CMS32H6157 用户手册

基于 ARM® Cortex®-M0+的超低功耗 32 位测量 SOC

Rev. 0.2.0

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 CMS32H6157 微控制器产品的用户手册，用户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。

用户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：

CMS32H6157 数据手册_vx.x.x pdf

通常在芯片选型的初期，首先要看数据手册，以评估该产品是否能够满足设计上的功能需求；在基本选定所需产品后，需要查看用户手册，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，需要详细阅读用户手册，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 ARM 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 20
1.1 概述 20
1.2 Cortex-M0+内核特性 20
1.3 调试特性 20
1.4 SWD 接口引脚 22
1.5 ARM 参考文档 23
第 2 章 端口功能 24
2.1 端口通用功能 24
2.2 端口复用功能 25
2.3 控制端口功能的寄存器 28
2.3.1 端口输出控制寄存器（PMxx） 30
2.3.2 端口寄存器（Pxx） 31
2.3.3 端口置位控制寄存器（PSETxx） 32
2.3.4 端口清零控制寄存器（PCLRxx） 33
2.3.5 上拉电阻选择寄存器（PUxx） 34
2.3.6 下拉电阻选择寄存器（PDxx） 35
2.3.7 端口输出模式寄存器（POMxx） 36
2.3.8 端口模式控制寄存器（PMCxx） 37
2.3.9 端口回读寄存器（PREADxx） 38
2.3.10 端口复用功能配置寄存器（PxxCFG） 39
2.3.11 外部中断端口选择寄存器（INTPnPCFG） 42
2.3.12 外部复位端口屏蔽寄存器（RSTM） 45
2.4 未使用端口的处理 46
2.5 使用复用功能时的寄存器设定 47
2.5.1 使用复用功能时的基本思想 47
2.5.2 端口复用功能的配置方法 48
第 3 章 系统结构 59
3.1 概述 59
3.2 系统地址划分 60
3.3 外设地址分配 61
第 4 章 时钟发生电路 62
4.1 时钟发生电路的功能 62
4.2 时钟发生电路的结构 64
4.3 控制时钟发生电路的寄存器 67

4.3.1 时钟运行模式控制寄存器（CMC）.....67
4.3.2 系统时钟控制寄存器（CKC）.....69
4.3.3 时钟运行状态控制寄存器（CSC）.....70
4.3.4 振荡稳定时间计数器的状态寄存器（OSTC）.....71
4.3.5 振荡稳定时间选择寄存器（OSTS）.....73
4.3.6 外围允许寄存器 0、1（PER0、PER1）.....74
4.3.7 副系统时钟提供模式控制寄存器（OSMC）.....77
4.3.8 高速内部振荡器的频率选择寄存器（HOCODIV）.....78
4.3.9 高速内部振荡器的微调寄存器（HIOTRM）.....79
4.3.10 副系统时钟选择寄存器（SUBCKSEL）.....80
4.4 系统时钟振荡电路.....81
4.4.1 X1 振荡电路.....81
4.4.2 XT1 振荡电路.....82
4.4.3 高速内部振荡器.....85
4.4.4 低速内部振荡器.....85
4.5 时钟发生电路的运行.....86
4.6 时钟控制.....88
4.6.1 高速内部振荡器的设置例子.....88
4.6.2 X1 振荡电路的设置例子.....90
4.6.3 XT1 振荡电路的设置例子.....91
4.6.4 CPU 时钟的状态转移图.....92
4.6.5 CPU 时钟转移前的条件和转移后的处理.....98
4.6.6 CPU 时钟和主系统时钟的切换所需时间.....100
4.6.7 时钟振荡停止前的条件.....101
4.7 高速内振校正功能.....102
4.7.1 高速内振自调整功能.....102
4.7.2 寄存器说明.....103
4.7.2.1 高速内振频率校正控制寄存器（HOCOFC）.....103
4.7.3 动作说明.....104
4.7.3.1 动作概要.....104
4.7.3.2 动作设置流程.....107
4.7.4 使用注意事项.....108
4.7.4.1 SFR 访问.....108
4.7.4.2 复位时动作.....108
4.8 发振停止检出电路的功能.....109
4.8.1 发振停止检测电路的构成.....109

4.8.2 发振停止检测电路使用的寄存器 110
4.8.2.1 外围允许寄存寄存器 1（PER1） 110
4.8.2.2 发振停止检测控制寄存器（SCMCTL） 110
4.8.2.3 发振停止检测模式寄存器（SCMMD） 111
4.8.2.4 发振停止检测状态寄存器（SCMST） 111
4.8.3 发振停止检测电路的动作 112
4.8.3.1 发振停止检测电路的动作方法 112
4.8.4 深度睡眠模式下发振停止检测电路的动作 113
4.8.5 发振停止检测功能的注意事项 113
第 5 章 通用定时器单元 TIMER8 114
5.1 通用定时器单元的功能 116
5.1.1 独立通道运行功能 116
5.1.2 多通道联动运行功能 118
5.1.3 LIN-bus 支持功能（只限于通道 3） 119
5.2 通用定时器单元的结构 120
5.2.1 通用定时器单元寄存器列表 122
5.2.2 定时器计数寄存器mn （TCRmn） 123
5.2.3 定时器数据寄存器 mn（TDRmn） 125
5.3 控制通用定时器单元的寄存器 126
5.3.1 外围允许寄存器 0（PER0） 127
5.3.2 定时器时钟选择寄存器m（TPSm） 128
5.3.3 定时器模式寄存器 mn（TMRmn） 131
5.3.4 定时器状态寄存器 mn（TSRmn） 135
5.3.5 定时器通道允许状态寄存器m（TEm） 136
5.3.6 定时器通道开始寄存器 m（TSm） 137
5.3.7 定时器通道停止寄存器 m（TTm） 138
5.3.8 定时器输入输出选择寄存器（TIOS0） 139
5.3.9 定时器输出允许寄存器 m（TOEm） 140
5.3.10 定时器输出寄存器 m（TOm） 141
5.3.11 定时器输出电平寄存器 m（TOLm） 142
5.3.12 定时器输出模式寄存器 m（TOMm） 143
5.3.13 输入切换控制寄存器（ISC） 144
5.3.14 噪声滤波器允许寄存器（NFEN1） 145
5.3.15 控制定时器输入/输出引脚端口功能的寄存器 146
5.4 通用定时器单元的基本规则 147
5.4.1 多通道联动运行功能的基本规则 147

5.4.2 定时器通道开始寄存器 m（TSm）.....150
5.5 计数器的运行.....151
5.5.1 计数时钟（F_{TCLK}）.....151
5.5.2 计数器的开始时序.....153
5.5.3 计数器的运行.....154
5.6 通道输出（TOmn 引脚）的控制.....159
5.6.1 TOmn 引脚输出电路的结构.....159
5.6.2 TOmn 引脚的输出设定.....160
5.6.3 通道输出运行的注意事项.....161
5.6.4 TOmn 位的一次性操作.....165
5.6.5 有关开始计数时的定时器中断和 TOmn 引脚输出.....166
5.7 定时器输入（TImn）的控制.....167
5.7.1 TImn 引脚输入电路的结构.....167
5.7.2 噪声滤波器.....167
5.7.3 操作通道输入时的注意事项.....168
5.8 通用定时器单元的独立通道运行功能.....169
5.8.1 作为间隔定时器/方波输出的运行.....169
5.8.2 作为外部事件计数器的运行.....173
5.8.3 作为分频器的运行.....176
5.8.4 作为输入脉冲间隔测量的运行.....179
5.8.5 作为输入信号高低电平宽度测量的运行.....182
5.8.6 作为延迟计数器的运行.....186
5.9 通用定时器单元的多通道联动运行功能.....189
5.9.1 作为单触发脉冲输出功能的运行.....189
5.9.2 作为 PWM 功能的运行.....196
5.9.3 作为多重 PWM 输出功能的运行.....203
5.10 使用通用定时器单元时的注意事项.....212
5.10.1 使用定时器输出时的注意事项.....212
第 6 章 定时器 A.....213
6.1 定时器 A 的功能.....213
6.2 定时器 A 的结构.....214
6.3 控制定时器 A 的寄存器.....215
6.3.1 外围允许寄存器 0（PER0）.....216
6.3.2 副系统时钟提供模式控制寄存器（OSMC）.....217
6.3.3 定时器 A 计数寄存器 0（TA0）.....218
6.3.4 定时器 A 控制寄存器 0（TACR0）.....219

6.3.5 定时器 AI/O 控制寄存器 0（TAIOC0）.....220
6.3.6 定时器 A 控制寄存器 0（TAMR0）.....222
6.3.7 定时器 A 事件引脚选择寄存器 0（TAISR0）.....223
6.4 定时器 A 的运行.....224
6.4.1 重加载寄存器和计数器的改写.....224
6.4.2 定时器模式.....225
6.4.3 脉冲输出模式.....226
6.4.4 事件计数器模式.....227
6.4.5 脉宽测量模式.....229
6.4.6 脉冲周期测量模式.....230
6.4.7 与 EVENTC 的协作.....231
6.4.8 各模式的输出设置.....232
6.5 使用定时器 A 时的注意事项.....233
6.5.1 计数的开始和停止控制.....233
6.5.2 标志的存取（TACR0 寄存器的 TEDGF 位和 TUNDF 位）.....233
6.5.3 计数寄存器的存取.....234
6.5.4 模式的变更.....234
6.5.5 TAO 引脚和 TAIO 引脚的设置步骤.....234
6.5.6 不使用定时器 A 的情况.....235
6.5.7 定时器 A 运行时钟的停止.....235
6.5.8 深度睡眠模式（事件计数器模式）的设置步骤.....235
6.5.9 深度睡眠模式中（只限于事件计数器模式）的功能限制.....235
6.5.10 通过 TSTOP 位进行强制的计数停止.....235
6.5.11 数字滤波器.....235
6.5.12 选择 F_{IL} 作为计数源的情况.....236
第 7 章 实时时钟.....237
7.1 实时时钟的功能.....237
7.2 实时时钟的结构.....237
7.3 控制实时时钟的寄存器.....239
7.3.1 外围允许寄存器 0（PER0）.....240
7.3.2 实时时钟选择寄存器（RTCCL）.....241
7.3.3 实时时钟控制寄存器 0（RTCC0）.....242
7.3.4 实时时钟控制寄存器 1（RTCC1）.....243
7.3.5 时钟误差校正寄存器（SUBCUD）.....245
7.3.6 秒计数寄存器（SEC）.....246
7.3.7 分钟计数寄存器（MIN）.....247

7.3.8 小时计数寄存器（HOUR）....248
7.3.9 日计数寄存器（DAY）....250
7.3.10 星期计数寄存器（WEEK）....251
7.3.11 月计数寄存器（MONTH）....252
7.3.12 年计数寄存器（YEAR）....253
7.3.13 闹钟分钟寄存器（ALARMWM）....254
7.3.14 闹钟小时寄存器（ALARMWH）....255
7.3.15 闹钟星期寄存器（ALARMWW）....256
7.3.16 实时时钟的运行开始....257
7.3.17 开始运行后睡眠模式的转移....258
7.3.18 实时时钟计数器的读写....259
7.3.19 实时时钟的闹钟设定....261
7.3.20 实时时钟的 1Hz 输出....262
7.3.21 实时时钟的时钟误差校正例子....263
第 8 章 15 位间隔定时器....265
8.1 15 位间隔定时器的功能....265
8.2 15 位间隔定时器的结构....265
8.3 控制 15 位间隔定时器的寄存器....266
8.3.1 外围允许寄存器 0（PER0）....266
8.3.2 实时时钟选择寄存器（RTCCL）....267
8.3.3 15 位间隔定时器的控制寄存器（ITMC）....268
8.4 15 位间隔定时器的运行....269
8.4.1 15 位间隔定时器的运行时序....269
8.4.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移....270
第 9 章 时钟输出/蜂鸣器输出控制电路....271
9.1 时钟输出/蜂鸣器输出控制电路的功能....271
9.2 控制时钟输出/蜂鸣器输出控制电路的寄存器....272
9.2.1 时钟输出选择寄存器 n（CKSn）....272
9.3 时钟输出/蜂鸣器输出控制电路的运行....274
9.4 时钟输出/蜂鸣器输出控制电路的注意事项....274
第 10 章 看门狗定时器....275
10.1 看门狗定时器的功能....275
10.2 看门狗定时器的结构....275
10.3 控制看门狗定时器的寄存器....277
10.3.1 看门狗定时器的允许寄存器（WDTE）....277
10.3.2 LOCKUP 控制寄存器（LOCKCTL）及其保护寄存器（PRCR）....278

10.3.3 WDTCFG 配置寄存器（WDTCFG0/1/2/3）.....279
10.4 看门狗定时器的运行.....280
10.4.1 看门狗定时器的运行控制.....280
10.4.2 看门狗定时器上溢时间的设定.....281
10.4.3 看门狗定时器窗口打开期间的设定.....282
10.4.4 看门狗定时器间隔中断的设定.....283
10.4.4.1 LOCKUP 期间看门狗定时器的运行.....283
10.4.4.2 WDTCFG 未配置时看门狗定时器的运行.....283
第 11 章 A/D 转换器.....284
11.1 A/D 转换器的功能.....284
11.2 控制 A/D 转换器的寄存器.....286
11.2.1 外围允许寄存器（PER1）.....287
11.2.2 A/D 转换器的模式寄存器 0（ADM0）.....288
11.2.3 A/D 转换器的模式寄存器 1（ADM1）.....293
11.2.4 A/D 转换器的模式寄存器 2（ADM2）.....294
11.2.5 A/D 转换器的触发模式寄存器（ADTRG）.....295
11.2.6 模拟输入通道指定寄存器（ADS）.....296
11.2.7 12 位 A/D 转换结果寄存器（ADCR）.....299
11.2.8 8 位 A/D 转换结果寄存器（ADCRH）.....300
11.2.9 转换结果比较上限值设定寄存器（ADUL）.....300
11.2.10 转换结果比较下限值设定寄存器（ADLL）.....301
11.2.11 A/D 采样时间控制寄存器（ADNSMP）.....302
11.2.12 A/D 采样时间延长寄存器（ADSMPWAIT）.....304
11.2.13 A/D 测试寄存器（ADTES）.....305
11.2.14 A/D 充放电控制寄存器（ADNDIS）.....306
11.3 输入电压和转换结果.....307
11.4 A/D 转换器的运行模式.....308
11.4.1 软件触发模式（选择模式、连续转换模式）.....308
11.4.2 软件触发模式（选择模式、单次转换模式）.....309
11.4.3 软件触发模式（扫描模式、连续转换模式）.....310
11.4.4 软件触发模式（扫描模式、单次转换模式）.....311
11.4.5 硬件触发无等待模式（选择模式、连续转换模式）.....312
11.4.6 硬件触发无等待模式（选择模式、单次转换模式）.....313
11.4.7 硬件触发无等待模式（扫描模式、连续转换模式）.....314
11.4.8 硬件触发无等待模式（扫描模式、单次转换模式）.....315
11.4.9 硬件触发等待模式（选择模式、连续转换模式）.....316

11.4.10 硬件触发等待模式（选择模式、单次转换模式）.....317
11.4.11 硬件触发等待模式（扫描模式、连续转换模式）.....318
11.4.12 硬件触发等待模式（扫描模式、单次转换模式）.....319
11.5 A/D 转换器的设定流程图.....320
11.5.1 软件触发模式的设定.....320
11.5.2 硬件触发无等待模式的设定.....321
11.5.3 硬件触发等待模式的设定.....322
11.5.4 选择温度传感器的输出电压/内部基准电压时的设定.....323
11.5.5 测试模式的设定.....324
第 12 章 SIGMA-DELTA ADC.....325
12.1 概述.....325
12.2 基本结构功能描述.....325
12.3 ADC 工作原理.....326
12.3.1 LDO.....326
12.3.2 模拟输入前端.....326
12.3.3 温度传感器.....326
12.3.4 低噪声 PGA 放大器.....326
12.3.5 ADC 时钟、数据输出速率.....327
12.3.6 复位和休眠模式.....328
12.3.7 建立时间.....328
12.4 SPI 串口通信.....329
12.4.1 数据格式.....329
12.4.2 数据准备/数据输入输出（DRDYB/DOUT）.....329
12.4.3 串行时钟输入（SCLK）.....329
12.4.4 串行数据发送.....330
12.4.5 功能配置.....331
12.4.6 SPI 命令字说明.....332
12.4.7 SPI 通信注意事项.....332
12.5 相关寄存器.....333
12.5.1 Sigma-Delta ADC 控制寄存器 1.....333
12.5.2 Sigma-Delta ADC 控制寄存器 2.....334
12.5.3 Sigma-Delta ADC 控制寄存器 3.....335
12.5.4 Sigma-Delta ADC 控制寄存器 4.....336
第 13 章 D/A 转换器.....337
13.1 D/A 转换器的功能.....337
13.2 D/A 转换器的结构.....338

13.3 控制 D/A 转换器的寄存器 339
13.3.1 外围允许寄存器 1（PER1） 339
13.3.2 D/A 转换器的模式寄存器（DAM） 340
13.3.3 D/A 转换值设置寄存器 i（DACSi）（i=0） 340
13.3.4 事件输出目标选择寄存器 n（ELSELRn）、n=00~15....340
13.4 D/A 转换器的运行 341
13.4.1 通常模式的运行 341
13.4.2 实时输出模式的运行 342
13.4.3 D/A 转换值的输出时序 343
13.5 使用 D/A 转换器时的注意事项 344
第 14 章 比较器 345
14.1 比较器的功能 345
14.2 比较器的结构 346
14.3 控制比较器的寄存器 348
14.3.1 外围允许寄存器 1（PER1） 349
14.3.2 比较器模式设定寄存器（COMPMDR） 350
14.3.3 比较器滤波控制寄存器（COMPFIR） 351
14.3.4 比较器输出控制寄存器（COMPOCR） 352
14.3.5 比较器负端基准选择寄存器（CnREFS） 354
14.3.6 比较器正端输入选择寄存器（CMPSELn） 355
14.3.7 比较器的迟滞控制寄存器（CMPnHY） 356
14.4 运行说明 357
14.4.1 比较器 n 的数字滤波器（n=0、1） 359
14.4.2 比较器 n 中断（n=0、1） 359
14.4.3 向联动控制器（EVENTC）输出的事件信号 360
14.4.4 比较器 n 的输出（n=0、1） 361
14.4.5 比较器时钟的停止和提供 362
第 15 章 运算放大器（OPA） 363
15.1 运算放大器的功能 363
15.2 运算放大器的寄存器 364
15.2.1 外围允许寄存器 1(PER1) 364
15.2.2 运算放大器控制寄存器（OPACTL） 365
15.2.3 运算放大器数模控制寄存器（OPADAC） 366
第 16 章 通用串行通信单元 367
16.1 通用串行通信单元的功能 368
16.1.1 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21） 368

16.1.2 UART（UART0~UART2）.....369
16.1.3 简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）.....370
16.2 通用串行通信单元的结构.....371
16.2.1 移位寄存器.....373
16.2.2 串行数据寄存器 mn（SDRmn）.....373
16.3 控制通用串行通信单元的寄存器.....375
16.3.1 外围允许寄存器 0（PER0）.....377
16.3.2 串行时钟选择寄存器 m（SPSm）.....378
16.3.3 串行模式寄存器 mn（SMRmn）.....379
16.3.4 串行通信运行设定寄存器 mn（SCRmn）.....381
16.3.5 串行数据寄存器 mn（SDRmn）.....384
16.3.6 串行标志清除触发寄存器 mn（SIRmn）.....385
16.3.7 串行状态寄存器 mn（SSRmn）.....386
16.3.8 串行通道开始寄存器 m（SSm）.....388
16.3.9 串行通道停止寄存器 m（STm）.....389
16.3.10 串行通道允许状态寄存器 m（SEm）.....390
16.3.11 串行输出允许寄存器 m（SOEm）.....391
16.3.12 串行输出寄存器 m（SOm）.....392
16.3.13 串行输出电平寄存器 m（SOLm）.....393
16.3.14 串行待机控制寄存器 m（SSCm）.....394
16.3.15 从属选择功能启用寄存器 m（SSEm）.....395
16.3.16 输入切换控制寄存器（ISC）.....396
16.3.17 噪声滤波器允许寄存器 0（NFEN0）.....397
16.4 运行停止模式.....398
16.4.1 以单元为单位停止运行的情况.....398
16.4.2 按通道停止运行的情况.....399
16.5 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的运行.....400
16.5.1 主控发送.....401
16.5.2 主控接收.....409
16.5.3 主控的发送和接收.....418
16.5.4 从属发送.....426
16.5.5 从属接收.....435
16.5.6 从属的发送和接收.....441
16.5.7 传送时钟频率的计算.....450
16.5.8 在 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信过程中发生错误时的处理步骤.....452
16.6 从属选择输入功能的时钟同步串行通信的运行.....453

16.6.1 从属发送 456
16.6.2 从属接收 466
16.6.3 从属的发送和接收....473
16.6.4 传送时钟频率的计算....483
16.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤....484
16.7 UART（UART0~UART2）通信的运行 485
16.7.1 UART 发送 486
16.7.2 UART 接收 494
16.7.3 低功耗 UART 模式功能....502
16.7.4 波特率的计算....508
16.7.5 在 UART（UART0~UART2）通信过程中发生错误时的处理步骤....512
16.8 LIN 通信的运行....513
16.8.1 LIN 发送....513
16.8.2 LIN 接收....516
16.9 简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的运行 521
16.9.1 地址段发送 522
16.9.2 数据发送 527
16.9.3 数据接收 530
16.9.4 停止条件的产生 534
16.9.5 传送速率的计算 535
16.9.6 在简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信过程中发生错误时的处理步骤 537
第 17 章 串行接口 SPI 538
17.1 串行接口 SPI 的功能 538
17.2 串行接口 SPI 的结构 538
17.3 控制串行接口 SPI 的寄存器 539
17.3.1 外围允许寄存器 1（PER1） 539
17.3.2 SPI 操作模式寄存器（SPIM） 540
17.3.3 SPI 时钟选择寄存器（SPIC） 541
17.3.4 SPI 状态寄存器（SPIS） 542
17.3.5 发送缓冲寄存器（SDRO） 543
17.3.6 接收缓冲寄存器（SDRI） 543
17.4 串行接口 SPI 的操作 544
17.4.1 主控的发送和接收....545
17.4.2 主控的接收 548
17.4.3 从属的发送和接收....551

17.4.4 从属的接收 554
第 18 章 串行接口 IICA 557
18.1 串行接口 IICA 的功能 557
18.1.1 运行停止模式 557
18.1.2 I2C 总线模式（支持多主控） 557
18.1.3 唤醒模式 557
18.2 串行接口 IICA 的结构 560
18.2.1 IICA 移位寄存器 n（IICAn） 561
18.2.2 从属地址寄存器 n（SVAn） 561
18.2.3 SO 锁存器 561
18.2.4 唤醒控制电路 561
18.2.5 串行时钟计数器 562
18.2.6 中断请求信号发生电路 562
18.2.7 串行时钟控制电路 562
18.2.8 串行时钟等待控制电路 562
18.2.9 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路 562
18.2.10 数据保持时间校正电路 562
18.2.11 开始条件生成电路 562
18.2.12 停止条件生成电路 562
18.2.13 总线状态检测电路 563
18.3 控制串行接口 IICA 的寄存器 564
18.3.1 外围允许寄存器 0（PER0） 564
18.3.2 IICA 控制寄存器 n0（IICCTLn0） 565
18.3.3 IICA 状态寄存器 n（IICSn） 569
18.3.4 IICA 标志寄存器 n（IICFn） 572
18.3.5 IICA 控制寄存器 n1（IICCTLn1） 574
18.3.6 IICA 低电平宽度设定寄存器 n（IICWLn） 576
18.3.7 IICA 高电平宽度设定寄存器 n（IICWHn） 576
18.3.8 控制 IICA 引脚端口功能的寄存器 577
18.4 I2C 总线模式的功能 578
18.4.1 引脚结构 578
18.4.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法 579
18.5 I2C 总线的定义和控制方法 581
18.5.1 开始条件 582
18.5.2 地址 583
18.5.3 传送方向的指定 583

18.5.4 应答（ACK）.....584
18.5.5 停止条件.....585
18.5.6 等待.....586
18.5.7 等待的解除方法.....588
18.5.8 中断请求（INTIICAn）的产生时序和等待控制.....589
18.5.9 地址匹配的检测方法.....591
18.5.10 错误的检测.....591
18.5.11 扩展码.....592
18.5.12 仲裁.....593
18.5.13 唤醒功能.....595
18.5.14 通信预约.....598
18.5.15 其他注意事项.....601
18.5.16 通信运行.....602
18.5.17 I^2C 中断请求（INTIICAn）的产生时序.....610
18.6 时序图.....630
第 19 章 IRDA.....645
19.1 IrDA 的功能.....645
19.2 控制 IrDA 的寄存器.....646
19.2.1 外围允许寄存器 0（PER0）.....646
19.2.2 IrDA 控制寄存器（IRCR）.....647
19.3 IrDA 的运行.....648
19.3.1 IrDA 通信的操作步骤.....648
19.3.2 发送.....649
19.3.3 接收.....649
19.3.4 高电平脉宽的选择.....650
19.4 使用 IrDA 时的注意事项.....651
第 20 章 LCD 控制器/驱动器.....652
20.1 LCD 控制器 /驱动器的功能.....652
20.2 LCD 控制器 /驱动器的结构.....653
20.3 控制 LCD 控制器 /驱动器的寄存器.....655
20.3.1 LCD 模式寄存器 0（LCDM0）.....656
20.3.2 LCD 模式寄存器 1（LCDM1）.....658
20.3.3 副系统时钟提供模式控制寄存器（OSMC）.....660
20.3.4 LCD 时钟控制寄存器 0 （LCDC0）.....661
20.3.5 LCD 升压电平控制寄存器（VLCD）.....662
20.3.6 LCD 输入切换控制寄存器（ISCLCD）.....663

20.3.7 LCD 端口功能寄存器.....665
20.4 LCD 显示数据寄存器.....666
20.5 LCD 显示寄存器的选择.....669
20.5.1 A 图形区和 B 图形区的数据显示.....670
20.5.2 闪烁显示（A 图形区和 B 图形区的数据的交替显示）.....670
20.6 LCD 驱动电压 V_{L1}、V_{L2}、V_{L3}、V_{L4} 的提供.....671
20.6.1 内部电阻分割方式.....671
20.6.2 外部电阻分割方式.....672
20.6.3 内部升压方式.....673
20.6.4 电容分割方式.....674
第 21 章 增强型 DMA.....675
21.1 DMA 的功能.....675
21.2 DMA 的结构.....677
21.3 控制 DMA 的寄存器.....678
21.3.1 DMA 控制数据区和 DMA 向量表区的分配.....679
21.3.2 控制数据的分配.....680
21.3.3 向量表.....682
21.3.4 外围允许寄存器 1（PER1）.....684
21.3.5 DMA 控制寄存器 j（DMACRj）（j=0~23）.....685
21.3.6 DMA 块大小寄存器 j（DMBLSj）（j=0~23）.....687
21.3.7 DMA 传送次数寄存器 j（DMACTj）（j=0~23）.....688
21.3.8 DMA 传送次数重加载寄存器 j（DMRLDj）（j=0~23）.....689
21.3.9 DMA 源地址寄存器 j（DMSARj）（j=0~23）.....690
21.3.10 DMA 目标地址寄存器 j（DMDARj）（j=0~23）.....691
21.3.11 DMA 启动允许寄存器 i（DMAENi）（i=0~2）.....692
21.3.12 DMA 基址寄存器（DMABAR）.....694
21.3.13 DMAENi 置位寄存器 i（DMSETi）（i=0~2）.....695
21.3.14 DMAENi 复位寄存器 i（DMCLRi）（i=0~2）.....695
21.4 DMA 的运行.....696
21.4.1 启动源.....696
21.4.2 正常模式.....697
21.4.3 重复模式.....700
21.4.4 链传送.....703
21.5 使用 DMA 时的注意事项.....705
21.5.1 DMA 控制数据和向量表的设置.....705
21.5.2 DMA 控制数据区和 DMA 向量表区的分配.....705

21.5.3 DMA 的执行时钟数 706
21.5.4 DMA 的响应时间 707
21.5.5 DMA 的启动源 707
21.5.6 待机模式中的运行 708
第 22 章 联动控制器(EVENTC) 709
22.1 EVENTC 的功能 709
22.2 EVENTC 的结构 709
22.3 控制寄存器 710
22.3.1 输出目标选择寄存器 n（ELSELRn）（n=00~15） 711
22.4 EVENTC 的运行 714
第 23 章 中断功能 716
23.1 中断功能的种类 716
23.2 中断源和结构 716
23.3 控制中断功能的寄存器 720
23.3.1 中断请求标志寄存器（IF00~IF31） 720
23.3.2 中断屏蔽标志寄存器（MK00~MK31） 721
23.3.3 外部中断上升沿允许寄存器（EGP0）、外部中断下降沿允许寄存器（EGN0） 724
23.4 中断处理的操作 725
23.4.1 可屏蔽中断请求的接受 725
23.4.2 不可屏蔽中断请求的接受 725
第 24 章 键中断功能 726
24.1 键中断的功能 726
24.2 键中断的结构 726
24.3 控制键中断的寄存器 728
24.3.1 键返回模式寄存器（KRM） 728
第 25 章 待机功能 729
25.1 待机功能 729
25.2 睡眠模式 730
25.2.1 睡眠模式的设置 730
25.2.2 睡眠模式的解除 733
25.3 深度睡眠模式 734
25.3.1 深度睡眠模式的设置 734
25.3.2 深度睡眠模式的解除 737
第 26 章 复位功能 738
26.1 复位时序 740
26.2 确认复位源的寄存器 743

26.2.1 复位控制标志寄存器（RESF）.....743
第 27 章 上电复位电路.....745
27.1 上电复位电路的功能.....745
27.2 上电复位电路的结构.....746
27.3 上电复位电路的运行.....747
第 28 章 电压检测电路.....750
28.1 电压检测电路的功能.....750
28.2 电压检测电路的结构.....751
28.3 控制电压检测电路的寄存器.....752
28.3.1 电压检测寄存器（LVIM）.....752
28.3.2 电压检测电平寄存器（LVIS）.....753
28.4 电压检测电路的运行.....756
28.4.1 用作复位模式时的设置.....756
28.4.2 用作中断模式时的设置.....757
28.4.3 用作中断&复位模式时的设置.....759
28.5 电压检测电路的注意事项.....764
第 29 章 安全功能.....766
29.1 安全功能的概要.....766
29.2 安全功能使用的寄存器.....767
29.3 安全功能的运行.....768
29.3.1 闪存 CRC 运算功能（高速 CRC）.....768
29.3.1.1 闪存 CRC 控制寄存器（CRC0CTL）.....768
29.3.1.2 闪存 CRC 运算结果寄存器（PGCRCL）.....769
29.3.2 CRC 运算功能（通用 CRC）.....771
29.3.2.1 CRC 输入寄存器（CRCIN）.....771
29.3.2.2 CRC 数据寄存器（CRCD）.....772
29.3.3 RAM 奇偶校验错误检测功能.....773
29.3.3.1 RAM 奇偶校验错误控制寄存器（RPECTL）.....773
29.3.4 SFR 保护功能.....775
29.3.4.1 SFR 保护控制寄存器（SFRGD）.....775
29.3.5 频率检测功能.....776
29.3.5.1 定时器输入选择寄存器 0（TIS0）.....776
29.3.6 A/D 测试功能.....777
29.3.6.1 A/D 测试寄存器（ADTES）.....779
29.3.6.2 模拟输入通道指定寄存器（ADS）.....779
29.3.7 产品唯一身份标识寄存器.....780

第 30 章 温度传感器和内部参考电压 781
30.1 温度传感器 781
30.2 温度传感器的寄存器.....782
30.2.1 温度传感器校准数据寄存器 TSN25.....782
30.2.2 温度传感器校准数据寄存器 TSN85.....782
30.3 温度传感器的使用说明 783
30.3.1 温度传感器的使用原理 783
30.3.2 温度传感器的使用方法 783
30.4 内部参考电压.....784
30.4.1 VDD 校准数据寄存器 VDDCDR.....784
30.4.2 内部参考电压的使用说明.....784
第 31 章 选项字节 785
31.1 选项字节的功能 785
31.1.1 用户选项字节（000C0H~000C2H） 785
31.1.2 闪存数据保护选项字节（000C3H，500004H） 786
31.2 用户选项字节的格式.....787
31.3 闪存数据保护选项字节的格式 793
第 32 章 FLASH 控制.....794
32.1 FLASH 控制功能描述 794
32.2 FLASH 存储器结构.....794
32.3 控制 FLASH 的寄存器 795
32.3.1 Flash 写保护寄存器(FLPROT) 795
32.3.2 FLASH 操作控制寄存器（FLOPMD1,FLOPMD2） 796
32.3.3 Flash 擦除控制寄存器(FLERMD).....797
32.3.4 Flash 状态寄存器(FLSTS).....798
32.3.5 Flash 全片擦除时间控制寄存器(FLCERCNT).....798
32.3.6 Flash 页擦除时间控制寄存器（FLSERCNT） 799
32.3.7 Flash 写入时间控制寄存器（FLPROCNT） 800
32.3.8 Flash 擦写保护控制寄存器（FLSECPR） 801
32.4 FLASH 操作方法 802
32.4.1 页擦除（sector erase） 802
32.4.2 全片擦除（chip erase） 803
32.4.3 编程（word program） 803
32.5 闪存读取.....804
32.6 FLASH 操作的注意事项 804
第 33 章 修改履历 805

第1章 CPU

1.1 概述

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

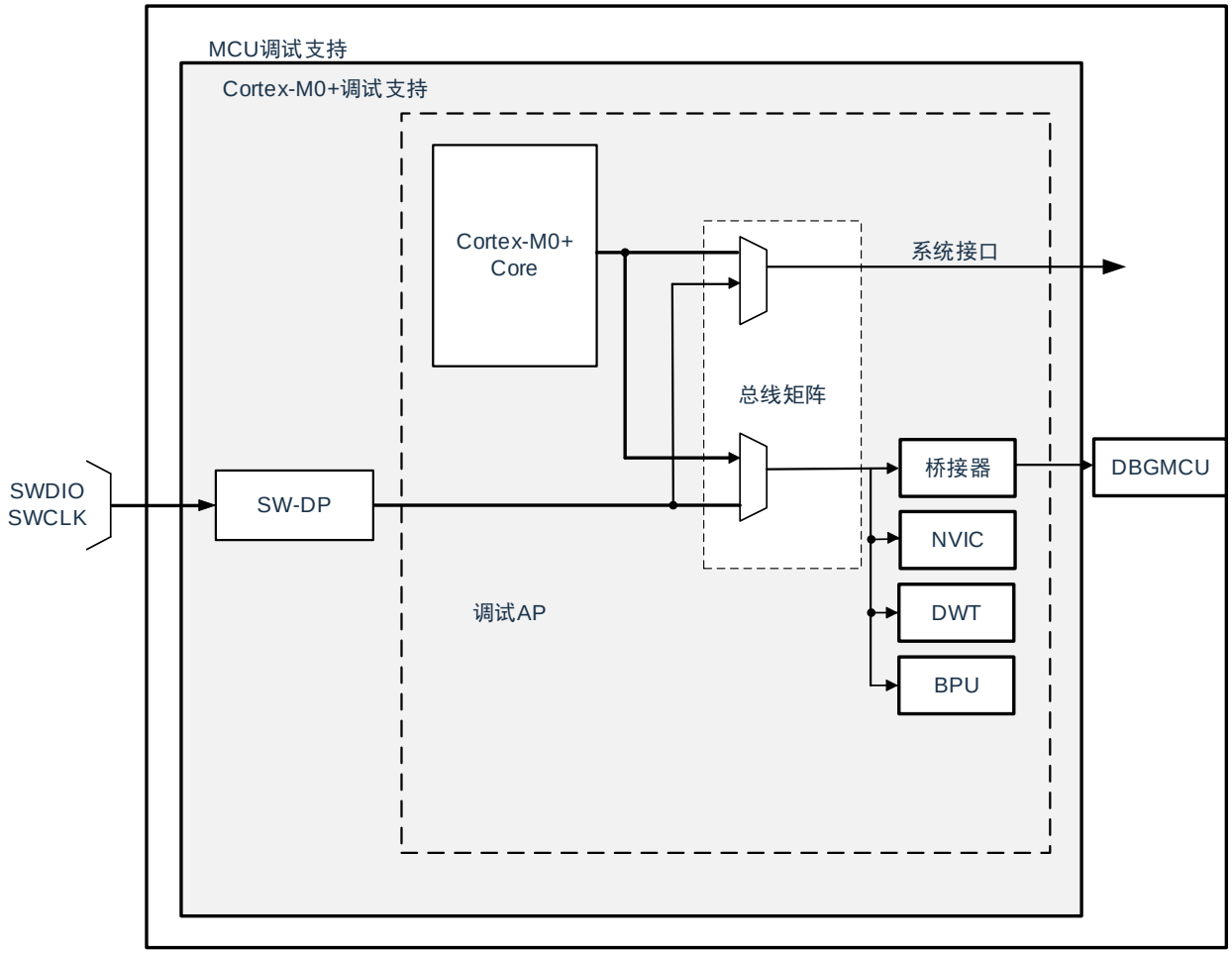
1.2 Cortex-M0+内核特性

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

1.3 调试特性

- 2线SWD调试接口
- 支持暂停、恢复和单步执行程序
- 访问处理器的内核寄存器和特殊功能寄存器
- 4个硬件断点(BPU)
- 无限个软件断点(BKPT指令)
- 2个数据观察点(DWT)
- 内核执行的时候访问存储器

图 1-1：Cortex-M0+的调试框图



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

1.4 SWD接口引脚

本产品的 2 个 GPIO 可用作 SWD 接口引脚，这些引脚在所有的封装里都存在。

表 1-1：SWD 调试端口引脚

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	PA14
SWDIO	串行数据输入 /输出	PA13

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。

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBGSTOPCR ＜31:16＞	-							SWDIS	-							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGSTOPCR ＜15:0＞	-														FRZEN1	FRZEN0

Bit24 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，PA13 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。PA13 可用作 GPIO

Bit1 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

Bit0 FRZEN1: 在调试器连接的状态下，并且 CPU 处于调试状态时（HALTED=1），通信系周边模块动作/停止 注2

0= 周边动作

1= 周边停止

注 1：本制品的定时器系周边模块包括：通用定时器单元 Timer8，定时器 A，15 位间隔定时器和实时时钟(RTC)。

注 2：本制品的通信系周边模块包括：通信串行通信单元，串行接口 SPI 和串行接口 IICA。

1.5 ARM参考文档

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

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

第2章 端口功能

2.1 端口通用功能

本产品的通用 I/O 端口（GPIO）有 PA00~PA15、PB00~PB15、PC00~PC12、PD02~PD09、PH00~PH04。不同的产品型号使用的通用 I/O 端口有差异，详情请参见各产品系列的数据手册。

每个 GPIO 端口有相关的控制和配置寄存器，可以由软件分别配置为多种模式以满足特定应用的需求。基本的工作模式有：

- 输入悬空
- 输入上拉
- 输入下拉
- 开漏输出（支持上拉）
- 推挽式输出
- 模拟通道

当端口被配置成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，此时读取端口回读寄存器 PREADxx 的结果为“0”。

PA00~PA07、PB00~PB02、PB10~PB15、PC00~PC06、PD04、PD05、PD08、PD09 端口可作为模拟通道使用，且复位后这些端口默认为模拟端口。要使用其数字功能需要配置 PMCxx 寄存器对应位为 0。

PA08~PA15、PB03~PB08、PC07~ PC12、PD02、PD06、PD07、PH00~PH04 只能作为数字端口使用。

作为数字 GPIO 使用时，除 PH00、PH03、PH04 以外的端口均支持输出(推挽或开漏)、输入、上拉、下拉功能。PH00 不支持下拉，PH03 和 PH04 不支持上拉和下拉。

为了避免贯通电流，PB03~PB08 复位后默认关闭输入功能，若想开启 GPIO 的输入功能需要配置 ISCLCD 寄存器，当 ISCLCD.ISCCAP 设置为 1 时，PB03 和 PB04 的施密特输入使能。当 ISCLCD.ISCVL 设置为 1 时，PB05~PB08 的施密特输入使能。详情请参考“第 20 章 LCD 控制器/驱动器”。

PA13、PA14 端口默认作为调试口使用并且复位后默认上拉，若要使用其 GPIO 功能需要配置 DBGSTOPCR.SWDIS 寄存器位为 1，将调试功能屏蔽。调试端口的对应关系和默认状态见表 2-1：SW 调试端口，调试功能具体的使用方法请参考“第 1 章 CPU”。

表 2-1：SW 调试端口

端口名称	SW 调试口	IO 类型	默认状态
PA13	SWDIO	I/O	内部上拉
PA14	SWCLK	I	内部上拉

芯片复位后普通的数字端口（PA08~PA12、PA15、PC07~ PC12、PD02、PD06、PD07、PH01~PH04）的默认状态为高阻输入（floating input），目的是防止芯片被异常复位时，对外部器件产生异常动作。但为了避免高阻输入而产生的漏电，用户要在芯片启动之后对端口进行相应的配置（配置成内部上拉/下拉输入或者输出）。

2.2 端口复用功能

各端口除了支持通用 GPIO 功能，还可复用为周边模块的功能端口。如模拟模块 ADC/DAC/CMP/AMP/LCD 的输入输出信号、数字功能模块（如 SPI，UART，I^2C，Timer 等）的输入输出信号。使用端口复用功能时请按照以下说明进行配置：

- 每个端口都可配置为外部中断，中断触发类型可以配置成上升沿触发，下降沿触发或者双沿触发。中断端口的选择需要配置 INTPnPCFG 寄存器（n=0~5）详情请参考 2.3.11 外部中断端口选择寄存器（INTPnPCFG），触发类型的选择需要配置 EPG0 和 EGN0 寄存器，详情请参考“第 23 章 中断功能”。
- PH00（RESETB）默认作为外部复位输入端口使用且默认上拉。要使用其 GPIO 功能需要配置 RSTM 寄存器，将 PH00 端口的复位功能屏蔽。PH00 端口不支持下拉功能。
- PH01（X1），PH02（X2/EXCLK）可配置为外部高速晶体震荡器端口，作为振荡器端口的配置和使用方法详见“第 4 章 时钟发生电路”。
- PH03（XT1），PH04（XT2/EXCLKS）可配置为外部低速晶体震荡器端口，作为振荡器端口的配置和使用方法详见“第 4 章 时钟发生电路”。这两个端口不支持上拉和下拉功能。
- 端口复用为 LCD 端口时（COM0~7，SEG0~41，VLCD1~4，CAPL，CAPH），需要将端口模式控制寄存器 PMCxx 对应位配置为 0，并且将 LCD 端口模式寄存器 SEG0~SEG3 对应的位配置为 1，详情请参考“第 20 章 LCD 控制器/驱动器”。
- 每个带有复用功能的端口对应一个端口复用功能配置寄存器 PxxCFG，通过设置 PxxCFG 可以分别为每个端口选择第 1 功能~第 7 功能，注意不要把同一个复用输入功能配置到不同的端口，PxxCFG 寄存器的操作说明请参考 2.3.10 端口复用功能配置寄存器（PxxCFG）。端口复用功能的详细设置方法请参考 2.5.2 端口复用功能的配置方法。

表 2-2：各端口复用功能列表

端口功能配置							
默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能
PA00	TXD2/SDO20	TI00	TO00	-	-	VC0OUT	-
PA01	RXD2/SDI20/ SDA20	-	TI01/TO01	-	SPI0_MOSI	-	PCLBUZ0
PA02	TXD1/SDO10	-	TI02/TO02	-	SPI0_MISO	VC1OUT	PCLBUZ1
PA03	RXD1/SDI10/ SDA10	-	TI03/TO03	TI00_GATE	SPI0_NSS	-	-
PA04	TXD1/SDO10	-	TI04/TO04		SPI0_NSS	-	-
PA05	SS10	-	TI05/TO05	TI06_GATE	SPI0_SCK	-	PCLBUZ0
PA06	SS20	-	TI06/TO06	TI07_GATE	SPI0_MISO	VC0OUT	-
PA07	SCLK10/SCL10	-	TI07/TO07	-	SPI0_MOSI	VC1OUT	PCLBUZ1
PA08	TXD0/SDO00	TI00	TO00	TI01_GATE	-	-	-
PA09	TXD0/SDO00	SCLA0	TI01/TO01	TI02_GATE	-	-	PCLBUZ0
PA10	RXD0/SDI00/ SDA00	SDAA0	TI02/TO02	TI03_GATE	-	-	PCLBUZ1
PA11	SS00	SCLA0	TI03/TO03	TI04_GATE	SPI0_MISO	VC0OUT	-
PA12	SS11	SDAA0	TI04/TO04	TI05_GATE	SPI0_MOSI	VC1OUT	-
PA13/ SWDIO	RXD0/SDI00/ SDA00	-	TI05/TO05	RTC1HZ	-	KR4	PCLBUZ0
PA14/ SWCLK	TXD0/SDO00	-	TI06/TO06	-	-	KR1	PCLBUZ1
PA15	RXD0/SDI00/ SDA00	-	TI07/TO07	-	SPI0_NSS	KR0	-
PB00	TXD1/SDO10	TI00	TO00	-	-	-	PCLBUZ0
PB01	SCLK20/SCL20	-	TI01/TO01	-	-	-	PCLBUZ1
PB02	TXD2/SDO20	-	TI02/TO02	-	-	TA_TI/TA_TO	
PB03	SCLK11/SCL11	-	TI03/TO03	TI04_GATE	SPI0_SCK		PCLBUZ0
PB04	SDI11/SDA11	-	TI04/TO04	TI05_GATE	SPI0_MISO	TA_TI/TA_TO	-
PB05	SDO11	-	TI05/TO05	TI06_GATE	SPI0_MOSI		-
PB06	TXD0/SDO00	SCLA0	TI06/TO06	-	-	TA_TI/TA_TO	-
PB07	RXD0/SDI00/ SDA00	SDAA0	TI07/ TO07	-	-	TA_TON	-
PB08	SCLK00/SCL00	-	-	-	-	-	-
PB10	TXD1/SDO10	SCLA0	TI02/TO02	-	SPI0_SCK	-	-
PB11	RXD1/SDI10/ SDA10	SDAA0	TI03/TO03	TI00_GATE	--	-	-
PB12	TXD1/SDO10	-	TI04/TO04	-	SPI0_NSS	VC0OUT	-
PB13	SCLK10/SCL10	SCLA0	TI05/TO05	TI01_GATE	SPI0_SCK	-	-
PB14	SS01	SDAA0	TI06/TO06	RTC1HZ	SPI0_MISO	-	-
PB15	RXD2/SDI20/ SDA20	-	TI07/TO07	TI02_GATE	SPI0_MOSI	-	-

端口功能配置							
默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能
PC00	SS21	TI00	TO00	TI03_GATE	-	-	-
PC01	SCLK21/SCL21	-	TI01/TO01	-	-	TA_TI/ TA_TO	-
PC02	SDI21/SDA21	-	TI02/TO02	-	SPI0_MISO	TA_TON	-
PC03	SDO21	-	TI03/TO03	-	SPI0_MOSI	TA_TI/ TA_TO	-
PC04	TXD1/SDO10	-	TI04/TO04	-	-	-	PCLBUZ1
PC05	RXD1/SDI10/ SDA10	-	TI05/TO05	-	-	-	-
PC06	SCLK01/SCL01	-	TI06/TO06	-	-	-	-
PC07	SDI01/SDA01	-	TI07/TO07	-	-	KR7	-
PC08	SDO01	TI00	TO00	-	-	KR6	-
PC09	SCLK00/SCL00	-	TI01/TO01	-	-	KR5	-
PC10	TXD2/SDO20	-	TI02/TO02	-	-	-	-
PC11	RXD2/SDI20/ SDA20	-	TI03/TO03	-	-	-	-
PC12	TXD2/SDO20	-	TI04/TO04	-	-	-	-
PD02	SCLK20/SCL20	-	TI02/TO02	-	-	-	-
PD04	SCLK10/SCL10		-	-	-	-	-
PD05	SCLK20/SCL20		-	-	-	-	-
PD06	-	SCLA0	-	-	-	KR3	-
PD07	SCLK00/SCL00	SDAA0	-	-	-	KR2	-
PH00/ RESETB	-		-	-	-	-	-
PH01	TXD1/SDO10	SDAA0	-	-	-	-	-
PH02	RXD1/SDI10/ SDA10	SCLA0	TI01/ TO01	-	-	-	-
PH03	-	-	-	-	-	-	-
PH04	-	-	-	-	-	-	-

2.3 控制端口功能的寄存器

通过以下寄存器控制端口功能。

- 端口输出控制寄存器（PMxx）
- 端口寄存器（Pxx）
- 上拉电阻选择寄存器（PUxx）
- 下拉电阻选择寄存器（PDxx）
- 端口输出模式寄存器（POMx）
- 端口模式控制寄存器（PMCxx）
- 端口置位控制寄存器（PSETxx）
- 端口清零控制寄存器（PCLRxx）
- 端口状态回读寄存器（PREADxx）
- 端口复用功能配置寄存器（PxxCFG，详见 2.3.10）
- 外部中断端口选择寄存器（INTPnPCFG，n=0~5）

注意：由于不同的产品型号使用的端口数量及端口配置有差异，因此公开的端口控制寄存器因产品而不同。各产品公开的端口控制寄存器列表，请参照表 2-3。

表 2-3：各产品分配的端口控制寄存器

端口		寄存器名称								
		PMxx 寄存器	Pxx 寄存器	PSETxx 寄存器	PCLRxx 寄存器	PUxx 寄存器	PDxx 寄存器	POMxx 寄存器	PREADxx 寄存器	PMCxx 寄存器
端口 A	00	PMA0	PA0	PSETA0	PCLRA0	PUA0	PDA0	POMA0	PREADA0	PMCA0
	01	PMA1	PA1	PSETA1	PCLRA1	PUA1	PDA1	POMA1	PREADA1	PMCA1
	02	PMA2	PA2	PSETA2	PCLRA2	PUA2	PDA2	POMA2	PREADA2	PMCA2
	03	PMA3	PA3	PSETA3	PCLRA3	PUA3	PDA3	POMA3	PREADA3	PMCA3
	04	PMA4	PA4	PSETA4	PCLRA4	PUA4	PDA4	POMA4	PREADA4	PMCA4
	05	PMA5	PA5	PSETA5	PCLRA5	PUA5	PDA5	POMA5	PREADA5	PMCA5
	06	PMA6	PA6	PSETA6	PCLRA6	PUA6	PDA6	POMA6	PREADA6	PMCA6
	07	PMA7	PA7	PSETA7	PCLRA7	PUA7	PDA7	POMA7	PREADA7	PMCA7
	08	PMA8	PA8	PSETA8	PCLRA8	PUA8	PDA8	POMA8	PREADA8	-
	09	PMA9	PA9	PSETA9	PCLRA9	PUA9	PDA9	POMA9	PREADA9	-
	10	PMA10	PA10	PSETA10	PCLRA10	PUA10	PDA10	POMA10	PREADA10	-
	11	PMA11	PA11	PSETA11	PCLRA11	PUA11	PDA11	POMA11	PREADA11	-
	12	PMA12	PA12	PSETA12	PCLRA12	PUA12	PDA12	POMA12	PREADA12	-
	13	PMA13	PA13	PSETA13	PCLRA13	PUA13	PDA13	POMA13	PREADA13	-
	14	PMA14	PA14	PSETA14	PCLRA14	PUA14	PDA14	POMA14	PREADA14	-
	15	PMA15	PA15	PSETA15	PCLRA15	PUA15	PDA15	POMA15	PREADA15	-

端口		位名								
		PMxx 寄存器	Pxx 寄存器	PSETxx 寄存器	PCLRxx 寄存器	PUxx 寄存器	PDxx 寄存器	POMxx 寄存器	PREADxx 寄存器	PMCxx 寄存器
端口 B	00	PMB0	PB0	PSETB0	PCLRB0	PUB0	PDB0	POMB0	PREADB0	PMCB0
	01	PMB1	PB1	PSETB1	PCLRB1	PUB1	PDB1	POMB1	PREADB1	PMCB1
	02	PMB2	PB2	PSETB2	PCLRB2	PUB2	PDB2	POMB2	PREADB2	PMCB2
	03	PMB3	PB3	PSETB3	PCLRB3	PUB3	PDB3	POMB3	PREADB3	-
	04	PMB4	PB4	PSETB4	PCLRB4	PUB4	PDB4	POMB4	PREADB4	-
	05	PMB5	PB5	PSETB5	PCLRB5	PUB5	PDB5	POMB5	PREADB5	-
	06	PMB6	PB6	PSETB6	PCLRB6	PUB6	PDB6	POMB6	PREADB6	-
	07	PMB7	PB7	PSETB7	PCLRB7	PUB7	PDB7	POMB7	PREADB7	-
	08	PMB8	PB8	PSETB8	PCLRB8	PUB8	PDB8	POMB8	PREADB8	-
	10	PMB10	PB10	PSETB10	PCLRB10	PUB10	PDB10	POMB10	PREADB10	PMCB10
	11	PMB11	PB11	PSETB11	PCLRB11	PUB11	PDB11	POMB11	PREADB11	PMCB11
	12	PMB12	PB12	PSETB12	PCLRB12	PUB12	PDB12	POMB12	PREADB12	PMCB12
	13	PMB13	PB13	PSETB13	PCLRB13	PUB13	PDB13	POMB13	PREADB13	PMCB13
	14	PMB14	PB14	PSETB14	PCLRB14	PUB14	PDB14	POMB14	PREADB14	PMCB14
	15	PMB15	PB15	PSETB15	PCLRB15	PUB15	PDB15	POMB15	PREADB15	PMCB15
端口 C	00	PMC0	PC0	PSETC0	PCLRC0	PUC0	PDC0	POMC0	PREADC0	PMCC0
	01	PMC1	PC1	PSETC1	PCLRC1	PUC1	PDC1	POMC1	PREADC1	PMCC1
	02	PMC2	PC2	PSETC2	PCLRC2	PUC2	PDC2	POMC2	PREADC2	PMCC2
	03	PMC3	PC3	PSETC3	PCLRC3	PUC3	PDC3	POMC3	PREADC3	PMCC3
	04	PMC4	PC4	PSETC4	PCLRC4	PUC4	PDC4	POMC4	PREADC4	PMCC4
	05	PMC5	PC5	PSETC5	PCLRC5	PUC5	PDC5	POMC5	PREADC5	PMCC5
	06	PMC6	PC6	PSETC6	PCLRC6	PUC6	PDC6	POMC6	PREADC6	PMCC6
	07	PMC7	PC7	PSETC7	PCLRC7	PUC7	PDC7	POMC7	PREADC7	-
	08	PMC8	PC8	PSETC8	PCLRC8	PUC8	PDC8	POMC8	PREADC8	-
	09	PMC9	PC9	PSETC9	PCLRC9	PUC9	PDC9	POMC9	PREADC9	-
	10	PMC10	PC10	PSETC10	PCLRC10	PUC10	PDC10	POMC10	PREADC10	-
	11	PMC11	PC11	PSETC11	PCLRC11	PUC11	PDC11	POMC11	PREADC11	-
	12	PMC12	PC12	PSETC12	PCLRC12	PUC12	PDC12	POMC12	PREADC12	-
端口 D	02	PMD2	PD2	PSETD2	PCLRD2	PUD2	PDD2	POMD2	PREADD2	-
	04	PMD4	PD4	PSETD4	PCLRD4	PUD4	PDD4	POMD4	PREADD4	PMCD4
	05	PMD5	PD5	PSETD5	PCLRD5	PUD5	PDD5	POMD5	PREADD5	PMCD5
	06	PMD6	PD6	PSETD6	PCLRD6	PUD6	PDD6	POMD6	PREADD6	-
	07	PMD7	PD7	PSETD7	PCLRD7	PUD7	PDD7	POMD7	PREADD7	-
RESETB/PH00		PMH0	PH0	PSETH0	PCLRH0	PUH0	-	POMH0	PREADH0	-
X1/PH01		PMH1	PH1	PSETH1	PCLRH1	PUH1	PDH1	POMH1	PREADH1	-
X2/PH02		PMH2	PH2	PSETH2	PCLRH2	PUH2	PDH2	POMH2	PREADH2	-
XT1/PH03		PMH3	PH3	PSETH3	PCLRH3	-	-	POMH3	PREADH3	-
XT2/PH04		PMH4	PH4	PSETH4	PCLRH4	-	-	POMH4	PREADH4	-

2.3.1 端口输出控制寄存器（PMxx）

当端口作为数字通道使用时，这是以位为单位设定其输出是否使能的寄存器。在产生复位信号后，各端口默认为输入状态。当将端口用作复用功能的端口时，必须参照“2.5 使用复用功能时的寄存器设定”进行设定。

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

图 2-1：端口输出控制寄存器的格式

复位后： FFFFH

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PMA	PMA 15	PMA 14	PMA 13	PMA 12	PMA 11	PMA 10	PMA9	PMA8	PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0	0x010
PMB	PMB 15	PMB 14	PMB 13	PMB 12	PMB 11	PMB 10	PMB9	PMB8	PMB7	PMB6	PMB5	PMB4	PMB3	PMB2	PMB1	PMB0	0x012
PMC	1	PMC 14	PMC 13	PMC 12	PMC 11	PMC 10	PMC9	PMC8	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	0x014
PMD	1						PMD9	PMD8	PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0x016
PMH	1											PMH4	PMH3	PMH2	PMH1	PMH0	0x01E
R/W	R/W																

PMmn: Pmn 端口的输出模式的选择（m=A,B,C,D,H，n=0~15）

0= 输出模式（用作输出端口（输出缓冲器 ON））

1= 输入模式（用作输入端口（输出缓冲器 OFF））

注意：必须给未分配的位设定初始值。

2.3.2 端口寄存器（Pxx）

该寄存器用于设定各端口输出锁存器的值。在 PMxx 为 0 时读此寄存器可以得到对应端口的输出锁存器的值，而在 PMxx 为 1 时读此寄存器可以得到对应端口的端口电平。

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

图 2-2：端口寄存器的格式

复位后： 0000H（输出锁存器）

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PA	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	0x000
PB	PB15	PB14	PB13	PB12	PB11	PB10	0	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0x002
PC	0			PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0x004
PD	0						PD9	PD8	PD7	PD6	PD5	PD4	0	PD2	0	0	0x006
PH	0											PH4	PH3	PH2	PH1	PH0	0x00E
R/W	R/W																

Pmn: m=A,B,C,D,H，n=0~15

	输出数据的控制（输出模式）	输入数据的读取（输入模式）
0=	端口输出“0”。	端口输入低电平。
1=	端口输出“1”。	端口输入高电平。

注意：必须给未分配的位设定初始值。

2.3.3 端口置位控制寄存器（PSETxx）

这是以位为单位来置位端口输出锁存器的寄存器。产生复位信号后，这些寄存器的值变为“0000H”。

寄存器地址=基址+偏址；端口置位控制寄存器的基址为 0x40040000，偏址见下图。

图 2-3：端口置位控制寄存器的格式

复位后： 0000H

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PSETA	PSETA15	PSETA14	PSETA13	PSETA12	PSETA11	PSETA10	PSETA9	PSETA8	PSETA7	PSETA6	PSETA5	PSETA4	PSETA3	PSETA2	PSETA1	PSETA0	0x060
PSETB	PSETB15	PSETB14	PSETB13	PSETB12	PSETB11	PSETB10	PSETB9	PSETB8	PSETB7	PSETB6	PSETB5	PSETB4	PSETB3	PSETB2	PSETB1	PSETB0	0x062
PSETC	0	PSETC14	PSETC13	PSETC12	PSETC11	PSETC10	PSETC9	PSETC8	PSETC7	PSETC6	PSETC5	PSETC4	PSETC3	PSETC2	PSETC1	PSETC0	0x064
PSETD	0						PSETD9	PSETD8	PSETD7	PSETD6	PSETD5	PSETD4	PSETD3	PSETD2	PSETD1	PSETD0	0x066
PSETH	0											PSETH4	PSETH3	PSETH2	PSETH1	PSETH0	0x06E
R/W	W																

PSETmn: Pmn 端口的置位控制（m=A,B,C,D,H，n=0~15）

0= 无操作

1= 对应的 Pmn 置 1

注意：必须给未分配的位设定初始值。

2.3.4 端口清零控制寄存器（PCLRxx）

这是以位为单位来置位端口输出锁存器的寄存器。产生复位信号后，这些寄存器的值变为“0000H”。

寄存器地址=基址+偏址；端口清零控制寄存器的基址为 0x40040000，偏址见下图。

图 2-4：端口清零控制寄存器的格式

复位后： 0000H

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PCLRA	PCLRA15	PCLRA14	PCLRA13	PCLRA12	PCLRA11	PCLRA10	PCLRA9	PCLRA8	PCLRA7	PCLRA6	PCLRA5	PCLRA4	PCLRA3	PCLRA2	PCLRA1	PCLRA0	0x070
PCLRB	PCLRB15	PCLRB14	PCLRB13	PCLRB12	PCLRB11	PCLRB10	PCLRB9	PCLRB8	PCLRB7	PCLRB6	PCLRB5	PCLRB4	PCLRB3	PCLRB2	PCLRB1	PCLRB0	0x072
PCLRC	0	PCLRC14	PCLRC13	PCLRC12	PCLRC11	PCLRC10	PCLRC9	PCLRC8	PCLRC7	PCLRC6	PCLRC5	PCLRC4	PCLRC3	PCLRC2	PCLRC1	PCLRC0	0x074
PCLRD	0						PCLRD9	PCLRD8	PCLRD7	PCLRD6	PCLRD5	PCLRD4	PCLRD3	PCLRD2	PCLRD1	PCLRD0	0x076
PCLRH	0											PCLRH4	PCLRH3	PCLRH2	PCLRH1	PCLRH0	0x07E
R/W	W																

PCLRmn: Pmn 端口的清零控制（m=A,B,C,D,H，n=0~15）

0= 无操作

1= 对应的 Pmn 清零

注意：必须给未分配的位设定初始值。

2.3.5 上拉电阻选择寄存器（PUxx）

内部上拉电阻的选择寄存器。通过设定该寄存器可以将处于输入模式（PMmn=1）或者 N 沟道漏极开路输出模式的端口，以位为单位使用内部上拉电阻进行上拉。对于设定为输出模式的端口，与上拉电阻选择寄存器的设定无关，不连接内部上拉电阻。当用作复用功能的输出端口或者设定为模拟功能时也同样不能上拉。

在产生复位信号后，PA13、PA14、PH00 这三个端口的上拉功能默认打开（PUA13，PUA14、PUH0 复位值为“1”），其他端口的上拉功能默认不打开。

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

图 2-5：上拉电阻选择寄存器的格式

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址	复位后
PUA	PUA 15	PUA 14	PUA 13	PUA 12	PUA 11	PUA 10	PUA 9	PUA 8	PUA 7	PUA 6	PUA 5	PUA 4	PUA 3	PUA 2	PUA 1	PUA 0	0x020	0x02E
PUB	PUB 15	PUB 14	PUB 13	PUB 12	PUB 11	PUB 10	PUB 9	PUB 8	PUB 7	PUB 6	PUB 5	PUB 4	PUB 3	PUB 2	PUB 1	PUB 0	0x022	0000H
PUC	0	PUC 14	PUC 13	PUC 12	PUC 11	PUC 10	PUC 9	PUC 8	PUC 7	PUC 6	PUC 5	PUC 4	PUC 3	PUC 2	PUC 1	PUC 0	0x024	0000H
PUD	0						PUD 9	PUD 8	PUD 7	PUD 6	PUD 5	PUD 4	PUD 3	PUD 2	PUD 1	PUD 0	0x026	0000H
PUH	0													PUH 2	PUH 1	PUH 0	0x02E	0001H
R/W	R/W																	

PUmn: Pmn 端口的内部上拉电阻的选择（m=A,B,C,D,H，n=0~15）

0= 不连接内部上拉电阻。

1= 连接内部上拉电阻。

注意：必须给未分配的位设定初始值。

2.3.6 下拉电阻选择寄存器（PDxx）

内部下拉电阻的选择寄存器。通过设定该寄存器可以将处于输入模式（PMmn=1）的端口，以位为单位使用内部下拉电阻进行下拉。对于设定为输出模式的端口，与下拉电阻选择寄存器的设定无关，不连接内部下拉电阻。当用作复用功能的输出端口或者设定为模拟功能时也同样不能下拉。

在产生复位信号后，端口的下拉功能默认关闭，寄存器的复位值为“0000H”。

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

图 2-6：下拉电阻选择寄存器的格式

复位后： 0000H

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PDA	PDA 15	PDA 14	PDA 13	PDA 12	PDA 11	PDA 10	PDA9	PDA8	PDA7	PDA6	PDA5	PDA4	PDA3	PDA2	PDA1	PDA0	0x030
PDB	PDB 15	PDB 14	PDB 13	PDB 12	PDB 11	PDB 10	PDB9	PDB8	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0	0x032
PDC	0	PDC 14	PDC 13	PDC 12	PDC 11	PDC 10	PDC9	PDC8	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0x034
PDD	0						PDD9	PDD8	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0	0x036
PDH	0													PDH2	PDH1	0	0x03E
R/W	R/W																

PDmn: Pmn 端口的内部下拉电阻的选择（m=A，B，C，D，H，n=0~15）

0= 不连接内部下拉电阻。

1= 连接内部下拉电阻。

注意：必须给未分配的位设定初始值。

2.3.7 端口输出模式寄存器（POMxx）

这是以位为单位设定输出模式的寄存器。在和不同电位的外部设备进行串行通信以及和同电位的外部设备进行 I^2C 通信时，选择 N 沟道漏极开路输出模式。

在产生复位信号后，这些寄存器的值变为“0000H”。

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

图 2-7：端口输出模式寄存器的格式

复位后： 0000H

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
POMA	POMA15	POMA14	POMA13	POMA12	POMA11	POMA10	POMA9	POMA8	POMA7	POMA6	POMA5	POMA4	POMA3	POMA2	POMA1	POMA0	0x040
POMB	POMB15	POMB14	POMB13	POMB12	POMB11	POMB10	POMB9	POMB8	POMB7	POMB6	POMB5	POMB4	POMB3	POMB2	POMB1	POMB0	0x042
POMC	0	POMC14	POMC13	POMC12	POMC11	POMC10	POMC9	POMC8	POMC7	POMC6	POMC5	POMC4	POMC3	POMC2	POMC1	POMC0	0x044
POMD	0						POMD9	POMD8	POMD7	POMD6	POMD5	POMD4	POMD3	POMD2	POMD1	POMD0	0x046
POMH	0											POMH4	POMH3	POMH2	POMH1	POMH0	0x04E
R/W	R/W																

POMmn: Pmn 端口的输出模式的选择（m=A,B,C,D,H，n=0~15）

0= 通常的输出模式

1= N 沟道漏极开路输出模式

注意：必须给未分配的位设定初始值。

2.3.8 端口模式控制寄存器（PMCxx）

PMC 寄存器以位为单位设定端口作为模拟通道使用或者其他功能使用，其他功能包括数字端口功能和 LCD 端口功能。

在产生复位信号后，PA00~PA07、PB00~PB03、PB10~PB15、PC00~PC06、PD04、PD05、PD08、PD09 默认作为模拟通道使用，其他端口默认作为数字通道使用。没有 PMC 寄存器的端口只有数字功能，不能作为模拟通道使用。

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

图 2-8：端口模式控制寄存器的格式

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址	复位后
PMCA	0								PMCA7	PMCA6	PMCA5	PMCA4	PMCA3	PMCA2	PMCA1	PMCA0	0x050	00FFH
PMCB	PMCB15	PMCB14	PMCB13	PMCB12	PMCB11	PMCB10	0							PMCB2	PMCB1	PMCB0	0x052	FC07H
PMCC	0									PMCC6	PMCC5	PMCC4	PMCC3	PMCC2	PMCC1	PMCC0	0x054	407FH
PMCD	0										PMCD5	PMCD4	0				0x056	0333H
R/W	R/W																	

PMCmn: Pmn 端口作为模拟通道的选择（m=A，B，C，D，n=0~15）

0= 作为模拟以外的端口，例如数字端口或 LCD 端口

1= 模拟通道

注意：必须给未分配的位设定初始值。

2.3.9 端口回读寄存器（PREADxx）

这是一个只读寄存器，在端口作为数字端口使用时，可以通过读此寄存器得到端口电平。

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

图 2-9：端口回读寄存器的格式

复位后： xxxxH

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	偏址
PREADA	PREADA15	PREADA14	PREADA13	PREADA12	PREADA11	PREADA10	PREADA9	PREADA8	PREADA7	PREADA6	PREADA5	PREADA4	PREADA3	PREADA2	PREADA1	PREADA0	0x080
PREADB	PREADB15	PREADB14	PREADB13	PREADB12	PREADB11	PREADB10	PREADB9	PREADB8	PREADB7	PREADB6	PREADB5	PREADB4	PREADB3	PREADB2	PREADB1	PREADB0	0x082
PREADC	0	PREADC14	PREADC13	PREADC12	PREADC11	PREADC10	PREADC9	PREADC8	PREADC7	PREADC6	PREADC5	PREADC4	PREADC3	PREADC2	PREADC1	PREADC0	0x084
PREADD	0						PREADD9	PREADD8	PREADD7	PREADD6	PREADD5	PREADD4	PREADD3	PREADD2	PREADD1	PREADD0	0x086
PREADH	0											PREADH4	PREADH3	PREADH2	PREADH1	PREADH0	0x08E
R/W	R																

备注：复位后寄存器的读出值取决于各端口状态。

PREADmn: m=A,B,C,D,H，n=0~15

输出模式/输入模式

0= 端口为低电平。

1= 端口为高电平。

2.3.10 端口复用功能配置寄存器（PxxCFG）

端口复用功能配置寄存器可实现将外围模块的数字输入、输出功能重定向到不同端口。每个端口对应一个端口复用功能配置寄存器。除 PH00，PA13，PA14 以外的端口复位后默认的功能为 GPIO 功能。为了避免贯通电流，PB03~PB08 复位后默认关闭输入功能，若想开启这几个端口的数字输入功能需要配置 ISCLCD 寄存器，当 ISCLCD.ISCCAP 设置为 1 时，PB03 和 PB04 的数字输入使能。当 ISCLCD.ISCVL 设置为 1 时，PB05~PB08 的数字输入使能。端口复用功能配置寄存器的复位值为“00H”。

寄存器地址=基址+偏址；PxxCFG 寄存器的基址为 0x40040400，偏址见表 2-5：端口复用功能配置寄存器列表。

图 2-10：端口复用功能配置寄存器的格式

复位后： 00H

符号	7	6	5	4	3	2	1	0
PxxCFG	0					PxxCFG[2:0]		
R/W	R/W							

表 2-4：端口复用功能的选择方法

寄存器名称	寄存器配置	端口功能
PxxCFG[2:0]	3'b000(default)	默认功能
	3'b001	第 1 功能
	3'b010	第 2 功能
	3'b011	第 3 功能
	3'b100	第 4 功能
	3'b101	第 5 功能
	3'b110	第 6 功能
	3'b111	第 7 功能

每个端口对应一个 PxxCFG 寄存器，通过设置 PxxCFG 可以分别为每个端口选择第 1 功能~第 7 功能，详见表 2-2：各端口复用功能列表。注意不要把同一个复用输入功能配置到不同的端口。

表 2-5：端口复用功能配置寄存器列表

基地址	偏移地址	寄存器名称	读写属性	位宽	复位值
0x40040400	0x000	PA00CFG[2:0]	R/W	8	00H
	0x001	PA01CFG[2:0]	R/W	8	00H
	0x002	PA02CFG[2:0]	R/W	8	00H
	0x003	PA03CFG[2:0]	R/W	8	00H
	0x004	PA04CFG[2:0]	R/W	8	00H
	0x005	PA05CFG[2:0]	R/W	8	00H
	0x006	PA06CFG[2:0]	R/W	8	00H
	0x007	PA07CFG[2:0]	R/W	8	00H
	0x008	PA08CFG[2:0]	R/W	8	00H
	0x009	PA09CFG[2:0]	R/W	8	00H
	0x00a	PA10CFG[2:0]	R/W	8	00H
	0x00b	PA11CFG[2:0]	R/W	8	00H
	0x00c	PA12CFG[2:0]	R/W	8	00H
	0x00d	PA13CFG[2:0]	R/W	8	00H
	0x00e	PA14CFG[2:0]	R/W	8	00H
	0x00f	PA15CFG[2:0]	R/W	8	00H
	0x010	PB00CFG[2:0]	R/W	8	00H
	0x011	PB01CFG[2:0]	R/W	8	00H
	0x012	PB02CFG[2:0]	R/W	8	00H
	0x013	PB03CFG[2:0]	R/W	8	00H
	0x014	PB04CFG[2:0]	R/W	8	00H
	0x015	PB05CFG[2:0]	R/W	8	00H
	0x016	PB06CFG[2:0]	R/W	8	00H
	0x017	PB07CFG[2:0]	R/W	8	00H
	0x018	PB08CFG[2:0]	R/W	8	00H
	0x01a	PB10CFG[2:0]	R/W	8	00H
	0x01b	PB11CFG[2:0]	R/W	8	00H
	0x01c	PB12CFG[2:0]	R/W	8	00H
	0x01d	PB13CFG[2:0]	R/W	8	00H
	0x01e	PB14CFG[2:0]	R/W	8	00H
	0x01f	PB15CFG[2:0]	R/W	8	00H

基地址	偏移地址	寄存器名称	读写属性	位宽	复位值
0x40040400	0x020	PC00CFG[2:0]	R/W	8	00H
	0x021	PC01CFG[2:0]	R/W	8	00H
	0x022	PC02CFG[2:0]	R/W	8	00H
	0x023	PC03CFG[2:0]	R/W	8	00H
	0x024	PC04CFG[2:0]	R/W	8	00H
	0x025	PC05CFG[2:0]	R/W	8	00H
	0x026	PC06CFG[2:0]	R/W	8	00H
	0x027	PC07CFG[2:0]	R/W	8	00H
	0x028	PC08CFG[2:0]	R/W	8	00H
	0x029	PC09CFG[2:0]	R/W	8	00H
	0x02a	PC10CFG[2:0]	R/W	8	00H
	0x02b	PC11CFG[2:0]	R/W	8	00H
	0x02c	PC12CFG[2:0]	R/W	8	00H
	0x032	PD02CFG[2:0]	R/W	8	00H
	0x034	PD04CFG[2:0]	R/W	8	00H
	0x035	PD05CFG[2:0]	R/W	8	00H
	0x036	PD06CFG[2:0]	R/W	8	00H
	0x037	PD07CFG[2:0]	R/W	8	00H
	0x038	PD08CFG[2:0]	R/W	8	00H
	0x039	PD09CFG[2:0]	R/W	8	00H
	0x080	PH00CFG[2:0]	R/W	8	00H
	0x081	PH01CFG[2:0]	R/W	8	00H
	0x082	PH02CFG[2:0]	R/W	8	00H
	0x083	PH03CFG[2:0]	R/W	8	00H
	0x084	PH04CFG[2:0]	R/W	8	00H

2.3.11 外部中断端口选择寄存器（INTPnPCFG）

本产品支持 6 路外部中断 INTP0~5，每路外部中断都可以重定向到多个端口上。通过配置外部中断端口选择寄存器（INTPnPCFG），可实现将 INTPn 的输入功能重定向到不同的端口。外部中断端口选择寄存器的复位值为“00H”。（n=0~5）

寄存器地址=基址+偏址；INTPnPCFG 寄存器的基址为 0x40040400，偏址见下表。

表 2-6：外部中断端口选择寄存器列表

寄存器名称	偏移地址	功能	读写属性	复位值
INTP0PCFG	0x0E0	INTP0 重定向端口选择	R/W	00H
INTP1PCFG	0x0E1	INTP1 重定向端口选择	R/W	00H
INTP2PCFG	0x0E2	INTP2 重定向端口选择	R/W	00H
INTP3PCFG	0x0E3	INTP3 重定向端口选择	R/W	00H
INTP4PCFG	0x0E4	INTP4 重定向端口选择	R/W	00H
INTP5PCFG	0x0E5	INTP5 重定向端口选择	R/W	00H

图 2-11：外部中断端口选择寄存器的格式

地址：　见上图　　复位后：　00H

符号	7	6	5	4	3	2	1	0
INTPnPCFG	0				INTPnPCFG[3:0]			
R/W	R/W							

序号	寄存器名称	寄存器配置	INTP0 端口选择
0	INTP0PCFG	4'h0(default)	PA00
1		4'h1	PA01
2		4'h2	PA02
3		4'h3	PA03
4		4'h4	PA04
5		4'h5	PA05
6		4'h6	PA06
7		4'h7	PA07
8		4'h8	PA08
9		4'h9	PA09
10		4'hA	PA10
11		4'hB	PA11
12		4'hC	PA12
13		4'hD	PA13
14		4'hE	PA14
15		4'hF	PA15

序号	寄存器名称	寄存器配置	INTP1 端口选择
0	INTP1PCFG	4'h0(default)	PB00
1		4'h1	PB01
2		4'h2	PB02
3		4'h3	PB03
4		4'h4	PB04
5		4'h5	PB05
6		4'h6	PB06
7		4'h7	PB07

序号	寄存器名称	寄存器配置	INTP2 端口选择
0	INTP2PCFG	4'h0(default)	PC00
1		4'h1	PC01
2		4'h2	PC02
3		4'h3	PC03
4		4'h4	PC04
5		4'h5	PC05
6		4'h6	PC06
7		4'h7	PC07
8		4'h8	PC08
9		4'h9	PC09
10		4'hA	PC10
11		4'hB	PC11
12		4'hC	PC12
13		4'hD	PC13
14		4'hE	PH03
15		4'hF	PH04

序号	寄存器名称	寄存器配置	INTP3 端口选择
0	INTP3PCFG	4'h0(default)	PD00
1		4'h1	PD01
2		4'h2	PD02
3		4'h3	PB08
4		4'h4	PH00
5		4'h5	PH01
6		4'h6	PH02

序号	寄存器名称	寄存器配置	INTP4 端口选择
0	INTP4PCFG	4'h0(default)	PC14
1		4'h1	PD04
2		4'h2	PD05
3		4'h3	PD06
4		4'h4	PD07
5		4'h5	PD08
6		4'h6	PD09

序号	寄存器名称	寄存器配置	INTP5 端口选择
0	INTP5PCFG	4'h0(default)	PD03
1		4'h1	PB09
2		4'h2	PB10
3		4'h3	PB11
4		4'h4	PB12
5		4'h5	PB13
6		4'h6	PB14
7		4'h7	PB15

2.3.12 外部复位端口屏蔽寄存器（RSTM）

PH00（RESETB）默认作为外部复位输入端口使用，低电平时发生系统复位。需要使用 PH00 的 GPIO 功能时，需要先配置寄存器 RSTM，将其外部复位功能屏蔽。

图 2-12：外部复位端口屏蔽寄存器的格式

基地址：0x40020400

基地址： 0x40020400 复位后： 00H 偏址： 0x00C

符号	7	6	5	4	3	2	1	0
RSTM	0							RSTM
R/W	R/W							

Bit0 RSTM: PH00(RESETB)端口的外部复位功能屏蔽

0= PH00 作为外部复位端口

1= PH00 作为 GPIO 端口

2.4 未使用端口的处理

各未使用端口的处理如表 2-7 所示。

表 2-7：各未使用端口的处理

端口名	输入/输出	未使用时的推荐连接方法
PA00~PA15	输入/输出	输入：单独通过电阻连接 EV_{DD} 或者 EV_{SS}。 输出：置为开路。
PB00~PB15		
PC00~PC12		
PD02~PD09		
PH01~PH04		单独通过电阻连接 V_{DD} 或者 V_{SS}。
RESETB/PH00	输入/输出	直接或者通过电阻连接 V_{DD}。

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

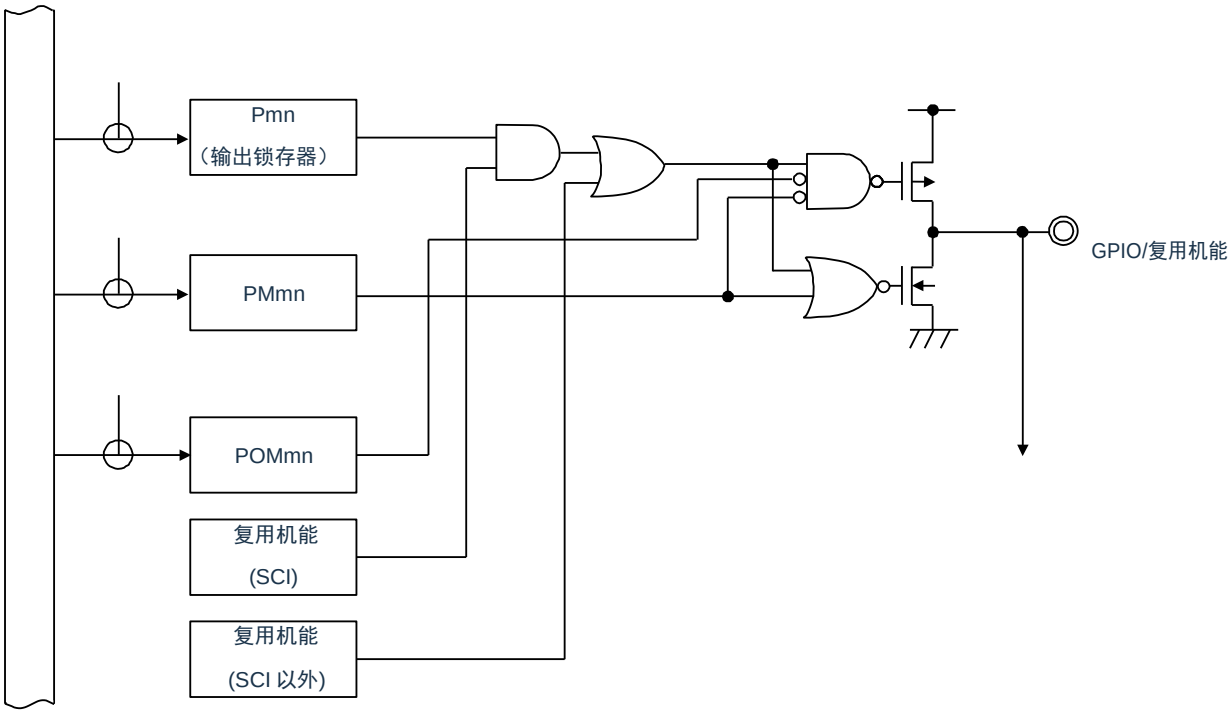
2.5 使用复用功能时的寄存器设定

2.5.1 使用复用功能时的基本思想

首先，对于具有模拟功能的端口，通过端口模式控制寄存器（PMCxx）设定端口是用作模拟功能还是用作其他功能。

用作数字输入/输出时电路的基本结构如图 2-13 所示。复用的 SCI 功能的输出与端口的输出锁存器一起连接到 AND 门，然后再和其他复用的非 SCI 功能（定时器、RTC、时钟/蜂鸣器的输出、IICA 等）的输出一起连接到 OR 门。当将端口用作通用输出端口使用或者复用输出端口使用时，其设定的基本思想为：不使用的复用功能不能影响要使用的功能的输出。如表 2-8：设定的基本思想 所示。

图 2-13：端口的输出基本结构



注 1：当没有 POM 寄存器时，此信号为 Low 电平（0）。

注 2：当没有复用功能时，此信号为 High 电平（1）。

注 3：当没有复用功能时，此信号为 Low 电平（0）。

表 2-8：设定的基本思想

使用的端口输出功能	不使用的复用功能的输出设定		
	端口功能	SCI 的输出功能	SCI 以外的输出功能
通用数字输出功能	—	High 电平输出（1）	Low 电平输出（0）
SCI 的输出功能	High（1）	—	Low 电平输出（0）
SCI 以外的输出功能	Low（0）	High 电平输出（1）	Low 电平输出（0）注

注：因为 1 个端口有可能复用多个 SCI 以外的输出功能，所以需要将不使用的复用功能的输出置为 Low 电平（0）。有关具体的设定方法，请参照“2.5.2 端口复用功能的配置方法”。

2.5.2 端口复用功能的配置方法

表 2-9：复用功能的配置方法

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PA00	模拟通道	ANI00 VC0_INP0	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG23	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD2/SDO20	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO00	-	-	VC0OUT	-	-	0	0	0	0
	复用输入	-	-	TI00	-	-	-	-	-	-	0	1	×	×
PA01	模拟通道	ANI01 VC0_INP1	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG22	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 and 型	-	SDA20	-	-	-	-	-	-	-	0	0	1	1
	复用输出 or 型	-	-	-	TO01		SPI0_MO	-	PCLBUZ0	-	0	0	0	0
	复用输入	-	RxD2/SDI20	-	TI01	-	SPI0_SI	-	-		0	1	×	×
PA02	模拟通道	ANI02 VC0_INP2	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG21	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD1/SDO10	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-		-	TO02		SPI0_SO	VC1OUT	PCLBUZ1	-	0	0	0	0
	复用输入	-	-	-	TI02	-	SPI0_MI	-	-	-	0	1	×	×
PA03	模拟通道	ANI03 VC0_INP3	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG20	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 and 型	-	SDA10	-	-	-	-	-	-	-	0	0	1	1
	复用输出 or 型	-		-	TO03	-	-	-	-	-	0	0	0	0
	复用输入	-	RxD1/SDI10	-	TI03	TI00_GATE	SPI0_NSS	-	-	-	0	1	×	×
PA04	模拟通道	ANI04 VC0_INP4 OPA0_DA	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG19	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD1/SDO10	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO04	-	-	-	-	-	0	0	0	0
	复用输入	-	-	-	TI04	-	SPI0_NSS	-	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PA05	模拟通道	ANI05 VC0_INN2 / VC1_INN2 DAC0	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG18	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO05	-	SPI0_SCK	-	PCLBUZ0	-	0	0	0	0
	复用输入	-	SS10	-	TI05	TI06_GATE	SPI0_SCK	-	-	-	0	1	×	×
PA06	模拟通道	ANI06 VC0_INN0	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG17	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO06	-	SPI0_SO	VC0OUT	-	-	0	0	0	0
	复用输入	-	SS20	-	TI06	TI07_GATE	SPI0_MI	-	-	-	0	1	×	×
PA07	模拟通道	ANI07 VC0_INN1	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG16	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SCLK10/ SCL10	-	-	-		-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO07	-	SPI0_MO	VC1OUT	PCLBUZ1	-	0	0	0	0
	复用输入	-	SCLK10	-	TI07	-	SPI0_SI	-	-	-	0	1	×	×
PA08	LCD 通道	-	-	-	-	-	-	-	-	SEG35	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD0/ SDO00	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO00	-	-	-	-	-	0	0	0	0
	复用输入	-	-	TI00	-	TI01_GATE	-	-	-	-	0	1	×	×
PA09	LCD 通道		-	-	-	-	-	-	-	COM0	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输入	-	-		TI01	TI02_GATE	-	-	-	-	0	1	×	×
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-	-	-	-	0	0	0	1
	复用输出 or 型	-	-	-	TO01		-	-	PCLBUZ0	-	0	0	0	0
	复用输出 and 型	-	TXD0/ SDO00	-	-	-	-	-	-	-	0	0	1	0
PA10	LCD 通道	-	-	-	-	-	-	-	-	COM1	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输入	-	RXD0/SDI00	-	TI02	TI03_GATE	-	-	-	-	0	1	×	×
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用双向开漏输出 and 型	-	SDA00	-	-	-	-	-	-	-	0	0	1	1
	复用输出 or 型	-	-	-	TO02	-	-	-	PCLBUZ1	-	0	0	0	0

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PA11	LCD 通道	-	-	-	-	-	-	-	-	COM2	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO03	-	SPI0_SO	VC0OUT	-	-	0	0	0	0
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-	-	-	-	0	0	0	1
	复用输入	-	SS00	-	TI03	TI04_GATE	SPI0_MI	-	-	-	0	1	×	×
PA12	LCD 通道	-	-	-	-	-	-	-	-	COM3	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO04	-	SPI0_MO	VC1OUT	-	-	0	0	0	0
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用输入	-	SS11	-	TI04	TI05_GATE	SPI0_SI	-	-	-	0	1	×	×
PA13	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO05	RTC1HZ	-	-	PCLBUZ0	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA00	-	-	-	-	-	-	-	0	0	1	1
	复用双向	SWDIO	-	-	-	-	-	-	-	-	0	×	×	0
	复用输入	-	RXD0/SDI00	-	TI05	-	-	KR4	-	-	0	1	×	×
PA14	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO06	-	-	-	PCLBUZ1	-	0	0	0	0
	复用输出 and 型	-	TXD0/SDO00	-	-	-	-	-	-	-	0	0	1	0
	复用输入	SWCLK	-	-	TI06	-	-	KR1	-	-	0	1	×	×
PA15	LCD 通道	-	-	-	-	-	-	-	-	SEG32	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO07	-	-	-	-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA00	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	RXD0/SDI00	-	TI07	-	SPI0_NSS	KR0	-	-	0	1	×	×
PB00	模拟通道	ANI08/AVREFM	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG13	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD1/SDO10	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO00	-	-	-	PCLBUZ0	-	0	0	0	0
	复用输入	-	-	TI00	-	-	-	-	-	-	0	1	×	×
PB01	模拟通道	ANI09/AVREFP	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG12	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SCLK20/SCL20	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO01	-	-		PCLBUZ1	-	0	0	0	0
	复用输入	-	SCLK20	-	TI01	-	-	-	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PB02	模拟通道	ANI10 VC1_INP0	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG11	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD2/SDO20	-	-	-	-		-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO02	-	-	TA_TO	-	-	0	0	0	0
	复用输入	-	-	-	TI02	-		TA_TI	-	-	0	1	×	×
PB03	LCD 通道		-	-	-	-	-	-	-	CAPH	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO03	-	SPI0_SCK	-	PCLBUZ0	-	0	0	0	0
	复用输出 and 型	-	SCLK11/SCL11	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	SCLK11	-	TI03	TI04_GATE	SPI0_SCK	-	-	-	0	1	×	×
PB04	LCD 通道		-	-	-	-	-	-	-	CAPL	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO04	-	SPI0_SO	TA_TO	-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA11	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	SDI11	-	TI04	TI05_GATE	SPI0_MI	TA_TI	-	-	0	1	×	×
PB05	LCD 通道		-	-	-	-	-	-	-	VL4	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SDO11	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO05	-	SPI0_MO	-	-	-	0	0	0	0
	复用输入	-	-	-	TI05	TI06_GATE	SPI0_SI	-	-	-	0	1	×	×
PB06	LCD 通道		-	-	-	-	-	-	-	VL3	0	1	0	0
	数字通用	GPIO	-		-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-	-	-	-	0	0	0	1
	复用输出 or 型	-	-	-	TO06	-	-	TA_TO	-	-	0	0	0	0
	复用输出 and 型	-	TXD0/SDO00	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	-	-	TI06	-	-	TA_TI	-	-	0	1	×	×
PB07	LCD 通道		-	-	-	-	-	-	-	VL2	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO07	-	-	TA_TON	-	-	0	0	0	0
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用双向开漏输出 and 型	-	SDA00	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	RXD0/SDI00	-	TI07	-	-	-	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PB08	LCD 通道	-	-	-	-	-	-	-	-	VL1	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	-	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	SCLK00/ SCL00	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	SCLK00	-	-	-	-	-	-	-	0	1	×	×
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 or 型			SCLA0							0	0	0	1
	复用输入	-	-			TI07_GATE	TI00	-	-	-	0	1	×	×
	复用输出 or 型	-	-	-	TO00	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	TXD0/ SDO00	-	-	-	-	-	-	-	0	0	1	0
PB10	模拟通道	ANI11 VC1_INP1	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG10	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 or 型			SCLA0							0	0	0	1
	复用输入	-	-		TI02	-	SPI0_SCK	-	-	-	0	1	×	×
	复用输出 and 型	-	TXD1/ SDO10	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-		TO02	-	SPI0_SCK	-	-	-	0	0	0	0
PB11	模拟通道	ANI12 VC1_INP2	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG09	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO03	-	-	-	-	-	0	0	0	0
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用双向开漏输出 and 型	-	SDA10	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	RxD1/ SDI10	-	TI03	TI00_GATE	-	-	-	-	0	1	×	×
PB12	模拟通道	ANI13 VC1_INP3	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG08	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-		TO04	-	-	VC0OUT	-	-	0	0	0	0
	复用输出 and 型	-	TXD1/ SDO10			-	-	-	-	-	0	0	1	0
	复用输入	KR7	-	-	TI04	-	SPI0_NSS	-	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PB13	模拟通道	ANI14 VC1_INN0	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道									SEG07	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-	-	-	-	0	0	0	1
	复用输出 or 型	-	-	-	TO05	-	SPI0_SCK	-	-	-	0	0	0	0
	复用输出 and 型	-	SCLK10/SCL10	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	SCLK10	-	TI05	TI01_GATE	SPI0_SCK	-	-	-	0	1	×	×
PB14	模拟通道	ANI15	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道									SEG06	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO06	RTC1HZ	SPI0_SO	-	-	-	0	0	0	0
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用输入	-	SS01	-	TI06	-	SPI0_MI	-	-	-	0	1	×	×
PB15	模拟通道	ANI16	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG05	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO07	-	SPI0_MO	-	-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA20	-	-	-		-	-	-	0	0	1	1
	复用输入	-	RxD2/SDI20	-	TI07	TI02_GATE	SPI0_SI	-	-	-	0	1	×	×
PC00	模拟通道	ANI17 VC1_INP4	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG27	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-			TO00					-	0	0	0	0
	复用输入	-	SS21	TI00	-	TI03_GATE	-	-	-	-	0	1	×	×
PC01	模拟通道	ANI18 VC1_INP5	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG26	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 and 型	-	SCLK21/SCL21							-	0	0	1	0
	复用输出 or 型	-			TO01			TA_TO		-	0	0	0	0
	复用输入	-	SCLK21	-	TI01	-	-	TA_TI	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PC02	模拟通道	ANI19 VC1_INN1	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG25	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 or 型	-			TO02		SPI0_SO	TA_TON		-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA21							-	0	0	1	0
	复用输入	-	SDI21	-	TI02	-	SPI0_MI	-	-	-	0	1	×	×
PC03	模拟通道	ANI20	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG24	0	1	0	0
	复用输出 or 型	-			TO03		SPI0_MO	TA_TO		-	0	0	0	0
	复用输出 and 型	-	SDO21							-	0	0	1	0
	复用输入	-	-	-	TI03	-	SPI0_SI	TA_TI	-	-	0	1	×	×
PC04	模拟通道	ANI21	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG15	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 or 型	-			TO04				PCLBUZ1	-	0	0	0	0
	复用输出 and 型	-	TXD1/ SDO10							-	0	0	1	0
	复用输入	-	-	-	TI04	-	-	-	-	-	0	1	×	×
PC05	模拟通道	ANI22	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG14	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用双向开漏输出 and 型	-	SDA10	-	-	-	-	-	-	-	0	0	1	1
	复用输出 or 型	-	-	-	TO05	-	-	-	-	-	0	0	0	0
	复用输入	-	RxD1/ SDI10	-	TI05	-	-	-	-	-	0	1	×	×
PC06	模拟通道	ANI23	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	SEG04	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 or 型	-			TO06	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	SCLK01/ SCL01	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	SCLK01	-	TI06	-	-	-	-	-	0	1	×	×
PC07	LCD 通道		-	-	-	-	-	-	-	SEG38	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO07	-	-	-	-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA01	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	SDI01	-	TI07	-	-	KR7	-	-	0	1	×	×

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PC08	LCD 通道	-	-	-	-	-	-	-	-	SEG37	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO00	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	SDO01	-		-	-	-	-	-	0	0	1	0
	复用输入	-	-	TI00	-	-	-	KR6	-	-	0	1	×	×
PC09	LCD 通道	-	-	-	-	-	-	-	-	SEG36	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输入	-	SCLK00	-	TI01	-	-	KR5	-	-	0	1	×	×
	复用输出 or 型	-	-	-	TO01	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	SCLK00/ SCL00	-	-	-	-	-	-	-	0	0	1	0
PC10	LCD 通道		-	-	-	-	-	-	-	SEG00/ COM4	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输入	-	-	-	TI02	-	-	-	-	-	0	1	×	×
	复用输出 and 型	-	TXD2/ SDO20	-	-	-	-	-	-	-	0	0	1	0
	复用输出 or 型	-	-	-	TO02	-	-	-	-	-	0	0	0	0
PC11	LCD 通道	-	-	-	-	-	-	-	-	SEG01/ COM5	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-		TO03	-	-	-	-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA20	-	-	-	-	-	-	-	0	0	1	1
	复用输入	-	RxD2/ SDI20	-	TI03	-	-	-	-	-	0	1	×	×
PC12	LCD 通道		-	-	-	-	-	-	-	SEG02/ COM6	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO04	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	TXD2/ SDO20	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	-	-	TI04	-	-	-	-	-	0	1	×	×
	数字通用	GPIO	-	-	-	-	-	-		-	0	按照需求配置		
	复用输入	-	SS10	-	-	-	-	-	-	-	0	1	×	×
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PD02	LCD 通道		-	-	-	-	-	-	-	SEG03/COM7	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO02	-	-	-	-	-	0	0	0	0
	复用输出 and 型	-	SCLK20/SCL20	-	-	-	-	-	-	-	0	0	1	0
	复用输入	-	SCLK20	-	TI02	-	-	-	-	-	0	1	×	×
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-	-	-	TO01	-	-	-	PCLBUZ1	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA00	-	-	-	-	-	-	-	0	0	1	1
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用输入	-	RXD0/SDI00	-	TI01	-	SPI0_NSS	-	-	-	0	1	×	×
PD04	模拟通道	ANI24 VC0_INP5 OPA0O	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道		-	-	-	-	-	-	-	-	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SCLK10/SCL10						-	-	0	0	1	0
	复用输入	-	SCLK10	-		-	-	-	-	-	0	1	×	×
PD05	模拟通道	ANI25	-	-	-	-	-	-	-	-	1	×	×	×
	LCD 通道	-	-	-	-	-	-	-	-	SEG39	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SCLK20/SCL20						-	-	0	0	1	0
	复用输入	-	SCLK20	-		-	-	-	-	-	0	1	×	×
PD06	LCD 通道		-	-	-	-	-	-	-	SEG34	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-		-	-	0	0	0	1
	复用输入							KR3			0	1	×	×
PD07	LCD 通道		-	-	-	-	-	-	-	SEG33	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	SCLK00/SCL00	-	-	-	-	-	-	-	0	0	1	1
	复用双向开漏输出 or 型	-	-	SDAA0	-	-	-	-	-	-	0	0	0	1
	复用输入	-	SCLK00					KR2	-	-	0	1	×	×
PH00	LCD 通道	-	-	-	-	-	-	-	-	-	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		

端口名称	端口模式	端口功能									寄存器设置			
		默认功能	第 1 功能	第 2 功能	第 3 功能	第 4 功能	第 5 功能	第 6 功能	第 7 功能	LCD 功能	PMCxx	PMxx	Pxx	POMxx
PH01	LCD 通道	-	-	-	-	-	-	-	-	-	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 and 型	-	TXD1/ SDO10	-	-	-	-		-	-	0	0	1	0
	复用双向开漏输出 or 型	-	-	SDAA0		-	-	-	-	-	0	0	0	1
PH02	LCD 通道	-	-	-	-	-	-	-	-	-	0	1	0	0
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
	复用输出 or 型	-			TO01	-	-		-	-	0	0	0	0
	复用双向开漏输出 and 型	-	SDA10	-	-	-	-	-	-	-	0	0	1	1
	复用双向开漏输出 or 型	-	-	SCLA0	-	-	-		-	-	0	0	0	1
	复用输入	-	RxD1/ SDI10	-	TI01	-	-	-	-	-	0	1	×	×
PH03	模拟通道	-	-	-	-	-	-	-	-	-	1	×	×	×
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		
PH04	模拟通道	-	-	-	-	-	-	-	-	-	1	×	×	×
	数字通用	GPIO	-	-	-	-	-	-	-	-	0	按照需求配置		

配置说明：

- PA00~PA07、PB00~PB03、PB10~PB15、PC00~PC06、PD04、PD05、PD08、PD09 上电后默认作为模拟端口使用，如果要用于数字通用 GPIO 或数字复用功能，端口需要配置成数字模式（PMCxx=0）。
- PA13、PA14 端口默认作为调试口使用并且复位后默认上拉，若要使用其 GPIO 功能需要配置 DBGSTOPCR.SWDIS 寄存器位为 1，将调试功能屏蔽。DBGSTOPCR.SWDIS 寄存器说明请参考“第 1 章 CPU”。
- PA08~PA12、PA15、PC07~PC12、PD02、PD06、PD07、PH01~PH04 端口上电后默认作为数字 GPIO 使用。
- 使用端口的数字输入功能时，端口必须配置成输入模式（PMxx=1）。
- 使用端口的数字输出功能时，端口必须配置成输出模式（推挽或开漏）（PMxx=0）。
- 使用端口的复用输出功能时，需要按照上表设置该端口输出锁存器 Pxx
- 使用 PH00 端口的 GPIO 功能时，需要先将该端口的复位功能屏蔽掉，设置 RSTM=1。
- 使用 PH01、PH02 端口的 GPIO 功能或者复用功能时，确认其 X1 震荡模式和外部时钟输入模式没有开启。请参考“第 4 章 时钟发生电路”。
- 使用 PH03、PH04 端口的 GPIO 功能或者复用功能时，确认其 XT1 震荡模式和外部时钟输入模式没有开启。请参考“第 4 章 时钟发生电路”。
- 每个端口都可配置为外部中断，中断触发类型可以配置成上升沿触发，下降沿触发或者双沿触发。中断端口的选择需要配置 INTPnPCFG 寄存器（n=0~5）详情请参考 2.3.11 外部中断端口选择寄存器（INTPnPCFG），触发类型的选择需要配置 EPG0 和 EGN0 寄存器，详情请参考“第 23 章 中断功能”。
- 端口复用为 LCD 端口时（COM0~7，SEG0~41，VLCD1~4，CAPL，CAPH），需要将端口模式控制寄存器 PMCxx 对应位配置为 0，并且将 LCD 端口模式寄存器 SEG0~SEG3 对应的位配置为 1，SEG0~SEG3 的配置方法，请参考“第 20 章 LCD 控制器/驱动器”。
- 为了避免贯通电流，PB03~PB08 复位后默认关闭输入功能，若想开启 GPIO 的输入功能需要配置 ISCLCD 寄存器，当 ISCLCD.ISCCAP 设置为 1 时，PB03 和 PB04 的施密特输入使能。当 ISCLCD.ISCVL 设置为 1 时，PB05~PB08 的施密特输入使能。详情请参考“第 20 章 LCD 控制器/驱动器”。

