节的格式	630
第26章	FLASH控制	631
26.1	FLASH控制功能描述	631
26.2	FLASH存储器结构	631
26.3	控制FLASH的寄存器	632
26.3.1	Flash写保护寄存器 (FLPROT)	632
26.3.2	FLASH操作控制寄存器 (FLOPMD1,FLOPMD2)	633
26.3.3	Flash擦除控制寄存器 (FLERMD)	633
26.3.4	Flash状态寄存器 (FLSTS)	634
26.3.5	Flash全片擦除时间控制寄存器 (FLCERCNT)	634
26.3.6	Flash页擦除时间控制寄存器 (FLSERCNT)	635
26.3.7	Flash 写入时间控制寄存器 (FLPROCNT)	636
26.3.8	Flash 模式时间控制寄存器 (FLNVSCNT/FLPRVCNT/FLERVCNT)	637
26.4	FLASH操作方法	639
26.4.1	页擦除 (sector erase)	639
26.4.2	全片擦除 (chip erase)	640
26.4.3	编程 (word program)	640
26.5	闪存读取	640
26.6	FLASH操作的注意事项	640
附录	修订记录	641

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。

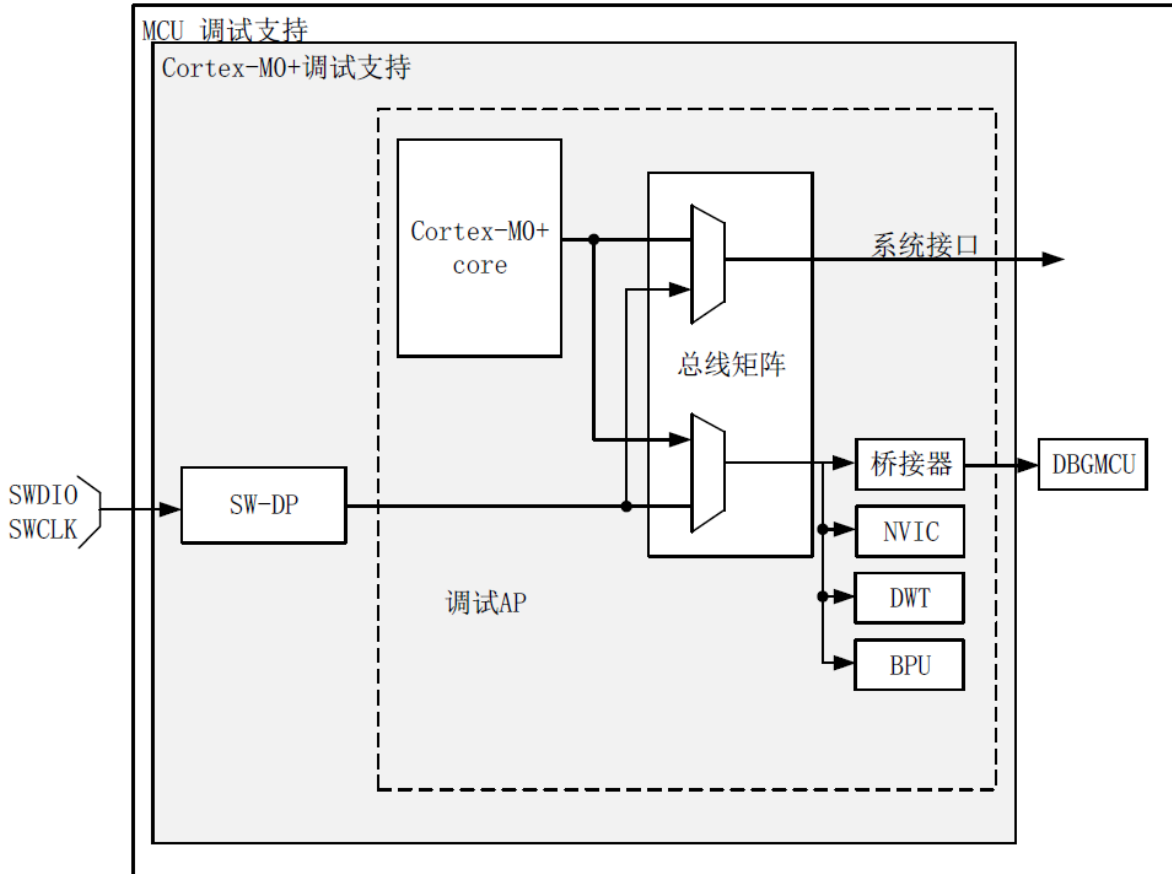
1.2 Cortex-M0+内核特性

- 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fCLK或fIL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
SWDIO	串行数据输入 /输出	P20

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

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

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

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

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

注1：本产品的定时器系周边模块包括：通用定时器单元Timer4

注2：本产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 端口通用功能

每个端口位可以自由编程，通过配置端口控制寄存器PMxx、Pxx、PUxx、PDxx、POMxx、PMCxx可将端口配置成多种通用GPIO模式。

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟通道
- 开漏输出
- 推挽式输出
- 推挽式复用功能
- 开漏复用功能

每个端口分配的寄存器和位因端口硬件特性和产品而不同，请参照表2-1 各产品分配的PMxx、Pxx、PUxx、PDxx、POMxx、PMCxx寄存器及其位”和各产品系列的数据手册。

以24pin产品为例，各端口上电后默认的初始功能如下：

引脚名称	默认功能
P00	外部复位输入
P01	数字IO
P02	数字IO
P10	数字IO
P11	数字IO
P12	数字IO
P13	数字IO
P20	数字IO
P21	数字IO
P22	模拟IO
P23	模拟IO
P24	模拟IO
P25	模拟IO
P26	模拟IO
P30	模拟IO
P31	模拟IO
P32	模拟IO
P33	模拟IO
P34	模拟IO
P35	数字IO
P36	数字IO
P37	数字IO

P00~P02和P10~P13端口只能作为数字通道使用。

P20~P26和P30~P37端口既可作为数字通道使用，也可作为模拟通道使用。当PMCxx配置为1时，端口作为模拟通道使用，本产品所有的模拟功能只绑定到固定端口，不支持重定向配置，参见各产品系列的数据手册。当PMCxx配置为“0”时，端口作为数字通道使用，通过配置PxxCFG寄存器可切换各端口的数字兼用功能，详细配置方法请参照“表 2-2 端口功能配置方法”。

2.2端口复用功能

可通过配置寄存器PMxx，PMCxx，PxxCFG将外设I/O功能通过软件编程重定向到除P00以外的任意端口。参见“2.3.7端口复用功能配置寄存器（PxxCFG）”和各产品系列的数据手册。

2.3控制端口功能的寄存器

通过以下寄存器控制端口。

- 端口模式寄存器 (PMxx)
- 端口寄存器 (Pxx)
- 上拉电阻选择寄存器 (PUxx)
- 下拉电阻选择寄存器 (PDxx)
- 端口输出模式寄存器 (POMx)
- 端口模式控制寄存器 (PMCxx)
- 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。有关各产品分配的寄存器和位，请参照表2-1。必须给未分配的位设置初始值。

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

端口		位名							24 引脚	20 引脚	16 引脚
		PMxx 寄存器	Pxx 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	—	—	PU00	—	—	—	—	○	○	○
	1	PM01	P01	—	—	POM01	—	P01CFG	○	○	○
	2	PM02	P02	—	—	POM02	—	P02CFG	○	○	○
端口1	0	PM10	P10	PU10	—	POM10	—	P10CFG	○	○	○
	1	PM11	P11	PU11	—	POM11	—	P11CFG	○	○	—
	2	PM12	P12	—	—	POM12	—	P12CFG	○	○	○
	3	PM13	P13	—	—	POM13	—	P13CFG	○	○	○
端口2	0	PM20	P20	PU20	PD20	POM20	PMC20	P20CFG	○	○	○
	1	PM21	P21	PU21	PD21	POM21	PMC21	P21CFG	○	○	○
	2	PM22	P22	PU22	PD22	POM22	PMC22	P22CFG	○	—	—
	3	PM23	P23	PU23	PD23	POM23	PMC23	P23CFG	○	—	—
	4	PM24	P24	PU24	PD24	POM24	PMC24	P24CFG	○	○	—
	5	PM25	P25	PU25	PD25	POM25	PMC25	P25CFG	○	○	○
	6	PM26	P26	PU26	PD26	POM26	PMC26	P26CFG	○	○	○
端口3	0	PM30	P30	PU30	PD30	POM30	PMC30	P30CFG	○	○	○
	1	PM31	P31	PU31	PD31	POM31	PMC31	P31CFG	○	○	○
	2	PM32	P32	PU32	PD32	POM32	PMC32	P32CFG	○	○	○
	3	PM33	P33	PU33	PD33	POM33	PMC33	P33CFG	○	—	—
	4	PM34	P34	PU34	PD34	POM34	PMC34	P34CFG	○	—	—
	5	PM35	P35	PU35	PD35	POM35	PMC35	P35CFG	○	○	○
	6	PM36	P36	PU36	PD36	POM36	PMC36	P36CFG	○	○	—
	7	PM37	P37	PU37	PD37	POM37	PMC37	P37CFG	○	○	—

2.3.1 端口模式寄存器 (PMxx)

这是以位为单位设置端口输入/输出的寄存器。在产生复位信号后，这些寄存器的值变为“FFH”。当将端口引脚用作复用功能的引脚时，必须设置此寄存器，并参照“2.3.7端口复用功能配置寄存器 (PxxCFG)”进行设置。

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

图 2-1 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PM0	1	1	1	1	1	PM02	PM01	1	0x320	FFH	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	0x321	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	0x322	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	0x323	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))

注意 必须给未分配的位设置初始值。

2.3.2 端口寄存器 (Pxx)

这是以位为单位设置端口输出锁存器的值的寄存器。在输入模式中读引脚电平，而在输出模式中读端口的输出锁存器的值。在产生复位信号后，这些寄存器的值变为“00H”。当将端口引脚用作复用功能的引脚时，不需要设置此寄存器。

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

图2-2 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	0	P02	P01	P00 ^{注1}	0x300	00H 注1	R/W
P1	0	0	0	0	P13	P12	P11	P10	0x301	00H (输出锁存器)	R/W
P2	0	P26	P25	P24	P23	P22	P21	P20	0x302	00H (输出锁存器)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	0x303	00H (输出锁存器)	R/W

注1：P00为只读，因为P00端口只有输入模式

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 上拉电阻选择寄存器 (PUxx)

内部上拉电阻的选择寄存器。通过设置PUxx可以对使用内部上拉电阻的引脚并且POMmn位为“0”而且设置为输入模式 (PMmn=1) 的位, 以位为单位进行上拉。对于设置为输出模式的引脚, 上拉电阻选择寄存器的设置无效, 不连接内部上拉电阻。当用作复用功能的输出引脚或者设置为模拟功能时也相同。

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

图 2-3 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PU0	0	0	0	0	0	0	0	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	E0H	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 必须给未分配的位设置初始值。

2.3.4 下拉电阻选择寄存器 (PDxx)

内部下拉电阻的选择寄存器。通过设置PDxx可以对使用内部下拉电阻的引脚并且POMmn位为“0”而且设置为输入模式 (PMmn=1) 的位, 以位为单位进行下拉。对于设置为输出模式的引脚, 下拉电阻选择寄存器的设置无效, 不连接内部下拉电阻。当用作复用功能的输出引脚或者设置为模拟功能时也相同。

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

图 2-4 下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
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引脚的输入缓冲器的选择 (m=2、3, n=0~7)
0	不连接内部下拉电阻。
1	连接内部下拉电阻。

注 必须给未分配的位设置初始值。

2.3.5 端口输出模式寄存器 (POMxx)

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

在产生复位信号后，这些寄存器的值变为“00H”。

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

注意 对于设置N沟道漏极开路输出（V_{DD}/EV_{DD}耐压）模式（POMmn=1）的位，不连接内部上/下拉电阻。

图 2-5 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
POM0	0	0	0	0	0	POM02	POM01	0	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（V _{DD} /EV _{DD} 耐压）模式

注 必须给未分配的位设置初始值。

2.3.6 端口模式控制寄存器 (PMCxx)

PMC寄存器以位为单位设置数字输入/输出或者模拟输入。当将端口引脚用作复用功能的引脚时，必须将该端口设置为数字模式。

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

图 2-6 端口模式控制寄存器的格式

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
PMC2	1	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	0x062	fcH	R/W
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	0x063	1fH	R/W

PMCMn	Pmn引脚的数字输入/输出或者模拟输入的选择 (m=2、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

注 必须给未分配的位设置初始值。

2.3.7 端口复用功能配置寄存器 (PxxCFG)

端口复用功能配置寄存器 (PxxCFG) 可实现将外围I/O的数字功能重定向到除P00以外的各端口。端口复用功能配置寄存器的复位值为“00H”。各端口配置寄存器的地址参照下图。

寄存器地址=基址+偏址；

图 2-7 端口复用功能配置寄存器 (PxxCFG) 的格式

基址地址：0x40040800

寄存器名称	偏移地址	读写属性	复位值
P01CFG	0x002	R/W	00H
P02CFG	0x004	R/W	00H
P10CFG	0x010	R/W	00H
P11CFG	0x012	R/W	00H
P12CFG	0x014	R/W	00H
P13CFG	0x016	R/W	00H
P20CFG	0x020	R/W	00H
P21CFG	0x022	R/W	00H
P22CFG	0x024	R/W	00H
P23CFG	0x026	R/W	00H
P24CFG	0x028	R/W	00H
P25CFG	0x02A	R/W	00H
P26CFG	0x02C	R/W	00H
P30CFG	0x030	R/W	00H
P31CFG	0x032	R/W	00H
P32CFG	0x034	R/W	00H
P33CFG	0x036	R/W	00H
P34CFG	0x038	R/W	00H
P35CFG	0x03A	R/W	00H
P36CFG	0x03C	R/W	00H
P37CFG	0x03E	R/W	00H

配置说明：

- 使用端口的数字复用功能时，端口必须配置成数字模式 (PMCxx=0)。
- 对于复用的输入功能，端口必须配置成输入模式 (PMxx=1)。
- 对于复用的输出功能，端口必须配置成输出模式(推挽或开漏) (PMxx=0)。
- 对于双向复用功能，端口必须配置成输出模式(推挽或开漏) (PMxx=0)。这时，输入驱动器被配置成浮空输入模式。
- 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“4.3.1 时钟运行模式控制寄存器 (CMC)”
- 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“4.3.1 时钟运行模式控制寄存器 (CMC)”
- P00 没有端口复用功能，仅能作为 GPIO 输入使用。使用 P00 的 GPIO 输入功能时，需要先配置寄存器 RSTM，将其外部复位功能屏蔽。参照“2.5.1 P00 端口 (默认为 RESETB)”。
- 不能将同一个复用输入功能同时配置到不同端口，不能将不同的复用功能同时配置到同一个端口。

表 2-2 端口功能配置方法

功能名称	输入/输出	PxxCFP	PMCxx	PMxx	POMxx	Pxx	备注
模拟功能	输入/输出	6'h00	1	1	×	×	所有的模拟功能只定向到固定端口, 不支持配置, 参见各产品系列的数据手册
GPIO	输入	6'h00	0	1	0	×	P00仅可作为外部复位输入或GPIO输入使用
	输出		0	0	0	0/1	
	N沟道漏极开路输出		0	0	1	0/1	
INTP0	输入	6'h02	0	1	0	×	
INTP1	输入	6'h03	0	1	0	×	
INTP2	输入	6'h04	0	1	0	×	
INTP3	输入	6'h05	0	1	0	×	
TI00	输入	6'h06	0	1	0	×	
TI01	输入	6'h07	0	1	0	×	
TI02	输入	6'h08	0	1	0	×	
TI03	输入	6'h09	0	1	0	×	
TI10	输入	6'h0a	0	1	0	×	
TI11	输入	6'h0b	0	1	0	×	
TI12	输入	6'h0c	0	1	0	×	
TI13	输入	6'h0d	0	1	0	×	
TO00	输出	6'h0e	0	0	0	×	
TO01	输出	6'h0f	0	0	0	×	
TO02	输出	6'h10	0	0	0	×	
TO03	输出	6'h11	0	0	0	×	
TO10	输出	6'h12	0	0	0	×	
TO11	输出	6'h13	0	0	0	×	
TO12	输出	6'h14	0	0	0	×	
TO13	输出	6'h15	0	0	0	×	
SCLA0	输入/输出	6'h16	0	0	1*	×	*作为IICA端口时, POMxx自动置1, 不需要软件配置
SDAA0	输入/输出	6'h17	0	0	1*	×	*作为IICA端口时, POMxx自动置1, 不需要软件配置
CLKBUZ0	输出	6'h18	0	0	0	×	
CLKBUZ1	输出	6'h19	0	0	0	×	
VCOUT0	输出	6'h1a	0	0	0	×	
VCOUT1	输出	6'h1b	0	0	0	×	
RTC1HZ	输出	6'h1c	0	0	0	×	
SS00	输入	6'h1f	0	1	0	×	
SCLK00	输入	6'h20	0	1	0	×	
	输出		0	0	0	×	
SCLK01	输入	6'h21	0	1	0	×	
	输出		0	0	0	×	
SCLK10	输入	6'h22	0	1	0	×	
	输出		0	0	0	×	
SCLK11	输入	6'h23	0	1	0	×	
	输出		0	0	0	×	
SCLK20	输入	6'h24	0	1	0	×	
	输出		0	0	0	×	
SCLK21	输入	6'h25	0	1	0	×	
	输出		0	0	0	×	
SDI00/RxD0	输入	6'h26	0	1	0	×	
SDI01	输入	6'h27	0	1	0	×	
SDI10/RxD1	输入	6'h28	0	1	0	×	
SDI11	输入	6'h29	0	1	0	×	
SDI20/RxD2	输入	6'h2a	0	1	0	×	
SDI21	输入	6'h2b	0	1	0	×	
SDO00/TxD0	输出	6'h2c	0	0	0	×	
SDO01	输出	6'h2d	0	0	0	×	
SDO10/TxD1	输出	6'h2e	0	0	0	×	
SDO11	输出	6'h2f	0	0	0	×	
SDO20/TxD2	输出	6'h30	0	0	0	×	
SDO21	输出	6'h31	0	0	0	×	

2.4未使用引脚的处理

各未使用引脚的处理如表2-3所示。

表2-3 各未使用引脚的处理

引脚名	输入/输出	未使用时的推荐连接方法
P00	输入	直接或者通过电阻连接 V_{DD} 。(默认为RESETB)
P01~P02	输入/输出	输入：单独通过电阻连接 V_{DD} 或者 V_{SS} 。 输出：置为开路。
P10~P13		
P20~P26		
P30~P37	输入：单独通过电阻连接 EV_{DD} 或者 EV_{SS} 。 输出：置为开路。	

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

2.5特殊引脚的设置

2.5.1 P00端口（默认为RESETB）

P00默认作为外部复位输入引脚使用，低电平时发生系统复位。需要使用P00的GPIO输入功能时，需要先配置寄存器RSTM，将其外部复位功能屏蔽。

基地址：0x40020400

符号	7	6	5	4	3	2	1	0	偏址	复位后	R/W
RSTM	0	0	0	0	0	0	0	RSTM	0x007	00H	R/W

RSTM	P00引脚的外部复位功能屏蔽
0	P00作为外部复位引脚
1	P00作为GPIO输入引脚

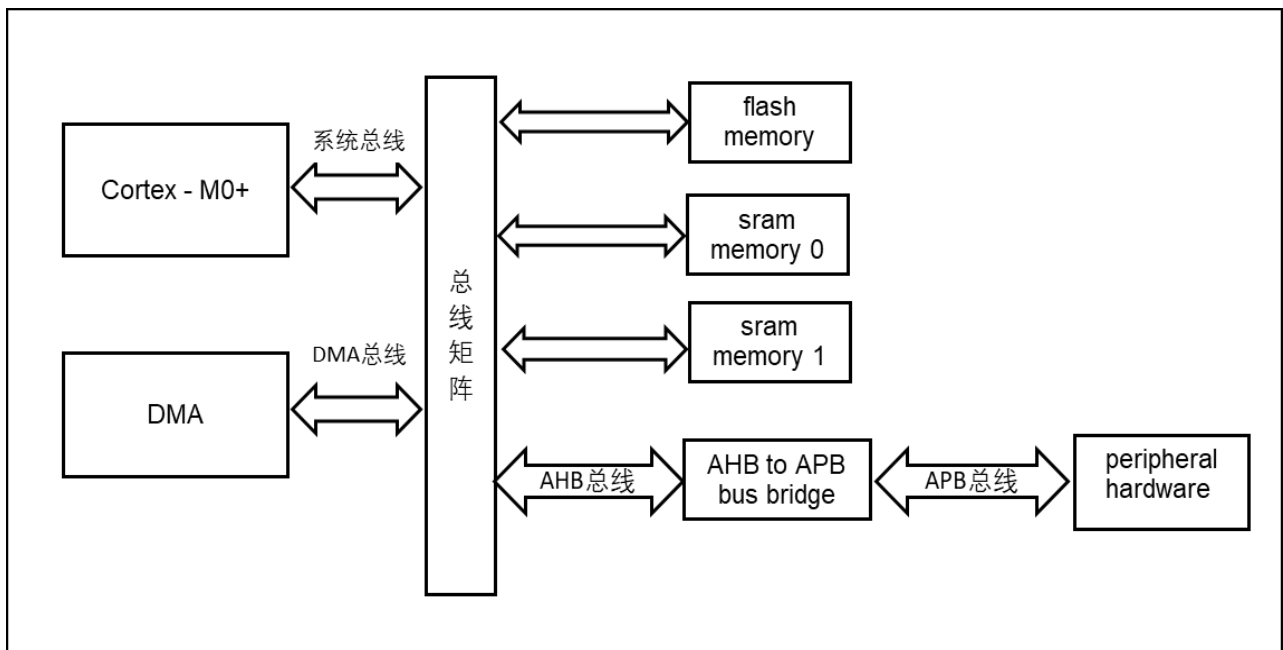
第3章 系统结构

3.1 概述

本产品系统由以下部分组成：

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

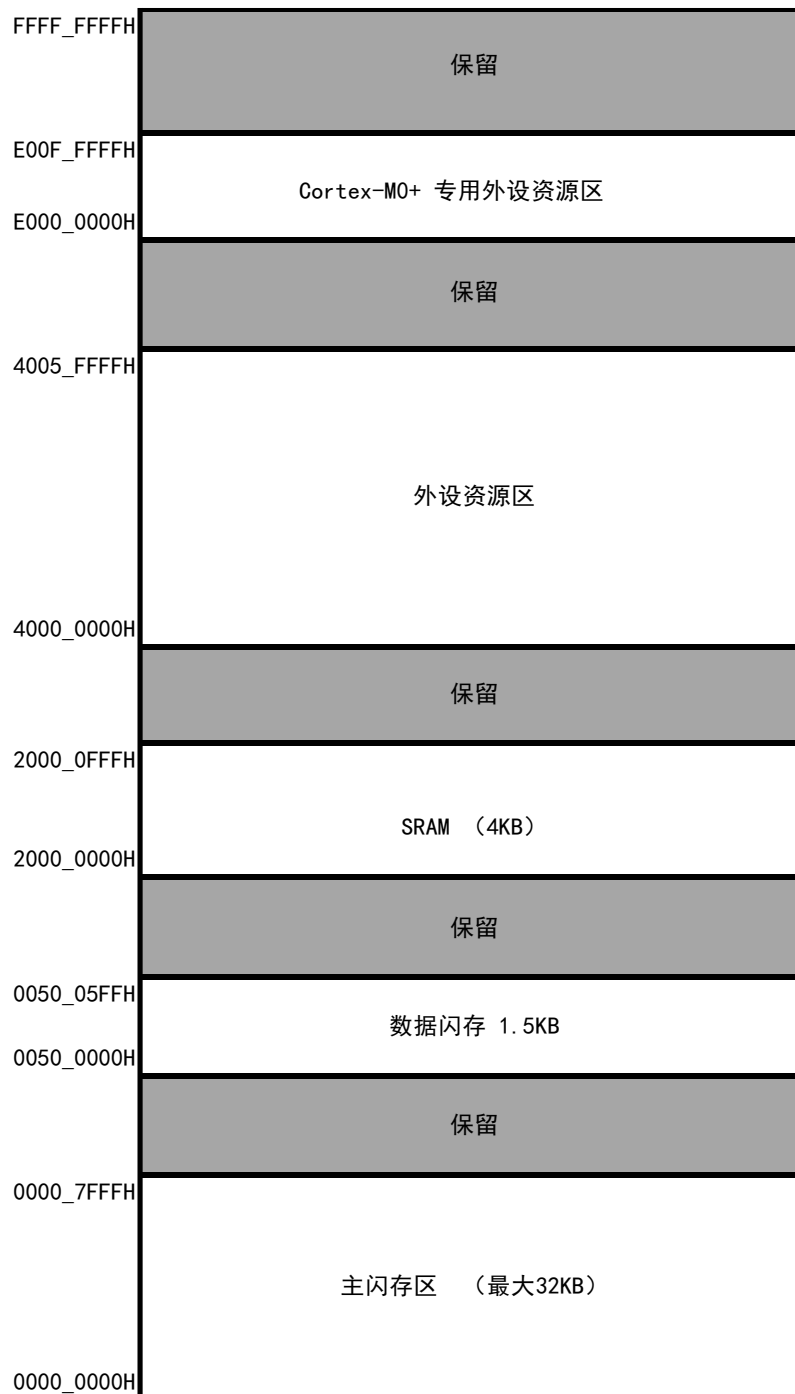
图3-1 系统结构示意图



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

3.2系统地址划分

图3-2 地址区域划分示意图



外设地址分配

表3-1 外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_4FFF	保留	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	中断控制	
0x4000_7000 - 0x4001_8FFF	保留	
0x4001_9000 - 0x4001_9FFF	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 23 章 安全机能
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_25FF	保留	
0x4004_2600 - 0x4004_29FF	保留	
0x4004_2A00 - 0x4004_2BFF	保留	
0x4004_2C00 - 0x4004_2FFF	保留	
0x4004_3000 - 0x4004_31FF	保留	
0x4004_3200 - 0x4004_32FF	通用 CRC	详见第 23 章 安全机能
0x4004_3300 - 0x4004_33FF	保留	
0x4004_3400 - 0x4004_37FF	联动控制器	
0x4004_3800 - 0x4004_3BFF	比较器	
0x4004_3C00 - 0x4004_3FFF	保留	
0x4004_4000 - 0x4004_46FF	IrDA	
0x4004_4700 - 0x4004_4AFF	保留	
0x4004_4B00 - 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	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 $f_X=1\sim 20\text{MHz}$ 的时钟振荡，并且能通过进入深度睡眠模式或者设置MSTOP位（时钟运行状态控制寄存器（CSC）的bit7）使振荡停止。

② 高速内部振荡器（高速OCO）

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

另外，能由EXCLK/X2/P02引脚提供外部主系统时钟（ $f_{\text{EX}}=1\sim 20\text{MHz}$ ），并且能通过进入深度睡眠模式或者设置MSTOP位将外部主系统时钟的输入置为无效。

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

(2) 副系统时钟

• XT1振荡电路

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

另外，能由EXCLKS/XT2/P13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 $f_{IL}=30\text{kHz}/15\text{kHz}$ 的时钟振荡。

不能将低速内部振荡器时钟用作CPU时钟。

SysTick定时器使用低速内部振荡器时钟作为外部参考时钟。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 看门狗定时器
- 实时时钟
- 15位间隔定时器

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

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

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

备注

- f_X ：X1时钟振荡频率
- f_{HOCO} ：高速内部振荡器的时钟频率
- 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

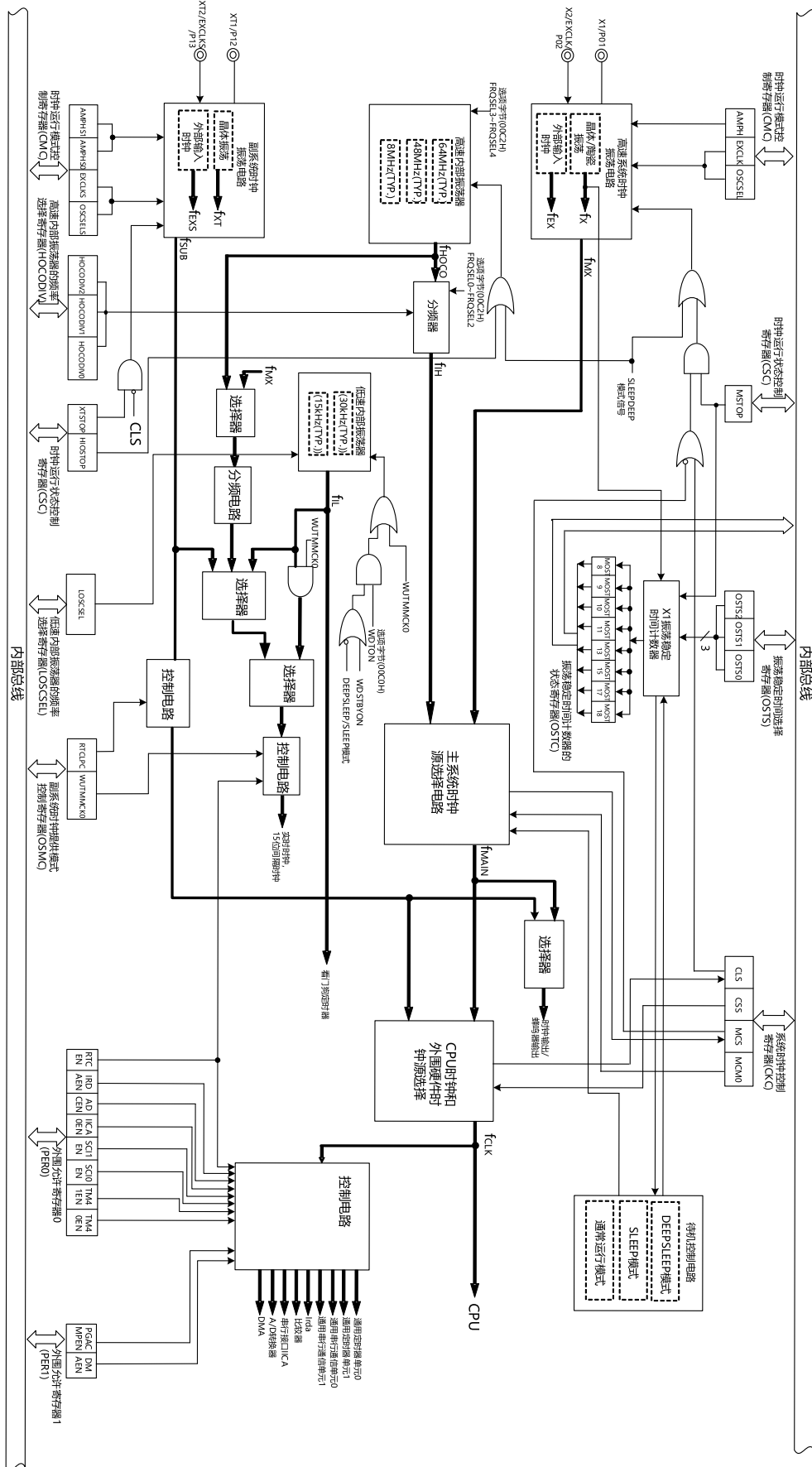
4.2 时钟发生电路的结构

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) 低速内部振荡器的频率选择寄存器 (LOSCSEL) 频率选择寄存器的保护寄存器 (PRCR) 外部复位引脚的复位功能屏蔽寄存器 (RSTM)
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)
- 低速内部振荡器的频率选择寄存器 (LOSCSEL)
- 频率选择寄存器的保护寄存器 (PRCR)
- 外部复位引脚的复位功能屏蔽寄存器 (RSTM)

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/P01、X2/EXCLK/P02、XT1/P12、XT2/EXCLKS/P13引脚的运行模式以及选择振荡电路增益的寄存器。

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

在产生复位信号后，此寄存器的值变为“00H”。

图4-2 时钟运行模式控制寄存器（CMC）的格式

地址：40020400H 复位后：00H R/W

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

EXCLK	OSCSEL	高速系统时钟 引脚的运行模式	X1/P01引脚	X2/EXCLK/P02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/P12引脚	XT2/EXCLKS/P13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	超低功耗振荡
1	1	禁止设置。

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

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

- 注意1.在解除复位后，只能通过8位存储器操作指令写1次CMC寄存器。当以初始值（“00H”）使用CMC寄存器时，为了防止程序失控时的误动作（如果误写“00H”以外的值就不能恢复），必须在解除复位后将CMC寄存器置“00H”。
- 2.在解除复位后并且在通过设置时钟运行状态控制寄存器（CSC）开始X1或者XT1振荡前，必须设置CMC寄存器。
- 3.当X1时钟振荡频率超过10MHz时，必须将AMPH位置“1”。
- 4.必须在解除复位后并且在选择 f_{IH} 作为 f_{CLK} 的状态（将 f_{CLK} 切换为 f_{MX} 或者 f_{SUB} 前的状态）下设置AMPH位、AMPHS1位和AMPHS0位。
- 5.必须通过软件对 f_{XT} 的振荡稳定时间进行计数。
- 6.系统时钟的频率上限为64MHz，但是X1振荡电路的频率上限为20MHz。

备注 f_x : X1时钟振荡频率

4.3.2 系统时钟控制寄存器（CKC）

这是选择CPU/外围硬件时钟和主系统时钟的寄存器。

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

图4-3 系统时钟控制寄存器（CKC）的格式

地址：40020404H 复位后：00H R/W^{注1}

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

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

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

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

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

注 1.bit7和bit5是只读位。

2.禁止在将CSS位置“1”的状态下更改MCM0位的值。

备注 f_{HOCO} : 高速内部振荡器的时钟频率
 f_{IH} : 高速内部振荡器的时钟频率
 f_{MX} : 高速系统时钟频率
 f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率

注意1.必须将bit0~3置“0”。

2.给CPU和外围硬件提供CSS位设置的时钟。如果更改CPU时钟，就同时更改外围硬件的时钟（实时时钟、15位间隔定时器、时钟输出/蜂鸣器输出和看门狗定时器除外）。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。

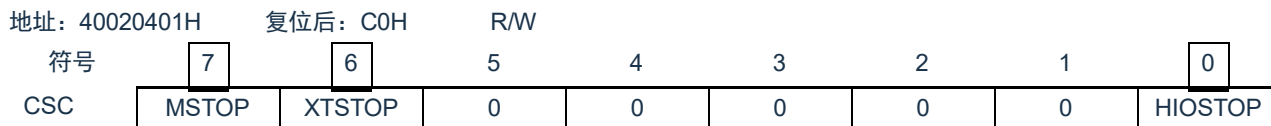
3.如果将副系统时钟用作外围硬件时钟，就无法保证A/D转换器和IICA的运行。有关外围硬件的运行特性，请参照各外围硬件的章节和数据手册。

4.3.3 时钟运行状态控制寄存器（CSC）

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

在产生复位信号后，此寄存器的值变为“C0H”。

图 4-4 时钟运行状态控制寄存器（CSC）的格式



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。
- 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。
 - 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。
 - 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。
 - 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。
 - 有关用于停止时钟振荡（外部时钟输入无效）的寄存器标志设置和停止前的条件，请参照表4-2。

表4-2 时钟停止方法

时钟	时钟停止前的条件（外部时钟输入无效）	CSC寄存器的标志设置
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寄存器。能在以下情况下确认X1时钟的振荡稳定时间：

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

能通过8位存储器操作指令读OSTC寄存器。

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

备注 在以下情况下，振荡稳定时间计数器开始计数：

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

图4-5 振荡稳定时间计数器的状态寄存器（OSTC）的格式



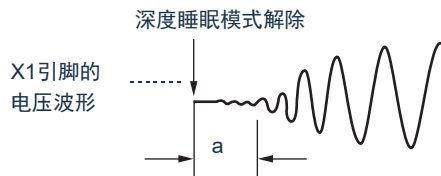
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	振荡稳定时间状态		
								$f_X=10\text{MHz}$	$f_X=20\text{MHz}$	
0	0	0	0	0	0	0	0	小于 $2^8/f_X$	小于25.6us	小于12.8us
1	0	0	0	0	0	0	0	至少 $2^8/f_X$	至少25.6us	至少12.8us
1	1	0	0	0	0	0	0	至少 $2^9/f_X$	至少51.2us	至少25.6us
1	1	1	0	0	0	0	0	至少 $2^{10}/f_X$	至少102us	至少51.2us
1	1	1	1	0	0	0	0	至少 $2^{11}/f_X$	至少204us	至少102us
1	1	1	1	1	0	0	0	至少 $2^{13}/f_X$	至少819us	至少409us
1	1	1	1	1	1	0	0	至少 $2^{15}/f_X$	至少3.27ms	至少1.63ms
1	1	1	1	1	1	1	0	至少 $2^{17}/f_X$	至少13.1ms	至少6.55ms
1	1	1	1	1	1	1	1	至少 $2^{18}/f_X$	至少26.2ms	至少13.1ms

注意1. 在经过上述时间后，各位从MOST8位开始依次变为“1”并且保持“1”的状态。

2. 振荡稳定时间计数器只在振荡稳定时间选择寄存器（OSTS）所设振荡稳定时间内进行计数。在以下情况下，OSTS寄存器的振荡稳定时间的设置值必须大于通过OSTC寄存器确认的计数值。

- 当CPU时钟为高速内部振荡器时钟或者副系统时钟并且要开始X1时钟的振荡时
- 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时（因此必须注意，解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态）

3. X1时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图a）。



备注 f_X : X1时钟振荡频率

4.3.5 振荡稳定时间选择寄存器 (OSTS)

这是选择X1时钟的振荡稳定时间的寄存器。

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

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

能通过OSTC寄存器确认OSTS寄存器事先设置的时间。

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

图4-6 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: 40020403H 复位后: 07H R/W

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

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

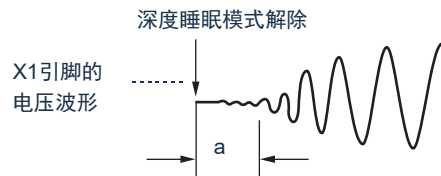
注意1.要更改OSTS寄存器的设置时，必须在将时钟运行状态控制寄存器 (CSC) 的MSTOP位置“0”前进行更改。

2.振荡稳定时间计数器只在OSTS寄存器所设振荡稳定时间内进行计数。

在以下情况下，OSTS寄存器的振荡稳定时间的设置值必须大于在开始振荡后通过OSTC寄存器确认的计数值。

- 当CPU时钟为高速内部振荡器时钟或者副系统时钟并且要开始X1时钟的振荡时
- 当CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到深度睡眠模式后解除深度睡眠模式时 (因此必须注意，解除深度睡眠模式后的OSTC寄存器只设置OSTS寄存器所设振荡稳定时间内的状态)

3.X1时钟的振荡稳定时间不包含时钟开始振荡前的时间 (下图a)。



备注 f_X : X1时钟振荡频率

4.3.6 外围允许寄存器0、1 (PER0、PER1)

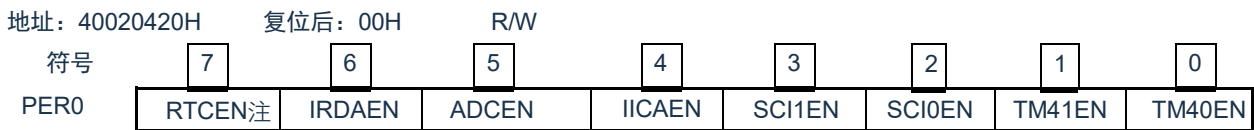
这是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

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

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

通过8位存储器操作指令设置PER0寄存器和PER1寄存器。
在产生复位信号后，这些寄存器的值变为“00H”。

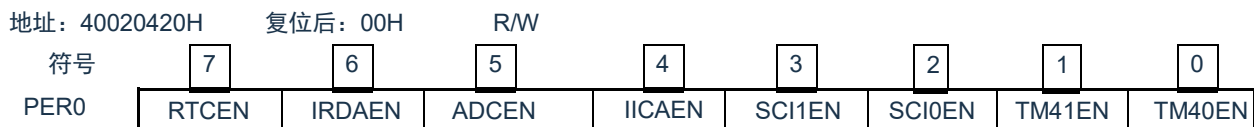
图 4-7 外围允许寄存器0 (PER0) 的格式(1/3)



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

注: RTCEN位只在上电复位时被初始化，而在其他复位时保持不变。

图4-7 外围允许寄存器0 (PER0) 的格式(2/3)



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

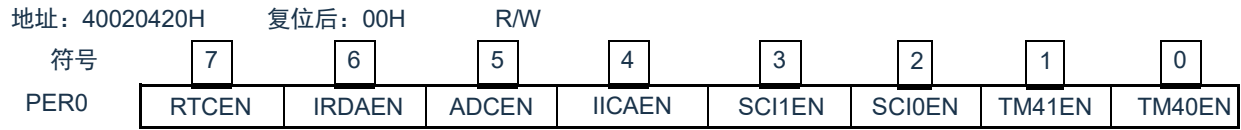
ADCEN	提供A/D转换器的输入时钟的控制
0	停止提供输入时钟。 •不能写A/D转换器使用的SFR。 •A/D转换器处于复位状态。
1	提供输入时钟。 •能读写A/D转换器使用的SFR。

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

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

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

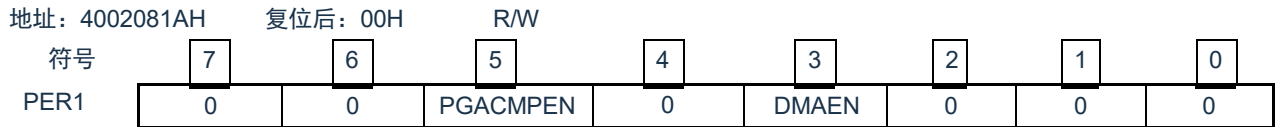
图4-7 外围允许寄存器0 (PER0) 的格式(3/3)



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。

图4-8 外围允许寄存器1 (PER1) 的格式



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

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

4.3.7 副系统时钟提供模式控制寄存器（OSMC）

OSMC寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

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

另外，能通过OSMC寄存器选择实时时钟和15位间隔定时器的运行时钟。

通过8位存储器操作指令设置OSMC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图4-9 副系统时钟提供模式控制寄存器（OSMC）的格式

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

RTCLPC	深度睡眠模式和CPU以副系统时钟运行的睡眠模式中的设置
0	允许给外围功能提供副系统时钟 (有关允许运行的外围功能，请参照表19-1~表19-2)。
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。

图4-10 高速内部振荡器的频率选择寄存器（HOCODIV）的格式

地址：40021C20H	复位后：选项字节（000C2H）的FRQSEL2~FRQSEL0位的设置值						R/W	
符号	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

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

注意1.必须在选择高速内部振荡器时钟（ f_{IH} ）作为CPU/外围硬件时钟（ f_{CLK} ）的状态下设置HOCODIV寄存器。

2.在通过HOCODIV寄存器更改频率后，经过以下转移时间之后进行频率切换：

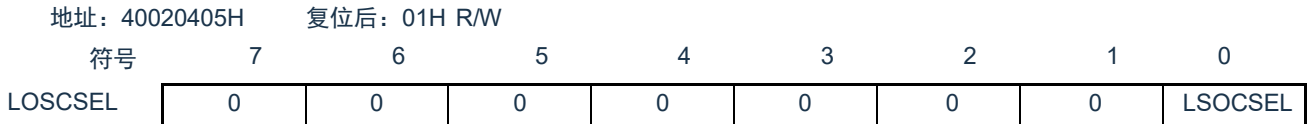
- 以更改前的频率，最多进行3个时钟的运行。
- 以更改后的频率，最多等待3个CPU/外围硬件的时钟。

4.3.9 低速内部振荡器的频率选择寄存器（LOSCSEL）及其保护寄存器（PRCR）

LOSCSEL寄存器是低速内部振荡器的频率选择寄存器。PRCR寄存器是其写保护寄存器。

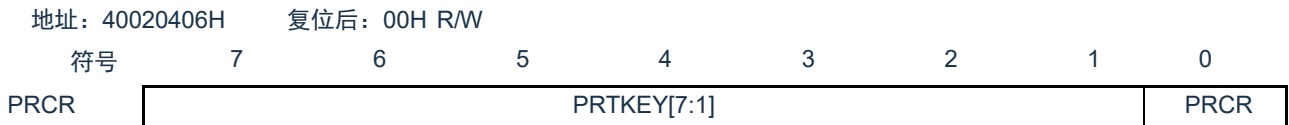
通过8位存储器操作指令设置LOSCSEL，PRCR寄存器。在产生复位信号后，LOSCSEL寄存器的值变为“01H”。PRCR寄存器的值变为“00H”。

图4-11 低速内部振荡器的频率选择寄存器（LOSCSEL）及其保护寄存器（PRCR）的格式(1/2)



LOSCSEL	低速内部振荡器的频率选择
0	•低速内部振荡器频率为15K
1	•低速内部振荡器频率为30K

图4-11 低速内部振荡器的频率选择寄存器（LOSCSEL）及其保护寄存器（PRCR）的格式(2/2)



PRCR	低速内部振荡器的频率选择寄存器写保护
0	•LOSCSEL寄存器不可写
1	•LOSCSEL寄存器可写

PRTKEY[7:1]	PRCR的写保护
78H	•PRCR可写
其它	•PRCR不可写

4.3.10 外部复位引脚的复位功能屏蔽寄存器 (RSTM)

RSTM寄存器是外部复位引脚的复位功能屏蔽寄存器。

通过8位存储器操作指令设置该寄存器。

在产生复位信号后，该寄存器的值变为“00H”。

图4-12 外部复位引脚的复位功能屏蔽寄存器 (RSTM)

地址: 40020407H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RSTM	0	0	0	0	0	0	0	RSTM

RSTM	外部复位引脚的复位功能屏蔽寄存器
0	•外部复位引脚的复位功能不屏蔽，P00作为复位端子使用
1	•外部复位引脚的复位功能屏蔽，P00作为GPIO使用

4.4 系统时钟振荡电路

4.4.1 X1振荡电路

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

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

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

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

X1振荡电路的外接电路例子如图4-13所示。

图4-13 X1振荡电路的外接电路例子



注意事项如下页所示。

4.4.2 XT1振荡电路

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

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

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

当不使用XT1振荡电路时，必须设置为输入端口模式（EXCLKS、OSCSELS=0、0）。而且，当也不用作输入端口时，请参“表2-3 各未使用引脚的处理”。XT1振荡电路的外接电路例子如图4-14所示。

图4-14 XT1振荡电路的外接电路例子

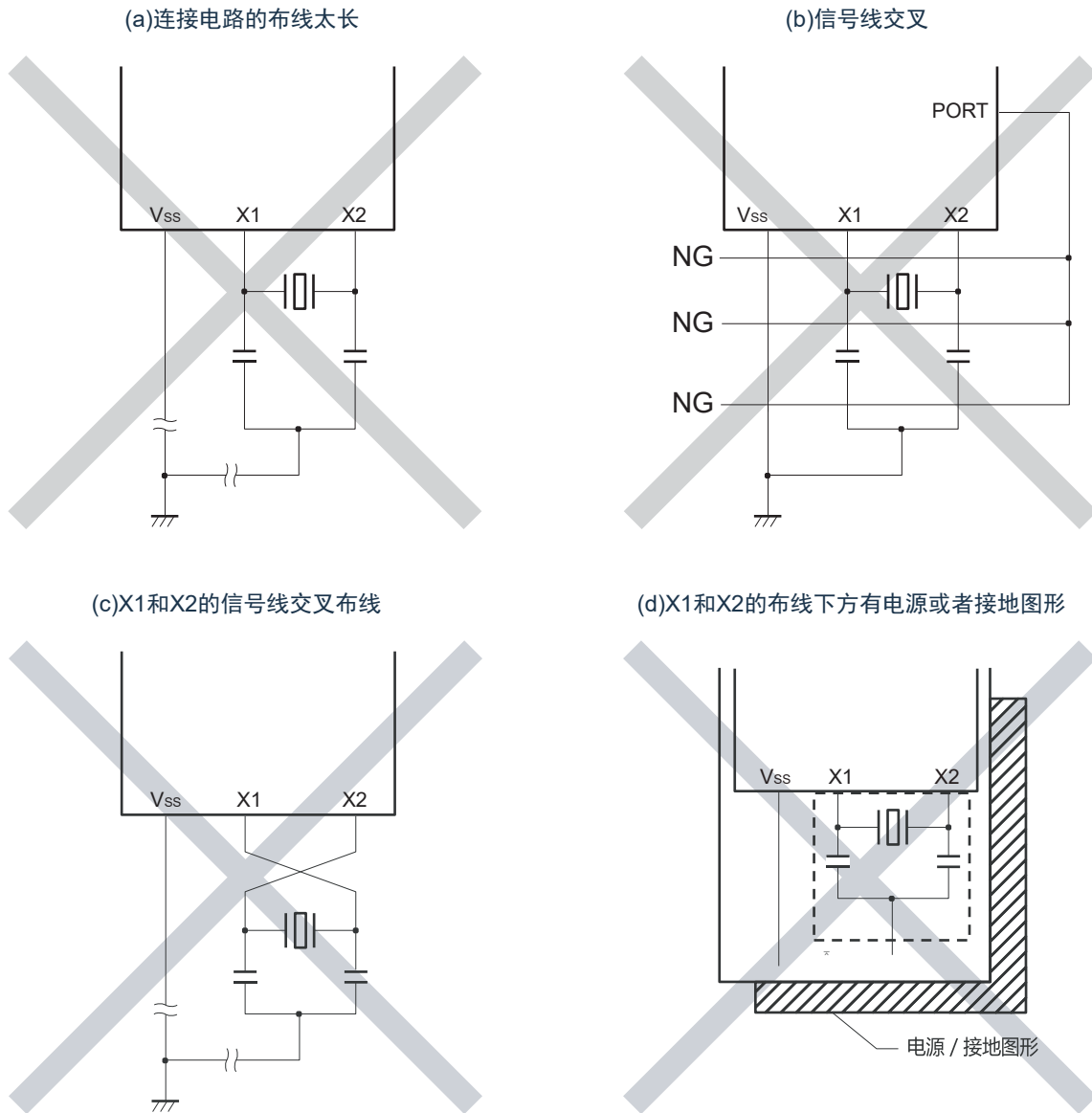


注意 当使用X1振荡电路和XT1振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图4-13和图4-14中的虚线部分进行布线：

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

不正确的谐振器连接例子如图4-15所示。

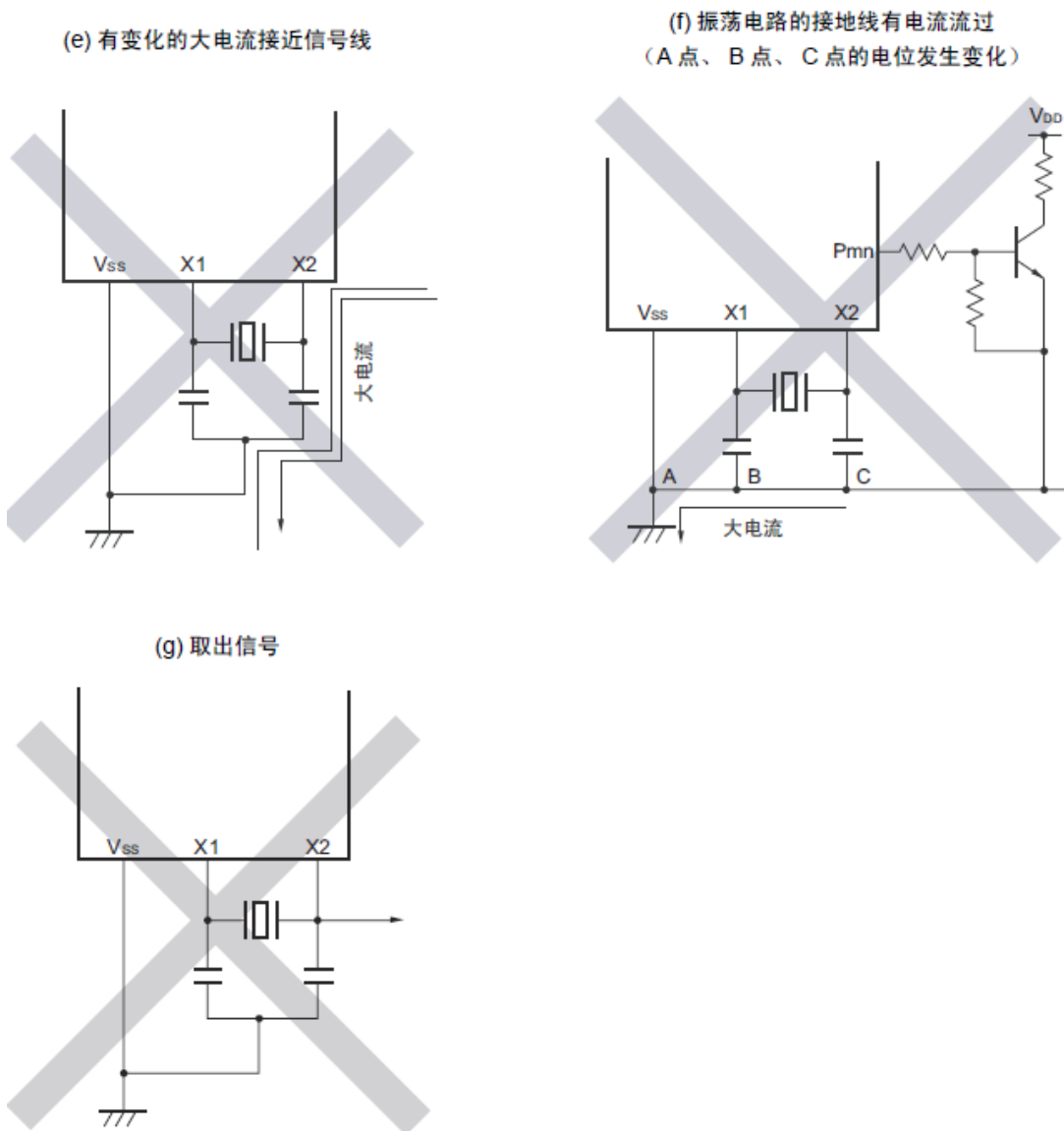
图4-15 不正确的谐振器连接例子(1/2)



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

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

图 4-15 不正确的谐振器连接例子(2/2)



注意 当X2和XT1并行布线时，X2的串扰噪声会叠加到XT1而导致误动作。

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

4.4.3 高速内部振荡器

BAT32G133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

BAT32G133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，但是不能用作CPU时钟。

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

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

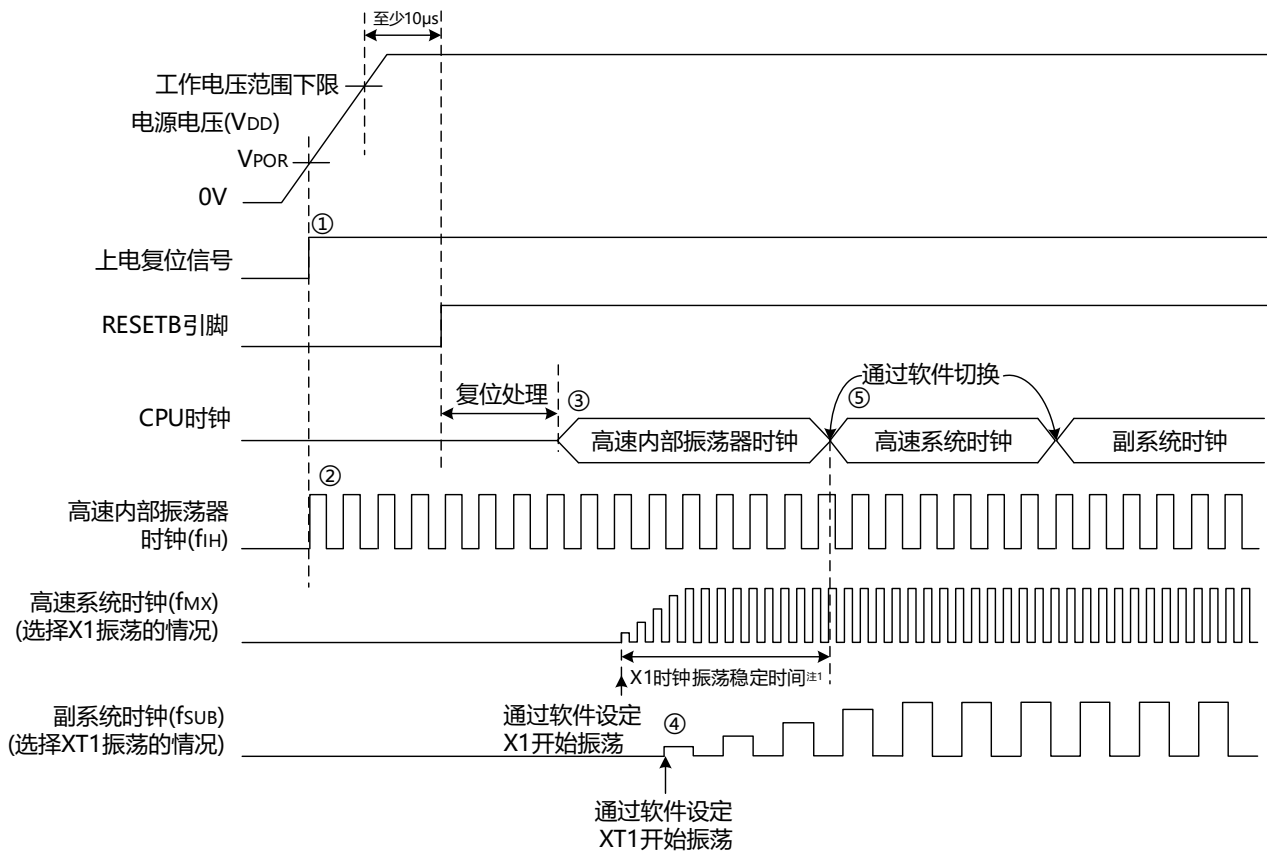
4.5 时钟发生电路的运行

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

- 主系统时钟 f_{MAIN}
 - 高速系统时钟 f_{MX}
 - X1时钟 f_{X}
 - 外部主系统时钟 f_{EX}
 - 高速内部振荡器时钟 f_{IH}
- 副系统时钟 f_{SUB}
 - XT1时钟 f_{XT}
 - 外部副系统时钟 f_{EXS}
- 低速内部振荡器时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

BAT32G133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4-16所示。

图4-16 接通电源时的时钟发生电路的运行



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

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

注意 如果使用EXCLK引脚输入的外部时钟，就不需要振荡稳定等待时间。

4.6时钟控制

4.6.1 高速内部振荡器的设置例子

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

【选项字节的设置】

地址: 000C2H

选项字节

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

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
					f_{HOCO}	f_{IH}
1	0	0	0	0	64MHz	64MHz
0	1	0	0	0	48MHz	48MHz
1	0	0	0	1	64MHz	32MHz
0	1	0	0	1	48MHz	24MHz
1	0	0	1	0	64MHz	16MHz
0	1	0	1	0	48MHz	12MHz
1	0	0	1	1	64MHz	8MHz
0	1	0	1	1	48MHz	6MHz
1	0	1	0	0	64MHz	4MHz
0	1	1	0	0	48MHz	3MHz
1	0	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=0		FRQSEL4=1	
			FRQSEL3=0	FRQSEL3=1	FRQSEL3=0	FRQSEL3=1
0	0	0	禁止设置。	$f_{IH}=48\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=64\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
0	0	1	禁止设置。	$f_{IH}=24\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
0	1	0	禁止设置。	$f_{IH}=12\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=16\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
0	1	1	禁止设置。	$f_{IH}=6\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=8\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
1	0	0	禁止设置。	$f_{IH}=3\text{MHz}$ $f_{HOCO}=48\text{MHz}$	$f_{IH}=4\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
1	0	1	禁止设置。	禁止设置。	$f_{IH}=2\text{MHz}$ $f_{HOCO}=64\text{MHz}$	禁止设置。
上述以外			禁止设置。			

4.6.2 X1振荡电路的设置例子

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

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

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

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

②通过OSTS寄存器选择解除深度睡眠模式时的X1振荡电路的振荡稳定时间。

例)要通过10MHz谐振器至少等待102us时，必须设置为以下的值。

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

③将CSC寄存器的MSTOP位清“0”，使X1振荡电路开始振荡。

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP						HIOSTOP
	0	1	0	0	0	0	0	0

④通过OSTC寄存器等待X1振荡电路的振荡稳定。

例)要通过10MHz谐振器至少等待102us时，必须等到各位变为以下的值。

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	1	1	0	0	0	0	0

⑤通过CKC寄存器的MCM0位将X1振荡时钟设置为CPU/外围硬件时钟。

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

4.6.3 XT1振荡电路的设置例子

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

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

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

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 0	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	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 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	MSTOP 1	XTSTOP 0	0	0	0	0	0	HIOSTOP 0

- 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。

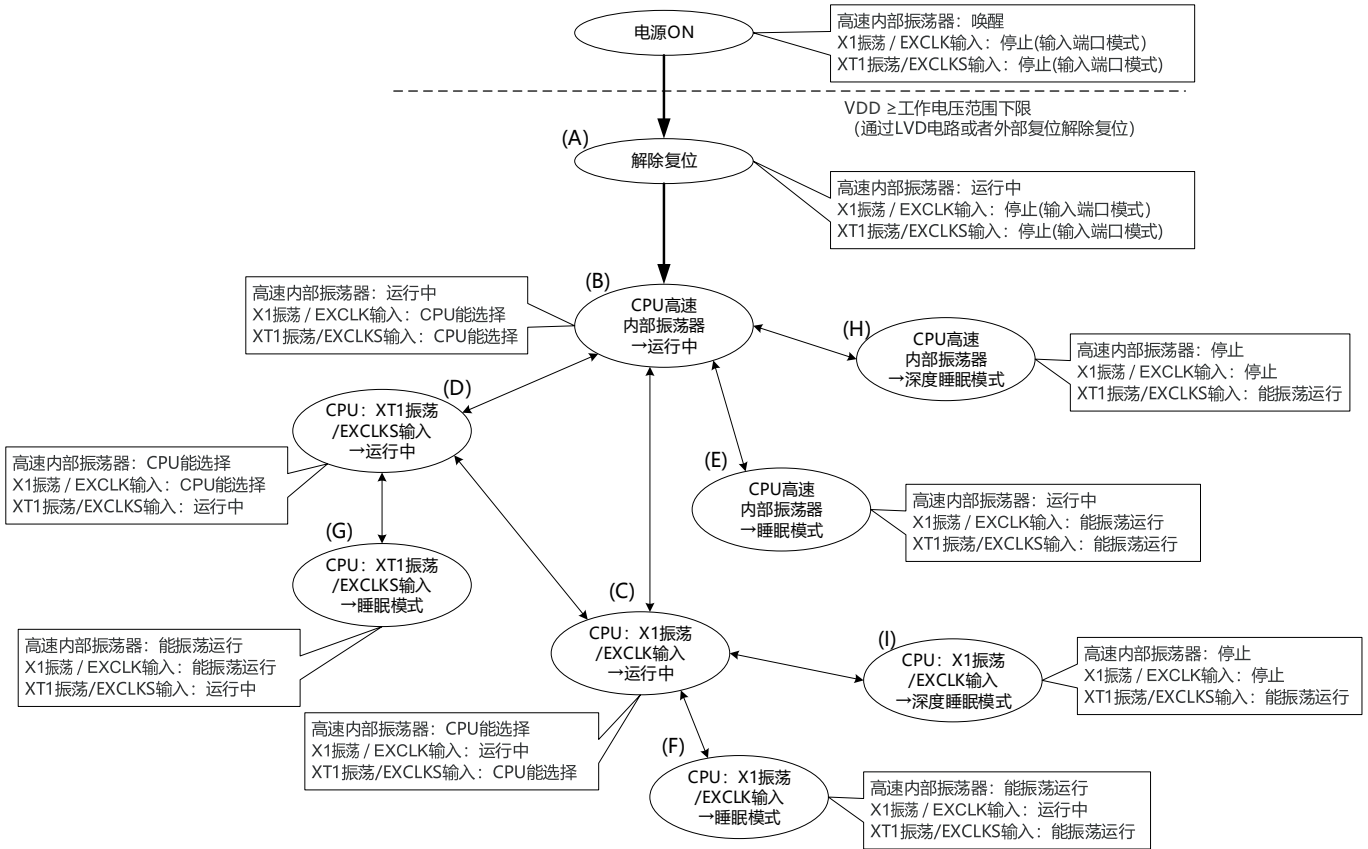
- ⑤通过CKC寄存器的CSS位将XT1振荡时钟设置为CPU/外围硬件时钟。

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	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4-17所示。

图4-17 CPU时钟的状态转移图



CPU时钟的转移和SFR寄存器的设置例子等如表4-3所示。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(1/5)

(1)在解除复位(A)后, CPU转移到高速内部振荡器时钟运行(B)。

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

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

(SFR寄存器的设置顺序) →

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

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

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

(SFR寄存器的设置顺序) →

状态转移	SFR寄存器的设置标志				CMC寄存器注		CSC寄存器	振荡稳定的等待	CKC寄存器
	EXCLKS	OSCSELS	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	×	×	0	需要	1		

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

- 备注1.×: 忽略
2.表4-3的(A)~(I)对应图4-17的(A)~(I)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(2/5)

(4)CPU从高速内部振荡器时钟运行(B)转移到高速系统时钟运行(C)。

(SFR寄存器的设置顺序) →

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

如果已设置就不需要。

在高速系统时钟运行中不需要。

- 注 1.在解除复位后,只能设置1次时钟运行模式控制寄存器(CMC)。如果已设置就不需要。
 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(参照用户手册)后设置时钟。

(5)CPU从高速内部振荡器时钟运行(B)转移到副系统时钟运行(D)。

(SFR寄存器的设置顺序) →

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

如果已设置就不需要。

在副系统时钟运行中不需要。

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

备注1.×: 忽略

2.表4-3的(A)~(I)对应图4-17的(A)~(I)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(3/5)

(6)CPU从高速系统时钟运行(C)转移到高速内部振荡器时钟运行(B)。

(SFR寄存器的设置顺序) →

SFR寄存器的设置标志	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。

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

(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) →

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) →

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。

备注1.表4-3的(A)~(I)对应图4-17的(A)~(I)。

2.高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(4/5)

(9)CPU从副系统时钟运行(D)转移到高速系统时钟运行(C)。

(SFR寄存器的设置顺序) →

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

在高速系统时钟运行中不需要。

注 必须对振荡稳定时间选择寄存器 (OSTS) 的振荡稳定时间进行以下的设置
 •期待的振荡稳定时间计数器的状态寄存器 (OSTC) 的振荡稳定时间 ≤ 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。

(10)• CPU在高速内部振荡器时钟运行中(B)转移到睡眠模式(E)。


- CPU在高速系统时钟运行中(C)转移到睡眠模式(F)。
- CPU在副系统时钟运行中(D)转移到睡眠模式(G)。

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

备注 表4-3的(A)~(I)对应图4-17的(A)~(I)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(5/5)

- (11)• CPU在高速内部振荡器时钟运行中(B)转移到深度睡眠模式(H)。
- CPU在高速系统时钟运行中(C)转移到深度睡眠模式(I)。

(设置顺序) 

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

备注 表4-3的(A)~(I)对应图4-17的(A)~(I)。

4.6.5 CPU时钟转移前的条件和转移后的处理

CPU时钟转移前的条件和转移后的处理如下所示。

表4-4 有关CPU时钟的转移(1/2)

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

表4-4 有关CPU时钟的转移(2/2)

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

4.6.6 CPU时钟和主系统时钟的切换所需时间

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

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

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

如果切换CPU时钟,就同时切换外围硬件时钟。

表4-5 切换主系统时钟所需要的最长时间

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

表4-6 f_{IH} f_{MX} 所需要的最大时钟数

切换前的设置值		切换后的设置值	
MCM0		MCM0	
		0 ($f_{MAIN}=f_{IH}$)	1 ($f_{MAIN}=f_{MX}$)
0 ($f_{MAIN}=f_{IH}$)	$f_{MX} \geq f_{IH}$		2个时钟
	$f_{MX} < f_{IH}$		2个 f_{IH}/f_{MX} 时钟
1 ($f_{MAIN}=f_{MX}$)	$f_{MX} \geq f_{IH}$	2个 f_{MX}/f_{IH} 时钟	
	$f_{MX} < f_{IH}$	2个时钟	

表4-7 f_{MAIN} f_{SUB} 所需要的最大时钟数

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

备注1.表4-6和表4-7中的时钟数是切换前的CPU时钟数。

2.表4-6和表4-7中的时钟数是舍入小数部分的时钟数。

例:主系统时钟从高速系统时钟切换到高速内部振荡器时钟的情况(选择 $f_{IH}=8\text{MHz}$ 、 $f_{MX}=10\text{MHz}$ 振荡的情况)
 $2f_{MX}/f_{IH}=2(10/8)=2.5$ 3个时钟

4.6.7 时钟振荡停止前的条件

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

表4-8 时钟振荡停止前的条件和标志设置

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

第5章 通用定时器单元Timer4

本产品搭载两个通用定时器单元,每个单元含有4个通道。本产品的引脚兼用功能可配置到除P00以外的任意端口,因此通用定时器单元的通道数可灵活配置。

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)。
- 3、本章的下述内容主要针对24引脚产品进行说明。

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

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

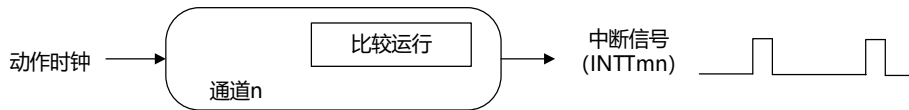
通用定时器单元有以下功能：

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。

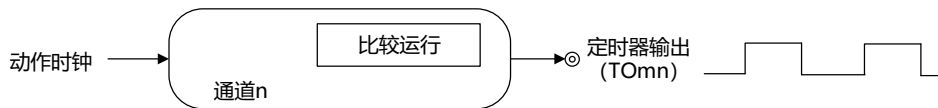
(1) 间隔定时器

能用作以固定间隔产生中断（INTTMmn）的基准定时器。



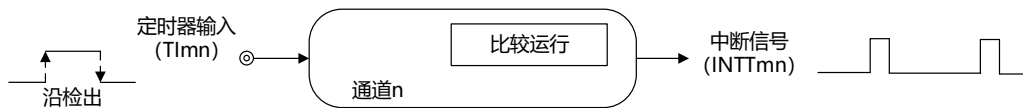
(2) 方波输出

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。



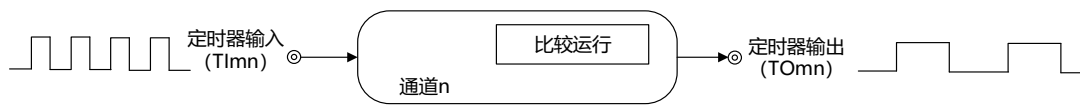
(3) 外部事件计数器

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

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



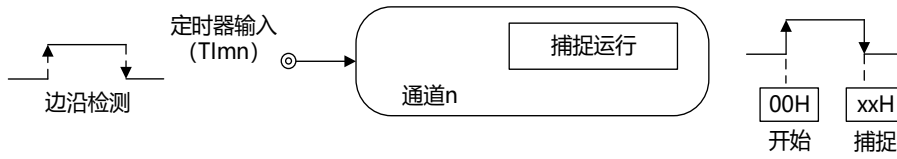
(5) 输入脉冲间隔的测量

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。



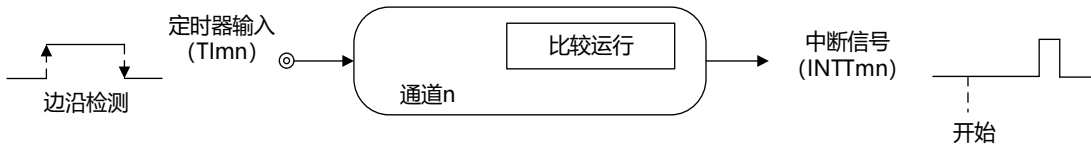
(6) 输入信号的高低电平宽度的测量

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)

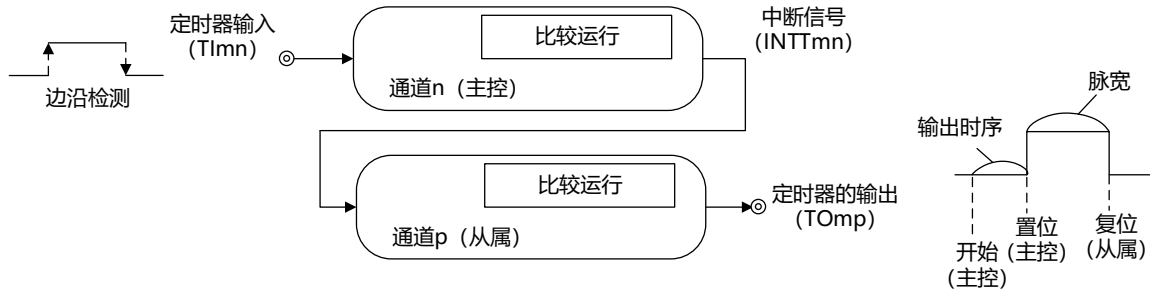
5.1.2 多通道联动运行功能

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。

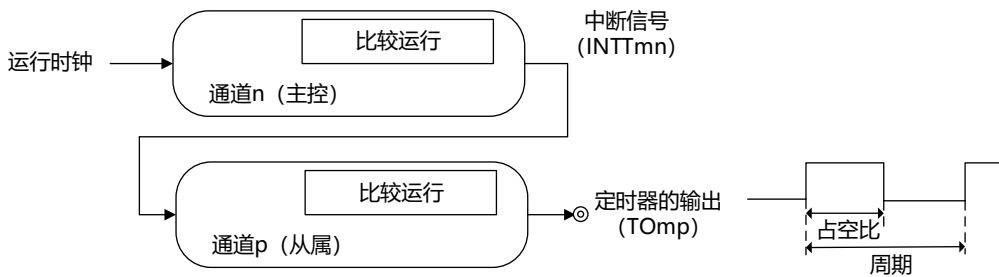
(1) 单触发脉冲输出

将2个通道成对使用，生成能任意设置输出时序和脉宽的单触发脉冲。



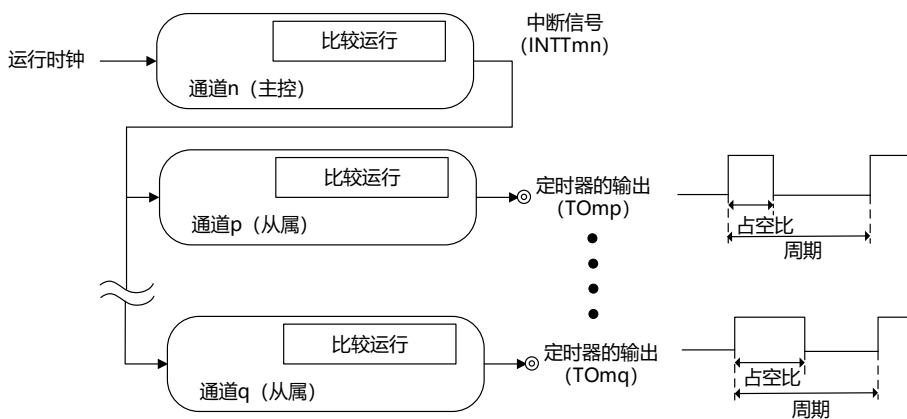
(2) PWM (Pulse Width Modulation) 输出

将2个通道成对使用，生成能任意设置周期和占空比的脉冲。



(3) 多重PWM (Pulse Width Modulation) 输出

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



注意 有关多通道联动运行功能规则的详细内容，请参照5.4.1多通道联动运行功能的基本规则。

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

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。只能使用单元0的通道1和通道3。

注意 在使用8位定时器运行功能时，有几个规则。

详细内容请参照“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）”。

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 ^{注1} 、TI10~TI13 ^{注1}
定时器的输出	TO00~TO03 ^{注1} 、TO10~TO13 ^{注1} 、输出控制电路
控制寄存器	<单元设置部的寄存器> <ul style="list-style-type: none"> • 外围允许寄存器0 (PER0) • 定时器时钟选择寄存器m (TPSm) • 定时器通道允许状态寄存器m (TEm) • 定时器通道开始寄存器m (TSM) • 定时器通道停止寄存器m (TTm) • 定时器输入选择寄存器0 (TIOS0)^{注2} • 定时器输出允许寄存器m (TOEm) • 定时器输出寄存器m (TOM) • 定时器输出电平寄存器m (TOLm) • 定时器输出模式寄存器m (TOMm)
	<每个通道的寄存器> <ul style="list-style-type: none"> • 定时器模式寄存器mn (TMRmn) • 定时器状态寄存器mn (TSRmn) • 噪声滤波器允许寄存器1、2 (NFEN1、NFEN2) • 端口模式控制寄存器 (PMCxx)^{注3} • 端口模式寄存器 (PMxx)^{注3} • 端口复用功能配置寄存器 (PxxCFG)^{注3}

注1: 通道0~3的定时器输入/输出引脚可任意配置到除P00以外的各端口。详细内容请参照“第2章 引脚功能 2.3.7”。

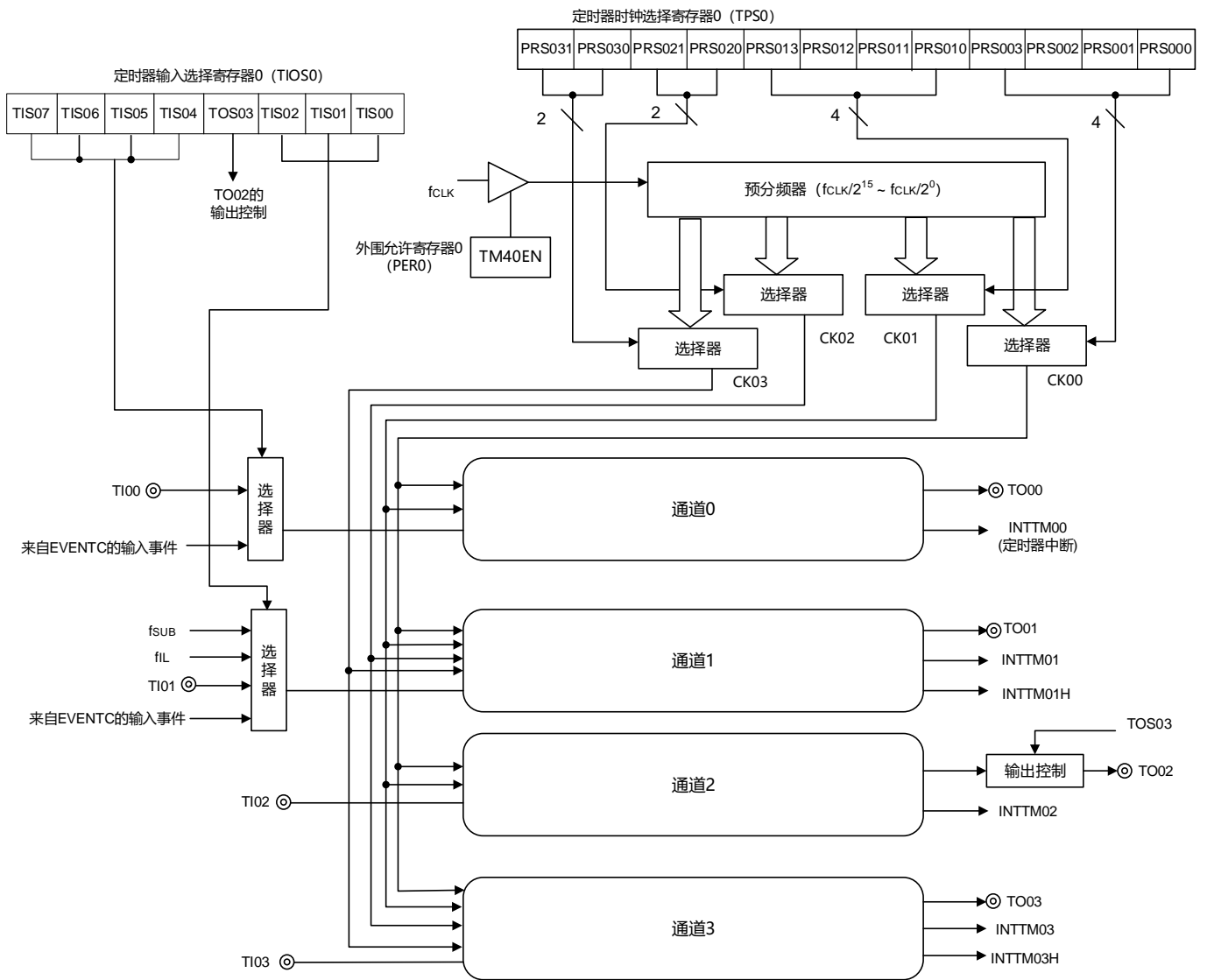
注2: 仅用于单元0的通道选择

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2.3.7”。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

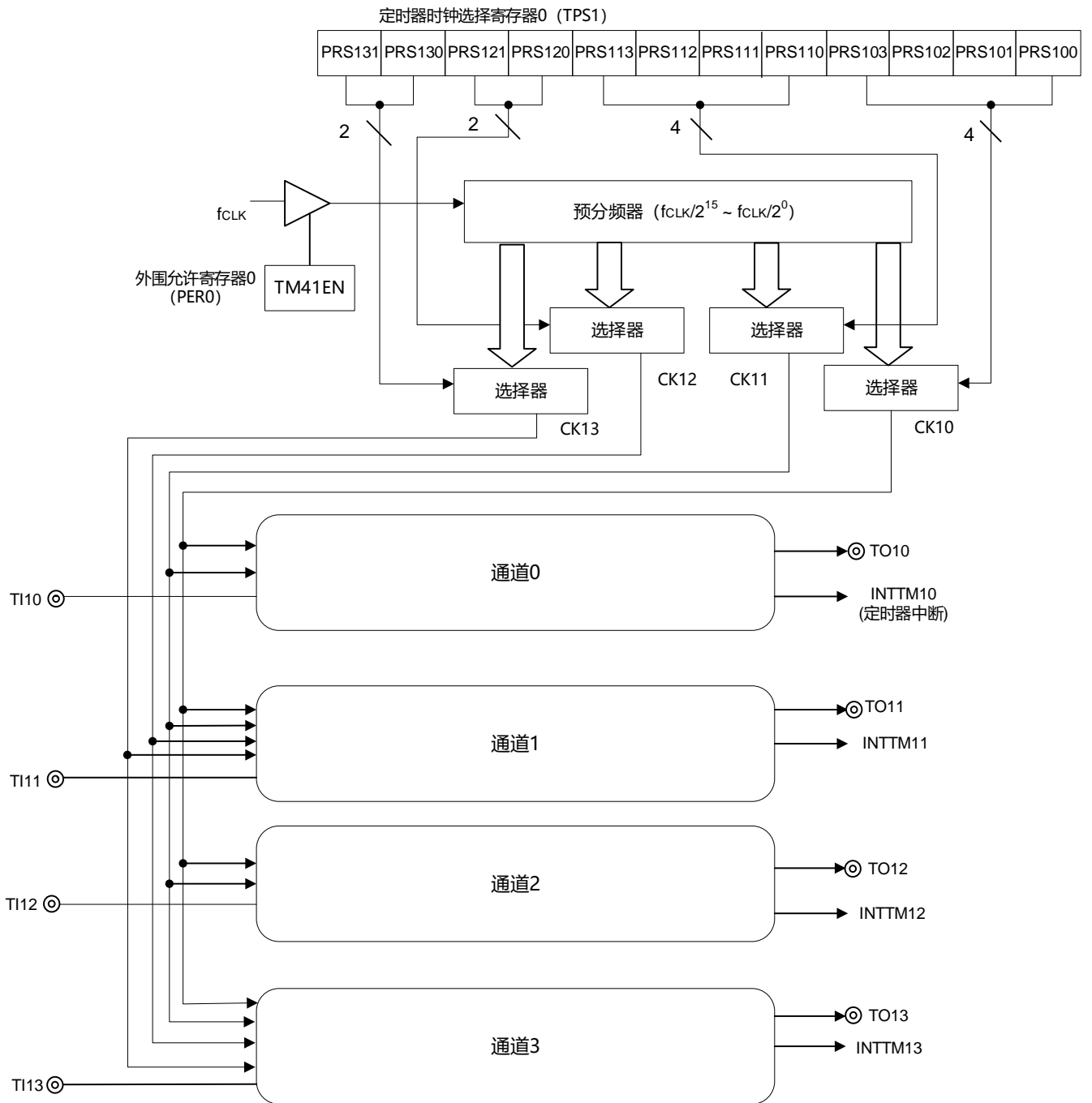
通用定时器单元的框图如图5-1所示。

图5-1 通用定时器单元0的整体框图



备注 f_{SUB} : 副系统时钟频率
 f_{IL} : 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)”）。

图5-3 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																

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

能通过读定时器计数寄存器mn (TCRmn) 来读计数值。

在以下情况下，计数值变为“FFFFH”。

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

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

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

如下所示，TCRmn寄存器的读取值因运行模式和运行状态而不同。

表5-2 各运行模式中的定时器计数寄存器mn (TCRmn) 的读取值

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

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

备注 m: 单元号 (m=0, 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”。

图5-4 定时器数据寄存器mn (TDRmn) (n=0、2) 的表格

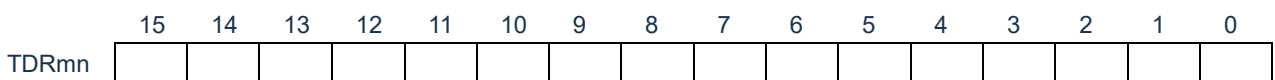
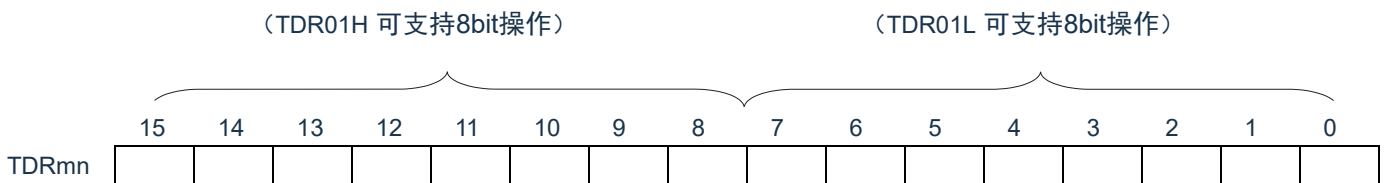


图5-5 定时器数据寄存器mn (TDRmn) (n=1、3) 的表格



- (i) 定时器数据寄存器mn (TDRmn) 用作比较寄存器的情况
从TDRmn寄存器的设置值开始递减计数，当计数值变为“0000H”时，产生中断信号 (INTTMmn)。保持TDRmn寄存器的值，直到被改写为止。

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

- (ii) 定时器数据寄存器mn (TDRmn) 用作捕捉寄存器的情况
通过输入捕捉触发，将定时器计数寄存器mn (TCRmn) 的计数值捕捉到TDRmn寄存器。
能选择TIMn引脚的有效边沿作为捕捉触发信号。通过定时器模式寄存器mn (TMRmn) 来设置捕捉触发的选择。

备注 m: 单元号 (m=0, 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置初始值。

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

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

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

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。 <ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 能读写通用定时器单元1使用的SFR。

注意1.要设置通用定时器单元时，必须先在TM4mEN位为“1”的状态下设置以下的寄存器。当TM4mEN位为“0”时，定时器阵列单元的控制寄存器的值为初始值，忽视写操作（定时器输入输出选择寄存器0 (TIOS0)、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。

能改写PRSm00~PRSm03位的情况（n=0~3）：

选择CKm0作为运行时钟（CKSmn1、CKSmn0=0、0）的通道全部处于停止状态（TEmn=0）。

能改写PRSm10~PRSm13位的情况（n=0~3）：

选择CKm2作为运行时钟（CKSmn1、CKSmn0=0、1）的通道全部处于停止状态（TEmn=0）。

能改写PRSm20位和PRSm21位的情况（n=1、3）：

选择CKm1作为运行时钟（CKSmn1、CKSmn0=1、0）的通道全部处于停止状态（TEmn=0）。

能改写PRSm30位和PRSm31位的情况（n=1、3）：

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

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

图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	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	运行时钟 (CKmk) 的选择注 (k=0、1)
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

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

注意

1. 必须将bit15、14、11、10置“0”。
2. 如果选择 f_{CLK} （无分频）作为运行时钟（CKmk）并且将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。

备注

1. f_{CLK} : CPU/外围硬件的时钟频率
2. TPSm寄存器选择的时钟波形从上升沿开始只有1个 f_{CLK} 周期为高电平。详细内容请参照“5.5.1计数时钟（fTCLK）”。

图5-8 定时器时钟选择寄存器m (TPSm) 的表格(2/2)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSm21	PRSm20	运行时钟 (CKm2) 的选择注
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) 的选择注
0	0	$f_{CLK}/2^8$
0	1	$f_{CLK}/2^{10}$
1	0	$f_{CLK}/2^{12}$
1	1	$f_{CLK}/2^{14}$

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

注意 必须将bit15、14、11、10置“0”。

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

表5-3运行时钟CKSm2和CKSm3能设置的间隔时间

时钟		间隔时间注 ($f_{CLK}=32\text{MHz}$)			
		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 ○包含5%以内的误差。

备注

1. f_{CLK} : CPU/外围硬件的时钟频率

2.有关TPSm寄存器所选 $f_{CLK}/2^r$ 波形的详细内容，请参照“5.5.1计数时钟（fTCLK）”。

5.3.3 定时器模式寄存器mn (TMRmn)

TMRmn寄存器是设置通道n运行模式的寄存器，进行运行时钟（fMCK）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置、定时器输入有效边沿的选择以及运行模式（间隔、捕捉、事件计数器、单次计数、捕捉&单次计数）的设置。

禁止在运行中（TEmn=1）改写TMRmn寄存器。但是，能在一部分的功能运行中（TEmn=1）改写bit7和bit6（CISmn1、CISmn0）（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通用定时器单元的多通道联动运行功能”）。

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

注意 TMRmn寄存器的bit11因通道而不同。

TMRm2 : MASTERmn位 (n=2)
TMRm1、TMRm3 : SPLITmn位 (n=1、3)
TMRm0 : 固定为“0”。

图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	0	MD mn3	MD mn2	MD mn1	MD mn0

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

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

CKSmn1	CKSmn0	通道n运行时钟 (f_{MCK}) 的选择
0	0	定时器时钟选择寄存器m (TPSm) 设置的运行时钟CKm0
0	1	定时器时钟选择寄存器m (TPSm) 设置的运行时钟CKm2
1	0	定时器时钟选择寄存器m (TPSm) 设置的运行时钟CKm1
1	1	定时器时钟选择寄存器m (TPSm) 设置的运行时钟CKm3
运行时钟 (f_{MCK}) 用于边沿检测电路。通过设置CCSmn位来产生采样时钟和计数时钟 (f_{TCLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。		

CCSmn	通道n计数时钟 (f_{TCLK}) 的选择
0	CKSmn0位和CKSmn1位指定的运行时钟 (f_{MCK})
1	TImn引脚输入信号的有效边沿 • 单元0的情况: 通道0: 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_{TCLK}) 用于计数器、输出控制电路和中断控制电路。	

注 1.bit11是只读位，固定为“0”，忽视写操作。

注意

1.必须将bit13、5、4置“0”。

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

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-10 定时器模式寄存器mn (TMRmn) 的表格(2/4)

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

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

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

(TMRmn (n=2) 的bit11)

MASTERmn	通道n的独立通道运行/多通道联动运行（从属或者主控）的选择
0	用作独立通道运行功能或者多通道联动运行功能的从属通道。
1	用作多通道联动运行功能的主控通道。
只能将通道2设置为主控通道 (MASTERmn=1)。 通道0固定为“0”（因为通道0为最高位的通道，所以与此位的设置无关，用作主控通道）。对于用作独立通道运行功能的通道，将MASTERmn位置“0”。	

(TMRmn (n=1、3) 的bit11)

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

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

注 1.bit11是只读位，固定为“0”，忽视写操作。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-11 定时器模式寄存器mn (TMRmn) 的表格(3/4)

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

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

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

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

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)

图5-12 定时器模式寄存器mn (TMRmn) 的表格(4/4)

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

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

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

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

各模式的运行因MDmn0位而变 (参照下表)。

运行模式 (MDmn3~MDmn1位的设置 (参照上表))	MD mn0	开始计数和中断的设置
• 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 ^{注2} (1、0、0)	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效 ^{注3} 。此时不产生中断。
• 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。
- 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。
- 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5-4。

通过16位存储器操作指令读TSRmn寄存器。

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

图5-13 定时器状态寄存器mn (TSRmn) 的表格

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

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

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

表5-4各运行模式中的OVF位的变化和置位/清除条件

定时器运行模式	OVF位	置位/清除条件
<ul style="list-style-type: none"> • 捕捉模式 • 捕捉&单次计数模式 	清除	在捕捉时没有发生上溢的情况
	置位	在捕捉时发生上溢的情况
<ul style="list-style-type: none"> • 间隔定时器模式 • 事件计数器模式 • 单次计数模式 	清除	— (不能使用)
	置位	

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

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”。

图5-14 定时器通道允许状态寄存器m (TEm) 的表格

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

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

通过16位存储器操作指令设置TSm寄存器。

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

图5-15 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	TSHm3	0	TSHm1	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（参照表5-5 从计数允许状态到定时器计数寄存器mn (TCRmn) 开始计数为止的运行）。

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（参照表5-5 从计数允许状态到定时器计数寄存器mn (TCRmn) 开始计数为止的运行）。

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（参照表5-5 从计数允许状态到定时器计数寄存器mn (TCRmn) 开始计数为止的运行）。在通道1和通道3为8位定时器模式时，TSm1和TSm3为低8位定时器的运行允许（开始）触发。

注意

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

备注

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

5.3.7 定时器通道停止寄存器m (TTm)

TTm寄存器是设置各通道计数停止的触发寄存器。

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

通过16位存储器操作指令设置TTm寄存器。

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

图5-16 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	TTHm 3	0	TTHm 1	0	0	0	0	0	TTm 3	TTm 2	TTm 1	TTm 0

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置TIOS0寄存器。在产生复位信号后，TIOS0寄存器的值变为“00H”。

图5-17 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			设置禁止

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	EVENTC的事件输入信号

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

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

注意 1.选择的定时器输入的高低电平宽度需要大于等于 $1/f_{MCK}+10ns$ 。因此，在选择 f_{SUB} 作为 f_{CLK} 时（CKC寄存器的CSS=1），不能将TIS02位置“1”。

2.在通过定时器输入选择寄存器0（TIOS0）选择ELC的事件输入信号时，必须通过定时器时钟选择寄存器0（TPS0）选择 f_{CLK} 。

5.3.9 定时器输出允许寄存器m (TOEm)

TOEm寄存器是设置允许或者禁止各通道定时器输出的寄存器。

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

通过16位存储器操作指令设置TOEm寄存器。

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

图5-18 定时器输出允许寄存器m (TOEm) 的表格

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

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

注意 必须将bit15~4置“0”。

备注 m: 单元号 (m=0, 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。

通过16位存储器操作指令设置TOm寄存器。

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

图5-19 定时器输出寄存器m (TOm) 的表格

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

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

通过16位存储器操作指令设置TOLm寄存器。

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

图5-20 定时器输出电平寄存器m (TOLm) 的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	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 必须将bit15~4和bit0置“0”。

备注1.如果在定时器运行中改写此寄存器的值, 就在下一次定时器输出信号发生变化时反相定时器的输出逻辑, 而不是在改写后立即反相。

2.m: 单元号 (m=0, 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置TOMm寄存器。

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

图5-21 定时器输出模式寄存器m (TOMm) 的表格

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

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

注意 必须将bit15~4和bit0置“0”。

备注 m: 单元号 (m=0, 1) n: 通道号

n=0~3 (主控通道时: n=0、2)

p: 从属通道号

n=0: p=1、2、3

n=2: p=3

(有关主控通道和从属通道关系的详细内容，请参照“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寄存器设置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置NFEN1寄存器。在产生复位信号后，NFEN1寄存器的值变为“00H”。

注 详细内容请参照“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”。

图5-22 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 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寄存器设置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_{MCK} ）进行同步后检测2个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟（ f_{MCK} ）进行同步注。

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

注 详细内容请参照“5.5.1(2)选择Tl_{mn}引脚输入信号的有效边沿的情况（CCS_{mn}=1）”、“5.5.2计数器的开始时序”和“5.7定时器输入（Tl_{mn}）的控制”。

图5-23 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 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，定时器输入/输出引脚可任意配置到除P00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TO00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”。

将端口复用功能配置寄存器 P21CFG 置“0x0e”。

在将定时器输入引脚的复用端口用作定时器的输入时，必须将各端口对应的端口模式寄存器（PMxx）的位置“1”并且将端口模式控制寄存器（PMCxx）的位置“0”。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/TI00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”。

将端口复用功能配置寄存器P20CFG置“0x06”。

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）设置为主控通道。

(2)通道0以外的任何通道都能设置为从属通道。

(3)只能将主控通道的低位通道设置为从属通道。

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

(4)能对1个主控通道设置多个从属通道。

(5)当使用多个主控通道时，不能设置跨越主控通道的从属通道。

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

(6)和主控通道联动的从属通道需要设置相同的运行时钟。和主控通道联动的从属通道的CKSmn0位和CKSmn1位（定时器模式寄存器mn（TMRmn）的bit15和bit14）的值需要是相同的设置值。

(7)主控通道能将INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。

(8)从属通道能将主控通道的INTTMmn（中断）、开始软件触发和计数时钟用作源时钟，但是不能将自己的INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。

(9)主控通道不能将其他高位主控通道的INTTMmn（中断）、开始软件触发和计数时钟用作源时钟。

(10)为了同时启动要联动的通道，需要同时设置联动通道的通道开始触发位（TSmn）。

(11)只有联动的全部通道或者主控通道才能使用计数运行中的TSmn位的设置。不能只使用从属通道的TSmn位的设置。

(12)为了同时停止要联动的通道，需要同时设置联动通道的通道停止触发位（TTmn）。

(13)在联动运行时，因为主控通道和从属通道需要相同的运行时钟，所以不能选择CKm2/CKm3。

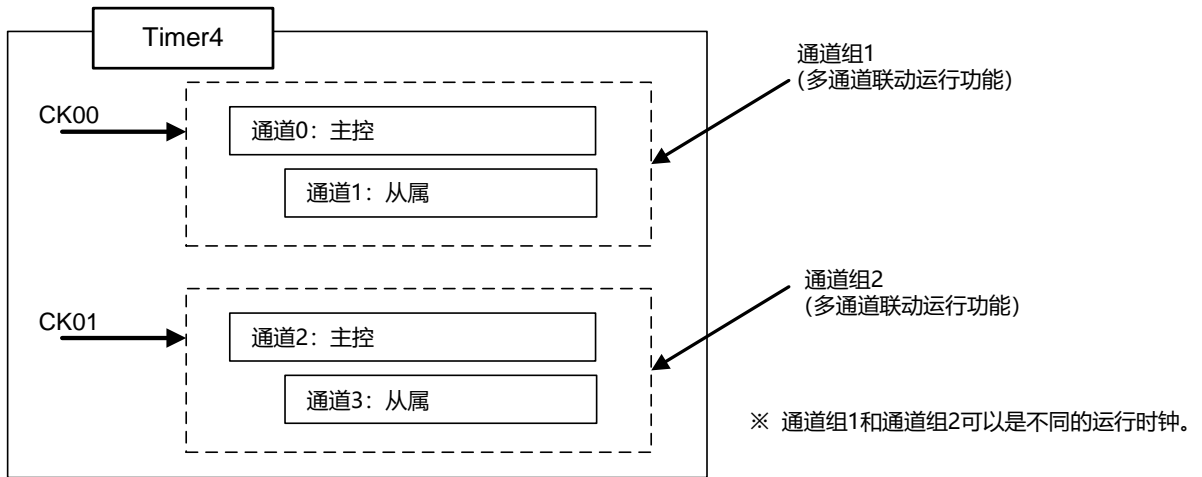
(14)定时器模式寄存器m0（TMRm0）没有主控位而固定为“0”。但是，因为通道0是最高位的通道，所以在联动运行时能将通道0用作主控通道。

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

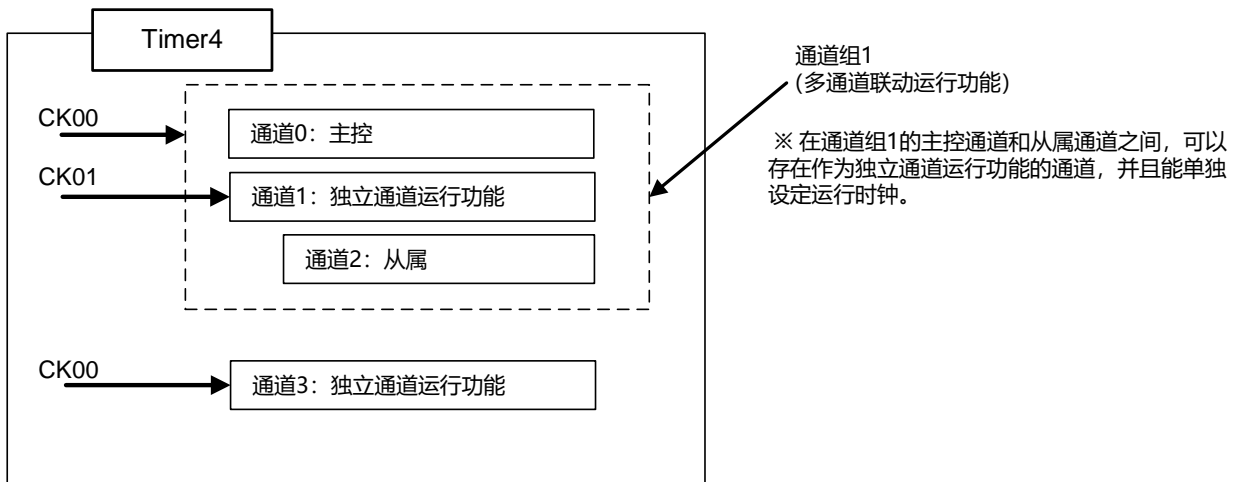
如果设置2个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

备注 m：单元号（m=0, 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INTTm1H/INTTm3H（中断）（和MDmn0位为“1”的运行相同）。
- (5)高8位定时器的运行时钟的选择取决于低位TMRmn寄存器的CKSmn1位和CKSmn0位的设置。
- (6)对于高8位定时器，通过操作TSHm1/TSHm3位来开始通道的运行，并且通过操作TTHm1/TTHm3位来停止通道的运行。能通过TEHm1/TEHm3位确认通道的状态。
- (7)低8位定时器的运行取决于TMRmn寄存器的设置，有以下3种支持低8位定时器运行的功能：
 - 间隔定时器功能
 - 外部事件计数器功能
 - 延迟计数功能
- (8)对于低8位定时器，通过操作TSm1/TSm3位来开始通道的运行，并且通过操作TTm1/TTm3位来停止通道的运行。能通过TEm1/TEm3位确认通道的状态。
- (9)在16位定时器运行时，TSHm1/TSHm3/TTHm1/TTHm3位的操作无效。通过操作TSm1/TSm3位和TTm1/TTm3位使通道1和通道3运行。TEHm3位和TEHm1位不变。
- (10)8位定时器功能不能使用联动运行功能（单触发脉冲、PWM和多重PWM）。

备注 m：单元号（m=0） n：通道号（n=1、3）

5.5 计数器的运行

5.5.1 计数时钟 (f_{TCLK})

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

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

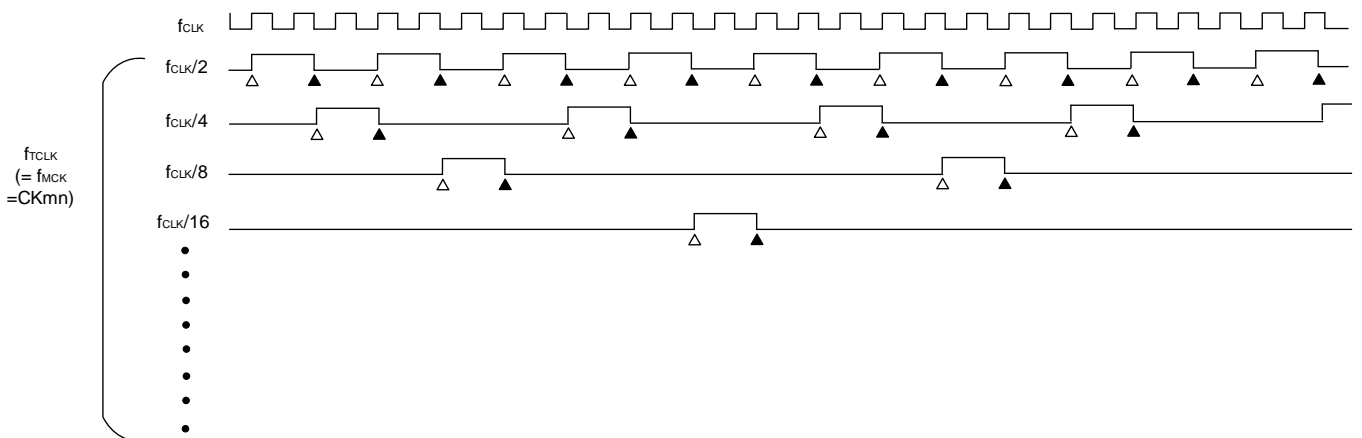
通用定时器单元被设计为与 f_{CLK} 同步运行, 因此计数时钟 (f_{TCLK}) 的时序如下。

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

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

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

图5-24 f_{CLK} 和计数时钟 (f_{TCLK}) 的时序 (CCSmn=0的情况)

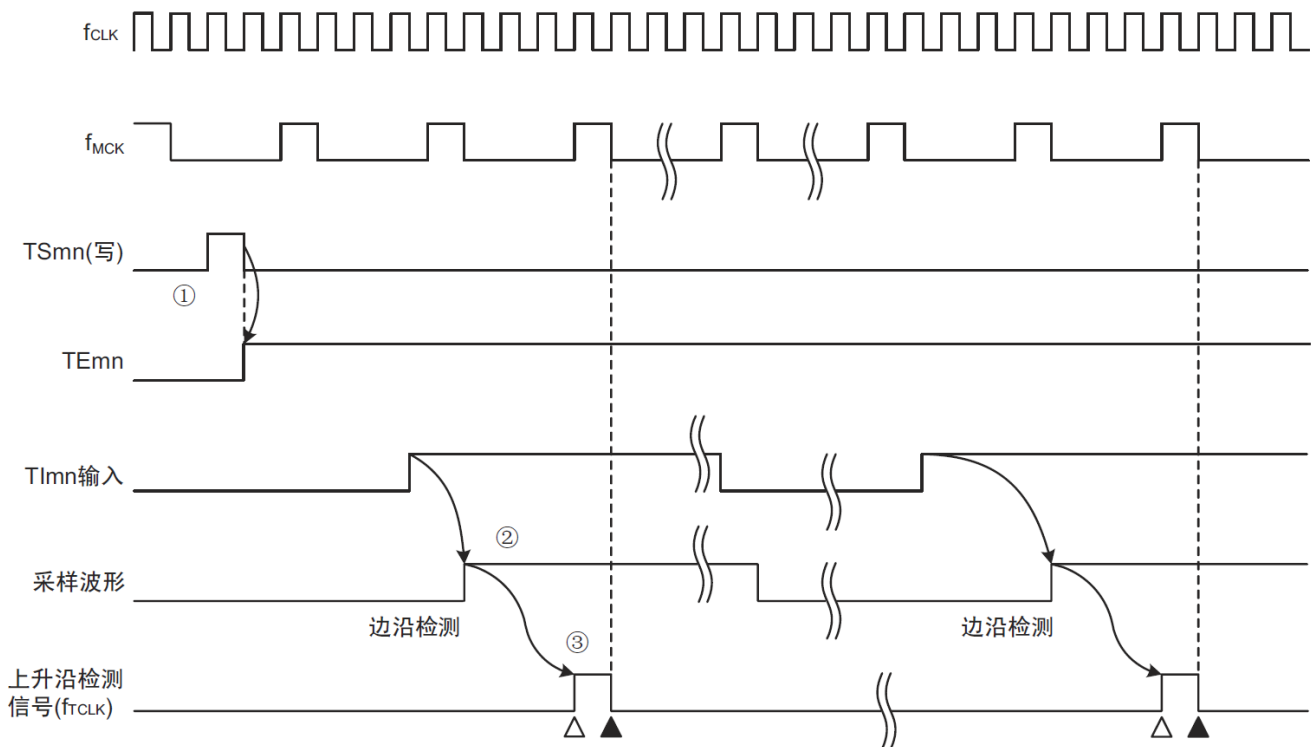


- 备注 1. Δ : 计数时钟的上升沿
 ▲: 同步、计数器的递增/递减
 2. f_{CLK} : CPU/外围硬件的时钟

(2) 选择TImn引脚输入信号的有效边沿的情况 (CCSmn=1)

计数时钟 (f_{TCLK}) 是检测TImn引脚输入信号的有效边沿并且与下一个f_{MCK}上升沿同步的信号。实际上, 这是比TImn引脚的输入信号延迟了1~2个f_{MCK}时钟的信号 (在使用噪声滤波器时, 延迟3~4个f_{MCK}时钟)。为了取得与f_{TCLK}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_{TCLK}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5-25 计数时钟 (f_{TCLK}) 的时序 (CCSmn=1, 未使用噪声滤波器的情况)



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

备注1.△: 计数时钟的上升沿

▲: 同步、计数器的递增/递减

2.f_{CLK}: CPU外围硬件时钟

f_{MCK}: 通道n的运行时钟

3.输入脉冲间隔的测量、输入信号高低电平的测量、延迟计数器和单触发脉冲输出功能的TImn输入也是同样的波形。

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5-5所示。

表5-5 从计数允许状态到定时器计数寄存器mn (TCRmn) 开始计数为止的运行

定时器的运行模式	将TSMn位置“1”后的运行
• 间隔定时器模式	从检测到开始触发 (TSMn=1) 到产生计数时钟为止, 不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器, 并且通过后续的计数时钟进行递减计数 (参照 “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)”)。

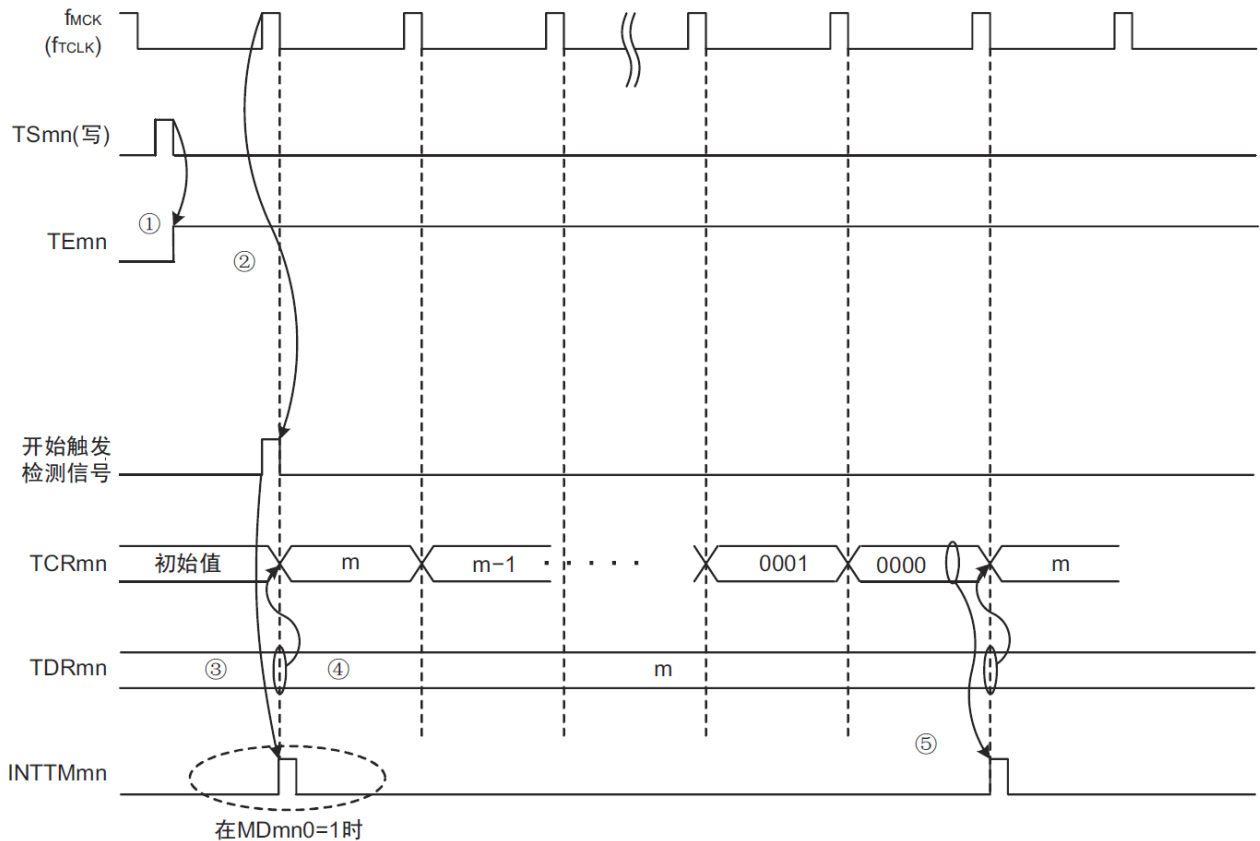
5.5.3 计数器的运行

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5-26 运行时序 (间隔定时器模式)



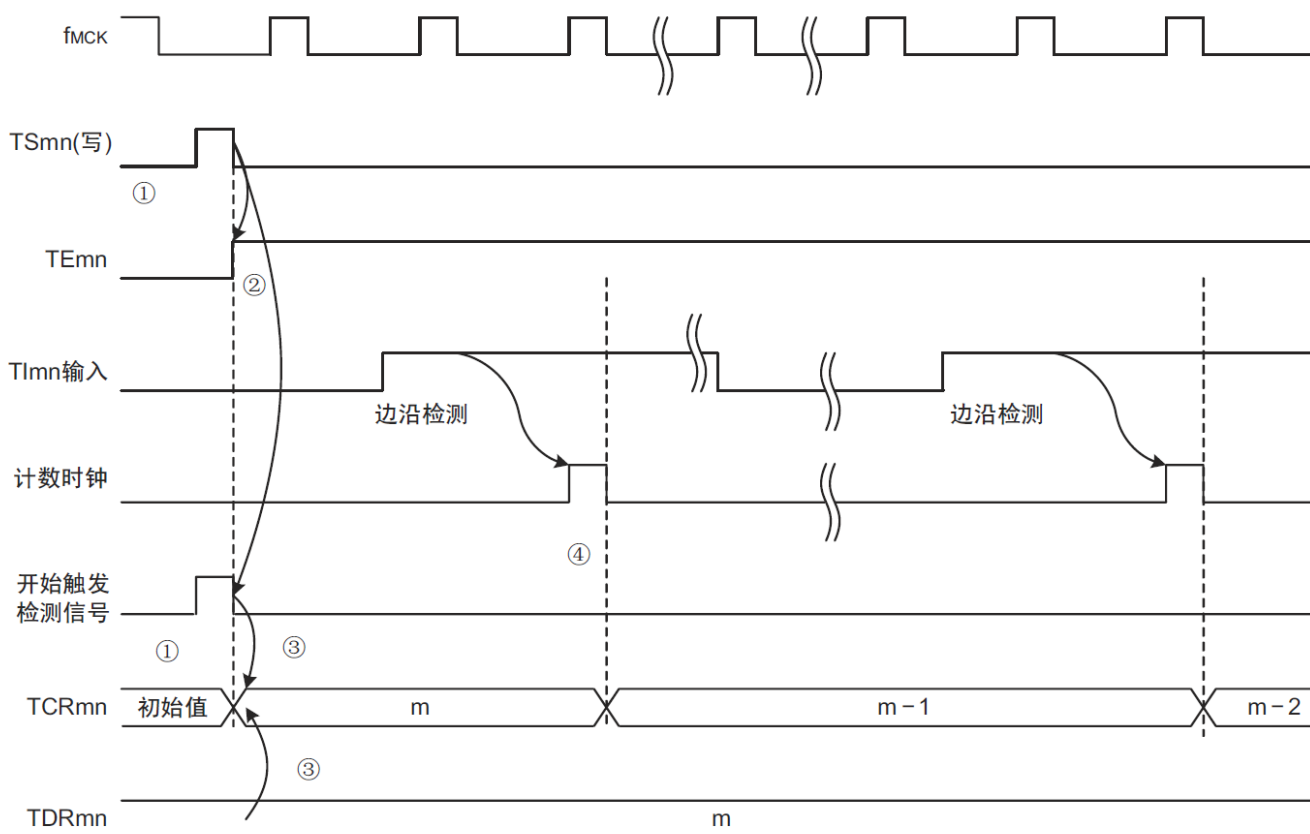
注意 因为第1个计数时钟周期的运行在写TSmn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

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

(2) 事件计数器模式的运行

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

图5-27 运行时序 (事件计数器模式)

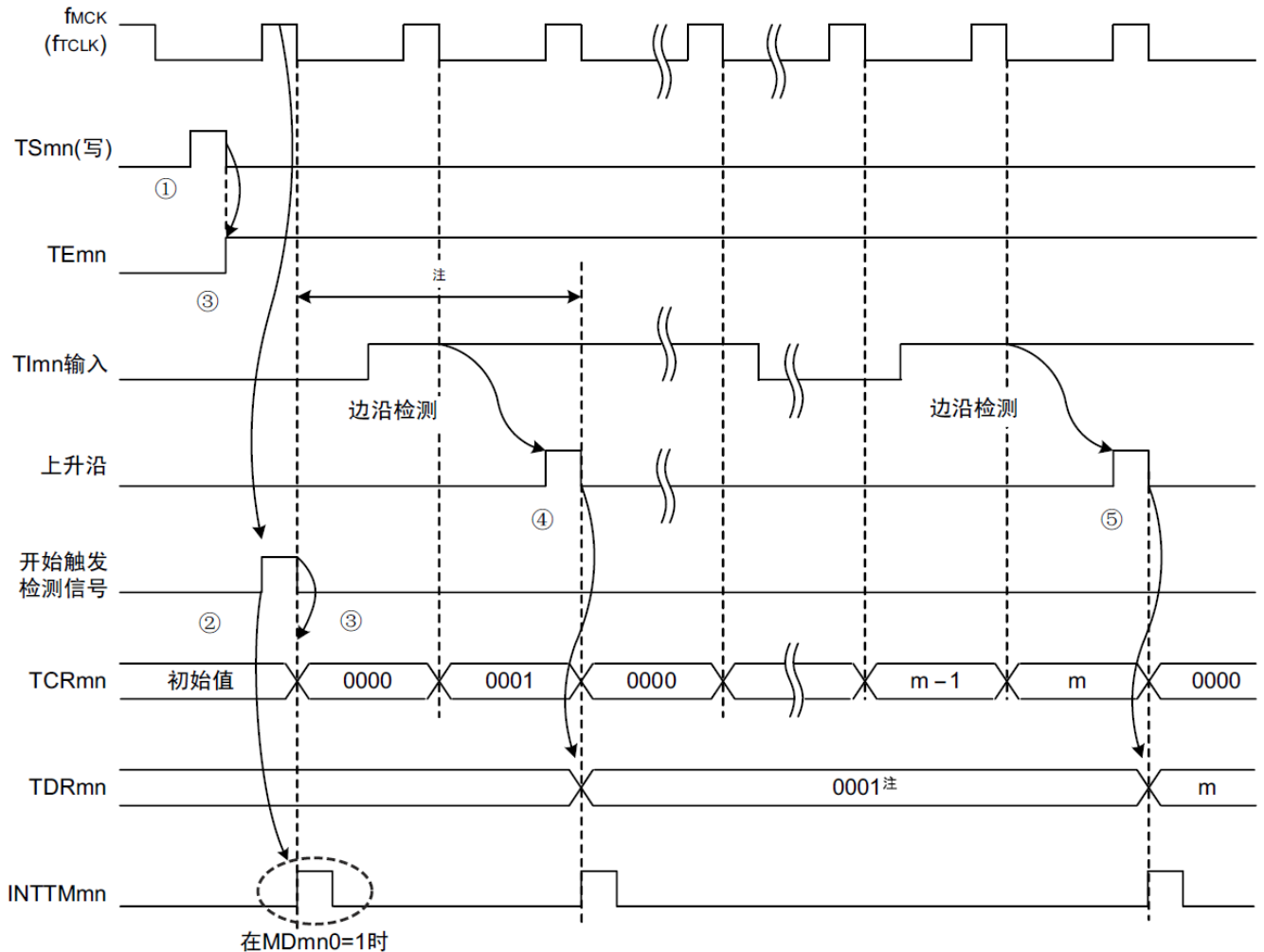


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从TI_{mn}输入开始再延迟2个f_{MCK}周期 (合计3~4个周期)。1个周期的误差是因为TI_{mn}输入与计数时钟 (f_{MCK}) 不同步。

(3) 捕捉模式的运行（输入脉冲的间隔测量）

- ①通过给TSMn位写“1”，进入运行允许状态（TEmn=1）。
- ②定时器计数寄存器mn（TCRmn）保持初始值，直到产生计数时钟为止。
- ③通过允许运行后的第1个计数时钟（fMCK）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5-28 运行时序（捕捉模式：输入脉冲的间隔测量）



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

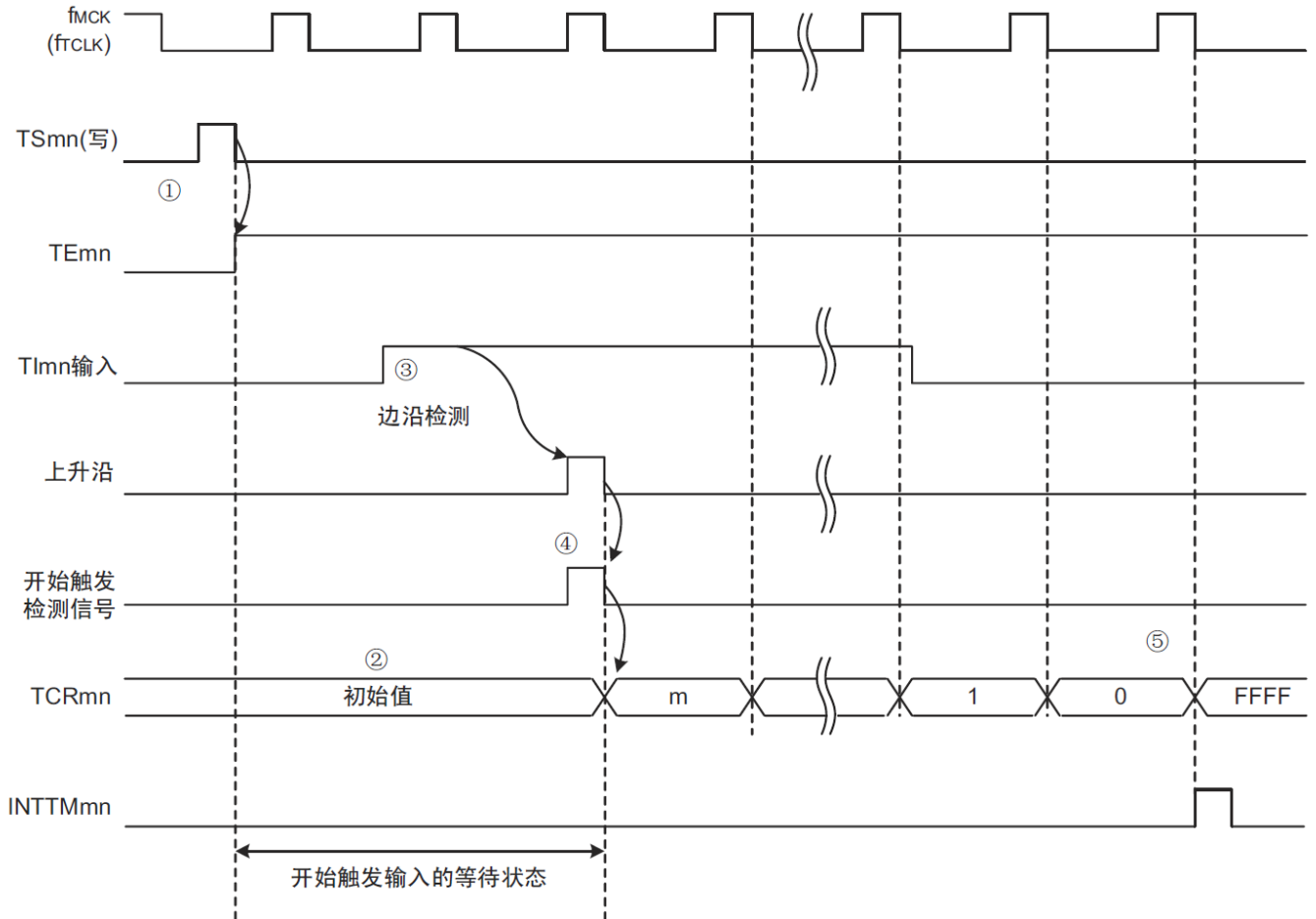
注意 因为第1个计数时钟周期的运行在写TSMn位后并且在产生计数时钟前延迟计数的开始，所以产生最大为1个时钟周期的误差。另外，如果需要开始计数时序的信息，就将MDmn0位置“1”，以便能在开始计数时产生中断。

备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TImn输入开始再延迟2个fMCK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（fMCK）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5-29 运行时序（单次计数模式）

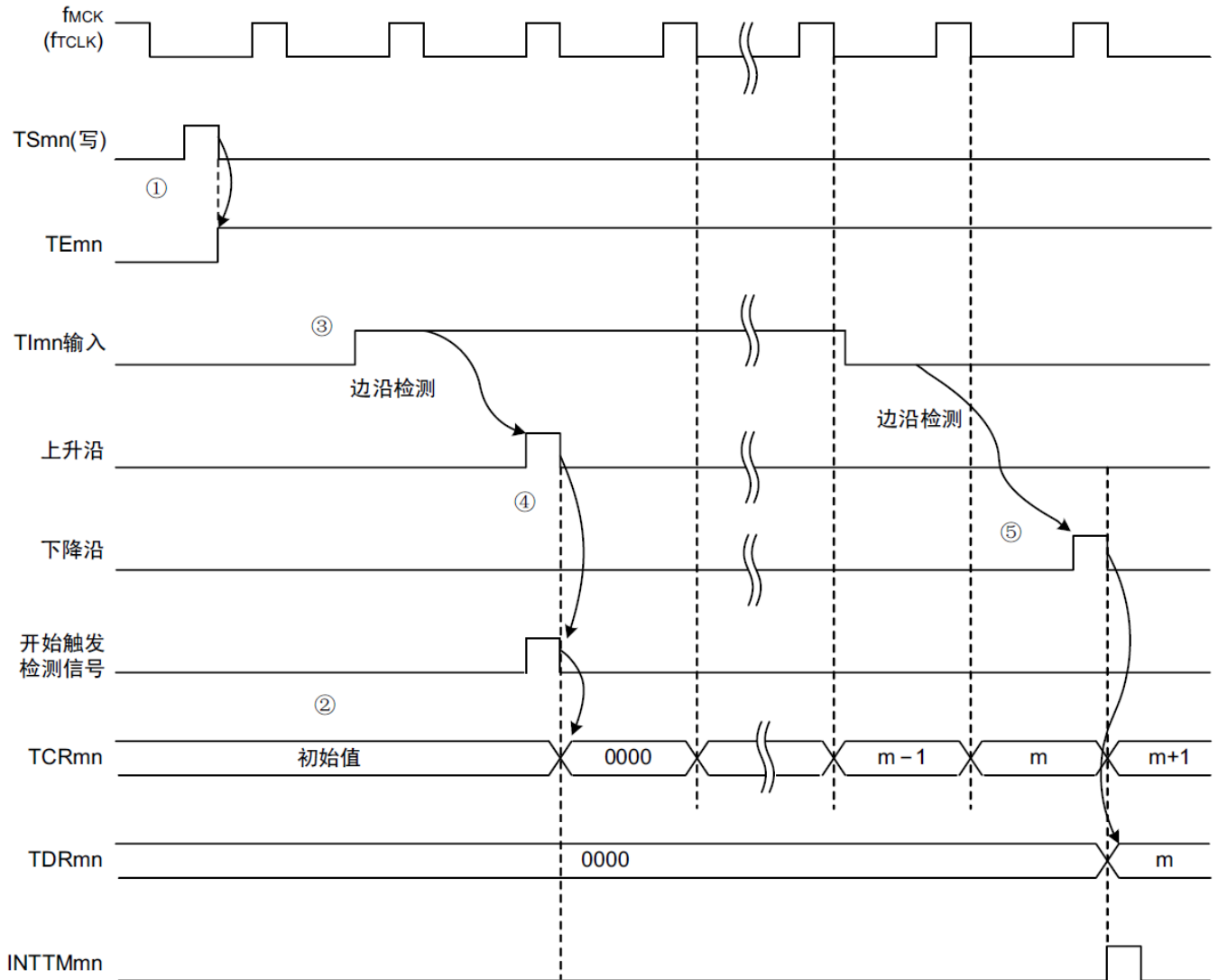


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TIMn输入开始再延迟2个 f_{MCK} 周期（合计3~4个周期）。1个周期的误差是因为TIMn输入与计数时钟（ f_{MCK} ）不同步。

(5) 捕捉& 单次计数模式的运行（高电平宽度的测量）

- ①通过给定时器通道开始寄存器m（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5-30 运行时序（捕捉&单次计数模式：高电平宽度的测量）

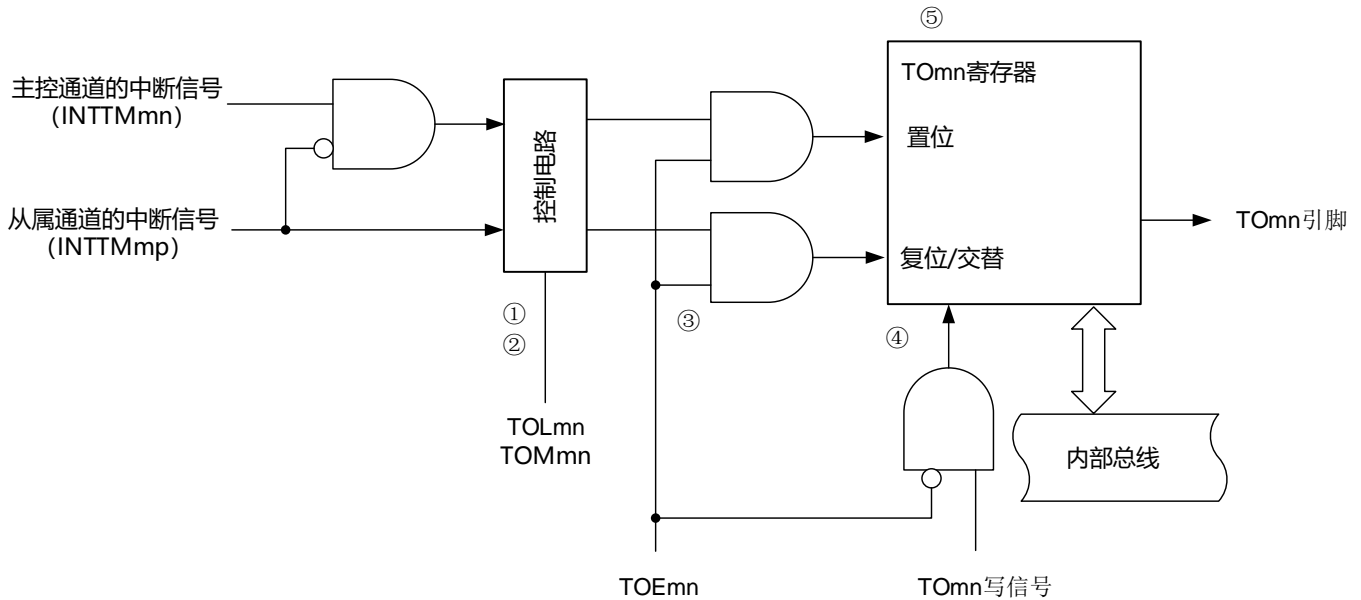


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从TIMn输入开始再延迟2个f_{MCK}周期（合计3~4个周期）。1个周期的误差是因为TIMn输入和计数时钟（f_{MCK}）不同步。

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5-31 输出电路的结构



以下说明TOmn引脚的输出电路。

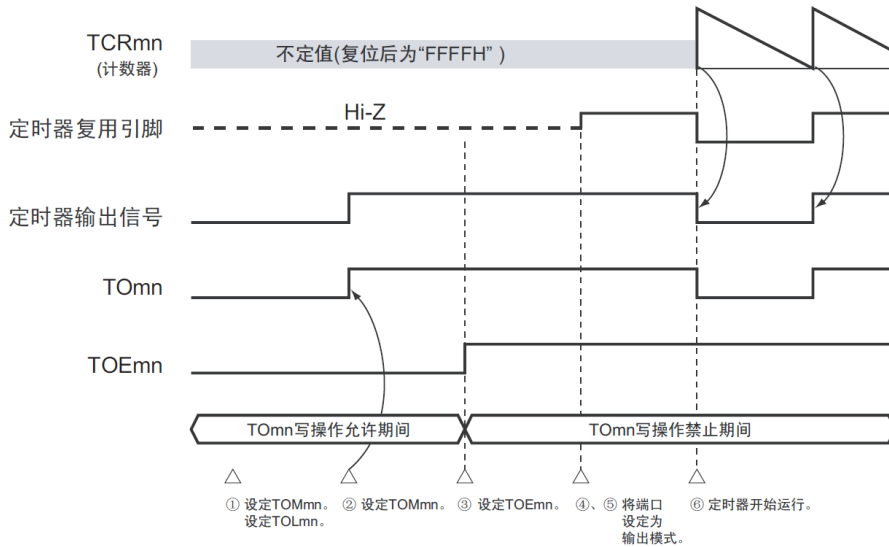
- ①当TOMmn位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器m（TOLm）的设置值，只将INTTMmp（从属通道定时器中断）传给定时器输出寄存器m（TOM）。
- ②当TOMmn位为“1”（从属通道输出模式）时，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOM寄存器。此时，TOLm寄存器有效并且进行以下信号的控制：
 TOLmn=0时：正相运行（INTTMmn 置位、INTTMmp 复位）
 TOLmn=1时：反相运行（INTTMmn 复位、INTTMmp 置位）
 当同时产生INTTMmn和INTTMmp时（PWM输出的0%输出），优先INTTMmp（复位信号）而屏蔽INTTMmn（置位信号）。
- ③在允许定时器输出（TOEmn=1）的状态下，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOM寄存器。TOM寄存器的写操作（TOmn写信号）无效。当TOEmn位为“1”时，除了中断信号以外，不改变TOmn引脚的输出。要对TOmn引脚的输出电平进行初始化时，需要在设置为禁止定时器输出（TOEmn=0）后给TOM寄存器写值。
- ④在禁止定时器输出（TOEmn=0）的状态下，对象通道的TOMmn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。
- ⑥ 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 TOmn引脚的输出设置

从TOmn输出引脚的初始设置到定时器开始运行的步骤和状态变化如下所示。

图5-32 从设置定时器的输出到开始运行的状态变化



①设置定时器输出的运行模式。

- TOMmn位 (0: 主控通道输出模式、1: 从属通道输出模式)
- TOLmn位 (0: 正逻辑输出、1: 负逻辑输出)

②通过设置定时器输出寄存器m (TOM)，将定时器输出信号设置为初始状态。

③给TOEmn位写“1”，允许定时器输出（禁止写TOM寄存器）。

④通过端口模式控制寄存器 (PMCxx) 将端口设置为数字输入/输出。

⑤将端口的输入/输出设置为输出。

⑥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寄存器的设置值变更

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) 的设置值的变更不会影响定时器的运行，能在定时器运行中更改设置值。但是，为了在各定时器的运行中从TOMn引脚输出期待的波形，必须设置为“5.8通用定时器单元的独立通道运行功能”和“5.9通用定时器单元的多通道联动运行功能”所示的各运行的寄存器设置内容例子的值。

如果在产生各通道的定时器中断 (INTTMmn) 信号前后更改除了TOM寄存器以外的TOEm寄存器和TOLm寄存器的设置值，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。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

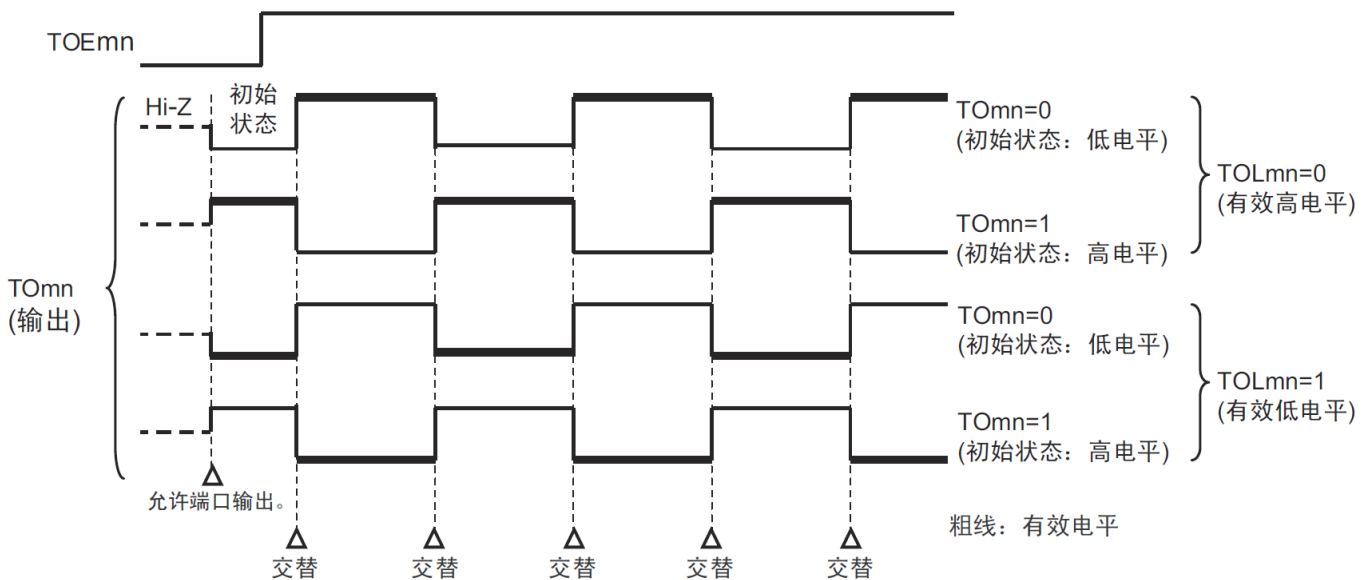
(2) 有关TOMn引脚的初始电平和定时器开始运行后的输出电平

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置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。

(a) 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) 的设置无效。如果在设置初始电平后开始定时器的运行，就通过产生交替信号反相TOMn引脚的输出电平。

图5-33 交替输出时 (TOMmn=0) 的TOMn引脚输出状态



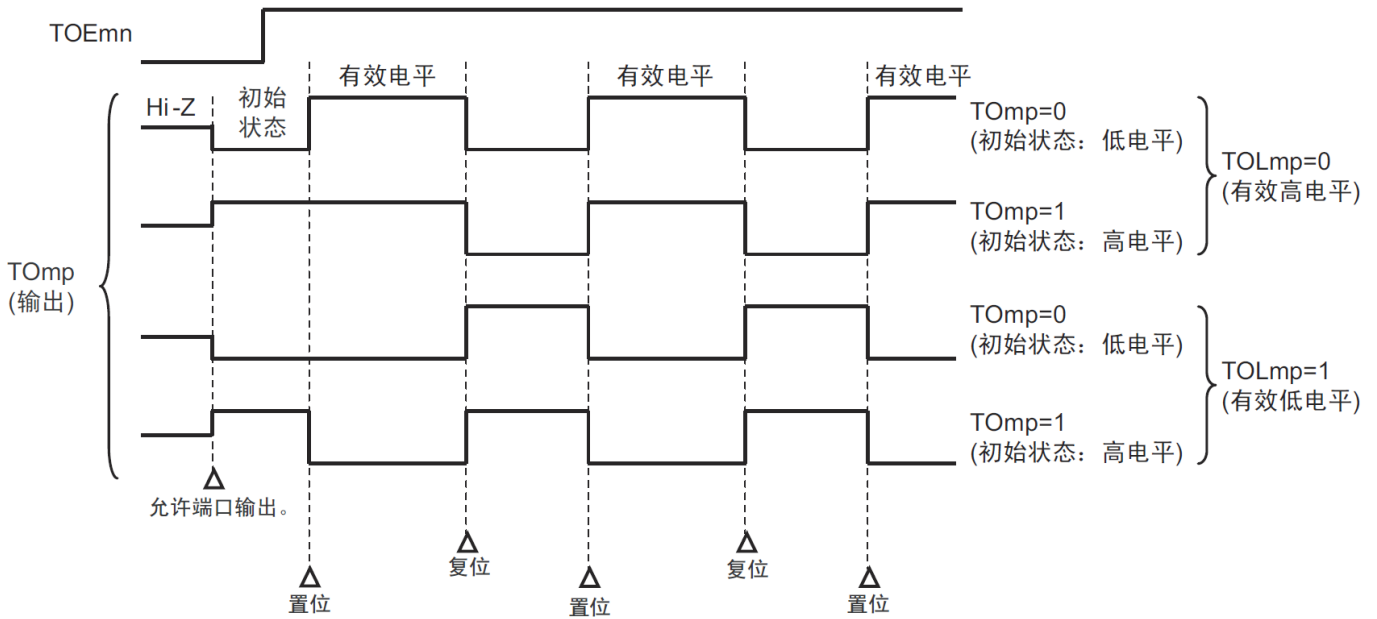
备注1.交替: 反相TOMn引脚的输出状态。

2.m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

(b) 在从属通道输出模式 (TOMmn=1) 中开始运行的情况 (PWM输出)

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

图5-34 PWM输出时 (TOMmn=1) 的TOmn引脚输出状态



备注1.置位: TOmp引脚的输出信号从无效电平变为有效电平。

复位: TOmp引脚的输出信号从有效电平变为无效电平。

2.m: 单元号 (m=0, 1) n: 通道号 (p=1~3)

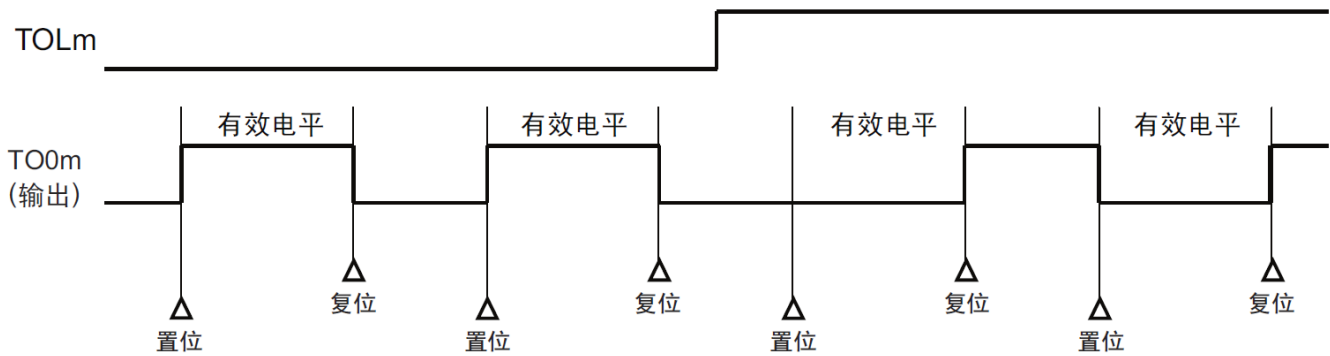
(3) 有关从属通道输出模式 (TOMmn=1) 的TOMn引脚变化

(a) 在定时器运行中更改定时器输出电平寄存器m (TOLm) 的设置的情况

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

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

图5-35 在定时器运行中更改TOLm寄存器的内容时的运行



备注1.置位: TOMn引脚的输出信号从无效电平变为有效电平。

复位: TOMn引脚的输出信号从有效电平变为无效电平。

2.m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

(b) 置位/复位时序

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

当置位条件和复位条件同时产生时, 优先复位条件。

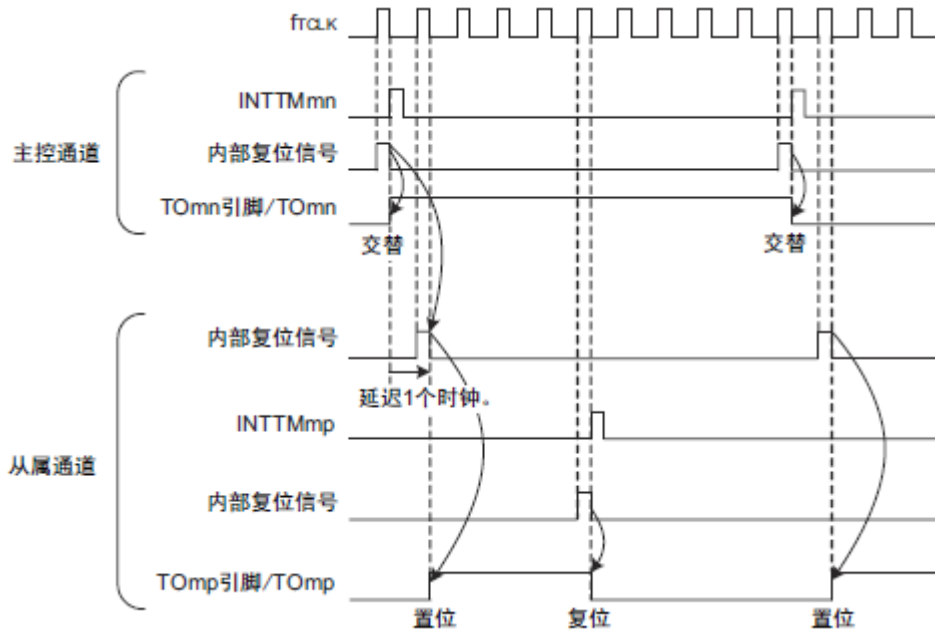
按照以下方法设置主控/从属通道时的置位/复位运行状态如图5-35所示。

主控通道: TOEmn=1、TOMmn=0、TOLmn=0

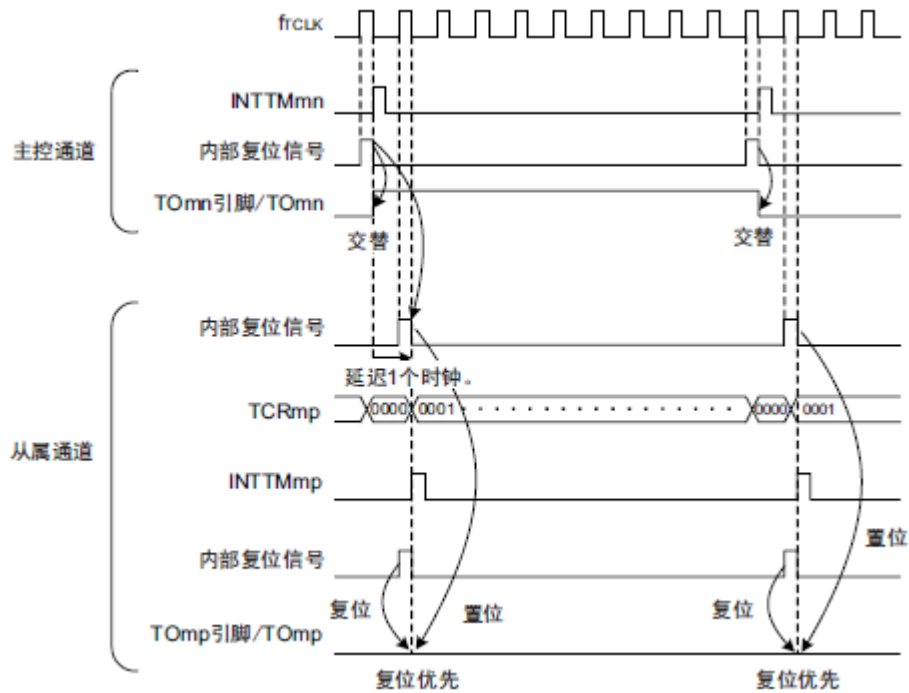
从属通道: TOEmp=1、TOMmp=1、TOLmp=0

图5-36 置位/复位时序运行状态

(1) 基本运行时序



(2) 0%占空比的运行时序



备注1.内部复位信号：T0mn引脚的复位/交替信号

内部置位信号：T0mn引脚的置位信号

2.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

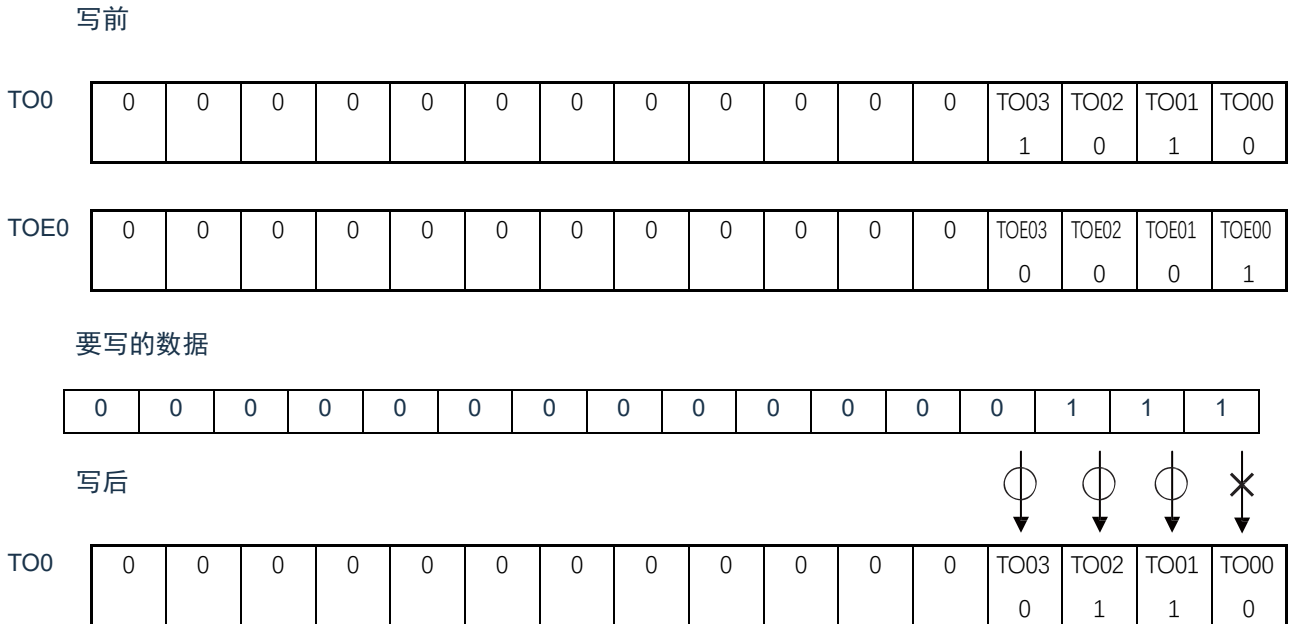
n=0：p=1、2、3

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) 有全部通道的设置位 (TOmn), 因此能一次性地操作全部通道的TOmn位。

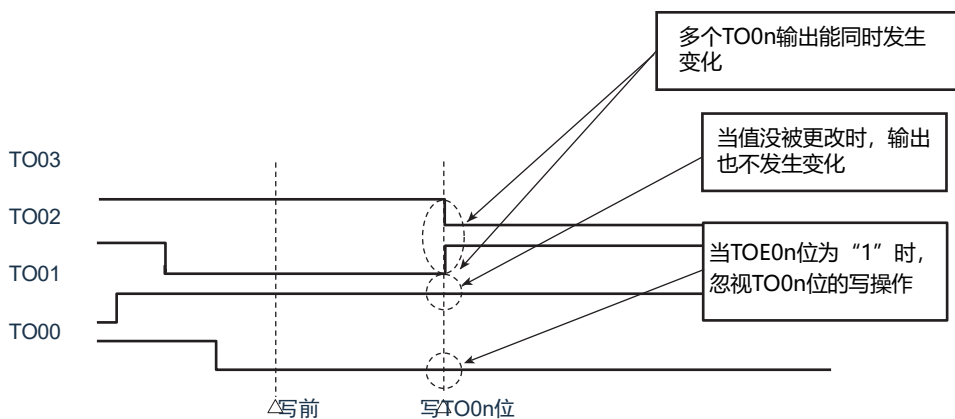
图5-37 TO0n位的一次性操作例子



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

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

图5-38 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)

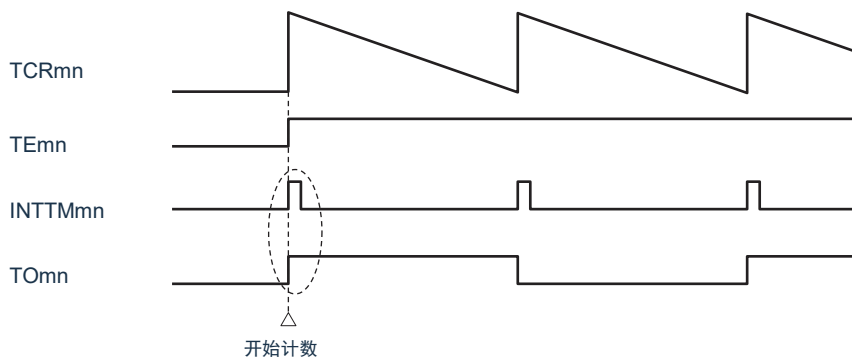
5.6.5 有关开始计数时的定时器中断和TOMn引脚输出

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位是设置是否在开始计数时产生定时器中断的位。

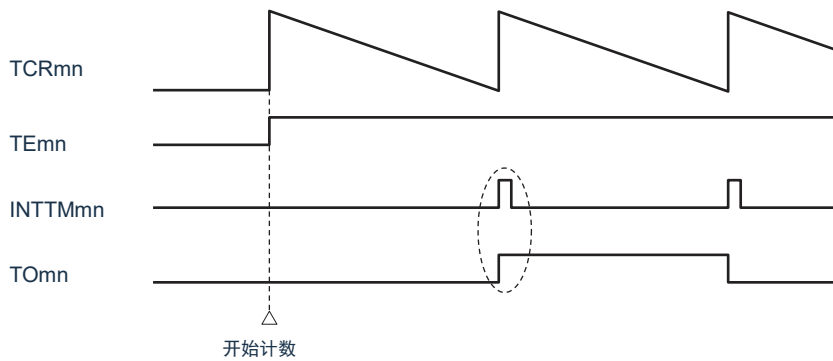
当MDmn0位为“1”时，能通过产生定时器中断（INTTMmn）得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和TOMn输出。设置为间隔定时器模式（TOEmn=1、TOMmn=0）时的运行例子如下所示。

