



CMS32F759xx_737xx用户手册

基于ARM® Cortex®-M0+的超低功耗32位微控制器

V0.3.2

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CMS32F759xx_737xx微控制器产品的技术参考手册，技术参考手册是有关如何使用本系列产品的应用说明资料，包含各个功能模块的结构、功能描述、工作模式以及寄存器配置等详细信息,并对每种功能模块都有专门的章节进行介绍。

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

数据手册信息如下：

CMS32F759xx_737xx: CMS32F759xx_737xx_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、SysTick定时器和NVIC的详细说明，请参照对应ARM的文档。

目录

文档使用说明	2
第1章 CPU	17
1.1 概述	17
1.2 Cortex-M0+ 内核特性	17
1.3 调试特性	17
1.4 SWD 接口引脚	19
1.5 ARM 参考文档	20
第2章 引脚功能 (GPIO)	21
2.1 端口功能	21
2.2 GPIO 结构	21
2.3 端口复用功能	22
2.4 寄存器映射	24
2.4.1 控制功能寄存器映射	24
2.4.2 输出输入复用功能寄存器映射	25
2.4.3 特殊功能端口 RESINB 控制寄存器映射	27
2.5 寄存器说明	28
2.5.1 端口寄存器 (Px)	29
2.5.2 端口模式寄存器 (PMx)	31
2.5.3 上拉电阻选择寄存器 (PUx)	33
2.5.4 下拉电阻选择寄存器 (PDx)	35
2.5.5 端口输出模式寄存器 (POMx)	37
2.5.6 端口模式控制寄存器 (PMCx)	39
2.5.7 端口置位控制寄存器 (PSETx)	41
2.5.8 端口清零控制寄存器 (PCLRx)	43
2.5.9 端口回读寄存器 (PREADx)	45
2.5.10 TTL 和施密特输入选择 (PTTLx)	47
2.5.11 端口输出复用功能配置寄存器 (PS_XX)	49
2.5.12 特殊功能端口 RESINB 说明 (RSTM)	73
第3章 系统结构	74
3.1 概述	74
3.2 系统地址划分	75
第4章 时钟发生电路	77
4.1 时钟发生电路的功能	77
4.2 时钟发生电路的结构	79
4.3 寄存器映射	81
4.4 寄存器说明	82
4.4.1 系统时钟控制寄存器 (CKC)	82
4.4.2 时钟运行状态控制寄存器 (CSC)	83
4.4.3 PLL 控制寄存器 (PLLCR)	84

4.4.4	系统时钟主控制寄存器(MCKC)	85
4.4.5	时钟运行模式控制寄存器(CMC)	86
4.4.6	振荡稳定时间选择寄存器(OSTS)	87
4.4.7	振荡稳定时间计数器的状态寄存器 (OSTC)	88
4.4.8	外围允许寄存器 (PER0、PER10、PER13、PER14)	90
4.4.9	15 位间隔定时器运行时钟选择寄存器 (OSMC)	93
4.4.10	高速内部振荡器的频率选择寄存器 (HOCODIV)	94
4.4.11	低速内部振荡器时钟选择寄存器 (SUBCKSEL)	95
4.4.12	发振停止检测控制寄存器 (SCMCTL)	95
4.4.13	发振停止检测模式寄存器 (SCMMD)	96
4.4.14	发振停止检测状态寄存器 (SCMST)	96
4.5	系统时钟振荡电路	97
4.5.1	X1 振荡电路	97
4.5.2	XT1 振荡电路	98
4.5.3	高速内部振荡器	98
4.5.4	低速内部振荡器	99
4.5.5	PLL 锁相环	99
4.6	时钟发生电路的运行	100
4.7	时钟控制	101
4.7.1	高速内部振荡器的设置例子	101
4.7.2	X1 振荡电路的设置例子	103
4.7.3	XT1 振荡电路的设置例子	104
4.7.4	CPU 时钟的状态转移图	105
4.7.5	CPU 时钟转移前的条件和转移后的处理	108
4.7.6	CPU 时钟和主系统时钟的切换所需时间	110
4.7.7	时钟振荡停止前的条件	111
4.8	发振停止检出电路	112
4.8.1	发振停止检出电路的功能	112
4.8.2	发振停止检测电路的构成	112
4.8.3	发振停止检测电路的动作	113
4.8.4	发振停止检测功能的注意事项	114
第5章	硬件除法器 (DIV)	115
5.1	特点	115
5.2	功能描述	115
5.3	寄存器映射	115
5.4	寄存器说明	116
5.4.1	被除数寄存器 (DIVIDEND)	116
5.4.2	除数寄存器 (DIVISOR)	116
5.4.3	商寄存器 (QUOTIENT)	116
5.4.4	余数寄存器 (REMAINDER)	116
5.4.5	状态寄存器 (STATUS)	116
第6章	通用定时器单元Timer4	117

6.1	通用定时器的功能	118
6.1.1	独立通道运行功能.....	118
6.1.2	多通道联动运行功能.....	120
6.1.3	8 位定时器运行功能（只限于通道 1 和通道3）	121
6.1.4	LIN-bus 支持功能（只限于单元0 的通道3）	121
6.2	通用定时器单元的结构	122
6.2.1	寄存器映射.....	125
6.2.2	定时器计数寄存器 mn（TCRmn）	127
6.2.3	定时器数据寄存器 mn（TDRmn）	128
6.2.4	外围允许寄存器 0（PER0）	129
6.2.5	定时器时钟选择寄存器 m（TPSm）	130
6.2.6	定时器模式寄存器 mn（TMRmn）	132
6.2.7	定时器状态寄存器 mn（TSRmn）	140
6.2.8	定时器通道允许状态寄存器 m（TEm）	141
6.2.9	定时器通道开始寄存器 m（TSM）	142
6.2.10	定时器通道停止寄存器 m（TTm）	143
6.2.11	定时器输入输出选择寄存器（TIOS0）	144
6.2.12	定时器输出允许寄存器 m（TOEm）	145
6.2.13	定时器输出寄存器 m（TOM）	146
6.2.14	定时器输出电平寄存器 m（TOLm）	146
6.2.15	定时器输出模式寄存器 m（TOMm）	147
6.2.16	噪声滤波器允许寄存器 1（NFEN1）	148
6.2.17	噪声滤波器允许寄存器 2（NFEN2）	149
6.2.18	控制定时器输入/输出引脚端口功能的寄存器.....	150
6.3	通用定时器单元的基本规则	151
6.3.1	多通道联动运行功能的基本规则.....	151
6.3.2	8 位定时器运行功能的基本规则（只限于通道 1 和通道 3）	153
6.4	计数器的运行.....	154
6.4.1	计数时钟（f _{TCLK} ）	154
6.4.2	计数器的开始时序.....	156
6.4.3	计数器的运行.....	157
6.5	通道输出（TOMn 引脚）的控制	162
6.5.1	TOMn 引脚输出电路的结构.....	162
6.5.2	TOMn 引脚的输出设定.....	163
6.5.3	通道输出运行的注意事项.....	164
6.5.4	TOMn 位的一次性操作.....	168
6.5.5	有关开始计数时的定时器中断和 TOMn 引脚输出.....	169
6.6	定时器输入（TIMn）的控制	170
6.6.1	TIMn 引脚输入电路的结构.....	170
6.6.2	噪声滤波器.....	170
6.6.3	操作通道输入时的注意事项.....	171
6.7	通用定时器单元的独立通道运行功能	171
6.7.1	作为间隔定时器/ 方波输出的运行	171

6.7.2	作为外部事件计数器的运行.....	175
6.7.3	作为分频器的运行.....	178
6.7.4	作为输入脉冲间隔测量的运行.....	181
6.7.5	作为输入信号高低电平宽度测量的运行.....	184
6.7.6	作为延迟计数器的运行.....	188
6.8	通用定时器单元的多通道联动运行功能.....	191
6.8.1	作为单触发脉冲输出功能的运行.....	191
6.8.2	作为 PWM 功能的运行.....	197
6.8.3	作为多重 PWM 输出功能的运行.....	204
第7章	15位间隔定时器.....	212
7.1	15 位间隔定时器的功能.....	212
7.2	15 位间隔定时器的结构.....	212
7.3	寄存器映射.....	213
7.4	寄存器说明.....	214
7.4.1	外围允许寄存器 0 (PER0).....	214
7.4.2	实时时钟选择寄存器 (RTCCL).....	215
7.4.3	15 位间隔定时器的控制寄存器 (ITMC).....	216
7.5	15 位间隔定时器的运行.....	217
7.5.1	15 位间隔定时器的运行时序.....	217
7.5.2	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.....	218
第8章	实时时钟 (RTC).....	219
8.1	实时时钟的功能.....	219
8.2	实时时钟的结构.....	219
8.3	寄存器映射.....	221
8.4	寄存器说明.....	222
8.4.1	外围允许寄存器 0 (PER0).....	222
8.4.2	实时时钟选择寄存器 (RTCCL).....	223
8.4.3	实时时钟控制寄存器 0 (RTCC0).....	224
8.4.4	实时时钟控制寄存器 1 (RTCC1).....	225
8.4.5	秒计数寄存器 (SEC).....	226
8.4.6	分钟计数寄存器 (MIN).....	226
8.4.7	小时计数寄存器 (HOUR).....	227
8.4.8	日计数寄存器 (DAY).....	229
8.4.9	星期计数寄存器 (WEEK).....	230
8.4.10	月计数寄存器 (MONTH).....	230
8.4.11	年计数寄存器 (YEAR).....	231
8.4.12	闹钟分钟寄存器 (ALARMWM).....	232
8.4.13	闹钟小时寄存器 (ALARMWH).....	232
8.4.14	闹钟星期寄存器 (ALARMWW).....	233
8.5	实时时钟的运行.....	234
8.5.1	实时时钟的运行开始.....	234
8.5.2	开始运行后睡眠模式的转移.....	235

8.5.3	实时时钟计数器的读写.....	236
8.5.4	实时时钟的闹钟设置.....	237
第9章	时钟输出/蜂鸣器输出控制电路.....	238
9.1	时钟输出/蜂鸣器输出控制电路的功能.....	238
9.2	时钟输出/蜂鸣器输出控制电路的结构.....	239
9.3	寄存器映射.....	239
9.3.1	时钟输出选择寄存器（CKS0）.....	240
9.3.2	时钟输出选择寄存器（CKS1）.....	241
9.4	配置时钟输出/蜂鸣器输出引脚端口功能的寄存器.....	242
9.5	时钟输出/蜂鸣器输出控制电路的运行.....	243
9.5.1	输出引脚的运行.....	243
9.6	时钟输出/蜂鸣器输出控制电路的注意事项.....	243
第10章	独立看门狗定时器（IWDG）.....	244
10.1	概述.....	244
10.1.1	独立看门狗定时器概述.....	244
10.2	特性.....	244
10.2.1	独立看门狗定时器特性.....	244
10.3	独立看门狗定时器结构.....	245
10.4	寄存器映射.....	247
10.5	寄存器说明.....	248
10.5.1	独立看门狗定时器的允许寄存器（WDTE）.....	248
10.5.2	LOCKUP 控制寄存器（LOCKCTL）.....	248
10.5.3	保护寄存器（PRCR）.....	249
10.6	独立看门狗定时器的运行.....	250
10.6.1	独立看门狗定时器的运行控制.....	250
10.6.2	独立看门狗定时器溢时间的设定.....	251
10.6.3	独立看门狗定时器窗口打开期间的设定.....	252
10.6.4	独立看门狗定时器间隔中断的设定.....	253
10.6.5	LOCKUP 期间独立看门狗定时器的运行.....	253
第11章	窗口看门狗定时器（WWDG）.....	254
11.1	窗口看门狗定时器的功能.....	254
11.2	窗口看门狗定时器的结构.....	254
11.3	寄存器映射.....	256
11.3.1	窗口看门狗定时器的清零寄存器（WDTE）.....	256
11.3.2	窗口看门狗控制寄存器(WWDTOPT).....	257
11.3.3	窗口看门狗分频寄存器(WWDTDIV).....	257
11.4	窗口看门狗定时器的运行.....	258
11.4.1	窗口看门狗定时器的运行控制.....	258
11.4.2	窗口看门狗定时器上溢时间的设定.....	259
11.4.3	窗口看门狗定时器窗口打开期间的设定.....	260
11.4.4	窗口看门狗定时器间隔中断的设定.....	261

第12章	EPWM输出控制电路	262
12.1	输出控制电路的功能	262
12.2	输出控制电路的结构	262
12.3	寄存器映射	263
12.3.1	EPWM 输入源选择寄存器(EPWMSRC)	263
12.3.2	EPWM 输出控制寄存器(EPWMCTL)	264
12.3.3	EPWM 强制截断输入选择寄存器(EPWMSTC)	265
12.3.4	EPWM 强制截断输出选择寄存器(EPWMSTL)	266
12.3.5	EPWM 状态寄存器(EPWMSTR)	267
12.3.6	EPWM 死区控制寄存器 0(EPWMDTC0)	267
12.3.7	EPWM 死区控制寄存器 1(EPWMDTC1)	268
12.3.8	EPWM 输出引脚的端口功能的控制寄存器	268
12.4	EPWM 输出控制电路的运行	269
12.4.1	初始设定	269
12.4.2	通常运行	270
12.4.3	强制截断处理	270
12.5	无刷直流电机的控制例子	272
12.5.1	硬件连接例子	272
12.5.2	三相无刷直流电机的控制时序	273
12.5.3	寄存器的设定例子	274
12.6	步进电机的控制例子	275
12.6.1	硬件连接例子	275
12.6.2	控制方法	276
12.6.3	寄存器的设定例子	277
第13章	通用串行通信单元	278
13.1	通用串行通信单元的功能	279
13.1.1	3 串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)	279
13.1.2	带从属片选功能的简易 SPI	280
13.1.3	UART (UART0~UART5)	281
13.1.4	简易 I ² C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)	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)	287
13.3	控制通用串行通信单元的寄存器	288
13.4	外围允许寄存器 (PER0、PER14)	290
13.4.1	串行时钟选择寄存器 m (SPSm)	292
13.4.2	串行模式寄存器 mn (SMRmn)	293
13.4.3	串行通信运行设定寄存器 mn (SCRmn)	294
13.4.4	串行数据寄存器 mn (SDRmn)	296
13.4.5	串行标志清除触发寄存器 mn (SIRmn)	297
13.4.6	串行状态寄存器 mn (SSRmn)	298
13.4.7	串行通道开始寄存器 m (SSm)	300

13.4.8	串行通道停止寄存器 m (STm)	301
13.4.9	串行通道允许状态寄存器 m (SEm)	302
13.4.10	串行输出允许寄存器 m (SOEm)	303
13.4.11	串行输出寄存器 m (SOm)	304
13.4.12	串行输出电平寄存器 m (SOLm)	305
13.4.13	输入切换控制寄存器 (ISC)	306
13.4.14	噪声滤波器允许寄存器 0 (NFEN0)	307
13.5	控制串行输入/输出引脚端口功能的寄存器	308
13.6	运行停止模式	309
13.6.1	以单元为单位停止运行的情况	309
13.6.2	按通道停止运行的情况	311
13.7	3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21) 通信的运行	312
13.7.1	主控发送	313
13.7.2	主控接收	321
13.7.3	主控的发送和接收	330
13.7.4	从属发送	338
13.7.5	从属接收	346
13.7.6	从属的发送和接收	352
13.7.7	传送时钟频率的计算	361
13.7.8	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	363
13.8	从属选择输入功能的时钟同步串行通信的运行	364
13.8.1	从属发送	367
13.8.2	从属接收	377
13.8.3	从属的发送和接收	384
13.8.4	传送时钟频率的计算	394
13.8.5	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	395
13.9	UART (UART0~UART5) 通信的运行	396
13.9.1	UART 发送	397
13.9.2	UART 接收	405
13.9.3	波特率的计算	412
13.9.4	在 UART (UART0~UART5) 通信过程中发生错误时的处理步骤	416
13.10	LIN 通信的运行	417
13.10.1	LIN 发送	417
13.10.2	LIN 接收	420
13.11	简易 I2C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 通信的运行	425
13.11.1	地址段发送	426
13.11.2	数据发送	431
13.11.3	数据接收	434
13.11.4	停止条件的产生	438
13.11.5	传送速率的计算	439

第14章 串行接口 IICA (IICA) 442

14.1	串行接口 IICA 的功能	442

14.2	串行接口 IICA 的结构	445
14.3	寄存器映射	448
14.3.1	外围允许寄存器 0 (PER0)	449
14.3.2	IICA 控制寄存器 n0 (IICCTLn0)	450
14.3.3	IICA 状态寄存器 n (IICSn)	453
14.3.4	IICA 标志寄存器 n (IICFn)	456
14.3.5	IICA 控制寄存器 n1 (IICCTLn1)	458
14.3.6	IICA 低电平宽度设定寄存器 n (IICWLn)	460
14.3.7	IICA 高电平宽度设定寄存器 n (IICWHn)	460
14.3.8	控制 IICA 引脚端口功能的寄存器	460
14.4	I2C 总线模式的功能	461
14.4.1	引脚结构	461
14.4.2	通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法	462
14.5	I ² C 总线的定义和控制方法	464
14.5.1	开始条件	465
14.5.2	地址	466
14.5.3	传送方向的指定	466
14.5.4	应答 (ACK)	467
14.5.5	停止条件	468
14.5.6	等待	469
14.5.7	等待的解除方法	471
14.5.8	中断请求 (INTIICAn) 的产生时序和等待控制	472
14.5.9	地址匹配的检测方法	473
14.5.10	错误的检测	473
14.5.11	扩展码	473
14.5.12	仲裁	474
14.5.13	唤醒功能	476
14.5.14	通信预约	479
14.5.15	其他注意事项	483
14.5.16	通信运行	484
14.5.17	I ² C 中断请求 (INTIICAn) 的产生时序	492
14.6	时序图	510
第15章	增强型DMA	525
15.1	DMA 的功能	525
15.2	DMA 的结构	527
15.3	寄存器映射	528
15.4	控制 DMA 的寄存器	529
15.4.1	DMA 控制数据区和 DMA 向量表区的分配	529
15.4.2	控制数据的分配	530
15.4.3	向量表	532
15.4.4	外围允许寄存器 10 (PER10)	534
15.4.5	DMA 通道触发源选择寄存器 j (DMA_CHj_SEL) (j=0~7)	534

15.4.6	DMA 控制寄存器 j (DMACRj) (j=0~7)	536
15.4.7	DMA 块大小寄存器 j (DMBLSj) (j=0~7)	537
15.4.8	DMA 传送次数寄存器 j (DMACTj) (j=0~7)	538
15.4.9	DMA 传送次数重加载寄存器 j (DMRLDj) (j=0~7)	539
15.4.10	DMA 源地址寄存器 j (DMSARj) (j=0~7)	540
15.4.11	DMA 目标地址寄存器 j (DMDARj) (j=0~7)	540
15.4.12	DMA 启动允许寄存器 i (DMAEN0)	541
15.4.13	DMA 基址寄存器 (DMABAR)	542
15.4.14	DMAEN0 置位寄存器 (DMASET0)	543
15.4.15	DMAEN0 复位寄存器 (DMACLR0)	544
15.5	DMA 的运行	545
15.5.1	启动源	545
15.5.2	正常模式	546
15.5.3	重复模式	549
15.5.4	链传送	552
15.6	使用 DMA 时的注意事项	554
15.6.1	DMA 控制数据和向量表的设置	554
15.6.2	DMA 控制数据区和 DMA 向量表区的分配	554
15.6.3	DMA 的执行时钟数	555
15.6.4	DMA 的响应时间	556
15.6.5	DMA 的启动源	556
15.6.6	待机模式中的运行	556
第16章	快速模数转换器 (ADC)	557
16.1	快速模数转换器的功能	557
16.2	寄存器映射	560
16.3	寄存器说明	561
16.3.1	A/D 转换器的模式寄存器 0 (ADM0)	561
16.3.2	A/D 转换器的模式寄存器 1 (ADM1)	565
16.3.3	A/D 转换器的模式寄存器 2 (ADM2)	566
16.3.4	A/D 转换器的触发模式寄存器 (ADTRG)	567
16.3.5	模拟输入通道指定寄存器 (ADS)	568
16.3.6	A/D 内部通道选择寄存器 (ADICHS)	571
16.3.7	12 位 A/D 转换通道标识符寄存器 (ADSCH)	572
16.3.8	12 位 A/D 转换结果寄存器 (ADCR)	572
16.3.9	8 位 A/D 转换结果寄存器 (ADCRH)	573
16.3.10	转换结果比较上限值设置寄存器 (ADUL)	573
16.3.11	转换结果比较下限值设置寄存器 (ADLL)	573
16.3.12	A/D 采样时间控制寄存器 (ADNSMP)	574
16.3.13	A/D 采样时间延长寄存器 (ADSMPWAIT)	575
16.3.14	A/D 测试寄存器 (ADTES)	575
16.3.15	A/D 状态寄存器 (ADFLG)	576
16.3.16	A/D 充放电控制寄存器 (ADNDIS)	576

16.4	输入电压和转换结果	577
16.5	A/D 转换器的运行模式	578
16.5.1	软件触发模式（选择模式、连续转换模式）	578
16.5.2	软件触发模式（选择模式、单次转换模式）	579
16.5.3	软件触发模式（扫描模式、连续转换模式）	580
16.5.4	软件触发模式（扫描模式、单次转换模式）	581
16.5.5	硬件触发无等待模式（选择模式、连续转换模式）	582
16.5.6	硬件触发无等待模式（选择模式、单次转换模式）	583
16.5.7	硬件触发无等待模式（扫描模式、连续转换模式）	584
16.5.8	硬件触发无等待模式（扫描模式、单次转换模式）	585
16.5.9	硬件触发等待模式（选择模式、连续转换模式）	586
16.5.10	硬件触发等待模式（选择模式、单次转换模式）	587
16.5.11	硬件触发等待模式（扫描模式、连续转换模式）	588
16.5.12	硬件触发等待模式（扫描模式、单次转换模式）	589
16.6	A/D 转换器的设置流程图	590
16.6.1	软件触发模式的设置	590
16.6.2	硬件触发无等待模式的设置	591
16.6.3	硬件触发等待模式的设置	592
16.6.4	选择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）	593
16.6.5	测试模式的设置	594
第17章	模拟比较器（ACMP0/1）	595
17.1	概述	595
17.2	特性	595
17.3	结构框图	596
17.4	寄存器映射	597
17.5	寄存器说明	598
17.5.1	模拟比较器 0 控制寄存器 0(C0CON0)	598
17.5.2	模拟比较器 0 控制寄存器 1(C0CON1)	599
17.5.3	模拟比较器 1 控制寄存器 0(C1CON0)	600
17.5.4	模拟比较器 1 控制寄存器 1(C1CON1)	601
17.5.5	模拟比较器事件控制寄存器(CEVCON)	602
17.5.6	模拟比较器写使能控制寄存器(ACMPLOCK)	602
第18章	数模转换器（DAC）	603
18.1	概述	603
18.2	特性	603
18.3	结构框图	603
18.4	寄存器映射	603
18.5	寄存器说明	604
18.5.1	DAC 控制寄存器 0(DACCON0)	604
18.5.2	DAC 写使能控制寄存器(DACLOCK)	604
第19章	硬件LED矩阵驱动器	605

19.1	概述	605
19.2	特性	605
19.3	寄存器映射	605
19.4	相关寄存器	606
19.4.1	LED 控制寄存器 (LEDCON)	606
19.4.2	COM 口有效时间选择寄存器 (LEDCOMTIME)	607
19.4.3	调光时间选择寄存器 (LEDSEGTIME)	607
19.4.4	COM 口使能控制寄存器 (LEDCOMEN)	607
19.4.5	SEG 口使能控制寄存器 (LEDSEGEN)	607
19.4.6	COM 口对应 SEG 数据寄存器 (LEDCnDATA) (n=0-7)	607
19.4.7	COM 口灌电流选择寄存器 (LEDCOMDR)	608
19.4.8	SEG 口驱动电流选择寄存器 (LEDSEGDR)	609
19.4.9	LED 管脚驱动使能寄存器 (LEDDRVEN0)	610
19.4.10	LED 寄存器写使能控制寄存器 (LEDLOCK)	612
19.5	LED 驱动器输出波形	613
第20章	硬件LCD驱动器	614
20.1	概述	614
20.2	特性	614
20.3	寄存器映射	615
20.4	相关寄存器	617
20.4.1	LCD 控制寄存器 (LCDCON0)	617
20.4.2	LCD 控制寄存器 (LCDCON1)	619
20.4.3	LCDCOM 口使能控制寄存器 (LCDCOMEN)	619
20.4.4	LCDSEG 口使能控制寄存器 (LCDSEGEN)	620
20.4.5	SEG 数据寄存器 (LCDSEGN) (n=0~39)	620
20.4.6	LCDIO 口使能控制寄存器 0 (LCDIOEN0)	620
20.4.7	LCDIO 口使能控制寄存器 1 (LCDIOEN1)	621
20.4.8	LCD 寄存器写使能控制寄存器 (LCDLOCK)	622
20.5	COM-SEG 数据表	623
20.5.1	1/4DUTY	623
20.5.2	1/5DUTY	624
20.5.3	1/6DUTY	625
20.5.4	1/8DUTY	626
第21章	联动控制器 (EVENTC)	627
21.1	EVENTC 的功能	627
21.2	EVENTC 的结构	627
21.3	寄存器映射	628
21.3.1	输出目标选择寄存器 n (ELSELRn) (n=00~22)	629
21.4	EVENTC 的运行	631
第22章	嵌套向量中断控制器(NVIC)	632
22.1	特性	632

22.2	异常模式和系统中断映射	632
22.3	向量表	633
22.4	中断功能的基本结构	635
22.5	寄存器映射	637
22.6	寄存器说明	639
22.6.1	中断请求标志寄存器 (IF00~IF31)	639
22.6.2	中断屏蔽标志寄存器 (MK00~MK31)	640
22.6.3	外部中断上升沿允许寄存器 (EGP0)	642
22.6.4	外部中断下降沿允许寄存器 (EGN0)	643
22.6.5	外部中断上升沿允许寄存器 (EGP1)	644
22.6.6	外部中断下降沿允许寄存器 (EGN1)	645
22.7	中断处理的操作	646
22.7.1	可屏蔽中断请求的接受	646
22.7.2	不可屏蔽中断请求的接受	646
第23章	待机功能	647
23.1	待机功能	647
23.2	睡眠模式	648
23.2.1	睡眠模式的设定	648
23.2.2	睡眠模式的解除	650
23.3	深度睡眠模式	651
23.3.1	深度睡眠模式的设定	651
23.3.2	深度睡眠模式的解除	653
第24章	复位功能	654
24.1	复位时序	656
24.2	确认复位源的寄存器	658
24.2.1	寄存器映射	658
24.2.2	复位控制标志寄存器 (RESF)	658
第25章	上电复位电路	661
25.1	上电复位电路的功能	661
25.2	上电复位电路的结构	662
25.3	上电复位电路的运行	663
第26章	电压检测电路	666
26.1	电压检测电路的功能	666
26.2	电压检测电路的结构	667
26.3	寄存器映射	667
26.4	控制电压检测电路的寄存器	667
26.4.1	电压检测寄存器 (LVIM)	668
26.4.2	电压检测电平寄存器 (LVIS)	669
26.5	电压检测电路的运行	670
26.5.1	用作复位模式时的设定	670

26.6	用作中断模式时的设定	671
26.7	用作中断&复位模式时的设定	673
26.8	电压检测电路的注意事项	678
第27章	安全功能	680
27.1	安全功能的概要	680
27.2	寄存器映射	681
27.3	安全功能的运行	682
27.3.1	闪存 CRC 运算功能（高速 CRC）	682
27.3.2	CRC 运算功能（通用 CRC）	685
27.3.3	SFR 保护功能	687
27.3.4	频率检测功能	688
27.3.5	A/D 测试功能	689
27.3.6	输入/输出引脚的数字输出信号电平检测功能	690
27.3.7	产品唯一身份标识寄存器	691
第28章	选项字节	692
28.1	选项字节的功能	692
28.1.1	用户选项字节（000C0H~000C2H）	692
28.2	闪存数据保护选项字节（000C3H, 500004H）	693
28.3	寄存器映射	693
28.4	用户选项字节	694
28.4.1	用户选项字节 0（000C0H）	694
28.4.2	用户选项字节 1（000C1H）	695
28.4.3	用户选项字节 2（000C2H）	698
28.4.4	闪存数据保护选项字节 1（000C3H）	699
28.4.5	闪存数据保护选项字节 2（500004H）	699
第29章	FLASH控制	700
29.1	FLASH 控制功能描述	700
29.2	FLASH 存储器结构	700
29.3	寄存器映射	701
29.4	寄存器说明	702
29.4.1	Flash 操作控制寄存器(FLCMD)	702
29.4.2	FLASH 密钥寄存器(FLKEY)	702
29.4.3	FLASH 数据寄存器 0(FLDATA0)	702
29.4.4	FLASH 数据寄存器 1(FLDATA1)	702
29.4.5	Flash 地址寄存器(FLADDR)	702
29.4.6	Flash 控制寄存器(FLCR)	703
29.4.7	Flash 状态寄存器(FLSR)	704
29.4.8	Flash 访问控制寄存器（FLACR）	705
29.4.9	Flash 区域控制寄存器（FLREGION）	705
29.4.10	Flash 保护寄存器（FLPRT）	706
29.5	FLASH 操作方法	708

29.5.1	编程.....	708
29.5.2	扇擦除.....	709
29.5.3	全片擦除（chip erase）.....	710
29.6	FLASH 操作的注意事项.....	711
第30章	附录 修订记录.....	712

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

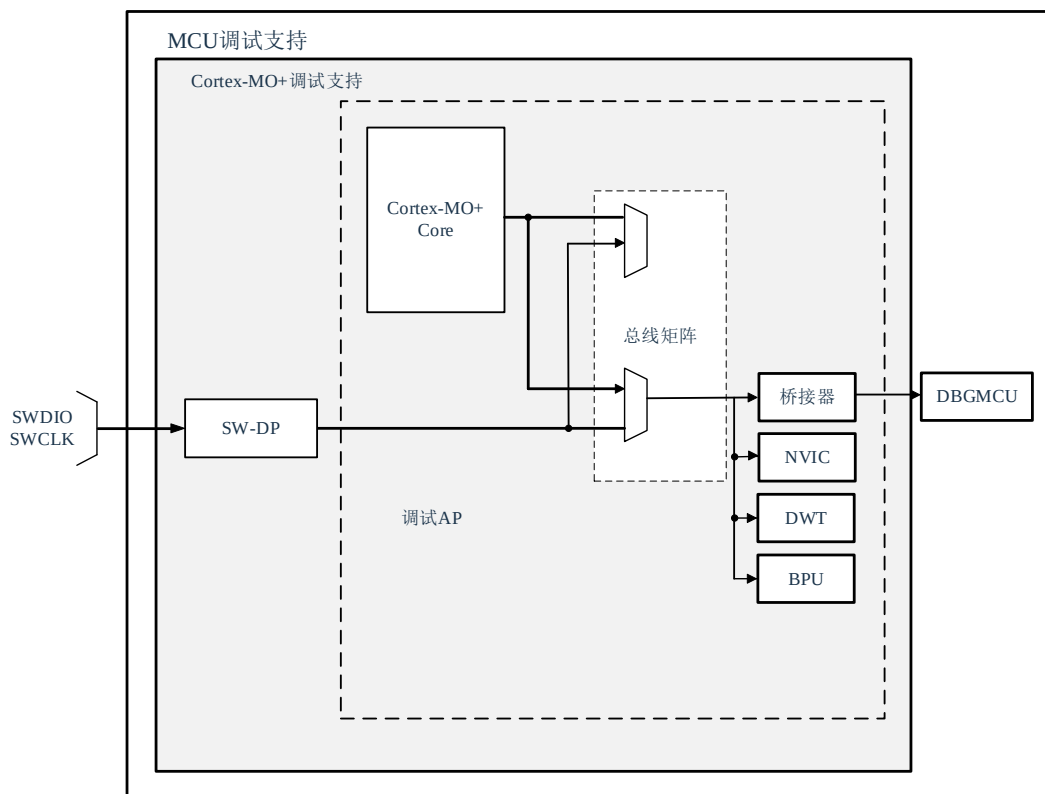
ARM Cortex-M0+处理器是32位RISC内核，采用2级流水线，仅支持特权模式

- 1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1: 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。

注2: 若使用SWD在线调试, SWDIO/SWDCLK严禁复用成触摸或LCD等模拟功能。

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	PC11/PD1
SWDIO	串行数据输入/输出	PC12/PD2

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。

位	符号	描述	复位值
31:25	-	保留	-
24	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，PC11/PC12或PD1/PD2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。PC11/PC12或PD1/PD2可用作GPIO	0
23:2	-	保留	0x0
1	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	0
0	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	0

注1：本制品的定时器系周边模块包括：通用定时器单元Timer4。

注2：本制品的通信系周边模块包括：通用串行通信单元,串行IICA。

注3：若使用SWD 在线调试，SWDIO/SWDCLK 严禁复用成触摸或LCD 等模拟功能。

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：

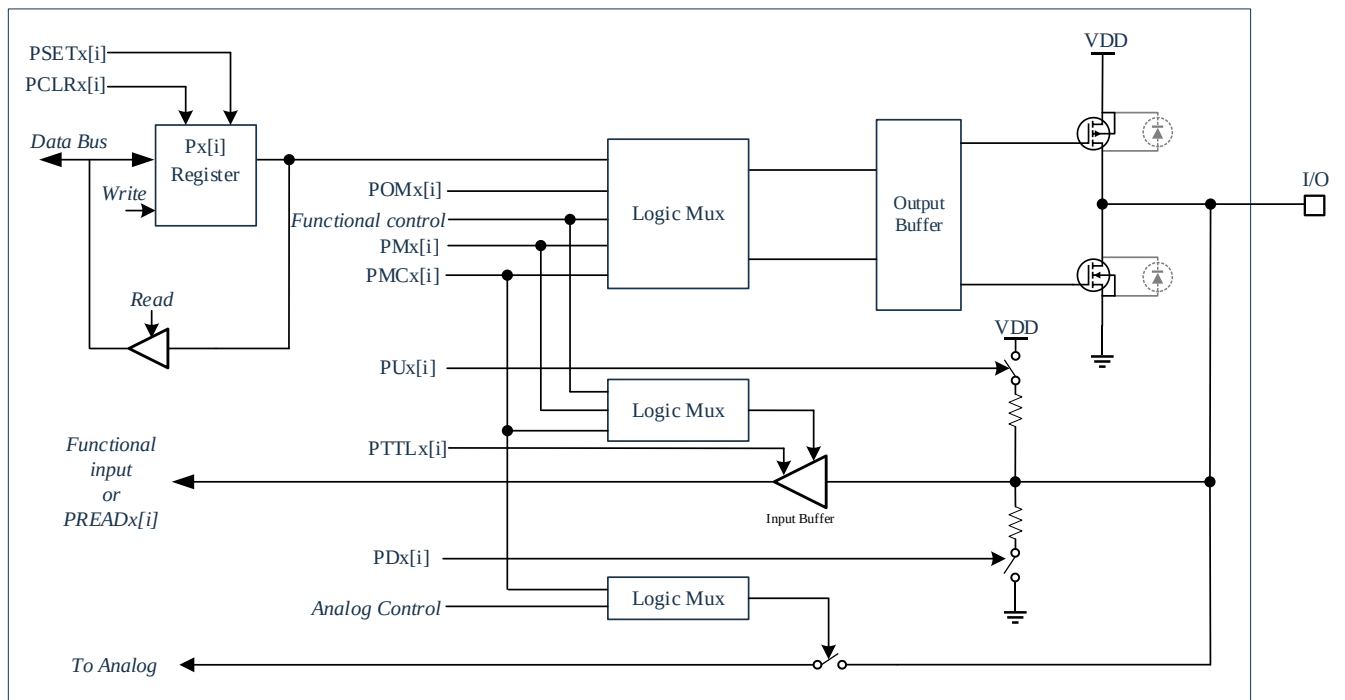
- Cortex®-M0+技术参考手册(TRM)
- ARM®调试接口V5
- ARM® CoreSight设计套件版本r1p1 技术参考手册
- ARM® CoreSight™ MTB-M0+ Technical Reference Manual

第2章 引脚功能（GPIO）

2.1 端口功能

具体端口功能参见各产品系列的数据手册。

2.2 GPIO结构



2.3 端口复用功能

具体端口复用功能参见各产品系列的数据手册。详细端口复用功能参见下表2-1。

表2-1 端口复用功能数字映射表

	ICA	SAU0										SAU1					SAU2	SAU3	SAU4	TI	TO	EPWM	BUZ/ACM
引脚	ICA	SP00	IC00	SP01	IC01	UART0	SP10	IC10	SP11	IC11	UART1	SP20	IC20	SP21	IC21	UART2	UART3	UART4	UART5				
PA0																	TXD3/RXD3			TI00	TO00	PWM0	BUZ
PA1	SCLA0																TXD3/RXD3			TI01	TO01	PWM1	BUZ
PA2	SDAA0																TXD3/RXD3			TI02	TO02	PWM2	
PA3		SS00															TXD3/RXD3			TI03	TO03	PWM3	BUZ
PA4		SCLK00	SCL00	SCLK01	SCL01		TXD0/ RXD0										TXD3/RXD3			TI10	TO10	PWM4	BUZ
PA5		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI11	TO11	PWM5	
PA6		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI12	TO12	PWM6	
VSS																							
VDD																							
PA7																	TXD3/RXD3			TI13	TO13	PWM7	
PA8																	TXD3/RXD3			TI00	TO00	PWM0	C1_O
PA9												SCLK20	SCL20	SCLK21	SCL21		TXD3/RXD3			TI01	TO01	PWM1	C0_O
PA10												SDI20/SDO20	SDA20	SDI21/SDO21	SDA21	TXD2/RXD2	TXD3/RXD3			TI02	TO02	PWM2	
PA11												SDI20/SDO20	SDA20	SDI21/SDO21	SDA21	TXD2/RXD2	TXD3/RXD3			TI03	TO03	PWM3	
VDD																							
VSS																							
PB0																	TXD3/RXD3			TI00	TO00	PWM0	BUZ
PB1																	TXD3/RXD3			TI01	TO01	PWM1	BUZ
PB2	SCLA0																TXD3/RXD3			TI02	TO02	PWM2	
PB3	SDAA0											SCLK20	SCL20	SCLK21	SCL21		TXD3/RXD3			TI03	TO03	PWM3	
PB4												SDI20/SDO20	SDA20	SDI21/SDO21	SDA21	TXD2/RXD2	TXD3/RXD3			TI10	TO10	PWM4	
PB5												SDI20/SDO20	SDA20	SDI21/SDO21	SDA21	TXD2/RXD2	TXD3/RXD3			TI11	TO11	PWM5	
PB6																	TXD3/RXD3			TI12	TO12	PWM6	
PB7																	TXD3/RXD3			TI13	TO13	PWM7	
PB8	SCLA0																TXD3/RXD3	TXD4/RXD4		TI00	TO00	PWM0	
PB9	SDAA0																TXD3/RXD3	TXD4/RXD4		TI01	TO01	PWM1	
PB10																	TXD3/RXD3		TXD5/RXD5	TI02	TO02	PWM2	BUZ
PB11																	TXD3/RXD3	TXD5/RXD5		TI03	TO03	PWM3	BUZ
PB12																	TXD3/RXD3			TI10	TO10	PWM4	
PB13																	TXD3/RXD3			TI11	TO11	PWM5	
PB14																	TXD3/RXD3			TI12	TO12	PWM6	
PB15																	TXD3/RXD3			TI13	TO13	PWM7	
PC0																	TXD3/RXD3	TXD4/RXD4		TI00	TO00	PWM0	BUZ
PC1																	TXD3/RXD3	TXD4/RXD4		TI01	TO01	PWM1	BUZ
PC2		SS00															TXD3/RXD3			TI02	TO02	PWM2	C1_O
PC3		SCLK00	SCL00	SCLK01	SCL01		TXD0/ RXD0										TXD3/RXD3			TI03	TO03	PWM3	C0_O
PC4		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI10	TO10	PWM4	
PC5		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI11	TO11	PWM5	
PC6																	TXD3/RXD3			TI12	TO12	PWM6	
PC7																	TXD3/RXD3			TI13	TO13	PWM7	C1_O
PC8																	TXD3/RXD3			TI00	TO00	PWM0	
PC9	SCLA0																TXD3/RXD3		TXD5/RXD5	TI01	TO01	PWM1	
PC10	SDAA0																TXD3/RXD3	TXD5/RXD5		TI02	TO02	PWM2	BUZ
PC11																	TXD3/RXD3			TI03	TO03	PWM3	BUZ
PC12																	TXD3/RXD3			TI10	TO10	PWM4	
PC13																	TXD3/RXD3			TI11	TO11	PWM5	
PC14							SDI10/ SDI010	SDA10	SDI11/ SDI011	SDA11		TXD1/ RXD1					TXD3/RXD3			TI12	TO12	PWM6	
PC15							SDI10/ SDI010	SDA10	SDI11/ SDI011	SDA11		TXD1/ RXD1					TXD3/RXD3			TI13	TO13	PWM7	
PD0																	TXD3/RXD3			TI00	TO00	PWM0	BUZ
PD1	SCLA0																TXD3/RXD3	TXD4/RXD4		TI01	TO01	PWM1	BUZ
PD2	SDAA0																TXD3/RXD3	TXD4/RXD4		TI02	TO02	PWM2	
PD3		SS00															TXD3/RXD3		TXD5/RXD5	TI03	TO03	PWM3	BUZ
PD4		SCLK00	SCL00	SCLK01	SCL01		TXD0/ RXD0										TXD3/RXD3		TXD5/RXD5	TI10	TO10	PWM4	BUZ
PD5		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI11	TO11	PWM5	
PD6		SDI00/ SDI000	SDA00	SDI01/ SDI001	SDA01		TXD0/ RXD0										TXD3/RXD3			TI12	TO12	PWM6	
PD7																	TXD3/RXD3			TI13	TO13	PWM7	C0_O
PD8																	TXD3/RXD3			TI00	TO00	PWM0	
PD9	SCLA0																TXD3/RXD3		TXD5/RXD5	TI01	TO01	PWM1	
PD10	SDAA0																TXD3/RXD3	TXD5/RXD5		TI02	TO02	PWM2	BUZ
PD11																	TXD3/RXD3			TI03	TO03	PWM3	BUZ
PD12																	TXD3/RXD3			TI10	TO10	PWM4	BUZ
PD13							SCLK10	SCL10	SCLK11	SCL11		SDI10/ SDI010	SDA10	SDI11/ SDI011	SDA11		TXD3/RXD3			TI11	TO11	PWM5	
PD14							SDI10/ SDI010	SDA10	SDI11/ SDI011	SDA11		TXD1/ RXD1					TXD3/RXD3			TI12	TO12	PWM6	
PD15							SDI10/ SDI010	SDA10	SDI11/ SDI011	SDA11		TXD1/ RXD1					TXD3/RXD3			TI13	TO13	PWM7	BUZ

注1：本产品IO复用功能，需要用户单独配置PMC，PM等寄存器。

注2：当选择到IIC功能时，开漏功能自动开启。

注3：针对复用功能，即可做输入又可做输出功能，选择好PS_XX 后，输入通道自动开启。

表2-2: 模拟功能和特殊功能管脚

引脚	ADC	TK	LCD	LED	ACMP	其他
PA0	AN0	TK0	LCD_SEG33	LED_SEG33		
PA1	AN1	TK1	LCD_SEG34	LED_SEG34		
PA2	AN2	TK2	LCD_SEG35	LED_SEG35		
PA3	AN3	TK3	LCD_SEG36	LED_SEG36		
PA4	AN4	TK4	LCD_SEG37	LED_SEG37		
PA5	AN5	TK5	LCD_SEG38	LED_SEG38		
PA6	AN6	TK6	LCD_SEG39	LED_SEG39		
VSS						
VDD						
PA7	AN7					
PA8	AN8				C1_O	
PA9	AN9				C0_O	
PA10	AN10					
PA11	AN11					
VDD						
VSS						
PB0	AN12	TK12				
PB1	AN13	TK13				RESETB
PB2	AN14	TK14				X1/XT1
PB3	AN15	TK15				X2/XT2
PB4	AN16	TK16	LCD_COM0	LED_COM0	C1P2	
PB5	AN17	TK17	LCD_COM1	LED_COM1	C1P1	
PB6	AN18	TK18	LCD_COM2	LED_COM2	C1N	
PB7	AN19	TK19	LCD_COM3	LED_COM3	C1P0	
PB8	AN20	TK20	LCD_COM4/LCD_SEG0	LED_COM4/LED_SEG0	C0N	
PB9	AN21	TK21	LCD_COM5/LCD_SEG1	LED_COM5/LED_SEG1	C0P0	
PB10	AN22	TK22	LCD_COM6/LCD_SEG2	LED_COM6/LED_SEG2	C0P1	
PB11	AN23	TK23	LCD_COM7/LCD_SEG3	LED_COM7/LED_SEG3	C0P2	
PB12	AN24					
PB13	AN25					
PB14	AN26					
PB15	AN27					
PC0	AN28					
PC1	AN29					
PC2	AN30	TK30	LCD_SEG4	LED_SEG4	C1_O	
PC3	AN31	TK31	LCD_SEG5	LED_SEG5	C0_O	
PC4	AN32	TK32	LCD_SEG6	LED_SEG6		
PC5	AN33	TK33	LCD_SEG7	LED_SEG7		
PC6	AN34	TK34				
PC7	AN35	TK35	LCD_SEG8	LED_SEG8	C1_O	
PC8	AN36	TK36	LCD_SEG9	LED_SEG9		
PC9	AN37	TK37	LCD_SEG10	LED_SEG10		
PC10	AN38	TK38	LCD_SEG11	LED_SEG11		
PC11	AN39	TK39	LCD_SEG12	LED_SEG12		SWDCK
PC12	AN40	TK40	LCD_SEG13	LED_SEG13		SWDIO
PC13	AN41	TK41	LCD_SEG14	LED_SEG14		
PC14	AN42	TK42	LCD_SEG15	LED_SEG15		
PC15	AN43	TK43	LCD_SEG16	LED_SEG16		
PD0	AN44	TK44	LCD_SEG17	LED_SEG17		
PD1	AN45	TK45	LCD_SEG18	LED_SEG18		SWDCLK2
PD2	AN46	TK46	LCD_SEG19	LED_SEG19		SWDIO2
PD3	AN47	TK47	LCD_SEG20	LED_SEG20		
PD4	AN48	TK48	LCD_SEG21	LED_SEG21		
PD5	AN49	TK49	LCD_SEG22			
PD6	AN50	TK50	LCD_SEG23			
PD7	AN51	TK51	LCD_SEG24		C0_O	
PD8	AN52	TK52	LCD_SEG25			
PD9	AN53	TK53	LCD_SEG26			
PD10	AN54	TK54	LCD_SEG27			
PD11	AN55	TK55	LCD_SEG28			
PD12	AN56	TK56	LCD_SEG29			
PD13	AN57	TK57	LCD_SEG30	LED_SEG30		
PD14	AN58	TK58	LCD_SEG31	LED_SEG31		
PD15	AN59	TK59	LCD_SEG32	LED_SEG32		

2.4 寄存器映射

2.4.1 控制功能寄存器映射

(端口控制寄存器的基地址=0x40040000)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PA	0x000	R/W	以位为单位, 设定端口输出锁存器值的寄存器; 输入模式中读此寄存器可以得到引脚电平, 在输出模式中读此寄存器可得到端口的输出锁存器的值。	0x00
PB	0x002	R/W		0x00
PC	0x004	R/W		0x00
PD	0x006	R/W		0x00
PMA	0x020	R/W	在端口作为数字通道使用时, 以位为单位, 设定端口输入或输出的寄存器。	0xFF
PMB	0x022	R/W		0xFF
PMC	0x024	R/W		0xFF
PMD	0x026	R/W		0xFF
PUA	0x030	R/W	端口内部上拉电阻选择寄存器, PB1, PC12, PD2 端口上拉功能自动打开。	0x00
PUB	0x032	R/W		0x02
PUC	0x034	R/W		0x1000
PUD	0x036	R/W		0x04
PDA	0x040	R/W	端口内部下拉电阻选择寄存器 PC11, PD1端口下拉功能自动打开。	0x00
PDB	0x042	R/W		0x00
PDC	0x044	R/W		0x0800
PDD	0x046	R/W		0x02
POMA	0x050	R/W	开漏模式寄存器, 只有在端口配置为输出模式时, N沟道漏极开路才会开启。	0x00
POMB	0x052	R/W		0x00
POMC	0x054	R/W		0x00
POMD	0x056	R/W		0x00
PMCA	0x060	R/W	端口模式寄存器, 以位为单位设定端口作为数字或模拟通道使用; PB1, PB2, PB3, PC11, PC12, PD1, PD2默认作为数字通道使用。	0xFFF
PMCB	0x062	R/W		0xFFF1
PMCC	0x064	R/W		0xE7FF
PMCD	0x066	R/W		0xFFF9
PSETA	0x070	WO	以位为单位来置位端口输出锁存器的寄存器。	0x00
PSETB	0x072	WO		0x00
PSETC	0x074	WO		0x00
PSETD	0x076	WO		0x00
PCLRA	0x080	WO	以位为单位来清零端口输出锁存器的寄存器。	0x00
PCLRB	0x082	WO		0x00
PCLRC	0x084	WO		0x00
PCLRD	0x086	WO		0x00
PREADA	0x090	RO	在端口作为数字GPIO时, 可以通过读此寄存器得到对应端口电平。	0xx
PREADB	0x092	RO		0xx
PREADC	0x094	RO		0xx
PREADD	0x096	RO		0xx
PTTLA	0x0a0	R/W	GPIO的TTL/SIMT输入选择, 默认选择SMIT输入	0x00
PTTLB	0x0a2	R/W		0x00
PTTLC	0x0a4	R/W		0x00
PTTLD	0x0a6	R/W		0x00

2.4.2 输出输入复用功能寄存器映射

(输出输入复用功能寄存器的基地址=0x40040800)

RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值	使用条件	优先级
PS_INTP0	0x00	R/W	外部中断0输入端口选择寄存器	0x7F	PMC=0, PM=1	高
PS_INTP1	0x01	R/W	外部中断1输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP2	0x02	R/W	外部中断2输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP3	0x03	R/W	外部中断3输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP4	0x04	R/W	外部中断4输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP5	0x05	R/W	外部中断5输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP6	0x06	R/W	外部中断6输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP7	0x07	R/W	外部中断7输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP8	0x08	R/W	外部中断8输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP9	0x09	R/W	外部中断9输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP10	0x0A	R/W	外部中断10输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_INTP11	0x0B	R/W	外部中断11输入端口选择寄存器	0x7F	PMC=0, PM=1	
-	-	-	保留	-	-	
PS_SCLA	0x0E	R/W	IICA SCL端口选择寄存器	0x7	PMC=0, PM=0	
PS_SDAA	0x0F	R/W	IICA SDA端口选择寄存器	0x7	PMC=0, PM=0	
PS_SCK00	0x10	R/W	SPI00 SCLK /IIC00 SCL端口选择寄存器	0x3	SPI00主机: PMC=0, PM=0 SPI00从机: PMC=0, PM=1 IIC00主/从机: PMC=0, PM=0	
PS_SCK01	0x11	R/W	SPI01 SCLK /IIC01 SCL端口选择寄存器	0x3	SPI01主机: PMC=0, PM=0 SPI01从机: PMC=0, PM=1 IIC01主/从机: PMC=0, PM=0	
PS_SCK10	0x12	R/W	SPI10 SCLK /IIC10 SCL端口选择寄存器	0x3	SPI10主机: PMC=0, PM=0 SPI10从机: PMC=0, PM=1 IIC10主/从机: PMC=0, PM=0	
PS_SCK11	0x13	R/W	SPI11 SCLK /IIC11 SCL端口选择寄存器	0x3	SPI11主机: PMC=0, PM=0 SPI11从机: PMC=0, PM=1 IIC11主/从机: PMC=0, PM=0	
PS_SCK20	0x14	R/W	SPI20 SCLK /IIC20 SCL端口选择寄存器	0x3	SPI20主机: PMC=0, PM=0 SPI20从机: PMC=0, PM=1 IIC20主/从机: PMC=0, PM=0	
PS_SCK21	0x15	R/W	SPI21 SCLK /IIC21 SCL端口选择寄存器	0x3	SPI21主机: PMC=0, PM=0 SPI21从机: PMC=0, PM=1 IIC21主/从机: PMC=0, PM=0	
-	-	-	保留	-	-	
PS_SDO00	0x20	R/W	SPI00 MOSO/UART0 TXD/IIC00 SDA 端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=0 UART0: PMC=0, PM=0 IIC00主/从机: PMC=0, PM=0	
PS_SDO01	0x21	R/W	SPI00 MOSO /IIC00 SDA 端口选择寄存器	0x7	SPI01主/从机: PMC=0, PM=0 IIC01主/从机: PMC=0, PM=0	
PS_SDO10	0x22	R/W	SPI10 MOSO/UART1 TXD/IIC10 SDA 端口选择寄存器	0x7	SPI10主/从机: PMC=0, PM=0 UART1: PMC=0, PM=0 IIC10主/从机: PMC=0, PM=0	
PS_SDO11	0x23	R/W	SPI11 MOSO /IIC11 SDA 端口选择寄存器	0x7	SPI11主/从机: PMC=0, PM=0 IIC11主/从机: PMC=0, PM=0	
PS_SDO20	0x24	R/W	SPI20 MOSO/UART2 TXD/IIC20 SDA 端口选择寄存器	0x7	SPI20主/从机: PMC=0, PM=0 UART22: PMC=0, PM=0 IIC20主/从机: PMC=0, PM=0	
PS_SDO21	0x25	R/W	SPI21 MOSO /IIC21 SDA 端口选择寄存器	0x7	SPI21主/从机: PMC=0, PM=0 IIC21主/从机: PMC=0, PM=0	
PS_SDO30	0x26	R/W	UART3 TXD端口选择寄存器	0x7F	PMC=0, PM=0	
PS_SDO40	0x28	R/W	UART4TXD端口选择寄存器	0xF	PMC=0, PM=0	

PS_SDO50	0x2A	R/W	UART5 TXD端口选择寄存器	0xF	PMC=0, PM=0	
PS_SDI00	0x30	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI01	0x31	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI10	0x32	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI11	0x33	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI20	0x34	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI21	0x35	R/W	SPI00 MISI/UART0 RXD端口选择寄存器	0x7	SPI00主/从机: PMC=0, PM=1 UART0: PMC=0, PM=1	
PS_SDI30	0x36	R/W	UART3 RXD端口选择寄存器	0x7F	UART3: PMC=0, PM=1	
PS_SDI40	0x38	R/W	UART4 RXD端口选择寄存器	0xF	UART4: PMC=0, PM=1	
PS_SDI50	0x3A	R/W	UART5 RXD端口选择寄存器	0xF	UART5: PMC=0, PM=1	
PS_SS00	0x40	R/W	SPI00 SSI 端口选择寄存器	0x3	主机: PMC=0, PM=0 从机: PMC=0, PM=1	
-	-	-	保留	-	-	
PS_TI00	0x50	R/W	TAU0 TI0输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI01	0x51	R/W	TAU0 TI1输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI02	0x52	R/W	TAU0 TI2输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI03	0x53	R/W	TAU0 TI3输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI10	0x54	R/W	TAU1 TI0输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI11	0x55	R/W	TAU1 TI1输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI12	0x56	R/W	TAU1 TI2输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TI13	0x57	R/W	TAU1 TI3输入端口选择寄存器	0x7F	PMC=0, PM=1	
PS_TO00	0x58	R/W	TAU0 TO0输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO01	0x59	R/W	TAU0 TO1输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO02	0x5A	R/W	TAU0 TO2输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO03	0x5B	R/W	TAU0 TO3输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO10	0x5C	R/W	TAU1 TO0输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO11	0x5D	R/W	TAU1 TO1输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO12	0x5E	R/W	TAU1 TO2输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_TO13	0x5F	R/W	TAU1 TO3输出端口选择寄存器	0x7F	PMC=0, PM=0	
PS_PWM0	0x60	R/W	PWM0输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM1	0x61	R/W	PWM1输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM2	0x62	R/W	PWM2输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM3	0x63	R/W	PWM3输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM4	0x64	R/W	PWM4输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM5	0x65	R/W	PWM5输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM6	0x66	R/W	PWM6输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_PWM7	0x67	R/W	PWM7输出端口选择寄存器	0xF	PMC=0, PM=0	
PS_BUZ0	0x68	R/W	蜂鸣器0输出端口选择	0xF	PMC=0, PM=0	
PS_BUZ1	0x69	R/W	蜂鸣器1输出端口选择	0xF	PMC=0, PM=0	
PS_C00	0x6A	R/W	比较器0输出端口选择	0x3	PMC=0, PM=0	
PS_C10	0x6B	R/W	比较器1输出端口选择	0x3	PMC=0, PM=0	低
PMS	0x7B	R/W	端口读取模式选择寄存器; 在端口为输出模式时, 该寄存器选择读端口锁存器的值或引脚输出电平。详见第27.3.5章	0x00		

2.4.3 特殊功能端口RESINB控制寄存器映射

（寄存器的基地址=0x40020400）

RO：只读，WO：只写，R/W：读写

寄存器	偏移量	读/写	描述	复位值
RSTM	0x0B	R/W	选择RESINB(PB1)口作为外部复位口或GPIO口的寄存器	0x00

2.5 寄存器说明

通过以下寄存器控制端口：

- ◆ 端口寄存器 (Px)
- ◆ 端口模式寄存器 (PMx)
- ◆ 上拉电阻选择寄存器 (PUx)
- ◆ 下拉电阻选择寄存器 (PDx)
- ◆ 端口输出模式寄存器 (POMx)
- ◆ 端口模式控制寄存器 (PMCx)
- ◆ 端口置位控制寄存器 (PSETx)
- ◆ 端口清零控制寄存器 (PCLR_x)
- ◆ 端口输入复用功能配置寄存器 (PS_XX)
- ◆ 端口电平选择寄存器 (PTTLx)
- ◆ 端口回读寄存器寄存器 (PREADx)
- ◆ 特殊功能端口RESINB控制寄存器(RSTM)

2.5.1 端口寄存器（Px）

这是以位为单位设定端口输出锁存器值的寄存器 Px（x=A~D）。在输入模式中读此寄存器可以得到引脚电平，而在输出模式中读可以得到端口的输出锁存器的值。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	Px15	端口x的电平寄存器第15位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
14	Px14	端口x的电平寄存器第14位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
13	Px13	端口x的电平寄存器第13位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
12	Px12	端口x的电平寄存器第12位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
11	Px11	端口x的电平寄存器第11位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
10	Px10	端口x的电平寄存器第10位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
9	Px9	端口x的电平寄存器第9位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
8	Px8	端口x的电平寄存器第8位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
7	Px7	端口x的电平寄存器第7位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
6	Px6	端口x的电平寄存器第6位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
5	Px5	端口x的电平寄存器第5位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
4	Px4	端口x的电平寄存器第4位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
3	Px3	端口x的电平寄存器第3位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
2	Px2	端口x的电平寄存器第2位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0
1	Px1	端口x的电平寄存器第1位 0: 输出模式时输出“0”；输入模式时输入低电平	0

		1: 输出模式时输出“1”；输入模式时输入高电平	
0	Px0	端口x的电平寄存器第0位 0: 输出模式时输出“0”；输入模式时输入低电平 1: 输出模式时输出“1”；输入模式时输入高电平	0

注：必须给未分配的位设定初始值。

2.5.2 端口模式寄存器 (PMx)

当端口作为数字通道使用时，这是以位为单位设定其输入/输出的寄存器PMx (x=A~D)。在产生复位信号后，所有端口默认为输入状态。具体寄存器描述如下：

位	符号	描述	复位值
15	PMx15	端口x的模式寄存器第15位 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))	1
14	PMx14	端口x的模式寄存器第14位 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))	1
13	PMx13	端口x的模式寄存器第13位 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))	1
12	PMx12	端口x的模式寄存器第12位 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))	1
11	PMx11	端口x的模式寄存器第11位 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))	1
10	PMx10	端口x的模式寄存器第10位 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))	1
9	PMx9	端口x的模式寄存器第9位 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))	1
8	PMx8	端口x的模式寄存器第8位 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))	1
7	PMx7	端口x的模式寄存器第7位 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))	1
6	PMx6	端口x的模式寄存器第6位 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))	1
5	PMx5	端口x的模式寄存器第5位 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))	1
4	PMx4	端口x的模式寄存器第4位 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))	1
3	PMx3	端口x的模式寄存器第3位 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))	1
2	PMx2	端口x的模式寄存器第2位 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))	1
1	PMx1	端口x的模式寄存器第1位 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))	1

0	PMx0	端口x的模式寄存器第0位 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）)	1

注：必须给未分配的位设定初始值。

2.5.3 上拉电阻选择寄存器（PUx）

内部上拉电阻的选择寄存器 PUx（x=A~D）。只能在对应的 PMCx 对应位等于 0 的时候，设置上拉电阻有效。

在产生复位信号后，PB1，PC12，PD2 端口上拉功能自动打开，其他端口的上拉功能默认不打开。具体寄存器描述如下：

位	符号	描述	复位值
15	PUx15	Px引脚的内部上拉电阻选择第15位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
14	PUx14	Px引脚的内部上拉电阻选择第14位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
13	PUx13	Px引脚的内部上拉电阻选择第13位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
12	PUx12	Px引脚的内部上拉电阻选择第12位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	当x=C时，复位值为1； 当x=其他时，复位值为0
11	PUx11	Px引脚的内部上拉电阻选择第11位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
10	PUx10	Px引脚的内部上拉电阻选择第10位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
9	PUx9	Px引脚的内部上拉电阻选择第9位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
8	PUx8	Px引脚的内部上拉电阻选择第8位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
7	PUx7	Px引脚的内部上拉电阻选择第7位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
6	PUx6	Px引脚的内部上拉电阻选择第6位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
5	PUx5	Px引脚的内部上拉电阻选择第5位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
4	PUx4	Px引脚的内部上拉电阻选择第4位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
3	PUx3	Px引脚的内部上拉电阻选择第3位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0
2	PUx2	Px引脚的内部上拉电阻选择第2位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	当x=D时，复位值为1； 当x=其他时，复位值为0

1	PUx1	Px引脚的内部上拉电阻选择第1位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	当x=B时，复位值为1； 当x=其他时，复位值为0
0	PUx0	Px引脚的内部上拉电阻选择第0位 0: 不连接内部上拉电阻 1: 连接内部上拉电阻	0

注：必须给未分配的位设定初始值。

2.5.4 下拉电阻选择寄存器（PDx）

内部下拉电阻选择寄存器PDx（x=A~D）。只能在对应的PMCx对应位等于0的时候，设置下拉电阻有效；RESINB(PB1)端口无下拉功能。

在产生复位信号后，PC11，PD1端口下拉功能自动打开，其他端口的下拉功能默认不打开。

具体寄存器描述如下：

位	符号	描述	复位值
15	PDx15	Px引脚的内部下拉电阻选择第15位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
14	PDx14	Px引脚的内部下拉电阻选择第14位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
13	PDx13	Px引脚的内部下拉电阻选择第13位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
12	PDx12	Px引脚的内部下拉电阻选择第12位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
11	PDx11	Px引脚的内部下拉电阻选择第11位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	当x=C时，复位值为1； 当x=其他时，复位值为0
10	PDx10	Px引脚的内部下拉电阻选择第10位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
9	PDx9	Px引脚的内部下拉电阻选择第9位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
8	PDx8	Px引脚的内部下拉电阻选择第8位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
7	PDx7	Px引脚的内部下拉电阻选择第7位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
6	PDx6	Px引脚的内部下拉电阻选择第6位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
5	PDx5	Px引脚的内部下拉电阻选择第5位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
4	PDx4	Px引脚的内部下拉电阻选择第4位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
3	PDx3	Px引脚的内部下拉电阻选择第3位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0
2	PDx2	Px引脚的内部下拉电阻选择第2位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0

1	PDx1	Px引脚的内部下拉电阻选择第1位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	当x=D时，复位值为1； 当x=其他时，复位值为0
0	PDx0	Px引脚的内部下拉电阻选择第0位 0: 不连接内部下拉电阻 1: 连接内部下拉电阻	0

注：必须给未分配的位设定初始值。

2.5.5 端口输出模式寄存器（POMx）

端口输出模式寄存器POMx（x=A~D），只有在配置成输出模式N沟道漏极开路才会开启。端口复用为IIC功能时，会强制开启开漏模式。

在产生复位信号后，这些寄存器的值变为“00H”。

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

具体寄存器描述如下：

位	符号	描述	复位值
15	POMx15	Px引脚输出模式选择第15位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
14	POMx14	Px引脚输出模式选择第14位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
13	POMx13	Px引脚输出模式选择第13位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
12	POMx12	Px引脚输出模式选择第12位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
11	POMx11	Px引脚输出模式选择第11位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
10	POMx10	Px引脚输出模式选择第10位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
9	POMx9	Px引脚输出模式选择第9位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
8	POMx8	Px引脚输出模式选择第8位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
7	POMx07	Px引脚输出模式选择第7位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
6	POMx6	Px引脚输出模式选择第6位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
5	POMx5	Px引脚输出模式选择第5位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
4	POMx4	Px引脚输出模式选择第4位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
3	POMx3	Px引脚输出模式选择第3位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
2	POMx2	Px引脚输出模式选择第2位 0: 通常输出模式	0

		1: N沟道漏极开路输出模式	
1	POMx1	Px引脚输出模式选择第1位 0: 通常输出模式 1: N沟道漏极开路输出模式	0
0	POMx0	Px引脚输出模式选择第0位 0: 通常输出模式 1: N沟道漏极开路输出模式	0

注：必须给未分配的位设定初始值。

2.5.6 端口模式控制寄存器 (PMCx)

端口模式控制寄存器 PMCx(x=A~D)，以位为单位设定端口作为数字（输入/输出）或者作为模拟（输入）通道。

在产生复位信号后，PB1，PB2，PB3，PC11，PC12，PD1，PD2 默认作为数字通道使用（PMCB1，PMCB2，PMCB3，PMCC11，PMCC12，PMCD1，PMCD2 复位值为“0”），其他端口默认作为模拟通道使用即 PMCx 对应位等于 1。具体寄存器描述如下：

位	符号	描述	复位值
15	PMCx15	Px引脚数字（输入/输出）或模拟（输入）第15位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
14	PMCx14	Px引脚数字（输入/输出）或模拟（输入）第14位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
13	PMCx13	Px引脚数字（输入/输出）或模拟（输入）第13位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
12	PMCx12	Px引脚数字（输入/输出）或模拟（输入）第12位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当x=C时，复位值为0； 当x=其他时，复位值为1
11	PMCx11	Px引脚数字（输入/输出）或模拟（输入）第11位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当x=C时，复位值为0； 当x=其他时，复位值为1
10	PMCx10	Px引脚数字（输入/输出）或模拟（输入）第10位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
9	PMCx9	Px引脚数字（输入/输出）或模拟（输入）第9位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
8	PMCx8	Px引脚数字（输入/输出）或模拟（输入）第8位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
7	PMCx7	Px引脚数字（输入/输出）或模拟（输入）第7位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
6	PMCx6	Px引脚数字（输入/输出）或模拟（输入）第6位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
5	PMCx5	Px引脚数字（输入/输出）或模拟（输入）第5位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
4	PMCx4	Px引脚数字（输入/输出）或模拟（输入）第4位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1
3	PMCx3	Px引脚数字（输入/输出）或模拟（输入）第3位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当x=B时，复位值为0； 当x=其他时，复位值为1
2	PMCx2	Px引脚数字（输入/输出）或模拟（输入）第2位	当x=B/D时，复

		0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	位值为0； 当x=其他时， 复位值为1
1	PMCx1	Px引脚数字（输入/输出）或模拟（输入）第1位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	当x=B/D时，复 位值为0； 当x=其他时， 复位值为1
0	PMCx0	Px引脚数字（输入/输出）或模拟（输入）第0位 0: 数字输入/输出（模拟输入以外的复用功能） 1: 模拟输入	1

注1：必须给未分配的位设定初始值。

注2：若使用SWD在线调试，SWDIO/SWDCLK严禁复用成触摸或LCD等模拟功能。

2.5.7 端口置位控制寄存器（PSETx）

这是以位为单位来置位端口输出锁存器的寄存器PSETx（x=A~D）。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PSETx15	Px引脚的置位控制第15位 0: 无操作 1: 对应的Px15置1	0
14	PSETx14	Px引脚的置位控制第14位 0: 无操作 1: 对应的Px14置1	0
13	PSETx13	Px引脚的置位控制第13位 0: 无操作 1: 对应的Px13置1	0
12	PSETx12	Px引脚的置位控制第12位 0: 无操作 1: 对应的Px12置1	0
11	PSETx11	Px引脚的置位控制第11位 0: 无操作 1: 对应的Px11置1	0
10	PSETx10	Px引脚的置位控制第10位 0: 无操作 1: 对应的Px10置1	0
9	PSETx9	Px引脚的置位控制第9位 0: 无操作 1: 对应的Px09置1	0
8	PSETx8	Px引脚的置位控制第8位 0: 无操作 1: 对应的Px08置1	0
7	PSETx7	Px引脚的置位控制第7位 0: 无操作 1: 对应的Px07置1	0
6	PSETx6	Px引脚的置位控制第6位 0: 无操作 1: 对应的Px06置1	0
5	PSETx5	Px引脚的置位控制第5位 0: 无操作 1: 对应的Px05置1	0
4	PSETx4	Px引脚的置位控制第4位 0: 无操作 1: 对应的Px04置1	0
3	PSETx3	Px引脚的置位控制第3位 0: 无操作 1: 对应的Px03置1	0
2	PSETx2	Px引脚的置位控制第2位 0: 无操作 1: 对应的Px02置1	0
1	PSETx1	Px引脚的置位控制第1位 0: 无操作 1: 对应的Px01置1	0

0	PSETx0	Px引脚的置位控制第0位 0: 无操作 1: 对应的Px00置1	0

注：必须给未分配的位设定初始值。

2.5.8 端口清零控制寄存器（PCLR_x）

这是以位为单位来清零端口输出锁存器的寄存器 PCLR_x（x=A~D）。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PCLR _x 15	Px引脚的清零控制第15位 0: 无操作 1: 对应的Px15清零	0
14	PCLR _x 14	Px引脚的清零控制第14位 0: 无操作 1: 对应的Px14清零	0
13	PCLR _x 13	Px引脚的清零控制第13位 0: 无操作 1: 对应的Px13清零	0
12	PCLR _x 12	Px引脚的清零控制第12位 0: 无操作 1: 对应的Px12清零	0
11	PCLR _x 11	Px引脚的清零控制第11位 0: 无操作 1: 对应的Px11清零	0
10	PCLR _x 10	Px引脚的清零控制第10位 0: 无操作 1: 对应的Px10清零	0
9	PCLR _x 9	Px引脚的清零控制第9位 0: 无操作 1: 对应的Px09清零	0
8	PCLR _x 8	Px引脚的清零控制第8位 0: 无操作 1: 对应的Px08清零	0
7	PCLR _x 7	Px引脚的清零控制第7位 0: 无操作 1: 对应的Px07清零	0
6	PCLR _x 6	Px引脚的清零控制第6位 0: 无操作 1: 对应的Px06清零	0
5	PCLR _x 5	Px引脚的清零控制第5位 0: 无操作 1: 对应的Px05清零	0
4	PCLR _x 4	Px引脚的清零控制第4位 0: 无操作 1: 对应的Px04清零	0
3	PCLR _x 3	Px引脚的清零控制第3位 0: 无操作 1: 对应的Px03清零	0
2	PCLR _x 2	Px引脚的清零控制第2位 0: 无操作 1: 对应的Px02清零	0
1	PCLR _x 1	Px引脚的清零控制第1位 0: 无操作 1: 对应的Px01清零	0

0	PCLR _x 0	P _x 引脚的清零控制第0位 0: 无操作 1: 对应的P _x 00清零	0

注：必须给未分配的位设定初始值。

2.5.9 端口回读寄存器(PREADx)

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

位	符号	描述	复位值
15	PREADx15	Px引脚的端口电平第15位 0: 端口为低电平 1: 端口为高电平	x
14	PREADx14	Px引脚的端口电平第14位 0: 端口为低电平 1: 端口为高电平	x
13	PREADx13	Px引脚的端口电平第13位 0: 端口为低电平 1: 端口为高电平	x
12	PREADx12	Px引脚的端口电平第12位 0: 端口为低电平 1: 端口为高电平	x
11	PREADx11	Px引脚的端口电平第11位 0: 端口为低电平 1: 端口为高电平	x
10	PREADx10	Px引脚的端口电平第10位 0: 端口为低电平 1: 端口为高电平	x
9	PREADx9	Px引脚的端口电平第9位 0: 端口为低电平 1: 端口为高电平	x
8	PREADx8	Px引脚的端口电平第8位 0: 端口为低电平 1: 端口为高电平	x
7	PREADx7	Px引脚的端口电平第7位 0: 端口为低电平 1: 端口为高电平	x
6	PREADx6	Px引脚的端口电平第6位 0: 端口为低电平 1: 端口为高电平	x
5	PREADx5	Px引脚的端口电平第5位 0: 端口为低电平 1: 端口为高电平	x
4	PREADx4	Px引脚的端口电平第4位 0: 端口为低电平 1: 端口为高电平	x
3	PREADx3	Px引脚的端口电平第3位 0: 端口为低电平 1: 端口为高电平	x
2	PREADx2	Px引脚的端口电平第2位 0: 端口为低电平 1: 端口为高电平	x
1	PREADx1	Px引脚的端口电平第1位	x

		0: 端口为低电平 1: 端口为高电平	
0	PREADx0	Px引脚的端口电平第0位 0: 端口为低电平 1: 端口为高电平	x

2.5.10 TTL和施密特输入选择(PTTLx)

这是以位为单位来控制引脚电平输入方式的寄存器 PTTLx (x=A~D)。在产生复位信号后，这些寄存器的值变为“00H”。具体寄存器描述如下：

位	符号	描述	复位值
15	PTTLx15	Px引脚输入电平选择控制第15位 0: 施密特输入 1: TTL输入	0
14	PTTLx14	Px引脚输入电平选择控制第14位 0: 施密特输入 1: TTL输入	0
13	PTTLx13	Px引脚输入电平选择控制第13位 0: 施密特输入 1: TTL输入	0
12	PTTLx12	Px引脚输入电平选择控制第12位 0: 施密特输入 1: TTL输入	0
11	PTTLx11	Px引脚输入电平选择控制第11位 0: 施密特输入 1: TTL输入	0
10	PTTLx10	Px引脚输入电平选择控制第10位 0: 施密特输入 1: TTL输入	0
9	PTTLx9	Px引脚输入电平选择控制第9位 0: 施密特输入 1: TTL输入	0
8	PTTLx8	Px引脚输入电平选择控制第8位 0: 施密特输入 1: TTL输入	0
7	PTTLx07	Px引脚输入电平选择控制第7位 0: 施密特输入 1: TTL输入	0
6	PTTLx6	Px引脚输入电平选择控制第6位 0: 施密特输入 1: TTL输入	0
5	PTTLx5	Px引脚输入电平选择控制第5位 0: 施密特输入 1: TTL输入	0
4	PTTLx4	Px引脚输入电平选择控制第4位 0: 施密特输入 1: TTL输入	0
3	PTTLx3	Px引脚输入电平选择控制第3位 0: 施密特输入 1: TTL输入	0
2	PTTLx2	Px引脚输入电平选择控制第2位 0: 施密特输入 1: TTL输入	0
1	PTTLx1	Px引脚输入电平选择控制第1位 0: 施密特输入 1: TTL输入	0

0	PTTLx0	Px引脚输入电平选择控制第0位	0
		0: 施密特输入 1: TTL输入	

注：必须给未分配的位设定初始值。

2.5.11 端口输出复用功能配置寄存器（PS_XX）

端口输出复用功能配置寄存器PS_XX可实现将一部分外围模块的输出功能映射到端口，具体映射功能参见表2-1；端口输出复用功能配置寄存器的复位值为“FFH”，此时端口为默认的兼用功能和GPIO功能。设置步骤如下：

- ◆ 设置PMCmn=0，选择数字功能
- ◆ 根据PS_XX寄存器功能，设置输入或输出功能（参考2.3.2列表信息）
- ◆ 设置PS_XX寄存器选择对应管脚
- ◆ 开启对应模块的功能（如PWM，IICA，SPI00等）

注：

- （1）IICA，IIC00，IIC01，IIC10，IIC11，IIC20，IIC21的功能按照上述步骤设置后会自动开启开漏功能，如果需要接上拉电阻，可以外接上拉或单独设置PUx寄存器
- （2）通过PS_XX寄存器选择GPIO，说明该功能具有唯一性，不能通过多个GPIO同时输出
- （3）不同功能选择PS_XX后，输出具有优先级（PMx=0），优先级的顺序按照2.3.2表格所描述。

举例如下：

当设置PS_TO00=0x00，PS_TXD3=0x00，说明INTP0和UART3都选择了PA00输出，由于TAU0 TO0的优先级高于UART3 TXD的优先级，所以PA00输出的是TO0的功能

- （4）INTP0~INTP11，UART3 TXD、RXD是任一GPIO映射，具体说明见对应的寄存器PS_XX

PS_XX具体寄存器描述如下。

2.5.11.1 INTP0输入端口配置寄存器（PS_INTP0）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP0[6:0]	PS_INTP0=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP0的输入功能 0x01: PA01对应INTP0的输入功能 0x0B: PA11对应INTP0的输入功能 0x10: PB00对应INTP0的输入功能 0x1F: PB15对应INTP0的输入功能 0x2F: PC15对应INTP0的输入功能 0x3F: PD15对应INTP0的输入功能 其他: GPIO功能	0x7F

2.5.11.2 INTP1输入端口配置寄存器 (PS_INTP1)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP1[6:0]	PS_INTP1=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP1的输入功能 0x01: PA01对应INTP1的输入功能 0x0B: PA11对应INTP1的输入功能 0x10: PB00对应INTP1的输入功能 0x1F: PB15对应INTP1的输入功能 0x2F: PC15对应INTP1的输入功能 0x3F: PD15对应INTP1的输入功能 0x3F: PD15对应INTP1的输入功能 其他: GPIO功能	0x7F

2.5.11.3 INTP2输入端口配置寄存器 (PS_INTP2)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP2[6:0]	PS_INTP2=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP2的输入功能 0x01: PA01对应INTP2的输入功能 0x0B: PA11对应INTP2的输入功能 0x10: PB00对应INTP2的输入功能 0x1F: PB15对应INTP2的输入功能 0x2F: PC15对应INTP2的输入功能 0x3F: PD15对应INTP2的输入功能 0x3F: PD15对应INTP2的输入功能 其他: GPIO功能	0x7F

2.5.11.4 INTP3输入端口配置寄存器（PS_INTP3）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP3[6:0]	PS_INTP3=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP3的输入功能 0x01: PA01对应INTP3的输入功能 0x0B: PA11对应INTP3的输入功能 0x10: PB00对应INTP3的输入功能 0x1F: PB15对应INTP3的输入功能 0x2F: PC15对应INTP3的输入功能 0x3F: PD15对应INTP3的输入功能 0x3F: PD15对应INTP3的输入功能 其他: GPIO功能	0x7F

2.5.11.5 INTP4输入端口配置寄存器（PS_INTP4）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP4[6:0]	PS_INTP4=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP4的输入功能 0x01: PA01对应INTP4的输入功能 0x0B: PA11对应INTP4的输入功能 0x10: PB00对应INTP4的输入功能 0x1F: PB15对应INTP4的输入功能 0x2F: PC15对应INTP4的输入功能 0x3F: PD15对应INTP4的输入功能 0x3F: PD15对应INTP4的输入功能 其他: GPIO功能	0x7F

2.5.11.6 INTP5输入端口配置寄存器（PS_INTP5）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP5[6:0]	PS_INTP5=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP5的输入功能 0x01: PA01对应INTP5的输入功能 0x0B: PA11对应INTP5的输入功能 0x10: PB00对应INTP5的输入功能 0x1F: PB15对应INTP5的输入功能 0x2F: PC15对应INTP5的输入功能 0x3F: PD15对应INTP5的输入功能 0x3F: PD15对应INTP5的输入功能 其他: GPIO功能	0x7F

2.5.11.7 INTP6输入端口配置寄存器（PS_INTP6）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP6[6:0]	PS_INTP6=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP6的输入功能 0x01: PA01对应INTP6的输入功能 0x0B: PA11对应INTP6的输入功能 0x10: PB00对应INTP6的输入功能 0x1F: PB15对应INTP6的输入功能 0x2F: PC15对应INTP6的输入功能 0x3F: PD15对应INTP6的输入功能 0x3F: PD15对应INTP6的输入功能 其他: GPIO功能	0x7F

2.5.11.8 INTP7输入端口配置寄存器（PS_INTP7）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP7[6:0]	PS_INTP7=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP7的输入功能 0x01: PA01对应INTP7的输入功能 0x0B: PA11对应INTP7的输入功能 0x10: PB00对应INTP7的输入功能 0x1F: PB15对应INTP7的输入功能 0x2F: PC15对应INTP7的输入功能 0x3F: PD15对应INTP7的输入功能 其他: GPIO功能	0x7F

2.5.11.9 INTP8输入端口配置寄存器（PS_INTP8）

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP8[6:0]	PS_INTP8=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP8的输入功能 0x01: PA01对应INTP8的输入功能 0x0B: PA11对应INTP8的输入功能 0x10: PB00对应INTP8的输入功能 0x1F: PB15对应INTP8的输入功能 0x2F: PC15对应INTP8的输入功能 0x3F: PD15对应INTP8的输入功能 其他: GPIO功能	0x7F

2.5.11.10 INTP9输入端口配置寄存器 (PS_INTP9)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP9[6:0]	PS_INTP9=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP9的输入功能 0x01: PA01对应INTP9的输入功能 0x0B: PA11对应INTP9的输入功能 0x10: PB00对应INTP9的输入功能 0x1F: PB15对应INTP9的输入功能 0x2F: PC15对应INTP9的输入功能 0x3F: PD15对应INTP9的输入功能 其他: GPIO功能	0x7F

2.5.11.11 INTP10输入端口配置寄存器 (PS_INTP10)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP10[6:0]	PS_INTP10=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP10的输入功能 0x01: PA01对应INTP10的输入功能 0x0B: PA11对应INTP10的输入功能 0x10: PB00对应INTP10的输入功能 0x1F: PB15对应INTP10的输入功能 0x2F: PC15对应INTP10的输入功能 0x3F: PD15对应INTP10的输入功能 其他: GPIO功能	0x7F

2.5.11.12 INTP11输入端口配置寄存器 (PS_INTP11)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_INTP11[6:0]	PS_INTP11=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应INTP11的输入功能 0x01: PA01对应INTP11的输入功能 0x0B: PA11对应INTP11的输入功能 0x10: PB00对应INTP11的输入功能 0x1F: PB15对应INTP11的输入功能 0x2F: PC15对应INTP11的输入功能 0x3F: PD15对应INTP11的输入功能 其他: GPIO功能	0x7F

2.5.11.13 IICA SCLA端口配置寄存器 (PS_SCLA)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SCLA[2:0]	Px引脚输出复用功能 0x00: PA1对应IICA SCLA功能 0x01: PB2对应IICA SCLA功能 0x02: PB8对应IICA SCLA功能 0x03: PC9对应IICA SCLA功能 0x04: PD1对应IICA SCLA功能 0x05: PD9对应IICA SCLA功能 其他: GPIO功能	0x7

2.5.11.14 IICA SDAA端口配置寄存器 (PS_SDAA)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDAA[2:0]	Px引脚输出复用功能 0x00: PA2对应IICA SDAA功能 0x01: PB3对应IICA SDAA功能 0x02: PB9对应IICA SDAA功能 0x03: PC10对应IICA SDAA功能 0x04: PD2对应IICA SDAA功能 0x05: PD10对应IICA SDAA功能 其他: GPIO功能	0x7

2.5.11.15 串行通信单元SPI0 SCK/ IIC0 SCL端口配置寄存器 (PS_SCK00)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK00[1:0]	Px引脚输出复用功能 0x00: PA4对应SPI0 SCKO/SCKI, IIC0 SCL功能 0x01: PC3对应SPI0 SCKO/SCKI, IIC0 SCL功能 0x02: PD4对应SPI0 SCKO/SCKI, IIC0 SCL功能 其他: GPIO功能	0x3

2.5.11.16 串行通信单元SPI0 MOSO/ IIC0 SDA/UART0 TXD端口配置寄存 (PS_SDO00)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO00[2:0]	Px引脚输出复用功能 0x00: PA5对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 0x01: PA6对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 0x02: PC4对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 0x03: PC5对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 0x04: PD5对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 0x05: PD6对应SPI0 MOSO, IIC0 SDA, UART0 TXD功能 其他: GPIO功能	0x7

2.5.11.17 串行通信单元SPI0 MISI/ UART0 RXD端口配置寄存器 (PS_SDI00)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI00[2:0]	Px引脚输出复用功能 0x00: PA5对应SPI0 MISI, UART0 RXD功能 0x01: PA6对应SPI0 MISI, UART0 RXD功能 0x02: PC4对应SPI0 MISI, UART0 RXD功能 0x03: PC5对应SPI0 MISI, UART0 RXD功能 0x04: PD5对应SPI0 MISI, UART0 RXD功能 0x05: PD6对应SPI0 MISI, UART0 RXD功能 其他: GPIO功能	0x7

2.5.11.18 串行通信单元SPI0 SS端口配置寄存器 (PS_SS00)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SS00[1:0]	Px引脚输出复用功能 0x00: PA3对应SPI0 SSI功能 0x01: PC2对应SPI0 SSI功能 0x02: PD3对应SPI0 SSI功能 其他: GPIO功能	0x3

2.5.11.19 串行通信单元SPI01 SCK/ IIC01 SCL端口配置寄存器 (PS_SCK01)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK01[1:0]	Px引脚输出复用功能 0x00: PA4对应SPI01 SCKO/SCKI, IIC01 SCL功能 0x01: PC3对应SPI01 SCKO/SCKI, IIC01 SCL功能 0x02: PD4对应SPI01 SCKO/SCKI, IIC01 SCL功能 其他: GPIO功能	0x3

2.5.11.20 串行通信单元SPI01 MOSO/ IIC01 SDA端口配置寄存器 (PS_SDO01)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO01[2:0]	Px引脚输出复用功能 0x00: PA5对应SPI01 MOSO, IIC01 SDA功能 0x01: PA6对应SPI01 MOSO, IIC01 SDA功能 0x02: PC4对应SPI01 MOSO, IIC01 SDA功能 0x03: PC5对应SPI01 MOSO, IIC01 SDA功能 0x04: PD5对应SPI01 MOSO, IIC01 SDA功能 0x05: PD6对应SPI01 MOSO, IIC01 SDA功能 其他: GPIO功能	0x7

2.5.11.21 串行通信单元SPI01 MISI端口配置寄存器 (PS_SDI01)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI01[2:0]	Px引脚输出复用功能 0x00: PA5对应SPI01 MISI 0x01: PA6对应SPI01 MISI 0x02: PC4对应SPI01 MISI 0x03: PC5对应SPI01 MISI 0x04: PD5对应SPI01 MISI 0x05: PD6对应SPI01 MISI 其他: GPIO功能	0x7

2.5.11.22 串行通信单元SPI10 SCK/ IIC10 SCL端口配置寄存器 (PS_SCK10)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK10[1:0]	Px引脚输出复用功能 0x00: PC12对应SPI10 SCKO/SCKI, IIC10 SCL功能 0x01: PD12对应SPI10 SCKO/SCKI, IIC10 SCL功能 其他: GPIO功能	0x3

2.5.11.23 串行通信单元SPI10 MOSO/ IIC10 SDA/UART1 TXD端口配置寄存器(PS_SDO10)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO10[2:0]	Px引脚输出复用功能 0x00: PC13对应SPI10 MOSO, IIC10 SDA, UART1 TXD功能 0x01: PC14对应SPI10 MOSO, IIC10 SDA, UART1 TXD功能 0x02: PD13对应SPI10 MOSO, IIC10 SDA, UART1 TXD功能 0x03: PD14对应SPI10 MOSO, IIC10 SDA, UART1 TXD功能 其他: GPIO功能	0x7

2.5.11.24 串行通信单元SPI10 MISI/ UART1 RXD端口配置寄存器 (PS_SDI10)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI10[2:0]	Px引脚输出复用功能 0x00: PC13对应SPI10 MISI, UART1 RXD功能 0x01: PC14对应SPI10 MISI, UART1 RXD功能 0x02: PD13对应SPI10 MISI, UART1 RXD功能 0x03: PD14对应SPI10 MISI, UART1 RXD功能 其他: GPIO功能	0x7

2.5.11.25 串行通信单元SPI11 SCK/ IIC11 SCL端口配置寄存器 (PS_SCK11)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK11[1:0]	Px引脚输出复用功能 0x00: PC12对应SPI11 SCKO/SCKI, IIC11 SCL功能 0x01: PD12对应SPI11 SCKO/SCKI, IIC11 SCL功能 其他: GPIO功能	0x7

2.5.11.26 串行通信单元SPI11 MOSO/ IIC11 SDA端口配置寄存器 (PS_SDO11)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO11[2:0]	Px引脚输出复用功能 0x00: PC13对应SPI11 MOSO, IIC11 SDA功能 0x01: PC14对应SPI11 MOSO, IIC11 SDA功能 0x02: PD13对应SPI11 MOSO, IIC11 SDA功能 0x03: PD14对应SPI11 MOSO, IIC11 SDA功能 其他: GPIO功能	0x7

2.5.11.27 串行通信单元SPI11 MISI端口配置寄存器 (PS_SDI11)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI11[2:0]	Px引脚输出复用功能 0x00: PC13对应SPI11 MISI功能 0x01: PC14对应SPI11 MISI功能 0x02: PD13对应SPI11 MISI功能 0x03: PD14对应SPI11 MISI功能 其他: GPIO功能	0x7

2.5.11.28 串行通信单元SPI20 SCK/ IIC20 SCL端口配置寄存器 (PS_SCK20)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK20[1:0]	Px引脚输出复用功能 0x00: PA9对应SPI20 SCKO/SCKI, IIC20 SCL功能 0x01: PB3对应SPI20 SCKO/SCKI, IIC20 SCL功能 其他: GPIO功能	0x3

2.5.11.29 串行通信单元SPI20 MOSO/ IIC20 SDA/UART2 TXD端口配置寄存 (PS_SDO20)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO20[2:0]	Px引脚输出复用功能 0x00: PA10对应SPI20 MOSO, IIC20 SDA, UART2 TXD功能 0x01: PA11对应SPI20 MOSO, IIC20 SDA, UART2 TXD功能 0x02: PB4 对应SPI20 MOSO, IIC20 SDA, UART2 TXD功能 0x03: PB5 对应SPI20 MOSO, IIC20 SDA, UART2 TXD功能 其他: GPIO功能	0x7

2.5.11.30 串行通信单元SPI20 MISI/ UART2 RXD端口配置寄存器 (PS_SDI20)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI20[2:0]	Px引脚输出复用功能 0x00: PA10对应SPI20 MISI, UART2 RXD功能 0x01: PA11对应SPI20 MISI, UART2 RXD功能 0x02: PB4 对应SPI20 MISI, UART2 RXD功能 0x03: PB5 对应SPI20 MISI, UART2 RXD功能 其他: GPIO功能	0x7

2.5.11.31 串行通信单元SPI21 SCK/ IIC21 SCL端口配置寄存器 (PS_SCK21)

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_SCK21[1:0]	Px引脚输出复用功能 0x00: PA9对应SPI21 SCKO/SCKI, IIC21 SCL功能 0x01: PB3对应SPI21 SCKO/SCKI, IIC21 SCL功能 其他: GPIO功能	0x3

2.5.11.32 串行通信单元SPI21 MOSO/ IIC21 SDA端口配置寄存器 (PS_SDO21)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDO21[2:0]	Px引脚输出复用功能 0x00: PA10对应SPI21 MOSO, IIC21 SDA功能 0x01: PA11对应SPI21 MOSO, IIC21 SDA功能 0x02: PB4 对应SPI21 MOSO, IIC21 SDA功能 0x03: PB5 对应SPI21 MOSO, IIC21 SDA功能 其他: GPIO功能	0x7

2.5.11.33 串行通信单元SPI21 MISI端口配置寄存器 (PS_SDI21)

位	符号	描述	复位值
7:3	--	保留	--
2:0	PS_SDI21[2:0]	Px引脚输出复用功能 0x00: PA10对应SPI21 MISI功能 0x01: PA11对应SPI21 MISI功能 0x02: PB4 对应SPI21 MISI功能 0x03: PB5 对应SPI21 MISI功能 其他: GPIO功能	0x7

2.5.11.34 串行通信单元UART3 TXD端口配置寄存器 (PS_SDO30)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_SDO30[6:0]	PS_SDO30=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应 UART3 TXD功能 0x01: PA01对应 UART3 TXD功能 0x0B: PA11对应 UART3 TXD功能 0x10: PB00对应 UART3 TXD功能 0x1F: PB15对应 UART3 TXD功能 0x2F: PC15对应 UART3 TXD功能 0x3F: PD15对应 UART3 TXD功能 其他: GPIO功能	0x7F

2.5.11.35 串行通信单元UART3 RXD端口配置寄存器 (PS_SDI30)

位	符号	描述	复位值
7	--	保留	--
6:0	PS_SDI30[6:0]	PS_SDI30=0xmn (m=0~3对应管脚A~D,n=0~F对应管脚序号0~15) 0x00: PA00对应 UART3 RXD功能 0x01: PA01对应 UART3 RXD功能 0x0B: PA11对应 UART3 RXD功能 0x10: PB00对应 UART3 RXD功能 0x1F: PB15对应 UART3 RXD功能 0x2F: PC15对应 UART3 RXD功能 0x3F: PD15对应 UART3 RXD功能 其他: GPIO功能	0x7F

2.5.11.36 串行通信单元UART4 TXD端口配置寄存器（PS_SDO40）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_SDO40[2:0]	Px引脚输出复用功能 0x00: PB8对应 UART4 TXD功能 0x01: PB9对应 UART4 TXD功能 0x02: PC0对应 UART4 TXD功能 0x03: PC1对应 UART4 TXD功能 0x04: PD1对应 UART4 TXD功能 0x05: PD2对应 UART4TXD功能 其他: GPIO功能	0xF

2.5.11.37 串行通信单元UART4 RXD端口配置寄存器（PS_SDI40）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_SDI40[6:0]	Px引脚输出复用功能 0x00: PB8对应 UART4 RXD功能 0x01: PB9对应 UART4 RXD功能 0x02: PC0对应 UART4RXD功能 0x03: PC1对应 UART4 RXD功能 0x04: PD1对应 UART4 RXD功能 0x05: PD2对应 UART4 RXD功能 其他: GPIO功能	0xF

2.5.11.38 串行通信单元UART5 TXD端口配置寄存器（PS_SDO50）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_SDO50[3:0]	Px引脚输出复用功能 0x00: PB10对应 UART5 TXD功能 0x01: PB11对应 UART5 TXD功能 0x02: PC09对应 UART5 TXD功能 0x03: PC10对应 UART5 TXD功能 0x04: PD03对应 UART5 TXD功能 0x05: PD04对应 UART5 TXD功能 0x06: PD09对应 UART5 TXD功能 0x07: PD10对应 UART5 TXD功能 其他: GPIO功能	0xF

2.5.11.39 串行通信单元UART5 RXD端口配置寄存器（PS_SDI50）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_SDI50[3:0]	Px引脚输出复用功能 0x00: PB10对应 UART5 RXD功能 0x01: PB11对应 UART5 RXD功能 0x02: PC09对应 UART5 RXD功能 0x03: PC10对应 UART5 RXD功能 0x04: PD03对应 UART5 RXD功能 0x05: PD04对应 UART5 RXD功能 0x06: PD09对应 UART5 RXD功能 0x07: PD10对应 UART5 RXD功能 其他: GPIO功能	0xF

2.5.11.40 TAU0 TI0输入端口配置寄存器（PS_TI00）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI00[3:0]	Px引脚输出复用功能 0x00: PA0对应TAU0 TI0的输入功能 0x01: PA8对应TAU0 TI0的输入功能 0x02: PB0对应TAU0 TI0的输入功能 0x03: PB8对应TAU0 TI0的输入功能 0x04: PC0对应TAU0 TI0的输入功能 0x05: PC8对应TAU0 TI0的输入功能 0x06: PD0对应TAU0 TI0的输入功能 0x07: PD8对应TAU0 TI0的输入功能 其他: GPIO功能	0xF

2.5.11.41 TAU0 TI1输入端口配置寄存器（PS_TI01）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI01[3:0]	Px引脚输出复用功能 0x00: PA1对应TAU0 TI1的输入功能 0x01: PA9对应TAU0 TI1的输入功能 0x02: PB1对应TAU0 TI1的输入功能 0x03: PB9对应TAU0 TI1的输入功能 0x04: PC1对应TAU0 TI1的输入功能 0x05: PC9对应TAU0 TI1的输入功能 0x06: PD1对应TAU0 TI1的输入功能 0x07: PD9对应TAU0 TI1的输入功能 其他: GPIO功能	0xF

2.5.11.42 TAU0 TI2输入端口配置寄存器（PS_TI02）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI02[3:0]	Px引脚输出复用功能 0x00: PA2 对应TAU0 TI2的输入功能 0x01: PA10对应TAU0 TI2的输入功能 0x02: PB2 对应TAU0 TI2的输入功能 0x03: PB10对应TAU0 TI2的输入功能 0x04: PC2 对应TAU0 TI2的输入功能 0x05: PC10对应TAU0 TI2的输入功能 0x06: PD2 对应TAU0 TI2的输入功能 0x07: PD10对应TAU0 TI2的输入功能 其他: GPIO功能	0xF

2.5.11.43 TAU0 TI3输入端口配置寄存器（PS_TI03）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI03[3:0]	Px引脚输出复用功能 0x00: PA3 对应TAU0 TI3的输入功能 0x01: PA11对应TAU0 TI3的输入功能 0x02: PB3 对应TAU0 TI3的输入功能 0x03: PB11对应TAU0 TI3的输入功能 0x04: PC3 对应TAU0 TI3的输入功能 0x05: PC11对应TAU0 TI3的输入功能 0x06: PD3 对应TAU0 TI3的输入功能 0x07: PD11对应TAU0 TI3的输入功能 其他: GPIO功能	0xF

2.5.11.44 TAU1 TI0输入端口配置寄存器（PS_TI10）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI10[3:0]	Px引脚输出复用功能 0x00: PA4 对应TAU1 TI0的输入功能 0x01: PB4 对应TAU1 TI0的输入功能 0x02: PB12对应TAU1 TI0的输入功能 0x03: PC4 对应TAU1 TI0的输入功能 0x04: PC12对应TAU1 TI0的输入功能 0x05: PD4 对应TAU1 TI0的输入功能 0x06: PD12对应TAU1 TI0的输入功能 其他: GPIO功能	0xF

2.5.11.45 TAU1 TI1输入端口配置寄存器 (PS_TI11)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI11[3:0]	Px引脚输出复用功能 0x00: PA5 对应TAU1 TI1的输入功能 0x01: PB5 对应TAU1 TI1的输入功能 0x02: PB13对应TAU1 TI1的输入功能 0x03: PC5 对应TAU1 TI1的输入功能 0x04: PC13对应TAU1 TI1的输入功能 0x05: PD5 对应TAU1 TI1的输入功能 0x06: PD13对应TAU1 TI1的输入功能 其他: GPIO功能	0xF

2.5.11.46 TAU1 TI2输入端口配置寄存器 (PS_TI12)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI12[3:0]	Px引脚输出复用功能 0x00: PA6 对应TAU1 TI2的输入功能 0x01: PB6 对应TAU1 TI2的输入功能 0x02: PB14对应TAU1 TI2的输入功能 0x03: PC6 对应TAU1 TI2的输入功能 0x04: PC14对应TAU1 TI2的输入功能 0x05: PD6 对应TAU1 TI2的输入功能 0x06: PD14对应TAU1 TI2的输入功能 其他: GPIO功能	0xF

2.5.11.47 TAU1 TI3输入端口配置寄存器 (PS_TI13)

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TI13[3:0]	Px引脚输出复用功能 0x00: PA7 对应TAU1 TI3的输入功能 0x01: PB7 对应TAU1 TI3的输入功能 0x02: PB15对应TAU1 TI3的输入功能 0x03: PC7 对应TAU1 TI3的输入功能 0x04: PC15对应TAU1 TI3的输入功能 0x05: PD7 对应TAU1 TI3的输入功能 0x06: PD15对应TAU1 TI3的输入功能 其他: GPIO功能	0xF

2.5.11.48 TAU0 TO0输入端口配置寄存器（PS_TO00）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO00[3:0]	Px引脚输出复用功能 0x00: PA0 对应TAU0 TO0的输出功能 0x01: PA8 对应TAU0 TO0的输出功能 0x02: PB0 对应TAU0 TO0的输出功能 0x03: PB8 对应TAU0 TO0的输出功能 0x04: PC0 对应TAU0 TO0的输出功能 0x05: PC8 对应TAU0 TO0的输出功能 0x06: PD0 对应TAU0 TO0的输出功能 0x07: PD8 对应TAU0 TO0的输出功能 其他: GPIO功能	0xF

2.5.11.49 TAU0 TO1输入端口配置寄存器（PS_TO01）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO01[3:0]	Px引脚输出复用功能 0x00: PA1 对应TAU0 TO1的输出功能 0x01: PA9 对应TAU0 TO1的输出功能 0x02: PB1 对应TAU0 TO1的输出功能 0x03: PB9 对应TAU0 TO1的输出功能 0x04: PC1 对应TAU0 TO1的输出功能 0x05: PC9 对应TAU0 TO1的输出功能 0x06: PD1 对应TAU0 TO1的输出功能 0x07: PD9 对应TAU0 TO1的输出功能 其他: GPIO功能	0xF

2.5.11.50 TAU0 TO2输入端口配置寄存器（PS_TO02）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO02[3:0]	Px引脚输出复用功能 0x00: PA2 对应TAU0 TO2的输出功能 0x01: PA10对应TAU0 TO2的输出功能 0x02: PB2 对应TAU0 TO2的输出功能 0x03: PB10对应TAU0 TO2的输出功能 0x04: PC2 对应TAU0 TO2的输出功能 0x05: PC10对应TAU0 TO2的输出功能 0x06: PD2 对应TAU0 TO2的输出功能 0x07: PD10对应TAU0 TO2的输出功能 其他: GPIO功能	0xF

2.5.11.51 TAU0 TO3输入端口配置寄存器（PS_TO03）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO03[3:0]	Px引脚输出复用功能 0x00: PA3 对应TAU0 TO3的输出功能 0x01: PA11对应TAU0 TO3的输出功能 0x02: PB3 对应TAU0 TO3的输出功能 0x03: PB11对应TAU0 TO3的输出功能 0x04: PC3 对应TAU0 TO3的输出功能 0x05: PC11对应TAU0 TO3的输出功能 0x06: PD3 对应TAU0 TO3的输出功能 0x07: PD11对应TAU0 TO3的输出功能 其他: GPIO功能	0xF

2.5.11.52 TAU1 TO0输入端口配置寄存器（PS_TO10）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO10[3:0]	Px引脚输出复用功能 0x00: PA4 对应TAU1 TO0的输出功能 0x01: PB4 对应TAU1 TO0的输出功能 0x02: PB12对应TAU1 TO0的输出功能 0x03: PC4 对应TAU1 TO0的输出功能 0x04: PC12对应TAU1 TO0的输出功能 0x05: PD4 对应TAU1 TO0的输出功能 0x06: PD12对应TAU1 TO0的输出功能 其他: GPIO功能	0xF

2.5.11.53 TAU1 TO1输入端口配置寄存器（PS_TO11）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO11[3:0]	Px引脚输出复用功能 0x00: PA5 对应TAU1 TO1的输出功能 0x01: PB5 对应TAU0 TO1的输出功能 0x02: PB13对应TAU1 TO1的输出功能 0x03: PC5 对应TAU1 TO1的输出功能 0x04: PC13对应TAU1 TO1的输出功能 0x05: PD5 对应TAU1 TO1的输出功能 0x06: PD13对应TAU1 TO1的输出功能 其他: GPIO功能	0xF

2.5.11.54 TAU1 TO2输入端口配置寄存器（PS_TO12）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO12[3:0]	Px引脚输出复用功能 0x00: PA6 对应TAU1 TO2的输出功能 0x01: PB6 对应TAU0 TO2的输出功能 0x02: PB14对应TAU1 TO2的输出功能 0x03: PC6 对应TAU1 TO2的输出功能 0x04: PC14对应TAU1 TO2的输出功能 0x05: PD6 对应TAU1 TO2的输出功能 0x06: PD14对应TAU1 TO2的输出功能 其他: GPIO功能	0xF

2.5.11.55 TAU1 TO3输入端口配置寄存器（PS_TO13）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_TO13[3:0]	Px引脚输出复用功能 0x00: PA7 对应TAU1 TO3的输出功能 0x01: PB7 对应TAU0 TO3的输出功能 0x02: PB15对应TAU1 TO3的输出功能 0x03: PC7 对应TAU1 TO3的输出功能 0x04: PC15对应TAU1 TO3的输出功能 0x05: PD7 对应TAU1 TO3的输出功能 0x06: PD15对应TAU1 TO3的输出功能 其他: GPIO功能	0xF

2.5.11.56 PWM0端口配置寄存器（PS_PWM0）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM0[3:0]	Px引脚输出复用功能 0x00: PA0 对应PWM0的输出功能 0x01: PA8 对应PWM0的输出功能 0x02: PB0 对应PWM0的输出功能 0x03: PB8 对应PWM0的输出功能 0x04: PC0 对应PWM0的输出功能 0x05: PC8 对应PWM0的输出功能 0x06: PD0 对应PWM0的输出功能 0x07: PD8 对应PWM0的输出功能 其他: GPIO功能	0xF

2.5.11.57 PWM1 端口配置寄存器（PS_PWM1）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM1[3:0]	Px引脚输出复用功能 0x00: PA1 对应PWM1的输出功能 0x01: PA9 对应PWM1的输出功能 0x02: PB1 对应PWM1的输出功能 0x03: PB9 对应PWM1的输出功能 0x04: PC1 对应PWM1的输出功能 0x05: PC9 对应PWM1的输出功能 0x06: PD1 对应PWM1的输出功能 0x07: PD9 对应PWM1的输出功能 其他: GPIO功能	0xF

2.5.11.58 PWM2 端口配置寄存器（PS_PWM2）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM2[3:0]	Px引脚输出复用功能 0x00: PA2 对应PWM2的输出功能 0x01: PA10对应PWM2的输出功能 0x02: PB2 对应PWM2的输出功能 0x03: PB10对应PWM2的输出功能 0x04: PC2 对应PWM2的输出功能 0x05: PC10对应PWM2的输出功能 0x06: PD2 对应PWM2的输出功能 0x07: PD10对应PWM2的输出功能 其他: GPIO功能	0xF

2.5.11.59 PWM3 端口配置寄存器（PS_PWM3）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM3[3:0]	Px引脚输出复用功能 0x00: PA3 对应PWM3的输出功能 0x01: PA11对应PWM3的输出功能 0x02: PB3 对应PWM3的输出功能 0x03: PB11对应PWM3的输出功能 0x04: PC3 对应PWM3的输出功能 0x05: PC11对应PWM3的输出功能 0x06: PD3 对应PWM3的输出功能 0x07: PD11对应PWM3的输出功能 其他: GPIO功能	0xF

2.5.11.60 PWM4 端口配置寄存器（PS_PWM4）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM4[3:0]	Px引脚输出复用功能 0x00: PA4 对应PWM4的输出功能 0x01: PB4 对应PWM4的输出功能 0x02: PB12对应PWM4的输出功能 0x03: PC4 对应PWM4的输出功能 0x04: PC12对应PWM4的输出功能 0x05: PD4 对应PWM4的输出功能 0x06: PD12对应PWM4的输出功能 其他: GPIO功能	0xF

2.5.11.61 PWM5 端口配置寄存器（PS_PWM5）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM5[3:0]	Px引脚输出复用功能 0x00: PA5 对应PWM5的输出功能 0x01: PB5 对应PWM5的输出功能 0x02: PB13对应PWM5的输出功能 0x03: PC5 对应PWM5的输出功能 0x04: PC13对应PWM5的输出功能 0x05: PD5 对应PWM5的输出功能 0x06: PD13对应PWM5的输出功能 其他: GPIO功能	0xF

2.5.11.62 PWM6 端口配置寄存器（PS_PWM6）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM6[3:0]	Px引脚输出复用功能 0x00: PA6 对应PWM6的输出功能 0x01: PB6 对应PWM6的输出功能 0x02: PB14对应PWM6的输出功能 0x03: PC6 对应PWM6的输出功能 0x04: PC14对应PWM6的输出功能 0x05: PD6 对应PWM6的输出功能 0x06: PD14对应PWM6的输出功能 其他: GPIO功能	0xF

2.5.11.63 PWM7 端口配置寄存器（PS_PWM7）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_PWM7[3:0]	Px引脚输出复用功能 0x00: PA7 对应PWM7的输出功能 0x01: PB7 对应PWM7的输出功能 0x02: PB15对应PWM7的输出功能 0x03: PC7 对应PWM7的输出功能 0x04: PC15对应PWM7的输出功能 0x05: PD7 对应PWM7的输出功能 0x06: PD15对应PWM7的输出功能 其他: GPIO功能	0xF

2.5.11.64 蜂鸣器BUZ0端口配置寄存器（PS_BUZ0）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_BUZ0[3:0]	Px引脚输出复用功能 0x00: PA0 对应BUZ0 功能 0x01: PA3 对应BUZ0 功能 0x02: PB0 对应BUZ0 功能 0x03: PB10对应BUZ0 功能 0x04: PC0 对应BUZ0 功能 0x05: PC10对应BUZ0 功能 0x06: PD0 对应BUZ0 功能 0x07: PD3 对应BUZ0 功能 0x08: PD11 对应BUZ0 功能 0x09: PD10对应BUZ0 功能 0x0A: PD15 对应BUZ0 功能 其他: GPIO功能	0xF

2.5.11.65 蜂鸣器BUZ1端口配置寄存器（PS_BUZ1）

位	符号	描述	复位值
7:4	--	保留	--
3:0	PS_BUZ1[3:0]	Px引脚输出复用功能 0x00: PA1 对应BUZ1功能 0x01: PA4 对应BUZ1 功能 0x02: PB1 对应BUZ1 功能 0x03: PB11对应BUZ1功能 0x04: PC1 对应BUZ1功能 0x05: PC11对应BUZ1功能 0x06: PD1 对应BUZ1功能 0x07: PD4 对应BUZ1 功能 0x08: PD12 对应BUZ1 功能 其他: GPIO功能	0xF

2.5.11.66 比较器0端口配置寄存器（PS_C0O）

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_C0O[3:0]	Px引脚输出复用功能 0x00: PA9 对应ACMP0O 功能 0x01: PC3 对应ACMP0O 功能 0x02: PD7 对应ACMP0O 功能 其他: GPIO功能	0x3

2.5.11.67 比较器1端口配置寄存器（PS_C1O）

位	符号	描述	复位值
7:2	--	保留	--
1:0	PS_C1O[3:0]	Px引脚输出复用功能 0x00: PA8 对应ACMP0O 功能 0x01: PC2 对应ACMP0O 功能 0x02: PC7 对应ACMP0O 功能 其他: GPIO功能	0x3

2.5.12 特殊功能端口RESINB说明(RSTM)

本产品上电默认 RESINB（PB1）有效，如果需要把该口做 GPIO 使用，那需要通过寄存器关掉复位功能，寄存器描述如下：

位	符号	描述	复位值
7:1	--	保留为0	0
0	RSTM	RESINB引脚的外部复位功能屏蔽 0: RESINB外部复位引脚 1: RESINB做GPIO引脚	0

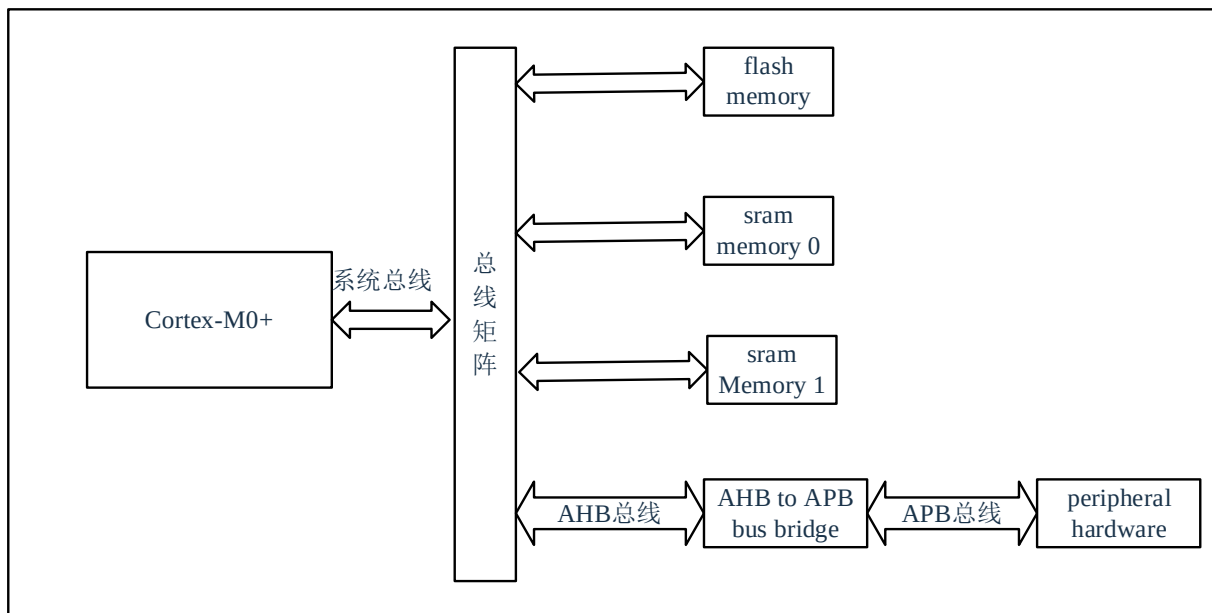
第3章 系统结构

3.1 概述

本产品系统由以下部分组成：

- 1个AHB总线Master：
 - Cortex-M0+
- 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：总线矩阵协调内核系统总线其他总线的访问。
- 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_9FFF	保留	
0x4001_A000 - 0x4001_A013	硬件除法器	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1000 - 0x4002_101F	IWDG	
0x4002_1020 - 0x4002_10FF	WWDG	
0x4002_1100 - 0x4002_17FF	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见第 27 章 安全功能
0x4002_1C00 - 0x4002_1FFF	时钟控制	
0x4002_2200 - 0x4002_2208	发振停止检测	
0x4004_0000 - 0x4004_0F9F	GPIO	
0x4004_0FA0 - 0x4004_10FF	时钟输出/蜂鸣器输出	
0x4004_1100 - 0x4004_19FF	串行通信单元 0/1/2/3	
0x4004_1A30 - 0x4004_1D7F	IICA	
0x4004_1D80 - 0x4004_1F67	定时器阵列 0	
0x4004_2180 - 0x4004_2367	定时器阵列 1	
0x4004_2200 - 0x4004_31FF	保留	
0x4004_3200 - 0x4004_33FF	通用 CRC	详见第 27 章 安全功能
0x4004_3400 - 0x4004_37FF	联动控制器	
0x4004_3800 - 0x4004_43FF	保留	
0x4004_4400 - 0x4004_47FF	EPWM	
0x4004_4800 - 0x4004_4EFF	保留	
0x4004_4F00 - 0x4004_4FFF	实时时钟	
0x4004_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_5AFF	保留	
0x4004_5B00 - 0x4004_5BFF	外部中断控制	
0x4004_5C00 - 0x4006_0FFF	保留	
0x4006_8200 - 0x4006_823F	ACMP	
0x4006_8380 - 0x4006_83BF	TOUCH	
0x4006_83C0 - 0x4006_83FF	保留	
0x4006_8400 - 0x4006_84FF	LCD	
0x4006_8500 - 0x4006_8FFF	LED	
0x4006_9000 - 0xDFFF_FFFF	保留	-

第4章 时钟发生电路

4.1 时钟发生电路的功能

时钟发生电路是产生给CPU和外围硬件提供时钟的电路。有以下2种系统时钟和时钟振荡电路。

(1) 主系统时钟

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（CSC寄存器的bit7）使振荡停止。

② 高速内部振器（高速OCO）

能通过选项字节（000C2H）从 $f_{\text{HOCO}}=64\text{MHz}$ 中选择频率进行振荡， f_{IH} 为 f_{HOCO} 的分频。在解除复位后，CPU一定以此高速内部振荡器时钟开始运行。能通过进入深度睡眠模式或者设置HIOSTOP位（CSC寄存器的bit0）使振荡停止。能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改选项字节设置的频率。有关频率设置，请参照高速内部振荡器的频率选择寄存器（HOCODIV）。

③ PLL锁相环电路

能通过外部晶振输入4M/8M/16MHz频率，经过PLL倍频之后给系统提供最高64MHz时钟。通过PLL控制器PLLCR来控制PLL的振荡和停止。

另外，能由EXCLK/X2/PB3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）和高速内部振荡器时钟/PLL的切换。

(2) 副系统时钟

① 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 $f_{\text{XT}}=32.768\text{kHz}$ 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/PB3引脚提供外部副系统时钟（ $f_{\text{EXS}}=32.768\text{kHz}$ ），并且能通过设置XTSTOP位将外部副系统时钟的输入置为无效。

② 低速内部振荡器（低速OCO）

能使 $f_{\text{L}}=32\text{kHz}(\text{TYP.})$ 的时钟振荡。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 独立看门狗定时器
- 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： 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} : 高速内部振荡器分频后的时钟频率

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

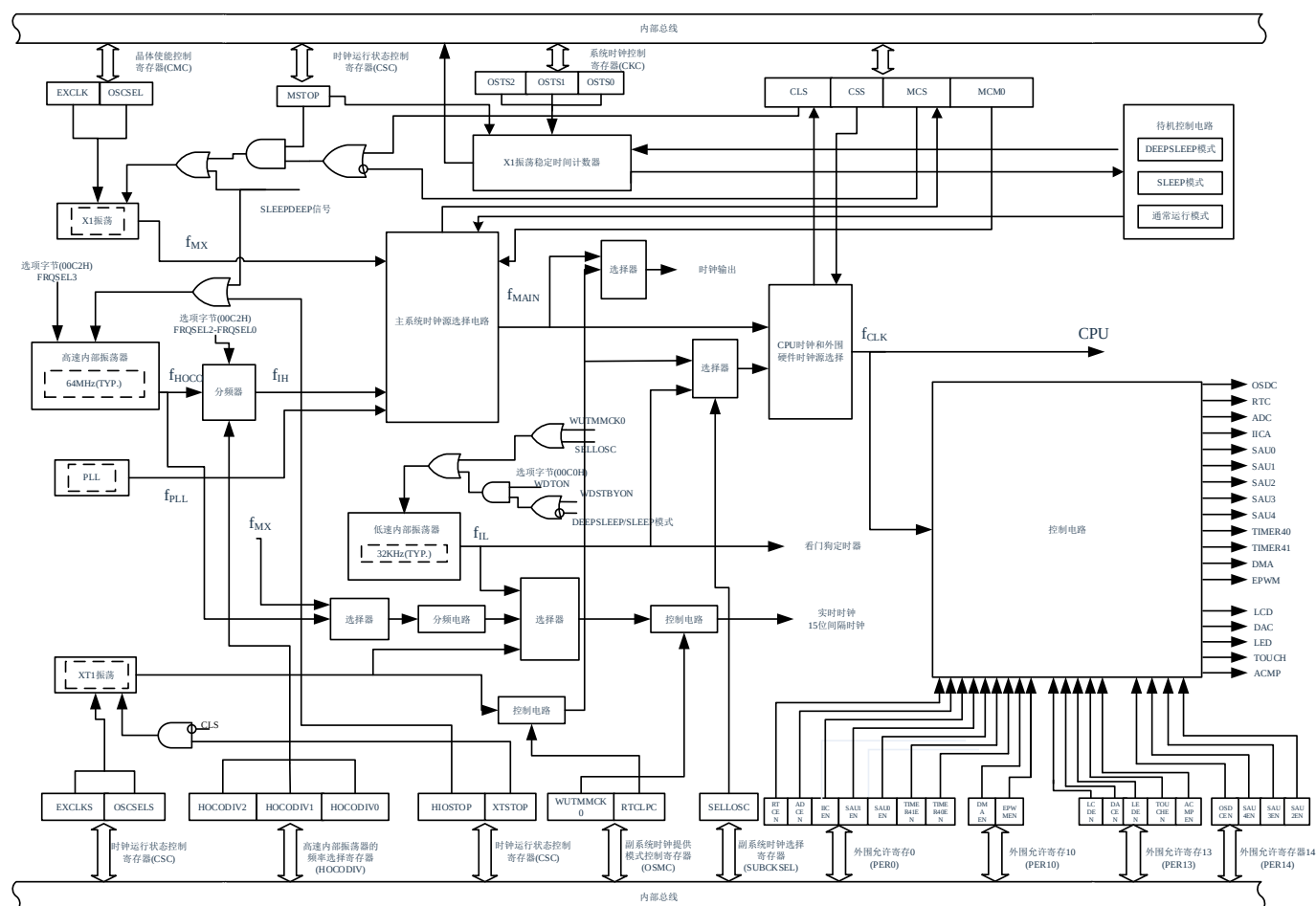
f_{IL} : 低速内部振荡器的时钟频率

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（PER0、PER10、PER13、PER14） 高速内部振荡器的频率选择寄存器（HOCODIV） 低速内部振荡器时钟寄存器（SUBCKSEL）
振荡电路	高速内部振荡器 低速内部振荡器 X1振荡电路 XT1振荡电路

图4-1 时钟发生电路的框图



f_{IH} : 高速内部振荡器分频后的时钟频率

f_{MAIN} : 主系统时钟频率

f_{CLK} : CPU/外围硬件的时钟频率

f_{II} : 低速内部振荡器的时钟频率

f_{MX} ：高速外部晶振时钟振荡频率

f_{SUB} ：低速外部振荡器的时钟频率

f_{PLL} ：PLL时钟频率

4.3 寄存器映射

(以下寄存器基地址 = 0x4002_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CMC	0x000	R/W	时钟运行模式控制寄存器	0x00
CSC	0x001	R/W	时钟运行状态控制寄存器	0xC0
OSTC	0x002	RO	振荡稳定时间计数器的状态寄存器	0x00
OSTS	0x003	R/W	振荡稳定时间选择寄存器	0x07
CKC	0x004	R/W	系统时钟控制寄存器	0x00
SUBCKSEL	0x007	R/W	低速内部振荡器时钟选择寄存器	0x00
PER0	0x020	R/W	外围允许寄存器0	0x00
OSMC	0x023	R/W	15位间隔定时器运行时钟选择寄存器	0x00

(以下寄存器基地址 = 0x4002_0810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器10	0x00
PER13	0x003	R/W	外围允许寄存器13	0x00
PER14	0x004	R/W	外围允许寄存器14	0x00

(以下寄存器基地址 = 0x4002_1C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
HOCODIV	0x024	R/W	高速内部振荡器的频率选择寄存器	选项字节(000C2H)的FRQSEL[2:0]位的设置值

(以下寄存器基地址 = 0x4002_0C00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
MCKC	0x000	R/W	系统时钟主控制寄存器	0x00
PLLCR	0x004	R/W	PLL控制寄存器	0x00

(以下寄存器基地址 = 0x4002_2200) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SCMCTL	0x000	R/W	发振停止检测控制寄存器	0xFFFF
SCMMD	0x002	R/W	发振停止检测模式寄存器	0x00
SCMST	0x004	R/W	发振停止检测状态寄存器	0x00

4.4 寄存器说明

4.4.1 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。

位	符号	描述	复位值
7	CLS ^{注1}	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})	0
6	CSS	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})	0
5	MCS ^{注1}	主系统时钟时钟(f _{MAIN})的状态 0: 高速内部振荡时钟或PLL作为主系统时钟 (f _{MAIN}) 1: X1振荡时钟作为主系统时钟 (f _{MAIN})	0
4	MCM0	主系统时钟时钟(f _{MAIN})的选择 0: 高速内部振荡时钟或PLL作为主系统时钟 (f _{MAIN}) 1: X1振荡时钟作为主系统时钟 (f _{MAIN})	0
3:0	--	保留	0x0

注1: bit5、bit7是只读位。

注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、时钟输出/蜂鸣器输出和看门狗定时器除外）。因此，如果要更改CPU/外围硬件的时钟，就必须停止各外围功能。

4.4.2 时钟运行状态控制寄存器（CSC）

这是控制高速内部振荡器时钟运行的寄存器。通过8位存储器操作指令设置CSC寄存器。

在产生复位信号后，此寄存器的值变为“C0H”。

位	符号	描述	复位值
7	MSTOP	X1振荡时钟运行控制 0: X1振荡电路运行 1: X1振荡电路停止	1
6	XTSTOP	XT1振荡时钟运行控制 0: XT1振荡电路运行 1: XT1振荡电路停止	1
5:1	--	保留	0x00
0	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	0

注意：

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

4.4.3 PLL控制寄存器(PLLCR)

这是控制系统时钟用PLL运行的寄存器。PLL的输入时钟频率范围为4~16MHz，输入时钟通过倍频（PLL_P）后频率最大为128MHz（VCO），然后通过分频系数（PLL_PDIV）和分频系数（RDIV）得到f_{PLL}。

使用f_{PLL}做系统时钟时，必须将CKC寄存器的bit4（MCM0）和bit6（CSS）设置为0。

位	符号	描述	复位值
31:8	-	保留	-
7	PLLCKSEL	PLL输入时钟选择： 0: 高速内部振荡时钟分频后的时钟 1: X1振荡时钟	0
6:5	-	保留	0
4	PLLON	PLL模块使能： 0: 禁止 1: 使能	0
3	PLL _P DIV	PLL时钟分频选择位： 0: 2分频 1: 4分频	0
2:0	PLL _P	PLL时钟倍频选择位： 000: 4倍频 001: 8倍频 010: 12倍频 011: 16倍频 100: 24倍频 101: 32倍频 110: 64倍频 111: 128倍频	0x0

4.4.4 系统时钟主控制寄存器(MCKC)

位	符号	描述	复位值
31:8	-	保留	-
7	PLLCKSTR	系统时钟选择PLL切换成功状态位： 0: 系统时钟选择 f_{IH} 输出 1: 系统时钟选择PLL输出	0
6:4	-	保留	-
3:1	RDIV	PLL输出时钟的分频系数选择位： 000: 不分频 001: 4分频 010: 6分频 011: 8分频 100: 32分频 其他: 32分频	0x0
0	MCKSEL	系统时钟选择位： 0: 选择 F_{IH} 输出 1: 选择PLL输出	0

注：PLL的时钟输出最高64MHz。

4.4.5 时钟运行模式控制寄存器(CMC)

这是设置 X1、X2 引脚的运行模式的寄存器。

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

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述				复位值
7:6	EXCLK: OSCSEL	X1时钟引脚运行模式选择	0	0	X1、X2端口用作端口模式	0
			0	1	X1振荡模式，X1/X2引脚连接晶体或者陶瓷谐振器。	
			1	0	X1、X2端口用作端口模式	
			1	1	外部时钟输入模式，X2引脚用作外部时钟输入	
5	HSEDIV	X1输出分频	0		直接输出	0
			1		二分频输出	
4	HSEDRV	X1驱动选择	0		低驱动模式(建议10M以下)	0
			1		高驱动模式(建议10M以上)	
3:2	EXCLKS: OSCSELS	XT1时钟引脚运行模式选择	0	0	XT1、XT2端口用作端口模式	0
			0	1	XT1振荡模式，XT1/XT2引脚连接晶体或者陶瓷谐振器	
			1	0	XT1、XT2端口用作端口模式	
			1	1	外部时钟输入模式，XT2引脚用作外部时钟输入	
1			保留			0
0	LSEDRV	XT1驱动选择	0		低驱动模式	0
			1		高驱动模式	

注：EXCLKS，OSCSELS位，只在上电复位被初始化，而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振荡前，必须设置CMC寄存器
3. 必须通过软件对f_{XT}的振荡稳定时间进行计算。
4. 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。

4.4.6 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低速内部振荡器时钟切换到 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。

在产生复位信号后，此寄存器的值变为“07H”。

位	符号	描述						复位值
7:3	-	保留						0x0
2:0	OSTS[2:0]	振荡稳定时间的选择						0x7
		OSTS2	OSTS1	OSTS0		$f_X=4\text{MHz}$	$f_X=8\text{MHz}$	
		0	0	0	$2^8/f_X$	64us	32us	
		0	0	1	$2^9/f_X$	128us	64us	
		0	1	0	$2^{10}/f_X$	256us	128us	
		0	1	1	$2^{11}/f_X$	512us	256us	
		1	0	0	$2^{13}/f_X$	2.04ms	1.02ms	
		1	0	1	$2^{15}/f_X$	8.19ms	4.09ms	
		1	1	0	$2^{17}/f_X$	32.7ms	16.3ms	
		1	1	1	$2^{18}/f_X$	65.5ms	32.7ms	

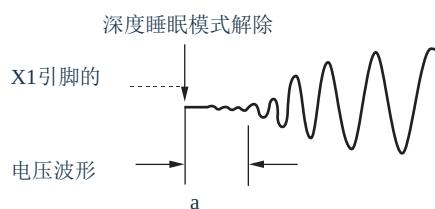
注1：要更改OSTS寄存器的设置时，必须在将时钟运行状态控制寄存器（CSC）的MSTOP位置“0”前进行更改。

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

注3：X1时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图a）。



4.4.7 振荡稳定时间计数器的状态寄存器 (OSTC)

这是表示 X1 时钟的振荡稳定时间计数器计数状态的状态寄存器。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低速内部振荡器时钟并且开始 X1 时钟的振荡时。

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

能通过 8 位存储器操作指令读 OSTC 寄存器。

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

备注：在以下情况下，振荡稳定时间计数器开始计数：

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

位	符号	描述	复位值
7	MOST8	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
6	MOST9	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
5	MOST10	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
4	MOST11	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
3	MOST13	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
2	MOST15	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
1	MOST17	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0
0	MOST18	振荡稳定时间状态 0: 未到稳定时间 1: 已到稳定时间	0

振荡稳定时间状态与X1时钟开始振荡时间的关系如下表所示。

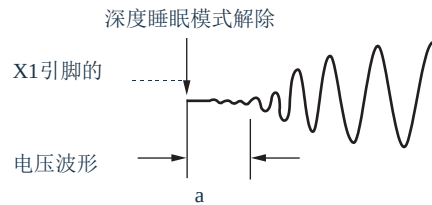
MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	振荡稳定时间状态		
									$f_x=4\text{MHz}$	$f_x=8\text{MHz}$
0	0	0	0	0	0	0	0	小于 $2^8/f_x$	小于64us	小于32us
1	0	0	0	0	0	0	0	至少 $2^8/f_x$	至少64us	至少32us
1	1	0	0	0	0	0	0	至少 $2^9/f_x$	至少128us	至少64us
1	1	1	0	0	0	0	0	至少 $2^{10}/f_x$	至少256us	至少128us
1	1	1	1	0	0	0	0	至少 $2^{11}/f_x$	至少512us	至少256us
1	1	1	1	1	0	0	0	至少 $2^{13}/f_x$	至少2.04ms	至少1.02ms
1	1	1	1	1	1	0	0	至少 $2^{15}/f_x$	至少8.19ms	至少4.09ms
1	1	1	1	1	1	1	0	至少 $2^{17}/f_x$	至少32.7ms	至少16.3ms
1	1	1	1	1	1	1	1	至少 $2^{18}/f_x$	至少65.5ms	至少32.7ms

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

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

注3：X1时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图a）。



4.4.8 外围允许寄存器（PER0、PER10、PER13、PER14）

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”。寄存器功能汇总：

PER0	RTC、ADC、IICA、SAU1、SAU0、TIMER41、TIMER40
PER10	DMA、EPWM
PER13	LCD、DAC、LED、TOUCH、ACMP
PER14	SAU2、SAU3、SAU4、OSDC

通过8位存储器操作指令设置PER0寄存器和PER10、PER13、PER14寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

外围允许寄存器0（PER0）

位	符号	描述	复位值
7	RTCCEN	提供实时时钟（RTC）和15位间隔定时器输入时钟的控制 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 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	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

外围允许寄存器10 (PER10)

位	符号	描述	复位值
7:4	-	保留	0
3	DMAEN	提供DMA模块输入时钟的控制 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	0
2	EPWMEN	提供EPWM模块输入时钟的控制 0: 停止提供输入时钟, EPWM不能运行 1: 提供输入时钟, EPWM能运行	0
1:0	-	保留	0

外围允许寄存器13 (PER13)

位	符号	描述	复位值
7	LCDEN	提供LCD模块输入时钟的控制 0: 停止提供输入时钟, LCD模块不能运行 1: 提供输入时钟, LCD模块能运行	0
6	-	保留	0
5	DACEN	提供DAC模块的输入时钟的控制 0: 停止提供输入时钟, DAC不能运行 1: 提供输入时钟, DAC能运行	0
4	LEDEN	提供LED模块的输入时钟的控制 0: 停止提供输入时钟, LED不能运行 1: 提供输入时钟, LED能运行	0
3	-	保留	0
2	TOUCHEN	提供TOUCH模块输入时钟的控制 0: 停止提供输入时钟, TOUCH不能运行 1: 提供输入时钟, TOUCH能运行	0
1	ACMPEN	提供ACMP模块输入时钟的控制 0: 停止提供输入时钟, ACMP不能运行 1: 提供输入时钟, ACMP能运行	0
0	-	保留	0

外围允许寄存器14 (PER14)

位	符号	描述	复位值
7:4	-	保留	0
3	OSDCEN	提供时钟监控OSDC模块输入时钟的控制 0: 停止提供输入时钟, OSDC不能运行 1: 提供输入时钟, OSDC能运行	0
2	SAU4EN	提供通用串行通信单元模块SAU4输入时钟的控制 0: 停止提供输入时钟, SAU4不能运行 1: 提供输入时钟, SAU4能运行	0
1	SAU3EN	提供通用串行通信单元模块SAU3输入时钟的控制 0: 停止提供输入时钟, SAU3不能运行 1: 提供输入时钟, SAU3能运行	0
0	SAU2EN	提供通用串行通信单元模块SAU2输入时钟的控制 0: 停止提供输入时钟, SAU2不能运行 1: 提供输入时钟, SAU2能运行	0

4.4.9 15位间隔定时器运行时钟选择寄存器（OSMC）

通过OSMC寄存器选择15位间隔定时器、RTC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”。

位	符号	描述	复位值
7	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 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	0
6:5	--	保留	0x0
4	WUTMMCK0	15位间隔定时器的运行时钟的选择: 0: 低速内部振荡时钟停止给15位间隔定时器/RTC提供时钟 1: 低速内部振荡时钟为15位间隔定时器/RTC提供时钟	0
3:0	--	保留	0x0

4.4.10 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。

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FRQSEL[2:0]位的设置值。

位	符号		描述	复位值
7:3	--		保留	--
2:0	HOCODIV[2:0]	--	高速内部振荡器时钟频率的选择	选项字节（000C2H）的FRQSEL[2:0]位的设置值
			FRQSEL3=1	
		000	$f_{IH}=64\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		001	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		010	$f_{IH}=16\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		011	$f_{IH}=8\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		100	$f_{IH}=4\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		101	$f_{IH}=2\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		上述以外	禁止设置	

注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。

注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。

4.4.11 低速内部振荡器时钟选择寄存器（SUBCKSEL）

SUBCKSEL寄存器是选择低速内部振荡器时钟 f_{IL} 的寄存器。

通过8位存储器操作指令设置SUBCKSEL寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	--	保留	0x0
0	SELLOCO	低速内部振荡器时钟选择 0: 选择低速外部振荡器时钟 1: 选择低速内部振荡器时钟	0

4.4.12 发振停止检测控制寄存器（SCMCTL）

发振停止检测控制寄存器（SCMCTL）是控制发振停止检测电路的动作开始，动作停止，以及设置发振停止判定时间的寄存器。

OSCDE位为0时，发振停止检测电路不开始动作。

用16位操作指令来操作SCMCTL寄存器。

位	符号	描述	复位值
15	OSDCE	发振停止检测动作的动作 0: 发振停止检测动作停止 1: 发振停止检测动作开始	0
14:12	--	保留	0x0
11:0	OSDCCMP11-0	设置发振停止判定时间 ^注 发振停止判定时间 = 内部低速振荡时钟（ f_{IL} ）周期 *((OSDCCMP11~OSDCCMP0设置值)+1)	0xFFFF

注：OSDCCMP[11:0]的值禁止设置为0x000,0x001,0x002三个值。

注意：

1. 修改OSDCCMP11~OSDCCMP0的设置值时，必须将OSDCE设置为0。（发振停止检测动作停止）
2. 发振停止检测电路通过软件设置OSDCE=0（发振停止检测动作停止）或产生端子复位以及其他内部复位来停止发振检测动作。
3. bit14-12必须设置为0。

4.4.13 发振停止检测模式寄存器（SCMMD）

发振停止检测模式寄存器（SCMMD）是选择发振停止检测的对象为主系统时钟（fmx）或者副系统时钟（fsx），以及发振停止检出后的动作是产生复位还是中断的寄存器。

用16位操作指令来操作SCMMD寄存器。

位	符号	描述	复位值
15:8	KEY	在改写MDSEL和CKSEL时，必须同时给SCMMD的高8位（KEY）写0x3C。例，复位后，SCMMD寄存器的初值为0x00，通过给SCMMD寄存器写0x3C01，来将CKSEL位置1	0x0
7:2	--	保留	0x0
1	MDSEL	发振停止检出后的动作 0: 发振停止检出后产生中断 1: 发振停止检出后产生复位	0
0	CKSEL	发振停止检测的对象 0: 检测主系统时钟的发振状态 1: 检测副系统时钟的发振状态	0

4.4.14 发振停止检测状态寄存器（SCMST）

发振停止检测状态寄存器（SCMST）是显示发振停止检测状态的寄存器。

用8位操作指令来操作SCMST寄存器。

位	符号	描述	复位值
7:1	--	保留	0x0
0	OSTDF	发振停止检测的状态 0: 发振停止没有检出 1: 发振停止有检出	0

注：发振停止有检出后，将 OSTDF 位置 1，通过写寄存器只能将其写 0。

4.5 系统时钟振荡电路

4.5.1 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。

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

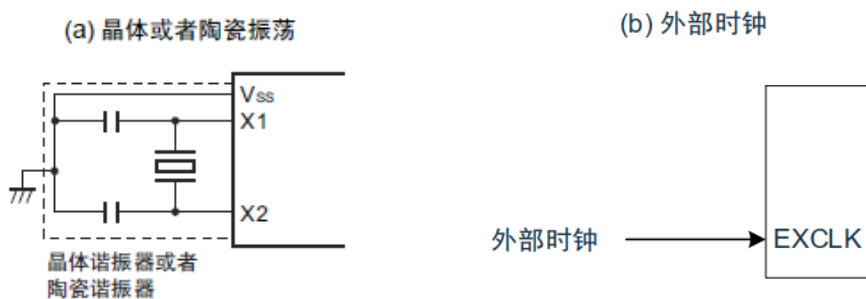
晶体或者陶瓷振荡：EXCLK、OSCSEL=0、1

外部时钟输入：EXCLK、OSCSEL=1、1

当不使用 X1 振荡电路时，必须设置为端口模式（EXCLK、OSCSEL=0、0）

X1 振荡电路的外接电路例子如下图所示。

图4-2 X1振荡电路的外接电路例子



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

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

4.5.2 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）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 bit2（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 bit3 和 bit2（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，XT1 振荡电路的外接电路例子如图 4-3 所示。

图4-3XT1振荡电路的外接电路例子



注意：当使用X1振荡电路和XT1振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图4-2和图4-3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。

4.5.3 高速内部振荡器

CMS32F759xx_737xx内置高速内部振荡器。能通过选项字节（000C2H）从f_{IH}=64MHz、32MHz、16MHz、8MHz、4MHz、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。

4.5.4 低速内部振荡器

CMS32F759xx_737xx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位间隔定时器以及SysTick定时器的外部参考时钟，也可用作CPU时钟和外围模块时钟。

当选项字节（000C0H）的bit4（WDTON）或者15位间隔定时器运行时钟选择寄存器（OSMC）的bit4（WUTMMCK0）为“1”时，或者低速内部振荡器时钟选择寄存器(SUBCKSEL)的bit0(SELLOCO)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，在WUTMMCK0为0且SELLOCO为0，WDTON位为“1”并且选项字节（000C0H）的bit0（WDSTBYON）为“0”时，如果进入深度睡眠模式或睡眠模式，低速内部振荡器就停止振荡。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。

4.5.5 PLL锁相环

CMS32F759xx_737xx内置PLL锁相环，可用作系统时钟。

PLL可以用来倍频主系统时钟，PLL的输入时钟经过倍频后最高可达128MHz（VCO），然后通过分频系数（PLLPDIV）和分频系数（RDIV）得到 f_{PLL} 。由PLL控制寄存器PLLCR的bit7（PLLCKSEL）来选择PLL的输入时钟是内部高速振荡器分频后的时钟还是X1振荡时钟，输入时钟的范围为4~16MHz，主要使用的输入频率为4/8/16MHz。由PLLCR的bit4（PLLON）来控制PLL是否工作。

当主系统时钟控制器MCKC的bit0（MCKSEL）为“1”时，选择PLL时钟作为主系统时钟，此时须将CKC寄存器的bit4（MCM0）和bit6（CSS）设置为0。

4.6 时钟发生电路的运行

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

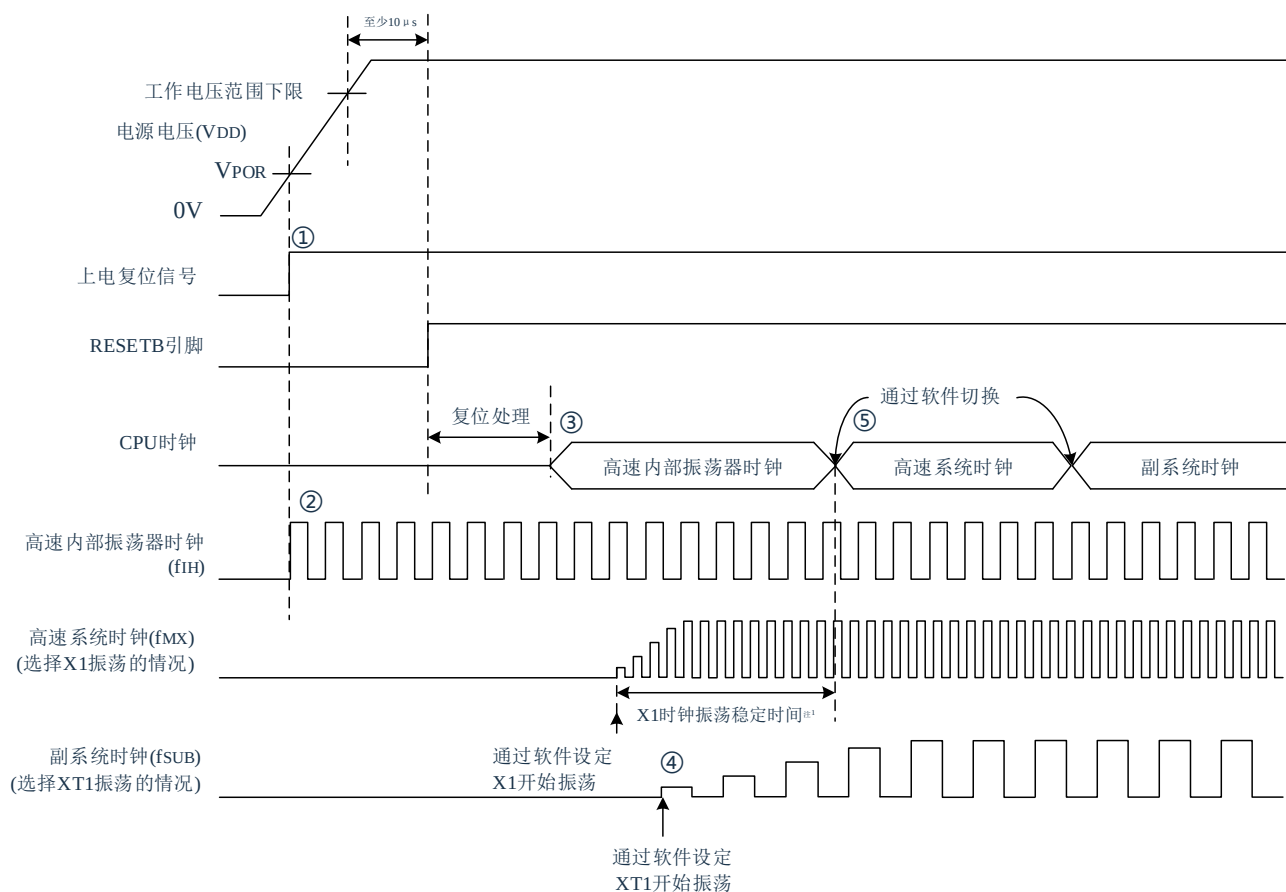
F_{MX} ：主系统时钟

F_{IH} ：高速内部振荡器时钟

F_{SUB} ：副系统时钟

CMS32F759xx_737xx 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 4-4 所示。

图4-4 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“4.7.2 X1振荡电路的设置例子和4.7.3 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“4.7.2 X1振荡电路的设置例子和4.7.3 XT1振荡电路的设置例子”）。

注：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。

4.7 时钟控制

4.7.1 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～FRQSEL3位，从64MHz、32MHz、16MHz、8MHz、4MHz、2MHz中选择高速内部振荡器的频率。另外，能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改频率。

选项字节000C2的设置

位	符号	描述			复位值
7:4	--	保留			--
3:0	FRQSEL[3:0]	高速内部振荡器时钟频率的选择			--
		FRQSEL[3:0]的值	f _{HOCO}	f _{IH}	
		1000	64MHz	64MHz	
		1001	64MHz	32MHz	
		1010	64MHz	16MHz	
		1011	64MHz	8MHz	
		1100	64MHz	4MHz	
		1101	64MHz	2MHz	
		上述以外	禁止设定		

【高速内部振荡器的频率选择寄存器（HOCODIV）的设置】

位	符号		描述	复位值
7:4	--		保留	--
3	FRQSEL3		须为1	--
2:0	HOCODIV[2:0]	--	高速内部振荡器时钟频率的选择	选项字节（000C2H）的FRQSEL[2:0]位的设置值
		000	$f_{IH}=64\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		001	$f_{IH}=32\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		010	$f_{IH}=16\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		011	$f_{IH}=8\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		100	$f_{IH}=4\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		101	$f_{IH}=2\text{MHz}$ $f_{HOCO}=64\text{MHz}$	
		上述以外	禁止设置	

4.7.2 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 EXCLK、OSCSEL 配置为 2'b01，将对应时钟输入口选择为 X1 振荡模式。

bit	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	HSEDIV	HSEDRV	EXCLKS	OSCSELS	0	LSEDRV

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 8MHz 谐振器至少等待 128us 时，必须设置 OSTS 寄存器的值为 0x02。

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

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

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

④ 通过 OSTC 寄存器等待 X1 振荡电路的振荡稳定。

例) 要通过 8MHz 谐振器至少等待 128us 时，必须等到 OSTC 寄存器的值变为 0xE0。

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

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。

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

4.7.3 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”。

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

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

bit	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	HSEDIV	HSEDRV	EXCLKS	OSCSELS	0	LSEDRV

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

bit	7	6	5	4	3	2	1	0
CSC	MSTOP1	XTSTOP0	0	0	0	0	0	HISTOP0

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。

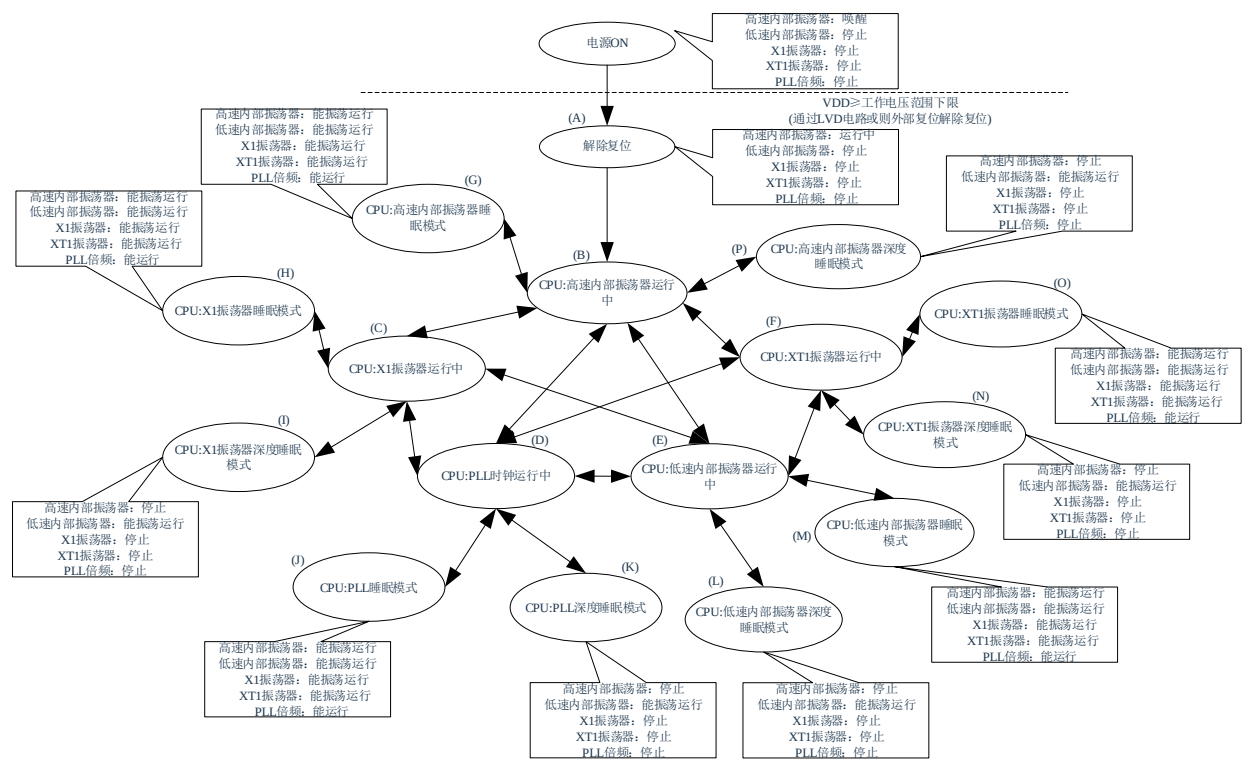
⑤ 通过 CKC 寄存器的 CSS 位将 XT1 振荡时钟设置为 CPU/外围硬件时钟。

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

4.7.4 CPU时钟的状态转移图

本产品的CPU时钟状态转移图如图4-5所示。

图4-5 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(1/4)

(1) 在解除复位(A)后，CPU转移到高速内部振荡器时钟运行(B)。

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

(2) 在解除复位(A)后，CPU转移到X1振荡器时钟运行(C)。

状态转移	SFR寄存器的设置
(B)→(C) (D)→(C) (E)→(C)	寄存器设置顺序：
	配置CKC寄存器的MCM0为1，选择X1振荡时钟作为主时钟输出
	判断CKC寄存器的MCS，等待该位为1
	配置CKC寄存器的CSS为0，选择主时钟作为CPU和外围电路的时钟
	判断CKC寄存器的CLS，等待该位为0

表4-3 CPU时钟的转移和SFR寄存器的设置例子(2/4)

(3) 在解除复位(A)后, CPU转移到PLL时钟运行(D)。

状态转移	SFR寄存器的设置
(B)→(D) (D)→(D) (E)→(D) (F)→(D)	寄存器设置顺序:
	① 配置PLLCKR寄存器的PLLCKSEL, PLLP, PLLON, 选择PLL的输入时钟源, 设置倍频系数, 使能PLL
	② 配置MCKC寄存器的MCKSEL位为1, 选择PLL时钟作为输出
	③ 判断MCKC寄存器的PLLCKSTR, 等待该位为1
	④ 配置CKC寄存器的MCM0为0, 选择PLL作为主时钟输出
	⑤ 判断CKC寄存器的MCS, 等待该位为0
	⑥ 配置CKC寄存器的CSS为0, 选择主时钟作为CPU和外围电路的时钟
	⑦ 判断CKC寄存器的CLS, 等待该位为0

(4) 在解除复位(A)后, CPU转移到低速内部振荡时钟运行(E)。

状态转移	SFR寄存器的设置
(B)→(E) (D)→(E) (C)→(E) (F)→(E)	寄存器设置顺序:
	① SUBCKSEL寄存器SELLOCO为1
	② 配置CKC寄存器的CSS为1, 选择低速内部振荡时钟为CPU和外围电路的时钟
	③ 判断CKC寄存器的CLS, 等待该位为1

(5) 在解除复位(A)后, CPU转移到低速外部振荡时钟运行(F)。

状态转移	SFR寄存器的设置
(B)→(F) (D)→(F) (E)→(F)	寄存器设置顺序:
	④ SUBCKSEL寄存器SELLOCO为0
	⑤ 配置CKC寄存器的CSS为1, 选择低速内部振荡时钟为CPU和外围电路的时钟
	⑥ 判断CKC寄存器的CLS, 等待该位为1

(6) CPU由其他时钟转移到高速内部振荡时钟运行(B)。

状态转移	SFR寄存器的设置
(C)→(B) (D)→(B) (E)→(B) (F)→(B)	寄存器设置顺序:
	① 配置MCKC寄存器的MCKSEL位为0, 选择内部高速振荡器时钟作为输出
	② 判断MCKC寄存器的PLLCKSTR, 等待该位为0
	③ 配置CKC寄存器的MCM0为0, 选择高HOCO作为主时钟输出
	④ 判断CKC寄存器的MCS, 等待该位为0
	⑤ 配置CKC寄存器的CSS为0, 选择主时钟作为CPU和外围电路的时钟
	⑥ 判断CKC寄存器的CLS, 等待该位为0

注1: 表4-3的(A)~(P)对应图4-4的(A)~(P)。

注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(3/4)

- (7) CPU从高速内部时钟运行(B)转移到睡眠模式(G)。
CPU从高速外部时钟运行(C)转移到睡眠模式(H)。
CPU从PLL时钟运行(D)转移到睡眠模式(J)。
CPU从低速内部时钟运行(E)转移到睡眠模式(M)。
CPU从低速外部时钟运行(F)转移到睡眠模式(O)。

状态转移	设置内容
(B)→(G) (C)→(H) (D)→(J) (E)→(M) (F)→(O) (E)→(M) (F)→(O) (E)→(M) (F)→(O)	执行WFI指令

注：表4-3的(A)~(P)对应图4-4的(A)~(P)。

表4-3 CPU时钟的转移和SFR寄存器的设置例子(4/4)

- (8) CPU从高速内部时钟运行(B)转移到深度睡眠模式(I)。
CPU从高速外部时钟运行(C)转移到深度睡眠模式(K)。
CPU从PLL时钟运行(D)转移到深度睡眠模式(L)。
CPU从低速内部时钟运行(E)转移到深度睡眠模式(N)。
CPU从低速外部时钟运行(F)转移到深度睡眠模式(P)。

(设置顺序)

状态转移	设置内容	
(B)→(I) (C)→(K) (D)→(L) (E)→(N) (F)→(P)	停止不能在深度睡眠模式中运行的外围功能	SCR寄存器bit2 (SLEEPDEEP) 置为1, 并执行WFI指令

注：表4-3的(A)~(P)对应图4-4的(A)~(P)。

4.7.5 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

CPU时钟		转移前的条件	转移后的处理
转移前	转移后		
高速内部振荡时钟	低速内部振荡时钟	选择低速内部振荡时钟 • SELLOCO=1	如果停止高速内部振荡器的振荡（HIOSTOP=1），就能减小工作电流。 当PLL时钟源为高速内部振荡器时钟，CPU时钟切换到PLL时，不能停止高速内部振荡器
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
	XT1振荡时钟	XT1振荡稳定。 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
低速内部振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	若看门狗和LSITIMER没有运行，可关闭低速内部时钟振荡(SELLOCO=0) 就能减小工作电流。
	PLL时钟	允许PLL运行。 • PLLON=1 • PLL输入时钟源振荡稳定	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
	XT1振荡时钟	XT1振荡稳定。 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
PLL时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	可以关闭PLL使能(PLLON=0)，以减少工作电流
	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	
	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	
	XT1振荡时钟	XT1振荡稳定 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
X1振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	如果停止X1振荡时钟(MSTOP=1)，可以减少工作电流。当PLL时钟源为X1振荡时钟，CPU时钟切换到PLL时钟时，不能停止X1时钟振荡。
	低速内部振荡时钟	选择低速内部振荡时钟 • SELLOCO=1	
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	
	XT1	XT1振荡稳定 • OSCSELS=1,EXCLKS=0,XTSTOP=0 • 经过振荡稳定时间后	
XT1振荡时钟	高速内部振荡时钟	允许高速内部振荡器振荡 • HIOSTOP=0 • 经过振荡稳定时间后	
	低速内部振荡时钟	选择低速内部振荡时钟 SELLOCO=1	
	PLL时钟	允许PLL运行 • PLLON=1 • PLL输入时钟源振荡稳定	

	X1振荡时钟	X1振荡稳定 • OSCSEL=1,EXCLK=0,MSTOP=0 • 经过振荡稳定时间后	

4.7.6 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(CSS)、bit4(MCM0)进行CPU时钟的切换(主系统时钟和副系统时钟)和主系统时钟的切换(高速系统时钟和X1振荡时钟)。

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)。

能通过CKC寄存器的bit7(CLS)、bit5(MCS)和MCKC寄存器的bit7(PLLCKSTR)来判断CPU是运行时钟。

如果切换CPU时钟，就同时切换外围硬件时钟。

表4-5 时钟切换所需要时钟数

切换后时钟 切换前时钟	高速内部振荡器时钟 (f _{IH})	X1振荡时钟(f _X)	PLL(f _{PLL})	低速内部振荡器时钟 (f _{IL})	XT1振荡时钟
高速内部振荡器时钟(f _{IH})		2.5个f _{IH} / f _X 时钟	2.5个f _{IH} / f _{PLL} 时钟	2.5个f _{IH} / f _{IL} 时钟	2.5个f _{IH} / f _{XT} 时钟
X1振荡时钟(f _X)	2个时钟		2.5个f _X / f _{PLL} 时钟	2.5个f _X / f _{IL} 时钟	2.5个f _X / f _{XT} 时钟
PLL(f _{PLL})	2个时钟	2.5个f _{PLL} / f _X 时钟		2.5个f _{PLL} / f _{IL} 时钟	2.5个f _{PLL} / f _{XT} 时钟
低速内部振荡器时钟(f _{IL})	2个时钟	2个时钟	2个时钟		2.5个f _{PLL} / f _{XT} 时钟
XT1振荡时钟	2个时钟	2个时钟	2个时钟	2个时钟	

注1：表4-5中的时钟数是切换前的CPU时钟数。

注2：表4-5中的时钟数是舍入小数部分的时钟数。

例：CPU从主系统时钟切换到副系统时钟的情况（选择f_{MAIN}=f_{IH}=2MHz、f_{SUB}=32.768KHz振荡的情况）

$$2.5f_{\text{MAIN}}/f_{\text{SUB}}=2.5(2000/32.768)=152.6\approx 153\text{个时钟}$$

4.7.7 时钟振荡停止前的条件

用于停止时钟振荡的寄存器标志设置和停止前的条件如下所示。

表4-6 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	SFR寄存器的标志
高速内部振荡器时钟	CLS=1 (CPU以高速内部振荡器时钟或PLL时钟以外的时钟运行)	HIOSTOP=1
X1振荡时钟	MCS=0或者CLS=1; 当以PLL时钟运行时PLLCKSEL=0 (CPU以高速系统时钟以外的时钟运行,且当以PLL时钟运行时, 时钟源不为X1振荡时钟)	MSTOP=1
PLL时钟	MCS=0或者CLS=1或PLLCKSEL=0 (CPU以PLL时钟以外的时钟运行)	PLLON=0
低速内部振荡器时钟	CLS=0,WDTON=0,WUMMCK0=0 (CPU以低速内部振荡器时钟以外的时钟运行)	SELLOCO=0
XT1振荡时钟	CLS=0 (CPU以副系统时钟以外的时钟运行)	XTSTOP=1

4.8 发振停止检出电路

4.8.1 发振停止检出电路的功能

发振停止检出功能是用内部低速振荡时钟（ f_{IL} ）来监控主系统时钟或者副系统时钟的动作状态，在一段时间内，检测到动作停止时，判断X1发振电路或XT1发振电路发生异常，输出发振停止检出信号，该信号可做中断信号，也可做复位信号。

发振停止检测电路，需要在复位解除后通过软件设置来使能。

发振停止检测电路，通过软件设置来停止检测动作。或者，因为发生端子复位或其他内部复位而停止发振检测动作。复位发生后，需要再次通过软件设置来使能发振停止检测动作。

发振停止检测电路判断发振停止的时间（发振停止判定时间）由发振停止检测控制寄存器（OSDC）的OSDCCMP11~OSDCCMP0来设置。

发振停止判定时间 = 内部低速振荡时钟（ f_{IL} ）周期 $\times ((OSDCCMP11 \sim OSDCCMP0 \text{ 设置值}) + 1)$

以内部低速振荡时钟（ f_{IL} ）频率为32K为例：

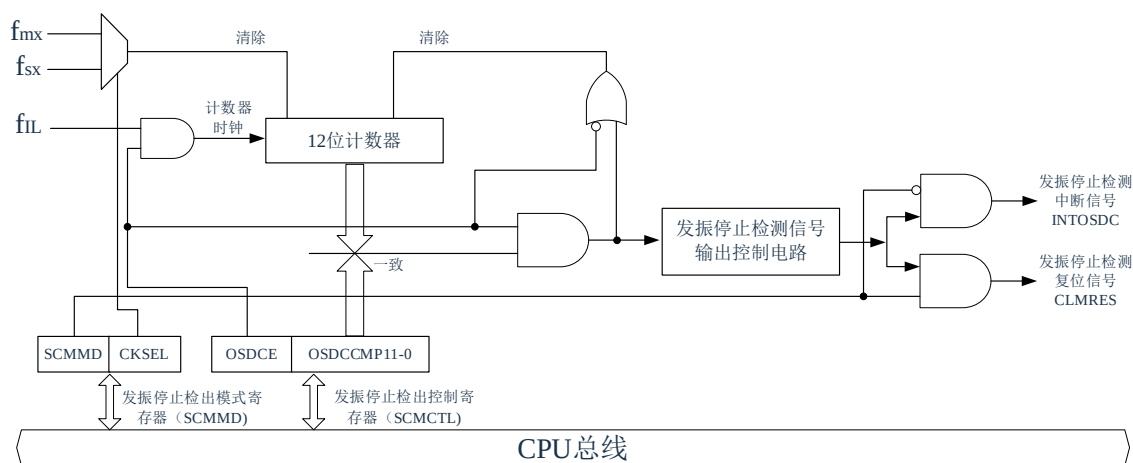
OSDCCMP11~OSDCCMP0=003H时：108us(MIN.),125us(TYP.),147us(MAX.)

OSDCCMP11~OSDCCMP0=FFFH时：111ms(MIN.),128ms(TYP.),150ms(MAX.)