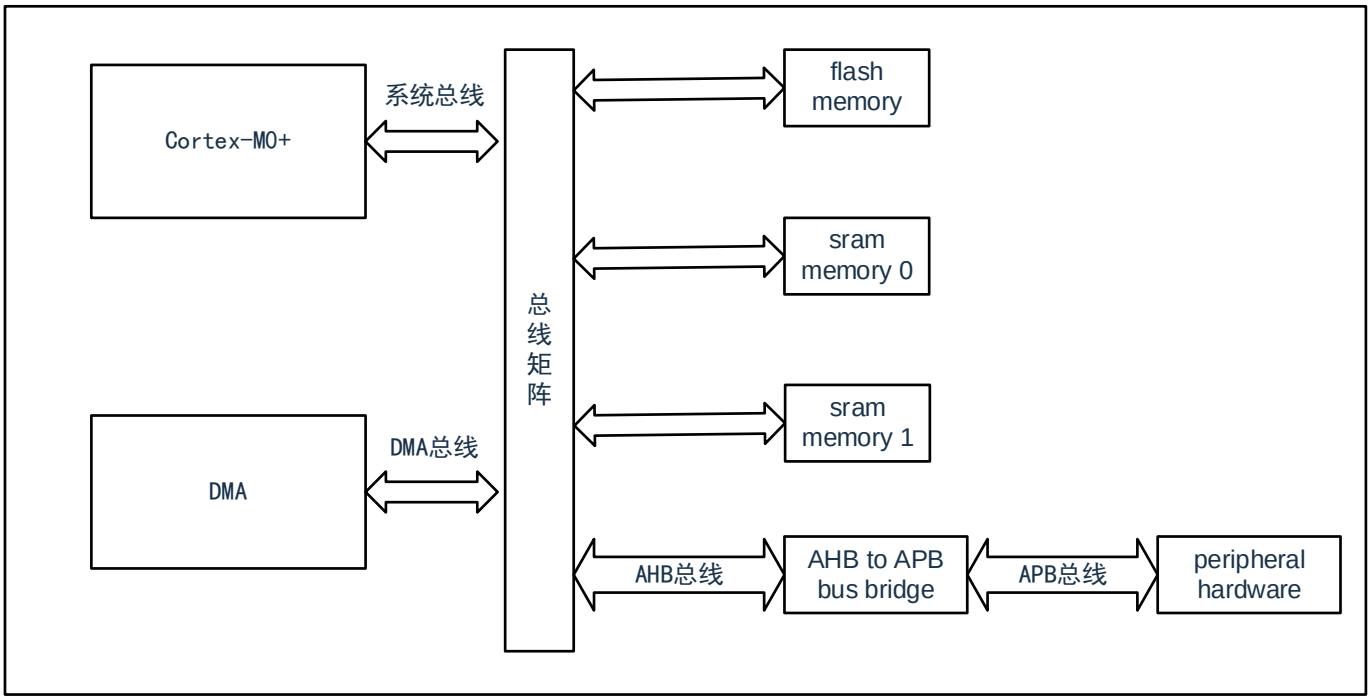
第3章 系统结构

3.1 概述

本产品系统由以下部分组成：

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

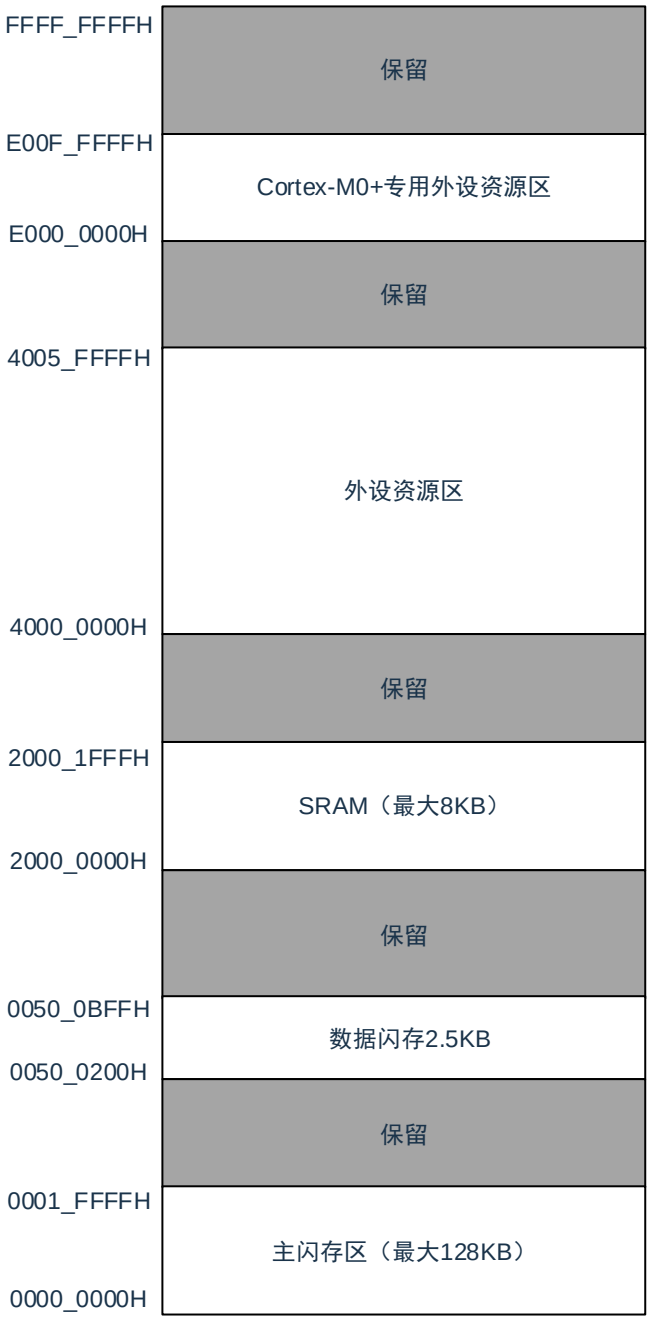
图 3-1：系统结构示意图



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

3.2 系统地址划分

图 3-2：地址区域划分示意图



3.3 外设地址分配

表 3-1：外设的寄存器组起始地址

起始地址	外设	备注
0x4000_0000 - 0x4000_4FFF	保留	
0x4000_5000 - 0x4000_5FFF	DMA	
0x4000_6000 - 0x4000_6FFF	中断控制	
0x4000_7000 - 0x4001_8FFF	保留	
0x4001_9000 - 0x4001_AFFF	保留	
0x4001_B000 - 0x4001_BFFF	DBGREG	
0x4001_C000 - 0x4001_CFFF	DIV	
0x4001_D000 - 0x4001_FFFF	保留	
0x4002_0000 - 0x4002_03FF	FLASH 控制	
0x4002_0400 - 0x4002_0FFF	时钟控制	
0x4002_1000 - 0x4002_1002	看门狗定时器	
0x4002_1003 - 0x4002_1800	保留	
0x4002_1800 - 0x4002_1BFF	高速 CRC	详见第 29 章 安全功能
0x4002_1C00 - 0x4002_1FFF	FLASH 控制	
0x4002_2000 - 0x4003_FFFF	保留	
0x4004_0000 - 0x4004_0FFF	GPIO	
0x4004_1000 - 0x4004_13FF	pcbz	
0x4004_1400 - 0x4004_17FF	通用 CRC	详见第 29 章 安全功能
0x4004_1800 - 0x4004_1BFF	联动控制器	
0x4004_1C00 - 0x4004_1FFF	外部中断控制	
0x4004_2000 - 0x4004_23FF	键中断	
0x4004_2400 - 0x4004_27FF	实时时钟	
0x4004_2800 - 0x4004_2FFF	保留	
0x4004_3000 - 0x4004_33FF	通用定时器单元	
0x4004_3400 - 0x4004_3FFF	保留	
0x4004_4000 - 0x4004_43FF	定时器 A	
0x4004_4400 - 0x4004_4FFF	保留	
0x4004_5000 - 0x4004_53FF	AD 转换器	
0x4004_5400 - 0x4004_57FF	DA 转换	
0x4004_5800 - 0x4004_5BFF	比较器	
0x4004_5C00 - 0x4004_5FFF	放大器	
0x4004_6000 - 0x4004_6BFF	串行通信单元	
0x4004_6C00 - 0x4004_6FFF	保留	
0x4004_7000 - 0x4004_73FF	串行接口 IICA0	
0x4004_7400 - 0x4004_7FFF	保留	
0x4004_7800 - 0x4004_7BFF	SPIHS0	
0x4004_7C00 - 0x4004_7FFF	保留	
0x4004_8000 - 0x4004_83FF	IrDA	
0x4004_8400 - 0x4004_8FFF	保留	
0x4004_9000 - 0x4004_93FF	LCD	

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 3 种系统时钟和时钟振荡电路。

(1) 主系统时钟

① X1 振荡电路

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

② 高速内部振荡器（高速 OCO）

能通过选项字节（000C2H）从 F_{HOCO}=32MHz、16MHz、8MHz、4MHz、2MHz 和 1MHz(典型值)中选择频率进行振荡，F_{IH} 和 F_{HOCO} 的频率相同。在解除复位后，CPU 一定以此高速内部振荡器时钟开始运行。能通过进入深度睡眠模式或者设置 HIOSTOP 位（CSC 寄存器的 bit0）使振荡停止。能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改选项字节设置的频率。有关频率设置，请参照“图 4-10：高速内部振荡器的频率选择寄存器（HOCODIV）的格式”。

另外，能由 EXCLK/X2/PH02 引脚提供外部主系统时钟（F_{EX}=1~20MHz），并且能通过进入深度睡眠模式或者设置 MSTOP 位将外部主系统时钟的输入置为无效。

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

(2) 副系统时钟

- XT1 振荡电路

能通过给 XT1 引脚和 XT2 引脚连接 32.768KHz 的谐振器使 F_{XT}=32.768KHz 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/PH04 引脚提供外部副系统时钟（F_{EXS}=32.768KHz），并且能通过设置 XTSTOP 位将外部副系统时钟的输入置为无效。

(3) 低速内部振荡器时钟（低速 OCO）

能使 F_{IL}=32.768KHz(典型值)的时钟振荡。

可以将低速内部振荡器时钟用作 CPU 时钟。

SysTick 定时器使用低速内部振荡器时钟作为外部参考时钟。

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

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

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

备注：F_X：X1 时钟振荡频率

F_{HOCO}：高速内部振荡器的时钟频率（最大 32MHz）

F_{IH}：高速内部振荡器的时钟频率（最大 32MHz）

F_{EX}：外部主系统时钟频率

F_{XT}：XT1 时钟振荡频率

F_{EXS}：外部副系统时钟频率

F_{IL}：低速内部振荡器的时钟频率

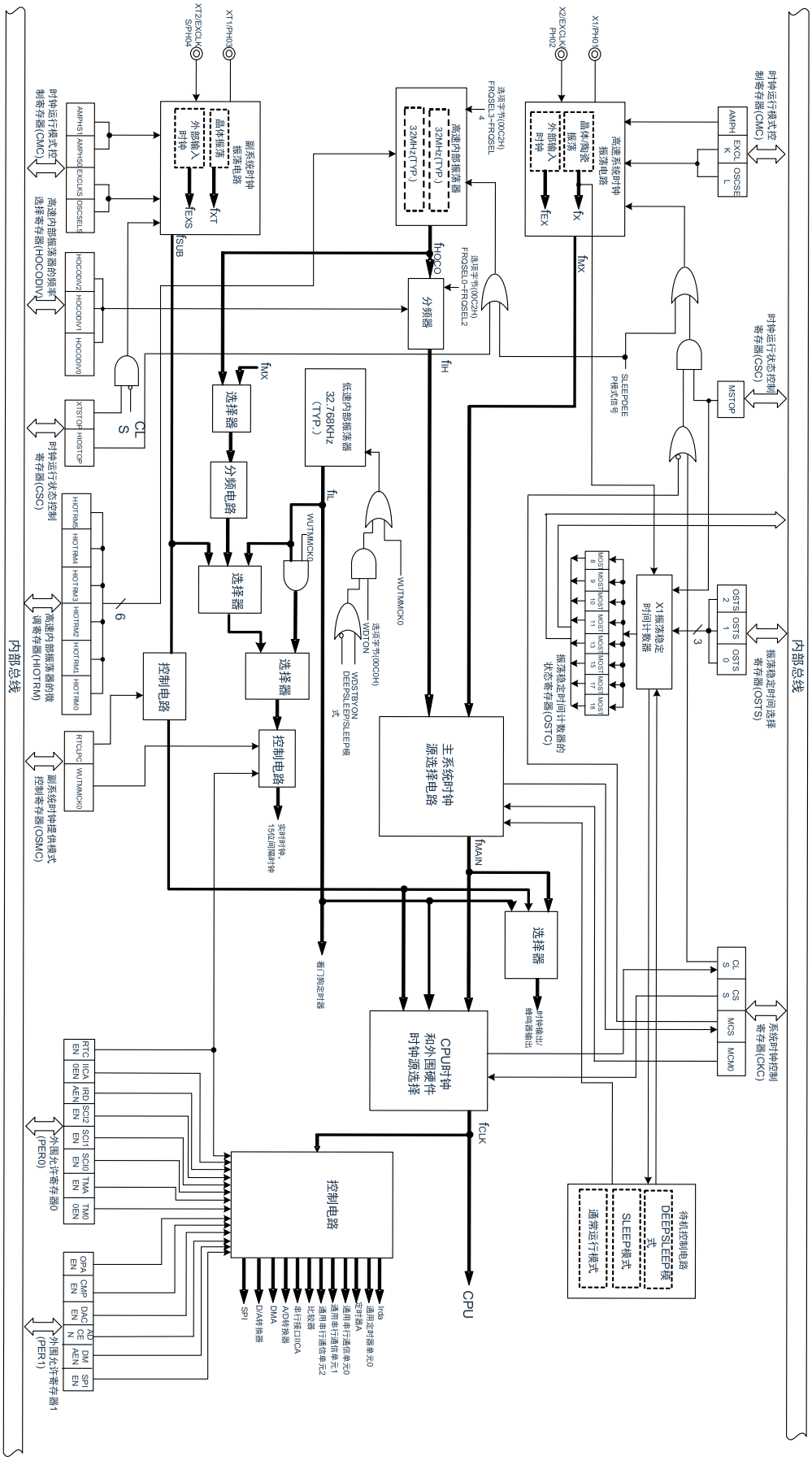
4.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表 4-1：时钟发生电路的结构

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

图 4-1：时钟发生电路的框图



备注：F_X：X1时钟振荡频率

F_{HOCO}：高速内部振荡器的时钟频率（最大32MHz）

F_{IH}：高速内部振荡器的时钟频率（最大32MHz）

F_{EX}：外部主系统时钟频率

F_{MX}：高速系统时钟频率

F_{MAIN}：主系统时钟频率

F_{XT}：XT1时钟振荡频率

F_{EXS}：外部副系统时钟频率

F_{SUB}：副系统时钟频率

F_{CLK}：CPU/外围硬件的时钟频率

F_{IL}：低速内部振荡器的时钟频率

4.3 控制时钟发生电路的寄存器

通过以下寄存器控制时钟发生电路。

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

注意：分配的寄存器和位因产品而不同。必须给未分配的位设置初始值。

4.3.1 时钟运行模式控制寄存器（CMC）

这是设置 X1/PH01、X2/EXCLK/PH02、XT1/PH03、XT2/EXCLKS/PH04 引脚的运行模式以及选择振荡电路增益的寄存器。

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

在产生复位信号后，此寄存器的值变为“00H”。

图 4-2：时钟运行模式控制寄存器（CMC）的格式

地址： 40020400H 复位后： 00H

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

Bit7~ Bit6	EXCLK OSCSEL:	高速系统时钟引脚的运行模式	X1/PH01 引脚	X2/EXCLK/PH02 引脚
	00=	端口模式	输入输出端口	
	01=	X1 振荡模式	连接晶体或者陶瓷谐振器。	
	10=	端口模式	输入输出端口	
	11=	外部时钟输入模式	输入输出端口	外部时钟输入
Bit5~ Bit4	EXCLKS OSCSELS:	副系统时钟引脚的运行模式	XT1/PH03 引脚	XT2/EXCLKS/PH04 引脚
	00=	端口模式	输入输出端口	
	01=	XT1 振荡模式	连接晶体谐振器。	
	10=	端口模式	输入输出端口	
	11=	外部时钟输入模式	输入输出端口	外部时钟输入
Bit2~ Bit1	AMPHS1 AMPHS0:	XT1 振荡电路的振荡模式选择		
	00=	低功耗振荡（默认）		
	01=	通常的振荡		
	10=	超低功耗振荡		
	11=	禁止设置。		
	AMPH:	X1 时钟振荡频率的控制		
	0=	1MHz≤F_X≤10MHz		
	1=	10MHz＜F_X≤20MHz		

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

注意：

1. 在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。当以初始值（“00H”）使用 CMC 寄存器时，为了防止程序失控时的误动作（如果误写“00H”以外的值就不能恢复），必须在解除复位后将 CMC 寄存器置“00H”。
2. 在解除复位后并且在通过设置时钟运行状态控制寄存器（CSC）开始 X1 或者 XT1 振荡前，必须设置 CMC 寄存器
3. 当 X1 时钟振荡频率超过 10MHz 时，必须将 AMPH 位置“1”。
4. 必须在解除复位后并且在选择 F_{IH} 作为 F_{CLK} 的状态（将 F_{CLK} 切换为 F_{MX} 或者 F_{SUB} 前的状态）下设置 AMPH 位、AMPHS1 位和 AMPHS0 位。
5. 必须通过软件对 F_{XT} 的振荡稳定时间进行计数。
6. 系统时钟的频率上限为 32MHz，但是 X1 振荡电路的频率上限为 20MHz。

备注：F_X：X1 时钟振荡频率

4.3.2 系统时钟控制寄存器（CKC）

这是选择 CPU/外围硬件时钟和主系统时钟的寄存器。

通过 8 位存储器操作指令设置 CKC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-3：系统时钟控制寄存器（CKC）的格式

地址： 40020404H 复位后： 00H

符号	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0			
R/W	R/W 注1							

Bit7 CLS: CPU/外围硬件时钟（F_{CLK}）的状态
0= 主系统时钟（F_{MAIN}）
1= 副系统时钟（F_{SUB}）

Bit6 CSS 注2: CPU/外围硬件时钟（F_{CLK}）的选择
0= 主系统时钟（F_{MAIN}）
1= 副系统时钟（F_{SUB}）

Bit5 MCS: 主系统时钟（F_{MAIN}）的状态
0= 高速内部振荡器时钟（F_{IH}）
1= 高速系统时钟（F_{MX}）

Bit4 MCM0 注2: 主系统时钟（F_{MAIN}）的运行控制
0= 选择高速内部振荡器时钟（F_{IH}）作为主系统时钟（F_{MAIN}）。
1= 选择高速系统时钟（F_{MX}）作为主系统时钟（F_{MAIN}）。

注 1：bit7 和 bit5 是只读位，必须将 bit0~3 置“0”。

注 2：禁止在将 CSS 位置“1”的状态下更改 MCM0 位的值。

备注：

1. 给 CPU 和外围硬件提供 CSS 位设置的时钟。如果更改 CPU 时钟，就同时更改外围硬件时钟（实时时钟、15 位间隔定时器、时钟输出/蜂鸣器输出和看门狗定时器除外）。因此，如果更改 CPU/外围硬件的时钟，就必须停止各外围功能。
2. 如果将副系统时钟用作外围硬件时钟，就无法保证 A/D 转换器和 IICA 的运行。有关外围硬件的运行特性，请参照各外围硬件的章节和数据手册的电气特性。
3. F_{HOCO}：高速内部振荡器的时钟频率（最大 32MHz）
   F_{IH}：高速内部振荡器的时钟频率（最大 32MHz）
   F_{MX}：高速系统时钟频率
   F_{MAIN}：主系统时钟频率
   F_{SUB}：副系统时钟频率

4.3.3 时钟运行状态控制寄存器（CSC）

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

在产生复位信号后，此寄存器的值变为“C0H”。

图 4-4：时钟运行状态控制寄存器（CSC）的格式

地址： 40020401H 复位后： C0H

符号	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0					HIOSTOP
R/W	R/W							

Bit7 MSTOP: 高速系统时钟的运行控制

	X1 振荡模式	外部时钟输入模式	输入端口模式
0=	X1 振荡电路运行	EXCLK 引脚的外部时钟有效	输入端口
1=	X1 振荡电路停止	EXCLK 引脚的外部时钟无效	

Bit6 XTSTOP: 副系统时钟的运行控制

	XT1 振荡模式	外部时钟输入模式	输入端口模式
0=	XT1 振荡电路运行	EXCLKS 引脚的外部时钟有效	输入端口
1=	XT1 振荡电路停止	EXCLKS 引脚的外部时钟无效	

Bit0 HIOSTOP: 高速内部振荡器时钟的运行控制

0= 高速内部振荡器运行

1= 高速内部振荡器停止

注意：

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

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

表 4-2：时钟停止方法

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

4.3.4 振荡稳定时间计数器的状态寄存器（OSTC）

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

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

能通过 8 位存储器操作指令读 OSTC 寄存器。

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

备注：在以下情况下，振荡稳定时间计数器开始计数：

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

图 4-5：振荡稳定时间计数器的状态寄存器（OSTC）的格式

地址： 40020402H 复位后： 00H

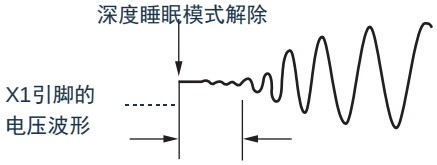
符号	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
R/W	R							

Bit7~Bit0 OSTC: 振荡稳定时间状态

		F_X=10MHz	F_X=20MHz
00000000=	小于 $2^8/F_X$	小于 25.6us	小于 12.8us
10000000=	至少 $2^8/F_X$	至少 25.6us	至少 12.8us
11000000=	至少 $2^9/F_X$	至少 51.2us	至少 25.6us
11100000=	至少 $2^{10}/F_X$	至少 102us	至少 51.2us
11110000=	至少 $2^{11}/F_X$	至少 204us	至少 102us
11111000=	至少 $2^{13}/F_X$	至少 819us	至少 409us
11111100=	至少 $2^{15}/F_X$	至少 3.27ms	至少 1.63ms
11111110=	至少 $2^{17}/F_X$	至少 13.1ms	至少 6.55ms
11111111=	至少 $2^{18}/F_X$	至少 26.2ms	至少 13.1ms

注意：

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



a

备注：F_X：X1 时钟振荡频率

4.3.5 振荡稳定时间选择寄存器（OSTS）

这是选择 X1 时钟的振荡稳定时间的寄存器。

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

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

能通过 OSTC 寄存器确认 OSTS 寄存器事先设置的时间。

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

图 4-6：振荡稳定时间选择寄存器（OSTS）的格式

地址： 40020403H 复位后： 07H

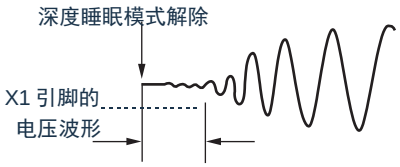
符号	7	6	5	4	3	2	1	0
OSTS	0					OSTS2	OSTS1	OSTS0
R/W	R/W							

Bit2~Bit0 OSTS＜2:0＞: 振荡稳定时间状态

		F_X=10MHz	F_X=20MHz
000=	$2^8/F_X$	25.6us	12.8us
001=	$2^9/F_X$	51.2us	25.6us
010=	$2^{10}/F_X$	102us	51.2us
011=	$2^{11}/F_X$	204us	102us
100=	$2^{13}/F_X$	819us	409us
101=	$2^{15}/F_X$	3.27ms	1.63ms
110=	$2^{17}/F_X$	13.1ms	6.55ms
111=	$2^{18}/F_X$	26.2ms	13.1ms

注意：

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



a

备注：F_X：X1 时钟振荡频率

4.3.6 外围允许寄存器0、1（PER0、PER1）

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

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

- 实时时钟、15位间隔定时器
- IrDA
- A/D转换器
- 串行接口IICA0
- 通用串行通信单元0/1/2
- Timer80
- D/A转换器
- 比较器
- 增强型DMA
- 定时器A

通过 8 位存储器操作指令设置 PER0 寄存器，PER1 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 4-7：外围允许寄存器 0（PER0）的格式

地址： 40020420H 复位后： 00H

符号	7	6	5	4	3	2	1	0
PER0	RTCEN 注	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN
R/W	R/W							

Bit7 RTCEN: 提供实时时钟（RTC）和 15 位间隔定时器的输入时钟的控制
0= 停止提供输入时钟。
• 不能写实时时钟（RTC）和15位间隔定时器使用的SFR。
• 实时时钟（RTC）和 15 位间隔定时器处于复位状态。
1= 提供输入时钟。
• 能读写实时时钟（RTC）和 15 位间隔定时器使用的 SFR。

Bit6 IICAEN: 提供串行接口 IICA 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写串行接口IICA使用的SFR。
• 串行接口 IICA 处于复位状态。
1= 提供输入时钟。
• 能读写串行接口 IICA0 使用的 SFR。

Bit5 IRDAEN: 提供串行接口 IRDA 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写IRDA使用的SFR。
• IRDA 处于复位状态。
1= 提供输入时钟。
• 能读写 IRDA 使用的 SFR。

Bit4 SCI2EN: 提供通用串行通信单元 2 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写通用串行通信单元2使用的SFR。
• 通用串行通信单元 2 处于复位状态。
1= 提供输入时钟。
• 能读写通用串行通信单元 2 使用的 SFR。

Bit3 SCI1EN: 提供通用串行通信单元 1 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写通用串行通信单元1使用的SFR。

• 通用串行通信单元 1 处于复位状态。

1= 提供输入时钟。
• 能读写通用串行通信单元 1 使用的 SFR。

Bit2 SCI0EN: 提供通用串行通信单元 0 的输入时钟的控制

0= 停止提供输入时钟。
• 不能写通用串行通信单元0使用的SFR。
• 通用串行通信单元 0 处于复位状态。

1= 提供输入时钟。
• 能读写通用串行通信单元 0 使用的 SFR。

Bit1 TMAEN: 提供 TMA 模块的输入时钟的控制

0= 停止提供输入时钟。
• 不能写TMA使用的SFR。
• TMA 处于复位状态。

1= 提供输入时钟。
• 能读写 TMA 使用的 SFR。

Bit0 TM80EN: 提供 Timer8 的输入时钟的控制

0= 停止提供输入时钟。
• 不能写通用定时器单元0使用的SFR。
• 通用定时器单元 0 处于复位状态。

1= 提供输入时钟。
• 能读写通用定时器单元 0 使用的 SFR。

注：RTCEN 位只在上电复位时被初始化，而在其他复位时保持不变。

图 4-8：外围允许寄存器 1（PER1）的格式

地址： 4002081AH 复位后： 00H

符号	7	6	5	4	3	2	1	0
PER1	OPAEN	CMPEN	DACEN	ADCEN	0	OSDCEN	DMAEN	SPIHSEN
R/W	R/W							

Bit7 OPAEN: 提供 OPA 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写OPA使用的SFR。
• OPA 处于复位状态。
1= 提供输入时钟。
• 能读写 OPA 使用的 SFR。

Bit6 CMPEN: 提供比较器的输入时钟的控制
0= 停止提供输入时钟。
• 不能写比较器使用的SFR。
• 比较器处于复位状态。
1= 提供输入时钟。
• 能读写比较器使用的 SFR。

Bit5 DACEN: 提供 D/A 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写D/A使用的SFR。
• D/A 处于复位状态。
1= 提供输入时钟。
• 能读写 D/A 使用的 SFR。

Bit4 ADCEN: 提供 A/D 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写A/D使用的SFR。
• A/D 处于复位状态。
1= 提供输入时钟。
• 能读写 A/D 使用的 SFR。

Bit2 OSDCEN: 提供 OSDC 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写OSDC使用的SFR。
• OSDC 处于复位状态。
1= 提供输入时钟。
• 能读写 OSDC 使用的 SFR。

Bit1 DMAEN: 提供 DMA 的输入时钟的控制
0= 停止提供输入时钟。
• DMA 不能运行。
1= 提供输入时钟。
• DMA 能运行。

Bit0 SPIHSEN: 提供 SPIHS 的输入时钟的控制
0= 停止提供输入时钟。
• 不能写SPIHS使用的SFR。
• SPIHS 处于复位状态。
1= 提供输入时钟。
• 能读写 SPIHS 使用的 SFR。

4.3.7 副系统时钟提供模式控制寄存器（OSMC）

OSMC 寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

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

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

通过 8 位存储器操作指令设置 OSMC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-9：副系统时钟提供模式控制寄存器（OSMC）的格式

地址： 40020423H 复位后： 00H

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

Bit7 RTCLPC: 深度睡眠模式和 CPU 以副系统时钟运行的睡眠模式中的设置

0= 允许给外围功能提供副系统时钟
（有关允许运行的外围功能，请参照表 25-1~表 25-3）。

1= 停止给实时时钟和 15 位间隔定时器以外的外围功能提供副系统时钟。

Bit4 WUTMMCK0: 实时时钟、15 位间隔定时器和定时器 A 的运行时钟的选择

0=
- 副系统时钟为实时时钟和15位间隔定时器的运行时钟。
- 不能选择低速内部振荡器作为定时器 A 的计数源。

1=
- 低速内部振荡器时钟为实时时钟和15位间隔定时器的运行时钟。
- 能选择低速内部振荡器或者副系统时钟作为定时器 A 的计数源。

4.3.8 高速内部振荡器的频率选择寄存器（HOCODIV）

这是更改选项字节（000C2H）设置的高速内部振荡器频率的寄存器。

通过 8 位存储器操作指令设置 HOCODIV 寄存器。

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

图 4-10：高速内部振荡器的频率选择寄存器（HOCODIV）的格式

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

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

Bit2~Bit0 HOCODIV<2:0>: 高速内部振荡器时钟频率的选择

FRQSEL4,3=00

000= F_{HOCO}=32MHz
F_{IH}=32MHz

001= F_{HOCO}=32MHz
F_{IH}=16MHz

010= F_{HOCO}=32MHz
F_{IH}=8MHz

011= F_{HOCO}=32MHz
F_{IH}=4MHz

100= F_{HOCO}=32MHz
F_{IH}=2MHz

101= F_{HOCO}=32MHz
F_{IH}=1MHz

上述以外: 禁止设置。

注意：

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

4.3.9 高速内部振荡器的微调寄存器（HIOTRM）

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

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

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

图 4-11：高速内部振荡器的微调寄存器（HIOTRM）的格式

地址： 40021C00H 复位后： 注

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

Bit5~Bit0 HIOTRM<5:0>: 高速内部振荡器

000000= 最低速

000001=

000010=

000011=

000100=

•

•

•

111110=

111111= 最高速

注：复位值是发货时的调整值。

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

4.3.10 副系统时钟选择寄存器（SUBCKSEL）

SUBCKSEL 寄存器是选择副系统时钟 F_{SUB} 和低速内部振荡器时钟 F_{IL} 以及选择低速内部振荡器时钟频率的寄存器。

通过 8 位存储器操作指令设置 SUBCKSEL 寄存器。

在产生复位信号后，此寄存器的值变为“02H”。

图 4-12：副系统时钟选择寄存器（SUBCKSEL）的格式

地址： 40020407H 复位后： 02H

符号	7	6	5	4	3	2	1	0
SUBCKSEL	0						LOCOSEL	SELLOSC
R/W	R/W							

Bit1 LOCOSEL: 低速内部振荡器时钟的频率选择

0= • 低速内部振荡器时钟频率为 16.384KHz。

1= • 低速内部振荡器时钟频率为 32.768KHz（Default）。

Bit0 SELLOSC: 副系统时钟和低速内部振荡器时钟的选择

0= • 选择副系统时钟（Default）。

1= • 选择低速内部振荡器时钟。

4.4 系统时钟振荡电路

4.4.1 X1振荡电路

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

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

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

当不使用 X1 振荡电路时，必须设置为输入端口模式（EXCLK、OSCSEL=0、0）。而且，当也不用作输入端口时，请参照“2.4 各未使用端口的处理”。

X1 振荡电路的外接电路例子如图 4-13 所示。

图 4-13：X1 振荡电路的外接电路例子

注意事项如下页所示。

4.4.2 XT1振荡电路

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

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

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

当不使用 XT1 振荡电路时，必须设置为输入端口模式（EXCLKS、OSCSELS=0、0）。而且，当也不用作输入端口时，请参考 2.4 各未使用端口的处理。XT1 振荡电路的外接电路例子如图 4-14 所示。

图 4-14：XT1 振荡电路的外接电路例子

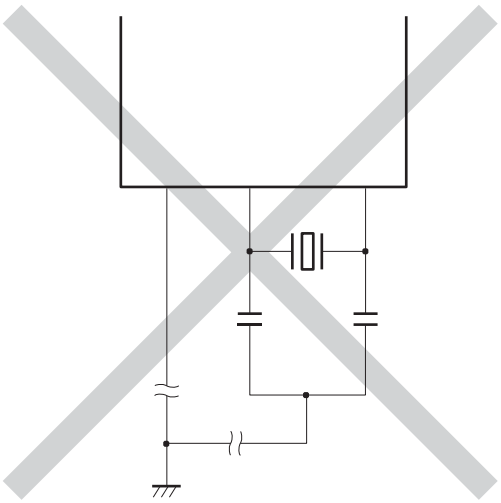
注意：当使用X1振荡电路和XT1振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图4-13和4-14中的虚线部分进行布线：

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

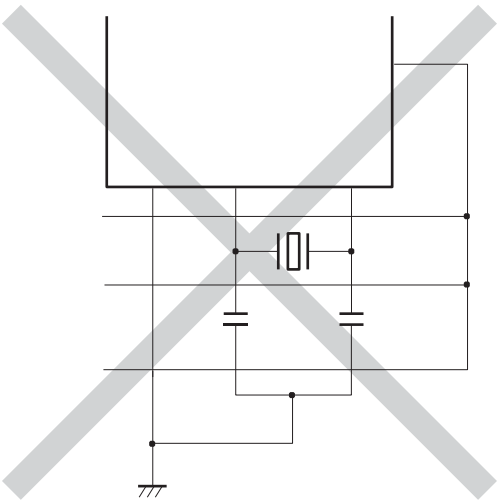
不正确的谐振器连接例子如图 4-15 所示。

图 4-15：不正确的谐振器连接例子(1/2)

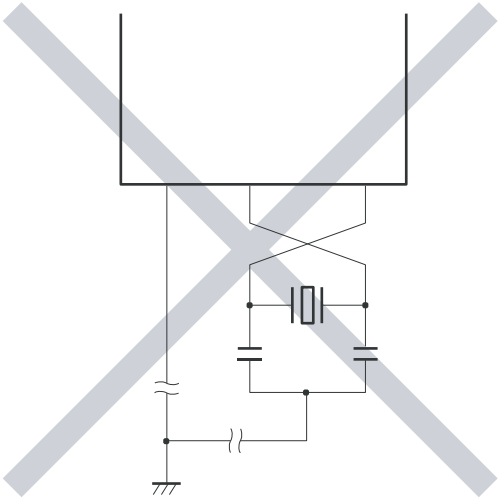
(a)连接电路的布线太长



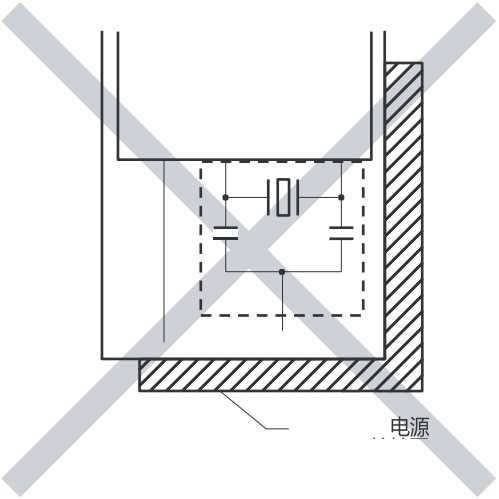
(b)信号线交叉



(c)X1 和 X2 的信号线交叉布线



(d)X1 和 X2 的布线下方有电源或者接地图形

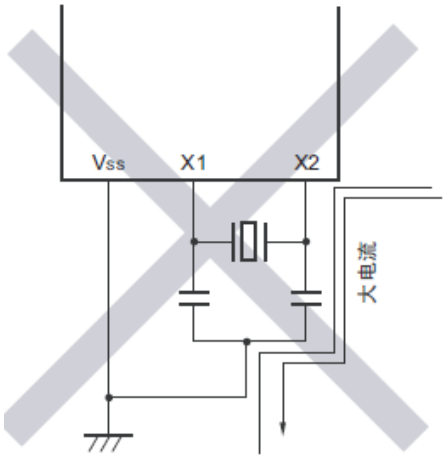


注意：

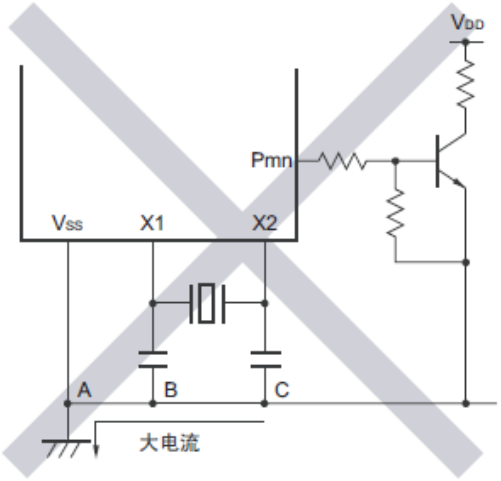
1. 在多层板或者双面板中，不能在 X1 引脚、X2 引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。
2. 在使用副系统时钟的情况下，请阅读时分别用 XT1 和 XT2 代替 X1 和 X2，并且在 XT2 侧插入串联电阻。

图 4-15：不正确的谐振器连接例子(2/2)

(e)有变化的大电流接近信号线

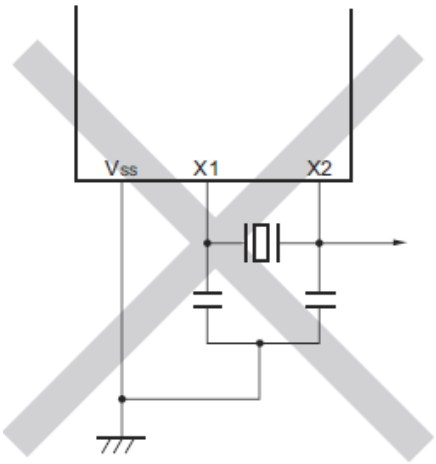


(f)振荡电路的接地线有电流流过
（A 点、B 点、C 点的电位发生变化）



(g)取出信号

(g) 取出信号



注意：

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

4.4.3 高速内部振荡器

CMS32H6157 内置高速内部振荡器。能通过选项字节(000C2H)从 32MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 和 1MHz 中选择频率。能通过时钟运行状态控制寄存器(CSC)的 bit0(HIOSTOP)控制振荡。

在解除复位后，高速内部振荡器自动开始振荡。

4.4.4 低速内部振荡器

CMS32H6157 内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器、实时时钟、15 位间隔定时器的时钟和定时器 A 的时钟，以及 SysTick 定时器的外部参考时钟，也可用作系统时钟。

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

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

4.5 时钟发生电路的运行

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

F_{MAIN}：主系统时钟

F_{MX}：高速系统时钟

F_{X}：X1时钟

F_{EX}：外部主系统时钟

F_{IH}：高速内部振荡器时钟

F_{SUB}：副系统时钟

F_{XT}：XT1时钟

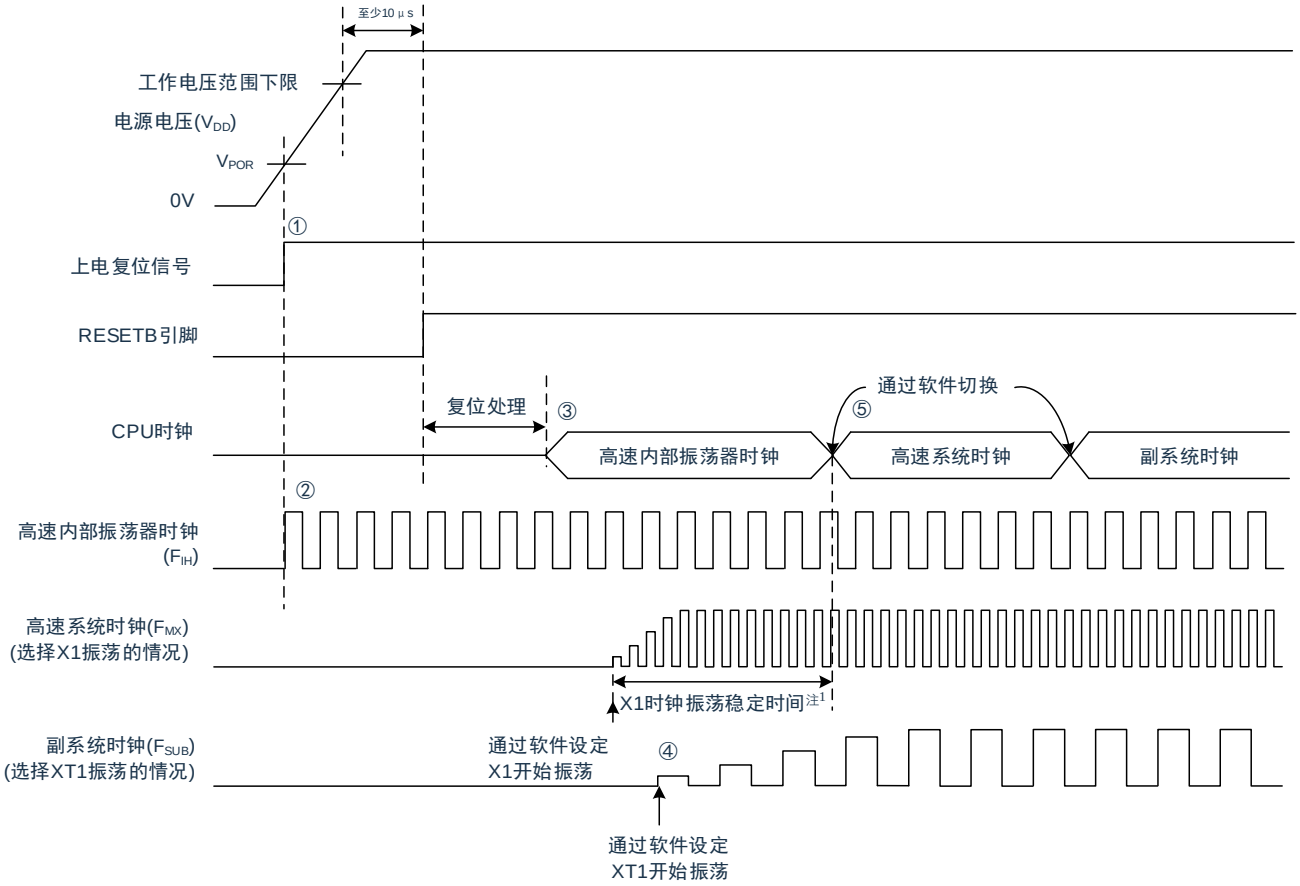
F_{EXS}：外部副系统时钟

F_{IL}：低速内部振荡器时钟

F_{CLK}：CPU/外围硬件时钟

CMS32H6157 在解除复位后，CPU 通过高速内部振荡器的输出开始运行。接通电源时的时钟发生电路的运行如图 4-16 所示。

图 4-16：接通电源时的时钟发生电路的运行



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

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

4.6 时钟控制

4.6.1 高速内部振荡器的设置例子

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

【选项字节的设置】

地址： 000C2H

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

Bit4~Bit0 FRQSEL＜4:0＞: 高速内部振荡器的频率

	F_{HOCO}	F_{IH}
00000=	32MHz	32MHz
00001=	32MHz	16MHz
00010=	32MHz	8MHz
00011=	32MHz	4MHz
00100=	32MHz	2MHz
00101=	32MHz	1MHz
上述以外:	禁止设置。	

【高速内部振荡器的频率选择寄存器（HOCODIV）的设置】

地址： 0x40021C20

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

Bit2~Bit0 HOCODIV＜2:0＞: 高速内部振荡器的频率

000= F_{HOCO}=32MHz
F_{IH}=32MHz

001= F_{HOCO}=32MHz
F_{IH}=16MHz

010= F_{HOCO}=32MHz
F_{IH}=8MHz

011= F_{HOCO}=32MHz
F_{IH}=4MHz

100= F_{HOCO}=32MHz
F_{IH}=2MHz

101= F_{HOCO}=32MHz
F_{IH}=1MHz

上述以外: 禁止设置。

4.6.2 X1振荡电路的设置例子

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

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

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

符号	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL1	EXCLKS0	OSCS ELS0	0	AMPHS10	AMPHS00	AMPH 0/1

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

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

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

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

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

④ 通过 OSTC 寄存器等待 X1 振荡电路的振荡稳定。

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

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

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

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

4.6.3 XT1振荡电路的设置例子

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

【寄存器的设置】必须按照①~⑤的顺序设置寄存器。

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

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

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

符号	7	6	5	4	3	2	1	0
CMC	EXCLK0	OSCSEL0	EXCLKS0	OSCSELS1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH0

AMPHS0 位和 AMPHS1 位：设置 XT1 振荡电路的振荡模式。

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

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

④ 必须通过软件和定时器功能等，等待副系统时钟所需的振荡稳定时间。

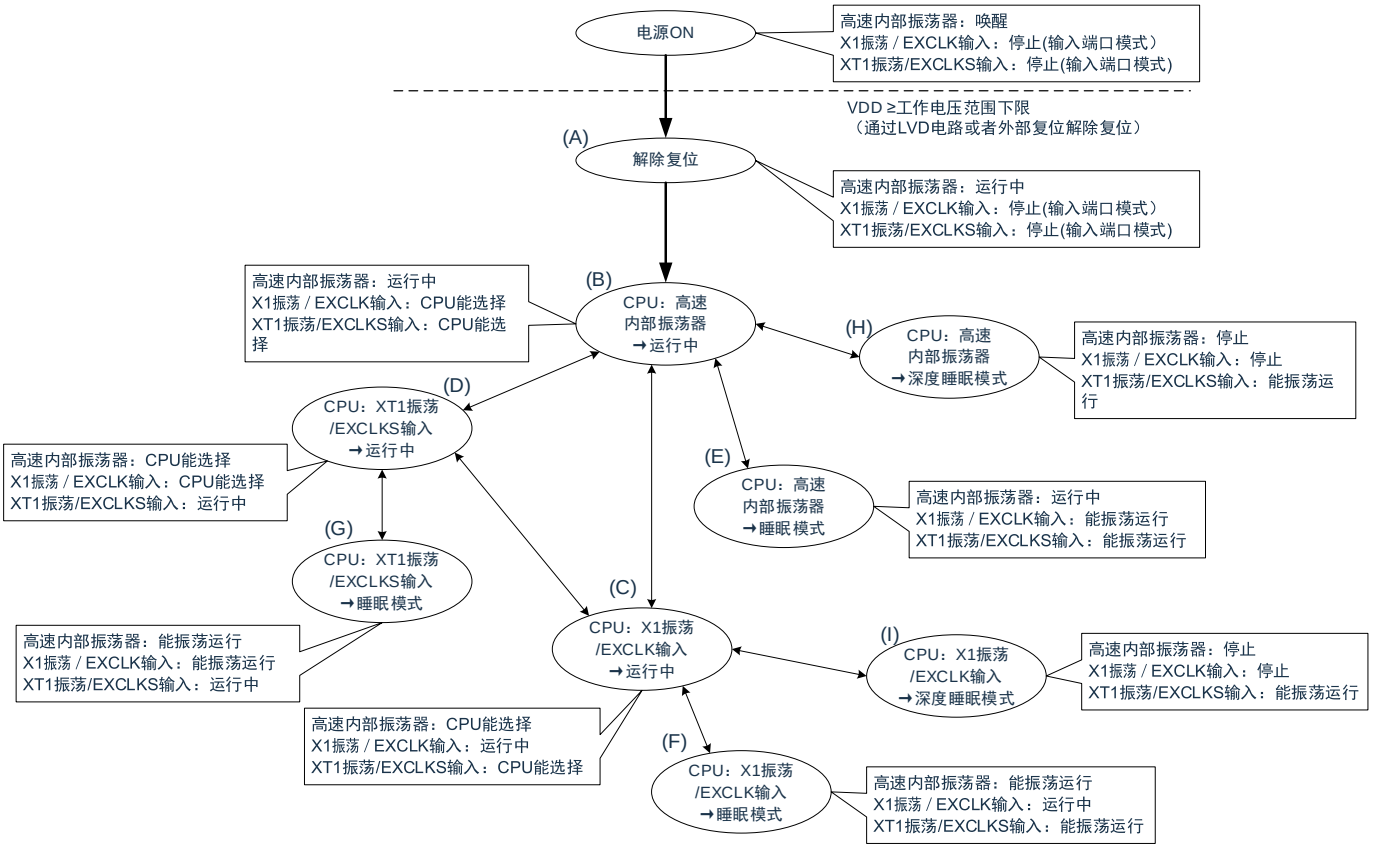
⑤ 通过 CKC 寄存器的 CSS 位将 XT1 振荡时钟设置为 CPU/外围硬件时钟。

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

4.6.4 CPU时钟的状态转移图

本产品的 CPU 时钟状态转移图如图 4-17 所示。

图 4-17：CPU 时钟的状态转移图



CPU 时钟的转移和 SFR 寄存器的设置例子等如表 4-3 所示。

表 4-3：CPU 时钟的转移和 SFR 寄存器的设置例子(1/5)

(1) 在解除复位(A)后，CPU 转移到高速内部振荡器时钟运行(B)。

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

(2) 在解除复位(A)后，CPU 转移到高速系统时钟运行(C)。

（CPU 在解除复位后立即以高速内部振荡器时钟运行(B)）

（SFR 寄存器的设置顺序）→

SFR寄存器的设置标志 / 状态转移	CMC寄存器[注1]			OSTS 寄存器	CSC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A)→(B)→(C) （X1时钟：1MHz≤F_X≤10MHz）	0	1	0	[注2]	0	需要确认	1
(A)→(B)→(C) （X1时钟：10MHz＜F_X≤20MHz）	0	1	1	[注2]	0	需要确认	1
(A)→(B)→(C) （外部主时钟）	1	1	×	[注2]	0	不需要确认	1

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

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

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

注意：必须在电源电压达到设置的时钟可运行电压（参照数据手册）后设置时钟。

(3) 在解除复位(A)后，CPU 转移到副系统时钟运行(D)。

（CPU 在解除复位后立即以高速内部振荡器时钟运行(B)）

（SFR 寄存器的设置顺序）→

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

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

备注：

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

4-3：CPU 时钟的转移和 SFR 寄存器的设置例子(2/5)

(4) CPU 从高速内部振荡器时钟运行(B)转移到高速系统时钟运行(C)。

（SFR 寄存器的设置顺序）

SFR寄存器的设置标志 / 状态转移	CMC寄存器[注1]			OSTS 寄存器	CSC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B)→(C) （X1时钟：1MHz≤F_X≤10MHz）	0	1	0	[注2]	0	需要确认	1
(B)→(C) （X1时钟：10MHz＜F_X≤20MHz）	0	1	1	[注2]	0	需要确认	1
(B)→(C) （外部主时钟）	1	1	×	[注2]	0	不需要确认	1

如果已设置就不需要。　　在高速系统时钟运行中不需要。

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

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

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

注意：必须在电源电压达到设置的时钟可运行电压（参照数据手册）后设置时钟。

(5) CPU 从高速内部振荡器时钟运行(B)转移到副系统时钟运行(D)。

（SFR 寄存器的设置顺序）

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

如果已设置就不需要。　　在副系统时钟运行中不需要。

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

备注：

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

表 4-3：CPU 时钟的转移和 SFR 寄存器的设置例子(3/5)

(6) CPU 从高速系统时钟运行(C)转移到高速内部振荡器时钟运行(B)。

（SFR 寄存器的设置顺序）→

SFR寄存器的设置标志 / 状态转移	CSC寄存器	振荡精度稳定的等待	CKC寄存器
	HIOSTOP		MCM0
(C)→(B)	0	注	0

在高速内部振荡器时钟运行中不需要。

注：FRQSEL4=0 时：45us~65us

FRQSEL4=1 时：45us~135us

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

(7) CPU 从高速系统时钟运行(C)转移到副系统时钟运行(D)。

（SFR 寄存器的设置顺序）→

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

在副系统时钟运行中不需要。

(8) CPU 从副系统时钟运行(D)转移到高速内部振荡器时钟运行(B)。

（SFR 寄存器的设置顺序）→

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

在高速内部振荡器时钟运行中不需要。

注 1：FRQSEL4=0 时：45us~65us

注 2：FRQSEL4=1 时：45us~135us

备注：

1. 表 4-3 的(A)~(I)对应图 4-17 的(A)~(I)。
2. 高速内部振荡器时钟的振荡精度稳定等待因温度条件和深度睡眠模式期间而变。

表 4-3：CPU 时钟的转移和 SFR 寄存器的设置例子(4/5)

(9) CPU 从副系统时钟运行(D)转移到高速系统时钟运行(C)。

（SFR 寄存器的设置顺序）→

SFR寄存器的设置标志 / 状态转移	OSTS 寄存器	CSC寄存器 MSTOP	OSTC寄存器	CKC寄存器 CSS
(D)→(C) （X1时钟：1MHz≤F_X≤10MHz）	注	0	需要确认	0
(D)→(C) （X1时钟：10MHz＜F_X≤20MHz）	注	0	需要确认	0
(D)→(C) （外部主时钟）	注	0	不需要确认	0

在高速系统时钟运行中不需要。

注：必须对振荡稳定时间选择寄存器（OSTS）的振荡稳定时间进行以下的设置：

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

注意：必须在电源电压达到设置的时钟可运行电压（参照数据手册）后设置时钟。

(10) CPU 在高速内部振荡器时钟运行中(B)转移到睡眠模式(E)。

CPU 在高速系统时钟运行中(C)转移到睡眠模式(F)。

CPU 在副系统时钟运行中(D)转移到睡眠模式(G)。

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

备注：表 4-3 的(A)~(I)对应图 4-17 的(A)~(I)。

表 4-3：CPU 时钟的转移和 SFR 寄存器的设置例子(5/5)

(11) CPU 在高速内部振荡器时钟运行中(B)转移到深度睡眠模式(H)。

CPU 在高速系统时钟运行中(C)转移到深度睡眠模式(I)。

（设置顺序）

<table>
<tr><th colspan="2">状态转移</th><th colspan="3">设置内容</th></tr>
<tr><td colspan="2">(B)→(H)</td><td rowspan="3">停止
不能在深度睡眠模式中运行的外围功能。</td><td>—</td><td rowspan="3">SCR寄存器bit2（SLEEPDEEP）置为1，并执行WFI指令。</td></tr>
<tr><td rowspan="2">(C)→(I)</td><td>X1振荡</td><td>设置OSTS寄存器。</td></tr>
<tr><td>外部时钟</td><td>—</td></tr>
</table>

备注：表 4-3 的(A)~(I)对应图 4-17 的(A)~(I)。

4.6.5 CPU时钟转移前的条件和转移后的处理

CPU 时钟转移前的条件和转移后的处理如下所示。

表 4-4：有关 CPU 时钟的转移(1/2)

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

表 4-4：有关 CPU 时钟的转移(2/2)

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

4.6.6 CPU时钟和主系统时钟的切换所需时间

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

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

表 4-7)。

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

如果切换 CPU 时钟，就同时切换外围硬件时钟。

表 4-5：切换主系统时钟所需要的最长时间

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

表 4-6：F_{IH}◄►F_{MX} 所需要的最大时钟数

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

表 4-7：F_{MAIN}◄► F_{SUB} 所需要的最大时钟数

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

备注：

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

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

$2F_{MX}/F_{IH}$=2(10/8)=2.5x3 个时钟

4.6.7 时钟振荡停止前的条件

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

表 4-8：时钟振荡停止前的条件和标志设置

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

4.7 高速内振校正功能

4.7.1 高速内振自调整功能

本功能以副系统时钟 F_{SX}（32.768KHz）或主系统时钟 F_{MX}（8M）为基准来测量高速内振的频率，并实时校正高速内振 F_{HOCO} 的频率精度。

表 4-9 是高速内振频率校正功能的操作规范，图 4-18 是高速内振频率校正功能的动作框图。

表 4-9：高速内振频率校正功能的操作规范

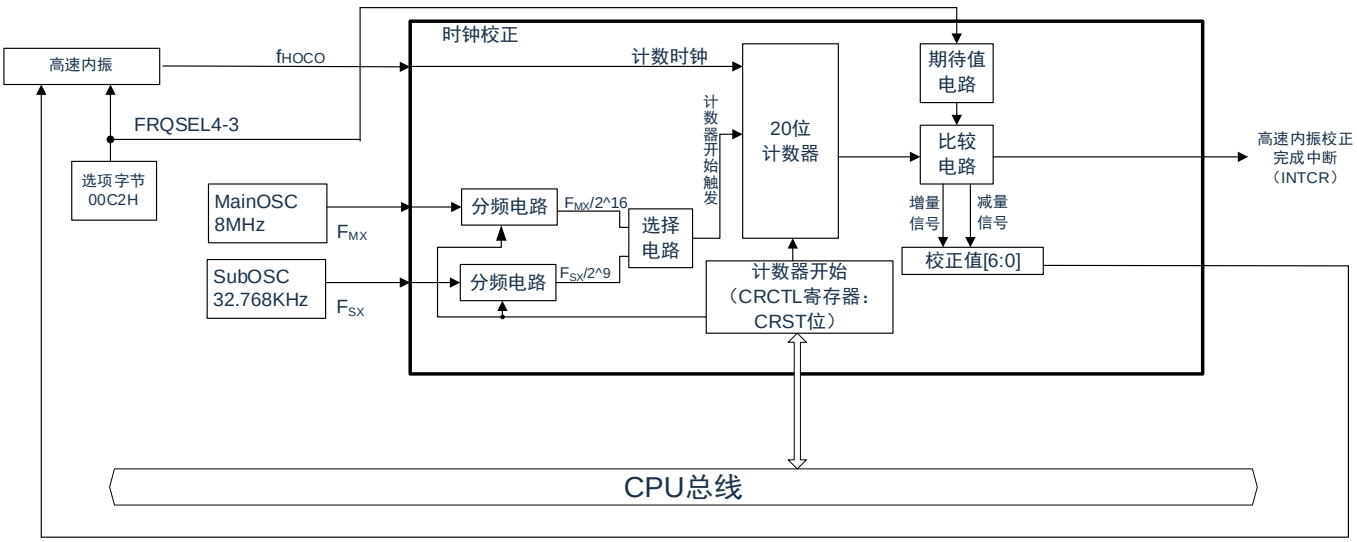
项目	内容
基准时钟	• $F_{SX}/2^9$（副系统时钟32.768KHz）或$F_{MX}/2^{16}$(主系统时钟8MHz)
校正对象时钟	• F_{HOCO}（高速内振）
动作模式	• 连续动作模式 连续进行高速内振频率校正的模式 • 间隔动作模式 使用定时器钟端等，来间隔进行高速内振频率校正的模式
时钟精度调整功能	• 校正时间： 基准时钟为$F_{SX}/2^9$：校正周期（31.25ms）X（校正次数-0.5）注 基准时钟为$F_{MX}/2^{16}$：校正周期（16.384ms）X（校正次数-0.5）注
中断	• 高速内振频率校正完成时产生中断（中断许可打开时）

注：校正时间：根据校正次数不同而变化。

校正周期：频率测定阶段和频率校正阶段的总时间。

校正次数：频率收束到期待值范围的校正次数。

图 4-18：高速内振频率校正功能的动作框图



4.7.2 寄存器说明

表 4-10 是高速内振频率校正功能使用的寄存器一览。

表 4-10：高速内振频率校正功能寄存器一览

项目	结构
控制寄存器	高速内振频率校正控制寄存器（HOCOFC）

4.7.2.1 高速内振频率校正控制寄存器（HOCOFC）

高速内振频率校正功能的控制寄存器。

通过 32 位存储器操作指令设定 HOCOFC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-19：高速内振频率校正控制寄存器（HOCOFC）的格式

地址：0x40022400　　复位后：00H

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

FCMD[注1]	高速内振频率校正功能动作模式
0	连续动作模式
1	间隔动作模式

FCREF	高速内振频率校正基准时钟选择
0	高速内振频率校正选择副系统时钟F_{SX}作为基准
1	高速内振频率校正选择主系统时钟F_{MX}作为基准

FCEND[注2]	高速内振频率校正完成标志
0	高速内振频率校正未完成
1	高速内振频率校正完成

FCST[注3]	高速内振频率校正电路动作控制/状态
0	高速内振频率校正电路动作停止/停止中
1	高速内振频率校正电路动作开始/动作中
连续动作模式时，软件写0来停止动作。 间隔动作模式时，校正完成后，硬件将FCST位清除。	

注1：FCST位为1时，禁止改写FCMD位。

注2：FCEND为高速内振频率校正完成的标志位，高速内振频率校正完成时自动置1，通过给寄存器写0来清零。

注 3：给 FCST 位写 1 时，先确认当前 FCST 位的值为 0 再给它写 1。由于硬件清除优先，间隔动作完成后立刻（高速内振频率校正完成中断产生时）给 FCST 位写 1 时，应在高速内振频率校正完成中断产生后至少经过 F_{HOCO} 1 个周期再执行该操作。给 FCST 位写 0（高速内振频率校正电路动作停止）后，F_{HOCO} 2 个周期内禁止给 FCST 位写 1（高速内振频率校正电路动作开始）。

注意：位 6，3~1 必须写 0.

4.7.3 动作说明

4.7.3.1 动作概要

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

表 4-11 是高速内振输入频率和校正周期，图 4-20 是高速内振频率校正动作时序图（详细）。

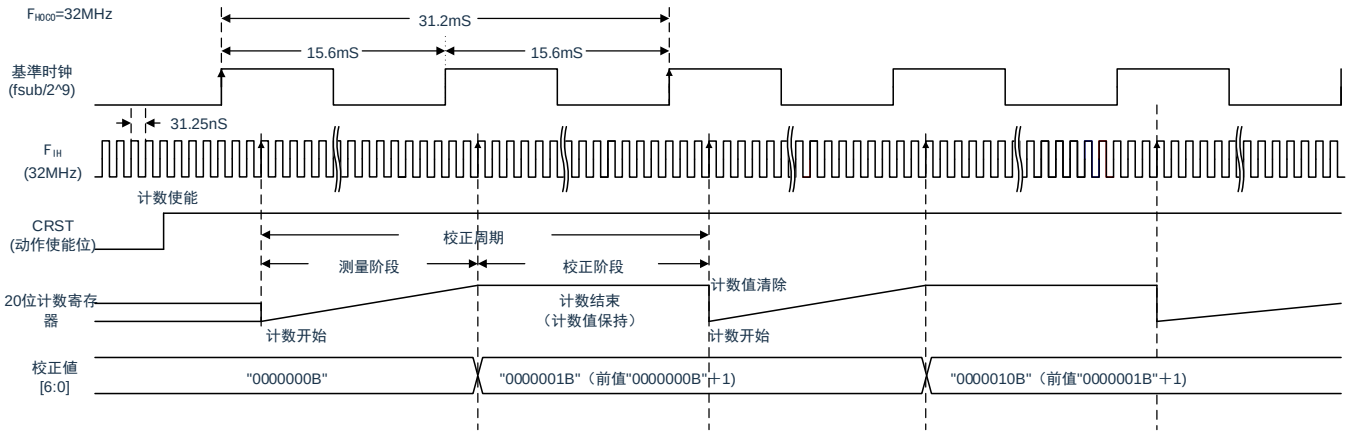
表 4-11：高速内振输入频率和校正周期

F_{HOCO}（MHz）	校正周期（ms）
32	F_{SX} 作为基准：31.25 F_{MX} 作为基准：16.384 （频率测量阶段+频率校正阶段）

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

本节后续的动作说明均以副系统时钟（F_{SX}）作为基准时钟，若要以 F_{MX} 作为基准作为基准时钟，只需将 HOCOFC 寄存器的 bit5（FCREF）设置为 1。

图 4-20：高速内振频率校正动作时序图（详细）



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

(1) 连续动作模式

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

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

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

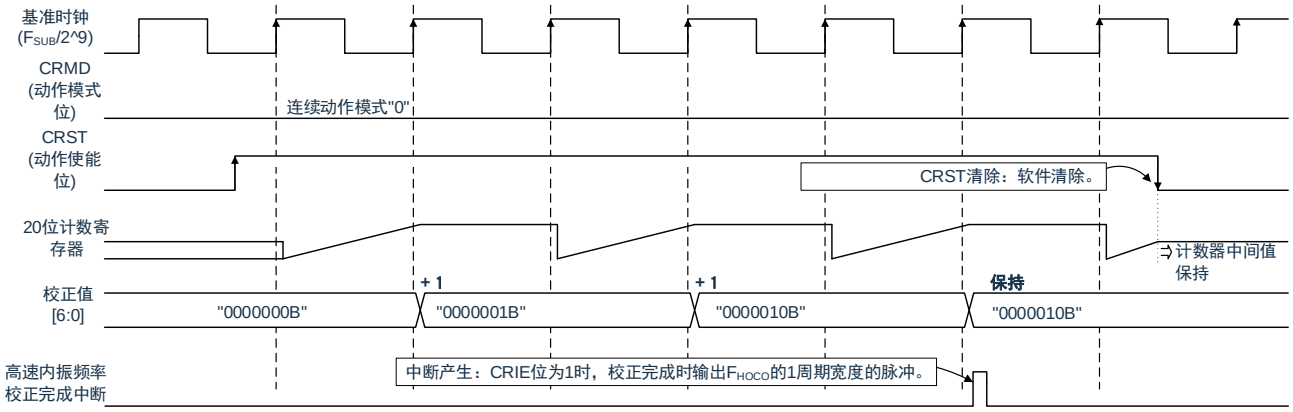
然后，将计数值和期待值进行比较，根据以下所述进行校正值调整。（频率校正阶段）

- 计数值比期待值大时：校正值-1
- 计数值比期待值小时：校正值+1
- 计数值在期待值范围内时：校正值保持（高速内振时钟频率校正结束）

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

图 4-21 是连续动作模式的时序图。

图 4-21：连续动作模式时序图



(2) 间隔动作模式

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

HOCOFC寄存器的FCST位设置为1时高速内振时钟频率校正动作开始。

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

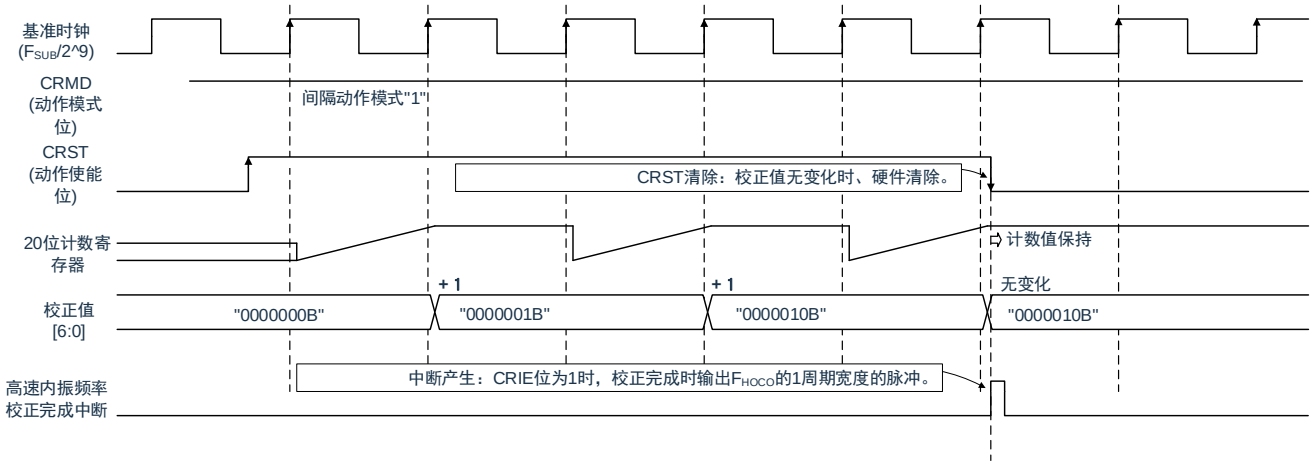
然后，将计数值和期待值进行比较，根据以下所述进行校正值调整。（频率校正阶段）

- 计数值比期待值大时：校正值-1
- 计数值比期待值小时：校正值+1
- 计数值在期待值范围内时：校正值保持（高速内振时钟频率校正结束）

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

图 4-22 是连续动作模式的时序图。

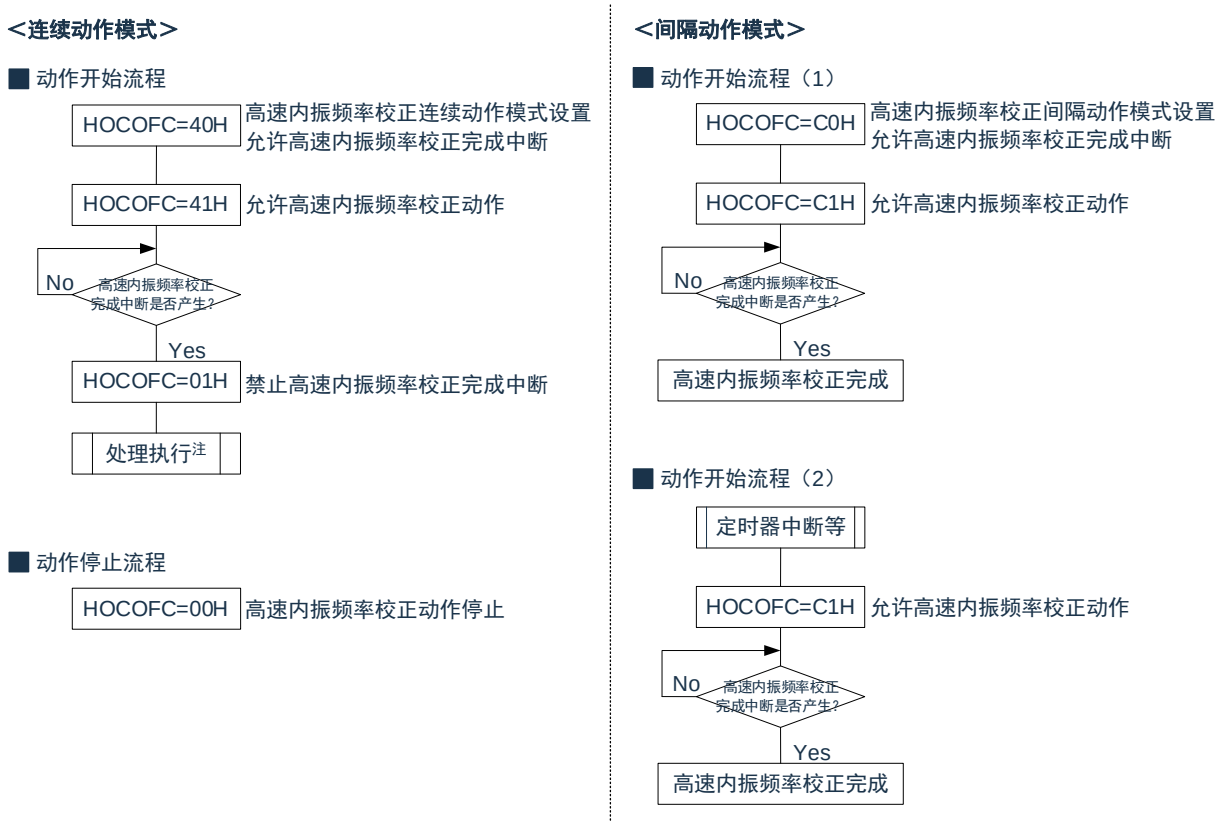
图 4-22：间隔动作模式时序图



4.7.3.2 动作设置流程

高速内振时钟频率校正功能使用时的动作开始/停止流程如下图所示。

图 4-23：动作模式设定流程（例）



注：在停止高速内振时钟频率校正功能前，高速内振时钟频率校正动作反复执行。

4.7.4 使用注意事项

4.7.4.1 SFR访问

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

4.7.4.2 复位时动作

进入深度睡眠之前，必须停止高速内振时钟频率校正功能。

4.8 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（F_{MX}）或者副系统时钟（F_{SX}）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╳((OSDCCMP11~OSDCCMP0 设置值)+1)

以内部低速振荡时钟（F_{IL}）频率为 16.384K 为例：

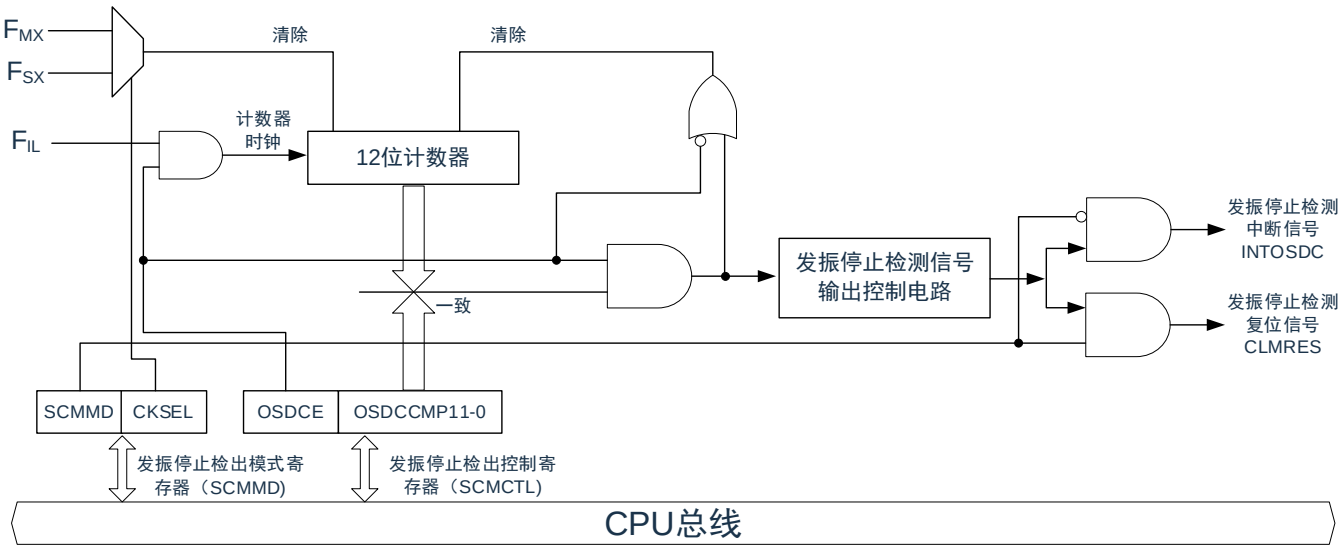
OSDCCMP11~OSDCCMP0=003H 时：212us(MIN.),244us(TYP.),287us(MAX.)

OSDCCMP11~OSDCCMP0=FFFH 时：217ms(MIN.),250ms(TYP.),294ms(MAX.)

4.8.1 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 4-24：发振停止检测电路的框图（例）



4.8.2 发振停止检测电路使用的寄存器

4.8.2.1 外围允许寄存寄存器1（PER1）

使用发振停止检测电路时，必须要将 PER1 的 bit2（OSDCEN）设置为 1。

寄存器详细描述见“4.3.6 外围允许寄存器 0、1（PER0、PER1）”。

4.8.2.2 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。

图 4-25：发振停止检测控制寄存器（SCMCTL）的格式

地址：0x40022200　　复位后：0FFFH　R/W

符号	15	14	13	12	11	10	9	8
SCMCTL	OSCDE	0	0	0	OSDCCMP11-8			

符号	7	6	5	4	3	2	1	0
SCMCTL	OSDCCMP7-0							

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

OSDCCMP11-0	发振停止判定时间
000H … 002H	禁止设置
003H … FFFH	设置发振停止判定时间。 发振停止判定时间=内部低速振荡时钟（fIL）周期╳((OSDCCMP11~OSDCCMP0设置值)+1)

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。

4.8.2.3 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（F_{MX}）或者副系统时钟（F_{SX}），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。

图 4-26：发振停止检测模式寄存器（SCMMD）的格式

地址：0x40022202　　复位后：0000H　R/W

符号	15	14	13	12	11	10	9	8
SCMMD	KEY[注]							

符号	7	6	5	4	3	2	1	0
SCMMD	0	0	0	0	0	0	MDSEL	CKSEL

CKSEL	发振停止检测的对象
0	检测主系统时钟（F_{MX}）的发振状态
1	检测副系统时钟（F_{SX}）的发振状态

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

注：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。

4.8.2.4 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。

图 4-27：发振停止检测状态寄存器（SCMST）的格式

地址：0x40022204　　复位后：000H　R/W [注]

符号	7	6	5	4	3	2	1	0
SCMST	0	0	0	0	0	0	0	OSTDF

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

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。

4.8.3 发振停止检测电路的动作

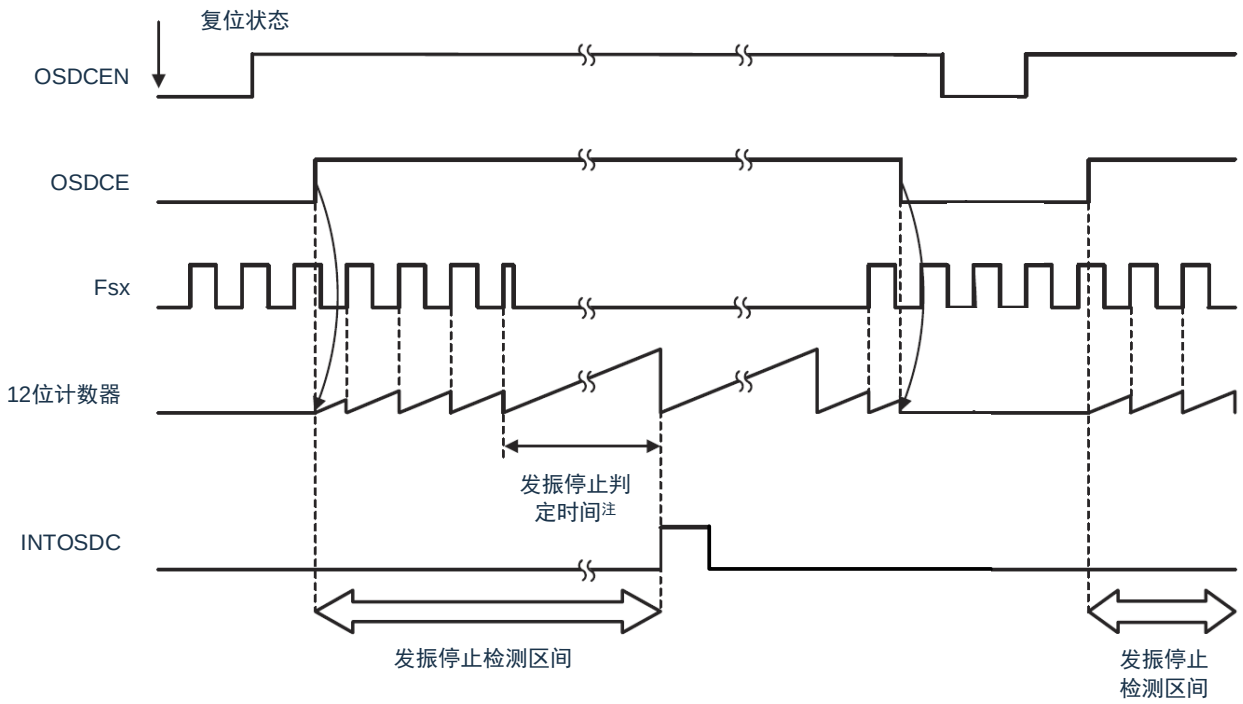
4.8.3.1 发振停止检测电路的动作方法

1. 外部复位解除后，主系统时钟（F_{MX}）/副系统时钟（F_{SX}）开始发振。
2. 写外围使能寄存器（PER2），使能发振停止检测电路[注]。
3. 写发振停止检测模式寄存器（SCMMD），选择检测主系统时钟（F_{MX}）或副系统时钟（F_{SX}）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。
4. 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。
5. 发振停止检测电路动作过程中，主系统时钟（F_{MX}）/副系统时钟（F_{SX}）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。
6. 当CPU时钟为主系统时钟（F_{MX}）且有主系统时钟（F_{MX}）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（F_{HOCO}/8），当CPU时钟为副系统时钟（F_{SX}）且有副系统时钟（F_{SX}）发振停止检出时，CPU时钟会切换到内部低速振荡时钟（F_{IL}）。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PER2的bit4（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PER2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23章 中断功能”。

图 4-28：发振停止检测电路的时序

（以检测对象为 F_{SX}，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（F_{IL}）周期╳((OSDCCMP11~OSDCCMP0 设置值)+1)。

4.8.4 深度睡眠模式下发振停止检测电路的动作

若进入深度睡眠之前，发振停止检测电路是使能状态，进入深度睡眠后会自动关闭发振停止检测功能，待待机解除信号来了以后，再重新打开发振停止检测功能。

关于待机解除信号，请参考“待机功能”

4.8.5 发振停止检测功能的注意事项

发振停止检测电路，跟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 通用定时器单元Timer8

本制品搭载一个通用定时器单元 Timer8，含有 8 个通道。通用定时器单元的通道数因产品而不同。

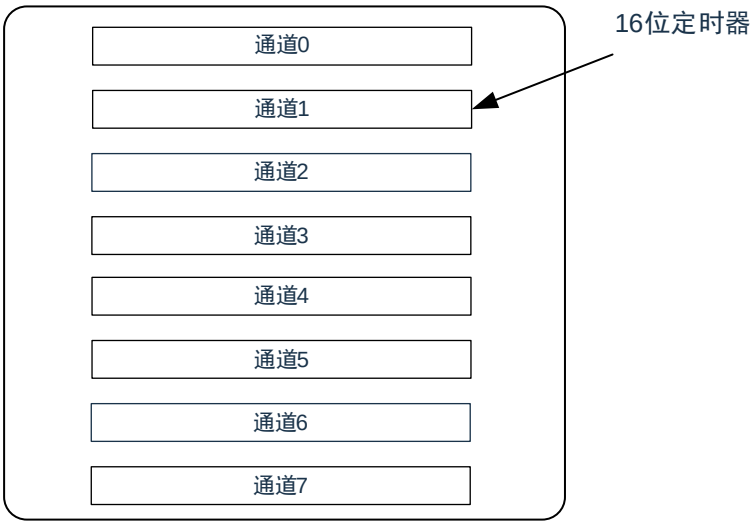
说明：

1. 本章下文中的标号“m”代表单元号，本产品搭载了一个通用定时器Timer8，所以m=0。
2. 本章下文中的标号“n”代表通道号 （在本章中，n=0~7），各通道的定时器输入/输出引脚的有无因产品 而不同。详细内容请参照“第2章 端口功能”。

通用定时器单元 Timer8 有 8 个 16 位定时器。

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

通用定时器单元 Timer8



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5.8.1） • 方波输出（参照5.8.1） • 外部事件计数器（参照5.8.2） • 分频器（参照5.8.3） • 输入脉冲间隔的测量（参照5.8.4） • 输入信号的高低电平宽度的测量（参照5.8.5） • 延迟计数器（参照5.8.6）	• 单触发脉冲输出（参照5.9.1） • PWM 输出（参照5.9.2） • 多重PWM输出（参照5.9.3）

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

5.1 通用定时器单元的功能

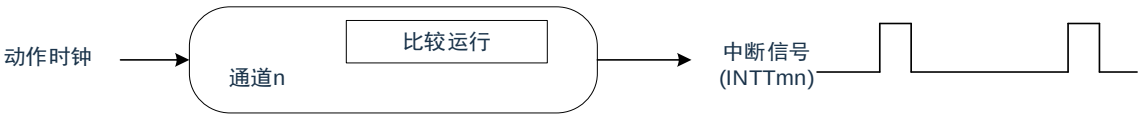
通用定时器单元有以下功能：

5.1.1 独立通道运行功能

独立通道运行功能是不受其他通道运行模式的影响而能独立使用任意通道的功能。

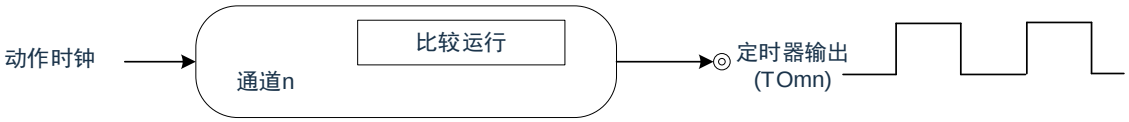
(1) 间隔定时器

能用作以固定间隔产生中断（INTTMmn）的基准定时器。



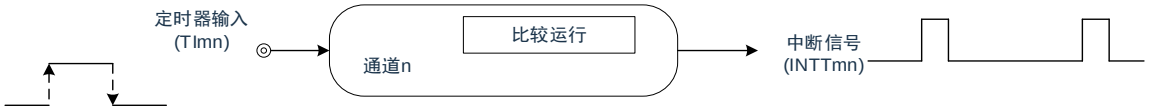
(2) 方波输出

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



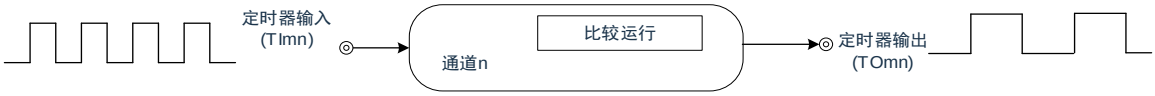
(3) 外部事件计数器

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



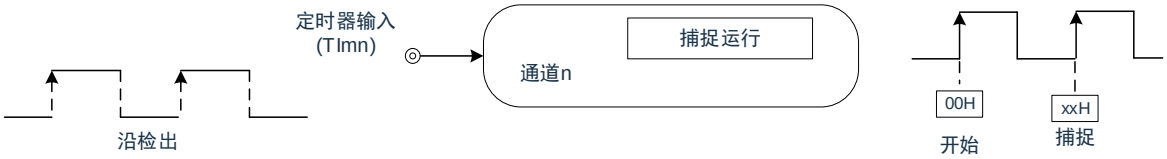
(4) 分频器功能（只限于通道 0）

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



(5) 输入脉冲间隔的测量

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

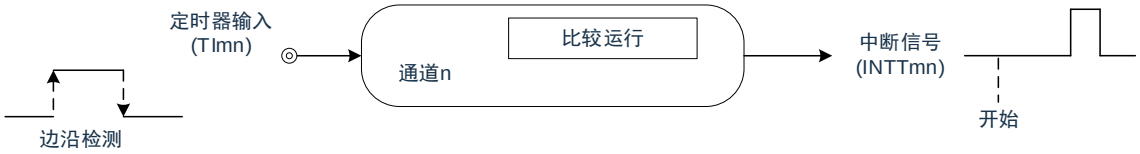


(6) 输入信号的高低电平宽度的测量

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

(7) 延迟计数器

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



备注：

1. m：单元号（m=0） n：通道号（n=0~7）
2. 各通道的定时器输入/输出引脚的有无因产品而不同。

5.1.2 多通道联动运行功能

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

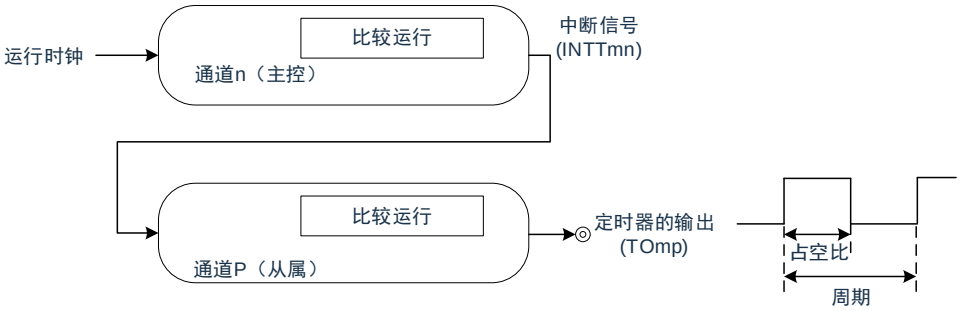
多通道联动运行功能可用作以下模式。

(1) 单触发脉冲输出

将 2 个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。

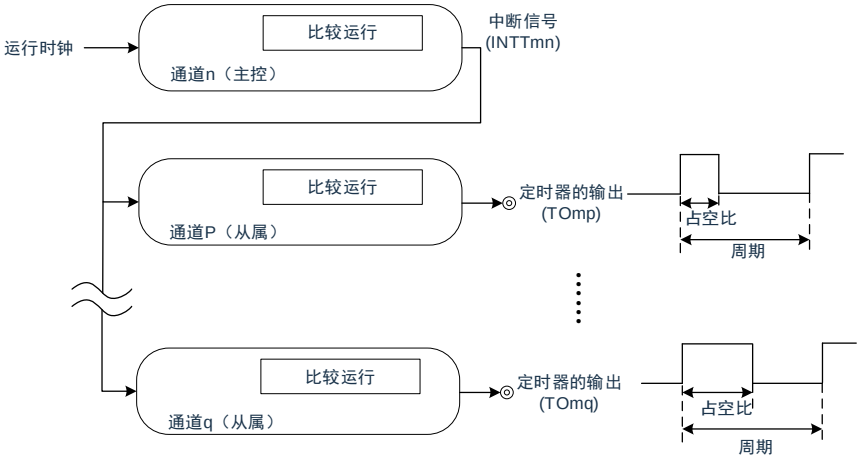
(2) PWM（Pulse Width Modulation）输出

将 2 个通道成对使用，生成能任意设定周期和占空比的脉冲。



(3) 多重 PWM（Pulse Width Modulation）输出

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



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

备注：m：单元号（m=0） n：通道号（n=0~7）

p、q：从属通道号（n＜p＜q≤7）

5.1.3 LIN-bus支持功能（只限于通道3）

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

(1) 唤醒信号的检测

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

(2) 间隔段的检测

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

(3) 同步段脉宽的测量

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

备注：有关LIN-bus 支持功能的运行设定，请参照“5.3.13：输入切换控制寄存器（ISC）” 和“5.8.5：作为输入信号高低电平宽度测量的运行”。

5.2 通用定时器单元的结构

通用定时器单元由以下硬件构成。

表 5-1：通用定时器单元的结构

项目	结构
计数器	定时器计数寄存器mn（TCRmn）
寄存器	定时器数据寄存器mn（TDRmn）
定时器的输入	TI00~TI07[注1]、RxD0引脚（用于LIN-bus）
定时器的输出	TO00~TO07[注1]、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0（TIS0） • 定时器输出允许寄存器m（TOEm） • 定时器输出寄存器m（TOm） • 定时器输出电平寄存器m（TOLm） • 定时器输出模式寄存器m（TOMm）
	＜每个通道的寄存器＞ • 定时器模式寄存器mn（TMRmn） • 定时器状态寄存器mn（TSRmn） • 输入切换控制寄存器（ISC） • 噪声滤波器允许寄存器（NFEN1） • 端口模式控制寄存器（PMCxx）[注2] • 端口模式寄存器（PMxx）[注2] • 端口寄存器（Pxx）[注2]

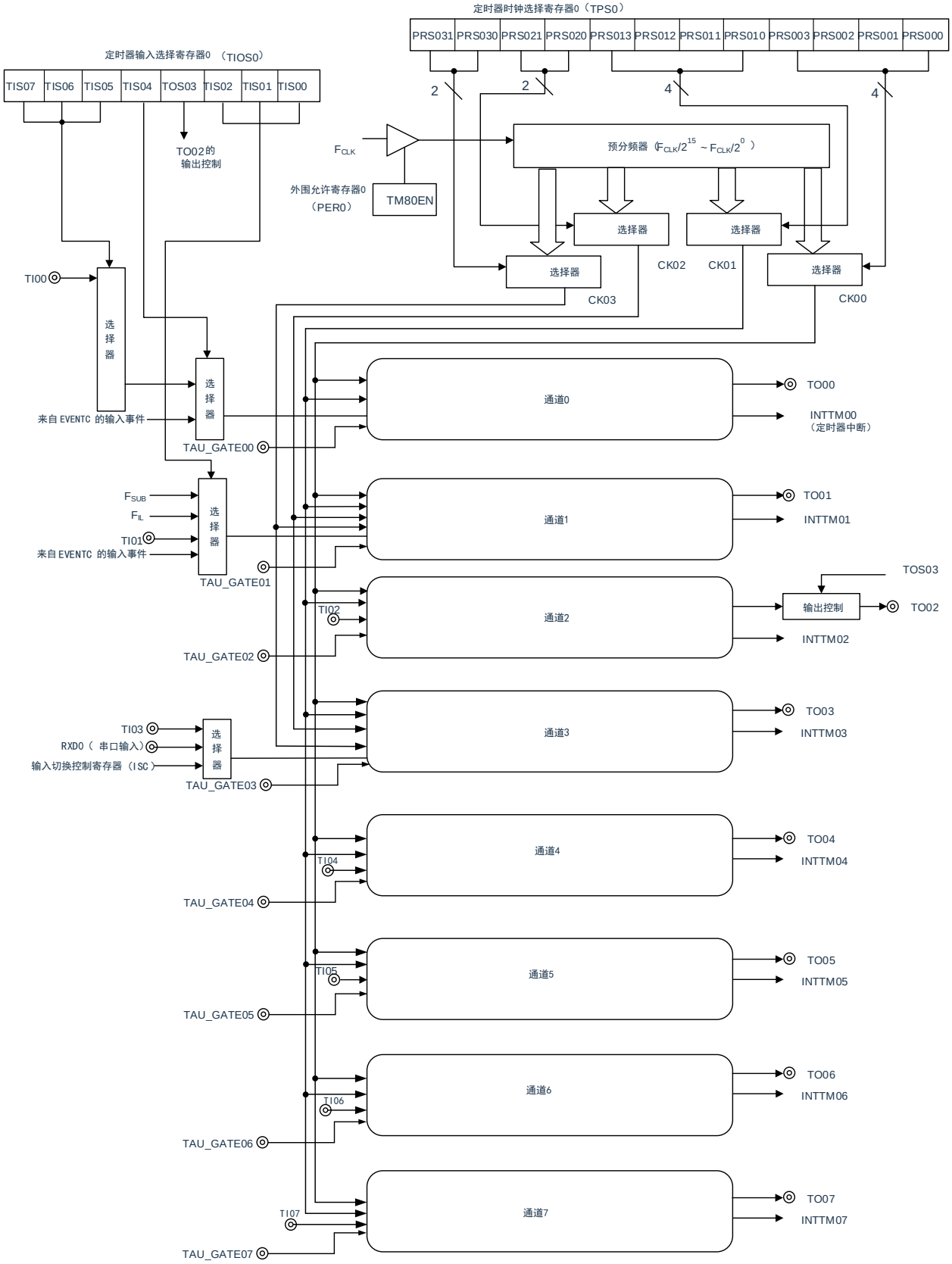
注 1：各通道的定时器输入/输出引脚的有无因产品而不同。

注 2：设定的端口模式控制寄存器（PMCxx）、端口模式寄存器（PMxx）和端口寄存器（Pxx）因产品而不同。详细内容请参照 “”。

备注：m：单元号（m=0） n：通道号（n=0~7）

通用定时器单元的框图如图 5-1 所示。

图 5-1：通用定时器的整体框图



备注：F_{SUB}：副系统时钟频率

F_{IL}：低速内部振荡器时钟频率

5.2.1 通用定时器单元寄存器列表

寄存器基址：0x40043000

偏移地址	寄存器名	读写属性	位宽	复位值
0x000	TCR00	R	16	FFFFH
0x002	TCR01	R	16	FFFFH
0x004	TCR02	R	16	FFFFH
0x006	TCR03	R	16	FFFFH
0x008	TCR04	R	16	FFFFH
0x00A	TCR05	R	16	FFFFH
0x00C	TCR06	R	16	FFFFH
0x00E	TCR07	R	16	FFFFH
0x010	TMR00	R/W	16	0000H
0x012	TMR01	R/W	16	0000H
0x014	TMR02	R/W	16	0000H
0x016	TMR03	R/W	16	0000H
0x018	TMR04	R/W	16	0000H
0x01A	TMR05	R/W	16	0000H
0x01C	TMR06	R/W	16	0000H
0x01E	TMR07	R/W	16	0000H
0x020	TSR00	R	16	0000H
0x022	TSR01	R	16	0000H
0x024	TSR02	R	16	0000H
0x026	TSR03	R	16	0000H
0x028	TSR04	R	16	0000H
0x02A	TSR05	R	16	0000H
0x02C	TSR06	R	16	0000H
0x02E	TSR07	R	16	0000H
0x030	TE0	R	16	0000H
0x032	TS0	R/W	16	0000H
0x034	TT0	R/W	16	0000H
0x036	TPS0	R/W	16	0000H
0x038	TO0	R/W	16	0000H
0x03A	TOE0	R/W	16	0000H
0x03C	TOL0	R/W	16	0000H
0x03E	TOM0	R/W	16	0000H
0x040	TDR00	R/W	16	0000H
0x042	TDR01	R/W	16	0000H
0x044	TDR02	R/W	16	0000H
0x046	TDR03	R/W	16	0000H
0x048	TDR04	R/W	16	0000H
0x04A	TDR05	R/W	16	0000H
0x04C	TDR06	R/W	16	0000H
0x04E	TDR07	R/W	16	0000H

5.2.2 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“5.3.3：定时器模式寄存器 mn（TMRmn）”）。

图 5-2：定时器计数寄存器 mn（TCRmn）的表格

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

备注：m：单元号（m=0） n：通道号（n=0~7）

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（PER0）的 TM80EN 位时
- 在 PWM 输出模式中从属通道的计数结束时
- 在延迟计数模式中从属通道的计数结束时
- 在单触发脉冲输出模式中主控/从属通道的计数结束时
- 在多重 PWM 输出模式中从属通道的计数结束时

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

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

如下所示，TCRmn 寄存器的读取值因运行模式和运行状态而不同。

表 5-2：各运行模式中的定时器计数寄存器 mn（TCRmn）的读取值

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

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

备注：m：单元号（m=0） n：通道号（n=0~7）

5.2.3 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。

在产生复位信号后，TDRmn 寄存器的值变为“0000H”。

图 5-3：定时器数据寄存器 mn（TDRmn）（n=0~7）的表格

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

(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。

注意：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。

备注：m：单元号（m=0） n：通道号（n=0~7）

5.3 控制通用定时器单元的寄存器

控制通用定时器单元的寄存器如下所示：

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

注意：分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

备注：m：单元号（m= 0） n：通道号（n=0~7）

5.3.1 外围允许寄存器0（PER0）

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

要使用通用定时器时，必须将 bit0（TM80EN）置“1”。

通过 32 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“00000000H”。

图 5-4：外围允许寄存器 0（PER0）的表格

地址：40020420H　　复位后：00000000H　　R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN注	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

TM80EN	通用定时器的输入时钟的控制
0	停止提供输入时钟。 • 不能写通用定时器使用的SFR。 • 通用定时器处于复位状态。
1	提供输入时钟。 • 能读写通用定时器使用的SFR。

注意：要设定通用定时器单元时，必须先在 TM80EN 位为“1”的状态下设定以下的寄存器。当 TM80EN 位为“0”时，定时器阵列单元的控制寄存器的值为初始值，忽视写操作（定时器输入输出选择寄存器 0（TIOS0）、输入切换控制寄存器（ISC）、噪声滤波器允许寄存器（NFEN1）、端口模式控制寄存器 PMCx、端口模式寄存器 PMx 和端口寄存器 Px 除外）。

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

5.3.2 定时器时钟选择寄存器m（TPSm）

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

只有在以下情况下才能改写定时器运行中的TPSm 寄存器。

1) 能改写 PRSm00~PRSm03 位的情况（n=0~7）：
   选择 CKm0 作为运行时钟（CKSmn1、CKSmn0=0、0）的通道全部处于停止状态（TEmn=0）。
2) 能改写 PRSm10~PRSm13 位的情况（n=0~7）：
   选择 CKm2 作为运行时钟（CKSmn1、CKSmn0=0、1）的通道全部处于停止状态（TEmn=0）。
3) 能改写 PRSm20 位和 PRSm21 位的情况（n=1、3）：
   选择 CKm1 作为运行时钟（CKSmn1、CKSmn0=1、0）的通道全部处于停止状态（TEmn=0）。
4) 能改写 PRSm30 位和 PRSm31 位的情况（n=1、3）：
   选择 CKm3 作为运行时钟（CKSmn1、CKSmn0=1、1）的通道全部处于停止状态（TEmn=0）。

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

图 5-5：定时器时钟选择寄存器 m（TPSm）的表格(1/2)

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

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

注：在更改选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）的情况下，必须停止通用定时器单元（TTm=0,100FH）。即使在选择运行时钟（F_{MCK}）或者 TImn 引脚输入信号的有效边沿时，也需要停止通用定时器单元。

注意：

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

备注：

1. F_{CLK}：CPU/外围硬件的时钟频率
2. TPSm 寄存器选择的时钟波形从上升沿开始只有 1 个 F_{CLK} 周期为高电平（m=1~15）。详细内容请参照“5.5.1 计数时钟（F_{TCLK}）”。

图 5-6：定时器时钟选择寄存器 m（TPSm）的表格(2/2)

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

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

PRS m31	PRS m30	运行时钟（CKm3）的选择注					
			F_{CLK}=2MHz	F_{CLK}=4MHz	F_{CLK}=8MHz	F_{CLK}=20MHz	F_{CLK}=32MHz
0	0	$F_{CLK}/2^8$	7.81KHz	15.6KHz	31.3KHz	78.1KHz	125KHz
0	1	$F_{CLK}/2^{10}$	1.95KHz	3.91KHz	7.81KHz	19.5KHz	31.3KHz
1	0	$F_{CLK}/2^{12}$	488Hz	977Hz	1.95KHz	4.88KHz	7.81KHz
1	1	$F_{CLK}/2^{14}$	122Hz	244Hz	488Hz	1.22KHz	1.95KHz

注：在更改选择为 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=0,100FH）。即使在选择运行时钟（F_{MCK}）或者 TImn 引脚输入信号的有效边沿时，也需要停止通用定时器单元。

注意：必须将 bit15、14、11、10 置“0”。

5.3.3 定时器模式寄存器mn（TMRmn）

TMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟（F_{MCK}）的选择、计数时钟的选择、主控/从属的选择、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“5.8 通用定时器单元的独立通道运行功能”和“5.9 定时器阵列单元的多通道联动运行功能”）。

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

注意：TMRmn 寄存器的 bit11 因通道而不同。

TMRm2、TMRm4、TMRm6：MASTERmn 位（n=2、4、6）

TMRm0、TMRm5、TMRm7：固定为“0”。

图 5-7：定时器模式寄存器 mn（TMRmn）的表格(1/4)

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

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

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

CCSmn	通道n计数时钟（F_{TCLK}）的选择
0	CKSmn0位和CKSmn1位指定的运行时钟（F_{MCK}）
1	TImn引脚输入信号的有效边沿 • 单元0的情况： 通道0：TIS0选择的输入信号的有效边沿 通道1：TIS0选择的输入信号的有效边沿 通道3：ISC选择的输入信号的有效边沿
计数时钟（F_{TCLK}）用于计数器、输出控制电路和中断控制电路。	

注：bit11 是只读位，固定为“0”，忽视写操作。

注意：

1. 必须将 bit13 置 “0”。
2. 要更改选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，即使选择了 CKSmn0 位和 CKSmn 位指定的运行时钟（F_{MCK}）或者 TImn 引脚输入信号的有效边沿作为计数时钟（F_{TCLK}），也必须停止定时器阵列单元（TTm=0,10FFH）。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-8：定时器模式寄存器 mn（TMRmn）的表格(2/4)

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

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

（TMRmn（n=2,4,6）的 bit11）

MASTERmn	通道n的独立通道运行/多通道联动运行（从属或者主控）的选择
0	用作独立通道运行功能或者多通道联动运行功能的从属通道。
1	用作多通道联动运行功能的主控通道。
只能将通道2,4,6设定为主控通道（MASTERmn=1）。 通道0固定为“0”（因为通道 0 为最高位的通道，所以与此位的设定无关，用作主控通道）。 对于用作独立通道运行功能的通道，将 MASTERmn位置“0”。	

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

注 1：bit11 是只读位，固定为“0”，忽视写操作。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-9：定时器模式寄存器 mn（TMRmn）的表格(3/4)

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

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

CISmn1	CISmn0	TImn引脚的有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	双边沿（测量低电平宽度时） 开始触发：下降沿，捕捉 触发：上升沿
1	1	双边沿（测量高电平宽度时） 开始触发：上升沿，捕捉 触发：下降沿
当STSmn2~STSmn0位不为“010B”并且使用双边沿指定时，必须将CISmn1~CISmn0位置“10B”。		

GTSmn1	GTSmn0	计数时钟源启动的设定
0	0	门控信号无效
0	1	门控信号为高 计数时钟源启动
1	0	门控信号为低 计数时钟源启动
1	1	门控信号无效

注 1：bit11 是只读位，固定为“0”，忽视写操作。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-10：定时器模式寄存器 mn（TMRmn）的表格(4/4)

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

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

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

运行模式（MDmn3~MDmn1位的设定（参照上表））	MD mn0	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2（1、0、0）	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效注3。此时不产生中断。
• 捕捉&单次计数模式（1、1、0）	0	在开始计数时不产生定时器中断（定时器的输出也不发生变化）。 计数运行中的开始触发无效。 此时不产生中断。