图5-39 开始计数时的定时器中断和TOMn输出的运行例子

(a) MDmn0位为“1”的情况



(b) MDmn0位为“0”的情况



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

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

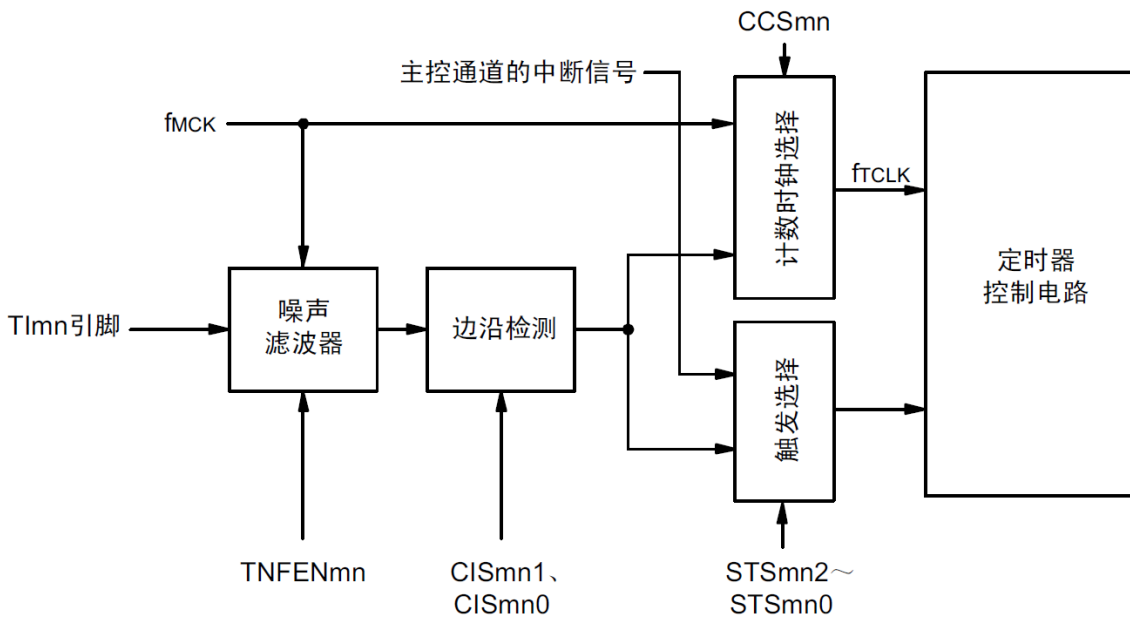
备注 m：单元号（m=0, 1） n：通道号（n=0~3）

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。

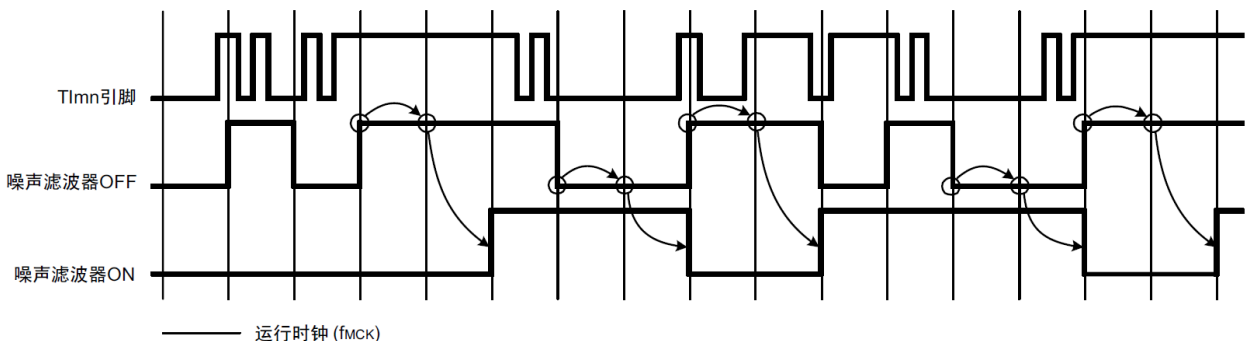
图5-40 输入电路的结构



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5-41 TImn输入引脚在噪声滤波器ON或者OFF情况下的采样波形



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

5.7.3 操作通道输入时的注意事项

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

(1) 噪声滤波器为OFF的情况

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

(2) 噪声滤波器为ON的情况

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

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：

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

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

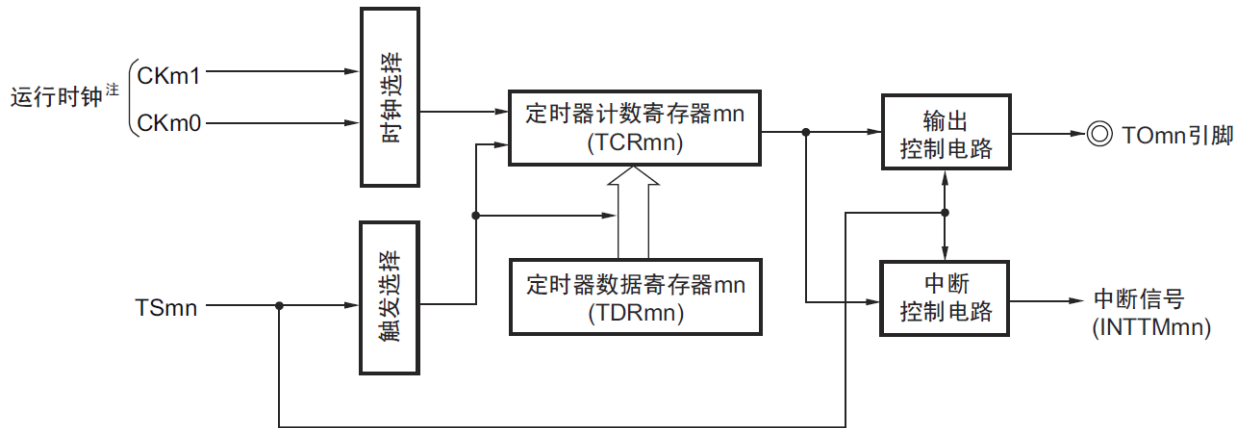
在间隔定时器模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

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。

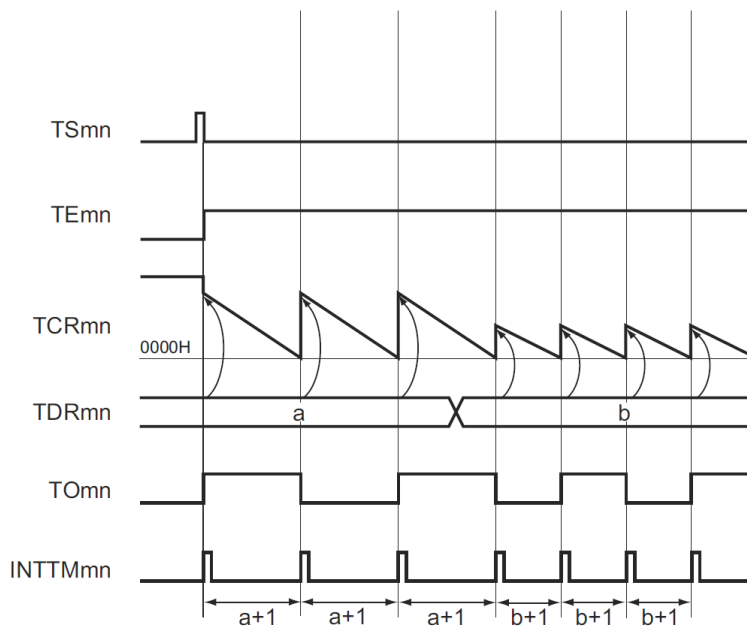
能随时改写TDRmn寄存器，改写的TDRmn寄存器的值从下一个周期开始有效。

图5-42 作为间隔定时器/方波输出运行的基本时序例子 (MDmn=1)



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

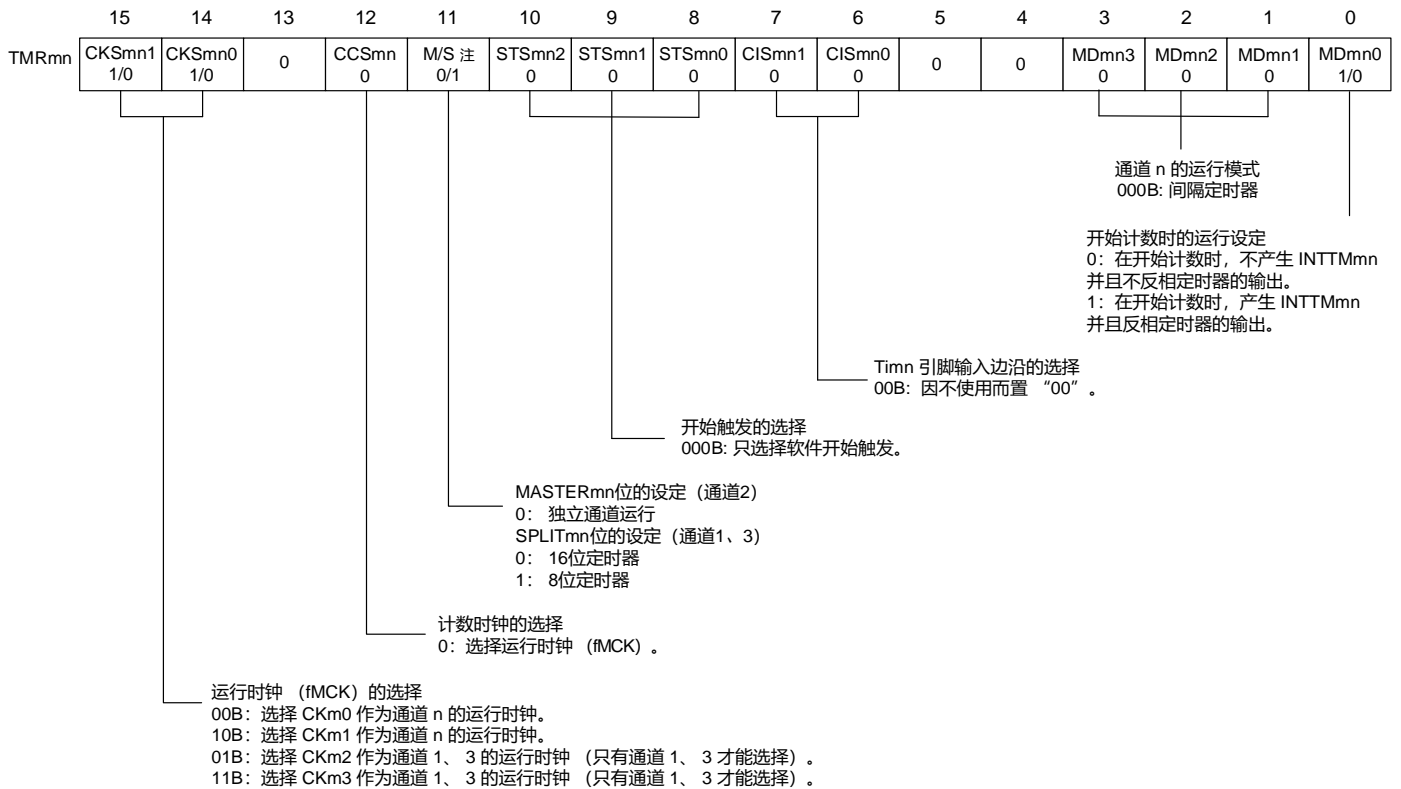
图5-43 作为间隔定时器/方波输出运行的基本时序例子 (MDmn=1)



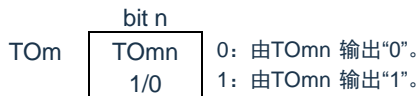
- 备注1. m: 单元号 (m=0, 1) n: 通道号 (n=0 ~3)
 2. TSmn : 定时器通道开始寄存器m (TSm) 的bit n
 TEMn : 定时器通道允许状态寄存器m (TEm) 的bit n
 TCRmn : 定时器计数寄存器 mn (TCRmn)
 TDRmn : 定时器数据寄存器 mn (TDRmn)
 TOMn : TOMn 引脚输出信号

图5-44 间隔定时器/方波输出时的寄存器设置内容例子

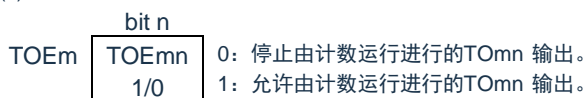
(a) 定时器模式寄存器mn (TMRmn)



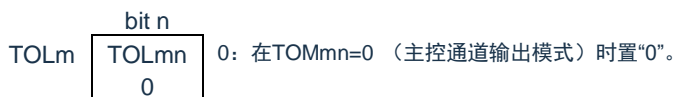
(b) 定时器输出寄存器m (TOM)



(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注 TMRm2 : MASTERmn 位
TMRm1、 TMRm3 : SPLITmn位
TMRm0: 固定为“0”。

备注 m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~3)

图5-45 间隔定时器/方波输出功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的TM4mEN 位置 "1"。	定时器单元m的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器mn (TMRmn) (确定通道的运行模式)。 给定定时器数据寄存器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。 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TOEmn 位置 "1", 允许TOmn 输出。	因为通道处于运行停止状态, 所以TOmn 不变。
	将端口寄存器和端口模式寄存器置 "0"。	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。
	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。
停止 运行	将TTmn (TTHm1、TTHm3) 位置 "1"。 因为TTmn (TTHm1、TTHm3) 位是触发位, 所以自动返回到 "0"。	TEmn (TEHm1、TEHmn) 位变为 "0" 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 TOmn 输出不被初始化而保持状态。
	将TOEmn 位置 "0" 并且给TOmn 位设定值。	TOmn 引脚输出TOmn 位设定的电平。
TAU 停止	要保持TOmn 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将TOmn 位置 "0"。 不需要保持TOmn 引脚输出电平的情况: 不需要设定。	通过端口功能保持TOmn 引脚的输出电平。
	将PER0 寄存器的TM4mEN 位置 "0"。	定时器单元m 的输入时钟处于停止提供状态。 对全部电路和各通道的SFR 进行初始化。 (TOmn位变为 "0" 并且TOmn引脚变为端口功能)

重新
开始
运行

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。

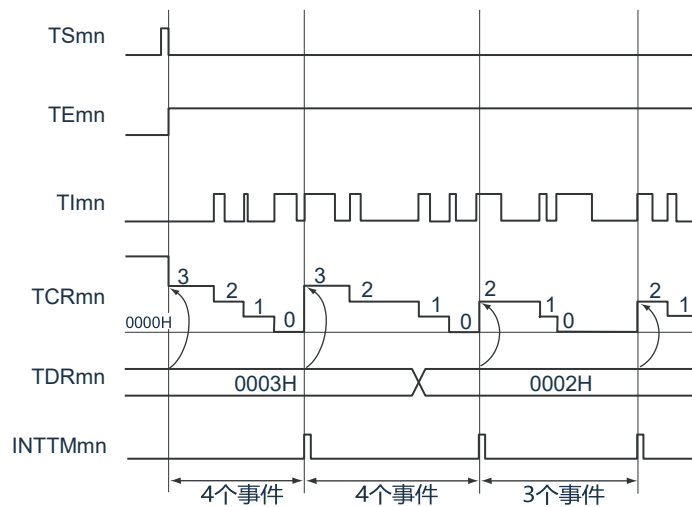
TCRmn寄存器在检测到TIMn引脚输入的有效边沿的同时进行递减计数。如果TCRmn变为“0000H”，就再次装入TDRmn寄存器的值并且输出INTTMmn。

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。

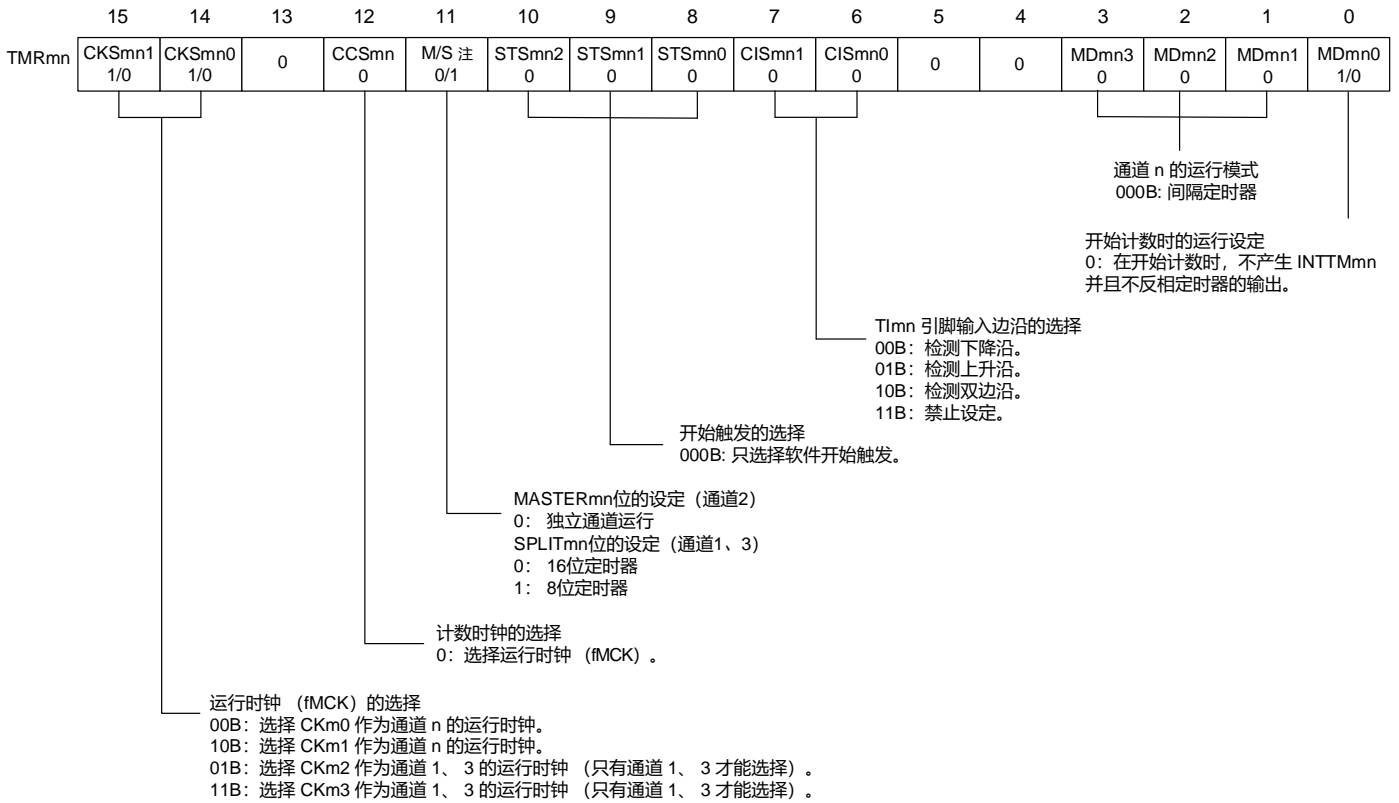
图5-46 作为外部事件计数器运行的基本时序例子



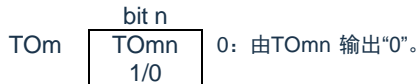
- 备注 1. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)
 2. TSMn : 定时器通道开始寄存器m (TSM) 的bit n
 TE mn : 定时器通道允许状态寄存器m (TEm) 的bit n
 TIMn : TIMn引脚输入信号
 TCRmn : 定时器计数寄存器mn (TCRmn)
 TDRmn : 定时器数据寄存器mn (TDRmn)

图5-47 外部事件计数器模式时的寄存器设置内容例子

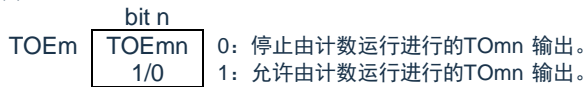
(a) 定时器模式寄存器mn (TMRmn)



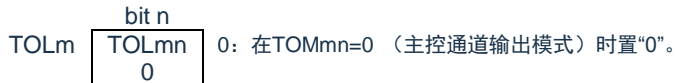
(b) 定时器输出寄存器m (TOM)



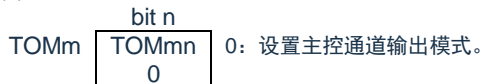
(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注 TMRm2 : MASTERmn 位

TMRm1、 TMRm3 : SPLITmn位

TMRm0 : 固定为“0”。

备注 m: 单元号 (m= 0, 1) n: 通道号 (n=0 ~3)

图5-48 外部事件计数器功能时的操作步骤

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

重新
开始
运行

5.8.3 作为分频器的运行

以单元0的通道0为例，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。

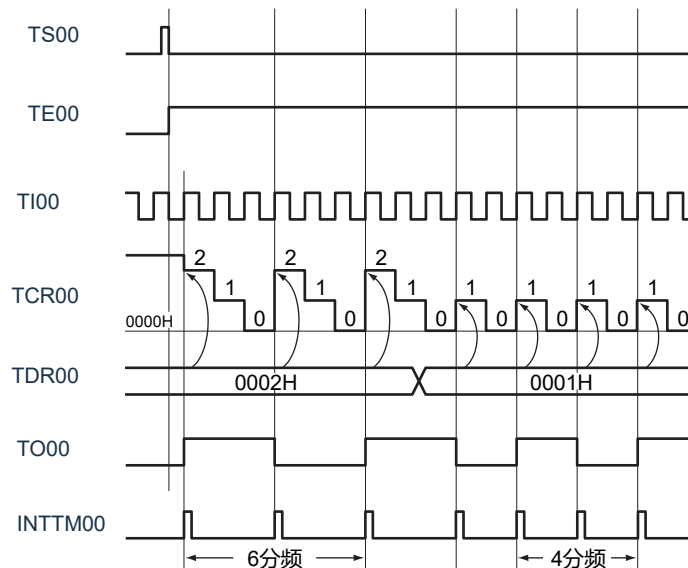
如果选择 TI00 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

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。

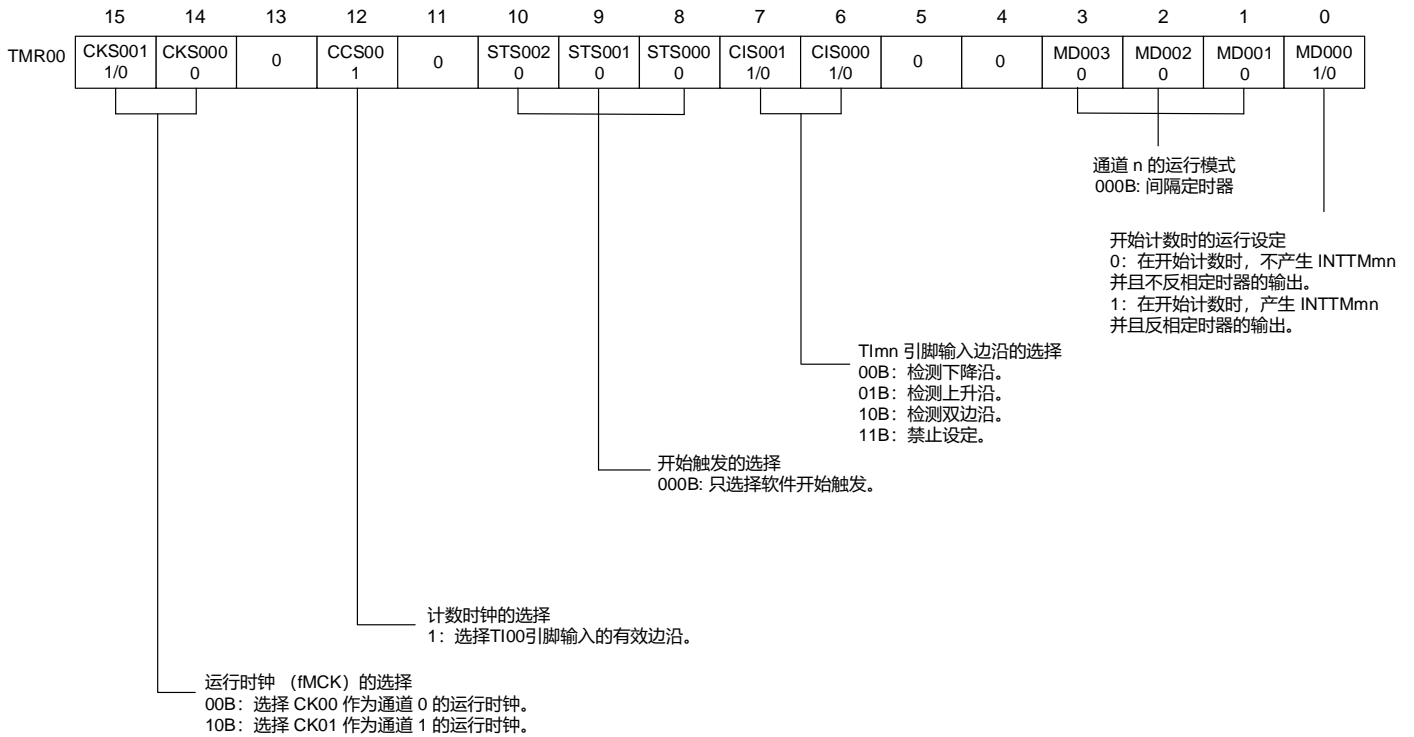
图5-49 作为分频器运行的基本时序例子 (MD000=1)



- 备注
- TS00 : 定时器通道开始寄存器0 (TS0) 的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5-50 作为分频器运行时的寄存器设置内容例子

(a) 定时器模式寄存器0 (TMR00)



(b) 定时器输出寄存器0 (TO0)

bit 0
TO0

TO00
1/0

 0: 由TO00 输出“0”。
1: 由TO00 输出“1”。

(c) 定时器输出允许寄存器0 (TOE0)

bit 0
TOE0

TOE00
1/0

 0: 停止由计数运行进行的TO00 输出。
1: 允许由计数运行进行的TO00 输出。

(d) 定时器输出电平寄存器0 (TOL0)

bit 0
TOL0

TOL00
0

 0: 在主导通道输出模式 (TOM00=0) 中置“0”。

(e) 定时器输出模式寄存器0 (TOM0)

bit n
TOM0

TOM00
0

 0: 设置主导通道输出模式。

图5-51 分频器功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的TM4mEN 位置 "1"。	定时器单元0 的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器0 (TPS0)。 确定CK00 ~ CK03 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00 (TMR00) (确定通道的运行模式, 选择检测边沿)。 给定时器数据寄存器00 (TDR00) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	将定时器输出模式寄存器0 (TOM0) 的TOM00 位置 "0" (主控通道输出模式)。 将TOL00 位置 "0"。 设定TO00 位并且确定TO00 输出的初始电平。	TO00 引脚处于Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 "0" 时, 输出TO00 初始设定的电平。
	将TOE00 位置 "1", 允许TO00 输出。	因为通道处于运行停止状态, 所以TO00 不变。
	将端口寄存器和端口模式寄存器置 "0"。	TO00 引脚输出TO00 设定的电平。
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"。	TE00位变为 "1" 并且开始计数。 将TDR00 寄存器的值装入定时器计数寄存器00 (TCR00)。 当TMR00寄存器的MD000位为 "1" 时, 产生INTTM00 并且TO00 进行交替输出。
	能任意更改TDR00 寄存器的设定值。 能随时读TCR00 寄存器。 不使用TSR00 寄存器。 能更改TO0 寄存器和TOE0 寄存器的设定值。 禁止更改TMR00 寄存器、TOM00 位和TOL00 位的设定值。	计数器 (TCR00) 进行递减计数。如果计数到 "0000H", 就再次将TDR00寄存器的值装入TCR00 寄存器并且继续计数。当检测到TCR00 为 "0000H" 时, 产生INTTM00 并且TO00 进行交替输出。 此后, 重复此运行。
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"。	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。
	将TOE00 位置 "0" 并且给TO00 位设定值。	TO00 引脚输出TO00 位设定的电平。
Timer4 停止	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。	通过端口功能保持TO00 引脚的输出电平。
	将PER0 寄存器的TM4mEN 位置 "0"。	定时器单元0 的输入时钟处于停止提供状态。 对全部电路和各通道的SFR 进行初始化。 (TO00位变为 "0" 并且TO00引脚变为端口功能)

重新开始运行

5.8.4 作为输入脉冲间隔测量的运行

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

脉冲间隔能用以下计算式进行计算：

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

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

在捕捉模式中，定时器计数寄存器mn (TCR_{mn}) 用作递增计数器。

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

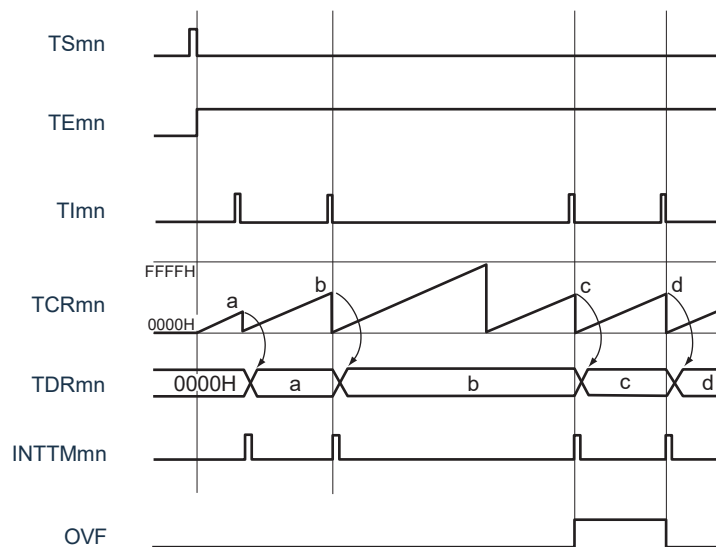
如果检测到Tl_{mn}引脚输入的有效边沿，就将TCR_{mn}寄存器的计数值传送 (捕捉) 到定时器数据寄存器 mn (TDR_{mn})，同时将TCR_{mn}寄存器清“0000H”，然后输出INTT_{mn}。此时，如果计数器发生上溢，就将定时器状态寄存器mn (TSR_{mn}) 的OVF位置“1”。如果计数器没有发生上溢，就清除OVF位。此后，继续同样的运行。

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

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

将TMR_{mn}寄存器的STS_{mn}2~STS_{mn}0位置“001B”，并且将Tl_{mn}的有效边沿用于开始触发和捕捉触发。

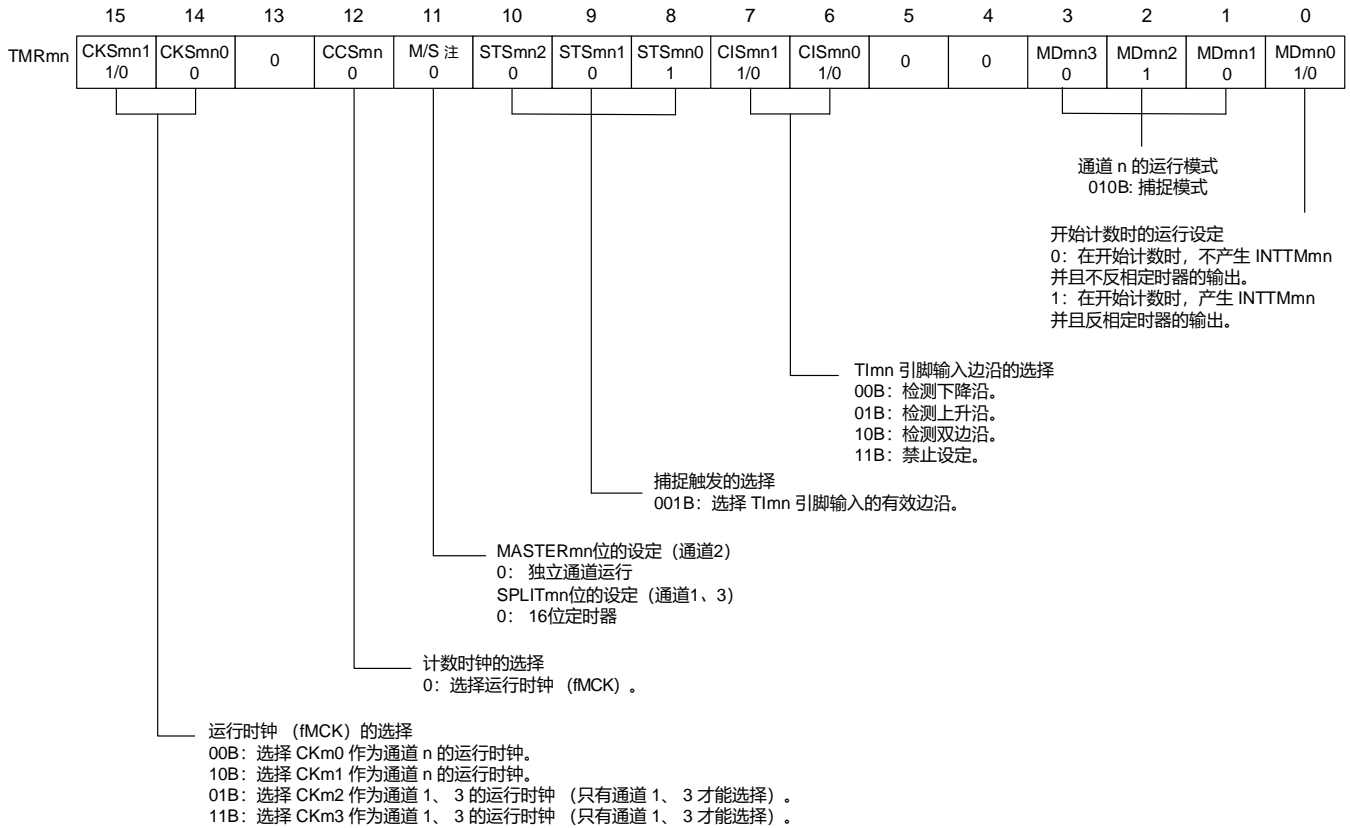
图5-52 作为输入脉冲间隔测量的运行基本时序例子 (MD_{mn}0=0)



- 备注1. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)
 2. TS_{mn} : 定时器通道开始寄存器m (TS_m) 的bit n
 TE_{mn} : 定时器通道允许状态寄存器m (TE_m) 的bit n
 Tl_{mn} : Tl_{mn}引脚输入信号
 TCR_{mn} : 定时器计数寄存器mn (TCR_{mn})
 TDR_{mn} : 定时器数据寄存器mn (TDR_{mn})
 OVF : 定时器状态寄存器mn (TSR_{mn}) 的bit0

图5-53 测量输入脉冲间隔时的寄存器设置内容例子

(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器m (TOM)

bit n
TOMn 0: 由TOMn 输出“0”。

(c) 定时器输出允许寄存器m (TOEm)

bit n
TOEmn 0: 停止由计数运行进行的TOMn 输出。

(d) 定时器输出电平寄存器m (TOLm)

bit n
TOLmn 0: 在主导通道输出模式 (TOMmn=0) 中置“0”。

(e) 定时器输出模式寄存器m (TOMm)

bit n
TOMmn 0: 设置主导通道输出模式。

注 TMRm2 : MASTERmn位
TMRm1、TMRm3 : SPLITmn位
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-54 输入脉冲间隔测量功能时的操作步骤

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

重新
开始
运行

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TlImn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TlImn的信号宽度（高低电平宽度）。TlImn的信号宽度能用以下计算式进行计算。

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

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

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

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

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

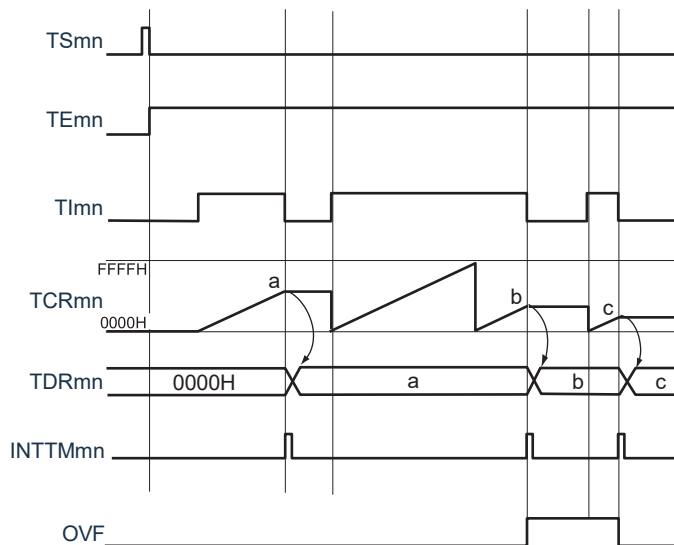
即使计数器进行了2个周期或者2个周期以上的完整计数，也认为发生上溢而将TSRmn寄存器的OVF位置“1”。但是，在发生2次或者2次以上的上溢时，无法通过OVF位正常测量间隔值。

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

TMRmn寄存器的CISmn1、CISmn0=10B：测量低电平宽度。

TMRmn寄存器的CISmn1、CISmn0=11B：测量高电平宽度。

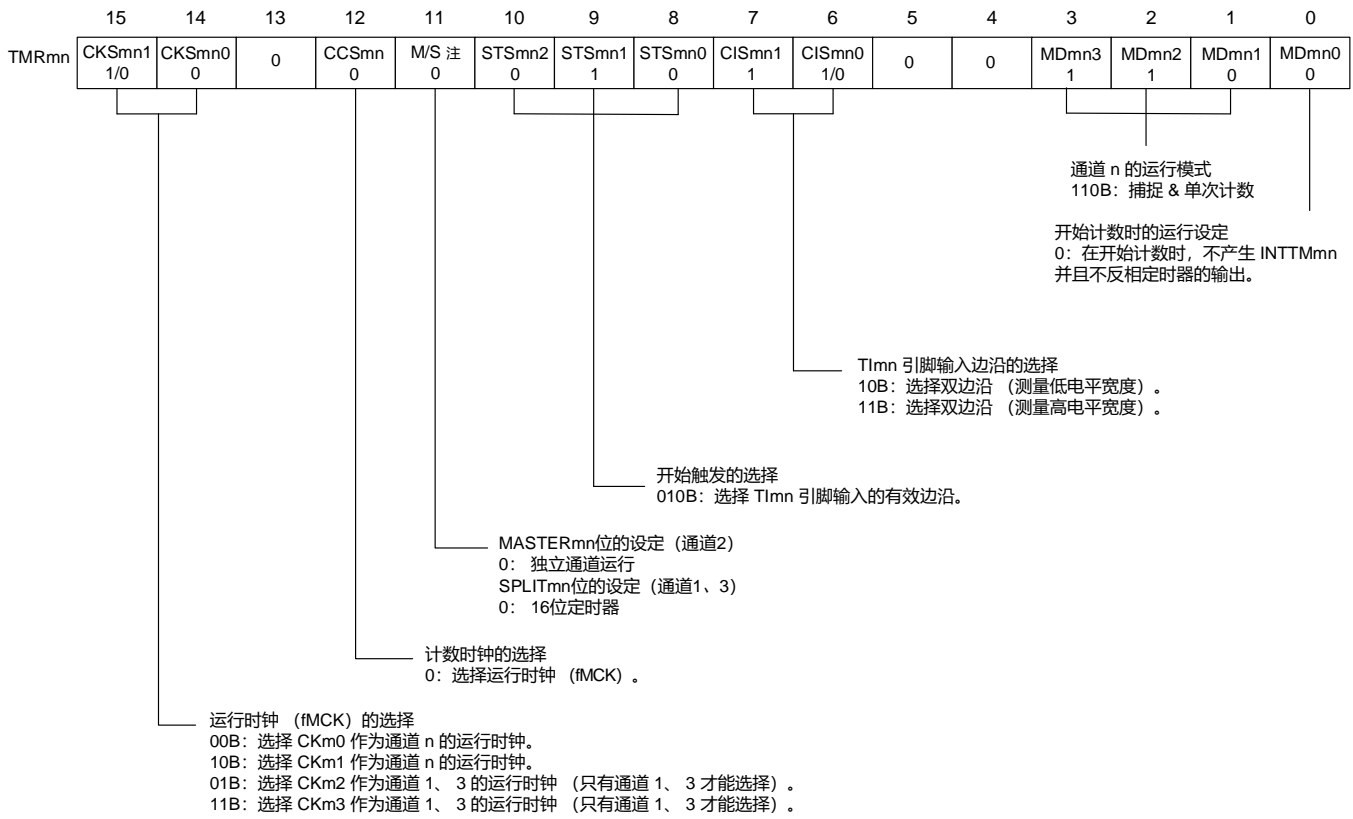
图5-55 作为输入信号高低电平宽度测量的运行基本时序例子



- 备注1. m: 单元号 (m=0, 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
 - TE mn : 定时器通道允许状态寄存器m (TE m) 的bit n
 - TI mn : TI mn引脚输入信号
 - TCRmn: 定时器计数寄存器mn (TCRmn)
 - TDRmn: 定时器数据寄存器mn (TDRmn)
 - OVF : 定时器状态寄存器mn (TSRmn) 的bit0

图5-56 测量输入信号的高低电平宽度时的寄存器设置内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)



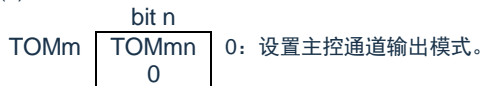
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



注 TMRm2 : MASTERmn位
TMRm1、TMRm3 : SPLITmn位
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-57 输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
Timer4 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的TM4mEN 位置 “1”。	定时器单元m 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm)。 确定CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器1 (NFEN1) 的对应位置 “0” (OFF) 或者 “1” (ON)。 设定定时器模式寄存器mn (TMRmn) (确定通道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。
	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。
重新 开始 运行	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 禁止更改 TMRmn 寄存器、TOMmn 位、TOLmn 位、TOmn 位和 TOEmn 位的设定值。	在检测到 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。

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

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

中断产生周期能用以下计算式进行计算：

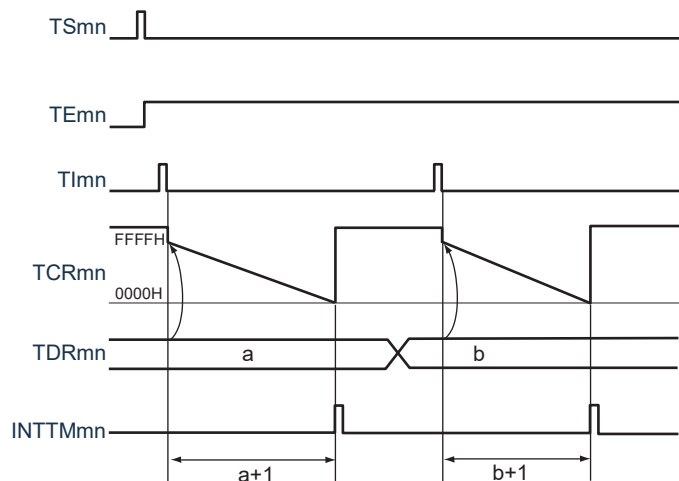
$$\text{INTTM}_{mn} \text{ (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR}_{mn} \text{ 的设定值} + 1)$$

在单次计数模式中，定时器计数寄存器mn（TCR_{mn}）用作递减计数器。

如果将定时器通道开始寄存器m（TS_m）的通道开始触发位（TS_{mn}、TSH_{m1}、TSH_{m3}）置“1”，TE_{mn}位、TEH_{m1}位和TEH_{m3}位就变为“1”，并且进入Tl_{mn}引脚的有效边沿检测等待状态。通过Tl_{mn}引脚输入的有效边沿检测，开始TCR_{mn}寄存器的运行，并且装入定时器数据寄存器mn（TDR_{mn}）的值。TCR_{mn}寄存器通过计数时钟，从装入的TDR_{mn}寄存器的值开始递减计数。如果TCR_{mn}变为“0000H”，就输出INTTM_{mn}，并且在检测到下一个Tl_{mn}引脚输入的有效边沿前停止计数。

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

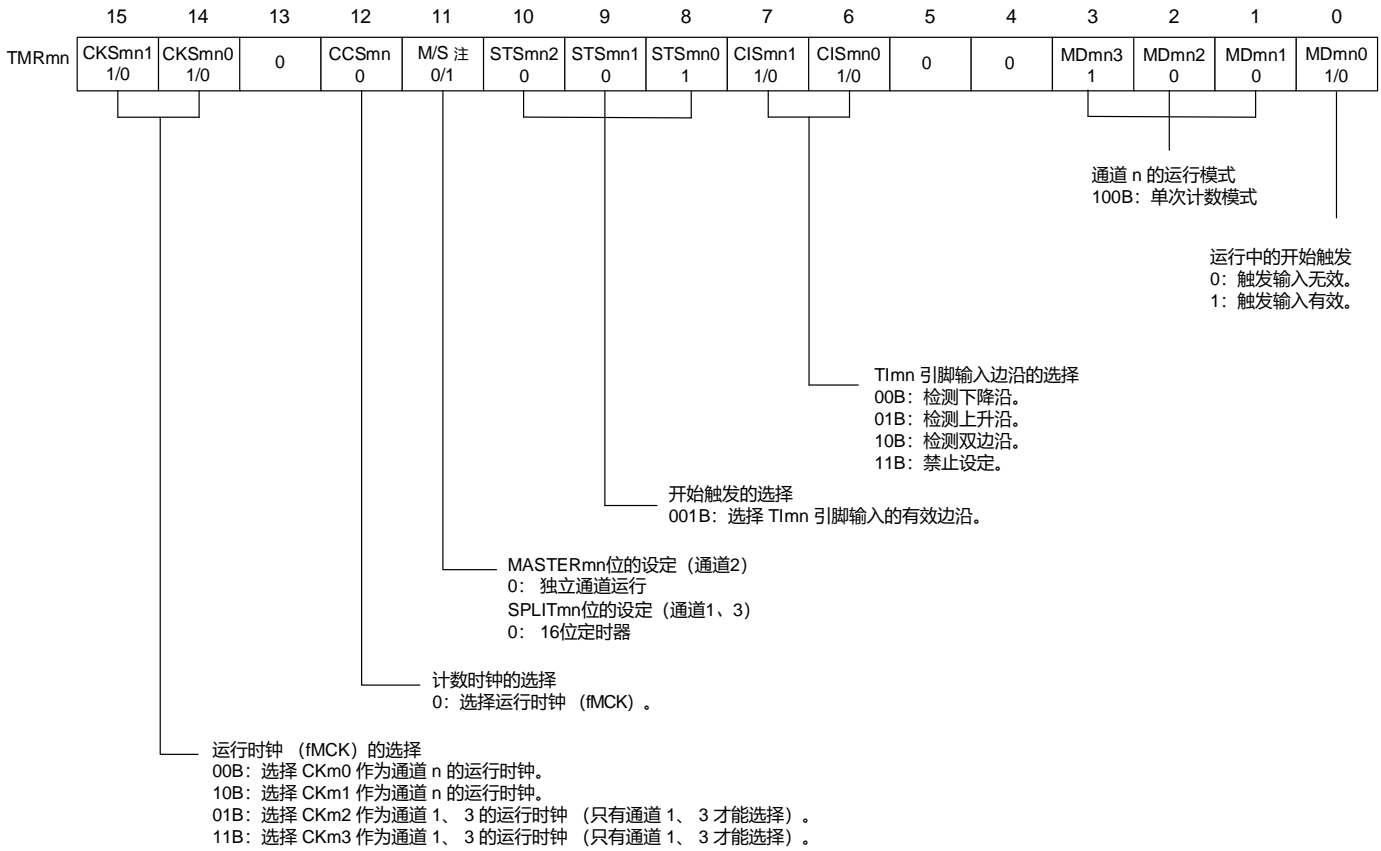
图5-58 作为延迟计数器的运行基本时序例子



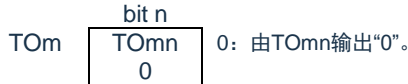
- 备注1. m: 单元号 (m=0, 1) n: 通道号 (n=0~3)
2. TS_{mn} : 定时器通道开始寄存器m (TS_m) 的bit n
 TE_{mn} : 定时器通道允许状态寄存器m (TE_m) 的bit n
 Tl_{mn} : Tl_{mn}引脚输入信号
 TCR_{mn}: 定时器计数寄存器mn (TCR_{mn})
 TDR_{mn}: 定时器数据寄存器mn (TDR_{mn})

图5-59 延迟计数器功能时的寄存器设置内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)



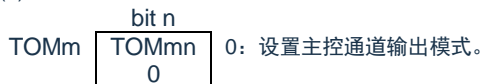
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



注 TMRm2 : MASTERmn位
TMRm1、 TMRm3 : SPLITmn位
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3)

图5-60 延迟计数器功能时的操作步骤

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

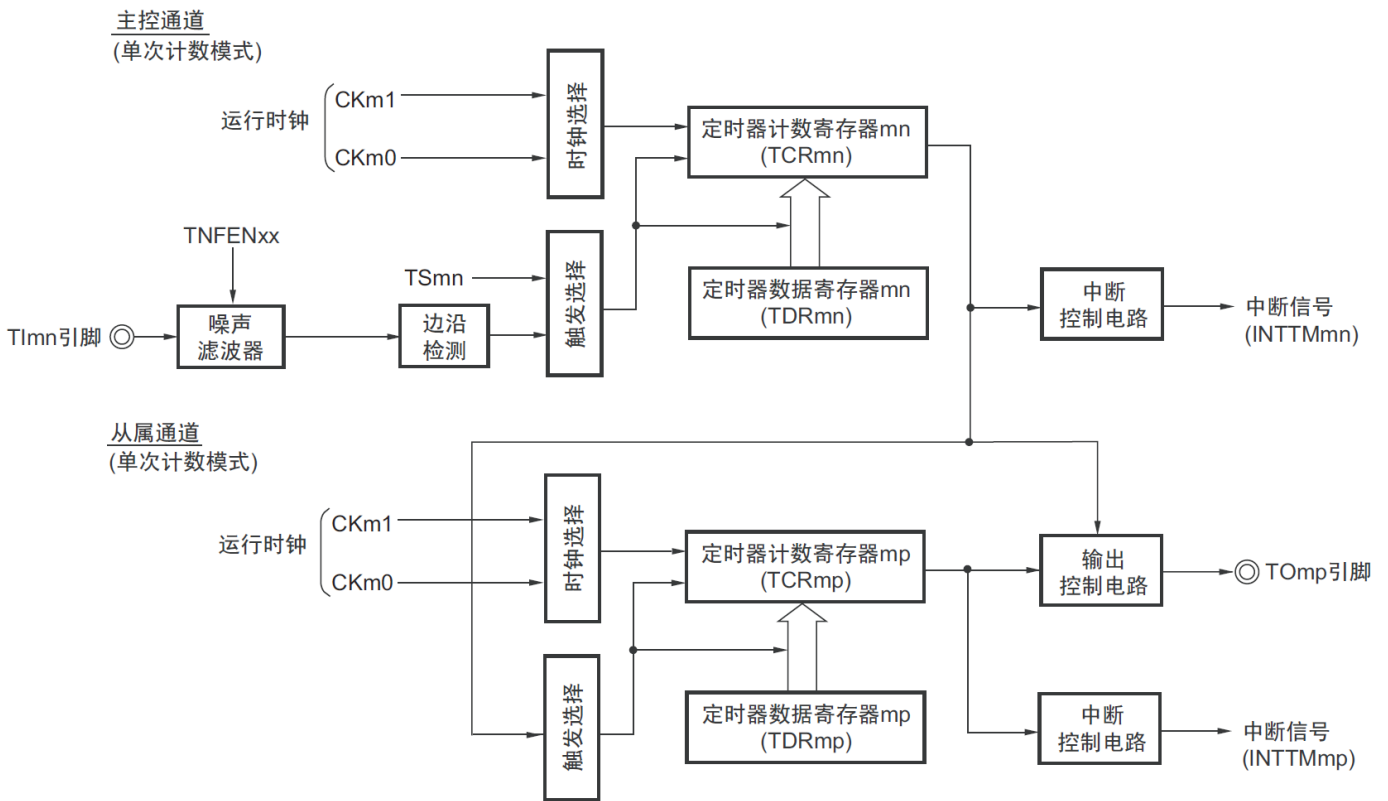
在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的INTTMmn作为开始触发，从属通道的TCRmp寄存器开始运行并且装入TDRmp寄存器的值。TCRmp寄存器通过计数时钟，从装入的TDRmp寄存器值开始递减计数。如果计数值变为“0000H”，就输出INTTMmp，并且在检测到下一个开始触发（主控通道的INTTMmn）前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOMP的输出电平变为有效电平，如果TCRmp变为“0000H”，就变为无效电平。

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。

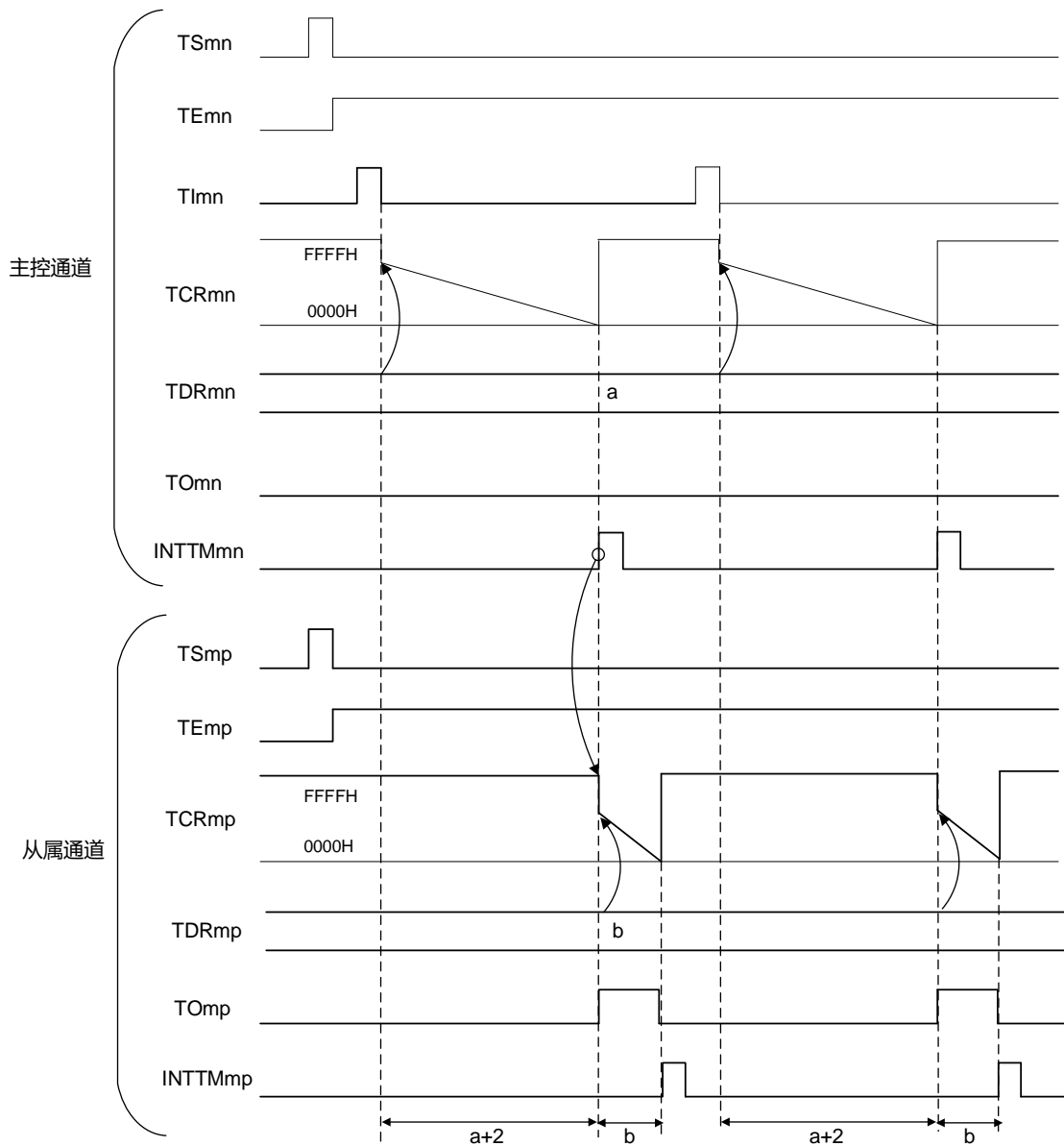
备注 m: 单元号 (m=0, 1) n: 主控通道号 (n=0, 2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

图5-61 作为单触发脉冲输出功能运行的框图



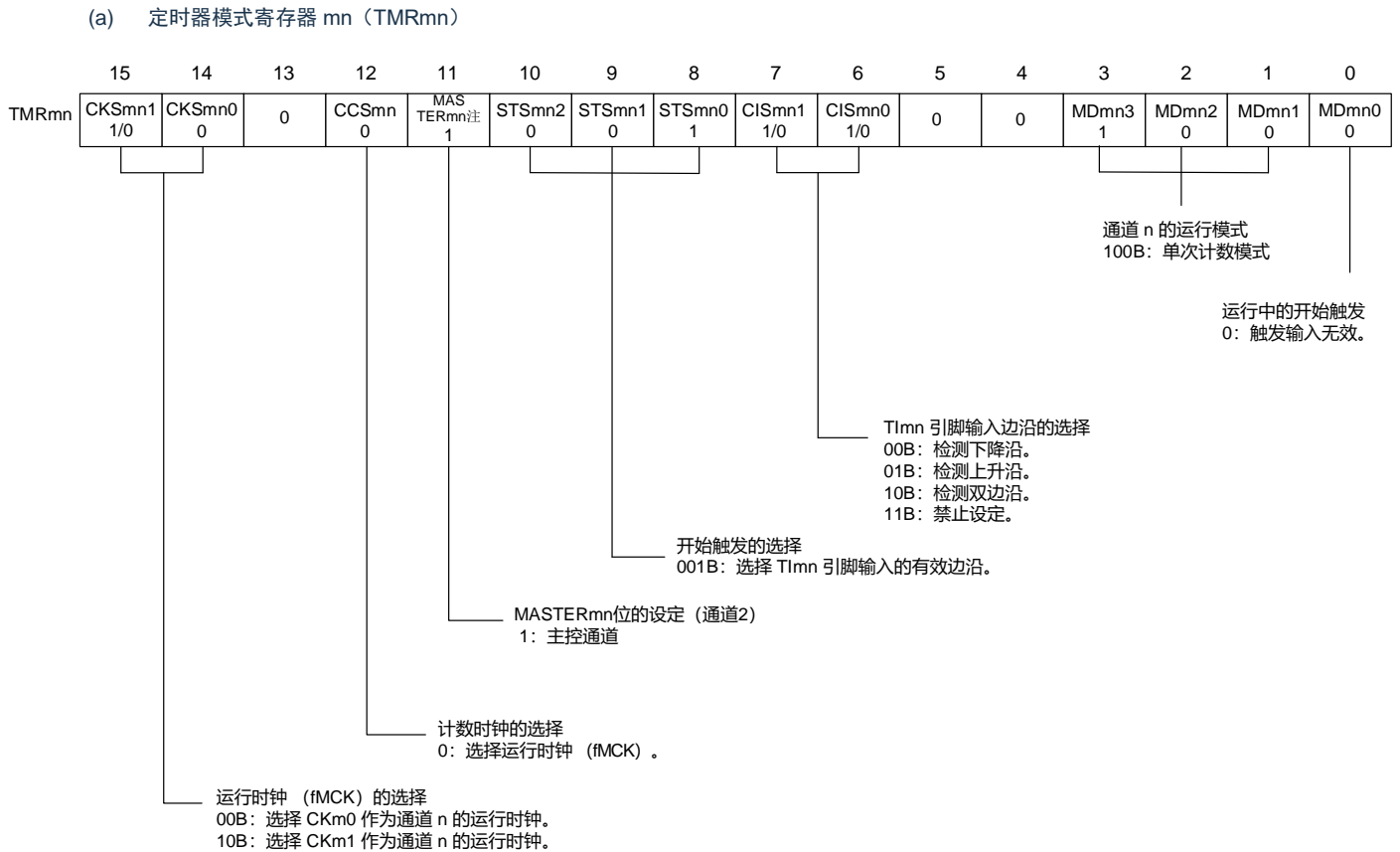
备注m: 单元号 (m=0, 1) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

图5-62 作为单触发脉冲输出功能的运行基本时序例子



- 备注1. m: 单元号 (m=0, 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5-63 单触发脉冲输出功能时（主控通道）的寄存器设置内容例子



(b) 定时器输出寄存器 m (TOM)

bit n
TOM bit n: TOMn 0: 由TOMn 输出“0”。

(c) 定时器输出允许寄存器 m (TOEm)

bit n
TOEm bit n: TOEmn 0: 停止由计数运行进行的TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n
TOLm bit n: TOLmn 0: 在TOMmn=0 (主控通道输出模式) 时置“0”。

(e) 定时器输出模式寄存器 m (TOMm)

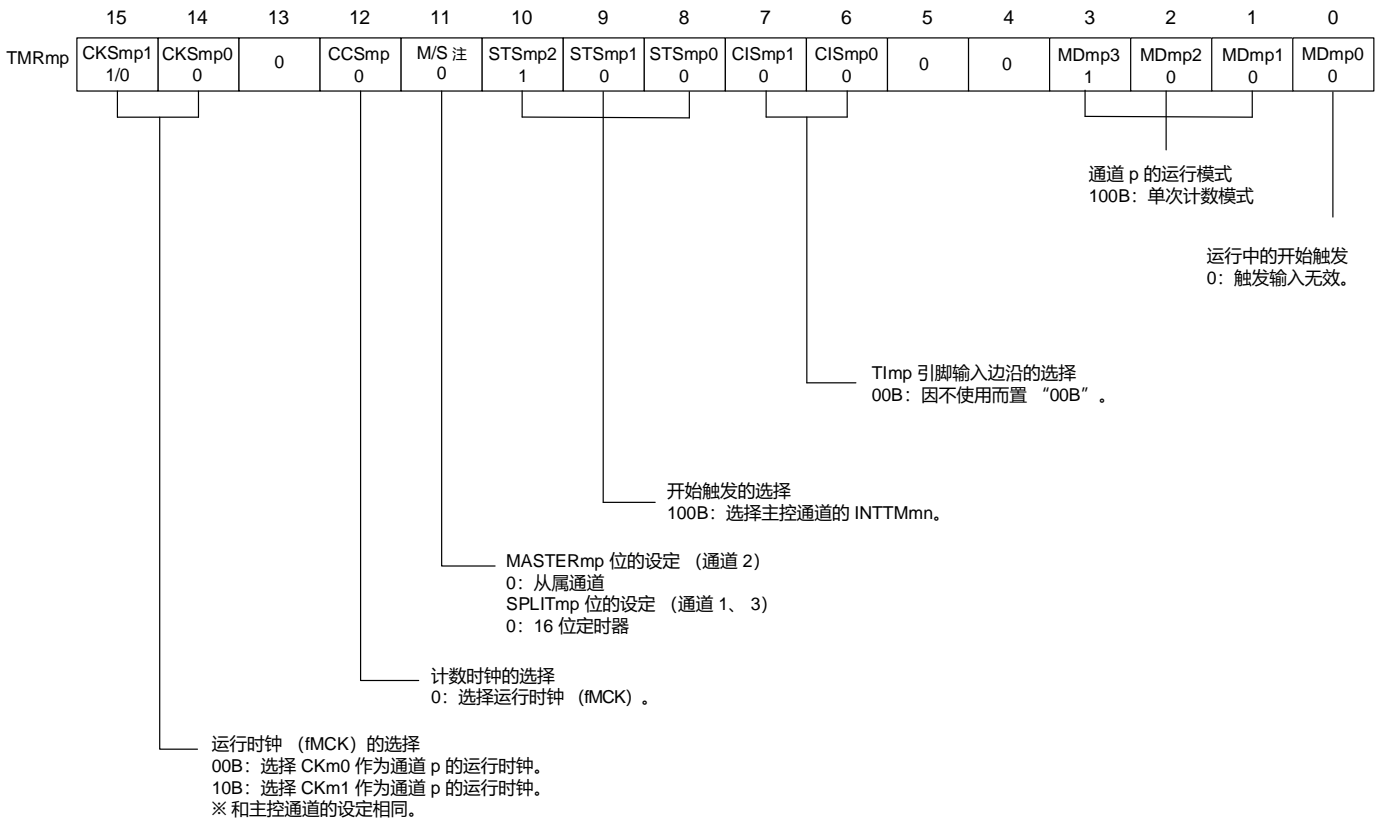
bit n
TOMm bit n: TOMmn 0: 设置主控通道输出模式。

注 TMRm2 : MASTERmn=1
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0, 1) n: 主控通道号 (n=0、2)

图5-64 单触发脉冲输出功能时（从属通道）的寄存器设置内容例子

(a) 定时器模式寄存器 mp (TMRmp)



(b) 定时器输出寄存器 m (TOM)

bit p
TOM

TOMp
1/0

 0: 由TOMp 输出“0”。
1: 由TOMp 输出“1”。

(c) 定时器输出允许寄存器 m (TOEm)

bit p
TOEm

TOEmp
1/0

 0: 停止由计数运行进行的TOMp 输出。
1: 允许由计数运行进行的TOMp 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit p
TOLm

TOLmp
1/0

 0: 正逻辑输出 (高电平有效)
1: 负逻辑输出 (低电平有效)

(e) 定时器输出模式寄存器 m (TOMm)

bit p
TOMm

TOMmp
1

 1: 设置从属通道输出模式。

注 TMRm2: MASTERmp位
TMRm1、 TMRm3: SPLITmp位

备注 m: 单元号 (m=0, 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5-65 单触发脉冲输出功能时的操作步骤(1/2)

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

图5-66 单触发脉冲输出功能时的操作步骤(2/2)

重新开始运行	开始运行	<p>将 TOEmp 位 (从属) 置 “1” (只限于重新开始运行)。</p> <p>将定时器通道开始寄存器 m (TSm) 的 TSmn (主控) 和 TSmp (从属) 位同时置 “1”。</p> <p>因为 TSmn 位和 TSmp 位是触发位, 所以自动返回 “0”。</p>	<p>TEmn 位和 TEmn 位都变为 “1”, 主控通道进入开始触发 (检测 TImn 引脚输入的有效边沿或者将主控通道的 TSmn 位置 “1”) 的检测等待状态。计数器还处于停止状态。</p>
		<p>通过检测主控通道的开始触发, 开始主控通道的计数。</p> <ul style="list-style-type: none"> 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。 	<p>主控通道开始计数。</p>
	运行中	<p>只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。</p> <p>禁止更改 TMRmp、TDRmn、TDRmp 寄存器以及 TOMmn 位、TOMmp 位、TOLmn 位和 TOLmp 位的设定值。</p> <p>能随时读 TCRmn 寄存器和 TCRmp 寄存器。</p> <p>不使用 TSRmn 寄存器和 TSRmp 寄存器。</p> <p>能更改从属通道的 TOM 寄存器和 TOEm 寄存器的设定值。</p>	<p>主控通道通过检测开始触发 (检测 TImn 引脚输入的有效边沿或者将主控通道的 TSmn 位置 “1”), 将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn), 并且进行递减计数。如果 TCRmn 计数到 “0000H”, 就产生 INTTMmn, 并且在下一次 TImn 引脚输入前停止计数。</p> <p>从属通道以主控通道的INTTMmn为触发, 将TDRmp寄存器的值装入 TCRmp 寄存器并且计数器开始递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后, 将TOmp的输出电平置为有效电平。然后, 如果 TCRmp 计数到 “0000H”, 就在将 TOmp的输出电平置为无效电平后停止计数。此后, 重复此运行。</p>
	停止运行	<p>将 TTmn 位 (主控) 和 TTmp 位 (从属) 同时置 “1”。</p> <p>因为 TTmn 位和 TTmp 位是触发位, 所以自动返回到 “0”。</p>	<p>TEmn 位和 TEmn 位都变为 “0” 并且停止计数。</p> <p>TCRmn 寄存器和 TCRmp 寄存器保持计数值而停止计数。 TOmp 输出不被初始化而保持状态。</p>
	<p>将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设定值。</p>	<p>TOmp 引脚输出 TOmp 设定的电平。</p>	
Timer4 停止	<p>要保持 TOmp 引脚输出电平的情况:</p> <p>在给端口寄存器设定要保持的值后将 TOmp 位置 “0”。</p> <p>不需要保持 TOmp 引脚输出电平的情况:</p> <p>不需要设定。</p>	<p>通过端口功能保持 TOmp 引脚的输出电平。</p>	
	<p>将 PER0 寄存器的 TM4mEN 位置 “0”。</p>	<p>定时器单元 m 的输入时钟处于停止提供状态。对全部电路和各通道的 SFR 进行初始化。</p>	

注 不能将从属通道的 TSmn 位置 “1”。

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

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：

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

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

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

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

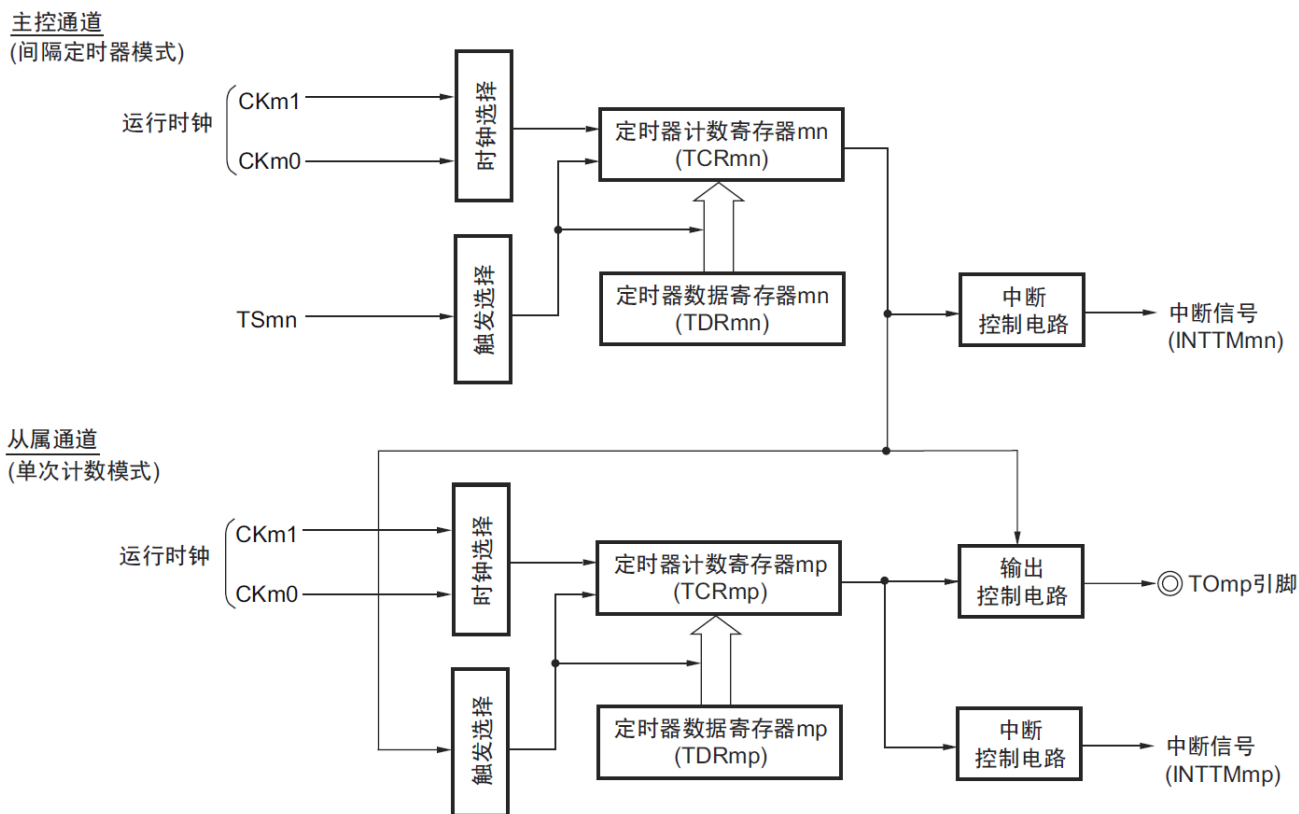
当用作PWM功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为PWM输出（TOmp）的占空比。

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

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

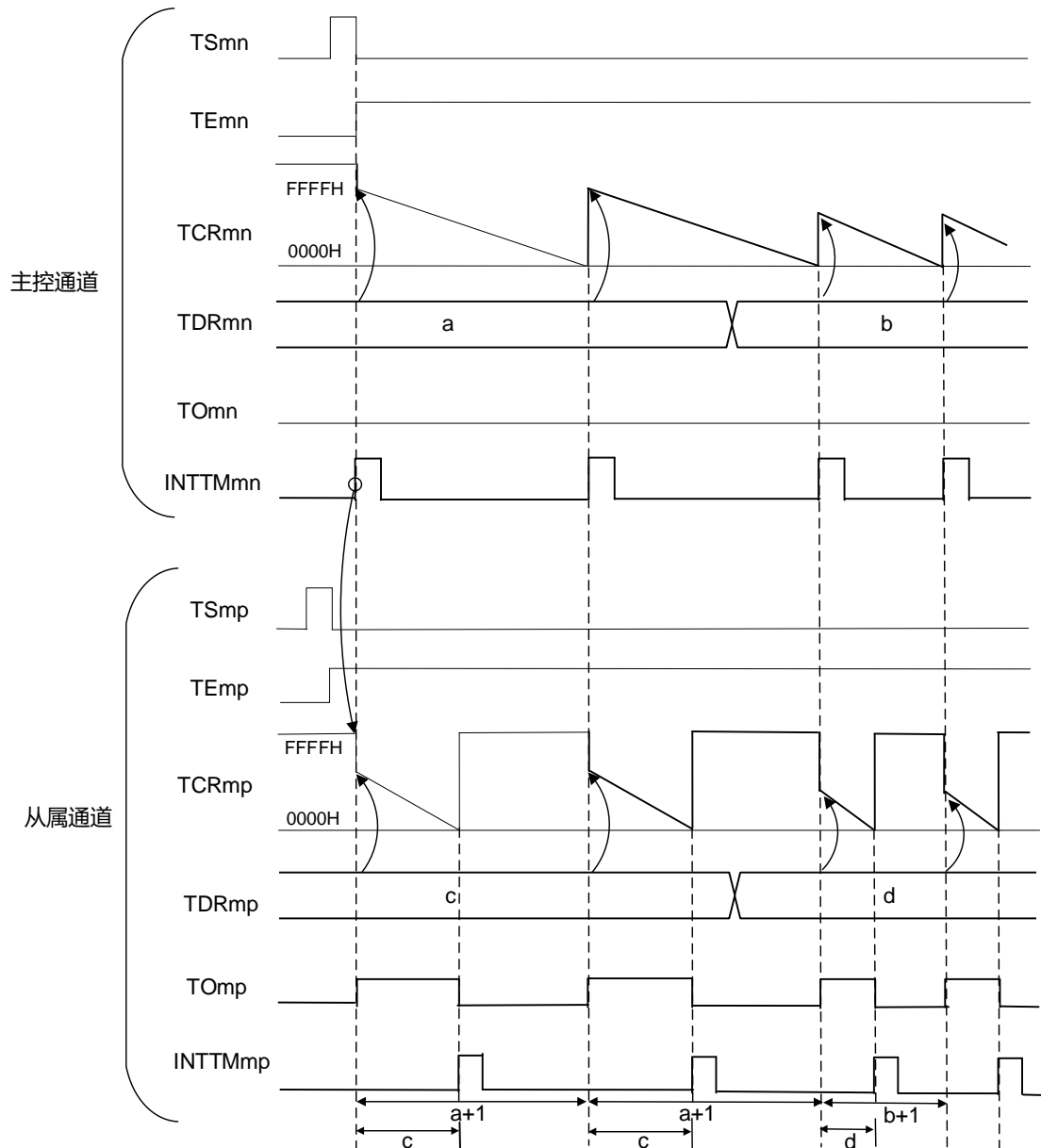
备注 m：单元号（m=0, 1） n：主控通道号（n=0、2） p：从属通道号（n=0：p=1、2、3，n=2：p=3）

图5-67 作为PWM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)

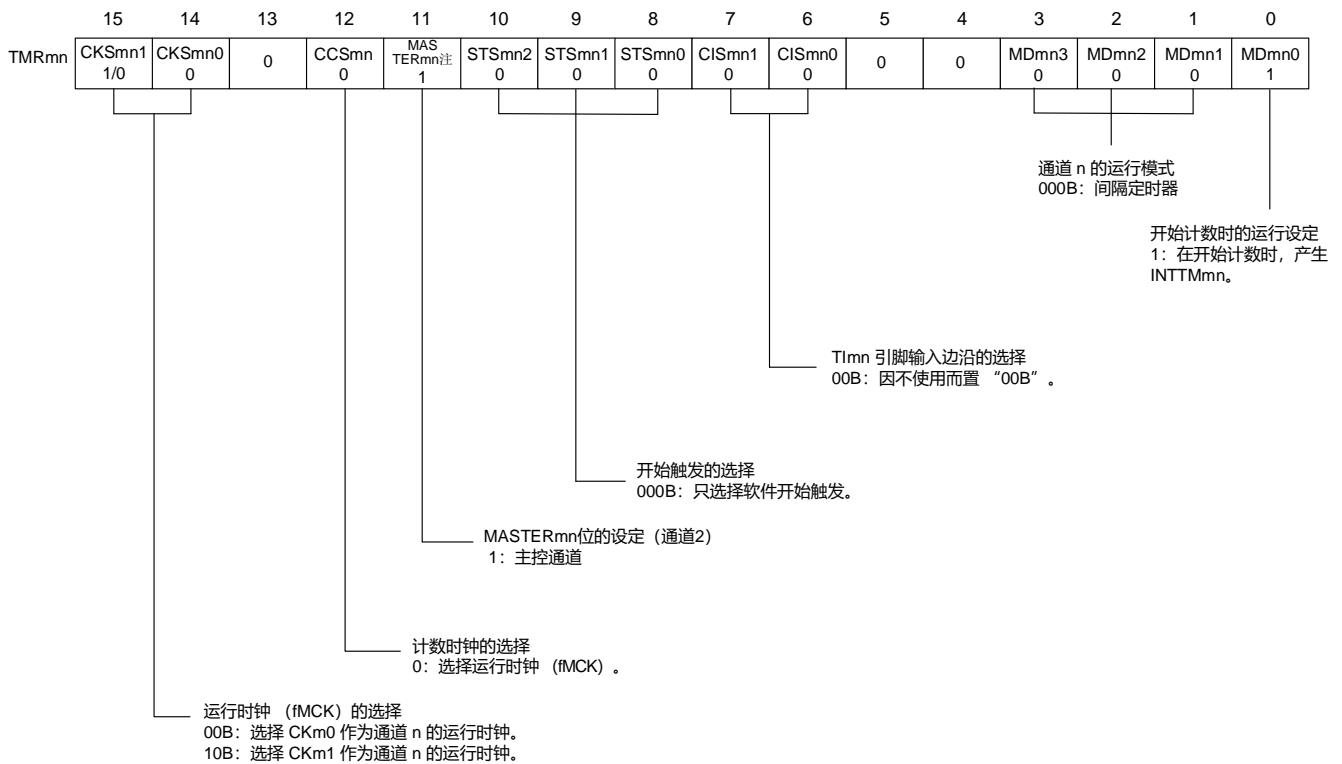
图5-68 作为PWM 功能的运行基本时序例子



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
 TE mn、TE mp : 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5-69 PWM功能时（主控通道）的寄存器设置内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)

bit n
TOMn 0 0: 由TOMn输出“0”。

(c) 定时器输出允许寄存器 m (TOEm)

bit n
TOEmn 0 0: 停止由计数运行进行的TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n
TOLmn 0 0: 在TOMmn=0 (主控通道输出模式) 时置“0”。

(e) 定时器输出模式寄存器 m (TOMm)

bit n
TOMmn 0 0: 设置主控通道输出模式。

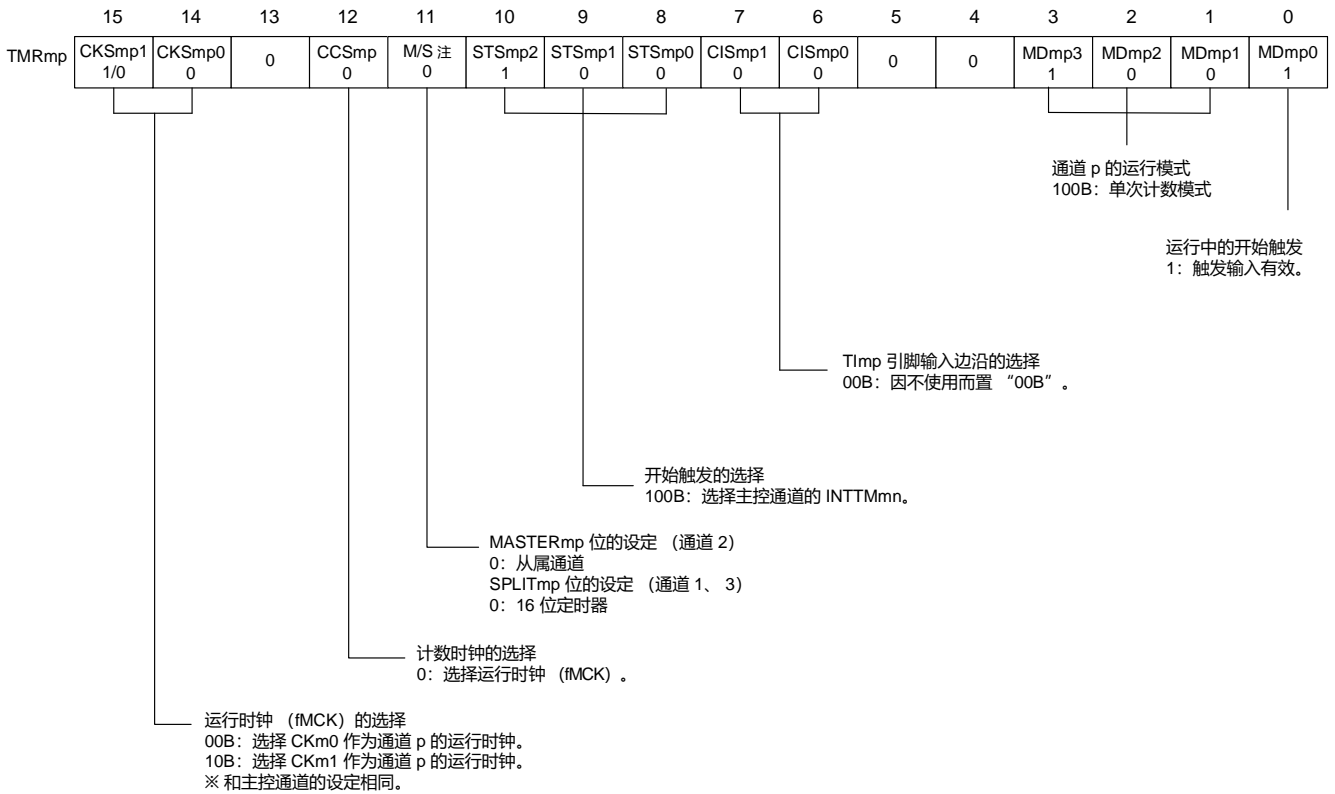
注 TMRm2 : MASTERmn=1

TMRm0 : 固定为“0”。

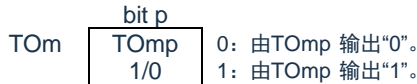
备注 m: 单元号 (m=0, 1) n: 主控通道号 (n=0, 2)

图5-70 PWM功能时（从属通道）的寄存器设置内容例子

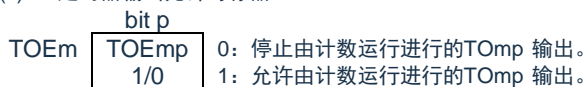
(a) 定时器模式寄存器 mp (TMRmp)



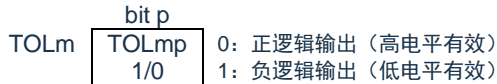
(b) 定时器输出寄存器 m (TOM)



(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



注 TMRm2 : MASTERmp位

TMRm1、TMRm3 : SPLITmp位

备注 m: 单元号 (m=0, 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5-71 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。

图5-72 PWM功能时的操作步骤(2/2)

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

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

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重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：

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

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

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

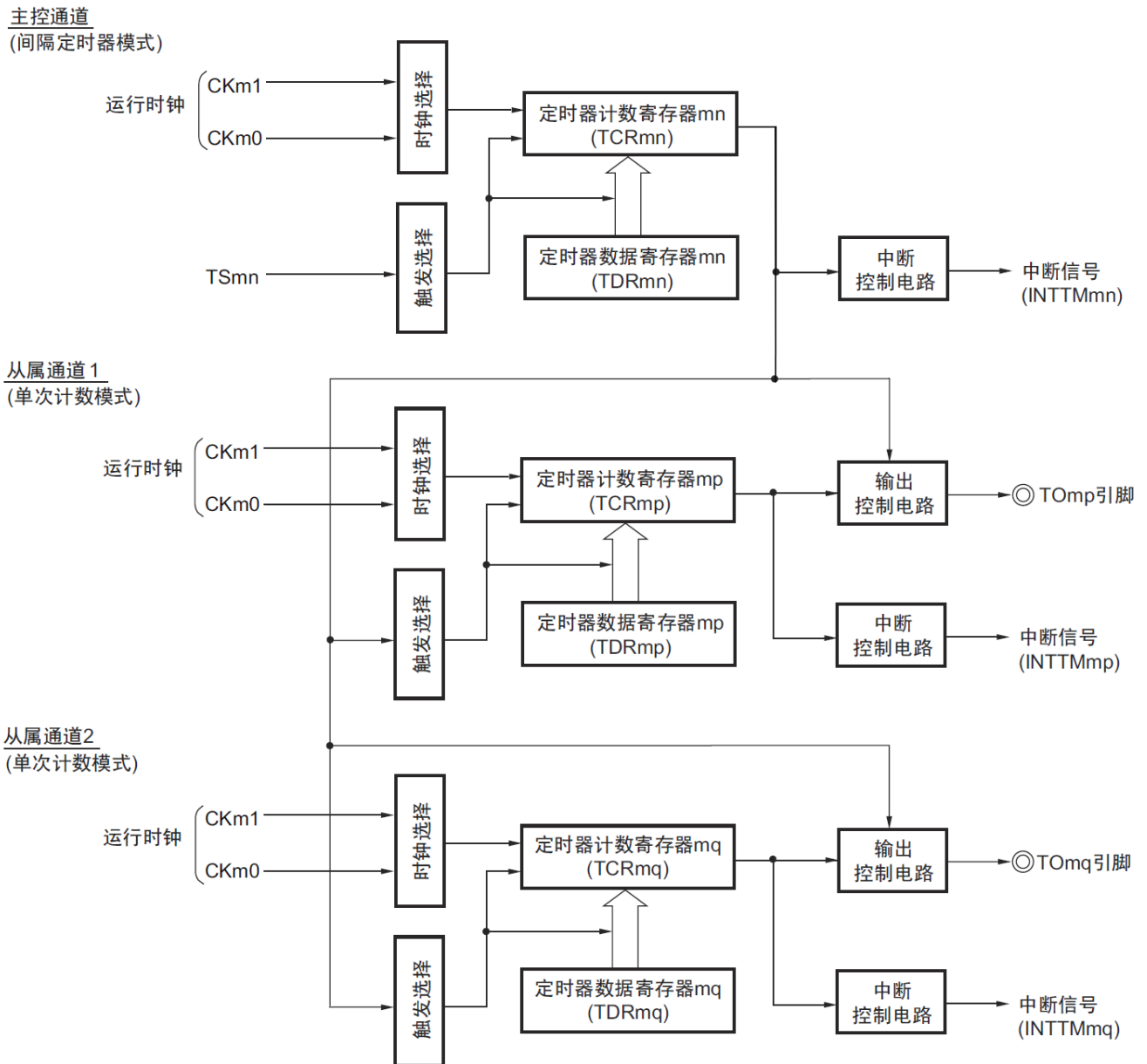
和从属通道1的TCRmp寄存器相同，在单次计数模式中，从属通道2的TCRmq寄存器运行并且对占空比进行计数以及从TOMq引脚输出PWM波形。以主控通道的INTTMmn为开始触发，将TDRmq寄存器的值装入TCRmq寄存器并且进行递减计数。如果TCRmq变为“0000H”，就输出INTTMmq，并且在输入下一个开始触发（主控通道的INTTMmn）前停止计数。在从主控通道产生INTTMmn并且经过1个计数时钟后，TOMq的输出电平变为有效电平，如果TCRmq变为“0000H”，就变为无效电平。

当通过如此的运行将通道0用作主控通道时，最多能同时输出3种PWM信号。

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

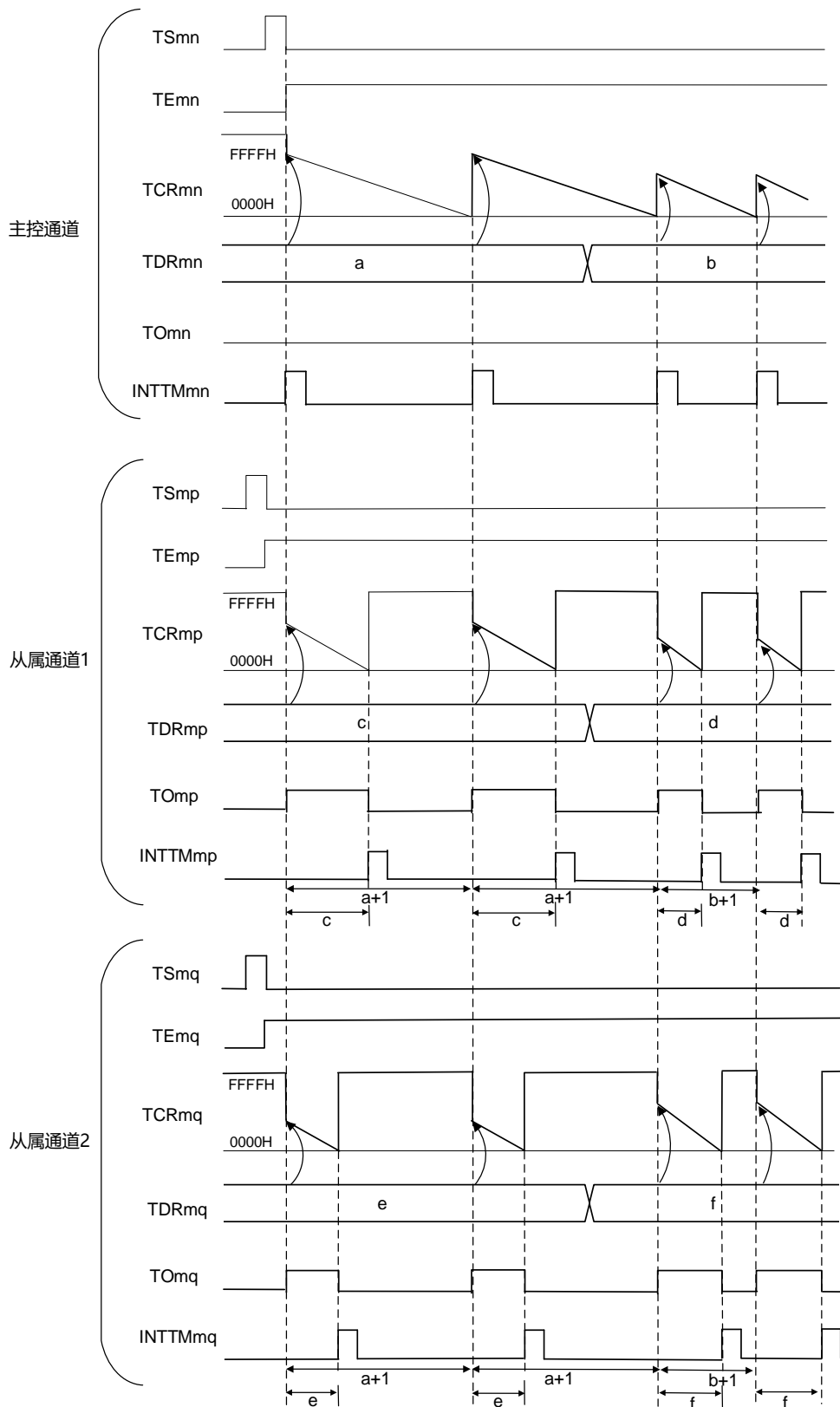
备注 m: 单元号 (m=0, 1) n: 主控通道号 (n=0)
 p: 从属通道号 q: 从属通道号
 $n < p < q \leq 3$ (p和q是大于n的整数)

图5-73 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
 $n < p < q \leq 3$ (p和q是大于n的整数)

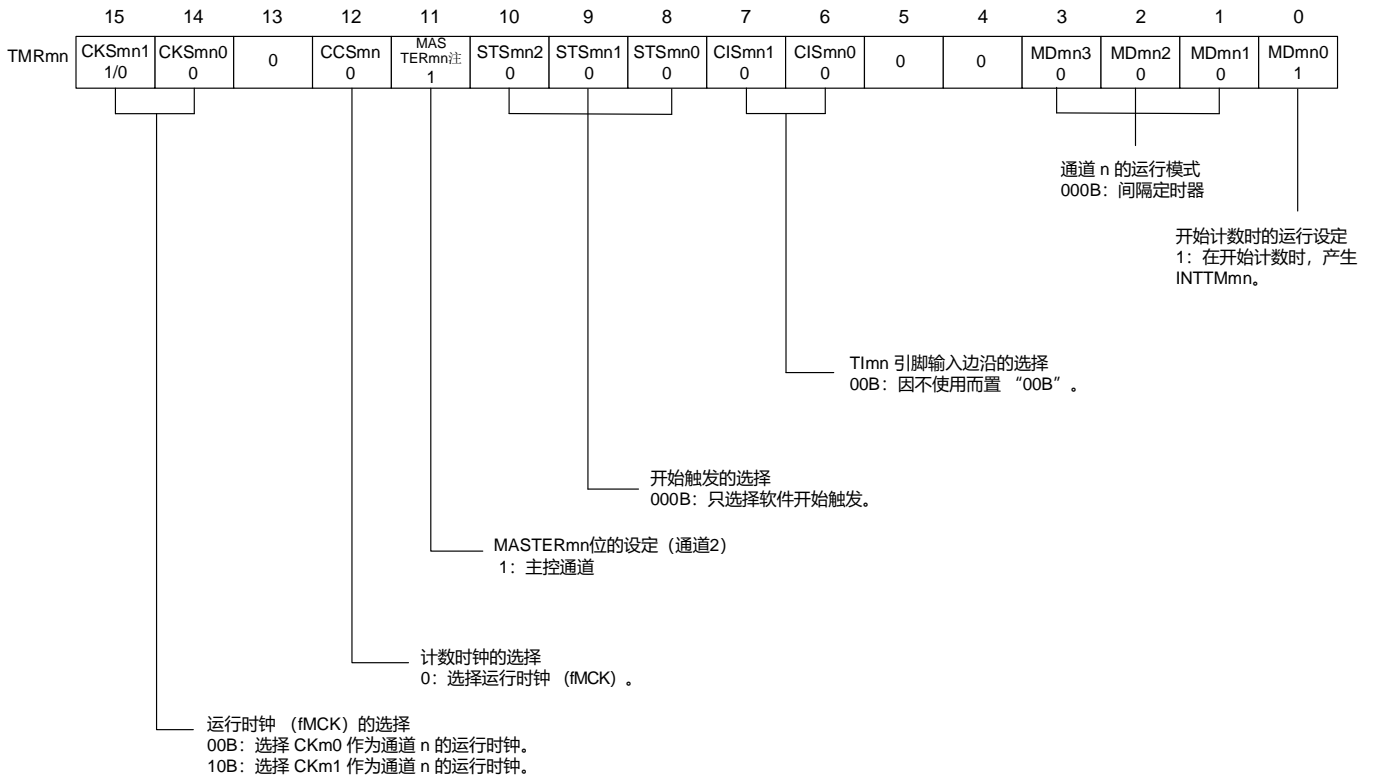
图5-74 作为多重PWM输出功能的运行基本时序例子（输出2种PWM的情况）



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

图5-75 多重PWM输出功能时（主控通道）的寄存器设置内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)

bit n
TOM

TOMn
0

 0: 由TOMn输出“0”。

(c) 定时器输出允许寄存器 m (TOEm)

bit n
TOEm

TOEmn
0

 0: 停止由计数运行进行的TOMn输出。

(d) 定时器输出电平寄存器 m (TOLm)

bit n
TOLm

TOLmn
0

 0: 在TOMmn=0 (主控通道输出模式) 时置“0”。

(e) 定时器输出模式寄存器 m (TOMm)

bit n
TOMm

TOMmn
0

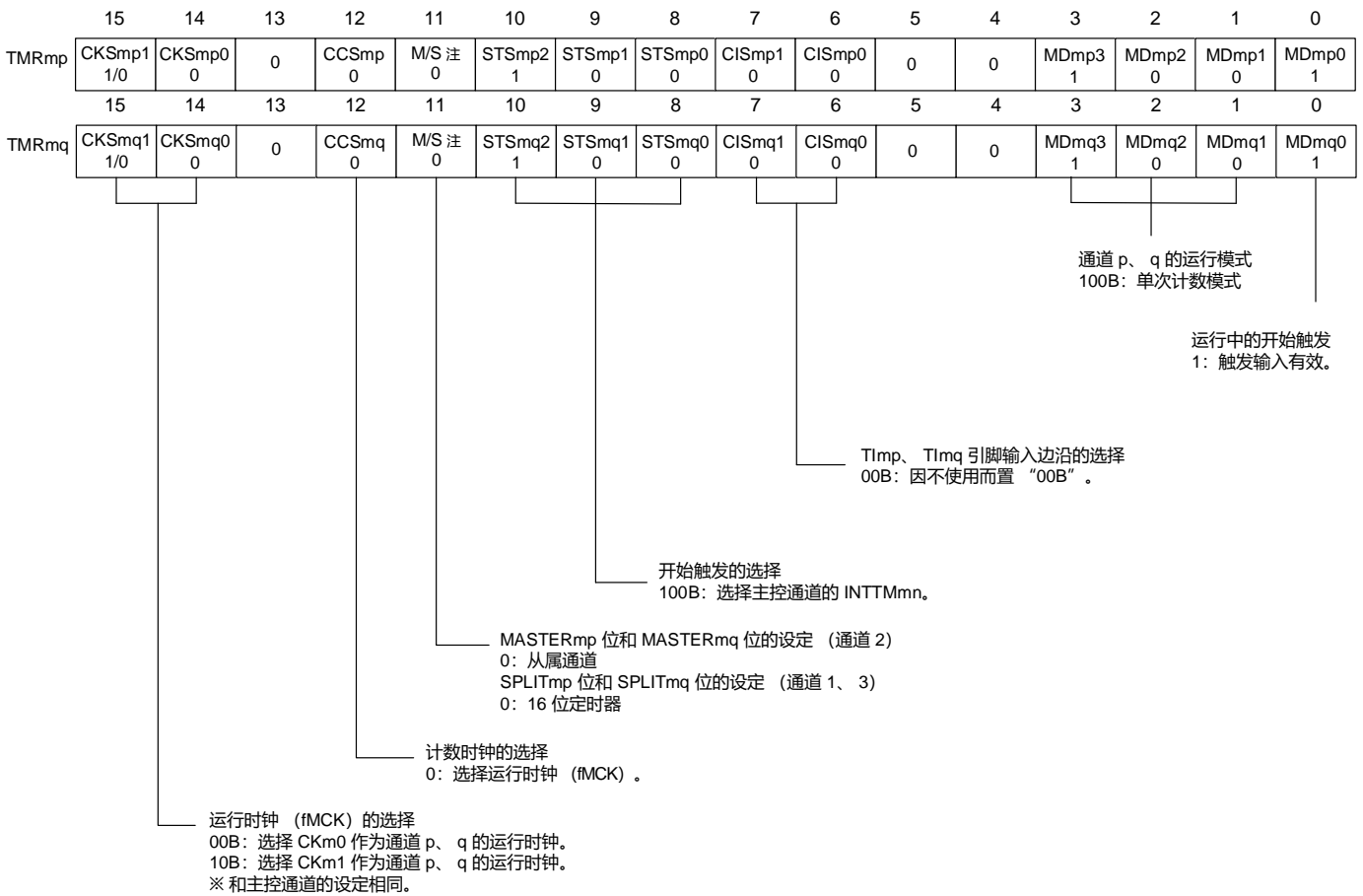
 0: 设置主控通道输出模式。

注 TMRm2 : MASTERmn=1
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0, 1) n: 主控通道号 (n=0)

图5-76 多重PWM输出功能时（从属通道）的寄存器设置内容例子（输出2种PWM的情况）

(a) 定时器模式寄存器 mp、mq (TMRmp、TMRmq)



(b) 定时器输出寄存器 m (TOM)

	bit q	bit p	
TOM	TOmq 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	TOEmq 1/0	TOEmp 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	TOELq 1/0	TOELp 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	TOMLq 1	TOMLp 1	1: 设置从属通道输出模式。

注 TMRm2 : MASTERmp位、MASTERmq位
TMRm1、 TMRm3 : SPLITmp位、SPLITmq位
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5-77 多重PWM输出功能时的操作步骤（输出2种PWM的情况）(1/2)

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

图5-78 多重PWM输出功能时的操作步骤（输出2种PWM的情况）(2/2)

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

备注 m: 单元号 (m=0, 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)

第6章 实时时钟

6.1 实时时钟的功能

实时时钟有以下功能。

- 持有年、月、星期、日、小时、分钟和秒的计数器，最长能计数到99年。
- 固定周期中断功能（周期：0.5秒、1秒、1分钟、1小时、1日、1个月）
- 闹钟中断功能（闹钟：星期、小时、分钟）
- 1Hz的引脚输出功能

6.2 实时时钟的结构

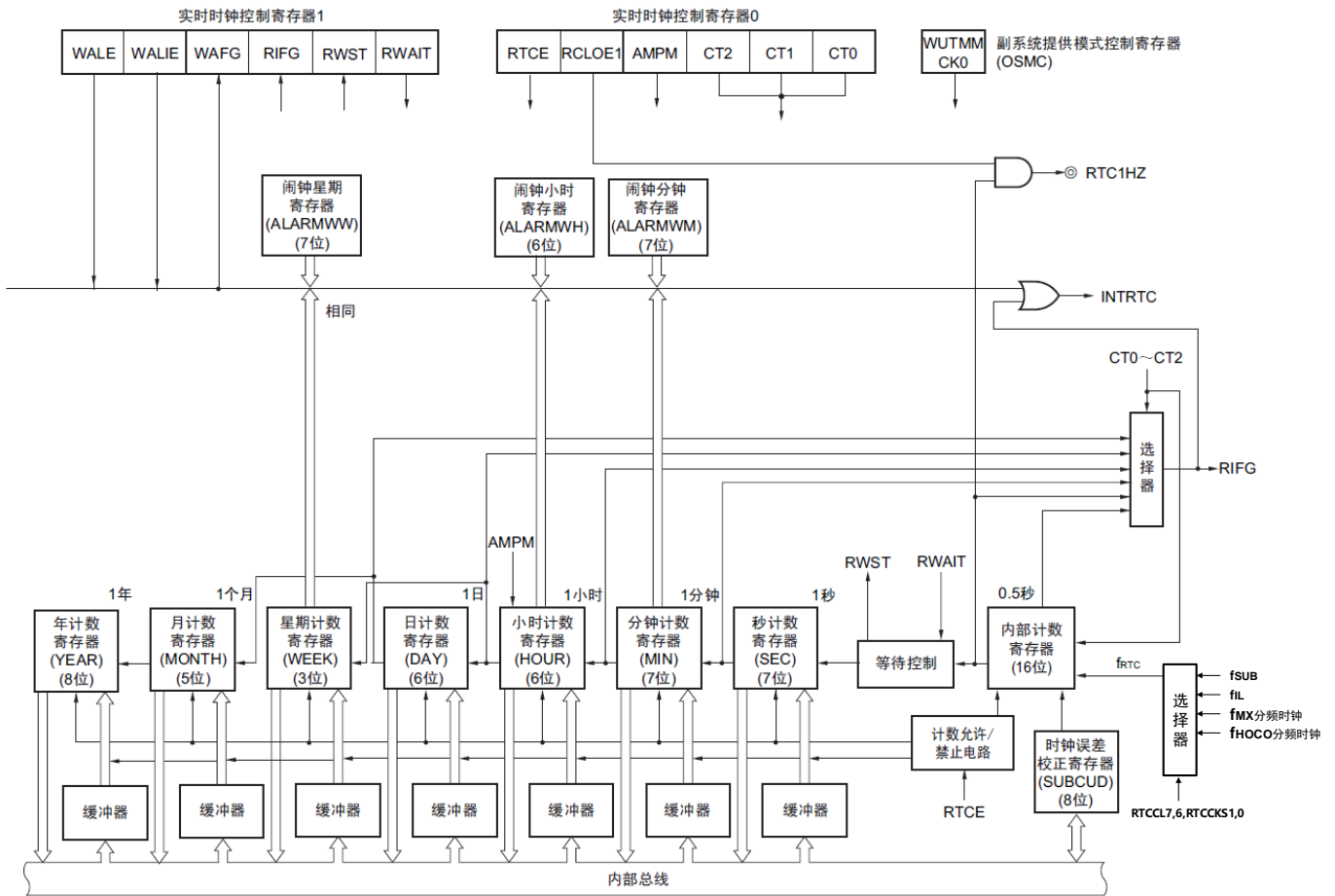
实时时钟由以下硬件构成。

表6-1 实时时钟的结构

项目	结构
计数器	内部计数器（16位）
控制寄存器	外围允许寄存器0（PER0.bit7）
	实时时钟选择寄存器（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：以上RTC控制寄存器的复位只受POR复位控制。

图6-1 实时时钟的框图



注意 只有在选择fmx/fhoco的分频时钟（分周后≈32,768KHZ）或者副系统时钟（f_{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_{IL}=30kHz）时，只能使用固定周期中断功能。

选择f_{IL}时的固定周期中断间隔用以下计算式进行计算：

$$\text{固定周期(RTCC0寄存器选择的值)} \quad f_{\text{SUB}}/f_{\text{IL}}$$

6.3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)

6.3.1 外围允许寄存器0 (PER0)

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

要使用实时时钟时，必须将bit7 (RTCEN) 置“1”。通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图6-2外围允许寄存器0 (PER0) 的格式

地址: 0x40020420	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	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。 <ul style="list-style-type: none"> 不能写实时时钟 (RTC) 和15位间隔定时器使用的SFR。 实时时钟 (RTC) 和15位间隔定时器处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> 能读写实时时钟 (RTC) 和15位间隔定时器使用的SFR。

注意1.如果要使用实时时钟，必须先在计数时钟 (f_{RTC}) 振荡稳定的状态下将RTCEN位置“1”，然后设置以下的寄存器。当RTCEN位为“0”时，忽视实时时钟控制寄存器的写操作，而且读取值为初始值 (实时时钟选择寄存器 (RTCCL)、端口模式寄存器和端口寄存器除外)。

- 实时时钟控制寄存器0 (RTCC0)
- 实时时钟控制寄存器1 (RTCC1)
- 秒计数寄存器 (SEC)
- 分钟计数寄存器 (MIN)
- 小时计数寄存器 (HOUR)
- 日计数寄存器 (DAY)
- 星期计数寄存器 (WEEK)
- 月计数寄存器 (MONTH)
- 年计数寄存器 (YEAR)
- 时钟误差校正寄存器 (SUBCUD)
- 闹钟分钟寄存器 (ALARMWM)
- 闹钟小时寄存器 (ALARMWH)
- 闹钟星期寄存器 (ALARMWW)

2.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。

6.3.2 实时时钟选择寄存器 (RTCCL)

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

图6-3 实时时钟选择寄存器 (RTCCL) 的格式

地址: 0x4004047C	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	0	0	0	0	RTCKS1	RTCKS0

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

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

6.3.3 实时时钟控制寄存器0 (RTCC0)

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

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

图6-4 实时时钟控制寄存器0 (RTCC0) 的格式

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

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

RCLOE1	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11-2所示。

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

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

注意1.在RTCE位为“1”时，不能更改RCLOE1位。

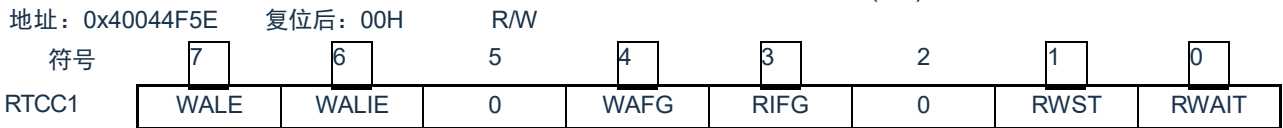
2.在RTCE位为“0”时，即使将RCLOE1位置“1”也不输出1Hz。

备注 ×：忽略

6.3.4 实时时钟控制寄存器1 (RTCC1)

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

图6-5 实时时钟控制寄存器1 (RTCC1) 的格式(1/2)



WALE	闹钟的运行控制
0	一致运行无效。
1	一致运行有效。

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

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

WAFG	闹钟检测状态标志
0	闹钟不一致。
1	检测到闹钟一致。

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

图6-5 实时时钟控制寄存器1 (RTCC1) 的格式(2/2)

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

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

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

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

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

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

备注 1.固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这2个中断的情况下，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）。

6.3.5 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置SUBCUD寄存器。在产生复位信号后，此寄存器的值变为“0000H”。

图6-6 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置							
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置							
0	$\{(F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)-1\} \times 2$ 增加							
1	$\{(\overline{F11}, \overline{F10}, \overline{F9}, \overline{F8}, \overline{F7}, \overline{F6}, \overline{F5}, \overline{F4}, \overline{F3}, \overline{F2}, \overline{F1}, \overline{F0})+1\} \times 2$ 减少							
当 $(F12,F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$ 、 $(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1)$ 、 $(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$ 或者 $(1,0,0,0,0,0,0,0,0,0,0,0,0,0,1)$ 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								

注意：“/”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

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

6.3.6 秒计数寄存器 (SEC)

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

在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。必须以BCD码设置十进制的00~59。

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

图6-7秒计数寄存器 (SEC) 的格式

地址: 0x40044F52 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“6.4.3 实时时钟计数器的读写”记载的步骤进行。

备注 如果写秒计数寄存器 (SEC)，就清除内部计数器 (16位)。

6.3.7 分钟计数寄存器 (MIN)

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

在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视秒计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的00~59。

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

图6-8分钟计数寄存器 (MIN) 的格式

地址: 0x40044F53 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“6.4.3实时时钟计数器的读写”记载的步骤进行。

6.3.8 小时计数寄存器 (HOUR)

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

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

必须根据实时时钟控制寄存器0 (RTCC0) 的bit3 (AMPM) 设置的时间系统，以BCD码设置十进制的00~23或者01~12、21~32。

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

但是，如果在复位后将AMPM位置“1”，此寄存器的值就变为“00H”。

图6-9小时计数寄存器 (HOUR) 的格式

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

注意

- 1.当选择AMPM位为“0”（12小时系统）时，HOUR寄存器的bit5（HOUR20）表示AM（0）/PM（1）。
- 2.要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“6.4.3实时时钟计数器的读写”记载的步骤进行。

AMPM位的设置值、小时计数寄存器（HOUR）的值和时间的关系如表6-2所示。

表6-2 时间位的表示

24小时表示 (AMPM=1)		12小时表示 (AMPM=0)	
时间	HOUR寄存器	时间	HOUR寄存器
0时	00H	AM12时	12H
1时	01H	AM1时	01H
2时	02H	AM2时	02H
3时	03H	AM3时	03H
4时	04H	AM4时	04H
5时	05H	AM5时	05H
6时	06H	AM6时	06H
7时	07H	AM7时	07H
8时	08H	AM8时	08H
9时	09H	AM9时	09H
10时	10H	AM10时	10H
11时	11H	AM11时	11H
12时	12H	PM12时	32H
13时	13H	PM1时	21H
14时	14H	PM2时	22H
15时	15H	PM3时	23H
16时	16H	PM4时	24H
17时	17H	PM5时	25H
18时	18H	PM6时	26H
19时	19H	PM7时	27H
20时	20H	PM8时	28H
21时	21H	PM9时	29H
22时	22H	PM10时	30H
23时	23H	PM11时	31H

当AMPM位为“0”时，HOUR寄存器的值为12小时表示；当AMPM位为“1”时，HOUR寄存器的值为24小时表示。

在12小时表示时，HOUR寄存器的bit5表示上午/下午。上午（AM）为“0”，下午（PM）为“1”。

6.3.9 日计数寄存器 (DAY)

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

- 01~31（1、3、5、7、8、10、12月）
- 01~30（4、6、9、11月）
- 01~29（2月闰年）
- 01~28（2月平常年）

在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视小时计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的01~31。

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

图6-10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“6.4.3实时时钟计数器的读写”记载的步骤进行。

6.3.10 星期计数寄存器 (WEEK)

这是用0~6（十进制）表示星期计数值的8位寄存器。与日计数器同步进行递增计数。在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。必须以BCD码设置十进制的00~06。

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

图6-11星期计数寄存器 (WEEK) 的格式

地址: 0x40044F55H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

注意1.月计数寄存器 (MONTH) 和日计数寄存器 (DAY) 的对应值不自动保存到星期寄存器 (WEEK)。必须在解除复位后进行以下的设置:

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

2.要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“6.4.3实时时钟计数器的读写”记载的步骤进行。

6.3.11 月计数寄存器 (MONTH)

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

在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视日计数寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的01~12。

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

图6-12 月计数寄存器 (MONTH) 的格式

地址：0x40044F57H 复位后：01H R/W

符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“6.4.3实时时钟计数器的读写”记载的步骤进行。

6.3.12 年计数寄存器 (YEAR)

这是用0~99（十进制）表示年计数值的8位寄存器。通过月计数器 (MONTH) 的上溢进行递增计数。00、04、08、.....、92、96是闰年。在写时，数据先被写到缓冲器，在经过最多2个 f_{RTC} 时钟后被写到计数器。在写操作过程中忽视MONTH寄存器的上溢并且设置为写入值。必须以BCD码设置十进制的00~99。

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

图6-13 年计数寄存器 (YEAR) 的格式

地址：0x40044F58H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

注意 要在计数器运行中 (RTCE=1) 读写此寄存器时，必须按照“6.4.3实时时钟计数器的读写”记载的步骤进行。

6.3.13 闹钟分钟寄存器 (ALARMWM)

这是设置闹钟分钟的寄存器。

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

注意 必须以BCD码设置十进制的00~59。如果设置范围以外的值，就不检测闹钟。

图6-14 闹钟分钟寄存器 (ALARMWM) 的格式

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

6.3.14 闹钟小时寄存器 (ALARMWH)

这是设置闹钟小时的寄存器。

通过8位存储器操作指令设置ALARMWH寄存器。在产生复位信号后，此寄存器的值变为“12H”。但是，如果在复位后将AMPM位置“1”，此寄存器的值就变为“00H”。

注意 必须以BCD码设置十进制的00~23或者01~12、21~32。如果设置范围以外的值，就不检测闹钟。

图6-15 闹钟小时寄存器 (ALARMWH) 的格式

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

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

6.3.15 闹钟星期寄存器 (ALARMWW)

这是设置闹钟星期的寄存器。

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

图6-16 闹钟星期寄存器 (ALARMWW) 的格式

地址: 0x40044F5CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

闹钟时间的设置例子如下所示。

闹钟设置时间	星期							12小时表示				24小时表示				
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	10时	1时	10分	1分	10时	1时	10分	1分	
	W	W	W	W	W	W	W									
每天 上午0时00分	0	1	2	3	4	5	6									
每天 上午1时30分	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0	0
每天 上午11时59分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9	
星期一至星期五 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	

6.3.16 端口模式寄存器和端口寄存器

1Hz输出时，RTC1HZ输出引脚可以复用到除P00以外的任意端口。

- 将输出端口的PxxCFG配置为“0x1c”。
- 将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”。

详细内容请参照“第2章 引脚功能 2.3.7端口复用功能配置寄存器 (PxxCFG)”。

6.4 实时时钟的运行

6.4.1 实时时钟的运行开始

图6-17 实时时钟的运行开始步骤



注 1.必须先将在计数时钟 (f_{RTC}) 振荡稳定的状态下将RTCEN位置“1”。

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“6.4.6实时时钟的时钟误差校正例子”。

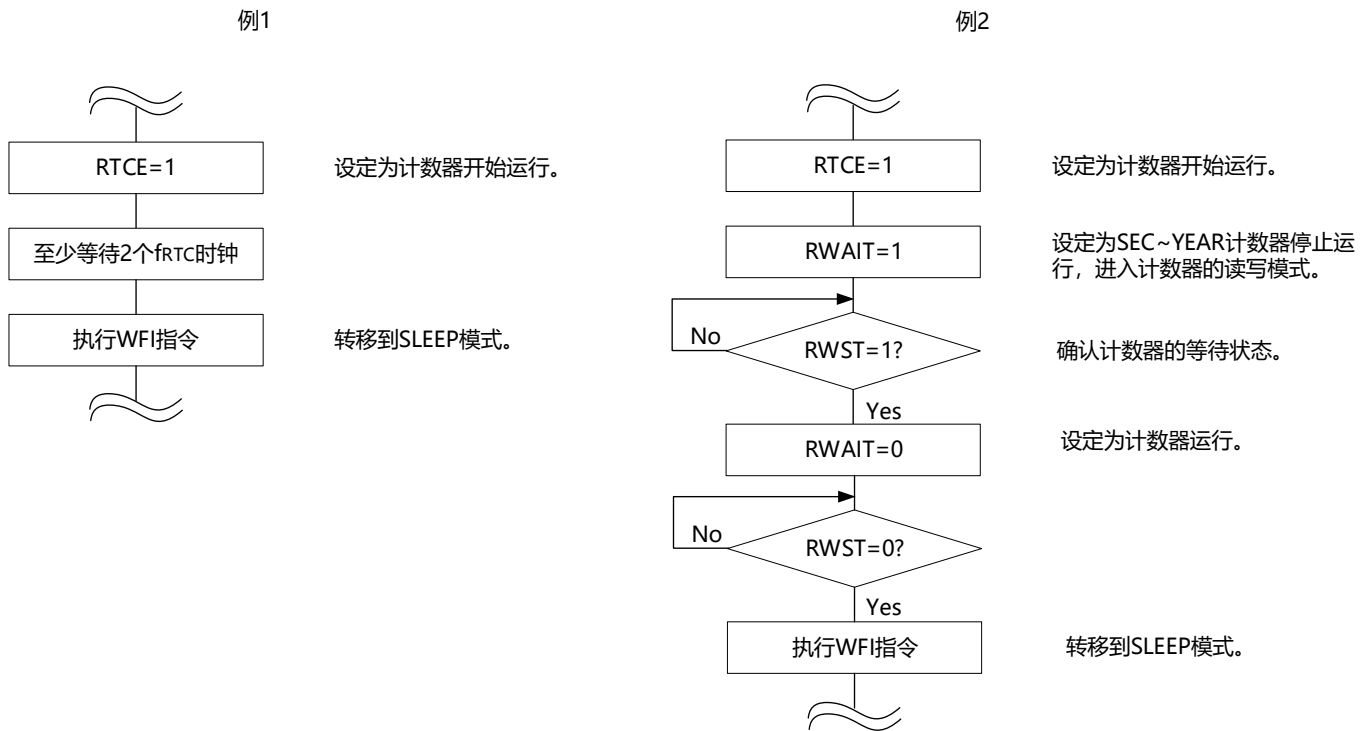
3.在RTCE位为“1”后不等待INTRTC位变为“1”而转移到睡眠模式的情况下, 请确认“6.4.2开始运行后睡眠模式的转移”的步骤。

6.4.2 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6-18的例1）。
- 在将RTCE位置“1”后将RWAIT位置“1”，通过轮询确认RWST位变为“1”。然后，将RWAIT位置“0”并且再次通过轮询确认RWST位变为“0”，然后转移到睡眠模式（参照图6-18的例2）。

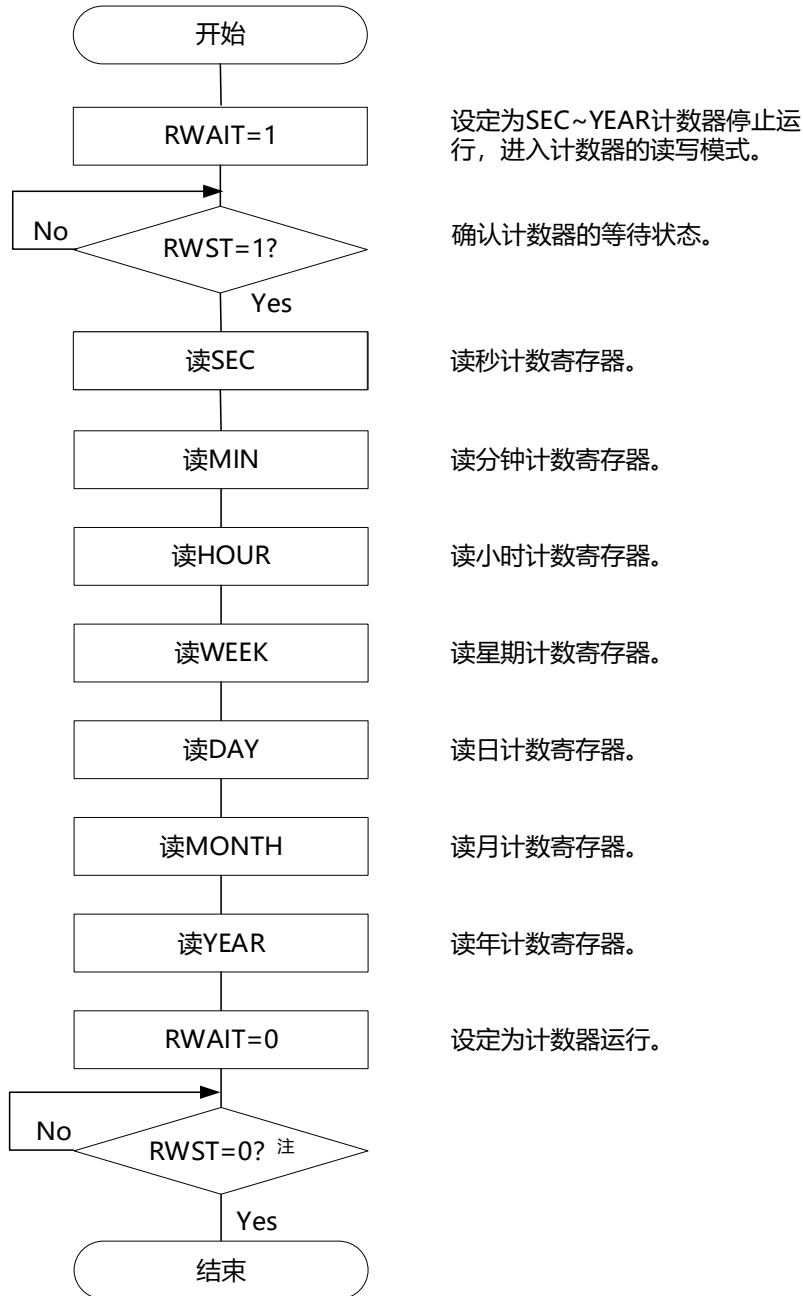
图6-18 将RTCE位置“1”后的睡眠/深度睡眠模式的转移步骤



6.4.3 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6-19 实时时钟计数器的读操作步骤

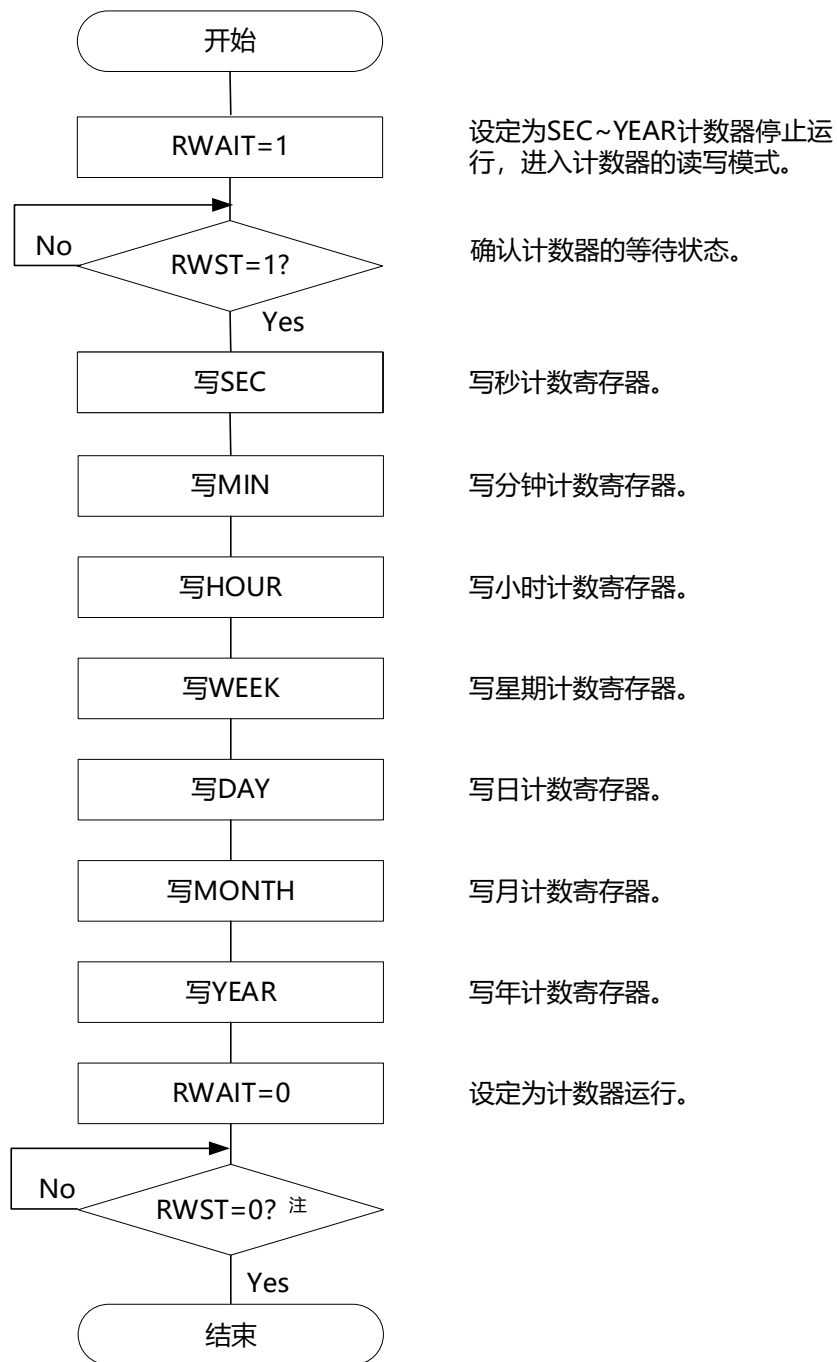


注 必须在转移到睡眠模式前，确认RWST位为“0”。

注意 必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。

备注 不限制秒/分钟/小时/星期/日/月和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。

图6-20 实时时钟计数器的读操作步骤



注 必须在转移到SLEEP模式前，确认RWST位为“0”。

注意1. 必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。

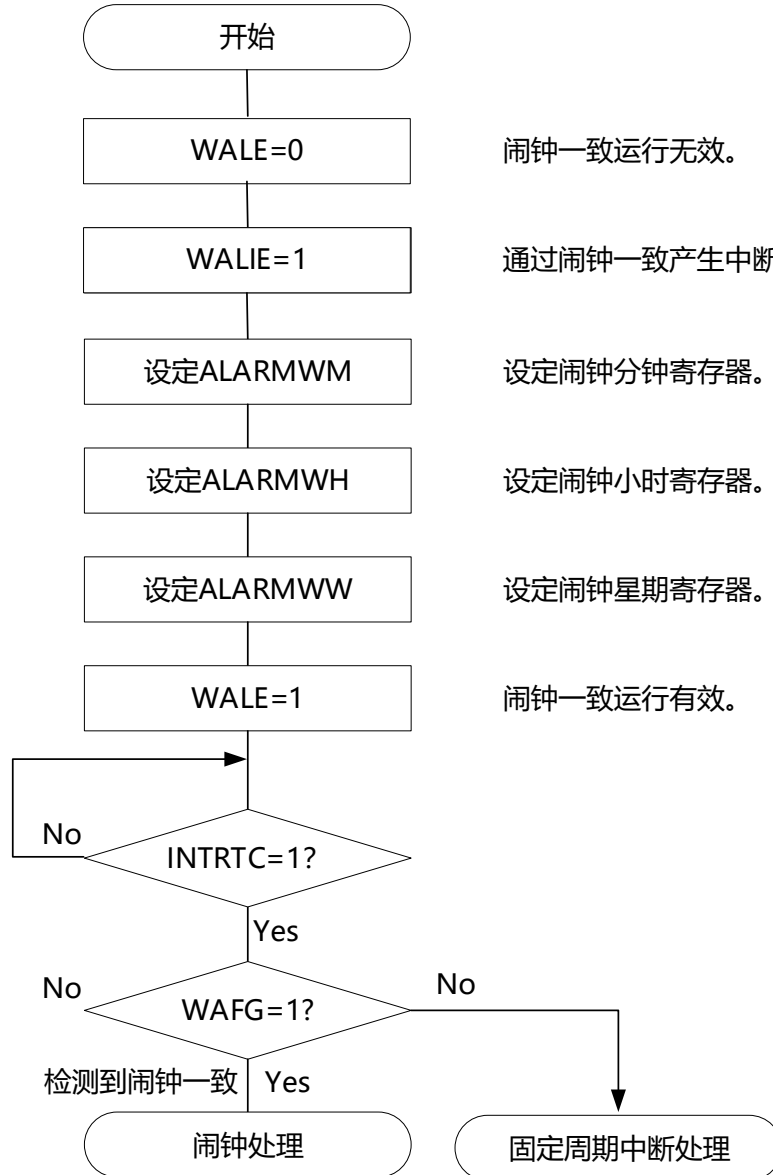
2. 要在计数器运行中（RTCE=1）改写SEC、MIN、HOUR、WEEK、DAY、MONTH、YEAR寄存器时，必须通过中断屏蔽标志寄存器将INTRTC设置为禁止中断处理后进行改写，并且必须在改写后清除WAFG标志、RIFG标志和RTCIF标志。

备注 不限制秒/分钟/小时/星期/日/月/和年计数寄存器/的读操作顺序。可以不读全部寄存器而只读部分寄存器。

6.4.4 实时时钟的闹钟设置

必须先将WALE位置“0”（闹钟运行无效），然后设置闹钟时间。

图6-21 闹钟设置步骤

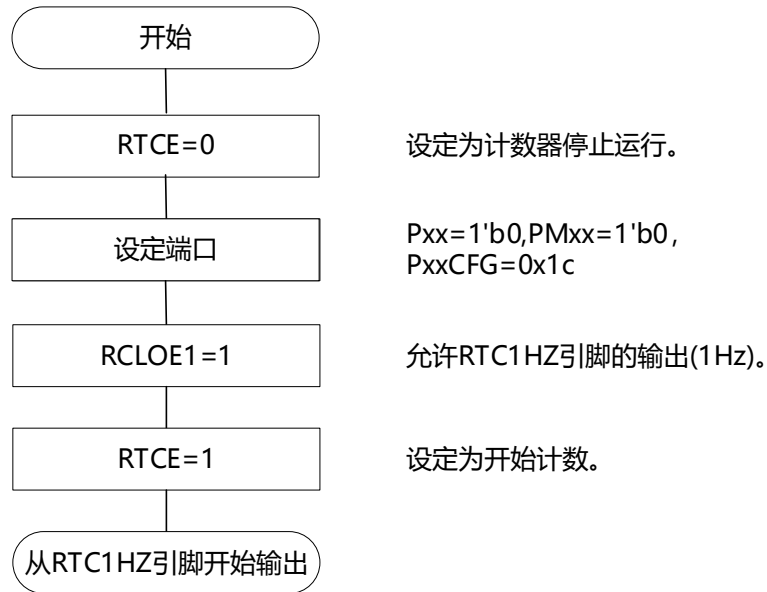


备注1.不限制闹钟分钟寄存器（ALARMWMM）、闹钟小时寄存器（ALARMWH）和闹钟星期寄存器（ALARMWW）的写操作顺序。

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。

6.4.5 实时时钟的1Hz输出

图6-22 1Hz输出的设置步骤



注意1. 必须先在计数时钟 (f_{SUB}) 振荡稳定的状态下将RTCEN位置“1”。

6.4.6 实时时钟的时钟误差校正例子

能通过给时钟误差校正寄存器设置值进行高精度的时钟快慢校正。

校正值的计算方法的例子

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

（DEV=0的情况）

$$\text{校正值注} = 1\text{分钟的校正计数值} / 3 = (\text{振荡频率} / \text{目标频率} - 1) \times 32768 \times 60 / 3$$

（DEV=1的情况）

$$\text{校正值注} = 1\text{分钟的校正计数值} = (\text{振荡频率} / \text{目标频率} - 1) \times 32768 \times 60$$

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= $\{(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) - 1\} \times 2$

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

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

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

备注

1.校正值为2、4、6、8、.....、8186、8188或者-2、-4、-6、-8、.....、-8186、-8188。

2.振荡频率是计数时钟（ f_{RTC} ）的值，能用以下计算式进行计算：

时钟误差校正寄存器为初始值（“00H”）时的RTC1HZ引脚的输出频率 \times 32768

3.目标频率是使用时钟误差校正寄存器进行校正后的频率。

校正例子

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

【振荡频率的测量】

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

注 有关RTC1Hz输出的设置步骤，请参照“6.4.5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）

$$\text{振荡频率} = 32768 \times 0.9999817 \approx 32767.4\text{Hz}$$

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

。

$$\begin{aligned} \text{校正} &= 1 \text{ 分钟的校正计数} = (\text{振荡频率} / \text{目标频率} - 1) \times 32768 \times 60 \\ &= (32767.4 / 32768 - 1) \times 32768 \times 60 \\ &= -36 \end{aligned}$$

【(F12~F0) 的设置值的计算】

（校正=-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)。

$$\begin{aligned} -\{(\text{F11} \sim \text{F0}) - 1\} \times 2 &= -36 \\ (\text{F11} \sim \text{F0}) &= 17 \\ (\text{F11} \sim \text{F0}) &= (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1) \\ (\text{F11} \sim \text{F0}) &= (1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0) \end{aligned}$$

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=-36（SUBCUD寄存器的bit12~0：1,1,1,1,1,1,1,0,1,1,1,0）来设置校正寄存器，就能校正到32768Hz（0ppm）。

第7章 15位间隔定时器

7.1 15位间隔定时器的功能

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

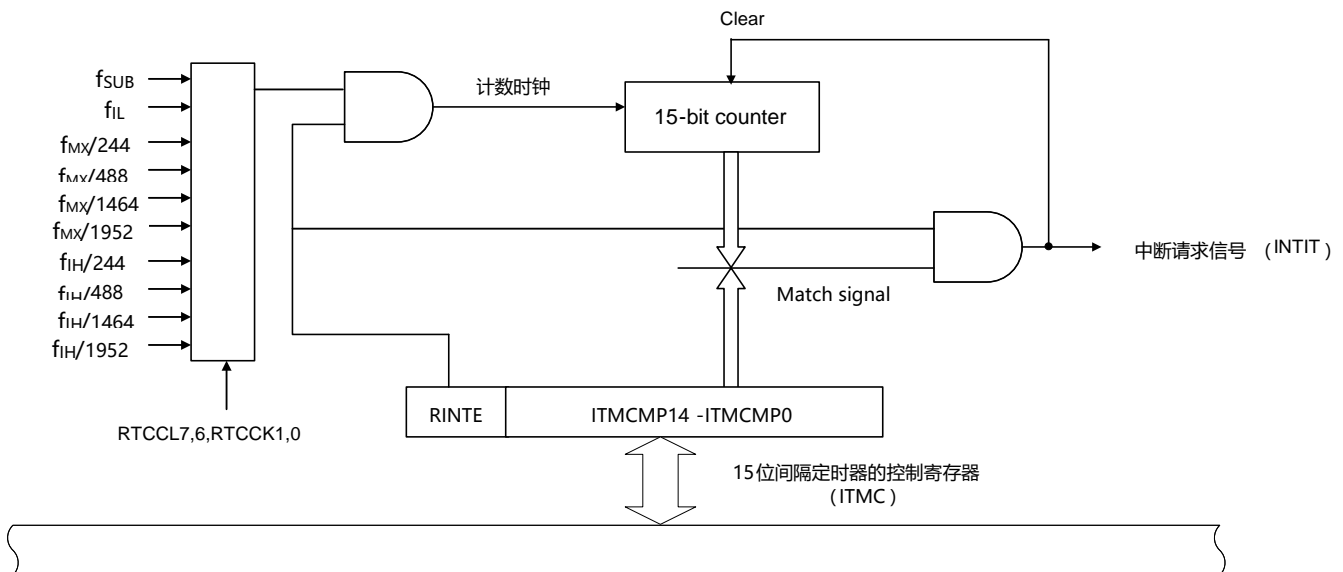
7.2 15位间隔定时器的结构

15位间隔定时器由以下硬件构成。

表7-1 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7-1 15位间隔定时器的框图



7.3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)

7.3.1 外围允许寄存器0 (PER0)

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

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

图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	IRDAEN	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。 <ul style="list-style-type: none"> • 不能写实时时钟 (RTC) 和15位间隔定时器使用的SFR。 • 实时时钟 (RTC) 和15位间隔定时器处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> • 能读写实时时钟 (RTC) 和15位间隔定时器使用的SFR。

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_{RTC})。

图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	0	0	0	0	RTCCKS1	RTCCKS0

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})

RTCCKS1	RTCCKS0	RTCCL6	实时时钟、15位间隔定时器的计数时钟的运行时钟的选择
0	0	0/1	副系统时钟 (f_{SUB})
0	1		低速内部振荡器时钟 (f_{IL}) (必须设置WUTMMCK0为1)
1	0	0	主时钟 f_{max}/f_{HOCO} (通过RTCCL7选择)/1952
1	0	1	主时钟 f_{max}/f_{HOCO} (通过RTCCL7选择)/1464
1	1	0	主时钟 f_{max}/f_{HOCO} (通过RTCCL7选择)/488
1	1	1	主时钟 f_{max}/f_{HOCO} (通过RTCCL7选择)/244

7.3.3 15位间隔定时器的控制寄存器（ITMC）

这是设置15位间隔定时器的运行开始和停止以及比较值的寄存器。

通过16位存储器操作指令设置ITMC寄存器。

在产生复位信号后，此寄存器的值变为“7FFFH”。

图7-4 15位间隔定时器控制寄存器（ITMC）的格式

地址：0x40044F50 复位后：7FFFH R/W

符号	15	14~0
ITMC	RINTE	ITCMP14~ITCMP0

RINTE	15位间隔定时器的运行控制
0	停止计数器的运行（清除计数）。
1	开始计数器的运行。

ITCMP14~ITCMP0	15位间隔定时器比较值的设置
0001H	这些位产生“计数时钟周期（ITCMP设置值+1）”的固定周期中断。
•	
•	
7FFFH	禁止设置。
0000H	
ITCMP14~ITCMP0为“0001H”或者“7FFFH”时的中断周期例子	
•ITCMP14~ITCMP0=0001H，计数时钟： $f_{SUB}=32.768kHz$ $1/32.768[kHz] \times (1+1)=0.06103515625[ms] \approx 61.03[us]$	
• ITCMP14~ITCMP0=7FFFH，计数时钟： $f_{SUB}=32.768kHz$ $1/32.768[kHz] \times (32767+1)=1000[ms]$	

注意1.要将RINTE位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将INTIT设置为禁止中断处理后进行改写。要重新开始运行（从“0”改为“1”）时，必须在清除ITIF标志后设置为允许中断处理。

2.RINTE位的读取值在设置RINTE位后的1个计数时钟之后被反映。

3.在从睡眠模式转移到通常运行模式后，如果要设置ITMC寄存器并且再次转移到睡眠模式，就必须在确认ITMC寄存器的写入值被反映后或者在设置ITMC寄存器后至少经过1个计数时钟之后再转移到睡眠模式。

4.要更改ITCMP14~ITCMP0位的设置时，必须在RINTE位为“0”的状态下进行。

但是，能在将RINTE位从“0”改为“1”或者从“1”改为“0”的同时更改ITCMP14~ITCMP0位的设置。

7.4 15位间隔定时器的运行

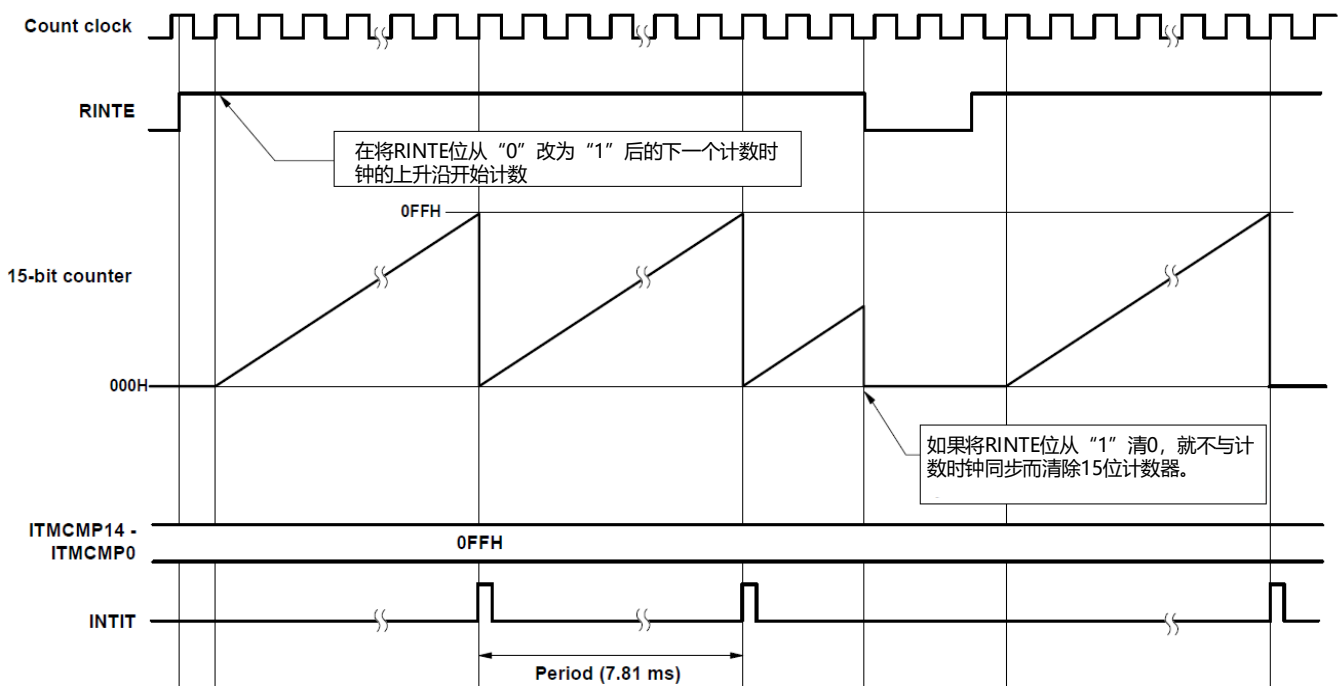
7.4.1 15位间隔定时器的运行时序

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

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

15位间隔定时器的基本运行如图7-5所示。

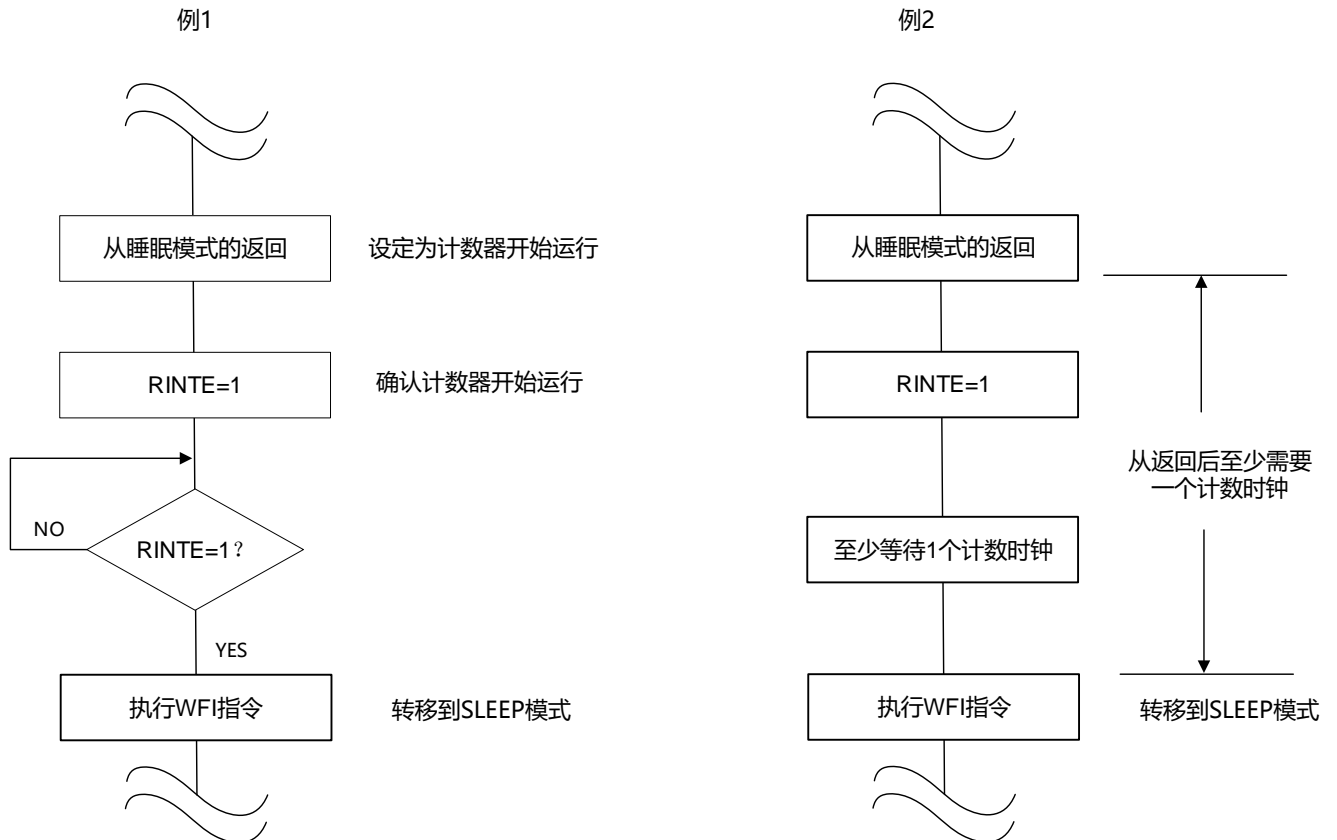
图7-5 15位间隔定时器的运行时序
(ITCMP14~ITCMP0=0FFH, 计数时钟: $f_{SUB}=32.768kHz$)



7.4.2 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8章 时钟输出/蜂鸣器输出控制电路

注意 本章的下述内容主要针对24引脚产品进行说明。

8.1 时钟输出/蜂鸣器输出控制电路的功能

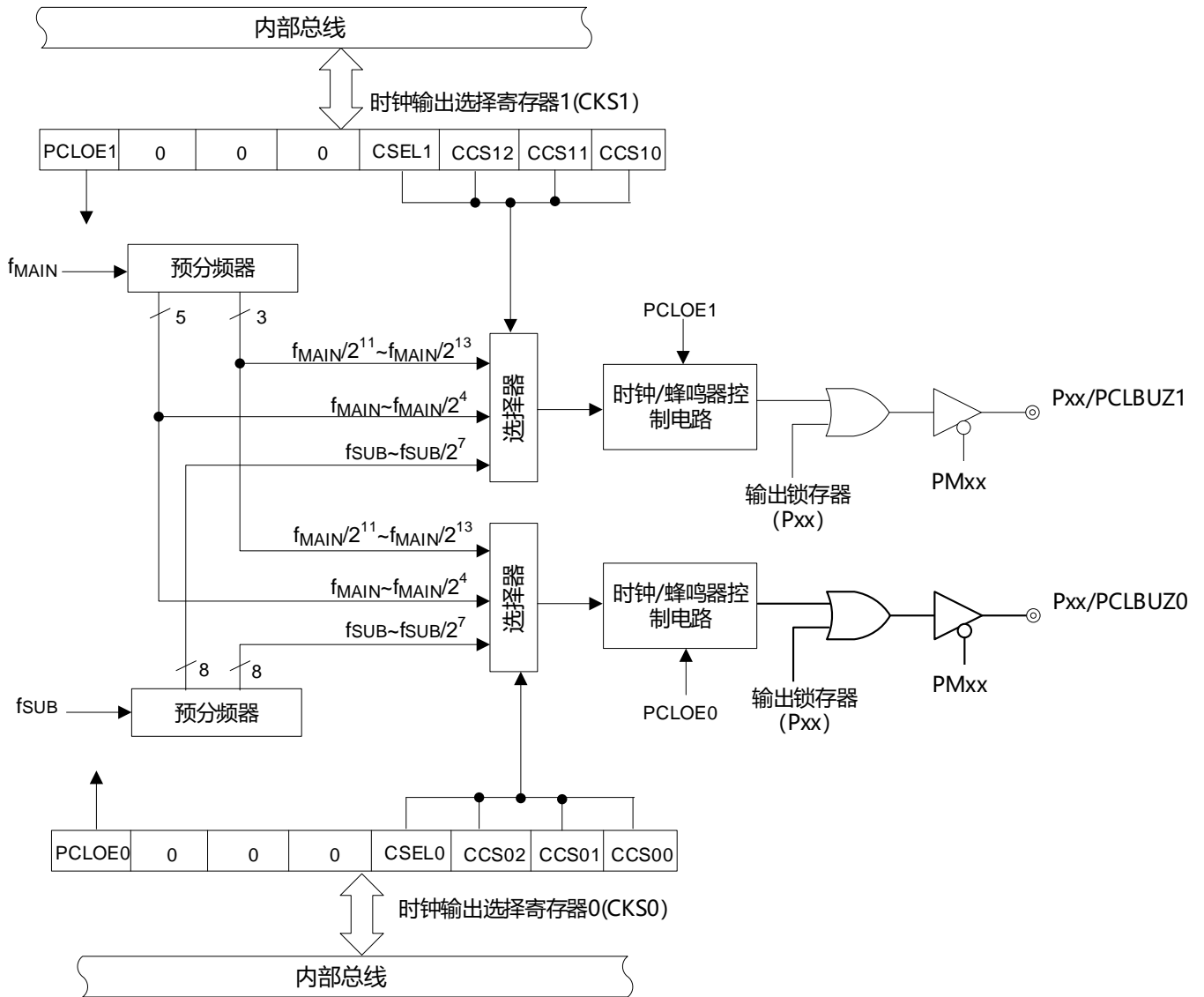
时钟输出是输出提供给外围IC时钟的功能，蜂鸣器输出是输出蜂鸣器频率方波的功能。本产品能选择除P00以外的任意引脚用作时钟输出或者蜂鸣器输出。PCLBUZn引脚输出由时钟输出选择寄存器n (CKSn) 选择的时钟。

时钟输出/蜂鸣器输出控制电路的框图如图8-1所示。

注意 在副系统时钟提供模式控制寄存器 (OSMC) 的RTCLPC位为“1”时并且在CPU以副系统时钟 (f_{SUB}) 运行的SLEEP模式中，不能从PCLBUZn引脚输出副系统时钟 (f_{SUB})。

备注 n=0、1

图8-1 时钟输出/蜂鸣器输出控制电路的框图



注 有关能从PCLBUZ0引脚和PCLBUZ1引脚输出的频率，请参照数据手册的“AC特性”。

8.2 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8-1 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)

8.3 控制时钟输出/蜂鸣器输出控制电路的寄存器

8.3.1 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 (PCLBUZn) 的输出以及设置输出时钟的寄存器。通过CKSn寄存器选择PCLBUZn引脚输出的时钟。通过8位存储器操作指令设置CKSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图8-2 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	PCLBUZn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	PCLBUZn引脚输出时钟的选择
0	0	0	0	f_{MAIN}
0	0	0	1	$f_{MAIN}/2$
0	0	1	0	$f_{MAIN}/2^2$
0	0	1	1	$f_{MAIN}/2^3$
0	1	0	0	$f_{MAIN}/2^4$
0	1	0	1	$f_{MAIN}/2^{11}$
0	1	1	0	$f_{MAIN}/2^{12}$
0	1	1	1	$f_{MAIN}/2^{13}$
1	0	0	0	f_{SUB}
1	0	0	1	$f_{SUB}/2$
1	0	1	0	$f_{SUB}/2^2$
1	0	1	1	$f_{SUB}/2^3$
1	1	0	0	$f_{SUB}/2^4$
1	1	0	1	$f_{SUB}/2^5$
1	1	1	0	$f_{SUB}/2^6$
1	1	1	1	$f_{SUB}/2^7$

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

- 备注
- 1.n=0、1
 2. f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率

8.3.2 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/蜂鸣器输出功能复用到除P00以外的任意端口。必须设置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）。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（PCLBUZ0）的情况：

将端口模式寄存器2的PM20位置“0”。

将端口模式寄存器2的PMC20位置“0”。

将端口复用功能配置寄存器P20CFG置“0x18”。

8.4 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。

PCLBUZ0引脚输出由时钟输出选择寄存器0（CKS0）选择的时钟/蜂鸣器。

PCLBUZ1引脚输出由时钟输出选择寄存器1（CKS1）选择的时钟/蜂鸣器。

8.4.1 输出引脚的运行

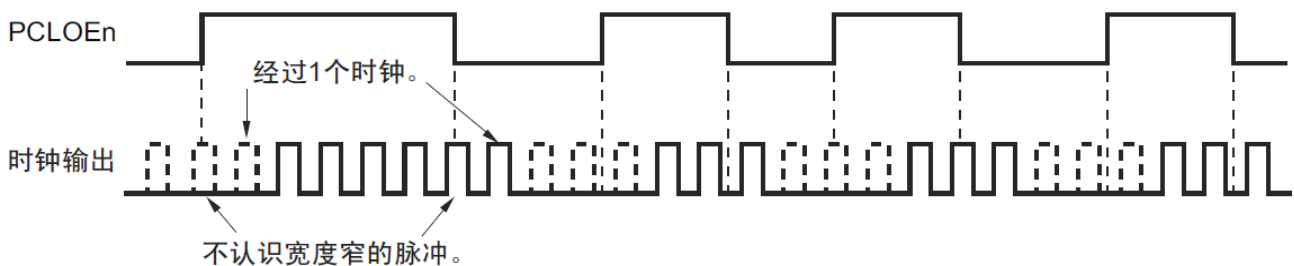
PCLBUZn引脚按照以下步骤进行输出：

- ① 设置端口复用功能配置寄存器（PxxCFG），将用作PCLBUZ0引脚的端口对应的端口模式寄存器（PMxx）和端口模式控制寄存器（PMCxx）的位置“0”。
- ② 通过PCLBUZn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8-3所示。