4.8.2 发振停止检测电路的构成

发振停止检测电路由以下框图构成。

图4-6 发振停止检测电路的框图（例）



4.8.3 发振停止检测电路的动作

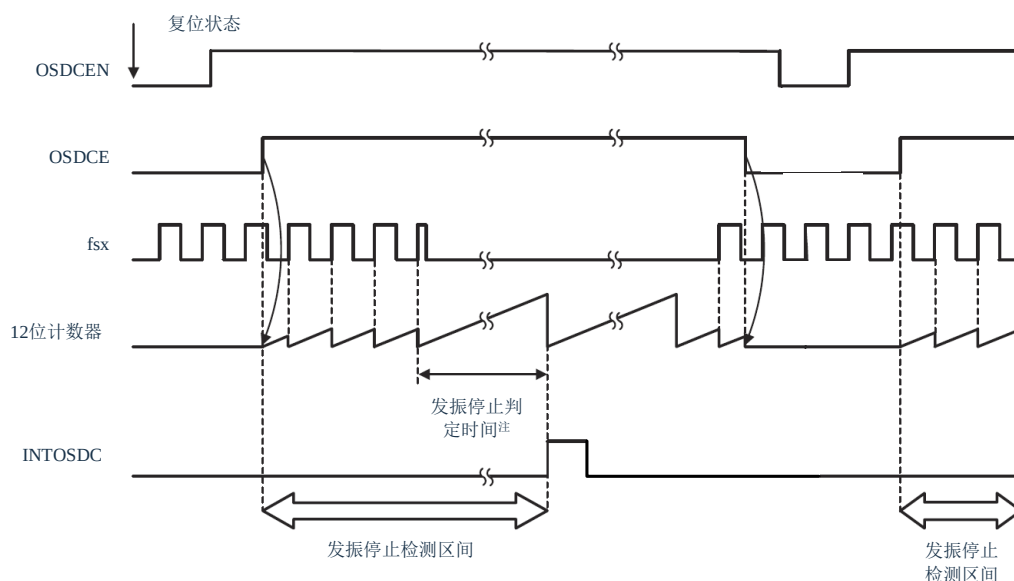
4.8.3.1 发振停止检测电路的动作方法

外部复位解除后，主系统时钟（fmx）/副系统时钟（fsx）开始发振。

- (1) 写外围使能寄存器（PER14），使能发振停止检测电路注。
- (2) 写发振停止检测模式寄存器（SCMMD），选择检测主系统时钟（fmx）或副系统时钟（fsx）的发振状态，并选择检测到发振停止后的动作，是产生复位还是产生中断。
- (3) 写发振停止检测控制寄存器（SCMCTL）的OSDCE位，发振停止检测电路开始动作。
- (4) 发振停止检测电路动作过程中，主系统时钟（fmx）/副系统时钟（fsx）在发振停止判定时间内一直是停振的，输出发振停止检出信号，产生复位或者中断。
- (5) 当CPU时钟为主系统时钟（fmx）或者CPU时钟是PLL时钟并用主系统时钟（fmx）作为PLL输入，且有主系统时钟（fmx）发振停止检出时，CPU时钟会切换到内部高速振荡时钟的8分频（fHOCO/8），当CPU时钟为副系统时钟（fsx）且有副系统时钟（fsx）发振停止检出时，CPU时钟会切换到内部低速振荡时钟（fL）。通过软件写SCMST寄存器为0来清除OSTDF会将CPU时钟切回原来的时钟。

注：由于PER14的bit3（OSDCEN）写1时，可能会有误检出，故使用发振停止检出中断时，必须在写完PER14之后，先清中断标志位，再打开中断使能。发振停止检测中断对应的相关中断寄存器请参考“第22章 嵌套向量中断控制器”。

图4-7 发振停止检测电路的时序（以检测对象为fsx，发振停止检出后产生中断为例）



注：发振停止判定时间=内部低速振荡时钟（fL）周期*((OSDCCMP11~OSDCCMP0设置值)+1)。

4.8.4 发振停止检测功能的注意事项

发振停止检测电路，跟独立看门狗定时器一起使用。

发振停止检测，在以下的任意条件下可以使用：

选项字节（00C0H）的bit0（WDSTBYON），bit4（WDTON）为1，OSMC寄存器的bit4（WUTMMCK0）为0时；

OSMC寄存器的bit4（WUTMMCK0）为1时。

第5章 硬件除法器 (DIV)

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

5.1 特点

- ◆ 32位有符号（2的补码）整数除法计算
- ◆ 32位有符号被除数，32位有符号除数
- ◆ 32位有符号商和32位有符号余数输出
- ◆ 写除数寄存器自动触发除法计算
- ◆ 除0警告标志
- ◆ 指示运算中的BUSY标志
- ◆ 有计算结束的中断请求

5.2 功能描述

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

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

5.3 寄存器映射

以下寄存器基地址：DIV_BASE = 4001_A000H； RO：只读，WO：只写，R/W：读写

寄存器名称	偏移量	R/W	寄存器描述	复位值
DIVIDEND	0x00	R/W	被除数寄存器	0x0
DIVISOR	0x04	R/W	除数寄存器	0x1
QUOTIENT	0x08	RO	商寄存器	0x0
REMAINDER	0x0C	RO	余数寄存器	0x0
STATUS	0x10	RO	状态寄存器	0x0

5.4 寄存器说明

5.4.1 被除数寄存器 (DIVIDEND)

被除数寄存器是保存被除数的寄存器，其数值作为32位有符号整数参与除法运算。

位	符号	描述	复位值
31:0	DIVIDEND	32位被除数	0

5.4.2 除数寄存器 (DIVISOR)

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

位	符号	描述	复位值
31:0	DIVISOR	32位除数	0x1

5.4.3 商寄存器 (QUOTIENT)

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

位	符号	描述	复位值
31:0	QUOTIENT	32位结果的商	0x0

5.4.4 余数寄存器 (REMAINDER)

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

位	符号	描述	复位值
31:0	REMAINDER	32位计算结果的余数	0x0

5.4.5 状态寄存器 (STATUS)

通过状态寄存器可以查询硬件除法器的状态，包括除零标志和BUSY标志。

位	符号	描述	复位值
31:2	-	保留，必须为0	0x0
1	DIVBYZERO	指示除零的情况，每次写除数寄存器时更新 0: 除数不为0 1: 除数为0	0
0	BUSY	指示除法运算状态 0: 除法运算完成 1: 除法运算进行中	0

第6章 通用定时器单元Timer4

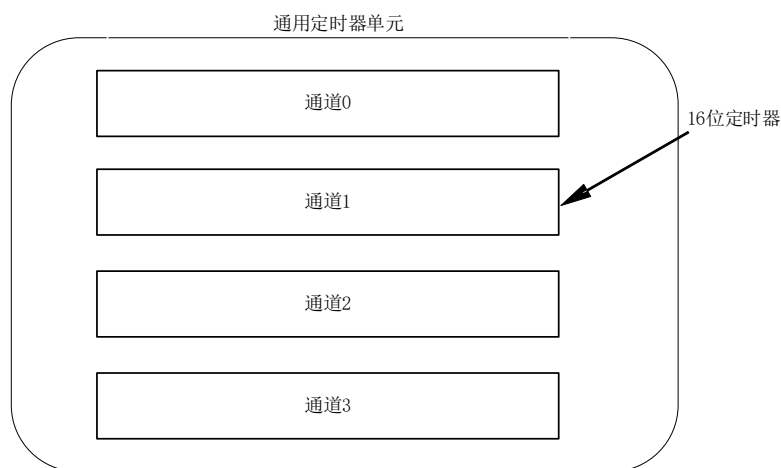
本制品搭载两个通用定时器单元,每个单元含有4个通道。

说明:

- 1、本章下文中的标号“m”代表单元号,本产品搭载了两个通用定时器Timer4,因此m=0, 1。
- 2、本章下文中的标号“n”代表通道号(在本章中n=0~3)。
- 3、本章的下述内容主要针对64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
• 间隔定时器 (参照 6.7.1) • 方波输出 (参照6.7.1) • 外部事件计数器 (参照 6.7.2) • 分频器 (参照 6.7.3) • 输入脉冲间隔的测量 (参照6.7.4) • 输入信号的高低电平宽度的测量 (参照 6.7.5) • 延迟计数器 (参照 6.7.6)	• 单触发脉冲输出 (参照 6.8.1) • PWM输出 (参照6.8.2) • 多重PWM输出 (参照 6.8.3)

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)

6.1 通用定时器的功能

通用定时器单元有以下功能：

6.1.1 独立通道运行功能

独立通道运行功能是不受其他通道运行模式的影响而能独立使用任意通道的功能。

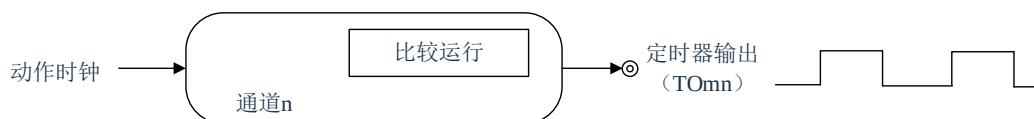
间隔定时器

能用作以固定间隔产生中断（INTTmn）的基准定时器。



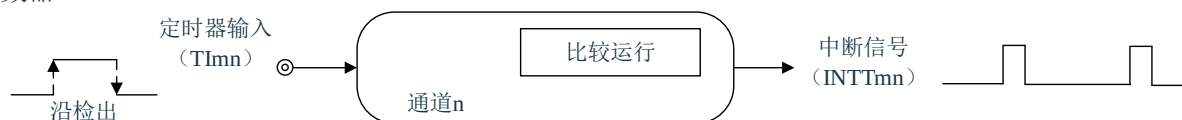
方波输出

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



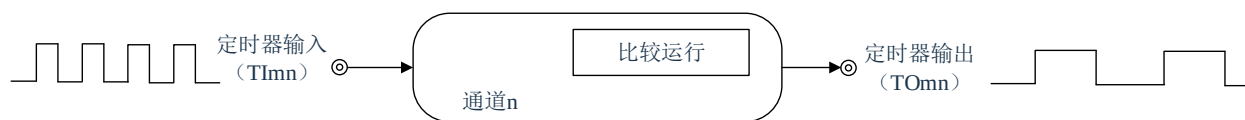
外部事件计数器

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。



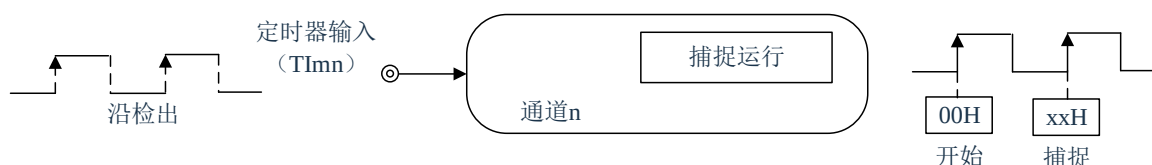
分频器功能（只限于单元 m 的通道 0）

对定时器输入引脚（TIm0）的输入时钟进行分频，然后从输出引脚（TOm0）输出。



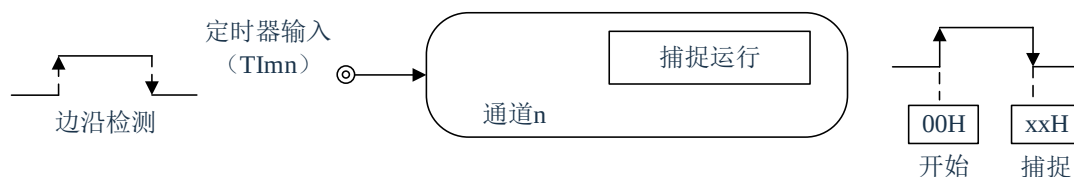
输入脉冲间隔的测量

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。



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。



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。



注1：m：单元号（m=0~1） n：通道号（n=0~3）

注2：通道0~3的定时器输入/输出引脚可配置请参照“第2章 引脚功能”。

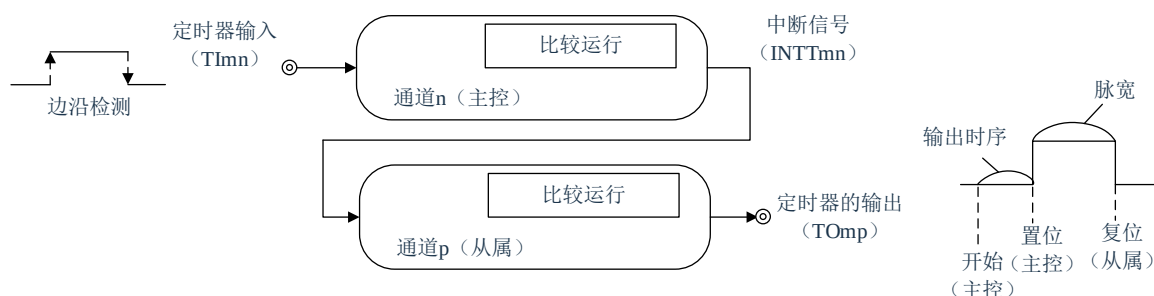
6.1.2 多通道联动运行功能

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

多通道联动运行功能可用作以下模式。

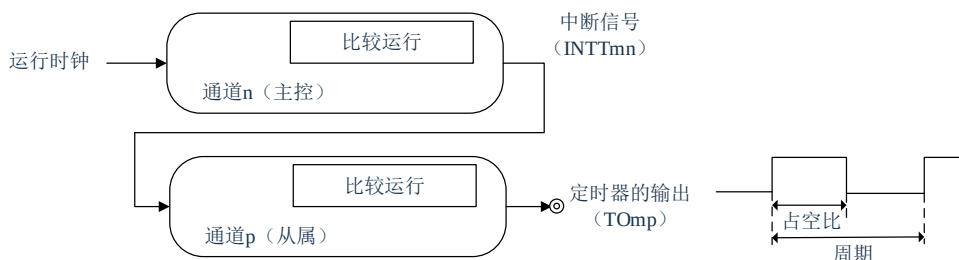
单触发脉冲输出

将2个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



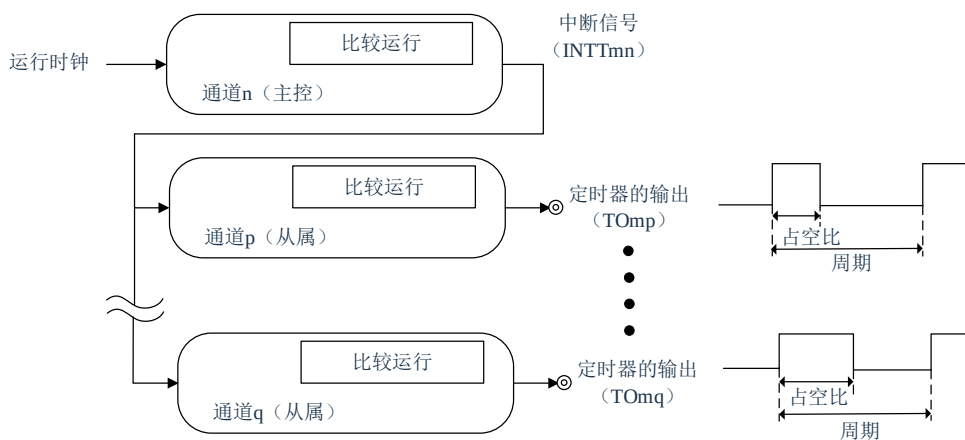
PWM（Pulse Width Modulation）输出

将2个通道成对使用，生成能任意设定周期和占空比的脉冲。



多重 PWM（Pulse Width Modulation）输出

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



注1：有关多通道联动运行功能规则的详细内容，请参照“6.3.1 多通道联动运行功能的基本规则”。

注2：m：单元号（m=0~1） n：通道号（n=0~3） p、q：从属通道号（n < p < q ≤ 3）。

6.1.3 8 位定时器运行功能（只限于通道1和通道3）

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

注意：在使用8 位定时器运行功能时，有几个规则。

详细内容请参照“6.3.2 8位定时器运行功能的基本规则（只限于通道1和通道3）”。

6.1.4 LIN-bus 支持功能（只限于单元0 的通道3）

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

(1) 唤醒信号的检测

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

(2) 间隔段的检测

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

(3) 同步段脉宽的测量

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

注：有关LIN-bus 支持功能的运行设定，请参照“13.3.13 输入切换控制寄存器（ISC）”和“6.7.5 作为输入信号高低电平宽度测量的运行”。

6.2 通用定时器单元的结构

通用定时器单元由以下硬件构成。

表6-1 通用定时器单元的结构

项目	结构
计数器	定时器计数寄存器mn (TCRmn)
寄存器	定时器数据寄存器mn (TDRmn)
定时器的输入	TI10~TI13、TI00~TI03注1
定时器的输出	TO00~TO03注1、TO10~TO13、输出控制电路
控制寄存器	<单元设定部的寄存器> • 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m (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> • 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 • 端口输出复用功能配置寄存器 (PS_TO00-PS_TO03、PS_TO10-PS_TO13) 注3 • 端口输入复用功能配置寄存器 (PS_TI00-PS_TI03、PS_TI10-PS_TI13) 注3

注1：通用定时器单元0和通用定时器单元1的输入/输出引脚复用到固定端口。详细内容请参照“第2章 引脚功能”。

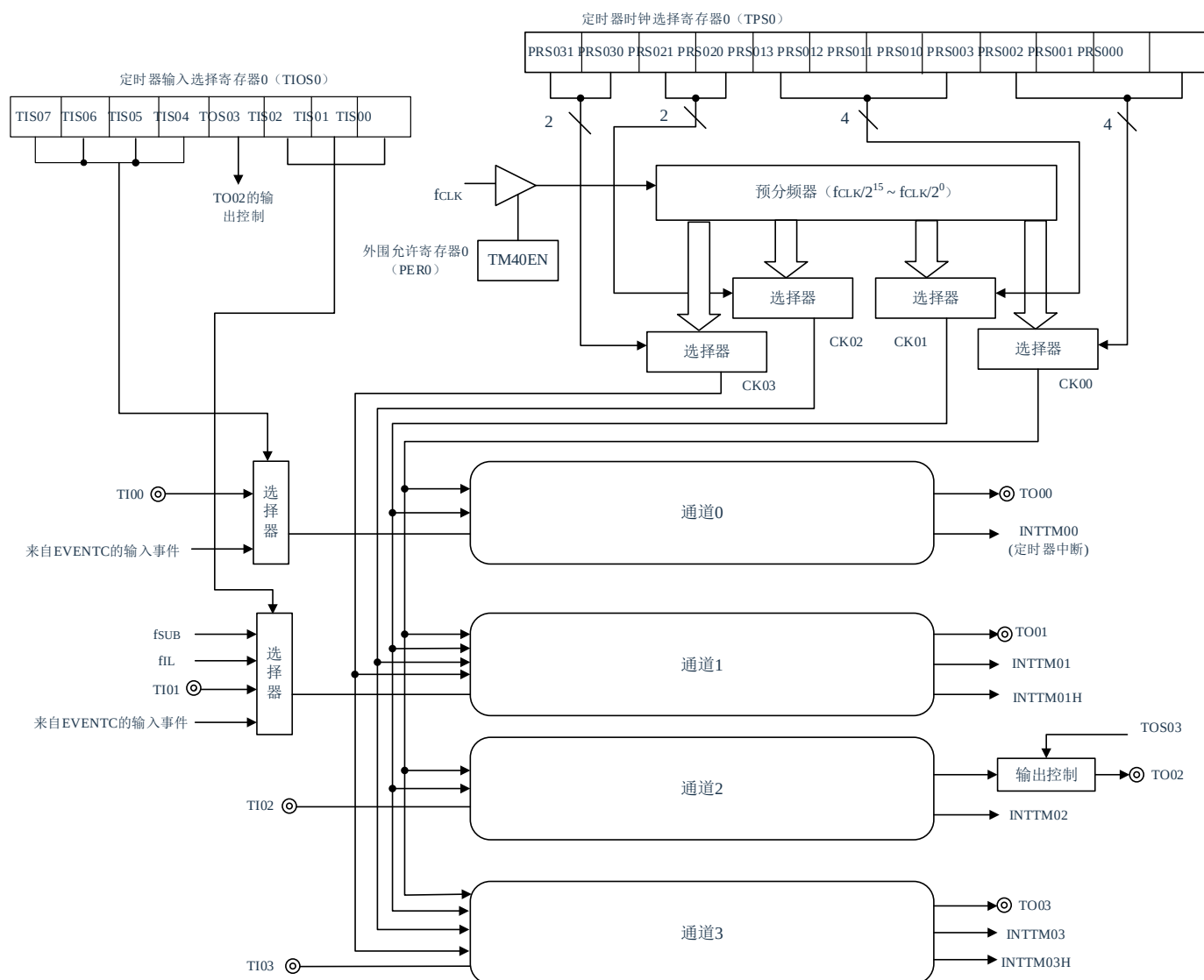
注2：通用定时器单元0和通用定时器单元1的通道选择

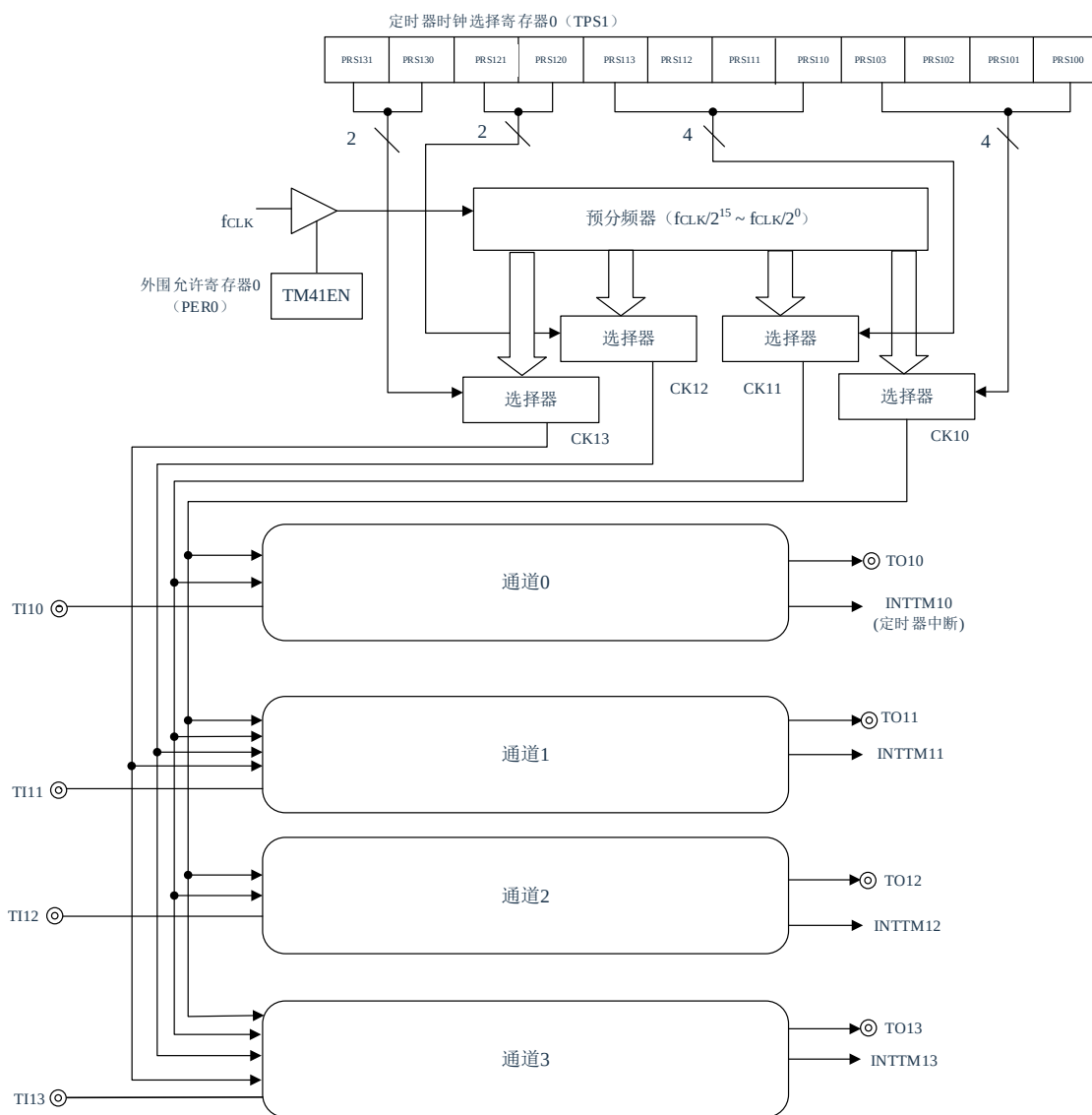
注3：用于通道0~3的定时器输入/输出引脚配置。详细内容请参照“第2章 引脚功能”。

注4：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 6-1 所示。

图6-1 通用定时器单元m (0~1) 的整体框图





6.2.1 寄存器映射

(以下寄存器基地址 = 0x4004_1D80) RO: 只读, WO: 只写, R/W: 读写

寄存器名	偏移地址	读写属性	位宽	描述	复位值
TCR00	0x000	RO	16	定时器通道 0 计数寄存器	0xFFFF
TCR01	0x002	RO	16	定时器通道 1 计数寄存器	0xFFFF
TCR02	0x004	RO	16	定时器通道 2 计数寄存器	0xFFFF
TCR03	0x006	RO	16	定时器通道 3 计数寄存器	0xFFFF
TMR00	0x010	R/W	16	定时器通道 0 模式寄存器	0x0000
TMR01	0x012	R/W	16	定时器通道 1 模式寄存器	0x0000
TMR02	0x014	R/W	16	定时器通道 2 模式寄存器	0x0000
TMR03	0x016	R/W	16	定时器通道 3 模式寄存器	0x0000
TSR00	0x020	RO	16	定时器通道 0 状态寄存器	0x0000
TSR01	0x022	RO	16	定时器通道 1 状态寄存器	0x0000
TSR02	0x024	RO	16	定时器通道 2 状态寄存器	0x0000
TSR03	0x026	RO	16	定时器通道 3 状态寄存器	0x0000
TE0	0x030	RO	16	定时器通道允许状态寄存器	0x0000
TS0	0x032	R/W	16	定时器通道开始寄存器	0x0000
TT0	0x034	R/W	16	定时器通道停止寄存器	0x0000
TPS0	0x036	R/W	16	定时器时钟选择寄存器	0x0000
TO0	0x038	R/W	16	定时器输出寄存器	0x0000
TOE0	0x03A	R/W	16	定时器输出允许寄存器	0x0000
TOL0	0x03C	R/W	16	定时器输出电平寄存器	0x0000
TOM0	0x03E	R/W	16	定时器输出模式寄存器	0x0000
TOM0L	0x03E	R/W	8	定时器输出模式寄存器低 8 位	0x00
TDR00	0x198	R/W	16	定时器通道 0 数据寄存器	0x0000
TDR01	0x19A	R/W	16	定时器通道 1 数据寄存器	0x0000
TDR02	0x1E4	R/W	16	定时器通道 2 数据寄存器	0x0000
TDR03	0x1E6	R/W	16	定时器通道 3 数据寄存器	0x0000

(以下寄存器基地址 = 0x4004_2180) RO: 只读, WO: 只写, R/W: 读写

寄存器名	偏移地址	读写属性	位宽	描述	复位值
TCR10	0x000	RO	16	定时器通道 0 计数寄存器	0xFFFF
TCR11	0x002	RO	16	定时器通道 1 计数寄存器	0xFFFF
TCR12	0x004	RO	16	定时器通道 2 计数寄存器	0xFFFF
TCR13	0x006	RO	16	定时器通道 3 计数寄存器	0xFFFF
TMR10	0x010	R/W	16	定时器通道 0 模式寄存器	0x0000
TMR11	0x012	R/W	16	定时器通道 1 模式寄存器	0x0000
TMR12	0x014	R/W	16	定时器通道 2 模式寄存器	0x0000
TMR13	0x016	R/W	16	定时器通道 3 模式寄存器	0x0000
TSR10	0x020	RO	16	定时器通道 0 状态寄存器	0x0000
TSR11	0x022	RO	16	定时器通道 1 状态寄存器	0x0000
TSR12	0x024	RO	16	定时器通道 2 状态寄存器	0x0000
TSR13	0x026	RO	16	定时器通道 3 状态寄存器	0x0000
TE1	0x030	RO	16	定时器通道允许状态寄存器	0x0000
TS1	0x032	R/W	16	定时器通道开始寄存器	0x0000
TT1	0x034	R/W	16	定时器通道停止寄存器	0x0000
TPS1	0x036	R/W	16	定时器时钟选择寄存器	0x0000
TO1	0x038	R/W	16	定时器输出寄存器	0x0000
TOE1	0x03A	R/W	16	定时器输出允许寄存器	0x0000
TOL1	0x03C	R/W	16	定时器输出电平寄存器	0x0000
TOM1	0x03E	R/W	16	定时器输出模式寄存器	0x0000
TOM1L	0x03E	R/W	8	定时器输出模式寄存器低 8 位	0x00
TDR10	0x198	R/W	16	定时器通道 0 数据寄存器	0x0000
TDR11	0x19A	R/W	16	定时器通道 1 数据寄存器	0x0000
TDR12	0x1E4	R/W	16	定时器通道 2 数据寄存器	0x0000
TDR13	0x1E6	R/W	16	定时器通道 3 数据寄存器	0x0000

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
NFEN1	0x001	R/W	噪声滤波器允许寄存器1	0x0
NFEN2	0x002	R/W	噪声滤波器允许寄存器2	0x0
TIOS0	0x004	R/W	定时器输入输出选择寄存器	0x0

6.2.2 定时器计数寄存器mn (TCRmn)

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

通过定时器模式寄存器mn (TMRmn) 的MDmn3~MDmn0位来选择运行模式，进行递增和递减计数的切换（参照“6.2.6定时器模式寄存器mn (TMRmn)”）。

表6-2 定时器计数寄存器 mn (TCRmn) 的表格

位	符号	描述	复位值
15:0	TCRmn	计数时钟计数寄存器(只读)	0xFFFF

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TM40EN位和TM41EN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：即使读TCRmn寄存器，也不将计数值捕捉到定时器数据寄存器 mn (TDRmn)。

如下所示，TCRmn寄存器的读取值因运行模式和运行状态而不同。

表6-3 各运行模式中的定时器计数寄存器mn (TCRmn) 的读取值

运行模式	计数方式	定时器计数寄存器mn (TCRmn) 的读取值 ^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	0xFFFF	停止时的值	不定值	—
捕捉模式	递增计数	0x0000	停止时的值	不定值	—
事件计数器模式	递减计数	0xFFFF	停止时的值	不定值	—
单次计数模式	递减计数	0xFFFF	停止时的值	不定值	0xFFFF
捕捉&单次计数模式	递增计数	0x0000	停止时的值	不定值	TDRmn寄存器的捕捉值+1

注1：m：单元号（m=0~1） n：通道号（n=0~3）

注2：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。

6.2.3 定时器数据寄存器mn（TDRmn）

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

能随时改写 TDRmn 寄存器。

能以 16 位为单位读写此寄存器。

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

在产生复位信号后，TDRmn 寄存器的值变为“0000H”。

表6-4 通道0定时器数据寄存器TDRm0

位	符号	描述	复位值
15:0	TDRm0	定时器通道0数据寄存器	0x0000

表6-5 通道1定时器数据寄存器TDRm1

位	符号	描述	复位值
15:8	TDRm1H	定时器通道1数据寄存器bit15:8	0x00
7:0	TDRm1L	定时器通道1数据寄存器bit7:0	0x00

表6-6 通道2定时器数据寄存器TDRm2

位	符号	描述	复位值
15:0	TDRm2	定时器通道2数据寄存器	0x00

表6-7 通道3定时器数据寄存器TDRm3

位	符号	描述	复位值
15:8	TDRm3H	定时器通道3数据寄存器bit15:8	0x00
7:0	TDRm3L	定时器通道3数据寄存器bit7:0	0x00

(i) 定时器数据寄存器mn（TDRmn）用作比较寄存器的情况

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

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

(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）来设定捕捉触发的选择。

注：m：单元号（m=0~1） n：通道号（n=0~3）。

6.2.4 外围允许寄存器0（PER0）

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

要使用通用定时器单元 0 时，必须将 bit0（TM40EN）置“1”，使用通用定时器单元 1 时，必须将 bit1（TM41EN）置“1”。通过 8 位存储器操作指令设定 PER0 寄存器。在产生复位信号后，PER0 寄存器的值变为“00H”。

表6-8 外围允许寄存器0（PER0）的表格

位	符号	描述	复位值
7	RTCEN	提供RTC模块输入时钟的控制 0: 停止提供输入时钟，RTC模块不能运行 1: 提供输入时钟，RTC模块能运行	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

注：要设定通用定时器单元时，必须先在TM40EN位或者TM41EN位为“1”的状态下设定以下的寄存器。当TM40EN位或者TM41EN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m（TIOSm）、端口模式控制寄存器PMCx、端口模式寄存器PMx除外）。

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

6.2.5 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定 TPSm 寄存器。在产生复位信号后，TPSm 寄存器的值变为“0000H”。

表6-9 定时器时钟选择寄存器m (TPSm) 的表格

位	符号	描述	复位值
15:14	-	必须为0	0x0
13:12	CKm3	定时器运行时钟选择CKm3: 00: $f_{clk}/2^8$ 01: $f_{clk}/2^{10}$ 10: $f_{clk}/2^{12}$ 11: $f_{clk}/2^{14}$	0x0
11:10	-	必须为0	0x0
9:8	CKm2	定时器运行时钟选择CKm2: 00: $f_{clk}/2$ 01: $f_{clk}/2^2$ 10: $f_{clk}/2^4$ 11: $f_{clk}/2^6$	0x0
7:4	CKm1	定时器运行时钟选择 $f_{clk}/2^{CKm1}$	0x0
3:0	CKm0	定时器运行时钟选择 $f_{clk}/2^{CKm0}$	0x0

注1: f_{CLK} : CPU/外围硬件的时钟频率。

注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“6.4.1 计数时钟（ f_{TCLK} ）”。

注3: 必须将bit15、14、11、10置“0”。

注4: 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5: 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。

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6-10 运行时钟CKSm2和CKSm3能设定的间隔时间

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

- 注1：包含5%以内的误差。
- 注2：f_{CLK}：CPU/外围硬件的时钟频率
- 注3：有关TPSm寄存器所选f_{CLK}/2ⁿ波形的详细内容，请参照“6.4.1 计数时钟（f_{TCLK}）”。

6.2.6 定时器模式寄存器mn (TMRmn)

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

禁止在运行中 (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“6.7 通用定时器单元的独立通道运行功能”和“6.8 定时器阵列单元的多通道联动运行功能”）。

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

注：TMRmn 寄存器的 bit11 因通道而不同。

TMRm: MASTERmn 位 (n=2)

TMRm1、TMRm3: SPLITmn 位 (n=1、3)

TMRm0: 固定为“0”。

表6-11 定时器通道0模式寄存器TMRm0

位	符号	描述	复位值
15:14	CKSm01- CKSm00	通道n运行时钟 (f_{MCK}) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 (f_{CLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	0	必须为0	0
12	CCSm0	通道n计数时钟 (f_{CLK}) 的选择 0: CKSm00位和CKSm01位指定的运行时钟 (f_{MCK}) 1: TIOSm选择的输入信号的有效边沿 计数时钟 (f_{CLK}) 用于计数器、输出控制电路和中断控制电路	0
11	0	必须为0	0
10:8	STSm02- STSm00	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效（不选择其他触发源）。 001: 将TIm0引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TIm0引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 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） 其他 禁止设定	0x0
7:6	CISm01- CISm00	TIm0引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿（测量低电平宽度时） 开始触发：下降沿，捕捉触发：上升沿 11: 双边沿（测量高电平宽度时） 开始触发：上升沿，捕捉触发：下降沿	0x0
5:4	0	保留	0x0

3:0	MDm03- MDm00	通道n运行模式及中断的设定 0000: 间隔定时器模式，在开始计数时不产生定时器中断。 0001: 间隔定时器模式，在开始计数时产生定时器中断。 0100: 捕捉模式，在开始计数时不产生定时器中断。 0101: 捕捉模式，在开始计数时产生定时器中断。 0110: 事件计数器模式，在开始计数时不产生定时器中断。 1000: 单次计数模式，计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式，计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式，在开始计数时不产生定时器中断，计数运行中的开始触发无效。 其他: 禁止设定	0x0

MDm03- MDm00 详细说明见下表:

MDm03	MDm02	MDm01	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MDm00位而变（参照下表）。					

运行模式（MDm03~MDm01位的设定（参照上表））	MDm00	开始计数和中断的设定
• 间隔定时器模式（0、0、0） • 捕捉模式（0、1、0）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
• 事件计数器模式（0、1、1）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
• 单次计数模式注1（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2。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: 在单次计数模式中，不控制开始计数时的中断输出（INTTM00）和TOM0输出。

注2: 如果在运行中产生开始触发（TSM0=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）。

表6-12定时器通道1模式寄存器TMRm1

位	符号	描述	复位值
15:14	CKSm11- CKSm10	通道n运行时钟 (f_{MCK}) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟CKm3 运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 (f_{TCLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	0	必须为0	0
12	CCSm1	通道n计数时钟 (f_{TCLK}) 的选择 0: CKSm10位和CKSm11位指定的运行时钟 (f_{MCK}) 1: TIOSm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	0
11	SPLITm0	通道1的8位定时器/16位定时器的运行选择 0: 用作16位定时器。(用作独立通道功能或则多通道联动运行功能的从属通道) 1: 用作8位定时器。	0
10:8	STSm12- STSm10	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TIm1引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TIm1引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 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) 其他 禁止设定	0x0
7:6	CISm11- CISm10	TIm1引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	0	保留	0x0
3:0	MDm13- MDm10	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	0x0

MDm13- MDm10 详细说明见下表

MDm13	MDm12	MDm11	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MDm10位而变（参照下表）。					

运行模式（MDm13~MDm11位的设定（参照上表））	MDm10	开始计数和中断的设定
- 间隔定时器模式（0、0、0） - 捕捉模式（0、1、0）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
事件计数器模式（0、1、1）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
单次计数模式注1（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2。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：在单次计数模式中，不控制开始计数时的中断输出（INTTM01）和TOM1输出。

注2：如果在运行中产生开始触发（TSm1=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）。

表6-13定时器通道2模式寄存器TMRm2

位	符号	描述	复位值
15:14	CKSm21- CKSm20	通道n运行时钟 (f_{MCK}) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 (f_{TCLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	0	必须为0	0
12	CCS02	通道n计数时钟 (f_{TCLK}) 的选择 0: CKSm20位和CKSm21位指定的运行时钟 (f_{MCK}) 1: TIm2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	
11	MASTER	通道2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则多通道联动运行功能的从属通道。 通道0固定为“0” (因为通道0为最高位的通道, 所以与此位的设定无关, 用作主控通道) 1: 用作多通道联动运行功能的主控通道。 通道0固定为“0” (因为通道0为最高位的通道, 所以与此位的设定无关, 用作主控通道) 只能将通道2设为主控通道 (MASTERmn=1)	0
10:8	STSm22- STSm20	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TIm2引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TIm2引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 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) 其他 禁止设定	0x0
7:6	CISm21- CISm20	TIm2引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	0	保留	0x0
3:0	MDm23- MDm20	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。	0x0

		1001: 单次计数模式，计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式，在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	

MDm23- MDm20 详细说明见下表

MDm23	MDm22	MDm21	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MDm20位而变（参照下表）。					

运行模式（MDm23～MDm21位的设定（参照上表））	MDm20	开始计数和中断的设定
- 间隔定时器模式（0、0、0） - 捕捉模式（0、1、0）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
事件计数器模式（0、1、1）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
单次计数模式注1（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2。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。

注1：在单次计数模式中，不控制开始计数时的中断输出（INTTM02）和TOM2输出。

注2：如果在运行中产生开始触发（TSM2=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）。

表6-14 定时器通道3模式寄存器TMRm3

位	符号	描述	复位值
15:14	CKSm31- CKSm30	通道n运行时钟 (f_{MCK}) 的选择 00: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm0 01: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm2 10: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm1 11: 定时器时钟选择寄存器m (TPSm) 设定的运行时钟 CKm3 运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定CCSmn位来产生采样时钟和计数时钟 (f_{TCLK})。只有通道1和通道3才能选择运行时钟CKm2和CKm3。	0x0
13	0	必须为0	0
12	CCSm3	通道n计数时钟 (f_{TCLK}) 的选择 0: CKSm30位和CKSm31位指定的运行时钟 (f_{MCK}) 1: TIm3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	0
11	SPLITm3	通道1的8位定时器/16位定时器的运行选择 0: 用作16位定时器。(用作独立通道功能或则多通道联动运行功能的从属通道) 1: 用作8位定时器。	0
10:8	STSm32- STSm30	通道n的开始触发和捕捉触发的设定 000: 只有软件触发开始有效 (不选择其他触发源)。 001: 将TIm3引脚输入的有效边沿用于开始触发和捕捉触发。 010: 将TIm3引脚输入的双边沿分别用于开始触发和捕捉触发。 100: 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) 其他 禁止设定	0x0
7:6	CISm31- CISm30	TIm3引脚的有效边沿选择 00: 下降沿 01: 上升沿 10: 双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿 11: 双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿	0x0
5:4	0	保留	0x0
3:0	MDm33- MDm30	通道n运行模式及中断的设定 0000: 间隔定时器模式, 在开始计数时不产生定时器中断。 0001: 间隔定时器模式, 在开始计数时产生定时器中断。 0100: 捕捉模式, 在开始计数时不产生定时器中断。 0101: 捕捉模式, 在开始计数时产生定时器中断。 0110: 事件计数器模式, 在开始计数时不产生定时器中断 1000: 单次计数模式, 计数运行中的开始触发无效。此时不产生中断。 1001: 单次计数模式, 计数运行中的开始触发有效。此时不产生中断。 1100: 捕捉&单次计数模式, 在开始计数时不产生定时器中断, 计数运行中的开始触发无效。 其他: 禁止设定	0x0

MDm33- MDm30 详细说明见下表

MDm33	MDm32	MDm31	通道n运行模式的设定	对应功能	TCR的计数运行
0	0	0	间隔定时器模式	间隔定时器/方波输出/ 分频器功能/PWM输出（主控）	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器/单触发脉冲输出/PWM输出（从属）	递减计数
1	1	0	捕捉&单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因MDm30位而变（参照下表）。					

运行模式（MDm33～MDm31位的设定 （参照上表））	MDm30	开始计数和中断的设定
- 间隔定时器模式（0、0、0） - 捕捉模式（0、1、0）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
	1	在开始计数时产生定时器中断（定时器的输出也发生变化）。
事件计数器模式（0、1、1）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。
单次计数模式^{注1}（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2} 。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。

注1：在单次计数模式中，不控制开始计数时的中断输出（INTTM03）和TOM3输出。

注2：如果在运行中产生开始触发（TSm3=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）。

6.2.7 定时器状态寄存器mn（TSRmn）

TSRmn 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSRmn 寄存器只在捕捉模式（MDmn3~MDmn1=010B）和捕捉&单次计数模式（MDmn3~MDmn1=110B）中有效。有关各运行模式中的 OVF 位的变化和置位/清除条件，请参照表 6-16。

通过 16 位存储器操作指令读 TSRmn 寄存器。

在产生复位信号后，TSRmn 寄存器的值变为“0000H”。

表6-15 定时器状态寄存器mn（TSRmn）的表格

位	符号	描述	复位值
15:1	-	保留	0x0
0	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）。	0

注：m：单元号（m=0~1）n：通道号（n=0~3）

表6-16 各运行模式中的OVF位的变化和置位/清除条件

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

注：m：单元号（m=0~1）n：通道号（n=0~3）

6.2.8 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。

在产生复位信号后，TEm 寄存器的值变为“0000H”。

表6-17 定时器通道允许状态寄存器m (TEm) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	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	0
10	-	保留	0
9	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	0
8:4	-	保留	0x0
3:0	TEm3-TEm0	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	0x0

6.2.9 定时器通道开始寄存器m (TSm)

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

通过 16 位存储器操作指令设定 TSm 寄存器。

在产生复位信号后，TSm 寄存器的值变为“0000H”。

表6-18 定时器通道开始寄存器m (TSm) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	TSHm3	通道3为8位定时器模式时的高8位定时器的运行允许（开始）触发 0: 没有触发 1: 将TEHm3位置“1”，进入计数允许状态。如果在计数允许状态下开始TCRm3寄存器的计数，就进入间隔定时器模式（参照“6.4.2 计数器的开始时序”的表6-27）	0
10	-	保留	0
9	TSHm1	通道1为8位定时器模式时的高8位定时器的运行允许（开始）触发 0: 没有触发 1: 将TEHm1位置“1”，进入计数允许状态。如果在计数允许状态下开始TCRm1寄存器的计数，就进入间隔定时器模式（参照“6.4.2 计数器的开始时序”的表6-27）	0
8:4	-	保留	0x0
3:0	TSm3-TSm0	通道n的运行允许（开始）触发: 0: 没有触发 1: 将TEmn位置“1”，进入计数允许状态。计数允许状态下的TCRmn寄存器的计数开始因各运行模式而不同（参照“6.4.2 计数器的开始时序”的表6-27）。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。	0x0

注1：必须将bit15~12、10、8~4置“0”。

注2：在从不使用TIMn引脚输入的功能切换到使用TIMn引脚输入的功能时，从设定定时器模式寄存器mn (TMRmn) 到将 TSmn (TSHm1、TSHm3) 位置“1”为止，需要2个运行时钟 (f_{MCK}) 的等待。

注3：TSm寄存器的读取值总是“0”。

注4：m：单元号 (m=0~1)

6.2.10 定时器通道停止寄存器m (TTm)

TTm 寄存器是设定各通道计数停止的触发寄存器。

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

通过 16 位存储器操作指令设定 TTm 寄存器。

在产生复位信号后，TTm 寄存器的值变为“0000H”。

表6-19 定时器通道停止寄存器m (TTm) 的表格

位	符号	描述	复位值
15:12	-	保留	0x0
11	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	0
10	-	保留	0
9	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。	0
8:4	-	保留	0x0
3:0	TTm3-TTm0	通道n的运行允许（开始）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。	0x0

注1：必须将bit15~12、10、8~4置“0”。

注2：TTm寄存器的读取值总是“0”。

注3：m：单元号（m=0~1）

6.2.11 定时器输入输出选择寄存器（TIOS0）

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

表6-20 定时器输入选择寄存器0（TIOS0）的表格

位	符号	描述	复位值
7:5	TIS07- TIS05	通道0使用的定时器输入的选择 000: 定时器输入引脚（TI00）的输入信号 其他: 禁止设定	0x0
4	TIS04	通道0使用的定时器输入的选择 0: 通过TISm7~TISm5选择的输入信号 1: ELC的事件输入信号	0
3	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）。	0
2:0	TIS02- TIS00	通道1使用的定时器输入的选择 000: 定时器输入引脚（TI01）的输入信号 001: ELC的事件输入信号 010: 定时器输入引脚（TI01）的输入信号。 011: 定时器输入引脚（TI01）的输入信号 100: 低速内部振荡器时钟（f _{IL} ） 101: 副系统时钟（f _{SUB} ） 其他: 禁止设定	0x0

注1：选择的定时器输入高低电平宽度需要大于等于1/fMCK+10ns。因此在选择f_{IL}作为fCLK时（CKC寄存器的CSS=1），不能将TIS02位置“1”。

注2：m：单元号（m=0~1）

6.2.12 定时器输出允许寄存器m（TOEm）

TOEm 寄存器是设定允许或者禁止各通道定时器输出的寄存器。

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

通过 16 位存储器操作指令设定 TOEm 寄存器。

在产生复位信号后，TOEm 寄存器的值变为“0000H”。

表6-21 定时器输出允许寄存器m（TOEm）的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:0	TOEmn	通道n的定时器输出的允许/禁止： 0: 禁止定时器输出。 定时器的运行不反映到TOmn位，固定输出。能写TOmn位，并且从TOmn引脚输出TOmn位设定的电平。 1: 允许定时器输出 定时器的运行反映到TOmn位，产生输出波形。忽视TOmn位的写操作	0x0

注1：必须将bit15~4置“0”。

注2：m：单元号（m=0~1）n：通道号（n=0~3）

6.2.13 定时器输出寄存器m (TOM)

TOM 寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚 (TOMn) 输出。

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

要将 TI00/TO00、TI01/TO01、TI02/TO02、TI03/TO03、TI10/TO10、TI11/TO11、TI12/TO12、TI13/TO13 引脚用作端口功能时，必须将相应的 TOMn 位置“0”。

通过 16 位存储器操作指令设定 TOM 寄存器。

在产生复位信号后，TOM 寄存器的值变为“0000H”。

表6-22 定时器输出寄存器m (TOM) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:0	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”	0x0

注1: m: 单元号 (m=0~1) n: 通道号 (n=0~3)

6.2.14 定时器输出电平寄存器m (TOLm)

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

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

通过 16 位存储器操作指令设定 TOLm 寄存器。

在产生复位信号后，TOLm 寄存器的值变为“0000H”。

表6-23 定时器输出电平寄存器m (TOLm) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:1	TOLm3- TOLm1	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)	0x0
0	-	保留为0	0

注1: 如果在定时器运行中改写此寄存器的值，会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。

注2: m: 单元号 (m=0~1) n: 通道号 (n=0~3)。

6.2.15 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 的设定。

通过 16 位存储器操作指令设定 TOMm 寄存器。

在产生复位信号后，TOMm 寄存器的值变为“0000H”。

表6-24 定时器输出模式寄存器m (TOMm) 的表格

位	符号	描述	复位值
15:4	-	必须为0	0x0
3:1	TOLm3- TOLm1	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）	0x0
0	-	保留为0	0

注1: 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。

注2: （有关主控通道和从属通道关系的详细内容，请参照“6.3.1 多通道联动运行功能的基本规则”）。

6.2.16 噪声滤波器允许寄存器1 (NFEN1)

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

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

注:详细内容请参照“6.4.1(2) 选择TIMn引脚输入信号的有效边沿的情况（CCSmn=1）”、“6.4.2 计数器的开始时序”和“6.6 定时器输入（TIMn）的控制”。

表6-25 噪声滤波器允许寄存器1 (NFEN1) 的表格

位	符号	描述	复位值
7:4	-	必须为0	0x0
3	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	0
2	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	0
1	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	0
0	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	0

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”。

6.2.17 噪声滤波器允许寄存器2 (NFEN2)

NFEN2寄存器设定噪声滤波器是否用于单元1各通道定时器输入引脚的输入信号。对于需要消除噪声的引脚，必须将对应位置“1”，使噪声滤波器有效。当噪声滤波器有效时，在通过对象通道的运行时钟（f_{MCK}）进行同步后检测2个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟（f_{MCK}）进行同步注。

通过8位存储器操作指令设定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“6.4.1(2) 选择TIMn引脚输入信号的有效边沿的情况（CCSmn=1）”、“6.4.2 计数器的开始时序”和“6.6 定时器输入（TIMn）的控制”。

表6-26 噪声滤波器允许寄存器2 (NFEN2) 的表格

位	符号	描述	复位值
7:4	-	必须为0	0x0
3	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	0
2	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	0
1	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	0
0	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	0

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”。

6.2.18 控制定时器输入/输出引脚端口功能的寄存器

在使用通用定时器单元时，定时器 0 的输入/输出引脚复用到固定端口。详细内容请参照“第 2 章 引脚功能”。

在将定时器 0 的输出引脚复用到某端口时，必须将该端口对应的端口模式控制寄存器（PMCxx）的位、端口模式寄存器（PMxx）的位置“0”。并设置端口复用功能配置寄存器（PS_TO00）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

（例）将 PB0 配置为 TO00 用作定时器输出的情况

将端口模式控制寄存器的 PMCB0 位置“0”。

将端口模式寄存器的 PMB0 位置“0”。

将端口输出复用功能配置寄存器 PS_TO00 置“0x02”。

在将定时器 0 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（PS_TI00）。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

（例）将 PA0 配置为 TI00 用作定时器输入的情况

将端口模式控制寄存器的 PMCA0 位置“0”。

将端口模式寄存器的 PMA0 位置“1”。

将端口输入复用功能配置寄存器 PS_TI00 置“0x00”。

6.3 通用定时器单元的基本规则

6.3.1 多通道联动运行功能的基本规则

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

多通道联动运行功能的基本规则如下所示。

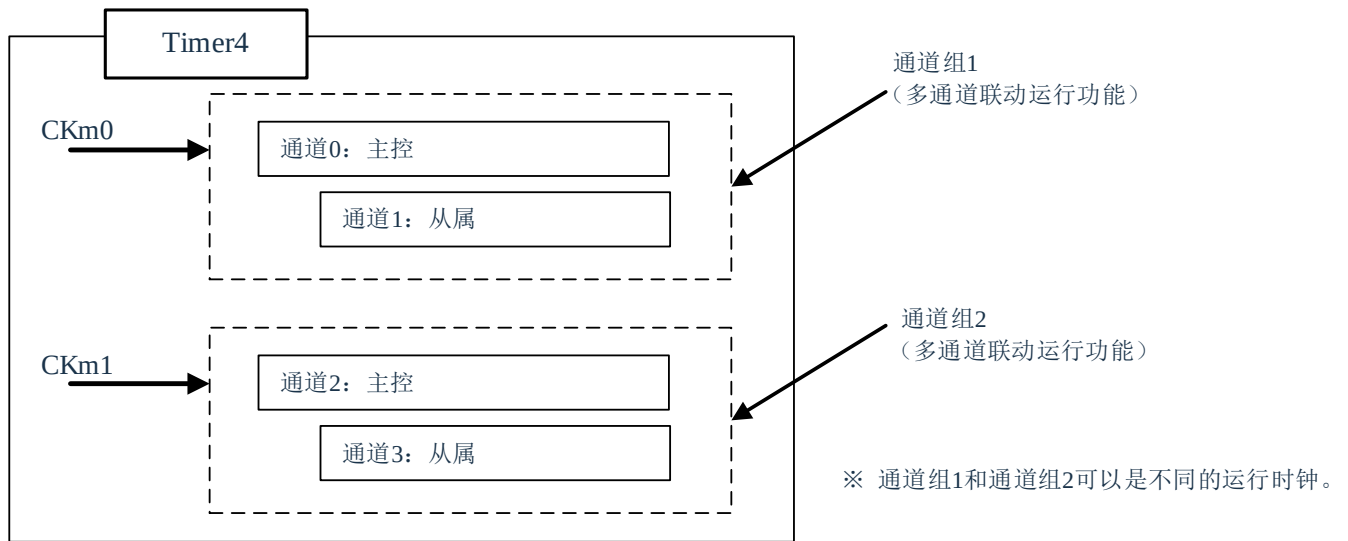
- 1) 只能将偶数通道（通道 0、通道 2）设定为主控通道。
- 2) 通道 0 以外的任何通道都能设定为从属通道。
- 3) 只能将主控通道的低位通道设定为从属通道。
- 4) 例在将通道 0 设定为主控通道时，能将通道 1 开始的通道（通道 1、通道 2、通道 3）设定为从属通道。
- 5) 能对 1 个主控通道设定多个从属通道。
- 6) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。
- 7) 例在将通道 0 和通道 2 设定为主控通道时，能将通道 1 设定为主控通道 0 的从属通道，而不能将通道 3 设定为主控通道 0 的从属通道。
- 8) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的 CKSmn0 位和 CKSmn1 位（定时器模式寄存器 mn（TMRmn）的 bit15 和 bit14）的值需要是相同的设定值。
- 9) 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。
- 10) 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。
- 11) 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。
- 12) 为了同时启动要联动的通道，需要同时设定联动通道的通道开始触发位（TSmn）。
- 13) 只有联动的全部通道或者主控通道才能使用计数运行中的 TSmn 位的设定。不能只使用从属通道的 TSmn 位的设定。
- 14) 为了同时停止要联动的通道，需要同时设定联动通道的通道停止触发位（TTmn）。
- 15) 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。
- 16) 定时器模式寄存器 m0（TMRm0）没有主控位而固定为“0”。但是，因为通道 0 是最高位的通道，所以在联动运行时能将通道 0 用作主控通道。

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

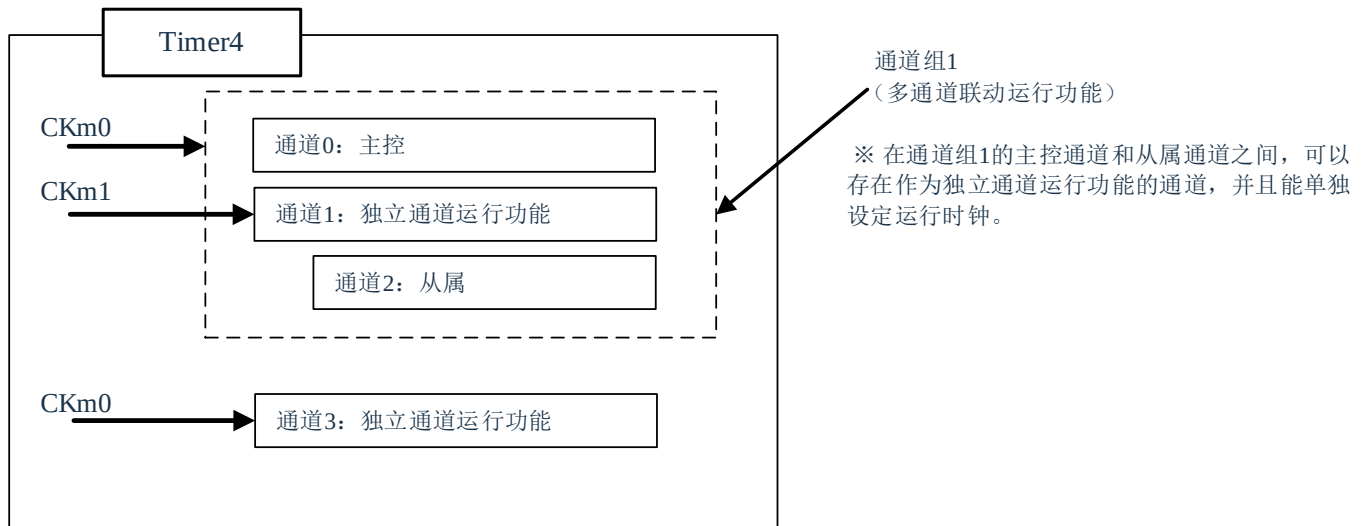
如果设定 2 个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

注：m：单元号（m=0~1） n：通道号（n=0~3）。

例 1



例2



6.3.2 8 位定时器运行功能的基本规则（只限于通道1和通道3）

8 位定时器运行功能是将 16 位定时器的通道用作 2 个 8 位定时器的通道的功能。

只有通道 1 和通道 3 才能使用 8 位定时器运行功能，使用时需要遵守几个规则。

8 位定时器运行功能的基本规则如下所示。

8 位定时器运行功能只适用于通道 1 和通道 3。

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”。

高 8 位定时器能用作间隔定时器功能。

在开始运行时，高 8 位定时器输出 INTTMM1H（中断）（和 MDmn0 位为“1”的运行相同）。

高 8 位定时器的运行时钟的选择取决于低位 TMRmn 寄存器的 CKSmn1 位和 CKSmn0 位的设定。

对于高 8 位定时器，通过操作 TSHm1 位来开始通道的运行，并且通过操作 TTHm1 位来停止通道的运行。能通过 TEHm1 位确认通道的状态。

低 8 位定时器的运行取决于 TMRmn 寄存器的设定，有以下 3 种支持低 8 位定时器运行的功能：

- ◆ 间隔定时器功能
- ◆ 外部事件计数器功能
- ◆ 延迟计数功能

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。

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。

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）。

注：m：单元号（m=0~1） n：通道号（n=1、3）

6.4 计数器的运行

6.4.1 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

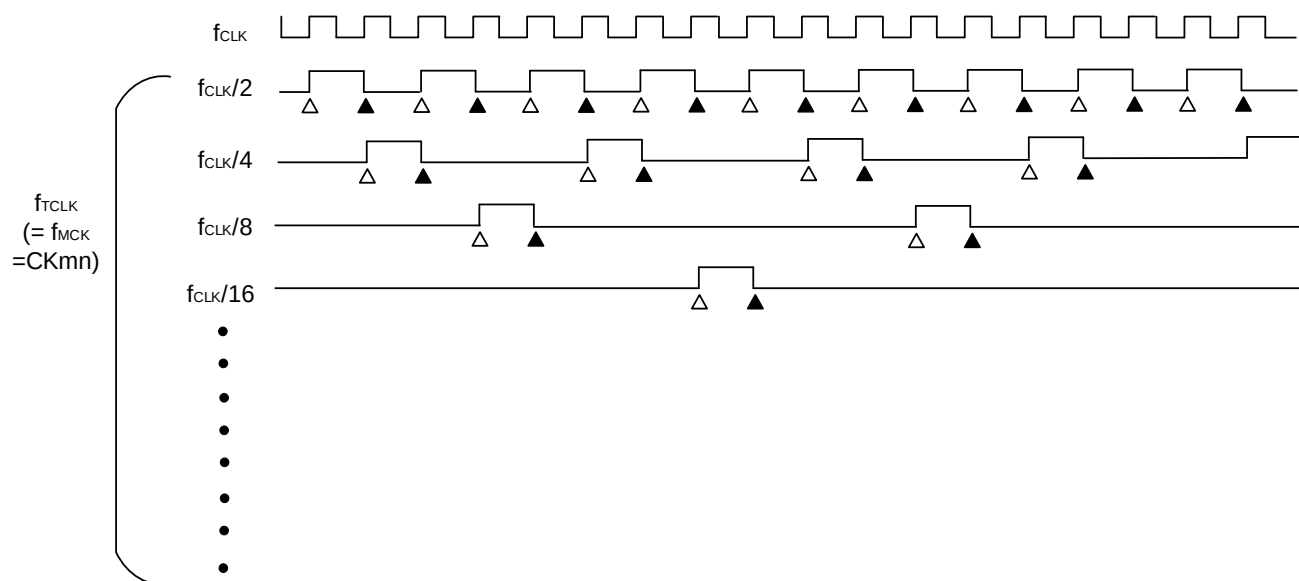
通用定时器单元被设计为与 f_{CLK} 同步运行, 因此计数时钟 (f_{TCLK}) 的时序如下。

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

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

图6-2 f_{CLK} 和计数时钟 (f_{TCLK}) 的时序 (CCSmn=0的情况)



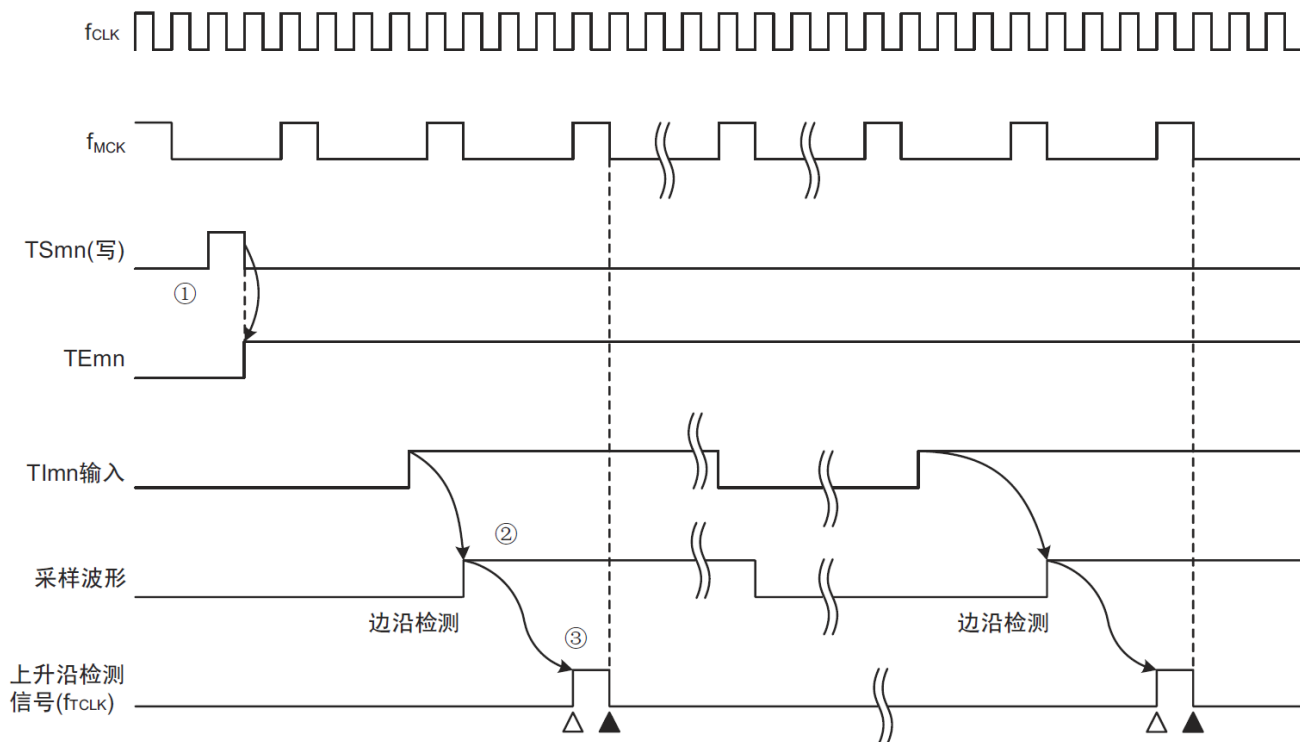
注1: Δ: 计数时钟的上升沿; ▲: 同步、计数器的递增/递减

注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。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} 时后进行计数，出于方便而将其称为“在 TIMn 引脚输入信号的有效边沿进行计数”。

图6-3 计数时钟 (f_{TCLK}) 的时序 (CCSmn=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。

通过 f_{MCK} 对 TIMn 输入的上升沿进行采样。

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）。

注1: Δ: 计数时钟的上升沿; ▲: 同步、计数器的递增/递减

注2: f_{CLK} : CPU/外围硬件的时钟

f_{MCK} : 通道n的运行时钟

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

6.4.2 计数器的开始时序

通过将定时器通道开始寄存器 m (TS m) 的 TS m n 位置位, 定时器计数寄存器 mn (TCR mn) 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 (TCR mn) 开始计数为止的运行如表 6-27 所示。

表6-27 从计数允许状态到定时器计数寄存器 mn (TCR mn) 开始计数为止的运行

定时器的运行模式	将TS m n位置“1”后的运行
间隔定时器模式	从检测到开始触发 (TS m n=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TDR mn 寄存器的值装入TCR mn 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“6.4.3(1) 间隔定时器模式的运行”)。
事件计数器模式	通过给TS m n位写“1”, 将TDR mn 寄存器的值装入TCR mn 寄存器。 如果检测到TI m n的输入边沿, 就通过后续的计数时钟进行递减计数。(参照“6.4.3 (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TCR mn 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“6.4.3(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 (TE mn =0) 的状态下给TS m n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TDR mn 寄存器的值装入TCR mn 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“6.4.3(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 (TE mn =0) 的状态下给TS m n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TCR mn 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“6.4.3(5)捕捉&单次计数模式的运行 (高电平宽度的测量)”)。

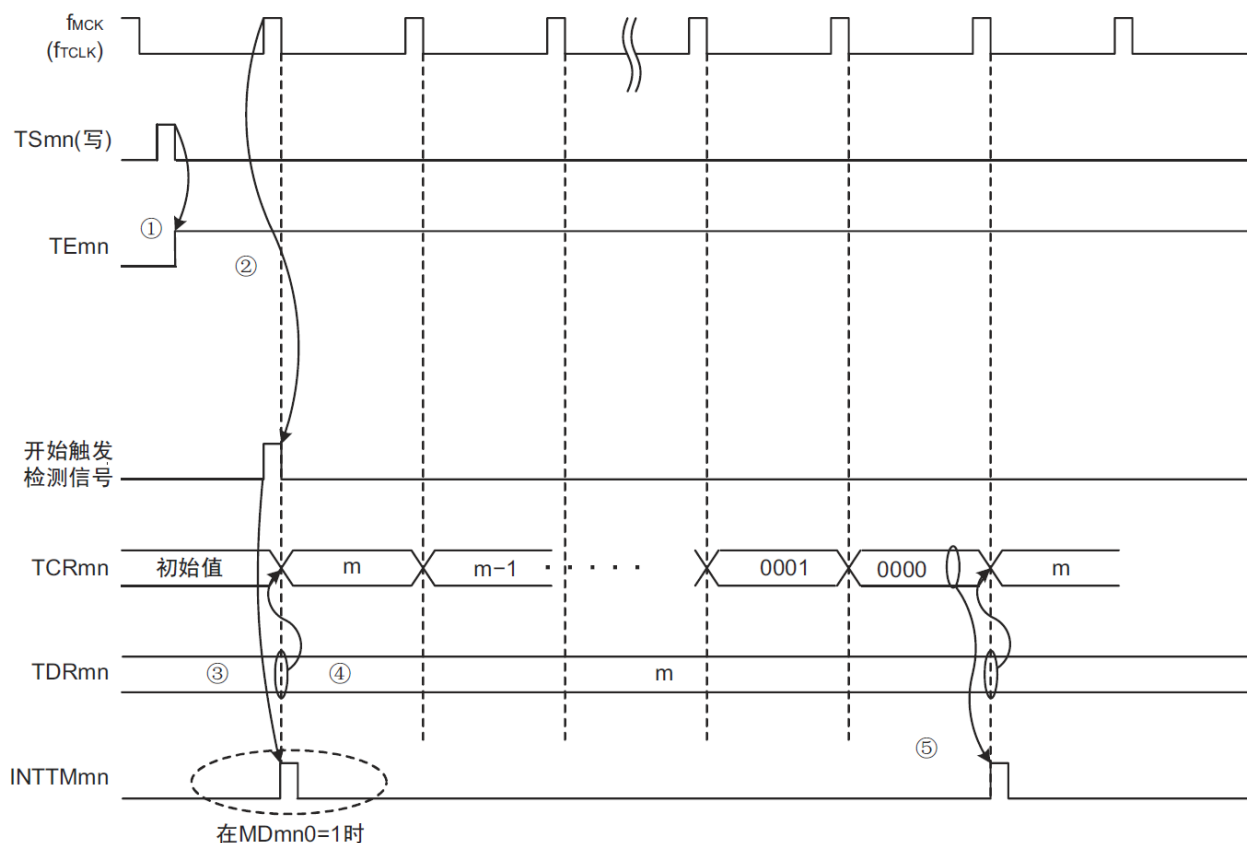
6.4.3 计数器的运行

以下说明各模式的计数器运行。

间隔定时器模式的运行

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

图6-4 运行时序（间隔定时器模式）



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

注2: f_{CLK}、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。

事件计数器模式的运行

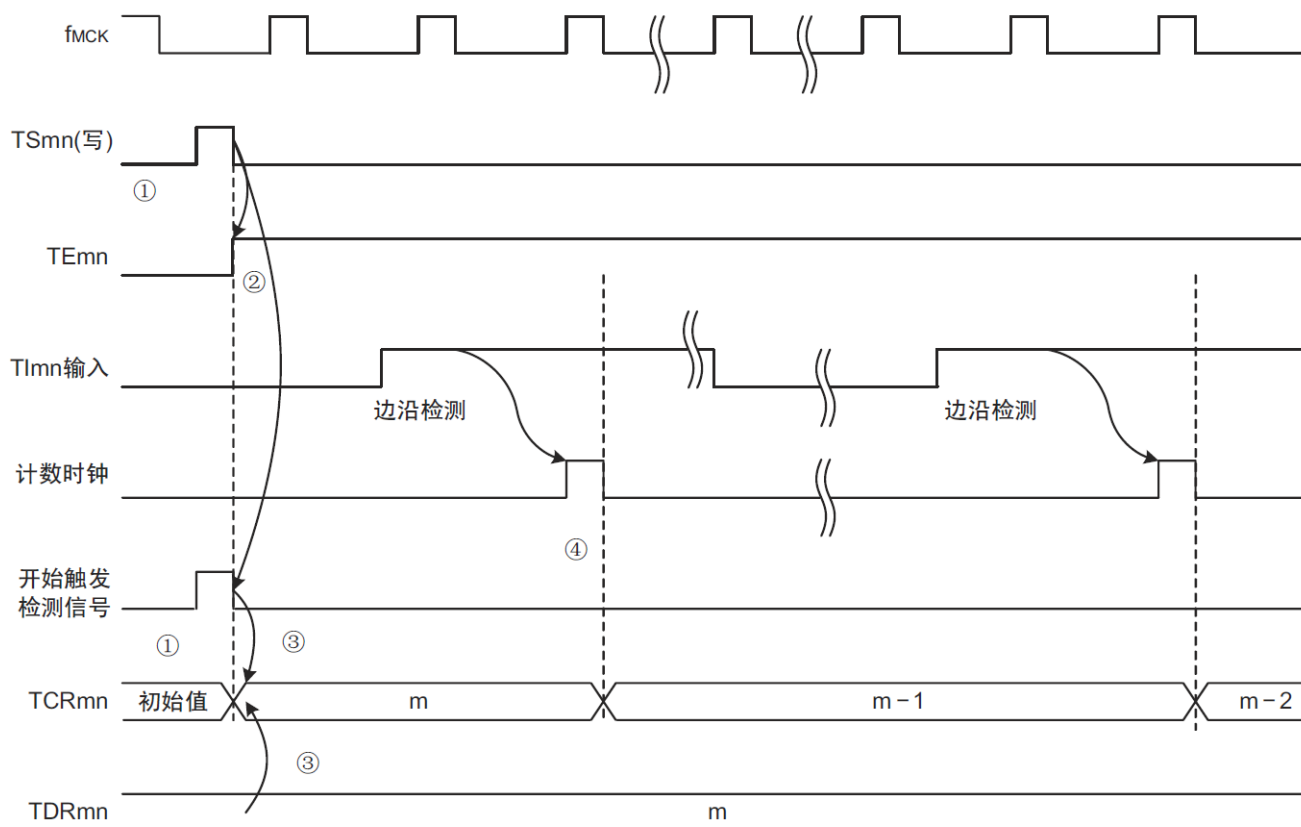
在运行停止状态（TE_{mn}=0）的期间，定时器计数寄存器 mn（TCR_{mn}）保持初始值。

通过给 TS_{mn} 位写“1”，进入运行允许状态（TE_{mn}=1）。

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。

此后，在 TI_{mn} 输入的有效边沿，通过计数时钟对 TCR_{mn} 寄存器的值进行递减计数。

图6-5 运行时序（事件计数器模式）



注：1个周期的误差是因为TI_{mn}输入与计数时钟（f_{MCK}）不同步。

捕捉模式的运行（输入脉冲的间隔测量）

通过给 $TSmn$ 位写“1”，进入运行允许状态（ $TEmn=1$ ）。

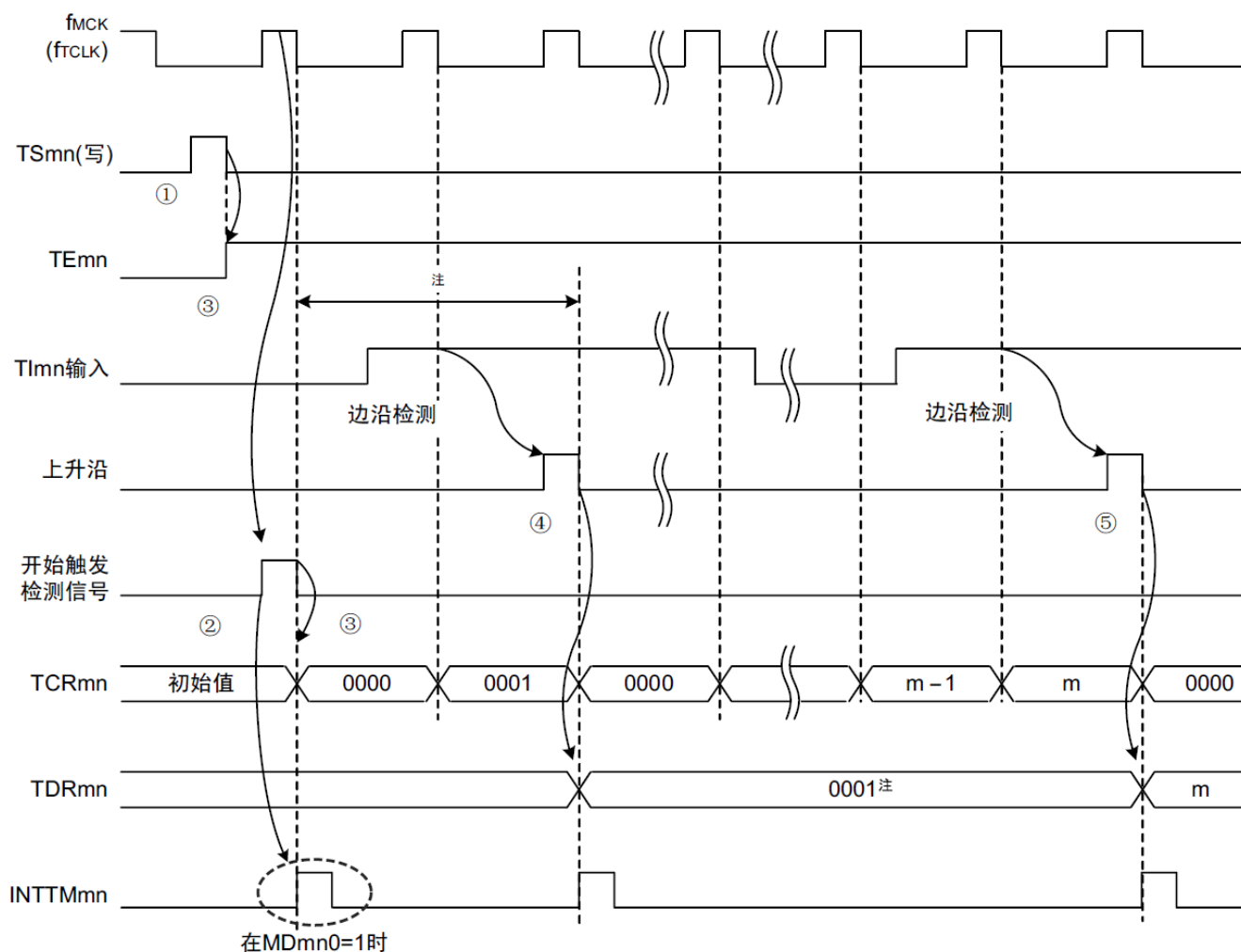
定时器计数寄存器 mn （ $TCRmn$ ）保持初始值，直到产生计数时钟为止。

通过允许运行后的第 1 个计数时钟（ f_{MCK} ）产生开始触发信号。然后，将“0000H”装入 $TCRmn$ 寄存器并且以捕捉模式开始计数（当 $MDmn0$ 位为“1”时，通过开始触发信号产生 $INTTMmn$ ）。

如果检测到 $TIImn$ 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。

如果检测到下一个 $TIImn$ 输入的有效边沿，就将 $TCRmn$ 寄存器的值捕捉到 $TDRmn$ 寄存器，并且产生 $INTTMmn$ 中断。

图6-6 运行时序（捕捉模式：输入脉冲的间隔测量）



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

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

注3：1个周期的误差是因为 $TIImn$ 输入与计数时钟（ f_{MCK} ）不同步。

单次计数模式的运行

通过给 TSmn 位写“1”，进入运行允许状态 (TEmn=1)。

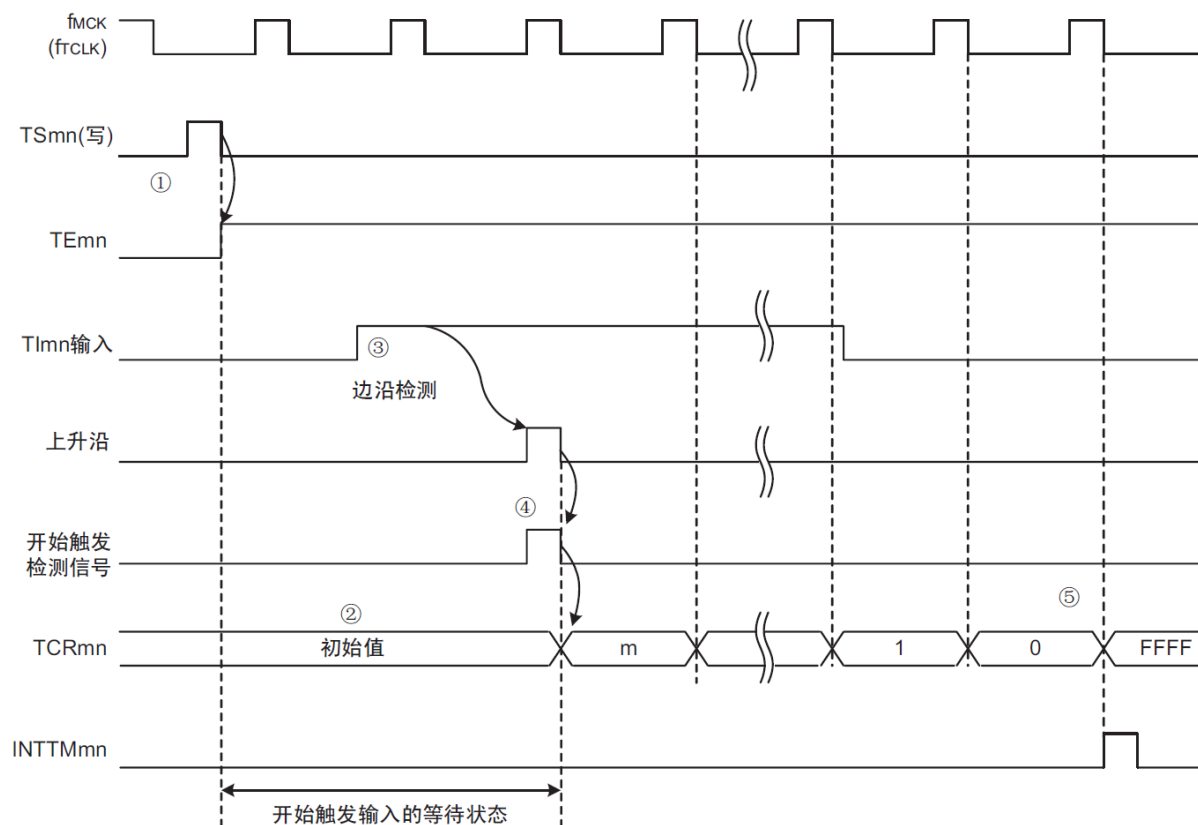
定时器计数寄存器 mn (TCRmn) 保持初始值，直到产生开始触发信号为止。

检测 TImn 输入的上升沿。

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。

当 TCRmn 寄存器递减计数到“0000H”时，产生 INTTMmn 中断，并且 TCRmn 寄存器的值变为“FFFFH”，停止计数。

图6-7 运行时序（单次计数模式）



注：1个周期的误差是因为TImn输入与计数时钟（f_{MCK}）不同步。

捕捉&单次计数模式的运行（高电平宽度的测量）

通过给定时器通道开始寄存器 m (TS m) 的 TS m n 位写“1”，进入运行允许状态 (TE m n=1)。

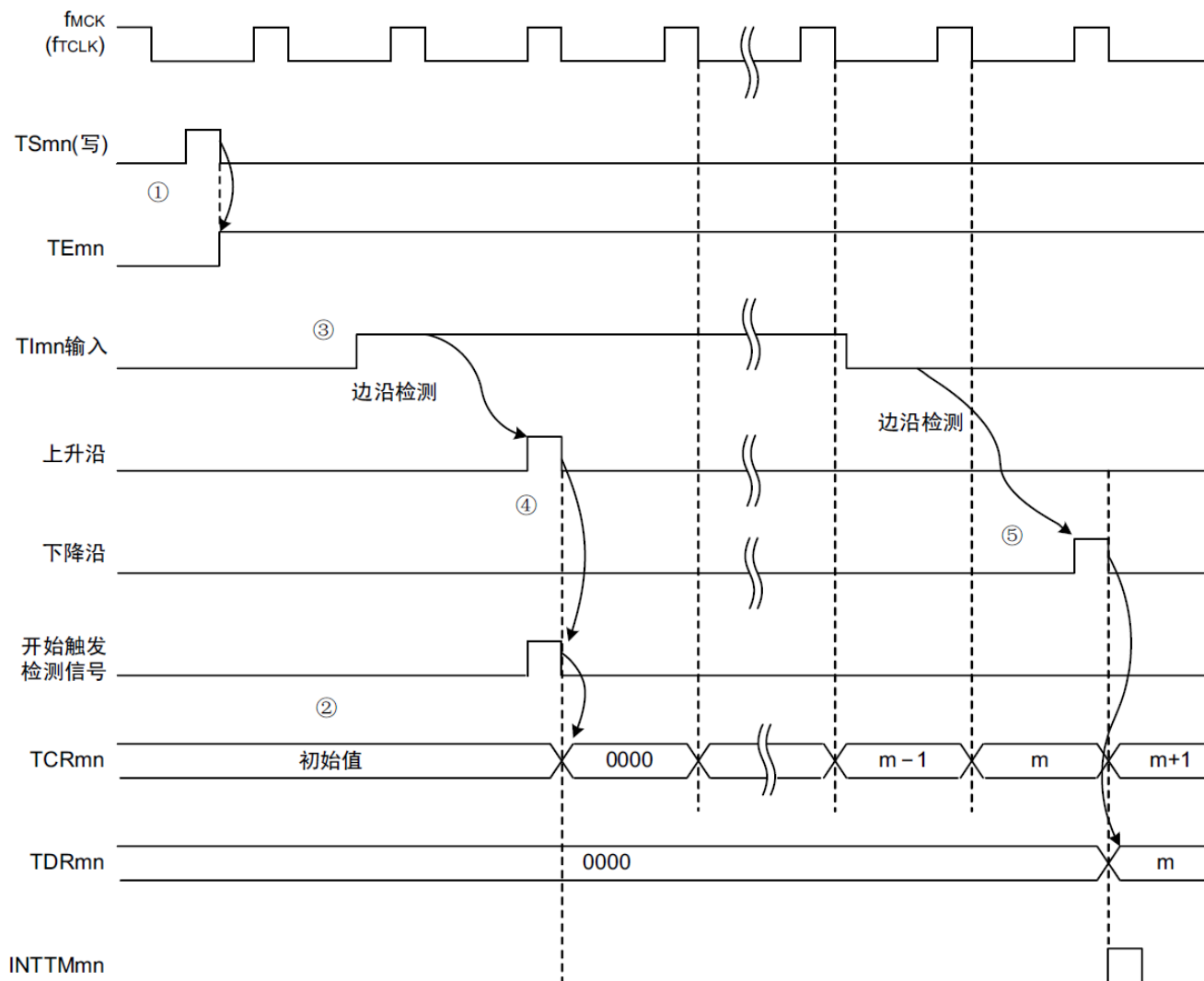
定时器计数寄存器 mn (TCR mn) 保持初始值，直到产生开始触发信号为止。

检测 TI mn 输入的上升沿。

在产生开始触发信号后将“0000H”装入 TCR mn 寄存器，并且开始计数。

如果检测到 TI mn 输入的下降沿，就将 TCR mn 寄存器的值捕捉到 TDR mn 寄存器，并且产生 INTT m n 中断。

图6-8 运行时序（捕捉 & 单次计数模式：高电平宽度的测量）

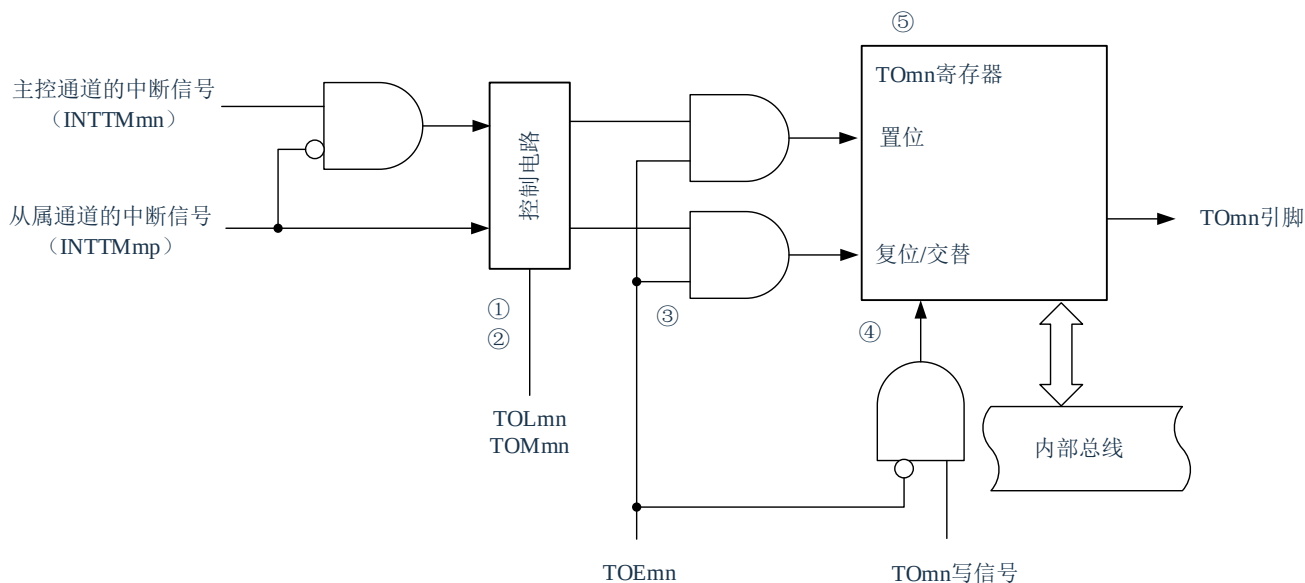


注：1个周期的误差是因为TI mn 输入与计数时钟 (f_{MCK}) 不同步。

6.5 通道输出（TOmn引脚）的控制

6.5.1 TOmn引脚输出电路的结构

图6-9 输出电路的结构



当 TOMmn 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 m（TOLm）的设定值，只将 INTTMmp（从属通道定时器中断）传给定时器输出寄存器 m（TOM）。

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）。

在允许定时器输出（TOEmn=1）的状态下，将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。TOM 寄存器的写操作（TOmn 写信号）无效。

当 TOEmn 位为“1”时，除了中断信号以外，不改变 TOmn 引脚的输出。

要对 TOmn 引脚的输出电平进行初始化时，需要在设定为禁止定时器输出（TOEmn=0）后给 TOM 寄存器写值。

在禁止定时器输出（TOEmn=0）的状态下，对象通道的 TOmn 位的写操作（TOmn 写信号）有效。当定时器输出为禁止状态（TOEmn=0）时，不将 INTTMmn（主控通道定时器中断）和 INTTMmp（从属通道定时器中断）传给 TOM 寄存器。

能随时读 TOM 寄存器，并且能确认 TOmn 引脚的输出电平。

注：m：单元号（m=0~1）n：通道号 n=0~3（主控通道：n=0、2）

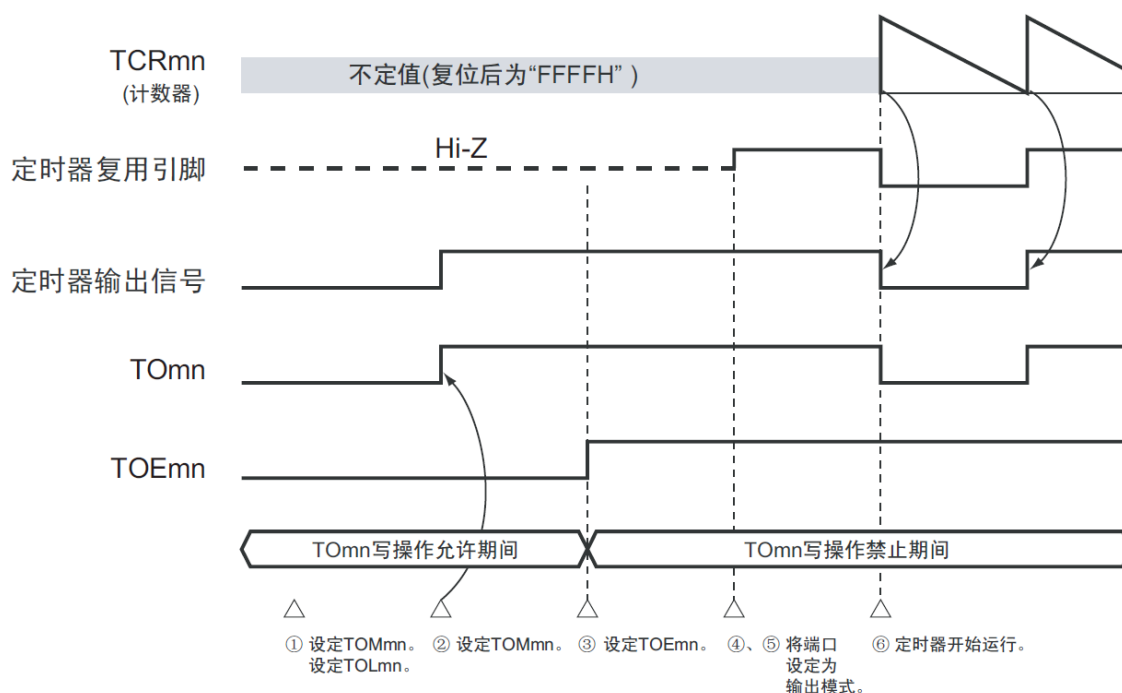
p：从属通道号

n=0：p=1、2、3 n=2：p=3

6.5.2 TOmn引脚的输出设定

从TOmn输出引脚的初始设定到定时器开始运行的步骤和状态变化如下所示。

图6-10 从设定定时器的输出到开始运行的状态变化



设定定时器输出的运行模式。

TOMn 位 (0: 主控通道输出模式、1: 从属通道输出模式)

TOLm 位 (0: 正逻辑输出、1: 负逻辑输出)

通过设定定时器输出寄存器 m (TOM)，将定时器输出信号设定为初始状态。

给 TOEmn 位写 "1"，允许定时器输出 (禁止写 TOM 寄存器)。

通过端口模式控制寄存器 (PMCxx) 将端口设定为数字输入/输出。

将端口的输入/输出设定为输出。

允许定时器运行 (TSmn=1)。

注: m: 单元号 (m=0~1) n: 通道号 n=0~3

6.5.3 通道输出运行的注意事项

(1) 有关定时器运行中的 TO_m 、 $TOEm$ 、 TOL_m 、 TOM_m 寄存器的设定值变更

定时器的运行（定时器计数寄存器 mn （ TCR_{mn} ）和定时器数据寄存器 mn （ TDR_{mn} ）的运行）和 TO_{mn} 输出电路相互独立。因此，定时器输出寄存器 m （ TO_m ）、定时器输出允许寄存器 m （ $TOEm$ ）和定时器输出电平寄存器 m （ TOL_m ）的设定值的变更不会影响定时器的运行，能在定时器运行中更改设定值。但是，为了在各定时器的运行中从 TO_{mn} 引脚输出期待的波形，必须设定为 6.7 和 6.8 所示的各运行的寄存器设定内容例子的值。

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

注：m：单元号（ $m=0\sim1$ ） n：通道号 $n=0\sim3$

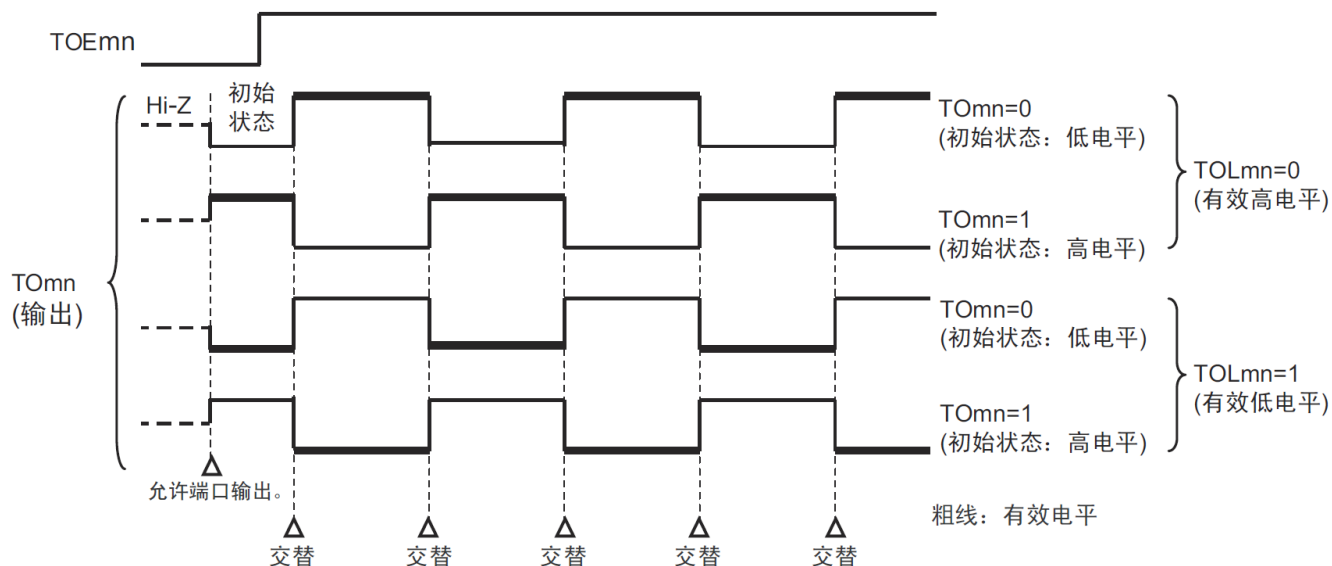
(2) 有关 TO_{mn} 引脚的初始电平和定时器开始运行后的输出电平

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

(a) 在主导通道输出模式（ $TOM_{mn}=0$ ）中开始运行的情况

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

图6-11 交替输出时（ $TOM_{mn}=0$ ）的 TO_{mn} 引脚输出状态



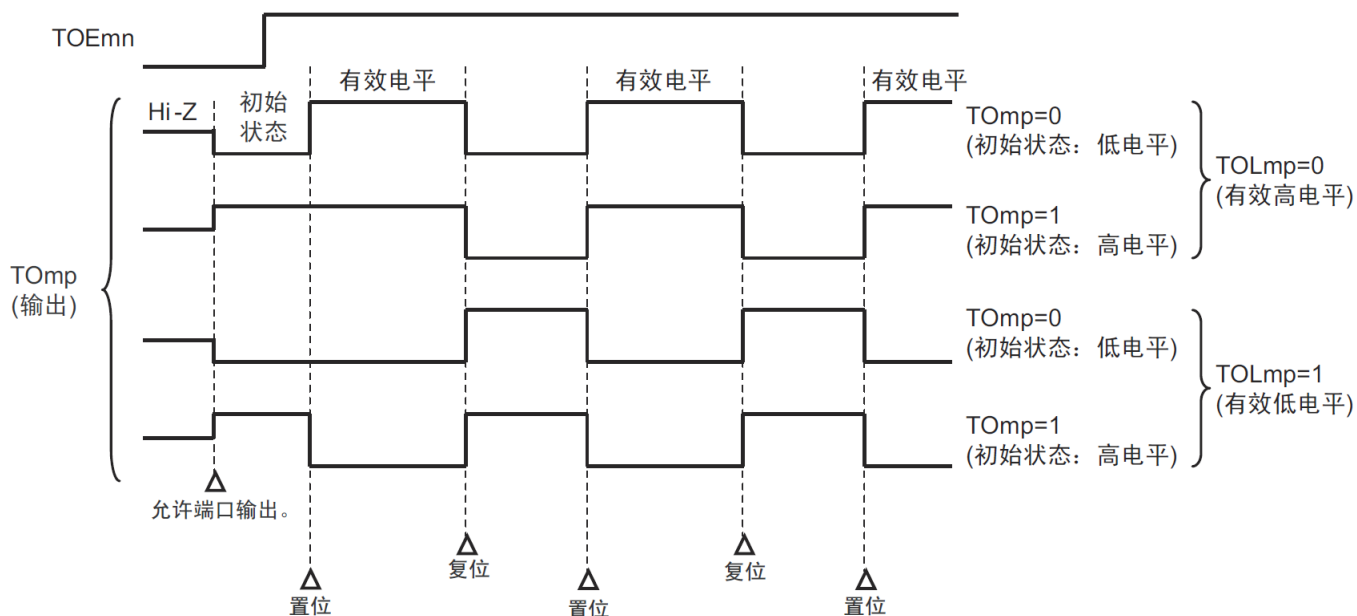
注1：交替：反相 TO_{mn} 引脚的输出状态。

注2：m：单元号（ $m=0\sim1$ ） n：通道号（ $n=0\sim3$ ）

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

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

图6-12 PWM输出时 (TOMmn=1) 的TOmn引脚输出状态



注1: 置位: TOmp引脚的输出信号从无效电平变为有效电平。

注2: 复位: TOmp引脚的输出信号从有效电平变为无效电平。

注3: m: 单元号 (m=0~1) n: 通道号 (p=1~3)

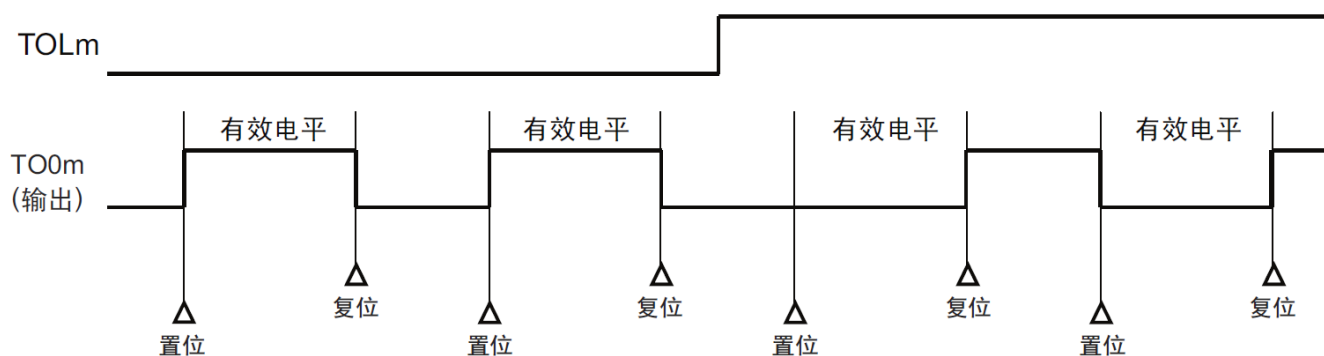
(3) 有关从属通道输出模式 (TOMmn=1) 的 TOMn 引脚变化

在定时器运行中更改定时器输出电平寄存器 m (TOLm) 的设定的情况

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

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

图6-13 在定时器运行中更改TOLm寄存器的内容时的运行



注1: 置位: TOmp引脚的输出信号从无效电平变为有效电平。

注2: 复位: TOmp引脚的输出信号从有效电平变为无效电平。

注3: m: 单元号 (m=0~1) n: 通道号 (n=1~3)

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(INTTMnn)时的TOMn引脚/TOMn位的置位时序延迟1个计数时钟。

当置位条件和复位条件同时产生时, 优先复位条件。

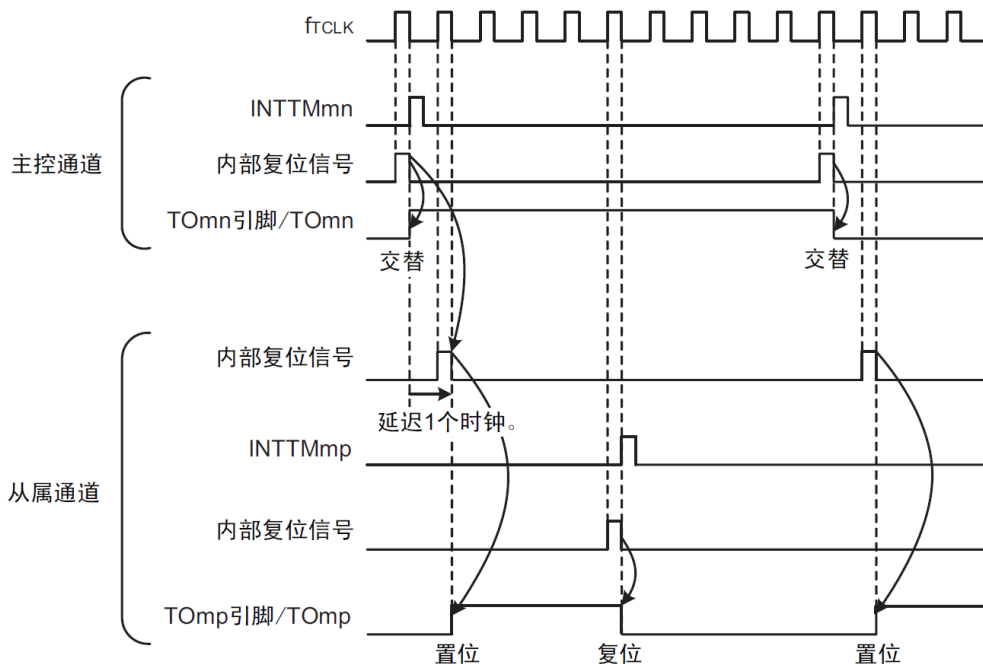
按照以下方法设定主控/从属通道时的置位/复位运行状态如图 6-14 所示。

主控通道: TOEmn=1、TOMmn=0、TOLmn=0

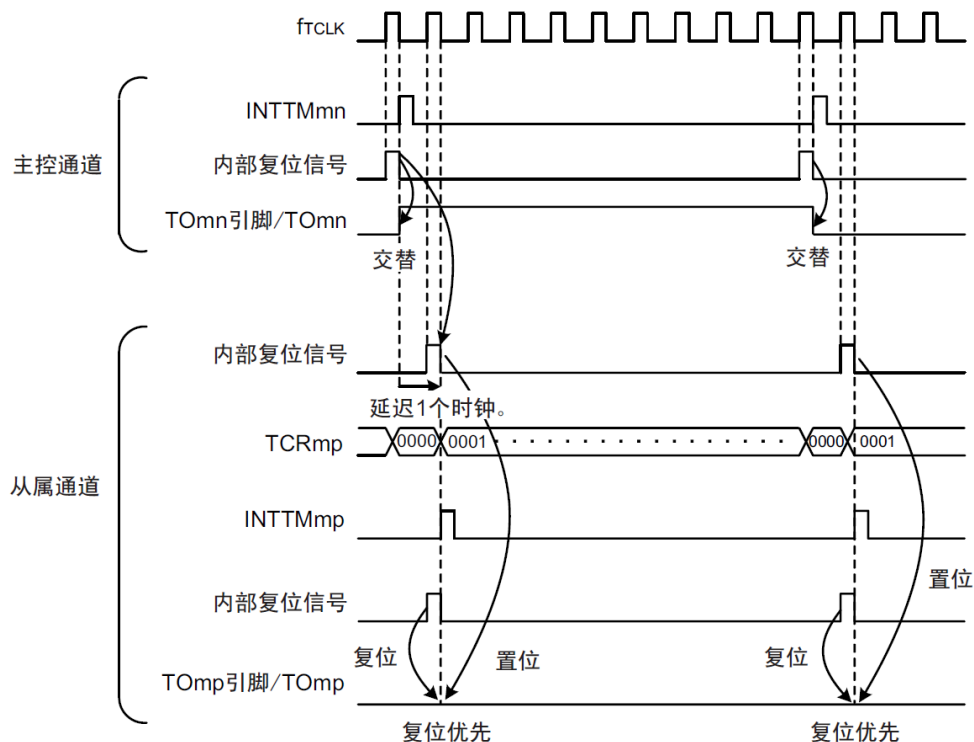
从属通道: TOEmp=1、TOMmp=1、TOLmp=0

图6-14 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



注1: 内部复位信号: TOn引脚的复位/交替信号

注2: 内部置位信号: TOn引脚的置位信号

注3: m: 单元号 (m=0~1) n: 通道号 n=0~3 (主控通道: n=0、2)

p: 从属通道号 n=0: p=1、2、3 n=2: p=3

6.5.4 TOmn位的一次性操作

和定时器通道开始寄存器 m (TS m) 相同, 定时器输出寄存器 m (TO m) 有全部通道的设定位 (TO mn), 因此能一次性地操作全部通道的 TO mn 位。

表6-28 TO $0n$ 位的一次性操作例子

写前

TO0	0	0	0	0	0	0	0	0	0	0	0	TO03	TO02	TO01	TO00
												1	0	1	0

TOE0

	0	0	0	0	0	0	0	0	0	0	0	TOE03	TOE02	TOE01	TOE00
												0	0	0	1

要写的数据

	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

写后

写后

TO03

TO02

TO01

TO00

0

0

0

0

0

0

0

0

0

0

0

0

0

1

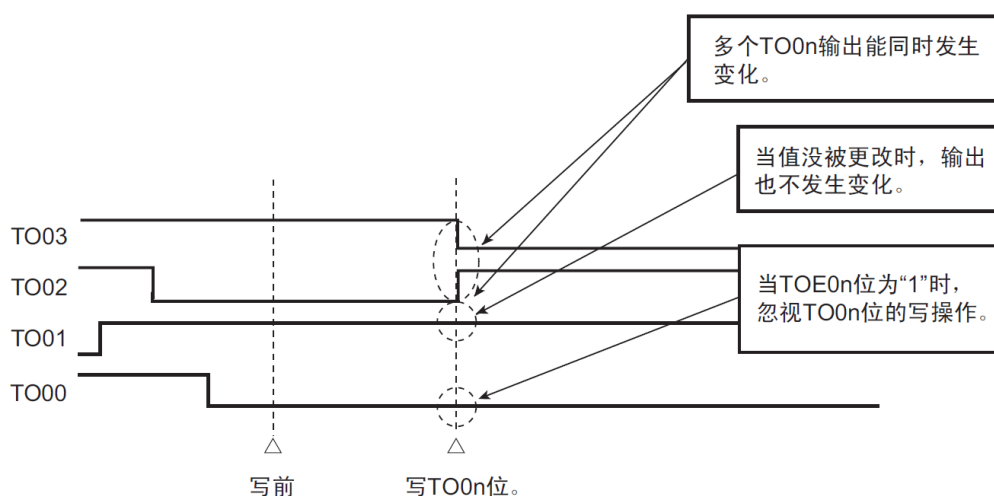
1

0

只能写 TOE mn 位为“0”的 TO mn 位, 忽视 TOE mn 位为“1”的 TO mn 位的写操作。

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

图6-15 一次性操作TO $0n$ 位时的TO $0n$ 引脚状态



注: m : 单元号 ($m=0\sim1$) n : 通道号 ($n=0\sim3$)

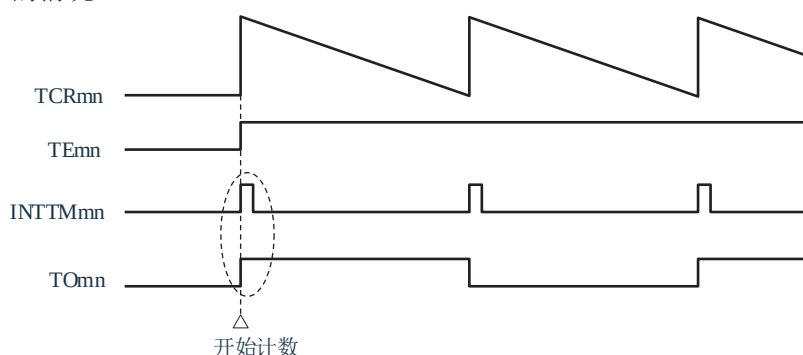
6.5.5 有关开始计数时的定时器中断和TOmn引脚输出

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

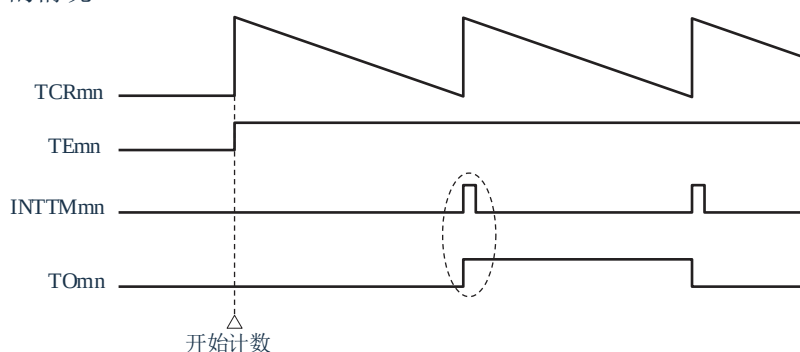
当 MDmn0 位为“1”时，能通过产生定时器中断（INTTMmn）得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和 TOmn 输出。设定为间隔定时器模式（TOEmn=1、TOMmn=0）时的运行例子如下所示。

图6-16 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。

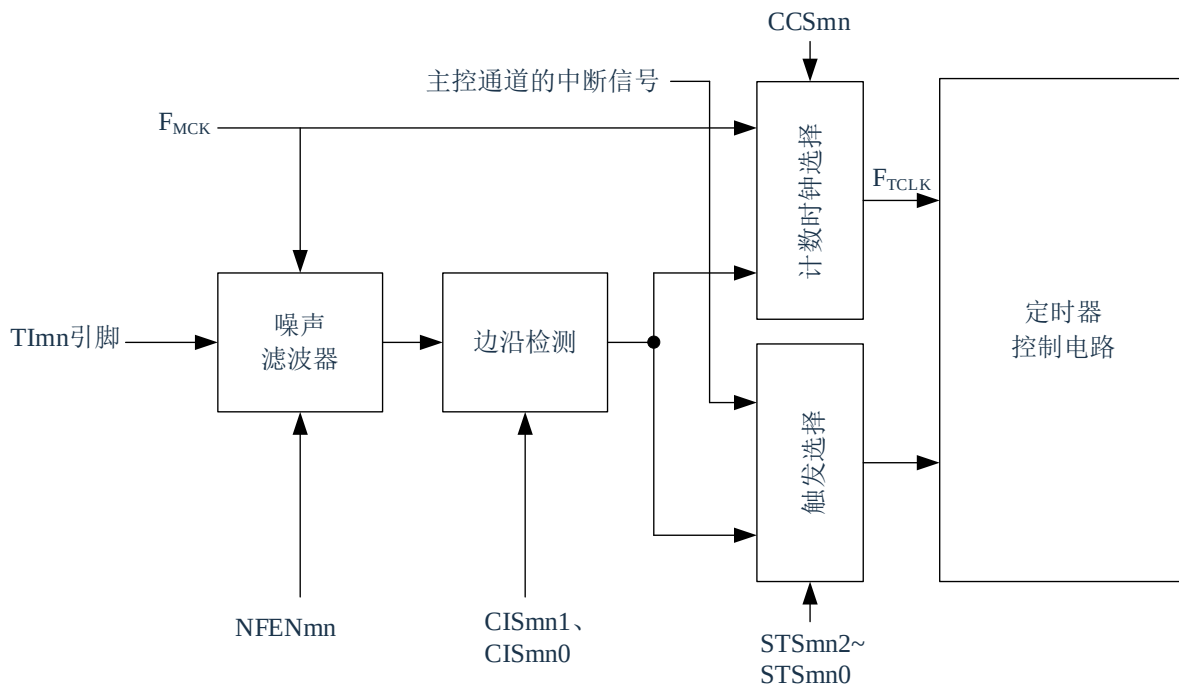
注：m：单元号（m=0~1） n：通道号（n=0~3）

6.6 定时器输入（TImn）的控制

6.6.1 TImn引脚输入电路的结构

定时器输入引脚的信号通过边沿检测电路输入到定时器控制电路。输入电路的结构如下所示。

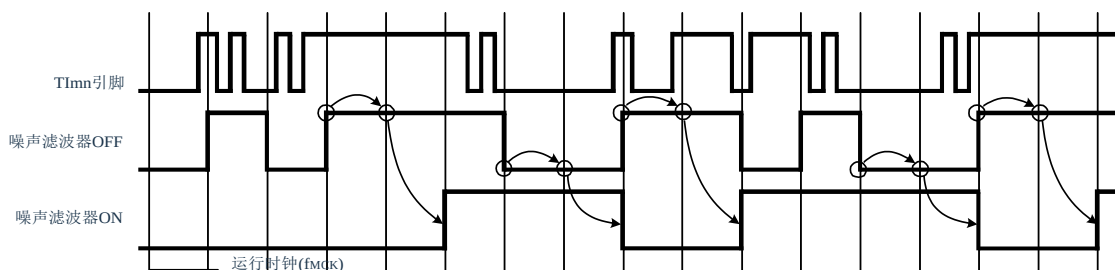
图6-17 输入电路的结构



6.6.2 噪声滤波器

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

图6-18 TImn输入引脚在噪声滤波器ON或者OFF情况下的采样波形



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

6.6.3 操作通道输入时的注意事项

从设定为使用定时器输入引脚到设定定时器输入引脚对应的通道运行允许触发，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。

6.7 通用定时器单元的独立通道运行功能

6.7.1 作为间隔定时器/方波输出的运行

(1) 间隔定时器

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

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

(2) 作为方波输出的运行

TOmn在产生INTTMmn的同时进行交替输出，输出占空比为50%的方波。

TOmn输出方波的周期和频率能用以下计算式进行计算：

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

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

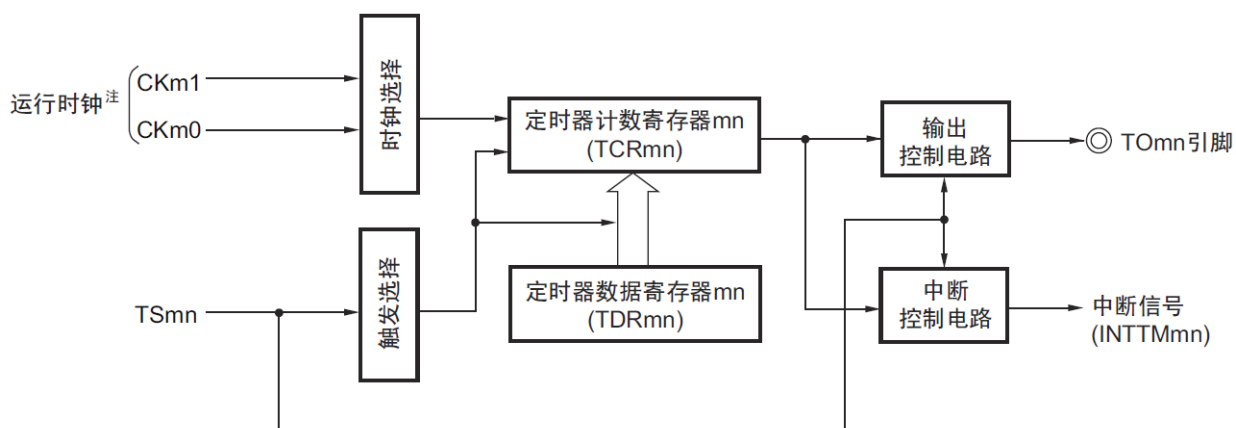
在间隔定时器模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

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

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

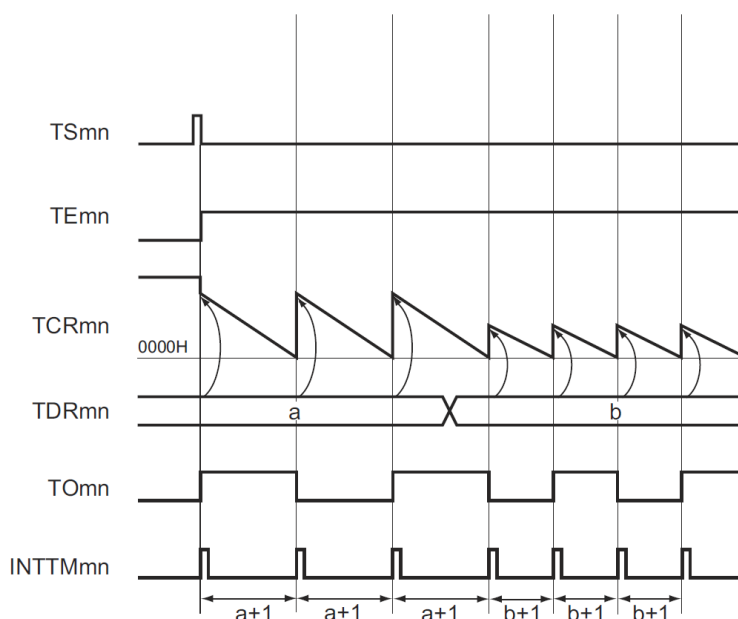
能随时改写TDRmn寄存器，改写的TDRmn寄存器的值从下一个周期开始有效。

图6-19 作为间隔定时器 / 方波输出运行的基本时序例子 (MDmn0=1)



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

图6-20 作为间隔定时器 / 方波输出运行的基本时序例子 (MDmn0=1)



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

注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

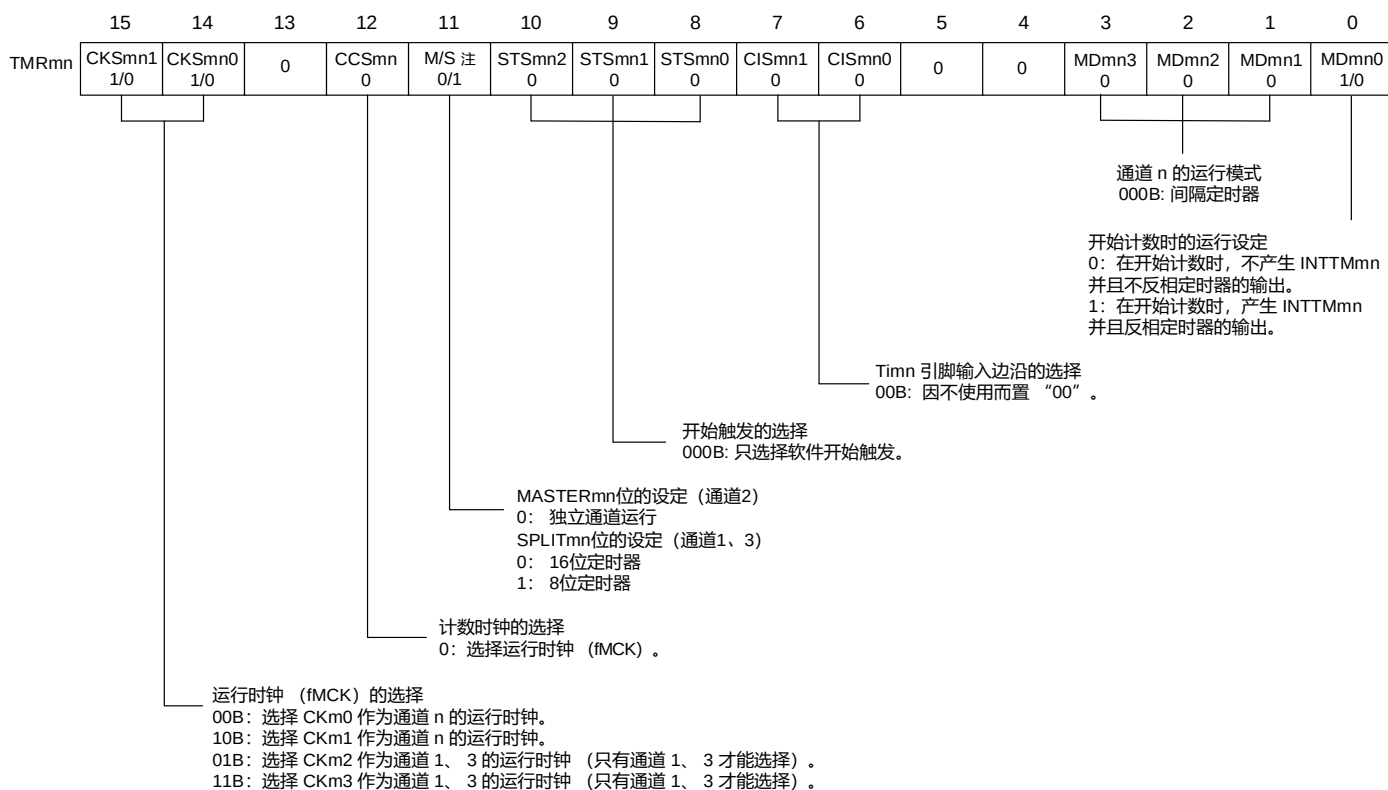
TCRmn：定时器计数寄存器 mn (TCRmn)

TDRmn：定时器数据寄存器 mn (TDRmn)

T0mn：T0mn 引脚输出信号

图6-21 间隔定时器 / 方波输出时的寄存器设定内容例子

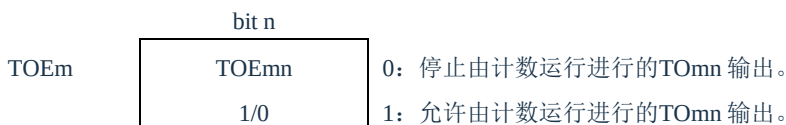
(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器m (TOM)



(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注：TMRm2：MASTERmn 位

TMRm1、TMRm3：SPLITmn位

TMRm0：固定为“0”。

m：单元号（m= 0~1） n：通道号（n=0 ~3）

表6-29 间隔定时器 / 方波输出功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 m 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器0 (PER0) 的TM4mEN 位置 “1” 。 →	定时器单元m 的输入时钟处于提供状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器m (TPSm) 。 确定CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器mn (TMRmn) (确定通道的运行模式)。 给定时器数据寄存器mn (TDRmn) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	使用TOMn 输出的情况: 将定时器输出模式寄存器m (TOMm) 的TOMmn位置 “0” (主控通道输出模式)。 将TOLmn 位置 “0” 。 →	TOMn 引脚处于Hi-Z 输出状态。
	设定TOMn 位, 确定TOMn 输出的初始电平。 →	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。
重新 开始 运行	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。
	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。 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在的值后将TOMn 位置 “0” 。 →	通过端口功能保持TOMn 引脚的输出电平。
	不需要保持TOMn 引脚输出电平的情况: 不需要设定。 将PER0 寄存器的TM4mEN 位置 “0” 。 →	定时器单元m 的输入时钟处于停止提供状态。 对全部电路和各通道的SFR 进行初始化。 (TOMn位变为 “0” 并且TOMn引脚变为端口功能)

注：m：单元号（m= 0~1） n：通道号（n=0 ~3）

6.7.2 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。

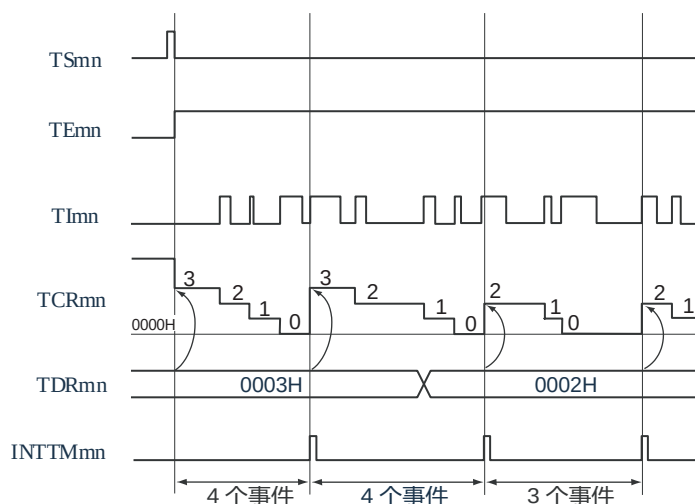
TCRmn 寄存器在检测到 TIMn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入TDRmn 寄存器的值并且输出INTTMmn。

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6-22 作为外部事件计数器运行的基本时序例子



注：TSmn：定时器通道开始寄存器 m（TSm）的 bit n

TEmn：定时器通道允许状态寄存器 m（TEm）的 bit n

TImn：TIMn 引脚输入信号

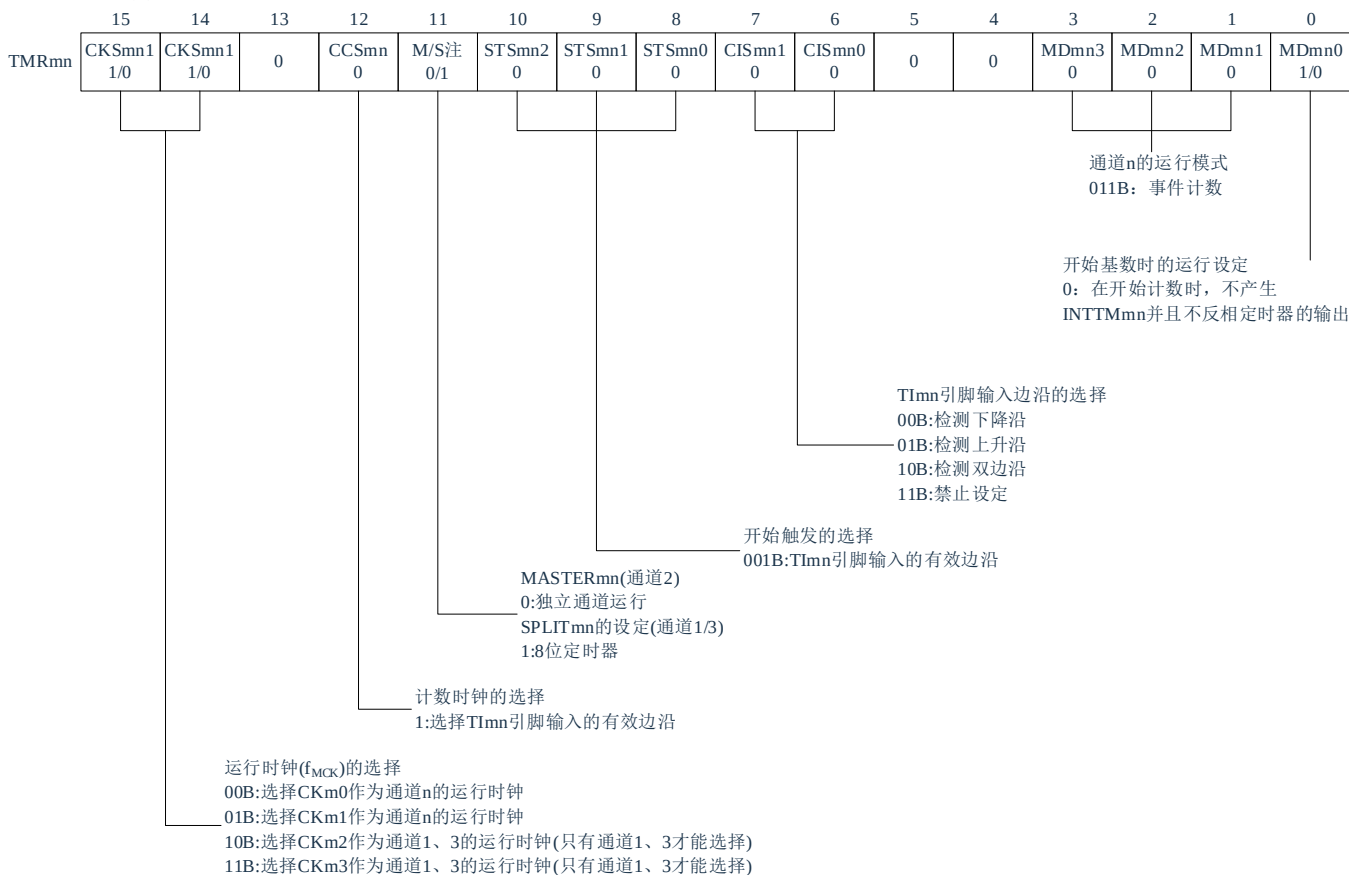
TCRmn：定时器计数寄存器mn（TCRmn）

TDRmn：定时器数据寄存器mn（TDRmn）

m：单元号（m= 0~1） n：通道号（n=0 ~3）

图6-23 外部事件计数器模式时的寄存器设定内容例子

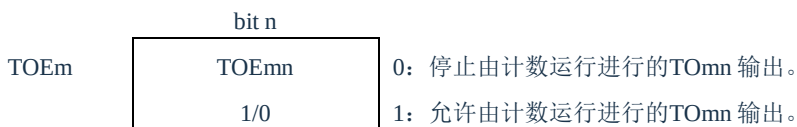
1. (a)定时器模式寄存器mn (TMRmn)



2. (b)定时器输出寄存器m (TOMm)



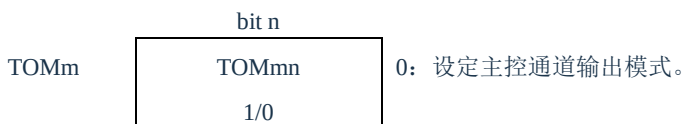
3. (c)定时器输出允许寄存器m (TOEm)



4. (d)定时器输出电平寄存器m (TOLm)



5. (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”。

m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

表6-30 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	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

注: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

6.7.3 作为分频器的运行

能对TI00 引脚输入的时钟进行分频并且用作TO00 引脚输出的分频器。

TO00 输出的分频时钟频率能用以下计算式进行计算：

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

在间隔定时器模式中，定时器计数寄存器 00（TCR00）用作递减计数器。

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

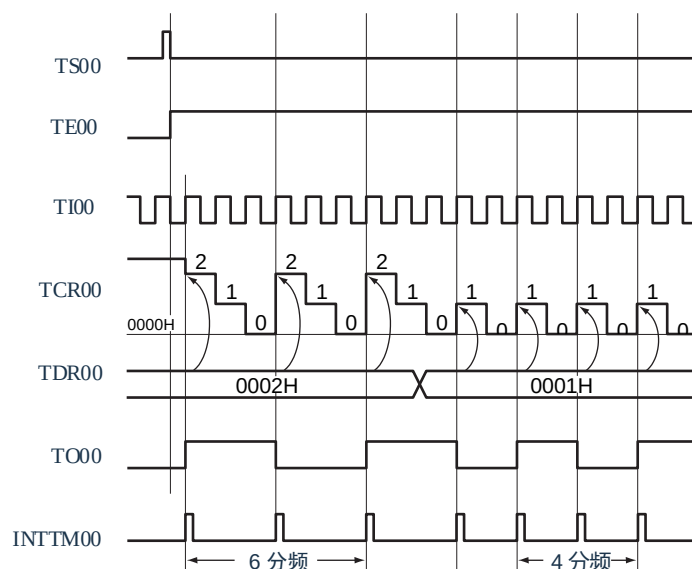
如果选择 TI0 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

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

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

图6-24 作为分频器运行的基本时序例子（MD000=1）



注：TS00：定时器通道开始寄存器0（TS0）的bit0

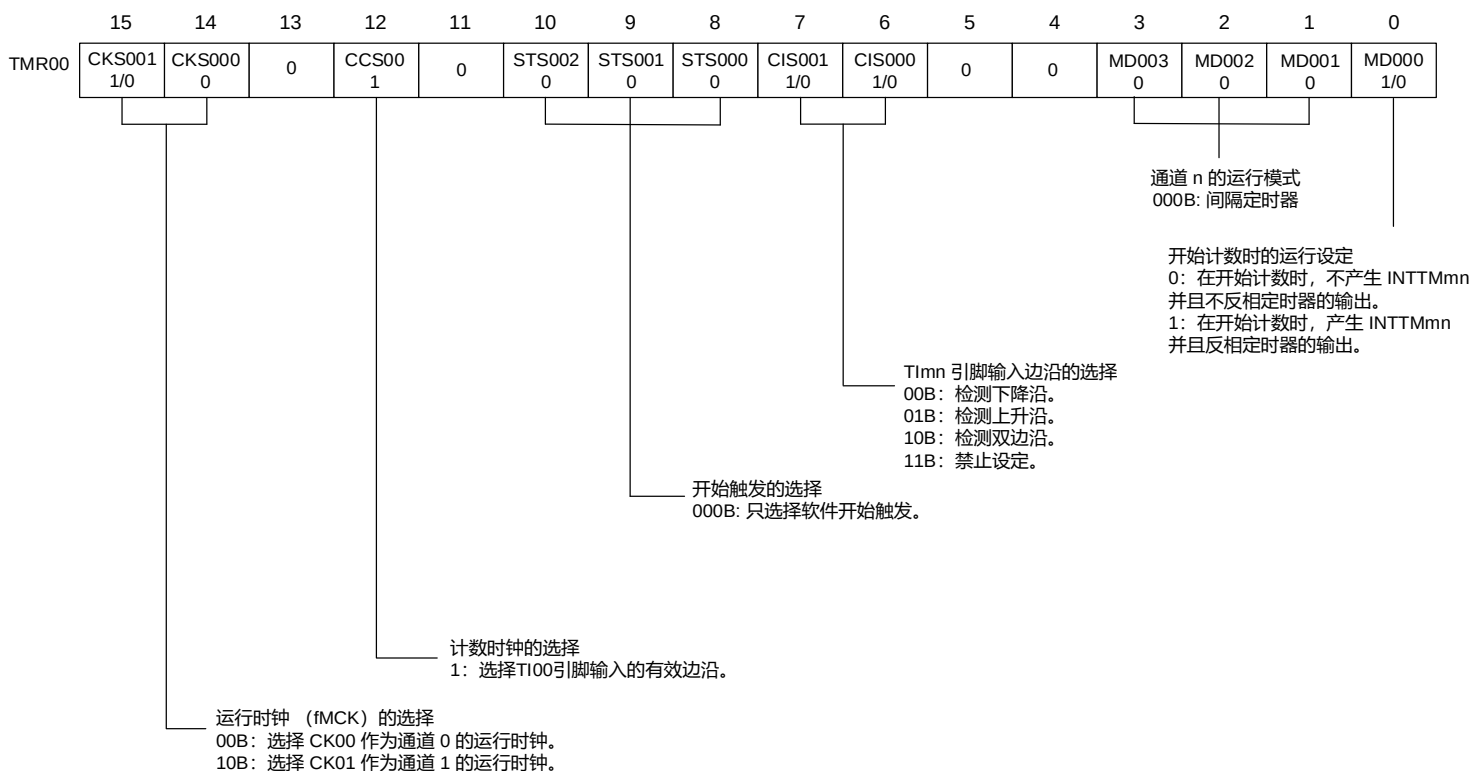
TE00：定时器通道允许状态寄存器0（TE0）的bit0

TI00：TI00引脚输入信号；TCR00：定时器计数寄存器00（TCR00）

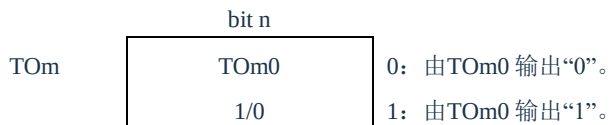
TDR00：定时器数据寄存器00（TDR00）；TO00：TO00引脚输出信号

图6-25 作为分频器运行时的寄存器设定内容例子

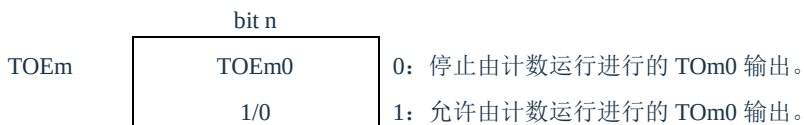
(a)定时器模式寄存器m0 (TMR00)



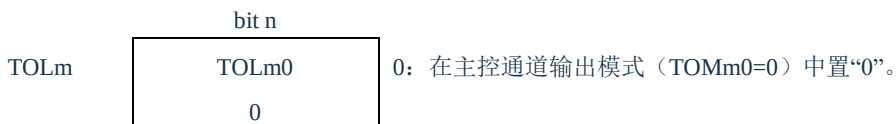
(b)定时器输出寄存器m (TOM)



(c)定时器输出允许寄存器 m (TOEm)



(d)定时器输出电平寄存器0 (TOLm)



(e)定时器输出模式寄存器0 (TOMm)



表6-31 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(PER0)的TM4mEN位置“1”。 →	定时器单元m的输入时钟处于提供状态。 (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m0 (TMRm0) (确定通道的运行模式, 选择检测边沿)。 给定时器数据寄存00(TDRm0)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 (TOMm) 的TOMm0位置“1” (主控通道输出模式)	TOM0引脚处于Hi-Z输出状态
	将TOLm0置“0”。	
	设定TOM0位并确定TOM0输出的初始电平。 →	当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出TOM0初始设置的电平。
	将TOEm0位置“1”, 允许TOM0输出。 → 将端口寄存器和端口模式寄存器置“0” →	因为通道处于运行停止状态, 所以TOM0不变。 TOM0输出TOM0设定的电平。
开始运行	将TOEm0位置“1”(只限于重新开始运行)。 将TSm0位置“1” 因为TSm0是触发位, 所以自动返回到“0”	TEm0变为“1”并且开始计数。 将TDRm0寄存器的值装入定时计数器m0 (TCRm0)。当TMR00寄存器MDm00位为“1”时, 产生INTTMm0并且TOM0交替输出。
运行中	能任意更改TDRm0寄存器的设定值。 能随时读TCRm0寄存器。 不使用TSRm0寄存器。 能更改TOM寄存器和TOEm寄存器的设定值。 禁止更改TMRm0寄存器, TOMm0位和TOLm0位的设定值。	计数器(TCRm0)进行递减计数。如果计数到“0000H”, 就再次将TDRm0寄存器的值装入TCRm0寄存器并且继续计数。当检测到TCRm0为“0000H”时, 产生INTTMm0并且交替输出。此后, 重复此运行。
停止运行	将TTm0位置“1”。 → 因为TTm0位是触发位, 所以自动返回到“0”。	TEm0位变为“0”并且停止计数。 TCRm0寄存器保持计数值而停止计数。 TOM0输出不被初始化而保持状态。
	将TOEm0位置“0”并且给TOM0位设定值。 →	TOM0引脚输出TOM0位设定的电平。
TIMER4 停止	要保持TOM0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TOM0位置“0”。 → 不需要保持TOM0引脚输出电平的情况: 不需要设定。	通过端口保持TOM0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。 (TOM0位变为“0”并且TOM0引脚变为端口功能)

重新开始运行

注: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

6.7.4 作为输入脉冲间隔测量的运行

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

脉冲间隔能用以下计算式进行计算：

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

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

在捕捉模式中，定时器计数寄存器mn (TCRmn) 用作递增计数器。

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

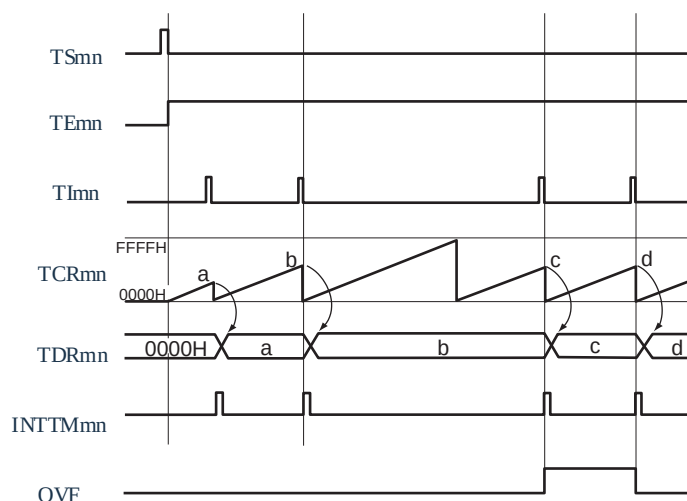
如果检测到 TIMn 引脚输入的有效边沿，就将 TCRmn 寄存器的计数值传送 (捕捉) 到定时器数据寄存器 mn (TDRmn)，同时将 TCRmn 寄存器清“0000H”，然后输出 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器mn (TSRmn) 的 OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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

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

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

图6-26 作为输入脉冲间隔测量的运行基本时序例子 (MDmn0=0)



注1: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

注2: TSmn: 定时器通道开始寄存器m (TSm) 的bit n

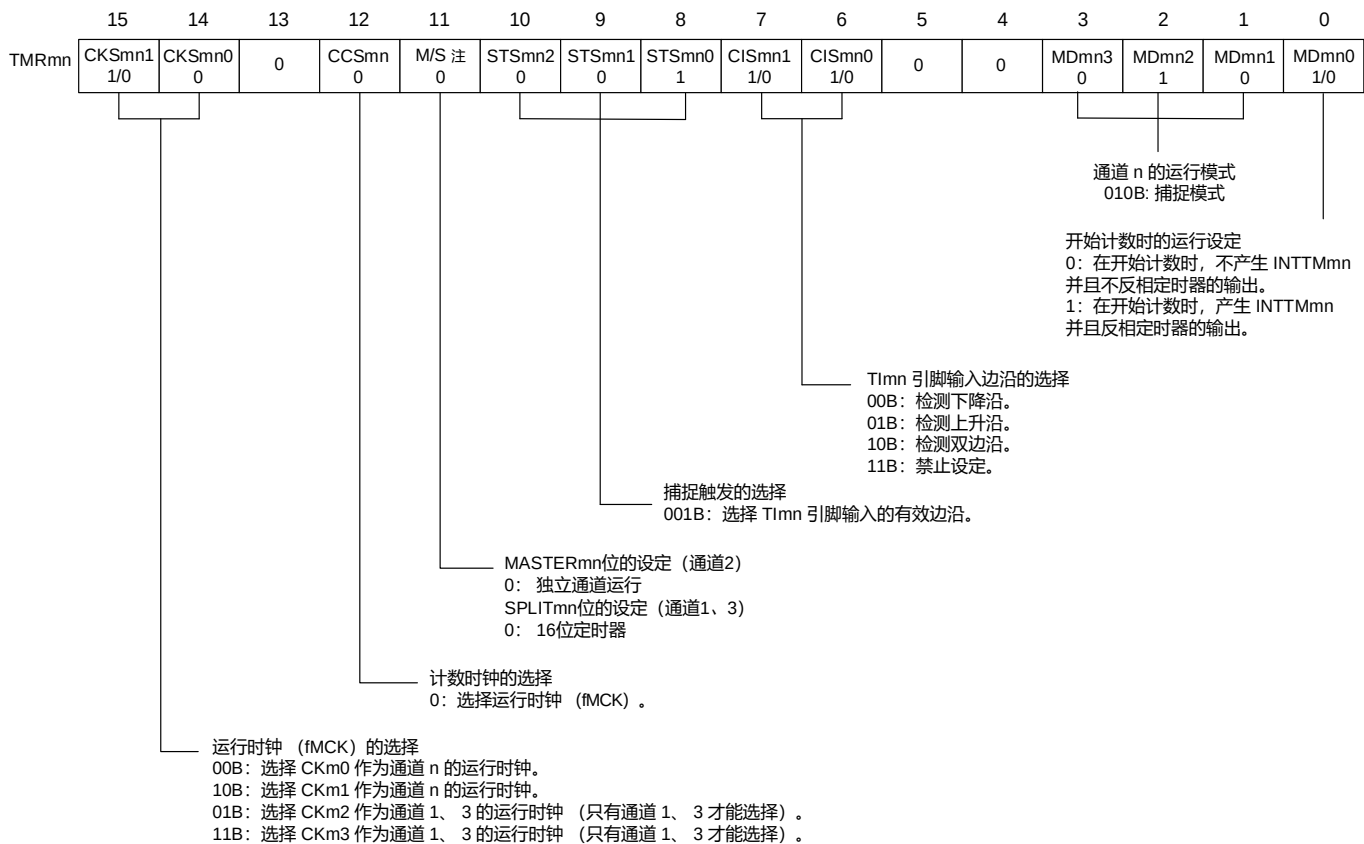
TEMn: 定时器通道允许状态寄存器m (TEm) 的bit n

TImn: TIMn 引脚输入信号; TCRmn : 定时器计数寄存器mn (TCRmn)

TDRmn : 定时器数据寄存器mn (TDRmn); OVF: 定时器状态寄存器mn (TSRmn) 的bit0

图6-27 测量输入脉冲间隔时的寄存器设定内容例子

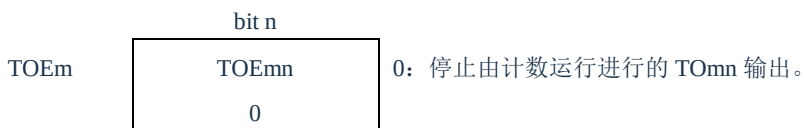
(a)定时器模式寄存器mn (TMRmn)



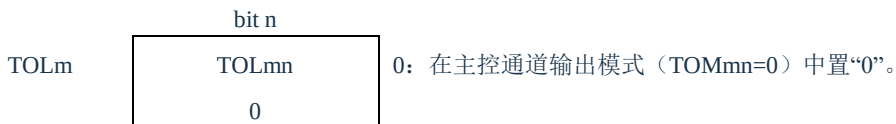
(b)定时器输出寄存器m (TOM)



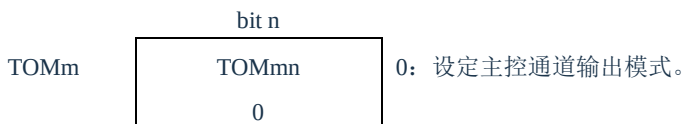
(c)定时器输出允许寄存器m (TOEm)



(d)定时器输出电平寄存器m (TOLm)



(e)定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

表6-32 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(PER0)的 TM40EN 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 00 (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 是触发位, 所以自动返回到“0”	TEmn 变为“1”并且开始计数。 将定时计数器 mn (TCRmn) 清 “0000H”。当 TMRmn 寄存器 MDmn0 位为 “1”时, 产生 INTTMmn。
	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。 能随时读 TCRmn 寄存器。 能随时读 TDRmn 寄存器。 能随时读 TSRmn 寄存器。 禁止更改 TOMmn 位, TOLmn 位, TOMn 和 TOEmn 位的设定值。	计数器 (TCRmn) 进行递增计数。如果检测 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。
	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 TM40EN 位置“0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

重新开始运行

注: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

6.7.5 作为输入信号高低电平宽度测量的运行

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

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

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

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

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

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

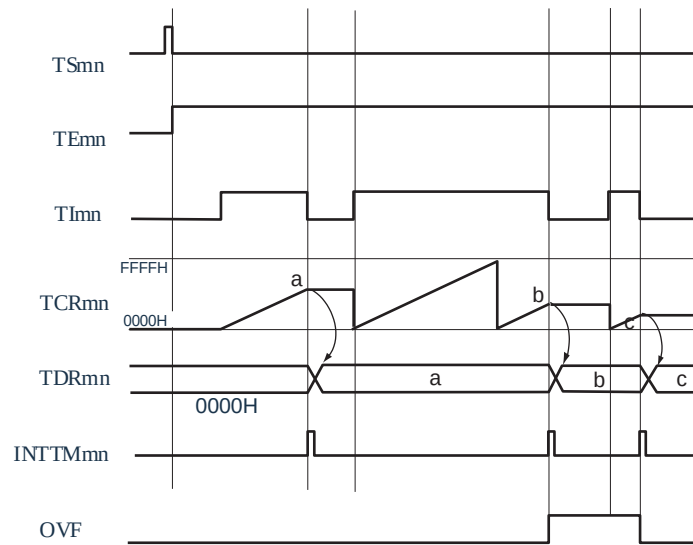
即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSRmn 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

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

TMRmn 寄存器的 CISmn1、CISmn0=10B：测量低电平宽度。

TMRmn 寄存器的 CISmn1、CISmn0=11B：测量高电平宽度。

图6-28 作为输入信号高低电平宽度测量的运行基本时序例子



注1: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

注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

TIImn: TIImn 引脚输入信号

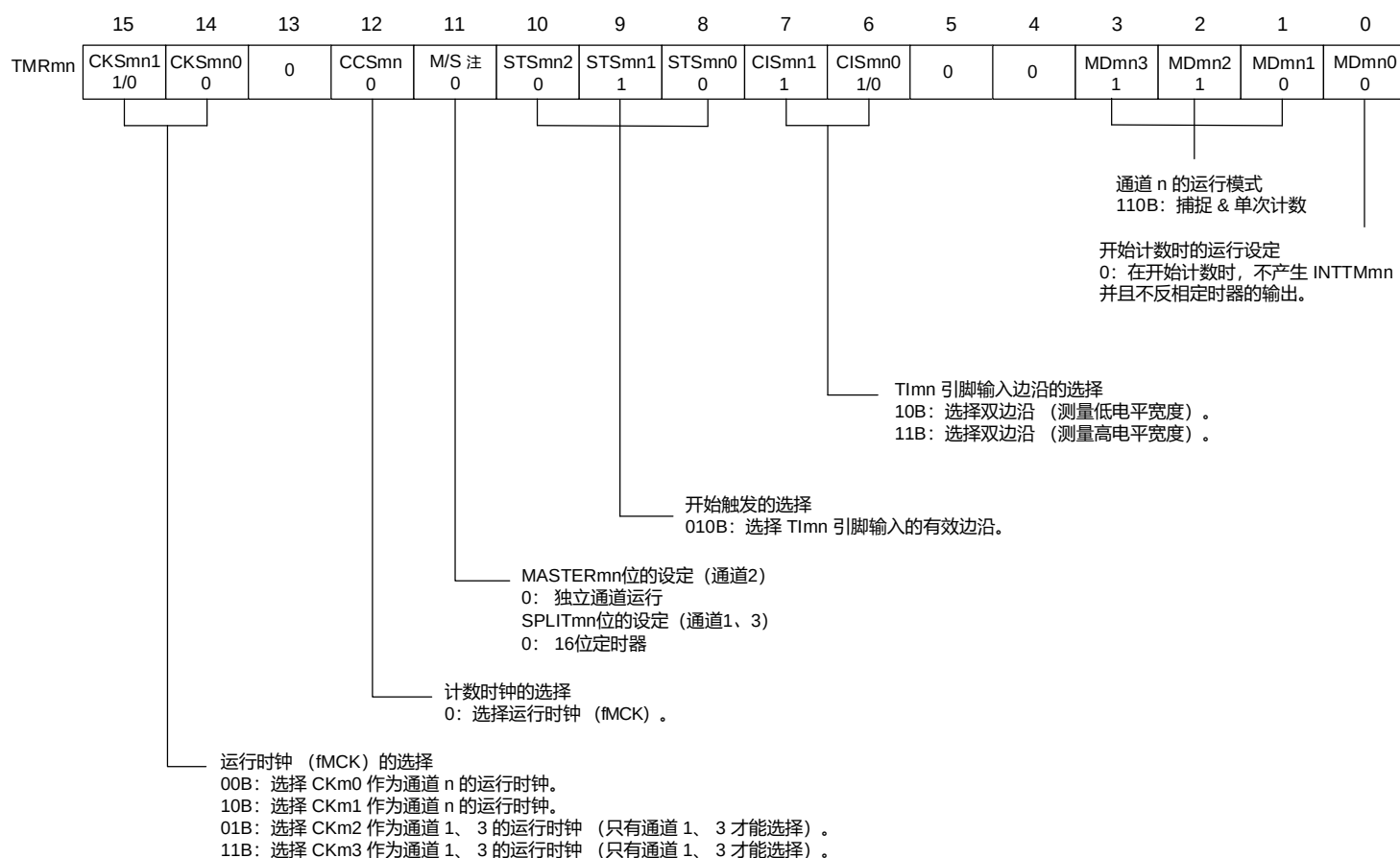
TCRmn : 定时器计数寄存器mn (TCRmn)

TDRmn : 定时器数据寄存器mn (TDRmn)

OVF: 定时器状态寄存器mn (TSRmn) 的bit0

图6-29 测量输入信号的高低电平宽度时的寄存器设定内容例子

(a) 定时器模式寄存器mn (TMRmn)



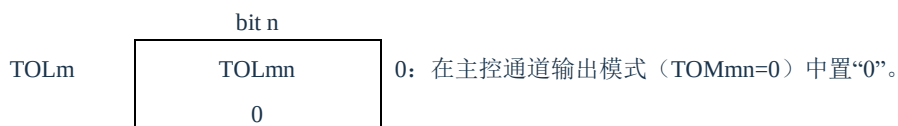
(b) 定时器输出寄存器m (TOM)



(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

表6-33 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(PER0)的TM40EN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00 (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是触发位, 所以自动返回到“0”。	TEmn变为“1”并开始触发 (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。	将定时计数器mn (TCRmn) 清 “0000H” 并且开始递增计数。
运行中	任意更改TDRmn寄存器的设定值。 能随时读TCRmn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TM40EN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。

注: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

重新开始运行

6.7.6 作为延迟计数器的运行

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

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

中断产生周期能用以下计算式进行计算：

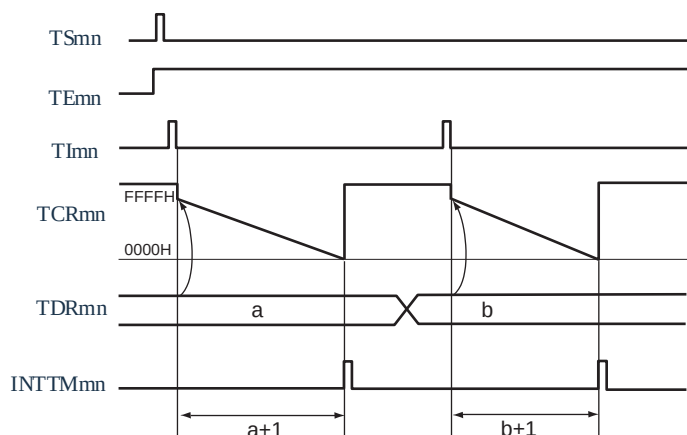
$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDRmn 的设定值} + 1)$$

在单次计数模式中，定时器计数寄存器mn（TCRmn）用作递减计数器。

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

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

图6-30 作为延迟计数器的运行基本时序例子



注1: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

注2: TSmn: 定时器通道开始寄存器m（TSM）的bit n

TEMn: 定时器通道允许状态寄存器m（TEM）的bit n

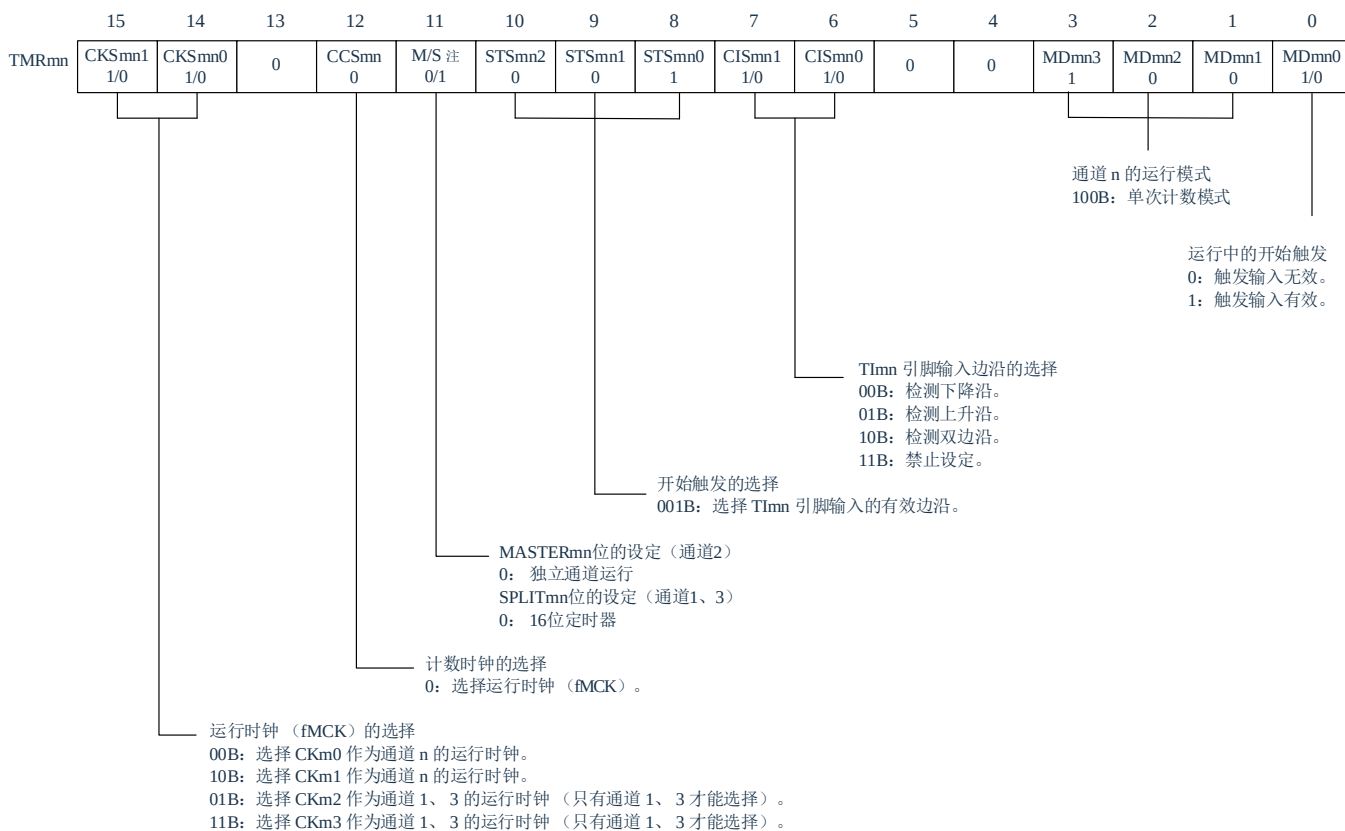
TImn: TIMn 引脚输入信号

TCRmn: 定时器计数寄存器mn（TCRmn）

TDRmn: 定时器数据寄存器mn（TDRmn）

图6-31 延迟计数器功能时的寄存器设定内容例子

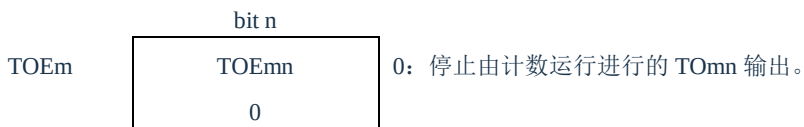
(a) 定时器模式寄存器mn (TMRmn)



(b) 定时器输出寄存器m (TOM)



(c) 定时器输出允许寄存器m (TOEm)



(d) 定时器输出电平寄存器m (TOLm)



(e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

表6-34 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(PER0)的 TM40EN 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 00 (TMRmn) (确定通道 n 的运行 模式)。 给定定时器数据寄存器 mn (TDRmn) 设定输出延时时间。 将 TOEmn 位置 “0” 并且停止 T0mn 的运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
重新 开始 运行	将 TSmn 位置“1”。	TEmn 变为“1”并开始触发 (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。
	通过检测到以下一个开始触发, 开始递减计数。 ·检测到 TIMn 引脚输入的有效边沿。	将 TDR 寄存器的值装入定时计数寄存器 mn(TCRmn)。
	·通过软件将 TSmn 位置 “1”。	
运行中	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) 开始递减计数。如果 TCRmn 计数到 “0000H”, 就产生 INTTMmn。并且在检 测到下一次触发 (检测 TIMn 引脚输入的有效边 沿或者将 TSmn 位置 “1”) 前, TCRmn 为 “0000H” 并停止计数。 此后, 重复此运行。
停止 运行	将 TTmn 位置“1”。	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。

注: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~3)

6.8 通用定时器单元的多通道联动运行功能

6.8.1 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。

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6-32 作为单触发脉冲输出功能运行的框图

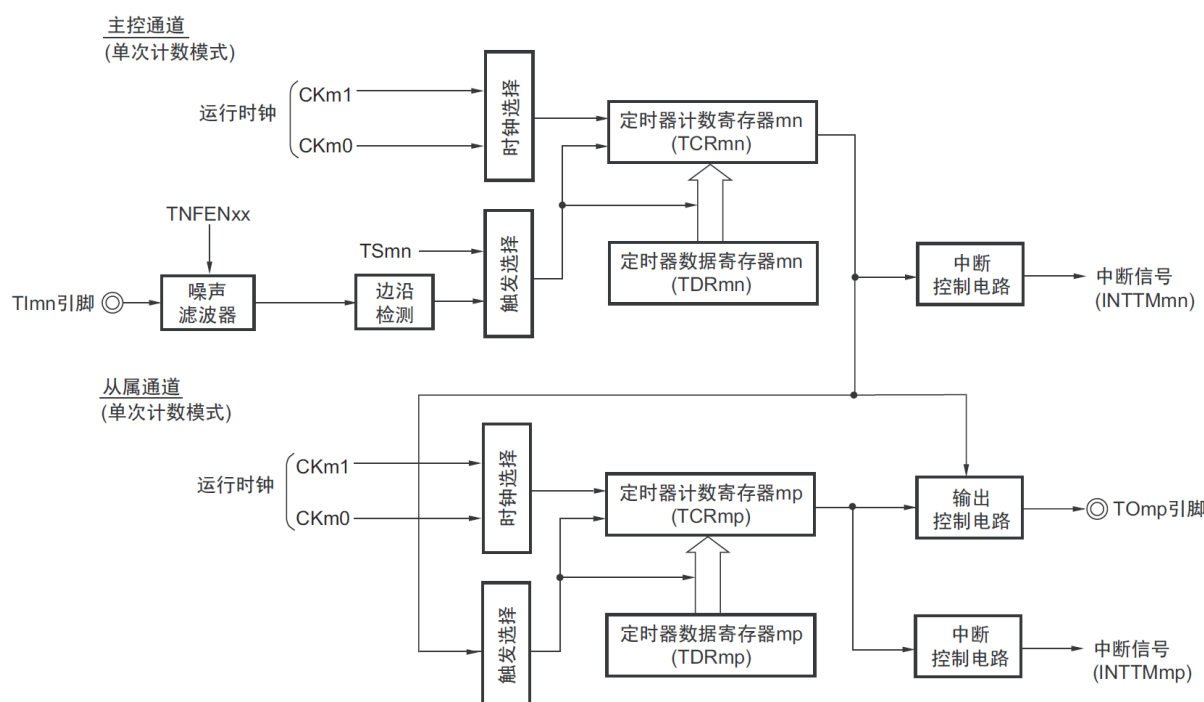
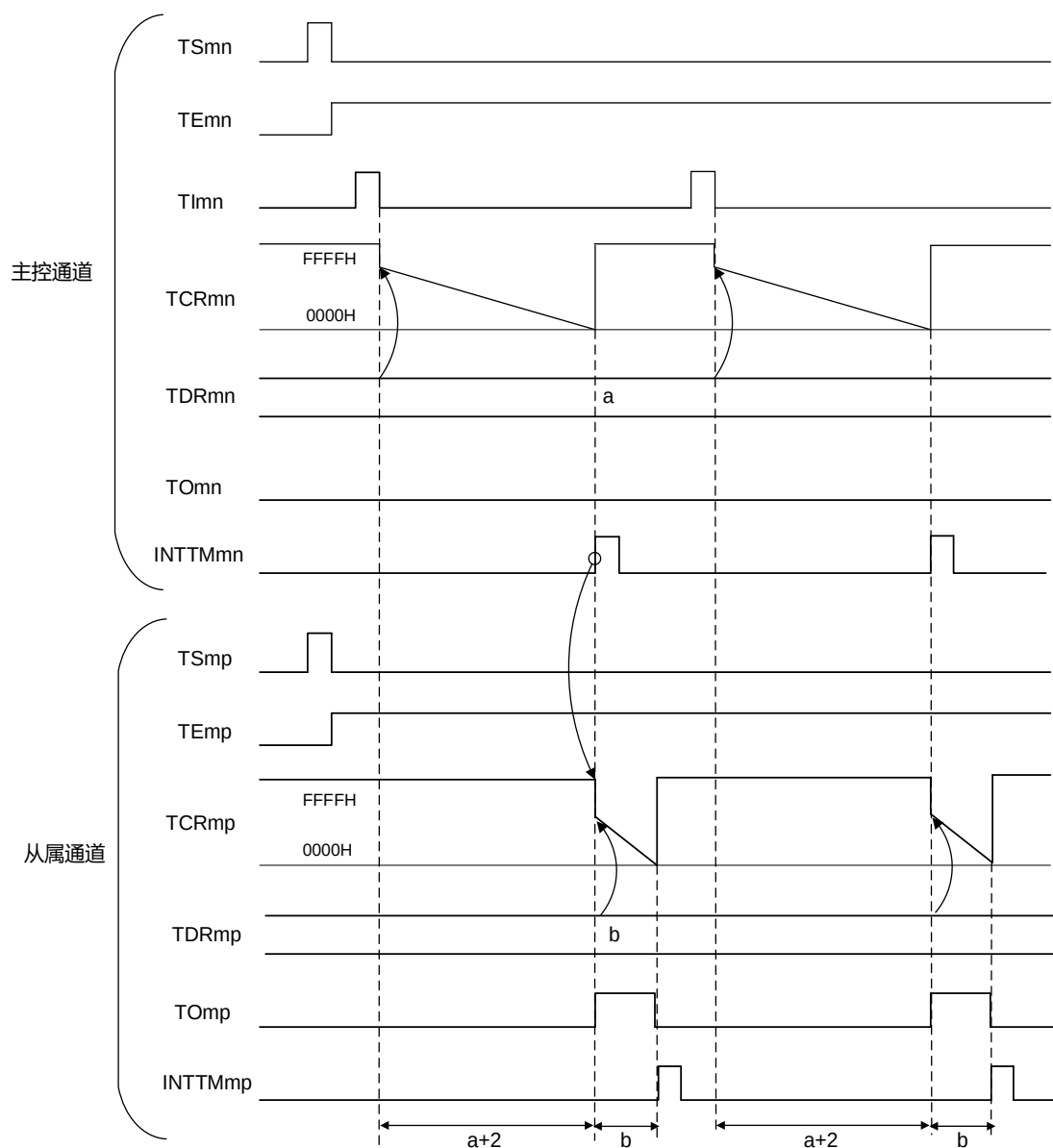


图6-33 作为单触发脉冲输出功能的运行基本时序例子



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)

注2: TSmn、TSmp: 定时器通道开始寄存器m (TSm) 的bit n、p

TE mn、TEmp: 定时器通道允许状态寄存器m (TEm) 的bit n、p

TImn、TImp: TImn 引脚和TImp 引脚的输入信号

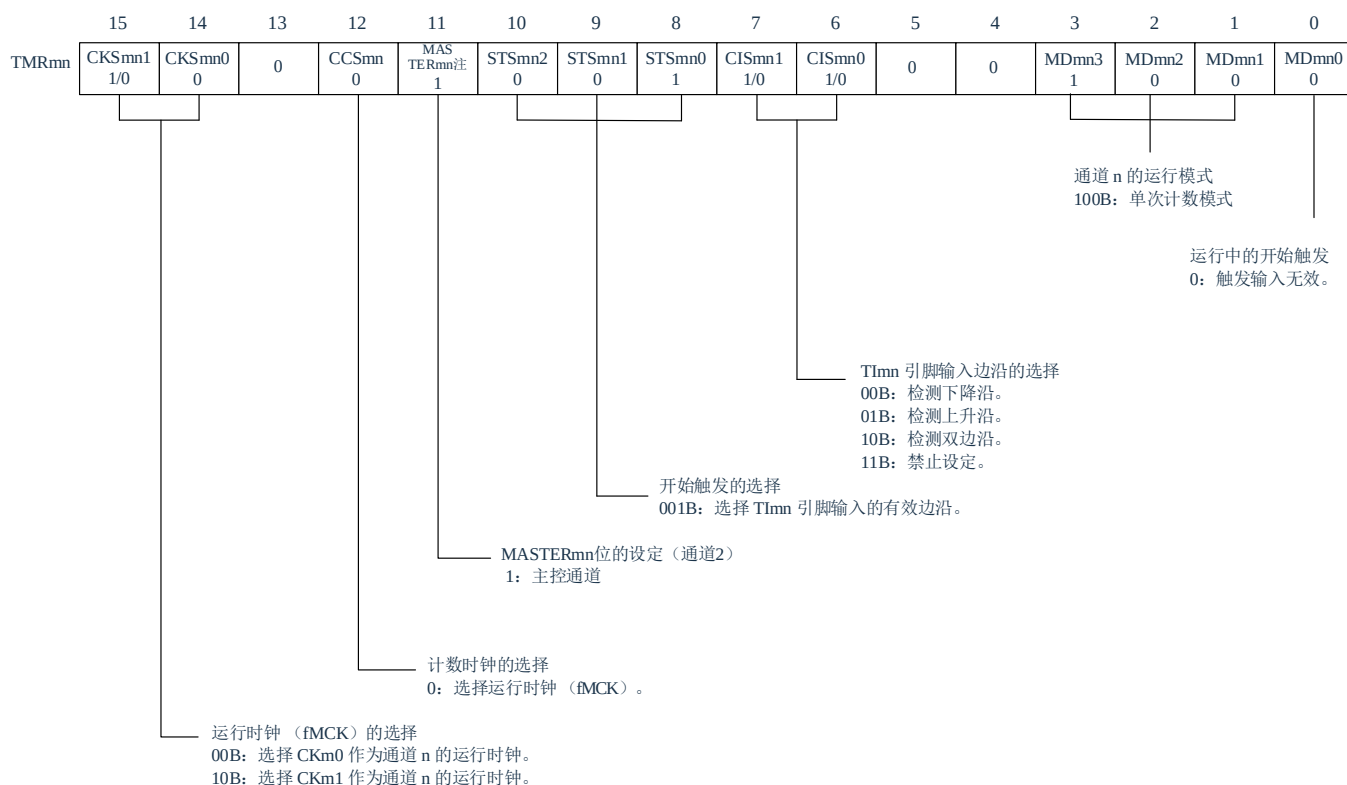
TCRmn、TCRmp : 定时器计数寄存器mn、mp (TCRmn、TCRmp)

TDRmn、TDRmp : 定时器数据寄存器mn、mp (TDRmn、TDRmp)

TOmn、TOmp: TOmn 引脚和TOmp 引脚的输出信号

图6-34 单触发脉冲输出功能时（主控通道）的寄存器设定内容例子

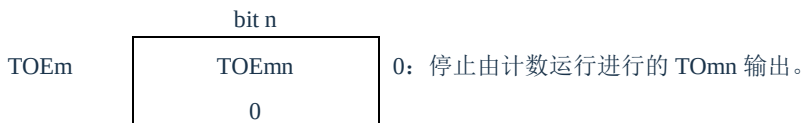
(a)定时器模式寄存器mn（TMRmn）



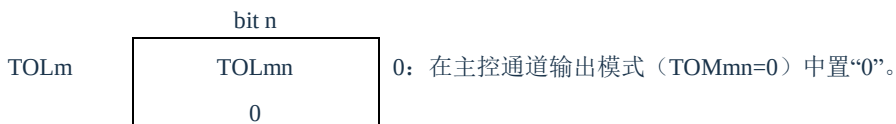
(b)定时器输出寄存器m（TOM）



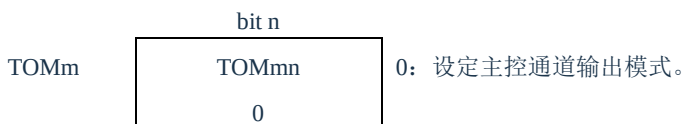
(c)定时器输出允许寄存器m（TOEm）



(d)定时器输出电平寄存器m（TOLm）



(e)定时器输出模式寄存器m（TOMm）



注1: TMRm2: MASTERmn 位

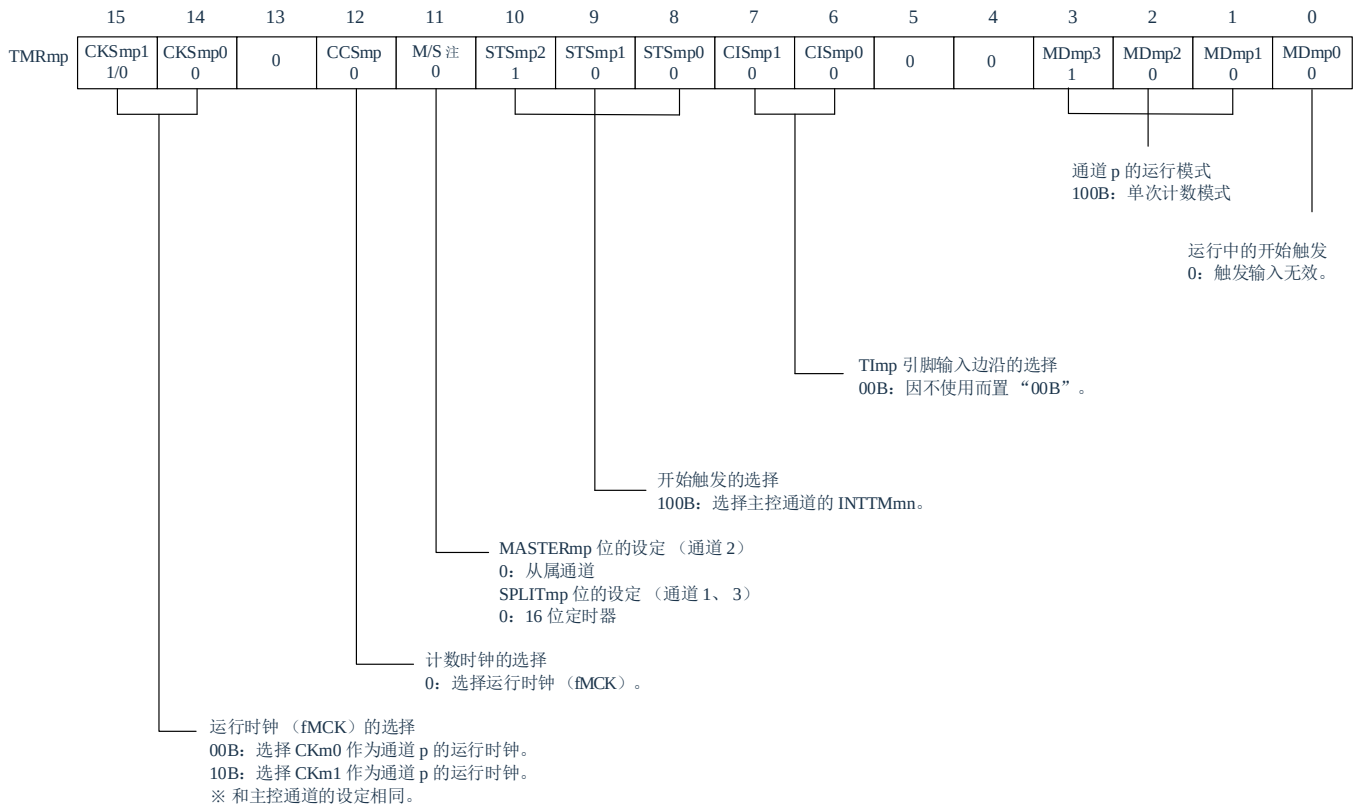
TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~2)

图6-35 单触发脉冲输出功能时（从属通道）的寄存器设定内容例子

(a)定时器模式寄存器mp（TMRmp）



(b)定时器输出寄存器m（TOM）

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

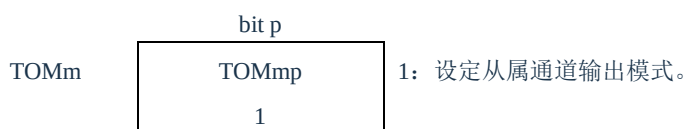
(c)定时器输出允许寄存器m（TOEm）

	bit p	
TOEm	TOEmp	0: 停止由计数运行进行的 TOmp 输出。
	1/0	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	0: 正逻辑输出（高电平有效）
	1/0	1: 负逻辑输出（低电平有效）

(e) 定时器输出模式寄存器m (TOMm)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

表6-35 单触发脉冲输出功能时的操作步骤(1/2)

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

表6-35 单触发脉冲输出功能时的操作步骤(2/2)

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

注1：m：单元号（m= 0~1） n：主控通道号（n=0）

p：从属通道号 q：从属通道号

$n < p < q \leq 3$ （p和q是大于n的整数）

注2：不能将从属通道的TSMn位置1

6.8.2 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：

脉冲周期= {TDRmn（主控）的设定值+1} *计数时钟周期
 占空比[%]= {TDRmp（从属）的设定值} / {TDRmn（主控）的设定值+1} *100
 0% 输出 : TDRmp（从属）的设定值= 0000H
 100% 输出 : TDRmp（从属）的设定值≥{TDRmn（主控）的设定值+1}

备注：当 TDRmp（从属）的设定值 > {TDRmn（主控）的设定值+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）的设定值装入定时器计数寄存器 mn（TCRmn），并且通过计数时钟进行递减计数。当计数到“0000H”时，在输出 INTTMmn 后再次将 TDRmn 寄存器的值装入 TCRmn 寄存器，并且进行递减计数。此后，在将定时器通道停止寄存器 m（TTm）的通道停止触发位（TTmn）置“1”前，重复此运行。