注 1：bit11 是只读位，固定为“0”，忽视写操作。

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

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

备注：m：单元号（m=0） n：通道号（n=0~7）

5.3.4 定时器状态寄存器mn（TSRmn）

TSRmn 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSRmn 寄存器只在捕捉模式（MDmn3~MDmn1=010B）和捕捉&单次计数模式（MDmn3~MDmn1=110B 中有效。有关各运行模式中的 OVF 位的变化和置位/清除条件，请参照表 5-5。

通过 16 位存储器操作指令读 TSRmn 寄存器。

图 5-11：定时器状态寄存器 mn（TSRmn）的表格

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

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

备注：m：单元号（m=0）n：通道号（n=0~7）

表 5-3：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 • 捕捉&单次计数模式	清除	在捕捉时没有发生上溢的情况
	置位	在捕捉时发生上溢的情况
• 间隔定时器模式 • 事件计数器模式 • 单次计数模式	清除	— （不能使用）
	置位	

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

5.3.5 定时器通道允许状态寄存器m（TEm）

TEm 寄存器是表示各通道定时器运行的允许或者停止状态的寄存器。

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

通过 16 位存储器操作指令读 TEm 寄存器。

图 5-12：定时器通道允许状态寄存器 m（TEm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	0	0	0	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

备注：m=0

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

备注：m：单元号（m=0）n：通道号（n=0~7）

5.3.6 定时器通道开始寄存器m（TSm）

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

通过 16 位存储器操作指令设置 TSm 寄存器。

图 5-13：定时器通道开始寄存器 m（TSm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	0	0	0	0	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

备注：m=0

TSmn	通道n的运行允许（开始）触发
0	没有触发。
1	将TEmn位置“1”，进入计数允许状态。计数允许状态下的TCRmn寄存器的计数开始因各运行模式而不同（参照“5.5.2计数器的开始时序”的表5-6）。

注意：

1. 必须将 bit15~8 置“0”。
2. 在从不使用 TImn 引脚输入的功能切换到使用 TImn 引脚输入的功能时，从设定时器模式寄存器 mn（TMRmn）到将 TSmn 位置“1”为止，需要以下期间的等待：

   TImn 引脚噪声滤波器有效时（TNFENmn=1）：4 个运行时钟（F_{MCK}）

   TImn 引脚噪声滤波器无效时（TNFENmn=0）：2 个运行时钟（F_{MCK}）

备注：TSm 寄存器的读取值总是“0”。

m：单元号（m=0）n：通道号（n=0~7）

5.3.7 定时器通道停止寄存器m（TTm）

TTm 寄存器是设定各通道计数停止的触发寄存器。

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

通过 16 位存储器操作指令设定 TTm 寄存器。

图 5-14：定时器通道停止寄存器 m（TTm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	0	0	0	0	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0

备注：m=0

TTmn	通道n的运行停止触发
0	没有触发。
1	将TEmn位清“0”，进入计数停止状态。

注意：必须将 bit15~8 置“0”。

备注：

1. TTm 寄存器的读取值总是“0”。
2. m：单元号（m=0）n：通道号（n=0~7）

5.3.8 定时器输入输出选择寄存器（TIOS0）

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

图 5-15：定时器输入选择寄存器 0（TIOS0）的表格

地址：0x40040C04H　　复位后：00H　　R/W

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

TIS07	TIS06	TIS05	通道0使用的定时器输入的选择
0	0	0	定时器输入引脚（TI00）的输入信号
其他			设定禁止

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

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

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

注意：

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

5.3.9 定时器输出允许寄存器m（TOEm）

TOEm 寄存器是设定允许或者禁止各通道定时器输出的寄存器。

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

通过 16 位存储器操作指令设定 TOEm 寄存器。

图 5-16：定时器输出允许寄存器 m（TOEm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

备注：m=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。

注意：必须将 bit15~8 置“0”。

备注：m：单元号（m=0）n：通道号（n=0~7）

5.3.10 定时器输出寄存器m（TOm）

TOm 寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚（TOmn）输出。

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

要将 TOmn 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。

图 5-17：定时器输出寄存器 m（TOm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	TOm7	TOm6	TOm5	TOm4	TOm3	TOm2	TOm1	TOm0

备注：m=0

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

注意：必须将 bit15~8 置“0”。

备注：m：单元号（m=0）n：通道号（n=0~7）

5.3.11 定时器输出电平寄存器m（TOLm）

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

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

通过 16 位存储器操作指令设定 TOLm 寄存器。

图 5-18：定时器输出电平寄存器 m（TOLm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0

备注：m=0

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

注意：必须将 bit15~8 和 bit0 置“0”。

备注：

1. 如果在定时器运行中改写此寄存器的值，就在下一次定时器输出信号发生变化时反相定时器的输出逻辑，而不是在改写后立即反相。
2. m：单元号（m=0）n：通道号（n=0~7）

5.3.12 定时器输出模式寄存器m（TOMm）

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

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

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

通过 16 位存储器操作指令设定 TOMm 寄存器。

图 5-19：定时器输出模式寄存器 m（TOMm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	TOMm7	TOMm6	TOMm5	TOMm4	TOMm3	TOMm2	TOMm1	0

备注：m=0

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

注意：必须将 bit15~8 和 bit0 置“0”。

备注：m：单元号（m=0）；

n：通道号（n=0~7）

主控通道号：n=0、2、4、6

从属通道号 p：n＜p≤7

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

5.3.13 输入切换控制寄存器（ISC）

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

有关 SSIE00 位的设定，请参照“输入切换控制寄存器（ISC）”。

在产生复位信号后，ISC 寄存器的值变为“00H”。

图 5-20：输入切换控制寄存器（ISC）的表格

地址：0x40040C03H　　复位后：00H　　R/W

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

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

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

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

注意：必须将 bit6~2 置“0”。

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

5.3.14 噪声滤波器允许寄存器（NFEN1）

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

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

图 5-21：噪声滤波器允许寄存器（NFEN1）的表格

地址：0x40040C01　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	TI07 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

TNFEN06	TI06 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

TNFEN05	TI05 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

TNFEN04	TI04 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

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

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

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

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

注：能通过设定输入切换控制寄存器（ISC）的 ISC1 位来切换适用的引脚。ISC1=0：可选择是否使用 TI03 引脚的噪声滤波器。ISC1=1：可选择是否使用 RxD0 引脚的噪声滤波器。

备注：通道的定时器输入/输出引脚的有无因产品而不同。

5.3.15 控制定时器输入/输出引脚端口功能的寄存器

在使用通用定时器单元时，必须设定与对象通道复用的端口功能的控制寄存器（端口模式寄存器（PMxx）、端口寄存器（Pxx）和端口模式控制寄存器（PMCxx））。详细内容请参照“2.3.1 端口模式寄存器（PMxx）”、“2.3.2 端口寄存器（Pxx）”和“2.3.8 端口模式控制寄存器（PMCxx）”。

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

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

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

5.4 通用定时器单元的基本规则

5.4.1 多通道联动运行功能的基本规则

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

多通道联动运行功能的基本规则如下所示。

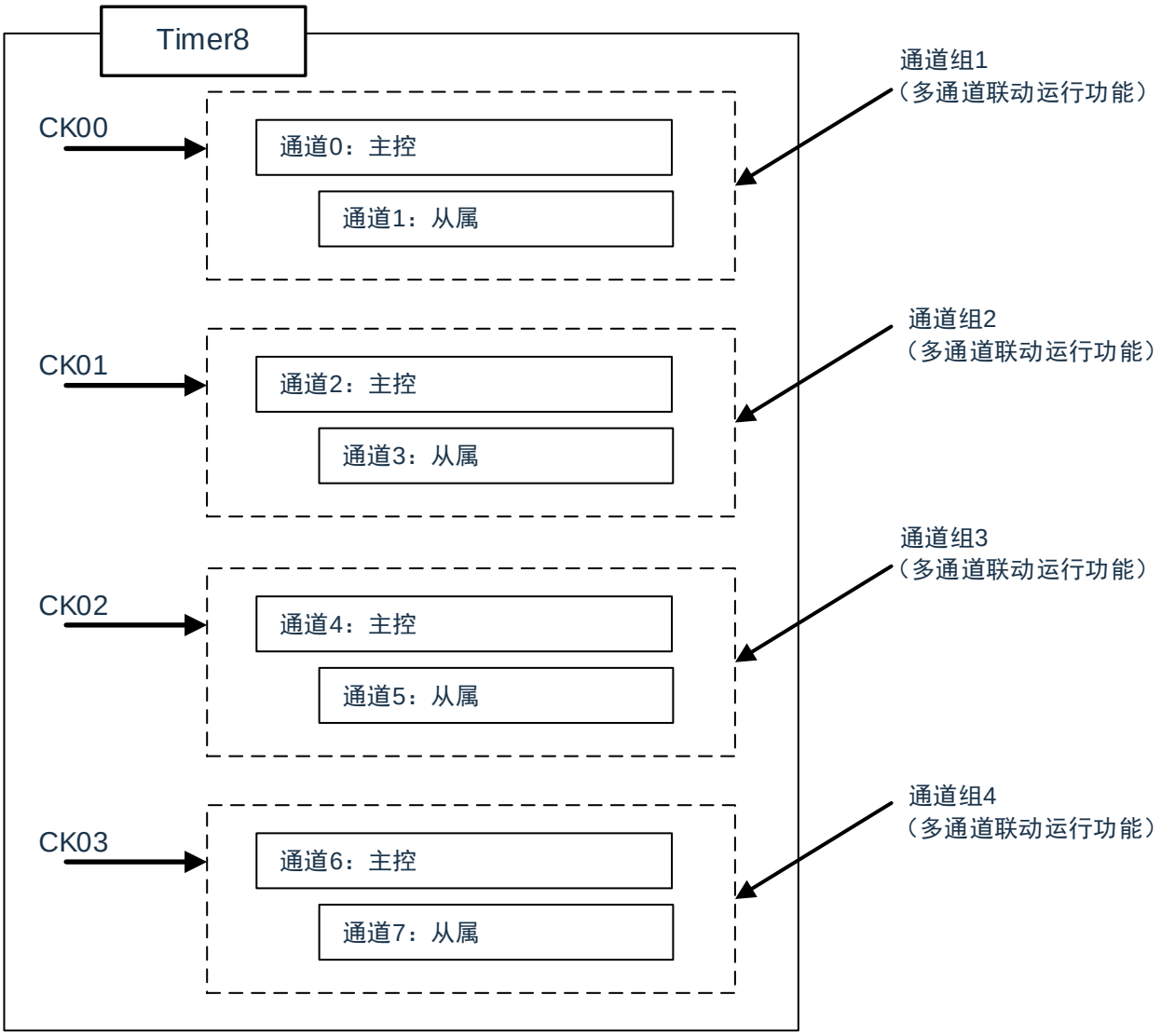
1) 只能将偶数通道（通道0、通道2、通道4、通道6）设定为主控通道。
2) 通道0以外的任何通道都能设定为从属通道。
3) 只能将主控通道的低位通道设定为从属通道。
   例在将通道0设定为主控通道时，能将通道1开始的通道（通道1~7）设定为从属通道。
4) 能对1个主控通道设定多个从属通道。
5) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。
   例在将通道 0 和通道 2 设定为主控通道时，能将通道 1 设定为主控通道 0 的从属通道，而不能将通道 3 设定为主控通道 0 的从属通道。
6) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的CKSmn0位和CKSmn1位（定时器模式寄存器mn（TMRmn）的bit15和bit14）的值需要是相同的设定值。
7) 主控通道能将INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
8) 从属通道能将主控通道的INTTMmn（中断）、开始软件触发和计数时钟用作源时钟，但是不能将自己的INTTMmn（中断）、开始软件触发和计数时钟传给低位通道。
9) 主控通道不能将其他高位主控通道的INTTMmn（中断）、开始软件触发和计数时钟用作源时钟。
10) 为了同时启动要联动的通道，需要同时设定联动通道的通道开始触发位（TSmn）。
11) 只有联动的全部通道或者主控通道才能使用计数运行中的TSmn位的设定。不能只使用从属通道的TSmn位的设定。
12) 为了同时停止要联动的通道，需要同时设定联动通道的通道停止触发位（TTmn）。
13) 在联动运行时，因为主控通道和从属通道需要相同的运行时钟，所以不能选择CKm2/CKm3。
14) 定时器模式寄存器m0（TMRm0）没有主控位而固定为“0”。但是，因为通道0是最高位的通道，所以在联动运行时能将通道0用作主控通道。

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

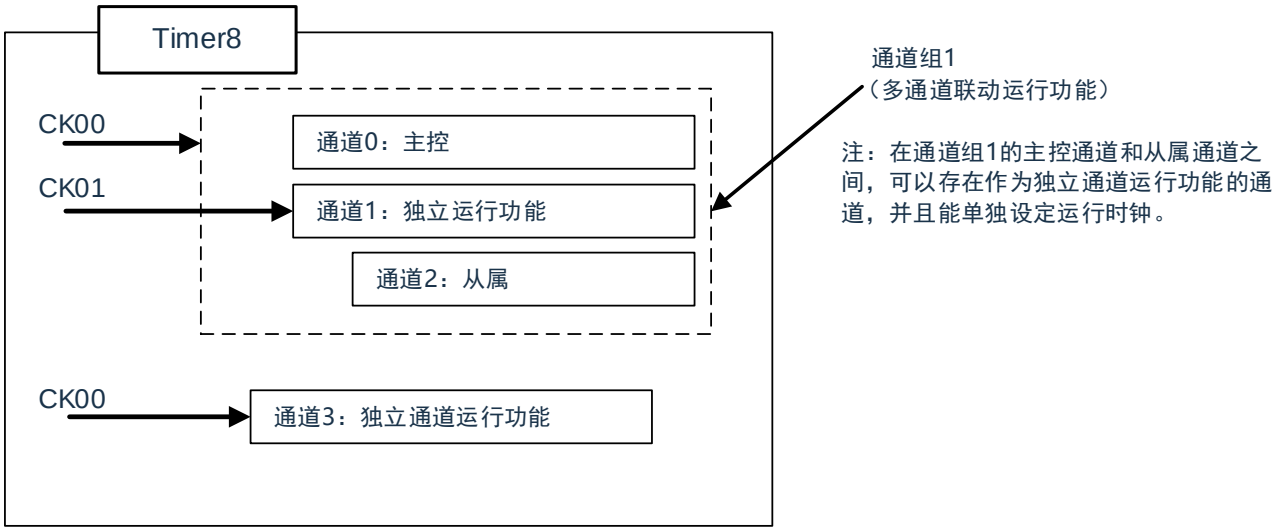
如果设定 2 个或者更多的相互不联动的通道群，通道群之间就不适用上述的基本规则。

备注：m：单元号（m=0）n：通道号（n=0~7）

例 1


Timer8
CK00
通道0：主控
通道1：从属
通道组1
（多通道联动运行功能）
CK01
通道2：主控
通道3：从属
通道组2
（多通道联动运行功能）
CK02
通道4：主控
通道5：从属
通道组3
（多通道联动运行功能）
CK03
通道6：主控
通道7：从属
通道组4
（多通道联动运行功能）

例 2


Timer8
CK00
CK01
通道0：主控
通道1：独立运行功能
通道2：从属
CK00
通道3：独立通道运行功能
通道组1
（多通道联动运行功能）
注：在通道组1的主控通道和从属通道之间，可以存在作为独立通道运行功能的通道，并且能单独设定运行时钟。

5.4.2 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 位是触发位，所以如果变为运行允许状态（TEmn =1），就立即清除 TSmn 位。

通过 16 位存储器操作指令设定 TSm 寄存器。

图 5-22：定时器通道开始寄存器 m（TSm）的表格

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	0	0	0	0	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

备注：m=0

TSmn	通道n的运行允许（开始）触发
0	没有触发。
1	将TEmn位置“1”，进入计数允许状态。计数允许状态下的TCRmn寄存器的计数开始因各运行模式而不同（参照“5.5.2计数器的开始时序”的表5-6）。

注意：

1. 必须将 bit15~8 置“0”。
2. 在从不使用 TImn 引脚输入的功能切换到使用 TImn 引脚输入的功能时，从设定定时器模式寄存器 mn（TMRmn）到将 TSmn 位置“1”为止，需要以下期间的等待：

   TImn 引脚噪声滤波器有效时（TNFENmn=1）：4 个运行时钟（F_{MCK}）

   TImn 引脚噪声滤波器无效时（TNFENmn=0）：2 个运行时钟（F_{MCK}）

备注：

1. TSm 寄存器的读取值总是“0”。
2. m：单元号（m=0）n：通道号（n=0~7）

5.5 计数器的运行

5.5.1 计数时钟（F_{TCLK}）

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

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

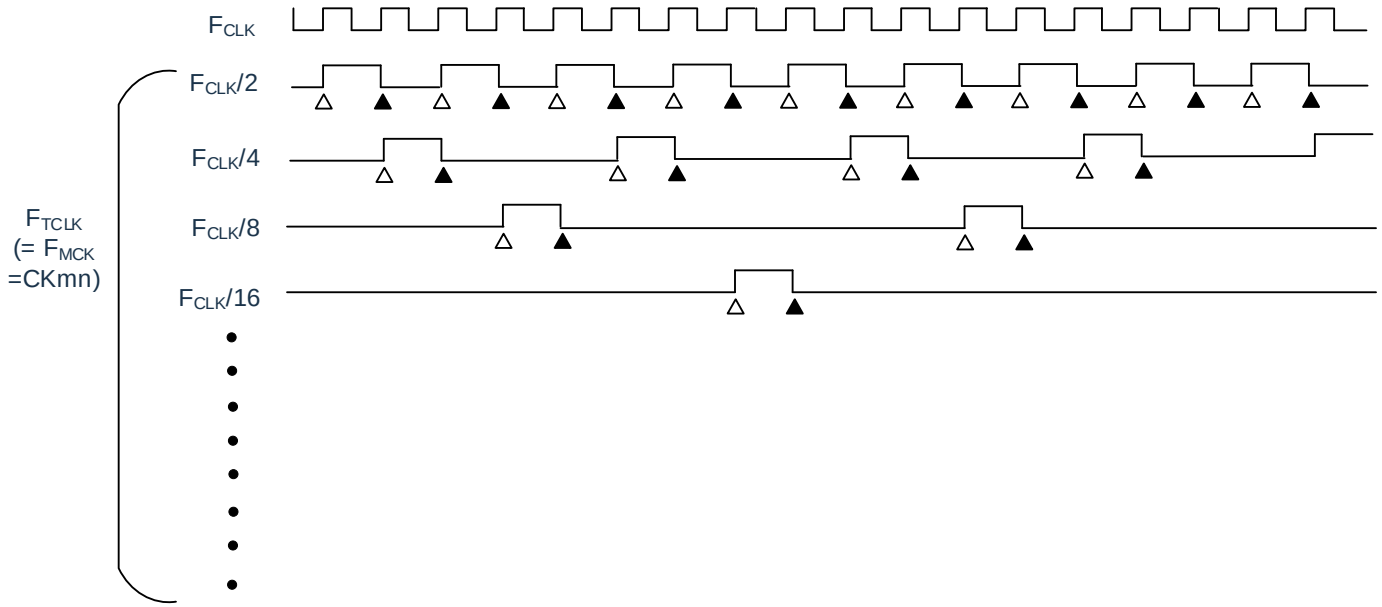
通用定时器单元被设计为与 F_{CLK} 同步运行，因此计数时钟（F_{TCLK}）的时序如下。

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

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

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

图 5-23：F_{CLK} 和计数时钟（F_{TCLK}）的时序（CCSmn=0 的情况）



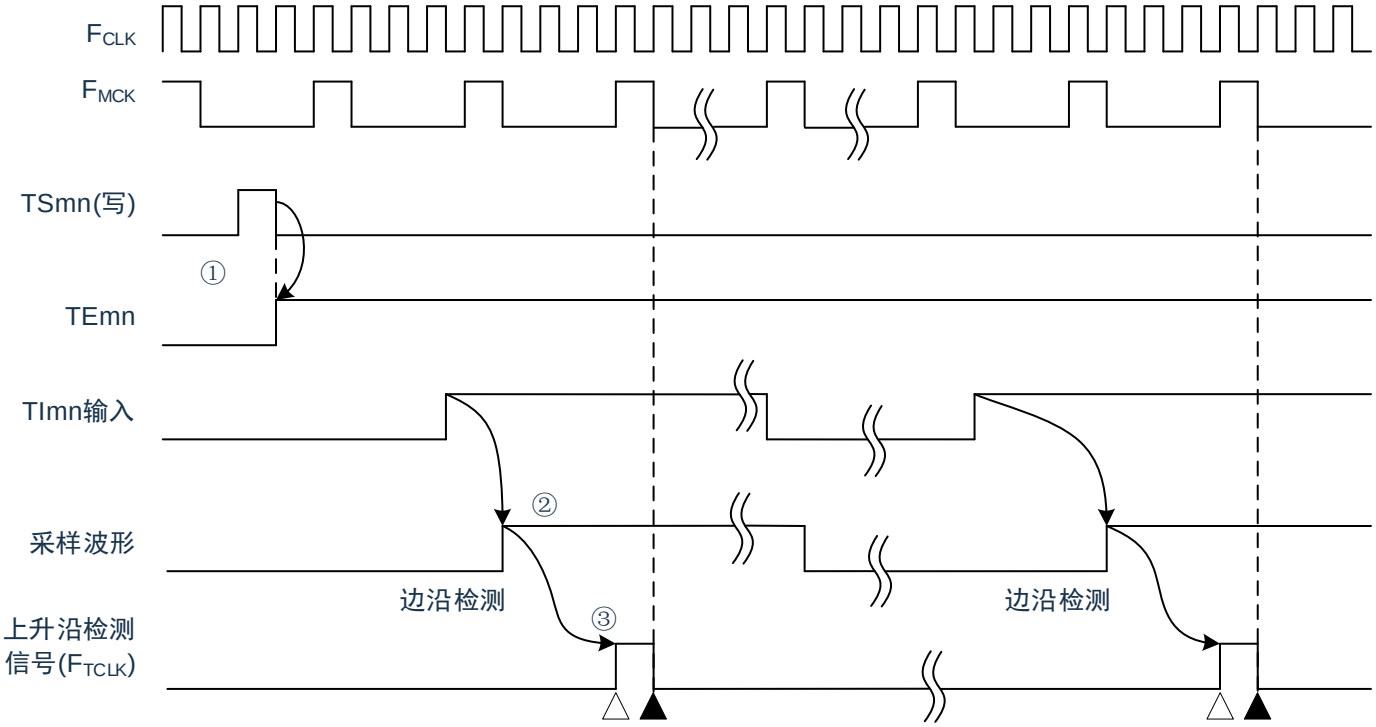
备注：

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

(2) 选择 TImn 引脚输入信号的有效边沿的情况（CCSmn=1）

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

图 5-24：计数时钟（F_{TCLK}）的时序（CCSmn=1，未使用噪声滤波器的情况）



① 通过将TSmn位置位来开始定时器的运行，并且等待TImn输入的有效边沿。

② 通过F_{MCK}对TImn输入的上升沿进行采样。

③ 在采样信号的上升沿检测边沿，并且输出检测信号（计数时钟）。

备注：

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

5.5.2 计数器的开始时序

通过将定时器通道开始寄存器 m（TSm）的 TSmn 位置位，定时器计数寄存器 mn（TCRmn）进入运行允许状态。

从计数允许状态到定时器计数寄存器 mn（TCRmn）开始计数为止的运行如表 5-4 所示。

表 5-4：从计数允许状态到定时器计数寄存器 mn（TCRmn）开始计数为止的运行

定时器的运行模式	将TSmn位置“1”后的运行
• 间隔定时器模式	从检测到开始触发（TSmn=1）到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器，并且通过后续的计数时钟进行递减计数（参照“5.5.3(1)间隔定时器模式的运行”）。
• 事件计数器模式	通过给TSmn位写“1”，将TDRmn寄存器的值装入TCRmn寄存器。 如果检测到TImn的输入边沿，就通过后续的计数时钟进行递减计数。（参照“5.5.3(2)事件计数器模式的运行”）。
• 捕捉模式	从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器，并且通过后续的计数时钟进行递增计数（参照“5.5.3(3)捕捉模式的运行（输入脉冲的间隔测量）”）。
• 单次计数模式	通过在定时器停止运行（TEmn=0）的状态下给TSmn位写“1”，进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将TDRmn寄存器的值装入TCRmn寄存器，并且通过后续的计数时钟进行递减计数（参照“5.5.3(4)单次计数模式的运行”）。
• 捕捉&单次计数模式	通过在定时器停止运行（TEmn=0）的状态下给TSmn位写“1”，进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止，不执行任何操作。 通过第1个计数时钟将“0000H”装入TCRmn寄存器，并且通过后续的计数时钟进行递增计数（参照“5.5.3(5)捕捉&单次计数模式的运行（高电平宽度的测量）”）。

5.5.3 计数器的运行

以下说明各模式的计数器运行。

(1) 间隔定时器模式的运行

① 通过给 TSmn 位写“1”，进入运行允许状态（TEmn=1）。定时器计数寄存器 mn（TCRmn）保持初始值，直到产生计数时钟为止。

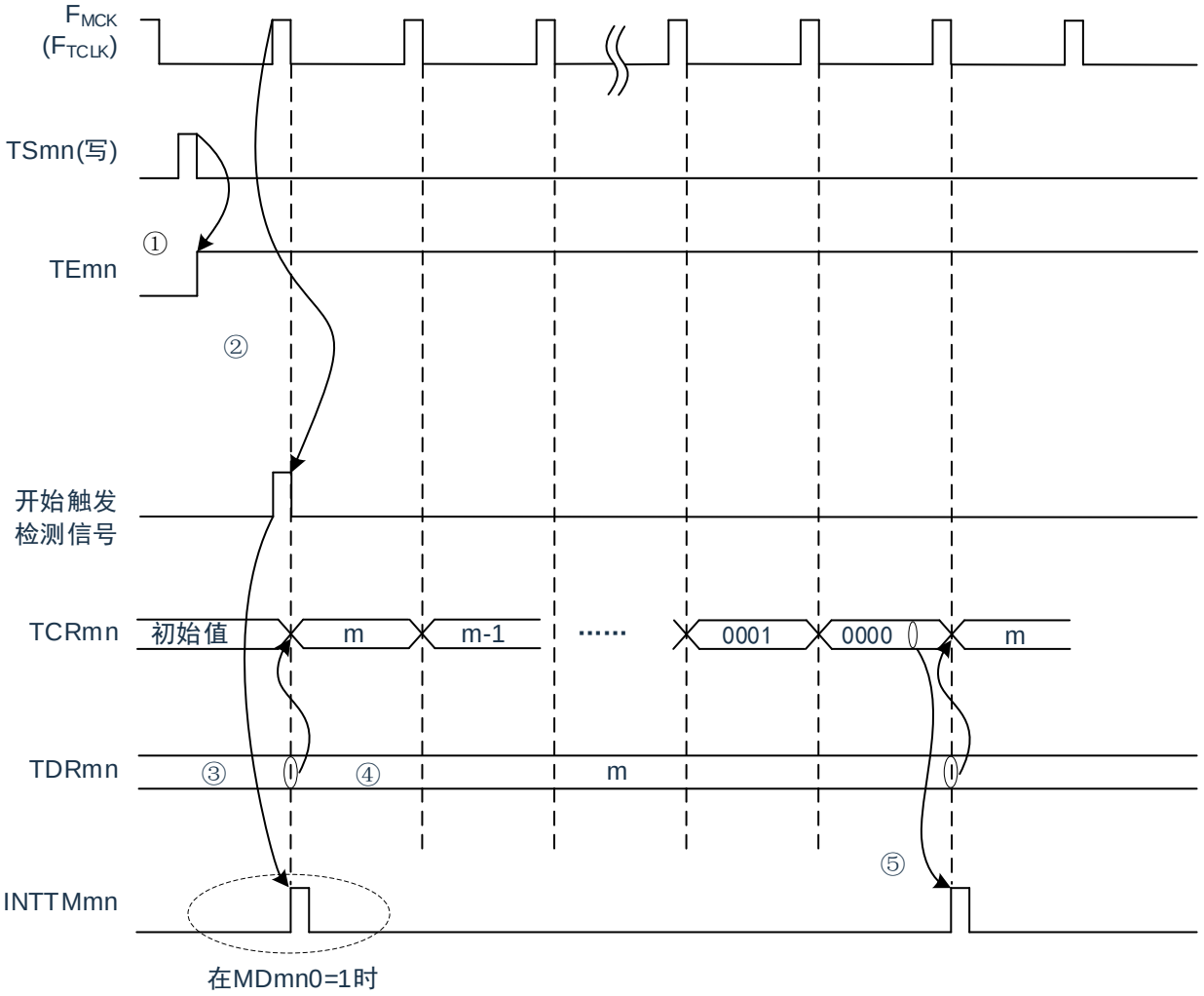
② 通过允许运行后的第 1 个计数时钟（F_{MCK}）产生开始触发信号。

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 5-25：运行时序（间隔定时器模式）



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

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

(2) 事件计数器模式的运行

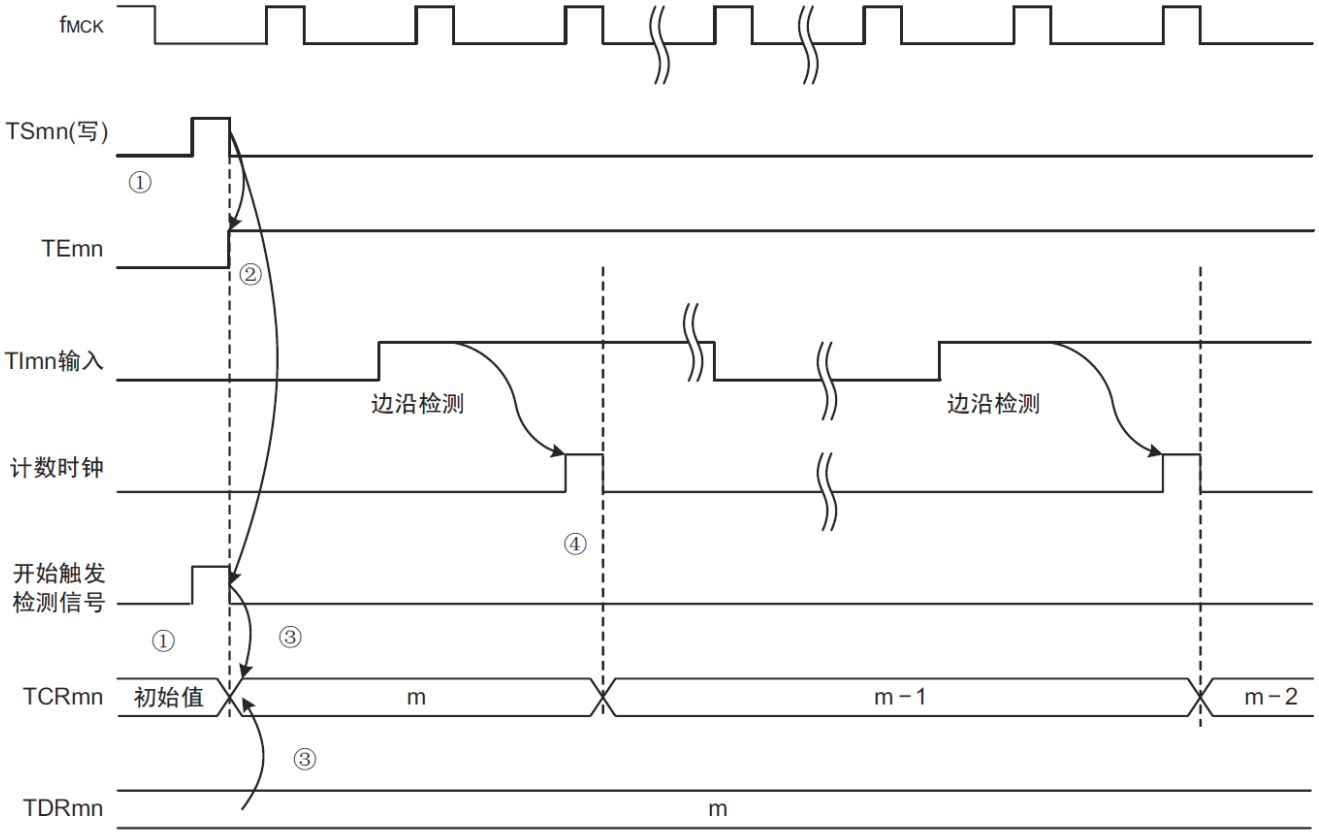
① 在运行停止状态（TEmn=0）的期间，定时器计数寄存器 mn（TCRmn）保持初始值。

② 通过给 TSmn 位写“1”，进入运行允许状态（TEmn=1）。

③ 在 TSmn 位和 TEmn 位都变为“1”的同时将定时器数据寄存器 mn（TDRmn）的值装入 TCRmn 寄存器，并且开始计数。

④ 此后，在 TImn 输入的有效边沿，通过计数时钟对 TCRmn 寄存器的值进行递减计数。

图 5-26：运行时序（事件计数器模式）



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

(3) 捕捉模式的运行（输入脉冲的间隔测量）

① 通过给 TSmn 位写“1”，进入运行允许状态（TEmn=1）。

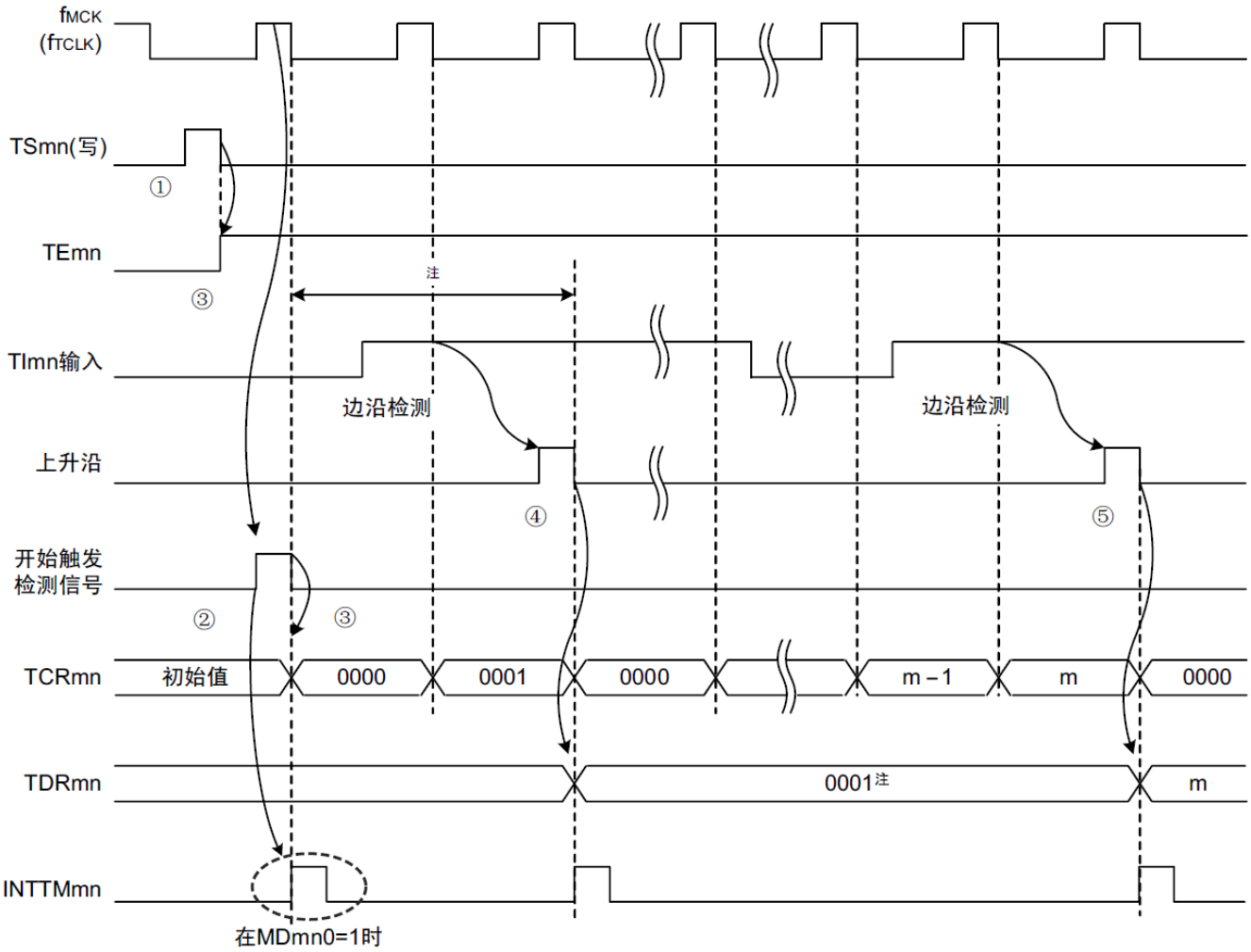
② 定时器计数寄存器 mn（TCRmn）保持初始值，直到产生计数时钟为止。

③ 通过允许运行后的第 1 个计数时钟（F_{MCK}）产生开始触发信号。然后，将“0000H”装入 TCRmn 寄存器并且以捕捉模式开始计数（当 MDmn0 位为“1”时，通过开始触发信号产生 INTTMmn）。

④ 如果检测到 TImn 输入的有效边沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。此时的捕捉值没有意义。TCRmn 寄存器从“0000H”开始继续计数。

⑤ 如果检测到下一个 TImn 输入的有效边沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。

图 5-27：运行时序（捕捉模式：输入脉冲的间隔测量）



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

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

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

(4) 单次计数模式的运行

① 通过给 TSmn 位写“1”，进入运行允许状态（TEmn=1）。

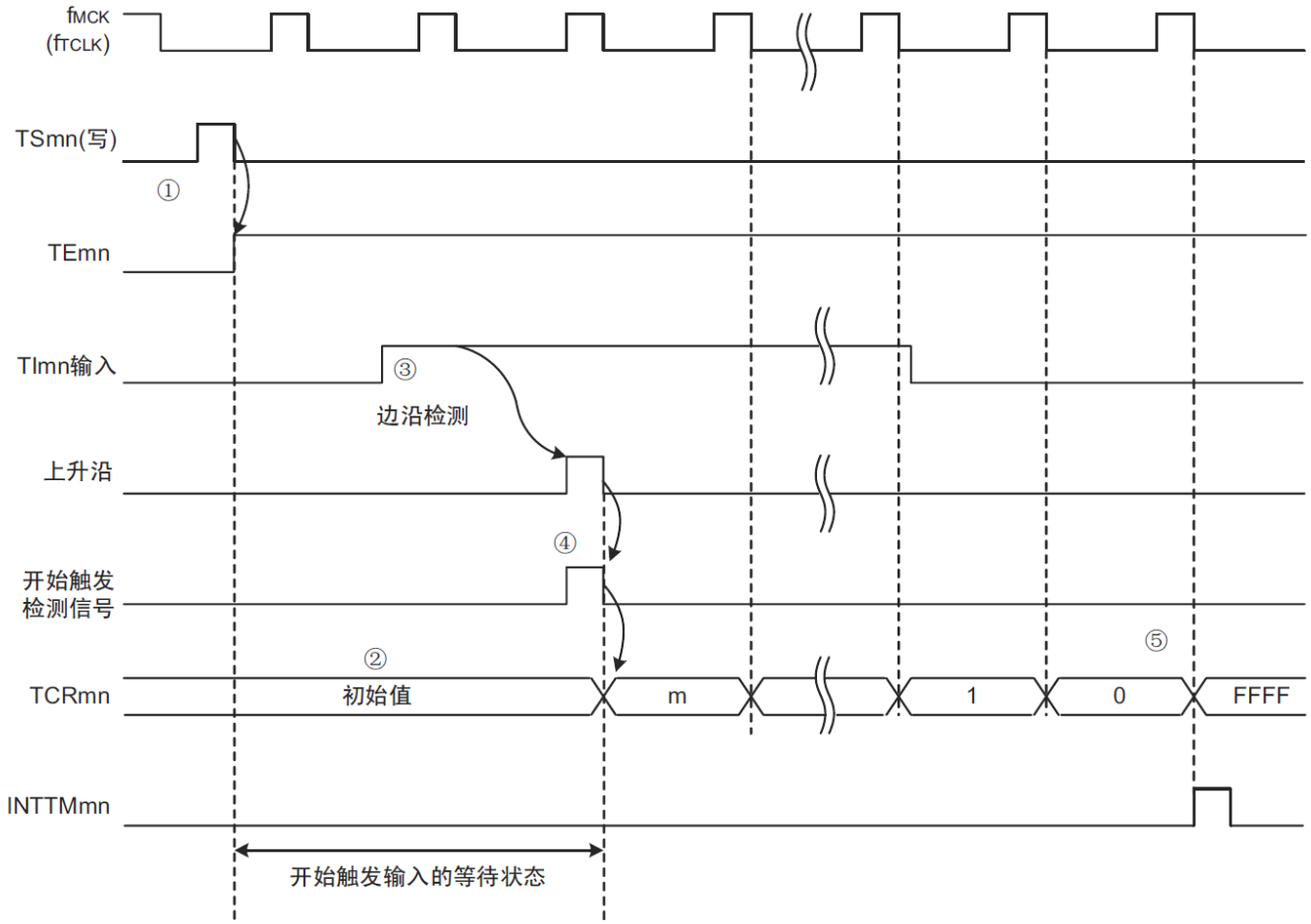
② 定时器计数寄存器 mn（TCRmn）保持初始值，直到产生开始触发信号为止。

③ 检测 TImn 输入的上升沿。

④ 在产生开始触发信号后将 TDRmn 寄存器的值（m）装入 TCRmn 寄存器，并且开始计数。

⑤ 当 TCRmn 寄存器递减计数到“0000H”时，产生 INTTMmn 中断，并且 TCRmn 寄存器的值变为“FFFFH”，停止计数。

图 5-28：运行时序（单次计数模式）



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

(5) 捕捉&单次计数模式的运行（高电平宽度的测量）

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

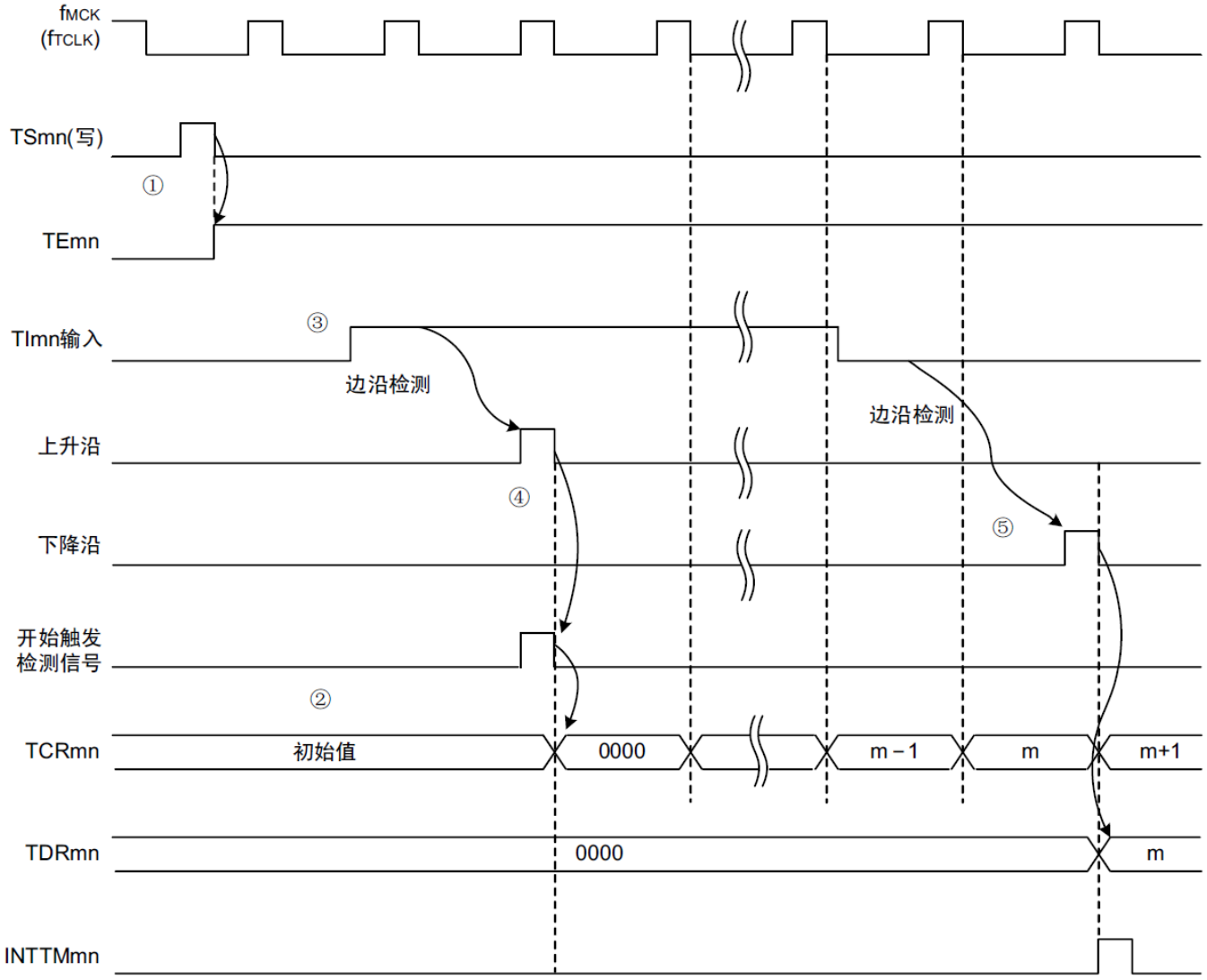
② 定时器计数寄存器 mn（TCRmn）保持初始值，直到产生开始触发信号为止。

③ 检测 TImn 输入的上升沿。

④ 在产生开始触发信号后将“0000H”装入 TCRmn 寄存器，并且开始计数。

⑤ 如果检测到 TImn 输入的下降沿，就将 TCRmn 寄存器的值捕捉到 TDRmn 寄存器，并且产生 INTTMmn 中断。

图 5-29：运行时序（捕捉&单次计数模式：高电平宽度的测量）

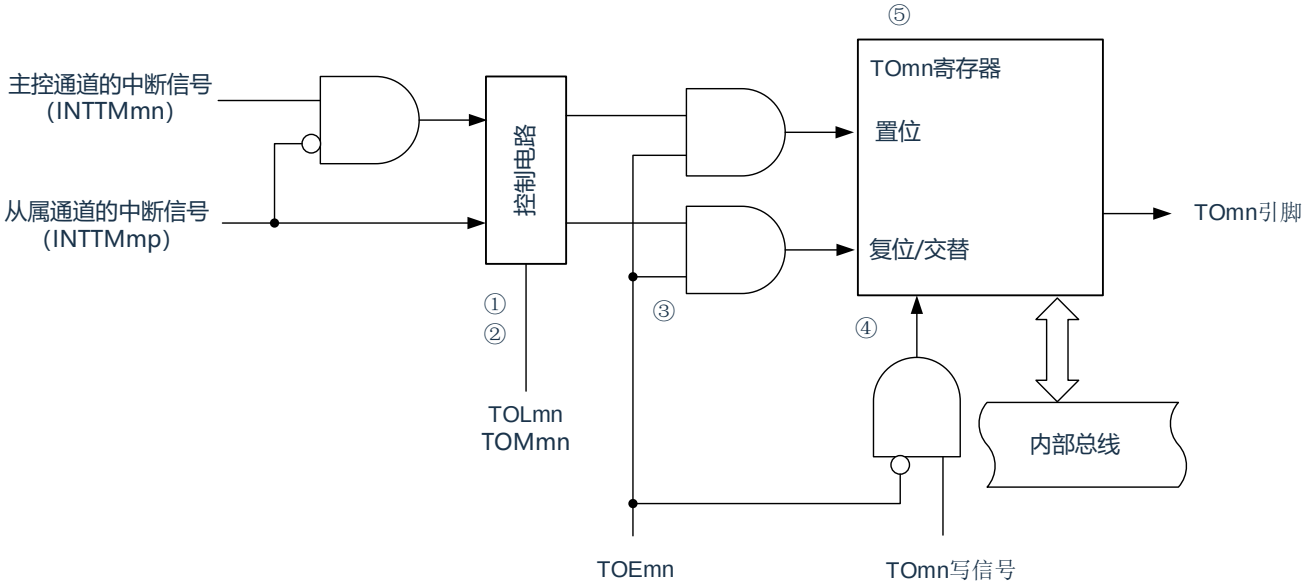


备注：这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TImn 输入开始再延迟 2 个 F_{MCK} 周期（合计 3~4 个周期）。1 个周期的误差是因为 TImn 输入和计数时钟（F_{MCK}）不同步。

5.6 通道输出（TOmn引脚）的控制

5.6.1 TOmn引脚输出电路的结构

图 5-30：输出电路的结构



以下说明 TOmn 引脚的输出电路。

① 当TOMmn位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器m（TOLm）的设定值，只将INTTMmp（从属通道定时器中断）传给定时器输出寄存器m（TOm）。

② 当TOMmn位为“1”（从属通道输出模式）时，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOm寄存器。

此时，TOLm寄存器有效并且进行以下信号的控制：

TOLmn=0时：正相运行（INTTMmn→置位、INTTMmp→复位）TOLmn=1时：反相运行（INTTMmn→复位、INTTMmp→置位）

当同时产生INTTMmn和INTTMmp时（PWM输出的0%输出），优先INTTMmp（复位信号）而屏蔽INTTMmn（置位信号）。

③ 在允许定时器输出（TOEmn=1）的状态下，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOm寄存器。TOm寄存器的写操作（TOmn写信号）无效。

当TOEmn位为“1”时，除了中断信号以外，不改变TOmn引脚的输出。

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

④ 在禁止定时器输出（TOEmn=0）的状态下，对象通道的TOmn位的写操作（TOmn写信号）有效。当定时器输出为禁止状态（TOEmn=0）时，不将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOm寄存器。

⑤ 能随时读TOm寄存器，并且能确认TOmn引脚的输出电平。

备注：m：单元号（m=0）

n：通道号，n=0~7（主控通道：n=0、2、4、6）

p：从属通道号

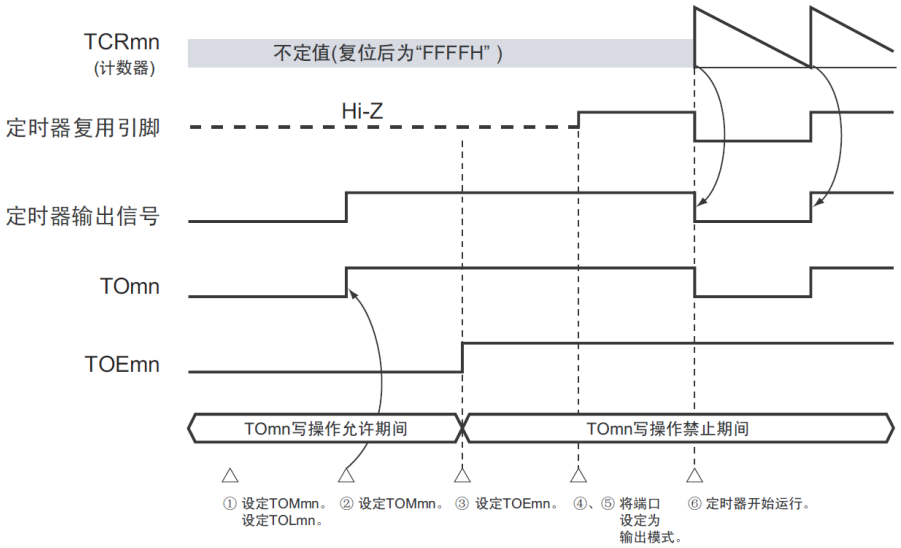
n=0：p=1、2、3

n=2：p=3

5.6.2 TOmn引脚的输出设定

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 5-31：从设定定时器的输出到开始运行的状态变化



① 设定定时器输出的运行模式。
- TOMmn 位（0：主控通道输出模式、1：从属通道输出模式）
- TOLmn 位（0：正逻辑输出、1：负逻辑输出）

② 通过设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。

③ 给 TOEmn 位写“1”，允许定时器输出（禁止写 TOm 寄存器）。

④ 通过端口模式控制寄存器（PMCxx）将端口设定为数字输入/输出（参照“5.3.15 控制定时器输入/输出引脚端口功能的寄存器”）。

⑤ 将端口的输入/输出设定为输出（参照“5.3.15 控制定时器输入/输出引脚端口功能的寄存器”）。

⑥ 允许定时器运行（TSmn=1）。

备注：m：单元号（m=0）n：通道号（n=0~7）

5.6.3 通道输出运行的注意事项

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

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

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

备注：m：单元号（m=0）n：通道号（n=0~7）

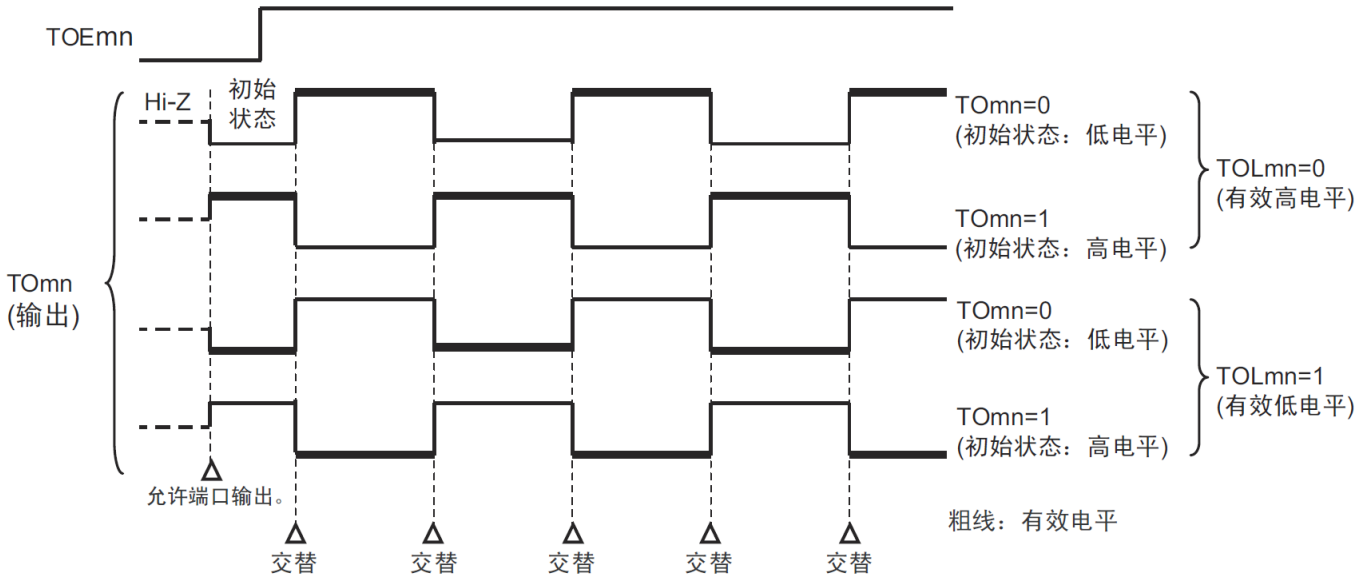
(2) 有关 TOmn 引脚的初始电平和定时器开始运行后的输出电平

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

(a) 在主控通道输出模式（TOMmn=0）中开始运行的情况

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

图 5-32：交替输出时（TOMmn=0）的 TOmn 引脚输出状态



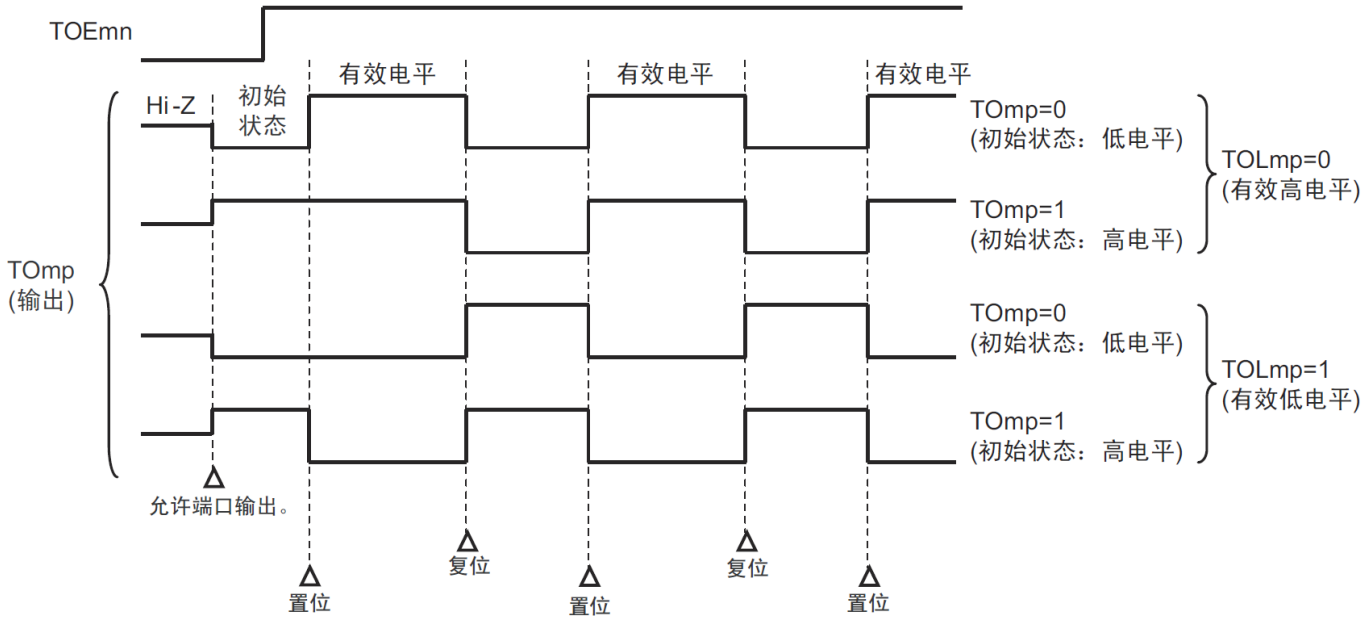
备注：

1. 交替：反相 TOmn 引脚的输出状态。
2. m：单元号（m=0）n：通道号（n=0~7）

(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）的设定。

图 5-33：PWM 输出时（TOMmn=1）的 TOmn 引脚输出状态



备注：

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

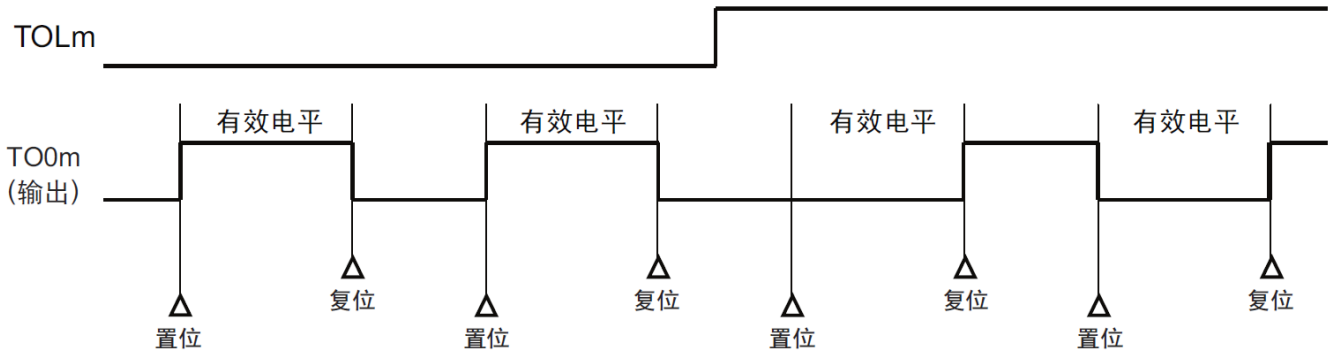
(3) 有关从属通道输出模式（TOMmn=1）的 TOmn 引脚变化

(a) 在定时器运行中更改定时器输出电平寄存器 m（TOLm）的设定的情况

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

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

图 5-34：在定时器运行中更改 TOLm 寄存器的内容时的运行



备注：

1. 置位：TOmn 引脚的输出信号从无效电平变为有效电平。
   复位：TOmn 引脚的输出信号从有效电平变为无效电平。
2. m：单元号（m=0）n：通道号（n=0~7）

(b) 置位/复位时序

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

当置位条件和复位条件同时产生时，优先复位条件。

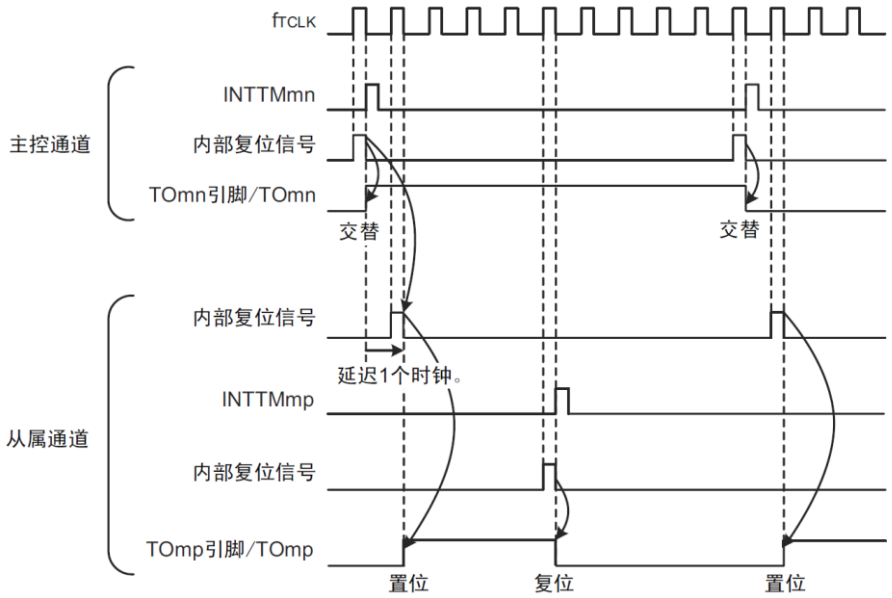
按照以下方法设定主控/从属通道时的置位/复位运行状态如图 5-35 所示。

主控通道：TOEmn=1、TOMmn=0、TOLmn=0

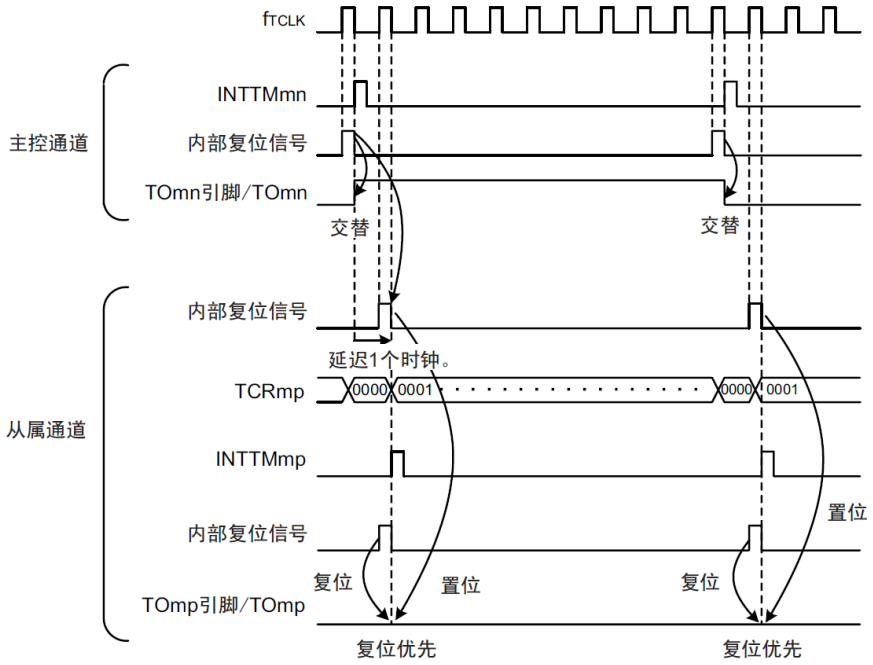
从属通道：TOEmp=1、TOMmp=1、TOLmp=0

图 5-35：置位/复位时序运行状态

(1) 基本运行时序



(2) 0%占空比的运行时序



备注：

1. 内部复位信号：TOmn 引脚的复位/交替信号

   内部置位信号：TOmn 引脚的置位信号

2. m：单元号（m=0）

   n：通道号 n=0~7（主控通道：n=0、2、4、6）

   p：从属通道号

   n=0：p=1、2、3

   n=2：p=3

5.6.4 TOmn位的一次性操作

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

图 5-36：TO0n 位的一次性操作例子

写前

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO031	TO020	TO011	TO000
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE030	TOE020	TOE010	TOE001

要写的数据

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

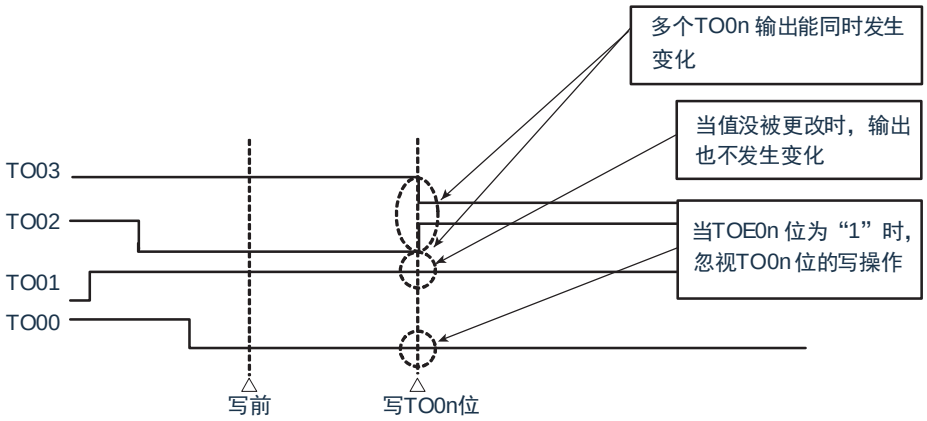
写后

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO030	TO021	TO011	TO000

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

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

图 5-37：一次性操作 TO0n 位时的 TO0n 引脚状态



备注：m：单元号（m=0）n：通道号（n=0~7）

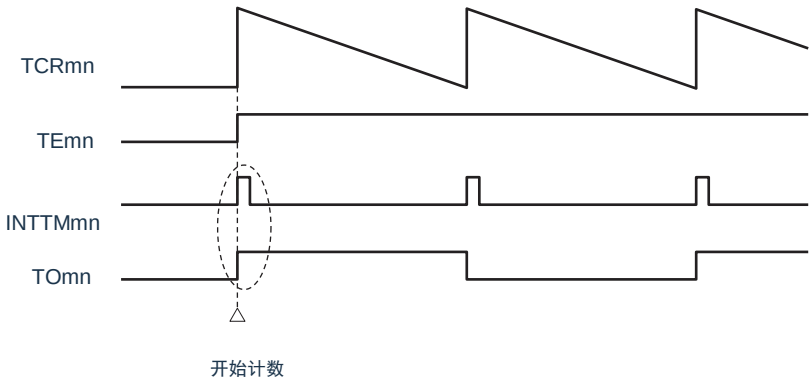
5.6.5 有关开始计数时的定时器中断和TOmn引脚输出

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

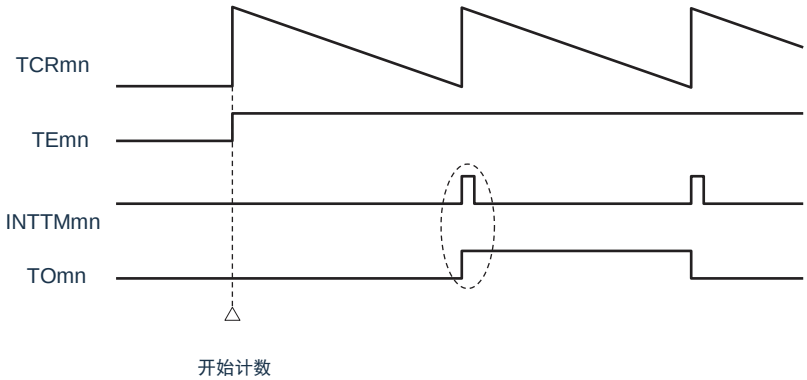
当 MDmn0 位为“1”时，能通过产生定时器中断（INTTMmn）得知计数的开始时序。在其他模式中，不控制开始计数时的定时器中断和 TOmn 输出。设定为间隔定时器模式（TOEmn=1、TOMmn=0）时的运行例子如下所示

图 5-38：开始计数时的定时器中断和 TOmn 输出的运行例子

(a) MDmn0 位为“1”的情况



(b) MDmn0 位为“0”的情况



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

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

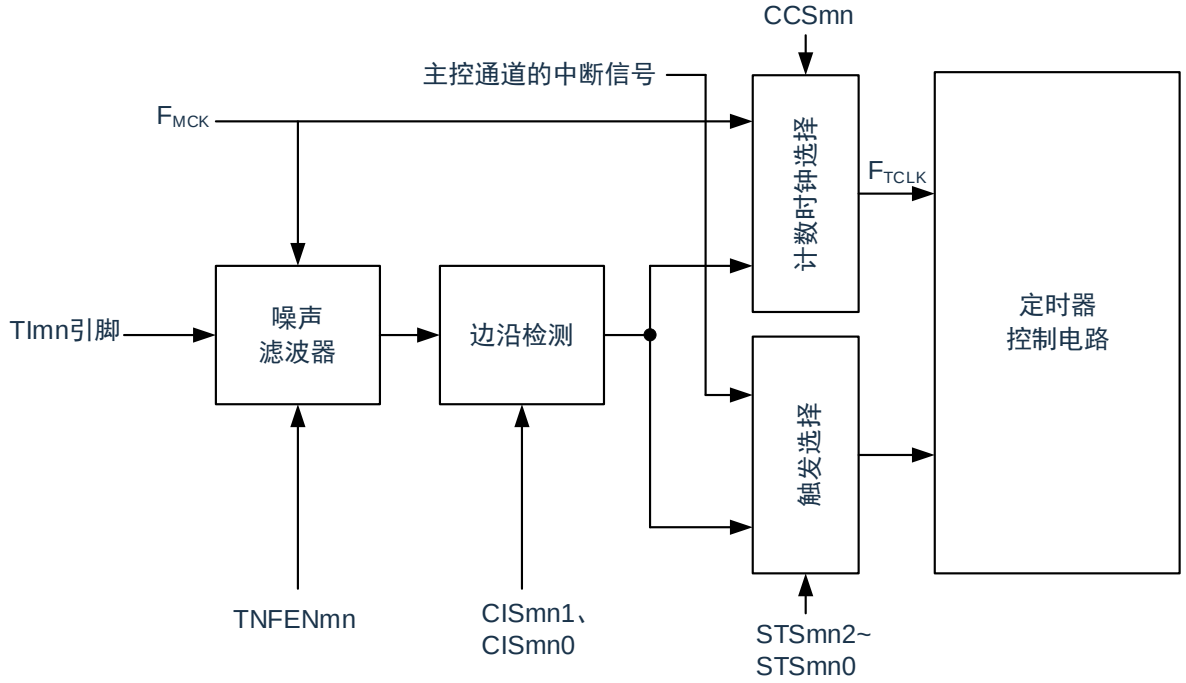
备注：m：单元号（m=0）n：通道号（n=0~7）

5.7 定时器输入（TImn）的控制

5.7.1 TImn引脚输入电路的结构

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

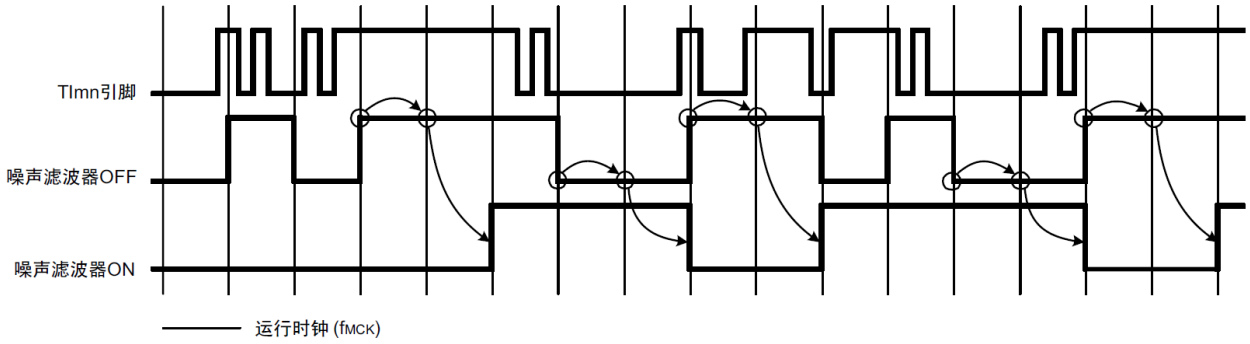
图 5-39：输入电路的结构



5.7.2 噪声滤波器

当噪声滤波器无效时，只通过通道 n 的运行时钟（F_{MCK}）进行同步；当噪声滤波器有效时，在通过通道 n 的运行时钟（F_{MCK}）进行同步后检测 2 个时钟是否一致。TM4mn 输入引脚在噪声滤波器 ON 或者 OFF 的情况下，经过噪声滤波器电路后的波形如下所示。

图 5-40：TImn 输入引脚在噪声滤波器 ON 或者 OFF 情况下的采样波形



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

5.7.3 操作通道输入时的注意事项

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

(1) 噪声滤波器为 OFF 的情况

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

(2) 噪声滤波器为 ON 的情况

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

5.8 通用定时器单元的独立通道运行功能

5.8.1 作为间隔定时器/方波输出的运行

(a) 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：

INTTMmn（定时器中断）的产生周期=计数时钟周期×(TDRmn 的设定值+1)

(b) 作为方波输出的运行

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

TOmn 输出方波的周期和频率能用以下计算式进行计算：

• TOmn 输出的方波周期=计数时钟周期×(TDRmn 的设定值+1)×2

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

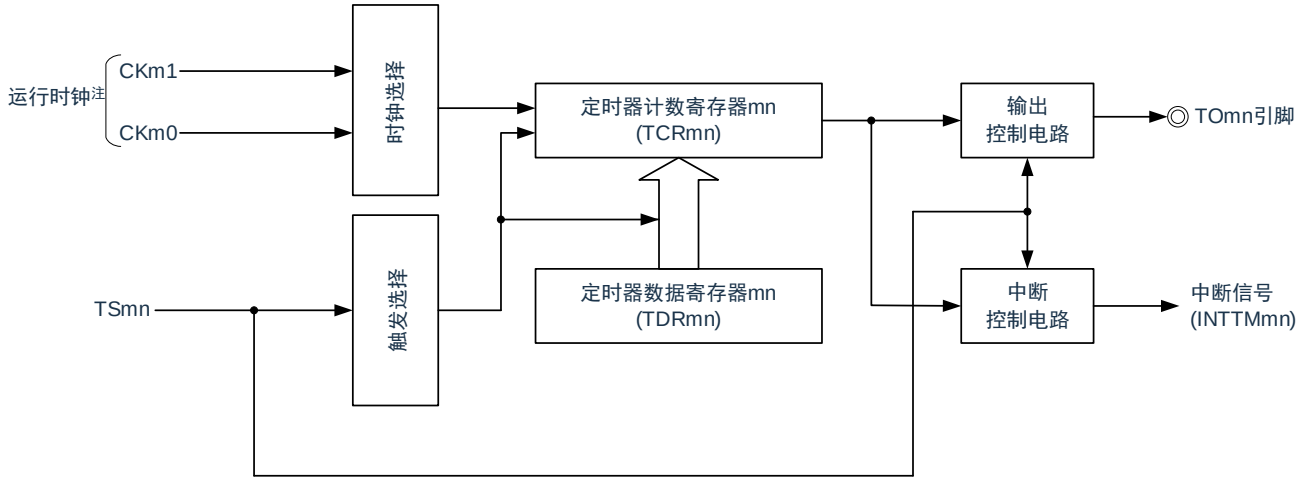
在间隔定时器模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

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

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

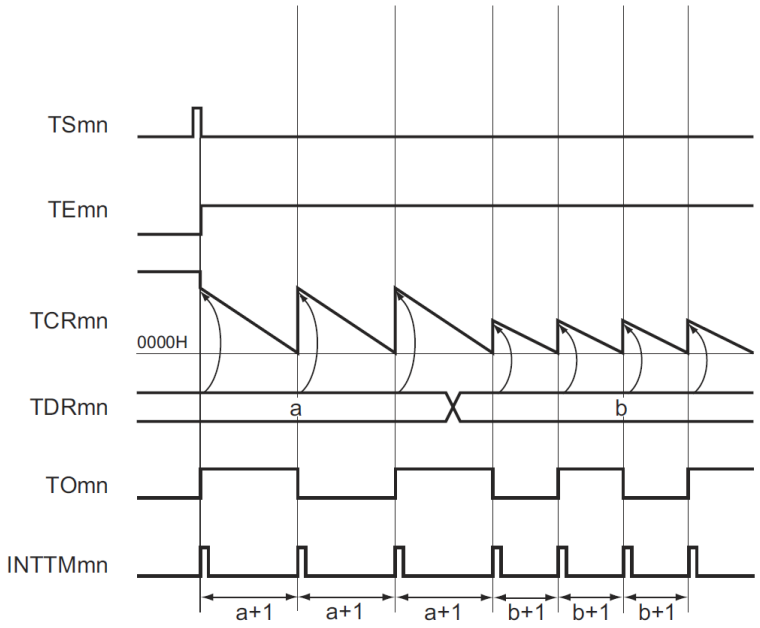
能随时改写 TDRmn 寄存器，改写的 TDRmn 寄存器的值从下一个周期开始有效。

图 5-41：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 5-42：作为间隔定时器/方波输出运行的基本时序例子（MDmn0=1）



备注：

1. m：单元号（m=0）n：通道号（n=0~7）
2. TSmn：定时器通道开始寄存器 m（TSm）的 bit n

   TEmn：定时器通道允许状态寄存器 m（TEm）的 bit n

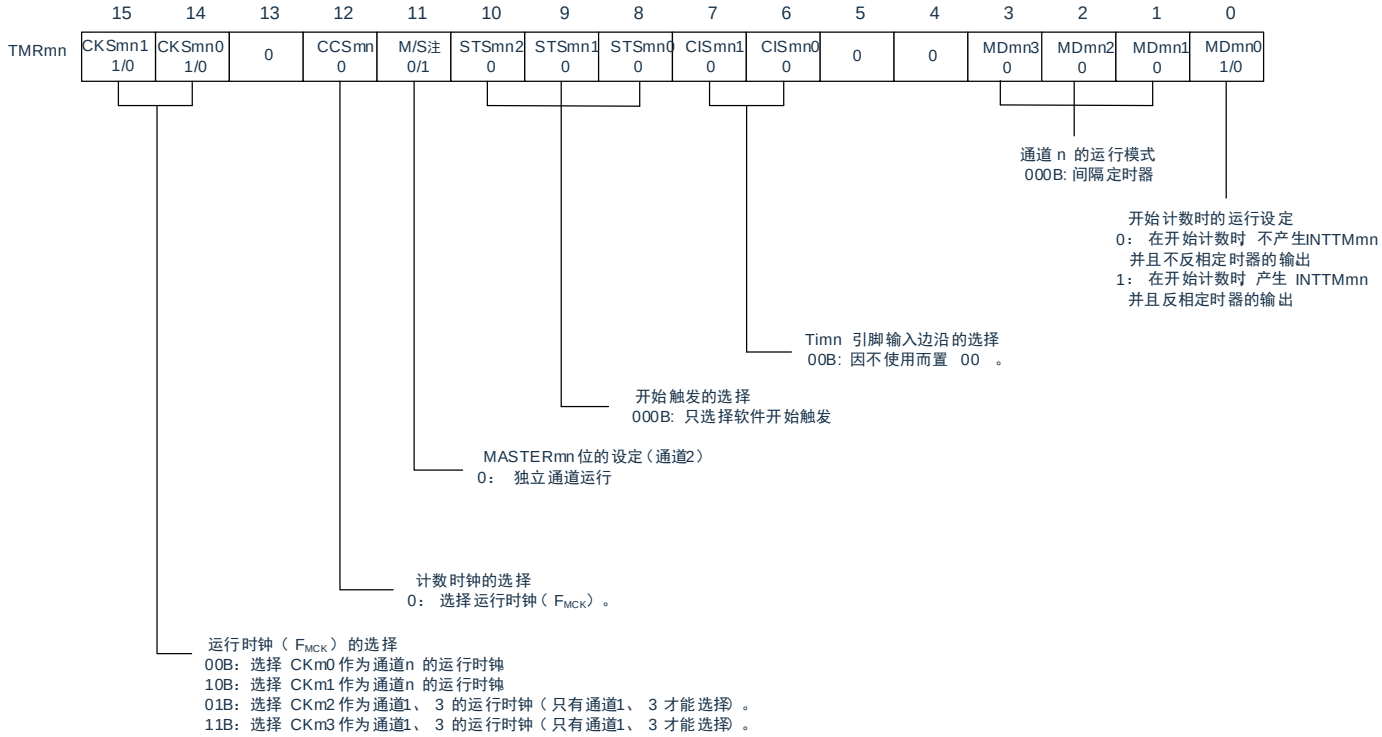
   TCRmn：定时器计数寄存器 mn（TCRmn）

   TDRmn：定时器数据寄存器 mn（TDRmn）

   TOmn：TOmn 引脚输出信号

图 5-43：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）

	bit n	
TOm	Tomn 0/1	0：由 TOmn 输出 “0”。 1：由 TOmn 输出 “1”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit n	
TOEm	TOEmn 0/1	0：停止由计数运行进行的 TOmn 输出。 1：允许由计数运行进行的 TOmn 输出。

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

注：TMRm2：MASTERmn 位

TMRm0：固定为“0”。

备注：m：单元号 （m=0） n：通道号 （n=0 ~ 7）

图 5-44：间隔定时器/ 方波输出功能时的操作步骤

		软件操作	硬件状态
	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 TM80EN 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	设定时器模式寄存器 mn (TMRmn)(确定通道的运行模式)。 给定时器数据寄存器 mn(TDRmn)设定间隔(周期)值。	通道处于运行停止状态。 (提供时钟，消耗一部分电力)
		使用 TOmn 输出的情况： 将定时器输出模式寄存器 m（TOMm）的 TOMmn 位置 “0” (主控通道输出模式)。 将 TOLmn 位置 “0”。 设定 TOmn 位，确定 TOmn 输出的初始电平。 将 TOEmn 位置 “1”，允许 TOmn 输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmn 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 “0” 时，输出 TOmn 初始设定的电平。 因为通道处于运行停止状态，所以 TOmn 不变。 TOmn 引脚输出 TOmn 设定的电平。
	开始运行	（只在使用 TOmn 输出并且重新开始时，将 TOEmn 位置 “1”）将 TSmn 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）。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。
重新开始运行	停止运行	将 TTmn 位置 “1”。 运行因为 TTmn 位是触发位，所以自动返回到 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 TOmn 输出不被初始化而保持状态。
		将 TOEmn 位置 “0” 并且给 TOmn 位设定值。	TOmn 引脚输出 TOmn 位设定的电平。
	TAU 停止	要保持 TOmn 引脚输出电平的情况： 在给端口寄存器设定要保持的值后将 TOmn 位置 “0”。 不需要保持 TOmn 引脚输出电平的情况： 不需要设定。	通过端口功能保持 TOmn 引脚的输出电平。
		将 PER0 寄存器的 TM80EN 位置“0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 （TOmn 位变为“0” 并且 TOmn 引脚变为端口功能）

备注：m：单元号（m=0） n：通道号（n=0~7）

5.8.2 作为外部事件计数器的运行

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

规定的计数值=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）置“1”，将定时器数据寄存器 mn（TDRmn）的值装入 TCRmn 寄存器。

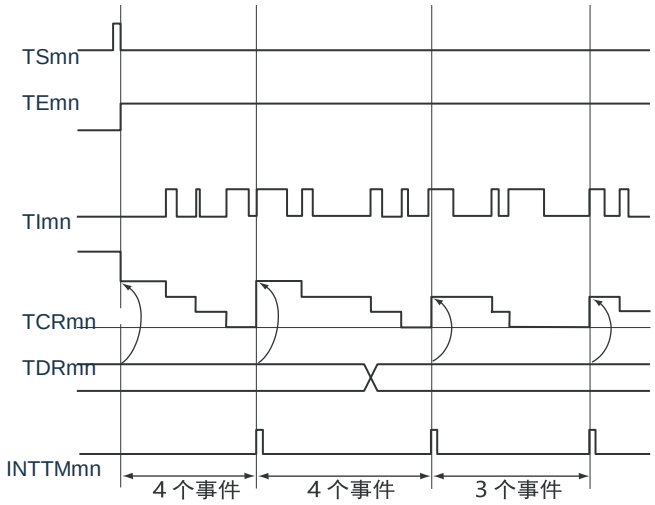
TCRmn 寄存器在检测到 TImn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入 TDRmn 寄存器的值并且输出 INTTMmn。

此后，继续同样的运行。

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

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

图 5-45：作为外部事件计数器运行的基本时序例子

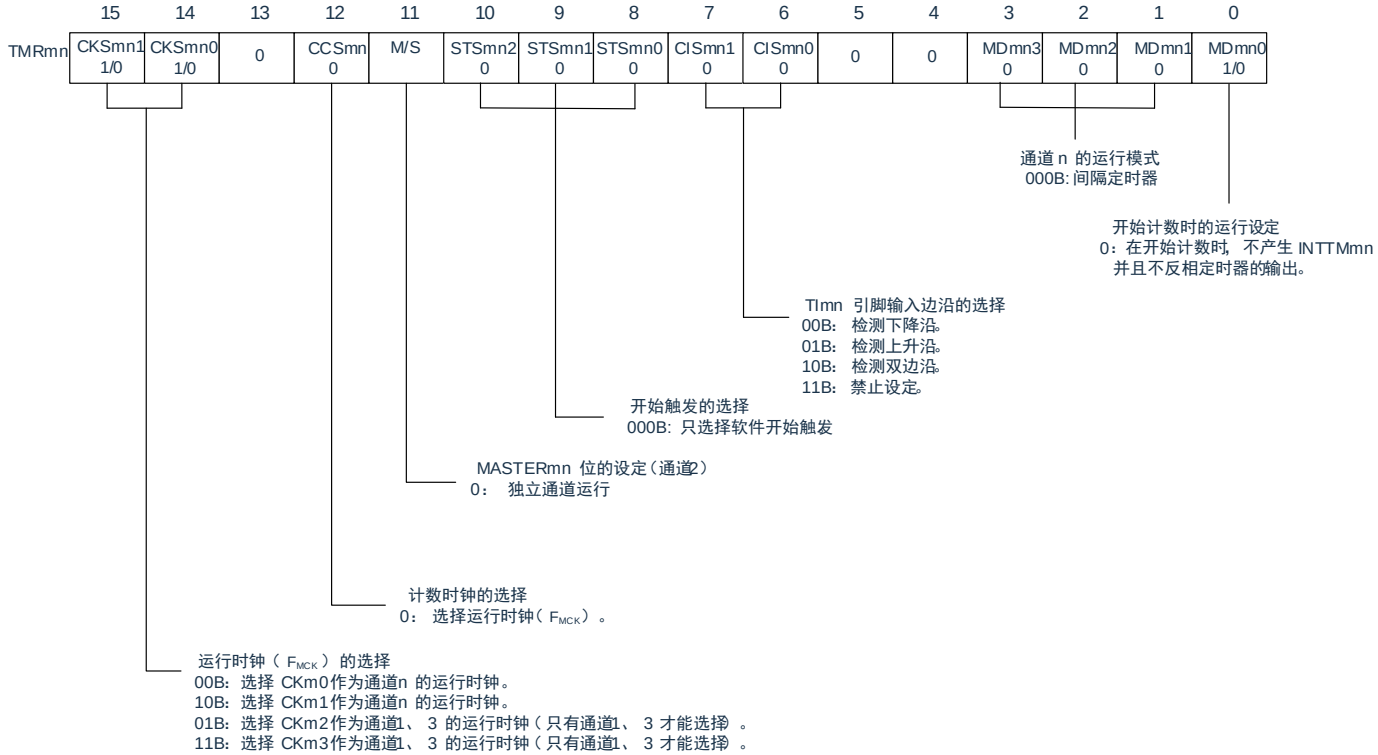


备注：

1. m：单元号（m=0）n：通道号（n=0~7）
2. TSmn：定时器通道开始寄存器 m（TSm）的 bit n
   TEmn：定时器通道允许状态寄存器 m（TEm）的 bit n
   TImn：TImn 引脚输入信号
   TCRmn：定时器计数寄存器 mn（TCRmn）
   TDRmn：定时器数据寄存器 mn（TDRmn）

图 5-46：外部事件计数器模式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）

	bit n	
TOm	TOmn 1/0	0：由 TOmn 输出 “0”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit n	
TOEm	TOEmn 1/0	0：停止由计数运行进行的 TOmn 输出。 1：允许由计数运行进行的 TOmn 输出。

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmn 位

TMRm0、TMRm5、TMRm7：固定为“0”。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-47：外部事件计数器功能时的操作步骤

	软件操作	硬件状态
Timer8 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 TM80EN 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 （NFEN1）的 对应位置 “0” （OFF）或者 “1” （ON）。 设定定时器模式寄存器 mn （TMRmn）（确定通道的运行模式）。 给定时器数据寄存器 mn （TDRmn）设定计数值。 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“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。
Timer8 停止	将 PER0 寄存器的 TM80EN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注：m：单元号（m=0） n：通道号 （n=0~7）

5.8.3 作为分频器的运行

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

TOmn 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/{(TDRmn 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/(TDRmn 的设定值+1)

在间隔定时器模式中，定时器计数寄存器（TCRmn）用作递减计数器。

在将定时器通道开始寄存器（TSm）的通道开始触发位（TSmn）置“1”后，通过检测到 TImn 的有效边沿将定时器数据寄存器（TDRmn）的值装入 TCRmn 寄存器。此时，如果定时器模式寄存器（TMRmn）的 MDmn0 位为“0”，就不输出 INTTMmn 并且 TOmn 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 TImn 引脚输入的有效边沿进行递减计数。如果 TCRmn 变为“0000H”，TOmn 就进行交替输出。同时，将 TDRmn 寄存器的值装入 TCRmn 寄存器并且继续计数。

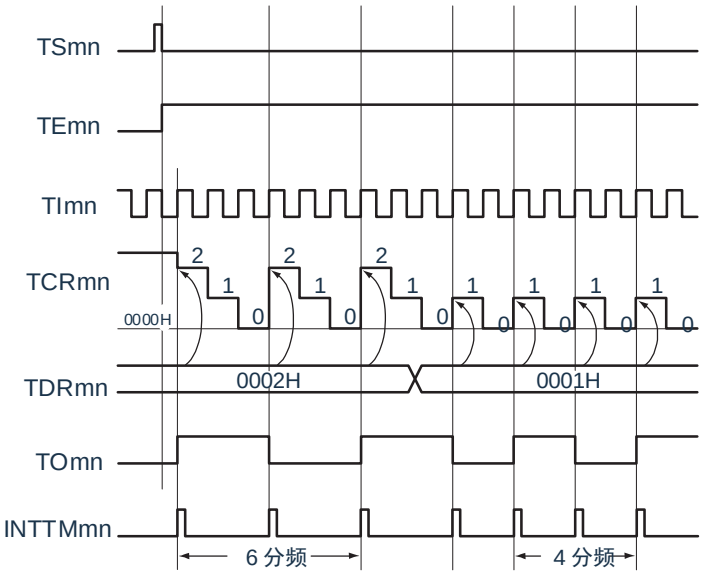
如果选择 TImn 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TOmn 输出的分频时钟周期。

TOmn 输出的时钟周期包含 1 个运行时钟周期的采样误差。

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

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

图 5-48：作为分频器运行的基本时序例子（MDmn0=1）



备注：TSmn：定时器通道开始寄存器（TSm）的 bit n

TEmn：定时器通道允许状态寄存器（TEm）的bit n

TImn：TImn 引脚输入信号

TCRmn：定时器计数寄存器（TCRmn）

TDRmn：定时器数据寄存器（TDRmn）

TOmn：TOmn引脚输出信号

m：单元号（m=0） n：通道号（n=0~7）

图 5-49：作为分频器运行时的寄存器设定内容例子（单元 0 的通道 0）

(a) 定时器模式寄存器 00（TMR00）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS001 1/0	CKS000 0	0	CCS00 1	0	STS002 0	STS001 0	STS000 0	CIS001 1/0	CIS000 1/0	0	0	MD003 0	MD002 0	MD001 0	MD000 1/0

通道 n 的运行模式
000B: 间隔定时器

开始计数时的运行设定
0：在开始计数时，不产生 INTTMmn 并且不反相定时器的输出。
1：在开始计数时，产生 INTTMmn 并且反相定时器的输出。

TImn 引脚输入边沿的选择
00B：检测下降沿。
01B：检测上升沿。
10B：检测双边沿。
11B：禁止设定。

开始触发的选择
000B: 只选择软件开始触发。

计数时钟的选择
1：选择TI00引脚输入的有效边沿。

运行时钟 （fMCK）的选择
00B：选择 CK00 作为通道 0 的运行时钟。
10B：选择 CK01 作为通道 1 的运行时钟。

(b) 定时器输出寄存器 0（TO0）

	bit 0	
TO0	TO00 1/0	0：由 TO00 输出“0”。 1：由 TO00 输出“1”。

(c) 定时器输出允许寄存器 0（TOE0）

	bit 0	
TOE0	TOE00 1/0	0：停止由计数运行进行的 TO00 输出。 1：允许由计数运行进行的 TO00 输出。

(d) 定时器输出电平寄存器 0（TOL0）

	bit 0	
TOL0	TOL00 0	0：在主控通道输出模式（TOM00=0）中置“0”。

(e) 定时器输出模式寄存器 0（TOM0）

	bit n	
TOM0	TOM00 0	0：设定主控通道输出模式。

图 5-50：分频器功能时的操作步骤(以通道 0 为例)

	软件操作	硬件状态
Timer8 初始设定		定时器单元 0 的输入时钟处于停止提供状态。 （停止提供时钟，不能写各寄存器）
	将外围允许寄存器 0（PER0）的 TM80EN 位置“1”。	定时器单元 0 的输入时钟处于提供状态，各通道处于运行停止状态。 （开始提供时钟，能写各寄存器）
	设定定时器时钟选择寄存器 0 （TPS0）。 确定 CK00 ~ CK03 的时钟频率。	
通道初始设定	将噪声滤波器允许寄存器(NFEN1)的 对应位置“0”（OFF）或者“1”（ON）。 设定定时器模式寄存器 00（TMR00）（确定通道的运行模式，选择检测边沿）。 给定时器数据寄存器 00 （TDR00）设定间隔(周期)值。	通道处于运行停止状态。 （提供时钟，消耗一部分电力）
	将定时器输出模式寄存器 0 （TOM0）的 TOM00 位 置 “0” （主控通道输出模式）。 将 TOL00 位置 “0”。 设定 TO00 位并且确定 TO00 输出的初始电平 将 TOE00 位置 “1”，允许 TO00 输出。 将端口寄存器和端口模式寄存器置 “0”	TO00 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为“0” 时，输出 TO00 初始设定的平。 因为通道处于运行停止状态，所以 TO00 变。 TO00 引脚输出 TO00 设定的电平。
开始运行	将 TOE00 位置 “1” （只限于重新开始运行）。 将 TS00 位置 “1”。 因为 TS00 位是触发位，所以自动返回到“0”。	TE00 位变为 “1” 并且开始计数。 将 TDR00 寄存器的值装入定时器计数寄存器 00（TCR00）。当 TMR00 寄存器的 MD000 位为 “1”时，产生 INTTM00 并且 TO00 进行交替输出。
运行中	能任意更改 TDR00 寄存器的设定值。 能随时读 TCR00 寄存器。 不使用 TSR00 寄存器。 能更改 TO0 寄存器和 TOE0 寄存器的设值。 禁止更改 TMR00 寄存器、 TOM00 位和 TOL00 位的设定值。	计数器 （TCR00）进行递减计数。如果计数到“0000H”，就再次将 TDR00 寄存器的值装入 TCR00 寄存器并且继续计数。当检测到 TCR00 为 “0000H” 时，产生 INTTM00 并且 TO00 进行交替输出。 此后，重复此运行。
停止运行	将 TT00 位置 “1”。 因为 TT00 位是触发位，所以自动返回到“0”。	TE00 位变为 “0” 并且停止计数。 TCR00 寄存器保持计数值而停止计数。 TO00 输出不被初始化而保持状态。
	将 TOE00 位置“0”并且给 TO00 位设定值。	TO00 引脚输出 TO00 设定的电平。
Timer8 停止	要保持 TO00 引脚输出电平的情况： 在给端口寄存器设定要保持的值后将 TO00 位置“0”。 不需要保持 TO00 引脚输出电平的情况：不需要设定。	通过端口功能保持 TO00 引脚的输出电平。
	将 PER0 寄存器的 TM80EN 位置 “0”。	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 （TO00 位变为 “0” 并且 TO00 引脚变为端口功能）

重新开始运行

5.8.4 作为输入脉冲间隔测量的运行

能在 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：

TImn 输入脉冲间隔=计数时钟的周期 x((10000HxTSRmn:OVF)+(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。

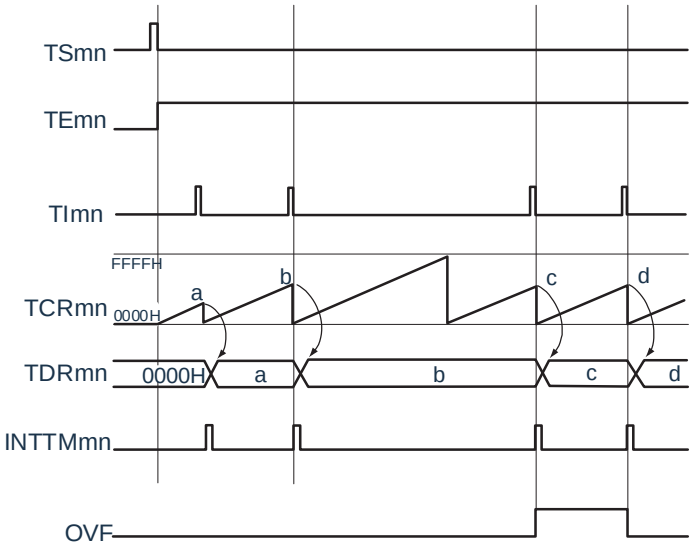
如果检测到 TImn 引脚输入的有效边沿，就将 TCRmn 寄存器的计数值传送（捕捉）到定时器数据寄存器 mn（TDRmn），同时将 TCRmn 寄存器清“0000H”，然后输出 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置“1”。如计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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

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

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

图 5-51：作为输入脉冲间隔测量的运行基本时序例子（MDmn0=0）



备注：

1. M：单元号（m=0） n：通道号（n=0~7）
2. TSmn：定时器通道开始寄存器 m（TSm）的 bitn
   TEmn：定时器通道允许状态寄存器 m（TEm）的 bitn
   TImn：TImn 引脚输入信号
   TCRmn：定时器计数寄存器 mn（TCRmn）
   TDRmn：定时器数据寄存器 mn（TDRmn）
   OVF：定时器状态寄存器 mn（TSRmn）的 bit 0

图 5-52：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）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	M/S注 0	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 0	MDmn2 1	MDmn1 0	MDmn0 1/0

通道 n 的运行模式
010B: 捕捉模式

开始计数时的运行设定
0：在开始计数时，不产生 INTTMmn
并且不反相定时器的输出。
1：在开始计数时，产生 INTTMmn
并且反相定时器的输出。

TImn 引脚输入边沿的选择
00B：检测下降沿
01B：检测上升沿
10B：检测双边沿
11B：禁止设定

捕捉触发的选择
001B：选择 TImn 引脚输入的有效边沿

MASTERmn位的设定（通道2）
0：独立通道运行

计数时钟的选择
0：选择运行时钟（F_{MCK}）。

运行时钟（F_{MCK}）的选择
00B：选择 CKm0作为通道n 的运行时钟。
10B：选择 CKm1作为通道n 的运行时钟。
01B：选择 CKm2作为通道1、3 的运行时钟（只有通道1、3 才能选择）。
11B：选择 CKm3作为通道1、3 的运行时钟（只有通道1、3 才能选择）。

(b) 定时器输出寄存器 m（TOm）

	bit n
TOm	TOmn 0

0：由 TOmn 输出 “0”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit n
TOEm	TOEmn 0

0：停止由计数运行进行的 TOmn 输出。

(d) 定时器输出电平寄存器 m（TOLm）

	bit n
TOLm	TOLmn 0

0：在主控通道输出模式 （TOMmn=0）中置 “0”。

(e) 定时器输出模式寄存器 m（TOMm）

	bit n
TOMm	TOMmn 0

0：设定主控通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmn 位
TMRm0、TMRm5、TMRm7：固定为 “0”。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-53：输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
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 的 TM80EN 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（NFEN1）的对应位置 “0” （OFF）或者 “1” （ON）。 设定定时器模式寄存器 mn （TMRmn）（确定通道的运行模式）。	通道处于运行停止状态。（提供时钟，消耗一部分电力）
开始运行	将 TSmn 位置 “1”。 因为 TSmn 位是触发位，所以自动返回到“0”。	TEmn 位变为 “1” 并且开始计数。 将定时器计数寄存器 mn（TCRmn）清 “0000H”。 当 TMRmn 寄存器的 MDmn0 位为 “1” 时，产生 INTTMmn。
运行中	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。 能随时读 TDRmn 寄存器。 能随时读 TCRmn 寄存器。 能随时读 TSRmn 寄存器。 禁止更改 TOMmn 位、 TOLmn 位、 TOmn 位和 TOEmn 位的设定值。	计数器（TCRmn）从 “0000H” 开始递增计数，如果检测到 TImn 引脚输入的有效边沿或者将 TSmn 位置 “1”，就将计数值传送（捕捉）到定时器数据寄存器 mn（TDRmn），同时将 TCRmn 寄存器清“0000H” 并且产生 INTTMmn。 此时，如果发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置位。如果没有发生上溢，就清除 OVF 位。此后，重复此运行。
停止运行	将 TTmn 位置 “1”。 因为 TTmn 位是触发位，所以自动返回“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。
TAU 停止	将 PER0 寄存器的 TM80EN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

重新开始运行

备注：m：单元号（m=0） n：通道号（n=0~7）

5.8.5 作为输入信号高低电平宽度测量的运行

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

能通过在 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。

TImn 输入的信号宽度=计数时钟的周期 x((10000HxTSRmn:OVF)+(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。

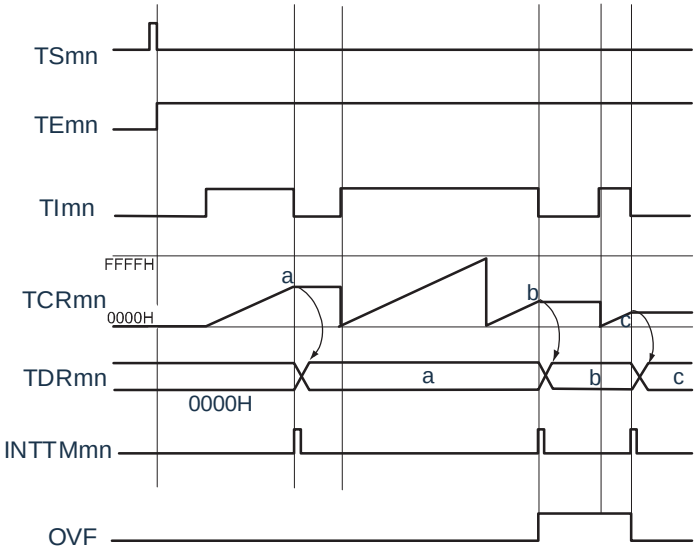
即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSRmn 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

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 5-54：作为输入信号高低电平宽度测量的运行基本时序例子