2.n=0、1

图8-3 PCLBUZn引脚的时钟输出时序



8.5 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PCLBUZn输出（CSELn=0）时，如果在设置停止输出（PCLOEn=0）后的1.5个PCLBUZn引脚的输出时钟内转移到深度睡眠模式，PCLBUZn的输出宽度就变窄。

第9章 看门狗定时器

9.1 看门狗定时器的功能

看门狗定时器通过选项字节（000C0H）设置计数运行。看门狗定时器以低速内部振荡器时钟（ f_{IL} ）运行。看门狗定时器用于检测程序失控。在检测到程序失控时，产生内部复位信号。

下述情况判断为程序失控。

- 当看门狗定时器的计数器发生上溢时
- 当对看门狗定时器的允许寄存器（WDTE）执行位操作指令时
- 当给WDTE寄存器写“ACH”以外的数据时
- 在窗口关闭期间给WDTE寄存器写数据时

当因看门狗定时器而发生复位时，将复位控制标志寄存器（RESF）的bit4（WDTRF）置“1”。有关RESF寄存器的详细内容，请参照“第20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。

9.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表9-1 看门狗定时器的结构

项目	结构
计数器	内部计数器（17位）
控制寄存器	看门狗定时器的允许寄存器（WDTE）

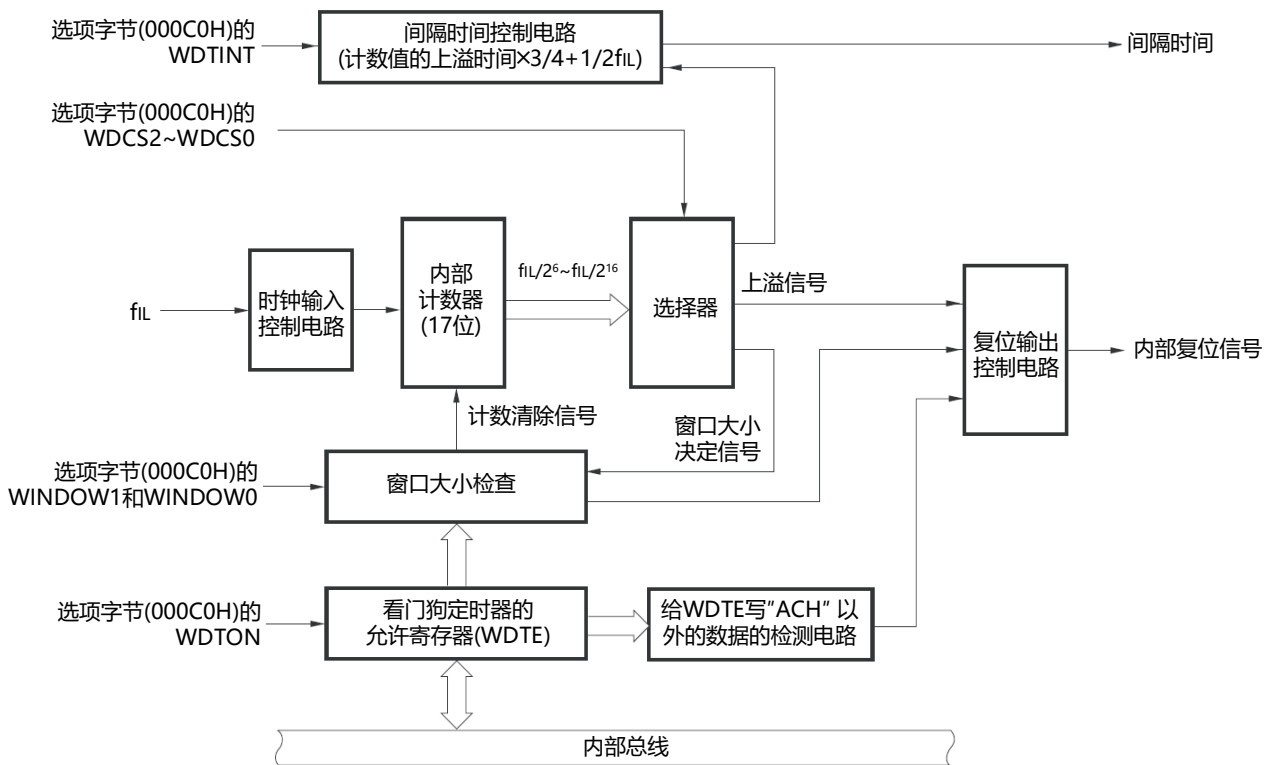
通过选项字节控制计数器的运行以及设置上溢时间、窗口打开期间和间隔中断。

表9-2 选项字节和看门狗定时器的设置内容

看门狗定时器的设置内容	选项字节（000C0H）
看门狗定时器的间隔中断的设置	bit7（WDTINT）
窗口打开期间的设置	bit6和bit5（WINDOW1、WINDOW0）
看门狗定时器的计数器运行控制	bit4（WDTON）
看门狗定时器的上溢时间的设置	bit3~1（WDCS2~WDCS0）
看门狗定时器的计数器运行控制（睡眠时）	bit0（WDSTBYON）

备注 有关选项字节，请参照“第24章温度传感器”

图9-1 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

9.3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。

9.3.1 看门狗定时器的允许寄存器（WDTE）

通过给WDTE寄存器写“ACH”，清除看门狗定时器的计数器并且重新开始计数。通过8位存储器操作指令设置WDTE寄存器。在产生复位信号后，此寄存器的值变为“9AH”或者“1AH”注。

图9-2 看门狗定时器的允许寄存器（WDTE）的格式



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

WDTON位的设置值	WDTE寄存器的复位值
0（禁止看门狗定时器的计数运行）	1AH
1（允许看门狗定时器的计数运行）	9AH

- 注意1.当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。
 2.当对WDTE寄存器执行位操作指令时，产生内部复位信号。
 3.WDTE寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

9.4看门狗定时器的运行

9.4.1 看门狗定时器的运行控制

1. 当使用看门狗定时器时，通过选项字节（000C0H）设置以下内容：
 - 必须将选项字节（000C0H）的bit4（WDTON）置“1”，允许看门狗定时器的计数运行（在解除复位后，计数器开始运行）（详细内容请参照第24章温度传感器）。
2. 在解除复位后，看门狗定时器开始计数。
3. 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
4. 此后，解除复位后第2次以后的WDTE寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写WDTE寄存器，就产生内部复位信号。
5. 如果不给WDTE寄存器写“ACH”而超过上溢时间，就产生内部复位信号。以下情况会产生内部复位信号：
 - 当对WDTE寄存器执行位操作指令时
 - 当给WDTE寄存器写“ACH”以外的数据时

注意1.只在解除复位后第1次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任意时候写WDTE，就清除看门狗定时器并且重新开始计数。

2.从给WDTE寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大2个 f_{IL} 时钟的误差。

3.在计数值发生上溢前，都能清除看门狗定时器。

4.如下所示，看门狗定时器在睡眠或者深度睡眠模式中的运行因选项字节（000C0H）的bit0（WDSTBYON）的设置值而不同。

	WDSTBYON=0	WDSTBYON=1
睡眠模式	停止看门狗定时器运行。	继续看门狗定时器运行。
深度睡眠模式		

当WDSTBYON位为“0”时，在解除睡眠或者深度睡眠模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。

当解除深度睡眠模式后以X1振荡时钟运行时，CPU在经过振荡稳定时间后开始运行。

如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除深度睡眠模式后，如果要以X1振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设置。

9.4.2 看门狗定时器上溢时间的设置

通过选项字节（000C0H）的bit3~1（WDCS2~WDCS0）设置看门狗定时器的上溢时间。

在发生上溢时，产生内部复位信号。如果在上溢时间前的窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除计数并且重新开始计数。能设置的上溢时间如下所示。

表9-3 看门狗定时器上溢时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=40\text{kHz}(\text{MAX.})$ 的情况)
0	0	0	$2^6/f_{IL}$ (1.6ms)
0	0	1	$2^7/f_{IL}$ (3.2ms)
0	1	0	$2^8/f_{IL}$ (6.4ms)
0	1	1	$2^9/f_{IL}$ (12.8ms)
1	0	0	$2^{11}/f_{IL}$ (51.2ms)
1	0	1	$2^{13}/f_{IL}$ (204.8ms)
1	1	0	$2^{14}/f_{IL}$ (409.6ms)
1	1	1	$2^{16}/f_{IL}$ (1638.4ms)

备注 f_{IL} ：低速内部振荡器的时钟频率

9.4.3 看门狗定时器窗口打开期间的设置

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

- 如果在窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
- 在窗口关闭期间，即使给WDTE寄存器写“ACH”，也会检测到异常并且产生内部复位信号。

注意 只在解除复位后第1次写WDTE寄存器时，与窗口打开期间无关，只要在上溢时间前的任意时候写WDTE，就清除看门狗定时器并且重新开始计数。

能设置的窗口打开期间如下所示。

表9-4 看门狗定时器窗口打开期间的设置

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

注意 当选项字节（000C0H）的bit0（WDSTBYON）为“0”时，与WINDOW1位和WINDOW0位的值无关，窗口打开期间为100%。

备注 当将上溢时间设置为 $2^9/f_{IL}$ 的情况时，窗口关闭时间和打开时间如下所示。

	窗口打开期间的设置	
	75%	100%
窗口关闭时间	0~6.4ms	无
窗口打开时间	6.4~12.8ms	0~12.8ms

<当窗口打开期间为75%时>

- 上溢时间： $2^9/f_{IL}(\text{MAX.})=2^9/40\text{kHz}(\text{MAX.})=12.8\text{ms}$
- 窗口关闭时间： $0\sim 2^9/f_{IL}(\text{MIN.}) \times (1-0.75)=0\sim 2^9/20\text{kHz} \times 0.25=0\sim 6.4\text{ms}$
- 窗口打开时间： $2^9/f_{IL}(\text{MIN.}) \times (1-0.75)\sim 2^9/f_{IL}(\text{MAX.})=6.4\sim 12.8\text{ms}$

9.4.4 看门狗定时器间隔中断的设置

能通过设置选项字节（000C0H）的bit7（WDTINT），在达到上溢时间的75%+1/2f_{IL}时产生间隔中断（INTWDTI）。

表9-5 看门狗定时器间隔中断的设置

WDTINT	看门狗定时器间隔中断的使用/不使用
0	不使用间隔中断。
1	在达到上溢时间的75%+1/2f _{IL} 时，产生间隔中断。

注意 当解除深度睡眠模式后以X1振荡时钟运行时，CPU在经过振荡稳定时间后开始运行。如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除深度睡眠模式后，如果要以X1振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设置。

备注 即使在产生INTWDTI后也继续计数（继续到给看门狗定时器的允许寄存器（WDTE）写“ACH”为止）。如果在上溢时间前不给WDTE寄存器写“ACH”，就产生内部复位信号。

第10章 A/D转换器

A/D转换器的模拟输入通道数因产品而不同。

引脚数	20引脚	24引脚
模拟输入通道	11ch	15ch
	(ANI0,ANI1,ANI4~ANI9,ANI12~ANI14)	(ANI0~ANI14)

10.1 A/D转换器的功能

A/D转换器是将模拟输入转换为数字值的转换器，最多能控制18个模拟通道的A/D转换（15个引脚输入通道和3个内部通道）。

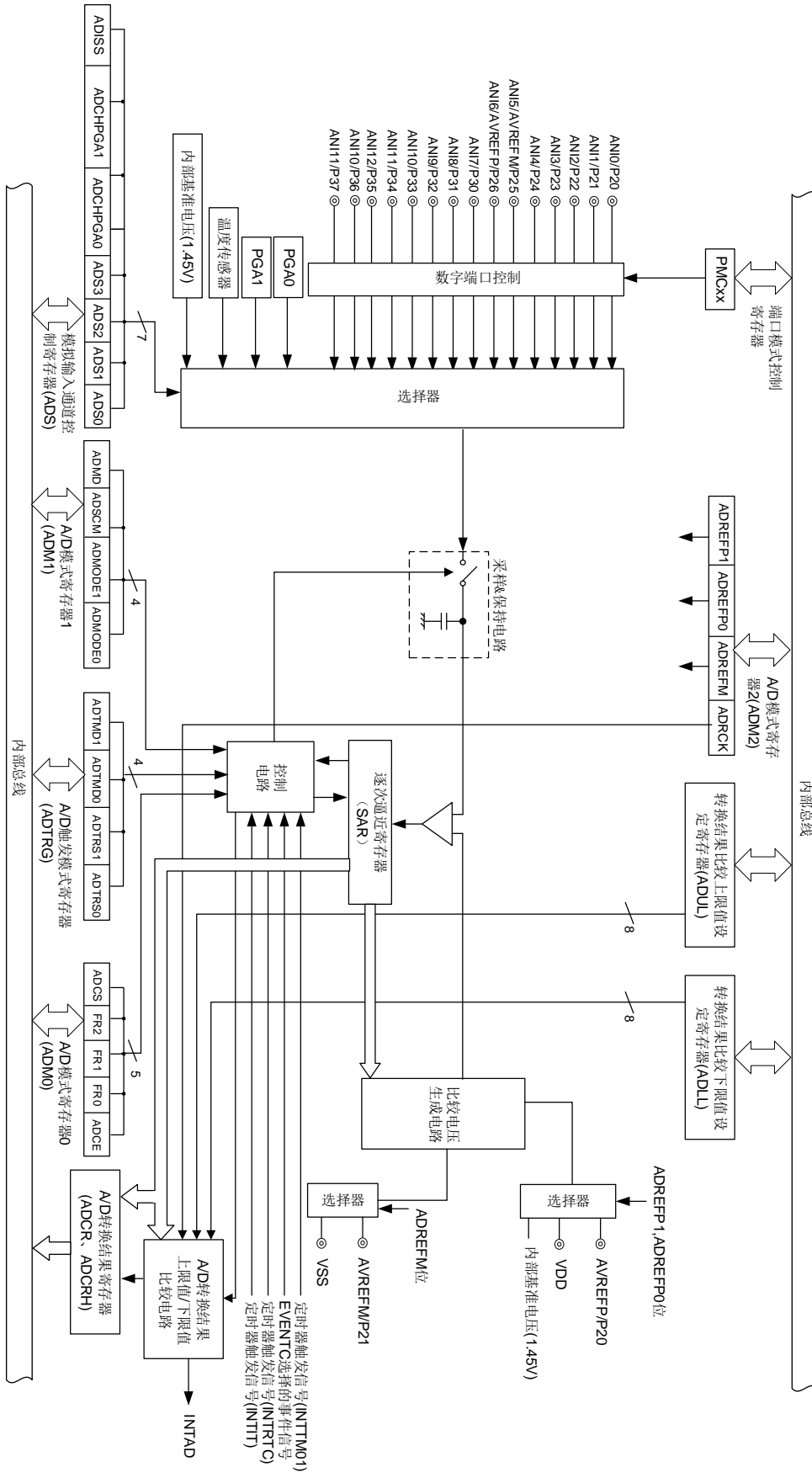
A/D转换器有以下功能。

- 12位分辨率的A/D转换
从ANI0~ANI14, PGA0, PGA1和温度传感器中选择1个通道的模拟输入，重复进行12位分辨率的A/D转换。每结束1次A/D转换，就产生中断请求（INTAD）（选择模式的情况）。

能通过下述的模式组合设置各种A/D转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断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~ANI14中连续的4个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行1次A/D转换。
	连续转换模式	对所选通道进行连续的A/D转换，直到被软件停止为止。
采样时间	采样时钟5.5~255个ADCLK	采样时间可以通过ADNSMP寄存器来选择，默认使用13.5个转换时钟（ f_{AD} ）。

图 10-1 A/D转换器的框图



10.2 控制A/D转换器的寄存器

控制A/D转换器的寄存器如下所示：

寄存器基地址：

CSC_BASE =4002_0420H;

ADC_BASE =4004_5000H;

PORT_BASE =4004_0000H

寄存器名称	寄存器描述	R/W	复位值	寄存器地址
PER0	外围使能寄存器0	R/W	00H	CSC_BASE+20H
ADM0	A/D转换器的模式寄存器0	R/W	00H	ADC_BASE+00H
ADM1	A/D转换器的模式寄存器1	R/W	00H	ADC_BASE+02H
ADM2	A/D转换器的模式寄存器2	R/W	00H	ADC_BASE+04H
ADTRG	A/D转换器的触发模式寄存器	R/W	00H	ADC_BASE+06H
ADS	模拟输入通道指定寄存器	R/W	00H	ADC_BASE+08H
ADLL	转换结果比较下限值设置寄存器	R/W	00H	ADC_BASE+0AH
ADUL	转换结果比较上限值设置寄存器	R/W	00H	ADC_BASE+0BH
ADNSMP	A/D转换器的采样时间控制寄存器	R/W	0dH	ADC_BASE+0CH
ADCR	12位A/D转换结果寄存器	R	0000H	ADC_BASE+0EH
ADCRH	8位A/D转换结果寄存器	R	00H	ADC_BASE+0FH
ADTES	A/D测试寄存器	R/W	00H	ADC_BASE+10H
ADNDIS	A/D转换器的充放电控制寄存器	R/W	00H	ADC_BASE+11H
ADSMPWAIT	A/D转换器的采样时间延长控制寄存器	R/W	00H	ADC_BASE+15H
ADFLG	A/D硬模块状态寄存器	R	00H	ADC_BASE+16H
PMC2	端口模式控制寄存器2	R/W	FCH	PORT_BASE+62H
PMC3	端口模式控制寄存器3	R/W	1FH	PORT_BASE+63H

10.2.1 外围允许寄存器0 (PER0)

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

要使用A/D转换器时，必须将bit5 (ADCEN) 置“1”。

通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图10-2 外围允许寄存器0 (PER0) 的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
PER0	RTCEN	IRDAEN	ADCEN	IICA0EN	SCI1EN	SCI0EN	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。 •不能写A/D转换器使用的SFR。 •A/D转换器处于复位状态。
1	提供输入时钟。 •能读写A/D转换器使用的SFR。

注意1.要设置A/D转换器时，必须先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）。

- A/D转换器的模式寄存器0 (ADM0)
- A/D转换器的模式寄存器1 (ADM1)
- A/D转换器的模式寄存器2 (ADM2)
- A/D转换器的触发模式寄存器 (ADTRG)
- 模拟输入通道指定寄存器 (ADS)
- 转换结果比较下限值设置寄存器 (ADLL)
- 转换结果比较上限值设置寄存器 (ADUL)
- A/D采样时间控制寄存器 (ADNSMP)
- 12位A/D转换结果寄存器 (ADCR)
- 8位A/D转换结果寄存器 (ADCRH)
- A/D测试寄存器 (ADTES)
- A/D充放电控制寄存器 (ADNDIS)
- A/D采样时间延长控制寄存器 (ADSMPWAIT)
- A/D硬模块状态寄存器 (ADFLG)

10.2.2 A/D转换器的模式寄存器0 (ADM0)

用于设置A/D转换时钟、转换开始或者停止的寄存器。通过8位存储器操作指令设置ADM0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图10-3 A/D转换器的模式寄存器0 (ADM0) 的格式

复位值：00H R/W

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

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

ADCE	A/D电压比较器的运行控制 ^{注2}
0	停止A/D电压比较器的运行。
1	允许A/D电压比较器的运行。

- 注
- 1.有关FR2~FR0位和A/D转换的详细内容，请参照“表10-3A/D转换时间的选择(1/2)”。
 - 2.A/D转换器开始运行需要1us稳定时间。在软件触发模式或者硬件触发无等待模式中，在将ADCE位置“1”后至少经过1us时间，然后将ADCS位置“1”，则本次转换结果有效。如果等待时间小于1us而将ADCS位置“1”，就必须忽视本次转换结果。在硬件触发等待模式中，由设计保证1us的等待时间。

注意

- 1.必须在转换停止状态ADCS=0下更改FR2~FR0位。
- 2.禁止进行ADCS=1、ADCE=0的设置。
- 3.禁止通过8位操作指令将ADCS=0、ADCE=0的状态设置为ADCS=1、ADCE=1。必须按照“10.5A/D转换器的设置流程图”进行设置。

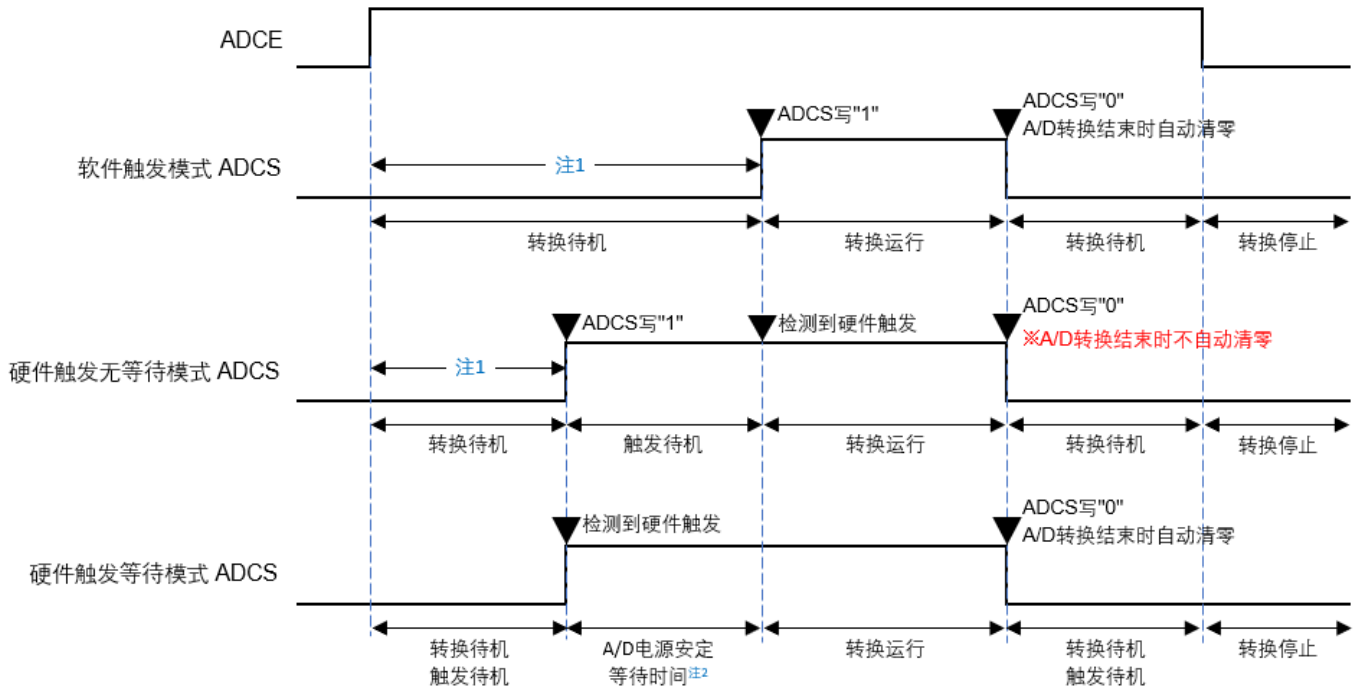
表10-1 ADCS位和ADCE位的设置

ADCS	ADCE	A/D转换运行
0	0	转换停止状态
0	1	转换待机状态
1	0	禁止设置。
1	1	转换运行状态

表10-2 ADCS位的置位和清除条件

A/D转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给ADCS位写“1”时	当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时 •在A/D转换结束时自动清“0”。
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时 •当设置的4个通道转换结束时，自动清“0”。
硬件触发无等待模式	选择模式	连续转换模式	当给ADCS位写“1”时	当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时
硬件触发等待模式	选择模式	连续转换模式	当输入硬件触发时	当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时 •在A/D转换结束时自动清“0”。
	扫描模式	连续转换模式		当给ADCS位写“0”时
		单次转换模式		•当给ADCS位写“0”时 •当设置的4个通道转换结束时，自动清“0”。

图10-4 使用A/D各种模式时的动作状态图



- 注
- 1.在软件触发模式或者硬件触发无等待模式中，为了稳定内部电路，从ADCE位上升到ADCS位上升的时间至少需要1us (TBD)。
 - 2.在硬件触发等待模式，A/D电源稳定时间1us由设计保证。

注意1.要使用硬件触发等待模式时，禁止将ADCS位置“1”（当检测到硬件触发信号时，自动切换为“1”）。但是，为了设置为A/D转换待机状态，能将ADCS位置“0”。

2.必须在ADCS位为“0”（停止转换/转换待机状态）时改写ADCE位。

3.为了结束A/D转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2个 f_{CLK} 时钟+A/D转换时间

硬件触发等待模式时：2个 f_{CLK} 时钟+A/D电源稳定等待时间+A/D转换时间

备注 f_{CLK} ：CPU/外围硬件的时钟频率

表10-3 A/D转换时间的选择(1/2)

(1)无A/D电源稳定等待时间 (软件触发模式/硬件触发无等待模式)

A/D转换器的模式寄存器0 (ADM0)			A/D转换器的模式寄存器1 (ADM1) ^{注1}		模式	转换时钟ADCLK的频率 (f_{AD})	12位分辨率的转换时间 ^{注2}	
FR2	FR1	FR0	ADM0[1]	ADM0[0]			ADC转换时间 = (采样时钟数 + 逐次比较时钟数) / f_{AD}	ADC转换时间 (以13.5个采样时钟为例)
0	0	0	0	0	高速变换模式	$f_{CLK}/32$	45个ADCLK (采样时钟数13.5个+逐次比较时钟数31.5个)	$45/f_{AD}$
0	0	1				$f_{CLK}/16$		
0	1	0				$f_{CLK}/8$		
0	1	1				$f_{CLK}/4$		
1	0	0				$f_{CLK}/2$		
1	0	1				$f_{CLK}/1$		
0	0	0	1	1	低电流模式	$f_{CLK}/32$	54个ADCLK (采样时钟数13.5个+逐次比较时钟数40.5个)	$54/f_{AD}$
0	0	1				$f_{CLK}/16$		
0	1	0				$f_{CLK}/8$		
0	1	1				$f_{CLK}/4$		
1	0	0				$f_{CLK}/2$		
1	0	1				$f_{CLK}/1$		

注1.要将FR2~FR0位、ADM0[1:0]位改写为不同数据时,必须在转换停止状态(ADCS=0)下进行。

注2.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_{AD}

其中采样时钟个数可通过ADNSMP寄存器进行调整,默认为13.5个ADCLK。逐次比较时钟个数由转换模式决定,高速变换模式时为31.5个ADCLK,低电流模式时为40.5个ADCLK。高速变换模式时ADCLK支持的最快时钟为64MHz,低电流模式时ADCLK支持的最快时钟为27MHz。实际使用时,请按照数据手册中“AC特性”要求配置转换模式和转换时钟频率。

备注 f_{CLK} : CPU/外围硬件的时钟频率

表10-4 A/D转换时间的选择(2/2)

(2)有A/D电源稳定等待时间（硬件触发等待模式注1）

A/D转换器的模式寄存器0 (ADM0)			A/D转换器的模式寄存器1 (ADM1)		模式	转换时钟ADCLK的频率 (f_{AD})	A/D电源稳定等待时间	ADC转换时钟数	A/D电源稳定等待时间+ADC转换时间 ^{注2}
FR2	FR1	FR0	ADM0[1]	ADM0[0]					
0	0	0	0	0	高速变换模式	$f_{CLK}/32$	1us	45个ADCLK (采样时钟数13.5个+逐次比较时钟数31.5个)	$1us + 45/f_{AD}$
0	0	1				$f_{CLK}/16$			
0	1	0				$f_{CLK}/8$			
0	1	1				$f_{CLK}/4$			
1	0	0				$f_{CLK}/2$			
1	0	1				$f_{CLK}/1$			
0	0	0	1	1	低电流模式	$f_{CLK}/32$	1us	54个ADCLK (采样时钟数13.5个+逐次比较时钟数40.5个)	$1us + 54/f_{AD}$
0	0	1				$f_{CLK}/16$			
0	1	0				$f_{CLK}/8$			
0	1	1				$f_{CLK}/4$			
1	0	0				$f_{CLK}/2$			
1	0	1				$f_{CLK}/1$			

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。在连续转换模式时，仅在第1次检测到硬件触发后，发生A/D电源稳定等待时间（参照表10-4）。

注2.硬件触发后ADC转换需要的时间= $1us + (\text{采样时钟数} + \text{逐次比较时钟数}) / f_{AD}$

其中采样时钟个数可通过ADNSMP寄存器进行调整，默认为13.5个ADCLK。逐次比较时钟个数由转换模式决定，高速变换模式时为31.5个ADCLK，低电流模式时为40.5个ADCLK。高速变换模式时ADCLK支持的最快时钟为64MHz，低电流模式时ADCLK支持的最快时钟为27MHz。实际使用时，请按照数据手册中“AC特性”要求配置转换模式和转换时钟频率。

注意1.要将FR2~FR0位、ADM0[1:0]位改写为不同数据时，必须在转换停止状态（ADCS=0）下进行。

2.硬件触发等待模式中的转换时间包含检测到硬件触发后的A/D电源稳定等待时间。

备注 f_{CLK} : CPU/外围硬件的时钟频率

10.2.3 A/D转换器的模式寄存器1 (ADM1)

这是设置A/D转换模式的寄存器。
 通过8位存储器操作指令设置ADM1寄存器。
 在产生复位信号后，此寄存器的值变为“00H”。

图10-5 A/D转换器的模式寄存器1 (ADM1) 的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADM1	ADMD	0	0	0	ADSCM	0	ADMODE1	ADMODE0

ADMD	A/D转换通道选择模式的设置
0	选择模式
1	扫描模式

ADSCM	A/D转换模式的设置
0	连续转换模式
1	单次转换模式

ADMODE1	ADMODE0	A/D转换模式
0	0	高速变换模式 (ADCLK最快时钟为64MHz)
1	1	低电流模式 (ADCLK最快时钟为27MHz)
其他		设置禁止

注意 必须将bit6~4, 2置“0”。

注意1.要改写ADM1寄存器时，必须在转换停止状态 (ADCS=0) 下进行。

2.为了正常结束A/D转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2个 f_{CLK} 时钟+A/D转换时间

硬件触发等待模式时：2个 f_{CLK} 时钟+A/D电源稳定等待时间+A/D转换时间

3.A/D转换时，逐次比较时钟个数由转换模式决定，高速变换模式时为31.5个ADCLK，低电流模式时为40.5个ADCLK。高速变换模式时ADCLK支持的最快时钟为64MHz，低电流模式时ADCLK支持的最快时钟为27MHz。实际使用时，请按照数据手册中“AC特性”要求配置转换模式和转换时钟频率。

备注1. f_{CLK} ：CPU/外围硬件的时钟频率

10.2.4 A/D转换器的模式寄存器2 (ADM2)

通过8位存储器操作指令设置ADM2寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图10-6 A/D转换器的模式寄存器2 (ADM2) 的格式(1/3)

复位值: 00H R/W

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

ADREFP1	ADREFP0	A/D转换器的正 (+) 基准电压源的选择
0	0	由V _{DD} 提供。
0	1	由AV _{REFP} 外部端子提供。
1	0	由内部基准电压提供 (1.45V)。
1	1	设置禁止

ADREFM	A/D转换器的负 (-) 基准电压源的选择
0	由V _{SS} 提供。
1	由AV _{REFM} 外部端子提供。

ADRCK	转换结果上限值和下限值的检查
0	当ADLL寄存器 ≤ ADCR寄存器 ≤ ADUL寄存器 (AREA1) 时, 产生中断信号 (INTAD)。
1	当ADCR寄存器 < ADLL寄存器 (AREA2) 或者ADUL寄存器 < ADCR寄存器 (AREA3) 时, 产生中断信号 (INTAD)。
AREA1~AREA3的中断信号 (INTAD) 的产生范围如图15-8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

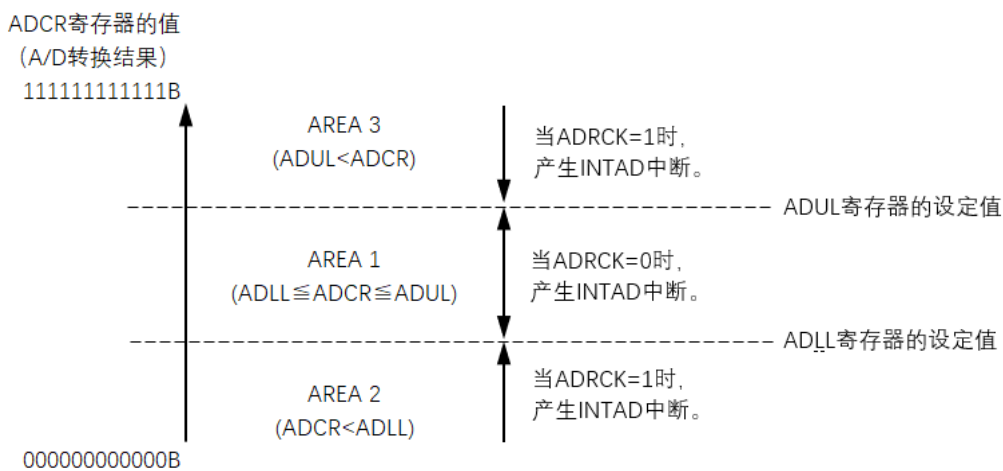


图 10-7 ADRCK位的中断信号产生范围

- 注意
- 1.要改写ADM2寄存器时, 必须在转换停止状态 (ADCS=0) 下进行。
 - 2.当使用AV_{REFP}和AV_{REFM}时, 必须将该端口设置为模拟端口模式 (PMCxx=1)。

备注 当不发生INTAD时, A/D转换结果不保存到ADCR寄存器和ADCRH寄存器。

10.2.5 A/D转换器的触发模式寄存器（ADTRG）

这是设置A/D转换触发模式和硬件触发信号的寄存器。
 通过8位存储器操作指令设置ADTRG寄存器。
 在产生复位信号后，此寄存器的值变为“00H”。

图10-8 A/D转换器的触发模式寄存器（ADTRG）的格式

复位值：00H R/W

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

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

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

注意1.要改写ADTRG寄存器时，必须在转换停止状态（ADCS=0，ADCE=0）下进行。

2.为了正常结束A/D转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2个 f_{CLK} 时钟+A/D转换时间

硬件触发等待模式时：2个 f_{CLK} 时钟+A/D电源稳定等待时间+A/D转换时间

备注1. f_{CLK} ：CPU/外围硬件的时钟频率

10.2.6 模拟输入通道指定寄存器 (ADS)

这是指定要进行A/D转换的模拟电压输入通道的寄存器。

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

图10-9 模拟输入通道指定寄存器 (ADS) 的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADS	ADISS	0	ADCHPGA1	ADCHPGA0	ADS3	ADS2	ADS1	ADS0

○选择模式 (ADM1.ADMD=0)

ADS 寄存器设置值							CH 选择
ADISS	ADCHPGA1	ADCHPGA0	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	ANI0(P20)
0	0	0	0	0	0	1	ANI1(P21)
0	0	0	0	0	1	0	ANI2(P22)
0	0	0	0	0	1	1	ANI3(P23)
0	0	0	0	1	0	0	ANI4(P24)
0	0	0	0	1	0	1	ANI5(P25)
0	0	0	0	1	1	0	ANI6(P26)
0	0	0	0	1	1	1	ANI7(P30)
0	0	0	1	0	0	0	ANI8(P31)
0	0	0	1	0	0	1	ANI9(P32)
0	0	0	1	0	1	0	ANI10(P33)
0	0	0	1	0	1	1	ANI11(P34)
0	0	0	1	1	0	0	ANI12(P35)
0	0	0	1	1	0	1	ANI13(P36)
0	0	0	1	1	1	0	ANI14(P37)
0	0	0	1	1	1	1	ANI0~ANI14 通道全关
1	0	0	0	0	0	0	温度传感器的输出电压 ^{注1}
1	0	0	0	0	0	1	内部基准电压 (1.45V)
0	1	0	0	0	0	0	PGA1
0	0	1	0	0	0	0	PGA0
其他设置禁止							

注 1. 如果选择内部基准电压 (1.45V) 作为比较器0或者比较器1的基准电压，就不能选择温度传感器输出。

○扫描模式 (ADM1.ADMD=1)

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

注意1.扫描模式时必须将bit4,bit5和bit6置“0”。

2.对于由PMCx寄存器设置为模拟输入的端口，才可以通过ADS指定为模拟输入进行A/D转换。

3.对于由端口模式控制寄存器 (PMCxx) 设置为数字输入/输出的引脚，不能通过ADS寄存器进行设置。

4.要改写ADISS位时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

5.当将 AV_{REFP} 用作A/D转换器的正 (+) 基准电压时，不能选择ANI6作为A/D转换通道。

6.当将 AV_{REFM} 用作A/D转换器的负 (-) 基准电压时，不能选择ANI5作为A/D转换通道。

7.在将ADISS位置“1”后，不能使用第1次的转换结果。有关详细设置流程，请参照“10.5.4选择温度传感器的输出电压/内部基准电压时的设置”。

8.要转移到深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”。

10.2.7 12位A/D转换结果寄存器（ADCR）

这是保存A/D转换结果的16位寄存器，此寄存器只可读。每当A/D转换结束时，就从逐次逼近寄存器（SAR）装入转换结果注。

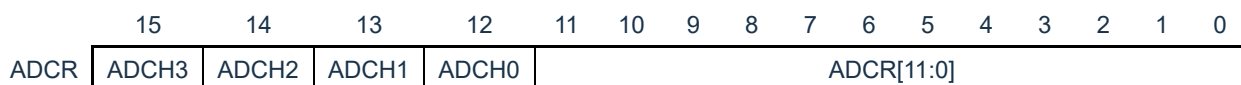
此寄存器的高4位在选择模式时读出值固定为“0”，在扫描模式时可由ADM2.CHRDE=1配置为此次转换结果的通道号。

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

注 如果A/D转换结果的值不在A/D转换结果比较功能（通过ADRCK位和ADUL/ADLL寄存器进行设置，参照图 10-7 ADRCK位的中断信号产生范围）的设置值范围内，就不保存A/D转换结果。

图10-10 12位A/D转换结果寄存器（ADCR）的格式

复位值：0000H R



注意1.如果在仅仅需要8位分辨率A/D转换结果，可以通过ADCRH寄存器读取转换结果的高8位。

2.当对ADCR寄存器进行16位存取时，能从bit11依次读转换结果的高12位。

○选择模式（ADM1.ADMD=0）

ADCH0~3的读出值固定为4 ‘b0000

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

ADCH3	ADCH2	ADCH1	ADCH0	转换通道标识
0	0	0	0	ANI0(P20)
0	0	0	1	ANI1(P21)
0	0	1	0	ANI2(P22)
0	0	1	1	ANI3(P23)
0	1	0	0	ANI4(P24)
0	1	0	1	ANI5(P25)
0	1	1	0	ANI6(P26)
0	1	1	1	ANI7(P30)
1	0	0	0	ANI8(P31)
1	0	0	1	ANI9(P32)
1	0	1	0	ANI10(P33)
1	0	1	1	ANI11(P34)
1	1	0	0	ANI12(P35)
1	1	0	1	ANI13(P36)
1	1	1	0	ANI14(P37)

10.2.8 8位A/D转换结果寄存器（ADCRH）

这是保存A/D转换结果的8位寄存器，保存12位分辨率的高8位注。

通过8位存储器操作指令读ADCRH寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注 如果A/D转换结果的值不在A/D转换结果比较功能（通过ADRCK位和ADUL/ADLL寄存器进行设置（参照图10-7 ADRCK位的中断信号产生范围））的设置值范围内，就不保存A/D转换结果。

图10-11 8位A/D转换结果寄存器（ADCRH）的格式

复位值：00H R



注意 必须在转换结束后并且在配置ADM0、ADS寄存器前读转换结果。否则，就可能读不到正确的转换结果。

10.2.9 转换结果比较上限值设置寄存器 (ADUL)

这是用于检查A/D转换结果上限值的设置寄存器。

将A/D转换结果和ADUL寄存器的值进行比较，并且在A/D转换器的模式寄存器2 (ADM2) 的ADRCK位的设置范围内 (参照图 10-7 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置ADUL寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

注意1. 只将12位A/D转换结果寄存器 (ADCR) 的高8位和ADUL寄存器以及ADLL寄存器进行比较。

2. 要改写ADUL寄存器和ADLL寄存器时，必须在转换停止状态 (ADCS=0) 下进行。

3. 在设置ADUL寄存器和ADLL寄存器时，必须使ADUL > ADLL。

图10-12 转换结果比较上限值设置寄存器 (ADUL) 的格式

复位值: FFH R/W

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

10.2.10 转换结果比较下限值设置寄存器 (ADLL)

这是用于检查A/D转换结果下限值的设置寄存器。

将A/D转换结果和ADLL寄存器的值进行比较，并且在A/D转换器的模式寄存器2 (ADM2) 的ADRCK位的设置范围内 (参照图 10-7 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置ADLL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图10-13 转换结果比较下限值设置寄存器 (ADLL) 的格式

复位值: 00H R/W

	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

注意1. 只将12位A/D转换结果寄存器 (ADCR) 的高8位和ADUL寄存器以及ADLL寄存器进行比较。

2. 要改写ADUL寄存器和ADLL寄存器时，必须在转换停止状态 (ADCS=0) 下进行。

3. 在设置ADUL寄存器和ADLL寄存器时，必须使ADUL > ADLL。

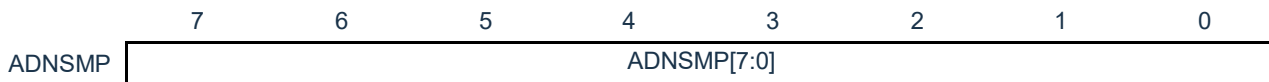
10.2.11 A/D采样时间控制寄存器（ADNSMP）

此寄存器可控制A/D采样时间。

通过8位存储器操作指令设置ADNSMP寄存器。在产生复位信号后，此寄存器的值变为“0dH”。

图10-14 A/D采样时间控制寄存器（ADNSMP）的格式

复位值：0dH R/W



采样时钟数设置：

ADNSMP[7:0]	采样时间	备注
8'h05	5.5 个 ADCLK	
8'h06	6.5 个 ADCLK	
8'h07	7.5 个 ADCLK	
8'h08	8.5 个 ADCLK	
8'h09	9.5 个 ADCLK	
8'h0a	10.5 个 ADCLK	
8'h0b	11.5 个 ADCLK	
8'h0c	12.5 个 ADCLK	
8'h0d	13.5 个 ADCLK	默认值
8'h0e	14.5 个 ADCLK	
8'h0f	15.5 个 ADCLK	
8'h10	16.5 个 ADCLK	
8'h11	17.5 个 ADCLK	
8'h12	18.5 个 ADCLK	
8'h13	19.5 个 ADCLK	
8'h14	20.5 个 ADCLK	
.....	
8'hff	255.5 个 ADCLK	

注意：要改写ADNSMP寄存器时，必须在转换停止状态（ADCS=0）下进行。

进行一次ADC转换需要的时间：

高速变换模式：ADC转换时间 = (采样时钟个数 + 逐次比较时钟个数 (31.5个)) / f_{AD}

低电流变换模式：ADC转换时间 = (采样时钟个数 + 逐次比较时钟个数 (40.5个)) / f_{AD}

其中AD采样时钟个数可通过ADNSMP寄存器进行调整，默认为13.5个ADCLK。逐次比较时钟个数由转换模式决定，高速变换模式时为31.5个ADCLK，低电流模式时为40.5个ADCLK。

不同条件下，各转换通道推荐的采样时间见下表：

采样时间计算公式：采样时钟个数 / $f_{AD} \geq$ 推荐的采样时间

A/D 转换模式	AVDD[V]	ANI0~ANI15[ns]	PGA0/PGA1/温度传感器/内部基准电压[ns]
高速变换	4.5~5.5	211	633
	2.7~5.5	250	750
	2.4~5.5	422	1266
低电流变换	2.7~5.5	500	759
	2.4~5.5	844	1281
	1.8~5.5	1688	2563

注意：实际使用时，请按照数据手册中“AC特性”要求配置转换模式，采样时钟数和转换时钟频率。

10.2.12 A/D采样时间延长寄存器 (ADSMPWAIT)

此寄存器用于延长A/D采样时间。

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

图10-15 A/D采样时间延长寄存器 (ADSMPWAIT) 的格式

复位值：00H R/W

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

ADSMPWAIT	A/D转换对象
0	为“0”时，A/D采样时间直接由ADNSMP寄存器进行设置
1	为“1”时任意延长A/D采样时间，当由“1”变“0”后，再由ADNSMP继续控制采样时间

注意：转换停止状态 (ADCS=0) 下设置ADSMPWAIT=1，在 (ADCS=1) 时可改写ADSMPWAIT为“0”。

10.2.13 A/D测试寄存器 (ADTES)

此寄存器用于设置A/D转换器的测试模式。

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

图10-16 A/D测试寄存器 (ADTES) 的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	ADTES2	ADTES1	ADTES0

ADTES2	ADTES1	ADTES0	A/D运行模式
0	0	0	通常转换
0	0	1	0码的自诊断测试
0	1	1	半码的自诊断测试
1	0	1	全码的自诊断测试
上述以外			禁止设置。

注意 必须将bit7~3置“0”。

10.2.14 A/D状态寄存器 (ADFLG)

此寄存器表示A/D转换器的状态。

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

图10-17 A/D状态寄存器 (ADFLG) 的格式

复位值: 00H R

	7	6	5	4	3	2	1	0
ADFLG	0	0	0	ADFLG4	ADFLG3	ADFLG2	ADFLG1	ADFLG0

ADFLG4	A/D转换状态
0	单次转换模式时，A/D转换未结束
1	单次转换模式时，转换结束（2个ADCLK后自动清零）。 连续转换模式时ADFLG4保持1'b0

ADFLG3	A/D转换状态
0	非A/D转换结束前1个ADCLK
1	A/D转换结束前1个ADCLK(1个ADCLK后自动清零)

ADFLG2	A/D转换状态
0	非A/D转换结束前2个ADCLK
1	A/D转换结束前2个ADCLK(1个ADCLK后自动清零)

ADFLG1	A/D转换状态
0	非逐次比较期间
1	逐次比较期间

ADFLG0	A/D转换状态
0	非A/D采样期间
1	A/D采样期间

10.2.15 A/D充放电控制寄存器 (ADNDIS)

此寄存器用来控制A/D转换器的充放电动作及时间。

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

图10-18 A/D充放电控制寄存器 (ADNDIS) 的格式

复位值：00HW

	7	6	5	4	3	2	1	0
ADNDIS	0	0	0	ADNDIS4	ADNDIS3	ADNDIS2	ADNDIS1	ADNDIS0

ADNDIS[4]	充放电控制
1'b0	放电
1'b1	充电

ADNDIS[3:0]	充放电时间
4'b0000	不进行充、放电
4'b0010	2个ADCLK
4'b0011	3个ADCLK
4'b0100	4个ADCLK
4'b0101	5个ADCLK
4'b0110	6个ADCLK
.....
4'b1111	15个ADCLK

注：禁止设置充放电时间为1个ADCLK，即ADNDIS[3:0]=4'b0001

10.2.16 控制模拟输入引脚端口功能的寄存器

在将ANI0~ANI14引脚用作A/D转换器的模拟输入时，必须将各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”。

10.3 输入电压和转换结果

模拟输入引脚（ANI0~ANI14）的模拟输入电压和理论上的A/D转换结果（12位A/D转换结果寄存器（ADCR））有以下表达式的关系。

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

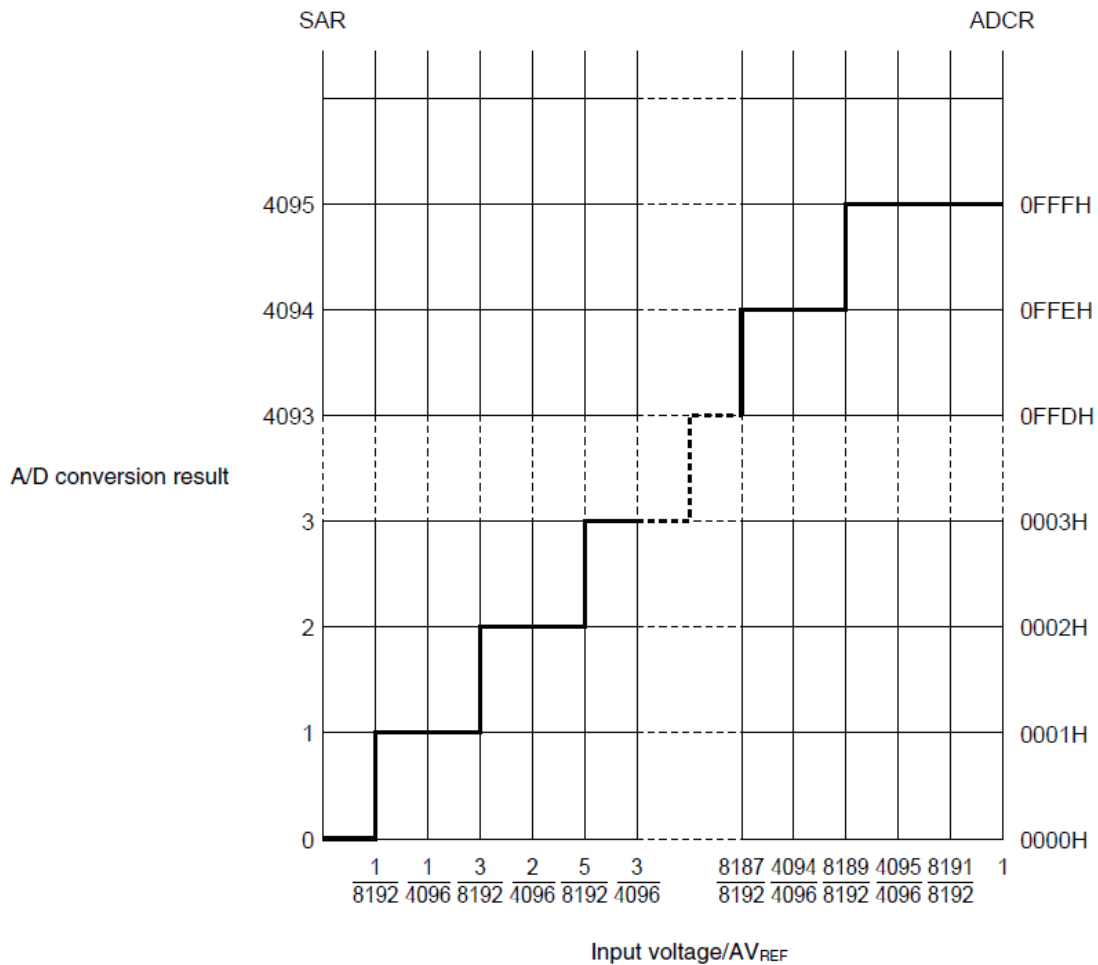
or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

- INT() : 将括号中的数值的整数部分返回的函数
- V_{AIN} : 模拟输入电压
- AV_{REF} : AV_{REF} 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10-19 模拟输入电压和A/D转换结果的关系



备注 AV_{REF} 是A/D转换器的正（+）基准电压，可选择 AV_{REFP} 或者 V_{DD} 。

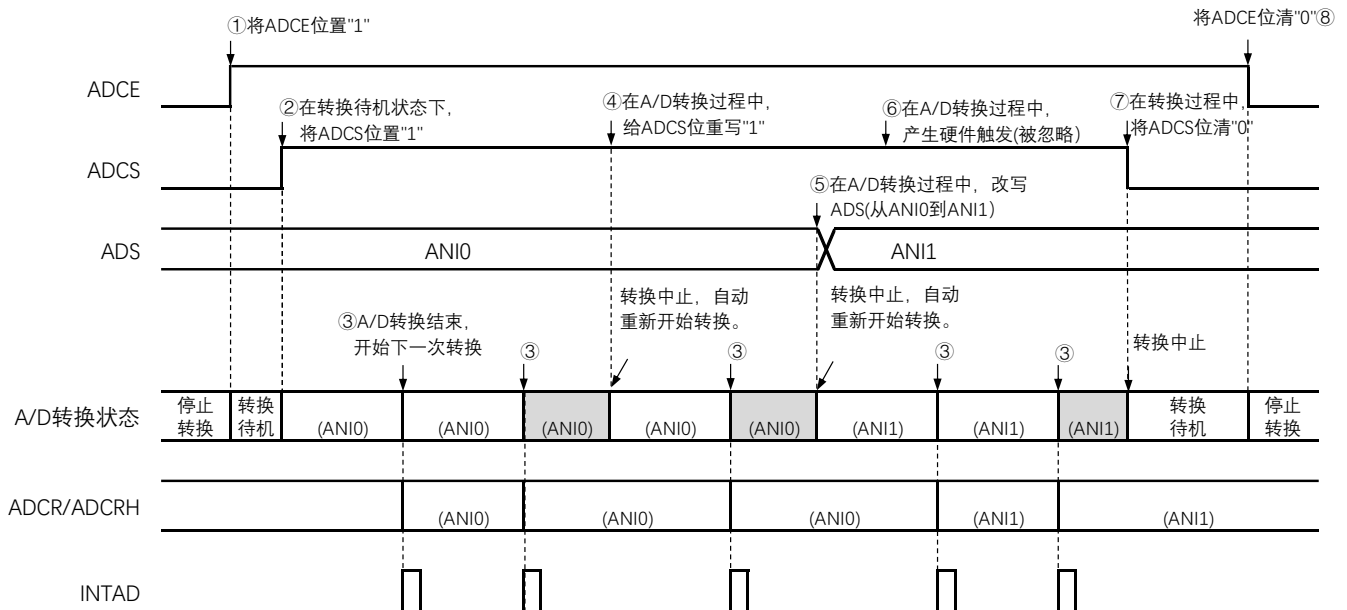
10.4 A/D转换器的运行模式

A/D转换器的各模式的运行如下所示。有关各模式的设置步骤，请参照“10.5 A/D转换器的设置流程图”。

10.4.1 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。

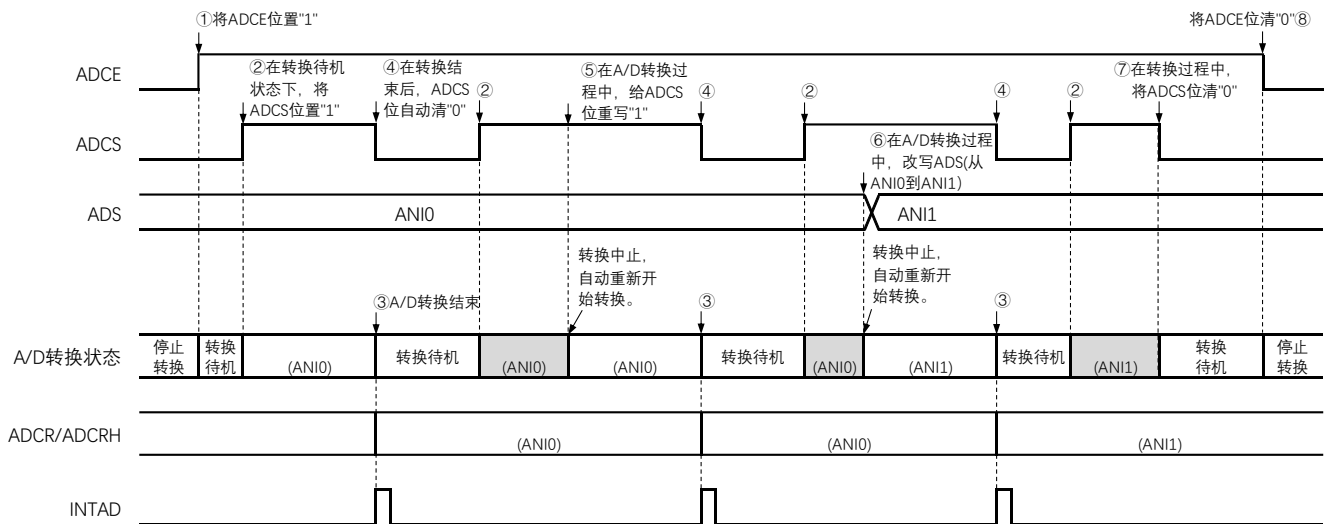
图 10-20 软件触发模式（选择模式、连续转换模式）的运行时序例子



10.4.2 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。

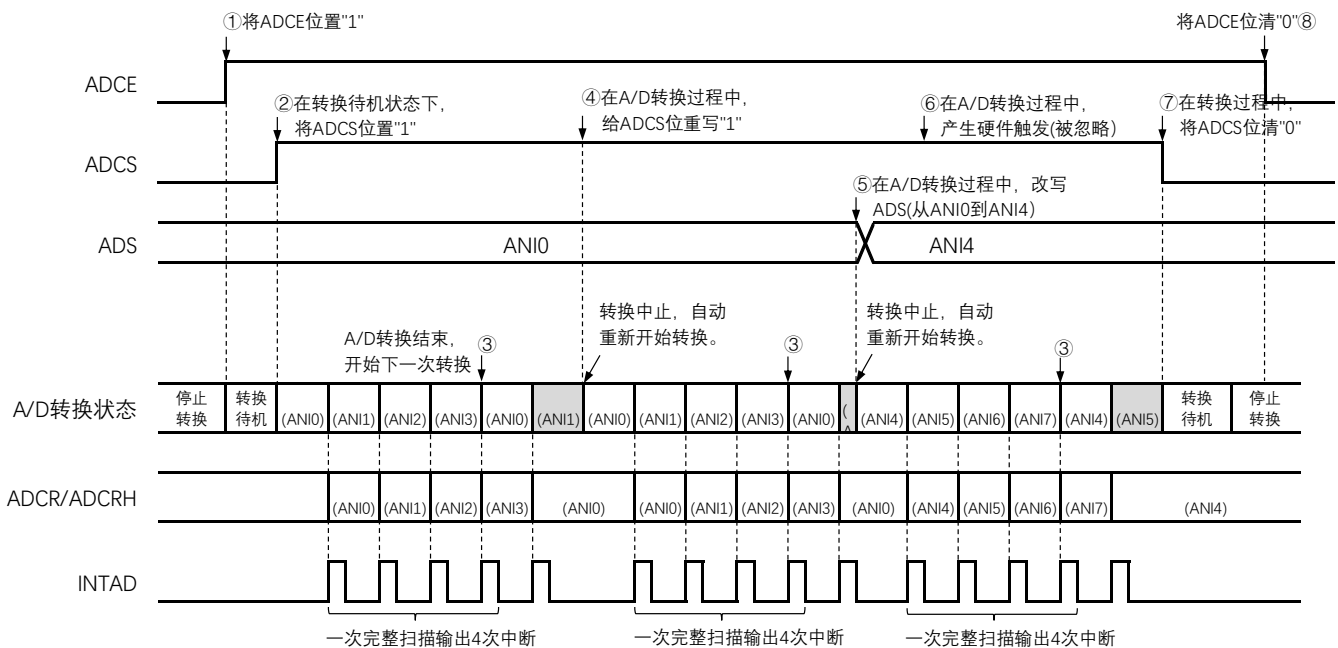
图 10-21 软件触发模式（选择模式、单次转换模式）的运行时序例子



10.4.3 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。

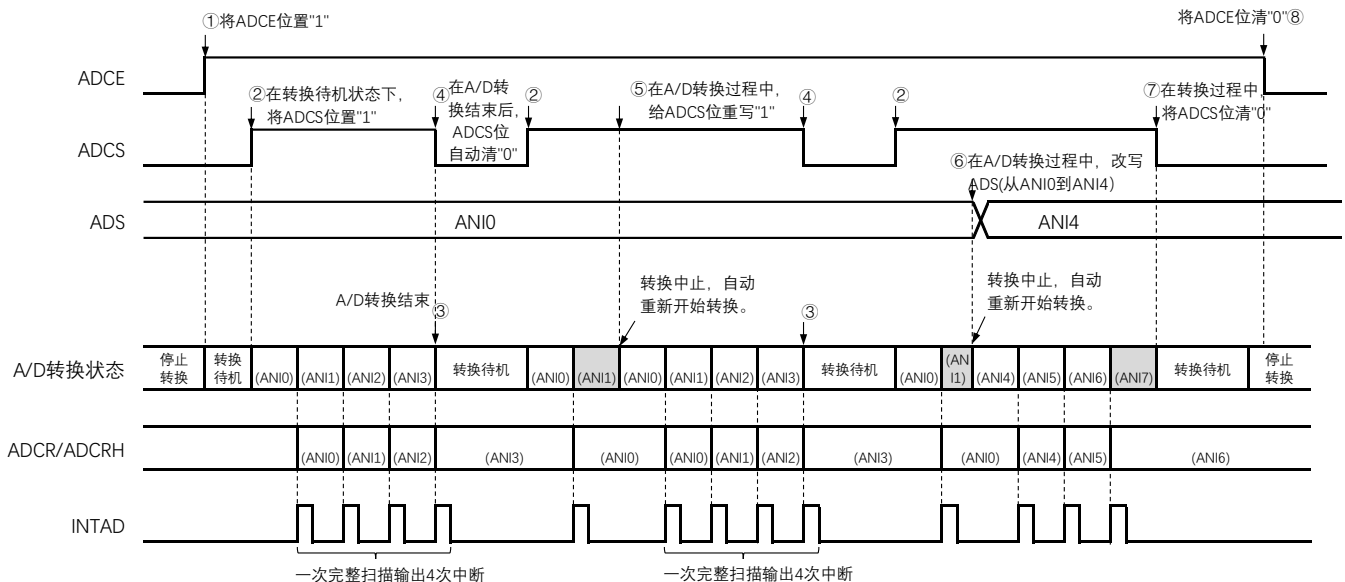
图10-22 软件触发模式（扫描模式、连续转换模式）的运行时序例子



10.4.4 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。

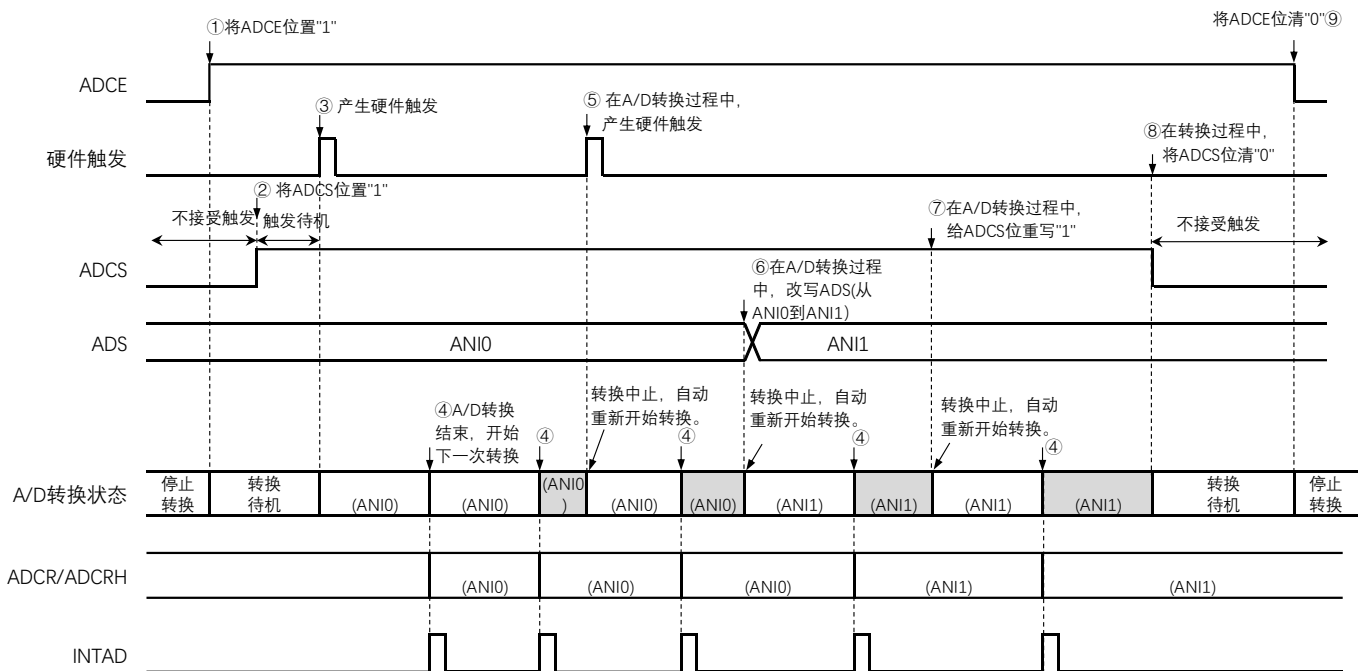
图10-23 软件触发模式（扫描模式、单次转换模式）的运行时序例子



10.4.5 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。

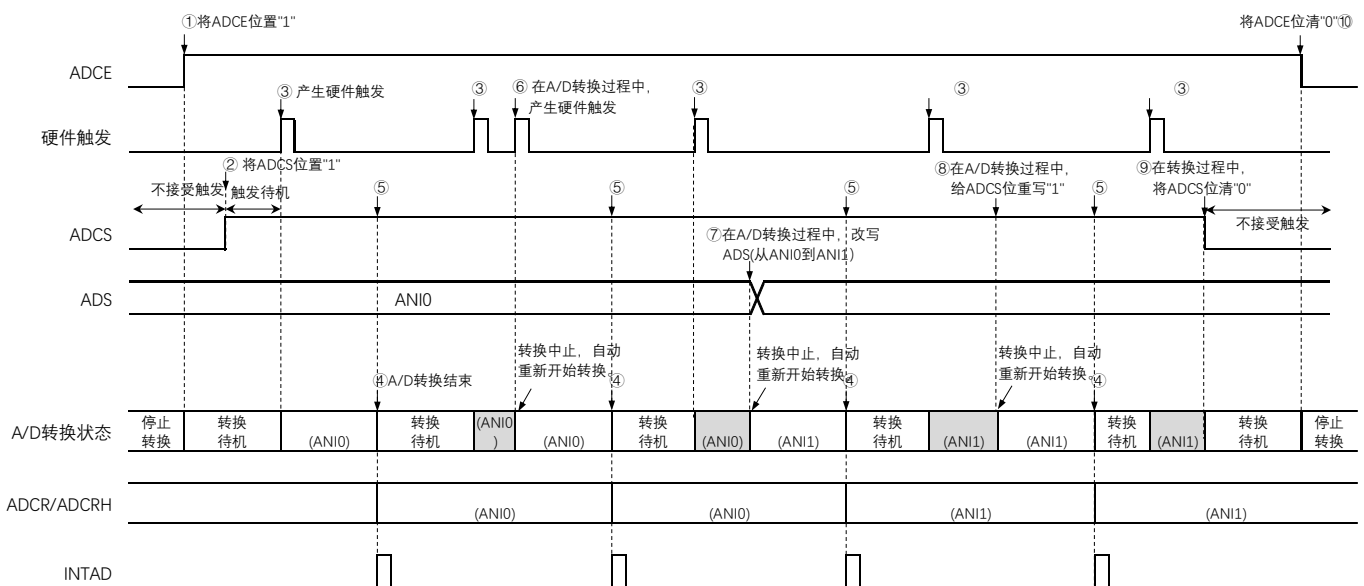
图10-24 硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



10.4.6 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。

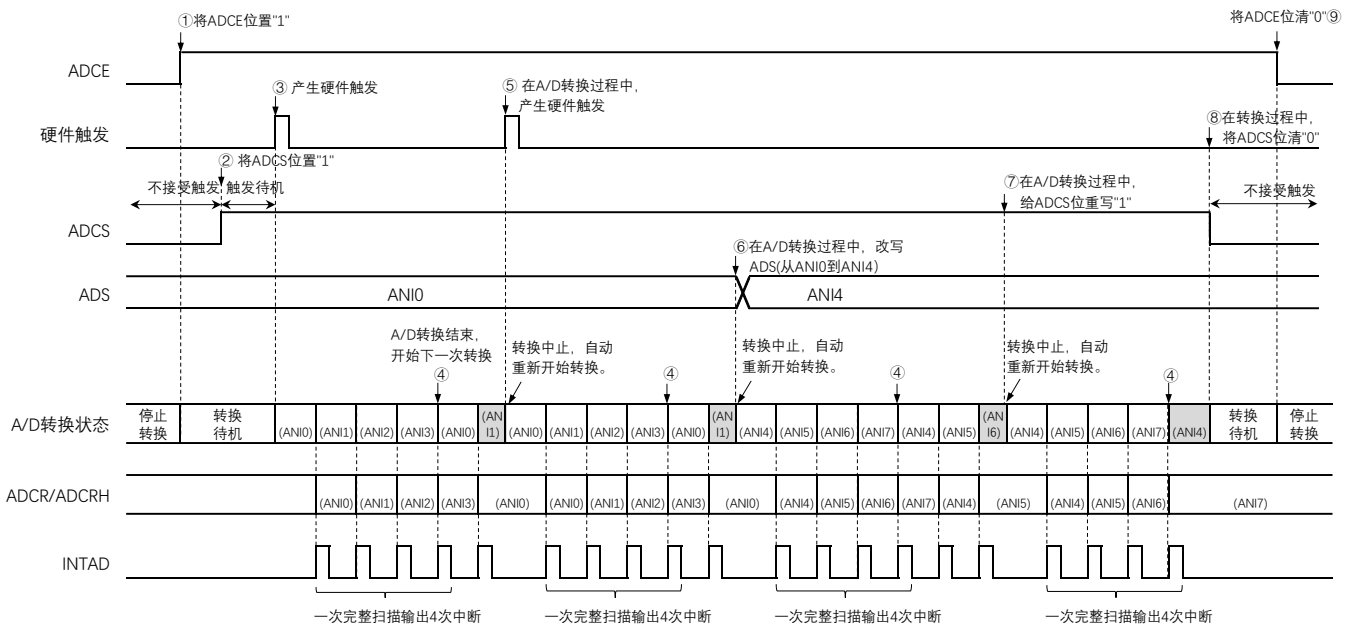
图10-25 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



10.4.7 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。

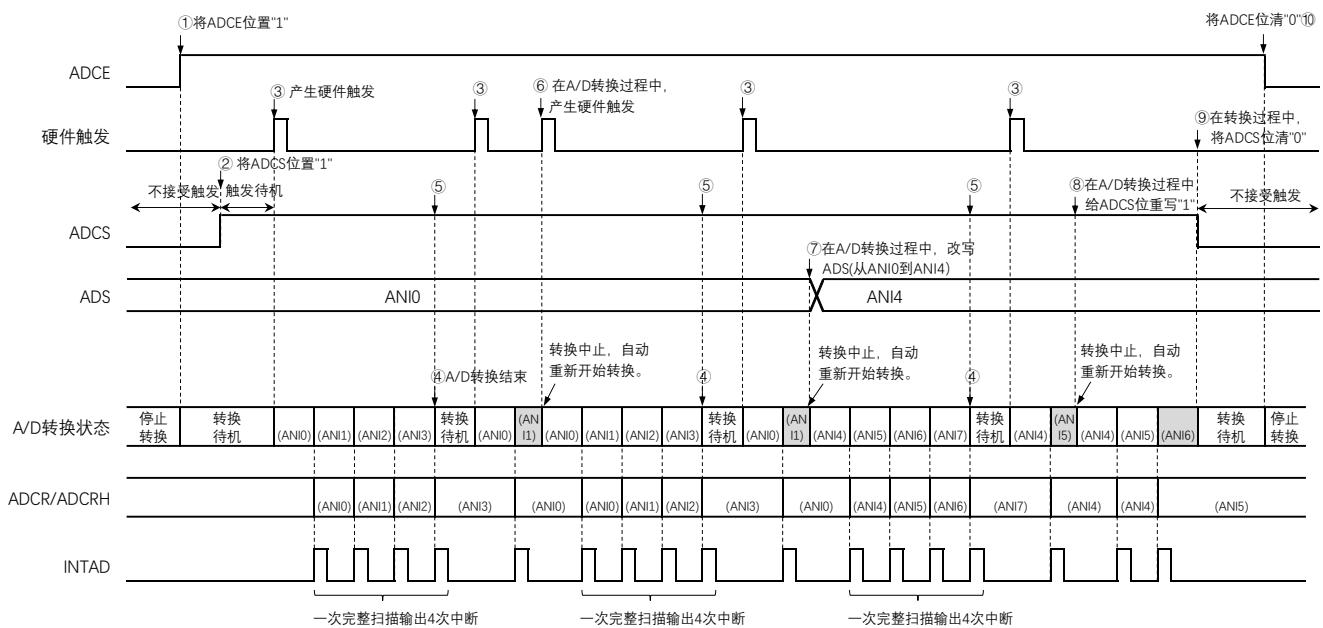
图10-26 硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



10.4.8 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。

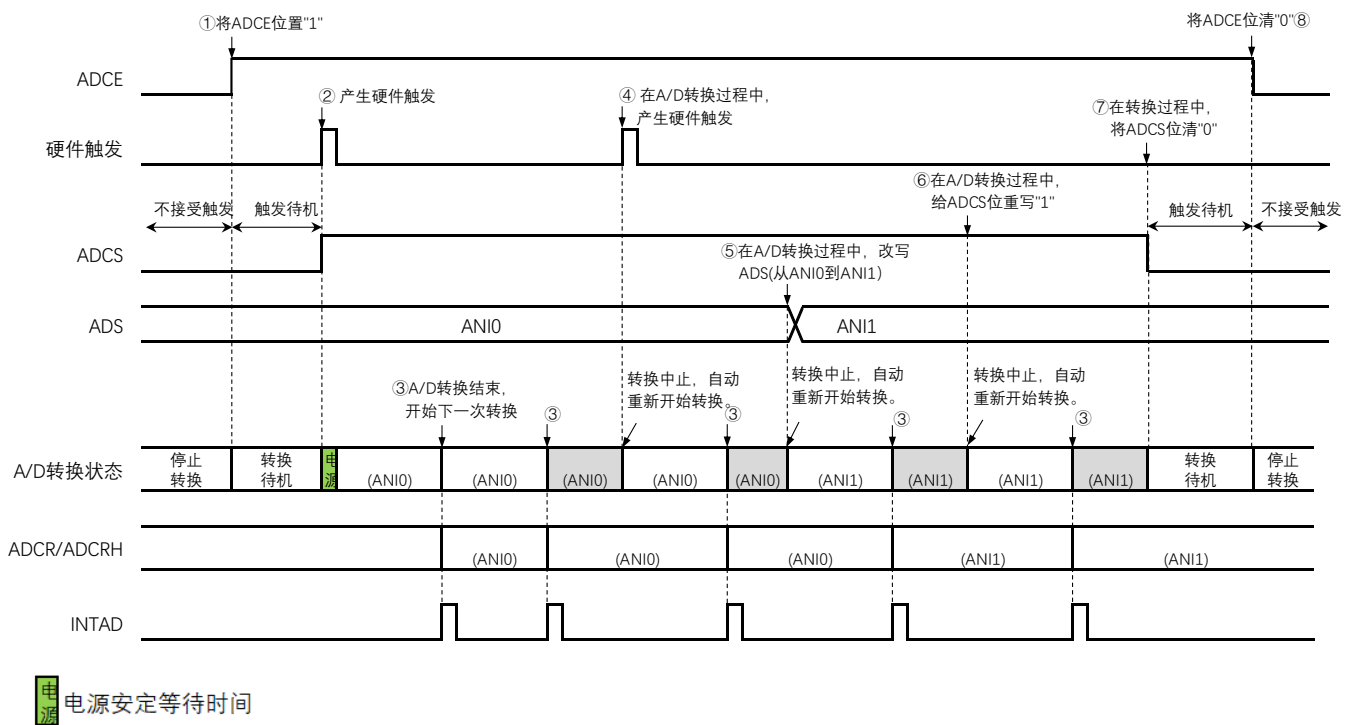
图10-27 硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



10.4.9 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。

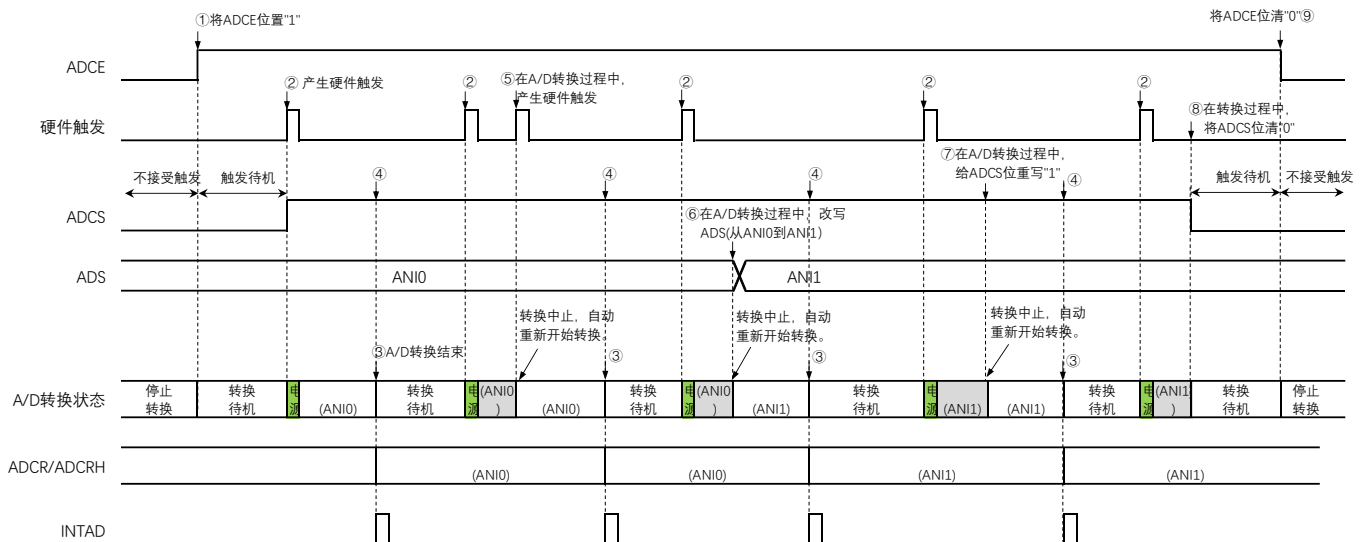
图10-28 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



10.4.10 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10-29 硬件触发等待模式（选择模式、单次转换模式）的运行时序例子

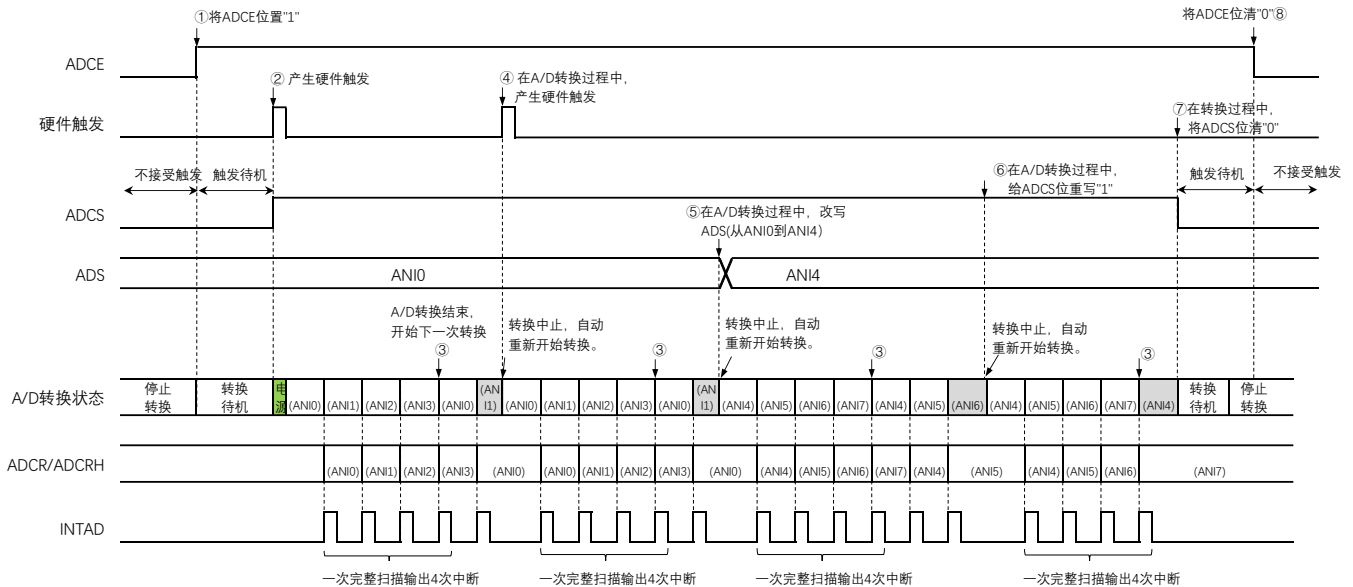


电源安定等待时间

10.4.11 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A/D转换。

图10-30 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

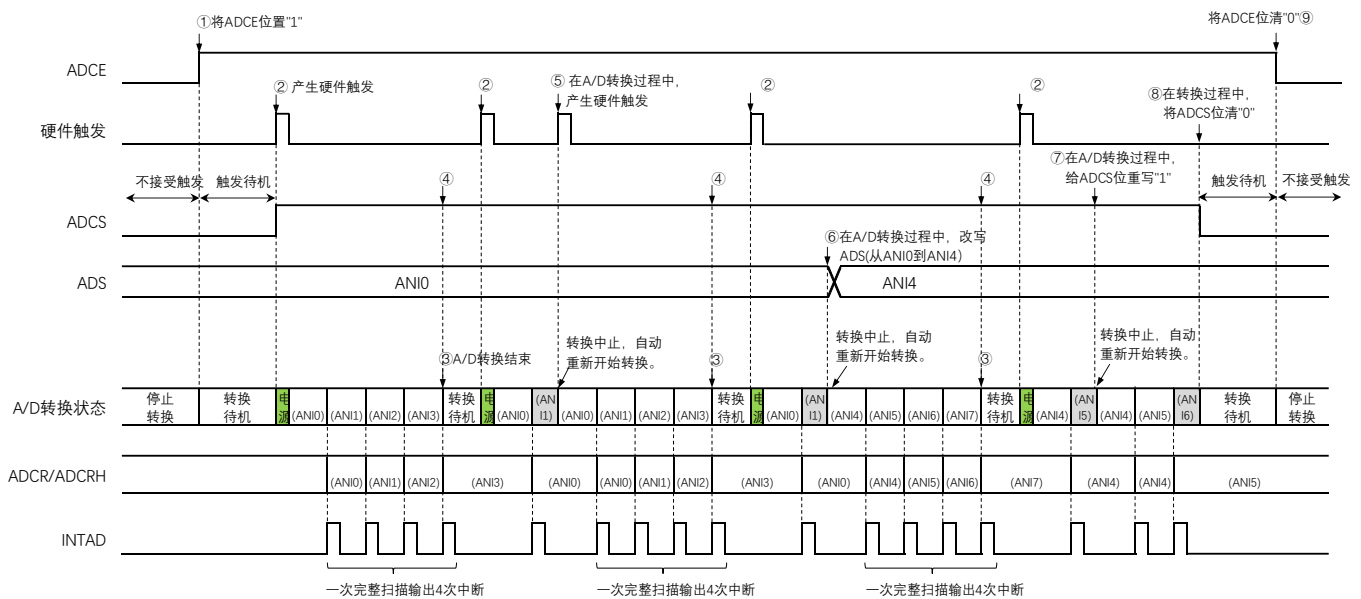


电源安定等待时间

10.4.12 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 10-31 硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子



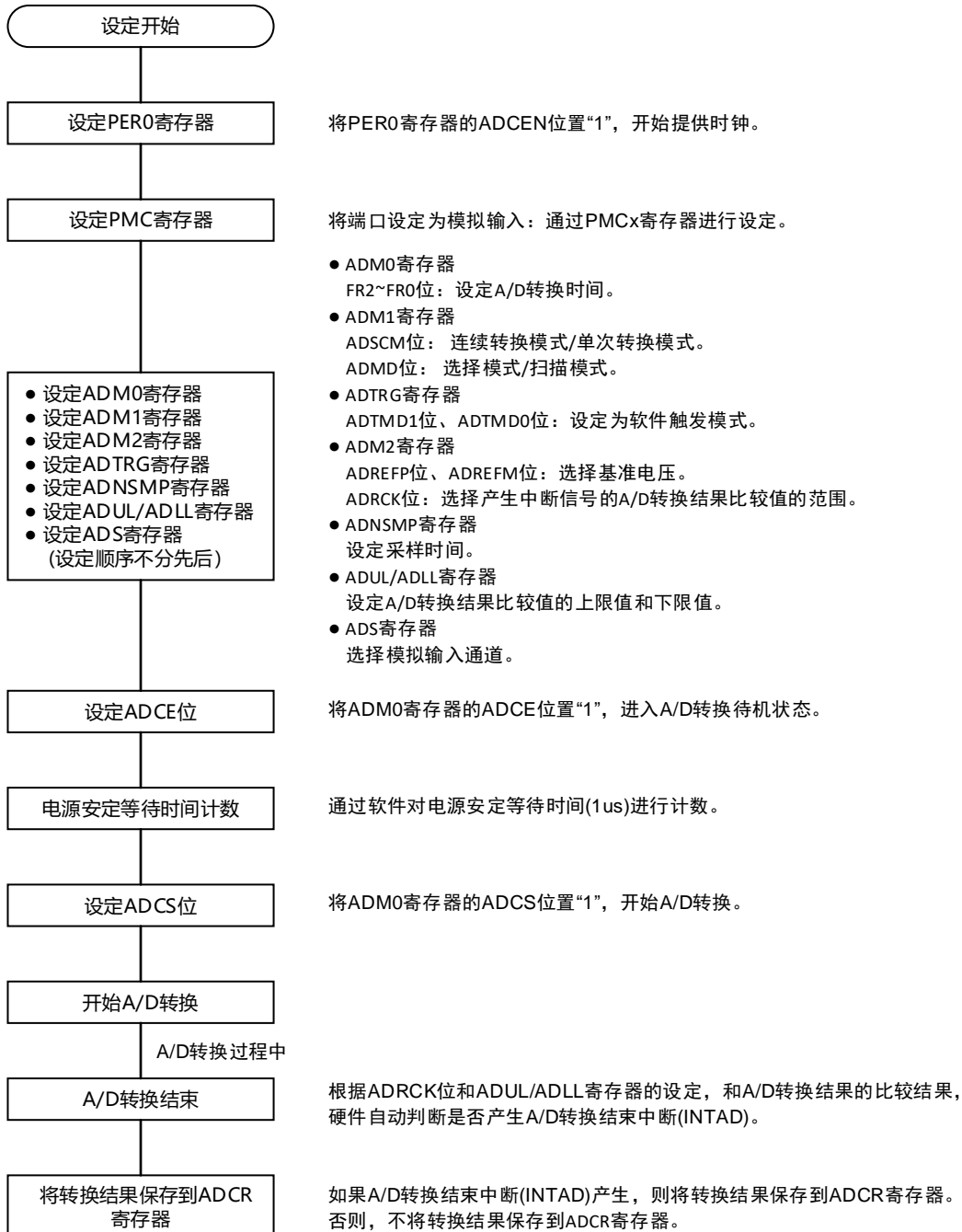
电源安定等待时间

10.5 A/D转换器的设置流程图

各运行模式的A/D转换器的设置流程图如下所示。

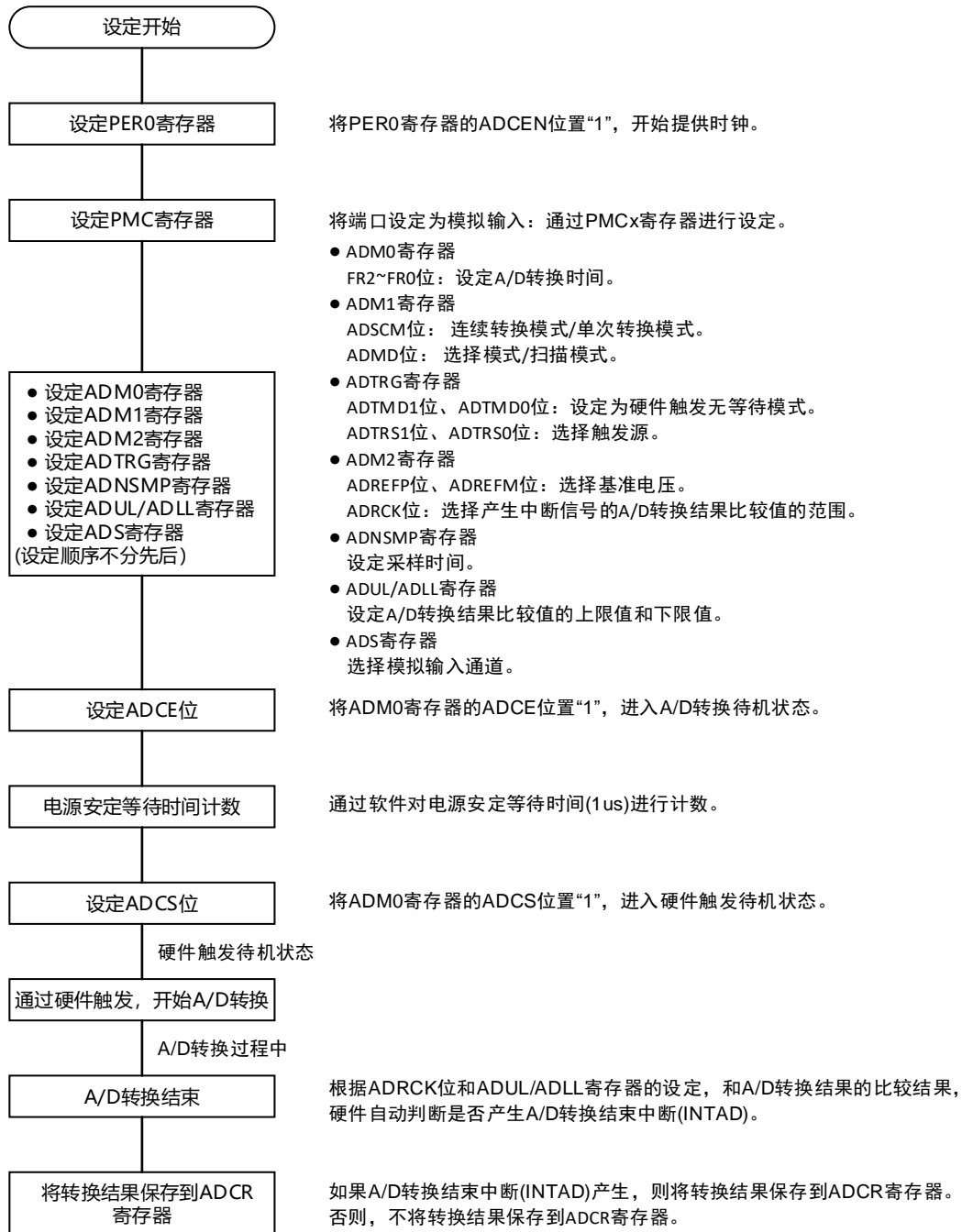
10.5.1 软件触发模式的设置

图10-32 软件触发模式的设置



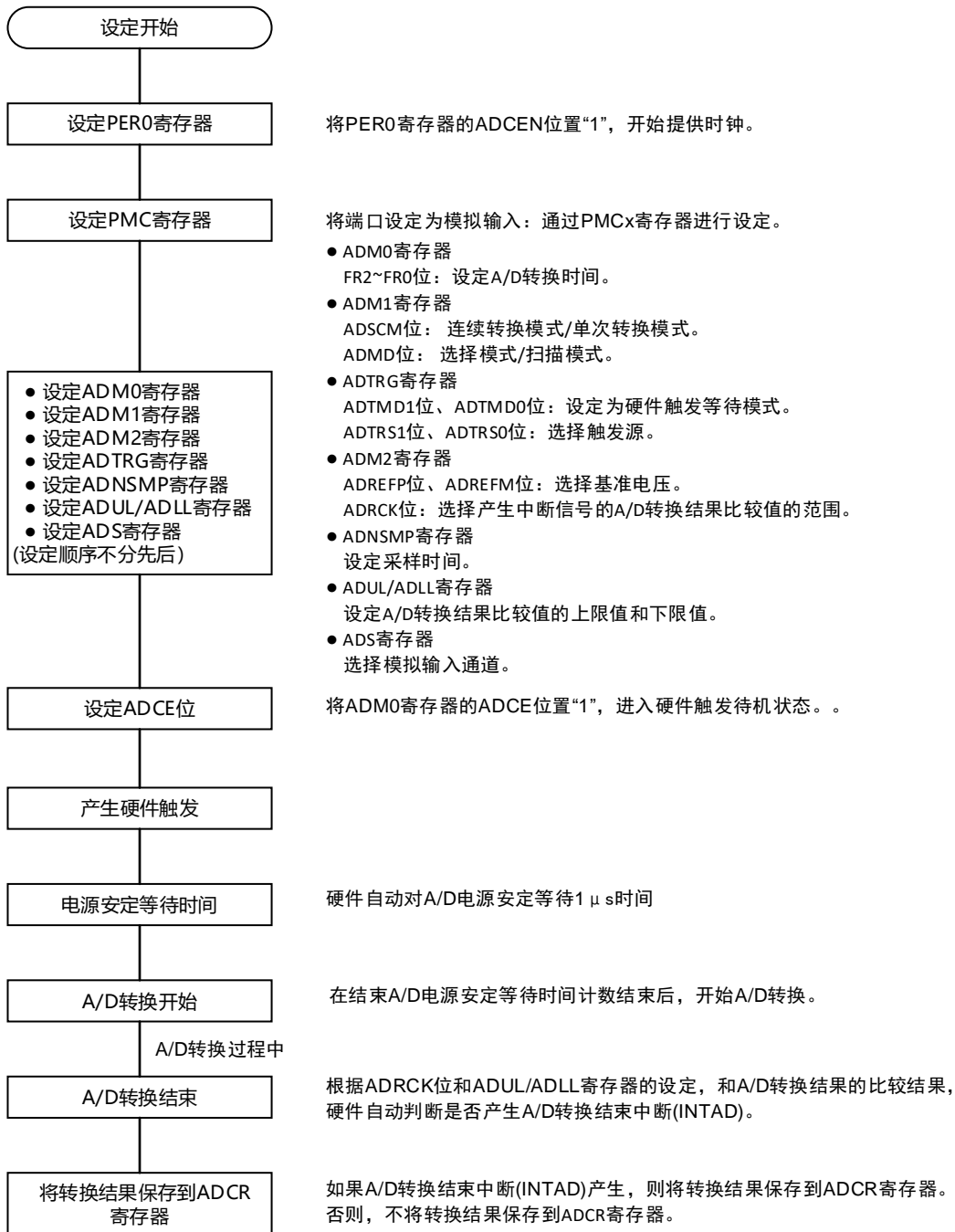
10.5.2 硬件触发无等待模式的设置

图10-33 硬件触发无等待模式的设置



10.5.3 硬件触发等待模式的设置

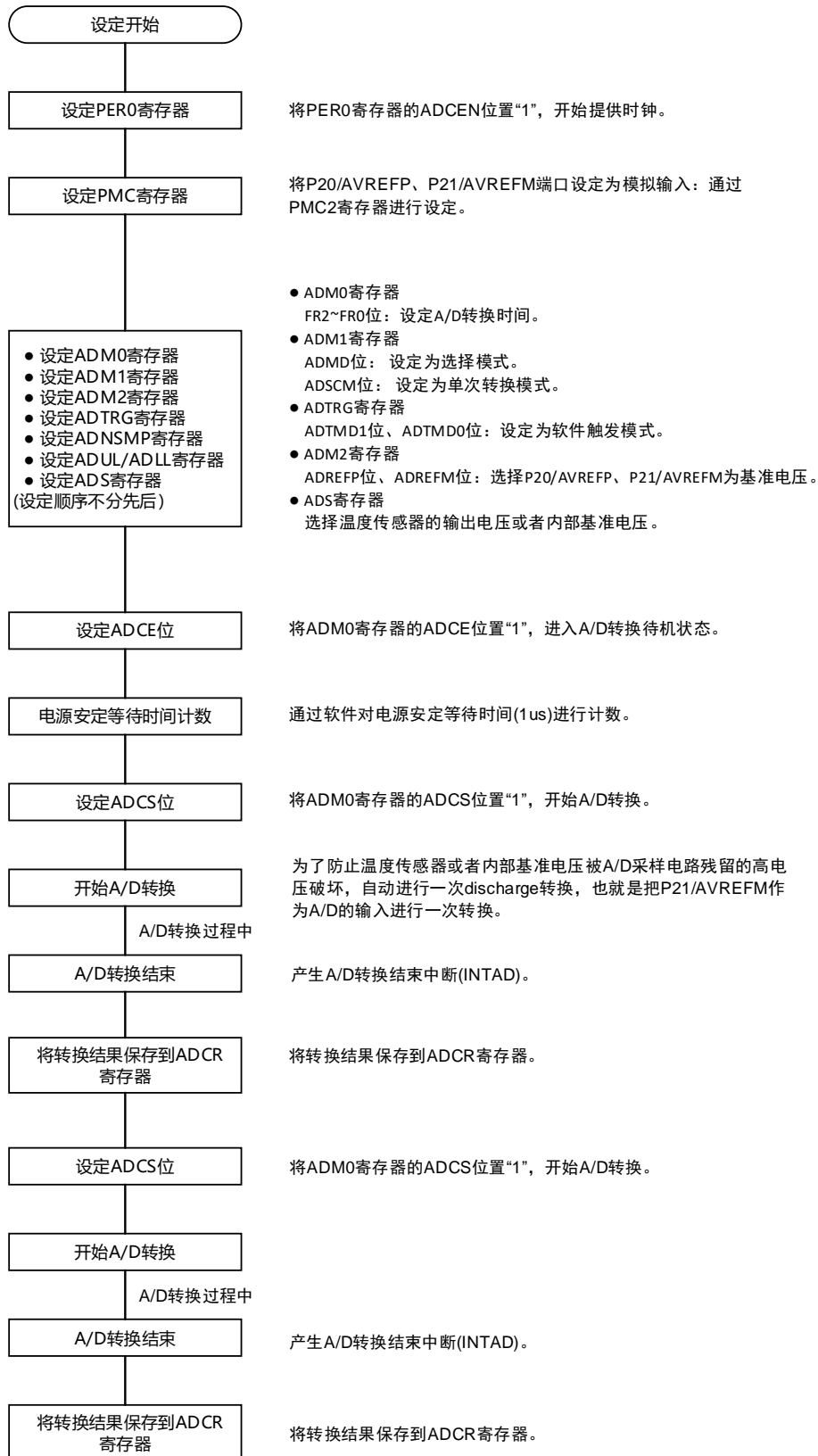
图10-34 硬件触发等待模式的设置



10.5.4 选择温度传感器的输出电压/内部基准电压时的设置

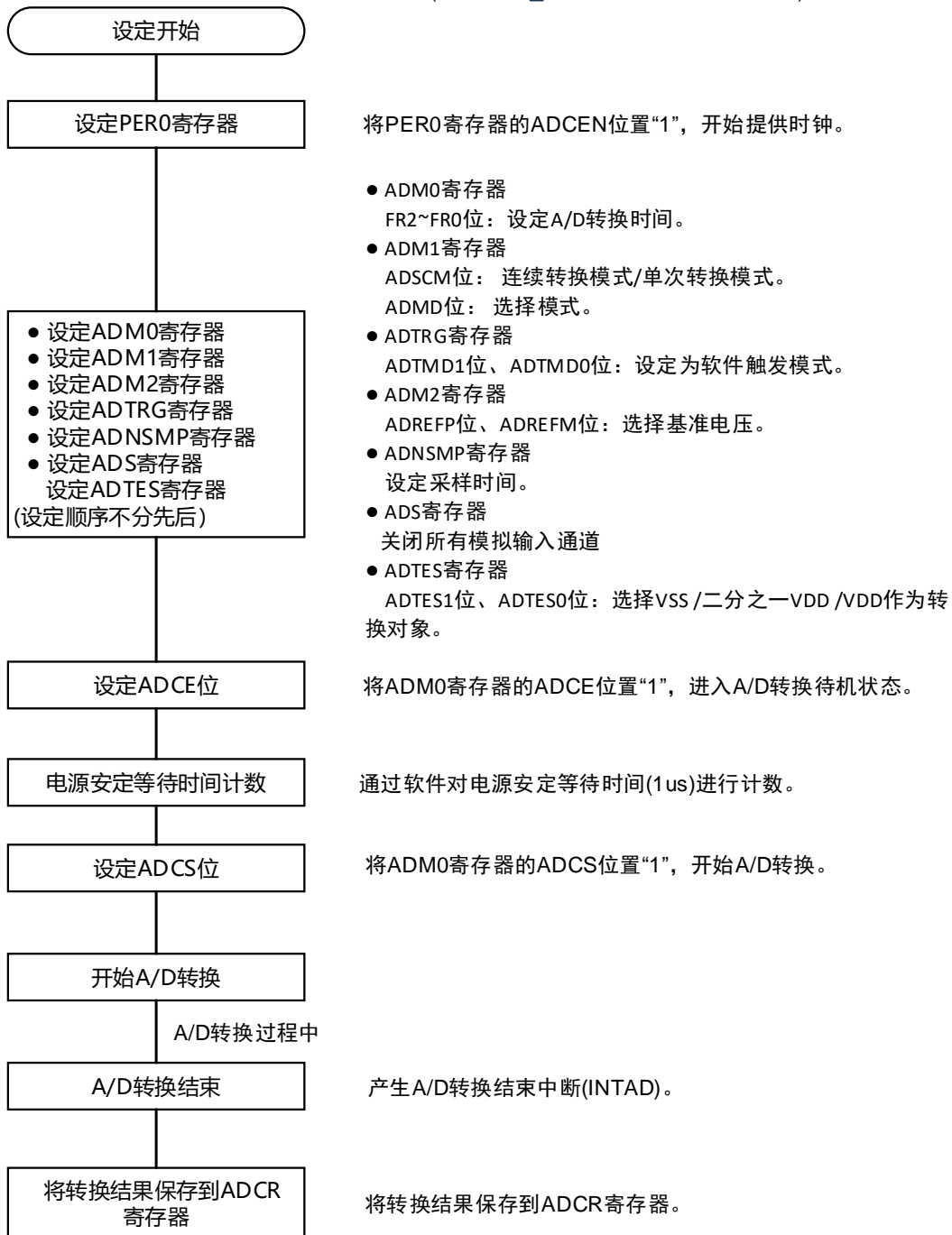
(以软件触发模式、单次转换模式为例)

图10-35 选择温度传感器的输出电压/内部基准电压时的设置



10.5.5 测试模式的设置

图10-36 测试模式的设置(VSS/half_VDD/VDD作为转换对象)



第11章 比较器

本产品内置2个通道的比较器。

11.1 比较器的功能

比较器有以下功能：

- CMP1的输入引脚可选择外部端口，内部基准电压以及内置DAC基准电压。
- 比较器0和比较器1的比较结果可以通过引脚输出（VCOUT0，VCOUT1）。

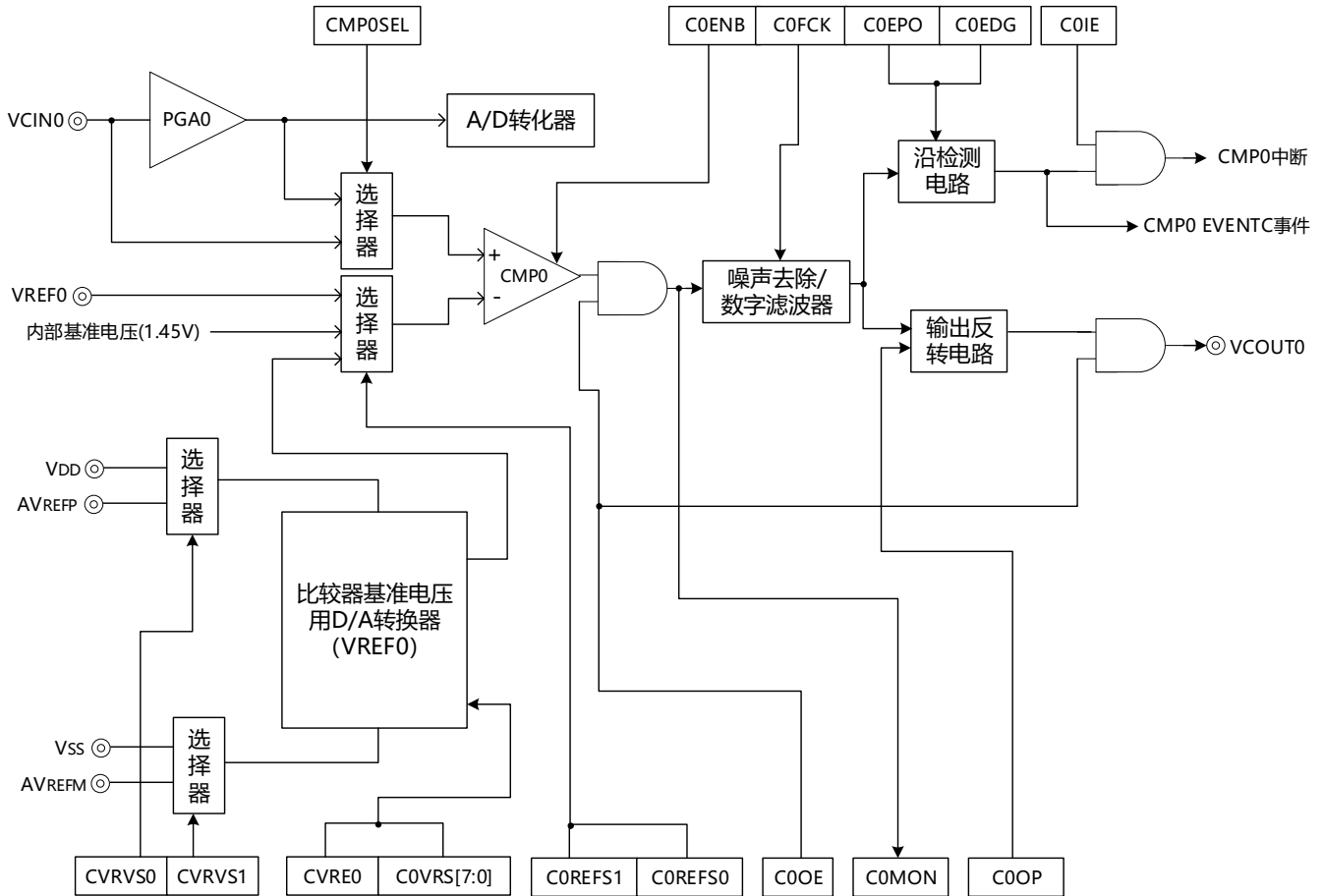
表 11-1 比较器的功能概要

项目	内容
CMP	<ul style="list-style-type: none"> • 2通道比较器（CMP0和CMP1） • 比较器负端可以选择基准电压： CMP0的负端可选外部引脚输入，CMP0的内置基准电压和内部基准电压(1.45V) CMP1的负端可选外部引脚输入（4个），CMP1的内置基准电压和内部基准电压(1.45V) • 负端的内部参考电压可设置（256steps） • CMP0的正端可选择PGA的输出 • CMP1的正端可以选择外部引脚输入（4个） • 正端输入电压>负端输入电压时，输出高电平 正端输入电压<负端输入电压时，输出低电平 • 数字滤波器的滤波宽度可选 • 输出反转功能 • 比较结果可以从引脚（VCOUT0，VCOUT1）输出 • 能检测比较器输出的有效边沿并且产生中断信号 • 与Timer4组合可输出TIMER WINDOW

11.2 比较器的结构

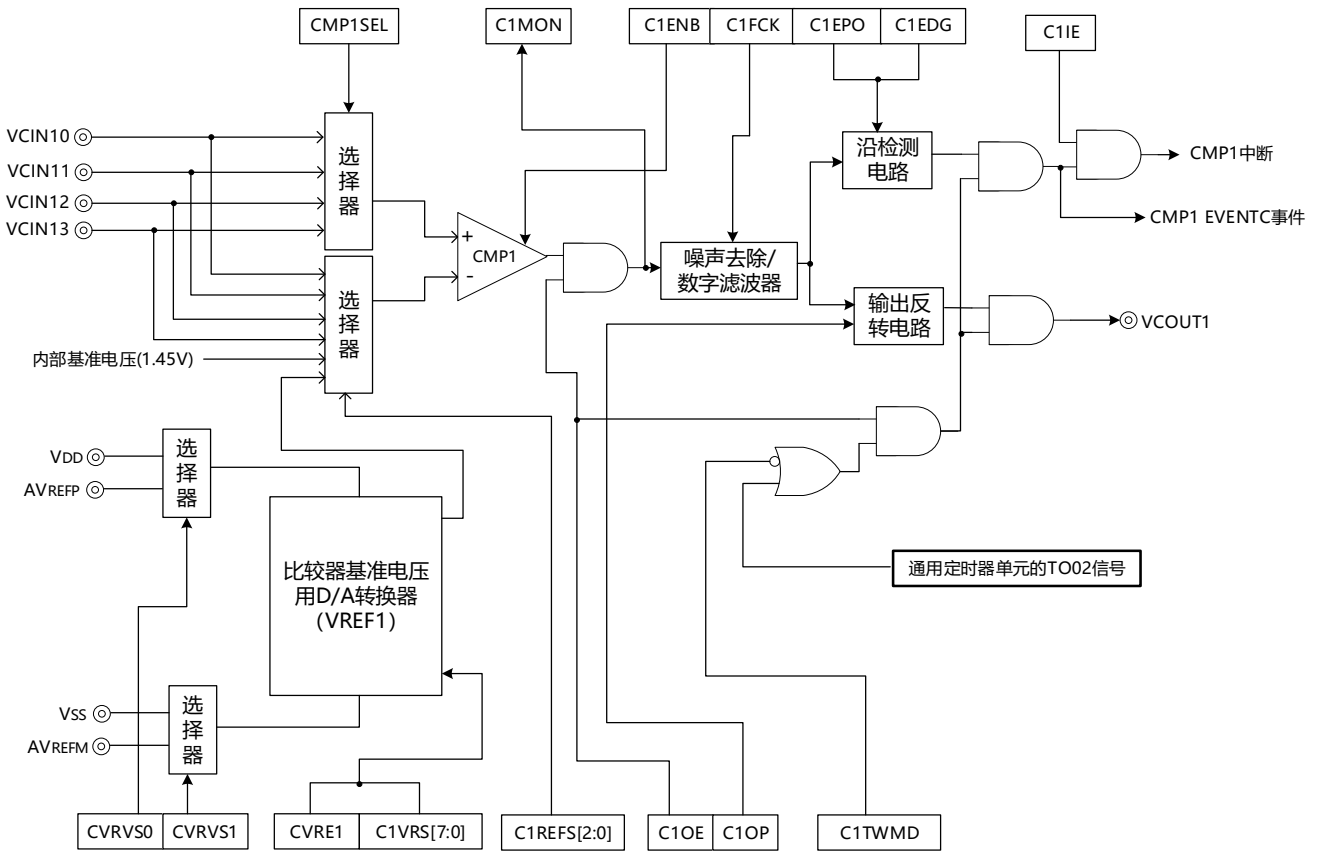
比较器0的框图如图11-1所示。

图11-1 比较器0的框图



比较器1的框图如图11-2 比较器1的框图所示。

图11-2 比较器1的框图



11.3 控制比较器的寄存器

控制比较器的寄存器如表11-2 控制比较器的寄存器所示。

表11-2 控制比较器的寄存器

寄存器名	符号
外围允许寄存器1	PER1
比较器模式设置寄存器	COMPMDR
比较器滤波控制寄存器	COMPFIR
比较器输出控制寄存器	COMPOCR
比较器内置基准电压控制寄存器	CVRCTL
比较器内置基准电压选择寄存器0	C0RVM
比较器内置基准电压选择寄存器1	C1RVM
比较器0输入选择控制寄存器	CMPSEL0
比较器1输入选择控制寄存器	CMPSEL1
端口模式控制寄存器	PMCxx
端口模式寄存器	PMxx
端口复用功能配置寄存器	PxxCFG

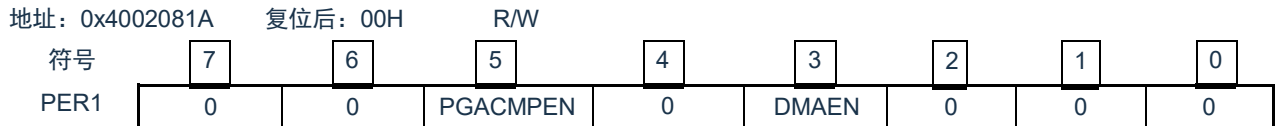
11.3.1 外围允许寄存器1 (PER1)

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

要使用比较器时，必须将bit5 (PGACMPEN) 置“1”。

通过8位存储器操作指令设置PER1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-3 外围允许寄存器1 (PER1) 的格式



PGACMPEN	比较器输入时钟的控制
0	停止提供输入时钟。 •不能写比较器使用的SFR。 •比较器处于复位状态。
1	提供输入时钟。 •能读写比较器使用的SFR。

注意 要设置比较器时，必须先将PGACMPEN位置“1”。

当PGACMPEN位为“0”时，忽视比较器的控制寄存器的写操作，并且读取值都为初始值（端口寄存器（PMCxx, PMxx和PxxCFG）除外）。

11.3.2 比较器模式设置寄存器 (COMPMDR)

COMPMDR寄存器是设置比较器动作许可/禁止以及检测比较器输出的寄存器。

CiENB位在比较器输出许可(COMPOCR寄存器的CiOE位置“1”)时,禁止设置为“0”。

以下几种情况,禁止将CiENB位置“1”(i=0,1):

- CMP负端输入选择内置基准电压,而内置基准电压动作停止(CVRCTL寄存器的CVREi位为“0”)时
- CMP0的输入选择PGA的输出,而PGA动作停止(CVRCTL寄存器的CMPSEL0位为“1”且PGAEN寄存器的PGAEN位为“0”)时

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

图11-4 比较器模式设置寄存器 (COMPMDR) 的格式

地址: 40043840H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
COMPMDR	C1MON	0	0	C1ENB	C0MON	0	0	C0ENB

C1MON	比较器1的监视标志 ^{注1、2}
0	IVCMP1 < 比较器1的基准电压, 或者比较器1停止运行。
1	IVCMP1 > 比较器1的基准电压

C1ENB	比较器1运行的允许
0	禁止比较器1的运行。
1	允许比较器1的运行。

C0MON	比较器0的监视标志 ^{注1、2}
0	IVCMP0 < 比较器0的基准电压, 或者比较器0停止运行。
1	IVCMP0 > 比较器0的基准电压

C0ENB	比较器0运行的允许
0	禁止比较器0的运行。
1	允许比较器0的运行。

注 1. 在解除复位后立即变为“0”(初始值),如果在允许比较器的运行后将C0ENB位和C1ENB位都置“0”,就为不定值。

2. 忽视此位的写入值。

11.3.3 比较器滤波控制寄存器 (COMPFIR)

COMPFIR寄存器是数字滤波器的控制寄存器。通过8位存储器操作指令设置COMPFIR寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-5 比较器滤波控制寄存器 (COMPFIR) 的格式

地址: 40043841H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0	

C1EDG	比较器1边沿检测的选择 ^{注1}	
0	通过比较器1的单边沿检测产生中断请求。	
1	通过比较器1的双边沿检测产生中断请求。	

C1EPO	比较器1边沿极性的切换 ^{注1}	
0	通过比较器1的上升沿产生中断请求。	
1	通过比较器1的下降沿产生中断请求。	

C1FCK1	C1FCK0	比较器1滤波器的选择 ^{注1}
0	0	比较器1没有滤波器。
0	1	比较器1有滤波器，通过 f_{CLK} 进行采样。
1	0	比较器1有滤波器，通过 $f_{CLK}/8$ 进行采样。
1	1	比较器1有滤波器，通过 $f_{CLK}/32$ 进行采样。

C0EDG	比较器0边沿检测的选择 ^{注2}	
0	通过比较器0的单边沿检测产生中断请求。	
1	通过比较器0的双边沿检测产生中断请求。	

C0EPO	比较器0边沿极性的切换 ^{注2}	
0	通过比较器0的上升沿产生中断请求。	
1	通过比较器0的下降沿产生中断请求。	

C0FCK1	C0FCK0	比较器0滤波器的选择 ^{注2}
0	0	比较器0没有滤波器。
0	1	比较器0有滤波器，通过 f_{CLK} 进行采样。
1	0	比较器0有滤波器，通过 $f_{CLK}/8$ 进行采样。
1	1	比较器0有滤波器，通过 $f_{CLK}/32$ 进行采样。

注1.如果更改C1FCK1~C1FCK0位、C1EPO位和C1EDG位，就可能产生比较器1的中断请求和向EVENTC输出的事件信号。必须在将EVENTC的ELSELR14寄存器（不链接比较器1的输出）置“0”后更改这些位。另外，必须将中断请求标志寄存器的IF清“0”。

如果将C1FCK1~C1FCK0位从“00B”（比较器1无滤波器）改为其他值（比较器1有滤波器），就必须在更新滤波器的输出前经过4次采样后，使用比较器1的中断请求或者向EVENTC输出的事件信号。

2.如果更改C0FCK1~C0FCK0位、C0EPO位和C0EDG位，就可能产生比较器0的中断请求和向EVENTC输出的事件信号。必须在将EVENTC的ELSELR13寄存器（不链接比较器0的输出）置“0”后更改这些位。另外，必须将中断请求标志寄存器的IF清“0”。

如果将C0FCK1~C0FCK0位从“00B”（比较器0无滤波器）改为其他值（比较器0有滤波器），就必须在更新滤波器的输出前经过4次采样后，使用比较器0的中断请求或者向EVENTC输出的事件信号。

11.3.4 比较器输出控制寄存器 (COMPOCR)

COMPOCR寄存器是设置比较器输出的极性，输出的许可/禁止以及中断输出的许可/禁止的控制寄存器。

以下的几种情况下，禁止将COMPOCR寄存器的CiOE位置“1”（输出许可）。(i=0,1)

比较器动作停止(COMPMDR寄存器的CiENB位为“0”)时CMP负端输入选择内置基准电压，而内置基准电压动作停止(CVRCTL寄存器的CVREi位为“0”)时CMP0的输入选择PGA的输出，而PGA动作停止(CVRCTL寄存器的CMPSEL0位为“1”且PGAEN寄存器的PGAEN位为“0”)时通过8位存储器操作指令设置COMPOCR寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图11-6 比较器输出控制寄存器 (COMPOCR) 的格式

地址: 40043842H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
COMPOCR	C10TWMD	C10P	C10E	C11E	0	C00P	C00E	C01E

C10TWMD	比较器1的TIMER WINDOW输出模式控制位 ^{注1}
0	比较器1普通输出模式(由C10E控制)
1	比较器1TIMER WINDOW输出模式(由TO02和C10E共同控制)

C10P	VCOUT1输出极性的选择
0	从VCOUT1进行比较器1的输出。
1	从VCOUT1进行比较器1的反相输出。

C10E	VCOUT1引脚输出的允许 ^{注2, 注3}
0	禁止比较器1的VCOUT1引脚输出。
1	允许比较器1的VCOUT1引脚输出。

C11E	比较器1中断请求的允许 ^{注4}
0	禁止比较器1的中断请求。
1	允许比较器1的中断请求。

C00P	VCOUT0输出极性的选择
0	从VCOUT0进行比较器0的输出。
1	从VCOUT0进行比较器0的反相输出。

C00E	VCOUT0引脚输出的允许 ^{注5, 注6}
0	禁止比较器0的VCOUT0引脚输出。
1	允许比较器0的VCOUT0引脚输出 ^{注4, 注8} 。

C01E	比较器0中断请求的允许 ^{注7}
0	禁止比较器0的中断请求。
1	允许比较器0的中断请求。

- 注1. 比较器1使用TIMER WINDOW模式时，必须将寄存器COMPFIR的bit7(C1EDG)置为“1”。C1OE和C1OTWMD位不能同时设置，要先设置C1OTWMD位之后，再将C1OE位置为“1”。
- 注2. 当改写C1OE位时，可能会产生比较器1中断请求和EVENTC事件。请在将EVENTC的ELSELR14寄存器设置为0（不链接到比较器1的输出）后改写此位。此外，在改写C1OE位后请将中断请求标志寄存器的IF位初始化（无中断请求）。
- 注3. 比较器1的结果可输出到除P00以外的任意引脚。选择某引脚输出时，将该引脚的PxxCFG寄存器置为“0x1b，PMxx配置为1'b0，PMCxx配置为1'b0”。
- 注4. 当改写C0OE位时，可能会产生比较器0中断请求和EVENTC事件。请在将EVENTC的ELSELR13寄存器设置为0（不链接到比较器0的输出）后改写此位。此外，在改写C0OE位后请将中断请求标志寄存器的IF位初始化（无中断请求）。
- 注5. 如果将C1IE从“0”（禁止中断请求）改为“1”（允许中断请求），中断请求标志寄存器的IF就可能变为“1”（有中断请求），因此必须在将中断请求标志寄存器的IF清“0”后使用中断。
- 注6. 比较器0的结果可输出到除P00以外的任意引脚，选择某引脚输出时，将该引脚的PxxCFG寄存器置为“0x1b，PMxx配置为1'b0，PMCxx配置为1'b0”。详细内容请参考“第2章 引脚功能 2.3.7”。
- 注7. 如果将C0IE从“0”（禁止中断请求）改为“1”（允许中断请求），中断请求标志寄存器的IF就可能变为“1”（有中断请求），因此必须在将中断请求标志寄存器的IF清“0”后使用中断。

11.3.5 比较器内置基准电压控制寄存器（CVRCTL）

CVRCTL寄存器是设置比较器的内置基准电压允许/停止动作的寄存器。

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

注意在内置基准电压停止动作(CVREi=0)时，改写CVRCTL寄存器的CVRVSi位。

图11-7 比较器内置基准电压控制寄存器（CVRCTL）的格式

地址: 40043843H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
CVRCTL	0	0	CVRE1	CVRVS1	0	0	CVRE0	CVRVS0

CVRE1	内置基准电压1的控制位
0	禁止内置基准电压1的运行
1	允许内置基准电压1的运行

CVRVS1	内置基准电压的ground端选择位
0	内置基准电压的ground端选择Vss
1	内置基准电压的ground端选择AVREFM ^{注1}

CVRE0	内置基准电压0的控制位
0	禁止内置基准电压0的运行
1	允许内置基准电压0的运行

CVRVS0	内置基准电压的电源端选择位
0	内置基准电压的电源端选择VDD
1	内置基准电压的电源端选择AVREFP ^{注2}

注1. P25引脚同时兼用AVREFM和VCIN13，所以P25引脚作为CMP1的输入信号时，禁止将CVRVS1位置“1”。

注2. P26引脚同时兼用AVREFP和VCIN12，所以P26引脚作为CMP1的输入信号时，禁止将CVRVS0位置“1”。

11.3.6 比较器内置基准电压选择寄存器 (CiRVM)

CiRVM寄存器是设置比较器的内置基准电压的寄存器。

在内置基准电压停止动作(CVREi=0)时, 改写CiRVM寄存器。

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

图11-8 比较器内置基准电压选择寄存器i (CiRVM) 的格式

地址: 400438434H(C0RVM),400438435H(C1RVM), 复位后: 00H RW

符号	7	6	5	4	3	2	1	0
CiRVM	CiRVS7	CiRVS6	CiRVS5	CiRVS4	CiRVS3	CiRVS2	CiRVS1	CiRVS0

CiRV S7	CiRV S6	CiRV S5	CiRV S4	CiRV S3	CiRV S2	CiRV S1	CiRV S0	比较器的内置基准电压的设置
0	0	0	0	0	0	0	0	{{(AVREFP或VDD)/256}x0
0	0	0	0	0	0	0	1	{{(AVREFP或VDD)/256}x1
0	0	0	0	0	0	1	0	{{(AVREFP或VDD)/256}x2
0	0	0	0	0	0	1	1	{{(AVREFP或VDD)/256}x3
...								...
1	1	1	1	1	1	0	0	{{(AVREFP或VDD)/256}x252
1	1	1	1	1	1	0	1	{{(AVREFP或VDD)/256}x253
1	1	1	1	1	1	1	0	{{(AVREFP或VDD)/256}x254
1	1	1	1	1	1	1	1	{{(AVREFP或VDD)/256}x255

11.3.7 比较器0的输入信号选择控制寄存器（CMPSEL0）

CMPSEL0寄存器是比较器0的正端，负端的输入信号的选择寄存器。

在比较器0停止动作(C0ENB=0)时，改写CMPSEL0寄存器。

通过8位存储器操作指令设置CMPSEL0寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-9 比较器0的输入信号选择控制寄存器（CMPSEL0）的格式

地址: 4004384AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
CMPSEL0	CMP0SEL	0	0	0	0	0	C0REFS1	C0REFS0

CMP0SEL	比较器0的正端输入信号选择位
0	选择外部引脚(VCIN0引脚)
1	选择PGA0输出信号

C0REFS1	C0REFS0	比较器0的负端输入信号选择位
0	0	选择内置基准电压VREF0
0	1	选择内部基准电压(1.45V)
1	0	选择外部引脚(IVREF0引脚)
1	1	禁止设置

11.3.8 比较器1的输入信号选择控制寄存器（CMPSEL1）

CMPSEL1寄存器是比较器1的正端，负端的输入信号的选择寄存器。

在比较器1停止动作(C1ENB=0)时，改写CMPSEL1寄存器。

通过8位存储器操作指令设置CMPSEL1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图11-10 比较器1的输入信号选择控制寄存器（CMPSEL1）的格式

地址：4004384BH	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
CMPSEL1	CMP1SEL1	CMP1SEL0	0	0	0	C0REFS1	C0REFS1	C0REFS0

CMP1SEL1	CMP1SEL0	比较器1的正端输入信号选择位
0	0	选择外部引脚(VCIN10引脚)
0	1	选择外部引脚(VCIN11引脚)
1	0	选择外部引脚(VCIN12引脚)
1	1	选择外部引脚(VCIN13引脚)

C0REFS2	C0REFS1	C0REFS0	比较器1的负端输入信号选择位
0	0	0	选择内置基准电压VREF1
0	0	1	选择内部基准电压(1.45V)
0	1	0	选择外部引脚(VCIN10引脚)
0	1	1	选择外部引脚(VCIN11引脚)
1	0	0	选择外部引脚(VCIN12引脚)
1	0	1	选择外部引脚(VCIN13引脚)
1	1	0	禁止设置
1	1	1	禁止设置

注意：切换 CMP1 的模拟输入时，为了防止两个输入信号之前的贯通电流，切换间隔必须在 3us 以上。

11.3.9 控制模拟输入引脚端口功能的寄存器

在将VCIN0引脚、VCIN10-VCIN13引脚和VREF0引脚用作比较器的模拟输入时，必须将各端口对应的端口模式寄存器（PMxx）的位和端口模式控制寄存器（PMCxx）的位置“1”。

在使用VCOUT0和VCOUT1功能时，必须设置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）。详细内容请参照“第2章 引脚功能2.3.7”。

11.4 运行说明

比较器0和比较器1能各自独立运行。设置方法和运行相同，CMP0和PGA可以组合起来联动。

比较器的独立运行和联动的设置步骤如表11-3所示。

表11-3 比较器相关寄存器的设置步骤

步骤	寄存器	位	设置值
1	PGACTL	PGAVG0/1/2	选择增益 ^{注3}
2	PGACTL	PVRVS	0 (Vss引脚选择) ^{注3}
3	PGACTL	PGAEN	1 (允许运行) ^{注3}
4	等待PGA的稳定时间 (最小10μs)		
5	COMPSELi	CMP0SEL/CMP1SELi	比较器i正端输入选择
6	COMPSELi	CiREFS	比较器i负端输入选择
7	CiRVM	CiRVSn	设置内置基准电压的值
8	CVRCTL	CVRVSi	选择内置基准电压的电源和地
9	CVRCTL	CVREi	1 (内置基准电压i允许运行)
10	等待基准电压的稳定时间 (最小20μs)		
11	将VCIN0、VCIN1x、IVREF0引脚 (输入), PGAI(输入) ^{注3} 设置为模拟输入功能。		
12	COMPMDR	CIENB	1 (允许运行)
13	等待比较器的稳定时间 (最小3μs)		
14	COMPFIR	CiFCK	使用或者不使用数字滤波器, 选择采样时钟。
		CiEOP、CiEDG	选择中断请求的边沿检测条件 (上升沿、下降沿或者双边沿)。
15	COMPOCR	CiOP、CiOE	设置VCOUTi的输出 (选择极性, 设置允许或者禁止输出)。请参照“11.4.4比较器i的输出 (i=0、1)”。
		CiIE	设置允许或者禁止中断请求的输出。请参照“11.4.4比较器i的输出 (i=0、1)”。
		C1OTWMD	设置比较器1的TIMER WINDOW输出许可/禁止
16	MKxx ^{注1}	MKL	使用中断时: 选择屏蔽中断。
17	IFxx ^{注1}	IFL	使用中断时: 0 (无中断请求: 初始化) ^{注2}

注1 MKxx, IFxx是比较器的中断控制寄存器, 详细内容请参考“第18章 中断功能”。

注2 比较器设置后, 到运行稳定期间, 可能会产生不需要的中断请求, 必须将中断请求标志位初始化。

注3 比较器0和PGA联动时, 必须要设置

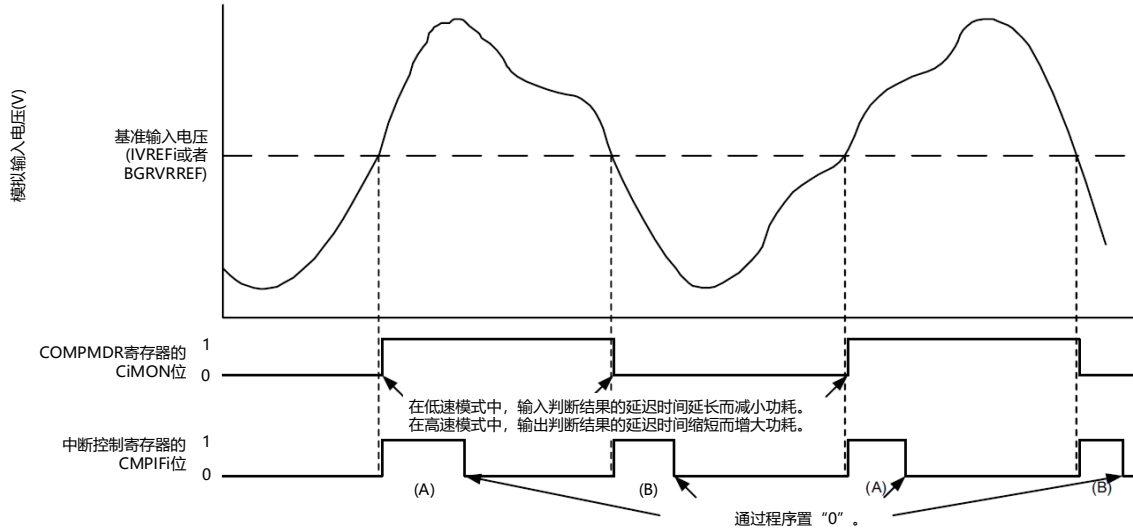
备注 i=0、1, n=0-7, x=0-3

比较器i (i=0、1) 的运行例子如图11-11所示。在基本模式中，当模拟输入电压高于基准输入电压时，COMPMDR寄存器的CiMON位为“1”；当模拟输入电压低于基准输入电压时，CiMON位为“0”。

要使用比较器i中断时，必须将COMPOCR寄存器的CiIE位置“1”（允许中断请求）。此时，如果比较结果发生变化，就产生比较器i的中断请求。有关中断请求的详细内容，请参照“11.4.2 比较器i中断 (i=0、1)”。

图11-11 比较器i (i=0、1) 的运行例子（基本模式）

• 基本模式的运行例子



注意：上图是COMPFIR寄存器的CiFCK1~CiFCK0位为“00B”（无滤波器）并且CiEDG位为“1”（双边沿）的情况（CiEDG位为“0”并且CiEPO位为“0”（上升沿）时的CMPIFi只限于(A)的变化，CiEDG位为“0”并且CiEPO位为“1”（下降沿）时的CMPIFi只限于(B)的变化）。

11.4.1 比较器i的数字滤波器 (i=0、1)

比较器i内置数字滤波器，能通过COMPFIR寄存器的CiFCK1~CiFCK0位选择采样时钟。按各采样时钟对比较器i的输出信号进行采样，在电平3次都相同后的下一个采样时钟，数字滤波器输出此采样值。

图11-12是比较器i的数字滤波器的结果，图11-13是比较器i的数字比较器i (i=0、1) 的数字滤波器和中断运行例子。

图11-12 比较器i (i=0、1) 的数字滤波器和沿检出结构

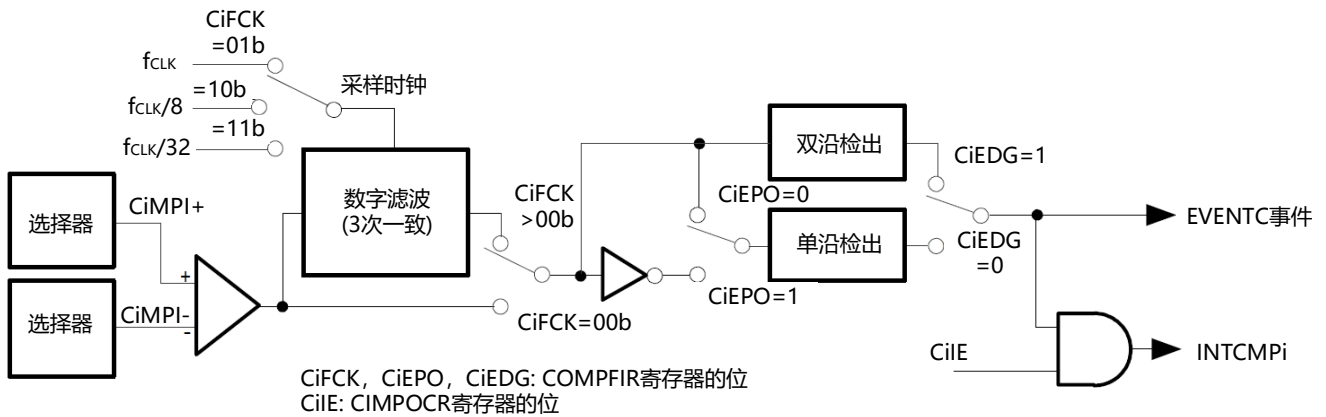
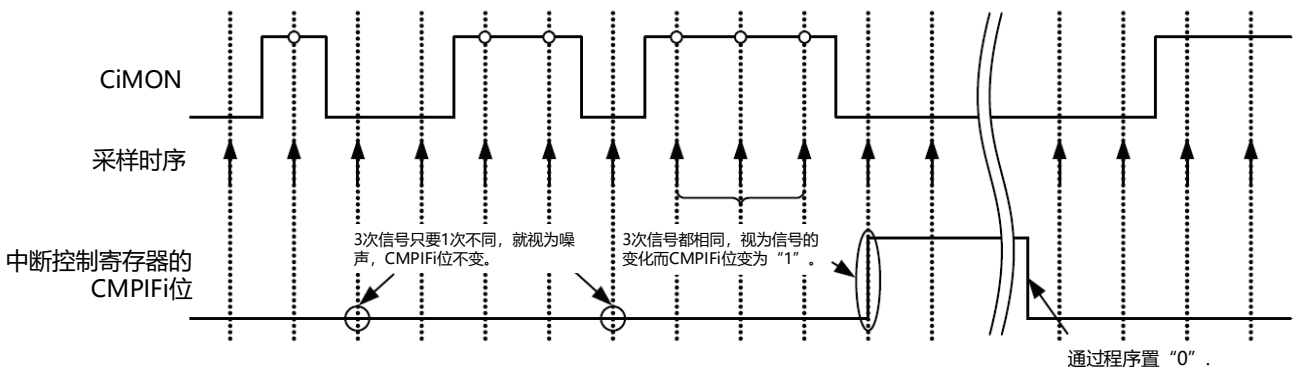


图11-13 比较器i (i=0、1) 的数字滤波器和中断运行例子



注意 上图是COMPFIR寄存器的CiFCK1~CiFCK0位为“01B”、“10B”或者“11B”（有数字滤波器）时的运行例子。

11.4.2 比较器i中断 (i=0、1)

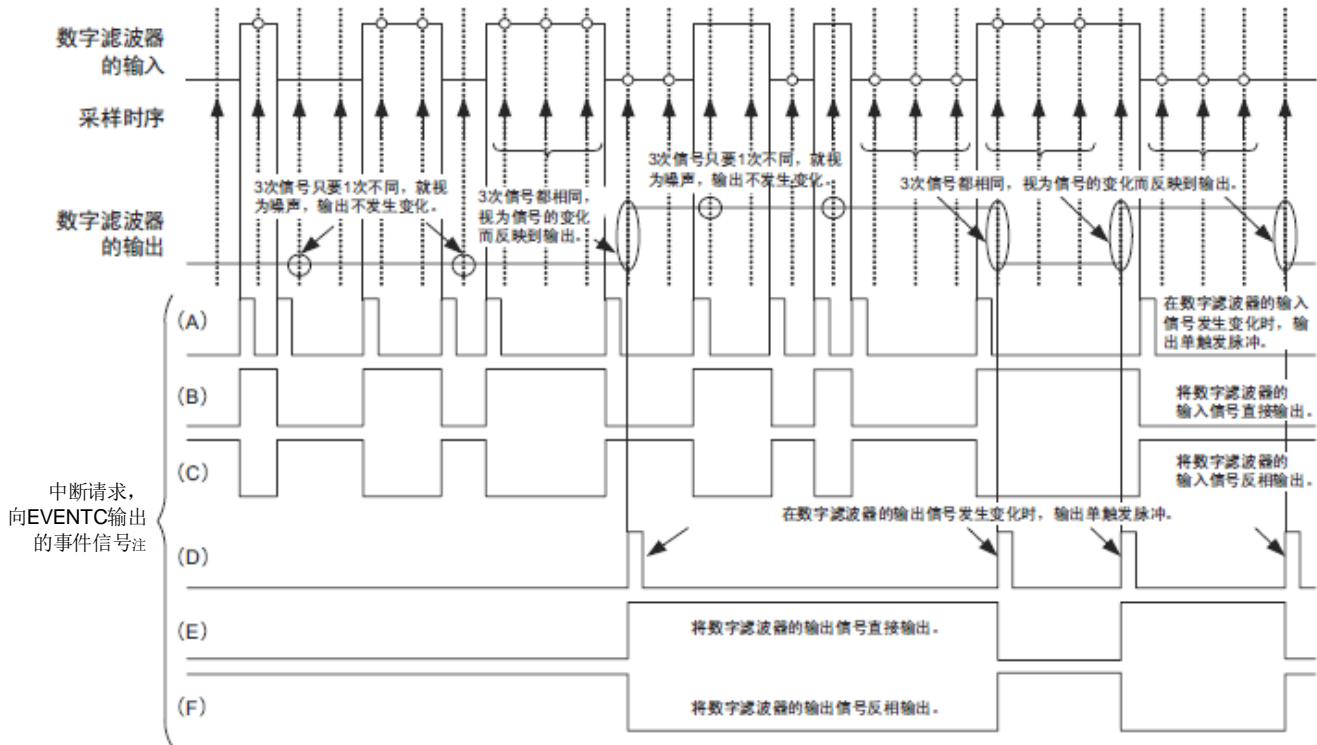
比较器产生比较器0和比较器1共2个中断请求。比较器i中断各有1个优先级指定标志、中断屏蔽标志、中断请求标志和中断向量。

要使用比较器i中断时，必须将COMPOCR寄存器的CiIE位置“1”（允许中断请求的输出）。通过COMPFIR寄存器设置中断请求的产生条件，还能给比较器的输出附加数字滤波器。数字滤波器可选择3种采样时钟。有关寄存器的设置和产生中断请求的对应，请参照“11.3.3 比较器滤波控制寄存器（COMPFIR）”和“11.3.4 比较器输出控制寄存器（COMPOCR）”。

11.4.3 向联动控制器（EVENTC）输出的事件信号

和中断请求的产生条件相同，通过检测COMPFR寄存器设置的数字滤波器的输出边沿，产生向EVENTC输出的事件信号。但是，和中断请求不同的是与COMPOCR寄存器的CiIE位无关，总是向EVENTC输出事件信号。必须通过EVENTC的ELSELR13寄存器和ELSELR14寄存器设置事件输出目标的选择和事件链接的停止。

图11-14 数字滤波器、中断请求和向EVENTC输出事件信号的运行



注 当CiIE位 (i=0、1) 为“1”时，中断请求和向EVENTC输出的事件信号为相同的波形。当CiIE位 (i=0、1) 为“0”时，只有中断请求固定为“0”。

(A)、(B)、(C)的波形是COMPFR寄存器的CiFCK位 (i=0、1) 为“00B”（无数字滤波器）的情况，(D)、(E)、(F)的波形是COMPFR寄存器的CiFCK位 (i=0、1) 为“01B”、“10B”或者“11B”（有数字滤波器）的情况。(A)、(D)是CiEDG位为“1”（双边沿）的情况，(B)、(E)是CiEDG位为“0”并且CiEPO位为“0”（上升沿）的情况，(C)、(F)是CiEDG位为“0”并且CiEPO位为“1”（下降沿）的情况。

11.4.4 比较器i的输出 (i=0、1)

能将比较器的比较结果输出到外部引脚，还能通过COMPOCR寄存器的CiOP位和CiOE位设置输出极性（正相输出或者反相输出）以及是否允许输出。有关寄存器的设置和比较器输出的对应，请参照“11.3.4比较器输出控制寄存器（COMPOCR）”。

要将比较器的比较结果输出到VCOUTi的输出引脚时，必须按照以下的步骤设置端口（在复位后，端口为输入状态）：

- ① 设置比较器的模式（“表11-3 比较器相关寄存器的设置步骤”的步骤2~5）。
- ② 设置比较器的VCOUTi输出（设置COMPOCR寄存器，选择极性并且允许输出）。
- ③ 将VCOUTi的输出引脚对应的端口模式控制寄存器的位置“0”。
- ④ 将VCOUTi的输出引脚对应的端口寄存器的位置“0”。
- ⑤ 将VCOUTi的输出引脚对应的端口方向寄存器设置为输出（从引脚开始输出）。

11.4.5 比较器时钟的停止和提供

在通过设置外围允许寄存器1（PER1）来停止比较器时钟的情况下，必须按照以下步骤进行设置：

- ① 将COMPMDR寄存器的CiENB位置“0”（停止比较器的运行）。
- ② 将中断请求标志寄存器的IF位置“0”（清除比较器停止运行前不需要的中断）。
- ③ 将PER1寄存器的PGACMPEN位置“0”。

如果通过设置PER1寄存器来停止时钟，比较器的内部寄存器就全部被初始化，因此在要再次使用比较器时，必须按照表11-3 比较器相关寄存器的设置步骤的步骤设置寄存器。

注意1.如果将比较器模式设置寄存器（COMPMDR）的比较器n基准电压选择位（CnVRF）置“1”（比较器n基准电压为内部基准电压（1.45V）），就不能通过A/D转换器对温度传感器的输出进行A/D转换。

2.如果在以下某个状态下允许DMA启动，就开始DMA传送，并且在传送结束后产生中断。因此，必须根据需要在确认比较器的监视标志（CnMON）后置为允许DMA启动。

- 设置为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的上升沿产生中断请求（CnEPO=0）而且IVCMP>IVREF（或者内部基准电压1.45V）。
 - 设置为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的下降沿产生中断请求（CnEPO=1）而且IVCMP<IVREF（或者内部基准电压1.45V）。
- (n=0、1)

第12章 可编程增益放大器 (PGA)

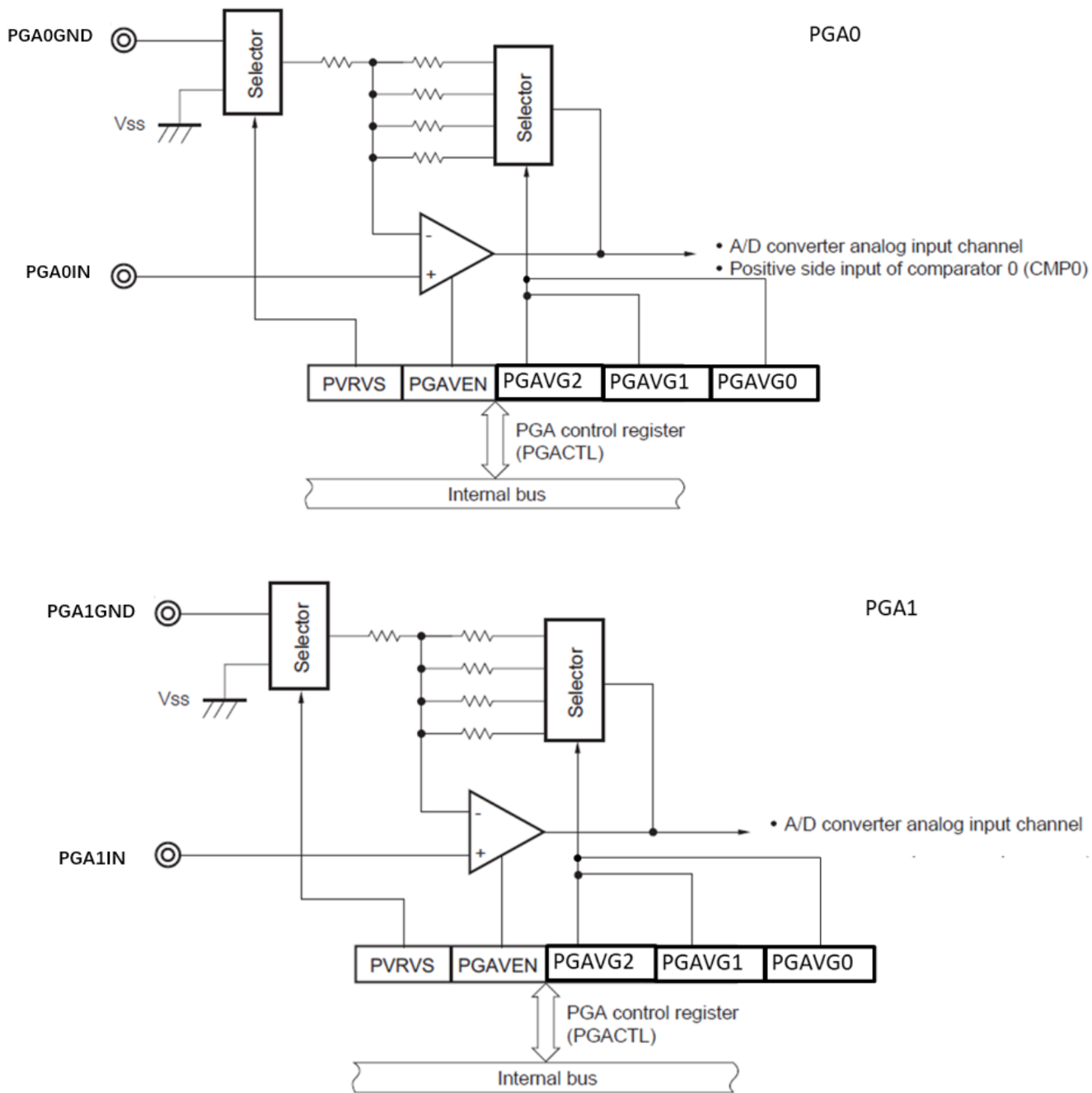
12.1 可编程增益放大器的功能

本产品内置两路可编程增益放大器 (PGA0和PGA1)，具有如下功能：

- 每路PGA的放大增益有7种选择：4倍，8倍，10倍，12倍，14倍，16倍，32倍
- 可选择外部引脚作为PGA负端反馈电阻的地
- PGA0的输出可被选为用于A/D转换器的模拟输入或者比较器0 (CMP0) 正端的模拟输入
- PGA1的输出可被选为用于A/D转换器的模拟输入

12.2 可编程增益放大器的结构

图12-1 可编程增益放大器的框图



12.3 可编程增益放大器的寄存器

表12-1 控制可编程增益放大器的寄存器

寄存器名称	符号
外围允许寄存器1	PER1
可编程增益放大器控制寄存器	PGA0CTL
可编程增益放大器控制寄存器	PGA1CTL
端口模式控制寄存器	PMCxx
端口模式寄存器	PMxx

12.3.1 外围允许寄存器1(PER1)

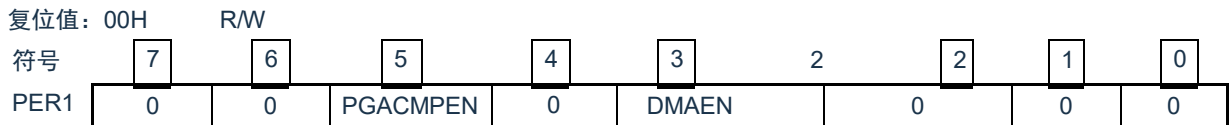
PER1寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用可编程增益放大器时，必须将此寄存器的bit5 (PGACMPEN) 置“1”。

通过8位存储器操作指令设置PER1寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图12-2 外围允许寄存器1 (PER1) 的格式



PGACMPEN	比较器/可编程增益放大器的输入时钟的控制
0	停止提供输入时钟。 比较器或可编程增益放大器的寄存器不可写比较器或可编程增益放大器处于复位状态
1	提供输入时钟。 比较器或可编程增益放大器的寄存器可读写

注意

在配置比较器或可编程增益放大器的寄存器之前，确认PGACMPEN的bit位先要置1。

如果PGACMPEN=0,对比较器或可编程增益放大器的控制寄存器进行写操作是无效的,并且所有的读出值为默认值。(端口模式寄存器 (PMXX) 和端口寄存器PXX除外)

12.3.2 可编程增益放大器控制寄存器 (PGAnCTL)

PGA0CTL和PGA1CTL寄存器用于控制可编程增益放大器开始工作，停止工作和放大倍数。

可通过8位存储器操作指令设置PGA0CTL和PGA1CTL寄存器。在产生复位信号后，此寄存器复位值为00H。

图12-3 PGA控制寄存器 (PGAnCTL) 的格式

复位值: 00H	R/W							
	7	6	5	4	3	2	1	0
PGAnCTL	PGAnEN	-	-	-	PVRVS ^N _{ole}	PGAnV G2	PGAnV G1	PGAnV G0
n=0,1								

PGAnEN	可编程增益放大器运行控制
0	放大器停止工作
1	允许放大器工作

PVRVS	反馈电阻地的选择
0	选择 V _{ss}
1	选择 PGMNGND 引脚

PGAnVG2	PGAnVG1	PGAnVG0	PGAn 增益
0	0	0	4 倍
0	0	1	8 倍
0	1	0	10 倍
0	1	1	12 倍
1	0	0	14 倍
1	0	1	16 倍
1	1	0	32 倍
其他			设置禁止

注意 PGMNGND设置为1后，可编程增益放大器运行需要10us的稳定时间

12.3.3 控制模拟输入引脚端口功能的寄存器

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

12.4 可编程增益放大器的运行

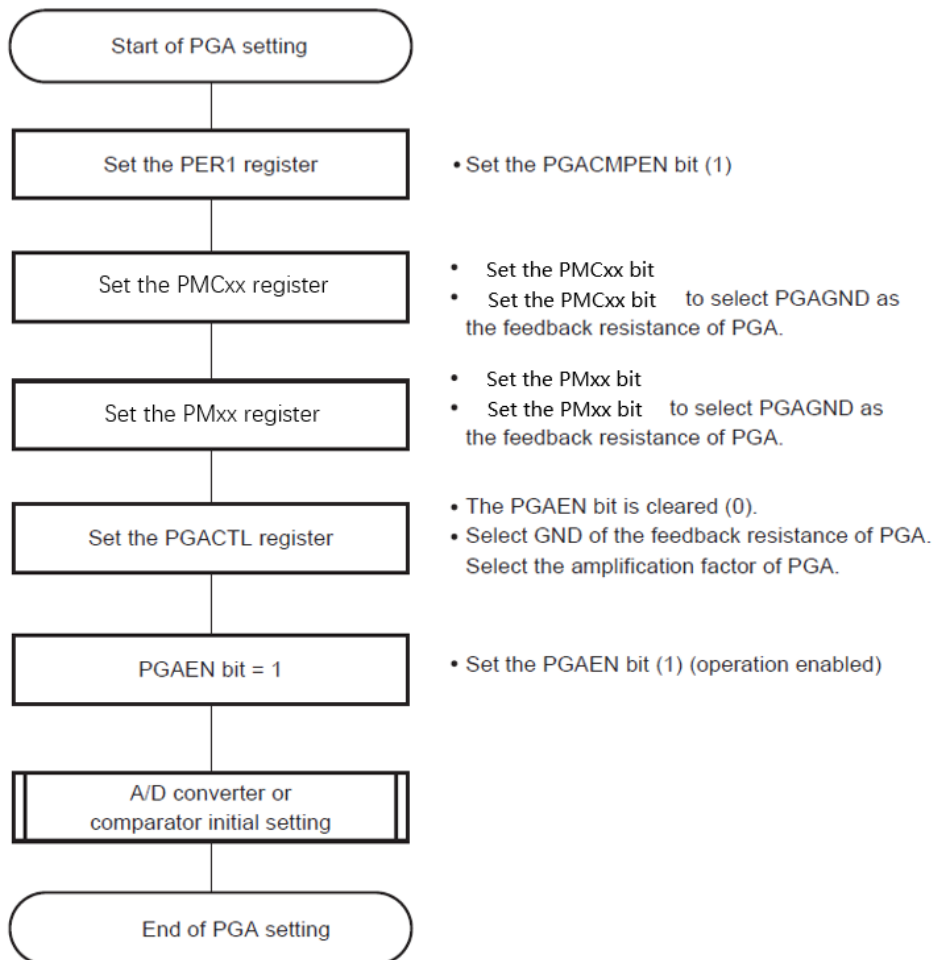
对PGAIN引脚输入的模拟电压进行放大，放大增益有7种选择：4倍，8倍，10倍，12倍，14倍，16倍，32倍。