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

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

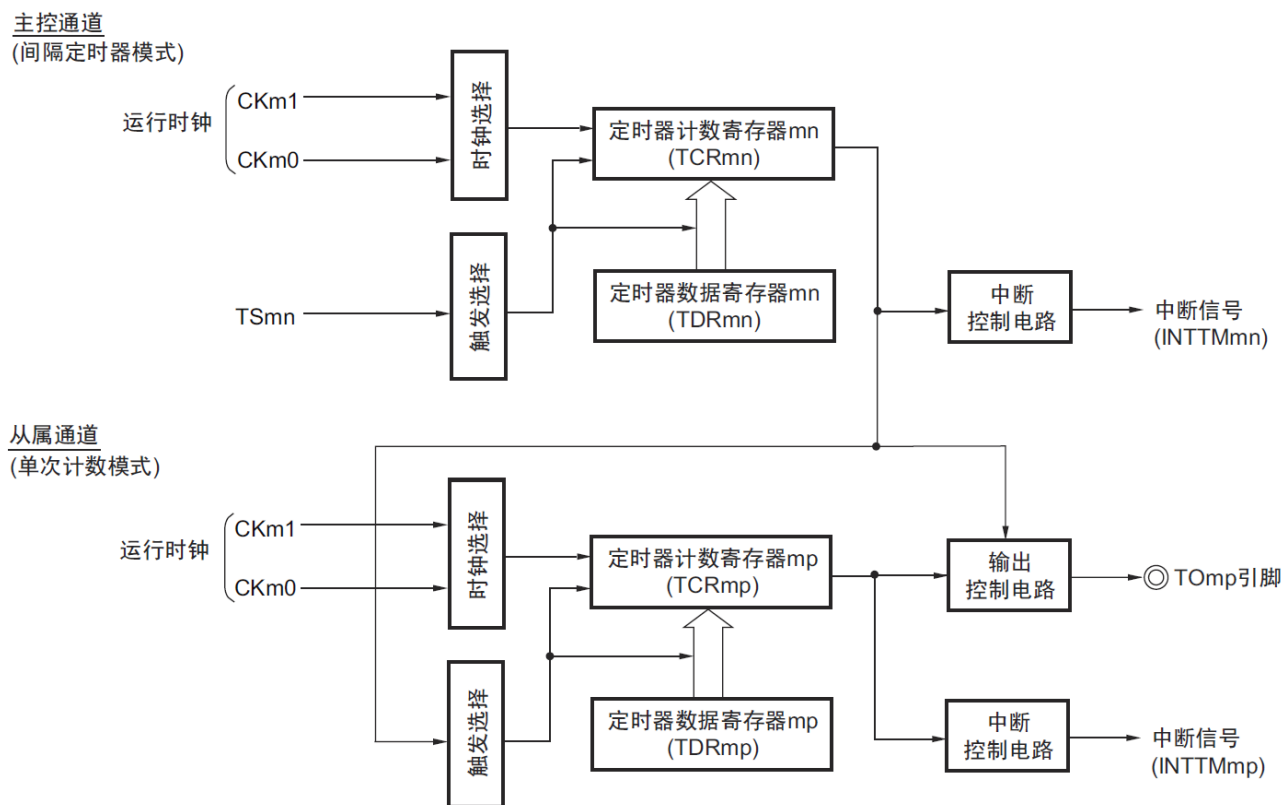
在从主控通道产生 INTTMmn 并且经过 1 个时钟后，PWM 输出（TOmp）变为有效电平，并且在从属通道的 TCRmp 寄存器的值为“0000H”时变为无效电平。

注意 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。

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）

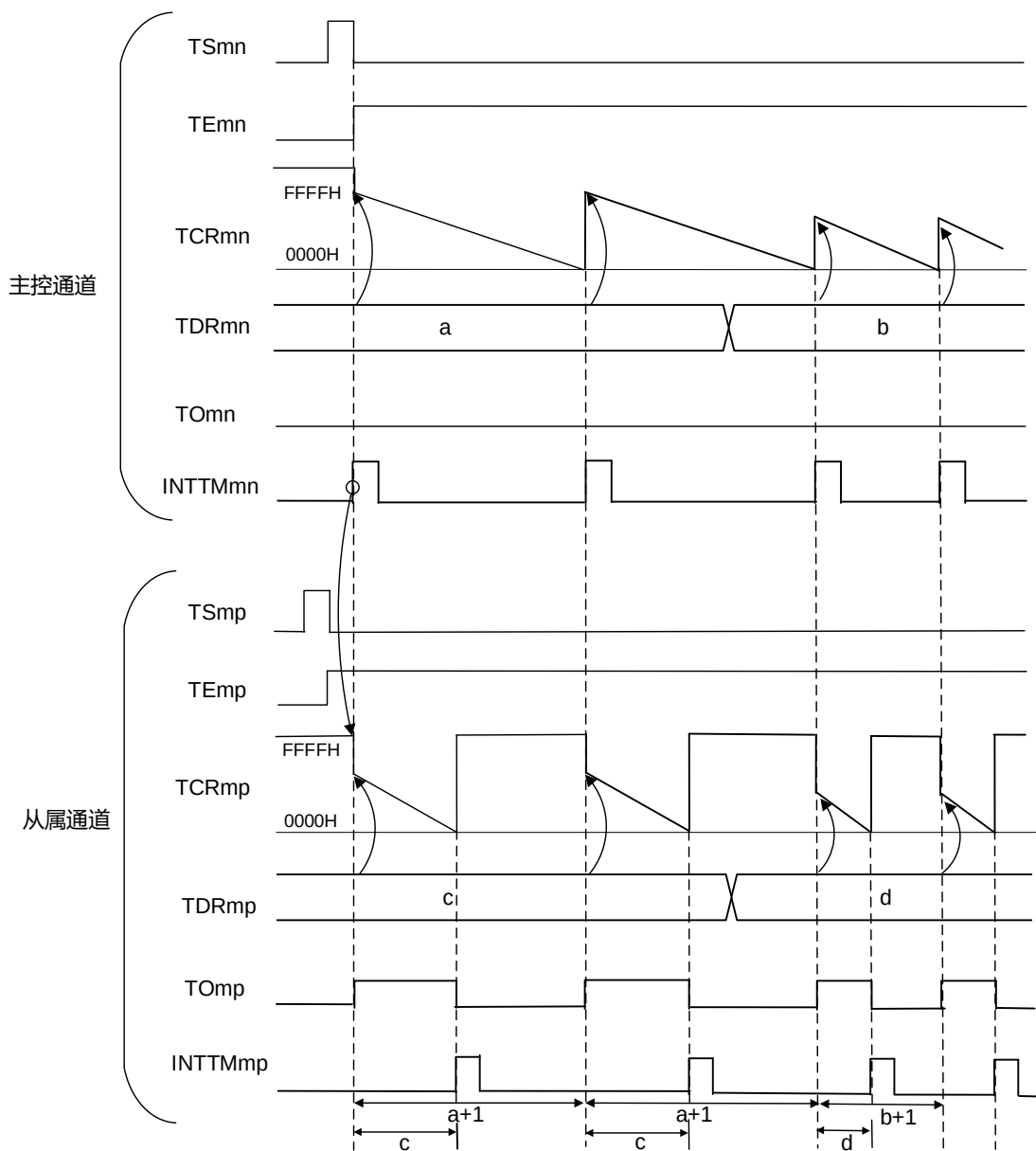
图6-36 作为PWM 功能运行的框图



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6-37 作为PWM功能的运行基本时序例子



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)

注2: TSmn、TSmp: 定时器通道开始寄存器m (TSM) 的bit n、p

TEmn、TEmp: 定时器通道允许状态寄存器m (TEM) 的bit n、p

TImn、TImp: TImn 引脚和TImp 引脚的输入信号

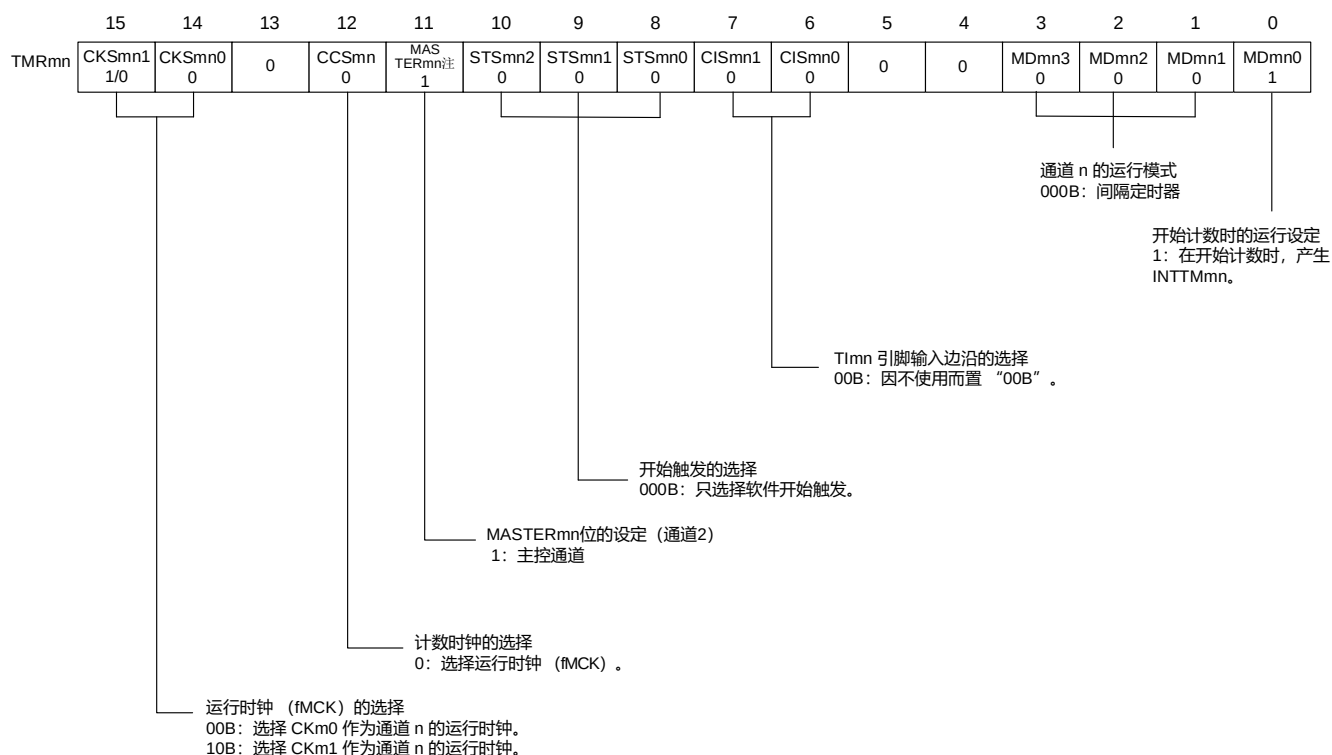
TCRmn、TCRmp : 定时器计数寄存器mn、mp (TCRmn、TCRmp)

TDRmn、TDRmp : 定时器数据寄存器mn、mp (TDRmn、TDRmp)

TOmn、TOmp: TOmn 引脚和TOmp 引脚的输出信号

图6-38 PWM 功能时（主控通道）的寄存器设定内容例子

(a)定时器模式寄存器mn（TMRmn）



(b)定时器输出寄存器m（TOMm）



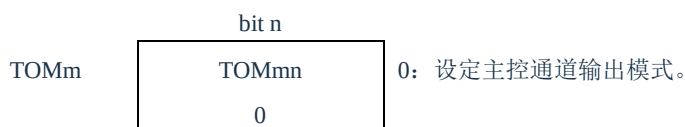
(c)定时器输出允许寄存器m（TOEm）



(d)定时器输出电平寄存器m（TOLm）



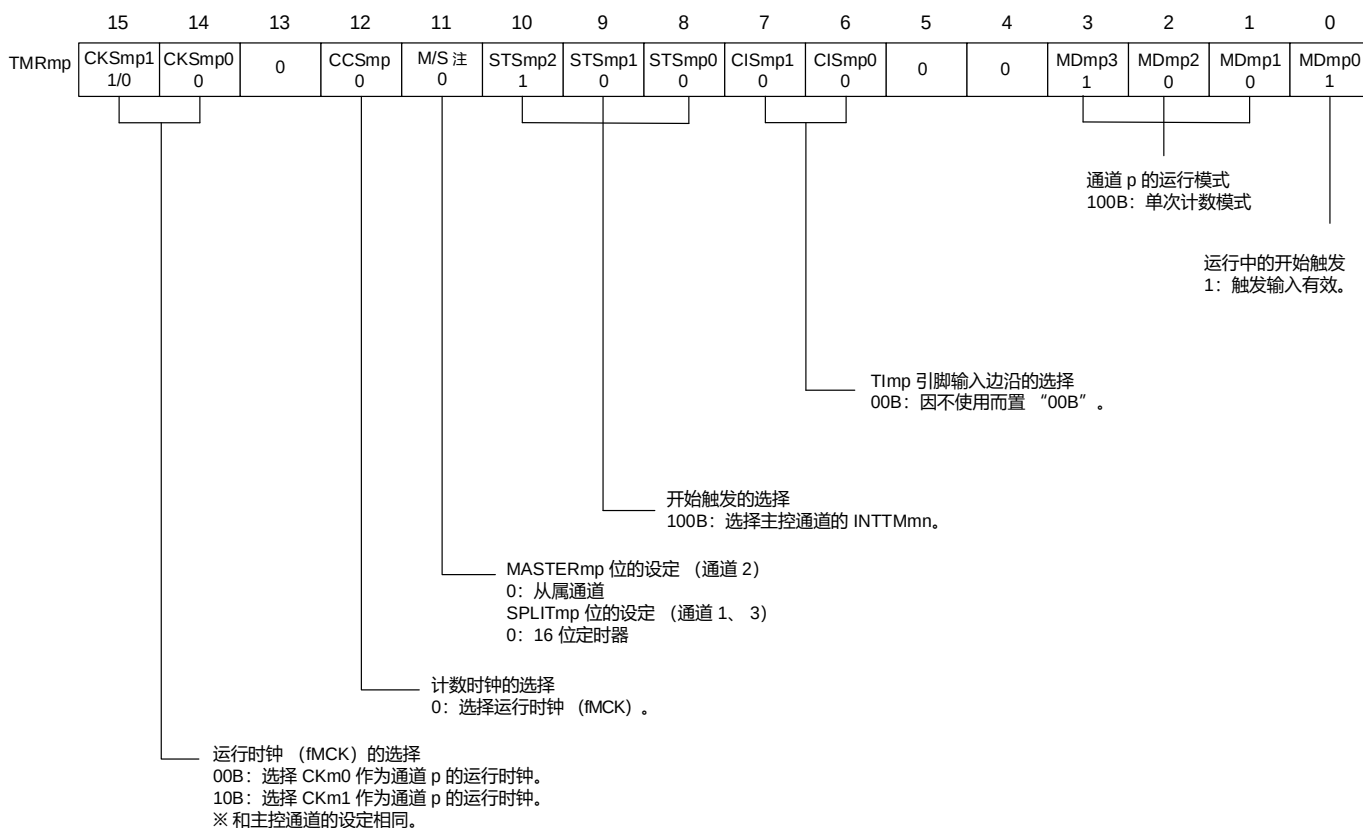
(e)定时器输出模式寄存器m（TOMm）



注：m：单元号（m= 0~1） n：主控通道号（n=0、2）

图6-39 PWM 功能时（从属通道）的寄存器设定内容例子

(a)定时器模式寄存器mp（TMRmp）



(b)定时器输出寄存器m（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。

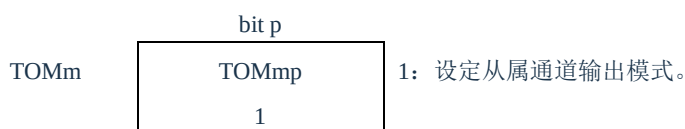
(d)定时器输出电平寄存器m（TOLm）

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)



注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

TMRm0: 固定为“0”。

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0 ~2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

表6-36 PWM 功能时的操作步骤 (1/2)

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

表6-36 PWM 功能时的操作步骤 (2/2)

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

注1：m：单元号（m=0） n：主控通道号（n=0）

p：从属通道号 q：从属通道号

$n < p < q \leq 3$ （p和q是大于n的整数）

注2：不能将从属通道的 TSmn 位置1

6.8.3 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：

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

备注：当 TDRmp（从属1）的设定值 > {TDRmn（主控）的设定值+1} 或者 {TDRmq（从属2）的设定值} > {TDRmn（主控）的设定值+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，必须要在主控通道产生 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）。

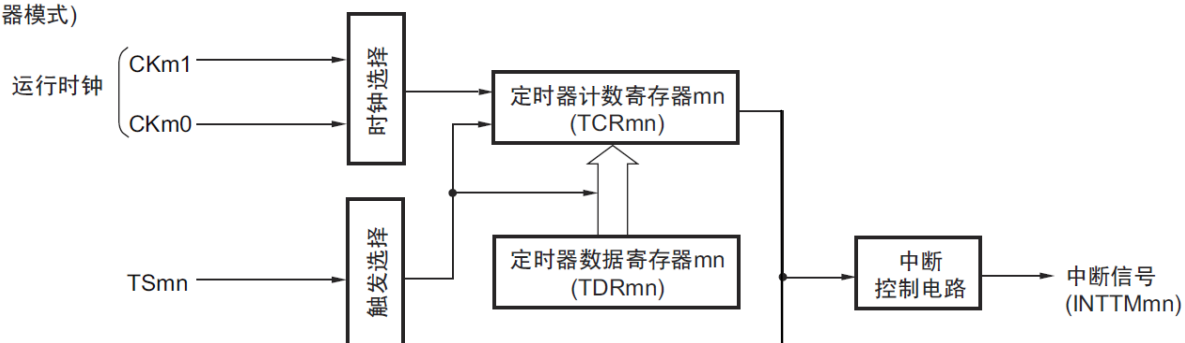
注：m：单元号（m=0~1），n：主控通道号（n=0）

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6-40 作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）

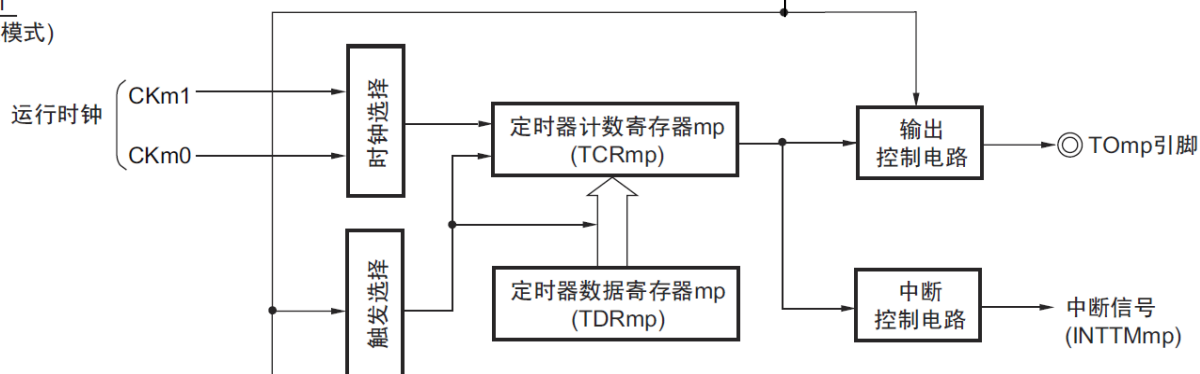
主控通道

(间隔定时器模式)



从属通道 1

(单次计数模式)



从属通道 2

(单次计数模式)

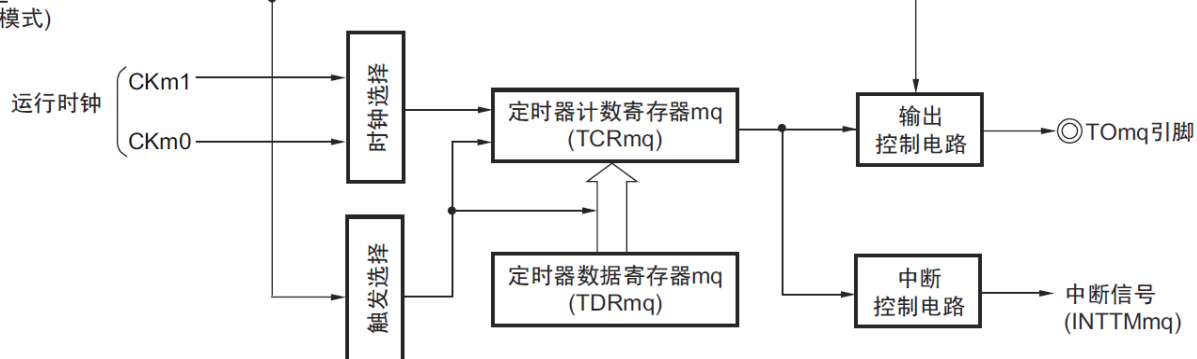
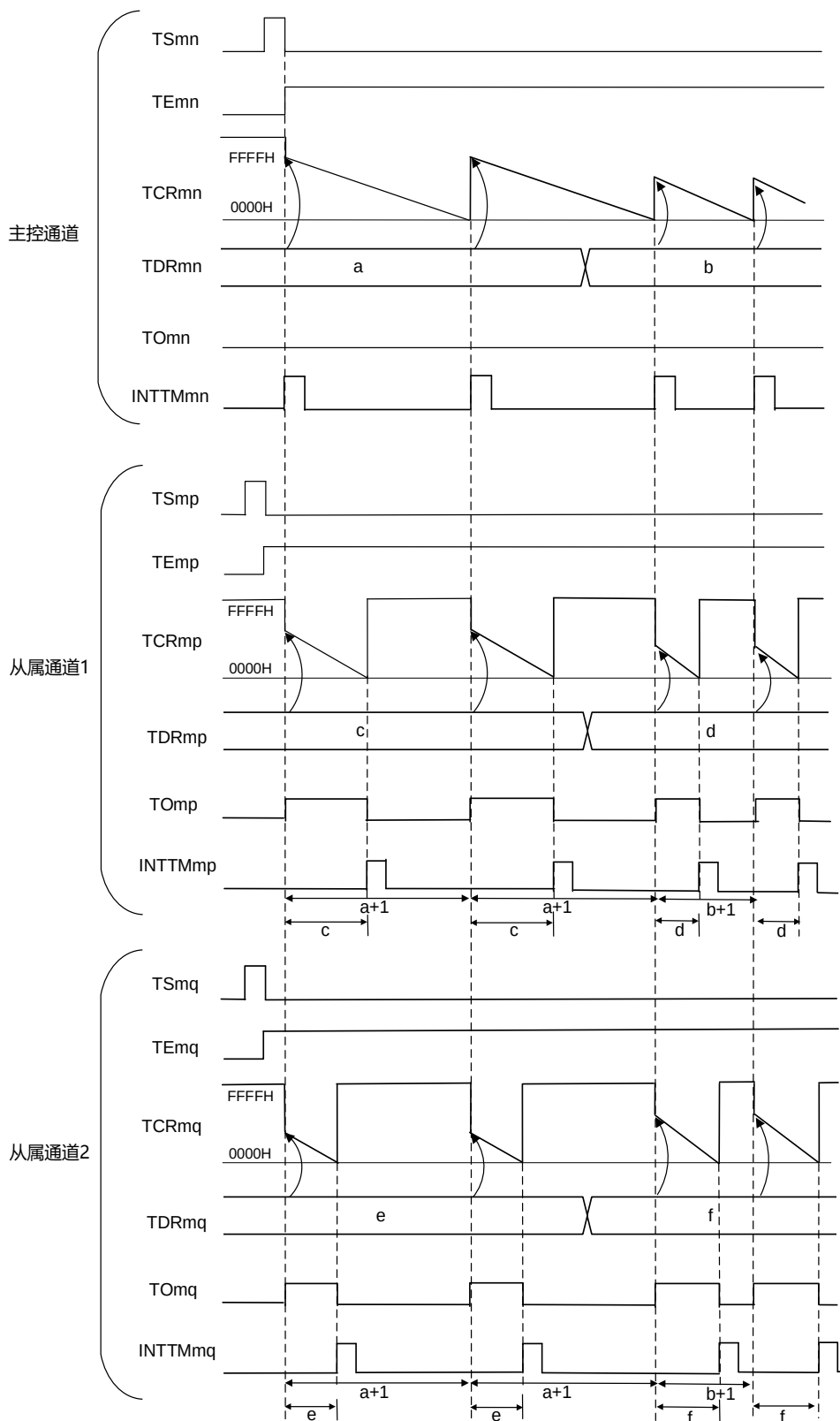


图6-41 作为多重 PWM 输出功能的运行基本时序例子（输出 2 种 PWM 的情况）



注1: m: 单元号 (m=0~1), n: 主控通道号 (n=0)

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

注2: TS_{mn} 、 TS_{mp} 、 TS_{mq} : 定时器通道开始寄存器m (TSm) 的bit n、p、q

TE_{mn} 、 TE_{mp} 、 TE_{mq} : 定时器通道允许状态寄存器m (TEm) 的bit n、p、q

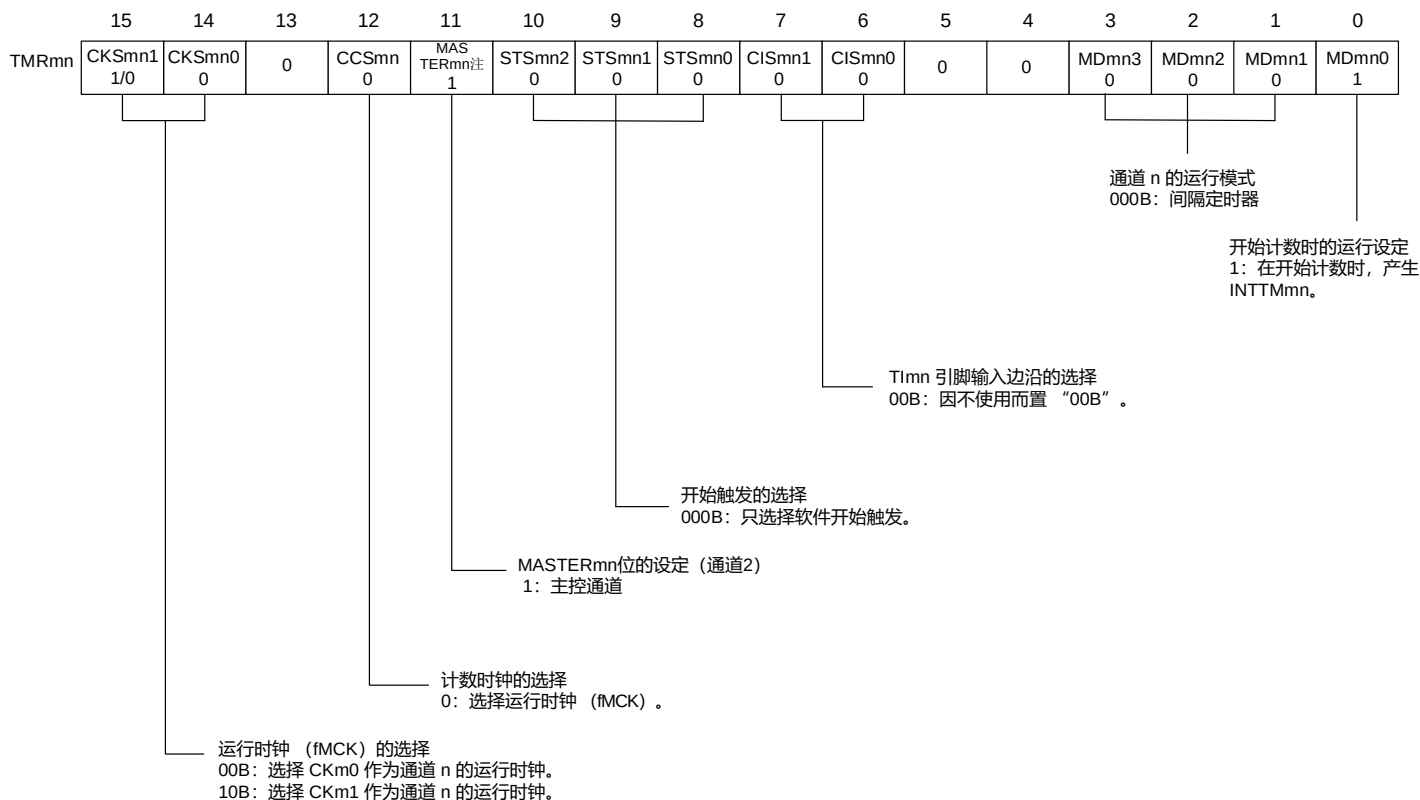
TCRmn、TCRmp、TCRmq：定时器计数寄存器mn、mp、mq（TCRmn、TCRmp、TCRmq）

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6-42 多重 PWM 输出功能时（主控通道）的寄存器设定内容例子

(a)定时器模式寄存器mn（TMRmn）



(b)定时器输出寄存器m（TOm）



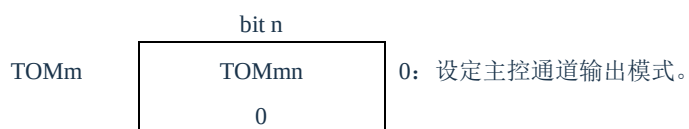
(c)定时器输出允许寄存器m（TOEm）



(d)定时器输出电平寄存器m（TOLm）



(e) 定时器输出模式寄存器m (TOMm)



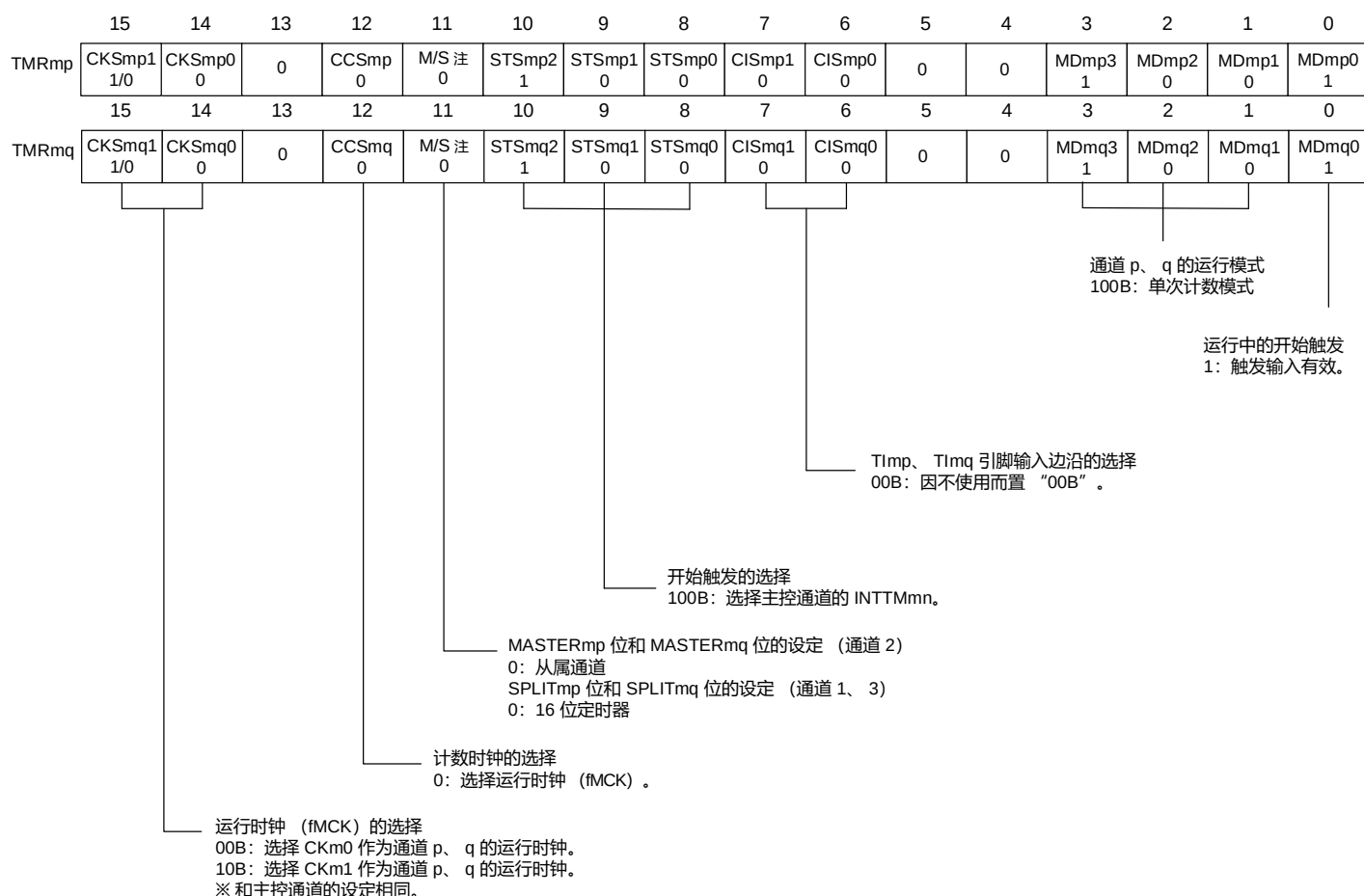
注1: TMRm2: MASTERmn 位

TMRm1、TMRm3 : SPLITmn位

注2: m: 单元号 (m= 0~1) n: 通道号 (n=0)

图6-43 多重 PWM 输出功能时（从属通道）的寄存器设定内容例子（输出 2 种 PWM 的情况）

(a)定时器模式寄存器mp、mq（TMRmp、TMRmq）



(b)定时器输出寄存器 m（TOM）

TOM	bit q	bit p	
	TOMq	TOMP	
	1/0	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）

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

(d)定时器输出电平寄存器 m（TOLm）

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

(e)定时器输出模式寄存器 m (TOMm)

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

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)

表6-37 多重 PWM 输出功能时的操作步骤 (输出 2 种 PWM 的情况) (1/2)

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

表6-37 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	(只在重新开始运行时将TOEmp 位和TOEmq 位 (从属) 置 “1”) 将定时器通道开始寄存器m (TSm) 的TSmn 位 (主控)、TSmp 位和TSmq 位 (从属) 同时置 “1” 。因为TSmn 位、TSmp 位和TSmq 位是触发位, 所以自动返回到 “0” 。	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。
	运行中	禁止更改TMRmn、TMRmp、TMRmq 寄存器以及 TOMmn 位、TOMmp 位、TOMmq 位、TOLmn 位、TOLmp、TOLmq 位的设定值。 能在主控通道产生INTTMmn 后更改TDRmn、TDRmp、TDRmq 寄存器的设定值。 能随时读TCRmn、TCRmp、TCRmq 寄存器。 不使用TSRmn、TSRmp、TSRmq 寄存器。	主控通道将TDRmn 寄存器的值装入定时器计数寄存器mn (TCRmn) , 并且进行递减计数。如果TCRmn 计数到 “0000H” , 就产生INTTMmn。同时, 将TDRmn 寄存器的值装入TCRmn 寄存器, 并且重新开始递减计数。 从属通道1 以主控通道的INTTMmn 信号为触发, 将TDRmp 寄存器的值传送到TCRmp 寄存器, 并且计数器开始递减计数。在从主控通道输出INTTMmn 并且经过1 个计数时钟后, 将TOmp 的输出电平置为有效电平。然后, 如果计数到 “0000H” , 就在将TOmp 的输出电平置为无效电平后停止计数。 从属通道2 以主控通道的INTTMmn 信号为触发, 将TDRmq 寄存器的值传送到TCRmq 寄存器, 并且计数器开始递减计数。在从主控通道输出INTTMmn 并且经过1 个计数时钟后, 将TOMq 的输出电平置为有效电平。然后, 如果计数到 “0000H” , 就在将TOMq 的输出电平置为无效电平后停止计数。此后, 重复此运行。
	停止 运行	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” 。	TEmn 位、TEmp 位和TEmq 位都变为 “0” 并且停止计数。 TCRmn、TCRmp、TCRmq 寄存器保持计数值而停止计数。 TOmp 和TOMq 输出不被初始化而保持状态。
		将从属通道的TOEmp 位和TOEmq 位置 “0” 并且给 TOmp 位和TOMq 位设定值。	TOmp 引脚和TOMq 引脚输出TOmp 和TOMq 设定的电平。
	Timer4 停止	要保持TOmp 引脚和TOMq 引脚的输出电平的情况: 在给端口寄存器设定要保持的值后将TOmp 位和TOMq 位置 “0” 。不需要保持TOmp 引脚和TOMq 引脚的输出电平的情况: 不需要设定。 将 PER0 寄存器的 TM4mEN 位置 “0” 。	通过端口功能保持TOmp 引脚和TOMq 引脚的输出电平。 定时器单元m 的输入时钟处于停止提供状态。对全部电路和各通道的SFR 进行初始化。 (TOmp 位和TOMq 位变为 “0” 并且TOmp 引脚和TOMq 引脚变为端口功能)

注: m: 单元号 (m= 0~1) , n: 主控通道号 (n=0)

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

第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。

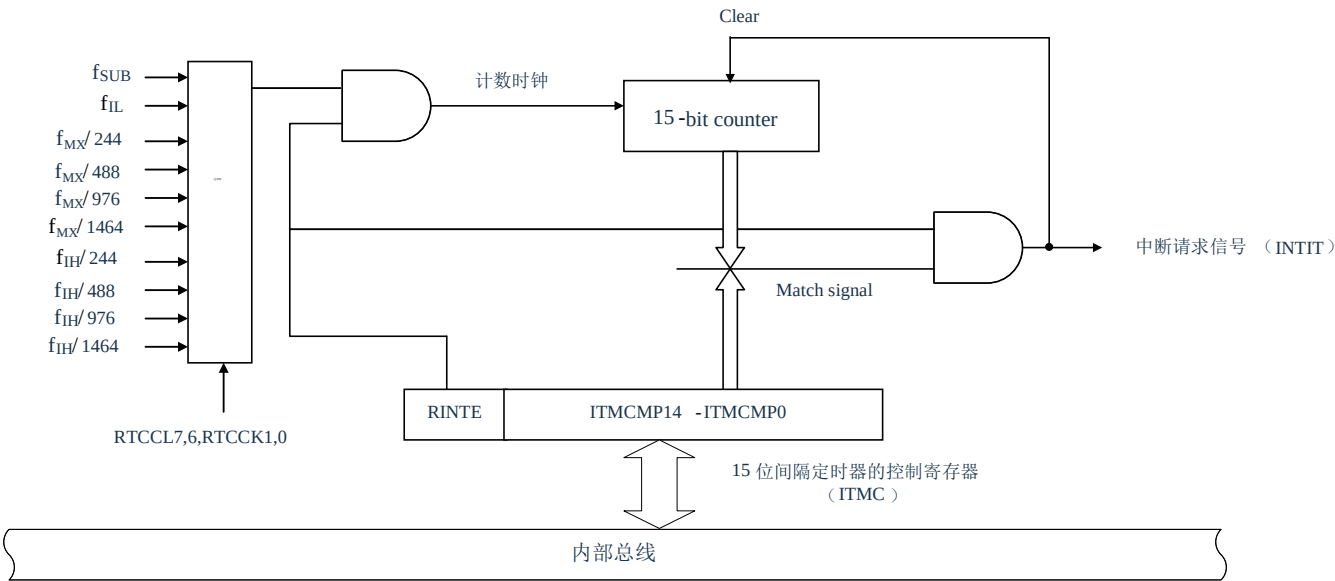
7.2 15位间隔定时器的结构

15位间隔定时器由以下硬件构成。

表7-115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 寄存器映射

(以下寄存器基地址 = 0x4004_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER0	0x020	R/W	外围允许寄存器0	0x00

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
RTCCL	0x00C	R/W	实时时钟选择寄存器	0x00

(以下寄存器基地址 = 0x4004_4F50) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
ITMC	0x000	R/W	15位间隔定时器控制寄存器	0x7FFF

7.4 寄存器说明

7.4.1 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”。

位	符号	描述	复位值
7	RTCEN	提供实时时钟（RTC）和15位间隔定时器输入时钟的控制 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。 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。	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

7.4.2 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} ）。

位	符号	描述	复位值
7	RTCCL7	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})	0
6	RTCCL6	15位间隔定时器的计数时钟的运行时钟的选择（详情见下表）	0
5:2	--	保留	0x0
1	RTCKS1	15位间隔定时器的计数时钟的运行时钟的选择（详情见下表）	0
0	RTCKS0	15位间隔定时器的计数时钟的运行时钟的选择（详情见下表）	0

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

7.4.3 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”。

位	符号	描述	复位值
15	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	0x0
14:0	ITCMP	15位间隔定时器比较值的设定	0x7FFF

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 $\times$ （ITCMP设置值+1）”的固定周期中断。
•	
•	
7FFFH	禁止设置。
0000H	

ITCMP14~ITCMP0为“0001H”或者“7FFFH”时的中断周期例子

- ITCMP14~ITCMP0=0001H，计数时钟： $f_{SUB}=32.768\text{kHz}$
 $1/32.768[\text{kHz}] \times (1+1) = 0.06103515625[\text{ms}] \approx 61.03[\mu\text{s}]$
- ITCMP14~ITCMP0=7FFFH，计数时钟： $f_{SUB}=32.768\text{kHz}$
 $1/32.768[\text{kHz}] \times (32767+1) = 1000[\text{ms}]$

注意：

1. 要将RINTE位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将INTIT设置为禁止中断处理后进行改写。要重新开始运行（从“0”改为“1”）时，必须在清除ITIF标志后设置为允许中断处理。
2. RINTE位的读取值在设置RINTE位后的1个计数时钟之后被反映。
3. 在从睡眠模式转移到通常运行模式后，如果要设置ITMC寄存器并且再次转移到睡眠模式，就必须在确认ITMC寄存器的写入值被反映后或者在设置ITMC寄存器后至少经过1个计数时钟之后再转移到睡眠模式。
4. 要更改ITCMP14~ITCMP0位的设置时，必须在RINTE位为“0”的状态下进行。
但是，能在将RINTE位从“0”改为“1”或者从“1”改为“0”的同时更改ITCMP14~ITCMP0位的设置。

7.5 15位间隔定时器的运行

7.5.1 15位间隔定时器的运行时序

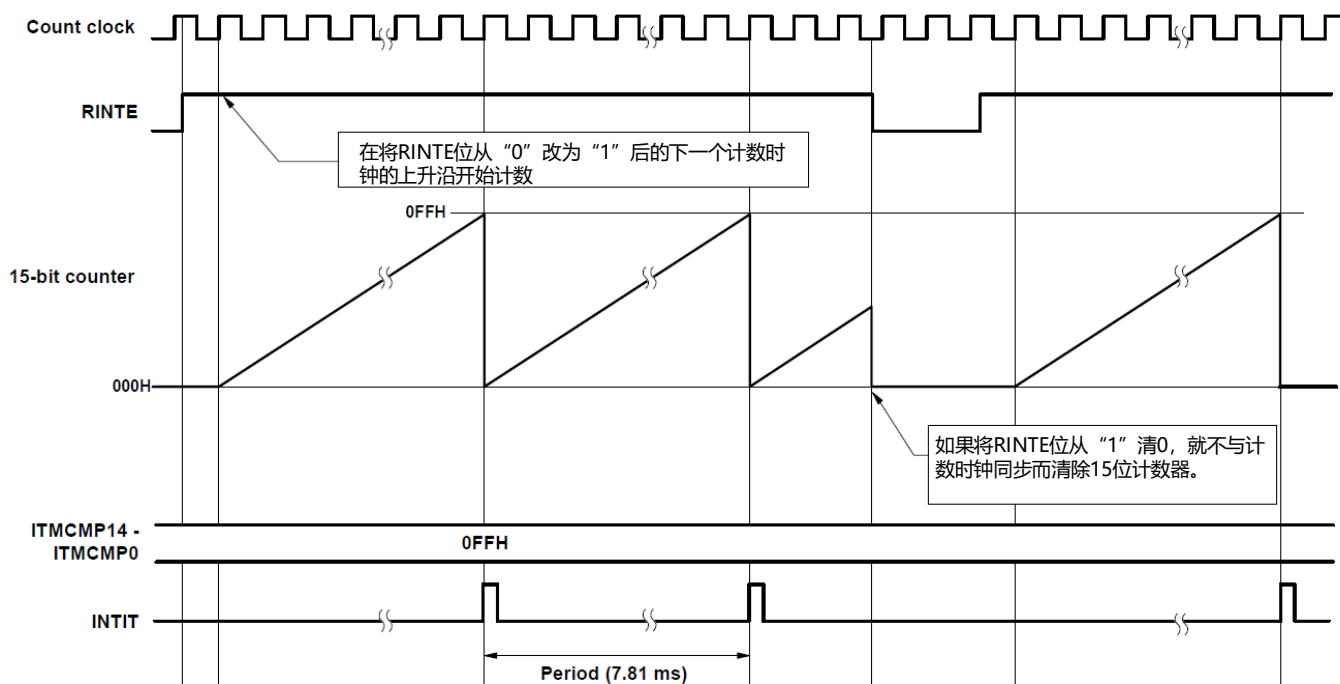
以ITCMP14~ITCMP0位设置的计数值为间隔，作为重复产生中断请求（INTIT）的15位间隔定时器运行。如果将RINTE位置“1”，15位计数器就开始计数。

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7-2所示。

图7-2 15位间隔定时器的运行时序

（ITCMP14~ITCMP0=0FFH，计数时钟： $f_{SUB}=32.768kHz$ ）

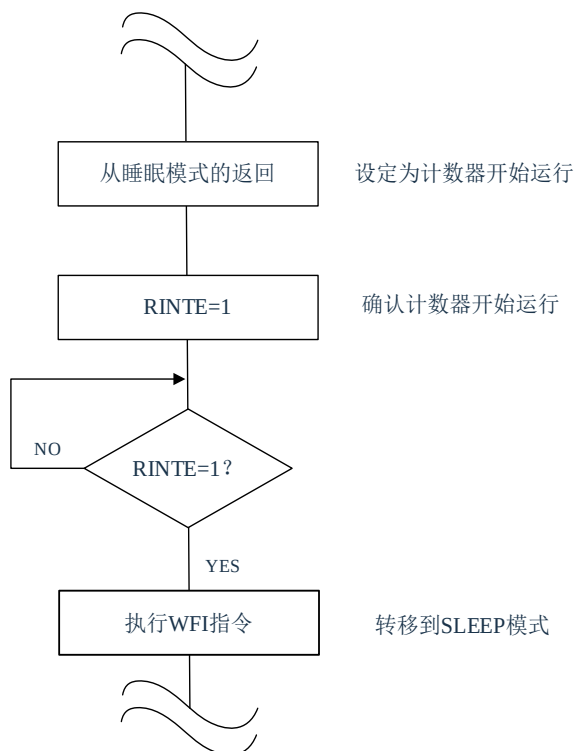


7.5.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

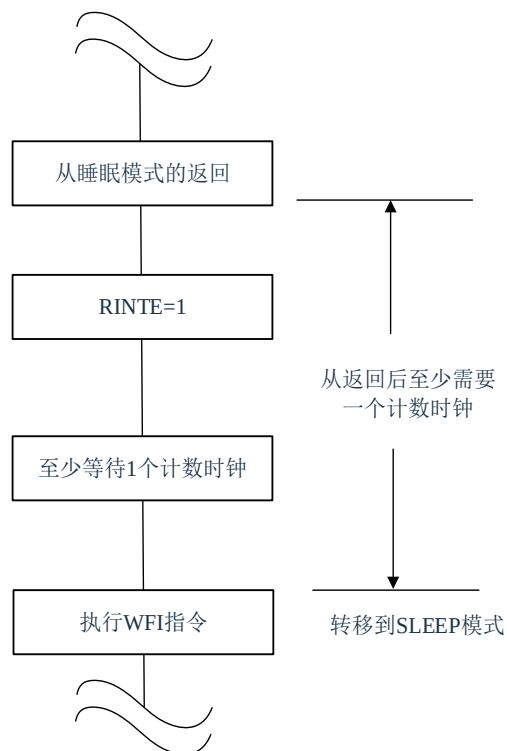
在从睡眠模式返回后，如果要将RINTE位置“1”并且再次转移到睡眠模式，就必须在将RINTE位置“1”后确认RINTE位的写入值被反映，或者在返回后至少经过1个计数时钟的时间之后再转移到睡眠模式。

- 在将RINTE位置“1”后，通过轮询确认RINTE位变为“1”，然后转移到睡眠模式（参照下图的例1）。
- 在将RINTE位置“1”后至少经过1个计数时钟的时间之后转移到睡眠模式（参照下图的例2）。

例1



例2



第8章 实时时钟（RTC）

8.1 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）。

8.2 实时时钟的结构

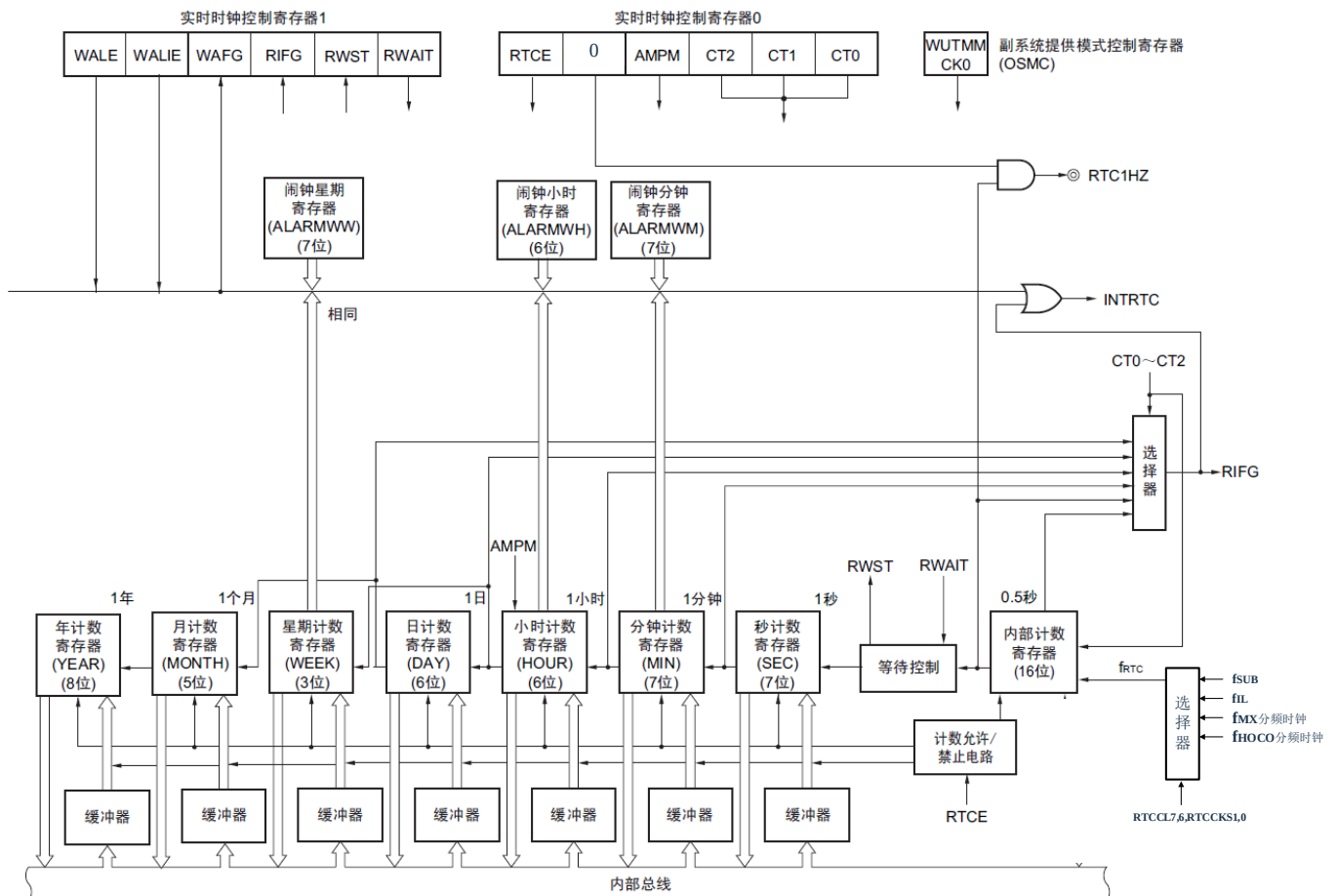
实时时钟由以下硬件构成。

表8-1 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）
	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8-1 实时时钟的框图



注意：只有在选择fmx/fhoco的分频时钟（分频后≈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}=32kHz）时，只能使用固定周期中断功能。

选择f_{IL}时的固定周期中断间隔用以下计算式进行计算：固定周期(RTCC0寄存器选择的值)X f_{SUB} / f_{IL}

8.3 寄存器映射

(以下寄存器基地址 = 0x4002_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER0	0x020	R/W	外围允许寄存器0	0x00

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
RTCCL	0x00C	R/W	实时时钟选择寄存器	0x00

(以下寄存器基地址 = 0x4004_4F50) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SEC	0x002	R/W	秒计数寄存器	0x00
MIN	0x003	R/W	分钟计数寄存器	0x00
HOURL	0x004	R/W	小时计数寄存器	0x00
DAY	0x006	R/W	日计数寄存器	0x00
WEEK	0x005	R/W	星期计数寄存器	0x00
MONTH	0x007	R/W	月计数寄存器	0x00
YEAR	0x008	R/W	年计数寄存器	0x00
-	-	-	-	-
ALARMWM	0x00a	R/W	闹钟分钟计数寄存器	0x00
ALARMWH	0x00b	R/W	闹钟小时计数寄存器	0x00
ALARMWW	0x00c	R/W	闹钟星期计数寄存器	0x00
RTCC0	0x00d	R/W	实时时钟控制寄存器0	0x00
RTCC1	0x00e	R/W	实时时钟控制寄存器1	0x00

8.4 寄存器说明

8.4.1 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”。

位	符号	描述	复位值
7	RTCEN	提供实时时钟（RTC）和15位间隔定时器输入时钟的控制 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。 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。	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

注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）
- 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2：能通过将15位间隔定时器运行时钟选择寄存器（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。

8.4.2 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} ）

位	符号	描述	复位值
7	RTCCL7	实时时钟的计数时钟的时钟源的选择： 0: 选择高速系统时钟(fmx) 1: 选择高速内部振荡器(fhoco)	0
6	RTCCL6	实时时钟的计数时钟的运行时钟的选择（详情见下表）	0
5:2	--	保留	0x0
1	RTCCKS1	实时时钟的计数时钟的运行时钟的选择（详情见下表）	0
0	RTCCKS0	实时时钟的计数时钟的运行时钟的选择（详情见下表）	0

表8-1 实时时钟计数时钟的运行时钟的选择

RTCCKS1	RTCCKS0	RTCCL6	实时时钟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fmx/fhoco(通过RTCCL7选择)/1464
1	0	1	主时钟fmx/fhoco(通过RTCCL7选择)/976
1	1	0	主时钟fmx/fhoco(通过RTCCL7选择)/488
1	1	1	主时钟fmx/fhoco(通过RTCCL7选择)/244

8.4.3 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、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”。

位	符号	描述	复位值
7	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	0
6:4	--	保留	0
3	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	0
2:0	CT2-CT0	固定周期中断（INTRTC）的选择（详情见下表）	0

注1：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。

表8-1 固定周期中断（INTRTC）的选择

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1：×：忽略

8.4.4 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”。

位	符号	描述	复位值
7	WALE	闹钟的运行控制： 0: 一致运行无效 1: 一致运行有效	0
6	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	0
5	--	保留	0
4	WAFG	闹钟检测状态标志： 0: 闹钟不一致 1: 检测到闹钟一致	0
3	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	0
2	--	保留	0
1	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	0
0	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	0

注1：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）。

注2：WAFG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。

注3：RIFG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。

注4：RWST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。

注5：RWAIT此位控制计数器的运行。要读写计数值时，必须给此位写“1”。因为内部计数器（16位）继续运行，所以必须在1秒内结束读写，然后返回到“0”。从将RWAIT位置“1”到能读写计数值（RWST=1）为止，最多需要1个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。

注6：固定周期中断和闹钟一致中断使用相同中断源（INTRTC）。在同时使用这2个中断的情况下，当发生INTRTC中断时，可以通过确认固定周期中断状态标志（RIFG）和闹钟检测状态标志（WAFG）来判断发生的是哪个中断。

注7：如果写秒计数寄存器（SEC），就清除内部计数器（16位）。

8.4.5 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”。

位	符号	描述	复位值
7	--	保留	0
6	SEC40	秒计数值	0
5	SEC20	秒计数值	0
4	SEC10	秒计数值	0
3	SEC8	秒计数值	0
2	SEC4	秒计数值	0
1	SEC2	秒计数值	0
0	SEC1	秒计数值	0

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

注2：如果写秒计数寄存器（SEC），就清除内部计数器（16位）。

8.4.6 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”。

位	符号	描述	复位值
7	--	保留	0
6	MIN40	分计数值	0
5	MIN20	分计数值	0
4	MIN10	分计数值	0
3	MIN8	分计数值	0
2	MIN4	分计数值	0
1	MIN2	分计数值	0
0	MIN1	分计数值	0

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

8.4.7 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”。

位	符号	描述	复位值
7:6	--	保留	0x0
5	HOUR20	小时计数值	0
4	HOUR10	小时计数值	0
3	HOUR8	小时计数值	0
2	HOUR4	小时计数值	0
1	HOUR2	小时计数值	0
0	HOUR1	小时计数值	0

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

注2：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）。

AMPM 位的设置值、小时计数寄存器（HOUR）的值和时间的关系如下表所示：

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

注1：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。

注2：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”。

8.4.8 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”。

位	符号	描述	复位值
7:6	--	保留	0x0
5	DAY20	日计数值	0
4	DAY10	日计数值	0
3	DAY8	日计数值	0
2	DAY4	日计数值	0
1	DAY2	日计数值	0
0	DAY1	日计数值	1

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

8.4.9 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”。

位	符号	描述	复位值
7:3	--	保留	0x0
2	WEEK4	星期计数值	0
1	WEEK2	星期计数值	0
0	WEEK1	星期计数值	0

星期计数寄存器（WEEK）值与时间的关系如下表所示：

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

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

注2：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。

8.4.10 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”。

	符号	描述	复位值
7:5	--	保留	0x0
4	MONTH10	月计数值	0
3	MONTH8	月计数值	0
2	MONTH4	月计数值	0
1	MONTH2	月计数值	0
0	MONTH1	月计数值	1

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

8.4.11 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fRTC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”。

位	符号	描述	复位值
7	YEAR80	年计数值	0
6	YEAR40	年计数值	0
5	YEAR20	年计数值	0
4	YEAR10	年计数值	0
3	YEAR8	年计数值	0
2	YEAR4	年计数值	0
1	YEAR2	年计数值	0
0	YEAR1	年计数值	0

注1：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“8.5.3 实时时钟计数器的读写”记载的步骤进行。

8.4.12 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”。

位	符号	描述	复位值
7	--	保留	0
6	WM40	闹钟分钟计数值	0
5	WM20	闹钟分钟计数值	0
4	WM10	闹钟分钟计数值	0
3	WM8	闹钟分钟计数值	0
2	WM4	闹钟分钟计数值	0
1	WM2	闹钟分钟计数值	0
0	WM1	闹钟分钟计数值	0

注1：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。

8.4.13 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”。

位	符号	描述	复位值
7	--	保留	0
5	WH20	闹钟小时计数值	0
4	WH10	闹钟小时计数值	0
3	WH8	闹钟小时计数值	0
2	WH4	闹钟小时计数值	0
1	WH2	闹钟小时计数值	0
0	WH1	闹钟小时计数值	0

注1：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。

注2：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）。

8.4.14 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”。

位	符号	描述	复位值
7	--	保留	0
6	WW6	闹钟星期计数值	0
5	WW5	闹钟星期计数值	0
4	WW4	闹钟星期计数值	0
3	WW3	闹钟星期计数值	0
2	WW2	闹钟星期计数值	0
1	WW1	闹钟星期计数值	0
0	WW0	闹钟星期计数值	0

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								
	W	W	W	W	W	W	W	10时	1时	10分	1分	10时	1时	10分	1分
	0	1	2	3	4	5	6								
每天 上午0时00分	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天 上午1时30分	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 上午11时59分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一～星期五 下午0时00分	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 下午1时30分	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 下午11时59分	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

8.5 实时时钟的运行

8.5.1 实时时钟的运行开始

图8-2 实时时钟的运行开始步骤



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

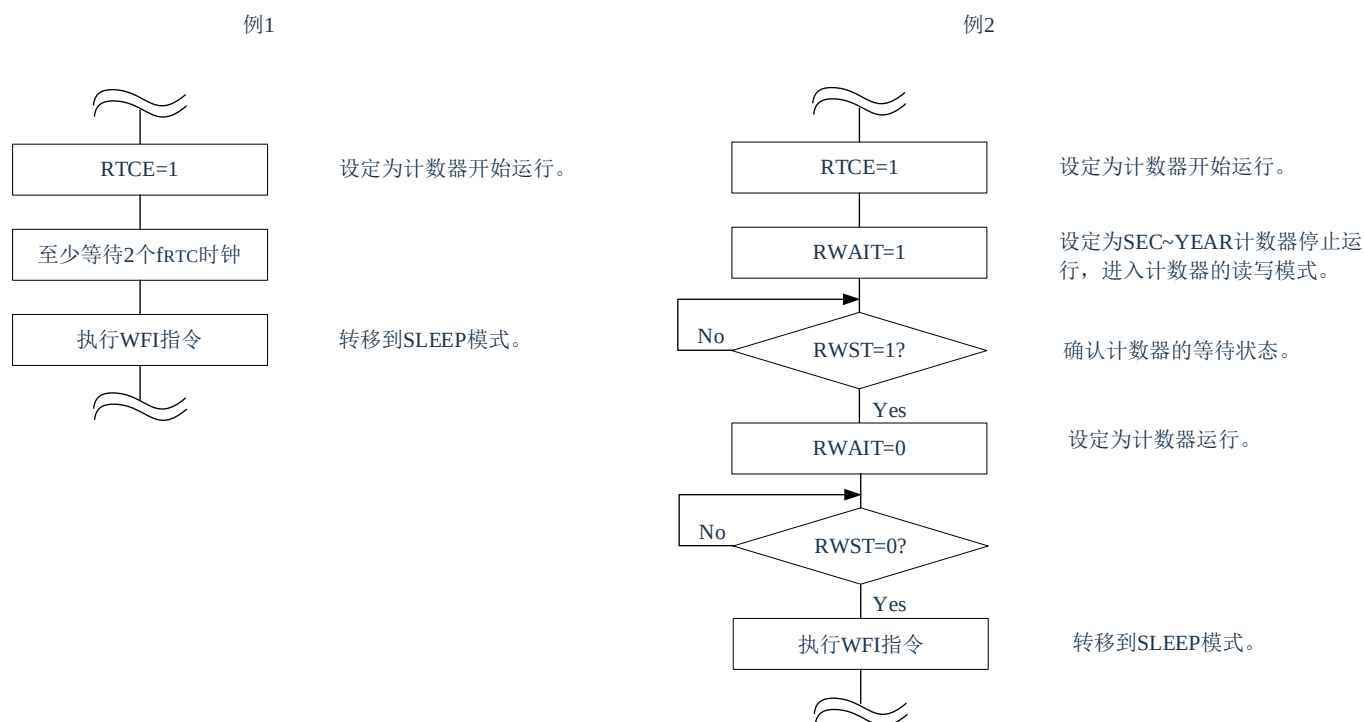
注2：在RTCE位为“1”后不等待INTRTC位变为“1”而转移到睡眠模式的情况下，请确认“8.5.2 开始运行后睡眠模式的转移”的步骤。

8.5.2 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8-3的例1）。
- 在将RTCE位置“1”后将RWAIT位置“1”，通过轮询确认RWST位变为“1”。然后，将RWAIT位置“0”并且再次通过轮询确认RWST位变为“0”，然后转移到睡眠模式（参照图8-3的例2）。

图8-3 将RTCE位置“1”后的睡眠/深度睡眠模式的转移步骤



8.5.3 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8-4 实时时钟计数器的读操作步骤



注1：必须在转移到睡眠模式前，确认RWST位为“0”。

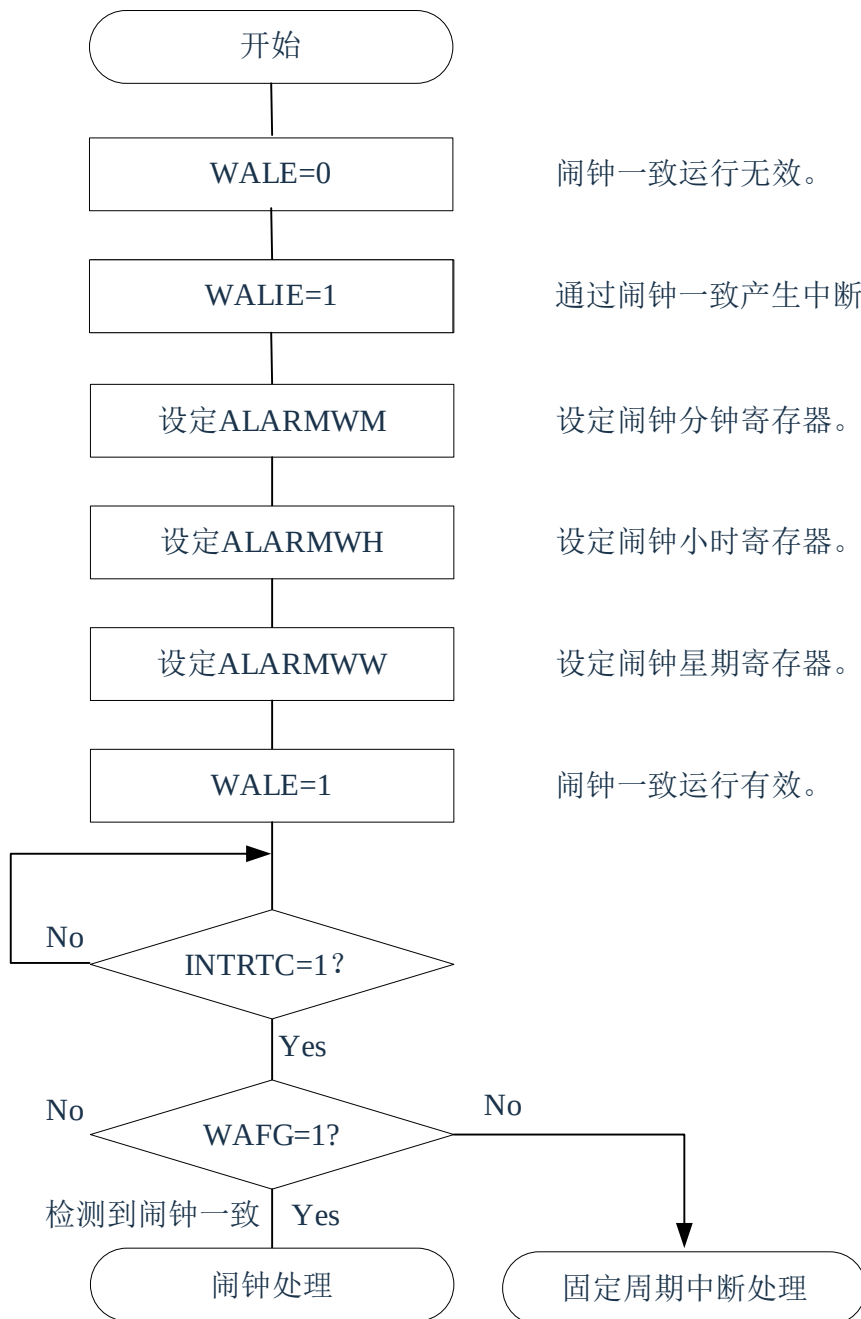
注2：必须在1秒内进行将RWAIT位置“1”到RWAIT位置“0”的处理。

注3：不限制秒/分钟/小时/星期/日/月/和年计数寄存器的读操作顺序。可以不读全部寄存器而只读部分寄存器。

8.5.4 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8-5 闹钟设置步骤



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

注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。

第9章 时钟输出/蜂鸣器输出控制电路

9.1 时钟输出/蜂鸣器输出控制电路的功能

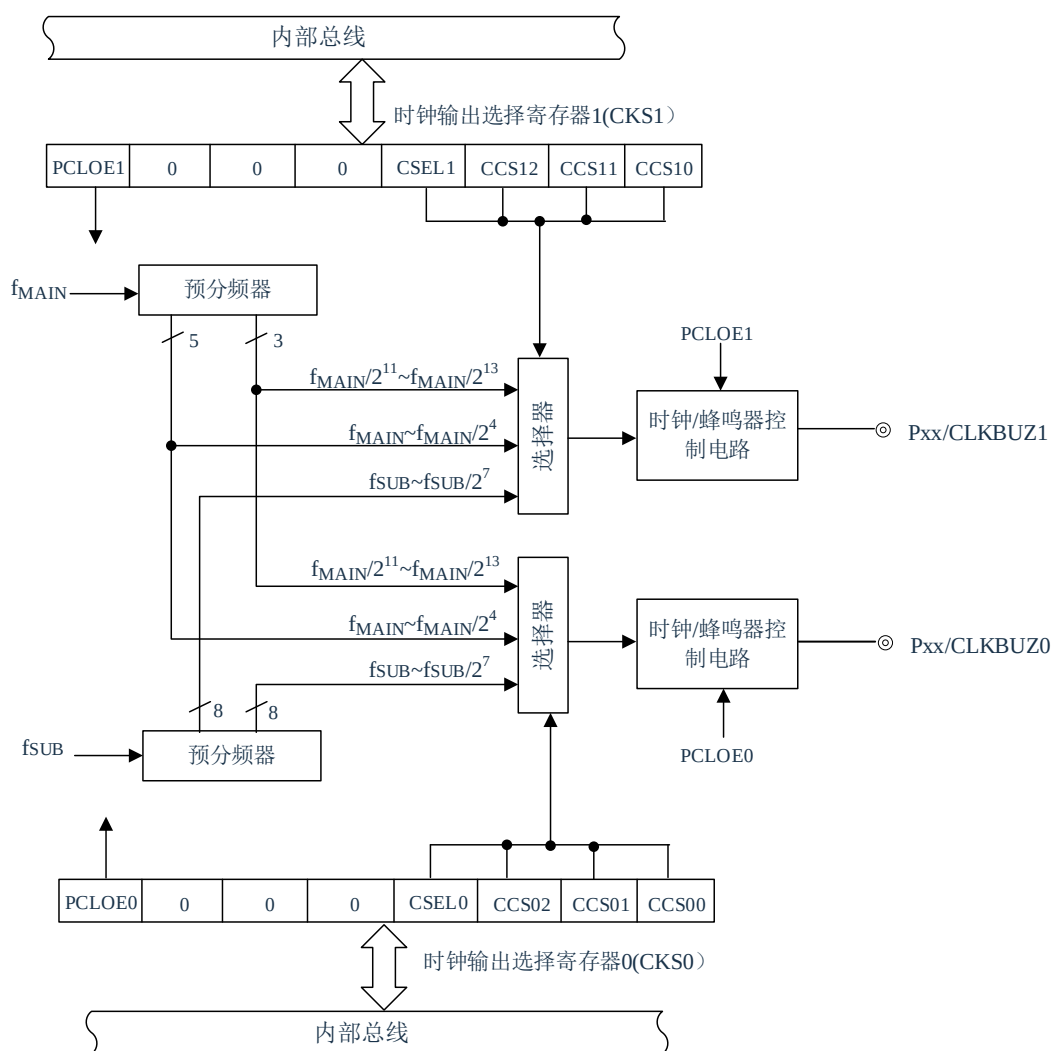
时钟输出是输出提供给外围 IC 时钟的功能，蜂鸣器输出是输出蜂鸣器频率方波的功能。

本产品有两个时钟输出/蜂鸣器输出引脚，其中 CLKBUZ0 能从 PA0、PA3、PB0、PB10、PC0、PC10、PD0、PD3、PD11、PD10、PD15 中选择引脚用作时钟输出或者蜂鸣器输出，CLKBUZ1 能从 PA1、PA4、PB1、PB11、PC1、PC11、PD1、PD4、PD12 中选择引脚用作时钟输出或者蜂鸣器输出。

CLKBUZn 引脚输出由时钟输出选择寄存器 n（CKSn）选择的时钟。

时钟输出/蜂鸣器输出控制电路的框图如图 9-1 所示。

图9-1 时钟输出/蜂鸣器输出控制电路的框图



注1：有关能从CLKBUZ0引脚和CLKBUZ1引脚输出的频率，请参照“数据手册的AC特性”。

注2：n=0、1。

9.2 时钟输出/蜂鸣器输出控制电路的结构

时钟输出/蜂鸣器输出控制电路由以下硬件构成。

表9-1 时钟输出/蜂鸣器输出控制电路的结构

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

9.3 寄存器映射

(CKS0/1 基地址 = 0x4004_0FA5) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CKS0	0x000	R/W	时钟输出选择寄存器0	0x0
CKS1	0x001	R/W	时钟输出选择寄存器1	0x0

9.3.1 时钟输出选择寄存器（CKS0）

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（CLKBUZ0）的输出以及设定输出时钟的寄存器。通过 CKS0 寄存器选择 CLKBUZ0 引脚输出的时钟。通过 8 位存储器操作指令设定 CKS0 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	PCLOE0	CLKBUZ0引脚输出允许/禁止的指定 0: 禁止输出（默认值） 1: 允许输出	0
6:4	--	保留	--
3	CSEL0	CLKBUZ0引脚输出时钟的选择	0
2	CCS02	CLKBUZ0引脚输出时钟的选择	0
1	CCS01	CLKBUZ0引脚输出时钟的选择	0
0	CCS00	CLKBUZ0引脚输出时钟的选择	0

具体CLKBUZ0引脚输出时钟的选择如下表：

CSEL0	CCS02	CCS01	CCS00	CLKBUZ0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1：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2：输出时钟的切换必须在设定为禁止输出（PCLOE0=0）后进行，在选择副系统时钟（CSEL0=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PCLOE0“1”

注3：在选择主系统时钟（CSEL0=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PCLOE0置“0”

注4： f_{MAIN} ：主系统时钟频率， f_{SUB} ：副系统时钟频率

注5：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CLKBUZ0脚输出副系统时钟（ f_{SUB} ）

9.3.2 时钟输出选择寄存器（CKS1）

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（CLKBUZ1）的输出以及设定输出时钟的寄存器。通过 CKS1 寄存器选择 CLKBUZ1 引脚输出的时钟。通过 8 位存储器操作指令设定 CKS1 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	PCLOE1	CLKBUZ1引脚输出允许/禁止的指定 0: 禁止输出（默认值） 1: 允许输出	0
6:4	--	保留	--
3	CSEL1	CLKBUZ1输出时钟的选择	0
2	CCS12	CLKBUZ1输出时钟的选择	0
1	CCS11	CLKBUZ1输出时钟的选择	0
0	CCS10	CLKBUZ1输出时钟的选择	0

具体CLKBUZ1引脚输出时钟的选择如下表：

CSEL1	CCS12	CCS11	CCS10	CLKBUZn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1：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2：输出时钟的切换必须在设定为禁止输出（PCLOE1=0）后进行，在选择副系统时钟（CSEL1=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PCLOE1“1”。

注3：在选择主系统时钟（CSEL1=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PCLOE1“0”。

注4： f_{MAIN} ：主系统时钟频率， f_{SUB} ：副系统时钟频率

注5：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CLKBUZ1脚输出副系统时钟（ f_{SUB} ）。

9.4 配置时钟输出/蜂鸣器输出引脚端口功能的寄存器

本产品有两个时钟输出/蜂鸣器输出引脚，其中 CLKBUZ0 能从 PA0、PA3、PB0、PB10、PC0、PC10、PD0、PD3、PPD11、PD10、PD15 中选择引脚用作时钟输出或者蜂鸣器输出，CLKBUZ1 能从 PA1、PA4、PB1、PB11、PC1、PC11、PD1、PD4、PD12 中选择引脚用作时钟输出或者蜂鸣器输出。

使用时钟输出/蜂鸣器输出功能时，必须设定端口复用功能配置寄存器（PS_BUZx），寄存器（Pxx），端口模式寄存器（PMxx）和端口模式控制寄存器（PMCxx）。详细内容请参照“第2章 引脚功能”。

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

（例）将 PA0 用作时钟输出/蜂鸣器输出（CLKBUZ0）的情况：

将端口寄存器 1 的 PA0 位置 “0”。

将端口模式寄存器 1 的 PMA0 位置 “0”。

将端口模式控制寄存器 1 的 PMCA0 位置 “0”。

将端口复用功能配置寄存器 PS_BUZ0 置 “0x00”。

（例）将 PA1 用作时钟输出/蜂鸣器输出（CLKBUZ1）的情况：

将端口寄存器 1 的 PA1 位置 “0”。

将端口模式寄存器 1 的 PMA1 位置 “0”。

将端口模式控制寄存器 1 的 PMCA1 位置 “0”。

将端口复用功能配置寄存器 PS_BUZ1 置 “0x00”。

9.5 时钟输出/蜂鸣器输出控制电路的运行

相对应的端口在复用后可做时钟输出或蜂鸣器输出。

CLKBUZ0 引脚输出由时钟输出选择寄存器 0 (CKS0) 选择的时钟/蜂鸣器。

CLKBUZ1 引脚输出由时钟输出选择寄存器 1 (CKS1) 选择的时钟/蜂鸣器。

9.5.1 输出引脚的运行

相对应的端口在复用后可用作时钟输出或者蜂鸣器输出。

设定端口复用功能配置寄存器 (PS_BUZn), 将用作 CLKBUZn 引脚的端口对应的端口寄存器 (Pxx), 端口模式寄存器 (PMxx) 和端口模式控制寄存器 (PMCxx) 的位置“0”。

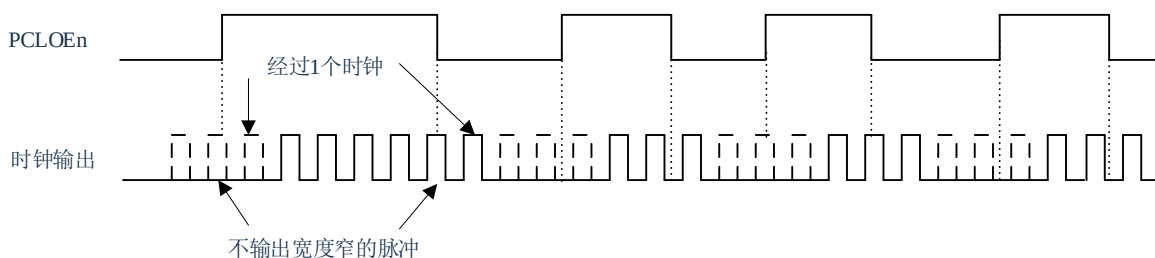
通过 CLKBUZn 引脚的时钟输出选择寄存器 (CKSn) 的 bit0~3 (CCSn0~CCSn2、CSELn) 选择输出频率 (输出为禁止状态)。

将 CKSn 寄存器的 bit7 (PCLOEn) 置“1”, 允许时钟/蜂鸣器的输出。

注1: 用作时钟输出时的控制电路在允许或者禁止时钟输出 (PCLOEn位) 后的1个时钟之后, 开始或者停止时钟输出。此时不输出宽度窄的脉冲。通过PCLOEn位允许或者停止输出以及时钟输出的时序如图9-2所示。

注2: n=0、1

图9-2 CLKBUZn引脚的时钟输出时序



9.6 时钟输出/蜂鸣器输出控制电路的注意事项

当选择主系统时钟作为 CLKBUZn 输出 (CSELn=0) 时, 如果在设定停止输出 (PCLOEn=0) 后的 1.5 个 CLKBUZn 引脚的输出时钟内转移到深度睡眠模式, CLKBUZn 的输出宽度就变窄。

第10章 独立看门狗定时器 (IWDT)

10.1 概述

10.1.1 独立看门狗定时器概述

独立看门狗定时器通过选项字节 (000C0H) 设定计数运行。独立看门狗定时器以低速内部振荡器时钟 (f_{IL}) 运行。

10.2 特性

10.2.1 独立看门狗定时器特性

独立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：

- 当独立看门狗定时器的计数器发生上溢时
- 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

当因独立看门狗定时器而发生复位时，将复位控制标志寄存器 (RESF) 的 bit5 (IWDTRF) 置“1”。有关 RESF 寄存器的详细内容，请参照“第 24 章 复位功能”。当达到上溢时间的 $75\% + 1/2f_{IL}$ 时，能产生间隔中断。

10.3 独立看门狗定时器结构

独立看门狗定时器由以下硬件构成。

表10-1 独立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	独立看门狗定时器的允许寄存器（WDTE）

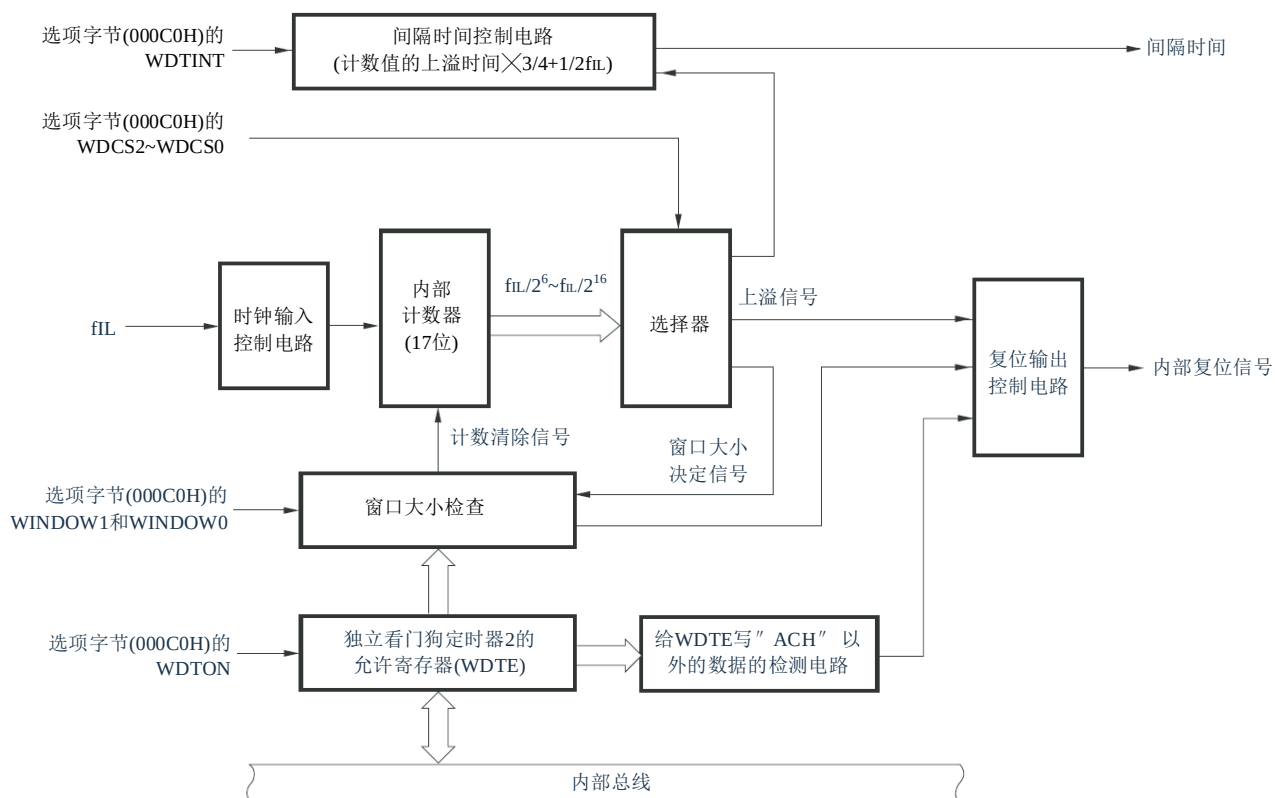
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表10-2 选项字节和独立看门狗定时器的设定内容

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

注：有关选项字节，请参照“第28章 选项字节”。

图10-1 独立狗定时器的框图



注: f_{IL} : 低速内部振荡器的时钟频率

10.4 寄存器映射

(WDTE 基地址 = 0x4002_1001) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
WDTE	0x000	R/W	独立看门狗定时器的允许寄存器	0x1A/0x9A

(LOCKCTL基地址 = 0x4002_0405) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LOCKCTL	0x000	R/W	控制寄存器	0x1

(PRCR基地址 = 0x4002_0406) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PRCR	0x000	R/W	保护寄存器	0x0

10.5 寄存器说明

10.5.1 独立看门狗定时器的允许寄存器 (WDTE)

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

位	符号	描述	复位值
7:0	WDTE	写0xAC清除独立看门狗定时器的计数器并且重新开始计数	0x1A/0x9A

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

WDTON位的设定值	WDTE寄存器的复位值
0（禁止独立看门狗定时器的计数运行）	1AH
1（允许独立看门狗定时器的计数运行）	9AH

注1：当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。

注2：WDTE寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

10.5.2 LOCKUP控制寄存器 (LOCKCTL)

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否引起独立看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 LOCKCTL 寄存器。

在产生复位信号后，LOCKCTL 寄存器的值变为“01H”。

位	符号	描述	复位值
7:1	-	保留	-
0	Lockup_rst	LOCKUP功能的配置 0: LOCKUP不导致IWDG复位 1: LOCKUP导致IWDG复位	1

10.5.3 保护寄存器 (PRCR)

LOCKCTL 寄存器是 Cortex-M0+ LockUp 功能是否引起独立看门狗定时器运行的配置寄存器，PRCR 是其写保护寄存器。

通过 8 位存储器操作指令设置 PRCR 寄存器。

在产生复位信号后 PRCR 寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	PRTKEY	PRCR的写保护 78H: PRCR可写 其他: PRCR不可写	0x0
0	PRCR	LOCKUP控制寄存器写保护 0: LOCKCTL寄存器不可写 1: LOCKCTL寄存器可写	0

10.6 独立看门狗定时器的运行

10.6.1 独立看门狗定时器的运行控制

- 当使用独立看门狗定时器时，通过选项字节（000C0H）设定以下内容：
 - 必须将选项字节（000C0H）的 bit4（WDTON）置“1”，允许独立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 28 章 选项字节）。

WDTON	独立看门狗定时器的计数器
0	禁止计数运行（解除复位后停止计数）。
1	允许计数运行（解除复位后开始计数）。

- 必须通过选项字节（000C0H）的 bit3~1（WDCS2~WDCS0）设定上溢时间（详细内容请参照 10.6.2 和第 28 章）。
 - 必须通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照 10.6.3 和第 28 章）。
- 在解除复位后，独立看门狗定时器开始计数。
 - 在开始计数后并且在选项字节所设上溢时间前，如果给独立看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除独立看门狗定时器并且重新开始计数。
 - 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。
 - 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：
 - 当给 WDTE 寄存器写“ACH”以外的数据时

注 1：只在解除复位后第 1 次写独立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，就清除独立看门狗定时器且重新开始计数。

注 2：从给 WDTE 寄存器写“ACH”到清除独立看门狗定时器的计数器为止，有可能产生最大 2 个 fIL 时钟的误差。

注 3：在计数值发生上溢前，都能清除看门狗定时器。

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

	WDSTBYON=0	WDSTBYON=1
睡眠模式	停止独立看门狗定时器运行。	继续独立看门狗定时器运行。
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独立看门狗定时器的计数。此时，将计数器清“0”，开始计数。

如果从解除深度睡眠模式到独立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。

10.6.2 独立看门狗定时器溢时间的设定

通过选项字节 (000C0H) 的 bit3~1 (WDCS2~WDCS0) 设定独立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独立看门狗定时器的允许寄存器 (WDTE) 写“ACH”，就清除计数并且重新开始计数。能设定的上溢时间如下所示。

表10-3 独立看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	独立看门狗定时器的上溢时间 ($f_{IL}=32\text{kHz}$ 的情况)
0	0	0	$2^6/f_{IL}$ (2ms)
0	0	1	$2^7/f_{IL}$ (4ms)
0	1	0	$2^8/f_{IL}$ (8ms)
0	1	1	$2^9/f_{IL}$ (16ms)
1	0	0	$2^{11}/f_{IL}$ (64ms)
1	0	1	$2^{13}/f_{IL}$ (256ms)
1	1	0	$2^{14}/f_{IL}$ (512ms)
1	1	1	$2^{16}/f_{IL}$ (2048ms)

备注 f_{IL} : 低速内部振荡器的时钟频率

10.6.3 独立看门狗定时器窗口打开期间的设定

通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定独立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独立看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除独立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，就清除独立看门狗定时器并且重新开始计数。

能设定的窗口打开期间如下所示。

表10-4 独立看门狗定时器窗口打开期间的设定

WINDOW1	WINDOW0	独立看门狗定时器的窗口打开期间
0	-	禁止设定
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%。

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

	窗口打开期间的设定	
	75%	100%
窗口关闭时间	0~8ms	无
窗口打开时间	0~10.6ms	0~10.6ms

<当窗口打开期间为75%时>

上溢时间：

$$2^9/f_{IL}(\text{MAX.})=2^9/48\text{kHz}(\text{MAX.})=10.6\text{ms}$$

窗口关闭时间：

$$0\sim 2^9/f_{IL}(\text{MIN.})\times(1-0.75)=0\sim 2^9/16\text{kHz}\times 0.25=0\sim 8\text{ms}$$

窗口打开时间：

$$2^9/f_{IL}(\text{MIN.})\times(1-0.75)\sim 2^9/f_{IL}(\text{MAX.})=8\sim 10.6\text{ms}$$

10.6.4 独立看门狗定时器间隔中断的设定

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

表10-5独立看门狗定时器间隔中断的设定

WDTINT	独立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独立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。

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

10.6.5 LOCKUP期间独立看门狗定时器的运行

当 LOCKUP 控制寄存器 LOCKCTL 的 lockup_rst 位设置为 1 时，一旦内核进入 LOCKUP 状态，低速内部振荡器开始振荡，独立看门狗定时器的计时器自动开始运行，并将上溢时间的控制位 (WDCS2~WDCS0) 设置为 3'b010，即设置上溢时间为 8ms。

第11章 窗口看门狗定时器（WWDT）

11.1 窗口看门狗定时器的功能

窗口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：

- 当窗口看门狗定时器的计数器发生上溢时
- 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

当因窗口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（WWDTRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 24 章 复位功能”。当达到上溢时间的 $75\% + 1/2f_{\text{wwdt}}$ 时，能产生间隔中断。

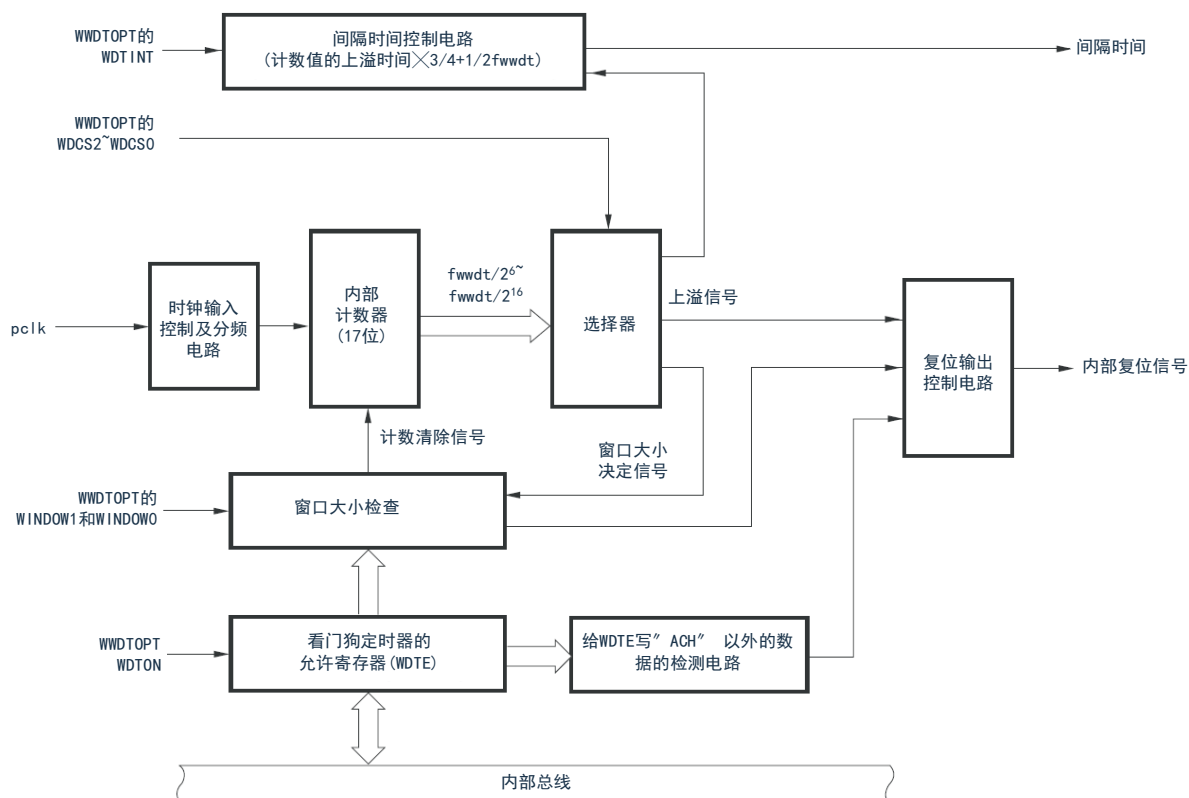
11.2 窗口看门狗定时器的结构

窗口看门狗定时器由以下硬件构成。

表11-1 窗口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	窗口看门狗定时器的清零寄存器（WDTE）
	窗口看门狗定时器的控制寄存器（WWDTOPT）
	窗口看门狗定时器的时钟分频寄存器（WWDTDIV）

图11-1 窗口看门狗定时器的框图



注: f_{wwdt} : 计数时钟频率, 详见11.3.3

11.3 寄存器映射

(窗口看门狗定时器基地址 = 0x4002_1020) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
WDTE	0x000	R/W	窗口看门狗定时器的清零寄存器	0x1A
WWDTOPT	0x002	R/W	窗口看门狗定时器的控制寄存器	0x00
WWDTDIV	0x004	R/W	窗口看门狗定时器的分频寄存器	0x00

11.3.1 窗口看门狗定时器的清零寄存器 (WDTE)

通过给 WDTE 寄存器写“ACH”，清除窗口看门狗定时器的计数器并且重新开始计数。在产生复位信号后，此寄存器的值变为“1AH”。

位	符号	描述	复位值
7:0	WDTE	写0xAC清除窗口看门狗定时器的计数器并且重新开始计数	0x1A

注：WDTE寄存器的读取值因WWDTOPT的WDTON位的设定值而不同。要使窗口看门狗定时器运行时，必须将WDTON位置“1”。

WDTON位的设定值	WDTE寄存器的读取值
0（禁止窗口看门狗定时器运行）	1AH
1（允许窗口看门狗定时器运行）	9AH

注1：当给WDTE寄存器写“ACH”以外的值时，产生内部复位信号。

注2：WDTE寄存器的读取值为“9AH/1AH”（和写入值（“ACH”）不同）。

11.3.2 窗口看门狗控制寄存器(WWDTOPT)

位	符号	描述	复位值
7	WDTINT	窗口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_{wwdt} 时, 产生间隔中断	0
6:5	WINDOW[1:0]	窗口看门狗定时器的窗口打开期间 0X: 禁止设定。 10: 75% 11: 100%	0x0
4	WDTON	窗口看门狗定时器运行控制 0: 禁止窗口看门狗的运行 1: 允许窗口看门狗运行	0
3:1	WDTC[2:0]	窗口看门狗定时器的上溢时间 000: $2^6/f_{wwdt}$ 001: $2^7/f_{wwdt}$ 010: $2^8/f_{wwdt}$ 011: $2^9/f_{wwdt}$ 100: $2^{11}/f_{wwdt}$ 101: $2^{13}/f_{wwdt}$ 110: $2^{14}/f_{wwdt}$ 111: $2^{16}/f_{wwdt}$	0x0
0	WDSTBYON	窗口看门狗定时器的计数器运行控制 (睡眠模式) 0: 在睡眠模式中, 停止计数器的运行 1: 在睡眠模式中, 允许计数器的运行	0

注1: f_{wwdt} 为窗口看门狗计数时钟, 详细描述见11.3.3

注2: 在窗口看门狗运行时给WDTON写0会使窗口看门计数值清零

11.3.3 窗口看门狗分频寄存器(WWDTDIV)

位	符号	描述	复位值
7:3	--	保留	0
2:0	WWDT_DIV[2:0]	WWDT计数时钟(f_{wwdt})选择 000: 计数时钟为fclk/16 001: 计数时钟为fclk/32 010: 计数时钟为fclk/64 011: 计数时钟为fclk/128 100: 计数时钟为fclk/256 101: 计数时钟为fclk/512 110: 计数时钟为fclk/1024 111: 计数时钟为fclk/2048	0x0

11.4 窗口看门狗定时器的运行

11.4.1 窗口看门狗定时器的运行控制

- 当使用窗口看门狗定时器时，通过 WWDTOPT 设定以下内容：
 - 必须将 WWDTOPT 的 bit4（WDTON）置“1”，允许窗口看门狗定时器的计数运行

WDTON	窗口看门狗定时器的计数器
0	禁止计数运行并清除计数值
1	允许计数运行

- 必须通过 WWDTOPT 的 bit3～1（WDCS2～WDCS0）设定上溢时间（详细内容请参照 11.4.2）。
 - 必须通过 WWDTOPT 的 bit6 和 bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照 11.4.3）。
- 在 WDTON 置 1 后，窗口看门狗定时器开始计数。
 - 在开始计数后并且在所设上溢时间前，如果给窗口看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除窗口看门狗定时器并且重新开始计数。
 - 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。
 - 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：
 - 当给 WDTE 寄存器写“ACH”以外的数据时

注 1：只在解除复位后第 1 次写窗口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何时候写 WDTE，就清除窗口看门狗定时器并且重新开始计数。

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

注 3：在计数值发生上溢前，都能清除窗口看门狗定时器。

注 4：如下所示，窗口看门狗定时器在睡眠模式中的运行 WWDTOPT 的 bit0（WDSTBYON）的设定值而不同。

	WDSTBYON=0	WDSTBYON=1
睡眠模式	停止窗口看门狗定时器运行。	继续窗口看门狗定时器运行。
深度睡眠模式	停止窗口看门狗定时器运行。	停止窗口看门狗定时器运行。

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

11.4.2 窗口看门狗定时器上溢时间的设定

通过 WWDTOPT 的 bit3~1 (WDCS2~WDCS0) 设定窗口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窗口看门狗定时器的允许寄存器 (WDTE) 写“ACH”，就清除计数并且重新开始计数。能设定的上溢时间如下所示。

表11-2 窗口看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	窗口看门狗定时器的上溢时间
0	0	0	$2^6 / f_{\text{wwdt}}$
0	0	1	$2^7 / f_{\text{wwdt}}$
0	1	0	$2^8 / f_{\text{wwdt}}$
0	1	1	$2^9 / f_{\text{wwdt}}$
1	0	0	$2^{11} / f_{\text{wwdt}}$
1	0	1	$2^{13} / f_{\text{wwdt}}$
1	1	0	$2^{14} / f_{\text{wwdt}}$
1	1	1	$2^{16} / f_{\text{wwdt}}$

11.4.3 窗口看门狗定时器窗口打开期间的设定

通过 WWDTOPT 的 bit6 和 bit5 (WINDOW1、WINDOW0) 设定窗口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窗口看门狗定时器的允许寄存器 (WDTE) 写“ACH”，就清除窗口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，就清除窗口看门狗定时器并且重新开始计数。

能设定的窗口打开期间如下所示。

表11-3 窗口看门狗定时器窗口打开期间的设定

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

注意:当WWDTOPT的bit0 (WDSTBYON) 为“0”时，与WINDOW1位和WINDOW0位的值无关，窗口打开期间为100%。

当将上溢时间设定为 $2^{10}/f_{\text{wwdt}}$,WWDTDIV=0x05情况时，窗口关闭时间和打开时间如下所示。

	窗口打开期间的设定	
	75%	100%
窗口关闭时间	0~2.048ms	无
窗口打开时间	2.048~8.192ms	0~8.192ms

<当窗口打开期间为75%时>

上溢时间：

$$2^{10}/f_{\text{WWDT}}=2^{10}/125\text{kHz}=8.192\text{ms}$$

窗口关闭时间：

$$0\sim 2^{10}/f_{\text{WWDT}}\times(1-0.75)=0\sim 2^{10}/125\text{kHz}\times 0.25=2.048\text{ms}$$

窗口打开时间：

$$2^{10}/f_{\text{WWDT}}\times(1-0.75)\sim 2^{10}/125\text{kHz}=2.048\sim 8.192\text{ms}$$

11.4.4 窗口看门狗定时器间隔中断的设定

能通过 WWDTOPT 的 bit7 (WDTINT)，在达到上溢时间的 $75\% + 1/2f_{\text{wwdt}}$ 时产生间隔中断 (INTWWDT)。

表11-4 窗口看门狗定时器间隔中断的设定

WDTINT	窗口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 $75\% + 1/2f_{\text{wwdt}}$ 时，产生间隔中断。

备注：即使在产生窗口看门狗定时器间隔中断后也继续计数（继续到给窗口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12章 EPWM输出控制电路

12.1 输出控制电路的功能

使用通用定时器的PWM输出功能，实现一个直流电机或者两个步进电机的控制。通过截断源ACMP0/1的输出，INTP0的输入以及EVENTC事件输出，能截断输出。通过软件的设定，能从强制截断时进行Hi-Z输出、低电平输出、高电平输出以及禁止截断输出（正常输出）的4种输出中进行选择。

12.2 输出控制电路的结构

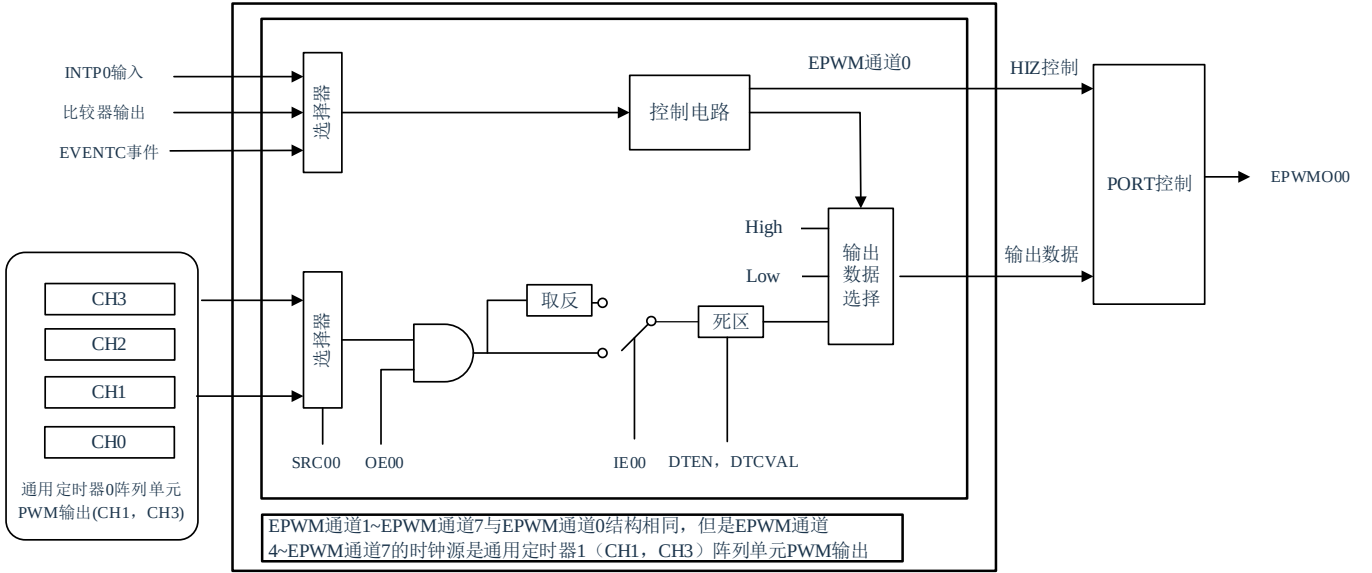
EPWM输出控制电路由以下硬件构成。

表12-1 EPWM的输出控制电路的结构

项目	结构
控制寄存器	EPWM输入源选择寄存器(EPWMSRC)
	EPWM输出控制寄存器(EPWMCTL)
	EPWM强制截断输入选择寄存器(EPWMSTC)
	EPWM强制截断输出选择寄存器(EPWMSTL)
	EPWM状态寄存器(EPWMSTR)
	EPW死区控制寄存器(EPWMDTC0, EPWMDTC1)
输出	EPWM输出(EPWMO00~EPWMO07)

EPWM输出控制电路的框图如图12-1所示。

图12-1 EPWM输出控制电路的框图



12.3 寄存器映射

(EPWM基地址 = 0x40044400) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
EPWMSRC	0x00	R/W	EPWM输入源选择寄存器	0x00
EPWMSTC	0x04	R/W	EPWM强制截断输入选择寄存器	0x00
EPWMCTL	0x08	R/W	EPWM输出控制寄存器	0x00
EPWMSTL	0x0C	R/W	EPWM强制截断输出选择寄存器	0x00
EPWMSTR	0x10	R/W	EPWM状态寄存器	0x00
EPWMDTC0	0x14	R/W	EPWM死区控制寄存器0	0x00
EPWMDTC1	0x18	R/W	EPWM死区控制寄存器1	0x00

12.3.1 EPWM输入源选择寄存器(EPWMSRC)

EPWMSRC寄存器选择实时输出电路的输入时钟的源时钟。EPWM的通道0-3可选择通用定时器的输出TO01或者TO03作为源时钟，EPWM的通道4-7可选择通用定时器的定时器输出TO11或者TO13作为源时钟，并且输入到EPWM。

通过8位存储器操作指令设定EPWMSRC寄存器。

通过产生复位信号，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	SRC07	EPWM07输出时钟源选择 0: 选择TO11 1: 选择TO13	0
6	SRC06	EPWM06输出时钟源选择 0: 选择TO11 1: 选择TO13	0
5	SRC05	EPWM05输出时钟源选择 0: 选择TO11 1: 选择TO13	0
4	SRC04	EPWM04输出时钟源选择 0: 选择TO11 1: 选择TO13	0
3	SRC03	EPWM03输出时钟源选择 0: 选择TO01 1: 选择TO03	0
2	SRC02	EPWM02输出时钟源选择 0: 选择TO01 1: 选择TO03	0
1	SRC01	EPWM01输出时钟源选择 0: 选择TO01 1: 选择TO03	0
0	SRC00	EPWM00输出时钟源选择 0: 选择TO01 1: 选择TO03	0

12.3.2 EPWM输出控制寄存器(EPWMCTL)

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

通过16位存储器操作指令设定EPWMCTL寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15	IE07	EPWMO07输出的取反控制 0: 不取反 1: 取反	0
14	IE06	EPWMO06输出的取反控制 0: 不取反 1: 取反	0
13	IE05	EPWMO05输出的取反控制 0: 不取反 1: 取反	0
12	IE04	EPWMO04输出的取反控制 0: 不取反 1: 取反	0
11	IE03	EPWMO03输出的取反控制 0: 不取反 1: 取反	0
10	IE02	EPWMO02输出的取反控制 0: 不取反 1: 取反	0
9	IE01	EPWMO01输出的取反控制 0: 不取反 1: 取反	0
8	IE00	EPWMO00输出的取反控制 0: 不取反 1: 取反	0
7	OE07	EPWMO07输出使能控制 0: 禁止输出 1: 使能输出	0
6	OE06	EPWMO06输出使能控制 0: 禁止输出 1: 使能输出	0
5	OE05	EPWMO05输出使能控制 0: 禁止输出 1: 使能输出	0
4	OE04	EPWMO04输出使能控制 0: 禁止输出 1: 使能输出	0
3	OE03	EPWMO03输出使能控制 0: 禁止输出 1: 使能输出	0
2	OE02	EPWMO02输出使能控制 0: 禁止输出 1: 使能输出	0
1	OE01	EPWMO01输出使能控制 0: 禁止输出 1: 使能输出	0
0	OE00	EPWMO00输出使能控制 0: 禁止输出 1: 使能输出	0

12.3.3 EPWM强制截断输入选择寄存器(EPWMSTC)

EPWMSTC寄存器进行强制截断输入源的选择。

通过8位存储器操作指令设定EPWMSTC寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	SHUT_EN	截断使能控制 0: 截断禁止 1: 截断使能	0
6:5	-	保留，必须为0	0x0
4	REL_SEL	输出强制截断的解除时序选择 0: 通过硬件或软件产生的解除信号发生后，截断立即解除，脉冲输出恢复 1: 通过硬件或软件产生的解除信号发生后，等待下列时序： 选择TO01、TO03为源时钟的通道：在下一个TO01、TO03的边沿时截断解除，脉冲输出恢复； 选择TO11、TO13为源时钟的通道：在下一个TO11、TO13的边沿时截断解除，脉冲输出恢复；	0
3	HS_SEL	输出强制截断的模式选择 0: 软件解除 1: 硬件解除	0
2	IN_EG	输出强制截断的源/输出强制截断解除的源的沿的选择 0: 上升沿：输出强制截断 下降沿：输出强制截断解除 1: 上升沿：输出强制截断解除 下降沿：输出强制截断	0
1:0	SC_SEL[1:0]	截断源的选择 00: ACMP0的输出 01: ACMP1的输出 10: INTP0端子输入 11: 来自EVENTC的事件输入	0x0

注1：在截断相关位设置后至少间隔三个时钟，再设置SHUT_EN。

注2：IN_EG只在选择INTP0输入，ACMP0/1输出时有效。

注3：使用EVENTC解除强制截止时，必须选择软件解除（HS_SEL置为0）。

注4：选择ACMP0/1，INTP0输入的有效宽度必须大于一个时钟周期。

注5：bit7位 SHUT_EN为只写位。

12.3.4 EPWM强制截断输出选择寄存器(EPWMSTL)

EPWMSTL寄存器进行强制截断时EPWMO端口输出状态。

通过16位存储器操作指令设定EPWMSTL寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:14	IO7[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
13:12	IO6[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
11:10	IO5[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
9:8	IO4[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
7:6	IO3[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
5:4	IO2[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
3:2	IO1[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0
1:0	IO0[1:0]	截断时端口输出的选择 00: 正常输出 01: HI-Z输出 10: 低电平输出 11: 高电平输出	0x0

12.3.5 EPWM状态寄存器(EPWMSTR)

EPWMSTR寄存器可清除强制截断信号以及显示截断状态。如果将清除触发位HZCLR置“1”，就解除截断状态。当截断状态标志SHTFLG的信号为高电平时，就进入强制截断状态。Bit0为只写位，读取值为“0”。Bit8~1为只读位。

通过16位存储器操作指令设定EPWMSTR寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:9	-	保留，必须保持复位值	0x0
8:1	SHTFLG[7:0]	EPWM通道00 - EPWM通道07强制截断状态标志 0: 通常输出状态 1: 强制截断状态	0x0
0	HZCLR	软件清除强制截断信号 0: 无效 1: 写软件解除截断状态	0

12.3.6 EPWM死区控制寄存器0(EPWMDTC0)

EPWMDTC0寄存器控制EPWMO00-EPWMO03的波形输出的死区使能和死区数据控制。

通过16位存储器操作指令设定EPWMDTC的值。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:8	DTCVAL0	EPWMO0-EPWMO3输出死区时间（4个通道公用）	0x0
7:4	-	保留	0x0
3	EN3	EPWMO03输出死区使能 0: 禁止 1: 使能	0
2	EN2	EPWMO02输出死区使能 0: 禁止 1: 使能	0
1	EN1	EPWMO01输出死区使能 0: 禁止 1: 使能	0
0	EN0	EPWMO00输出死区使能 0: 禁止 1: 使能	0

12.3.7 EPWM死区控制寄存器1(EPWMDTC1)

EPWMDTC1寄存器控制EPWMO04-EPWMO07的波形输出的死区使能和死区数据控制。

通过16位存储器操作指令设定EPWMDTC的值。

在产生复位信号后，此寄存器的值变为“0000H”。

位	符号	描述	复位值
15:8	DTCVAL1	EPWMO4-EPWMO7输出死区时间（4个通道公用）	0x0
7	EN7	EPWMO07输出死区使能 0: 禁止 1: 使能	0
6	EN6	EPWMO06输出死区使能 0: 禁止 1: 使能	0
5	EN5	EPWMO05输出死区使能 0: 禁止 1: 使能	0
4	EN4	EPWMO04输出死区使能 0: 禁止 1: 使能	0
3:0	-	保留	0x0

12.3.8 EPWM输出引脚的端口功能的控制寄存器

使用EPWM输出时，必须设定与EPWM输出引脚（EPWMO_n引脚）复用的端口功能的控制寄存器（端口模式寄存器（PM_{xx}，PMC_{xx}）。详细内容请参照第2章引脚功能。

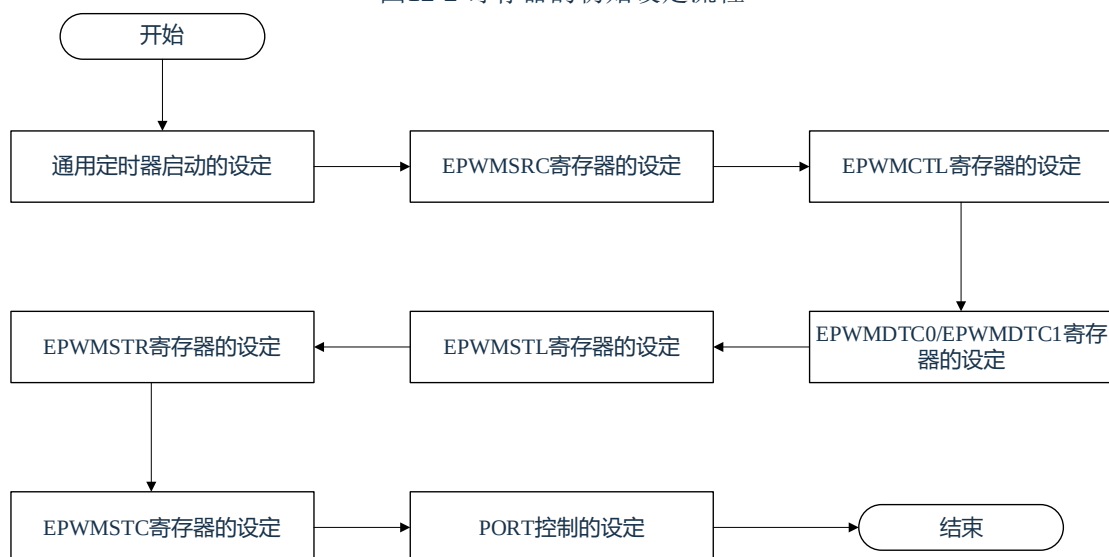
12.4 EPWM输出控制电路的运行

12.4.1 初始设定

定时器波形通过EPWSRC寄存器选择通用定时器0/通用定时器1的输出（TO01、TO03、TO11、TO13）作为源时钟。能通过设定EPWMCTL寄存器，设定定时器波形的正相或者反相，固定低电平或者高电平。

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

图12-2 寄存器的初始设定流程

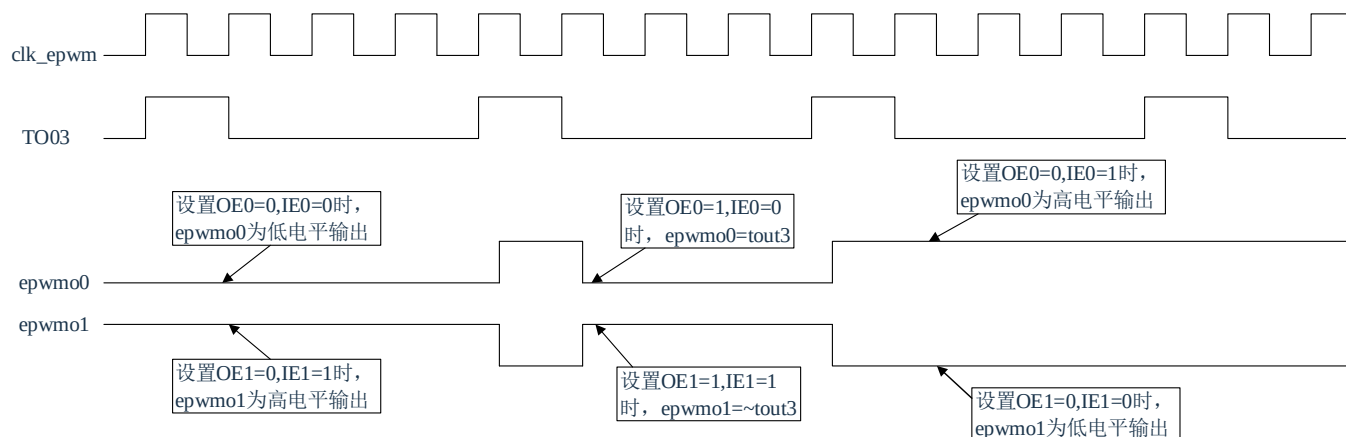


12.4.2 通常运行

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

详细内容请参照“表12-2 截断信号的运行说明表”。

图12-3 输出时序图



12.4.3 强制截断处理

EPWM能通过EPWMSTC的寄存器的bit1-0选择ACMP0/1输出，INTP0输入以及EVENTC事件，使EPWM输出进入强制截断状态。

(1) 强制截断的发生

通过ACMP0/1输出，INTP0输入以及EVENTC事件进入截断状态。通过EPWMSTC的寄存器的bit2(IN_EG)，能选择上升沿或者下降沿，并且在1~2个时钟后进入截断状态。详细内容请参照图12-4。

(2) 强制截断的解除

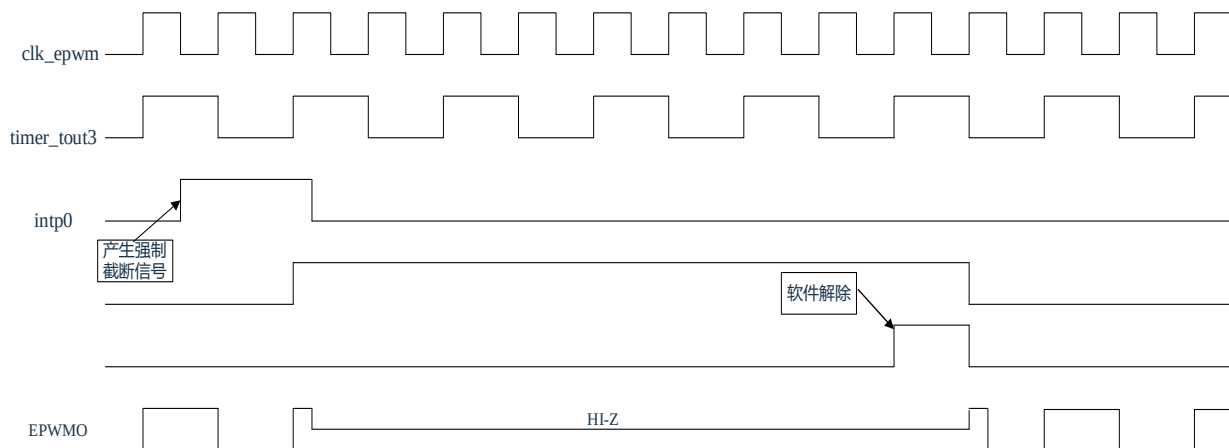
- 软件解除：**EPWMSTC的寄存器bit3（HS_SEL）为0时，使用软件解除模式。EPWMSTR寄存器的bit0（HZCLR）是截断状态的清除位。当截断状态标志SHTFLG为高电平时，如果将HZCLR位置“1”，截断状态标志SHTFLG就变为低电平，并且解除强制截断状态。
- 硬件解除：**EPWMSTC的寄存器bit3（HS_SEL）为1时，使用硬件解除模式。通过ACMP0/1输出或者INTP0输入的边沿解除强制截断状态。

表12-2 截断信号的运行说明表(n=0~7)

位	IO _n 1-0	OE0 _n	IE0 _n	SHTFLG	EPWM输出引脚
设定值	00	1	0	*	正转波形
	00	1	1	*	反转波形
	01	*	*	*	低电平输出
	10	*	*	*	高电平输出
	11	*	*	1	HI-Z输出

截断信号的产生和解除时序图如下所示(HS_SEL=0,REL_SEL=0):

图12-4 INTP0截断的产生和解除时序图



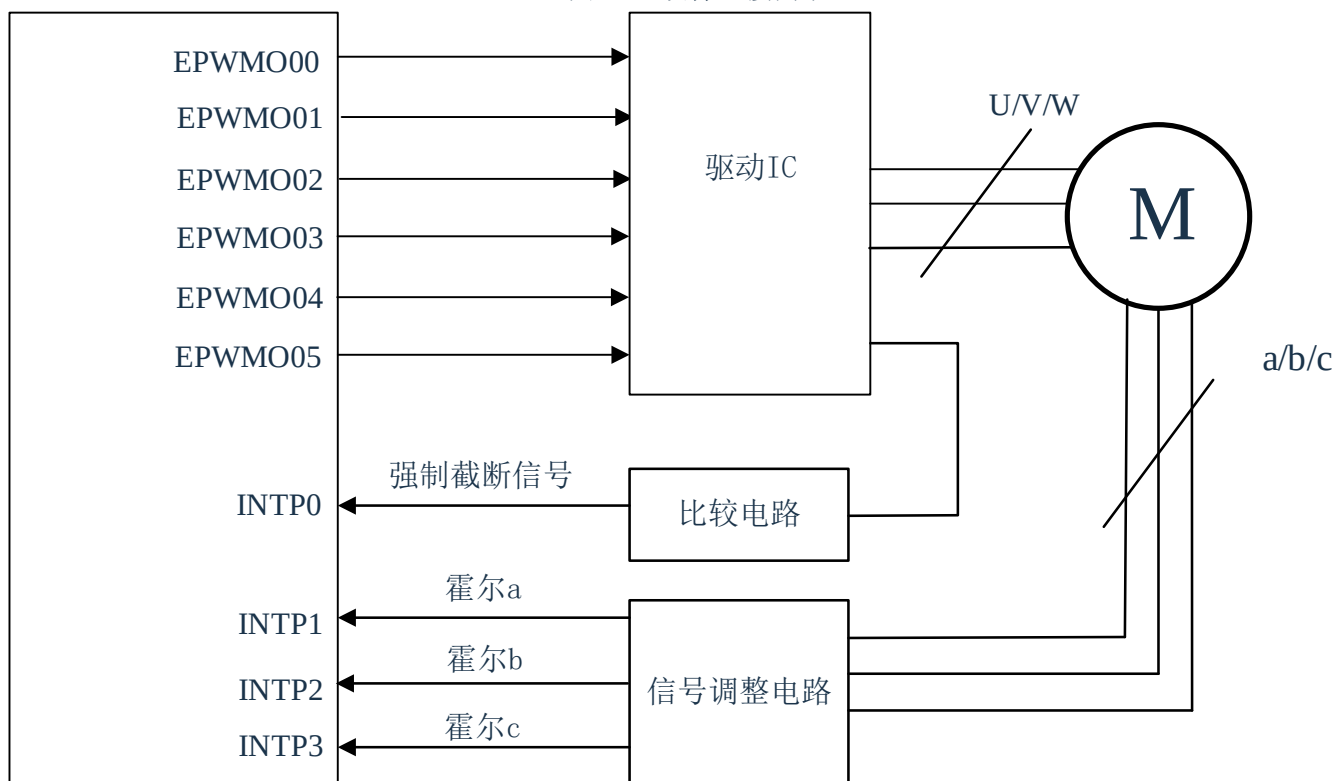
12.5 无刷直流电机的控制例子

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

12.5.1 硬件连接例子

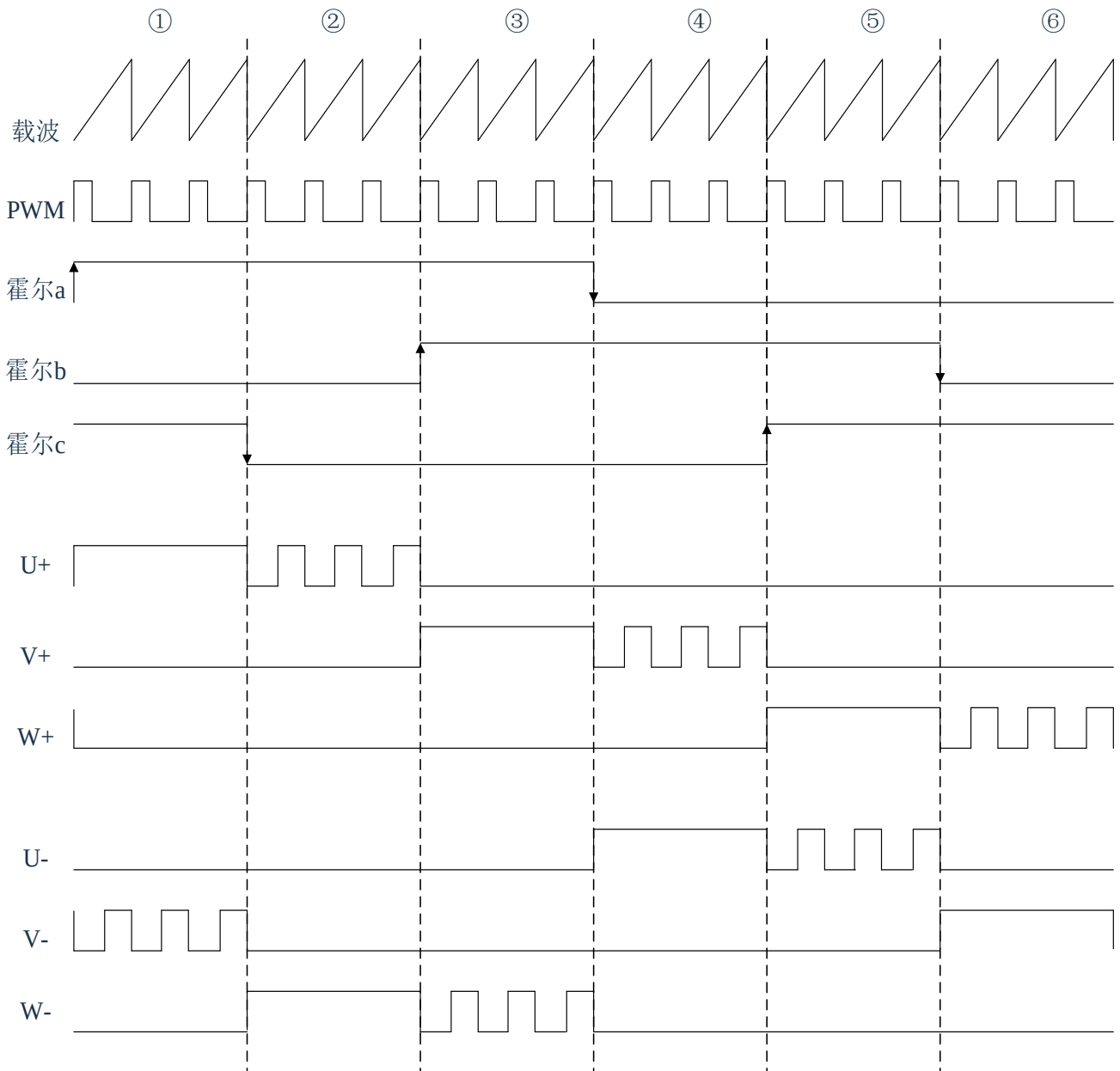
无刷直流电机的硬件连接例子如图12-5所示。在此例中，EPWMO00~EPWMO05（输出）用于BLDC电机的输出控制，INTP1~INTP3（输入）用于霍尔传感器的输出信号，INTP0（输入）用于强制截断信号。

图 12-5 硬件连接例子



12.5.2 三相无刷直流电机的控制时序

图12-6 三相无刷直流电机的控制时序



12.5.3 寄存器的设定例子

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

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

表12-3 EPWMCTL0寄存器的设定例子

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

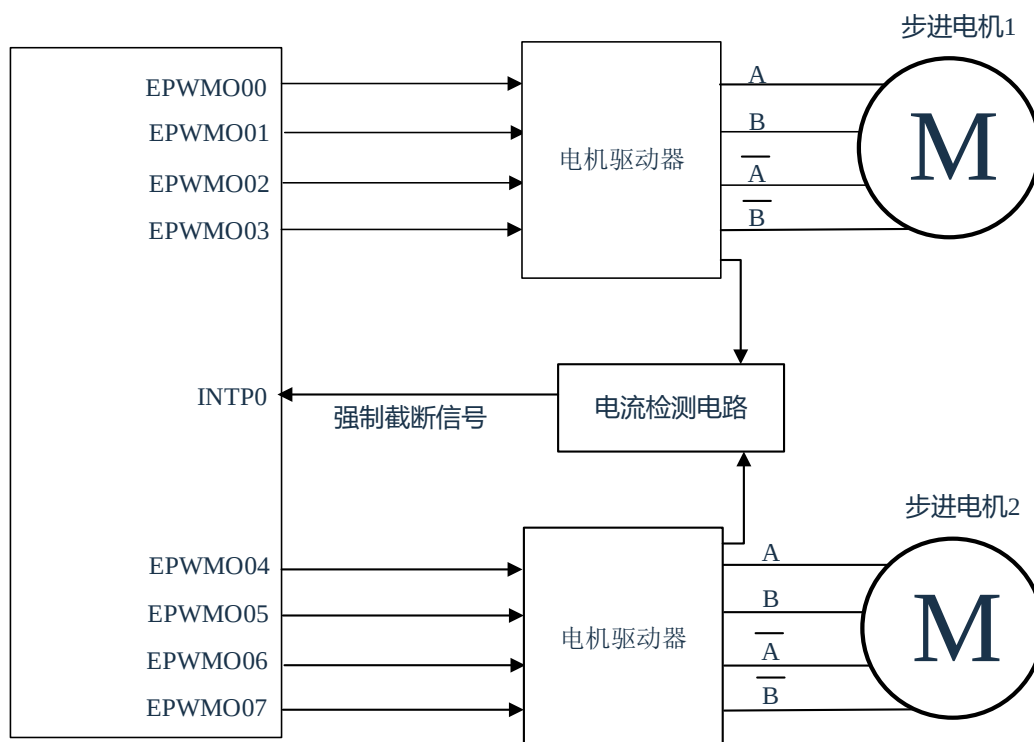
12.6 步进电机的控制例子

以下对使用8个实时输出控制2台2相步进电机的例子进行说明。

12.6.1 硬件连接例子

控制2台步进电机的硬件连接例子如图12-7所示。

图12-7 硬件连接例子



12.6.3 寄存器的设定例子

表12-4 控制步进电机的寄存器的设定例子

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

第13章 通用串行通信单元

本产品搭载 5 个通用串行通信单元，单元 0 有 4 个串行通道，其余每个单元有 2 个串行通道，单元 0/1 各通道能实现 3 线串行(SSPI)、UART 和简易 I²C 的通信功能，单元 2/3/4 各通道能实现 UART 的通信功能。其通道功能分配如下：

表 13-1：通用串行通道单元通道功能分配

单元	通道	用作 SSPI	用作 UART	用作简易 I ² C
0	0	SSPI00	UART0（支持 LIN-bus）	IIC00
	1	SSPI01		IIC01
	2	SSPI10	UART1	IIC10
	3	SSPI11		IIC11
1	0	SSPI20	UART2	IIC20
	1	SSPI21		IIC21
2	0	-	UART3	-
	1			
3	0	-	UART4	-
	1			
4	0	-	UART5	-
	1			

注意：

“-”表示在此型号产品中不支持；

在单元 0 的通道 0 和通道 1 使用 UART0 时，不能使用 SSPI00 和 SSPI01 和 IIC00 和 IIC01；

在单元 0 的通道 2 和通道 3 使用 UART1 时，不能使用 SSPI10 和 SSPI11 和 IIC10 和 IIC11；

在单元 1 的通道 0 和通道 1 使用 UART2 时，不能使用 SSPI20 和 SSPI21 和 IIC20 和 IIC21。

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 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。

有关具体的设定例子，请参照“13.7 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、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定

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期
- 最大传送速率
- 主控通信：Max.6MHz
- 从属通信：Max.6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注意：必须在满足 SCLK 周期时间（ T_{KCY} ）特性的范围内使用。详细内容请参照数据手册。

13.1.2 带从属片选功能的简易SPI

支持从属片选输入功能的SPI串行通信接口。这是使用一个从属片选输入（SS）、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4条通信线进行通信的时钟同步通信接口。

[数据的发送和接收]

- 7位、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定

[时钟控制]

- 输入/输出时钟的相位控制
- 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

从属通信：6MHz

[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

13.1.3 UART (UART0~UART5)

这是通过串行数据发送 (TxD) 和串行数据接收 (RxD) 共 2 条线进行异步通信的功能。使用这 2 条通信线, 按数据帧 (由起始位、数据、奇偶校验位和停止位构成) 与其他通信方进行异步 (使用内部波特率) 的数据发送和接收。能通过使用发送专用 (偶数通道) 和接收专用 (奇数通道) 共 2 个通道来实现全双工 UART 通信。

有关具体的设定例子, 请参照“13.9 UART (UART0~UART5) 通信的运行”。

[数据的发送和接收]

- 7 位、8 位的数据长度
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定、反相的选择
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误或者溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

13.1.4 简易I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）

这是通过串行时钟（SCL）和串行数据（SDA）共 2 条线与多个设备进行时钟同步通信的功能。因为此简易 I²C 是为了与 EEPROM、闪存、A/D 转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。有关具体的设定例子，请参照“13.11 简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收（仅限单主控的主控功能）
- ACK 输出功能、ACK 检测功能
- 8 位数据长（在发送地址时，用高 7 位指地址，用最低位进行 R/W 控制）
- 通过软件产生开始条件和停止条件。

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误、溢出错误

[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注：在接收最后的数据时，如果给 SOEmn 位（串行输出允许寄存器 m（SOEm））写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“13.11.3(2) 处理流程”。

备注：在使用全功能的 I²C 总线时，请参照“第 14 章 串行接口 IICA”。

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--2：通用串行通信单元的结构

项目	结构
移位寄存器	16位
缓冲寄存器	串行数据寄存器mn (SDRmn) 注
串行时钟输入/输出	SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11、SCLKOI20、SCKOI21引脚（用于3线串行I/O）、SCL00、SCL01、SCL10、SCL11、SCL20、SCL21引脚（用于简易I2C）
串行数据输入	SDI00、SDI01、SDI10、SDI11、SDI20、SDI21引脚（用于3线串行I/O）、RxD0（用于支持LIN-bus的UART）、RxD1、RxD2、RxD3、RxD4、RxD5引脚（用于UART）
串行数据输出	SDO00、SDO01、SDO10、SDO11、SDO20、SDO21引脚（用于3线串行I/O）、TxD0（用于支持LIN-bus的UART）、TxD1、TxD2、TxD3、TxD4、TxD5引脚（用于UART）
串行数据输入/输出	SDA00、SDA01、SDA10、SDA11、SDA20、SDA21引脚（用于简易I2C）
从属选择输入	SAU0_SS引脚（用于串行通信单元0做SPI从属选择输入功能） SAU1_SS引脚（用于串行通信单元1做SPI从属选择输入功能）
控制寄存器	<单元设定部的寄存器> 外围允许寄存器（PER0、PER14） 串行时钟选择寄存器m（SPSm） 串行通道允许状态寄存m（SEm） 串行通道开始寄存器m（SSm） 串行通道停止寄存器m（STm） 串行输出允许寄存器m（SOEm） 串行输出寄存器m（SOM） 串行输出电平寄存器m（SOLm） 输入切换控制寄存器（ISC） 噪声滤波器允许寄存器0（NFEN0）
	<各通道部的寄存器> 串行数据寄存器mn（SDRmn） 串行模式寄存器mn（SMRmn） 串行通信运行设定寄存器mn（SCRmn） 串行状态寄存器mn（SSRmn） 串行标志清除触发寄存器mn（SIRmn）
	端口复用功能配置寄存器（PS_xx） 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：SEmn=1 的动作期间；

备注：m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）；

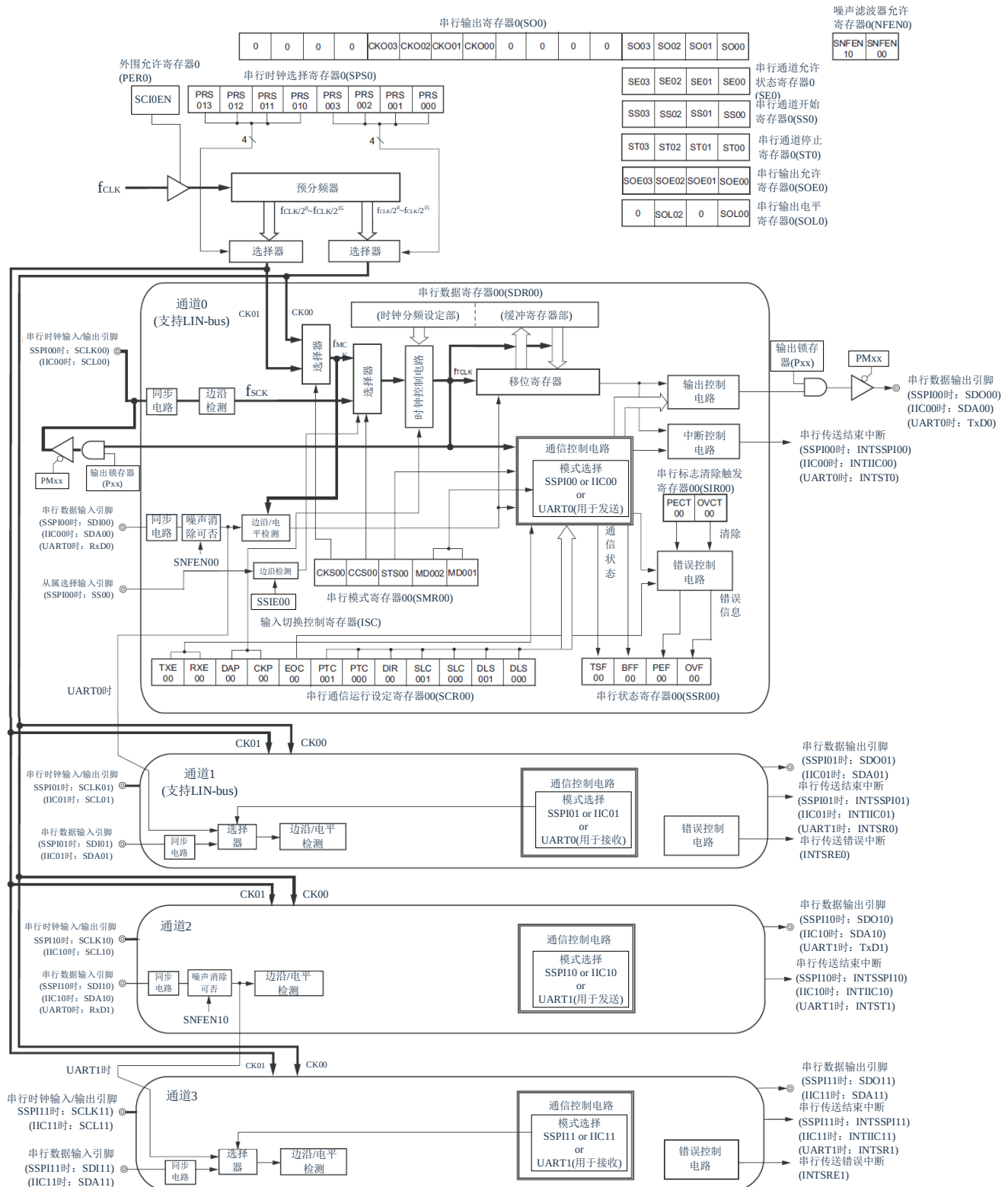
p：SSPI 号（p=00、01、10、11、20、21）；

q：UART 号（q=0、1、2、3、4、5）；

r：IIC 号（r=00、01、10、11、20、21）。

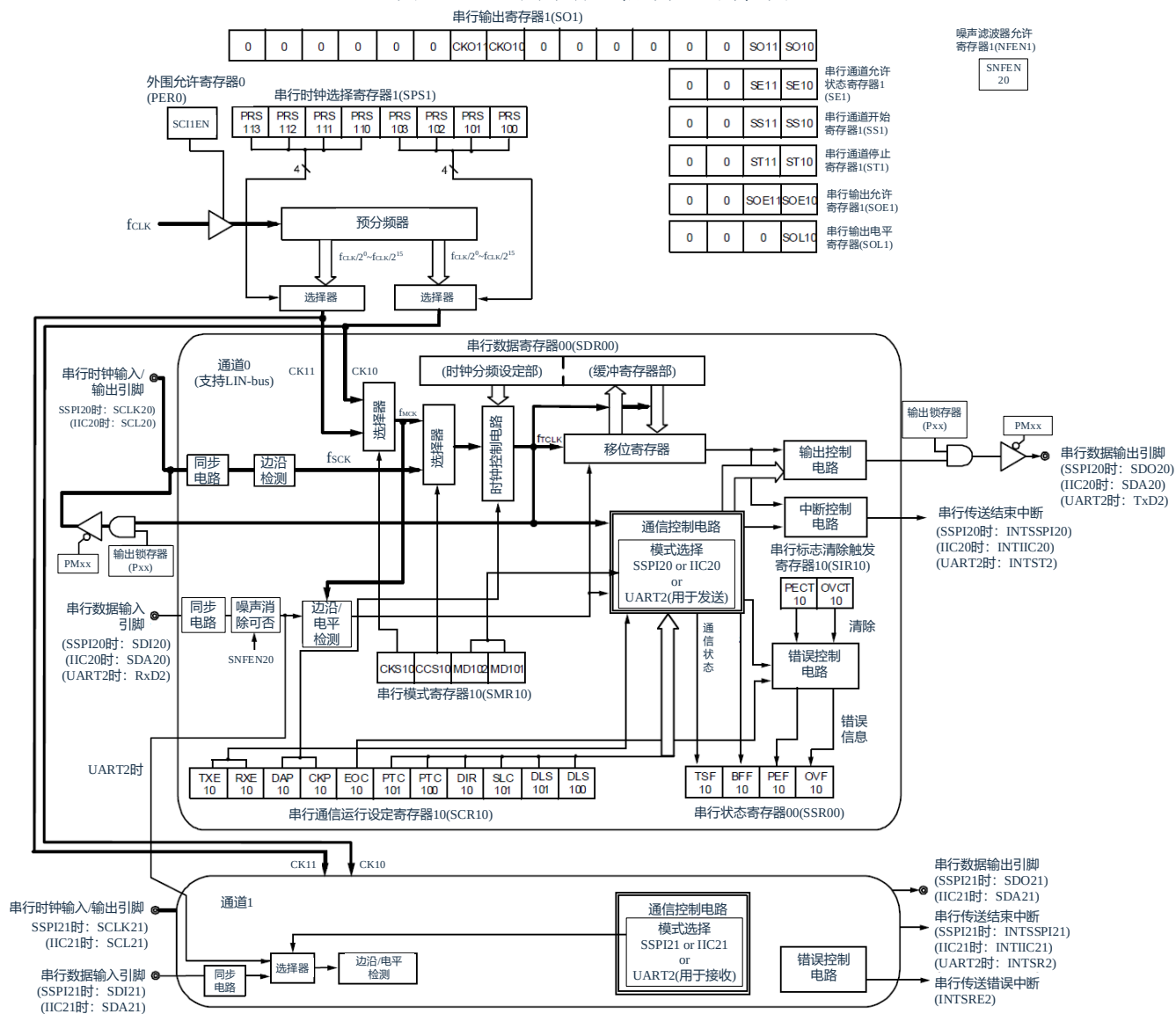
通用串行通信单元的框图如图 13-1 所示：（以单元 0 为例）

图 13-1：通用串行通信单元 0 的框图



通用串行通信单元的框图如图 13-2 所示: (以单元 1 为例)

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 16 位寄存器。

在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出。不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，在运行期间(SEmn=1)使用串行数据寄存器 mn(SDRmn)。

表 13-3：移位寄存器的格式

位	符号	描述	复位值
15:0	SDRmn	串行数据移位寄存器	0x0

13.2.2 串行数据寄存器mn (SDRmn)

SDRmn 寄存器是通道 n 的发送和接收数据寄存器（16 位）。

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

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

与数据的输出顺序无关，根据串行通信运行设定寄存器 mn（SCRmn）的 bit1~bit0（DLSmn1~DLSmn0）的设定，保存到 SDRmn 寄存器的数据如下所示：

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

能以 16 位为单位读写 SDRmn 寄存器。

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

根据通信模式，能用以下 SFR 名称，读写 SDRmnL 寄存器。

- SSPIp 通信.....SDIOpL
- UARTq 接收.....RXDq（UARTq 接收数据寄存器）
- UARTq 发送.....TXDq（UARTq 发送数据寄存器）
- IICr 通信.....SDIOr（IICr 数据寄存器）

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

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

备注：m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）；

p：SSPI 号（p=00、01、10、11、20、21）；

q：UART 号（q=0、1、2、3、4、5）；

r：IIC 号（r=00、01、10、11、20、21）。

表 13-4：串行数据寄存器 mn（SDRmn）的格式

位	符号	描述	复位值
15:0	SDRmn	串行数据寄存器	0x0

备注：

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”；

m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）。

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)
- 外围允许寄存器 14 (PER14)
- 串行时钟选择寄存器 m (SPSm)
- 串行模式寄存器 mn (SMRmn)
- 串行通信运行设定寄存器 mn (SCRmn)
- 串行数据寄存器 mn (SDRmn)
- 串行标志清除触发寄存器 mn (SDIRmn)
- 串行状态寄存器 mn (SSRmn)
- 串行通道开始寄存器 m (SSm)
- 串行通道停止寄存器 m (STm)
- 串行通道允许状态寄存器 m (SEm)
- 串行输出允许寄存器 m (SOEm)
- 串行输出电平寄存器 m (SOLm)
- 串行输出寄存器 m (SOM)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器 0 (NFEN0)
- 端口复用功能配置寄存器 (PS_xx)
- 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、2、3、4)；

n：通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)。

串行通信单元寄存器列表:

单元 0 寄存器基地址: 0x40041100

单元 1 寄存器基地址: 0x40041500

单元 2 寄存器基地址: 0x40041600

单元 3 寄存器基地址: 0x40041700

单元 4 寄存器基地址: 0x40041800

表 13-5: 寄存器列表

偏移地址	寄存器名称	R/W	复位值
0x000	SSRm0	R	0000H
0x002	SSRm1	R	0000H
0x004	SSR02	R	0000H
0x006	SSR03	R	0000H
0x008	SIRm0	R/W	0000H
0x00A	SIRm1	R/W	0000H
0x00C	SIR02	R/W	0000H
0x00E	SIR03	R/W	0000H
0x010	SMRm0	R/W	0020H
0x012	SMRm1	R/W	0020H
0x014	SMR02	R/W	0020H
0x016	SMR03	R/W	0020H
0x018	SCRm0	R/W	0087H
0x01A	SCRm1	R/W	0087H
0x01C	SCR02	R/W	0087H
0x01E	SCR03	R/W	0087H
0x020	SEm	R/O	0000H
0x022	SSm	R/W	0000H
0x024	STm	R/W	0000H
0x026	SPSm	R/W	0000H
0x028	SOM	R/W	0303H
0x02A	SOEm	R/W	0000H
0x034	SOLm	R/W	0000H
0x040	SDRm0	R/W	0000H
0x042	SDRm1	R/W	0000H
0x044	SDR02	R/W	0000H
0x046	SDR03	R/W	0000H

备注: 单元号 m=0, 1、2、3、4。

(以下寄存器基地址 = 0x4004_0470) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
NFEN0	0x000	R/W	噪声滤波器允许寄存器0	0x00
ISC	0x003	R/W	输入切换控制寄存器	0x00

13.4 外围允许寄存器（PER0、PER14）

PER0 和 PER14 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用通用串行通信单元 0 时，必须将 PER0 的 bit2（SAU0EN）置“1”。

要使用通用串行通信单元 1 时，必须将 PER0 的 bit3（SAU1EN）置“1”。

要使用通用串行通信单元 2 时，必须将 PER14 的 bit0（SAU2EN）置“1”。

要使用通用串行通信单元 3 时，必须将 PER14 的 bit1（SAU3EN）置“1”。

要使用通用串行通信单元 4 时，必须将 PER14 的 bit3（SAU4EN）置“1”。

表 13-6: 围允许寄存器 0（PER0）的格式

位	符号	描述	复位值
7	RTCEN	提供实时时钟（RTC）和15位间隔定时器输入时钟的控制 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。 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。	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

外围允许寄存器14（PER14）

位	符号	描述	复位值
7:4	-	保留	0
3	OSDCEN	提供时钟监控OSDC模块输入时钟的控制 0: 停止提供输入时钟，OSDC不能运行 1: 提供输入时钟，OSDC能运行	0
2	SAU4EN	提供通用串行通信单元模块SAU4输入时钟的控制 0: 停止提供输入时钟，SAU4不能运行 1: 提供输入时钟，SAU4能运行	0
1	SAU3EN	提供通用串行通信单元模块SAU3输入时钟的控制 0: 停止提供输入时钟，SAU3不能运行 1: 提供输入时钟，SAU3能运行	0
0	SAU2EN	提供通用串行通信单元模块SAU2输入时钟的控制 0: 停止提供输入时钟，SAU2不能运行 1: 提供输入时钟，SAU2能运行	0

注意：要设定通用串行通信单元 m 时，必须先在 SAUmEN 位为“1”的状态下设定以下的寄存器。当 SAUmEN 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（PS_xx）、端口输出模式寄存器（POMx）、端口模式寄存器（PMx）、端口模式控制寄存器（PMCx）以及端口寄存器（Px）除外）：

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

13.4.1 串行时钟选择寄存器m (SPSm)

SPSm 寄存器是 16 位寄存器，选择提供给各通道的 2 种公共运行时钟（CKm0、CKm1）。通过 SPSm 寄存器的 bit7~4 选择 CKm1，通过 bit3~0 选择 CKm0。

禁止在运行过程中（SEmn=1）改写 SPSm 寄存器。

通过 16 位存储器操作指令设定 SPSm 寄存器。

能用 SPSmL 并且通过 8 位存储器操作指令设定 SPSm 寄存器的低 8 位。

在产生复位信号后，SPSm 寄存器的值变为“0000H”。

表 13-7：串行时钟选择寄存器 m (SPSm) 的格式

位	符号	描述	复位值
15:8	-		
7:4	PRSm13~PRSm10	运行时钟CKm1的选择 0000: FCLK 0001: FCLK/2 0010: FCLK/2 ² 0011: FCLK/2 ³ 0100: FCLK/2 ⁴ 0101: FCLK/2 ⁵ 0110: FCLK/2 ⁶ 0111: FCLK/2 ⁷ 1000: FCLK/2 ⁸ 1001: FCLK/2 ⁹ 1010: FCLK/2 ¹⁰ 1011: FCLK/2 ¹¹ 1100: FCLK/2 ¹² 1101: FCLK/2 ¹³ 1110: FCLK/2 ¹⁴ 1111: FCLK/2 ¹⁵	0x0
3:0	PRSm03~PRSm00	运行时钟CKm0的选择 0000: FCLK 0001: FCLK/2 0010: FCLK/2 ² 0011: FCLK/2 ³ 0100: FCLK/2 ⁴ 0101: FCLK/2 ⁵ 0110: FCLK/2 ⁶ 0111: FCLK/2 ⁷ 1000: FCLK/2 ⁸ 1001: FCLK/2 ⁹ 1010: FCLK/2 ¹⁰ 1011: FCLK/2 ¹¹ 1100: FCLK/2 ¹² 1101: FCLK/2 ¹³ 1110: FCLK/2 ¹⁴ 1111: FCLK/2 ¹⁵	0x0

注：要在通用串行通信单元（SCI）运行过程中更改被选择为 FCLK 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. FCLK: CPU/外围硬件的时钟频率；
2. m: 单元号（m=0、1、2、3、4）；
3. k=0、1。

13.4.2 串行模式寄存器mn (SMRmn)

SMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟 (F_{MCK}) 的选择、能否使用串行时钟 (F_{SCLK}) 输入的指定、开始触发的设定、运行模式 (SSPI、UART、简易 I²C) 的设定以及中断源的选择。另外，只在 UART 模式中设定接收数据的反相电平。

禁止在运行过程中 ($SEmn=1$) 改写 SMRmn 寄存器，但是能在运行过程中改写 MDmn0 位。

通过 16 位存储器操作指令设定 SMRmn 寄存器。

在产生复位信号后，SMRmn 寄存器的值变为“0020H”。

表 13-8: 串行模式寄存器 mn (SMRmn) 的格式

位	符号	描述	复位值
15	CKSmn	通道n运行时钟 (F_{MCK}) 的选择 0: SPSm寄存器设定的运行时钟CKm0 1: SPSm寄存器设定的运行时钟CKm1 运行时钟 (F_{MCK}) 用于边沿检测电路。通过设定CCSmn位和SDRmn寄存器的高7位，生成传送时钟 (F_{TCLK})。	0
14	CCSmn	通道n传送时钟 (F_{TCLK}) 的选择 0: CKSmn位指定的运行时钟 F_{MCK} 的分频时钟 1: 来自SCLKp引脚的输入时钟 F_{SCLK} (SSPI模式的从属传送) 传送时钟 F_{TCLK} 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当CCSmn位为“0”时，通过SDRmn寄存器的高7位进行运行时钟 (F_{MCK}) 的分频设定。	0
13:9	-	-	-
8	STSmn ^{注1}	开始触发源的选择 0: 只有软件触发有效 (在SSPI、UART发送、简易I ² C时选择)。 1: RxDq引脚的有效边沿 (在UART接收时选择) 在将SSm寄存器置“1”后满足上述条件时，开始传送。	0
7	-	-	-
6	SISmn0 ^{注1}	UART模式中的通道n接收数据的电平反相控制 0: 将下降沿检测为起始位。 不将输入的通信数据进行反相。 1: 将上升沿检测为起始位。 将输入的通信数据进行反相。	0x0
5:3	-	-	0x4
2:1	MDmn2~MDmn1	通道n运行模式的设定 00: SSPI模式 01: UART模式 10: 简易I ² C模式 11: 禁止设定。	0x0
0	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。	0

注 1: 只限于 SMR01、SMR03、SMR11、SMR21、SMR31、SMR41 寄存器；

注意: 必须将 bit13~9、7、4、3 (SMR00、SMR02、SMR10、SMR20、SMR30、SMR40 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、2、3、4)；

n: 通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)；p: SSPI 号 (p=00、01、10、11、20、21)；

q: UART 号 (q=0、1、2、3、4、5)；

r: IIC 号 (r=00、01、10、11、20、21)。

13.4.3 串行通信运行设定寄存器mn (SCRmn)

SCRmn 寄存器是通道 n 的通信运行设定寄存器，设定数据发送和接收模式、数据和时钟相位、是否屏蔽错误信号、奇偶检验位、起始位、停止位和数据长度等。

禁止在运行过程中 (SEmn=1) 改写 SCRmn 寄存器。

通过 16 位存储器操作指令设定 SCRmn 寄存器。

在产生复位信号后，SCRmn 寄存器的值变为“0087H”。

表 13-9: 串行通信运行设定寄存器 mn (SCRmn) 的格式

位	符号	描述	复位值
15:14	TXEmn,RXEmn	通道n运行模式的设定 00: 禁止通信。 01: 只进行接收。 10: 只进行发送。 11: 使能发送和接收。	0x0
13:12	DAPmn,CKPmn	SSPI模式中的数据和时钟的相位选择 00: 类型1 01: 类型2 10: 类型3 11: 类型4 在UART模式中，必须将DAPmn和CKPmn位都置0	0x0
11			0x0
10	EOCmn	错误中断信号 (INTSREx (x=0~1)) 的屏蔽控制 0: 禁止产生错误中断INTSREx (产生INTSRx)。 1: 允许产生错误中断INTSREx (在发生错误时不产生INTSRx)。 在SSPI模式和简易I ² C模式中或者在UART发送时，必须将EOCmn位置“0” ^{注2} 。	0x0
9:8	PTCmn1~PTCmn0	UART模式中的奇偶校验位的设定 发送 00: 不输出奇偶校验位。 01: 输出奇偶校验 ^{注3} 。 10: 输出偶校验。 11: 输出奇校验。 接收 接收时没有奇偶校验。 不判断奇偶校验。 判断偶校验。 判断奇校验。 在SSPI模式和简易I ² C模式中，必须将PTCmn1位和PTCmn0位都置“0”。	0x0
7	DIRmn	SSPI和UART模式中的数据传送顺序的选择 0: 进行MSB优先的输入/输出。 1: 进行LSB优先的输入/输出。 在简易I ² C模式中，必须将DIRmn位置“0”。	0x1
6	-	-	0
5:4	SLCmn1~SLCmn0	UART模式中的停止位的设定 00: 无停止位 01: 停止位长度=1位 10: 停止位长度=2位 (只限于mn=00、10、20) 11: 禁止设定。 如果选择了传送结束中断，就在传送完所有停止位后产生中断。 在UART接收时或者在简易I ² C模式中，必须设定为1个停止位 (SLCmn1、SLCmn0=0、1)。 在SSPI模式中，必须设定为无停止位 (SLCmn1、SLCmn0=0、0)。 在UART发送时，必须设定为1位 (SLCmn1、SLCmn0=0、1) 或者2位 (SLCmn1、SLCmn0=1、0)。	0x0 0x0
3:2	-	保留	-
1:0	DLSmn1~DLSmn0	SSPI模式中的数据长度的设定 10: 7位数据长度 (保存在SDRmn寄存器的bit0~6) 串行功能对应 SSPI UART IIC ○ ○ ×	0x7

		11: 8位数据长度（保存在SDRmn寄存器的bit0~7） 其他 禁止设置	○ ○ ○	
		在简易I ² C模式中，必须设置DLSmn1~DLSmn0=11B。		

DAPmn	CKPmn	SSPI 模式中的数据与时钟的相位选择	类型
0	0	SCLKp	1
0	1	SCLKp	2
1	0	SCLKp	3
1	1	SCLKp	4
在 UART 模式和简易 I ² C 模式中，必须将 DAPmn 位和 CKPmn 位都置“0”。			

注 1：只限于 SCR00、SCR02、SCR10、SCR20、SCR30、SCR40 寄存器；

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

注意：必须将 bit6、10、11 置“0”（也必须将 SCR01、SCR03、SCR11、SCR21、SCR31、SCR41 寄存器的 bit5 置“0”）；

备注：m：单元号（m=0、1、2、3、4）；

 n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）；

 p：SSPI 号（p=00、01、10、11、20、21）。

13.4.4 串行数据寄存器mn（SDRmn）

SDRmn 寄存器是通道 n 发送和接收的数据寄存器（16 位）。

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

如果将串行模式寄存器 mn（SMRmn）的 CCSmn 位置“0”，由 SDRmn 寄存器的 bit15~9（高 7 位）设定的运行时钟的分频时钟就用作传送时钟。

通过 16 位存储器操作指令设定 SDRmn 寄存器。

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

表 13-10：串行数据寄存器 mn（SDRmn）的格式

位	符号	描述	复位值
15:9		运行时钟分频的传送时钟设定 7'h0: FMCK 7'h1: FMCK/2 7'h2: FMCK/3 7'h3: FMCK/4 • • 7'h7E: FMCK/127 7'h7F: FMCK/128 当运行停止（SEmn=0）时，bit15~9用作运行时钟（FMCK）的分频设定寄存器。在运行期间（SEmn=1）bit15~9用作发送接收缓冲寄存器	0x0
8:0		在运行期间（SEmn=1）用作发送接收缓冲寄存器	0x0

注意：

1. 运行停止（SEmn=0）时，必须将 bit8~0 清零；
2. 在使用 UART 时，禁止将 SDRmn[15:9]设定为“0000000B”和“0000001B”；
3. 在使用简易 I²C 时，禁止将 SDRmn[15:9]设定为“0000000B”，SDRmn[15:9]的设定值必须大于等于“0000001B”；
4. 在运行停止（SEmn=0）时，禁止通过 8 位存储器操作指令改写 SDRmn[7:0]（否则，SDRmn[15:9]全部被清“0”）；

备注：

SDRmn 寄存器在运行期间的功能，请参照“13.2 通用串行通信单元的结构”；

m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）。

13.4.5 串行标志清除触发寄存器mn（SIRmn）

这是用于清除通道 n 各错误标志的触发寄存器。

如果将各位（FECTmn、PECTmn、OVCTmn）置“1”，就将串行状态寄存器 mn（SSRmn）的对应位（FEFmn、PEFmn、OVFmn）清“0”。因为 SIRmn 寄存器是触发寄存器，所以如果清除 SSRmn 寄存器的对应位，也会立即清除 SIRmn 寄存器。

通过 16 位存储器操作指令设定 SIRmn 寄存器。

能用 SIRmnL 并且通过 8 位存储器操作指令设定 SIRmn 寄存器的低 8 位。

在产生复位信号后，SIRmn 寄存器的值变为“0000H”。

表 13-11：串行标志清除触发寄存器 mn（SIRmn）的格式

位	符号	描述	复位值
15:3			0x0
2	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”。	0x0
1	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”。	0x0
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”。	0x0

注 1：只限于 SIR01、SIR03、SIR11、SIR21、SIR31、SIR41 寄存器；

注意：必须将 bit15~3（SIR00、SIR10 寄存器为 bit15~2）置“0”；

备注：

m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）；

SIRmn 寄存器的读取值总是“0000H”。

13.4.6 串行状态寄存器mn（SSRmn）

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

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

在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

表 13-12：串行状态寄存器 mn（SSRmn）的格式

位	符号	描述	复位值
15:7			0x0
6	TSFmn	通道n通信状态的表示标志 0: 通信停止状态或者通信待机状态 1: 通信运行状态 [清除条件] • 当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信待机状态）时 • 当通信结束时 [置位条件] • 当开始通信时	0
5	BFFmn	通道n缓冲寄存器的状态表示标志 0: SDRmn寄存器没有保存有效数据。 1: SDRmn寄存器保存了有效数据。 [清除条件] • 在发送过程中传送完从SDRmn寄存器到移位寄存器的发送数据时 • 在接收过程中从SDRmn寄存器读完接收数据时 • 当将STmn寄存器的STmn位置“1”（通信停止状态）或者将SSmn寄存器的SSmn位置“1”（通信允许状态）时 [置位条件] • 在SCRmn寄存器的TXEmn位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给SDRmn寄存器写发送数据时 • 在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到SDRmn寄存器时 • 当发生接收错误时	0
4:3			0x0
2	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”层面上名称时 [置位条件] • 在UART接收结束时没有检测到停止位时	0
1	PEFmn	通道n奇偶校验错误的检测标志 0: 没有发生错误。 1: 发生错误（UART接收时）或者未检测到ACK（I ² C发送时）。 [清除条件] • 当给SIRmn寄存器的PECTmn位写“1”时 [置位条件] • 在UART接收结束时发送数据的奇偶校验和奇偶校验位不同（奇偶校验错误）时 • 在I ² C发送时并且在ACK接收时序从属方没有返回ACK信号（未检测到ACK）时	0
0	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]	0

		•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、SSR21、SSR31、SSR41 寄存器。

备注：如果在 BFFmn 位为“1”时写 SDRmn 寄存器，就会丢弃被保存的发送或者接收数据，并且检测到溢出错误（OVEmn=1）。

m：单元号（m=0、1、2、3、4）

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）

13.4.7 串行通道开始寄存器m (SSm)

SSm 寄存器是设定允许各通道的通信/开始计数的触发寄存器。

如果给各位 (SSmn) 写“1”，就将串行通道允许状态寄存器 m (SEm) 的对应位 (SEmn) 置“1” (运行允许状态)。因为 SSmn 位是触发位，所以如果 SEmn 位为“1”就立即清除 SSmn 位。

通过 16 位存储器操作指令设定 SSm 寄存器。

能用 SSmL 并且通过 8 位存储器操作指令设定 SSm 寄存器的低 8 位。

在产生复位信号后，SSm 寄存器的值变为“0000H”。

表 13-13: 串行通道开始寄存器 m (SSm) 的格式

位	符号	描述	复位值
15:4			0x0
3	SS03	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 。 仅通道0有效	0x0
2	SS02	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 。 仅通道0有效	0x0
1	SSm1	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 。	0x0
0	SSm0	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 。	0x0

注：如果在通信过程中将 SSmn 位置“1”，就停止通信进入待机状态。此时，控制寄存器和移位寄存器的值、SCLKmn 引脚和 SDOmn 引脚、FEFmn 标志、PEFmn 标志和 OVFmn 标志保持状态；

注意：必须将 SS0 寄存器的 bit15~4 置“0”，SSm(m≠0)寄存器的 bit15~2 置“0”；

备注：

m: 单元号 (m=0、1、2、3、4)；

n: 通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)；

SSm 寄存器的读取值总是“0000H”。

13.4.8 串行通道停止寄存器m (STm)

STm 寄存器是设定允许各通道的通信/停止计数的触发寄存器。

如果给各位 (STmn) 写“1”，就将串行通道允许状态寄存器 m (SEm) 的对应位 (SEmn) 清“0” (运行停止状态)。因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

通过 16 位存储器操作指令设定 STm 寄存器。

能用 STmL 并且通过 8 位存储器操作指令设定 STm 寄存器的低 8 位。

在产生复位信号后，STm 寄存器的值变为“0000H”。

表 13-14: 串行通道停止寄存器 m (STm) 的格式

位	符号	描述	复位值
15:4			0x0
3	ST03	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 。 仅通道0有效	0
2	ST02	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 。 仅通道0有效	0
1	STm1	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 。	0
0	STm0	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 。	0

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 置“0”，STm(m≠0)寄存器的 bit15~2 置“0”；

备注：

m: 单元号 (m=0、1、2、3、4)；

n: 通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)；

STm 寄存器的读取值总是“0000H”。

13.4.9 串行通道允许状态寄存器m (SEm)

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

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

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

对于停止运行的通道 n，能通过软件设定 SOm 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令读取 SEm 寄存器。

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

在产生复位信号后，SEm 寄存器的值变为“0000H”。

表 13-15: 串行通道允许状态寄存器 m (SEm) 的格式

位	符号	描述	复位值
15:4			0x0
3	SE03	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 仅通道0有效	0
2	SE02	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 仅通道0有效	0
1	SEm1	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	0
0	SEm0	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	0

备注：m：单元号（m=0、1、2、3、4）；

n：通道号（m=0 时 n=0、1、2、3；m≠0 时 n=0、1）。

13.4.10 串行输出允许寄存器m (SOEm)

SOEm 寄存器设定允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOm) 的 SOmn 位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道 n，能通过软件设定 SOm 寄存器的 SOmn 位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令设定 SOEm 寄存器。

能用 SOEmL 并且通过 8 位存储器操作指令设定 SOEm 寄存器的低 8 位。

在产生复位信号后，SOEm 寄存器的值变为“0000H”。

表 13-16: 串行输出允许寄存器 m (SOEm) 的格式

位	符号	描述	复位值
15:4			0x0
3	SOE03	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。 仅通道0有效	0
2	SOE02	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。 仅通道0有效	0
1	SOEm1	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。	0
0	SOEm0	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。	0

备注：SOE0 寄存器的 bit15~4 置“0”，SOEm (m≠0) 寄存器的 bit15~2 置“0”；

m: 单元号 (m=0、1、2、3、4)；

n: 通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)。

13.4.11 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 CKOm 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 CKOm 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 CKOm 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 CKOm 位和 SOMn 位置“1”。

通过 16 位存储器操作指令设定 SOM 寄存器。

在产生复位信号后, SOM 寄存器的值变为“0303H”。

表 13-17: 串行输出寄存器 m (SOM) 的格式

位	符号	描述	复位值
15:12			0x0
11	CKO03	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”。 仅通道0有效	1
10	CKO02	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”。 仅通道0有效	1
9	CKOm1	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”。	1
8	CKOm0	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”。	1
7:4			0x0
3	SO03	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”。 仅通道0有效	1
2	SO02	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”。 仅通道0有效	1
1	SOM1	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”。	1
0	SOM0	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”。	1

注意: 必须将 SO0 寄存器的 bit15~12 和 bit7~4 置“0”, SOM(m≠0)寄存器的 bit15~10 和 bit7~2 置“0”;

备注: m: 单元号 (m=0、1、2、3、4);

n: 通道号 (m=0 时 n=0、1、2、3; m≠0 时 n=0、1)。

13.4.12 串行输出电平寄存器m (SOLm)

SOLm 寄存器是设定各通道的数据输出电平反相的寄存器。

只有在 UART 模式中才能设定此寄存器。在 SSPI 模式和简易 I²C 模式中，必须将对应位置“0”。只在允许串行输出时 (SOEmn=1)，将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时 (SOEmn=0)，将 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定 SOLm 寄存器。

能用 SOLmL 并且通过 8 位存储器操作指令设定 SDOLm 寄存器的低 8 位。

在产生复位信号后，SOLm 寄存器的值变为“0000H”。

表 13-18: 串行输出电平寄存器 m (SOLm) 的格式

位	符号	描述	复位值
15:3			
2	SOL02	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。 仅通道0有效	0
1			0x0
0	SOLm0	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。	0

注意：必须将 SOL0 寄存器的 bit15~3，bit1 置“0”，SOLm 寄存器的 bit15~1 置“0”；

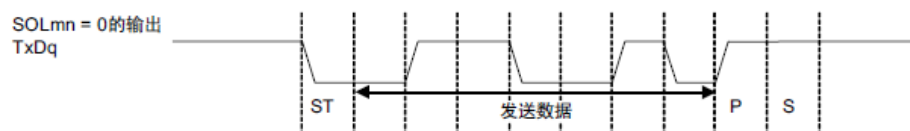
备注：m：单元号 (m=0、1、2、3、4)；

n：通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)。

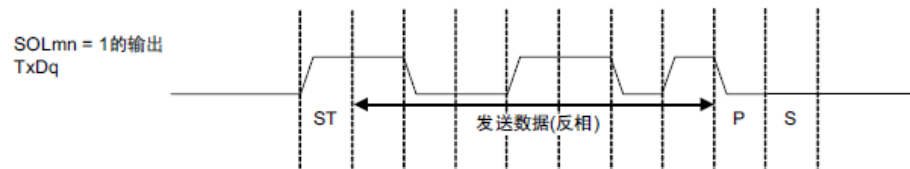
当进行 UART 发送时，发送数据的电平反相例子如图 13-2 所示：

图 13-2 发送数据的电平反相例子

(a) 正相输出 (SOLmn = 0)



(b) 反相输出 (SOLmn = 1)



备注：m：单元号 (m=0、1、2、3、4)；

n：通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)。

13.4.13 输入切换控制寄存器（ISC）

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

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

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

通过 8 位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

表 13-19：输入切换控制寄存器（ISC）的格式

位	符号	描述	复位值
7	-	-	
6	SSIE00	SSPI00通信的从属模式中通道0的SS00输入的设定 0: SS00引脚输入无效。 1: SS00引脚输入有效。	0
5:2	-	-	0x0
1	ISC1	定时器Timer4的通道3的输入切换 0: 将TI03引脚的输入信号用作定时器的输入（通常运行）。 1: 将RxD0引脚的输入信号用作定时器的输入（检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽）。	0
0	ISC0	外部中断（INTP0）的输入切换 0: 将INTP0引脚的输入信号用作外部中断的输入（通常运行）。 1: 将RxD0引脚的输入信号用作外部中断的输入（检测唤醒信号）。	0

注意：必须将 bit7,5~2 置“0”。

13.4.14 噪声滤波器允许寄存器0 (NFEN0)

NFEN0 寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 SSPI 或者简易 I²C 通信的引脚，必须将对应位置“0”，使噪声滤波器无效。对于用于 UART 通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟 (F_{MCK}) 进行同步后检测 2 个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟 (F_{MCK}) 进行同步。