备注：

1. m：单元号（m=0） n：通道号（n=0~7）
2. TSmn：定时器通道开始寄存器m（TSm）的 bit n

   TEmn：定时器通道允许状态寄存器 m（TEm）的bit n

   TImn：TImn 引脚输入信号

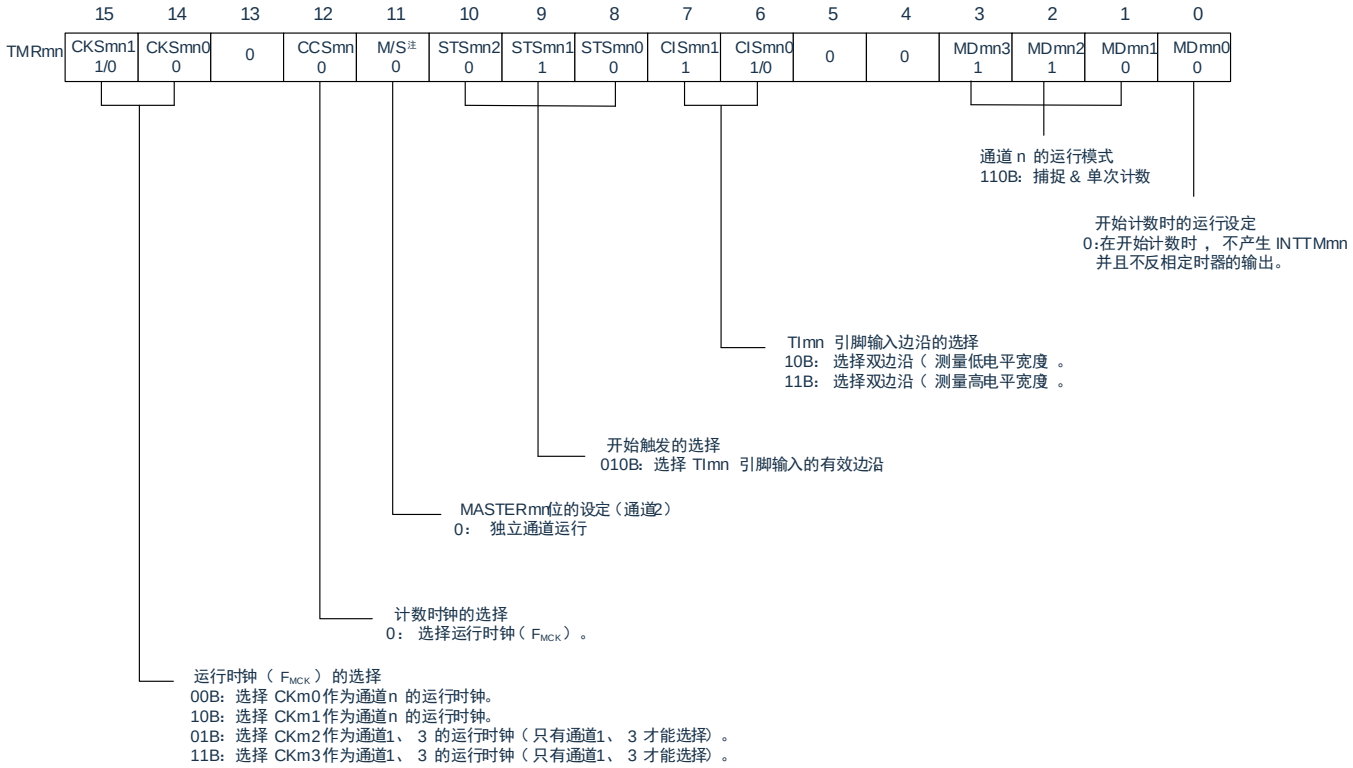
   TCRmn：定时器计数寄存器 mn（TCRmn）

   TDRmn：定时器数据寄存器 mn（TDRmn）

   OVF：定时器状态寄存器 mn（TSRmn）的 bit0

图 5-55：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）

	bit n	
TOm	Tomn 0	0：由 TOmn 输出“0”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmn 位

TMRm0、TMRm5、TMRm7：固定为“0”。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-56：输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
Timer8 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 TM8mEN 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（NFEN1）的对应位置 “0” （OFF）或者 “1” （ON）。 设定定时器模式寄存器 mn （TMRmn）（确定通道的运行模式）。 将 TOEmn 位置 “0”，并且停止 TOmn 的运行。	通道处于运行停止状态。（提供时钟，消耗一部分电力）
开始运行	将 TSmn 位置 “1”。 因为 TSmn 位是触发位，所以自动返回到 “0”。	TEmn 位变为 “1” 并且进入 TImn 引脚的开始边沿检测等待状态。
	检测 TImn 引脚输入的计数开始边沿。	将定时器计数寄存器 mn （TCRmn）清 “0000H” 并 且开始递增计数。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 禁止更改 TMRmn 寄存器、TOMmn 位、TOLmn 位、TOmn 位和 TOEmn 位的设定值。	在检测到 TImn 引脚的开始边沿后，计数器（TCRmn）从 “0000H” 开始递增计数。如果检测到 TImn 引脚的捕捉边沿，就将计数值传送到定时器数据寄存器 mn （TDRmn），并且产生 INTTMmn。 此时，如果发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置位。如果没有发生上溢，就清除 OVF 位。TCRmn 寄存器在检测到下一个 TImn 引脚的开始边沿前停止计数。 此后，重复此运行。
停止运行	将 TTmn 位置 “1”。 因为 TTmn 位是触发位，所以自动返回到 “0”。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 保持 TSRmn 寄存器的 OVF 位。
Timer8 停止	将 PER0 寄存器的 TM8mEN 位置 “0”。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

重新开始运行

备注：m：单元号（m=0） n：通道号（n=0~7）

5.8.6 作为延迟计数器的运行

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

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

中断产生周期能用以下计算式进行计算：

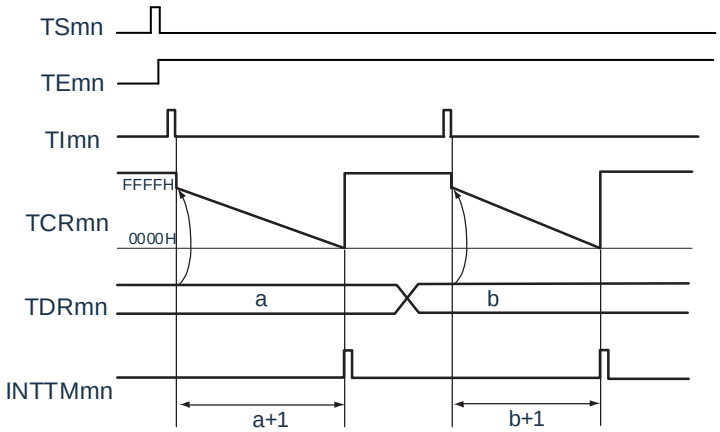
INTTMmn（定时器中断）的产生周期=计数时钟的周期×(TDRmn 的设定值+1)

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）置“1”，TEmn7 位就变为“1”，并且进入 TImn 引脚的有效边沿检测等待状态。通过 TImn 引脚输入的有效边沿检测，开始 TCRmn 寄存器的运行，并且装入定时器数据寄存器 mn（TDRmn）的值。TCRmn 寄存器通过计数时钟，从装入的 TDRmn 寄存器的值开始递减计数。如果 TCRmn 变为“0000H”，就输出 INTTMmn，并且在检测到下一个 TImn 引脚输入的有效边沿前停止计数。

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

图 5-57：作为延迟计数器的运行基本时序例子



备注：

1. m：单元号（m=0） n：通道号（n=0~7）
2. TSmn：定时器通道开始寄存器 m（TSm）的 bit n
   TEmn：定时器通道允许状态寄存器 m（TEm）的 bit n
   TImn：TImn 引脚输入信号
   TCRmn：定时器计数寄存器 mn（TCRmn）
   TDRmn：定时器数据寄存器 mn（TDRmn）

图 5-58：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）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 1/0	0	CCSmn 0	M/S	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 1	MDmn2 0	MDmn1 0	MDmn0 1/0

通道 n 的运行模式
100B： 单次计数模式

运行中的开始触发
0： 触发输入无效
1： 触发输入有效

TImn 引脚输入边沿的选择
00B： 检测下降沿。
01B： 检测上升沿。
10B： 检测双边沿。
11B： 禁止设定。

开始触发的选择
001B ：选择 TImn 引脚输入的有效边沿

MASTERmn位的设定（通道2）
0： 独立通道运行

计数时钟的选择
0： 选择运行时钟（ F_{MCK} ）。

运行时钟（ F_{MCK} ）的选择
00B：选择 CKm0作为通道n 的运行时钟。
10B：选择 CKm1作为通道n 的运行时钟。
01B：选择 CKm2作为通道1、 3 的运行时钟（只有通道1、 3 才能选择 。
11B：选择 CKm3作为通道1、 3 的运行时钟（只有通道1、 3 才能选择 。

(b) 定时器输出寄存器 m（TOm）

	bit n
TOm	TOmn 0

0：由 TOmn 输出“0”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit n
TOEm	TOEmn 0

0：停止由计数运行进行的 TOmn 输出。

(d) 定时器输出电平寄存器 m（TOLm）

	bit n
TOLm	TOLmn 0

0：在主控通道输出模式 （TOMmn=0）中置 “0”。

(e) 定时器输出模式寄存器 m（TOMm）

	bit n
TOMm	TOMmn 0

0：设定主控通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmn 位

TMRm0、TMRm5、TMRm7：固定为“0”。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-59：延迟计数器功能时的操作步骤

	软件操作	硬件状态
Timer8 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 TM80EN 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（NFEN1）的 对应位置 “0” （OFF）或者 “1” （ON）。 设定定时器模式寄存器 mn （TMRmn）（确定通道 n 的运行模式）。 给定时器数据寄存器 mn （TDRmn）设定输出延迟 时间。 将 TOEmn 位置 “0” 并且停止 TOmn 的运行。	
开始运行	将 TSmn 位置 “1” 。 因为 TSmn 位是触发位，所以自动返回到 “0” 。	TEmn 位变为 “1” 并且进入开始触发（检测 TImn 引 脚输入的有效边沿或者将 TSmn 位置 “1” ）的检测等 待状态。
	通过检测到下一个开始触发，开始递减计数。 • 检测 TImn 引脚输入的有效边沿。 • 通过软件将 TSmn 位置 “1” 。	将 TDRmn 寄存器的值装入定时器计数寄存器 mn（TCRmn）。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。	计数器（TCRmn）进行递减计数。如果 TCRmn 计数到 “0000H” ，就产生 INTTMmn，并且在检测到下 一次开始触发（检测 TImn 引脚输入的有效边沿或 者将 TSmn 位置 “1” ）前 TCRmn 为 “0000H” 而停止 计数。
停止运行	将 TTmn 位置 “1” 。 运行 因为 TTmn 位是触发位，所以自动返回到 “0” 。	TEmn 位变为 “0” 并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
Timer8 停止	将 PER0 寄存器的 TM80EN 位置 “0” 。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注：m：单元号（m=0） n：通道号（n=0~7）

5.9 通用定时器单元的多通道联动运行功能

5.9.1 作为单触发脉冲输出功能的运行

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

延迟={TDRmn（主控）的设定值+2}×计数时钟周期
脉宽={TDRmp（从属）的设定值}×计数时钟周期

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

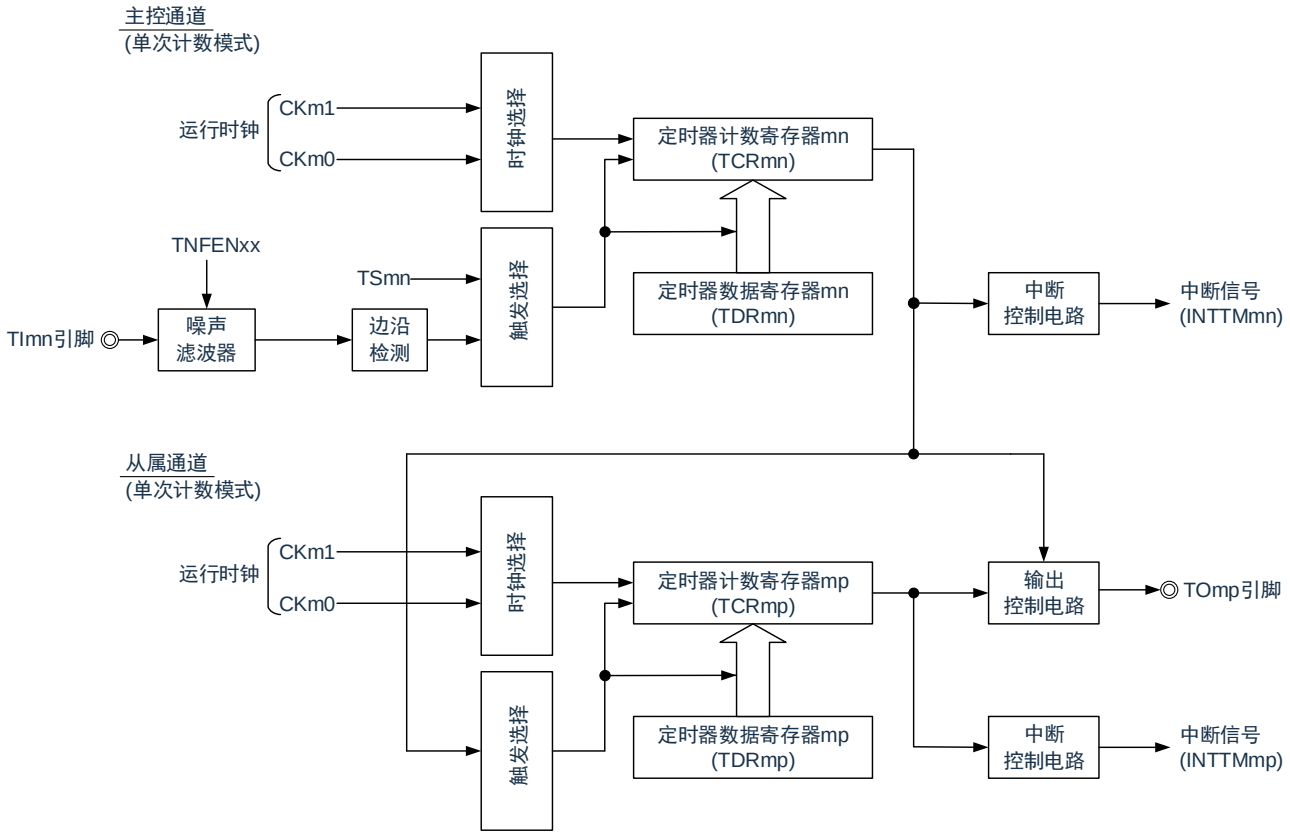
在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的 INTTMmn 作为开始触发，从属通道的 TCRmp 寄存器开始运行并且装入 TDRmp 寄存器的值。TCRmp 寄存器通过计数时钟，从装入的 TDRmp 寄存器值开始递减计数。如果计数值变为“0000H”，就输出 INTTMmp，并且在检测到下一个开始触发（主控通道的 INTTMmn）前停止计数。在从主控通道产生 INTTMmn 并且经过 1 个计数时钟后，TOmp 的输出电平变为有效电平，如果 TCRmp 变为“0000H”，就变为无效电平。

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

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

备注：m：单元号（m=0）n：主控通道号（n=0、2、4、6）
P：从属通道号（n＜p≤7）

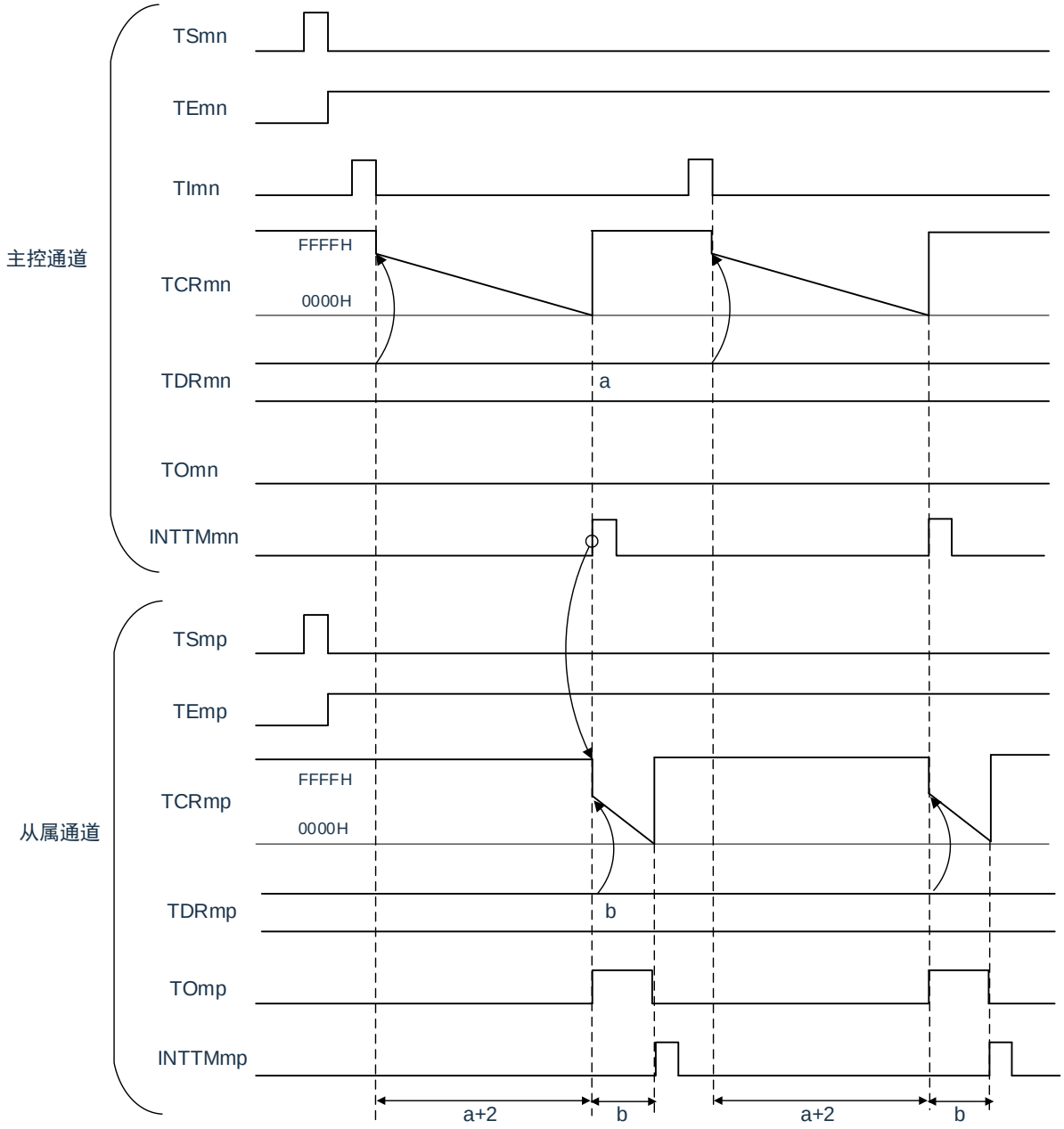
图 5-60：作为单触发脉冲输出功能运行的框图



备注：m：单元号（m=0）n：主控通道号（n=0、2、4、6）

P：从属通道号（n＜p≤7）

图 5-61：作为单触发脉冲输出功能的运行基本时序例子



备注：

1. m：单元号（m=0）n：主控通道号（n=0、2、4、6）
   p：从属通道号（n＜p≤7）
2. TSmn、TSmp：定时器通道开始寄存器 m（TSm）的 bit n、p
   TEmn、TEmp：定时器通道允许状态寄存器 m（TEm）的 bit n、p
   TImn、TImp：TImn 引脚和 TImp 引脚的输入信号
   TCRmn、TCRmp：定时器计数寄存器 mn、mp（TCRmn、TCRmp）
   TDRmn、TDRmp：定时器数据寄存器 mn、mp（TDRmn、TDRmp）
   TOmn、TOmp：TOmn 引脚和 TOmp 引脚的输出信号

图 5-62：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）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MAS TERmn注 1	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 1	MDmn2 0	MDmn1 0	MDmn0 0

通道 n 的运行模式
100B：单次计数模式

运行中的开始触发
0：触发输入无效。

TImn 引脚输入边沿的选择
00B：检测下降沿。
01B：检测上升沿。
10B：检测双边沿。
11B：禁止设定。

开始触发的选择
001B：选择 TImn 引脚输入的有效边沿。

MASTERmn位的设定（通道2）
1：主控通道

计数时钟的选择
0：选择运行时钟（fMCK）。

运行时钟（fMCK）的选择
00B：选择 CKm0 作为通道 n 的运行时钟。
10B：选择 CKm1 作为通道 n 的运行时钟。

(b) 定时器输出寄存器 m（TOm）

	bit n	
TOm	TOmn 0	0：由 TOmn 输出 “0”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 时器输出电平寄存器 m（TOLm）

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

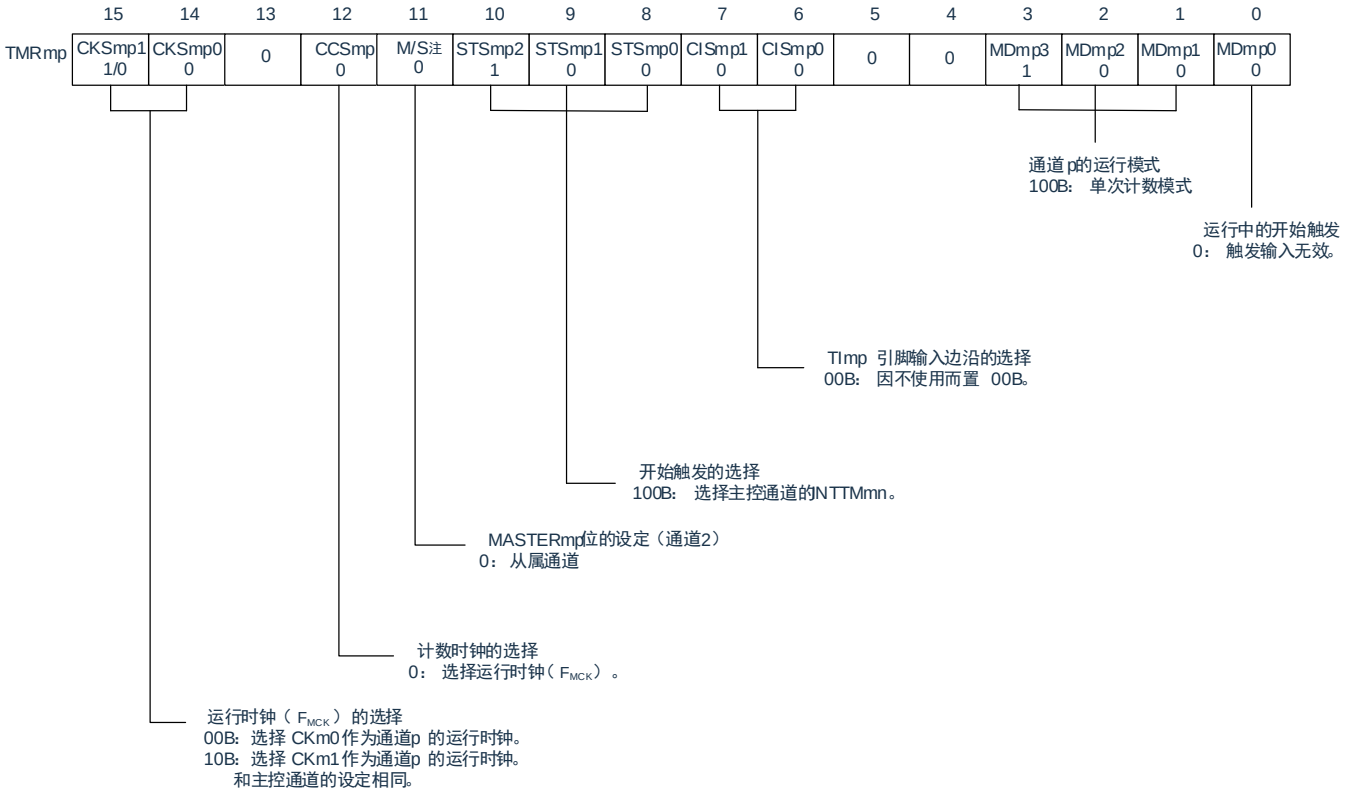
(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6）

图 5-63：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 1/0	0：由 TOmp 输出“0”。 1：由 TOmp 输出“1”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器m（TOMm）

	bit p	
TOMm	TOMmp 1	1：设定从属通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmp 位

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6） p：从属通道号（n＜p≤7）

图 5-64：单触发脉冲输出功能时的操作步骤(1/2)

	软件操作	硬件状态
Timer8 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 TM80EN 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 和 CKm1 的时钟频率。	
通道初始设定	将噪声滤波器允许寄存器（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。

备注：m：单元号（m=0） n：通道号（n=0~7）

图 5-65：单触发脉冲输出功能时的操作步骤(2/2)

	软件操作	硬件状态
开始运行	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。
运行中	将 TTmn 位 （主控）和 TTmp 位 （从属）同时置“1”。 因为 TTmn 位和 TTmp 位是触发位，所以自动返回到 “0”。	TEmn 位和 TEmp 位都变为 “0” 并且停止计数。 TCRmn 寄存器和 TCRmp 寄存器保持计数值而停 止计数。 TOmp 输出不被初始化而保持状态。
	将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设 定值。	TOmp 引脚输出 TOmp 设定的电平。
停止运行	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。
Timer8 停止	将 PER0 寄存器的 TM80EN 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）

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6） p：从属通道号（n＜p≤7）

5.9.2 作为PWM功能的运行

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

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）。

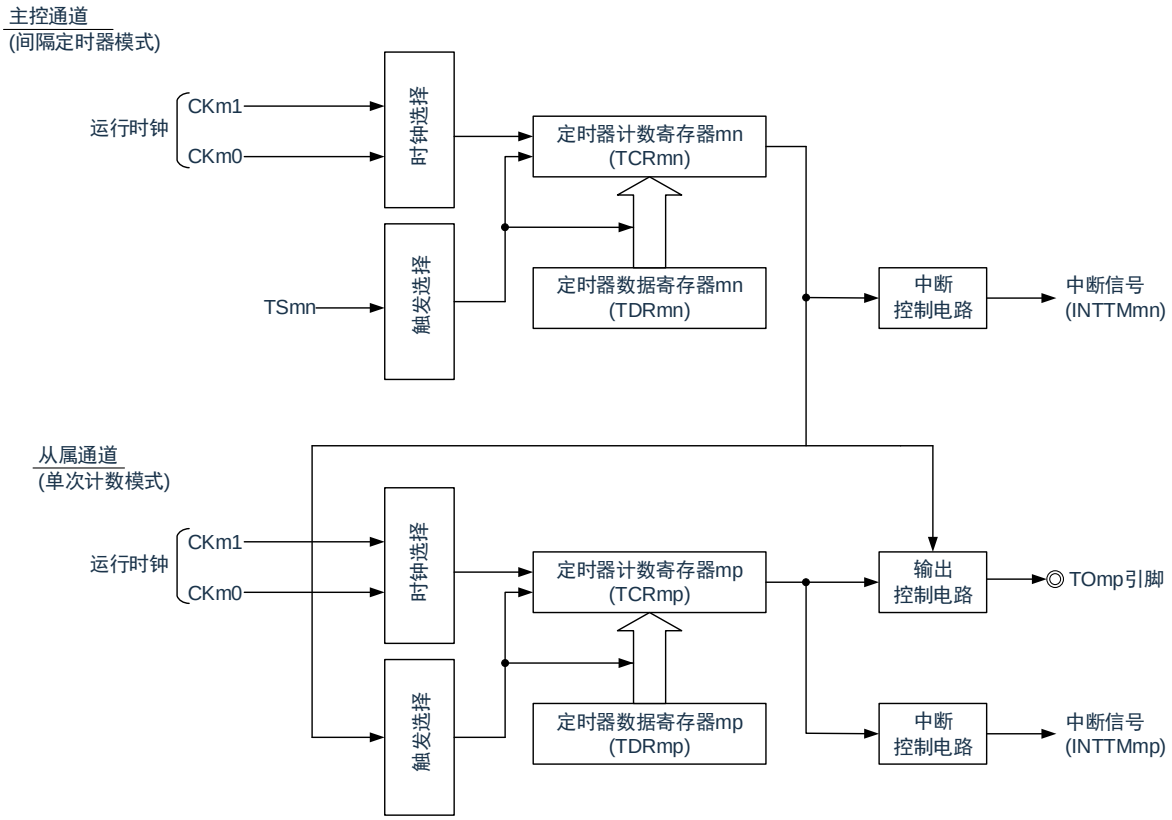
当用作 PWM 功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出（TOmp）的占空比。

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。

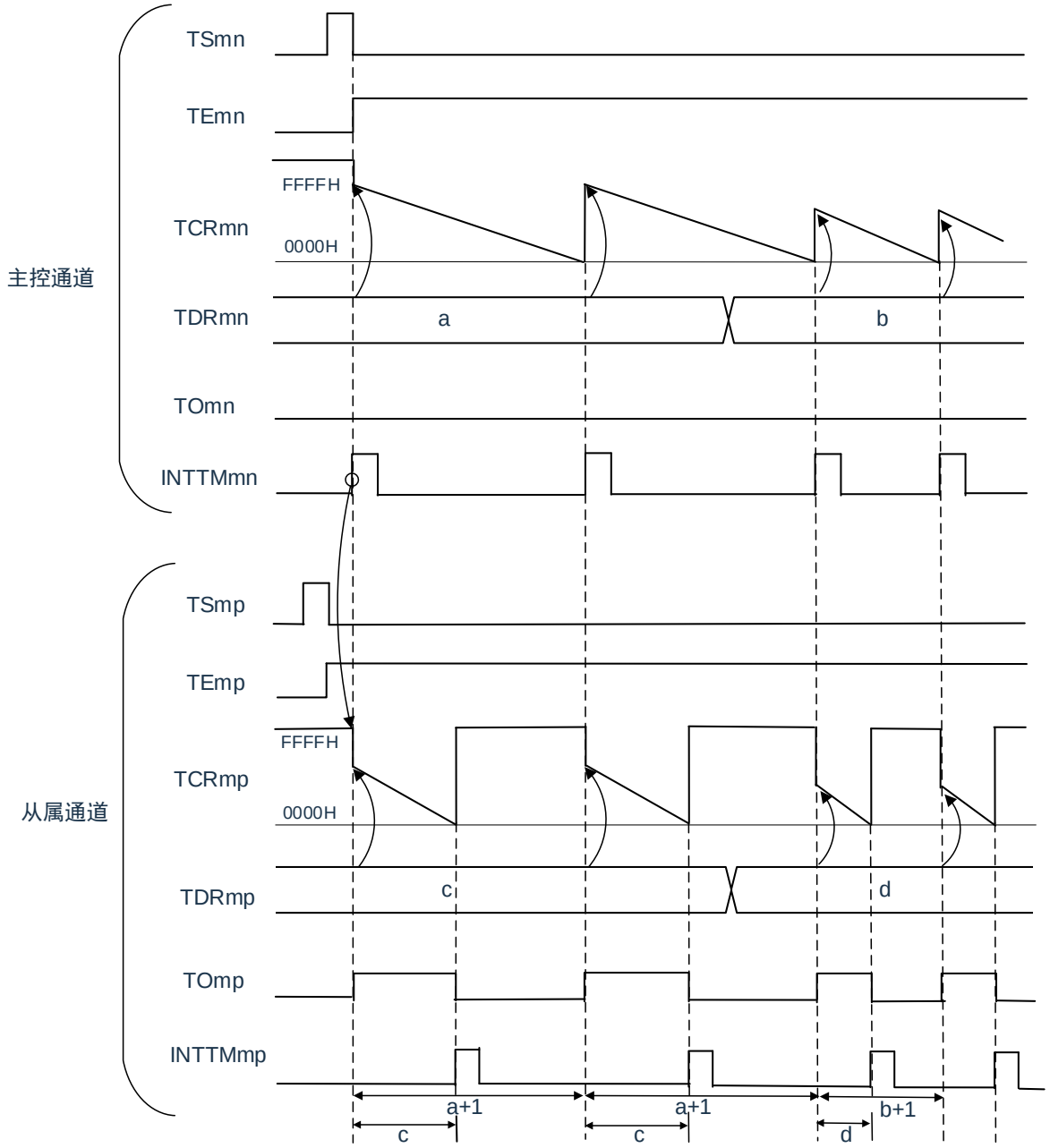
备注：m：单元号（m=0）n：主控通道号（n=0、2、4、6）p：从属通道号（m=0 时：n＜p≤7）

图 5-66：作为 PWM 功能运行的框图



备注：m：单元号（m=0）n：主控通道号（n=0、2、4、6）p：从属通道号（n＜p≤7）

图 5-67：作为 PWM 功能的运行基本时序例子

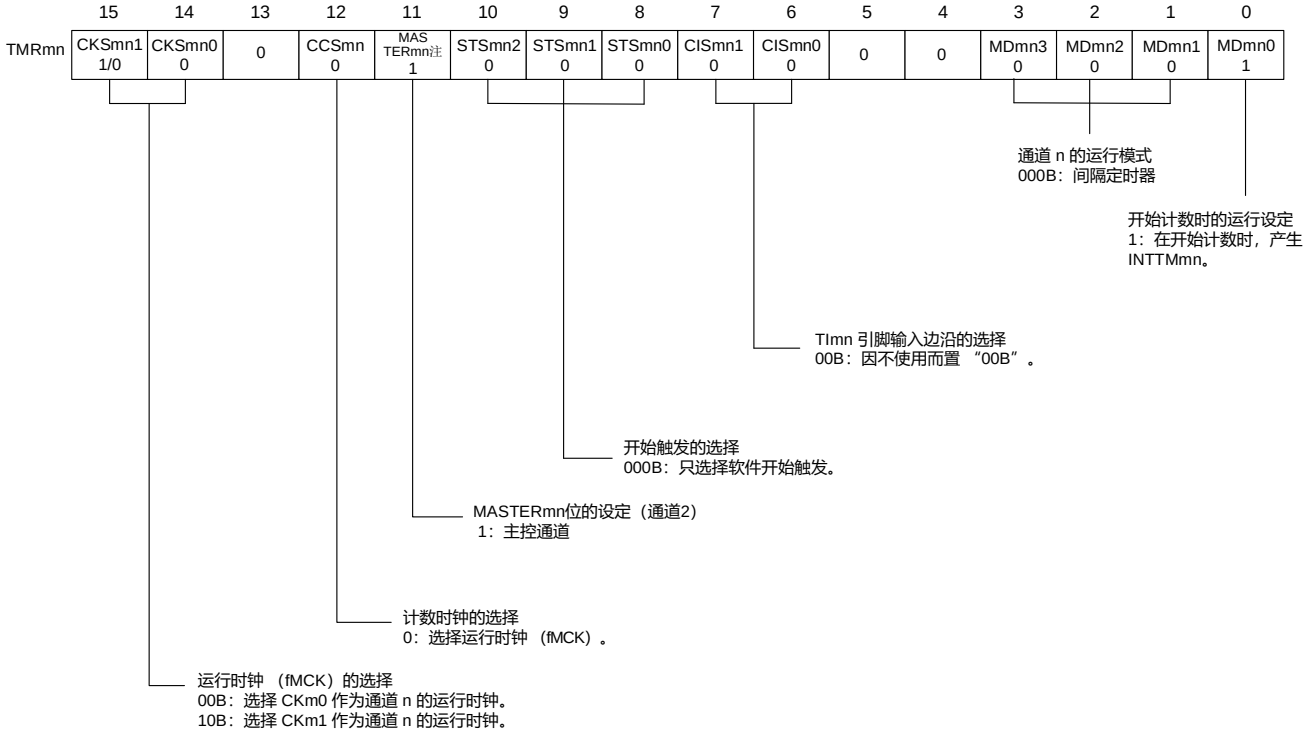


备注：

1. m：单元号（m=0）n：主控通道号（n=0、2、4、6）
   P：从属通道号（n＜p≤7）
2. TSmn、TSmp：定时器通道开始寄存器 m（TSm）的 bitn、p
   TEmn、Temp：定时器通道允许状态寄存器 m（TEm）的 bitn、p
   TCRmn、TCRmp：定时器计数寄存器 mn、mp（TCRmn、TCRmp）
   TDRmn、TDRmp：定时器数据寄存器 mn、mp（TDRmn、TDRmp）
   TOmn、TOmp：TOmn 引脚和 TOmp 引脚的输出信号

图 5-68：PWM 功能时（主控通道）的寄存器设定内容例子

(a) 定时器模式寄存器 mn（TMRmn）



(b) 定时器输出寄存器 m（TOm）

	bit n	
TOm	TOmn 0	0：由 TOmn 输出 “0”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

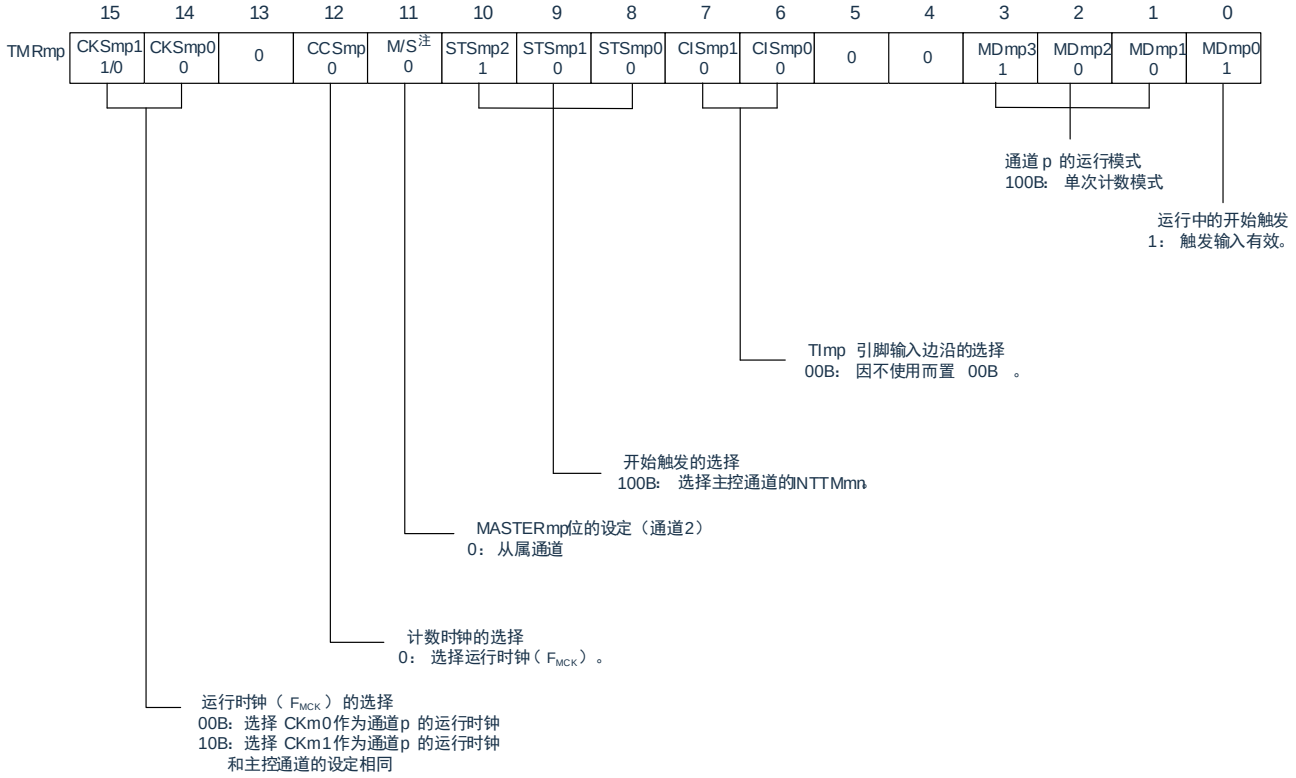
注：TMRm2：MASTERmn=1

TMRm0：固定为“0”。

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6）

图 5-69：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）

	bit p	
TOMm	TOMmp 1	1：设定从属通道输出模式。

注：TMRm2、TMRm4、TMRm6：MASTERmp 位

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6）p：从属通道号（m=0 时：n＜p≤7）

图 5-70：PWM 功能时的操作步骤(1/2)

	软件操作	硬件状态
Timer8 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 TM80EN 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 和 CKm1 的时钟频率。	
通道初始设定	设定使用的 2 个通道的定时器模式寄存器 mn、 mp（TMRmn、 TMRmp）（确定通道的运行模式）。 给主控通道的定时器数据寄存器 mn （TDRmn）设定间隔 （周期）值，并且给从属通道的 TDRmp 寄存器设定占空比的值。	通道处于运行停止状态。 （提供时钟，消耗一部分电力）
	从属通道的设定 将定时器输出模式寄存器 m （TOMm）的 TOMmp 位置 “1” （从属通道输出模式）。 设定 TOLmp 位。 设定 TOmp 位并且确定 TOmp 输出的初始电平。 将 TOEmp 位置 “1”，允许 TOmp 输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmp 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为“0”时，输出 TOmp 初始设定的电平。 因为通道处于运行停止状态，所以 TOmp 不变。 TOmp 引脚输出 TOmp 设定的电平。

图 5-71：PWM 功能时的操作步骤(2/2)

	软件操作	硬件状态
开始运行	将 TOEmp 位 （从属）置 “1” （只限于重新开始运 行）。 将定时器通道开始寄存器 m （TSm）的 TSmn 位（主控）和 TSmp 位 （从属）同时置 “1” 。 因为 TSmn 位和 TSmp 位是触发位，所以自动返回到 “0” 。	TEmn 位和 TEmp 位都变为 “1” 。 主控通道开始计数并且产生 INTTMmn。以此为触发，从属通道也开始计数。
运行中	禁止更改 TMRmn 寄存器和 TMRmp 寄存器以及 TOMmn 位、 TOMmp 位、 TOLmn 位和 TOLmp 位的设定值。 能在主控通道产生 INTTMmn 后更改 TDRmn 寄存器和 TDRmp 寄存器的设定值。 能随时读 TCRmn 寄存器和 TCRmp 寄存器。 不使用 TSRmn 寄存器和 TSRmp 寄存器。	主控通道将 TDRmn 寄存器的值装入定时器计数寄存器 mn （TCRmn） ，并且进行递减计数。 如果 TCRmn 计数到 “0000H” ，就产生 INTTMmn。同时，将 TDRmn 寄存器的值装入 TCRmn 寄存器，并且重新开始递减计数。 从属通道以主控通道的 INTTMmn 为触发，将 TDRmp 寄存器的值装入 TCRmp 寄存器，并且计数器进行递减计数。在从主控通道输出 INTTMmn 并且经过 1 个计数时钟后，将 TOmp 的输出电平置为有效电平。然后，如果 TCRmp 计数到 “0000H” ，就在将 TOmp 的输出电平置为无效电平后停止计数。 此后，重复此运行。
停止运行	将 TTmn 位 （主控）和 TTmp 位 （从属）同时置“1” 。 因为 TTmn 位和 TTmp 位是触发位，所以自动返回到 “0” 。	TEmn 位和 TEmp 位都变为 “0” 并且停止计数。 TCRmn 寄存器和 TCRmp 寄存器保持计数值而停 止计数。 TOmp 输出不被初始化而保持状态。
	将从属通道的 TOEmp 位置 “0” 并且给 TOmp 位设定值。	TOmp 引脚输出 TOmp 设定的电平。
Timer8 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 TM80EN 位置 “0” 。	定时器单元 m 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 （TOmp 位变为“0” 并且 TOmp 引脚变为端口功能）

重新开始运行

备注：m：单元号（m=0） n：主控通道号（n=0、2、4、6）p：从属通道号（m=0 时：n＜p≤7）

5.9.3 作为多重PWM输出功能的运行

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

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

脉冲周期={TDRmn（主控）的设定值+1}×计数时钟周期
占空比 1[%]={TDRmp（从属 1）的设定值}/{TDRmn（主控）的设定值+1}×100
占空比 2[%]={TDRmq（从属 2）的设定值}/{TDRmn（主控）的设定值+1}×100

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。

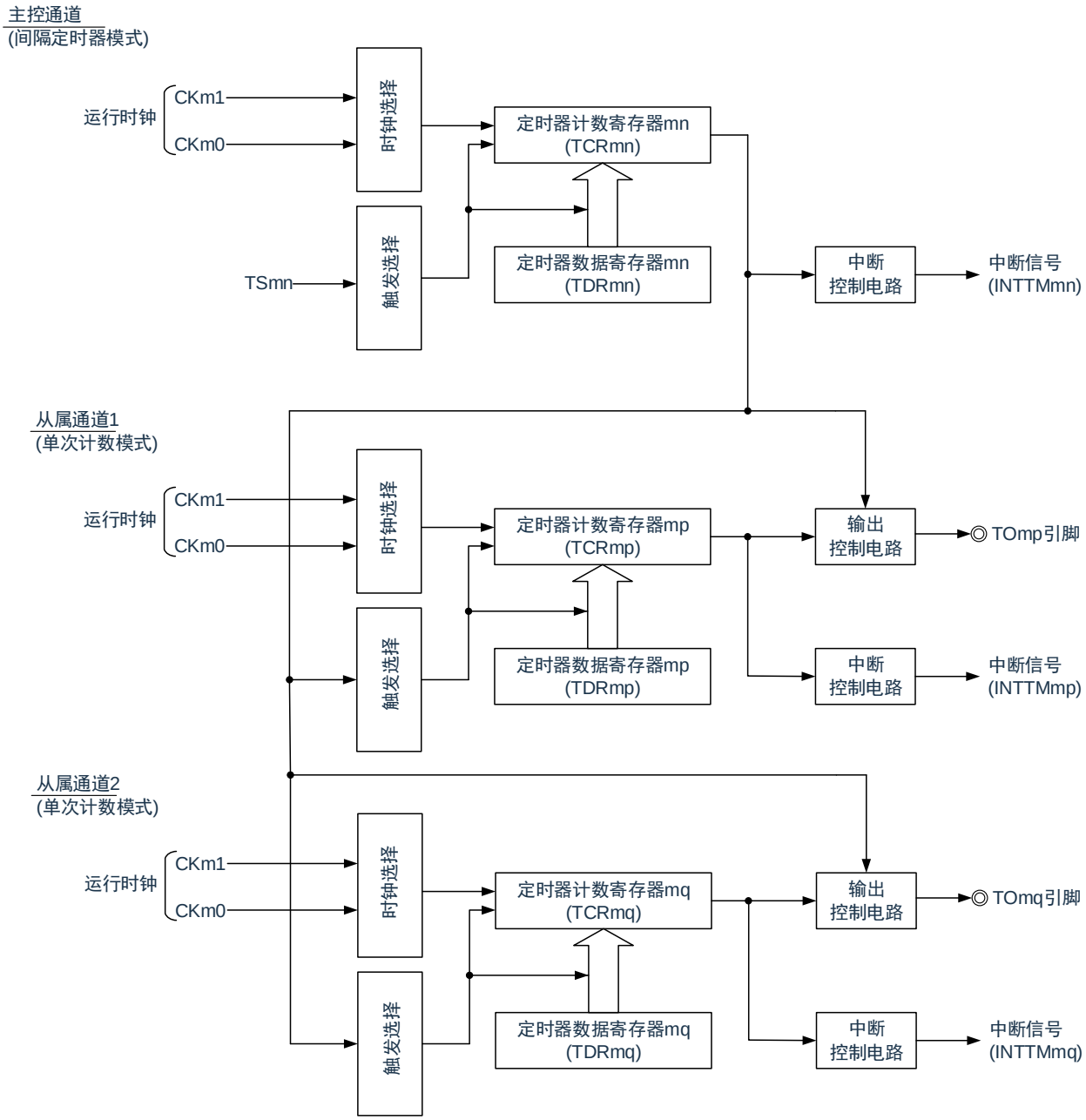
当通过如此的运行将通道 0 用作主控通道时，最多能同时输出 3 种 PWM 信号。

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

备注：m：单元号（m=0） n：主控通道号（n=0、2、4）

p：从属通道号 q：从属通道号 n＜p＜q≤7（p 和 q 是大于 n 的整数）

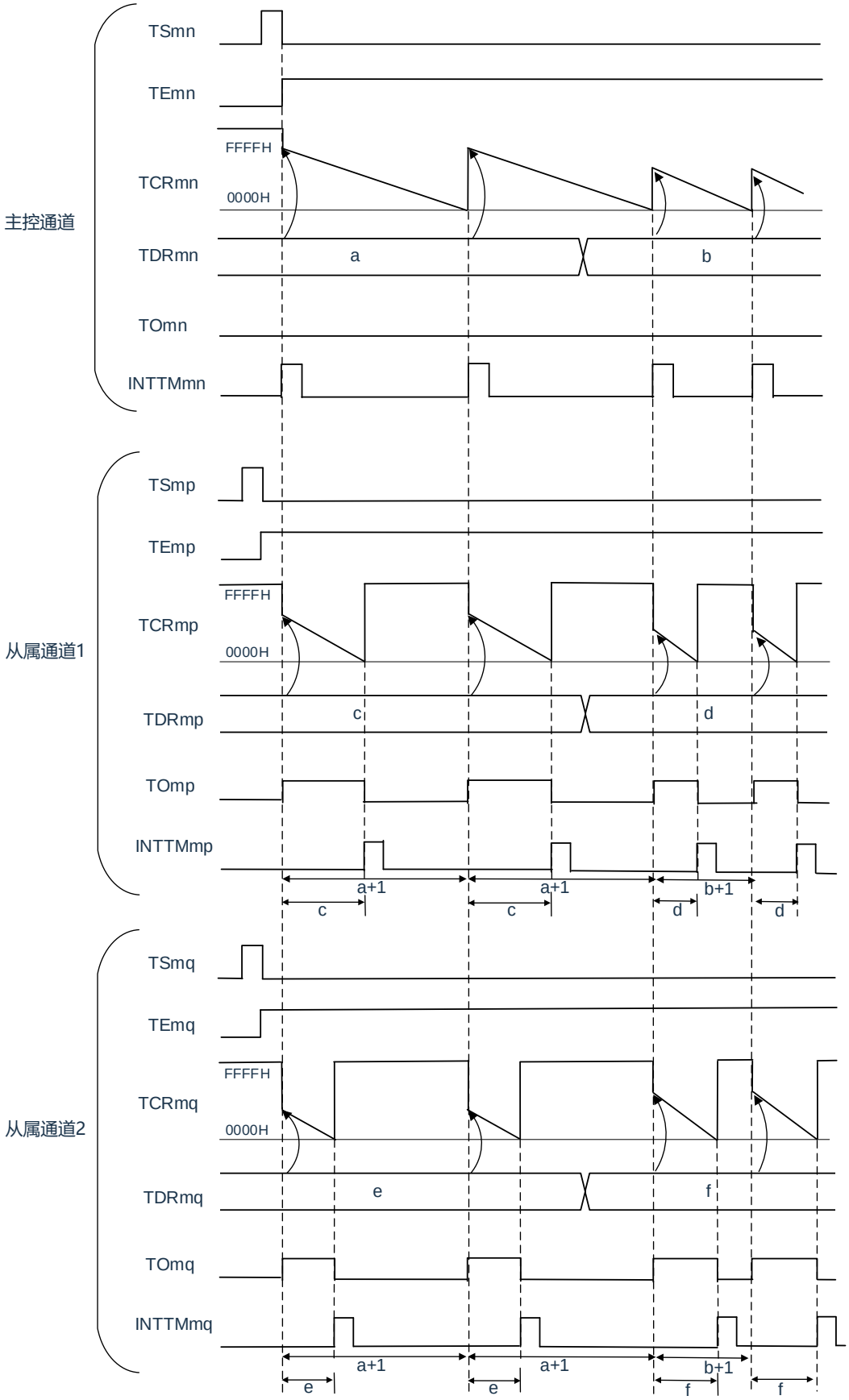
图 5-72：作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）



备注：m：单元号（m=0） n：主控通道号（n=0、2、4）

p：从属通道号 q：从属通道号 n＜p＜q≤7（p 和 q 是大于 n 的整数）

图 5-73：作为多重 PWM 输出功能的运行基本时序例子（输出 2 种 PWM 的情况）



备注：

1. m：单元号（m=0）

   n：主控通道号（n=0、2、4）

   p：从属通道号

   q：从属通道号

   m=0 时：n＜p＜q≤7（p 和 q 是大于 n 的整数）

2. TSmn、TSmp、TSmq：定时器通道开始寄存器 m（TSm）的 bitn、p、q

   TEmn、TEmp、TEmq：定时器通道允许状态寄存器 m（TEm）的 bitn、p、q

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

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

   TOmn、TOmp、TOmq：TOmn、TOmp、TOmq 引脚的输出信号

图 5-74：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）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MAS TERmn注 1	STSmn2 0	STSmn1 0	STSmn0 0	CISmn1 0	CISmn0 0	0	0	MDmn3 0	MDmn2 0	MDmn1 0	MDmn0 1

通道 n 的运行模式
000B：间隔定时器

开始计数时的运行设定
1：在开始计数时，产生 INTTMmn。

TImn 引脚输入边沿的选择
00B：因不使用而置 “00B”。

开始触发的选择
000B：只选择软件开始触发。

MASTERmn位的设定（通道2）
1：主控通道

计数时钟的选择
0：选择运行时钟（fMCK）。

运行时钟（fMCK）的选择
00B：选择 CKm0 作为通道 n 的运行时钟。
10B：选择 CKm1 作为通道 n 的运行时钟。

(b) 定时器输出寄存器 m（TOm）

	bit n	
TOm	TOmn 0	0：由 TOmn 输出 “0”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

	bit n	
TOMm	TOMmn 0	0：设定主控通道输出模式。

注：TMRm2、TMRm4：MASTERmn=1

TMRm0、TMRm5、TMRm7：固定为“0”。

备注：m：单元号（m=0） n：主控通道号（n=0、2、4）

图 5-75：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）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1 1/0	CKSmp0 0	0	CCSmp 0	M/S注 0	STSmp2 1	STSmp1 0	STSmp0 0	CISmp1 0	CISmp0 0	0	0	MDmp3 1	MDmp2 0	MDmp1 0	MDmp0 1
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1 1/0	CKSmq0 0	0	CCSmq 0	M/S注 0	STSmq2 1	STSmq1 0	STSmq0 0	CISmq1 0	CISmq0 0	0	0	MDmq3 1	MDmq2 0	MDmq1 0	MDmq0 1

通道p、 q 的运行模式
100B： 单次计数模式

运行中的开始触发
1： 触发输入有效

TImp、 TImq 引脚输入边沿的选择
00B： 因不使用而置 00B 。

开始触发的选择
100B： 选择主控通道的NTTMmn

MASTERmp位和 MASTERmq位的设定（通道2）
0： 从属通道

计数时钟的选择
0： 选择运行时钟（ F_{MCK} ）。

运行时钟（ F_{MCK} ）的选择
00B：选择 CKm0作为通道p、 q 的运行时钟
10B：选择 CKm1作为通道p、 q 的运行时钟
和主控通道的设定相同

(b) 定时器输出寄存器 m（TOm）

	bit q	bit p
TOm	TOmq 1/0	TOmp 1/0

0：由 TOmp 和 TOmq 输出“0”。

1：由 TOmp 和 TOmq 输出“1”。

(c) 定时器输出允许寄存器 m（TOEm）

	bit q	bit p
TOEm	TOEmq 1/0	TOEmp 1/0

0：停止由计数运行进行的 TOmp 和 TOmq 输出。

1：允许由计数运行进行的 TOmp 和 TOmq 输出。

(d) 定时器输出电平寄存器 m（TOLm）

	bit q	bit p
TOLm	TOELq 1/0	TOELp 1/0

0：正逻辑输出 （高电平有效）

1：负逻辑输出 （低电平有效）

(e) 定时器输出模式寄存器 m（TOMm）

	bit q	bit p
TOMm	TOMLq 1	TOMLp 1

1：设定从属通道输出模式。

注：TMRm2、TMRm4：MASTERmp 位、MASTERmq 位

备注：m：单元号（m=0） n：主控通道号（n=0、2、4）

p：从属通道号 q：从属通道号 m=0 时：n＜p＜q≤7（p 和 q 是大于 n 的整数）

图 5-76：多重PWM 输出功能时的操作步骤（输出2 种PWM 的情况）(1/2)

	软件操作	硬件状态
Timer8 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 TM80EN 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 和 CKm1 的时钟频率。	
通道初始设定	设定使用的各通道的定时器模式寄存器 mn、 mp、（TMRmn、 TMRmp、 ）（确定通道的运行模式）。 给主控通道的定时器数据寄存器 mn （TDRmn）设定间隔 （周期）值，并且给从属通道的 TDRmp 寄存器和 TDRmq 寄存器设定占空比的值。	通道处于运行停止状态。 （提供时钟，消耗一部分电力）
	从属通道的设定 将定时器输出模式寄存器 m（TOMm）的 TOMmp 位和 TOMmq 位置 “1” （从属通道输出模式）。 将 TOLmp 位和 TOLmq 位置 “0”。 设定 TOmp 位和 TOmq 位，并且确定 TOmp 和 TOmq 输出的初始电平。 将 TOEmp 位和 TOEmq 位置 “1”，允许 TOmp 和 TOmq 的输出。 将端口寄存器和端口模式寄存器置 “0”。	TOmp 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为“0”时，输出 TOmp 和 TOmq 初始设定的电平。 因为通道处于运行停止状态，所以 TOmp 和 TOmq 不变。 TOmp 引脚和 TOmq 引脚输出 TOmp 和 TOmq 设定 的电平。

图 5-77：多重 PWM 输出功能时的操作步骤（输出 2 种 PWM 的情况）(2/2)

	软件操作	硬件状态
开始运行	（只在重新开始运行时将 TOEmp 位和 TOEmq 位 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 位和 TEmp、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。
Timer8 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。	通过端口功能保持 TOmp 引脚的输出电平。
	将 PER0 寄存器的 TM80EN 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 位和 TOmq 位变为 “0” 并且 TOmp 引脚和 TOmq 引脚变为端口功能）

重新开始运行

备注：m：单元号（m=0） n：主控通道号（n=0、2、4）

p：从属通道号 q：从属通道号 n＜p＜q≤7（p 和 q 是大于 n 的整数）

5.10 使用通用定时器单元时的注意事项

5.10.1 使用定时器输出时的注意事项

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

详细内容请参照 “第 2 章 端口功能”。

第6章 定时器A

6.1 定时器A的功能

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

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

定时器 A 的规格和框图分别如图 6-1 和表 6-1 所示。

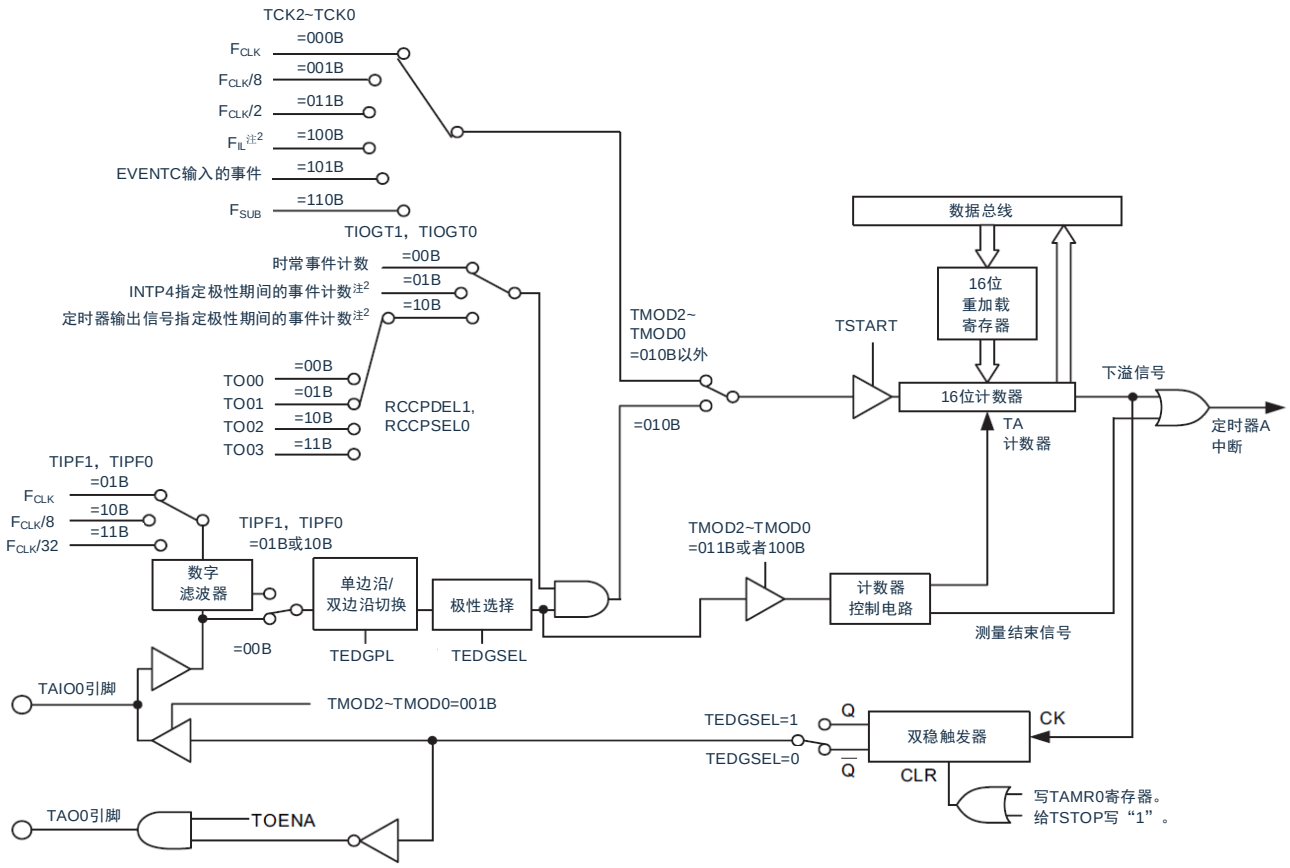
表 6-1：定时器 A 的规格

项目		内容
运行模式	定时器模式	对计数源进行计数。
	脉冲输出模式	对计数源进行计数，在定时器发生下溢时输出极性相反的脉冲。
	事件计数器模式	对外部事件进行计数。 也能在深度睡眠模式中运行。
	脉宽测量模式	测量外部输入的脉冲宽度。
	脉冲周期测量模式	测量外部输入的脉冲周期。
计数源（运行时钟）		可选择F_{CLK}、$F_{CLK}/2$、$F_{CLK}/8$、F_{IL}、F_{SUB}或者EVENTC输入的事件。
中断		•当计数器发生下溢时 •在脉宽测量模式中外部输入（TAIO）的有效宽度测量结束时 •在脉冲周期测量模式中输入外部输入（TAIO）的设置边沿时
选择功能		•与EVENTC的协作：可选择EVENTC输入的事件作为计数源。

6.2 定时器A的结构

定时器 A 的框图和引脚结构分别如图 6-1 和表 6-2 所示。

图 6-1：定时器 A 的框图



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

注 2：能通过 TAISR0 寄存器的 RCCPSEL2 位选择极性。

表 6-2：定时器 A 的引脚结构

引脚名	输入/输出	功能
INTP1	输入	定时器A的事件计数器模式控制
TAIO注	输入/输出	定时器A的外部事件输入和脉冲输出
TAO注	输出	定时器A的脉冲输出

注：能通过 PxxCFG 寄存器选择 TAO 引脚的配置，并且能通过该寄存器的 PB02CFG、PB04CFG、PB06CFG、PC01CFG、PC03CFG 位选择 TAIO 引脚的配置。详细内容请参照“第 2 章 端口功能”。

6.3 控制定时器A的寄存器

控制定时器 A 的寄存器如表 6-3 所示。

表 6-3：控制定时器 A的寄存器

寄存器名	符号
端口复用功能配置寄存器	PxxCFG
外围允许寄存器0	PER0
副系统时钟提供模式控制寄存器	OSMC
定时器A计数寄存器0[注]	TA0
定时器A控制寄存器0	TACR0
定时器AI/O控制寄存器0	TAIOC0
定时器A模式寄存器0	TAMR0
定时器A事件引脚选择寄存器0	TAISR0
端口寄存器x	Px
端口模式寄存器x	PMx

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

6.3.1 外围允许寄存器0（PER0）

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

要使用定时器 A 时，必须将 bit0（TMA）置“1”。

通过 8 位存储器操作指令设置 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 6-2：外围允许寄存器 0（PER0）的格式

地址：40020420H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

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

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

6.3.2 副系统时钟提供模式控制寄存器（OSMC）

能通过 WUTMMCK0 位选择定时器 A 的运行时钟。

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

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

图 6-3：副系统时钟提供模式控制寄存器（OSMC）的格式

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

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

6.3.3 定时器A计数寄存器0（TA0）

这是 16 位寄存器。如果写此寄存器，就将数据写到重加载寄存器。如果读此寄存器，就读计数值。重加载寄存器和计数器的状态因 TACR0 寄存器的 TSTART 位的值而变。详细内容请参照“6.4.1 重加载寄存器和计数器的改写”。

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

图 6-4：定时器 A 计数寄存器 0（TA0）的格式

地址：40044004H 复位后：FFFFH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA0																

—	功能	设置范围
bit15~0	16位计数器[注1,2]	0000H~FFFFH

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

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

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

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

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

6.3.4 定时器A控制寄存器0（TACR0）

TACR0 寄存器是控制寄存器 A 的计数和停止以及表示定时器 A 状态的寄存器。

通过 8 位存储器操作指令设置 TACR0 寄存器。在产生复位信号后，TACR0 寄存器的值变为“00H”。

图 6-5：定时器 A 控制寄存器 0（TACR0）的格式

地址：40044000H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
TACR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

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

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

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

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

TSTART	定时器A的计数开始[注2]
0	停止计数。
1	开始计数。
通过给TSTART位写“1”开始计数；通过给TSTART位写“0”停止计数。如果将TSTART位置“1”（开始计数），TCSTF位就与计数源同步变为“1”（正在计数）。另外，在给TSTART位写“0”后，TCSTF位与计数源同步变为“0”（停止计数）。详细内容请参照“6.5.1计数的开始和停止控制”。	

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

注 2：有关使用 TSTART 位和 TCSTF 位时的注意事项，请参照“6.5.1 计数的开始和停止控制”。

6.3.5 定时器AI/O控制寄存器0（TAIOC0）

TAIOC0 寄存器是设置定时器 A 的输入/输出的寄存器。通过 8 位存储器操作指令设置 TAIOC0 寄存器。

在产生复位信号后，TAIOC0 寄存器的值变为“00H”。

图 6-6：定时器 AI/O 控制寄存器 0（TAIOC0）的格式

地址：40044001H 复位后：00H R/W

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

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

TIPF1	TIPF0	TAIO输入滤波器的选择
0	0	没有滤波器。
0	1	有滤波器，通过F_{CLK}进行采样。
1	0	有滤波器，通过F_{CLK}/8进行采样。
1	1	有滤波器，通过F_{CLK}/32进行采样。
这些位指定TAIO输入滤波器的采样频率。对TAIO引脚的输入进行采样，如果采样值连续3次相同，就确定此值为输入值。		

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

TEDGSEL	输入/输出的极性切换
功能因运行模式而不同（参照表 6-4和表 6-5）。	

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

注 2：TIOGT0 位和 TIOGT1 位只在事件计数器模式中有效。

表 6-4：TAIO 输入/输出的边沿和极性切换

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

表 6-5：TAO 输出的极性切换

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

6.3.6 定时器A控制寄存器0（TAMR0）

TAMR0 寄存器是设置寄存器 A的运行模式的寄存器。通过 8 位存储器操作指令设置 TAMR0 寄存器。

在产生复位信号后，TAMR0 寄存器的值变为“00H”。

图 6-7：定时器 A 控制寄存器 0（TAMR0）的格式

地址：40044002H 复位后：00H R/W

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

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

TEDGPL	TAIO边沿极性的选择[注5]
0	单边沿
1	双边沿

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

注 1：如果选择事件计数器模式，就选择外部输入（TAIO）作为计数源，而与 TCK0~TCK2 位的设置无关。

注 2：不能在计数过程中切换计数源。如果要切换计数源，就必须在 TACR0 寄存器的 TSTART 位和 TCSTF 位都为“0”（停止计数）时进行切换。

注 3：只有在停止计数（TACR0 寄存器的 TSTART 位和 TCSTF 位都为“0”（停止计数））时才能更改运行模式，不能在计数过程中进行更改。

注 4：要选择 F_{IL} 作为计数源时，必须将副系统时钟提供模式控制寄存器（OSMC）的 WUTMMCK0 位置“1”。但是，当选择 F_{SUB} 作为实时时钟或者 12 位间隔定时器的计数源时，不能选择 F_{IL} 作为定时器 A 的计数源。

注 5：TEDGPL 位只在事件计数器模式中有效。

注 6：通过写 TAMR0 寄存器，对定时器 A 的 TAO 引脚和 TAIO 引脚的输出进行初始化。有关初始化时的输出电平，请参照“图 6-6：定时器 AI/O 控制寄存器 0（TAIOC0）的格式”的说明。

6.3.7 定时器A事件引脚选择寄存器0（TAISR0）

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

在产生复位信号后，TAISR0 寄存器的值变为“00H”。

图 6-8：定时器 A 事件引脚选择寄存器 0（TAISR0）的格式

地址：40044003H 复位后：00H R/W

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

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

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

注：RCCPSEL0~RCCPSEL2 位只在事件计数器模式中有效。

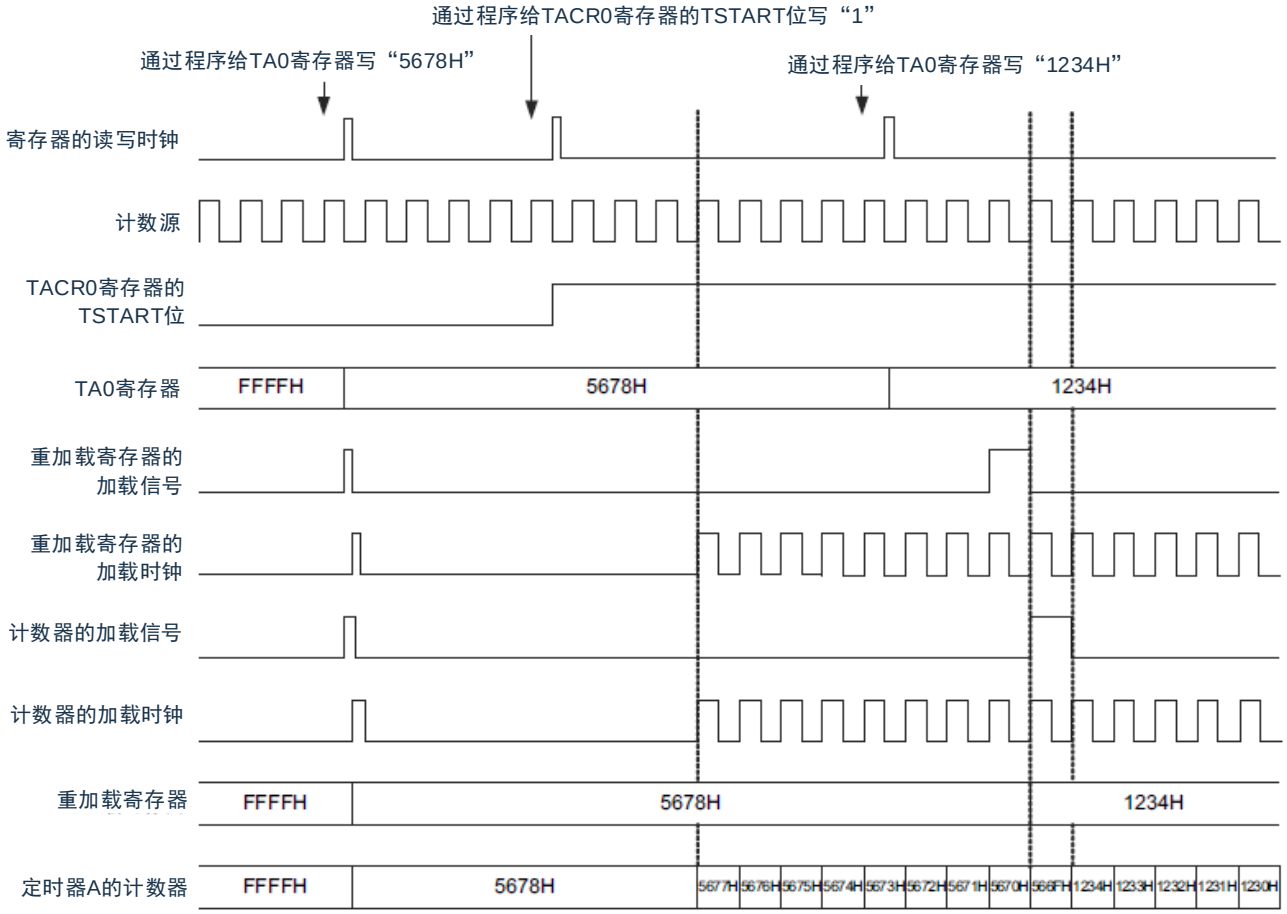
6.4 定时器A的运行

6.4.1 重加载寄存器和计数器的改写

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

由 TSTART 位的值决定的改写时序图如图 6-9 所示。

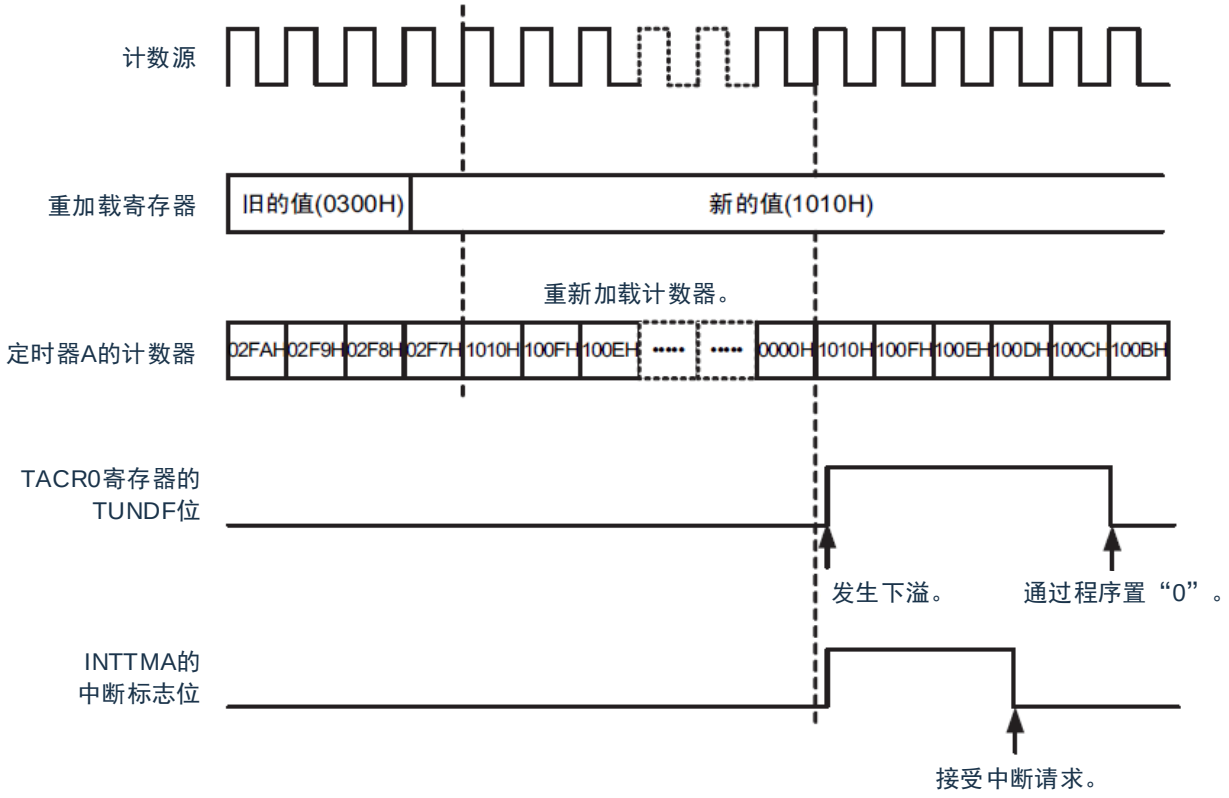
图 6-9：由 TSTART 位的值决定的改写时序图



6.4.2 定时器模式

这是通过 TAMR0 寄存器的 TCK0~TCK2 位选择的计数源进行递减计数的模式。在定时器模式中，每当输入计数源时计数值就减 1，如果计数值变为“0000H”并且输入下一个计数源，就发生下溢并且产生中断请求。定时器模式的运行例子如图 6-10 所示。

图 6-10：定时器模式的运行例子



6.4.3 脉冲输出模式

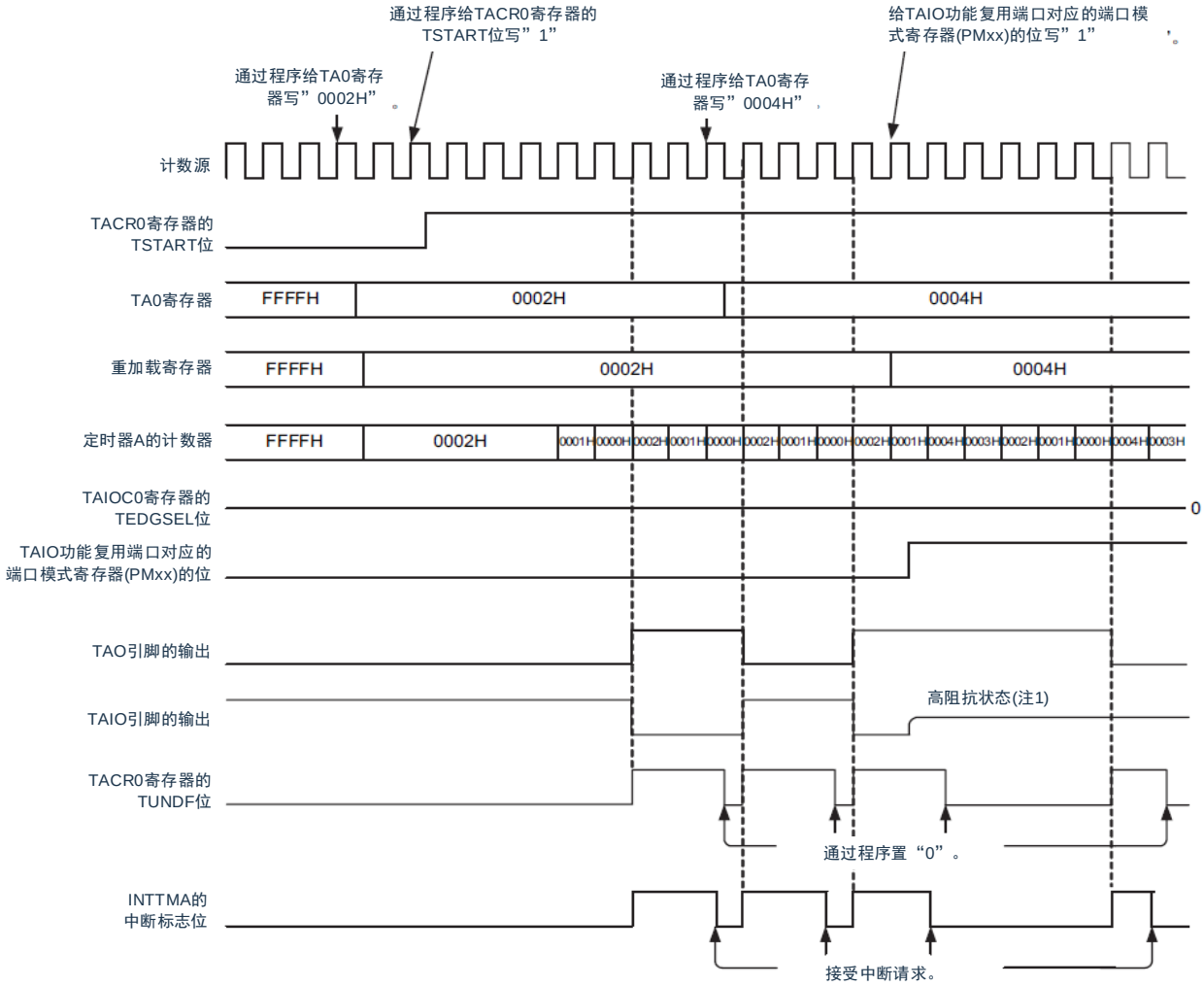
在此模式中，通过 TAMR0 寄存器的 TCK0~TCK2 位选择的计数源进行递减计数，每当发生下溢时，就将 TAIO 引脚和 TAO 引脚的输出电平反相输出。

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

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

另外，能通过 TAIOC0 寄存器的 TEDGSEL 位选择输出电平。脉冲输出模式的运行例子如图 6-11 所示

图 6-11：脉冲输出模式的运行例子



注1：通过被选择为TAIO功能的端口输出允许控制，变为高阻抗状态。

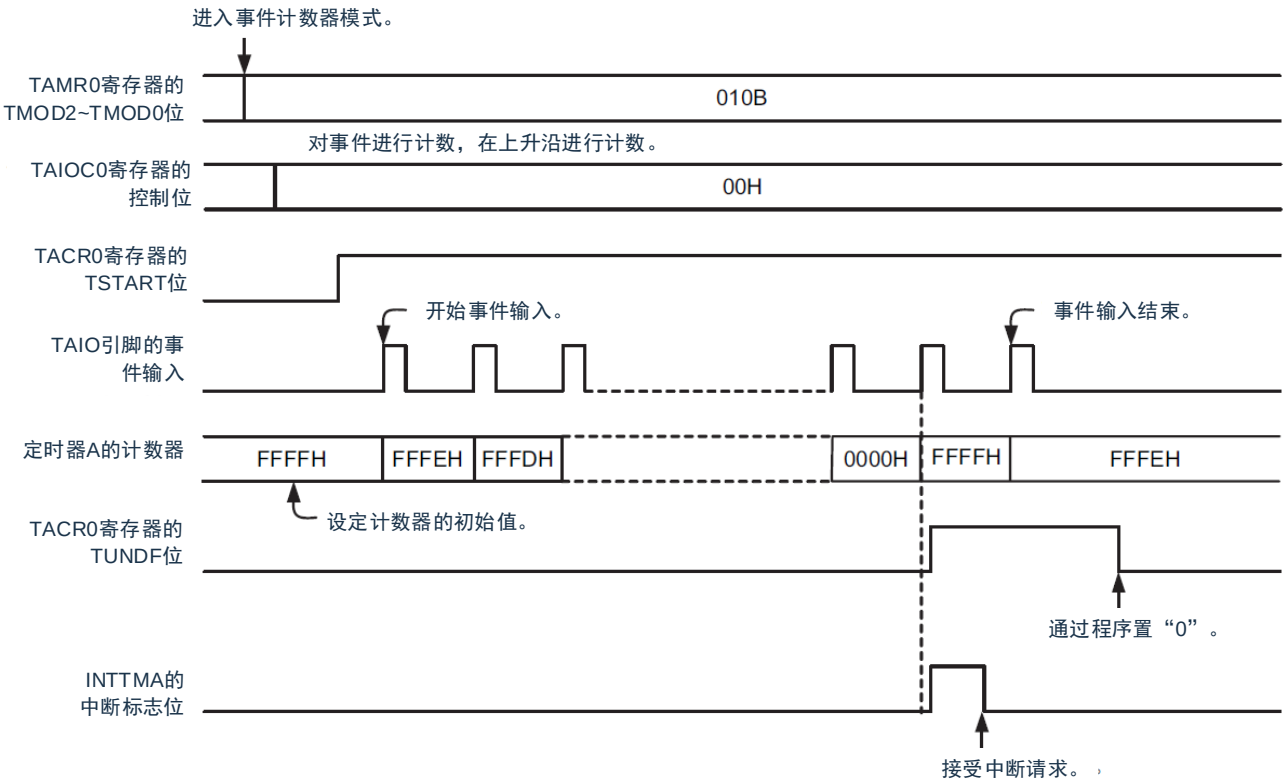
6.4.4 事件计数器模式

这是通过 TAIO 引脚输入的外部事件信号（计数源）进行递减计数的模式。

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

即使在事件计数器模式中 TAO 引脚也能进行交替输出。要使用事件计数器模式时，请参照“6.5.5 TAO 引脚和 TAIO 引脚的设置步骤”。事件计数器模式的运行例子 1 如图 6-12 所示。

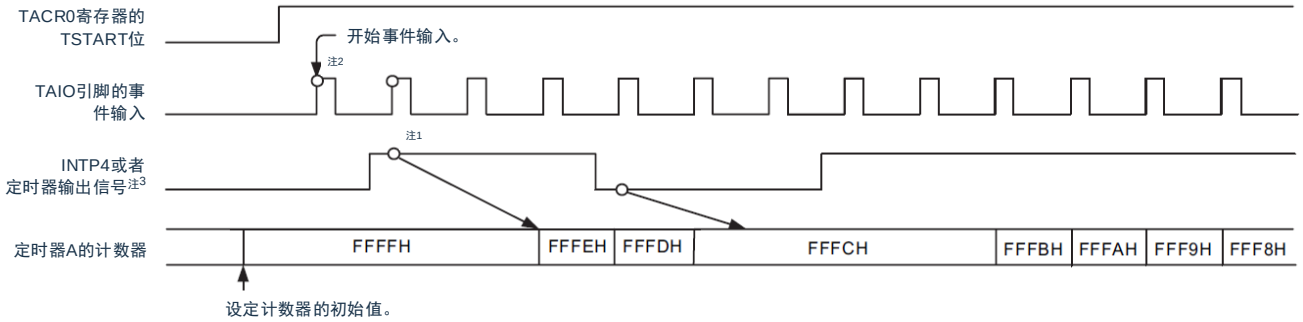
图 6-12：事件计数器模式的运行例子 1



事件计数器模式中的指定时间计数（TAIOC0 寄存器的 TIOGT1 位和 TIOGT0 位为“01B”或者“10B”）的运行例子如图 6-13 所示。

图 6-13：事件计数器模式的运行例子 2

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



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

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

6.4.5 脉宽测量模式

这是测量 TAIO 引脚输入的外部信号脉宽的模式。

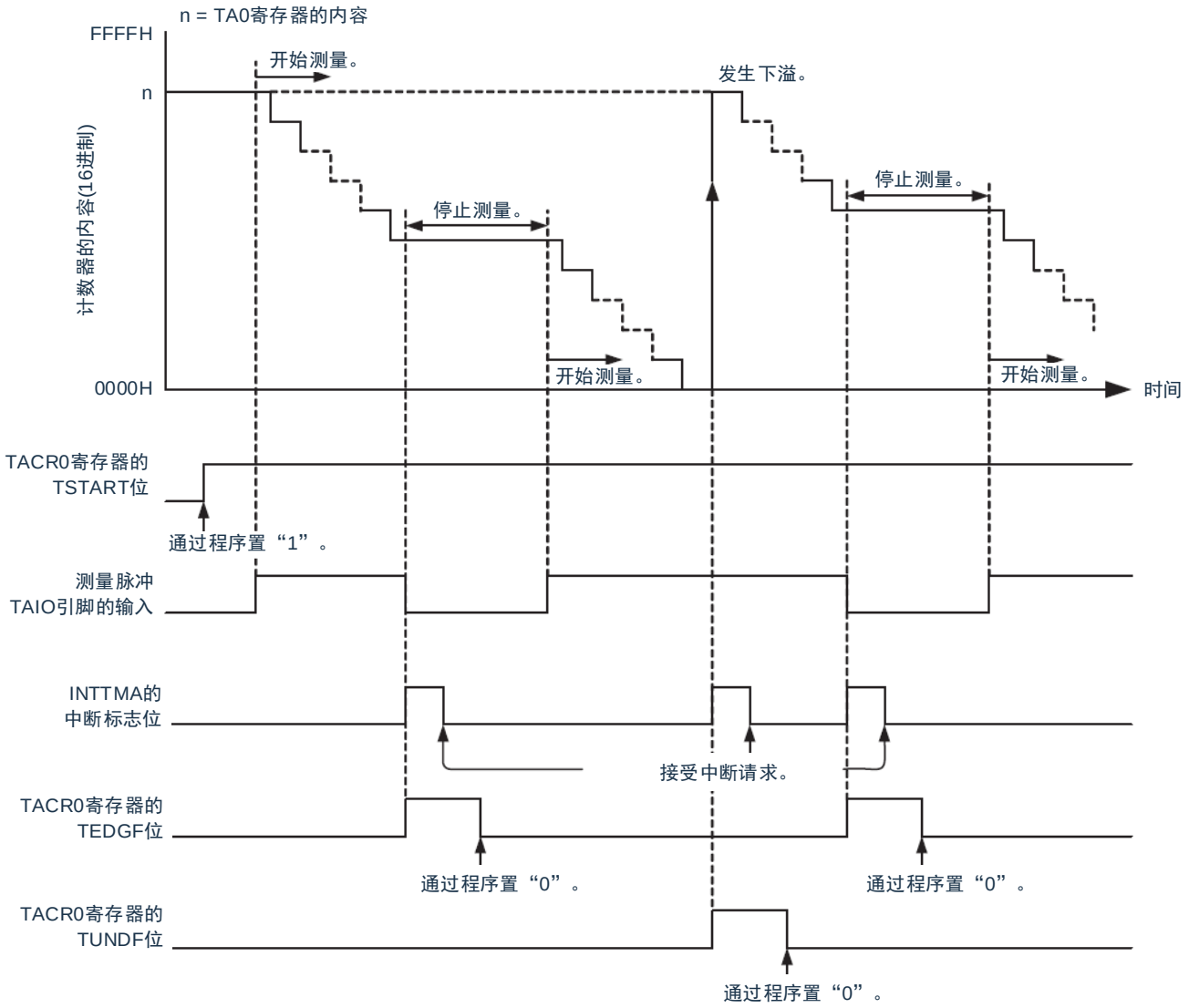
在脉宽测量模式中，如果给 TAIO 引脚输入 TAIOC0 寄存器的 TEDGSEL 位指定的电平，就通过选择的计数源开始递减计数。如果 TAIO 引脚输入的指定电平结束，计数器就停止计数，TACR0 寄存器的 TEDGF 位变为“1”（有有效边沿）并且产生中断请求。通过在计数器停止计数时读计数值进行脉宽数据的测量。如果在测量过程中计数器发生下溢，TACR0 寄存器的 TUNDF 位就变为“1”（发生下溢）并且产生中断请求。

脉宽测量模式的运行例子如图 6-14 所示。

要存取 TACR0 寄存器的 TEDGF 位和 TUNDF 位时，请参照“6.5.2 标志的存取（TACR0 寄存器的 TEDGF 位和 TUNDF 位）”。

图 6-14：脉冲测量模式的运行例子

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



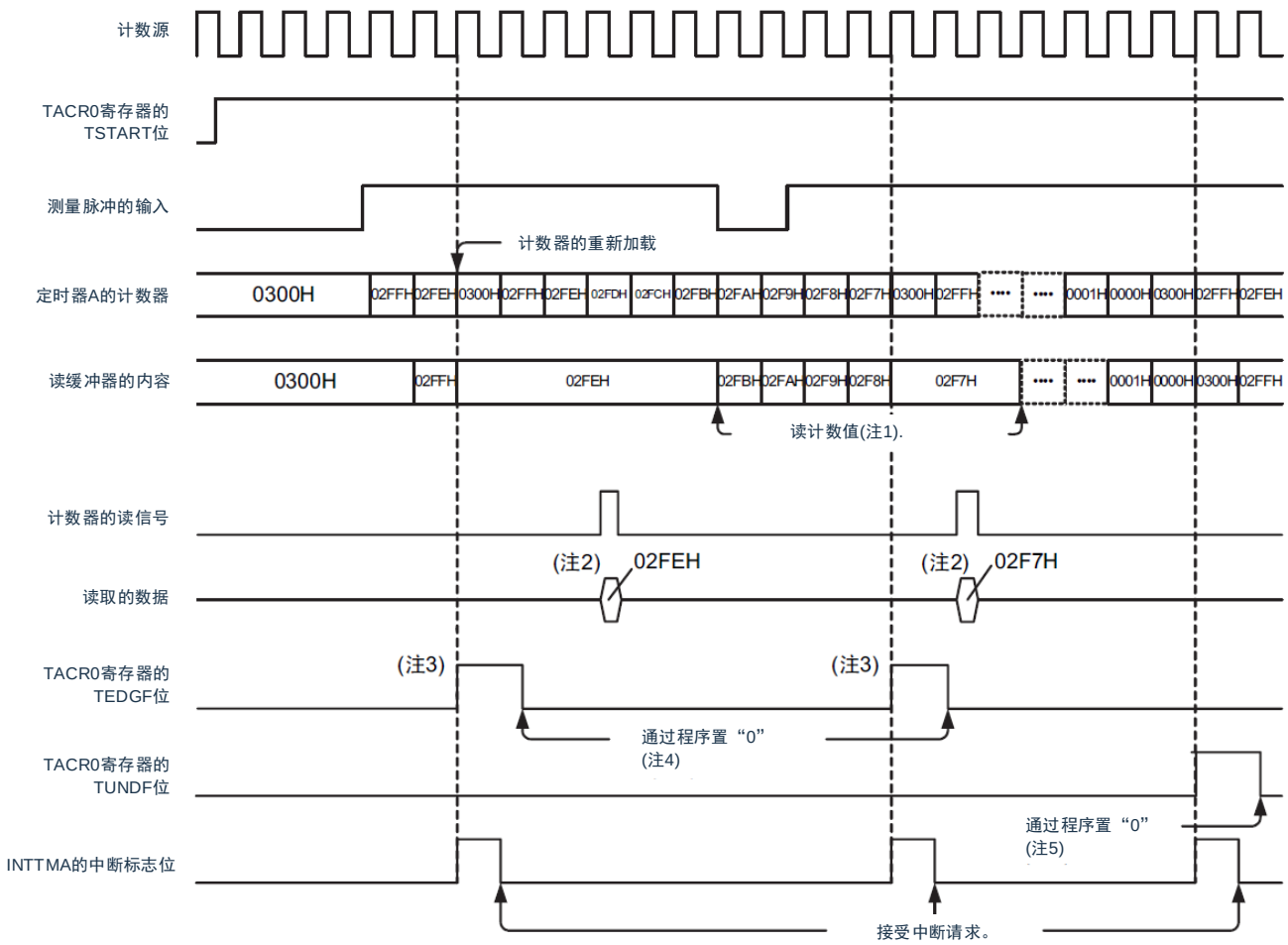
6.4.6 脉冲周期测量模式

这是测量 TAIO 引脚输入的外部信号脉冲周期的模式。

通过 TAMR0 寄存器的 TCK0~TCK2 位选择的计数源，计数器进行递减计数。如果给 TAIO 引脚输入 TAIOC0 寄存器的 TEDGSEL 位指定期间的脉冲，就在计数源的上升沿将计数值传送到读缓冲器，并且在下一个上升沿将重加载寄存器的值加载到计数器，同时 TACR0 寄存器的 TEDGF 位变为“1”（有有效边沿）并且产生中断请求。此时，读 TA0 寄存器（读缓冲器），读取值和重加载值的差为输入脉冲的周期数据。周期数据被保持到读缓冲器为止。如果计数器发生下溢，TACR0 寄存器的 TUNDF 位就变为“1”（发生下溢）并且产生中断请求。脉冲周期测量模式的运行例子如图 6-15 所示。

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

图 6-15：脉冲周期测量模式的运行例子



这是TA0寄存器的初始值为“0300H”并且将TAIOC0寄存器的TEDGSEL位置“0”以及在测量脉冲上升前进行测量的情况。

注1：必须在从TEDGF位变为“1”(有有效边沿)到输入下一个有效边沿为止的期间读TA0寄存器。读缓冲器的内容被保持到读TA0寄存器为止，因此，如果在输入有效边沿前不读TA0寄存器，就保持以前周期的测量结果。
注2：如果在脉冲周期测量模式中读TA0寄存器，读取值就为读缓冲器的内容。
注3：如果在输入测量脉冲的有效边沿后输入外部脉冲的设定边沿，TACR0寄存器的TEDGF位就变为“1”(有有效边沿)。
注4：要通过程序置“0”时，必须通过8位操作指令给TACR0寄存器的TEDGF位写“0”.
注5：要通过程序置“0”时，必须通过8位操作指令给TACR0寄存器的TUNDF位写“0”.