放大之后的电压可以用于A/D转换器的模拟输入，和比较器0 (CMP0) 的正端输入信号。

可编程增益放大器的开始运行和停止运行的步骤如下。

12.4.1 可编程增益放大器的开始运行步骤

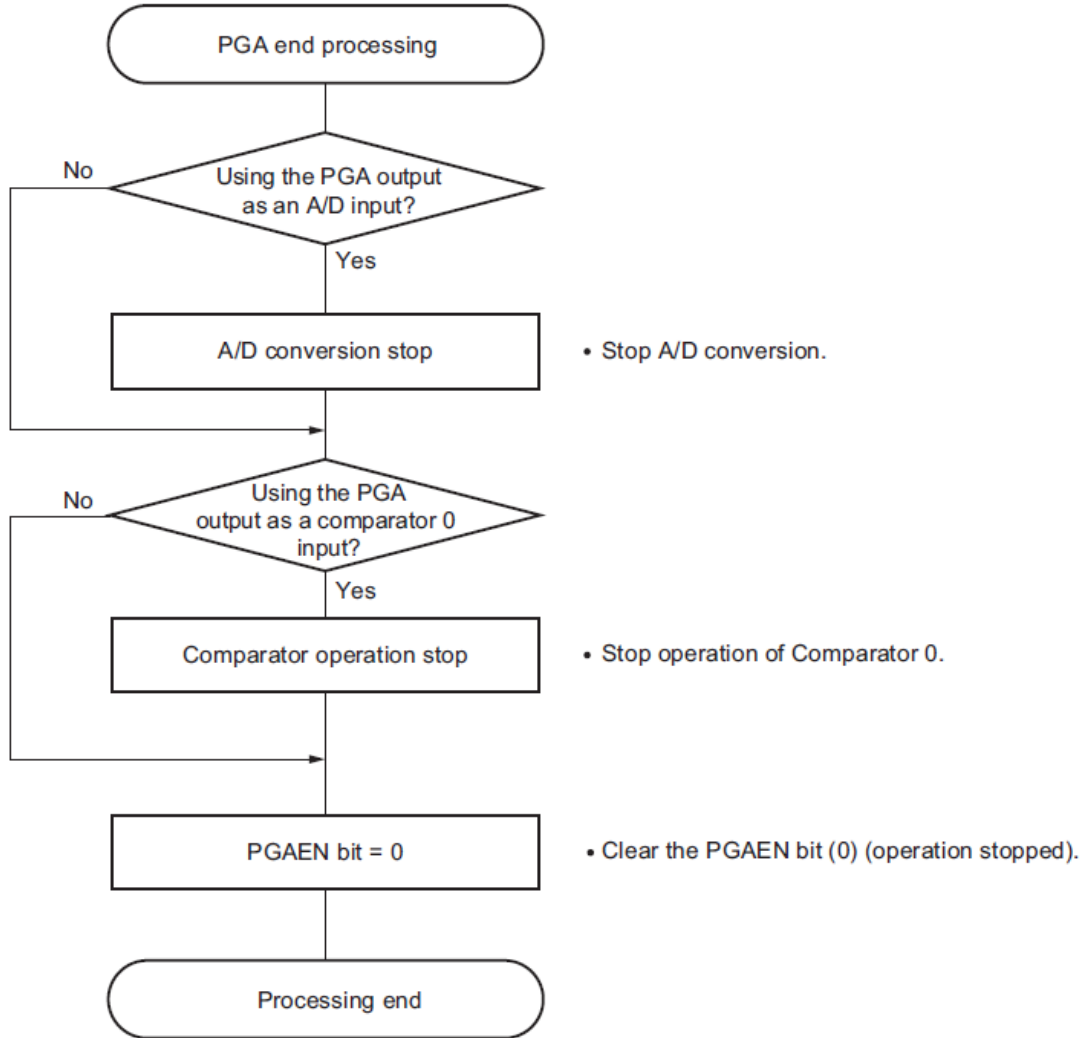
以PGA0为例，设置步骤如下：



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

12.4.2 可编程增益放大器的停止运行步骤

以PGA0为例，设置步骤如下：



注意 1、当重新开始PGA和A/D转换或者放大器时，在设置PGAEN位为1后需要10us的PGA稳定时间。

2、即使停止PGA操作，也可以使用直通的引脚进行A/D转换和比较器动作。

第13章 通用串行通信单元

通用串行通信单元的1个单元有4个串行通道，各通道能实现3线串行（SSPI）、UART的通信功能。本产品支持的各通道的功能分配如下：

单元	通道	用作SSPI	用作UART
0	0	SSPI00 (支持从属选择输入功能)	UART0
	1	SSPI01	
	2	SSPI10	UART1
	3	SSPI11	
1	0	SSPI20	UART2
	1	SSPI21	

在单元0的通道0和通道1使用UART0时，不能使用SSPI00和SSPI01，但是能使用通道2和通道3的SSPI10、UART1。

注意 本章的下述内容主要针对24引脚产品的单元和通道结构进行说明。

13.1 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。

13.1.1 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)

与主控设备输出的串行时钟 (SCLK) 同步进行数据的发送和接收。

这是使用1条串行时钟 (SCLK)、1条发送串行数据 (SDO) 和1条接收串行数据 (SDI) 共3条通信线进行通信的时钟同步通信功能。

有关具体的设置例子, 请参照“13.5 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信的运行”。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注

主控通信: $\text{Max.f}_{\text{CLK}}/2$

从属通信: $\text{Max.f}_{\text{MCK}}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注 必须在满足SCLK周期时间 (t_{KCY}) 特性的范围内使用。详细内容请参照数据手册。

13.1.1.2 UART (UART0~UART2)

这是通过串行数据发送 (TxD) 和串行数据接收 (RxD) 共2条线进行异步通信的功能。使用这2条通信线, 按数据帧 (由起始位、数据、奇偶校验位和停止位构成) 与其他通信方进行异步 (使用内部波特率) 的数据发送和接收。能通过使用发送专用 (偶数通道) 和接收专用 (奇数通道) 共2个通道来实现全双工 UART通信。

有关具体的设置例子, 请参照“13.7 UART (UART0~UART2) 通信的运行”。

[数据的发送和接收]

- 7位、8位或者9位的数据长度注
- MSB/LSB优先的选择
- 发送和接收数据的电平设置、反相的选择
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误或者溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

13.2 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13-1 通用串行通信单元的结构

项目	结构
移位寄存器	8位或者9位注 ¹
缓冲寄存器	串行数据寄存器mn (SDRmn) 的低8位或者9位注 ^{1、2}
串行时钟输入/输出	SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚 (用于3线串行I/O)
串行数据输入	SDI00、SDI01、SDI10、SDI11、SDI20、SDI21引脚 (用于3线串行I/O)、 RxD0、RxD1、RxD2引脚 (用于UART)
串行数据输出	SDO00、SDO01、SDO10、SDO11、SDO20、SDO21引脚 (用于3线串行I/O)、 TxD0、TxD1、TxD2引脚 (用于UART)
从属选择输入	SS00引脚 (用于从属选择输入功能)
控制寄存器	<单元设置部的寄存器> <ul style="list-style-type: none"> •外围允许寄存器0 (PER0) •串行时钟选择寄存器m (SPSm) •串行通道允许状态寄存m (SEm) •串行通道开始寄存器m (SSm) •串行通道停止寄存器m (STm) •串行输出允许寄存器m (SOEm) •串行输出寄存器m (SOM) •串行输出电平寄存器m (SOLm) •串行待机控制寄存器m (SSCm) •输入切换控制寄存器 (ISC) •噪声滤波器允许寄存器0 (NFEN0)
	<各通道部的寄存器> <ul style="list-style-type: none"> •串行数据寄存器mn (SDRmn) •串行模式寄存器mn (SMRmn) •串行通信运行设置寄存器mn (SCRmn) •串行状态寄存器mn (SSRmn) •串行标志清除触发寄存器mn (SIRmn)
	<ul style="list-style-type: none"> •端口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 1.用作移位寄存器和缓冲寄存器的位数因单元和通道而不同。

- mn=00、01: 低9位
- 上述以外: 低8位

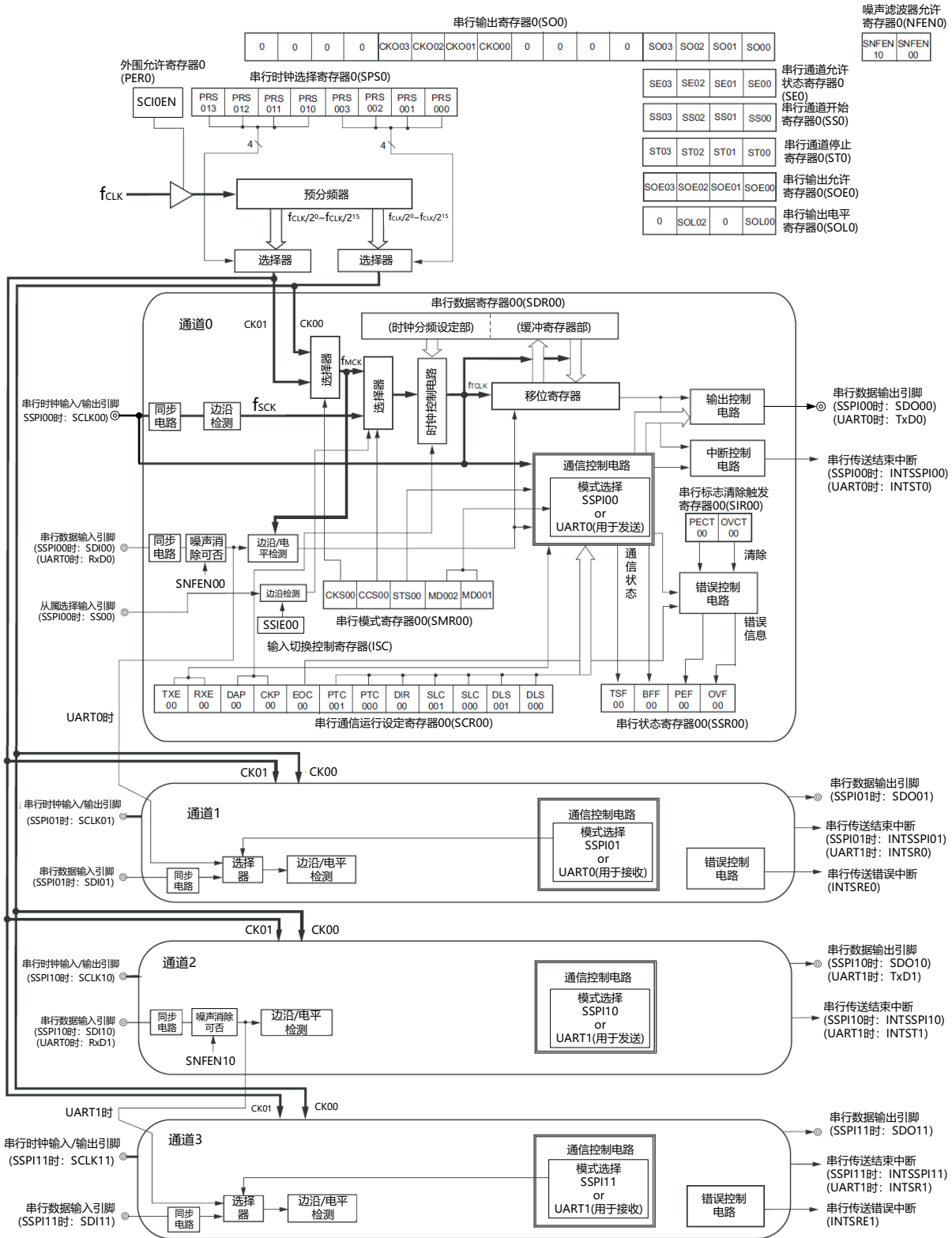
2.根据通信模式,能用以下SFR名称读写串行数据寄存器mn (SDRmn) 的低8位。

- SSPIp通信.....SIOp (SSPIp数据寄存器)
- UARTq接收.....RXDq (UARTq接收数据寄存器)
- UARTq发送.....TXDq (UARTq发送数据寄存器)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
q: UART号 (q=0~2)

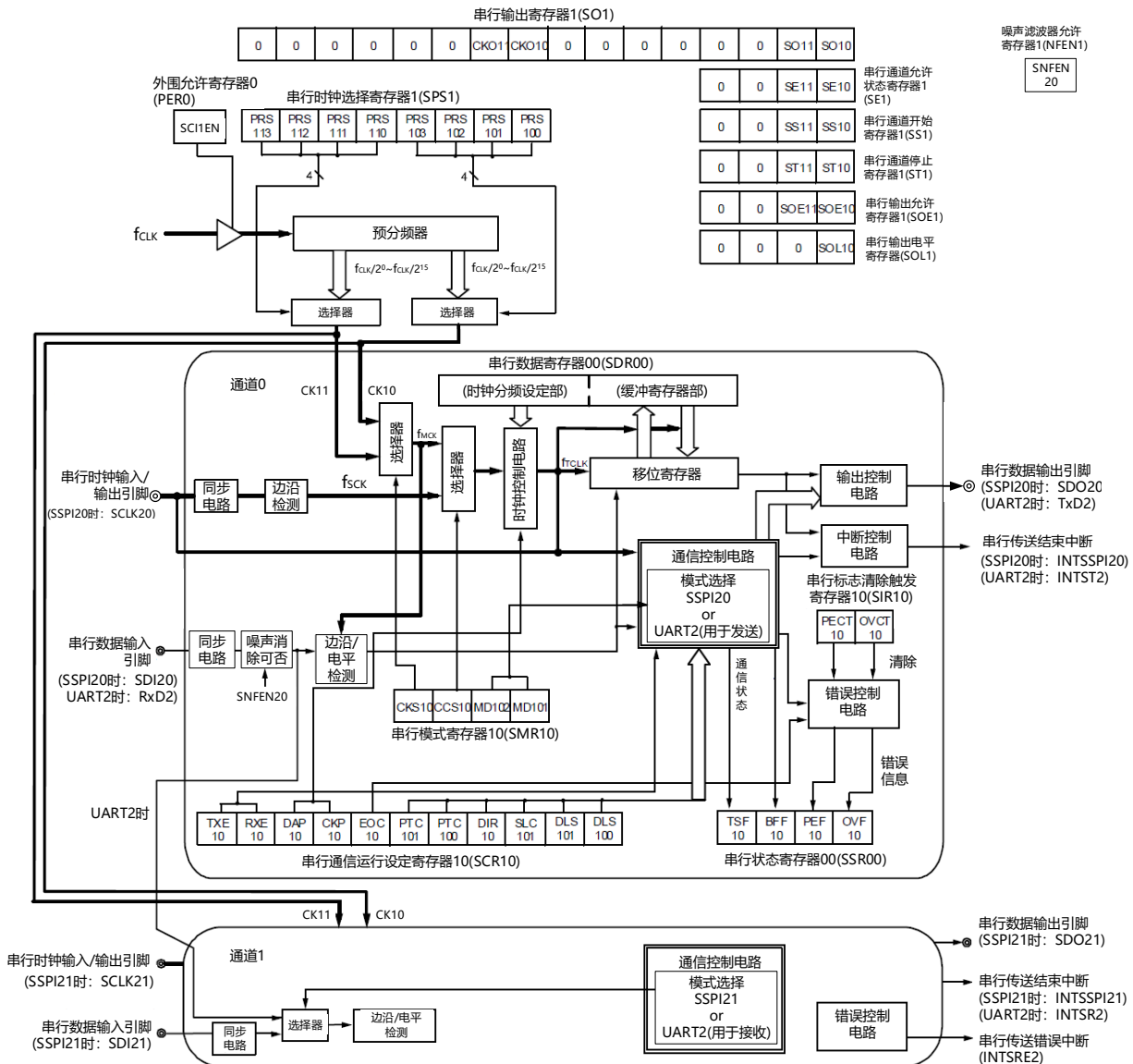
通用串行通信单元0的框图如图13-1所示。

图13-1 通用串行通信单元0的框图



通用串行通信单元1的框图如图13-2所示。

图13-2 通用串行通信单元1的框图

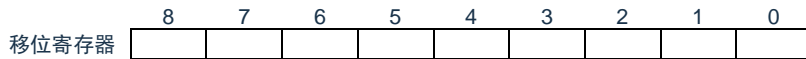


13.2.1 移位寄存器

这是进行并行和串行相互转换的9位寄存器。

在以9位数据长度进行UART通信时，使用9位（bit0~8）注1。在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出注1。不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，使用串行数据寄存器mn（SDRmn）的低8位或者低9位。



13.2.2 串行数据寄存器mn（SDRmn）的低8位或者低9位

SDRmn寄存器是通道n的发送和接收数据寄存器（16位）。

bit8~0（低9位）注或者bit7~0（低8位）用作发送和接收缓冲寄存器，bit15~9用作运行时钟（fMCK）的分频设置寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到低8位或者低9位；在发送数据时，将被传送到移位寄存器的发送数据设置到低8位或者低9位。

与数据的输出顺序无关，根据串行通信运行设置寄存器mn（SCRmn）的bit0和bit1（DLSmn0、DLSmn1）的设置，保存到低8位或者低9位的数据如下所示：

- 7位数据长度（保存在SDRmn寄存器的bit0~6）
- 8位数据长度（保存在SDRmn寄存器的bit0~7）
- 9位数据长度（保存在SDRmn寄存器的bit0~8）注1

能以16位为单位读写SDRmn寄存器。

根据通信模式，能用以下SFR名称，以8位为单位读写SDRmn寄存器的低8位或者低9位注2。

- SSPIp通信.....SDIOp（SSPIp数据寄存器）
- UARTq接收.....RXDq（UARTq接收数据寄存器）
- UARTq发送.....TXDq（UARTq发送数据寄存器）

在产生复位信号后，SDRmn寄存器的值变为“0000H”。

注 1.只有UART0支持9位数据长度。

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

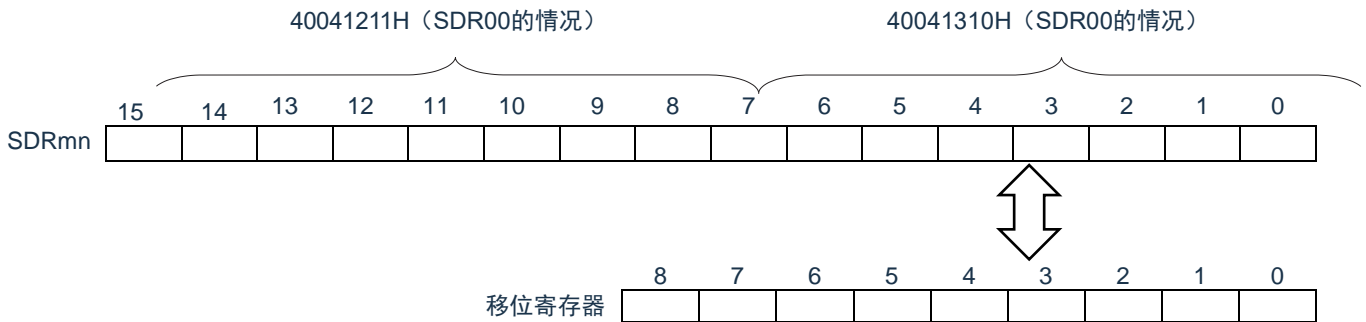
备注1.在接收结束后，bit0~8中超过数据长度的部分的位为“0”。

2.m: 单元号（m=0、1） n: 通道号（n=0~3） p: SSPI号（p=00、01、10、11、20、21）

q: UART号（q=0~2）

图13-3 串行数据寄存器mn (SDRmn) (mn=00、01、10、11) 的格式

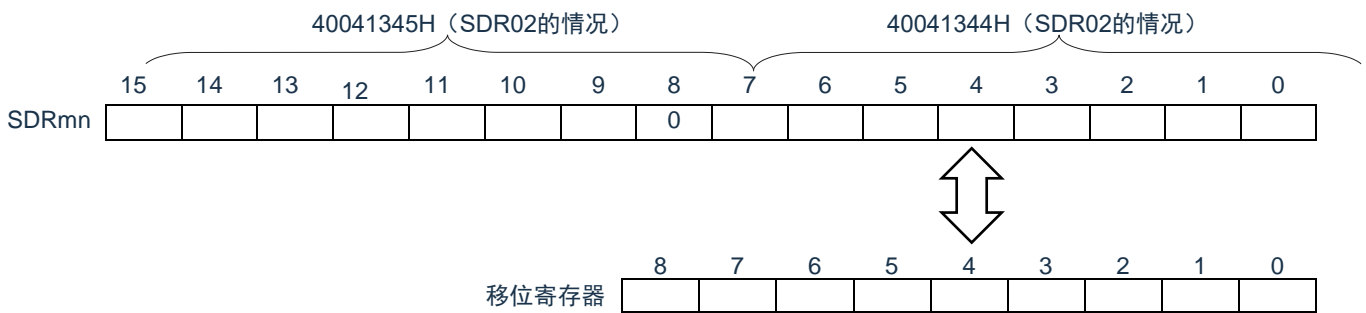
地址: 40041310H (SDR00)、40041312H (SDR01) 复位后: 0000H R/W
 40041748H (SDR10)、4004174AH (SDR11)



备注 有关SDRmn寄存器的高7位的功能, 请参照“13.3.5 串行数据寄存器mn (SDRmn)”。

图13-4 串行数据寄存器mn (SDRmn) (mn=02、03、10、11、12、13) 的格式

地址: 40041344H (SDR02)、40041346H (SDR03) 复位后: 0000H R/W



注意 必须将bit8置“0”。

有关SDRmn寄存器的高7位的功能, 请参照“13.3 控制通用串行通信单元的寄存器”。

13.3 控制通用串行通信单元的寄存器

控制通用串行通信单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 串行时钟选择寄存器m (SPSm)
- 串行模式寄存器mn (SMRmn)
- 串行通信运行设置寄存器mn (SCRmn)
- 串行数据寄存器mn (SDRmn)
- 串行标志清除触发寄存器mn (SDIRmn)
- 串行状态寄存器mn (SSRmn)
- 串行通道开始寄存器m (SSm)
- 串行通道停止寄存器m (STm)
- 串行通道允许状态寄存器m (SEm)
- 串行输出允许寄存器m (SOEm)
- 串行输出电平寄存器m (SOLm)
- 串行输出寄存器m (SOM)
- 串行待机控制寄存器m (SSCm)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器0 (NFEN0)
- 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)
- 端口寄存器 (Px)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.1 外围允许寄存器0 (PER0)

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

要使用通用串行通信单元0时，必须将bit2 (SCI0EN) 置“1”。

要使用通用串行通信单元1时，必须将bit3 (SCI1EN) 置“1”。

通过8位存储器操作指令设置PER0寄存器。在产生复位信号后，PER0寄存器的值变为“00H”。

图13-5 外围允许寄存器0 (PER0) 的格式

地址: 0x40020420 复位后: 00H R/W

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

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

注意1. 要设置通用串行通信单元m时，必须先在SCI_mEN位为“1”的状态下设置以下的寄存器。当SCI_mEN位为“0”时，忽视通用串行通信单元m的控制寄存器的写操作，而且读取值都为初始值（输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器0 (NFEN0)、端口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)。

- 串行时钟选择寄存器m (SPSm)
- 串行模式寄存器mn (SMRmn)
- 串行通信运行设置寄存器mn (SCRmn)
- 串行数据寄存器mn (SDRmn)
- 串行标志清除触发寄存器mn (SIRmn)
- 串行状态寄存器mn (SSRmn)
- 串行通道开始寄存器m (SSm)
- 串行通道停止寄存器m (STm)
- 串行通道允许状态寄存器m (SEm)
- 串行输出允许寄存器m (SOEm)
- 串行输出电平寄存器m (SOLm)
- 串行输出寄存器m (SOM)
- 串行待机控制寄存器m (SSCm)

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

图13-6 串行时钟选择寄存器m (SPSm) 的格式

地址：40041126H (SPS0)、40041566H (SPS1) 复位后：0000H R/W

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

PRSmk3	PRSmk2	PRSmk1	PRSmk0	运行时钟 (CKmk) 的选择 ^注
0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

注 要在通用串行通信单元（SCI）运行过程中更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止SCI的运行（串行通道停止寄存器m (STm) =000FH）后进行更改。

注意 必须将bit15~8置“0”。

备注1. f_{CLK} : CPU/外围硬件的时钟频率

2.m: 单元号 (m=0、1)

3.k=0、1

13.3.3 串行模式寄存器mn (SMRmn)

SMRmn寄存器是设置通道n运行模式的寄存器，进行运行时钟 (f_{MCK}) 的选择、能否使用串行时钟 (f_{SCLK}) 输入的指定、开始触发的设置、运行模式 (SSPI、UART) 的设置以及中断源的选择。另外，只在UART模式中设置接收数据的反相电平。

禁止在运行过程中 (SEmn=1) 改写SMRmn寄存器，但是能在运行过程中改写MDmn0位。

通过16位存储器操作指令设置SMRmn寄存器。在产生复位信号后，SMRmn寄存器的值变为“0020H”。

图13-7 串行模式寄存器mn (SMRmn) 的格式(1/2)

地址: 40041110H (SMR00) ~40041116H (SMR03) 复位后: 0020H R/W

40041550H (SMR10) ~40041552H (SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSm _n 注1	0	SISmn ₀ 注1	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	通道n运行时钟 (f_{MCK}) 的选择
0	SPSm寄存器设置的运行时钟CKm0
1	SPSm寄存器设置的运行时钟CKm1
运行时钟 (f_{MCK}) 用于边沿检测电路。通过设置CCSmn位和SDRmn寄存器的高7位，生成传送时钟 (f_{TCLK})。	

CCSmn	通道n传送时钟 (f_{TCLK}) 的选择
0	CKSmn位指定的运行时钟 f_{MCK} 的分频时钟
1	来自SCLKp引脚的输入时钟 f_{SCLK} (SSPI模式的从属传送)
传送时钟 f_{TCLK} 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当CCSmn位为“0”时，通过SDRmn寄存器的高7位进行运行时钟 (f_{MCK}) 的分频设置。	

STSmn注1	开始触发源的选择
0	只有软件触发有效 (在SSPI、UART发送时选择)。
1	RxDq引脚的有效边沿 (在UART接收时选择)
在将SSm寄存器置“1”后满足上述条件时，开始传送。	

注 1.只限于SMR01、SMR03、SMR11寄存器。

注意 必须将bit13~9、7、4、3 (SMR00、SMR02、SMR10寄存器为bit13~6、4、3) 置“0”，并且将bit5置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
q: UART号 (q=0~2)

图13-7 串行模式寄存器mn (SMRmn) 的格式(2/2)

地址: 40041110H (SMR00) ~40041116H (SMR03) 复位后: 0020H R/W
 40041550H (SMR10) ~40041552H (SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSm _n ^{注1}	0	SISmn ₀ ^{注1}	1	0	0	MDmn2	MDmn1	MDmn0

SISmn ₀ ^{注1}	UART模式中的通道n接收数据的电平反相控制
0	将下降沿检测为起始位。
1	将上升沿检测为起始位。

MDmn2	MDmn1	通道n运行模式的设置
0	0	SSPI模式
0	1	UART模式
1	0	禁止设置。
1	1	禁止设置。

MDmn0	通道n中断源的选择
0	传送结束中断
1	缓冲器空中断 (在数据从SDRmn寄存器传送到移位寄存器时发生)

在连续发送时, 如果MDmn0位为“1”并且SDRmn的数据为空, 就写下一个发送数据。

注 1.只限于SMR01、SMR03、SMR11、寄存器。

注意 必须将bit13~9、7、4、3 (SMR00、SMR02、SMR10寄存器为bit13~6、4、3) 置“0”, 并且将bit5置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
 q: UART号 (q=0~2)

13.3.4 串行通信运行设置寄存器mn (SCRmn)

SCRmn寄存器是通道n的通信运行设置寄存器，设置数据发送和接收模式、数据和时钟相位、是否屏蔽错误信号、奇偶检验位、起始位、停止位和数据长度等。

禁止在运行过程中 (SEmn=1) 改写SCRmn寄存器。

通过16位存储器操作指令设置SCRmn寄存器。

在产生复位信号后，SCRmn寄存器的值变为“0087H”。

图13-8 串行通信运行设置寄存器mn (SCRmn) 的格式(1/2)

地址: 40041118H (SCR00) ~4004111EH (SCR03) 复位后: 0087H R/W
40041558H (SCR10) ~4004155AH (SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1注1	SLCmn0	0	1	DLSmn1注2	DLSmn0

TXEmn	RXEmn	通道n运行模式的设置
0	0	禁止通信。
0	1	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 模式和简易 I²C 模式中，必须将 DAPmn 位和 CKPmn 位都置“0”。

EOCmn	错误中断信号 (INTSREx (x=0~3)) 的屏蔽控制
0	禁止产生错误中断INTSREx (产生INTSRx)。
1	允许产生错误中断INTSREx (在发生错误时不产生INTSRx)。

在SSPI模式中或者在UART发送时，必须将EOCmn位置“0”注3。

- 注 1.只限于SCR00、SCR02、SCR10寄存器。
2.只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。
3.在EOCmn位为“0”并且不使用SSPImn时，有可能产生错误中断INTSREn。

注意 必须将bit3、6、11置“0” (也必须将SCR01、SCR03、SCR11寄存器的bit5置“0”)，并且将bit2置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)

图13-8 串行通信运行设置寄存器mn (SCRmn) 的格式(2/2)

地址: 40041118H (SCR00) ~4004111EH (SCR03) 复位后: 0087H R/W
 40041558H (SCR10) ~4004155AH (SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 ^{注1}	SLC mn0	0	1	DLSm n1 ^{注2}	DLS mn0

PTCmn1	PTCmn0	UART模式中的奇偶校验位的设置	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出奇偶校验 ^{注3} 。	不判断奇偶校验。
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。

在SSPI模式

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 ^{注1}	SLCmn0	UART模式中的停止位的设置
0	0	无停止位
0	1	停止位长度=1位
1	0	停止位长度=2位 (只限于mn=00、02、10)
1	1	禁止设置。

如果选择了传送结束中断, 就在传送完所有停止位后产生中断。
 在UART接收时, 必须设置为1个停止位 (SLCmn1、SLCmn0=0、1)。
 在SSPI模式中, 必须设置为无停止位 (SLCmn1、SLCmn0=0、0)。
 在UART发送时, 必须设置为1位 (SLCmn1、SLCmn0=0、1) 或者2位 (SLCmn1、SLCmn0=1、0)。

DLSmn1 ^{注2}	DLSmn0	SSPI和UART模式中的数据长度的设置
0	1	9位数据长度 (保存在SDRmn寄存器的bit0~8) (只在UART模式中可选择)
1	0	7位数据长度 (保存在SDRmn寄存器的bit0~6)
1	1	8位数据长度 (保存在SDRmn寄存器的bit0~7)
其他		禁止设置。

- 注 1.只限于SCR00、SCR02、SCR10寄存器。
 2.只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。
 3.与数据的内容无关, 总是附加“0”。

注意 必须将bit3、6、11置“0” (也必须将SCR01、SCR03、SCR11寄存器的bit5置“0”), 并且将bit2置“1”。
 备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)

13.3.5 串行数据寄存器mn (SDRmn)

SDRmn寄存器是通道n发送和接收的数据寄存器（16位）。

SDR00、SDR01、SDR10、SDR11的bit8~0（低9位）或者SDR02、SDR03、SDR10、SDR11的bit7~0（低8位）用作发送和接收缓冲寄存器，bit15~9（高7位）用作运行时钟（fMCK）的分频设置寄存器。

如果将串行模式寄存器mn（SMRmn）的CCSmn位置“0”，由SDRmn寄存器的bit15~9（高7位）设置的运行时钟的分频时钟就用作传送时钟。

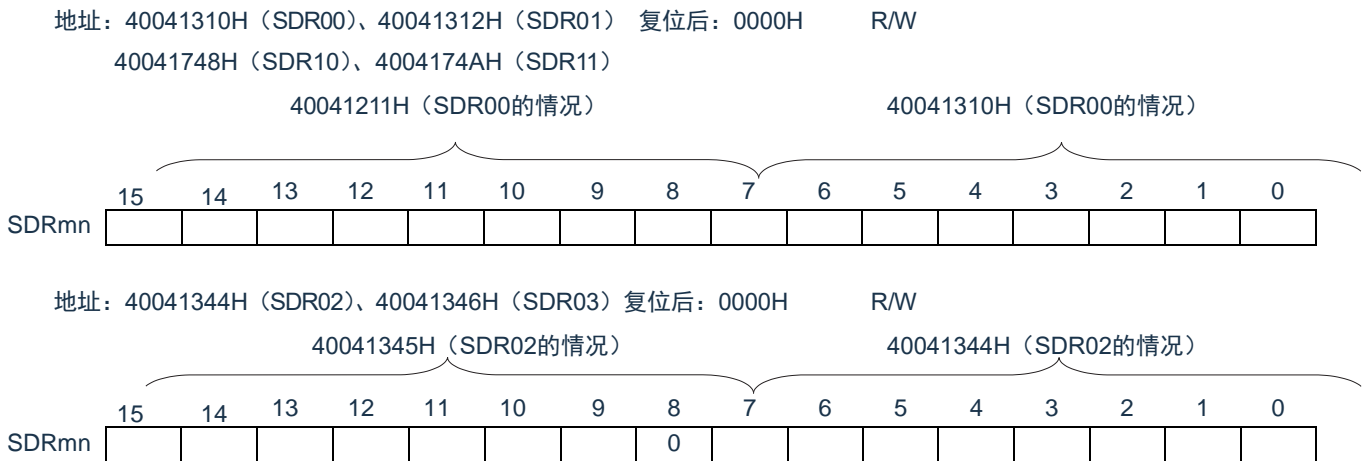
如果将CCSmn位置“1”，就必须将SDR00、SDR01、SDR10、SDR11的bit15~9（高7位）置“000000B”。SCLKp引脚的输入时钟fSCLK（SSPI模式的从属传送）为传送时钟。

SDRmn寄存器的低8位或者低9位用作发送和接收缓冲寄存器。在接收数据时，将移位寄存器转换的并行数据保存到低8位或者低9位；在发送数据时，将被传送到移位寄存器的发送数据设置到低8位或者低9位。

能以16位为单位读写SDRmn寄存器。但是，只有在运行停止（SEmn=0）时才能读写高7位。在运行中（SEmn=1）只能写SDRmn寄存器的低8位或者低9位，而且SDRmn寄存器的高7位的读取值总是“0”。

在产生复位信号后，SDRmn寄存器的值变为“0000H”。

图13-9 串行数据寄存器mn (SDRmn) 的格式



SDRmn[15:9]							运行时钟分频的传送时钟设置
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

注意1. 必须将SDR02、SDR03、SDR10、SDR11寄存器的bit8置“0”。

2. 在使用UART时，禁止将SDRmn[15:9]设置为“0000000B”和“0000001B”。

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

备注1. 有关SDRmn寄存器的低8位或者低9位的功能，请参照“13.2 通用串行通信单元的结构”。

2.m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.6 串行标志清除触发寄存器mn (SIRmn)

这是用于清除通道n各错误标志的触发寄存器。

如果将各位 (FECTmn、PECTmn、OVCTmn) 置“1”，就将串行状态寄存器mn (SSRmn) 的对应位 (FEFmn、PEFmn、OVFmn) 清“0”。因为SDIRmn寄存器是触发寄存器，所以如果清除SSRmn寄存器的对应位，也会立即清除SDIRmn寄存器。

通过16位存储器操作指令设置SIRmn寄存器。能用SIRmnL并且通过8位存储器操作指令设置SIRmn寄存器的低8位。在产生复位信号后，SIRmn寄存器的值变为“0000H”。

图13-10 串行标志清除触发寄存器mn (SIRmn) 的格式

地址：40041108H (SIR00) ~4004110EH (SIR03) 复位后：0000H R/W
 40041548H (SIR10) ~4004154AH (SIR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn ^{注1}	PEC Tmn	OVC Tmn

FECTmn ^{注1}	通道n帧错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的FEFmn位清“0”。

PECTmn	通道n奇偶校验错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的PEFmn位清“0”。

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

注 1.只限于SIR01、SIR03、SIR11寄存器。

注意 必须将bit15~3 (SIR00、SIR02、SIR10寄存器为bit15~2) 置“0”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3)

2.SIRmn寄存器的读取值总是“0000H”。

13.3.7 串行状态寄存器mn (SSRmn)

SSRmn寄存器表示通道n的通信状态和发生错误的情况。表示的错误为帧错误、奇偶校验错误和溢出错误。

通过16位存储器操作指令读取SSRmn寄存器。能用SSRmnL并且通过8位存储器操作指令读取SSRmn寄存器的低8位。在产生复位信号后，SSRmn寄存器的值变为“0000H”。

图13-11 串行状态寄存器mn (SSRmn) 的格式(1/2)

地址：40041100H (SSR00) ~40041106H (SSR03) 复位后：0000HR

40041540H (SSR10) ~40041542H (SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{注1}	PEF mn	OVF mn

TSFmn	通道n通信状态的表示标志
0	通信停止状态或者通信待机状态
1	通信运行状态
[清除条件]	
<ul style="list-style-type: none"> •当将STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信待机状态）时 •当通信结束时 	
[置位条件]	
<ul style="list-style-type: none"> •当开始通信时 	

BFFmn	通道n缓冲寄存器的状态表示标志
0	SDRmn寄存器没有保存有效数据。
1	SDRmn寄存器保存了有效数据。
[清除条件]	
<ul style="list-style-type: none"> •在发送过程中传送完从SDRmn寄存器到移位寄存器的发送数据时 •在接收过程中从SDRmn寄存器读完接收数据时 •当将STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信允许状态）时 	
[置位条件]	
<ul style="list-style-type: none"> •在SCRmn寄存器的TXEmn位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给SDRmn寄存器写发送数据时 •在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到SDRmn寄存器时 •当发生接收错误时 	

注 1.只限于SSR01、SSR03、SSR11寄存器。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

图13-11 串行状态寄存器mn (SSRmn) 的格式(2/2)

地址: 40041100H (SSR00) ~40041106H (SSR03)

复位后: 0000H R

40041540H (SSR10) ~40041542H (SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn ^{注1}	PEF mn	OVF mn

FEFmn ^{注1}	通道n帧错误的检测标志
0	没有发生错误。
1	发生错误 (UART接收时)。
[清除条件] •当给SIRmn寄存器的FECTmn位写“1”时 [置位条件] •在UART接收结束时没有检测到停止位时	

PEFmn	通道n奇偶校验错误的检测标志
0	没有发生错误。
1	发生错误 (UART接收时)。
[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、SSR03、SSR11寄存器。

注意1.如果在BFFmn位为“1”时写SDRmn寄存器, 就会破坏被保存的发送或者接收数据, 并且检测到溢出错误 (OVEmn=1)。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.8 串行通道开始寄存器m (SSm)

SSm寄存器是设置允许各通道的通信/开始计数的触发寄存器。

如果给各位 (SSmn) 写“1”，就将串行通道允许状态寄存器m (SEm) 的对应位 (SEmn) 置“1” (运行允许状态)。因为SSmn位是触发位，所以如果SEmn位为“1”就立即清除SSmn位。

通过16位存储器操作指令设置SSm寄存器。能用SSmL并且通过8位存储器操作指令设置SSm寄存器的低8位。在产生复位信号后，SSm寄存器的值变为“0000H”。

图13-12 串行通道开始寄存器m (SSm) 的格式

地址: 40041122H (SS0) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

地址: 40041562H (SS1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	通道n运行开始的触发
0	没有触发。
1	将SEmn位置“1”，转移到通信待机状态 ^注 。

注 如果在通信过程中将SSmn位置“1”，就停止通信进入待机状态。此时，控制寄存器和移位寄存器的值、SCLKmn引脚和SDOmn引脚、FEFmn标志、PEFmn标志和OVFmn标志保持状态。

注意1. 必须将SS0寄存器的bit15~4、SS1寄存器的bit15~2置“0”。

2. 在UART接收时，必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个 f_{MCK} 时钟，然后将SSmn置“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3)

2.SSm寄存器的读取值总是“0000H”。

13.3.9 串行通道停止寄存器m (STm)

STm寄存器是设置允许各通道的通信/停止计数的触发寄存器。

如果给各位 (STmn) 写“1”，就将串行通道允许状态寄存器m (SEm) 的对应位 (SEmn) 清“0” (运行停止状态)。因为STmn位是触发位，所以如果SEmn位为“0”就立即清除STmn位。

通过16位存储器操作指令设置STm寄存器。能用STmL并且通过8位存储器操作指令设置STm寄存器的低8位。在产生复位信号后，STm寄存器的值变为“0000H”。

图13-13 串行通道停止寄存器m (STm) 的格式

地址: 40041124H (ST0)	复位后: 0000H	R/W																		
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

地址: 40041564H (ST1)	复位后: 0000H	R/W																
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10

STmn	通道n运行的停止触发
0	没有触发。
1	将SEmn位清“0”，停止通信运行注。

注 控制寄存器和移位寄存器的值、SCLKmn引脚和SDOmn引脚以及FEFmn标志、PEFmn标志和OVFmn标志保持状态。

注意 必须将ST0寄存器的bit15~4、ST1寄存器的bit15~2置“0”。

备注 1.m: 单元号 (m=0、1) n: 通道号 (n=0~3)
2.STm寄存器的读取值总是“0000H”。

13.3.10 串行通道允许状态寄存器m (SEm)

SEm寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

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

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

对于停止运行的通道n，能通过软件设置SOm寄存器的CKOmn位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令读取SEm寄存器。能用SEmL并且通过8位存储器操作指令读取SEm寄存器的低8位。在产生复位信号后，SEm寄存器的值变为“0000H”。

图13-14 串行通道允许状态寄存器m (SEm) 的格式

地址: 40041120H (SE0) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

地址: 40041560H (SE1) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE10

SEmn	通道n运行的允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.11 串行输出允许寄存器m (SOEm)

SOEm寄存器设置允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道n，无法通过软件改写后述的串行输出寄存器m (SOm)的SOmn位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道n，能通过软件设置SOm寄存器的SOmn位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过16位存储器操作指令设置SOEm寄存器。能用SOEmL并且通过8位存储器操作指令设置SOEm寄存器的低8位。在产生复位信号后，SOEm寄存器的值变为“0000H”。

图13-15 串行输出允许寄存器m (SOEm) 的格式

地址: 4004112AH	复位后: 0000H	R/W																	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00		

地址: 4004156AH	复位后: 0000H	R/W																	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE11	SOE10		

SOE mn	通道n串行输出的允许或者停止															
0	停止串行通信的输出。															
1	允许串行通信的输出。															

注意 必须将SOE0寄存器的bit15~4、SOE1寄存器的bit15~2置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.12 串行输出寄存器m (SOm)

SOm寄存器是各通道串行输出的缓冲寄存器。

从通道n的串行数据输出引脚输出此寄存器的SOmn位的值。

从通道n的串行时钟输出引脚输出此寄存器的CKOmn位的值。

只有在禁止串行输出时 (SOEmn=0) 才能通过软件改写此寄存器的SOmn位。当允许串行输出 (SOEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的SOmn位的值。

只有在通道停止运行时 (SEmn=0) 才能通过软件改写此寄存器的CKOmn位。当允许通道运行 (SEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的CKOmn位的值。

要将串行接口的引脚用作端口功能等非串行接口功能时, 必须将相应的CKOmn位和SOmn位置“1”。

通过16位存储器操作指令设置SOm寄存器。在产生复位信号后, SOm寄存器的值变为“0F0FH”。

图13-16 串行输出寄存器m (SOm) 的格式

地址: 40041128H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

地址: 40041568H 复位后: 0303H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	0	0	CKO 11	CKO 10	0	0	0	0	0	0	SO 11	SO 10

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

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

注意 必须将SO0寄存器的bit15~12和bit7~4置“0”。
必须将SO1寄存器的bit15~10和bit7~2置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

13.3.13 串行输出电平寄存器m (SOLm)

SOLm寄存器是设置各通道的数据输出电平反相的寄存器。

只有在UART模式中才能设置此寄存器。在SSPI模式中，必须将对应位置“0”。只在允许串行输出时 (SOEmn=1)，将此寄存器的各通道n反相设置反映到引脚输出。在禁止串行输出时 (SOEmn=0)，将SOMn位的值直接输出。禁止在运行过程中 (SEmn=1) 改写SOLm寄存器。

通过16位存储器操作指令设置SOLm寄存器。能用SOLmL并且通过8位存储器操作指令设置SDOLm寄存器的低8位。在产生复位信号后，SOLm寄存器的值变为“0000H”。

图13-17 串行输出电平寄存器m (SOLm) 的格式

地址: 40041134H	复位后: 0000H															R/W
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

地址: 40041574H	复位后: 0000H															R/W
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 10

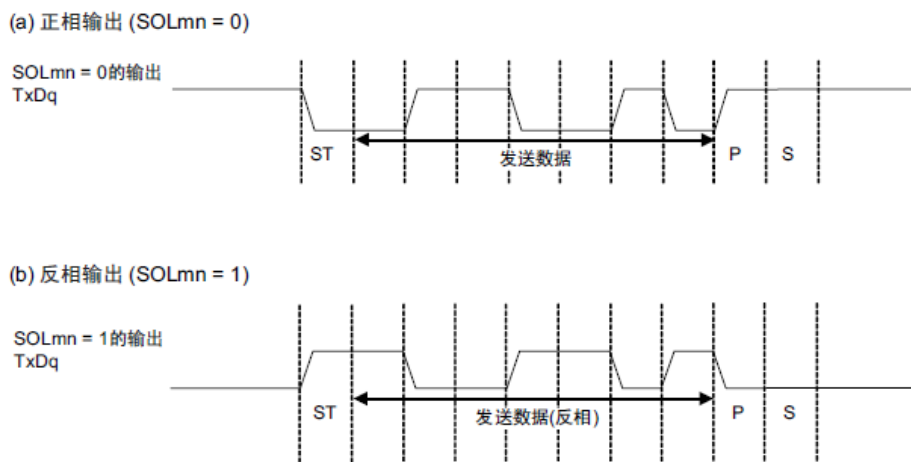
SOLmn	UART模式中的通道n发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

注意 必须将SOL0寄存器的bit15~3和bit1、SOL1寄存器的bit15~1置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2)

当进行UART发送时，发送数据的电平反相例子如图13-18所示。

图13-18 发送数据的电平反相例子



备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2)

13.3.14 输入切换控制寄存器 (ISC)

SS1E00位在SSPI00通信的从属模式中控制通道0的SS00引脚输入。在给SS00引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收；在给SS00引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设置进行发送和接收。

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

图13-19 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	0	0

SSDIE00	SSPI00通信的从属模式中通道0的SS00输入的设置
0	SS00引脚输入无效。
1	SS00引脚输入有效。

注意 必须将bit6~0置“0”。

13.3.15 噪声滤波器允许寄存器0 (NFEN0)

NFEN0寄存器设置噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于SSPI的引脚，必须将对应位置“0”，使噪声滤波器无效。

对于用于UART通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟 (f_{MCK}) 进行同步后检测2个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟 (f_{MCK}) 进行同步。

通过8位存储器操作指令设置NFEN0寄存器。在产生复位信号后，NFEN0寄存器的值变为“00H”。

图13-20 噪声滤波器允许寄存器0 (NFEN0) 的格式

地址: 40040470H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	RxD2引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD2引脚时，必须将SNFEN20位置“1”。 当用作RxD2引脚以外的功能时，必须将SNFEN20位置“0”。	

SNFEN10	RxD1引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD1引脚时，必须将SNFEN10位置“1”。 当用作RxD1引脚以外的功能时，必须将SNFEN10位置“0”。	

SNFEN00	RxD0引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD0引脚时，必须将SNFEN00位置“1”。 当用作RxD0引脚以外的功能时，必须将SNFEN00位置“0”。	

注意 必须将bit7~5、3、1置“0”。

13.3.16 控制串行输入/输出引脚端口功能的寄存器

在使用通用串行通信单元时，必须设置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”。

（例）P22用作串行数据输出的情况：

将端口模式控制寄存器2的PMC22位置“0”。

将端口模式寄存器2的PM22位置“0”。

将端口复用功能配置寄存器P22CFG置“0x2c”。

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”。

（例）P23用作串行数据输入的情况：

将端口模式控制寄存器2的PMC23位置“0”。

将端口模式寄存器2的PM23位置“1”。

将端口复用功能配置寄存器P23CFG置“0x26”。

13.4 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。

13.4.1 以单元为单位停止运行的情况

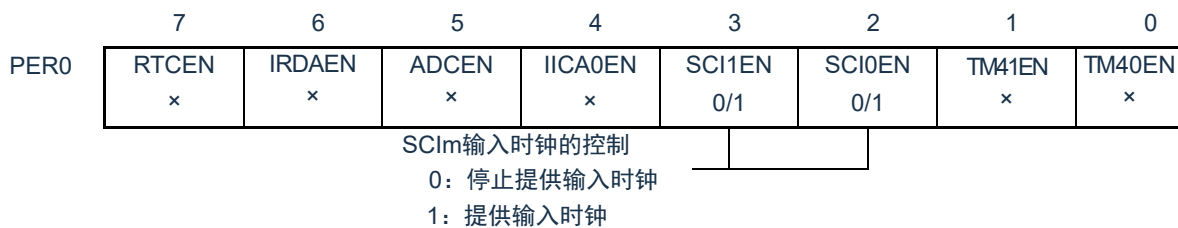
通过外围允许寄存器0（PER0）设置以单元为单位的停止运行。

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

要停止通用串行通信单元0时，必须将bit2（SCI0EN）置“0”；要停止通用串行通信单元1时，必须将bit3（SCI1EN）置“0”。

图13-21 以单元为单位停止运行时的外围允许寄存器0（PER0）的设置

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



注意1.当SCI_mEN位为“0”时，忽视通用串行通信单元m的控制寄存器的写操作，而且读取值都为初始值。但是，以下的寄存器除外：

- 输入切换控制寄存器（ISC）
- 噪声滤波器允许寄存器0（NFEN0）
- 端口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）
- 端口寄存器（Px）

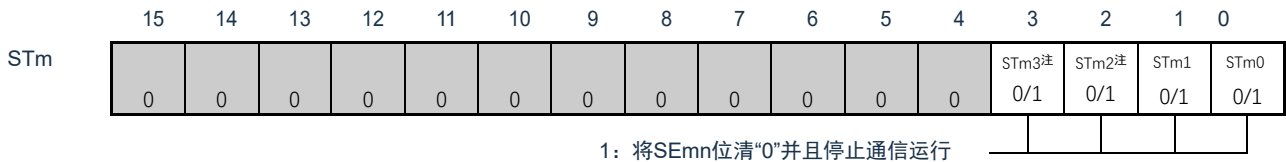
备注 x：这是通用串行通信单元未使用的位（取决于其他外围功能的设置）。
0/1：根据用户的用途置“0”或者“1”。

13.4.2 按通道停止运行的情况

通过以下各寄存器设置按通道停止运行。

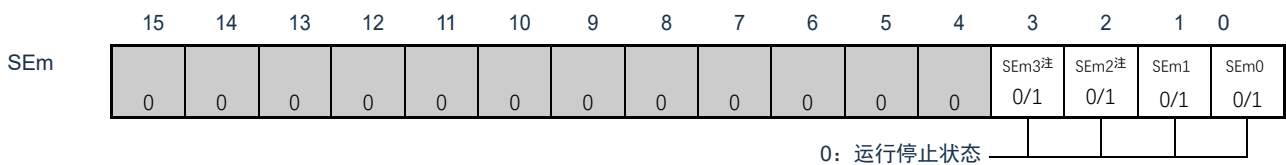
图13-22 按通道停止运行时的各寄存器的设置

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



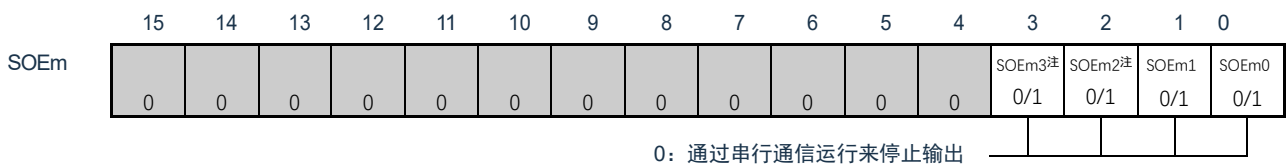
※因为STmn位是触发位，所以如果SEmn位为“0”就立即清除STmn位。

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



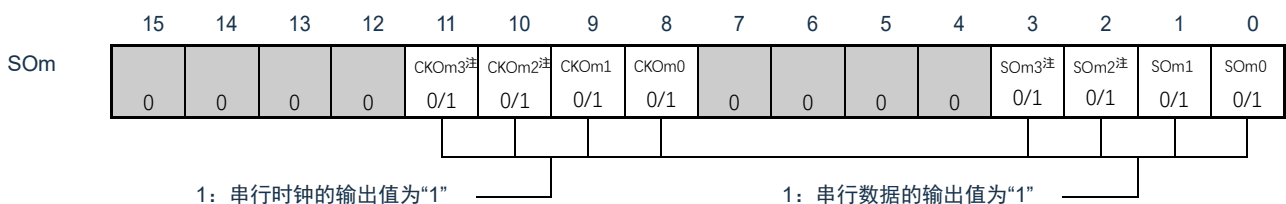
※SEm寄存器是只读状态寄存器，通过STm寄存器来停止运行。对于已经停止运行的通道，能通过软件设置SOM寄存器的CKOmn位的值。

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



※对于已经停止串行输出的通道，能通过软件设置SOM寄存器的SOmn位的值。

(d)串行输出寄存器m (SOM)这是各通道串行输出的缓冲寄存器。



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

注 只限于通用串行通信单元0。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3)

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

13.5 3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的运行

这是通过串行时钟（SCLK）和串行数据（SDI和SDO）共3条线实现的时钟同步通信功能。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注

主控通信：Max.f_{CLK}/2（只限于SSPI00）

主控通信：Max.f_{CLK}/4

从属通信：Max.f_{MCK}/6

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注 必须在满足SCLK周期时间（t_{KCY}）特性的范围内使用。详细内容请参照数据手册。

SCI0的通道0~3以及SCI1的通道0~1是支持3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）的通道。

3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）有以下6种通信运行：

- 主控发送（参照13.5.1）
- 主控接收（参照13.5.2）
- 主控的发送和接收（参照13.5.3）
- 从属发送（参照13.5.4）
- 从属接收（参照13.5.5）
- 从属的发送和接收（参照13.5.6）

13.5.1 主控发送

主控发送是指本产品输出传送时钟并且将数据发送到其他设备的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00、 SDO00	SCLK01、 SDO01	SCLK10、 SDO10	SCLK11、 SDO11	SCLK20、 SDO20	SCLK21、 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	无					
传送数据长度	7位或者8位					
传送速率 ^注	Max. $f_{CLK}/2$ [Hz]（只限于SSPI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \cdot 2^{15} \cdot 128)$ [Hz] f_{CLK} : 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0: 在串行时钟开始运行时，开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

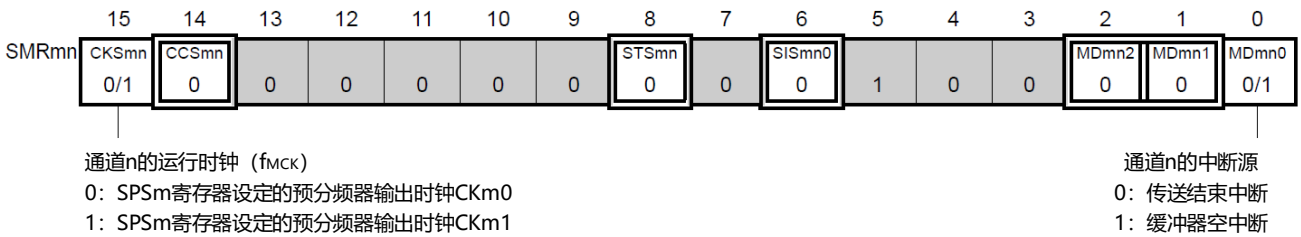
注 必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

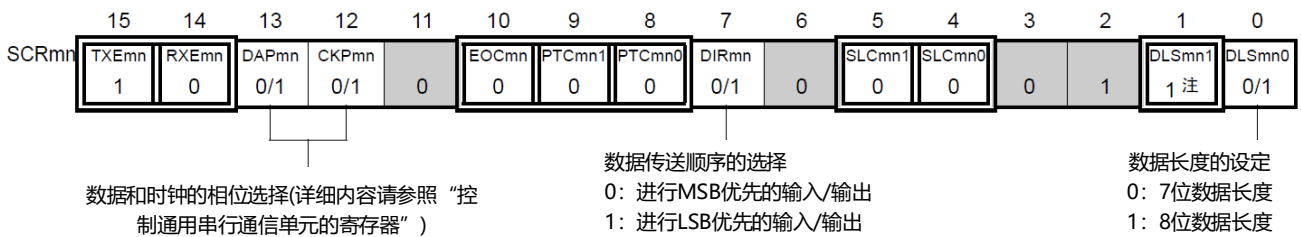
(1) 寄存器的设置

图13-23 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控发送时的寄存器设置内容例子

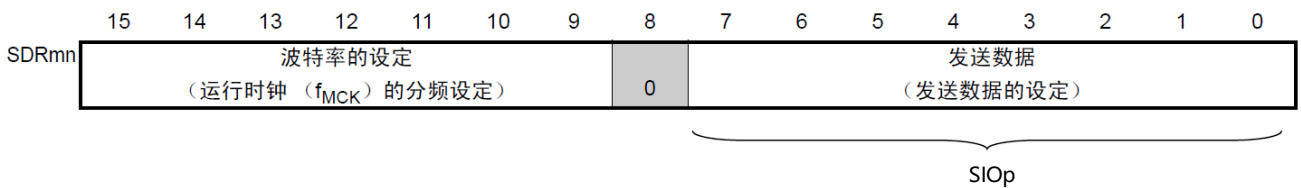
(a) 串行模式寄存器mn(SMRmn)



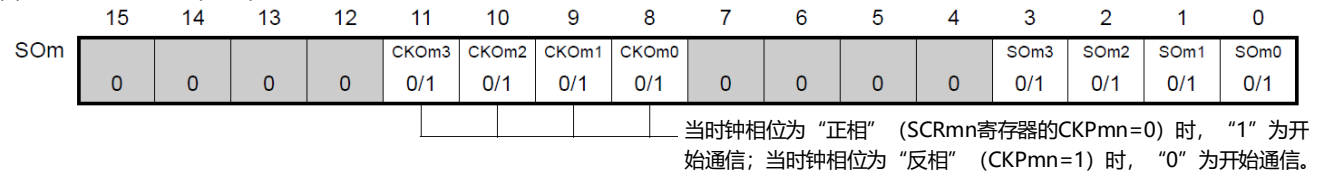
(b) 串行通信运行设定寄存器mn(SCRmn)



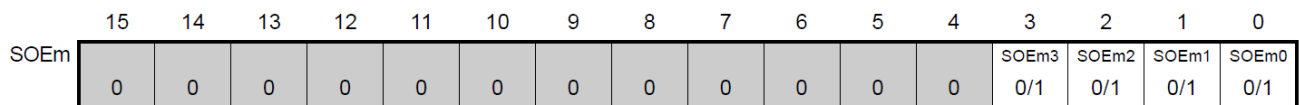
(c) 串行数据寄存器mn(SDRmn)



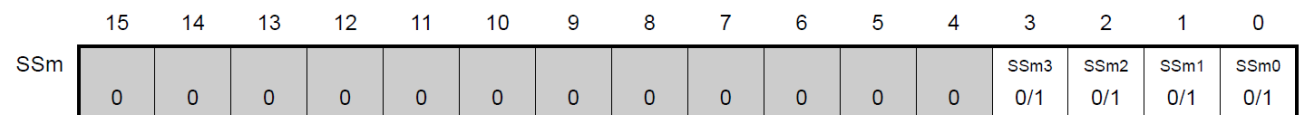
(d) 串行输出寄存器m(SOm)只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm)只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



注: 只限于SCR00, SCR01寄存器,其他固定为“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

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

(2) 操作步骤

图13-24 主控发送的初始设置步骤

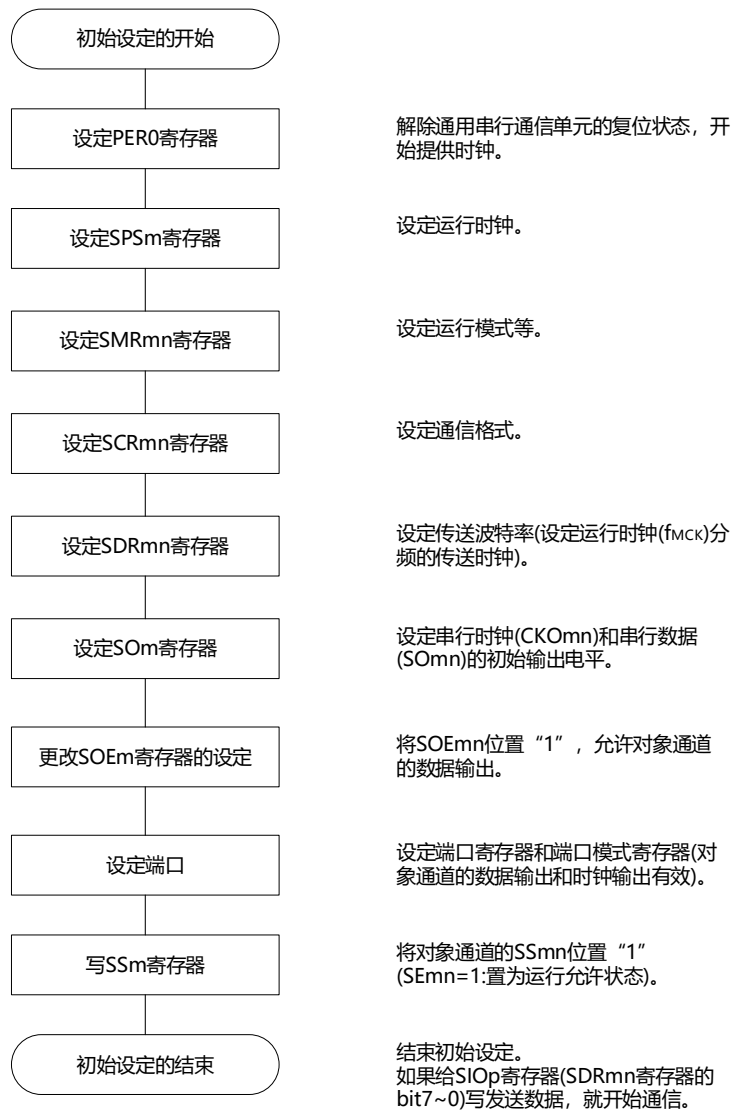


图13-25 主控发送的中止步骤

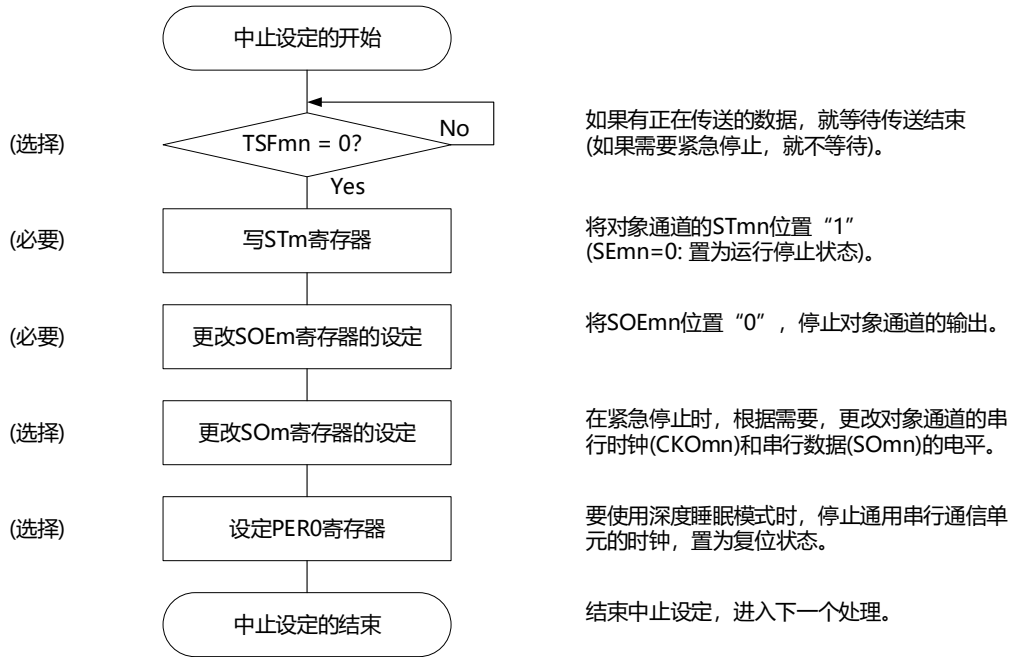
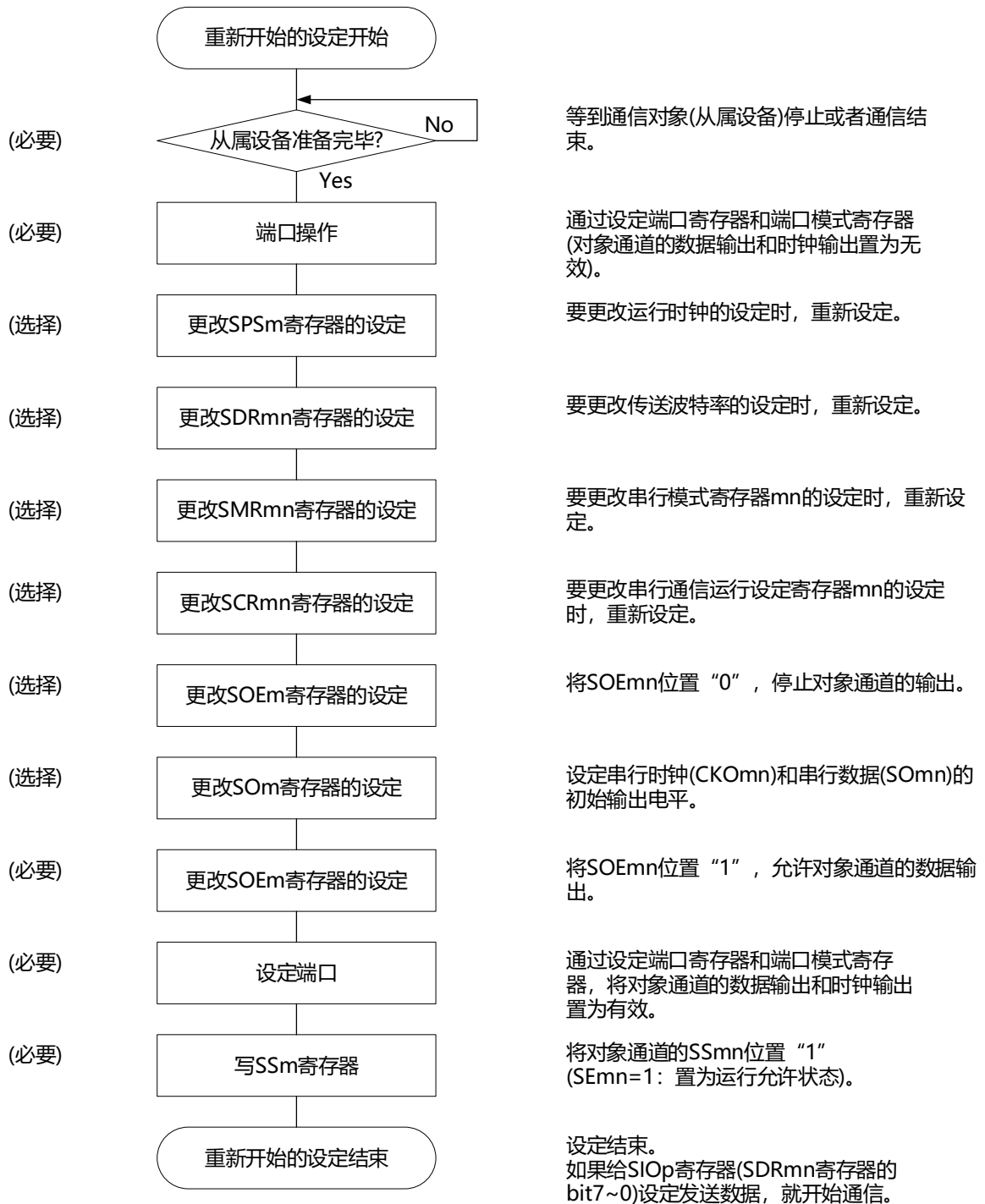


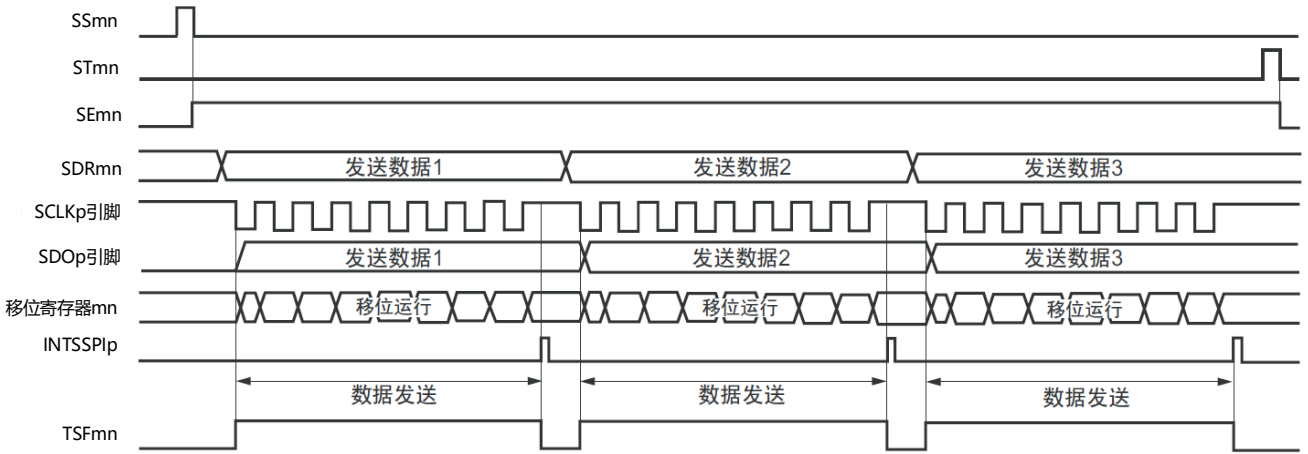
图13-26 重新开始主控发送的设置步骤



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

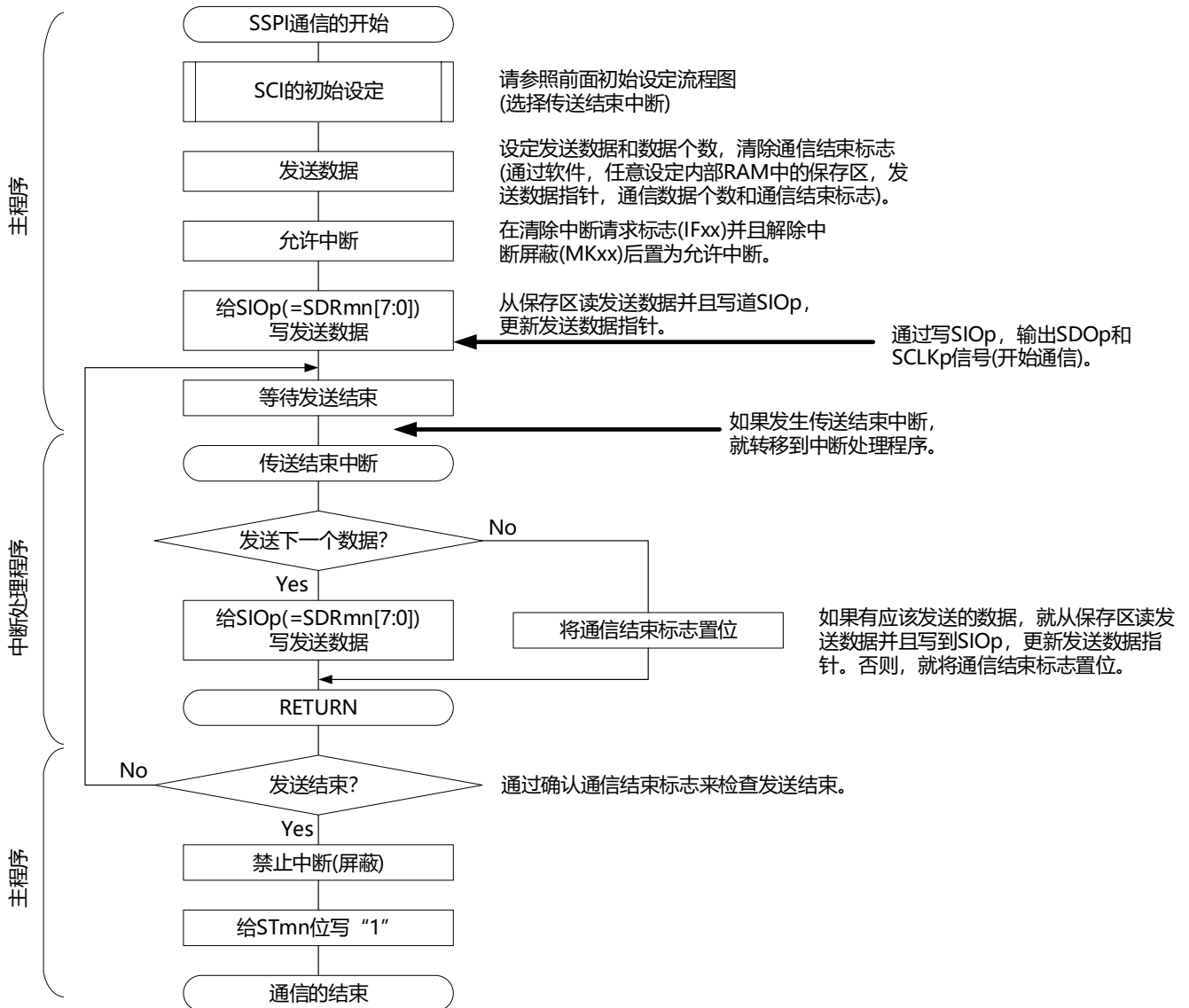
(3) 处理流程（单次发送模式）

图13-27 主控发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



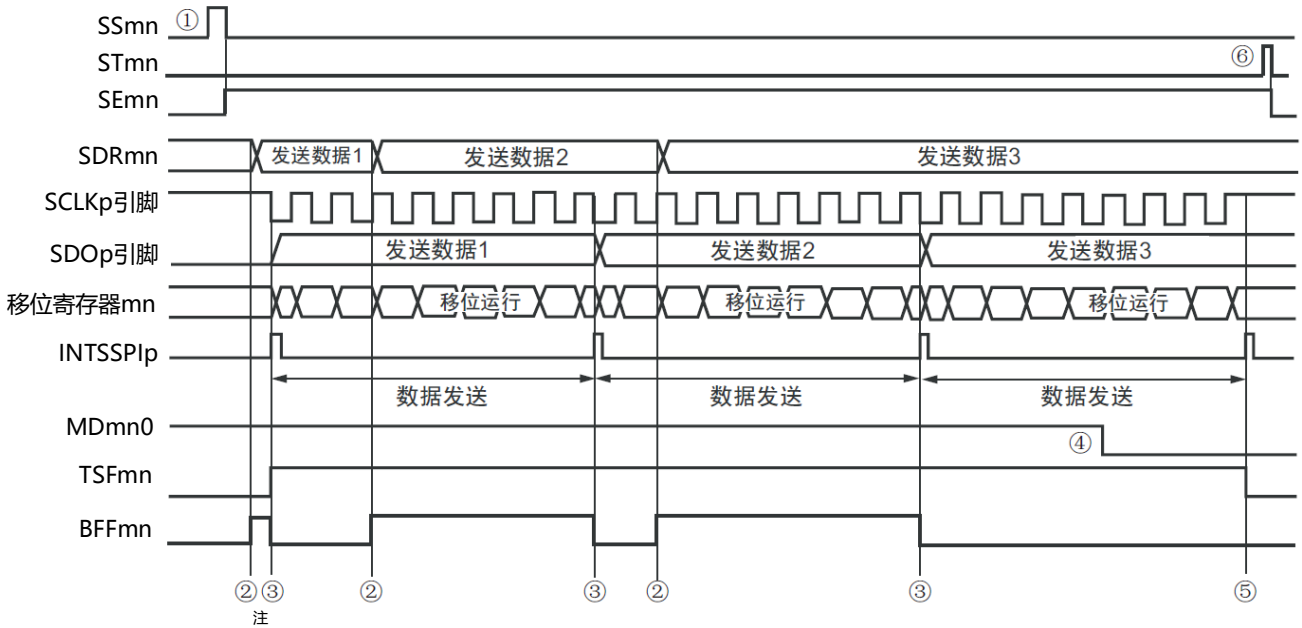
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-28 主控发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图13-29 主控发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

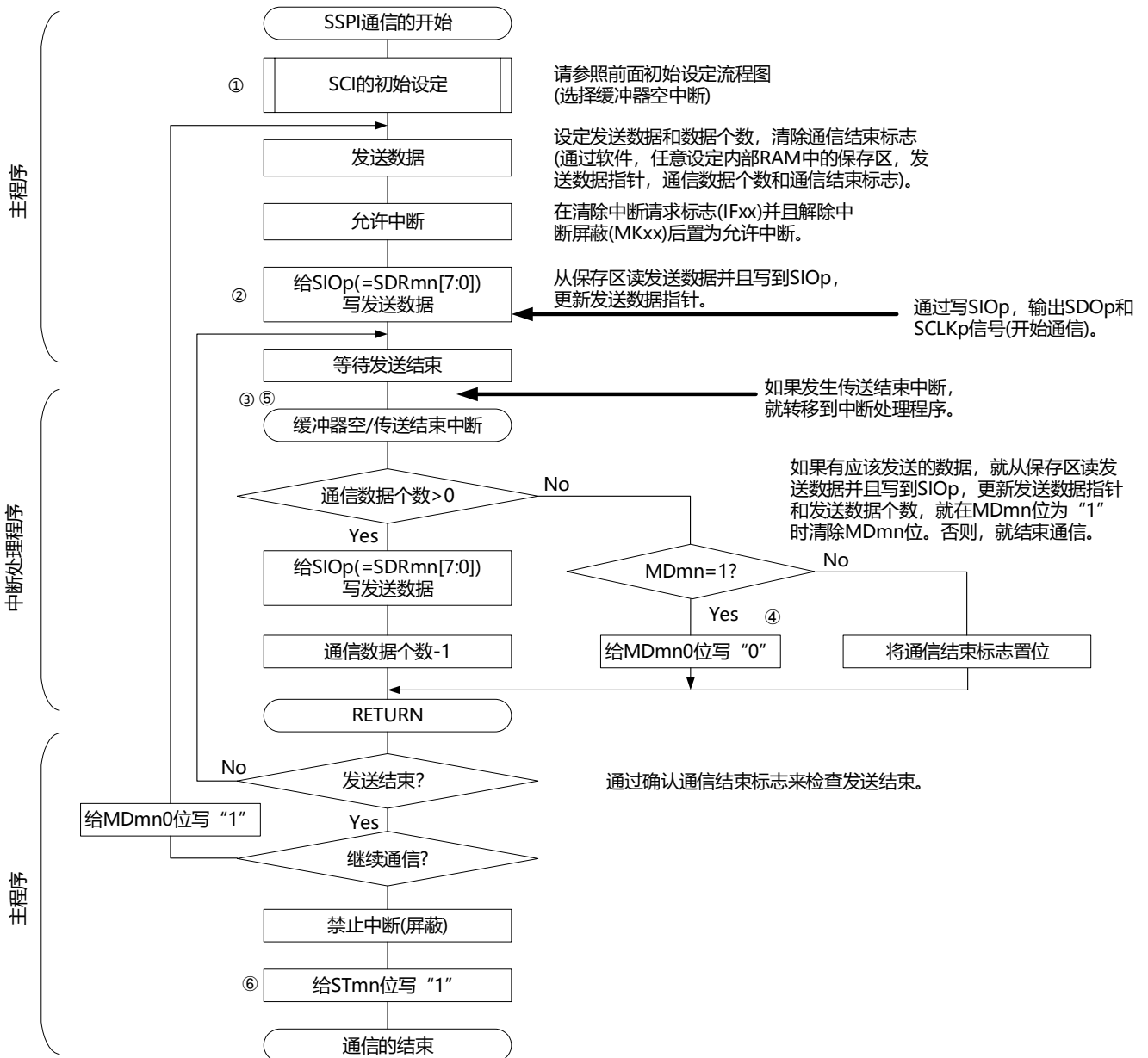


注 如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。

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

备注 m：单元号（m=0、1） n：通道号（n=0~3） p：SSPI号（p=00、01、10、11、20、21）
mn=00~03、10~11

图13-30 主控发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图13-29 主控发送（连续发送模式）的时序图”中的①~⑥。

13.5.2 主控接收

主控接收是指本产品输出传送时钟并且从其他设备接收数据的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00、 SDO00	SCLK01、 SDO01	SCLK10、 SDO10	SCLK11、 SDO11	SCLK20、 SDO20	SCLK21、 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位或者8位					
传送速率 ^注	Max. $f_{CLK}/2$ [Hz]（只限于SSPI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} ：系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB优先或者LSB优先					

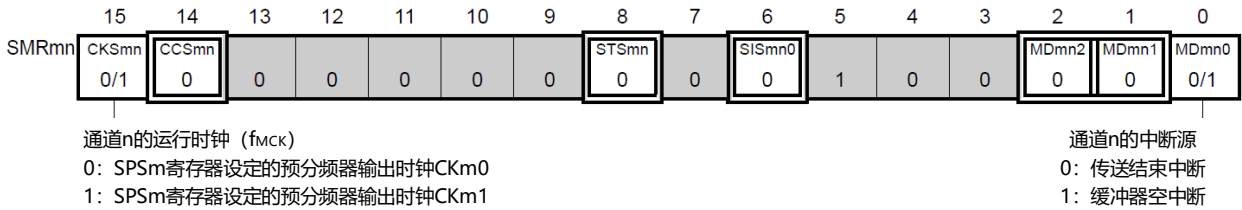
注 必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注 m：单元号（m=0、1） n：通道号（n=0~3） p：SSPI号（p=00、01、10、11、20、21）
mn=00~03、10~11

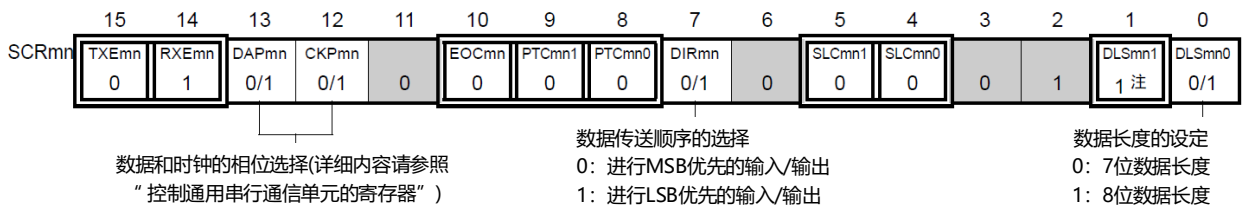
(1) 寄存器的设置

图13-31 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控接收时的寄存器设置内容例子

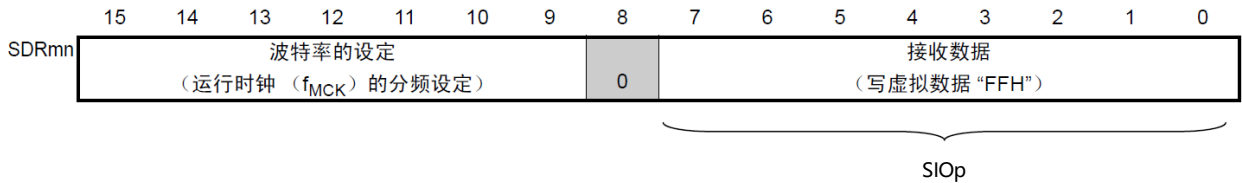
(a) 串行模式寄存器mn(SMRmn)



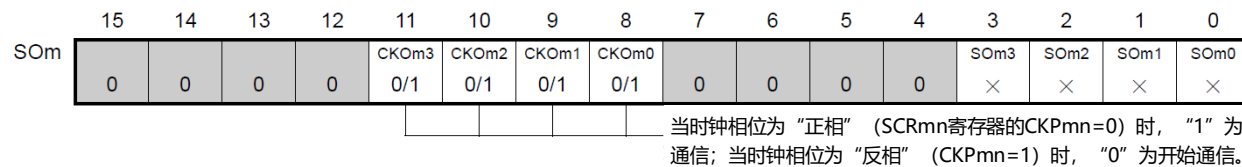
(b) 串行通信运行设定寄存器mn(SCRmn)



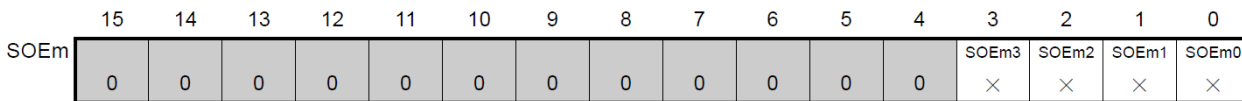
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



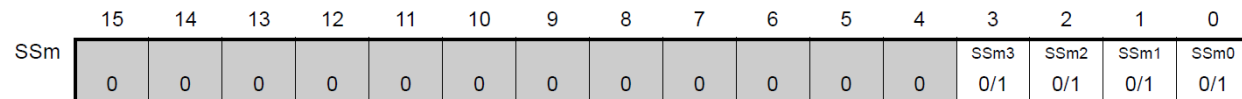
(d) 串行输出寄存器m(SOm)只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm)在此模式中不使用



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



注 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

2. : 在SSPI主控接收模式中为固定设置。 : 不能设置(设置初始值)。
×: 这是在此模式中不能使用的位(在其他模式中也不使用的情况下,设置初始值)。
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-32 主控接收的初始设置步骤

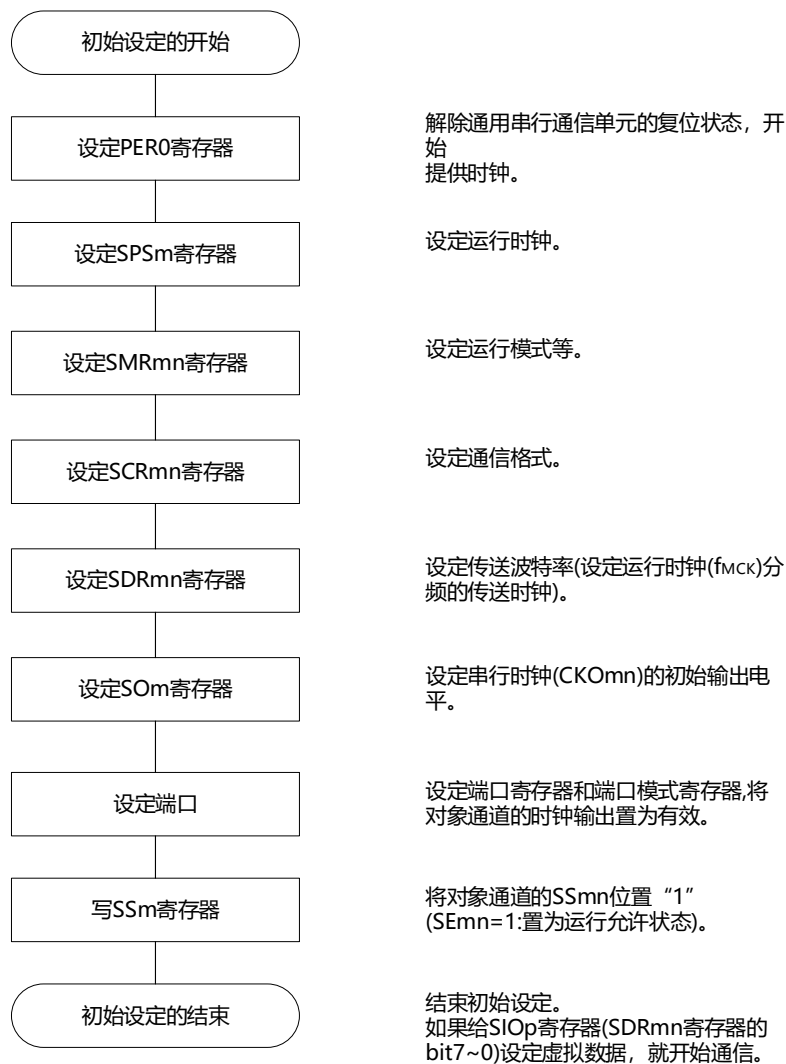


图13-33 主控接收的中止步骤

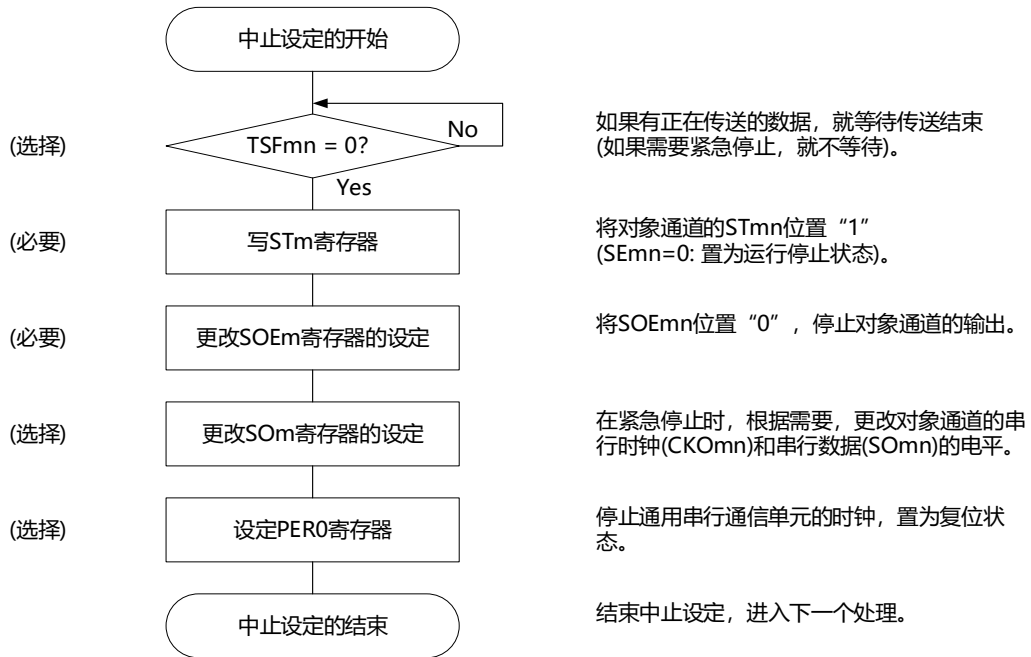


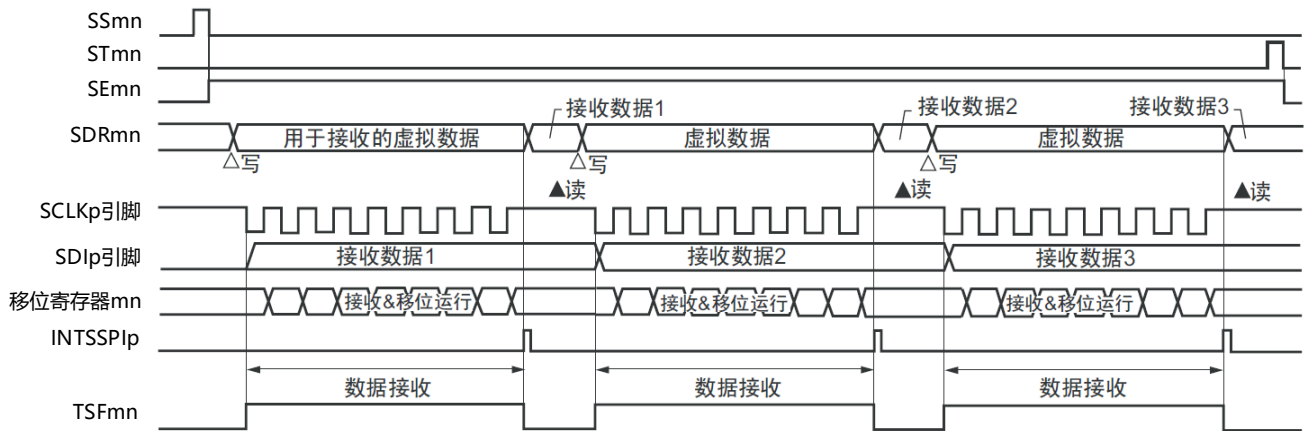
图13-34 重新开始主控接收的设置步骤



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

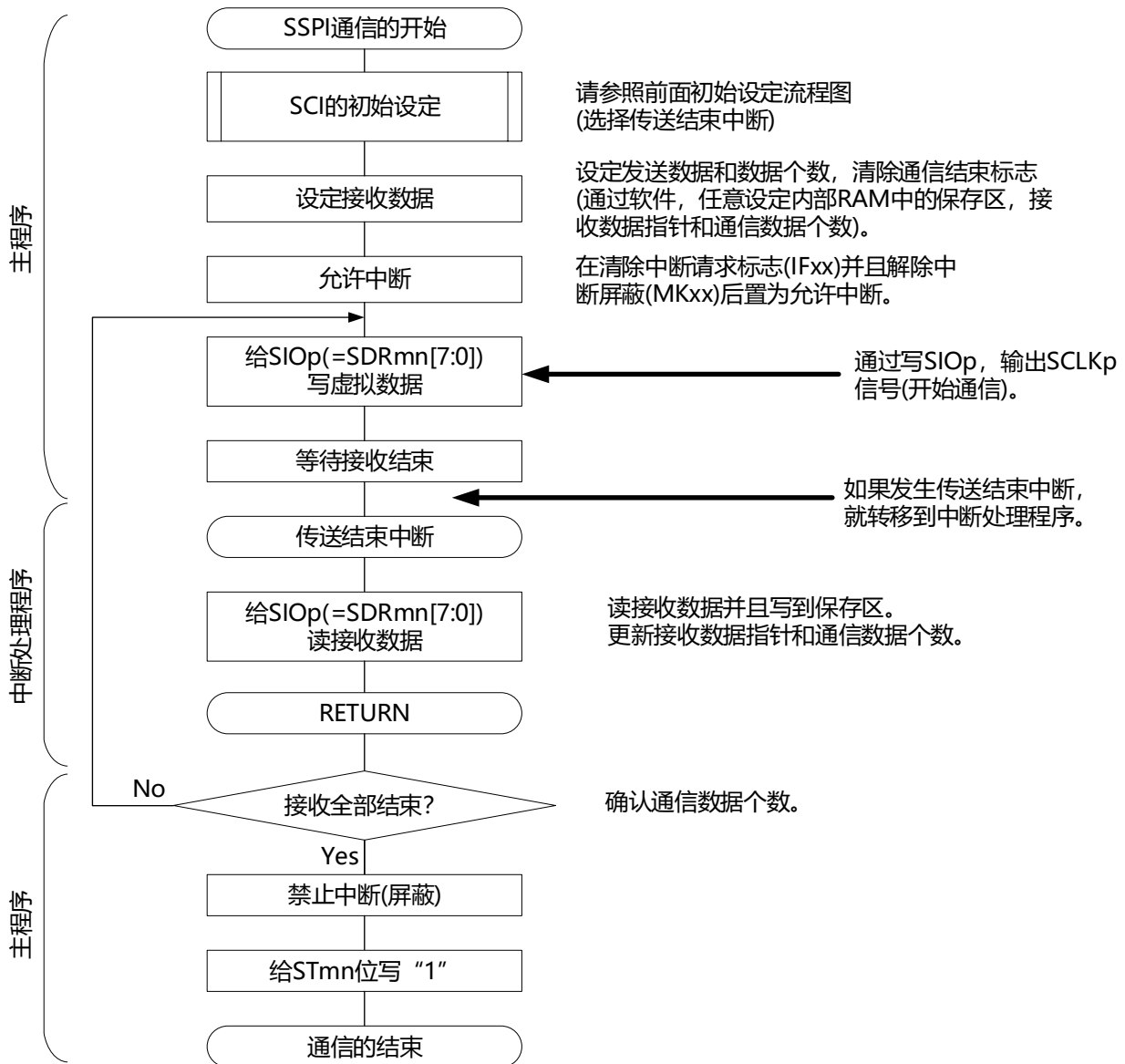
(3) 处理流程（单次接收模式）

图13-35 主控接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



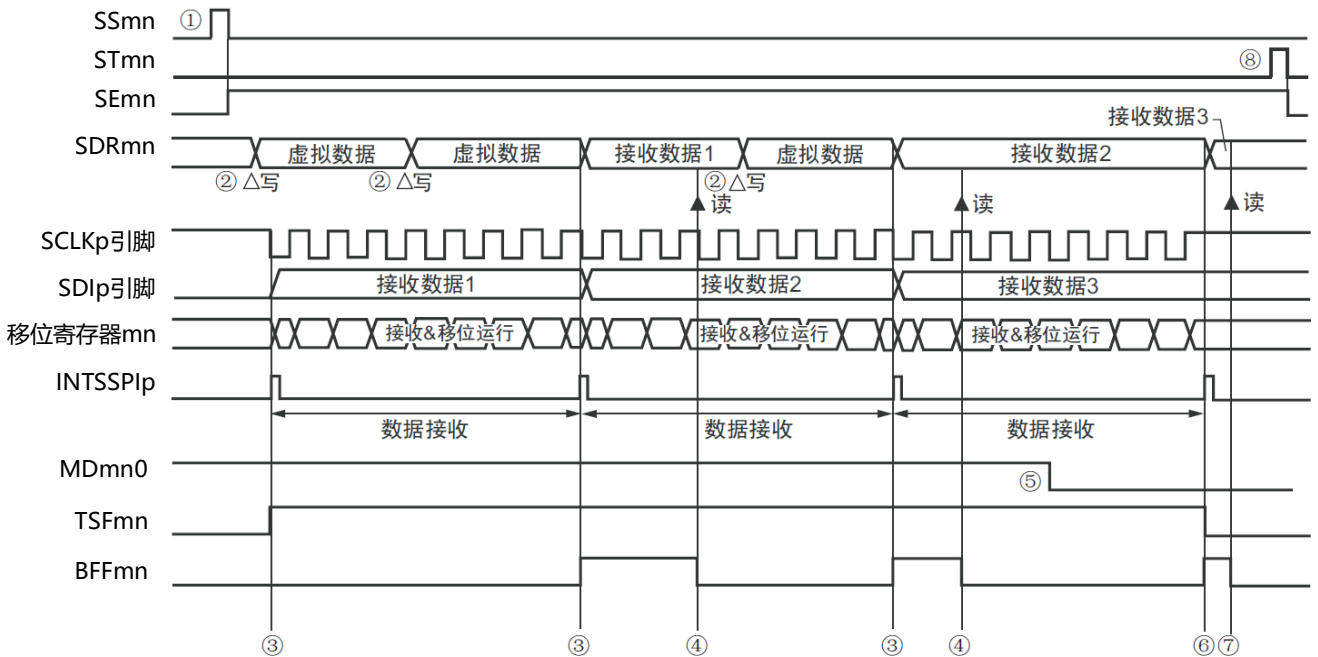
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-36 主控接收（单次接收模式）的流程图



(4) 处理流程（连续接收模式）

图13-37 主控接收（连续接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

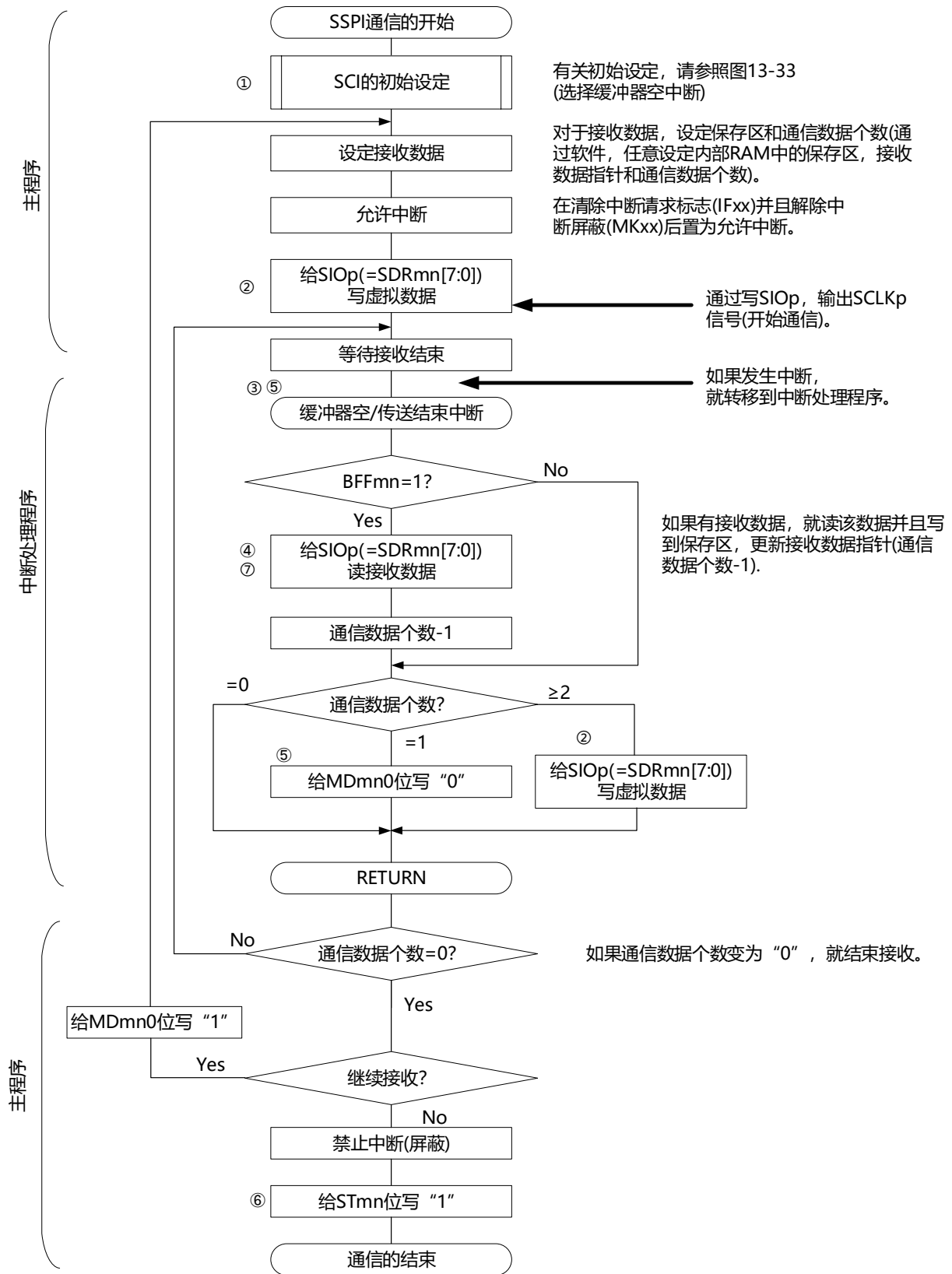


注意 即使在运行过程中也能改写MDmn0位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注1.图中的①~⑧对应“图13-38主控接收（连续接收模式）的流程图”中的①~⑧。

2.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-38 主控接收（连续接收模式）的流程图



备注 图中的①~⑧对应“图13-37 主控接收（连续接收模式）的时序图”中的①~⑧。

13.5.3 主控的发送和接收

主控的发送和接收是指本产品输出传送时钟并且和其他设备进行数据发送和接收的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00、SDI00、SDO00	SCLK01、SDI01、SDO01	SCLK10、SDI10、SDO10	SCLK11、SDI11、SDO11	SCLK20、SDI20、SDO20	SCLK21、SDI21、SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位或者8位					
传送速率注	Max. $f_{CLK}/2$ [Hz]（只限于SSPI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} ：系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB优先或者LSB优先					

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

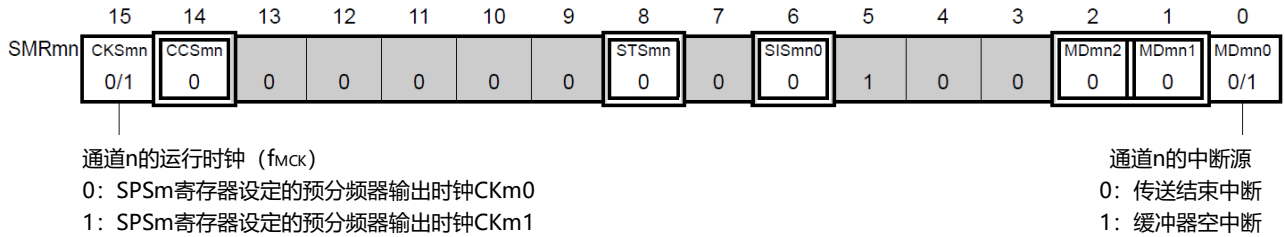
备注 m：单元号（m=0、1） n：通道号（n=0~3） p：SSPI号（p=00、01、10、11、20、21）

mn=00~03、10~11

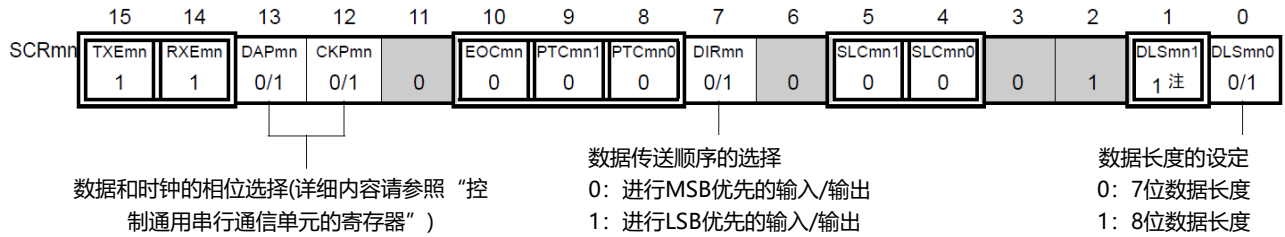
(1) 寄存器的设置

图13-39 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控发送和接收时的寄存器设置内容例子

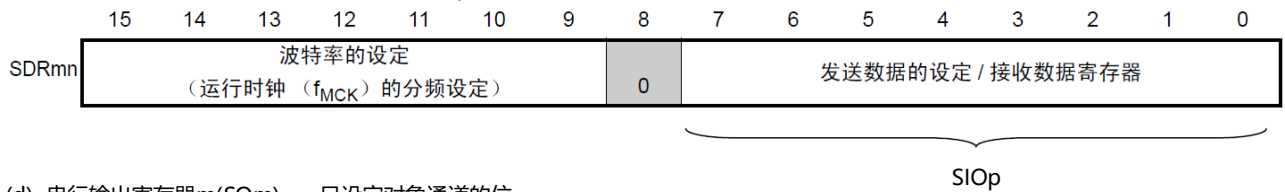
(a) 串行模式寄存器mn(SMRmn)



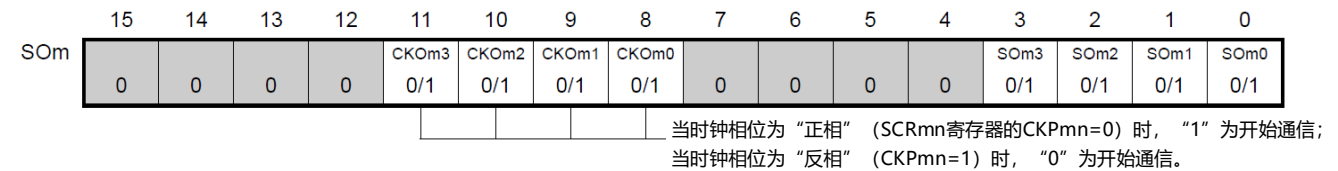
(b) 串行通信运行设定寄存器mn(SCRmn)



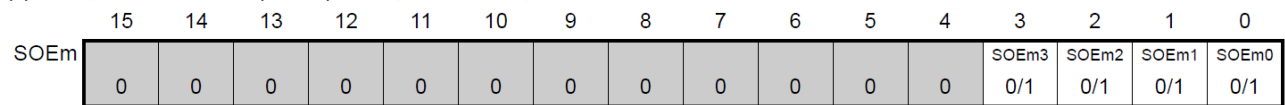
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



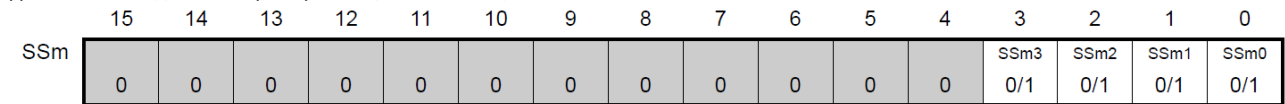
(d) 串行输出寄存器m(SOm)只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm)只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



注 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

☐ : 在SSPI主控发送和接收模式中为固定设置。■ : 不能设置(设置初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-40 主控发送和接收的初始设置步骤

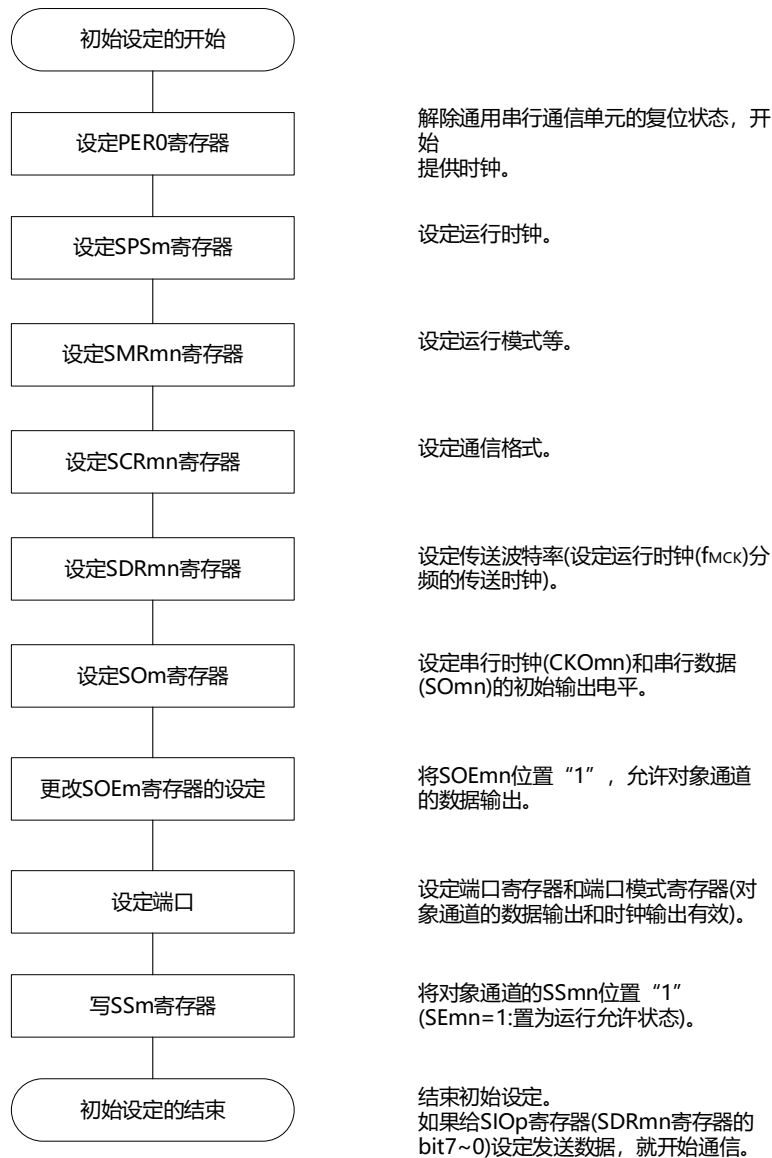


图13-41 主控发送和接收的中止步骤

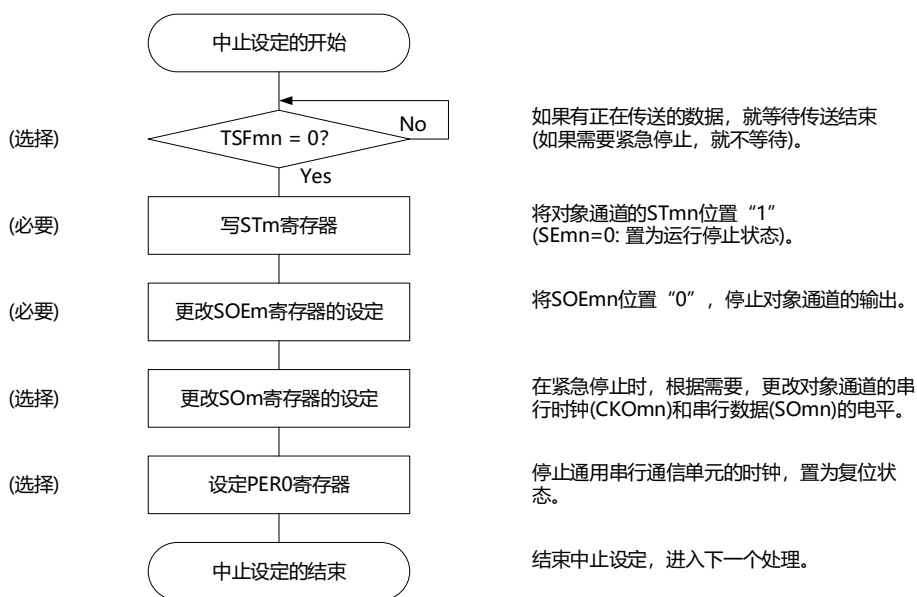
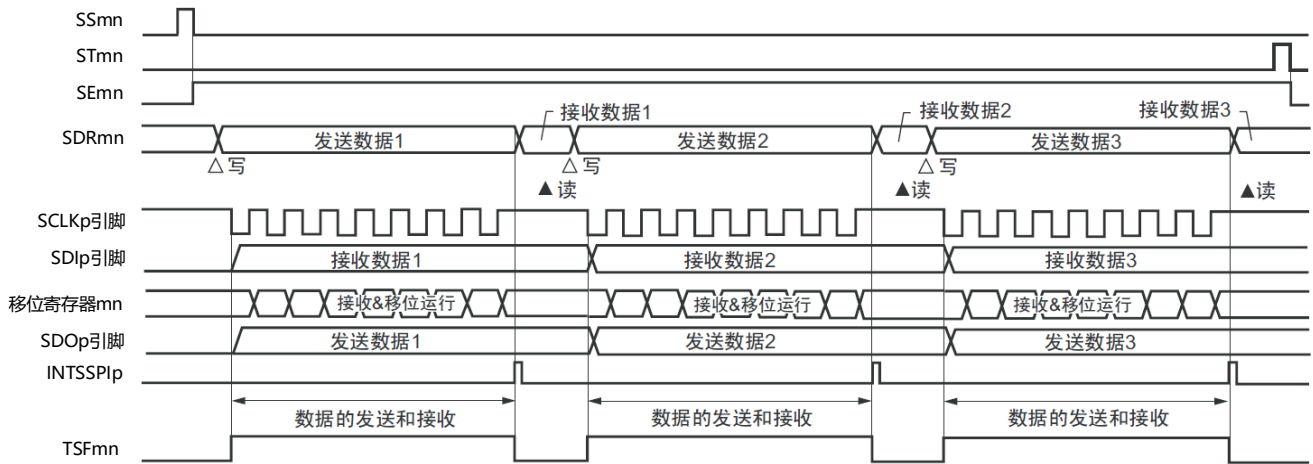


图13-42重新开始主控发送和接收的设置步骤



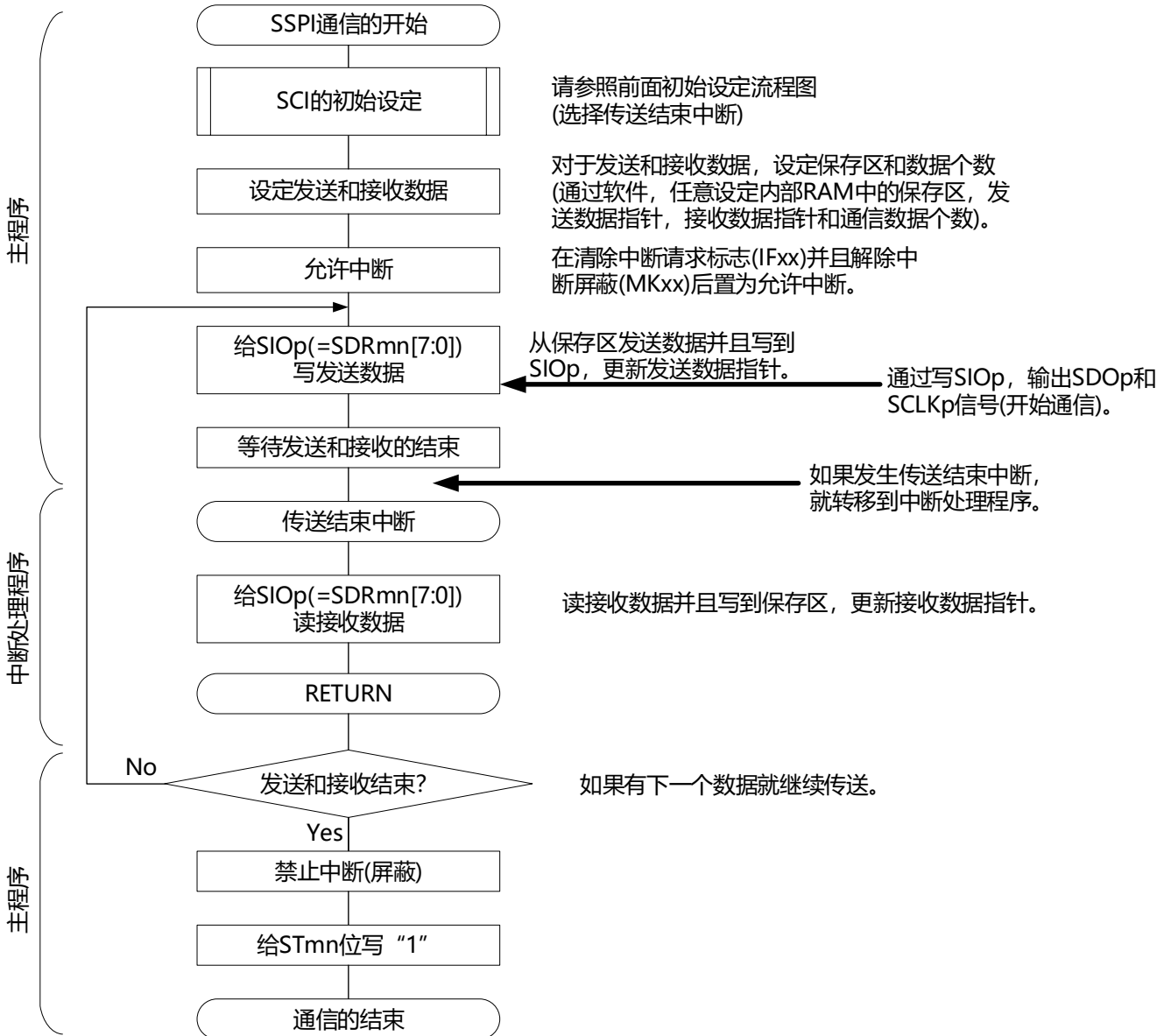
(3) 处理流程（单次发送和接收模式）

图13-43 主控发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



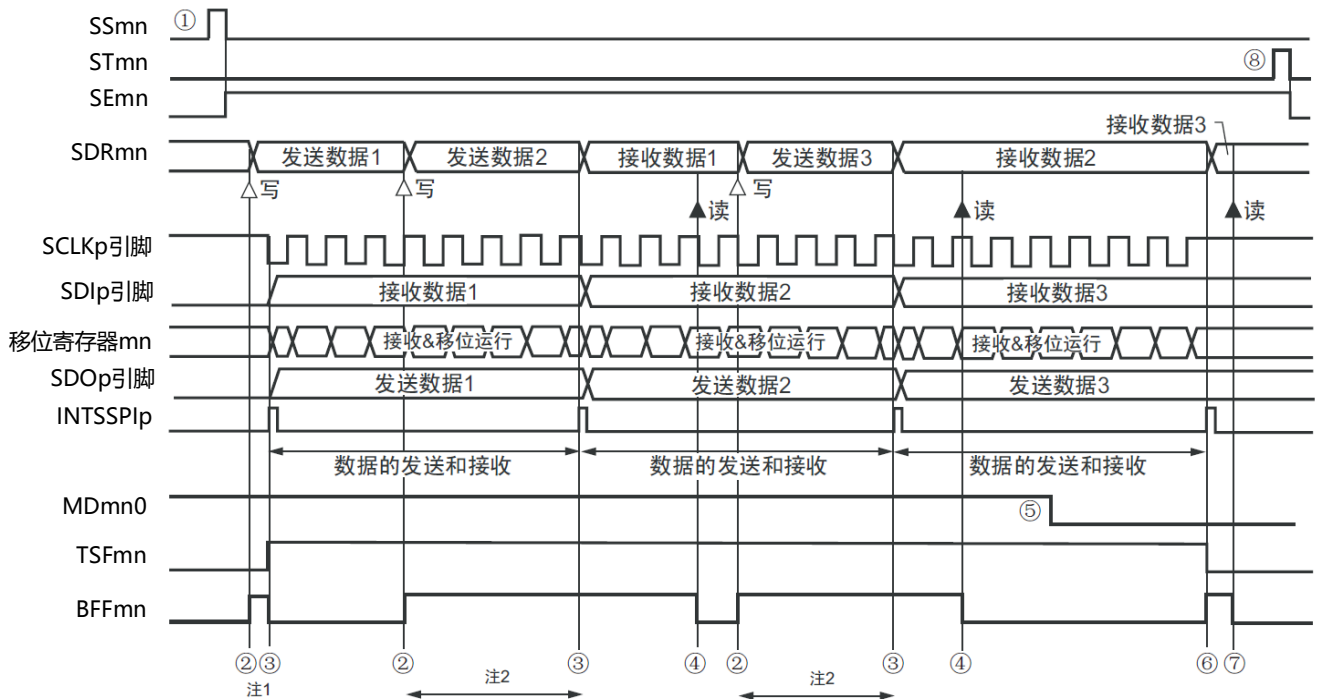
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-44 主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图13-45 主控发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）

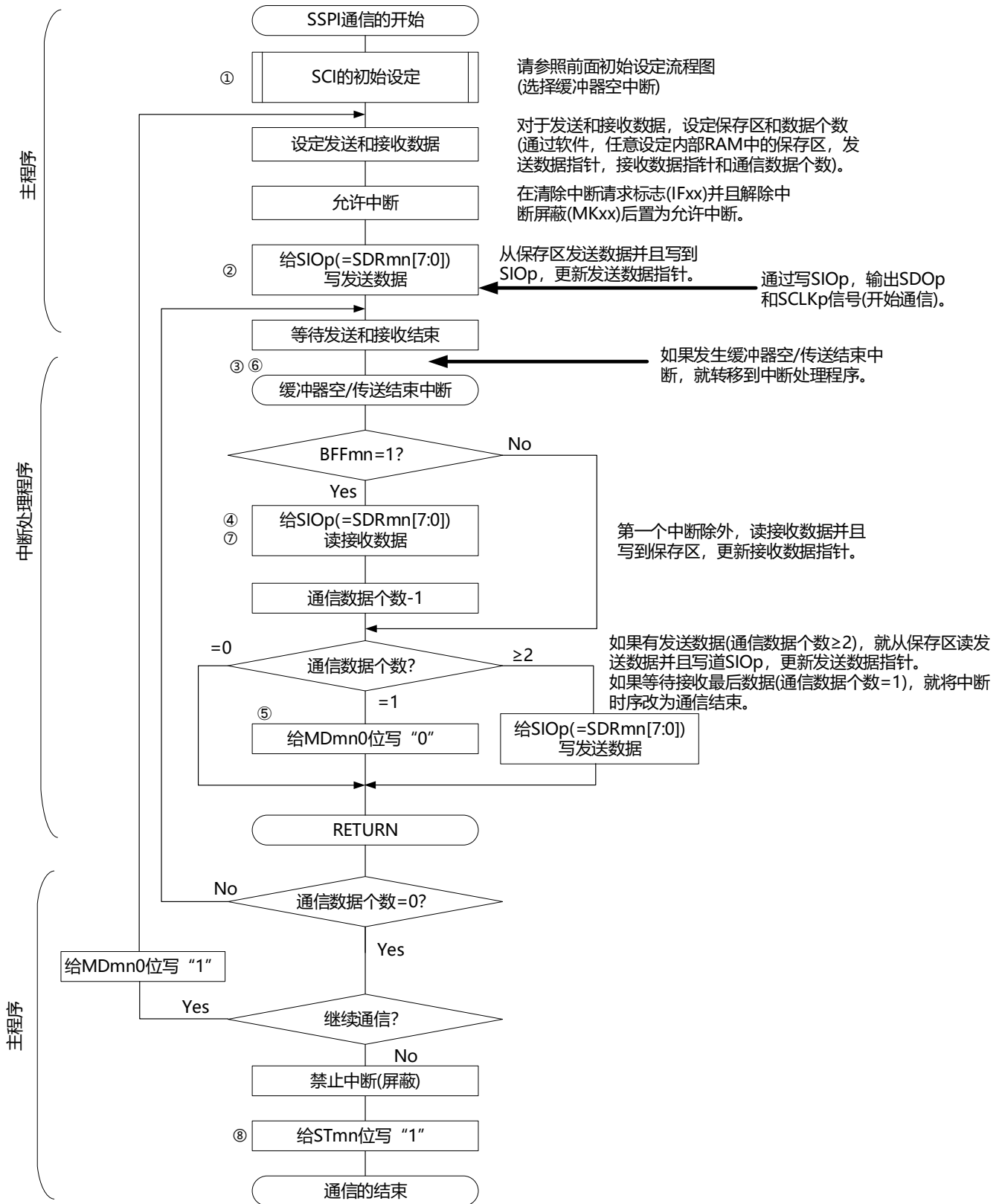


- 注 1.如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn）时）给SDRmn寄存器写发送数据，就重写发送数据。
- 2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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

- 备注 1.图中的①~⑧对应“图13-46 主控发送和接收（连续发送和接收模式）的流程图”中的①~⑧。
- 2.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-46 主控发送和接收（连续发送和接收模式）的流程图



备注 图中的①~⑧对应“图13-45 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

13.5.4 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BAT32G133微控制器将数据发送到其他设备的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00、 SDO00	SCLK01、 SDO01	SCLK10、 SDO10	SCLK11、 SDO11	SCLK20、 SDO20	SCLK21、 SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位或者8位					
传送速率	Max.f _{MCK} /6[Hz] ^{注1、2}					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 •DAPmn=0：在串行时钟开始运行时，开始数据输出。 •DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB优先或者LSB优先					

注 1.因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为f_{MCK}/6[Hz]。

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

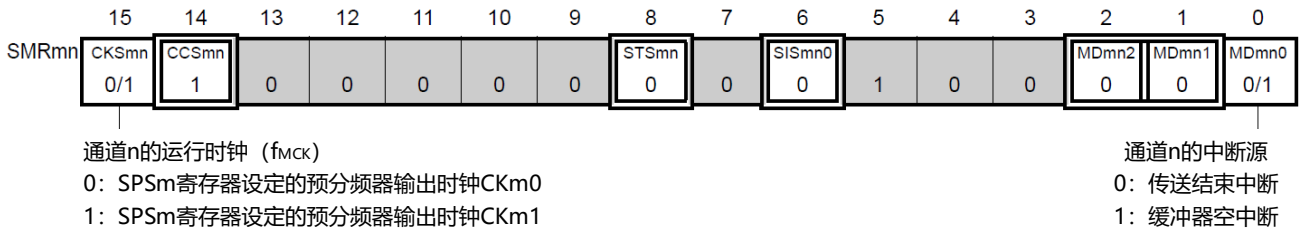
备注1.f_{MCK}：对象通道的运行时钟频率

2.m：单元号（m=0、1） n：通道号（n=0~3） mn=00~03、10~11

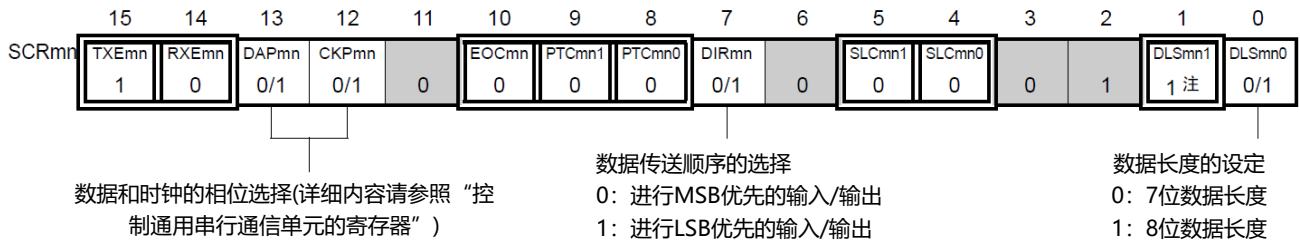
(1) 寄存器的设置

图13-47 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
从属发送时的寄存器设置内容例子

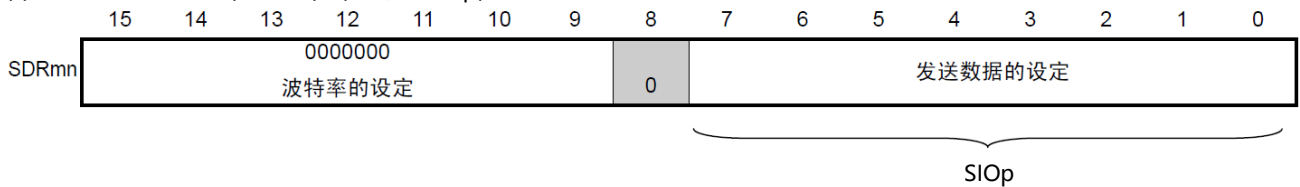
(a) 串行模式寄存器mn(SMRmn)



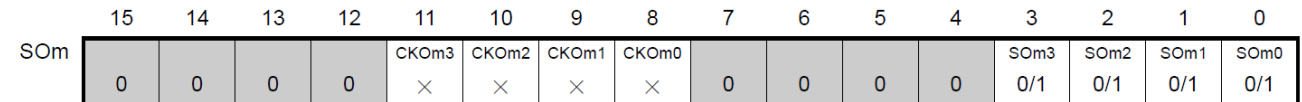
(b) 串行通信运行设定寄存器mn(SCRmn)



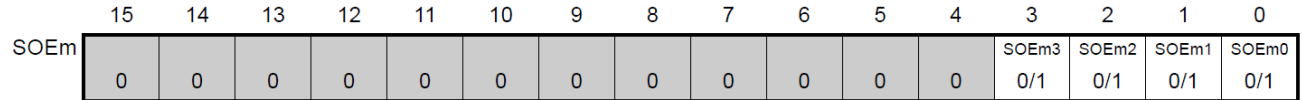
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



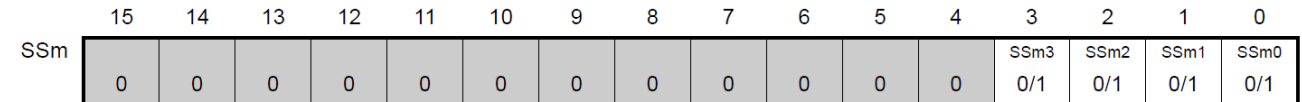
(d) 串行输出寄存器m(SOm)只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm)只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



注 只限于SCR00寄存器和SCR01寄存器,其他固定为“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
 mn=00~03、10~11

2. : 在SSPI从属发送模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-48 从属发送的初始设置步骤

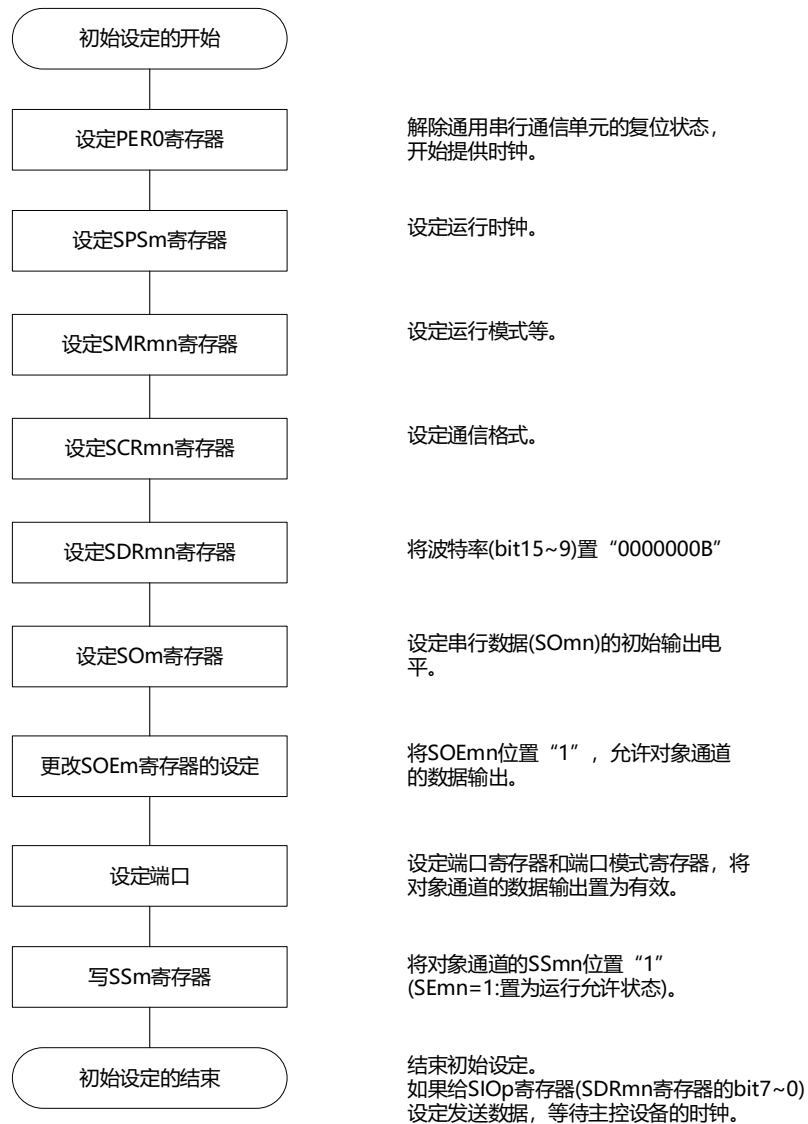


图13-49 从属发送的中止步骤

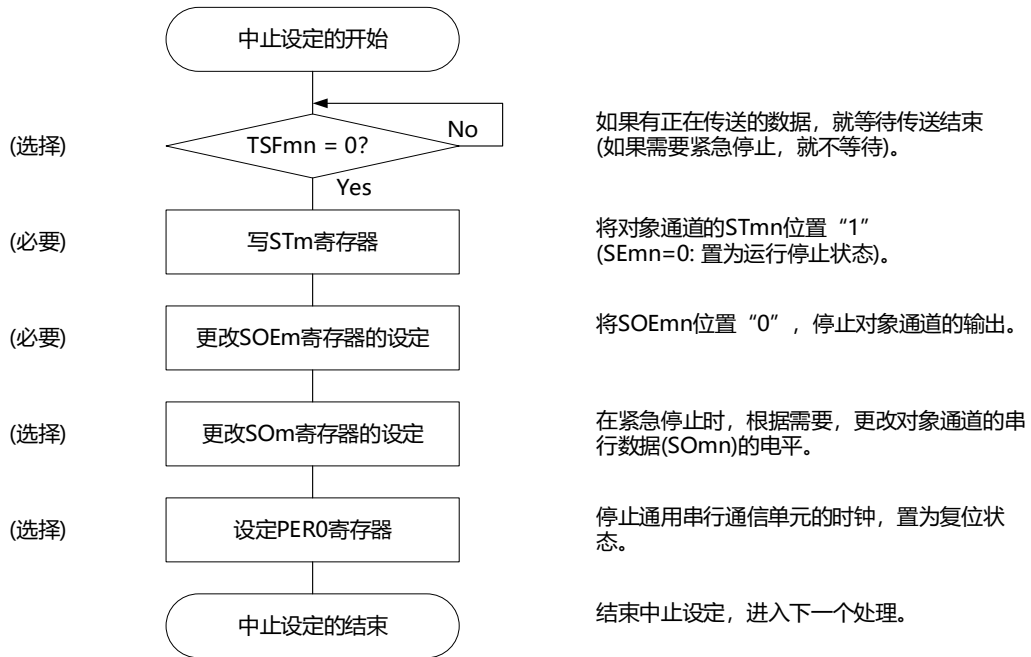


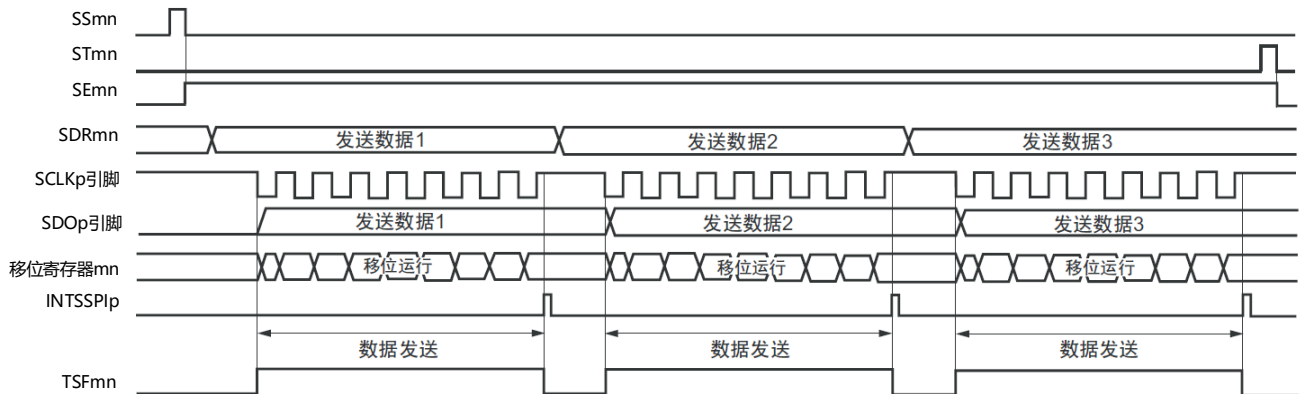
图13-50 重新开始从属发送的设置步骤



备注 如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设置而不是进行重新开始的设置。

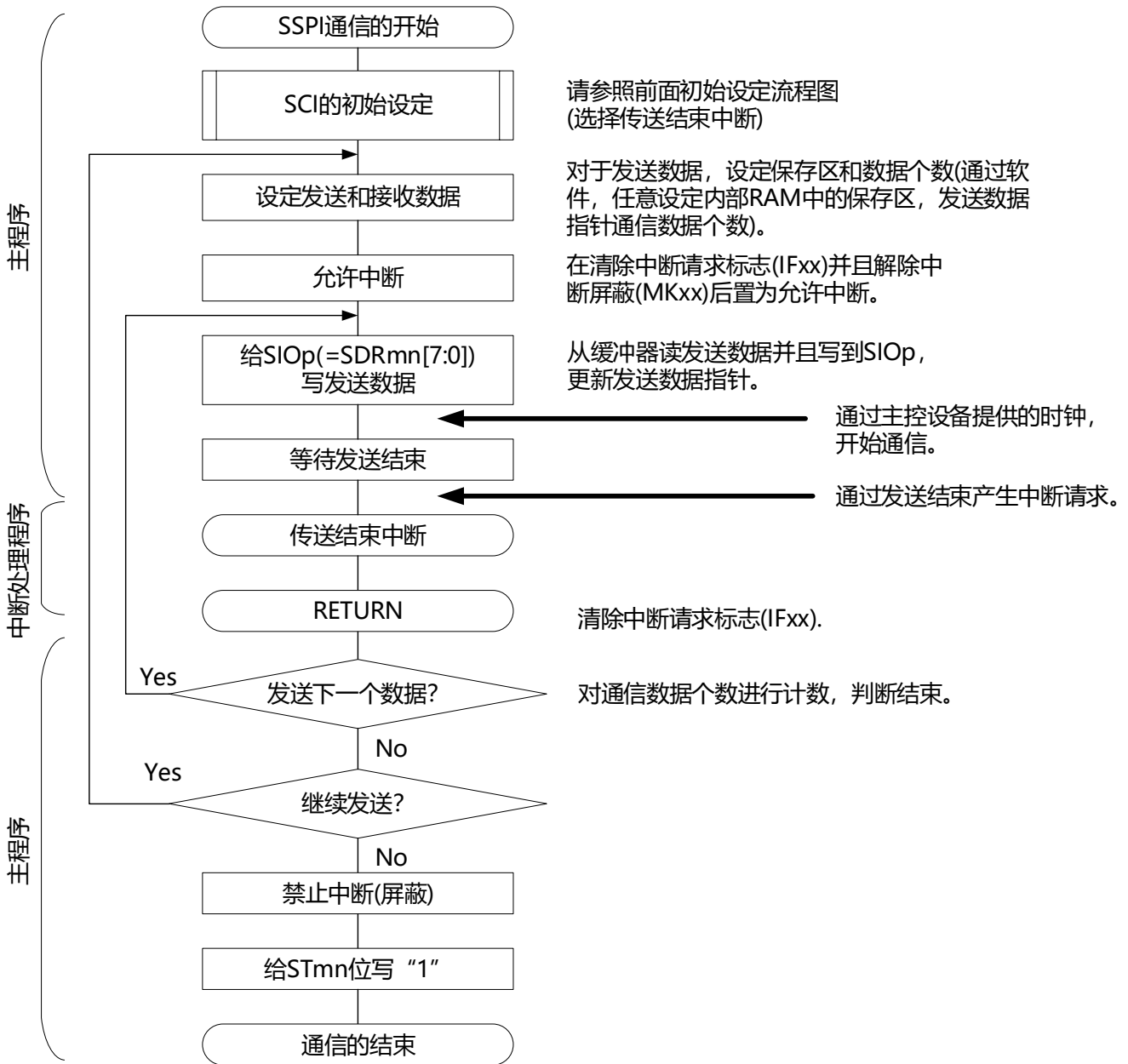
(3) 处理流程（单次发送模式）

图13-51 从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



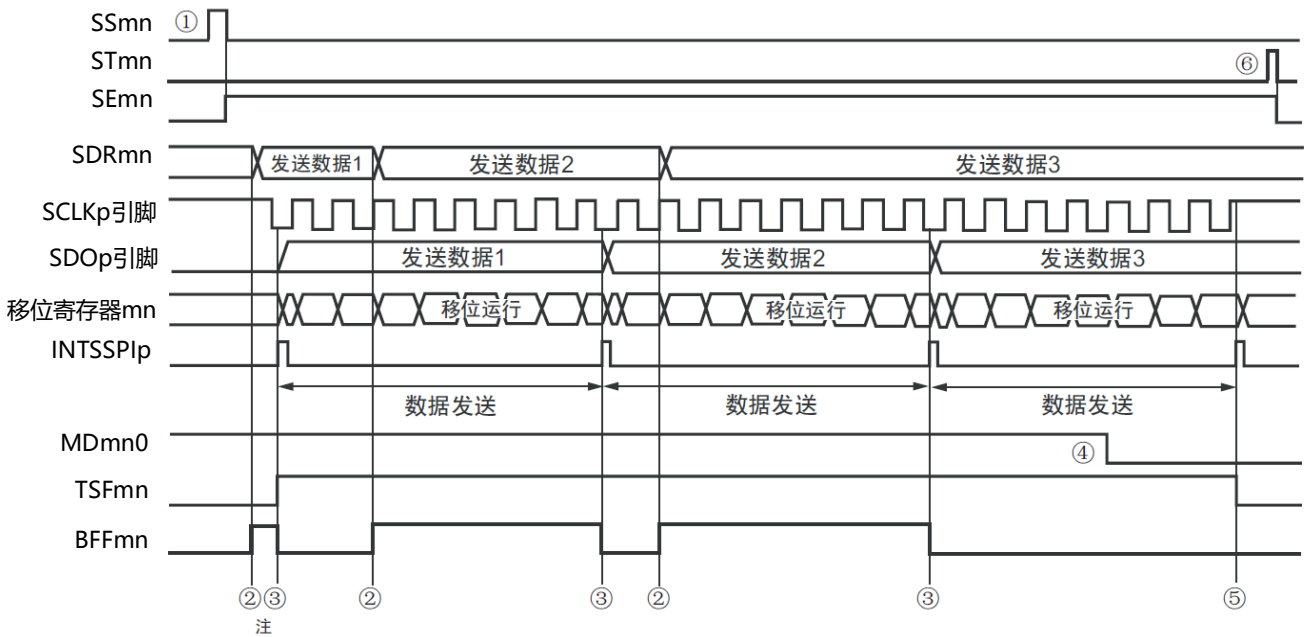
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-52 从属发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图13-53 从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

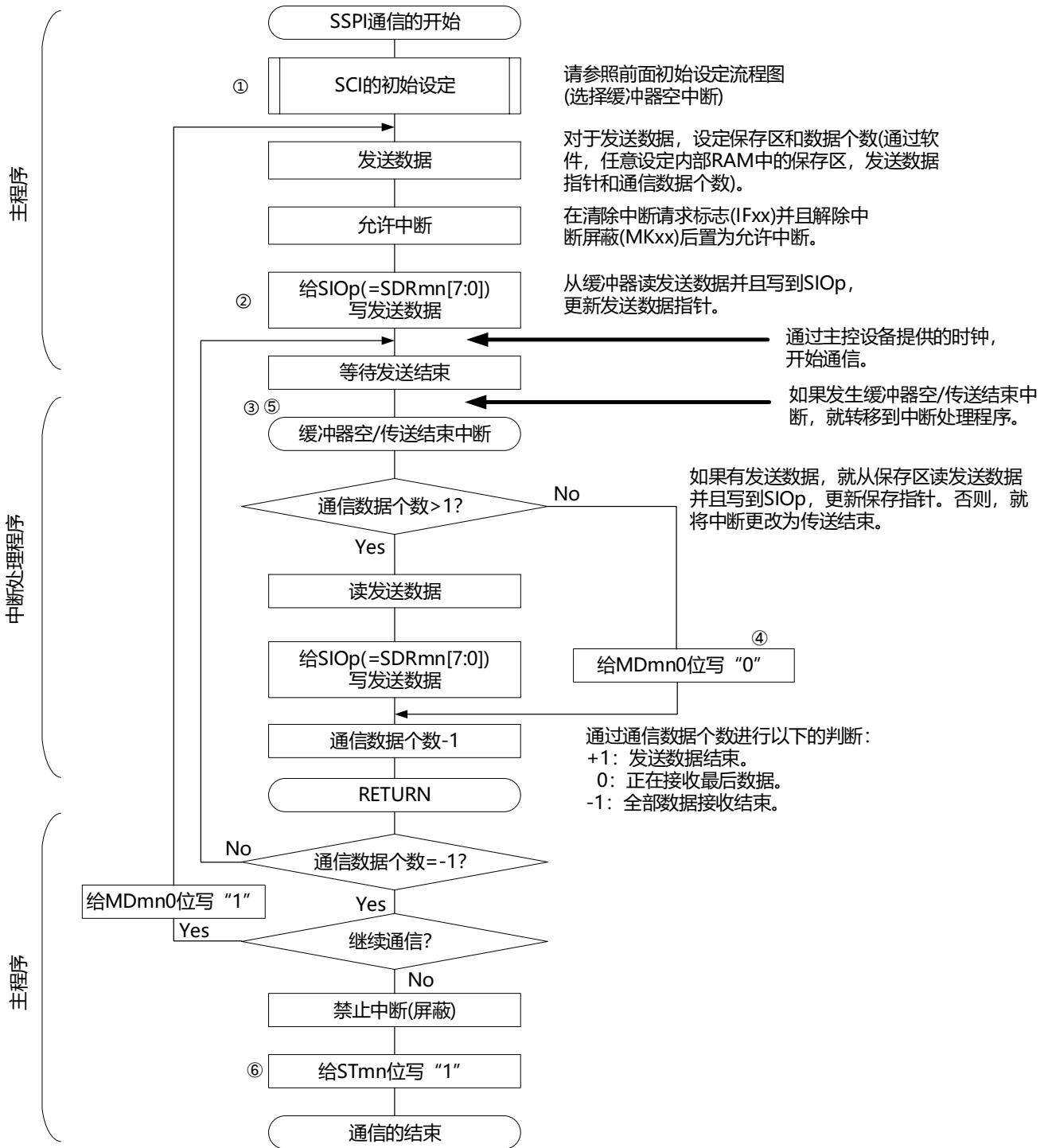


注 如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。

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

备注 m：单元号（m=0、1） n：通道号（n=0~3） p：SSPI号（p=00、01、10、11、20、21）
mn=00~03、10~11

图13-54 从属发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图13-53从属发送（连续发送模式）的时序图”中的①~⑥。

13.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的 通道0	SCI0的 通道1	SCI0的 通道2	SCI0的 通道3	SCI1的 通道0	SCI1的 通道1
使用的引脚	SCLK00、 SDI00	SCLK01、 SDI01	SCLK10、 SDI10	SCLK11、 SDI11	SCLK20、 SDI20	SCLK21、 SDI21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	只限于传送结束中断（禁止设置缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位或者8位					
传送速率	Max.f _{MCK} /6[Hz] ^{注1、2}					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 •DAPmn=0：在串行时钟开始运行时，开始数据输出。 •DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB优先或者LSB优先					

注 1.因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为f_{MCK}/6[Hz]。

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

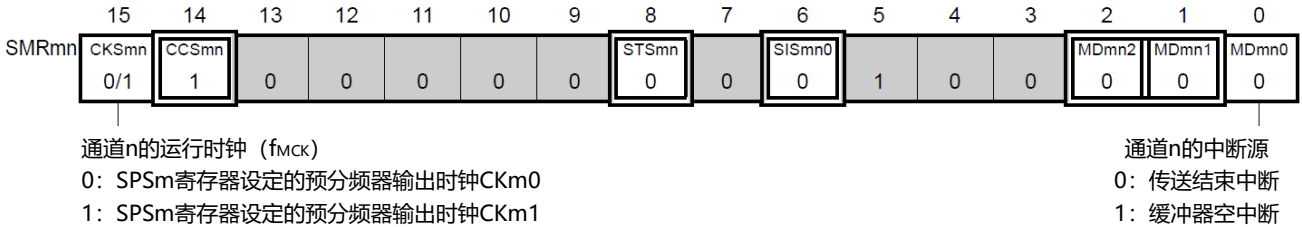
备注1.f_{MCK}：对象通道的运行时钟频率

2.m：单元号（m=0、1） n：通道号（n=0~3） mn=00~03、10~11

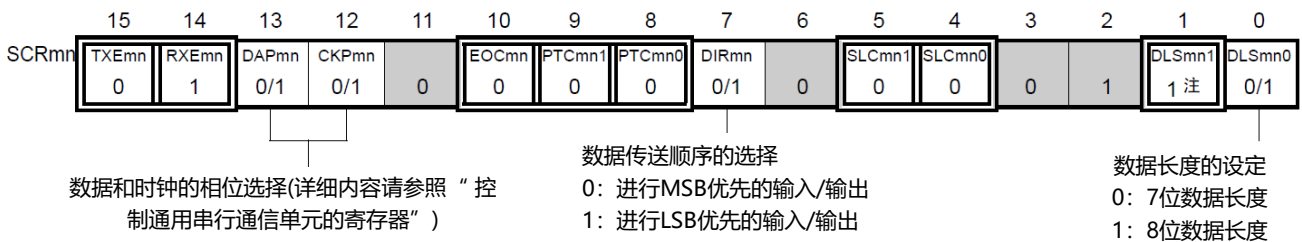
(1) 寄存器的设置

图13-55 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
从属接收时的寄存器设置内容例子

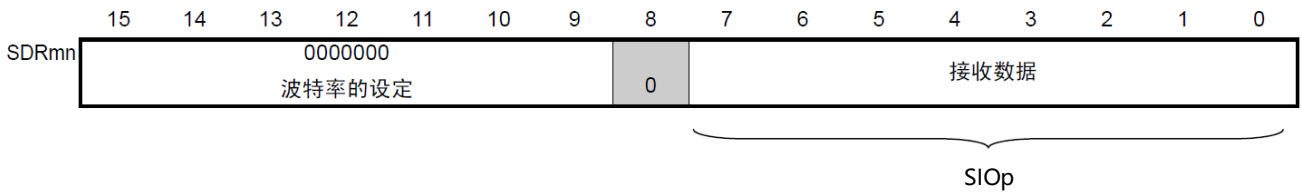
(a) 串行模式寄存器mn(SMRmn)



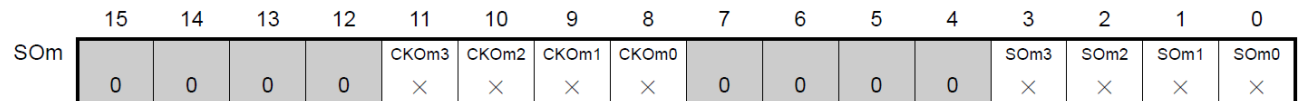
(b) 串行通信运行设定寄存器mn(SCRmn)



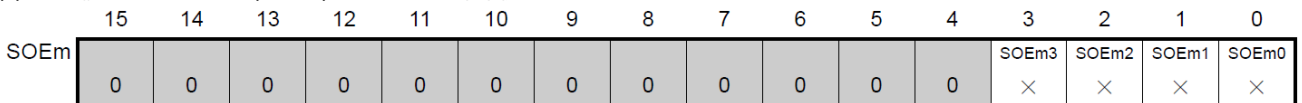
(c) 串行数据寄存器mn(SDRmn) (低8位: SIOp)



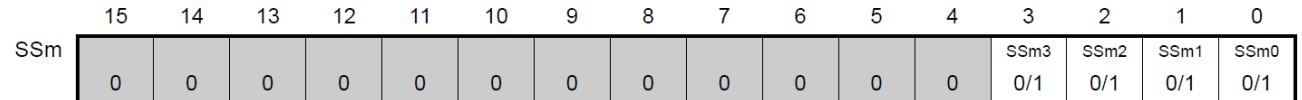
(d) 串行输出寄存器m(SOm)在此模式中不使用。



(e) 串行输出允许寄存器m(SOEm)在此模式中不使用。



(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。



注 只限于SCR00寄存器和SCR01寄存器，其他固定为“1”。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21、30、31)
 mn=00~03、10~11