通过 8 位存储器操作指令设定 NFEN0 寄存器。

在产生复位信号后，NFEN0 寄存器的值变为“00H”。

表 13-20: 噪声滤波器允许寄存器 0 (NFEN0) 的格式

位	符号	描述	复位值
7:6	-	-	0x0
5	SNFEN40	RxD5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RxD5引脚时，必须将SNFEN50位置“1”。 当用作RxD5引脚以外的功能时，必须将SNFEN50位置“0”。	0x0
4	SNFEN30	RxD4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RxD4引脚时，必须将SNFEN40位置“1”。 当用作RxD4引脚以外的功能时，必须将SNFEN40位置“0”。	0x0
3	SNFEN20	RxD3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RxD3引脚时，必须将SNFEN30位置“1”。 当用作RxD3引脚以外的功能时，必须将SNFEN30位置“0”。	0x0
2	SNFEN10	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SNFEN10位置“1”。 当用作RxD2引脚以外的功能时，必须将SNFEN10位置“0”。	0x0
1	SNFEN02	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SNFEN02位置“1”。 当用作RxD1引脚以外的功能时，必须将SNFEN02位置“0”。	0x0
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”。	0x0

注意：必须将 bit7~6 置“0”。

13.5 控制串行输入/输出引脚端口功能的寄存器

在使用通用串行通信单元时，必须设定复用端口功能的控制寄存器（端口模式寄存器（PMxx）、端口复用功能配置寄存器（PS_xx）、端口输出模式寄存器（POMxx）和端口模式控制寄存器（PMCxx））。

详细内容请参照“第2章 引脚功能”。

在将串行数据输出引脚或者串行时钟输出引脚的复用端口用作串行数据输出或者串行时钟输出时，必须将各端口对应的端口模式控制寄存器（PMCxx）的位和端口模式寄存器（PMxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

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

在将串行数据输入引脚或者串行时钟输入引脚的复用端口用作串行数据输入或者串行时钟输入时，必须将各端口对应的端口模式寄存器（PMxx）的位置“1”，并且将端口模式控制寄存器（PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

13.6 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。

13.6.1 以单元为单位停止运行的情况

通过外围允许寄存器（PER0、PER14）设定以单元为单位的停止运行。

PER0、PER14 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过给不使用的硬件停止提供时钟，以降低功耗和噪声。

要停止通用串行通信单元 0 时，必须将 PER0 的 bit2（SAU0EN）置“0”；

要停止通用串行通信单元 1 时，必须将 PER0 的 bit3（SAU1EN）置“0”；

要停止通用串行通信单元 2 时，必须将 PER14 的 bit0（SAU2EN）置“0”；

要停止通用串行通信单元 3 时，必须将 PER14 的 bit1（SAU3EN）置“0”；

要停止通用串行通信单元 4 时，必须将 PER14 的 bit2（SAU4EN）置“0”；

表 13-21：以单元为单位停止运行时的外围允许寄存器 0（PER0）的设定

位	符号	描述	复位值
7:4			
3	SAU1EN	提供通用串行通信单元1的输入时钟的控制 停止提供输入时钟。 0: 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。	
2	SAU0EN	提供通用串行通信单元0的输入时钟的控制 停止提供输入时钟。 0: 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。	
1:0			

外围允许寄存器14（PER14）

位	符号	描述	复位值
7:3			
2	SAU4EN	提供通用串行通信单元模块SAU4输入时钟的控制 0: 停止提供输入时钟，SAU4不能运行 1: 提供输入时钟，SAU4能运行	0
1	SAU3EN	提供通用串行通信单元模块SAU3输入时钟的控制 0: 停止提供输入时钟，SAU3不能运行 1: 提供输入时钟，SAU3能运行	0
0	SAU2EN	提供通用串行通信单元模块SAU2输入时钟的控制 0: 停止提供输入时钟，SAU2不能运行 1: 提供输入时钟，SAU2能运行	0

备注：

- 当 SAUmEN 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（PS_xx）
 - 端口输出模式寄存器（POMx）
 - 端口模式寄存器（PMx）
 - 端口寄存器（Px）
- ×：这是通用串行通信单元未使用的位（取决于其他外围功能的设定）；
0/1：根据用户的用途置“0”或者“1”。

13.6.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

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

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3	STm2	STm1	STm0

- ① 将 SE_{mn} 位清“0”并且停止通信运行
- ② 因为 ST_{mn} 位是触发位，所以如果 SE_{mn} 位为“0”就立即清除 ST_{mn} 位。

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

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3	SEm2	SEm1	SEm0

- ① 0: 运行停止状态
- ② SEm 寄存器是只读状态寄存器，通过 STm 寄存器来停止运行。对于已经停止运行的通道，能通过软件设定 SOm 寄存器的 CKO_{mn} 位的值。

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

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0

- ① 0: 通过串行通信运行来停止输出
- ② 对于已经停止串行输出的通道，能通过软件设定 SOm 寄存器的 SO_{mn} 位的值。

(4) 串行输出寄存器 m (SOm) 这是各通道串行输出的缓冲寄存器。

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0

- ① CKOm3/2/1/0: 1: 串行时钟的输出值为“1”; 0: 串行时钟的输出值为“0”
- ② SOm3/2/1/0: 1: 串行数据的输出值为“1”; 0: 串行数据的输出值为“0”
- ③ 当将各通道对应的引脚用作端口功能时，必须将相应的 CKO_{mn} 位和 SO_{mn} 位置“1”。

备注：只限于通用串行通信单元 0；

m: 单元号 (m=0、1、2、3、4)；

n: 通道号 (m=0 时 n=0、1、2、3；m≠0 时 n=0、1)；

0/1: 根据用户的用途置“0”或者“1”。

13.7 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、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定

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率
 - 主控通信：Max.6MHz
 - 从属通信：Max.6MHz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

SCI0 的通道 0~3，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 13.7.1）
- 主控接收（参照 13.7.2）
- 主控的发送和接收（参照 13.7.3）
- 从属发送（参照 13.7.4）
- 从属接收（参照 13.7.5）
- 从属的发送和接收（参照 13.7.6）

注意：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。

13.7.1 主控发送

主控发送是指本产品输出传送时钟并且将数据发送到其他设备的运行。

表 13-22：主控发送

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	SCL1的通道0	SCL1的通道1
使用的引脚	SCLKOI00、 SDO00	SCLKOI01、 SDO01	SCLKOI10、 SDO10	SCLKOI11、 SDO11	SCLKOI20、 SDO20	SCLKOI21、 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、8位					
传送速率 ^注	Max.6M[Hz] Min.F _{CLK} /(2x2 ¹¹ x128)[Hz] F _{CLK} : 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0: 正相 CKPmn=1: 反相					
数据方向	MSB优先或者LSB优先					

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

m: 单元号（m=0、1）；

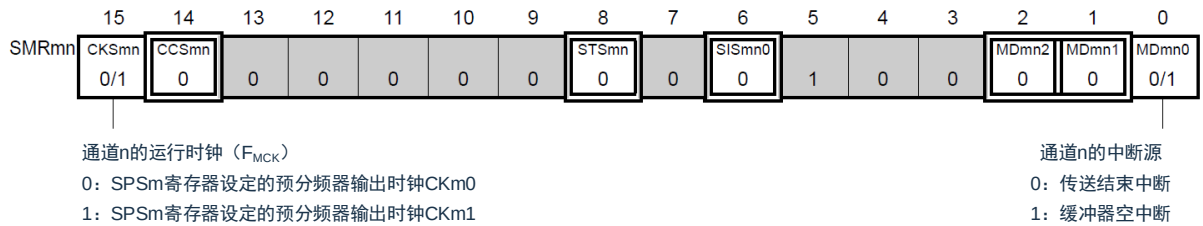
n: 通道号（n=0、1、2、3）；

mn=00~03、10~11。

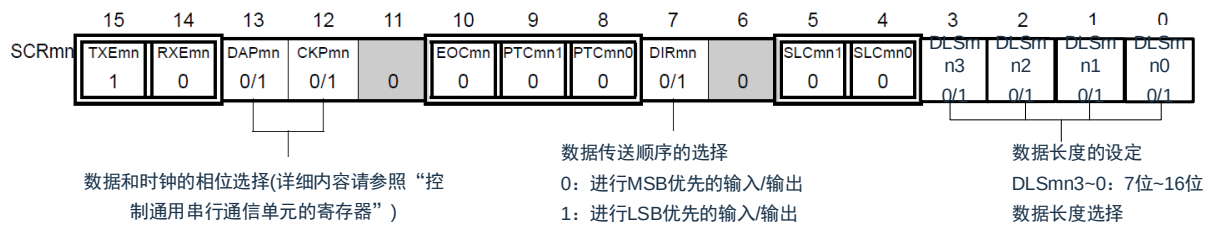
(1) 寄存器的设定

图 13-4：3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）
主控发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

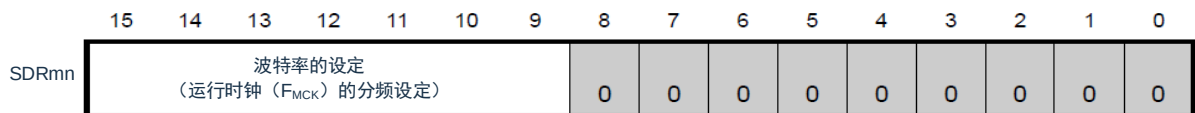


(b) 串行通信运行设定寄存器mn(SCRmn)

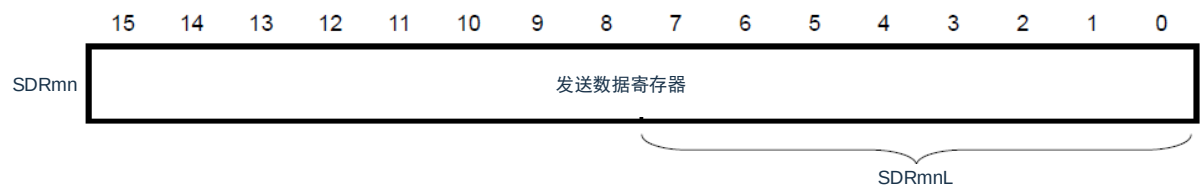


(c) 串行数据寄存器mn(SDRmn)

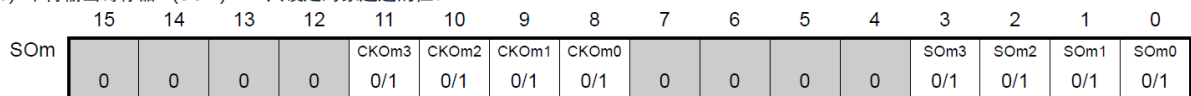
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)

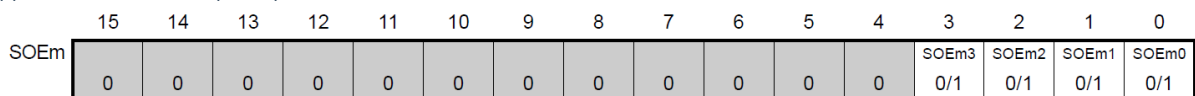


(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。

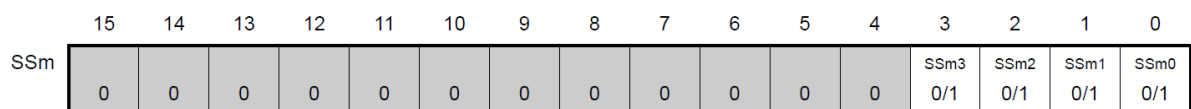


当时钟相位为“正相”(SCRmn寄存器的CKPmn=0)时,“1”为开始通信;当时钟相位为“反相”(CKPmn=1)时,“0”为开始通信。

(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1、2、3) mn=00~03、10~11

■: 不能设定(设定初始值)。0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-5: 主控发送的初始设置步骤



图 13-6: 主控发送的中止步骤

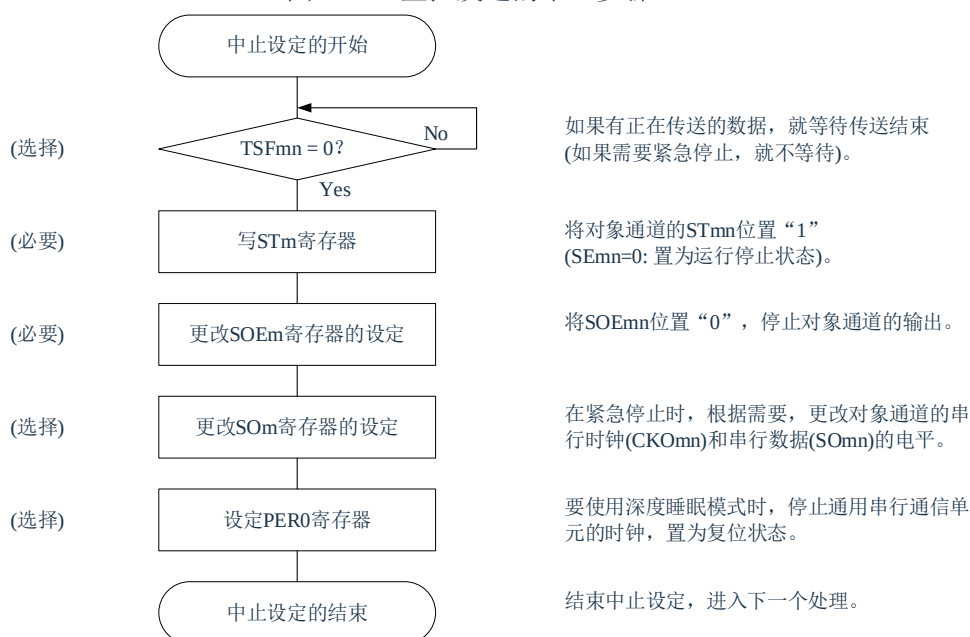
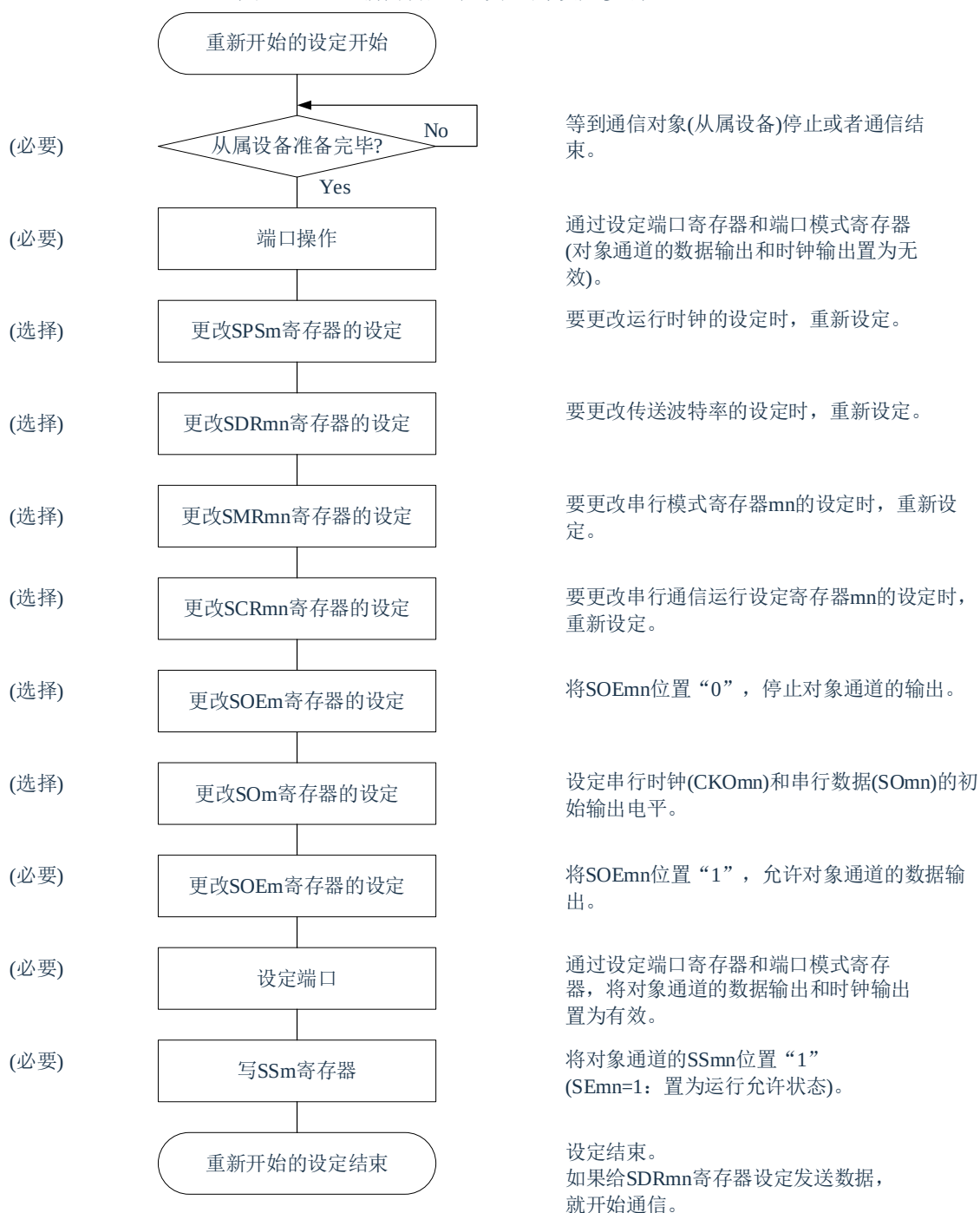


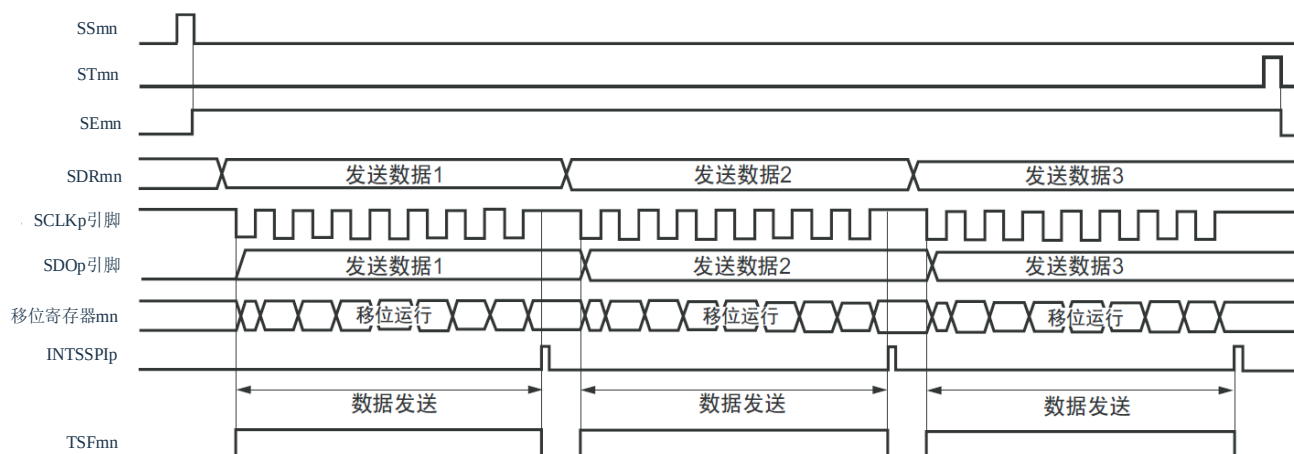
图 13-7: 重新开始主控发送的设定步骤



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

(3) 处理流程（单次发送模式）

图 13-8：主控发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

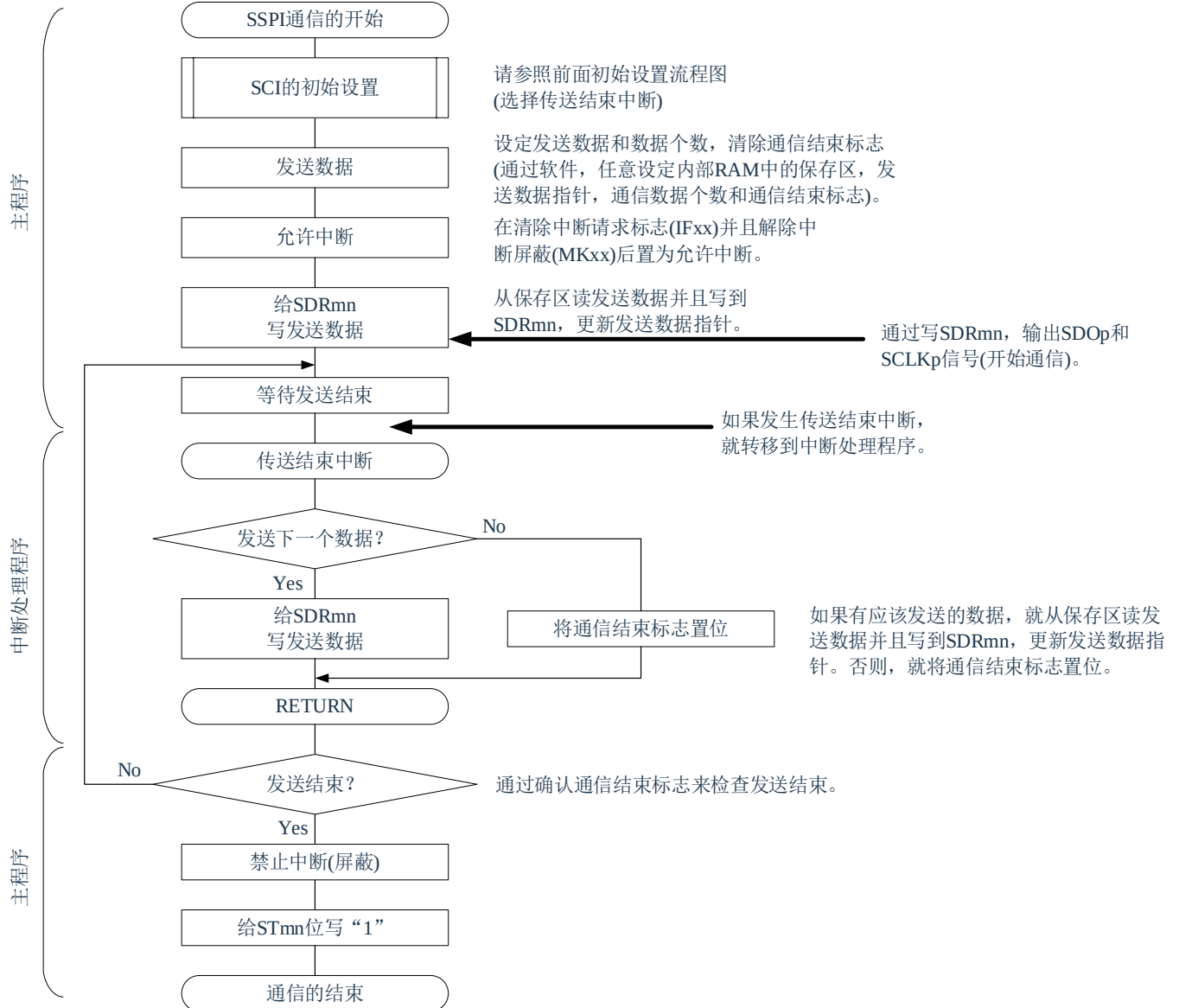


备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI号（p=00、01、10、11、20、21）。

图 13-9: 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 13-10：主控发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

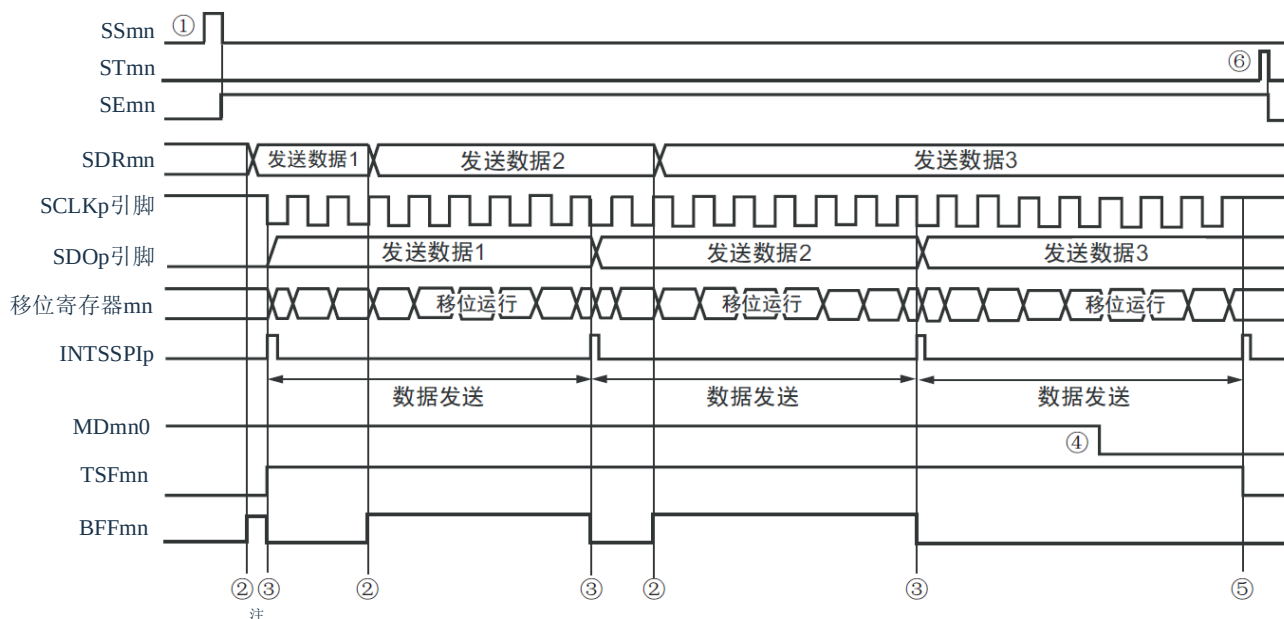
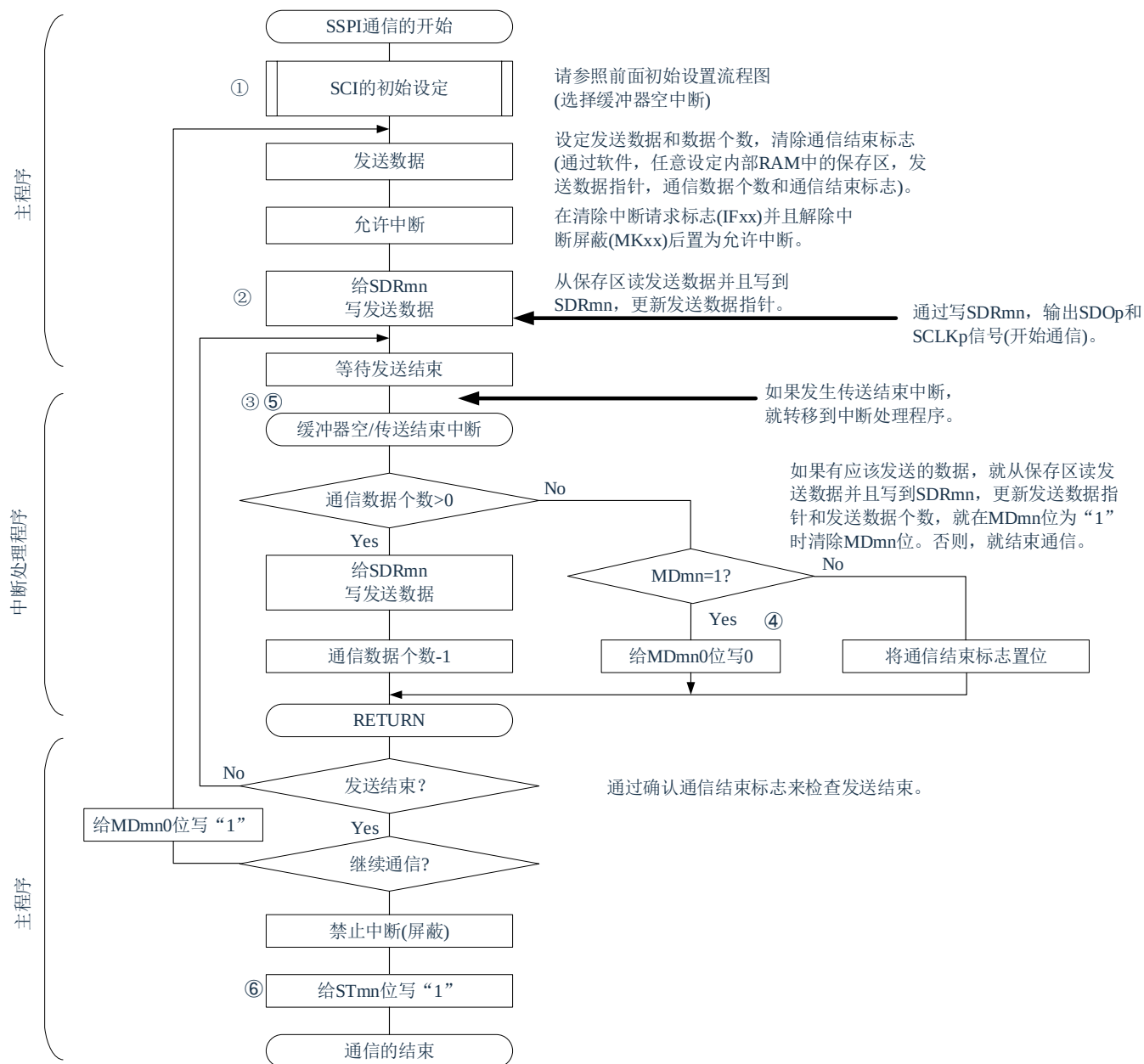


图 13-11: 主控发送（连续发送模式）的流程图



备注: 图中的①~⑥对应“图 13-10 主控发送（连续发送模式）的时序图”中的①~⑥。

13.7.2 主控接收

主控接收是指本产品输出传送时钟并且从其他设备接收数据的运行。

表 13-23: 主控接收

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	SCLKOI00、SDI00	SCLKOI01、SDI01	SCLKOI10、SDI10	SCLKOI11、SDI11	SCLKOI20、SDI20	SCLKOI21、SDI21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位、8 位					
传送速率 ^注	Max.6M[Hz] Min.FCLK/(2x2 ¹¹ x128)[Hz]FCLK: 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0: 正相 CKPmn=1: 反相					
数据方向	MSB 优先或者 LSB 优先					

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

备注: m: 单元号 (m=0、1);

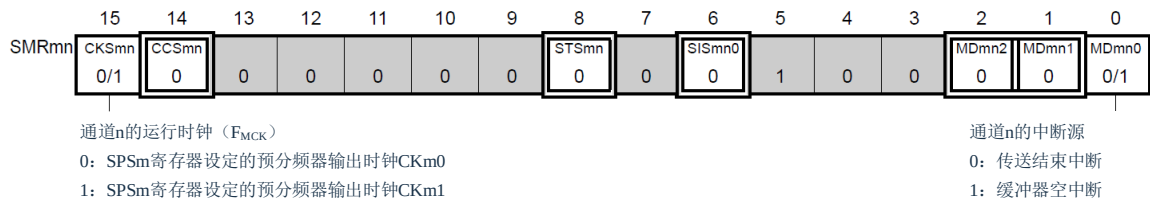
n: 通道号 (n=0、1、2、3);

p: SSPI 号 (p=00、01、10、11、20、21)。

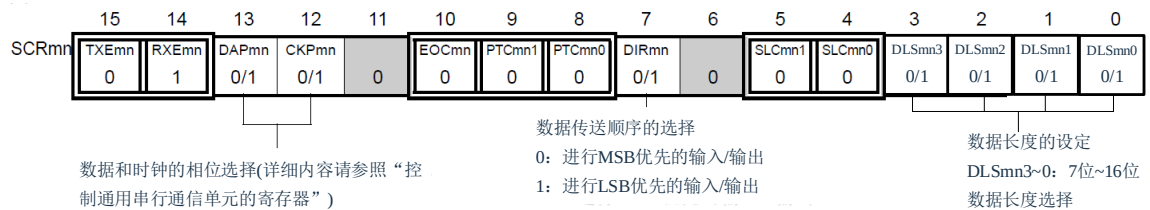
(1) 寄存器的设定

图 13-12: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
主控接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

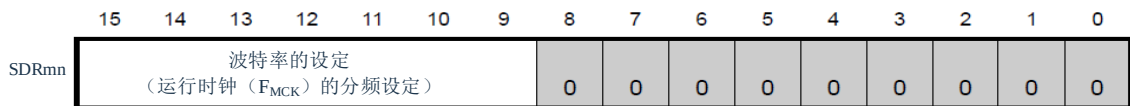


(b) 串行通信运行设定寄存器mn(SCRmn)

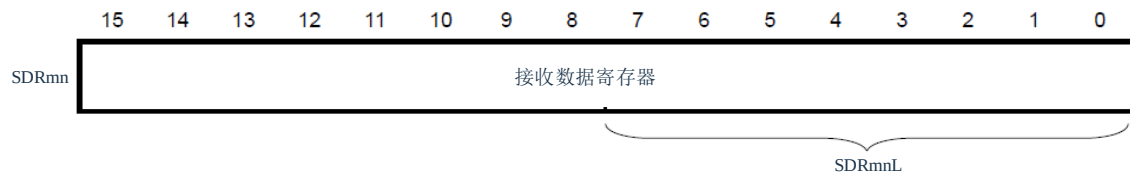


(c) 串行数据寄存器mn(SDRmn)

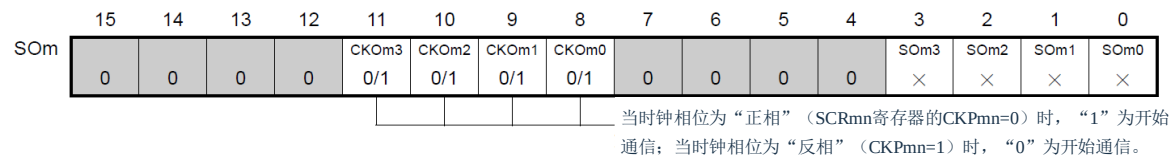
(1) 运行停止 (SEmn=0) 时



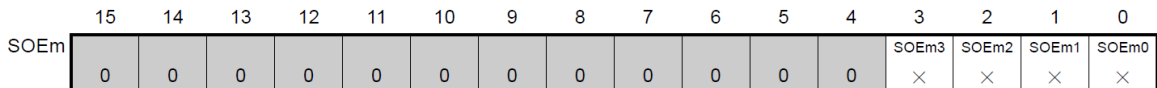
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



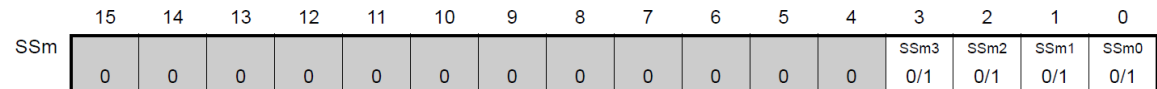
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……在此模式中不使用



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1、2、3);

p: SSPI 号 (p=00、01、10、11、20、21);

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-13: 主控接收的初始设定步骤

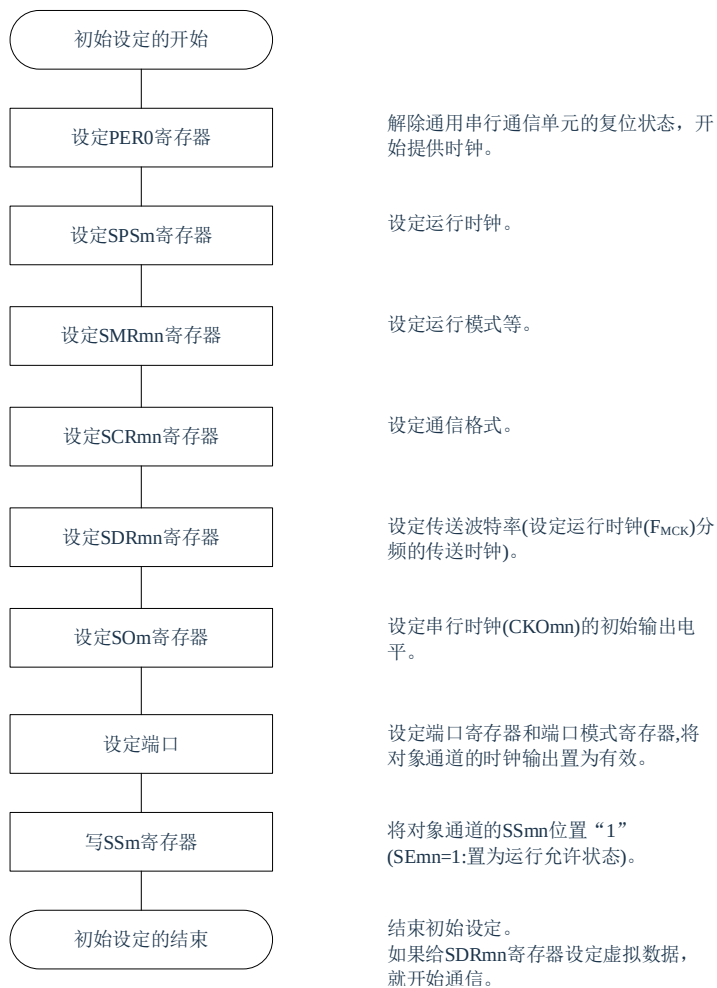


图 13-14：主控接收的中止步骤

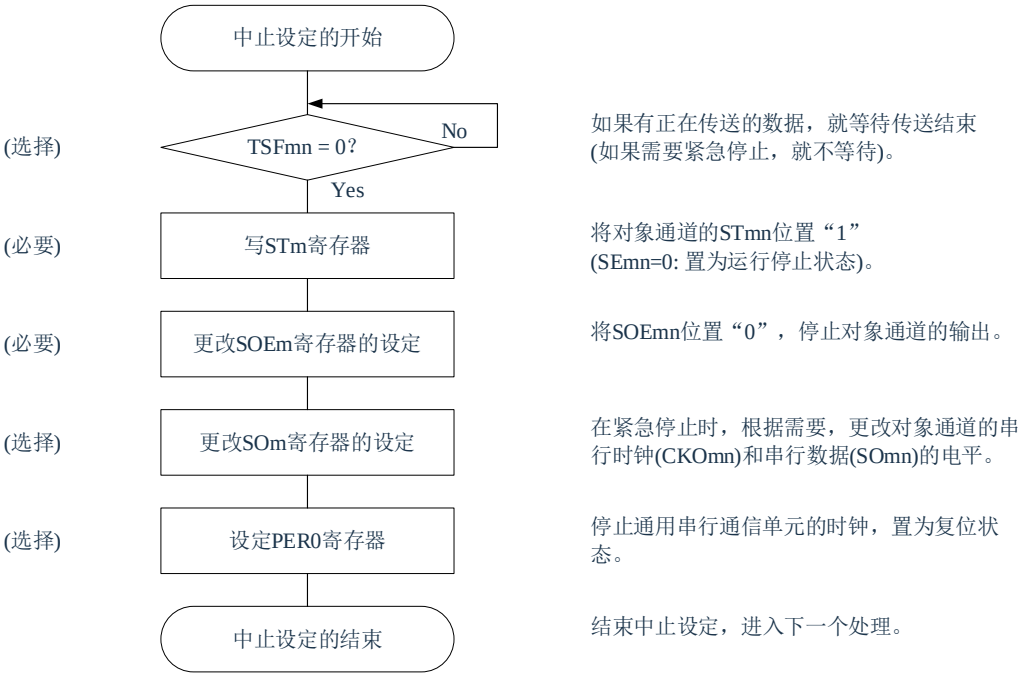
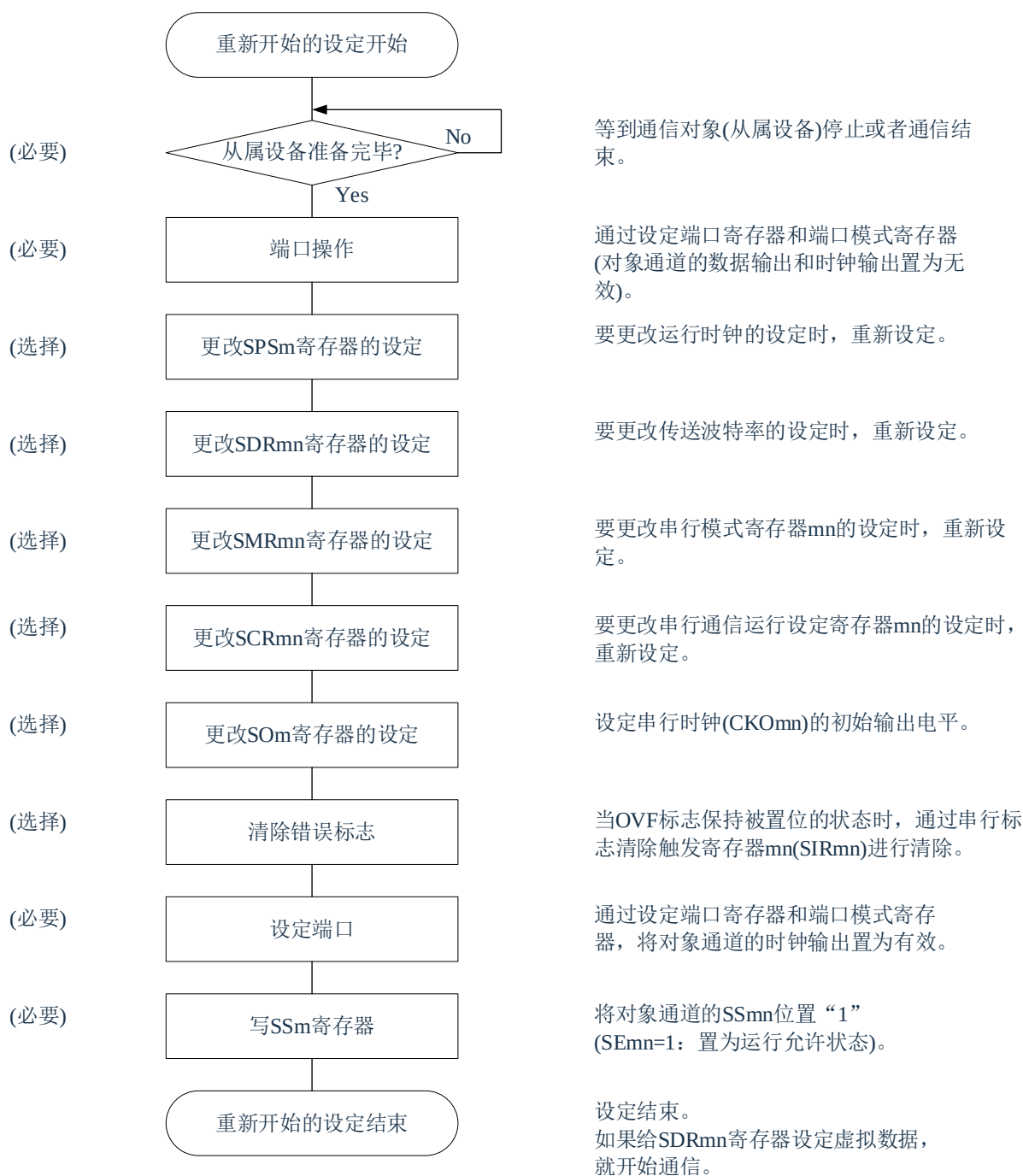


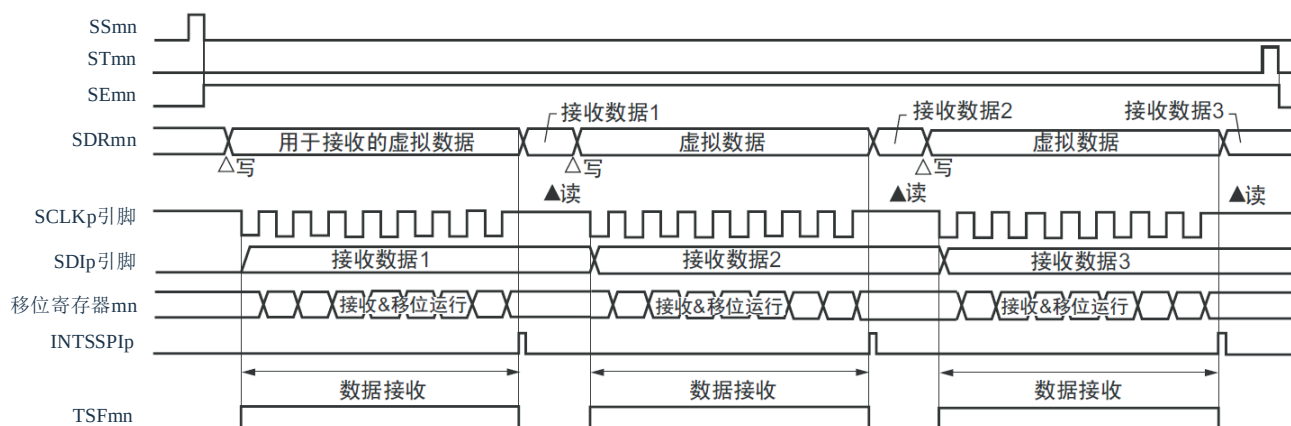
图 13-15: 重新开始主控接收的设定步骤



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

(3) 处理流程（单次接收模式）

图 13-16：主控接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

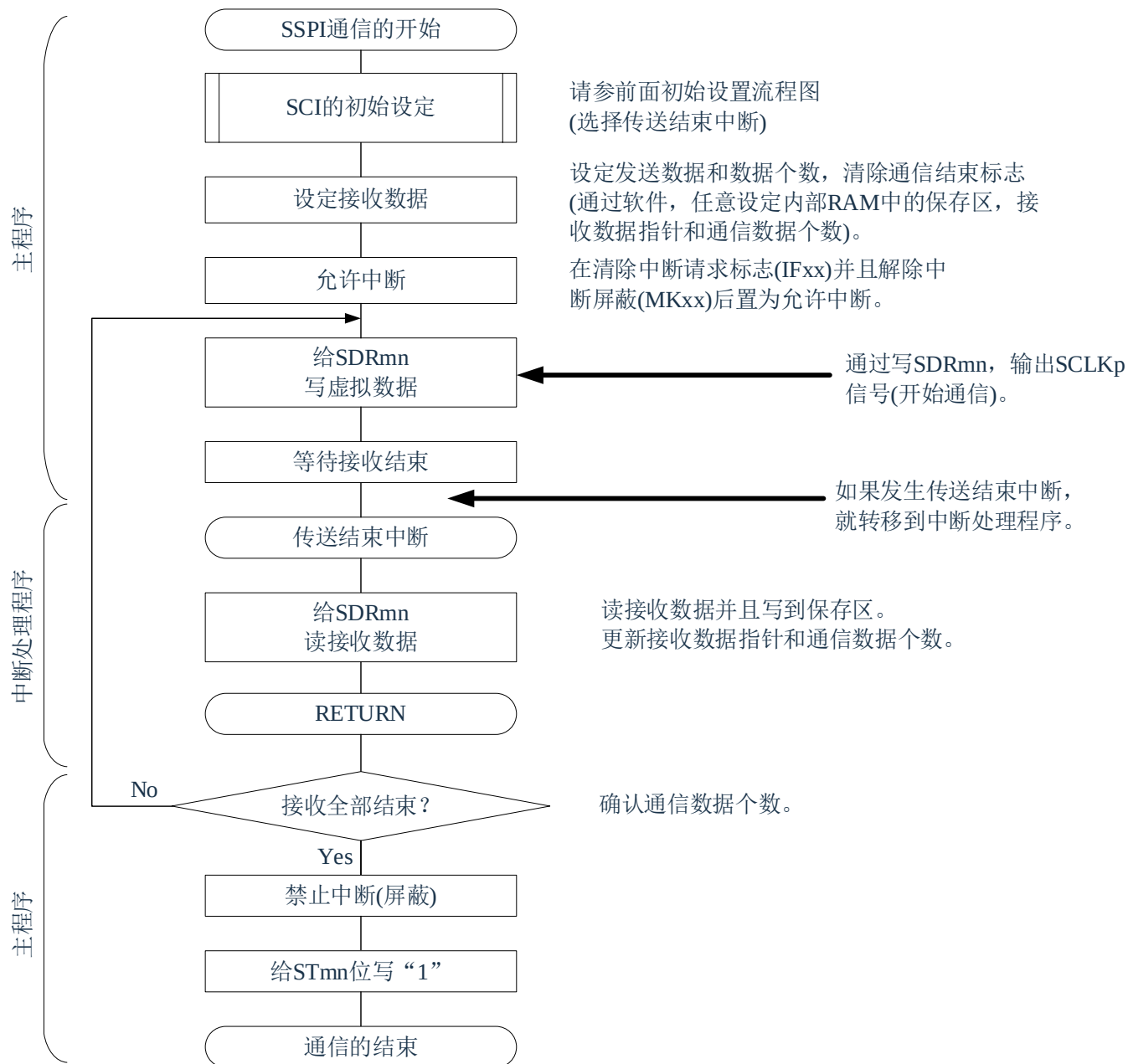


备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

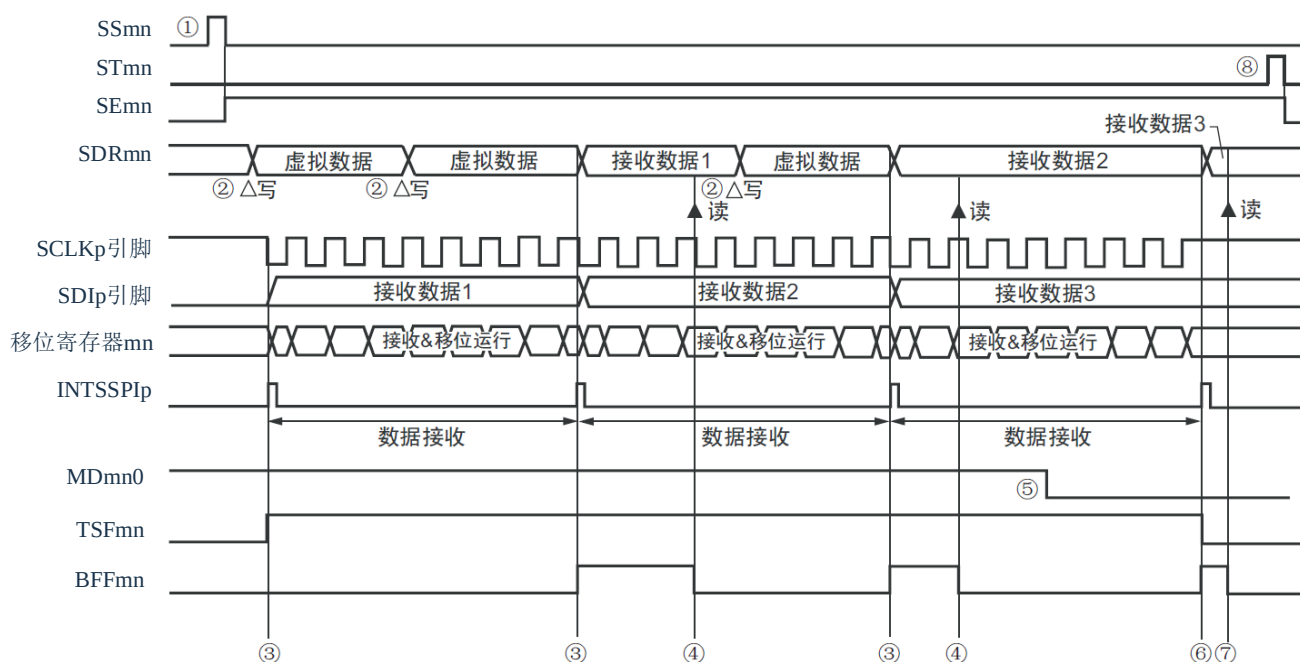
p：SSPI 号（p=00、01、10、11、20、21）。

图 13-17: 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 13-18：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：

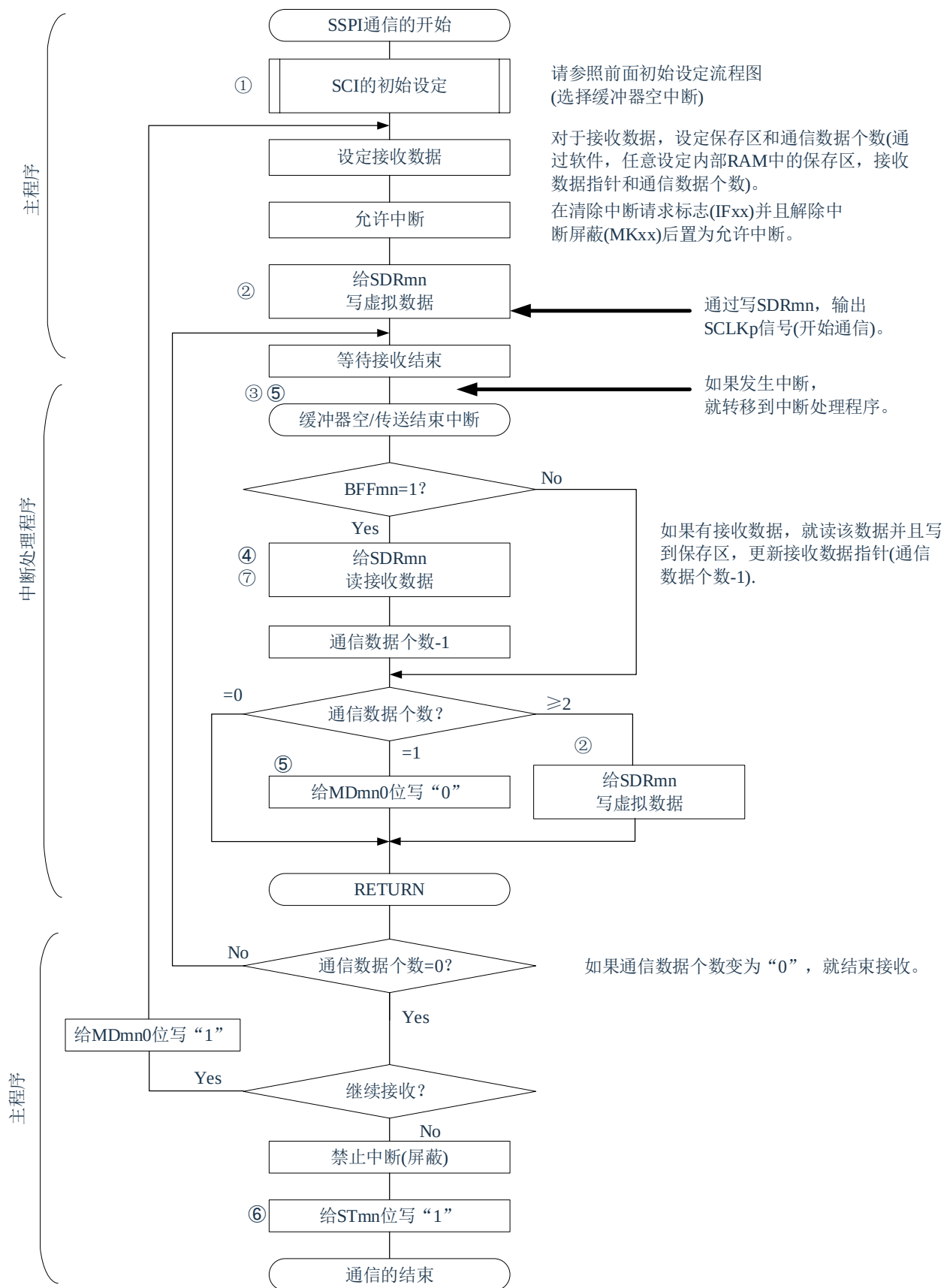
图中的①~⑧对应“图 13-19 主控接收（连续接收模式）的流程图”中的①~⑧；

m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

图 13-19: 主控接收（连续接收模式）的流程图



13.7.3 主控的发送和接收

主控的发送和接收是指本产品输出传送时钟并且和其他设备进行数据发送和接收的运行。

表 13-24: 主控的发送和接收

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	SCLKOI00、SDI00、SDO00	SCLKOI01、SDI01、SDO01	SCLKOI10、SDI10、SDO10	SCLKOI11、SDI11、SDO11	SCLKOI20、SDI20、SDO20	SCLKOI21、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、8 位					
传送速率 ^注	Max.6M[Hz] Min.F _{CLK} /(2x2 ¹¹ x128)[Hz]F _{CLK} : 系统时钟频率					
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。					
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 CKPmn=0: 正相 CKPmn=1: 反相					
数据方向	MSB 优先或者 LSB 优先					

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

备注: m: 单元号 (m=0、1);

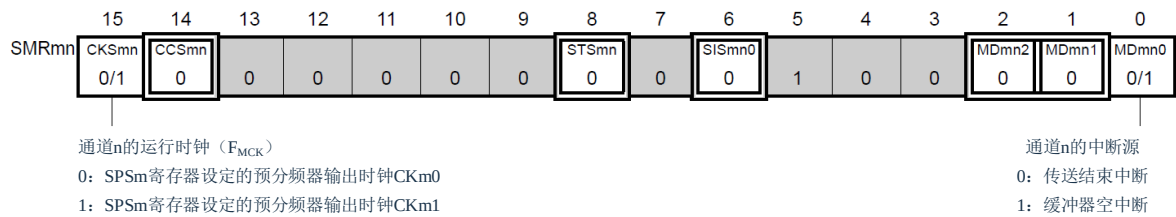
n: 通道号 (n=0、1、2、3);

p: SSPI 号 (p=00、01、10、11、20、21)。

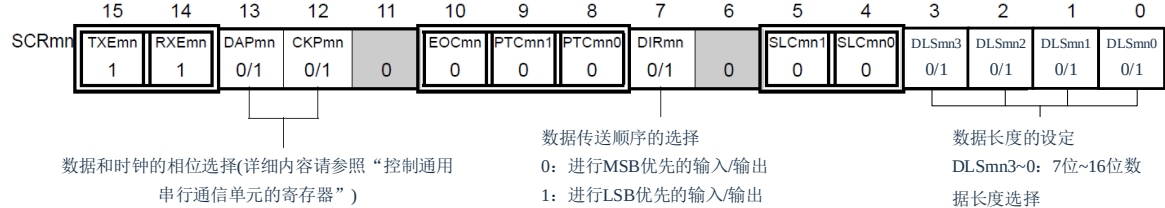
(1) 寄存器的设定

图 13-20: 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定内容例子

(a) 串行模式寄存器mn(SMRmn)

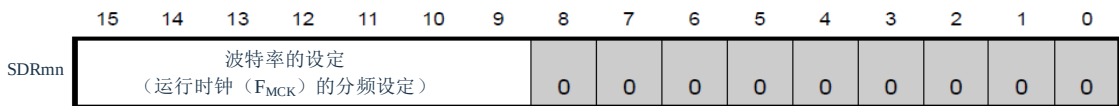


(b) 串行通信运行设定寄存器mn(SCRmn)

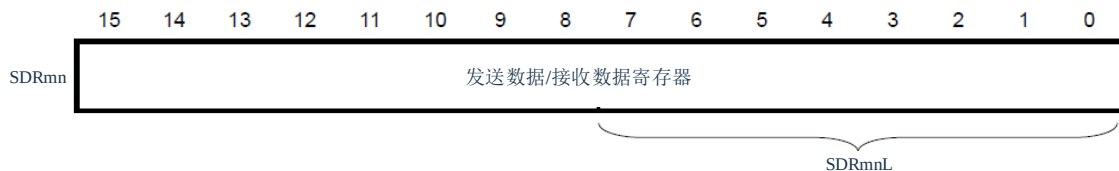


(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止 (SEmn=0) 时



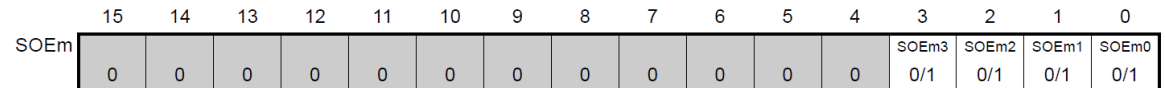
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



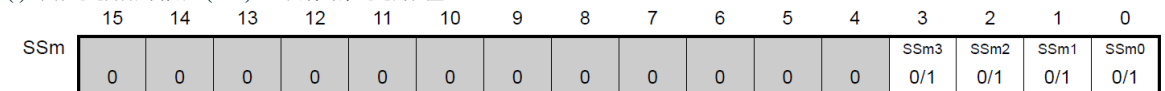
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1、2、3);

p: SSPI 号 (p=00、01、10、11、20、21);

□: 在 SSPI 主控接收模式中为固定设定;

■ 不能设定 (设定初始值);

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-21: 主控发送和接收的初始设定步骤



图 13-22: 主控发送和接收的中止步骤

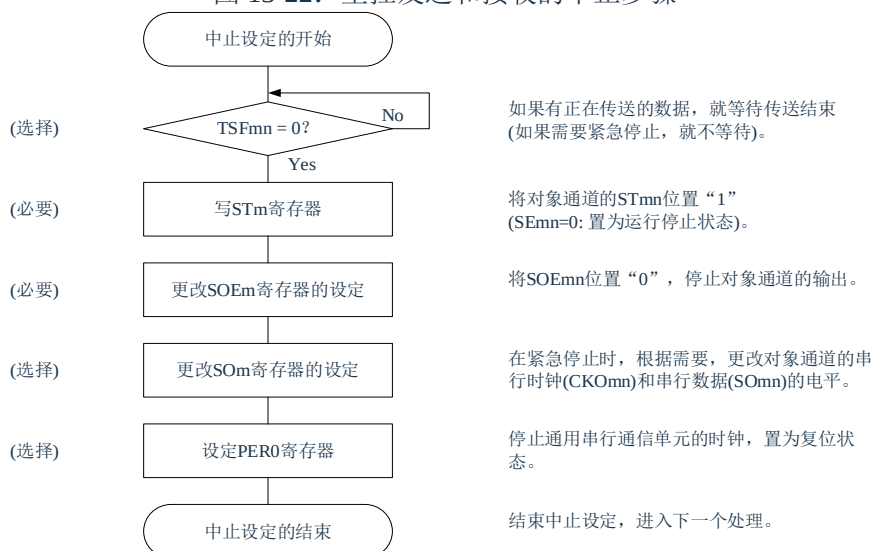
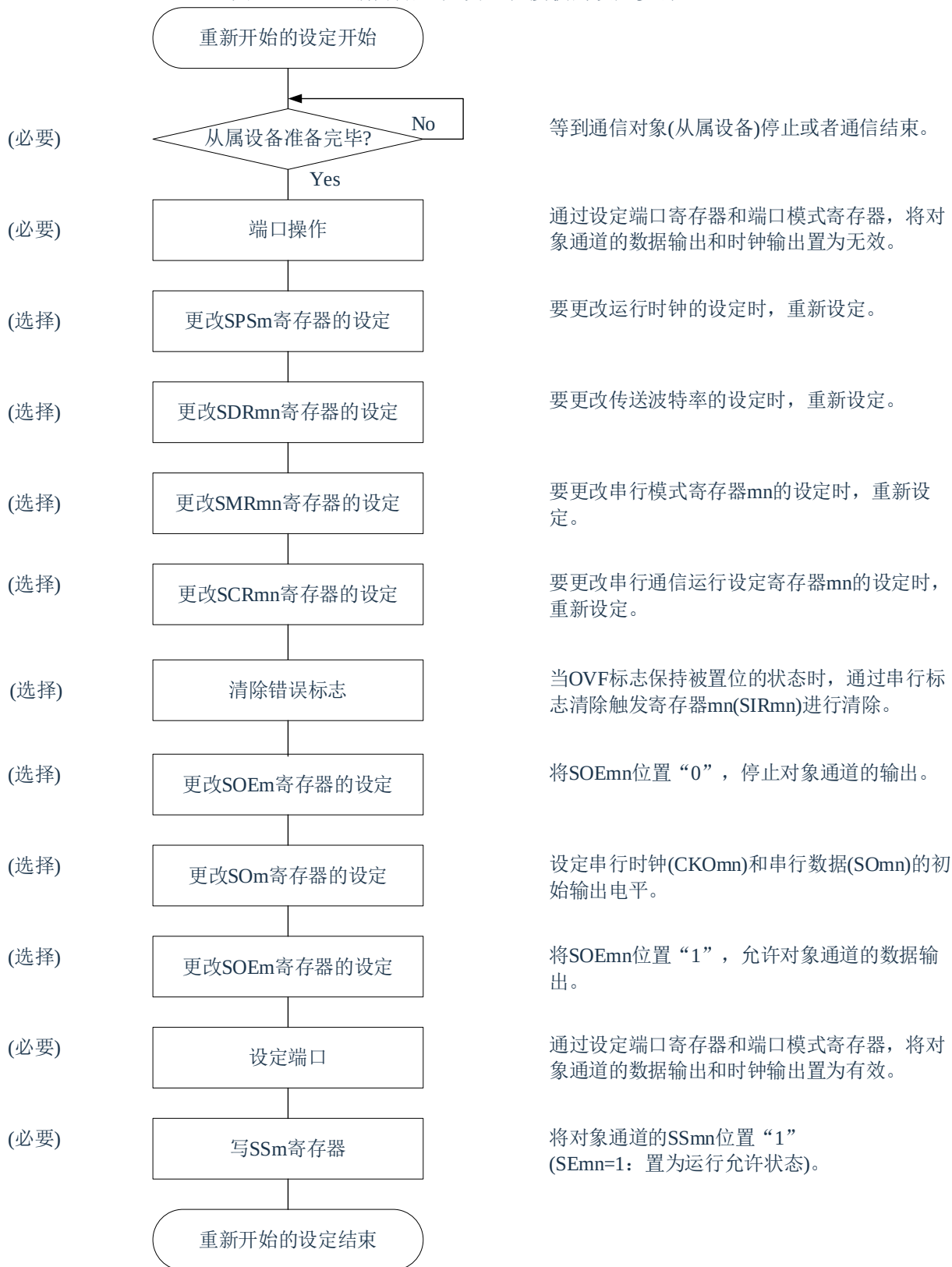
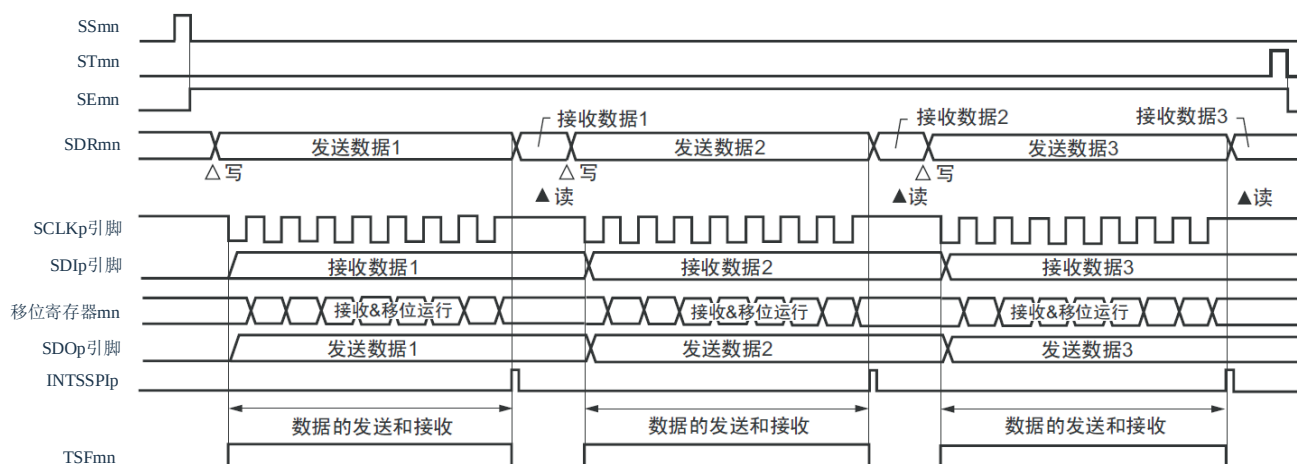


图 13-23: 重新开始主控发送和接收的设定步骤



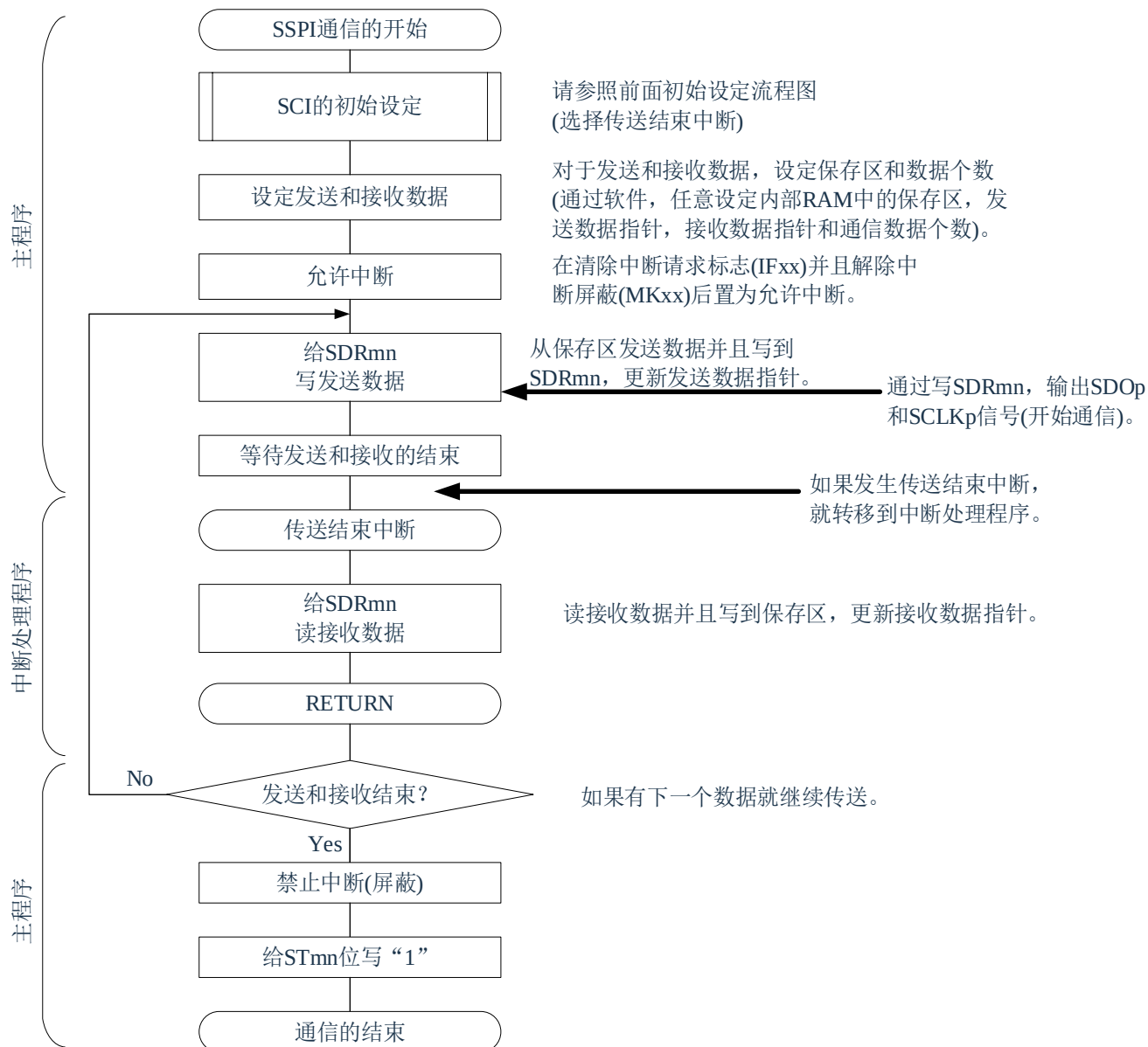
(3) 处理流程（单次发送和接收模式）

图 13-24：主控发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



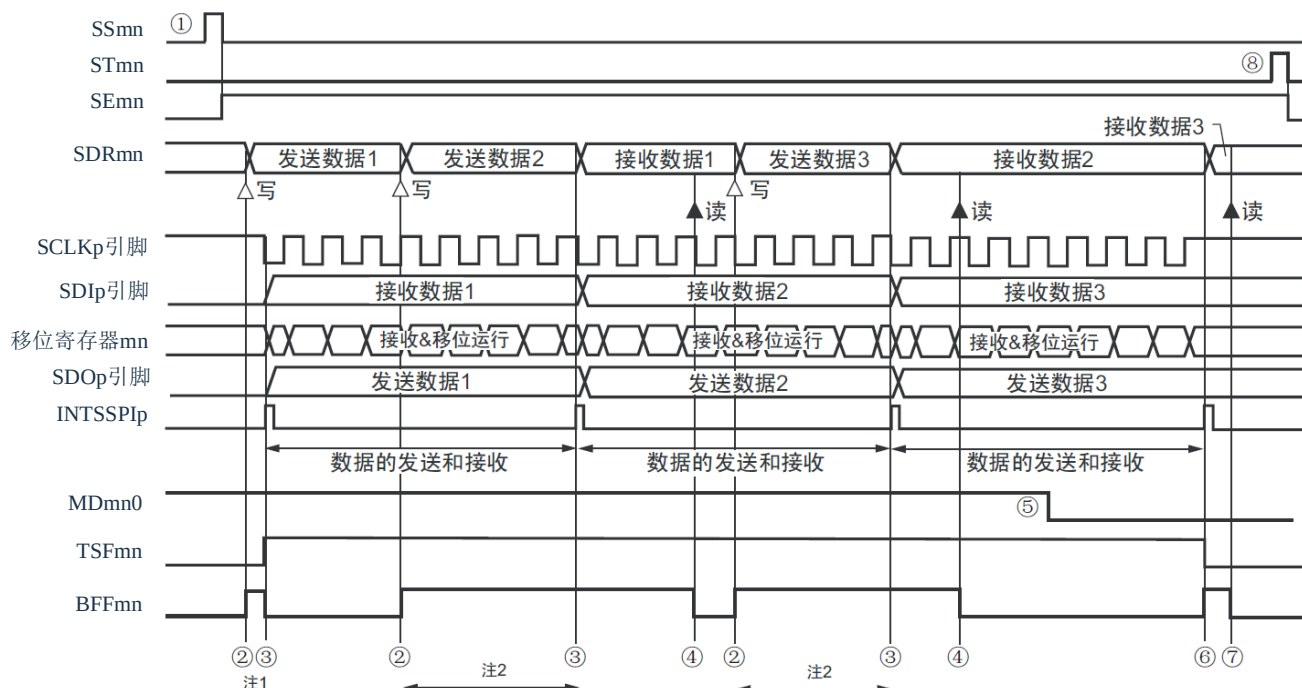
备注：m：单元号（m=0、1）；n：通道号（n=0、1、2、3）；p：SSPI号（p=00、01、10、11、20、21）。

图 13-25: 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 13-26：主控发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

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

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

备注：

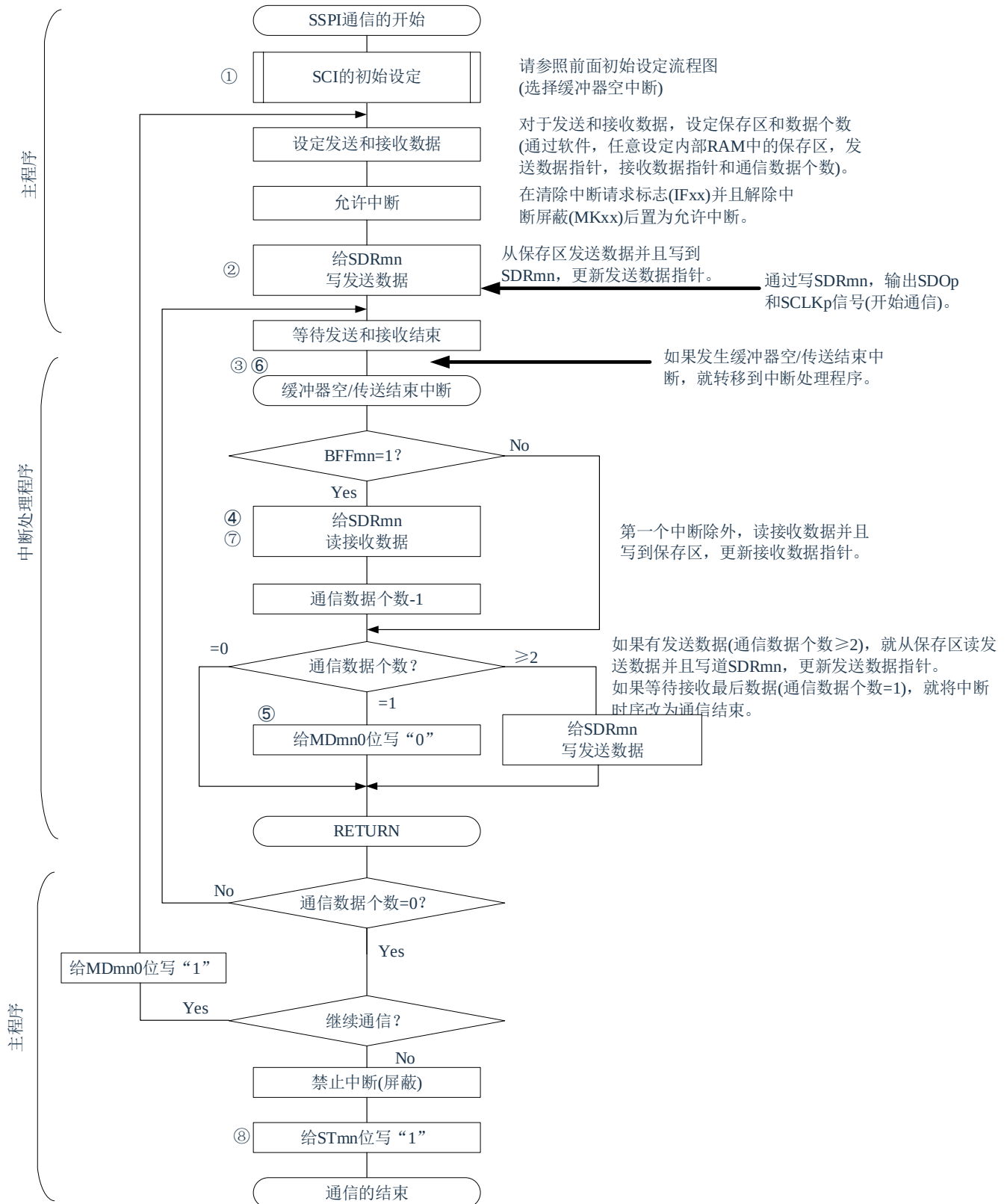
图中的①~⑧对应“图 13-27 主控发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

图 13-27: 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 13-26 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

13.7.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下微控制器将数据发送到其他设备的运行。

表 13-25：从属发送

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	SCLKOI00、SDO00	SCLKOI01、SDO01	SCLKOI10、SDO10	SCLKOI11、SDO11	SCLKOI20、SDO20	SCLKOI21、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、8位					
传送速率	Max.6M[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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11、SCLKOI20、SCLKOI21 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 6M[Hz]；

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

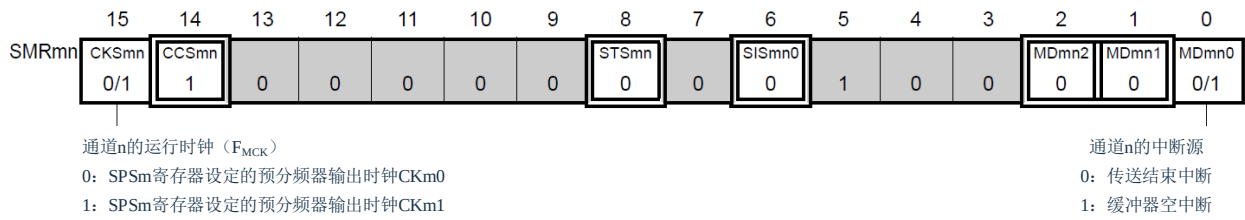
备注：

1. FMCK：对象通道的运行时钟频率；
2. m：单元号（m=0、1）；
n：通道号（n=0、1、2、3）。

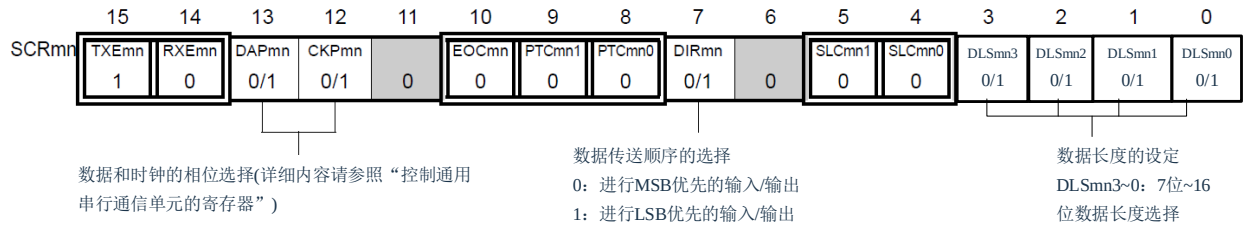
(1) 寄存器的设定

图 13-28: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
从属发送时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

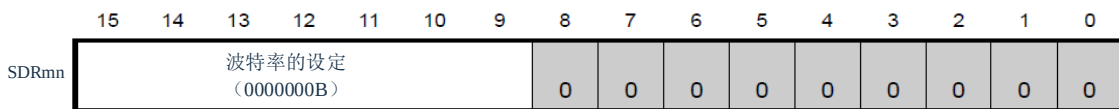


(b) 串行通信运行设定寄存器mn(SCRmn)

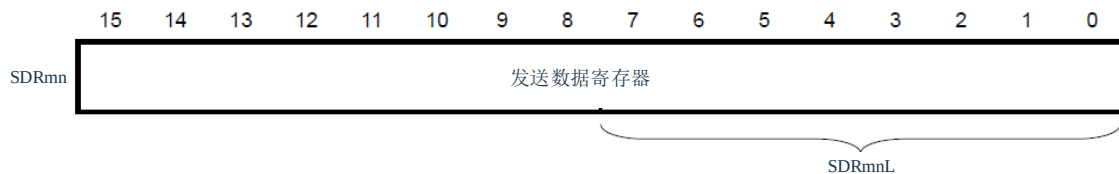


(c) 串行数据寄存器mn(SDRmn)

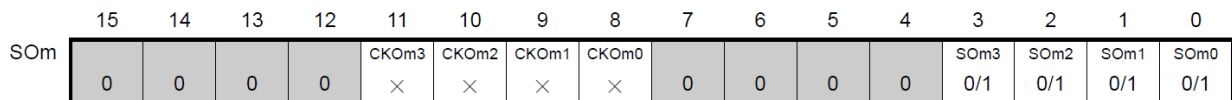
(1) 运行停止 (SEmn=0) 时



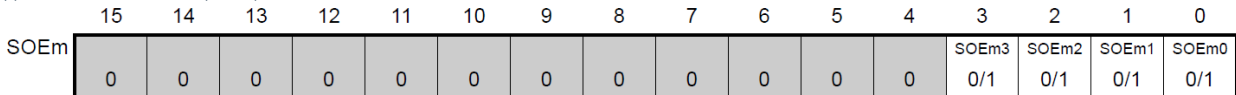
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



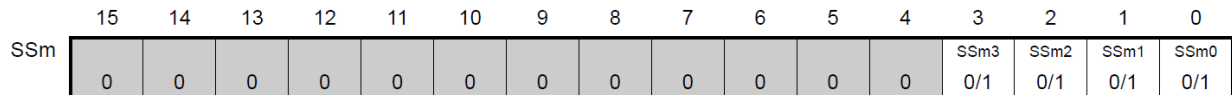
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



备注: m: 单元号 (m=0、1);

n: 通道号 (n=0、1、2、3);

p: SSPI 号 (p=00、01、10、11、20、21);

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-29: 从属发送的初始设定步骤



图 13-30: 从属发送的中止步骤

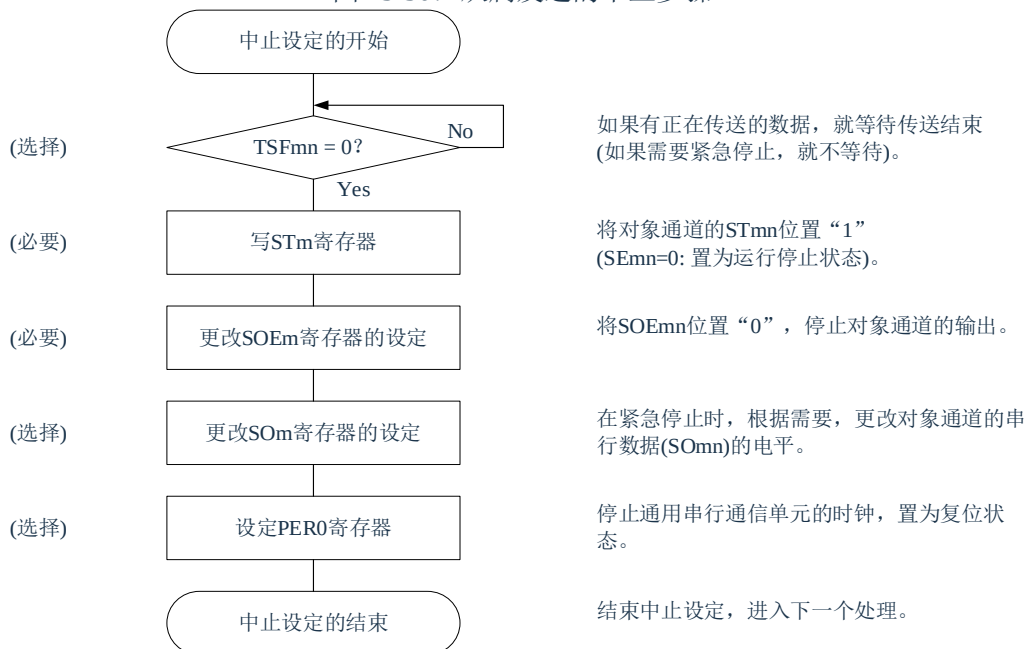
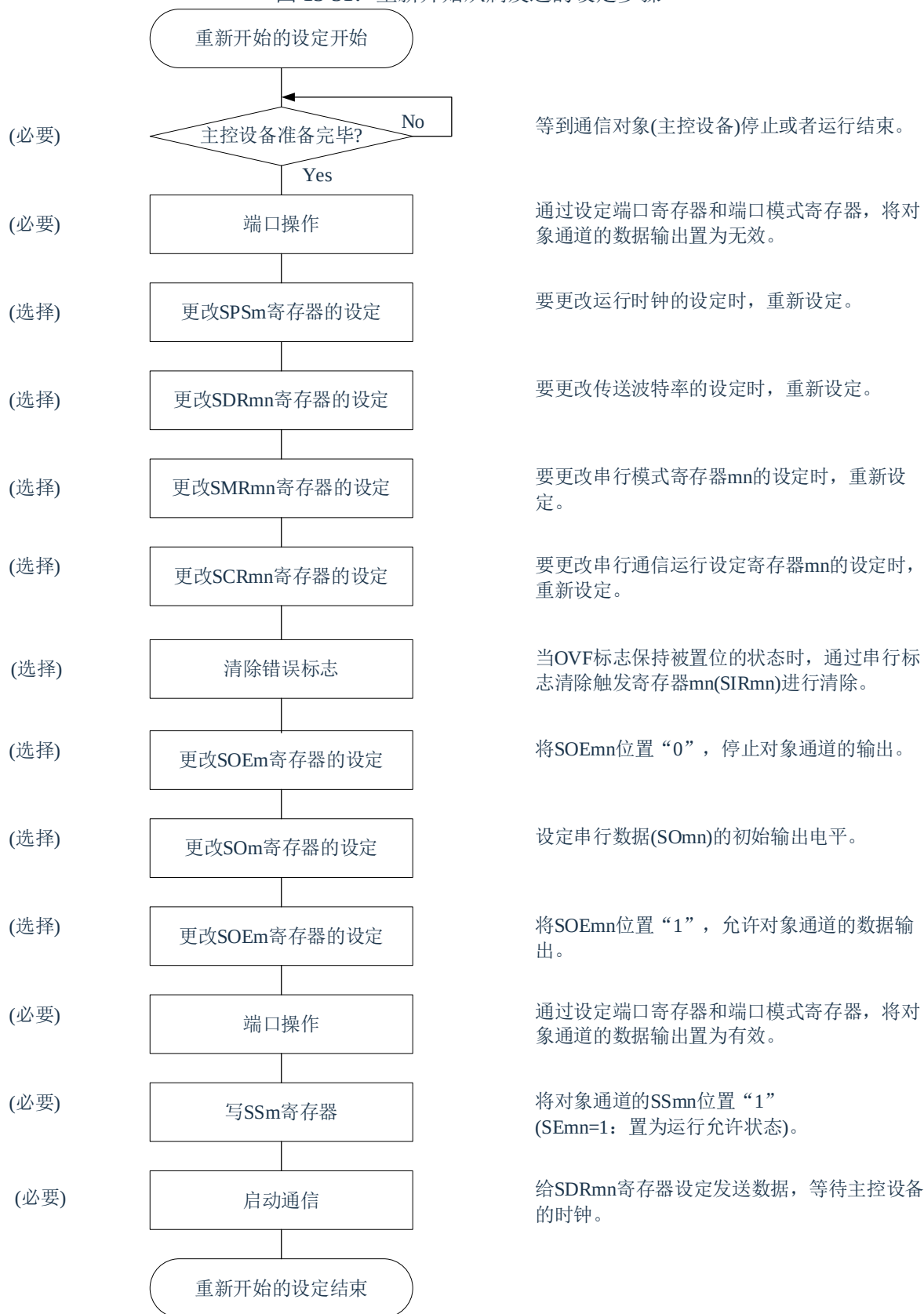


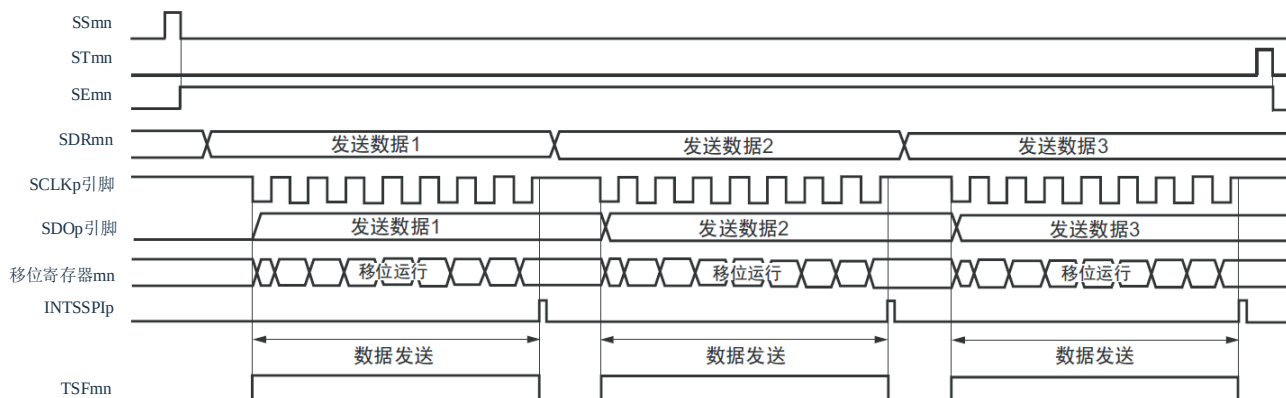
图 13-31：重新开始从属发送的设定步骤



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

(3) 处理流程（单次发送模式）

图 13-32：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

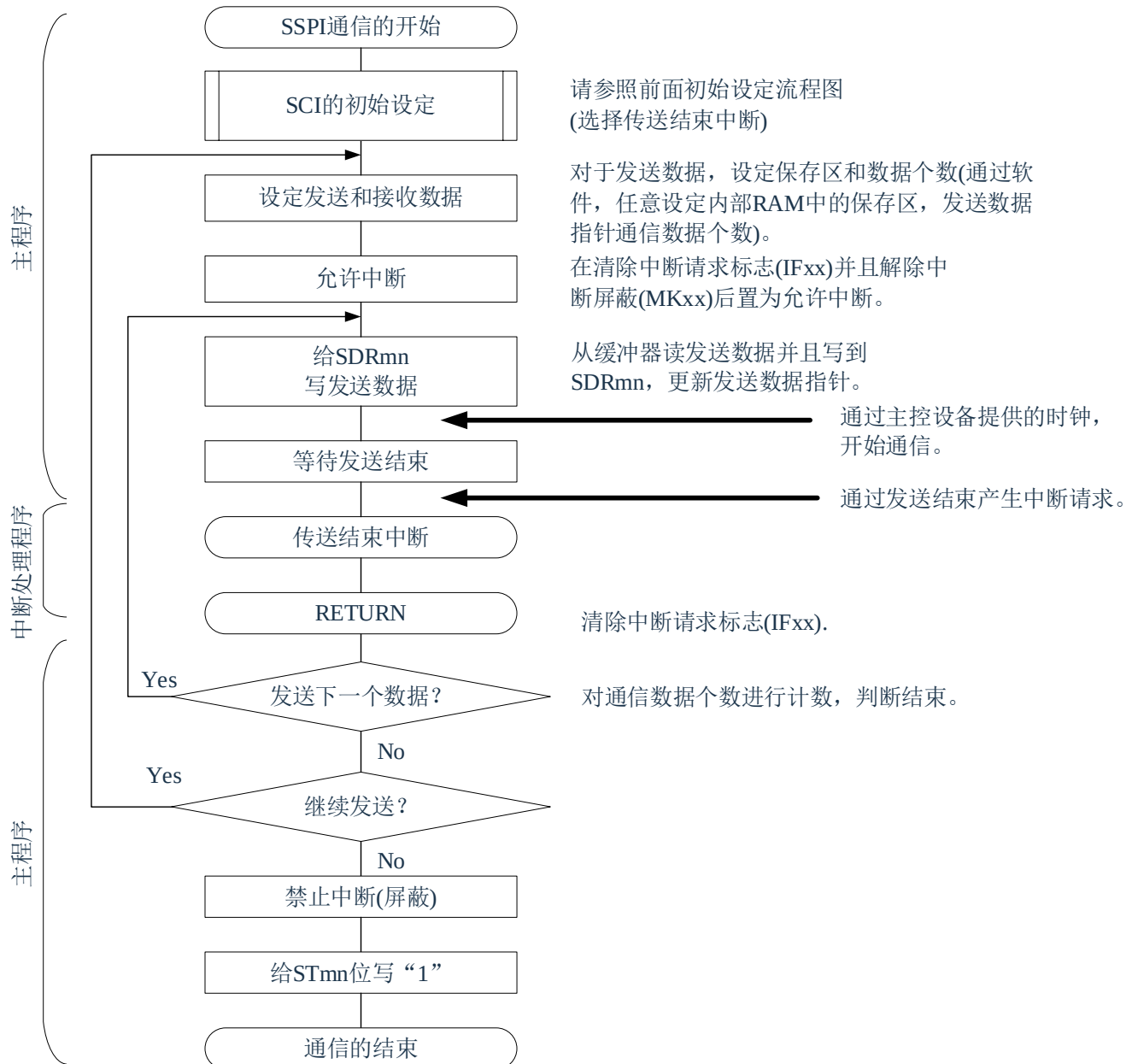


备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

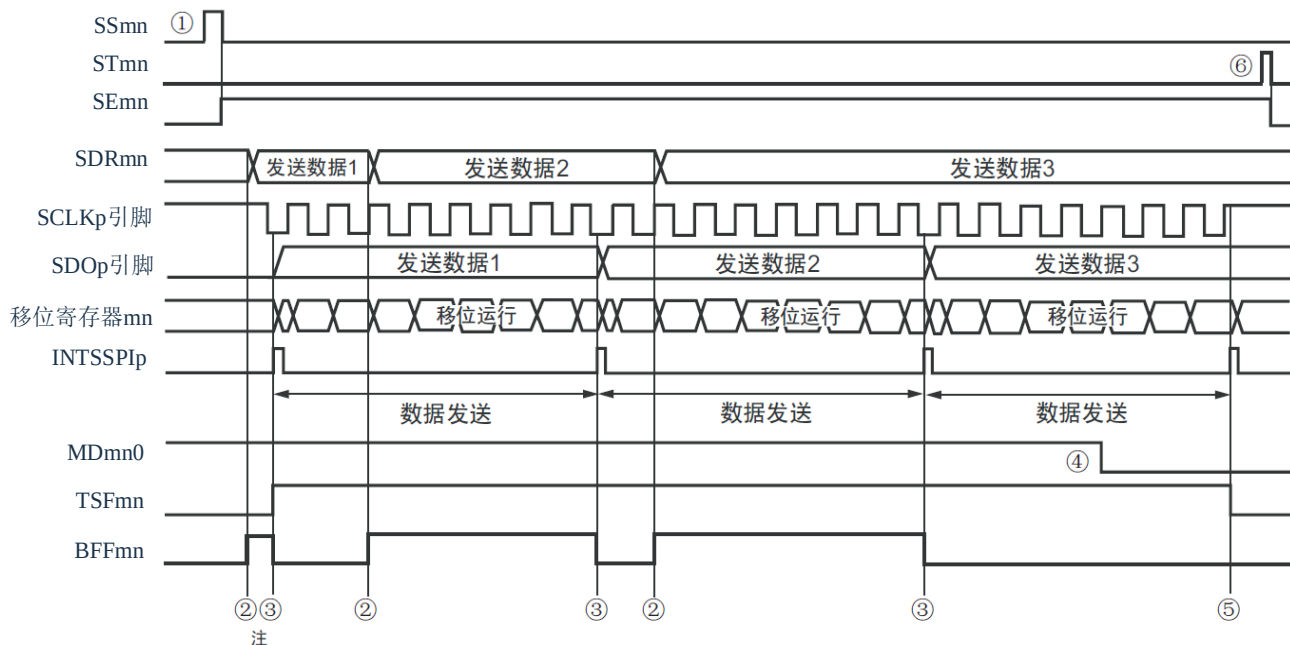
p：SSPI 号（p=00、01、10、11、20、21）。

图 13-33: 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 13-34：从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

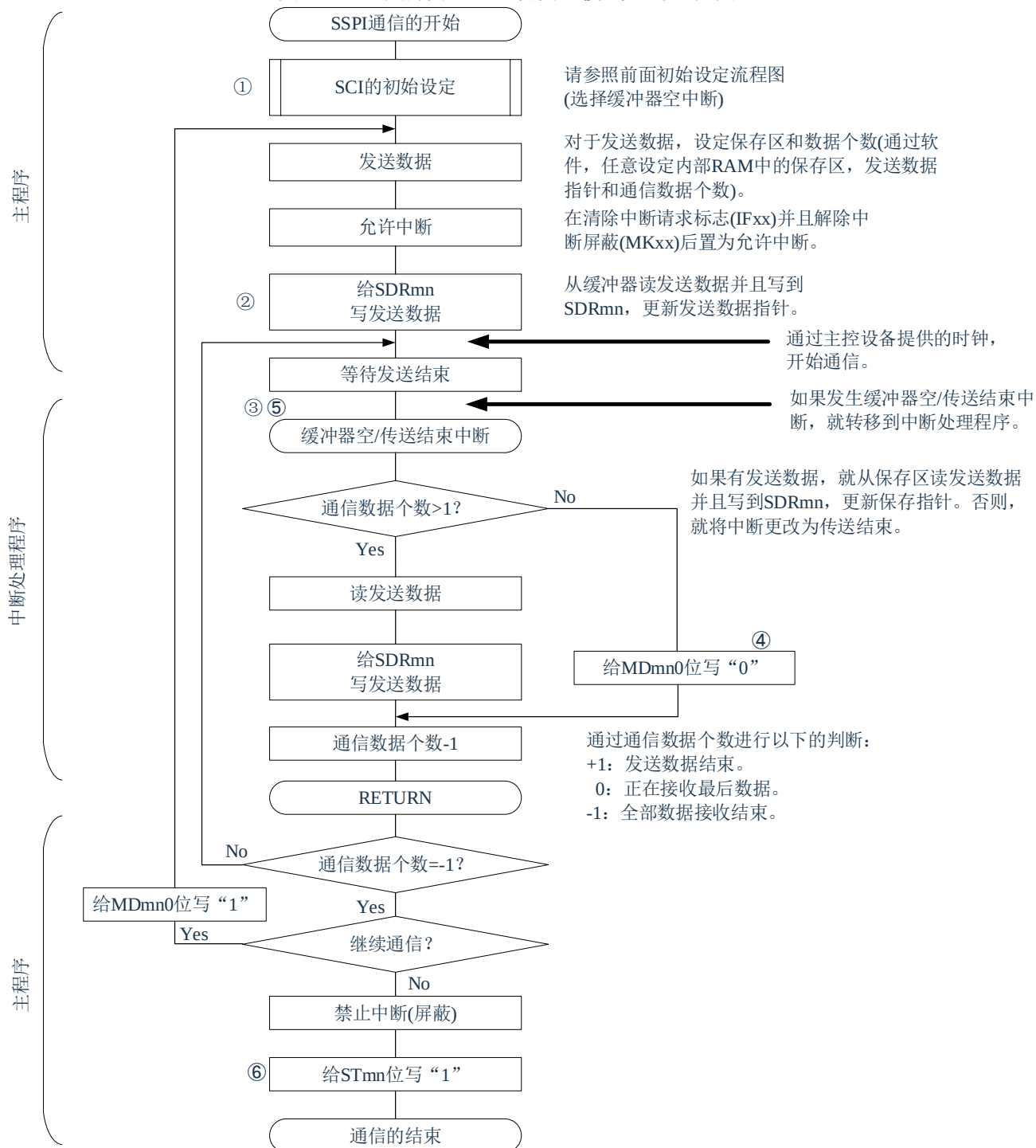
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

图 13-35：从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“

图13-34 从属发送（连续发送模式）的时序图”中的①~⑥。

13.7.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

表 13-26：从属接收

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	SCLKOI00、SDI00	SCLKOI01、SDI01	SCLKOI10、SDI10	SCLKOI11、SDI11	SCLKOI20、SDI20	SCLKOI21、SDI21
中断	INTSSPI00	INTSSPI01	INTSSPI10	INTSSPI11	INTSSPI20	INTSSPI21
	只限于传送结束中断（禁止设定缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7位、8位					
传送速率	Max.6M[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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11、SCLKOI20、SCLKOI21 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 6M[Hz]；

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

F_{MCK}：对象通道的运行时钟频率；

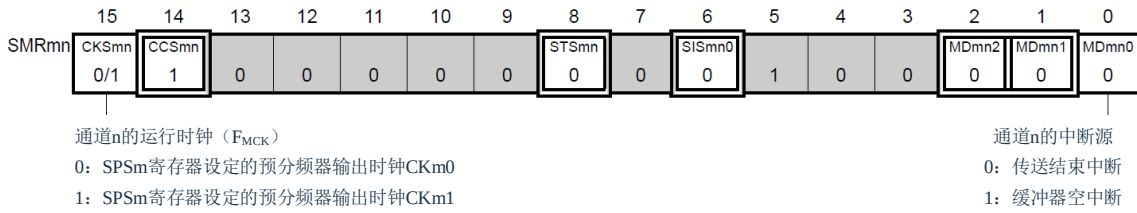
m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）。

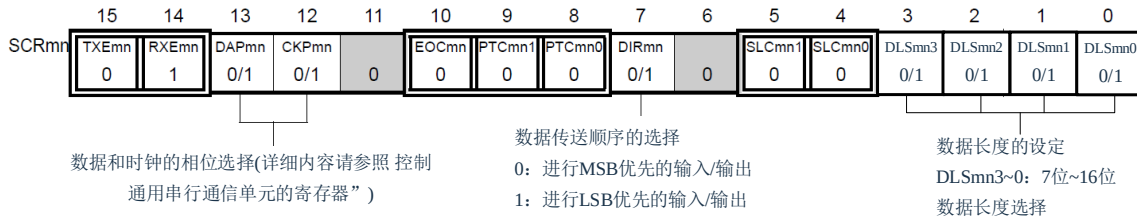
(1) 寄存器的设定

图 13-36: 3 线串行 I/O (SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21)
从属接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

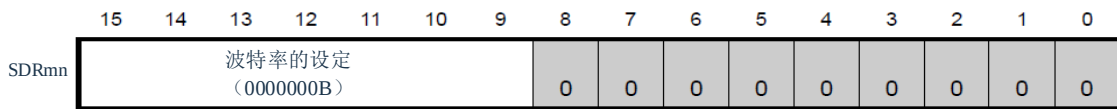


(b) 串行通信运行设定寄存器mn(SCRmn)

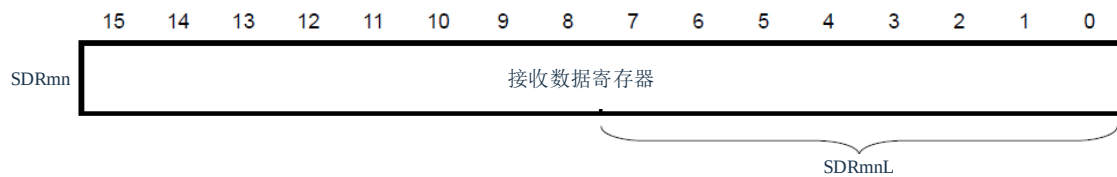


(c) 串行数据寄存器mn(SDRmn)

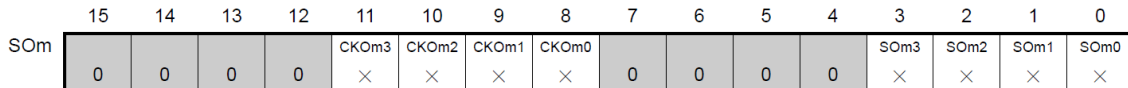
(1) 运行停止 (SEmn=0) 时



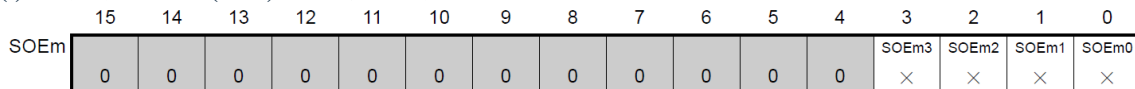
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



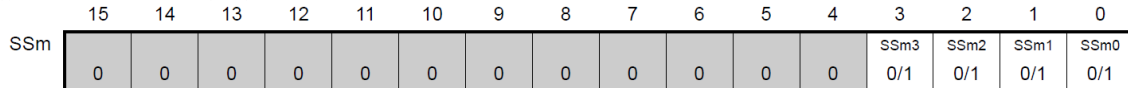
(d) 串行输出寄存器m(SOm) ……在此模式中不使用。



(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”。



备注: m: 单元号 (m=0、1) ;

n: 通道号 (n=0、1、2、3) ;

p: SSPI 号 (p=00、01、10、11、20、21) ;

□: 在 SSPI 主控接收模式中为固定设定;

■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-37：从属接收的初始设定步骤



图 13-38：从属接收的中止步骤

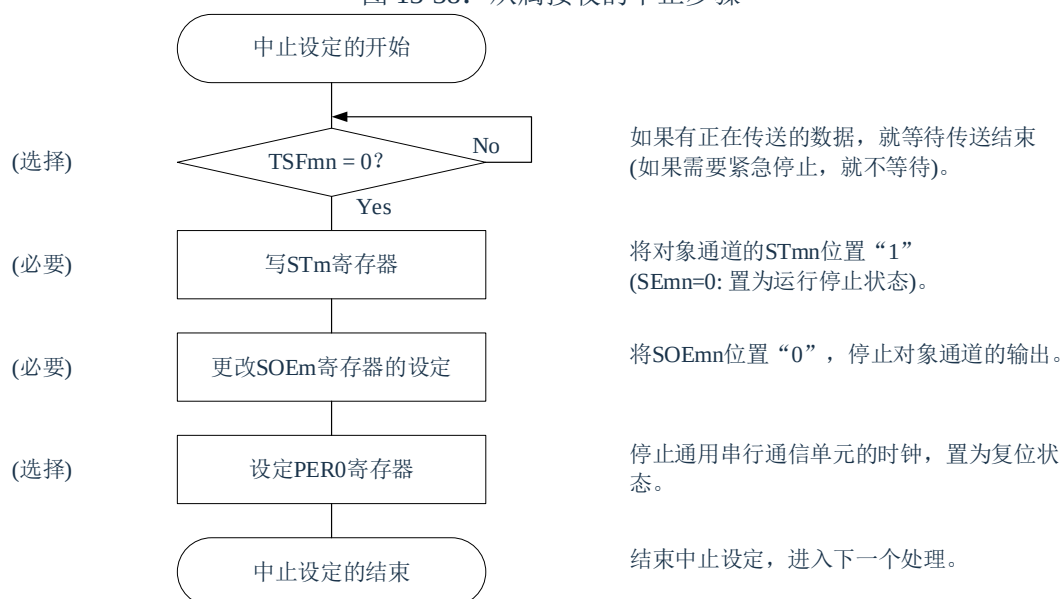


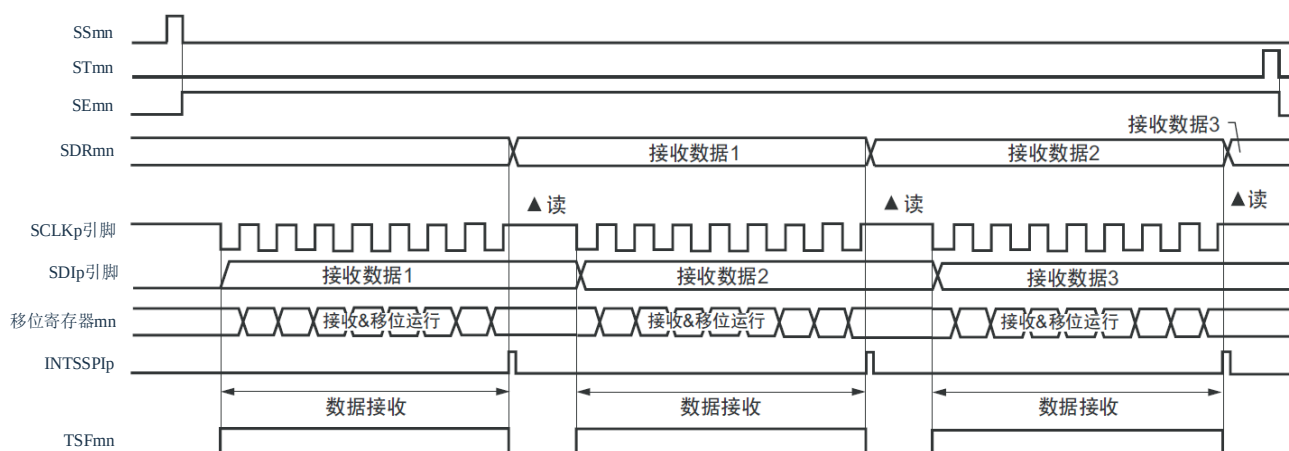
图 13-39: 重新开始从属接收的设定步骤



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

(3) 处理流程（单次接收模式）

图 13-40：从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

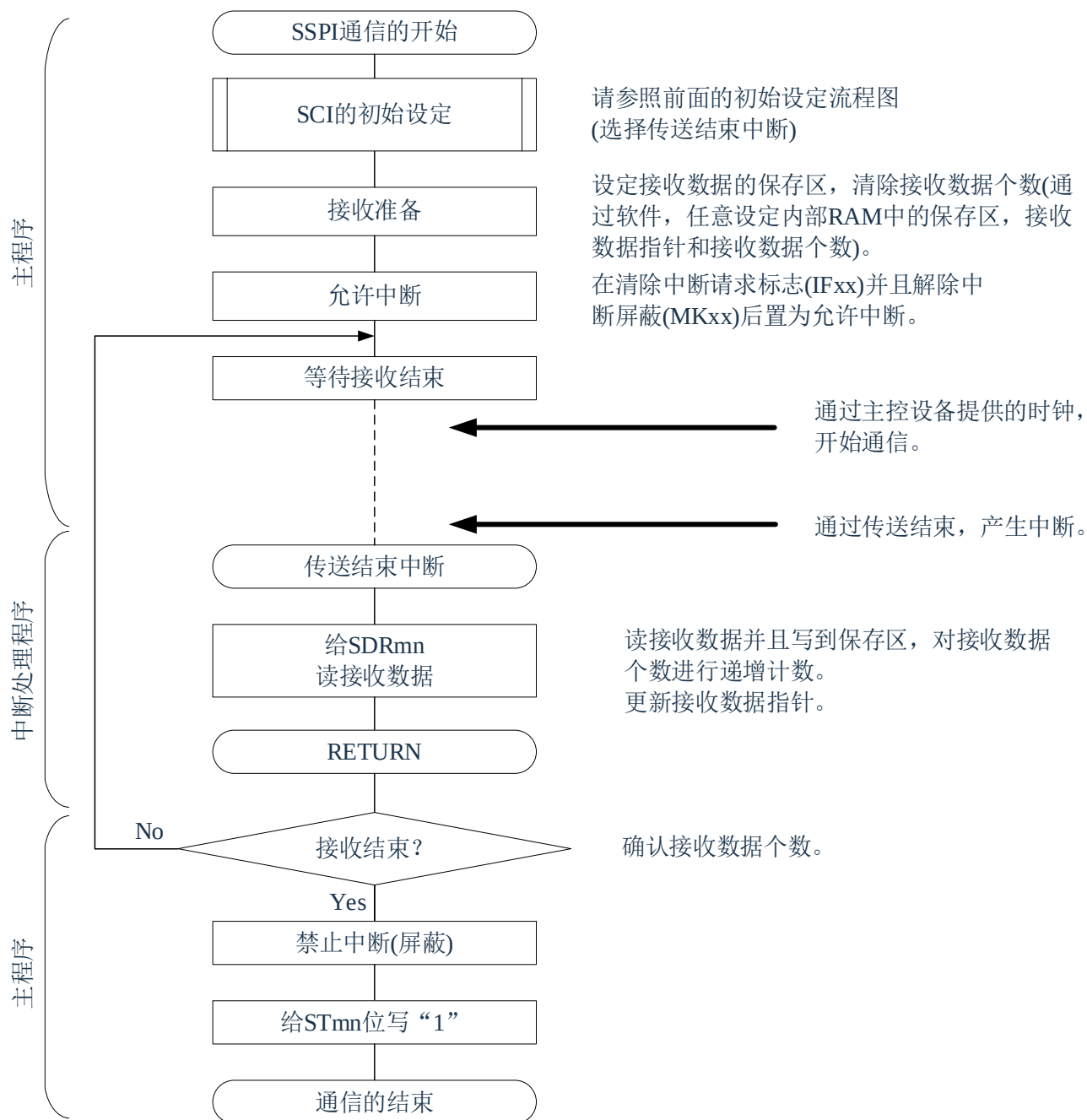


备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

图 13-41：从属接收（单次接收模式）的流程图



13.7.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品微控制器和其他设备进行数据发送和接收的运行。

表 13-27：从属的发送和接收

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	SCLKOI00、SDI00、SDO00	SCLKOI01、SDI01、SDO01	SCLKOI10、SDI10、SDO10	SCLKOI11、SDI11、SDO11	SCLKOI20、SDI20、SDO20	SCLKOI21、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、8位					
传送速率	Max.6M[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 SCLKOI00、SCLKOI01、SCLKOI10、SCLKOI11、SCLKOI20、SCLKOI21 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 6M[Hz]；

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

备注：

F_{MCK}：对象通道的运行时钟频率；

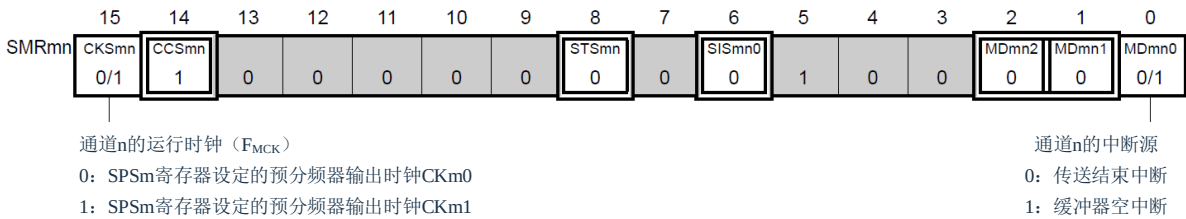
m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）。

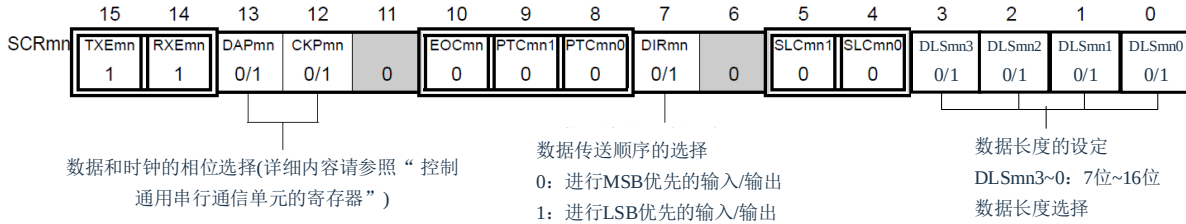
(1) 寄存器的设定

图 13-42：3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）
从属发送和接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

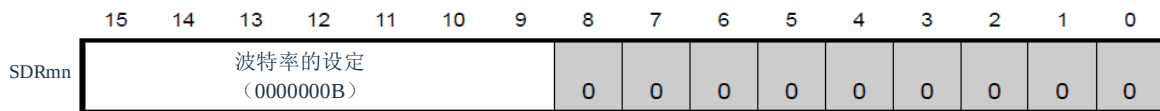


(b) 串行通信运行设定寄存器mn(SCRmn)

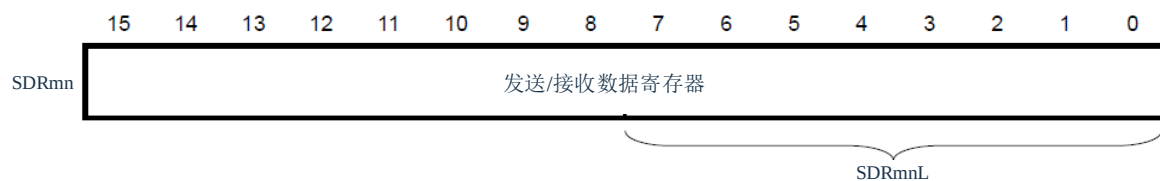


(c) 串行数据寄存器mn(SDRmn)

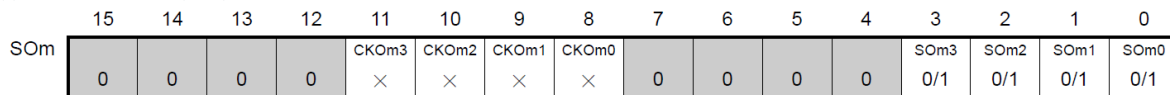
(1) 运行停止 (SEmn=0) 时



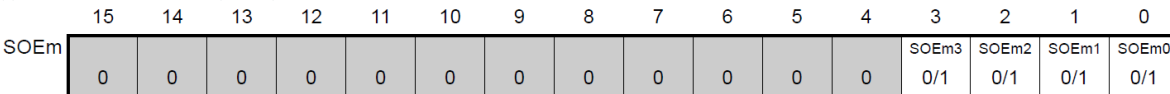
(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



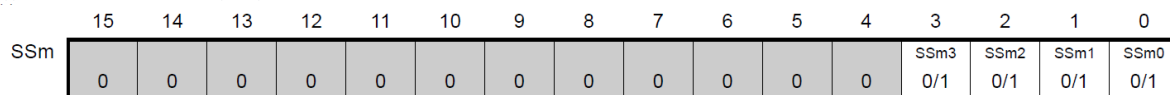
(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



注意：在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：m：单元号 (m=0、1)；

n：通道号 (n=0、1、2、3)；

p：SSPI 号 (p=00、01、10、11、20、21)；

□：在 SSPI 主控接收模式中为固定设定； ■ 不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-43: 从属发送和接收的初始设定步骤



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

图 13-44: 从属发送和接收的中止步骤

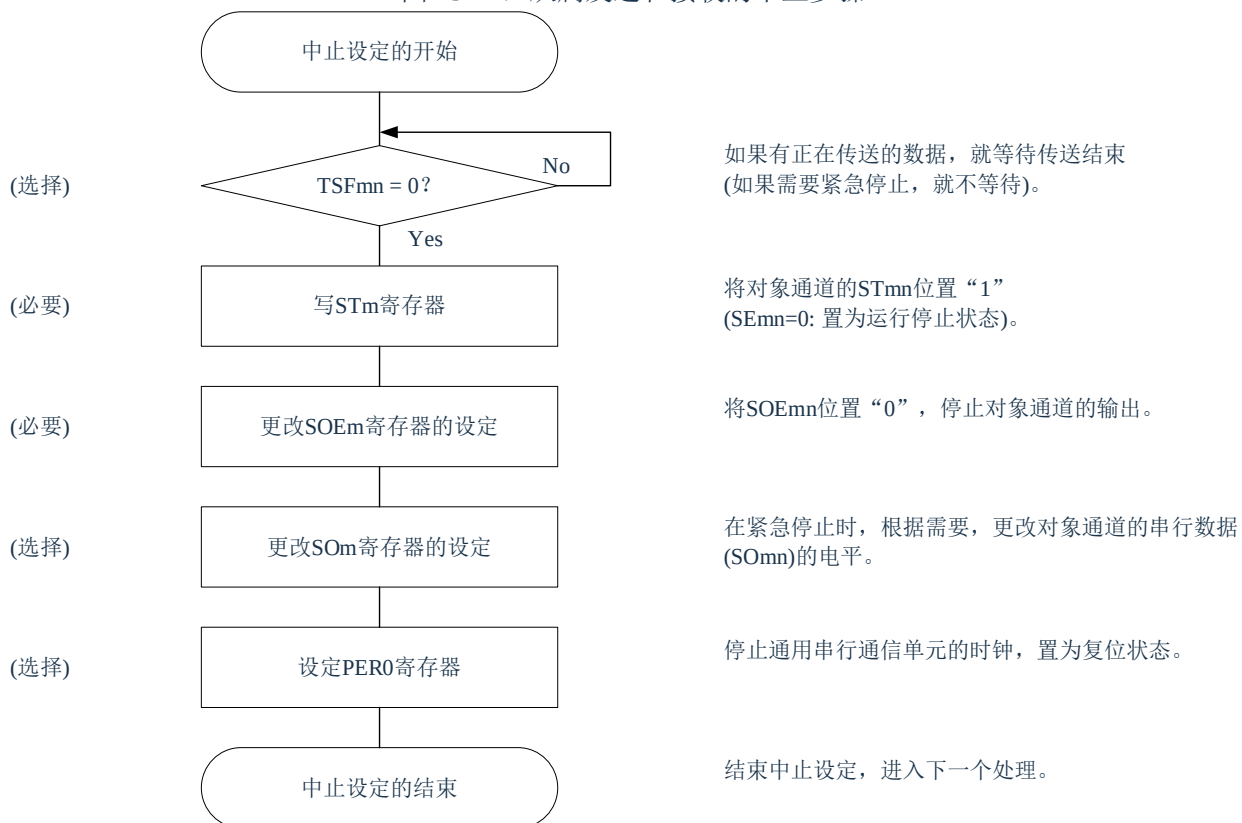
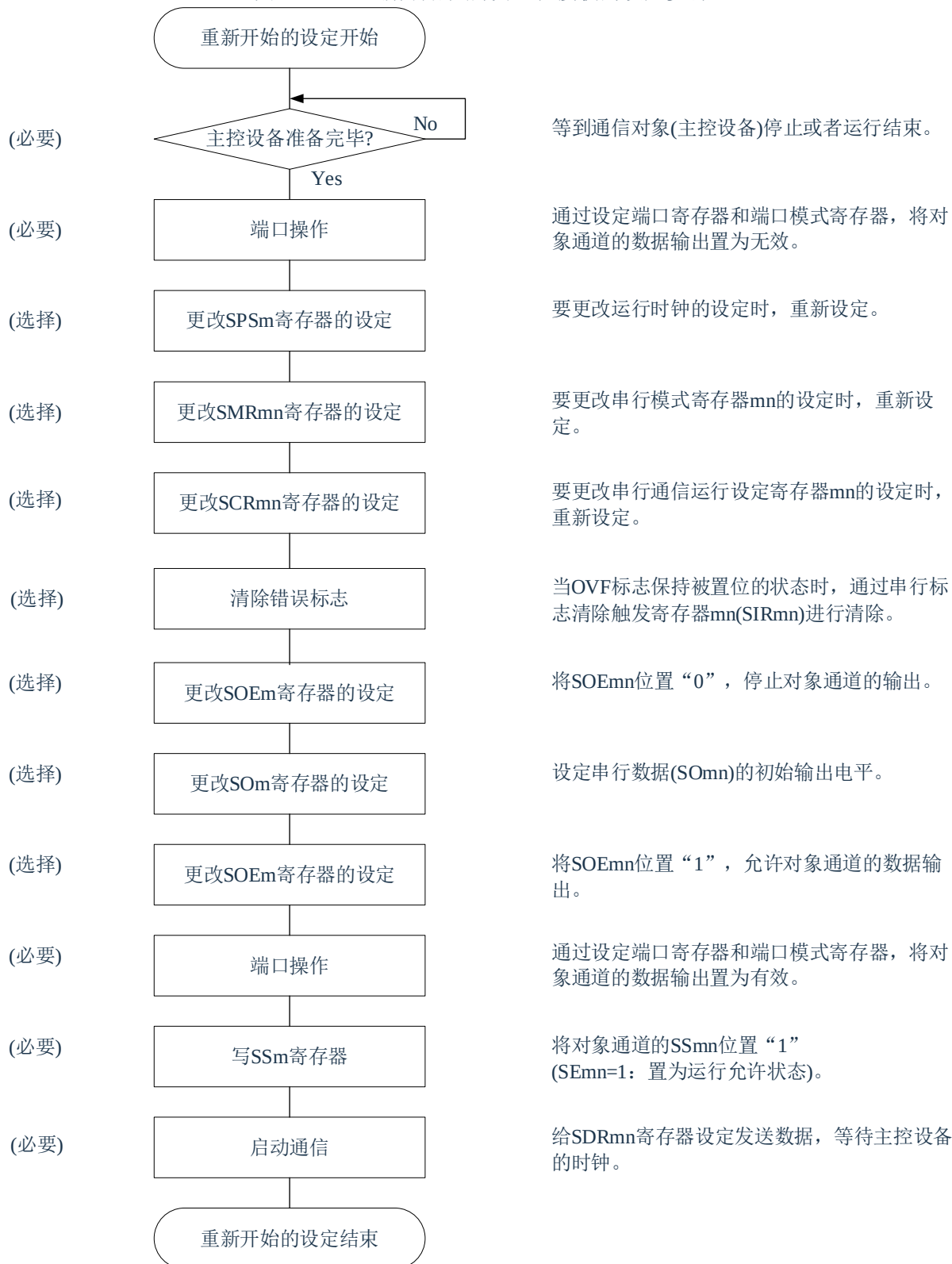


图 13-45：重新开始从属发送和接收的设定步骤



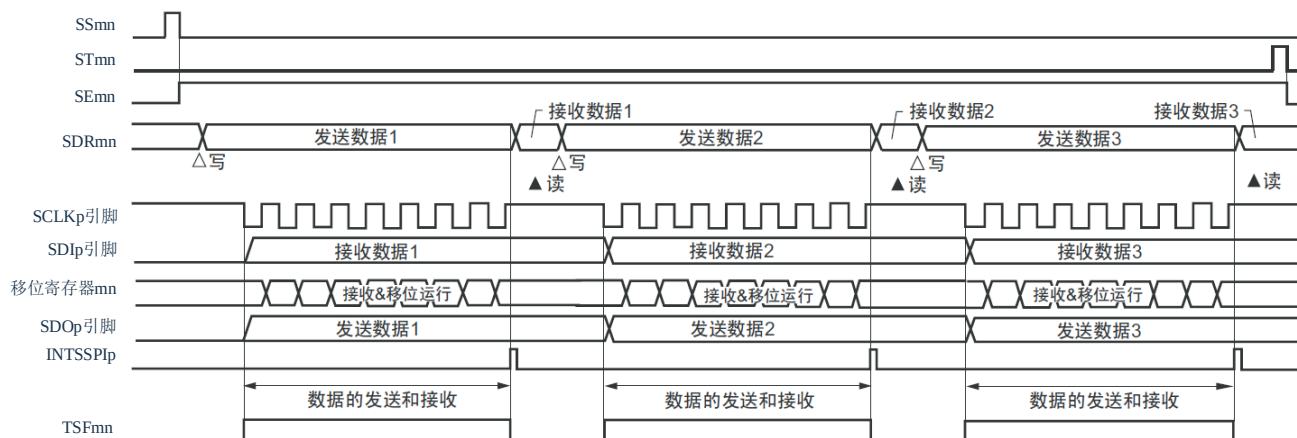
注意：

在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

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

(3) 处理流程（单次发送和接收模式）

图 13-46：从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

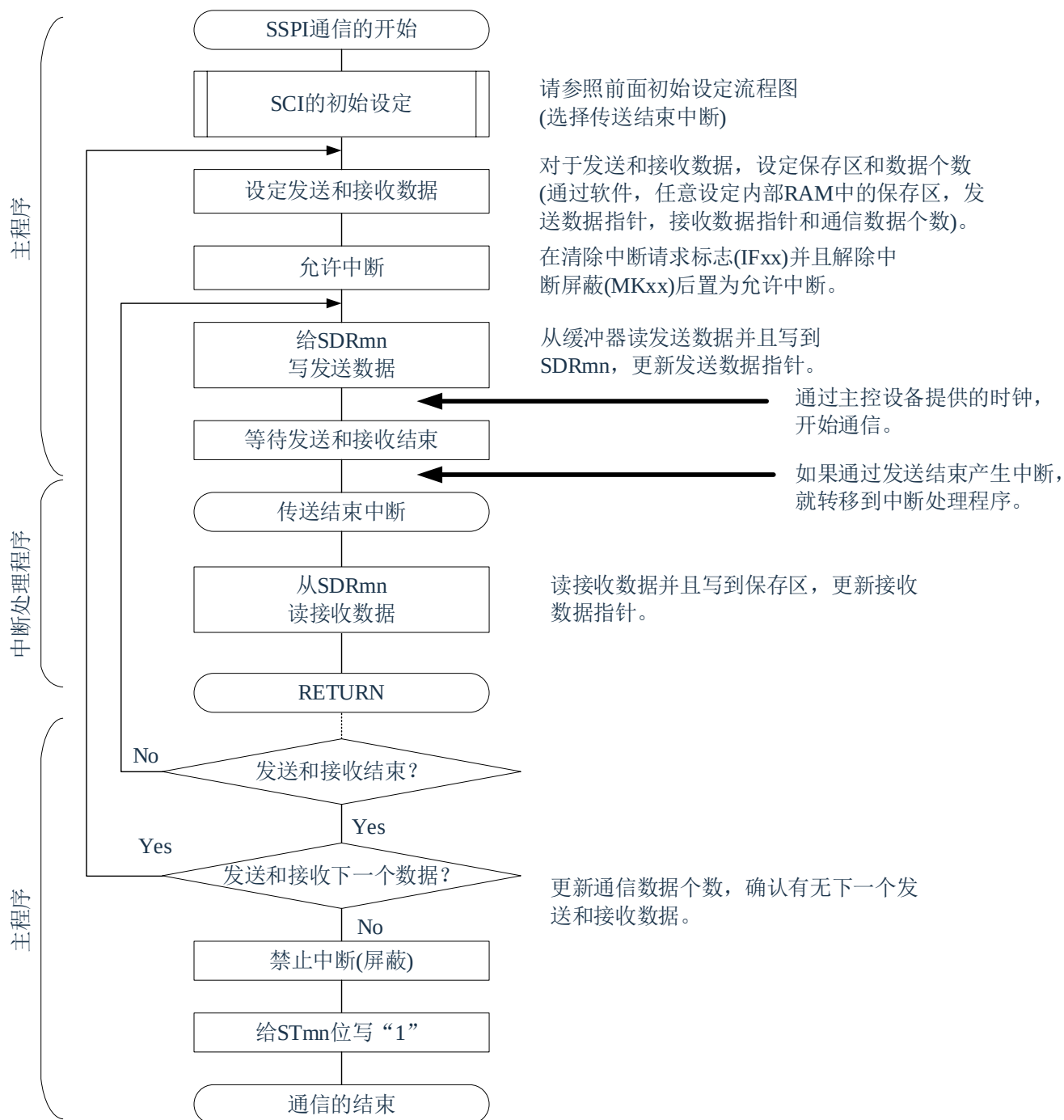


备注：m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

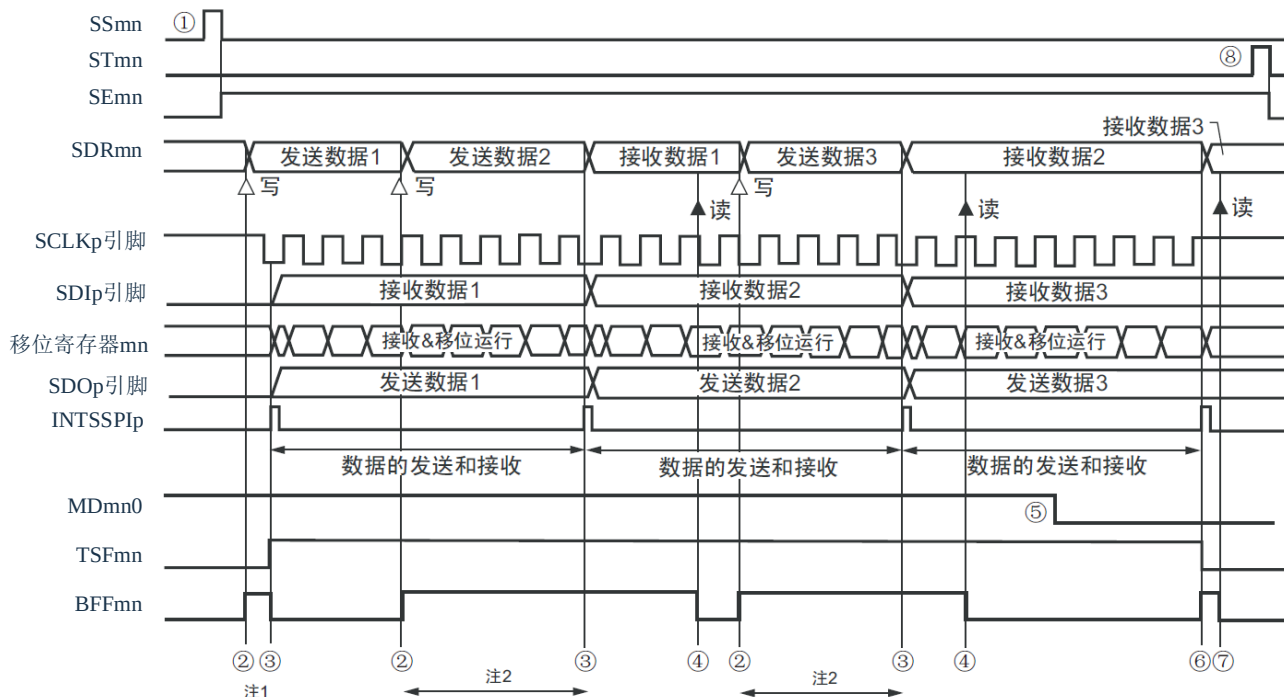
图 13-47：从属发送和接收（单次发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

(4) 处理流程（连续发送和接收模式）

图 13-48：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

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

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

备注：

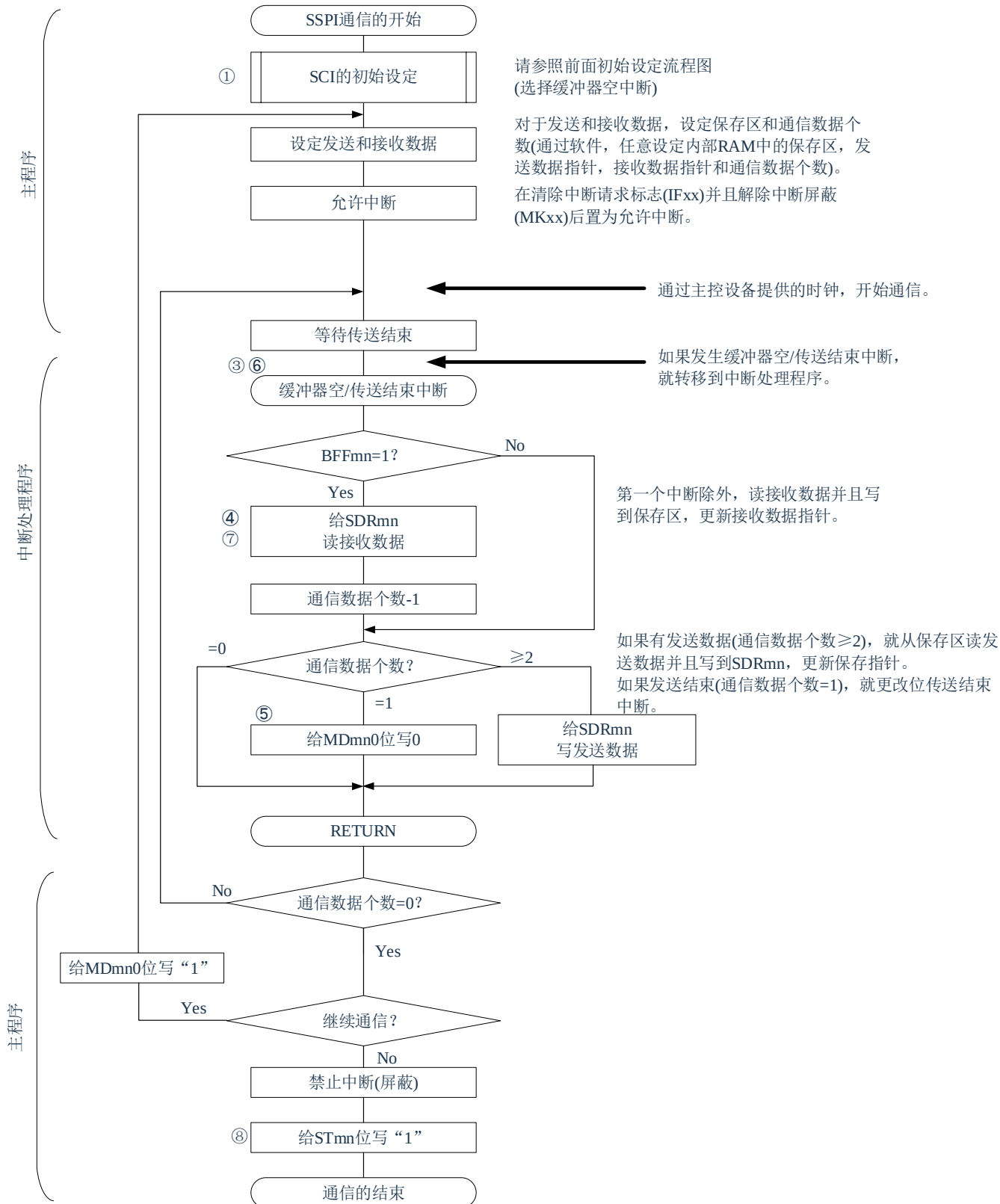
图中的①~⑧对应“图 13-49 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧；

m：单元号（m=0、1）；

n：通道号（n=0、1、2、3）；

p：SSPI 号（p=00、01、10、11、20、21）。

图 13-49：从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：图中的①~⑧对应“图 13-48 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

13.7.7 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：

- 主控设备

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

- 从属设备

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

备注：

- 1) 容许的最大传送时钟频率为 6MHz；
- 2) 因为 SDRmn[15:9]的值为串行数据寄存器 mn（SDRmn）的 bit15~9 的值（0000000B~1111111B），所以为 0~127；
- 3) 运行时钟（F_{MCK}）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn）。

表 13-28: 线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (F _{MCK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =32MHz 运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	500KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	250KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	125KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	62.5KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	31.25KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	15.63KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	7.81KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	3.91KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.95KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	32MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	500KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	250KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	125KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	62.5KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	31.25KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	15.63KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	7.81KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	3.91KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.95KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	977Hz

注: 要更改被选择为 F_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 时, 必须在停止通用串行通信单元 (SCI) 的运行 (串行通道停止寄存器 m (STm) = 000FH) 后进行更改;

X: 忽略;

m: 单元号 (m=0、1);

n: 通道号 (n=0、1、2、3)。

13.7.8 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表 13-29 所示：

表 13-29：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。→	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn（SDIRmn）写“1”。→	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0、1）；
n：通道号（n=0、1、2、3）。

13.8 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、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定

[时钟控制]

- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率主从属通信：Max.6Mhz

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

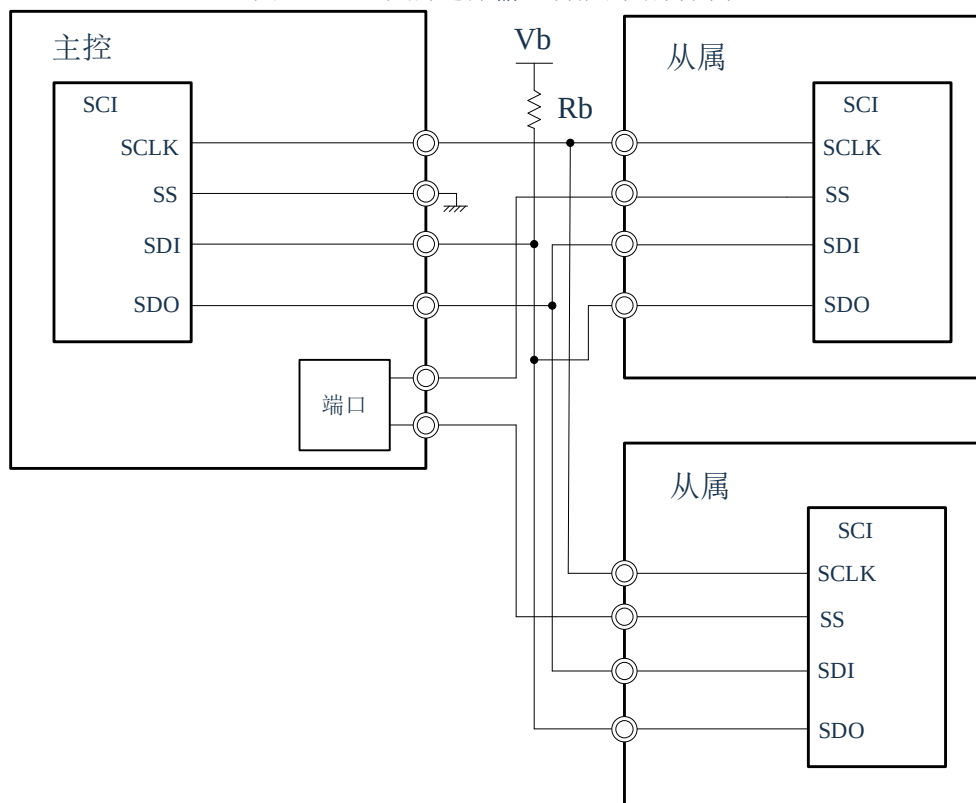
从属选择输入功能有以下 3 种通信运行：

- 从属发送（参照 13.8.1）
- 从属接收（参照 13.8.2）
- 从属的发送和接收（参照 13.8.3）

注意：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。

能通过使用从属选择输入功能，使 1 个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1 个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SDO 引脚的输出。当被选择为通信对象的从属设备时，SDO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO 引脚变为高电平输出，因此在连接多个从属设备的环境下需要将 SDO 引脚设定为 Nch-O.D 并且将该节点上拉。另外，即使输入主控设备的串行时钟也不进行发送和接收。

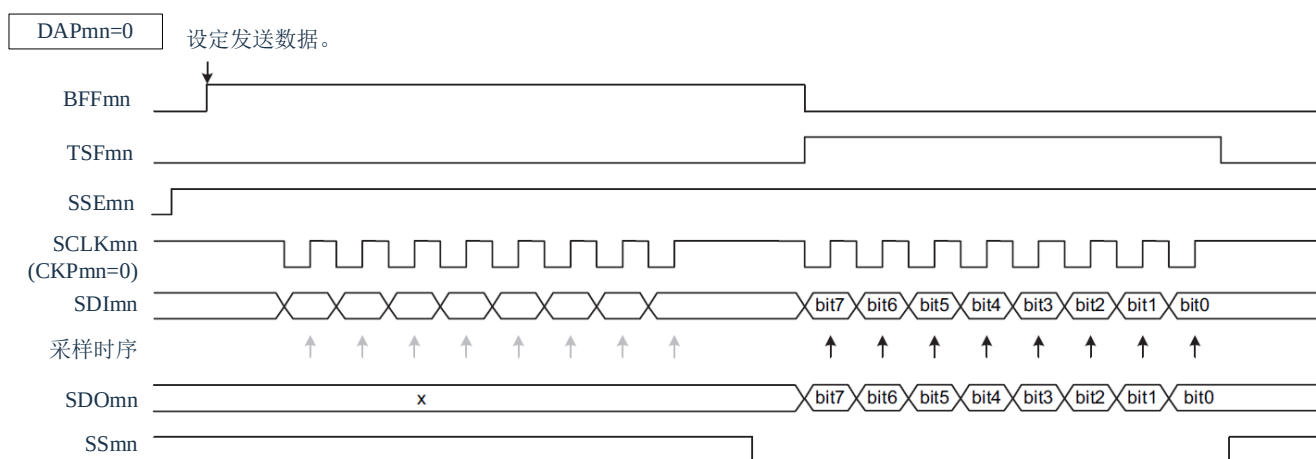
图 13-50：从属选择输入功能的结构例子



注意：

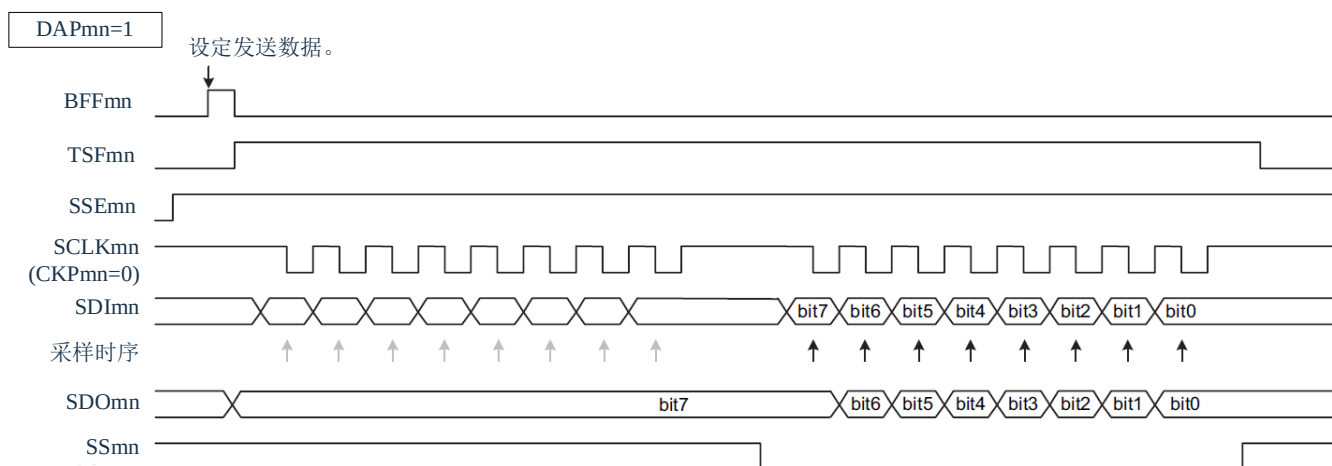
将 SDO00 引脚选择为 N 沟道漏极开路输出模式；
必须通过端口的操作输出从属选择信号。

图 13-51：从属选择输入功能的时序图



在 SSmn 为高电平期间，即使在 SCKmn(串行时钟)的下降沿也不进行发送，而且也不进行与上升沿同步的接收数据的采样。

在 SSmn 为低电平期间，与串行时钟的下降沿同步输出数据(移位)并且与上升沿同步接收数据。



当 DAPmn 位为“1”时，如果在 SSmn 为高电平期间设定发送数据，就将最初的数据(bit7)提供给数据输出。但是，即使在 SCLKmn(串行时钟)的上升沿也不移位，而且也不进行与下降沿同步的接受数据的采样。如果 SSmn 变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注：m：单元号（m=0）；

n：通道号（n=0）。

13.8.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下本产品将数据发送到其他设备的运行。

表 13-30：从属发送

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、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、8位
传送速率	Max.6M[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 6M[Hz]。

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

F_{MCK}：对象通道的运行时钟频率

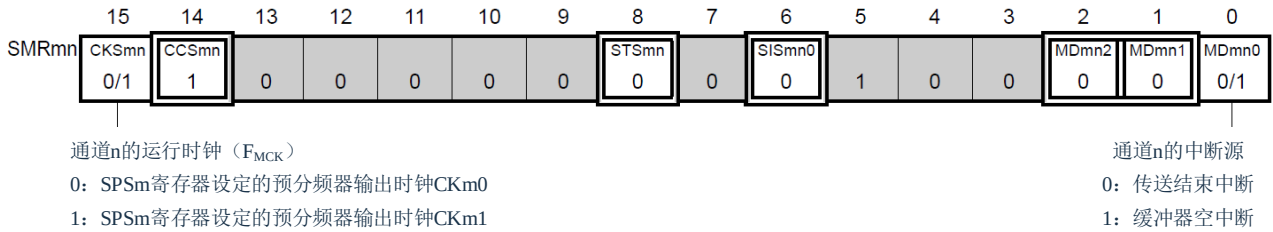
m：单元号（m=0）；

n：通道号（n=0）。

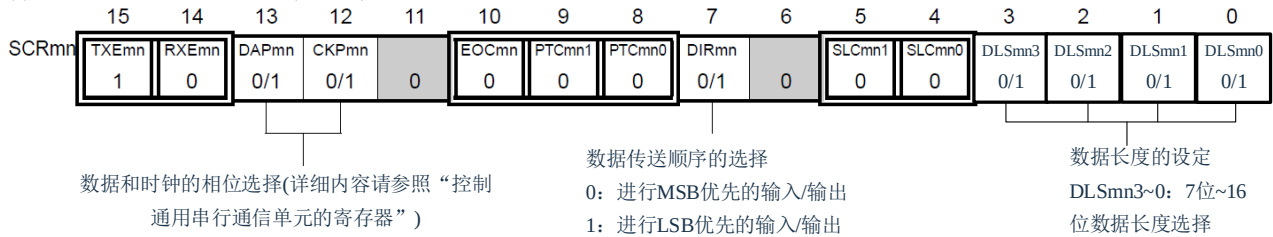
(1) 寄存器的设定

图 13-52：从属选择输入功能（SSPI00）从属发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

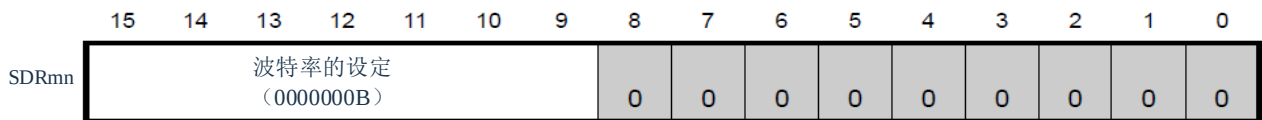


(b) 串行通信运行设定寄存器mn(SCRmn)

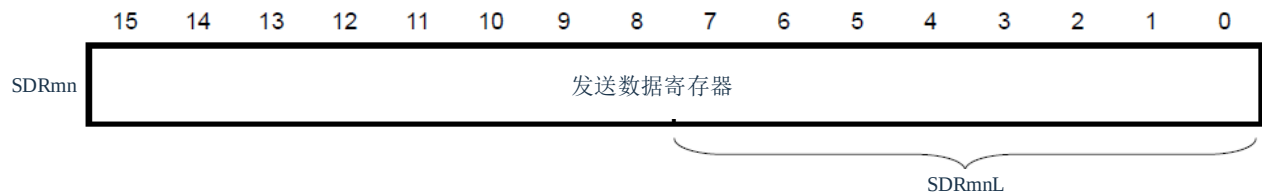


(c) 串行数据寄存器mn(SDRmn)

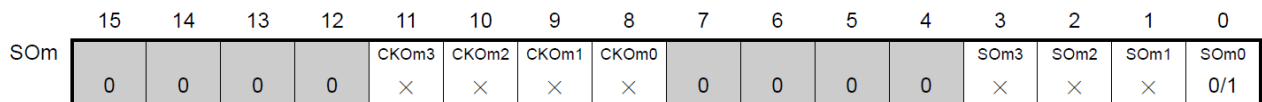
(1) 运行停止（SEmn=0）时



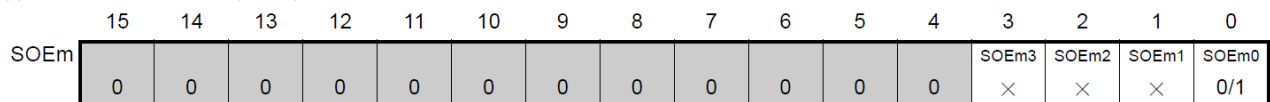
(2) 运行期间（SEmn=1）（低8位：SDRmnL）



(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI号（p=00）；

□：在SSPI主控接收模式中为固定设定； ■ 不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

图 13-52：从属选择输入功能（SSPI00）从属发送时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

□：在 SSPI 主控接收模式中为固定设定；

■：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

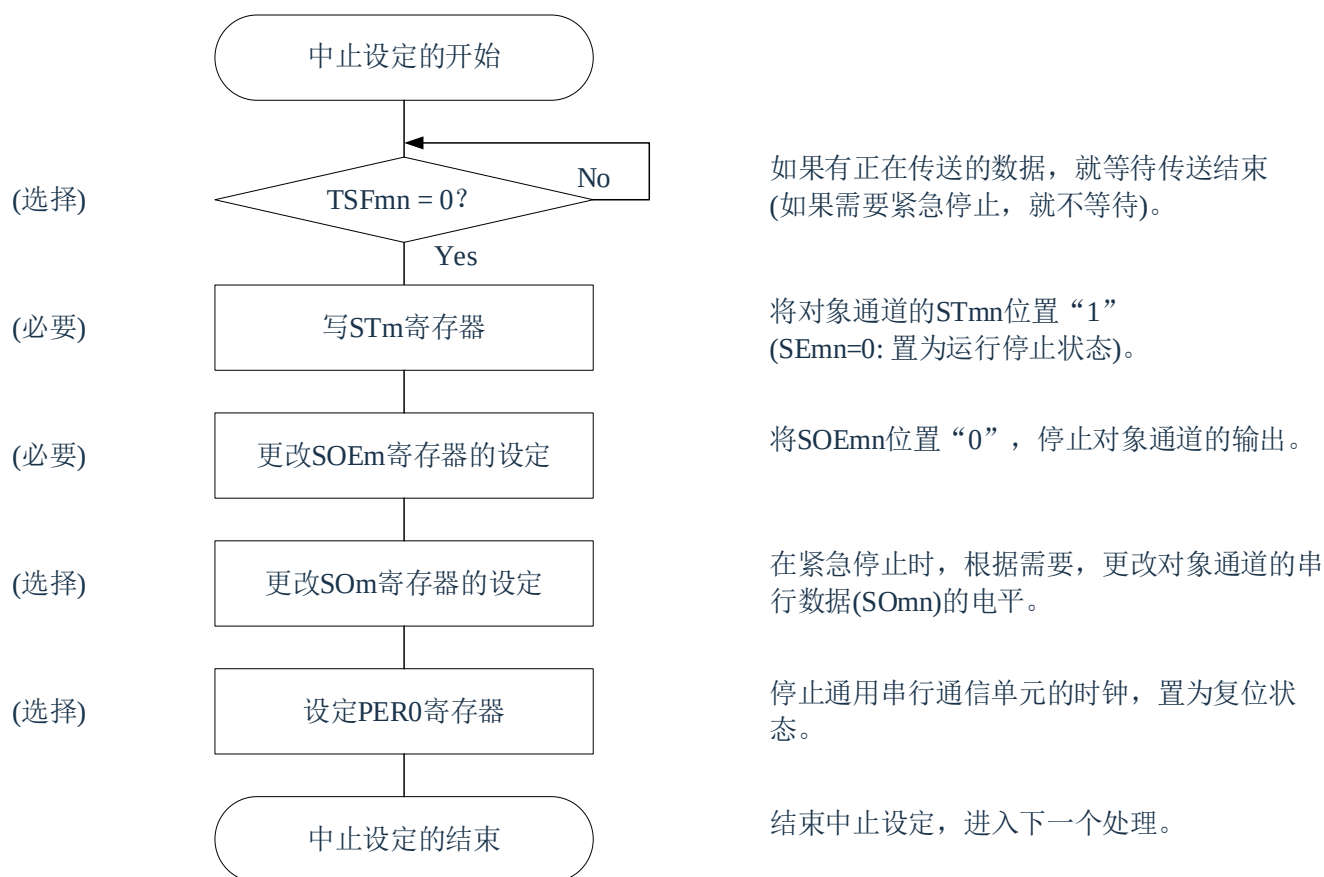
(2) 操作步骤

图 13-53: 从属发送的初始设定步骤



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

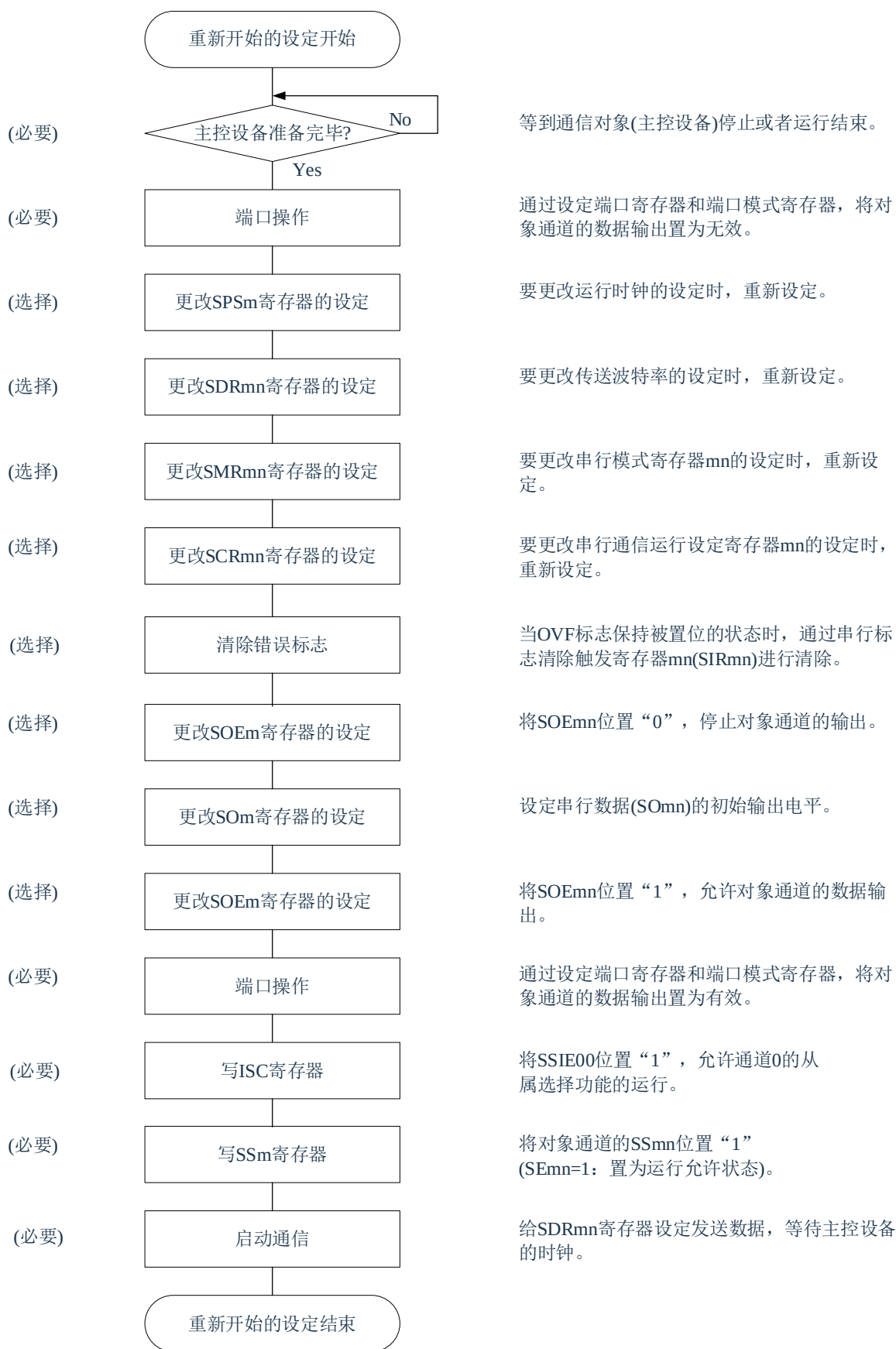
图 13-54: 从属发送的中止步骤



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

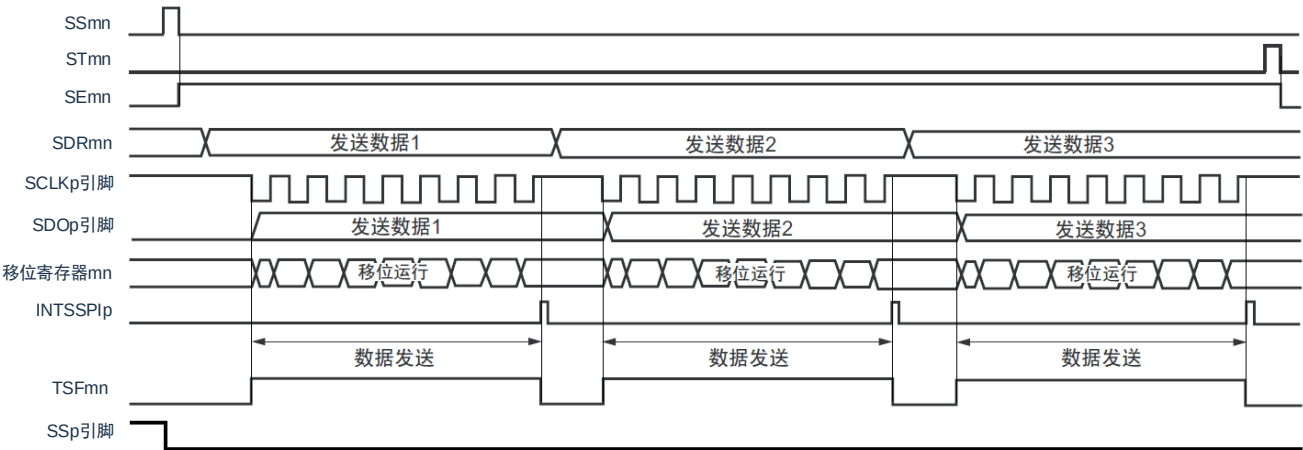
备注：m: 单元号 (m=0)；n: 通道号 (n=0)；p: SSPI 号 (p=00)；

图 13-55: 重新开始从属发送的设定步骤



(3) 处理流程（单次发送模式）

图 13-56：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

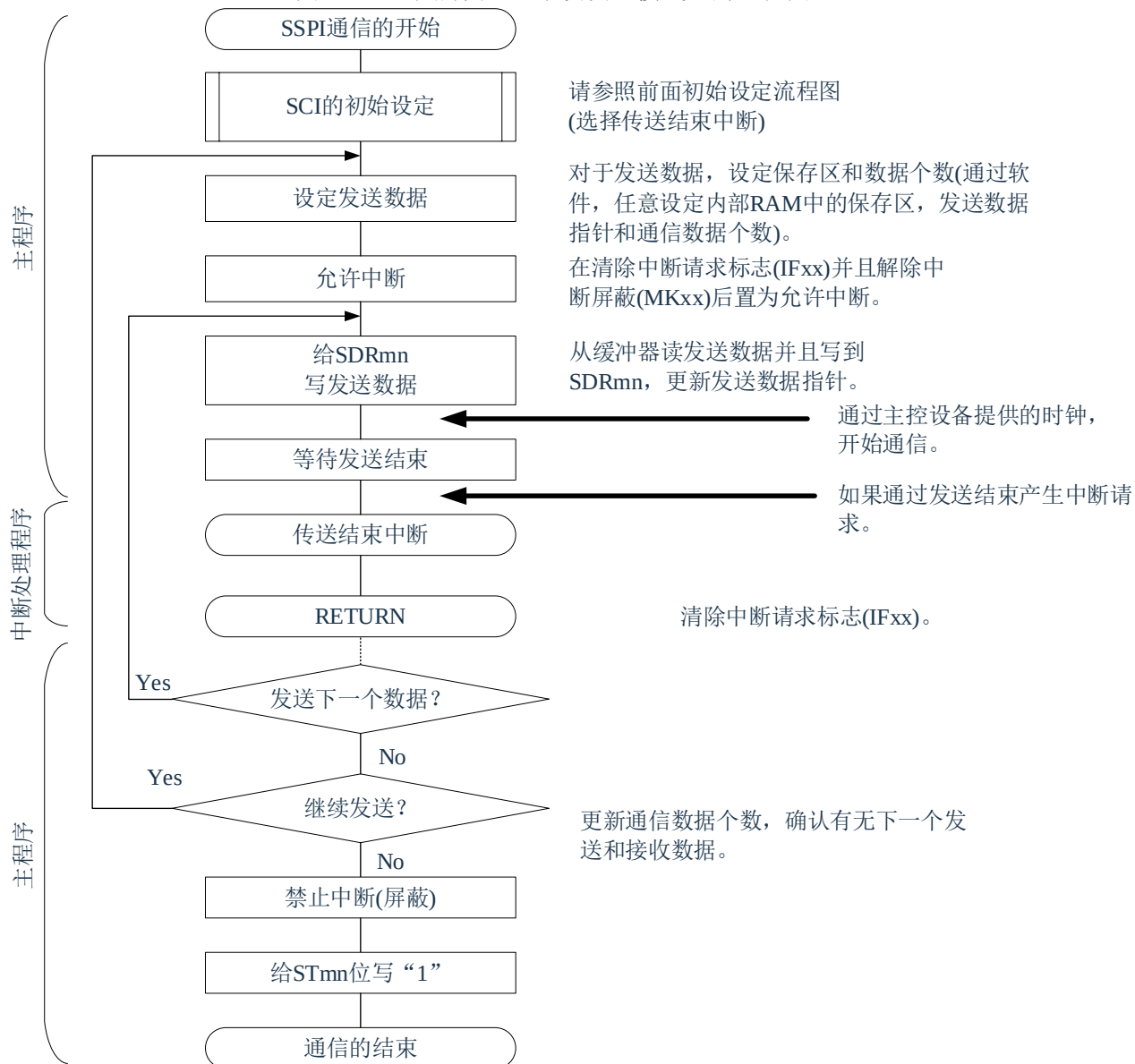


备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图 13-57：从属发送（单次发送模式）的流程图



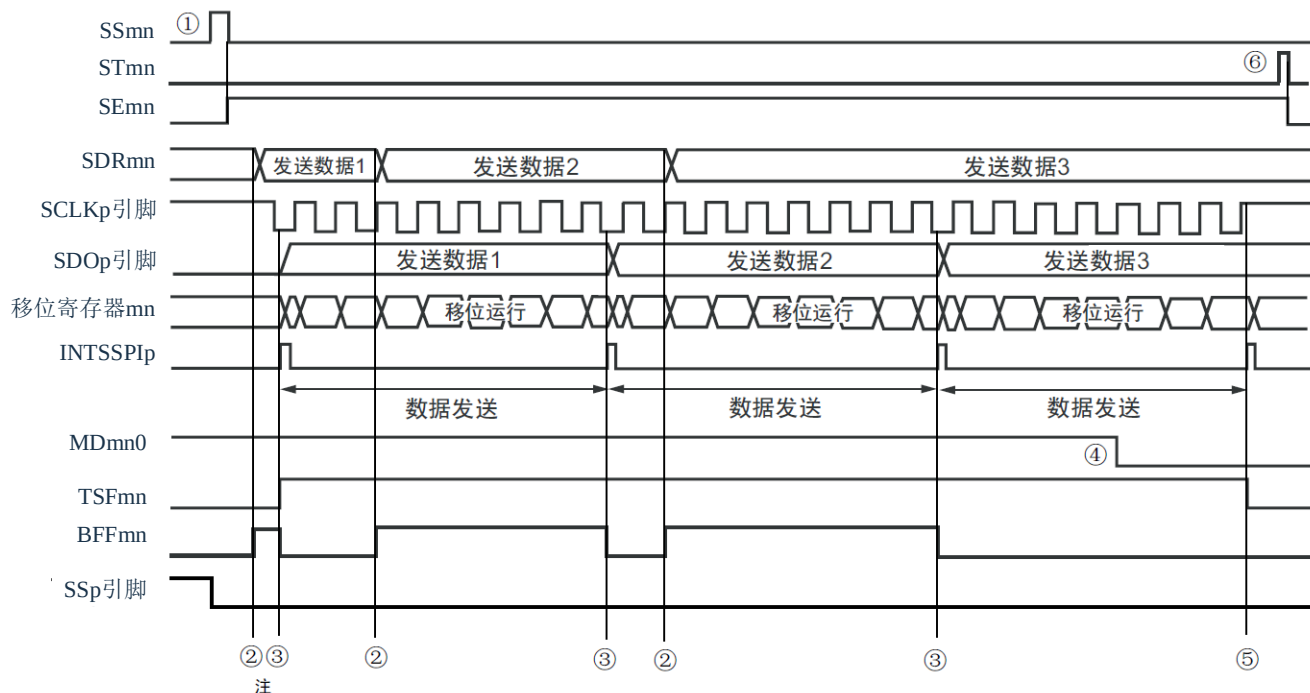
备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI号（p=00）。

(4) 处理流程（连续发送模式）

图 13-58：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。

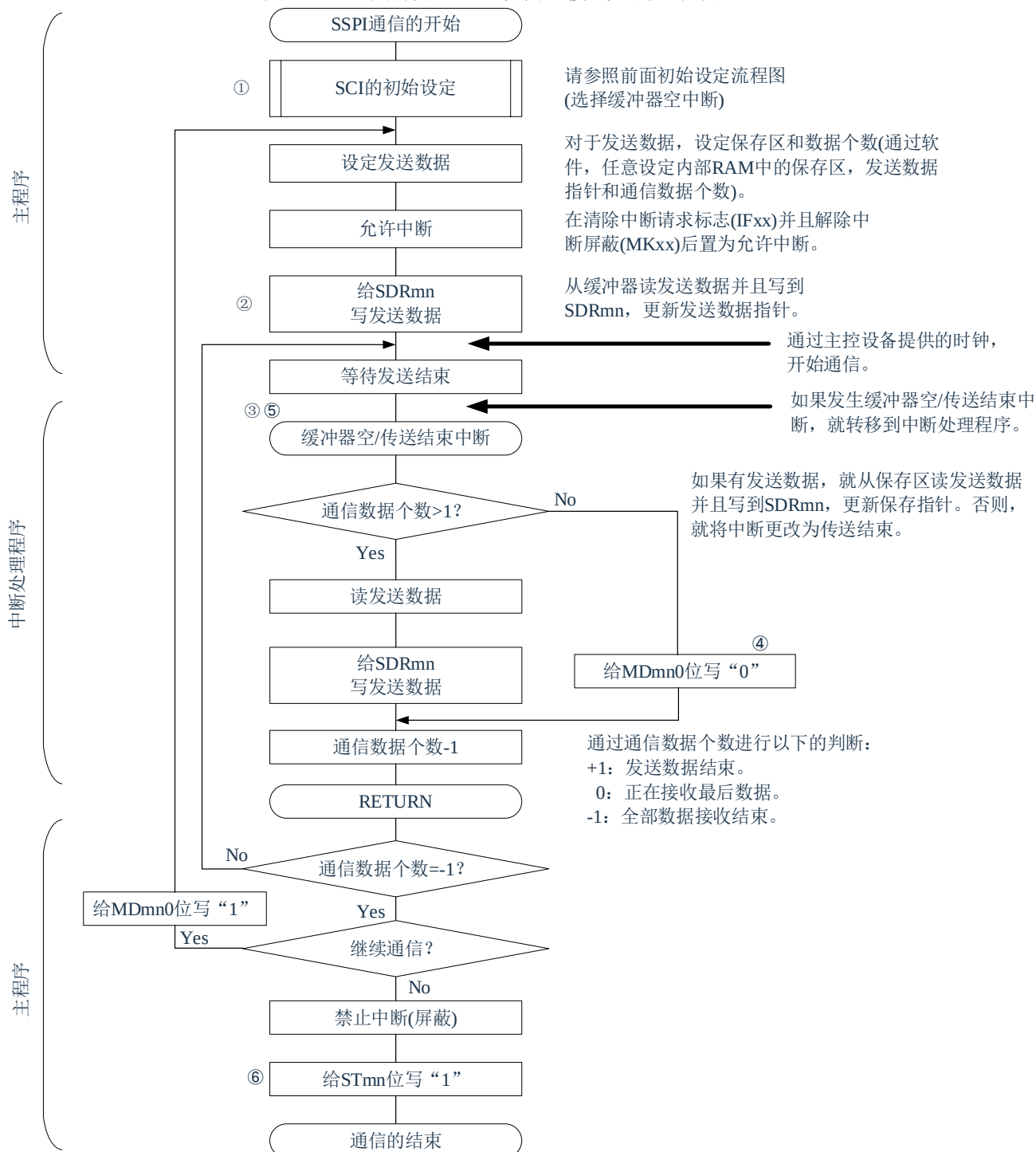
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图 13-59：从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 13-58 从属发送（连续发送模式）的时序图”中的①~⑥。

m: 单元号 (m=0) ;

n: 通道号 (n=0) ;

p: SSPI 号 (p=00) 。

13.8.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

表 13-31：从属接收

从属选择输入功能	SSPI00
对象通道	SCIO的通道0
使用的引脚	SCLKOI00、SDI00、SS00
中断	INTSSPI00
	只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[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 6M[Hz]。