6.4.7 与EVENTC的协作

能通过与 EVENTC 的协作，将 EVENTC 输入的事件设置为计数源。

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

EVENTC 设置步骤如下所示。

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

6.4.8 各模式的输出设置

各模式中的 TAO 引脚和 TAIO 引脚的状态如表 6-6 和表 6-7 所示。

表 6-6：TAO 引脚的设置

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

表 6-7：TAIO 引脚的设置

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

注：这是和 TAIO 功能复用端口对应的端口模式寄存器（PMxx）的位。

6.5 使用定时器A时的注意事项

6.5.1 计数的开始和停止控制

- 事件计数模式或者将计数源设置为非EVENTC的情况
  如果在计数停止过程中给TACR0寄存器的TSTART位写“1”（开始计数），就在3个计数源周期内TACR0寄存器的TCSTF位为“0”（停止计数）。除了TCSTF位以外，不能在TCSTF位变为“1”（正在计数）前存取定时器A的相关寄存器[注]。
  如果在计数过程中给TSTART位写“0”（停止计数），就在3个计数源周期内TCSTF位为“1”。在TCSTF位变为“0”时停止计数。除了TCSTF位以外，不能在TCSTF位变为“0”前存取定时器A的相关寄存器注。必须在将TSTART位从“0”改为“1”前清除中断寄存器。详细内容请参照“第23章 中断功能”。

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

- 事件计数模式或者将计数源设置为EVENTC的情况
  如果在计数停止过程中给TACR0寄存器的TSTART位写“1”（开始计数），就在2个CPU时钟周期内TACR0寄存器的TCSTF位为“0”（停止计数）。除了TCSTF位以外，不能在TCSTF位变为“1”（正在计数）前存取定时器A的相关寄存器[注]。
  如果在计数过程中给TSTART位写“0”（停止计数），就在2个CPU时钟周期内TCSTF位为“1”。在TCSTF位变为“0”时停止计数。除了TCSTF位以外，不能在TCSTF位变为“0”前存取定时器A的相关寄存器[注]。
  必须在将 TSTART 位从“0”改为“1”前清除中断寄存器。详细内容请参照“第 23 章中断功能”。

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

6.5.2 标志的存取（TACR0寄存器的TEDGF位和TUNDF位）

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

6.5.3 计数寄存器的存取

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

6.5.4 模式的变更

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

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

6.5.5 TAO引脚和TAIO引脚的设置步骤

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

更改步骤

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

要从 TAIO 引脚输入时，必须按照以下步骤进行设置。

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

6.5.6 不使用定时器A的情况

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

6.5.7 定时器A运行时钟的停止

能通过 PER0 寄存器的 TMA 位控制定时器 A时钟的提供或者停止。但是，不能在定时器 A 的时钟停止时存取以下的 SFR，而必须在提供定时器 A 时钟的状态下进行存取。

TA0 寄存器、TACR0 寄存器、TAMR0 寄存器、TAIOC0 寄存器和 TAISR0 寄存器

6.5.8 深度睡眠模式（事件计数器模式）的设置步骤

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

设置步骤

(1) 设置运行模式。

(2) 开始计数（TSTART=1、TCSTF=1）。

(3) 停止提供定时器 A 的时钟。

要在深度睡眠模式中停止事件计数器模式时，必须按照以下步骤进行运行停止处理。

(1) 提供定时器 A 的时钟。

(2) 停止计数（TSTART=0、TCSTF=0）

6.5.9 深度睡眠模式中（只限于事件计数器模式）的功能限制

要在深度睡眠模式中使事件计数器模式运行时，不能使用数字滤波器功能。

6.5.10 通过TSTOP位进行强制的计数停止

不能在通过 TACR0 寄存器的 TSTOP 位强制停止计数器的计数后的 1 个计数源周期内存取以下的 SFR。TA0 寄存器、TACR0 寄存器和 TAMR0 寄存器

6.5.11 数字滤波器

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

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

6.5.12 选择F_{IL}作为计数源的情况

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

第7章 实时时钟

7.1 实时时钟的功能

实时时钟有以下功能。

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

7.2 实时时钟的结构

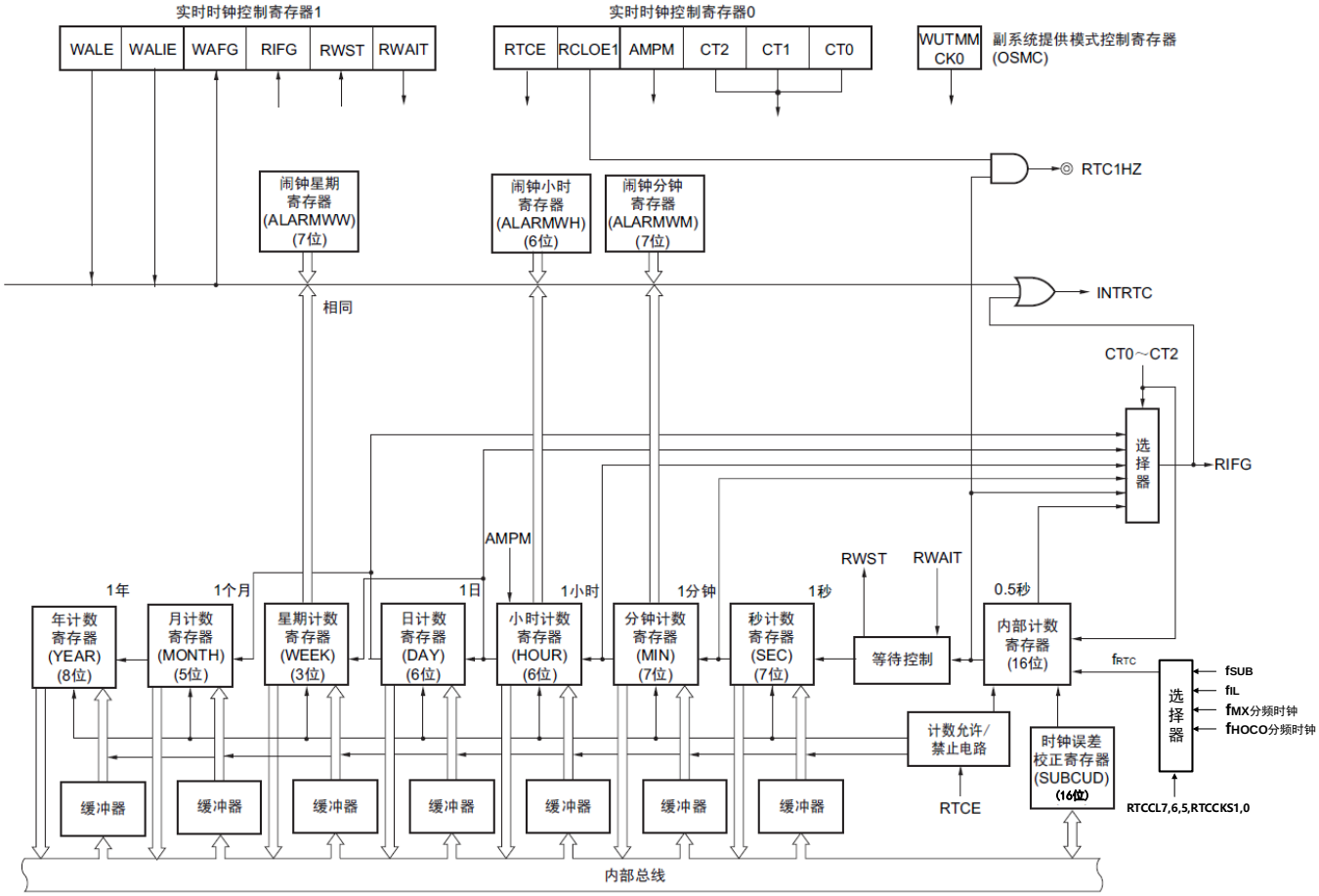
实时时钟由以下硬件构成。

表 7-1：实时时钟的结构

项目	结构
计数器	内部计数器（16位）
控制寄存器	外围允许寄存器0（PER0.bit7）
	实时时钟选择寄存器（RTCCL）
	实时时钟控制寄存器0（RTCC0）
	实时时钟控制寄存器1（RTCC1）
	秒计数寄存器（SEC）
	分钟计数寄存器（MIN）
	小时计数寄存器（HOUR）
	日计数寄存器（DAY）
	星期计数寄存器（WEEK）
	月计数寄存器（MONTH）
	年计数寄存器（YEAR）
	时钟误差校正寄存器（SUBCUD）
	闹钟分钟寄存器（ALARMWM）
	闹钟小时寄存器（ALARMWH）
	闹钟星期寄存器（ALARMWW）

注：以上 RTC 控制寄存器的复位只受 POR 复位控制。

图 7-1：实时时钟的框图



注意：只有在选择 F_{MX}/F_{HOCO} 的分周时钟（分周后≈32,768KHZ）或者副系统时钟（F_{SUB}=32.768KHz）作为实时时钟的运行时钟的情况下，才能进行年、月、星期、日、小时、分钟和秒的计数。当选择低速内部振荡器时钟（F_{IL}=32.768KHz）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)× F_{SUB}/F_{IL}

7.3 控制实时时钟的寄存器

通过以下寄存器控制实时时钟。

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

7.3.1 外围允许寄存器0（PER0）

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

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

图 7-2：外围允许寄存器 0（PER0）的格式

地址：40020420H 复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN注	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

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

注意：

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

7.3.2 实时时钟选择寄存器（RTCCL）

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

图 7-3：实时时钟选择寄存器（RTCCL）的格式

地址：0x40040C0C　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCCKS1	RTCCKS0

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

RTCCKS1	RTCCKS0	RTCCL6	RTCCL5	实时时钟、15 位间隔定时器的计数时钟的运行时钟的选择
0	0	x	x	副系统时钟（F_{SUB}）
0	1			低速内部振荡器时钟（F_{IL}）（必须设置 WUTMMCK0 为 1）
1	0	1	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/976
1	1	0	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/488
1	1	1	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/244

7.3.3 实时时钟控制寄存器0（RTCC0）

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

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

图 7-4：实时时钟控制寄存器 0（RTCC0）的格式

地址：0x4004240D　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1 注	0	AMPM	CT2	CT1	CT0

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

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

AMPM	12小时系统/24小时系统的选择
0	12小时系统（表示上午或者下午）
1	24小时系统
• 要更改AMPM位的值时，必须在将RWAIT位（实时时钟控制寄存器1（RTCC1）的bit0）置“1”后进行改写。如果更改AMPM位的值，小时计数寄存器（HOUR）的值就变为所设时间系统的对应值。 • 时间位的表示如表8-2所示。	

CT2	CT1	CT0	固定周期中断（INTRTC）的选择
0	0	0	不使用固定周期中断功能。
0	0	1	0.5秒一次（与秒累加同步）
0	1	0	1秒一次（与秒累加同时）
0	1	1	1分钟一次（每分钟的00秒）
1	0	0	1小时一次（每小时的00分00秒）
1	0	1	1日一次（每日的00点00分00秒）
1	1	×	1个月一次（每月的1日上午00点00分00秒）
要在计数器运行中（RTCE=1）更改CT2~CT0位的值时，必须在通过中断屏蔽标志寄存器将INTRTC设定为禁止中断处理后进行改写，并且必须在改写后清除RIFG标志和RTCIF标志，然后再设定为允许中断处理。			

注意：

1. 在 RTCE 位为“1”时，不能更改 RCLOE1 位。
2. 在 RTCE 位为“0”时，即使将 RCLOE1 位置“1”也不输出 1Hz。

备注：×：忽略

7.3.4 实时时钟控制寄存器1（RTCC1）

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

图 7-5：实时时钟控制寄存器 1（RTCC1）的格式(1/2)

地址：0x4004240E　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

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

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

WAFG	闹钟检测状态标志
0	闹钟不一致。
1	检测到闹钟一致。
这是表示检测到闹钟一致的状态标志。只在WALE位为“1”时有效，在检测到闹钟一致并且经过1个F_{RTC}时钟后变为“1”。 通过给此标志写“0”来清除此标志。写“1”的操作无效。	

图 7-5：实时时钟控制寄存器 1（RTCC1）的格式(2/2)

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

RWST	实时时钟的等待状态标志
0	计数器正在运行。
1	正处于计数器的读写模式。
这是表示RWAIT位的设定是否有效的状态。 必须在确认此标志为“1”后读写计数值。	

RWAIT	实时时钟的等待控制
0	设定为计数器运行。
1	设定为SEC~YEAR计数器停止运行，进入计数器的读写模式。
此位控制计数器的运行。要读写计数值时，必须给此位写“1”。 因为内部计数器（16位）继续运行，所以必须在1秒内结束读写，然后返回到“0”。 从将RWAIT位置“1”到能读写计数值（RWST=1）为止，最多需要1个F_{RTC}时钟的时间。 如果在RWAIT位为“1”时发生内部计数器（16位）上溢，就保持发生上溢的状态，在RWAIT位变为“0”后进行递增计数。 但是，当写秒计数寄存器时，不保持发生上溢的状态。	

备注：

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

7.3.5 时钟误差校正寄存器（SUBCUD）

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

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

图 7-6：时钟误差校正寄存器（SUBCUD）的格式

地址：0x40042414　　复位后：0000H　　R/W

符号	15	14	13	12	11	10	9	8
SUBCUD	DEV	0	0	F12	F11	F10	F9	F8

7	6	5	4	3	2	1	0
F7	F6	F5	F4	F3	F2	F1	F0

DEV	时钟误差的校正时序的设定
0	在秒位为“00”、“20”、“40”时（每20秒）进行时钟误差的校正。
1	只在秒位为“00”时（每60秒）进行时钟误差的校正。
在以下所示期间禁止写SUBCUD寄存器： • DVE=0：SEC=00H、20H、40H的期间 • DVE=1：SEC=00H的期间	

F12	时钟误差校正值的设定
0	{(F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)-1}×2增加
1	{(/F11,/F10,/F9,/F8,/F7,/F6,/F5,/F4,/F3,/F2,/F1,/F0)+1}×2减少
当(F12,F11,F10,F9,F8,F7,F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,0,0,0,0,0,0,0),(0,0,0,0,0,0,0,0,0,0,0,0,1),(1,0,0,0,0,0,0,0,0,0,0,0,0)或者(1,0,0,0,0,0,0,0,0,0,0,0,1)时,不进行时钟误差的校正。 校正值的范围：（F12=0）2、4、6、8、……、8186、8188 （F12=1）-2、-4、-6、-8、……、-8186、-8188	

注意：”/”表示各位取反后的值。

能通过时钟误差校正寄存器（SUBCUD）进行校正的范围如下所示。

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

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

7.3.6 秒计数寄存器（SEC）

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

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

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

图 7-7：秒计数寄存器（SEC）的格式

地址：0x40042402　　复位后：00H R/W

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

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

备注：如果写秒计数寄存器（SEC），就清除内部计数器（16 位）。

7.3.7 分钟计数寄存器（MIN）

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

在写时，数据先被写到缓冲器，在经过最多 2 个 F_{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定为写入值。必须以 BCD 码设定十进制的 00~59。

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

图 7-8：分钟计数寄存器（MIN）的格式

地址：0x40042403　　复位后：00H R/W

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

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

7.3.8 小时计数寄存器（HOUR）

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

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

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

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

但是，如果在复位后将 AMPM 位置“1”，此寄存器的值就变为“00H”。

图 7-9：小时计数寄存器（HOUR）的格式

地址：0x40042404　　复位后：12H R/W

符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意：

1. 当选择 AMPM 位为“0”（12 小时系统）时，HOUR 寄存器的 bit5（HOUR20）表示 AM（0）/PM（1）
2. 要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

AMPM 位的设定值、小时计数寄存器（HOUR）的值和时间的关系如表 7-2 所示。

表 7-2：时间位的表示

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

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

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

7.3.9 日计数寄存器（DAY）

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

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

在写时，数据先被写到缓冲器，在经过最多 2 个 F_{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定为写入值。必须以 BCD 码设定十进制的 01~31。

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

图 7-10：日计数寄存器（DAY）的格式

地址：0x40042406　　复位后：01H R/W

符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

7.3.10 星期计数寄存器（WEEK）

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

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

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

图 7-11：星期计数寄存器（WEEK）的格式

地址：0x40042405　　复位后：00H R/W

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

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

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

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

7.3.11 月计数寄存器（MONTH）

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

在写时，数据先被写到缓冲器，在经过最多 2 个 F_{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定为写入值。必须以 BCD 码设定十进制的 01~12。

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

图 7-12：月计数寄存器（MONTH）的格式

地址：0x40042407　　复位后：01H R/W

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

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

7.3.12 年计数寄存器（YEAR）

这是用 0~99（十进制）表示年计数值的 8 位寄存器。通过月计数器（MONTH）的上溢进行递增计数。00、04、08、……、92、96 是闰年。

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

图 7-13：年计数寄存器（YEAR）的格式

地址：0x40042408　　复位后：00H R/W

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

注意：要在计数器运行中（RTCE=1）读写此寄存器时，必须按照“7.3.18 实时时钟计数器的读写”记载的步骤进行。

7.3.13 闹钟分钟寄存器（ALARMWM）

这是设定闹钟分钟的寄存器。

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

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

图 7-14：闹钟分钟寄存器（ALARMWM）的格式

地址：0x4004240A　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

7.3.14 闹钟小时寄存器（ALARMWH）

这是设定闹钟小时的寄存器。

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

但是，如果在复位后将 AMPM 位置“1”，此寄存器的值就变为“00H”。

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

图 7-15：闹钟小时寄存器（ALARMWH）的格式

地址：0x4004240B　　复位后：12H R/W

符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

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

7.3.15 闹钟星期寄存器（ALARMWW）

这是设定闹钟星期的寄存器。

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

图 7-16：闹钟星期寄存器（ALARMWW）的格式

地址：0x4004240C　　复位后：00H R/W

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

闹钟时间的设定例子如下所示。

闹钟设定时间	星期							12小时表示				24小时表示			
	星期日 WW0	星期一 WW1	星期二 WW2	星期三 WW3	星期四 WW4	星期五 WW5	星期六 WW6	10时	1时	10分	1分	10时	1时	10分	1分
每天　上午0时00分	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天　上午1时30分	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天　上午11时59分	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一~星期五 下午0时00分	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日　下午1时30分	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 下午11时59分	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.3.16 实时时钟的运行开始

图 7-17：实时时钟的运行开始步骤

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

注 2：这只是需要校正时钟误差的情况。校正值的计算方法，请参照“7.3.21 实时时钟的时钟误差校正例子”。

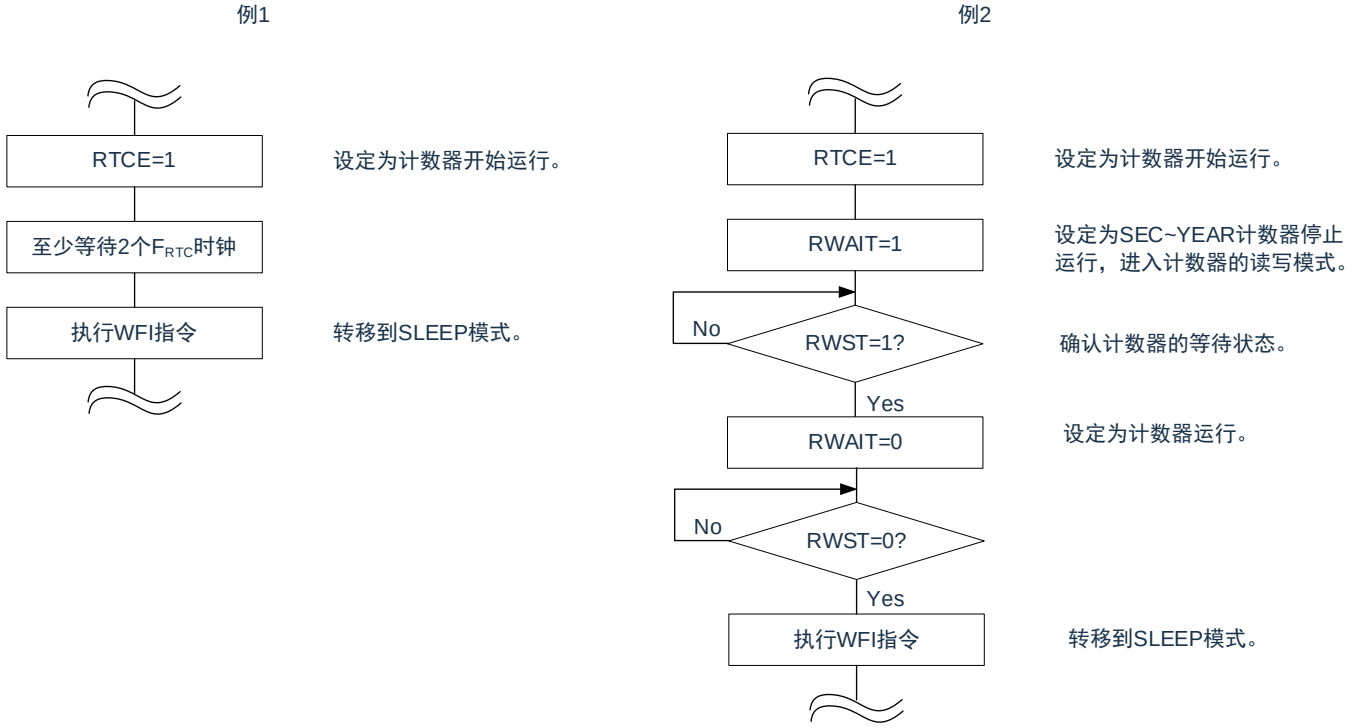
注 3：在 RTCE 位为“1”后不等待 INTRTC 位变为“1”而转移到睡眠模式的情况下，确认“7.3.17 开始运行后睡眠模式的转移”的步骤。

7.3.17 开始运行后睡眠模式的转移

要在将 RTCE 位置“1”后立即转移到睡眠(包括深度睡眠)模式时，必须进行以下某种处理。但是，在将 RTCE 位置“1”后，如果要在发生 INTRTC 中断后转移到睡眠模式，就不需要这些处理。

- 在将 RTCE 位置“1”后至少经过 2 个计数时钟（F_{RTC}）的时间之后转移到睡眠模式（参照图 7-18 的例 1）。
- 在将 RTCE 位置“1”后将 RWAIT 位置“1”，通过轮询确认 RWST 位变为“1”。然后，将 RWAIT 位置“0”并且再次通过轮询确认 RWST 位变为“0”，然后转移到睡眠模式（参照图 7-18 的例 2）。

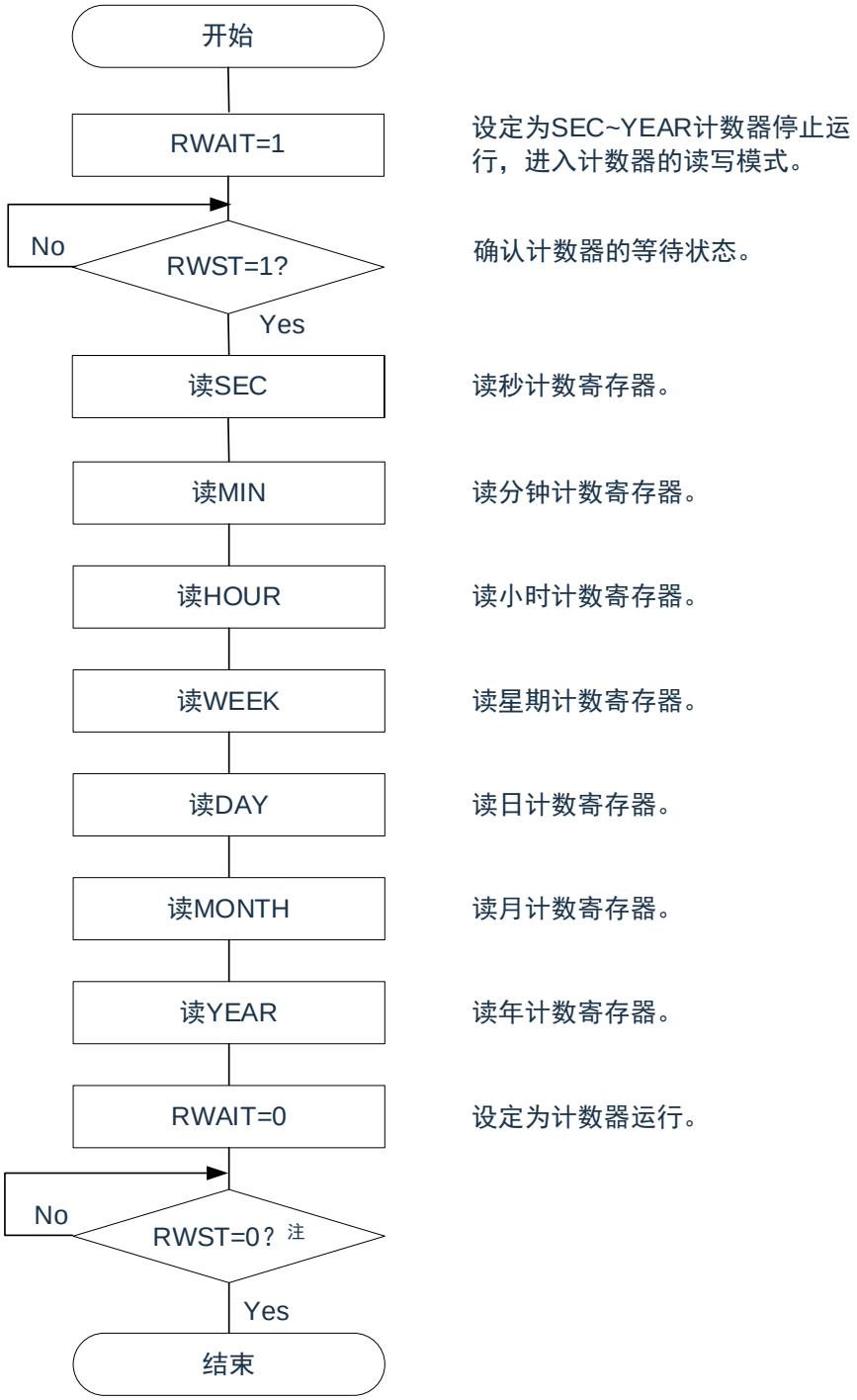
图 7-18：将 RTCE 位置“1”后的睡眠/深度睡眠模式的转移步骤



7.3.18 实时时钟计数器的读写

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

图 7-19：实时时钟计数器的读操作步骤

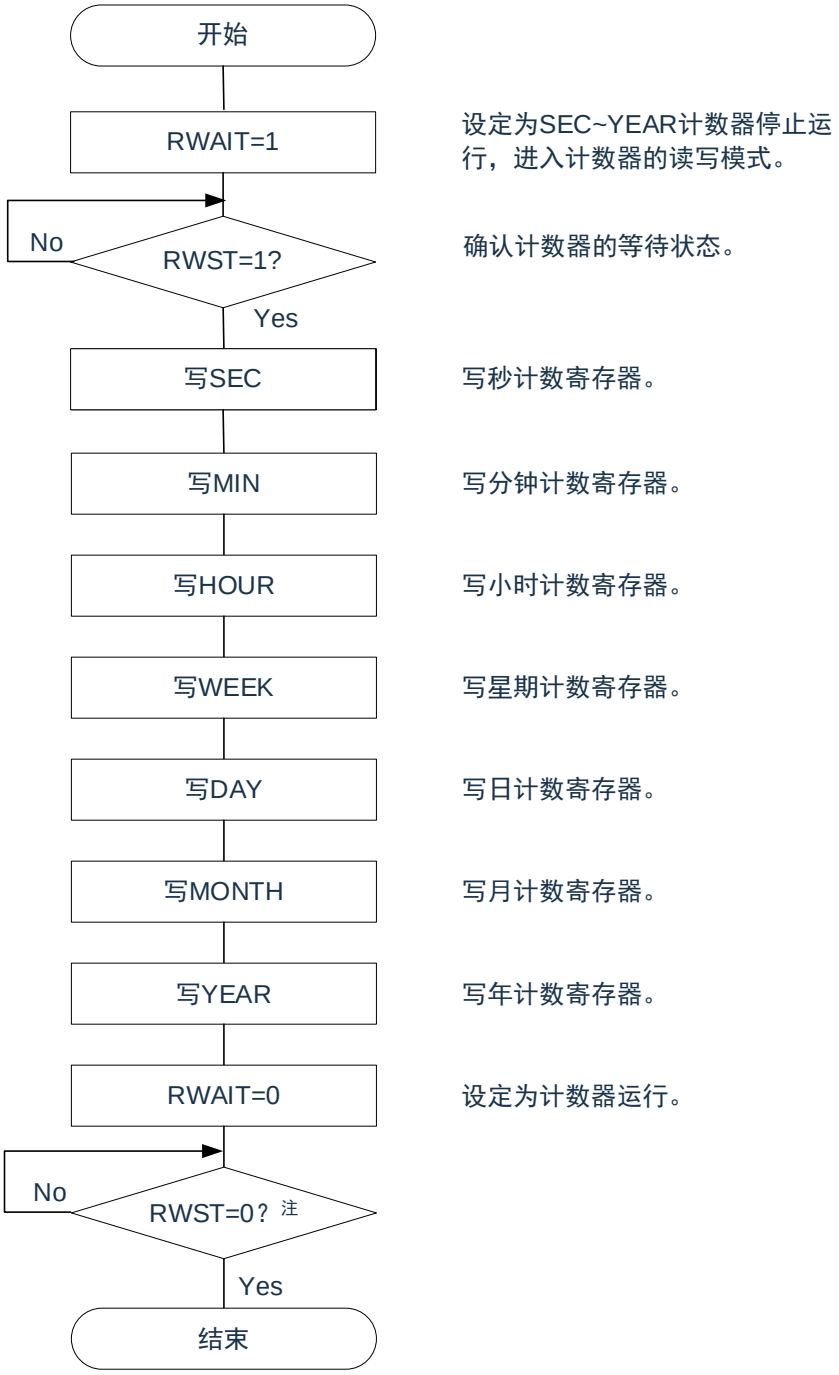


注：必须在转移到睡眠模式前，确认 RWST 位为“0”。

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

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

图 7-20：实时时钟计数器的读操作步骤



注：必须在转移到 SLEEP 模式前，确认 RWST 位为“0”。

注意：

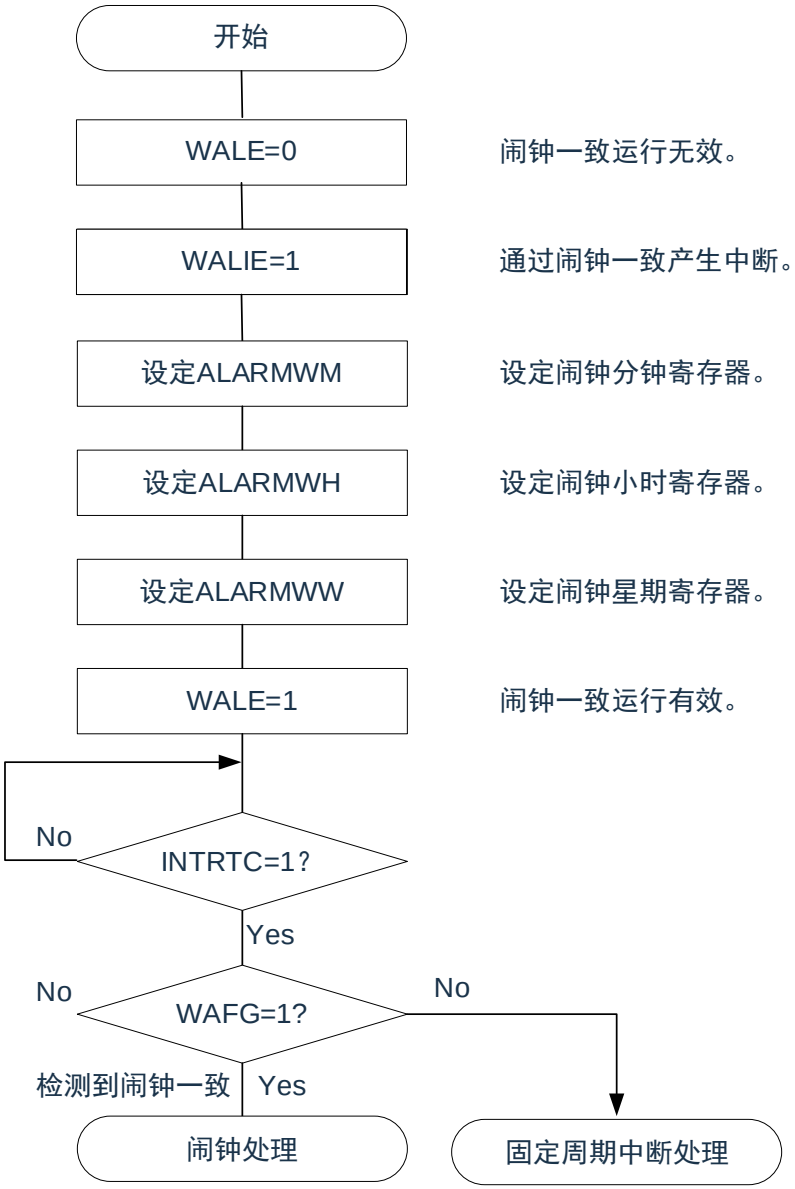
1. 必须在 1 秒内进行将 RWAIT 位置“1”到 RWAIT 位置“0”的处理。
2. 要在计数器运行中（RTCE=1）改写 SEC、MIN、HOUR、WEEK、DAY、MONTH、YEAR 寄存器时，必须在通过中断屏蔽标志寄存器将 INTRTC 设定为禁止中断处理后进行改写，并且必须在改写后清除 WAFG 标志、RIFG 标志和 RTCIF 标志。

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

7.3.19 实时时钟的闹钟设定

必须先将 WALE 位置“0”（闹钟运行无效），然后设定闹钟时间。

图 7-21：闹钟设定步骤

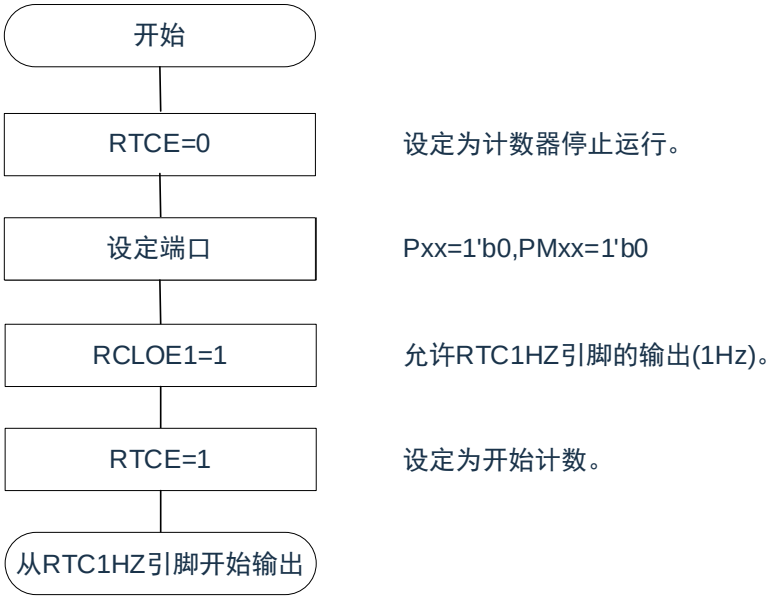


备注：

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

7.3.20 实时时钟的1Hz输出

图 7-22：1Hz 输出的设定步骤



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

7.3.21 实时时钟的时钟误差校正例子

能通过给时钟误差校正寄存器设定值进行高精度的时钟快慢校正。

校正值的计算方法的例子

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

（DEV=0 的情况）

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

（DEV=1 的情况）

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

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

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

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

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

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

备注：

1. 校正值为 2、4、6、8、……、8186、8188 或者-2、-4、-6、-8、……、-8186、-8188。
2. 振荡频率是计数时钟（F_{RTC}）的值，能用以下计算式进行计算：
   时钟误差校正寄存器为初始值（“00H”）时的 RTC1HZ 引脚的输出频率 0×32768
3. 目标频率是使用时钟误差校正寄存器进行校正后的频率。

校正例子

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

【振荡频率的测量】

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

注：有关 RTC1Hz 输出的设定步骤，请参照“7.3.20 实时时钟的 1Hz 输出”。

【校正值的计算】

（RTC1HZ 引脚的输出频率为 0.9999817Hz 的情况）

振荡频率=32768×0.9999817≈32767.4Hz

假设目标频率为 32768Hz（32767.4Hz+18.3ppm）并且 DEV=1。

适用 DEV 位为“1”时的校正值的计算式。

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

=(32767.4÷32768-1) ×32768×60

= -36

【（F12~F0）的设定值的计算】

（校正值= -36 的情况）

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

-{(/F11~/F0)-1}×2= -36

(/F11~/F0) =17

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

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

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

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

第8章 15位间隔定时器

8.1 15位间隔定时器的功能

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

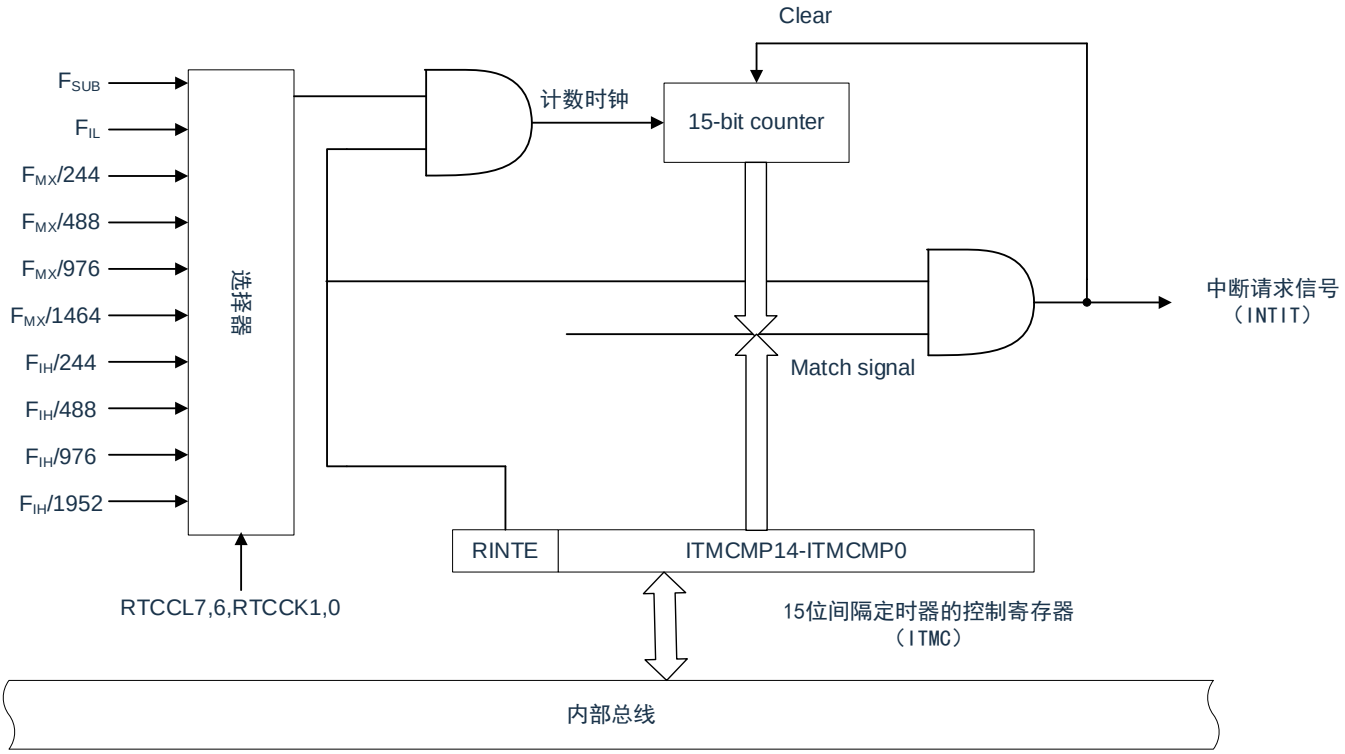
8.2 15位间隔定时器的结构

15 位间隔定时器由以下硬件构成。

表 8-1：15 位间隔定时器的结构

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

图 8-1：15 位间隔定时器的框图



8.3 控制15位间隔定时器的寄存器

通过以下寄存器控制 15 位间隔定时器。

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

8.3.1 外围允许寄存器0（PER0）

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

要使用 15 位间隔定时器时，必须将 bit7（RTCEN）置“1”。通过 8 位存储器操作指令设定 PER0 寄存器。在产生复位信号后，此寄存器的值变为“00000000H”。

图 9-2：外围允许寄存器 0（PER0）的格式

地址：40020420H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN注	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

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

8.3.2 实时时钟选择寄存器（RTCCL）

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

图 8-3：实时时钟选择寄存器（RTCCL）的格式

地址：0x40040C0C　　复位后：00H　　R/W

RTCCL

RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCCKS1	RTCCKS0

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

RTCCKS1	RTCCKS0	RTCCL6	RTCCL5	实时时钟、15 位间隔定时器的计数时钟的运行时钟的选择
0	0	x	x	副系统时钟（F_{SUB}）
0	1			低速内部振荡器时钟（F_{IL}）（必须设置 WUTMMCK0 为 1）
1	0	0	1	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/1952
1	0	0	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/1464
1	0	1	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/976
1	1	0	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/488
1	1	1	0	主时钟 F_{MAX}/F_{HOCO}(通过 RTCCL7 选择)/244

8.3.3 15位间隔定时器的控制寄存器（ITMC）

这是设定 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定 ITMC 寄存器。

在产生复位信号后，此寄存器的值变为“7FFFH”。

图 8-4：15 位间隔定时器控制寄存器（ITMC）的格式

地址：0x40042400　　复位后：7FFFH　R/W

符号	15	14~0
ITMC	RINTE	ITCMP14~ITCMP0

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

ITCMP14~ITCMP0	15位间隔定时器比较值的设定
0001H	这些位产生“计数时钟周期×（ITCMP设定值+1）”的固定周期中断。
• • •	
7FFFH	
0000H	禁止设定。
ITCMP14~ITCMP0为“0001H”或者“7FFFH”时的中断周期例子 • ITCMP14~ITCMP0=0001H，计数时钟：F_{SUB}=32.768KHz 1/32.768[KHz]×(1+1)=0.06103515625[ms]≈61.03[us] • ITCMP14~ITCMP0=7FFFH，计数时钟： F_{SUB}=32.768KHz1/32.768[KHz]×(32767+1)=1000[ms]	

注意：

1. 要将 RINTE 位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将 INTIT 设定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 标志后设定为允许中断处理。
2. RINTE 位的读取值在设定 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，如果要设定 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定 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 位的设定时，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 位的设定。

8.4 15位间隔定时器的运行

8.4.1 15位间隔定时器的运行时序

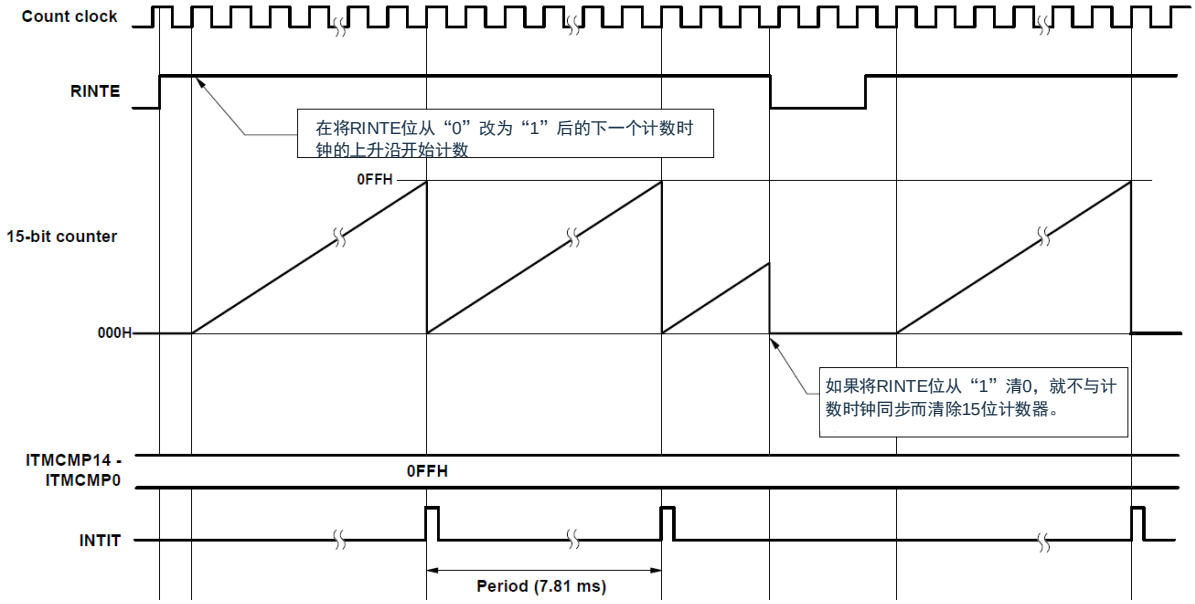
以 ITCMP14~ITCMP0 位设定的计数值为间隔，作为重复产生中断请求（INTIT）的 15 位间隔定时器运行。如果将 RINTE 位置“1”，15 位计数器就开始计数。

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

15 位间隔定时器的基本运行如图 8-5 所示。

图 8-5：15 位间隔定时器的运行时序

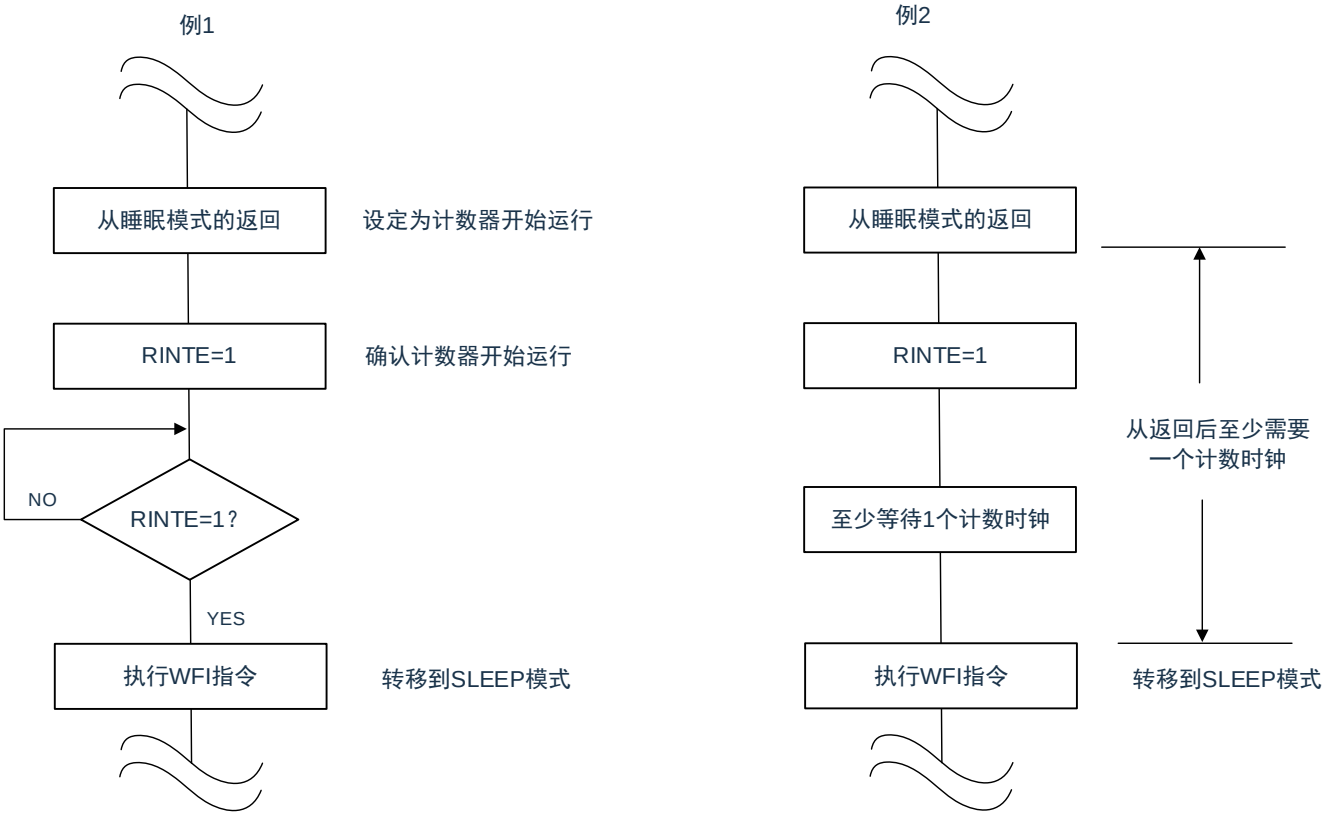
（ITCMP14~ITCMP0=0FFH，计数时钟：F_{SUB}=32.768KHz）



8.4.2 从睡眠模式返回后开始计数器的运行并且再次向睡眠模式的转移

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

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



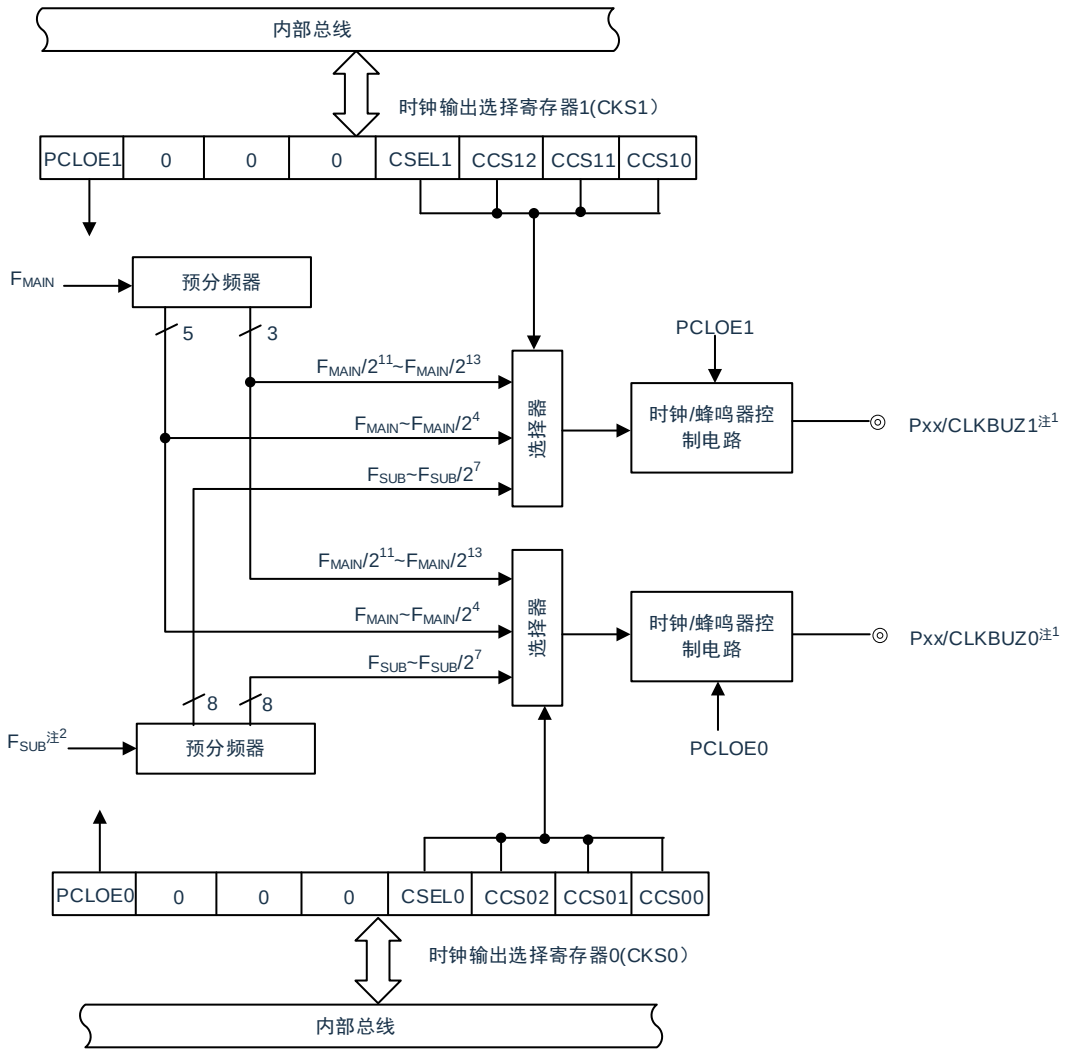
第9章 时钟输出/蜂鸣器输出控制电路

9.1 时钟输出/蜂鸣器输出控制电路的功能

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

本产品有两个时钟输出/蜂鸣器输出引脚 CLKBUZ0 和 CLKBUZ1。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。（n=0、1）

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：当将 OSMC 寄存器的 WUTMMCK0 位置“1”时，选择 F_{SUB} 作为时钟使出/蜂鸣器输出的输出时钟时，实际输出的是 F_{IL}。

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

9.2 控制时钟输出/蜂鸣器输出控制电路的寄存器

表 9-1：时钟输出/蜂鸣器输出控制电路的寄存器

项目	寄存器列表
控制寄存器	时钟输出选择寄存器n（CKSn）

9.2.1 时钟输出选择寄存器n（CKSn）

这是允许或者禁止时钟输出引脚或者蜂鸣器频率输出引脚（CLKBUZn）的输出以及设定输出时钟的寄存器。

通过 CKSn 寄存器选择 CLKBUZn 引脚输出的时钟。通过 8 位存储器操作指令设定 CKSn 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 9-2：时钟输出选择寄存器 n（CKSn）的格式

地址：0x40041001（CKS0）、0x40041000（CKS1）　　复位后：00H　R/W

符号	7	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	CLKBUZn引脚输出允许/ 禁止的指定
0	禁止输出（默认值）。
1	允许输出。

CSELn	CCSn2	CCSn1	CCSn0	CLKBUZn引脚输出时钟的选择
0	0	0	0	F_{MAIN}
0	0	0	1	$F_{MAIN}/2$
0	0	1	0	$F_{MAIN}/2^2$
0	0	1	1	$F_{MAIN}/2^3$
0	1	0	0	$F_{MAIN}/2^4$
0	1	0	1	$F_{MAIN}/2^{11}$
0	1	1	0	$F_{MAIN}/2^{12}$
0	1	1	1	$F_{MAIN}/2^{13}$
1	0	0	0	F_{SUB}
1	0	0	1	$F_{SUB}/2$
1	0	1	0	$F_{SUB}/2^2$
1	0	1	1	$F_{SUB}/2^3$
1	1	0	0	$F_{SUB}/2^4$
1	1	0	1	$F_{SUB}/2^5$
1	1	1	0	$F_{SUB}/2^6$
1	1	1	1	$F_{SUB}/2^7$

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

注意：

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

备注：n=0、1

F_{MAIN}：主系统时钟频率

F_{SUB}：副系统时钟频率

9.3 时钟输出/蜂鸣器输出控制电路的运行

能用 1 个引脚选择用作时钟输出或者蜂鸣器输出。

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

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

- 输出引脚的运行

CLKBUZn 引脚按照以下步骤进行输出：

将用作 CLKBUZn 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”。设定端口复用功能配置寄存器（PxxCFG）。

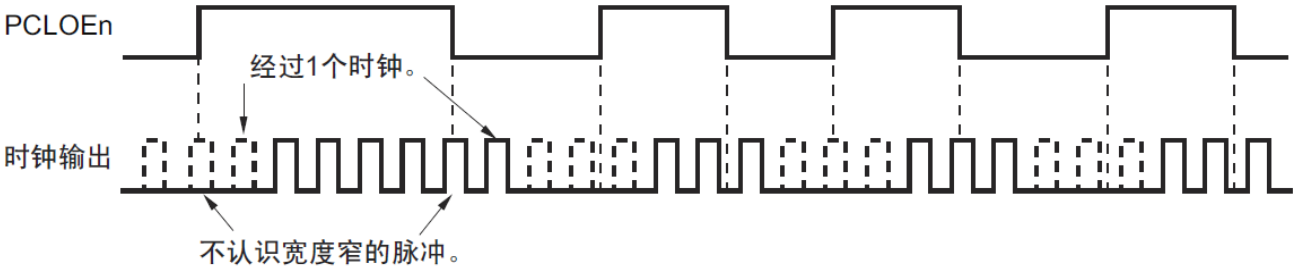
通过 CLKBUZn 引脚的时钟输出选择寄存器（CKSn）的 bit0~3（CCSn0~CCSn2、CSELn）选择输出频率（输出为禁止状态）。

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

备注：

1. CLKBUZ0 固定复用到 PA00 端口，使用 CLKBUZ0 时，不需要设置端口复用功能配置寄存器（PxxCFG）。
2. 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 9-3 所示。
3. n=0、1

图 9-3：CLKBUZn 引脚的时钟输出时序



9.4 时钟输出/蜂鸣器输出控制电路的注意事项

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

第10章 看门狗定时器

10.1 看门狗定时器的功能

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

下述情况判断为程序失控。

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

当因看门狗定时器而发生复位时，将复位控制标志寄存器（RESF）的 bit4（WDTRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 26 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。

10.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 10-1：看门狗定时器的结构

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

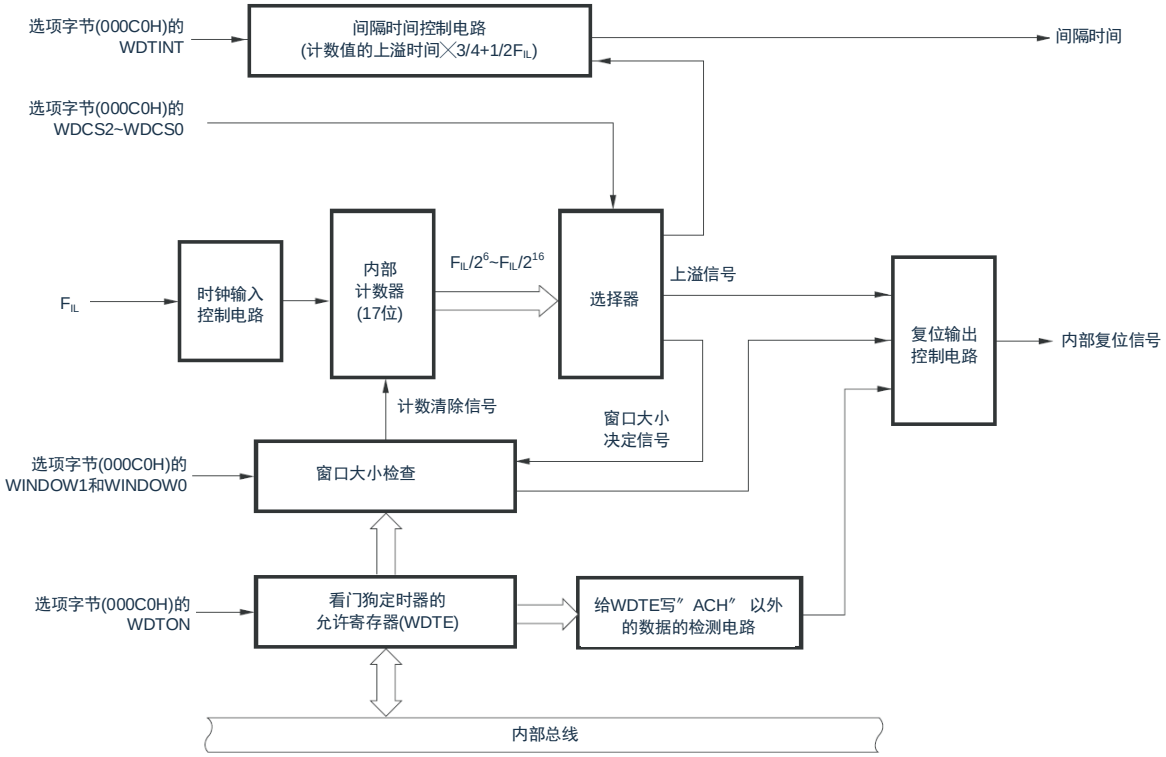
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表 10-2：选项字节和看门狗定时器的设定内容

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

备注：有关选项字节，请参照“第 31 章 选项字节”。

图 10-1：看门狗定时器的框图



备注：F_{IL}：低速内部振荡器的时钟频率

10.3 控制看门狗定时器的寄存器

通过看门狗定时器的允许寄存器（WDTE）控制看门狗定时器。

10.3.1 看门狗定时器的允许寄存器（WDTE）

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

图 10-2：看门狗定时器的允许寄存器（WDTE）的格式

地址：0x40021001　　复位后：9AH/1AH [注]　　R/W

WDTE								

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

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

注意：

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

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

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

通过 8 位存储器操作指令设置 LOCKCTL，PRCR 寄存器。

在产生复位信号后，LOCKCTL，PRCR 寄存器的值变为“00H”。

图 10-3：LOCKUP 控制寄存器（LOCKCTL）及其保护寄存器（PRCR）的格式（1/2）

地址：40020405H　　复位后：01H　R/W

LOCKCTL	0	0	0	0	0	0	0	lockup_rst

lockup_rst	LOCKUP功能的配置
0	• LOCKUP不导致WDT复位
1	• LOCKUP导致WDT复位

地址：40020406H　　复位后：00H　R/W

PRCR	PRTKEY[7:1]	PRCR

PRCR	LOCKUP控制寄存器写保护
0	• LOCKCTL寄存器不可写
1	• LOCKCTL寄存器可写

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

10.3.3 WDTCFG配置寄存器（WDTCFG0/1/2/3）

WDTCFG 配置寄存器是是否强制看门狗定时器运行的寄存器。

通过 8 位存储器操作指令设置 WDTCFG 寄存器。

在产生复位信号后，WDTCFG 寄存器的值变为“00H”。

图 10-4：WDTCFG 配置寄存器（WDTCFG0/1/2/3）

地址：40020408H　　复位后：00H　　R/W

WDTCFG0	WDTCFG0

地址：40020409H　　复位后：00H　　R/W

WDTCFG1	WDTCFG1

地址：4002040AH　　复位后：00H　　R/W

WDTCFG2	WDTCFG2

地址：4002040BH　　复位后：00H　　R/W

WDTCFG3	WDTCFG3

WDTCFG0	WDTCFG1	WDTCFG2	WDTCFG3	看门狗定时器功能的配置
0x1A	0x2B	0x3C	0x4D	•看门狗定时器强制运行停止
其他				•看门狗定时器强制运行打开

10.4 看门狗定时器的运行

10.4.1 看门狗定时器的运行控制

1) 当使用看门狗定时器时，通过选项字节（000C0H）设定以下内容：
   - 必须将选项字节（000C0H）的bit4（WDTON）置“1”，允许看门狗定时器的计数运行（在解除复位后，计数器开始运行）（详细内容请参照第31章 选项字节）。

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

   - 必须通过选项字节（000C0H）的bit3~1（WDCS2~WDCS0）设定上溢时间（详细内容请参照10.4.2和第31章 选项字节）。
   - 必须通过选项字节（000C0H）的bit6和bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照10.4.2和第31章 选项字节）。

2) 在解除复位后，看门狗定时器开始计数。
3) 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
4) 此后，解除复位后第2次以后的WDTE寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写WDTE寄存器，就产生内部复位信号。
5) 如果不给WDTE寄存器写“ACH”而超过上溢时间，就产生内部复位信号。以下情况会产生内部复位信号：
   - 当对WDTE寄存器执行位操作指令时
   - 当给WDTE寄存器写“ACH”以外的数据时

注意：

1. 只在解除复位后第 1 次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任意时候写 WDTE，就清除看门狗定时器并且重新开始计数。
2. 从给 WDTE 寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大 2 个 F_{IL} 时钟的误差。
3. 在计数值发生上溢前，都能清除看门狗定时器。
4. 如下所示，看门狗定时器在睡眠或者深度睡眠模式中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

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

5. 当 WDSTBYON 位为“0”时，在解除睡眠或者深度睡眠模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。
6. 当解除深度睡眠模式后以 X1 振荡时钟运行时，CPU 在经过振荡稳定时间后开始运行。
7. 如果从解除深度睡眠模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除深度睡眠模式后，如果要以 X1 振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设定。

10.4.2 看门狗定时器上溢时间的设定

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

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

表 10-3：看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间（F_{IL}=20KHz(MAX.)的情况）
0	0	0	$2^6/F_{IL}$（3.2ms）
0	0	1	$2^7/F_{IL}$（6.4ms）
0	1	0	$2^8/F_{IL}$（12.8ms）
0	1	1	$2^9/F_{IL}$（25.6ms）
1	0	0	$2^{11}/F_{IL}$（102.4ms）
1	0	1	$2^{13}/F_{IL}$（409.6ms）
1	1	0	$2^{14}/F_{IL}$（819.2ms）
1	1	1	$2^{16}/F_{IL}$（3276.8ms）

备注：F_{IL}：低速内部振荡器的时钟频率。

10.4.3 看门狗定时器窗口打开期间的设定

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

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

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

能设定的窗口打开期间如下所示。

表 10-4：看门狗定时器窗口打开期间的设定

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

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

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

	窗口打开期间的设定	
	75%	100%
窗口关闭时间	0~12.8ms	无
窗口打开时间	12.8~25.6ms	0~25.6ms

＜当窗口打开期间为 75%时＞

- 上溢时间：
  $2^9/F_{IL}$(MAX.)=2^9/20KHz(MAX.)=25.6ms
- 窗口关闭时间：
  0~$2^9/F_{IL}$(MIN.)×(1-0.75)=0~2^9/10KHz×0.25=0~12.8ms
- 窗口打开时间：
  $2^9/F_{IL}$(MIN.)×(1-0.75)~$2^9/F_{IL}$(MAX.)=12.8~25.6ms

10.4.4 看门狗定时器间隔中断的设定

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

表 10-5：看门狗定时器间隔中断的设定

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

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

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

10.4.4.1 LOCKUP期间看门狗定时器的运行

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

10.4.4.2 WDTCFG未配置时看门狗定时器的运行

当 WDTCFG 未配置时，看门狗定时器的计时器自动开始运行，上溢时间由选项字节里的上溢时间控制位（WDCS2~WDCS0）决定。

第11章 A/D转换器

11.1 A/D转换器的功能

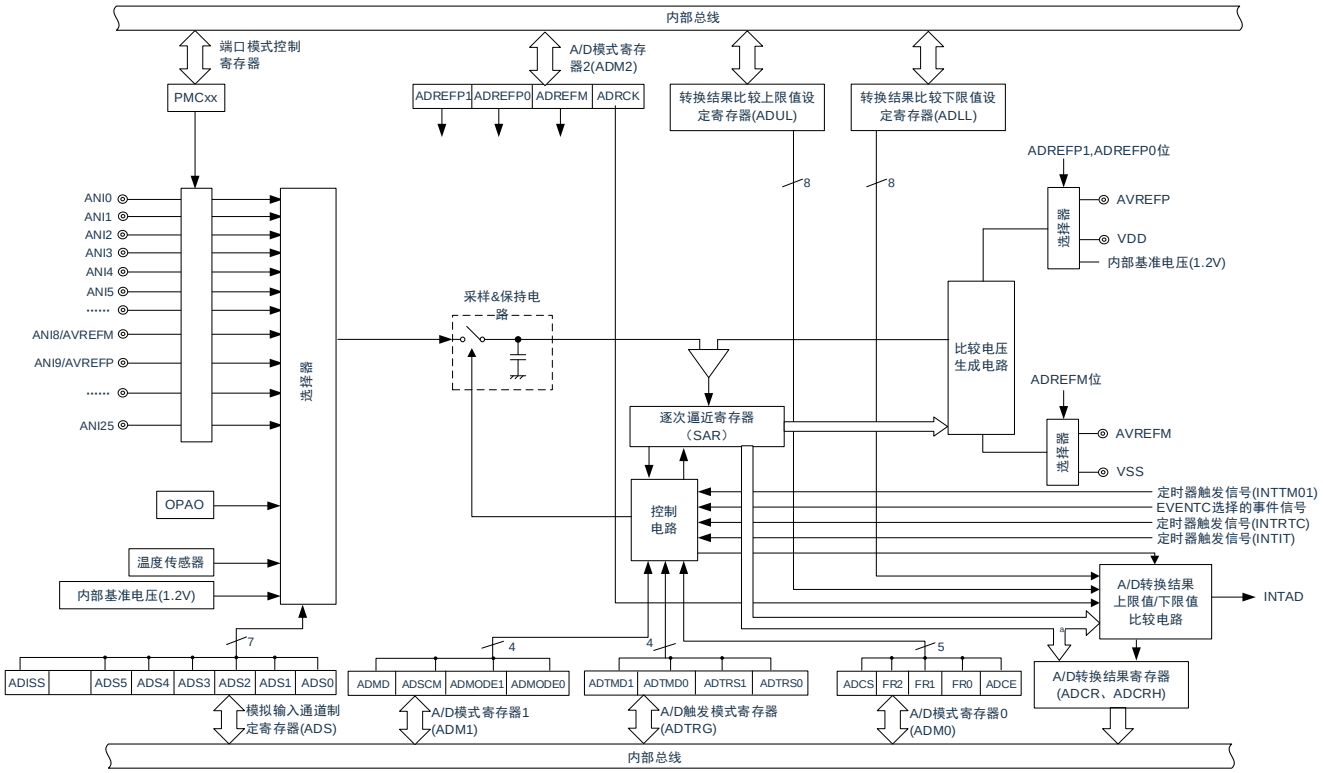
A/D 转换器是将模拟输入转换为数字值的转换器，A/D 转换器有以下的功能。

- 12位分辨率的A/D转换
- 可以选择转换外部引脚ANI0~ANI25输入的模拟电压，运算放大器OPA的输出（OPAO）、内部基准电压（1.45V）和温度传感器电压
- 每结束1次A/D转换，就产生中断请求（INTAD）

能通过下述的模式组合设定各种 A/D 转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断A/D电源的转换待机状态下，通过检测硬件触发来接通电源，在经过A/D电源稳定等待时间后自动开始转换。
通道选择模式	选择模式	选择1个通道的模拟输入，进行A/D转换。
	扫描模式	按顺序对4个通道的模拟输入进行A/D转换。能选择ANI0~ANI15中连续的4个通道作为模拟输入。
		按顺序对3个通道的模拟输入进行A/D转换。能选择ANI0~ANI15中连续的3个通道作为模拟输入。
		按顺序对2个通道的模拟输入进行A/D转换。能选择ANI0~ANI15中连续的2个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行1次A/D转换。
	连续转换模式	对所选通道进行连续的A/D转换，直到被软件停止为止。
采样时间	采样时钟5.5~255个ADCLK	采样时间可以通过ADNSMP寄存器来选择，默认使用13.5个转换时钟（F_{AD}）。

图 11-1：A/D 转换器的框图



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

11.2 控制A/D转换器的寄存器

控制 A/D 转换器的寄存器如下所示：

寄存器基地址：PGCSC_BASE=4002_0800H;ADC_BASE=4004_5000H;PORT_BASE=4004_0000H

寄存器名称	寄存器描述	R/W	复位值	寄存器地址
PER1	外围使能寄存器 1	R/W	00H	PGCSC_BASE+1AH
ADM0	A/D 转换器的模式寄存器 0	R/W	00H	ADC_BASE+00H
ADM1	A/D 转换器的模式寄存器 1	R/W	00H	ADC_BASE+02H
ADM2	A/D 转换器的模式寄存器 2	R/W	00H	ADC_BASE+04H
ADTRG	A/D 转换器的触发模式寄存器	R/W	00H	ADC_BASE+06H
ADS	模拟输入通道指定寄存器	R/W	00H	ADC_BASE+08H
ADLL	转换结果比较下限值设定寄存器	R/W	00H	ADC_BASE+0AH
ADUL	转换结果比较上限值设定寄存器	R/W	00H	ADC_BASE+0BH
ADNSMP	A/D 转换器的采样时间控制寄存器	R/W	0dH	ADC_BASE+0CH
ADCR	12 位 A/D 转换结果寄存器	R	0000H	ADC_BASE+0EH
ADCRH	8 位 A/D 转换结果寄存器	R	00H	ADC_BASE+0FH
ADTES	A/D 测试寄存器	R/W	00H	ADC_BASE+10H
ADNDIS	A/D 转换器的充放电控制寄存器	R/W	00H	ADC_BASE+11H
ADSMPWAIT	A/D 转换器的采样时间延长控制寄存器	R/W	00H	ADC_BASE+15H
PMCn	端口模式控制寄存器	R/W	注 1	PORT_BASE+注 1

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

注 1：通过 ADS 寄存器选择某通道时，需要配置该通道引脚的 PMC 寄存器，将其配置为模拟通道。

11.2.1 外围允许寄存器（PER1）

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

要使用 A/D 转换器时，必须将 bit4（ADCEN）置“1”。

通过 8 位存储器操作指令设定 PER1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-2：外围允许寄存器（PER1）

地址：0x4002081AH　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER1	XX	XX	XX	ADCEN	XX	XX	XX	XX

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

注意：要设定 A/D 转换器时，必须先在 ADCEN 位为“1”的状态下读写以下的寄存器。当 ADCEN 位为“0”时，A/D 转换器的控制寄存器的值为初始值，忽视写操作（端口模式控制寄存器（PMCxx）除外）。

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

11.2.2 A/D转换器的模式寄存器0（ADM0）

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

图 11-3：A/D 转换器的模式寄存器 0（ADM0）的格式

复位值：00H R/W

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

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

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

注 1：有关 FR2~FR0 位和 A/D 转换的详细内容，请参照“表 11-3 和表 11-4”。

注 2：A/D 转换器开始运行需要 1us 稳定时间。在软件触发模式或者硬件触发无等待模式中，在将 ADCE 位置“1”后至少经过 1us 时间，然后将 ADCS 位置“1”，则本次转换结果有效。如果等待时间小于 1us 而将 ADCS 位置“1”，就必须忽视本次转换结果。在硬件触发等待模式中，由设计保证 1us 的等待时间。

注意：

1. 必须在转换停止状态 ADCS=0 下更改 FR2~FR0 位。
2. 禁止进行 ADCS=1、ADCE=0 的设定。
3. 禁止通过 8 位操作指令将 ADCS=0、ADCE=0 的状态设定为 ADCS=1、ADCE=1。

表 11-1：ADCS 位和 ADCE 位的设定

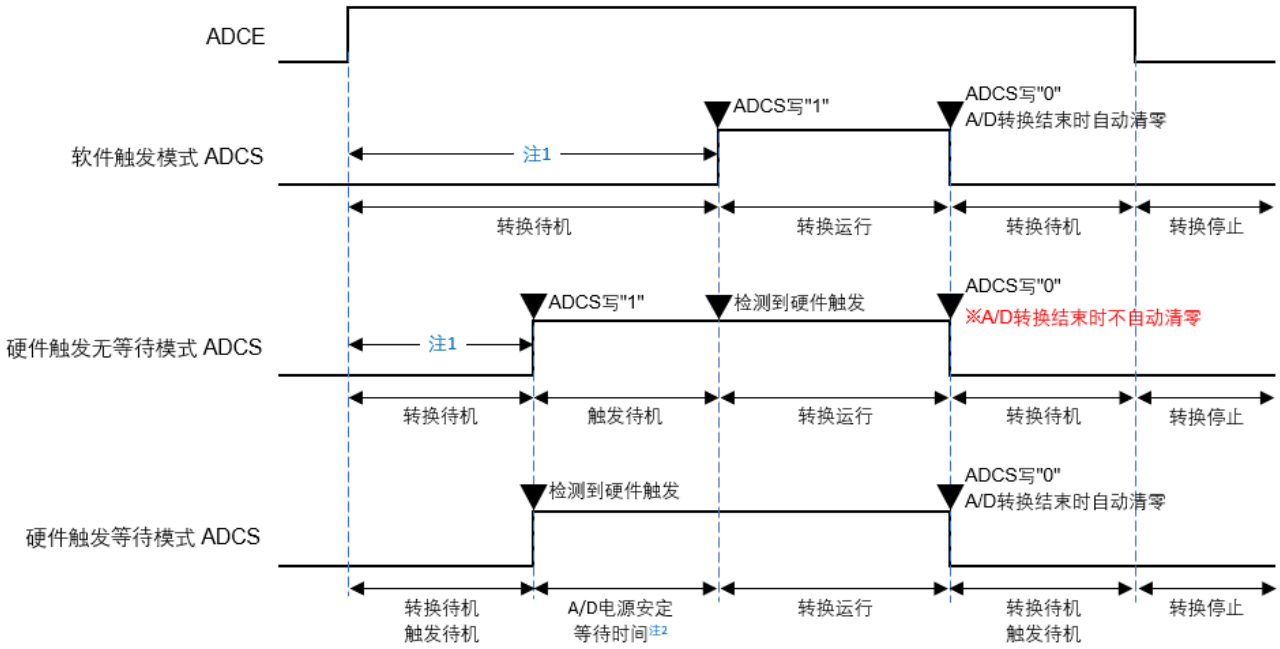
ADCS	ADCE	A/D转换运行
0	0	转换停止状态
0	1	转换待机状态
1	0	禁止设定。
1	1	转换运行状态

表 11-2：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 •当设定的n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 •当设定的n个通道转换结束时，自动清“0”

注：n=2，3，4

图 11-4：使用 A/D 各种模式时的动作状态图



注 1.在软件触发模式或者硬件触发无等待模式中，为了稳定内部电路，从 ADCE 位上升到 ADCS 位上升的时间至少需要 1us。

2.在硬件触发等待模式，A/D 电源稳定时间 1us 由设计保证。

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

硬件触发无等待模式时：2 个 F_{CLK} 时钟+A/D 转换时间

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

备注　F_{CLK}：CPU/外围硬件的时钟频率

表 11-3：A/D 转换时间的选择(1/2)

(1)无 A/D 电源稳定等待时间　　（软件触发模式/硬件触发无等待模式）

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

注 1. 要将 FR2～FR0 位、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。

注 2. 进行一次 ADC 转换需要的时间=（采样时钟数 + 逐次比较时钟数）/F_{AD}

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

备注　F_{CLK}：CPU/外围硬件的时钟频率

表 11-4：A/D 转换时间的选择(2/2)

(2)有 A/D 电源稳定等待时间（硬件触发等待模式[注1]）

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

注 1.硬件触发等待模式时，电源稳定时间由硬件设计保证，不需要设置。且在连续转换模式时，仅在第 1 次检测到硬件触发后，发生 A/D 电源稳定等待时间。

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

其中采样时钟个数可通过 ADNSMP 寄存器进行调整，默认为 13.5 个 ADCLK。逐次比较时钟个数由转换模式决定，高速变换模式时为 31.5 个 ADCLK，低电流模式时为 40.5 个 ADCLK。高速变换模式时 ADCLK 支持的最快时钟为 64MHz，低电流模式时 ADCLK 支持的最快时钟为 27MHz。

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

11.2.3 A/D转换器的模式寄存器1（ADM1）

这是设定 A/D 转换模式的寄存器。

通过 8 位存储器操作指令设定 ADM1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-5：A/D 转换器的模式寄存器 1（ADM1）的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADM1	ADMD	SMODE1	SMODE0	0	ADSCM	0	ADMODE1	ADMODE0

ADMD	A/D转换通道选择模式的设定
0	选择模式
1	扫描模式

SMODE1	SMODE0	扫描模式时，循环的通道数
0	0	4 通道扫描
0	1	3 通道扫描
1	0	2 通道扫描
其他		设定禁止

ADSCM	A/D转换模式的设定
0	连续转换模式
1	单次转换模式

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

注意：

1. 必须将 bit6~4，2 置“0”。
2. 要改写 ADM1 寄存器时，必须在转换停止状态（ADCS=0）下进行。
3. 为了正常结束 A/D 转换，必须至少将硬件触发间隔设定为以下时间：
   硬件触发无等待模式时：2 个 F_{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
4. A/D 转换时，逐次比较时钟个数由转换模式决定，高速变换模式时为 31.5 个 ADCLK，低电流模式时为 40.5 个 ADCLK。高速变换模式时 ADCLK 支持的最快时钟为 64MHz，低电流模式时 ADCLK 支持的最快时钟为 27MHz。实际使用时，请按照数据手册的“AC 特性”要求配置转换模式和转换时钟频率。

备注：F_{CLK}：CPU/外围硬件的时钟频率

11.2.4 A/D转换器的模式寄存器2（ADM2）

通过 8 位存储器操作指令设定 ADM2 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-6：A/D 转换器的模式寄存器 2（ADM2）的格式(1/3)

复位值：00H R/W

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

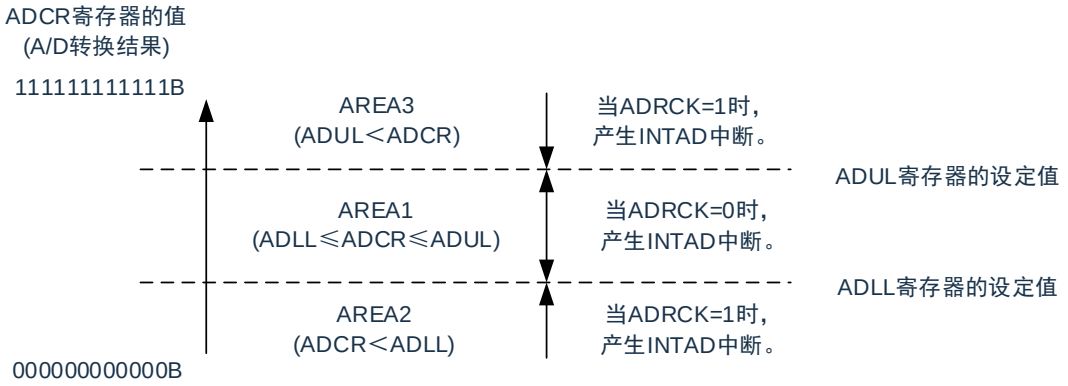
ADREFP1	ADREFP0	A/D转换器的正（+）基准电压源的选择
0	0	由V_{DD}提供。
0	1	由AV_{REFP}外部端子提供。
1	0	由内部基准电压提供（1.45V）。
1	1	设定禁止

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

ADRCK	转换结果上限值和下限值的检查
0	当ADLL寄存器≤ADCR寄存器≤ADUL寄存器（AREA1）时，产生中断信号（INTAD）。
1	当ADCR寄存器＜ADLL寄存器（AREA2）或者ADUL寄存器＜ADCR寄存器（AREA3）时，产生中断信号（INTAD）。
AREA1~AREA3的中断信号（INTAD）的产生范围如图14-8所示。	

CHRDE	A/D转换器扫描模式时通道标识的输出使能
0	扫描模式时，不在转换结果中标识通道号
1	扫描模式时，转换结果的高四位（ADCR寄存器的[15:12]）为此结果的通道号

图 11-7：ADRCK 位的中断信号产生范围



注意：

1. 要改写 ADM2 寄存器时，必须在转换停止状态（ADCS=0）下进行。
2. 当使用 AVREFP 和 AVREFM 时，必须将该端口设定为模拟端口模式（PMCxx=1）。

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

11.2.5 A/D转换器的触发模式寄存器（ADTRG）

这是设定 A/D 转换触发模式和硬件触发信号的寄存器。

通过 8 位存储器操作指令设定 ADTRG 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-8：A/D 转换器的触发模式寄存器（ADTRG）的格式

复位值：00H R/W

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

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

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

注意：

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

   硬件触发无等待模式时：2 个 F_{CLK} 时钟+A/D 转换时间

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

备注：F_{CLK}：CPU/外围硬件的时钟频率

11.2.6 模拟输入通道指定寄存器（ADS）

这是指定要进行 A/D 转换的模拟电压输入通道的寄存器。通过 8 位存储器操作指令设定 ADS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-9：模拟输入通道指定寄存器（ADS）的格式

复位值：00H R/W

	7	6	5	4	3	2	1	0
ADS	ADISS	0	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0

- 选择模式（ADM1.ADMD=0）

ADS 寄存器设定值		CH 选择	引脚名称
ADISS	ADC[5:0]		
0	6’h00	ANI0	PA00
0	6’h01	ANI1	PA01
0	6’h02	ANI2	PA02
0	6’h03	ANI3	PA03
0	6’h04	ANI4	PA04
0	6’h05	ANI5	PA05
0	6’h06	ANI6	PA06
0	6’h07	ANI7	PA07
0	6’h08	ANI8	PB00
0	6’h09	ANI9	PB01
0	6’h0a	ANI10	PB02
0	6’h0b	ANI11	PB10
0	6’h0c	ANI12	PB11
0	6’h0d	ANI13	PB12
0	6’h0e	ANI14	PB13
0	6’h0f	ANI15	PB14
0	6’h10	ANI16	PB15
0	6’h11	ANI17	PC00
0	6’h12	ANI18	PC01
0	6’h13	ANI19	PC02
0	6’h14	ANI20	PC03
0	6’h15	ANI21	PC04
0	6’h16	ANI22	PC05
0	6’h17	ANI23	PC06
0	6’h18	ANI24	PD04
0	6’h19	ANI25	PD05
0	6’h23	OPAO	-
1	6’h00	BGR(温度 sensor0)	-
1	6’h01	BGR(1.45V)	-
其他禁止设定			

备注：

1. 如果选择内部基准电压（1.45V）作为比较器 0 或者比较器 1 的基准电压，就不能选择温度传感器输出。
2. A/D 转换器的模拟输入通道因产品而不同。详细的通道分配信息请参考数据手册。

- 4 通道扫描模式（ADM1.ADMD=1）

ADISS	ADS[5:0]	模拟输入通道			
		扫描 0	扫描 1	扫描 2	扫描 3
1’b0	6’h00	ANI0	ANI1	ANI2	ANI3
1’b0	6’h01	ANI1	ANI2	ANI3	ANI4
1’b0	6’h02	ANI2	ANI3	ANI4	ANI5
1’b0	6’h03	ANI3	ANI4	ANI5	ANI6
1’b0	6’h04	ANI4	ANI5	ANI6	ANI7
1’b0	6’h05	ANI5	ANI6	ANI7	ANI8
1’b0	6’h06	ANI6	ANI7	ANI8	ANI9
1’b0	6’h07	ANI7	ANI8	ANI9	ANI10
1’b0	6’h08	ANI8	ANI9	ANI10	ANI11
1’b0	6’h09	ANI9	ANI10	ANI11	ANI12
1’b0	6’h0A	ANI10	ANI11	ANI12	ANI13
1’b0	6’h0B	ANI11	ANI12	ANI13	ANI14
1’b0	6’h0C	ANI12	ANI13	ANI14	ANI15
上述以外		设定禁止			

- 3 通道扫描模式（ADM1.ADMD=1）

ADISS	ADS[5:0]	模拟输入通道		
		扫描 0	扫描 1	扫描 2
1’b0	6’h00	ANI0	ANI1	ANI2
1’b0	6’h01	ANI1	ANI2	ANI3
1’b0	6’h02	ANI2	ANI3	ANI4
1’b0	6’h03	ANI3	ANI4	ANI5
1’b0	6’h04	ANI4	ANI5	ANI6
1’b0	6’h05	ANI5	ANI6	ANI7
1’b0	6’h06	ANI6	ANI7	ANI8
1’b0	6’h07	ANI7	ANI8	ANI9
1’b0	6’h08	ANI8	ANI9	ANI10
1’b0	6’h09	ANI9	ANI10	ANI11
1’b0	6’h0A	ANI10	ANI11	ANI12
1’b0	6’h0B	ANI11	ANI12	ANI13
1’b0	6’h0C	ANI12	ANI13	ANI14
1’b0	6’h0D	ANI13	ANI14	ANI15
上述以外		设定禁止		

- 2 通道扫描模式（ADM1.ADMD=1）

ADISS	ADS[5:0]	模拟输入通道	
		扫描 0	扫描 1
1’b0	6’h00	ANI0	ANI1
1’b0	6’h01	ANI1	ANI2
1’b0	6’h02	ANI2	ANI3
1’b0	6’h03	ANI3	ANI4
1’b0	6’h04	ANI4	ANI5
1’b0	6’h05	ANI5	ANI6
1’b0	6’h06	ANI6	ANI7
1’b0	6’h07	ANI7	ANI8
1’b0	6’h08	ANI8	ANI9
1’b0	6’h09	ANI9	ANI10
1’b0	6’h0A	ANI10	ANI11
1’b0	6’h0B	ANI11	ANI12
1’b0	6’h0C	ANI12	ANI13
1’b0	6’h0D	ANI13	ANI14
1’b0	6’h0E	ANI14	ANI15
上述以外		设定禁止	

注意：

1. 对于由 PMCxx 寄存器设定为模拟输入的端口，才可以通过 ADS 指定为模拟输入进行 A/D 转换。
2. 要改写 ADISS 位时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。
3. 当将 AVREFP 用作 A/D 转换器的正（+）基准电压时，不能选择 ANI9 作为 A/D 转换通道。
4. 当将 AVREFM 用作 A/D 转换器的负（-）基准电压时，不能选择 ANI8 作为 A/D 转换通道。
5. 在将 ADISS 位置“1”后，不能使用第 1 次的转换结果。有关详细设定流程，请参照“11.5.4 选择温度传感器的输出电压/内部基准电压时的设定”。
6. 要转移到深睡眠模式时或者要在 CPU 以副系统时钟运行中转移到睡眠模式时，不能将 ADISS 位置“1”。

11.2.7 12位A/D转换结果寄存器（ADCR）

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

图 11-10：12 位 A/D 转换结果寄存器（ADCR）的格式

复位值：0000H R

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

注意：

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

- 选择模式（ADM1.ADMD=0）

  ADCH0~3 的读出值固定为 4‘b0000

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

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

11.2.8 8位A/D转换结果寄存器（ADCRH）

这是保存 A/D 转换结果的 8 位寄存器，保存 12 位分辨率的高 8 位[注]。

通过 8 位存储器操作指令读 ADCRH 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注：如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定）的设定值范围内，就不保存 A/D 转换结果。

图 11-11：8 位 A/D 转换结果寄存器（ADCRH）的格式

复位值：00H R

	7	6	5	4	3	2	1	0
ADCRH								

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

11.2.9 转换结果比较上限值设定寄存器（ADUL）

这是用于检查 A/D 转换结果上限值的设定寄存器。

将 A/D 转换结果和 ADUL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设定范围内（参照

图 11-7：ADRCK 位的中断信号产生范围）控制中断信号（INTAD）的产生。通过 8 位存储器操作指令设定 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. 在设定 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL＞ADLL。

图 11-12：转换结果比较上限值设定寄存器（ADUL）的格式

复位值：FFH R/W

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

11.2.10 转换结果比较下限值设定寄存器（ADLL）

这是用于检查 A/D 转换结果下限值的设定寄存器。

将 A/D 转换结果和 ADLL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设定范围内（参照

图 11-7：ADRCK 位的中断信号产生范围）控制中断信号（INTAD）的产生。通过 8 位存储器操作指令设定 ADLL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-13：转换结果比较下限值设定寄存器（ADLL）的格式

复位值：00H R/W

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

注意：

1. 只将 12 位 A/D 转换结果寄存器（ADCR）的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。
2. 要改写 ADUL 寄存器和 ADLL 寄存器时，必须在转换停止状态（ADCS=0）下进行。
3. 在设定 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL＞ADLL。

11.2.11 A/D采样时间控制寄存器（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定 ADNSMP 寄存器。

在产生复位信号后，此寄存器的值变为“0dH”。

图 11-14：A/D 采样时间控制寄存器（ADNSMP）的格式

复位值：0dH R/W

	7	6	5	4	3	2	1	0
ADNSMP	ADNSMP[7:0]							

采样时钟数设定：

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

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

进行一次 ADC 转换需要的时间：

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

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

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

不同条件下，各转换通道推荐的采样时间见下表：

采样时间计算公式：采样时钟个数/F_{AD}≥推荐的采样时间

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

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

11.2.12 A/D采样时间延长寄存器（ADSMPWAIT）

此寄存器用于延长 A/D 采样时间。

通过 8 位存储器操作指令设定 ADSMPWAIT 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-15：A/D 采样时间延长寄存器（ADSMPWAIT）的格式

复位值：00H R/W

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

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

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

11.2.13 A/D测试寄存器（ADTES）

此寄存器用于设定 A/D 转换器的测试模式。

通过 8 位存储器操作指令设定 ADTES 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-16：A/D 测试寄存器（ADTES）的格式

复位值：00H R/W

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

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

注意：必须将 bit7~3 置“0”。

11.2.14 A/D充放电控制寄存器（ADNDIS）

此寄存器用来控制 A/D 转换器的充放电动作及时间。

通过 8 位存储器操作指令读写 ADNDIS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-17：A/D 充放电控制寄存器（ADNDIS）的格式

复位值：00HW

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

ADNDIS[4]	充放电控制
1’b0	放电
1’b1	充电

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

注意：禁止设定充放电时间为 1 个 ADCLK，即 ADNDIS[3:0]=4‘b0001

11.3 输入电压和转换结果

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

$$ADCR=INT(\frac{V_{AIN}}{AV_{REF}}\times 4096+0.5) \quad or \quad (ADCR-0.5)\times\frac{AV_{REF}}{4096}\le V_{AIN}<(ADCR+0.5)\times\frac{AV_{REF}}{4096}$$

INT()：将括号中的数值的整数部分返回的函数

V_{AIN}：模拟输入电压

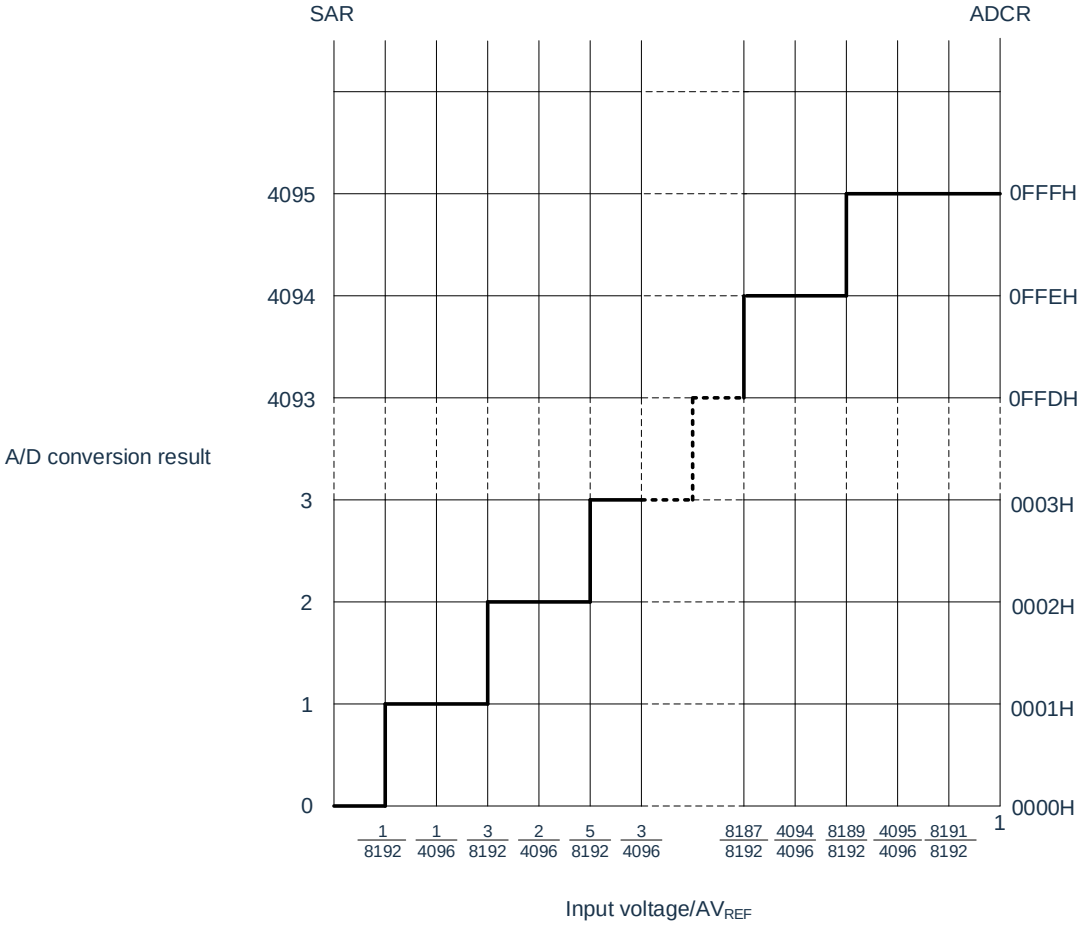
AV_{REF}：AV_{REF} 引脚电压

ADCR：A/D 转换结果寄存器（ADCR）的值

SAR：逐次逼近寄存器

模拟输入电压和 A/D 转换结果的关系如下图所示。

图 11-18：模拟输入电压和 A/D 转换结果的关系



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

11.4 A/D转换器的运行模式

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

11.4.1 软件触发模式（选择模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。

③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在A/D转换结束后立即开始下一次A/D转换。

④ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

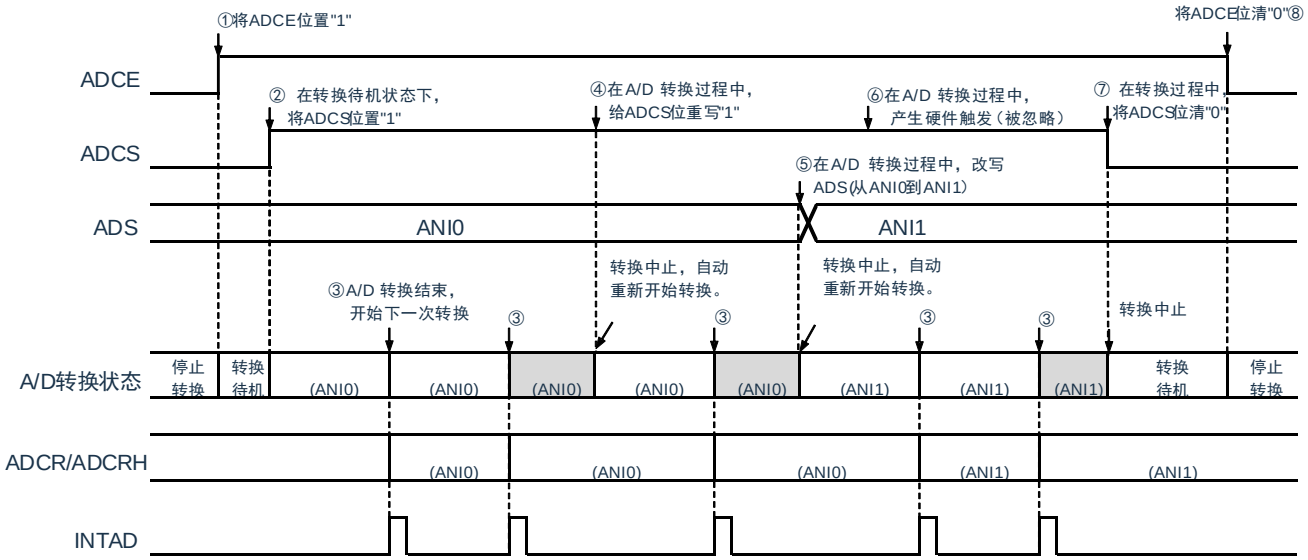
⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对由ADS寄存器重新指定的模拟输入进行A/D转换。

⑥ 即使在转换过程中输入硬件触发也不开始A/D转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。

⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

图 11-19：软件触发模式（选择模式、连续转换模式）的运行时序例子



11.4.2 软件触发模式（选择模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。

③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

④ 在A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。

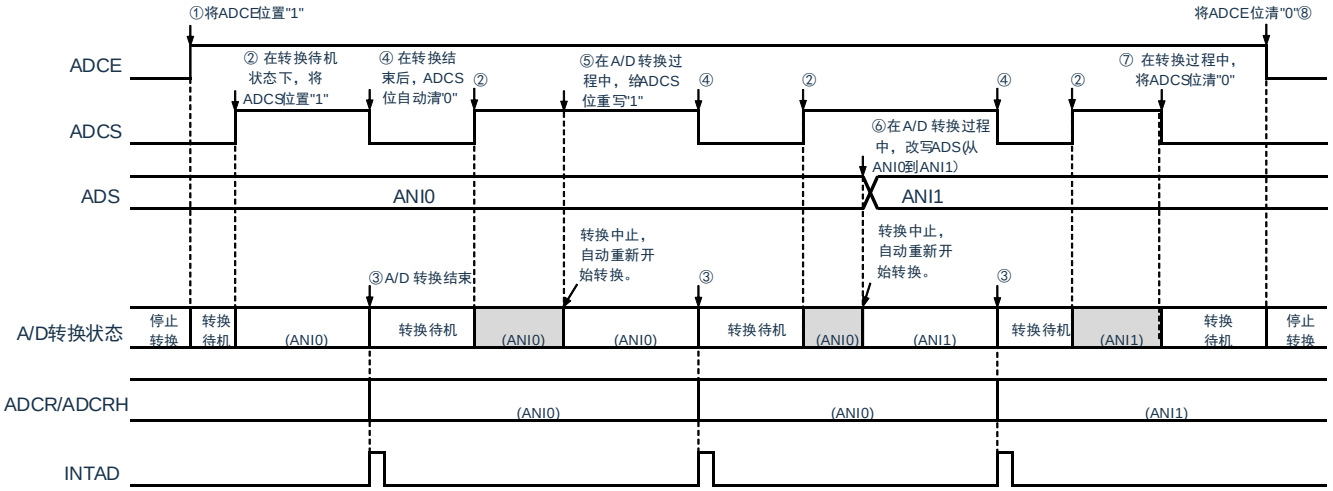
⑤ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对由ADS寄存器重新指定的模拟输入进行A/D转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。

⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

图 11-20：软件触发模式（选择模式、单次转换模式）的运行时序例子



11.4.3 软件触发模式（扫描模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。

③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换（4个通道）。

④ 如果在转换过程中给ADCS位重写“1”，当前A/D转换立刻中止，然后重新开始转换。

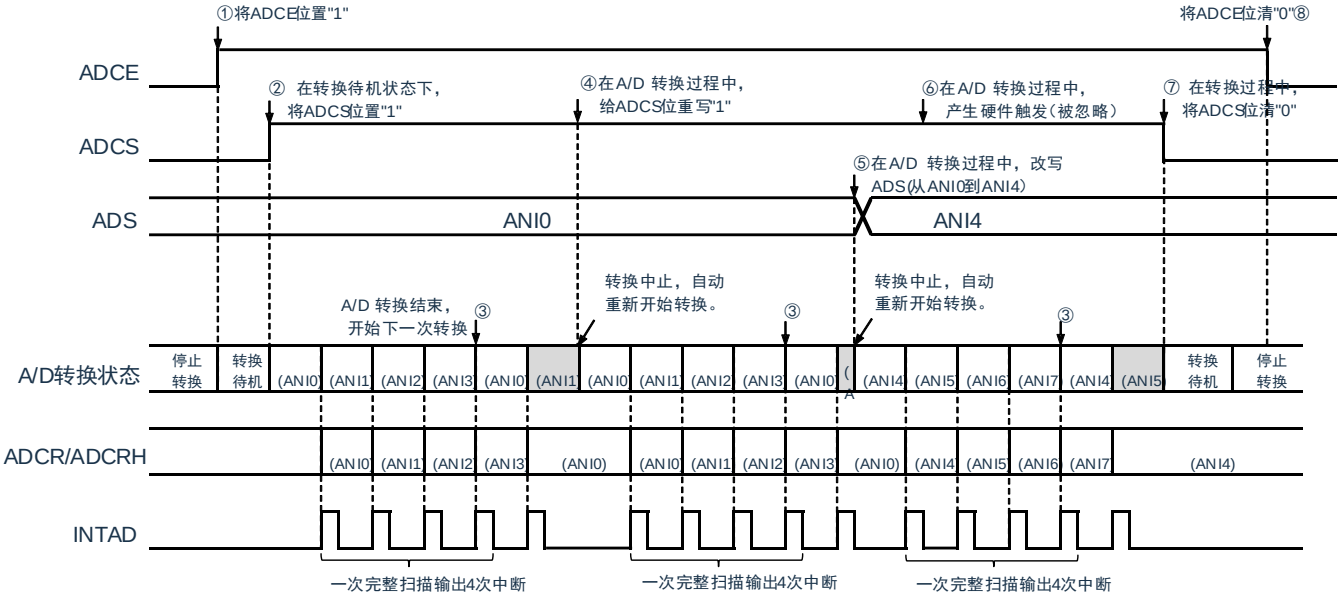
⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的最初通道进行A/D转换。

⑥ 即使在转换过程中输入硬件触发也不开始A/D转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。

⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

图 11-21：软件触发模式（扫描模式、连续转换模式）的运行时序例子



11.4.4 软件触发模式（扫描模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。

③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

④ 在4个通道的A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。

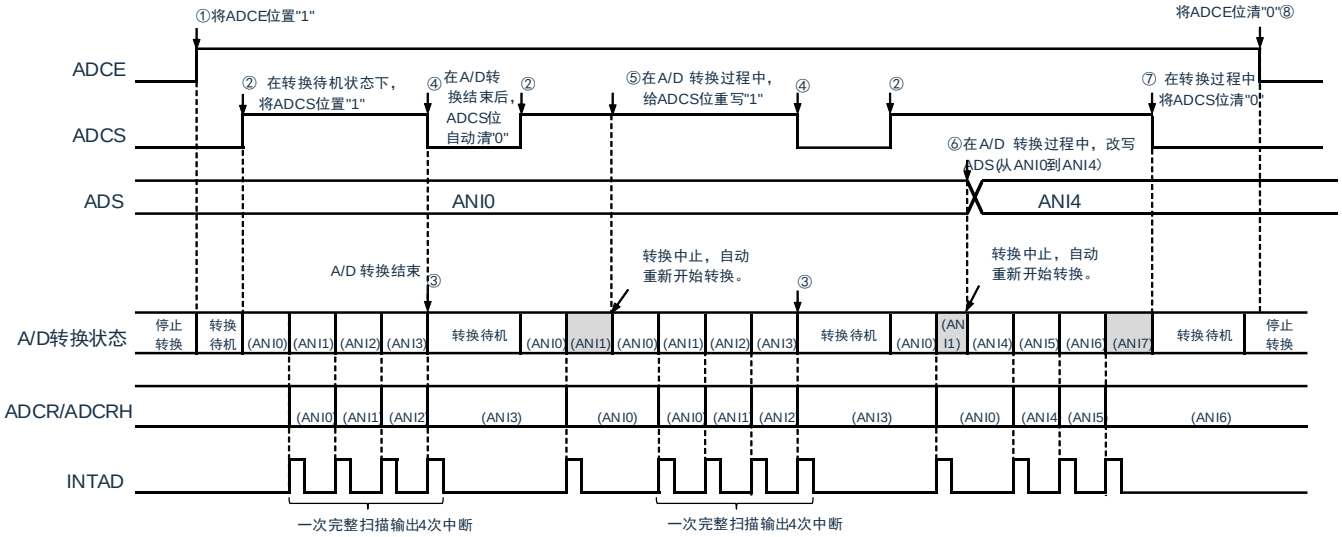
⑤ 如果在转换过程中给ADCS位重写“1”，当前A/D转换立刻中止，然后重新开始转换。

⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的最初通道进行A/D转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。

⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

图 11-22：软件触发模式（扫描模式、单次转换模式）的运行时序例子



11.4.5 硬件触发无等待模式（选择模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。

③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。

④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在A/D转换结束后立即开始下一次A/D转换。

⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。

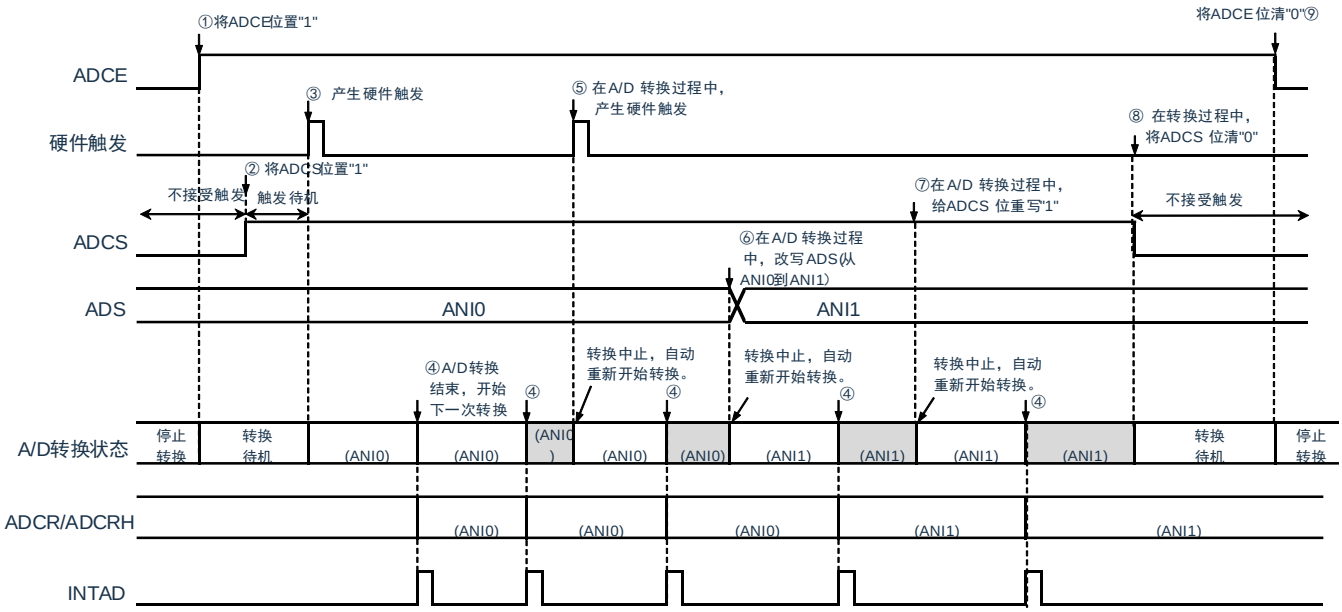
⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。

⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。

⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-23：硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



11.4.6 硬件触发无等待模式（选择模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。

③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。

④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

⑤ 在A/D转换结束后，ADCS位保持“1”的状态，进入A/D转换待机状态。

⑥ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。

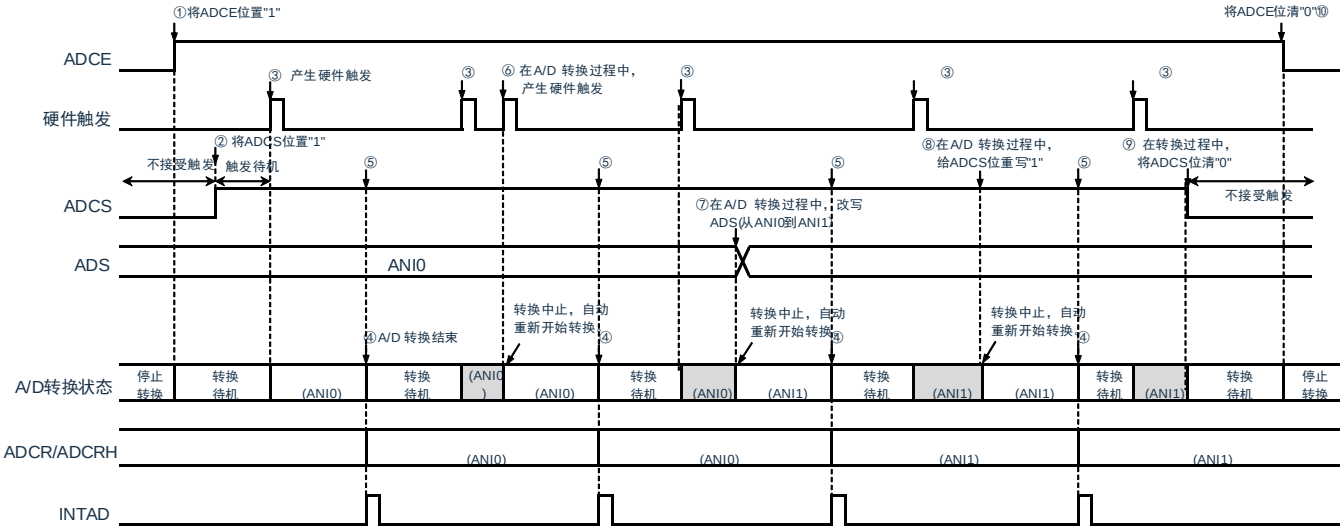
⑦ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。