2 : 在从属接收模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

2) 操作步骤

图13-56 从属接收的初始设置步骤

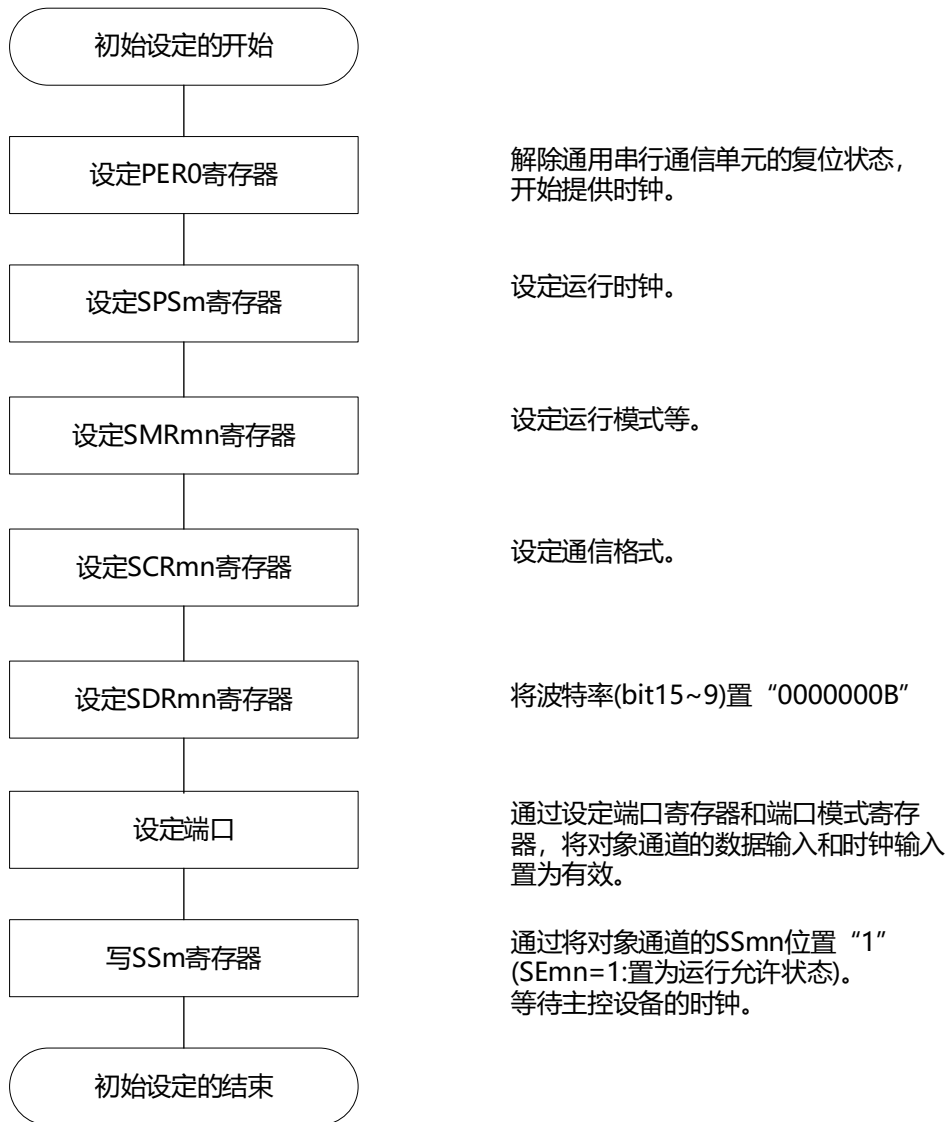


图13-57 从属接收的中止步骤

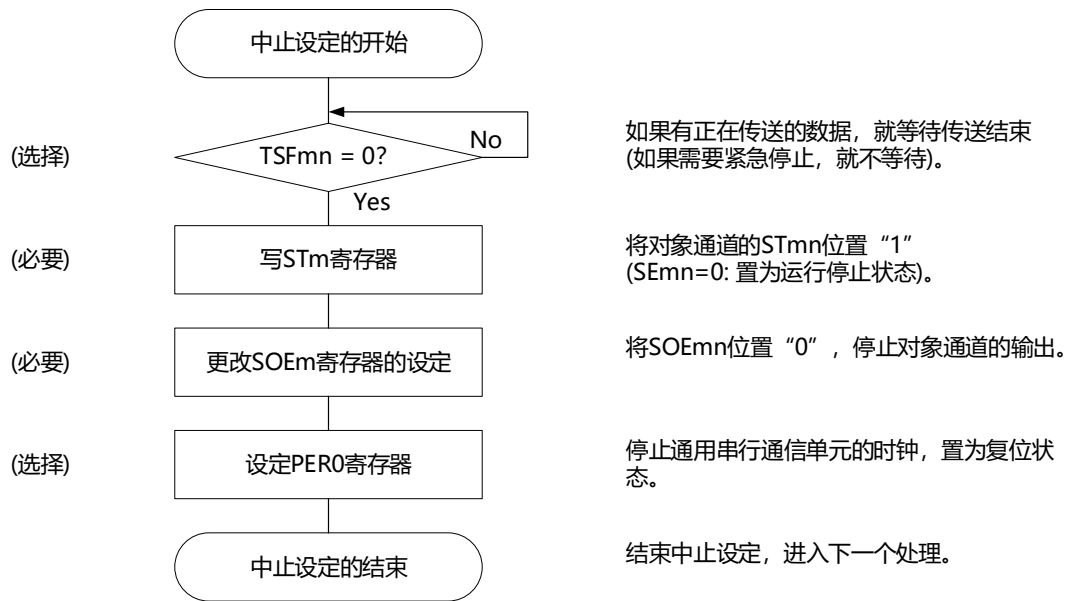


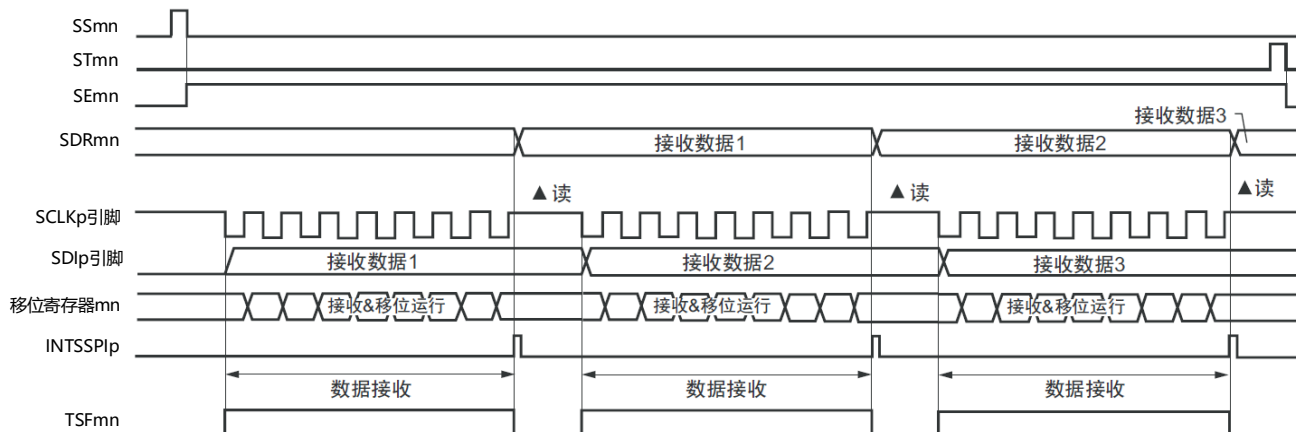
图13-58 重新开始从属接收的设置步骤



备注 如果在中止设置中改写PER0来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设置而不是进行重新开始的设置。

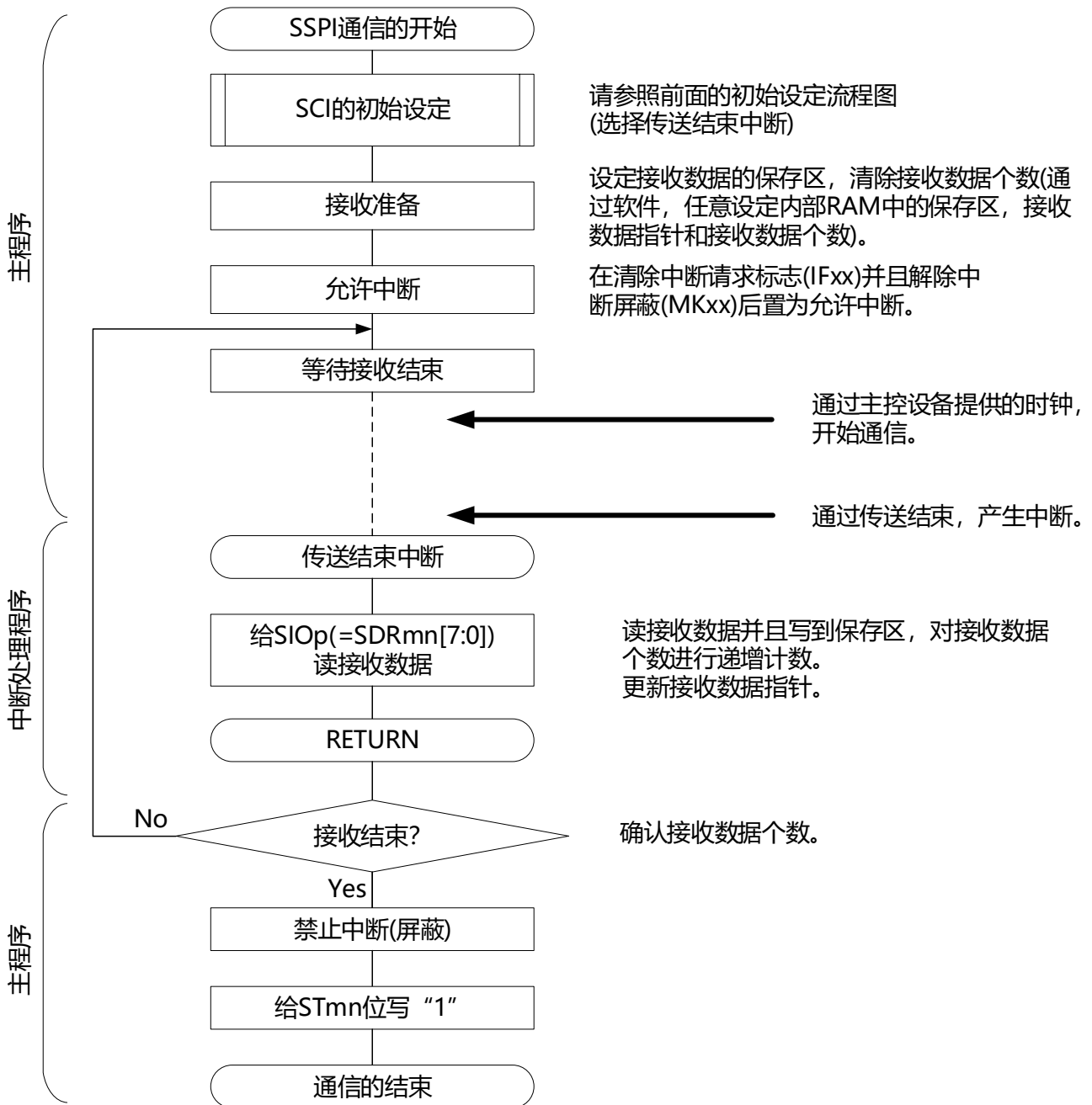
(3) 处理流程（单次接收模式）

图13-59 从属接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21、30、31)
mn=00~03、10~11

图13-60 从属接收（单次接收模式）的流程图



13.5.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品微控制器和其他设备进行数据发送和接收的运行。

3线串行I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLK00、SDI00、SDO00	SCLK01、SDI01、SDO01	SCLK10、SDI10、SDO10	SCLK11、SDI11、SDO11	SCLK20、SDI20、SDO20	SCLK21、SDI21、SDO21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
错误检测标志	只有溢出错误检测标志 (OVFmn)。					
传送数据长度	7位或者8位					
传送速率	Max. $f_{MCK}/6$ [Hz] ^{注1、2}					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 •DAPmn=0: 在串行时钟开始运行时, 开始数据输入/输出。 •DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输入/输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

注 1.因为在内部对SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率 $f_{MCK}/6$ [Hz]。

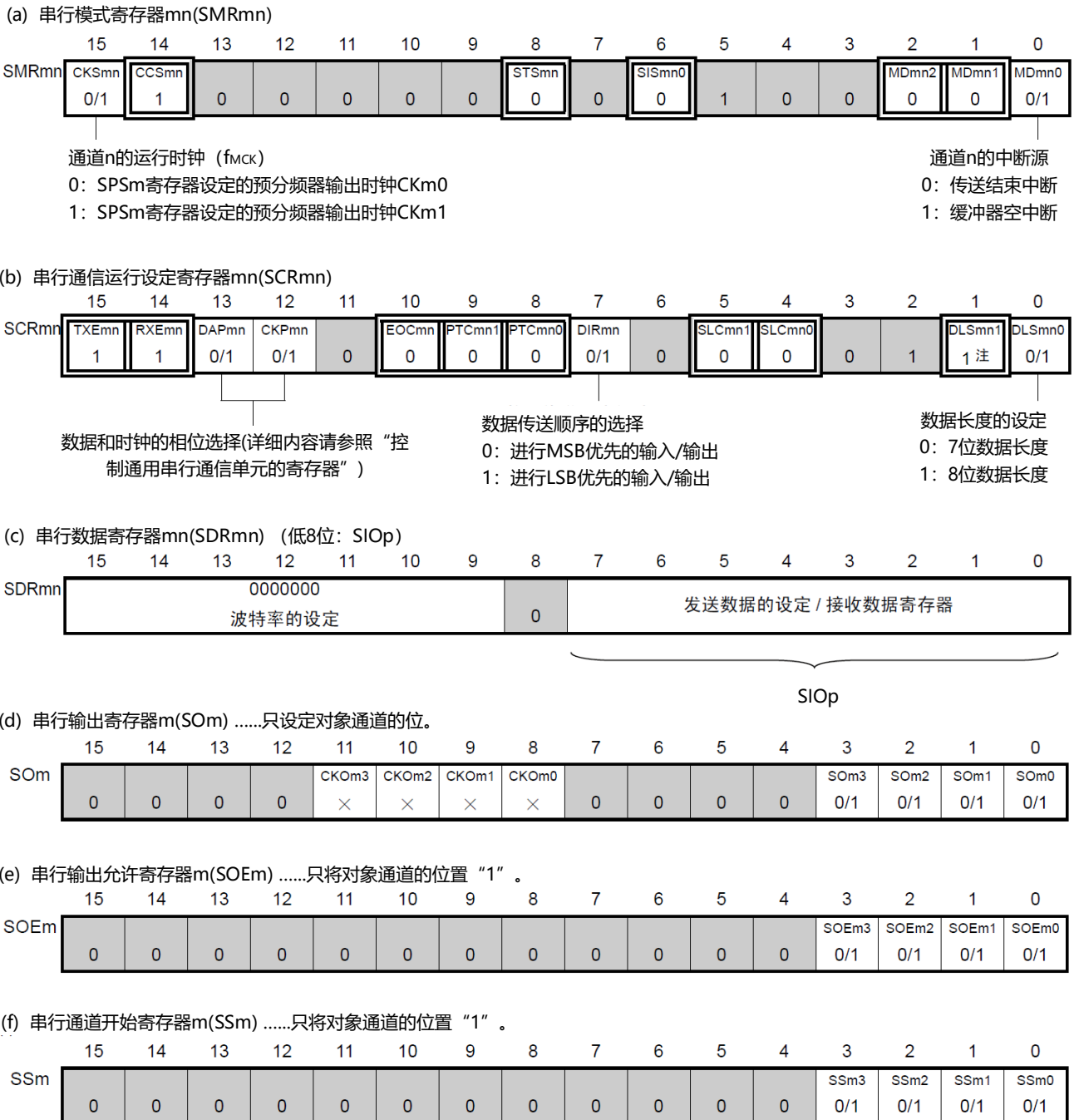
2.必须在满足此条件并且满足电特性的外围功能特性(参照数据手册)的范围内使用。

备注1. f_{MCK} : 对象通道的运行时钟频率

2.m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

(1) 寄存器的设置

图13-61 3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
从属发送和接收时的寄存器设置内容例子



注 只限于SCR00寄存器和SCR01寄存器，其他固定为“1”。

注意 在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

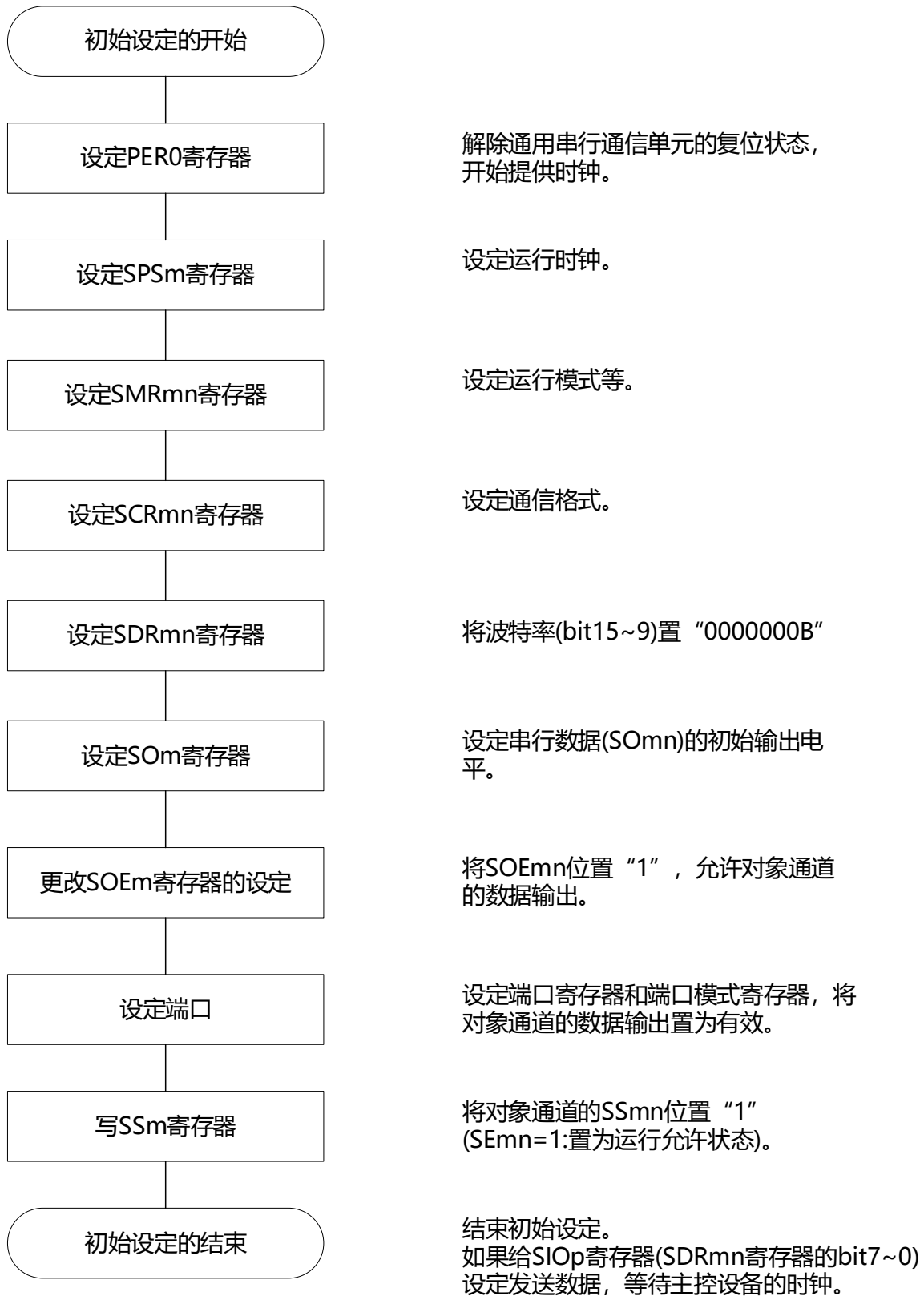
2. : 在SSPI从属发送和接收模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-62 从属发送和接收的初始设置步骤



注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

图13-63 从属发送和接收的中止步骤

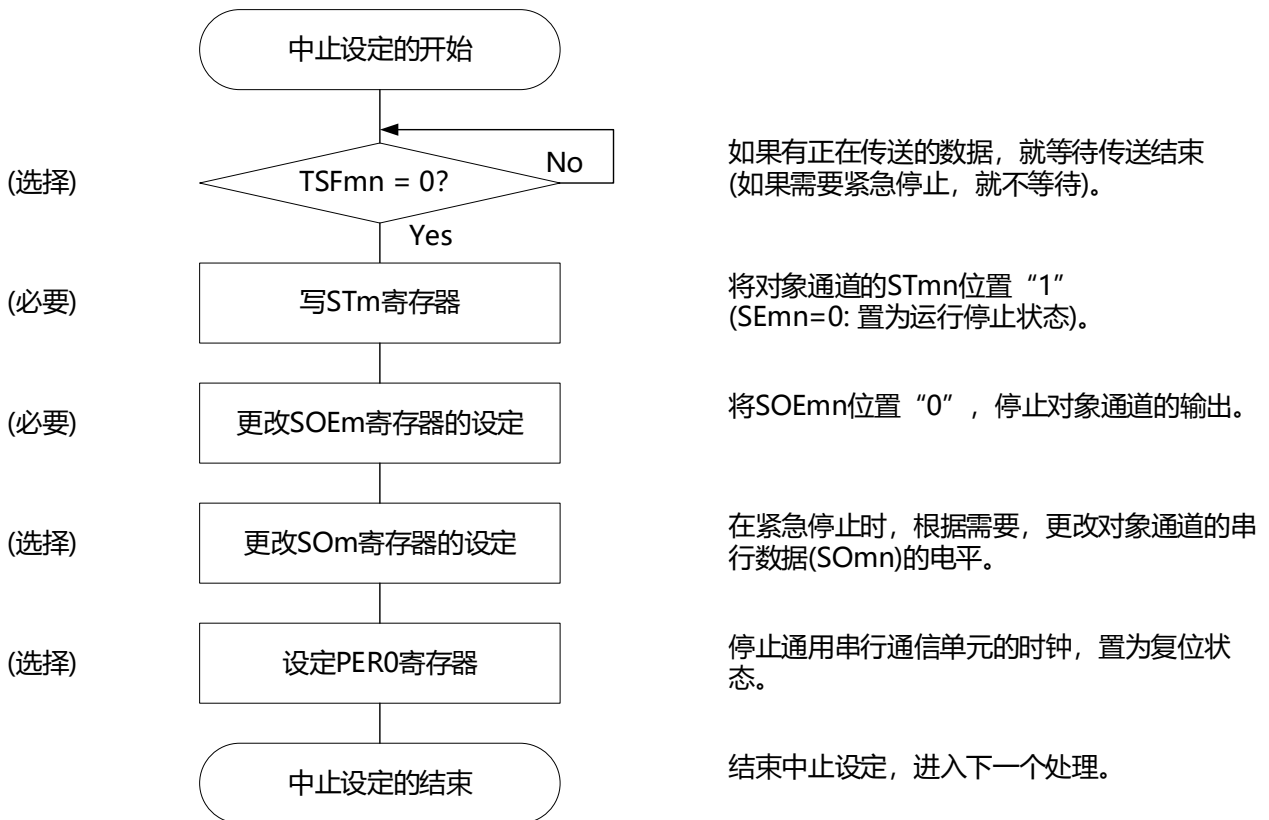
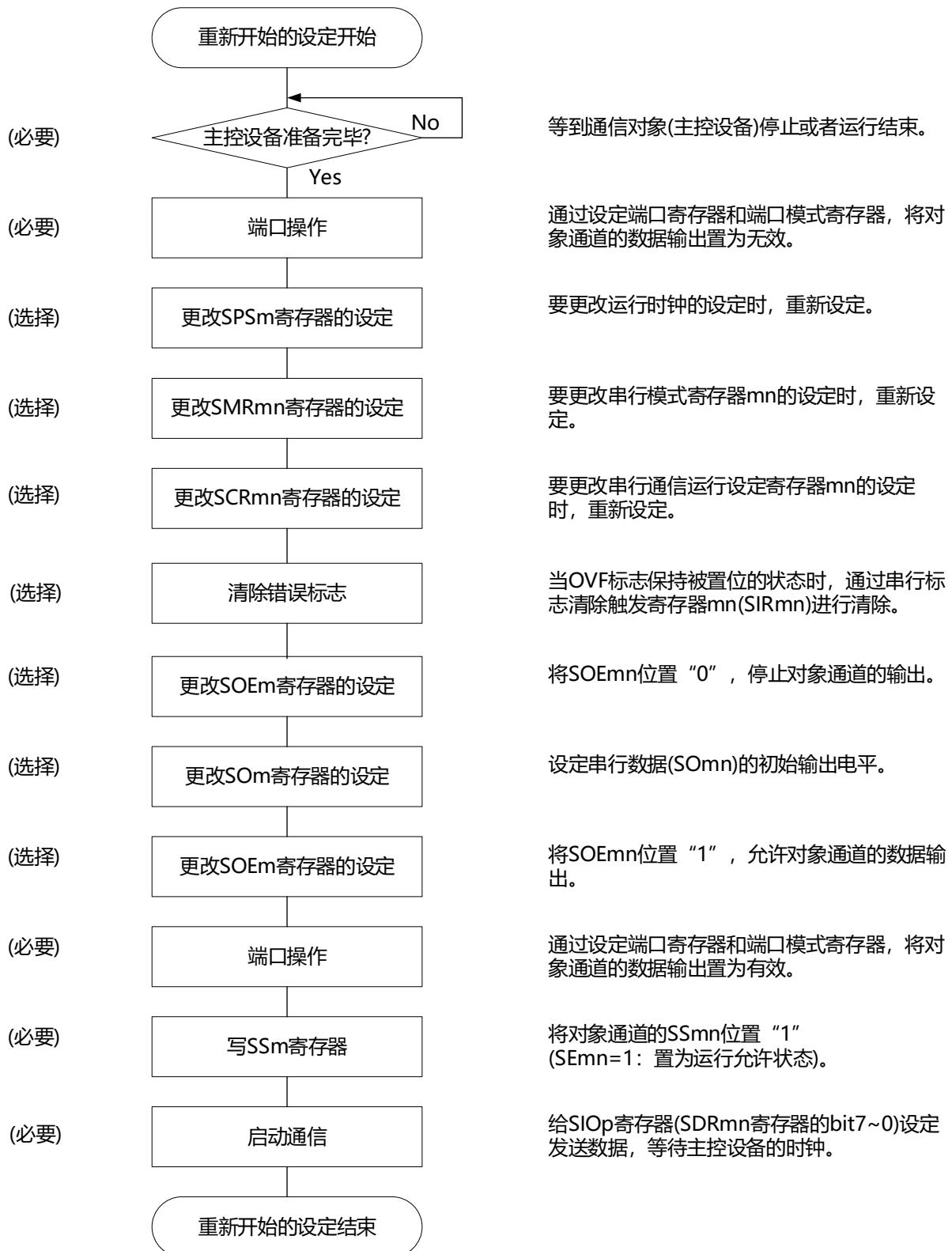


图13-64 重新开始从属发送和接收的设置步骤

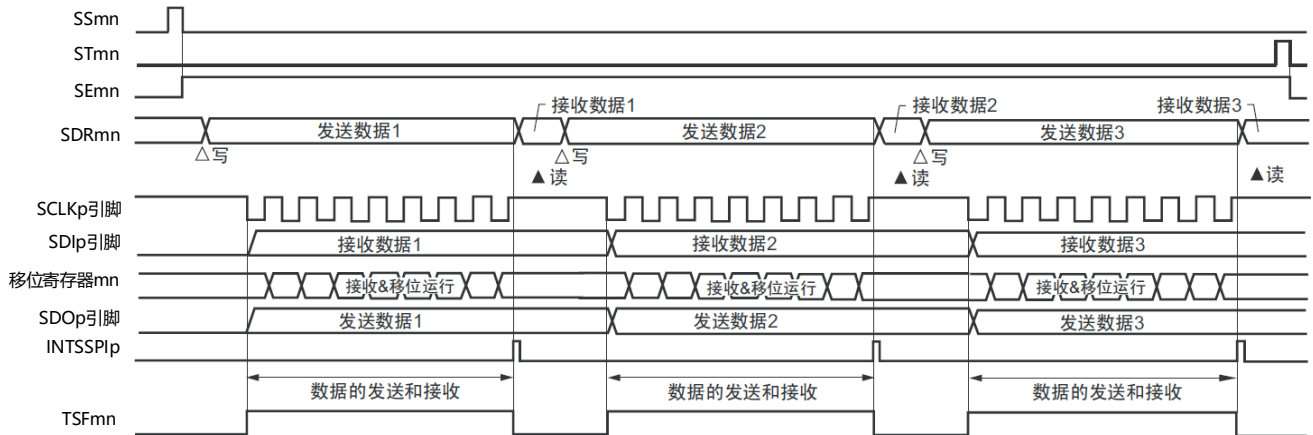


注意1. 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

2. 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始的设置。

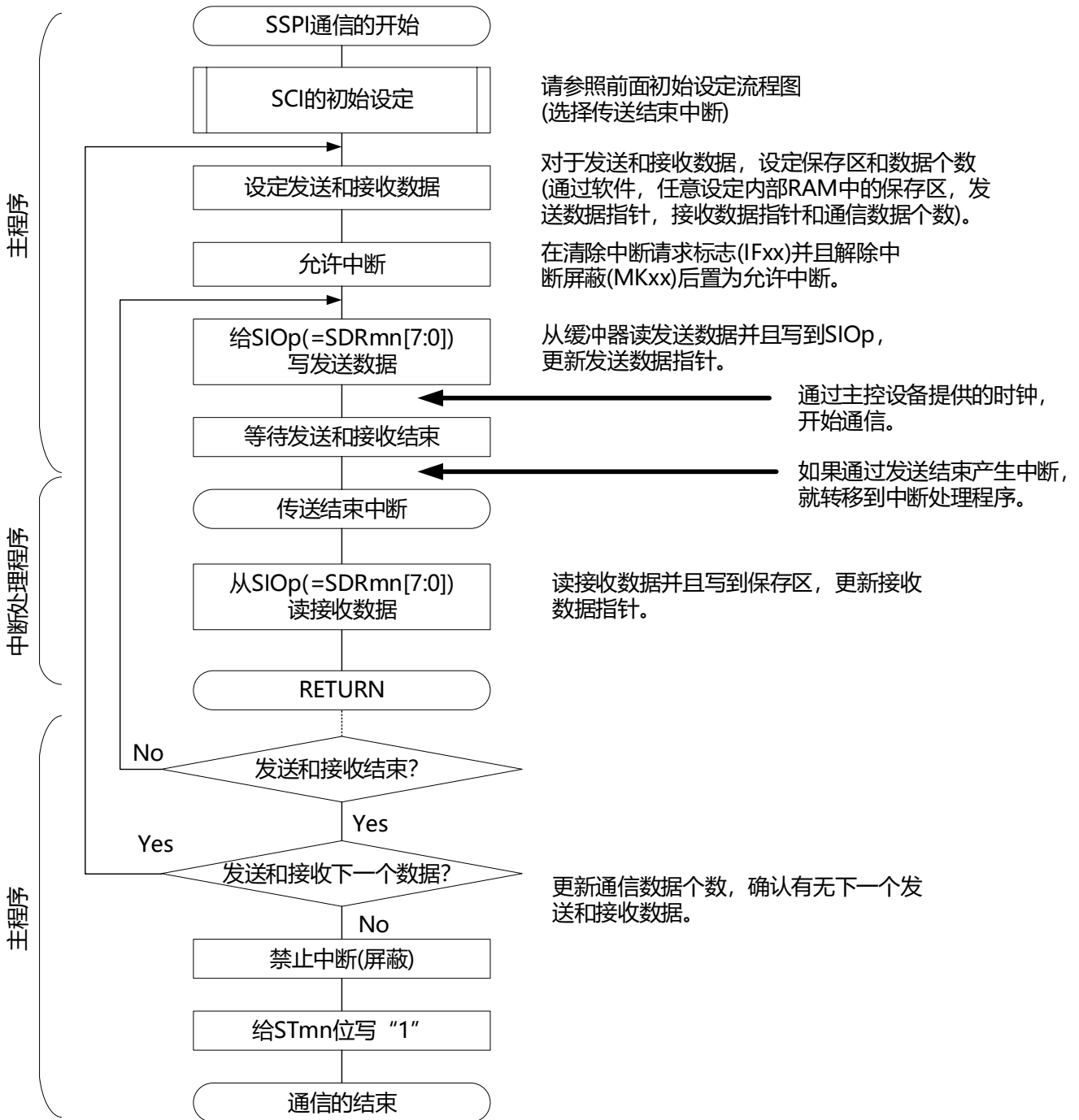
(3) 处理流程（单次发送和接收模式）

图13-65 从属发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

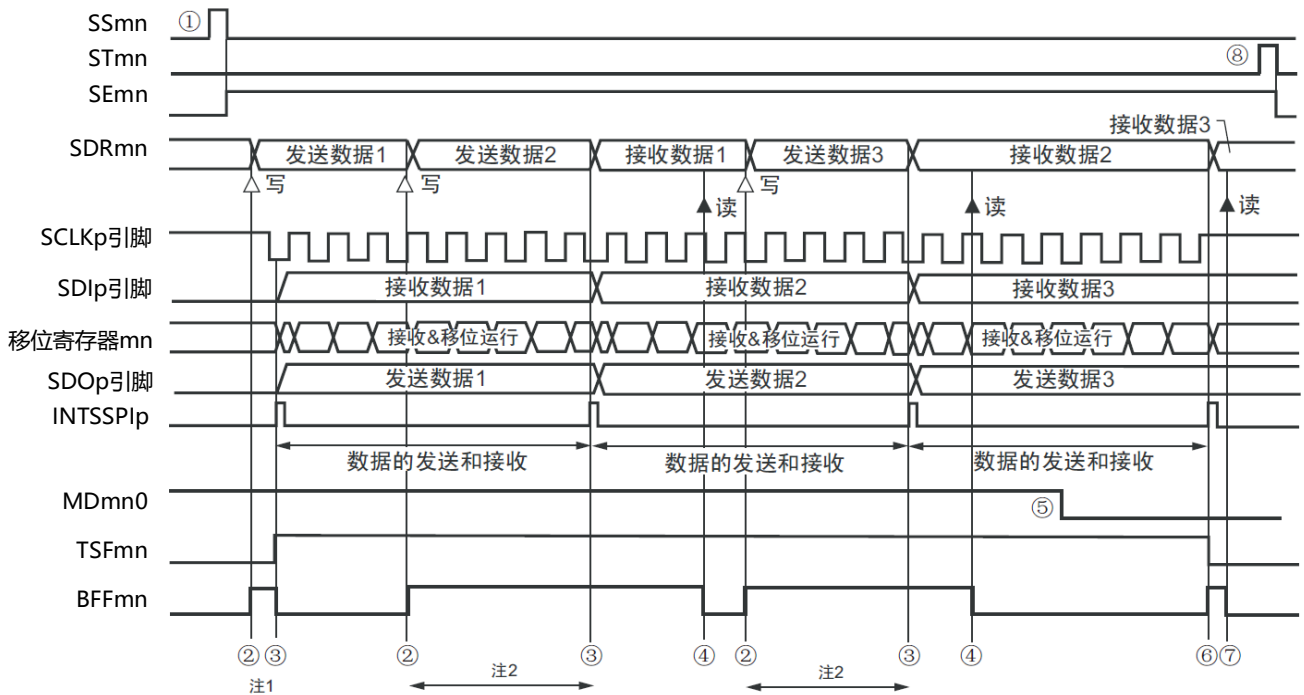
图13-66 从属发送和接收（单次发送和接收模式）的流程图



注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

(4) 处理流程（连续发送和接收模式）

图13-67 从属发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



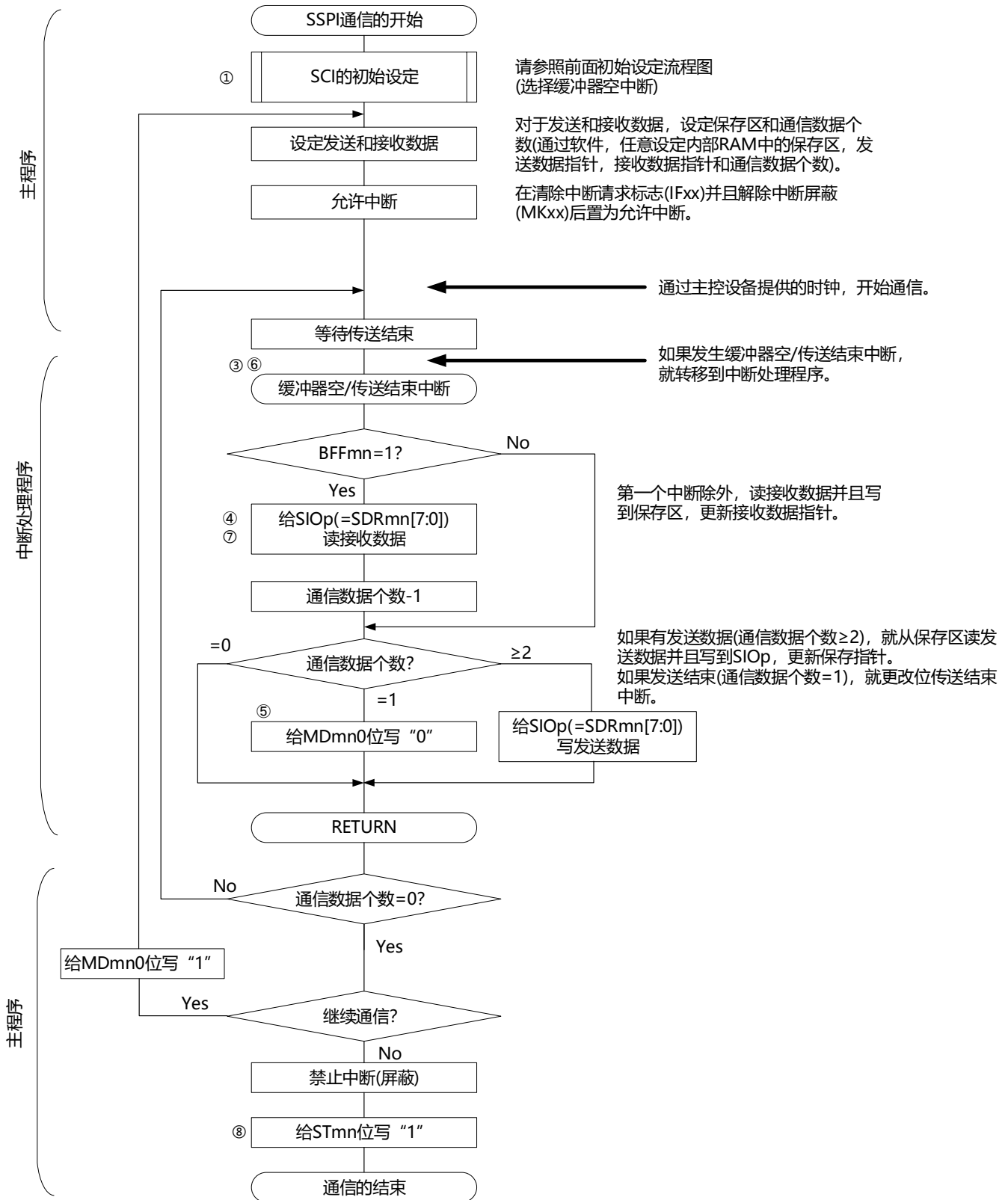
- 注 1.如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。
- 2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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

备注1.图中的①~⑧对应“图13-68从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

2.m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: SSPI号 (p=00、01、10、11、20、21)
mn=00~03、10~11

图13-68 从属发送和接收（连续发送和接收模式）的流程图



注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注 图中的①~⑧对应“图13-67从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

13.5.7 传送时钟频率的计算

3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的传送时钟频率能用以下计算式进行计算。

(1) 主控设备

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

(2) 从属设备

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

注 容许的最大传送时钟频率为 $f_{MCK}/6$ 。

备注 因为SDRmn[15:9]的值为串行数据寄存器mn（SDRmn）的bit15~9的值（0000000B~1111111B），所以为0~127。

运行时钟（f_{MCK}）取决于串行时钟选择寄存器m（SPSm）和串行模式寄存器mn（SMRmn）的bit15（CKSmn）。

表13-2 3线串行I/O运行时钟的选择

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

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器m（STm）=000FH）后进行更改。

备注1.X: 忽略

2.m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

13.5.8 在3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信过程中发生错误时的处理步骤

在3线串行I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信过程中发生错误时的处理步骤如图13-69所示。

图13-69 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn (SDIRmn) 写“1”。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器,只能清除读操作时的错误。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

13.6 从属选择输入功能的时钟同步串行通信的运行

SCI0的通道0是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- 7位或者8位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB优先的选择
- 发送和接收数据的电平设置

[时钟控制]

- 输入/输出时钟的相位控制
- 设置由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注从属通信：
 $\text{Max.f}_{\text{MCK}}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注 必须在满足SCLK周期时间 (t_{KCY}) 特性的范围内使用。详细内容请参照数据手册。

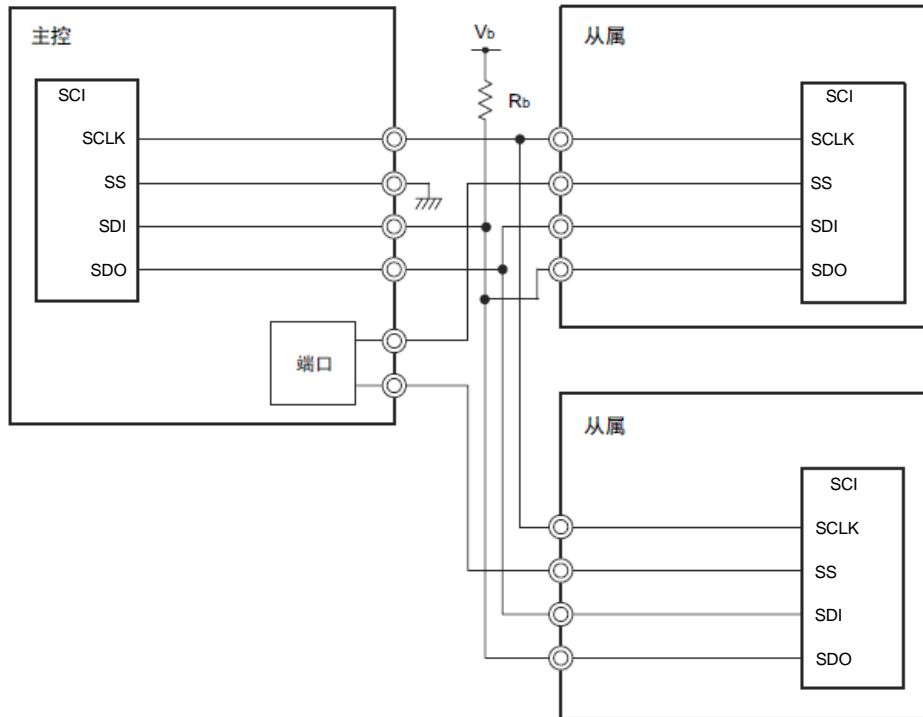
从属选择输入功能有以下3种通信运行：

- 从属发送 (参照13.6.1)
- 从属接收 (参照13.6.2)
- 从属的发送和接收 (参照13.6.3)

能通过使用从属选择输入功能，使1个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制SDO引脚的输出。当被选择为通信对象的从属设备时，SDO引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO引脚变为高电平输出，因此在连接多个从属设备的环境下需要将SDO引脚设置为Nch-O.D并且将该节点上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

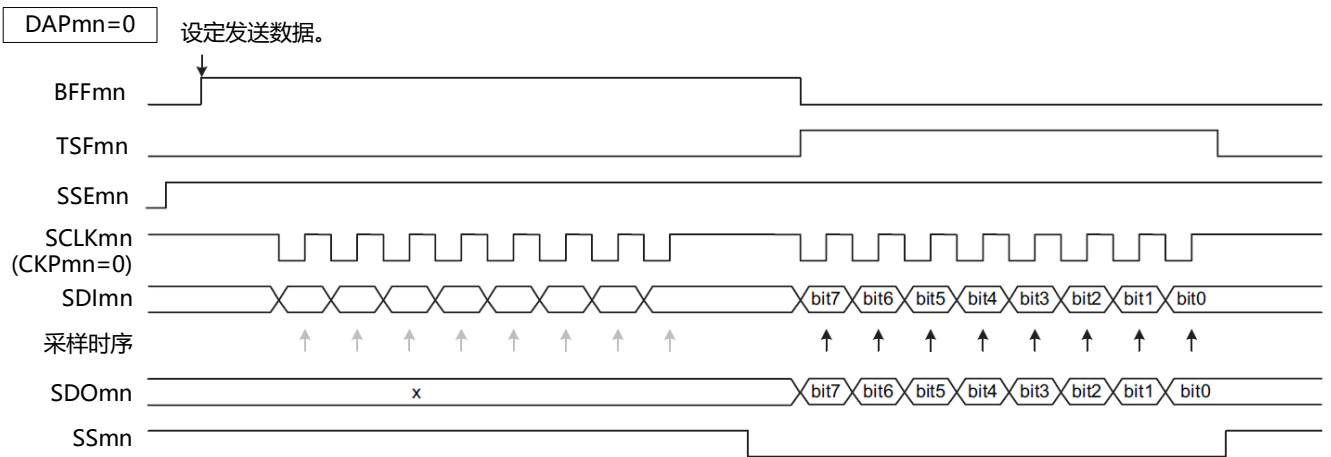
注意 必须通过端口的操作输出从属选择信号。

图13-70 从属选择输入功能的结构例子



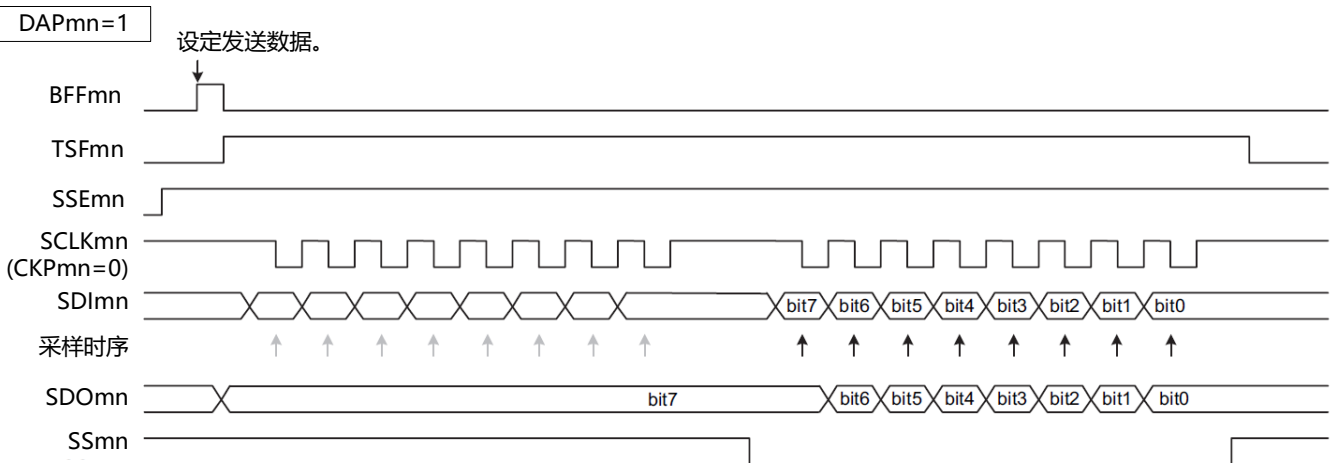
注意 将SDO00引脚选择为N沟道漏极开路输出模式。

图13-71 从属选择输入功能的时序图



在SSmn为高电平期间，即使在SCLKmn(串行时钟)的下降沿也不进行发送，而且也不进行与上升沿同步的接收数据的采样。

在SSmn为低电平期间，与串行时钟的下降沿同步输出数据(移位)并且与上升沿同步接收数据。



当DAPmn位为“1”时，如果在SSmn为高电平期间设置发送数据，就将最初的数据(bit7)提供给数据输出。但是，即使在SCLKmn(串行时钟)的上升沿也不移位，而且也不进行与下降沿同步的接受数据的采样。如果SSmn变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注 m: 单元号 (m=0) n: 通道号 (n=0)

13.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下本产品将数据发送到其他设备的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位或者8位
传送速率	Max. $f_{MCK}/6$ [Hz] ^{注1、2}
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 •DAPmn=0：在串行时钟开始运行时，开始数据输出。 •DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择功能的运行。

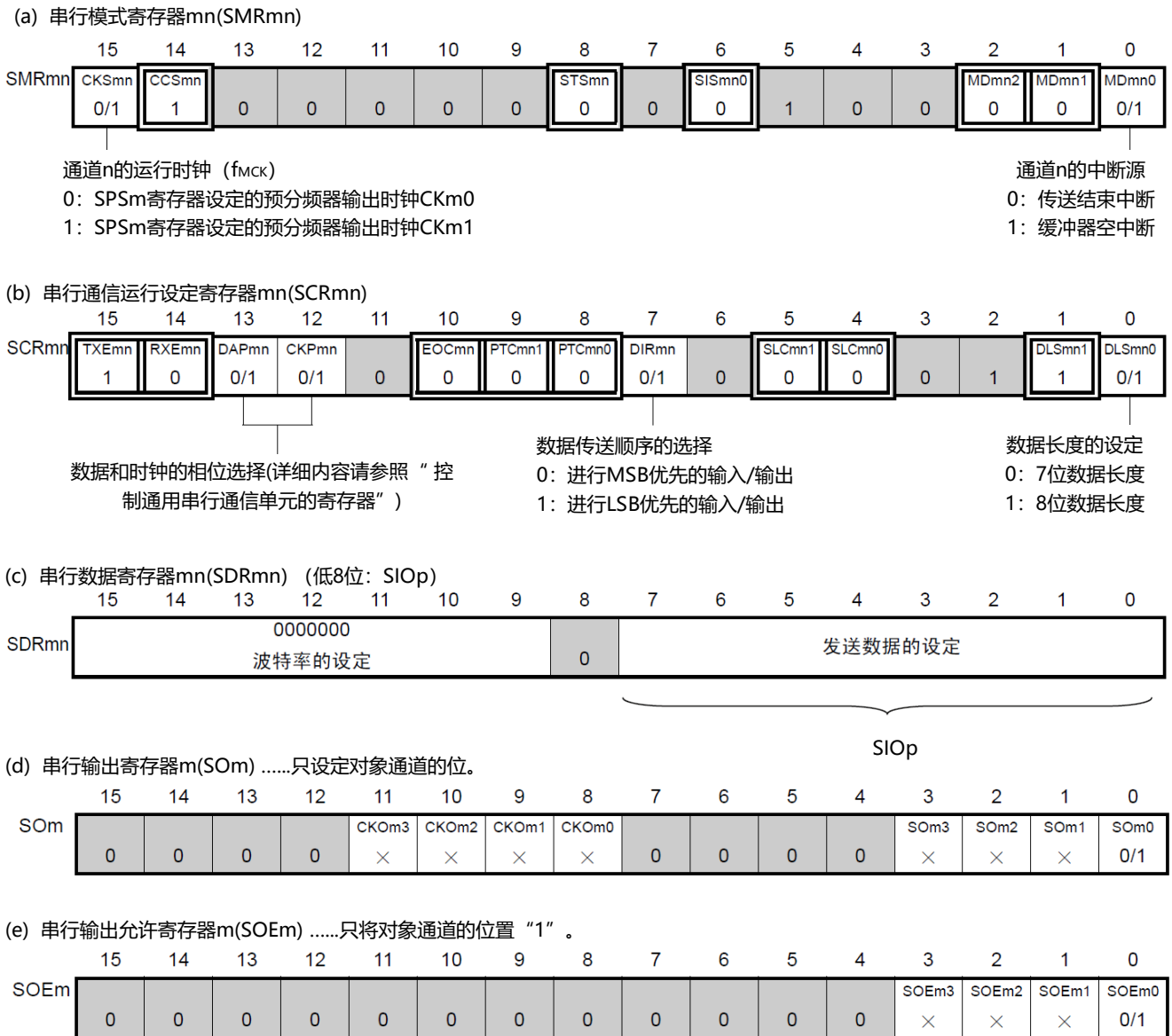
- 注 1.因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6$ [Hz]。
 2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注1. f_{MCK} ：对象通道的运行时钟频率

2.m：单元号（m=0） n：通道号（n=0）

(1) 寄存器的设置

图13-72 从属选择输入功能 (SSPI00) 从属发送时的寄存器设置内容例子(1/2)



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

2. : 在SSPI从属发送模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

图13-72 从属选择输入功能 (SSPI00) 从属发送时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器(SSm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													×	×	×	0/1

(g) 输入切换控制寄存器(ISC)这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

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

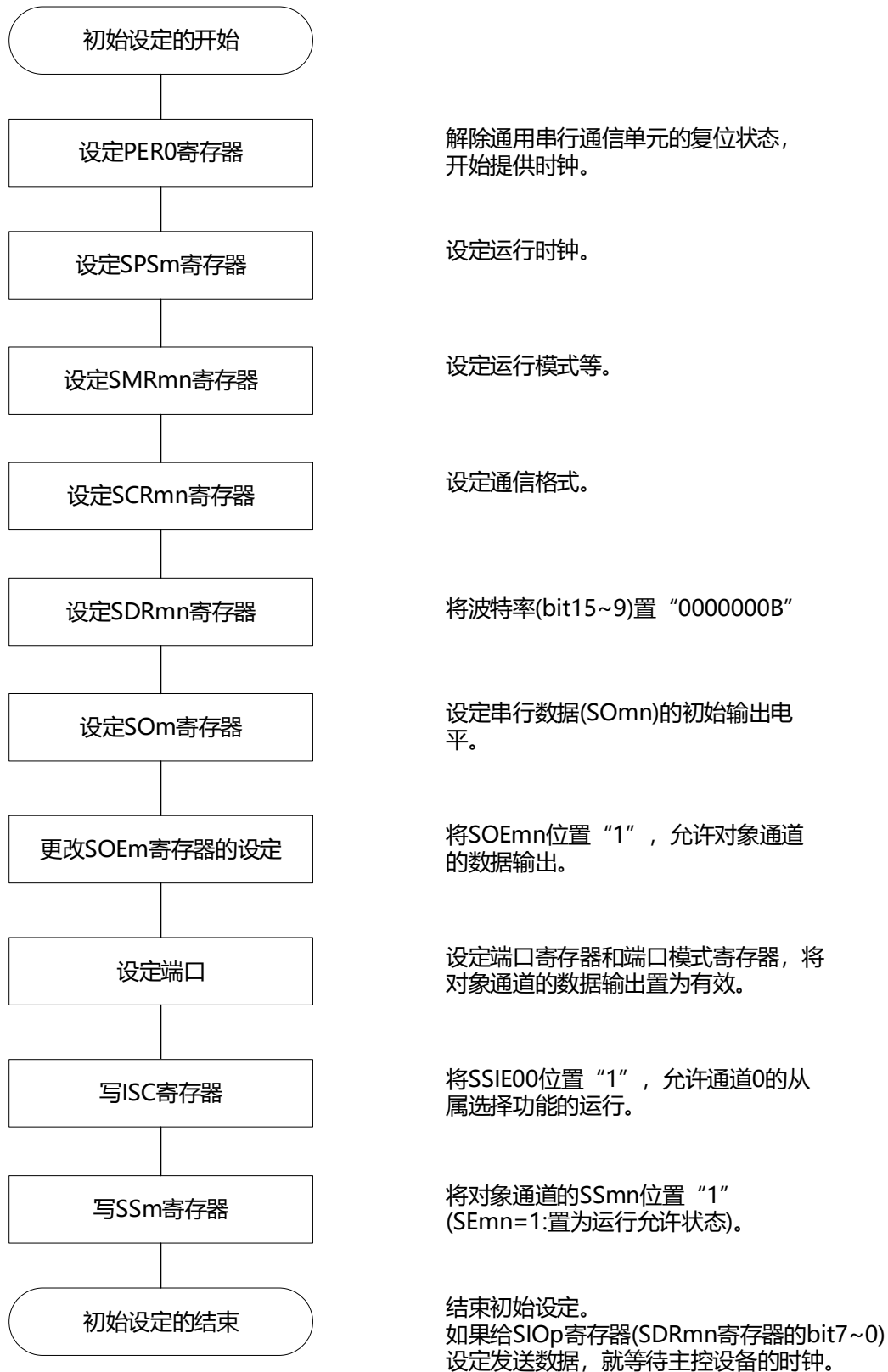
2 : 在SSPI从属发送模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

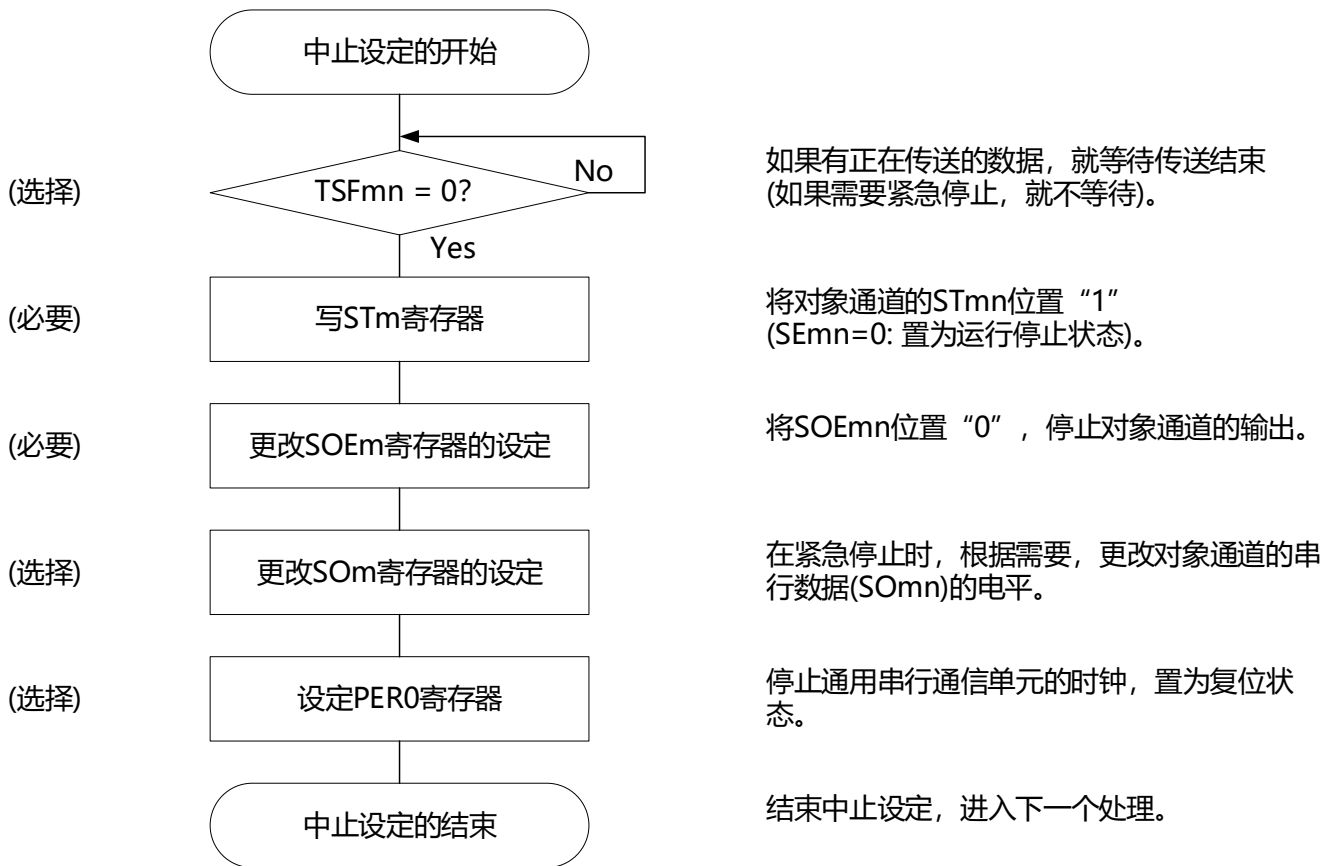
(2) 操作步骤

图13-73 从属发送的初始设置步骤



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

图13-74 从属发送的中止步骤



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

备注1.如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象 (主控设备) 停止或者通信结束后进行初始设置而不是进行重新开始设置。

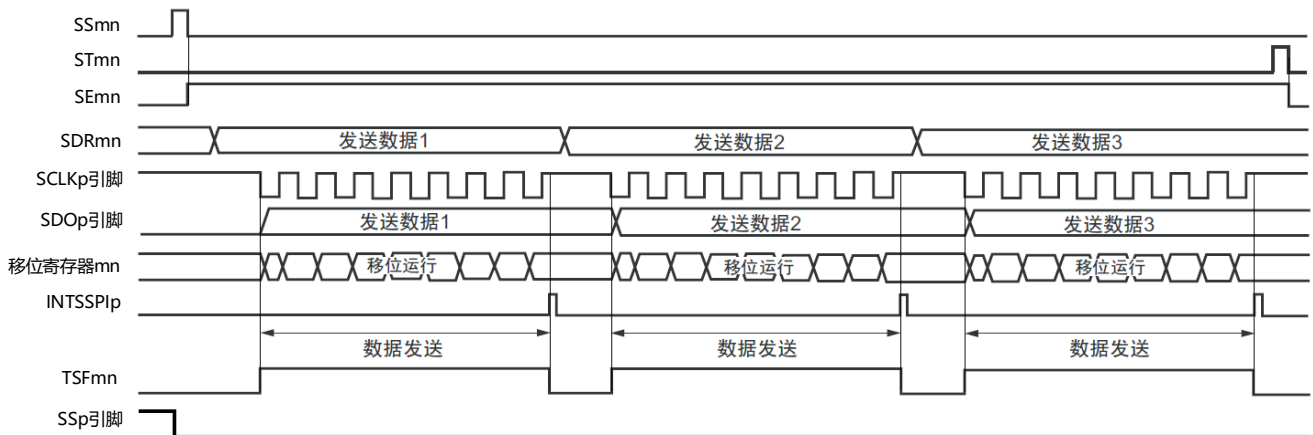
2.m: 单元号 (m=0) n: 通道号 (n=0) p: SSPI号 (p=00)

图13-75 重新开始从属发送的设置步骤



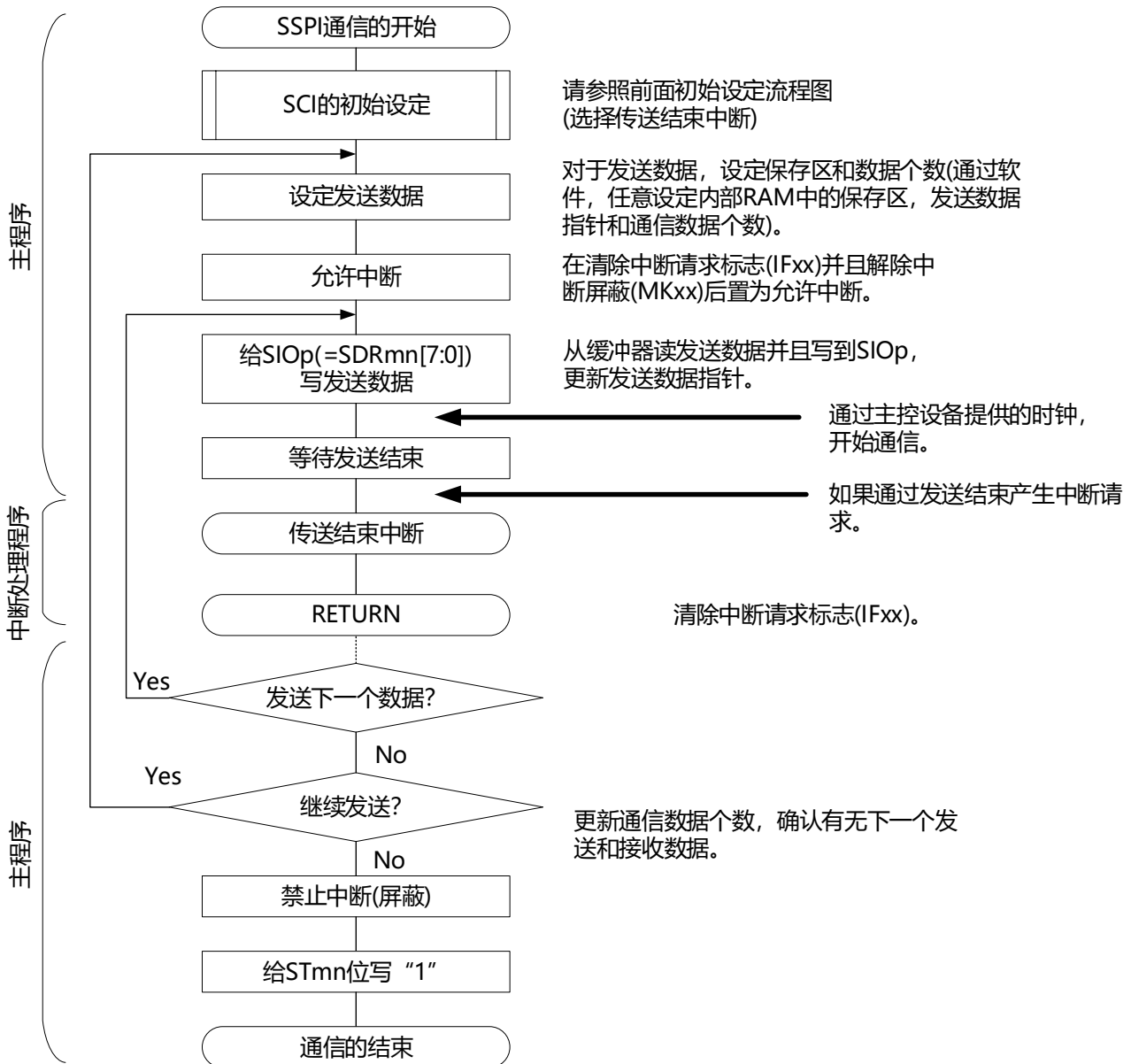
(3) 处理流程（单次发送模式）

图13-76 从属发送（单次发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）



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

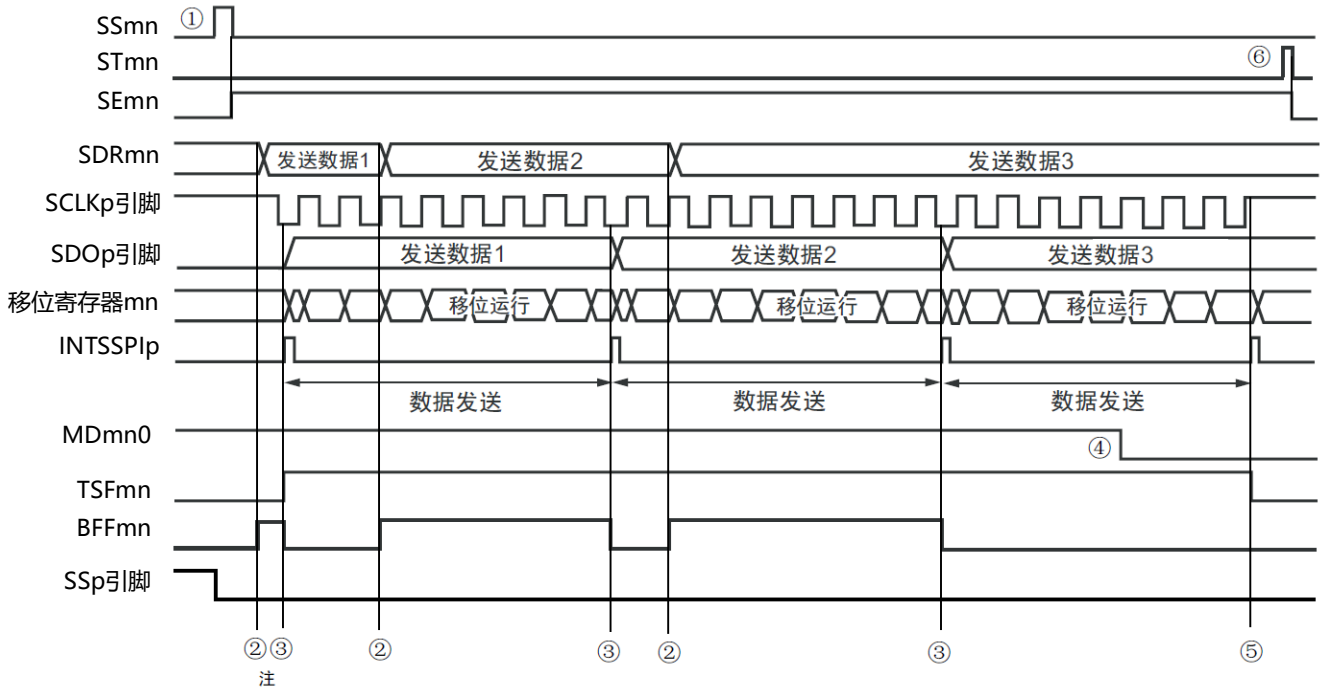
图13-77 从属发送（单次发送模式）的流程图



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

(4) 处理流程（连续发送模式）

图13-78 从属发送（连续发送模式）的时序图（类型1：DAPmn=0、CKPmn=0）

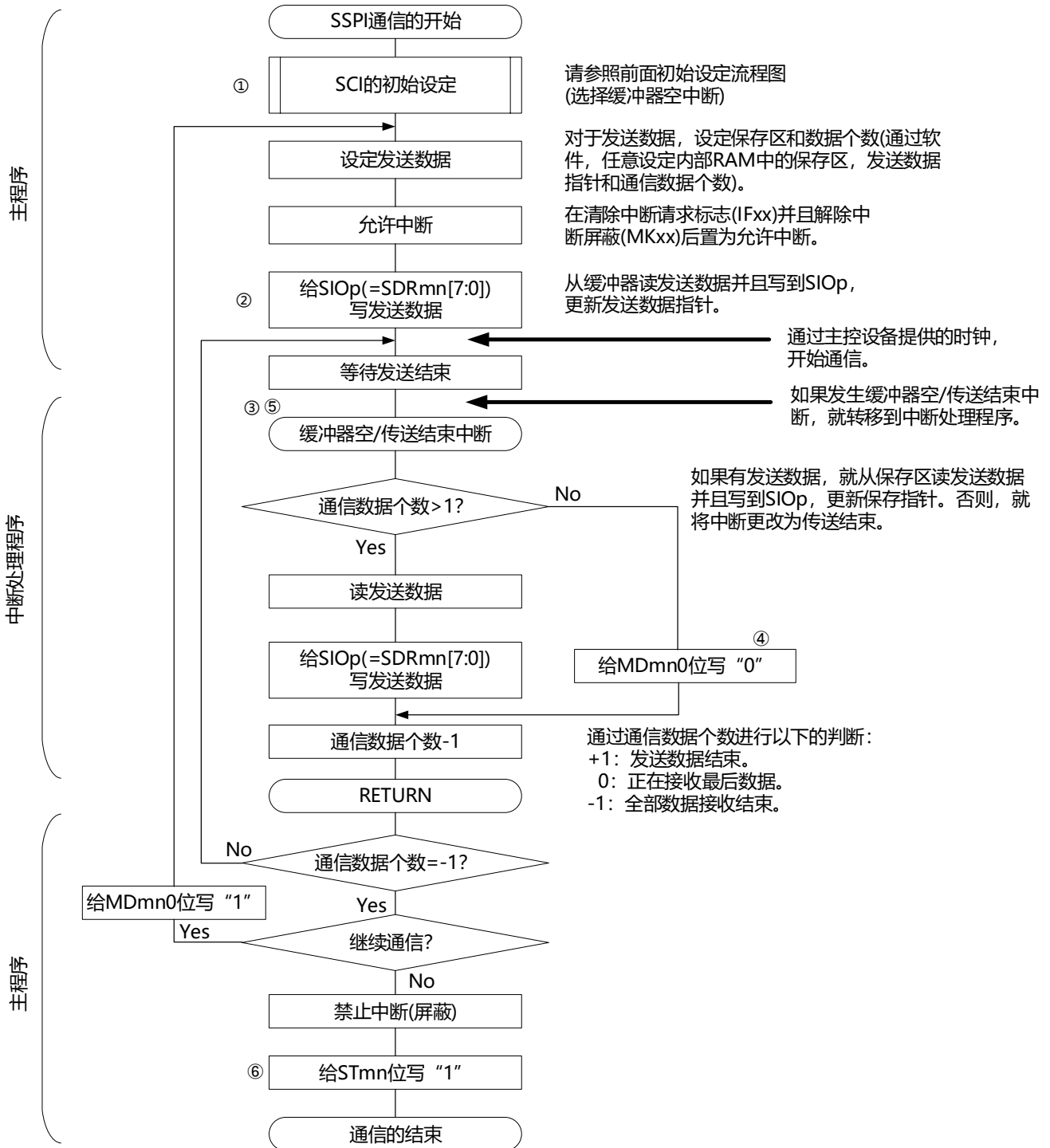


注 如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。

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

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

图13-79 从属发送（连续发送模式）的流程图



备注1.图中的①~⑥对应“图13-78从属发送（连续发送模式）的时序图”中的①~⑥。

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

13.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDI00、SS00
中断	INTSSPI00
	只限于传送结束中断（禁止设置缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位或者8位
传送速率	$\text{Max}f_{\text{MCK}}/6[\text{Hz}]$ 注1、2
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

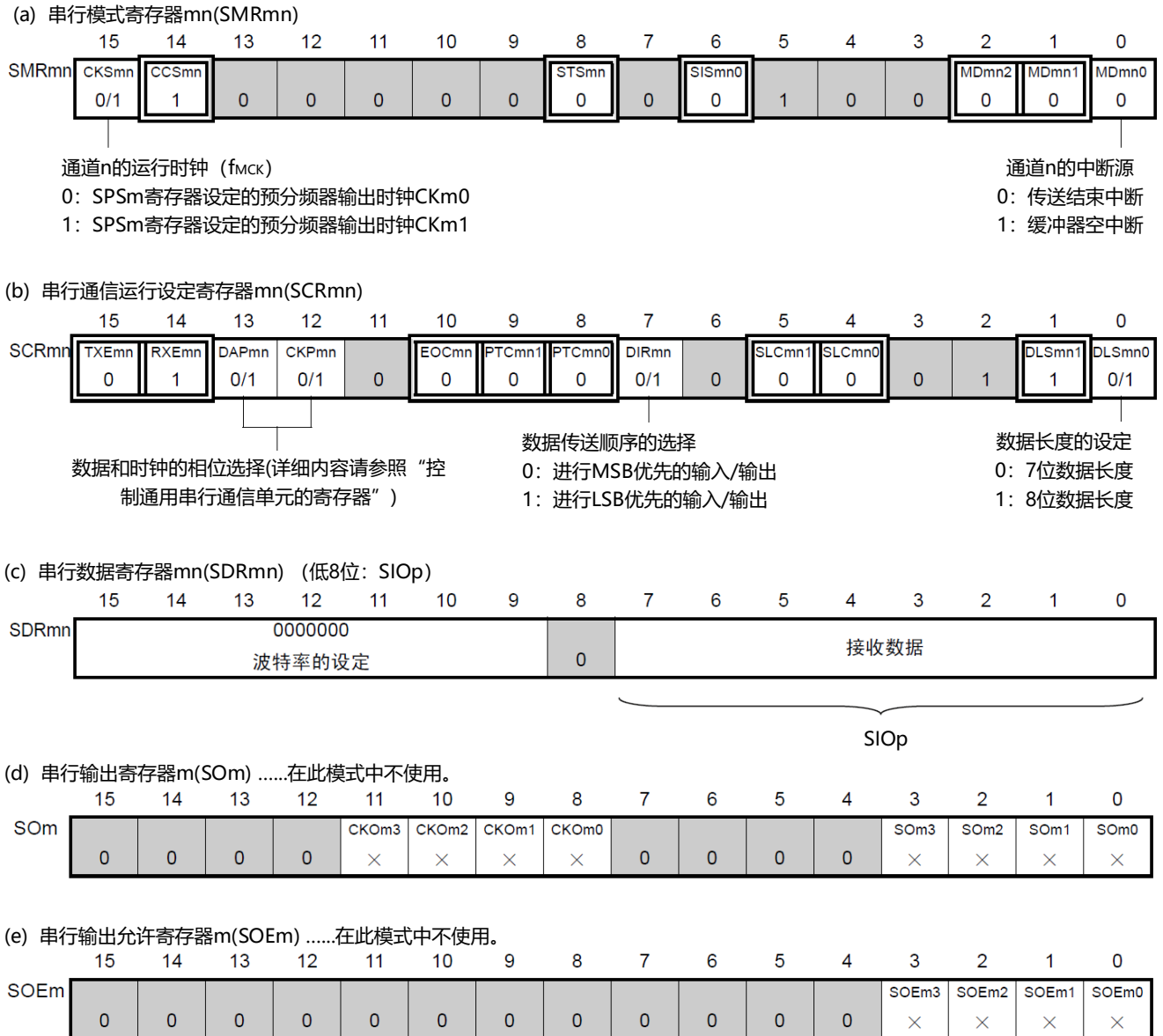
- 注 1.因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{\text{MCK}}/6[\text{Hz}]$ 。
 2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注1. f_{MCK} ：对象通道的运行时钟频率

2.m：单元号（m=0） n：通道号（n=0）

(1) 寄存器的设置

图13-80 从属选择输入功能 (SSPI00) 从属接收时的寄存器设置内容例子(1/2)



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

- 2 : 在从属接收模式中为固定设置 : 不能设置 (设置初始值)。
- ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设置初始值)。
- 0/1: 根据用户的用途置“0”或者“1”。

图13-81 从属选择输入功能 (SSPI00) 从属接收时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													×	×	×	0/1

(g) 输入切换控制寄存器(ISC)这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效
 1: SS00引脚的输入值有效

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

2. : 在从属接收模式中为固定设置 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-82 从属接收的初始设置步骤

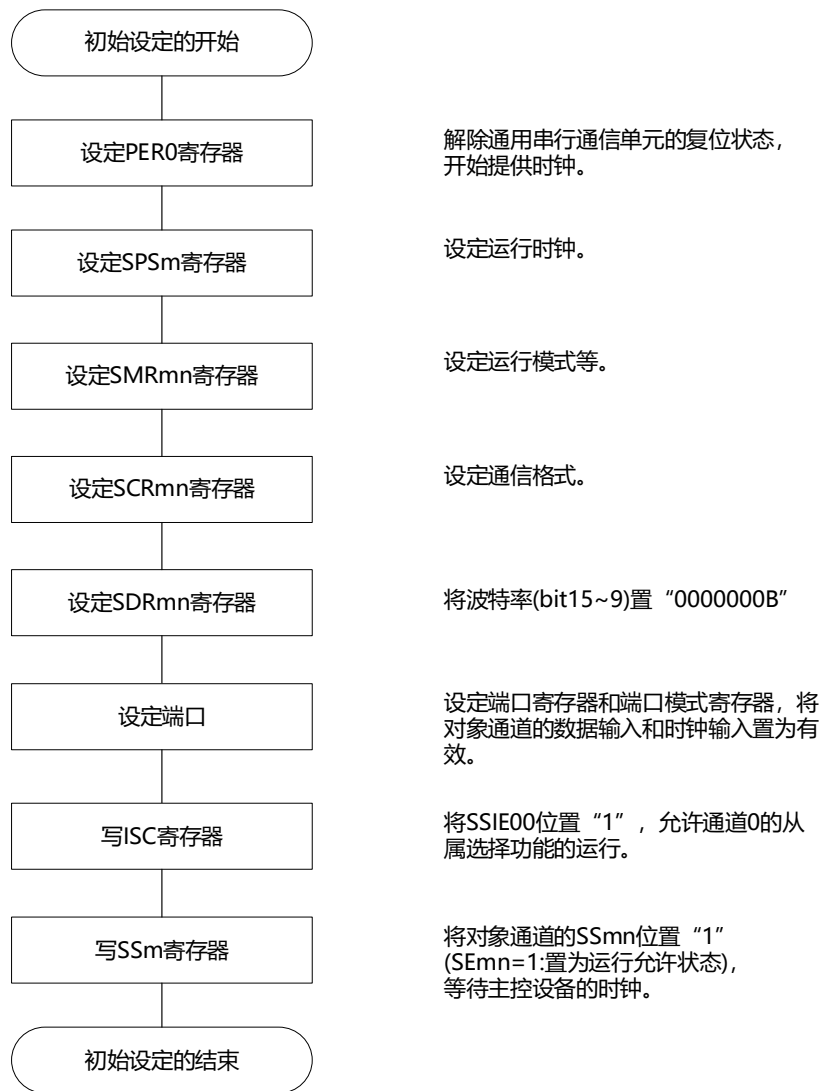
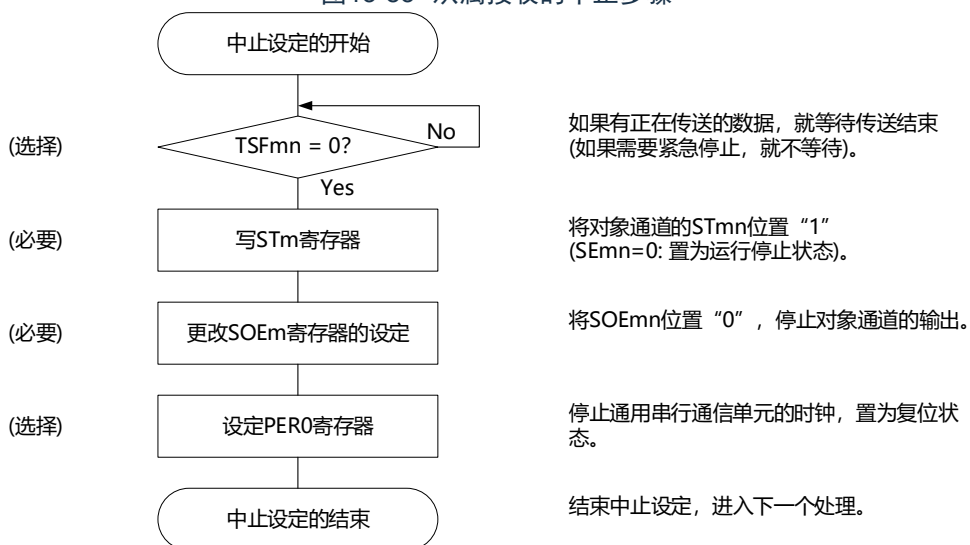


图13-83 从属接收的中止步骤



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

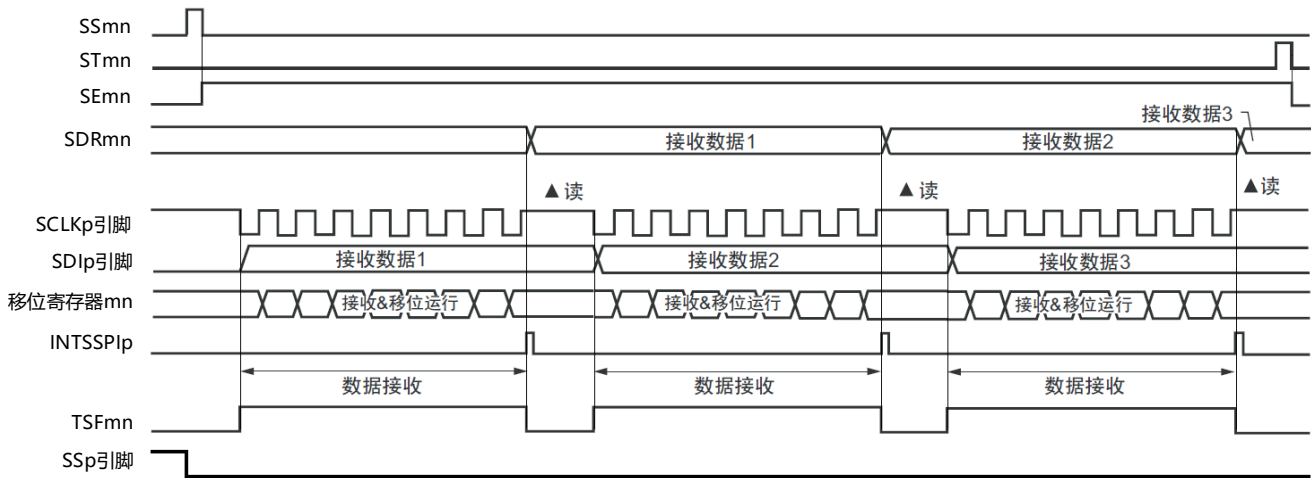
图13-84 重新开始从属接收的设置步骤



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

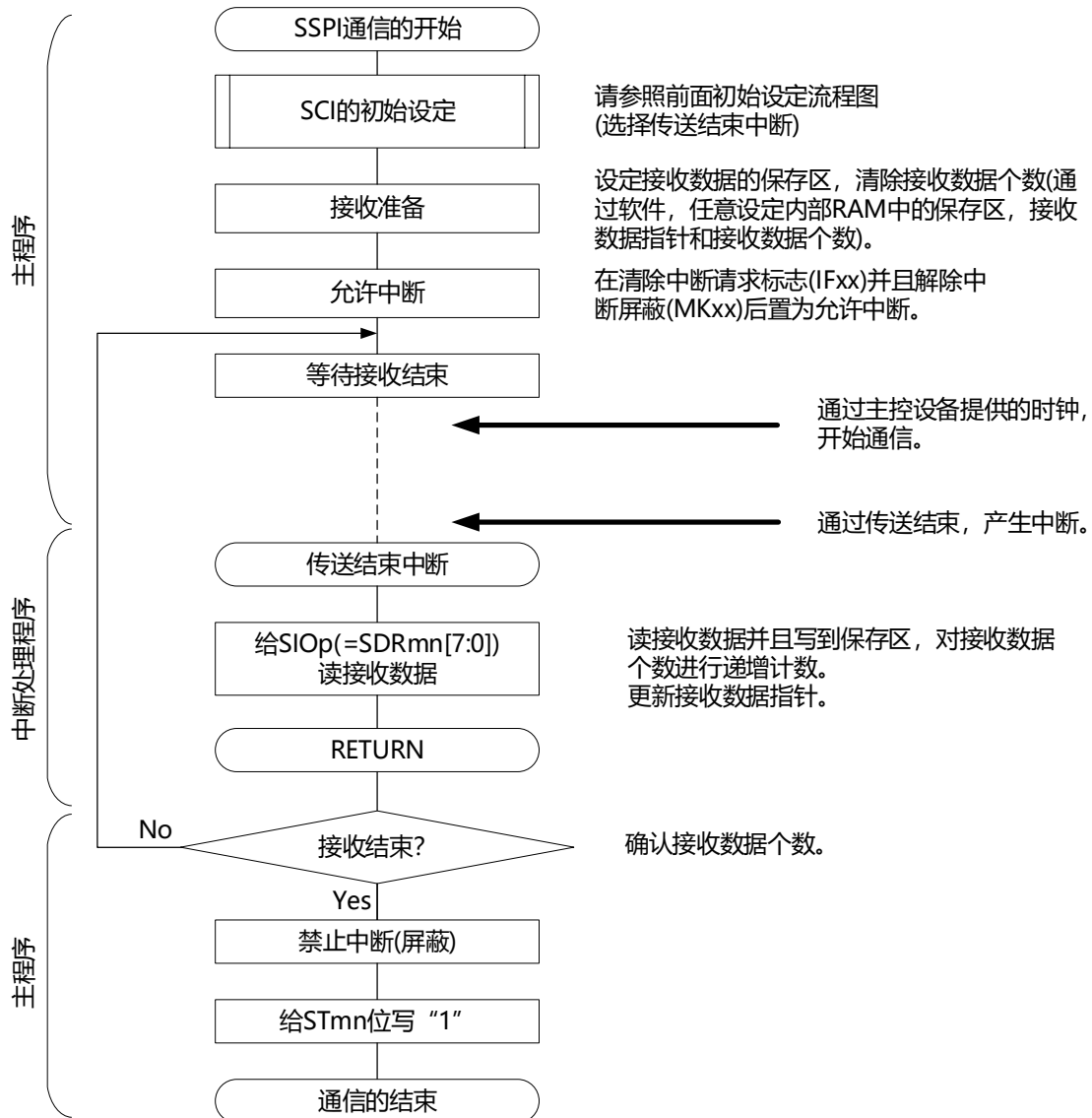
(3) 处理流程（单次接收模式）

图13-85 从属接收（单次接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



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

图13-86 从属接收（单次接收模式）的流程图



13.6.3 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品和其他设备进行数据发送和接收的运行。

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLK00、SDI00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位或者8位
传送速率	Max. $f_{MCK}/6$ [Hz]注1、2
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 •DAPmn=0: 在串行时钟开始运行时，开始数据输出。 •DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1.因为在内部对SCLK00引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 $f_{MCK}/6$ [Hz]。

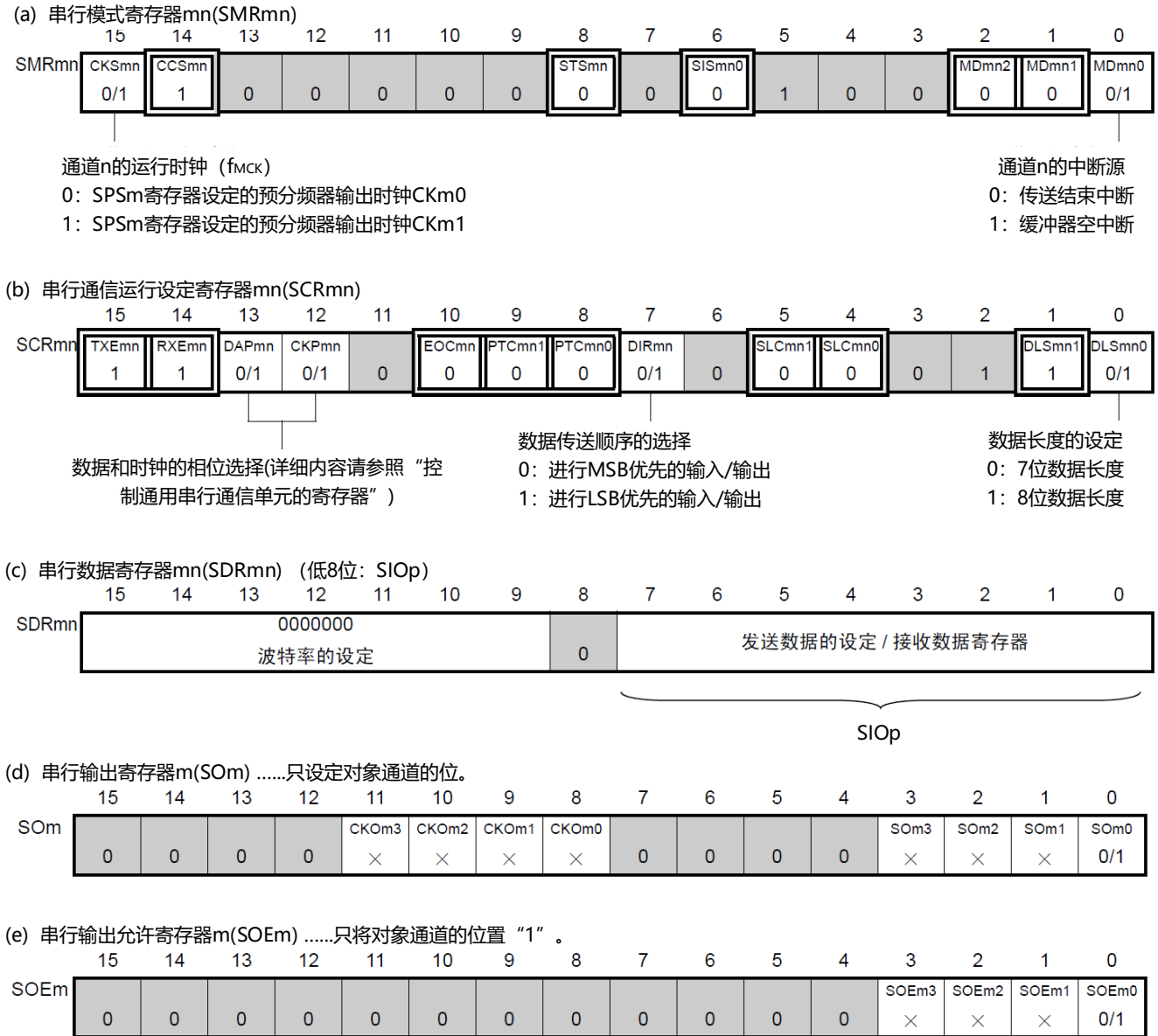
2.必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注1. f_{MCK} : 对象通道的运行时钟频率

2.m: 单元号 (m=0) n: 通道号 (n=0)

(1) 寄存器的设置

图13-87 从属选择输入功能 (SSPI00) 从属发送和接收时的寄存器设置内容例子(1/2)



注意 在主控设备开始输出时钟前，必须给SIOp寄存器设置发送数据。

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

- 2□ : 在从属接收模式中为固定设置 □ : 不能设置 (设置初始值)。
- × : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设置初始值)。
- 0/1: 根据用户的用途置“0”或者“1”。

图13-87 从属选择输入功能 (SSPI00) 从属发送和接收时的寄存器设置内容例子(2/2)

(f) 串行通道开始寄存器(SSm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	0/1

(g) 输入切换控制寄存器(ISC)这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

注意 在主控设备开始输出时钟前，必须给SIOP寄存器设置发送数据。

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

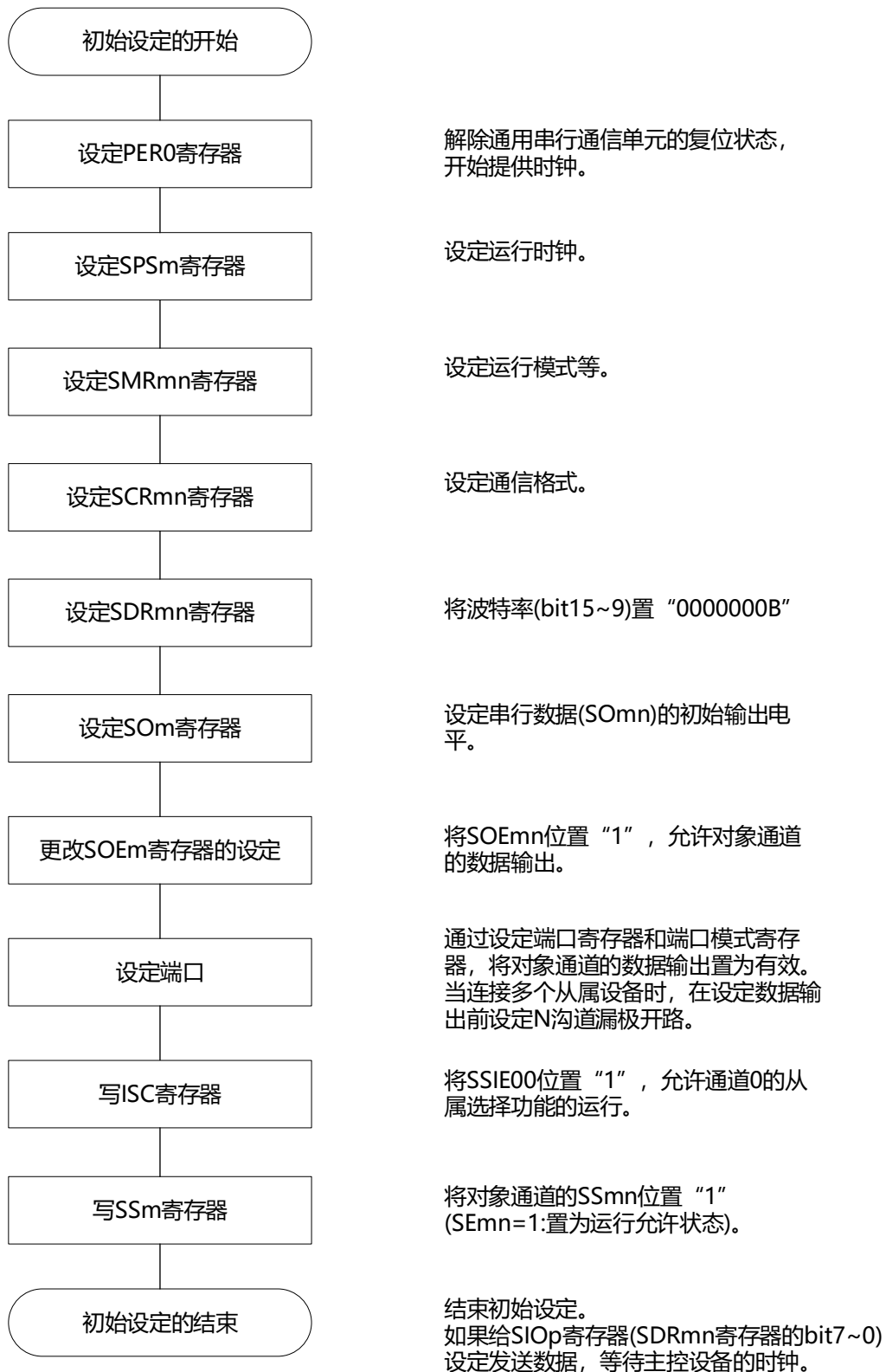
2□ : 在从属接收模式中为固定设置 □ : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

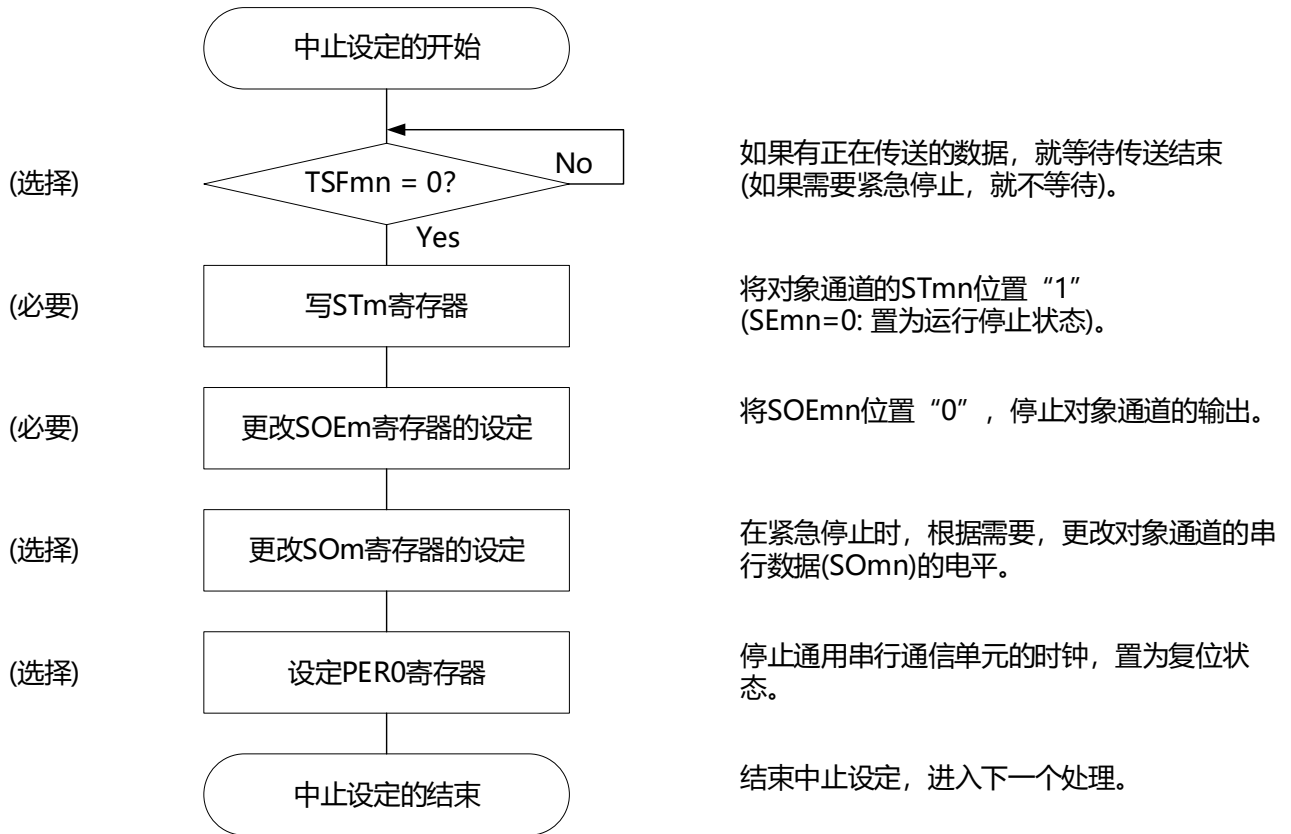
图13-88 从属发送和接收的初始设置步骤



注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

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

图13-89 从属发送和接收的中止步骤



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

图13-90 重新开始从属发送和接收的设置步骤

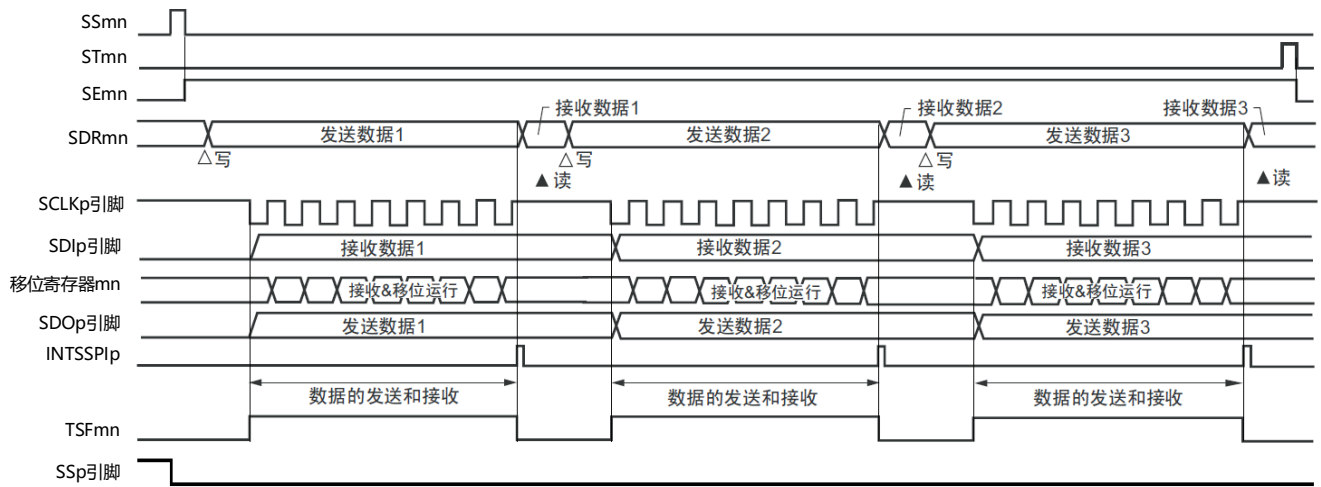


注意1. 在主机设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

2. 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象(主控设备)停止或者通信结束后进行初始设置而不是进行重新开始设置。

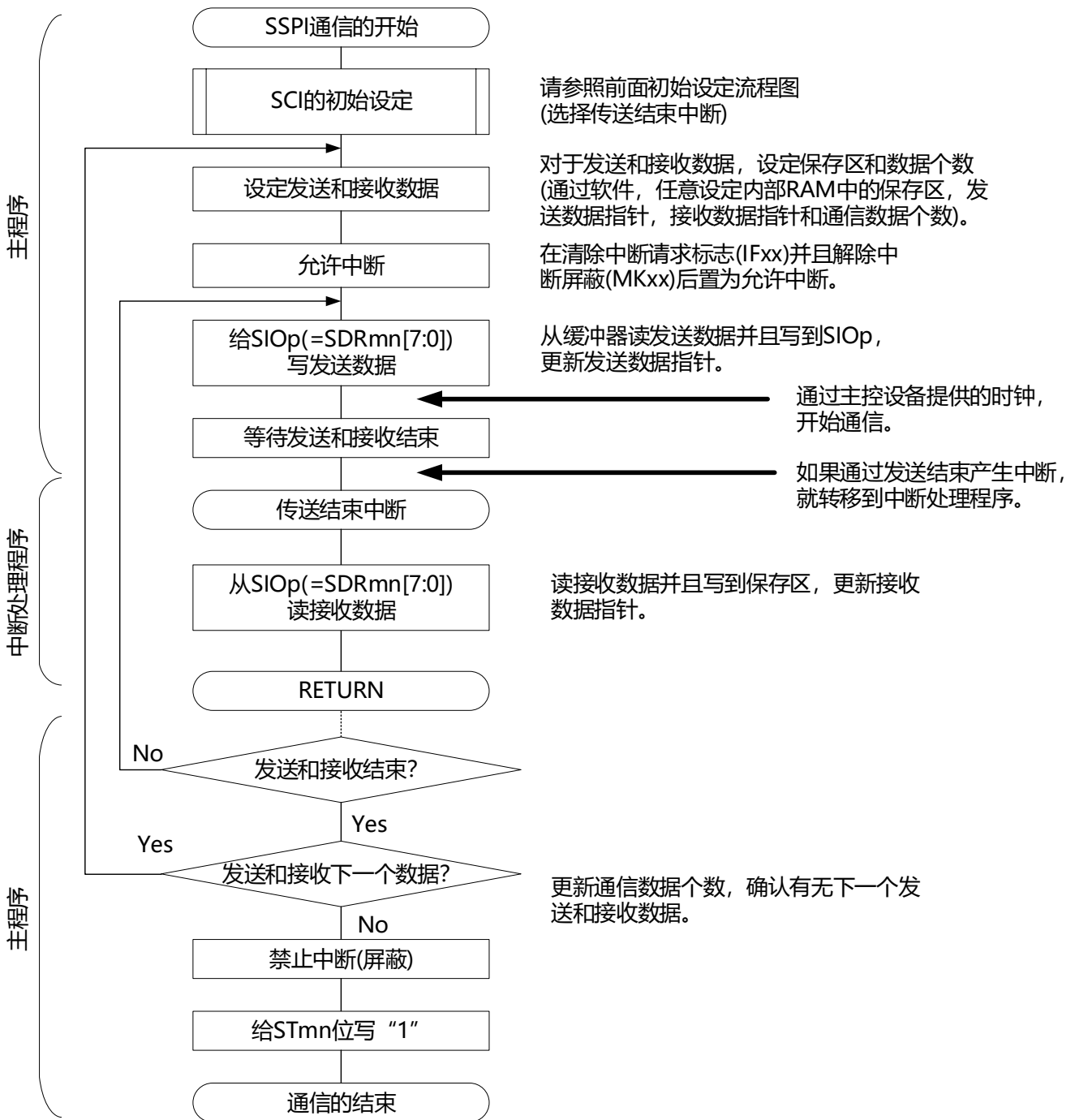
(3) 处理流程（单次发送和接收模式）

图13-91 从属发送和接收（单次发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



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

图13-92 从属发送和接收（单次发送和接收模式）的流程图

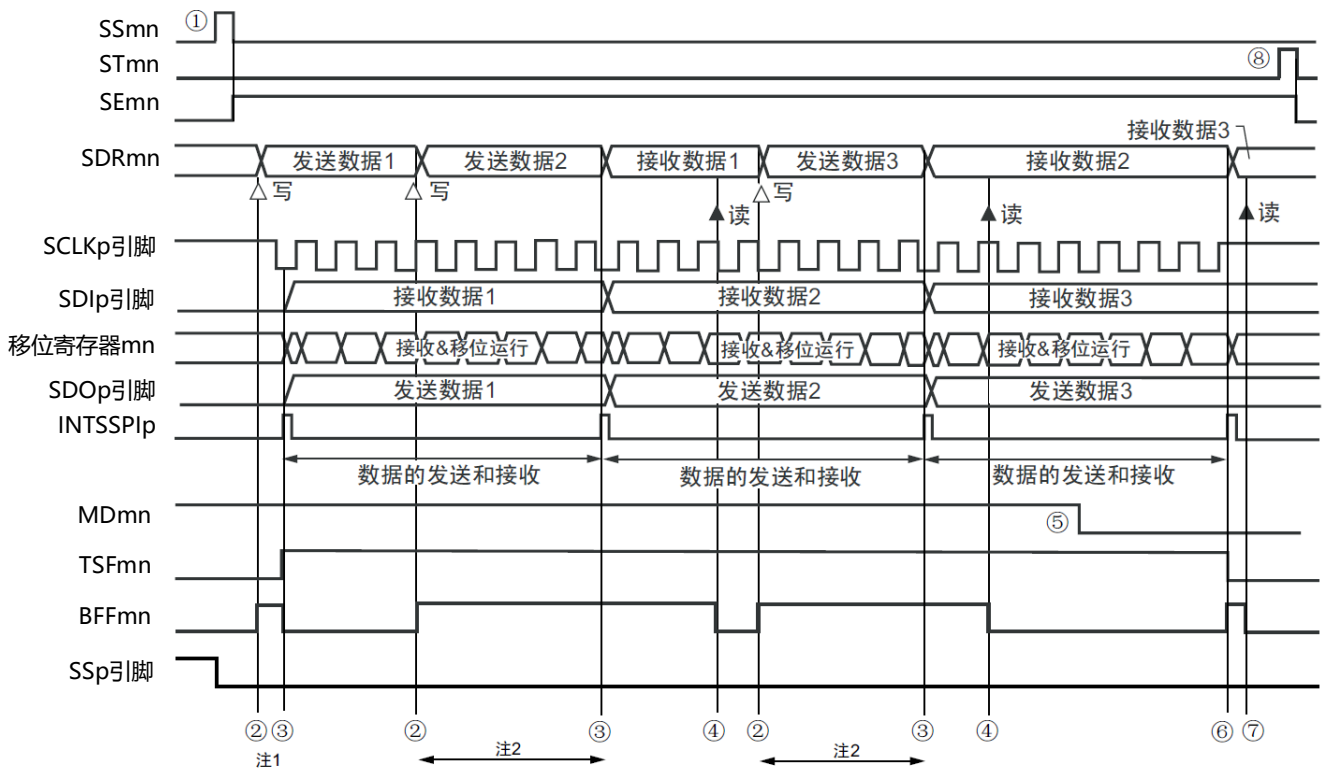


注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

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

(4) 处理流程（连续发送和接收模式）

图13-93 从属发送和接收（连续发送和接收模式）的时序图（类型1：DAPmn=0、CKPmn=0）



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

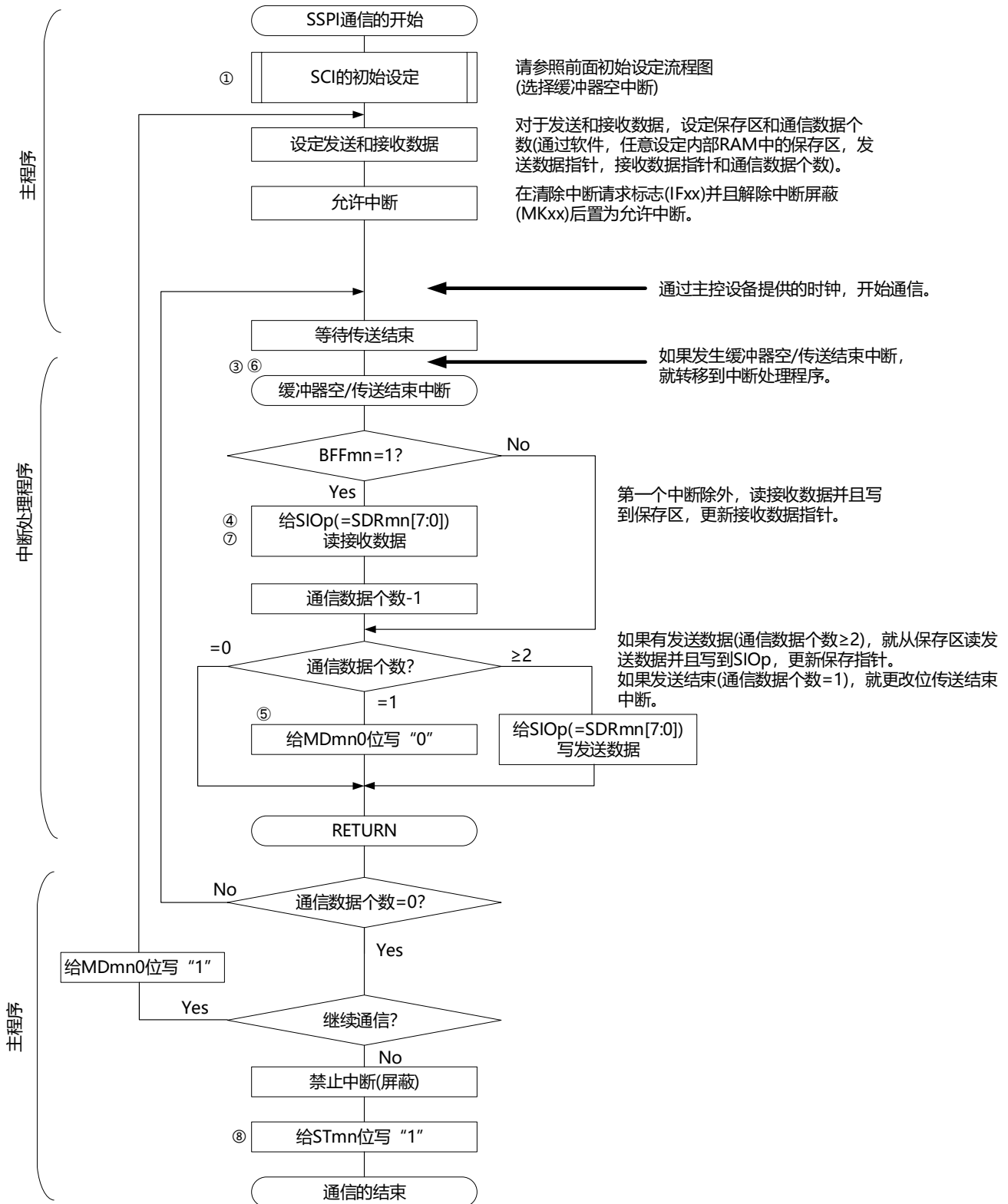
2.如果在此期间读取SDRmn寄存器，就能读发送数据。此时，不影响传送运行。

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

备注1.图中的①~⑧对应“图13-94从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

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

图13-94 从属发送和接收（连续发送和接收模式）的流程图



注意 在主控设备开始输出时钟前, 必须给SIOp寄存器设置发送数据。

备注1.图中的①~⑧对应“图13-93从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

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

13.6.4 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

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

注 容许的最大传送时钟频率为 $f_{MCK}/6$ 。

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

表13-3 从属选择输入功能运行时钟的选择

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

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器m（STm）=000FH）后进行更改。

备注1.X: 忽略

2.m: 单元号 (m=0) n: 通道号 (n=0)

13.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如图13-95所示。

图13-95 溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn (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)

13.7 UART (UART0~UART2) 通信的运行

这是通过串行数据发送 (TxD) 和串行数据接收 (RxD) 共2条线进行异步通信的功能。使用这2条通信线,按数据帧(由起始位、数据、奇偶校验位和停止位构成)与其他通信方进行异步(使用内部波特率)的数据发送和接收。能通过使用发送专用(偶数通道)和接收专用(奇数通道)共2个通道来实现全双工异步UART通信。

[数据的发送和接收]

- 7位、8位或者9位的数据长度^注
- MSB/LSB优先的选择
- 发送和接收数据的电平设置(选择电平是否反相)
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位的检测功能

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误和溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

注 只有UART0支持9位数据长度。

UART0使用SCI0的通道0和通道1。

UART1使用SCI0的通道2和通道3。

UART2使用SCI1的通道0和通道1。

各通道任意选择一个功能使用,除了所选功能以外,其他功能不能运行。

例如,在单元0的通道0和通道1使用UART0时,不能使用SSPI00。但是,在使用UART0的同时,不同通道的通道2和通道3能使用SSPI10、UART1。

注意 当用作UART时,发送方(偶数通道)和接收方(奇数通道)只能用于UART。

UART有以下2种通信运行:

- UART发送 (参照13.7.1)
- UART接收 (参照13.7.2)

13.7.1 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。

UART	UART0	UART1	UART2
对象通道	SCI0的通道0	SCI0的通道2	SCI1的通道0
使用的引脚	TxD0	TxD1	TxD2
中断	INTST0	INTST1	INTST2
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	无		
传送数据长度	7位、8位或者9位 ^{注1}		
传送速率	$\text{Max.f}_{\text{MCK}}/6[\text{bps}]$ ($\text{SDRmn}[15:9] \geq 2$)、 $\text{Min.f}_{\text{CLK}}/(2 \cdot 2^{15} \cdot 128)[\text{bps}]$ ^{注2}		
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> • 无奇偶校验位。 • 附加零校验。 • 附加偶校验。 • 附加奇校验。 		
停止位	可选择以下内容： <ul style="list-style-type: none"> • 附加1位。 • 附加2位。 		
数据方向	MSB优先或者LSB优先		

注 1.只有UART0支持9位数据长度。

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

备注1. f_{MCK} ：对象通道的运行时钟频率

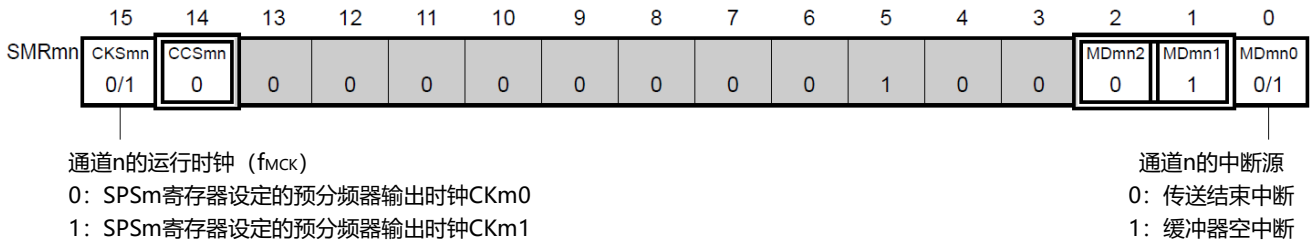
f_{CLK} ：系统时钟频率

2.m：单元号（m=0、1） n：通道号（n=0、2） mn=00、02、10

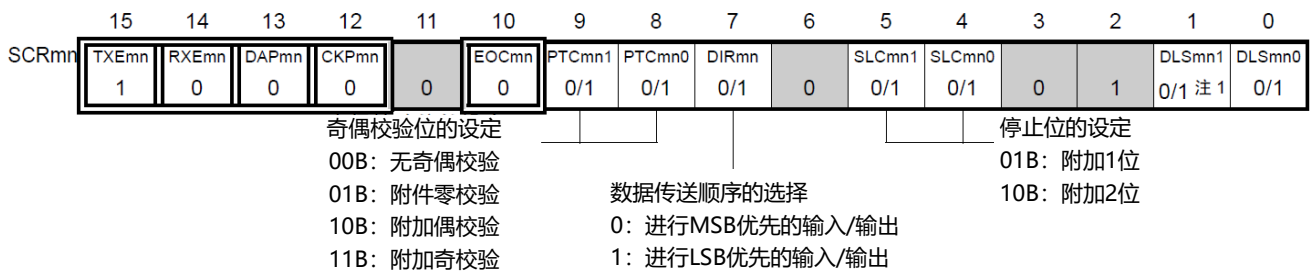
(1) 寄存器的设置

图13-96 UART (UART0~UART2) 的UART发送时的寄存器设置内容例子(1/2)

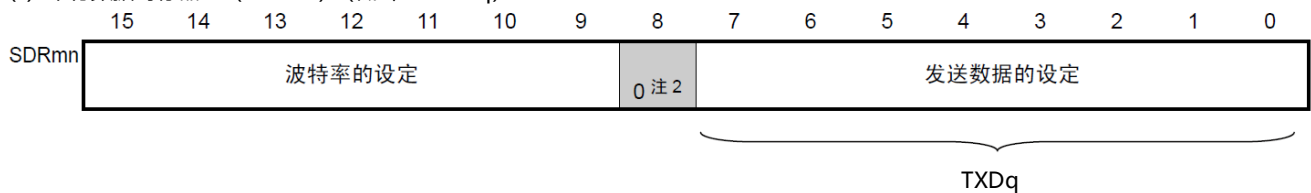
(a) 串行模式寄存器mn(SMRmn)



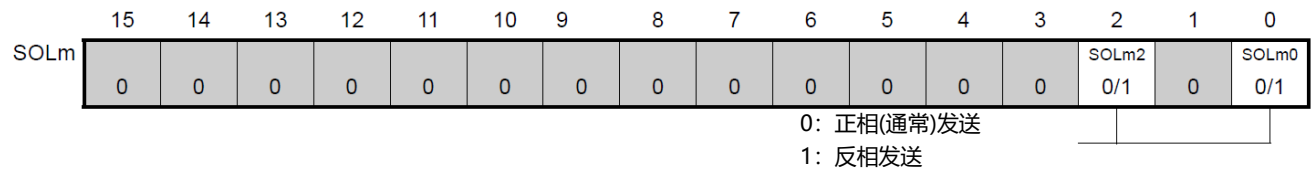
(b) 串行通信运行设定寄存器mn(SCRmn)



(c) 串行数据寄存器mn(SDRmn) (低8位: TXDq)



(d) 串行输出电平寄存器m(SOLm)只设定对象通道的位。



注 1.只限于SCR00寄存器, 其他固定为“1”。

2.当进行9位数据长度的通信时, SDRm0寄存器的bit0~8为发送数据的设置区。只有UART0才能进行9位数据长度的通信。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART号 (q=0~2) mn=00、02、10

☐ : 在UART发送模式中为固定设置。■ : 不能设置(设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

图13-97 UART (UART0~UART2) 的UART发送时的寄存器设置内容例子(2/2)

(e) 串行输出寄存器m(SOm)只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
					×	×	×	×					×	0/1 注	×	0/1 注

0: 串行数据输出值为“0”
1: 串行数据输出值为“1”

(f) 串行输出允许寄存器m(SOEm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
													×	0/1	×	0/1

(g) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													×	0/1	×	0/1

注 在开始发送前，当对应通道的SOLmn位为“0”时，必须置“1”；当对应通道的SOLmn位为“1”时，必须置“0”。
在通信过程中，值因通信数据而变。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART号 (q=0~2) mn=00、02、10

: 在UART发送模式中为固定设置。 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-98 UART发送的初始设置步骤

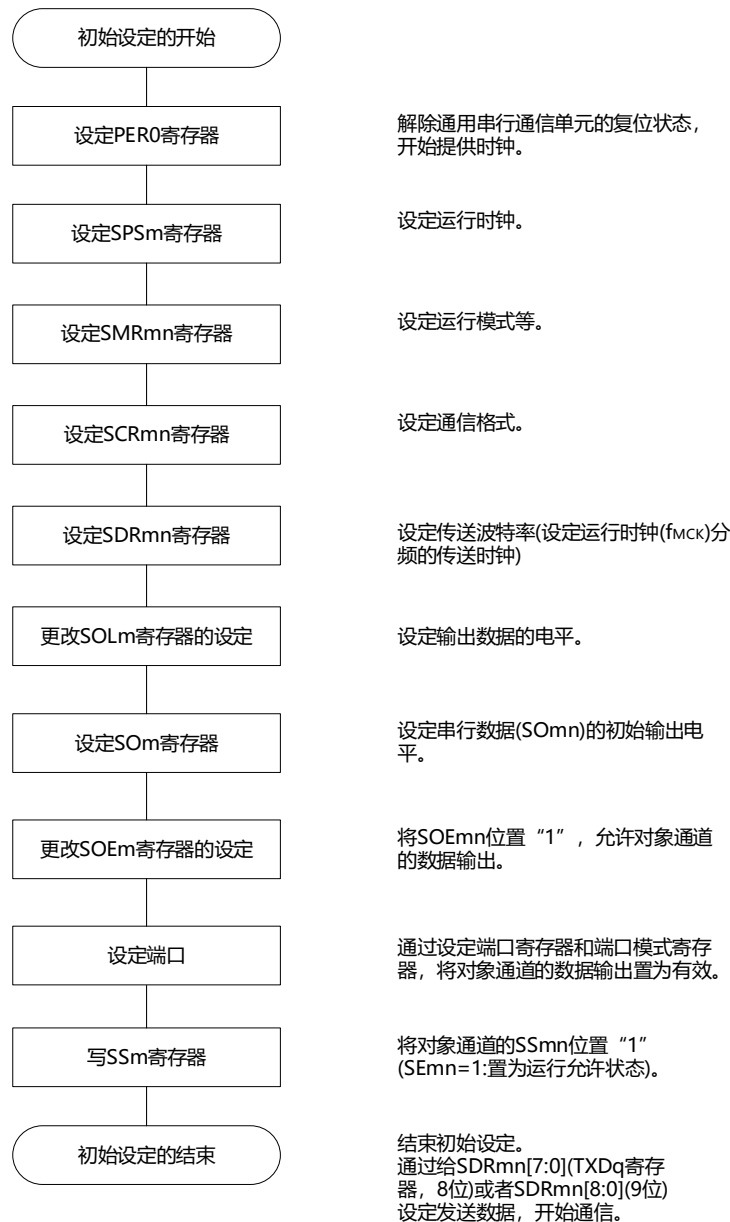


图13-99 UART发送的中止步骤

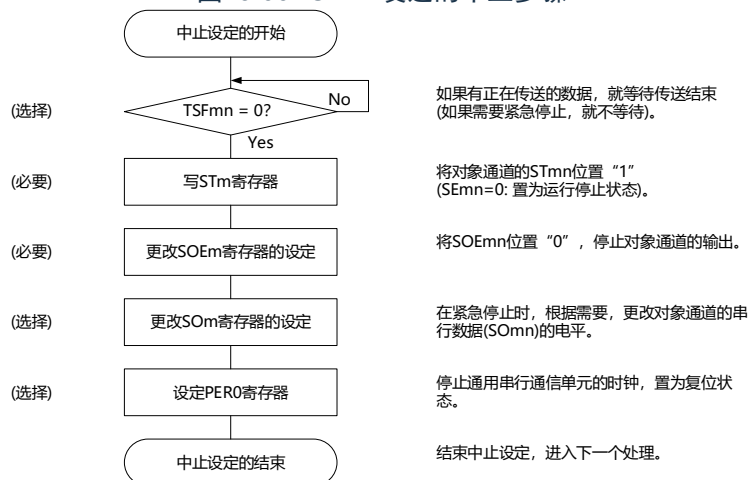


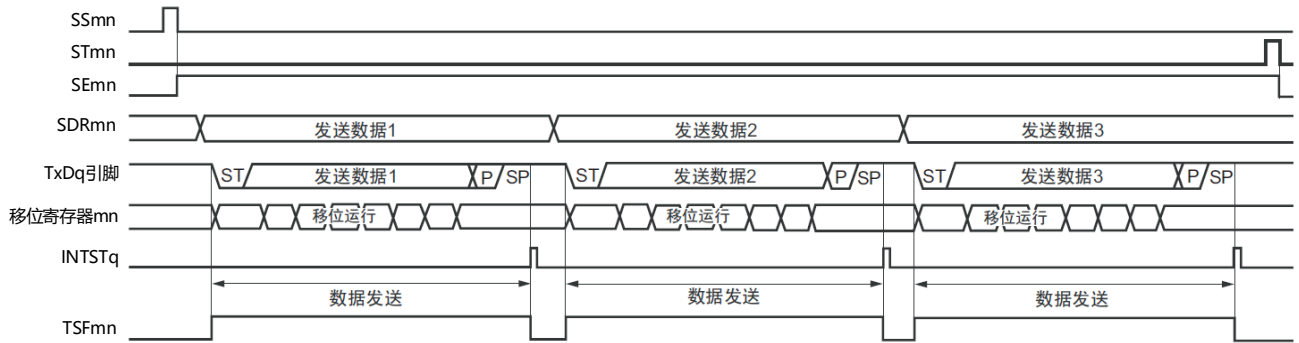
图13-100 重新开始UART发送的设置步骤



备注 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象停止或者通信结束后进行初始设置而不是进行重新开始设置。

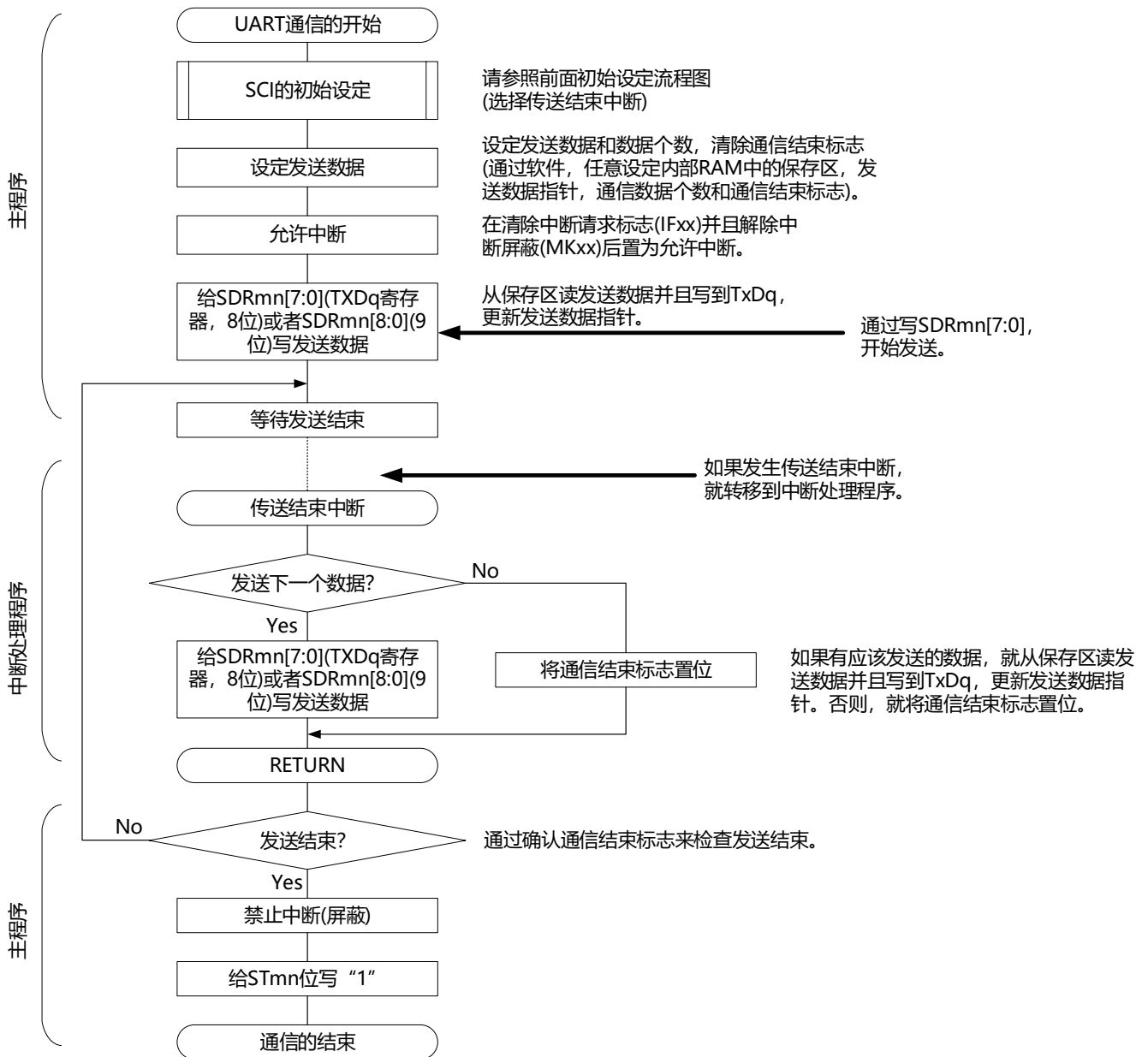
(3) 处理流程（单次发送模式）

图13-101 UART发送（单次发送模式）的时序图



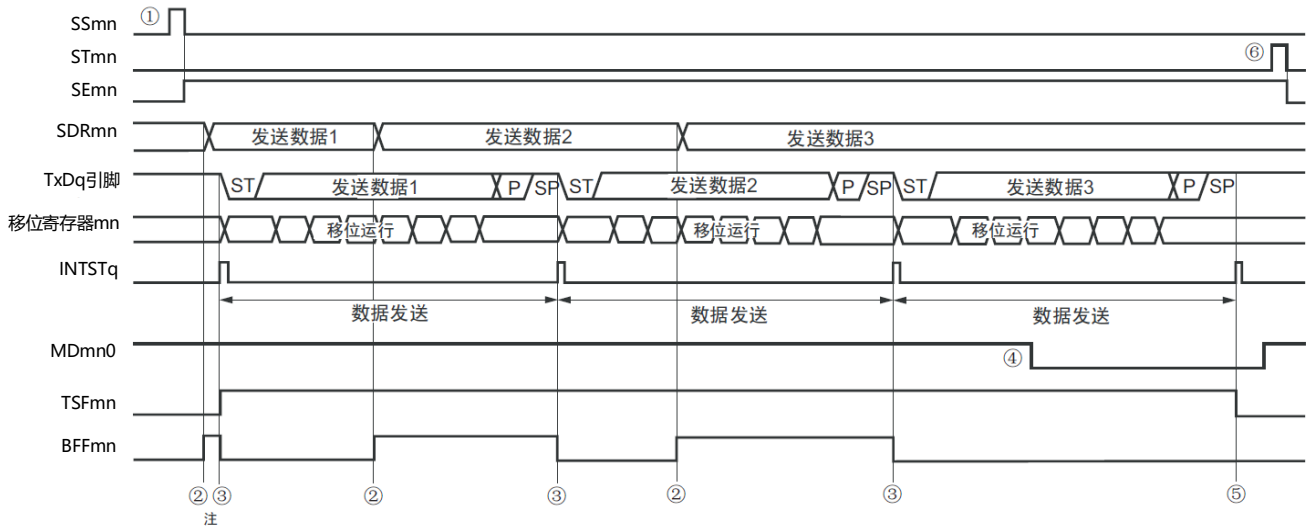
备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART号 (q=0~2) mn=00、02、10

图13-102 UART发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图13-103 UART发送（连续发送模式）的时序图

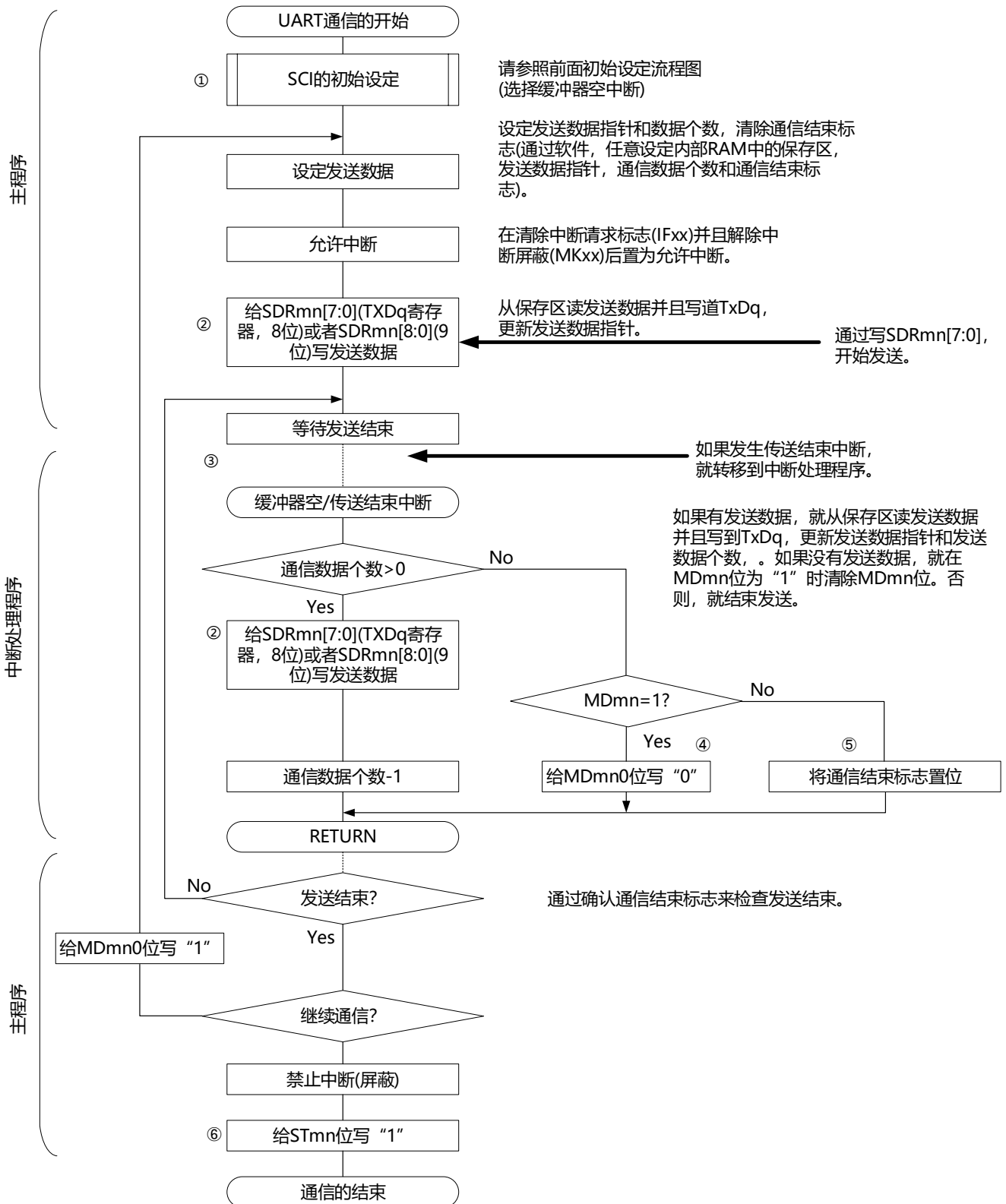


注 如果在串行状态寄存器mn（SSRmn）的BFFmn位为“1”期间（有效数据保存在串行数据寄存器mn（SDRmn））时给SDRmn寄存器写发送数据，就重写发送数据。

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

备注 m：单元号（m=0、1） n：通道号（n=0、2） q：UART号（q=0~2） mn=00、02、10

图13-104 UART发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图13-103 UART发送（连续发送模式）的时序图”中的①~⑥。

13.7.2 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，需要设置奇数通道和偶数通道的SMR寄存器。

UART	UART0	UART1	UART2
对象通道	SCI0的通道1	SCI0的通道3	SCI1的通道1
使用的引脚	RxD0	RxD1	RxD2
中断	INTSR0	INTSR1	INTSR2
	只限于传送结束中断（禁止设置缓冲器空中断）。		
错误中断	INTSRE0	INTSRE1	INTSRE2
错误检测标志	<ul style="list-style-type: none"> • 帧错误检测标志（FEFmn） • 奇偶校验错误检测标志（PEFmn） • 溢出错误检测标志（OVFmn） 		
传送数据长度	7位、8位或者9位 ^{注1}		
传送速率	Max. $f_{MCK}/6$ [bps] ($SDRmn[15:9] \geq 2$)、 Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
数据相位	正相输出 (默认值:)		
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> • 无奇偶校验位（无奇偶校验）。 • 附加零校验（无奇偶校验）。 • 偶校验 • 奇校验 		
停止位	附加1位。		
数据方向	MSB优先或者LSB优先		

注 1. 只有UART0支持9位数据长度。

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

备注1. f_{MCK} : 对象通道的运行时钟频率

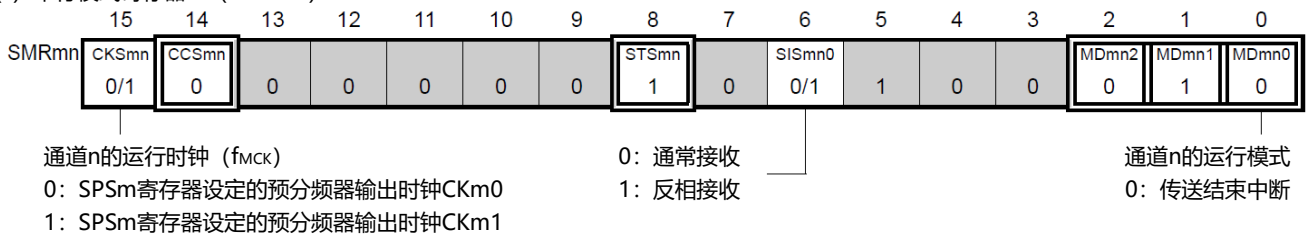
f_{CLK} : 系统时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11

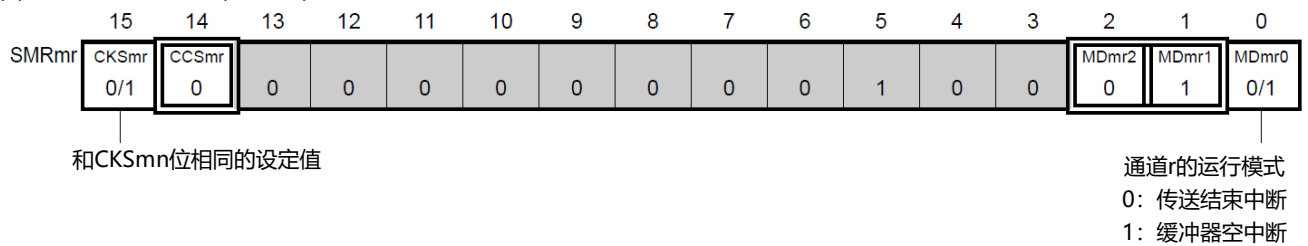
(1) 寄存器的设置

图13-105 UART (UART0~UART2) 的UART接收时的寄存器设置内容例子(1/2)

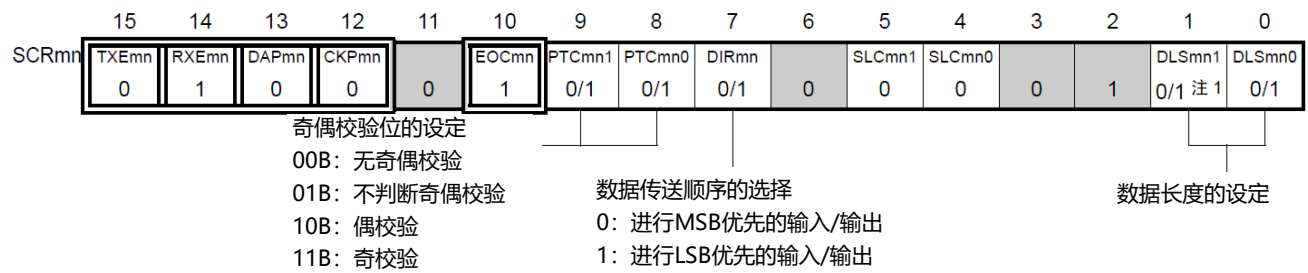
(a) 串行模式寄存器mn(SMRmn)



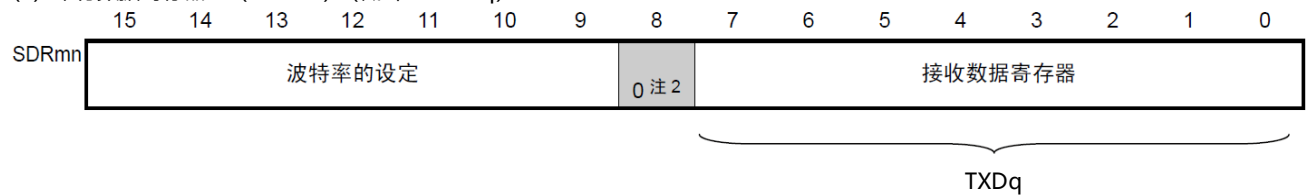
(b) 串行模式寄存器mr(SMRmr)



(c) 串行通信运行设定寄存器mn(SCRmn)



(d) 串行数据寄存器mn(SDRmn) (低8位: TXDq)



注 1.只限于SCR01寄存器, 其他固定为“1”。

2.当进行9位数据长度的通信时, SDRm1寄存器的bit0~8为发送数据的设置区。只有UART0才能进行9位数据长度的通信。

注意 在UART接收时, 还必须设置与通道n成对的通道r的SMRmr寄存器。

备注1.m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11

r: 通道号 (r=n-1) q: UART号 (q=0~2)

2 : 在UART接收模式中为固定设置。 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

图13-105 UART (UART0~UART2) 的UART接收时的寄存器设置内容例子(2/2)

(e) 串行输出寄存器m(SOm)在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
	0	0	0	0	×	×	×	×	0	0	0	0	×	×	×	×

(f) 串行输出允许寄存器m(SOEm)在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	×

(g) 串行通道开始寄存器m(SSm)只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	×

备注1.m: 单元号 (m=0、1)

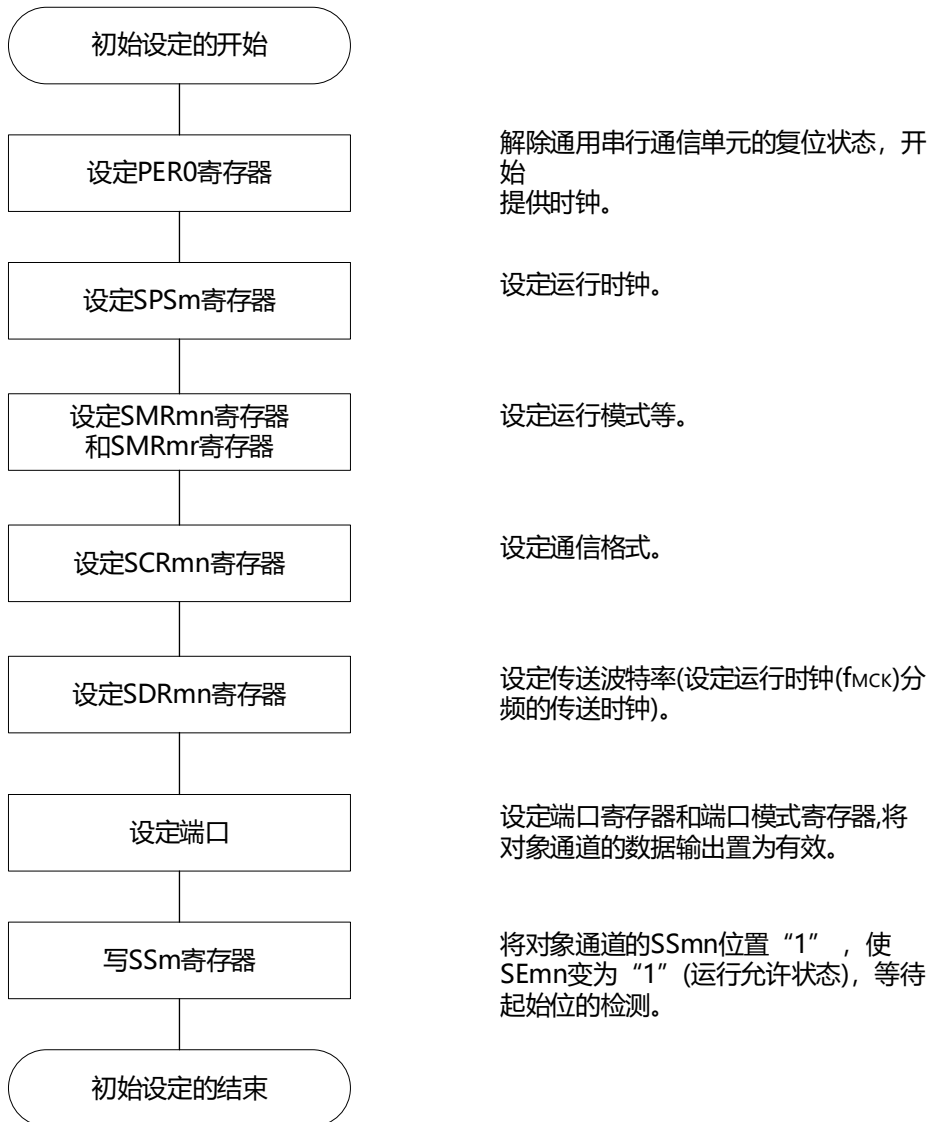
2 : 在UART接收模式中为固定设置。 : 不能设置 (设置初始值)。

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图13-106 UART接收的初始设置步骤



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

图13-107 UART接收的中止步骤

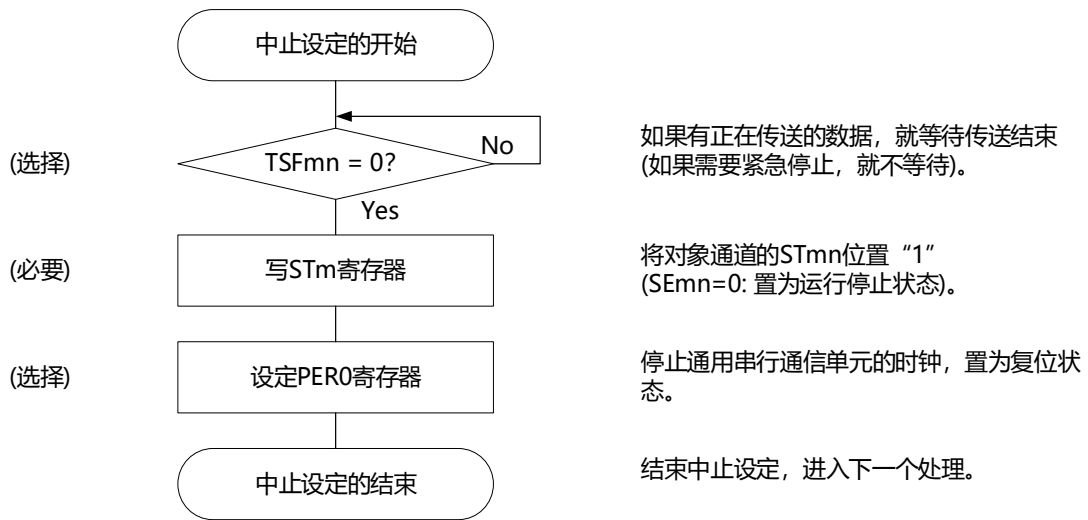


图13-108 重新开始UART接收的设置步骤

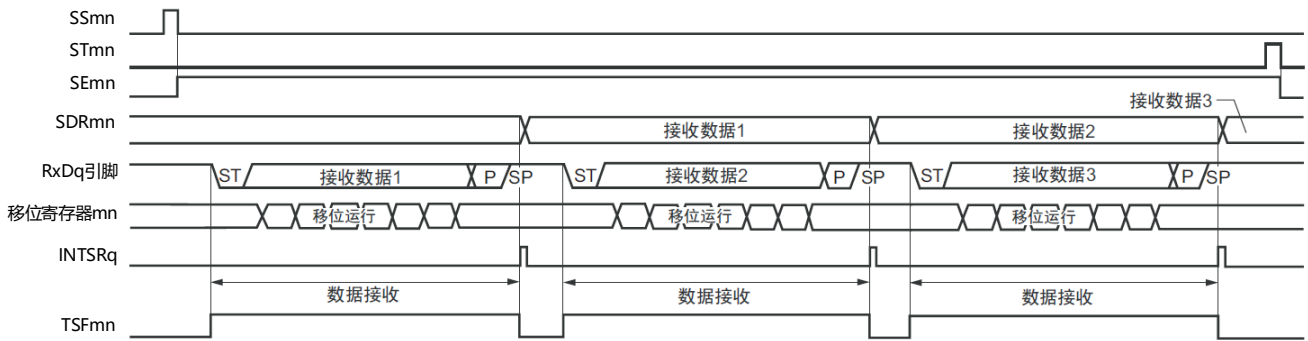


注意 必须在将SCRmn寄存器的RXEmn位置“1”后至少间隔4个 f_{MCK} 时钟, 然后将SSmn位置“1”。

备注 如果在中止设置中改写PER0来停止提供时钟, 就必须在等到通信对象停止或者通信结束后进行初始设置而不是进行重新开始设置。

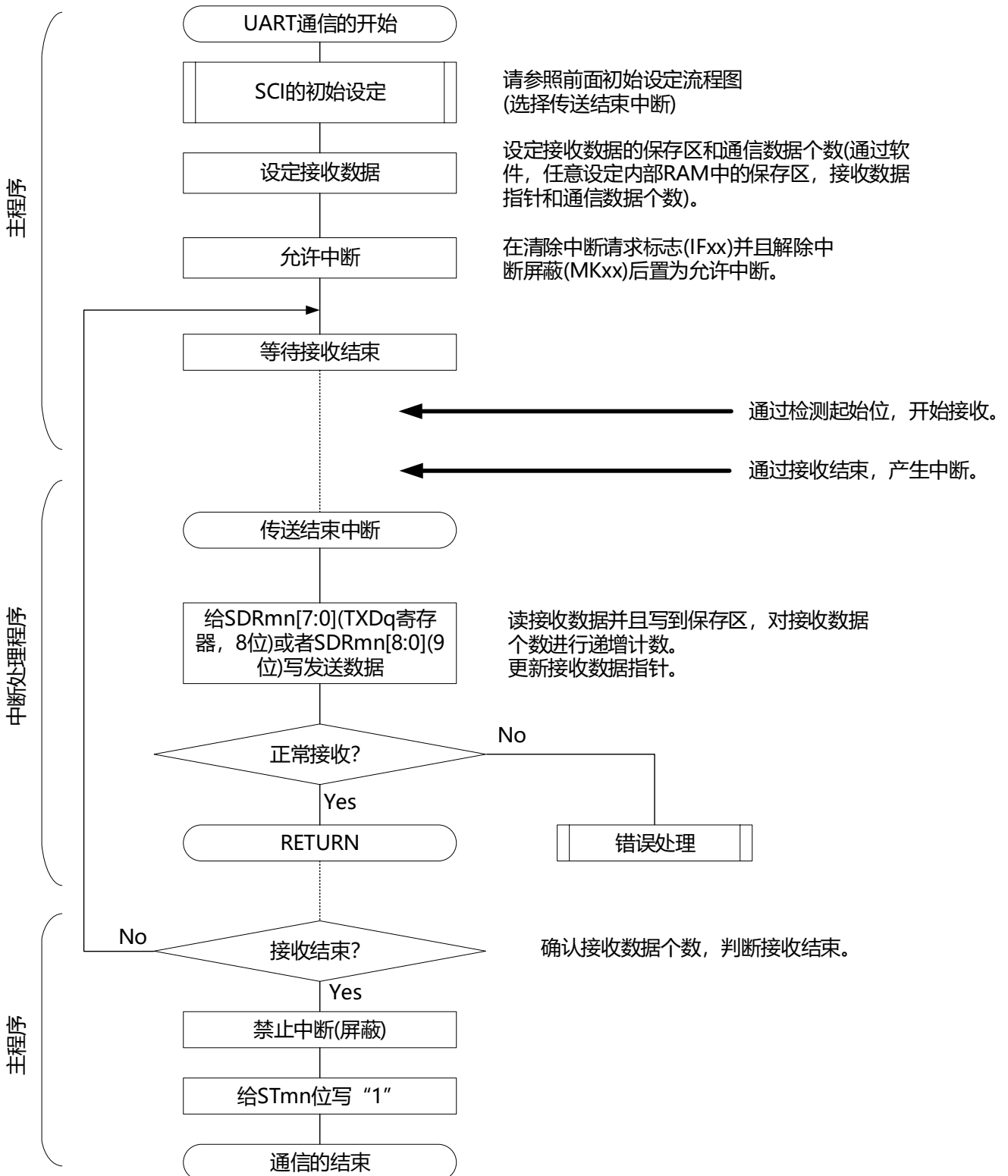
(3) 处理流程

图13-109 UART接收的时序图



备注 m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11
 r: 通道号 (r=n-1) q: UART号 (q=0~2)

图13-110 UART接收的流程图



13.7.3 波特率的计算

(1) 波特率的计算式

UART (UART0~UART2) 通信的波特率能用以下计算式进行计算:

$$\text{(波特率)} = \{\text{对象通道的运行时钟 (f}_{\text{MCK}}\text{) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

注意 禁止将串行数据寄存器mn (SDRmn) 的SDRmn[15:9]设置为“0000000B”和“0000001B”。

备注1.因为在使用UART时SDRmn[15:9]的值为SDRmn寄存器的bit15~9的值 (0000010B~1111111B), 所以为2~127。

2.m: 单元号 (m=0、1) n: 通道号 (n=0~2) mn=00~03、10~11

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器m (SPSm) 和串行模式寄存器mn (SMRmn) 的bit15 (CKSmn位)。