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

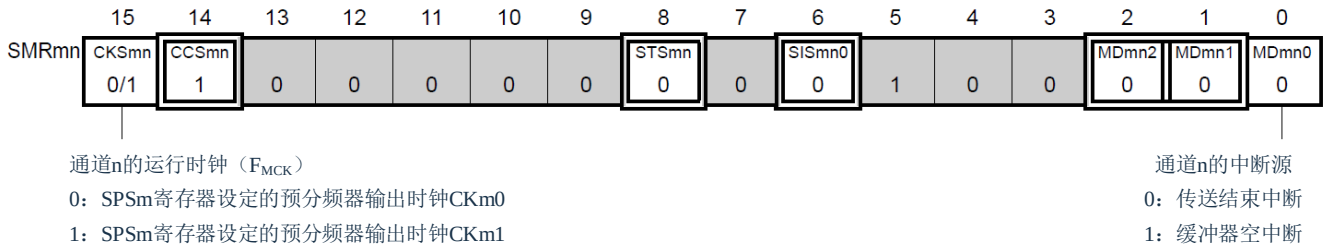
F_{MCK}：对象通道的运行时钟频率

m：单元号（m=0） n：通道号（n=0）。

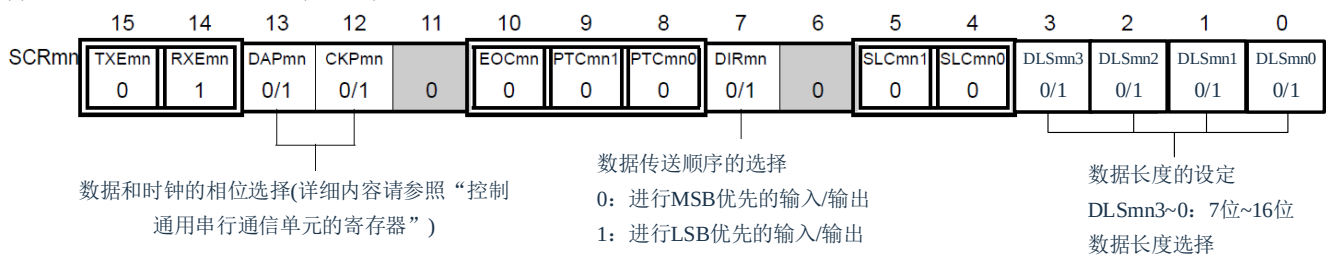
(1) 寄存器的设定

图 13-60：从属选择输入功能（SSPI00）从属接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

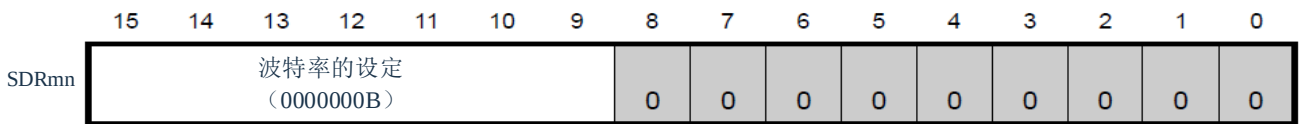


(b) 串行通信运行设定寄存器mn(SCRmn)

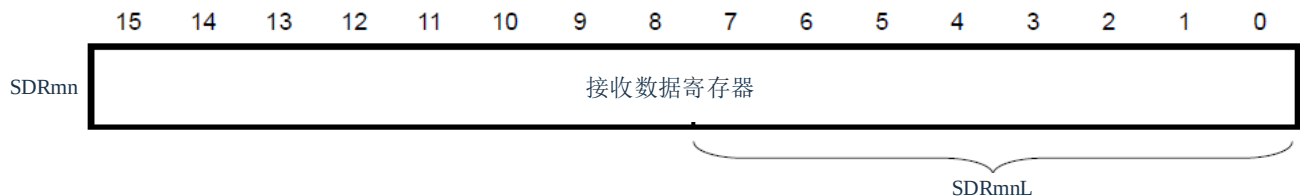


(c) 串行数据寄存器mn(SDRmn)

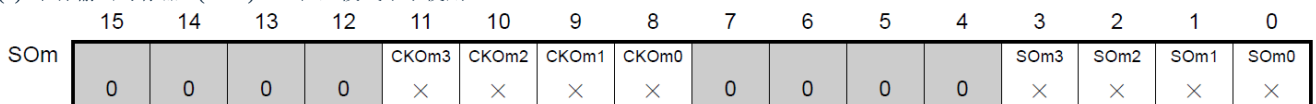
(1) 运行停止（SEmn=0）时



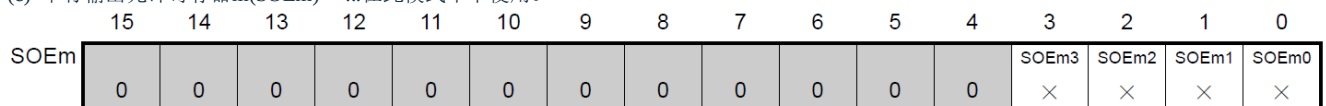
(2) 运行期间（SEmn=1）（低8位：SDRmnL）



(d) 串行输出寄存器m(SOm) ……在此模式中不使用。



(e) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。



备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

□：在 SSPI 主控接收模式中为固定设定；

■：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

图 13-60：从属选择输入功能（SSPI00）从属接收时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效

1: SS00引脚的输入值有效

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

☐：在 SSPI 主控接收模式中为固定设定；

☐：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-61：从属接收的初始设定步骤

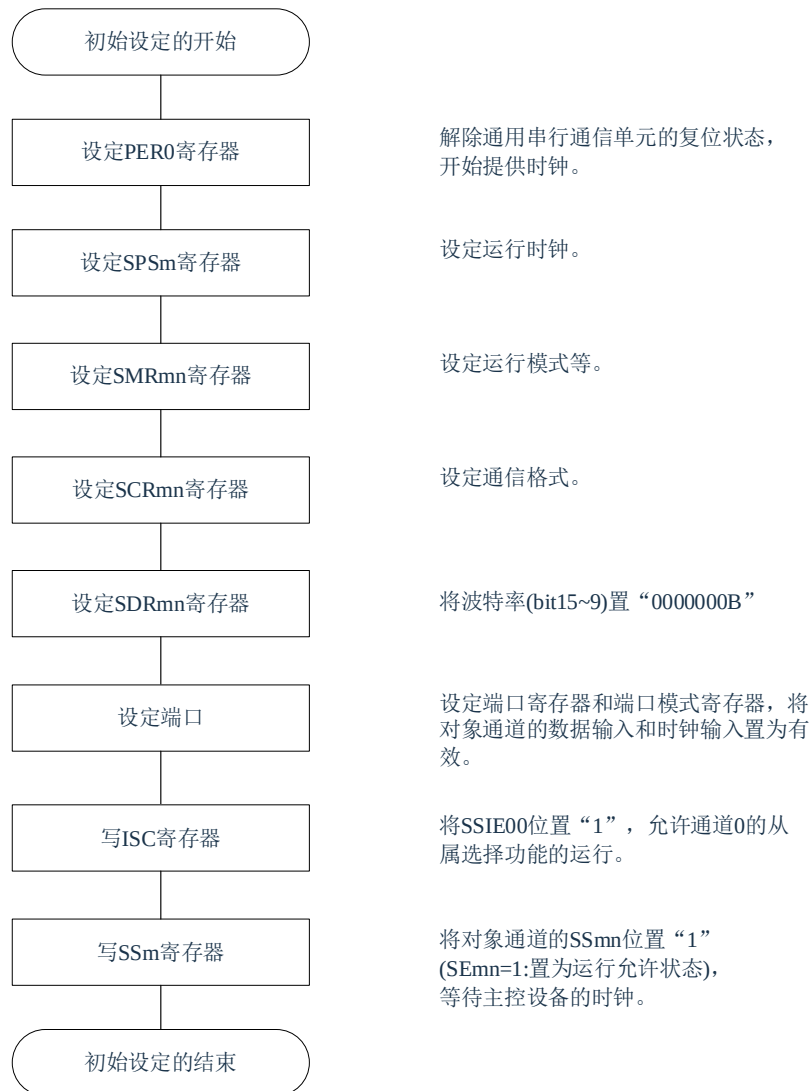
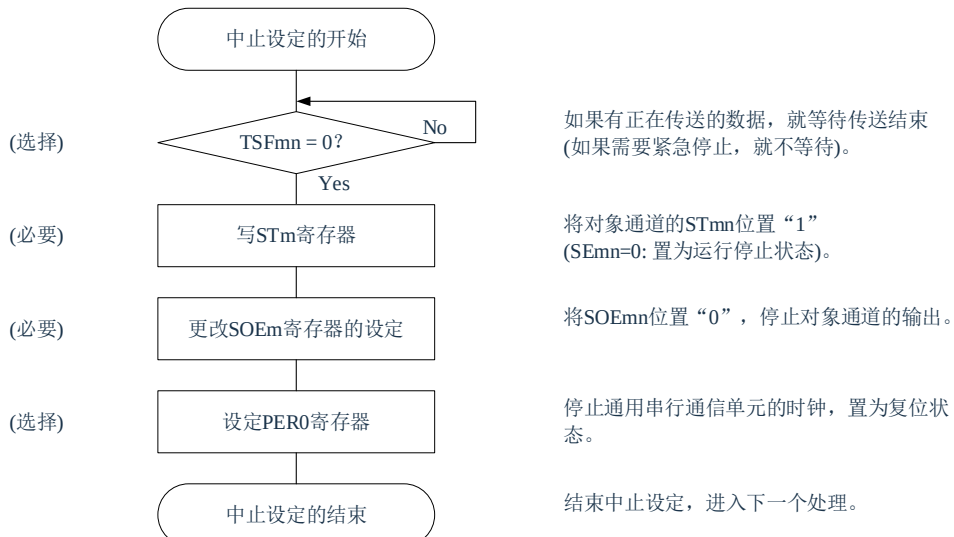
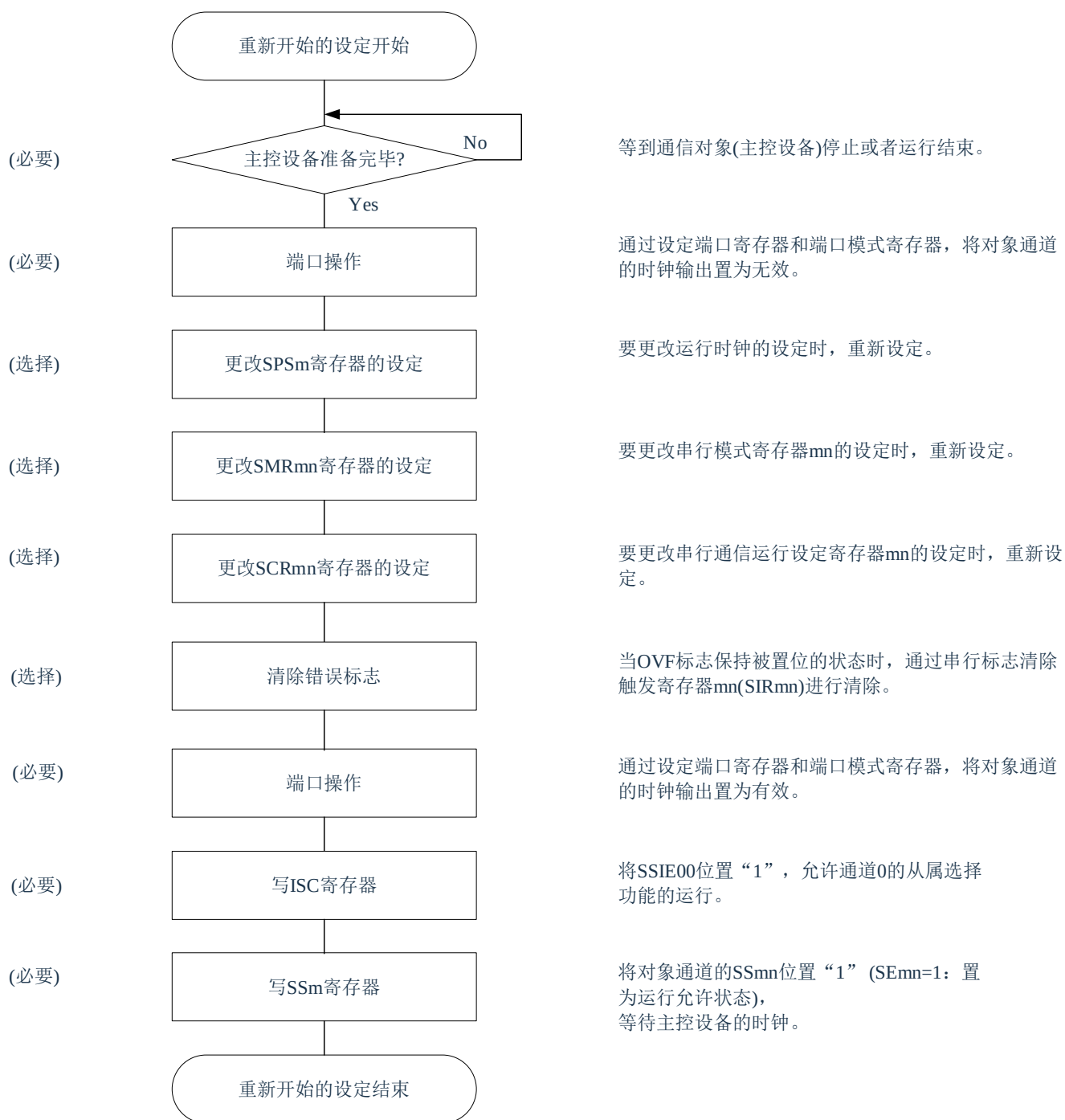


图 13-62：从属接收的中止步骤



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

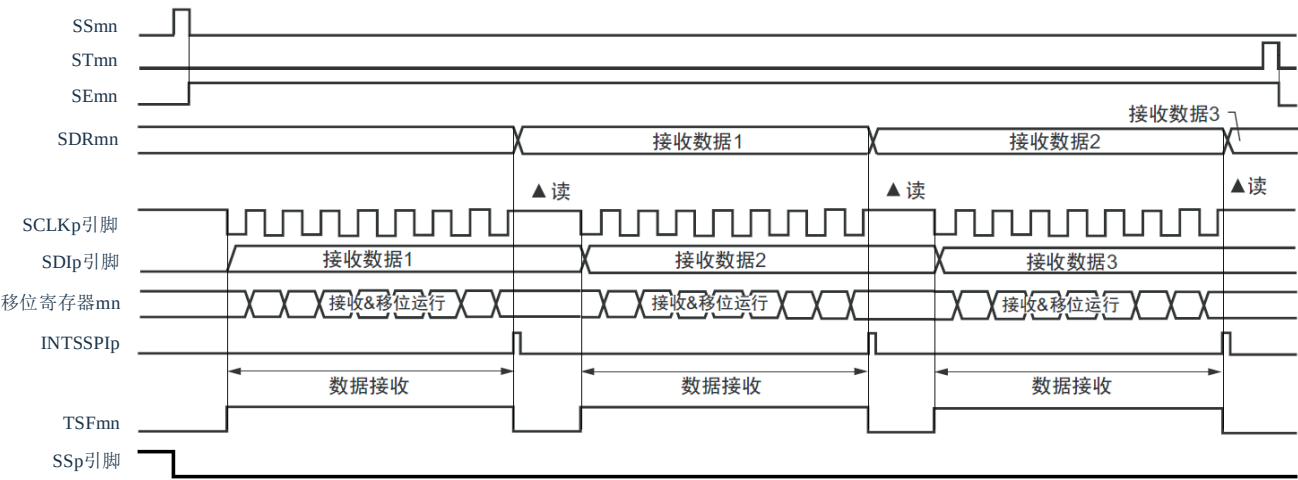
图 13-63: 重新开始从属接收的设定步骤



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

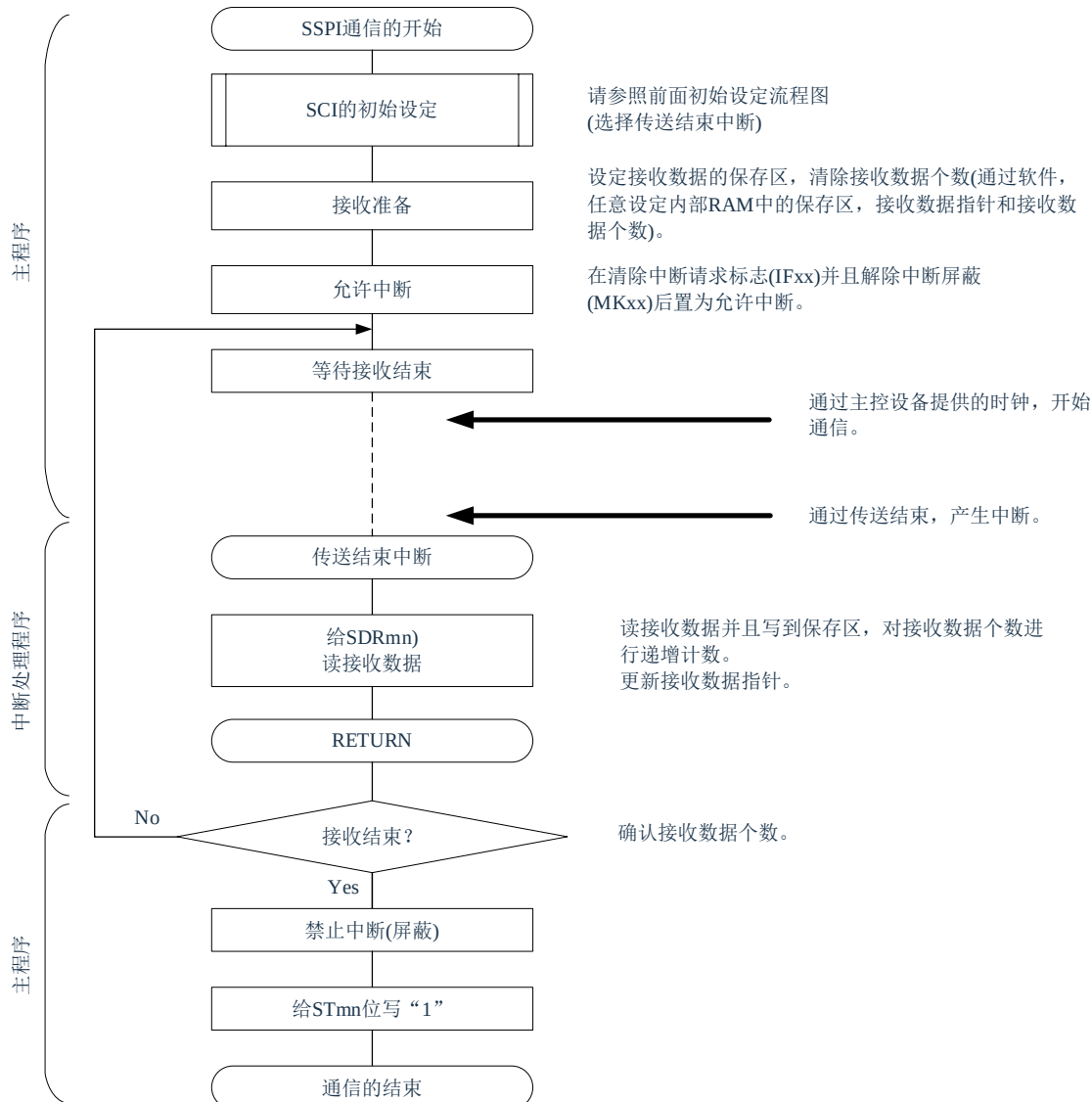
(3) 处理流程（单次接收模式）

图 13-64：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 13-65: 从属接收（单次接收模式）的流程图



13.8.3 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品和其他设备进行数据发送和接收的运行。

表 13-32：从属的发送和接收

从属选择输入功能	SSPI00
对象通道	SCI0的通道0
使用的引脚	SCLKOI00、SDI00、SDO00、SS00
中断	INTSSPI00
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位、8位
传送速率	Max.6M[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 6M[Hz]。

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

F_{MCK}：对象通道的运行时钟频率

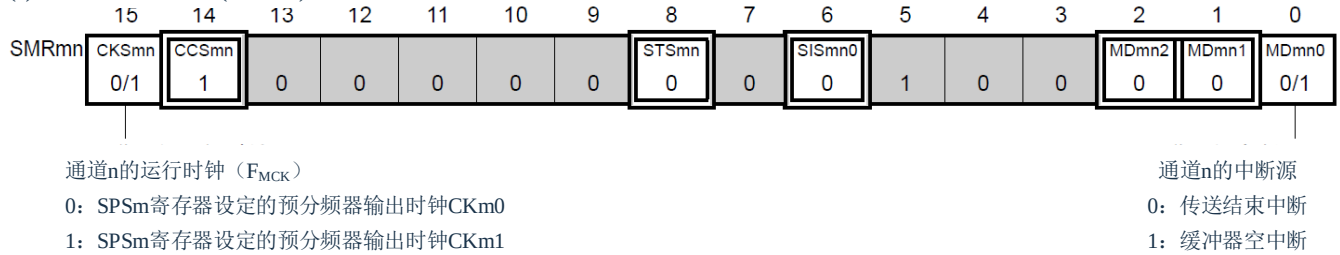
m：单元号（m=0）；

n：通道号（n=0）。

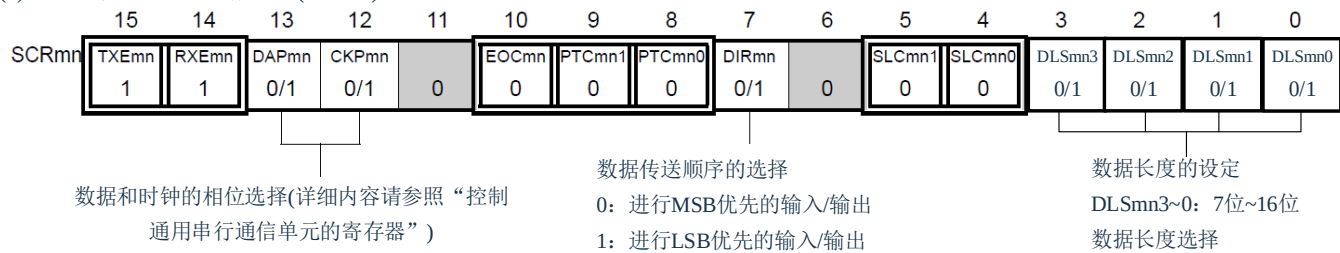
(1) 寄存器的设定

图 13-66：从属选择输入功能（SSPI00）从属发送和接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

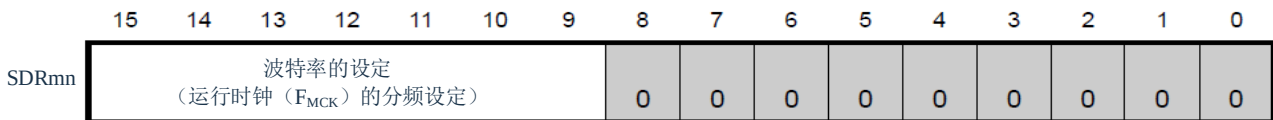


(b) 串行通信运行设定寄存器mn(SCRmn)

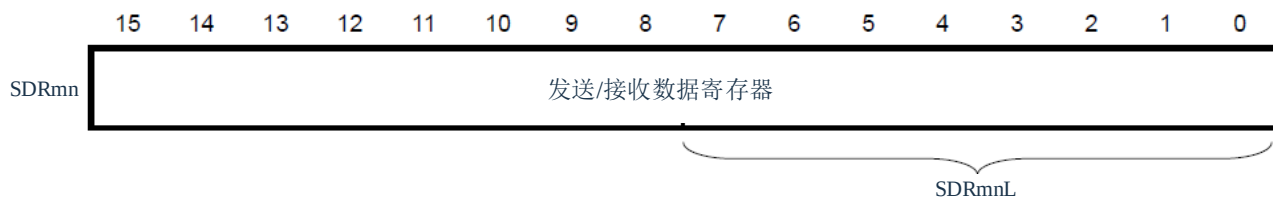


(c) 串行数据寄存器mn(SDRmn)

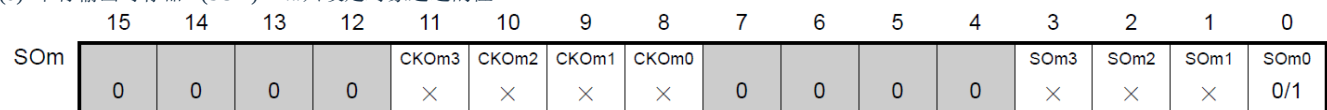
(1) 运行停止（SEmn=0）时



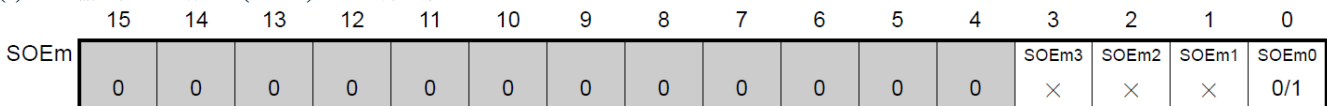
(2) 运行期间（SEmn=1）（低8位：SDRmnL）



(d) 串行输出寄存器m(SOm) ……只设定对象通道的位。



(e) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。



注意：在主机设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

□：在 SSPI 主控接收模式中为固定设定；

■：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

图 13-66：从属选择输入功能（SSPI00）从属发送和接收时的寄存器设定内容例子(2/2)

(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

(g) 输入切换控制寄存器(ISC) ……这是SSPI00从属通道(单元0的通道0)的SS00引脚的控制。

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

0: SS00引脚的输入值无效
1: SS00引脚的输入值有效

注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据；

备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）；

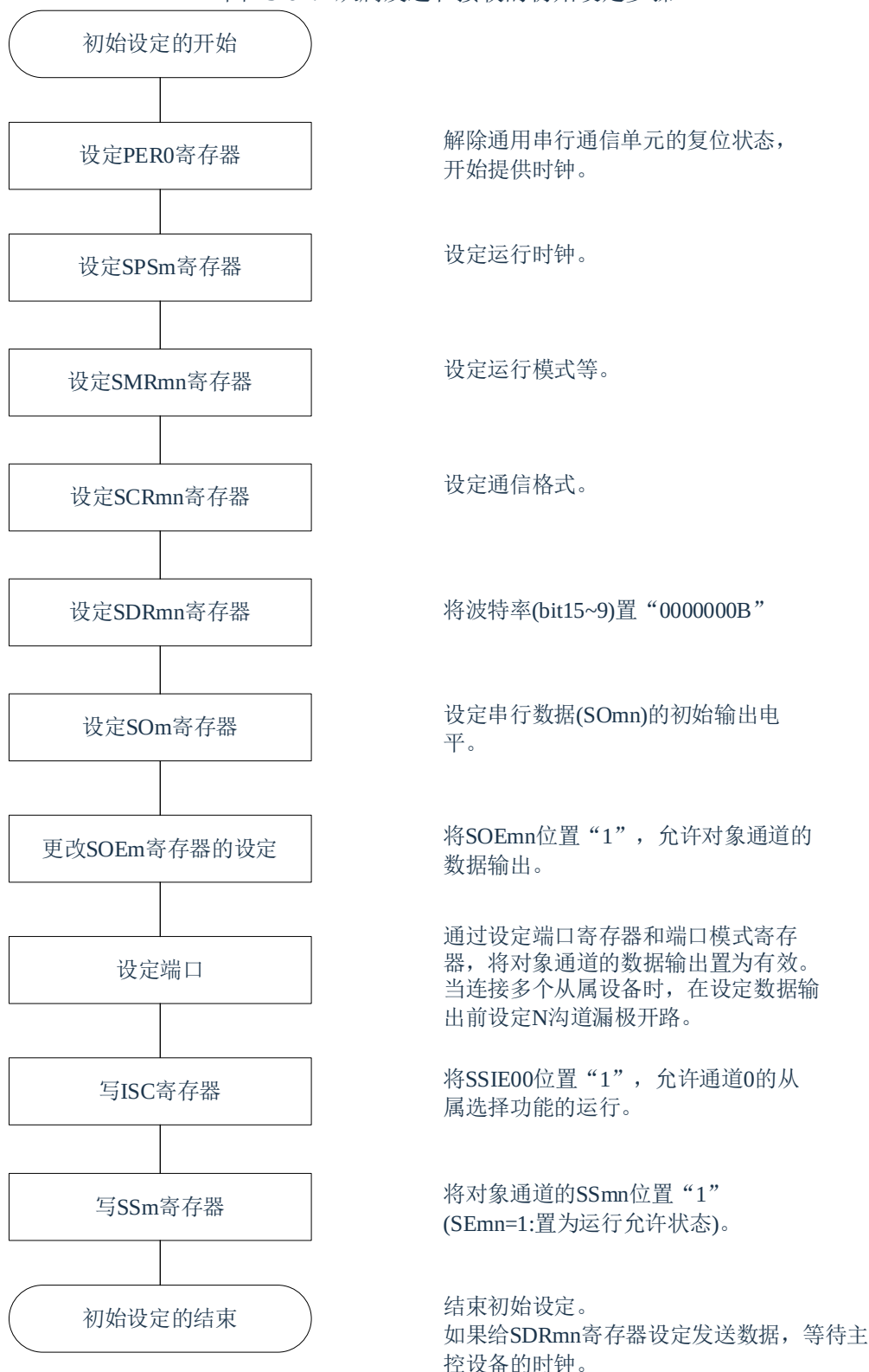
☐：在 SSPI 主控接收模式中为固定设定； ☐：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-67：从属发送和接收的初始设定步骤



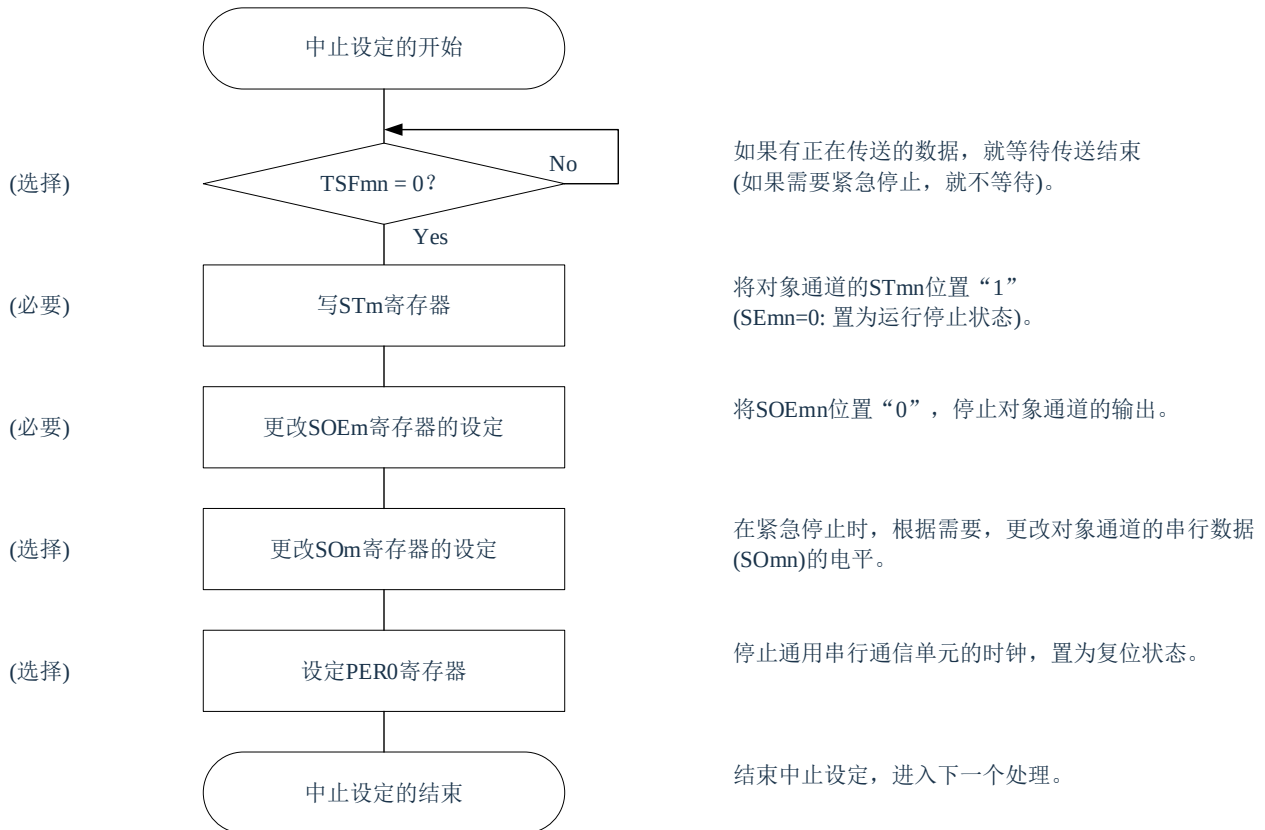
注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：m：单元号（m=0）；

n：通道号（n=0）；

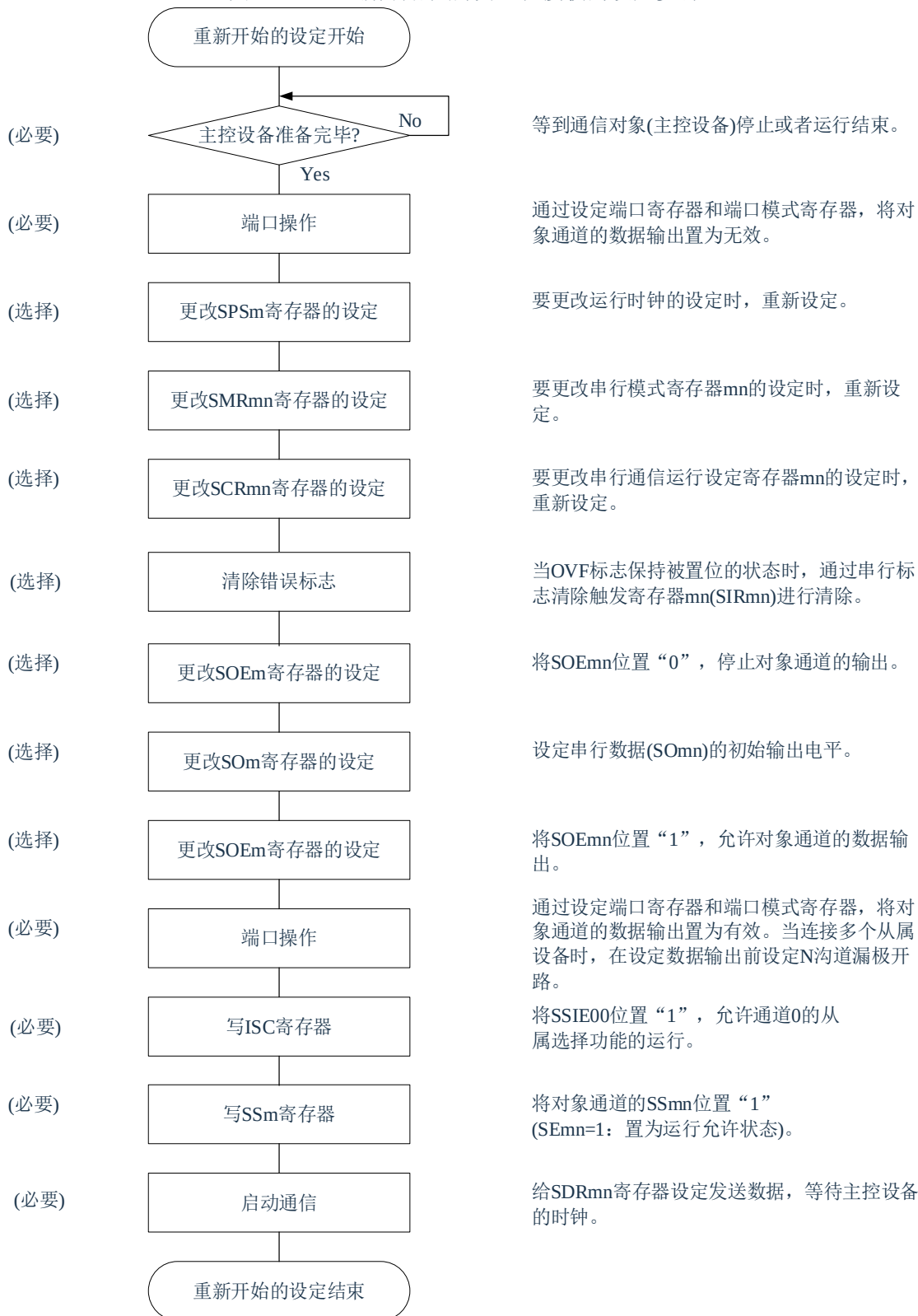
p：SSPI 号（p=00）。

图 13-68: 从属发送和接收的中止步骤



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

图 13-69：重新开始从属发送和接收的设定步骤



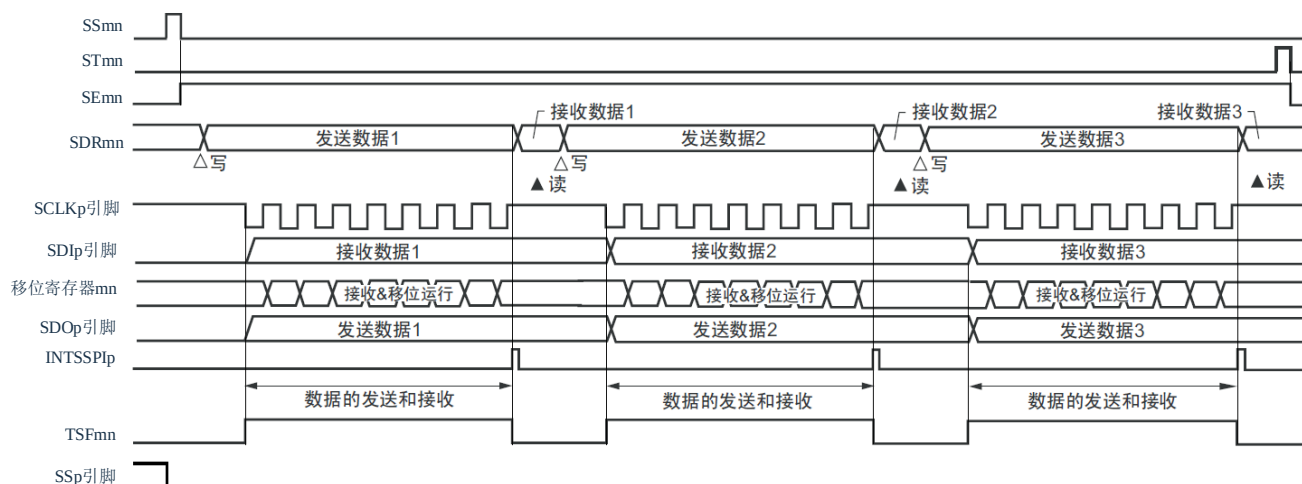
注意:

在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

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

(3) 处理流程（单次发送和接收模式）

图 13-70：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）

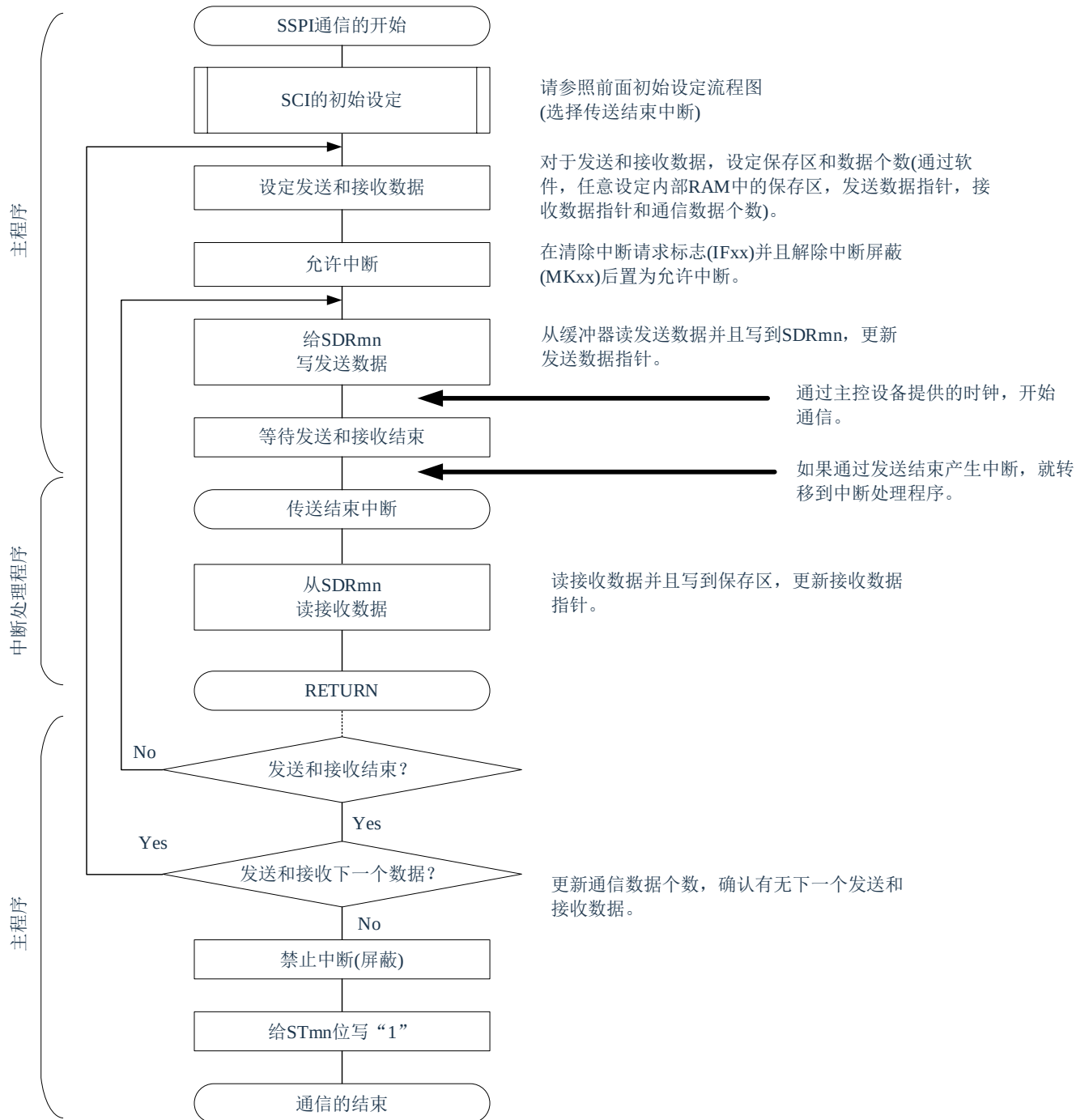


备注：m：单元号（m=0）；

n：通道号（n=0）；

p：SSPI 号（p=00）。

图 13-71：从属发送和接收（单次发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

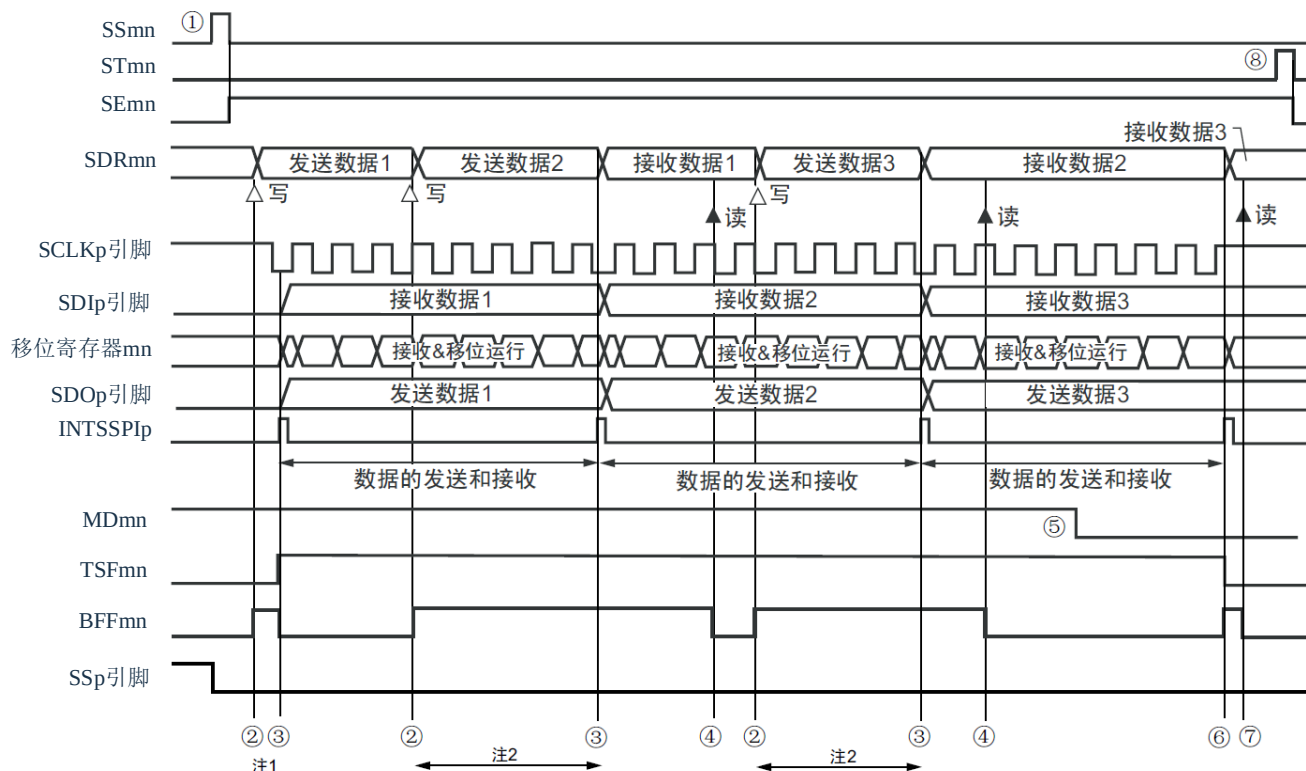
m: 单元号 (m=0) ;

n: 通道号 (n=0) ;

p: SSPI 号 (p=00) 。

(4) 处理流程（连续发送和接收模式）

图 13-72：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

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

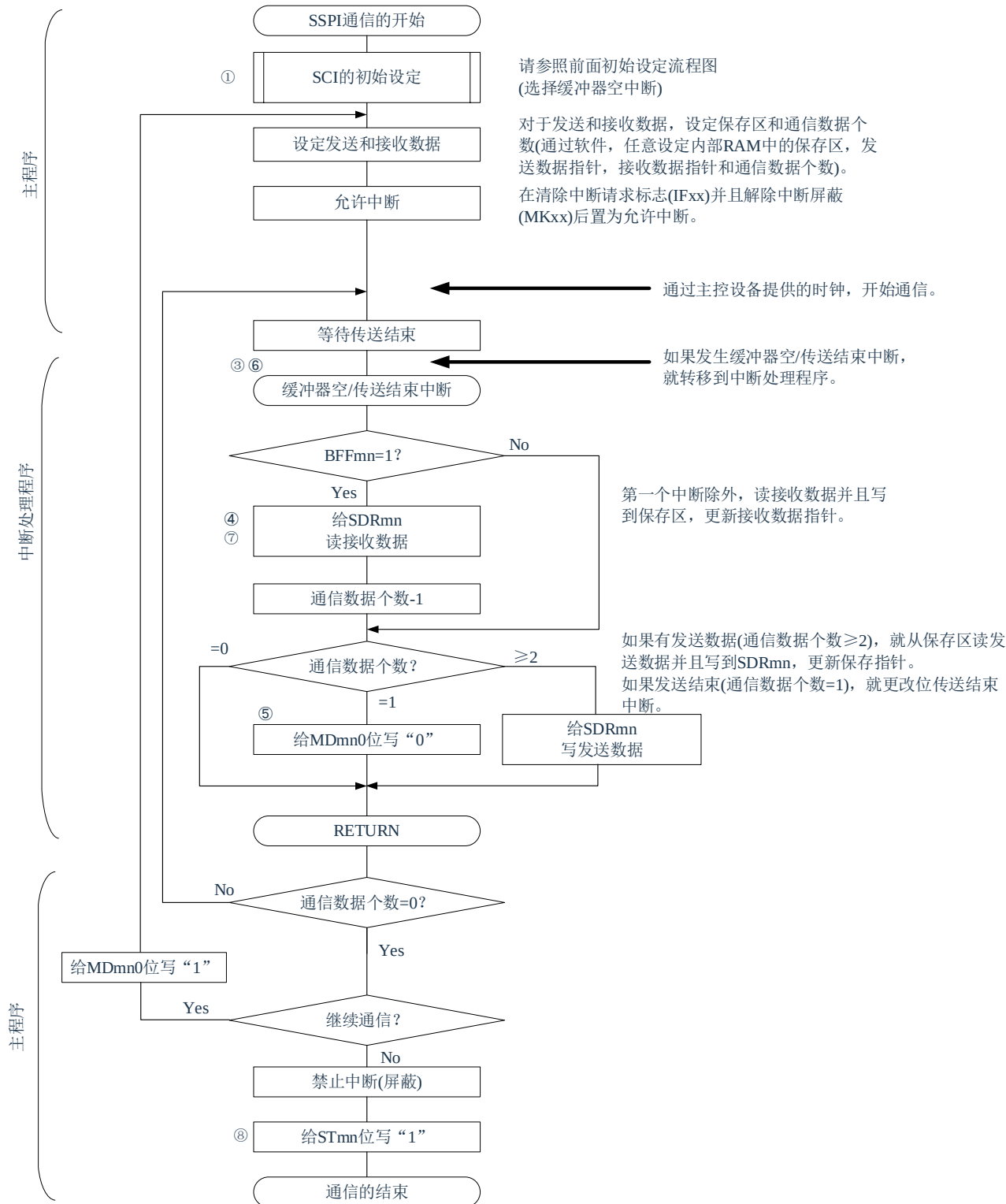
注意：即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注：

图中的①~⑧对应“图 13-73 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

m：单元号（m=0）n：通道号（n=0）p：SSPI 号（p=00）。

图 13-73: 从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

1. 图中的①~⑧对应“图 13-72 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)。

13.8.4 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。

从属设备：

$$(\text{传送时钟频率}) = \{\text{主控设备提供的串行时钟 (SCLK) 频率}\}^{\frac{1}{n}} [\text{Hz}]$$

注 1：容许的最大传送时钟频率为 6MHz。

m：单元号（m=0） n：通道号（n=0） p：SSPI 号（p=00）。

表 13-33：从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPS _m 寄存器								运行时钟（F _{MCK} ） ^注	
CKS _{mn}	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =32MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	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（ST_m）=000FH）后进行更改。

X：忽略

m：单元号（m=0） n：通道号（n=0）。

13.8.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如表 13-34 所示：

表 13-34：溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。	-	判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器mn（SDIRmn）写“1”。→	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0）n：通道号（n=0）。

13.9 UART（UART0~UART5）通信的运行

这是通过串行数据发送（TxD）和串行数据接收（RxD）共 2 条线进行异步通信的功能。使用这 2 条通信线，按数据帧（由起始位、数据、奇偶校验位和停止位构成）与其他通信方进行异步（使用内部波特率）的数据发送和接收。能通过使用发送专用（偶数通道 00/10）和接收专用（奇数通道 01/11）共 2 个通道来实现全双工异步 UART 通信。

[数据的发送和接收]

- 7 位、8 位的数据长度
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定（选择电平是否反相）
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位的检测功能

[中断功能]

- 传送结束中断、缓冲器空中断
- 帧错误、奇偶校验错误和溢出错误引起的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

UART0 使用 SCI0 的通道 0 和通道 1。

UART1 使用 SCI0 的通道 2 和通道 3。

UART2 使用 SCI1 的通道 0 和通道 1。

UART3 使用 SCI2 的通道 0 和通道 1。

UART4 使用 SCI3 的通道 0 和通道 1。

UART5 使用 SCI4 的通道 0 和通道 1。

各通道任意选择一个功能使用，除了所选功能以外，其他功能不能运行。

例如，在单元 0 的通道 0 和通道 1 使用 UART0 时，不能使用 SSPI00 和 IIC01。

UART 有以下 2 种通信运行：

- UART 发送 （参照 13.9.1）
- UART 接收 （参照 13.9.2）

注意：当用作 UART 时，发送方（偶数通道）和接收方（奇数通道）只能用于 UART。

13.9.1 UART发送

UART 发送是本产品微控制器将数据异步发送到其他设备的运行。

UART 使用的 2 个通道中的偶数通道用于 UART 发送。

表 13-35: UART 发送

UART	UART0	UART1	UART2	UART3	UART4	UART5
对象通道	SCI0的通道0	SCI0的通道2	SCI1的通道0	SCI2的通道0	SCI3的通道0	SCI4的通道0
使用的引脚	TxD0	TxD1	TxD2	TxD3	TxD4	TxD5
中断	INTST0	INTST1	INTST2	INTST3	INTST4	INTST5
	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					
传送速率	$\text{Max.F}_{\text{MCK}}/6[\text{bps}]$ （ $\text{SDRmn}[15:9] \geq 3$ ）、 $\text{Min.F}_{\text{CLK}}/(2 \times 2^{11} \times 128)[\text{bps}]^{\text{注}}$					
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。					
奇偶校验位	可选择以下内容： 无奇偶校验位。 附加零校验。 附加偶校验。 附加奇校验。					
停止位	可选择以下内容： 附加1位。 附加2位。					
数据方向	MSB优先或者LSB优先					

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

备注：

F_{MCK} ：对象通道的运行时钟频率

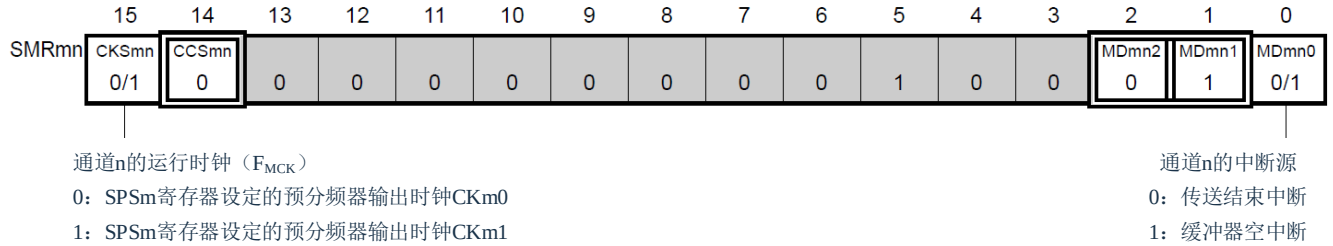
F_{CLK} ：系统时钟频率

m ：单元号（ $m=0、1、2、3、4$ ） n ：通道号（ $n=0、2$ ）。

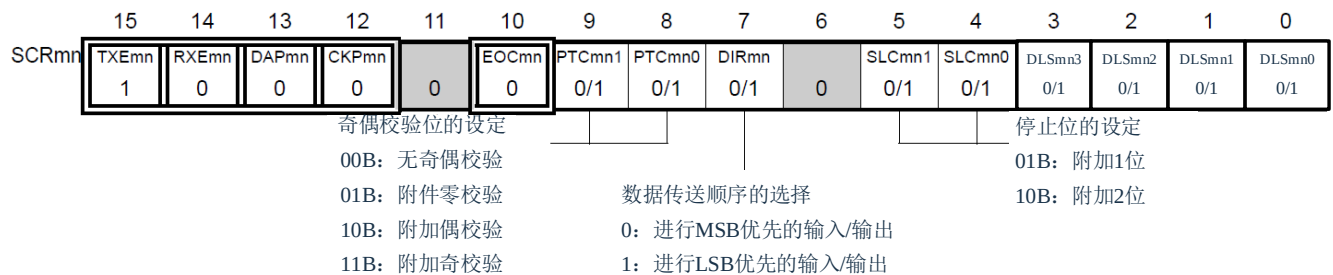
(1) 寄存器的设定

图 13-74: UART (UART0~UART5) 的 UART 发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

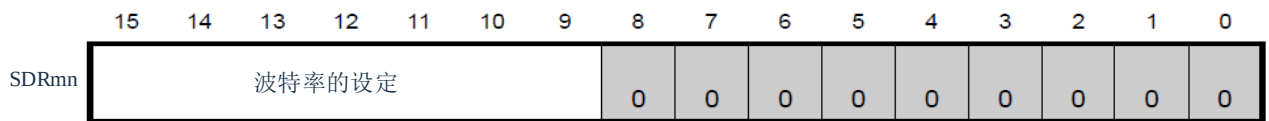


(b) 串行通信运行设定寄存器mn(SCRmn)

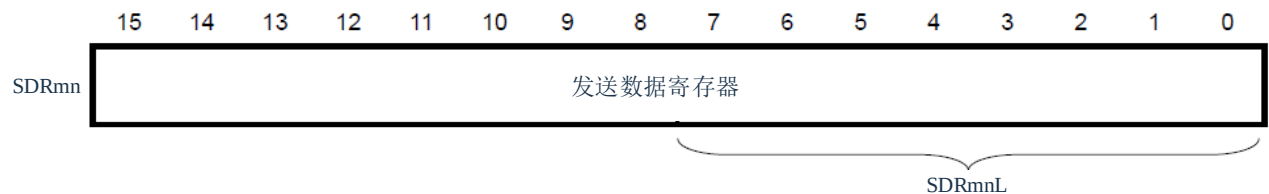


(c) 串行数据寄存器mn(SDRmn)

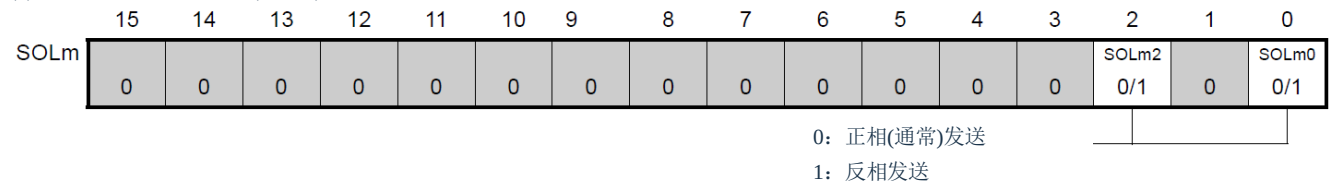
(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



(d) 串行输出电平寄存器m(SOLm) ...只设定对象通道的位。



备注: m: 单元号 (m=0、1、2、3、4) n: 通道号 (n=0、2) q: UART 号 (q=0、1、2、3、4、5)

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

图 13-74: UART (UART0~UART5) 的 UART 发送时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) ……只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 注

0: 串行数据输出值为“0”
1: 串行数据输出值为“1”

(f) 串行输出允许寄存器m(SOEm) ……只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

注意：在开始发送前，当对应通道的 SOLmn 位为“0”时，必须置“1”；当对应通道的 SOLmn 位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注：m: 单元号 (m=0、1、2、3、4) n: 通道号 (n=0、2) q: UART 号 (q=0、1、2、3、4、5)

☐: 在 SSPI 主控接收模式中为固定设定； ☐: 不能设定 (设定初始值)；

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-75: UART 发送的初始设定步骤



图 13-76: UART 发送的中止步骤

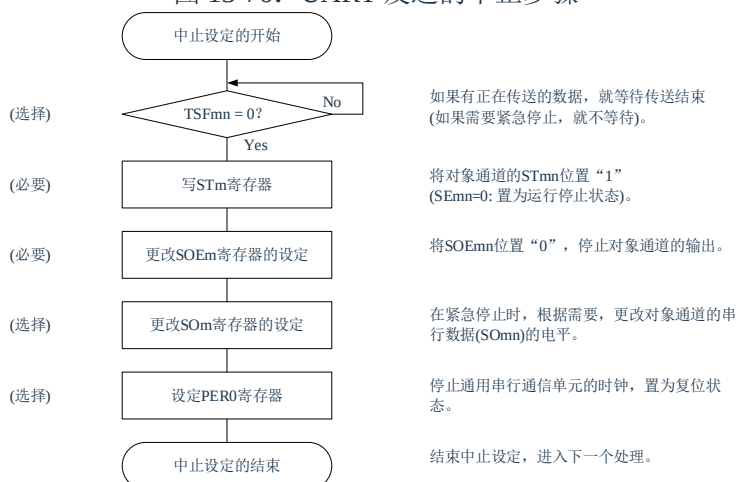
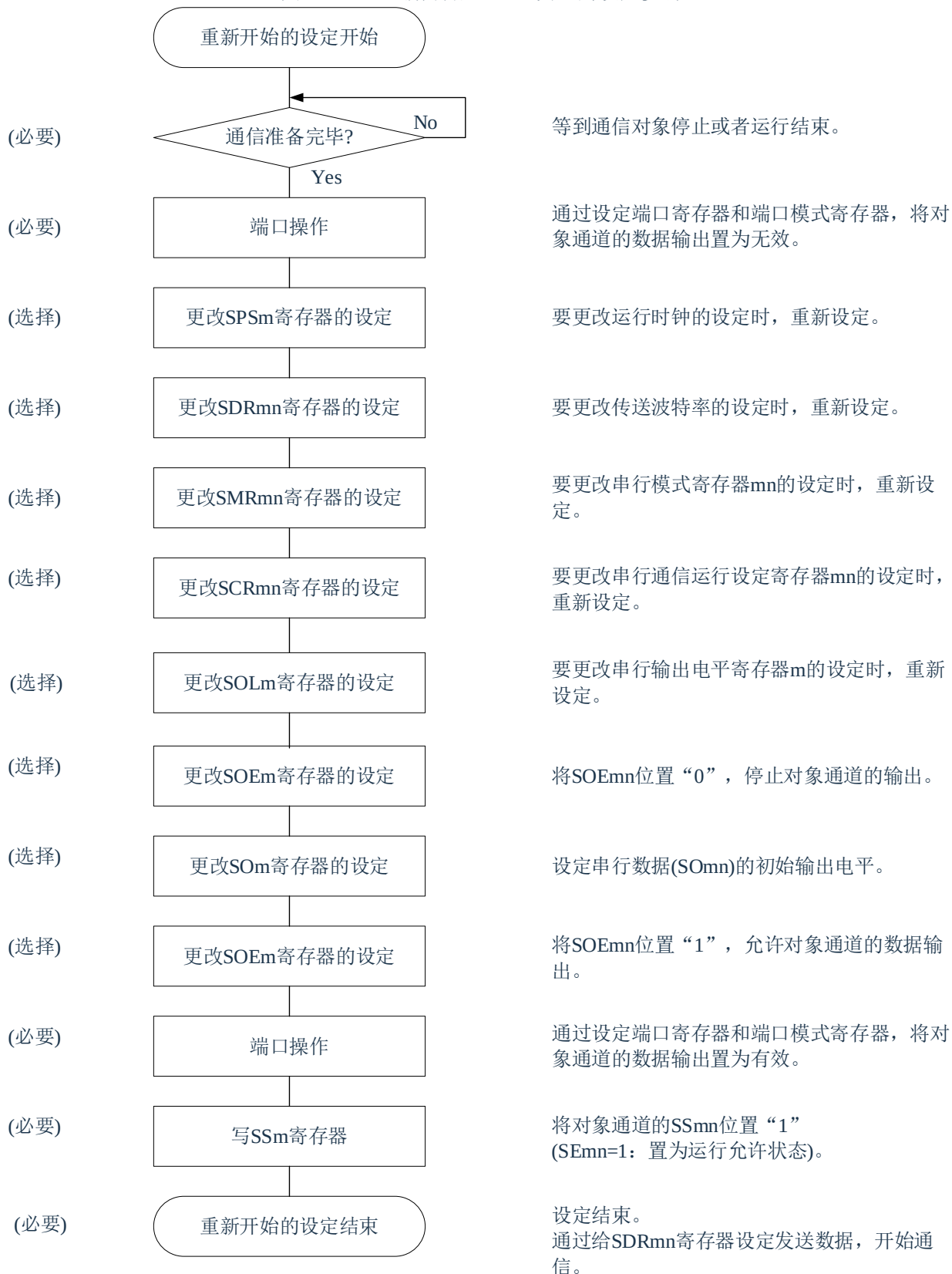


图 13-77：重新开始 UART 发送的设定步骤



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

(3) 处理流程（单次发送模式）

图 13-78: UART 发送（单次发送模式）的时序图

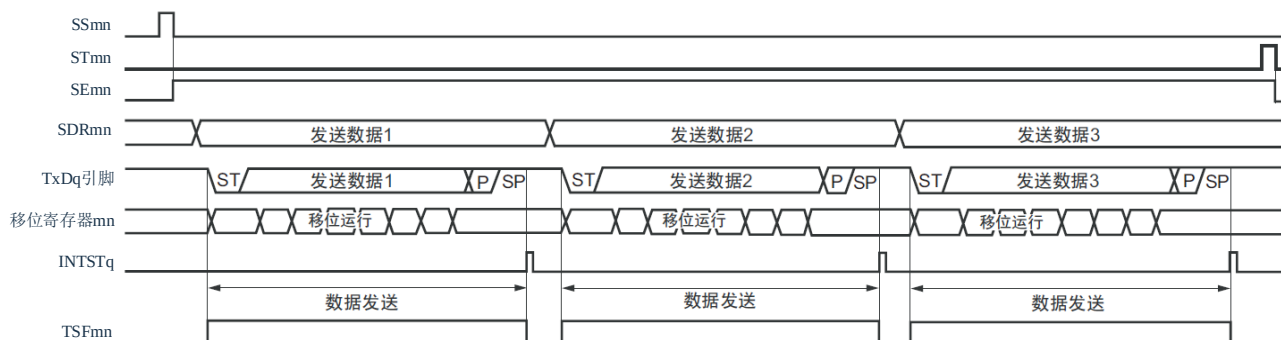
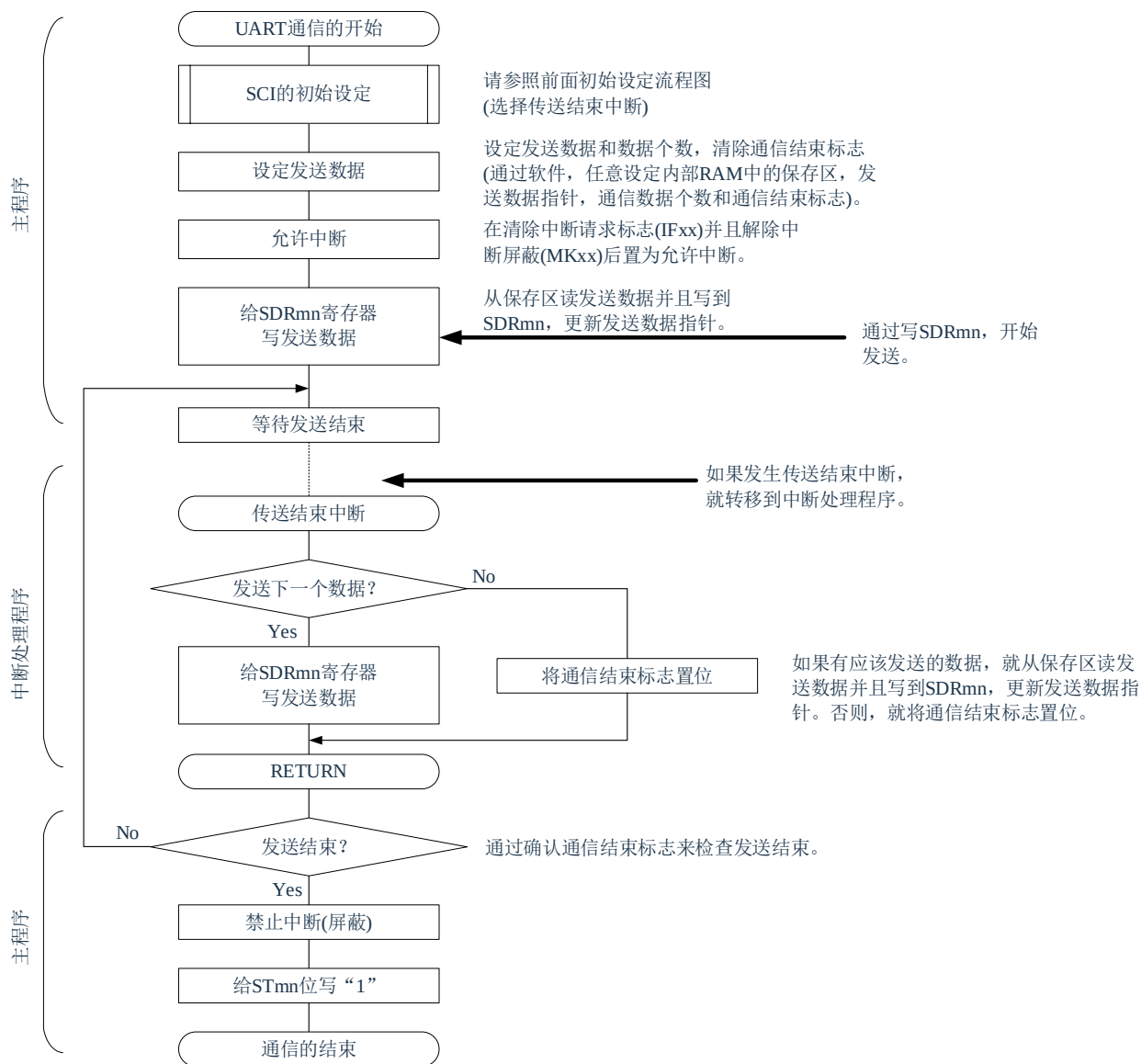


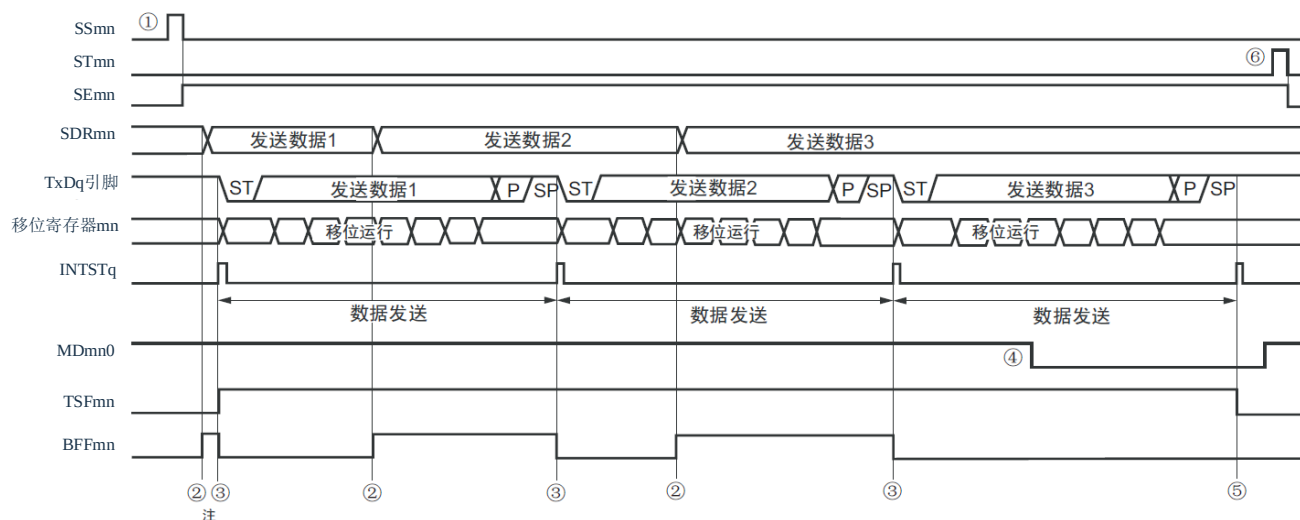
图 13-79: UART 发送（单次发送模式）的流程图



备注: m: 单元号 (m=0、1、2、3、4) n: 通道号 (n=0、2) q: UART 号 (q=0、1、2、3、4、5)。

(4) 处理流程（连续发送模式）

图 13-80：UART 发送（连续发送模式）的时序图

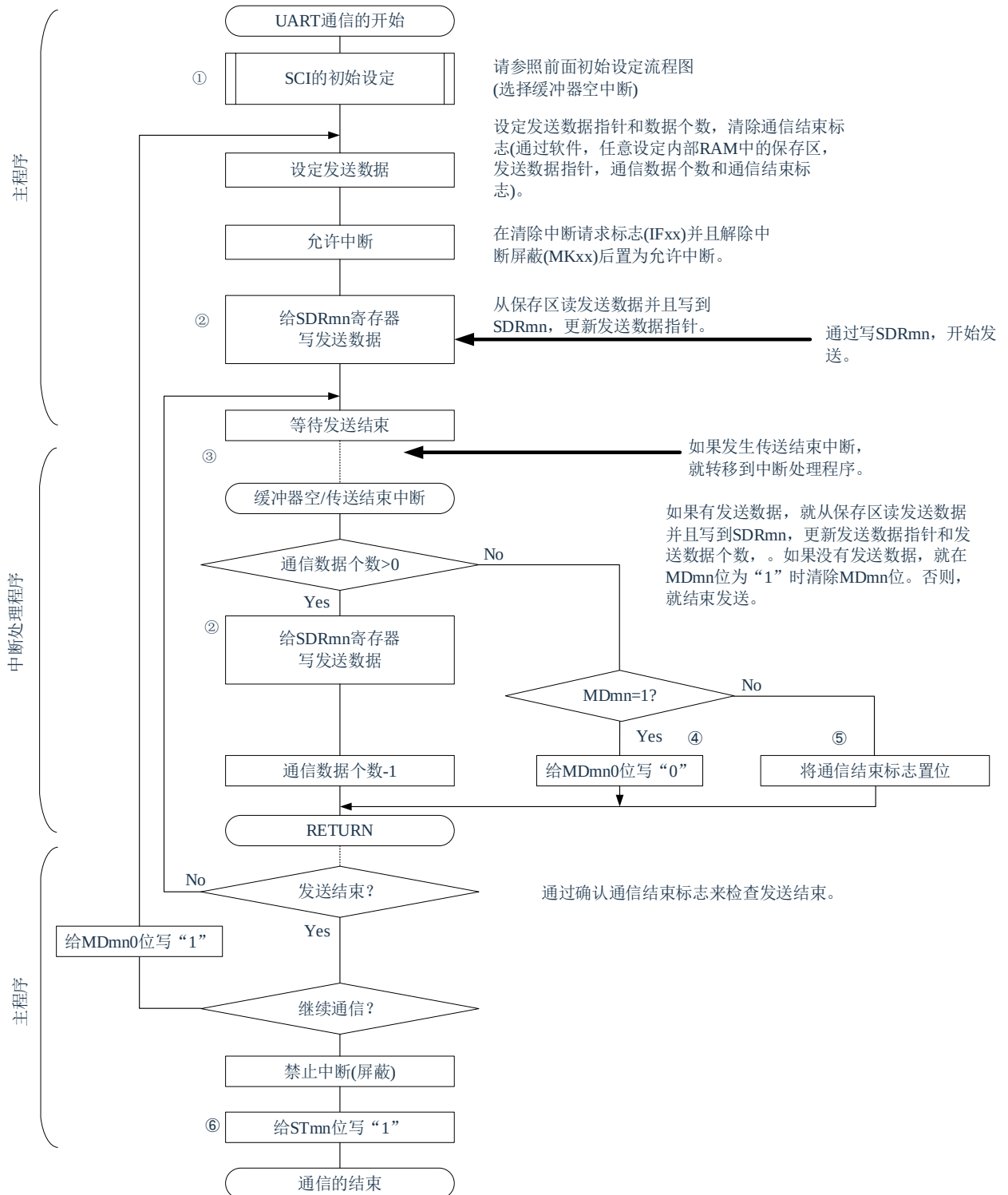


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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、2、3、4）n：通道号（n=0、2）q：UART 号（q=0、1、2、3、4、5）。

图 13-81: UART 发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 13-80 UART 发送（连续发送模式）的时序图”中的①~⑥。

13.9.2 UART接收

UART 接收是本产品微控制器其他设备异步接收数据的运行。

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

表 13-36: UART 接收

UART	UART0	UART1	UART2	UART3	UART4	UART5
对象通道	SCI0的通道1	SCI0的通道3	SCI1的通道1	SCI2的通道1	SCI3的通道1	SCI4的通道1
使用的引脚	RxD0	RxD1	RxD2	RxD3	RxD4	RxD5
中断	INTSR0	INTSR1	INTSR2	INTSR3	INTSR4	INTSR5
	只限于传送结束中断（禁止设定缓冲器空中断）。					
错误中断	INTSRE0	INTSRE1	INTSRE2	INTSRE3	INTSRE4	INTSRE5
错误检测标志	帧错误检测标志（FEFmn） 奇偶校验错误检测标志（PEFmn） 溢出错误检测标志（OVFmn）					
传送数据长度	7位、8位					
传送速率	$\text{Max.FMCK}/6[\text{bps}]$ （ $\text{SDRmn}[15:9] \geq 2$ ）、 $\text{Min.FCLK}/(2 \times 2^{15} \times 128)[\text{bps}]$					
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。					
奇偶校验位	可选择以下内容： 无奇偶校验位（无奇偶校验）。 附加零校验（无奇偶校验）。 偶校验 奇校验					
停止位	附加1位。					
数据方向	MSB优先或者LSB优先					

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

备注：

FMCK：对象通道的运行时钟频率

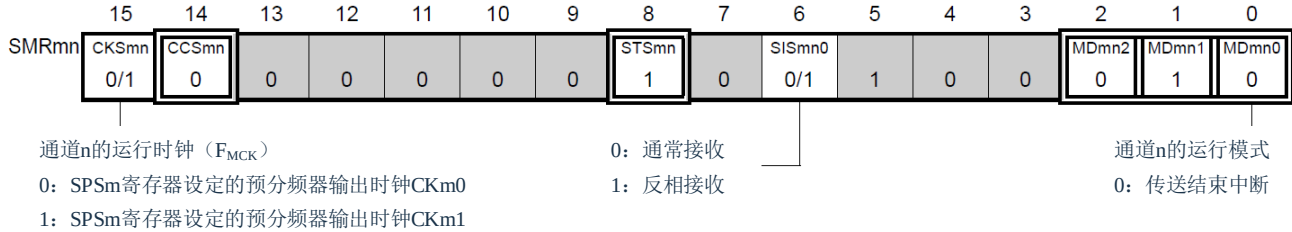
FCLK：系统时钟频率

m：单元号（m=0、1、2、3、4）n：通道号（n=1、3）。

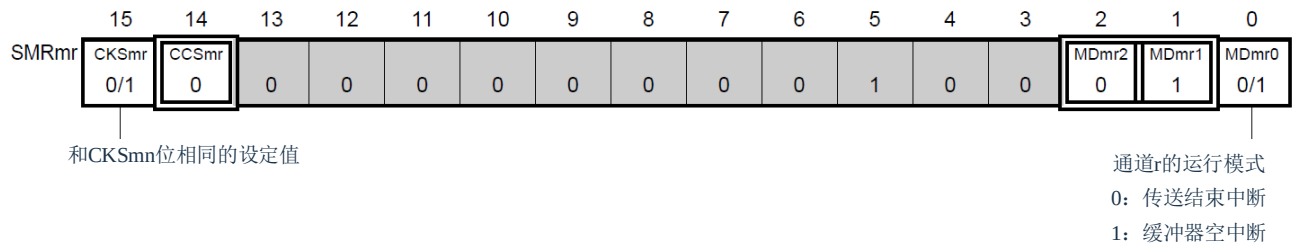
(1) 寄存器的设定

图 13-81: UART (UART0~UART5) 的 UART 接收时的寄存器设定内容例子(1/2)

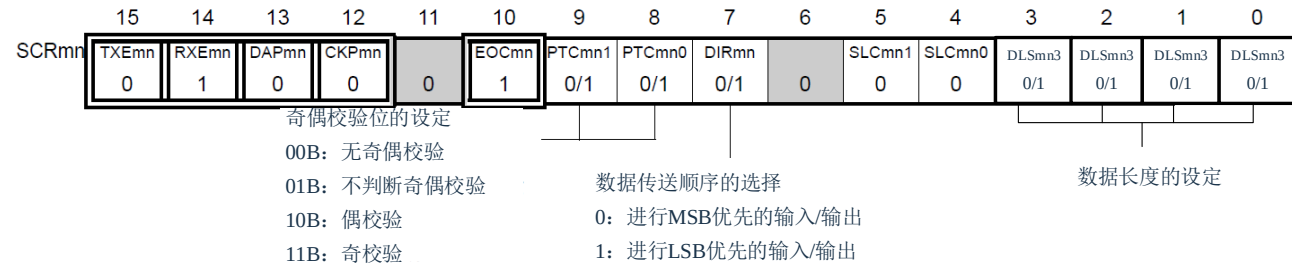
(a) 串行模式寄存器mn(SMRmn)



(b) 串行模式寄存器mr(SMRmr)

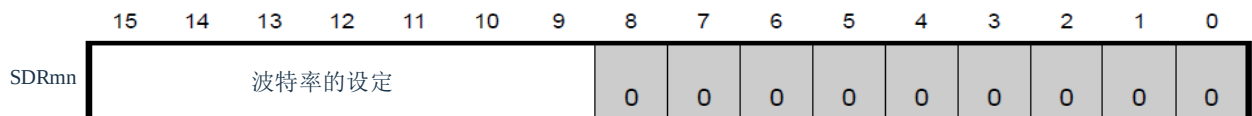


(c) 串行通信运行设定寄存器mn(SCRmn)

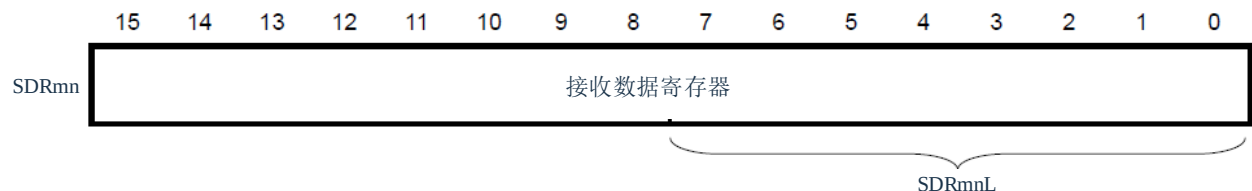


(d) 串行数据寄存器mn(SDRmn)

(1) 运行停止 (SEmn=0) 时



(2) 运行期间 (SEmn=1) (低8位: SDRmnL)



注意: 在 UART 接收时, 还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注: m: 单元号 (m=0、1、2、3、4); n: 通道号 (n=1、3);

r: 通道号 (r=n-1); q: UART 号 (q=0~5)。

□: 在 SSPI 主控接收模式中为固定设定; ■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

图 13-82: UART (UART0~UART5) 的 UART 接收时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) ……在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	×

(f) 串行输出允许寄存器m(SOEm) ……在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×

(g) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。

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

注意：在 UART 接收时，还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注：m：单元号（m=0、1、2、3、4）；n：通道号（n=1、3）；

r：通道号（r=n-1）；q：UART 号（q=0~5）。

□：在此模式中为固定设定；■：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 13-83: UART 接收的初始设定步骤



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

图 13-84: UART 接收的中止步骤

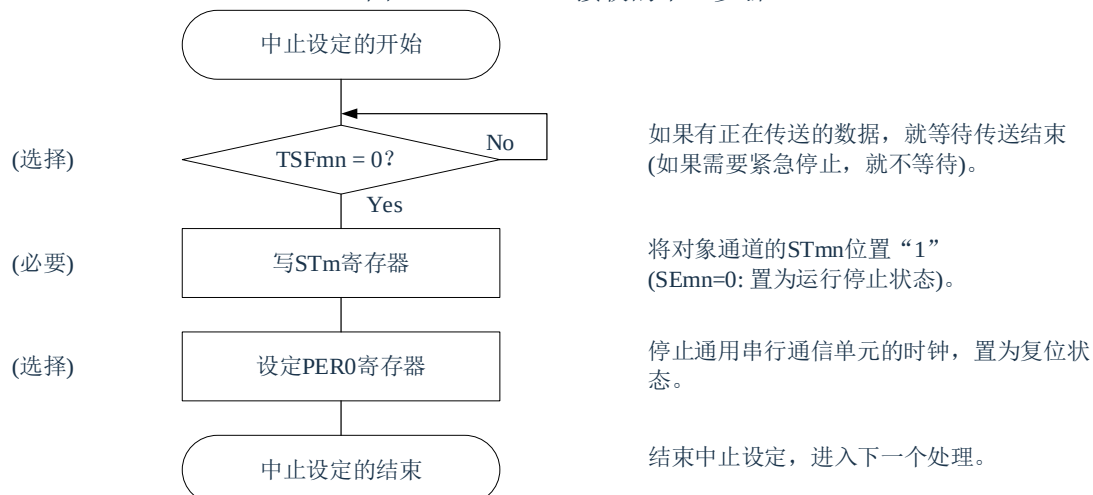
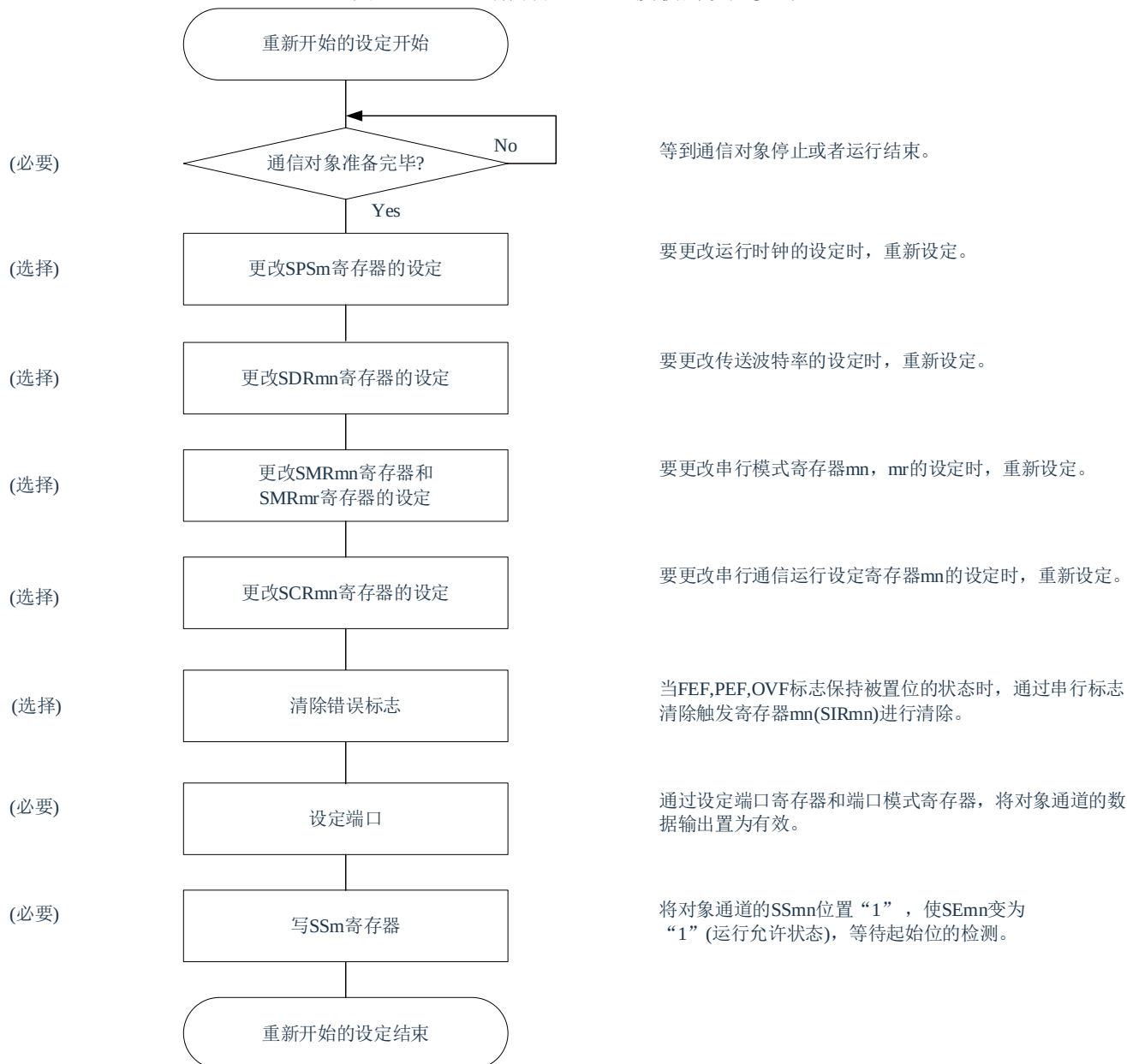


图 13-85: 重新开始 UART 接收的设定步骤



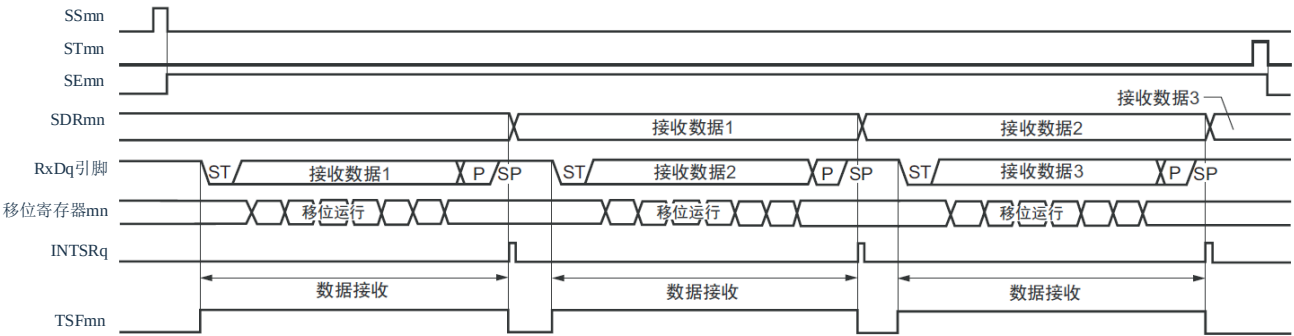
注意:

必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 FMCK 时钟，然后将 SSmn 位置“1”。

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

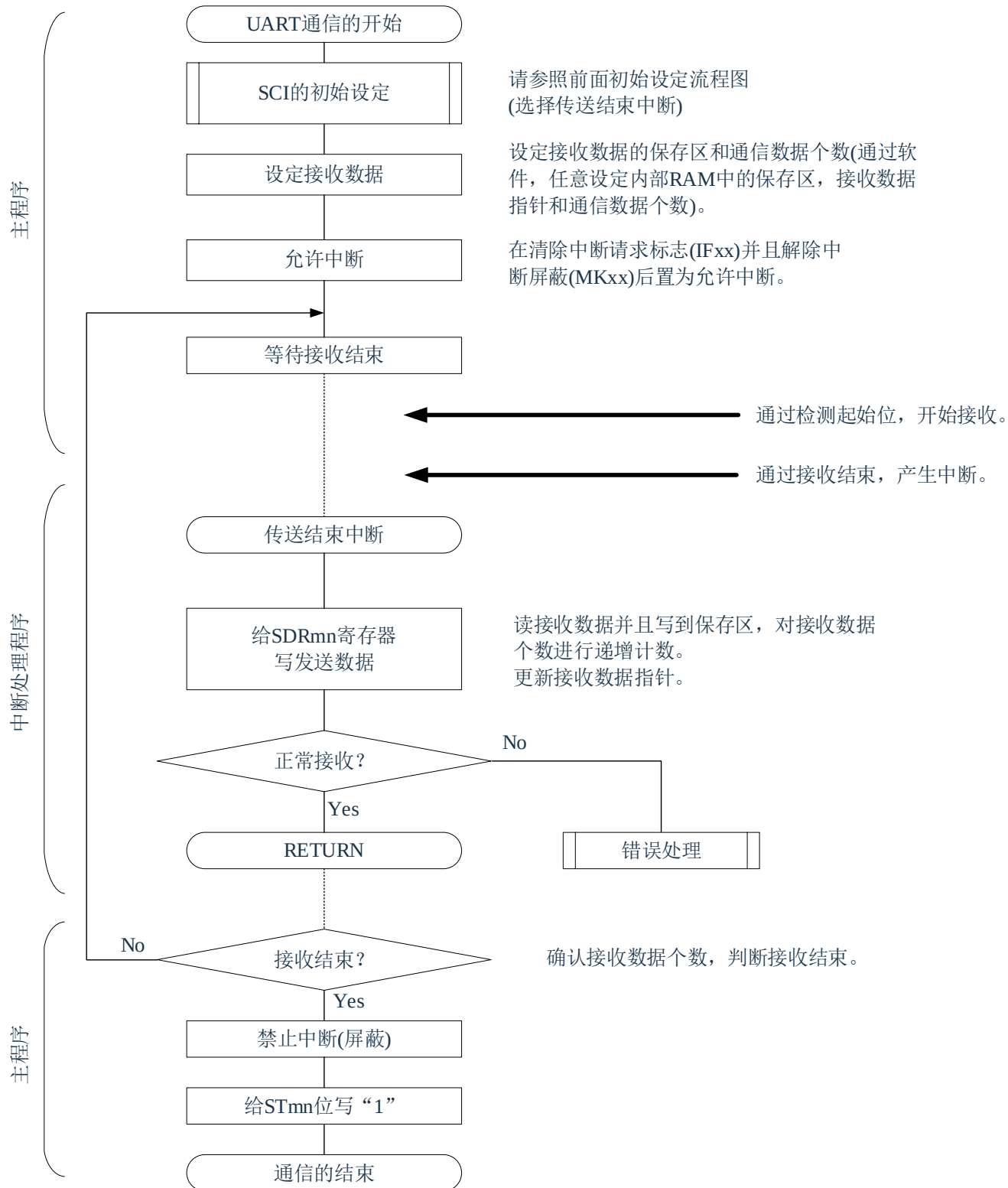
(3) 处理流程

图 13-86: UART 接收的时序图



备注：m：单元号（m=0、1、2、3、4）；n：通道号（n=1、3）；
r：通道号（r=n-1）；q：UART 号（q=0~5）。

图 13-87: UART 接收的流程图



13.9.3 波特率的计算

波特率的计算式

UART (UART0~UART5) 通信的波特率能用以下计算式进行计算：

$$(\text{波特率}) = \{ \text{对象通道的运行时钟 (F}_{\text{MCK}}) \text{ 频率} \} / (\text{SDRmn}[15:9] + 1) / 2 [\text{bps}]$$

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)。

注意：禁止将串行数据寄存器 mn (SDRmn) 的 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

备注：因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15~9 的值 (0000010B~1111111B)，所以为 2~127。

m：单元号 (m=0、1、2、3、4) n：通道号 (n=0、1、2、3)。

表 13-37: UART 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟 (F _{CLK}) ^注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F _{CLK} =32MHz运行时
0	X	X	X	X	0	0	0	0	F _{CLK}	32MHz
	X	X	X	X	0	0	0	1	F _{CLK} /2	16MHz
	X	X	X	X	0	0	1	0	F _{CLK} /2 ²	8MHz
	X	X	X	X	0	0	1	1	F _{CLK} /2 ³	4MHz
	X	X	X	X	0	1	0	0	F _{CLK} /2 ⁴	2MHz
	X	X	X	X	0	1	0	1	F _{CLK} /2 ⁵	1MHz
	X	X	X	X	0	1	1	0	F _{CLK} /2 ⁶	500KHz
	X	X	X	X	0	1	1	1	F _{CLK} /2 ⁷	250KHz
	X	X	X	X	1	0	0	0	F _{CLK} /2 ⁸	125KHz
	X	X	X	X	1	0	0	1	F _{CLK} /2 ⁹	62.5KHz
	X	X	X	X	1	0	1	0	F _{CLK} /2 ¹⁰	31.25KHz
	X	X	X	X	1	0	1	1	F _{CLK} /2 ¹¹	15.63KHz
	X	X	X	X	1	1	0	0	F _{CLK} /2 ¹²	7.81KHz
	X	X	X	X	1	1	0	1	F _{CLK} /2 ¹³	3.91KHz
	X	X	X	X	1	1	1	0	F _{CLK} /2 ¹⁴	1.95KHz
	X	X	X	X	1	1	1	1	F _{CLK} /2 ¹⁵	977Hz
1	0	0	0	0	X	X	X	X	F _{CLK}	32MHz
	0	0	0	1	X	X	X	X	F _{CLK} /2	16MHz
	0	0	1	0	X	X	X	X	F _{CLK} /2 ²	8MHz
	0	0	1	1	X	X	X	X	F _{CLK} /2 ³	4MHz
	0	1	0	0	X	X	X	X	F _{CLK} /2 ⁴	2MHz
	0	1	0	1	X	X	X	X	F _{CLK} /2 ⁵	1MHz
	0	1	1	0	X	X	X	X	F _{CLK} /2 ⁶	500KHz
	0	1	1	1	X	X	X	X	F _{CLK} /2 ⁷	250KHz
	1	0	0	0	X	X	X	X	F _{CLK} /2 ⁸	125KHz
	1	0	0	1	X	X	X	X	F _{CLK} /2 ⁹	62.5KHz
	1	0	1	0	X	X	X	X	F _{CLK} /2 ¹⁰	31.25KHz
	1	0	1	1	X	X	X	X	F _{CLK} /2 ¹¹	15.63KHz
	1	1	0	0	X	X	X	X	F _{CLK} /2 ¹²	7.81KHz
	1	1	0	1	X	X	X	X	F _{CLK} /2 ¹³	3.91KHz
	1	1	1	0	X	X	X	X	F _{CLK} /2 ¹⁴	1.95KHz
	1	1	1	1	X	X	X	X	F _{CLK} /2 ¹⁵	977Hz

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

X：忽略

m：单元号（m=0、1、2、3、4）n：通道号（n=0、1、2、3）。

发送时的波特率误差

UART（UART0~UART5）通信发送时的波特率误差能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

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

F_{CLK}=32MHz 时的 UART 波特率的设定例子如下所示：

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

备注：m：单元号（m=0、1、2、3、4）n：通道号（n=0、1、2、3）。

接收时的波特率容许范围

UART (UART0~UART5) 通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

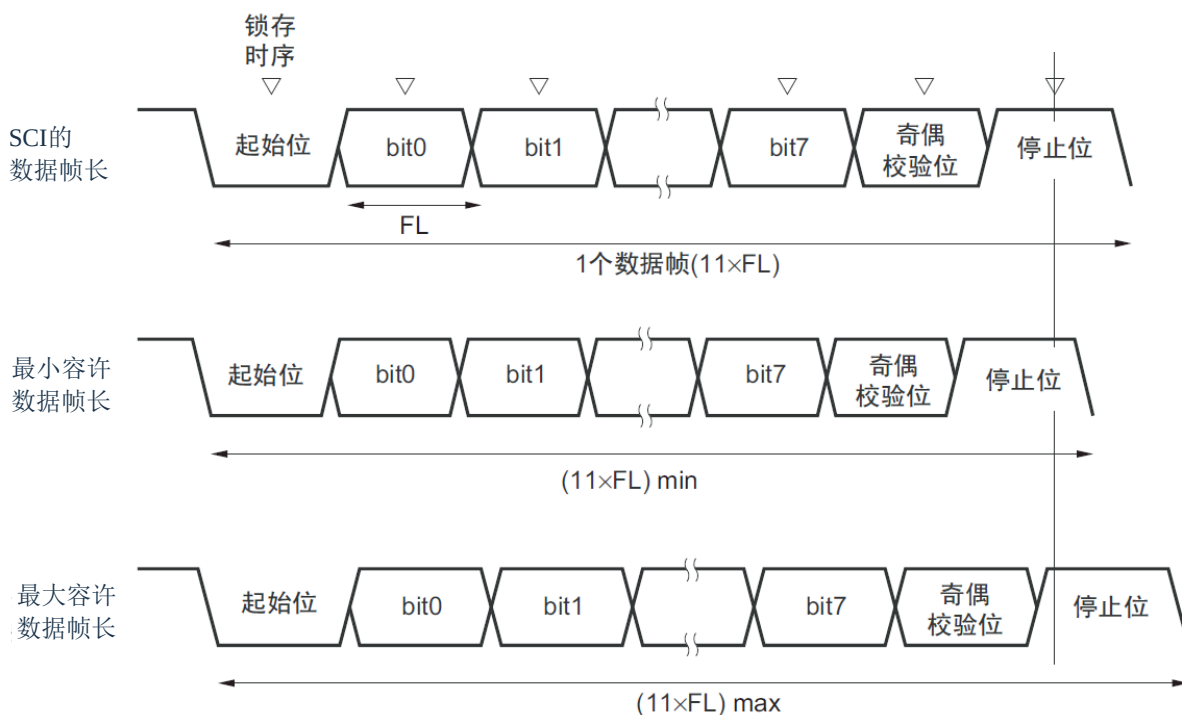
$$\begin{aligned} \text{(可接收的最大波特率)} &= \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate} \\ \text{(可接收的最小波特率)} &= \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate} \end{aligned}$$

Brate: 接收方的波特率的计算值 (参照“13.9.4 (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)。

图 13-88: 接收时的波特率容许范围 (1个数据帧=11位的情况)



如图 13-88 所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器 (SDRmn) 的 bit15~9 设定的分频比。如果最后的数据 (停止位) 能赶上此锁存时序，就能正常接收。

备注: m: 单元号 (m=0、1、2、3、4); n: 通道号 (n=0、1、2、3)。

13.9.4 在UART（UART0~UART5）通信过程中发生错误时的处理步骤

在 UART（UART0~UART5）通信过程中发生错误时的处理步骤如表 13-38 和表 13-39 所示：

表 13-38：发生奇偶校验错误或者溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器mn（SDRmn）。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。

表 13-39：发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。	SSRmn寄存器的BFFmn位为“0”并且通道n为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn（SIRmn）。	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m（STm）的STmn位置“1”。	串行通道允许状态寄存器m（SEm）的SEmn位为“0”并且通道n为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存m（SSm）的SSmn位置“1”。	串行通道允许状态寄存器m（SEm）的SEmn位为“1”并且通道n为可运行状态。	

备注：m：单元号（m=0、1、2、3、4）n：通道号（n=0、1、2、3）。

13.10 LIN通信的运行

13.10.1 LIN发送

在 UART 发送中，UART0 支持 LIN 通信。

LIN 发送使用单元 0 的通道 0。

表 13-40: LIN 发送

UART	UART0	UART1	UART2	UART3	UART4	UART5
LIN通信的支持	能	不能	不能	不能	不能	不能
对象通道	SCI0的通道0	—	—	—	—	—
使用的引脚	TxD0	—	—	—	—	—
中断	INTST0	—	—	—	—	—
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	无					
传送数据长度	8位					
传送速率 ^注	Max.F _{MCK} /6[bps]（SDR00[15:9]≥2）、Min.F _{CLK} /(2×2 ¹⁵ ×128)[bps]					
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位。					
数据方向	LSB优先					

注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用，并且在 LIN 通信中经常使用 2.4/9.6/19.2kbps。

备注：

F_{MCK}：对象通道的运行

时钟频率 F_{CLK}：系统时钟频率。

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。LIN 通信是单主控通信，一个主控设备最多可连接 15 台从属设备。

LIN 从属设备用于开关、传动装置和传感器等的控制，这些装置通过 LIN 连接到主控设备。

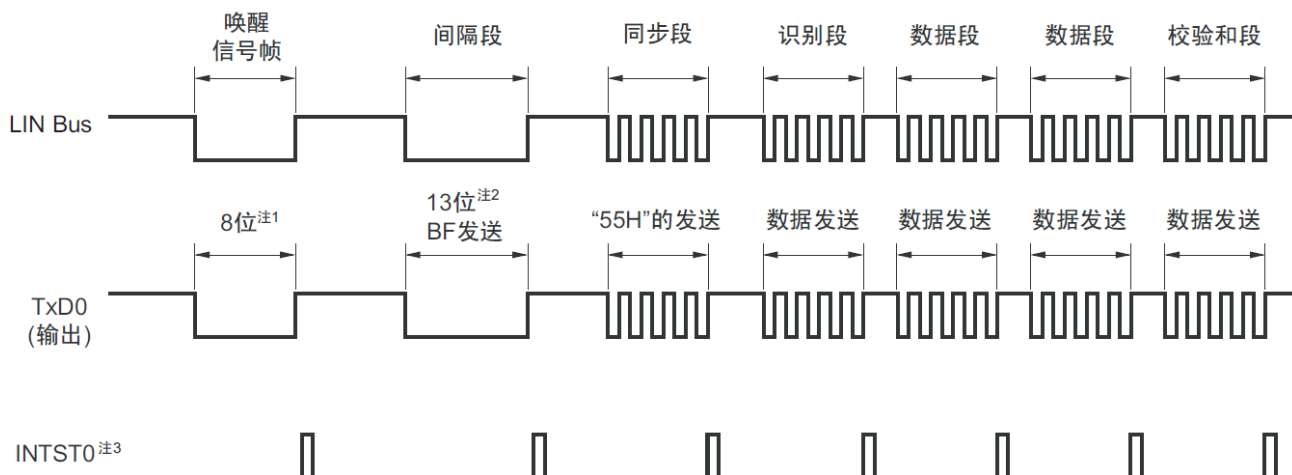
LIN 主控一般连接 CAN（Controller Area Network）等的网络。

LIN 总线是单线方式的总线，通过符合 ISDO9141 的收发器连接各节点。

根据 LIN 协议，主控设备发送附加波特率信息的帧，从属设备接收此帧并且校正与主控设备的波特率误差。因此，如果从属设备的波特率误差不大于±15%，就能进行通信。

LIN 的发送操作的概要如图 13-89 所示：

图 13-89：LIN 的发送操作



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

注 2：间隔段规定为 13 位宽的低电平输出，因此假设主传送使用的波特率为 N[bps]，间隔段使用的波特率如下：

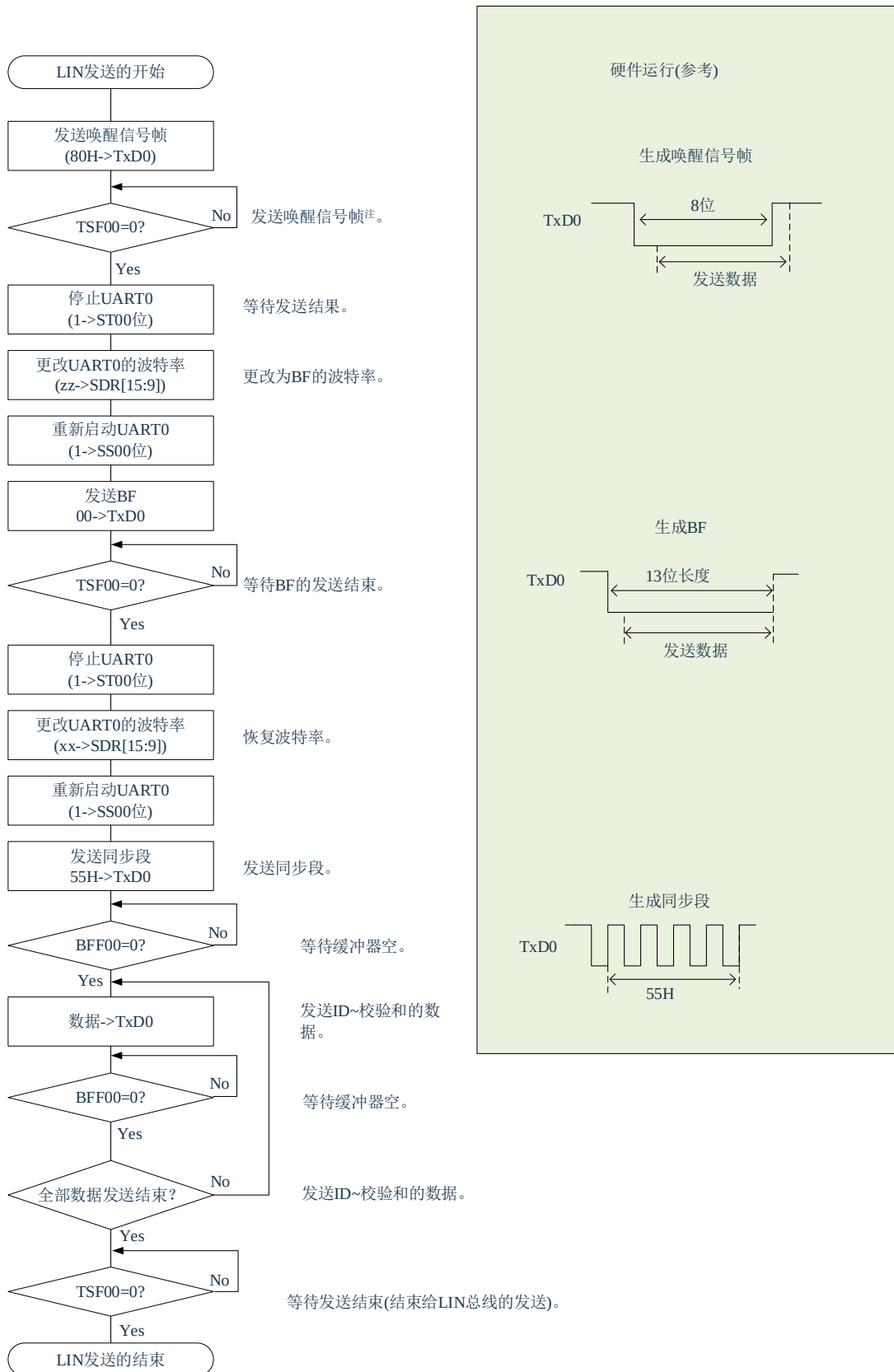
通过此波特率发送“00H”的数据，生成间隔段。

$$(\text{间隔段的波特率}) = 9/13 \times N$$

注 3：在各数据发送结束时输出 INTST0，而且在 BF 发送时也输出 INTST0。

备注：由软件控制各段间的间隔。

图 13-90: LIN 发送的流程图



备注:

只限于从 LIN-bus 睡眠状态启动的情况。

这是从结束 UART 的初始设定并且允许从属发送开始的流程。

13.10.2 LIN接收

在 UART 接收中，UART0 支持 LIN 通信。

LIN 接收使用单元 0 的通道 1。

表 13-41：LIN 接收

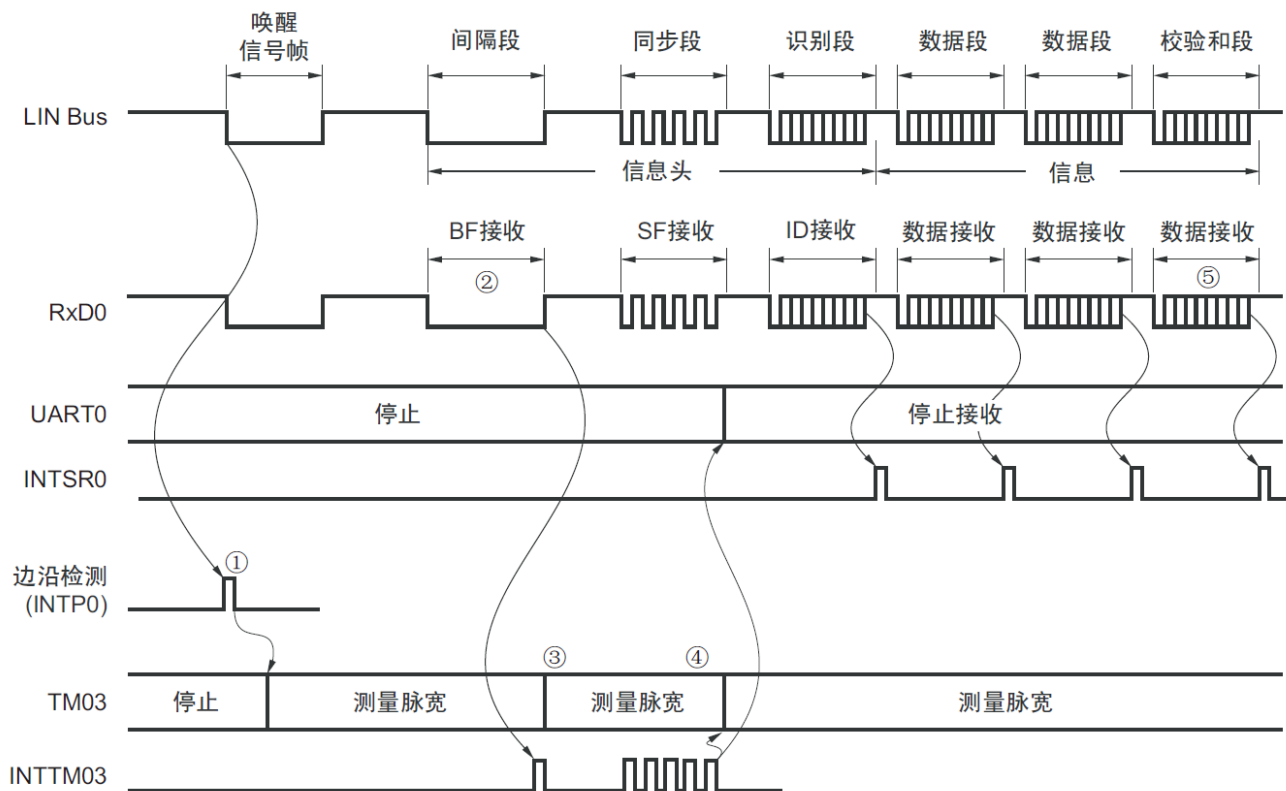
UART	UART0	UART1	UART0	UART1	UART0	UART1
LIN通信的支持	能	不能	不能	不能	不能	不能
对象通道	SCI0的通道1	—	—	—	—	—
使用的引脚	RxD0	—	—	—	—	—
中断	INTSR0	—	—	—	—	—
	只限于传送结束中断（禁止设定缓冲器空中断）。					
错误中断	INTSRE0	—	—	—	—	—
错误检测标志	帧错误检测标志（FEF01） 溢出错误检测标志（OVF01）					
传送数据长度	8位					
传送速率 ^注	Max.F _{MCK} /6[bps]（SDR01[15:9]≥2）、Min.F _{CLK} /(2 ~~×~~ 2 ¹⁵ ~~×~~ 128)[bps]					
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。					
奇偶校验位	无奇偶校验位（不进行奇偶校验）。					
停止位	附加1位。					
数据方向	LSB优先					

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

F_{MCK}：对象通道的运行 3、时钟频率 F_{CLK}：系统时钟频率。

LIN 接收操作的概要如图 13-91 所示：

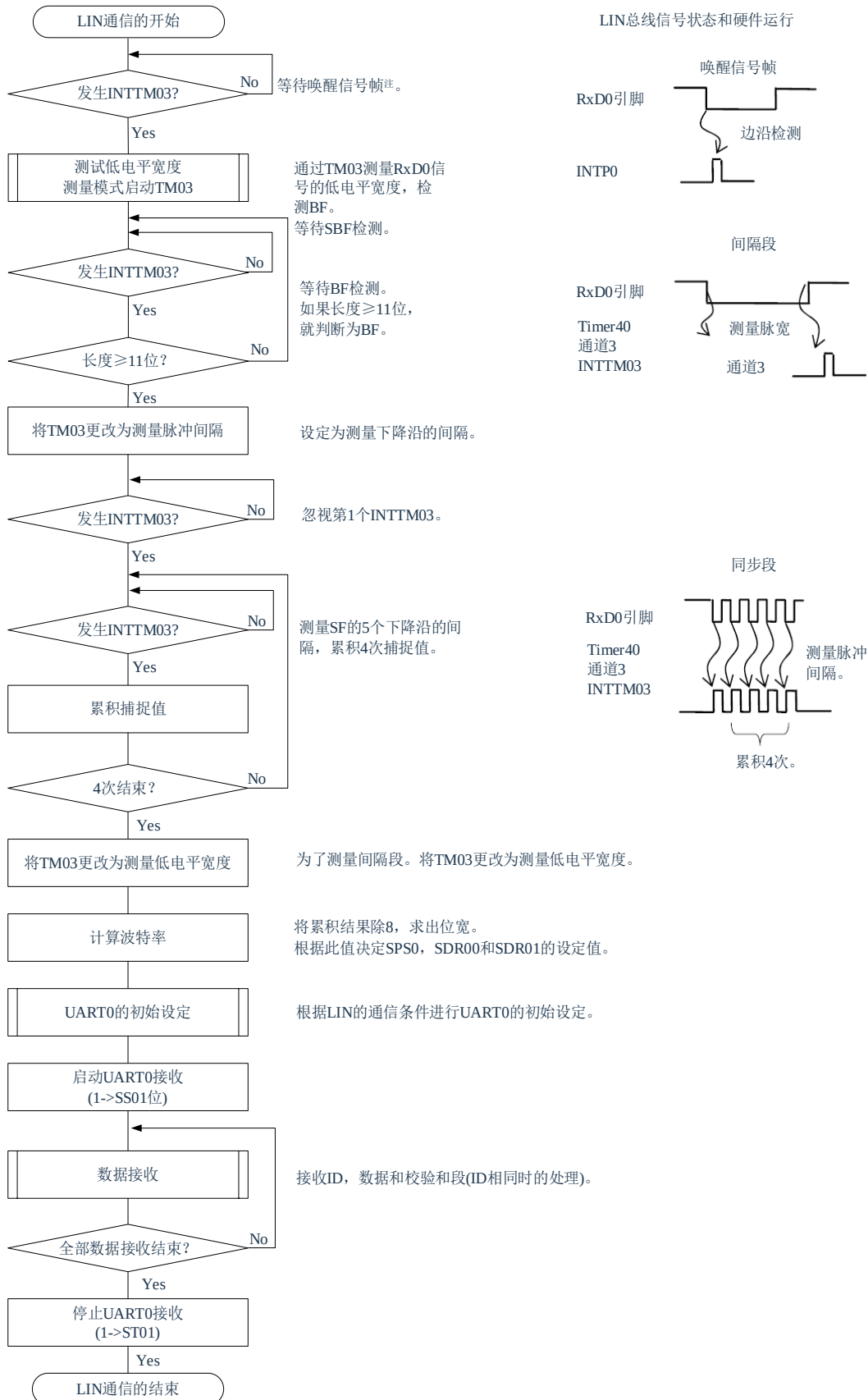
图 13-91：LIN 的接收操作



信号处理的流程如下：

1. 通过检测引脚的中断边沿（INTP0）来检测唤醒信号。当检测到唤醒信号时，为了测量 BF 的低电平宽度，将 TM03 设定为测量脉宽，然后进入 BF 接收等待状态。
2. 如果检测到 BF 的下降沿，TM03 就开始测量低电平宽度，并且在 BF 的上升沿进行捕捉。根据捕捉到的值判断是否为 BF 信号。
3. 当 BF 接收正常结束时，必须将 TM03 设定为测量脉冲间隔，并且测量 4 次同步段的 RxD0 信号下降沿的间隔（参照“6.7.4 作为输入脉冲间隔测量的运行”）。
4. 根据同步段（SF）的位间隔计算波特率误差。然后，必须在暂停 UART0 运行后调整（重新设定）波特率。
5. 必须通过软件区分校验和段。还必须通过软件在接收校验和段后对 UART0 进行初始化并且再次设定为 BF 接收等待状态。

图 13-92: LIN 接收的流程图



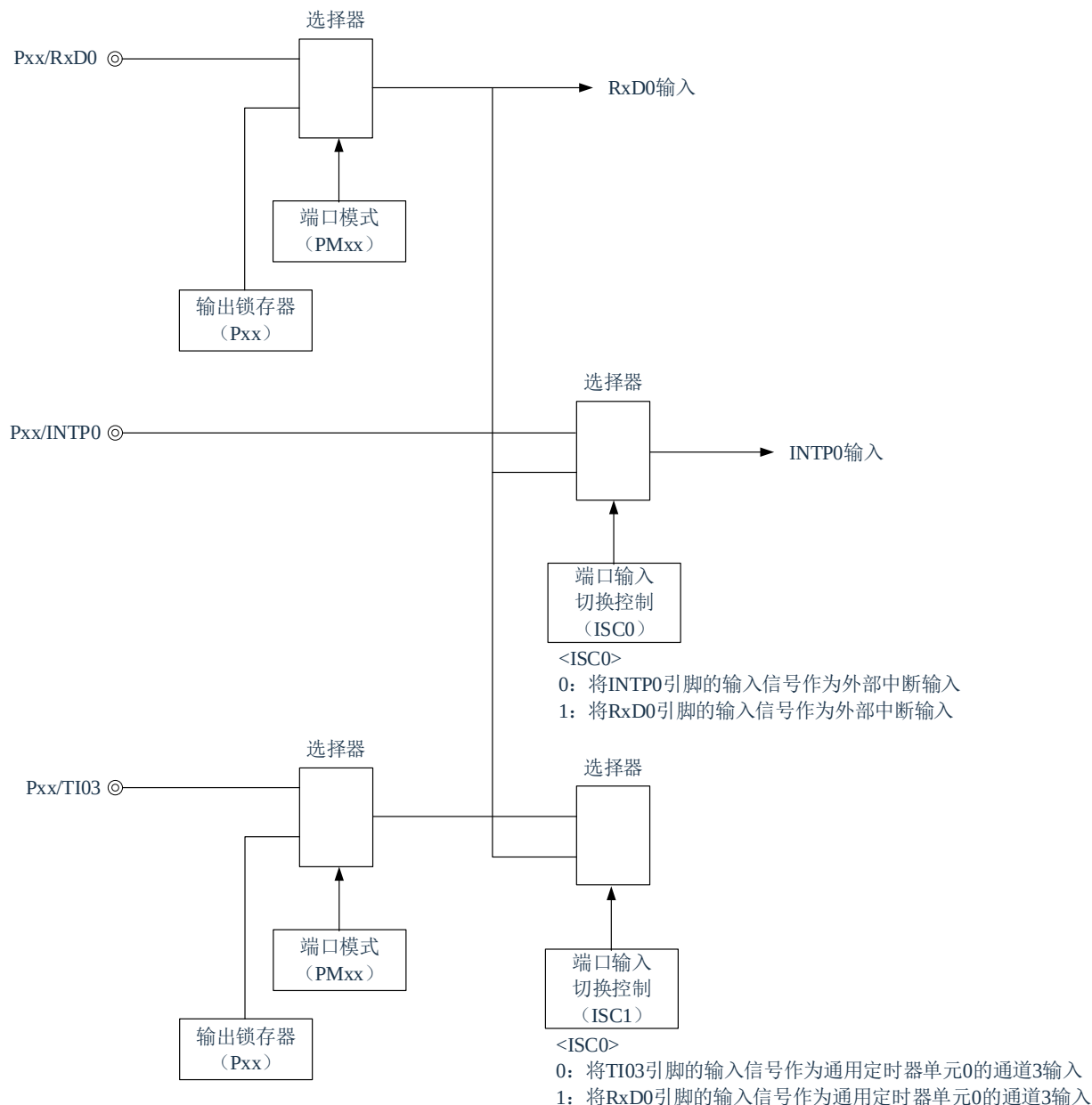
注意：只在睡眠状态下才需要。

用于 LIN 接收操作的端口结构图如图 13-93 所示。

通过外部中断（INTP0）的边沿检测，接收 LIN 主控发送的唤醒信号。能通过通用定时器单元的外部事件捕捉运行，测量 LIN 主控发送的同步段的长度以及计算波特率误差。

通过端口输入切换控制（ISC0/ISC1），能不在外部连线而将用于接收的端口输入（RxD0）的输入源输入到外部中断（INTP0）和定时器阵列单元。

图 13-93：用于 LIN 接收操作的端口结构图



备注：ISC0、ISC1：输入切换控制寄存器（ISC）的 bit0 和 bit1

用于 LIN 通信运行的外围功能总结如下：

<使用的外围功能>

外部中断（INTP0）：唤醒信号的检测。

用途：检测唤醒信号的边沿和通信的开始。

通用定时器单元的通道 3：波特率误差的检测、间隔段（BF）的检测。

用途：检测同步段（SF）的长度，并且通过将其长度除以位数来检测波特率误差（通过捕捉模式测量 RxD0 输入边沿的间隔）。测量低电平宽度，判断是否为间隔段（BF）。

通用串行通信单元 0（SCI0）的通道 0 和通道 1（UART0）。

13.11 简易I2C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的运行

这是通过串行时钟（SCL）和串行数据（SDA）共 2 条线与多个设备进行时钟同步通信的功能。因为此简易 I²C 是为了与 EEPROM、闪存、A/D 转换器等设备进行单通信而设计的，所以只用作主控设备。

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。

[数据的发送和接收]

- 主控发送、主控接收（只限于单主控的主控功能）
- ACK 输出功能注、ACK 检测功能
- 8 位数据长度（在发送地址时，用高 7 位指定地址，用最低位进行 R/W 控制）
- 通过软件产生开始条件和停止条件。

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误

[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 多主控功能（仲裁失败检测功能）
- 等待检测功能

SCI0 的通道 0~1、2~3 以及 SCI1 的通道 0~1 是支持简易 I²C(IIC00、IIC01、IIC10、IIC11、IIC20、IIC21)的通道。

简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）有以下 4 种通信运行：

- 地址段发送（参照 13.11.1）
- 数据发送（参照 13.11.2）
- 数据接收（参照 13.11.3）
- 停止条件的产生（参照 13.11.4）

注意：在接收最后的数据时，如果给 SDOEmn 位（SDOEm 寄存器）写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“13.11.3 (2)处理流程”。

备注：m：单元号（m=0、1）n：通道号（n=0、1、2、3）。

13.11.1 地址段发送

地址段发送是为了特别指定传送对象（从属设备）而在 I²C 通信时最先进行的发送运行。在产生开始条件后，将地址（7 位）和传送方向（1 位）作为 1 帧进行发送。

表 13-42：地址段发送

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、 SAU_SDA00 ^{注1}	SCLKOI01、 SAU_SDA01 ^{注1}	SCLKOI10、 SAU_SDA10 ^{注1}	SCLKOI11、 SAU_SDA11 ^{注1}	SCLKOI20、 SAU_SDA20	SCLKOI21、 SAU_SDA21
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK错误检测标志（PEFmn）					
传送数据长度	8位（将高7位作为地址并且将低1位作为R/W控制进行发送）					
传送速率 ^{注2}	Max.FMCK/4[Hz]（SDRmn[15:9]≥1）FMCK：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）					
数据电平	正相输出（默认值：高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位（用于ACK接收）。					
数据方向	MSB优先					

注 1：要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出模式（POMxx=1）。

详细内容请参照“第 2 章 引脚功能”

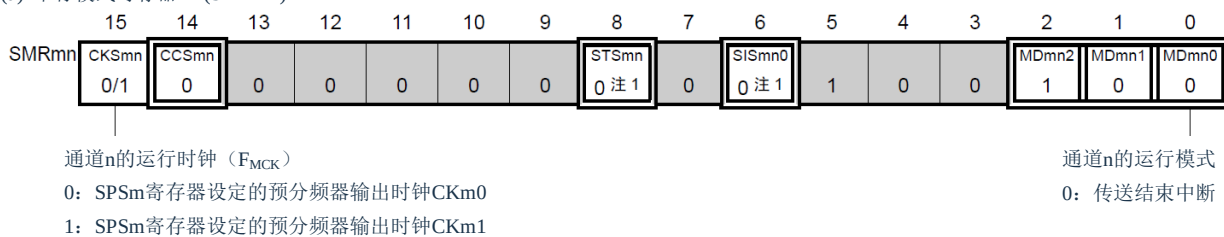
注意：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：m：单元号（m=0、1）n：通道号（n=0、1、2、3）

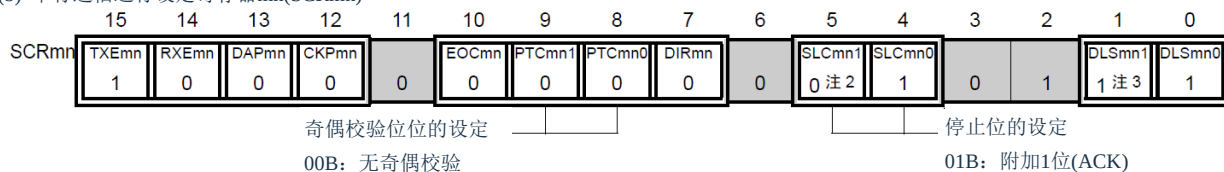
(1) 寄存器的设定

图 13-94：简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）地址段发送时的寄存器设定内容例子

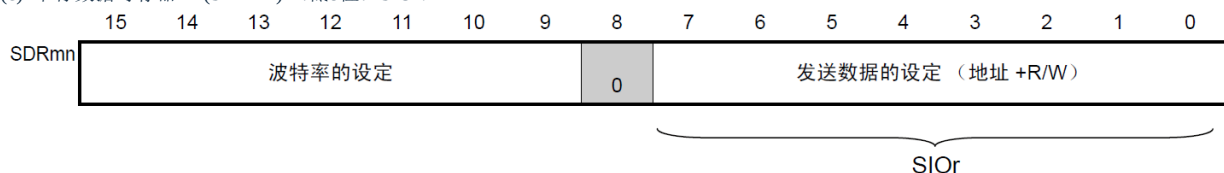
(a) 串行模式寄存器mn(SMRmn)



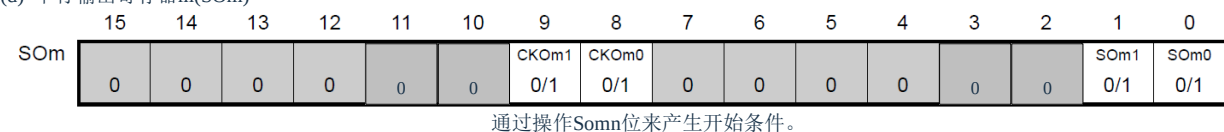
(b) 串行通信运行设定寄存器mn(SCRmn)



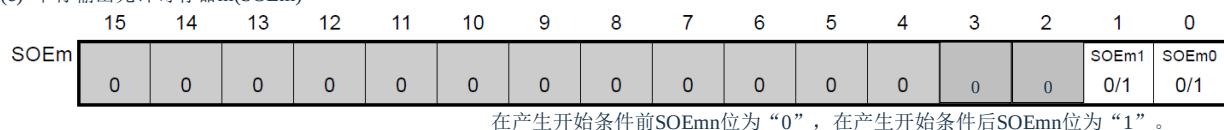
(c) 串行数据寄存器mn(SDRmn) (低8位: SIO_r)



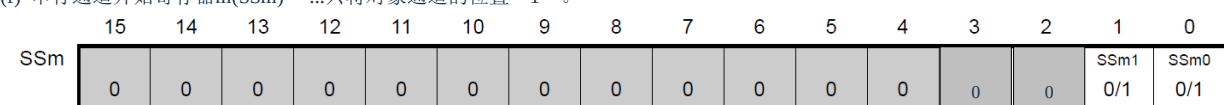
(d) 串行输出寄存器m(SOm)



(e) 串行输出允许寄存器m(SOEm)



(f) 串行通道开始寄存器m(SSm) ……只将对象通道的位置“1”。



注 1: 只限于 SMR01、SMR03、SMR11。

注 2: 只限于 SCR00、SCR02、SCR10。

注 3: 只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

m: 单元号 (m=0、1) n: 通道号 (n=0、1、2、3) r: IIC 号 (r=00、01、10、11、20、21)

□: 在 IIC 模式中为固定设定; ■: 不能设定 (设定初始值);

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

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

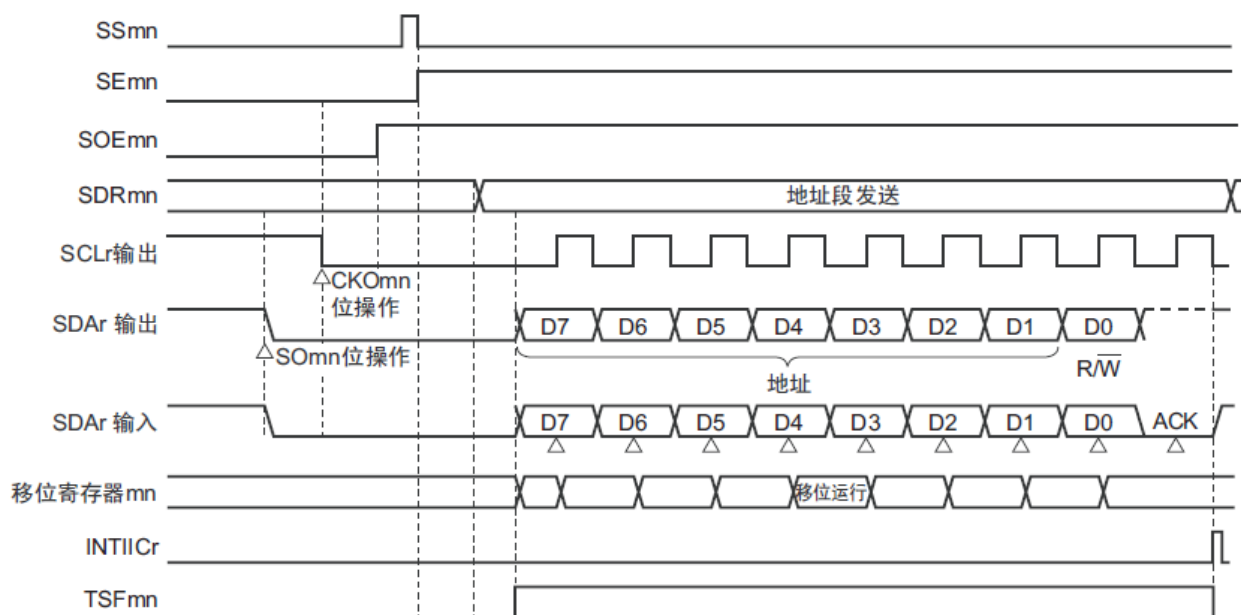
图 13-95: 地址段发送的初始设定步骤



备注：在初始设定结束时，简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）为禁止输出并且处于运行停止状态。

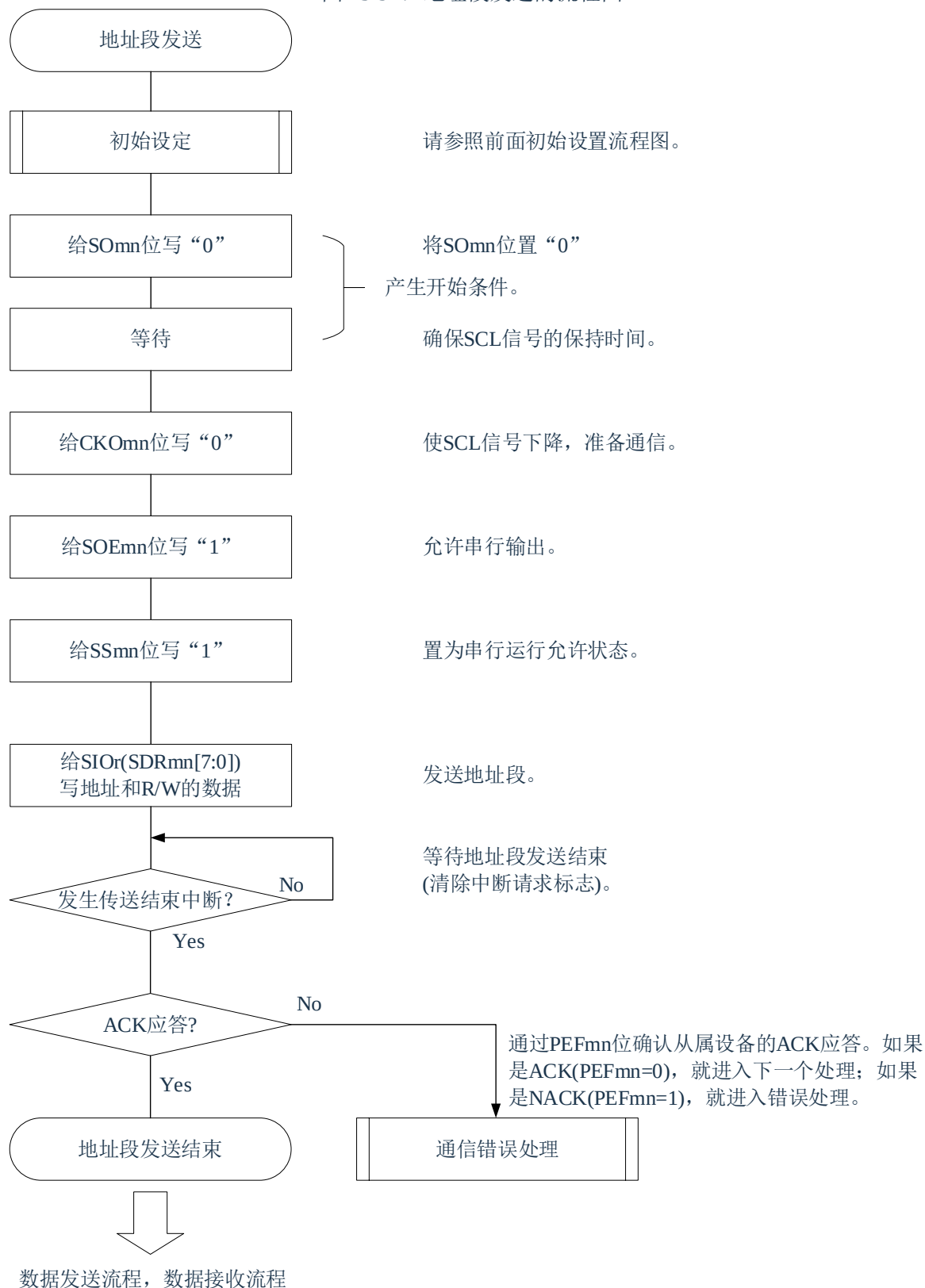
(3) 处理流程

图 13-96: 地址段发送的时序图



备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1、2、3) r: IIC 号 (r=00、01、02、03、10、11)

图 13-97：地址段发送的流程图



13.11.2 数据发送

数据发送是在发送地址段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

表 13-43：数据发送

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、 SAU_SDA00 ^{注1}	SCLKOI01、 SAU_SDA01	SCLKOI10、 SAU_SDA10 ^{注1}	SCLKOI11、 SAU_SDA11 ^{注1}	SCLKOI20、 SAU_SDA20	SCLKOI21、 SAU_SDA21
中断	INTIIC00	INTIIC01 ^{注1}	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK错误标志（PEFmn）					
传送数据长度	8位					
传送速率 ^{注2}	Max.FMCK/4[Hz]（SDRmn[15:9]≥1） FMCK：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）					
数据电平	正相输出（默认值：高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位（用于ACK接收）。					
数据方向	MSB优先					

注 1：要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出模式（POMxx=1）。

详细内容请参照“2.2 端口复用功能”和“2.3.2 输出输入复用功能寄存器映射”。

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

备注：m：单元号（m=0、1） n：通道号（n=0、1、2、3）

(1) 寄存器的设定

图 13-98：简易 I2C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）数据发送时的寄存器设定内容例子

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
0/1	0	0	0	0	0	0	0 注 1	0	0 注 1	1	0	0	1	0	0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
1	0	0	0	0	0	0	0	0	0	0 注 2	1	0	1	1 注 3	1

(c) 串行数据寄存器mn(SDRmn)（低8位：SIOr）……在数据发送和接收的过程中只有低8位有效(SIOr)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定注 4							0	发送数据的设定							
									SIOr							

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						CKOm1	CKOm0							SOM1	SOM0
0	0	0	0	0	0	0/1 注 5	0/1 注 5	0	0	0	0	0	0	0/1 注 5	0/1 注 5

(e) 串行输出允许寄存器m(SOEm)……在数据发送和接收的过程中不操作此寄存器。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														SOEm1	SOEm0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

(f) 串行通道开始寄存器m(SSm)……在数据发送和接收的过程中不操作此寄存器。

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

注 1：只限于 SMR01、SMR03、SMR11 寄存器。

注 2：只限于 SCR00、SMR02、SCR10 寄存器。

注 3：只限于 SCR00 和 SCR01 寄存器，其他固定为“1”。

注 4：因为已经在发送地址段时设定，所以不需要设定。

注 5：在通信运行过程中，值因通信数据而变。

备注：m：单元号（m=0、1）n：通道号（n=0、1）r：IIC 号（r=00、01、02、03、10、11）

□：在 IIC 模式中为固定设定；

■：不能设定（设定初始值）；

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

0/1：根据用户的用途置“0”或者“1”。

(2) 处理流程

图 13-99: 数据发送的时序图

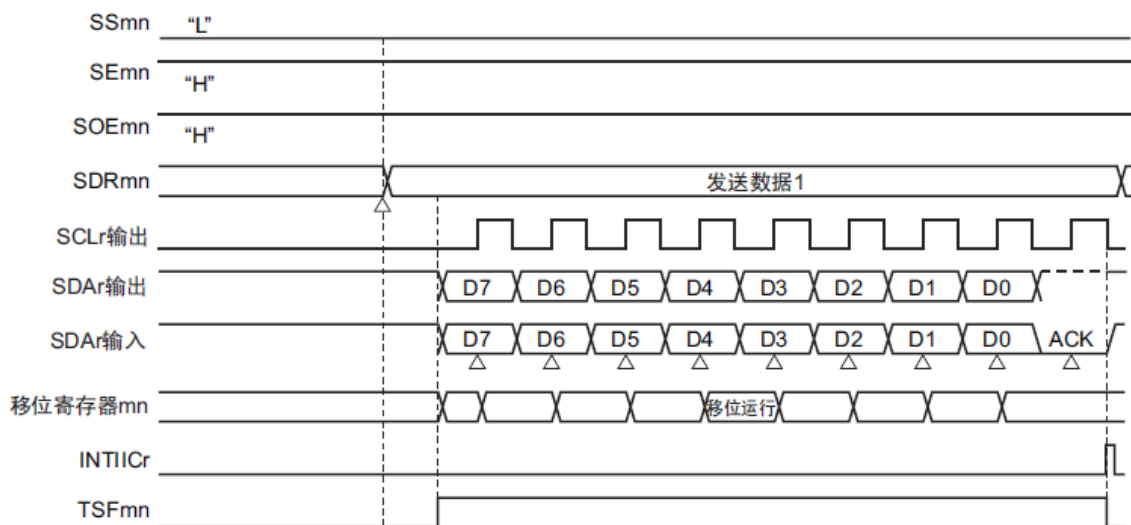
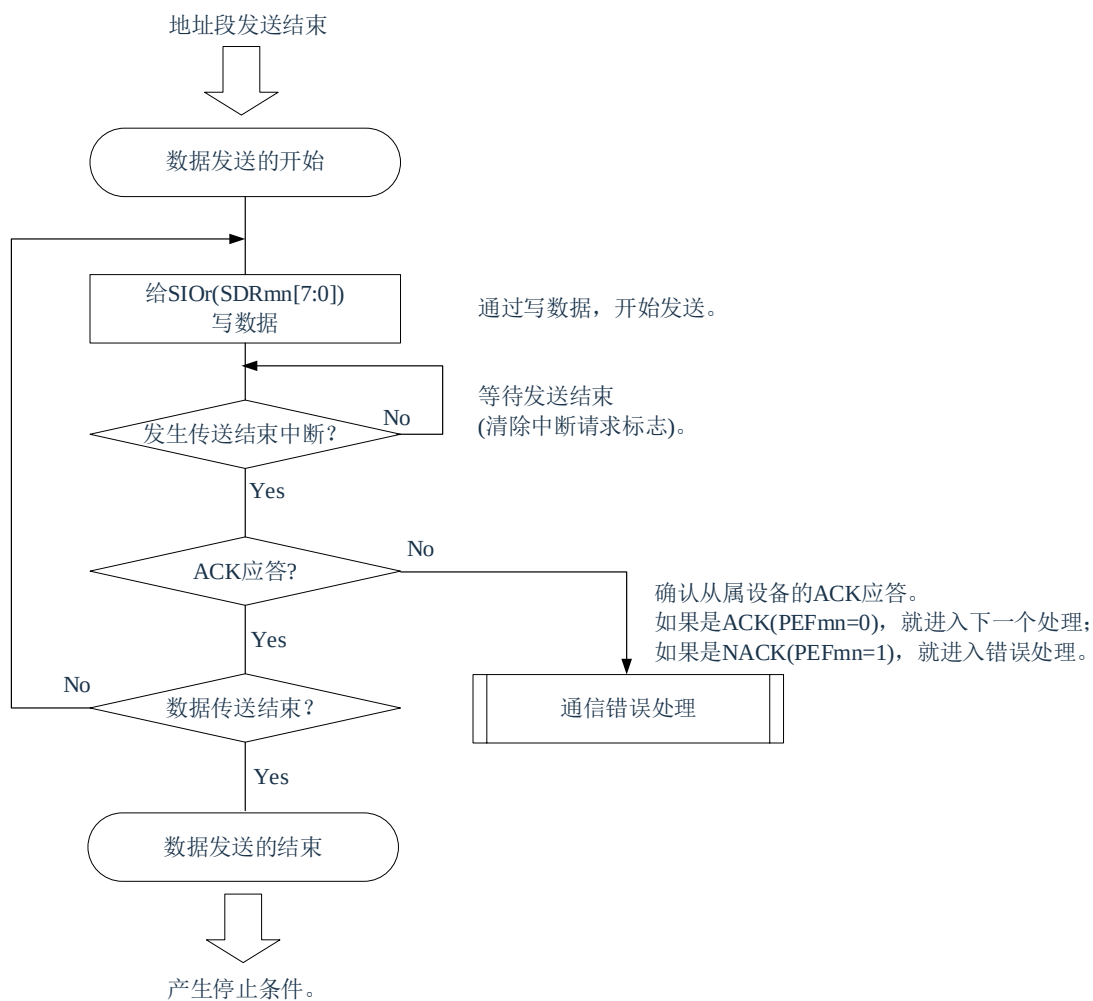


图 13-100: 数据发送的流程图



13.11.3 数据接收

数据接收是在发送地址段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

表 13-44：数据接收

简易I ² C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0的通道0	SCI0的通道1	SCI0的通道2	SCI0的通道3	SCI1的通道0	SCI1的通道1
使用的引脚	SCLKOI00、SDI00 ^{注1}	SCLKOI01、SDI01 ^{注1}	SCLKOI10、SDI10 ^{注1}	SCLKOI11、SDI11 ^{注1}	SCLKOI20、SDI20	SCLKOI21、SDI21
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	8位					
传送速率 ^{注2}	Max.FMCK/4[Hz]（SDRmn[15:9]≥1）FMCK：对象通道的运行时钟频率但是，必须在I ² C的各模式中满足以下条件： Max.1MHz（增强型快速模式） Max.400KHz（快速模式） Max.100KHz（标准模式）					
数据电平	正相输出（默认值：高电平）。					
奇偶校验位	无奇偶校验位。					
停止位	附加1位（ACK发送）。					
数据方向	MSB优先					

注 1：要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出模式（POMxx=1）。

详细内容请参照“2.2 端口复用功能”和“2.3.2 输出输入复用功能寄存器映射”。

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

备注：m：单元号（m=0、1）n：通道号（n=0、1、2、3）

(1) 寄存器的设定

图 13-101: 简易 I2C (IIC00、IIC01、IIC10、IIC11、IIC20、IIC21) 数据接收时的寄存器设定内容例子

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0 注 1	0	0 注 1	1	0	0	1	0	0

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLsmn1	DLsmn0
	0	1	0	0	0	0	0	0	0	0	0 注 2	1	0	1	1 注 3	1

(c) 串行数据寄存器mn(SDRmn) (低8位: SIO_r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDRmn	波特率的设定注 4								0	虚拟发送数据的设定 (FFH)							
SIO_r																	

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1 注 5	0/1 注 5	0	0	0	0	0	0	0/1 注 5	0/1 注 5

(e) 串行输出允许寄存器m(SOEm) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(f) 串行通道开始寄存器m(SSm) 在数据发送和接收的过程中不操作此寄存器。

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

注 1: 只限于 SMR01、SMR03、SMR11 寄存器。

注 2: 只限于 SCR00、SCR02、SCR10 寄存器。

注 3: 只限于 SCR00、SCR01 寄存器, 其他固定为“1”。

注 4: 因为已经在发送地址段时设定, 所以不需要设定。

注 5: 在通信运行过程中, 值因通信数据而变。

备注: m: 单元号 (m=0、1) n: 通道号 (n=0、1、2、3) r: IIC 号 (r=00、01、02、03、10、11)

□: 在 IIC 模式中为固定设定;

■: 不能设定 (设定初始值);

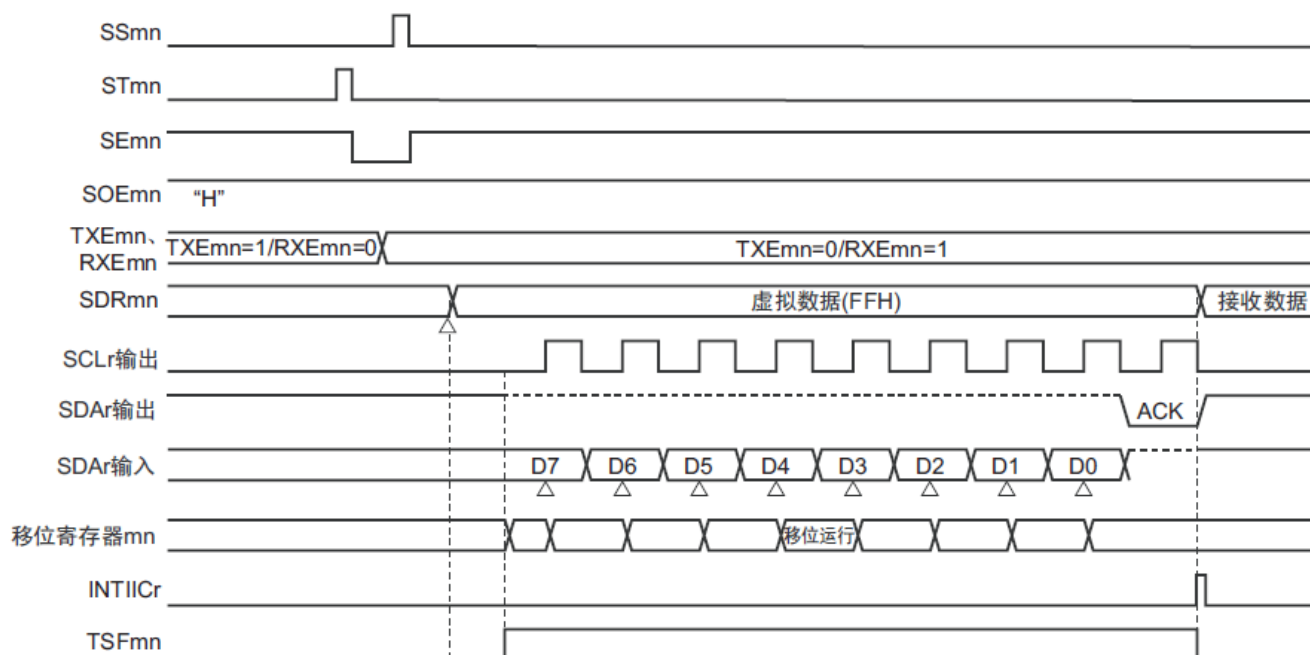
×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

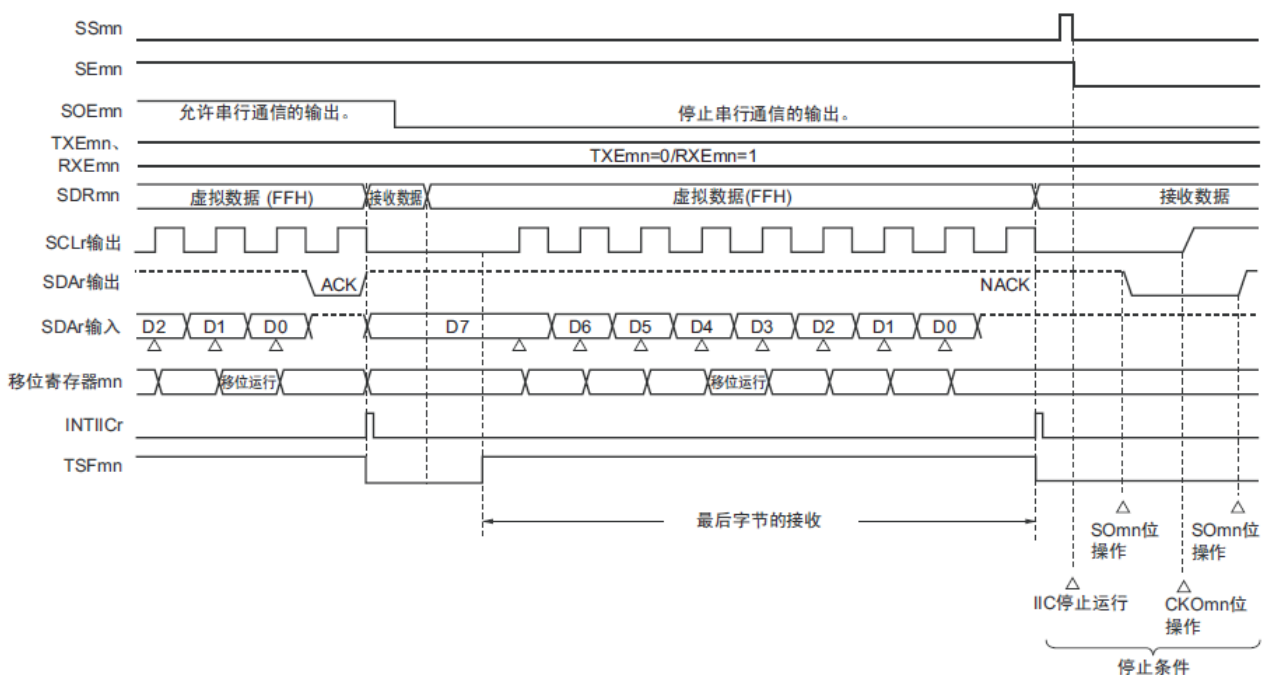
(2) 处理流程

图 13-102：数据接收的时序图

开始接收数据的情况

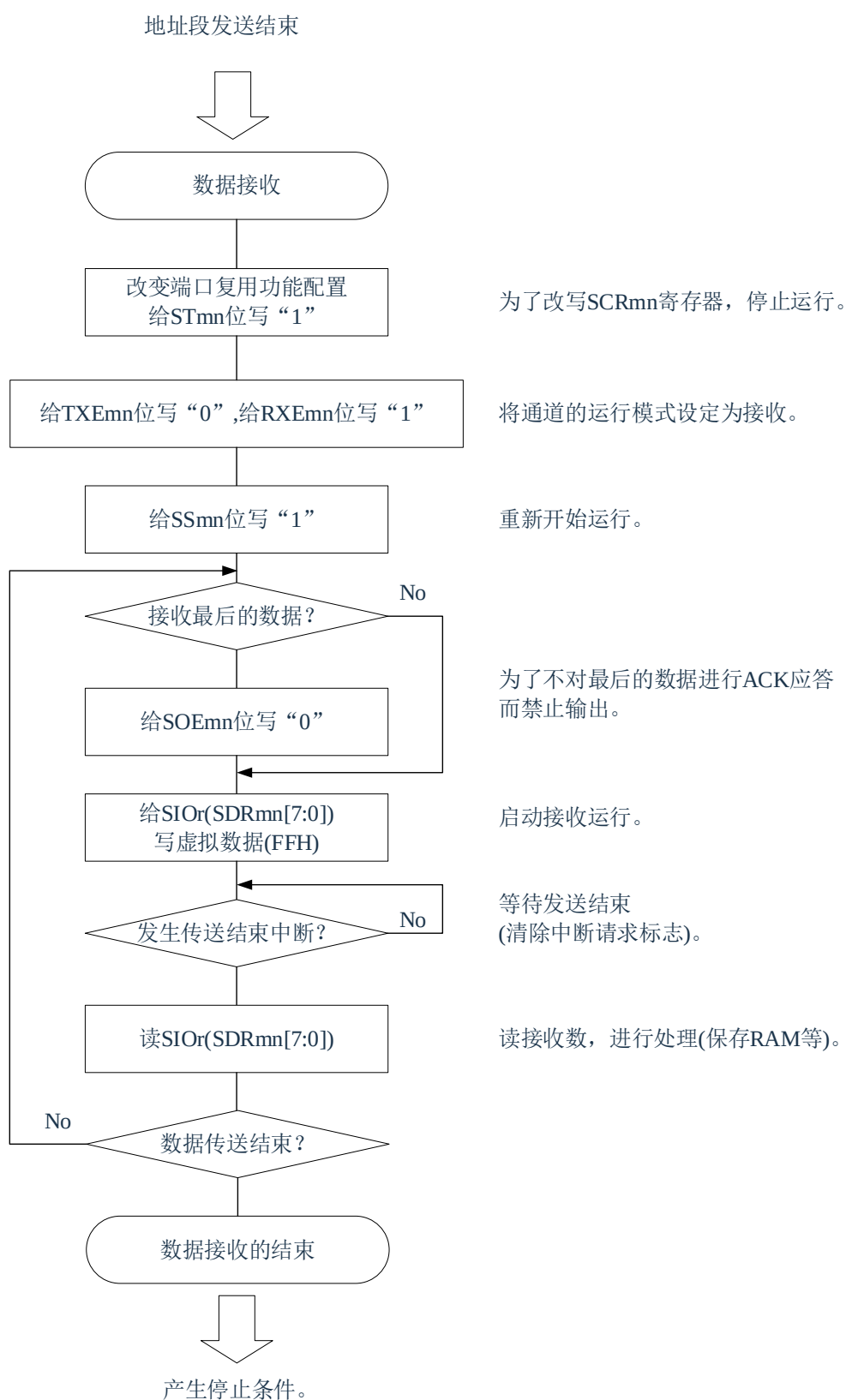


接收最后数据的情况



备注：m：单元号（m=0、1）n：通道号（n=0、1、2、3）r：IIC号（r=00、01、02、03、10、11）

图 13-103: 数据接收的流程图



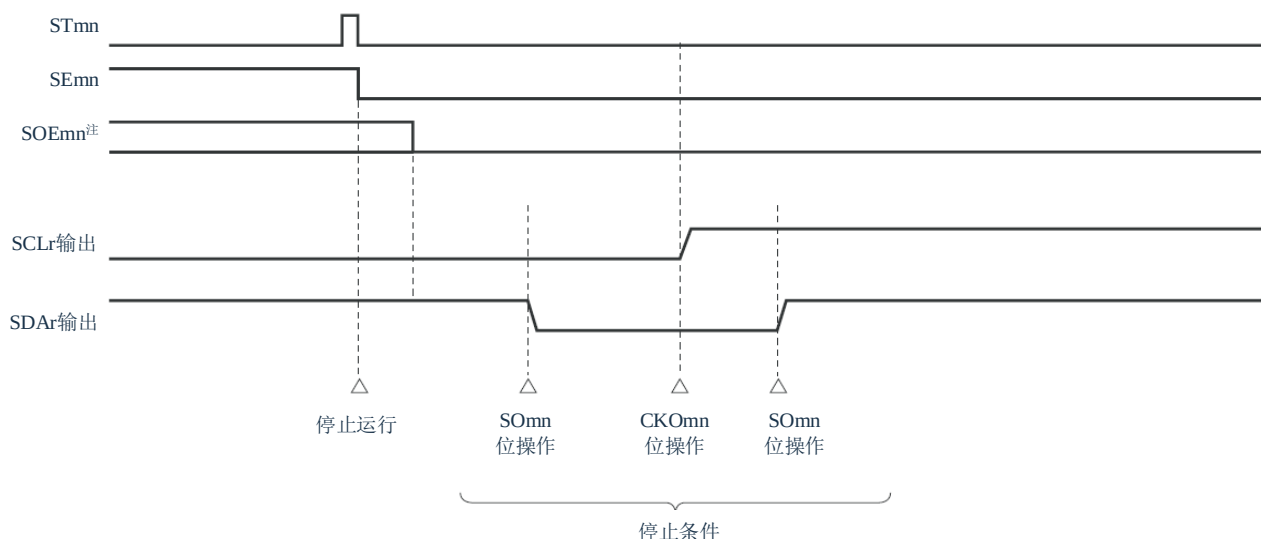
备注：在接收最后的数据时不输出 ACK（NACK）。此后，先通过将串行通道停止寄存器 m（STm）的 STmn 位置“1”来停止运行，然后产生停止条件来结束通信。

13.11.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。

处理流程

图 13-104：产生停止条件的时序图



注：在接收时，已经在接收最后的数据前将串行输出允许寄存器 m（SOEm）的 SOEmn 位置“0”。

图 13-105：产生停止条件的流程图



13.11.5 传送速率的计算

简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的传送速率能用以下计算式进行计算。

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

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）。

注意：

禁止将 SDRmn[15:9] 设定为“0000000B”，SDRmn[15:9] 的设定值必须大于等于“0000001B”。简易 I²C 输出的 SCL 信号的占空比为 50%。在 I²C 总线规格中，SCL 信号的低电平宽度大于高电平宽度。因此，如果设定为快速模式的 400kbps 或者增强型快速模式的 1Mbps，SCL 信号输出的低电平宽度就小于 I²C 总线的规格值。必须给 SDRmn[15:9] 设定能满足 I²C 总线规格的值。

因为 SDRmn[15:9] 的值为串行数据寄存器（SDRmn）的 bit15~9 的值（0000001B~1111111B），所以为 1~127。

备注：m：单元号（m=0、1）n：通道号（n=0、1、2、3）

表 13-45: 简易 I2C 运行时钟的选择

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

注 1: 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: X: 忽略

m: 单元号 (m=0、1); n: 通道号 (n=0、1、2、3)。

F_{MCK}=F_{CLK}=32MHz 时的 I²C 传送速率的设定例子如下所示:

表 13-46: I²C 传输速率的设定

I ² C 传送模式 (期待的传送速率)	F _{CLK} =32MHz			
	运行时钟 (F _{MCK})	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100KHz	F _{CLK} /2	79	100KHz	0.0%
400KHz	F _{CLK}	41	380KHz	5.0% ^注
1MHz	F _{CLK}	18	0.84MHz	16.0% ^注

注: 因为 SCL 信号的占空比为 50%, 所以不能将误差设定为“0”%左右。

13.11.5.1 在简易I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信过程中发生错误时的处理步骤

在简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信过程中发生错误时的处理步骤如表 13-47 和表 13-48 所示。

表 13-47：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。

表 13-48：在简易 I2C 模式中发生 ACK 错误时的处理步骤

软件操作	硬件状态	备注
读串行状态寄存器mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器mn（SDIRmn）。 →	清除错误标志。	通过将SSRmn寄存器的读取值直接写到SDIRmn寄存器，只能清除读操作时的错误。
将串行通道停止寄存器m（STm）的STmn位置“1”。	串行通道允许状态寄存器m（SEm）的SEmn位为“0”并且通道n为运行停止状态。	因为没有返回 ACK，所以从属设备没有做好接收的准备。因此，生成停止条件并且释放总线，再次从开始条件开始通信，或者也能生成重新开始条件并且从地址发送开始重新进行。
生成停止条件。		
生成开始条件。		
将串行通道开始寄存器m（SSm）的SSmn位置“1”。	串行通道允许状态寄存器m（SEm）的SEmn位为“1”并且通道n为可运行状态。	

备注： m：单元号（m=0、1） n：通道号（n=0、1、2、3） r：IIC号（r=00、01、02、03、10、1

第14章 串行接口IICA (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。

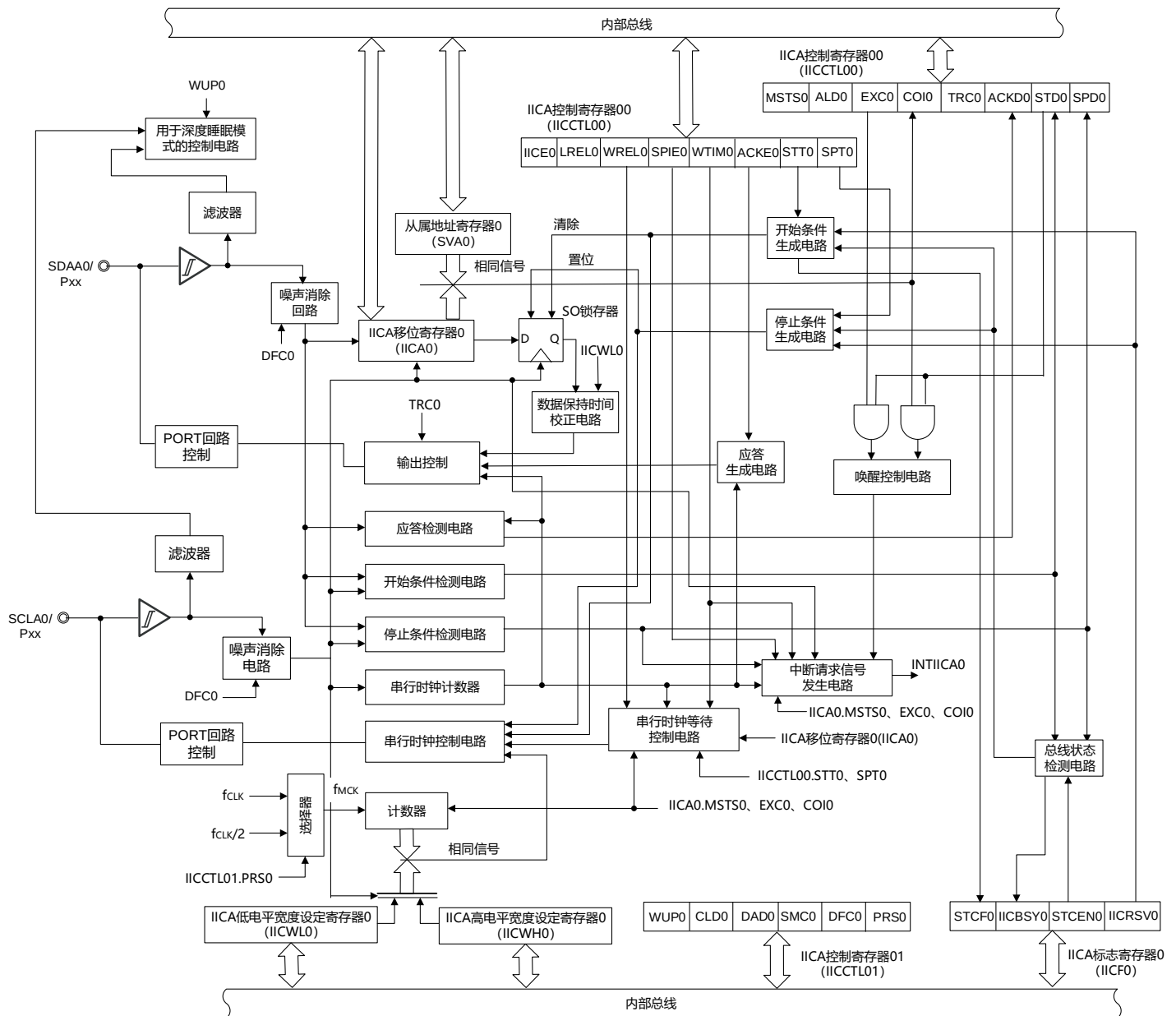
(3) 唤醒模式

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位进行设定。

串行接口IICA的框图如图14-1所示。

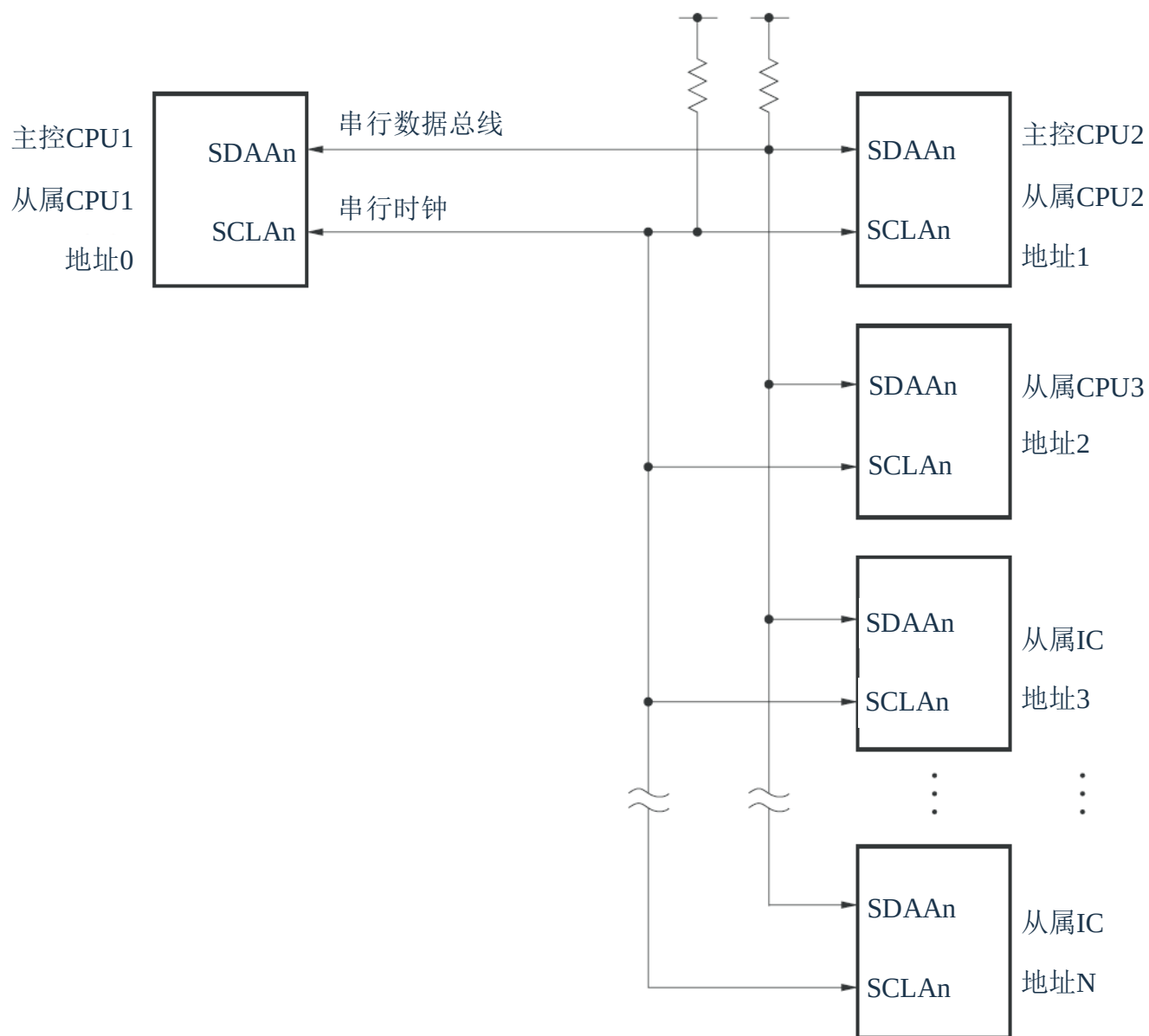
注：n=0。

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



注：n=0。

14.2 串行接口IICA的结构

串行接口IICA由以下硬件构成。

表14-1 串行接口IICA的结构

项目	结构
寄存器	IICA移位寄存器n（IICAn） 从属地址寄存器n（SVAn）
控制寄存器	外围允许寄存器0（PER0） IICA控制寄存器n0（IICCTLn0） IICA状态寄存器n（IICSn） IICA标志寄存器n（IICFn） IICA控制寄存器n1（IICCTLn1） IICA低电平宽度设定寄存器n（IICWLn） IICA高电平宽度设定寄存器n（IICWHn） 端口模式寄存器（PMxx） 端口模式控制寄存器（PMCxx） 端口复用功能配置寄存器（PS_xx）