⑧ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

⑨ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻停止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。

⑩ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-24：硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



11.4.7 硬件触发无等待模式（扫描模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。

③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。

④ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换。

⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。

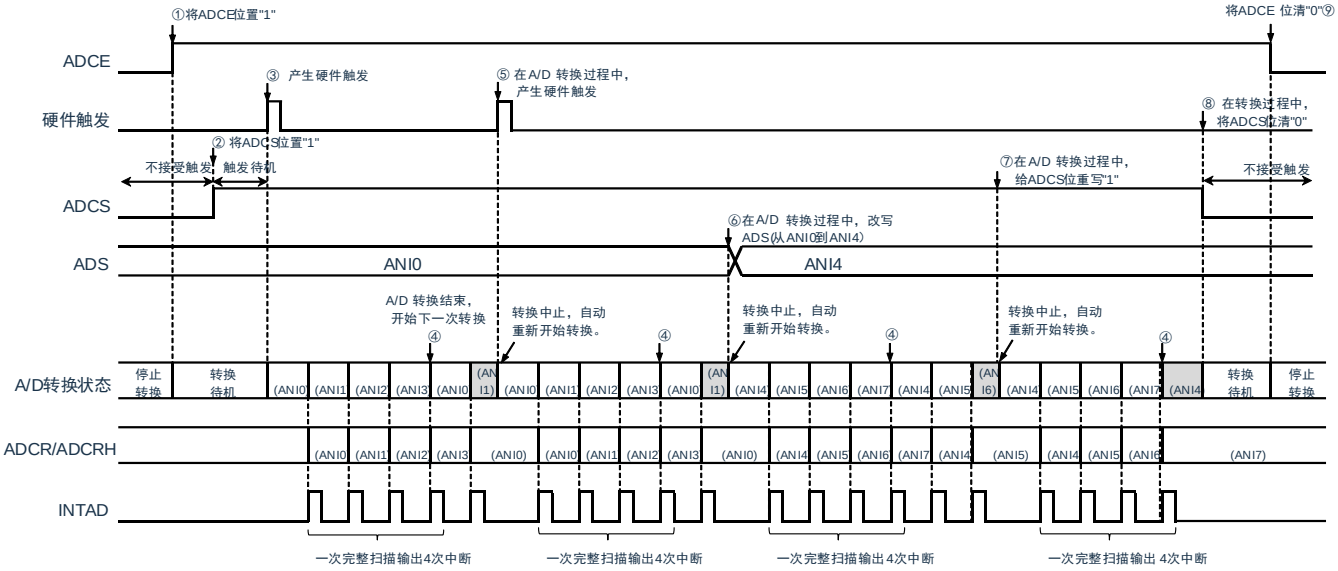
⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的通道进行A/D转换。

⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。

⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。

⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

图 11-25：硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



11.4.8 硬件触发无等待模式（扫描模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。

② 在通过软件对稳定等待时间（1us）进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。

③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。

④ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

⑤ 在4个通道的A/D转换结束后，ADCS位保持“1”的状态，进入A/D转换待机状态。

⑥ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。

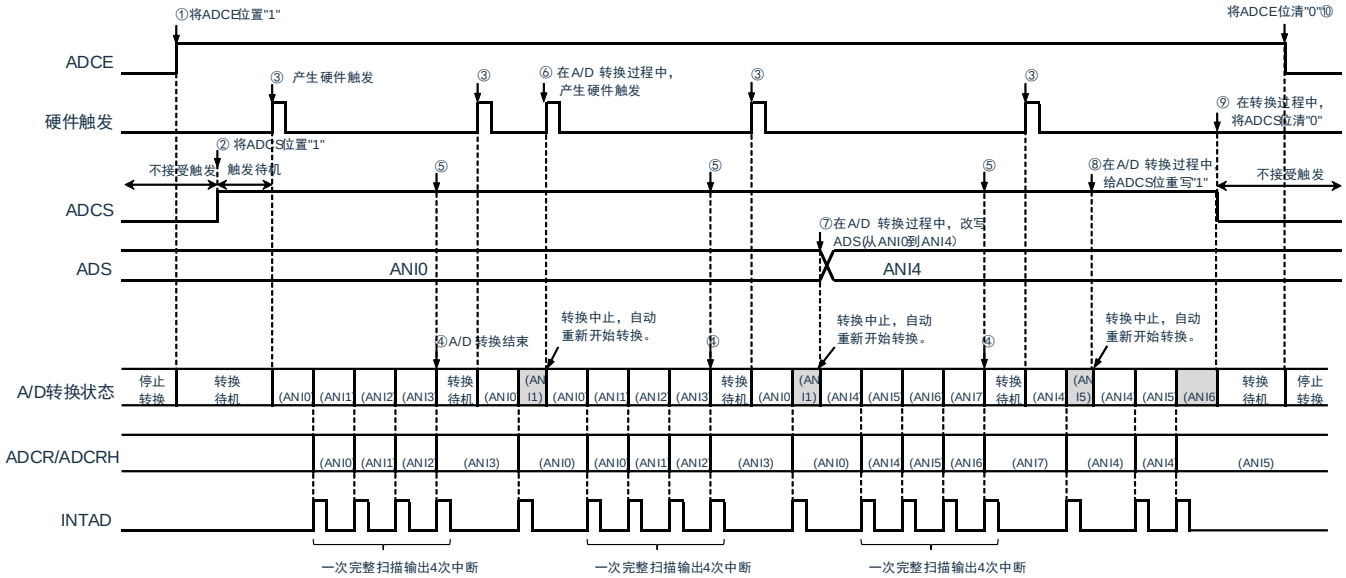
⑦ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的最初通道进行A/D转换。

⑧ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新从最初的通道开始转换。

⑨ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。

⑩ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-26：硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



11.4.9 硬件触发等待模式（选择模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。

② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。

③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在A/D转换结束后立即开始下一次A/D转换（此时，不需要硬件触发）。

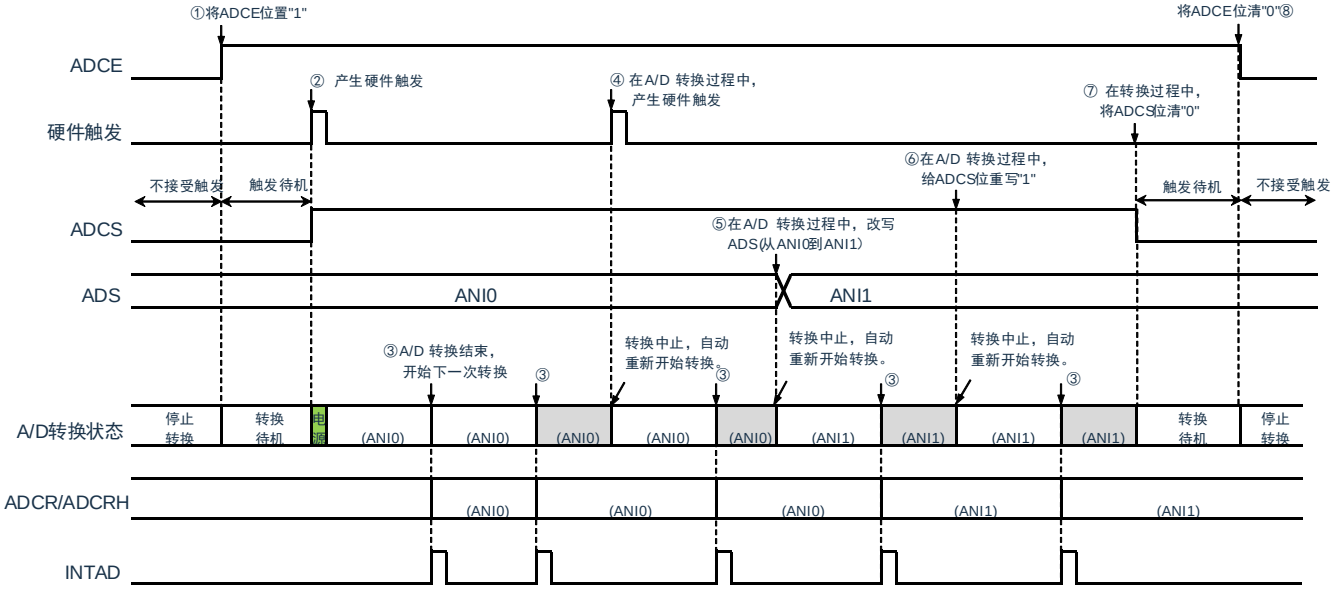
④ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。

⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。

⑥ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-27：硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



电源 电源安定等待时间

11.4.10 硬件触发等待模式（选择模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。

② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。

③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。

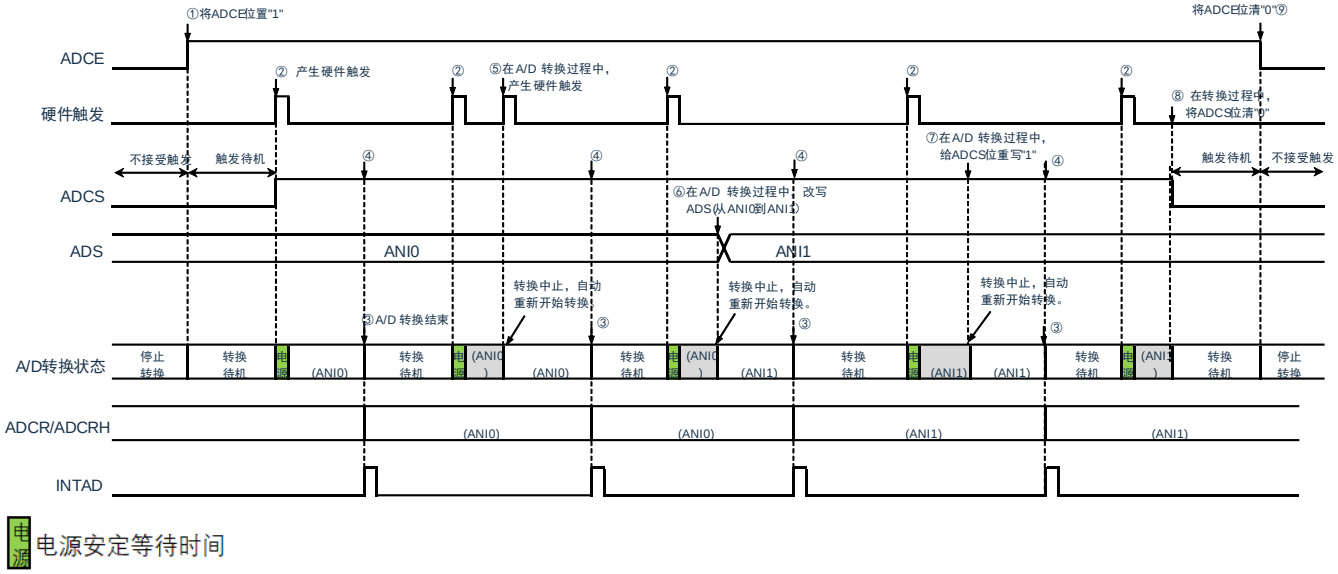
⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新开始转换。

⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后对ADS寄存器重新指定的模拟输入进行A/D转换。

⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后重新开始转换。

⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-28：硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



电源 电源安定等待时间

11.4.11 硬件触发等待模式（扫描模式、连续转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。

② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。依次从扫描0指定的模拟输入通道进行A/D转换。

③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换。

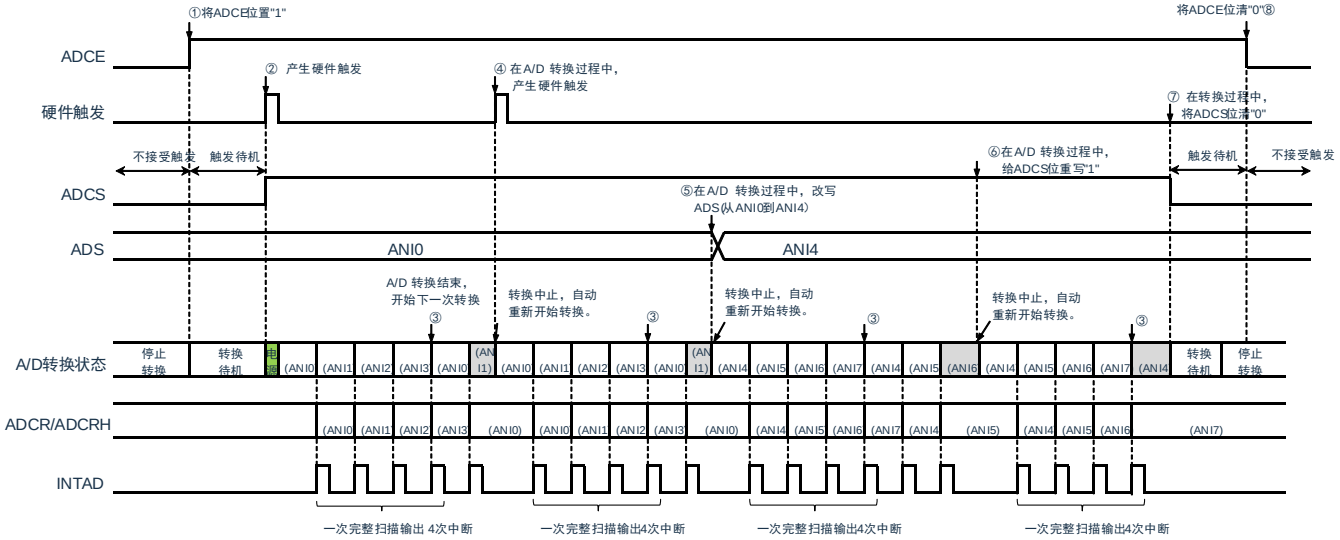
④ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。

⑤ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的通道开始进行扫描转换。

⑥ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后从最初的通道重新开始转换。

⑦ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-29：硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子



电源 电源安定等待时间

11.4.12 硬件触发等待模式（扫描模式、单次转换模式）

① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。

② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0~扫描3的4个模拟输入通道进行A/D转换。在输入硬件触发后自动将ADM0寄存器的ADCS位置“1”。依次从扫描0指定的模拟输入通道进行A/D转换。

③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。

④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。

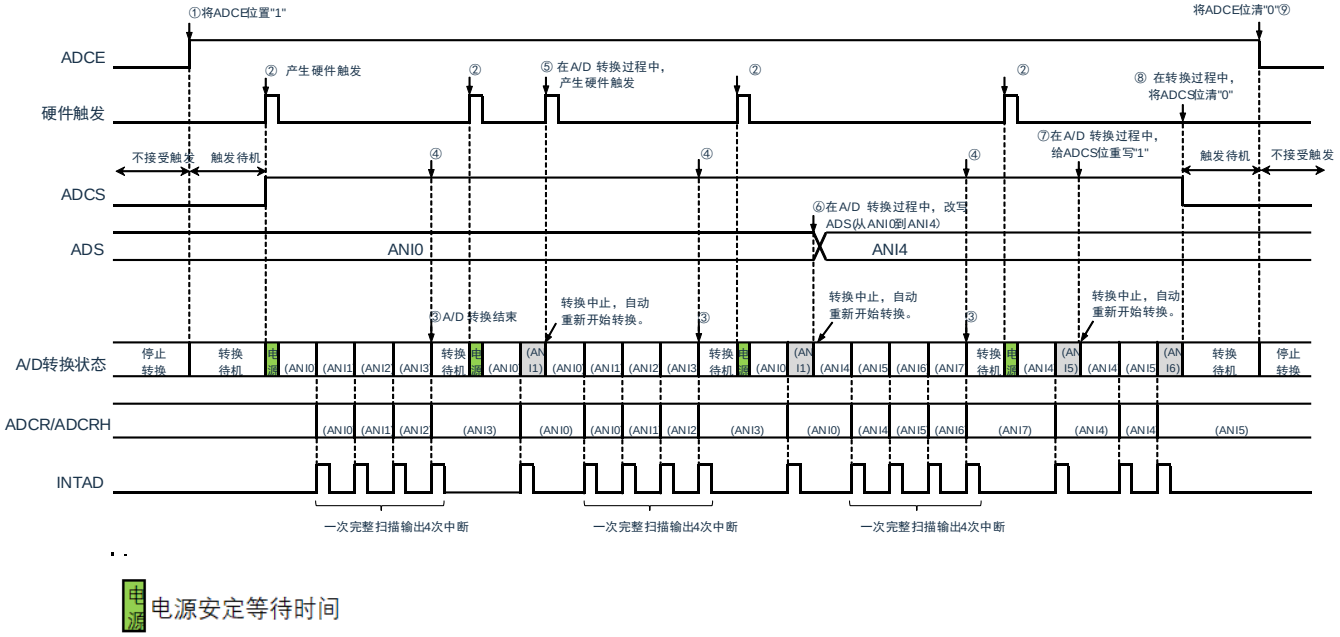
⑤ 如果在转换过程中输入硬件触发，当前的A/D转换立刻中止，然后重新从最初的通道开始扫描转换。

⑥ 如果在转换过程中改写或者重写ADS寄存器，当前的A/D转换立刻中止，然后从由ADS寄存器重新指定的通道开始扫描转换。

⑦ 如果在转换过程中给ADCS位重写“1”，当前的A/D转换立刻中止，然后从最初的通道开始扫描转换。

⑧ 如果在转换过程中将ADCS位置“0”，当前的A/D转换立刻中止，然后进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 11-30：硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子



11.5 A/D转换器的设定流程图

各运行模式的 A/D 转换器的设定流程图如下所示。

11.5.1 软件触发模式的设定

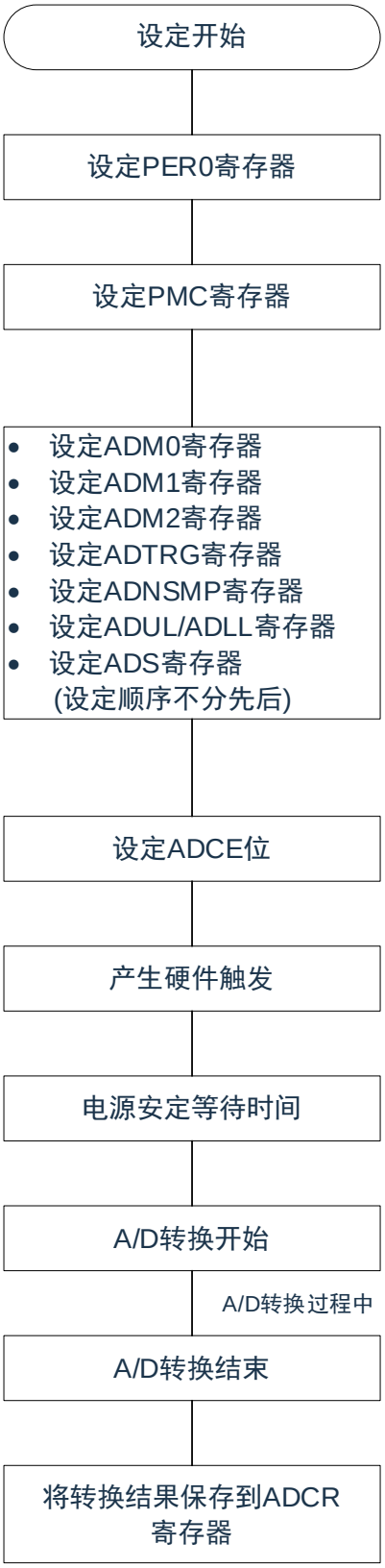
图 11-31：软件触发模式的设定

11.5.2 硬件触发无等待模式的设定

图 11-32：硬件触发无等待模式的设定

11.5.3 硬件触发等待模式的设定

图 11-33：硬件触发等待模式的设定



将PER0寄存器的ADCEN位置“1”，开始提供时钟。

将端口设定为模拟输入：通过PMCx寄存器进行设定。

- ADM0寄存器
  FR2~FR0位：设定A/D转换时间。
- ADM1寄存器
  ADSCM位：连续转换模式/单次转换模式。
  ADMD位：选择模式/扫描模式。
- ADM2寄存器
  ADREFP位、ADREFM位：选择基准电压。
  ADRCK位：选择产生中断信号的A/D转换结果比较值的范围。
- ADTRG寄存器
  ADTMD1位、ADTMD0位：设定为硬件触发等待模式。
  ADTRS1位、ADTRS0位：选择触发源。
- ADNSMP寄存器
  设定采样时间。
- ADUL/ADLL寄存器
  设定A/D转换结果比较值的上限值和下限值。
- ADS寄存器
  选择模拟输入通道。

将ADM0寄存器的ADCE位置“1”，进入硬件触发等待模式。

硬件自动对A/D电源安定等待1us时间。

在结束A/D电源安定等待时间计数结束后，开始A/D转换。

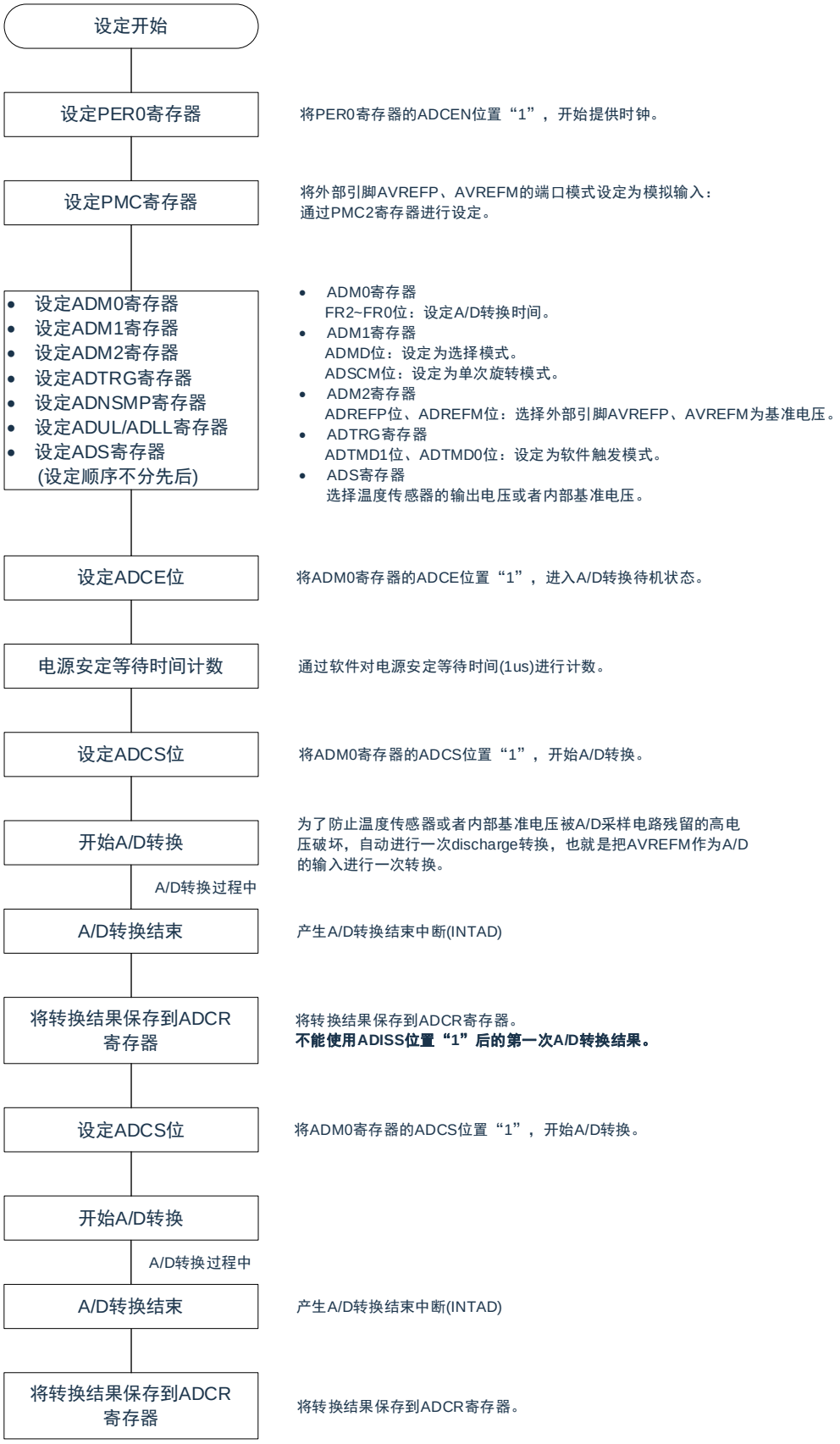
根据ADRCK位和ADUL/ADLL寄存器的设定，和A/D转换结果的比较结果，硬件自动判断是否产生A/D转换结束中断(INTAD)。

如果A/D转换结束中断(INTAD)产生，则将转换结果保存到ADCR寄存器，否则，不将转换结果保存到ADCR寄存器。

11.5.4 选择温度传感器的输出电压/内部基准电压时的设定

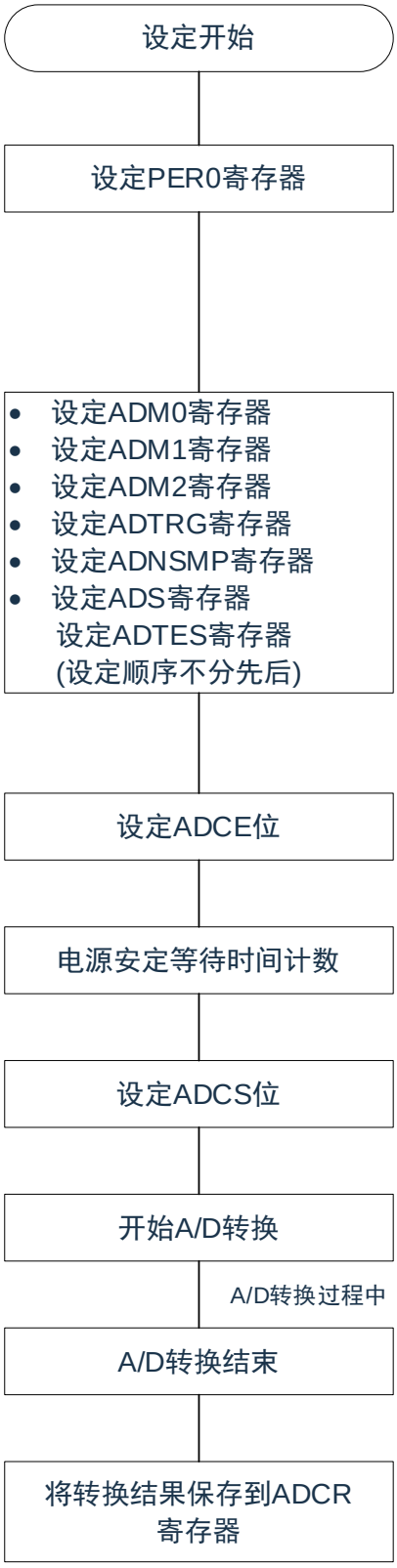
（以软件触发模式、单次转换模式为例）

图 11-34：选择温度传感器的输出电压/内部基准电压时的设定



11.5.5 测试模式的设定

图 11-35：测试模式的设定(V_{SS}/half_V_{DD}/V_{DD} 作为转换对象)



将PER0寄存器的ADCEN位置“1”，开始提供时钟。

- ADM0寄存器
  FR2~FR0位：设定A/D转换时间。
- ADM1寄存器
  ADSCM位：连续转换模式/单次转换模式。
  ADMD位：选择模式。
- ADM2寄存器
  ADREFP位、ADREFM位：选择基准电压。
- ADTRG寄存器
  ADTMD1位、ADTMD0位：设定为软件触发模式。
- ADNSMP寄存器
  设定采样时间。
- ADS寄存器
  关闭所有模拟输入通道。
- ADTES寄存器
  ADTES1位、ADTES0位：选择V_{SS}/二分之一V_{DD}/V_{DD}作为转换对象。

将ADM0寄存器的ADCE位置“1”，进入A/D转换待机状态。

通过软件对电源安定等待时间(1us)进行计数。

将ADM0寄存器的ADCS位置“1”，开始A/D转换。

产生A/D转换结束中断(INTAD)。

将转换结果保存到ADCR寄存器。

第12章 SIGMA-DELTA ADC

12.1 概述

24-Bit Sigma-Delta ADC 是高精度、低功耗模数转换模块。支持 1 路差分输入通道，内置一路线性稳压器（LDO）、温度传感器和高精度振荡器。LDO 可驱动 20mA 负载。Sigma-Delta ADC 的 PGA 放大倍数可选：1、2、4、8、16、32、64、128、256。Sigma-Delta ADC 正常模式下的 ADC 数据输出速率可选：2.5Hz-2.56KHz，默认为 5Hz。MCU 内部通过 2 线的 SPI 接口 SCLK、DRDYB/DOUT 与 Sigma-Delta ADC 进行通信，对其进行配置，例如通道选择、PGA 放大倍数选择、输出速率选择等。

12.2 基本结构功能描述

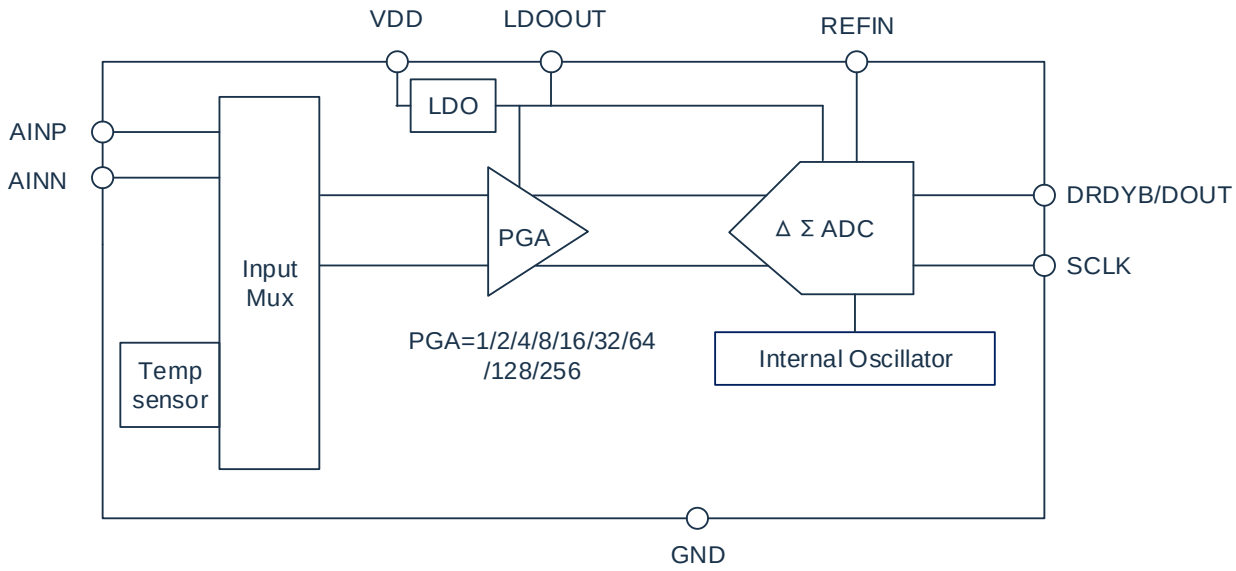
ADC 采用 Sigma-Delta 调制器，通过低噪声仪用放大器结构实现 PGA 放大，放大倍数可选：1、2、4、8、16、32、64、128、256。在 PGA=128，输出速率=10Hz 时，有效分辨率可达 20.6 位。

Sigma-Delta ADC 内置振荡器，无需外置晶振。

Sigma-Delta ADC 可以通过 DRDYB /DOUT 和 SCLK 进行多种功能模式的配置，例如用作温度检测、PGA 增益选择、ADC 数据输出速率选择等等。

Sigma-Delta ADC 具有休眠（Sleep）模式。

Sigma-Delta ADC 工作原理图框图如下：



12.3 ADC工作原理

12.3.1 LDO

Sigma-Delta ADC 中有 1 路 LDO 为片内模拟模块供电，同时也可以为片外电路提供 20mA 电流。LDOOUT 端口需外接至少 1uF 电容。可以通过配置 SEL_LDO[1:0]来设置 LDO 输出不同的输出电压，可以选择 2.4V、2.6V、3V、3.3V。也可以通过设置 BYPASSLDO 来配置 LDO 工作为开关状态或者线性稳压状态。当芯片进入休眠模式时，LDO 的输出降至 0V。

12.3.2 模拟输入前端

Sigma-Delta ADC 中有 1 路 ADC，集成了 1 路差分输入，信号输入可以是差分输入信号通道（AINP/AINN），也可以是温度传感器的输出信号。输入信号的切换由寄存器 CH_SEL[3:0]控制，CH_SEL[3]位于寄存器 SDADCCON1，CH_SEL[2:0] 位于寄存器 SDADCCON2。

12.3.3 温度传感器

模块内部提供温度测量功能。建议采用 PGA 增益 8，ADC 速度为 640Hz 的配置。温度传感器需要进行单点校正。校正方法：在某个温度点 A 下，使用温度传感器进行测量得到码值 Ya。那么其他温度点 B 对应的温度= Yb*(273.15+A)/Ya-273.15。A 温度单位是摄氏度。Ya 是 A 点对应温度码值。Yb 是 B 点对应温度码值。

对于参考电压可能变化的应用场合（如 Ratio Metric 测量，参考电压的绝对值不重要），可选择通过测量 BG 间接测量出实时的参考电压，进而计算出实时的温度。

12.3.4 低噪声PGA放大器

Sigma-Delta ADC 提供了一个基于斩波技术的低噪声、低漂移的 PGA 放大器与桥式传感器差分输出连接，通过 PGA_SEL[3:0] 来配置 2、4、8、16、32、64 、128、256 等不同的 PGA。当使用 PGA=2，4，8 时，第一级低噪声 PGA 放大器会被关断以节省功耗。当使用低噪声 PGA 放大器时，输入范围在 GND+0.75V 到 VDD-1V 之间，超出这个范围，会导致实际性能下降。当模拟输入跳过 PGA，直接输入至 ADC 时，增益为 1。

12.3.5 ADC时钟、数据输出速率

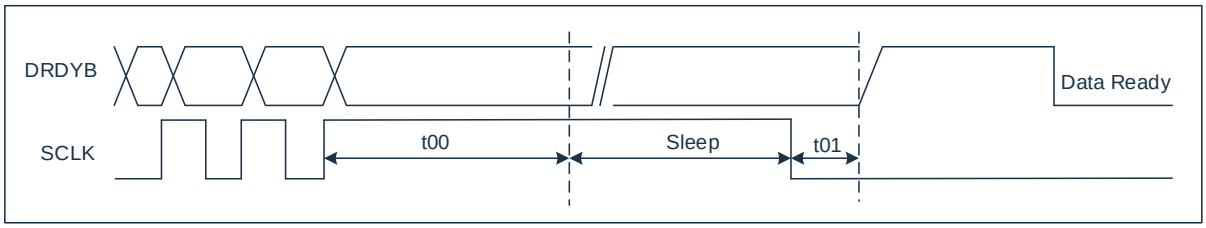
Sigma-Delta ADC 使用内部时钟来提供系统所需要的时钟频率，可通过 FADC 选择 328KHz 或者 656KHz。ADC 的输出速率可以通过 FADC、OSR[2:0]进行配置。时钟、输出速率、斩波频率如下表所示。

FADC	OSR[2:0]	输出速率	ADC 时钟
0	111	2.5Hz	328KHz
0	110	5Hz	328KHz
0	101	10Hz	328KHz
0	100	20Hz	328KHz
0	011	80Hz	328KHz
0	010	320Hz	328KHz
0	001	640Hz	328KHz
0	000	1.28KHz	328KHz
1	111	5Hz	656KHz
1	110	10Hz	656KHz
1	101	20Hz	656KHz
1	100	40Hz	656KHz
1	011	160Hz	656KHz
1	010	640Hz	656KHz
1	001	1.28KHz	656KHz
1	000	2.56KHz	656KHz

12.3.6 复位和休眠模式

当模块上电时，内置上电复位电路会产生复位信号，使模块自动复位。当 SCLK 从低电平变高电平并保持在高电平超过 100us，Sigma-Delta ADC 即进入休眠模式，此时功耗低于 50nA。当 SCLK 重新回到低电平时，模块会重新进入正常工作状态。当系统由休眠重新进入正常工作模式时，此时所有功能配置为休眠之前的状态，不需要进行功能配置。

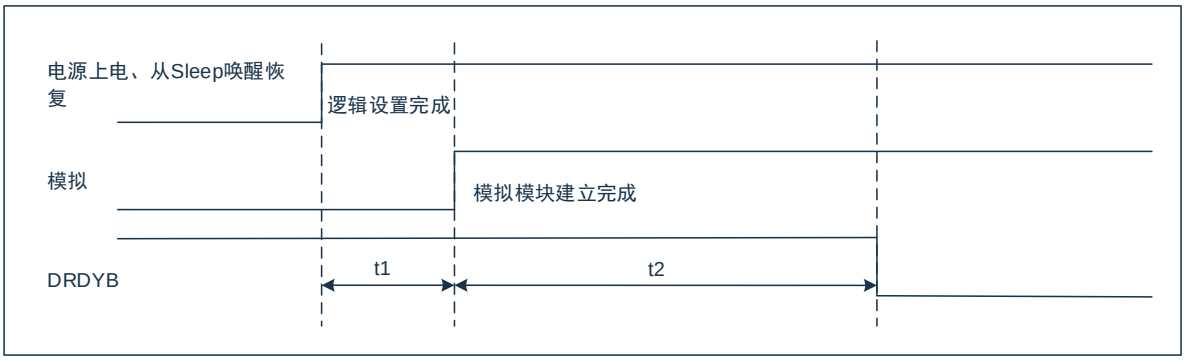
休眠模式示意图如下图所示。其中 t00 表示 SCLK 高电平保持时间，最小为 100us；t01 表示 SCLK 下降后低电平保持时间，最小为 10us。



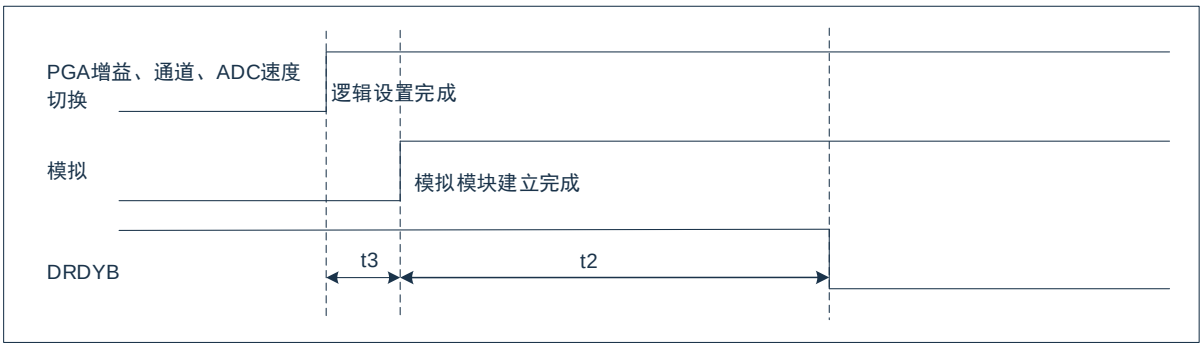
12.3.7 建立时间

Sigma-Delta ADC 的建立时间均为 3 个转换周期。

数据建立过程 1：



数据建立过程 2：



建立时间：

参数	描述	最小值	典型值	最大值	单位
建立时间					
t1	上电、从休眠唤醒恢复	-	0.4	-	ms
t2	数据建立时间	-	3	-	转换周期
t3	PGA、通道、ADC 速度切换后恢复时间	-	0.8	-	us

12.4 SPI串口通信

Sigma-Delta ADC 中采用 2 线 SPI 串行通信，通过 SCLK 和 DRDYB/DOUT 可以实现数据的接收以及功能配置。

12.4.1 数据格式

Sigma-Delta ADC 输出的数据为数字输出码为 24 位的 2 进制补码，其中 B23 为符号位，0 为正，1 为负。最高位（MSB）最先输出。下表为不同模拟输入信号对应的理想输出码。

模拟输入电压	数据																								十进制码	十六进制码
	B23	B22	B21	B20	B19	B18	B17	B16	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0		B[23:0]
Vref-1LSB	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8388607	7FFFFF
2LSB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	000002
1LSB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	000001
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000000
-1LSB	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	FFFFFF
-2LSB	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-2	FFFFFE
-Vref	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8388608	800000

12.4.2 数据准备/数据输入输出（DRDYB/DOUT）

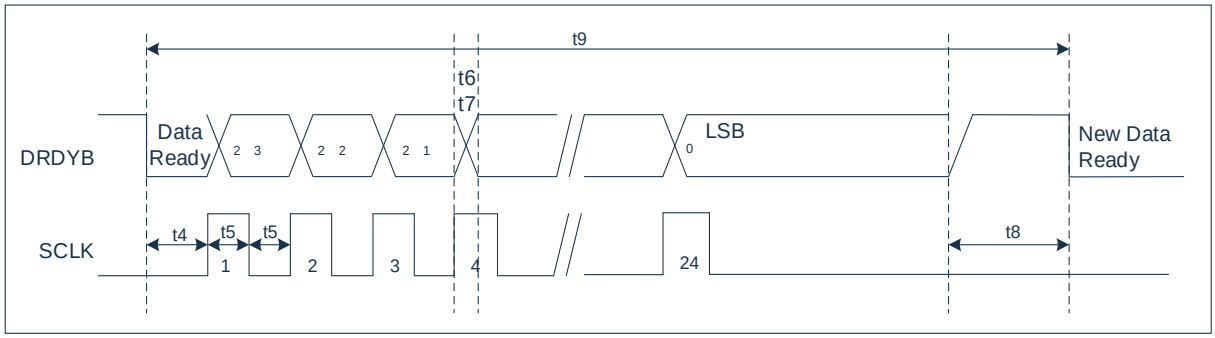
DRDYB/DOUT 引脚有 4 个用途。第一，当输出为低时，表示新的数据已经转换完成；第二，作为数据输出引脚，当数据准备好后，在第 1 个 SCLK 的上升沿后，DRDYB/DOUT 输出转换数据的符号位。在每一个 SCLK 的上升沿，数据会自动移 1 位。在 24 个 SCLK 后将所有的 24 位数据读出，如果这时暂停 SCLK 的发送，DRDYB/DOUT 会保持最后一位的数据，直到下一个数据准备好之前拉高，此后当 DRDYB/DOUT 被再次拉低，表示新的数据已经转换完成，可进行下一个数据读取；第三，在第 25、26 个 SCLK 时，输出寄存器状态更新标志；第四，作为寄存器数据写入或读出引脚，当需要配置寄存器或读取寄存器值时，SPI 需要发送 46 个 SCLK，根据 DRDYB/DOUT 输入的命令字，判断是写寄存器操作还是读寄存器操作。

12.4.3 串行时钟输入（SCLK）

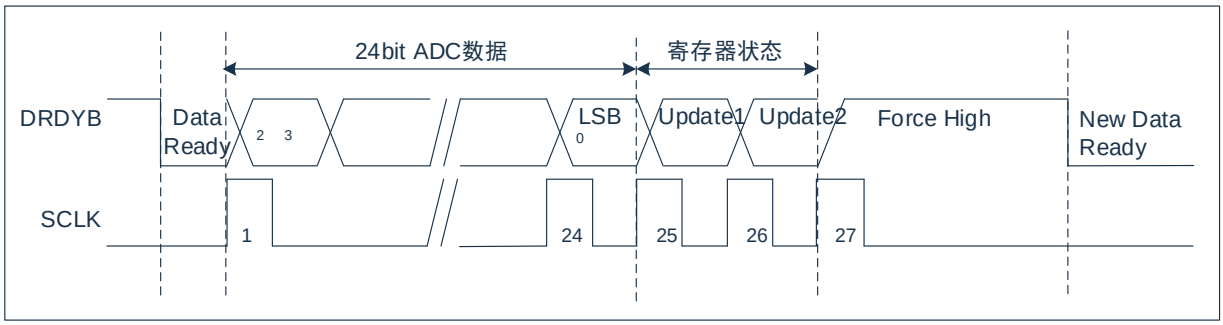
串行时钟输入 SCLK 是一个数字引脚。这个信号应保证是一个干净的信号，毛刺或慢速的上升沿都会可能导致读取错误数据或误入错误状态。因此，应保证 SCLK 的上升和下降时间都小于 50ns。

12.4.4 串行数据发送

读取数据时序图 1：



读取数据时序图 2：



Sigma-Delta ADC 可以持续的转换模拟输入信号，当将 DRDYB/DOUT 拉低后，表明数据已经准备好接受，输入的第一个 SCLK 来就可以将输出的最高位读出，在 24 个 SCLK 后将所有的 24 位数据读出，如果这时暂停 SCLK 的发送，DRDYB/DOUT 会保持最后一位的数据，直到其被拉高，如时序图 1。

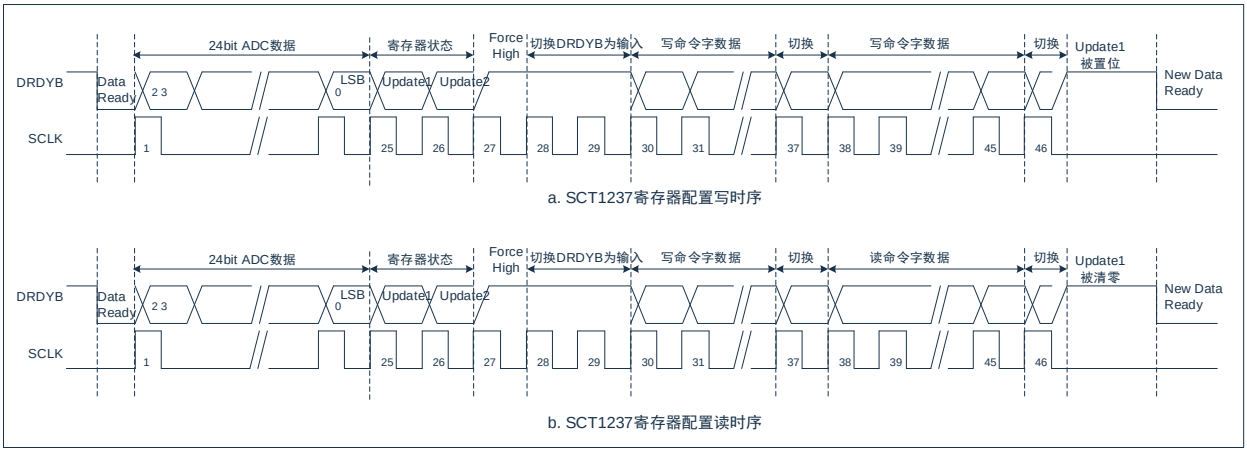
如果是持续发送 SCLK，则第 25 和 26 个 SCLK 输出配置寄存器是否有写操作标志，第 25 个 SCLK 对应的 DRDYB/DOUT 为 1 时表明配置寄存器 Config 被写入了新的值，第 26 个 SCLK 对应的 DRDYB/DOUT 为模块扩展保留位，目前输出一直为 0，通过第 27 个 SCLK 可以将 DRDYB/DOUT 拉高，此后当 DRDYB/DOUT 被再次拉低，表示新的数据已经准备好接受，进行下一个数据的转换。其基本时序如时序图 2 所示。

读取数据时间表：

参数	描述	最小值	典型值	最大值	单位
t4	DRDYB/DOUT 变低后到第一个 SCLK 上升沿	-	2	-	ns
t5	SCLK 高电平或低电平脉宽	455	-	-	ns
t6	SCLK 上升沿到新数据有效（传输延迟）	455	-	-	ns
t7	SCLK 上升沿到旧数据位有效(保持时间)	-	-	455	ns
t8	数据更新，不允许读之前的数据	-	26	-	us
t9	转换时间，10Hz	-	100	-	ms
	转换时间，40Hz	-	25	-	ms
	转换时间，640Hz	-	1.5625	-	ms

12.4.5 功能配置

Sigma-Delta ADC 可以通过 SCLK 和 DRDYB/DOUT 可以对寄存器进行读取和配置，功能配置时序如下所示：



功能配置过程简述，判断到 DRDYB/DOUT 由高变低之后：

1) 第 1 个到第 24 个 SCLK，读取 ADC 数据。如果不需要配置寄存器或者读取寄存器，可以省略下面的步骤。
2) 第 25 个到第 26 个 SCLK，读取寄存器写操作状态。
3) 第 27 个 SCLK，模块把 DRDYB/DOUT 输出拉高。
4) 第 28 个到第 29 个 SCLK，切换 DRDYB/DOUT 为输入。
5) 第 30 个到第 36 个 SCLK，输入寄存器写或读命令字数据(高位先输入)。
6) 第 37 个 SCLK，切换 DRDYB/DOUT 的方向 (如果是写寄存器，DRDYB/DOUT 为输入；如果是读寄存器，DRDYB/DOUT 为输出)。
7) 第 38 个到第 45 个 SCLK，输入寄存器配置数据或输出寄存器配置数据(高位先输入/输出)。
8) 第 46 个 SCLK，切换 DRDYB/DOUT 为输出，并把 DRDYB/DOUT 拉高。update1/ update2 被置位或清零。

12.4.6 SPI命令字说明

Sigma-Delta ADC 有 8 个命令字，命令字的长度为 7bits，命令字的说明如下：

名称	命令字（控制字）	功能	备注
SDADCCON1	0x65	写SDADCCON1	需SPI通信实现寄存器配置
	0x56	读SDADCCON1	需SPI通信实现寄存器配置
SDADCCON2	0x69	写SDADCCON2	需SPI通信实现寄存器配置
	0x5A	读SDADCCON2	需SPI通信实现寄存器配置
SDADCCON3	0x6D	写SDADCCON3	需SPI通信实现寄存器配置
	0x5E	读SDADCCON3	需SPI通信实现寄存器配置
SDADCCON4	0x61	写SDADCCON4	需SPI通信实现寄存器配置
	0x52	读SDADCCON4	需SPI通信实现寄存器配置

12.4.7 SPI通信注意事项

如前所述，Sigma-Delta ADC 可以有三类通信时序，分别为：

时序 1：读取 24 位 ADC 转换数据

时序 2：读取 24 位 ADC 转换数据+寄存器状态

时序 3：读取 24 位 ADC 转换数据+寄存器状态+读写控制字

发送三类时序时需注意：

1) 读取的数据是发送时序前最后一次 ADC 转换完成的结果；
2) 需判断 DRDYB 产生了下降沿后，再发送时钟；
3) 发送任何时序时，从 DRDYB 下降沿到数据全部发送完成的时间小于一个转换时间（即在两次下降沿之间完成时序发送），否则将造成 DRDYB 异常，需进入休眠态后方可恢复。

12.5 相关寄存器

12.5.1 Sigma-Delta ADC控制寄存器1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SDADCCON1	SET_LDO_1	SET_LDO_0	OSR_2	OSR_1	--	PGA_SEL2	PGA_SEL1	PGA_SEL0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	0	1	1	0

Bit7~Bit6 SET_LDO<1:0>: LDO 输出电压控制；
00= 3V；
01= 2.4V；
10= 2.6V；
11= 3.3V。

Bit5~Bit4 OSR<2:1>: OSR设置（转换速率相关）；
000= 64；
001= 128；
010= 256；
011= 1024；
100= 4096；
101= 8192；
110= 16384；
111= 32768。

Bit3 CHSEL<3>: 通道N选择高位，N为1或者2；
1= 通道2；
0= 通道1。

Bit2~Bit0 PGA_SEL<2:0>: PGA增益；
000= 2；
001= 4；
010= 8；
011= 16；
100= 32；
101= 64；
110= 128；
111= 256.

12.5.2 Sigma-Delta ADC控制寄存器2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SDADCCON2	CHSEL2	CHSEL1	CHSEL0	LPWR	FADC	OSR_0	ENCHOPB	FCHOP_ADC
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	1	0	0	0	0

Bit7~Bit5 CHSEL<2:0>: 通道N选择低位，N由CHSEL<3>通道N选择高位决定；
000= 通道N；
001= 通道N 正负交换（系统斩波）；
010= 通道N的温度；
011= 通道N的内短；
100= 通道N 直通ADC 并关PGA；
101= 通道N 直通ADC 并关PGA；
110= 通道N的BG；
111= 通道N内短直通ADC。

Bit4 LPWR: 功耗选择；
0= 最低功耗；
1= 正常功耗。

Bit3 FADC: Sigma-Delta ADC系统时钟；
0= 328KHz；
1= 656KHz（推荐）。

Bit2 OSR_0: OSR LSB。

Bit1 ENCHOPB: ADC斩波使能；
0= 打开斩波功能（推荐）；
1= 关闭斩波功能。

Bit0 FCHOP_ADC: ADC 斩波分频控制；
0= 16 分频；
1= 32分频。

12.5.3 Sigma-Delta ADC控制寄存器3

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SDADCCON3	--	--	--	--	--	BYPASSLDO	OCP_DIS_LDO	--
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	0	0	1	0	0	0

Bit7~Bit4 --: 保留，须为1000。

Bit3 --: 保留，须为1。

Bit2 BYPASSLDO: LDO Bypass控制；

0= LDO输出；

1= Bypass LDO，输出VDD。

Bit1 OCP_DIS_LDO: LDO 输出过流保护使能控制；

0= 使能；

1= 关闭。

Bit0 --: 保留，须为0。

12.5.4 Sigma-Delta ADC控制寄存器4

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SDADCCON4	--	--	--	--	--	--	FCHOP_1	FCHOP_0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	0	0	0	0	0	1

Bit7~Bit4 --: 保留，须为1000。

Bit3-Bi2 --: 保留，须为00。

Bit1-Bit0 FCHOP<1:0>: PGA斩波分频控制；

00= 4 分频；

01= 8分频；

10= 16分频；

11= 32分频。

第13章 D/A转换器

13.1 D/A转换器的功能

D/A 转换器是将数字输入转换为模拟信号的 8 位分辨率的转换器，能控制模拟输出。

D/A 转换器有以下功能：

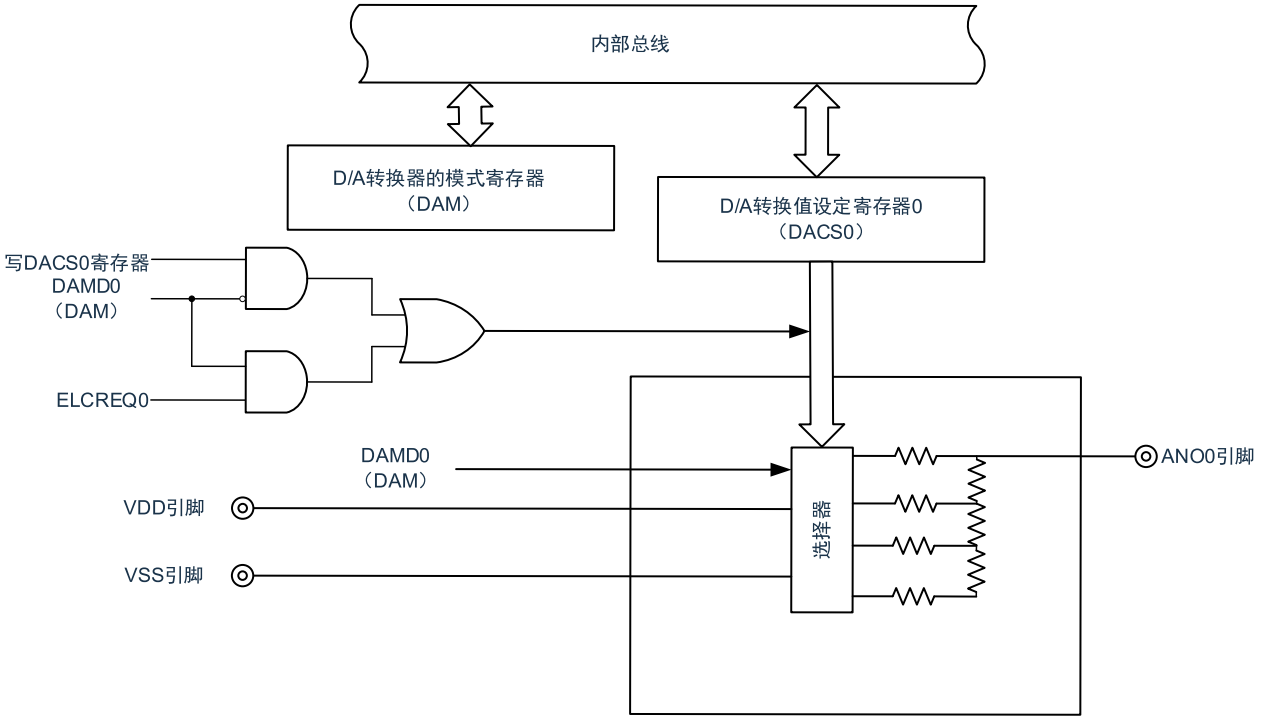
- 8 位分辨率
- R-2R 梯形方式
- 模拟输出电压
  - 8 位分辨率：V_{DD}*m8/256（m8：给 DACSi 寄存器设置的值）
- 运行模式
  - 通常模式
  - 实时输出模式

备注：i=0

13.2 D/A转换器的结构

D/A 转换器的框图如图 12-1 所示。

图 12-1：D/A 转换器的框图



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

13.3 控制D/A转换器的寄存器

通过以下寄存器控制 D/A 转换器。

- 外围允许寄存器 1（PER1）
- D/A 转换器的模式寄存器（DAM）
- D/A 转换值设置寄存器（DACS0）
- 事件输出目标选择寄存器 n（ELSELRn）、n=00~15

13.3.1 外围允许寄存器1（PER1）

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

要使用 D/A 转换器时，必须将 bit7（DACEN）置“1”。

通过 8 位存储器操作指令设置 PER1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-2：外围允许寄存器 1（PER1）的格式

地址：4002081AH复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
PER1	OPAEN	CMPEN	DACEN	ADCEN	0	OSDCEN	DMAEN	SPIHSEN

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

注意：要设置 D/A 转换器时，必须先将 DACEN 位置“1”。

当 DACEN 位为“0”时，忽视 D/A 转换器的控制寄存器的写操作，而且读取值都为初始值。

13.3.2 D/A转换器的模式寄存器（DAM）

这是控制 D/A 转换器运行的寄存器。

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

图 12-3：D/A 转换器的模式寄存器（DAM）的格式

地址：40045402H复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
DAM	0	0	0	DACE0	0	0	0	DAMD0

DACEi	D/A转换器的转换运行的控制
0	停止D/A转换。
1	允许D/A转换。

DAMDi	D/A转换器的运行模式的选择
0	通常运行模式
1	实时输出模式

备注：i=0

13.3.3 D/A转换值设置寄存器i（DACSi）（i=0）

这是设置在使用 D/A 转换器时输出到 ANO0 引脚和 ANO1 引脚的模拟电压值的寄存器。通过 8 位存储器操作指令设置 DACSi 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 12-4：D/A 转换值设置寄存器 i（DACSi）（i=0）的格式

地址：40045400H（DACS0）　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

备注：D/A 转换器的模拟输出电压（VANOi）如下：$VANOi=V_{DD}*(DACSi)/256$

当不使用 D/A 转换器时，为了减小不必要的消费电流，必须将 DACEi 位置“0”（禁止输出）并且将 DACSi 寄存器置“00H”，使 R-2R 的电阻没有电流流过。

13.3.4 事件输出目标选择寄存器n（ELSELRn）、n=00~15

当使用 D/A 转换器的实时输出模式时，以事件链接控制器的事件信号为启动触发，进行 D/A 转换。详细内容请参照“21.3.1 事件输出目标选择寄存器 n（ELSELRn）（n=00~15）”。

13.4 D/A转换器的运行

13.4.1 通常模式的运行

以 DACSi 寄存器的写操为启动触发，进行 D/A 转换。

其设置方法如下所示：

① 将PER1寄存器（外围允许寄存器1）的DACEN位置“1”，开始给D/A转换器提供输入时钟。

② 通过PMC寄存器（端口模式控制寄存器）将端口设置为模拟引脚。

③ 将DAM寄存器（D/A转换器的模式寄存器）的DAMDi位置“0”（通常模式）。

④ 给DACSi寄存器（D/A转换值设置寄存器i）设置ANOi引脚输出的模拟电压值。
以上①~④为初始设置。

⑤ 将DAM寄存器的DACEi位置“1”（允许D/A转换）。
开始 D/A 转换，在经过稳定时间后将④设置的模拟电压输出到 ANOi 引脚。

⑥ 此后，要进行D/A转换时，写DACSi寄存器。
在进行下一次 D/A 转换前，保持前一次 D/A 转换的结果。
如果将 DAM 寄存器的 DACEi 位置“0”（停止 D/A 转换），就停止 D/A 转换。

注意：

1. 即使将 DACEi 位的设置值进行“1”“0”“1”的设置，也在经过最后置“1”后的稳定时间之后将 DACSi 寄存器设置的模拟电压输出到 ANOi 引脚。
2. 如果在稳定时间内改写 DACSi 寄存器，就中止转换并且以改写的值重新开始转换。

备注：i=0

13.4.2 实时输出模式的运行

D/A 转换器的各通道以 EVENTC 的事件信号为启动触发，进行 D/A 转换。

其设置方法如下所示：

① 将PER1寄存器（外围允许寄存器1）的DACEN位置“1”，开始给D/A转换器提供输入时钟。

② 通过PMC寄存器（端口模式控制寄存器）将端口设置为模拟引脚。

③ 将DAM寄存器（D/A转换器的模式寄存器）的DAMDi位置“0”（通常模式）。

④ 给DACSi寄存器（D/A转换值设置寄存器i）设置ANOi引脚输出的模拟电压值。

⑤ 将DAM寄存器的DACEi位置“1”（允许D/A转换）。

⑥ 开始D/A转换，在经过稳定时间后将③设置的模拟电压输出到ANOi引脚。

⑦ 通过事件输出目标选择寄存器n（ELSELRn、n=00~15）设置用于实时输出模式的触发信号。

⑧ 将DAM寄存器的DAMDi位置“1”（实时输出模式）。

⑨ 开始事件发生源的运行。

以上①~⑧为初始设置

⑩ 此后，通过产生用于实时输出模式的触发信号，开始D/A转换，在经过稳定时间后将④设置的模拟电压输出到ANOi引脚。

必须在进行下一次 D/A 转换（产生用于实时输出模式的触发信号）前给 DACSi 寄存器设置 ANOi 引脚输出的模拟电压值。

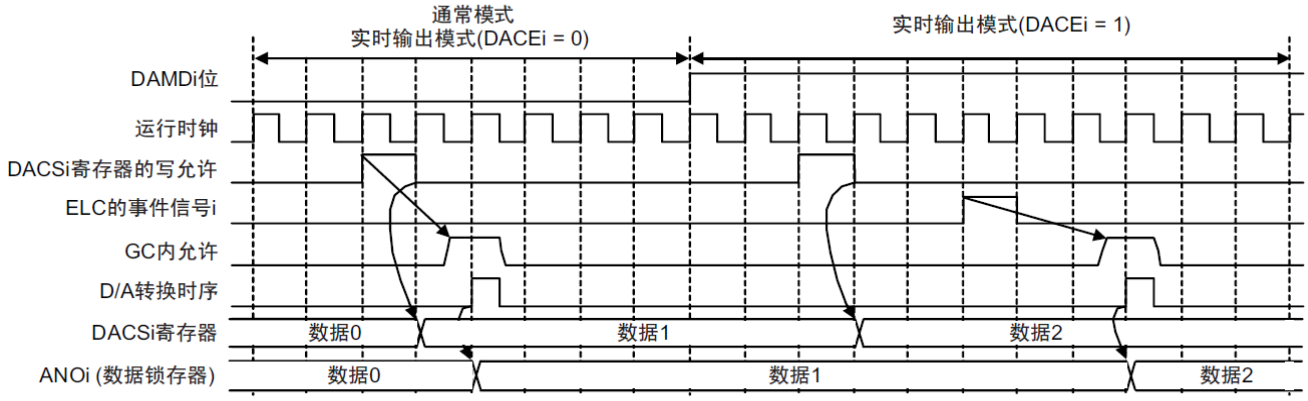
如果将 DAM 寄存器的 DACEi 位置“0”（停止 D/A 转换），就停止 D/A 转换。

注意：

1. 即使将 DACEi 位的设置值进行“1”“0”“1”的设置，也在经过最后置“1”后的稳定时间之后将 DACSi 寄存器设置的模拟电压输出到 ANOi 引脚。
2. 用于相同通道实时输出模式的触发信号的产生间隔必须大于稳定时间。如果在稳定时间内产生用于实时输出模式的触发信号，就中止 D/A 转换并且重新开始转换。
3. 用于相同通道实时输出模式的触发信号的产生间隔必须大于 3 个 F_{CLK} 时钟。如果以小于等于 3 个 F_{CLK} 时钟的间隔连续产生启动触发，就只在产生第 1 个触发时进行 D/A 转换。

13.4.3 D/A转换值的输出时序

D/A 转换值的输出时序如图 12-5 所示。



备注：i=0

- 通常运行模式和实时输出模式（不允许转换的情况）
  在写 DACSi 寄存器的 1 个周期后（运行时钟）写数据锁存器（从 ANOi 引脚输出）。
- 实时输出模式（允许转换的情况）
  在接受 EVENTC 事件信号后的 3 个周期后（运行时钟）写数据锁存器（从 ANOi 引脚输出）。

13.5 使用D/A转换器时的注意事项

使用 D/A 转换器时的注意事项如下所示。

(1) 当通过 PMC 寄存器（端口模式控制寄存器）将端口设置为模拟引脚时，和 ANO0 引脚复用的数字端口的输入/输出功能不工作。如果在通过 ADPC 寄存器将端口设置为模拟引脚的过程中读 P2 寄存器，就在输入模式中读取值为“0”，而在输出模式中读取值为 P2 的设置值。另外，即使设置为数字输出模式，也不将输出数据输出到引脚。

(2) 在睡眠模式和深度睡眠模式中，D/A 转换器继续运行。为了降低功耗，必须将 DACEi 位清“0”，并且在停止 D/A 转换后执行 WFI 指令。

备注：i=0

(3) 在停止实时输出模式时（包括变为通常模式的情况）需要按照以下步骤进行：

- 在停止触发输出源后至少等待 3 个时钟，然后将 DACEi 位和 DAMDi 位置“0”。
- 在将 DACEi 位和 DAMDi 位置“0”后将 PER1 寄存器的 DACEN 位置“0”（停止 DAC）

如果将 DACEN 位置“0”，就清除 DAC 内部的全部寄存器。

因此，在重新开始运行时，需要设置各 SFR。

(4) 当允许 D/A 转换时，不能对和 ANO0 引脚的模拟输入引脚进行 A/D 转换。

(5) 在实时输出模式中，必须在产生用于实时输出模式的触发信号前设置 DACSi 寄存器的值。

在触发信号有效期间，不能更改 DACSi 寄存器的设置值。

(6) 因为 D/A 转换器的输出阻抗高，所以无法从 ANO0 引脚和 ANO1 引脚获得电流。在负载的输入阻抗低的情况下，必须在负载和 ANO0 引脚、ANO1 引脚之间插入跟踪放大器。另外，必须尽量缩短跟踪放大器和负载之间的布线（因为输出阻抗高）。

如果布线太长，就必须在布线的周围进行接地图形等的处理。

(7) 如果要实时输出模式有效时进入深度睡眠模式，必须在进入深度睡眠模式前禁止 EVENTC 的事件链接

第14章 比较器

本产品内置 2 个比较器 CMP0 和 CMP1。

14.1 比较器的功能

比较器有以下功能：

- CMPn 的正端输入六选一，可选择外部端口，CMP0 可以选 OPA 的输出
- CMPn 的负端输入四选一，可选择外部端口、内部基准电压（1.45V）或者 8bit DAC 输出的电压。
- 比较器 0 和比较器 1 的比较结果可以通过引脚输出（VCOUT0，VCOUT1）。

表 14-1：比较器的功能概要

单元	功能
COMP0/COMP1	2 通道比较器（CMP0 和 CMP1）
	比较器负端可以选择的基准电压： CMP0 和 CMP1 的负端均可选择外部引脚输入（2 个）、内部基准电压（1.45V）或者 8bit DAC 的输出
	CMP0 的正端可选择外部引脚输入（6 个）或者 OPA 的输出
	CMP1 的正端可以选择外部引脚输入（6 个）
	正端输入电压＞负端输入电压时，输出高电平
	正端输入电压＜负端输入电压时，输出低电平
	数字滤波器的滤波宽度可选
	输出反转功能
	比较结果可以从引脚（VCOUT0，VCOUT1）输出
	能检测比较器输出的有效边沿并且产生中断信号
	与 Timer8 组合可输出 TIMER WINDOW
	支持比较器正迟滞，负迟滞，和双边迟滞，迟滞电压可选 20mV，40mV，60mV

14.2 比较器的结构

比较器 0 的框图如图 14-1 所示。比较器 1 的框图如图 14-2 所示。

图 14-1：比较器 0 的框图

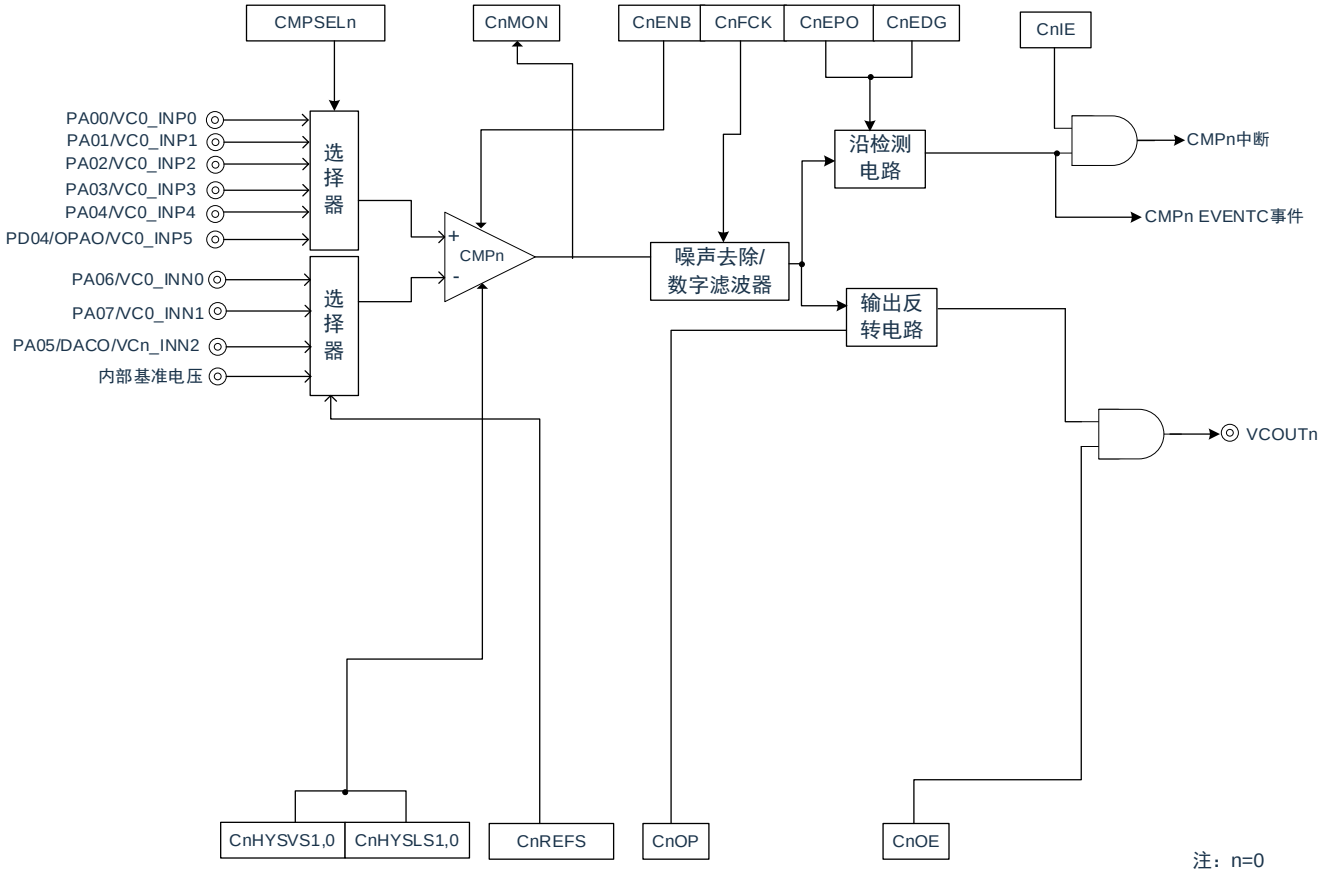
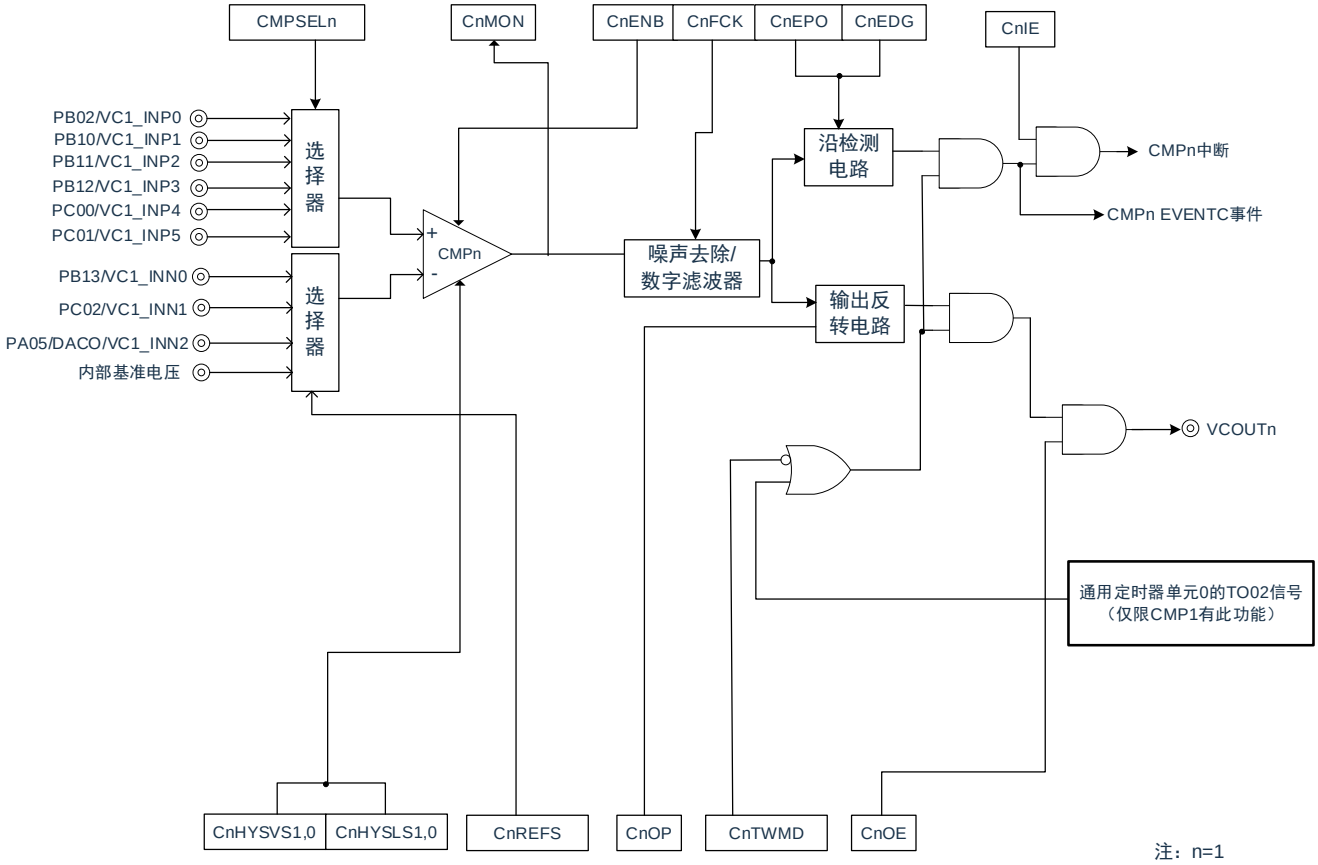


图 14-2：比较器 1 的框图



14.3 控制比较器的寄存器

控制比较器的寄存器如表 14-2 所示。比较器的寄存器地址如表 14-3：比较器的寄存器列表所示。

表 14-2：控制比较器的寄存器

寄存器名	符号
外围允许寄存器1	PER1
比较器模式设定寄存器	COMPMDR
比较器滤波控制寄存器	COMPFIR
比较器输出控制寄存器	COMPOCR
比较器0负端基准选择寄存器	C0REFS
比较器1负端基准选择寄存器	C1REFS
比较器0正端输入选择寄存器	CMPSEL0
比较器1正端输入选择寄存器	CMPSEL1
比较器0迟滞控制寄存器	CMP0HY
比较器1迟滞控制寄存器	CMP1HY
端口模式控制寄存器	PMCxx
端口输出控制寄存器	PMxx

表 14-3：比较器的寄存器列表

基地址	偏移地址	寄存器名称	读写属性	位宽	复位值
0x40045800	0x000	COMPMDR	R/W	8	00H
	0x001	COMPFIR	R/W	8	00H
	0x002	COMPOCR	R/W	8	00H
	0x008	C0REFS	R/W	8	00H
	0x009	C1REFS	R/W	8	00H
	0x00A	CMPSEL0	R/W	8	00H
	0x00B	CMPSEL1	R/W	8	00H
	0x00E	CMP0HY	R/W	8	00H
	0x00F	CMP1HY	R/W	8	00H

14.3.1 外围允许寄存器1（PER1）

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

要使用比较器时，必须将 bit6（CMPEN）置“1”。

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

图 14-3：外围允许寄存器 1（PER1）的格式

地址：0x4002081A　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER1	XX	XX	CMPEN	XX	XX	XX	XX	XX

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

注意：要设定比较器时，必须先将 CMPEN 位置“1”。

当 CMPEN 位为“0”时，对比较器的控制寄存器进行的写操作无效，并且读取值都为初始值（端口寄存器（PMCxx，PMxx）除外）。

14.3.2 比较器模式设定寄存器（COMPMDR）

COMPMDR 寄存器是设定比较器动作许可/禁止以及检测比较器输出的寄存器。CnENB 位在比较器输出许可(COMPOCR 寄存器的 CnOE 位置“1”)时，禁止设置为“0”。以下几种情况，禁止将 CnENB 位置“1”(n=0,1)：

- CMPn 选择 DAC 的输出作为基准电压，而 DAC 动作停止时
- CMP0 的输入选择 OPA 的输出，而 OPA 动作停止时

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

图 14-4：比较器模式设定寄存器（COMPMDR）的格式

符号	7	6	5	4	3	2	1	0
COMPMDR	C1MON	0	0	C1ENB	C0MON	0	0	C0ENB

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

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

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

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

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

注 2：忽视此位的写入值。

14.3.3 比较器滤波控制寄存器（COMPFIR）

COMPFIR 寄存器是数字滤波器的控制寄存器。通过 8 位存储器操作指令设定 COMPFIR 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 14-5：比较器滤波控制寄存器（COMPFIR）的格式

符号	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

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

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

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

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

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

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

注 1：如果更改 C1FCK1~C1FCK0 位、C1EPO 位和 C1EDG 位，就可能产生比较器 1 的中断请求和向 EVENTC 输出的事件信号。必须在联动控制器不链接比较器 1 的输出时更改这些位。另外，在更改后必须将中断请求标志寄存器的 IF 清“0”。如果将 C1FCK1~C1FCK0 位从“00B”（比较器 1 无滤波器）改为其他值（比较器 1 有滤波器），就必须经过 4 拍采样，等滤波器的输出更新后，再使用比较器 1 的中断请求或者向 EVENTC 输出的事件信号。

注 2：如果更改 C0FCK1~C0FCK0 位、C0EPO 位和 C0EDG 位，就可能产生比较器 0 的中断请求和向 EVENTC 输出的事件信号。必须在联动控制器不链接比较器 0 的输出时更改这些位。另外，在更改后必须将中断请求标志寄存器的 IF 清“0”。如果将 C0FCK1~C0FCK0 位从“00B”（比较器 0 无滤波器）改为其他值（比较器 0 有滤波器），就必须经过 4 拍采样，等滤波器的输出更新后，再使用比较器 0 的中断请求或者向 EVENTC 输出的事件信号。

14.3.4 比较器输出控制寄存器（COMPOCR）

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

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

- 比较器动作停止(COMPMDR寄存器的CnENB位为“0”)时
- CMPn选择DAC的输出作为基准电压，而DAC动作停止时
- CMP0的输入选择OPA的输出，而OPA动作停止时

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

图 14-6：比较器输出控制寄存器（COMPOCR）的格式

符号	7	6	5	4	3	2	1	0
COMPOCR	C1OTWMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

C1OTWMD	比较器1的TIMER WINDOW输出模式控制位[注1]
0	比较器1普通输出模式(由C1OE控制)
1	比较器1TIMER WINDOW输出模式(由TO02和C1OE共同控制)

C1OP	VCOUT1输出极性的选择
0	VCOUT1是比较器1的输出。
1	VCOUT1是比较器1的反相输出。

C1OE	VCOUT1引脚输出的允许[注2]
0	禁止VCOUT1输出到引脚。
1	允许VCOUT1输出到引脚。

C1IE	比较器1中断请求的允许[注3]
0	禁止比较器1的中断请求。
1	允许比较器1的中断请求。

C0OP	VCOUT0输出极性的选择
0	VCOUT0是比较器0的输出。
1	VCOUT0是比较器0的反相输出。

C0OE	VCOUT0引脚输出的允许[注4]
0	禁止VCOUT0输出到引脚。
1	允许VCOUT0输出到引脚[注4]，[注8]。

C0IE	比较器0中断请求的允许[注5]
0	禁止比较器0的中断请求。
1	允许比较器0的中断请求。

注 1：比较器 1 使用 TIMER WINDOW 模式时，必须将寄存器 COMPFIR 的 bit7(C1EDG)置“1”。C1OE 和 C1OTWMD 位不能同时设置，要先设定 C1OTWMD 位之后，再将 C1OE 位置为“1”。

注 2：比较器 1 的结果输出到引脚时，必须将该引脚的 Pxx，PMxx，PMCxx 置 0。

注 3：如果将 C1IE 从“0”（禁止中断请求）改为“1”（允许中断请求），中断请求标志寄存器的 IF 就可能变为“1”（有中断请求），因此必须在将中断请求标志寄存器的 IF 清“0”后使用中断。

注 4：比较器 0 的结果输出到引脚时，必须将该引脚的 Pxx，PMxx，PMCxx 置 0。

注 5：如果将 C0IE 从“0”（禁止中断请求）改为“1”（允许中断请求），中断请求标志寄存器的 IF 就可能变为“1”（有中断请求），因此必须在将中断请求标志寄存器的 IF 清“0”后使用中断。

14.3.5 比较器负端基准选择寄存器（CnREFS）

CnREFS 寄存器是设置比较器的负端基准的寄存器。在比较器停止动作(CnENB=0)时，可改写 CnREFS 寄存器。(n=0,1)

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

图 14-7：比较器负端基准选择寄存器（CnREFS）的格式

符号	7	6	5	4	3	2	1	0
CnREFS	0	0	0	0	0	0	CnREFS[1:0]	

CnREFS	比较器负端基准选择
2’b00	外部引脚 VCn_INN0
2’b01	外部引脚 VCn_INN1
2’b10	DAC0 输出/外部引脚 VCn_INN2
2’b11	1.45V 基准电压

注 1：如果 DAC 不动作，也可以从外部引脚输入比较器负端基准。

比较器负端 VCn_INN0~2 的引脚复用功能的配置方法，请参考“第 2 章 端口功能”。

14.3.6 比较器正端输入选择寄存器（CMPSELn）

CMPSELn 寄存器是比较器的正端输入选择寄存器。在比较器停止动作(CnENB=0)时，可改写 CMPSELn 寄存器。(n=0,1)

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

图 14-8：比较器的正端输入选择寄存器（CMPSELn）的格式

符号	7	6	5	4	3	2	1	0
CMPSELn	0	0	0	0	0	CnVIPSEL		

C0INPSEL	CMP0 正端输入选择
3’h0	外部引脚 VC0_INP0
3’h1	外部引脚 VC0_INP1
3’h2	外部引脚 VC0_INP2
3’h3	外部引脚 VC0_INP3
3’h4	外部引脚 VC0_INP4
3’h5	OPA 输出（OPAO）/外部引脚 VC0_INP5*注 1
其他	禁止设定

注 1：如果 OPA 不动作，也可以从外部引脚输入比较器 VC0_INP5

C1INPSEL	CMP1 正端输入选择
3’h0	外部引脚 VC1_INP0
3’h1	外部引脚 VC1_INP1
3’h2	外部引脚 VC1_INP2
3’h3	外部引脚 VC1_INP3
3’h4	外部引脚 VC1_INP4
3’h5	外部引脚 VC1_INP5
其他	禁止设定

比较器正端 VCn_INP0~5 的引脚复用功能的配置方法，请参考“第 2 章 端口功能”。

14.3.7 比较器的迟滞控制寄存器（CMPnHY）

CMPnHY 寄存器是比较器的迟滞功能控制寄存器。在比较器停止动作(CnENB=0)时，可以改写 CMPnHY 寄存器。通过 8 位存储器操作指令设定 CMPnHY 寄存器。在产生复位信号后，此寄存器的值变为“00H”。（n=0，1）

图 14-9：比较器的迟滞控制寄存器（CMPnHY）的格式

符号	7	6	5	4	3	2	1	0
CMPnHY	0	0	CnHYSVS1	CnHYSVS0	0	0	CnHYSLS1	CnHYSLS0

CnHYSVS1	CnHYSVS0	比较器n的迟滞电压选择位
0	0	无迟滞
0	1	20mV
1	0	40mV
1	1	60mV

CnHYSLS1	CnHYSLS0	比较器n的迟滞方式选择位
0	0	无迟滞
0	1	正迟滞
1	0	负迟滞
1	1	双边迟滞

14.4 运行说明

比较器 0 和比较器 1 能各自独立运行。CMP0 和 OPA 可以组合起来联动。比较器的独立运行和联动的设定步骤如表 14-4 所示。

表 14-4：比较器相关寄存器的设定步骤

步骤	寄存器	位	设定值
1	OPACTL	INPSEL，MODE_OPA,MODE_RPD	OPA模式设定[注3]
2	OPADAC	OPA_DAO[4:0],OPA_DAP	OPA内置5bitDAC的控制[注3]
3	OPACTL	OPAEN	1（OPA允许运行）[注3]
4	等待OPA的稳定时间（最小1us）		
5	DAM	DAMD	8bit DAC动作模式选择[注4]
6	DACS	DACS[7:0]	8bit DAC电压挡位选择[注4]
7	DAM	DACE	1（DAC允许运行）[注4]
8	等待DAC的稳定时间（最小6us）		
9	CMPSELn	CnVIPSEL/CnVIPSEL	比较器n正端输入选择
10	CnREFS	CnREFS	比较器n负端输入选择
11	将VC_INP、VC_INN引脚（输入），OPA(输入)[注3]设置为模拟输入功能。 PMCxx VCIN0引脚,VCn和IVREFn引脚的功能选择将PMCxx位置“1”（模拟输入）。 将PMxx位置“1”（输入模式）。		
12	COMPMDR	CnENB	1（允许运行）
13	等待比较器的稳定时间（最小3us）		
14	COMPFIR	CnFCK	使用或者不使用数字滤波器，选择采样时钟。
		CnEOP、CnEDG	选择中断请求的边沿检测条件（上升沿、下降沿或者双边沿）。
15	COMPOCR	CnOP、CnOE	设定VCOUTn的输出（选择极性，设定允许或者禁止输出）。请参照“14.4.4比较器n的输出（n=0、1）”。
		CnIE	设定允许或者禁止中断请求的输出。 请参照“14.4.2比较器n中断（n=0、1）
		C1OTWMD	设置比较器1的TIMER WINDOW输出许可/禁止
16	MKxx[注1]	MKL	使用中断时：选择屏蔽中断。
17	IFxx[注1]	IFL	使用中断时：0（无中断请求：初始化）[注2]

注 1：MKxx，IFxx 是比较器的中断控制寄存器，详细内容请参考“第 23 章 中断功能”。

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

注 3：比较器 0 和 OPA 联动时，按照需要的模式设定。

注 4：使用 8bit DAC 的输出作为比较器 n 的负端基准电压时，需要设定。

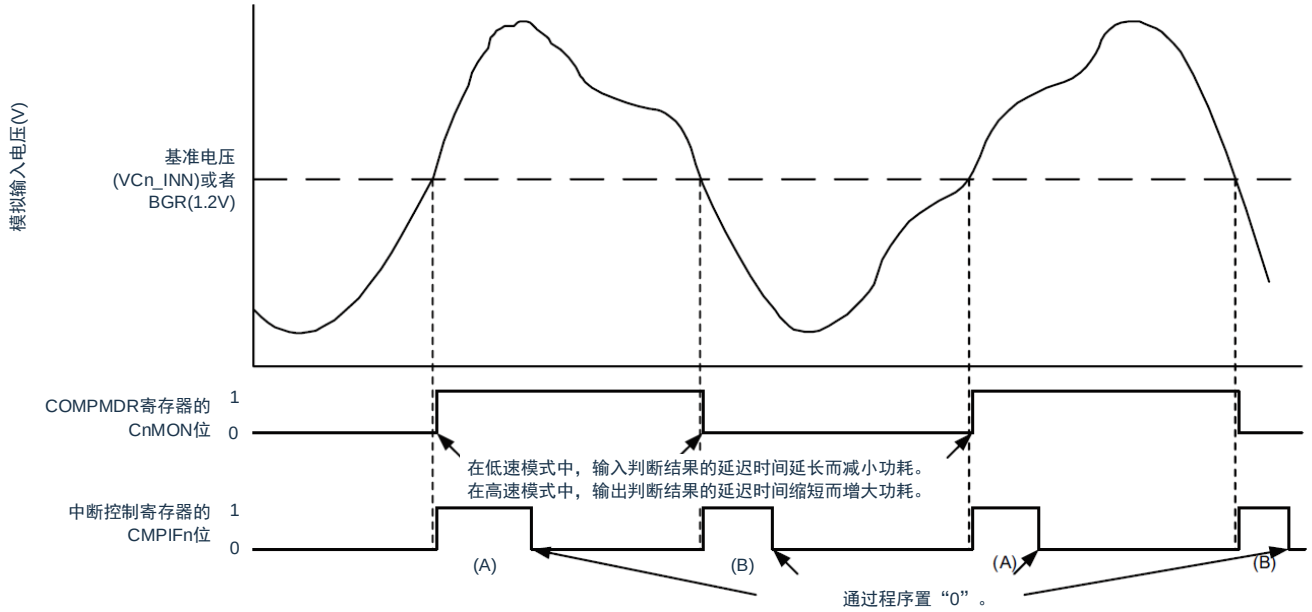
备注：n=0、1

比较器 n（n=0、1）的运行例子如图 14-10 所示。在基本模式中，当模拟输入电压高于基准输入电压时，COMPMDR 寄存器的 CnMON 位为“1”；当模拟输入电压低于基准输入电压时，CnMON 位为“0”。

要使用比较器 n 中断时，必须将 COMPOCR 寄存器的 CnIE 位置“1”（允许中断请求）。此时，如果比较结果发生变化，就产生比较器 n 的中断请求。有关中断请求的详细内容，请参照“14.4.2 比较器 n 中断（n=0、1）”。

图 14-10：比较器 n（n=0、1）的运行例子（基本模式）

- 基本模式的运行例子



注意：上图是 COMPFIR 寄存器的 CnFCK1~CnFCK0 位为“00B”（无滤波器）并且 CnEDG 位为“1”（双边沿）的情况（CnEDG 位为“0”并且 CnEPO 位为“0”（上升沿）时的 CMPIFn 只限于(A)的变化，CnEDG 位为“0”并且 CnEPO 位为“1”（下降沿）时的 CMPIFn 只限于(B)的变化）。

14.4.1 比较器n的数字滤波器（n=0、1）

比较器 n 内置数字滤波器，能通过 COMPFIR 寄存器的 CnFCK1~CnFCK0 位选择采样时钟。按各采样时钟对比较器 n 的输出信号进行采样，在电平 3 次都相同后的下一个采样时钟，数字滤波器输出此采样值。

图 14-11 是比较器 n 的数字滤波器的结果，图 14-12 是比较器 n 的数字比较器 n（n=0、1）的数字滤波器和中断运行例子。

图 14-11：比较器 n（n=0、1）的数字滤波器和沿检出结构

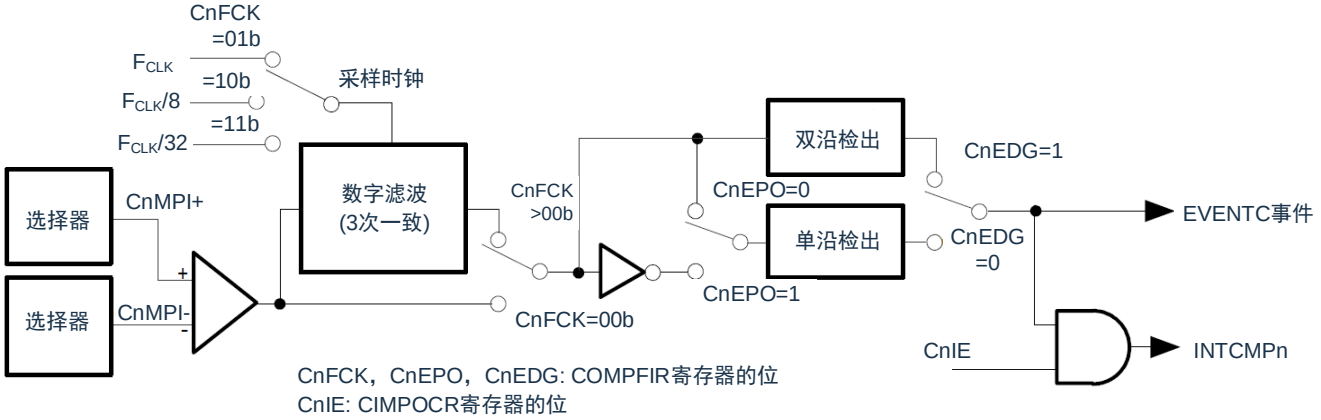
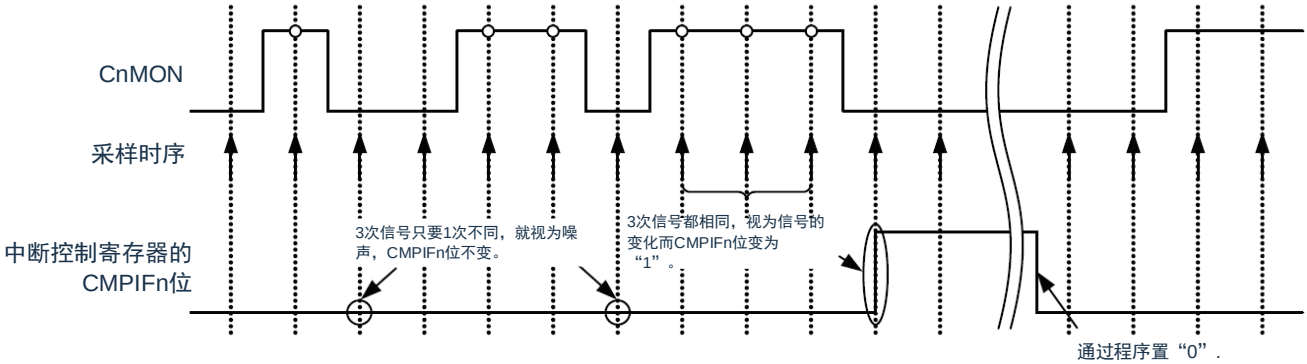


图 14-12：比较器 n（n=0、1）的数字滤波器和中断运行例子



注意：上图是 COMPFIR 寄存器的 CnFCK1~CnFCK0 位为“01B”、“10B”或者“11B”（有数字滤波器）时的运行例子。

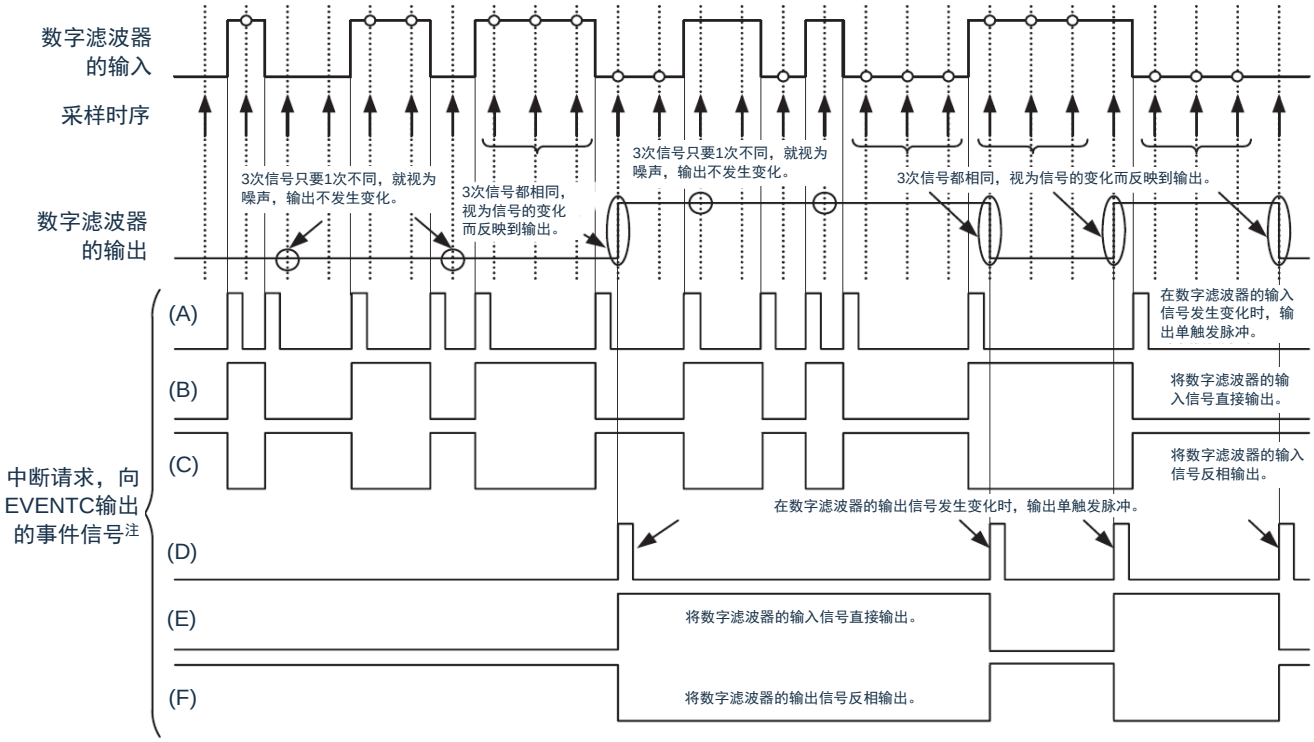
14.4.2 比较器n中断（n=0、1）

比较器可产生比较器 0 和比较器 1 共 2 个中断请求。要使用比较器 n 中断时，必须将 COMPOCR 寄存器的 CnIE 位置“1”（允许中断请求的输出）。通过 COMPFIR 寄存器设定中断请求的产生条件，还能给比较器的输出附加数字滤波器。数字滤波器可选择 3 种采样时钟。有关寄存器的设定和产生中断请求的对应，请参照“14.3.3 比较器滤波控制寄存器（COMPFIR）”和“14.3.4 比较器输出控制寄存器（COMPOCR）”。

14.4.3 向联动控制器（EVENTC）输出的事件信号

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

图 14-13：数字滤波器、中断请求和向 EVENTC 输出事件信号的运行



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

(A)、(B)、(C)的波形是 COMPFIR 寄存器的 CnFCK 位（n=0、1）为“00B”（无数字滤波器）的情况，(D)、I、(F)的波形是 COMPFIR 寄存器的 CnFCK 位（n=0、1）为“01B”、“10B”或者“11B”（有数字滤波器）的情况。(A)、(D)是 CnEDG 位为“1”（双边沿）的情况，(B)、I 是 CnEDG 位为“0”并且 CnEPO 位为“0”（上升沿）的情况，(C)、(F)是 CnEDG 位为“0”并且 CnEPO 位为“1”（下降沿）的情况。

14.4.4 比较器n的输出（n=0、1）

能通过 COMPOCR 寄存器的 CnOE 位设定比较器的比较结果是否输出到外部引脚，还能通过 COMPOCR 寄存器的 CnOP 位设定输出极性（正相输出或者反相输出）。有关寄存器的设定和比较器输出的对应，请参照“14.3.4 比较器输出控制寄存器（COMPOCR）”。

要将比较器的比较结果输出到 VCOUTn 引脚时，必须按照以下的步骤设定端口（在复位后，端口默认为输入状态），详细的设置方法请参考“第 2 章 端口功能”：

① 设定比较器的模式（“表14-4：比较器相关寄存器的设定步骤”的步骤2~5）。

② 设定比较器的VCOUTn输出（设定COMPOCR寄存器，选择极性并且允许输出）。

③ 设置PxxCFG寄存器，选择VCOUTn的输出兼用到哪个端口。

④ 将VCOUTn的输出引脚对应的端口模式控制寄存器PMCxx的位置“0”。

⑤ 将VCOUTn的输出引脚对应的端口输出锁存寄存器Pxx的位置“0”。

⑥ 将VCOUTn的输出引脚对应的端口方向寄存器PMxx设定为输出（从引脚开始输出）。

14.4.5 比较器时钟的停止和提供

在通过设定外围允许寄存器 1（PER1）来停止比较器时钟的情况下，必须按照以下步骤进行设定：

① 将COMPMDR寄存器的CnENB位置“0”（停止比较器的运行）。

② 将中断请求标志寄存器的IF位置“0”（清除比较器停止运行前不需要的中断）。

③ 将PER1寄存器的CMPEN位置“0”。

如果通过设定 PER1 寄存器来停止时钟，比较器的内部寄存器就全部被初始化，因此在要再次使用比较器时，必须按照表 14-4 的步骤设定寄存器。

注意：

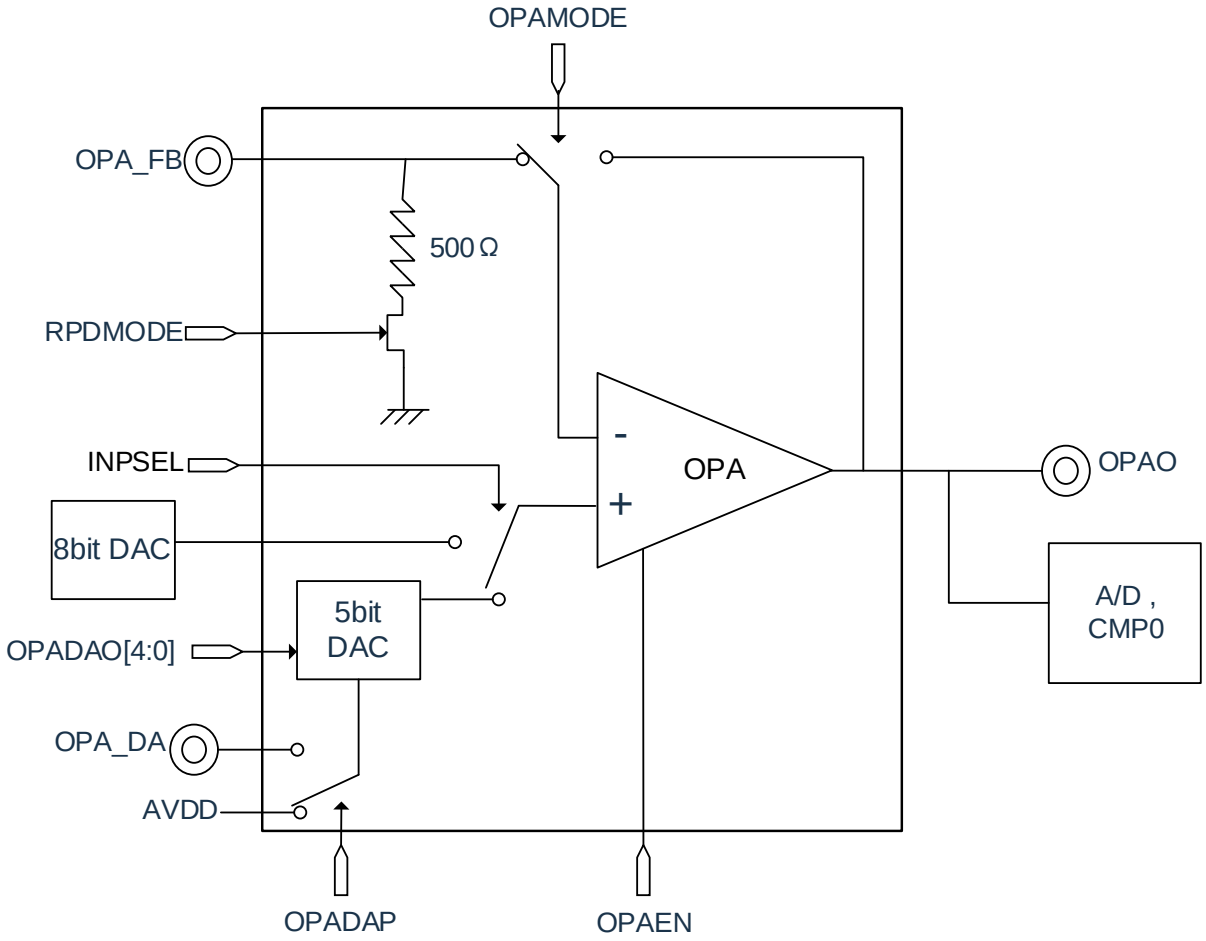
1. 当比较器负端基准电压选择内部参考电压（1.45V）时，A/D 转换器不能转换温度传感器电压。
2. 在下列任一情况下可启动 DMA 传输，并且在传送结束后产生中断。使能 DMA 之前必须根据需要先确认比较器的监视标志（CnMON），再设置允许 DMA 启动。
   - 设定为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的上升沿产生中断请求（CnEPO=0）而且 VCn_INP＞VCn_INN（或者内部基准电压 1.45V）。
   - 设定为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的下降沿产生中断请求（CnEPO=1）而且 VCn_INP＜VCn_INN（或者内部基准电压 1.45V）。
     （n=0、1）