表13-4 UART运行时钟的选择

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

注 要更改被选择为 f_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器m（STm）=000FH）后进行更改。

备注1.X: 忽略

2.m: 单元号 (m=0、1) n: 通道号 (n=0~2) mn=00~03、10~11

(2) 发送时的波特率误差

UART (UART0~UART2) 通信发送时的波特率误差能用以下计算式进行计算, 必须将发送方的波特率设置在接收方的波特率容许范围内。

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

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

UART波特率 (目标波特率)	$f_{\text{CLK}}=32\text{MHz}$			
	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	$f_{\text{CLK}}/2^9$	103	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	63	31250.0bps	±0.0%
38400bps	$f_{\text{CLK}}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	103	76923.1bps	+0.16%
153600bps	f_{CLK}	103	153846bps	+0.16%
312500bps	f_{CLK}	50	313725bps	±0.39%

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) mn=00、02、10

(3) 接收时的波特率容许范围

UART (UART0~UART2) 通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设置在接收方的波特率容许范围内。

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

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

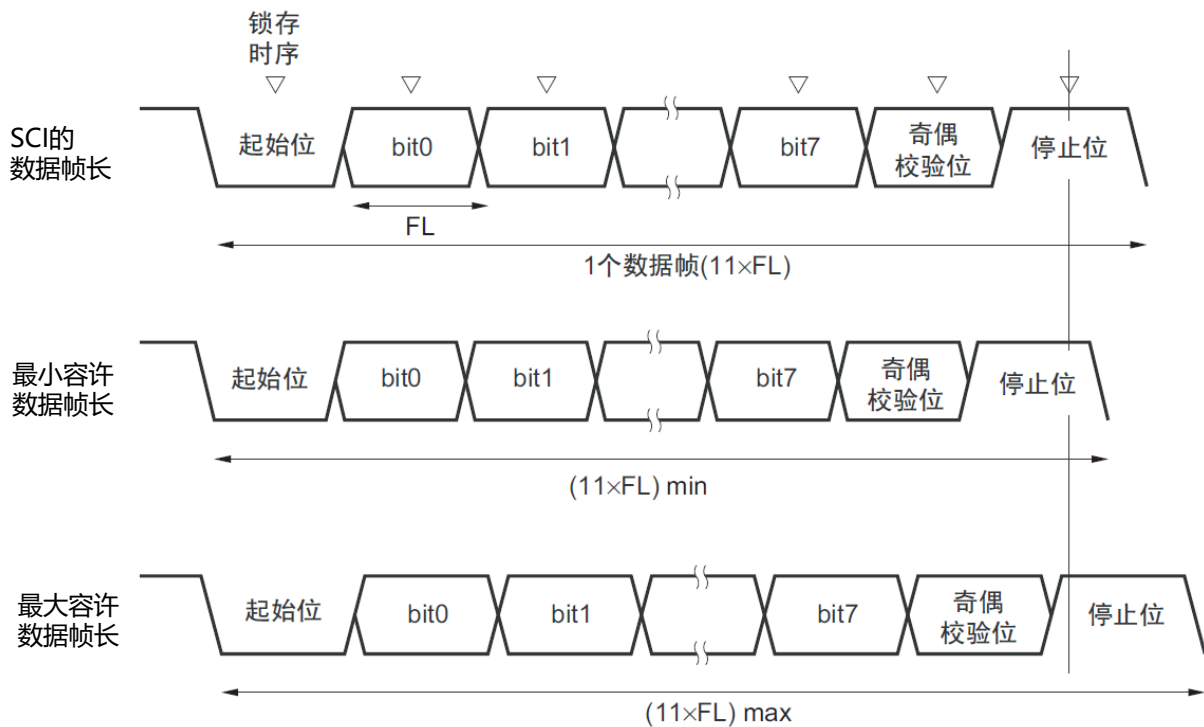
Brate: 接收方的波特率的计算值 (参照“13.7.4 (1)波特率的计算式”)

k : SDRmn[15:9]+1

Nfr : 1个数据帧的帧长[位]
= (起始位) + (数据长度) + (奇偶校验位) + (停止位)

备注 m: 单元号 (m=0、1n: 通道号 (n=1、3) mn=01、03、11

图13-111 接收时的波特率容许范围 (1个数据帧=11位的情况)



如上图所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器mn (SDRmn) 的bit15~9设置的分频比。如果最后的数据 (停止位) 能赶上此锁存时序，就能正常接收。

13.7.4 在UART (UART0~UART2) 通信过程中发生错误时的处理步骤

在UART (UART0~UART2) 通信过程中发生错误时的处理步骤如图13-112和图13-113所示。

图13-112 发生奇偶校验错误或者溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误种类, 读取值用于清除错误标志。
给串行标志清除触发寄存器mn (SDIRmn) 写“1”。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器, 只能清除读操作时的错误。

图13-113 发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn (SDRmn)。	→ SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn (SSRmn)。		判断错误种类, 读取值用于清除错误标志。
写串行标志清除触发寄存器mn (SIRmn)。	→ 清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器, 只能清除读操作时的错误。
将串行通道停止寄存器m (STm) 的STmn位置“1”。	→ 串行通道允许状态寄存器m (SEm) 的SEmn位为“0”并且通道n为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移, 所以可认为发生了帧错误。因此, 需要与通信方重新取得同步, 重新开始通信。
将串行通道开始寄存m (SSm) 的SSmn位置“1”。	→ 串行通道允许状态寄存器m (SEm) 的SEmn位为“1”并且通道n为可运行状态。	

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00~03、10~11

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) I²C总线模式（支持多主控）

此模式通过串行时钟（SCLAn）和串行数据总线（SDAAn）的2条线，与多个设备进行8位数据传送。符合I²C总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的I²C总线控制部分。

因为串行接口IICA的SCLAn引脚和SDAAn引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

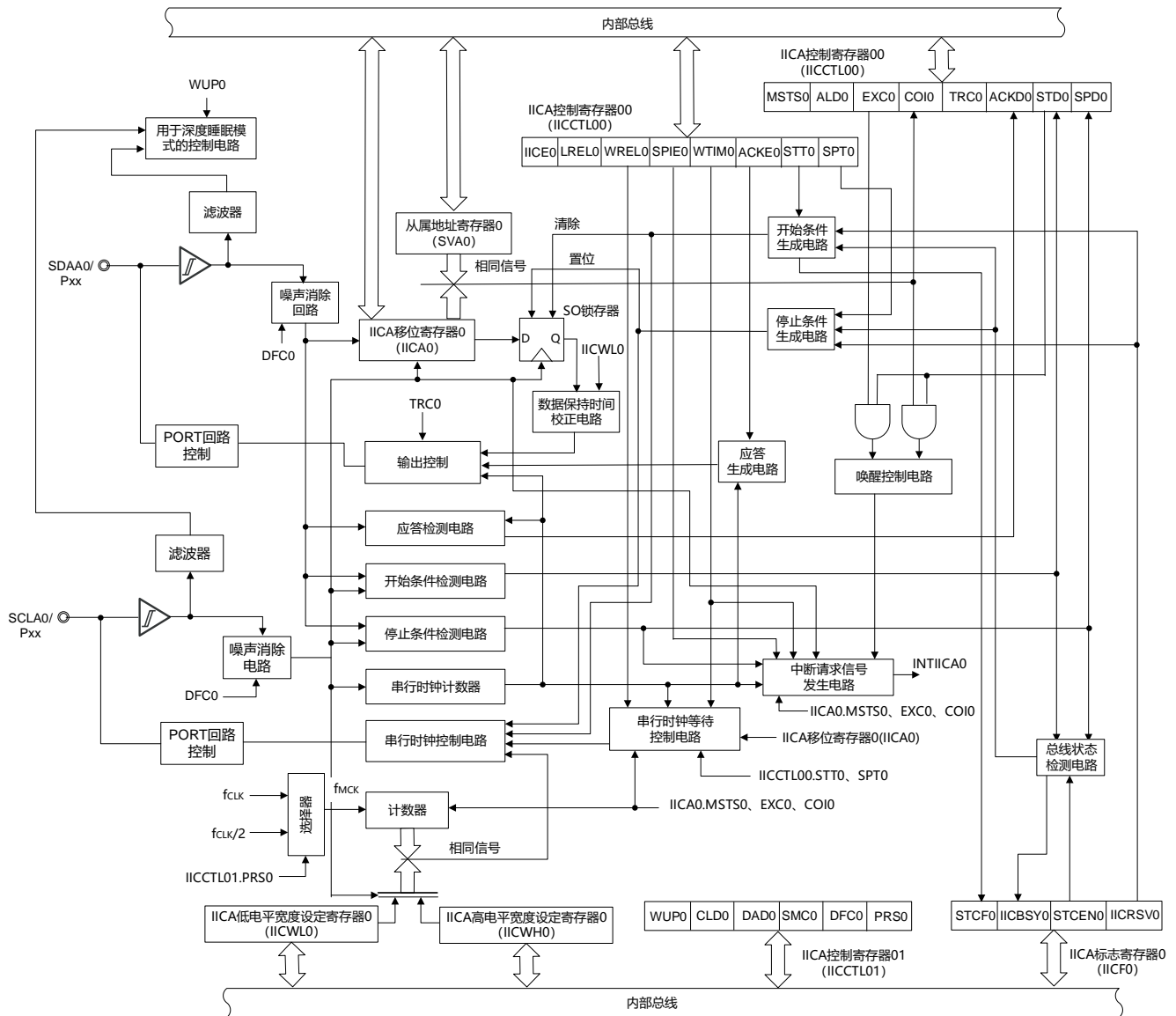
(3) 唤醒模式

在深度睡眠模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除深度睡眠模式。通过IICA控制寄存器n1（IICCTLn1）的WUPn位进行设置。

串行接口IICA的框图如图14-1所示。

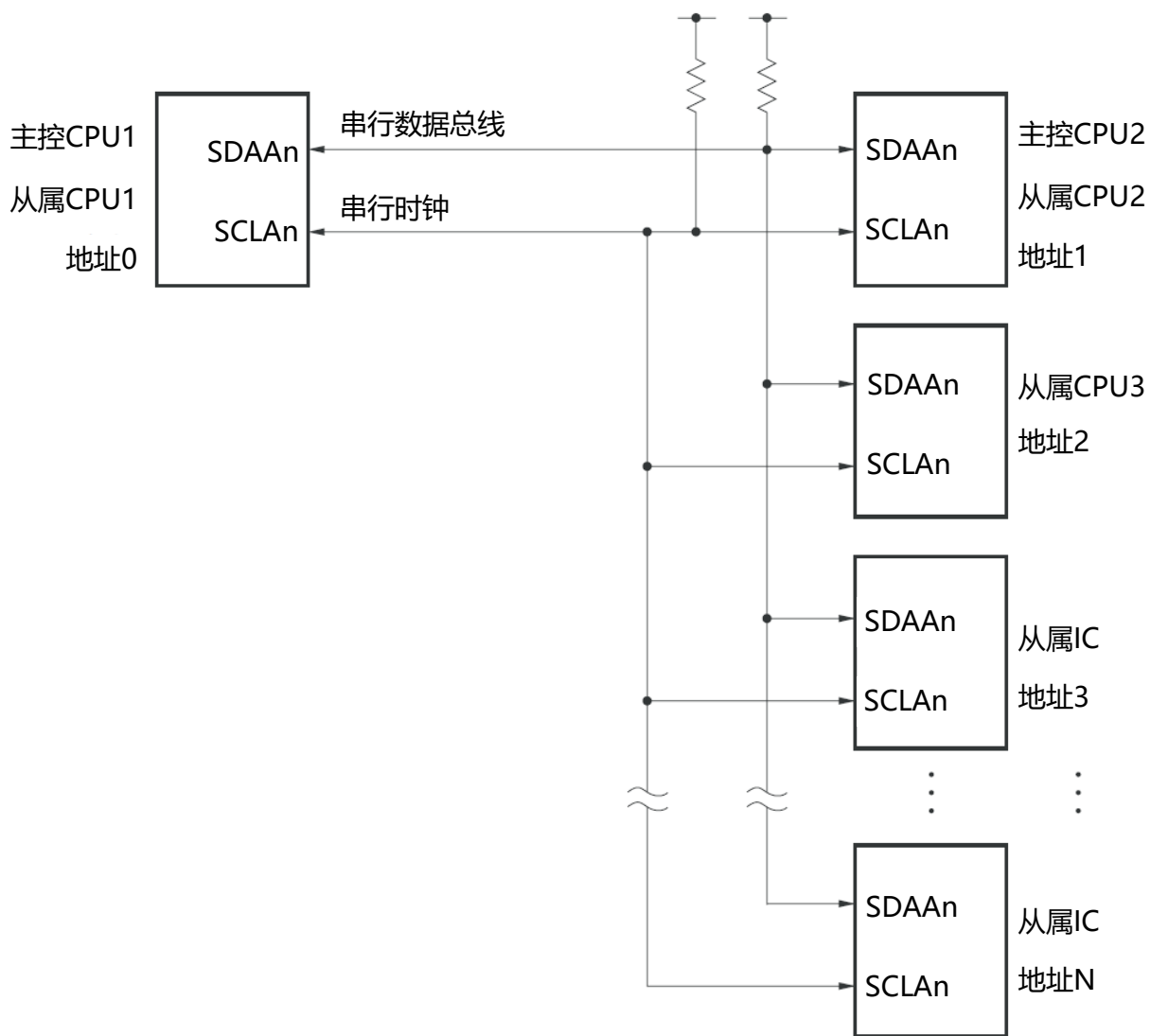
备注 n=0

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 I²C总线的串行总线结构例子



备注 n=0

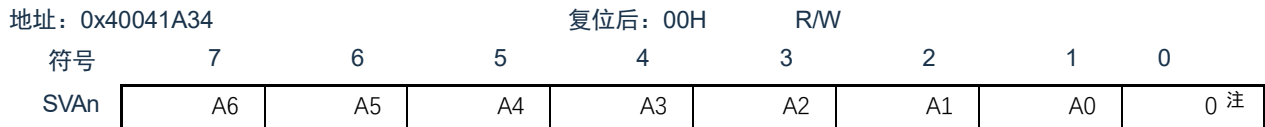
(2) 从属地址寄存器n (SVAn)

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置SVAn寄存器。但是，在STDn位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图14-4 从属地址寄存器n (SVAn) 的格式



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

当设置在从属地址寄存器n (SVAn) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIICAn)。

(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I²C中断请求。

- 第8个或者第9个串行时钟的下降（通过WTIMn位进行设置）
- 因检测到停止条件而产生中断请求（通过SPIEn位进行设置）。

备注 WTIMn位：IICA控制寄存器n0 (IICCTLn0) 的bit3
 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位设置总线状态检测电路的初始状态。

备注1.STTn位	: IICA控制寄存器n0（IICCTLn0）的bit1
SPTn位	: IICA控制寄存器n0（IICCTLn0）的bit0
IICRSVn位	: IICA标志寄存器n（IICFn）的bit0
IICBSYn位	: IICA标志寄存器n（IICFn）的bit6
STCFn位	: IICA标志寄存器n（IICFn）的bit7
STCENn位	: IICA标志寄存器n（IICFn）的bit1

2.n=0

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

备注 n=0

14.3.1 外围允许寄存器0 (PER0)

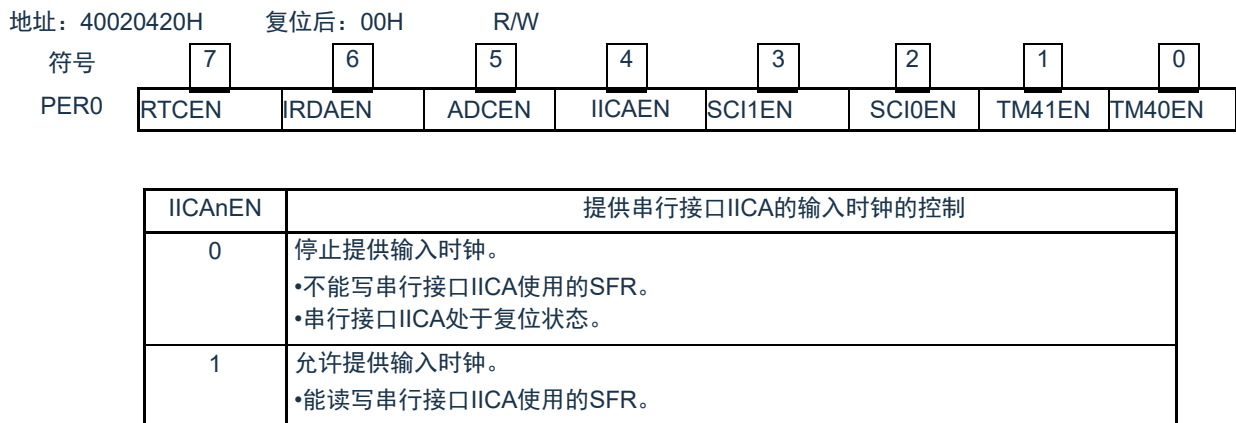
PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用串行接口IICAn时，必须将bit4 (IICAEN) 置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图14-5 外围允许寄存器0 (PER0) 的格式



注意1. 要设置串行接口IICA时，必须先IICAEN位为“1”的状态下设置以下的寄存器。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)。

- IICA控制寄存器n0 (IICCTLn0)
- IICA标志寄存器n (IICFn)
- IICA状态寄存器n (IICSn)
- IICA控制寄存器n1 (IICCTLn1)
- IICA低电平宽度设置寄存器n (IICWLn)
- IICA高电平宽度设置寄存器n (IICWHn)

备注 n=0

14.3.2 IICA控制寄存器n0 (IICCTLn0)

这是允许或者停止I²C运行、设置等待时序以及设置其他I²C运行的寄存器。

通过8位存储器操作指令设置IICCTLn0寄存器。但是，必须在IICEn位为“0”时或者在等待期间设置SPIEn位、WTIMn位和ACKEn位，而且在将IICEn位从“0”置为“1”时能同时设置这些位。

在产生复位信号后，此寄存器的值变为“00H”。

备注 n=0

图14-6 IICA控制寄存器n0 (IICCTLn0) 的格式(1/4)

地址: 0x40041A30

复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I ² C运行的允许
0	停止运行。对IICA状态寄存器n (IICSn) 进行复位 ^{注1} , 并且停止内部运行。
1	允许运行。
必须在SCLAn线和SDAAn线为高电平的状态下将此位置“1”。	
清除条件 (IICEn=0)	置位条件 (IICEn=1)
•通过指令清除。 •当复位时	•通过指令置位。

LRELn ^{注2, 3}	通信的退出
0	通常运行
1	退出当前的通信, 进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况下使用。 SCLAn线和SDAAn线变为高阻状态。 IICA控制寄存器n0 (IICCTLn0) 和IICA状态寄存器n (IICSn) 中的以下标志被清“0”: •STTn•SPTn•MSTSn•EXCn•COLn•TRCn•ACKDn•STDn
变为退出通信的待机状态, 保持到满足以下的通信参加条件为止。 •在检测到停止条件后作为主控设备启动。 •在检测到开始条件后地址匹配或者接收到扩展码。	
清除条件 (LRELn=0)	置位条件 (LRELn=1)
•在执行后自动清除。 •当复位时	•通过指令置位。

WRELn ^{注2, 3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRCn=1) 的第9个时钟等待期间将WRELn位 (解除等待) 置位, SDAAn线就变为高阻抗状态 (TRCn=0)。	
清除条件 (WRELn=0)	置位条件 (WRELn=1)
•在执行后自动清除。 •当复位时	•通过指令置位。

- 注 1.对IICA移位寄存器n (IICAn)、IICA标志寄存器n (IICFn) 的STCFn位和IICBSYn位以及IICA控制寄存器n1 (IICCTLn1) 的CLDn位和DADn位进行复位。
- 2.在IICEn位为“0”的状态下, 此位的信号无效。
- 3.LRELn位和WRELn位的读取值总是“0”。

注意 如果在SCLAn线为高电平、SDAAn线为低电平并且数字滤波器为ON (IICCTLn1寄存器的DFCn=1) 时允许I²C运行 (IICEn=1), 就立即检测开始条件。此时, 必须在允许I²C运行 (IICEn=1) 后连续通过位存储器操作指令将LRELn位置“1”。

备注 n=0

图14-6 IICA控制寄存器n0 (IICCTLn0) 的格式(2/4)

SPIEn ^{注1}	允许或者禁止停止条件检测产生的中断请求	
0	禁止	
1	允许	
当IICA控制寄存器n1 (IICCTLn1) 的WUPn位为“1”时, 即使将SPIEn位置“1”也不产生停止条件中断。		
清除条件 (SPIEn=0)		置位条件 (SPIEn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

WTIMn ^{注1}	等待和中断请求的控制	
0	在第8个时钟的下降沿产生中断请求信号。 主控设备: 在输出8个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入8个时钟后, 将时钟置为低电平, 然后等待主控设备。	
1	在第9个时钟的下降沿产生中断请求信号。 主控设备: 在输出9个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入9个时钟后, 将时钟置为低电平, 然后等待主控设备。	
在地址传送期间, 与此位的设置无关, 在第9个时钟的下降沿产生中断; 在地址传送结束后, 此位的设置有效。主控设备在地址传送期间的第9个时钟下降沿进入等待状态。接收到本地站地址的从属设备在产生应答 (ACK) 后的第9个时钟下降沿进入等待状态, 但是接收到扩展码的从属设备在第8个时钟下降沿进入等待状态。		
清除条件 (WTIMn=0)		置位条件 (WTIMn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

ACKEn ^{注1、2}	应答控制	
0	禁止应答。	
1	允许应答。在第9个时钟期间将SDAAn线置为低电平。	
清除条件 (ACKEn=0)		置位条件 (ACKEn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

- 注 1. 在IICEn位为“0”的状态下, 此位的信号无效。必须在此期间设置此位。
 2. 在地址传送过程中不是扩展码时, 设置值无效。当为从属设备并且地址匹配时, 与设置值无关而生成应答。

备注 n=0

图14-6 IICA控制寄存器n0 (IICCTLn0) 的格式(3/4)

STTn ^{注1、2}	开始条件的触发
0	不生成开始条件。
1	当总线被释放时（待机状态，IICBSYn位为“0”）：如果将此位置“1”，就生成开始条件（作为主控设备的启动）。当第三方正在通信时： <ul style="list-style-type: none"> •允许通信预约功能的情况（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 •禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除STTn位并且将STTn清除标志（STCFn）置“1”，不生成开始条件。等待状态（主控设备）： 在解除等待后生成重新开始条件。
有关置位时序的注意事项： <ul style="list-style-type: none"> •主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 •主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。 •禁止与停止条件的触发（SPTn）同时置“1”。 •在将STTn位置“1”后，禁止在满足清除条件前再次将此位“1”。 	
清除条件（STTn=0）	置位条件（STTn=1）
<ul style="list-style-type: none"> •在禁止通信预约的状态下将STTn位置“1”。 •在仲裁失败时 •主控设备生成开始条件。 •因LRELn位为“1”（退出通信）而进行的清除 •当IICEn位为“0”（停止运行）时 •当复位时 	<ul style="list-style-type: none"> •通过指令置位。

注 1.在IICEn位为“0”的状态下，此位的信号无效。
2.STTn位的读取值总是“0”。

备注1.如果在设置数据后读bit1（STTn），此位就变为“0”。
2.IICRSVn : IICA标志寄存器n（IICFn）的bit0
STCFn : IICA标志寄存器n（IICFn）的bit7
3.n=0

图14-6 IICA控制寄存器n0 (IICCTLn0) 的格式(4/4)

SPTn ^注	停止条件的触发	
0	不生成停止条件。	
1	生成停止条件（作为主控设备的传送结束）。	
有关置位时序的注意事项： <ul style="list-style-type: none"> •主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 •主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第9个时钟后的等待期间将此位置“1”。 •禁止与开始条件的触发（STTn）同时置“1”。 •只有在主控设备的情况下才能将SPTn位置“1”。 •在WTIMn位为“0”时，必须注意：如果在输出8个时钟后的等待期间将SPTn位置“1”，就在解除等待后的第9个时钟的高电平期间生成停止条件。必须在输出8个时钟后的等待期间将WTIMn位从“0”置为“1”并且在输出第9个时钟后的等待期间将SPTn位置“1”。 •在将SPTn位置“1”后，禁止在满足清除条件前再次将此位置“1”。 		
清除条件（SPTn=0）		置位条件（SPTn=1）
<ul style="list-style-type: none"> •当仲裁失败时 •在检测到停止条件后自动清除。 •因LRELn位为“1”（退出通信）而进行的清除 •当IICEn位为“0”（停止运行）时 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

注 SPTn位的读取值总是“0”。

注意 在IICA状态寄存器n (IICSn) 的bit3 (TRCn) 为“1”（发送状态）时，如果在第9个时钟将IICCTLn0寄存器的bit5 (WRELn) 置“1”来解除等待，就在清除TRCn位（接收状态）后将SDAAn线置为高阻抗。必须通过写IICA移位寄存器n进行TRCn位为“1”（发送状态）时的等待解除。

备注 n=0

图14-7 IICA状态寄存器n (IICSn) 的格式(2/3)

EXCn	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXCn=0)		置位条件 (EXCn=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时 		<ul style="list-style-type: none"> 当接收的地址数据的高4位为“0000”或者“1111”时（在第8个时钟的上升沿置位）

COIn	地址匹配的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COIn=0)		置位条件 (COIn=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 当复位时 		<ul style="list-style-type: none"> 当接收地址和本地站地址（从属地址寄存器n (SVAn)）相同时（在第8个时钟的上升沿置位）

TRCn	发送/接收的状态检测	
0	处于接收状态（发送状态除外）。将SDAAn线置为高阻抗。	
1	处于发送状态。设置为能将SOn锁存器的值输出到SDAAn线（在第1字节的第9个时钟的下降沿以后有效）。	
清除条件 (TRCn=0)		置位条件 (TRCn=1)
<主控设备和从属设备> <ul style="list-style-type: none"> 当检测到停止条件时 因LRELn位为“1”（退出通信）而进行的清除 当IICEn位从“1”变为“0”（停止运行）时 因WRELn位为“1”（解除等待）而进行的清除注 当ALDn位从“0”变为“1”（仲裁失败）时 当复位时 不参加通信的情况 (MSTSn、EXCn、COIn=0) <主控设备> <ul style="list-style-type: none"> 当第1字节的LSB（传送方向指示位）输出“1”时 <从属设备> <ul style="list-style-type: none"> 当检测到开始条件时 当第1字节的LSB（传送方向指示位）输入“0”时 		<主控设备> <ul style="list-style-type: none"> 当生成开始条件时 当第1字节（地址传送）的LSB（传送方向指示位）输出“0”（主控发送）时 <从属设备> <ul style="list-style-type: none"> 当主控设备的第1字节（地址传送）的LSB（传送方向指示位）输入“1”（从属发送）时

注 在IICA状态寄存器n (IICSn) 的bit3 (TRCn) 为“1”（发送状态）时，如果在第9个时钟将IICA控制寄存器n0 (IICCTLn0) 的bit5 (WRELn) 置“1”来解除等待，就在清除TRCn位（接收状态）后将SDAAn线置为高阻抗。必须通过写IICA移位寄存器n进行TRCn位为“1”（发送状态）时的等待解除。

备注1.LRELn: IICA控制寄存器n0 (IICCTLn0) 的bit6

IICEn: IICA控制寄存器n0 (IICCTLn0) 的bit7

2.n=0

图14-7 IICA状态寄存器n (IICSn) 的格式(3/3)

ACKDn	应答 (ACK) 的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件 (ACKDn=0)		置位条件 (ACKDn=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当下一个字节的第1个时钟上升时 因LRELn位为“1” (退出通信) 而进行的清除 当IICEn位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时

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)
<ul style="list-style-type: none"> 当检测到停止条件时 在地址传送后的下一个字节的第1个时钟上升时 因LRELn位为“1” (退出通信) 而进行的清除 当IICEn位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到开始条件时

SPDn	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件, 主控设备结束通信并且已释放总线。	
清除条件 (SPDn=0)		置位条件 (SPDn=1)
<ul style="list-style-type: none"> 在将此位置位后, 在检测到开始条件后的地址传送字节的第1个时钟上升时 当WUPn位从“1”变为“0”时 当IICEn位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到停止条件时

备注1.LRELn : IICA控制寄存器n0 (IICCTLn0) 的bit6
 IICEn : IICA控制寄存器n0 (IICCTLn0) 的bit7
 2.n=0

14.3.4 IICA标志寄存器n (IICFn)

这是设置I²C运行模式以及表示I²C总线状态的寄存器。

通过8位存储器操作指令设置IICFn寄存器。但是，只能读STTn清除标志 (STCFn) 和I²C总线状态标志 (IICBSYn)。

通过IICRSVn位设置允许或者禁止通信预约功能，并且通过STCENn位设置IICBSYn位的初始值。只有在禁止I²C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时才能写IICRSVn位和STCENn位。在允许运行后，只能读IICFn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图14-8 IICA标志寄存器n (IICFn) 的格式

地址: 0x40041B52

复位后: 00H

R/W注

符号

IICFn

STCFn	STTn清除标志	
0	发行开始条件。	
1	无法发行开始条件而清除STTn标志。	
清除条件 (STCFn=0)		置位条件 (STCFn=1)
<ul style="list-style-type: none"> •因STTn位为“1”而进行的清除 •当IICEn位为“0” (停止运行) 时 •当复位时 		<ul style="list-style-type: none"> •在设置为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将STTn位清“0”时

IICBSYn	I ² C总线状态标志	
0	总线释放状态 (STCENn=1时的通信初始状态)	
1	总线通信状态 (STCENn=0时的通信初始状态)	
清除条件 (IICBSYn=0)		置位条件 (IICBSYn=1)
<ul style="list-style-type: none"> •当检测到停止条件时 •当IICEn位为“0” (停止运行) 时 •当复位时 		<ul style="list-style-type: none"> •当检测到开始条件时 •STCENn位为“0”时的IICEn位的置位

STCENn	初始开始允许触发	
0	在允许运行 (IICEn=1) 后，通过检测停止条件来允许生成开始条件。	
1	在允许运行 (IICEn=1) 后，不检测停止条件而允许生成开始条件。	
清除条件 (STCENn=0)		置位条件 (STCENn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当检测到开始条件时 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

IICRSVn	通信预约功能禁止位	
0	允许通信预约。	
1	禁止通信预约。	
清除条件 (IICRSVn=0)		置位条件 (IICRSVn=1)
<ul style="list-style-type: none"> •通过指令清除。 •当复位时 		<ul style="list-style-type: none"> •通过指令置位。

注 bit6和bit7是只读位。

注意1.只有在停止运行 (IICEn=0) 时才能写STCENn位。

2.如果STCENn位为“1”，就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0)，因此为了避免在发行第

1. 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)

这是用于设置I²C运行模式以及检测SCLAn引脚和SDAAn引脚状态的寄存器。

通过8位存储器操作指令设置IICCTLn1寄存器。但是，只能读CLDn位和DADn位。

除了WUPn位以外，必须在禁止I²C运行（IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设置IICCTLn1寄存器。

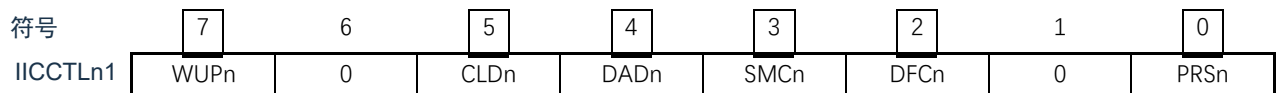
在产生复位信号后，此寄存器的值变为“00H”。

图14-9 IICA控制寄存器n1 (IICCTLn1) 的格式(1/2)

地址: 0x40041A31

复位后: 00H

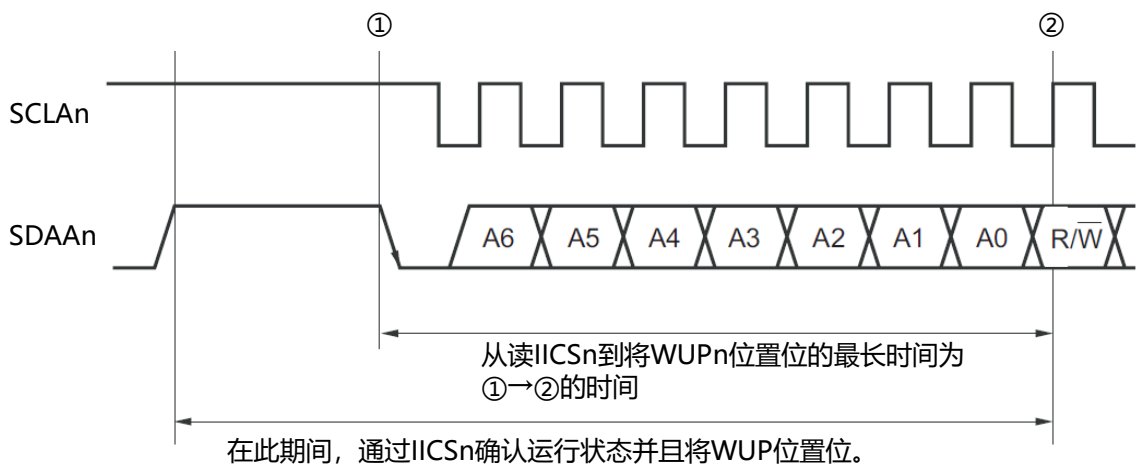
R/W^{注1}



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 _{MCK} 时钟，然后执行深度睡眠指令（参照“图14-28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)	
•通过指令清除（在地址匹配或者接收到扩展码后）。	
置位条件 (WUPn=1)	
•通过指令置位（MSTS _n =0、EXC _n =0、COIn=0并且STD _n =0（不参加通信） ^{注2} 。	

注 1.bit4和bit5是只读位。

2.在以下所示的期间，需要确认IICA状态寄存器n (IICS_n) 的状态并且将其置位。



备注 n=0

图14-9 IICA控制寄存器n1 (IICCTLn1) 的格式(2/2)

CLDn	SCLAn引脚的电平检测（只在IICEn位为“1”时有效）	
0	检测到SCLAn引脚为低电平。	
1	检测到SCLAn引脚为高电平。	
清除条件（CLDn=0）		置位条件（CLDn=1）
<ul style="list-style-type: none"> •当SCLAn引脚为低电平时 •当IICEn位为“0”（停止运行）时 •当复位时 		<ul style="list-style-type: none"> •当SCLAn引脚为高电平时

DADn	SDAAn引脚的电平检测（只在IICEn位为“1”时有效）	
0	检测到SDAAn引脚为低电平。	
1	检测到SDAAn引脚为高电平。	
清除条件（DADn=0）		置位条件（DADn=1）
<ul style="list-style-type: none"> •当SDAAn引脚为低电平时 •当IICEn位为“0”（停止运行）时 •当复位时 		<ul style="list-style-type: none"> •当SDAAn引脚为高电平时

SMCn	运行模式的切换
0	在标准模式中运行（最大传送速率：100kbps）。
1	在快速模式（最大传送速率：400kbps）或者增强型快速模式（最大传送速率：1Mbps）中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器OFF
1	数字滤波器ON
必须在快速模式或者增强型快速模式中使用数字滤波器。 数字滤波器用于消除噪声。 无论是将DFCn位置“1”还是清“0”，传送时钟都不变。	

PRSn	运行时钟（ f_{MCK} ）的控制
0	选择 f_{CLK} （ $1MHz \leq f_{CLK} \leq 20MHz$ ）。
1	选择 $f_{CLK}/2$ （ $20MHz < f_{CLK}$ ）。

注意1.IICA运行时钟（ f_{MCK} ）的最大工作频率为20MHz(Max.)。只有在 f_{CLK} 超过20MHz时才必须将IICA控制寄存器n1（IICCTLn1）的bit0（PRSn）置“1”。

2.在设置传送时钟的情况下，必须注意 f_{CLK} 的最小工作频率。串行接口IICA的 f_{CLK} 最小工作频率取决于运行模式。

快速模式： $f_{CLK}=3.5MHz(\text{Min.})$

增强型快速模式： $f_{CLK}=10MHz(\text{Min.})$

标准模式： $f_{CLK}=1MHz(\text{Min.})$

备注1.IICEn: IICA控制寄存器n0（IICCTLn0）的bit7

2.n=0

14.3.6 IICA低电平宽度设置寄存器n (IICWLn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号低电平宽度 (t_{LOW}) 和SDAAn引脚信号。

通过8位存储器操作指令设置IICWLn寄存器。

必须在禁止I²C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设置IICWLn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

有关IICWLn寄存器的设置方法, 请参照“14.4.2通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法”。

数据保持时间为IICWLn所设时间的1/4。

图14-10 IICA低电平宽度设置寄存器n (IICWLn) 的格式



14.3.7 IICA高电平宽度设置寄存器n (IICWHn)

此寄存器控制串行接口IICA输出的SCLAn引脚信号高电平宽度和SDAAn引脚信号。通过8位存储器操作指令设置IICWHn寄存器。

必须在禁止I²C运行 (IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0) 时设置IICWHn寄存器。在产生复位信号后, 此寄存器的值变为“FFH”。

图14-11 IICA高电平宽度设置寄存器n (IICWHn) 的格式



备注1.有关主控方传送时钟的设置方法, 请参照14.4.2(1); 有关从属方IICWLn寄存器和IICWHn寄存器的设置方法, 请参照14.4.2(2)。

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P00以外的任意端口。

通过设置端口复用功能配置寄存器（PxxCFG）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 I²C总线模式的功能

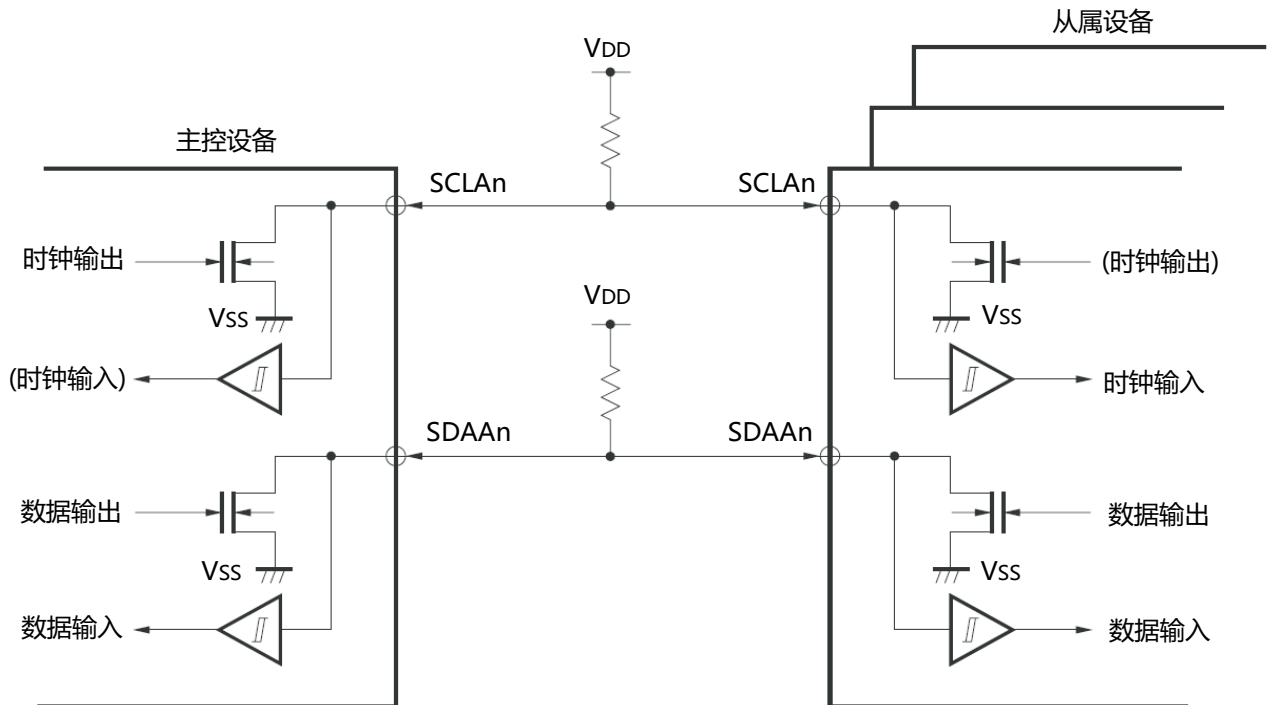
14.4.1 引脚结构

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14-12 引脚结构图



备注 n=0

14.4.2 通过IICWLn寄存器和IICWHn寄存器设置传送时钟的方法

(1) 主控方传送时钟的设置方法

$$\text{传送时钟} = \frac{f_{\text{MCK}}}{\text{IICWL} + \text{IICWH} + f_{\text{MCK}}(t_{\text{R}} + t_{\text{F}})}$$

此时，IICWLn寄存器和IICWHn寄存器的最佳设置值如下：
(全部设置值的小数部分都舍入)

- 快速模式

$$\text{IICWLn} = \frac{0.52}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.48}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWLn} = \frac{0.47}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.53}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWLn} = \frac{0.50}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.50}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

(2) 从属方IICWLn寄存器和IICWHn寄存器的设置方法 (全部设置值的小数部分都舍入)

- 快速模式

$$\text{IICWLn} = 1.3\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (1.2\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWLn} = 4.7\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (5.3\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWLn} = 0.50\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (0.50\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

注意1. IICA运行时钟 (f_{MCK}) 的最大工作频率为20MHz(Max.)。只有在 f_{CLK} 超过20MHz时才必须将IICA控制寄存器n1 (IICCTLn1) 的bit0 (PRSn) 置“1”。

2. 在设置传送时钟的情况下，必须注意 f_{CLK} 的最小工作频率。串行接口IICA的 f_{CLK} 最小工作频率取决于运行模式。

快速模式 : $f_{\text{CLK}} = 3.5\text{MHz}(\text{Min.})$

增强型快速模式 : $f_{\text{CLK}} = 10\text{MHz}(\text{Min.})$

标准模式 : $f_{\text{CLK}} = 1\text{MHz}(\text{Min.})$

备注1.因为SDAAn信号和SCLAn信号的上升时间 (t_R) 和下降时间 (t_F) 因上拉电阻和布线电容而不同, 所以必须各自计算。

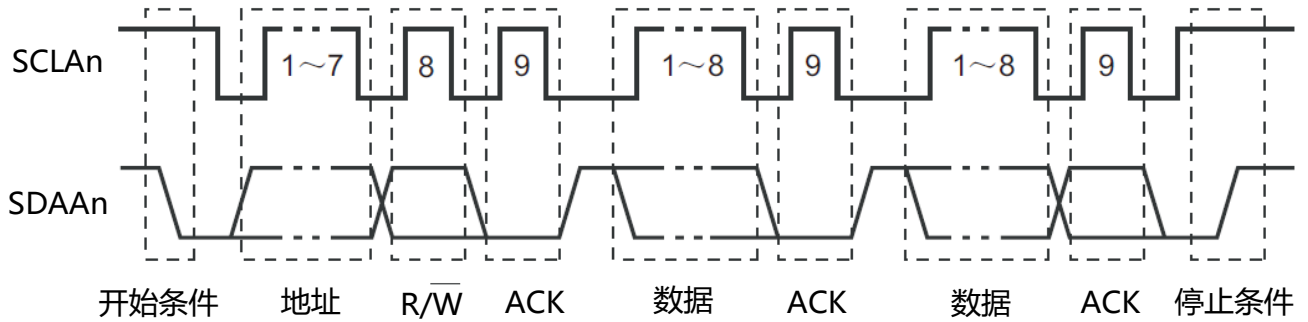
- 2.IICWLn : IICA低电平宽度设置寄存器n
- IICWHn : IICA高电平宽度设置寄存器n
- t_F : SDAAn信号和SCLAn信号的下降时间
- t_R : SDAAn信号和SCLAn信号的上升时间
- f_{MCK} : IICA运行时钟频率
- 3.n=0

14.5 I²C总线的定义和控制方法

以下说明I²C总线的串行数据通信格式和使用的信号。

I²C总线的串行数据总线上生成的“开始条件”、“地址”、“数据”和“停止条件”的各传送时序如下图所示。

图14-13 I²C总线的串行数据传送时序



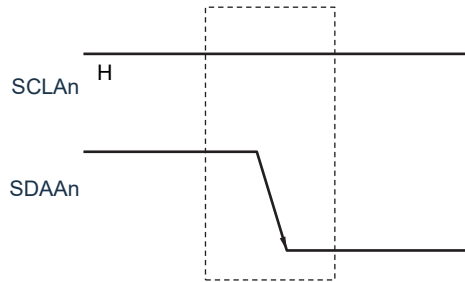
主控设备生成开始条件、从属地址和停止条件。

主控设备和从属设备都能生成应答（ACK）（在一般情况下，接收方输出8位数据）。主控设备连续输出串行时钟（SCLAn）。但是，从属设备能延长SCLAn引脚的低电平期间并且插入等待。

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。

图14-14 开始条件



在检测到停止条件（SPDn: IICA状态寄存器n (IICSn) 的bit0=1) 的状态下, 如果将IICA控制寄存器n0 (IICCTLn0) 的bit1 (STTn) 置“1”, 就输出开始条件。如果检测到开始条件, 就将IICSn寄存器的bit1 (STDn) 置“1”。

备注 n=0

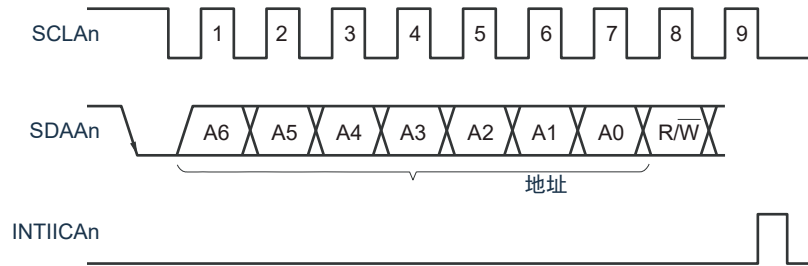
14.5.2 地址

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置完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查7位数据是否和从属地址寄存器n（SVAn）的内容相同。此时，如果7位数据和SVAn寄存器的值相同，该从属设备就被选中，在主控设备生成开始条件或者停止条件前，与主控设备进行通信。

图14-15 地址



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

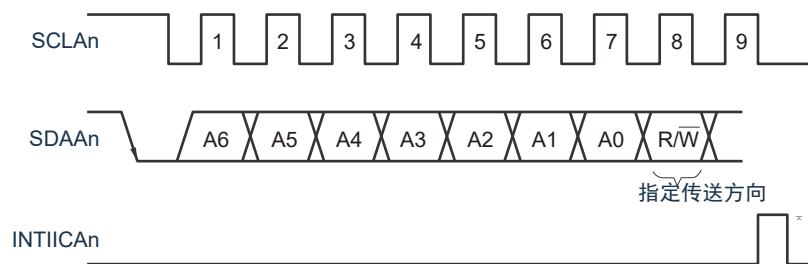
如果将从属地址和“14.5.3 传送方向的指定”中说明的传送方向构成的8位数据写到IICA移位寄存器n（IICAn），就输出地址。接收到的地址被写到IICAn寄存器。从属地址分配在IICAn寄存器的高7位。

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。

图14-16 传送方向的指定



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

备注 n=0

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。

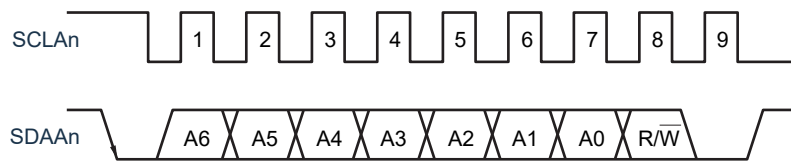
接收方在第9个时钟将SDAAn线置为低电平，生成应答 (正常接收)。

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位数据设置IICSn寄存器的bit3 (TRCn)。在接收 (TRCn=0) 的情况下，通常必须将ACKEn位置“1”。

在从属接收运行过程中 (TRCn=0) 不能接收数据或者不需要下一个数据时，必须将ACKEn位清“0”，通知主控方不能接收数据。

在主控接收运行过程中 (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14-17 应答



当接收到本地站的地址时，与ACKEn位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答 (NACK)。

在接收到扩展码时，通过事先将ACKEn位置“1”，生成应答。接收数据时的应答生成方法因等待时序的设置而不同，如下所示。

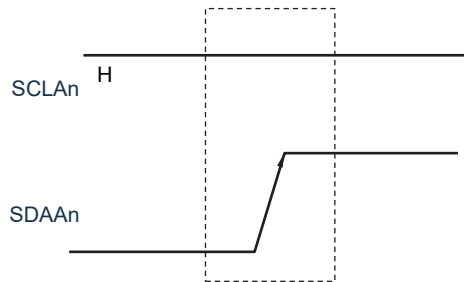
- 当选择8个时钟的等待时 (IICCTLn0寄存器的bit3 (WTIMn) =0): 通过在解除等待前将ACKEn位置“1”，与SCLAn引脚的第8个时钟下降沿同步生成应答。
- 当选择9个时钟的等待时 (IICCTLn0寄存器的bit3 (WTIMn) =1): 通过事先将ACKEn位置“1”，生成应答。

备注 n=0

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14-18 停止条件



如果将IICA控制寄存器n0 (IICCTLn0) 的bit0 (SPTn) 置“1”，就生成停止条件。如果检测到停止条件，就将IICA状态寄存器n (IICSn) 的bit0 (SPDn) 置“1”，并且在IICCTLn0寄存器的bit4 (SPIEn) 为“1”时产生INTIICAn。

备注 n=0

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14-19 等待(1/2)

- (1) 主控设备为9个时钟等待，从属设备为8个时钟等待的情况
 （主控设备：发送，从属设备：接收，ACKEn=1）

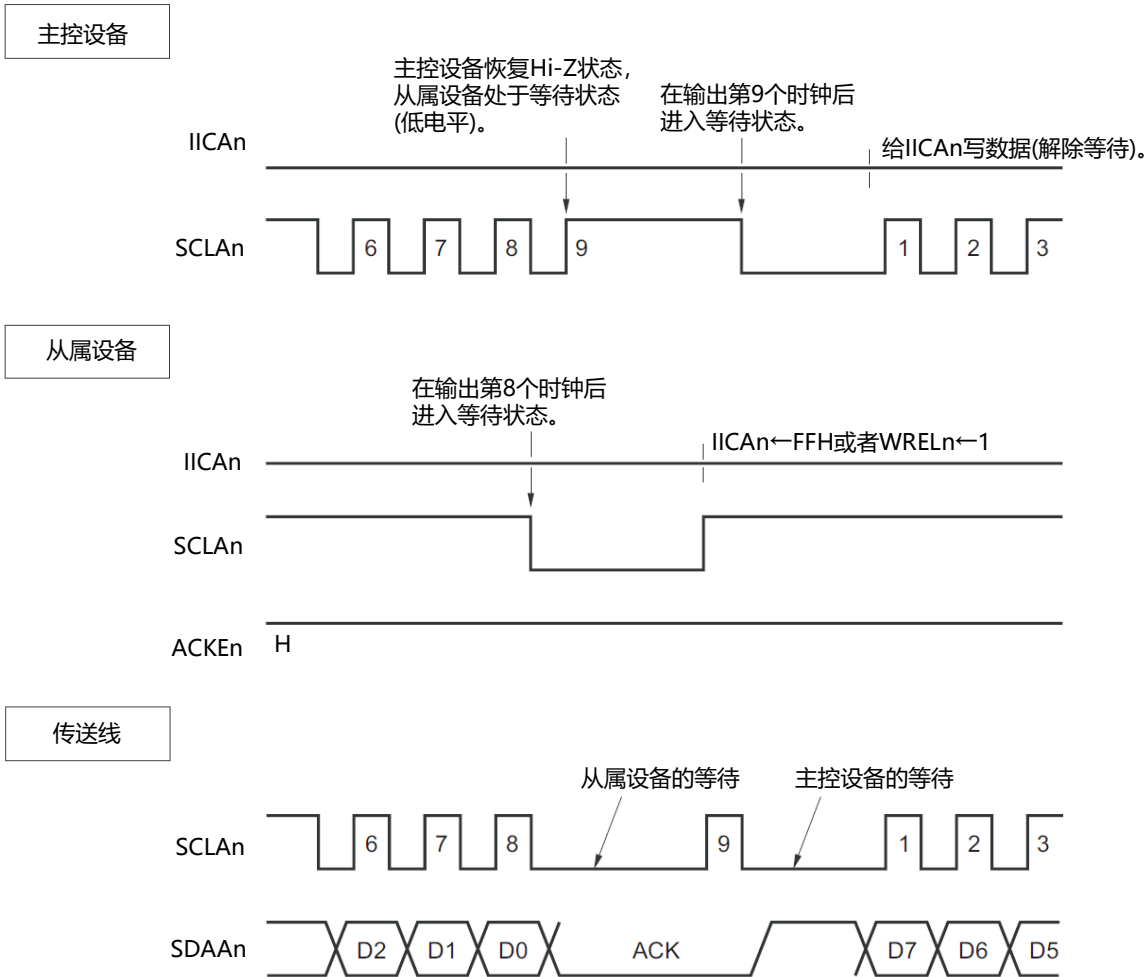
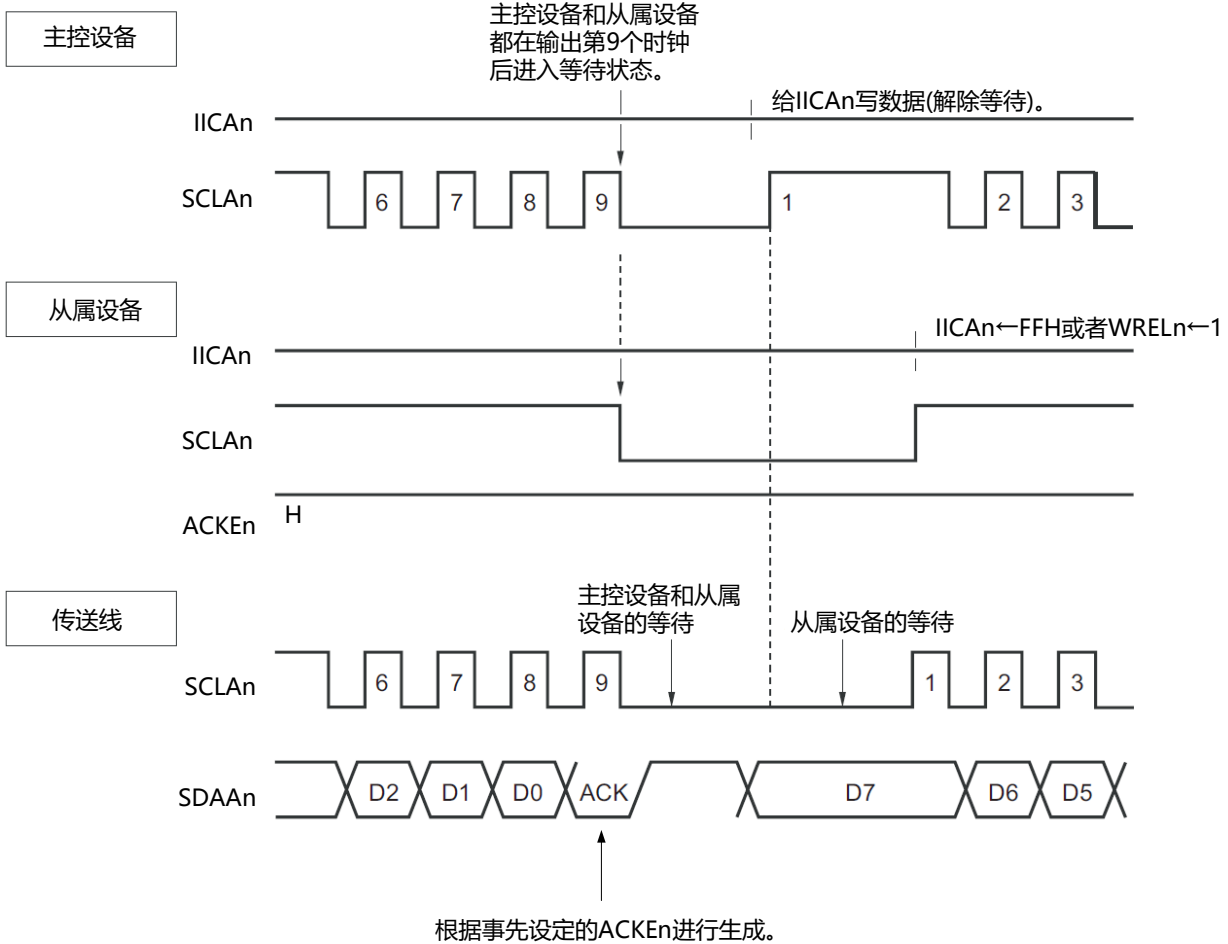


图14-19 等待(2/2)

(2) 主控设备和从属设备都为9个时钟等待的情况
 (主控设备：发送，从属设备：接收，ACKEn=1)



备注 ACKEn: IICA控制寄存器n0 (IICCTLn0) 的bit2
 WRELn: IICA控制寄存器n0 (IICCTLn0) 的bit5

通过设置IICA控制寄存器n0 (IICCTLn0) 的bit3 (WTIMn) 自动产生等待状态。通常，在接收方，如果IICCTLn0寄存器的bit5 (WRELn) 为“1”或者给IICA移位寄存器n (IICAn) 写“FFH”，就解除等待；在发送方，如果给IICAn寄存器写数据，就解除等待。主控设备还能通过以下方法解除等待：

- 将IICCTLn0寄存器的bit1 (STTn) 置“1”。
- 将IICCTLn0寄存器的bit0 (SPTn) 置“1”。

备注 n=0

14.5.7 等待的解除方法

在一般情况下，I²C能通过以下的处理来解除等待。

- 给IICA移位寄存器n (IICAn) 写数据。
- 将IICA控制寄存器n0 (IICCTLn0) 的bit5 (WRELn) 置位 (解除等待)。
- 将IICCTLn0寄存器的bit1 (STTn) 置位 (生成开始条件) 注。
- 将IICCTLn0寄存器的bit0 (SPTn) 置位 (生成停止条件) 注。

注 只限于主控设备。

如果执行了这些等待的解除处理，I²C就解除等待，重新开始通信。要在解除等待后发送数据 (包括地址) 时，必须给IICAn寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将IICCTLn0寄存器的bit5 (WRELn) 置“1”。要在解除等待后生成重新开始条件时，必须将IICCTLn0寄存器的bit1 (STTn) 置“1”。要在解除等待后生成停止条件时，必须将IICCTLn0寄存器的bit0 (SPTn) 置“1”。对于一次等待只能执行一次解除处理。

例如，如果在通过将WRELn位置“1”来解除等待后给IICAn寄存器写数据，SDAAn线的变化时序与IICAn寄存器的写时序就可能发生冲突，导致将错误的值输出到SDAAn线。除了这些处理以外，在中途中止通信的情况下，如果将IICEn位清“0”，就停止通信，因此能解除等待。在I²C总线状态因噪声而被死锁的情况下，如果将IICCTLn0寄存器的bit6 (LRELn) 置“1”，就退出通信，因此能解除等待。

注意 如果在WUPn位为“1”时执行等待的解除处理，就不解除等待。

备注 n=0

14.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设置IICA控制寄存器n0（IICCTLn0）的bit3（WTIMn），在表14-2所示的时序产生INTIICAn并且进行等待控制。

表14-2 INTIICAn的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 ^{注1、2}	8 ^{注2}	8 ^{注2}	9	8	8
1	9 ^{注1、2}	9 ^{注2}	9 ^{注2}	9	9	9

注 1. 只有在接收的地址和从属地址寄存器n（SVAn）的设置地址相同时，从属设备才在第9个时钟的下降沿产生INTIICAn信号并且进入等待状态。

此时，与IICCTLn0寄存器的bit2（ACKEn）的设置无关，生成应答。接收到扩展码的从属设备在第8个时钟的下降沿产生INTIICAn。如果在重新开始后地址不同，就在第9个时钟的下降沿产生INTIICAn，但是不进入等待状态。

2. 如果接收的地址和从属地址寄存器n（SVAn）的内容不同并且未接收到扩展码，就不产生INTIICAn并且也不进入等待状态。

备注 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属运行：与WTIMn位无关，根据上述注1和注2的条件决定中断和等待的时序。
- 主控运行：与WTIMn位无关，在第9个时钟的下降沿产生中断和等待的时序。

(2) 数据接收

- 主控运行/从属运行：通过WTIMn位决定中断和等待的时序。

(3) 数据发送

- 主控运行/从属运行：通过WTIMn位决定中断和等待的时序。

备注 n=0

(4) 等待的解除方法

等待的解除方法有以下4种：

- 给IICA移位寄存器n (IICAn) 写数据。
- 将IICA控制寄存器n0 (IICCTLn0) 的bit5 (WRELn) 置位 (解除等待)。
- 将IICCTLn0寄存器的bit1 (STTn) 置位 (生成开始条件) 注。
- 将IICCTLn0寄存器的bit0 (SPTn) 置位 (生成停止条件) 注。

注 只限于主控设备。

当选择8个时钟的等待 (WTIMn=0) 时，需要在解除等待前决定是否生成应答。

(5) 停止条件的检测

如果检测到停止条件，就产生INTIICAn (只限于SPIEn=1的情况)。

14.5.9 地址匹配的检测方法

在I²C总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。能通过硬件自动检测地址匹配。当主控设备发送的从属地址和从属地址寄存器n（SVAn）的设置地址相同或者只接收到扩展码时，产生INTIICAn中断请求。

14.5.10 错误的检测

在I²C总线模式中，因为发送过程中的串行数据总线（SDAAn）的状态被取到发送器件的IICA移位寄存器n（IICAn），所以能通过将开始发送前和发送结束后的IICA数据进行比较来检测发送错误。此时，如果2个数据不同，就判断为发生了发送错误。

备注 n=0

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寄存器的设置值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求（INTIICAn）。

- 高4位数据相同：EXCn=1
- 7位数据相同：COIn=1

备注 EXCn：IICA状态寄存器n（IICSn）的bit5
COIn：IICA状态寄存器n（IICSn）的bit4

(3)中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将IICA控制寄存器n0（IICCTLn0）的bit6（LRELn）置“1”，进入下一次通信的待机状态。

表14-3 主要扩展码的位定义

从属地址	R/W位	说明
0000000	0	全呼地址
11110xx	0	10位从属地址的指定（地址认证时）
11110xx	1	10位从属地址的指定（在地址相同后发行读命令时）

备注1.有关上述以外的扩展码，请参照NXP公司发行的I²C总线规格书。

2.n=0

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。

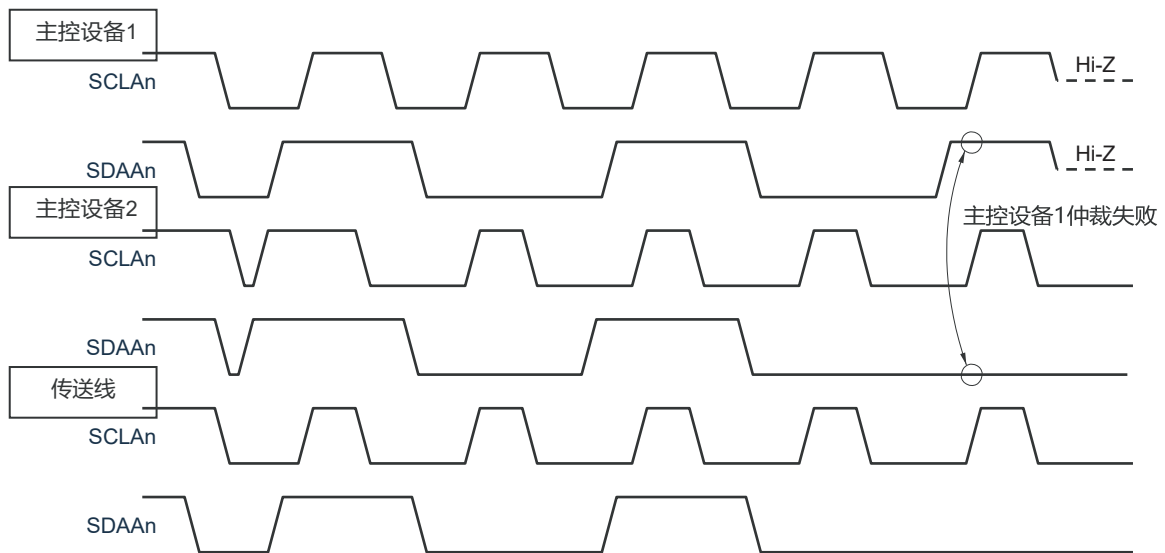
在仲裁失败时，仲裁失败的主控设备将IICA状态寄存器n（IICSn）的仲裁失败标志（ALDn）置“1”，并且将SCLAn线和SDAAn线都置为高阻抗状态，释放总线。

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 STDn: IICA状态寄存器n（IICSn）的bit1
STTn: IICA控制寄存器n0（IICCTLn0）的bit1

图14-20 仲裁时序例子



备注 n=0

表14-4 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时 (SPIEn=1) ^{注2}
想要生成重新开始条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件，但是检测到停止条件。	在生成停止条件时 (SPIEn=1) ^{注2}
想要生成停止条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件，但是SCLAn为低电平。	

注 1.当WTIMn位 (IICA控制寄存器n0 (IICCTLn0) 的bit3) 为“1”时，在第9个时钟的下降沿产生中断请求；当WTIMn位为“0”并且接收到扩展码的从属地址时，在第8个时钟的下降沿产生中断请求。

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I²C的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）的功能。在地址不同的情况下不产生不需要的INTIICAn信号，能提高处理效率。如果检测到开始条件，就进入唤醒待机状态。因为主控设备（已经生成开始条件的情况）也有可能因仲裁失败而变为从属设备，所以在发送地址的同时进入唤醒待机状态。

要在深度睡眠模式中使用唤醒功能时，必须将WUPn位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）。在产生此中断后，通过指令将WUPn位清“0”，返回到通常运行。

将WUPn位置“1”时的流程如图14-21所示，通过地址匹配将WUPn位置“0”时的流程如图14-22所示。

图14-21 将WUPn位置“1”时的流程

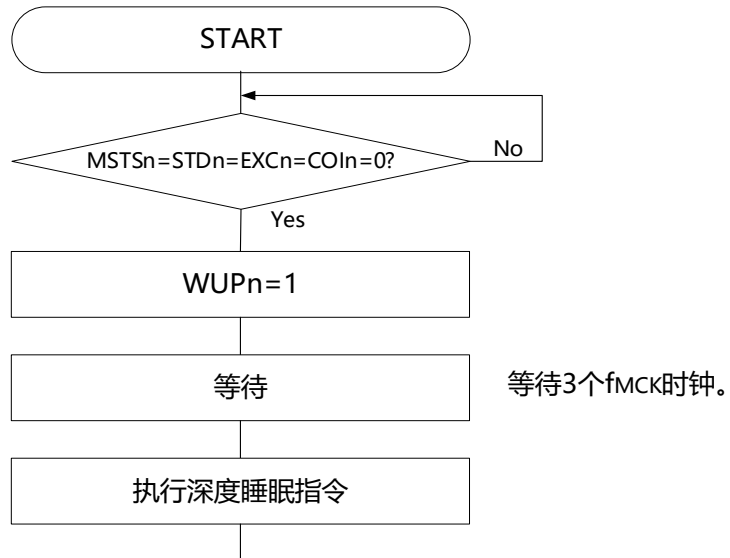
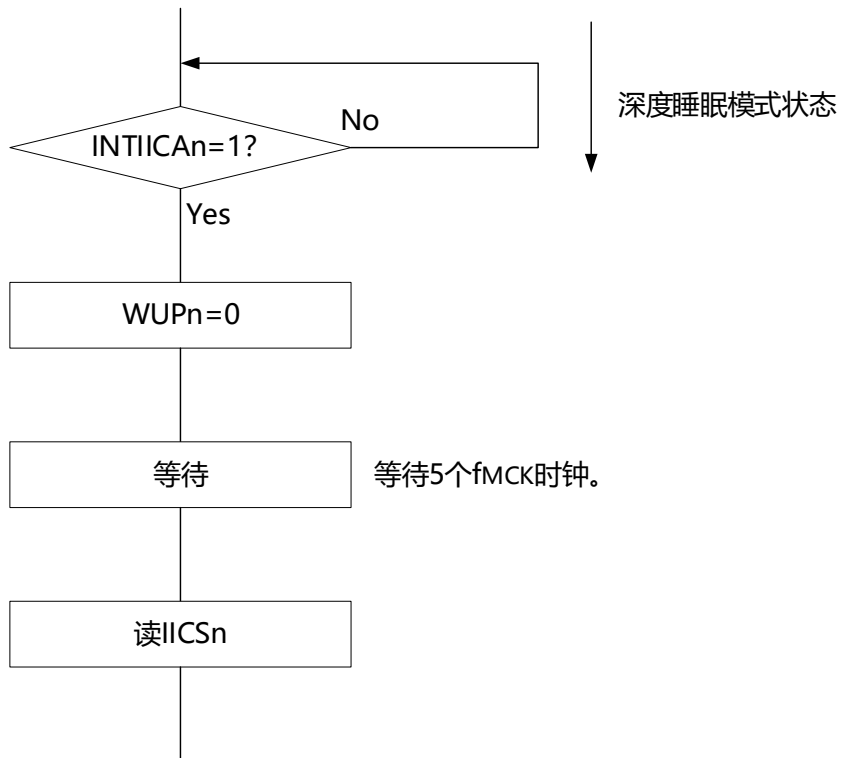


图14-22 通过地址匹配将WUPn位置“0”时的流程（包括接收扩展码）



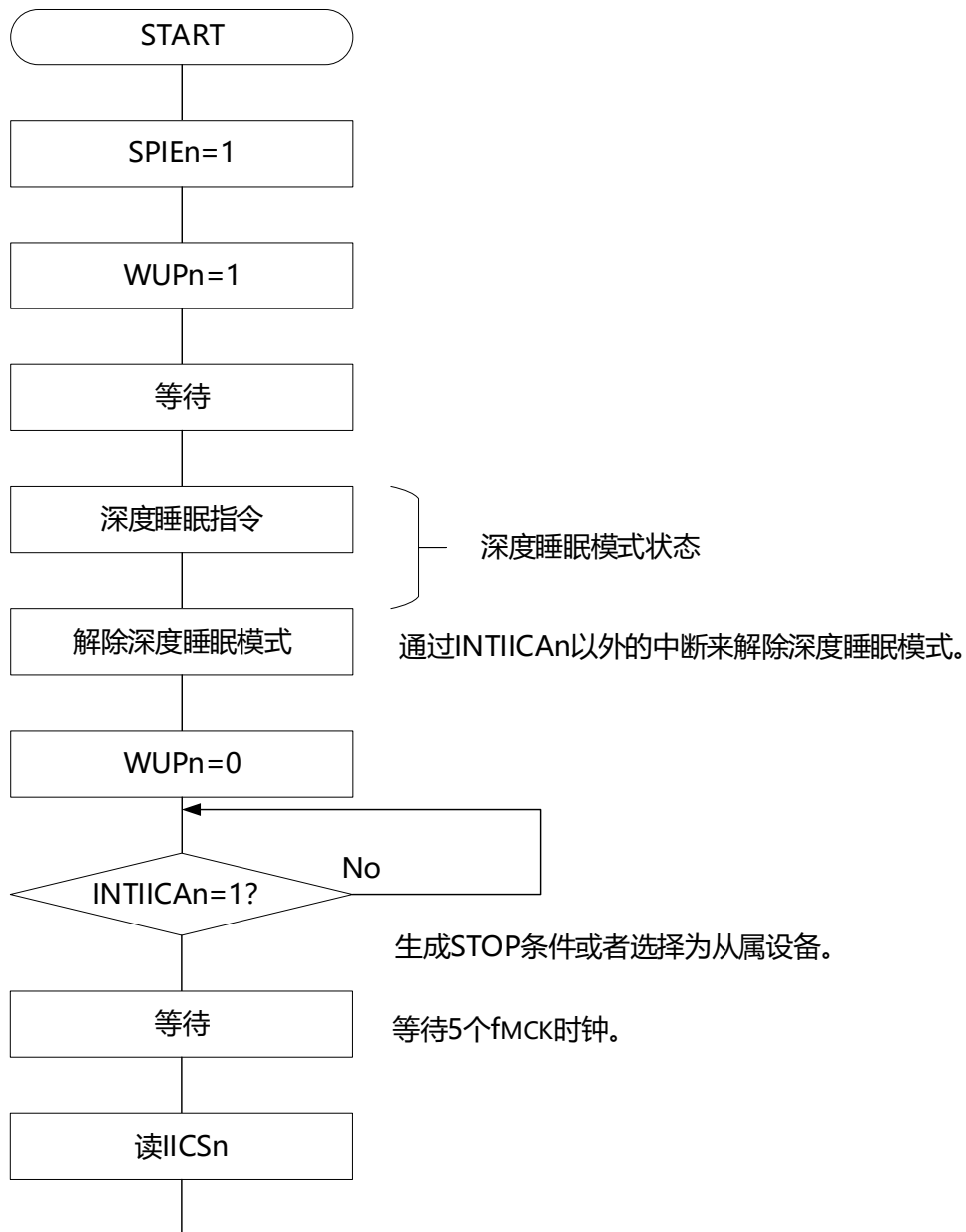
在确认串行接口IICA的运行状态后，
根据要执行的内容进行处理。

除了串行接口IICA产生的中断请求（INTIICAn）以外，必须通过以下的流程解除深度睡眠模式。

- 下一次IIC通信为主控设备运行的情况：图14-23的流程
- 下一次IIC通信为从属设备运行的情况：
 - 通过INTIICAn中断返回的情况：和图14-22的流程相同。
 - 通过INTIICAn中断以外的中断返回的情况：必须在产生INTIICAn中断前保持WUPn位为“1”的状态继续运行。

备注 n=0

图14-23 在通过INTIICAn以外的中断来解除深度睡眠模式后作为主控设备运行的情况



在确认串行接口IICA的运行状态后，
根据要执行的内容进行处理。

备注 n=0

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：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将IICA控制寄存器n0 (IICCTLn0) 的bit6 (LRELn) 置“1”，退出通信后释放了总线)

如果在不加入总线的状态下将IICCTLn0寄存器的bit1 (STTn) 置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将IICCTLn0寄存器的bit4 (SPIEn) 置“1”，在通过产生的中断请求信号 (INTIICAn) 检测到总线的释放（检测到停止条件）后，如果给IICA移位寄存器n (IICAn) 写地址，就自动作为主控设备开始通信。在检测到停止条件前，给IICAn寄存器写的的数据无效。

当将STTn位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时.....生成开始条件
- 总线未处于释放状态（待机状态）时.....通信预约

在将STTn位置“1”并且经过等待时间后，通过MSTSn位 (IICA状态寄存器n (IICSn) 的bit7) 确认是否作为通信预约运行。

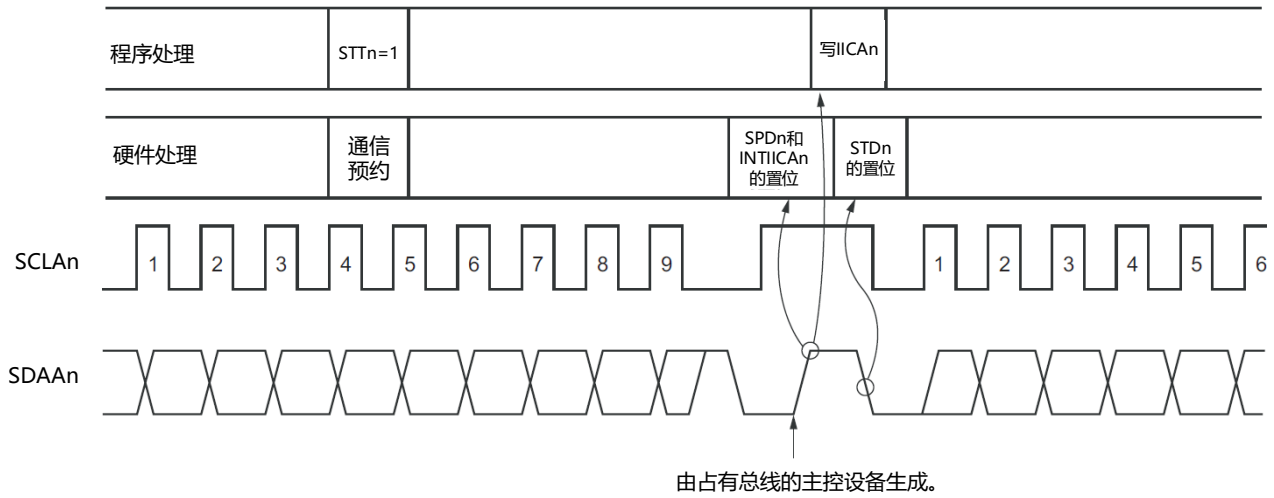
必须通过软件确保以下计算式计算的等待时间：

从将STTn 位置“1” 到确认MSTSn 标志为止的等待时间：
 $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) / f_{MCK} + t_F \times 2$

备注1. IICWLn : IICA低电平宽度设置寄存器n
 IICWHn : IICA高电平宽度设置寄存器n
 t_F : SDAAn信号和SCLAn信号的下降时间
 f_{MCK} : IICA运行时钟频率
 2.n=0

通信预约的时序如下图所示。

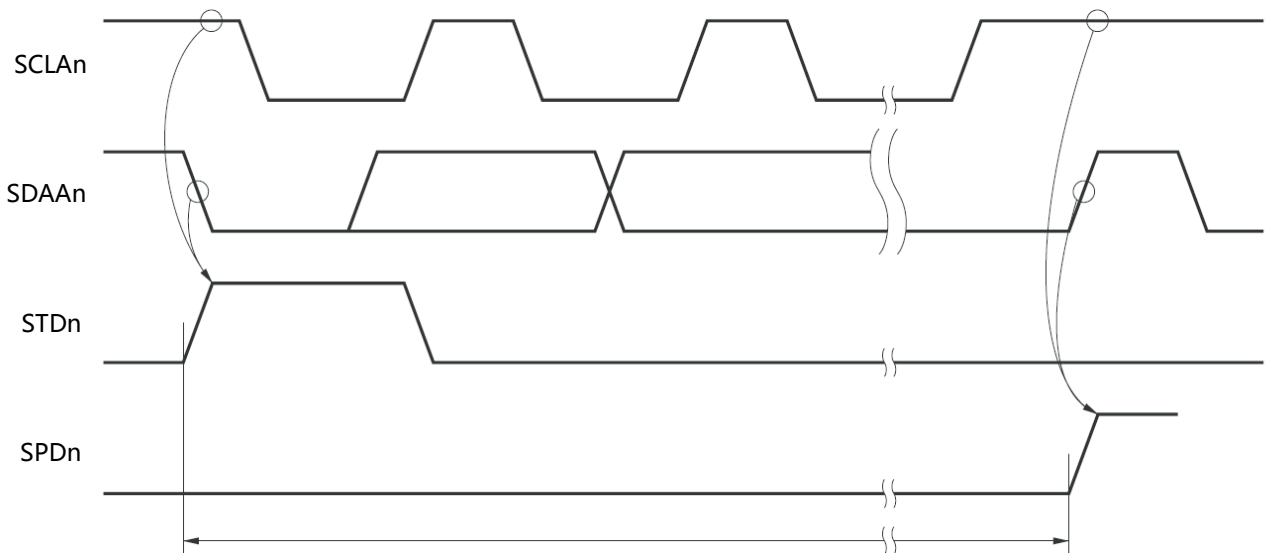
图14-24 通信预约的时序



- 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14-25所示的时序接受通信预约。在IICA状态寄存器n (IICSn) 的bit1 (STDn) 变为“1”后并且在检测到停止条件前, 将IICA控制寄存器n0 (IICCTLn0) 的bit1 (STTn) 置“1”进行通信预约。

图14-25 通信预约的接受时序

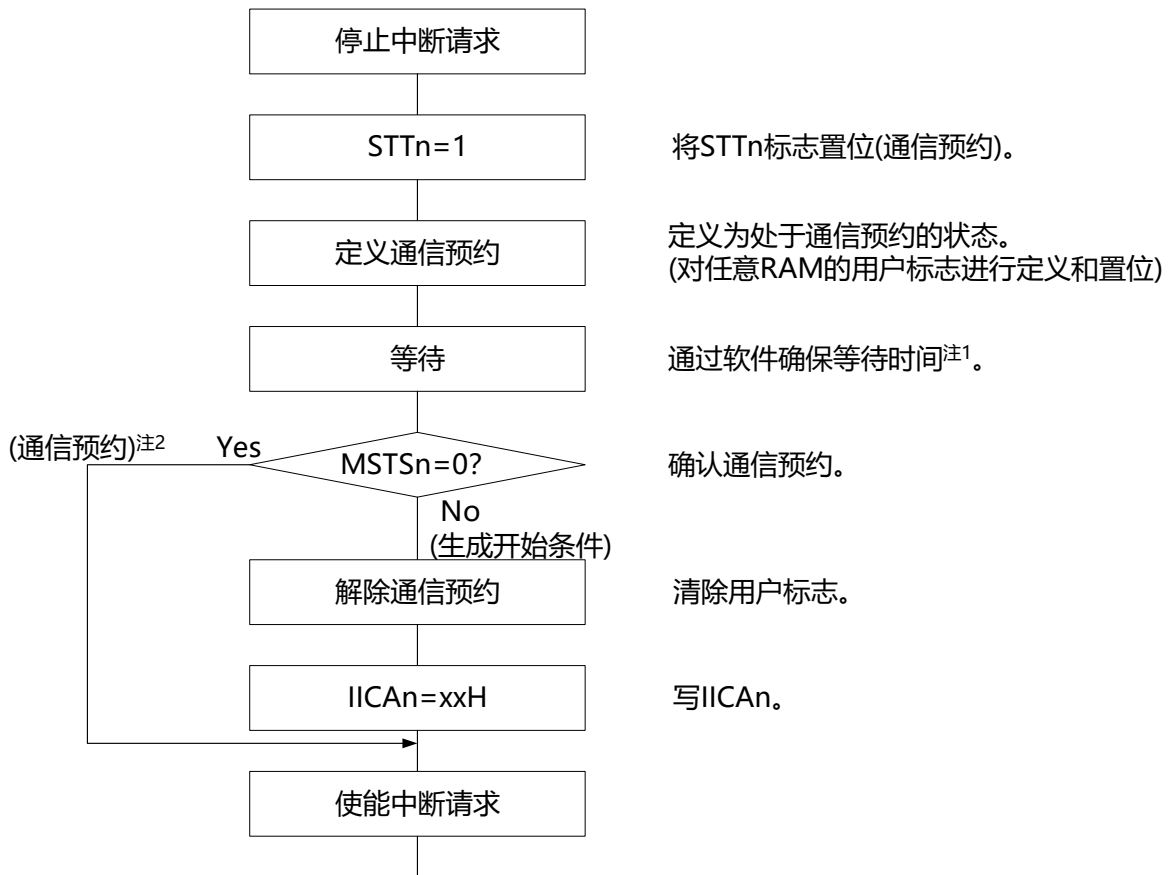


待机状态(在此期间, 能通过将STTn位置“1”进行通信预约)

通信预约的步骤如图14-26所示。

- 备注 n=0

图14-26 通信预约的步骤



注 1.等待时间如下： $(IICWLn的设置值+IICWHn的设置值+4)/f_{MCK}+t_f$ 2
 2.在通信预约运行时，通过停止条件中断请求来写IICA移位寄存器n（IICAn）。

备注1.STTn : IICA控制寄存器n0（IICCTLn0）的bit1
 MSTSn : IICA状态寄存器n（IICSn）的bit7
 IICAn : IICA移位寄存器n
 IICWLn : IICA低电平宽度设置寄存器n
 IICWHn : IICA高电平宽度设置寄存器n
 t_f : SDAAn信号和SCLAn信号的下降时间
 f_{MCK} : IICA运行时钟频率
 2.n=0

(2) 禁止通信预约功能的情况（IICA标志寄存器n（IICFn）的bit0（IICRSVn）=1）

在总线通信过程中，如果不参加此通信的状态下将IICA控制寄存器n0（IICCTLn0）的bit1（STTn）置“1”，就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下2种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将IICCTLn0寄存器的bit6（LRELn）置“1”，退出通信后释放了总线）

能通过STCFn（IICFn寄存器的bit7）来确认是生成了开始条件还是拒绝了请求。因为从STTn位为“1”到将STCFn位置“1”为止需要5个 f_{MCK} 时钟的时间，所以必须通过软件确保此时间。

备注 n=0

14.5.15 其他注意事项

(1) STCENn位为“0”的情况

在刚允许I²C运行 (IICEn=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSYn=1)。要在没有检测到停止条件的状态下进行主控通信时, 必须先生成停止条件, 在释放总线后进行主控通信。对于多主控, 在总线未释放 (未检测到停止条件) 的状态下不能进行主控通信。按照以下顺序生成停止条件:

- ①设置IICA控制寄存器n1 (IICCTLn1)。
- ②将IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) 置“1”。
- ③将IICCTLn0寄存器的bit0 (SPTn) 置“1”。

(2) STCENn位为“1”的情况

在刚允许I²C运行 (IICEn=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSYn=0)。因此在生成第1个开始条件 (STTn=1) 时, 为了不破坏其他的通信, 需要确认总线已被释放。

(3) 正在和其他设备进行I²C通信的情况

在SDAAn引脚为低电平并且SCLAn引脚为高电平时, 如果允许I²C运行而且中途参加通信, I²C的宏就视为SDAAn引脚从高电平变为低电平 (检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的I²C通信。为了避免这种情况, 必须按照以下顺序启动I²C:

- ①将IICCTLn0寄存器的bit4 (SPIEn) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIICAn)。
- ②将IICCTLn0寄存器的bit7 (IICEn) 置“1”, 允许I²C运行。
- ③等待检测到开始条件。
- ④在返回应答前 (将IICEn位置“1”后的4~72个f_{MCK}时钟内) 将IICCTLn0寄存器的bit6 (LRELn) 置“1”, 强制将检测置为无效。

(4) 在将STTn位和SPTn位 (IICCTLn0寄存器的bit1和bit0) 置位后, 禁止清“0”前的再置位。

(5) 如果进行了通信预约, 就必须将SPIEn位 (IICCTLn0寄存器的bit4) 置“1”, 在检测到停止条件时产生中断请求。在产生中断请求后, 通过给IICA移位寄存器n (IICAn) 写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因在开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测MSTSn位 (IICA状态寄存器n (IICSn) 的bit7) 时, 不需要将SPIEn位置“1”。

备注 n=0

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“初始设置”和“通信处理”。在启动时执行“初始设置”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在I²C总线的多主控系统中，只根据I²C总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1帧）为高电平，就将总线作为释放状态而参加通信。此流程大体分为“初始设置”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设置”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

用作I²C总线从属设备的例子如下所示。

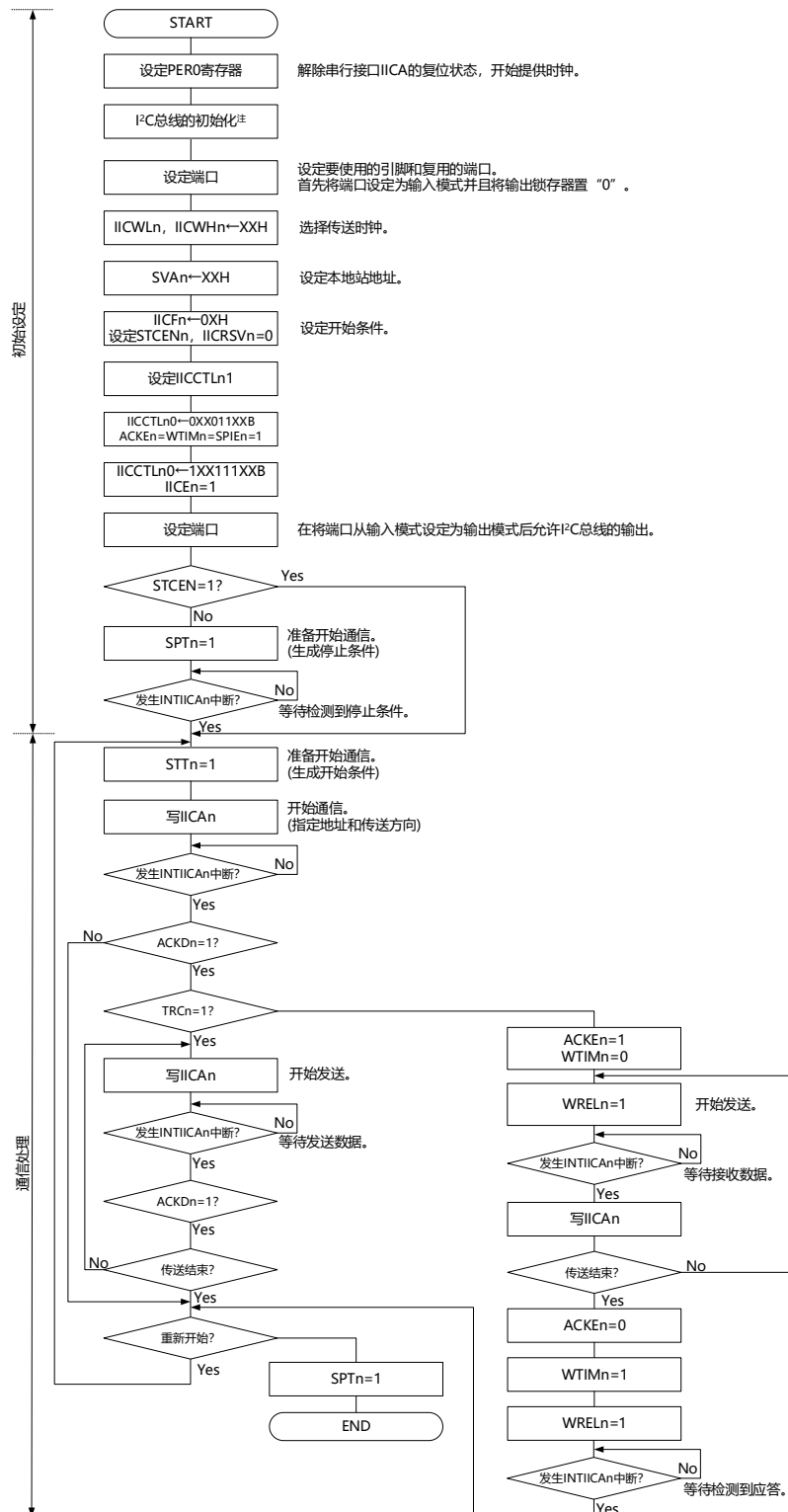
当用作从属设备时，通过中断开始运行。在启动时执行“初始设置”部分，然后通过“通信等待”，等待INTIICAn中断的发生。如果发生INTIICAn中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注 n=0

(1) 单主控系统的主控运行

图14-27 单主控系统的主控运行



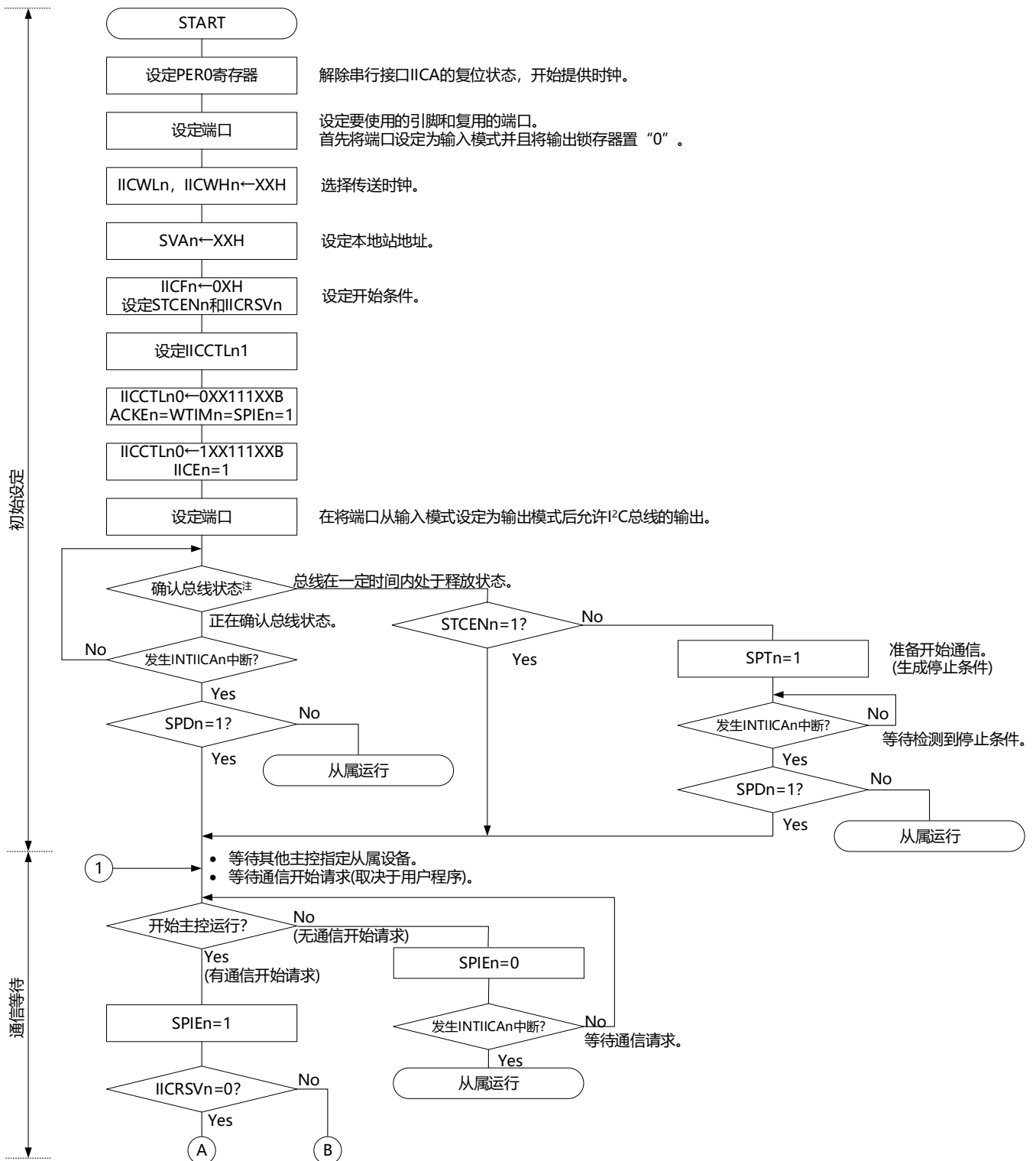
注 必须根据通信中的产品的规格，释放I²C总线（SCLAn引脚和SDAAn引脚为高电平）。例如，如果EEPROM处于给SDAAn引脚输出低电平的状态，就必须将SCLAn引脚设置为输出端口，并且在SDAAn引脚固定为高电平前从输出端口输出时钟脉冲。

备注1.发送和接收的格式必须符合通信中的产品的规格。

2.n=0

(2) 多主控系统的主控运行

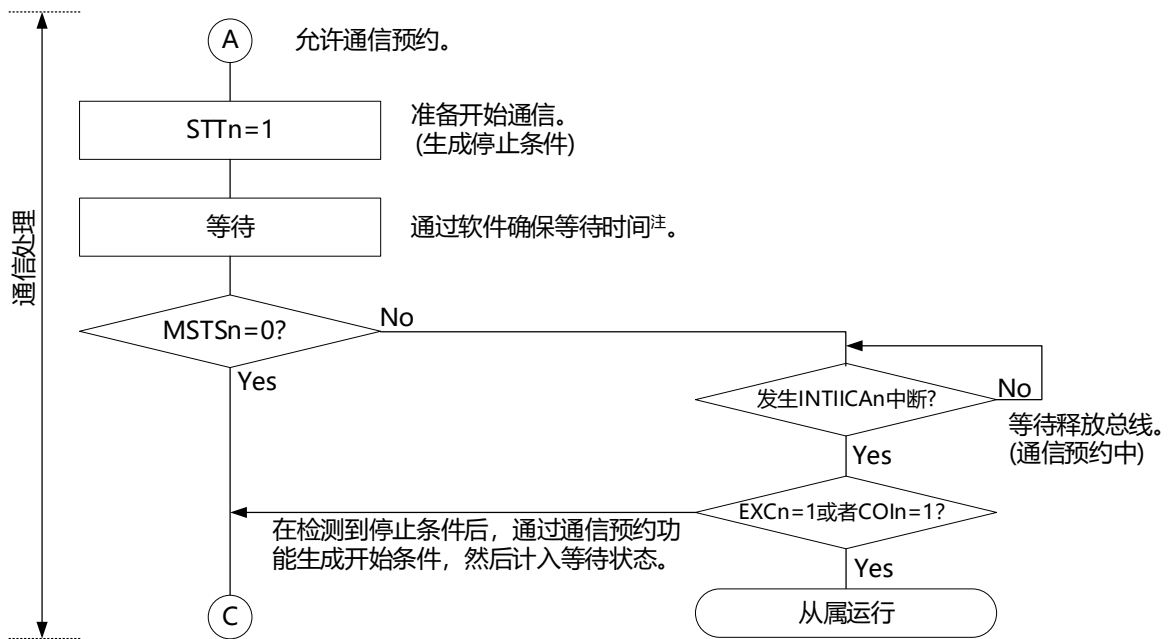
图14-28 多主控系统的主控运行(1/3)



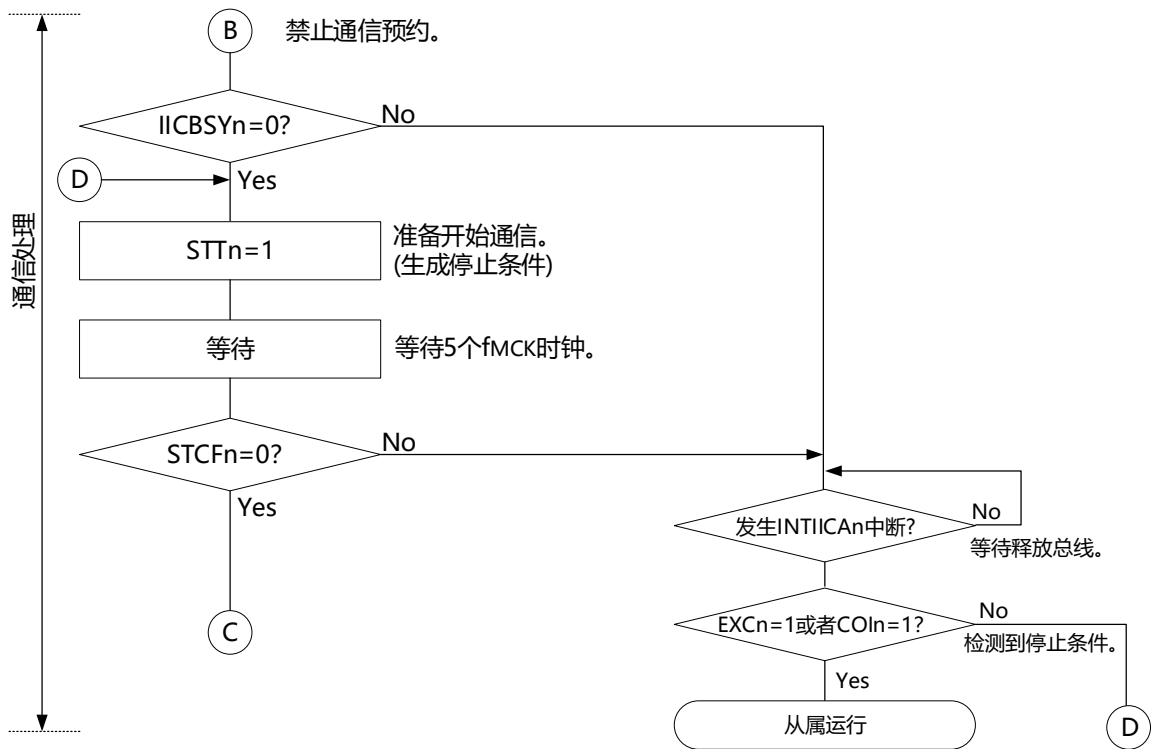
注 必须确认总线在一定时间内(例如, 1帧)处于释放状态(CLDn位=1、DADn位=1)。当SDAAn引脚固定为低电平时, 必须根据通信中的产品的规格, 判断是否释放I²C总线(SCLAn引脚和SDAAn引脚为高电平)。

备注 n=0

图14-28 多主控系统的主控运行(2/3)

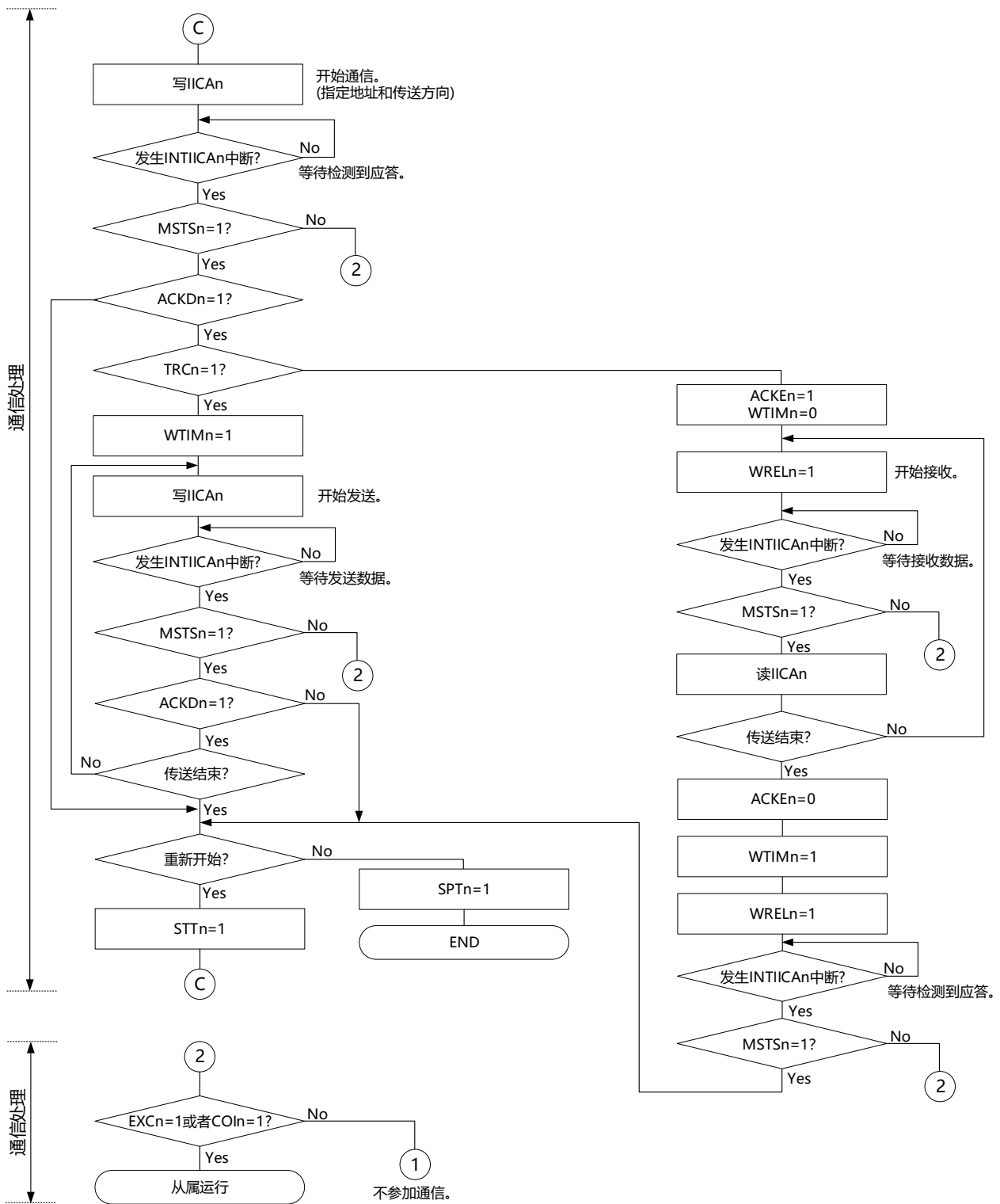


注 等待时间如下:
 $(IICWLn的设置 + IICWHn的设置 + 4) / f_{MCK} + t_F \times 2$



备注1. IICWLn : IICA低电平宽度设置寄存器n
 IICWHn : IICA高电平宽度设置寄存器n
 t_F : SDAAn信号和SCLAn信号的下降时间
 f_{MCK} : IICA运行时钟频率
 2.n=0

图14-28 多主控系统的主控运行(3/3)



备注1. 传送和接收的格式必须符合通信中的产品的规格。

2. 在多主控系统中用作主控设备的情况下，必须在每次发生INTIICAn中断时读MSTSn位，确认仲裁结果。

3. 在多主控系统中用作从属设备的情况下，必须在每次发生INTIICAn中断时通过IICA状态寄存器n (IICSn) 和 IICA标志寄存器n (IICFn) 确认状态，决定以后的处理。

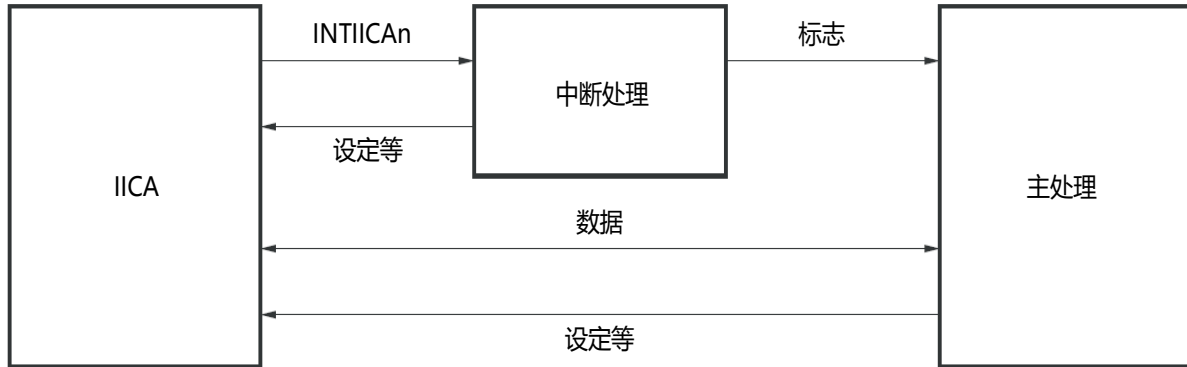
4. n=0

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过INTIICAn中断进行处理（需要对通信中的停止条件检测等的运行状态进行很大的变更处理）。

在此说明中，假设数据通信不支持扩展码，INTIICAn中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下3个标志并且代替INTIICAn将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下2种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态（有效地址的检测～停止条件的检测，未检测到主控设备的应答，地址不同）

② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和INTIICAn中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第1个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第1个数据（地址匹配被解释为下一次数据请求）。

③ 通信方向标志

此标志表示通信方向，和TRCn位的值相同。

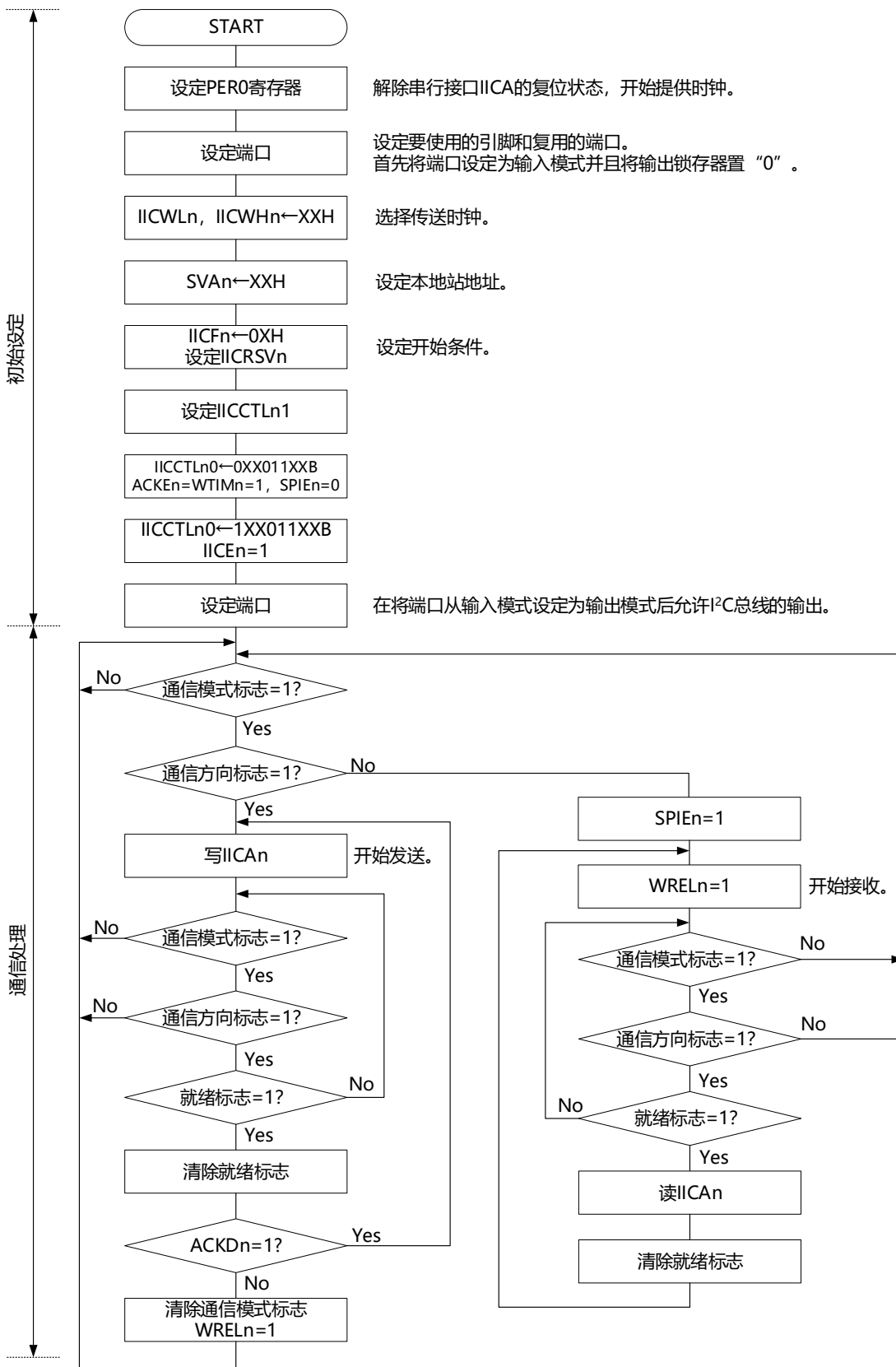
备注 n=0

从属运行的主处理部的运行如下所示。

启动串行接口IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图14-29 从属运行步骤(1)



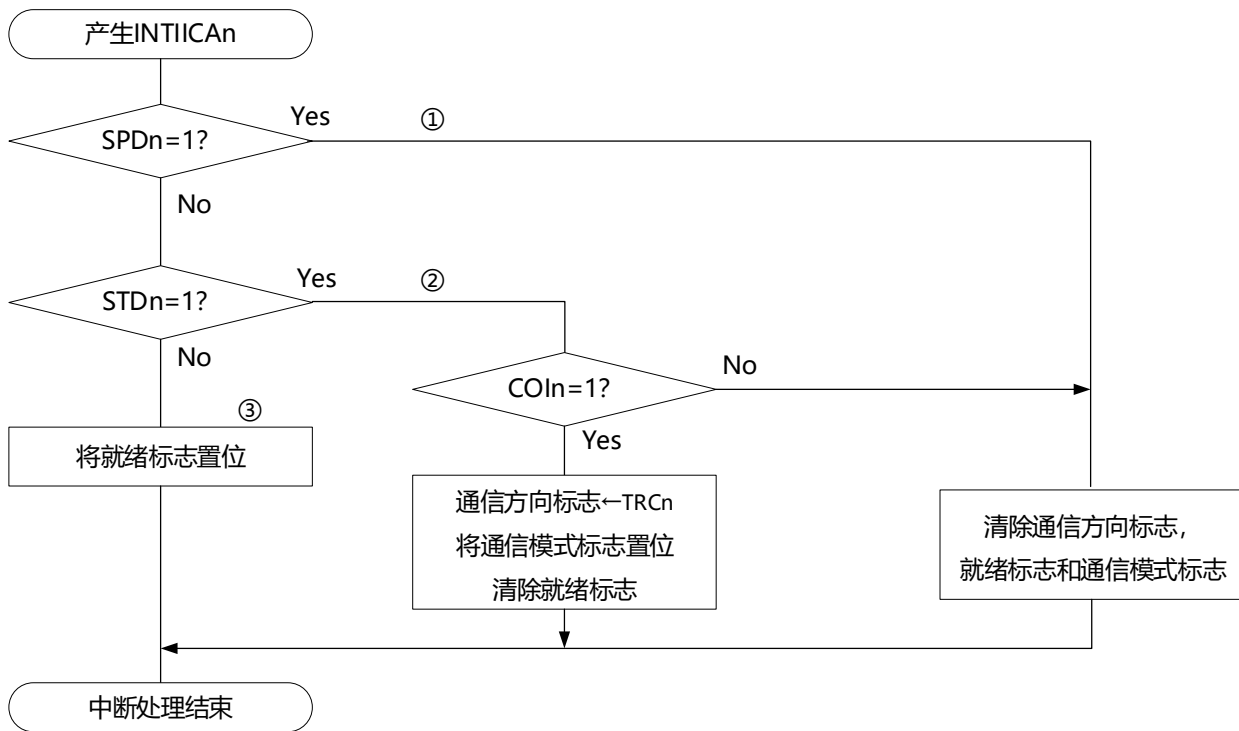
备注 1. 传送和接收的格式必须符合通信中的产品的规格。
2. n=0

从属设备通过INTIICAn中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过INTIICAn中断确认状态并且进行以下处理。

- ①如果生成停止条件，就结束通信。
- ②如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设置为通信模式并且解除等待，然后从中断返回（清除就绪标志）。
- ③当发送和接收数据时，只要将就绪标志置位，I²C总线就保持等待状态并且从中断返回。

备注 上述的①~③对应“图14-30 从属运行步骤(2)”的①~③。

图14-30 从属运行步骤(2)



备注 n=0

14.5.17 I²C中断请求（INTIICAn）的产生时序

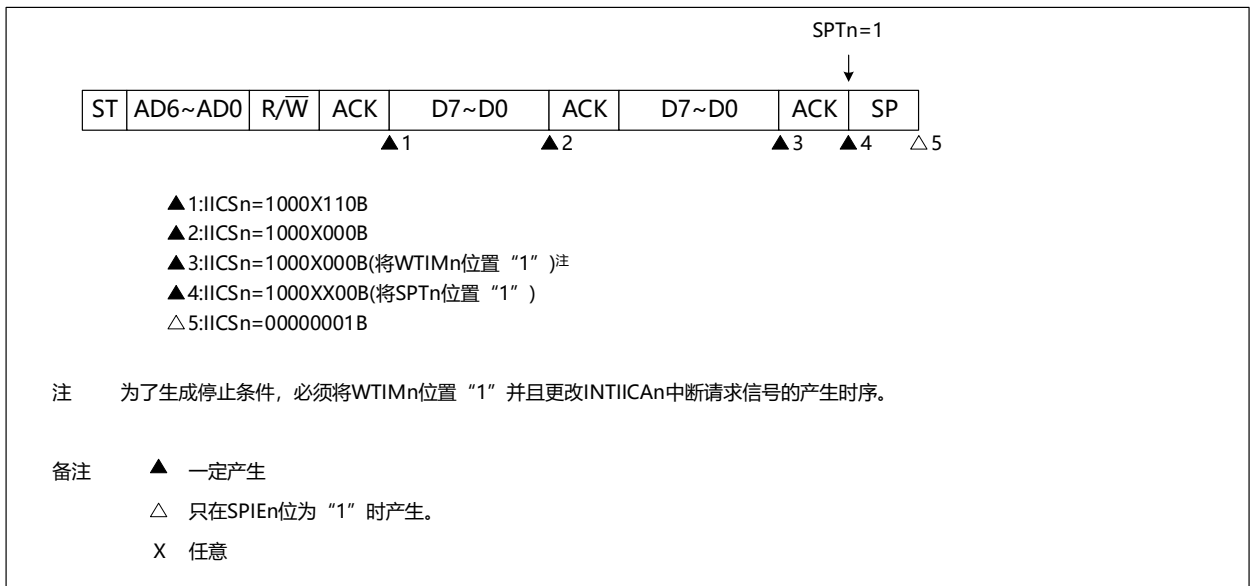
数据的发送和接收时序、INTIICAn中断请求信号的产生时序以及产生INTIICAn信号时的IICA状态寄存器n（IICSn）的值如下所示。

- 备注1.ST : 开始条件
AD6~AD0 : 地址
RW : 传送方向的指定
ACK : 应答
D7~D0 : 数据
SP : 停止条件
2.n=0

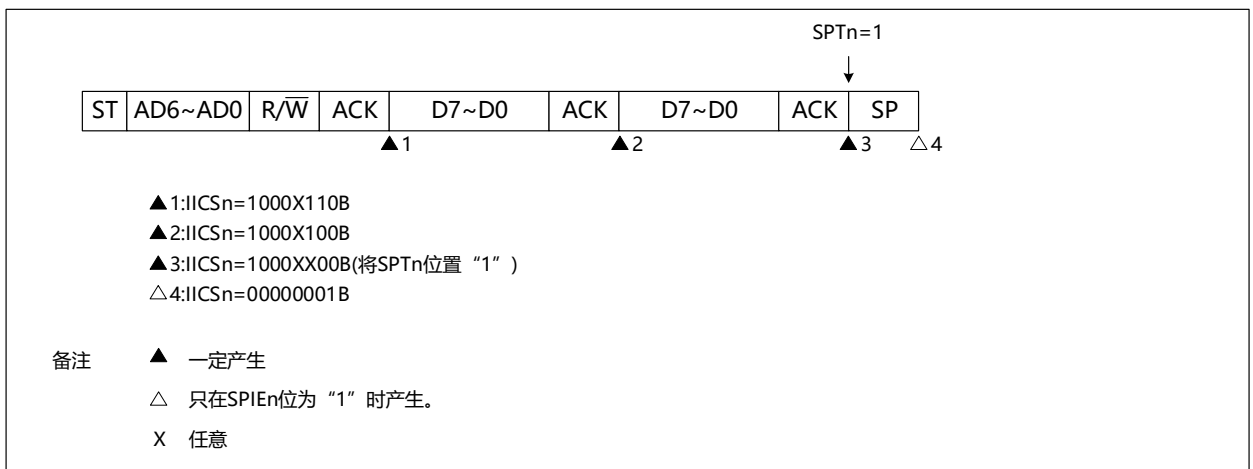
(1) 主控运行

(a) Start~Address~Data~Data~Stop (发送和接收)

(i) WTIMn=0的情况



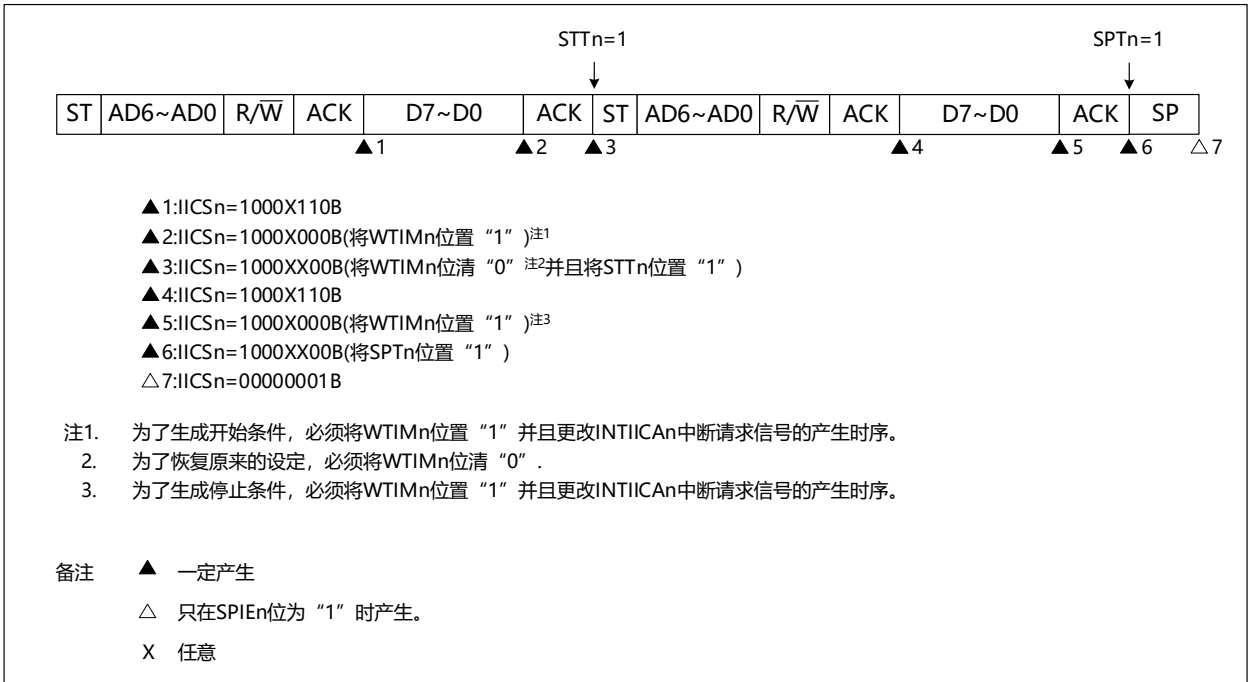
(ii) WTIMn=1的情况



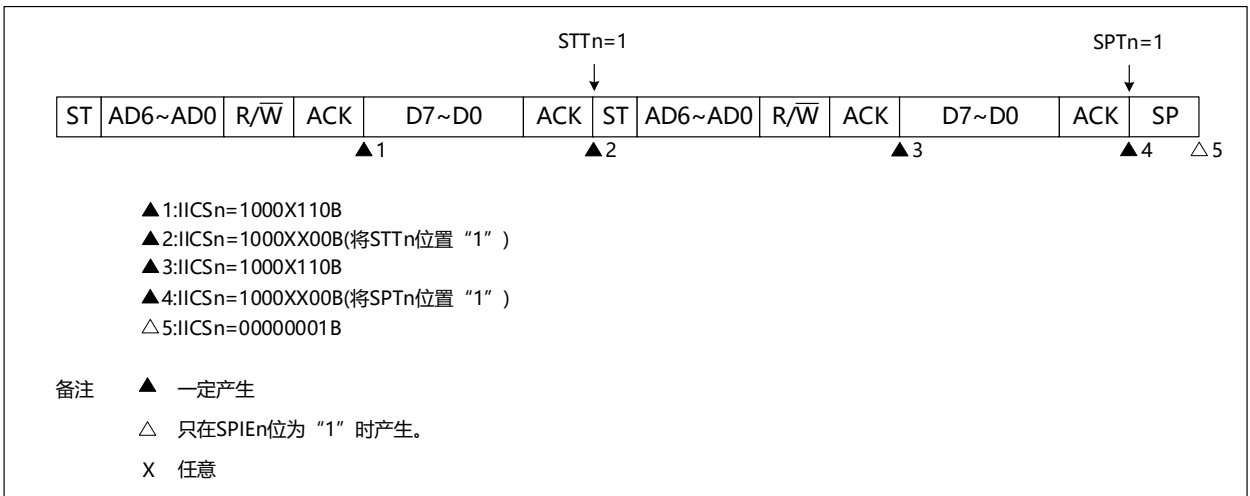
备注 n=0

(b) Start~Address~Data~Start~Address~Data~Stop (重新开始)

(i) WTIMn=0的情况



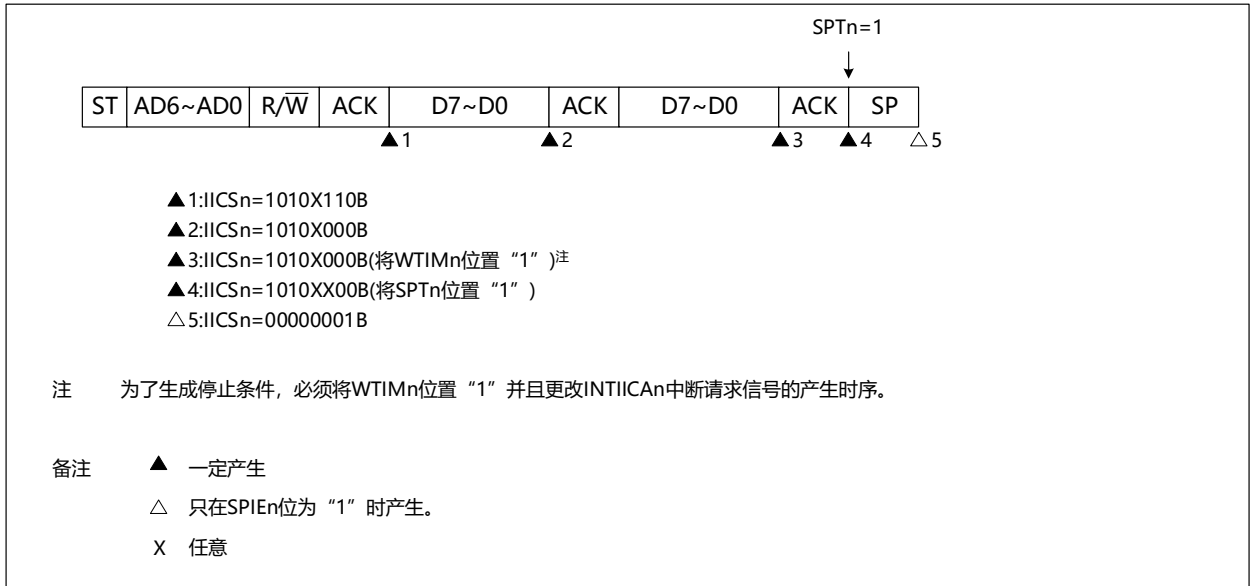
(ii) WTIMn=1的情况



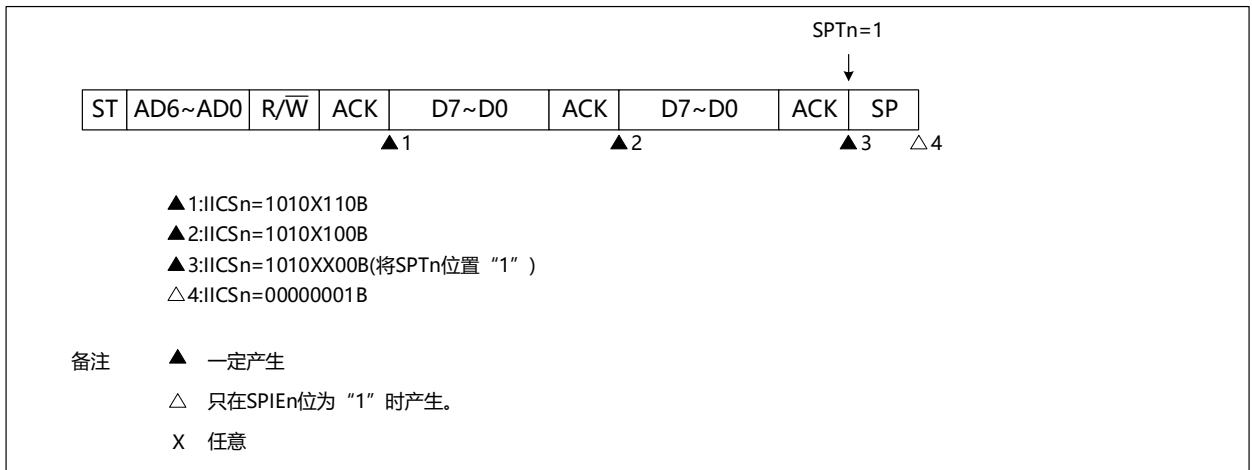
备注 n=0

(c) Start~Code~Data~Data~Stop (发送扩展码)

(i) WTIMn=0的情况

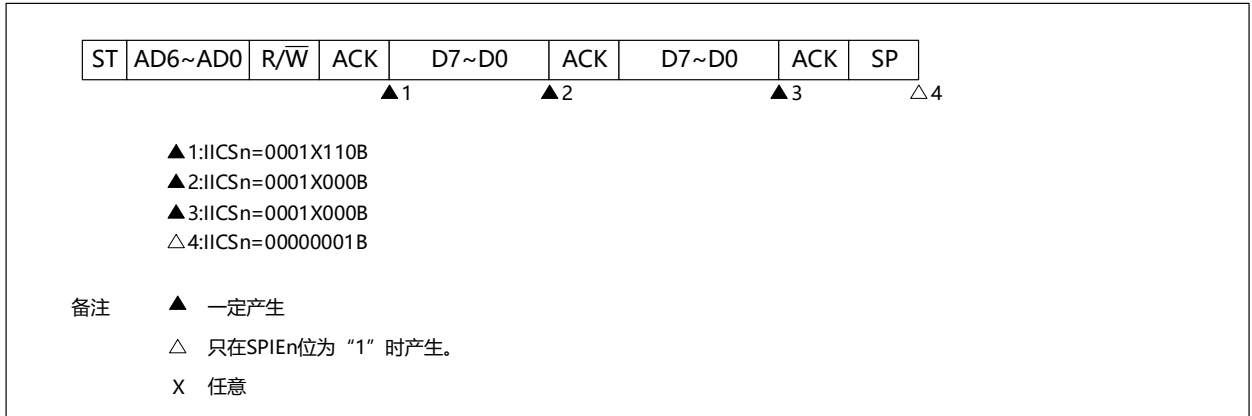


(ii) WTIMn=1的情况

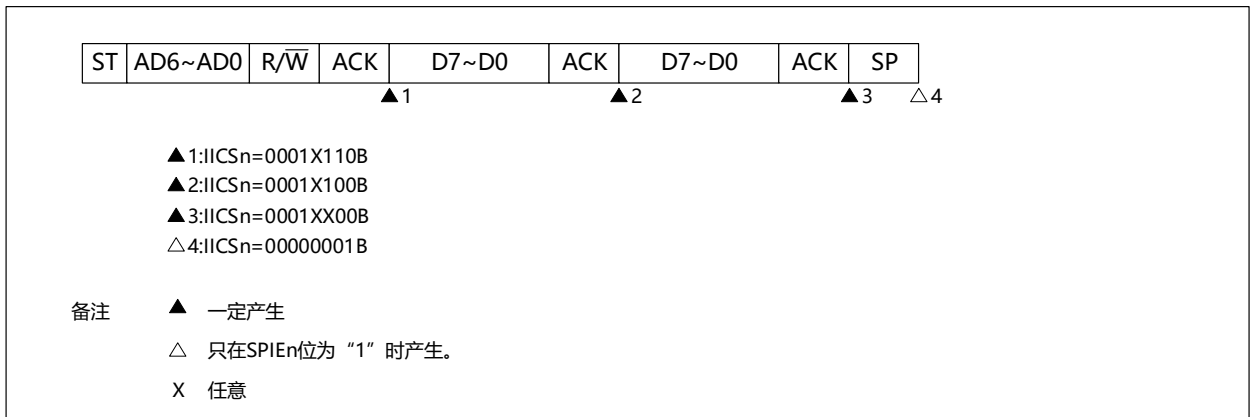


备注 n=0

- (2) 从属运行（接收从属地址的情况）
- (a) Start~Address~Data~Data~Stop
 - (i) WTIMn=0的情况



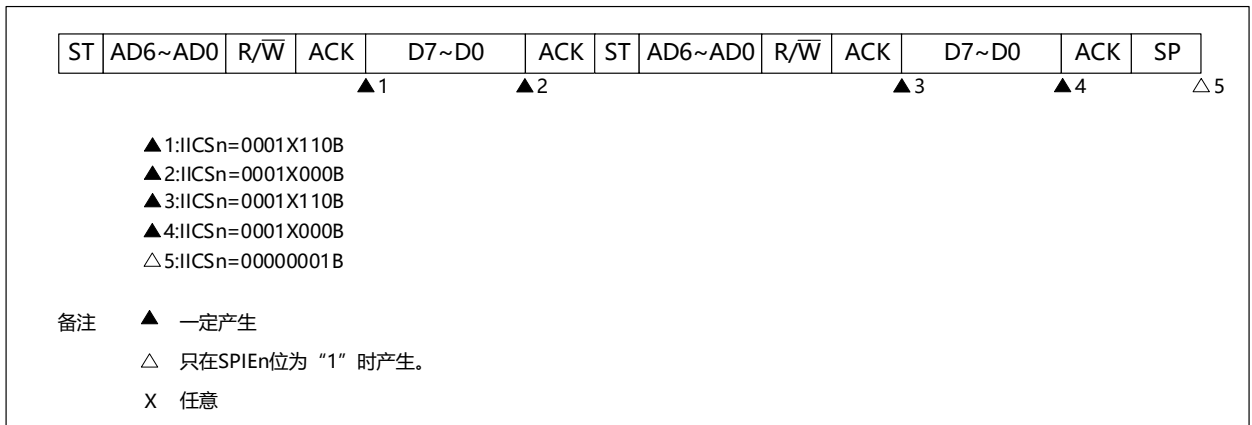
- (ii) WTIMn=1的情况



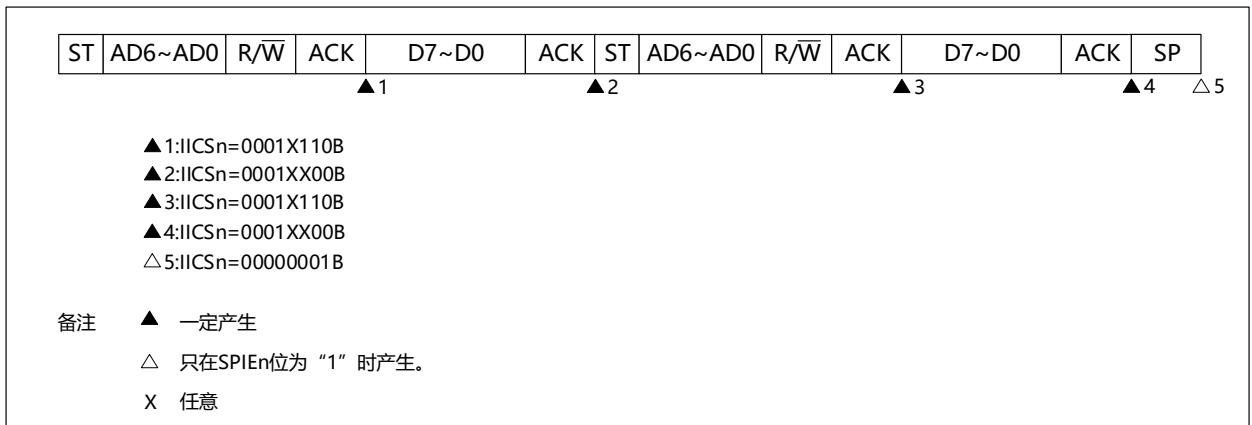
备注 n=0

(b) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）



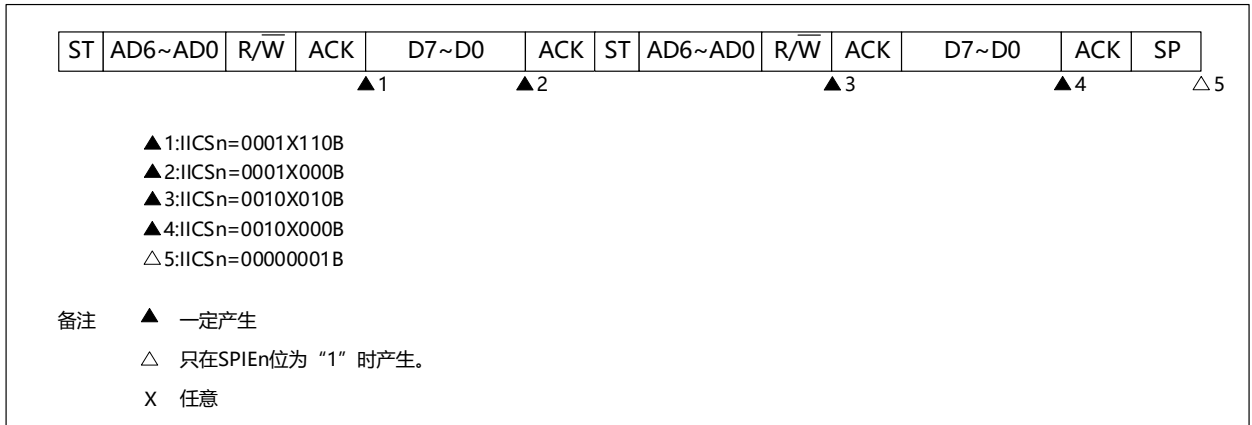
(ii)WTIMn=1的情况（在重新开始后SVAn相同）



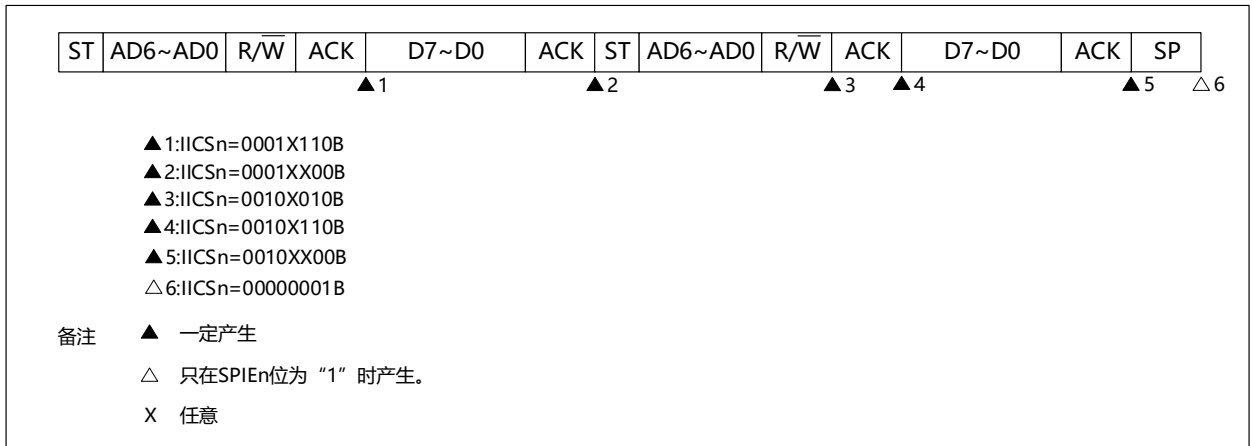
备注 n=0

(c) Start~Address~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（扩展码））



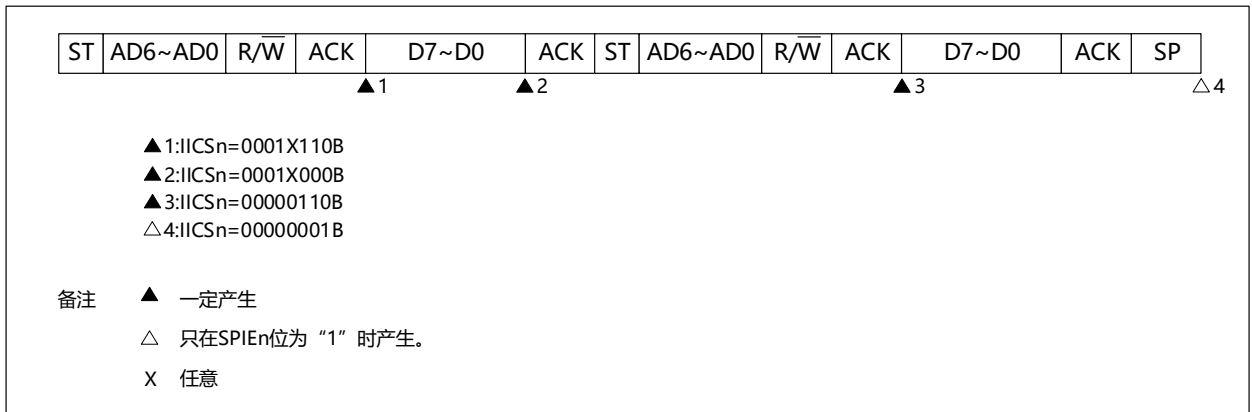
(ii)WTIMn=1的情况（在重新开始后地址不同（扩展码））



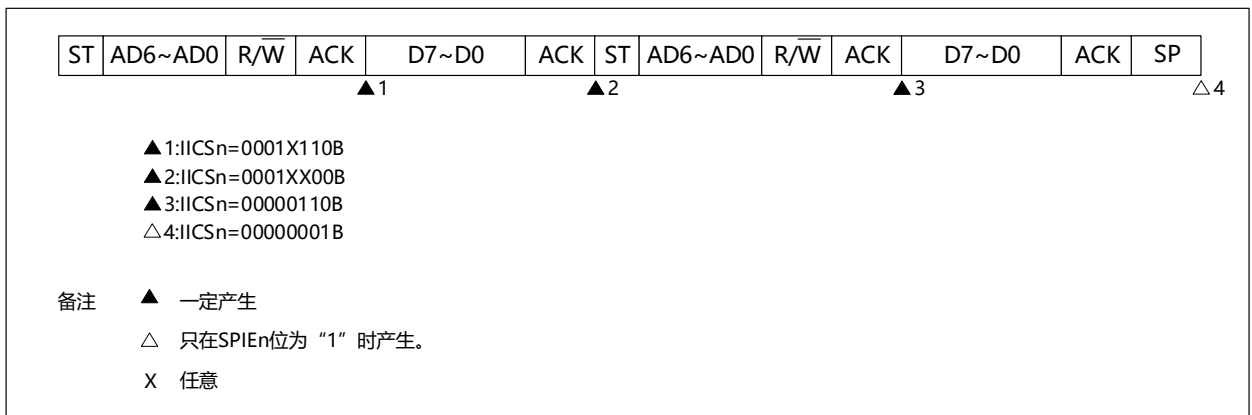
备注 n=0

(d) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（非扩展码））



(ii) WTIMn=1的情况（在重新开始后地址不同（非扩展码））



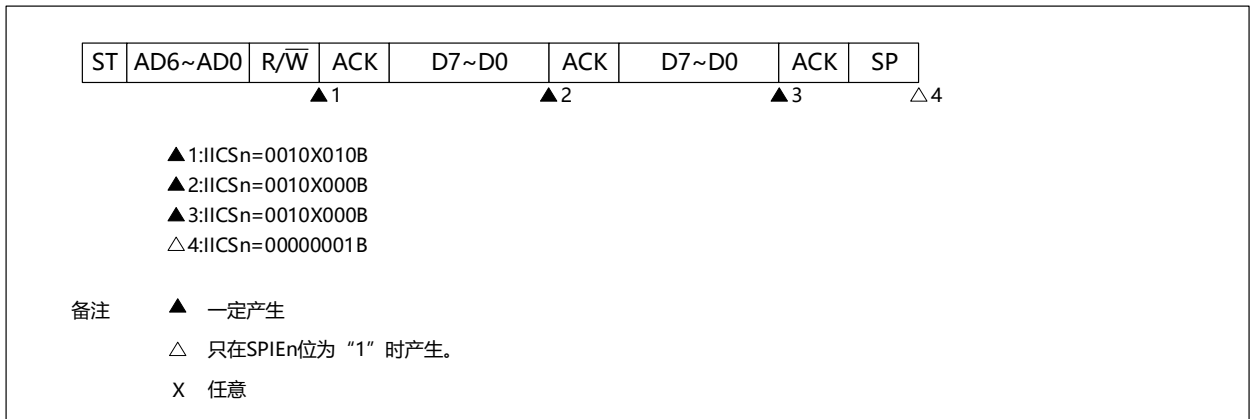
备注 n=0

(3) 从属运行（接收扩展码的情况）

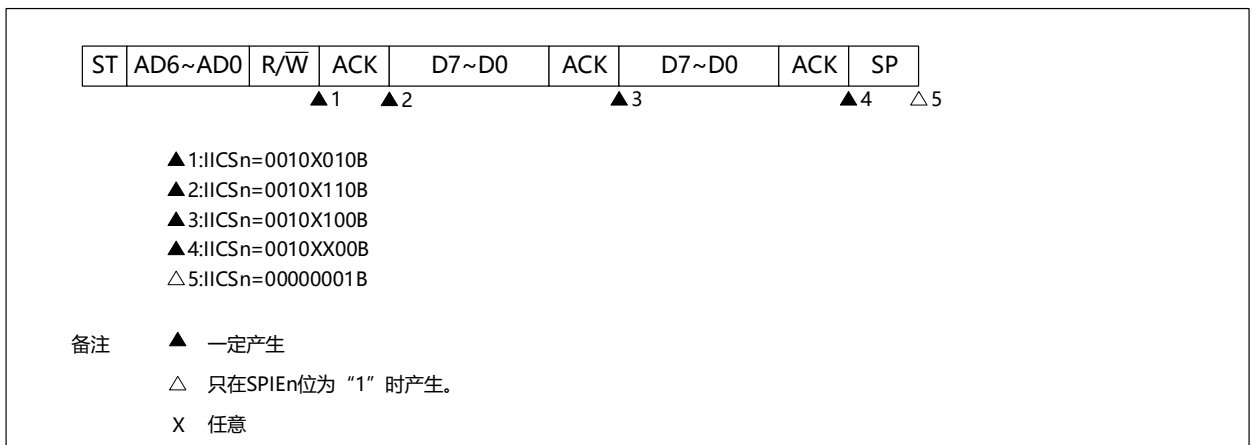
在接收扩展码时，始终参加通信。

(a) Start~Code~Data~Data~Stop

(i) WTIMn=0的情况



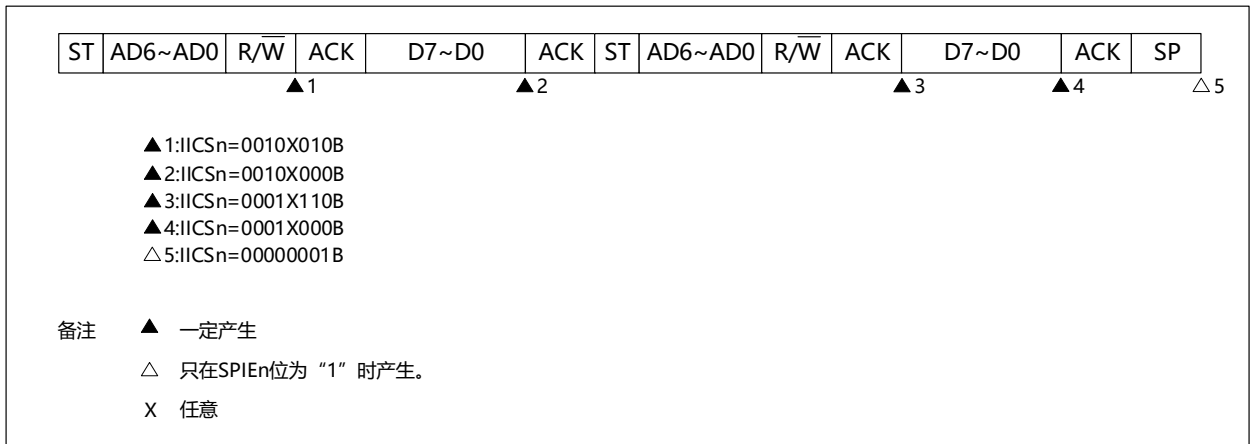
(ii) WTIMn=1的情况



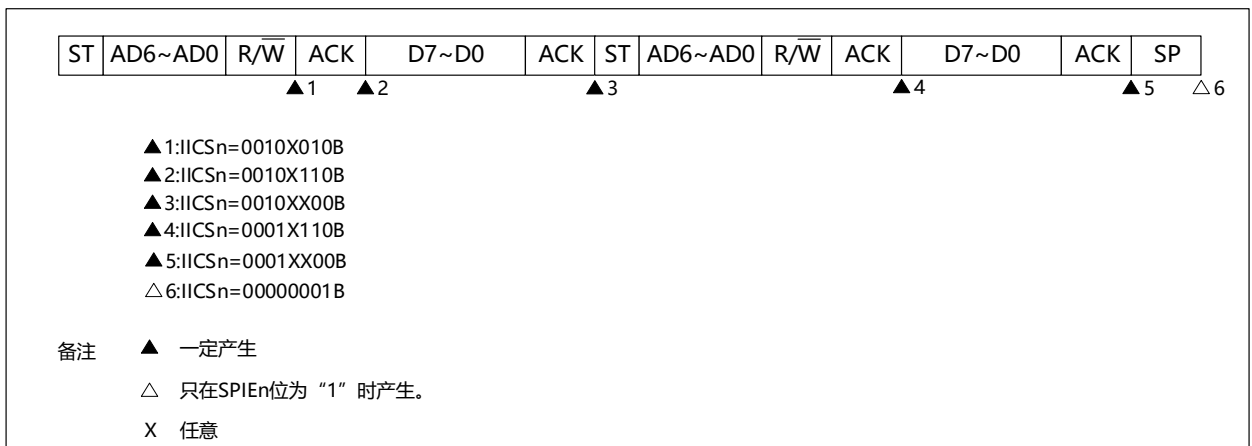
备注 n=0

(b) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）



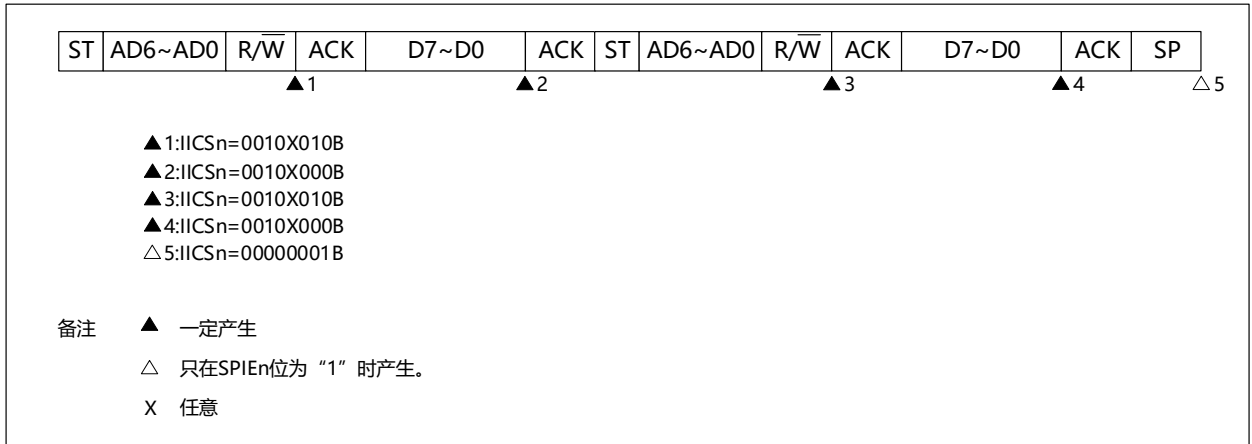
(ii) WTIMn=1的情况（在重新开始后SVAn相同）



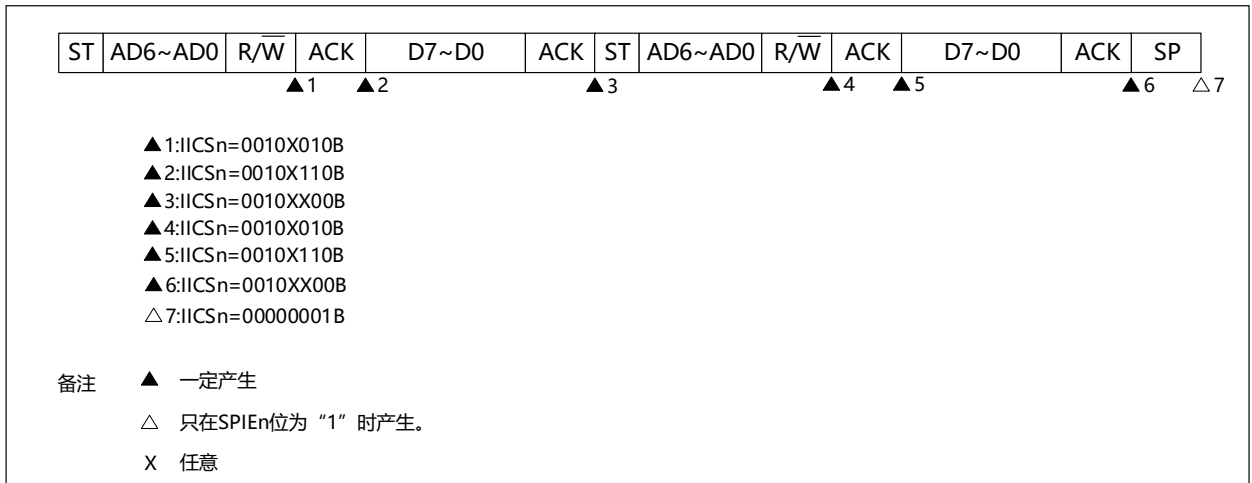
备注 n=0

(c) Start~Code~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况（在重新开始后接收扩展码）