注1：n=0。

注2：本产品可将IICA输入/输出引脚功能复用到多个端口。当某端口被配置为IICA引脚的复用功能后，该端口的N沟道漏极开路输出（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。

(1) IICA移位寄存器n（IICAn）

IICAn寄存器是与串行时钟同步进行8位串行数据和8位并行数据相互转换的寄存器，用于发送和接收。能通过读写IICAn寄存器来控制实际的发送和接收。

在等待期间，通过写IICAn寄存器来解除等待，开始传送数据。通过8位存储器操作指令去设定IICAn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

表14-2 IICAn移位寄存器n（IICAn）的格式

位	符号	描述	复位值
7:0	IICAn	IICAn移位寄存器n	0x0

注1：在数据传送过程中，不能给IICAn寄存器写数据。

注2：只能在等待期间读写IICAn寄存器。除了等待期间以外，禁止在通信状态下存取IICAn寄存器。但是在主控设备的情况下，能在将通信触发位（STTn）置“1”后写一次IICAn寄存器。

注3：当预约通信时，必须在检测到由停止条件产生的中断后给IICAn寄存器写数据。

注4：n=0。

(2) 从属地址寄存器n (SVAn)

这是在用作从属设备时保存7位本地站地址{A6,A5,A4,A3,A2,A1,A0}的寄存器。

通过8位存储器操作指令设定SVAn寄存器。但是，在STDn位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

表14-3 从属地址寄存器n (SVAn) 的格式

位	符号	描述	复位值
7:0	SVAn	从属地址寄存器n	0x0

注：bit0固定为“0”。

(3) SO锁存器

SO锁存器保持SDAAn引脚的输出电平。

(4) 唤醒控制电路

当设定在从属地址寄存器n (SVAn) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIICAn)。

(5) 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了8位数据的发送和接收。

(6) 中断请求信号发生电路

此电路控制产生中断请求信号 (INTIICAn)。由以下2种触发产生I2C中断请求。

- 第8个或者第9个串行时钟的下降（通过WTIMn位进行设定）
- 因检测到停止条件而产生中断请求（通过SPIEn位进行设定）。

注1：WTIMn位：IICA控制寄存器n0 (IICCTLn0) 的bit3

注2：SPIEn位：IICA控制寄存器n0 (IICCTLn0) 的bit4

(7) 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到SCLAn引脚的时钟。

(8) 串行时钟等待控制电路

此电路控制等待时序。

(9) 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

(10) 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

(11) 开始条件生成电路

如果将STTn位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下（IICRSVn位=1）并且没有释放总线（IICBSYn位=1）时，忽视开始条件请求并且将STCFn位置“1”。

(12) 停止条件生成电路

如果将SPTn位置“1”，此电路就生成停止条件。

(13) 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。但是，在刚运行时不能立即检测总线状态，因此必须通过STCENn位设定总线状态检测电路的初始状态。

注：TTn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。

n=0。

14.3 寄存器映射

(以下寄存器基地址 = 0x4004_1A30) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
IICCTLn0	0x000	R/W	IICA控制寄存器n0	0x00
IICCTLn1	0x001	R/W	IICA控制寄存器n1	0x00
IICWLn	0x002	R/W	IICA低电平宽度设定寄存器n	0xFF
IICWHn	0x003	R/W	IICA高电平宽度设定寄存器n	0xFF
SVA0	0x004	R/W	IICA从属地址寄存器	0x00
IICA0	0x120	R/W	IICA移位寄存器	0x00
IICSn	0x121	R	IICA状态寄存器n	0x00
IICFn	0x122	R/W	IICA标志寄存器n	0x00

注: n=0。

(以下寄存器基地址 = 0x4002_0400) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER0	0x020	R/W	外围允许寄存器0	0x00

14.3.1 外围允许寄存器0（PER0）

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

要使用串行接口IICAn时，必须将bit4（IICAEN）置“1”。

通过8位存储器操作指令设定PER0寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	RTCEN	提供实时时钟（RTC）和15位间隔定时器输入时钟的控制 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。 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。	0
6	-	保留	0
5	ADCEN	提供ADC模块的输入时钟的控制 0: 停止提供输入时钟，ADC不能运行 1: 提供输入时钟，ADC能运行	0
4	IICAEN	提供IICA模块的输入时钟的控制 0: 停止提供输入时钟，IICA不能运行 1: 提供输入时钟，IICA能运行	0
3	SAU1EN	提供通用串行通信单元模块SAU1输入时钟的控制 0: 停止提供输入时钟，SAU1不能运行 1: 提供输入时钟，SAU1能运行	0
2	SAU0EN	提供通用串行通信单元模块SAU0输入时钟的控制 0: 停止提供输入时钟，SAU0不能运行 1: 提供输入时钟，SAU0能运行	0
1	TIMER41EN	提供通用定时器单元模块TIMER41输入时钟的控制 0: 停止提供输入时钟，TIMER41不能运行 1: 提供输入时钟，TIMER41能运行	0
0	TIMER40EN	提供通用定时器单元模块TIMER40输入时钟的控制 0: 停止提供输入时钟，TIMER40不能运行 1: 提供输入时钟，TIMER40能运行	0

注1：设定串行接口IICA时，必须先在IICAEN位为“1”的状态下设定以下的寄存器。当IICAEN位为“0”时，串行接口IICA的控制寄存器的值为初始值，忽视写操作（端口复用功能配置寄存器（PS_{xx}）、端口模式寄存器（PM_{xx}）和端口模式控制寄存器（PMC_{xx}）除外）。

- (1) IICA控制寄存器n0（IICCTLn0）
- (2) IICA标志寄存器n（IICFn）
- (3) IICA状态寄存器n（IICSn）
- (4) IICA控制寄存器n1（IICCTLn1）
- (5) IICA低电平宽度设定寄存器n（IICWLn）
- (6) IICA高电平宽度设定寄存器n（IICWHn）

注2：n=0。

14.3.2 IICA控制寄存器n0（IICCTLn0）

这是允许或者停止I²C运行、设定等待时序以及设定其他I²C运行的寄存器。

通过8位存储器操作指令设定IICCTLn0寄存器。但是必须在IICEn位为“0”时或者在等待期间设定SPIEn位、WTIMn位和ACKEn位，而且在将IICEn位从“0”置为“1”时能同时设定这些位。

在产生复位信号后，此寄存器的值变为“00H”。

注：n=0。

位	符号	描述	复位值
7	IICEn	I²C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 注意：必须在SCLAn线和SDAAn线为高电平的状态下将此位置“1”。 置位条件（IICEn=1）： - 通过指令置位。 清除条件（IICEn=0）： - 通过指令清除。 - 当复位时。	0
6	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”： - ST - SPTn - MSTSn - EXCn - COIn - TRCn - ACKDn - STDn 变为退出通信的待机状态，保持到满足以下的通信参加条件为止。 - 在检测到停止条件后作为主控设备启动。 - 在检测到开始条件后地址匹配或者接收到扩展码。 置位条件（LRELn=1）： - 通过指令置位。 清除条件（LRELn=0）： - 在执行后自动清除。 - 当复位时	0
5	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）。 置位条件（WRELn=1）： - 通过指令置位。 清除条件（WRELn=0）： - 在执行后自动清除。 - 当复位时	0
4	SPIEn ^{注4}	允许或者禁止停止条件检测产生的中断请求 0: 禁止	0

		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。 置位条件 (SPIEn=1): - 通过指令置位。 清除条件 (SPIEn=0): - 在执行后自动清除。 - 当复位时	
3	WTIMn ^{注4}	等待和中断请求的控制 0: 在第8个时钟的下降沿产生中断请求信号。 主控设备: 在输出8个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入8个时钟后, 将时钟置为低电平, 然后等待主控设备。 1: 在第9个时钟的下降沿产生中断请求信号。 主控设备: 在输出9个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入9个时钟后, 将时钟置为低电平, 然后等待主控设备。 在地址传送期间, 与此位的设定无关, 在第9个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。主控设备在地址传送期间的第9个时钟下降沿进入等待状态。接收到本地站地址的从属设备在产生应答 (ACK) 后的第9个时钟下降沿进入等待状态, 但是接收到扩展码的从属设备在第8个时钟下降沿进入等待状态。 置位条件 (WTIMn=1): - 通过指令置位。 清除条件 (WTIMn=0): - 通过指令清除。 - 当复位时	0
2	ACKEn ^{注4-5}	应答控制 0: 禁止应答。 1: 允许应答。在第9个时钟期间将SDAA_n线置为低电平。 置位条件 (ACKEn=1): - 通过指令置位。 清除条件 (ACKEn=0): - 通过指令清除。 - 当复位时	0
1	STTn ^{注2-6}	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, IICBSY_n位为“0”): 如果将此位置“1”, 就生成开始条件 (作为主控设备的启动)。 当第三方正在通信时: - 允许通信预约功能的情况 (IICRSV_n=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 (IICRSV_n=1): 即使将此位置“1”, 也清除STT_n位并且将STT_n清除标志 (STCF_n) 置“1”, 不生成开始条件。 - 等待状态 (主控设备): 在解除等待后生成重新开始条件。 有关置位时序的注意事项: - 主控接收: 禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 - 主控发送: 在应答期间, 可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。 - 禁止与停止条件的触发 (SPT_n) 同时置“1”。 - 在将STT_n位置“1”后, 禁止在满足清除条件前再次将此位“1”。 置位条件 (STTn=1):	0

		- 通过指令置位。 清除条件 (STTn=0): - 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	
0	SPTn ^{注7}	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: - 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”。 置位条件 (SPTn=1): - 通过指令置位。 清除条件 (SPTn=0): - 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	0

注1: 对 IICA 移位寄存器 n (IICAn)、IICA 标志寄存器 n (IICFn) 的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1 (IICCTLn1) 的 CLDn 位和 DADn 位进行复位。

注2: 在 IICEn 位为“0”的状态下, 此位的信号无效。

注3: LRELn 位和 WRELn 位的读取值总是“0”。

注4: 在 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。

注5: 在地址传送过程中不是扩展码时, 设定值无效。当为从属设备并且地址匹配时, 与设定值无关而生成应答。

注6: STTn 位的读取值总是“0”。

注7: SPTn 位的读取值总是“0”。

注8: 如果在 SCLAn 线为高电平、SDAAAn 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”。

注9: n=0。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 SDAAAn 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。

注10: n=0

14.3.3 IICA状态寄存器n (IICSn)

这是表示I²C状态的寄存器。

只有在STTn位为“1”并且等待期间，才能8位存储器操作指令读IICSn寄存器。在产生复位信号后，此寄存器的值变为“00H”。

注1：在深度睡眠模式中允许地址匹配唤醒功能（WUPn=1）状态下，禁止读 IICSn 寄存器。在 WUPn 位为“1”的状态下，与 INTIICAn 中断请求无关，如果将 WUPn 位从“1”改为“0”（停止唤醒运行），就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许（SPIEn=1）因检测到停止条件而产生的中断，并且在检测到中断后读 IICSn 寄存器。

注2：STTn：IICA 控制寄存器 n0 (IICCTLn0) 的 bit1, n=0。

注3：WUPn：IICA 控制寄存器 n1 (IICCTLn1) 的 bit7, n=0。

位	符号	描述	复位值
7	MSTS _n	主控状态的确认标志 0: 从属状态或者通信待机状态。 1: 主控通信状态。 置位条件 (MSTS _n =1): • 当生成开始条件时 清除条件 (MSTS _n =0): • 当检测到停止条件时 • 当ALD _n 位为“1”（仲裁失败）时 • 因LREL _n 位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	0
6	ALD _n	仲裁失败的检测 0: 表示未发生仲裁或者赢得仲裁。 1: 表示仲裁失败。清除MSTS _n 位。 置位条件 (ALD _n =1): • 当仲裁失败时 清除条件 (ALD _n =0) • 在读IICS _n 寄存器后自动清除 ^{注1} 。 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	0
5	EXC _n	扩展码的接收检测 0: 未接收到扩展码。 1: 接收到扩展码。 置位条件 (EXC _n =1): • 当接收的地址数据的高4位为“0000”或者“1111”时 （在第8个时钟的上升沿置位） 清除条件 (EXC _n =0): • 当检测到开始条件时 • 当检测到停止条件时 • 因LREL _n 位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	0
4	COIn	地址匹配的检测 0: 地址不同。 1: 地址相同。 置位条件 (COIn=1): • 当接收地址和本地站地址（从属地址寄存器n (SVAn)）相同时（在第8个时钟的上升沿置位） 清除条件 (COIn=0): • 当检测到开始条件时 • 当检测到停止条件时	0

		- 因LRELn位为“1”（退出通信）而进行的清除 - 当IICEEn位从“1”变为“0”（停止运行）时 - 当复位时	
3	TRCn	发送/接收的状态检测 0: 处于接收状态（发送状态除外）。将SDAAn线置为高阻抗。 1: 处于发送状态。设定为能将SOn锁存器的值输出到SDAAn线（在第1字节的第9个时钟的下降沿以后有效）。 置位条件（TRCn=1）： <主控设备> - 当生成开始条件时 - 当第1字节（地址传送）的LSB（传送方向指示位）输出“0”（主控发送）时 <从属设备> - 当主控设备的第1字节（地址传送）的LSB（传送方向指示位）输入“1”（从属发送）时 清除条件（TRCn=0）： <主控设备和从属设备> - 当检测到停止条件时 - 因LRELn位为“1”（退出通信）而进行的清除 - 当IICEEn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2 - 当ALDn位从“0”变为“1”（仲裁失败）时 - 当复位时 <主控设备> - 当第1字节的LSB（传送方向指示位）输出“1”时 <从属设备> - 当检测到开始条件时 - 当第1字节的LSB（传送方向指示位）输入“0”时	0
2	ACKDn	应答（ACK）的检测 0: 未检测到应答。 1: 检测到应答。 置位条件（ACKDn=1）： - 在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时 清除条件（ACKDn=0）： - 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IICEEn位从“1”变为“0”（停止运行）时 - 当复位时	0
1	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。 置位条件（STDn=1）： - 当检测到开始条件时 清除条件（STDn=0）： - 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IICEEn位从“1”变为“0”（停止运行）时 - 当复位时	0
0	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。 置位条件（SPDn=1）： 当检测到停止条件时 清除条件（SPDn=0）：	0

		- 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	

- 注1：即使对 IICS_n 寄存器以外的位，执行位存储器操作指令，也清除此位。因此，在使用 ALD_n 位时，必须在读其他位前先读 ALD_n 位的数据。
- 注2：在 IICA 状态寄存器 n（IICS_n）的 bit3（TRC_n）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（WREL_n）置“1”来解除等待，就在清除 TRC_n 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 TRC_n 位为“1”（发送状态）时的等待解除。
- 注3：LREL_n：IICA 控制寄存器 n0（IICCTLn0）的 bit6。
- 注4：IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7。
- 注5：n=0。

14.3.4 IICA标志寄存器n (IICFn)

这是设定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定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位设定允许或者禁止通信预约功能，并且通过STCENn位设定IICBSYn位的初始值。只有在禁止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”。

位	符号	描述	复位值
7	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 置位条件 (STCFn=1): • 在设定为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将STTn位清“0”时 清除条件 (STCFn=0): • 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	0
6	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) 置位条件 (IICBSYn=1): • 当检测到开始条件时 • STCENn位为“0”时的IICEn位的置位 清除条件 (IICBSYn=0): • 当检测到停止条件时 • 当IICEn位为“0” (停止运行) 时 • 当复位时	0
5:2	-	保留	-
1	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。 置位条件 (STCENn=1): • 通过指令置位。 清除条件 (STCENn=0): • 通过指令清除。 • 当检测到开始条件时 • 当复位时	0
0	IICRSVn	通信预约功能禁止位 0: 允许通信预约。 1: 禁止通信预约。 置位条件 (IICRSVn=1): • 通过指令置位。 清除条件 (IICRSVn=0): • 通过指令清除。	0

注1: bit6和bit7是只读位。

注2: 只有在停止运行 (IICEn=0) 时才能写 STCENn 位。

注3: 如果 STCENn 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0), 因此为了避免在发行第1个开始条件 (STTn=1) 时破坏其他通信, 需要确认没有正在通信的第三方。

注4: 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

注5: STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1。

注6: IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7。

14.3.5 IICA控制寄存器n1 (IICCTLn1)

这是用于设定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定IICCTLn1寄存器。但是，只能读CLDn位和DADn位。

除了WUPn位以外，必须在禁止I²C运行（IICA控制寄存器n0 (IICCTLn0) 的bit7 (IICEn) =0）时设定IICCTLn1寄存器。

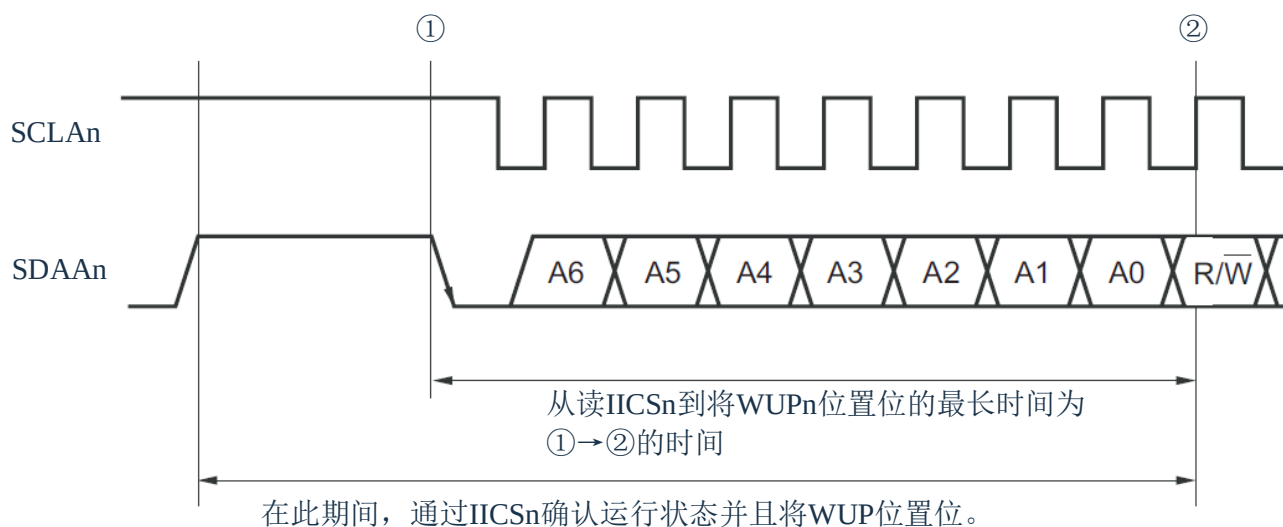
在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	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fMCK时钟，然后执行深度睡眠指令。(参照图14-12 WUPn位置“1”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。 置位条件（WUPn=1）： - 通过指令置位（MSTSn=0、EXCn=0、COIn=0并且STDn=0（不参加通信））^{※2}。 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）。	0
6	-	保留	-
5	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。 置位条件（CLDn=1）： - 当SCLAn引脚为高电平时 清除条件（CLDn=0）： - 当SCLAn引脚为低电平时 - 当IICEn位为“0”（停止运行）时 - 当复位时	0
4	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。 置位条件（DADn=1）： - 当SDAAn引脚为高电平时 清除条件（DADn=0）： - 当SDAAn引脚为低电平时 - 当IICEn位为“0”（停止运行）时 - 当复位时	0
3	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。	0
2	DFCn	数字滤波器的运行控制	0

		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。	
1	-	保留	-
0	PRSn	运行时钟 (f_{MCK}) 的控制 0: 选择 f_{CLK} ($1MHz \leq f_{CLK} \leq 20MHz$)。 1: 选择 $f_{CLK}/2$ ($20MHz < f_{CLK}$)。	0

注1: bit4和 bit5是只读位。

注2: 在以下所示的期间，需要确认 IICA 状态寄存器 n (IICSn) 的状态并且将其置位。



注3: 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”。

注4: 在设定传送时钟的情况下，必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 最小工作频率取决于运行模式。

快速模式: $f_{CLK}=3.5MHz(\text{Min.})$

增强型快速模式: $f_{CLK}=10MHz(\text{Min.})$

标准模式: $f_{CLK}=1MHz(\text{Min.})$

注5: IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

注6: n=0。

14.3.6 IICA低电平宽度设定寄存器n（IICWLn）

此寄存器控制串行接口IICA输出的SCLAn引脚信号低电平宽度（ t_{LOW} ）和SDAAn引脚信号。

通过8位存储器操作指令设定IICWLn寄存器。

必须在禁止I²C运行（IICA控制寄存器n0（IICCTLn0）的bit7（IICEn）=0）时设定IICWLn寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

有关IICWLn寄存器的设定方法，请参照“14.4.2通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法”。数据保持时间为IICWLn所设时间的1/4。

位	符号	描述	复位值
7:0	IICWLn	IICA低电平宽度设定寄存器	0xFF

14.3.7 IICA高电平宽度设定寄存器n（IICWHn）

此寄存器控制串行接口IICA输出的SCLAn引脚信号高电平宽度和SDAAn引脚信号。通过8位存储器操作指令设定IICWHn寄存器。

必须在禁止I²C运行（IICA控制寄存器n0（IICCTLn0）的bit7（IICEn）=0）时设定IICWHn寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

位	符号	描述	复位值
7:0	IICWHn	IICA高电平宽度设定寄存器	0xFF

注1：有关主控方传送时钟的设定方法，请参照14.4.2(1)；有关从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法，请参照14.4.2(2)。

注2：n=0。

14.3.8 控制IICA引脚端口功能的寄存器

本产品可将IICA的引脚功能复用到多个端口。

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”。

通过设置端口复用功能配置寄存器可将SCALn引脚和SDAAn引脚分别配置到两个端口。

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。

SCLAn: 串行时钟的输入/输出引脚

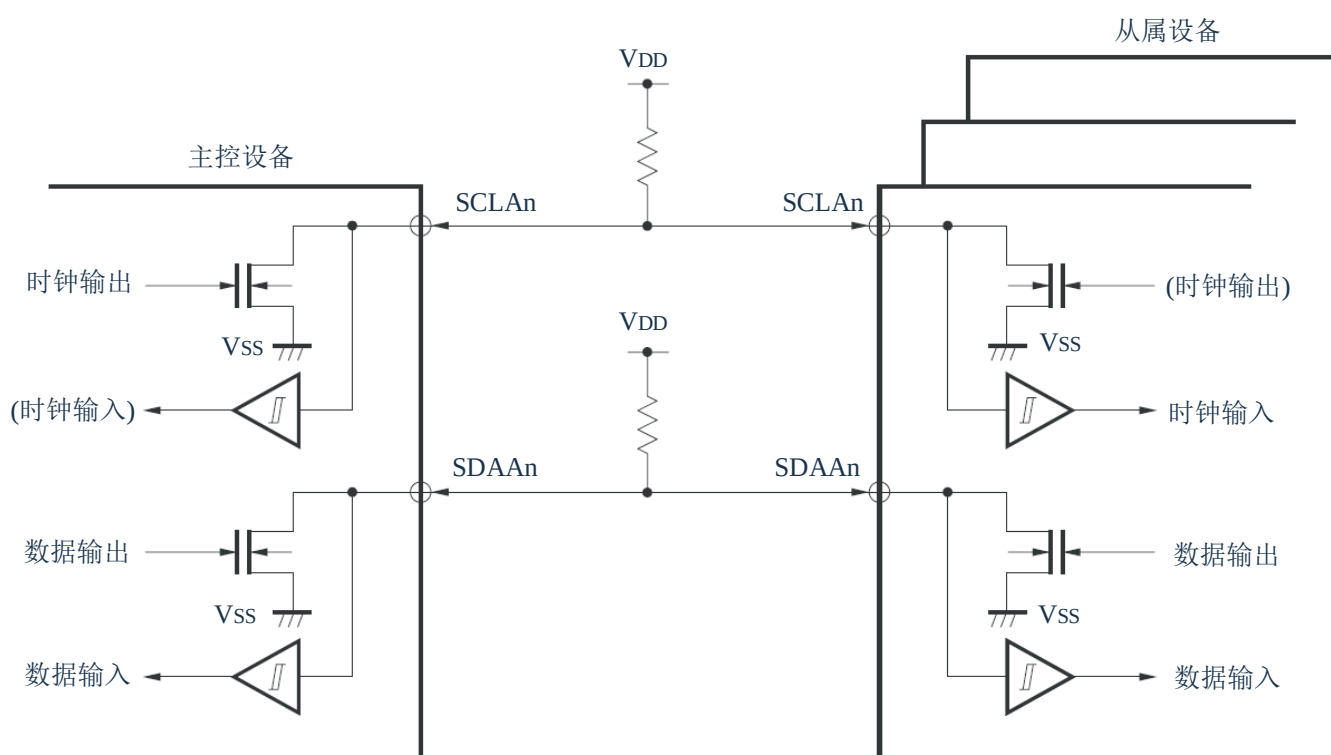
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

SDAAn: 串行数据的输入/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。

因为串行时钟线和串行数据总线的输出为 N 沟道漏极开路输出，所以需要外接上拉电阻。

图14-3 引脚结构图



注：n=0。

14.4.2 通过IICWL_n寄存器和IICWH_n寄存器设定传送时钟的方法

(1) 主控方传送时钟的设定方法

$$\text{传送时钟} = \frac{f_{\text{MCK}}}{\text{IICWL} + \text{IICWH} + f_{\text{MCK}}(t_{\text{R}} + t_{\text{F}})}$$

此时，IICWL_n寄存器和IICWH_n寄存器的最佳设定值如下：

(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWL}_n = \frac{0.52}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.48}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWL}_n = \frac{0.47}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.53}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWL}_n = \frac{0.50}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.50}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

(2) 从属方IICWL_n寄存器和IICWH_n寄存器的设定方法

(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWL}_n = 1.3\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (1.2\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWL}_n = 4.7\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (5.3\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWL}_n = 0.50\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (0.50\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

注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”。

注2：在设定传送时钟的情况下，必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 最小工作频率取决于运行模式。

快速模式： $f_{\text{CLK}} = 3.5\text{MHz}(\text{Min.})$

增强型快速模式： $f_{\text{CLK}} = 10\text{MHz}(\text{Min.})$

标准模式： $f_{\text{CLK}} = 1\text{MHz}(\text{Min.})$

注3：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。

注4：IICWL_n：IICA 低电平宽度设定寄存器 n。

IICWH_n：IICA 高电平宽度设定寄存器 n。

t_{F} ：SDAAn 信号和 SCLAn 信号的下降时间。

t_R : SDAAn 信号和 SCLAn 信号的上升时间。

f_{MCK} : IICA 运行时钟频率

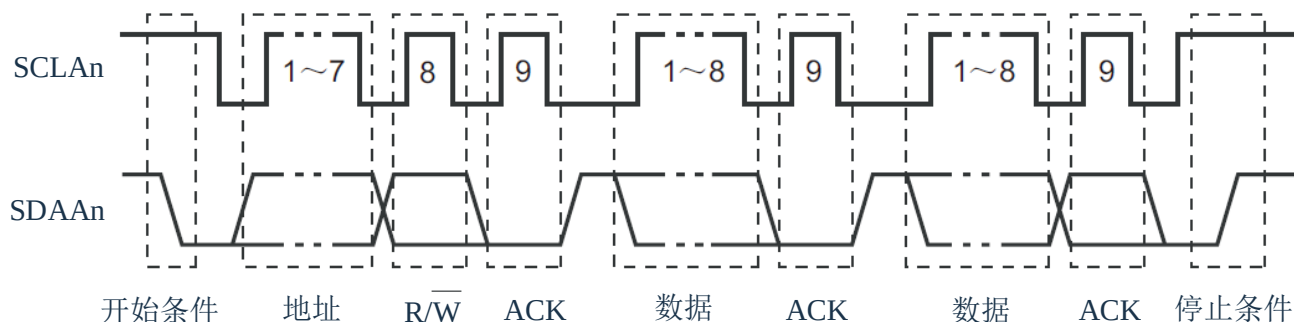
注5: $n=0$ 。

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14-4 I²C总线的串行数据传送时序



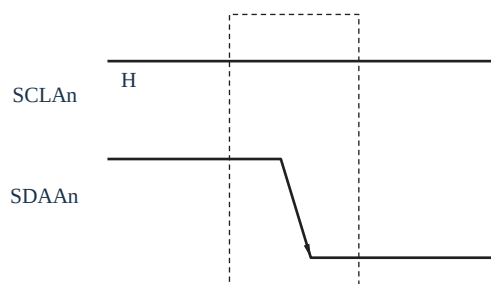
主控设备生成开始条件、从属地址和停止条件。

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14-5 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”。

注：n=0。

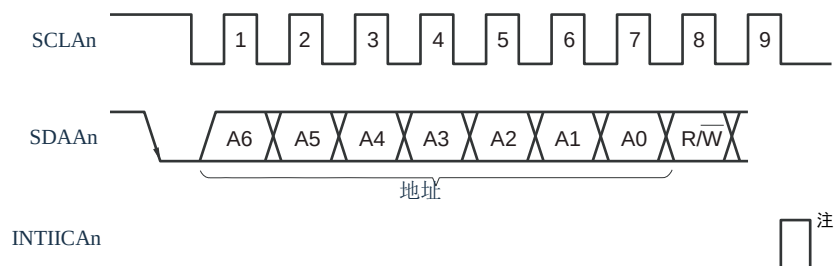
14.5.2 地址

开始条件的后续7位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的7位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查7位数据是否和从属地址寄存器n（SVAn）的内容相同。此时如果7位数据和SVAn寄存器的值相同，该从属设备就被选中，在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14-6 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。

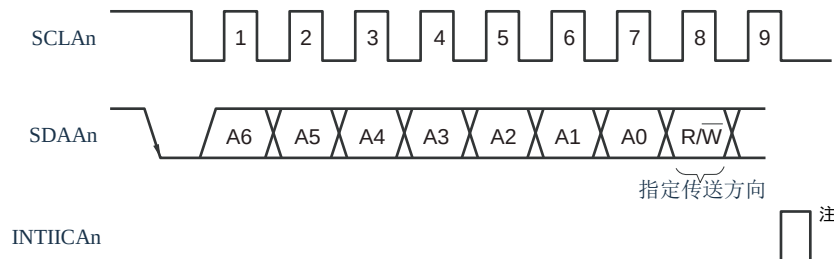
如果将从属地址和“14.5.3 传送方向的指定”中说明的传送方向构成的8位数据写到IICA移位寄存器n（IICAn），就输出地址。接收到的地址被写到IICAn寄存器。从属地址分配在IICAn寄存器的高7位。

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。

当此传送方向指定为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定为“1”时，表示主控设备从从属设备接收数据。

图14-7 传送方向的指定



注1：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生INTIICAn。

注2：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（IICS_n）的bit2（ACKD_n）确认应答的检测。在主机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，主机设备输出停止条件或者重新开始条件，中止发送。不返回应答的原因如下：

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

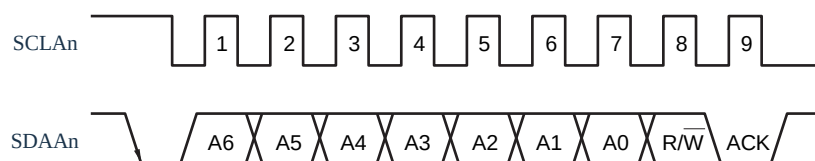
接收方在第9个时钟将SDAAn线置为低电平，生成应答（正常接收）。

通过将IICA控制寄存器n0（IICCTLn0）的bit2（ACKEn）置“1”，变为能自动生成应答的状态。通过7位地址信息后续的第8位数据设定IICS_n寄存器的bit3（TRC_n）。在接收（TRC_n=0）的情况下，通常必须将ACKEn位置“1”。

在从属接收运行过程中（TRC_n=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”，通知主机方不能接收数据。

在主机接收运行过程中（TRC_n=0）不需要下一个数据时，为了不生成应答，必须将ACKEn位清“0”，通知从属发送方数据的结束（停止发送）。

图14-8 应答



当接收到本地站的地址时，与ACKEn位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答（NACK）。

在接收到扩展码时，通过事先将ACKEn位置“1”，生成应答。接收数据时的应答生成方法因等待时序的设定而不同，如下所示。

当选择8个时钟的等待时（IICCTLn0寄存器的bit3（WTIM_n）=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。

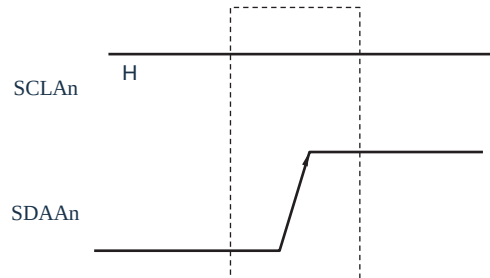
当选择9个时钟等待时（IICCTLn0寄存器的bit3（WTIM_n）=1）：通过事先将ACKEn位置“1”，生成应答。

注：n=0。

14.5.5 停止条件

在SCLAn引脚为高电平时，如果SDAAn引脚从低电平变为高电平，就生成停止条件。停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图14-9 停止条件



如果将IICA控制寄存器n0 (IICCTLn0)的bit0 (SPTn)置“1”，就生成停止条件。如果检测到停止条件，就将IICA状态寄存器n (IICSn)的bit0 (SPDn)置“1”，并且在IICCTLn0寄存器的bit4 (SPIEn)为“1”时产生INTIICAn。

注：n=0。

14.5.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送/接收（等待状态）。

通过将SCLAn引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图14-10 等待(1/2)

(1) 主控设备为9个时钟等待，从属设备为8个时钟等待的情况

（主控设备：发送，从属设备：接收，ACKEn=1）

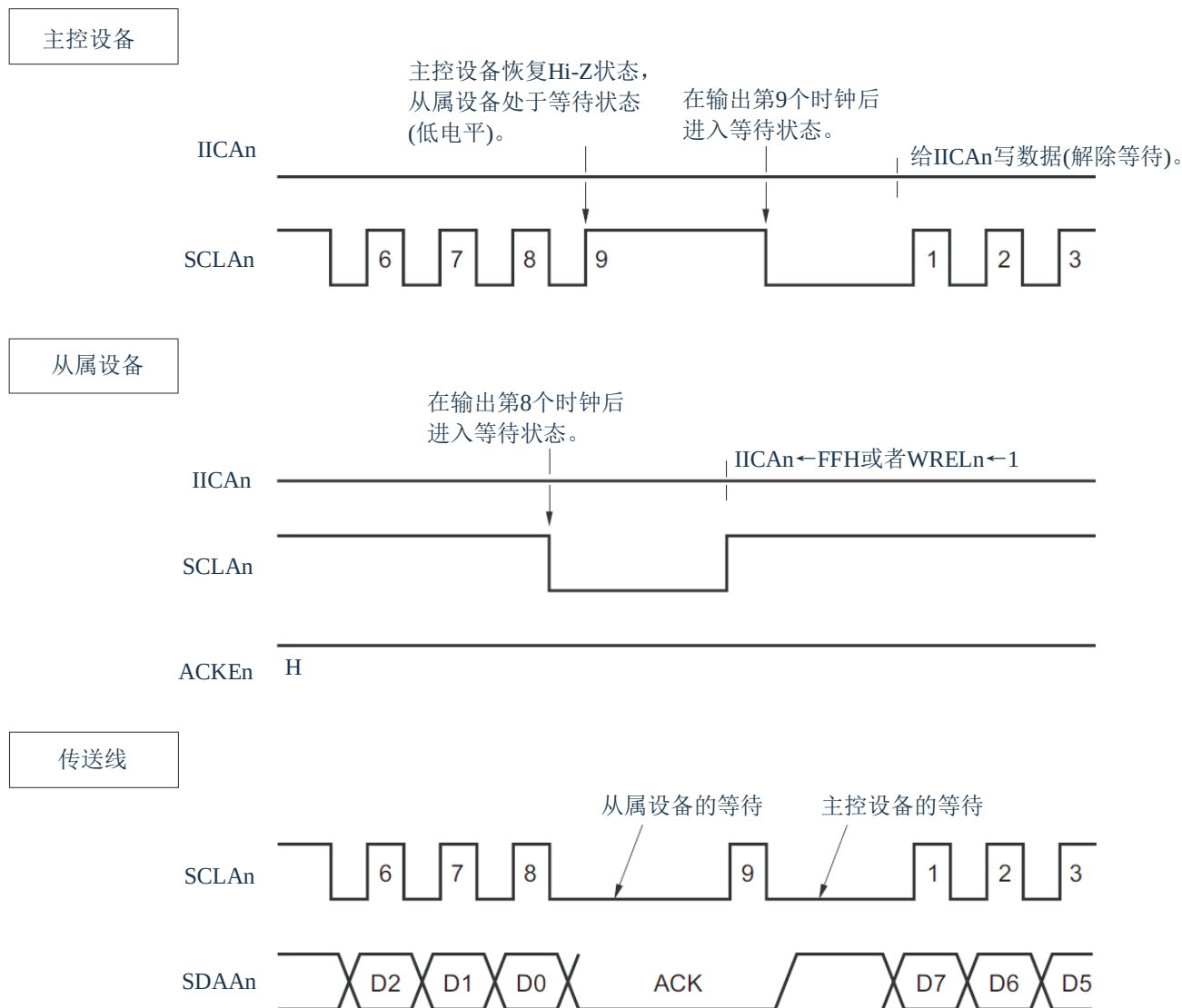
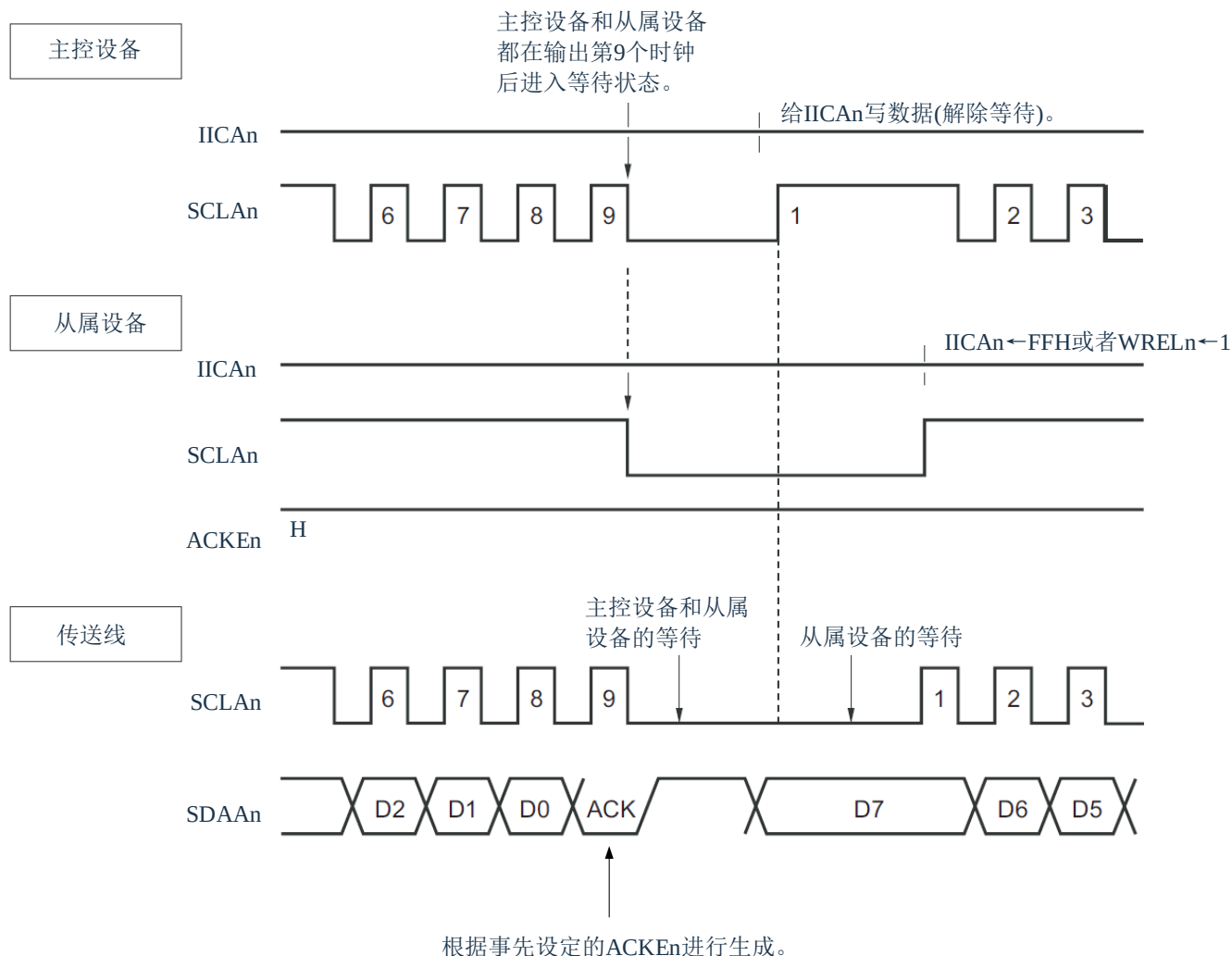


图14-10 等待(2/2)

(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)



注1: ACKEn: IICA控制寄存器n0 (IICCTLn0) 的bit2。

注2: WRELn: IICA控制寄存器n0 (IICCTLn0) 的bit5。

通过设定IICA控制寄存器n0 (IICCTLn0) 的bit3 (WTIMn) 自动产生等待状态。通常，在接收方，如果IICCTLn0寄存器的bit5 (WRELn) 为“1”或者给IICA移位寄存器n (IICAn) 写“FFH”，就解除等待；在发送方，如果给IICAn寄存器写数据，就解除等待。主控设备还能通过以下方法解除等待：

将IICCTLn0寄存器的bit1 (STTn) 置“1”。

将IICCTLn0寄存器的bit0 (SPTn) 置“1”。

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1：如果在WUPn位为“1”时执行等待的解除处理，就不解除等待。

注2：n=0。

14.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设定IICA控制寄存器n0（IICCTLn0）的bit3（WTIMn），在表14-4所示的时序产生INTIICAn并且进行等待控制。

表14-4 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）的设定地址相同时，从属设备才在第9个时钟的下降沿产生 INTIICAn 信号并且进入等待状态。此时，与 IICCTLn0寄存器的 bit2（ACKEn）的设定无关，生成应答。接收到扩展码的从属设备在第8个时钟的下降沿产生 INTIICAn。如果在重新开始后地址不同，就在第9个时钟的下降沿产生 INTIICAn，但是不进入等待状态。

注2：如果接收的地址和从属地址寄存器 n（SVAn）的内容不同并且未接收到扩展码，就不产生 INTIICAn 并且也不进入等待状态。

注3：表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- ① 从属运行：与 WTIMn 位无关，根据上述注1和注2的条件决定中断和等待的时序。
- ② 主控运行：与 WTIMn 位无关，在第9个时钟的下降沿产生中断和等待的时序。

(2) 数据接收

主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

(3) 数据发送

主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

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1：只限于主控设备。

注2：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）的设定地址相同或者只接收到扩展码时，产生INTIICAn中断请求。

14.5.10 错误的检测

在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。

注：n=0。

14.5.11 扩展码

- (1) 当接收地址的高4位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志（EXCn）置“1”，并且在第8个时钟的下降沿产生中断请求（INTIICAn）。
不影响保存在从属地址寄存器n（SVAn）的本地站地址。
- (2) 当SVAn寄存器的设定值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求（INTIICAn）。
 - 高4位数据相同：EXCn=1
 - 7位数据相同：COIn=1

注1：EXCn：IICA状态寄存器n（IICSn）的bit5。

注2：COIn：IICA状态寄存器n（IICSn）的bit4。

- (3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将IICA控制寄存器n0（IICCTLn0）的bit6（LRELn）置“1”，进入下一次通信的待机状态。

表14-5 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）

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。

注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。

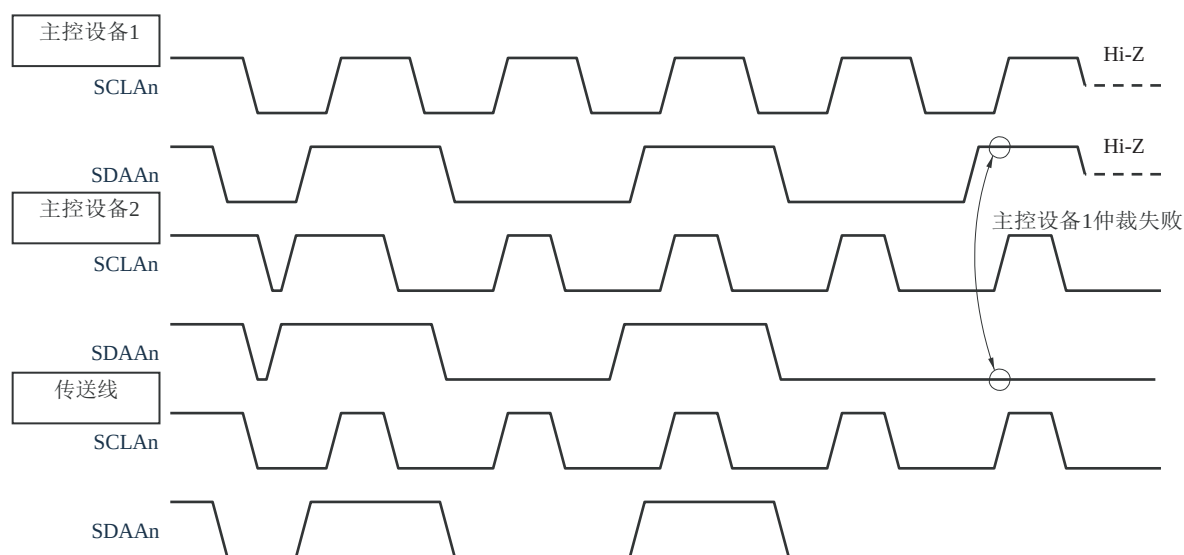
在发生下一次中断请求时（例如：在第8或者第9个时钟检测到停止条件），用软件通过ALDn位为“1”来检测仲裁的失败。

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”。

注1：STDn：IICA状态寄存器n（IICSn）的bit1。

注2：STTn：IICA控制寄存器n0（IICCTLn0）的bit1。

图14-11 仲裁时序例子



注：n=0。

表14-6 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。	在生成停止条件时（SPIEn=1） ^{注2}
在数据传送过程中检测到停止条件。	
想要生成重新开始条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件，但是检测到停止条件。	在生成停止条件时（SPIEn=1） ^{注2}
想要生成停止条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿 ^{注1}
想要生成重新开始条件，但是SCLAn为低电平。	

注1：当WTIMn位（IICA控制寄存器n0（IICCTLn0）的bit3）为“1”时，在第9个时钟的下降沿产生中断请求；当WTIMn位为“0”并且接收到扩展码的从属地址时，在第8个时钟的下降沿产生中断请求。

注2：当有可能发生仲裁时，必须在主控运行时将SPIEn位置“1”。

注3：SPIEn：IICA控制寄存器n0（IICCTLn0）的bit4。

注4：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14-12所示，通过地址匹配将WUPn位置“0”时的流程如图14-13所示。

图14-12 将WUPn位置“1”时的流程

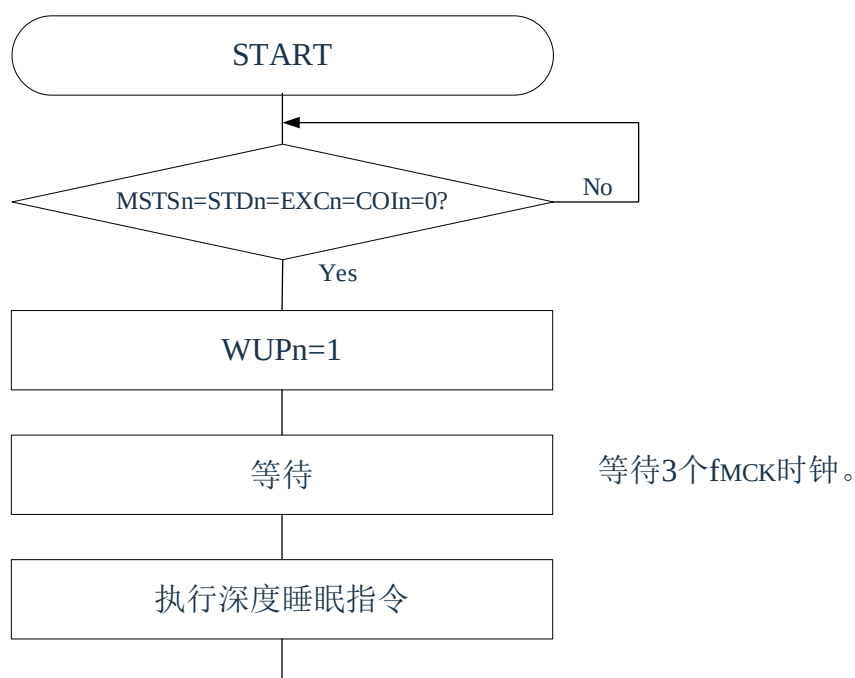
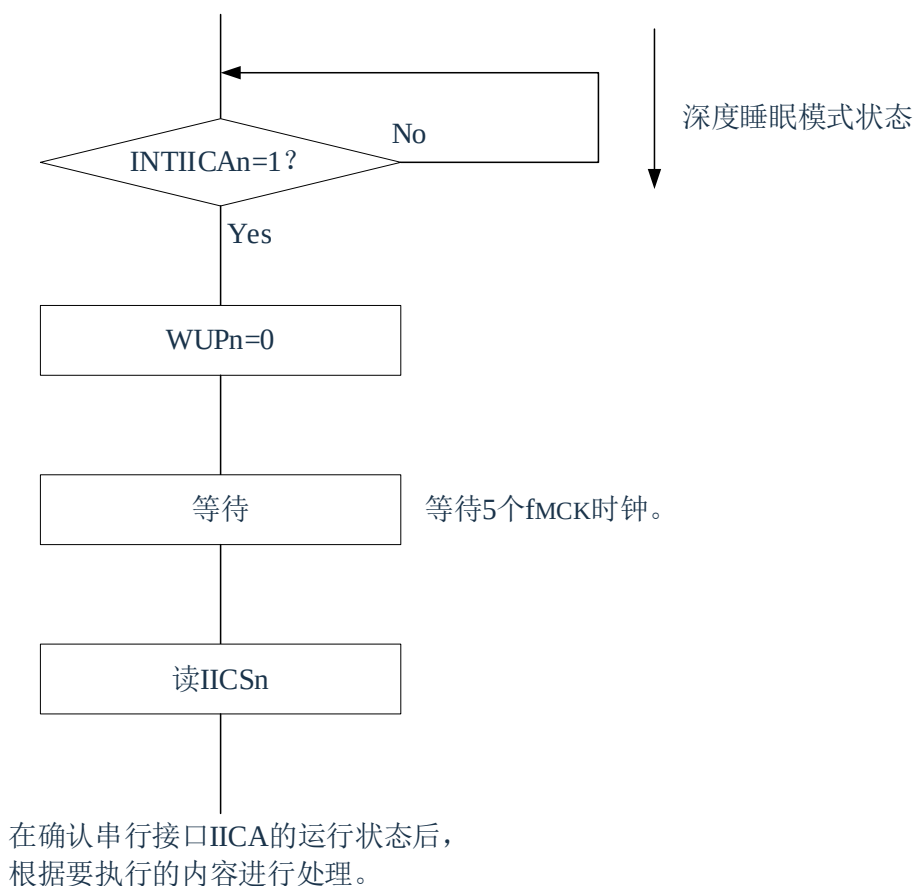


图14-13 通过地址匹配将WUPn位置“0”时的流程（包括接收扩展码）



除了串行接口IICA产生的中断请求（INTIICAn）以外，必须通过以下的流程解除深度睡眠模式。

(1) 下一次IIC通信为主控设备运行的情况：图14-14的流程

(2) 下一次IIC通信为从属设备运行的情况

通过INTIICAn中断返回的情况：和图14-13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”状态继续运行。

注：n=0。

图14-14 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。

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：

a) 在仲裁结果既不是主控设备也不是从属设备时。

b) 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICA 控制寄存器 n0（IICCTLn0）的 bit6（LRELn）置“1”，退出通信后释放了总线）

如果在不加入总线的状态下将IICCTLn0寄存器的bit1（STTn）置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将IICCTLn0寄存器的bit4（SPIEn）置“1”，在通过产生的中断请求信号（INTIICAn）检测到总线的释放（检测到停止条件）后，如果给IICA移位寄存器n（IICAn）写地址，就自动作为主控设备开始通信。在检测到停止条件前，给IICAn寄存器写的的数据无效。

当将 STTn 位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

① 总线处于释放状态时生成开始条件

② 总线未处于释放状态（待机状态）时通信预约

在将STTn位置“1”并且经过等待时间后，通过MSTS_n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MSTS_n 标志为止的等待时间：
(IICWL_n 的设定值+ IICWH_n 的设定值+ 4) / f_{MCK} + t_F * 2

注1：IICWL_n：IICA低电平宽度设定寄存器n。

注2：IICWH_n：IICA高电平宽度设定寄存器n。

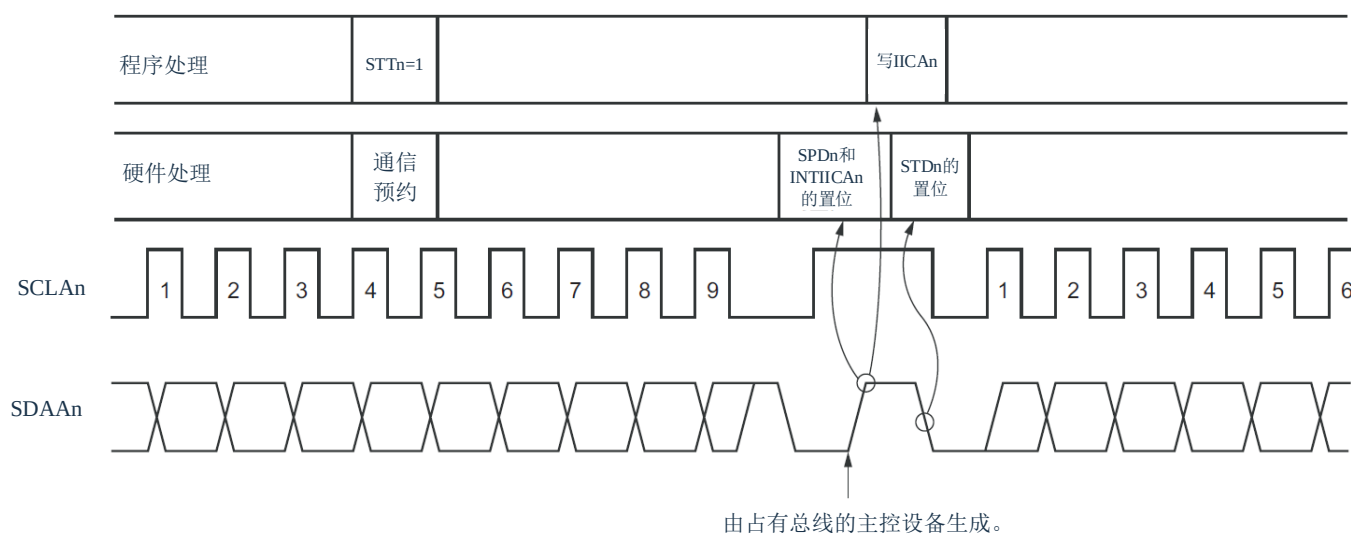
注3：t_F：SDAA_n信号和SCLAn信号的下降时间。

注4：f_{MCK}：IICA运行时钟频率。

注5：n=0

通信预约的时序如下图所示。

图14-15 通信预约的时序



注1: IICAn: IICA移位寄存器n。

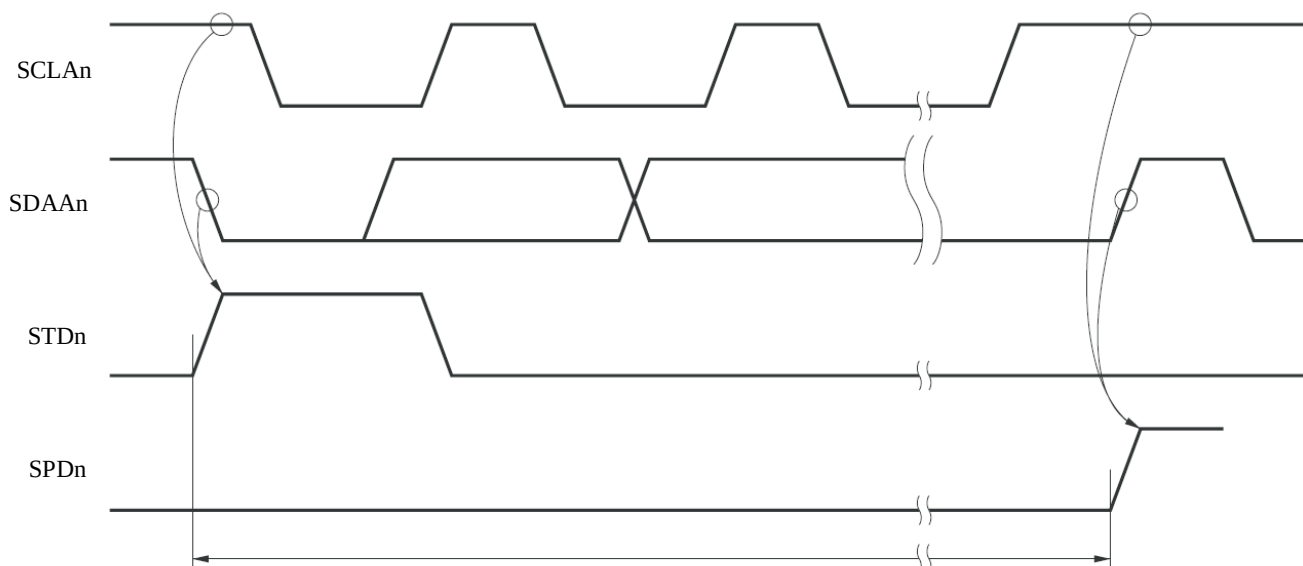
注2: STTn: IICA控制寄存器n0 (IICCTLn0) 的bit1。

注3: STDn: IICA状态寄存器n (IICSn) 的bit1。

注4: SPDn: IICA状态寄存器n (IICSn) 的bit0。

通过图14-16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14-16 通信预约的接受时序

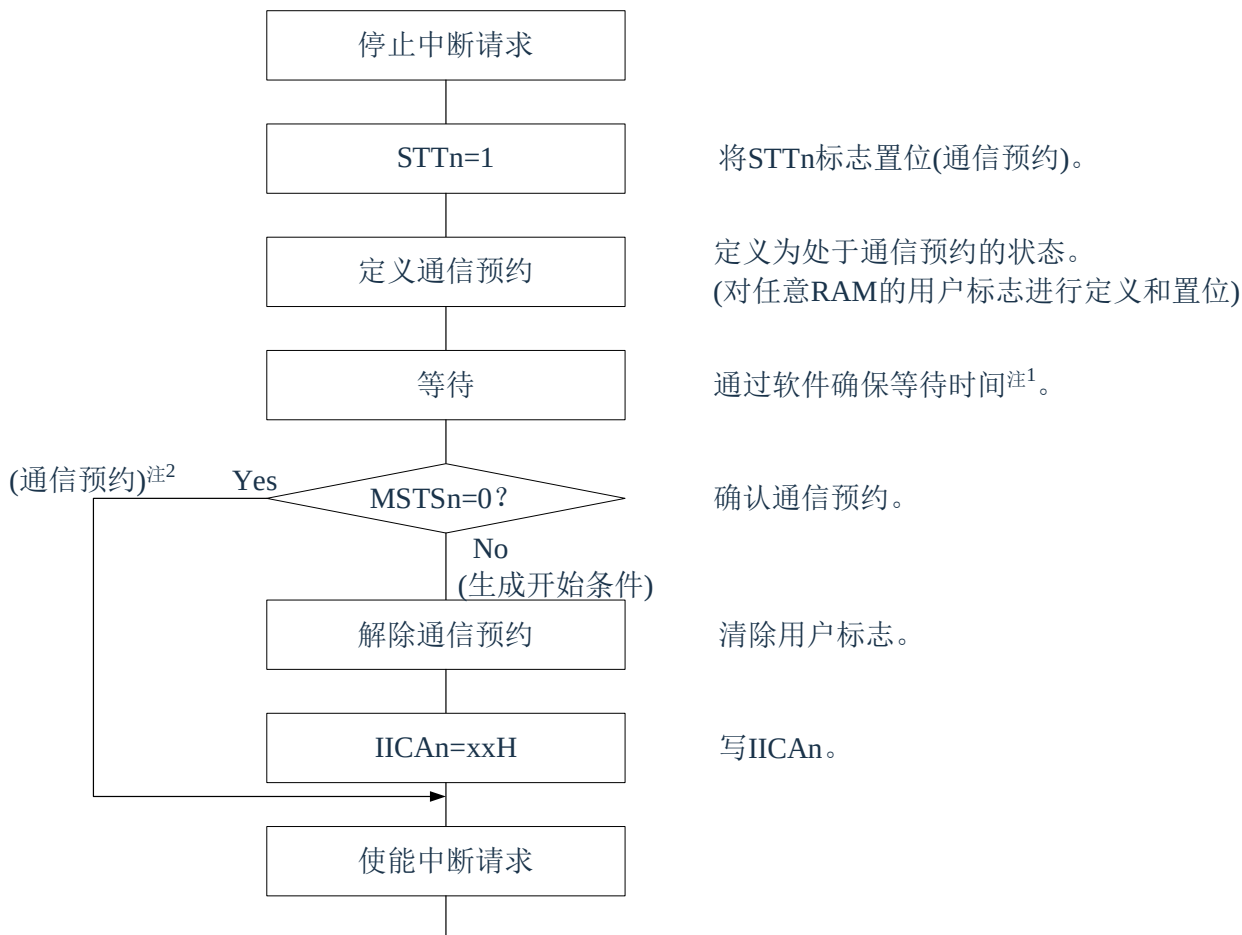


待机状态(在此期间，能通过将STTn位置“1”进行通信预约)

注: n=0

通信预约的步骤如图14-17所示。

图14-17 通信预约的步骤



注1: 等待时间如下: $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) / f_{MCK} + t_F$ 。

注2: 在通信预约运行时, 通过停止条件中断请求来写IICA移位寄存器n (IICAn)。

注3: STTn: IICA控制寄存器n0 (IICCTLn0) 的bit1。

注4: MSTS n: IICA状态寄存器n (IICS n) 的bit7。

注5: IICAn: IICA移位寄存器n。

注6: IICWLn: IICA低电平宽度设定寄存器n。

注7: IICWHn: IICA高电平宽度设定寄存器n。

注8: t_F : SDAAn信号和SCLAn信号的下降时间。

注9: f_{MCK} : IICA运行时钟频率。

注10: 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, 如果在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:

- a) 在仲裁结果既不是主控设备也不是从属设备时
- b) 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。

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:

- 设定 IICA 控制寄存器 n1 (IICCTLn1)。
- 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) 置“1”。
- 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:

- a) 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)。
- b) 将 IICCTLn0 寄存器的 bit7 (IICEn) 置“1”, 允许 I²C 运行。
- c) 等待检测到开始条件。
- d) 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MSTS_n位 (IICA状态寄存器n (IICSn) 的bit7) 时, 不需要将SPIEn位置“1”。

注: n=0。

14.5.16 通信运行

在此通过流程图表示以下3个运行步骤。

(1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在 I2C 总线的多主控系统中，只根据 I2C 总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1帧）为高电平，就将总线作为释放状态而参加通信。此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

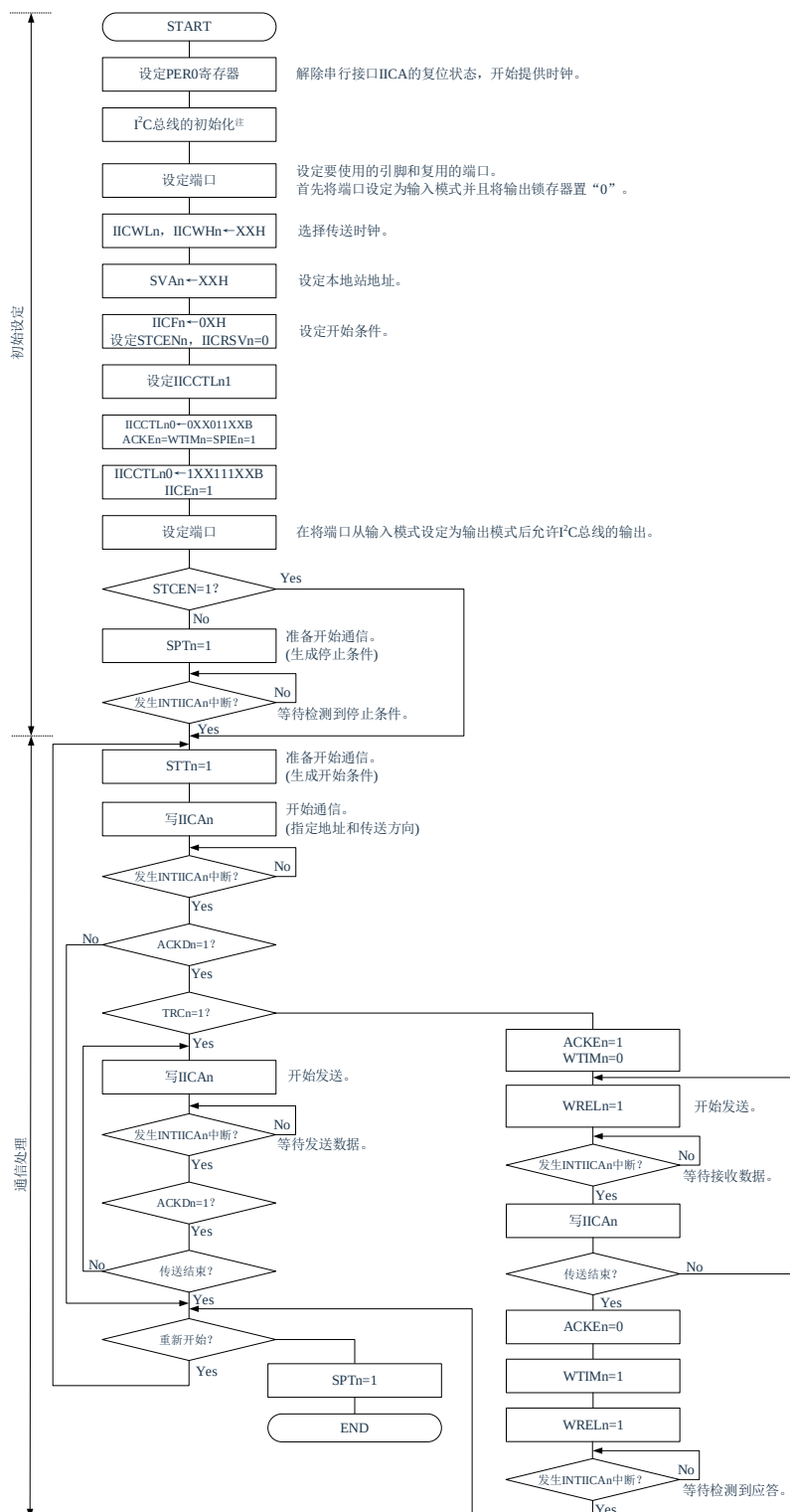
用作 I2C 总线从属设备的例子如下所示。

当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

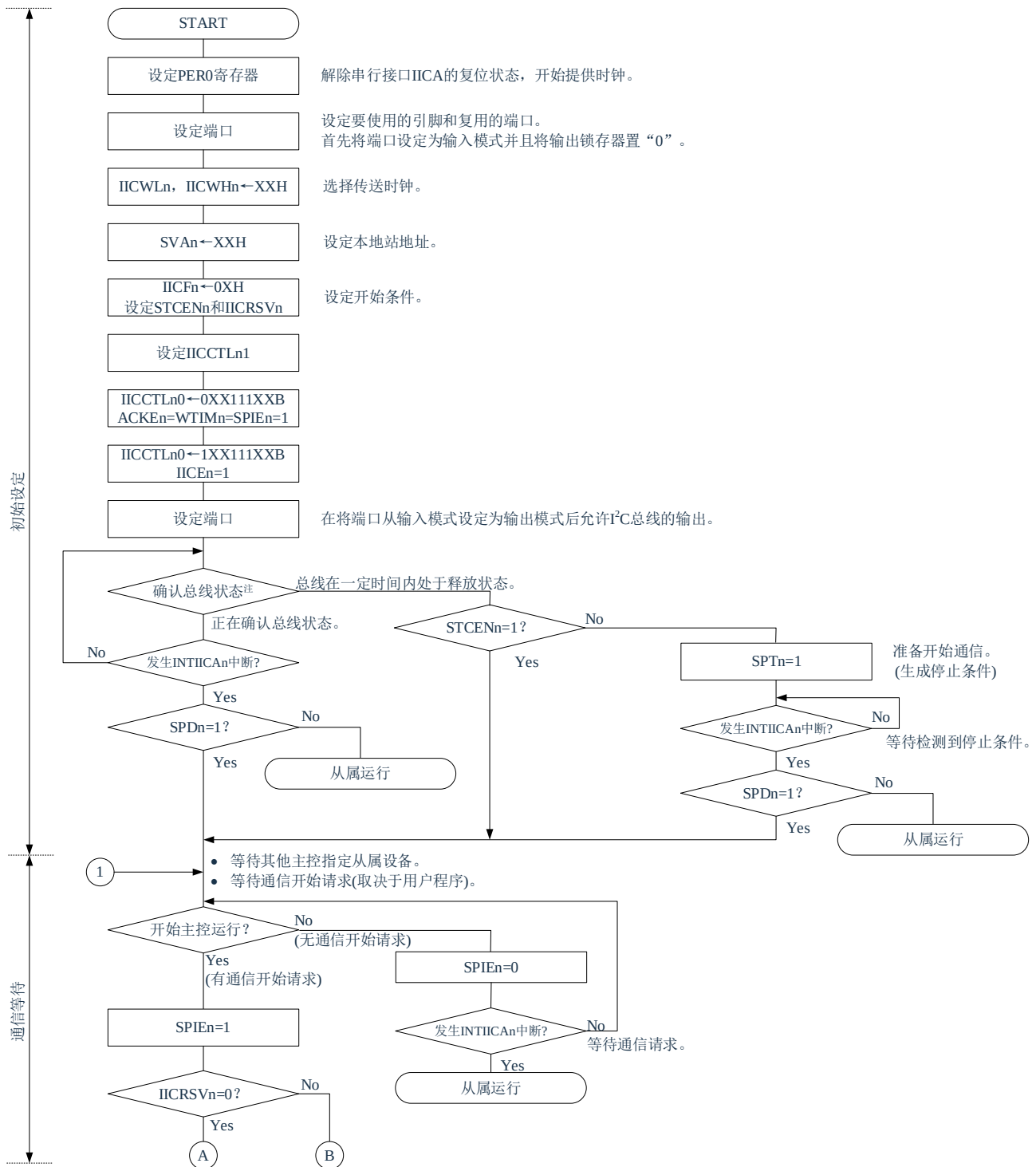
注：n=0。

图14-18 单主控系统的主控运行



② 多主控系统的主控运行

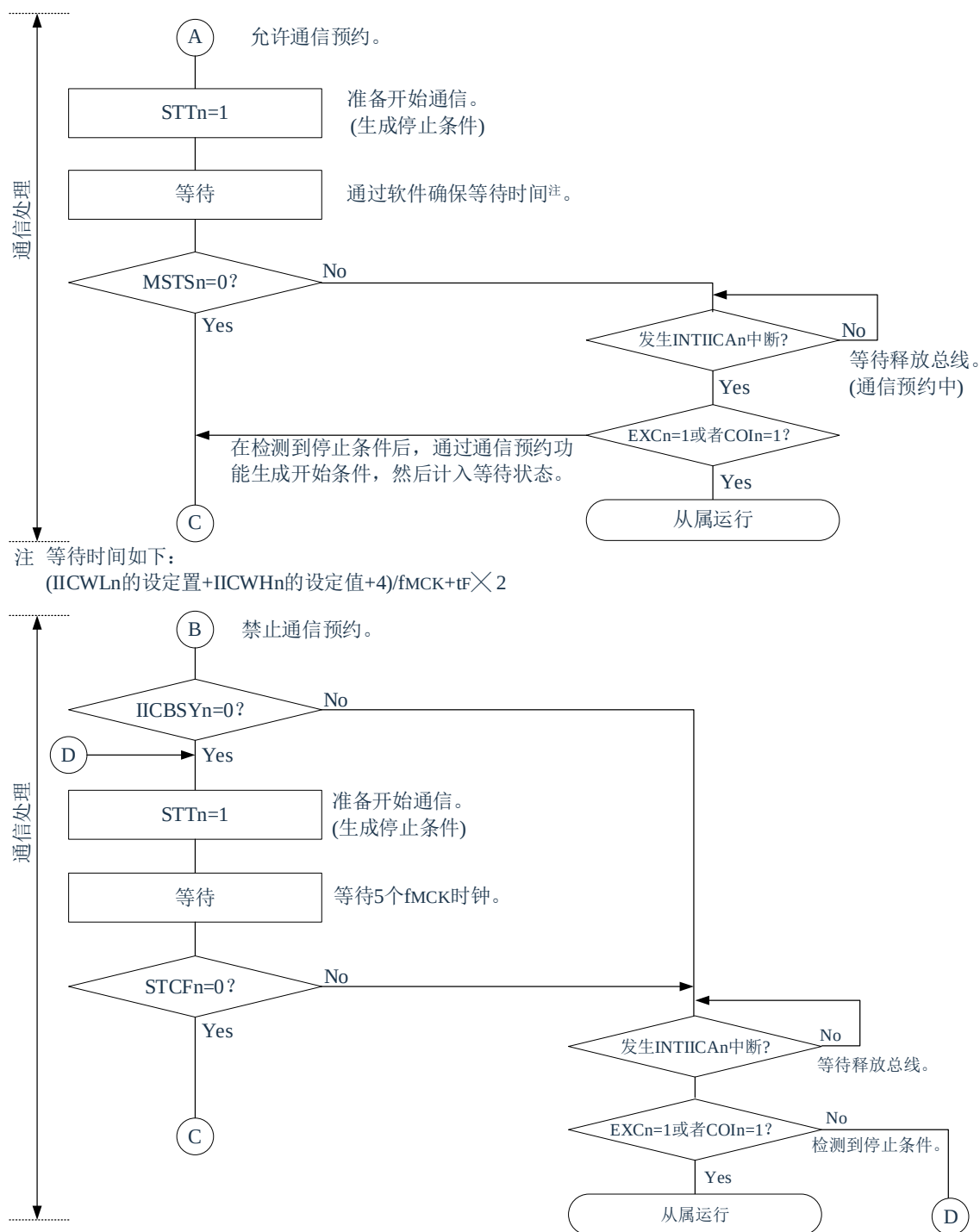
图14-19 多主控系统的主控运行(1/3)



注1: 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)。

注2: n=0。

图14-19 多主控系统的主控运行(2/3)



注1: IICWL_n: IICA低电平宽度设定寄存器n。

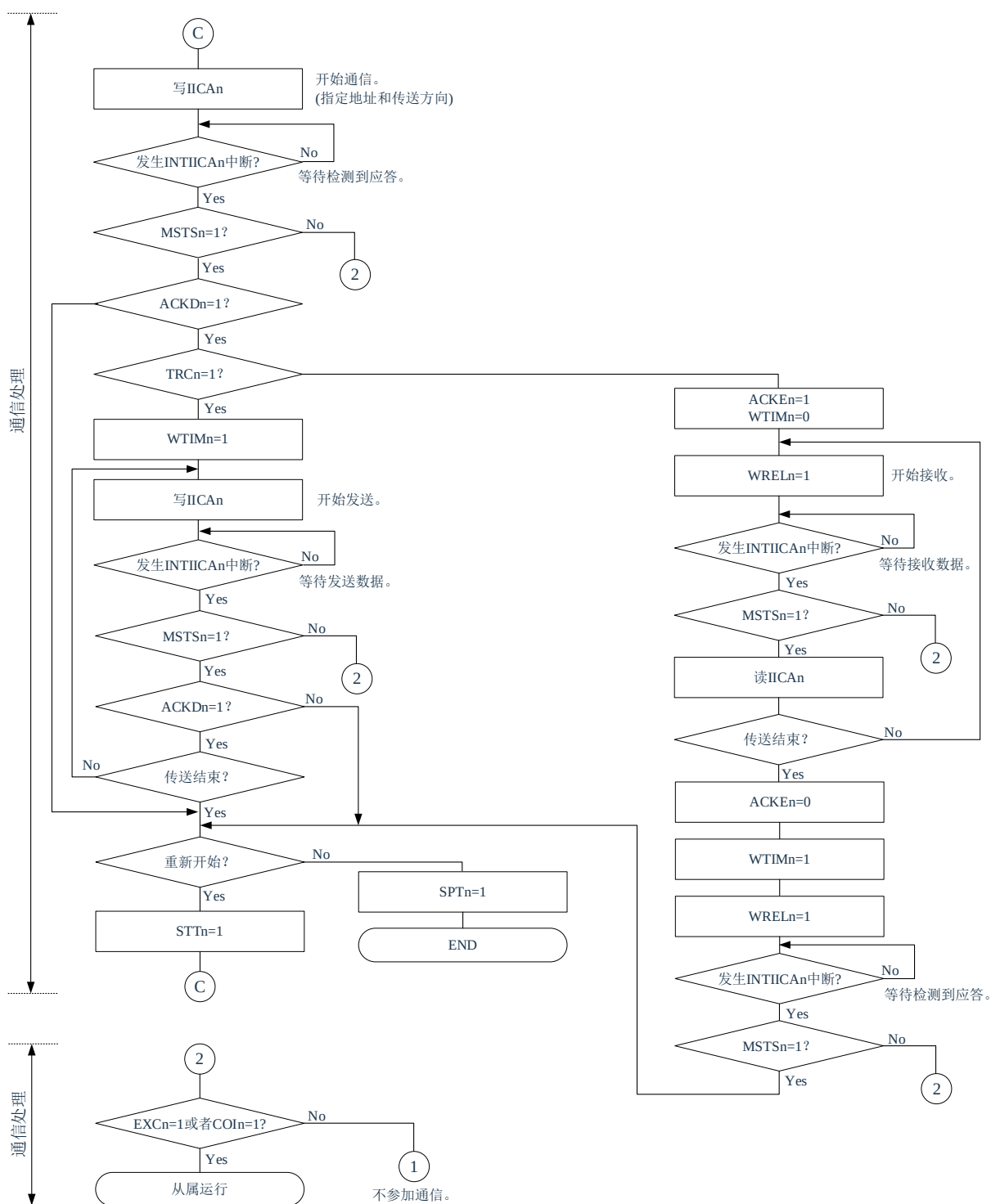
注2: IICWH_n: IICA高电平宽度设定寄存器n。

注3: t_F: SDAAn信号和SCLAn信号的下降时间。

注4: f_{MCK}: IICA运行时钟频率。

注5: n=0。

图14-19 多主控系统的主控运行(3/3)



注1: 传送和接收的格式必须符合通信中的产品的规格。

注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MSTS_n位, 确认仲裁结果。

注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 (IICS_n) 和IICA标志寄存器_n (IICF_n) 确认状态, 决定以后的处理。

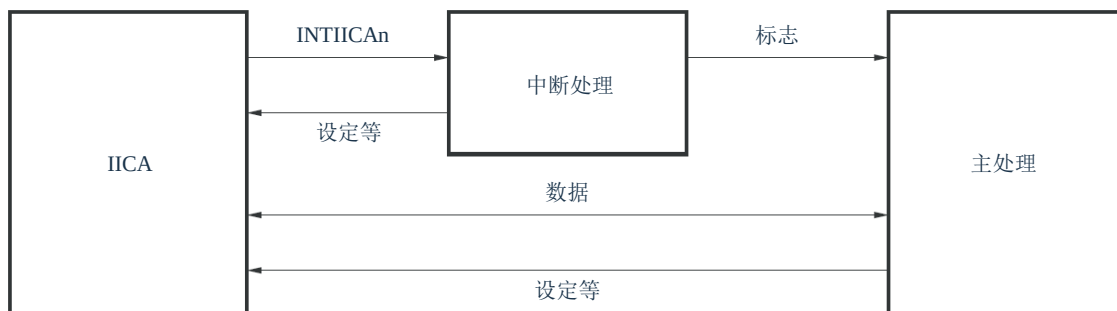
注4: n=0。

(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。

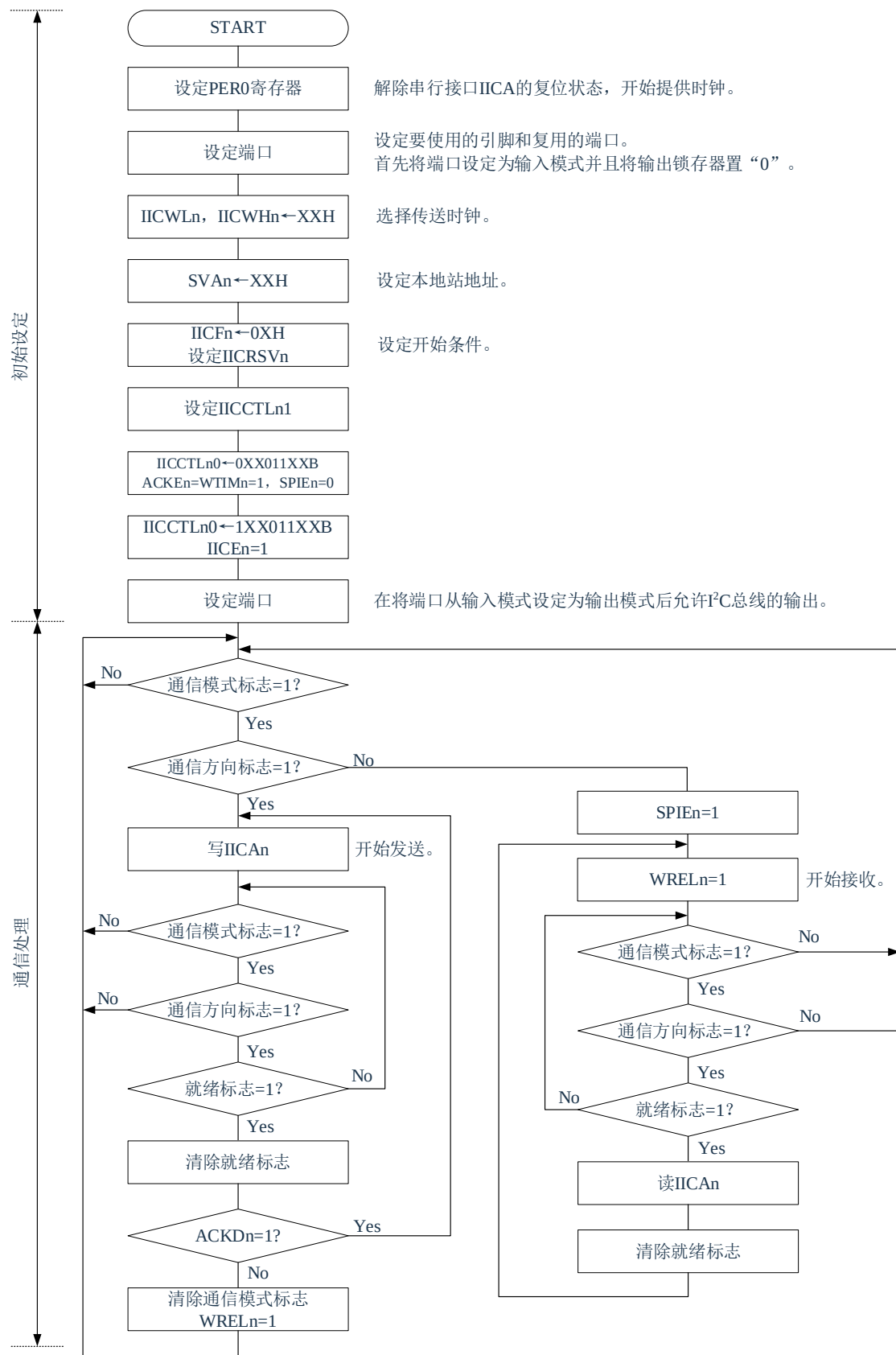
注：n=0。

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14-20 从属运行步骤(1)



注1：传送和接收的格式必须符合通信中的产品的规格。

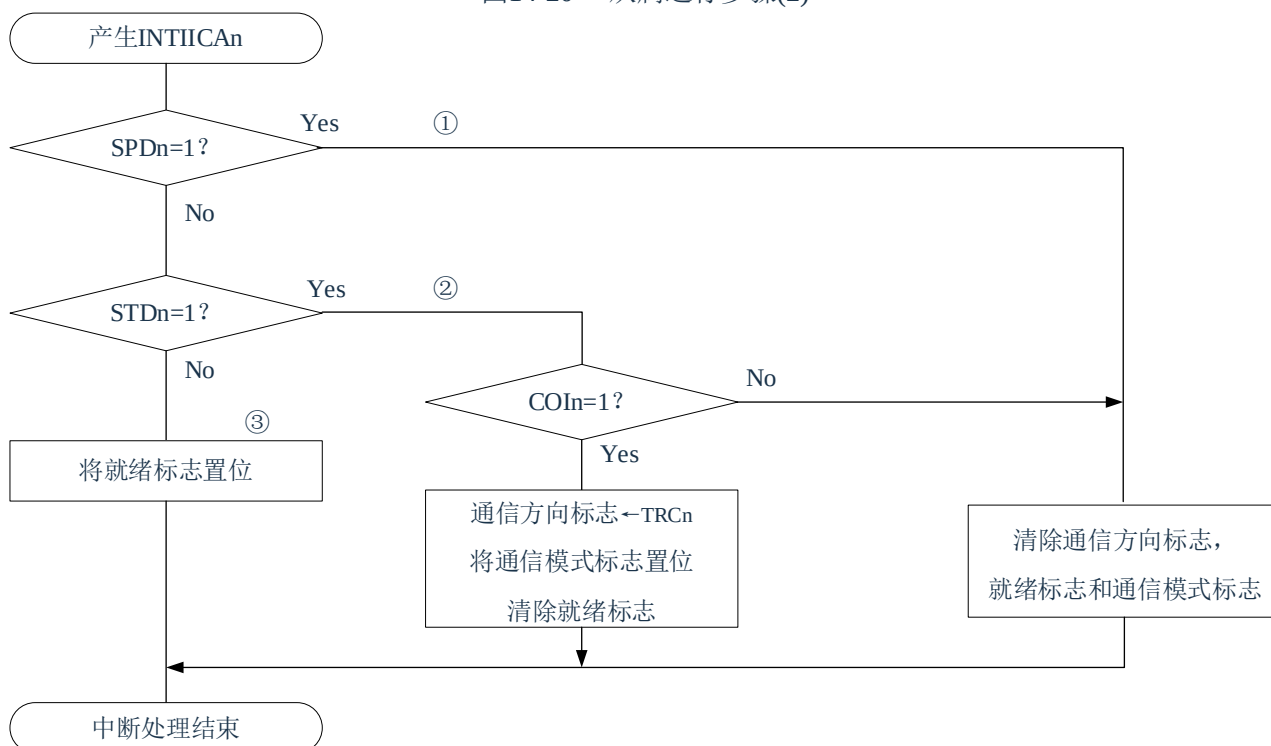
注2：n=0。

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。

- 如果生成停止条件，就结束通信。
- 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设定为通信模式并且解除等待，然后从中断返回（清除就绪标志）。
- 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。

注：上述的①～③对应“图14-20 从属运行步骤(2)”的①～③。

图14-20 从属运行步骤(2)



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。

注1: ST: 开始条件。

注2: AD6~AD0: 地址。

注3: R/W: 传送方向的指定。

注4: ACK: 应答。

注5: D7~D0: 数据。

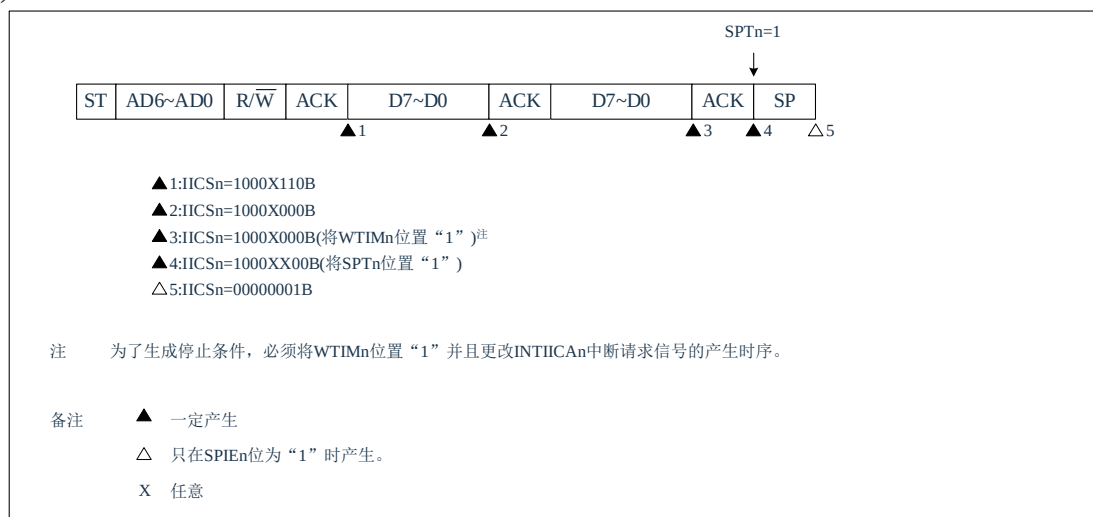
注6: SP: 停止条件。

注7: n=0。

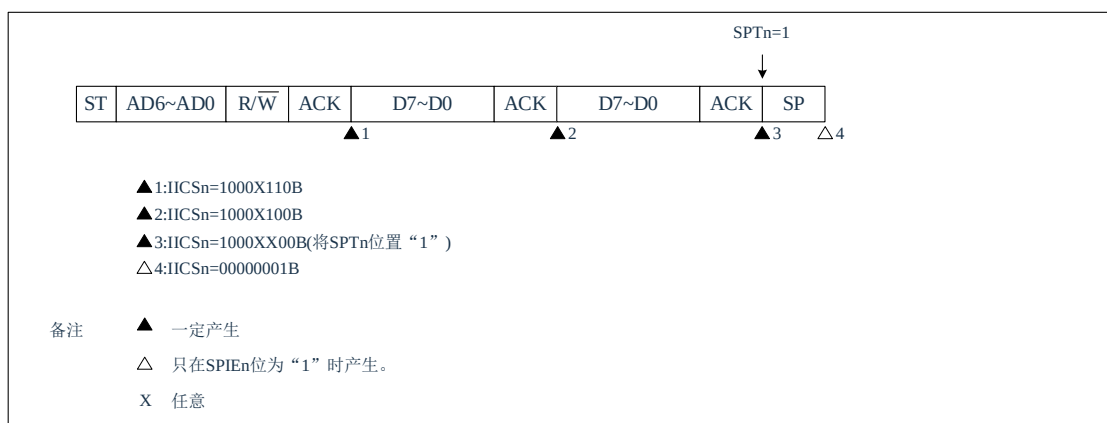
(1) 主控运行

a) Start~Address~Data~Data~Stop (发送和接收)

i) WTIMn=0的情况



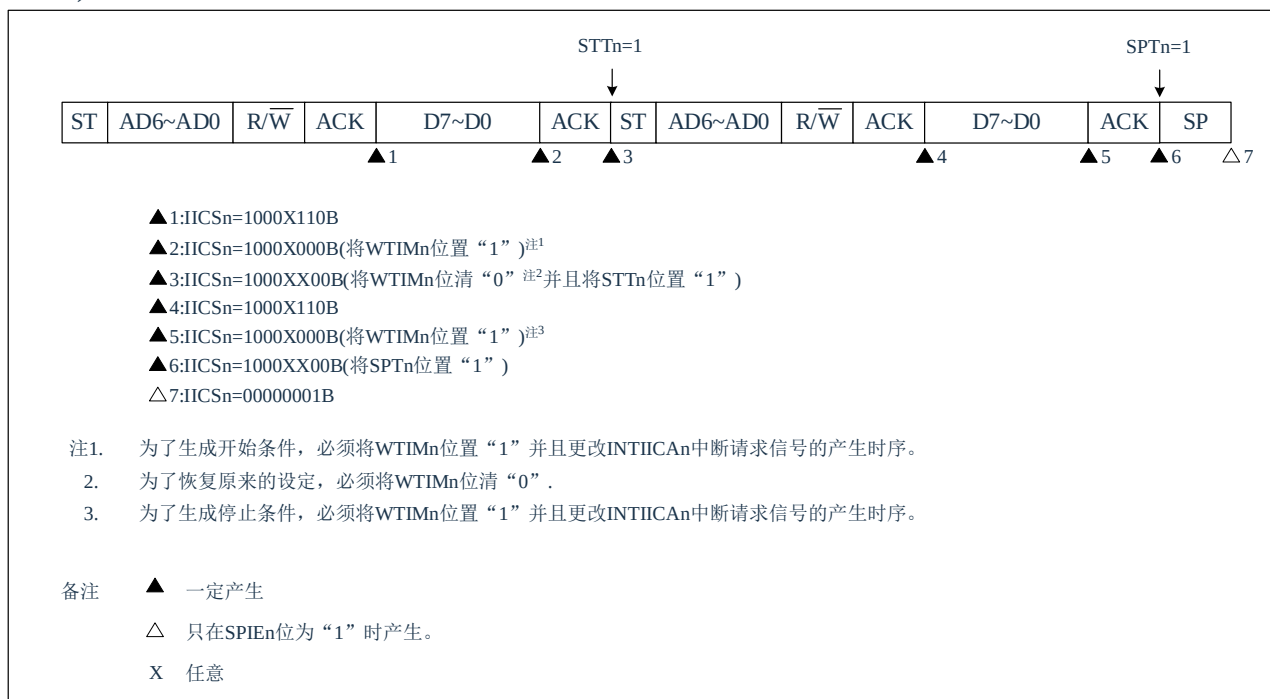
ii) WTIM0=1的情况



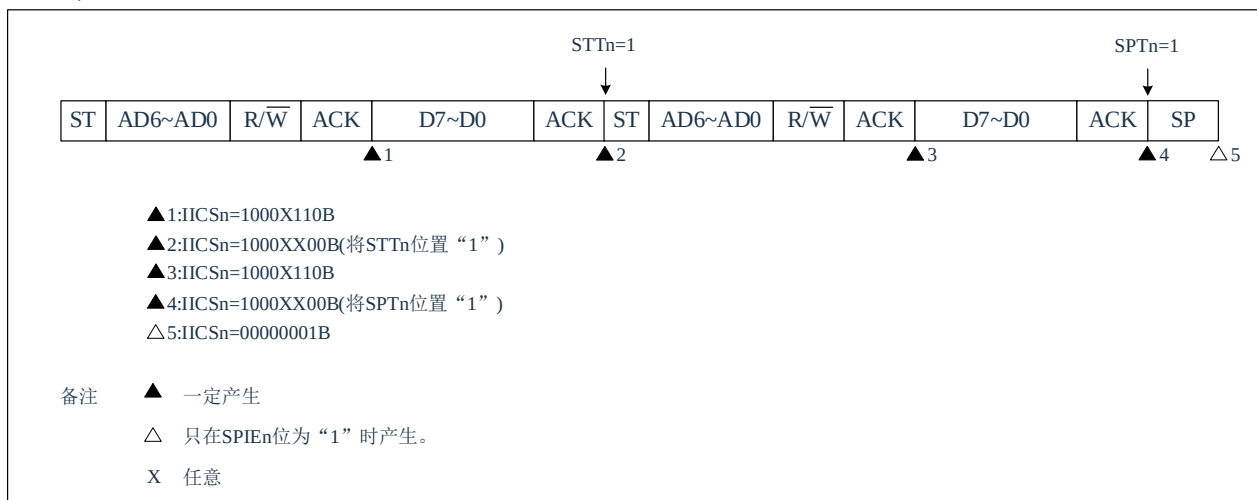
注: n=0。

b) Start~Address~Data~Start~Address~Data~Stop (重新开始)

i) WTIMn=0的情况



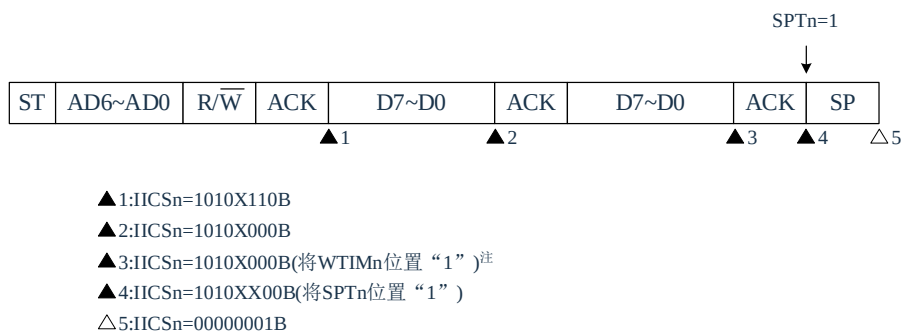
ii) WTIMn=1的情况



注: n=0。

c) Start~Code~Data~Data~Stop (发送扩展码)

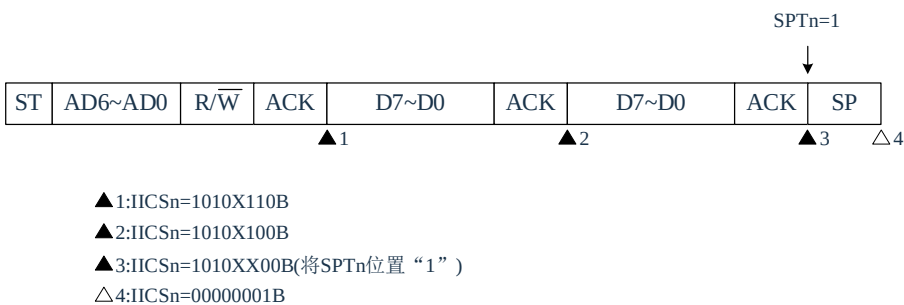
i) WTIMn=0的情况



注 为了生成停止条件，必须将WTIM_n位置“1”并且更改INTIICAn中断请求信号的产生时序。

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

ii) WTIMn=1的情况



备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

注: n=0。

(2) 从属运行（接收从属地址的情况）

a) Start~Address~Data~Data~Stop

i) WTIMn=0的情况



▲1:IICS_n=0001X110B

▲2:IICS_n=0001X000B

▲3:IICS_n=0001X000B

△4:IICS_n=00000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

ii) WTIMn=1的情况



▲1:IICS_n=0001X110B

▲2:IICS_n=0001X100B

▲3:IICS_n=0001XX00B

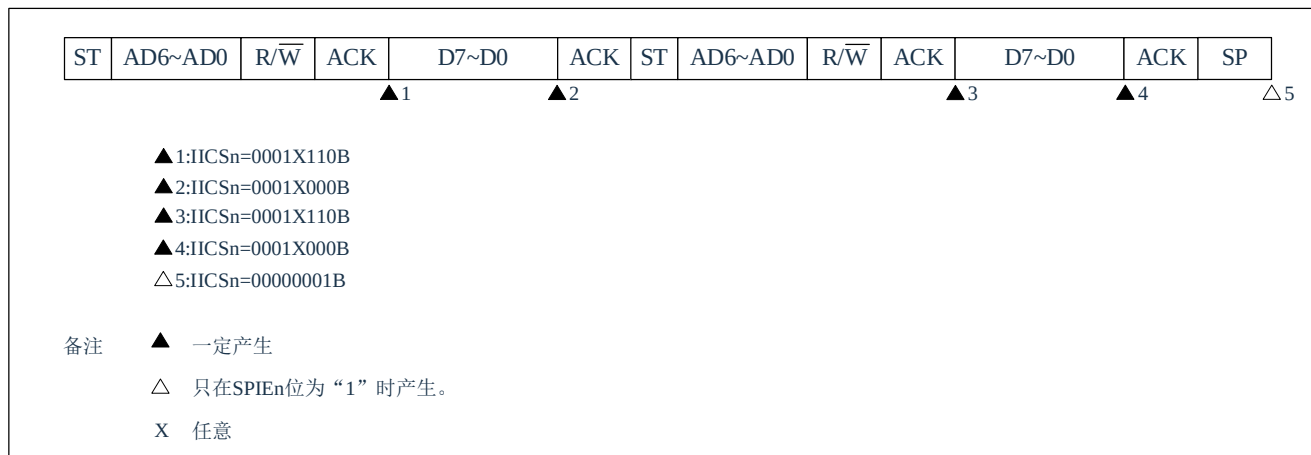
△4:IICS_n=00000001B

备注 ▲ 一定产生
 △ 只在SPIEn位为“1”时产生。
 X 任意

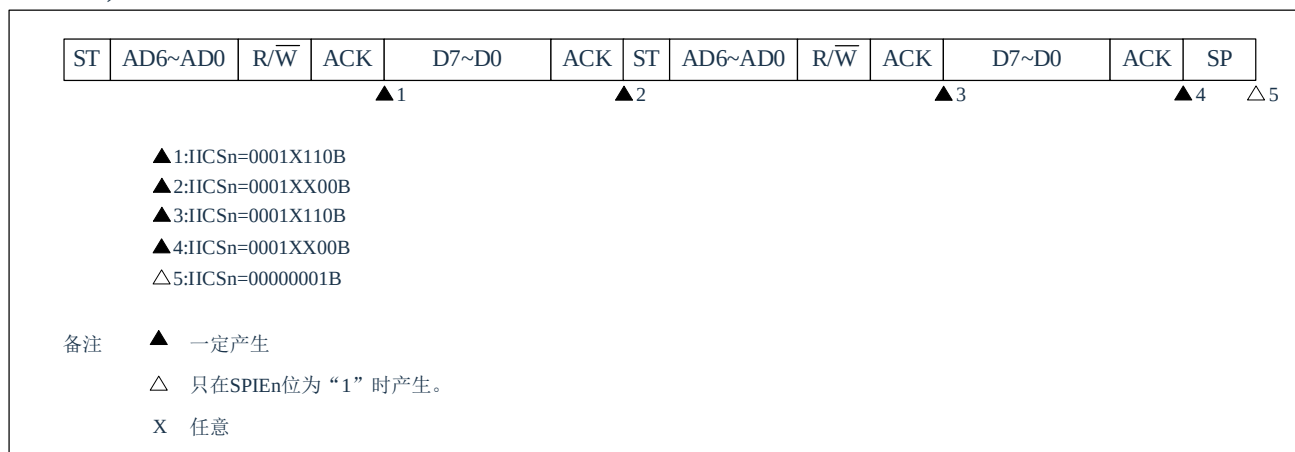
注：n=0。

b) Start~Address~Data~Start~Address~Data~Stop

i) WTIMn=0的情况 (在重新开始后 SVAn 相同)



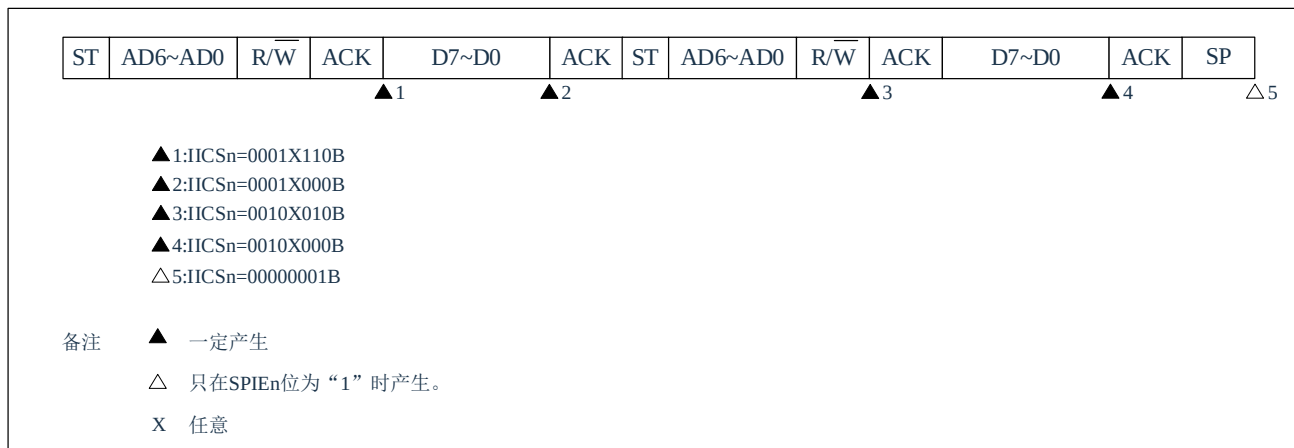
ii) WTIMn=1的情况 (在重新开始后 SVAn 相同)



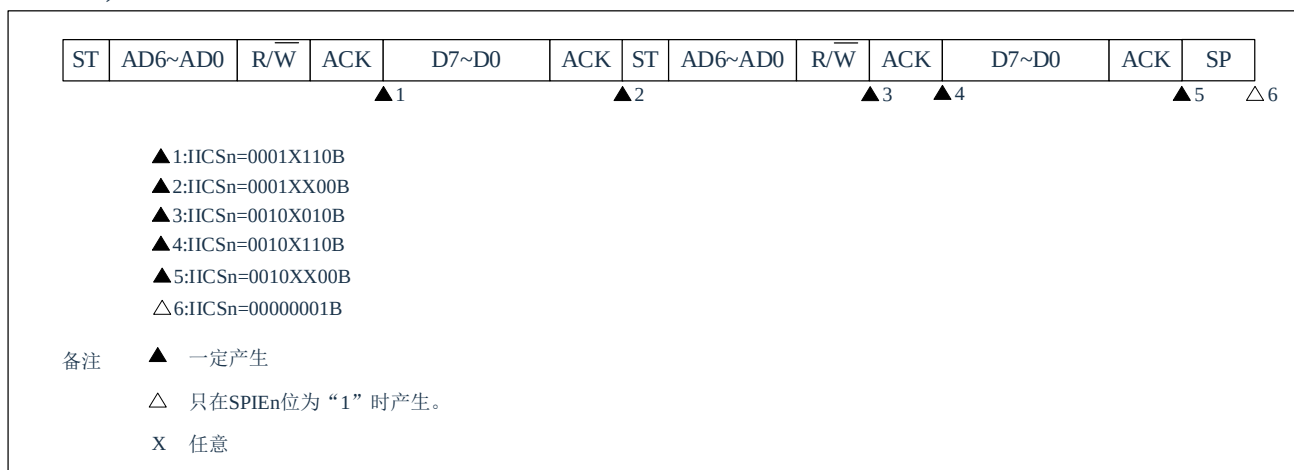
注：n=0。

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))



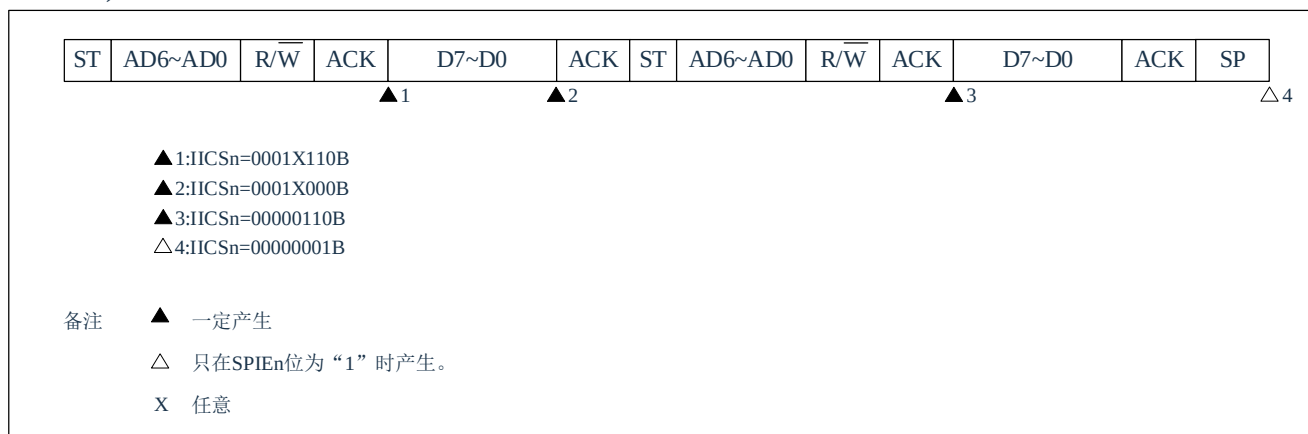
ii) WTIMn=1的情况 (在重新开始后地址不同 (扩展码))



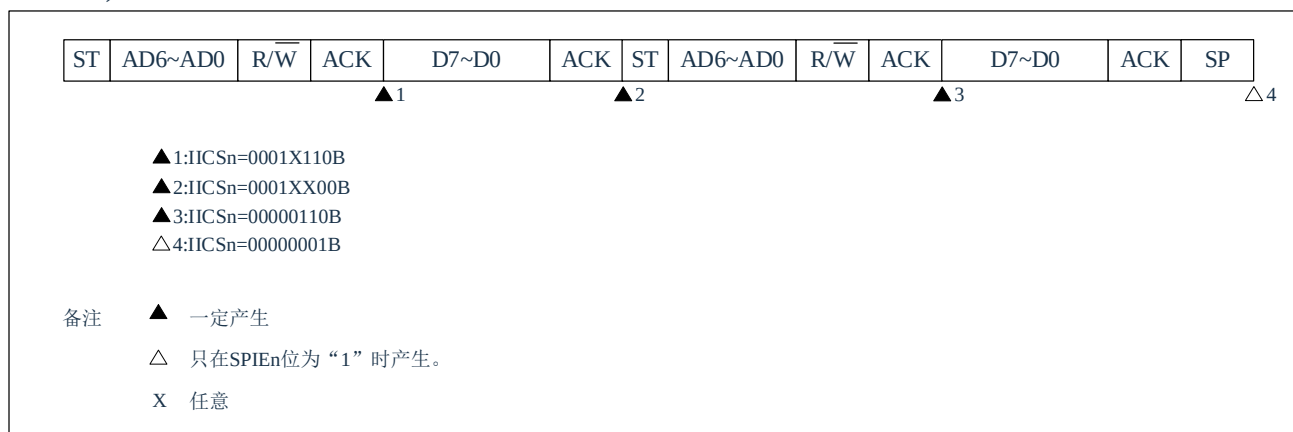
注：n=0。

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))



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=250的情况



▲1:IICS_n=0010X010B

▲2:IICS_n=0010X000B

▲3:IICS_n=0010X000B

△4:IICS_n=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

ii) WTIMn=1的情况



▲1:IICS_n=0010X010B

▲2:IICS_n=0010X110B

▲3:IICS_n=0010X100B

▲4:IICS_n=0010XX00B

△5:IICS_n=00000001B

备注 ▲ 一定产生

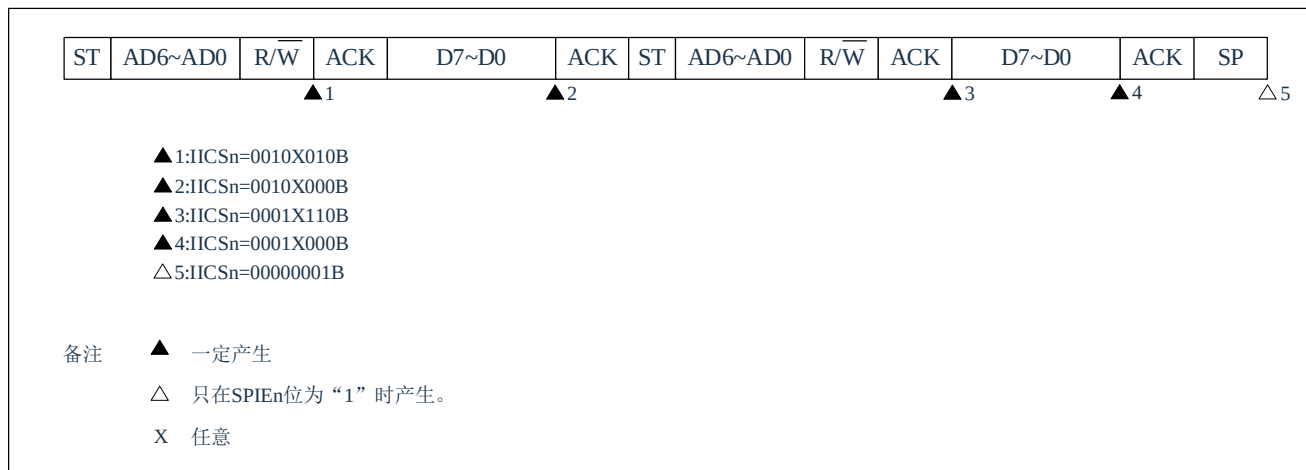
△ 只在SPIEn位为“1”时产生。

X 任意

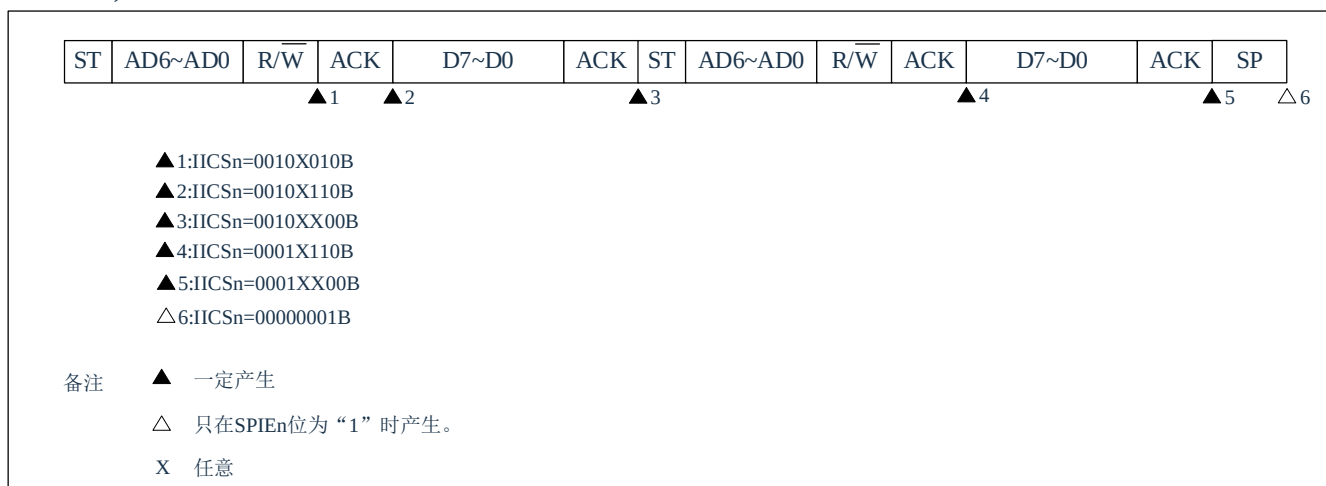
注：n=0。

b) Start~Code~Data~Start~Address~Data~Stop

i) WTIMn=0的情况 (在重新开始后 SVAn 相同)



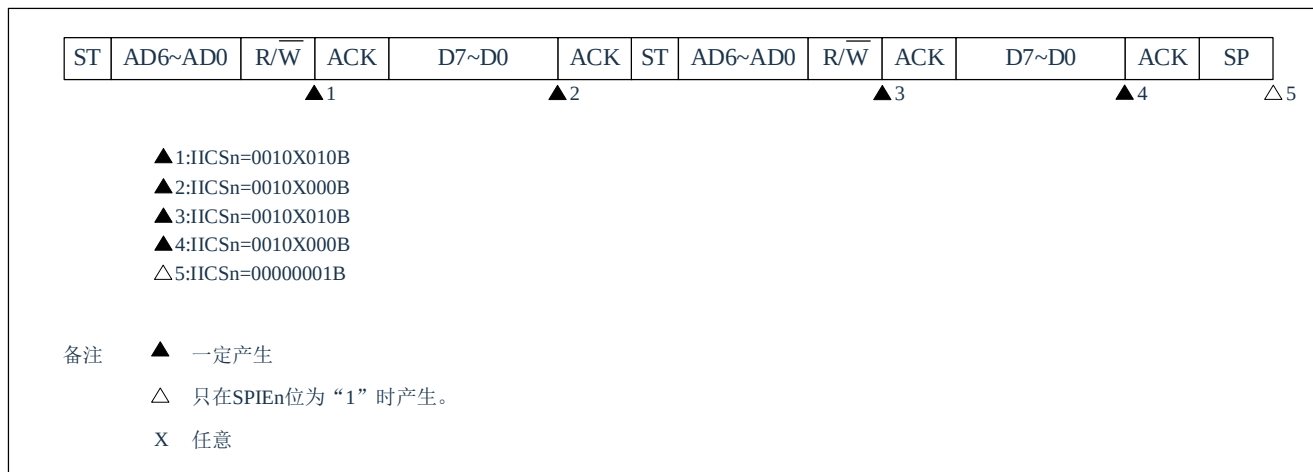
ii) WTIMn=1的情况 (在重新开始后 SVAn 相同)



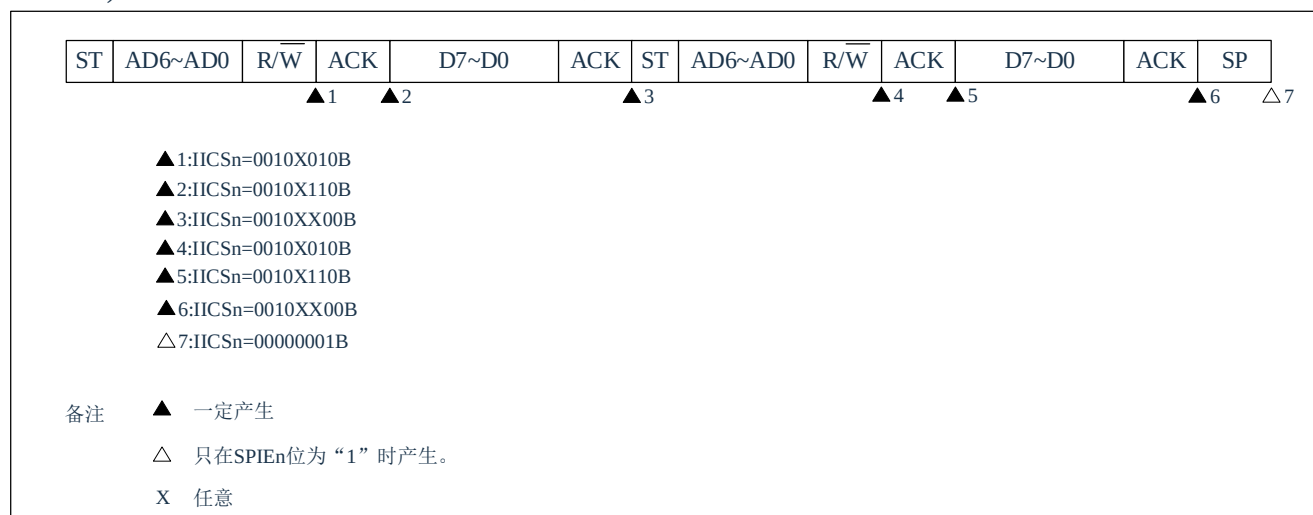
注：n=0。

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)



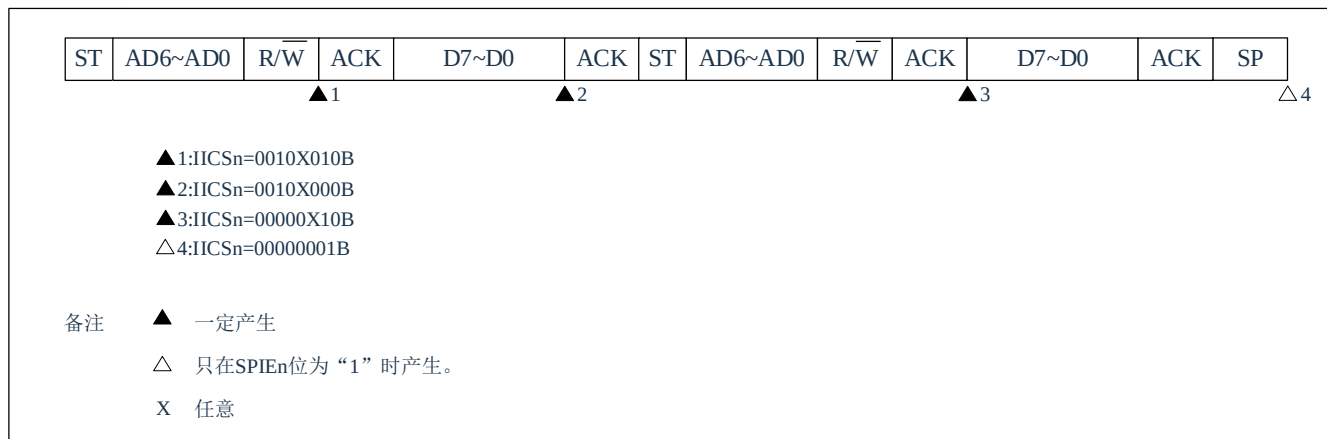
ii) WTIMn=1的情况 (在重新开始后接收扩展码)



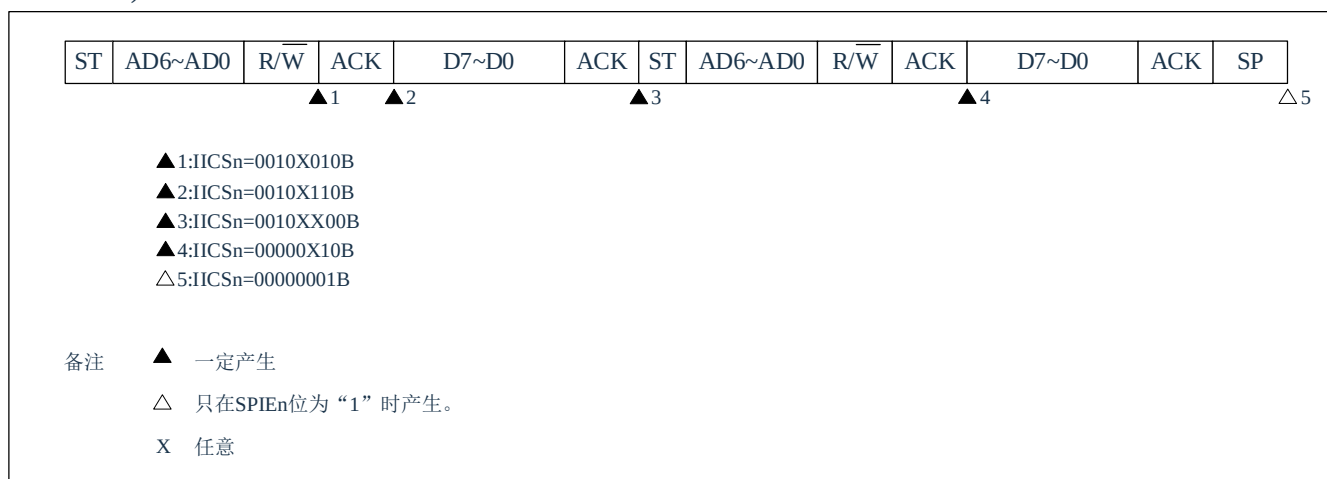
注：n=0。

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))



注: n=0。

(4) 不参加通信的运行

a) Start~Code~Data~Data~Stop



△1:IICSn=00000001B

备注 △ 只在SPIEn位为“1”时产生。

(5) 仲裁失败的运行 (在仲裁失败后作为从属设备运行)

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

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



▲1:IICSn=0101X110B

▲2:IICSn=0001X000B

▲3:IICSn=0001X000B

△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

注：n=0。

ii) WTIMn=1的情况



▲1:IICSn=0101X110B

▲2:IICSn=0001X100B

▲3:IICSn=0001XX00B

△4:IICSn=00000001B

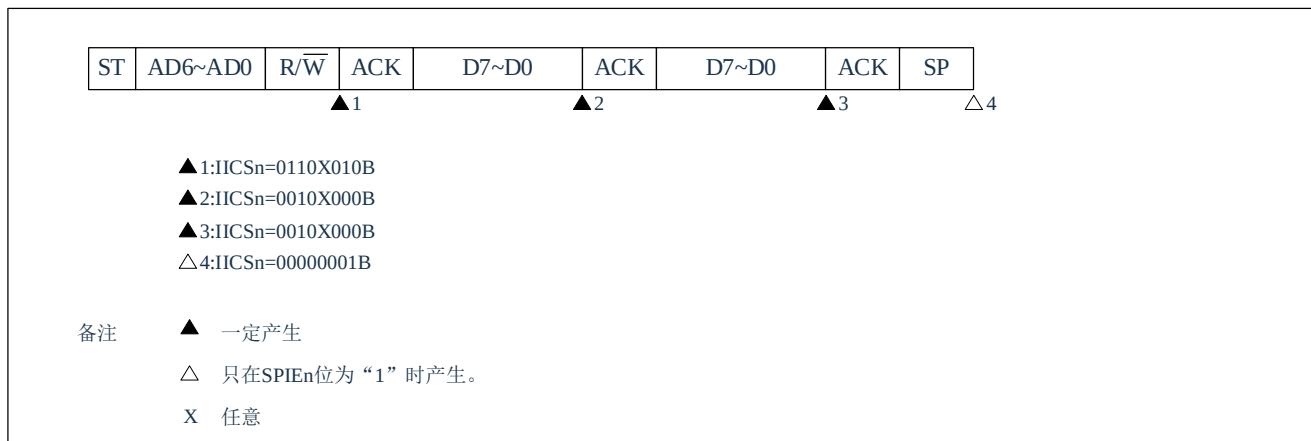
备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

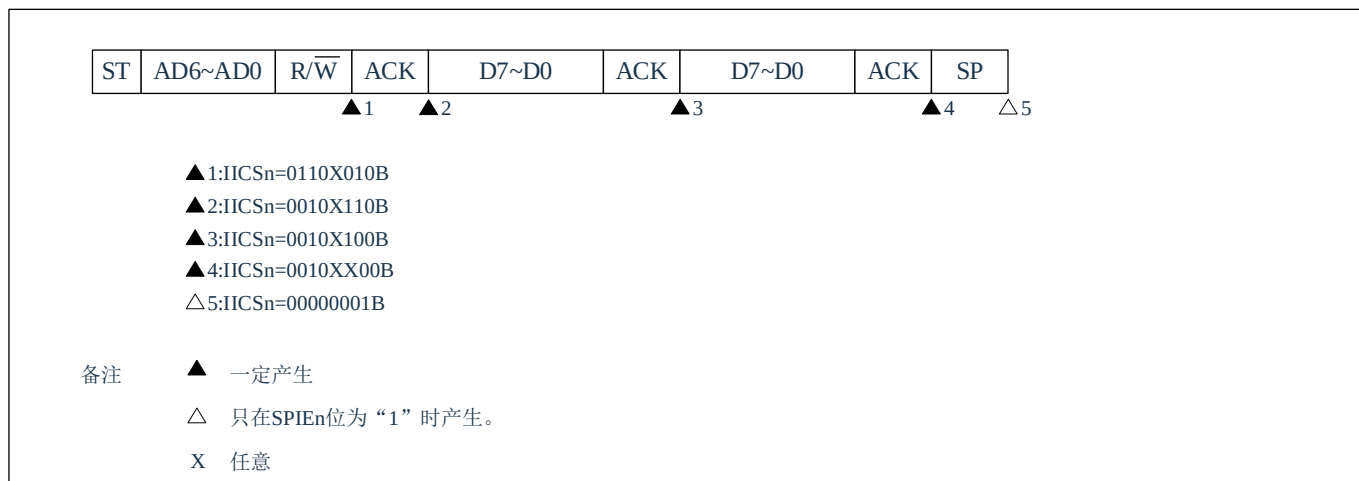
b) 在发送扩展码的过程中仲裁失败的情况

i) WTIMn=0的情况



注：n=0。

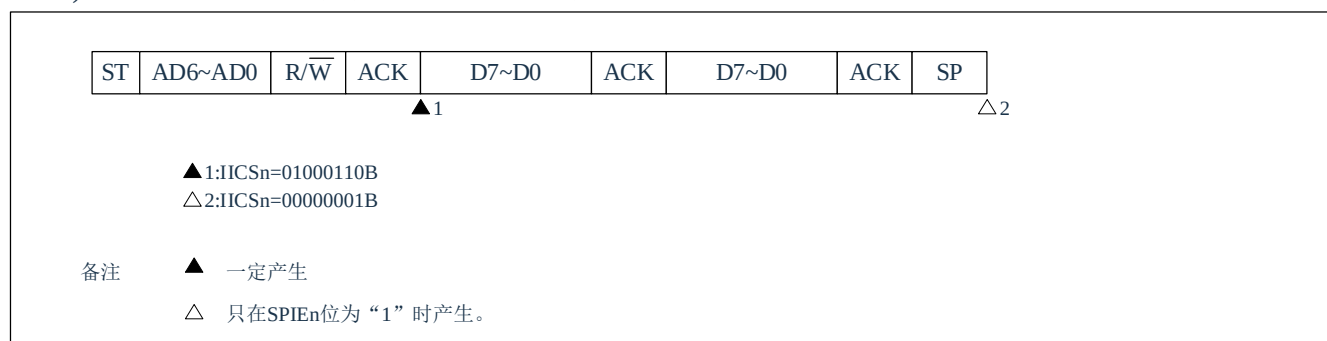
ii) WTIMn=1的情况



(6) 仲裁失败的运行（在仲裁失败后不参加通信）

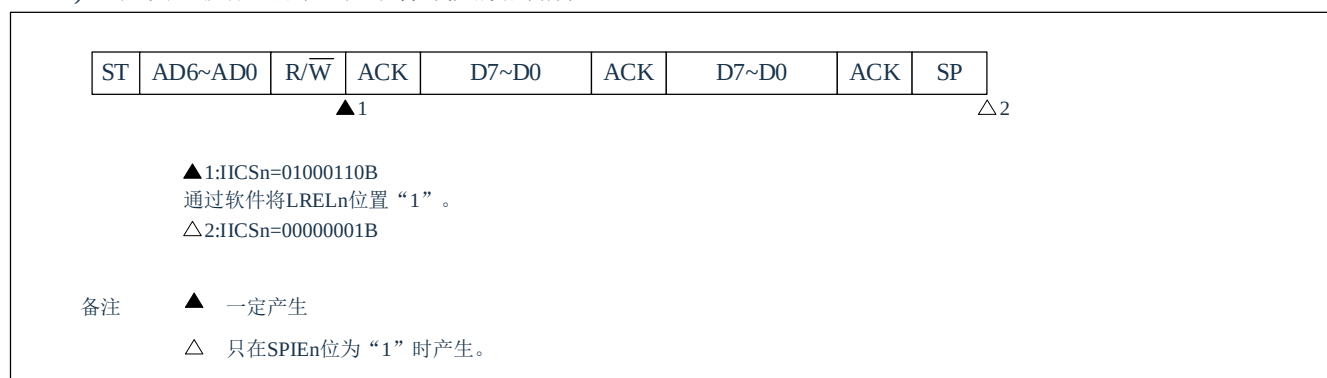
在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

a) 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）



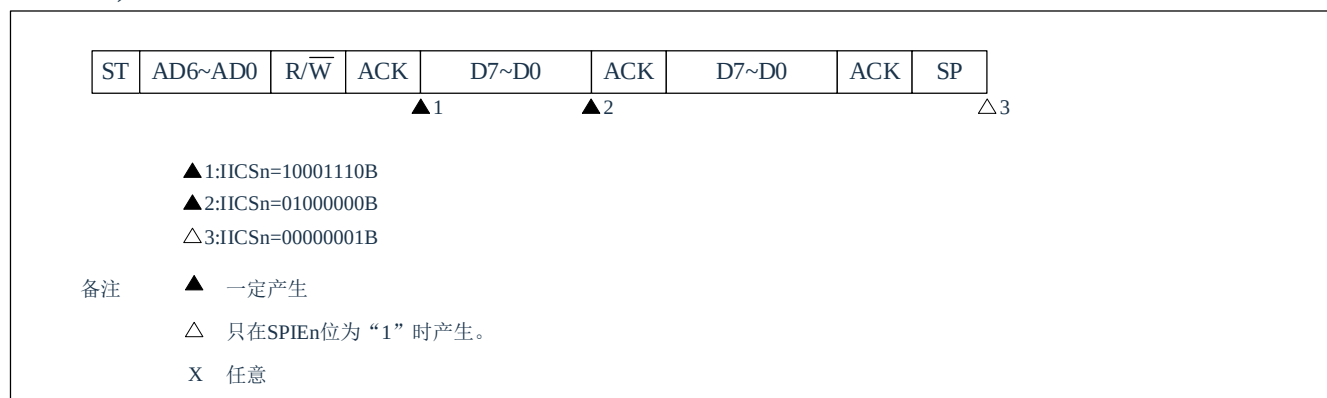
注：n=0。

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



注：n=0。

ii) WTIMn=1的情况



▲1:IICSn=10001110B

▲2:IICSn=01000100B

△3:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

注：n=0。

d) 在传送数据时因重新开始条件而仲裁失败的情况

i)非扩展码（例如，SVAn不同）



▲1:IICSn=1000X110B

▲2:IICSn=01000110B

△3:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

注：n=0。

ii) 扩展码



▲1:IICSn=1000X110B

▲2:IICSn=01000010B

通过软件将LRELn位置“1”。

△3:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

m=0~6

ST	AD6~AD0	R/ $\overline{W}$	ACK	D7~Dm	SP

▲1
△2

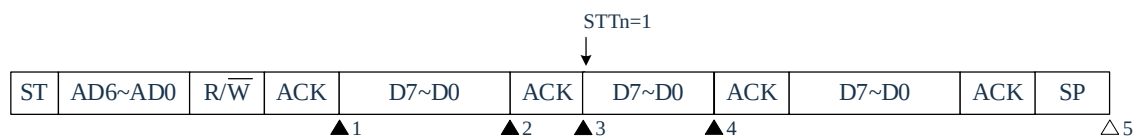
$\Delta 2: ICS_n = 01000001B$

△ 只在SPIEn位为“1”时产生。

 $m=0\sim 6$

注: $n=0$ 。

i) WTIM_n=0的情况



△5:IICS_n=00000001B

△ 只在SPIEn位为“1”时产生。

X 任意

Timing diagram for STTn=1. The sequence of signals is: AD6~AD0, R/W, ACK, D7~D0, ACK, D7~D0, ACK, D7~D0, ACK, and SP. Triangles 1, 2, 3, and 4 mark the start of the first ACK, the second ACK, the third ACK, and the end of the SP signal, respectively.

△4:IICS_n=00000001B

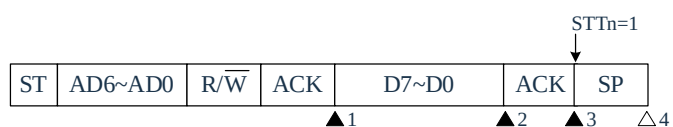
△ 只在SPIEn位为“1”时产生。

X 任意

注: $n=0$ 。

g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

i) WTIMn=0的情况



▲1:IICSn=1000X110B

▲2:IICSn=1000X000B(将WTIMn位置“1”)

▲3:IICSn=1000XX00B(将STTn位置“1”)

△4:IICSn=01000001B

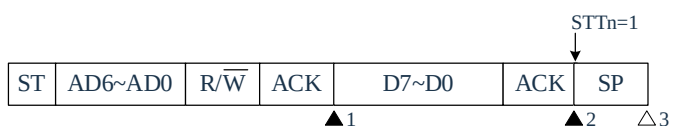
备注

▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

ii) WTIMn=1的情况



▲1:IICSn=1000X110B

▲2:IICSn=1000XX00B(将STTn位置“1”)

△3:IICSn=01000001B

备注

▲ 一定产生

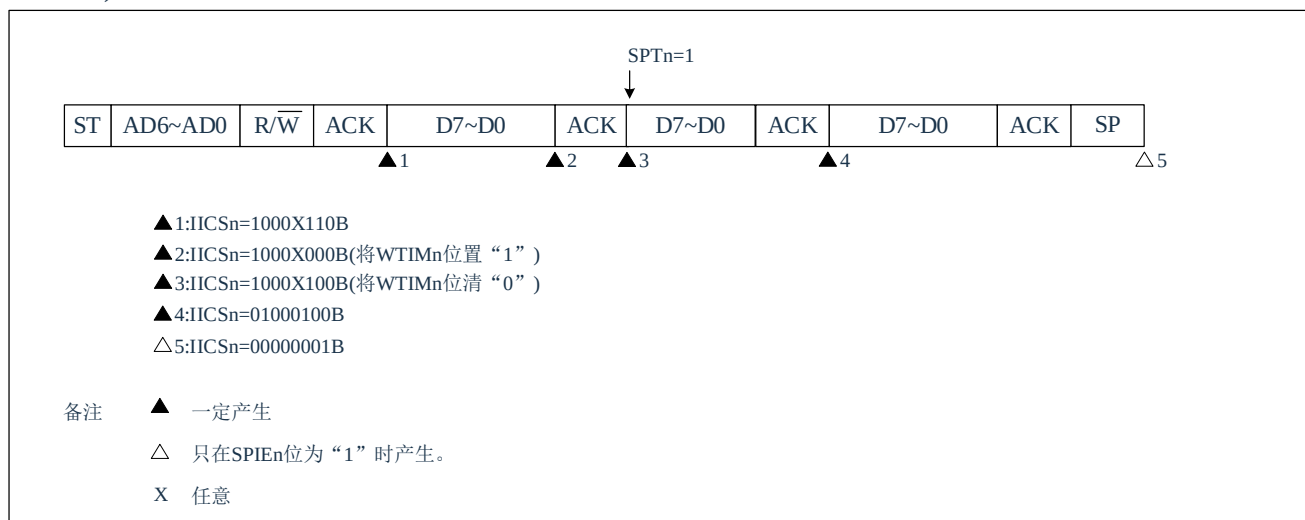
△ 只在SPIEn位为“1”时产生。

X 任意

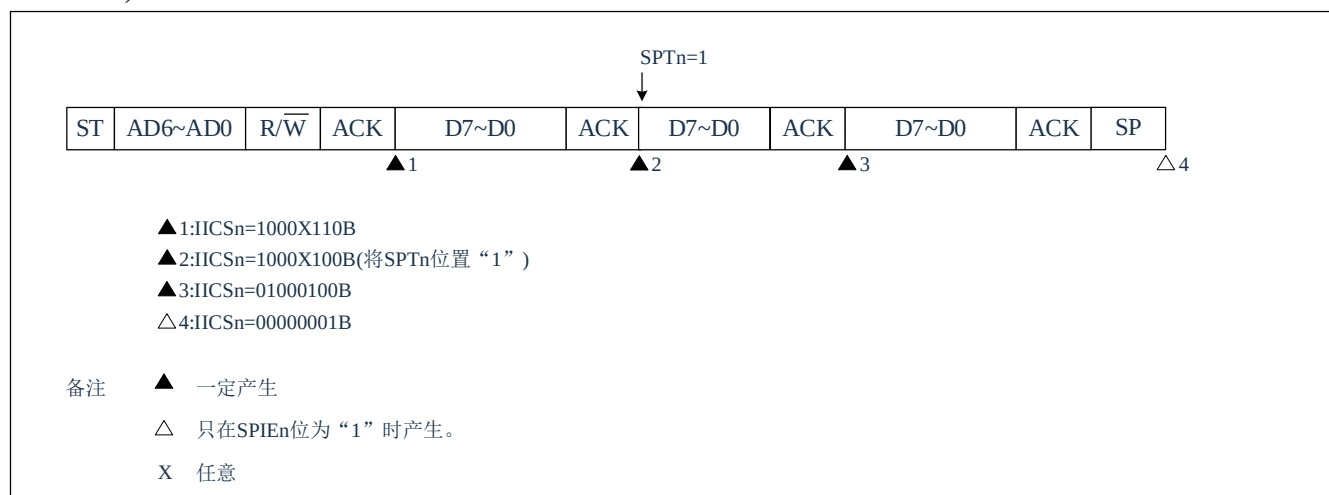
注: n=0。

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



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TRC_n位（IICA状态寄存器_n（IICS_n）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14-21和图14-22所示。

与串行时钟（SCL_{An}）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SDA_{An}引脚输出数据。

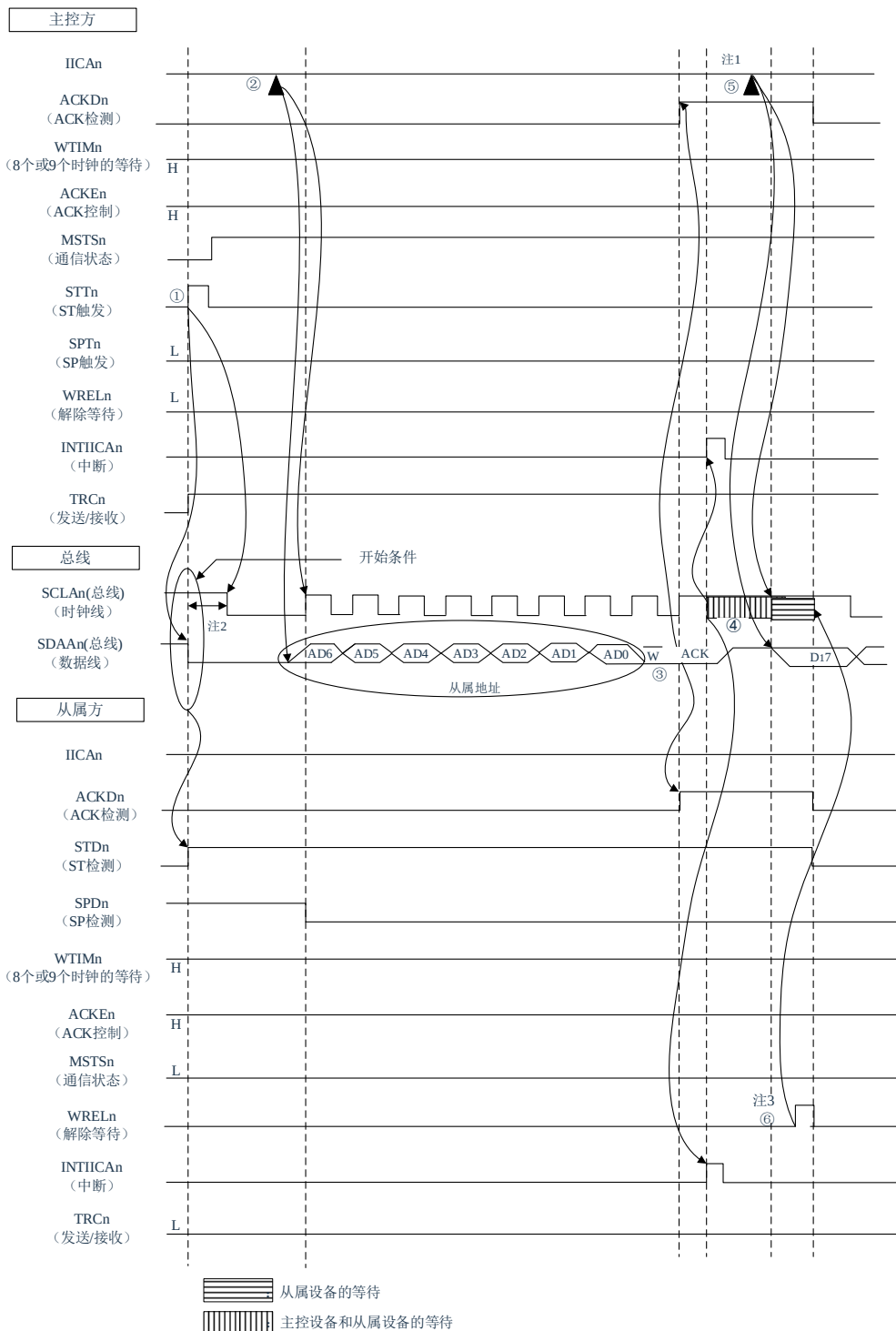
在SCL_{An}的上升沿将SDA_{An}引脚输入的数据取到IICAn。

注：n=0。

图14-21 主控设备&从属设备的通信例子

(主控设备: 选择9个时钟的等待, 从属设备: 选择9个时钟的等待) (1/4)

(1) 开始条件~地址~数据



注1: 要解除主控方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注2: SDAAAn引脚信号下降到SCLAn引脚信号下降的时间, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us

。

注3: 要解除从属方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

图14-21的“(1)开始条件~地址~数据”的①~⑥的说明如下:

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。

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。

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 ($ACKDn=1$)。

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。

主控方给 $IICAn$ 寄存器写发送数据，解除主控方的等待。

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1: 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)。

注2: 图14-21的①~⑮ 是通过I²C总线进行数据通信的一系列运行步骤。

注3: 图14-21的“(1)地址~数据~数据”说明步骤①~⑥。

注4: 图14-21的“(2)地址~数据~数据”说明步骤③~⑩。

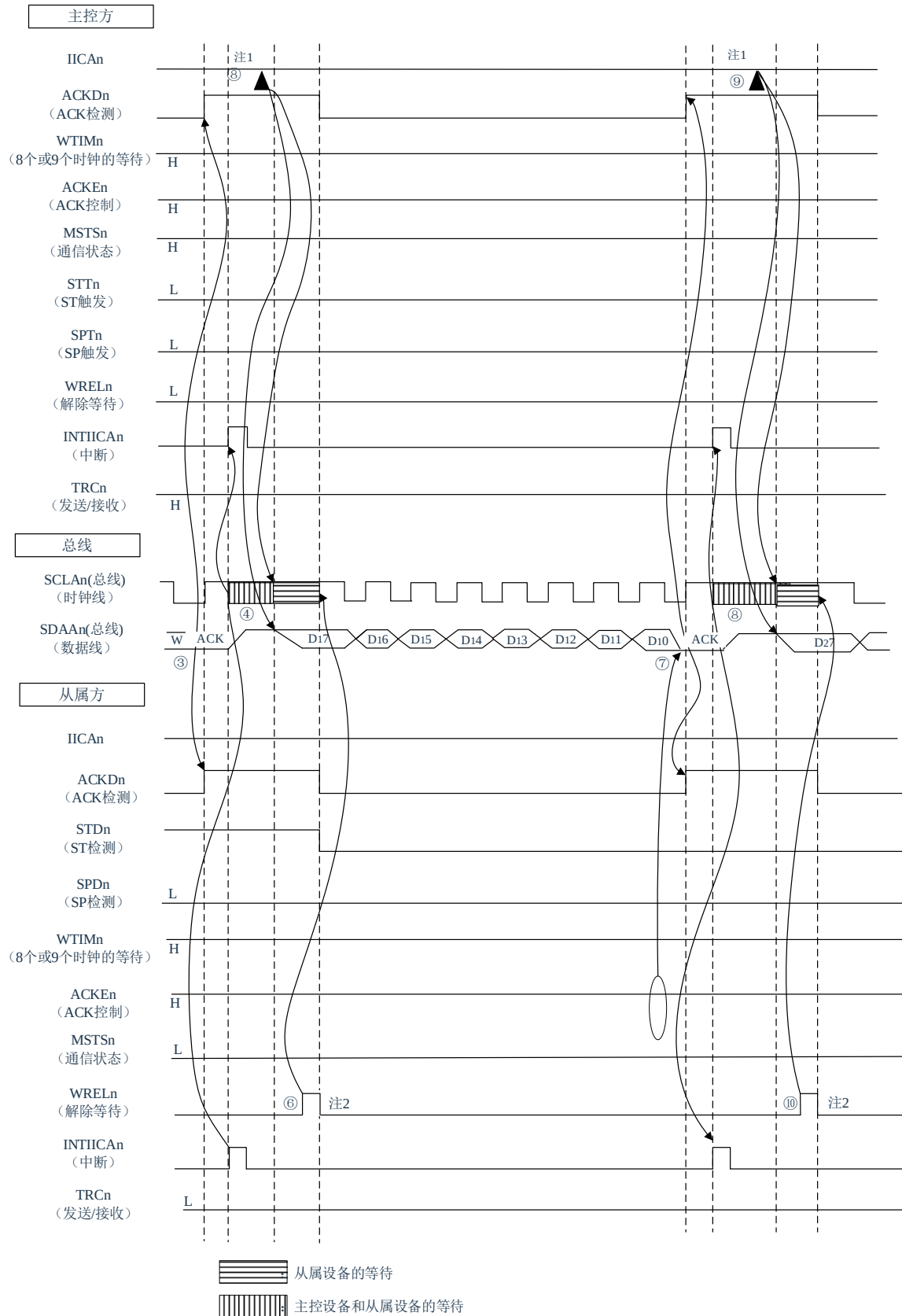
注5: 图14-21的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

注6: $n=0$ 。

图14-21 主控设备&从属设备的通信例子

(主控设备：选择9个时钟的等待，从属设备：选择9个时钟的等待) (2/4)

(2) 地址~数据~数据



注1: 要解除主控方发送期间的等待时，必须给IICAn写数据而不是将WRELn位置位。

注2: 要解除从属方接收期间的等待时，必须将IICAn置“FFH”或者将WRELn位置位。

注3: $n=0$

图14-21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1: 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)。

注2: 图14-21的①~⑮ 是通过I²C总线进行数据通信的一系列运行步骤。

注3: 图14-21的“(1)地址~数据~数据”说明步骤①~⑥。

注4: 图14-21的“(2)地址~数据~数据”说明步骤③~⑩。

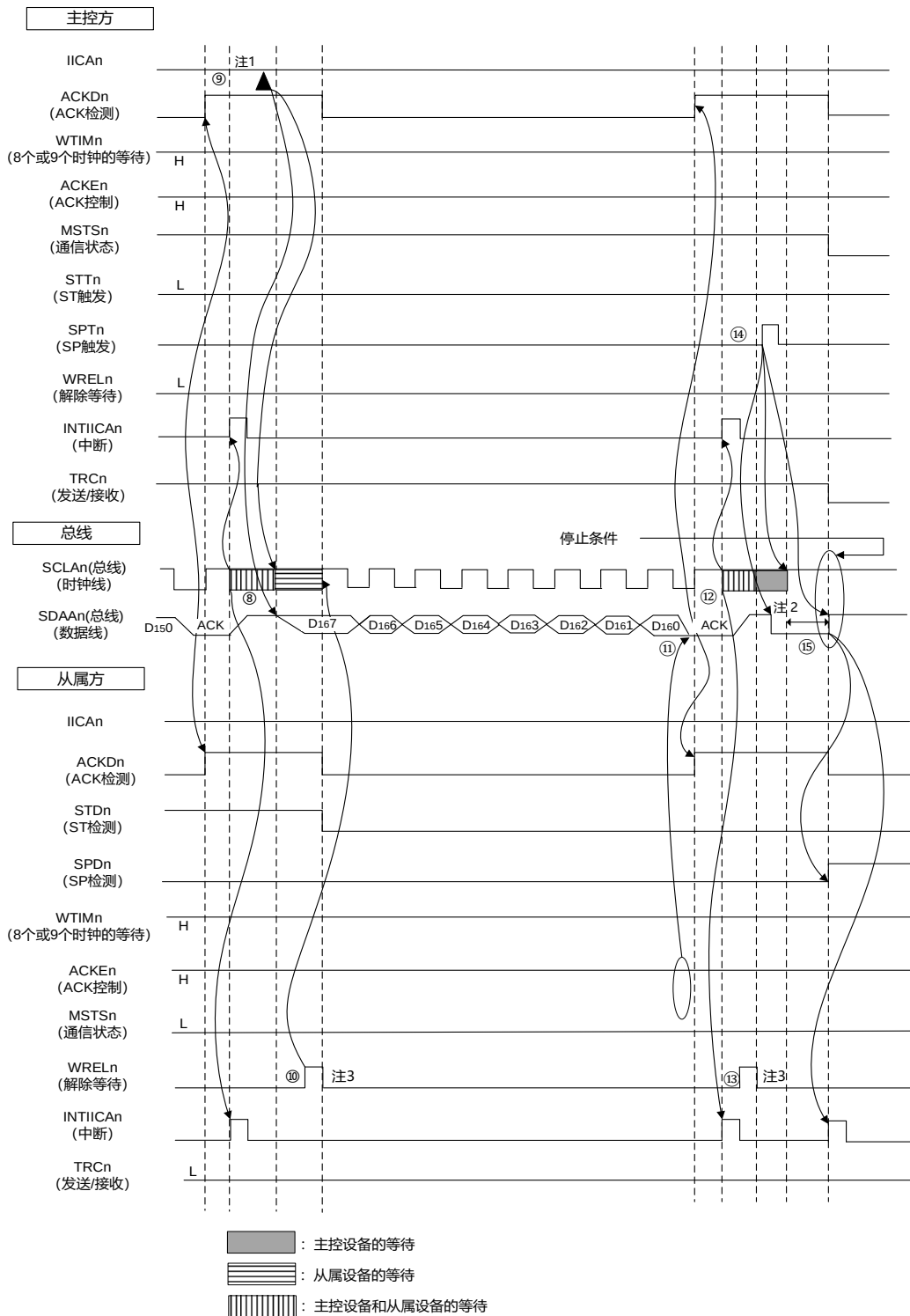
注5: 图14-21的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

注6: $n=0$ 。

图14-21 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)

数据~数据~停止条件



注1: 要解除主控方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注2: 在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的时, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us。

注3: 要解除从属方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

图14-21的“(3)数据~数据~停止条件”的⑦~⑮的说明如下

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 ($ACKDn=1$)。

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)。

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。

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)。

注1: 图14-21的①~⑮ 是通过I²C总线进行数据通信的一系列运行步骤。

注2: 图14-21的“(1)地址~数据~数据”说明步骤①~⑥。

注3: 图14-21的“(2)地址~数据~数据”说明步骤③~⑩。

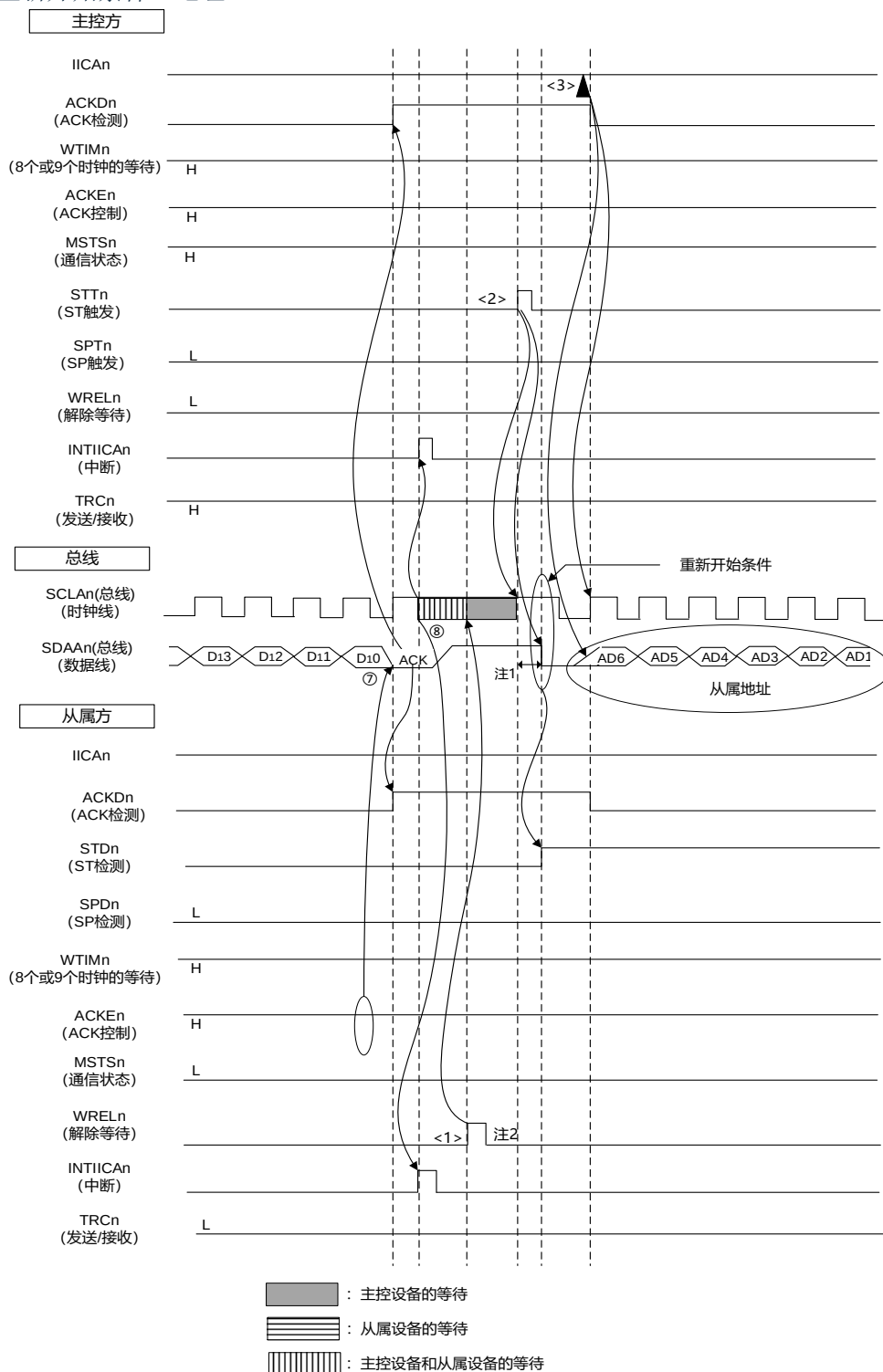
注4: 图14-21的“(3)数据~数据~停止条件”说明步骤⑦~⑮。

注5: $n=0$ 。

图14-21 主控设备&从属设备的通信例子

(主控设备: 选择9个时钟的等待, 从属设备: 选择9个时钟的等待) (4/4)

(4) 数据～重新开始条件～地址



注1: 在发行重新开始条件后, 从SCLAn引脚信号上升到生成开始条件的时间, 在设定为标准模式时至少为4.7us, 在设定为快速模式时至少为0.6us。

注2: 要解除从属方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注3: $n=0$

图14-21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。

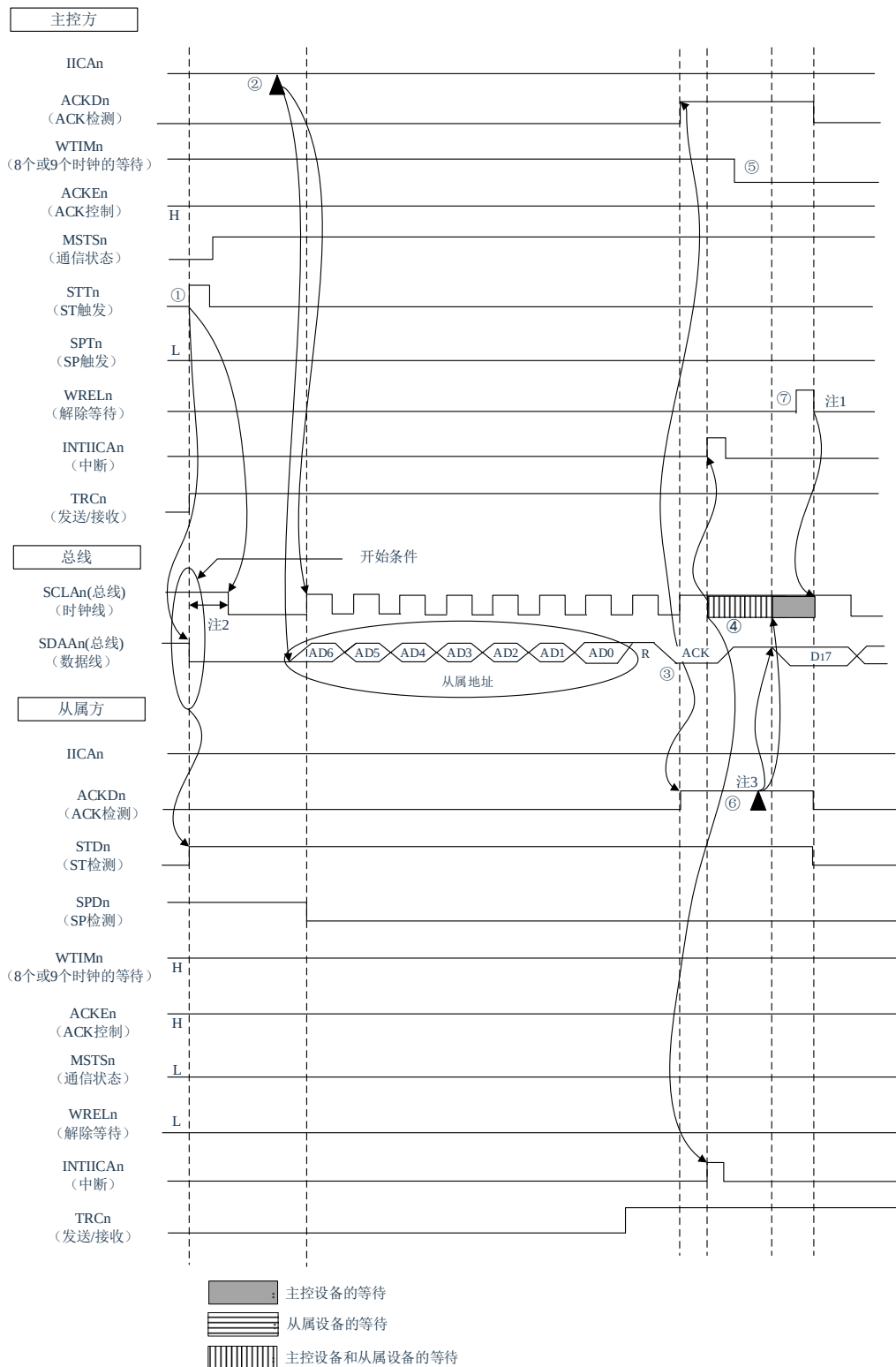
<3>如果主控方给IICA移位寄存器n (IICAn) 写地址+R/W (发送), 就发送从属地址。

注: n=0

图14-22 从属设备&主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (1/3)

(1) 开始条件~地址~数据



注1: 要解除主控方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注2: 从SDAAn引脚信号下降到SCLAn引脚信号下降的时间, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us。

注3: 要解除从属方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

图14-22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1：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)。

注2：图14-22的①~⑱是通过I2C总线进行数据通信的一系列运行步骤。

注3：图14-22的“(1)开始条件~地址~数据”说明步骤①~⑦。

注4：图14-22的“(2)地址~数据~数据”说明步骤③~⑫。

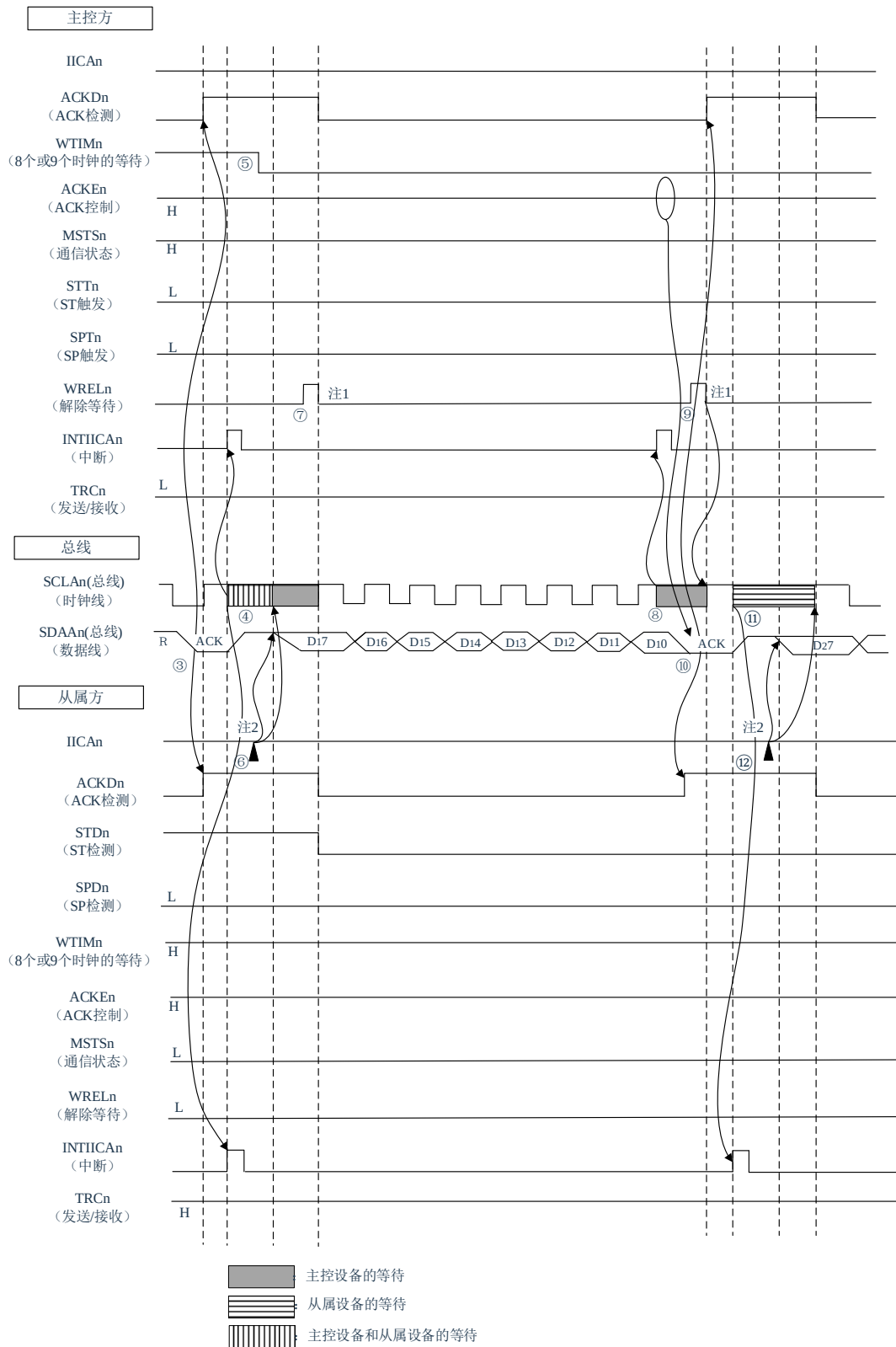
注5：图14-22的“(3)数据~数据~停止条件”说明步骤⑧~⑲。

注6： $n=0$ 。

图14-22 从属设备&主控设备的通信例子

(主控设备：选择8个时钟的等待，从属设备：选择9个时钟的等待) (2/3)

(2) 地址~数据~数据



注1: 要解除主控方接收期间的等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注2: 要解除从属方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

图14-22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1: 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: SDAAAn=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)。

注2: 图14-22的①~⑭是通过I2C总线进行数据通信的一系列运行步骤。

注3: 图14-22的“(1)开始条件~地址~数据”说明步骤①~⑦。

注4: 图14-22的“(2)地址~数据~数据”说明步骤③~⑫。

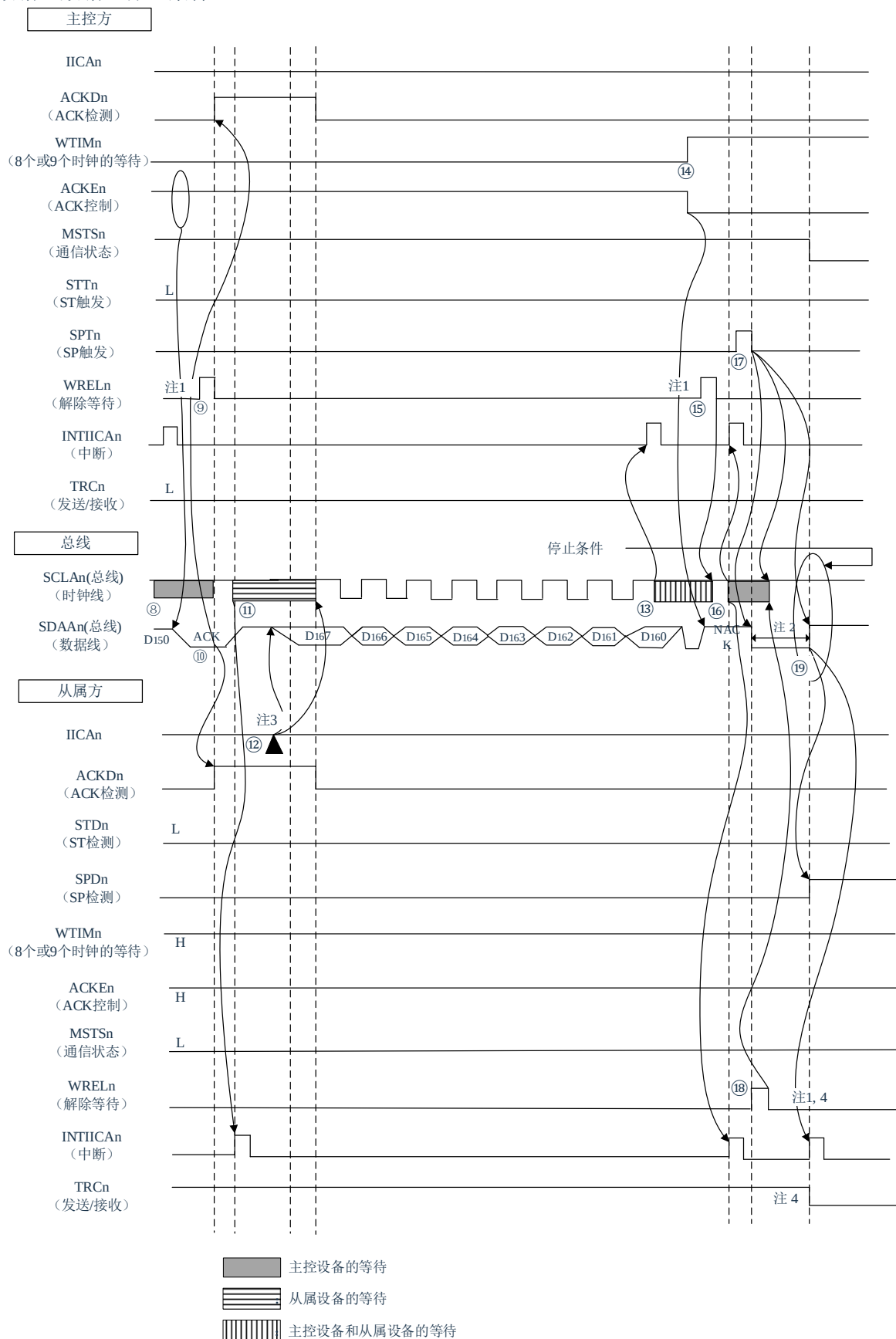
注5: 图14-22的“(3)数据~数据~停止条件”说明步骤⑧~⑮。

注6: n=0。

图14-22 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)

数据~数据~停止条件



注1: 要解除等待时, 必须将IICAn置“FFH”或者将WRELn位置位。

注2: 在发行停止条件后, 从SCLAn引脚信号上升到生成停止条件的時間, 在设定为标准模式时至少为4.0us, 在设定为快速模式时至少为0.6us。

注3: 要解除从属方发送期间的等待时, 必须给IICAn写数据而不是将WRELn位置位。

注4: 在从属方的发送期间, 如果通过WRELn位的置位来解除等待, 就清除TRCn位。

图14-22的“(3)数据~数据~停止条件”的⑧~⑲的说明如下:

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。

主控方读接收数据, 解除等待 (WRELn=1)。

从属方在第9个时钟的上升沿检测到ACK (ACKDn=1)。

⑪从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。

⑫如果从属方给IICA移位寄存器n (IICAn) 写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。

⑬主控方在第8个时钟的下降沿产生中断 (INTIICAn: 传送结束中断), 并且进入等待状态 (SCLAn=0)。

因为进行ACK控制 (ACKEn=1), 所以此阶段的总线数据线变为低电平 (SDAAn=0)。

⑭主控方设定为NACK应答 (ACKEn=0), 并且将等待时序改为第9个时钟 (WTIMn=1)。如果主控方解除等待 (WRELn=1), 从属方就在第9个时钟的上升沿检测到NACK (ACKDn=0)。

⑮主控方和从属方在第9个时钟的下降沿进入等待状态 (SCLAn=0), 并且都产生中断 (INTIICAn: 传送结束中断)。

⑯如果主控方发行停止条件 (SPTn=1), 就清除总线数据线 (SDAAn=0) 并且解除主控方的等待。此后, 主控方处于待机状态, 直到将总线时钟线置位 (SCLAn=1) 为止。

⑰从属方在确认NACK后停止发送, 为了结束通信, 解除等待 (WRELn=1)。如果解除从属方的等待, 就将总线时钟线置位 (SCLAn=1)。

⑱如果主控方确认到总线时钟线被置位 (SCLAn=1), 就在经过停止条件准备时间后将总线数据线置位

⑲ (SDAAn=1) 然后发行停止条件 (通过SCLAn=1使SDAAn从“0”变为“1”)。如果生成停止条件, 从属方就检测到停止条件, 并且产生中断 (INTIICAn: 停止条件中断)

第15章 增强型DMA

15.1 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15-1所示。

表15-1 DMA的规格(1/2)

项目		规格
启动源		同时最大8个启动源
可分配的控制数据		8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
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15-4DMA启动源和向量地址

表15-1 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	- 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	- 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

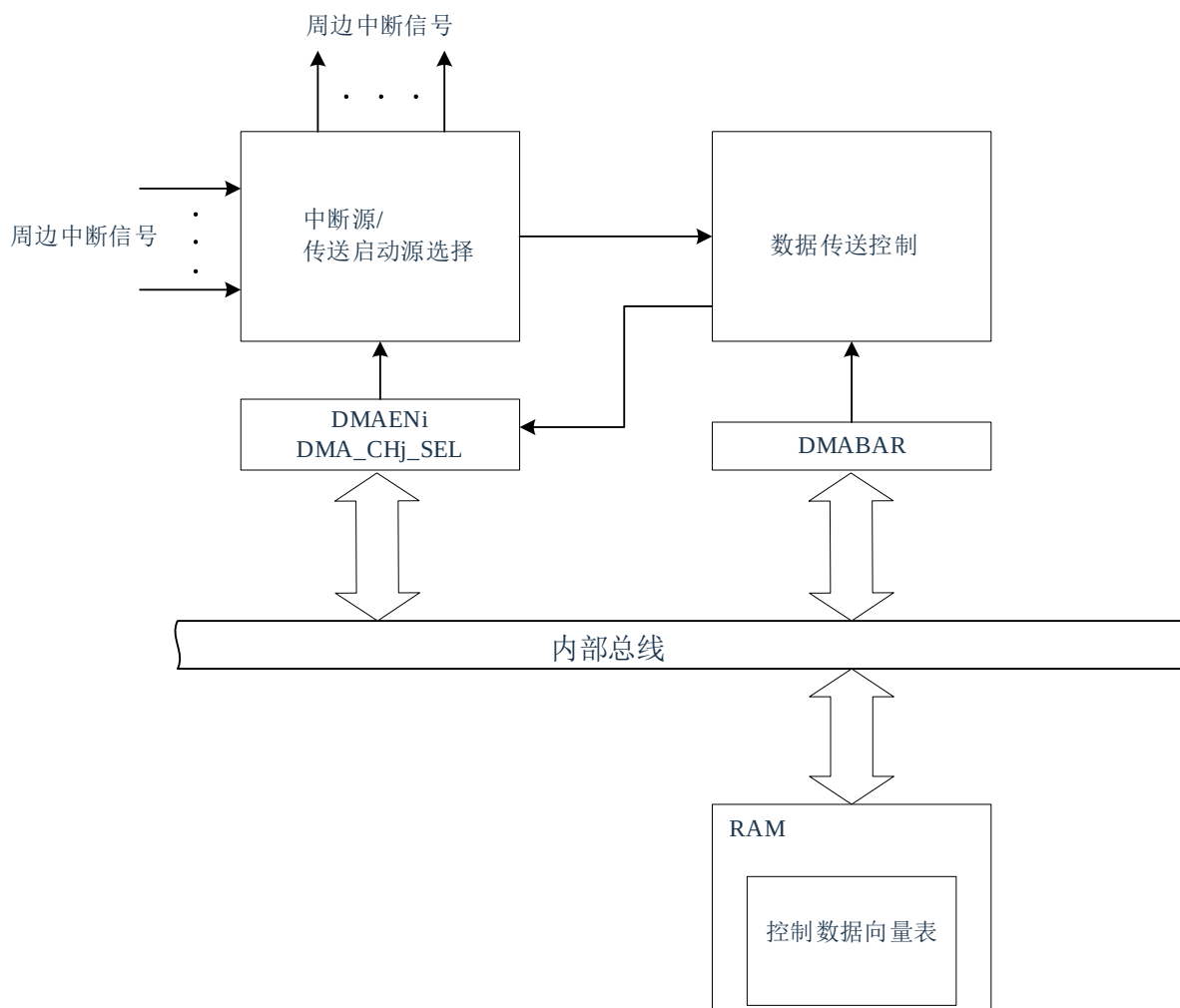
注1：在深度睡眠模式，因为闪存停止运行，所以不能作为DMA传送源。

注2：i=0，j=0~7。

15.2 DMA的结构

DMA的框图如图15-1所示

图15-1 DMA的框图



15.3 寄存器映射

(以下寄存器基地址 = 0x40020810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PER10	0x000	R/W	外围允许寄存器10	0x00

(以下寄存器基地址 = 0x40005000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMAEN0	0x000	R/W	DMA启动允许寄存器0	0x00
DMABAR	0x008	R/W	DMA基址寄存器	0x00
DMASET0	0x018	WO	DMAEN0置位寄存器	0x00
DMACLR0	0x020	WO	DMAEN0复位寄存器	0x00

(以下寄存器基地址 = 0x40020D00) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
DMA_CH0_SEL	0x000	R/W	DMA通道0触发源选择寄存器	0x3F
DMA_CH1_SEL	0x001	R/W	DMA通道1触发源选择寄存器	0x3F
DMA_CH2_SEL	0x002	R/W	DMA通道2触发源选择寄存器	0x3F
DMA_CH3_SEL	0x003	R/W	DMA通道3触发源选择寄存器	0x3F
DMA_CH4_SEL	0x004	R/W	DMA通道4触发源选择寄存器	0x3F
DMA_CH5_SEL	0x005	R/W	DMA通道5触发源选择寄存器	0x3F
DMA_CH6_SEL	0x006	R/W	DMA通道6触发源选择寄存器	0x3F
DMA_CH7_SEL	0x007	R/W	DMA通道7触发源选择寄存器	0x3F

DMA的控制数据如表15-2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144字节区域。

表15-2 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	DMBSLj
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

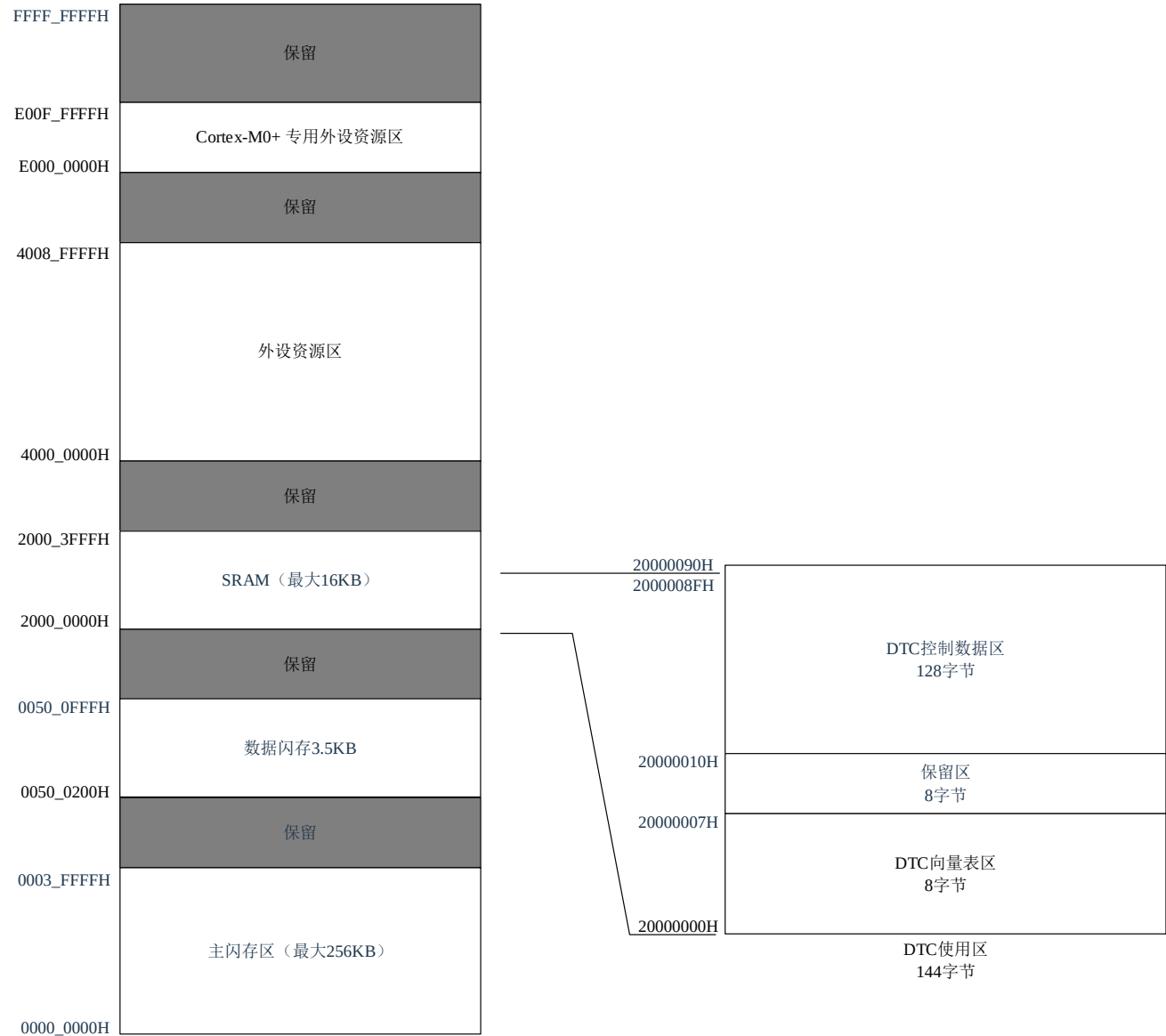
注: j=0~7。

15.4 控制DMA的寄存器

15.4.1 DMA控制数据区和DMA向量表区的分配

通过DMABAR寄存器将分配DMA的控制数据和向量表的144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15-2所示。
DMA控制数据区的144字节中DMA不使用的空间能用作RAM。

图15-2 DMABAR寄存器的设置值为“20000000H”时的存储器映像例子



15.4.2 控制数据的分配

从起始地址开始，按照DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj（j=0~7）寄存器的顺序分配控制数据。

起始地址由DMABAR寄存器设置，低8位由各启动源分配的向量表分别设置。

每个通道各自的触发源由DMA_CHj_SEL（j=0~7）寄存器设置。

控制数据的分配如图15-3所示。

注1：必须在对应的DMAENi（i=0）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。

注2：不能通过DMA传送进行DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj和DMDARj的存取。

图15-3 控制数据的分配(DMABAR设置为20000000H)

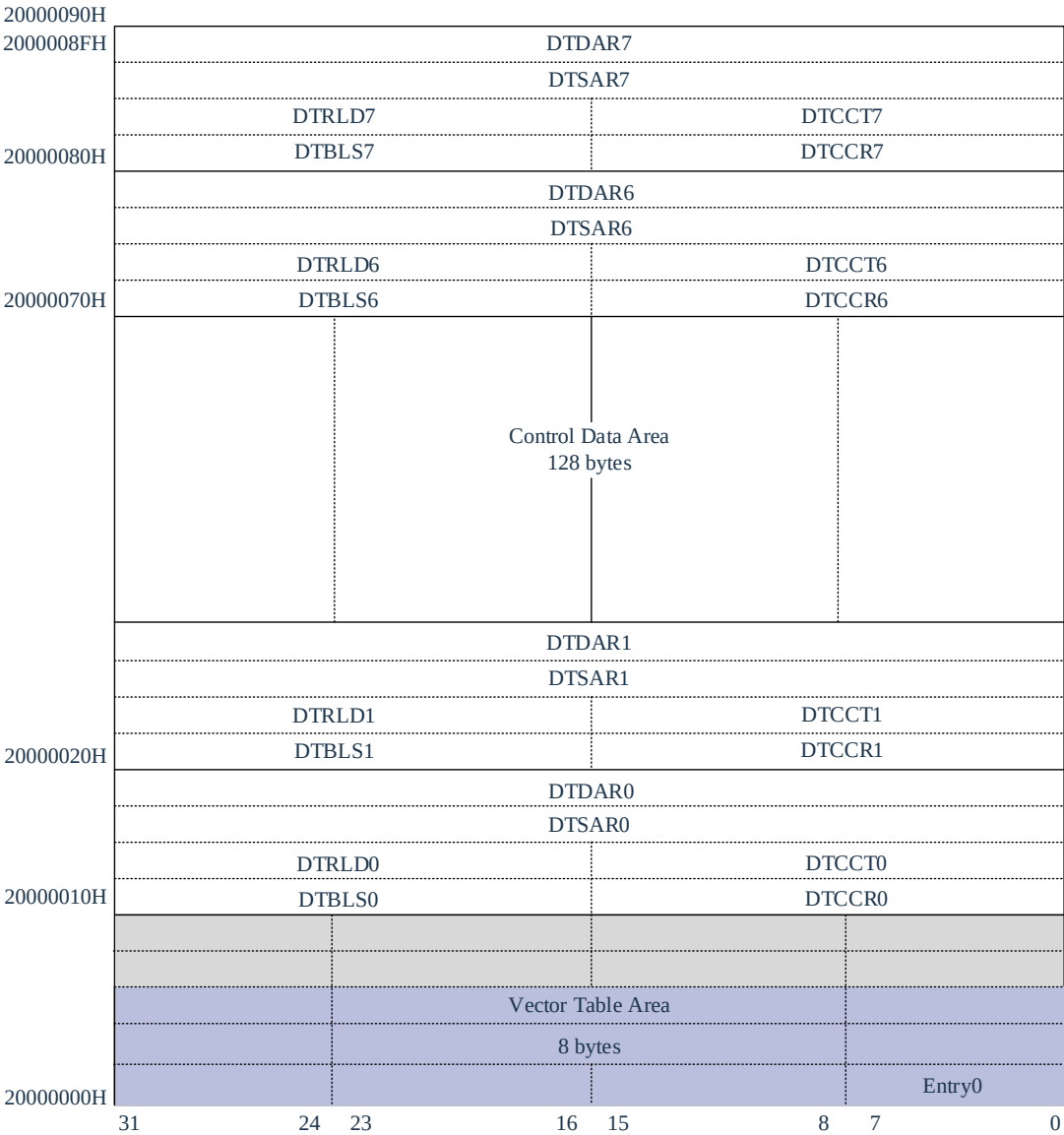


表15-3 控制数据的起始地址

j	地址
7	baseaddr+80H
6	baseaddr+70H
5	baseaddr+60H
4	baseaddr+50H
3	baseaddr+40H
2	baseaddr+30H
1	baseaddr+20H
0	baseaddr+10H

注：baseaddr：DMABAR寄存器的设置值。

15.4.3 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15-4所示。各启动源的向量表有1字节，保存“00H”~“07H”的数据，从8组的控制数据中选择1组数据。向量地址的高24位由DMABAR寄存器设置，低8位被分配了对应启动源的“00H”~“07H”。

注：必须在对应的DMAENi（i=0）寄存器的DMAENi0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15-4 控制数据的起始地址和向量表
DMABAR寄存器的设置值为“20000000H”的情况（例）

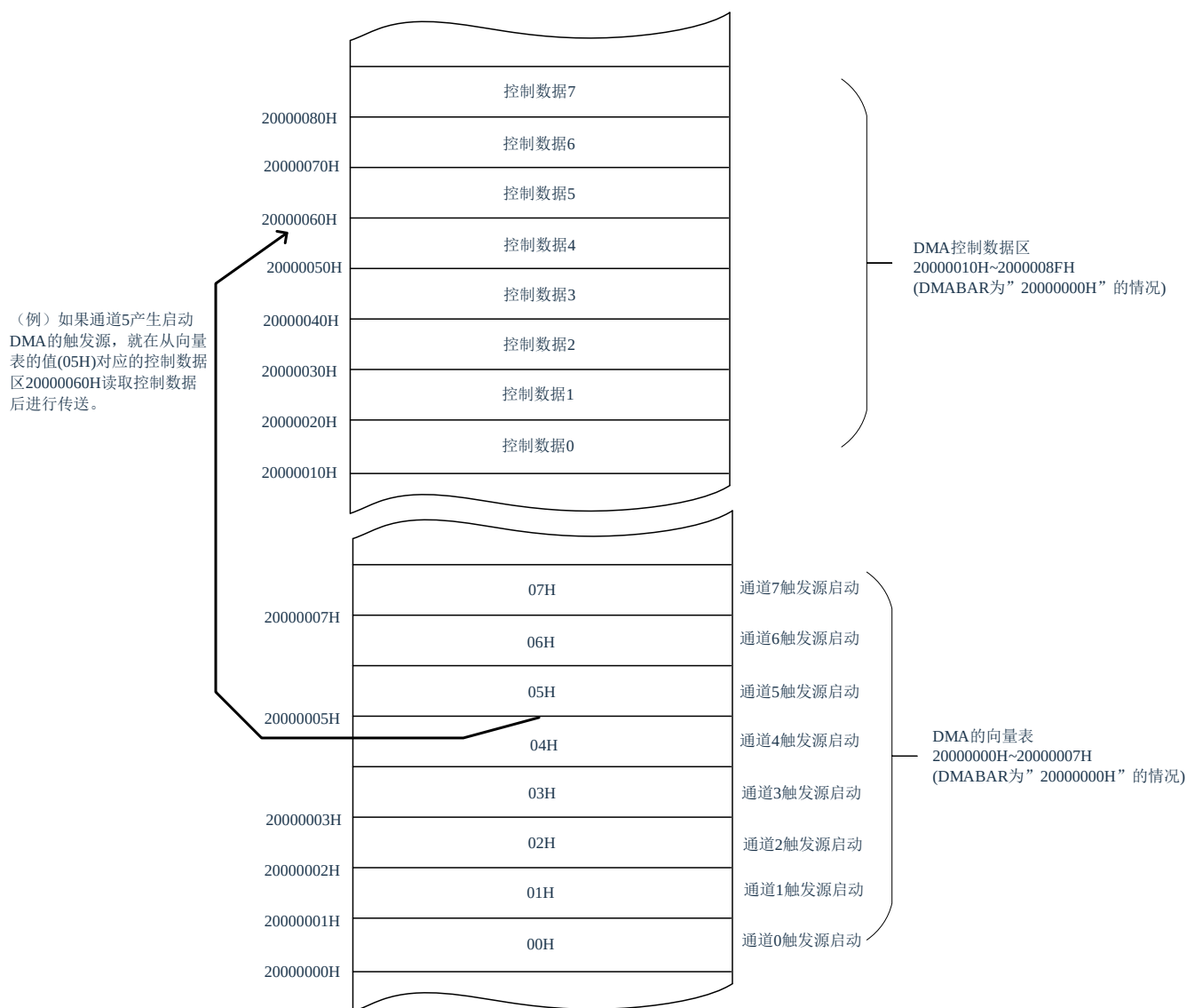


表15-4 DMA启动源和向量地址

DMA启动源（中断请求发生源）	源号	向量地址	优先级
通道0触发源启动	0	DMABAR寄存器的设置地址+00H	高 ↑ ↓ 低
通道1触发源启动	1	DMABAR寄存器的设置地址+01H	
通道2触发源启动	2	DMABAR寄存器的设置地址+02H	
通道3触发源启动	3	DMABAR寄存器的设置地址+03H	
通道4触发源启动	4	DMABAR寄存器的设置地址+04H	
通道5触发源启动	5	DMABAR寄存器的设置地址+05H	
通道6触发源启动	6	DMABAR寄存器的设置地址+06H	
通道7触发源启动	7	DMABAR寄存器的设置地址+07H	

15.4.4 外围允许寄存器10（PER10）

PER10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PER10寄存器。在产生复位信号后，此寄存器的值变为“00H”。

外围允许寄存器10（PER10）

位	符号	描述	复位值
7:4	-	保留	0
3	DMAEN	提供DMA模块输入时钟的控制 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	0
2	EPWMEN	提供EPWM模块输入时钟的控制 0: 停止提供输入时钟，EPWM不能运行 1: 提供输入时钟，EPWM能运行	0
1:0	-	保留	0

15.4.5 DMA通道触发源选择寄存器j（DMA_CHj_SEL）（j=0~7）

位	符号	描述	复位值
7:6	-	保留	0x0
5:0	TRGSEL	DMA通道 j 选择触发源(j=0~7): 0x00: INTP0 0x01: INTP1 0x02: INTP2 0x03: INTP3 0x04: INTP4 0x05: INTP5 0x06: INTP6 0x07: INTP7 0x08: INTP8 0x09: INTP9 0x0A: INTP10 0x0B: INTP11 0x0C: A/D转换结束 0x0D: UART0发送的传送结束/CSI00的传送结束或者缓冲器空/IIC00的传送结束 0x0E: UART0接收的传送结束/CSI01的传送结束或者缓冲器空/IIC01的传送结束 0x0F: UART1发送的传送结束/CSI10的传送结束或者缓冲器空/IIC10的传送结束 0x10: UART1接收的传送结束/CSI11的传送结束或者缓冲器空/IIC11的传送结束 0x11: UART2发送的传送结束/CSI20的传送结束或者缓冲器空/IIC20的传送结束 0x12: UART2接收的传送结束/CSI21的传送结束或者	0x3F

		缓冲器空/IIC21的传送结束 0x13: UART3发送的传送结束 0x14: UART3接收的传送结束 0x15: UART4发送的传送结束 0x16: UART4接收的传送结束 0x17: UART5发送的传送结束 0x18: UART5接收的传送结束 0x19: IICA0通信结束。 0x1A: Timer40的通道0的计数或捕捉结束 0x1B: Timer40的通道1的计数或捕捉结束 0x1C: Timer40的通道2的计数或捕捉结束 0x1D: Timer40的通道3的计数或捕捉结束 0x1E: Timer41的通道0的计数或捕捉结束 0x1F: Timer41的通道1的计数或捕捉结束 0x20: Timer41的通道2的计数或捕捉结束 0x21: Timer41的通道3的计数或捕捉结束 0x22: 15位间隔定时器产生计数中断 0x23: Flash读写擦除结束 0x24: 比较器检测0 0x25: 比较器检测1 0x26: INT LED 0x27: INT LCD 0x28: INT TOUCH 其他值: 禁止设置	

15.4.6 DMA控制寄存器j (DMACRj) (j=0~7)

DMACRj寄存器控制DMA的运行模式。

地址：参照“15.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:8	-	保留	不定值
7: 6	SZ	传送数据长度的选择 00: 8位 01: 16位 10: 32位 11: 禁止设置	
5	RPTINT	重复模式中中断的允许/禁止 ^{注1} 0: 禁止发生中断。 1: 允许发生中断。	
4	CHNE	链传送的允许/禁止 ^{注2} 0: 禁止链传送。 1: 允许链传送。	
3	DAMOD	传送目标地址的控制 ^{注3} 0: 固定 1: 递增	
2	SAMOD	传送源地址的控制 ^{注4} 0: 固定 1: 递增	
1	RPTSEL	重复区的选择 ^{注5} 0: 传送目标为重复区。 1: 传送源为重复区。	
0	MODE	传送模式的选择 0: 正常模式 1: 重复模式	

注1：在MODE位为“0”（正常模式）时，RPTINT位的设置无效。

注2：必须将DMACR7寄存器的CHNE位置“0”（禁止链传送）。

注3：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。

注4：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。

注5：在MODE位为“0”（正常模式）时，RPTSEL位的设置无效。

注6：不能通过DMA传送进行DMACRj寄存器的存取。

15.4.7 DMA块大小寄存器j (DMBLSj) (j=0~7)

此寄存器设置1次启动传送数据的块大小。

地址：参照“15.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMBLSj	设置1次启动传送数据的块大小	不定值

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1：不能通过DMA传送进行DMBLSj寄存器的存取。

15.4.8 DMA传送次数寄存器j (DMACTj) (j=0~7)

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。

地址：参照“15.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMACTj	设置DMA的数据传送次数，每当启动1次DMA传动就减1	不定值

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1：不能通过DMA传送进行DMACTj寄存器的存取。

15.4.9 DMA传送次数重加载寄存器j (DMRLDj) (j=0~7)

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。

地址：参照“15.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
15:0	DMRLDj	设置重复模式中的传送次数寄存器的初始值	不定值

注：不能通过DMA传送进行DMRLDj寄存器的存取。

15.4.10 DMA源地址寄存器j (DMSARj) (j=0~7)

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。

地址：参照“15.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMSARj	设置数据传送时的传送源地址	不定值

注：不能通过DMA传送进行DMSARj寄存器的存取

15.4.11 DMA目标地址寄存器j (DMDARj) (j=0~7)

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。

地址：参照“15.4.2.4.2 控制数据的分配”。 复位后：不定值 R/W

位	符号	描述	复位值
31:0	DMDARj	设置数据传送时的目标地址	不定值

注：不能通过DMA传送进行DMDARj寄存器的存取。

15.4.12 DMA启动允许寄存器i（DMAEN0）

这是控制允许或者禁止通过各通道启动DMA的8位寄存器。通道和DMAEN0位的对应如表15-5所示。
能通过8位存储器操作指令设置DMAEN寄存器。

注1：必须在不产生对应该位的启动源的位置更改DMAENi0～DMAENi7位。

注2：不能通过DMA传送进行DMAENi寄存器的存取。

注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”。

位	符号	描述	复位值
7:0	DMAEN0n (n=0～7)	DMA启动的允许in ^{注1} (n=0～7) 0: 禁止启动。 1: 允许启动。	0x0

注1：根据传送结束中断的发生条件，DMAENin位变为“0”（禁止启动）。n=0,1,2,3,4,5,6,7。

表15-5 通道和DMAEN00～DMAEN07位的对应

寄存器	DMAENi7位	DMAENi6位	DMAENi5位	DMAENi4位	DMAENi3位	DMAENi2位	DMAENi1位	DMAENi0位
DMAEN0	通道7允许启动	通道6允许启动	通道5允许启动	通道4允许启动	通道3允许启动	通道2允许启动	通道1允许启动	通道0允许启动

注1：必须将没有分配功能的位置“0”。

注2：i=0

15.4.13 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。

注2：只能改写1次DMABAR寄存器。

注3：不能通过DMA传送进行DMABAR寄存器的存取。

注4：有关DMA控制数据区和DMA向量表区的分配，请参照“15.4.1 DMA控制数据区和DMA向量表区的分配”的说明。

注5：设置该寄存器请保持256Byte对齐，也就是低8位设置为零。DMA硬件忽略低8位。

注6：该寄存器只能WORD访问，BYTE和HALFWORD访问忽略。

位	符号	描述	复位值
31:8	DMABARj	设置数据传送时的目标地址高24位	0x0
7:0	DMABARj	设置数据传送时的目标地址低8位，因256字节对齐，故设置为0	0x0

注：j=0

15.4.14 DMAEN0置位寄存器（DMASET0）

这是DMA启动允许寄存器DMAEN0的置位寄存器，设置相应的位为1则可以将DMAEN0的相应位置位为1。

位	符号	描述	复位值
31:4	--	保留，须为0	0x0
3:0	DMASET0n (n=0~3)	DMAEN0n的置位寄存器 (n=0~3) 0: 不产生置位。 1: 将DMAEN0的位n置位为1。	0x0

15.4.15 DMAEN0复位寄存器（DMACLR0）

这是DMA启动允许寄存器DMAEN0的复位寄存器，设置相应的位为1则可以将DMAEN0的相应位复位为0。

位	符号	描述	复位值
31:4	--	保留，须为0	0x0
3:0	DMACLR0n (n=0~3)	DMAEN0n的复位寄存器 (n=0~3) 0: 不产生复位。 1: 将DMAEN0的位n复位为0。	0x0

15.5 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8组控制数据保存到DMA控制数据区，并且进行8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=0~7) 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。

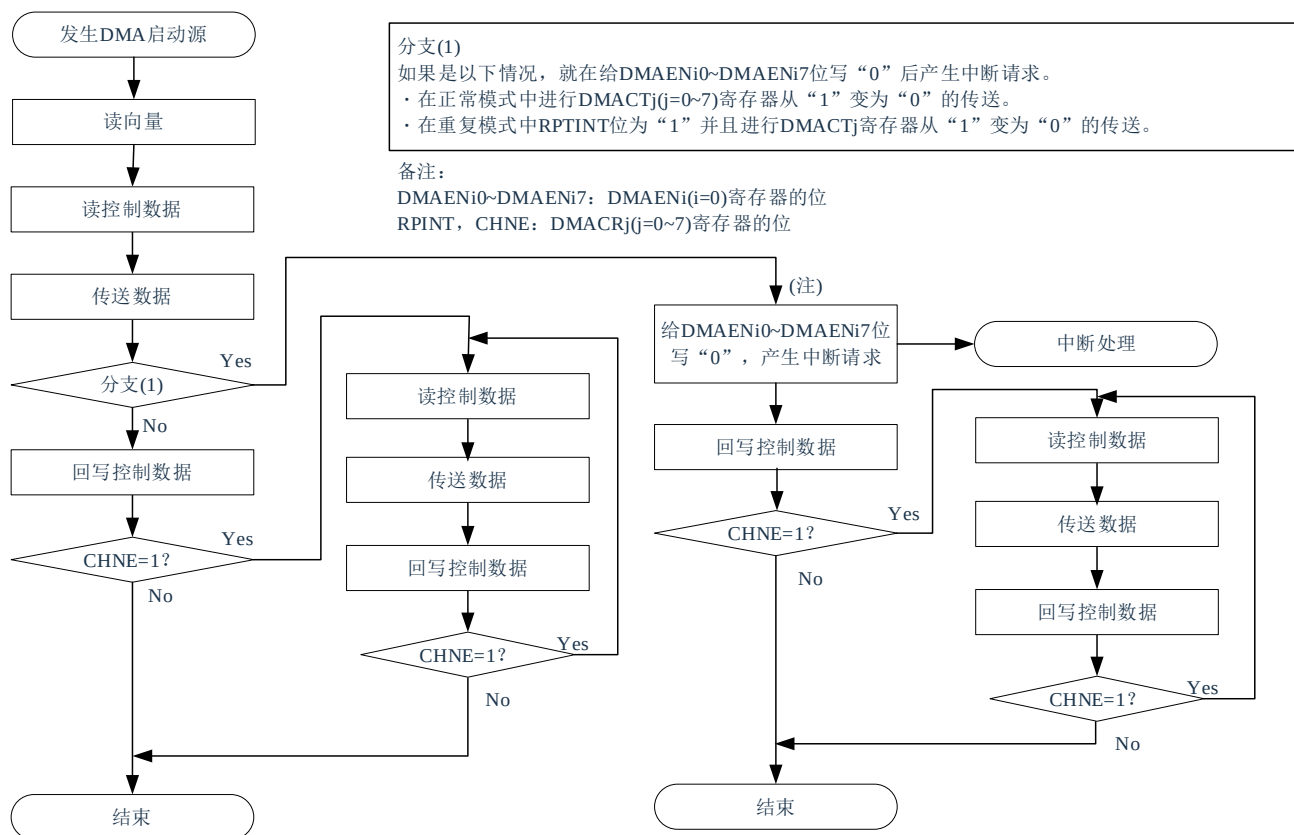
15.5.1 启动源

通过外围功能的中断信号启动DMA，并且通过DMAENi (i=0) 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=0~7) 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15-5所示。

图15-5 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”且不产生中断请求。

15.5.2 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=0~7）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）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15-6和图15-6所示

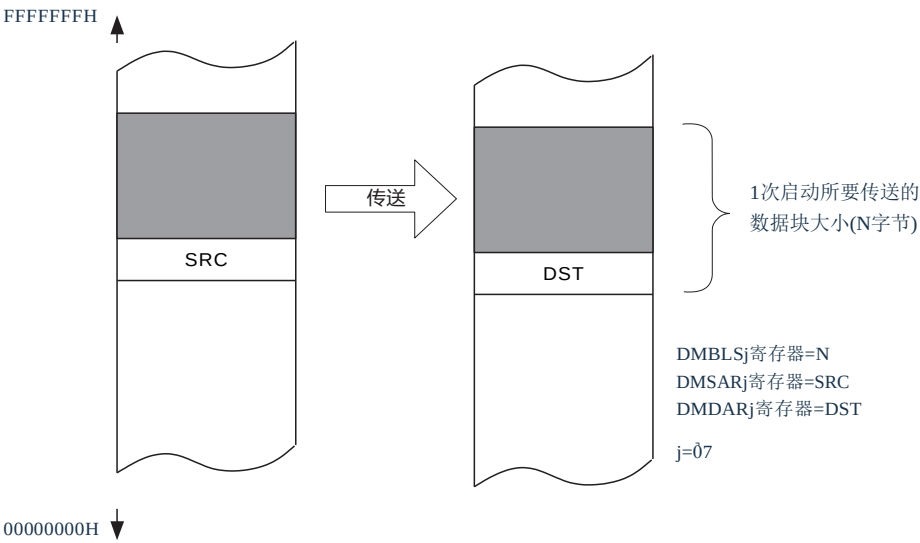
注：j=0

表15-6 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：j=0

图15-6 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

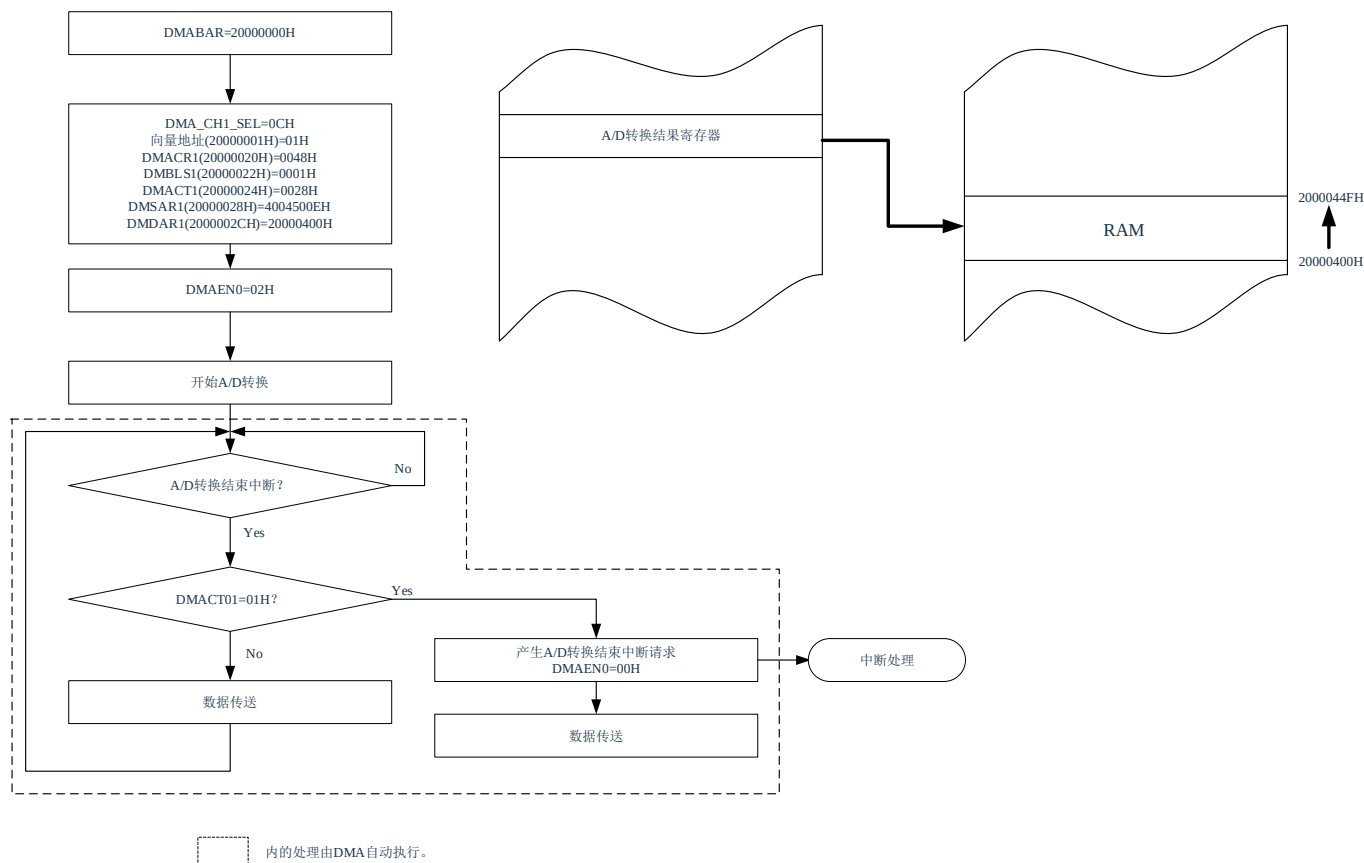
X：“0”或者“1”

(1) 正常模式的使用例子 1: 通道 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20000001H，控制数据分配在20000020H~2000002FH。
- 将A/D转换结果寄存器ADCR（4004500EH）的2字节数据传送40次到RAM的20000400H~2000044FH的80字节。

图15-7 正常模式的使用例子1: 连续取A/D转换结果

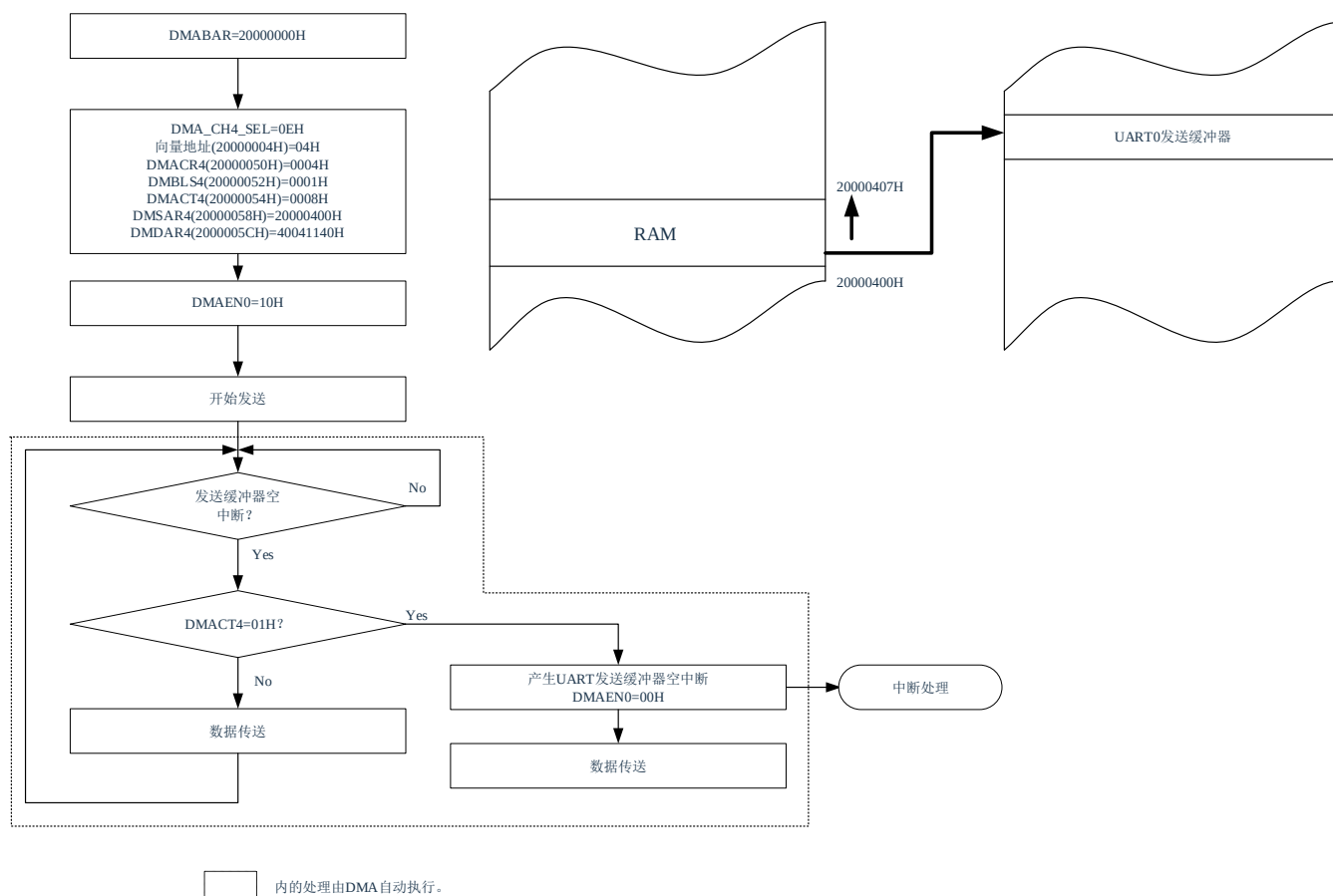


因为是正常模式，所以不使用DMRLD01寄存器的值。

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20000004H，控制数据分配在20000050H~2000005FH。
- 将RAM的20000400H~20000407H的8字节传送到UART0的发送缓冲器SDR00（40041140H）。

图15-8 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DMRLD04寄存器的值。必须通过软件开始第1次的UART0发送。通过发送传送结束启动DMA，然后自动进行第2次以后的发送。

15.5.3 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=0~7）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）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”。

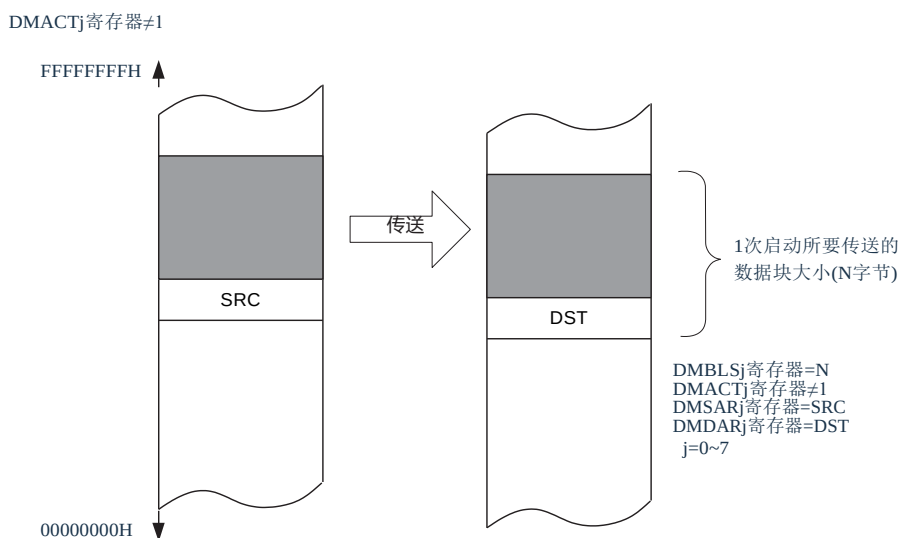
重复模式的寄存器功能和数据传送分别如表15-7和图15-9所示。

表15-7 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

注：j=0~7

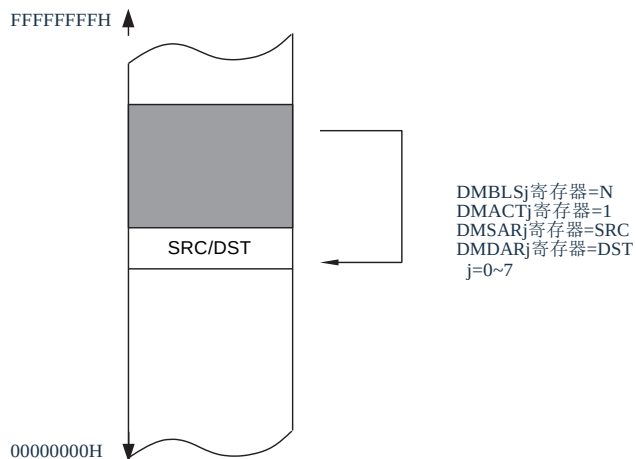
图15-9 重复模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC+N	DST
1	X	1	1	重复区	递增	SRC+N	DST+N
X	0	0	1	固定	重复区	SRC	DST+N
X	1	0	1	递增	重复区	SRC+N	DST+N

备注：X：“0”或者“1”

DMACTj寄存器=1



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	重复区	固定	SRC	DST
1	X	1	1	重复区	递增	SRC	DST+N
X	0	0	1	固定	重复区	SRC	DST
X	1	0	1	递增	重复区	SRC+N	DST

备注：X：“0”或者“1”

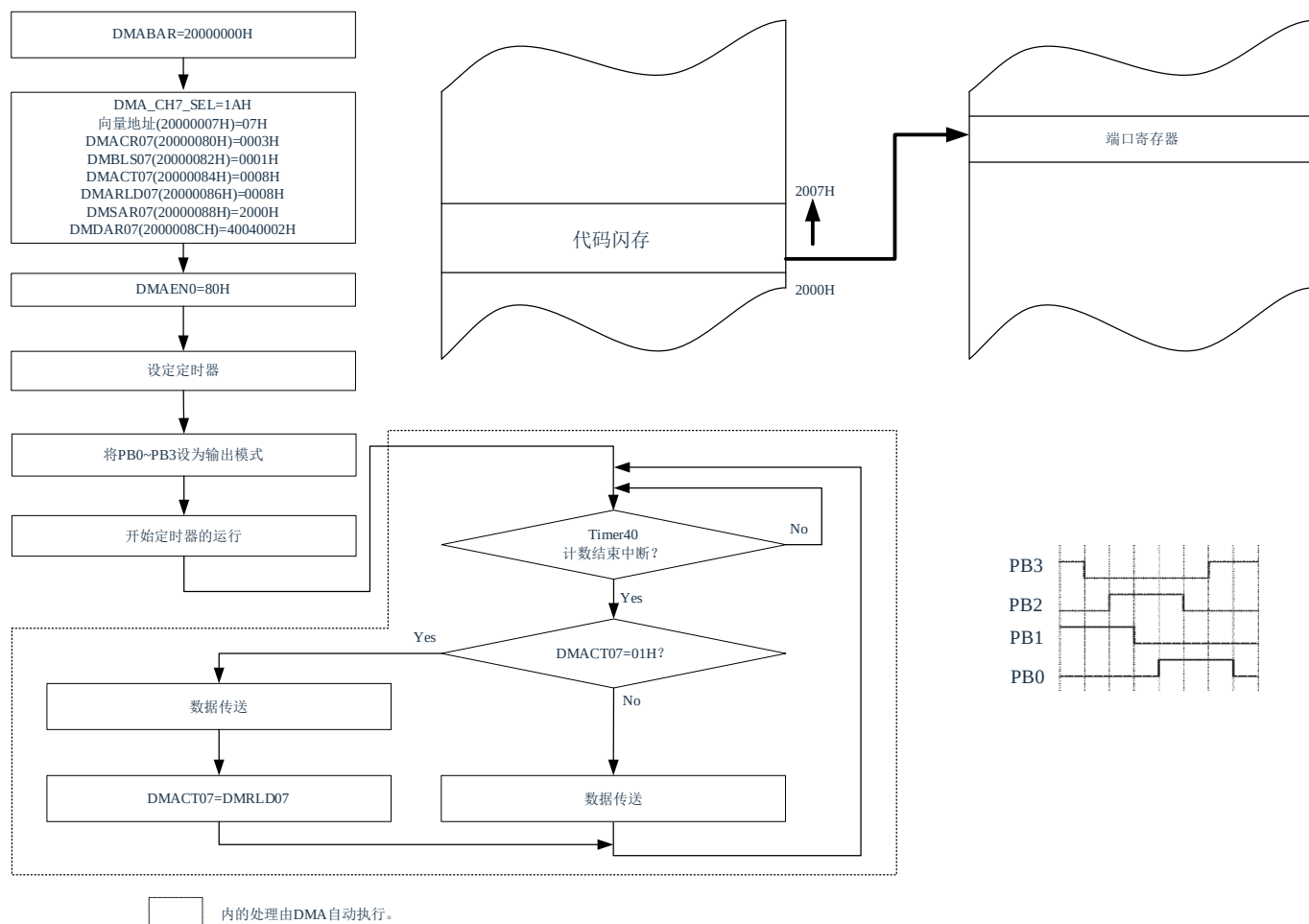
注：在使用重复模式时，必须将重复区的数据长度设置在65535字节以内。

(1) 重复模式的使用例子：通道7使用端口的步进马达控制脉冲输出。

通道7，当Timer40的通道0计数结束时，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PB(0x40040002H)。

- 向量地址分配在20000007H，控制数据分配在20000080H~2000008FH。
- 将代码闪存的02000H~02007H的8字节传送到端口寄存器1（40040002H）。
- 禁止重复模式中断。

图15-10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DMAEN0的bit7。

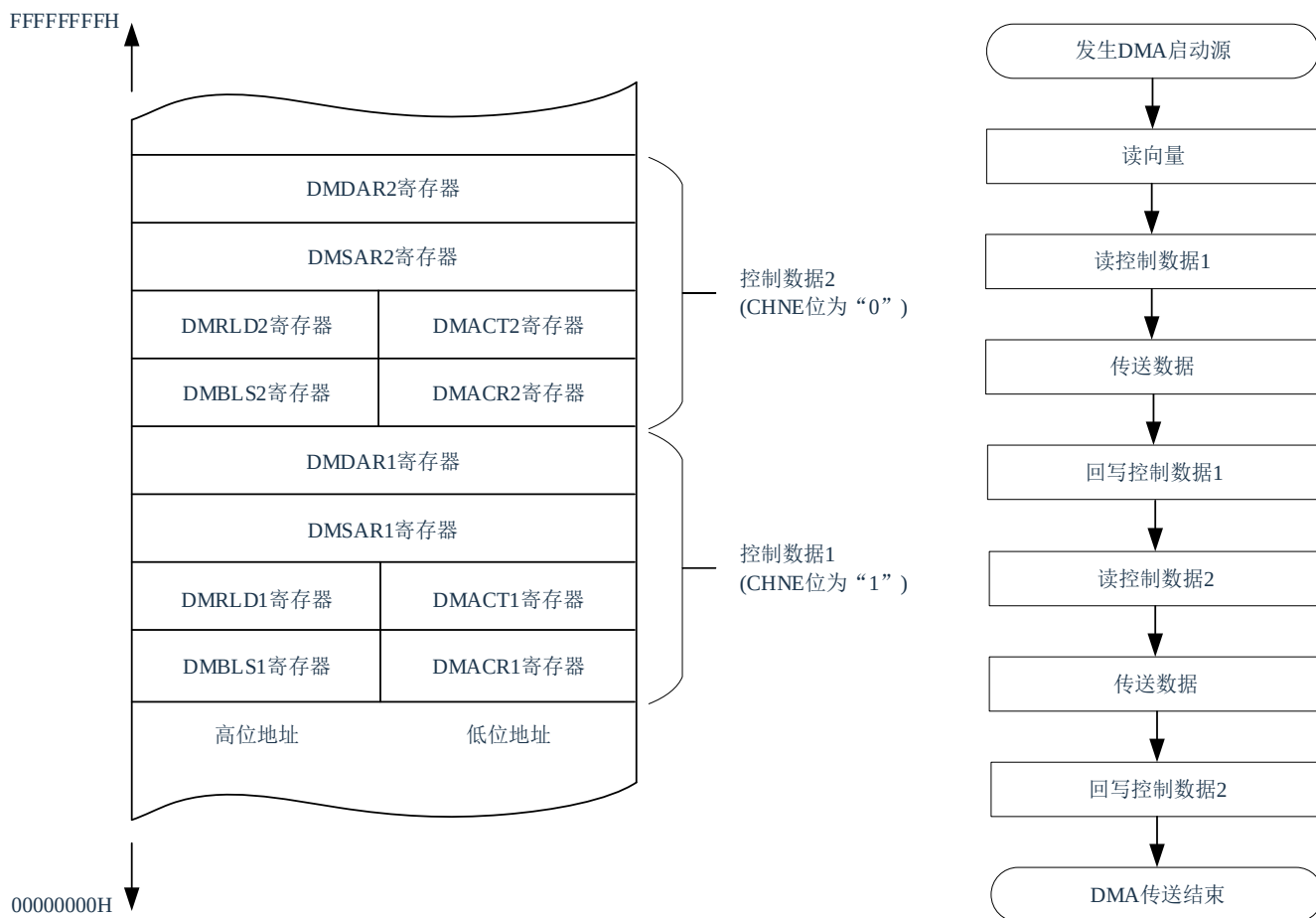
15.5.4 链传送

当DMACRj (j=0~7) 寄存器的CHNE位为“1 (允许链传送)”时，能通过1个通道的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。

链传送的流程图如图15-11所示。

图15-11 链传送的流程图



注1：必须将DMACR07寄存器的CHNE位置“0” (禁止链传送)。

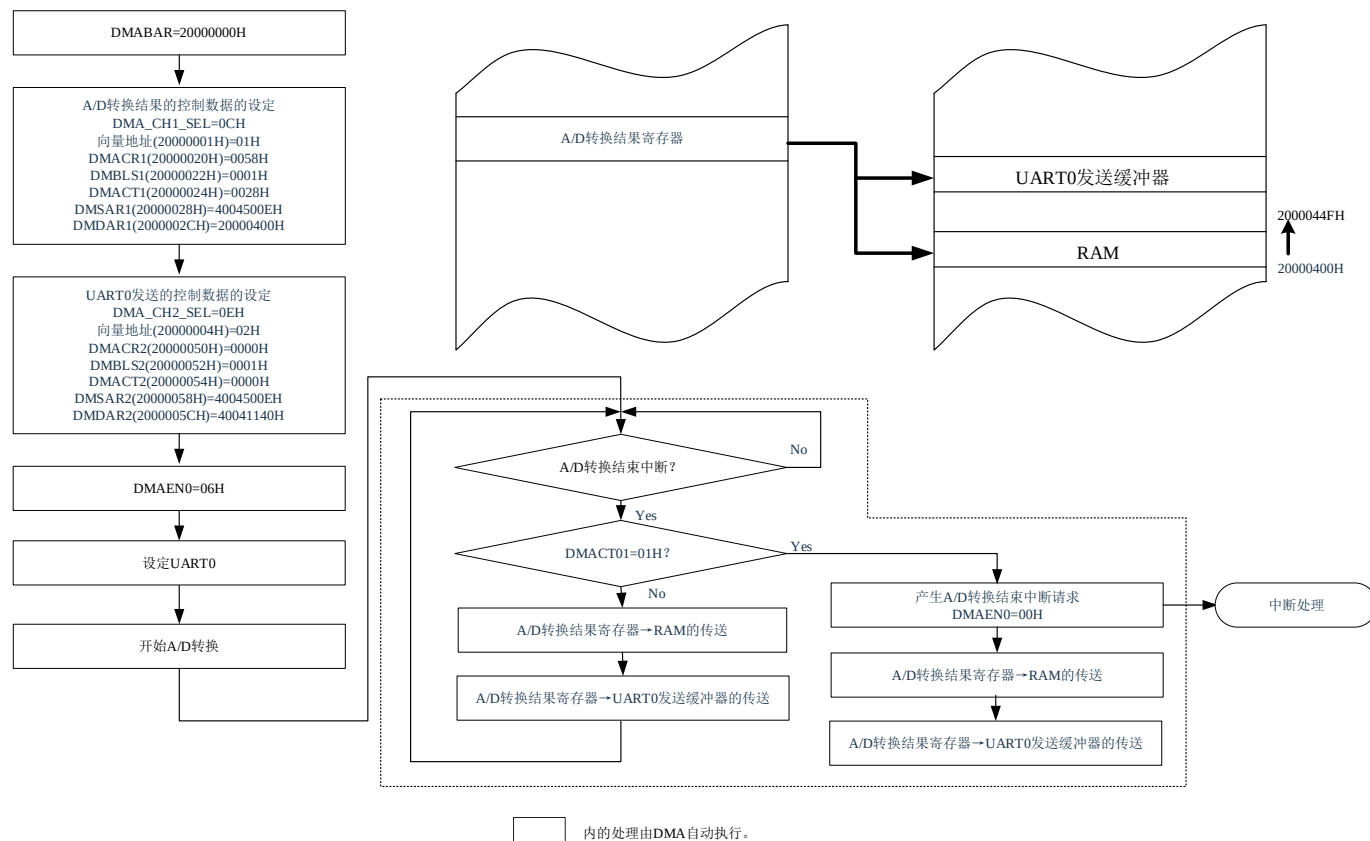
注2：在链传送的第2次以后的数据传送时，DMAENi (i=0) 寄存器的DMAENi0~DMAENi7位不变为“0” (禁止DMA启动)，并且不产生中断请求。

(1) 链传送的使用例子：连续取A/D转换结果进行UART0发送

通道1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20000001H和20000002H。
- A/D转换结果的控制数据分配在20000020H~2000002FH。
- UART0发送的控制数据分配在20000030H~2000003FH。
- 将A/D转换结果寄存器ADCR（4004500EH）的2字节数据传送到RAM的20000400H~2000044FH，并且/D转换结果寄存器的低位1字节（4004500EH）传送到UART0的发送缓冲器SDR00（40041140H）。

图15-12 链传送的使用例子：连续取A/D转换结果进行UART0发送



15.6 使用DMA时的注意事项

15.6.1 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）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）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。

15.6.2 DMA控制数据区和DMA向量表区的分配

- (1) 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。
- (2) 堆栈区、DMA控制数据区和DMA向量表区不能重叠。

15.6.3 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15-8所示。

表15-8 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15-9回写控制数据所需的时钟数”

注2：有关读写数据所需的时钟数，请参照“表15-10读写数据所需的时钟数”。

表15-9 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

注：j=0~7，X：“0”或者“1”

表15-10 读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	外设寄存器
读数据	1	4	4	2
写数据	1	—	—	2

15.6.4 DMA的响应时间

DMA响应时间如表15-11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15-11 DMA的响应时间

	最短时间
响应时间	2个时钟

注：1个时钟：1/f_{CLK}（f_{CLK}：CPU/外围硬件时钟）

15.6.5 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“15.4.3 向量表”。

15.6.6 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：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16章 快速模数转换器（ADC）

16.1 快速模数转换器的功能

快速模数转换器是将模拟输入转换为数字值的转换器，最多能控制67个模拟通道的A/D转换（60个引脚输入通道和7个内部通道）。

快速模数转换器有以下功能。

- 12位分辨率的A/D转换

从ANI0~ANI59，从BG、VDD、GND、VDD2AD、DAC、触摸LDO或模拟CAP输出中选择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。
通道选择模式	硬件触发等待模式	在切断电源的转换待机状态下，通过检测硬件触发来接通电源，在经过A/D电源稳定等待时间后自动开始转换。
	选择模式	选择1个通道的模拟输入，进行A/D转换。
转换模式	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~ANI59中连续的4个通道作为模拟输入。
	单次转换模式	对所选通道进行1次A/D转换。
采样时间	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16-1 A/D转换器的框图

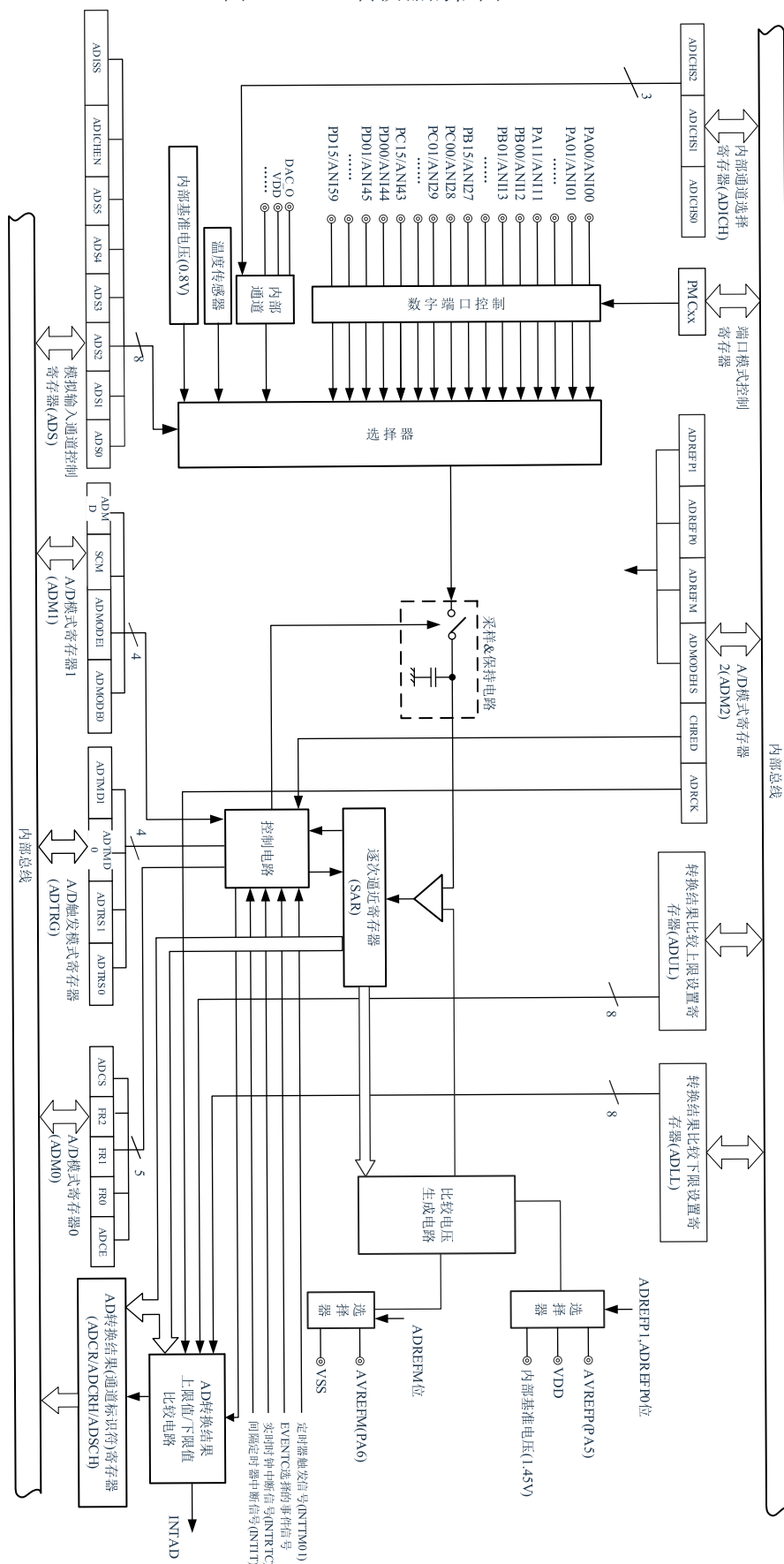
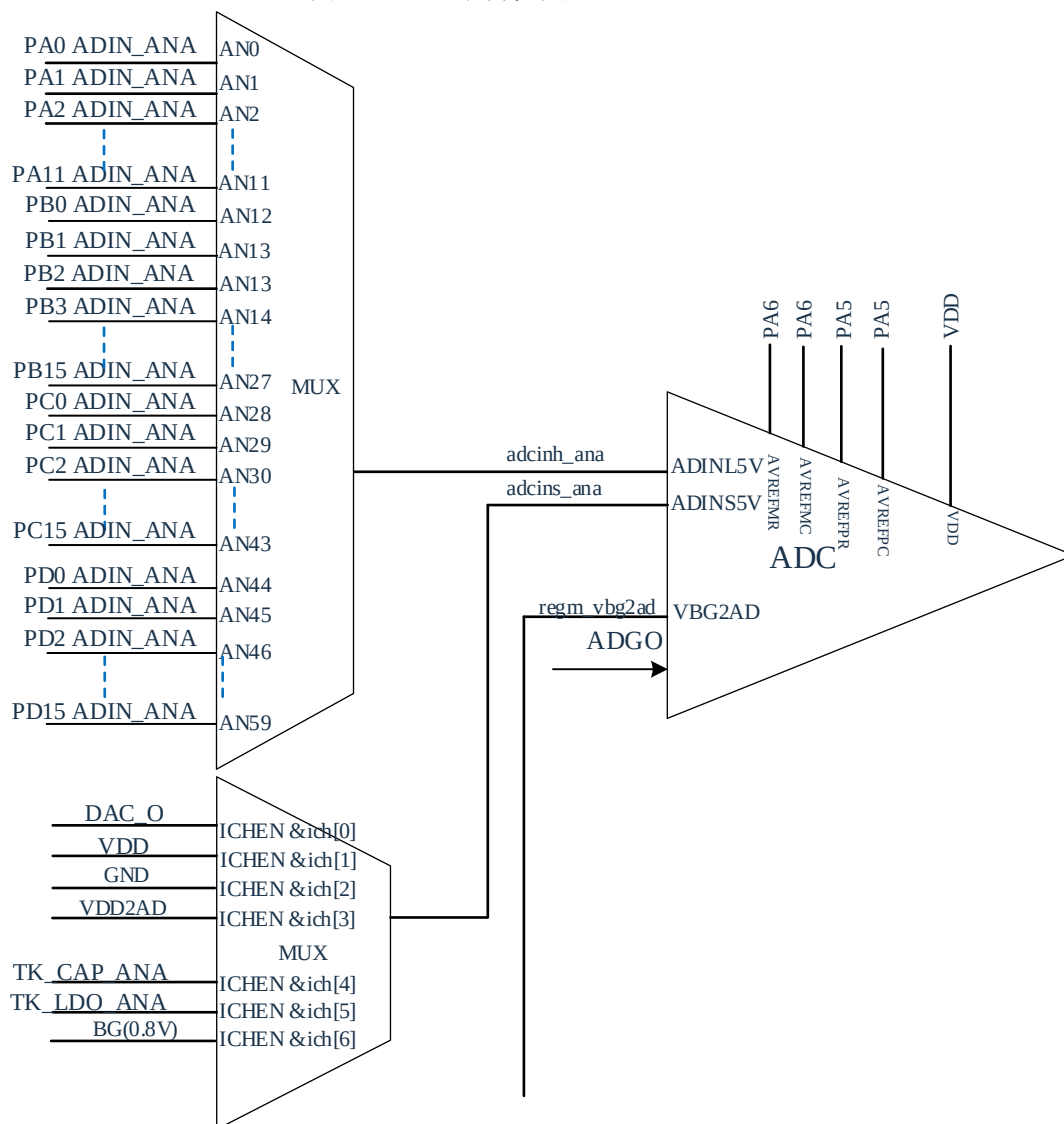


图16-2 ADC结构框图



16.2 寄存器映射

ADC基地址 = 0x40045000 RO: 只读, WO: 只写, R/W: 读写

寄存器名称	偏移量	读写	寄存器描述	复位值
ADM0	0x00	R/W	A/D转换器的模式寄存器0	00H
ADM1	0x02	R/W	A/D转换器的模式寄存器1	00H
ADM2	0x04	R/W	A/D转换器的模式寄存器2	00H
ADTRG	0x06	R/W	A/D转换器的触发模式寄存器	00H
ADS	0x08	R/W	模拟输入通道指定寄存器	00H
ADICHS	0x09	R/W	内部通道选择寄存器	00H
ADLL	0x0A	R/W	转换结果比较下限值设置寄存器	00H
ADUL	0x0B	R/W	转换结果比较上限值设置寄存器	FFH
ADNSMP	0x0C	R/W	A/D转换器的采样时间控制寄存器	0DH
ADSCH	0x0D	RO	扫描模式AD转换通道标识符寄存器	00H
ADCR	0x0E	RO	12位A/D转换结果寄存器	0000H
ADCRH	0x0F	RO	8位A/D转换结果寄存器	00H
ADTES	0x10	R/W	A/D测试寄存器	00H
ADNDIS	0x11	R/W	A/D转换器的充放电控制寄存器	00H
ADSMPWAIT	0x15	R/W	A/D转换器的采样时间延长控制寄存器	00H
ADFLG	0x16	RO	A/D硬模块状态寄存器	00H

16.3 寄存器说明

16.3.1 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”。

位	符号	描述	复位值
7	ADCS	A/D转换运行控制 0: 写：停止转换运行 读：停止转换运行/待机状态 1: 写：允许转换运行 读：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	0
6	-	保留，必须为0	-
5:3	FR<2:0>	ADC时钟分频选择位 000: f _{CLK} /32 001: f _{CLK} /16 010: f _{CLK} /8 011: f _{CLK} /4 100: f _{CLK} /2 101: f _{CLK} /1 其他: 保留，禁止选择	0x0
2:1	-	保留，必须为0	-
0	ADCE	A/D电压比较器使能位 ^{注2} 0: 禁止A/D电压比较器 1: 使能A/D电压比较器	0

注：

1. 有关FR2~FR0位、SHT1~SHT0位和A/D转换的详细内容，请参照“表16-3 A/D转换时间的选择”。
2. A/D转换器开始运行需要1μs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1μs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1μs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1μs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“16.5 A/D转换器的设置流程图”的步骤进行设置。

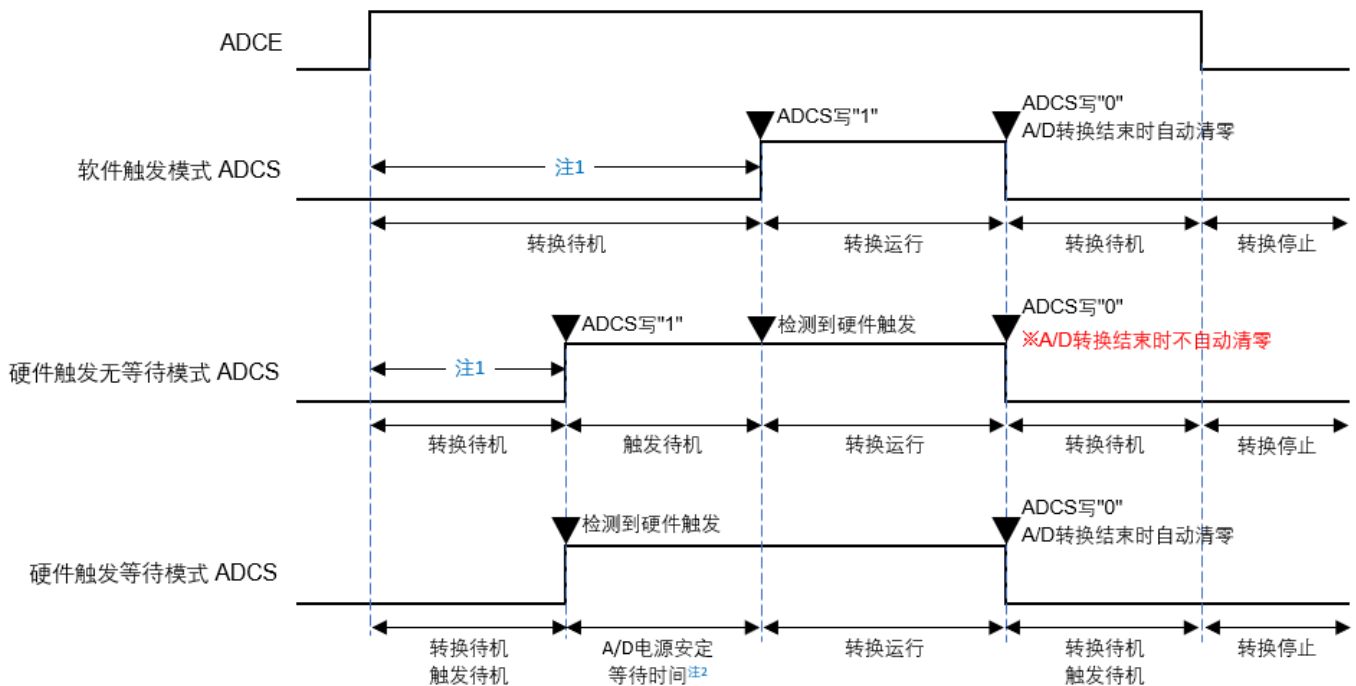
表16-1 ADCS位和ADCE位的设置

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16-2 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16-3 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。
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：
4. 硬件触发无等待模式时：2个 f_{CLK} 时钟+A/D转换时间
5. 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16-3 A/D转换时间的选择(1/2)

(1) 无A/D电源稳定等待时间（软件触发模式/硬件触发无等待模式）

A/D转换器的模式 寄存器0（ADM0）			A/D转换器的模式 寄存器1（ADM1）		模式	转换时钟 ADCLK的频率 (f_{AD})	12位分辨率的转换时间	
FR2	FR1	FR0	ADMODE[1]	ADM0D[0]			转换时钟数	转换时间
0	0	0	0	0	高速变 换模式	$f_{CLK}/32$	45个ADCLK (采样时钟数: 13.5个 ADCLK)	$1440/f_{CLK}$
0	0	1				$f_{CLK}/16$		$720/f_{CLK}$
0	1	0				$f_{CLK}/8$		$360/f_{CLK}$
0	1	1				$f_{CLK}/4$		$180/f_{CLK}$
1	0	0				$f_{CLK}/2$		$90/f_{CLK}$
1	0	1				$f_{CLK}/1$		$45/f_{CLK}$
0	0	0	1	1	低电流 模式	$f_{CLK}/32$	54个ADCLK (采样时钟数: 13.5个 ADCLK)	$1728/f_{CLK}$
0	0	1				$f_{CLK}/16$		$864/f_{CLK}$
0	1	0				$f_{CLK}/8$		$432/f_{CLK}$
0	1	1				$f_{CLK}/4$		$216/f_{CLK}$
1	0	0				$f_{CLK}/2$		$108/f_{CLK}$
1	0	1				$f_{CLK}/1$		$54/f_{CLK}$

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

备注： f_{CLK} ：CPU/外围硬件的时钟频率

表16-3 A/D转换时间的选择(2/2)

有A/D电源稳定等待时间（硬件触发等待模式^{注1}）

A/D转换器的模式 寄存器0（ADM0）			A/D转换器的模式 寄存器1（ADM1）		模式	转换时钟 ADCLK的频 率(f_{AD})	A/D电源 稳定等待 时间	转换时钟数	A/D电源稳定 等待时间 +转换时间
FR2	FR1	FR0	ADMODE[1]	ADM0D[0]					
0	0	0	0	0	高速变 换模式	$f_{CLK}/32$	$1\mu s$	45个ADCLK (采样时钟数 : 13.5个 ADCLK)	$1\mu s + 1440/f_{CLK}$
0	0	1				$f_{CLK}/16$			$1\mu s + 720/f_{CLK}$
0	1	0				$f_{CLK}/8$			$1\mu s + 360/f_{CLK}$
0	1	1				$f_{CLK}/4$			$1\mu s + 180/f_{CLK}$
1	0	0				$f_{CLK}/2$			$1\mu s + 90/f_{CLK}$
1	0	1				$f_{CLK}/1$			$1\mu s + 45/f_{CLK}$
0	0	0	1	1	低电流 模式	$f_{CLK}/32$	$1\mu s$	54个ADCLK (采样时钟数: 13.5个ADCLK)	$1\mu s + 1728/f_{CLK}$
0	0	1				$f_{CLK}/16$			$1\mu s + 864/f_{CLK}$
0	1	0				$f_{CLK}/8$			$1\mu s + 432/f_{CLK}$
0	1	1				$f_{CLK}/4$			$1\mu s + 216/f_{CLK}$
1	0	0				$f_{CLK}/2$			$1\mu s + 108/f_{CLK}$
1	0	1				$f_{CLK}/1$			$1\mu s + 54/f_{CLK}$

注1.: 在连续转换模式时，仅在第1次检测到硬件触发后，发生A/D电源稳定等待时间（参照表16-3）。

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

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。

16.3.2 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”。

位	符号	描述	复位值
7	ADMD	A/D通道选择模式 0: 选择模式 1: 扫描模式	0
6:4	-	保留，必须为0	-
3	ADSCM	A/D转换模式选择位 0: 连续转换模式 1: 单次转换模式	0
2	-	保留	-
1:0	ADMODE<1:0>	A/D转换模式 00: 高速转换模式 11: 低电流模式 其他: 禁止设置	0x0

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

2.为了正常结束A/D转换，必须至少将硬件触发间隔设置为以下时间：

硬件触发无等待模式时：2个 f_{CLK} 时钟+A/D转换时间

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

备注1： f_{CLK} ：CPU/外围硬件的时钟频率

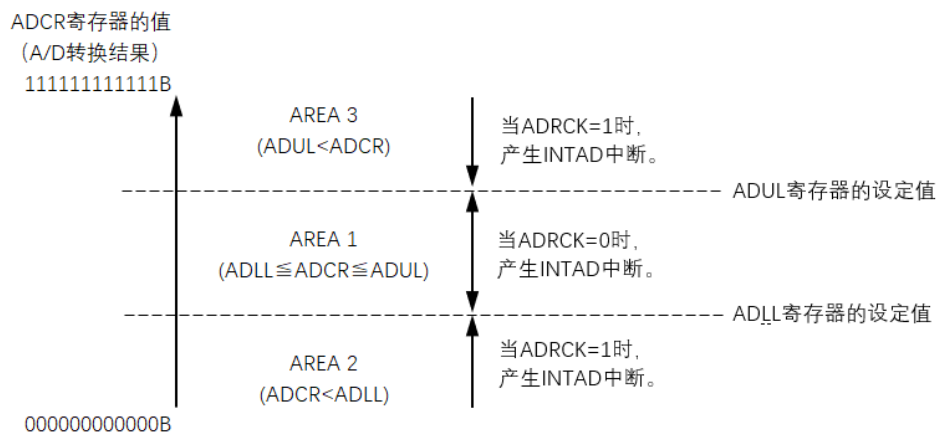
16.3.3 A/D转换器的模式寄存器2（ADM2）

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”。

位	符号	描述	复位值
7:6	ADREFP<1:0>	A/D转换器正端基准电压选择位 00: V_{DD} 01: PA5/ $AV_{REFP}/ANI5$ 10: 内部基准电压 11: 禁止选择	0x0
5	ADREFM	A/D转换器负端基准电压选择位 0: 由 V_{SS} 提供 1: PA6/ $AV_{REFM}/ANI6$	0
4	-	保留，必须为0	-
3	ADRCK	转换结果上限值和下限值的检查 AREA1~AREA3的中断信号（INTAD）的产生范围如图16-4所示 0: 当ADLL寄存器 $\leq$ ADCR寄存器 $\leq$ 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）。	0
2	-	保留，必须为0	-
1	CHRED	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ADSCH寄存器中标识通道号 1: 扫描模式时，ADSCH寄存器的低五位(ADSCH[4:0])位此结果的通道号	0
0	ADMODEHS	更高速模式选择 0: 正常高速模式 1: 更高速模式	0

图16-4 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ANI0和ANI1设置为模拟输入，并且通过端口模式寄存器设置为输入模式。

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

16.3.4 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”。

位	符号	描述	复位值
7:6	ADTMD<1:0>	A/D转换触发模式选择 00: 软件触发 01: 软件触发 10: 硬件触发无等待模式 11: 硬件触发等待模式	0x0
5:2	-	保留，必须为0	-
1:0	ADTRS<1:0>	硬件触发信号选择位 00: 定时器通道1计数结束或捕获结束中断信号 (INTTM01) 01: ELC选择的事件信号 10: 实时时钟中断信号(INTRTC) 11: 间隔定时器中断信号(INTIT)	0x0

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

16.3.5 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”。

位	符号	描述	复位值
7	ADISS	详见下表	0
6	ADICHEN	详见下表	0
5:0	ADS<5:0>	详见下表	0x0

选择模式（ADM1.ADMD=0）

ADS 寄存器设置值								CH 选择
ADISS	ADICHEN	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0	
0	0	0	0	0	0	0	0	ANI0
0	0	0	0	0	0	0	1	ANI1
0	0	0	0	0	0	1	0	ANI2
0	0	0	0	0	0	1	1	ANI3
0	0	0	0	0	1	0	0	ANI4
0	0	0	0	0	1	0	1	ANI5
0	0	0	0	0	1	1	0	ANI6
0	0	0	0	0	1	1	1	ANI7
0	0							
0	0	1	1	0	1	0	1	ANI53
0	0	1	1	0	1	1	0	ANI54
0	0	1	1	0	1	1	1	ANI55
0	0	1	1	1	0	0	0	ANI56
0	0	1	1	1	0	0	1	ANI57
0	0	1	1	1	0	1	0	ANI58
0	0	1	1	1	0	1	1	ANI59
0	0	1	1	1	1	1	1	ANI0~ANI59 通道全关
1	0	0	0	0	0	0	1	内部基准电压（1.45V）
0	1	0	0	0	0	0	0	内部通道使能
其他								禁止设置

扫描模式（ADM1.ADMD=1）

ADISS	ADS5	ADS4	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	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	0	0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	0	0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	0	0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	0	0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
0	0	0	1	1	0	0	ANI12	ANI13	ANI14	ANI15
0	0	0	1	1	0	1	ANI13	ANI14	ANI15	ANI16
0	0	0	1	1	1	0	ANI14	ANI15	ANI16	ANI17
0	0	0	1	1	1	1	ANI15	ANI16	ANI17	ANI18
0	0	1	0	0	0	0	ANI16	ANI17	ANI18	ANI19
0	0	1	0	0	0	1	ANI17	ANI18	ANI19	ANI20
0	0	1	0	0	1	0	ANI18	ANI19	ANI20	ANI21
0	0	1	0	0	1	1	ANI19	ANI20	ANI21	ANI22
0	0	1	0	1	0	0	ANI20	ANI21	ANI22	ANI23
0	0	1	0	1	0	1	ANI21	ANI22	ANI23	ANI24
0	0	1	0	1	1	0	ANI22	ANI23	ANI24	ANI25
0	0	1	0	1	1	1	ANI23	ANI24	ANI25	ANI26
0	0	1	1	0	0	0	ANI24	ANI25	ANI26	ANI27
0	0	1	1	0	0	1	ANI25	ANI26	ANI27	ANI28
0	0	1	1	0	1	0	ANI26	ANI27	ANI28	ANI29
0	0	1	1	0	1	1	ANI27	ANI28	ANI29	ANI30
0	0	1	1	1	0	0	ANI28	ANI29	ANI30	ANI31
0	0	1	1	1	0	1	ANI29	ANI30	ANI31	ANI32
0	0	1	1	1	1	0	ANI30	ANI31	ANI32	ANI33
0	0	1	1	1	1	1	ANI31	ANI32	ANI33	ANI34
0	1	0	0	0	0	0	ANI32	ANI33	ANI34	ANI35
0	1	0	0	0	0	1	ANI33	ANI34	ANI35	ANI36
0	1	0	0	0	1	0	ANI34	ANI35	ANI36	ANI37
0	1	0	0	0	1	1	ANI35	ANI36	ANI37	ANI38
0	1	0	0	1	0	0	ANI36	ANI37	ANI38	ANI39
0	1	0	0	1	0	1	ANI37	ANI38	ANI39	ANI40
0	1	0	0	1	1	0	ANI38	ANI39	ANI40	ANI41
0	1	0	0	1	1	1	ANI39	ANI40	ANI41	ANI42
0	1	0	1	0	0	0	ANI40	ANI41	ANI42	ANI43

0	1	0	1	0	0	1	ANI41	ANI42	ANI43	ANI44
0	1	0	1	0	1	0	ANI42	ANI43	ANI44	ANI45
0	1	0	1	0	1	1	ANI43	ANI44	ANI45	ANI46
0	1	0	1	1	0	0	ANI44	ANI45	ANI46	ANI47
0	1	0	1	1	0	1	ANI45	ANI46	ANI47	ANI48
0	1	0	1	1	1	0	ANI46	ANI47	ANI48	ANI49
0	1	0	1	1	1	1	ANI47	ANI48	ANI49	ANI50
0	1	1	0	0	0	0	ANI48	ANI49	ANI50	ANI51
0	1	1	0	0	0	1	ANI49	ANI50	ANI51	ANI52
0	1	1	0	0	1	0	ANI50	ANI51	ANI52	ANI53
0	1	1	0	0	1	1	ANI51	ANI52	ANI53	ANI54
0	1	1	0	1	0	0	ANI52	ANI53	ANI54	ANI55
0	1	1	0	1	0	1	ANI53	ANI54	ANI55	ANI56
0	1	1	0	1	1	0	ANI54	ANI55	ANI56	ANI57
0	1	1	0	1	1	1	ANI55	ANI56	ANI57	ANI58
0	1	1	1	0	0	0	ANI56	ANI57	ANI58	ANI59
其他							禁止设置			

注意：

1. 必须将bit4,bit5和bit6置“0”。
2. 对于由PMCx寄存器设置为模拟输入的端口，才可以通过ADS指定为模拟输入进行A/D转换。
2. 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。
3. 要改写ADISS位时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。
4. 当将 AV_{REFP} 用作A/D转换器的正（+）基准电压时，不能选择ANI0作为A/D转换通道。
5. 当将 AV_{REFM} 用作A/D转换器的负（-）基准电压时，不能选择ANI1作为A/D转换通道。
6. 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“16.6.4内部基准电压时的设置”。
7. 要转移到深度睡眠模式时或者要在CPU以副系统时钟运行中转移到睡眠模式时，不能将ADISS位置“1”。

16.3.6 A/D内部通道选择寄存器（ADICHS）

这是设置A/D内部通道的寄存器。

通过8位存储器操作指令设置ADICHS寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	-	保留，必须为0	-
2:0	ADICHS<2:0>	内部通道选择位（仅在ADS寄存器使能内部通道时 (ADS.ADICHEN=1)才有效) 000: DAC_O 001: VDD 010: GND 011: VDD2AD 100: TK_CAP_ANA 101: TK_LDO_ANA 110: BG(0.8V) 其他: 保留，禁止选择	0x0

16.3.7 12位A/D转换通道标识符寄存器（ADSCH）

这是保存A/D转换通道标识符的8位寄存器，此寄存器只可读。每当A/D转换结束时，就输出当前AD转换结果对应通道。

此寄存器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8位存储器操作指令读ADSCH寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:6	-	保留	
5:0	ADSCH<5:0>	转换通道标识符 选择模式(ADM1.ADMD=0)时，ADSCH[5:0]读出值始终为6'b0； 扫描模式(ADM1.ADMD=1)时，且ADM2.CHEDE=1，ADSCH[5:0]结果对应以通道。 000000: ANI0 000001: ANI1 000010: ANI2 111000: ANI56 111001: ANI57 111010: ANI58 111101: ANI59	0x0

16.3.8 12位A/D转换结果寄存器（ADCR）

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

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16-4））的设置值范围内，就不保存A/D转换结果。

位	符号	描述	复位值
15:12	-	保留	-
11:0	ADCR<11:0>	A/D转换结果	0x0

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。

16.3.9 8位A/D转换结果寄存器（ADCRH）

这是保存A/D转换结果的8位寄存器，保存12位分辨率的高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”。

位	符号	描述	复位值
7:0	ADCRH<7:0>	A/D转换结果	0x0

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ADRCCK位和ADUL/ADLL寄存器进行设置（参照图16-4））的设置值范围内，就不保存A/D转换结果。

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。

16.3.10 转换结果比较上限值设置寄存器（ADUL）

这是用于检查A/D转换结果上限值的设置寄存器。

将A/D转换结果和ADUL寄存器的值进行比较，并且在A/D转换器的模式寄存器2（ADM2）的ADRCCK位的设置范围内（参照图16-4）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。

位	符号	描述	复位值
7:0	ADUL<7:0>	转换结果比较上限值	0xff

16.3.11 转换结果比较下限值设置寄存器（ADLL）

这是用于检查A/D转换结果下限值的设置寄存器。

将A/D转换结果和ADLL寄存器的值进行比较，并且在A/D转换器的模式寄存器2（ADM2）的ADRCCK位的设置范围内（参照图16-4）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”。

位	符号	描述	复位值
7:0	ADLL<7:0>	转换结果比较下限值	0x0

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。

16.3.12 A/D采样时间控制寄存器（ADNSMP）

此寄存器可控制A/D采样时间。
通过8位存储器操作指令设置ADNSMP寄存器。
在产生复位信号后，此寄存器的值变为“0DH”。

位	符号	描述	复位值
7:0	ADNSMP<7:0>	A/D采样时间 00000000: 禁止选择 00000100: 5.5 个 ADCLK 00000101: 6.5 个 ADCLK 00000110: 00001101: 13.5 个 ADCLK 00001110: 11111110: 254.5 个 ADCLK 11111111: 255.5 个 ADCLK	0x0D

注意：要改写ADNSMP寄存器时，必须在转换停止状态（ADCS=0）下进行。
不同条件下，各转换通道需要保证的采样时间：

A/D 转换模式	AVDD[V]	ANI0~ANI59[ns]
高速变换	4.5~5.5	211
	2.7~5.5	250
	2.4~5.5	422
低电流变换	2.7~5.5	500
	2.4~5.5	844
	1.8~5.5	1688

16.3.13 A/D采样时间延长寄存器（ADSMPWAIT）

此寄存器用于延长A/D采样时间。

通过8位存储器操作指令设置ADSMPWAIT寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留，必须为0	-
0	ADSMPWAIT	采样时间延长控制位 0: A/D采样时间直接由ADNSMP寄存器进行设置 1: 保持采样延迟状态	0

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”。

16.3.14 A/D测试寄存器（ADTES）

此寄存器用于设置A/D转换器的测试模式。

通过8位存储器操作指令设置ADTES寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	-	保留，必须为0	-
2:0	ADTEST<2:0>	A/D运行模式 000: 正常转换模式 001: 0码的自诊断测试 010: 半码的自诊断测试 011: 全码的自诊断测试 其他: 禁止设置	0x0

16.3.15 A/D状态寄存器（ADFLAG）

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ADFLAG寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:5	-	保留，必须为0	-
4	ADFLAG4	A/D转换状态位4 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ADFLAG4保持1'b0	0
3	ADFLAG3	A/D转换状态位3 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)	0
2	ADFLAG2	A/D转换状态位2 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)	0
1	ADFLAG1	A/D转换状态位1 0: 非逐次比较期间 1: 逐次比较期间	0
0	ADFLAG0	A/D转换状态位0 0: 非A/D采样期间 1: A/D采样期间	0

16.3.16 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”。

位	符号	描述	复位值
7:5	-	保留，必须为0	-
4	ADNDIS4	充放电选择 0: 放电 1: 充电	0
3:0	ADNDIS<3:0>	充放电时间选择位 0000: 不进行充、放电 0001: 禁止选择 0010: 2个ADCLK 0011: 3个ADCLK 1111: 15个ADCLK	0x0

16.4 输入电压和转换结果

模拟输入引脚（ANI0～ANI59）的模拟输入电压和理论上的A/D转换结果（12位A/D转换结果寄存器（ADCR））有以下表达式的关系。

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

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

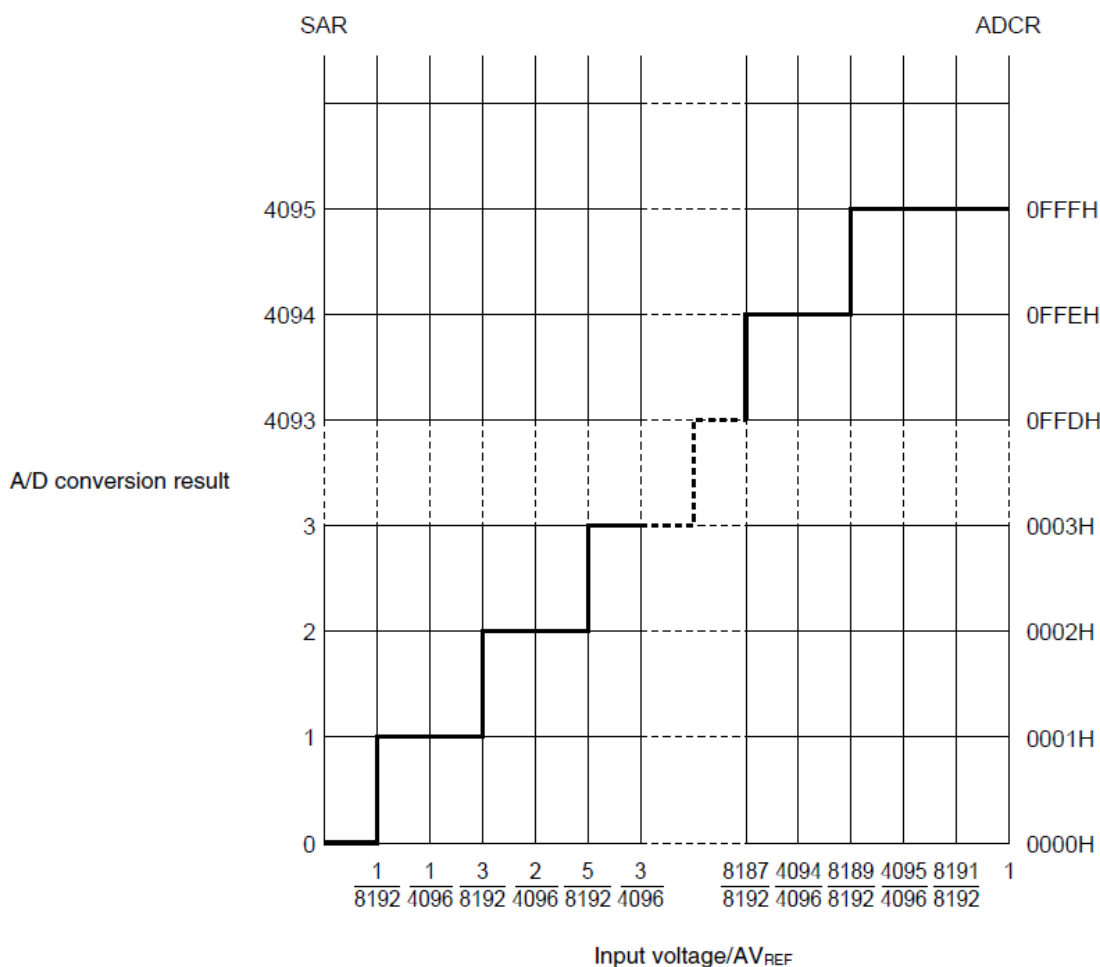
AV_{REF} : AV_{REF} 引脚电压

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转换结果的关系如图16-5所示。

图16-5 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} 。

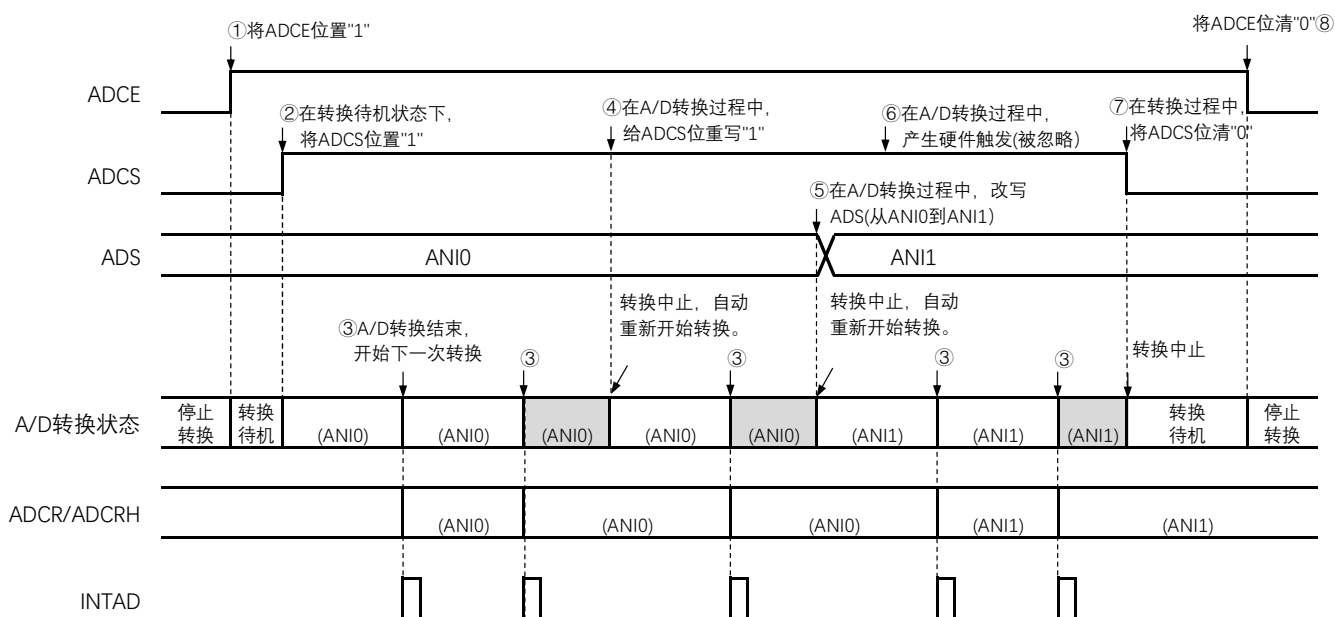
16.5 A/D转换器的运行模式

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“16.5A/D转换器的设置流程图”。

16.5.1 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。

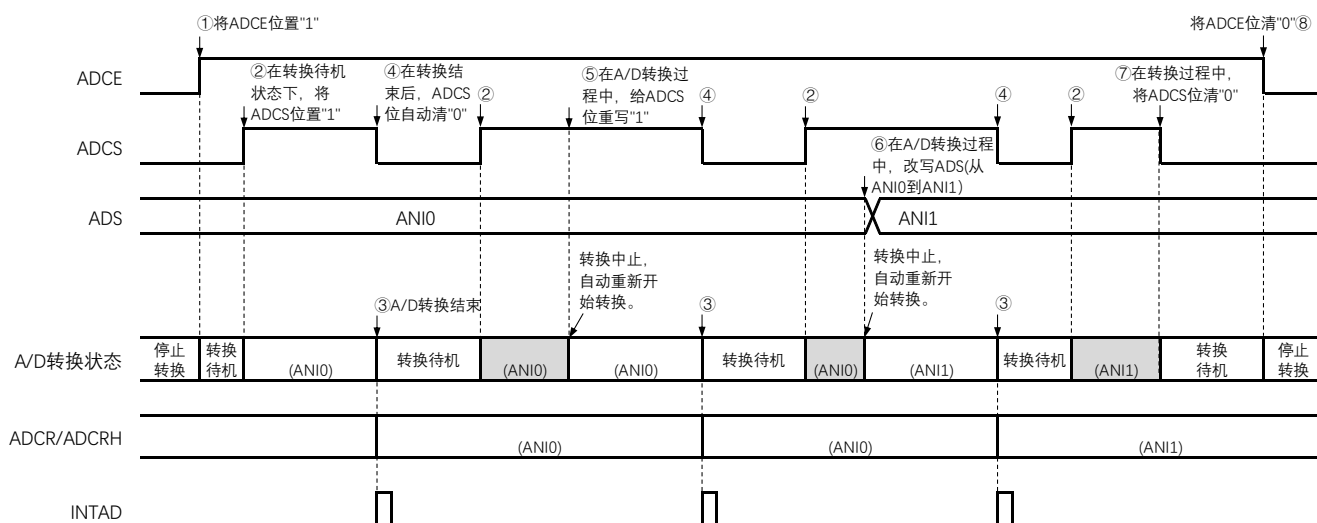
图16-6 软件触发模式（选择模式、连续转换模式）的运行时序例子



16.5.2 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。

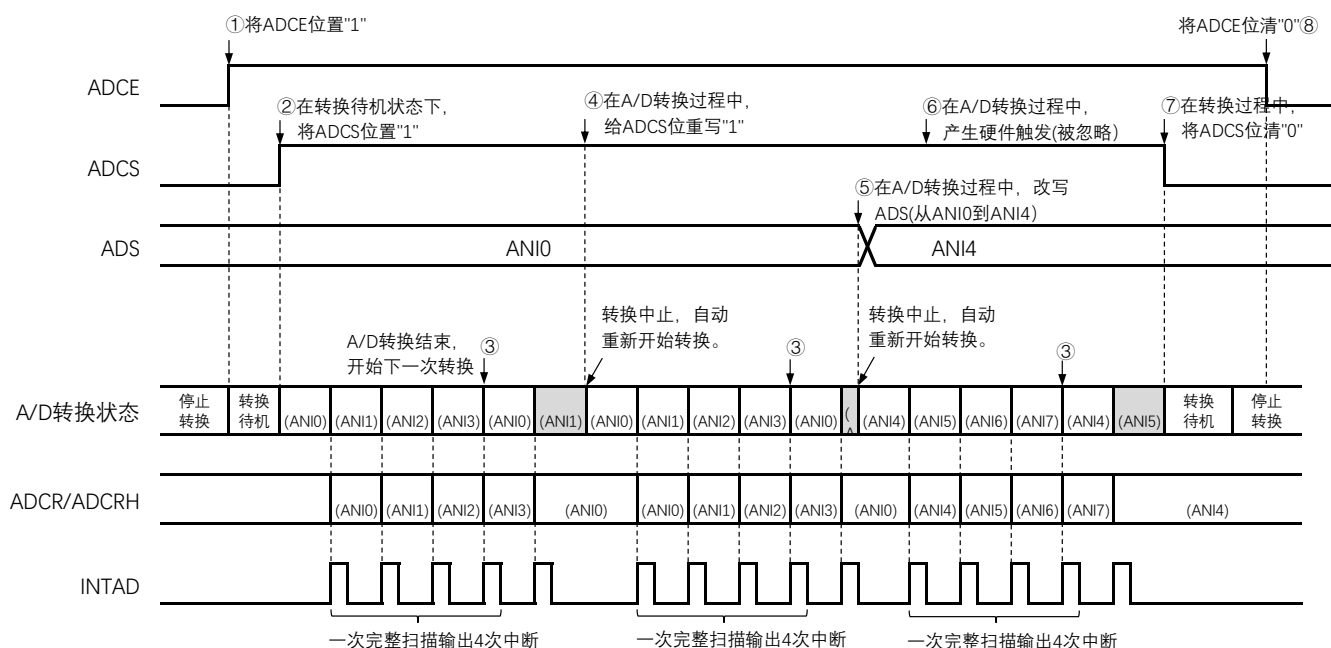
图16-7 软件触发模式（选择模式、单次转换模式）的运行时序例子



16.5.3 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（1 μ s）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。

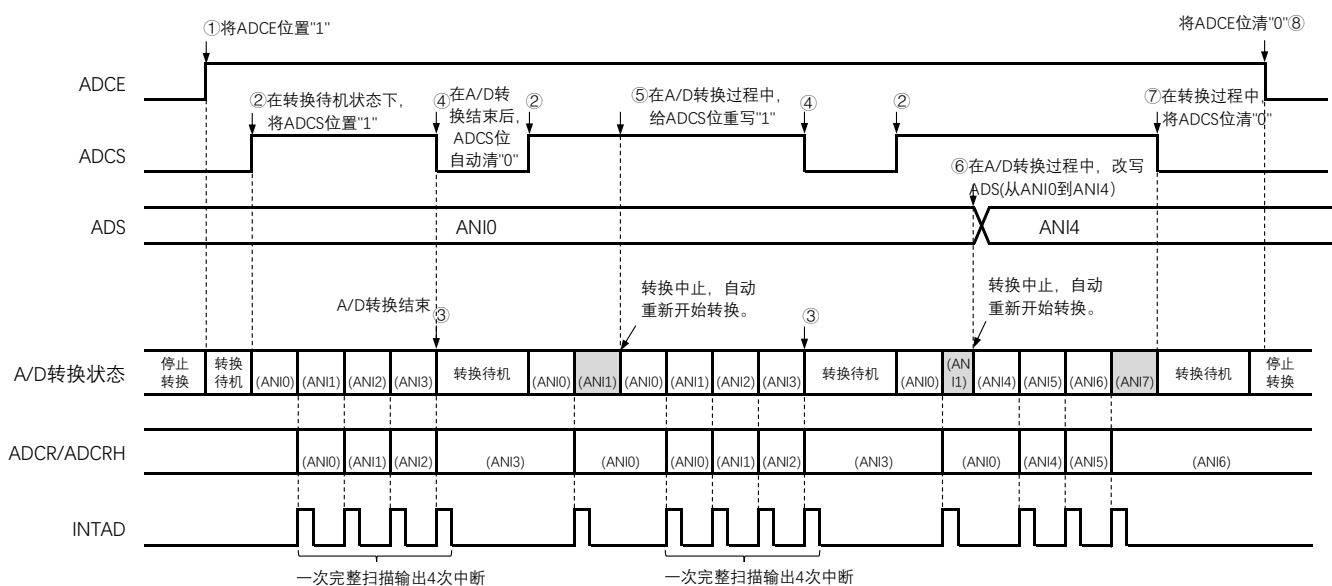
图16-8 软件触发模式（扫描模式、连续转换模式）的运行时序例子



16.5.4 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（1 μ s）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。

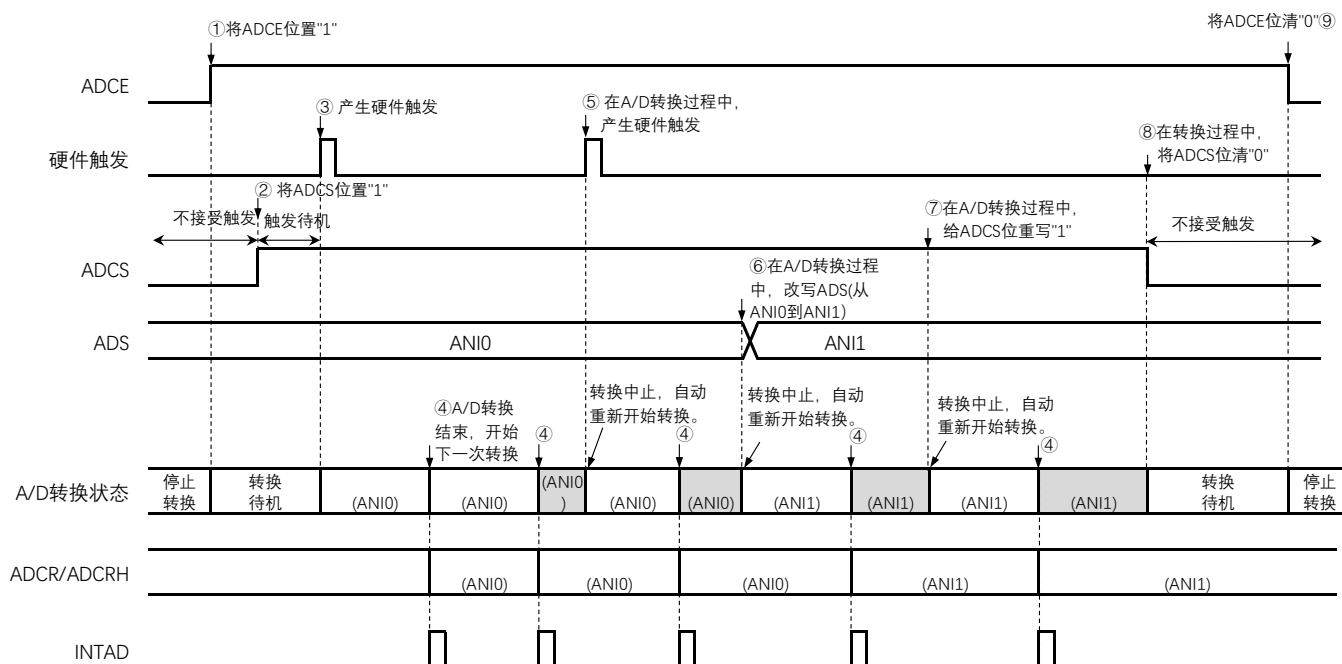
图16-9 软件触发模式（扫描模式、单次转换模式）的运行时序例子



16.5.5 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（1 μ s）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。

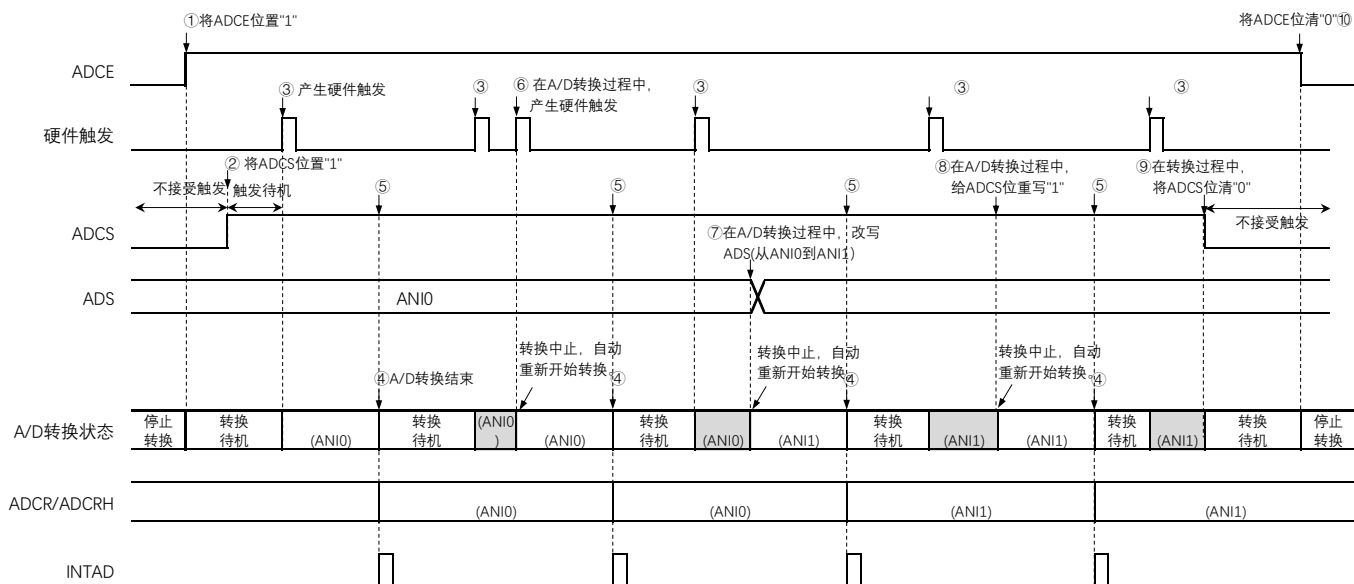
图16-10 硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



16.5.6 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（1μs）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。

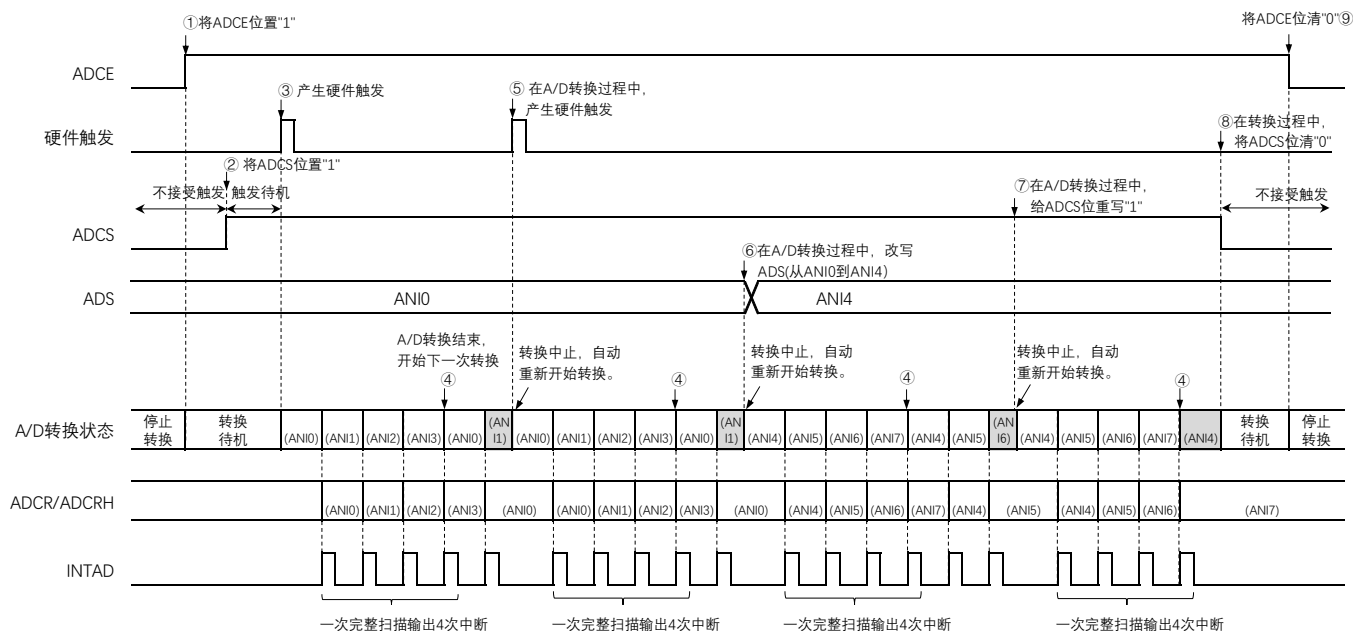
图16-11 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



16.5.7 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（1μs）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。

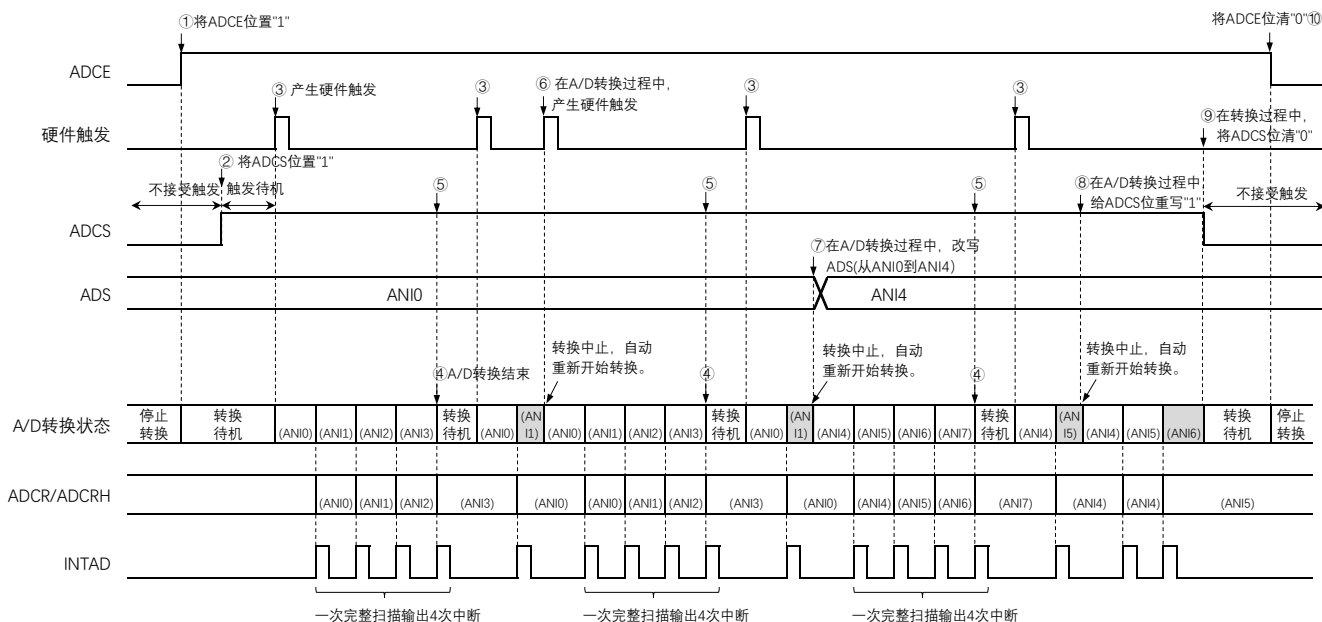
图16-12 硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



16.5.8 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（1μs）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。

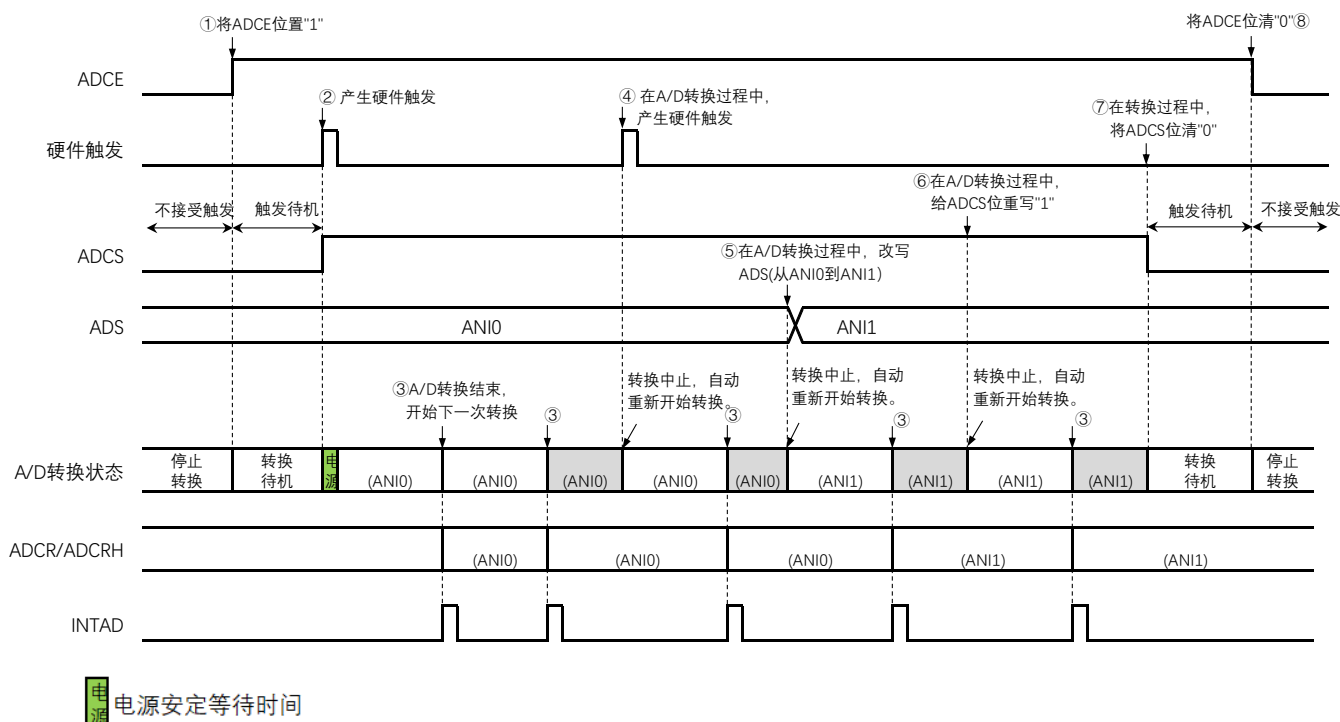
图16-13 硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



16.5.9 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。
- 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。

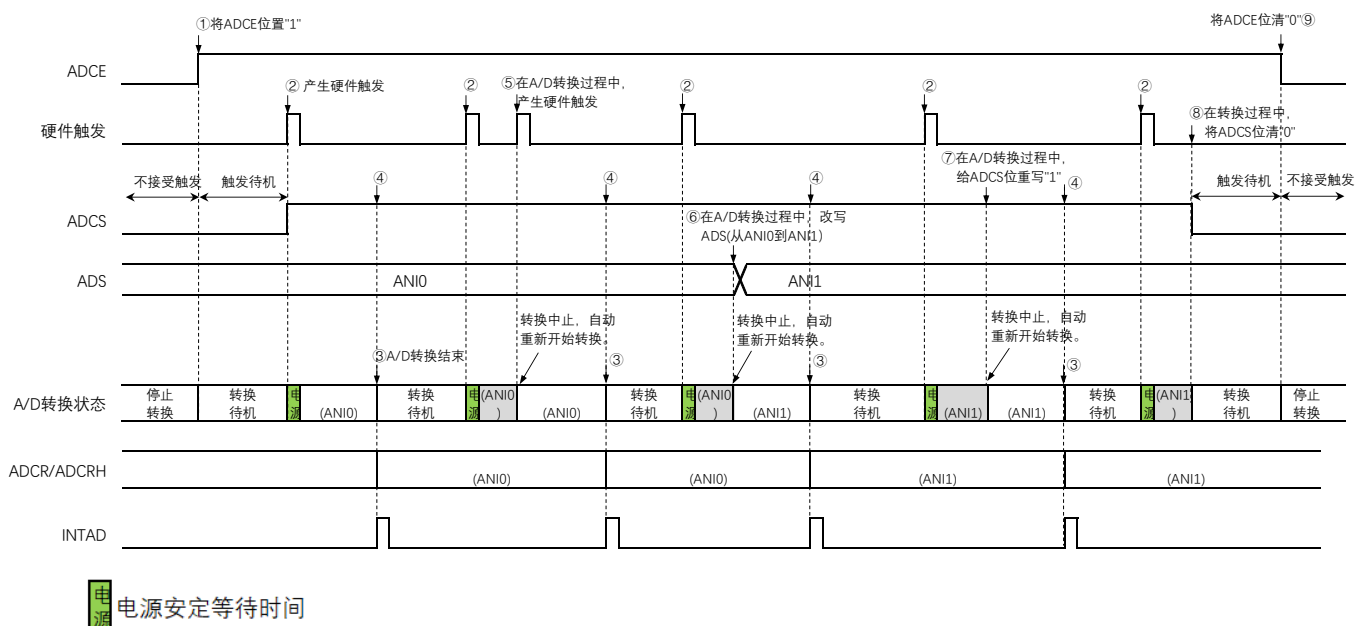
图16-14 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



16.5.10 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。

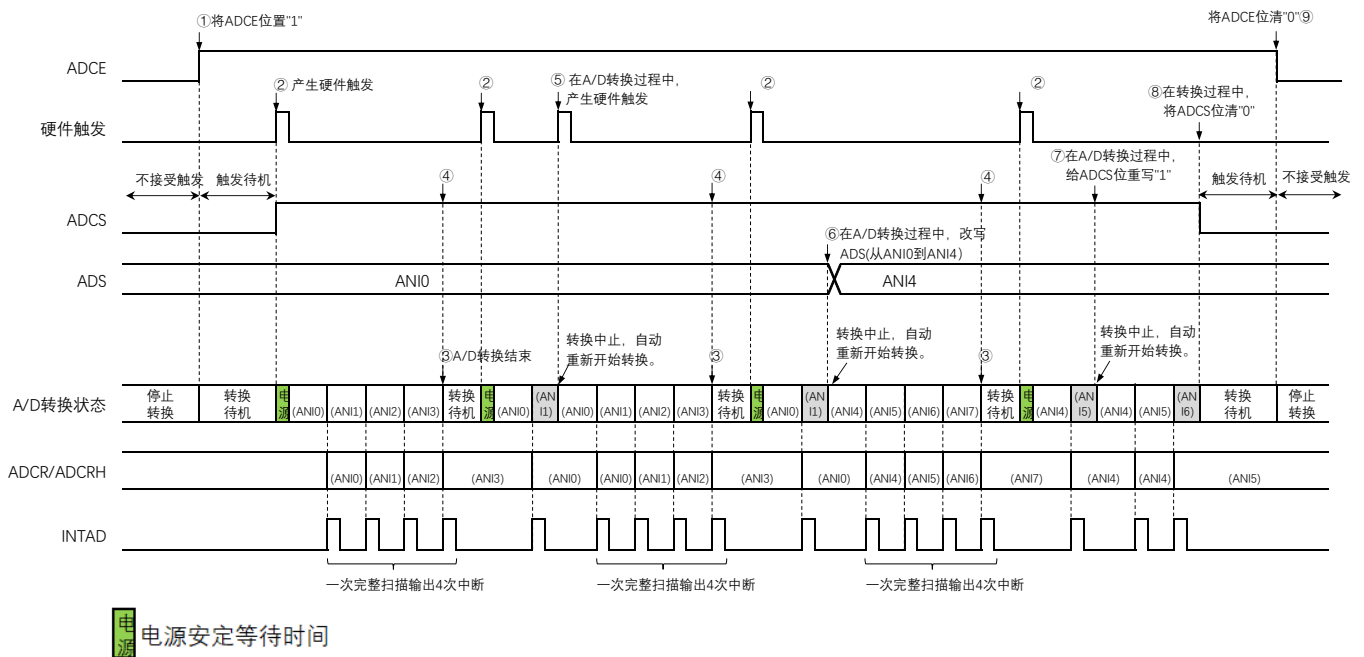
图16-15 硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



16.5.12 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。
- ⑨ 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16-17 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

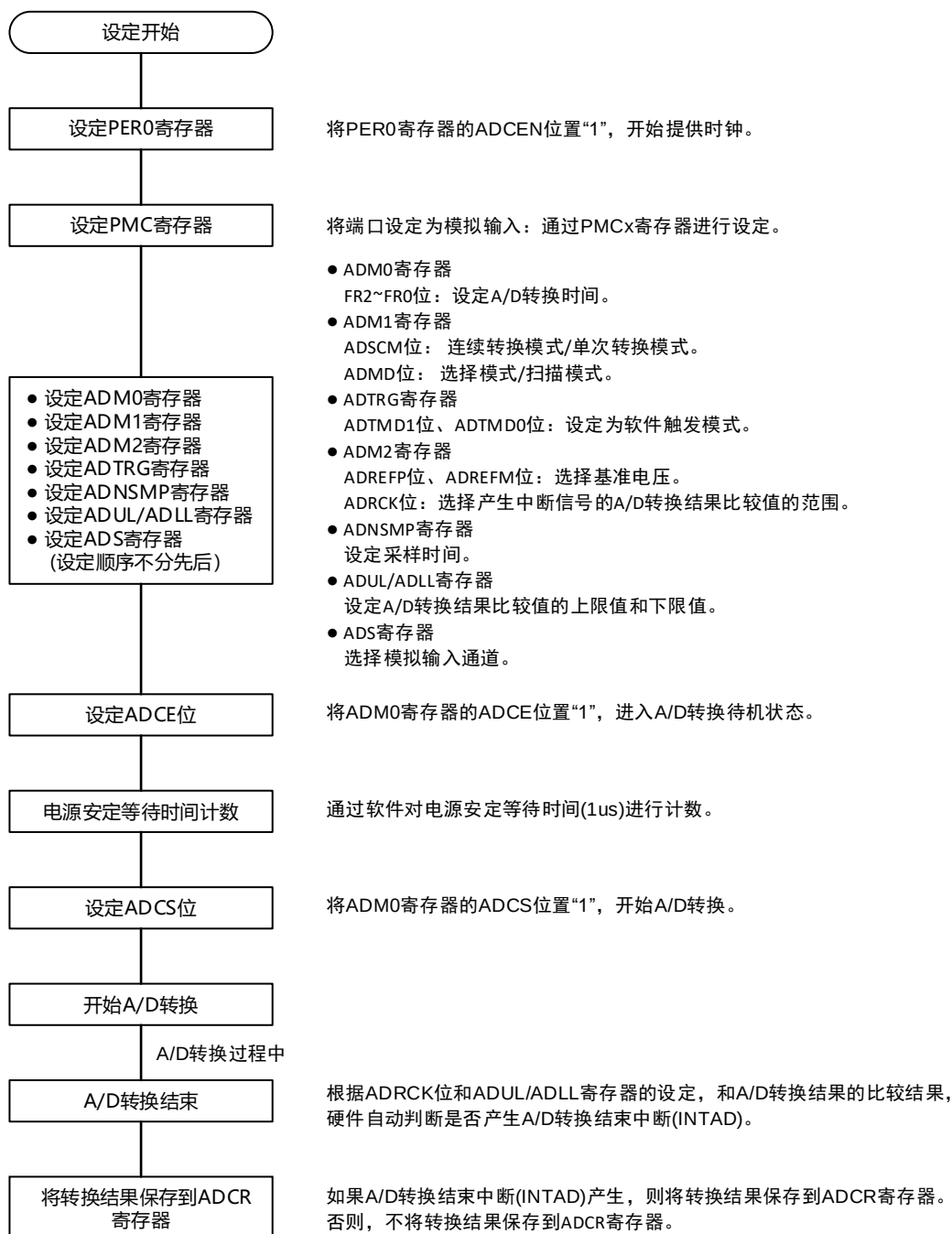


16.6 A/D转换器的设置流程图

各运行模式的A/D转换器的设置流程图如下所示。

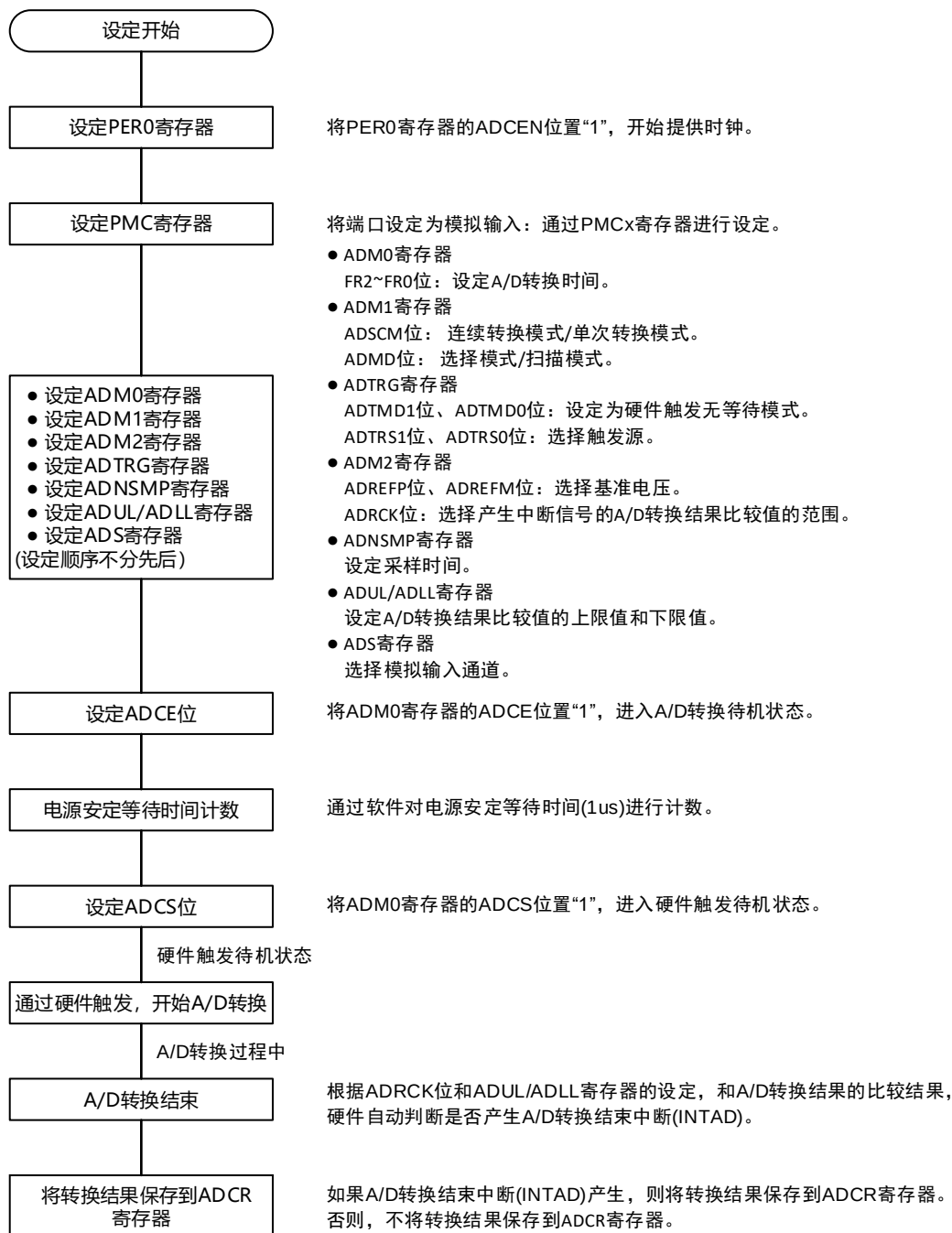
16.6.1 软件触发模式的设置

图16-18 软件触发模式的设置



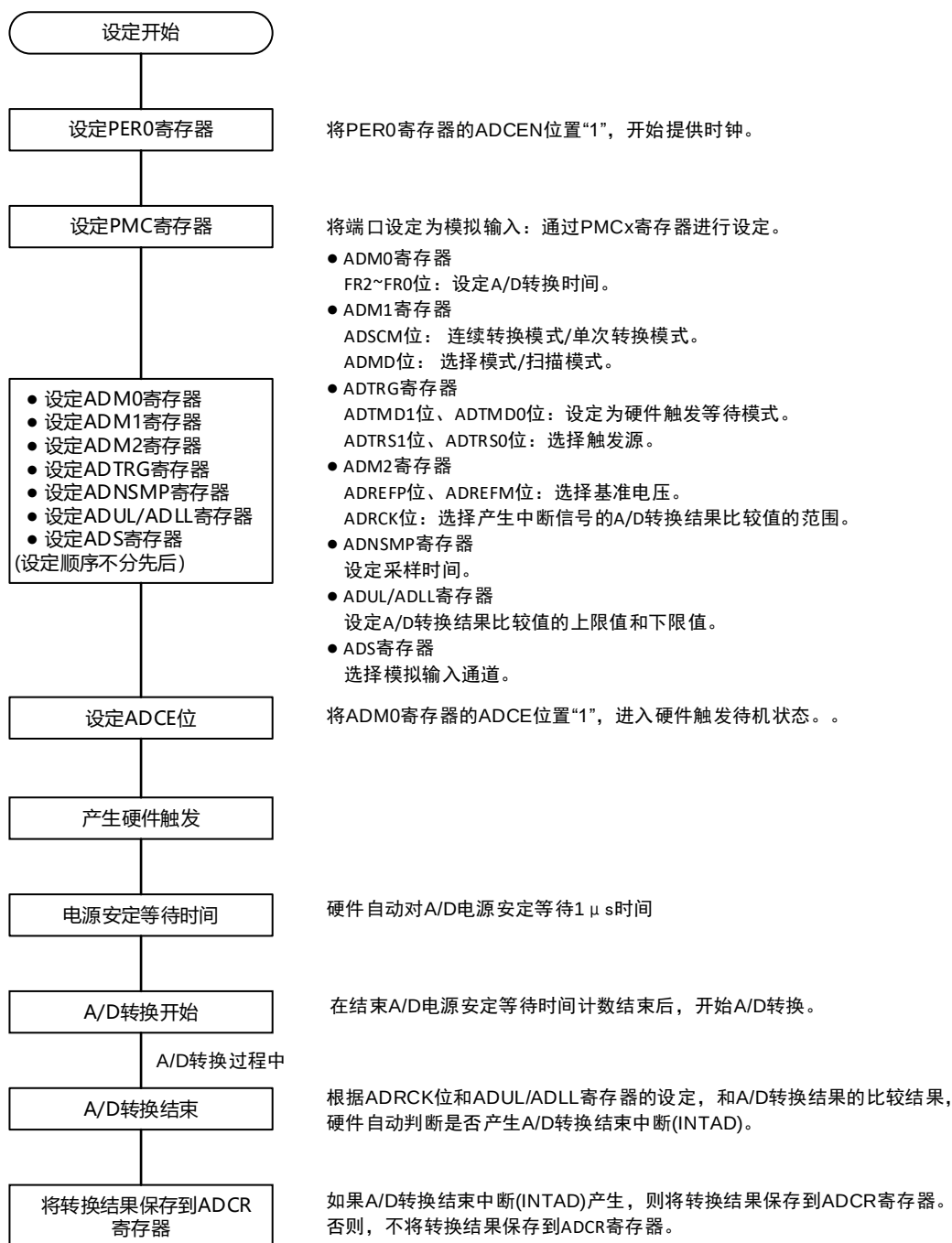
16.6.2 硬件触发无等待模式的设置

图16-19 硬件触发无等待模式的设置



16.6.3 硬件触发等待模式的设置

图16-20 硬件触发等待模式的设置



16.6.4 选择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16-21 选择内部基准电压时的设置



16.6.5 测试模式的设置

图16-22 测试模式的设置(VSS/half_VDD/VDD作为转换对象)



第17章 模拟比较器 (ACMP0/1)

17.1 概述

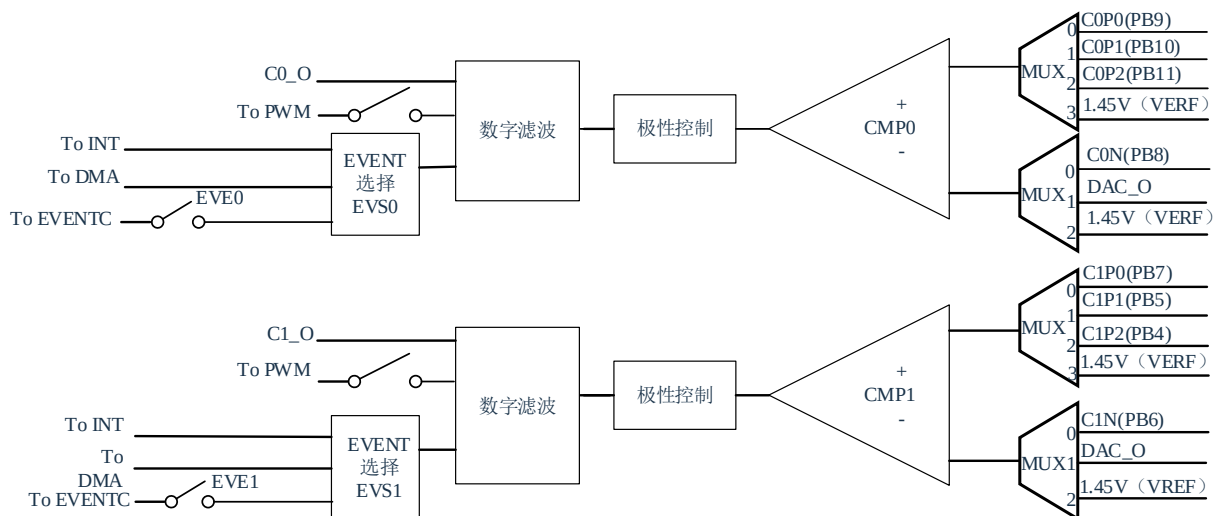
芯片内部包含两个模拟比较器。可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时，比较器输出逻辑1，反之输出0，也可以通过输出极性选择位进行改变。当比较器输出值发生改变时，每路比较器都可通过配置产生中断。

17.2 特性

- 模拟输入电压范围：(0~VDD)V。
- 支持单边/双边迟滞功能。
- 支持迟滞电压选择(10mV/20mV/60mV-典型值)。
- 每个比较器正、负端可多路选择
- 输出可滤波时间可选择：0~512* T_{sys} 。
- 比较器事件输出可作为增强型PWM的截断触发信号。
- 输出改变可产生中断。

17.3 结构框图

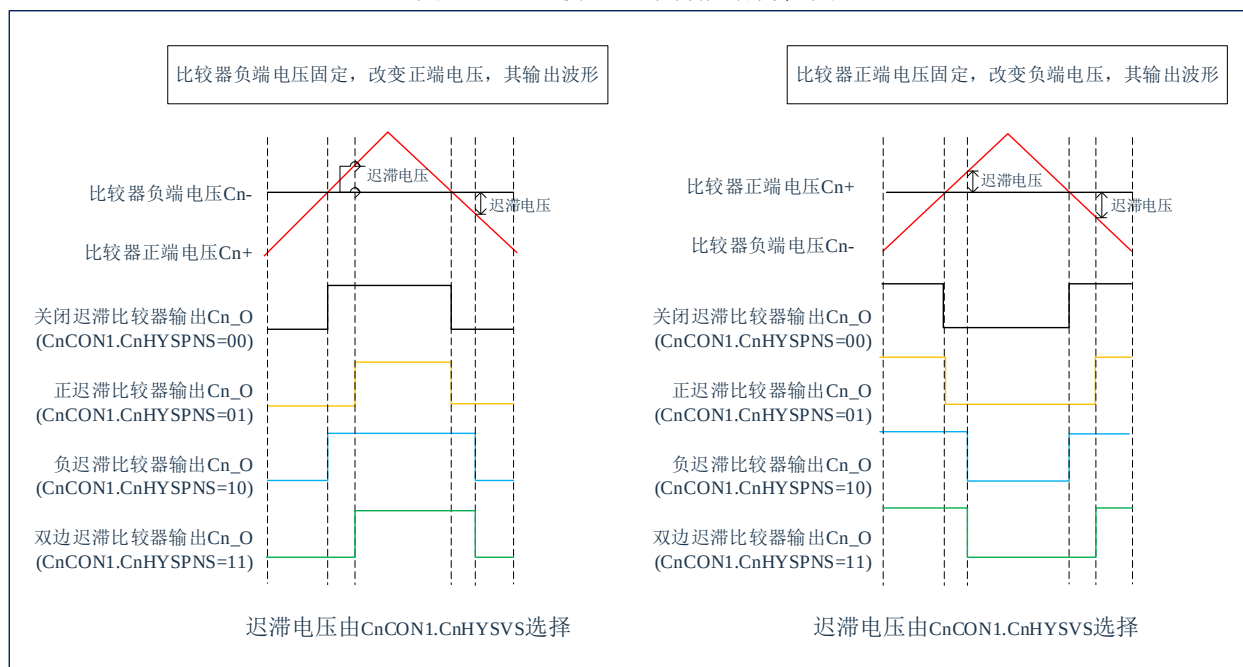
图 17-1: 比较器结构框图



注：若设置滤波使能，会产生以下几种情况：

- 滤波输入信号宽度<滤波设置时间：无信号输出。
- 滤波输入信号宽度=滤波设置时间：输出一个时钟周期的脉冲信号（无法使EPWM截断）
- 滤波输入信号宽度>滤波设置时间：正常滤波输出。

图 17-2: 比较器迟滞功能结构框图



17.4 寄存器映射

(ACMP基地址 = 0x40068200) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
C0CON0(P1B)	0x000	R/W	模拟比较器0控制寄存器0	0x00
C0CON1(P1B)	0x004	R/W	模拟比较器0控制寄存器1	0x00
C1CON0(P1B)	0x008	R/W	模拟比较器1控制寄存器0	0x00
C1CON1(P1B)	0x00C	R/W	模拟比较器1控制寄存器1	0x00
CEVCON(P1B)	0x010	R/W	模拟比较器事件控制寄存器	0x00
ACMPLOCK	0x014	R/W	模拟比较器写使能寄存器	0x00

注:

(P1B)标注的寄存器为被保护的寄存器。

(P1B): ACMPLOCK==55H时, 标注的寄存器允许写入; =其他值, 禁止写入。

17.5 寄存器说明

17.5.1 模拟比较器0控制寄存器0(C0CON0)

位	符号	描述	复位值
31:16	-	保留	0x0
15	C0_EN	模拟比较器0使能位 0: 禁止 1: 使能	0
14	C0_OEN	模拟比较器0输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	0x0
8	C0_OUT	模拟比较器0结果位(只读)	0
7:6	-	保留 -	0
5:4	C0_P_S	模拟比较器0正端通道选择 00: C0P0(PB9) 01: C0P1(PB10) 10: C0P2(PB11) 11: VREF(1.45V)	0x0
3:2	-	保留	0x0
1:0	C0_N_S	模拟比较器0负端通道选择 00: C0N(PB8) 01: DAC_O 10: VREF(1.45V) 11: 禁止	0x0

17.5.2 模拟比较器0控制寄存器1(C0CON1)

位	符号	描述	复位值
31:14	-	保留	0x0
13:12	C0_HYSPN_S	模拟比较器0迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11:10	C0_HYSV_S	模拟比较器0迟滞电压选择 00: 无迟滞 01: 10mV 10: 20mV 11: 60mV	0x0
9	C0_POS	模拟比较器0输出极性选择位 0: 正常输出 1: 反相输出	0
8	C0_FE	模拟比较器0输出滤波使能位 0: 禁止 1: 使能	0
7:4	-	保留	0x0
3:0	C0_FS	模拟比较器0输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0

17.5.3 模拟比较器1控制寄存器0(C1CON0)

位	符号	描述	复位值
31:16	-	保留	-
15	C1_EN	模拟比较器1使能位 0: 禁止 1: 使能	0
14	C1_OEN	模拟比较器1输出使能位 0: 禁止 1: 使能	0
13:9	-	保留	-
8	C1_OUT	模拟比较器1结果位(只读)	0
7:6	-	保留	0x0
5:4	C1_P_S	模拟比较器1正端通道选择 00: C1P0(PB7) 01: C1P1(PB5) 10: C1P2(PB4) 11: VREF(1.45V)	0x0
3:2	-	保留	0
1:0	C1_N_S	模拟比较器1负端通道选择 00: C1N(PB6) 01: DAC_O 10: VREF(1.45V) 其他: 禁止	0x0

17.5.4 模拟比较器1控制寄存器1(C1CON1)

位	符号	描述	复位值
31:14	-	保留	-
13:12	C1HYSPN_S	模拟比较器1迟滞方式选择 00: 无迟滞 01: 正迟滞 10: 负迟滞 11: 正负迟滞	0x0
11:10	C1_HYSV_S	模拟比较器1迟滞电压选择 00: 无迟滞 01: 10mV 10: 20mV 11: 60mV	0x0
9	C1_POS	模拟比较器1输出极性选择位 0: 正常输出 1: 反相输出	0
8	C1_FE	模拟比较器1输出滤波使能位 0: 禁止 1: 使能	0
7:4	-	保留	-
3:0	C1_FS	模拟比较器1输出滤波时间选择位 0000: (0~1)*Tpclk 0001: (1~2)*Tpclk 0010: (2~3)*Tpclk 0011: (4~5)*Tpclk 0100: (8~9)*Tpclk 0101: (16~17)*Tpclk 0110: (32~33)*Tpclk 0111: (64~65)*Tpclk 1000: (128~129)*Tpclk 1001: (256~257)*Tpclk 1010: (512~513)*Tpclk 其他: (0~1)*Tpclk	0x0

17.5.5 模拟比较器事件控制寄存器(CEVCON)

位	符号	描述	复位值
31:6	-	保留	0
5	EVE1	模拟比较器1事件输出使能位(不影响中断产生) 0: 禁止 1: 使能	0
4	EVE0	模拟比较器0事件输出使能位(不影响中断产生) 0: 禁止 1: 使能	0
3:2	EVS1	模拟比较器1事件产生条件选择位 00: 比较器1输出从0->1的跳变 01: 比较器1输出从1->0的跳变 10: 比较器1输出从0->1的跳变或从1->0的跳变 11: 保留	0x0
1:0	EVS0	模拟比较器0事件产生条件选择位 00: 比较器0输出从0->1的跳变 01: 比较器0输出从1->0的跳变 10: 比较器0输出从0->1的跳变或从1->0的跳变 11: 保留	0x0

17.5.6 模拟比较器写使能控制寄存器(ACMPLOCK)

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	ACMPLOCK[7:0]	ACMP寄存器写使能控制; 0x55: 寄存器写使能; 其他: 写寄存器无效。	0x0

第18章 数模转换器（DAC）

18.1 概述

芯片内部包含一路 8Bit 数模转换器。

DAC 的输入数字信号寄存器为 DAC_S[7:0]（控制器 CON0 的低 8bit），8bit 有效位。

数字信号输入范围为 0x00~0xFF，0x00 对应零模拟量输出 0V。

18.2 特性

模拟参考电压输入可选 VDD 或 VREF_1.45V。

18.3 结构框图

图 18-1：DAC 结构框图



名称	信号类型	备注
VREF	DAC参考电压输入	来源于 VDD或者VREF_1.45V
DAC_S[7:0]	数字信号输入	数字控制器（DACCON0）中设定的电压值
DAC_O	模拟信号输出	DAC通道模拟输出

DAC通道引脚的模拟输出电压公式：
$$DAC\_O = \frac{VREF}{256} \times DAC\_S[7:0]$$

18.4 寄存器映射

(DAC基地址 = 0x4006_8360)RO：只读，WO：只写，R/W：读写

寄存器	偏移量	读/写	描述	复位值
DACCON0(P1B)	0x000	R/W	DAC控制寄存器0	0x0
DACLOCK	0x004	R/W	DAC寄存器使能控制位	0x0

注：

(P1B)标注的寄存器为被保护的寄存器。

(P1B)：LOCK==55H时，标注的寄存器允许写入；=其他值，禁止写入。

18.5 寄存器说明

18.5.1 DAC控制寄存器0(DACCON0)

位	符号	描述	复位值
31:10	-	保留	0x0
9	DAC_S_VREF	DAC参考电压选择 0: 选择VDD 1: 选择VREF_1.45V	0
8	DAC_EN	DAC模块使能 0: 禁止 1: 使能	0
7:0	DAC_S[7:0]	$DAC\_O = (VREF/256) \times DAC\_S[7:0]$ DAC_S: DAC转换值的数据输入, 数据输入范围0x00~0xFF。	0x0

18.5.2 DAC写使能控制寄存器(DACLOCK)

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	DACLOCK[7:0]	DAC寄存器写使能控制; 0x55: 寄存器写使能; 其他: 写寄存器无效。	0x0

第19章 硬件LED矩阵驱动器

19.1 概述

该芯片内集成硬件LED矩阵驱动电路，可方便用户实现LED的显示驱动。

19.2 特性

硬件 LED 矩阵驱动器具有如下特性：

- ◆ 1/4、1/5、1/6、1/8 四种 DUTY 可选。
- ◆ 时钟源固定为 F_{FIX} (8MHz)。
- ◆ 最多支持 8COM*28SEG、6COM*30SEG、5COM*31SEG、4COM*32SEG。
- ◆ 支持调光功能。
- ◆ 死区时间可调。
- ◆ COM 口共阴、共阳两种驱动方式可选。
- ◆ COM 口电流 55mA、220mA 两档可选。
- ◆ SEG 口电流 16 档可选，最大电流可达 50mA。
- ◆ 支持 LED 扫描完成中断。

19.3 寄存器映射

(LED基地址 = 0x4006_8500)RO：只读，WO：只写，R/W：读写

寄存器	偏移量	读/写	描述	复位值
LEDCON	0x00	R/W	LED控制寄存器	0x00
LEDCOMTIME	0x04	R/W	COM口有效时间设置寄存器	0x00
LEDCOMEN	0x08	R/W	COM口使能寄存器	0x00
LEDSEGEN	0x0C	R/W	SEG口使能寄存器	0x00
LEDC0DATA	0x10	R/W	COM0对应SEG数据寄存器	0x00
LEDC1DATA	0x14	R/W	COM1对应SEG数据寄存器	0x00
LEDC2DATA	0x18	R/W	COM2对应SEG数据寄存器	0x00
LEDC3DATA	0x1C	R/W	COM3对应SEG数据寄存器	0x00
LEDC4DATA	0x20	R/W	COM4对应SEG数据寄存器	0x00
LEDC5DATA	0x24	R/W	COM5对应SEG数据寄存器	0x00
LEDC6DATA	0x28	R/W	COM6对应SEG数据寄存器	0x00
LEDC7DATA	0x2C	R/W	COM7对应SEG数据寄存器	0x00
LEDCOMDR	0x30	R/W	COM口灌电流选择寄存器	0x00
LEDSEGRDR	0x34	R/W	SEG口驱动电流选择寄存器	0x00
LEDLOCK	0x38	R/W	LED寄存器写使能控制寄存器	0x00
LEDSEGTIME	0x3C	R/W	调光时间选择寄存器	0x00
LEDDRVEN0	0x40	R/W	LED管脚驱动使能寄存器	0x00

19.4 相关寄存器

19.4.1 LED控制寄存器（LEDCON）

位	符号	描述	复位值
31	LEDEN	LED使能控制位； 0: LED禁止； 1: LED使能。(先配置其他相关设置，再启动LED)	0
30:29	DUTY[1:0]	LED占空比选择位； 00: 1/8DUTY； 01: 1/6DUTY； 10: 1/5DUTY； 11: 1/4DUTY。	0x0
28	CCCA	LED驱动模式选择位； 0: 共阴驱动模式； 1: 共阳驱动模式。	0
27:14	-	保留	0x0
13:12	DTSEL[1:0]	矩阵LED死区时间选择； 00: $8 * T_{FIX}$ ； 01: $16 * T_{FIX}$ ； 10: $32 * T_{FIX}$ ； 11: $64 * T_{FIX}$ 。	0
11:10	-	保留	0x0
9:8	LEDCLK[1:0]	LED计数时钟源选择； 00: $32 * T_{FIX}$ ； 01: $64 * T_{FIX}$ ； 10: $128 * T_{FIX}$ ； 11: $256 * T_{FIX}$ 。	0x0
7:5	-	保留	0x0
4	LEDCLK_EN	LED运行时钟使能控制位； 0: LED运行时钟关闭； 1: LED运行时钟开启。	0
3:0	-	保留	0x0

COM选择说明表

DUTY	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	有效SEG口
11	LED_C0	LED_C1	LED_C2	LED_C3	-	-	-	-	LED_S0-LED_S21 LED_S30-LED_S39
10	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	-	-	-	LED_S1-LED_S21 LED_S30-LED_S39
01	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	-	-	LED_S2-LED_S21 LED_S30-LED_S39
00	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	LED_C6	LED_C7	LED_S4-LED_S21 LED_S30-LED_S39

注：上表中ICOM0-ICOM7为LED内部COM驱动输出信号。LED_C0-LED_C7，LED_S0-LED_S31为LED内部驱动信号最终映射到的管脚名称。

19.4.2 COM口有效时间选择寄存器（LEDCOMTIME）

位	符号	描述	复位值
31:9	-	保留	0x0
7:0	COMT[7:0]	COM口有效时间设置； COM时间 = (COMT<7:0> + 1) * T _{LED_CLK} 。 注：禁止设置成0x00；	0x0

19.4.3 调光时间选择寄存器（LEDSEGTIME）

位	符号	描述	复位值
31:9	-	保留	0x0
7:0	SEGT[7:0]	SEG显示有效时间设置； 调光有效时间 = (SEGT<7:0> + 1) * T _{LED_CLK} 。 注：禁止设置成0x00；	0x0

备注：扫描一个COM的有效显示占空比为：(SEGT<7:0> + 1) / (COMT<7:0> + 1)；SEGT[7:0] ≤ COMT[7:0]。

19.4.4 COM口使能控制寄存器（LEDCOMEN）

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	COMEN[7:0]	LEDC7-LEDC0端口使能控制位； 0: 禁止； 1: 使能。	0x0

19.4.5 SEG口使能控制寄存器（LEDSEGEN）

位	符号	描述	复位值
31:22	SEGEN[31:22]	LEDS39-LEDS30端口使能控制位； 0: 禁止； 1: 使能。	0x0
21:0	SEGEN[21:0]	LEDS21-LEDS0端口使能控制位； 0: 禁止 1: 使能。	0x0

19.4.6 COM口对应SEG数据寄存器（LEDCnDATA）（n=0-7）

位	符号	描述	复位值
31:22	LEDCnDATA[31:22]	COMn（n=0-7）端口有效时，SEG39-SEG30端口数据输出； 0: 低电平； 1: 高电平。	0x0
21:0	LEDCnDATA[21:0]	COMn（n=0-7）端口有效时，SEG21-SEG0端口数据输出； 0: 低电平； 1: 高电平。	0x0

19.4.7 COM口灌电流选择寄存器（LEDCOMDR）

位	符号	描述	复位值
31:8	-	保留	0x0
7	COM7DR	COM7的驱动电流选择; 0: 55mA; 1: 220mA。	0
6	COM6DR	COM6的驱动电流选择; 0: 55mA; 1: 220mA。	0
5	COM5DR	COM5的驱动电流选择; 0: 55mA; 1: 220mA。	0
4	COM4DR	COM4的驱动电流选择; 0: 55mA; 1: 220mA。	0
3	COM3DR	COM3的驱动电流选择; 0: 55mA; 1: 220mA。	0
2	COM2DR	COM2的驱动电流选择; 0: 55mA; 1: 220mA。	0
1	COM1DR	COM1的驱动电流选择; 0: 55mA; 1: 220mA。	0
0	COM0DR	COM0的驱动电流选择; 0: 55mA; 1: 220mA;	0

19.4.8 SEG口驱动电流选择寄存器（LEDSEGDR）

位	符号	描述	复位值
31:28	SEG36-39DR	SEG36-SEG39的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
27:24	SEG32-35DR	SEG32-SEG35的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
23:20	SEG20-21DR/ SEG30-31DR	SEG20-SEG21/SEG30-SEG31的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
19:16	SEG16-19DR	SEG16-SEG19的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
15:12	SEG12-15DR	SEG12-SEG15的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
11:8	SEG8-11DR	SEG8-SEG11的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA; 0101: 19.5mA; 1101: 46.1mA; 0110: 22.1mA; 1110: 49.0mA; 0111: 26.9mA; 1111: 50.0mA.	0x0
7:4	SEG4-7DR	SEG4-SEG7的驱动电流选择; 0000: 0mA; 1000: 26.9mA; 0001: 4.9mA; 1001: 31.8mA; 0010: 7.4mA; 1010: 34.3mA; 0011: 12.2mA; 1011: 39.2mA; 0100: 14.7mA; 1100: 41.6mA;	0x0

		0101: 19.5mA; 0110: 22.1mA; 0111: 26.9mA;	1101: 46.1mA; 1110: 49.0mA; 1111: 50.0mA.	
3:0	SEG0-3DR	SEG0-SEG3的驱动电流选择; 0000: 0mA; 0001: 4.9mA; 0010: 7.4mA; 0011: 12.2mA; 0100: 14.7mA; 0101: 19.5mA; 0110: 22.1mA; 0111: 26.9mA;		0x0
		1000: 26.9mA; 1001: 31.8mA; 1010: 34.3mA; 1011: 39.2mA; 1100: 41.6mA; 1101: 46.1mA; 1110: 49.0mA; 1111: 50.0mA.		