第15章 运算放大器（OPA）

15.1 运算放大器的功能

本产品内置一路运算放大器（OPA），具有如下功能：

- 支持轨到轨
- OPA的工作模式支持恒流模式和Buffer模式，Buffer模式可用于测试OPA失调
- OPA负反馈端支持使用内部恒流电阻或外部引脚OPA_FB
- OPA的输入可选运放模块内置的5bit DAC或者本产品自带的8bit DAC
- OPA的输出可用于A/D转换器的模拟输入或者比较器0（CMP0）正端的模拟输入



15.2 运算放大器的寄存器

表 15-1：控制运算放大器的寄存器

寄存器名称	符号
外围允许寄存器 1	PER1
运算放大器控制寄存器	OPACTL
运算放大器数模控制寄存器	OPADAC
端口模式控制寄存器	PMCxx
端口模式寄存器	PMxx

15.2.1 外围允许寄存器1(PER1)

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

要使用运算放大器时，必须将此 PER1 寄存器的 bit7（OPAEN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER1 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 15-1：外围允许寄存器 1（PER1）

地址：0x4002081A　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PER1	OPAEN	XX 0	XX	XX	XX	XX	XX	XX

OPAEN	运算放大器的输入时钟的控制
0	停止提供输入时钟。 运算放大器的寄存器不可写 运算放大器处于复位状态
1	提供输入时钟。 运算放大器的寄存器可读写

注意：

1. 在配置比较器或运算放大器的寄存器之前，确认 OPAEN 的 bit 位先要置 1.
2. 如果 OPAEN=0,对运算放大器的控制寄存器进行写操作是无效的,并且所有的读出值为默认值。（端口模式寄存器（PMXX）和端口寄存器 PXX 除外）

15.2.2 运算放大器控制寄存器（OPACTL）

OPACTL 寄存器用于控制运算放大器开始工作，停止工作和选择工作模式。

可通过 8 位存储器操作指令设定 OPACTL 寄存器。在产生复位信号后，此寄存器复位值为 00H。

图 15-2：OPA 控制寄存器（OPACTL）的格式

地址：0x40045C00　　复位值：00H　　R/W

	7	6	5	4	3	2	1	0
OPACTL	OPAEN	0	RPDMODE	OPAMODE	0	INPSEL	0	0

OPAEN	放大器 OPA 使能信号
0	放大器停止动作
1	允许放大器动作

RPDMODE	反馈端选择
0	不支持内部恒流电阻
1	支持内部恒流电阻

OPAMODE	OPA 工作模式选择
0	恒流模式
1	Buffer 模式

INPSEL	OPA 正端输入选择
0	选择 5bit-DAC 的输出
1	选择 8bit-DAC 的输出

15.2.3 运算放大器数模控制寄存器（OPADAC）

OPADAC 寄存器用于控制运算放大器内置的 5bit DAC。当 OPA 的输入选择内置 5bit DAC 时，可以通过设置该寄存器，控制 5bit DAC 的电源选择和 DAC 输出的电压的挡位。

可通过 8 位存储器操作指令设定 OPADAC 寄存器。在产生复位信号后，此寄存器复位值为 00H。

图 15-3：OPA 数模控制寄存器（OPADAC）的格式

地址：0x40045C04　　复位值：00H　　R/W

	7	6	5	4	3	2	1	0
OPADAC	OPADAP	0	0	OPADAO[4:0]注 1				

OPADAP	OPA 内置 5bit DAC 电源选择信号
0	选择 OPA 电源作为 5bit DAC 的电源
1	选择外部引脚 OPA_DA 作为 5bit DAC 的电源

注 1：5bit DAC 转换器的模拟输出电压：$Vout = \frac{19+3*n}{560}*Vref$，其中 n=0,1,2…31

第16章 通用串行通信单元

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

各产品支持的通道功能分配如下：

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

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

在单元 1 的通道 0 和通道 1 使用 UART1 时，不能使用 SSPI10 和 SSPI11 和 IIC10、IIC11。

在单元 2 的通道 0 和通道 1 使用 UART2 时，不能使用 SSPI20 和 SSPI21 和 IIC20、IIC21。

16.1 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。

16.1.1 3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）

与主控设备输出的串行时钟（SCLK）同步进行数据的发送和接收。

这是使用 1 条串行时钟（SCLK）、1 条发送串行数据（SDO）和 1 条接收串行数据（SDI）共 3 条通信线进行通信的时钟同步通信功能。

有关具体的设定例子，请参照“16.5 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的运行”。

[数据的发送和接收]

- 7 位~16 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

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

  主控通信：最大值 $F_{CLK}/2$

  从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

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

16.1.2 UART（UART0~UART2）

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

有关具体的设定例子，请参照“16.7 UART（UART0~UART2）通信的运行”。

[数据的发送和接收]

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

[中断功能]

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

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

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

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

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。有关具体的设定例子，请参照“16.9 简易 I2C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收（只限于单主控的主控功能）
- ACK 输出功能[注]、ACK 检测功能
- 8 位数据长度（在发送地址时，用高 7 位指定地址，用最低位进行 R/W 控制）
- 开始条件和停止条件的手动产生

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误、溢出错误

※[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注：在接收最后的数据时，如果给 SOEmn 位（串行输出允许寄存器 m（SOEm））写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“16.9.3(2) 处理流程”。

备注：在使用全功能的 I²C 总线时，请参照“第 18 章 串行接口 IICA”。

16.2 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 16-1：通用串行通信单元的结构

项目	结构
移位寄存器	16位
缓冲寄存器	串行数据寄存器mn（SDRmn）注
串行时钟输入/输出	SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21引脚（用于3线串行I/O）、 SCL00、SCL01、SCL10、SCL11、SCL20、SCL21引脚（用于简易I^2C）
串行数据输入	SDI00、SDI01、SDI10、SDI11、SDI20、SDI21引脚（用于3线串行I/O）、 RxD0、RxD1、RxD2引脚（用于UART）
串行数据输出	SDO00、SDO01、SDO10、SDO11、SDO20、SDO21引脚（用于3线串行I/O）、 TxD0、TxD1、TxD2引脚（用于UART）
串行数据输入/输出	SDA00、SDA01、SDA10、SDA11、SDA20、SDA21引脚（用于简易I^2C）
从属选择输入	SS00引脚（用于从属选择输入功能）
控制寄存器	＜单元设定部的寄存器＞ • 外围允许寄存器0（PER0） • 串行时钟选择寄存器m（SPSm） • 串行通道允许状态寄存m（SEm） • 串行通道开始寄存器m（SSm） • 串行通道停止寄存器m（STm） • 串行输出允许寄存器m（SOEm） • 串行输出寄存器m（SOm） • 串行输出电平寄存器m（SOLm） • 串行待机控制寄存器m（SSCm） • 从属选择功能启用寄存器m（SSEm） • 输入切换控制寄存器（ISC） • 噪声滤波器允许寄存器0（NFEN0）
	＜各通道部的寄存器＞ • 串行数据寄存器mn（SDRmn） • 串行模式寄存器mn（SMRmn） • 串行通信运行设定寄存器mn（SCRmn） • 串行状态寄存器mn（SSRmn） • 串行标志清除触发寄存器mn（SIRmn）
	• 端口复用功能配置寄存器（PxxCFG） • 端口输出模式寄存器（POMxx） • 端口模式寄存器（PMxx） • 端口寄存器（Pxx）

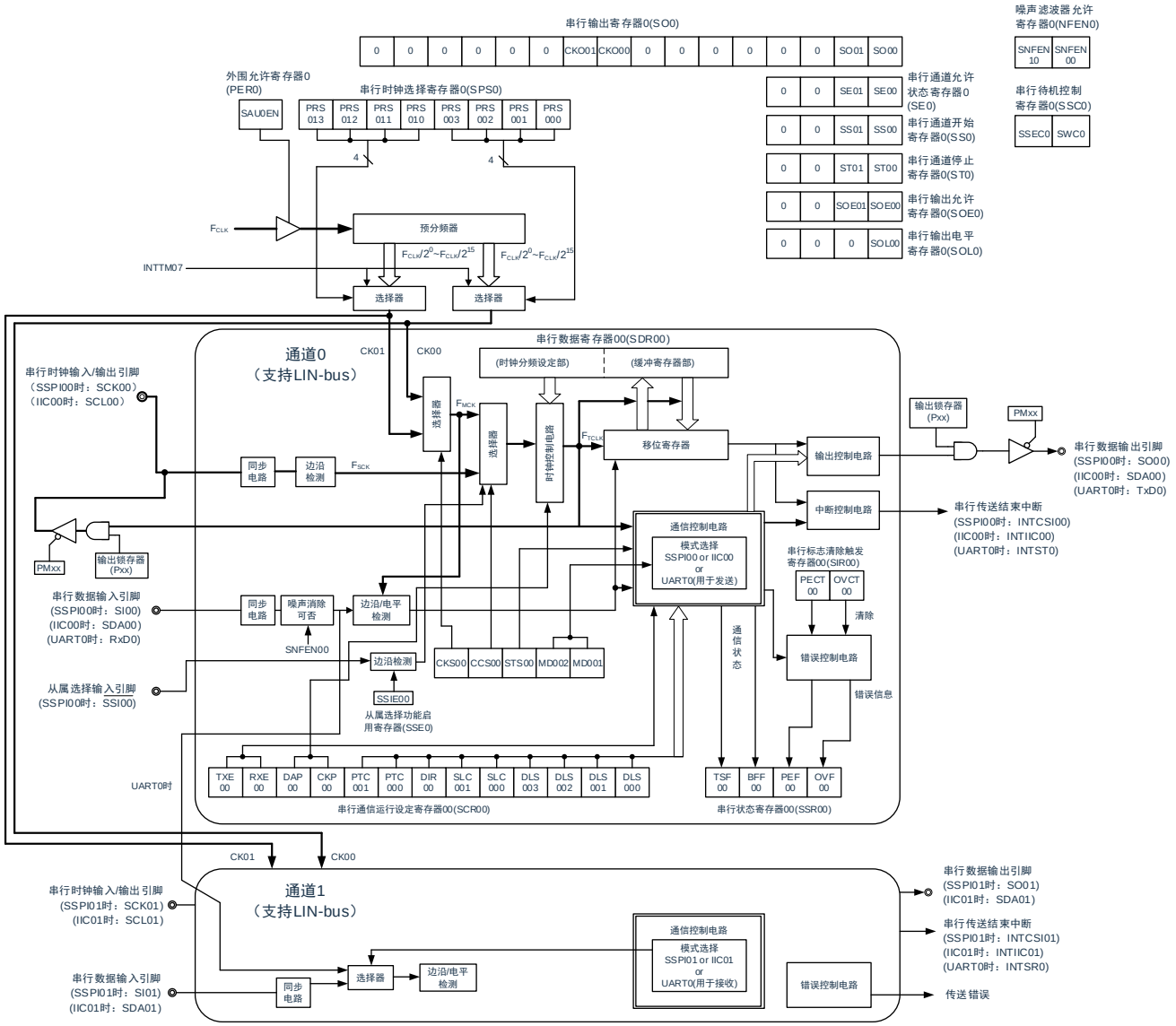
注：SEmn=1 的动作期间。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

q：UART 号（q=0~2）r：IIC 号（r=00、01、10、11、20、21）

通用串行通信单元的框图如图 15-1 所示。（以单元 0 为例）

图 16-1：通用串行通信单元 0 的框图



注：单元 0，1，2 的结构相同

16.2.1 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）。

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

16.2.2 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 bit3~bit0（DLSmn3~DLSmn0）的设定，保存到 SDRmn 寄存器的数据如下所示：

- 7 位数据长度（保存在 SDRmn 寄存器的 bit0~6）
- 8 位数据长度（保存在 SDRmn 寄存器的 bit0~7）
- 16 位数据长度（保存在 SDRmn 寄存器的 bit0~15）

能以 16 位为单位读写 SDRmn 寄存器。

SEmn=1 时，SDRmn 寄存器的低 8 位可以作为 SDRmnL 以 8 位为单位进行读写[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） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）
q：UART 号（q=0~2） r：IIC 号（r=00、01、10、11、20、21）

图 16-2：串行数据寄存器 mn（SDRmn）的格式

复位后：0000H　　R/W

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

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

备注：

1. 有关 SDRmn 寄存器的高 7 位的功能，请参照“16.3 控制通用串行通信单元的寄存器”。
2. m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3 控制通用串行通信单元的寄存器

控制通用串行通信单元的寄存器如下所示：

- 外围允许寄存器 0（PER0）
- 串行时钟选择寄存器 m（SPSm）
- 串行模式寄存器 mn（SMRmn）
- 串行通信运行设定寄存器 mn（SCRmn）
- 串行数据寄存器 mn（SDRmn）
- 串行标志清除触发寄存器 mn（SIRmn）
- 串行状态寄存器 mn（SSRmn）
- 串行通道开始寄存器 m（SSm）
- 串行通道停止寄存器 m（STm）
- 串行通道允许状态寄存器 m（SEm）
- 串行输出允许寄存器 m（SOEm）
- 串行输出电平寄存器 m（SOLm）
- 串行输出寄存器 m（SOm）
- 串行待机控制寄存器 m（SSCm）
- 从属选择功能启用寄存器 m（SSEm）
- 输入切换控制寄存器（ISC）
- 噪声滤波器允许寄存器 0（NFEN0）
- 端口复用功能配置寄存器（PxxCFG）
- 端口输出模式寄存器（POMx）
- 端口模式寄存器（PMx）
- 端口寄存器（Px）

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）

串行通信单元寄存器列表

单元 0 寄存器基地址：0x40046000

单元 1 寄存器基地址：0x40046400

单元 2 寄存器基地址：0x40046800

偏移地址	寄存器名称	R/W	复位值
0x000	SSRm0	R	0000H
0x002	SSRm1	R	0000H
0x004	SIRm0	R/W	0000H
0x006	SIRm1	R/W	0000H
0x008	SMRm0	R/W	0020H
0x00A	SMRm1	R/W	0020H
0x00C	SCRm0	R/W	0087H
0x00E	SCRm1	R/W	0087H
0x010	SEm	R	0000H
0x012	SSm	R/W	0000H
0x014	STm	R/W	0000H
0x016	SPSm	R/W	0000H
0x018	SOm	R/W	0303H
0x01A	SOEm	R/W	0000H
0x020	SOLm	R/W	0000H
0x022	SSEm	R/W	0000H
0x024	SSCm	R/W	0000H
0x040	SDRm0	R/W	0000H
0x040	SIOm0	R/W	00H
0x040	TXDm	R/W	00H
0x042	SDRm1	R/W	0000H
0x042	RXDm	R/W	00H
0x042	SIOm1	R/W	00H

备注：单元号 m=0，1，2

16.3.1 外围允许寄存器0（PER0）

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

要使用通用串行通信单元 0 时，必须将 bit2（SCI0EN）置“1”。

要使用通用串行通信单元 1 时，必须将 bit3（SCI1EN）置“1”。

要使用通用串行通信单元 2 时，必须将 bit4（SCI2EN）置“1”。

通过 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，PER0 寄存器的值变为“00H”。

图 16-3：外围允许寄存器 0（PER0）的格式

地址：40020420H　　复位后：00000000H　　R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IICA0EN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

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

注意：要设定通用串行通信单元 m 时，必须先在 SCImEN 位为“1”的状态下设定以下的寄存器。当 SCImEN 位为“0”时，忽视通用串行通信单元 m 的控制寄存器的写操作，而且读取值都为初始值（输入切换控制寄存器（ISC）、噪声滤波器允许寄存器 0（NFEN0）、端口复用功能配置寄存器（PxxCFG）、端口输出模式寄存器（POMx）、端口模式寄存器（PMx）、端口模式控制寄存器（PMCx）以及端口寄存器（Px）除外）。

- 串行时钟选择寄存器 m（SPSm）
- 串行模式寄存器 mn（SMRmn）
- 串行通信运行设定寄存器 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）

16.3.2 串行时钟选择寄存器m（SPSm）

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

禁止在运行过程中（SEmn=1）改写 SPSm 寄存器。

通过 16 位存储器操作指令设定 SPSm 寄存器。

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

在产生复位信号后，SPSm 寄存器的值变为“0000H”。

图 16-4：串行时钟选择寄存器 m（SPSm）的格式

复位后：0000H　　R/W

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

PRSmk3	PRSmk2	PRSmk1	PRSmk0	运行时钟（CKmk）的选择[注]
0	0	0	0	F_{CLK}
0	0	0	1	$F_{CLK}/2$
0	0	1	0	$F_{CLK}/2^2$
0	0	1	1	$F_{CLK}/2^3$
0	1	0	0	$F_{CLK}/2^4$
0	1	0	1	$F_{CLK}/2^5$
0	1	1	0	$F_{CLK}/2^6$
0	1	1	1	$F_{CLK}/2^7$
1	0	0	0	$F_{CLK}/2^8$
1	0	0	1	$F_{CLK}/2^9$
1	0	1	0	$F_{CLK}/2^{10}$
1	0	1	1	$F_{CLK}/2^{11}$
1	1	0	0	$F_{CLK}/2^{12}$
1	1	0	1	$F_{CLK}/2^{13}$
1	1	1	0	$F_{CLK}/2^{14}$
1	1	1	1	INTTM07

注：要在通用串行通信单元（SCI）运行过程中更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止 SCI 的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

注意：必须将 bit15~8 置“0”。

备注：

1. F_{CLK}：CPU/外围硬件的时钟频率
2. m：单元号（m=0、1、2）
3. k=0、1

16.3.3 串行模式寄存器mn（SMRmn）

SMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟（F_{MCK}）的选择、能否使用串行时钟（F_{SCLK}）输入的指定、开始触发的设定、运行模式（SSPI、UART、简易 I^2C）的设定以及中断源的选择。另外，只在 UART 模式中设定接收数据的反相电平。

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

通过 16 位存储器操作指令设定 SMRmn 寄存器。

在产生复位信号后，SMRmn 寄存器的值变为“0020H”。

图 16-5：串行模式寄存器 mn（SMRmn）的格式(1/2)

复位后：0020H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn[注1]	0	SISmn0[注1]	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	通道n运行时钟（F_{MCK}）的选择
0	SPSm寄存器设定的运行时钟CKm0
1	SPSm寄存器设定的运行时钟CKm1
运行时钟（F_{MCK}）用于边沿检测电路。通过设定CCSmn位和SDRmn寄存器的高7位，生成传送时钟（F_{TCLK}）。	

CCSmn	通道n传送时钟（F_{TCLK}）的选择
0	CKSmn位指定的运行时钟F_{MCK}的分频时钟
1	来自SCLKp引脚的输入时钟F_{SCLK}（SSPI模式的从属传送）
传送时钟F_{TCLK}用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当CCSmn位为“0”时，通过SDRmn寄存器的高7位进行运行时钟（F_{MCK}）的分频设定。	

STSmn[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^2C时选择）。
1	RxDq引脚的有效边沿（在UART接收时选择）
在将SSm寄存器置“1”后满足上述条件时，开始传送。	

注 1：只限于 SMR01、SMR11、SMR21 寄存器。

注意：必须将 bit13~9、7、4、3（SMR00、SMR10、SMR20 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） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）
q：UART 号（q=0~2） r：IIC 号（r=00、01、10、11、20、21）

图 16-5：串行模式寄存器 mn（SMRmn）的格式(2/2)

复位后：0020H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn注1	0	SISmn 0注1	1	0	0	MD mn2	MD mn1	MD mn0

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。

MDmn2	MDmn1	通道n运行模式的设定
0	0	SSPI模式
0	1	UART模式
1	0	简易I^2C模式
1	1	禁止设定。

MDmn0	通道n中断源的选择
0	传送结束中断
1	缓冲器空中断 （在数据从SDRmn寄存器传送到移位寄存器时发生）
在连续发送时，如果MDmn0位为“1”并且SDRmn的数据为空，就写下一个发送数据。	

注 1：只限于 SMR01、SMR11、SMR21、寄存器。

注意：必须将 bit13~9、7、4、3（SMR00、SMR10、SMR20 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） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）
q：UART 号（q=0~2） r：IIC 号（r=00、01、10、11、20、21）

16.3.4 串行通信运行设定寄存器mn（SCRmn）

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

禁止在运行过程中（SEmn=1）改写 SCRmn 寄存器。

通过 16 位存储器操作指令设定 SCRmn 寄存器。

在产生复位信号后，SCRmn 寄存器的值变为“0087H”。

图 16-6：串行通信运行设定寄存器 mn（SCRmn）的格式(1/3)

复位后：0087H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	0	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[注1]	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0

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

DAPmn	CKPmn	SSPI 模式中的数据和时钟的相位选择	类型
0	0	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp输入时序	1
0	1	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp输入时序	2
1	0	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp输入时序	3
1	1	SCLKp SDOp D7 D6 D5 D4 D3 D2 D1 D0 SDIp输入时序	4
在 UART 模式和简易 I²C 模式中，必须将 DAPmn 位和 CKPmn 位都置 “0”。			

注 1：只限于 SCR00、SCR10、SCR20 寄存器。

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

备注：m：单元号（m=0、1、2） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）

图 16-6：串行通信运行设定寄存器 mn（SCRmn）的格式(2/3)

复位后：0087H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	0	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[注1]	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0

PTCmn1	PTCmn0	UART模式中的奇偶校验位的设定	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出奇偶校验[注3]。	不判断奇偶校验。
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。
在SSPI模式和简易I²C模式中，必须将PTCmn1位和PTCmn0位都置“0”。			

DIRmn	SSPI和UART模式中的数据传送顺序的选择
0	进行MSB优先的输入/输出。
1	进行LSB优先的输入/输出。
在简易I²C模式中，必须将DIRmn位置“0”。	

SLCmn1[注1]	SLCmn0	UART模式中的停止位的设定
0	0	无停止位
0	1	停止位长度=1位
1	0	停止位长度=2位（只限于mn=00、10、20）
1	1	禁止设定。
如果选择了传送结束中断，就在传送完所有停止位后产生中断。 在UART接收时或者在简易I²C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）。		

注 1：只限于 SCR00、SCR10、SCR20 寄存器。

注 2：与数据的内容无关，总是附加“0”。

注意：必须将 bit6、10、11 置“0”。

备注：m：单元号（m=0、1、2） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）

图 16-6：串行通信运行设定寄存器 mn（SCRmn）的格式(3/3)

复位后：0087H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	0	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[注1]	SLCmn0	DLSmn3	DLSmn2	DLSmn1[注2]	DLSmn0

DLSmn3	DLSmn2	DLSmn1	DLSmn0	SSPI模式中的数据长度的设定	串行功能对应		
					SSPI	UART	IIC
0	1	1	0	7位数据长度（保存在SDRmn寄存器的bit0~6）	○	○	×
0	1	1	1	8位数据长度（保存在SDRmn寄存器的bit0~7）	○	○	○
1	0	0	0	9位数据长度（保存在SDRmn寄存器的bit0~8）	○	○	×
1	0	0	1	10位数据长度（保存在SDRmn寄存器的bit0~9）	○	×	×
1	0	1	0	11位数据长度（保存在SDRmn寄存器的bit0~10）	○	×	×
1	0	1	1	12位数据长度（保存在SDRmn寄存器的bit0~11）	○	×	×
1	1	0	0	13位数据长度（保存在SDRmn寄存器的bit0~12）	○	×	×
1	1	0	1	14位数据长度（保存在SDRmn寄存器的bit0~13）	○	×	×
1	1	1	0	15位数据长度（保存在SDRmn寄存器的bit0~14）	○	×	×
1	1	1	1	16位数据长度（保存在SDRmn寄存器的bit0~15）	○	○	×
其他				禁止设置。			
在简易I²C模式中，必须设置DLSmn3~DLSmn0=0111B。							

注 1：只限于 SCR00、SCR10、SCR20 寄存器。

注意：必须将 bit6、10、11 置“0”。

备注：m：单元号（m=0、1、2） n：通道号（n=0、1） p：SSPI 号（p=00、01、10、11、20、21）

16.3.5 串行数据寄存器mn（SDRmn）

SDRmn 寄存器是通道 n 发送和接收的数据寄存器（16 位）。

当运行停止（SEmn=0）时，bit15~9 用作运行时钟（F_{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。

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

通过 16 位存储器操作指令设定 SIRmn 寄存器。

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

图 16-7：串行数据寄存器 mn（SDRmn）的格式

复位后：0000H　　R/W

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

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

注意：

1. 运行停止（SEmn=0）时，必须将 bit8~0 清零。
2. 在使用 UART 时，禁止将 SDRmn[15:9]设定为“0000000B”和“0000001B”。
3. 在使用简易 I^2C 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. 运行停止（SEmn=0）时，禁止通过 8 位存储器操作指令改写 SDRmn[7:0]（否则，SDRmn[15:9]全部被清“0”）。

备注：

1. SDRmn 寄存器在运行期间的功能，请参照“16.2 通用串行通信单元的结构”。
2. m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3.6 串行标志清除触发寄存器mn（SIRmn）

这是用于清除通道 n 各错误标志的触发寄存器。

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

通过 16 位存储器操作指令设定 SIRmn 寄存器。

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

在产生复位信号后，SIRmn 寄存器的值变为“0000H”。

图 16-8：串行标志清除触发寄存器 mn（SIRmn）的格式

复位后：0000H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn[注1]	PECTmn	OVCTmn

FECTmn[注1]	通道n帧错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的FEFmn位清“0”。

PECTmn	通道n奇偶校验错误标志的清除触发
0	不清除。
1	将SSRmn寄存器的PEFmn位清“0”。

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

注 1：只限于 SIR01、SIR11、SIR21 寄存器。

注意：必须将 bit15~3（SIR00、SIR10、SIR20 寄存器为 bit15~2）置“0”。

备注：

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

16.3.7 串行状态寄存器mn（SSRmn）

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

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

在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

图 16-9：串行状态寄存器 mn（SSRmn）的格式(1/2)

复位后：0000H　　R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn 注1	PEF mn	OVF mn

TSFmn	通道n通信状态的表示标志
0	通信停止状态或者通信待机状态
1	通信运行状态
[清除条件] • 当将STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信待机状态）时 • 当通信结束时 [置位条件] • 当开始通信时	

BFFmn	通道n缓冲寄存器的状态表示标志
0	SDRmn寄存器没有保存有效数据。
1	SDRmn寄存器保存了有效数据。
[清除条件] • 在发送过程中传送完从SDRmn寄存器到移位寄存器的发送数据时 • 在接收过程中从SDRmn寄存器读完接收数据时 • 当将STm寄存器的STmn位置“1”（通信停止状态）或者将SSm寄存器的SSmn位置“1”（通信允许状态）时 [置位条件] • 在SCRmn寄存器的TXEmn位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给SDRmn寄存器写发送数据时 • 在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到SDRmn寄存器时 • 当发生接收错误时	

注 1：只限于 SSR01、SSR11、SSR21 寄存器。

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

备注：m：单元号（m=0、1、2） n：通道号（n=0、1）

图 16-9：串行状态寄存器 mn（SSRmn）的格式(2/2)

复位后：0000H　　R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn[注1]	PEF mn	OVF mn

FEFmn[注1]	通道n帧错误的检测标志
0	没有发生错误。
1	发生错误（UART接收时）。
[清除条件] • 当给SIRmn寄存器的FECTmn位写“1”时 [置位条件] • 在UART接收结束时没有检测到停止位时	

PEFmn	通道n奇偶校验错误的检测标志
0	没有发生错误。
1	发生错误（UART接收时）或者未检测到ACK（I^2C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^2C发送时并且在ACK接收时序从属方没有返回ACK信号（未检测到ACK）时	

OVFmn	通道n溢出错误的检测标志
0	没有发生错误。
1	发生错误。
[清除条件] • 当给SIRmn寄存器的OVCTmn位写“1”时 [置位条件] • 在SCRmn寄存器的RXEmn位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下，虽然接收数据被保存在SDRmn寄存器，但是没有读接收数据而写发送数据或者写下一个接收数据时 • 在SSPI模式的从属发送或者从属发送和接收过程中未准备好发送数据时	

注 1：只限于 SSR01、SSR11、SSR21 寄存器。

备注：m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3.8 串行通道开始寄存器m（SSm）

SSm 寄存器是设定允许各通道的通信/开始计数的触发寄存器。

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

通过 16 位存储器操作指令设定 SSm 寄存器。

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

在产生复位信号后，SSm 寄存器的值变为“0000H”。

图 16-10：串行通道开始寄存器 m（SSm）的格式

复位后：0000H　　R/W

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

SSmn	通道n运行开始的触发
0	没有触发。
1	将SEmn位置“1”，转移到通信待机状态注。

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

注意：必须将 SSm 寄存器的 bit15~2 置“0”。

备注：

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

16.3.9 串行通道停止寄存器m（STm）

STm 寄存器是设定允许各通道的通信/停止计数的触发寄存器。

如果给各位（STmn）写“1”，就将串行通道允许状态寄存器 m（SEm）的对应位（SEmn）清“0”（运行停止状态）。因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

通过 16 位存储器操作指令设定 STm 寄存器。

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

在产生复位信号后，STm 寄存器的值变为“0000H”。

图 16-11：串行通道停止寄存器 m（STm）的格式

复位后：0000H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1	STm0

STmn	通道n运行的停止触发
0	没有触发。
1	将SEmn位清“0”，停止通信运行[注]。

注：控制寄存器和移位寄存器的值、SCLKmn 引脚和 SDOmn 引脚以及 FEFmn 标志、PEFmn 标志和 OVFmn 标志保持状态。

注意：必须将 STm 寄存器的 bit15~2 置“0”。

备注：

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

16.3.10 串行通道允许状态寄存器m（SEm）

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

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

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

对于停止运行的通道 n，能通过软件设定 SOm 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令读取 SEm 寄存器。

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

在产生复位信号后，SEm 寄存器的值变为“0000H”。

图 16-12：串行通道允许状态寄存器 m（SEm）的格式

复位后：0000H　　R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1	SEm0

SEmn	通道n运行的允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

备注：m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3.11 串行输出允许寄存器m（SOEm）

SOEm 寄存器设定允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道 n，无法通过软件改写后述的串行输出寄存器 m（SOm）的 SOmn 位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道 n，能通过软件设定 SOm 寄存器的 SOmn 位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令设定 SOEm 寄存器。

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

在产生复位信号后，SOEm 寄存器的值变为“0000H”。

图 16-13：串行输出允许寄存器 m（SOEm）的格式

复位后：0000H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1	SOEm0

SOEmn	通道n串行输出的允许或者停止
0	停止串行通信的输出。
1	允许串行通信的输出。

注意：SOEm 寄存器的 bit15~2 置“0”。

备注：m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3.12 串行输出寄存器m（SOm）

SOm 寄存器是各通道串行输出的缓冲寄存器。

从通道 n 的串行数据输出引脚输出此寄存器的 SOmn 位的值。

从通道 n 的串行时钟输出引脚输出此寄存器的 CKOmn 位的值。

只有在禁止串行输出时（SOEmn=0）才能通过软件改写此寄存器的 SOmn 位。当允许串行输出（SOEmn=1）时，忽视通过软件的改写而只能通过串行通信更改此寄存器的 SOmn 位的值。

只有在通道停止运行时（SEmn=0）才能通过软件改写此寄存器的 CKOmn 位。当允许通道运行（SEmn=1）时，忽视通过软件的改写而只能通过串行通信更改此寄存器的 CKOmn 位的值。

要将串行接口的引脚用作端口功能等非串行接口功能时，必须将相应的 CKOmn 位和 SOmn 位置“1”。

通过 16 位存储器操作指令设定 SOm 寄存器。

在产生复位信号后，SOm 寄存器的值变为“0303H”。

图 16-14：串行输出寄存器 m（SOm）的格式

复位后：0303H　　R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOm1	SOm0

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

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

注意：必须将 SOm 寄存器的 bit15~10 和 bit7~2 置“0”。

备注：m：单元号（m=0、1、2） n：通道号（n=0、1）

16.3.13 串行输出电平寄存器m（SOLm）

SOLm 寄存器是设定各通道的数据输出电平反相的寄存器。

只有在 UART 模式中才能设定此寄存器。在 SSPI 模式和简易 I²C 模式中，必须将对应位置“0”。只在允许串行输出时（SOEmn=1），将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时（SOEmn=0），将 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”。

图 16-15：串行输出电平寄存器 m（SOLm）的格式

复位后：0000H　　R/W

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

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

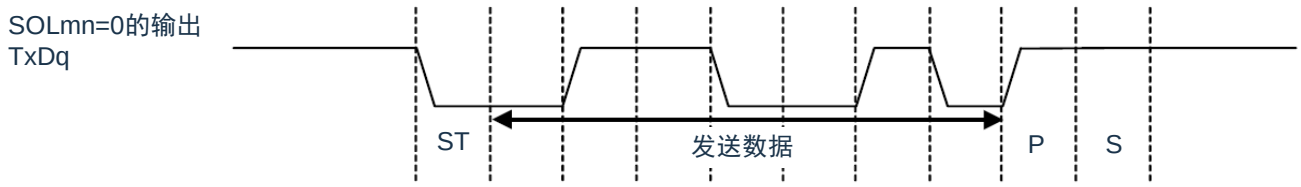
注意：必须将 SOL0、SOL1、SOL2 寄存器的 bit15~1 置“0”。

备注：m：单元号（m=0、1、2） n：通道号（n=0）

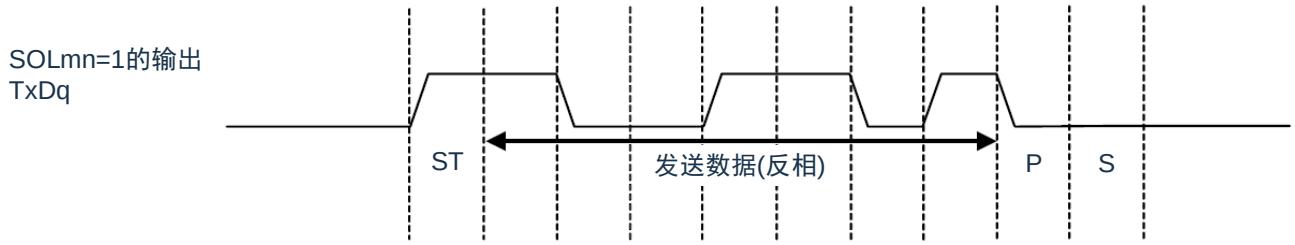
当进行 UART 发送时，发送数据的电平反相例子如图 16-16 所示。

图 16-16：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） n：通道号（n=0）

16.3.14 串行待机控制寄存器m（SSCm）

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

SSC1 寄存器是控制在 SSPI10 或者 UART1 的串行数据接收时从深度睡眠模式启动接收运行的寄存器。

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

通过 16 位存储器操作指令设定 SSCm 寄存器。

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

在产生复位信号后， SSCm 寄存器的值变为 “0000H”。

图 16-17：串行输出电平寄存器 m（SSCm）的格式

复位后：0000H　　R/W

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

SSECm	深度睡眠唤醒模式中的通信错误中断产生的允许或者停止的选择
0	允许产生错误中断。
1	禁止产生错误中断。

- 在UART低功耗模式中进行UART接收的情况下，只有在SWCm位和EOCmn位为“1”时才能将SSECm位置 “1” 或者 “0”，否则必须将 SSECm 位置 “0”。
- 禁止将 SSECm 位和 SWCm 位分别置 “1” 和 “0”。

SWCm	低功耗 UART 模式的设定
0	不使用深度睡眠唤醒模式功能。
1	用深度睡眠唤醒模式功能。

- 在深度睡眠模式中，通过硬件触发信号解除深度睡眠模式，并且在 CPU 不运行的状态下进行UART 的接收（低功耗 UART 模式）。
- 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟（f_{CLK}）时才能设定低功耗UART模式功能，而在选择其他时钟的情况下禁止设定。
- 即使使用低功耗UART模式，也必须在通常运行模式中将 SWCm 位置 “0” 并且在即将要转移到深度睡眠模式前将 SWCm 位改为 “1”。

  另外，必须在从深度睡眠模式返回到通常运行模式后将 SWCm 位改为 “0”。

16.3.15 从属选择功能启用寄存器m（SSEm）

SSEm 寄存器是控制在 SSPImn 通信中作为从属功能时 SSImn 端子的输入是有效还是无效。

通过 16 位存储器操作指令设定 SSCm 寄存器。

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

在产生复位信号后， SSCm 寄存器的值变为 “0000H”。

图 16-18：从属选择功能启用寄存器 m（SSEm）的格式

复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIEm1	SSIEm0

SSIEm	SSPImn 通信的从属模式中通道 n 的 SSIm 输入的设定
0	SSImn引脚输入无效。
1	SSImn引脚输入有效

16.3.16 输入切换控制寄存器（ISC）

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

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

通过 8 位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

图 16-19：输入切换控制寄存器（ISC）的格式

地址：40040C03H复位后：00H　　R/W

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

ISC1	定时器Timer8的通道3的输入切换
0	将TI03引脚的输入信号用作定时器的输入（通常运行）。
1	将RxD0引脚的输入信号用作定时器的输入（检测唤醒信号并且测量间隔段的低电平宽度和同步段的脉宽）。

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

注意：必须将 bit6~2 置“0”。

16.3.17 噪声滤波器允许寄存器0（NFEN0）

NFEN0 寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 SSPI 或者简易 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”。

图 16-20：噪声滤波器允许寄存器 0（NFEN0）的格式

地址：40040C00H　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	RxD2引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD2引脚时，必须将SNFEN20位置“1”。 当用作RxD2引脚以外的功能时，必须将SNFEN20位置“0”。	

SNFEN10	RxD1引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD1引脚时，必须将SNFEN10位置“1”。 当用作RxD1引脚以外的功能时，必须将SNFEN10位置“0”。	

SNFEN00	RxD0引脚的噪声滤波器的使用与否
0	噪声滤波器OFF
1	噪声滤波器ON
当用作RxD0引脚时，必须将SNFEN00位置“1”。 当用作RxD0引脚以外的功能时，必须将SNFEN00位置“0”。	

注意：必须将 bit7~5、3、1 置“0”。

16.4 运行停止模式

通用串行通信单元的各串行接口有运行停止模式。在运行停止模式中不能进行串行通信，因此能降低功耗。另外，在运行停止模式中能将用于串行接口的引脚用作端口功能。

16.4.1 以单元为单位停止运行的情况

通过外围允许寄存器 0（PER0）设定以单元为单位的停止运行。

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

要停止通用串行通信单元 0 时，必须将 bit2（SCI0EN）置“0”；要停止通用串行通信单元 1 时，必须将 bit3（SCI1EN）置“0”。 要停止通用串行通信单元 2 时，必须将 bit4（SCI2EN）置“0”。

外围允许寄存器 0（PER0）……只将要停止 SCIm 的对应位置“0”。

图 16-21：以单元为单位停止运行时的外围允许寄存器 0（PER0）的设定

	7	6	5	4	3	2	1	0
PER0	XX	XX	XX	SCI2EN	SCI1EN	SCI0EN	XX	XX

SCIm 输入时钟的控制

0：停止提供输入时钟

1：提供输入时钟

注意：当 SCImEN 位为“0”时，忽视通用串行通信单元 m 的控制寄存器的写操作，而且读取值都为初始值。但是，以下的寄存器除外：

- 输入切换控制寄存器（ISC）
- 噪声滤波器允许寄存器 0（NFEN0）
- 端口复用功能配置寄存器（PxxCFG）
- 端口输出模式寄存器（POMx）
- 端口模式寄存器（PMx）
- 端口寄存器（Px）

备注：×：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”。

16.4.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

图 16-22：按通道停止运行时的各寄存器的设定

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1 0/1	STm0 0/1

1：将SEmn位清“0”并且停止通信运行

※因为STmn位是触发位，所以如果SEmn位为“0”就立即清除STmn位

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

	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	SEm1 0/1	SEm0 0/1

0：运行停止状态

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

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

	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 0/1	SOEm0 0/1

0：通过串行通信运行来停止输出

※对于已经停止串行输出的通道，能通过软件设定SOm寄存器的SOmn位的值。

(d) 串行输出寄存器m(S0m) …...这是各通道串行输出的缓冲寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

1：串行时钟的输出值为“1”　　1：串行时钟的输出值为“1”

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

注：只限于通用串行通信单元 0。

备注：

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

16.5 3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的运行

这是通过串行时钟（SCLK）和串行数据（SDI 和 SDO）共 3 条线实现的时钟同步通信功能。

[数据的发送和接收]

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

[时钟控制]

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

主控通信：最大值 $F_{CLK}/2$

从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

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

SCI0 的通道 0~1，SCI1 的通道 0~1 以及 SCI2 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 16.5.1）
- 主控接收（参照 16.5.2）
- 主控的发送和接收（参照 16.5.3）
- 从属发送（参照 16.5.4）
- 从属接收（参照 16.5.5）
- 从属的发送和接收（参照 16.5.6）

16.5.1 主控发送

主控发送是指本产品输出传送时钟并且将数据发送到其他设备的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCLK00 SDO00	SCLK01 SDO01	SCLK10 SDO10	SCLK11 SDO11	SCLK20 SDO20	SCLK21 SDO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	无					
传送数据长度	7 位~16 位					
传送速率[注]	Max.F_{CLK}/2[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。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）mn=00~01、10~11、20~21

(1) 寄存器的设定

图 16-23：3 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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（运行时钟（fMCK）的分频设定）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送数据寄存器															

SDRmnL

(d) 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 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

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”。

	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 0/1	SOEm0 0/1

(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 0/1	SSm0 0/1

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）mn=00~01、10~11、20~21
2. ■：不能设定（设定初始值）。0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-24：主控发送的初始设置步骤

图 16-25：主控发送的中止步骤

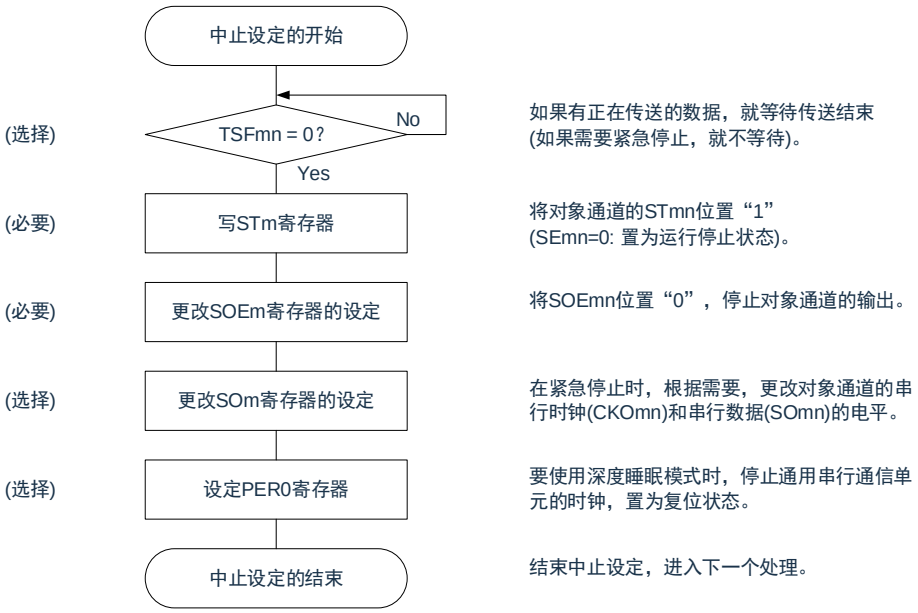
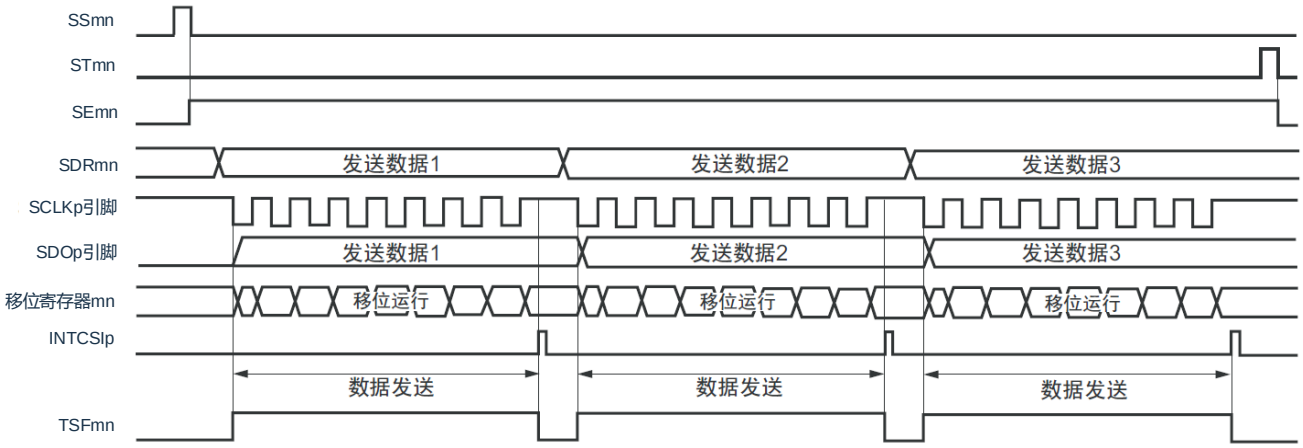


图 16-26：重新开始主控发送的设定步骤

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

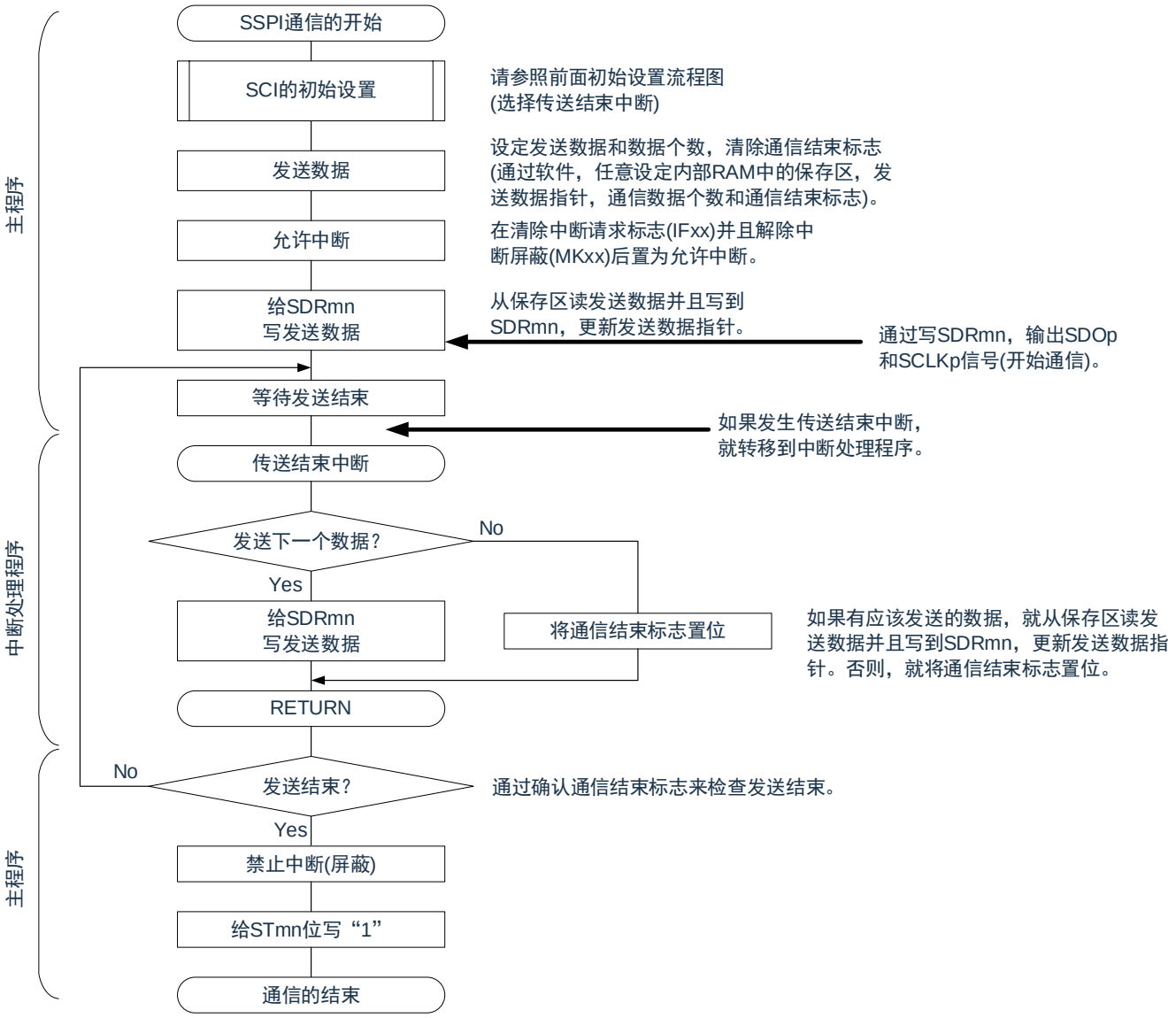
(3) 处理流程（单次发送模式）

图 16-27：主控发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



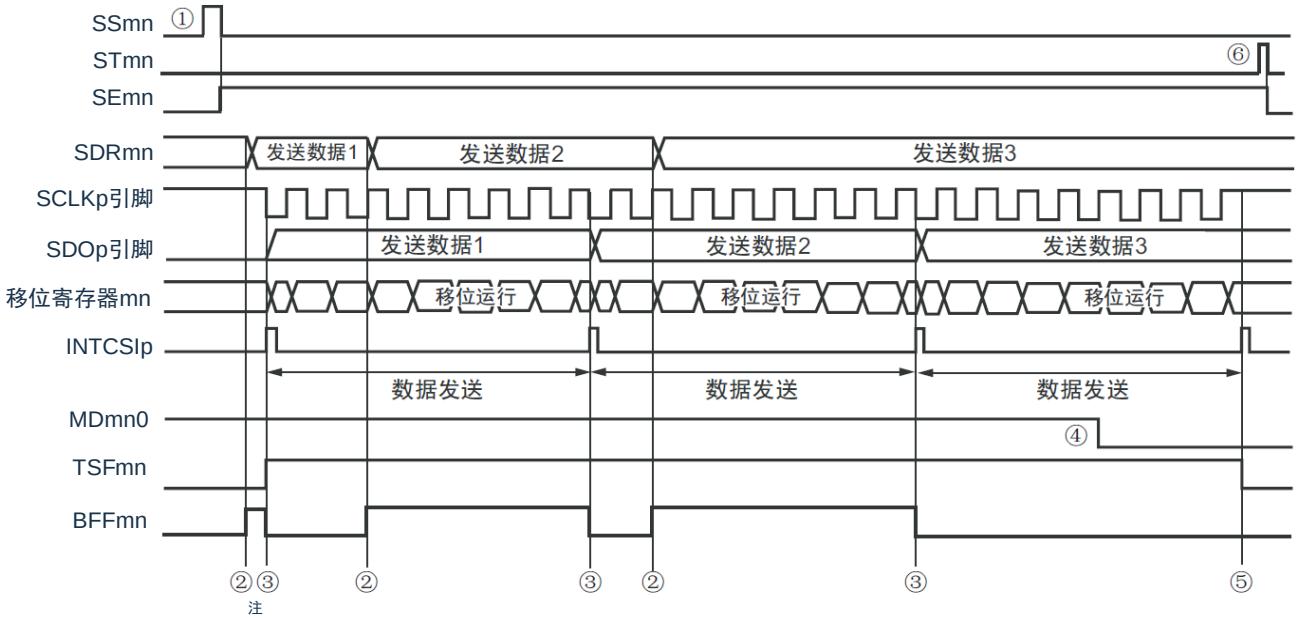
备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-28：主控发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 16-29：主控发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

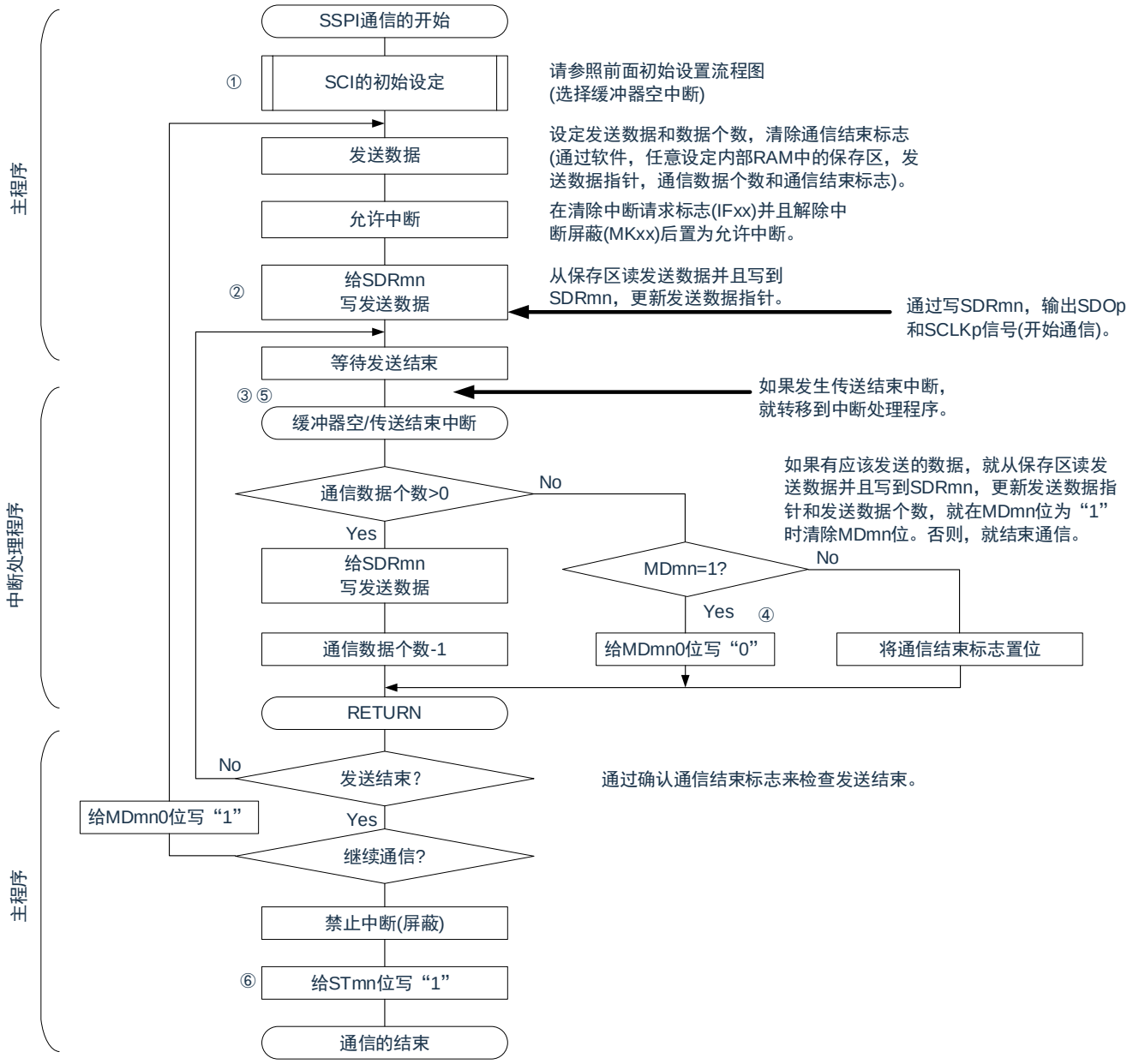


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-30：主控发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 16-29：主控发送（连续发送模式）的时序图”中的①~⑥。

16.5.2 主控接收

主控接收是指本产品输出传送时钟并且从其他设备接收数据的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCLK00 SDO00	SCLK01 SDO01	SCLK10 SDO10	SCLK11 SDO11	SCLK20 SDO20	SCLK21 SDO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位~16 位					
传送速率[注]	Max.F_{CLK}/2[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。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

(1) 寄存器的设定

图 16-31：3 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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（运行时钟（fMCK）的分频设定）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	接收数据寄存器															

SDRmnL

(d) 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 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 ×	SOm0 ×

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) …...在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(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 0/1	SSm0 0/1

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：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 16-32：主控接收的初始设定步骤

图 16-33：主控接收的中止步骤

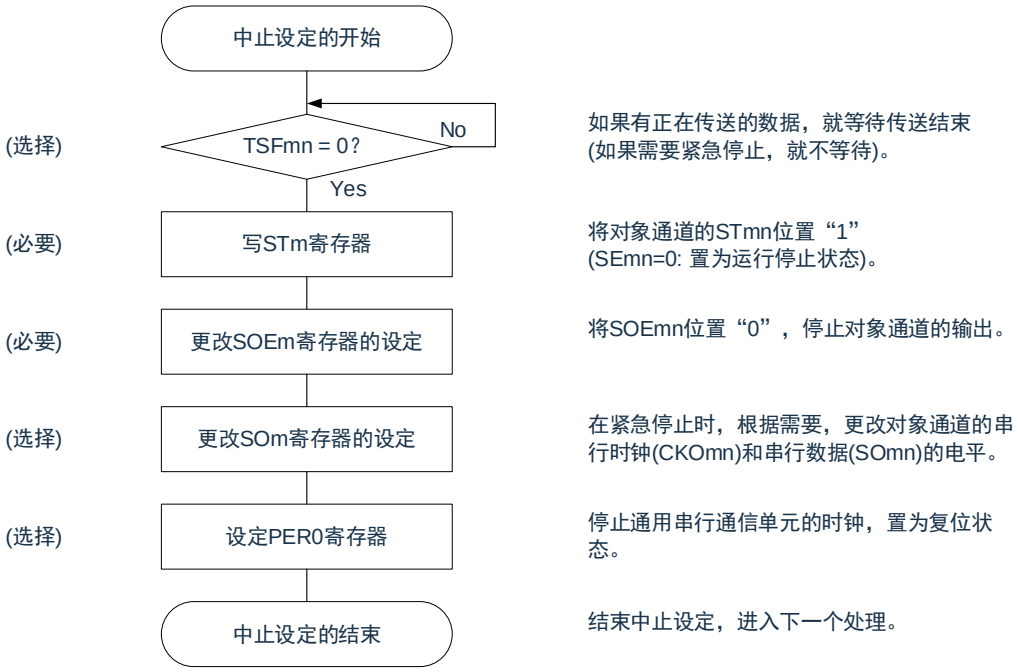
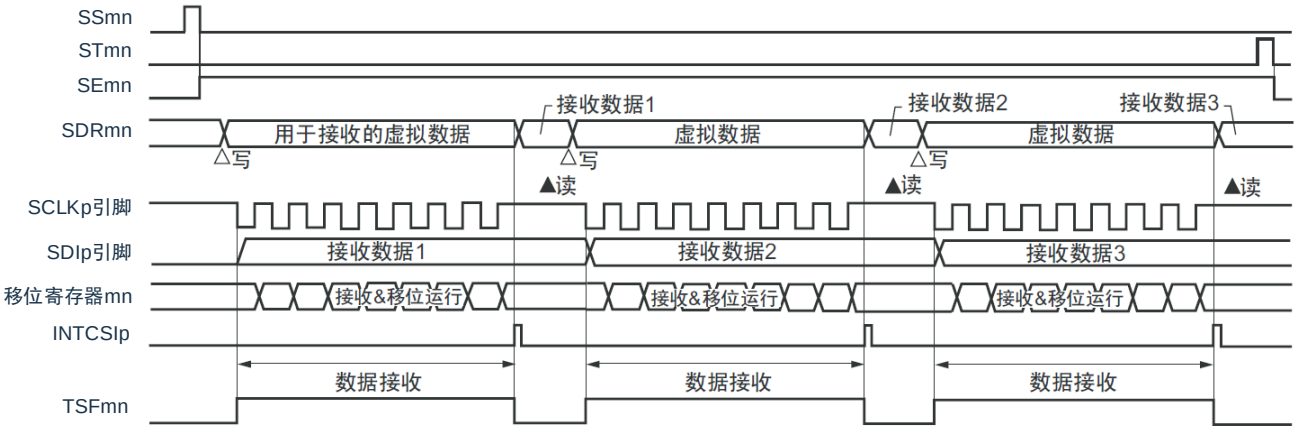


图 16-34：重新开始主控接收的设定步骤

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

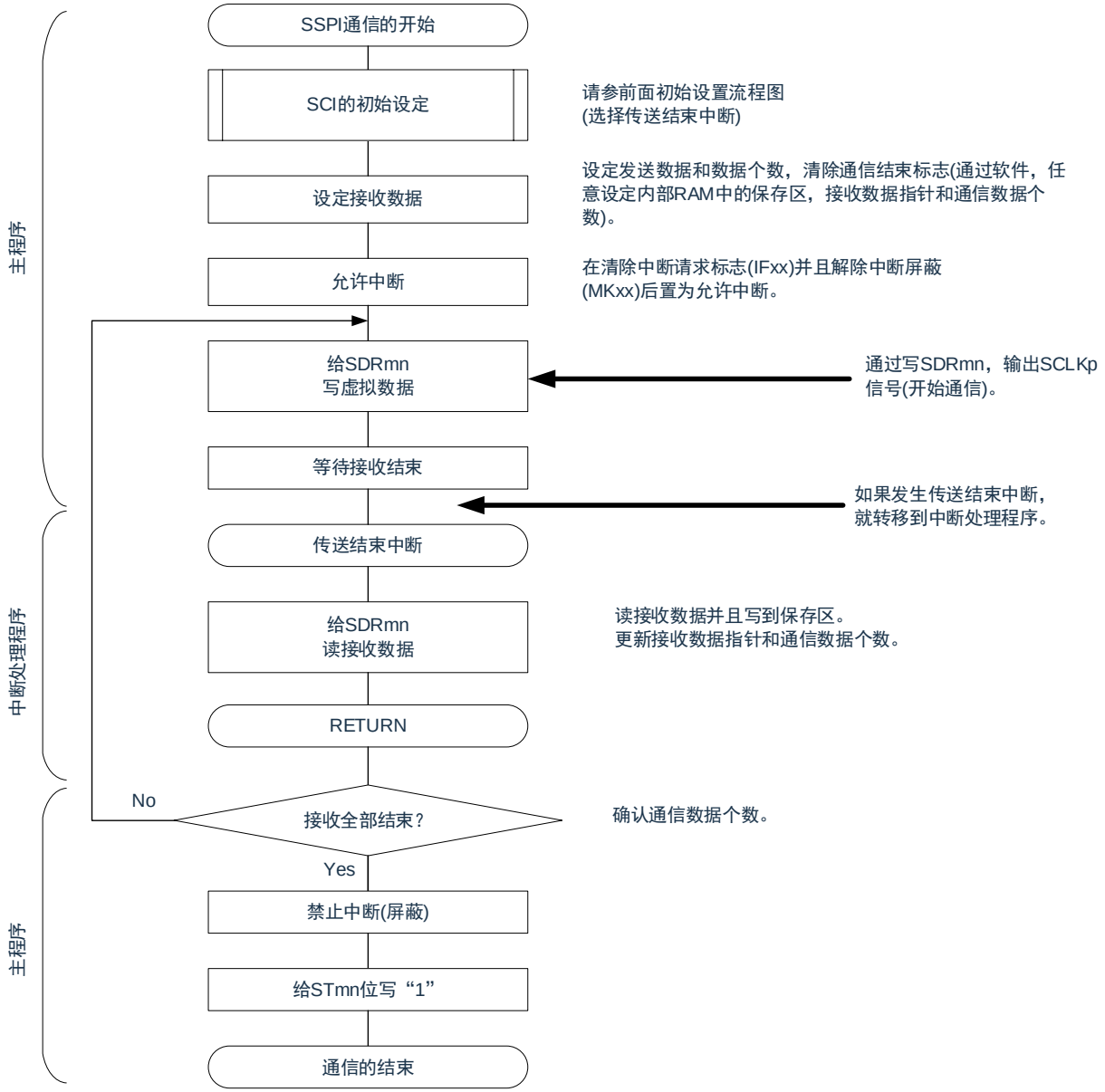
(3) 处理流程（单次接收模式）

图 16-35：主控接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



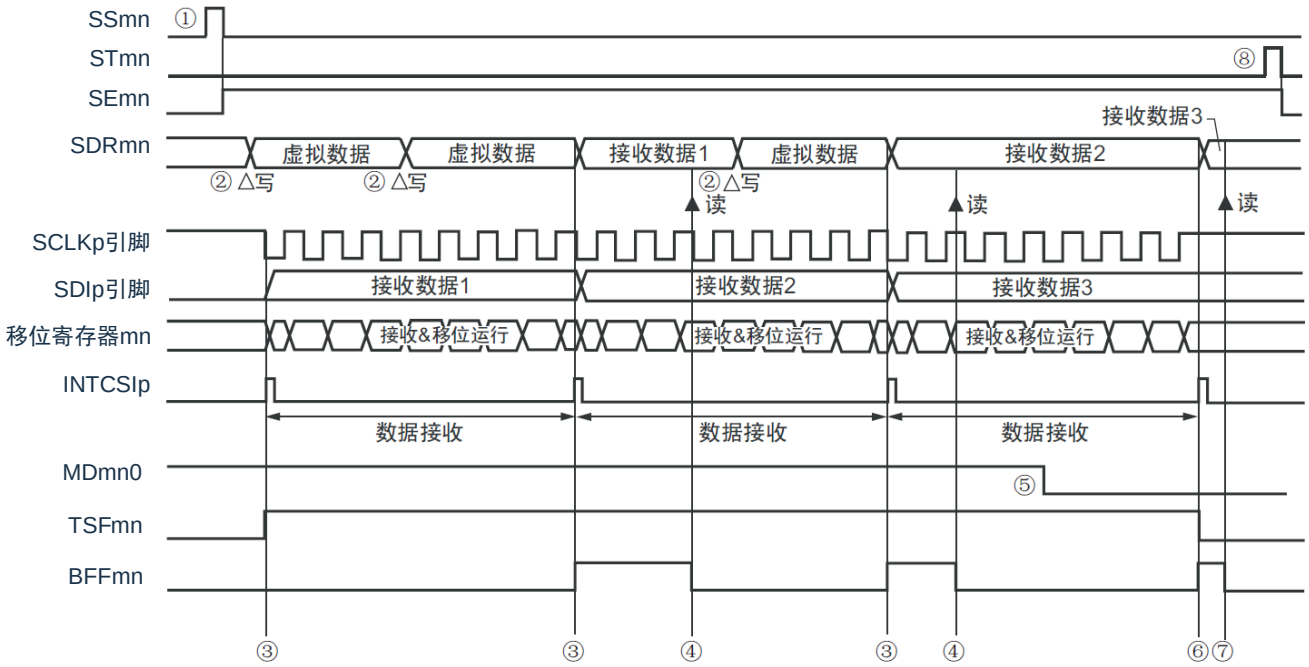
备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-36：主控接收（单次接收模式）的流程图



(4) 处理流程（连续接收模式）

图 16-37：主控接收（连续接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

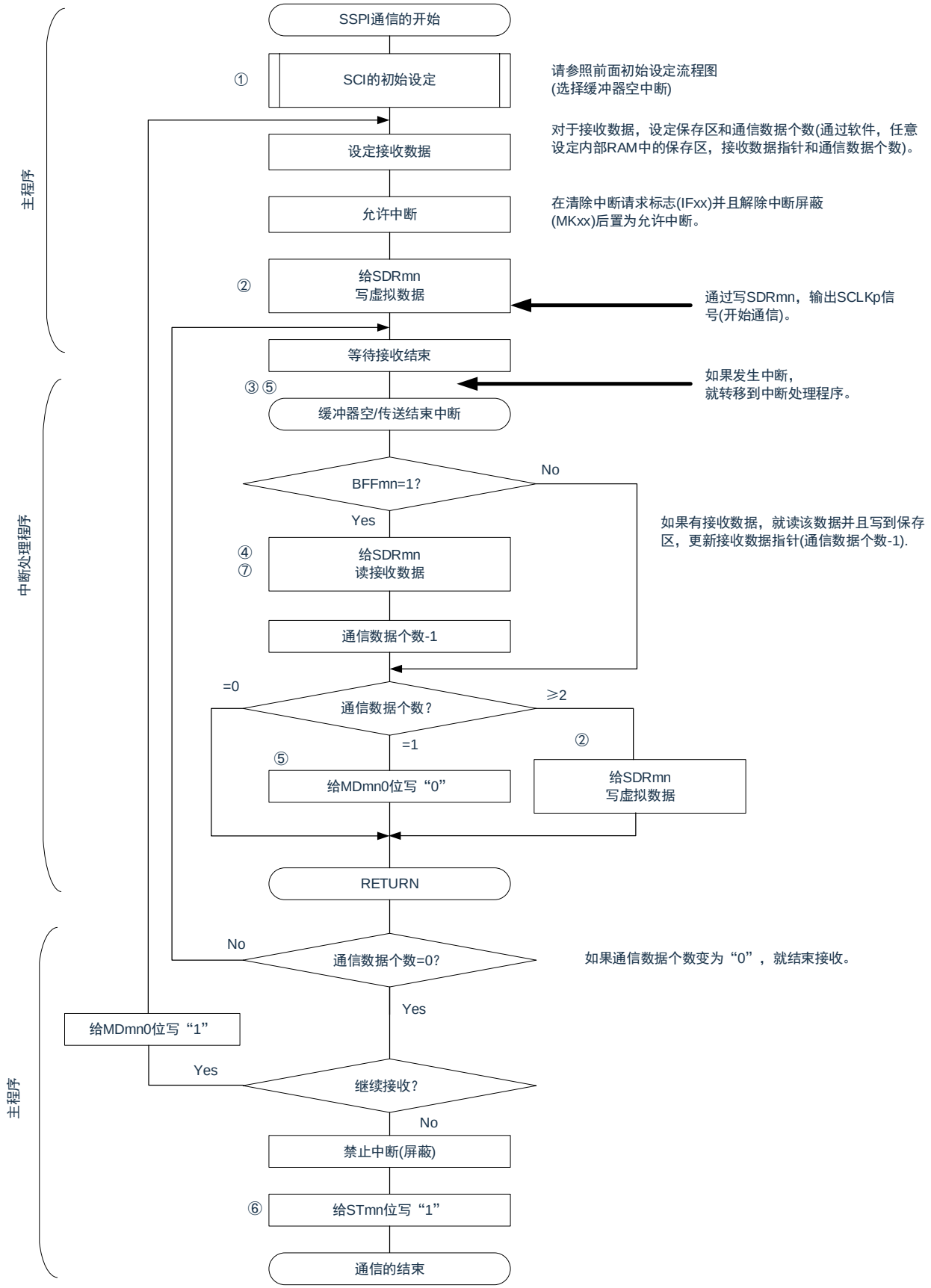


注意：即使在运行过程中也能改写 MDmn0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注：

1. 图中的①~⑧对应“图 16-38：主控接收（连续接收模式）的流程图”中的①~⑧。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-38：主控接收（连续接收模式）的流程图



备注：图中的①~⑧对应“图 16-37 主控接收（连续接收模式）的时序图”中的①~⑧。

16.5.3 主控的发送和接收

主控的发送和接收是指本产品输出传送时钟并且和其他设备进行数据发送和接收的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCLK00 SDI00 SDO00	SCLK01 SDI01 SDO01	SCLK10 SDI10 SDO10	SCLK11 SDI11 SDO11	SCLK20 SDI20 SDO20	SCLK21 SDI21 SDO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位~16 位					
传送速率[注]	Max.F_{CLK}/2[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。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

(1) 寄存器的设定

图 16-39：3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）

主控发送和接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（运行时钟（fMCK）的分频设定）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送/接收数据寄存器															

SDRmnL

(d) 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 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

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”。

	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 0/1	SOEm0 0/1

(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 0/1	SSm0 0/1

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：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 16-40：主控发送和接收的初始设定步骤

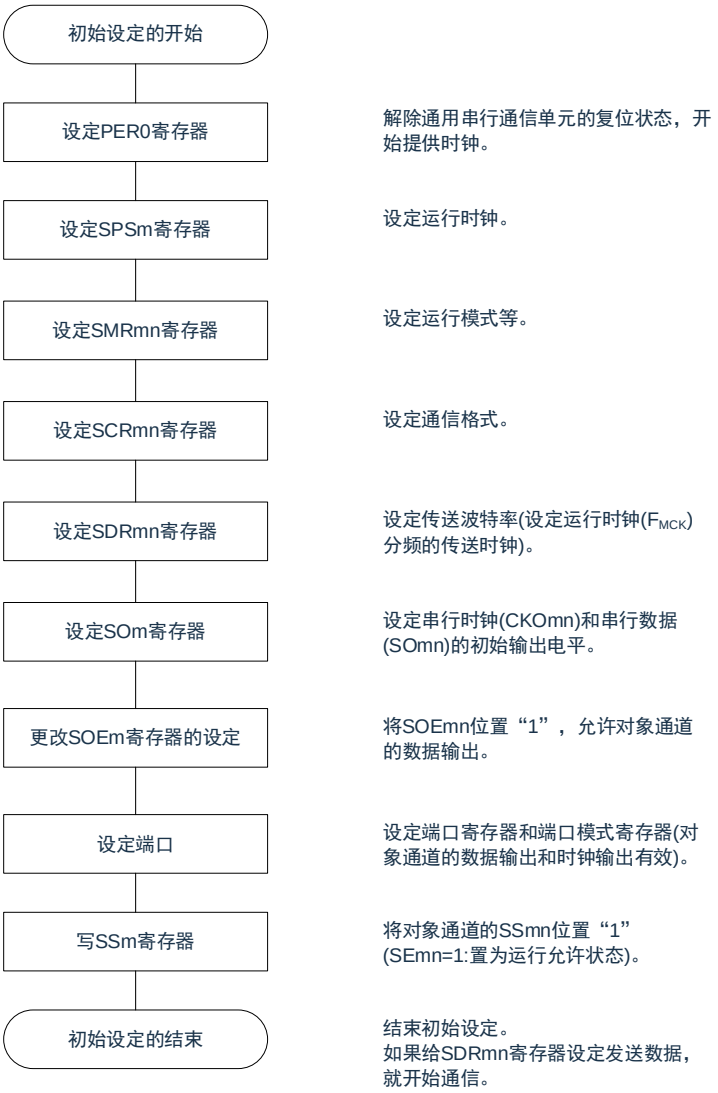


图 16-41：主控发送和接收的中止步骤

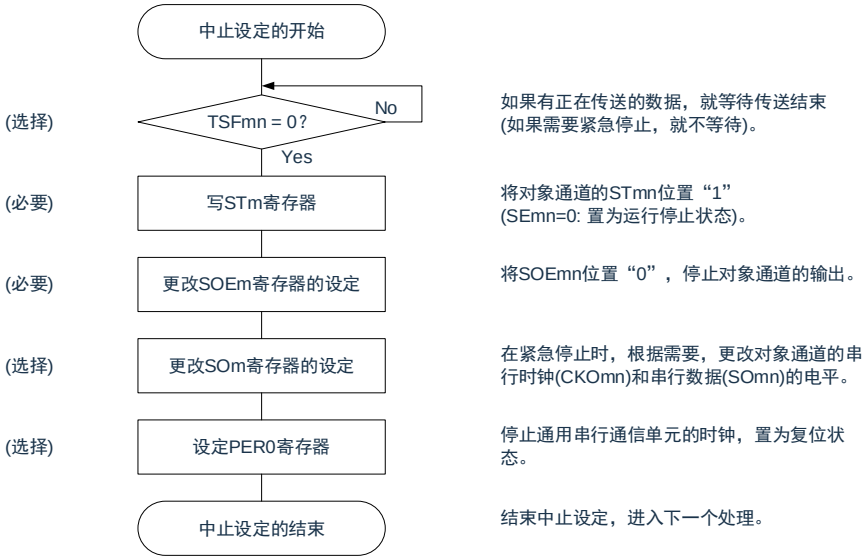
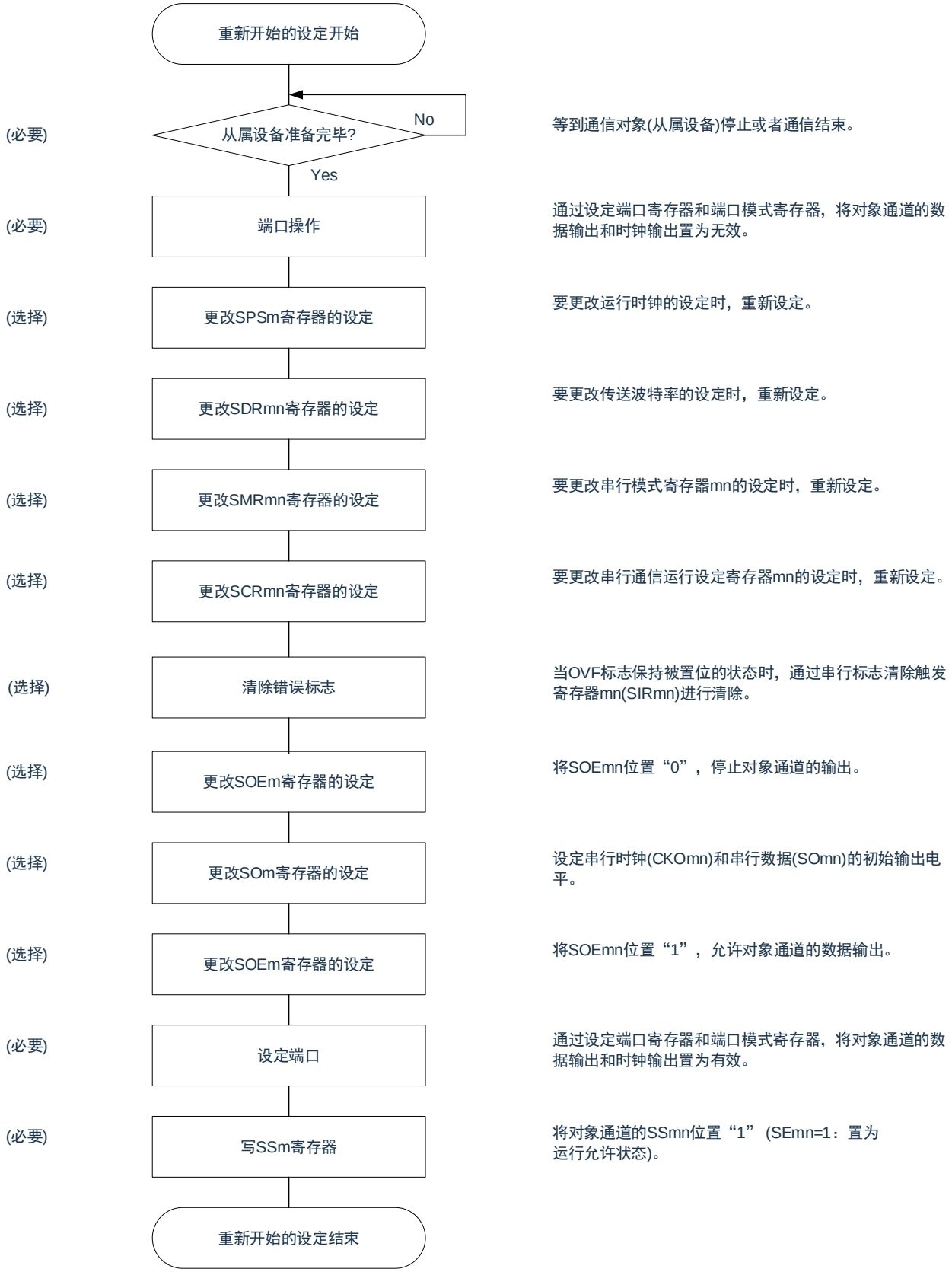
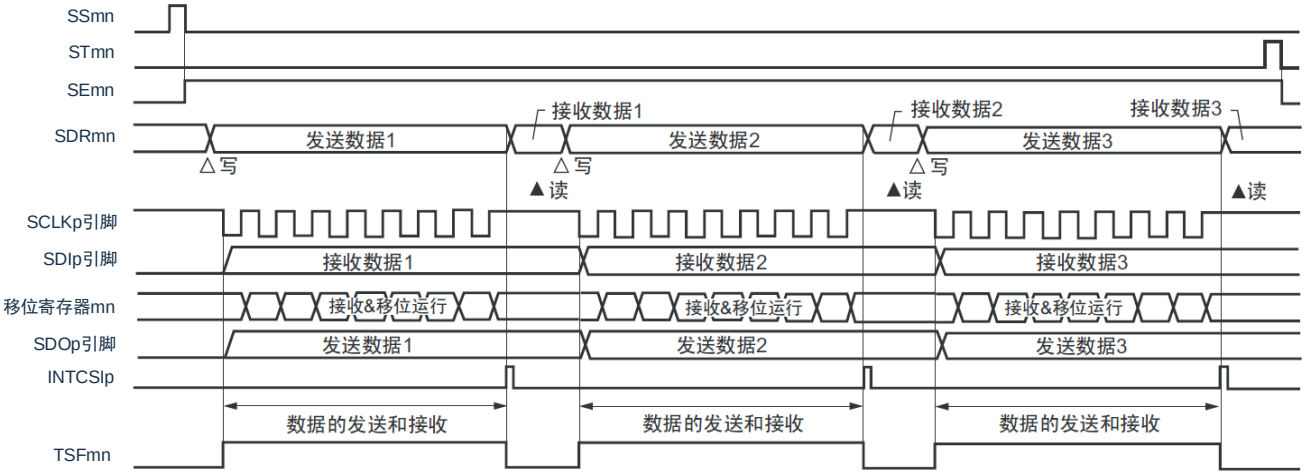


图 16-42：重新开始主控发送和接收的设定步骤



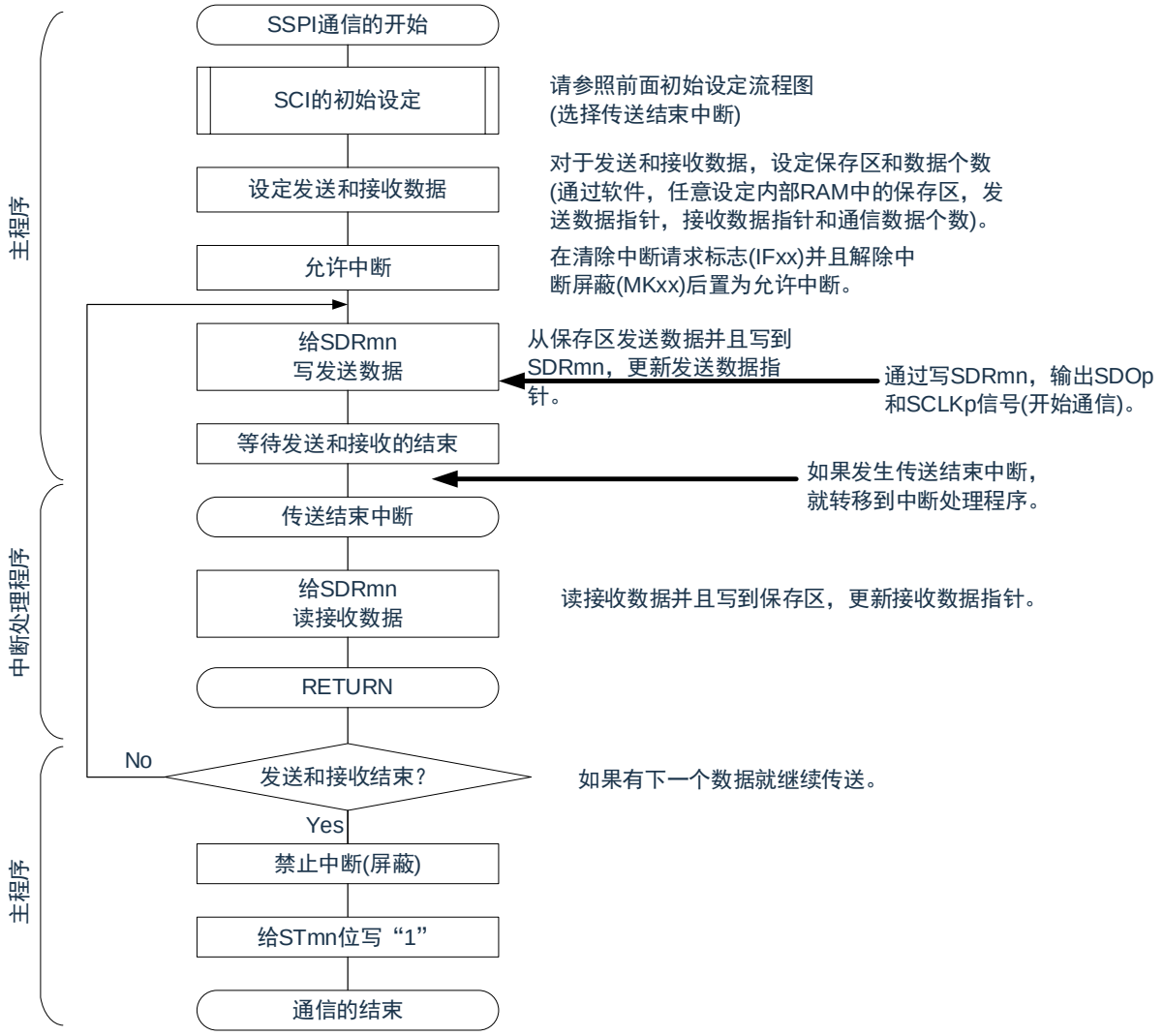
(3) 处理流程（单次发送和接收模式）

图 16-43：主控发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



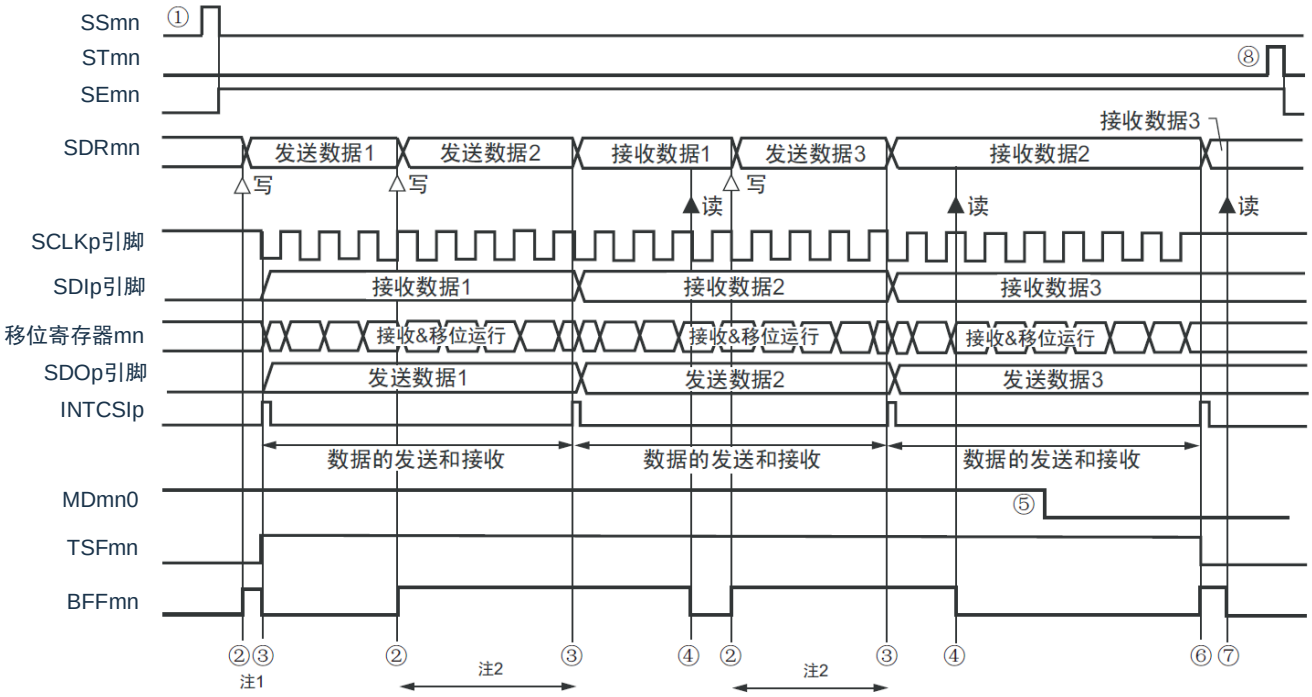
备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-44：主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图 16-45：主控发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

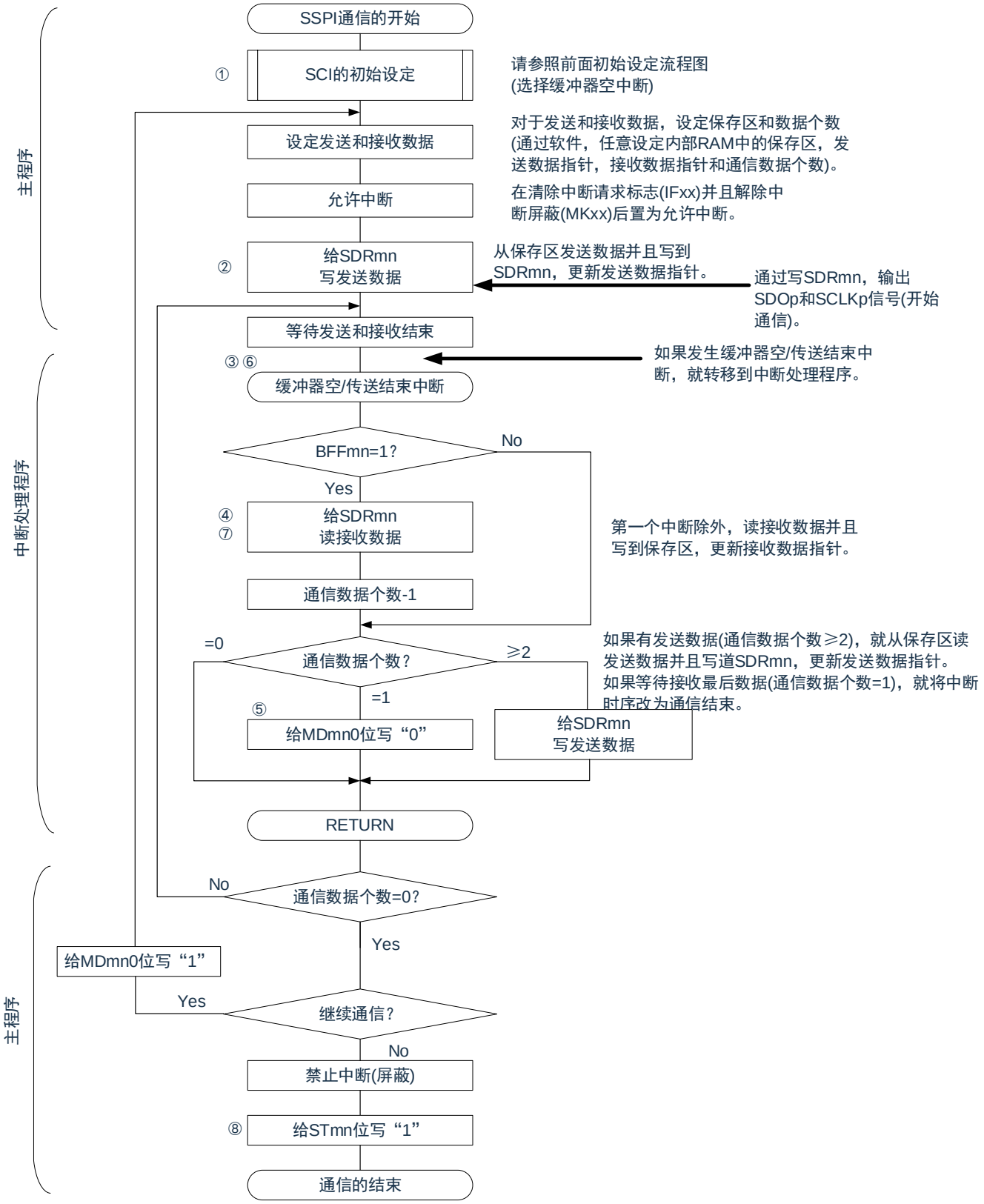
注 2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注：

1. 图中的①~⑧对应“图 16-46 主控发送和接收（连续发送和接收模式）的流程图”中的①~⑧。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-46：主控发送和接收（连续发送和接收模式）的流程图



备注：图中的①~⑧对应“图 16-45 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

16.5.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下 CMS32H6157 微控制器将数据发送到其他设备的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCLK00 SDO00	SCLK01 SDO01	SCLK10 SDO10	SCLK11 SDO11	SCLK20 SDO20	SCLK21 SDO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位~16 位					
传送速率[注]	Max.F_{CLK}/2[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 1：因为在内部对 SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-47：3 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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（00000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送数据寄存器															

SDRmnL

(d) 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 0/1	SOm0 0/1

(e) 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 0/1	SOEm0 0/1

(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 0/1	SSm0 0/1

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在 SSPI 从属发送模式中为固定设定。 ■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-48：从属发送的初始设定步骤

图 16-49：从属发送的中止步骤

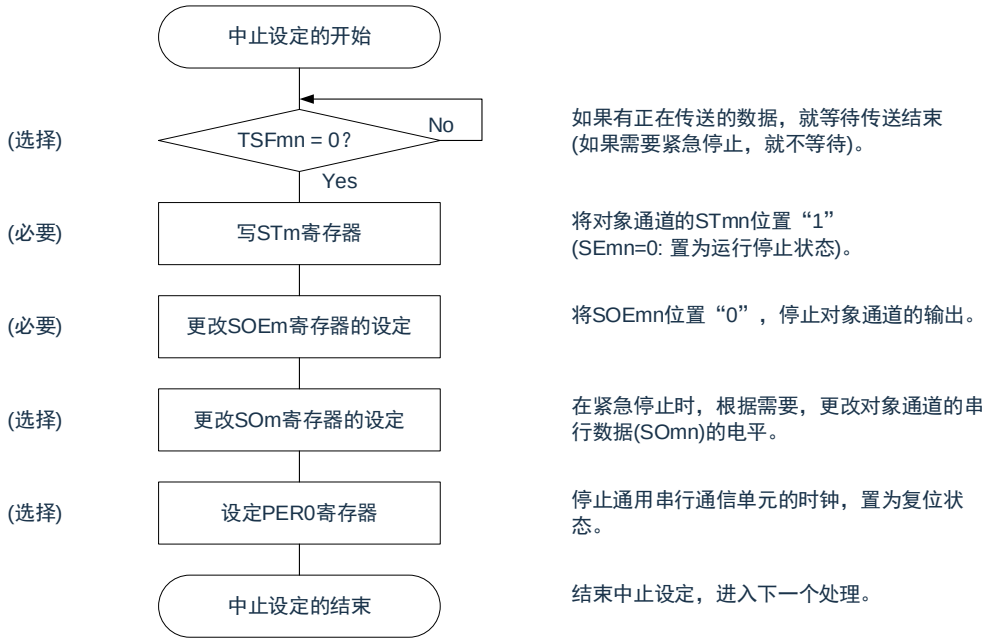
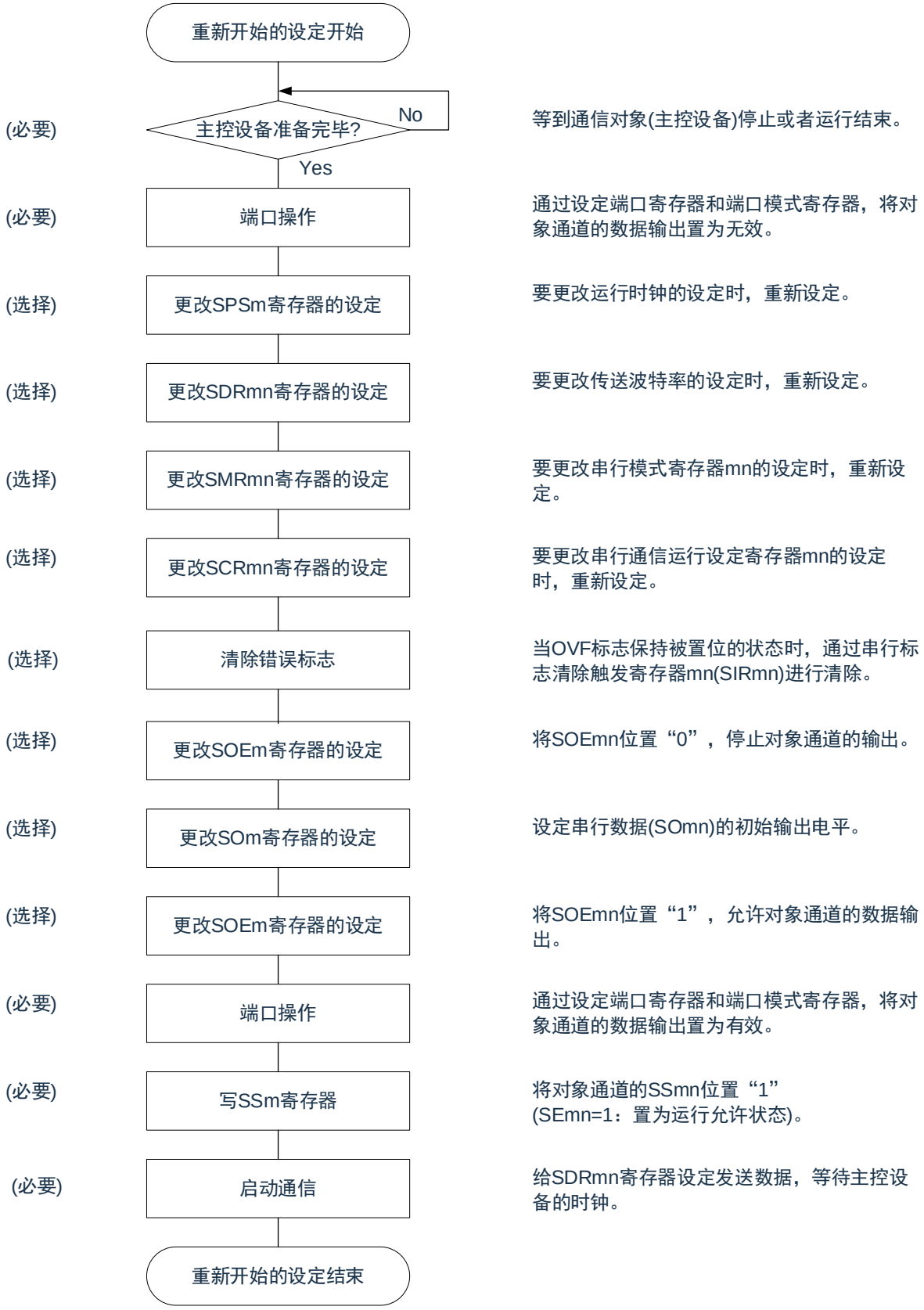