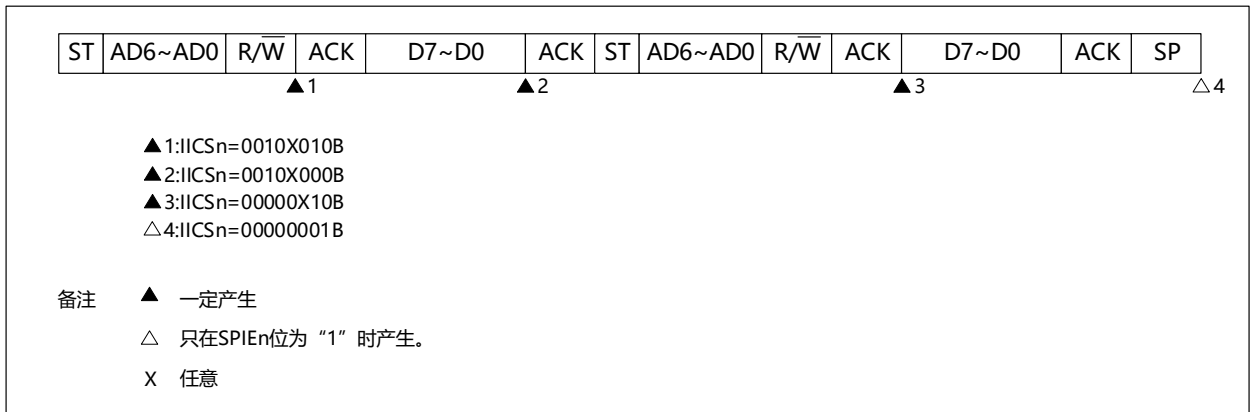
(ii) WTIMn=1的情况（在重新开始后接收扩展码）



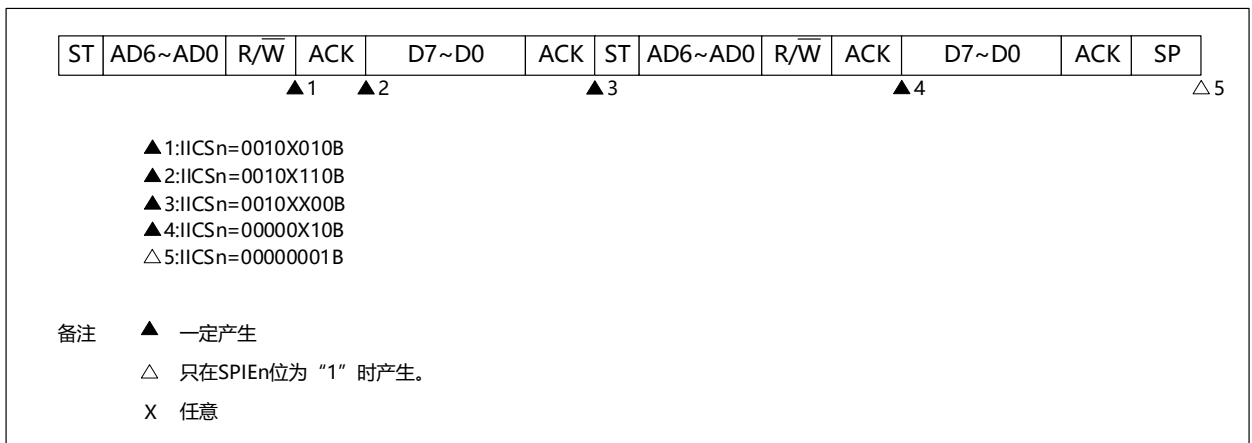
备注 n=0

(d) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（非扩展码））

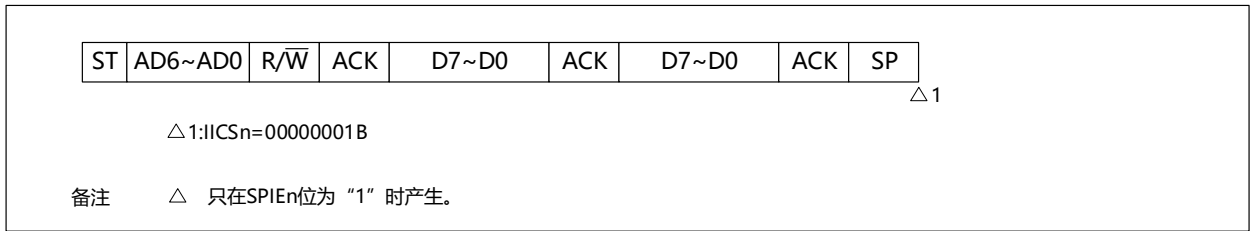


(ii)WTIMn=1的情况（在重新开始后地址不同（非扩展码））



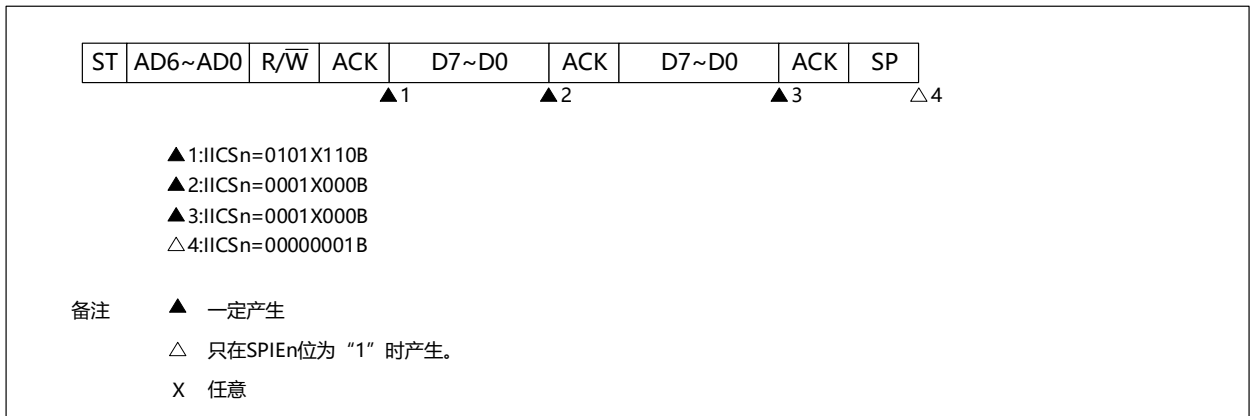
备注 n=0

- (4) 不参加通信的运行
- (a) Start~Code~Data~Data~Stop



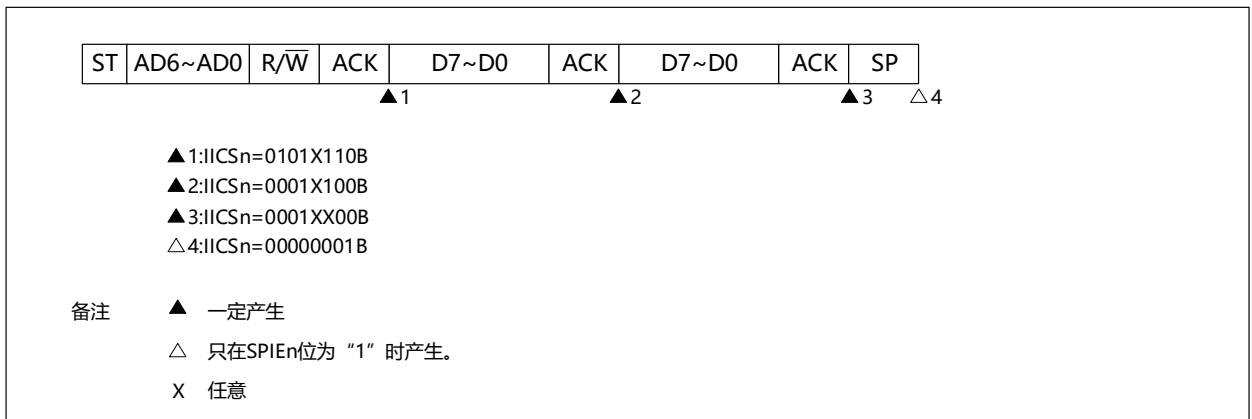
- (5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）
- 在多主控系统中用作主控设备时，必须在每次产生INTIICAn中断请求信号时读MSTS_n位，确认仲裁结果。

- (a) 在发送从属地址数据的过程中仲裁失败的情况
- (i) WTIMn=0的情况



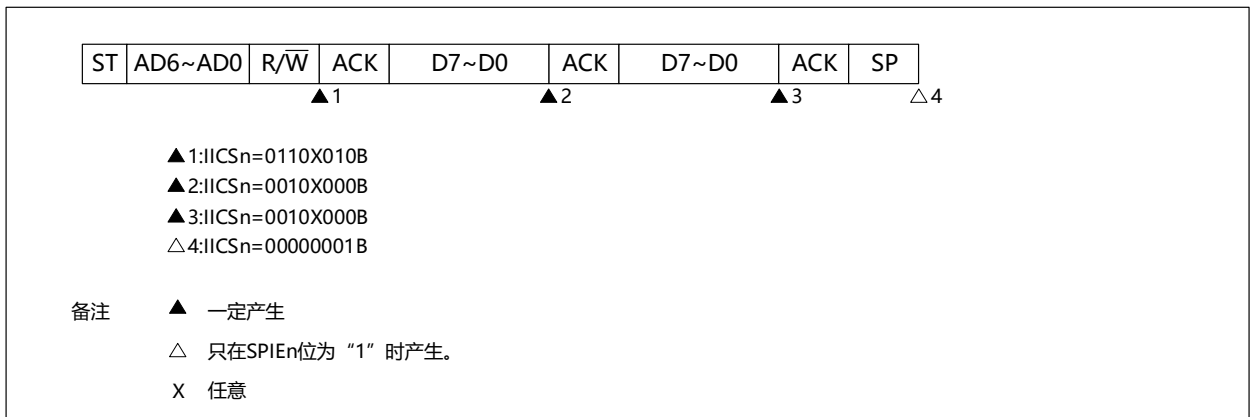
备注 n=0

(ii) WTIMn=1的情况



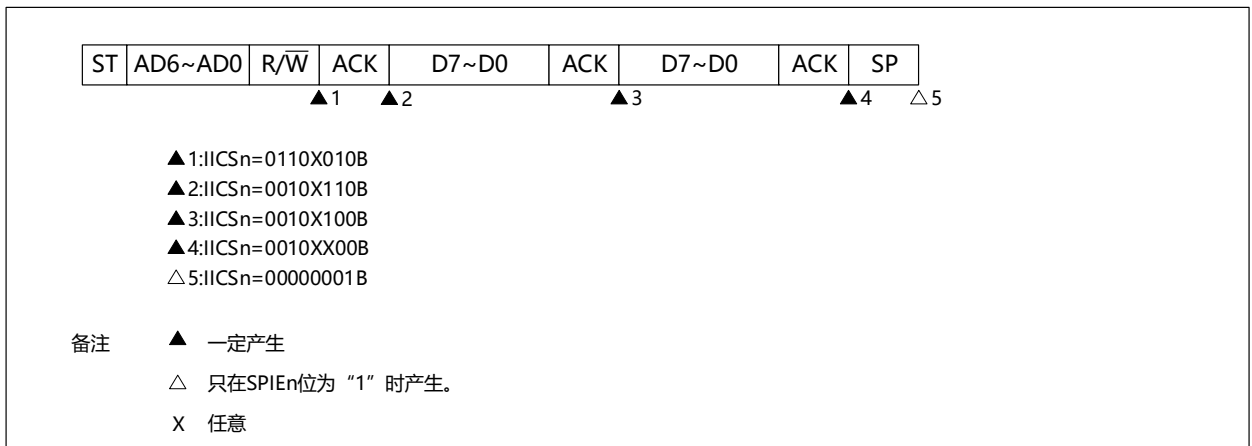
(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIMn=0的情况



备注 n=0

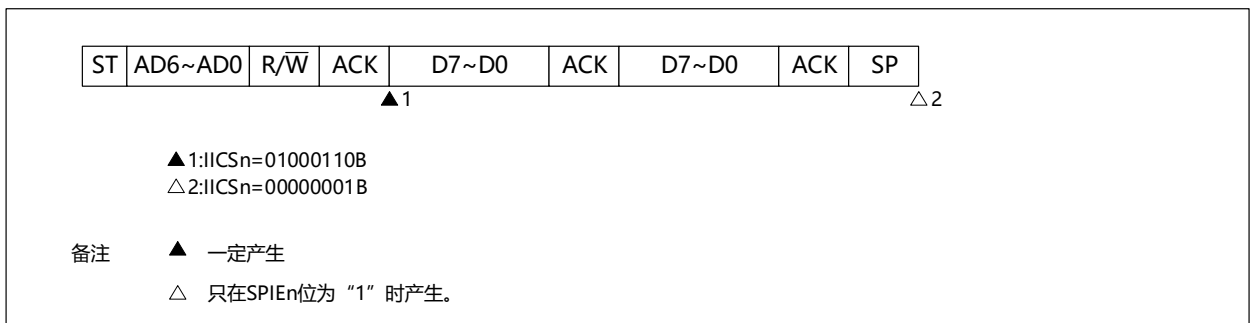
(ii) WTIMn=1的情况



(6) 仲裁失败的运行（在仲裁失败后不参加通信）

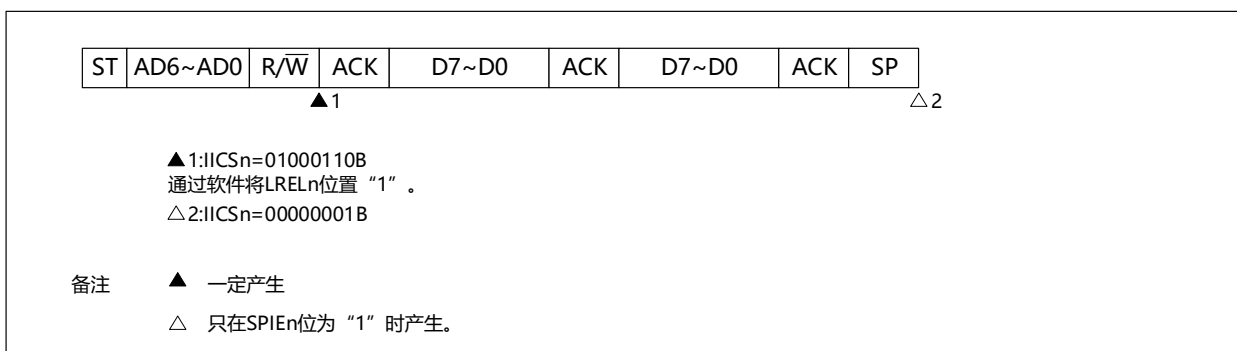
在多主控系统中用作主控设备时，必须在每次产生INTIICAn中断请求信号时读MSTS_n位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）



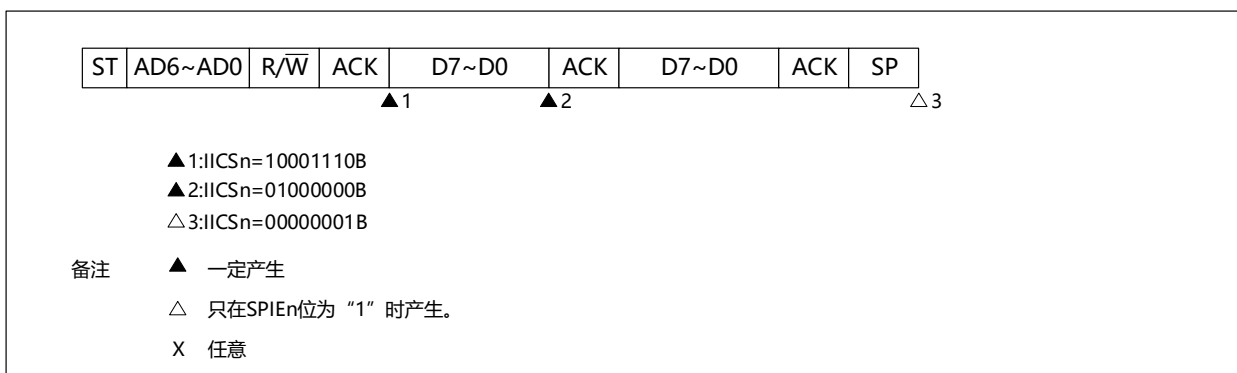
备注 n=0

(b) 在发送扩展码的过程中仲裁失败的情况



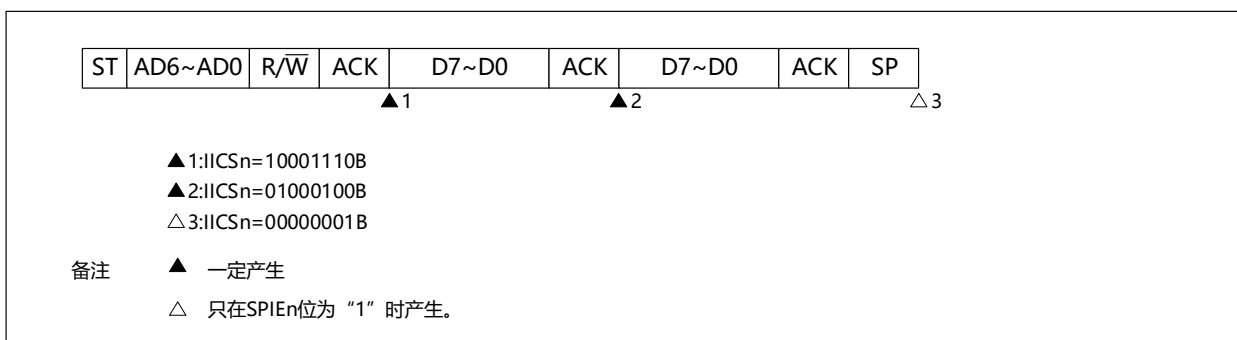
(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0的情况



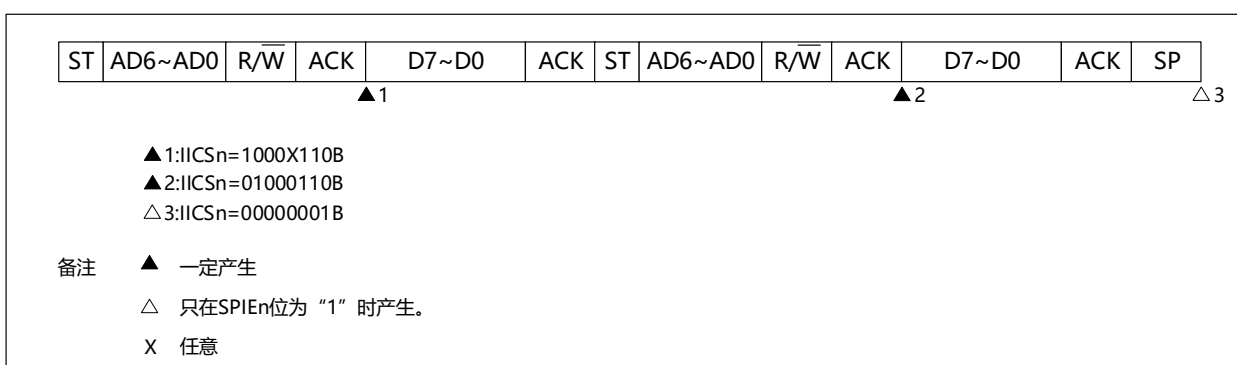
备注 n=0

(ii) WTIMn=1的情况



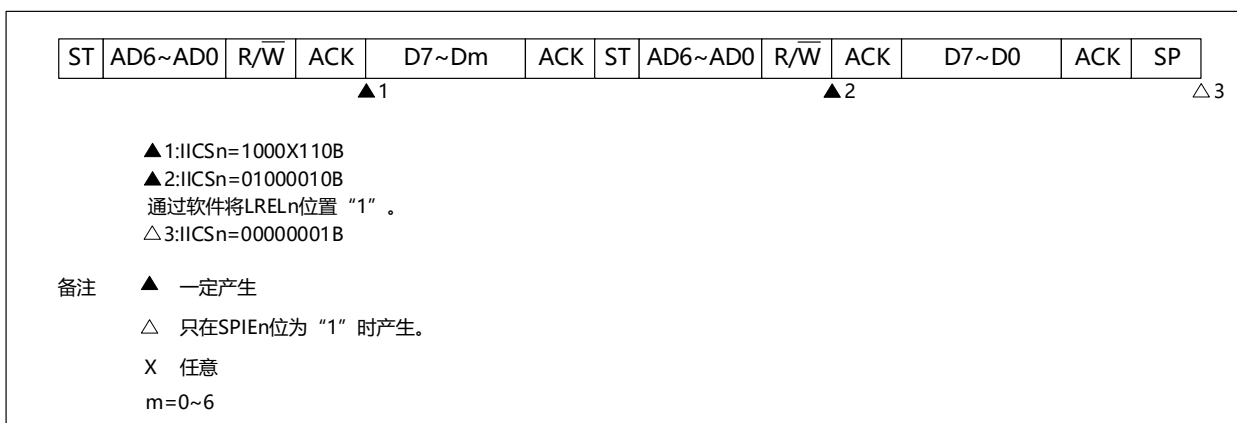
(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码（例如，SVAn不同）

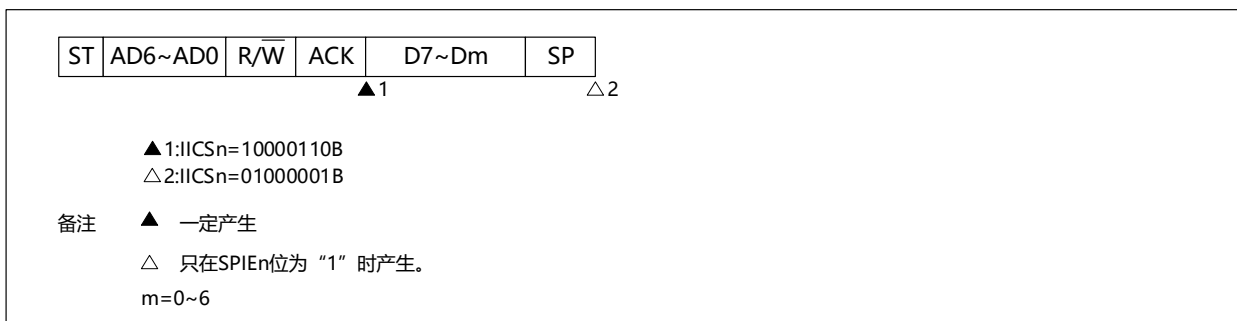


备注 n=0

(ii)扩展码



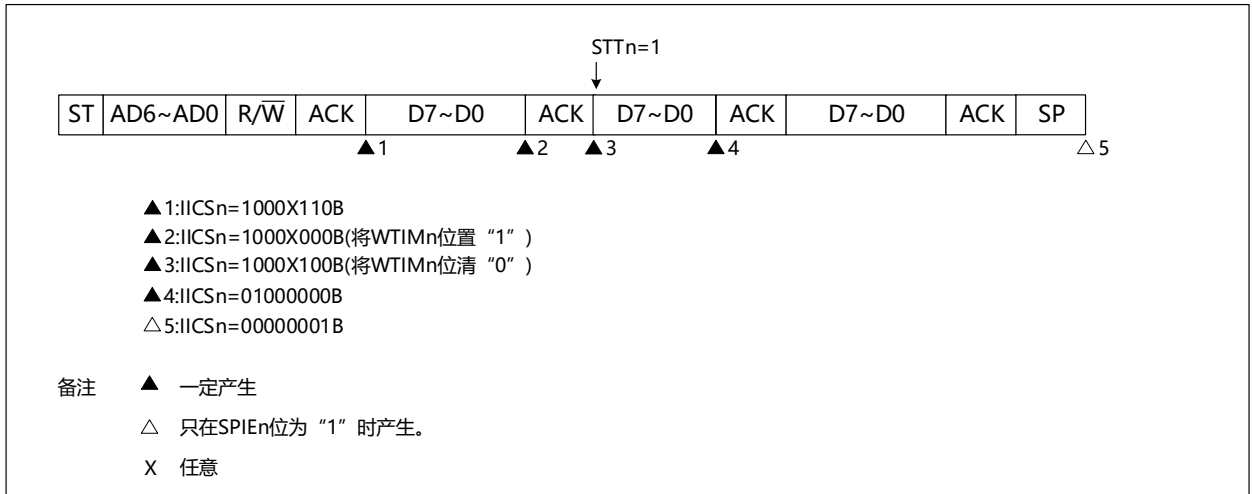
(e) 在传送数据时因停止条件而仲裁失败的情况



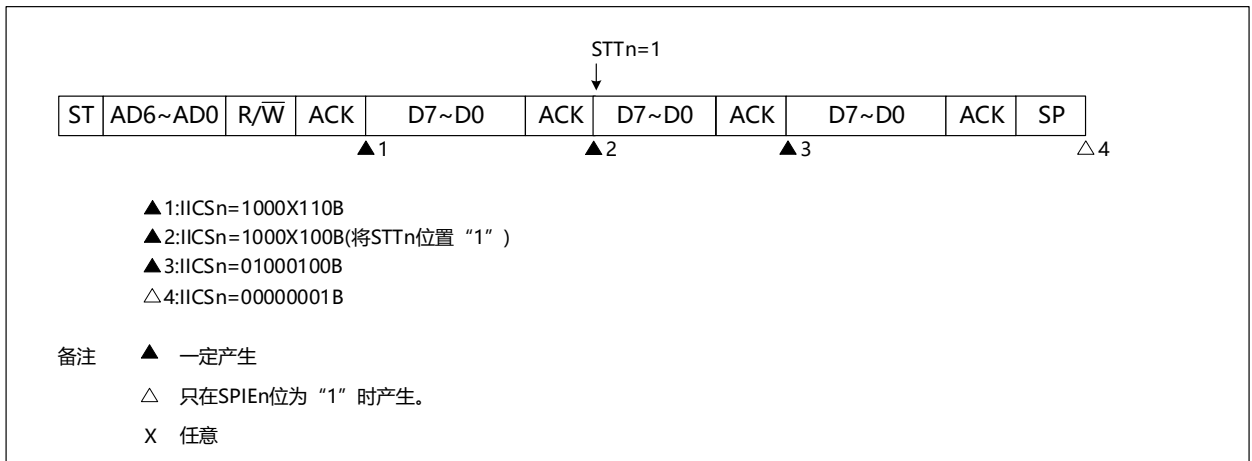
备注 n=0

(f) 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



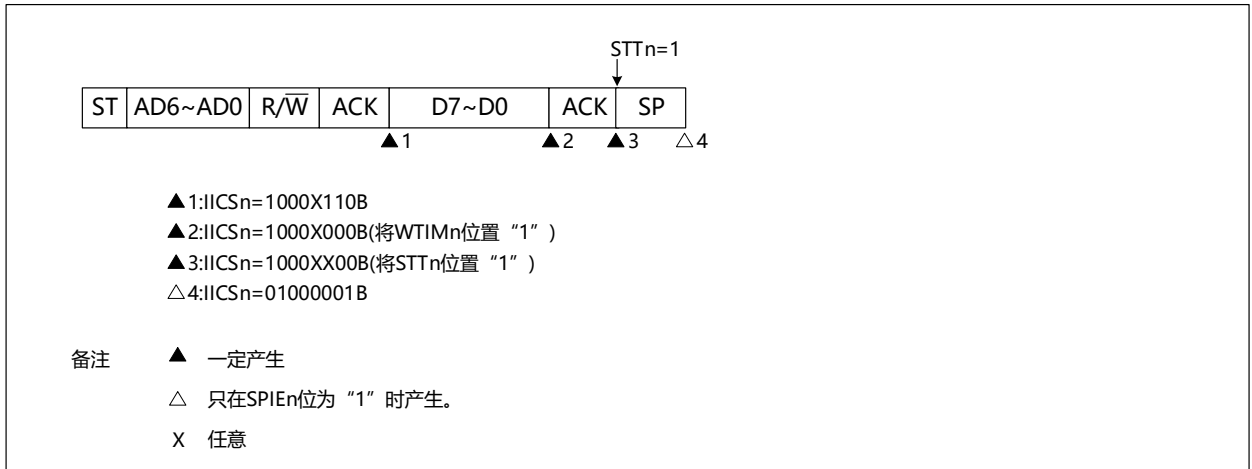
(ii) WTIMn=1的情况



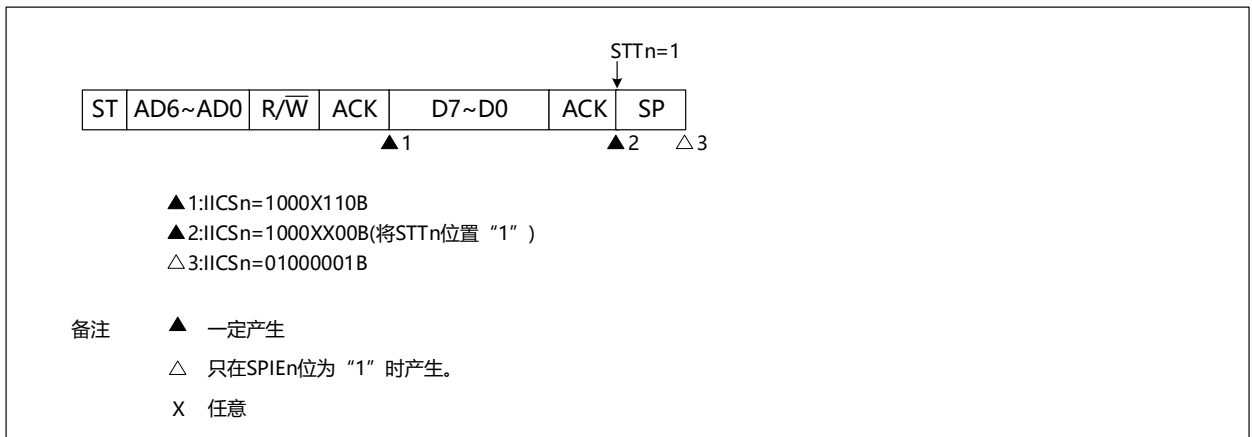
备注 n=0

(g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

(i) WTIMn=0的情况



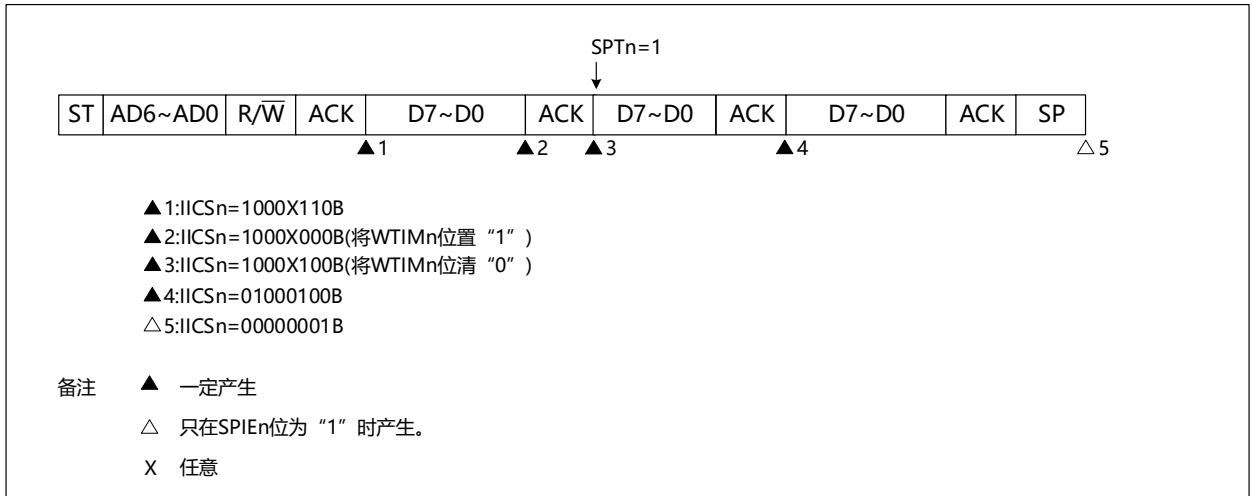
(ii) WTIMn=1的情况



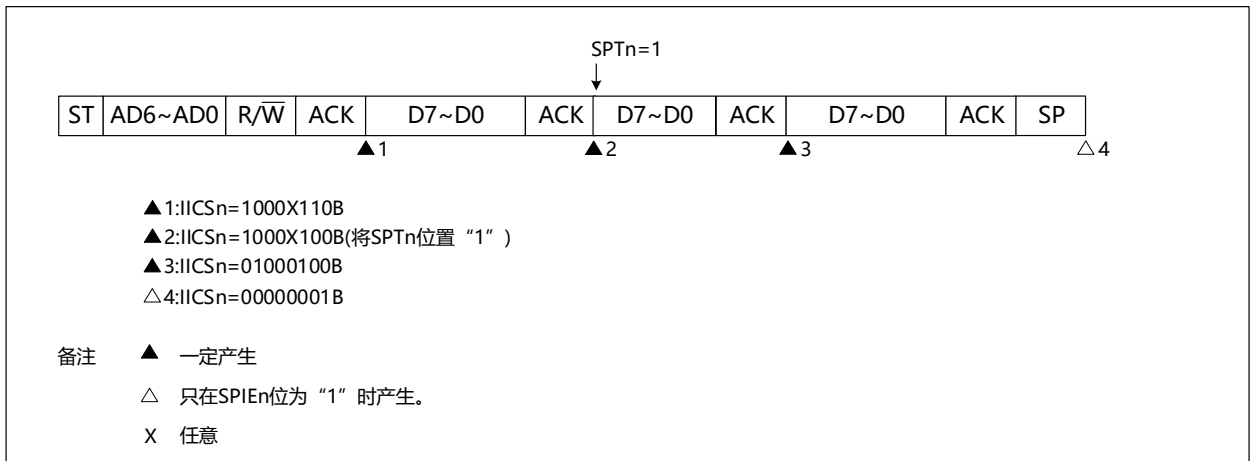
备注 n=0

(h) 在想要生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



(ii) WTIMn=1的情况



备注 n=0

14.6 时序图

在I2C总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。主控设备在从属设备地址之后发送表示数据传送方向的TRCn位（IICA状态寄存器n（IICSn）的bit3），开始与从属设备进行串行通信。数据通信的时序图如图 14-31 主控设备->从属设备的通信例子所示。

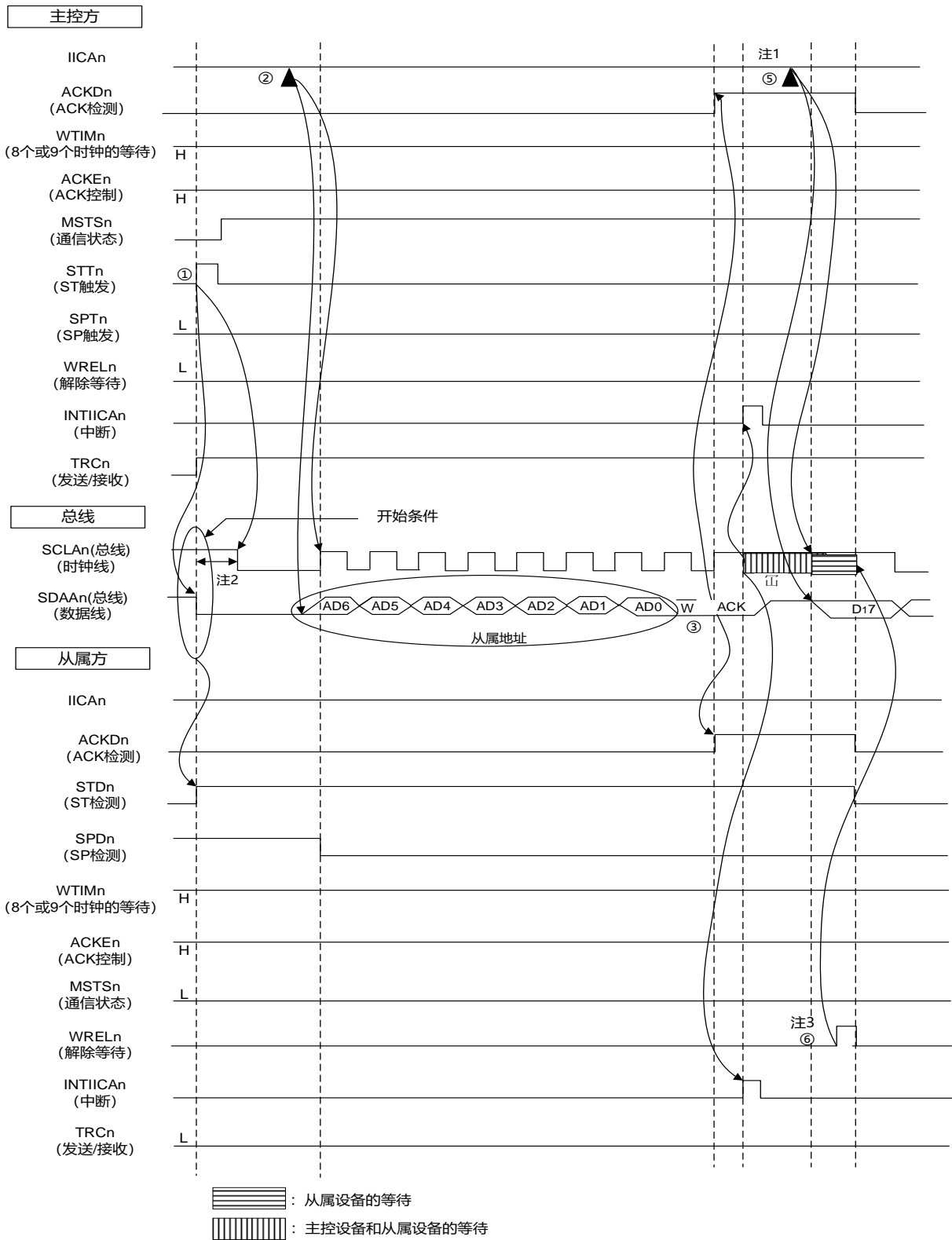
与串行时钟（SCLAn）的下降沿同步进行IICA移位寄存器n（IICAn）的移位，并且将发送数据传送到SO锁存器，以MSB优先从SDAAn引脚输出数据。

在SCLAn的上升沿将SDAAn引脚输入的数据取到IICAn。

备注 n=0

图 14-31 主控设备->从属设备的通信例子
(主控设备: 选择9个时钟的等待, 从属设备: 选择9个时钟的等待) (1/4)

(1) 开始条件~地址~数据



- 注
- 1.要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。
 - 2.从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。
 - 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 14-31 主控设备->从属设备的通信例子的“(1)开始条件~地址~数据”的①~⑥的说明如下：

- ①如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过SCLAn=1使SDAAn从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTSn=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。
- ②如果主控方给IICA移位寄存器n（IICAn）写地址+W（发送），就发送从属地址。
- ③在从属方，如果接收地址和本地站地址（SVAn的值）相同注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。
- ④主控方在第9个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤主控方给IICAn寄存器写发送数据，解除主控方的等待。
- ⑥如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

- 注
- 如果发送的地址和从属地址不同，从属方就不给主控方返回ACK（NACK：SDAAn=1），并且不产生INTIICAn中断（地址匹配中断），也不进入等待状态。
- 但是，主控方对于ACK和NACK都产生INTIICAn中断（地址发送结束中断）。

备注1. 图14-31的①~⑯是通过I²C总线进行数据通信的一系列运行步骤。

图14-31的“(1)开始条件~地址~数据”说明步骤①~⑥。

图14-31的“(2)地址~数据~数据”说明步骤③~⑩。

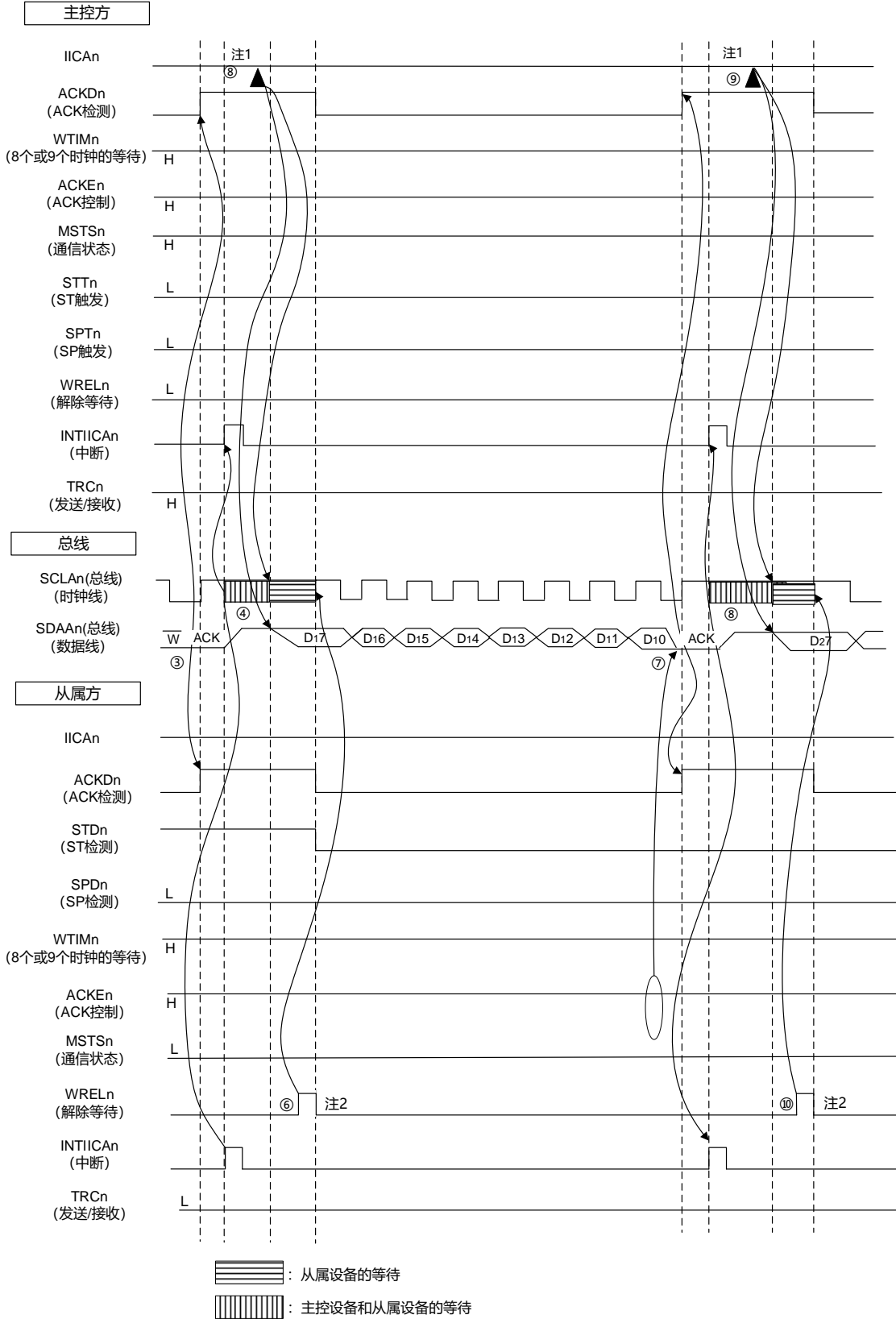
图14-31的“(3)数据~数据~停止条件”说明步骤⑦~⑯。

2.n=0

图14-31 主控设备->从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (2/4)

(2) 地址~数据~数据



- 注 1.要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。
 2.要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图14-31的“(2)地址~数据~数据”的③~⑩的说明如下：

- ③在从属方，如果接收地址和本地站地址（SVAn的值）相同^注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。
- ④主控方在第9个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）^注。
- ⑤主控方给IICA移位寄存器n（IICAn）写发送数据，解除主控方的等待。
- ⑥如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。
- ⑦在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。
- ⑧主控方和从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。
- ⑨主控方给IICAn寄存器写发送数据，解除主控方的等待。
- ⑩如果从属方读接收数据并且解除等待（WRELn=1），主控方就开始给从属方传送数据。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回ACK（NACK：SDAAn=1），并且不产生INTIICAn中断（地址匹配中断），也不进入等待状态。

但是，主控方对于ACK和NACK都产生INTIICAn中断（地址发送结束中断）。

备注1. 图14-31的①~⑮是通过I²C总线进行数据通信的一系列运行步骤。

图14-31的“(1)开始条件~地址~数据”说明步骤①~⑥。

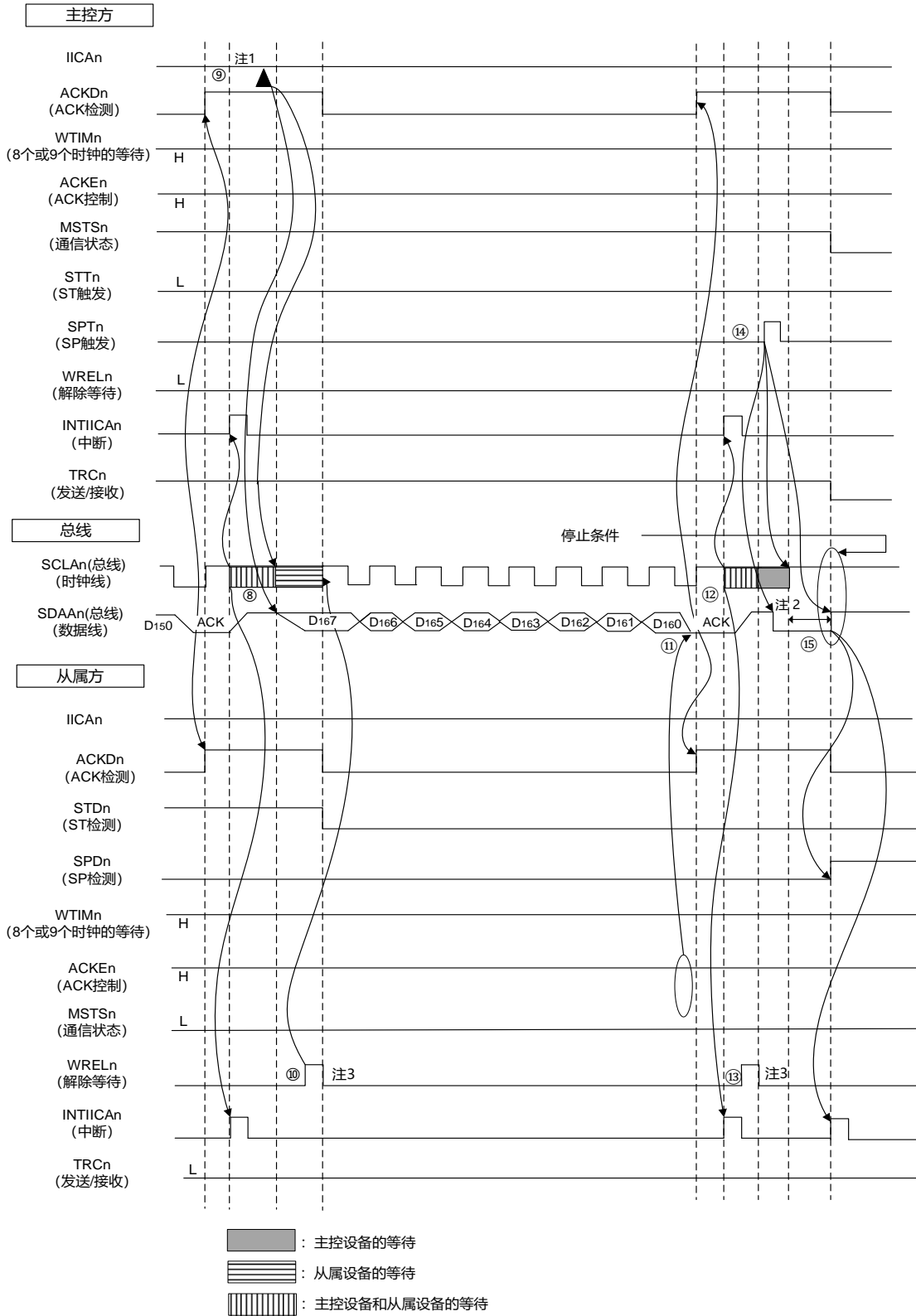
图14-31的“(2)地址~数据~数据”说明步骤③~⑩。

图14-31的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

2.n=0

图14-31 主控设备->从属设备的通信例子
(主控设备: 选择9个时钟的等待, 从属设备: 选择9个时钟的等待) (3/4)

(2) 数据~数据~停止条件



- 注 1.要解除主控方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。
 2.在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的时间, 在设置为标准模式时至少为4.0us, 在设置为快速模式时至少为0.6us。
 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 14-31 主控设备->从属设备的通信例子的“(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14-31的①~⑮是通过I²C总线进行数据通信的一系列运行步骤。

图14-31的“(1)开始条件~地址~数据”说明步骤①~⑥。

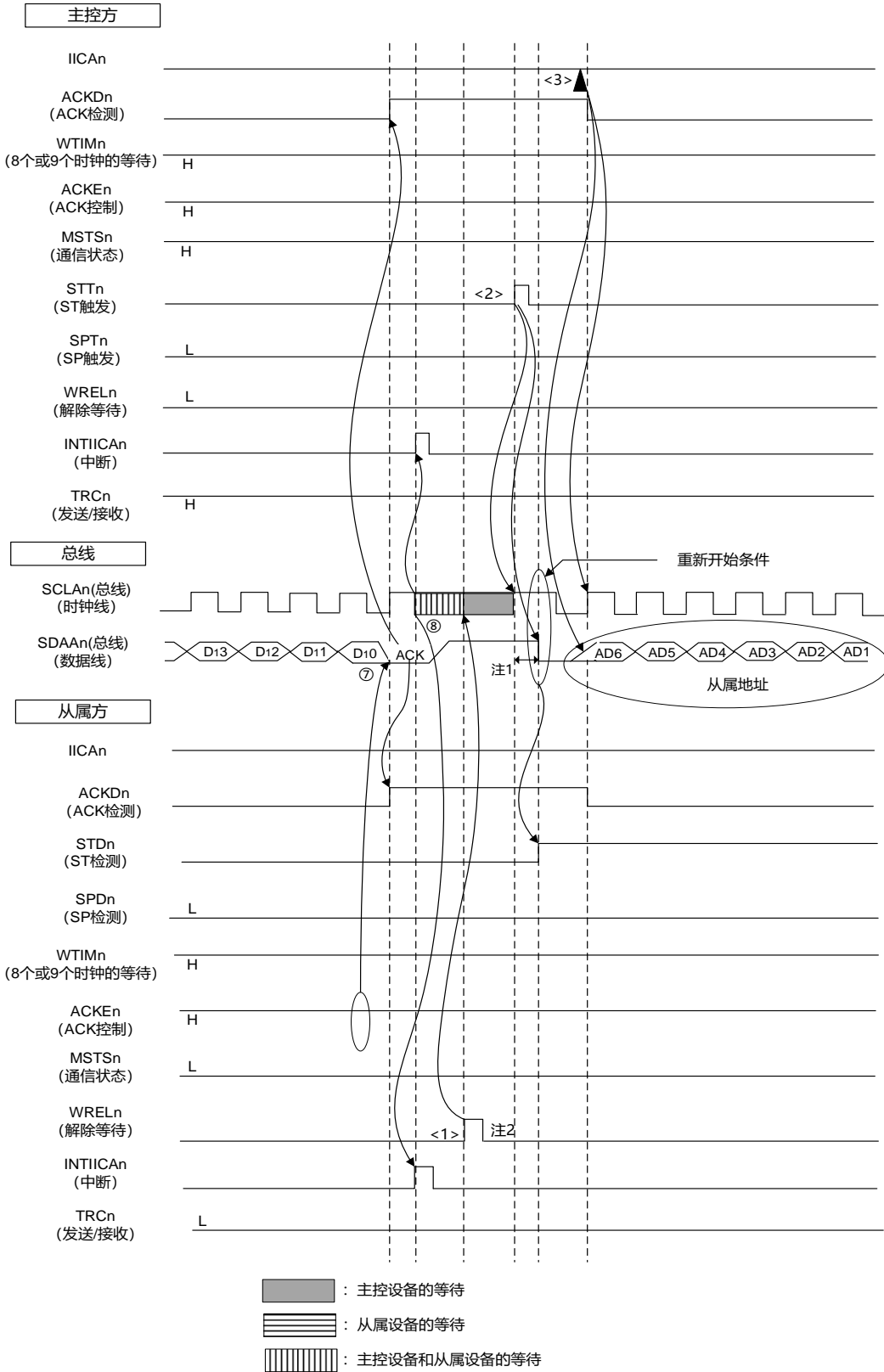
图14-31的“(2)地址~数据~数据”说明步骤③~⑩。

图14-31的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

2.n=0

图14-31 主控设备->从属设备的通信例子
(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (4/4)

(4) 数据~重新开始条件~地址



注 1.在发行重新开始条件后，从SCLAn引脚信号上升到生成开始条件的的时间，在设置为标准模式时至少为4.7us，在设置为快速模式时至少为0.6us。

2.要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

图 14-31 主控设备->从属设备的通信例子的“(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。

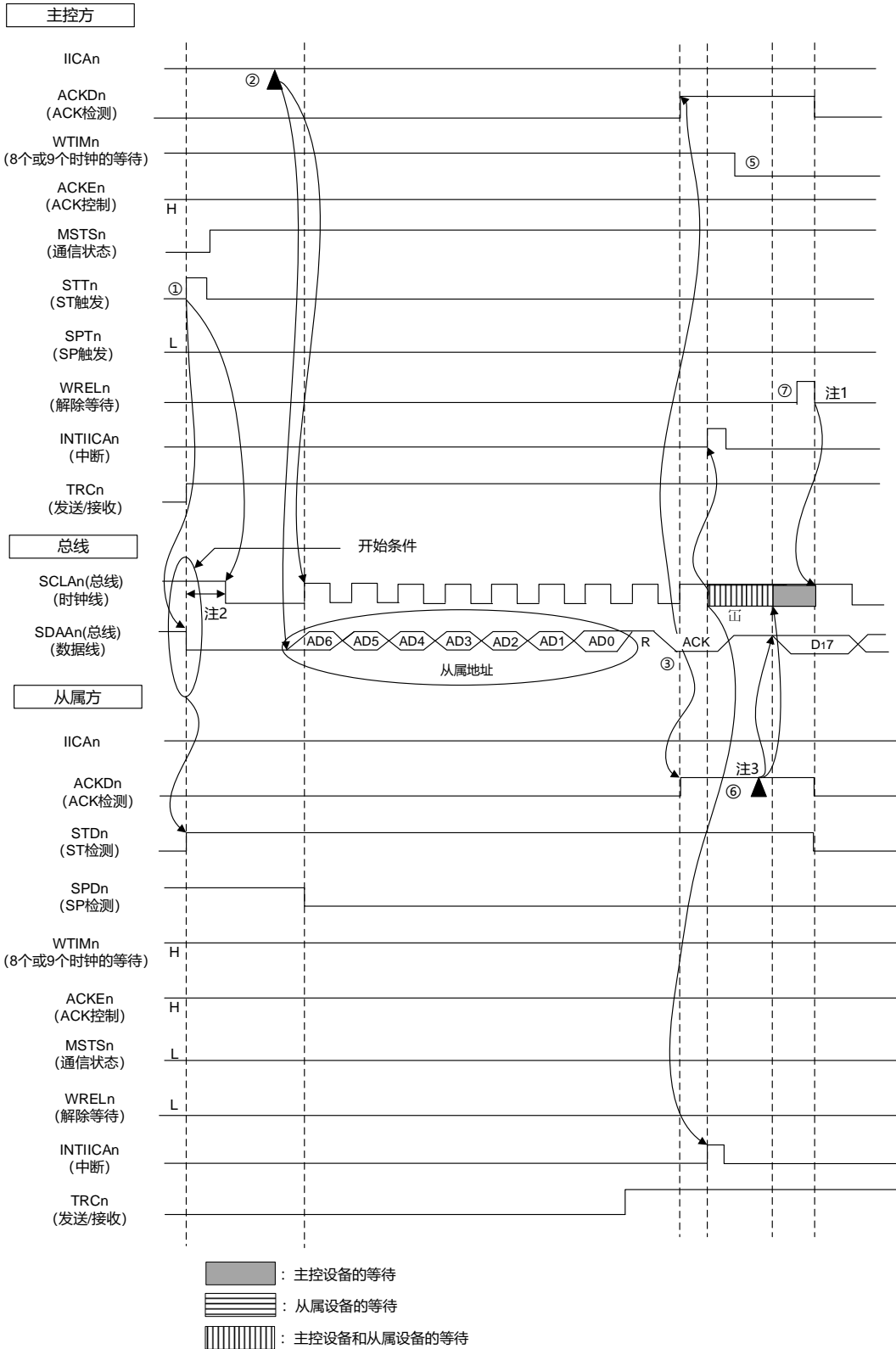
<3>如果主控方给IICA移位寄存器n (IICAn)写地址+R/W (发送),就发送从属地址。

备注 n=0

图14-32 从属设备->主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (1/3)

(1) 开始条件~地址~数据



- 注 1.要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。
 2.从SDAAn引脚信号下降到SCLAn引脚信号下降的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。
 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14-32的“(1)开始条件~地址~数据”的①~⑦的说明如下：

- ①如果在主控方将开始条件触发置位 (STTn=1)，总线数据线 (SDAAn) 就下降，生成开始条件 (通过SCLAn=1使SDAAn从“1”变为“0”)。此后，如果检测到开始条件，主控方就进入主控通信状态 (MSTSn=1)，在经过保持时间后总线时钟线下降 (SCLAn=0)，结束通信准备。
- ②如果主控方给IICA移位寄存器n (IICAn) 写地址+R (接收)，就发送从属地址。
- ③在从属方，如果接收地址和本地站地址 (SVAn的值) 相同注，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ④主控方在第9个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn: 地址匹配中断) 注。
- ⑤主控方将等待时序改为第8个时钟 (WTIMn=0)。
- ⑥从属方给IICAn寄存器写发送数据，解除从属方的等待。
- ⑦主控方解除等待 (WRELn=1)，开始来自从属设备的数据传送。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回ACK (NACK: SDAAn=1)，并且不产生INTIICAn中断 (地址匹配中断)，也不进入等待状态。
但是，主控方对于ACK和NACK都产生INTIICAn中断 (地址发送结束中断)。

备注1.图14-32的①~⑱是通过I²C总线进行数据通信的一系列运行步骤。

图14-32的“(1)开始条件~地址~数据”说明步骤①~⑦。

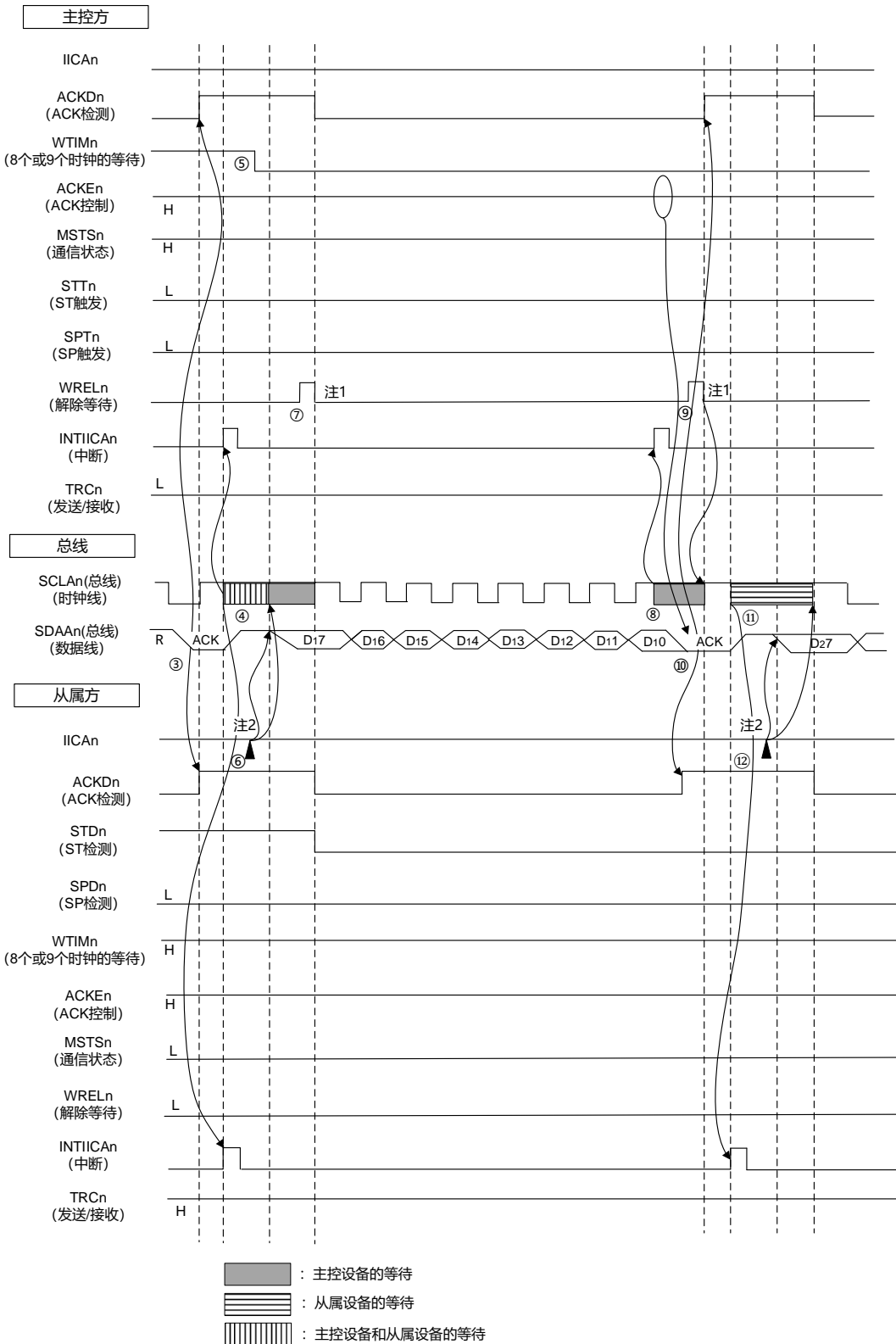
图14-32的“(2)地址~数据~数据”说明步骤③~⑫。

图14-32的“(3)数据~数据~停止条件”说明步骤⑧~⑱。

2.n=0

图14-32 从属设备->主控设备的通信例子
(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (2/3)

(2) 地址~数据~数据



- 注 1.要解除主控方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。
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14-32的“(2)地址~数据~数据”的③~⑫的说明如下:

- ③ 在从属方, 如果接收地址和本地站地址 (SVAn的值) 相同注, 就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ④ 主控方在第9个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断) 注。
- ⑤ 主控方将等待时序改为第8个时钟 (WTIMn=0)。
- ⑥ 从属方给IICA移位寄存器n (IICAn) 写发送数据, 解除从属方的等待。
- ⑦ 主控方解除等待 (WRELn=1), 开始来自从属设备的数据传送。
- ⑧ 主控方在第8个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的ACKEn位为“1”, 所以通过硬件给从属方发送ACK。
- ⑨ 主控方读接收数据, 解除等待 (WRELn=1)。
- ⑩ 从属方在第9个时钟的上升沿检测到ACK (ACKDn=1)。
- ⑪ 从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。
- ⑫ 如果从属方给IICAn寄存器写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。

注 如果发送的地址和从属地址不同, 从属方就不给主控方返回ACK (NACK: 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14-32的①~⑰是通过I²C总线进行数据通信的一系列运行步骤。

图14-32的“(1)开始条件~地址~数据”说明步骤①~⑦。

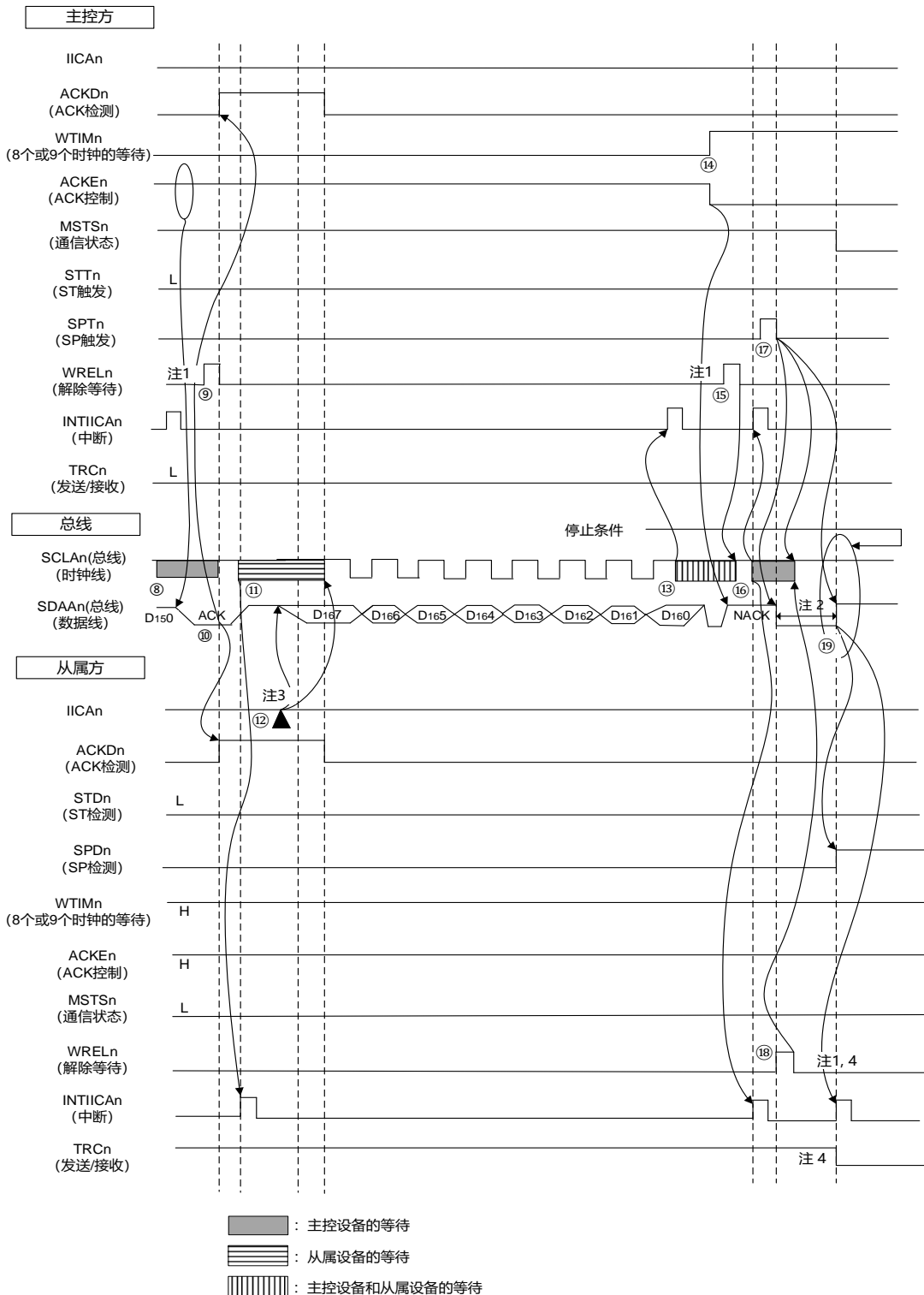
图14-32的“(2)地址~数据~数据”说明步骤③~⑫。

图14-32的“(3)数据~数据~停止条件”说明步骤⑧~⑰。

2.n=0

图14-32 从属设备->主控设备的通信例子
(主控设备：选择8个→9个时钟的等待，从属设备：选择9个时钟的等待) (3/3)

(3) 数据~数据~停止条件



- 注
- 1.要解除等待时，必须将IICAn置“FFH”或者将WRELn位置位。
 - 2.在发行停止条件后，从SCLAn引脚信号上升到生成停止条件的的时间，在设置为标准模式时至少为4.0us，在设置为快速模式时至少为0.6us。
 - 3.要解除从属方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。
 - 4.在从属方的发送期间，如果通过WRELn位的置位来解除等待，就清除TRCn位。

图14-32的“(3)数据~数据~停止条件”的⑧~⑱的说明如下：

- ①. 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。
- ②. 主控方读接收数据，解除等待（ $WRELn=1$ ）。
- ③. 从属方在第9个时钟的上升沿检测到ACK（ $ACKDn=1$ ）。
- ④. 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）。
- ⑤. 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。
- ⑥. 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（ $SDAAn=0$ ）。
- ⑦. 主控方设置为NACK应答（ $ACKEn=0$ ），并且将等待时序改为第9个时钟（ $WTIMn=1$ ）。如果主控方解除等待（ $WRELn=1$ ），从属方就在第9个时钟的上升沿检测到NACK（ $ACKDn=0$ ）。
- 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. 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。
- ⑩. 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（ $SCLAn=1$ ）。
- ⑪. 如果主控方确认到总线时钟线被置位（ $SCLAn=1$ ），就在经过停止条件准备时间后将总线数据线置位（ $SDAAn=1$ ），然后发行停止条件（通过 $SCLAn=1$ 使 $SDAAn$ 从“0”变为“1”）。如果生成停止条件，从属方就检测到停止条件，并且产生中断（INTIICAn：停止条件中断）。
- ⑫. （ $SDAAn=1$ ），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 IrDA

IrDA通过与通用串行通信单元（SCI）合作，实现发送和接收符合IrDA（InfraredDataAssociation（红外线数据协会））1.0协议的IrDA通信波形。

15.1 IrDA的功能

如果通过IRCR寄存器的IRE位将IrDA功能置为有效，SCI的TxD2信号和RxD2信号就能对符合IrDA1.0协议的波形进行编码或者解码（IrTxD/IrRxD引脚），之后通过连接红外线发送/接收的发送器或者接收器，实现支持IrDA1.0协议的红外线发送和接收。

在支持IrDA1.0协议的系统中，以9600bps的传送速率开始通信后，可根据需要改变传送速率。IrDA没有内置自动改变传送速率的功能，所以必须通过软件更改设置以改变传送速率。

在选择高速内部振荡器（ $f_{IH}=24、12、6、3\text{MHz}$ ）时，能设置以下的波特率。

- 115.2kbps、57.6kbps、38.4kbps、19.2kbps、9600bps、2400bps

IrDA与SCI的合作示意框图如图15-1所示。

图15-1 IrDA与SCI的合作示意框图

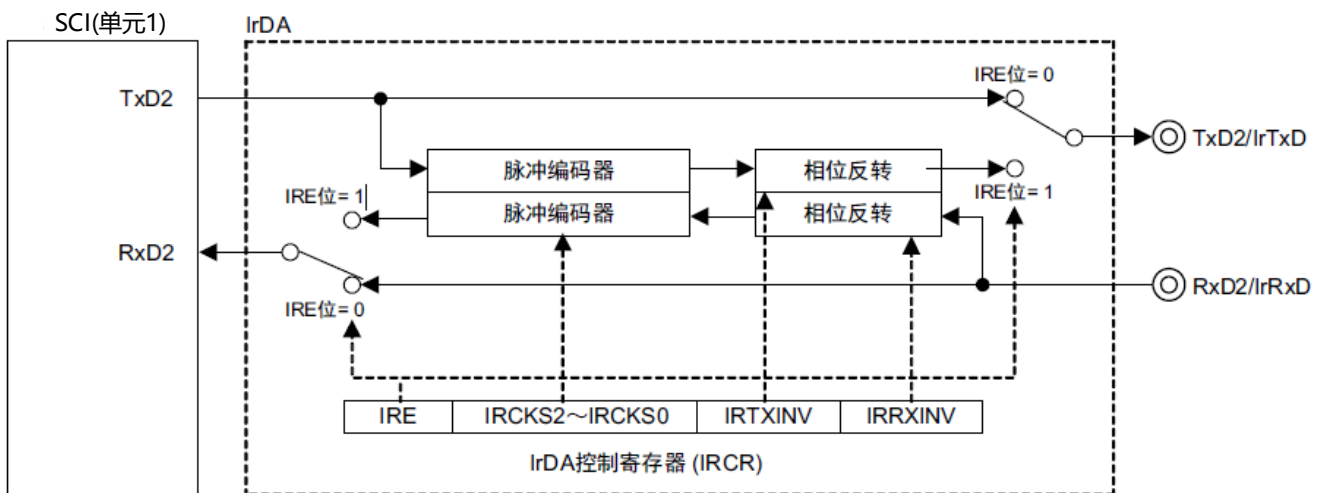


表15-1 IrDA的引脚结构

引脚名称	输入/输出	功能
IrTxD	输出	发送数据的输出引脚
IrRxD	输入	接收数据的输入引脚

15.2 控制IrDA的寄存器

通过以下寄存器控制IrDA功能。

- 外围允许寄存器0 (PER0)
- IrDA 控制寄存器 (IRCR)

15.2.1 外围允许寄存器0 (PER0)

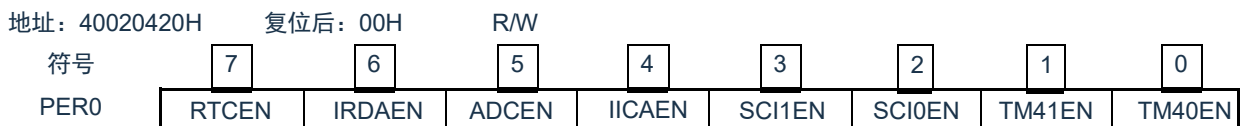
PER0寄存器是设置允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用IrDA时，必须将bit6 (IRDAEN) 置“1”。

通过8位存储器操作指令设置PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图15-2 外围允许寄存器0 (PER0) 的格式



IRDAEN	IrDA的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写 IrDA 使用的 SFR。 • IrDA处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> • 能读写 IrDA 使用的 SFR。

注意1.在设置IrDA时，必须先将IRDAEN位置“1”。当IRDAEN位为“0”时，忽视IrDA的控制寄存器的写操作，而且读取值全部为初始值。

15.2.2 IrDA控制寄存器（IRCR）

这是控制IrDA功能的寄存器。进行接收数据和发送数据的极性切换、IrDA的时钟选择、以及串行输入/输出引脚功能（通常的串行功能和IrDA功能）切换的选择。通过8位存储器操作指令设置IRCR寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图15-3 IrDA控制寄存器（IRCR）的格式

地址：40044000H

复位后：00H R/W

符号	7	6	5	4	3	2	1	0
IRCR	IRE	IRCKS2	IRCKS1	IRCKS0	IRTXINV	IRRXINV	0	0

IRE	IrDA的允许
0	串行输入/输出引脚用作通常的串行功能
1	串行输入/输出引脚用作IrDA功能

IRCKS2	IRCKS1	IRCKS0	IrDA的时钟选择
0	0	0	B3/16（B=位速率）
0	0	1	$f_{CLK}/2$
0	1	0	$f_{CLK}/4$
0	1	1	$f_{CLK}/8$
1	0	0	$f_{CLK}/16$
1	0	1	$f_{CLK}/32$
1	1	0	$f_{CLK}/64$
1	1	1	禁止设置

IRTXINV	IrTxD数据的极性切换
0	对发送数据进行IrTxD输出
1	反转发送数据后进行IrTxD输出

IRRXINV	IrRxD数据的极性切换
0	将IrRxD引脚的输入数据用作接收数据
1	将反转IrRxD引脚的输入数据后的数据用作接收数据

注意1. 必须将bit1和bit0置“0”。

2. 只有在IRE位为“0”时，才能设置IRCKS[2:0]位、IRTXINV位和IRRXINV位。

15.3 IrDA的运行

15.3.1 IrDA通信的操作步骤

(1) IrDA通信的初始设置流程

按照以下步骤进行IrDA的初始设置。

1. 将PER0寄存器的IRDAEN位置“1”。
2. 设置IRCR寄存器。
3. 设置SCI的相关寄存器（参照UART模式的设置步骤）。

(2) IrDA通信的停止流程

1. 通过设置端口寄存器和端口模式寄存器，设置IrDA通信停止后的IrTxD引脚状态。

备注 在通过步骤3进行IrDA复位时，IrTxD引脚有可能因切换为通常的串行接口UART的数据输出，而改变输出状态。

- 从IrTxD引脚输出低电平的情况
将端口寄存器置“0”。在进行此设置后，IrTxD引脚立即固定为低电平。
 - 从IrTxD引脚输出高电平的情况
将端口寄存器置“1”。通过此设置，在步骤3的IrDA复位后，IrTxD引脚立即固定为高电平。
 - 将IrTxD引脚置为Hi-Z状态的情况将端口模式寄存器置“1”。在进行此设置后，IrTxD引脚立即变为Hi-Z状态。
2. 将STm寄存器（SCI的相关寄存器）的STm0位和STm1位置“1”（停止SCI的通道0和通道1的运行）。
 3. 将PER0寄存器的IRDAEN位置“0”，进行IrDA复位。

不能在上述步骤以外的情况下将STm寄存器的STm0位和STm1位置“1”，或者将IrDA的IRE位置“0”。

(3) 发送IrDA帧错误时的步骤

在IrDA通信过程中发生帧错误时，为了设置为可接收后续数据的状态，必须进行以下设置。

1. 将SCI的STm寄存器的STm1位置“1”（停止SCI的通道1的运行）。
2. 将SCI的SSm寄存器的SSm1位置“1”（开始SCI的通道1的运行）。

备注 m: 单元号 (m=0)

有关SCI的帧错误处理，请参照“第13章 通用串行通信单元”。

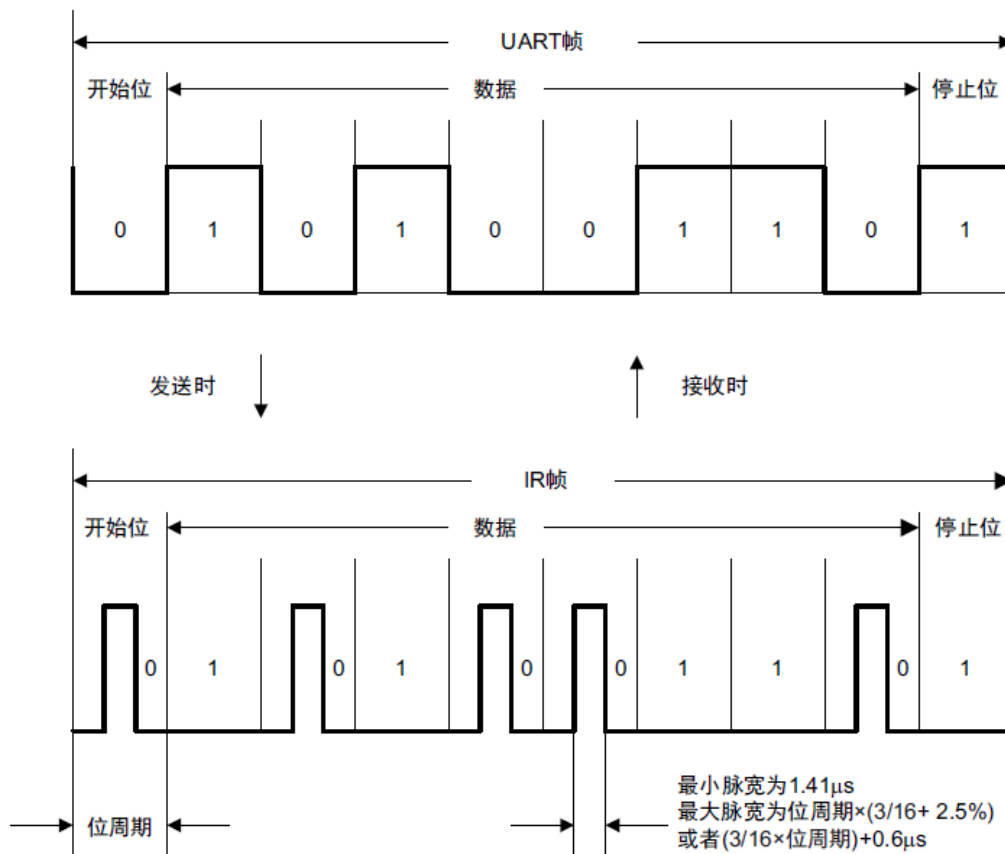
15.3.2 发送

在发送时，来自SCI的输出信号（UART帧）通过IrDA转换为IR帧（参照图15-4）。

在IRTXINV位为“0”并且串行数据为“0”时，输出位周期（1位位宽期间）3/16的高电平脉冲（初始值）。另外，根据IRCKS2~IRCKS0位的设置值可更改高电平脉宽。按标准，规定高电平脉宽最小为1.41us，最大为(3/16+2.5%) x 位周期，或者(3/16 x 位周期)+0.6us。

在CPU或者外围硬件时钟（f_{CLK}）为24MHz时，能设置的最小高电平脉宽为1.5us（满足上述规定的高电平脉宽不小于1.41us的条件）。另外，在串行数据为“1”时，不输出脉冲。

图15-4 IrDA的发送/接收运行图



15.3.3 接收

接收时，IR帧的数据通过IrDA转换为UART帧后，输入到SCI。在IRRXINV位为“0”，并且检测到高电平脉冲时，输出低电平数据。如果在1位期间内没有脉冲，就输出高电平数据。必须注意无法识别小于最小脉宽1.41us的脉冲。

15.3.4 高电平脉宽的选择

如果发送时的脉宽小于位速率 $\times 3/16$ ，可适用的IRCKS2~IRCKS0位的设置（最小脉宽）和设置时的高电平脉宽如表15-2所示。

表15-2 IRCKS2~IRCKS0位的设置值

fCLK [MHz]	项目	<上段>位速率[kbps]					
		<下段>位速率 $\times 3/16$ [us]					
		2.4	9.6	19.2	38.4	57.6	115.2
		78.13	19.53	9.77	4.87	3.26	1.63
1	IRCKS2~ IRCKS0	001	001	001	_注1	_注1	_注1
	高电平脉宽[us]	2.00	2.00	2.00	_注1	_注1	_注1
2	IRCKS2~ IRCKS0	010	010	010	010	010	_注1
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	_注1
3	IRCKS2~ IRCKS0	011	011	011	011	011	_注1
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	_注1
4	IRCKS2~ IRCKS0	011	011	011	011	011	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
6	IRCKS2~ IRCKS0	100	100	100	100	100	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
8	IRCKS2~ IRCKS0	100	100	100	100	100	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
12	IRCKS2~ IRCKS0	101	101	101	101	101	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
16	IRCKS2~ IRCKS0	101	101	101	101	101	000 ^{注2}
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
24	IRCKS2~ IRCKS0	110	110	110	110	110	000 ^{注2}
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50

注 1.“-”表示未达到通信标准。

2.脉宽不能小于位速率 $\times 3/16$ 。

15.4 使用IrDA时的注意事项

1. 能通过外围允许寄存器设置允许或者禁止提供IrDA的运行时钟。初始状态为禁止提供时钟，因此无法存取寄存器。在设置寄存器前，必须通过外围允许寄存器设置为允许提供IrDA运行时钟的状态。
2. 睡眠模式中，IrDA功能持续运行。
3. 在IrDA通信过程中，禁止使用SCI的初始化功能（SS位=1）。
4. 只有在IRE位为“0”时，才能设置IRCR寄存器的IRRXINV位、IRTXINV位和IRCKS[2:0]位。

第16章 增强型DMA

16.1 DMA的功能

DMA是不使用CPU而在存储器之间进行数据传送的功能。通过外围功能中断启动DMA进行数据传送。当DMA和CPU同时访问FLASH, SRAM0, SRAM1或外围模块中的同一单元时, 其总线使用权高于CPU。当DMA和CPU分别访问FLASH, SRAM0, SRAM1或外围模块中的不同单元时, 两者互不干扰, 可以并行执行。

DMA的规格如表16-1所示。

表16-1 DMA的规格(1/2)

项目		规格
启动源		最大24个启动源
可分配的控制数据		24组
可传送的地址空间	地址空间	全地址范围空间
	源	全地址范围空间可选
	目标	全地址范围空间可选
最大传送次数	正常模式	65535次
	重复模式	65535次
最大传送块大小	正常模式 (8位传送)	65535字节
	正常模式 (16位传送)	131070字节
	正常模式 (32位传送)	262140字节
	重复模式	65535字节
传送单位		8位/16位/32位
传送模式	正常模式	在进行DMACTj寄存器从“1”变为“0”的传送后结束。
	重复模式	在DMACTj寄存器从“1”变为“0”的传送结束后, 对重复区的地址进行初始化, 在将DMRLDj寄存器的值重新加载到DMACTj寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表16-5 DMA启动源和向量地址”

表16-1 DMA的规格(2/2)

项目		规格
中断请求	正常模式	在进行DMACTj寄存器从“1”变为“0”的数据传送时，向CPU请求启动源的中断，并进行中断处理。
	重复模式	在DMACRj寄存器的RPTINT位为“1”（允许产生中断）并且进行DMACTj寄存器从“1”变为“0”的数据传送时，向CPU请求启动源的中断，并进行中断处理。
传送开始		如果将DMAENi寄存器的DMAENi0~DMAENi7位置“1”（允许启动），就在每次发生DMA启动源时开始传送数据。
传送停止	正常模式	<ul style="list-style-type: none"> •将DMAENi0~DMAENi7位置“0”（禁止启动）。 •当DMACTj寄存器从“1”变为“0”的数据传送结束时
	重复模式	<ul style="list-style-type: none"> •将DMAENi0~DMAENi7位置“0”（禁止启动）。 •当RPTINT位为“1”（允许产生中断）并且DMACTj寄存器从“1”变为“0”的数据传送结束时

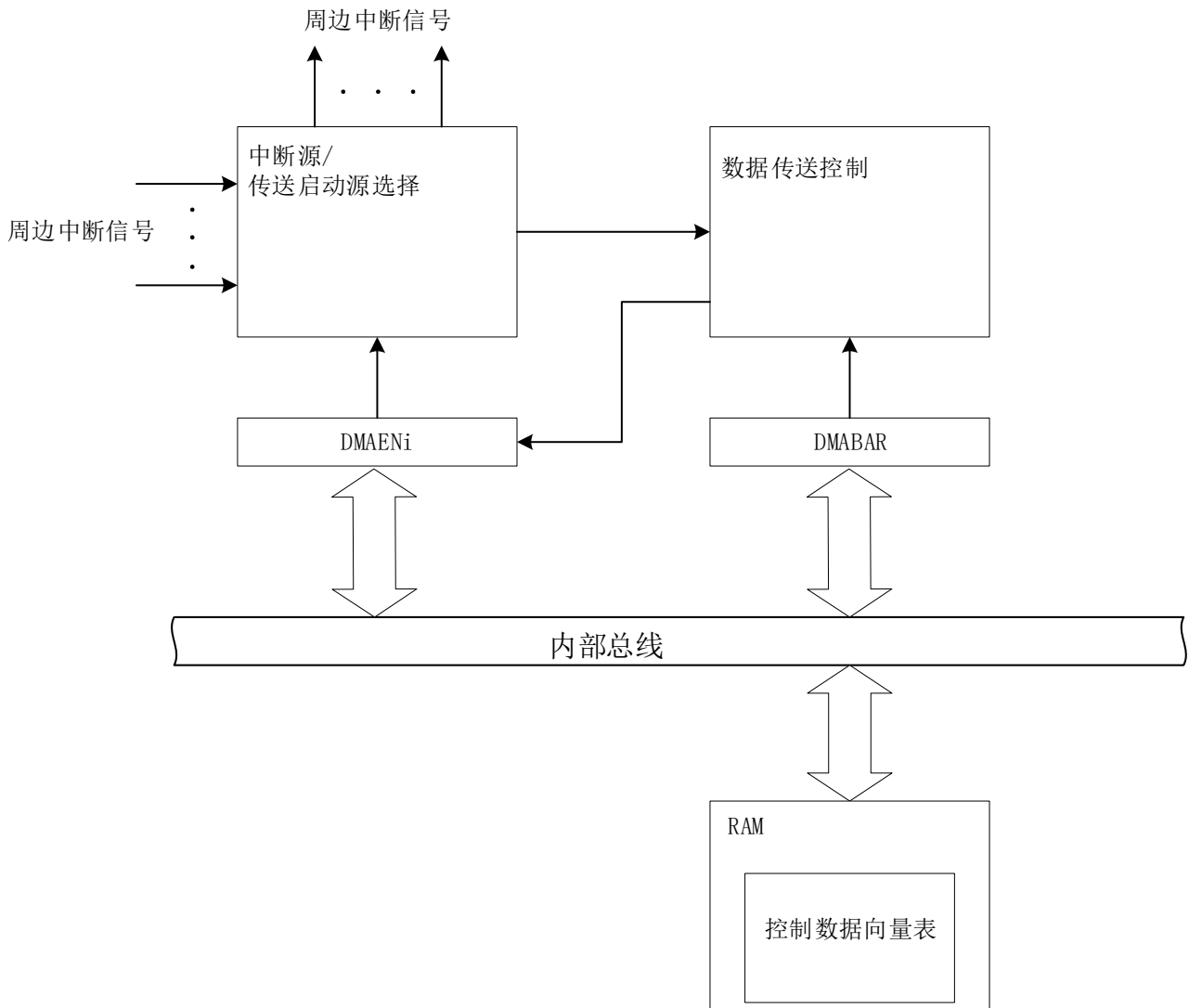
注 在深度睡眠模式，因为闪存停止运行，所以不能作为DMA传送源。

备注 i=0~2、j=0~23

16.2 DMA的结构

DMA的框图如图16-1所示

图16-1 DMA的框图



16.3 控制DMA的寄存器

控制DMA的寄存器如表16-2所示。

表16-2 控制DMA的寄存器

寄存器名	符号
外围允许寄存器1	PER1
DMA启动允许寄存器0	DMAEN0
DMA启动允许寄存器1	DMAEN1
DMA启动允许寄存器2	DMAEN2
DMA基址寄存器	DMABAR

DMA的控制数据如表16-3所示。

DMA的控制数据分配在RAM的DMA控制数据区。通过DMABAR寄存器设置DMA控制数据区和包含DMA向量表区（保存控制数据的起始地址）的416字节区域。

表16-3 DMA的控制数据

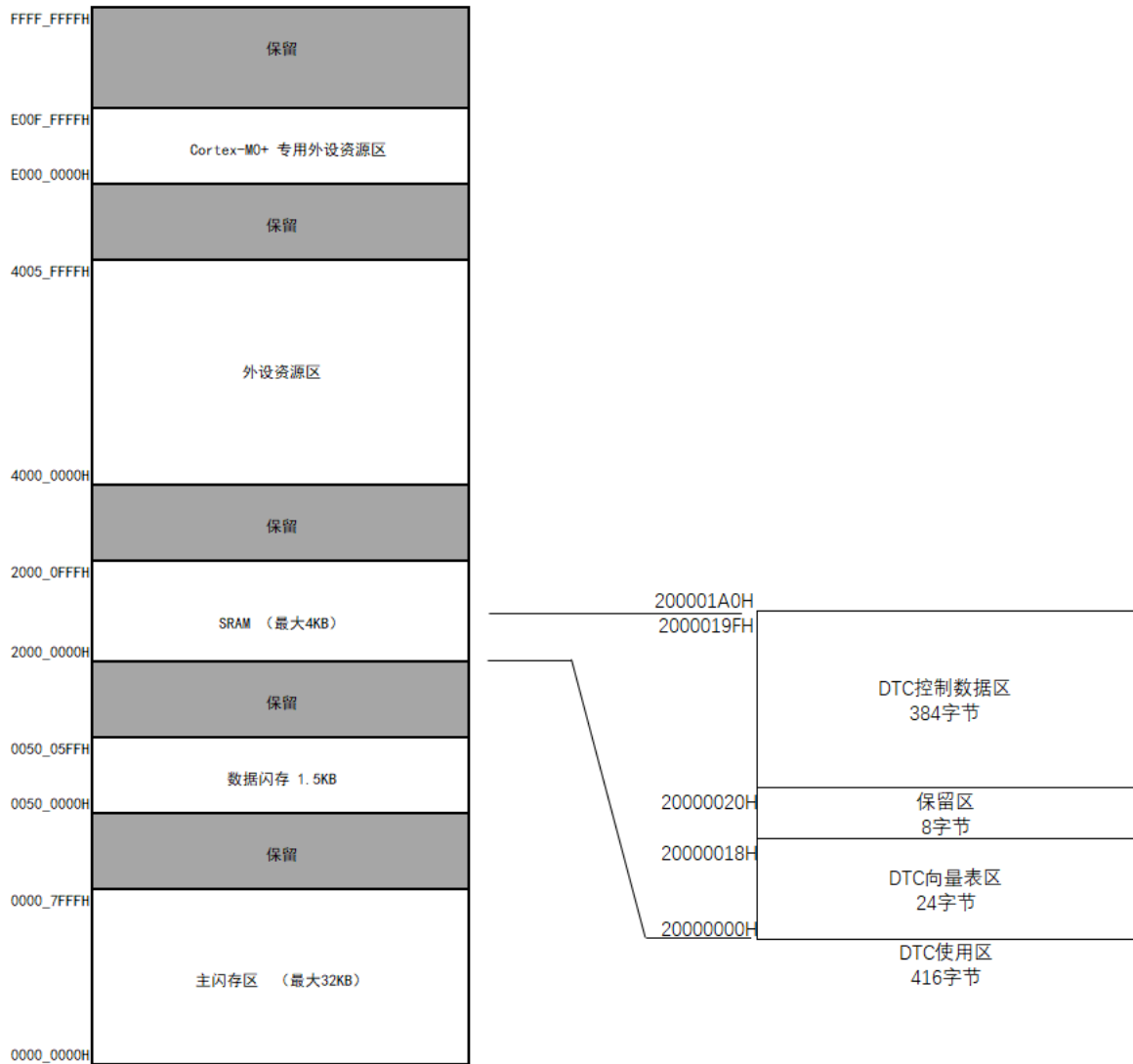
寄存器名	符号
DMA控制寄存器j	DMACRj
DMA块大小寄存器j	DMBLSj
DMA传送次数寄存器j	DMACTj
DMA传送次数重加载寄存器j	DMRLDj
DMA源地址寄存器j	DMSARj
DMA目标地址寄存器j	DMDARj

备注 j=0~23

16.3.1 DMA控制数据区和DMA向量表的分配

通过DMABAR寄存器将分配DMA的控制数据和向量表的416字节区域设置到RAM区。
DMABAR寄存器的设置值为“20000000H”时的存储器映像例子如图16-2所示。
DMA控制数据区的384字节中DMA不使用的空间能用作RAM。

图16-2 DMABAR寄存器的设置值为“20000000H”时的存储器映像例子



16.3.2 控制数据的分配

从起始地址开始，按照DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj (j=0~23) 寄存器的顺序分配控制数据。

起始地址由DMABAR寄存器设置，低10位由各启动源分配的向量表分别设置。
控制数据的分配如图16-3所示。

- 注意1.必须在对应的DMAENi (i=0~2) 的DMAENi0~DMAENi7位为“0”（禁止启动）时更改DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj寄存器的数据。
- 2.不能通过DMA传送进行DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj和DMDARj的存取。

图16-3 控制数据的分配(DMABAR设置为2000000H)

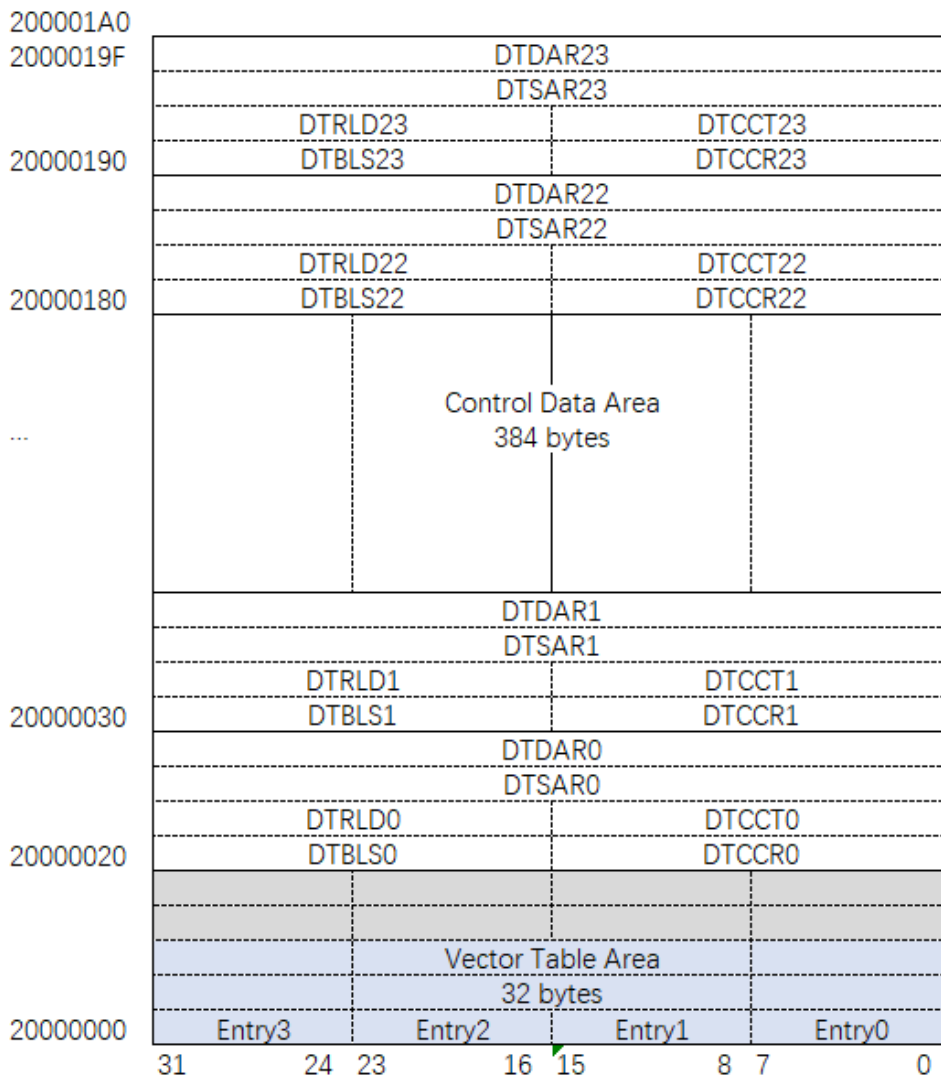


表16-4 控制数据的起始地址

j	地址	j	地址
11	baseaddr+D0H	23	baseaddr+190H
10	baseaddr+C0H	22	baseaddr+180H
9	baseaddr+B0H	21	baseaddr+170H
8	baseaddr+A0H	20	baseaddr+160H
7	baseaddr+90H	19	baseaddr+150H
6	baseaddr+80H	18	baseaddr+140H
5	baseaddr+70H	17	baseaddr+130H
4	baseaddr+60H	16	baseaddr+120H
3	baseaddr+50H	15	baseaddr+110H
2	baseaddr+40H	14	baseaddr+100H
1	baseaddr+30H	13	baseaddr+F0H
0	baseaddr+20H	12	baseaddr+E0H

备注 baseaddr: DMABAR寄存器的设置值

16.3.3 向量表

DMA一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在DMA控制数据区的控制数据。

DMA启动源和向量地址如表16-5所示。各启动源的向量表有1字节，保存“00H”~“17H”的数据，从24组的控制数据中选择1组数据。向量地址的高22位由DMABAR寄存器设置，低10位被分配了对应启动源的“00H”~“17H”。

注意 必须在对应的DMAENi (i=0~2) 寄存器的DMAENi0~DMAENi7位为“0”（禁止启动）时更改设置在向量表中的DMA控制数据区的起始地址。

图16-4 控制数据的起始地址和向量表
DMABAR寄存器的设置值为“2000000H”的情况（例）

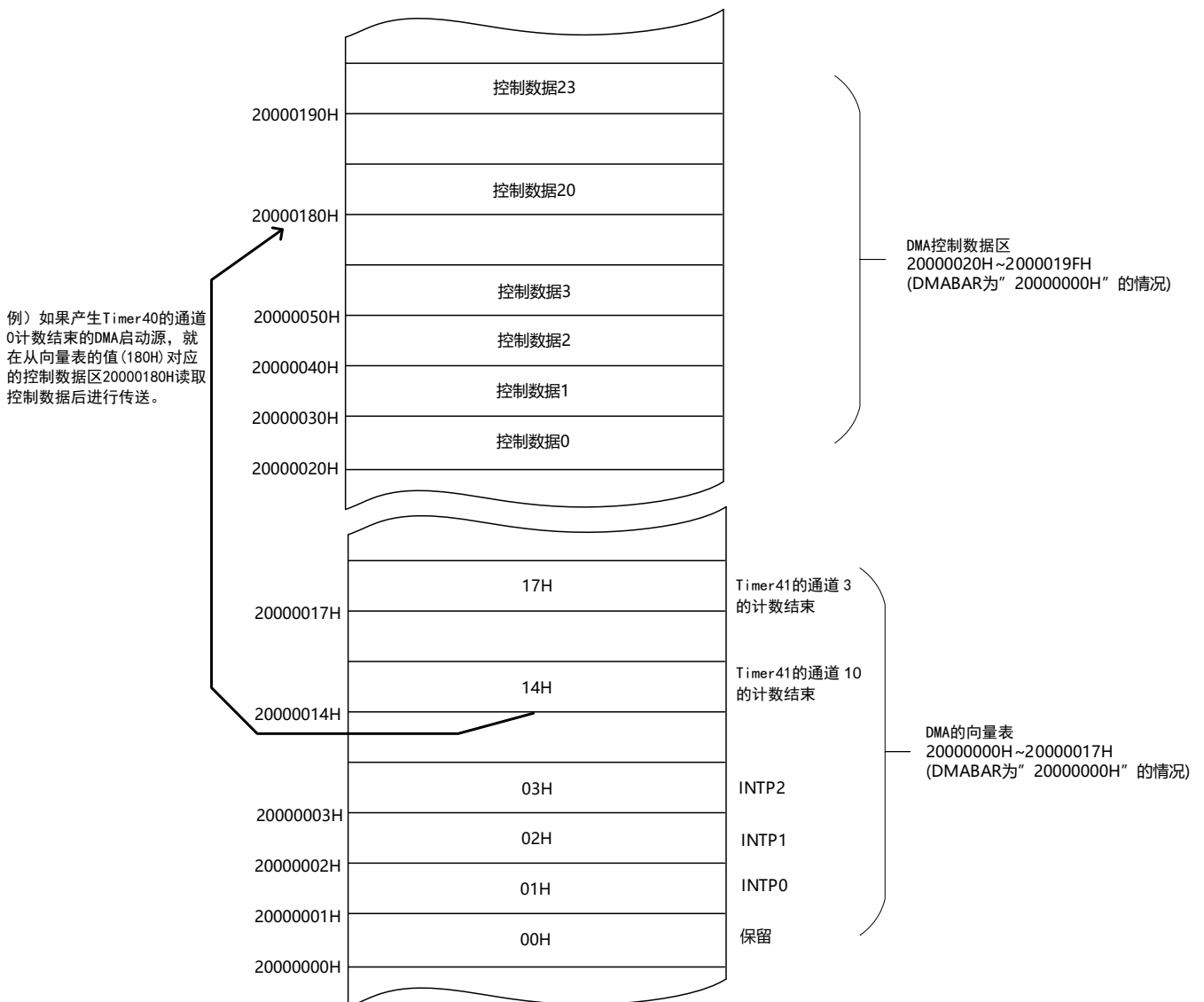




表16-5 DMA启动源和向量地址

DMA启动源（中断请求发生源）	源号	向量地址	优先级
Flash读写擦出结束	0	DMABAR寄存器的设置地址+00H	高   低
INTP0	1	DMABAR寄存器的设置地址+01H	
INTP1	2	DMABAR寄存器的设置地址+02H	
INTP2	3	DMABAR寄存器的设置地址+03H	
INTP3	4	DMABAR寄存器的设置地址+04H	
A/D转换结束	5	DMABAR寄存器的设置地址+05H	
比较器检测0	6	DMABAR寄存器的设置地址+06H	
比较器检测1	7	DMABAR寄存器的设置地址+07H	
UART0接收的传送结束/SSPI01的传送结束或者缓冲器空/IIC01的传送结束	8	DMABAR寄存器的设置地址+08H	
UART0发送的传送结束/SSPI00的传送结束或者缓冲器空/IIC00的传送结束	9	DMABAR寄存器的设置地址+09H	
UART1接收的传送结束/SSPI11的传送结束或者缓冲器空/IIC11的传送结束	10	DMABAR寄存器的设置地址+0AH	
UART1发送的传送结束/SSPI10的传送结束或者缓冲器空/IIC10的传送结束	11	DMABAR寄存器的设置地址+0BH	
UART2接收的传送结束/SSPI21的传送结束或者缓冲器空/IIC21的传送结束	12	DMABAR寄存器的设置地址+0CH	
UART2发送的传送结束/SSPI20的传送结束或者缓冲器空/IIC20的传送结束	13	DMABAR寄存器的设置地址+0DH	
IICA0通信结束。	14	DMABAR寄存器的设置地址+0EH	
15位间隔定时器产生计数中断	15	DMABAR寄存器的设置地址+0FH	
Timer40的通道0的计数或捕捉结束	16	DMABAR寄存器的设置地址+10H	
Timer40的通道1的计数或捕捉结束	17	DMABAR寄存器的设置地址+11H	
Timer40的通道2的计数或捕捉结束	18	DMABAR寄存器的设置地址+12H	
Timer40的通道3的计数或捕捉结束	19	DMABAR寄存器的设置地址+13H	
Timer41的通道0的计数或捕捉结束	20	DMABAR寄存器的设置地址+14H	
Timer41的通道1的计数或捕捉结束	21	DMABAR寄存器的设置地址+15H	
Timer41的通道2的计数或捕捉结束	22	DMABAR寄存器的设置地址+16H	
Timer41的通道3的计数或捕捉结束	23	DMABAR寄存器的设置地址+17H	

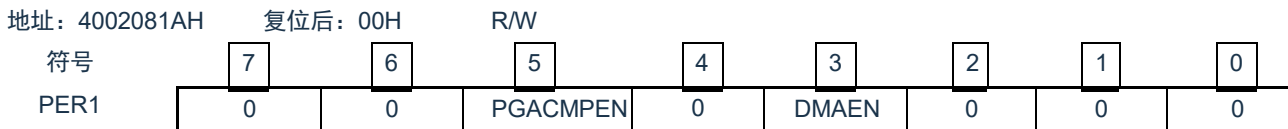
16.3.4 外围允许寄存器1 (PER1)

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

要使用DMA时，必须将bit3 (DMAEN) 置“1”。

通过8位存储器操作指令设置PER1寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图16-5 外围允许寄存器1 (PER1) 的格式



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

16.3.5 DMA控制寄存器j (DMACRj) (j=0~23)

DMACRj寄存器控制DMA的运行模式。

图16-6 DMA控制寄存器j (DMACRj) 的格式

地址：参照“16.3.2 控制数据的分配”。

复位后：不定值 R/W

符号：

	15	14	13	12	11	10	9	8
DMACRj	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	SZ		RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	传送数据长度的选择
00	8位
01	16位
10	32位
11	禁止设置

RPTINT	重复模式中中断的允许/禁止
0	禁止发生中断。
1	允许发生中断。
在MODE位为“0”（正常模式）时，RPTINT位的设置无效。	

CHNE	链传送的允许/禁止
0	禁止链传送。
1	允许链传送。
必须将DMACR23寄存器的CHNE位置“0”（禁止链传送）。	

DAMOD	传送目标地址的控制
0	固定
1	递增
在MODE位为“1”（重复模式）并且RPTSEL位为“0”（传送目标为重复区）时，DAMOD位的设置无效。	

SAMOD	传送源地址的控制
0	固定
1	递增
在MODE位为“1”（重复模式）并且RPTSEL位为“1”（传送源为重复区）时，SAMOD位的设置无效。	

RPTSEL	重复区的选择
0	传送目标为重复区。
1	传送源为重复区。
在MODE位为“0”（正常模式）时，RPTSEL位的设置无效。	

MODE	传送模式的选择
0	正常模式
1	重复模式

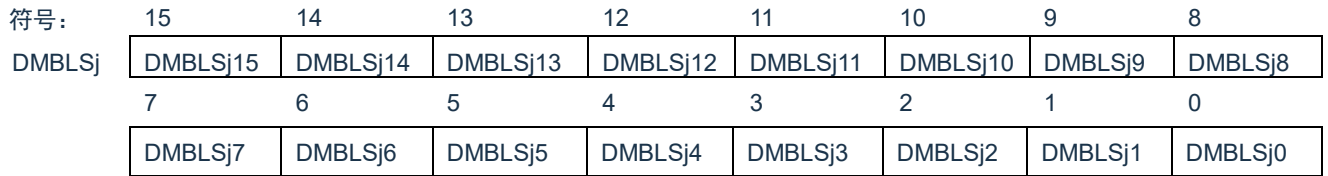
注意 不能通过DMA传送进行DMACRj寄存器的存取。

16.3.6 DMA块大小寄存器j (DMBLSj) (j=0~23)

此寄存器设置1次启动传送数据的块大小。

图16-7 DMA块大小寄存器j (DMBLSj) 的格式

地址：参照“16.3.2 控制数据的分配”。 复位后：不定值 R/W



DMBLSj	传送块大小		
	8位传送	16位传送	32位传送
00H	禁止设置	禁止设置	禁止设置
01H	1字节	2字节	4字节
02H	2字节	4字节	8字节
03H	3字节	6字节	12字节
⋮	⋮	⋮	⋮
FDH	253字节	506字节	1012字节
FEH	254字节	508字节	1016字节
FFH	255字节	510字节	1020字节
⋮	⋮	⋮	⋮
FFFFH	65535字节	131070字节	262140字节

注意 1.不能通过DMA传送进行DMBLSj寄存器的存取。

16.3.7 DMA传送次数寄存器j (DMACTj) (j=0~23)

此寄存器设置DMA的数据传送次数。每当启动1次DMA传送就减1。

图16-8 DMA传送次数寄存器j (DMACTj) 的格式

符号: DMACTj

	15	14	13	12	11	10	9	8
DMACTj	DMACTj15	DMACTj14	DMACTj13	DMACTj12	DMACTj11	DMACTj10	DMACTj9	DMACTj8
	7	6	5	4	3	2	1	0
	DMACTj7	DMACTj6	DMACTj5	DMACTj4	DMACTj3	DMACTj2	DMACTj1	DMACTj0

地址: 参照“16.3.2 控制数据的分配”。

复位后: 不定值

R/W

DMACTj	传送次数
00H	禁止设置
01H	1次
02H	2次
03H	3次
⋮	⋮
FDH	253次
FEH	254次
FFH	255次
⋮	⋮
FFFFH	65535次

注意 1. 不能通过DMA传送进行DMACTj寄存器的存取。

16.3.8 DMA传送次数重加载寄存器j (DMRLDj) (j=0~23)

此寄存器设置重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到DMACT寄存器，所以设置值必须和DMACT寄存器的初始值相同。

图16-9 DMA传送次数重加载寄存器j (DMRLDj) 的格式

地址：参照“16.3.2 控制数据的分配”。 复位后：不定值 R/W

符号：	15	14	13	12	11	10	9	8
DMRLDj	DMRLDj15	DMRLDj14	DMRLDj13	DMRLDj12	DMRLDj11	DMRLDj10	DMRLDj9	DMRLDj8
	7	6	5	4	3	2	1	0
	DMRLDj7	DMRLDj6	DMRLDj5	DMRLDj4	DMRLDj3	DMRLDj2	DMRLDj1	DMRLDj0

注意 1.不能通过DMA传送进行DMRLDj寄存器的存取。

16.3.9 DMA源地址寄存器j (DMSARj) (j=0~23)

此寄存器指定数据传送时的传送源地址。

当DMACRj寄存器的SZ位为“01”（16位传送）时，忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”（32位传送）时，忽视低2位而作为word地址进行处理。

图16-10 DMA源地址寄存器j (DMSARj) 的格式

地址：参照“16.3.2 控制数据的分配”。 复位后：不定值 R/W

符号	31	30	29	28	27	26	25	24
DMSARj	DMSARj3 1	DMSARj3 0	DMSARj2 9	DMSARj2 8	DMSARj2 7	DMSARj2 6	DMSARj2 5	DMSARj2 4
	23	22	21	20	19	18	17	16
	DMSARj2 3	DMSARj2 2	DMSARj2 1	DMSARj2 0	DMSARj1 9	DMSARj1 8	DMSARj1 7	DMSARj1 6
	15	14	13	12	11	10	9	8
DMSARj1 5	DMSARj1 4	DMSARj1 3	DMSARj1 2	DMSARj1 1	DMSARj1 0	DMSARj9	DMSARj8	
7	6	5	4	3	2	1	0	
DMSARj7	DMSARj6	DMSARj5	DMSARj4	DMSARj3	DMSARj2	DMSARj1	DMSARj0	

注意1.不能通过DMA传送进行DMSARj寄存器的存取。

16.3.10 DMA目标地址寄存器j (DMDARj) (j=0~23)

此寄存器指定数据传送时的传送目标地址。

当DMACRj寄存器的SZ位为“01”（16位传送）时，忽视最低位而作为偶地址进行处理。

当DMACRj寄存器的SZ位为“10”（32位传送）时，忽视低2位而作为word地址进行处理。

图16-11 DMA目标地址寄存器j (DMDARj) 的格式

地址：参照“16.3.2 控制数据的分配”。 复位后：不定值 R/W

符号	31	30	29	28	27	26	25	24
DMDARj	DMDARj3 1	DMDARj3 0	DMDARj2 9	DMDARj2 8	DMDARj2 7	DMDARj2 6	DMDARj2 5	DMDARj2 4
	23	22	21	20	19	18	17	16
	DMDARj2 3	DMDARj2 2	DMDARj2 1	DMDARj2 0	DMDARj1 9	DMDARj1 8	DMDARj1 7	DMDARj1 6
	15	14	13	12	11	10	9	8
DMDARj1 5	DMDARj1 4	DMDARj1 3	DMDARj1 2	DMDARj1 1	DMDARj1 0	DMDARj9	DMDARj8	
7	6	5	4	3	2	1	0	
DMDARj7	DMDARj6	DMDARj5	DMDARj4	DMDARj3	DMDARj2	DMDARj1	DMDARj0	

注意:不能通过DMA传送进行DMDARj寄存器的存取。

16.3.11 DMA启动允许寄存器i (DMAENi) (i=0~2)

这是控制允许或者禁止通过各中断源启动DMA的8位寄存器。中断源和DMAENi0~DMAENi7位的对应如表16-6所示。

能通过8位存储器操作指令设置DMAENi寄存器。

注意1.必须在不产生对应该位的启动源的位置更改DMAENi0~DMAENi7位。

2.不能通过DMA传送进行DMAENi寄存器的存取。

3.分配的功能因产品而不同，必须将没有分配功能的位置“0”。

图16-12 DMA启动允许寄存器i (DMAENi) (i=0~2) 的格式

地址: 40005000H (DMAEN0)、40005001H (DMAEN1)、
40005002H (DMAEN2) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DMAENi	DMAENj7	DMAENj6	DMAENj5	DMAENj4	DMAENj3	DMAENj2	DMAENj1	DMAENj0

DMAENi7	DMA启动的允许i7
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi7位变为“0”（禁止启动）。	

DMAENi6	DMA启动的允许i6
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi6位变为“0”（禁止启动）。	

DMAENi5	DMA启动的允许i5
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi5位变为“0”（禁止启动）。	

DMAENi4	DMA启动的允许i4
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi4位变为“0”（禁止启动）。	

DMAENi3	DMA启动的允许i3
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi3位变为“0”（禁止启动）。	

DMAENi2	DMA启动的允许i2
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi2位变为“0”（禁止启动）。	

DMAENi1	DMA启动的允许i1
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi1位变为“0”（禁止启动）。	

DMAENi0	DMA启动的允许i0
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi0位变为“0”（禁止启动）。	

表16-6 中断源和DMAENi0~DMAENi7位的对应

寄存器	DMAENi7位	DMAENi6位	DMAENi5位	DMAENi4位	DMAENi3位	DMAENi2位	DMAENi1位	DMAENi0位
DMAEN0	比较器检测1	比较器检测0	A/D转换结束	INTP3	INTP2	INTP1	INTP0	Flash擦除/写入结束
DMAEN1	15比特间隔定时器中断	IICA0通信结束	UART2发送的传送结束/SSPI20的传送结束或者缓冲器空/IIC20的传送结束	UART2接收的传送结束/SSPI21的传送结束或者缓冲器空/IIC21的传送结束	UART1发送的传送结束/SSPI10的传送结束或者缓冲器空/IIC10的传送结束	UART1接收的传送结束/SSPI11的传送结束或者缓冲器空/IIC11的传送结束	UART0发送的传送结束/SSPI00的传送结束或者缓冲器空/IIC00的传送结束	UART0接收的传送结束/SSPI01的传送结束或者缓冲器空/IIC01的传送结束
DMAEN2	定时器阵列单元1的通道3的计数结束或者捕捉结束	定时器阵列单元1的通道2的计数结束或者捕捉结束	定时器阵列单元1的通道1的计数结束或者捕捉结束	定时器阵列单元1的通道0的计数结束或者捕捉结束	定时器阵列单元0的通道3的计数结束或者捕捉结束	定时器阵列单元0的通道2的计数结束或者捕捉结束	定时器阵列单元0的通道1的计数结束或者捕捉结束	定时器阵列单元0的通道0的计数结束或者捕捉结束

注意 必须将没有分配功能的位置“0”。

备注 i=0~2

16.3.12 DMA基址寄存器 (DMABAR)

这是32位寄存器，设置保存DMA控制数据区起始地址的向量地址以及DMA控制数据区的地址。

注意1. 必须在将全部的DMA启动源设置为禁止启动的状态下更改DMABAR寄存器。

2. 只能改写1次DMABAR寄存器。
3. 不能通过DMA传送进行DMABAR寄存器的存取。
4. 有关DMA控制数据区和DMA向量表区的分配，请参照“16.3.1 DMA控制数据区和DMA向量表区的分配”的注意。
5. 设置该寄存器请保持512Byte对齐，也就是低8位设置为零。DMA硬件忽略低8位。
6. 该寄存器只能WORD访问，BYTE和HALFWORD访问忽略。

图16-13 DMA基址寄存器 (DMABAR) 的格式

地址：40005008H 复位后：00000000H R/W

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

16.4 DMA的运行

DMA一旦启动，就从DMA控制数据区读控制数据，根据此控制数据进行数据传送，并且将数据传送后的控制数据回写到DMA控制数据区。能将24组控制数据保存到DMA控制数据区，并且进行24组数据的传送。传送模式有正常模式和重复模式，传送大小有8位传送，16位传送和32位传送。在DMACRj (j=0~23) 寄存器的CHNE位为“1”（允许链传送）时，通过1个启动源读多个控制数据进行连续的数据传送（链传送）。

通过32位DMSARj寄存器和32位DMDARj寄存器分别指定传送源地址和传送目标地址。在数据传送后，根据控制数据递增或者固定DMSARj寄存器和DMDARj寄存器的值。

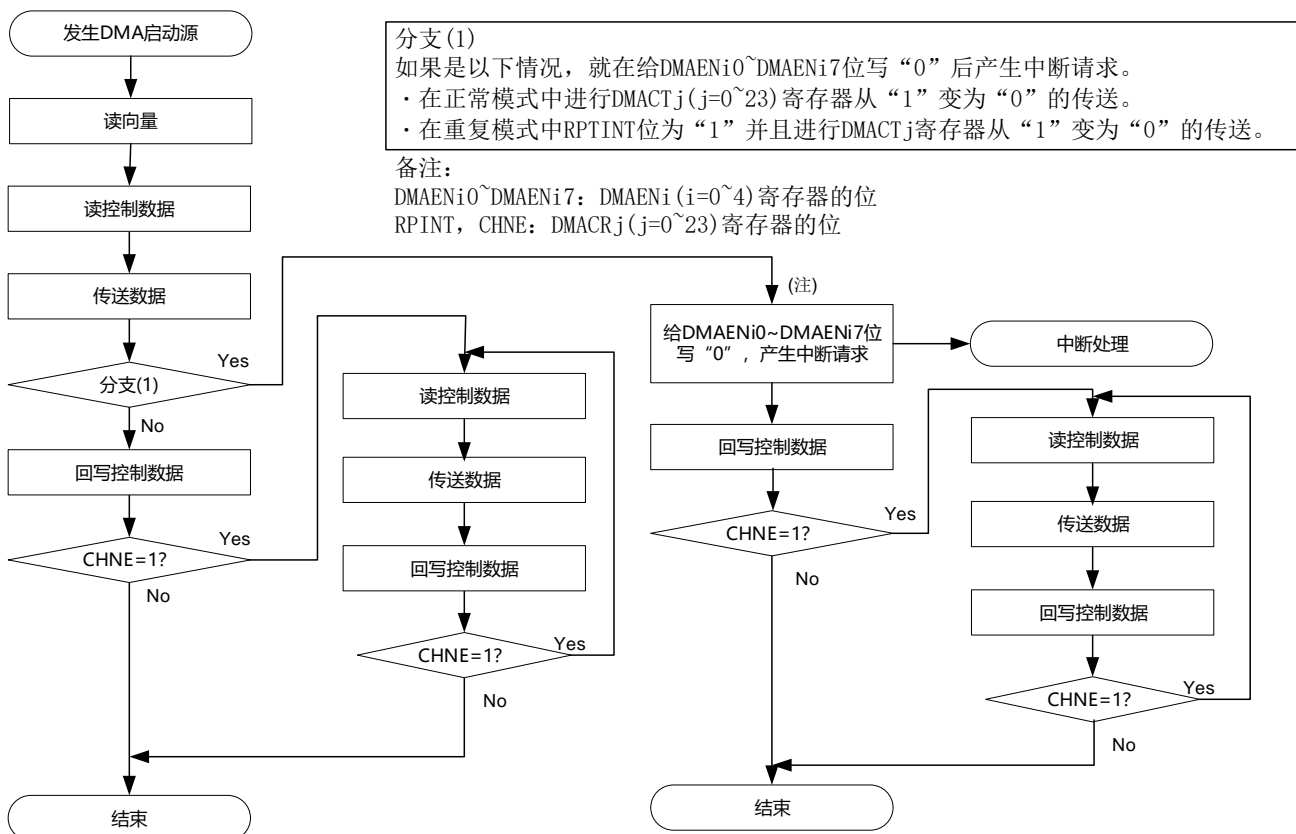
16.4.1 启动源

通过外围功能的中断信号启动DMA，并且通过DMAENi (i=0~2) 寄存器选择启动DMA的中断信号。当数据传送（在进行链传送时，连续进行最初的传送）的设置为下述两种情况时，就在DMA运行中将对应的DMAENi寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。

- 在正常模式中，进行DMACTj (j=0~23) 寄存器变为“0”的传送。
- 在重复模式中，DMACRj寄存器的RPTINT位为“1”（允许发生中断）并且进行DMACTj寄存器变为“0”的传送。

DMA的内部运行流程图如图16-14所示。

图16-14 DMA的内部运行流程图



注：在通过允许链传送(CHNE=1)的设置启动的数据传送中，不给DMAENi0~DMAENi7位写“0”并且不产生中断请求。

16.4.2 正常模式

在8位传送时，1次启动的传送数据为1~65535字节；在16位传送时，1次启动的传送数据为2~131070字节；在32位传送时，1次启动的传送数据为4~262140字节。传送次数为1~65535次。如果进行DMACTj (j=0~23) 寄存器变为“0”的数据传送，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi (i=0~2) 寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。

正常模式的寄存器功能和数据传送分别如表16-7和图16-15所示。

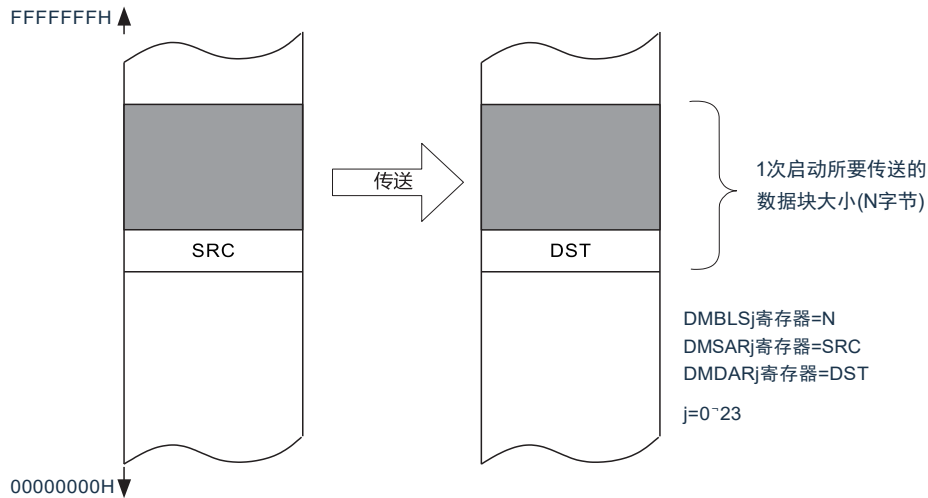
表16-7 正常模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	不使用注。
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

注 当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须进行初始化（00H）。备

注 j=0~23

图16-15 正常模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	固定	固定	SRC	DST
0	1	X	0	递增	固定	SRC+N	DST
1	0	X	0	固定	递增	SRC	DST+N
1	1	X	0	递增	递增	SRC+N	DST+N

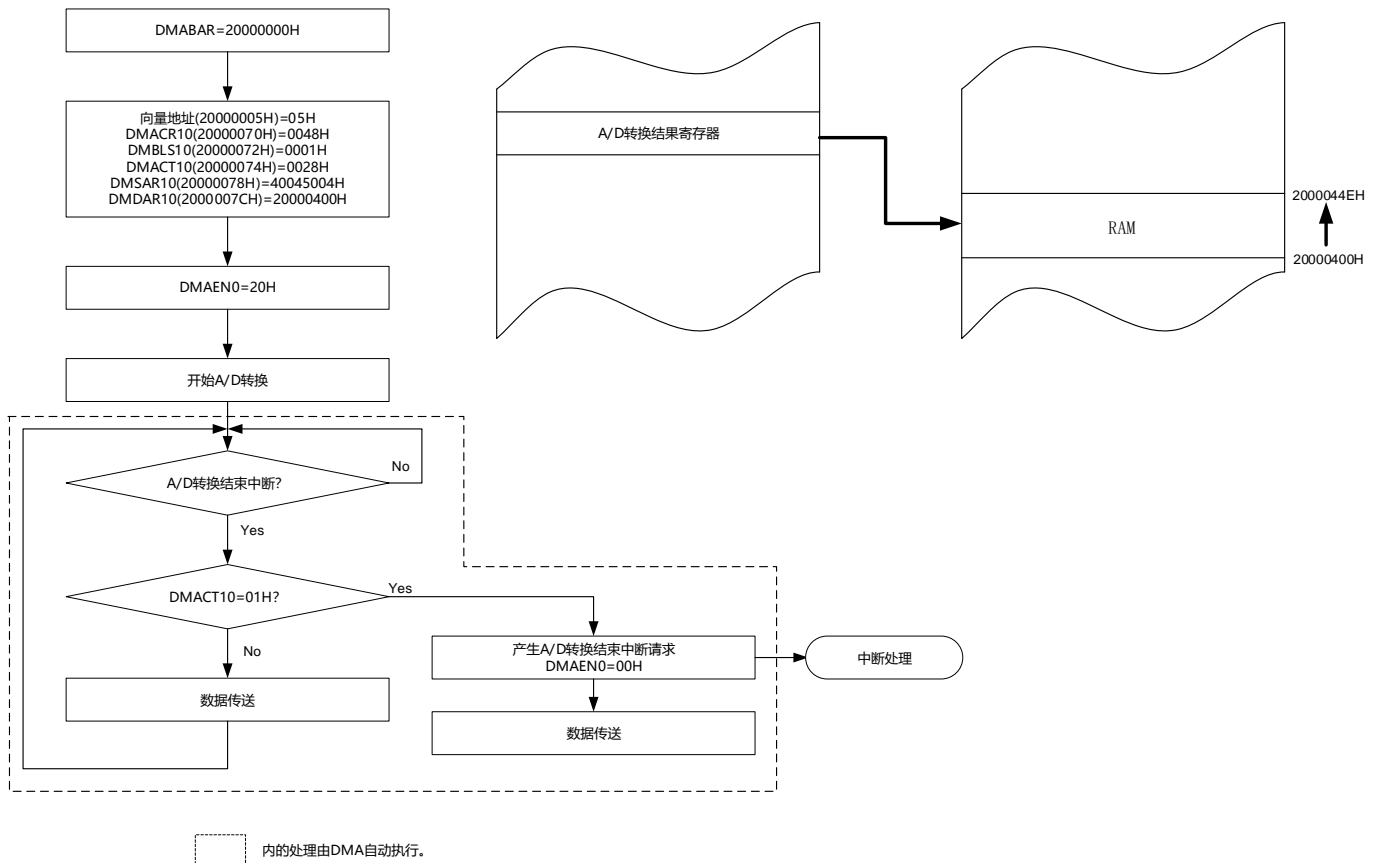
X: “0”或者“1”

(1) 正常模式的使用例子1：连续取A/D转换结果

通过A/D转换结束中断启动DMA，并且将A/D转换结果寄存器的值传送到RAM。

- 向量地址分配在20000005H，控制数据分配在20000070H~2000007FH。
- 将A/D转换结果寄存器（40045004H，40045005H）的2字节数据传送40次到RAM的20000400H~2000044FH的80字节。

图16-16 正常模式的使用例子1：连续取A/D转换结果



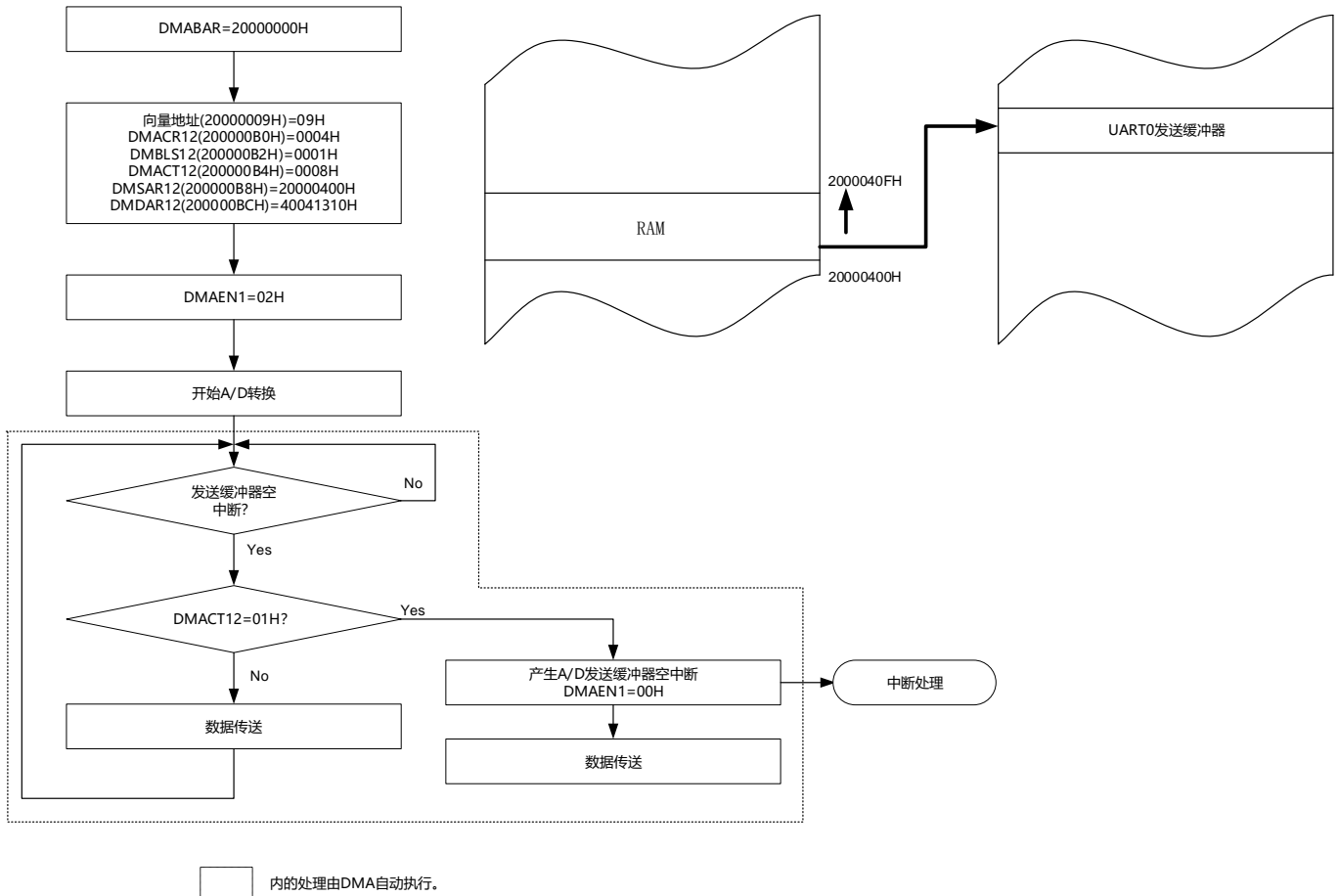
因为是正常模式，所以不使用DMRLD10寄存器的值。但是，当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须对DMRLD10寄存器进行初始化（0000H）。

(2) 正常模式的使用例子2: UART0连续发送

通过UART0的发送缓冲器空中断启动DMA, 并且将RAM的值传送到UART0的发送缓冲器。

- 向量地址分配在20000009H, 控制数据分配在200000B0H~200000BFH。
- 将RAM的20000400H~20000407H的8字节传送到UART0的发送缓冲器 (40041310H)。

图16-17 正常模式的使用例子2: UART0连续发送



因为是正常模式, 所以不使用DMRLD12寄存器的值。但是, 当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位 (RPERDIS=0) 时, 必须对DMRLD12寄存器进行初始化 (0000H)。

必须通过软件开始第1次的UART0发送。通过发送缓冲器空中断启动DMA, 然后自动进行第2次以后的发送。

16.4.3 重复模式

1次启动的传送数据为1~65535字节。将传送源或者传送目标指定为重复区，传送次数为1~65535次。一旦指定次数的传送结束，就对DMACTj (j=0~23) 寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当DMACRj寄存器的RPTINT位为“1”（允许发生中断）并且进行DMACTj寄存器变为“0”的数据传送时，就在DMA运行中向中断控制器产生对应启动源的中断请求，并且将对应的DMAENi (i=0~2) 寄存器的DMAENi0~DMAENi7位置“0”（禁止启动）。当DMACRj寄存器的RPTINT位为“0”（禁止发生中断）时，即使进行DMACTj寄存器变为“0”的数据传送，也不产生中断请求，而且DMAENi0~DMAENi7位不变为“0”。

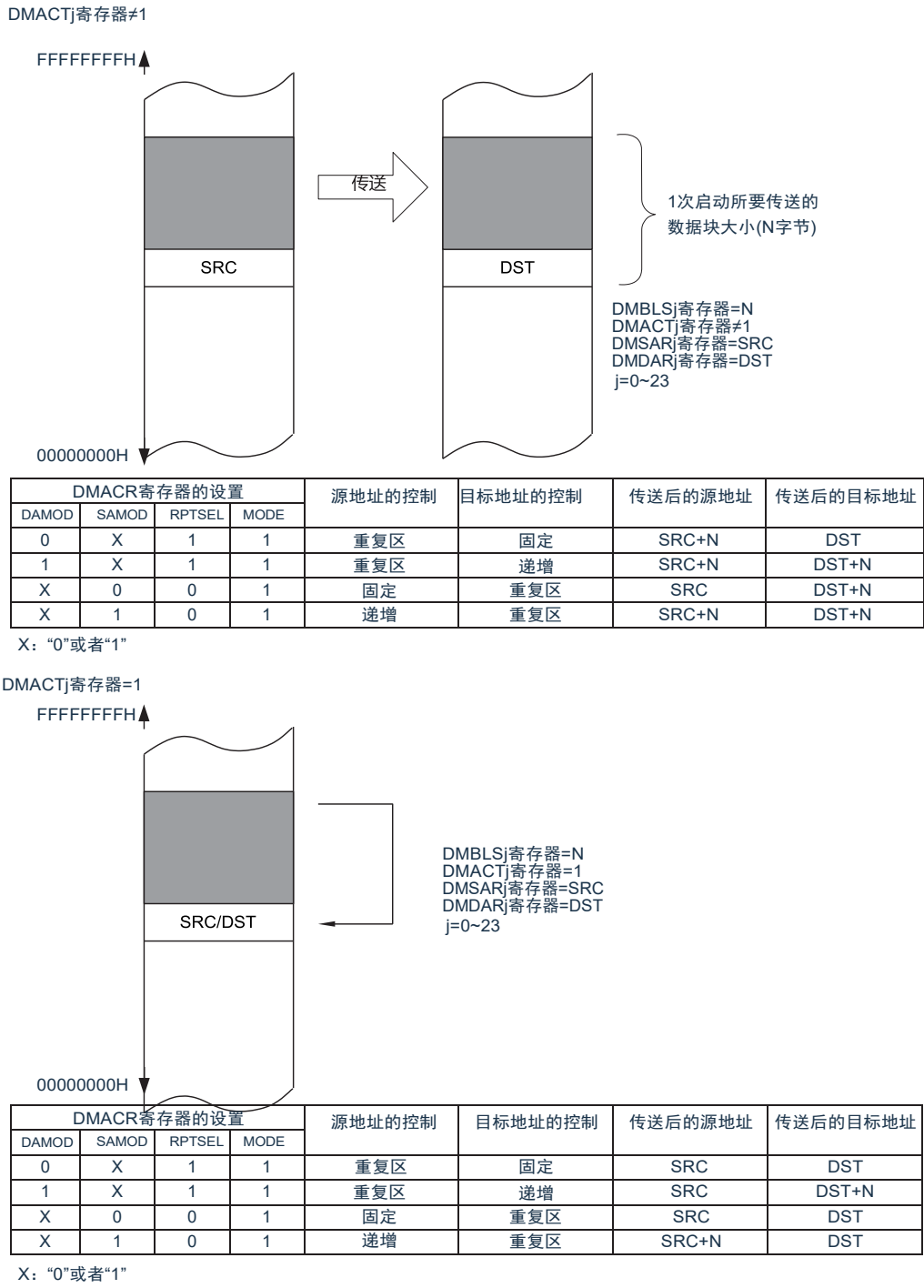
重复模式的寄存器功能和数据传送分别如表16-8和图16-18所示。

表16-8 重复模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	将此寄存器的值重新加载到DMACT寄存器。 (对数据的传送次数进行初始化)
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

备注 j=0~23

图16-18 重复模式的数据传送



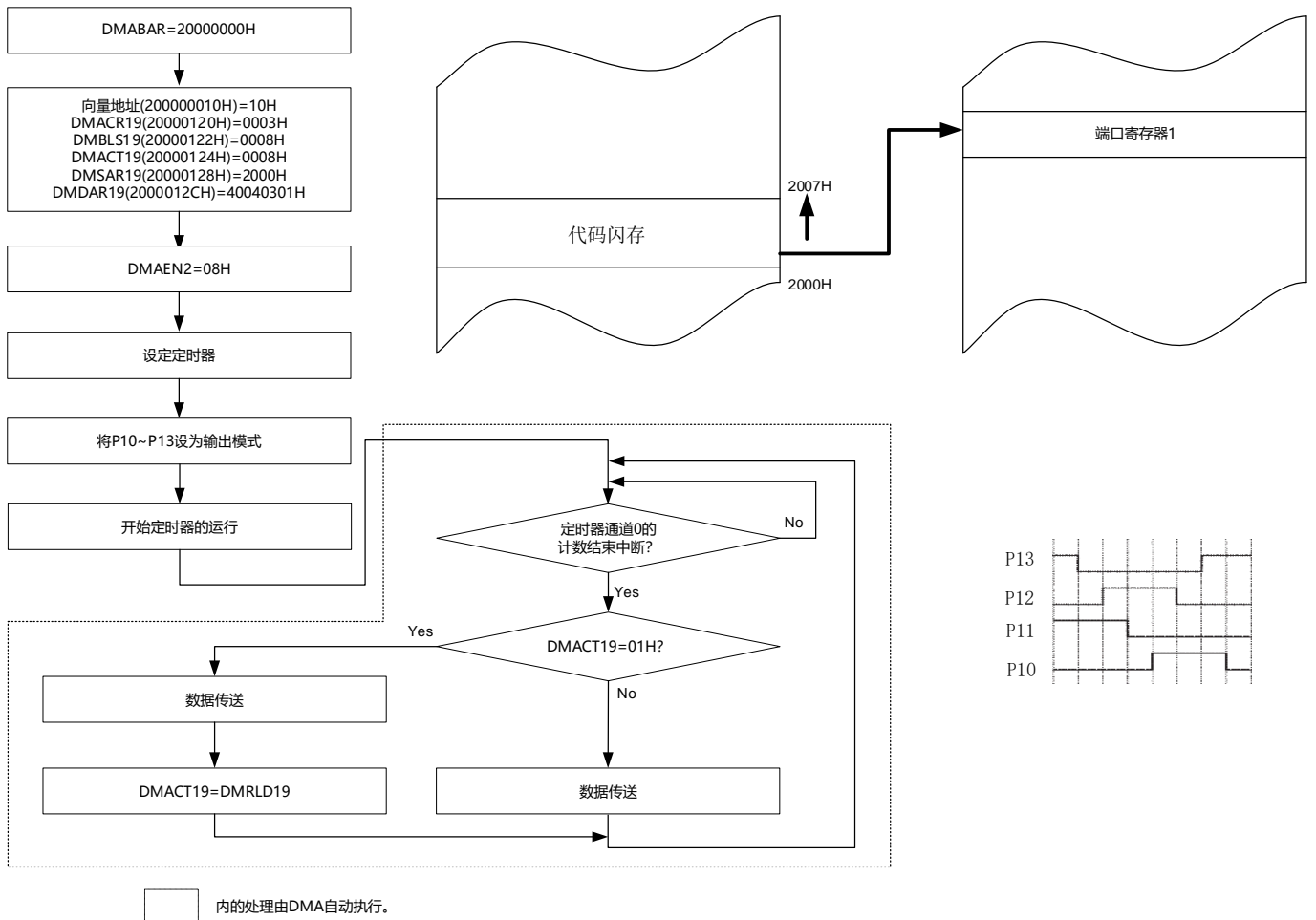
注意1.在使用重复模式时，必须将重复区的数据长度设置在65535字节以内。

(1) 重复模式的使用例子：使用端口的步进马达控制脉冲输出

使用Timer40的通道0间隔定时器功能启动DMA，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口。

- 向量地址分配在20000010H，控制数据分配在20000120H~2000012FH。
- 将代码闪存的02000H~02007H的8字节传送到端口寄存器1（40040301H）。
- 禁止重复模式中断。

图16-19 重复模式的使用例子1：使用端口的步进马达控制脉冲输出



要停止输出时，必须在停止定时器的运行后清除DMAEN2的bit0。

16.4.4 链传送

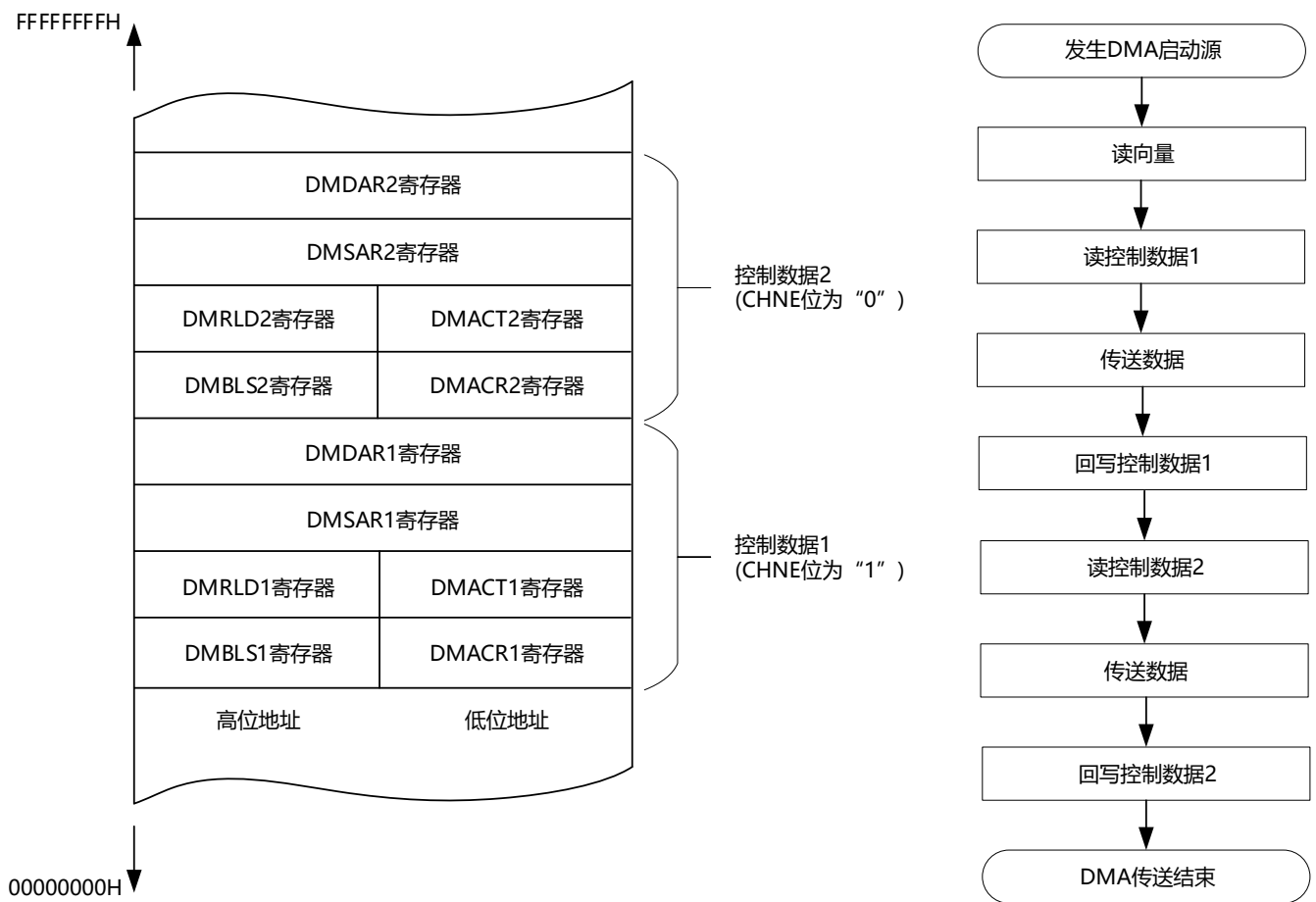
当DMACRj (j=0~23) 寄存器的CHNE位为“1 (允许链传送)”时，能通过1个启动源连续进行多个数据的传送。

DMA一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在DMA控制数据区的控制数据。如果读到的控制数据的CHNE位为“1” (允许链传送)，就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到CHNE位为“0” (禁止链传送) 的控制数据传送结束为止。

在使用多个控制数据进行链传送时，第一个控制数据设置的传送次数有效，而第2个以后处理的控制数据的传送次数无效。

链传送的流程图如图16-20所示。

图16-20 链传送的流程图



注意1. 必须将DMACR23寄存器的CHNE位置“0” (禁止链传送)。

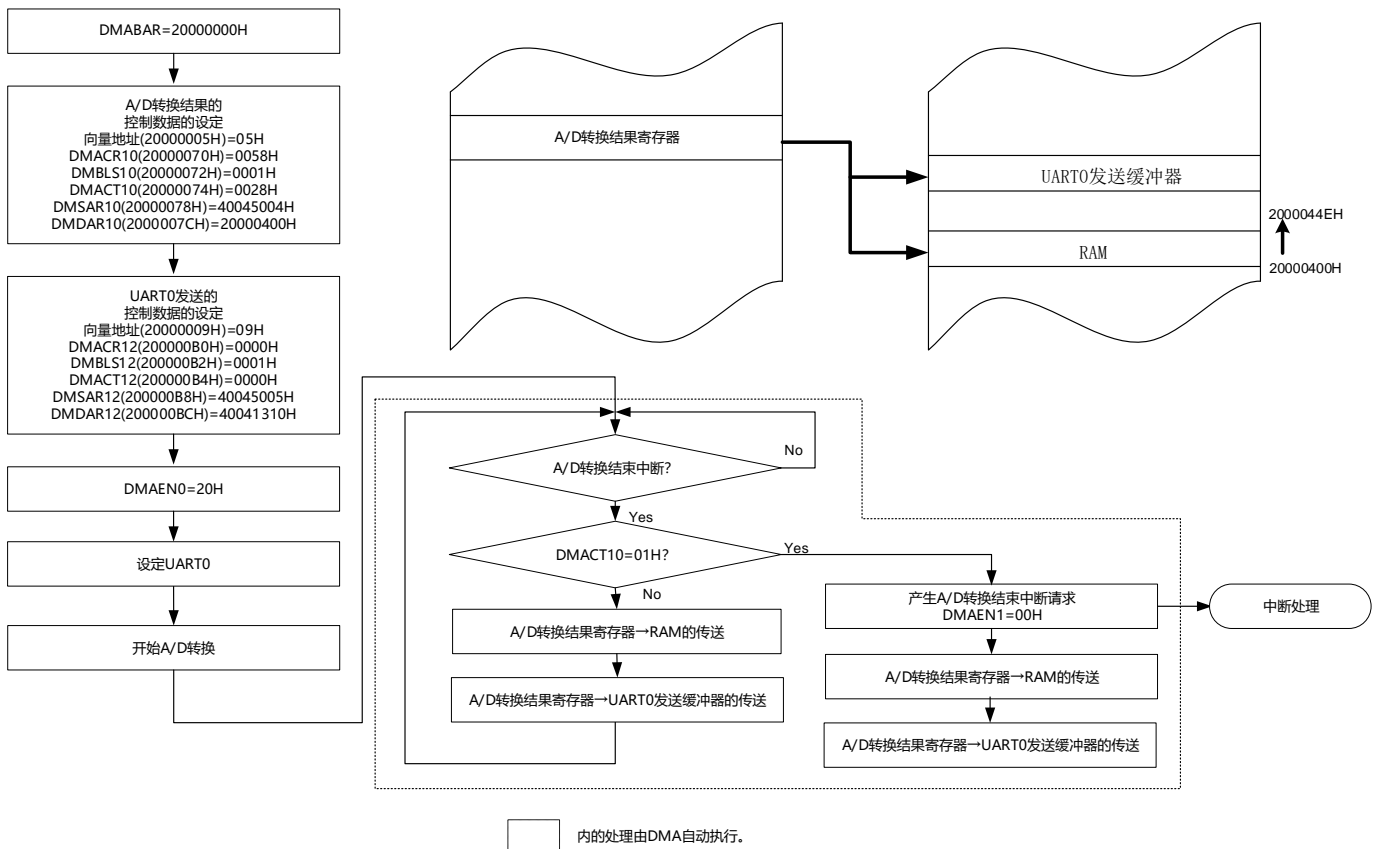
2. 在链传送的第2次以后的数据传送时，DMAENi (i=0~2) 寄存器的DMAENi0~DMAENi7位不变为“0” (禁止DMA启动)，并且不产生中断请求。

(1) 链传送的使用例子：连续取A/D转换结果进行UART0发送

通过A/D转换结束中断启动DMA，并且将A/D转换结果传送到RAM进行UART0发送。

- 向量地址分别为20000005H和20000009H。
- A/D转换结果的控制数据分配在20000070H~2000007FH。
- UART0发送的控制数据分配在200000B0H~200000BFH。
- 将A/D转换结果寄存器（40045004H，40045005H）的2字节数据传送到RAM的20000400H~2000044FH，并且将A/D转换结果寄存器的高位1字节（40045005H）传送到UART0的发送缓冲器（40041310H）。

图16-21 链传送的使用例子：连续取A/D转换结果进行UART0发送



16.5 使用DMA时的注意事项

16.5.1 DMA控制数据和向量表的设置

- 必须在将全部的DMA启动源设置为禁止启动的状态下更改DMA基址寄存器（DMABAR）。
- 只能改写1次DMA基址寄存器（DMABAR）。
- 必须在对应的DMAENi（i=0~2）寄存器的DMAENi0~DMAENi7位为“0”（禁止DMA启动）时更改DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj寄存器的数据。
- 必须在对应的DMAENi（i=0~2）寄存器的DMAENi0~DMAENi7位为“0”（禁止DMA启动）时更改设置在向量表中的DMA控制数据区的起始地址。

16.5.2 DMA控制数据区和DMA向量表区的分配

能分配DMA控制数据和向量表的区域因产品和使用条件而不同。

- 堆栈区、DMA控制数据区和DMA向量表区不能重叠。
- 当通过RAM奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，即使在使用正常模式时也必须对DMRLD寄存器进行初始化（0000H）。

16.5.3 DMA的执行时钟数

DMA启动时的执行情况和所需的时钟数如表16-9所示。

表16-9 DMA启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注1	注2	注2

- 注1.有关回写控制数据所需的时钟数，请参照“表16-10 回写控制数据所需的时钟数”。
- 2.有关读写数据所需的时钟数，请参照“表16-11 读写数据所需的时钟数”。

表16-10 回写控制数据所需的时钟数

DMACR寄存器的设置				地址设置		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DMACTj 寄存器	DMRLDj 寄存器	DMSARj 寄存器	DMDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	固定	重复区	回写	回写	不回写	回写	2
X	1	0	1	递增		回写	回写	回写	回写	3

备注 j=0~23, X: “0”或者“1”

表16-11 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器 (SFR)	扩展特殊功能寄存器 (2ndSFR)	
					无等待	等待
读数据	1	2	4	1	1	1+等待数 ^注
写数据	1	—	—	1	1	1+等待数 ^注

16.5.4 DMA的响应时间

DMA响应时间如表16-12所示。DMA响应时间是指从检测到DMA启动源到开始DMA传送的时间，不包括DMA的执行时钟数。

表16-12 DMA的响应时间

	最短时间	最长时间
响应时间	3个时钟	23个时钟

但是，在以下情况下DMA的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部RAM执行指令的情况最长响应时间：20个时钟

备注 1个时钟： $1/f_{CLK}$ (f_{CLK} : CPU/外围硬件时钟)

16.5.5 DMA的启动源

- 不能在从输入DMA启动源到结束DMA传送的期间输入相同的启动源。
- 在产生DMA启动源的位置，不能操作该启动源对应的DMA启动允许位。
- 如果DMA启动源发送竞争，就在CPU接受DMA传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“16.3.3 向量表”。
- 如果在以下某个状态下允许DMA启动，就开始DMA传送，并且在传送结束后产生中断。因此，必须根据需要在确认比较器的监视标志（CnMON）后置为允许DMA启动。
 - 设置为通过比较器注的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的上升沿产生中断请求（CnEPO=0）而且IVCMP>IVREF（或者内部基准电压1.45V）。
 - 设置为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的下降沿产生中断请求（CnEPO=1）而且IVCMP<IVREF（或者内部基准电压1.45V）。
 (n=0、1)

16.5.6 待机模式中的运行

状态	DMA运行
睡眠模式	能运行（禁止在低功耗RTC模式中运行）。
深度睡眠模式	能接受DMA启动源，并进行DMA传送 ^{注1}

注 1. 在深度睡眠模式中，能在检测到DMA启动源后进行DMA传送，并且在传送结束后返回到深度睡眠模式。但是，因为在深度睡眠模式中代码闪存和数据闪存停止运行，所以不能将闪存设置为传送源。

第17章 联动控制器(EVENTC)

17.1 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。

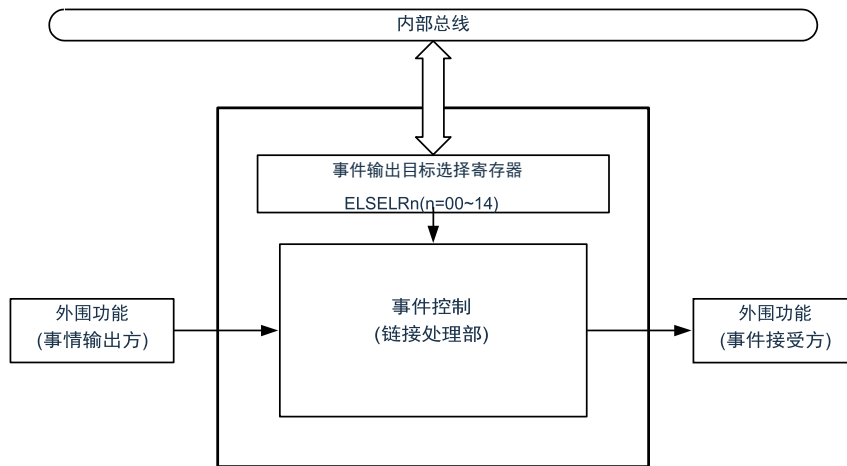
EVENTC有以下功能：

- 根据产品，能将15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3种外围功能中的1种外围功能运行的启动源。

17.2 EVENTC的结构

EVENTC的框图如图17-1所示。

图17-1 EVENTC的框图



17.3 控制寄存器

控制器寄存器如表17-1所示。

表17-1 控制EVENTC的寄存器

寄存器名	符号
事件输出目标选择寄存器00	ELSELR00
事件输出目标选择寄存器01	ELSELR01
事件输出目标选择寄存器02	ELSELR02
事件输出目标选择寄存器03	ELSELR03
事件输出目标选择寄存器04	ELSELR04
事件输出目标选择寄存器05	ELSELR05
事件输出目标选择寄存器06	ELSELR06
事件输出目标选择寄存器07	ELSELR07
事件输出目标选择寄存器08	ELSELR08
事件输出目标选择寄存器09	ELSELR09
事件输出目标选择寄存器10	ELSELR10
事件输出目标选择寄存器11	ELSELR11
事件输出目标选择寄存器12	ELSELR12
事件输出目标选择寄存器13	ELSELR13
事件输出目标选择寄存器14	ELSELR14

17.3.1 输出目标选择寄存器n (ELSELRn) (n=00~14)

ELSELRn寄存器将各事件信号链接到事件接受方外围功能（链接目标外围功能）接受事件时的运行。不能将多个事件输入链接到相同的事件输出目标（事件接受方）。否则，事件接受方外围功能的运行可能不定而无法正常地接受事件信号。另外，不能将事件链接发生源和事件输出目标设置为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设置ELSELRn寄存器。

ELSELRn寄存器（n=00~14）和外围功能的对应如表17-2所示，ELSELRn寄存器（n=00~14）的设置值和链接目标外围功能接受事件时的运行的对应如

表17-3所示。

图17-2 事件输出目标选择寄存器n (ELSELRn) 的格式

地址：40043400H (ELSELR00) ~4004340EH (ELSELR14) 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	0	ELSELn1	ELSELn0

ELSELn1	ELSELn0	事件链接的选择
0	0	禁止事件链接。
0	1	选择所链接的外围功能1的运行注1。
1	0	选择所链接的外围功能2的运行注1。
1	1	选择所链接的外围功能3的运行注1。

注1.请参照“

表17-3 ELSELRn寄存器（n=00~14）的设置值和链接目标外围功能接受事件时的运行的对应”。

表17-2 ELSELRn寄存器 (n=00~14) 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
ELSELR13	比较器检测0	INTCMP0
ELSELR14	比较器检测1	INTCMP1

表17-3 ELSELRn寄存器 (n=00~14) 的设置值和链接目标外围功能接受事件时的运行的对应

ELSELRn寄存器的 ELSELn1~ELSELn0位	链接目标 No.	链接目标外围功能	接受事件时的运行
01B	1	A/D转换器	开始A/D转换。
10B	2	Timer40通道0的 定时器输入 ^{注1}	延迟计数器、输入脉冲间隔的测量、 外部事件计数器
11B	3	Timer40通道1的 定时器输入 ^{注2}	延迟计数器、输入脉冲间隔的测量、 外部事件计数器

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的运行时钟设置为 f_{CLK} ,通过噪声滤波器允许寄存器1 (NFEN1) 将TI00引脚的噪声滤波器置为OFF (TNFEN0=0),并且通过定时器输入选择寄存器0 (TIS0) 将通道0使用的定时器输入设置为联动控制器的事件输入信号。

2.要选择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的运行时钟设置为 f_{CLK} ,通过噪声滤波器允许寄存器1 (NFEN1) 将TI01引脚的噪声滤波器置为OFF (TNFEN01=0),并且通过定时器输入选择寄存器0 (TIS0) 将通道1使用的定时器输入设置为EVENTC的事件输入信号。

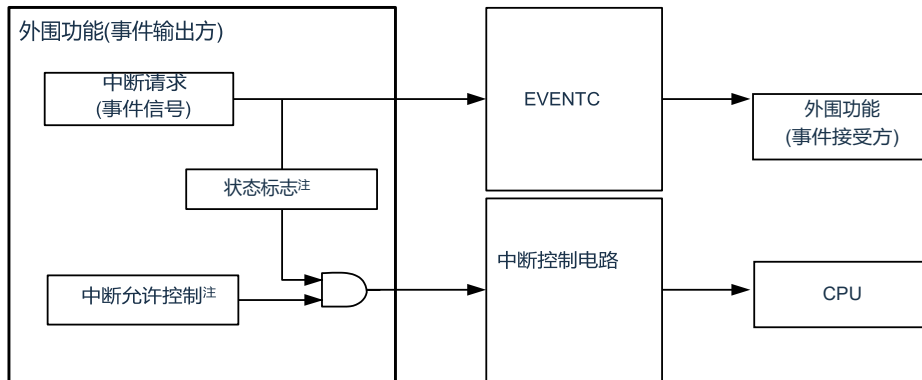
17.4 EVENTC的运行

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17-3所示。此图是以有中断请求状态标志和中断允许位（控制允许或者禁止）的外围功能为例的关系。

通过EVENTC接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行（参照“表17-3ELSELRn寄存器（n=00~14）的设置值和链接目标外围功能接受事件时的运行的对应”）。

图17-3 中断处理和EVENTC的关系



注 有些外围功能没有此功能。

接受事件的外围功能的响应如表17-4所示。

表17-4 接受事件的外围功能的响应

事件接受目标No.	事件链接目标的功能	事件接受后的运行	响应
1	A/D转换器	A/D转换	EVENTC事件直接变为A/D转换的硬件触发。
2	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_{CLK} 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_{CLK} 周期后进行边沿的检测。

第18章 中断功能

Cortex-M0+处理器内置了嵌套向量中断控制器(NVIC),支持最多32个中断请求(IRQ)输入,以及1个不可屏蔽中断(NMI)输入,另外,处理器还支持多个内部异常。

本系统中对32个中断请求(IRQ)输入和1个不可屏蔽中断(NMI)输入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。

18.1 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。

18.2 中断源和结构

中断源列表参照表18-1 中断源一览表(1/3)

表18-1 中断源一览表(1/3)

中断处理	中断源 编号	中断源		内部/外部	基本结构 类型注1	24引脚	20引脚	16引脚
		名称	触发					
可屏蔽	0	INTLVI	电压检测注2	内部	(A)	○	○	○
	1	INTP0	引脚输入边沿的检测	外部	(B)	○	○	○
	2	INTP1	引脚输入边沿的检测			○	○	○
	3	INTP2	引脚输入边沿的检测			○	○	○
	4	INTP3	引脚输入边沿的检测			○	○	○
	5	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	INTTM03H	定时器通道03的计数结束或者捕捉结束 (高8位定时器工作时)			○	○	○
	7	INTST2/ INTSSPI20	UART2发送的传送结束或者缓冲器空中断/SSPI20的传送结束或者缓冲器空中断			○	○	○
	8	INTSR2/ INTSSPI21	UART2接收的传送结束/SSPI21的传送结束或者缓冲器空中断			○	○	○
	9	INTSRE2	发生UART2接收的通信错误			○	○	○
	10	INTST0/ INTSSPI00	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			○	○	○
	11	INTSR0/ INTSSPI01	UART0接收的传送结束/SSPI01的传送结束或者缓冲器空中断			○	○	○
12	INTSRE0	发生UART0接收的通信错误	○			○	○	

注1.基本构成类型(A)~(C)分别对应图18-1 中断功能的基本结构的(A)~(C)。

2.这是将电压检测电平寄存器(LVIS)的bit7(LVIMD)置“0”的情况。

表18-1 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 ^{注1}	24引脚	20引脚	16引脚
		名称	触发					
可屏蔽	13	INTST1/ INTSSPI10	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	内部	(A)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	14	INTSR1/ INTSSPI11	UART1接收的传送结束 /SSPI11的传送结束或者缓 冲器空中断			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	15	INTSRE1	发生UART1接收的通信错误			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	16	INTIICA0	IICA0通信结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	17	INTTM00	定时器通道00的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	18	INTTM01	定时器通道01的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	19	INTTM02	定时器通道02的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	20	INTTM03	定时器通道03的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	21	INTAD	A/D转换结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	22	INTRTC	实时时钟的固定周期/ 闹钟一致检测			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	23	INTIT	间隔信号的检测			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	24	INTCMP0	比较器检测0			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	25	INTCMP1	比较器检测1			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	26	保留				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	27	INTTM10	定时器通道10的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	28	INTTM11	定时器通道11的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	29	INTTM12	定时器通道12的计数结束 或者捕捉结束			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	INTTM13	定时器通道13的计数结束 或者捕捉结束	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
31	INTFL	Flash编程終了	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

注 1.基本构成类型(A)~(C)分别对应图18-1 中断功能的基本结构的(A)~(C)。

表18-1 中断源一览表(3/3)

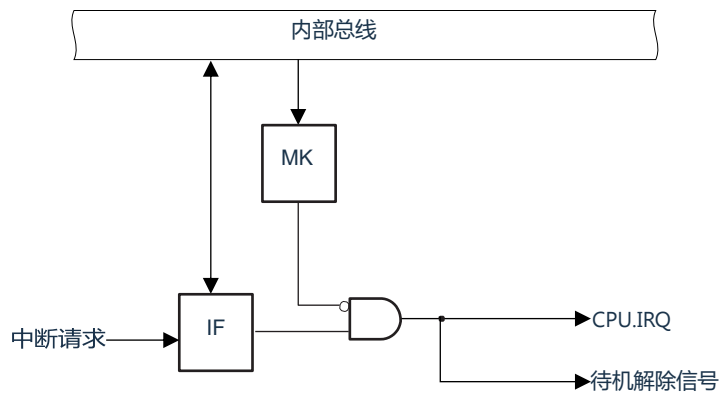
中断处理	中断源编号	中断源		内部/外部	基本结构类型注1
		名称	触发		
不可屏蔽	—	INTWDT	看门狗定时器间隔中断注2	内部	(C)

注1.基本构成类型(A)~(C)分别对应图18-1 中断功能的基本结构的(A)~(C)。

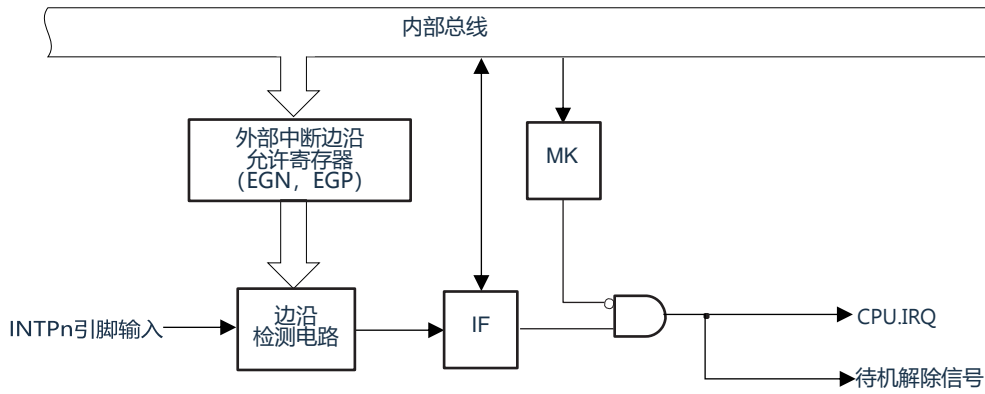
2.这是将选项字节(000C0H)的bit7(WDTINT)置“1”的情况。

图18-1 中断功能的基本结构

(A)内部可屏蔽中断

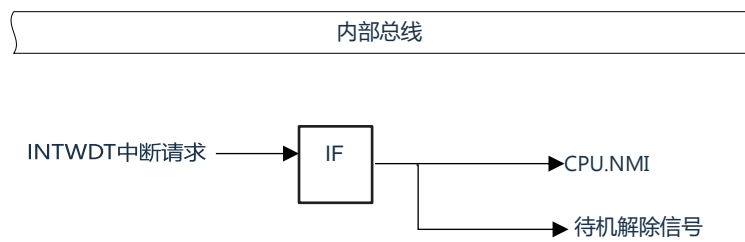


(B)外部可屏蔽中断 (INTPn)



注 n=0~3

(C)不可屏蔽中断



注：不可屏蔽中断的中断请求标志IF没有实体寄存器，不能通过总线读写寄存器来产生中断请求。

18.3 控制中断功能的寄存器

通过以下4种寄存器控制中断功能。

- 中断请求标志寄存器 (IF00~IF31)
- 中断屏蔽标志寄存器 (MK00~MK31)
- 外部中断上升沿允许寄存器 (EGP0)
- 外部中断下降沿允许寄存器 (EGN0)

18.3.1 中断请求标志寄存器 (IF00~IF31)

通过发生对应的中断请求或者执行指令，将中断请求标志置“1”。

通过产生复位信号或者执行指令，将中断请求标志清“0”。

通过8位存储器操作指令设置IF00L~IF31L寄存器

或通过32位存储器操作指令设置IF00~IF31寄存器。

在产生复位信号后，这些寄存器的值变为“0000_0000H”。

图18-2 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
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18-2.