19.4.9 LED管脚驱动使能寄存器 (LEDDRVEN0)

位	符号	描述	复位值
31	LEDDRVEN0_31	矩阵驱动模式SEG39(PA6)/软件驱动PA6管脚电流驱动使能位; 1: PA6管脚的拉电流驱动由LEDSEGDR[31:28]寄存器配置; 0: PA6管脚的拉电流驱动为默认值 (最大)。	0
30	LEDDRVEN0_30	矩阵驱动模式SEG38(PA5)/软件驱动PA5管脚电流驱动使能位; 1: PA5管脚的拉电流驱动由LEDSEGDR[31:28]寄存器配置; 0: PA5管脚的拉电流驱动为默认值 (最大)。	0
29	LEDDRVEN0_29	矩阵驱动模式SEG37(PA4)/软件驱动PA4管脚电流驱动使能位; 1: PA4管脚的拉电流驱动由LEDSEGDR[31:28]寄存器配置; 0: PA4管脚的拉电流驱动为默认值 (最大)。	0
28	LEDDRVEN0_28	矩阵驱动模式SEG36(PA3)/软件驱动PA3管脚电流驱动使能位; 1: PA3管脚的拉电流驱动由LEDSEGDR[31:28]寄存器配置; 0: PA3管脚的拉电流驱动为默认值 (最大)。	0
27	LEDDRVEN0_27	矩阵驱动模式SEG35(PA2)/软件驱动PA2管脚电流驱动使能位; 1: PA2管脚的拉电流驱动由LEDSEGDR[27:24]寄存器配置; 0: PA2管脚的拉电流驱动为默认值 (最大)。	0
26	LEDDRVEN0_26	矩阵驱动模式SEG34(PA1)/软件驱动PA1管脚电流驱动使能位; 1: PA1管脚的拉电流驱动由LEDSEGDR[27:24]寄存器配置; 0: PA1管脚的拉电流驱动为默认值 (最大)。	0
25	LEDDRVEN0_25	矩阵驱动模式SEG33(PA0)/软件驱动PA0管脚电流驱动使能位; 1: PA0管脚的拉电流驱动由LEDSEGDR[27:24]寄存器配置; 0: PA0管脚的拉电流驱动为默认值 (最大)。	0
24	LEDDRVEN0_24	矩阵驱动模式SEG32(PD15)/软件驱动PD15管脚电流驱动使能位; 1: PD15管脚的拉电流驱动由LEDSEGDR[27:24]寄存器配置; 0: PD15管脚的拉电流驱动为默认值 (最大)。	0
23	LEDDRVEN0_23	矩阵驱动模式SEG31(PD14)/软件驱动PD14管脚电流驱动使能位; 1: PD14管脚的拉电流驱动由LEDSEGDR[23:20]寄存器配置; 0: PD14管脚的拉电流驱动为默认值 (最大)。	0
22	LEDDRVEN0_22	矩阵驱动模式SEG30(PD13)/软件驱动PD13管脚电流驱动使能位; 1: PD13管脚的拉电流驱动由LEDSEGDR[23:20]寄存器配置; 0: PD13管脚的拉电流驱动为默认值 (最大)。	0
21	LEDDRVEN0_21	矩阵驱动模式SEG21(PD4)/软件驱动PD4管脚电流驱动使能位; 1: PD4管脚的拉电流驱动由LEDSEGDR[23:20]寄存器配置; 0: PD4管脚的拉电流驱动为默认值 (最大)。	0
20	LEDDRVEN0_20	矩阵驱动模式SEG20(PD3)/软件驱动PD3管脚电流驱动使能位; 1: PD3管脚的拉电流驱动由LEDSEGDR[23:20]寄存器配置; 0: PD3管脚的拉电流驱动为默认值 (最大)。	0
19	LEDDRVEN0_19	矩阵驱动模式SEG19(PD2)/软件驱动PD2管脚电流驱动使能位; 1: PD2管脚的拉电流驱动由LEDSEGDR[19:16]寄存器配置; 0: PD2管脚的拉电流驱动为默认值 (最大)。	0
18	LEDDRVEN0_18	矩阵驱动模式SEG18(PD1)/软件驱动PD1管脚电流驱动使能位; 1: PD1管脚的拉电流驱动由LEDSEGDR[19:16]寄存器配置; 0: PD1管脚的拉电流驱动为默认值 (最大)。	0

17	LEDDRVEN0_17	矩阵驱动模式SEG17(PD0)/软件驱动PD0管脚电流驱动使能位; 1: PD0管脚的拉电流驱动由LEDSEGDR[19:16]寄存器配置; 0: PD0管脚的拉电流驱动为默认值 (最大)。	0
16	LEDDRVEN0_16	矩阵驱动模式SEG16(PC15)/软件驱动PC15管脚电流驱动使能位; 1: PC15管脚的拉电流驱动由LEDSEGDR[19:16]寄存器配置; 0: PC15管脚的拉电流驱动为默认值 (最大)。	0
15	LEDDRVEN0_15	矩阵驱动模式SEG15(PC14)/软件驱动PC14管脚电流驱动使能位; 1: PC14管脚的拉电流驱动由LEDSEGDR[15:12]寄存器配置; 0: PC14管脚的拉电流驱动为默认值 (最大)。	0
14	LEDDRVEN0_14	矩阵驱动模式SEG14(PC13)/软件驱动PC13管脚电流驱动使能位; 1: PC13管脚的拉电流驱动由LEDSEGDR[15:12]寄存器配置; 0: PC13管脚的拉电流驱动为默认值 (最大)。	0
13	LEDDRVEN0_13	矩阵驱动模式SEG13(PC12)/软件驱动PC12管脚电流驱动使能位; 1: PC12管脚的拉电流驱动由LEDSEGDR[15:12]寄存器配置; 0: PC12管脚的拉电流驱动为默认值 (最大)。	0
12	LEDDRVEN0_12	矩阵驱动模式SEG12(PC11)/软件驱动PC11管脚电流驱动使能位; 1: PC11管脚的拉电流驱动由LEDSEGDR[15:12]寄存器配置; 0: PC11管脚的拉电流驱动为默认值 (最大)。	0
11	LEDDRVEN0_11	矩阵驱动模式SEG11(PC10)/软件驱动PC10管脚电流驱动使能位; 1: PC10管脚的拉电流驱动由LEDSEGDR[11:8]寄存器配置; 0: PC10管脚的拉电流驱动为默认值 (最大)。	0
10	LEDDRVEN0_10	矩阵驱动模式SEG10(PC9)/软件驱动PC9管脚电流驱动使能位; 1: PC9管脚的拉电流驱动由LEDSEGDR[11:8]寄存器配置; 0: PC9管脚的拉电流驱动为默认值 (最大)。	0
9	LEDDRVEN0_9	矩阵驱动模式SEG9(PC8)/软件驱动PC8管脚电流驱动使能位; 1: PC8管脚的拉电流驱动由LEDSEGDR[11:8]寄存器配置; 0: PC8管脚的拉电流驱动为默认值 (最大)。	0
8	LEDDRVEN0_8	矩阵驱动模式SEG8(PC7)/软件驱动PC7管脚电流驱动使能位; 1: PC7管脚的拉电流驱动由LEDSEGDR[11:8]寄存器配置; 0: PC7管脚的拉电流驱动为默认值 (最大)。	0
7	LEDDRVEN0_7	矩阵驱动模式SEG7(PC5)/软件驱动PC5管脚电流驱动使能位; 1: PC5管脚的拉电流驱动由LEDSEGDR[7:4]寄存器配置; 0: PC5管脚的拉电流驱动为默认值 (最大)。	0
6	LEDDRVEN0_6	矩阵驱动模式SEG6(PC4)/软件驱动PC4管脚电流驱动使能位; 1: PC4管脚的拉电流驱动由LEDSEGDR[7:4]寄存器配置; 0: PC4管脚的拉电流驱动为默认值 (最大)。	0
5	LEDDRVEN0_5	矩阵驱动模式SEG5(PC3)/软件驱动PC3管脚电流驱动使能位; 1: PC3管脚的拉电流驱动由LEDSEGDR[7:4]寄存器配置; 0: PC3管脚的拉电流驱动为默认值 (最大)。	0
4	LEDDRVEN0_4	矩阵驱动模式SEG4(PC2)/软件驱动PC2管脚电流驱动使能位; 1: PC2管脚的拉电流驱动由LEDSEGDR[7:4]寄存器配置; 0: PC2管脚的拉电流驱动为默认值 (最大)。	0
3	LEDDRVEN0_3	矩阵驱动模式SEG3(PB11)/软件驱动PB11管脚电流驱动使能位; 1: PB11管脚的拉电流驱动由LEDSEGDR[3:0]寄存器配置; 0: PB11管脚的拉电流驱动为默认值 (最大)。	0
2	LEDDRVEN0_2	矩阵驱动模式SEG2(PB10)/软件驱动PB10管脚电流驱动使能位; 1: PB10管脚的拉电流驱动由LEDSEGDR[3:0]寄存器配置; 0: PB10管脚的拉电流驱动为默认值 (最大)。	0
1	LEDDRVEN0_1	矩阵驱动模式SEG1(PB9)/软件驱动PB9管脚电流驱动使能位; 1: PB9管脚的拉电流驱动由LEDSEGDR[3:0]寄存器配置; 0: PB9管脚的拉电流驱动为默认值 (最大)。	0
0	LEDDRVEN0_0	矩阵驱动模式SEG0(PB8)/软件驱动PB8管脚电流驱动使能位; 1: PB8管脚的拉电流驱动由LEDSEGDR[3:0]寄存器配置; 0: PB8管脚的拉电流驱动为默认值 (最大)。	0

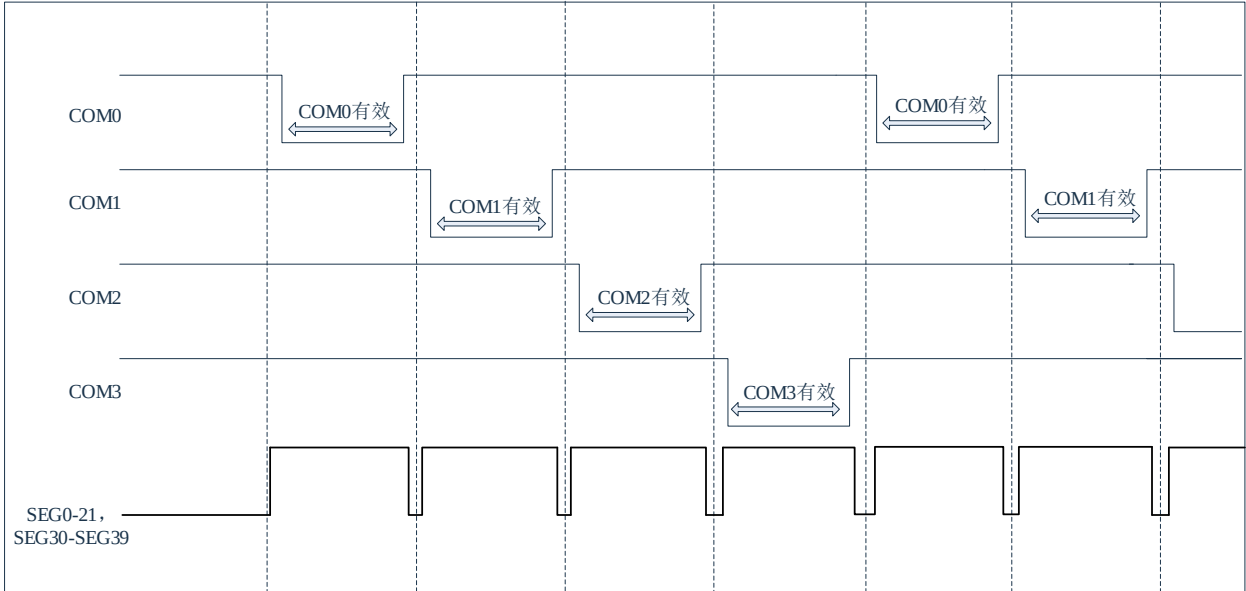
19.4.10 LED寄存器写使能控制寄存器（LEDLOCK）

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	LEDLOCK[7:0]	LED寄存器写使能控制： 0x55: 寄存器写使能； 其他: 写寄存器无效。	0x0

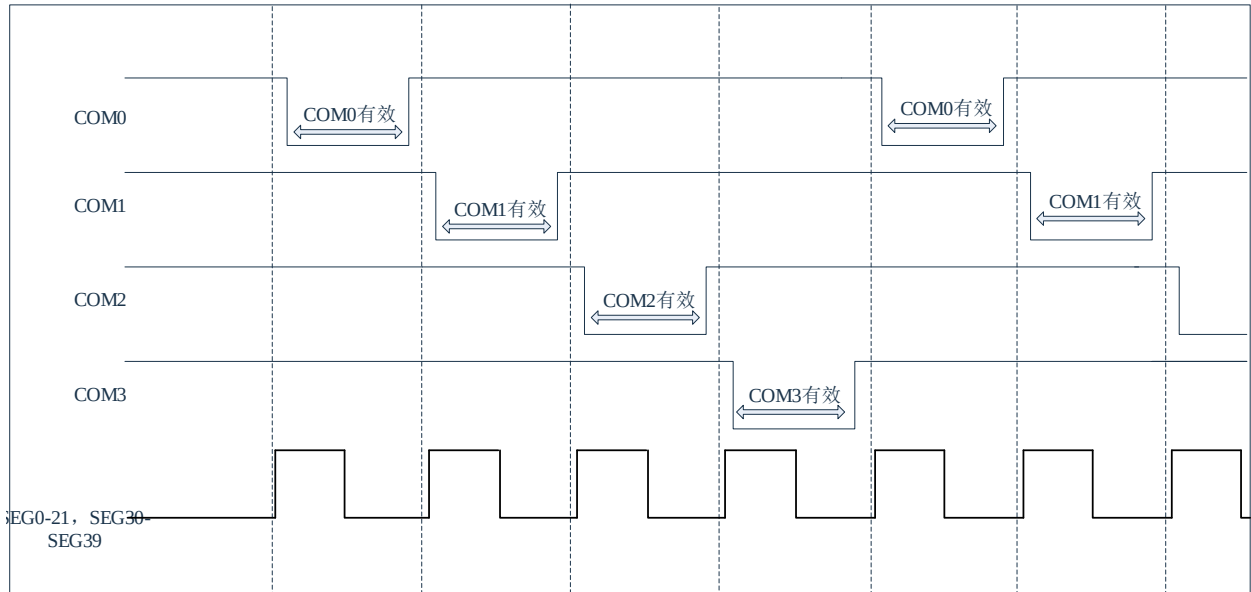
19.5 LED驱动器输出波形

根据 LED 驱动器相关配置寄存器，可设置对应的 LED 驱动器输出波形。

LED 配置 1/4DUTY、共阴驱动模式、seg 数据全为 1、seg 输出占空比为 1 的波形如下图所示：



LED 配置 1/4DUTY、共阴驱动模式、seg 数据全为 1、seg 输出占空比为 1/2 的波形如下图所示：



第20章 硬件LCD驱动器

20.1 概述

硬件LCD驱动器包含一个控制器、一个占空比发生器以及COM和SEG输出端口。

硬件LCD驱动器支持传统电阻和快速充电两种模式，偏置电阻可选60K Ω 、225K Ω 、900K Ω 三种。快速充电模式是一种新的设计方式，当选择225K Ω ，900K Ω 电阻模式时，会有一段时间选择到60K Ω ，这段时间后再自动切换到225K Ω 、900K Ω ，这样既能减小系统功耗，又能保证LCD的亮度。

LCD驱动器的时钟源支持系统时钟、LSI、XT1三种；其中LSI和XT1可在休眠条件下驱动LCD。

20.2 特性

LCD 驱动器具有如下特性：

- ◆ 最大支持的 LCD 通道：8COM x 36SEG、6COM x 38SEG、5COM x 39SEG、4COM x 40SEG。
- ◆ 支持对比度调节。
- ◆ 偏置电压可选：1/2、1/3、1/4。
- ◆ 占空比可选：1/4、1/5、1/6、1/8。
- ◆ 三种时钟源可选：系统时钟、LSI、XT1。
- ◆ 支持传统电阻和快速充电两种模式。
- ◆ 快速充电时间可选。
- ◆ 支持 LCD 对应 IO 的使能控制。
- ◆ 支持 LCD 中断。

20.3 寄存器映射

(LCD基地址 = 0x4006_8400)RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LCDCON0	0x00	R/W	LCD控制寄存器0	0x00
LCDCON1	0x04	R/W	LCD控制寄存器1	0x00
LCDCOMEN	0x08	R/W	COM口使能寄存器	0x00
LCDSEGEN	0x0C	R/W	SEG口使能寄存器	0x00
LCDSEG0	0x10	R/W	SEG0数据寄存器	0x00
LCDSEG1	0x14	R/W	SEG1数据寄存器	0x00
LCDSEG2	0x18	R/W	SEG2数据寄存器	0x00
LCDSEG3	0x1C	R/W	SEG3数据寄存器	0x00
LCDSEG4	0x20	R/W	SEG4数据寄存器	0x00
LCDSEG5	0x24	R/W	SEG5数据寄存器	0x00
LCDSEG6	0x28	R/W	SEG6数据寄存器	0x00
LCDSEG7	0x2C	R/W	SEG7数据寄存器	0x00
LCDSEG8	0x30	R/W	SEG8数据寄存器	0x00
LCDSEG9	0x34	R/W	SEG9数据寄存器	0x00
LCDSEG10	0x38	R/W	SEG10数据寄存器	0x00
LCDSEG11	0x3C	R/W	SEG11数据寄存器	0x00
LCDSEG12	0x40	R/W	SEG12数据寄存器	0x00
LCDSEG13	0x44	R/W	SEG13数据寄存器	0x00
LCDSEG14	0x48	R/W	SEG14数据寄存器	0x00
LCDSEG15	0x4C	R/W	SEG15数据寄存器	0x00
LCDSEG16	0x50	R/W	SEG16数据寄存器	0x00
LCDSEG17	0x54	R/W	SEG17数据寄存器	0x00
LCDSEG18	0x58	R/W	SEG18数据寄存器	0x00
LCDSEG19	0x5C	R/W	SEG19数据寄存器	0x00
LCDSEG20	0x60	R/W	SEG20数据寄存器	0x00
LCDSEG21	0x64	R/W	SEG21数据寄存器	0x00
LCDSEG22	0x68	R/W	SEG22数据寄存器	0x00
LCDSEG23	0x6C	R/W	SEG23数据寄存器	0x00
LCDSEG24	0x70	R/W	SEG24数据寄存器	0x00
LCDSEG25	0x74	R/W	SEG25数据寄存器	0x00
LCDSEG26	0x78	R/W	SEG26数据寄存器	0x00
LCDSEG27	0x7C	R/W	SEG27数据寄存器	0x00
LCDSEG28	0x80	R/W	SEG28数据寄存器	0x00
LCDSEG29	0x84	R/W	SEG29数据寄存器	0x00
LCDSEG30	0x88	R/W	SEG30数据寄存器	0x00
LCDSEG31	0x8C	R/W	SEG31数据寄存器	0x00
LCDSEG32	0x90	R/W	SEG32数据寄存器	0x00
LCDSEG33	0x94	R/W	SEG33数据寄存器	0x00
LCDSEG34	0x98	R/W	SEG34数据寄存器	0x00
LCDSEG35	0x9C	R/W	SEG35数据寄存器	0x00
LCDSEG36	0xA0	R/W	SEG36数据寄存器	0x00
LCDSEG37	0xA4	R/W	SEG37数据寄存器	0x00
LCDSEG38	0xA8	R/W	SEG38数据寄存器	0x00
LCDSEG39	0xAC	R/W	SEG39数据寄存器	0x00

LCDIOEN0	0xB0	R/W	LCDIO使能控制寄存器0	0x00
LCDIOEN1	0xB4	R/W	LCDIO使能控制寄存器1	0x00
LCDLOCK	0xB8	R/W	LCD寄存器写使能寄存器	0x00

20.4 相关寄存器

20.4.1 LCD控制寄存器（LCDCON0）

位	符号	描述	复位值
31	LCDEN	LCD使能控制位； 0: LCD禁止； 1: LCD使能。(先配置其他相关设置，再启动LCD)	0
30	-	保留	0
29:28	LCDDM[1:0]	LCD显示模式； 00: SEG全关 01: SEG全开 1x: 正常输出	0x0
27	COM_SEL	COM口选择（DUTY[1:0]=11时有效）； 控制详见COM选择说明表。	0
26	COM_MOD	模式选择； 0: MODE0 1: MODE1	0
25:24	DUTY[1:0]	LCD占空比选择位； 00: 1/8DUTY 01: 1/6DUTY 10: 1/5DUTY 11: 1/4DUTY	0x0
23:6	-	保留	0x0
5:4	CLKSEL[1:0]	LCD时钟源F _{LCD} 选择； 00: 系统时钟 01: XT1 1x: LSI	0x0
3:0	LCDPSC[3:0]	LCD时钟分频比选择； 0000: F _{LCD} /64; 1000: F _{LCD} /16384; 0001: F _{LCD} /128; 1001: F _{LCD} /32768; 0010: F _{LCD} /256; 1010: F _{LCD} /65536; 0011: F _{LCD} /512; 1011: F _{LCD} /65536; 0100: F _{LCD} /1024; 1100: F _{LCD} /65536; 0101: F _{LCD} /2048; 1101: F _{LCD} /65536; 0110: F _{LCD} /4096; 1110: F _{LCD} /65536; 0111: F _{LCD} /8192; 1111: F _{LCD} /65536。	0x0

COM 选择说明表

DUTY	COM_MOD	COM_SEL	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	有效SEG口
11	1	1	LCD_C3	LCD_C2	LCD_C1	LCD_C0	-	-	-	-	LCD_S0-39
		0	LCD_C7	LCD_C6	LCD_C5	LCD_C4	-	-	-	-	LCD_S4-39
	0	1	LCD_C4	LCD_C5	LCD_C6	LCD_C7	-	-	-	-	LCD_S4-39
		0	LCD_C0	LCD_C1	LCD_C2	LCD_C3	-	-	-	-	LCD_S0-39
10	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	-	-	-	LCD_S4-39
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	-	-	-	LCD_S1-39
01	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	LCD_C2	-	-	LCD_S4-39
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	-	-	LCD_S2-39
00	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	LCD_C2	LCD_C1	LCD_C0	LCD_S4-39
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	LCD_C6	LCD_C7	LCD_S4-39

注：上表中 ICOM0-ICOM7 为 LCD 内部 COM 驱动输出信号。LCD_C0-7，LCD_S0-39 为 LCD 内部驱动信号最终映射到的端口。

20.4.2 LCD控制寄存器（LCDCON1）

位	符号	描述	复位值
31	LCDTEN	LCD电源电压选择; 0: LCD电压由VDD提供; 1: LCD电压由内部电源V _{LCD} 提供。	0
30	-	保留	0
29:28	BIAS[1:0]	LCD显示偏置电压设置; 00: 1/2V _{LCD} ; 01: 1/3V _{LCD} ; 1x: 1/4V _{LCD} 。	0x0
27:24	LCDTVS[3:0]	LCD内部电压选择; 0000~1110: V _{LCD} = (15+LCDTVS<3:0>) *VDD/30; 1111: 保留, 禁止选择。	0x0
23:14	-	保留	0x0
13:12	LCDRM[1:0]	LCD分压电阻选择; 00: 60KΩ; 01: 225KΩ; 1x: 900KΩ。	0x0
11	-	保留	0
10	FCMODE	充电模式选择; 0: 传统电阻型模式; 1: 快速充电模式（当选择225KΩ/900KΩ分压电阻时, 该模式有效）。	0
9:8	FCCTLM[1:0]	快速充电模式时间控制位; 00: 1/8 COM周期; 01: 1/16 COM周期; 10: 1/32 COM周期; 11: 1/64 COM周期。	0x0
7:5	-	保留	0x0
4	EN_LCDBUF	输出模式选择; 0: 低功耗模式输出; 1: BUFFER输出。	0
3	S_LCDBUF	LCD开环闭环输出选择; 0: 闭环BUFFER输出; 1: 开环BUFFER输出。	0
2:0	S_LCDBUFOL_DR [2:0]	驱动能力档位选择; 000~111: 100uA~3mA。	0x0

20.4.3 LCDCOM口使能控制寄存器（LCDCOMEN）

位	符号	描述	复位值
31:24	COMEN[7:0]	LDC_C7-LCD_C0端口使能控制; 0: 禁止; 1: 使能。	0x0
23:8	-	保留	0x0
7:0	SEGEN[39:32]	LCD_S39-LCD_S32端口使能控制; 0: 禁止; 1: 使能。	0x0

20.4.4 LCDSEG口使能控制寄存器（LCDSEGEN）

位	符号	描述	复位值
31:0	SEGEN[31:0]	LCD_S31-LCD_C0端口使能控制；(LCD_S39-LCDS_32使能控制寄存器为LCDCOMEN) 0: 禁止； 1: 使能。	0x0

20.4.5 SEG数据寄存器（LCDSEGN）(n=0~39)

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	ICOM[7:0]	对应LCD_C7-LCD_C0的LCD_Sn端口输出数据； 0: 低电平； 1: 高电平。	0x0

20.4.6 LCDIO口使能控制寄存器0（LCDIOEN0）

位	符号	描述	复位值
31:24	LCDIOEN0[31:24]	LCD_S27-LCD_S20对应的IO使能控制； 0: 禁止； 1: 使能。	0x0
23:16	LCDIOEN0[23:16]	LCD_S19-LCD_S12对应的IO使能控制； 0: 禁止； 1: 使能。	0x0
15:8	LCDIOEN0[15:8]	LCD_S11-LCD_S4对应的IO使能控制； 0: 禁止； 1: 使能。	0x0
7:0	LCDIOEN0[7:0]	LCD_C7-LCD_C0对应的IO使能控制； 其中LCD_S0-LCD_S3和LCD_C4-LCD_C7一致； 0: 禁止； 1: 使能。	0x0

20.4.7 LCDIO口使能控制寄存器0（LCDIOEN1）

位	符号	描述	复位值
31:12	-	保留	0x0
11:8	LCDIOEN1[11:8]	LCD_S39-LCD_S36对应的IO使能控制; 0: 禁止; 1: 使能。	0x0
7:0	LCDIOEN1[7:0]	LCD_S35-LCD_S28对应的IO使能控制; 0: 禁止; 1: 使能。	0x0

硬件LCD的IO使能控制对应的COM、SEG口如下表所示。

表20-1 LCDIOEN设置与LDC_COM、LCDSEG口对应关系表

寄存器位	对应COM/SEG	寄存器位	对应COM/SEG	寄存器位	对应COM/SEG	寄存器位	对应COM/SEG
LCDEN0[0]	COM0	LCDEN0[11]	SEG7	LCDEN0[22]	SEG18	LCDEN1[1]	SEG29
LCDEN0[1]	COM1	LCDEN0[12]	SEG8	LCDEN0[23]	SEG19	LCDEN1[2]	SEG30
LCDEN0[2]	COM2	LCDEN0[13]	SEG9	LCDEN0[24]	SEG20	LCDEN1[3]	SEG31
LCDEN0[3]	COM3	LCDEN0[14]	SEG10	LCDEN0[25]	SEG21	LCDEN1[4]	SEG32
LCDEN0[4]	COM4/SEG0	LCDEN0[15]	SEG11	LCDEN0[26]	SEG22	LCDEN1[5]	SEG33
LCDEN0[5]	COM5/SEG1	LCDEN0[16]	SEG12	LCDEN0[27]	SEG23	LCDEN1[6]	SEG34
LCDEN0[6]	COM6/SEG2	LCDEN0[17]	SEG13	LCDEN0[28]	SEG24	LCDEN1[7]	SEG35
LCDEN0[7]	COM7/SEG3	LCDEN0[18]	SEG14	LCDEN0[29]	SEG25	LCDEN1[8]	SEG36
LCDEN0[8]	SEG4	LCDEN0[19]	SEG15	LCDEN0[30]	SEG26	LCDEN1[9]	SEG37
LCDEN0[9]	SEG5	LCDEN0[20]	SEG16	LCDEN0[31]	SEG27	LCDEN1[10]	SEG38
LCDEN0[10]	SEG6	LCDEN0[21]	SEG17	LCDEN1[0]	SEG28	LCDEN1[11]	SEG39

寄存器位	对应COM/SEG	寄存器位	对应COM/SEG	寄存器位	对应COM/SEG	寄存器位	对应COM/SEG
LCD_PB4	COM0	LCD_PC5	SEG7	LCD_PD1	SEG18	LCD_PD12	SEG29
LCD_PB5	COM1	LCD_PC7	SEG8	LCD_PD2	SEG19	LCD_PD13	SEG30
LCD_PB6	COM2	LCD_PC8	SEG9	LCD_PD3	SEG20	LCD_PD14	SEG31
LCD_PB7	COM3	LCD_PC9	SEG10	LCD_PD4	SEG21	LCD_PD15	SEG32
LCD_PB8	COM4/SEG0	LCD_PC10	SEG11	LCD_PD5	SEG22	LCD_PA0	SEG33
LCD_PB9	COM5/SEG1	LCD_PC11	SEG12	LCD_PD6	SEG23	LCD_PA1	SEG34
LCD_PB10	COM6/SEG2	LCD_PC12	SEG13	LCD_PD7	SEG24	LCD_PA2	SEG35
LCD_PB11	COM7/SEG3	LCD_PC13	SEG14	LCD_PD8	SEG25	LCD_PA3	SEG36
LCD_PC2	SEG4	LCD_PC14	SEG15	LCD_PD9	SEG26	LCD_PA4	SEG37
LCD_PC3	SEG5	LCD_PC15	SEG16	LCD_PD10	SEG27	LCD_PA5	SEG38
LCD_PC4	SEG6	LCD_PD0	SEG17	LCD_PD11	SEG28	LCD_PA6	SEG39

20.4.8 LCD寄存器写使能控制寄存器（LCDLOCK）

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	LCDLOCK[7:0]	LCD寄存器写使能控制； 0x55: 寄存器写使能； 其他: 写寄存器无效。	0x0

20.5 COM-SEG数据表

硬件 LCD 驱动器不同的 DUTY 对应如下数据表。

20.5.1 1/4DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	-	-	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	-	-	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	-	-	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	-	-	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	-	-	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	-	-	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	-	-	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	-	-	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	-	-	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	-	-	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	-	-	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	-	-	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	-	-	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	-	-	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	-	-	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	-	-	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	-	-	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	-	-	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	-	-	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	-	-	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	-	-	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	-	-	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	-	-	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	-	-	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	-	-	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	-	-	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	-	-	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	-	-	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	-	-	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	-	-	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	-	-	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	-	-	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	-	-	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	-	-	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	-	-	SEG33	SEG33	SEG33	SEG33
LCDSEG34	F672H	-	-	-	-	SEG34	SEG34	SEG34	SEG34
LCDSEG35	F673H	-	-	-	-	SEG35	SEG35	SEG35	SEG35

20.5.2 1/5DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	-	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	-	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	-	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	-	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	-	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	-	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	-	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	-	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	-	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	-	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	-	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	-	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	-	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	-	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	-	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	-	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	-	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	-	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	-	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	-	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	-	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	-	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	-	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	-	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	-	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	-	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	-	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	-	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	-	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	-	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	-	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	-	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	-	SEG31	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	-	SEG32	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	-	SEG33	SEG33	SEG33	SEG33	SEG33
LCDSEG34	F672H	-	-	-	SEG34	SEG34	SEG34	SEG34	SEG34

20.5.3 1/6DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	ICOM5	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33

20.5.4 1/8DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		ICOM7	ICOM6	ICOM5	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31

第21章 联动控制器 (EVENTC)

21.1 EVENTC的功能

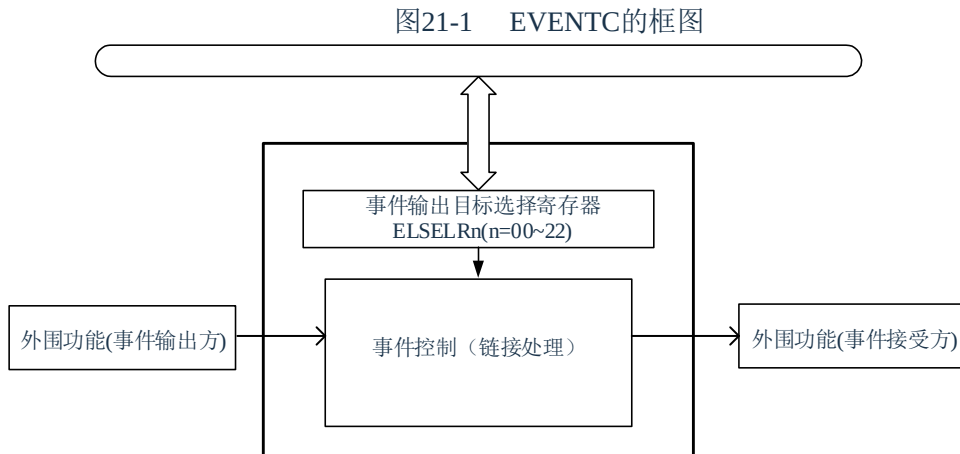
EVENTC将各外围功能输出的事件进行外围功能之间的相互链接。能通过事件链接不经过CPU而直接进行外围功能之间的协作运行。

EVENTC有以下功能：

- 根据产品，能将23种外围功能的事件信号直接链接到指定的外围功能。
- 根据产品，能将事件信号用作4种外围功能中的1种外围功能运行的启动源。

21.2 EVENTC的结构

EVENTC的框图如图21-1所示：



21.3 寄存器映射

(EVENTC基地址 = 4004_3400) RO: 只读, WO: 只写, R/W: 读写

寄存器	地址	读/写	描述	复位值
ELSELR00	0x00	R/W	事件输出目标选择寄存器00	0x00
ELSELR01	0x01	R/W	事件输出目标选择寄存器01	0x00
ELSELR02	0x02	R/W	事件输出目标选择寄存器02	0x00
ELSELR03	0x03	R/W	事件输出目标选择寄存器03	0x00
ELSELR04	0x04	R/W	事件输出目标选择寄存器04	0x00
ELSELR05	0x05	R/W	事件输出目标选择寄存器05	0x00
ELSELR06	0x06	R/W	事件输出目标选择寄存器06	0x00
ELSELR07	0x07	R/W	事件输出目标选择寄存器07	0x00
ELSELR08	0x08	R/W	事件输出目标选择寄存器08	0x00
ELSELR09	0x09	R/W	事件输出目标选择寄存器09	0x00
ELSELR10	0x0A	R/W	事件输出目标选择寄存器10	0x00
ELSELR11	0x0B	R/W	事件输出目标选择寄存器11	0x00
ELSELR12	0x0C	R/W	事件输出目标选择寄存器12	0x00
ELSELR13	0x0D	R/W	事件输出目标选择寄存器13	0x00
ELSELR14	0x0E	R/W	事件输出目标选择寄存器14	0x00
ELSELR15	0x0F	R/W	事件输出目标选择寄存器15	0x00
ELSELR16	0x10	R/W	事件输出目标选择寄存器16	0x00
ELSELR17	0x11	R/W	事件输出目标选择寄存器17	0x00
ELSELR18	0x12	R/W	事件输出目标选择寄存器18	0x00
ELSELR19	0x13	R/W	事件输出目标选择寄存器19	0x00
ELSELR20	0x14	R/W	事件输出目标选择寄存器20	0x00
ELSELR21	0x15	R/W	事件输出目标选择寄存器21	0x00
ELSELR22	0x16	R/W	事件输出目标选择寄存器22	0x00

21.3.1 输出目标选择寄存器n（ELSELRn）（n=00~22）

ELSELRn寄存器将各事件信号链接到事件接受方外围功能（链接目标外围功能）接受事件时的运行。不能将多个事件输入链接到相同的事件输出目标（事件接受方）。否则，事件接受方外围功能的运行可能不定而无法正常地接受事件信号。

另外，不能将事件链接发生源和事件输出目标设置为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设置ELSELRn寄存器。

位	符号	描述	复位值
7:4	-	保留	0x0
3:0	ELSELn3~ELSELn0	0x0: 禁止事件链接。 0x1: 选择所链接的外围功能1的运行 ^{注1} 。 0x2: 选择所链接的外围功能2的运行 ^{注1} 。 0x3: 选择所链接的外围功能3的运行 ^{注1} 。 0x4: 禁止事件链接。 0x5: 禁止事件链接。 0x6: 选择所链接的外围功能6的运行 ^{注1} 。 其他: 禁止事件链接。	0x0

注1: 请参照“表21-1 ELSELRn寄存器（n=00~22）的设置值和链接目标外围功能接受事件时的运行的对应”。

表21-1 ELSELRn寄存器（n=00~22）和外围功能的对应

寄存器名	事件发生源（事件输入n的输出源）	事件内容
ELSELR00	外部中断边沿检测0	INTP0
ELSELR01	外部中断边沿检测1	INTP1
ELSELR02	外部中断边沿检测2	INTP2
ELSELR03	外部中断边沿检测3	INTP3
ELSELR04	外部中断边沿检测4	INTP4
ELSELR05	外部中断边沿检测5	INTP5
ELSELR06	外部中断边沿检测6	INTP6
ELSELR07	外部中断边沿检测7	INTP7
ELSELR08	外部中断边沿检测8	INTP8
ELSELR09	外部中断边沿检测9	INTP9
ELSELR10	外部中断边沿检测10	INTP10
ELSELR11	RTC固定周期/闹钟一致检测	INTRTC
ELSELR12	Timer41通道10的计数结束/捕捉结束	INTTM10
ELSELR13	Timer41通道11的计数结束/捕捉结束	INTTM11
ELSELR14	Timer41通道12的计数结束/捕捉结束	INTTM12
ELSELR15	Timer41通道13的计数结束/捕捉结束	INTTM13
ELSELR16	Timer40通道00的计数结束/捕捉结束	INTTM00
ELSELR17	Timer40通道01的计数结束/捕捉结束	INTTM01
ELSELR18	Timer40通道02的计数结束/捕捉结束	INTTM02
ELSELR19	Timer40通道03的计数结束/捕捉结束	INTTM03
ELSELR20	比较器0检测	INTCMP0
ELSELR21	比较器1检测	INTCMP1
ELSELR22	停振检测	OSDC

表21-2 ELSELRn寄存器（n=00~22）的设置值和链接目标外围功能接受事件时的运行的对应

ELSELRn寄存器的 ELSELn3~ELSELn0位	链接目标NO.	链接目标外围功能	接受事件时的运行
0001B	1	A/D转换器	开始A/D转换。
0010B	2	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
0011B	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
0110B	6	EPWM的截断	产生EPWM的截断

注1：要选择通用定时器0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（TPS0）将通道0的运行时钟设置为 f_{CLK} ，通过通用定时器0中的噪声滤波器允许寄存器将TI00引脚的噪声滤波器置为OFF，并且通过定时器输入选择寄存器0（TIOS0）将通道0使用的定时器输入设置为联动控制器的事件输入信号。

注2：要选择通用定时器0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（TPS0）将通道1的运行时钟设置为 f_{CLK} ，通过通用定时器0中的噪声滤波器允许寄存器TI01引脚的噪声滤波器置为OFF，并且通过定时器输入选择寄存器0（TIOS0）将通道1使用的定时器输入设置为EVENTC的事件输入信号。

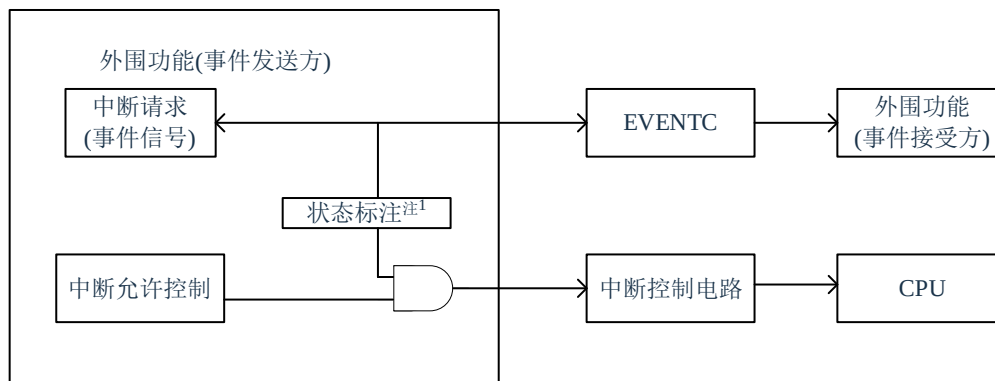
21.4 EVENTC的运行

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21-2所示。此图是以有中断请求状态标志和中断允许位（控制允许或者禁止）的外围功能为例的关系。

通过EVENTC接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行（参照“表21-2 ELSELRn寄存器（n=00~22）的设置值和链接目标外围功能接受事件时的运行的对应”）。

图21-2 中断处理和EVENTC的关系



注1：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21-3所示。

表21-3 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	通用定时器0 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fCLK周期 后进行边沿的检测。
3	通用定时器0 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fCLK周期 后进行边沿的检测。
6	EPWM截断	产生EPWM截断	EVENTC事件直接触发EPWM截断，根据EPWM 设置进行相应的EPWM输出

第22章 嵌套向量中断控制器(NVIC)

Cortex®-M0+ CPU提供一个用于中断处理的嵌套向量中断控制器（NVIC）。

22.1 特性

- 支持嵌套向量中断。
- 自动保存和恢复处理器状态。
- 动态改变优先级。
- 简化和确定的中断时间。

NVIC按照优先级处理所有支持的异常。所有的异常在“Handler模式”处理。NVIC支持32个(IRQ[31:0])离散中断，每个中断支持4级中断优先级。所有的中断和大部分系统异常可以配置成不同的优先级。当一个中断发生时，NVIC将比较新中断与当前中断的优先级，如果新中断优先级高，则立即处理新中断。

当接受一个中断后，中断服务程序（ISR）的开始地址可从内存中的向量表取得。软件不需要决定哪个中断被响应，也不用分配相关ISR的开始地址。当取得开始地址后，NVIC将自动保存处理器状态寄存器（PC、PSR、LR、R0~R3、R12）的值到栈中。在ISR结束后，NVIC将从栈中恢复相关寄存器的值，并运行在正常状态。因此花费少量且确定的时间处理中断请求。

NVIC支持“末尾连锁”，可以有效的处理背对背中断，即无需保存和恢复当前状态，从而减少结束当前ISR切换到挂起ISR的延迟时间。NVIC还支持“Late Arrival”，因此可以提高并发中断的效率。当较高优先级中断请求发生在当前ISR开始执行之前（保存处理器状态和获取起始地址阶段），NVIC将立即处理更高优先级的中断，从而提高实时性。

更多详细信息，请参考“ARM®Cortex®-M0+ 技术参考手册”和“ARM®v6-M 架构参考手册”。

22.2 异常模式和系统中断映射

下表列出了该系列产品支持的异常模式。与所有中断一样，软件可以为其中一些异常设置4级优先级。用户可配置的最高优先级为0，最低优先级为3。所有用户可配置中断的默认优先级为0。

异常名称	异常编号	优先级
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4~10	保留
SVCall	11	可配置
Reserved	12~13	保留
PendSV	14	可配置
SysTick	15	可配置
Interrupt (IRQ0~IRQ31)	16~47	可配置

注：优先级0在系统为第4优先级，排在“Reset”、“NMI”和“Hard Fault”三个系统异常之后。

22.3 向量表

表22-1 中断源一览表(1/2)

编号	中断号	向量地址	异常类型	描述	中断号	向量地址	异常类型	异常类型
1-15	-	0x00-0x3c	系统异常		扩展32个中断			
16	0	0x40	INTLVI	电压检测	32	0x40	INTWWDI	窗口看门狗溢出中断
17	1	0x44	INTP0	引脚输入的边沿检测	33	0x44	INTP6	引脚输入的边沿检测
18	2	0x48	INTP1		34	0x48	INTP7	
19	3	0x4c	INTP2		35	0x4c	INTP8	
20	4	0x50	INTP3		36	0x50	INTP9	
21	5	0x54	INTP4		37	0x54	INTP10	
22	6	0x58	INTP5		38	0x58	INTP11	
23	7	0x5c	INTST0	UART0发送结束	39	0x5c	INTST1	UART1发送结束
			INTSPI00	SPI00传输结束			INTSPI10	SPI10传输结束
			INTIIC00	IIC00传输结束			INTIIC10	IIC10传输结束
24	8	0x60	INTSR0	UART0接收结束	40	0x60	INTSR1	UART1接收结束
			INTSPI01	SPI01传输结束			INTSPI11	SPI11传输结束
			INTIIC01	IIC01传输结束			INTIIC11	IIC11传输结束
25	9	0x64	INTSE0	UART0接收错误	41	0x64	INTSE1	UART1接收错误
26	10	0x68	INTST2	UART2发送结束	42	0x68	INTST3	UART3发送结束
			INTSPI20	SPI20传输结束				
			INTIIC20	IIC20传输结束				
27	11	0x6c	INTSR2	UART2接收结束	43	0x6c	INTSR3	UART3接收结束
			INTSPI21	SPI21传输结束				
			INTIIC21	IIC21传输结束				
28	12	0x70	INTSE2	UART2接收错误	44	0x70	INTSE3	UART3接收错误
29	13	0x74	INTST4	UART4发送结束	45	0x74	INTST5	UART5发送结束
30	14	0x78	INTSR4	UART4接收结束	46	0x78	INTSR5	UART5接收结束
31	15	0x7c	INTSE4	UART4接收错误	47	0x7c	INTSE5	UART5接收错误
32	16	0x80	INTTM00	定时器通道00定时/捕捉结束	48	0x80	INTTM10	定时器通道10定时/捕捉结束
33	17	0x84	INTTM01	定时器通道01定时/捕捉结束	49	0x84	INTTM11	定时器通道11定时/捕捉结束
34	18	0x88	INTTM02	定时器通道02定时/捕捉结束	50	0x88	INTTM12	定时器通道12定时/捕捉结束
35	19	0x8c	INTTM03	定时器通道03定时/捕捉结束	51	0x8c	INTTM13	定时器通道13定时/捕捉结束
36	20	0x90	INTTM01H	定时器通道01定时/捕捉结束 (高8位工作)	52	0x90	INTTM10H	定时器通道11定时/捕捉结束 (高8位工作)
37	21	0x94	INTTM03H	定时器通道03定时/捕捉结束 (高8位工作)	53	0x94	INTTM13H	定时器通道13定时/捕捉结束 (高8位工作)
38	22	0x98	INTRTC	RTC中断	54	0x98	INTIT	间隔定时器中断
39	23	0x9c	INTCMP0	ACMP0中断	55	0x9c	INTOSDC	停振检测中断
40	24	0xa0	INTCMP1	ACMP1中断	56	0xa0	INTDTC0	DMA通道0完成
41	25	0xa4	INTAD	AD转换结束	57	0xa4	INTDTC1	DMA通道1完成
42	26	0xa8	INTLED	LED帧扫描结束	58	0xa8	INTDTC2	DMA通道2完成
43	27	0xac	INTLCD	LCD帧扫描结束	59	0xac	INTDTC3	DMA通道3完成
44	28	0xb0	INTTOUCH	触摸扫描结束	60	0xb0	INTDTC4	DMA通道4完成
45	29	0xb4	INTDIV	除法器计算结束	61	0xb4	INTDTC5	DMA通道5完成

46	30	0xb8	INTIICA	IICA通信接受	62	0xb8	INTDTC6	DMA通道6完成
47	31	0xbc	INTFL	FLASH编程完成/FLASH错误	63	0xbc	INTDTC7	DMA通道7完成

表22-1 中断源一览表(2/2)

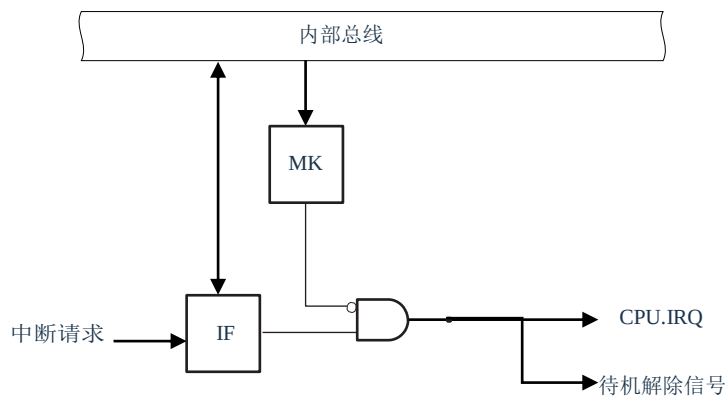
中断处理	中断源编号	中断源		内部/外部	基本结构类型注1
		名称	触发		
不可屏蔽	—	INTWDT	独立看门狗定时器间隔中断注2	内部	(D)

注：1.基本构成类型(A)~(D)分别对应图22-1的(A)~(D)。
2.这是将选项字节（000C0H）的bit7（WDTINT）置“1”的情况。

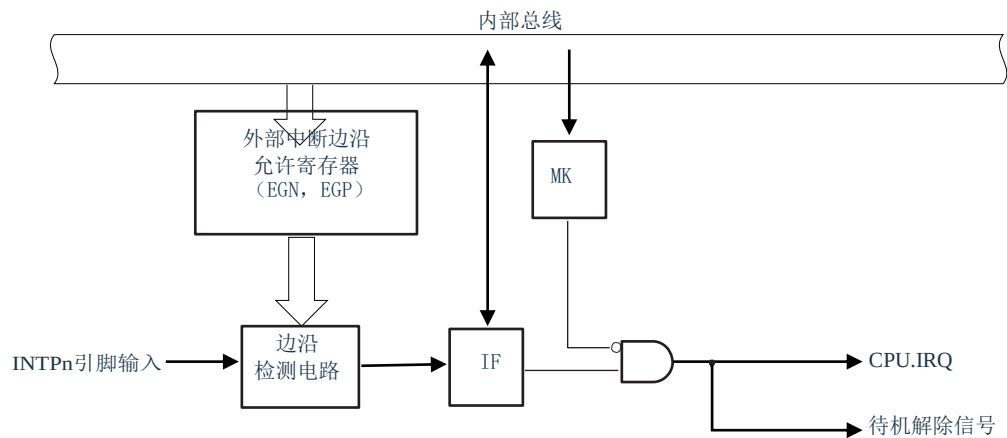
22.4 中断功能的基本结构

图22-1 中断功能的基本结构

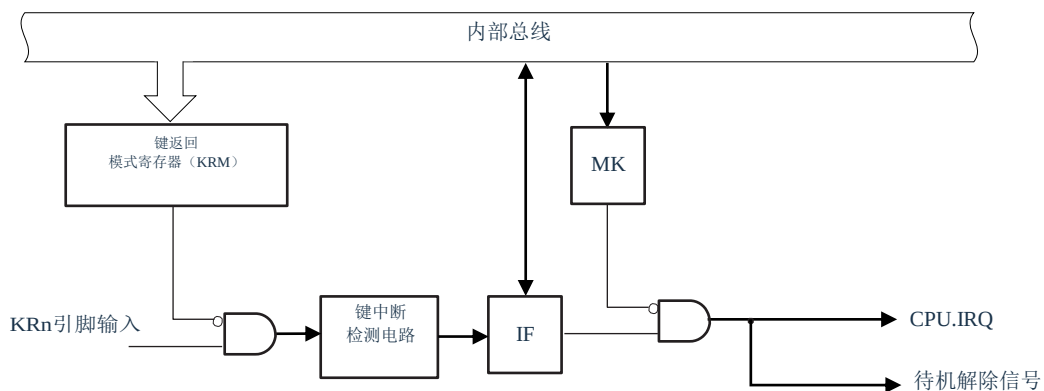
A. 内部可屏蔽中断



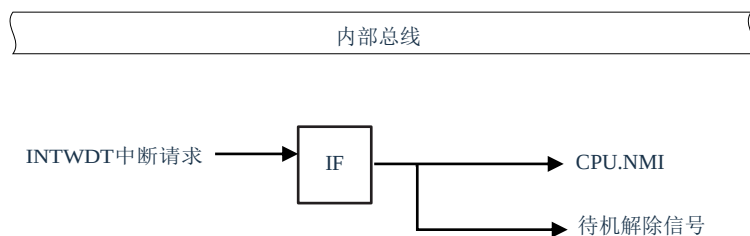
B. 外部可屏蔽中断 (INTPn)



C. 外部可屏蔽中断 (INTKR)



D. 不可屏蔽中断



注：不可屏蔽中断的中断请求标志IF没有实体寄存器，不能通过总线读写寄存器来产生中断请求。

22.5 寄存器映射

(NVIC基地址 = 0x4000_6000H) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
IF00	0x000	R/W	中断请求标志寄存器	0x0
IF01	0x004	R/W	中断请求标志寄存器	0x0
IF02	0x008	R/W	中断请求标志寄存器	0x0
IF03	0x00C	R/W	中断请求标志寄存器	0x0
IF04	0x010	R/W	中断请求标志寄存器	0x0
IF05	0x014	R/W	中断请求标志寄存器	0x0
IF06	0x018	R/W	中断请求标志寄存器	0x0
IF07	0x01C	R/W	中断请求标志寄存器	0x0
IF08	0x020	R/W	中断请求标志寄存器	0x0
IF09	0x024	R/W	中断请求标志寄存器	0x0
IF10	0x028	R/W	中断请求标志寄存器	0x0
IF11	0x02C	R/W	中断请求标志寄存器	0x0
IF12	0x030	R/W	中断请求标志寄存器	0x0
IF13	0x034	R/W	中断请求标志寄存器	0x0
IF14	0x038	R/W	中断请求标志寄存器	0x0
IF15	0x03C	R/W	中断请求标志寄存器	0x0
IF16	0x040	R/W	中断请求标志寄存器	0x0
IF17	0x044	R/W	中断请求标志寄存器	0x0
IF18	0x048	R/W	中断请求标志寄存器	0x0
IF19	0x04C	R/W	中断请求标志寄存器	0x0
IF20	0x050	R/W	中断请求标志寄存器	0x0
IF21	0x054	R/W	中断请求标志寄存器	0x0
IF22	0x058	R/W	中断请求标志寄存器	0x0
IF23	0x05C	R/W	中断请求标志寄存器	0x0
IF24	0x060	R/W	中断请求标志寄存器	0x0
IF25	0x064	R/W	中断请求标志寄存器	0x0
IF26	0x068	R/W	中断请求标志寄存器	0x0
IF27	0x06C	R/W	中断请求标志寄存器	0x0
IF28	0x070	R/W	中断请求标志寄存器	0x0
IF29	0x074	R/W	中断请求标志寄存器	0x0
IF30	0x078	R/W	中断请求标志寄存器	0x0
IF31	0x07C	R/W	中断请求标志寄存器	0x0
MK00	0x100	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK01	0x104	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK02	0x108	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK03	0x10C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK04	0x110	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK05	0x114	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK06	0x118	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK07	0x11C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK08	0x120	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK09	0x124	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK10	0x128	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK11	0x12C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF

MK12	0x130	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK13	0x134	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK14	0x138	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK15	0x13C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK16	0x140	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK17	0x144	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK18	0x148	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK19	0x14C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK20	0x150	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK21	0x154	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK22	0x158	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK23	0x15C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK24	0x160	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK25	0x164	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK26	0x168	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK27	0x16C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK28	0x170	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK29	0x174	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK30	0x178	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF
MK31	0x17C	R/W	中断请求屏蔽寄存器	0xFFFF_FFFF

(以下寄存器基地址 = 0x4004_5B30) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
EGP0	0x000	R/W	外部中断上升沿允许寄存器	0x0
EGN0	0x001	R/W	外部中断下降沿允许寄存器	0x0
EGP1	0x002	R/W	外部中断上升沿允许寄存器	0x0
EGN1	0x003	R/W	外部中断下降沿允许寄存器	0x0

22.6 寄存器说明

22.6.1 中断请求标志寄存器（IF00~IF31）

寄存器描述：

位	符号	描述	复位值
31:9	-	保留	0x0
8	IFmH	编号 32~63 的中断源的中断请求标志： 0: 不产生中断请求信号 1: 产生中断请求，处于中断请求状态	0
7:1	-	保留	0
0	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	0

注：

1. 中断源与中断请求标志寄存器的对应关系见表22-2。
2. 中断请求标志寄存器与CPU.IRQ的对应关系见图22-2。
3. 中断请求标志寄存器不会自清零，中断响应后必须将寄存器写0。

22.6.2 中断屏蔽标志寄存器 (MK00~MK31)

位	符号	描述	复位值
31:9	-	保留	0xFFFFFE
8	MKmH	编号 32~63 的中断源的中断处理控制: 0: 允许中断处理 1: 禁止中断处理	1
7:1	-	保留	0xFE
0	MKmL	编号 0~31 的中断源的中断处理控制: 0: 允许中断处理 1: 禁止中断处理	1

注:

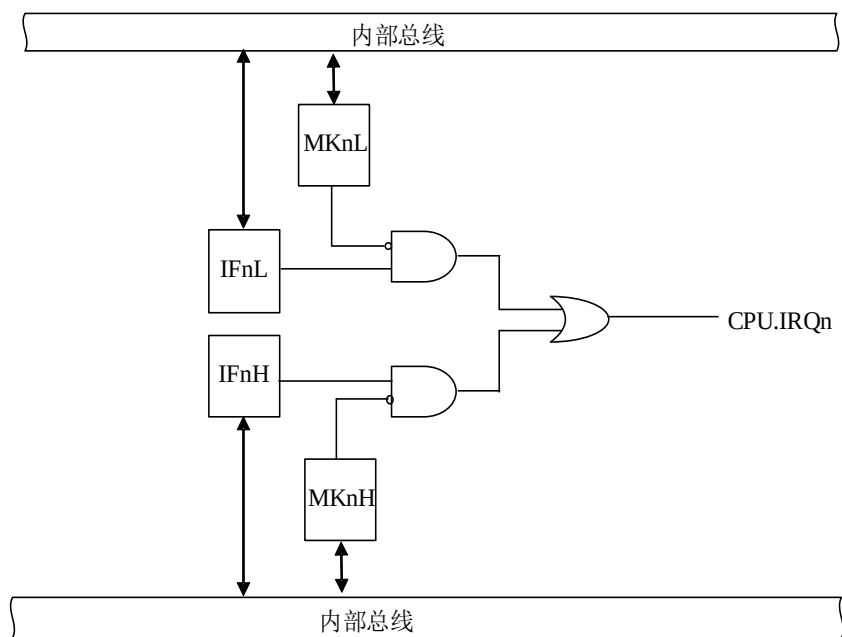
1. 中断源与中断请求屏蔽寄存器的对应关系见表22-2.
2. 中断请求屏蔽寄存器与CPU.IRQ的对应关系见图22-2.

表22-2 中断源和各标志寄存器的对应关系

编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器	编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器
0	INTLVI	IF00.IFL	MK00.MKL	32	INTWWDIT	IF00.IFH	MK00.MKH
1	INTP0	IF01.IFL	MK01.MKL	33	INTP6	IF01.IFH	MK01.MKH
2	INTP1	IF02.IFL	MK02.MKL	34	INTP7	IF02.IFH	MK02.MKH
3	INTP2	IF03.IFL	MK03.MKL	35	INTP8	IF03.IFH	MK03.MKH
4	INTP3	IF04.IFL	MK04.MKL	36	INTP9	IF04.IFH	MK04.MKH
5	INTP4	IF05.IFL	MK05.MKL	37	INTP10	IF05.IFH	MK05.MKH
6	INTP5	IF06.IFL	MK06.MKL	38	INTP11	IF06.IFH	MK06.MKH
7	INTST0 INTSSPI0 INTIIC00	IF07.IFL	MK07.MKL	39	INTST1 INTSSPI10 INTIIC10	IF07.IFH	MK07.MKH
8	INTSR0 INTSSPI01 INTIIC01	IF08.IFL	MK08.MKL	40	INTSR11 INTSSPI11 INTIIC11	IF08.IFH	MK08.MKH
9	INTSE0	IF09.IFL	MK09.MKL	41	INTSE1	IF09.IFH	MK09.MKH
10	INTST2	IF10.IFL	MK10.MKL	42	INTST3	IF10.IFH	MK10.MKH
11	INTSR2	IF11.IFL	MK11.MKL	43	INTSR3	IF11.IFH	MK11.MKH
12	INTSE2	IF12.IFL	MK12.MKL	44	INTSE3	IF12.IFH	MK12.MKH
13	INTST4	IF13.IFL	MK13.MKL	45	INTST5	IF13.IFH	MK13.MKH
14	INTSR4	IF14.IFL	MK14.MKL	46	INTSR5	IF14.IFH	MK14.MKH
15	INTSE4	IF15.IFL	MK15.MKL	47	INTSE5	IF15.IFH	MK15.MKH
16	INTTM00	IF16.IFL	MK16.MKL	48	INTTM10	IF16.IFH	MK16.MKH
17	INTTM01	IF17.IFL	MK17.MKL	49	INTTM11	IF17.IFH	MK17.MKH
18	INTTM02	IF18.IFL	MK18.MKL	50	INTTM12	IF18.IFH	MK18.MKH
19	INTTM03	IF19.IFL	MK19.MKL	51	INTTM13	IF19.IFH	MK19.MKH
20	INTTM01H	IF20.IFL	MK20.MKL	52	INTTM10H	IF20.IFH	MK20.MKH
21	INTTM03H	IF21.IFL	MK21.MKL	53	INTTM13H	IF21.IFH	MK21.MKH
22	INTRTC	IF22.IFL	MK22.MKL	54	INTIT	IF22.IFH	MK22.MKH
23	INTCMP0	IF23.IFL	MK23.MKL	55	INTOSDC	IF23.IFH	MK23.MKH

24	INTCMP1	IF24.IFL	MK24.MKL	56	INTDTC0	IF24.IFH	MK24.MKH
25	INTAD	IF25.IFL	MK25.MKL	57	INTDTC1	IF25.IFH	MK25.MKH
26	INTLED	IF26.IFL	MK26.MKL	58	INTDTC2	IF26.IFH	MK26.MKH
27	INTLCD	IF27.IFL	MK27.MKL	59	INTDTC3	IF27.IFH	MK27.MKH
28	INTTOUCH	IF28.IFL	MK28.MKL	60	INTDTC4	IF28.IFH	MK28.MKH
29	INTDIV	IF29.IFL	MK29.MKL	61	INTDTC5	IF29.IFH	MK29.MKH
30	INTIICA	IF30.IFL	MK30.MKL	62	INTDTC6	IF30.IFH	MK30.MKH
31	INTFL	IF31.IFL	MK31.MKL	63	INTDTC7	IF31.IFH	MK31.MKH

图22-2 各标志寄存器与CPU.IRQ的关系



22.6.3 外部中断上升沿允许寄存器 (EGP0)

EGP0和EGN0寄存器用于设定INTP0~INTP7的有效边沿。通过8位存储器操作指令设定EGP0、EGN0寄存器。在产生复位信号后，这些寄存器的值变为“00H”。

位	符号	描述	复位值
7	EGP7	INTP7 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
6	EGP6	INTP6 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
5	EGP5	INTP5 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
4	EGP4	INTP4 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
3	EGP3	INTP3 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
2	EGP2	INTP2 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
1	EGP1	INTP1 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
0	EGP0	INTP0 外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0

22.6.4 外部中断下降沿允许寄存器 (EGN0)

位	符号	描述	复位值
7	EGN7	INTP7 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
6	EGN6	INTP6 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
5	EGN5	INTP5 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
4	EGN4	INTP4 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
3	EGN3	INTP3 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
2	EGN2	INTP2 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
1	EGN1	INTP1 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
0	EGN0	INTP0 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0

22.6.5 外部中断上升沿允许寄存器 (EGP1)

EGP1和EGN1寄存器用于设定INTP8~INTP11的有效边沿。通过8位存储器操作指令设定EGP1、EGN1寄存器。

INTP00~INTP11共用一个数字滤波器，支持软件修改滤波时间。

在产生复位信号后，这些寄存器的值变为“00H”。

位	符号	描述	复位值
7:6	FILCNT	滤波时钟个数 0x0: 1个滤波时钟长度 0x1: 2个滤波时钟长度 0x2: 3个滤波时钟长度 0x3: 4个滤波时钟长度	0x0
5:4	FILCLK	滤波时钟filclk频率设置 0x0: FILCLK = PCLK 0x1: FILCLK = PCLK /4 0x2: FILCLK = PCLK /16 0x3: FILCLK = PCLK /64	0x0
3	EGP11	INTP11外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
2	EGP10	INTP10外部中断上升沿允许寄存器：： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
1	EGP9	INTP9外部中断上升沿允许寄存器： 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0
0	EGP8	INTP8外部中断上升沿允许寄存 0: 禁止外部中断上升沿 1: 允许外部中断上升沿	0

注：1. PCLK 指APB总线时钟

22.6.6 外部中断下降沿允许寄存器 (EGN1)

位	符号	描述	复位值
7:4	--	保留	0x0
3	EGN11	INTP11 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
2	EGN10	INTP10 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
1	EGN9	INTP9 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0
0	EGN8	INTP8 外部中断下降沿允许寄存器: 0: 禁止外部中断下降沿 1: 允许外部中断下降沿	0

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22-3所示。

表22-3 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
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7
EGP8	EGN8	INTP8
EGP9	EGN9	INTP9
EGP10	EGN10	INTP10
EGP11	EGN11	INTP11

注1: 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”。

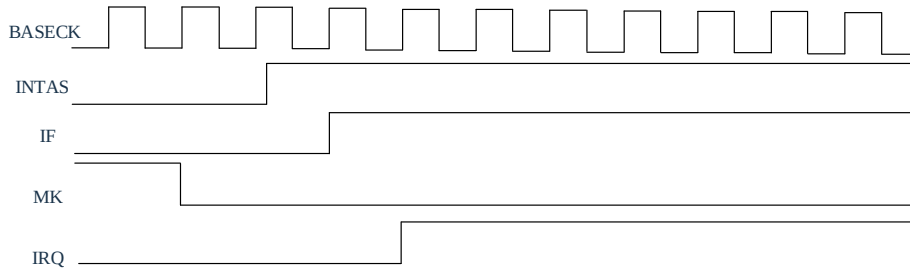
注2: 有关边沿检测的端口, 请参照“2.1端口功能”。

22.7 中断处理的操作

22.7.1 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。

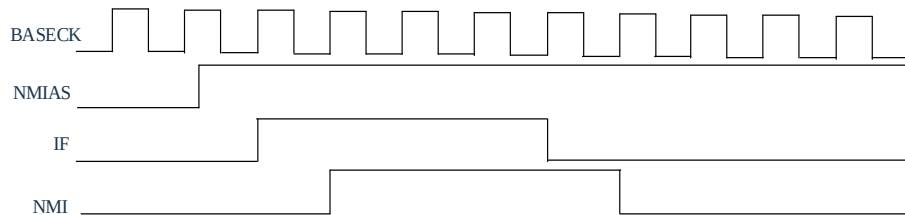
从中断请求标志被置1，到CPU的IRQ被置1，只需要1个时钟。



22.7.2 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23章 待机功能

23.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下2种模式。

(1) 睡眠模式

睡眠模式是停止 CPU 运行时钟的模式。在设定睡眠模式前，如果高速内部振荡器或者低速内部振荡器正在振荡，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，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设定为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注1：在转移到深度睡眠模式时，必须在停止以系统时钟运行的外围硬件后执行WFI指令。

注2：为了减小A/D转换器的工作电流，等待ADC当次转换完成后，再停止A/D运行后执行WFI指令。

注3：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28章 选项字节”。

23.2 睡眠模式

23.2.1 睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 0 时，执行 WFI 指令，就进入了睡眠模式。在睡眠模式，CPU 停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，振荡器等在睡眠模式下的状态见表 23-1。

无论设定前的 CPU 时钟是高速内部振荡器时钟或者低速内部振荡器时钟，都能设定睡眠模式。

注意：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23-1 睡眠模式中的运行状态

项目		睡眠模式
系统时钟	f_{IH}	继续运行
	f_{IL}	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态，保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM0		停止运行
SRAM1		停止运行
端口(锁存器)		保持休眠之前的设定
停振检测(OSDC)		能运行
通用定时单元(TIMER40/41)		能运行
15 位间隔定时器		能运行
实时时钟(RTC)		能运行
时钟输出/蜂鸣器输出		能运行
窗口看门狗定时器		休眠前设定了继续计数则休眠后可以继续计数
独立看门狗定时器		能运行
增强型 PWM(EPWM)		能运行
串行接口(IICA)		能运行
通用串行通信单元(SAU0/1/2/3/4)		能运行
数据传送控制器(DMA)		能运行
快速模数转换(ADC)		能运行
模拟比较器(ACMP0/1)		能运行
数模转换(DAC)		能运行
TOUCH		能运行
LED		能运行
LCD		能运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	能运行
	通用 CRC	停止运行
SFR 保护功能		停止运行

注1：停止运行：在转移到睡眠模式时停止运行。

注2： f_{IH} ：高速内部振荡器时钟。

注3： f_{IL} ：低速内部振荡器时钟。

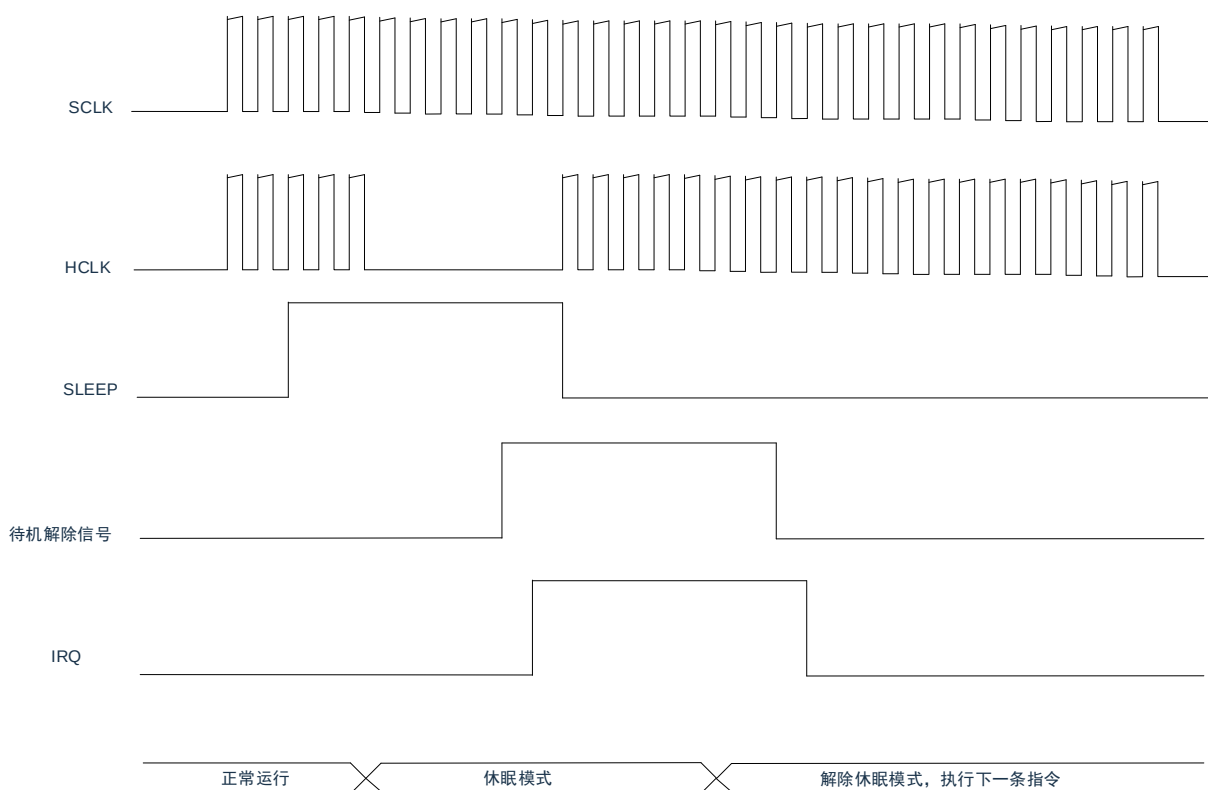
23.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位，POR 复位，低电压检测复位，WWDT 复位以及独立看门狗复位解除。

(1) 通过中断解除

当产生一个中断请求，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23-1 通过中断请求解除睡眠模式



注1：从待机解除信号产生到睡眠模式解除，开始执行中断服务程序，需要16个时钟。

注2：部分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。

(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23-2 通过复位解除睡眠模式



注1：有关复位处理，请参照“第24章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第25章 上电复位电路”。

23.3 深度睡眠模式

23.3.1 深度睡眠模式的设定

在 SCR 寄存器的 SLEEPDEEP 位为 1 时，执行 WFI 指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及振荡器都停止运行。但是，CPU 内部寄存器的值，RAM 数据，周边模块，I/O 的状态被保持。周边模块，振荡器在深度睡眠模式的运行状态见表 23-2。

注意 当中断请求标志为“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23-2 深度睡眠模式中的运行状态

项目		深度睡眠模式
系统时钟	f _H	停止运行
	f _L	通过 OSMC 寄存器和 SUBCKSEL 寄存器设置运行状态，保持设定好的状态
CPU		停止运行
代码闪存		停止运行
SRAM0		停止运行
SRAM1		停止运行
端口(锁存器)		保持休眠之前的设定
停振检测(OSDC)		禁止运行
通用定时单元(TIMER40/41)		禁止运行
15 位间隔定时器		能运行
实时时钟(RTC)		能运行
时钟输出/蜂鸣器输出		禁止运行
窗口看门狗定时器		禁止运行
独立看门狗定时器		能运行
增强型 PWM(EPWM)		禁止运行
通用串行通信单元(SAU0/1/2/3/4)		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
快速模数转换(ADC)		禁止运行
模拟比较器(ACMP0/1)		禁止运行
数模转换(DAC)		禁止运行
TOUCH		禁止运行
LED		禁止运行
LCD		能运行
上电复位功能		能运行
电压检测功能		能运行
外部中断		能运行
CRC 运算	高速 CRC	停止运行
	通用 CRC	停止运行

SFR 保护功能	停止运行

注1：停止运行：在转移到深度睡眠模式时停止运行。

禁止运行：在转移到深度睡眠模式前停止运行。

注2： f_{IH} ：高速内部振荡器时钟。

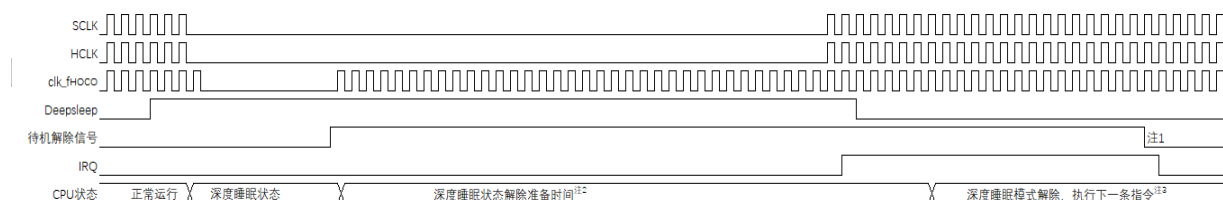
注3： f_{IL} ：低速内部振荡器时钟。

23.3.2 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 LVD 检测、INTP0-11、15 位间隔定时器定时器或者独立看门狗定时器中断请求，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。



注1：待机解除信号：有关待机解除信号的详细内容，请参照中断相关内容。

注2：深度睡眠状态解除准备时间：进入深度睡眠模式前CPU时钟为高速内部振荡时时：至少20us。

注3：等待：从CPU.IRQ有效到开始执行中断服务程序，需要14个时钟。

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23-3 通过复位解除深度睡眠模式



注1：有关复位处理，请参照“第24章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理，请参照“第25章 上电复位电路”。

第24章 复位功能

以下8种方法产生复位信号。

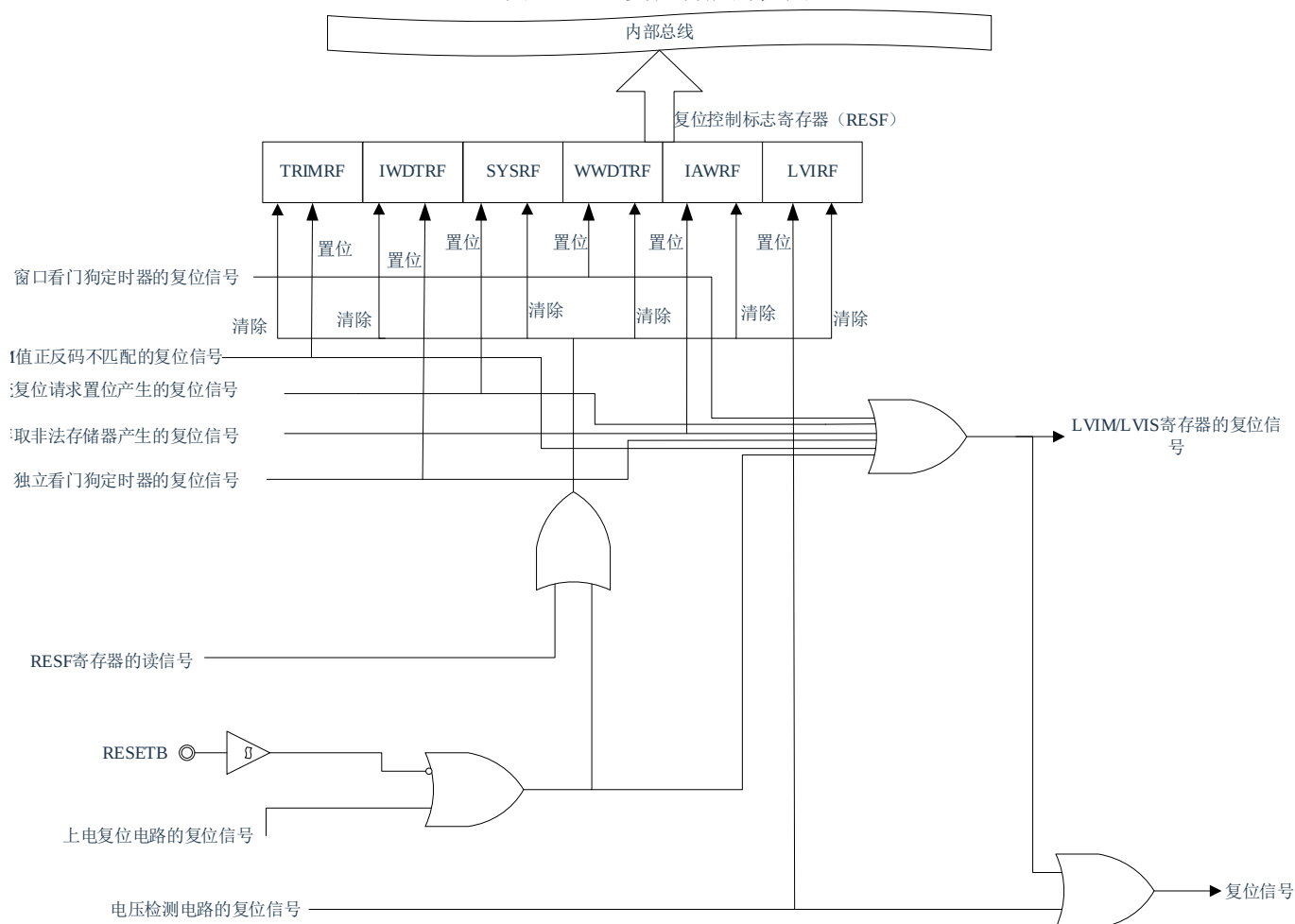
- (1) 通过 RESETB 引脚输入外部复位。
- (2) 通过独立看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。
- (5) 因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为 1 而产生内部复位。
- (6) 存取非法存储器而产生内部复位。
- (7) TRIM 修调的正反码复位
- (8) 通过窗口看门狗定时器的程序失控检测产生内部复位

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，或者独立看门狗定时器检测到程序失控，或者窗口看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者系统复位请求位被置位，或者存取非法存储器时，产生复位并且各硬件变为如表 24-1 所示的状态。

注 1：在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在用户手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。

图24-1 复位功能的框图



注1: LVD电路的内部复位不会对LVD电路进行复位。

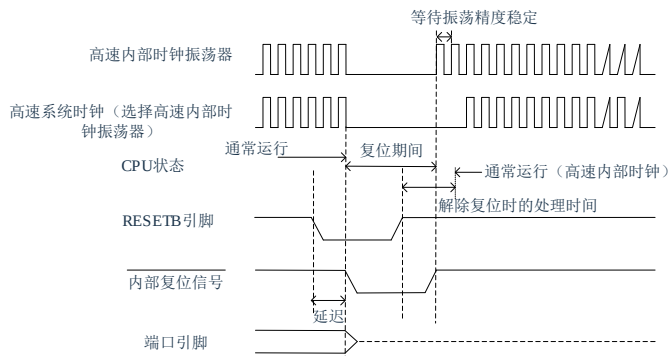
注2: LVIM: 电压检测寄存器。

注3: LVIS: 电压检测电平寄存器。

24.1 复位时序

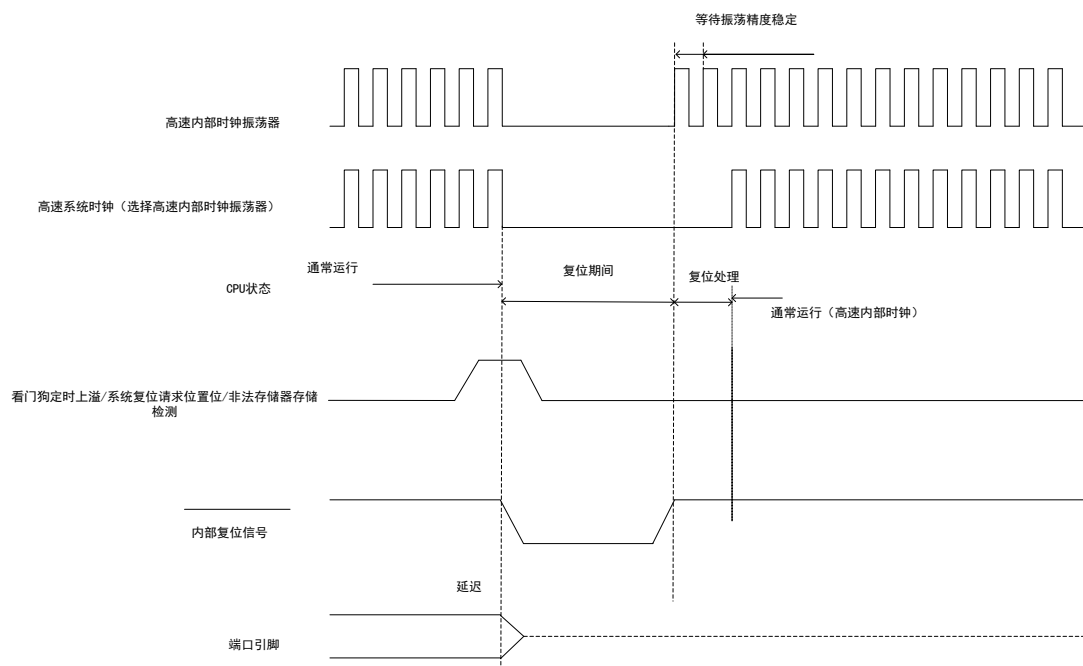
当给 RESETB 引脚输入低电平时，产生复位。然后，如果给 RESETB 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 24-2 RESETB 输入的复位时序



对于因窗口看门狗定时器的上溢、独立看门狗定时器的上溢、系统复位请求位的置位、TRIM 修调的正反码复位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 24-3 因窗口看门狗定时器的上溢、独立看门狗定时器上溢、系统复位请求位的置位、TRIM 修调的正反码复位或者非法存储器存取的检测而产生的复位时序。



注1：端口引脚PB1，PC11，PC12，PD1，PD2变为以下状态：

- 在外部复位期间PB1为外部输入信号确定，PC12，PD2为高电平，PC11，PD1为低电平。
- POR复位的期间PB1，PC12，PD2为高电平。PC11，PD1为低电平。

注2：看门狗定时器2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 25 章 上电复位电路”和“第 26 章 电压检测电路”。

备注：VPOR：POR 电源电压上升检测电压

VLVD: LVD 检测电压。

表 24-1 复位期间的运行状态

项目		复位期间						
		外部复位 RESINB	独立看门狗 定时器复位	窗口看门狗 复位	上电复位	LVD 电压 检测复位	写复位寄存 器复位	存取非法存 储器复位
系统时钟	f_{H}	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	f_{L}	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CPU		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
代码闪存		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SRAM0		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SRAM1		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
端口（锁存器）		PC12,PD2 上电后为高 电平, PC11,PD1 为低电平其 他脚复位前 中后均为高 阻态。	PB1,PC12,P D2 上电后 为高电平, PC11,PD1 为低电平其 他脚复位前 中后均为高 阻态。	PB1,PC12,P D2 上电后 为高电 平,PC11,PD 1 为低电平 其他脚复位 前中后均为 高阻态。	PB1,PC12, PD2 上电 后高电 平,PC11,P D1 为低电 平其他脚 复位前中 后均为高 阻态。	PB1,PC12,P D2 上电后 为高电 平,PC11,PD 1 为低电 平其他脚 复位前中 后均为高 阻态。	PB1,PC12,P D2 上电后 为高, PC11,PD1 为低电平其 他脚复位前 中后均为高 阻态。	PB1,PC12,P D2 上电后 为高, PC11,PD1 为低电平其 他脚复位前 中后均为高 阻态。
硬件除法器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
通用定时单元 TIMER4		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
15 位间隔定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
实时时钟（RTC）		能运行	能运行	能运行	停止运行	能运行	能运行	能运行
时钟输出/蜂鸣器输出		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
窗口看门狗定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
独立看门狗定时器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
增强型 PWM(EPWM)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
通用串行单元		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
I2C 串行接口控制器(IICA)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
数据传送控制器（DMA）		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
A/D 转换器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
模拟比较器(ACMP0/1)		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
DAC		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
硬件 LED 矩阵驱动器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
硬件 LCD 矩阵驱动器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
联动控制器		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
ADCLDO		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
上电复位功能		能进行检测 运行	能进行检测 运行	能进行检测 运行	能进行检测 运行	能进行检测 运行	能进行检测 运行	能进行检测 运行
电压检测功能		不能运行	不能运行	不能运行	不能运行	能运行	不能运行	不能运行
外部中断		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
CRC 运算功 能	高速 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
	通用 CRC	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
SFR 保护功能		停止运行	停止运行	停止运行	停止运行	停止运行	停止运行	停止运行
复位完成后上电状态(选项 字节配置)		重新配置	重新配置	重新配置	重新配置	重新配置	重新配置	重新配置

注1: f_{H} : 高速内部振荡器时钟; f_{L} : 低速内部振荡器时钟。

24.2 确认复位源的寄存器

24.2.1 寄存器映射

(复位控制基地址 = 0x4002_0440) RO: 只读; WO: 只写; R/W: 读写。

寄存器	偏移量	读/写	描述	复位值
RESF	0x000	RO	复位控制标志寄存器	-

24.2.2 复位控制标志寄存器(RESF)

该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、WWDTRF、WDTRF、IAWRF、LVIRF标志。要判断复位源时，必须将RESF寄存器的值保存到任意的RAM，然后通过其RAM值进行判断。

位	符号	描述	复位值
7	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清除了 RESF 寄存器。 1: 产生内部复位请求。	-
6	TRIMRF	TRIM 修调值正反码不匹配而产生的内部复位请求 0: 没有产生内部复位请求或清除了 RESF 寄存器。 1: 产生内部复位请求。	-
5	IWDTRF	独立看门狗定时器（IWDG）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。	-
4	WWDTRF	窗口看门狗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。	-
3	OSDCRF	发振停止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。	-
2	-	保留	
1	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。	-
0	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：RESF寄存器的状态因复位源而不同。请参照表24-2。

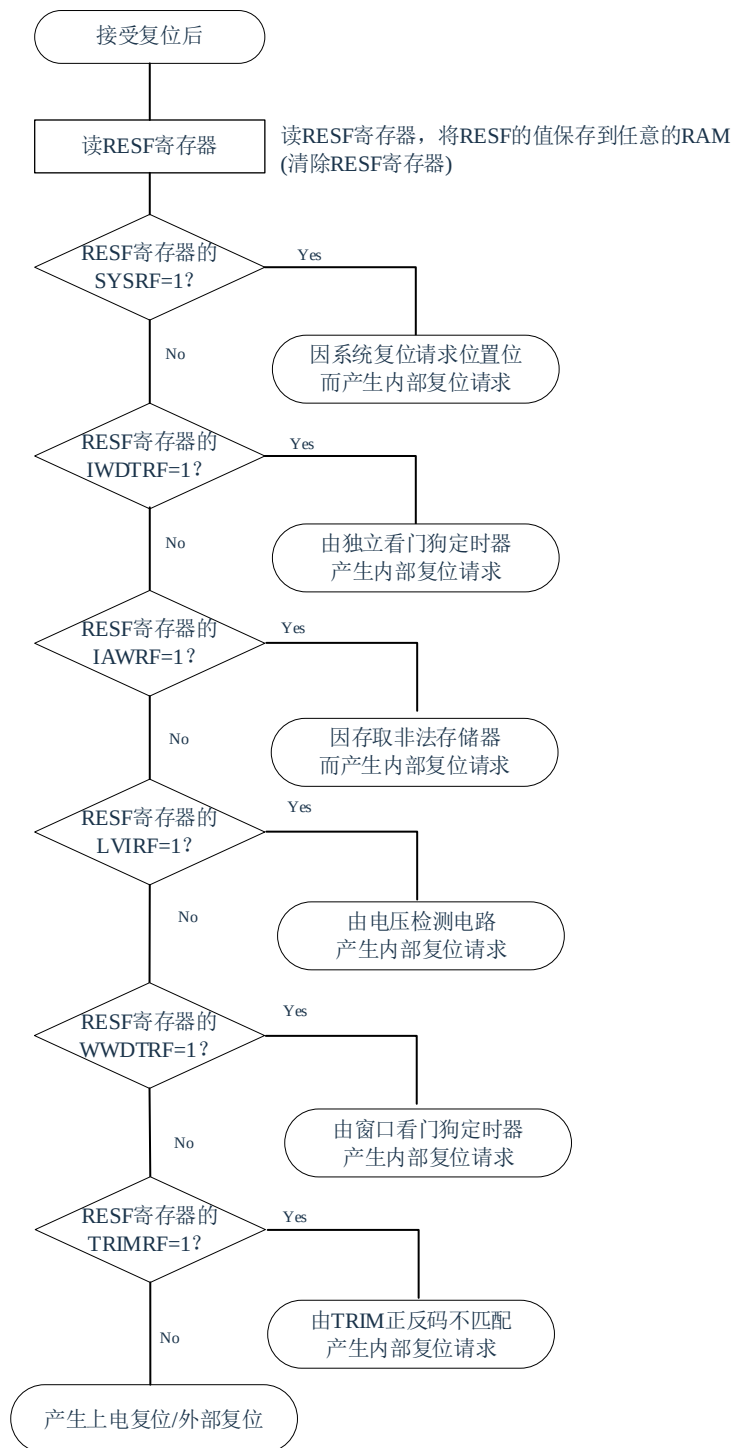
发生复位请求时的RESF寄存器状态如下表所示。

表24-2 发生复位请求时的RESF寄存器状态

复位源 标志	RESETB 输入	POR产生的 复位	TRIM修调正 反码不匹配产 生的复位	系统复位请求 位置位产生的 复位	IWDTP产生 的复位	WWDTP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”	保持	置“1”	保持	保持	保持	保持	
IWDTRF				保持	置“1”		置“1”		
IAWRF					保持			保持	
LVIRF					保持	置“1”			
WWDTRF			保持						
TRIMRF									

复位源的确认步骤如图24-3所示。

图24-3 复位源的确认步骤



第25章 上电复位电路

25.1 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 1：当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

注 2：芯片内置多个产生内部复位信号的硬件。当由独立看门狗定时器，窗口看门狗定时器（WWDT）、电压检测（LVD）电路、系统复位请求位置位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；当由独立看门狗定时器,WWDT、LVD、系统复位请求位的置位或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清“00H”，而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 24 章 复位功能”。

注 3： V_{POR} ：POR 电源电压上升检测电压

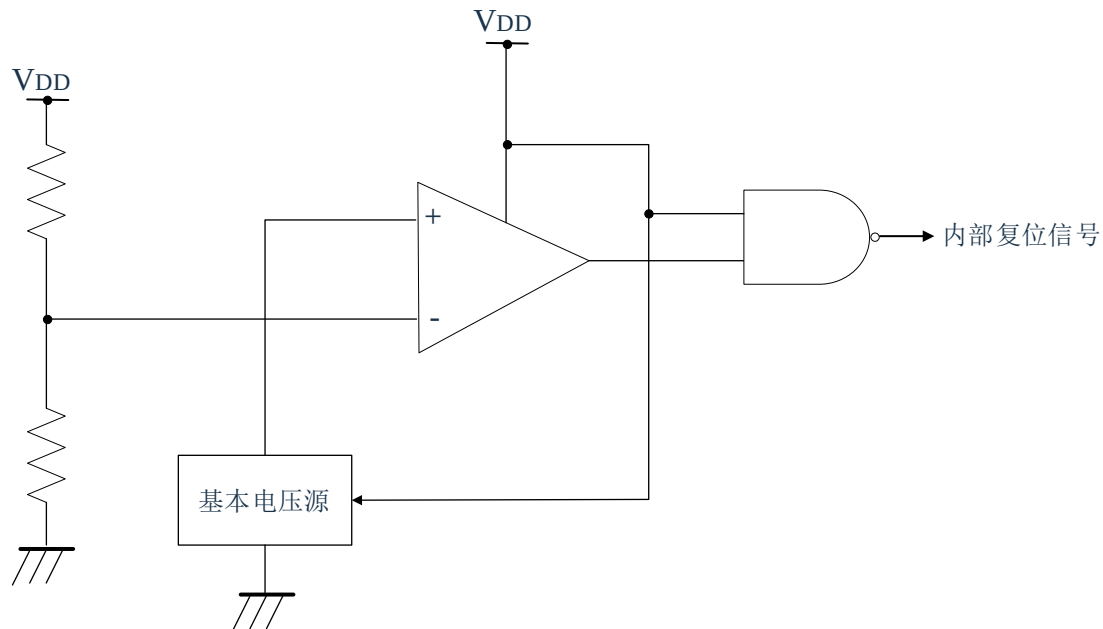
注 4： V_{PDR} ：POR 电源电压下降检测电压

注 5：有关 V_{POR} 、 V_{PDR} 详细内容请参照数据手册的 POR 电路特性。

25.2 上电复位电路的结构

上电复位电路的框图如图25-1所示。

图25-1 上电复位电路的框图

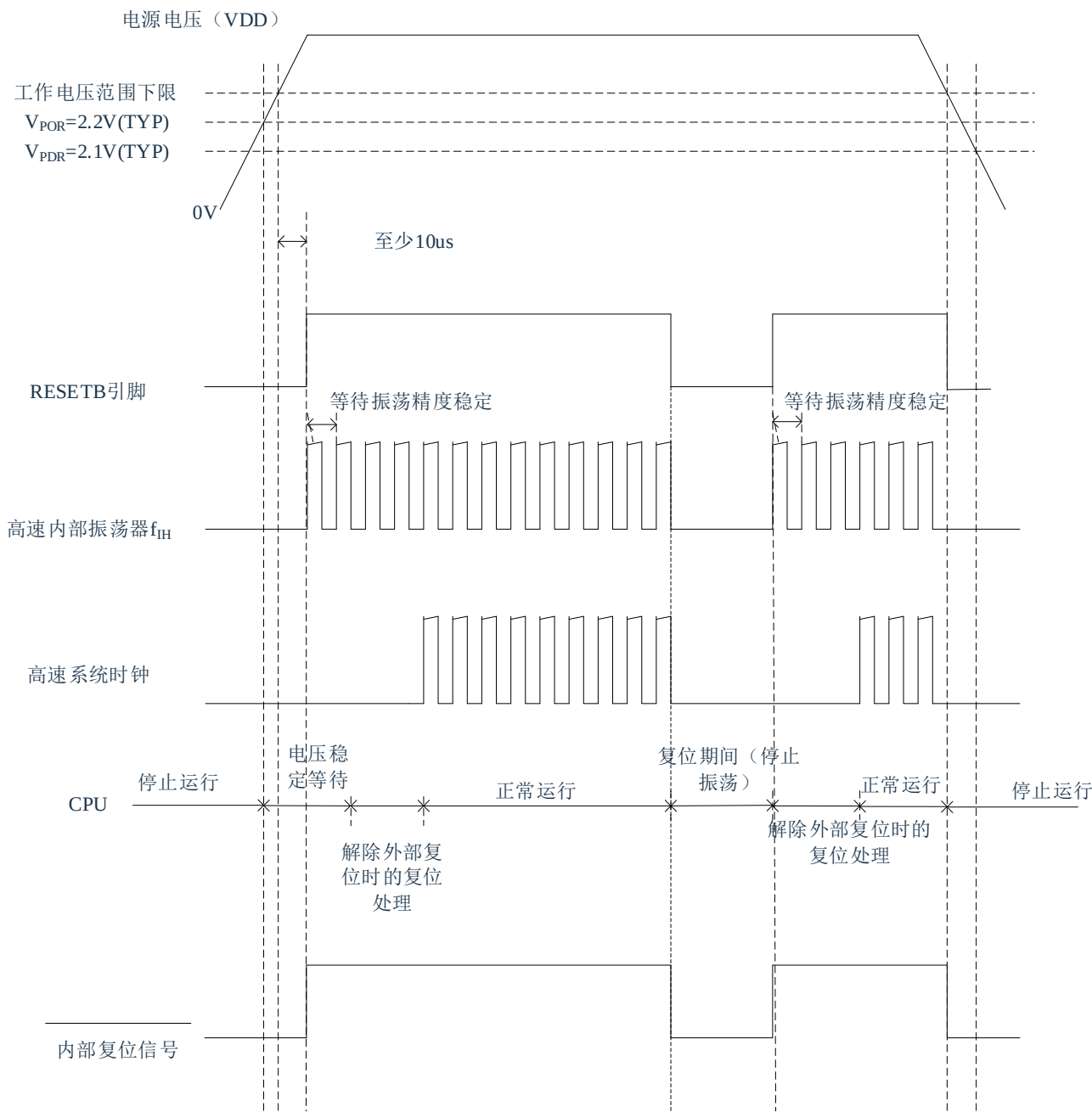


25.3 上电复位电路的运行

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 25-2 上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用 RESETB 引脚的外部复位输入的情况



注1: 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2: 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。

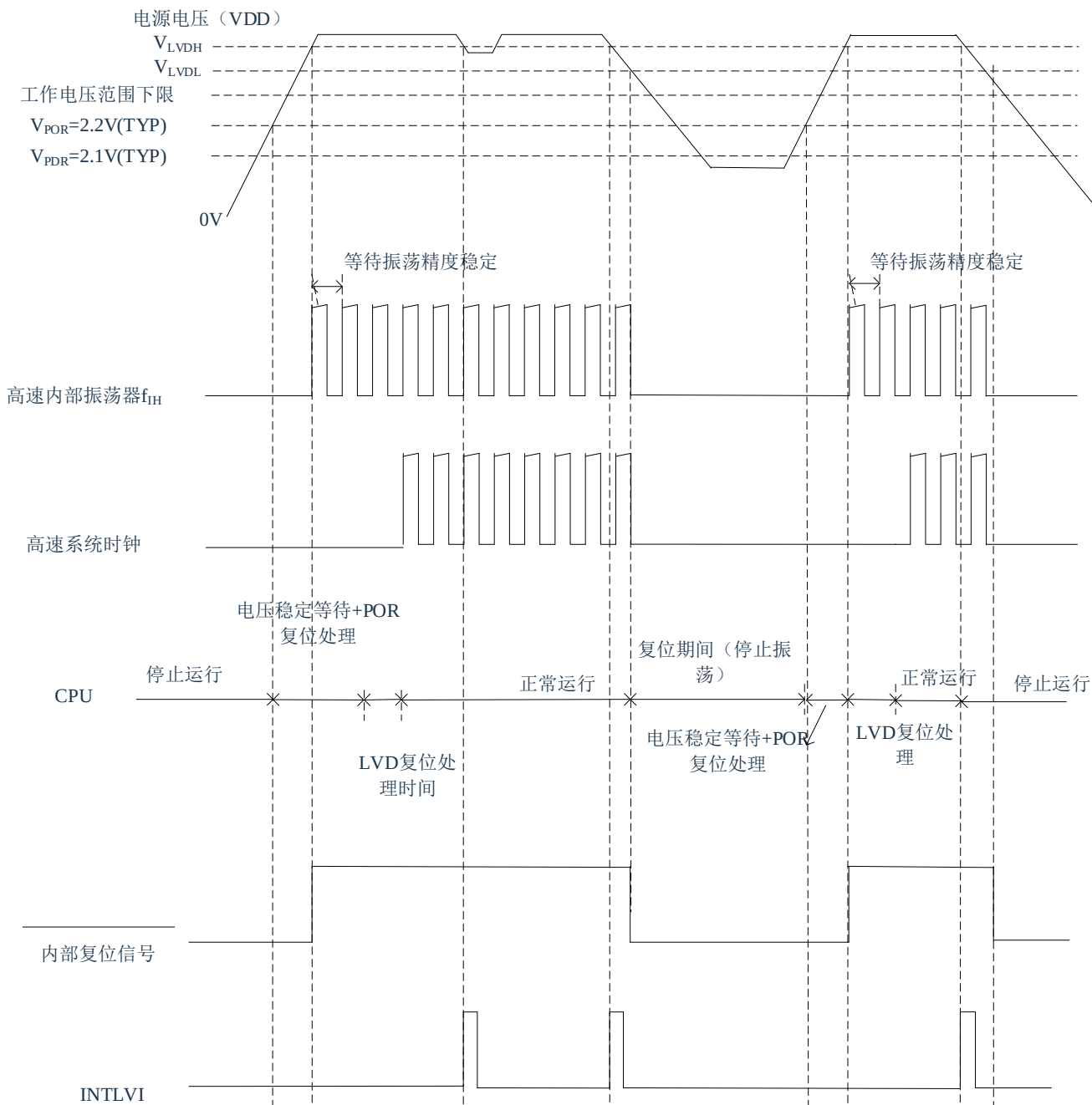
注3: V_{POR} : POR电源电压上升检测电压。

注4: V_{PDR} : POR电源电压下降检测电压。

注5: 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26章 电压检测电路”。

图25-2 上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD为中断&复位模式的情况（选项字节000C1H的LVIMDS1、LVIMDS0=1、0）



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：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26-5 工作电压的确认/复位的设定步骤”和“图26-6 中断&复位模式的初始设定步骤”进行设定。

注3：到开始通常运行为止的时间除了达到 V_{POR} （2.2V(TYP.）后的“电压稳定等待+POR复位处理”以外，在达到LVD检测电平（ V_{LVDH} ）后还需要以下的“LVD复位处理”。

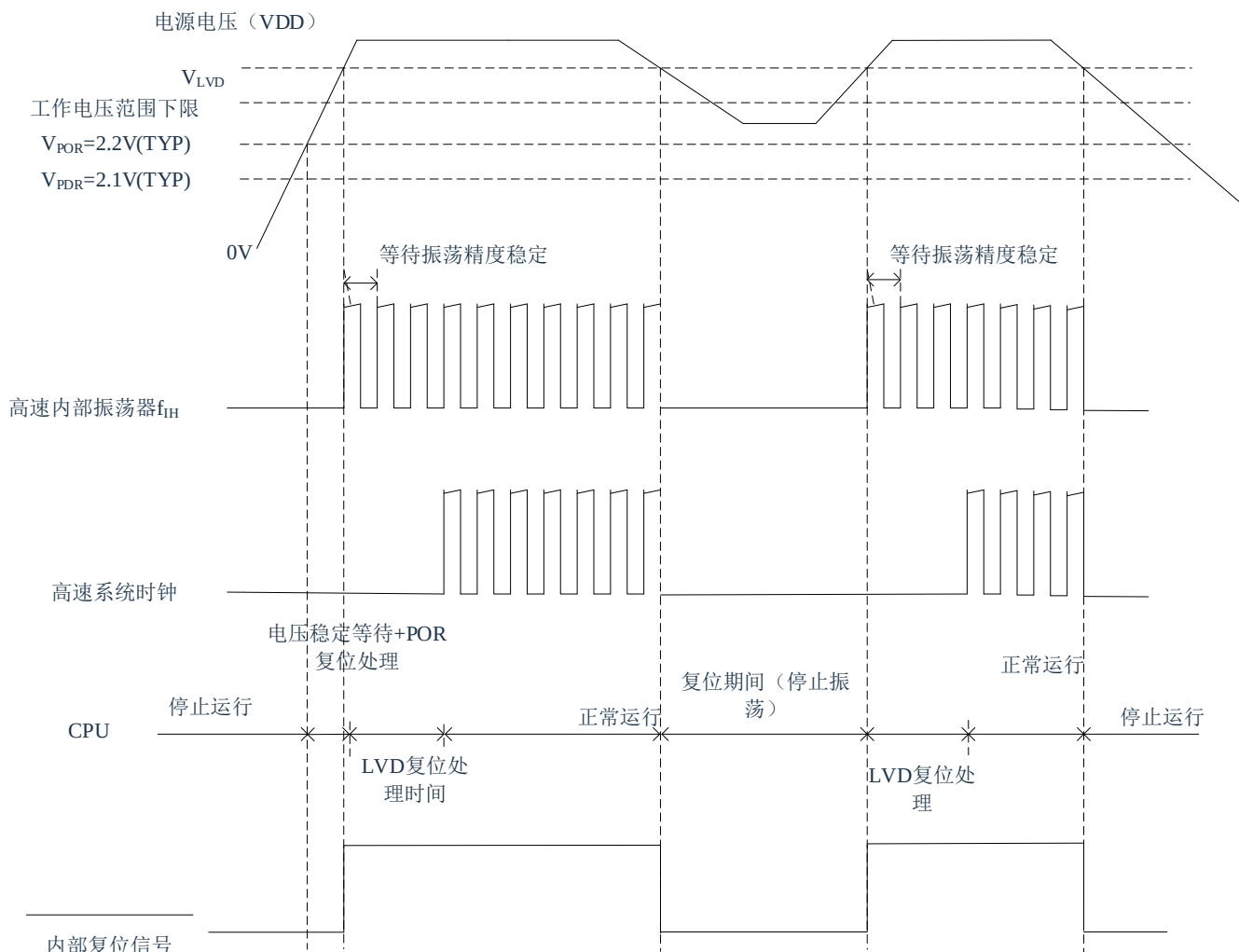
注4： V_{LVDH} 、 V_{LVDL} ：LVD检测电压。

注5： V_{POR} ：POR电源电压上升检测电压。

注6： V_{PDR} ：POR电源电压下降检测电压。

图25-2 上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3)LVD复位模式的情况（选项字节000C1H的LVIMDS1、LVIMDS0=1、1）



注1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注2：到开始通常运行为止的时间除了达到V_{POR}（2.2V(TYP.))后的“电压稳定等待+POR复位处理”以外，在达到LVD检测电平（V_{LVD}）后还需要“LVD复位处理”。

注3：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”。

注4：V_{LVDH}、V_{LVDL}:LVD检测电压。

注5：V_{POR}:POR电源上升检测电压。

注6：V_{PDR}:POR电源下降检测电压。

注7：当选择LVD中断模式（选项字节000C1H的LVIMD1、LVIMD0=0、1）时，从接通电源到开始正常运行的时间和“图25-2(3/3)LVD复位模式的情况”的LVD复位处理的时间相同。

第26章 电压检测电路

26.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设定运行模式和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）。电压检测（LVD）电路有以下功能。

将电源电压（VDD）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 8 种检测电平（参照“第 28 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

(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。

模式	中断&复位模式	复位模式	中断模式
设置	(LVIMDS1、LVIMDS0=1、0)	(LVIMDS1、LVIMDS0=1、1)	(LVIMDS1、LVIMDS0=0、1)
运行过程	在工作电压下降时，当检测到 $VDD < V_{LVDH}$ 时，产生中断请求信号；当检测到 $VDD < V_{LVDL}$ 时，产生内部复位；当检测到 $VDD \geq V_{LVDH}$ 时，解除内部复位。	当检测到 $VDD \geq V_{LVD}$ 时，解除内部复位；当检测到 $VDD < V_{LVD}$ 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 $VDD \geq V_{LVD}$ 为止。当检测到 $VDD \geq V_{LVD}$ 时，解除 LVD 的内部复位。在解除 LVD 的内部复位后，如果检测到 $VDD < V_{LVD}$ 或者 $VDD \geq V_{LVD}$ 时，就产生中断请求信号（INTLVI）。

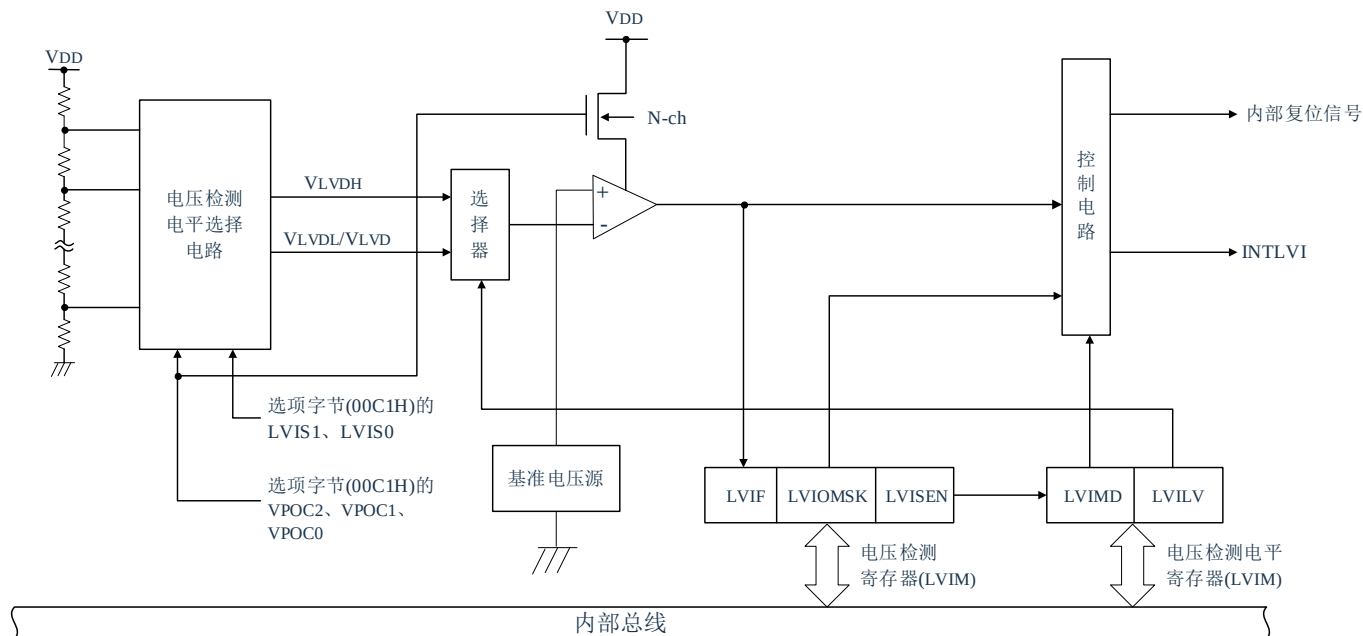
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

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 24 章 复位功能”。

26.2 电压检测电路的结构

电压检测电路的框图如图26-1所示。

图26-1 电压检测电路的框图



26.3 寄存器映射

(电压检测基地址 = 0x4002_0441) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
LVIM	0x000	R/W	电压检测寄存器	0x0
LVIS	0x001	R/W	电压检测电平寄存器	0x0

26.4 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)

26.4.1 电压检测寄存器（LVIM）

此寄存器设定允许或者禁止改写电压检测电平寄存器（LVIS），并且确认LVD输出的屏蔽状态。通过8位存储器操作指令设定LVIM寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	LVISEN ^{注1}	电压检测电平寄存器（LVIS）的允许/禁止改写的设定 0: 禁止 1: 允许	-
6:2	-	保留	-
1	LVIOMSK	LVD 输出的屏蔽状态标志 0: 屏蔽无效 1: 屏蔽有效 ^{注2}	-
0	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：只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能设定，在其他模式中不能更改初始值。

注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，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

26.4.2 电压检测电平寄存器（LVIS）

这是设定电压检测电平的寄存器。

通过 8 位存储器操作指令设定 LVIS 寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”注 1。

位	符号	描述	复位值
7	LVIMD ^{注1}	电压检测的运行模式 0: 中断模式 1: 复位模式	0
6:1	-	保留	-
0	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 _{LVDH} 或者 V _{LVD} ）	0

注1：复位值因复位源和选项字节的设定而变。在发生LVD复位时，不将此寄存器清“00H”。

在发生LVD以外的复位时，此寄存器的值如下：

- 选项字节的LVIMDS1、LVIMDS0=1、0时：00H
- 选项字节的LVIMDS1、LVIMDS0=1、1时：81H
- 选项字节的LVIMDS1、LVIMDS0=0、1时：01H

注2：只有在选择中断&复位模式（选项字节的LVIMDS1位和LVIMDS0位分别为“1”和“0”）时才能写“0”。在其他情况下不能设定。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注3：要改写LVIS寄存器时，必须按照图26-6和图26-7的步骤进行。

注4：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）的详细内容，请参照“第28章 选项字节”。

26.5 电压检测电路的运行

26.5.1 用作复位模式时的设定

通过选项字节 000C1H 设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1）和检测电压（ V_{LVD} ）。如果设定复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。bit7（LVIMD）为“1”（复位模式）。

bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。

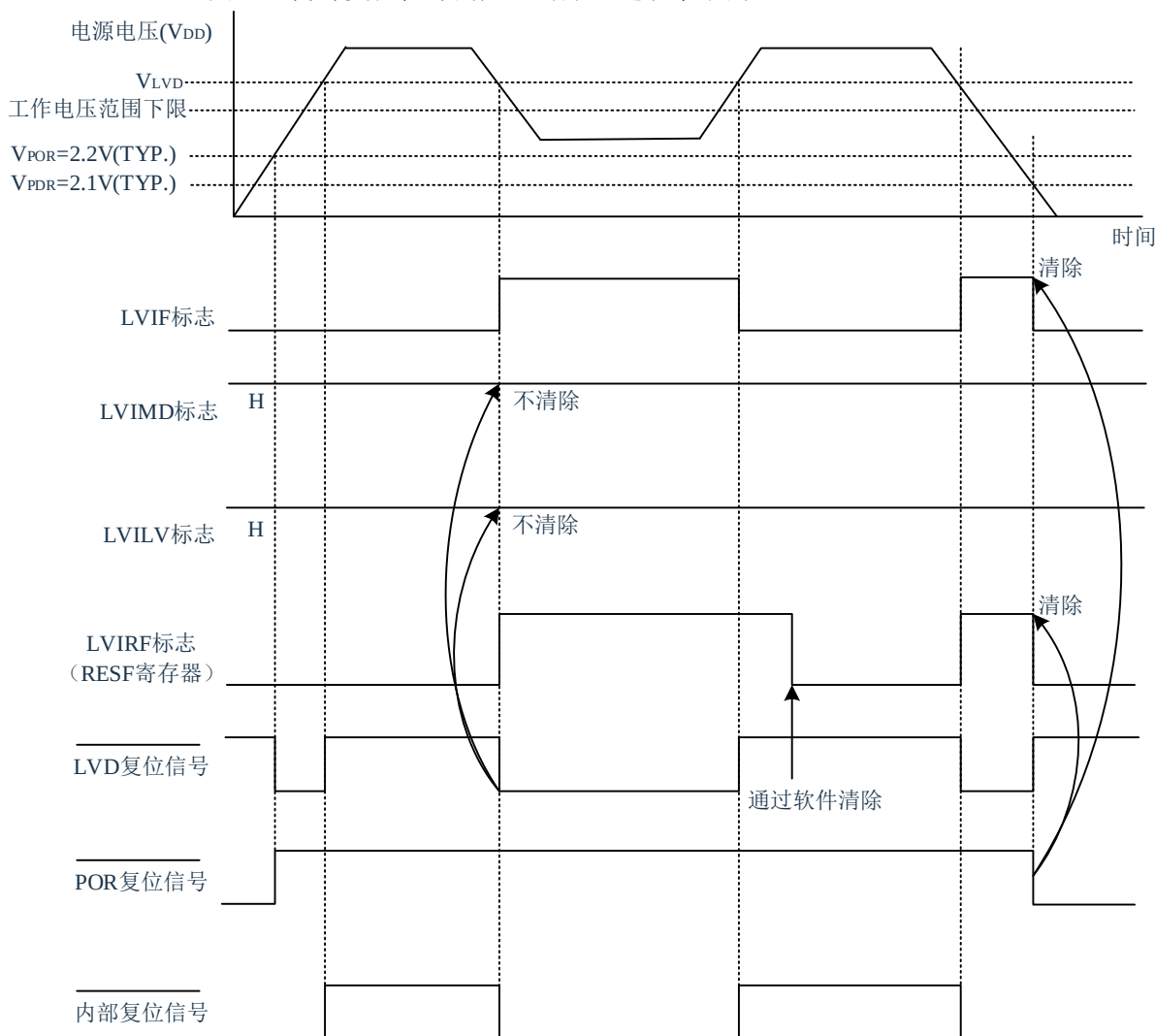
● LVD 复位模式的运行

当接通电源时，复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）在电源电压（VDD）超过电压检测电平（ V_{LVD} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就解除内部复位。

当工作电压下降时，如果电源电压（VDD）低于电压检测电平（ 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 26-2 所示。

图26-2 内部复位信号的产生时序（选项字节的LVIMDS1、LVIMDS0=1、1）



注1: V_{POR} : POR电源电压上升检测电压

注2: V_{PDR} : POR电源电压下降检测电压

26.6 用作中断模式时的设定

通过选项字节 000C1H 设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1）和检测电压（ V_{LVD} ）。如果设定中断模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“01H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。

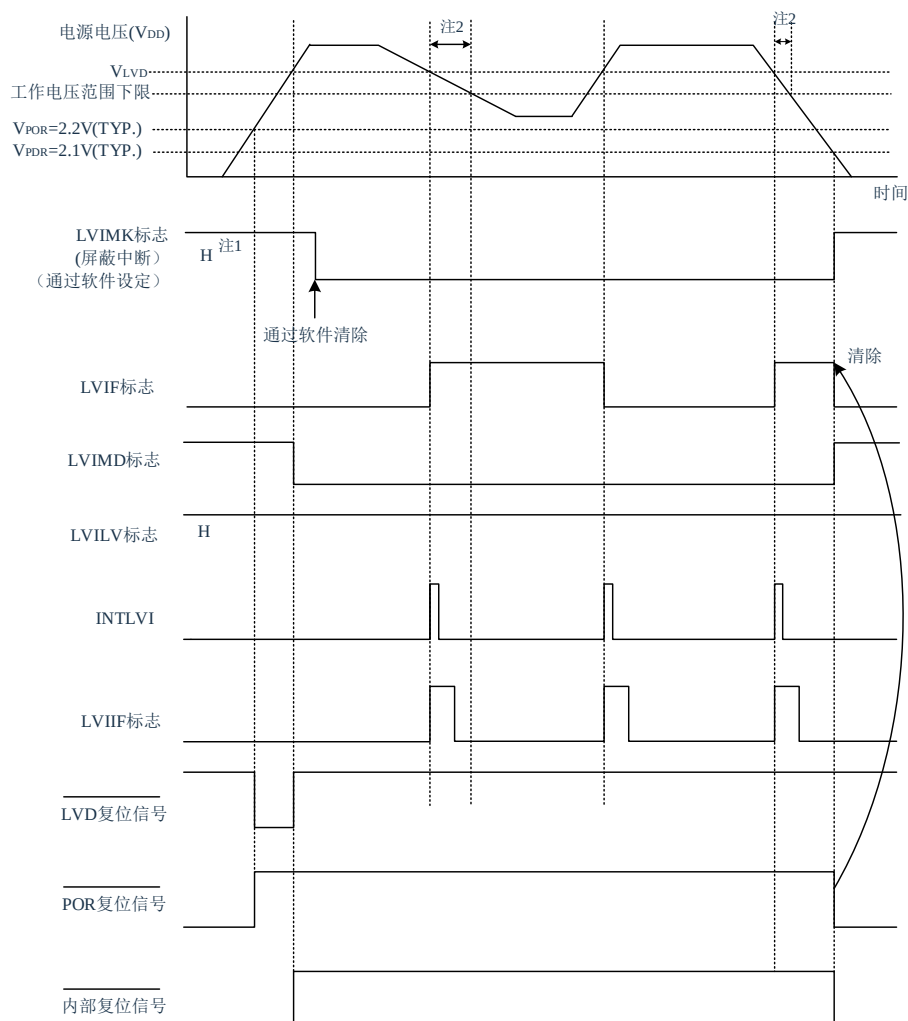
• LVD 中断模式的运行

在产生复位后，中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（VDD）超过电压检测电平（ V_{LVD} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就解除 LVD 的内部复位。

在解除 LVD 的内部复位后，如果电源电压（VDD）超过电压检测电平（ V_{LVD} ），就产生 LVD 的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的产生时序如图 26-3 所示。

图26-3 中断信号的产生时序（选项字节的LVIMDS1、LVIMDS0=0、1）



注1：在产生复位信号后，LVIMK标志变为“1”。

注2：当工作电压下降时，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。

26.7 用作中断&复位模式时的设定

通过选项字节 000C1H 设定运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0））和检测电压（ V_{LVDH} 、 V_{LVDL} ）。

如果设定中断&复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“0”（高电压检测电平： V_{LVDH} ）。

• LVD 中断&复位模式的运行

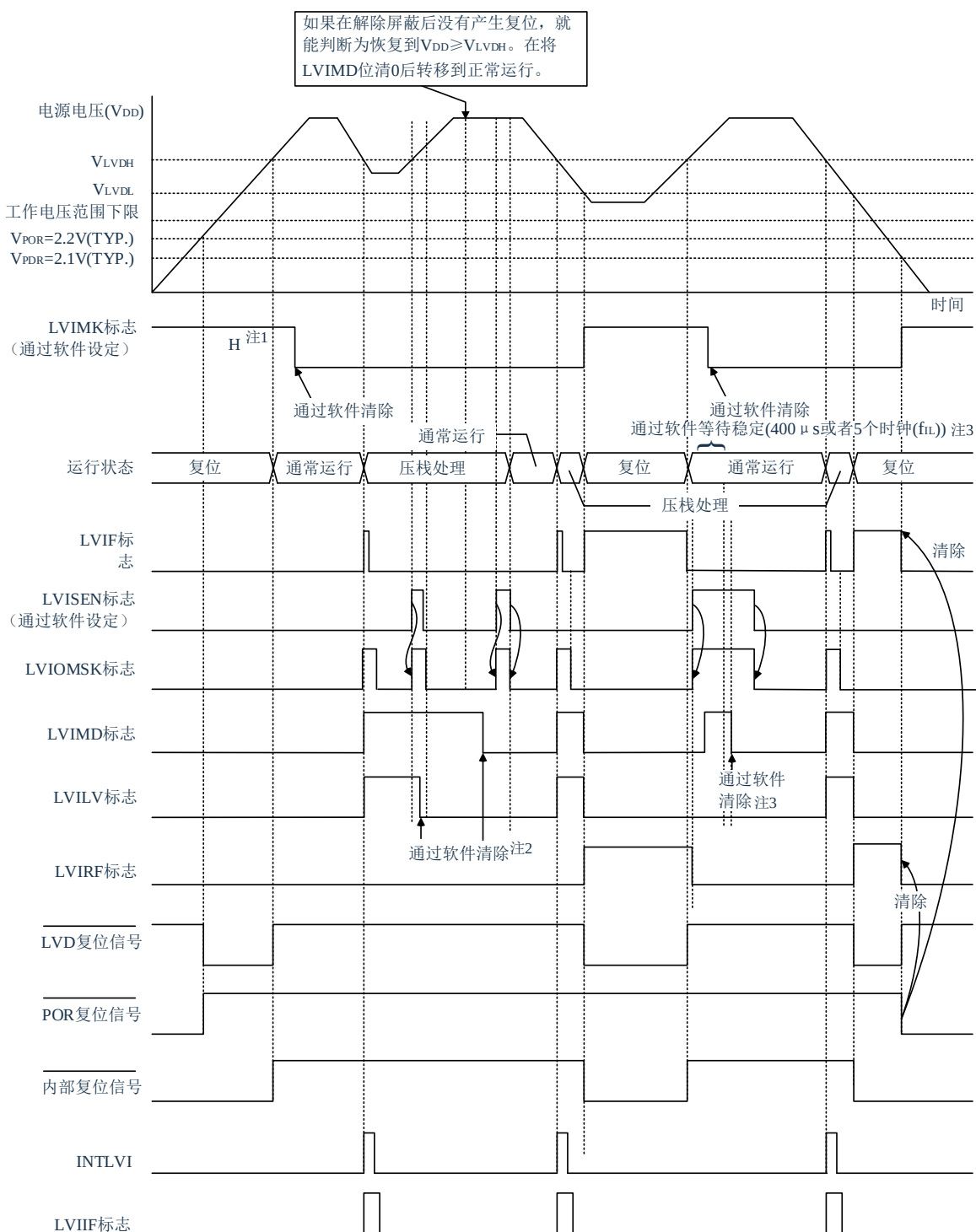
当接通电源时，中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（VDD）超过高电压检测电平（ V_{LVDH} ）前保持 LVD 的内部复位状态。如果电源电压（VDD）超过高电压检测电平（ V_{LVDH} ），就解除内部复位。

当工作电压下降时，如果电源电压（VDD）低于高电压检测电平（ V_{LVDH} ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（VDD）低于低电压检测电平（ V_{LVDL} ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压（VDD）在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高，也不产生中断请求信号。

当使用 LVD 中断&复位模式时，必须按照“图 26-6 工作电压的确认/复位的设定步骤”和“图 26-7 中断&复位模式的初始设定步骤”所示的流程图的步骤进行设定。

LVD 中断&复位模式的内部复位信号和中断信号的产生时序如图 26-4 示。

图 26-4 复位&中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(1/2)



注1：在产生复位信号后，LVIMK标志变为“1”。

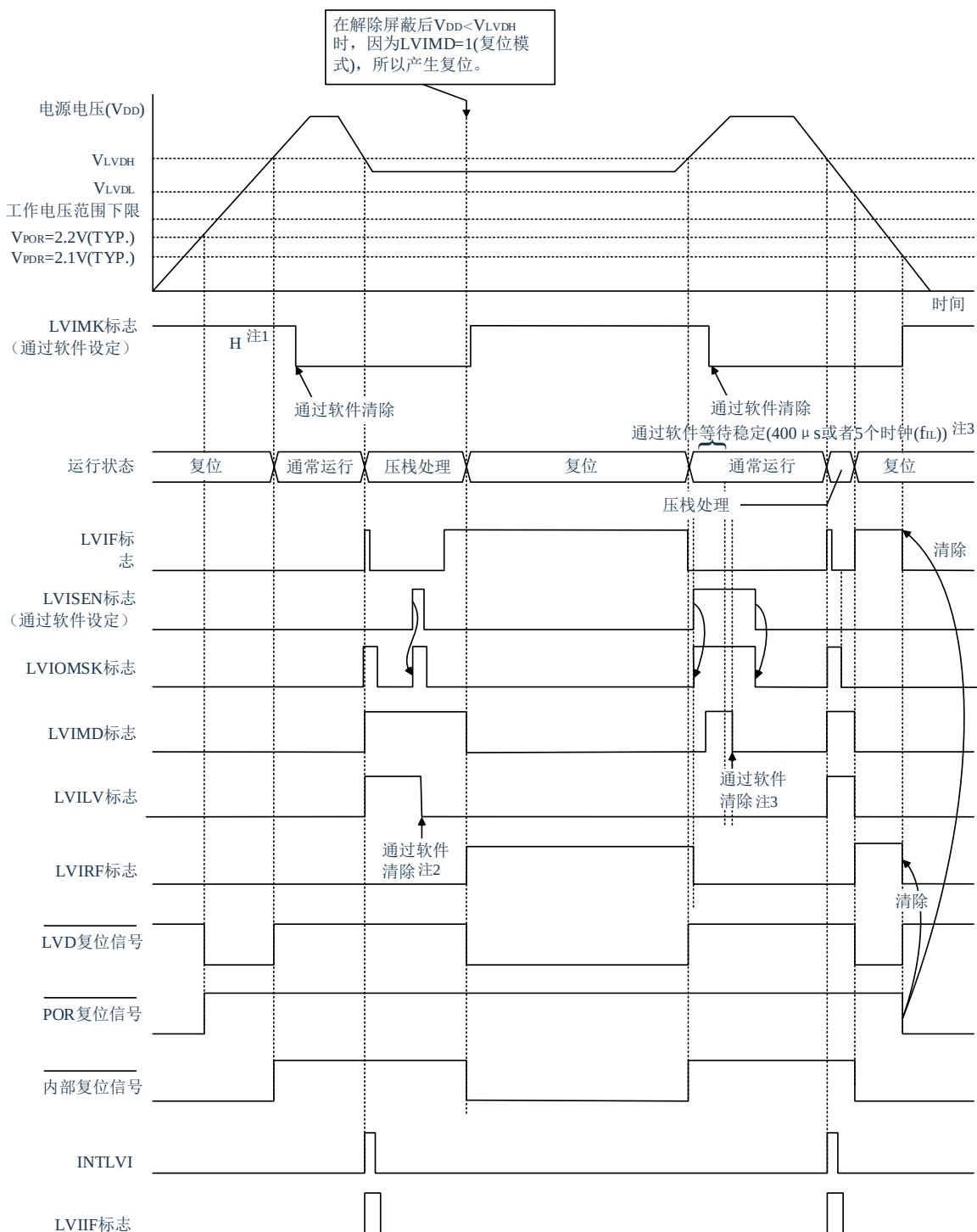
注2：当使用中断&复位模式时，必须在发生中断后按照“图26-6 工作电压的确认/复位的设定步骤”进行设定。

注3：当使用中断&复位模式时，必须在解除复位后按照“图26-7 中断&复位模式的初始设定步骤”进行设定。

注4： 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 26-5 中断&复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(2/2)



注1: 在产生复位信号后, LVIMK标志变为“1”。

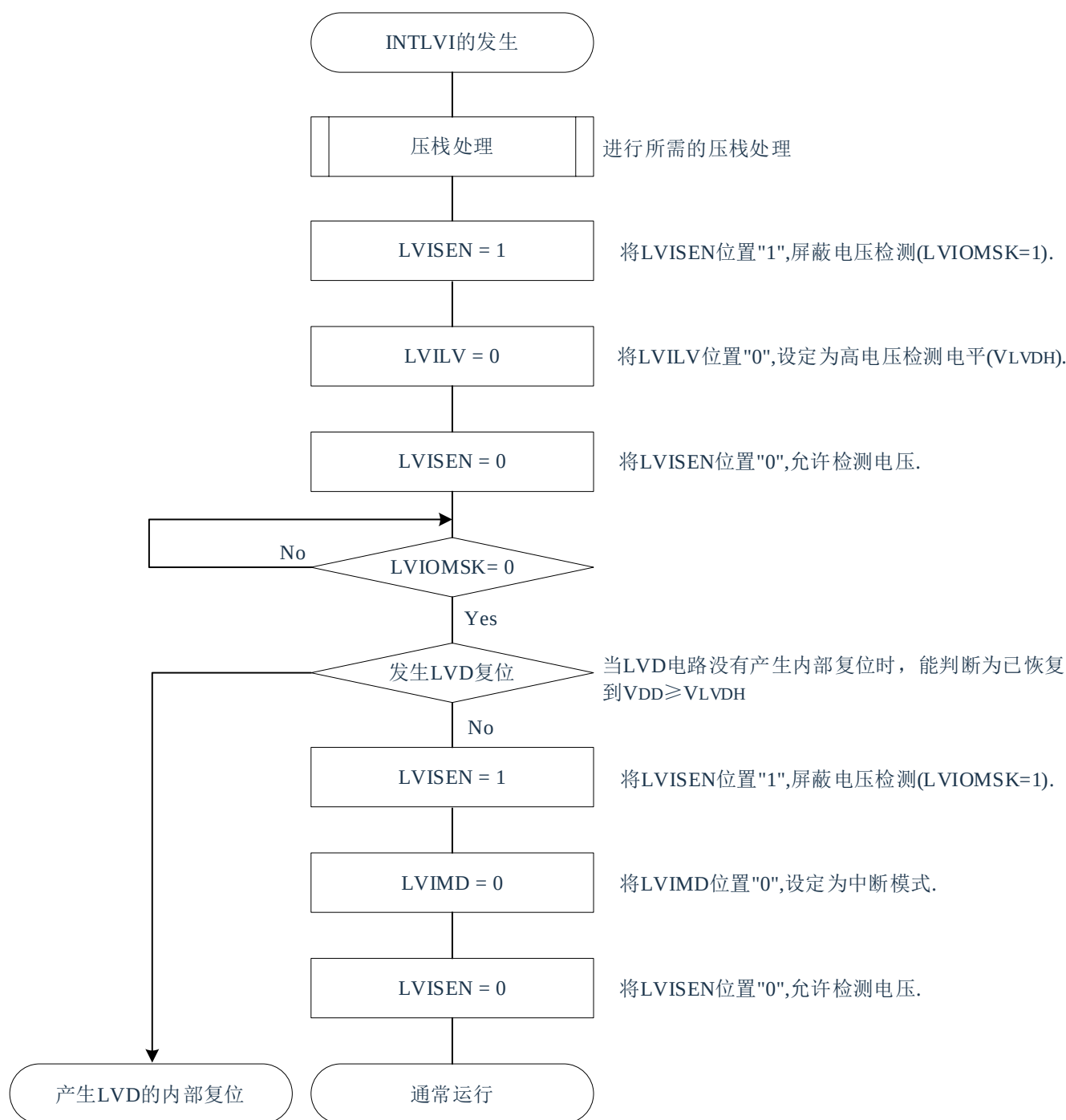
注2: 当使用中断&复位模式时, 必须在发生中断后按照“图26-6 工作电压的确认/复位的设定步骤”进行设定。

注3: 当使用中断&复位模式时, 必须在解除复位后按照“图26-7 中断&复位模式的初始设定步骤”进行设定。

注4: 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。

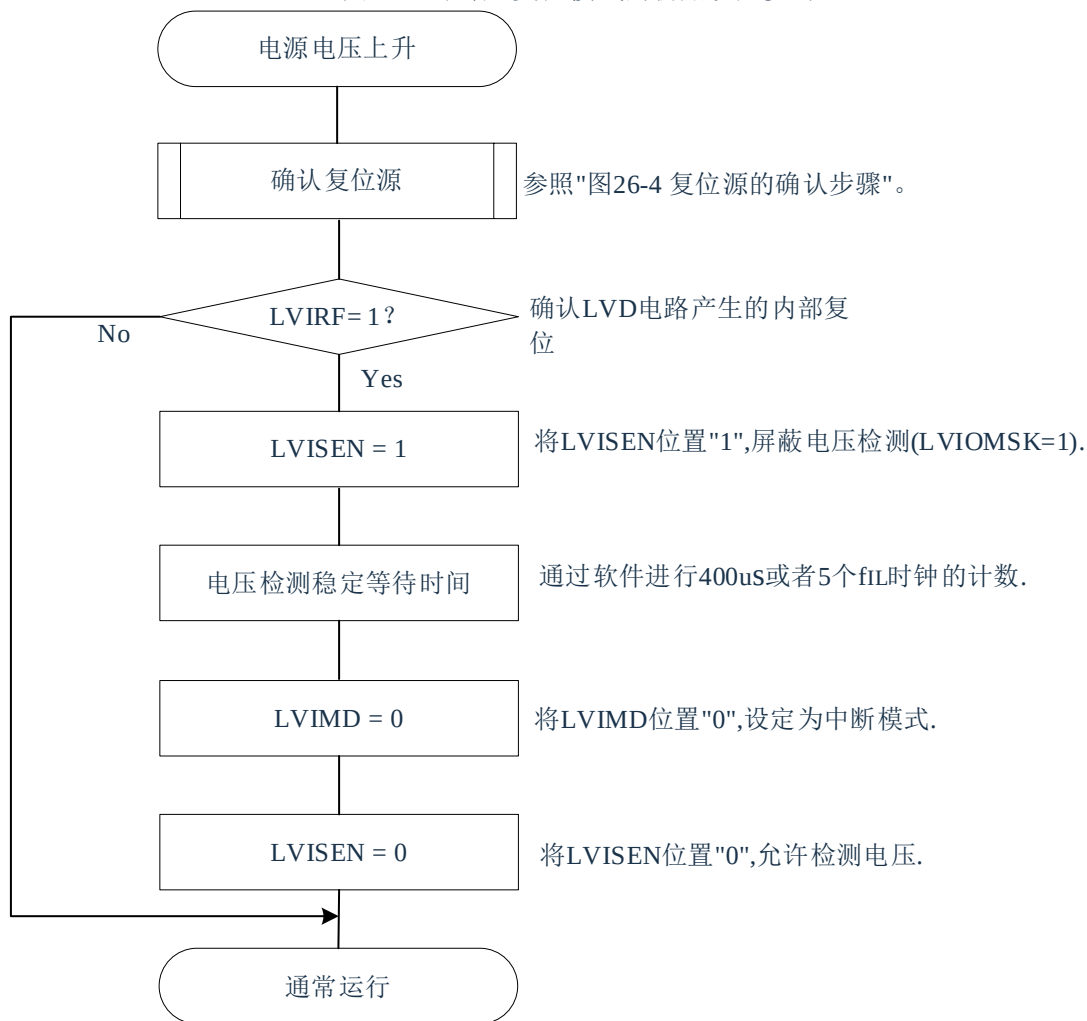
图26-6 工作电压的确认/复位的设定步骤



如果设定中断&复位模式 (LVIMDS1、LVIMDS0=1、0),就在解除 LVD 复位 (LVIRF=1) 后需要 400us 或者 5 个 f_{L} 时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时,必须将 LVISEN 位置“1”,屏蔽 LVD 产生的复位或者中断的产生。

中断&复位模式的初始设定步骤如图 26-7 所示。

图26-7 中断&复位模式的初始设定步骤



备注: f_{IL} : 低速内部振荡器时钟频率

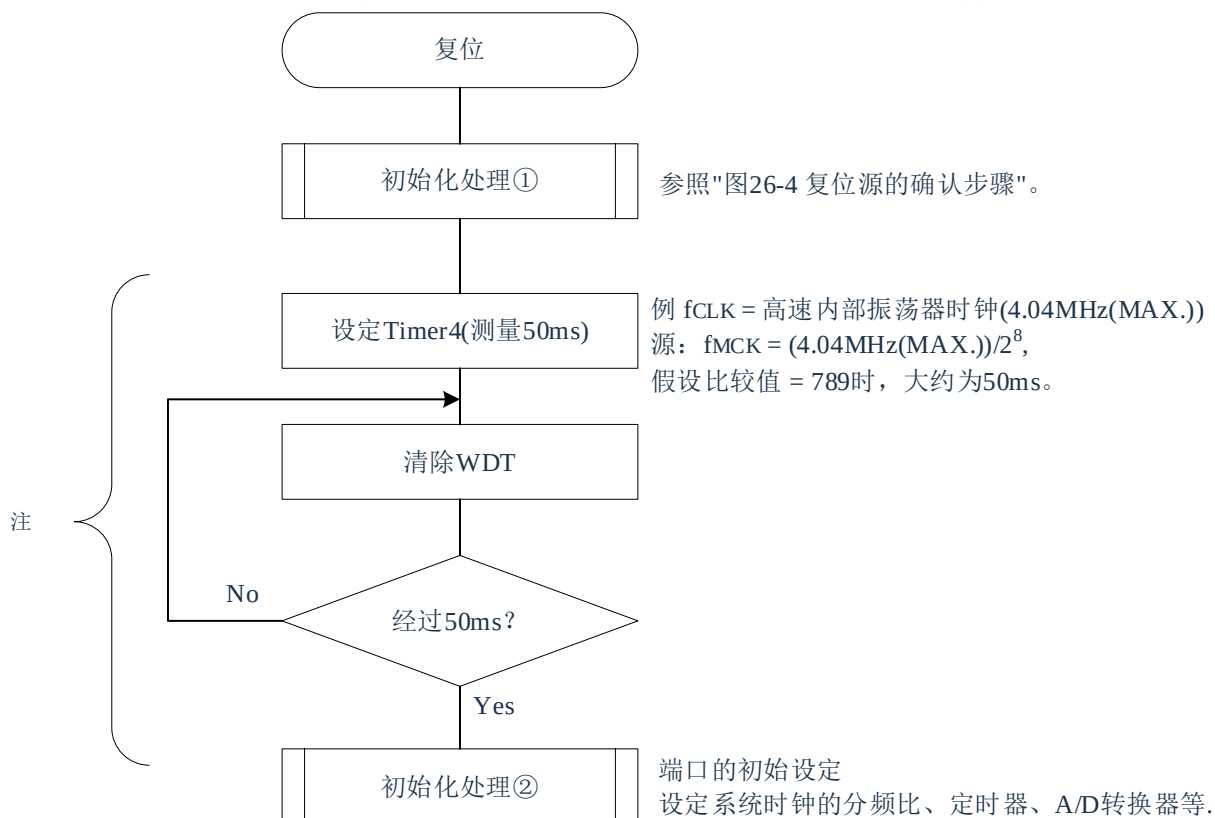
26.8 电压检测电路的注意事项

(1) 有关接通电源时的电压波动

对于电源电压 (V_{DD}) 在 LVD 检测电压附近发生一定时间波动的系统, 有可能重复进入复位状态和复位解除状态。能通过以下的处理, 任意设定解除复位到单片机开始运行的时间。

在解除复位后, 必须通过使用定时器的软件计数器, 在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 26-8 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

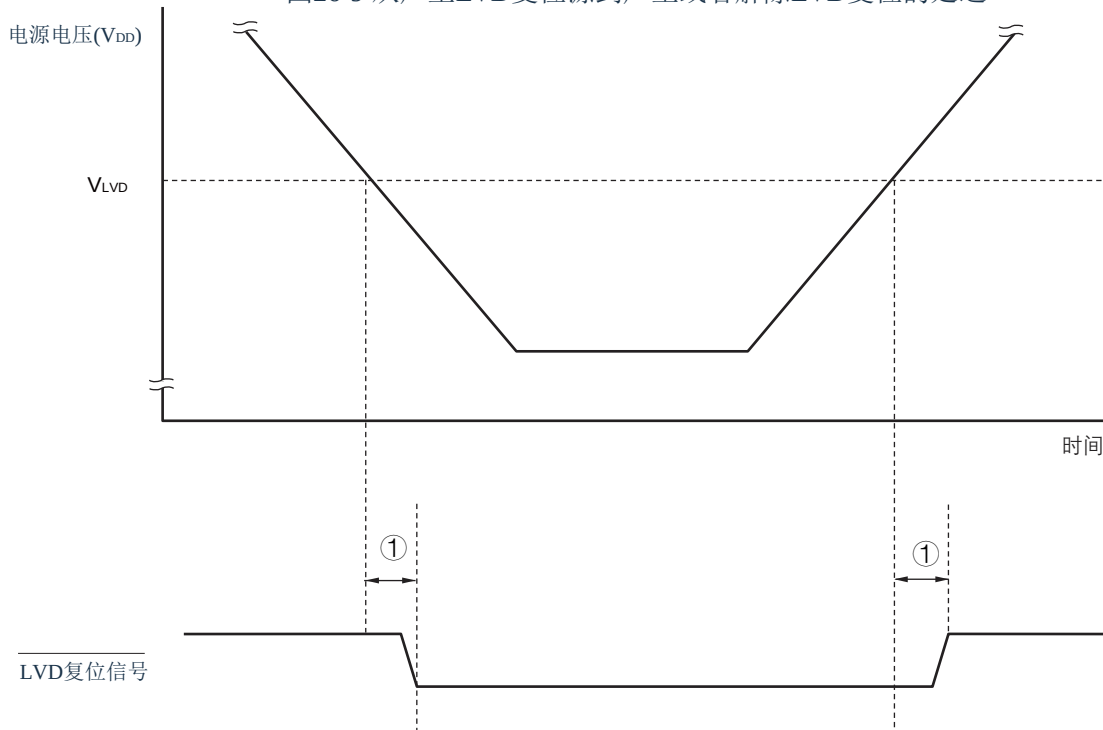


注1: 如果在此期间再次发生复位, 就不转移到初始化处理②。

(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}) $< V_{LVD}$ 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}) $\leq$ 电源电压 (V_{DD}) 到解除 LVD 复位为止也会发生延迟 (参照图 26-9)。

图26-9 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 并且设定为 LVD 中断模式时工作电压下降的情况

在将 LVD 置为 OFF 并且设定为 LVD 中断模式的情况下, 如果工作电压下降, 就必须在的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27章 安全功能

27.1 安全功能的概要

为了对应 IEC60730 安全标准，CMS32F759xx_737xx 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) SFR 保护功能

防止因 CPU 失控而改写 SFR。

(3) 频率检测功能

能使用通用定时器单元进行 CPU/外围硬件时钟频率的自检。

(4) A/D 测试功能

能通过 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、内部基准电压输出的 A/D 转换进行 A/D 转换器的自检。

(5) 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

27.2 寄存器映射

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
- 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)
- CRC输入寄存器（CRCIN） - CRC数据寄存器（CRCD）	CRC运算功能 (通用CRC)
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（TIOS0）	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

(闪存CRC基地址 = 0x4002_1810) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
CRC0CTL	0x000	R/W	闪存CRC控制寄存器	0x0
PGCRCL	0x002	R/W	闪存CRC运算结果寄存器	0x0

(通用CRC基地址 = 0x4004_32FA) RO: 只读, WO: 只写, R/W: 读写

寄存器	物理地址	读/写	描述	复位值
CRCIN	0x4004_33AC	R/W	闪存CRC控制寄存器	0x0
CRCD	0x4004_32FA	R/W	闪存CRC运算结果寄存器	0x0

(SFR基地址 = 0x4004_0478) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
SFRGD	0x000	R/W	SFR保护控制寄存器	0x0

(端口控制基地址 = 0x4004_087B) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
PMS	0x000	R/W	端口模式选择寄存器	0x0

(UID基地址 = 0x0050_1294) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
UID0	0x000	RO	产品唯一的身份标识的第[31:0]位	-
UID1	0X004	RO	产品唯一的身份标识的第[63:32]位	-
UID2	0X008	RO	产品唯一的身份标识的第[95:64]位	-
UID3	0X00C	RO	产品唯一的身份标识的第[127:96]位	-

27.3 安全功能的运行

27.3.1 闪存CRC运算功能（高速CRC）

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设定（初始化）程序中检查整个代码闪存区。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短（例如，256KB 闪存：4096us@64MHz^{注1}）。

CRC 生成多项式对应 CRC-16-CCITT 的“X16+X12+X5+1”。

以 bit31→bit0 的 MSB 优先进行运算。

注1：此为系统时钟，系统时钟与闪存CRC运算的时钟的关系为4:1，系统时钟为64MHZ，则闪存CRC的运算时钟为16MHZ。

注2：因为通用CRC为LSB优先，所以运算结果不同。

注3：在高速CRC校验过程中不能有中断产生。

闪存 CRC 控制寄存器（CRC0CTL）

这是设定高速 CRC 运算器的运行控制和运算范围的寄存器。通过 8 位存储器操作指令设定 CRC0CTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7	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	0
6	CRCCHK248	248K演算范围选择 0: 演算范围由[5:0]位控制 1: 00000H ~ 1EFFBH(248K-4byte)	0
5:0	FEA ^{注4}	高速CRC的演算范围 00000: 00000H ~ 1FFBH(8K-4byte) 00001: 00000H ~ 3FFBH(16K-4byte) 00010: 00000H ~ 5FFBH(24K-4byte) 00011: 00000H ~ 7FFBH(32K-4byte) 00100: 00000H ~ 9FFBH(40K-4byte) 00101: 00000H ~ BFFBH(48K-4byte) 00110: 00000H ~ DFFBH(56K-4byte) 00111: 00000H ~ FFFBH(64K-4byte) 01000: 00000H ~ 11FFBH(72K-4byte) 01001: 00000H ~ 13FFBH(80K-4byte) 01010: 00000H ~ 15FFBH(88K-4byte) 01011: 00000H ~ 17FFBH(96K-4byte) 01100: 00000H ~ 19FFBH(104K-4byte) 01101: 00000H ~ 1BFFBH(112K-4byte) 01110: 00000H ~ 1DFFBH(120K-4byte) 01111: 00000H ~ 1FFFBH(128K-4byte) 10000: 00000H ~ 21FFBH(136K-4byte) 10001: 00000H ~ 23FFBH(144K-4byte) 10010: 00000H ~ 25FFBH(152K-4byte) 10011: 00000H ~ 27FFBH(160K-4byte)	0x0

		10100:	00000H ~ 29FFBH(168K-4byte)	
		10101:	00000H ~ 2BFFBH(176K-4byte)	
		10110:	00000H ~ 2DFFBH(184K-4byte)	
		10111:	00000H ~ 2FFFBH(192K-4byte)	
		11000:	00000H ~ 31FFBH(200K-4byte)	
		11001:	00000H ~ 33FFBH(208K-4byte)	
		11010:	00000H ~ 35FFBH(216K-4byte)	
		11011:	00000H ~ 37FFBH(224K-4byte)	
		11100:	00000H ~ 39FFBH(232K-4byte)	
		11101:	00000H ~ 3BFFBH(240K-4byte)	
		11110:	00000H ~ 3DFFBH(248K-4byte)	
		11111:	00000H ~ 3FFFBH(256K-4byte)	

注4: bit5必须设置为0。

注5: 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。

27.3.1.1 闪存CRC运算结果寄存器L (PGCRCL)

这是保存高速 CRC 运算结果的寄存器。

通过 16 位存储器操作指令设定 PGCRCL 寄存器。

在产生复位信号后, 此寄存器的值变为“0000H”。

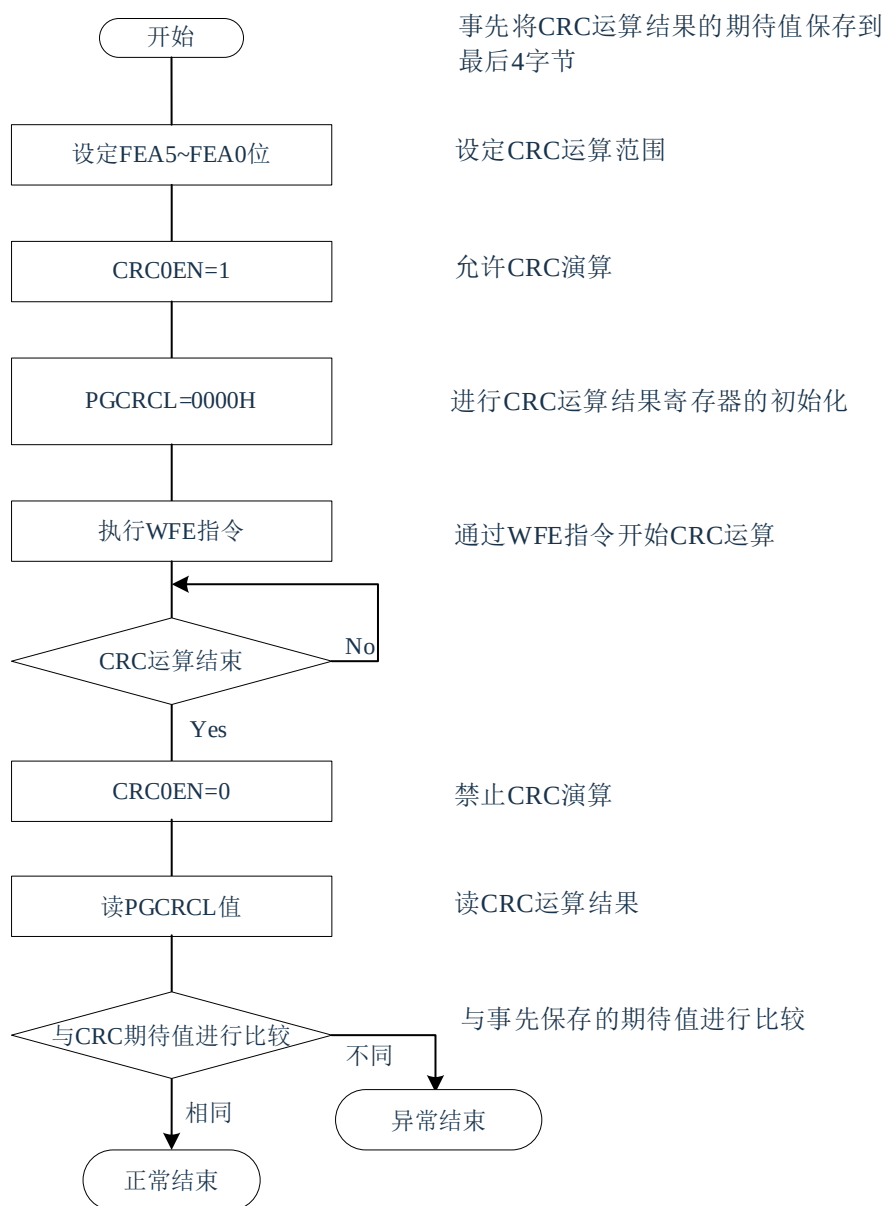
位	符号	描述	复位值
15:0	PGCRCL	保存高速 CRC 运算结果 0000H ~ FFFFH	0x0

注: 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27-1所示。

<操作流程>

图27-1：闪存CRC运算功能（高速CRC）的流程图



注意：

1. 只以代码闪存为CRC运算的对象。
2. 必须将CRC运算的期待值保存在代码闪存中的运算范围后的区域。

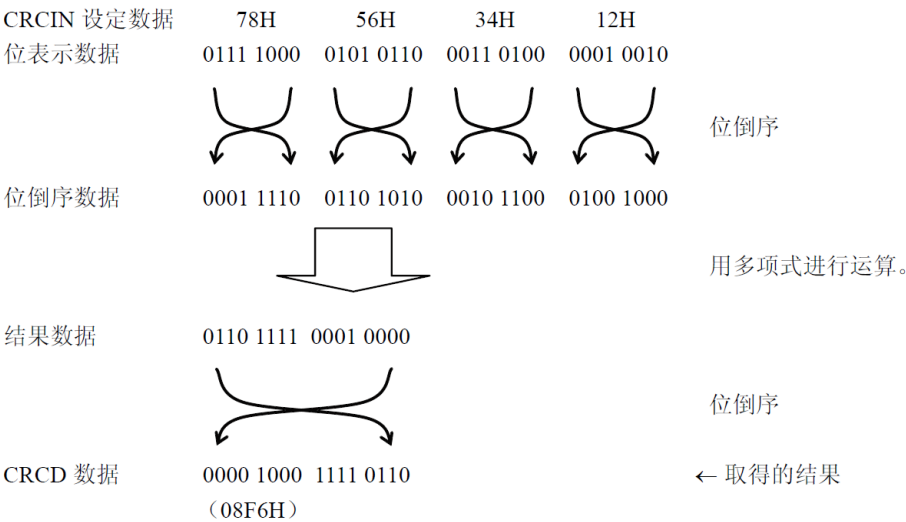
27.3.2 CRC运算功能（通用CRC）

为了必须保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。

此通用 CRC 能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定要确认的数据。

在主系统时钟运行模式运行模式中，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：在执行程序的过程中，因为调式程序将软件断点的设定行改写为断点指令，所以如果在 CRC 运算的对象区设定软件断点，CRC 的运算结果就不同。

27.3.2.1 CRC输入寄存器（CRCIN）

这是设定通用CRC的CRC计算数据的8位寄存器。能设定的范围为“00H~FFH”。

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

位	符号	描述	复位值
7:0	CRCIN	通用 CRC 数据输入 00H ~ FFH	0x0

27.3.2.2 CRC数据寄存器（CRCD）

这是保存通用 CRC 运算结果的寄存器。能设定的范围为“0000H~FFFFH”。

在写 CRCIN 寄存器后经过 1 个 CPU/外围硬件时钟（ f_{CLK} ），将 CRC 运算结果保存到 CRCD 寄存器。通过 16 位存储器操作指令设定 CRCD 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

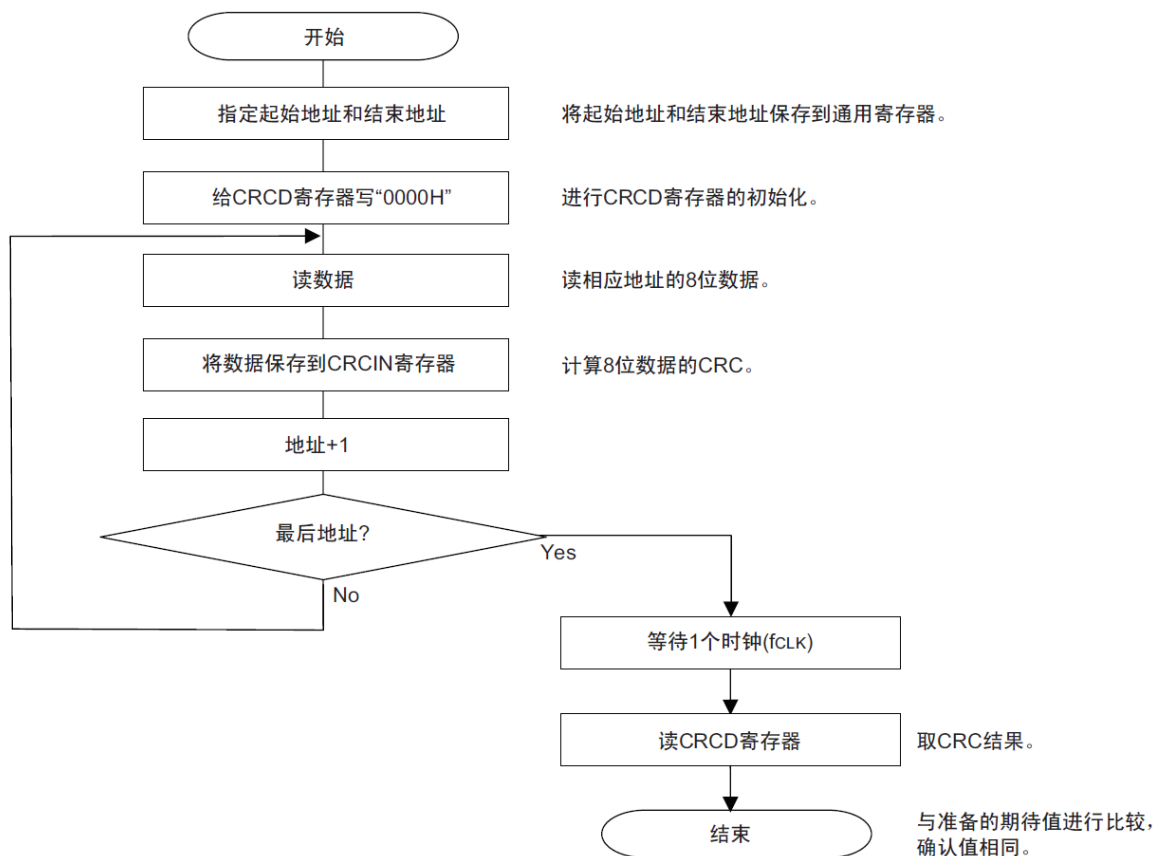
位	符号	描述	复位值
15:0	CRCD	保存通用 CRC 运算结果(0000H ~ FFFFH)	0x0

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27-2 CRC运算功能（通用CRC）的流程图



27.3.3 SFR保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护重要的 SFR，使其免遭改写。SFR 保护功能用于保护端口功能、时钟控制功能和电压检测电路的控制寄存器的数据。

如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

27.3.3.1 SFR保护控制寄存器（SFRGD）

此寄存器控制 SFR 保护功能是否有效。

SFR 保护功能使用 GPORT 位和 GCSC 位。

通过 8 位存储器操作指令设定 SFRGD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:3	保留	-	-
2	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、PS_xx ^注 。	0
1	-	保留	-
0	GCSC	时钟控制功能、电压检测电路的控制寄存器的保护 0: 无效，能读写时钟控制功能和电压检测电路的控制寄存器。 1: 有效。时钟控制功能和电压检测电路的控制寄存器的写操作无效，能读。 [被保护的 SFR]CMC、CSC、OSTS、CKC、PERx、 OSMC、LVIM、LVIS。	0

注：不保护Pxx（端口寄存器）。

27.3.4 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

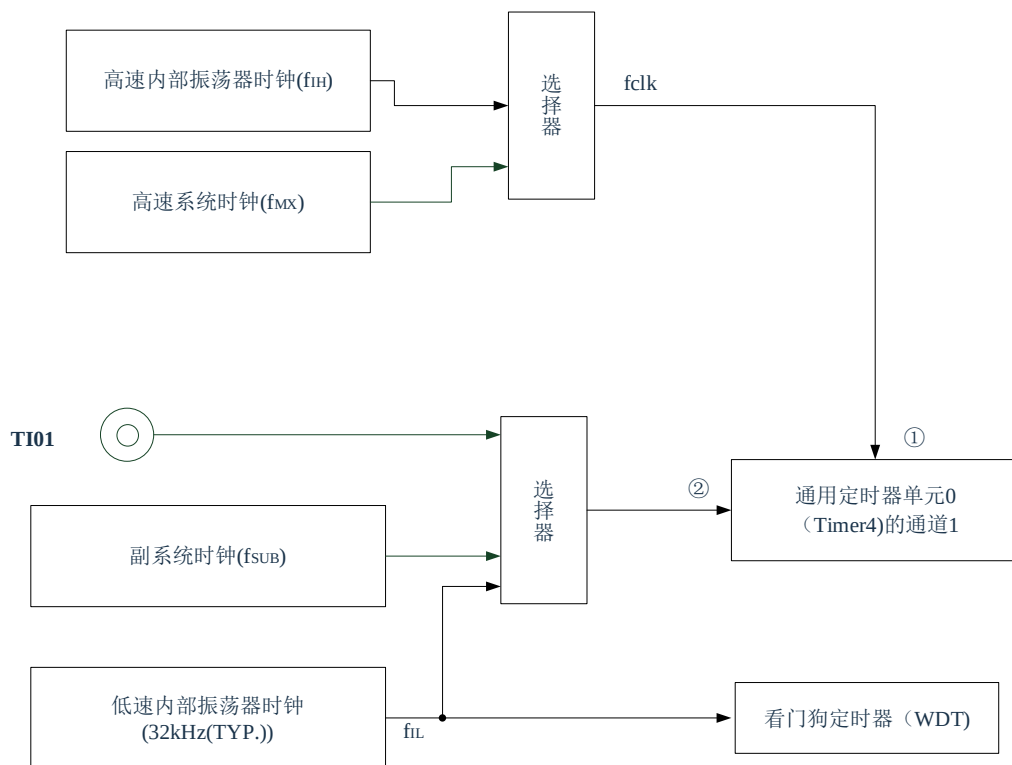
频率检测功能可使用 CPU/外围硬件的时钟频率 (f_{clk})，并且能通过测量 Timer4 通道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟停止振荡，就不能判断 2 个时钟的比率关系。

<要比较的时钟>

- ① CPU/外围硬件的时钟频率 (f_{clk}):
 - 高速内部振荡器时钟 (f_{IH})
- ② Timer4 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} : 32kHz (TYP.))
 - 副系统时钟 (f_{SUB}) 注

图27-3 频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。有关输入脉冲间隔的测量方法，请参考“6.7.4 作为输入脉冲间隔测量的运行”。

注：只有内置副系统时钟的产品才能选择。

27.3.4.1 定时器输入选择寄存器0 (TIOS0)

寄存器说明请参考6.2.11节。

27.3.5 A/D测试功能

IEC60730标准要求对A/D转换器进行测试。此A/D测试功能通过对A/D转换器的正(+)基准电压、负(-)基准电压、模拟输入通道(AN)、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AN_{Ix}引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ② 对AN_x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AN_{Ix}引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ⑥ 对AN_x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AN_{Ix}引脚作为A/D转换对象(ADTES2、ADTES1、ADTES0=0、0、0)。
- ⑩ 对AN_x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。

27.3.5.1 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(AN_{Ixx})、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,进行以下的设定:

- 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。

27.3.5.2 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AN_{Ixx}或者内部基准电压(1.45V)时,必须将A/D测试寄存器(ADTES)置“00H”。

A/D寄存器说明请参考16.2。

27.3.6 输入/输出引脚的数字输出信号电平检测功能

IEC60730 标准要求确认 I/O 功能是否正常。

输入/输出引脚的数字输出信号电平检测功能在引脚为输出模式时，能读引脚的数字输出电平。

27.3.6.1 端口模式选择寄存器（PMS）

此寄存器选择在引脚为输出模式（端口模式寄存器（PMm）的 PMmn 位为“0”）时是读端口的输出锁存器的值还是读引脚的输出电平。

通过 8 位存储器操作指令设定 PMS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

位	符号	描述	复位值
7:1	-	保留	-
0	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	0

注：m=A~D，n=0~15。

27.3.7 产品唯一身份标识寄存器

产品唯一的身份标识非常适合：

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。

产品唯一身份标识寄存器 0 (UID0)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[31:0]位，其值在出厂时编写	-

产品唯一身份标识寄存器 1 (UID1)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[63:32]位，其值在出厂时编写	-

产品唯一身份标识寄存器 2 (UID2)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[95:64]位，其值在出厂时编写	-

产品唯一身份标识寄存器 3 (UID3)

位	符号	描述	复位值
31:0	-	产品唯一身份标识的第[127:96]位，其值在出厂时编写	-

第28章 选项字节

28.1 选项字节的功能

芯片的闪存 000C0H~000C3H, 500004H 为选项字节区。

选项字节由用户选项字节（000C0H~000C2H）和闪存数据保护选项字节（000C3H, 500004H）构成。在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

注意：是否使用各功能无关，必须设定选项字节。

28.1.1 用户选项字节（000C0H~000C2H）

(1) 000C0H

独立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

独立看门狗定时器的上溢时间的设定

- 独立看门狗定时器的窗口打开期间的设定

独立看门狗定时器的间隔中断的设定

- 使用或者不使用间隔中断

(2) 000C1H

LVD运行模式的设定

- 中断&复位模式
- 复位模式
- 中断模式
- LVD为OFF（使用RESETB引脚的外部复位输入）

LVD检测电平（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）的设定

注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。

注2：工作电压范围取决于用户选项字节（000C2H）的设定。

(3) 000C2H

高速内部振荡器的频率设定

- 从 2MHz~64MHz 中选择

28.2 闪存数据保护选项字节（000C3H，500004H）

片上调试时闪存数据保护的选项控制

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

28.3 寄存器映射

寄存器	地址	读/写	描述	复位值
选项字节0	0x00C0H	R/W	独立看门狗定时器状态控制寄存器	0xFF
选项字节1	0x00C1H	R/W	LVD状态控制寄存器	0xFF
选项字节2	0x00C2H	R/W	高速内部振荡器频率控制寄存器	0xEE
选项字节3	0x00C3H	R/W	片上调试闪存数据保护控制寄存器1	0xFF
选项字节4	0x500004H	R/W	片上调试闪存数据保护控制寄存器2	0xFF

28.4 用户选项字节

28.4.1 用户选项字节0 (000C0H)

位	符号	描述	复位值
7	WDTINT	独立看门狗定时器的间隔中断的使用/不使用 0: 不使用间隔中断 1: 当达到上溢时间的75%+1/2f _{IL} 时, 产生间隔中断	1
6:5	WINDOW[1:0]	独立看门狗定时器的窗口打开期间 0X: 禁止设定。 10: 75% 11: 100%	0x3
4	WDTON	独立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)	1
3:1	WDTCS[2:0]	独立看门狗定时器的上溢时间 (f _{IL} =32kHz) 000: 2 ⁶ /f _{IL} (2.0ms) 001: 2 ⁷ /f _{IL} (4.0ms) 010: 2 ⁸ /f _{IL} (8.0ms) 011: 2 ⁹ /f _{IL} (16.0ms) 100: 2 ¹¹ /f _{IL} (64.0ms) 101: 2 ¹³ /f _{IL} (256.0ms) 110: 2 ¹⁴ /f _{IL} (512.0ms) 111: 2 ¹⁶ /f _{IL} (2048.0ms)	0x7
0	WDSTBYON	独立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1} 1: 在睡眠模式中, 允许计数器的运行	1

注1: 当WDSTBYON位为“0”时, 与WINDOW1位和WINDOW0位的值无关, 窗口打开期间为100%。

注2: f_{IL}: 低速内部振荡器的时钟频率。

28.4.2 用户选项字节1（000C1H）

位	符号	描述	复位值
7:5	VPOC[2:0]	检测电压设定	0x7
4	-	保留（须为1）	1
3:2	LVIS[1:0]	检测电压设定	0x2
1:0	LVIMDS[1:0]	模式选择 10: 中断&复位模式 11: 复位模式 01: 中断模式	0x3

• LVD检测电压设定（中断&复位模式）

检测电压			选项字节的设定值						
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
2.75V	2.70V	2.5V	0	-	1	1	0	1	0
3.05V	3.00V					0	1		
3.35V	3.30V					0	0		
3.75V	3.70V	3.0V		-	0	1	0		
4.05V	4.00V					0	1		
4.35V	4.30V					0	0		
—			禁止设定上述以外的值。						

- 注1：必须给bit4写“1”。
- 注2：有关LVD电路的详细内容，请参照“第26章 电压检测电路”。
- 注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。

• LVD检测电压设定（复位模式）

检测电压		选项字节的设定值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
2.55V	2.50V	0	-	1	1	1	1	1
2.75V	2.70V		-	1	1	0		
3.05V	3.00V		-	1	0	1		
3.35V	3.30V		-	1	0	0		
3.05V	3.00V		-	0	1	1		
3.75V	3.70V		-	0	1	0		
4.05V	4.00V		-	0	0	1		
4.35V	4.30V		-	0	0	0		
—			禁止设定上述以外的值。					

注1：必须给bit4写“1”。

注2：有关LVD电路的详细内容，请参照“第26章 电压检测电路”。

注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。

• LVD检测电压设定（中断模式）

检测电压		选项字节的设定值						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
2.55V	2.50V	0	-	1	1	1	0	1
2.75V	2.70V		-	1	1	0		
3.05V	3.00V		-	1	0	1		
3.35V	3.30V		-	1	0	0		
3.05V	3.00V		-	0	1	1		
3.75V	3.70V		-	0	1	0		
4.05V	4.00V		-	0	0	1		
4.35V	4.30V		-	0	0	0		
—		禁止设定上述以外的值。						

注1：必须给bit4写“1”。

注2：有关LVD电路的详细内容，请参照“第26章 电压检测电路”。

注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。.

• LVD为OFF时的设定（使用RESETB引脚的外部复位输入）

检测电压		选项字节的设定值						
V_{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

注意1：必须给bit4写“1”。

注意2：当电源电压上升时，必须在电源电压达到数据手册的AC特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H）的设定。

注1：×：忽略。

注2：有关LVD电路的详细内容，请参照“第26章 电压检测电路”。

注3：检测电压是TYP.值。详细内容请参照数据手册的LVD电路特性。

28.4.3 用户选项字节2 (000C2H)

位	符号	描述	复位值
7:5	-	保留（须为1）	0x7
4	-	保留（须为0）	0
3:0	FRQSE[3:0]	高速内部震荡器时钟频率选择	0x0E

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率	
				f_{HOCO}	f_{IH}
1	0	0	0	64MHz	64MHz
1	0	0	1	64MHz	32MHz
1	0	1	0	64MHz	16MHz
1	0	1	1	64MHz	8MHz
1	1	0	0	64MHz	4MHz
1	1	0	1	64MHz	2MHz
上述以外				禁止设定。	

注1：必须给bit7~5写“1”，bit4必须写“0”。

注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。

28.4.4 闪存数据保护选项字节1（000C3H）

位	符号	描述	复位值
7:0	OCDEN[7:0]	闪存数据保护的控制	0xFF

28.4.5 闪存数据保护选项字节2（500004H）

位	符号	描述	复位值
7:0	OCDM[7:0]	闪存数据保护的控制	0xFF

OCDM	OCDEN	闪存数据保护的控制
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：50_0004H地址属于数据闪存区，如果使用该地址做数据存储用，需先确定数值不会引起保护选项的误设。

第29章 FLASH控制

29.1 FLASH控制功能描述

本制品包含一颗 256KByte 容量的 FLASH 存储器，共划分为 512 个 Sector，每个 Sector 容量为 512Byte。可做为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。

29.2 FLASH存储器结构

FFFF_FFFFH	保留
E00F_FFFFH	Cortex-M0+专用外设资源区
E000_0000H	保留
4008_FFFFH	外设资源区
4000_0000H	保留
2000_3FFFH	SRAM1(8KB)
2000_2000H 2000_1FFFH	SRAM0(8KB)
2000_0000H	保留
0050_0FFFH	数据闪存3.5KB
0050_0200H	保留
0003_FFFFH	主闪存区（最大256KB）
0000_0000H	

29.3 寄存器映射

(FLASH控制基地址 = 0x4002_0000) RO: 只读, WO: 只写, R/W: 读写

寄存器	偏移量	读/写	描述	复位值
FLACR	0x000	R/W	FLASH访问寄存器	0x0
FLKEY	0x004	WO	FLASH密钥寄存器	0x0
FLSR	0x00C	R/W	FLASH状态寄存器	0x0
FLPRT	0x010	R/W	FLASH保护寄存器	0x0
FLCMD	0x014	R/W	FLASH操作控制寄存器	0x0
FLADDR	0x018	R/W	FLASH地址寄存器	0x00ffffff
FLDATA0	0x01C	R/W	FLASH数据寄存器0	0xffffffff
FLDATA1	0x020	R/W	FLASH数据寄存器1	0xffffffff
FLCR	0x024	R/W	FLASH控制寄存器	0x80000300
FLREGION	0x048	R/W	FLASH区域控制寄存器	0x0

29.4 寄存器说明

29.4.1 Flash操作控制寄存器(FLCMD)

Flash操作控制寄存器用于对FLASH操作进行选择。

位	符号	描述	复位值
31:8	-	保留	-
7:0	CMD	操作选择: 56H: CHIP 擦除 5CH: Sector 擦除 AAH: 写入	0x0

29.4.2 FLASH 密钥寄存器(FLKEY)

借助 Flash 密钥寄存器，可允许对 Flash 控制寄存器的访问，进而允许执行编程和擦除操作。

位	符号	描述	复位值
31:0	KEY	FLASH寄存器解锁并允许对其执行编程/擦除操作必须顺序编程以下值： (a): KEY = 0x45670123 (b): KEY = 0xCDEF89AB	0x0

29.4.3 FLASH 数据寄存器0(FLDATA0)

Flash数据寄存器0，将要写入的值写入该寄存器。

位	符号	描述	复位值
31:0	DATA0	FLASH写入的数据的低32位	0xffffffff

29.4.4 FLASH 数据寄存器1(FLDATA1)

Flash数据寄存器1，将要写入的值写入该寄存器。

位	符号	描述	复位值
31:0	DATA1	FLASH写入的数据的高32位	0xffffffff

29.4.5 Flash 地址寄存器(FLADDR)

Flash地址寄存器，将要写入或擦除的地址写入该寄存器。

位	符号	描述	复位值
31:24	-	保留	0x0
23:0	ADDR	写入或擦除的地址	0xffffffff

29.4.6 Flash 控制寄存器(FLCR)

Flash 控制寄存器用于配置和启动 Flash 操作。

位	符号	描述	复位值
31	LOCK	该位只能写入 1。该位置 1 时，表示 FLASH_CR 寄存器已锁定。当检测到解锁序列时，由硬件将该位清零。如果解锁操作失败，该位仍保持置 1，直到下一次复位	1
30:18	-	保留	0
17	ERRIE	错误中断使能： 0: 禁止错误中断 1: 使能错误中断	0x0
16	EOPIE	操作结束中断使能： 0: 禁止中断 1: 使能中断	0x0
15:10	-	保留	0
9:8	PSIZE	写入数据字长： 00: 禁止设置 01: X16 10: X32 11: 禁止设置	0x11
7:1	-	保留	0x0
0	STRT	FLASH擦写启动位： 0: 在BSY位清零后硬件清零 1: 触发擦除或写入操作(只能软件置1)	0

29.4.7 Flash 状态寄存器(FLSR)

Flash 状态寄存器提供正在执行的编程和擦除操作的相关信息。

位	符号	描述	复位值
31:17	-	保留	0x0
16	BSY	繁忙 ^{注1} : 0: 当前未执行任何 Flash 操作 1: 正在执行 Flash 操作（该位指示 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时清零。）	0
15:7	-	保留	0x0
6	PGPERR	编程并行位数错误: 0: 编程并行位数正常 1: 代码在控制寄存器未正确配置的情况下对 Flash 执行写访问，将由硬件为该位置 1。需软件写1清零	0
5	-	保留	0
4	WRPERR	写保护错误: 0: 写保护正常 1: 擦除/编程的地址属于 Flash 中处于写保护状态的区域，将由硬件为该位置 1。需软件写1清零	0
3:2	-	保留	0x0
1	OPERR	操作错误: 0: 操作正常 1: 检测到Flash操作（编程/擦除/读取）请求，但由于存在并行位数错误、对齐错误或写保护错误而无法运行，将由硬件对该位置1。	0
0	EOP	操作结束: 0: 操作未完成 1: 当成功完成一个或多个 Flash 操作（编程/擦除）时，由硬件将该位置 1。需软件写1清零。	0

注1: BSY只读。

29.4.8 Flash 访问控制寄存器（FLACR）

Flash 访问控制寄存器用于使能/关闭加速功能，并且可根据 CPU 频率控制 Flash 访问时间

位	符号	描述	复位值
31:8	-	保留	0x0
7	FLWAITEN	等待使能： 0: 禁止等待 1: 使能等待	0
6:4	-	保留	0x0
3:0	LATENCY	延迟（这些位表示 CPU 时钟周期与 Flash 访问时间之比）： 0000: 零等待周期 0001 一个等待周期 0010 两个等待周期 1111: 15个等待周期	0x0

29.4.9 Flash 区域控制寄存器（FLREGION）

Flash 区域选择寄存器用于选择FLASH访问区域。

位	符号	描述	复位值
31:8	-	保留	0x0
7:0	REGION	FLASH区域选择位： 55H: 程序闪存区 AAH: 数据闪存区	0x0

29.4.10 Flash 保护寄存器（FLPRT）

Flash保护寄存器用于保护程序区和数据区内容。

位	符号	描述	复位值
31	PRTALL	全空间保护控制位： 0: 全空间不写保护 1: 全空间写保护	0
30:29	-	保留	0x0
28	PRTDATA4	数据区(50_0C00H-50_0FFFH)保护控制位： 0: 不写保护 1: 写保护	0
27	PRTDATA3	数据区(50_0800H-50_0BFFH)保护控制位： 0: 不写保护 1: 写保护	0
26	PRTDATA2	数据区（50_0400H-50_07FFH）保护控制位 0: 不写保护 1: 写保护	0
25	PRTDATA1	数据区（50_0200H-50_03FFH）保护控制位： 0: 不写保护 1: 写保护	0
24:19	-	保留	0x0
18	PRTROM18	程序区252K-256K(3_F000H-3_FFFFH)保护控制位： 0: 不写保护 1: 写保护	0
17	PRTROM17	程序区248K-252K(3_E000H-3_EFFFH)保护控制位： 0: 不写保护 1: 写保护	0
16	PRTROM16	程序区244K-248K(3_D000H-3_DFFFH)保护控制位： 0: 不写保护 1: 写保护	0
15	PRTROM15	程序区240K-244K(3_C000H-3_CFFFH)保护控制位： 0: 不写保护 1: 写保护	0
14	PRTROM14	程序区224K-240K(3_8000H-3_BFFFH)保护控制位： 0: 不写保护 1: 写保护	0
13	PRTROM13	程序区208K-224K(3_4000H-3_7FFFH)保护控制位： 0: 不写保护 1: 写保护	0
12	PRTROM12	程序区192K-208K(3_0000H-3_3FFFH)保护控制位： 0: 不写保护 1: 写保护	0
11	PRTROM11	程序区176K-192K(2_C000H-2_FFFFH)保护控制位：	0

		0: 不写保护 1: 写保护	
10	PRTROM10	程序区160K-176K(2_8000H-2_BFFFH)保护控制位: 0: 不写保护 1: 写保护	0
9	PRTROM9	程序区144K-160K(2_4000H-2_7FFFH)保护控制位: 0: 不写保护 1: 写保护	0
8	PRTROM8	程序区128K-144K(2_0000H-2_3FFFH)保护控制位: 0: 不写保护 1: 写保护	0
7	PRTROM7	程序区112K-128K(1_C000H-1_FFFFH)保护控制位: 0: 不写保护 1: 写保护	0
6	PRTROM6	程序区96K-112K(1_8000H-1_BFFFH)保护控制位: 0: 不写保护 1: 写保护	0
5	PRTROM5	程序区80K-96K(1_4000H-1_7FFFH)保护控制位: 0: 不写保护 1: 写保护	0
4	PRTROM4	程序区64K-80K(1_0000H-1_3FFFH)保护控制位: 0: 不写保护 1: 写保护	0
3	PRTROM3	程序区48K-64K(C000H-FFFFH)保护控制位: 0: 不写保护 1: 写保护	0
2	PRTROM2	程序区32K-48K(8000H-BFFFH)保护控制位: 0: 不写保护 1: 写保护	0
1	PRTROM1	程序区16K-32K(4000H-7FFFH)保护控制位: 0: 不写保护 1: 写保护	0
0	PRTROM0	程序区0K-16K(0000H-3FFFH)保护控制位: 0: 不写保护 1: 写保护	0

29.5 FLASH操作方法

29.5.1 编程

1.写入操作流程如下：

(1) 解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

(2) 解除编程区域的保护：FLPRT.PRTROMn=0(n=0-18)或 FLPRT.PRTDATAm=0(m=1-4)。

(3) 设置写入模式 FLCMD=0xAA。

(4) 设置编程区域 FLREGION。

(5) 将目标地址写入 FLADDR。

(6) 将写入数据写入 FLDATA0/ FLDATA1。

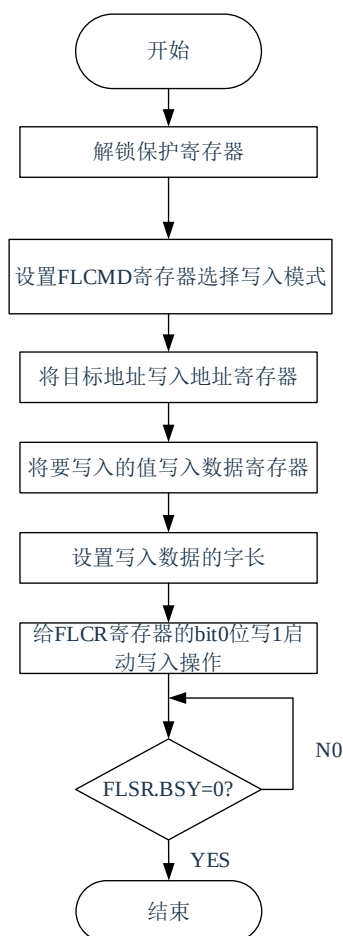
(7) 设置要写入数据字长 FLCR.PSIZE。

(8) 启动编程 FLCR.STRT=1。

(9) 软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

(10) 进行下次操作前，软件置”1”来清除 FLSR.EOP。

(11) 操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



29.5.2 扇擦除

sector 擦除，擦除时间由硬件实现。操作流程如下：

(1) 解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

(2) 解除编程区域的保护：FLPRT.PRTROMn=0(n=0-18)或 FLPRT.PRTDATAm=0(m=1-4)。

(3) 设置扇擦除模式 FLCMD=0x5C。

(4) 设置编程区域 FLREGION。

(5) 将目标地址写入 FLADDR（写擦除目标 sector 的首地址）。

(6) 启动编程 FLCR.STRT=1。

(7) 软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

(8) 进行下次操作前，软件置”1”来清除 FLSR.EOP。

(9) 操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



29.5.3 全片擦除（chip erase）

chip擦除，擦除时间由硬件实现。操作流程如下：

（1）解锁 FLASH 寄存器：按照顺序写

FLKEY=0x45670123;

FLKEY=0xCDEF89AB; 解锁后 FLCR.LOCK=0;

（2）解除编程区域的保护：FLPRT.PRTROMn=0(n=0-18)或 FLPRT.PRTDATAm=0(m=1-4),

FLPRT.PRTALL=0。

（3）设置全擦除模式 FLCMD=0x56。

（4）设置编程区域 FLREGION=0x55（全擦除只擦除程序闪存区）。

（5）将目标地址写入 FLADDR（写程序闪存区首地址）。

（6）启动编程 FLCR.STRT=1。

（7）软件查询状态寄存器 FLSR.BSY =0 时，表示操作完成；或软件查询状态寄存器 FLSR.EOP=1，表示操作完成。

（8）进行下次操作前，软件置”1”来清除 FLSR.EOP。

（9）操作完毕后，FLCR.LOCK=1,重新锁定 FLCR。



29.6 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。擦写参数的设置由硬件实现,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。

第30章 附录 修订记录

版本	日期	修订内容
V0.3.0	2025年3月	初始版本
V0.3.1	2025年4月	修改部分描述错误
V0.3.2	2025年8月	1. 修改18.1说明 2. 第2章新增GPIO结构说明 3. 修改29.4.6、29.4.1描述