2.中断请求标志寄存器与CPU.IRQ的对应关系见图18-4.

18.3.2 中断屏蔽标志寄存器 (MK00~MK31)

中断屏蔽标志设置允许或者禁止对应的可屏蔽中断处理。
 通过8位存储器操作指令设置MK00L~MK31L寄存器或
 通过32位存储器操作指令设置MK00~MK31寄存器。
 在产生复位信号后，这些寄存器的值变为“FFFF_FFFF”。

图18-3 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

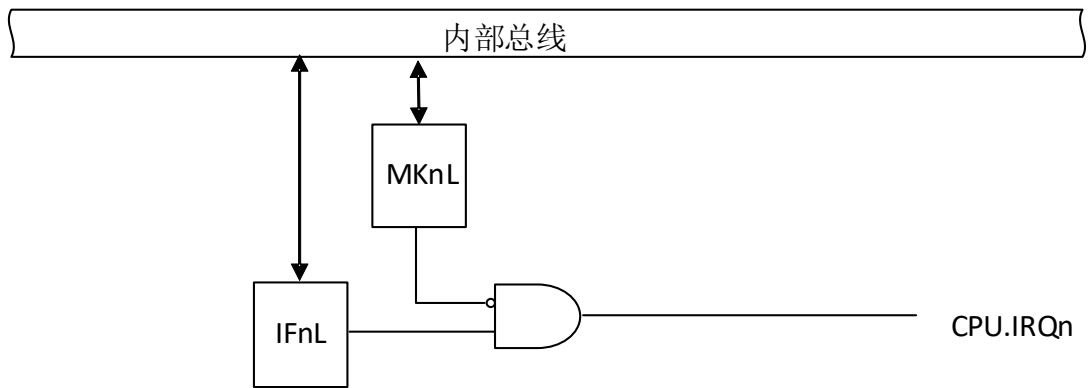
MKmL	编号0~31的中断源的中断处理控制 ^{注1}
0	允许中断处理。
1	禁止中断处理。

注: 1. 中断源与中断请求屏蔽寄存器的对应关系见表18-2
 2. 中断请求屏蔽寄存器与CPU.IRQ的对应关系见图18-4.

表18-2 中断源和各标志寄存器的对应关系

编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器
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	INTTM03H	IF06.IFL	MK06.MKL
7	INTST2/INTSSPI20	IF07.IFL	MK07.MKL
8	INTSR2/INTSSPI21	IF08.IFL	MK08.MKL
9	INTSRE2	IF09.IFL	MK09.MKL
10	INTST0/INTSSPI00	IF10.IFL	MK10.MKL
11	INTSR0/INTSSPI01	IF11.IFL	MK11.MKL
12	INTSRE0	IF12.IFL	MK12.MKL
13	INTST1/INTSSPI10	IF13.IFL	MK13.MKL
14	INTSR1/INTSSPI11	IF14.IFL	MK14.MKL
15	INTSRE1	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	INTCMP0	IF24.IFL	MK24.MKL
25	INTCMP1	IF25.IFL	MK25.MKL
26	INTRAMPRTERR	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18-4 标志寄存器与CPU.IRQ的关系



18.3.3 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)

这些寄存器设置INTP0~INTP3的有效边沿。
 通过8位存储器操作指令设置EGP0、EGN0寄存器。
 在产生复位信号后，这些寄存器的值变为“00H”。

图18-5 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18-3所示。

表18-3 对应EGPn位和EGNn位的中断请求信号

检测允许位		中断请求信号	24引脚	20引脚	16引脚
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配置，请参照“第2章 引脚功能 2.3.7 端口复用功能配置寄存器 (PxxCFG)”。

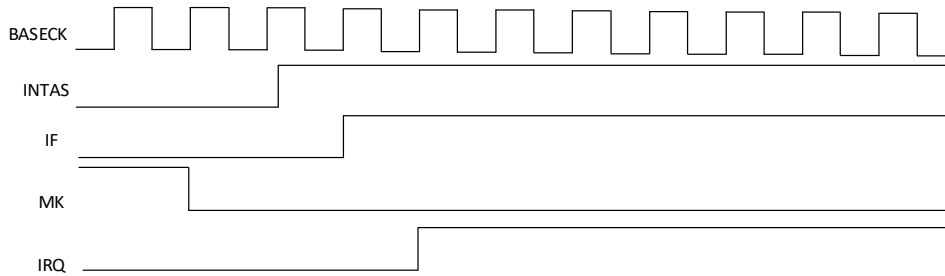
2.n=0~3

18.4 中断处理的操作

18.4.1 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。

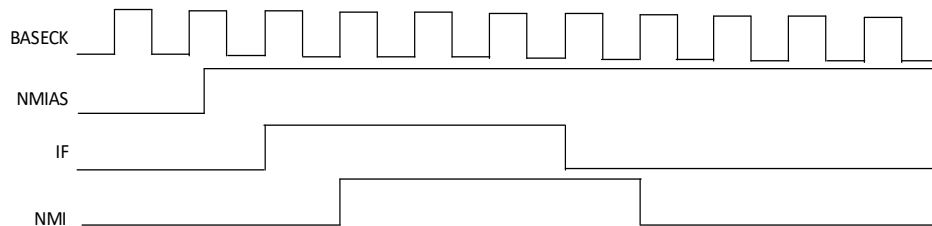
从中断请求标志被置1，到CPU的IRQ被置1，只需要1个时钟。



18.4.2 不可屏蔽中断请求的接受

如果产生不可屏蔽中断请求，中断请求标志将会被置“1”，并直接传递给NVIC。

从中断请求标志被置1，到CPU的NMI被置1，只需要1个时钟。



第19章 待机功能

19.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下2种模式。

(1) 睡眠模式

睡眠模式是停止CPU运行时钟的模式。在设置睡眠模式前，如果高速系统时钟振荡电路、高速内部振荡器或者副系统时钟振荡电路正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到深度睡眠模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

(2) 深度睡眠模式

深度睡眠模式是停止高速系统时钟振荡电路和高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小CPU的工作电流。

因为深度睡眠模式能通过中断请求来解除，所以也能进行间歇运行。但是，在X1时钟的情况下，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设置为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注意1.只有在CPU以主系统时钟运行时才能使用深度睡眠模式。当CPU以副系统时钟运行时，不能设置为深度睡眠模式。

无论CPU是以主系统时钟还是以副系统时钟运行，都能使用睡眠模式。

2.在转移到深度睡眠模式时，必须在停止以主系统时钟运行的外围硬件后执行WFI指令。

3.为了减小A/D转换器的工作电流，必须将A/D转换器模式寄存器0（ADM0）的bit7（ADCS）和bit0（ADCE）清“0”，在停止A/D转换运行后执行WFI指令。

4.能通过选项字节选择在睡眠模式或者深度睡眠模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“[第25章 选项字节](#)”。

19.2 睡眠模式

19.2.1 睡眠模式的设置

在SCR寄存器的SLEEPDEEP位为0时，执行WFI指令，就进入了睡眠模式。在睡眠模式，CPU停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，发振器等在睡眠模式下的状态见表19-1。

无论设置前的CPU时钟是高速系统时钟还是高速内部振荡器时钟或者副系统时钟，都能设置睡眠模式。

注意 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除睡眠模式。因此，即使在此情况下执行WFI指令，也不转移到睡眠模式。

表19-1 睡眠模式中的运行状态(1/2)

睡眠模式的设置		在CPU以主系统时钟运行的过程中执行WFI指令的情况		
		CPU以高速内部振荡器时钟 (f_{IH}) 运行	CPU以X1时钟 (f_X) 运行	CPU以外部主系统时钟 (f_{EX}) 运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	f_{IH}	继续运行 (不能停止)。	禁止运行。	
	f_X	禁止运行。	继续运行 (不能停止)。	不能运行。
副系统时钟	f_{EX}		不能运行。	继续运行 (不能停止)。
	f_{XT}	保持设置为睡眠模式前的状态。		
f_{EXS}				
低速内部振荡器时钟	f_{IL}	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设置。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存		停止运行。		
RAM		停止运行 (在执行DMA时, 能运行)。		
端口 (锁存器)		保持设置为睡眠模式前的状态。		
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9章 看门狗定时器”。		
时钟输出/蜂鸣器输出		能运行。		
A/D转换器				
比较器				
通用串行通信单元 (SCI)				
串行接口 (IICA)				
数据传送控制器 (DMA)				
联动控制器		能在可运行的功能块之间进行链接。		
上电复位功能		能运行。		
电压检测功能				
外部中断				
CRC运算功能	高速CRC	在RAM区的运算中执行DMA时, 能运行。		
	通用CRC			
RAM奇偶校验功能		在执行DMA时, 能运行。		
SFR保护功能				

备注 停止运行: 在转移到睡眠模式时自动停止运行。

禁止运行: 在转移到睡眠模式前停止运行。

f_{IH} : 高速内部振荡器时钟 f_{IL} : 低速内部振荡器时钟

f_X : X1时钟 f_{EX} : 外部主系统时

f_{XT} : XT1时钟 f_{EXS} : 外部副系统

表19-1 睡眠模式中的运行状态(2/2)

睡眠模式的设置		在CPU以副系统时钟运行的过程中执行WFI指令的情况	
		CPU以XT1时钟 (f_{XT}) 运行	CPU以外部副系统时钟 (f_{EXS}) 运行
系统时钟		停止给CPU提供时钟。	
主系统时钟	f_{IH}	禁止运行。	
	f_X		
	f_{EX}		
副系统时钟	f_{XT}	继续运行 (不能停止)。	不能运行。
	f_{EXS}	不能运行。	继续运行 (不能停止)。
低速内部振荡器时钟	f_{IL}	通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设置。 <ul style="list-style-type: none"> • WUTMMCK0=1: 振荡 • WUTMMCK0=0并且WDTON=0: 停止 • WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 • WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止 	
CPU		停止运行。	
代码闪存			
RAM		停止运行 (在执行DMA时, 能运行)。	
端口 (锁存器)		保持设置为睡眠模式前的状态。	
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			
看门狗定时器		参照“第9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。	
比较器		能运行。	
通用串行通信单元 (SCI)		当RTCLPC=0时, 能运行 (否则禁止运行)。	
串行接口 (IICA)		禁止运行。	
数据传送控制器 (DMA)		当RTCLPC=0时, 能运行 (否则禁止运行)。	
联动控制器		能在可运行的功能块之间进行链接。	
上电复位功能		能运行。	
电压检测功能			
外部中断			
CRC运算功能	高速CRC	禁止运行。	
	通用CRC	在RAM区的运算中执行DMA时, 能运行。	
RAM奇偶校验错误检测功能		在执行DMA时, 能运行。	
SFR保护功能			

备注 停止运行: 在转移到睡眠模式时自动停止运行。

禁止运行: 在转移到睡眠模式前停止运行。

f_{IH} : 高速内部振荡器时钟 f_{IL} : 低速内部振荡器时钟
 f_X : X1时钟 f_{EX} : 外部主系统时
 f_{XT} : XT1时钟 f_{EXS} : 外部副系统

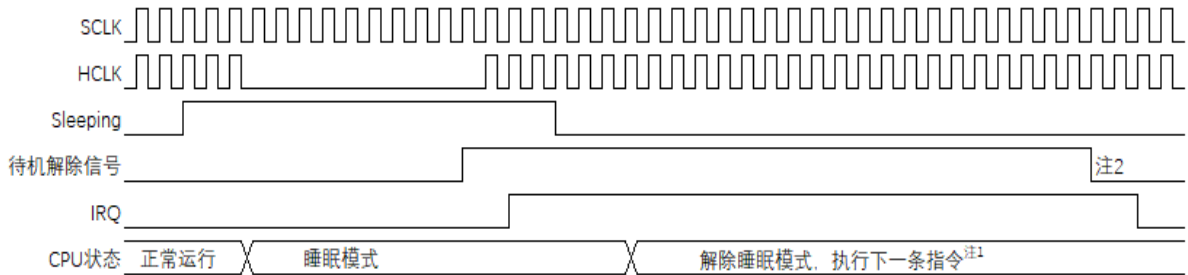
19.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位端子，POR复位，低电压检测复位，RAM奇偶校验错误复位，WDT复位，软件复位解除。

(1) 通过中断解除

当产生一个未屏蔽的中断，且处于允许接受中断的状态时，睡眠模式就被解除，CPU开始处理中断服务程序。

图19-1 过中断请求解除睡眠模式



注1. 从待机解除信号产生到睡眠模式解除，开始执行中断服务程序，需要16个时钟。

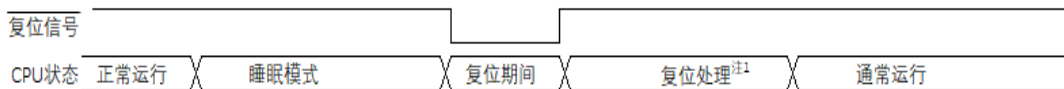
2. 待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

注意：进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。

(2) 通过复位解除

当有复位信号产生时，CPU处于复位状态，睡眠模式被解除。和通常的复位相同，在转移到复位向量地址后执行程序。

图19-2 过复位解除睡眠模式



注1：有关复位处理时间，请参照“第20章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第21章 上电复位电路”。

19.3 深度睡眠模式

19.3.1 深度睡眠模式的设置

在SCR寄存器的SLEEPDEEP位为1时，执行WFI指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及发振器都停止运行。但是，CPU内部寄存器的值，RAM数据，周边模块，I/O的状态被保持。周边模块，发振器在深度睡眠模式的运行状态见表19-2。

只有在设置前的CPU时钟为主系统时钟的情况下才能设置深度睡眠模式。

注意 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行WFI指令，就在一旦进入深度睡眠模式后立即被解除。在执行WFI指令并且经过深度睡眠模式解除时间后返回到运行模式。

表19-2 深度睡眠模式中的运行状态

深度睡眠模式的设置		在CPU以主系统时钟运行的过程中执行WFI指令的情况		
		CPU以高速内部振荡器时钟 (f_{IH}) 运行	CPU以X1时钟 (f_X) 运行	CPU以外部主系统时钟 (f_{EX}) 运行
系统时钟		停止给CPU提供时钟。		
主系统时钟	f_{IH}	停止		
	f_X			
f_{EX}				
副系统时钟	f_{XT}	保持设置为深度睡眠模式前的状态。		
	f_{EXS}			
f_{IL}		通过选项字节 (000C0H) 的bit0 (WDSTBYON) 和bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的WUTMMCK0位进行设置。 WUTMMCK0=1: 振荡 WUTMMCK0=0并且WDTON=0: 停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1: 振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0: 停止		
CPU		停止运行。		
代码闪存		停止运行。		
RAM				
端口 (锁存器)				
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		参照“第9章 看门狗定时器”。		
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。		
比较器		能运行 (只限于不使用数字滤波器的情况)。		
通用串行通信单元 (SCI)		只有SSPIp和UARTq才能唤醒。 除了SSPIp和UARTq以外, 禁止运行。		
串行接口 (IICA)		能通过地址匹配进行唤醒。		
数据传送控制器 (DMA)		能接受DMA启动源。		
联动控制器		能在可运行的功能块之间进行链接。		
上电复位功能		能运行。		
电压检测功能				
外部中断				
CRC运算功能	高速CRC	停止运行。		
	通用CRC			
RAM奇偶校验功能		停止运行。		
SFR保护功能				

备注1.停止运行: 在转移到深度睡眠模式时自动停止运行。

禁止运行: 在转移到深度睡眠模式前停止运行。

f_{IH} : 高速内部振荡器时钟

f_{IL} : 低速内部振荡器时钟

f_X : X1时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1时钟

f_{EXS} : 外部副系统时钟

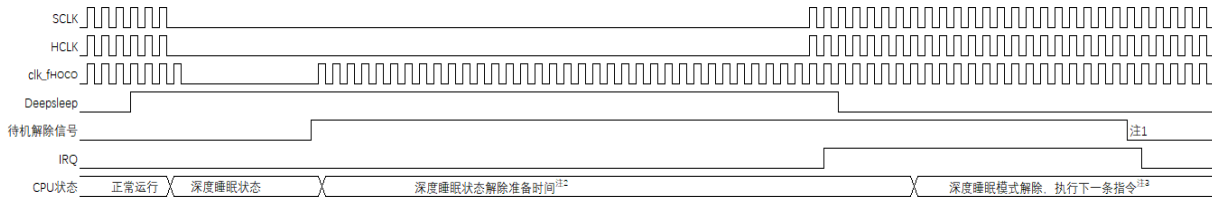
19.3.2 深度睡眠模式的解除

能通过以下2种方法解除深度睡眠模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除深度睡眠模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图19-3 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“第18章 中断功能 图18-1 中断功能的基本结构”。

2.深度睡眠状态解除准备时间：

进入深度睡眠模式前CPU时钟为高速内部振荡时钟或外部时钟输入时：

20us~40us

进入深度睡眠模式前CPU时钟为高速系统时钟(X1振荡)时：

20us~“40us和振荡稳定时间(通过OSTS进行设置)中较长的时间”

3.等待：从CPU.IRQ有效到开始执行中断服务程序，需要14个时钟。

注意：1.进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。

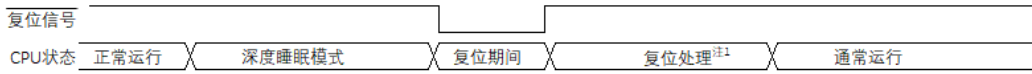
2.在CPU以高速系统时钟（X1振荡）运行并且要缩短深度睡眠模式解除后的振荡稳定时间时，必须在执行WFI指令前暂时将CPU时钟切换为高速内部振荡器时钟。

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

(b) 通过产生复位信号进行的解除

通过产生复位信号来解除深度睡眠模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图19-4 通过复位解除深度睡眠模式



注 有关复位处理时间，请参照“第20章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第21章 上电复位电路”。

第20章 复位功能

以下7种方法产生复位信号。

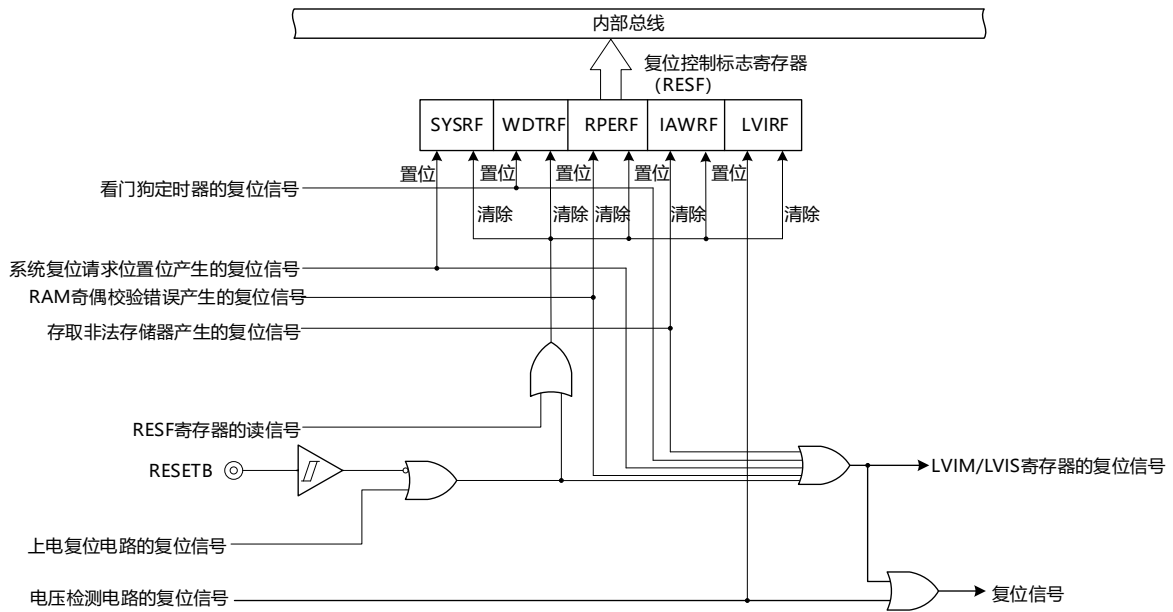
- (1)通过RESETB引脚输入外部复位。
- (2)通过看门狗定时器的程序失控检测产生内部复位。
- (3)通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4)通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5)因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为1而产生内部复位。
- (6)因RAM奇偶校验错误而产生内部复位。
- (7)因存取非法存储器而产生内部复位。

内部复位和外部复位相同，在产生复位信号后，从用户自定义的程序起始地址开始执行程序。

当给RESETB引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到POR电路和LVD电路的电压，或者系统复位请求位被置位，或者发生RAM奇偶检验错误，或者存取非法存储器时，产生复位并且各硬件变为如表20-1所示的状态。

- 注意1.在进行外部复位时，必须至少给RESETB引脚输入10us的低电平。如果在电源电压上升时进行外部复位，就必须在给RESETB引脚输入低电平后接通电源，而且在datasheet的AC特性所示的工作电压范围内至少保持10us的低电平，然后输入高电平。
- 2.在复位信号发生期间，停止X1时钟、XT1时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟和外部副系统时钟的输入无效。
 - 3.如果发生复位，就对各SFR进行初始化，因此端口引脚变为以下状态：
 - P20,P21,P35,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)。
 - P20,P21,P35,P36,P37以外的端口：在复位期间以及接受复位后为高阻抗。

图20-1 复位功能的框图



注意 LVD电路的内部复位不会对LVD电路进行复位。

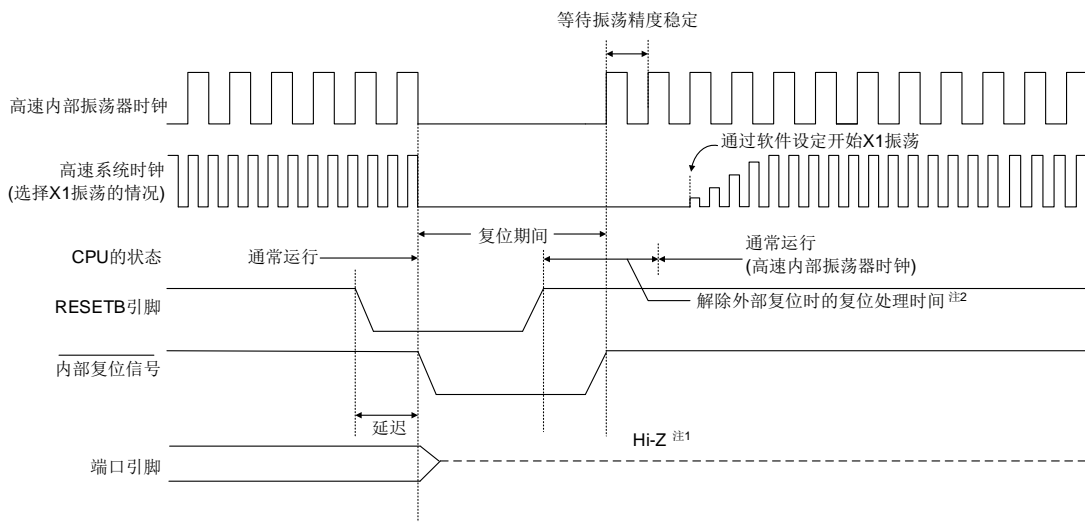
备注1.LVIM: 电压检测寄存器

2.LVIS: 电压检测电平寄存器

复位时序

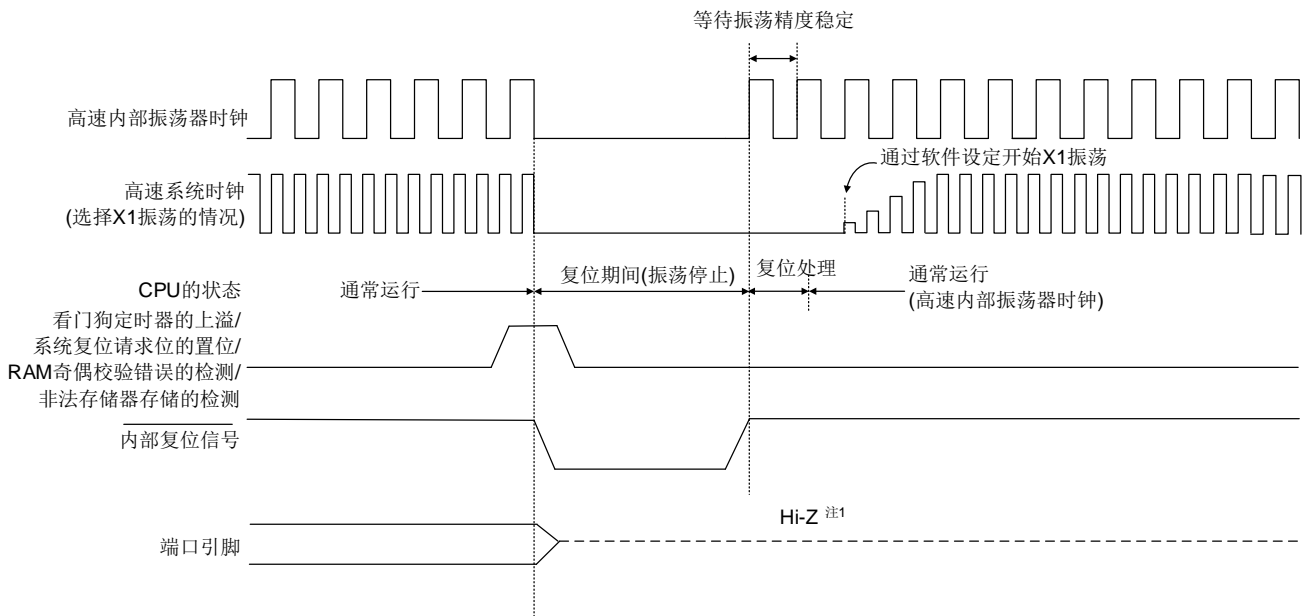
当给RESETB引脚输入低电平时，产生复位。然后，如果给RESETB引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图20-2 RESETB输入的复位时序



对于因看门狗定时器的上溢、系统复位请求位的置位、RAM奇偶校验错误的检测或者非法存储器存取的检测而产生的复位，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20-3 因看门狗定时器的上溢、系统复位请求位的置位、RAM奇偶校验错误的检测或者非法存储器存取的检测而产生的复位时序



注1. 端口引脚P20,P21,P35,P36,P37变为以下状态:

- 在外部复位或者POR复位的期间为高阻抗。
- 在其他复位期间以及接受复位后为高电平 (连接内部上拉电阻)

注意 看门狗定时器也不例外, 在发生内部复位时进行复位。

对于由POR电路和LVD电路的电压检测产生的复位, 如果在复位后满足 $V_{DD} \geq V_{POR}$ 或者 $V_{DD} \geq V_{LVD}$, 就解除复位状态, 并且在复位处理后以高速内部振荡器时钟开始执行程序。详细内容请参照“第21章 上电复位电路”和“第22章 电压检测电路”。

备注 V_{POR} : POR电源电压上升检测电压
 V_{LVD} : LVD检测电压

表20-1 复位期间的运行状态

项目		复位期间
系统时钟		停止给CPU提供时钟。
主系统时钟	f_{IH}	停止运行。
	f_X	停止运行（X1引脚和X2引脚处于输入端口模式）。
	f_{EX}	时钟输入无效（引脚处于输入端口模式）。
副系统时钟	f_{XT}	能运行。
	f_{EXS}	时钟输入无效（引脚处于输入端口模式）。
f_{IL}		停止运行。
CPU		
代码闪存		停止运行。
RAM		停止运行。
端口（锁存器）		高阻抗 ^{注1}
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		
比较器 ^{注1}		
通用串行通信单元（SCI）		
串行接口（IICA）		
数据传送控制器（DMA）		
上电复位功能		能进行检测运行。
电压检测功能		在LVD复位时，能运行。在其他复位时，停止运行。
外部中断		停止运行。
键中断功能		
CRC运算功能	高速CRC	
	通用CRC	
RAM奇偶校验功能		
SFR保护功能		

注 1. 端口引脚P20,P21,P35,P36,P37变为以下状态：

在外部复位或者POR复位的期间为高阻抗。在其他复位期间为高电平（连接内部上拉电阻）。

备注	f_{IH} : 高速内部振荡器时钟	f_X : X1振荡时钟
	f_{EX} : 外部主系统时钟	f_{XT} : XT1振荡时钟
	f_{EXS} : 外部副系统时钟	f_{IL} : 低速内部振荡器时钟

20.1 确认复位源的寄存器

20.1.1 复位控制标志寄存器 (RESF)

BAT32G133微控制器存在多种内部复位发生源。复位控制标志寄存器 (RESF) 保存发生复位请求的复位源。能通过8位存储器操作指令读RESF寄存器。

通过RESETB的输入、上电复位 (POR) 电路的复位和RESF寄存器的读取, 清除SYSRF、WDTRF、RPERF、IAWRF、LVIRF标志。要判断复位源时, 必须将RESF寄存器的值保存到任意的RAM, 然后通过其RAM值进行判断。

图20-4 复位控制标志寄存器 (RESF) 的格式

地址: 40020440H 复位后: 不定值注¹ R

符号	7	6	5	4	3	2	1	0
RESF	SYSRF	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

SYSRF	系统复位请求位被置位而产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

WDTRF	看门狗定时器 (WDT) 产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

RPERF	RAM奇偶校验错误产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

IAWRF	存取非法存储器产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

LVIRF	电压检测电路 (LVD) 产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

注 1. 因复位源而不同。请参照表20-2。

注意 在允许产生RAM奇偶校验错误复位(RPERDIS=0)的情况下, 当存取数据时, 必须对“所用RAM区”进行初始化; 当从RAM区执行指令时, 必须对“所用RAM区+10字节”的区域进行初始化。通过产生复位, 进入允许产生RAM奇偶校验错误复位 (RPERDIS=0) 的状态。详细内容请参照“23.3.3 RAM奇偶校验错误检测功能”。

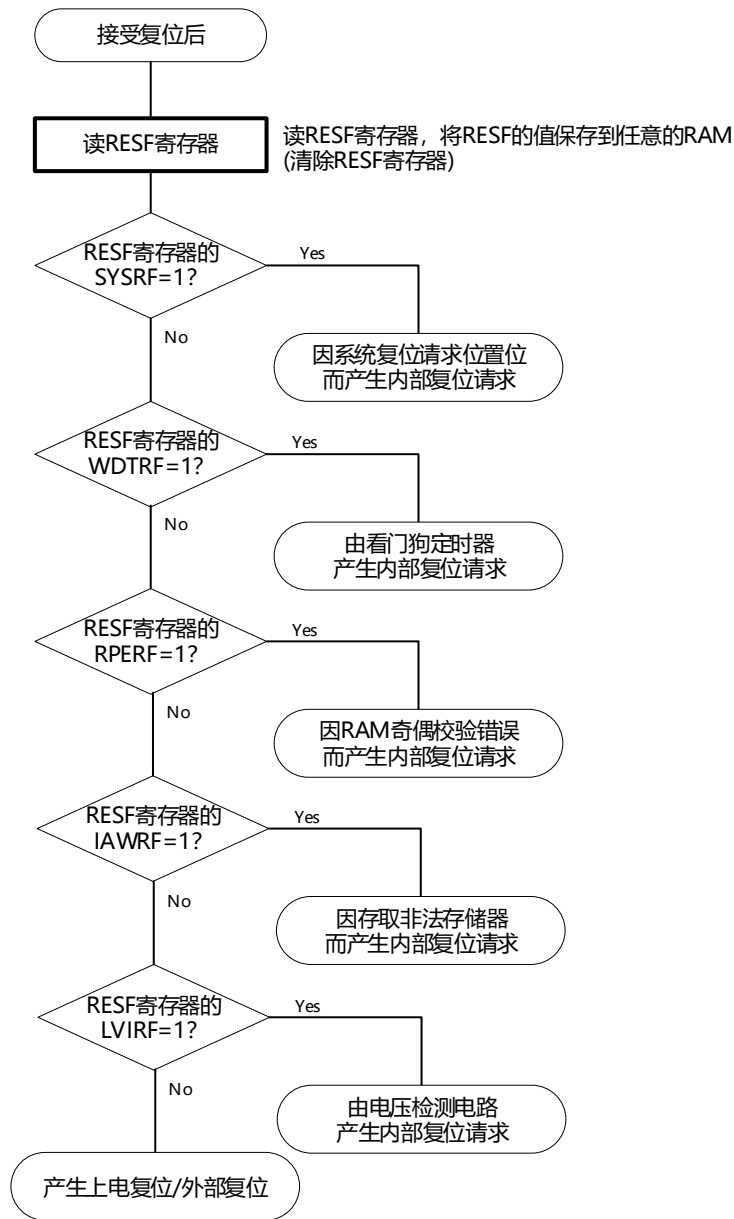
发生复位请求时的RESF寄存器状态如表20-2所示。

表20-2 发生复位请求时的RESF寄存器状态

标志 \ 复位源	RESETB 输入	POR产生 的复位	系统复位请 求位置位产 生的复位	WDT产生的 复位	RAM奇偶 校验错误产 生的复位	存取非法存 储器产生的 复位	LVD产生的 复位
SYSRF	清“0”	清“0”	置“1”	保持	保持	保持	保持
WDTRF			保持	置“1”			
RPERF			保持	置“1”			
IAWRF			保持	置“1”			
LVIRF			保持	置“1”			

复位源的确认步骤如图20-5所示。

图20-5 复位源的确认步骤



注意 上述流程是确认步骤的例子。

第21章 上电复位电路

21.1 上电复位电路的功能

上电复位电路（POR）有以下功能。

- 在接通电源时产生内部复位信号。
如果电源电压（ V_{DD} ）超过检测电压（ V_{POR} ），就解除复位。但是，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态。
- 将电源电压（ V_{DD} ）和检测电压（ V_{PDR} ）进行比较。当 $V_{DD} < V_{PDR}$ 时，产生内部复位信号。但是，当电源电压下降时，必须在电源电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注意 当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

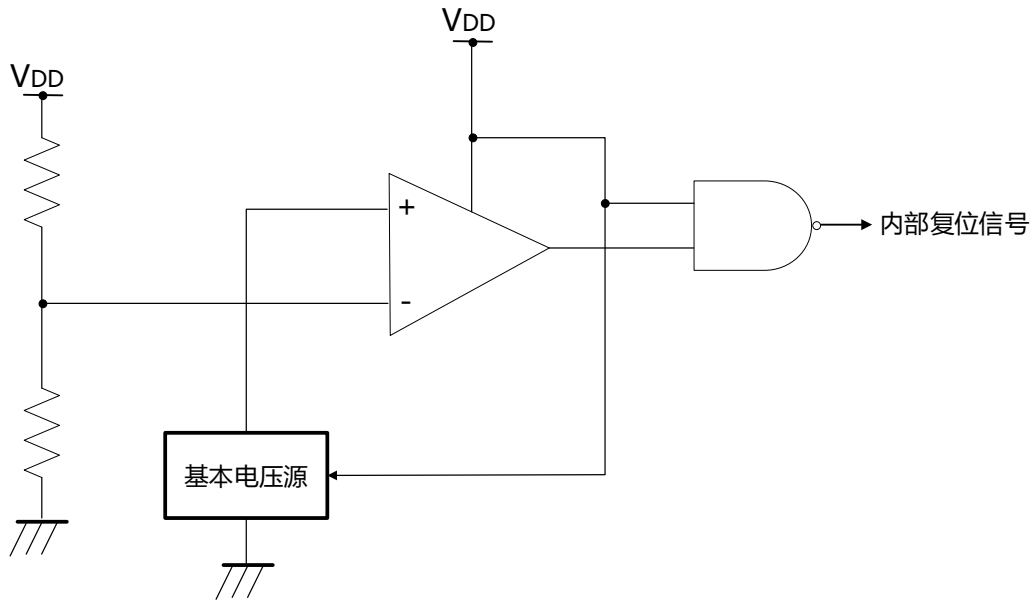
备注1. BAT32G133内置多个产生内部复位信号的硬件。当由看门狗定时器（WDT）、电压检测（LVD）电路、系统复位请求位置位、RAM奇偶校验错误或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在RESF寄存器；当由WDT、LVD、系统复位请求位的置位、RAM奇偶校验错误或者非法存储器的存取而产生内部复位信号时，不将RESF寄存器清“00H”而将标志置“1”。有关RESF寄存器的详细内容，请参照“第20章 复位功能”。

2. V_{POR} : POR电源电压上升检测电压
 V_{PDR} : POR电源电压下降检测电压
详细内容请参照数据手册的POR电路特性。

21.2 上电复位电路的结构

上电复位电路的框图如图21-1所示。

图21-1 上电复位电路的框图

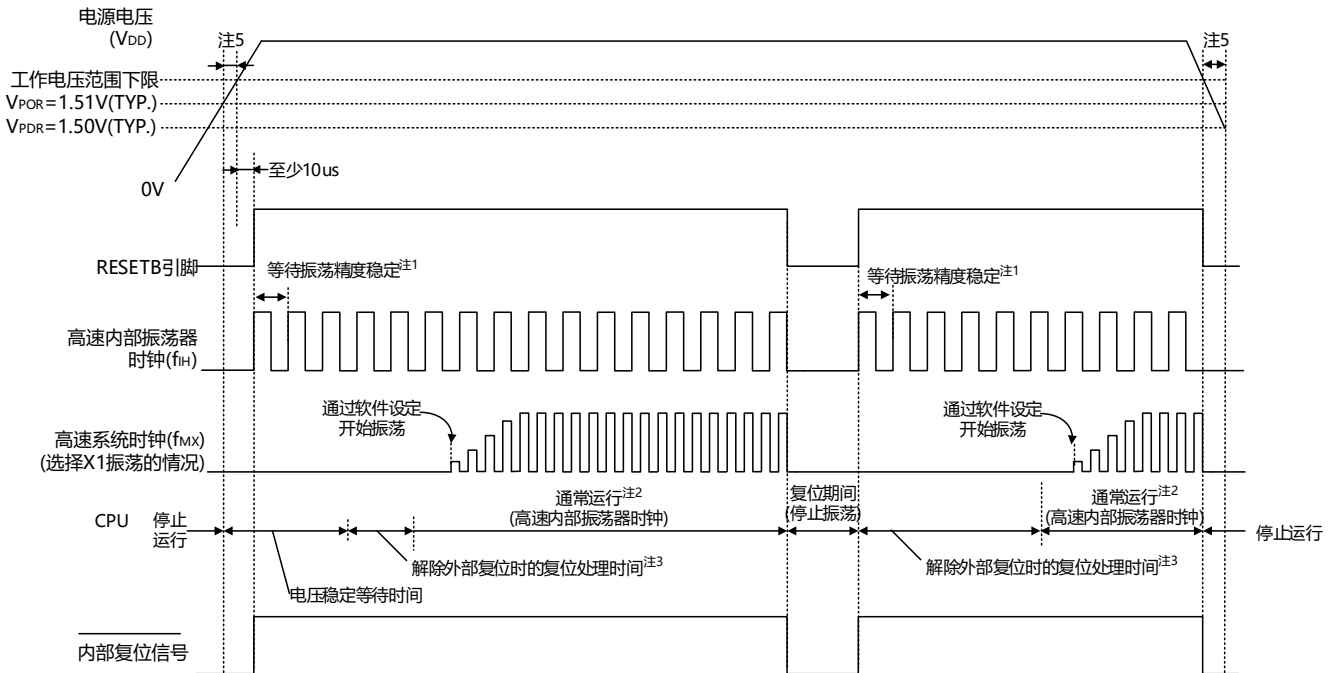


21.3 上电复位电路的运行

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21-2 上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用RESETB引脚的外部复位输入的情况



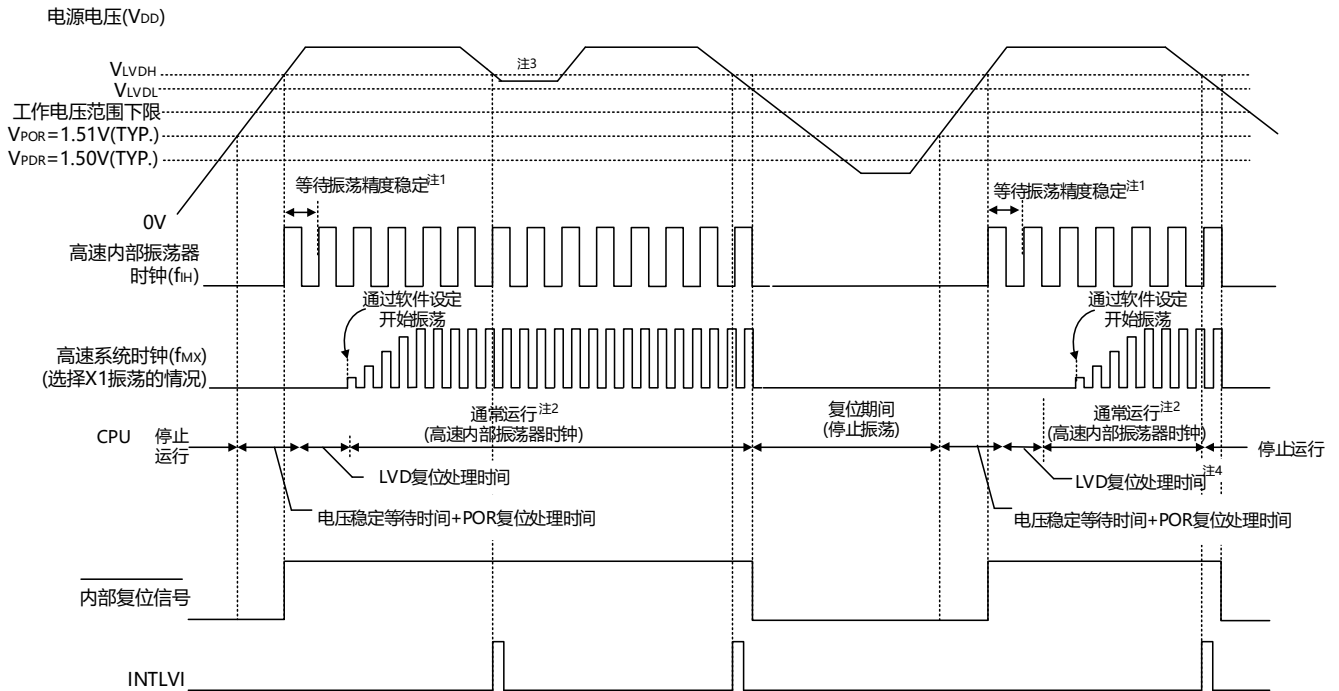
- 注 1.内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
- 2.能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。
- 3.到开始通常运行为止的时间除了达到V_{POR}（1.51V(TYP.))后的“电压稳定等待时间”以外，在将RESETB信号置为高电平（“1”）后还需要以下的“解除外部复位时的复位处理时间（解除POR后的第1次）”。
- 4.当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注 V_{POR}: POR电源电压上升检测电压
V_{PDR}: POR电源电压下降检测电压

注意 在LVD为OFF时，必须使用RESETB引脚的外部复位。详细内容请参照“第22章 电压检测电路”。

图21-2 电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD为中断&复位模式的情况 (选项字节000C1H的LVIMDS1、LVIMDS0=1、0)

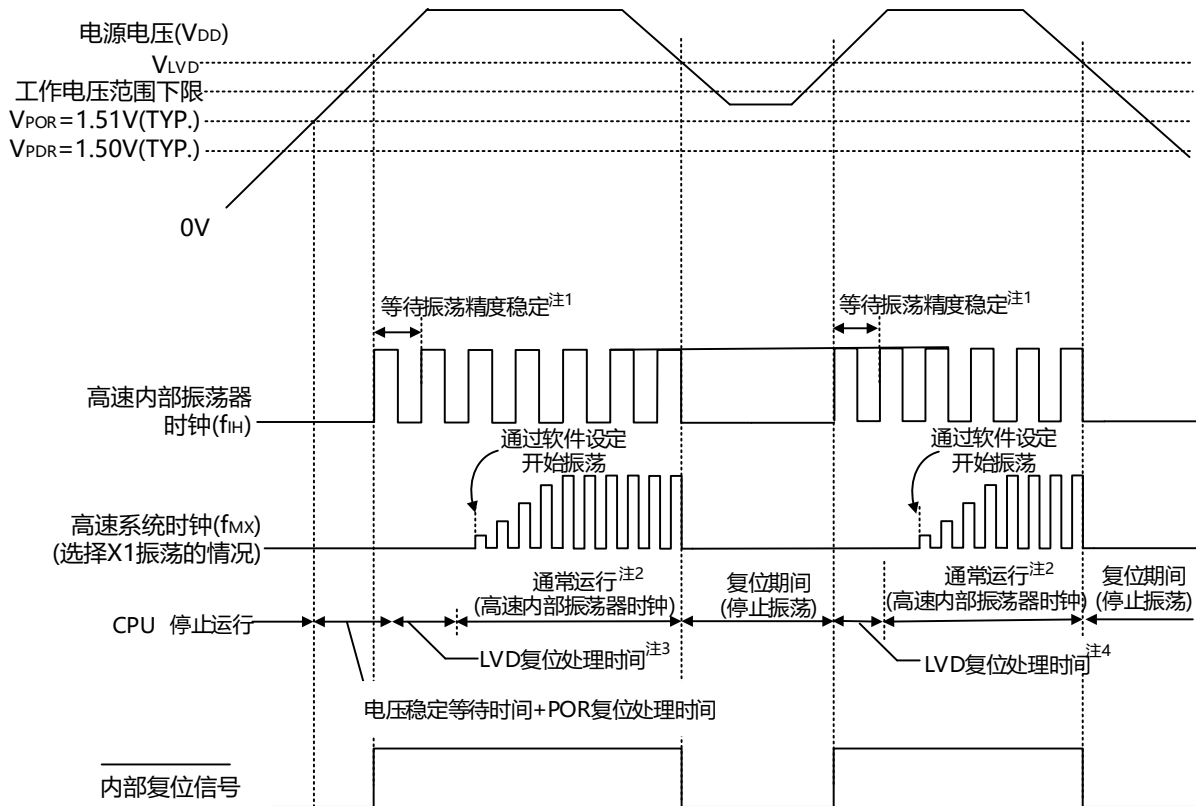


- 注 1.内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
- 2.能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用X1时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用XT1时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。
- 3.在产生中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的LVILV位和LVIMD位置“1”。因此，必须考虑可能出现电源电压在不低于低电压检测电压（V_{LVDL}）的状态下恢复到高电压检测电压（V_{LVDH}）或者更高的情况，在产生INTLVI后按照“图21-8 工作电压的确认/复位的设置步骤”和“图21-9 中断&复位模式的初始设置步骤”进行设置。
- 4.到开始通常运行为止的时间除了达到V_{POR}（1.51V(TYP.))后的“电压稳定等待时间+POR复位处理时间”以外，在达到LVD检测电平（V_{LVDH}）后还需要“LVD复位处理时间”。

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21-2 电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3) LVD复位模式的情况（选项字节000C1H的LVIMDS1、LVIMDS0=1、1）



- 注 1.内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
 2.能将CPU时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。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。
 3.到开始通常运行为止的时间除了达到V_{POR}（1.51V(TYP.))后的“电压稳定等待时间+POR复位处理时间”以外，在达到LVD检测电平（V_{LVD}）后还需要“LVD复位处理时间”。
 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复位处理时间”。

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21-4(3/3)LVD位模式的情况”的“注3”的时间相同。

第22章 电压检测电路

22.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设置运行模式和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）。电压检测（LVD）电路有以下功能。

- 将电源电压（ V_{DD} ）和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（ V_{LVDH} 、 V_{LVDL} ）能通过选项字节选择12种检测电平（参照“第25章 选项字节”）。
- 也能在深度睡眠模式中运行
- 当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设置。

(a) 中断&复位模式（选项字节的LVIMDS1、LVIMDS0=1、0）

通过选项字节000C1H选择2个检测电压（ V_{LVDH} 、 V_{LVDL} ），高电压检测电平（ V_{LVDH} ）用于解除复位或者产生中断，低电压检测电平（ V_{LVDL} ）用于产生复位。

(b) 复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）

将选项字节000C1H选择的1个检测电压（ V_{LVD} ）用于产生或者解除复位。

(c) 中断模式（选项字节的LVIMDS1、LVIMDS0=0、1）

将选项字节000C1H选择的1个检测电压（ V_{LVD} ）用于产生中断或者解除复位。在各模式中，产生以下的中断信号和内部复位信号。

中断&复位模式 (LVIMDS1、LVIMDS0=1、0)	复位模式 (LVIMDS1、LVIMDS0=1、1)	中断模式 (LVIMDS1、LVIMDS0=0、1)
在工作电压下降时，当检测到 $V_{DD} < V_{LVDH}$ 时，产生中断请求信号；当检测到 $V_{DD} < V_{LVDL}$ 时，产生内部复位；当检测到 $V_{DD} \geq V_{LVDH}$ 时，解除内部复位。	当检测到 $V_{DD} \geq V_{LVD}$ 时，解除内部复位；当检测到 $V_{DD} < V_{LVD}$ 时，产生内部复位。	在发生复位后，LVD的内部复位状态继续保持到 $V_{DD} \geq V_{LVD}$ 为止。当检测到 $V_{DD} \geq V_{LVD}$ 时，解除LVD的内部复位。在解除LVD的内部复位后，如果检测到 $V_{DD} < V_{LVD}$ 或者 $V_{DD} \geq V_{LVD}$ 时，就产生中断请求信号（INTLVI）。

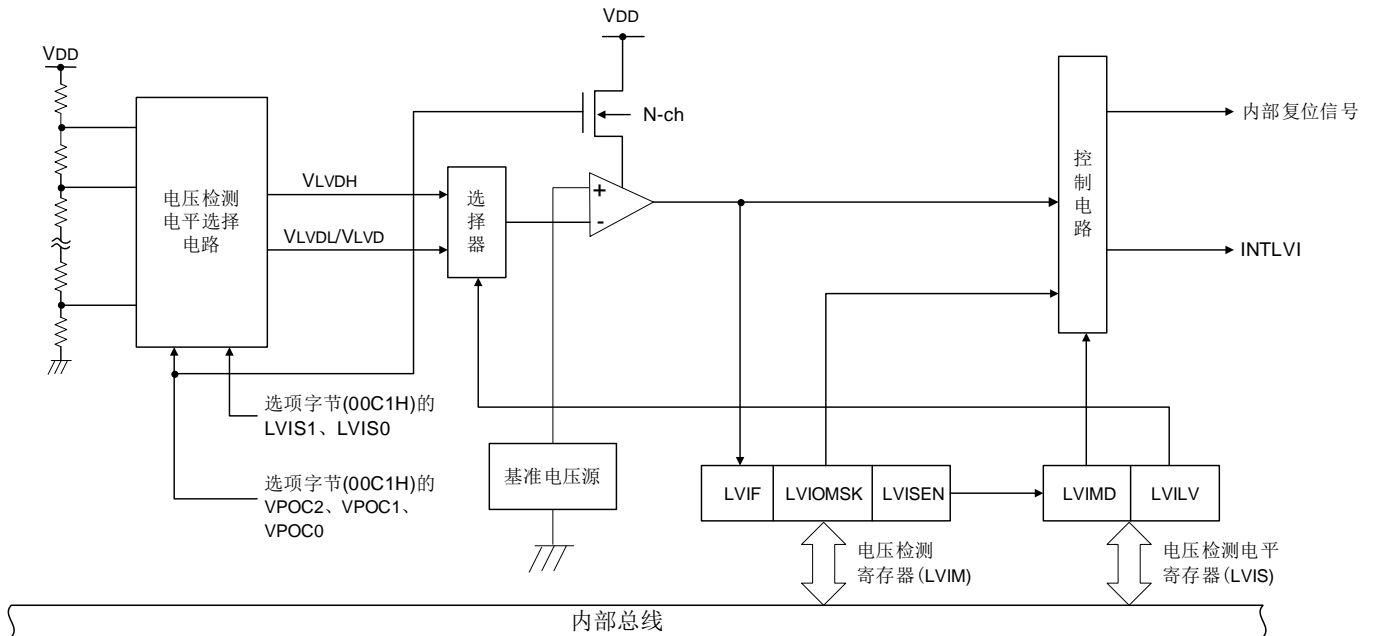
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位，就将复位控制标志寄存器（RESF）的bit0（LVIRF）置“1”。有关RESF寄存器的详细内容，请参照“第20章 复位功能”。

22.2 电压检测电路的结构

电压检测电路的框图如图22-1所示。

图22-1 电压检测电路的框图



22.3 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器 (LVIM)
- 电压检测电平寄存器 (LVIS)

22.3.1 电压检测寄存器 (LVIM)

此寄存器设置允许或者禁止改写电压检测电平寄存器 (LVIS)，并且确认LVD输出的屏蔽状态。通过8位存储器操作指令设置LVIM寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图22-2 电压检测寄存器 (LVIM) 的格式

地址: 40020441H 复位后: 00H^{注1} R/W^{注2}

符号	7	6	5	4	3	2	1	0
LVIM	LVISEN ^{注3}	0	0	0	0	0	LVIOMSK	LVIF

LVISEN ^{注3}	电压检测电平寄存器 (LVIS) 的允许/禁止改写的设置
0	禁止改写LVIS寄存器 (LVIOMSK=0 (LVD输出屏蔽无效))。
1	允许改写LVIS寄存器 (LVIOMSK=1 (LVD输出屏蔽有效))。

LVIOMSK	LVD输出的屏蔽状态标志
0	LVD输出屏蔽无效。
1	LVD输出屏蔽有效 ^{注4} 。

LVIF	电压检测标志
0	电源电压 (V _{DD}) ≥ 检测电压 (V _{LVD}) 或者LVD为OFF。
1	电源电压 (V _{DD}) < 检测电压 (V _{LVD})

- 注 1. 复位值因复位源而变。
 在LVD发生复位时，不对LVIM寄存器的值进行复位而保持原来的值；在其他复位时，将LVISEN清“0”。
2. bit0和bit1是只读位。
3. 只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能设置，在其他模式中不能更改初始值。
4. 只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时，LVIOMSK位才在以下期间自动变为“1”，屏蔽LVD产生的复位或者中断。
- LVISEN=1的期间
 - 从发生LVD中断开始到LVD检测电压稳定为止的等待时间
 - 从更改LVILV位的值到LVD检测电压稳定为止的等待时间

22.3.2 电压检测电平寄存器 (LVIS)

这是设置电压检测电平的寄存器。

通过8位存储器操作指令设置LVIS寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”^{注1}。

图22-3 电压检测电平寄存器 (LVIS) 的格式

地址: 40020442H 复位后: 00H/01H/81H^{注1} R/W

符号	7	6	5	4	3	2	1	0
LVIS	LVIMD ^{注2}	0	0	0	0	0	0	LVILV ^{注2}

LVIMD ^{注2}	电压检测的运行模式
0	中断模式
1	复位模式

LVILV ^{注2}	LVD检测电平
0	高电压检测电平 (V _{LVDH})
1	低电压检测电平 (V _{LVDL} 或者V _{LVD})

- 注 1. 复位值因复位源和选项字节的设置而变。在发生LVD复位时，不将此寄存器清“00H”。
 在发生LVD以外的复位时，此寄存器的值如下：
 •选项字节的LVIMDS1、LVIMDS0=1、0时：00H
 •选项字节的LVIMDS1、LVIMDS0=1、1时：81H
 •选项字节的LVIMDS1、LVIMDS0=0、1时：01H
2. 只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能写“0”。在其他情况下不能设置。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注意1. 要改写LVIS寄存器时，必须按照图22-7和图22-8的步骤进行。

2. 通过选项字节000C1H选择LVD的运行模式和各模式的检测电压 (V_{LVDH}、V_{LVDL}、V_{LVD})。用户选项字节 (000C1H/010C1H) 的格式如表22-1所示。有关选项字节的详细内容，请参照“第25章 选项字节”。

表22-1 用户选项字节（000C1H/010C1H）的格式(1/2)

地址：000C1H/010C1H注

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置（中断&复位模式）

检测电压			选项字节的设置值						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降	下降						LVIMDS1	LVIMDS0
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V		0	1	1	0		
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V		1	0	1	0		
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V	1	1	1	0			
3.02V	2.96V				0	1			
4.06V	3.98V				0	0			
—			禁止设置上述以外的值。						

• LVD的设置（复位模式）

检测电压		选项字节的设置值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	1	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—			禁止设置上述以外的值。					

备注1.检测电压是TYP值。详细内容请参照数据手册的LVD电路特性。

表22-1 用户选项字节（000C1H）的格式(2/2)

地址：000C1H

	7	6	5	4	3	2	1	0
	VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置（中断模式）

检测电压		选项字节的设置值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	0	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设置上述以外的值。						

• LVD为OFF（使用RESETB引脚的外部复位）

检测电压		选项字节的设定值						
V_{LVD}		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，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H）的设置。

备注1.×：忽略

2.检测电压是TYP值。详细内容请参照数据手册的LVD电路特性。

22.4 电压检测电路的运行

22.4.1 用作复位模式时的设置

通过选项字节000C1H设置运行模式（复位模式（LVIMDS1、LVIMDS0=1、1）和检测电压（ V_{LVD} ）。如果设置复位模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器（LVIM）的bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。
bit7（LVIMD）为“1”（复位模式）。
bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。

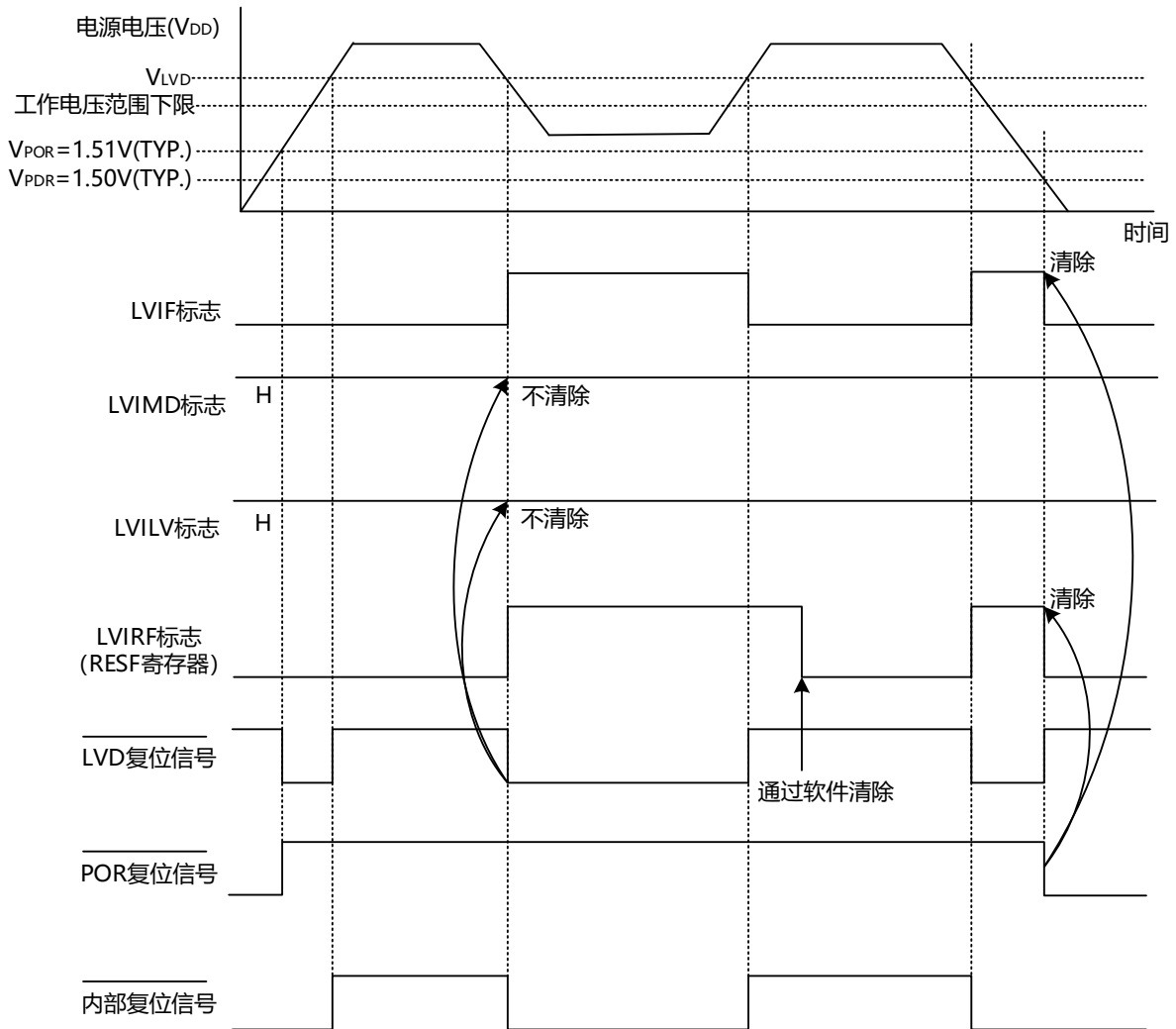
●LVD复位模式的运行

当接通电源时，复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）在电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ）前保持LVD的内部复位状态。如果电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ），就解除内部复位。

当工作电压下降时，如果电源电压（ V_{DD} ）低于电压检测电平（ V_{LVD} ），就产生LVD的内部复位。

LVD复位模式的内部复位信号的产生时序如图22-4所示。

图22-4 内部复位信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、1）



备注 V_{POR}: POR电源电压上升检测电压
 V_{PDR}: POR电源电压下降检测电压

22.4.2 用作中断模式时的设置

通过选项字节000C1H设置运行模式（中断模式（LVIMDS1、LVIMDS0=0、1）和检测电压（ V_{LVD} ）。如果设置中断模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器（LVIM）的bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVS））。
- 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} ）。

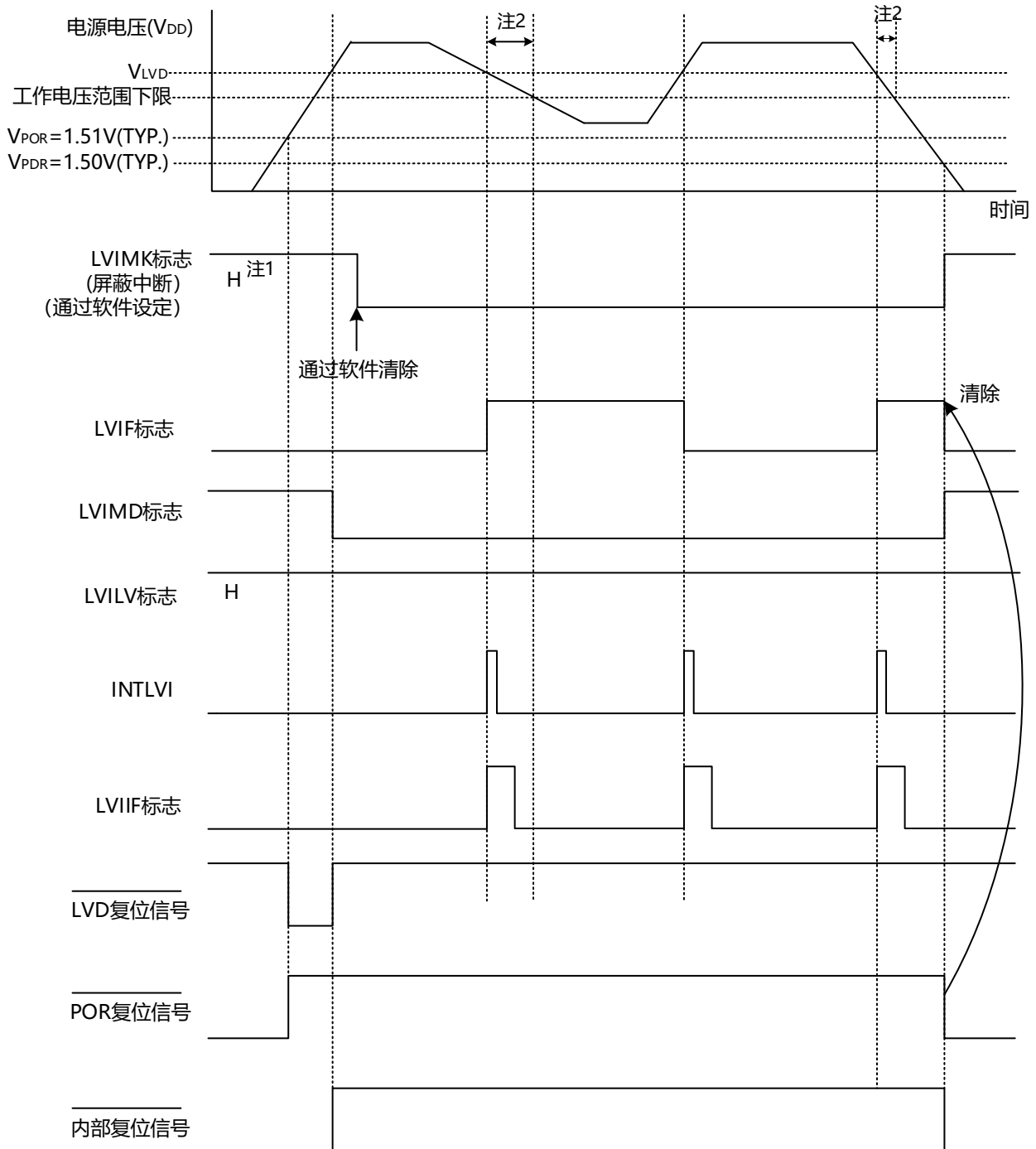
●LVD中断模式的运行

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22-5所示。

图22-5 中断信号的产生时序（选项字节的LVIMDS1、LVIMDS0=0、1）



- 注 1. 在产生复位信号后，LVIMK标志变为“1”。
2. 当工作电压下降时，必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注 V_{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

22.4.3 用作中断&复位模式时的设置

通过选项字节000C1H设置运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0）和检测电压（ V_{LVDH} 、 V_{LVDL} ）

如果设置中断&复位模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器（LVIM）的bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。
bit7（LVIMD）为“0”（中断模式）。
bit0（LVILV）为“0”（高电压检测电平： V_{LVDH} ）。

●LVD中断&复位模式的运行

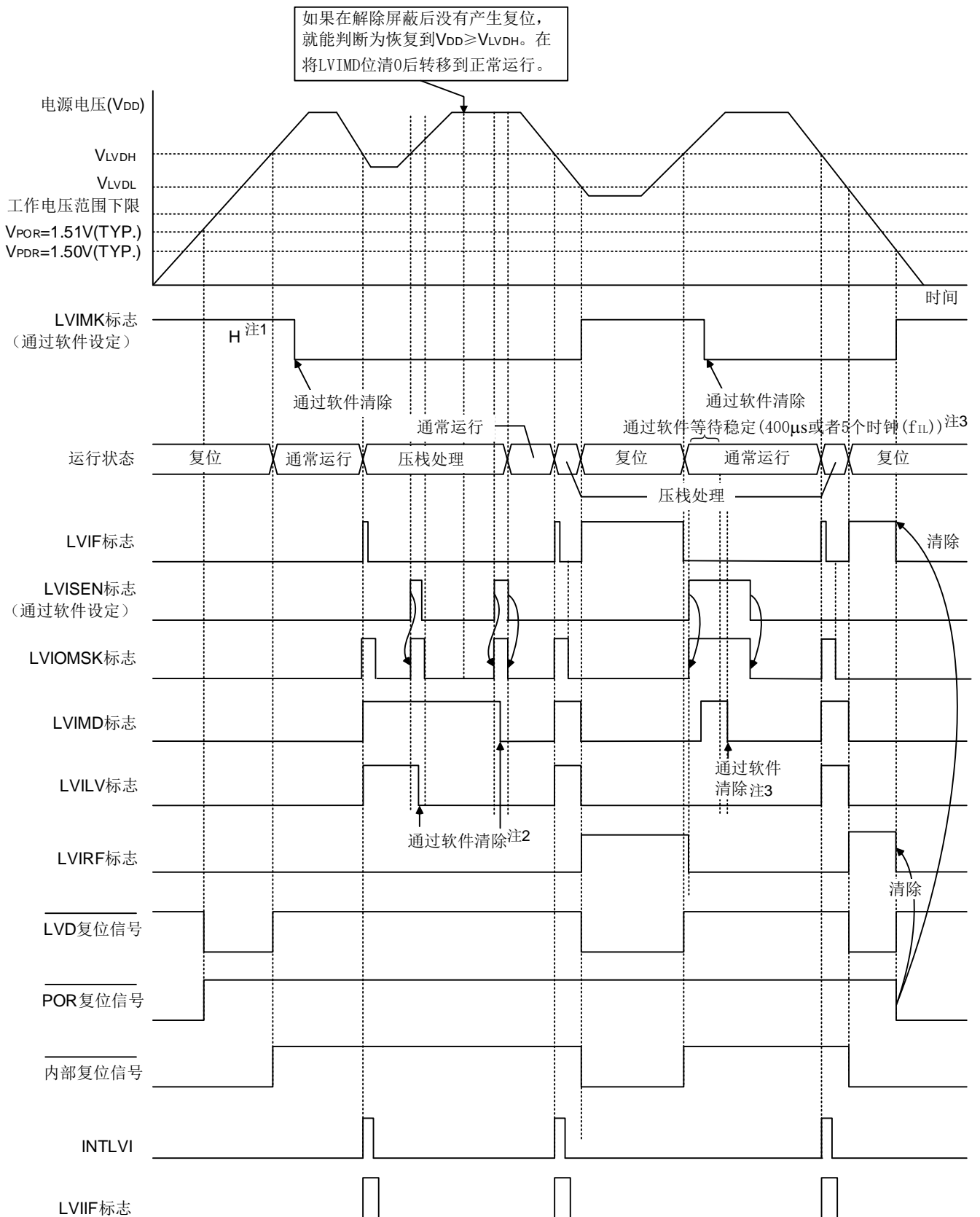
当接通电源时，中断&复位模式（选项字节的LVIMDS1、LVIMDS0=1、0）在电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ）前保持LVD的内部复位状态。如果电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ），就解除内部复位。

当工作电压下降时，如果电源电压（ V_{DD} ）低于高电压检测电平（ V_{LVDH} ），就产生LVD的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（ V_{DD} ）低于低电压检测电平（ V_{LVDL} ），就产生LVD的内部复位。但是，在发生INTLVI后，即使电源电压（ V_{DD} ）在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高，也不产生中断请求信号。

当使用LVD中断&复位模式时，必须按照“图22-7 工作电压的确认/复位的设置步骤”和“图22-8 中断&复位模式的初始设置步骤”所示的流程图的步骤进行设置。

LVD中断&复位模式的内部复位信号和中断信号的产生时序如图22-6示。

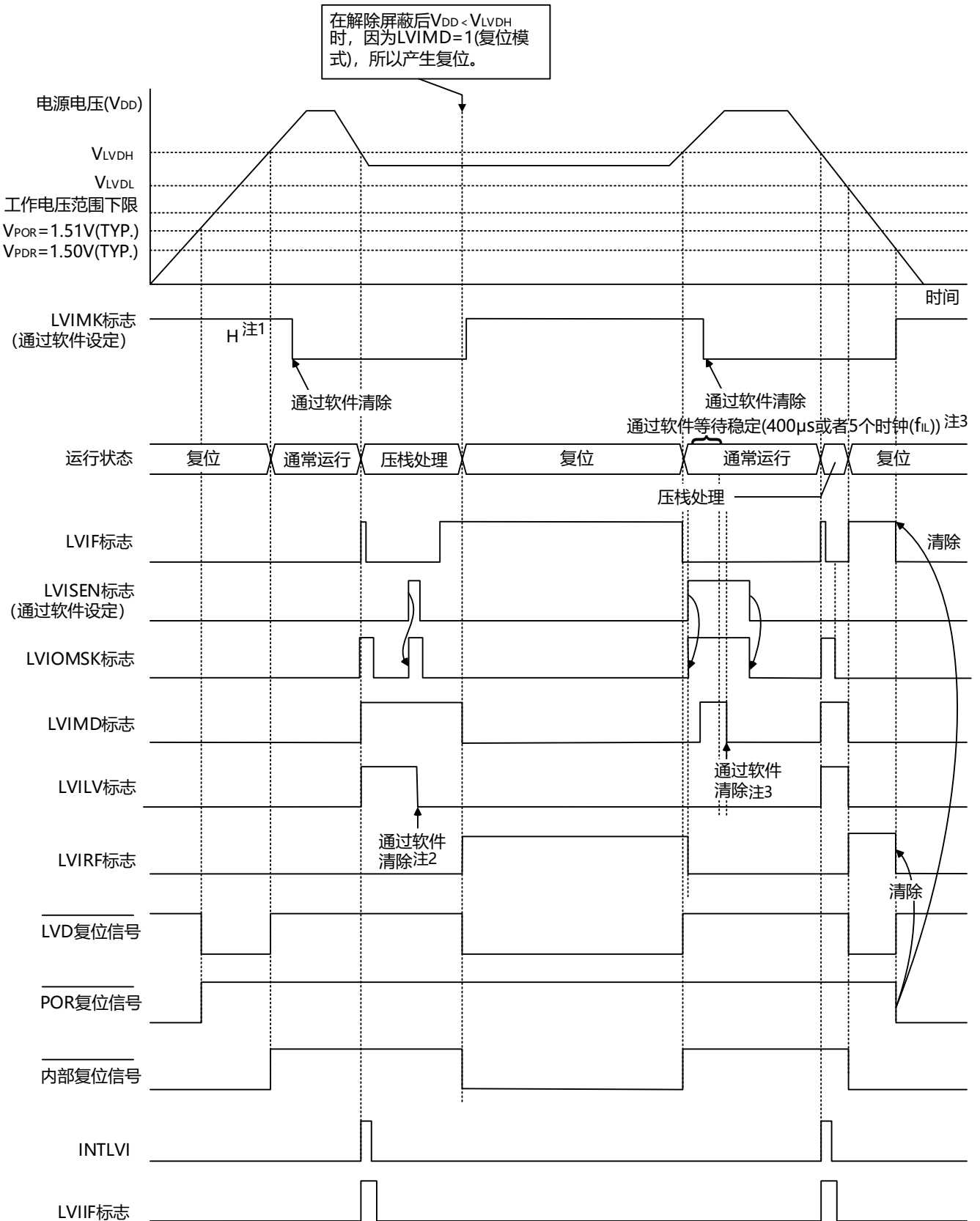
图22-6 复位&中断信号的产生时序 (选项字节的LVIMDS1、LVIMDS0=1、0) (1/2)



- 注
- 1.在产生复位信号后，LVIMK标志变为“1”。
 - 2.当使用中断&复位模式时，必须在发生中断后按照“图22-7 工作电压的确认/复位的设置步骤”进行设置。
 - 3.当使用中断&复位模式时，必须在解除复位后按照“图22-8 中断&复位模式的初始设置步骤”进行设置。

备注 V_{POR} : POR电源电压上升检测电压
 V_{PDR} : POR电源电压下降检测电压

图22-6 中断&复位信号的产生时序 (选项字节的LVIMDS1、LVIMDS0=1、0) (2/2)

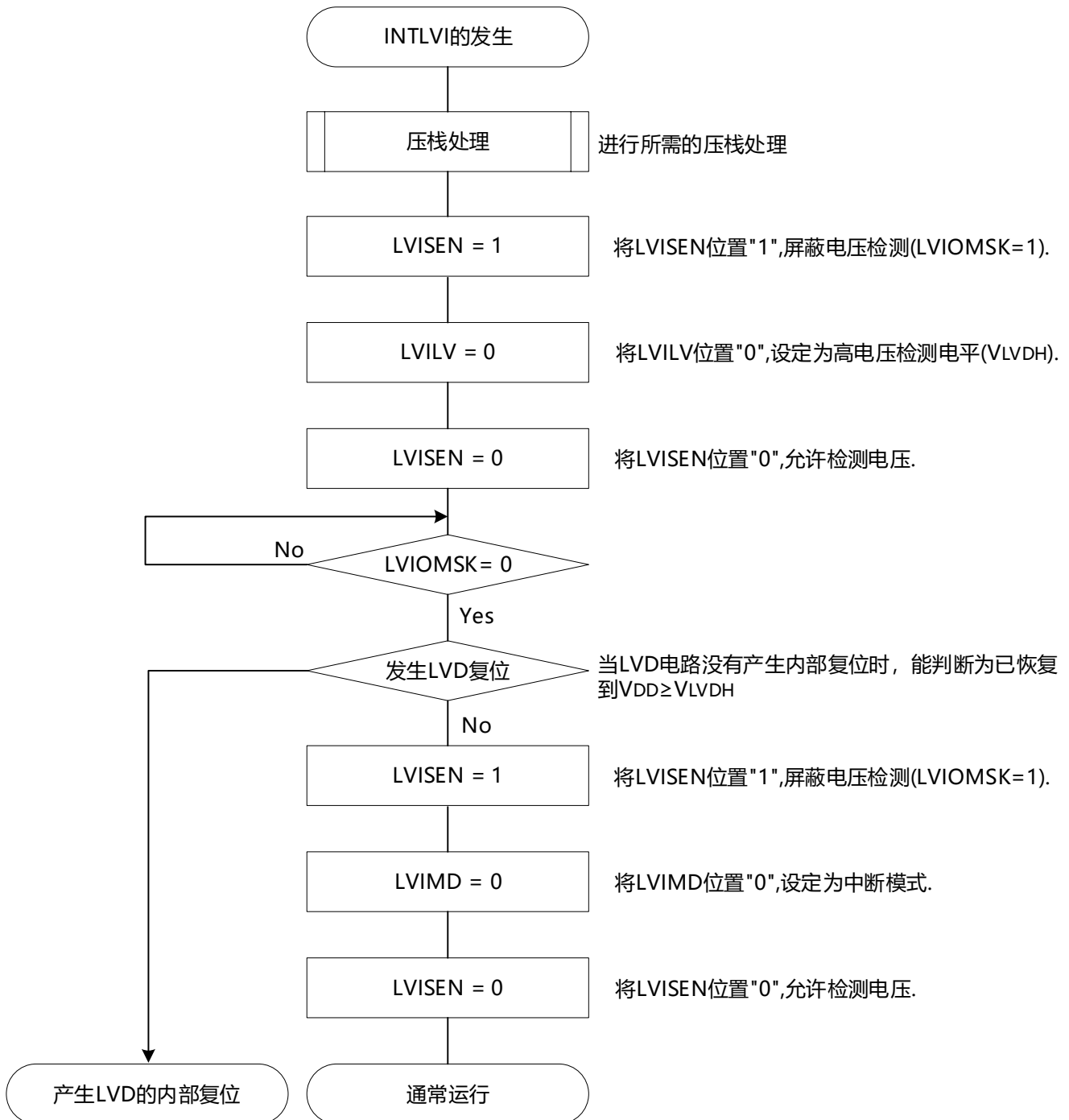


- 注
- 1.在产生复位信号后，LVIMK标志变为“1”。
 - 2.当使用中断&复位模式时，必须在发生中断后按照“图22-7 工作电压的确认/复位的设置步骤”进行设置。
 - 3.当使用中断&复位模式时，必须在解除复位后按照“图22-8 中断&复位模式的初始设置步骤”进行设置。

备注 V_{POR} : POR电源电压上升检测电压

V_{PDR} : POR电源电压下降检测电压

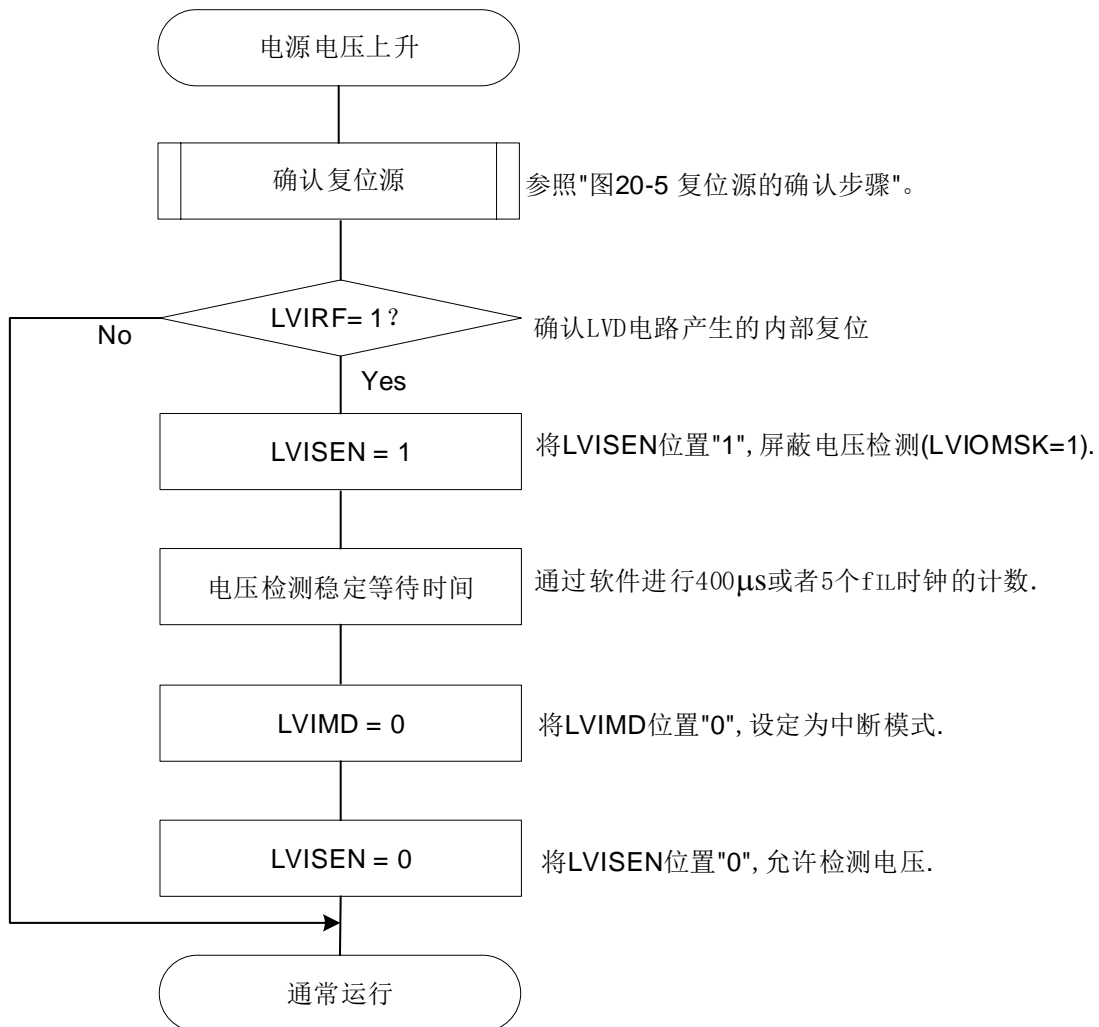
图22-7 工作电压的确认/复位的设置步骤



如果设置中断&复位模式 (LVIMDS1、LVIMDS0=1、0)，就在解除LVD复位 (LVIRF=1) 后需要400us或者5个 f_{IL} 时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将LVIMD位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写LVIMD位时，必须将LVISEN位置“1”，屏蔽LVD产生的复位或者中断的产生。

中断&复位模式的初始设置步骤如图22-8所示。

图22-8 中断&复位模式的初始设置步骤



备注 f_{IL} : 低速内部振荡器时钟频率

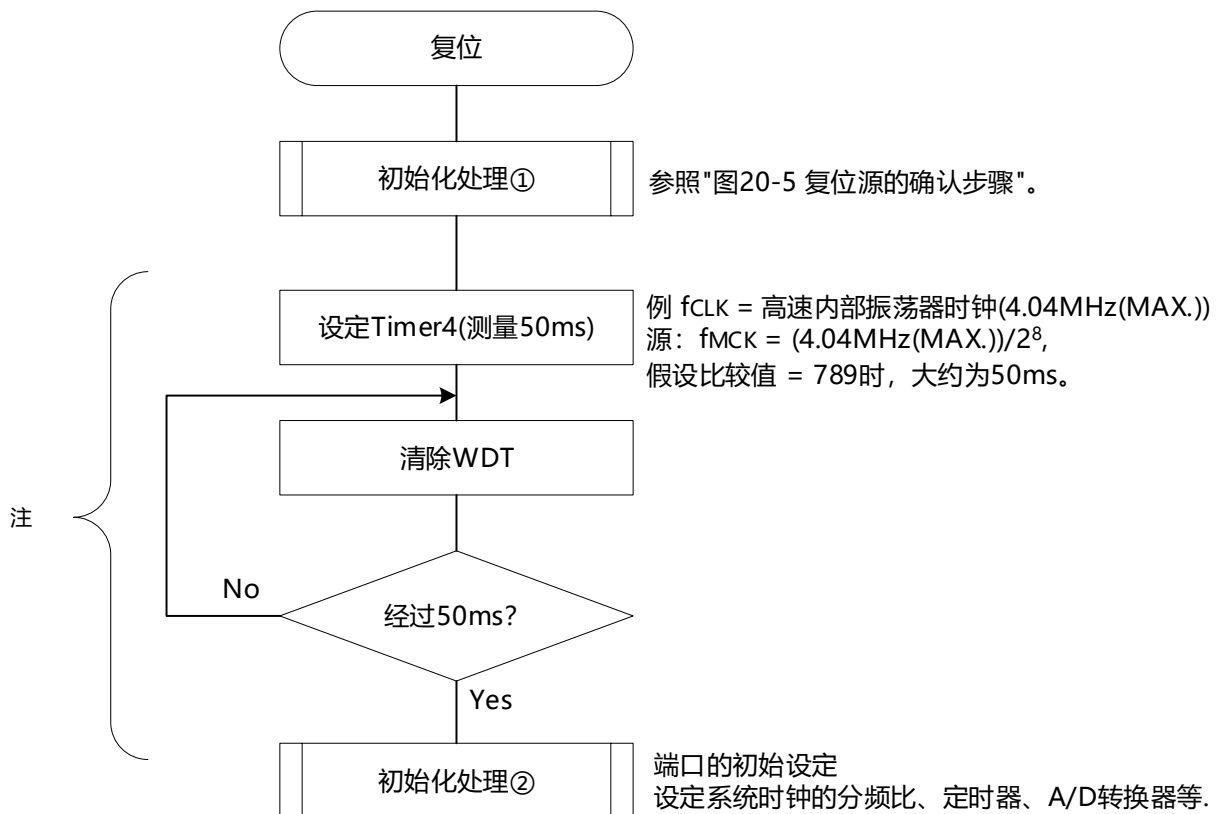
22.5 电压检测电路的注意事项

(1) 有关接通电源时的电压波动

对于电源电压 (V_{DD}) 在LVD检测电压附近发生一定时间波动的系统，有可能重复进入复位状态和复位解除状态。能通过以下的处理，任意设置解除复位到单片机开始运行的时间。

<处理> 在解除复位后，必须通过使用定时器的软件计数器，在等待各系统不同的电源电压波动时间后进行端口等的初始设置。

图22-9 LVD检测电压附近的电源电压波动不超过50ms时的软件处理例子

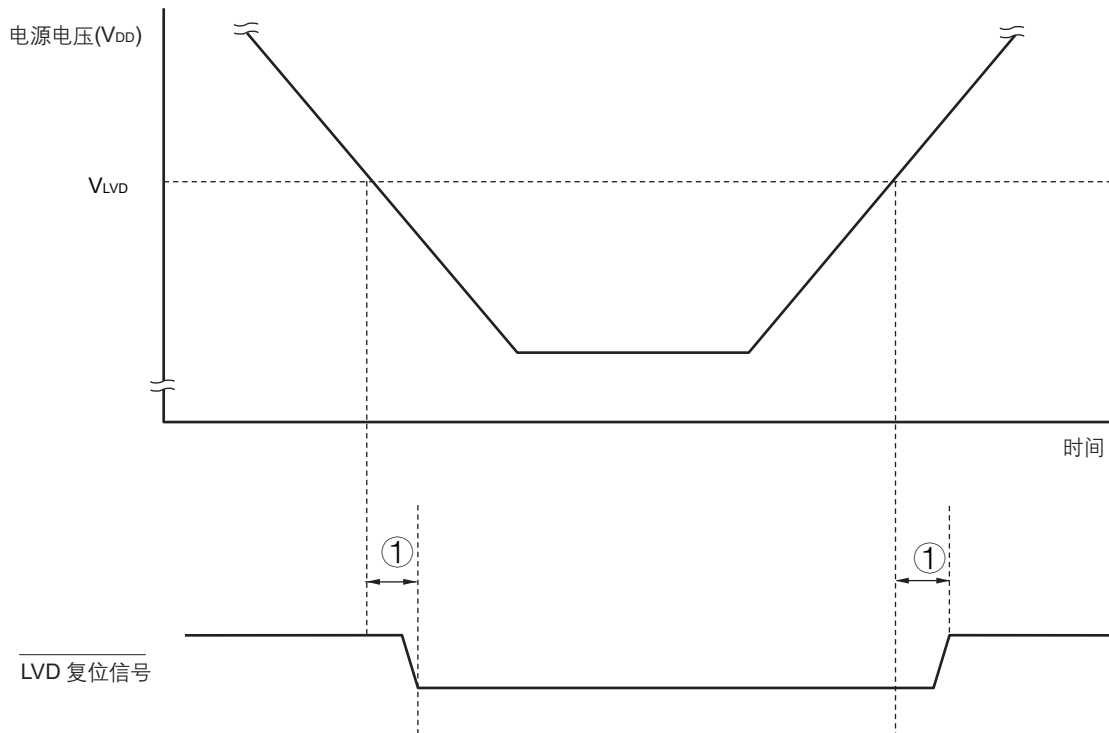


注 如果在此期间再次发生复位，就不转移到初始化处理②。

(2) 从产生LVD复位源到产生或者解除LVD复位的延迟

从满足电源电压 (V_{DD}) < LVD检测电压 (V_{LVD}) 到产生LVD复位为止会发生延迟。同样，从LVD检测电压 (V_{LVD}) ≤ 电源电压 (V_{DD}) 到解除LVD复位为止也会发生延迟 (参照图22-10)。

图22-10 从产生LVD复位源到产生或者解除LVD复位的延迟



①：检测延迟 (300us(MAX.))

(3) 有关将LVD置为OFF时接通电源的情况

当将LVD置为OFF时，必须使用RESETB引脚的外部复位。

在进行外部复位时，必须至少给RESETB引脚输入10us的低电平。如果在电源电压上升时进行外部复位，就必须在给RESETB引脚输入低电平后接通电源，而且在数据手册的AC特性所示的工作电压范围内至少保持10us的低电平，然后输入高电平。

(4) 有关将LVD置为OFF并且设置为LVD中断模式时工作电压下降的情况

在将LVD置为OFF并且设置为LVD中断模式的情况下，如果工作电压下降，就必须在工作电压低于数据手册的AC特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

第23章 安全功能

23.1 安全功能的概要

为了对应IEC60730和EC61508安全标准，BAT32G133内置以下安全功能。

此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

(1) 闪存CRC运算功能（高速CRC、通用CRC）

通过CRC运算检测闪存的数据错误。能根据不同的用途和使用条件，分别使用以下2个CRC。

- “高速CRC”... 在初始化程序中，能停止CPU的运行并且高速检查整个代码闪存区。
- “通用CRC”... 在CPU运行中，不限于代码闪存区而能用于多用途的检查。

(2) RAM 奇偶校验错误检测功能

在读RAM数据时，检测奇偶校验错误。

(3) SFR保护功能

防止因CPU失控而改写SFR。

(4) 频率检测功能

能使用通用定时器单元进行CPU/外围硬件时钟频率的自检。

(5) A/D测试功能

能通过A/D转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、温度传感器输出和内部基准电压输出的A/D转换进行A/D转换器的自检。

(6) 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

(7) 产品唯一身份标识寄存器（128位）

23.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
<ul style="list-style-type: none"> •闪存CRC控制寄存器 (CRC0CTL) •闪存CRC运算结果寄存器 (PGCRCL) 	闪存CRC运算功能 (高速CRC)
<ul style="list-style-type: none"> •CRC输入寄存器 (CRCIN) •CRC数据寄存器 (CRCD) 	CRC运算功能 (通用CRC)
•RAM奇偶校验错误控制寄存器 (RPECTL)	RAM奇偶校验错误检测功能
•特殊SFR保护控制寄存器 (SFRGD)	SFR保护功能
•定时器输入选择寄存器0 (TIS0)	频率检测功能
•A/D测试寄存器 (ADTES)	A/D测试功能
•端口模式选择寄存器 (PMS)	输入/输出引脚的数字输出信号电平检测功能

有关各寄存器的内容，在“23.3 安全功能的运行”中进行说明。

23.3 安全功能的运行

23.3.1 闪存CRC运算功能（高速CRC）

IEC60730标准要求确认闪存中的数据，并且建议CRC为确认手段。此高速CRC能在初始设置（初始化）程序中检查整个代码闪存区。

高速CRC停止CPU的运行并且通过1个时钟从闪存读32位数据进行运算。因此，其特点是完成检查的时间较短（例如，64KB闪存：512us@32MHz）。

CRC生成多项式对应CRC-16-CCITT的“ $X^{16}+X^{12}+X^5+1$ ”。

以bit31->bit0的MSB优先进行运算。

备注 因为通用CRC为LSB优先，所以运算结果不同。

23.3.1.1 闪存CRC控制寄存器（CRC0CTL）

这是设置高速CRC运算器的运行控制和运算范围的寄存器。通过8位存储器操作指令设置CRC0CTL寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 23-1 闪存CRC控制寄存器（CRC0CTL）的格式

地址：40021810H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	0	0	0	0	FEA1	FEA0

CRC0EN	高速CRC运算器的运行控制
0	停止运行。
1	通过执行WFE指令开始运算。

FEA1	FEA0	高速 CRC 的演算范围
0	0	00000H ~ 1FFBH(8K-4byte)
0	1	00000H ~ 3FFBH(16K-4byte)
1	0	00000H ~ 5FFBH(24K-4byte)
1	1	00000H ~ 7FFBH(32K-4byte)

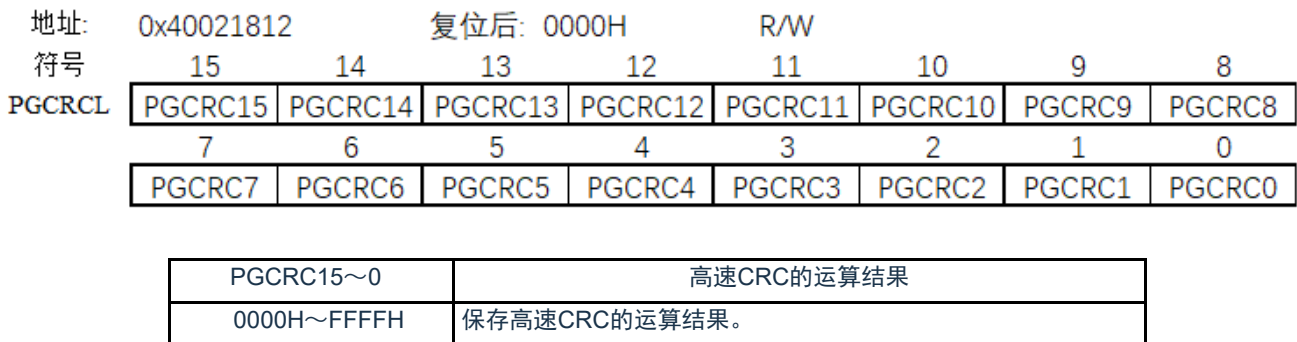
注：bit2-6必须设置为0。

备注 必须事先将用于比较的CRC运算结果期待值存入闪存的最后4字节，因此运算范围为减去4字节的范围。

23.3.1.2 闪存CRC运算结果寄存器 (PGCRCL)

这是保存高速CRC运算结果的寄存器。
 通过16位存储器操作指令设置PGCRCL寄存器。
 在产生复位信号后，此寄存器的值变为“0000H”。

图 23-2 闪存CRC运算结果寄存器 (PGCRCL) 的格式

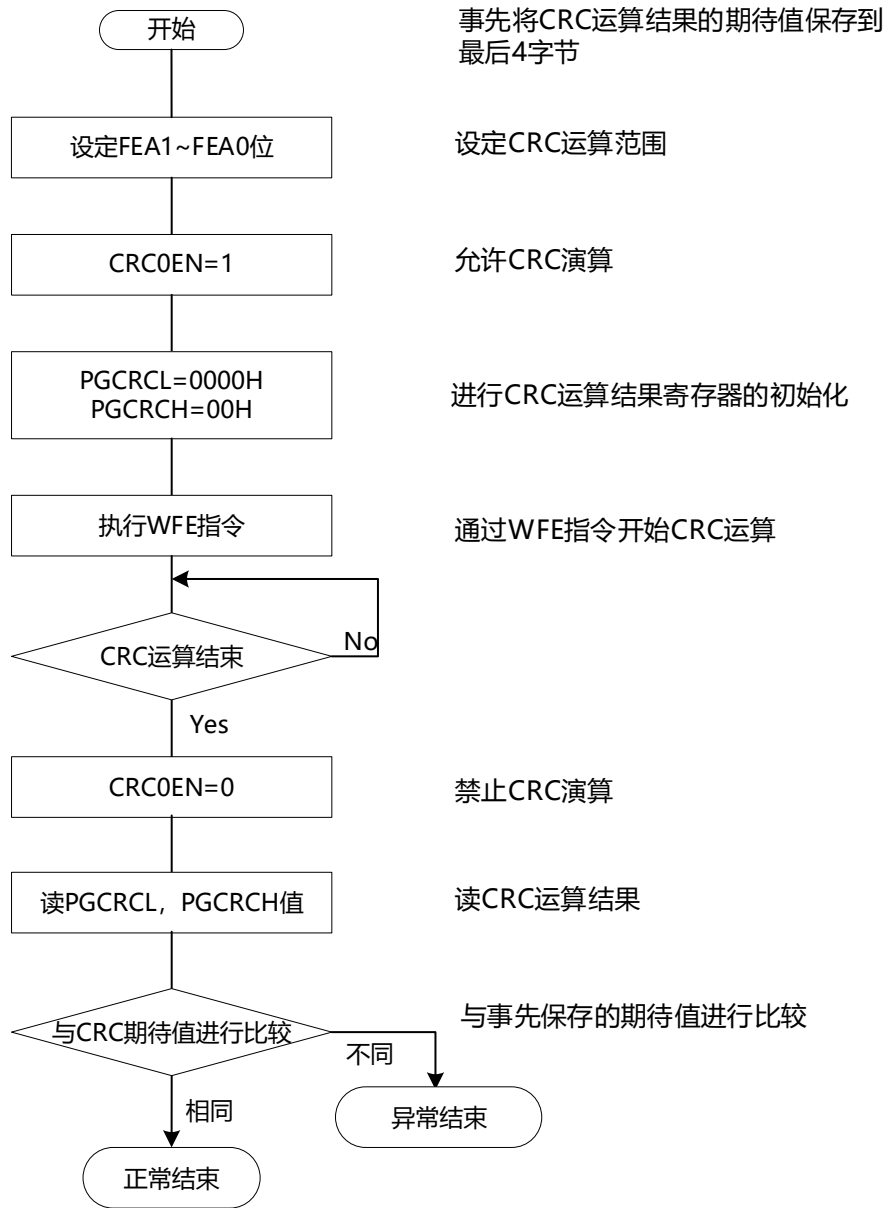


注意 只有在CRC0EN (CRC0CTL寄存器的bit7) 位为“1”时才能写PGCRCL寄存器。

闪存CRC运算功能 (高速CRC) 的流程图如图23-3所示。

< 操作流程 >

图23-3 闪存CRC运算功能（高速CRC）的流程图



注意1.只以代码闪存为CRC运算的对象。

2.必须将CRC运算的期待值保存在代码闪存中的运算范围后的区域。

3.连接调试器时CRC不能进行运算。

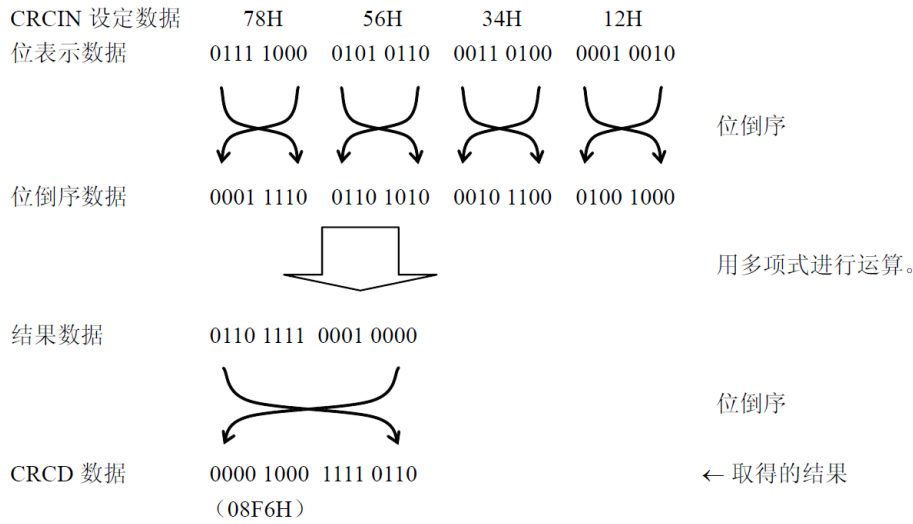
23.3.2 CRC运算功能（通用CRC）

为了必须保证运行过程中的安全，IEC61508标准要求即使在CPU运行中也需要确认数据。

此通用CRC能在CPU运行中作为外围功能进行CRC运算。通用CRC不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定要确认的数据。睡眠模式中的CRC运算功能只能在DMA传送过程中使用。

在主系统时钟运行模式或者副系统时钟运行模式中，都能使用CRC运算功能。

CRC生成多项式使用CRC-16-CCITT的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以LSB优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从LSB发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给CRCIN寄存器写值，从CRCD寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行CRC运算的结果。



注意 在执行程序的过程中，因为调式程序将软件断点的设置行改写为断点指令，所以如果在CRC运算的对象区设置软件断点，CRC的运算结果就不同。

23.3.2.1 CRC输入寄存器 (CRCIN)

这是设置通用CRC的CRC计算数据的8位寄存器。能设置的范围为“00H~FFH”。
通过8位存储器操作指令设置CRCIN寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图23-4 CRC输入寄存器 (CRCIN) 的格式

地址: 400433ACH 复位后: 00H R/W



bit7~0	功能
00H~FFH	数据输入

23.3.2.2 CRC数据寄存器 (CRCD)

这是保存通用CRC运算结果的寄存器。能设置的范围为“0000H~FFFFH”。

在写CRCIN寄存器后经过1个CPU/外围硬件时钟 (f_{CLK}), 将CRC运算结果保存到CRCD寄存器。

通过16位存储器操作指令设置CRCD寄存器。

在产生复位信号后, 此寄存器的值变为“0000H”。

图23-5 CRC数据寄存器 (CRCD) 的格式

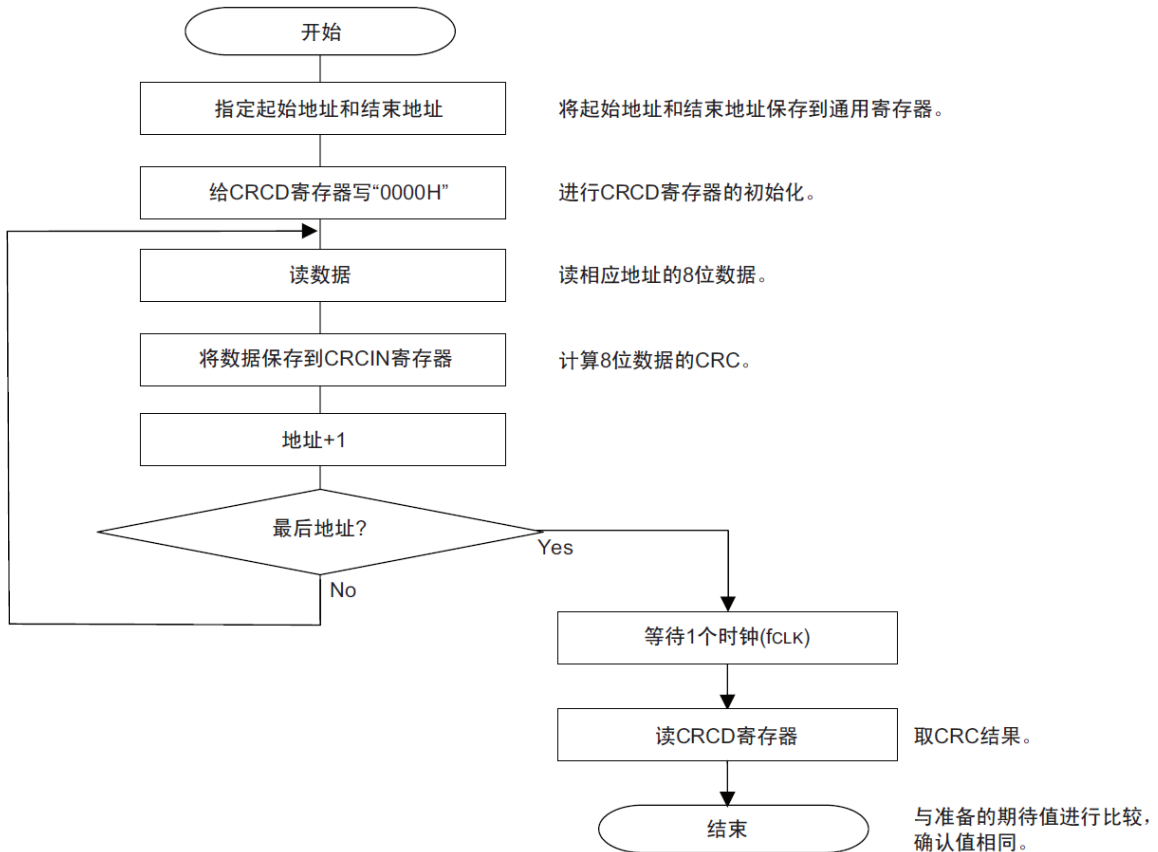


注意1.要读CRCD寄存器的写入值时, 必须在写CRCIN寄存器前读CRCD寄存器。

2.如果CRCD寄存器的写操作与运算结果的保存发生竞争, 就忽视写操作。

<操作流程>

图23-6 CRC运算功能 (通用CRC) 的流程图



23.3.3 RAM奇偶校验错误检测功能

IEC60730标准要求确认RAM数据。因此，BAT32G133的RAM每8位附加1位奇偶校验位。RAM奇偶校验错误检测功能在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位，并且能在发生奇偶校验错误时产生复位。

31.3.3.1 RAM奇偶校验错误控制寄存器（RPECTL）

此寄存器控制奇偶校验的错误确认位和因奇偶校验错误而产生复位。
通过8位存储器操作指令设置RPECTL寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图23-7 RAM奇偶校验错误控制寄存器（RPECTL）的格式

地址：40020425H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

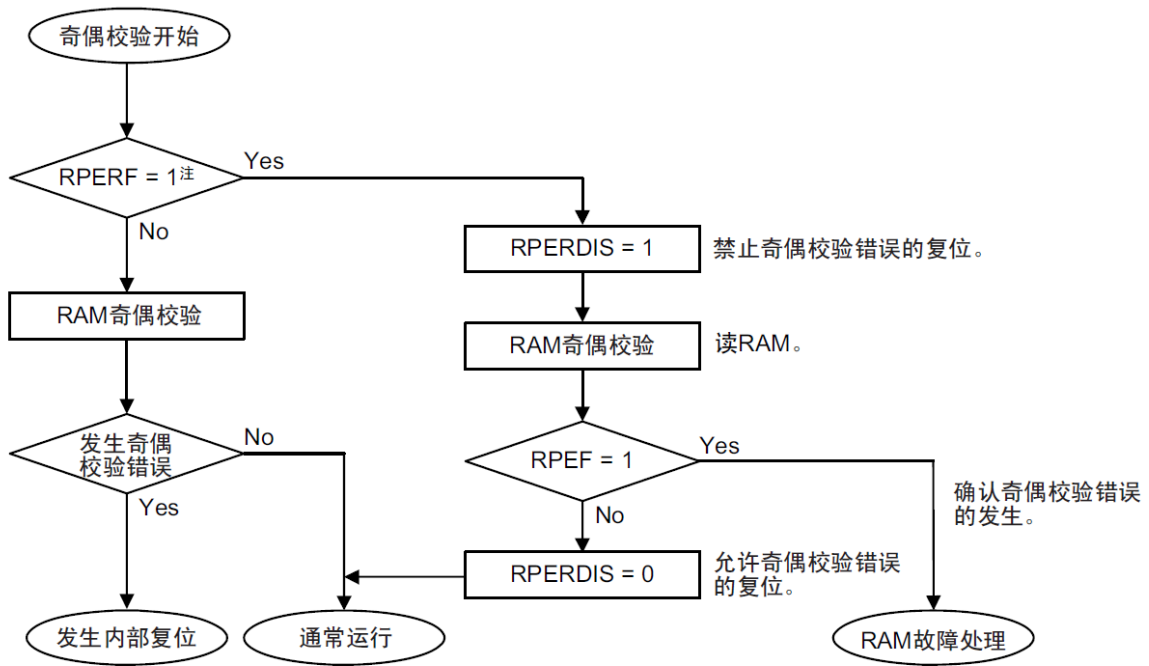
RPERDIS	奇偶校验错误复位的屏蔽标志
0	允许产生奇偶校验错误复位。
1	禁止产生奇偶校验错误复位。

RPEF	奇偶校验错误状态标志
0	没有发生奇偶校验错误。
1	发生奇偶校验错误。

注意 在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位。
因此，要允许产生RAM奇偶校验错误复位（RPERDIS=0）时，必须在存取数据时并且在读数据前对“所用RAM区”进行初始化。
因为是流水线运行，所以CPU进行预读，有可能因读所用RAM区前的未初始化RAM区而发生RAM奇偶校验错误。因此，要允许产生RAM奇偶校验错误复位（RPERDIS=0）时，必须在从RAM区执行指令时对“所用RAM区+10字节”的区域进行初始化。

- 备注1.初始状态为允许产生奇偶校验错误复位（RPERDIS=0）。
- 即使设置为禁止产生奇偶校验错误复位（RPERDIS=1），也在发生奇偶校验错误时将RPEF标志置“1”。如果在RPEF位为“1”的状态下设置为允许产生奇偶校验错误复位（RPERDIS=0），就在将RPERDIS清“0”时产生奇偶校验错误复位。
 - 因RAM奇偶校验错误而将RPECTL寄存器的RPEF标志置“1”，通过写“0”或者全部复位源将RPEF标志清“0”。当RPEF标志为“1”时，即使读未发生奇偶校验错误的RAM，RPEF标志也保持“1”的状态。
 - RAM奇偶校验检测的范围不包括通用寄存器。

图23-8 RAM奇偶校验的流程



注 有关RAM奇偶错误的内部复位的确认，请参照“第20章 复位功能”。

23.3.4 SFR保护功能

为了必须保证运行过程中的安全，IEC61508标准要求即使CPU失控也需要保护重要的SFR，使其免遭改写。SFR保护功能用于保护比较器功能、端口功能、中断功能、时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的数据。

如果设置为SFR保护功能，被保护的SFR的写操作就无效，但是能正常读取。

23.3.4.1 SFR保护控制寄存器（SFRGD）

此寄存器控制SFR保护功能是否有效。

SFR保护功能使用GCOMP位、GPORT位、GINT位和GCSC位。

通过8位存储器操作指令设置SFRGD寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图23-9 SFR保护控制寄存器（SFRGD）的格式

地址：40040478H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
SFRGD	0	0	0	0	GCOMP	GPORT	GINT	GCSC
GCOMP		比较器功能的控制寄存器的保护						
	0	无效。能读写比较器功能的控制寄存器。						
	1	有效。比较器功能的控制寄存器的写操作无效，能读。 [被保护的SFR]COMPMDR、COMPFIR、COMPOCR、CVRCTL、CxRVM、PGAxCTL、PGASHMD、CMPSELx						
GPORT		端口功能的控制寄存器的保护						
	0	无效。能读写端口功能的控制寄存器。						
	1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的SFR]PMxx、PUxx、PDxx、POMxx、PMCxx、PxxCFG、PIORx ^注						
GINT		中断功能的寄存器的保护						
	0	无效。能读写中断功能的控制寄存器。						
	1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的SFR]IFxx、MKxx、PRxx、EGPx、EGNx						
GCSC		时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的保护						
	0	无效。能读写时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器。						
	1	有效。时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的写操作无效，能读。 [被保护的SFR]CMC、CSC、OSTS、CKC、PERx、OSMC、LVIM、LVIS、RPECTL						

注 不保护Pxx（端口寄存器）。

23.3.5 频率检测功能

IEC60730标准要求确认振荡频率是否正常。

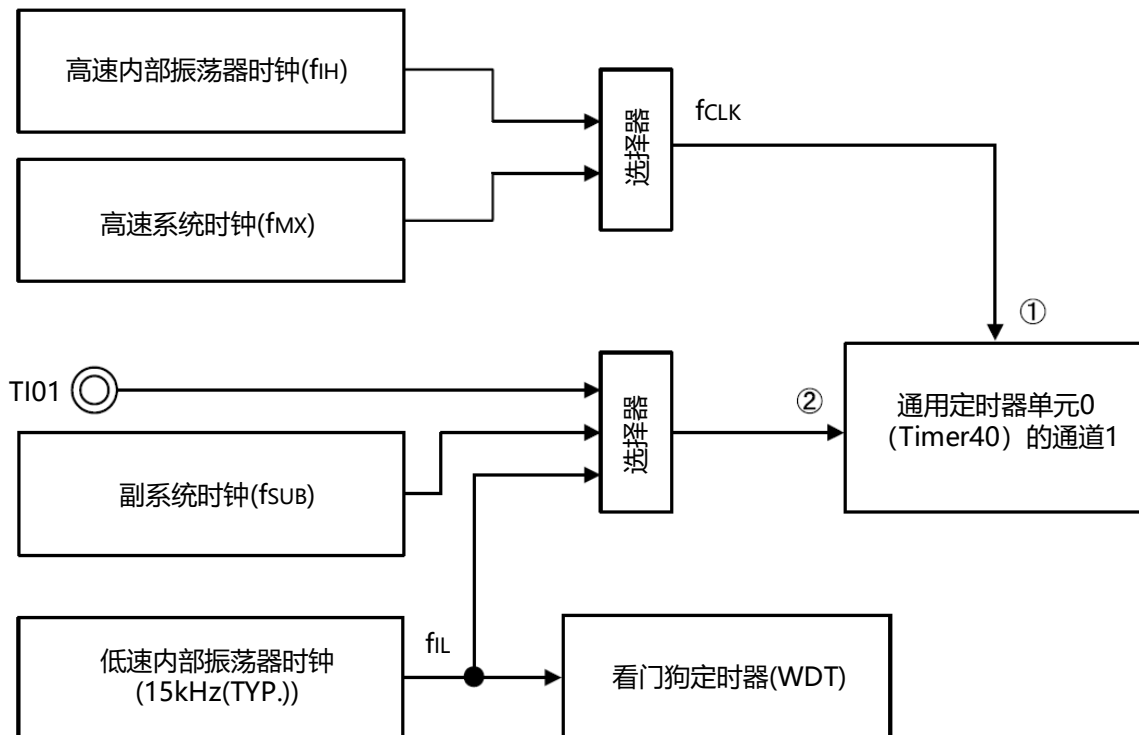
频率检测功能可使用CPU/外围硬件的时钟频率 (f_{CLK})，并且能通过测量Timer40通道1输入脉冲，判断2个时钟的比率关系是否正确。

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>

- ①CPU/外围硬件的时钟频率 (f_{CLK}):
 - 高速内部振荡器时钟 (f_{IH})
 - 高速系统时钟 (f_{MX})
- ②Timer40通道1输入:
 - 通道1的定时器输入 (TI01)
 - 低速内部振荡器时钟 (f_{IL} : 15kHz(TYP.))
 - 副系统时钟 (f_{SUB}) 注

图23-10 频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。有关输入脉冲间隔的测量方法，请参照“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。

23.3.5.1 定时器输入选择寄存器0 (TIS0)

寄存器说明请参考5.3.8节。

23.3.6 A/D测试功能

IEC60730标准要求进行A/D转换器的测试。此A/D测试功能通过对A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(ANI)、温度传感器的输出电压和内部基准电压进行A/D转换,确认A/D转换器是否正常运行。

能通过以下步骤确认模拟多路转换器:

- ① 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ② 对ANix引脚进行A/D转换(转换结果1-1)。
- ③ 通过ADTES寄存器选择A/D转换器的负(-)基准电压作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、1)。
- ④ 对A/D转换器的负(-)基准电压进行A/D转换(转换结果2-1)。
- ⑤ 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ⑥ 对ANix引脚进行A/D转换(转换结果1-2)。
- ⑦ 通过ADTES寄存器选择A/D转换器的正(+)基准电压作为A/D转换对象(ADTES2、ADTES1、ADTES0=1、0、1)。
- ⑧ 对A/D转换器的正(+)基准电压进行A/D转换(转换结果2-2)。
- ⑨ 通过ADTES寄存器选择ANix引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ⑩ 对ANix引脚进行A/D转换(转换结果1-3)。
- ⑪ 确认“转换结果1-1”、“转换结果1-2”和“转换结果1-3”相同。
- ⑫ 确认“转换结果2-1”的A/D转换结果全部为“0”并且“转换结果2-2”的A/D转换结果全部为“1”。通过以上步骤,能选择模拟多路转换器以及确认布线没有断线。

备注1.在①~⑩的转换过程中,如果模拟输入电压可变,就必须采用其他方法来确认模拟多路转换器。

2.转换结果含有误差,因此必须在比较转换结果时要适当考虑误差。

23.3.6.1 A/D测试寄存器 (ADTES)

此寄存器选择A/D转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANlxx)、温度传感器的输出电压和内部基准电压 (1.45V) 作为A/D转换对象。

当用作A/D测试功能时, 进行以下的设置:

- 在测量零刻度时, 选择负 (-) 基准电压作为A/D转换对象。
- 在测量满刻度时, 选择正 (+) 基准电压作为A/D转换对象。

A/D寄存器说明请参考10.2.10。

23.3.6.2 模拟输入通道指定寄存器 (ADS)

此寄存器指定A/D转换的模拟电压的输入通道。

要通过A/D测试功能测量ANlxx、温度传感器输出或者内部基准电压 (1.45V) 时, 必须将A/D测试寄存器 (ADTES) 置“00H”。

寄存器说明请参考10.2.13A/D测试寄存器 (ADTES)。

23.3.7 输入/输出引脚的数字输出信号电平检测功能

IEC60730标准要求确认I/O功能是否正常。

输入/输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

23.3.7.1 端口模式选择寄存器（PMS）

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的PMmn位为“0”）时是读端口的输出锁存器的值还是读引脚的输出电平。

通过8位存储器操作指令设置PMS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图23-11 端口模式选择寄存器（PMS）的格式

地址: 4004087BH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	在引脚为输出模式时读数据的选择
0	读Pmn寄存器的值。
1	读引脚的数字输出电平。

注意1.对于使用定时器M的脉冲输出强制截止功能而使引脚变为高阻抗状态的引脚，如果读引脚的数字输出电平，读取值就为“0”。

备注 m=0~3
n=0~7

23.3.8 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

- 用来作为序列号(例如USB字符序列号或者其他的终端应用)
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器内的安全性。
- 用来激活带安全机制的自举过程

128位的产品唯一身份标识所提供的参考号码对任意一个BAT32微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

基地址：0x0050_084C

地址偏移：0x00

只读，其值在出厂时编写

U_ID[31:0]

地址偏移：0x04

只读，其值在出厂时编写

U_ID[63:32]

地址偏移：0x08

只读，其值在出厂时编写

U_ID[95:64]

地址偏移：0x0C

只读，其值在出厂时编写

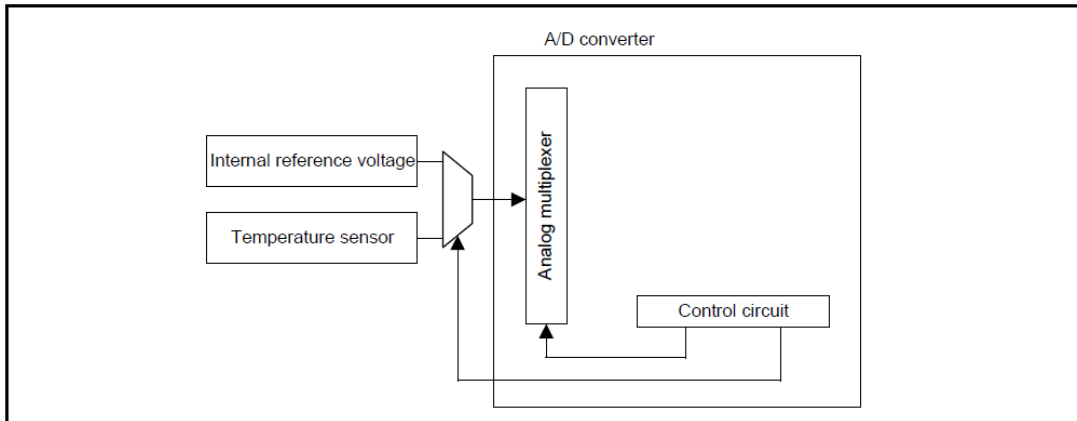
--

第24章 温度传感器

24.1 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 24-1显示了温度传感器框图。

图 24-1温度传感器框图



24.2 温度传感器的寄存器

24.2.1 温度传感器校准数据寄存器 TSCDR1

地址：0x4002_1C38

符号	7	6	5	4	3	2	1	0	复位后	R/W
TSCDR1	TSCDR1[7:0]								-	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。

24.2.2 温度传感器校准数据寄存器 TSCDR2

地址：0x4002_1C3C

符号	7	6	5	4	3	2	1	0	复位后	R/W
TSCDR2	TSCDR2[7:0]								-	R

只读寄存器，用于记录温度传感器的校准数据2，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

24.3 温度传感器的使用说明

24.3.1 温度传感器的使用原理

温度 (T) 与传感器电压输出 (Vs) 成正比, 因此温度的计算公式如下:

$$T = (V_s - V_1) / \text{slope} + T_1$$

T: 测量的温度 (°C)

Vs: 温度传感器在温度测量时的输出电压 (V)

T1: 在第一个点进行实验测量的温度 (°C)

V1: 温度传感器测量T1时的电压输出 (V)

T2: 在第二个点进行实验测量的温度 (°C)

V2: 温度传感器测量T2时的电压输出 (V)

Slope: 温度传感器的温度斜率(V/°C), $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$.

不同传感器的特性不同, 因此我们建议测量以下两种不同的样品温度:

- 1、使用A/D转换器测量温度传感器在温度T1时输出的电压V1。
- 2、使用A/D转换器测量温度传感器在第二个温度T2时输出的电压V2。
- 3、从两次结果中计算获得温度斜率($\text{slope} = (V_2 - V_1) / (T_2 - T_1)$)
- 4、随后, 通过将斜率代入温度特性的公式来获得温度($T = (V_s - V_1) / \text{slope} + T_1$).

24.3.2 温度传感器的使用方法

方法一：在本产品中，TSCDR1寄存器存储了在 $T_a=T_j=25^{\circ}\text{C}$ 和 $V_{DD}=3.0\text{V}$ 条件下测量的温度传感器的电压转换值（CAL25）。TSCDR2寄存器存储了在 $T_a=T_j=85^{\circ}\text{C}$ 和 $V_{DD}=3.0\text{V}$ 条件下测量的温度传感器的电压转换值（CAL85）。利用这两组值可以计算出温度斜率：

$$\text{slope} = (V_2 - V_1) / (85 - 25).$$

$$V_1 = 3.0 \times \text{CAL25} / 256 [\text{V}]$$

$$V_2 = 3.0 \times \text{CAL85} / 256 [\text{V}]$$

利用以上结果，可根据以下公式计算温度：

$$T = (V_s - V_1) / \text{slope} + 25 [^{\circ}\text{C}]$$

T： 测量的温度（ $^{\circ}\text{C}$ ）

V_s ：使用A/D转换器得到的温度传感器在T温度时的输出电压（V）

方法二：如果使用“电气特性”中给出的温度斜率，可直接使用以下公式计算测量温度：

$$T = (V_s - V_1) / \text{slope} + 25 [^{\circ}\text{C}]$$

注：这种方法产生的温度比方法一测量的精度要低。

第25章 选项字节

25.1 选项字节的功能

BAT32G133的闪存000C0H~000C3H, 500004H~500005H为选项字节区。

选项字节由用户选项字节(000C0H~000C2H)和闪存数据保护选项字节(000C3H, 500004H~500005H)构成。在接通电源或者复位启动时,自动参照选项字节进行指定功能的设置。在使用本产品时,必须通过选项字节进行以下功能的设置。对于没有配置功能的位,不能更改初始值。要在自编程过程中使用引导交换功能时,因为000C0H~000C3H被010C0H~010C3H替换,所以必须给010C0H~010C3H设置和000C0H~000C3H相同的值。

注意 与是否使用各功能无关,必须设置选项字节。

25.1.1 用户选项字节(000C0H~000C2H/010C0H~010C2H)

(1) 000C0H/010C0H

- 看门狗定时器的运行
 - 允许或者禁止计数器的运行。
 - 在睡眠/深度睡眠模式中允许或者停止计数器的运行。
- 看门狗定时器的上溢时间的设置
- 看门狗定时器的窗口打开期间的设置
- 看门狗定时器的间隔中断的设置
 - 使用或者不使用间隔中断。

注意 在引导交换时,000C0H被010C0H替换,因此必须给010C0H设置和000C0H相同的值。

(2) 000C1H/010C1H

- LVD运行模式的设置
 - 中断&复位模式
 - 复位模式
 - 中断模式
 - LVD为OFF(使用RESETB引脚的外部复位输入)。
- LVD检测电平(V_{LVDH} 、 V_{LVDL} 、 V_{LVD})的设置

注意 1.当电源电压上升时,必须在电源电压达到数据手册的AC特性所示的工作电压范围前,通过电压检测电路或者外部复位保持复位状态;当电源电压下降时,必须在电源电压低于工作电压范围前,通过深度睡眠模式的转移、电压检测电路或者外部复位,置为复位状态。

工作电压范围取决于用户选项字节(000C2H/010C2H)的设置。

2.在引导交换时,000C1H被010C1H替换,因此必须给010C1H设置和000C1H相同的值。

(3) 000C2H/010C2H

- 高速内部振荡器的频率设置
 - 从1MHz~32MHz、48MHz、64MHz中选择。

注意 在引导交换时,000C2H被010C2H替换,因此必须给010C2H设置和000C2H相同的值。

25.1.2 闪存数据保护选项字节（000C3H/010C3H，500004H~500005H）

○片上调试时闪存数据保护的 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。

○引导交换功能的 control

- 禁止或者允许使用引导交换功能。

注意 在引导交换时，000C3H被010C3H替换，因此必须给010C3H设置和000C3H相同的值。

25.2 用户选项字节的格式

图 25-1 用户选项字节（000C0H/010C0H）的格式

地址：000C0H/010C0H^{注1}

符号	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	看门狗定时器的间隔中断的使用/不使用
0	不使用间隔中断。
1	当达到上溢时间的75%+1/2 f_{IL} 时，产生间隔中断。

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间 ^{注2}
0	-	禁止设置。
1	0	75%
1	1	100%

WDTON	看门狗定时器的计数器运行控制
0	禁止计数器的运行（解除复位后停止计数）。
1	允许计数器的运行（解除复位后开始计数）。

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=40\text{kHz}(\text{MAX.})$)
0	0	0	$2^6/f_{IL}$ (1.6ms)
0	0	1	$2^7/f_{IL}$ (3.2ms)
0	1	0	$2^8/f_{IL}$ (6.4ms)
0	1	1	$2^9/f_{IL}$ (12.8ms)
1	0	0	$2^{11}/f_{IL}$ (51.2ms)
1	0	1	$2^{13}/f_{IL}$ (204.8ms)
1	1	0	$2^{14}/f_{IL}$ (409.6ms)
1	1	1	$2^{16}/f_{IL}$ (1638.4ms)

WDSTBYON	看门狗定时器的计数器运行控制（睡眠模式）
0	在睡眠模式中，停止计数器的运行 ^{注2} 。
1	在睡眠模式中，允许计数器的运行。

- 注 1.在引导交换时，000C0H被010C0H替换，因此必须给010C0H设置和000C0H相同的值。
 2.当WDSTBYON位为“0”时，与WINDOW1位和WINDOW0位的值无关，窗口打开期间为100%。

备注 f_{IL} ：低速内部振荡器的时钟频率

图 25-2 用户选项字节（000C1H/010C1H）的格式(1/4)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD的设置（中断&复位模式）

检测电压			选项字节的设置值						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降	下降						LVIMDS1	LVIMDS0
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		
—			禁止设置上述以外的值。						

注 在引导交换时，000C1H被010C1H替换，因此必须给010C1H设置和000C1H相同的值。

注意 必须给bit4写“1”。

备注1.有关LVD电路的详细内容，请参照“第22章 电压检测电路”。

2.检测电压是TYP值。详细内容请参照数据手册的LVD电路特性。

图 25-3 用户选项字节（000C1H/010C1H）的格式(2/4)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置（复位模式）

检测电压		选项字节的设置值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
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		
—			禁止设置上述以外的值。					

注：在引导交换时，000C1H被010C1H替换，因此必须给010C1H设置和000C1H相同的值。

注意：必须给bit4写“1”。

备注1.有关LVD电路的详细内容，请参照“第22章 电压检测电路”。

2.检测电压是TYP值。详细内容请参照数据手册的LVD电路特性。

图 25-4 用户选项字节（000C1H/010C1H）的格式(3/4)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD的设置（中断模式）

检测电压		选项字节的设置值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
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		
—			禁止设置上述以外的值。					

注：在引导交换时，000C1H被010C1H替换，因此必须给010C1H设置和000C1H相同的值。

注意：必须给bit4写“1”。

备注1.有关LVD电路的详细内容，请参照“第22章 电压检测电路”。

2.检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

图 25-5 用户选项字节（000C1H/010C1H）的格式(4/4)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD为OFF时的设置（使用RESETB引脚的外部复位输入）

检测电压		选项字节的设定值						
V_{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMD	LVIMD
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

注：在引导交换时，000C1H被010C1H替换，因此必须给010C1H设置和000C1H相同的值。

注意1.必须给bit4写“1”。

- 2.当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H/010C2H）的设置。

备注1.×：忽略

- 2.有关LVD电路的详细内容，请参照“第22章 电压检测电路”。
- 3.检测电压是TYP值。详细内容请参照数据手册的LVD电路特性。

图 25-6 用户选项字节（000C2H/010C2H）的格式

地址：000C2H/010C2H注

7	6	5	4	3	2	1	0
1	1	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

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

注 在引导交换时，000C2H被010C2H替换，因此必须给010C2H设置和000C2H相同的值。

注意1.必须给bit7~5写“1”。

2.工作频率范围和工作电压范围因闪存的各运行模式而不同。详细内容请参照数据手册的AC特性。

25.3 闪存数据保护选项字节的格式

闪存数据保护选项字节的格式如下所示。

图 25-7 闪存数据保护选项字节（000C3H/010C3H）的格式

地址：000C3H/010C3H^注



地址：500004H



OCDM	OCDEN	闪存数据保护的 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：在引导交换时，000C3H被010C3H替换，因此必须给010C3H设置和000C3H相同的值。

注意：50_0004H，50_0005H地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

地址：500005H



BTEN	引导交换功能的 control
0	使用引导交换功能，0000H~0FFFH和1000H~1FFFH内容交换
1	禁止交换功能使能

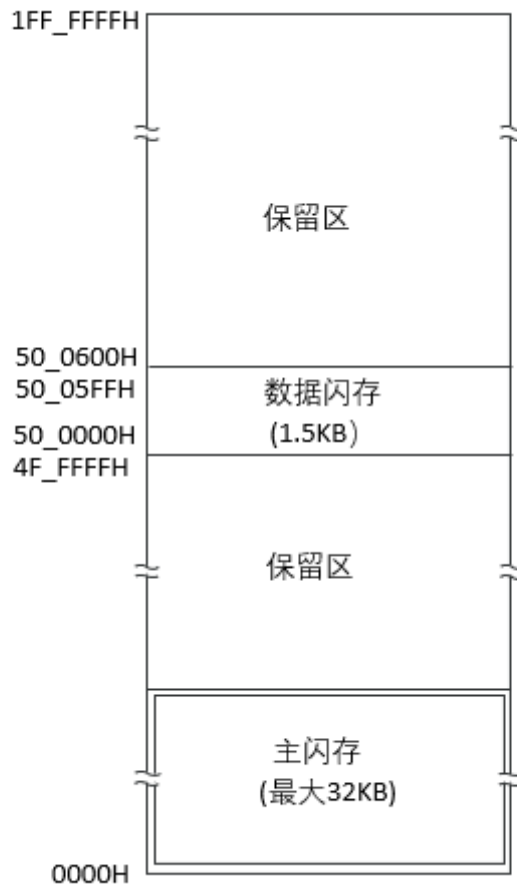
注意 50_0004H，50_0005H地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

第26章 FLASH控制

26.1 FLASH控制功能描述

本制品包含一颗32KByte容量的FLASH存储器，共划分为64个Sector，每个Sector容量为512Byte。可作为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

26.2 FLASH存储器结构



26.3 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)

26.3.1 Flash写保护寄存器(FLPROT)

Flash保护寄存器用于对FLASH操作控制寄存器进行保护。

地址：0x40020020 复位后：00000000H RW

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

WRP	操作寄存器 (FLOPMD1/FLOPMD2) 写保护
0	不允许改写 FLOPMD1/ FLOPMD2
1	允许改写 FLOPMD1/ FLOPMD2

PRKEY[7:1]	WRP写保护
78h	不允许改写 WRP
1	允许改写 WRP

26.3.2 FLASH操作控制寄存器 (FLOPMD1,FLOPMD2)

Flash操作控制寄存器，用于设置FLASH的擦除和写入操作。

地址：0x40020004 复位后：00000000H RW

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 RW

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		设置禁止

26.3.3 Flash擦除控制寄存器(FLERMD)

Flash擦除控制寄存器，用于设置FLASH擦除操作的类型。

地址：0x4002000C 复位后：00H RW

符号	7	6	5	4	3	2	1	0
FLERMD	0	0	0	ERMD1	ERMD0	0	0	0

ERMD1	ERMD0	操作
0	0	sector 擦除，擦除后不进行硬件校验
1	0	sector 擦除，擦除后进行硬件校验
0	1	chip 擦除注
1	1	设置禁止

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。

26.3.4 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注	0	OVF注

OVF	FLASH 擦写操作完了标志
0	FLASH 擦写操作未完成
1	FLASH 擦写操作完成

注: OVF需要软件写“1”进行清除。若不清除, 不能进行下一次擦写操作。

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。

26.3.5 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

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	擦除时间设置的选择 注
0	使用硬件设置的擦除时间
1	使用软件设置的擦除时间 (FLCERCNT[9:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLCERCNT。

FLCERCNT[9:0]	软件擦除时间设置
Chip 擦除时间= (CERCNT*2048*Tfclk), 需满足 $>20\text{ms}$ 的硬件要求	

26.3.6 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

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

Load	擦除时间设置的选择 注
0	使用硬件设置的擦除时间
1	使用软件设置的擦除时间（FLSERCNT[9:0]）

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时，可以使用硬件设置时间，不设置FLSERCNT。

FLSERCNT[9:0]	软件擦除时间设置
sector 擦除时间=（SERCNT*256*Tfclk），需满足 $>4\text{ms}$ 的硬件要求	

26.3.7 Flash 写入时间控制寄存器 (FLPROCNT)

通过FLPROCNT寄存器可以设置FLASH WORD写入的时间。

地址: 0x4002001C 复位后: 不定 R/W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Load1	-	-	-	-	-	-	FLPGSCNT[8:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Load0	-	-	-	-	-	-	FLPROCNT[8:0]								

Load0	写入时间 (Tprog) 设置 注
0	使用硬件设置的写入时间
1	使用软件设置的擦除时间 (FLPROCNT[9:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLPROCNT。

FLPROCNT[8:0]	软件擦除时间设置
写入时间= (PROCNT*4*Tfclk), 需满足 $>24\mu\text{s}$ 的硬件要求	

Load1	写入动作建立时间 (Tpgs) 设置 注
0	使用硬件设置的写入动作建立时间
1	使用软件设置的擦除时间 (FLPGSCNT8:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLPGSCNT。

FLPGSCNT[8:0]	软件擦除时间设置
写入动作建立时间= (PGSCNT*Tfclk), 需满足 $>5\mu\text{s}$ 的硬件要求	

26.3.8 Flash 模式时间控制寄存器 (FLNVSCNT/FLPRVCNT/FLERVCNT)

FLASH写入/擦除时，相关信号通过FLNVSCNT寄存器可以设置FLASH地址的setup时间 (Tnvs)。

地址: 0x40020018 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		
load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
-	-	-	-	-	-	-	FLNVSCNT[8:0]									-	-

FLNVSCNT

Load	SETUP 时间设置的选择 注
0	使用硬件设置的时间
1	使用软件设置的擦除时间 (FLNVSCNT[8:0])

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时，可以使用硬件设置时间，不设置FLNVSCNT。

FLNVSCNT[8:0]	软件擦除时间设置
setup 擦除时间= (NVSCNT*Tfclk)，需满足 $>6\mu\text{s}$ 的硬件要求	

通过FLPRVCNT寄存器可以设置FLASH写入时的recovery时间 (Trcv)。

地址: 0x40020038 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		
load	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
-	-	-	-	-	-	-	FLPRVCNT[8:0]									-	-

FLPRVCNT

Load	Program 的 recovery 时间设置的选择 注
0	使用硬件设置的时间
1	使用软件设置的擦除时间 (FLPRVCNT[8:0])

注：在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时，可以使用硬件设置时间，不设置FLPRVCNT。

FLPRVCNT[8:0]	软件擦除时间设置
recovery 时间= (PRVCNT*Tfclk)，需满足 $>5\mu\text{s}$ 的硬件要求	

通过FLERVCNT寄存器可以设置FLASH擦除时的recovery时间 (Trcv)。

地址: 0x4002003C 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
Load1	-	-	-	-	-	FLCRVCNT[25:16]									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Load0	-	-	-	-	-	-	FLSRVCNT[8:0]								

FLERVCNT

Load1	Chip 擦除时的 recovery 时间设置的选择 注
0	使用硬件设置的时间
1	使用软件设置的擦除时间 (FLCRVCNT[25:16])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLCRVCNT。

FLCRVCNT[8:0]	软件擦除时间设置
recovery 时间= (CRVCNT*16*Tfclk), 需满足 $>200\mu\text{s}$ 的硬件要求	

Load0	Sector 擦除时的 recovery 时间设置的选择 注
0	使用硬件设置的时间
1	使用软件设置的擦除时间 (FLSRVCNT[8:0])

注: 在主时钟为内部高速OCO或者外部输入时钟 $\leq 20\text{M}$ 时, 可以使用硬件设置时间, 不设置FLSRVCNT。

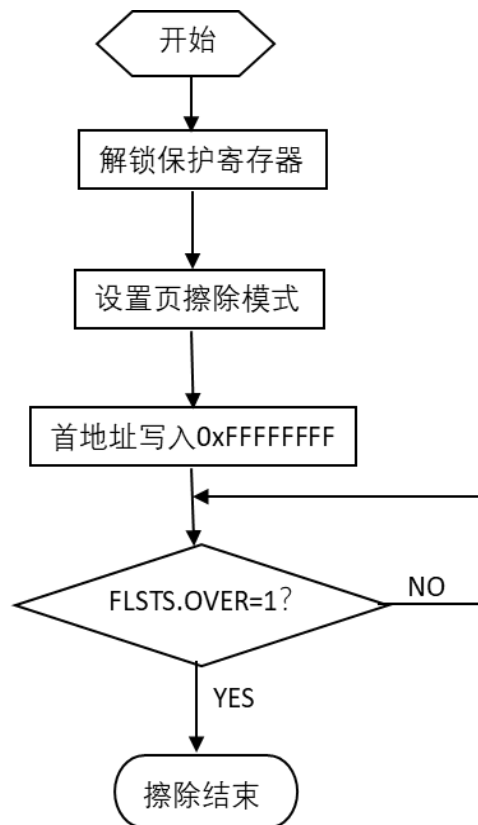
FLSRVCNT[8:0]	软件擦除时间设置
recovery 时间= (SRVCNT*8*Tfclk), 需满足 $>50\mu\text{s}$ 的硬件要求	

26.4 FLASH操作方法

26.4.1 页擦除 (sector erase)

sector擦除，擦除时间由硬件实现，也可以通过FLSERCNT配置。操作流程如下：

- 1) 设置FLERMD.ERMD0为1'b0，选择sector擦除模式,根据是否需要硬件校验选择设置ERMD1的值；
- 2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，
- 3) 向擦除目标sector的首地址写入任意数据。例：`* ((unsigned long *) 0x00000200) = 0xffffffff`。
- 4) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示擦除操作完成。
- 5) 如果设置了擦除后进行硬件校验（ERMD1=1），可以通过软件判断FLSTS.EVF，查询是否校验正确。
- 6) 进行下次操作前，软件置“1”来清除FLSTS。



26.4.2 全片擦除 (chip erase)

chip擦除，擦除时间由硬件实现，也可以通过FLCERCNT配置。操作流程如下：

- 1) 设置FLERMD.ERMD0为1'b1，选择chip擦除模式；
- 2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，
- 3) 向代码闪存区域的任意地址写入任意数据。
- 4) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示擦除操作完成。
- 5) 进行下次操作前，软件置"1"来清除FLSTS。

26.4.3 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。

26.5 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。

26.6 FLASH操作的注意事项

- FLASH存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。擦写参数的设置可以由硬件实现，也可通过修改参数寄存器进行软件修改；在使用内部高速OCO，MAINOSC/外部输入时钟=20M时，推荐使用硬件设置的擦写参数，无需设置参数寄存器。
- 如果擦写操作从FLASH内执行，则CPU会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从RAM里执行，CPU不会停止取指，当前可以继续下一条指令。
- 在FLASH处于编程操作中时，如果CPU执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

附录 修订记录

版本	日期	修订内容
V1.0	2019年10月	初版做成
V1.01	2019年11月	全文笔误修改
V1.02	2020年4月	全文笔误修改
V1.03	2020年7月	13.1.1 SSPI的主控最大传送速率修改为fCLK/2
		“第24章 温度传感器”追加
V1.04	2021年8月	19.3 修改待机功能比较器运行状态 23.3 删除“仅RAM运行”
V1.05	2022年11月	修改4.4、4.3.8、4.4.3、4.6.1、25.2高速内部振荡器的频率选择寄存器的描述
V1.0.6	2023年9月	更正图22-1、图22-6中的错误
	2024年3月	修改5.1.2 Timer4多重PWM数量 修改6.3.4章节中寄存器RWAIT位的描述