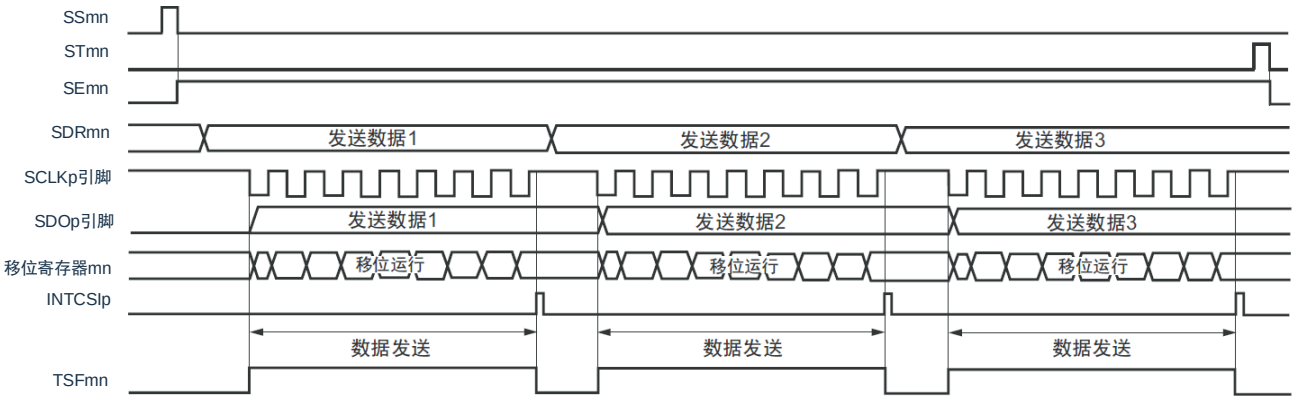
图 16-50：重新开始从属发送的设定步骤



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

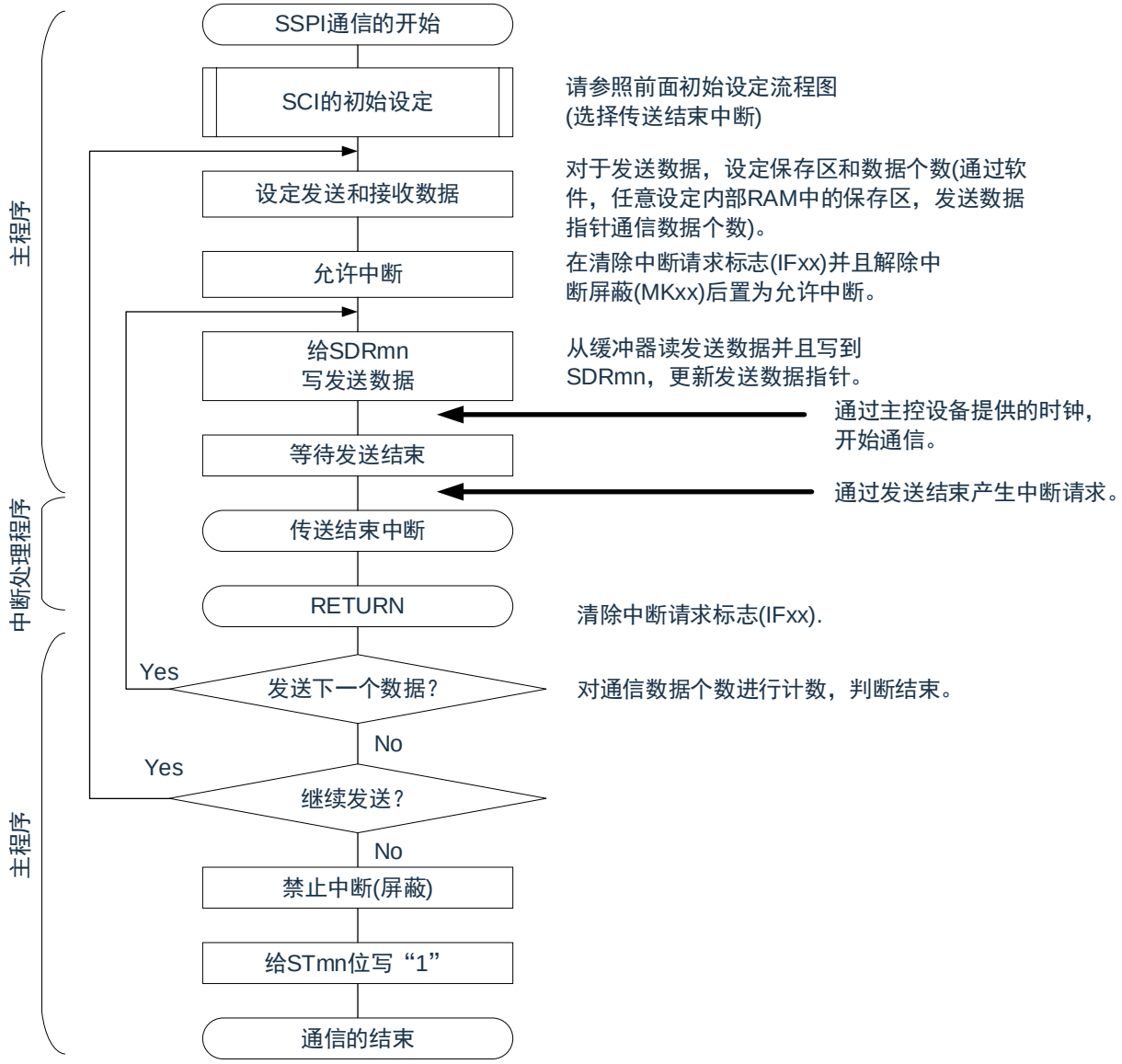
(3) 处理流程（单次发送模式）

图 16-51：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



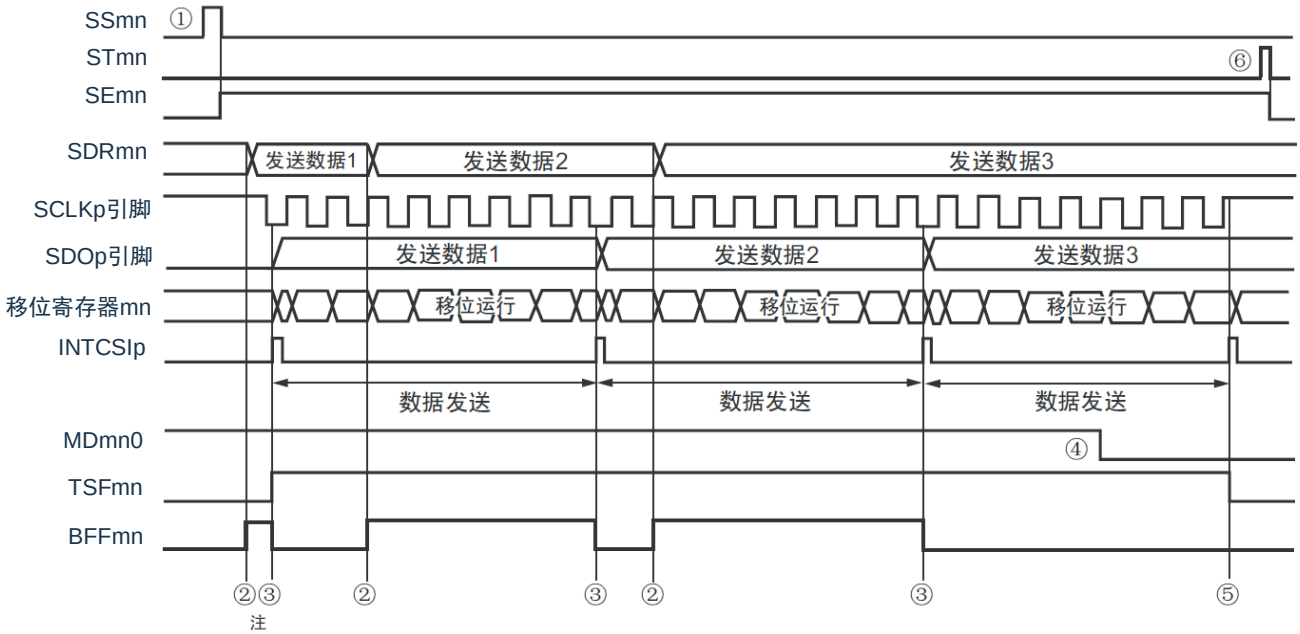
备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-52：从属发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 16-53：从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

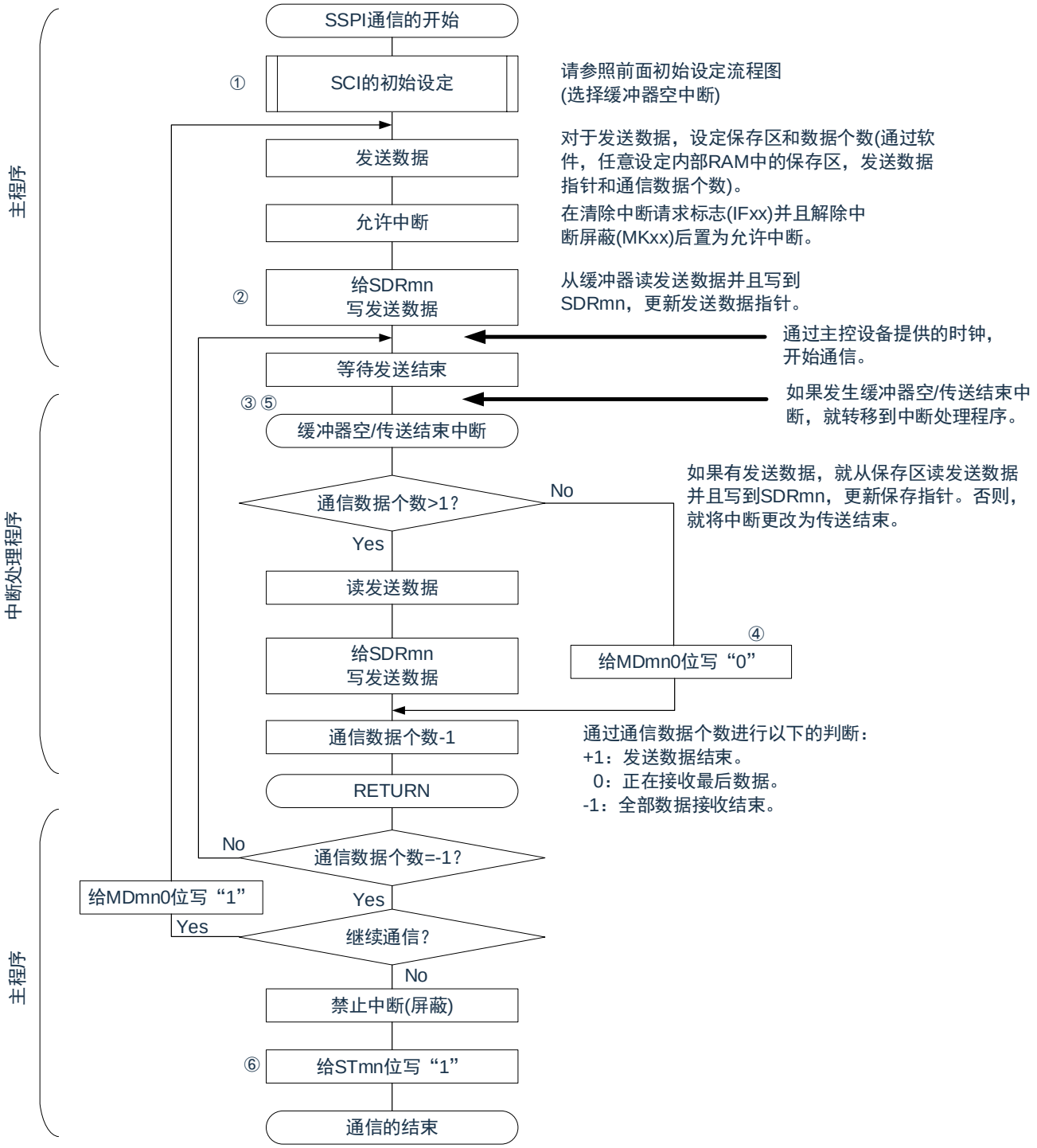


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-54：从属发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 16-53：从属发送（连续发送模式）的时序图”中的①~⑥。

16.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的	SCI0 的	SCI1 的	SCI1 的	SCI2 的	SCI2 的
	通道 0	通道 1	通道 0	通道 1	通道 0	通道 1
使用的引脚	SCLK00 SDI00	SCLK01 SDI01	SCLK10 SDI10	SCLK11 SDI11	SCLK20 SDI20	SCLK21 SDI21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	只限于传送结束中断（禁止设定缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位~16 位					
传送速率	Max.F_{MCK}/6[Hz]注 1,2					
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。					
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB 优先或者 LSB 优先					

注 1：因为在内部对 SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-55：3 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)

	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	0	0	1	0	0	0	0	0

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（0000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	接收数据寄存器															

SDRmnL

(d) 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	×	×

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) …...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	×	×

(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															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在从属接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-56：从属接收的初始设定步骤

图 16-57：从属接收的中止步骤

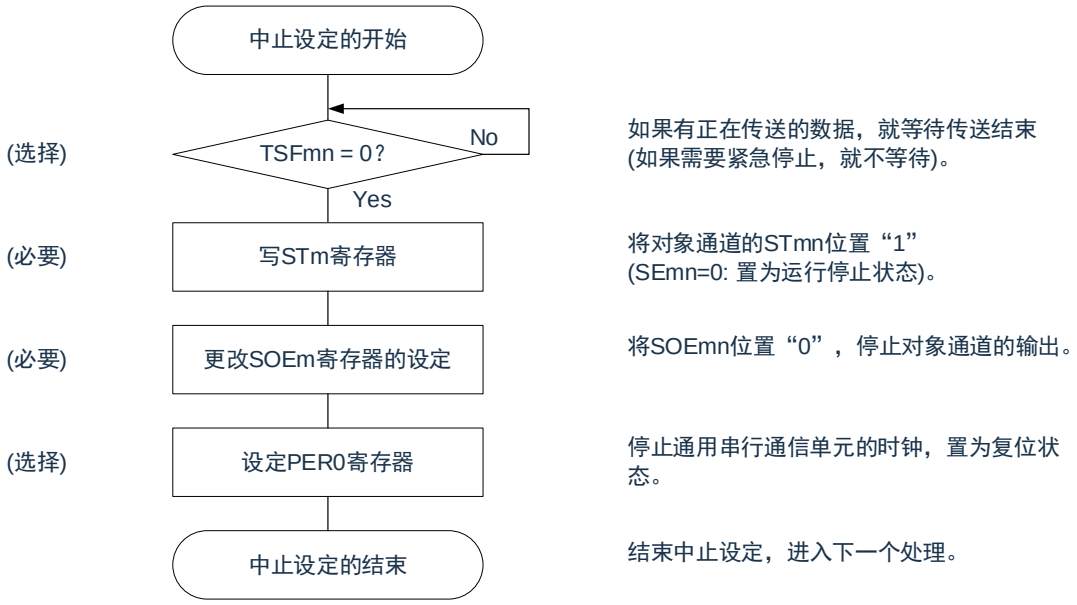
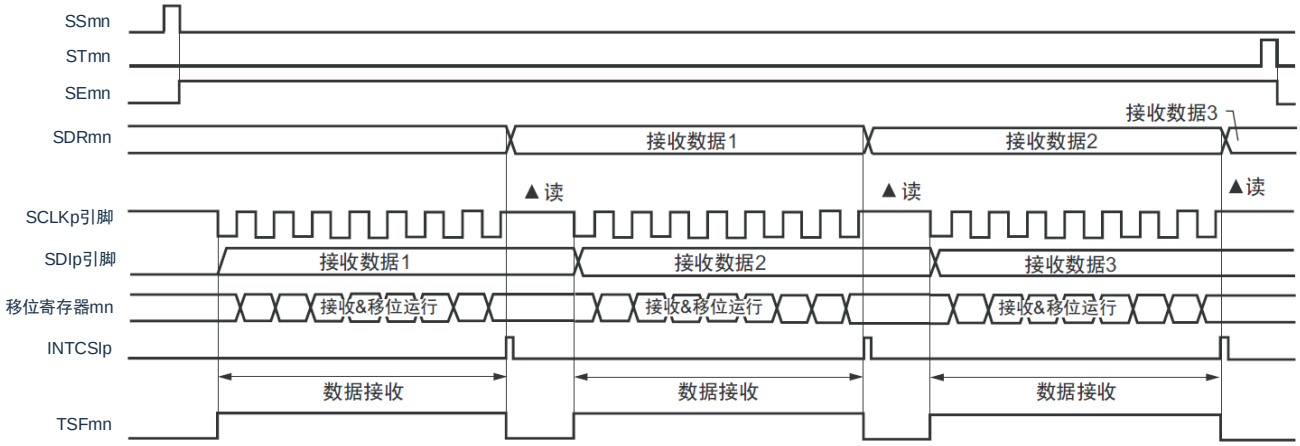


图 16-58：重新开始从属接收的设定步骤

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

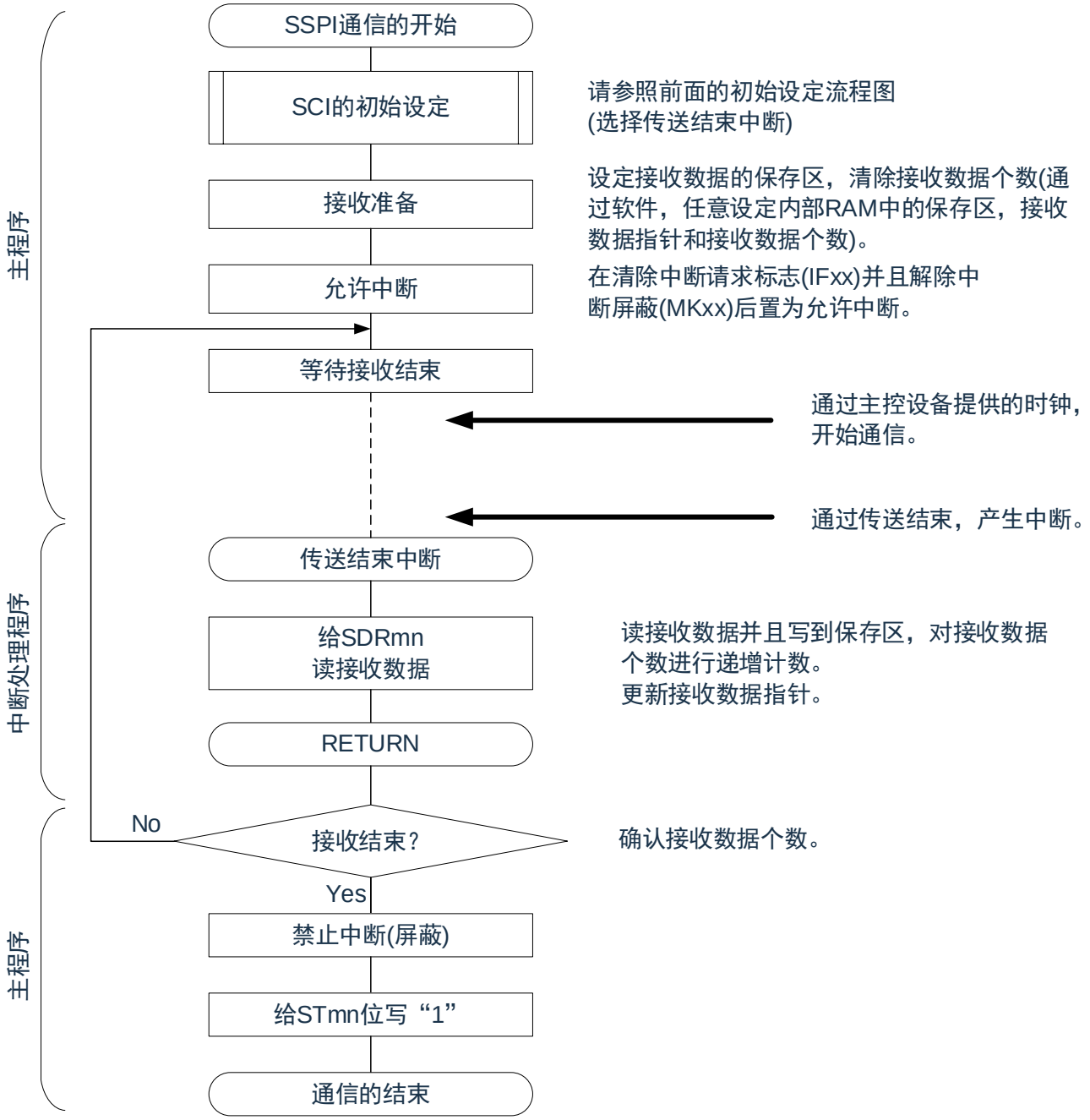
(3) 处理流程（单次接收模式）

图 16-59：从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-60：从属接收（单次接收模式）的流程图



16.5.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品微控制器和其他设备进行数据发送和接收的运行。

3 线串行 I/O	SSPI00	SSPI01	SSPI10	SSPI11	SSPI20	SSPI21
对象通道	SCI0 的	SCI0 的	SCI1 的	SCI1 的	SCI2 的	SCI2 的
	通道 0	通道 1	通道 0	通道 1	通道 0	通道 1
使用的引脚	SCLK00 SDI00 SDO00	SCLK01 SDI01 SDO01	SCLK10 SDI10 SDO10	SCLK11 SDI11 SDO11	SCLK20 SDI20 SDO20	SCLK21 SDI21 SDO21
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	7 位~16 位					
传送速率	Max.F_{MCK}/6[Hz]注 1,2					
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输入/输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输入/输出。					
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相					
数据方向	MSB 优先或者 LSB 优先					

注 1：因为在内部对 SCLK00、SCLK01、SCLK10、SCLK11、SCLK20、SCLK21 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-61：3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）

从属发送和接收时的寄存器设定内容例子

(a) 串行模式寄存器mn(SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（0000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送/接收数据寄存器															

SDRmnL

(d) 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 0/1	SOm0 0/1

(e) 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 0/1	SOEm0 0/1

(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 0/1	SSm0 0/1

注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在 SSPI 从属发送和接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：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 16-62：从属发送和接收的初始设定步骤

注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

图 16-63：从属发送和接收的中止步骤

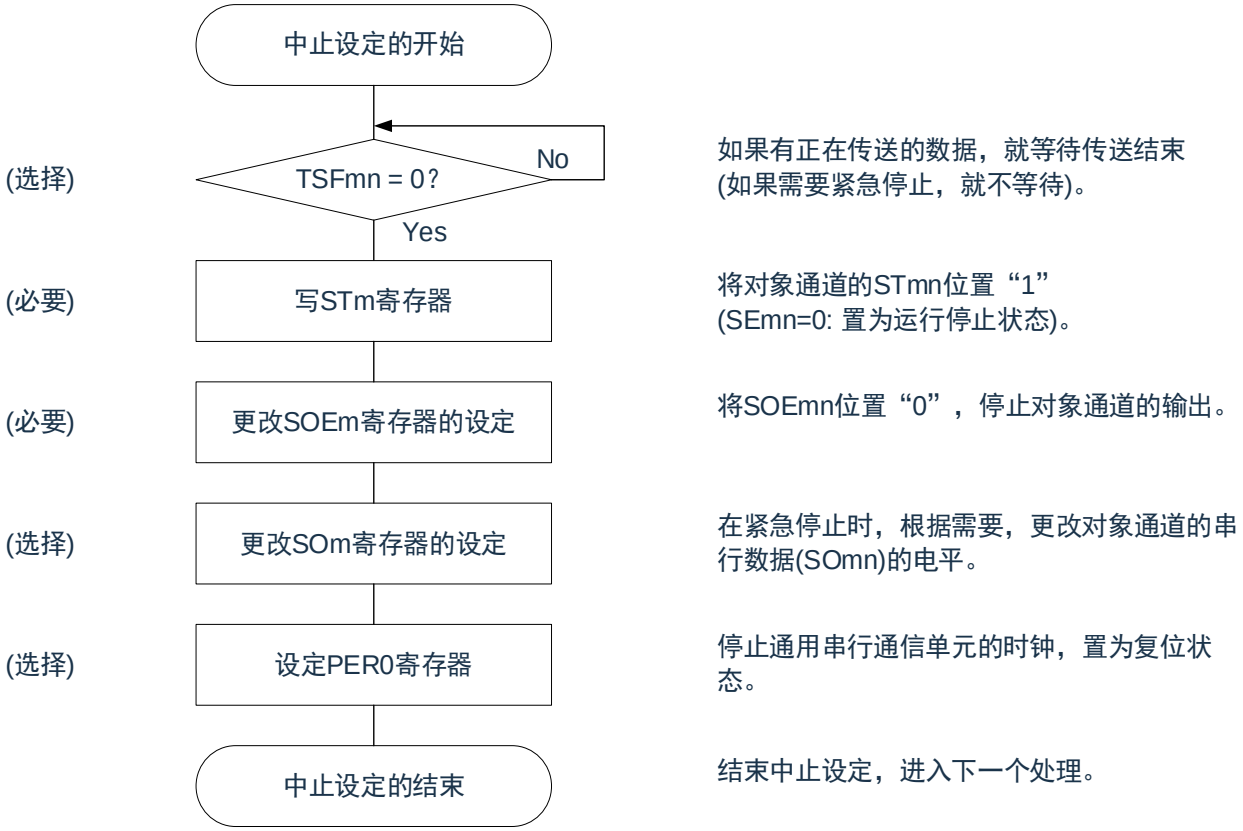
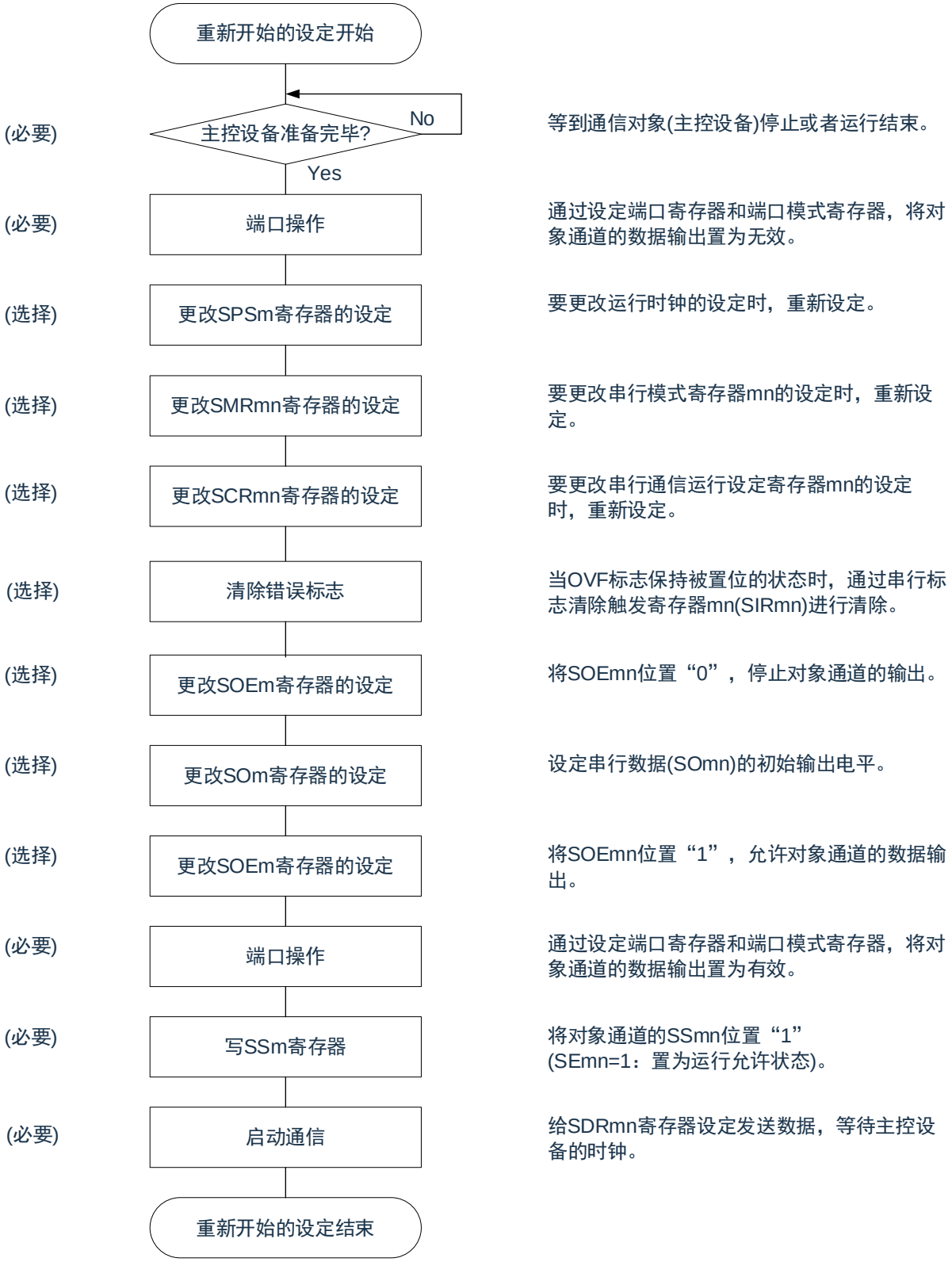


图 16-64：重新开始从属发送和接收的设定步骤

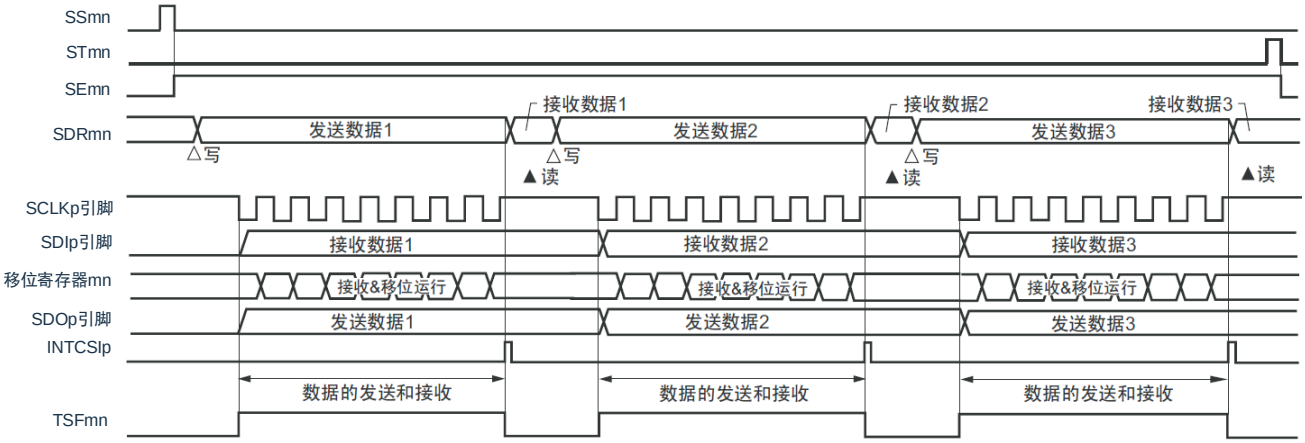


注意：

1. 在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。
2. 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

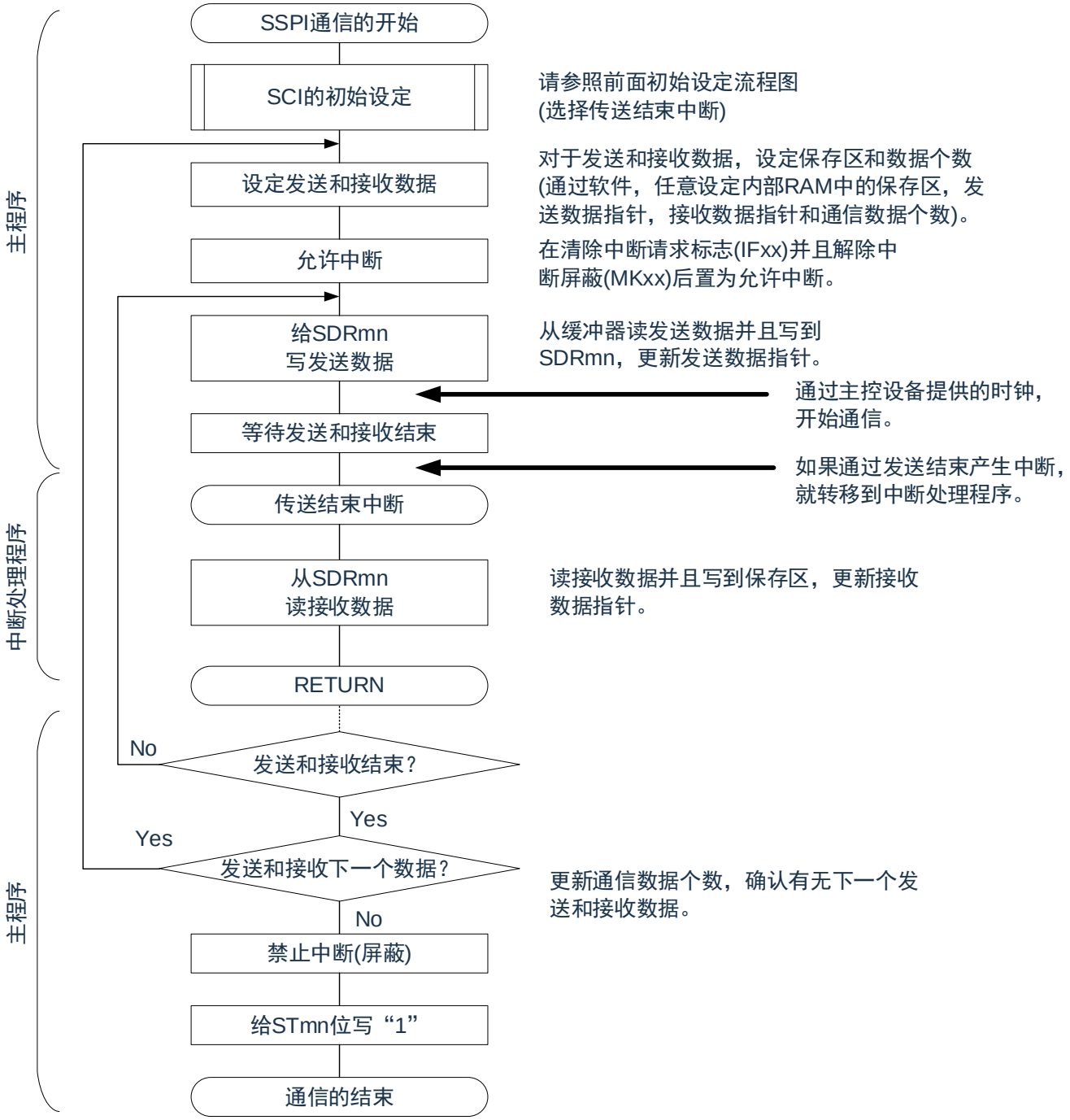
(3) 处理流程（单次发送和接收模式）

图 16-65：从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

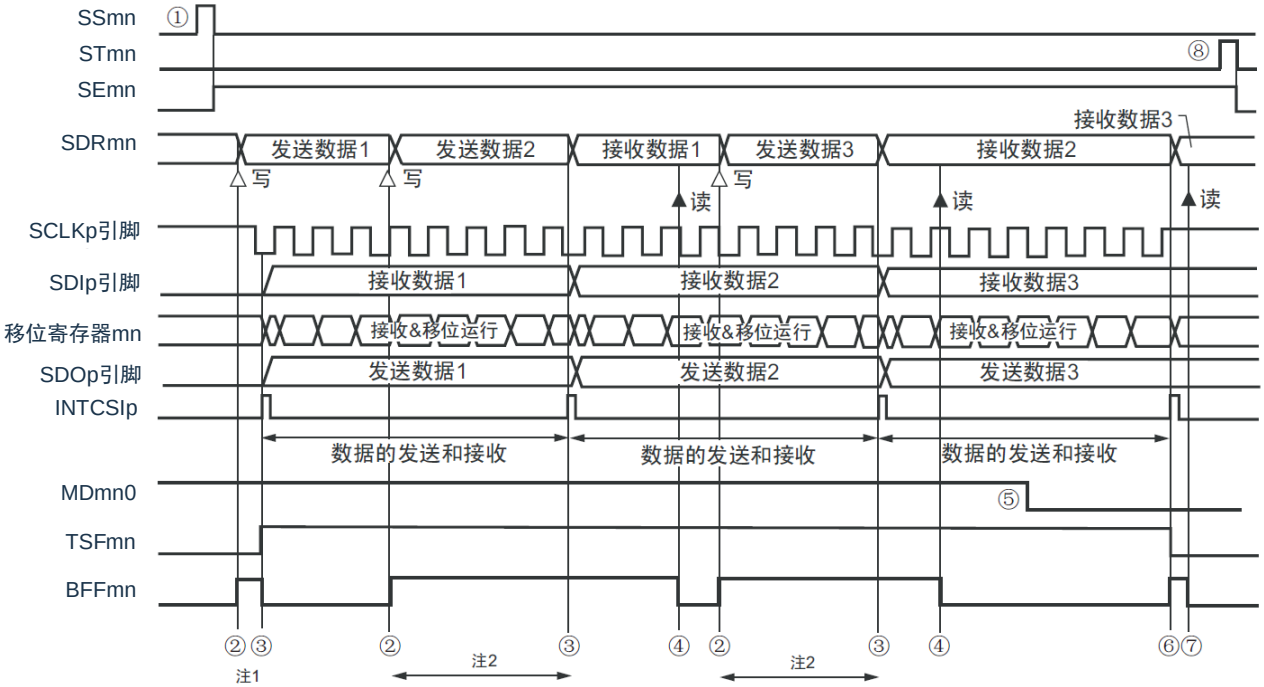
图 16-66：从属发送和接收（单次发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

(4) 处理流程（连续发送和接收模式）

图 16-67：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

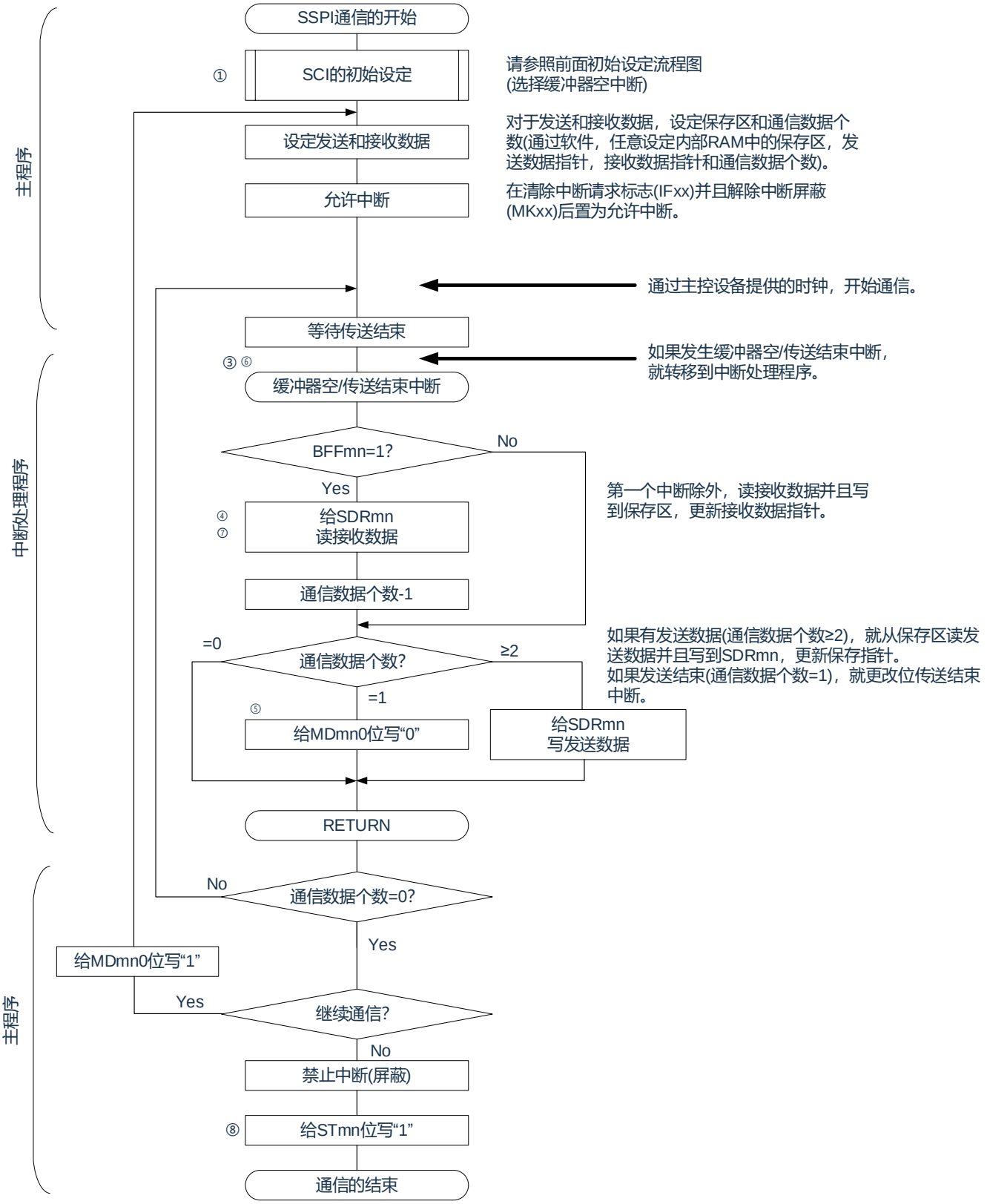
注 2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注：

1. 图中的①~⑧对应“图 16-68：从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-68：从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：图中的①~⑧对应“图 16-67 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

16.5.7 传送时钟频率的计算

3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信的传送时钟频率能用以下计算式进行计算。

(1) 主控设备

（传送时钟频率）={对象通道的运行时钟（F_{MCK}）频率}÷(SDRmn[15:9]+1) ÷2[Hz]

(2) 从属设备

（传送时钟频率）={主控设备提供的串行时钟（SCLK）频率}[注][Hz]

注：容许的最大传送时钟频率为 F_{MCK}/6。

备注：

1. 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。
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）。

表 16-2：3 线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟（F_{MCK}）注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F_{CLK}=32MHz运行时
0	X	X	X	X	0	0	0	0	F_{CLK}	32MHz
	X	X	X	X	0	0	0	1	$F_{CLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$F_{CLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$F_{CLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$F_{CLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$F_{CLK}/2^5$	1MHz
	X	X	X	X	0	1	1	0	$F_{CLK}/2^6$	500KHz
	X	X	X	X	0	1	1	1	$F_{CLK}/2^7$	250KHz
	X	X	X	X	1	0	0	0	$F_{CLK}/2^8$	125KHz
	X	X	X	X	1	0	0	1	$F_{CLK}/2^9$	62.5KHz
	X	X	X	X	1	0	1	0	$F_{CLK}/2^{10}$	31.25KHz
	X	X	X	X	1	0	1	1	$F_{CLK}/2^{11}$	15.63KHz
	X	X	X	X	1	1	0	0	$F_{CLK}/2^{12}$	7.81KHz
	X	X	X	X	1	1	0	1	$F_{CLK}/2^{13}$	3.91KHz
	X	X	X	X	1	1	1	0	$F_{CLK}/2^{14}$	1.95KHz
	X	X	X	X	1	1	1	1	$F_{CLK}/2^{15}$	977Hz
1	0	0	0	0	X	X	X	X	F_{CLK}	32MHz
	0	0	0	1	X	X	X	X	$F_{CLK}/2$	16MHz
	0	0	1	0	X	X	X	X	$F_{CLK}/2^2$	8MHz
	0	0	1	1	X	X	X	X	$F_{CLK}/2^3$	4MHz
	0	1	0	0	X	X	X	X	$F_{CLK}/2^4$	2MHz
	0	1	0	1	X	X	X	X	$F_{CLK}/2^5$	1MHz
	0	1	1	0	X	X	X	X	$F_{CLK}/2^6$	500KHz
	0	1	1	1	X	X	X	X	$F_{CLK}/2^7$	250KHz
	1	0	0	0	X	X	X	X	$F_{CLK}/2^8$	125KHz
	1	0	0	1	X	X	X	X	$F_{CLK}/2^9$	62.5KHz
	1	0	1	0	X	X	X	X	$F_{CLK}/2^{10}$	31.25KHz
	1	0	1	1	X	X	X	X	$F_{CLK}/2^{11}$	15.63KHz
	1	1	0	0	X	X	X	X	$F_{CLK}/2^{12}$	7.81KHz
	1	1	0	1	X	X	X	X	$F_{CLK}/2^{13}$	3.91KHz
	1	1	1	0	X	X	X	X	$F_{CLK}/2^{14}$	1.95KHz
	1	1	1	1	X	X	X	X	$F_{CLK}/2^{15}$	977Hz

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注：

1. X：忽略
2. m：单元号（m=0、1、2）n：通道号（n=0、1）

16.5.8 在3线串行I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（SSPI00、SSPI01、SSPI10、SSPI11、SSPI20、SSPI21）通信过程中发生错误时的处理步骤如图 16-69 所示。

图 16-69：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。 →	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。 →	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SDIRmn 寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）

16.6 从属选择输入功能的时钟同步串行通信的运行

每个通道都是支持从属选择输入功能的时钟同步串行通信的通道。

[数据的发送和接收]

- 7 位~16 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[时钟控制]

- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率[注]从属通信：最大值 $F_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

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

从属选择输入功能有以下 3 种通信运行：

- 从属发送（参照 16.6.1）
- 从属接收（参照 16.6.2）
- 从属的发送和接收（参照 16.6.3）

能通过使用从属选择输入功能，使 1 个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1 个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SDO 引脚的输出。当被选择为通信对象的从属设备时，SDO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SDO 引脚变为高阻。另外，即使输入主控设备的串行时钟也不进行发送和接收。

注意：必须通过端口的操作输出从属选择信号。

图 16-70：从属选择输入功能的结构例子

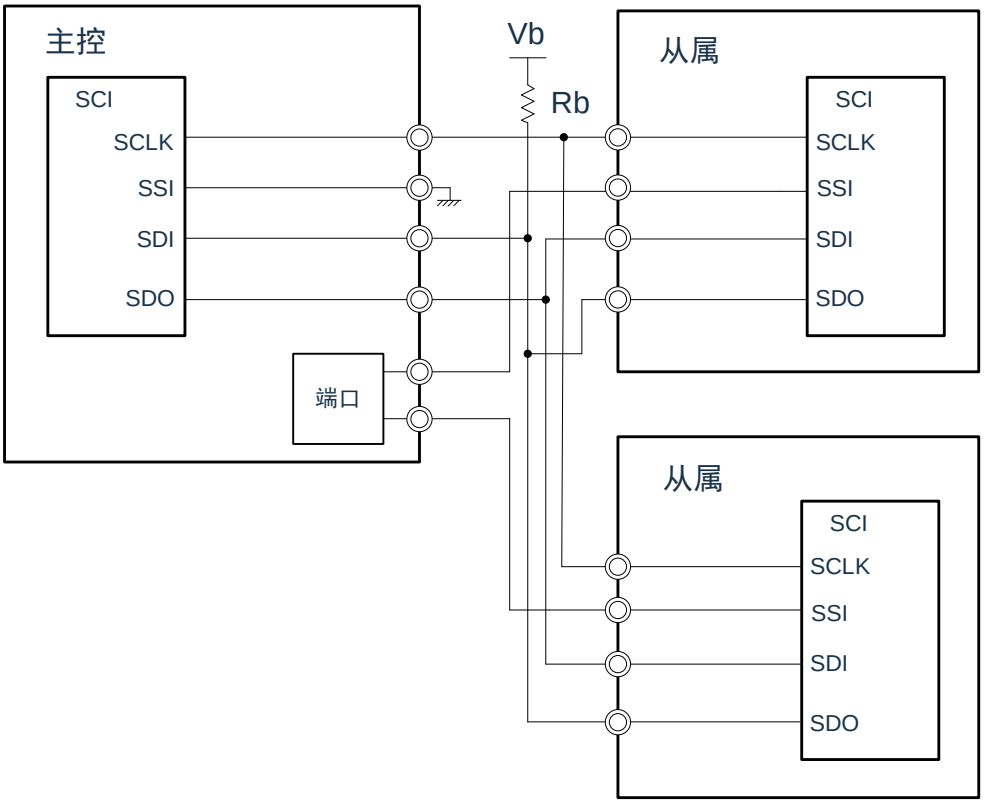
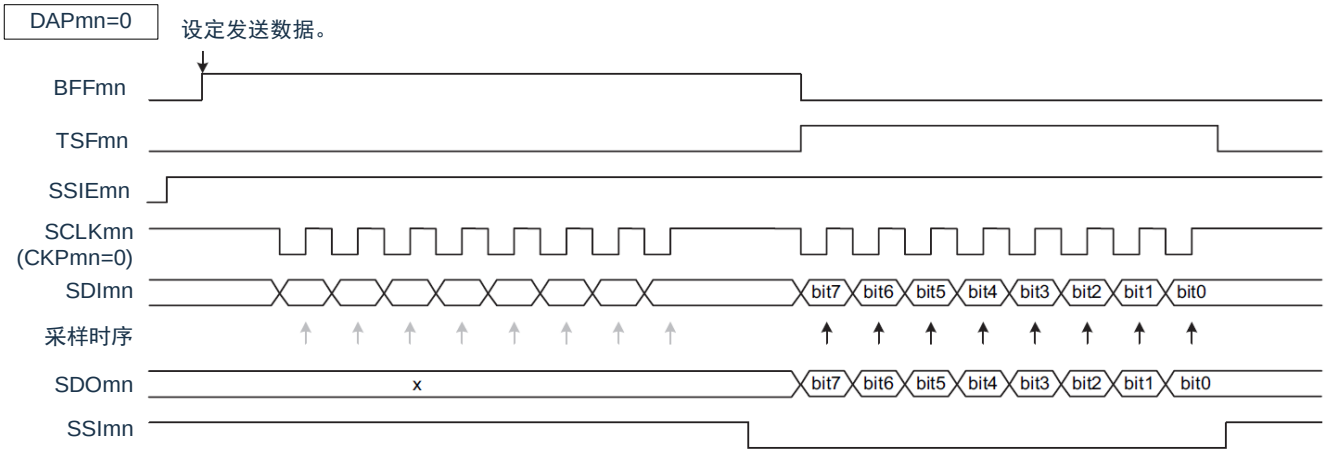
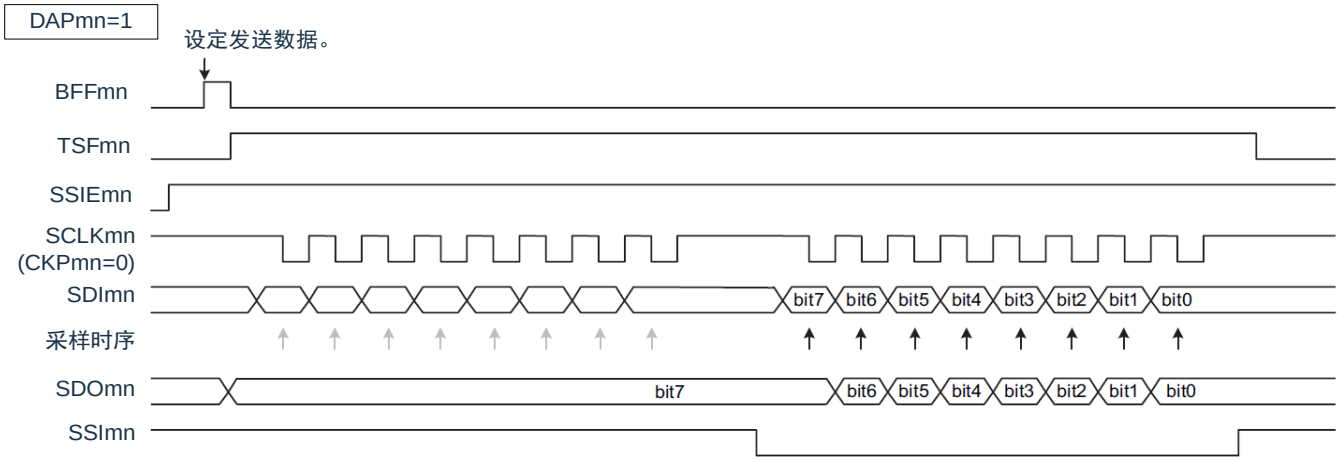


图 16-71：从属选择输入功能的时序图



在 SSImn 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 SSImn 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 SSImn 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 SSImn 变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）

16.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下本产品将数据发送到其他设备的运行。

从属选择输入功能	SSPImn
对象通道	SCIm的通道n
使用的引脚	SCLKmn、SDOmn、SSImn
中断	INTST0、INTSR0、INTST1、INTSR1、INTST2、INTSR2
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位~16位
传送速率	Max.F_{MCK}/6[Hz][注1,2]
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择功能的运行。

注 1：因为在内部对 SCLKmn 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-72：从属选择输入功能（SSPImn）从属发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（0000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送数据寄存器															

SDRmnL

(d) 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

(e) 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

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在 SSPI 从属发送模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

图 16-72：从属选择输入功能（SSPImn）从属发送时的寄存器设定内容例子(2/2)

(f)串行通道开始寄存器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 X	SSm0 0/1

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIEm1 0/1	SSIEm0 0/1

0:SSIm0引脚的输入值无效
1:SSIm0引脚的输入值有效

备注：

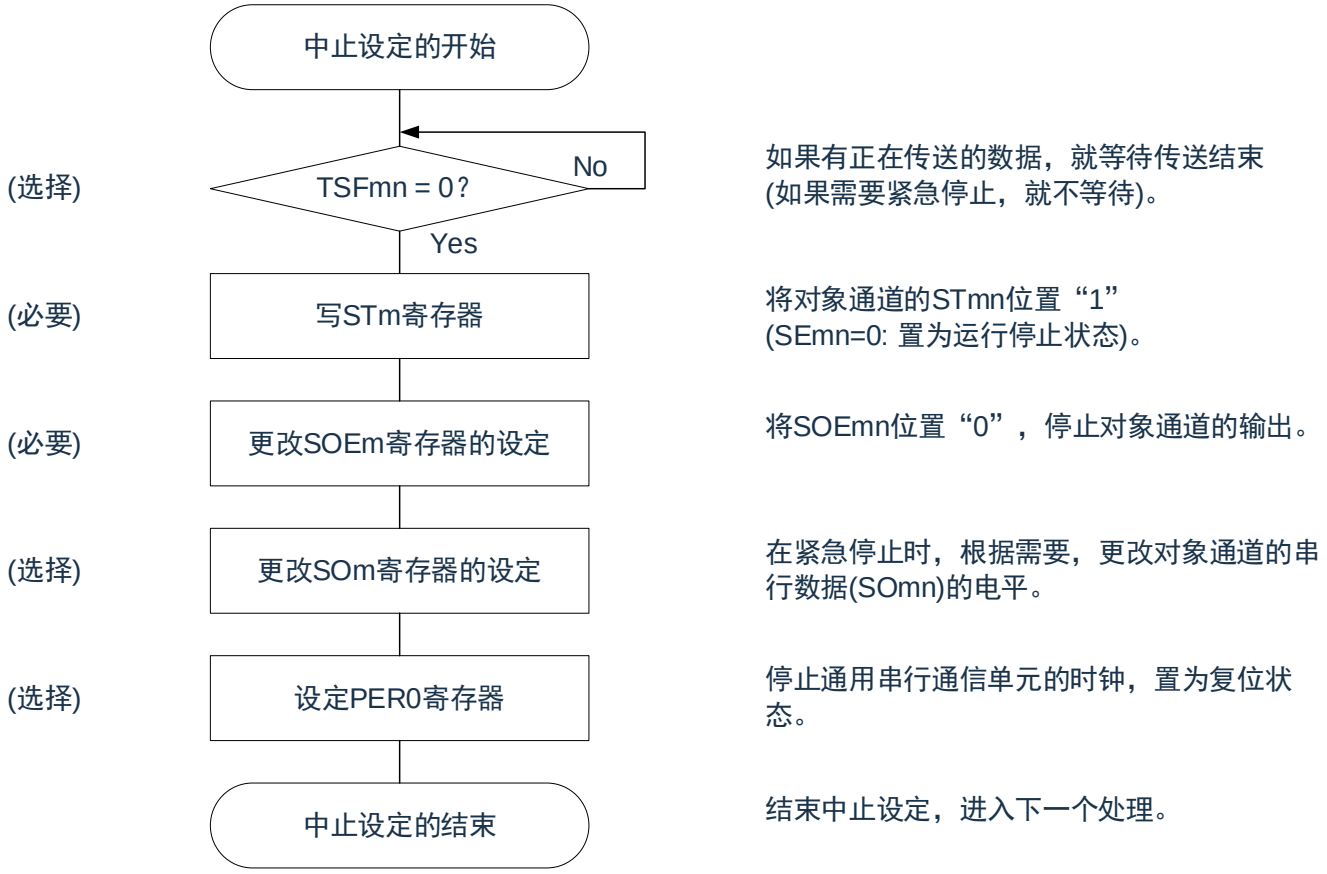
1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在 SSPI 从属发送模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-73：从属发送的初始设定步骤

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-74：从属发送的中止步骤



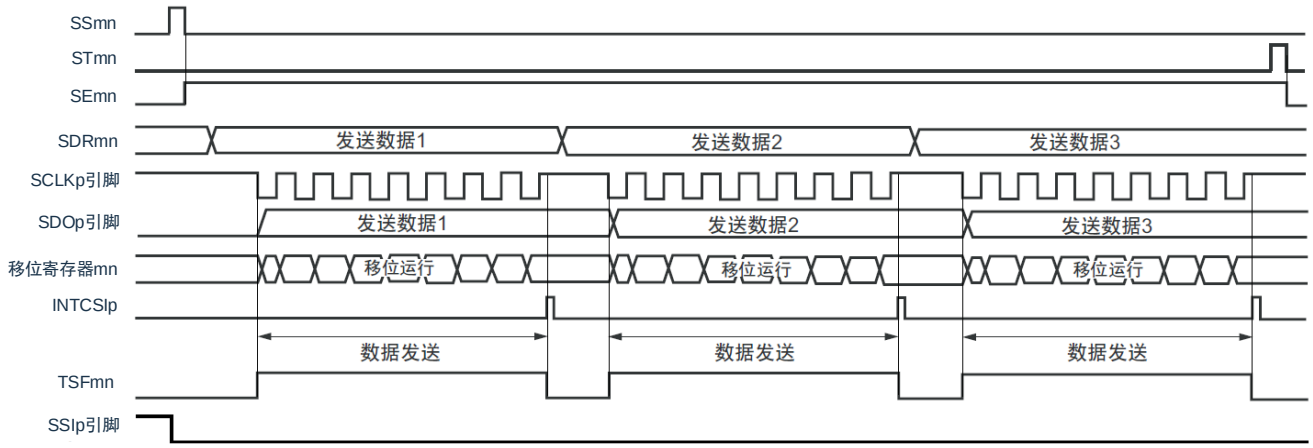
备注：

1. 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-75：重新开始从属发送的设定步骤

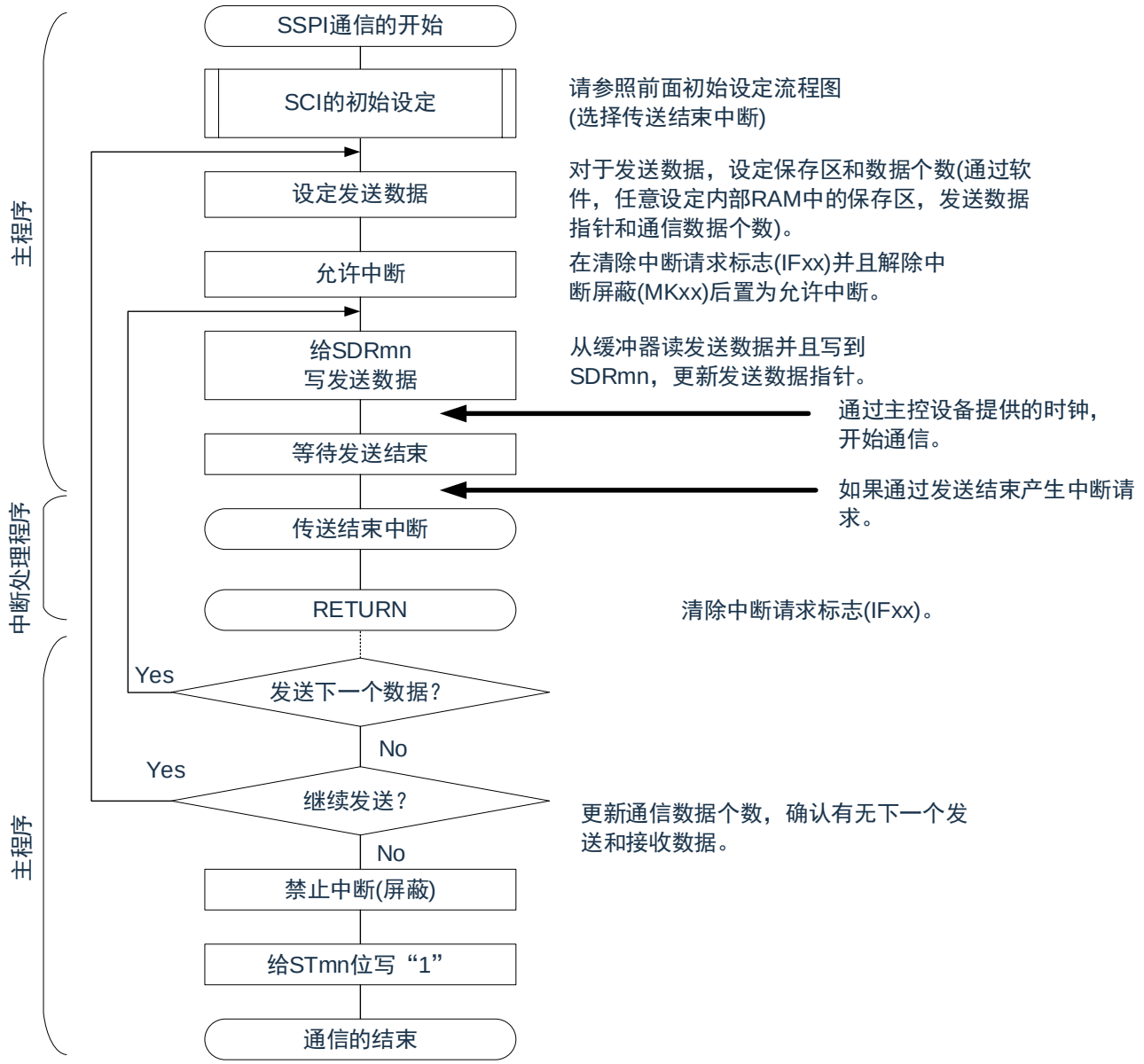
(3) 处理流程（单次发送模式）

图 16-76：从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

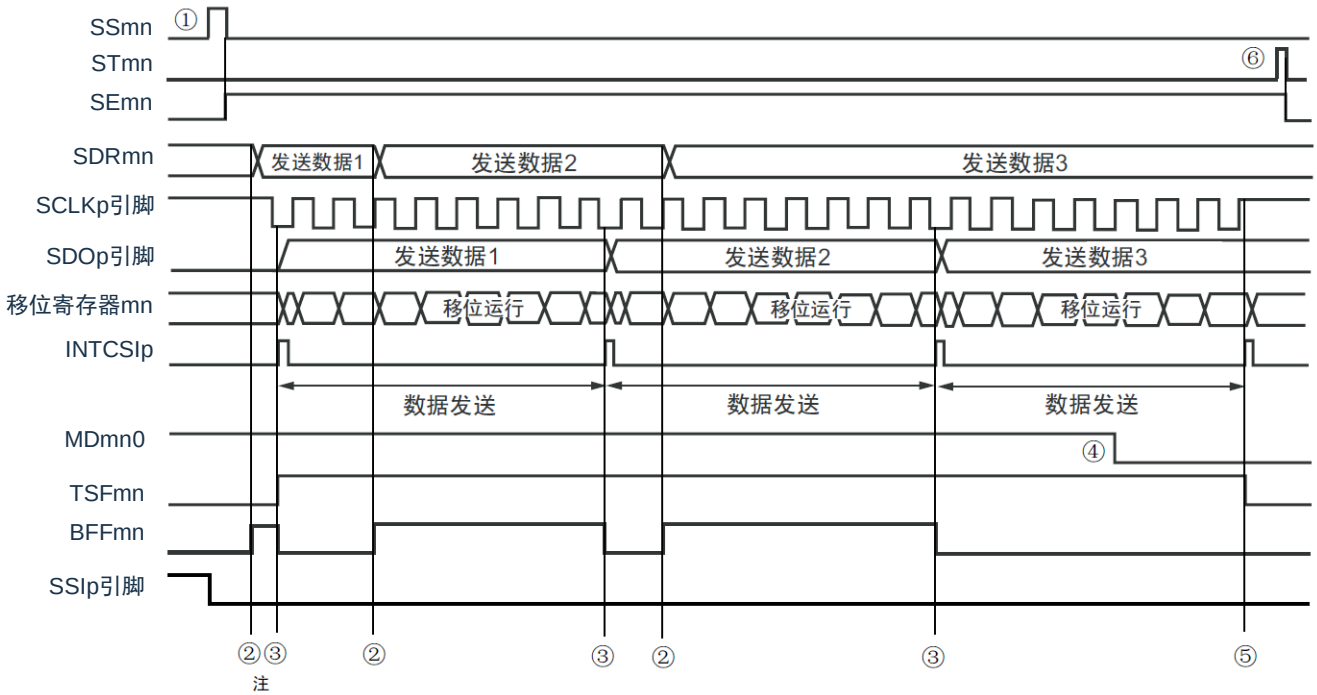
图 16-77：从属发送（单次发送模式）的流程图



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

(4) 处理流程（连续发送模式）

图 16-78：从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

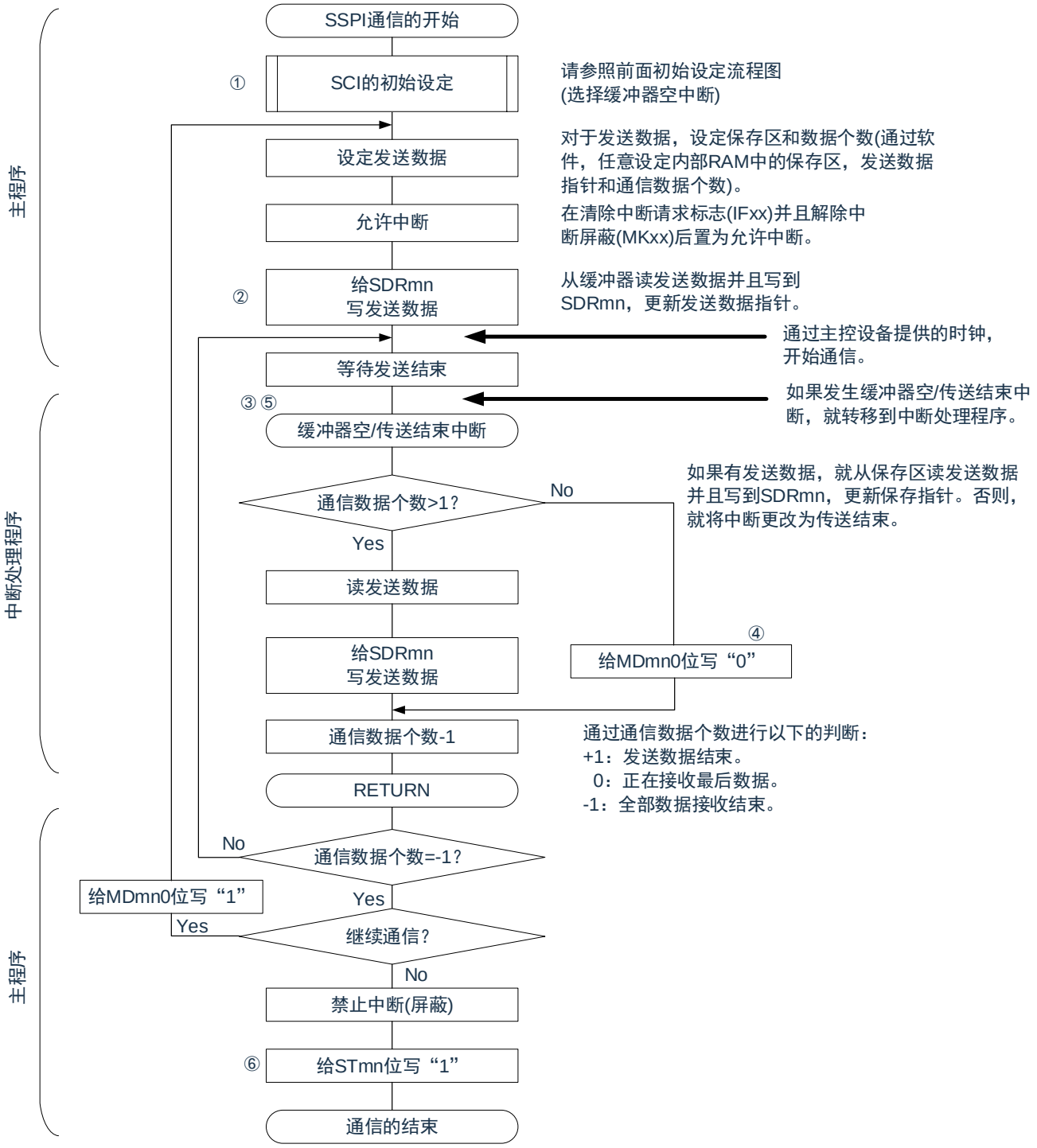


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-79：从属发送（连续发送模式）的流程图



备注：

1. 图中的①~⑥对应“图 16-78 从属发送（连续发送模式）的时序图”中的①~⑥。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

16.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下本产品从其他设备接收数据的运行。

从属选择输入功能	SSPImn
对象通道	SCIm的通道n
使用的引脚	SCLKmn、SDImn、SSImn
中断	INTST0、INTSR0、INTST1、INTSR1、INTST2、INTSR2
	只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位~16位
传送速率	Max.F_{MCK}/6[Hz]注1,2
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1：因为在内部对 SCLK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-80：从属选择输入功能（SSPImn）从属接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

	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	0	0	1	0	0	0	0	0

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn			PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0	DLSmn3	DLSmn2	DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0/1	0/1	0/1	0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（0000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	接收数据寄存器															

SDRmnL

(d) 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	×	×

(e) 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	×	×

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在从属接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

图 16-81：从属选择输入功能（SSPImn）从属接收时的寄存器设定内容例子(2/2)

(f)串行通道开始寄存器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 X	SSm0 0/1

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIEm1 0/1	SSIEm0 0/1

0:SSIm0引脚的输入值无效
1:SSIm0引脚的输入值有效

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在从属接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-82：从属接收的初始设定步骤

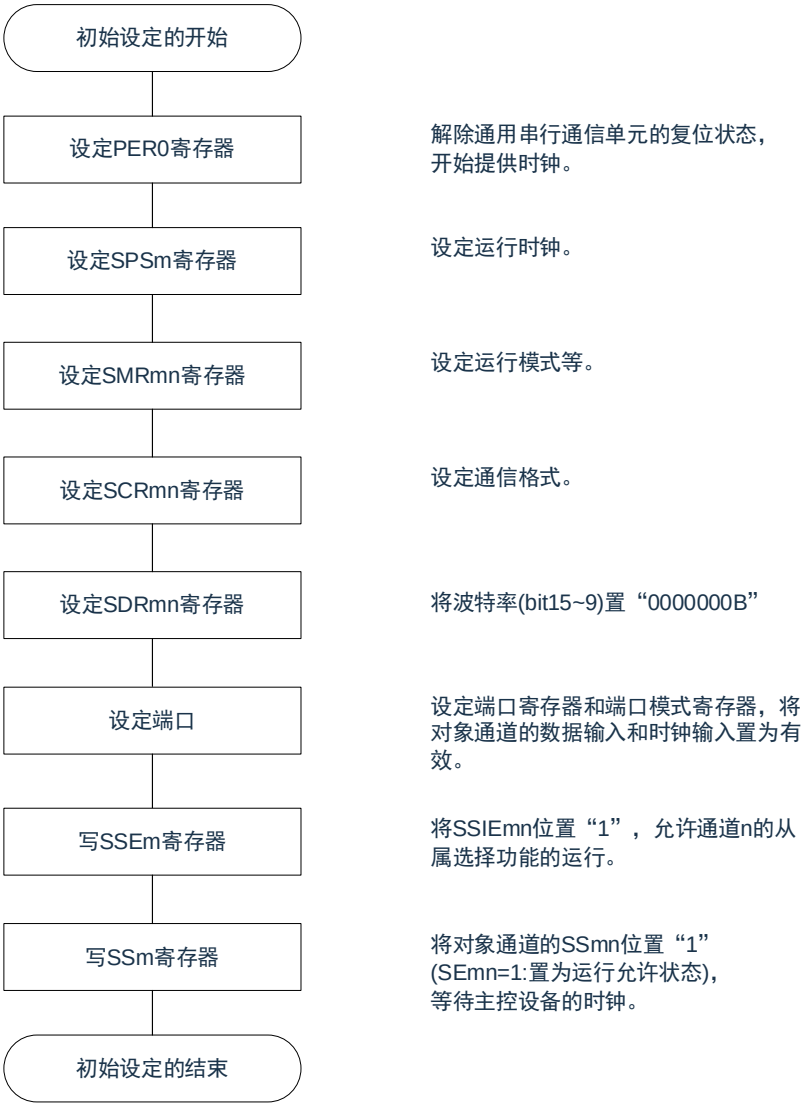


图 16-83：从属接收的中止步骤

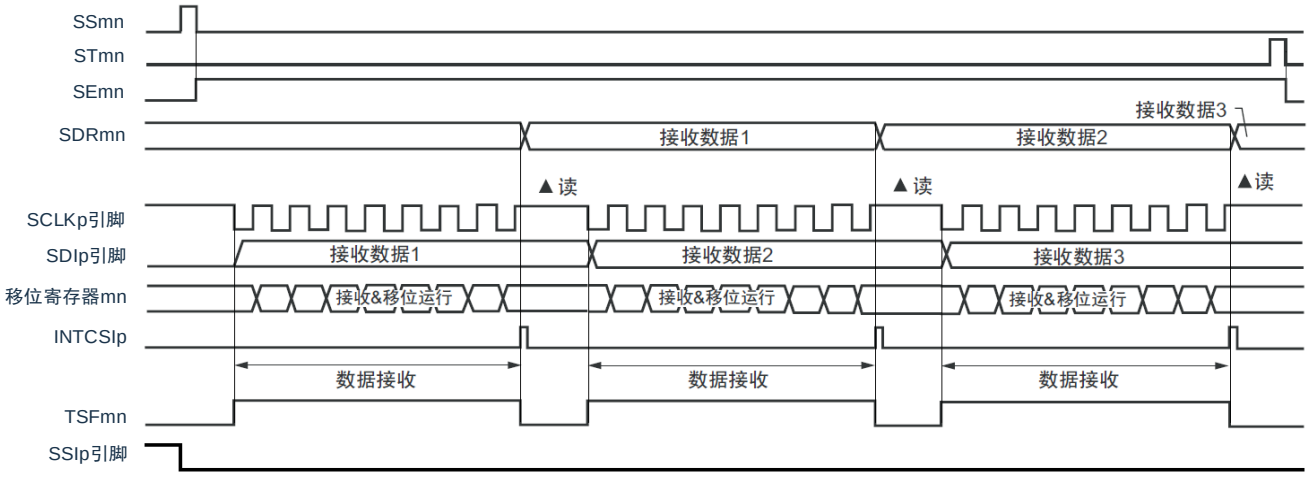
备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-84：重新开始从属接收的设定步骤

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

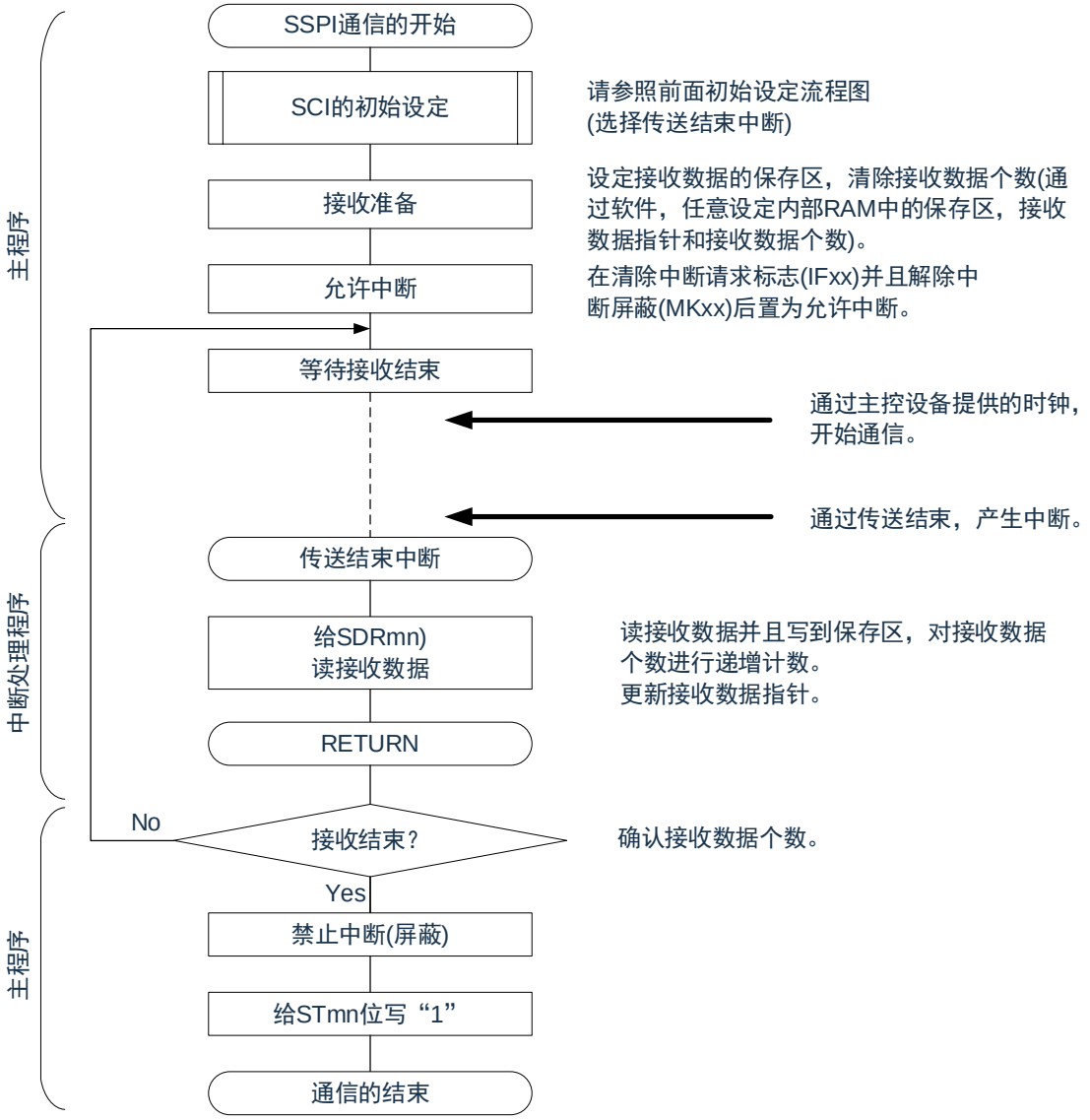
(3) 处理流程（单次接收模式）

图 16-85：从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-86：从属接收（单次接收模式）的流程图



16.6.3 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下本产品和其他设备进行数据发送和接收的运行。

从属选择输入功能	SSPImn
对象通道	SCIm的通道n
使用的引脚	SCLKmn、SDOmn、SDImn、SSImn
中断	INTST0、INTSR0、INTST1、INTSR1、INTST2、INTSR2
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7位~16位
传送速率	Max.F_{MCK}/6[Hz]注1,2
数据相位	能通过SCRmn寄存器的DAPmn位进行选择。 • DAPmn=0：在串行时钟开始运行时，开始数据输出。 • DAPmn=1：在串行时钟开始运行的半个时钟前，开始数据输出。
时钟相位	能通过SCRmn寄存器的CKPmn位进行选择。 • CKPmn=0：正相 • CKPmn=1：反相
数据方向	MSB优先或者LSB优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1：因为在内部对 SCLKmn 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为 F_{MCK}/6[Hz]。

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

备注：

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

(1) 寄存器的设定

图 16-87：从属选择输入功能（SSPImn）从属发送和接收时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

数据和时钟的相位选择(详细内容请参照“控制通用串行通信单元的寄存器”)

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

数据长度的设定
DLSmn3~0：7位~16位数据长度选择

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定（0000000B）							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送/接收数据寄存器															

SDRmnL

(d) 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

(e) 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

注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在从属接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

图 16-87：从属选择输入功能（SSPImn）从属发送和接收时的寄存器设定内容例子(2/2)

(f)串行通道开始寄存器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 X	SSm0 0/1

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSIEm1 0/1	SSIEm0 0/1

0:SSIm0引脚的输入值无效
1:SSIm0引脚的输入值有效

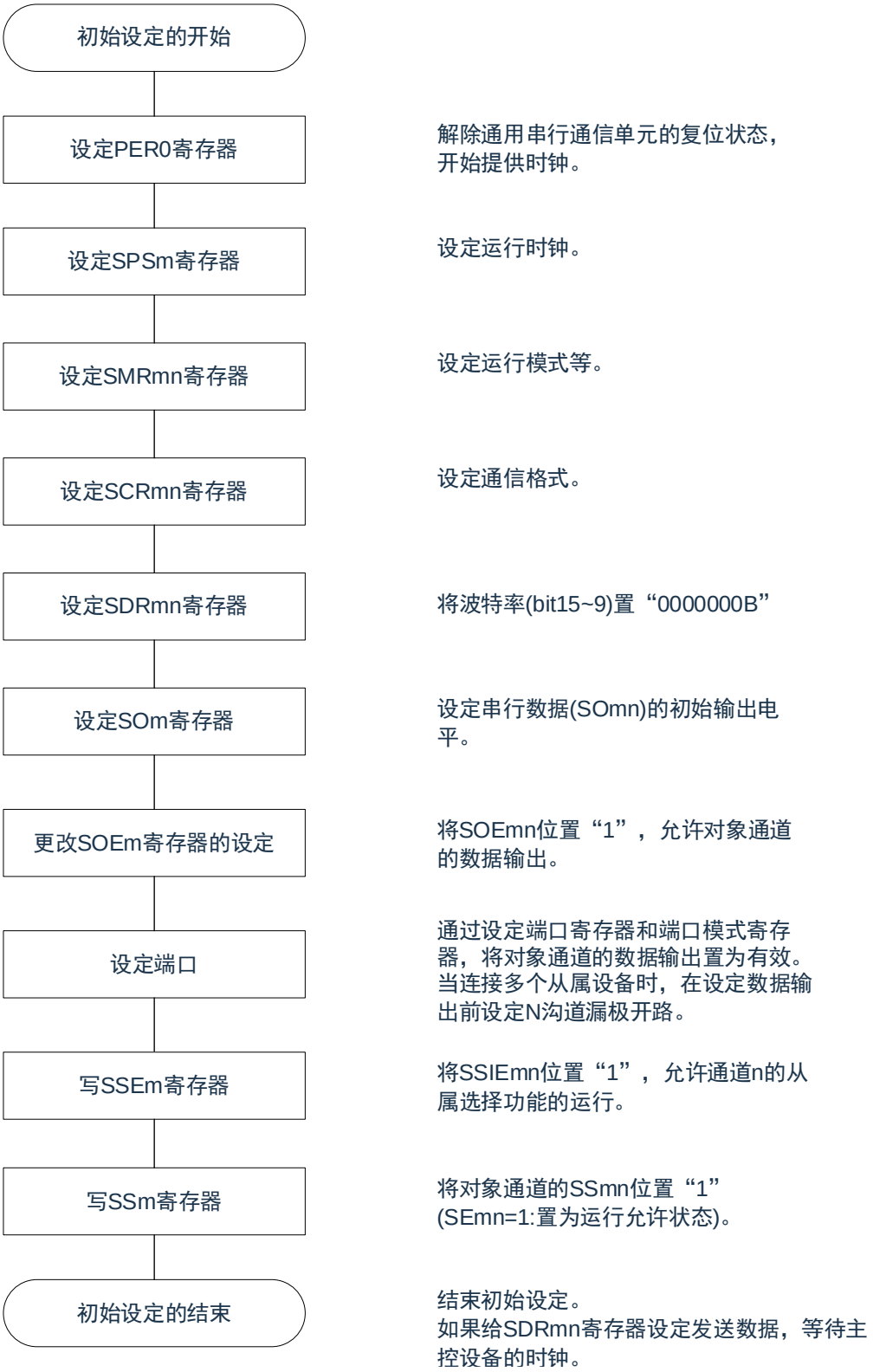
注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）
2. □：在从属接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

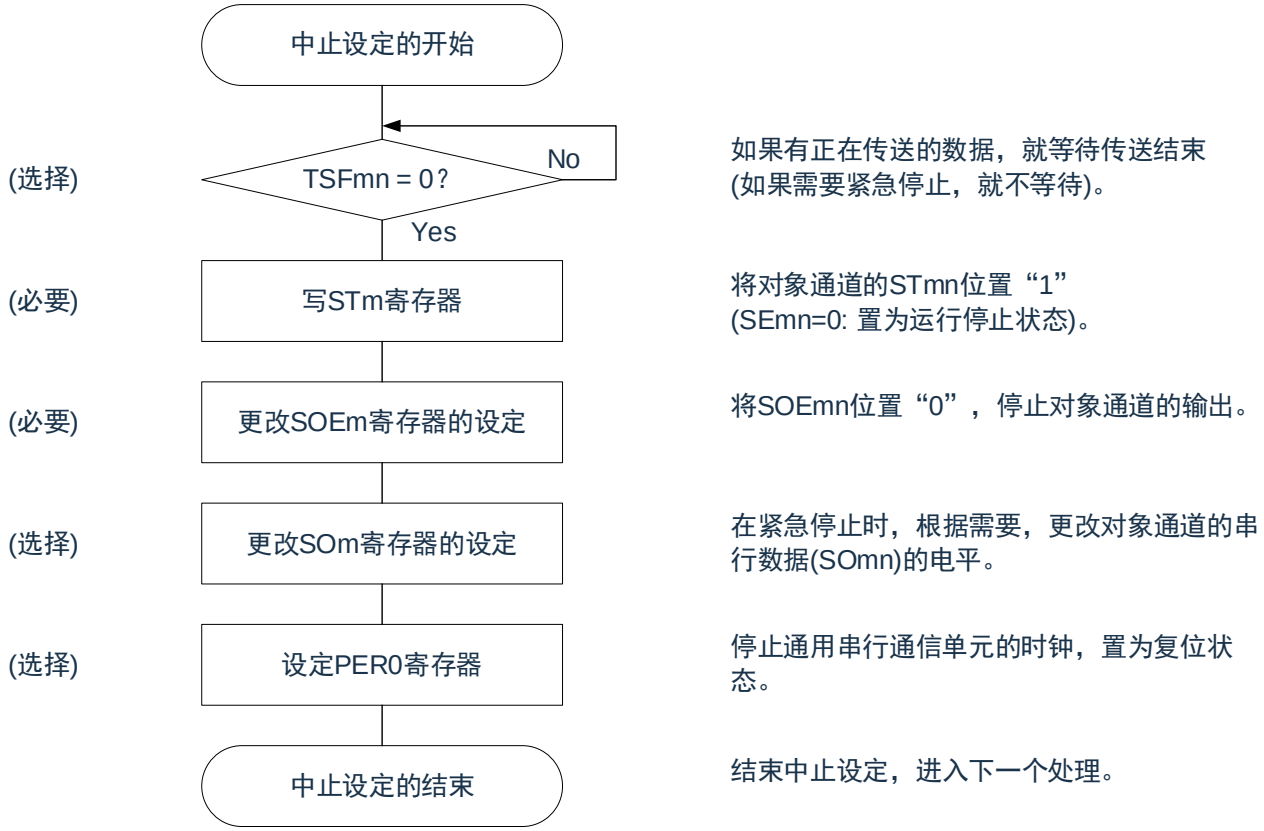
图 16-88：从属发送和接收的初始设定步骤



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-89：从属发送和接收的中止步骤



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

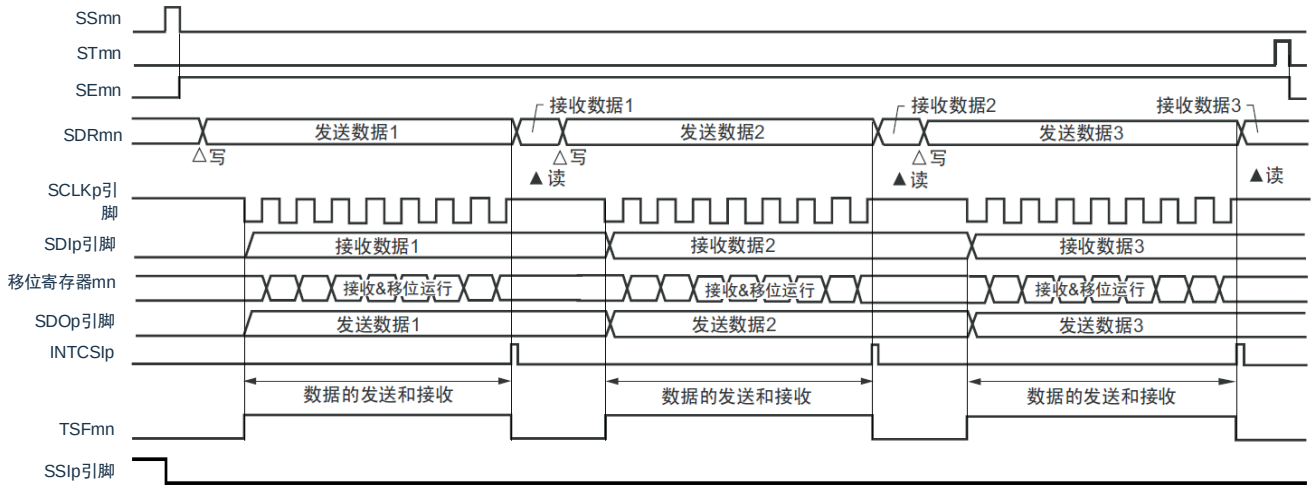
图 16-90：重新开始从属发送和接收的设定步骤

注意：

1. 在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。
2. 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

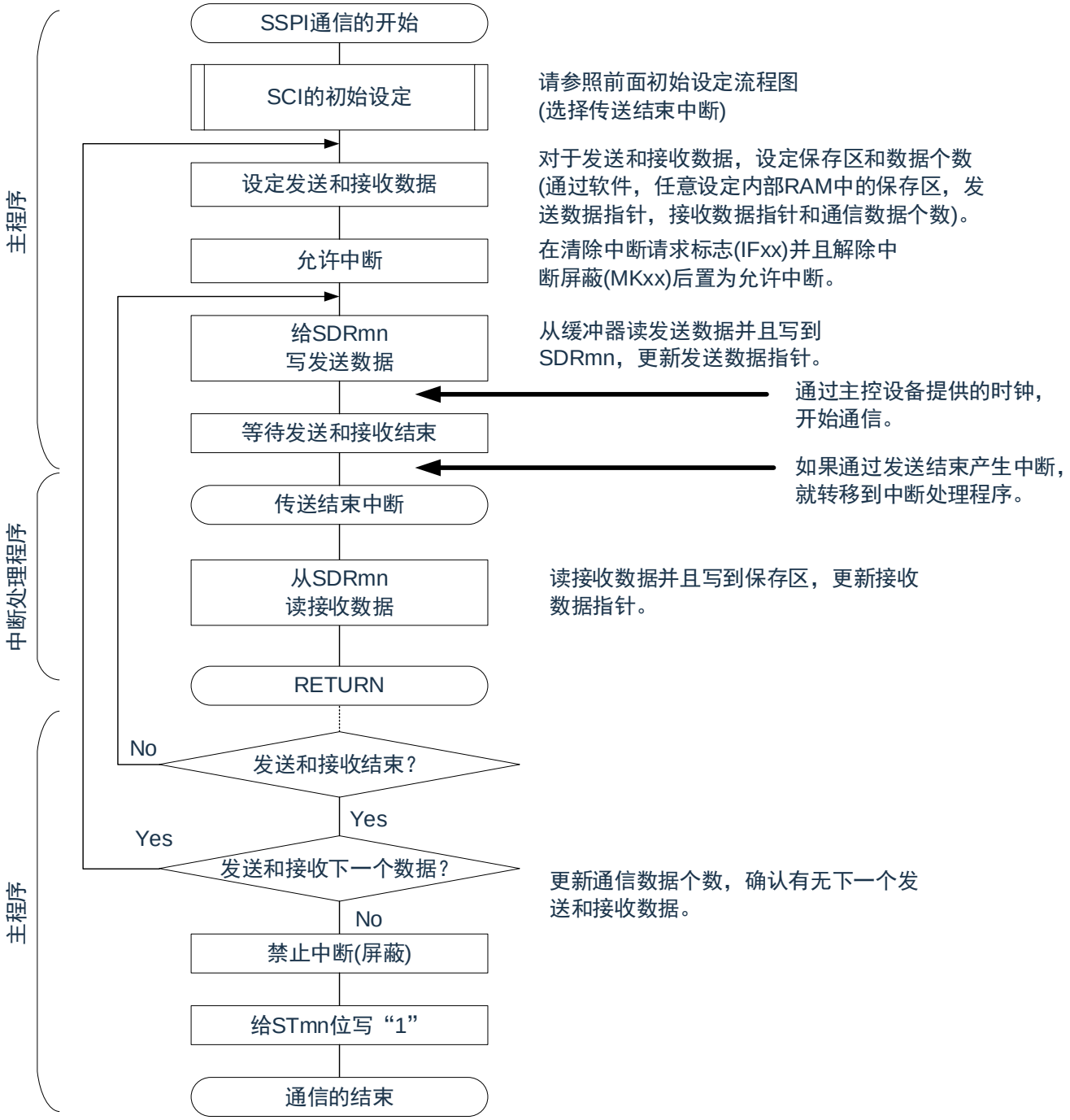
(3) 处理流程（单次发送和接收模式）

图 16-91：从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-92：从属发送和接收（单次发送和接收模式）的流程图

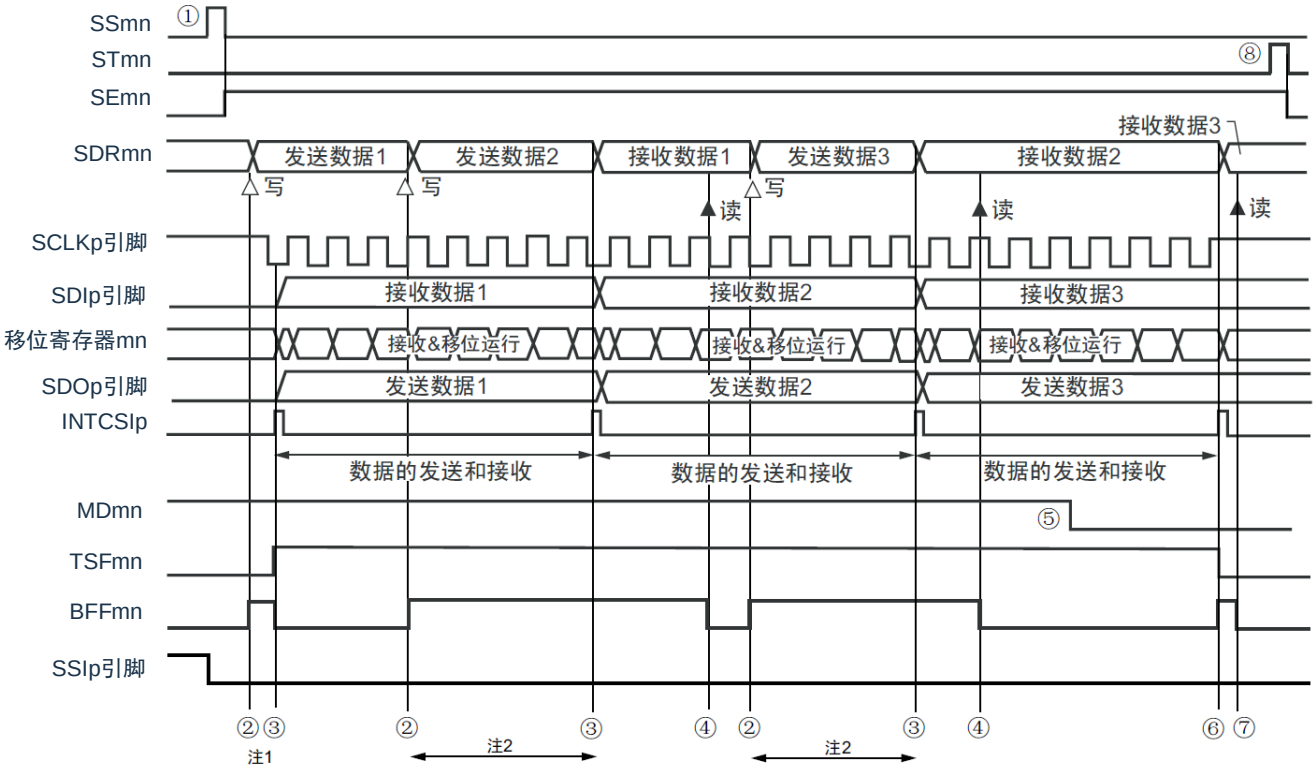


注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

(4) 处理流程（连续发送和接收模式）

图 16-93：从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

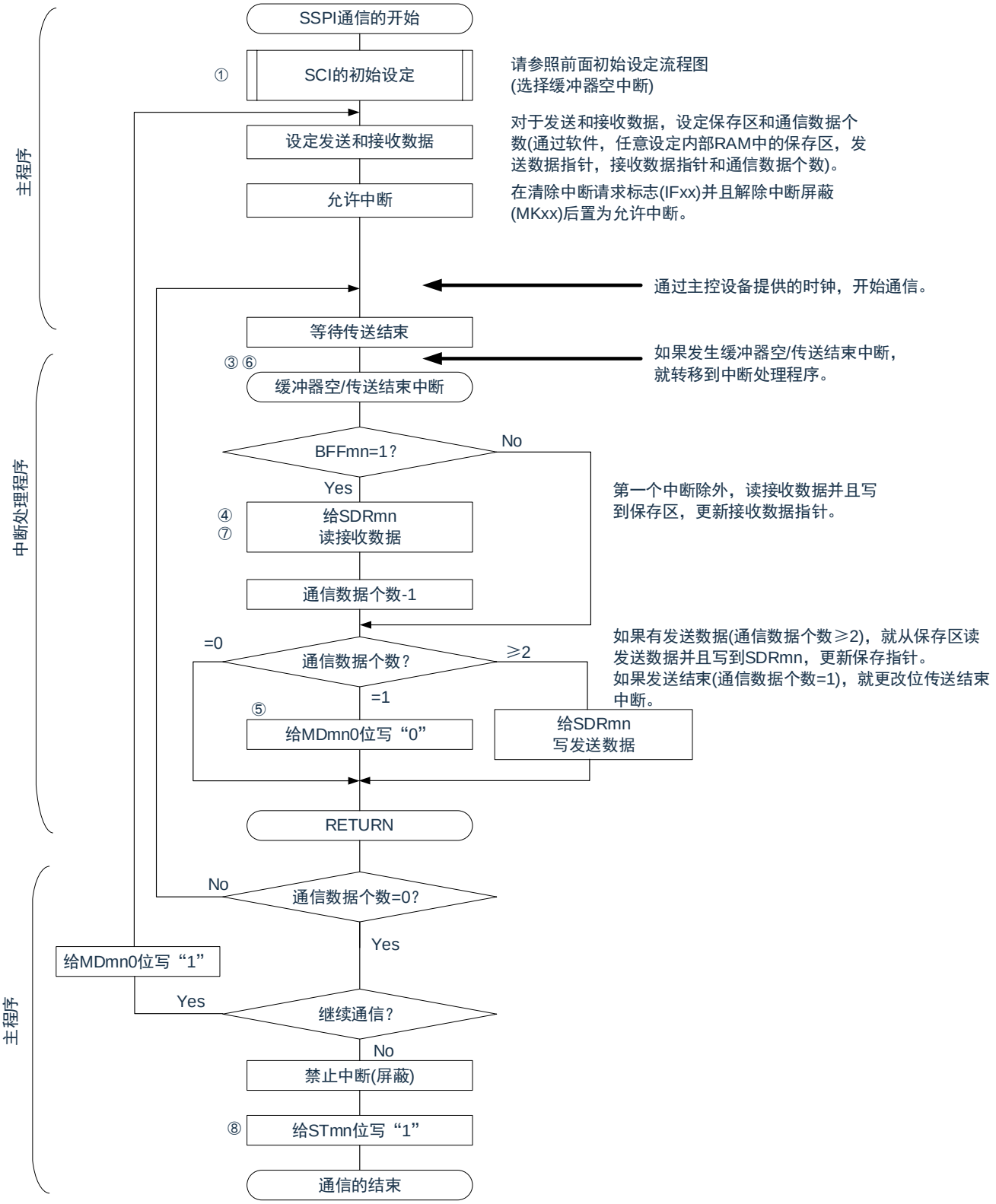
注 2：如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注：

1. 图中的①~⑧对应“图 16-94 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

图 16-94：从属发送和接收（连续发送和接收模式）的流程图



注意：在主控设备开始输出时钟前，必须给 SDRmn 寄存器设定发送数据。

备注：

1. 图中的①~⑧对应“图 16-93 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）p：SSPI 号（p=00、01、10、11、20、21）

16.6.4 传送时钟频率的计算

从属选择输入功能（SSPImn）通信的传送时钟频率能用以下计算式进行计算。

(1) 从属设备

（传送时钟频率）={主控设备提供的串行时钟（SCLK）频率}[注][Hz]

注：容许的最大传送时钟频率为 $F_{MCK}/6$。

表 16-3：从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟（F_{MCK}）[注]	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		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

注：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注：

1. X：忽略
2. m：单元号（m=0、1、2）n：通道号（n=0、1）

16.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如图 16-95 所示。

图 16-95：溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。 →	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。	-	判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。 →	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SDIRmn 寄存器，只能清除读操作时的错误。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）

16.7 UART（UART0~UART2）通信的运行

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

[数据的发送和接收]

- 7 位、8 位、9 位或者 16 位的数据长度[注]
- 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 SCI1 的通道 0 和通道 1。
UART2 使用 SCI2 的通道 0 和通道 1。
各通道任意选择一个功能使用，除了所选功能以外，其他功能不能运行。
例如，在单元 m 的通道 0 和通道 1 使用 UART0 时，不能使用 SSPI00 和 IIC01。

注：当用作 UART 时，发送方（偶数通道）和接收方（奇数通道）只能用于 UART。

UART 有以下 2 种通信运行：

- UART 发送（参照 16.7.1）
- UART 接收（参照 16.7.2）

16.7.1 UART发送

UART 发送是本产品微控制器将数据异步发送到其他设备的运行。

UART 使用的 2 个通道中的偶数通道用于 UART 发送。

UART	UART0	UART1	UART2
对象通道	SCI0 的通道 0	SCI1 的通道 0	SCI2 的通道 0
使用的引脚	TxD0	TxD1	TxD2
中断	INTST0	INTST1	INTST2
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	无		
传送数据长度	7 位、8 位、9 位或者 16 位		
传送速率	Max.F_{MCK}/6[bps]（SDRmn[15:9]≥3）、Min.F_{CLK}/(2 x 2^{11} x 128)[bps][注]		
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： • 无奇偶校验位。 • 附加零校验。 • 附加偶校验。 • 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：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用。

备注：

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

(1) 寄存器的设定

图 16-96：UART（UART0~UART2）的 UART 发送时的寄存器设定内容例子(1/2)

(a) 串行模式寄存器mn(SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	0	0	0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	0	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0/1	SLCmn0 0/1	DLSmn3 0/1	DLSmn2 0/1	DLSmn1 0/1	DLSmn0 0/1

奇偶校验位的设定
00B：无奇偶校验
01B：附件零校验
10B：附加偶校验
11B：附加奇校验

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

停止位的设定
01B：附加1位
10B：附加2位

(c) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定							0	0	0	0	0	0	0	0	0

(2) 运行期间（SEmn=1）（低8位：SDRmnL）

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	发送数据寄存器															

SDRmnL

(d) 串行输出电平寄存器m(SOLm) …...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
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm0 0/1

0：正相(通常)发送
1：反相发送

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0）q：UART 号（q=0~2）
2. □：在 UART 发送模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

图 16-96：UART（UART0~UART2）的 UART 发送时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) …...只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 注

0：串行数据输出值为 “0”

1：串行数据输出值为 “1”

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

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

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

注：在开始发送前，当对应通道的 SOLmn 位为“0”时，必须置“1”；当对应通道的 SOLmn 位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0）q：UART 号（q=0~2）
2. □：在 UART 发送模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-97：UART 发送的初始设定步骤

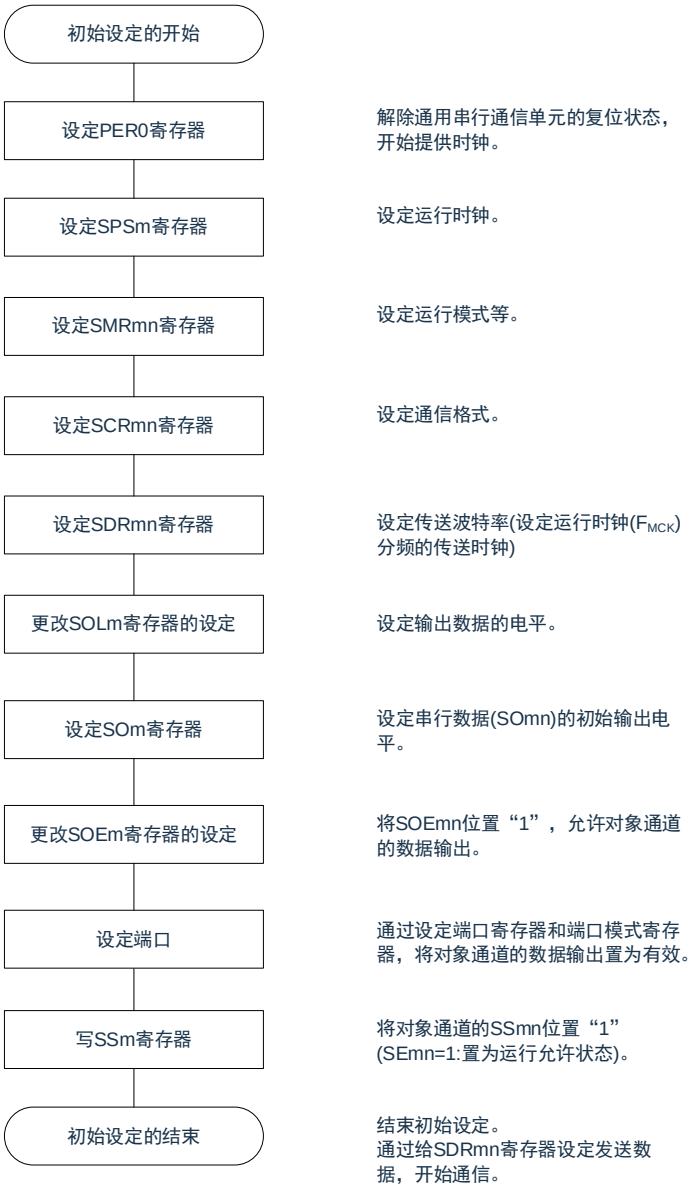


图 16-98：UART 发送的中止步骤

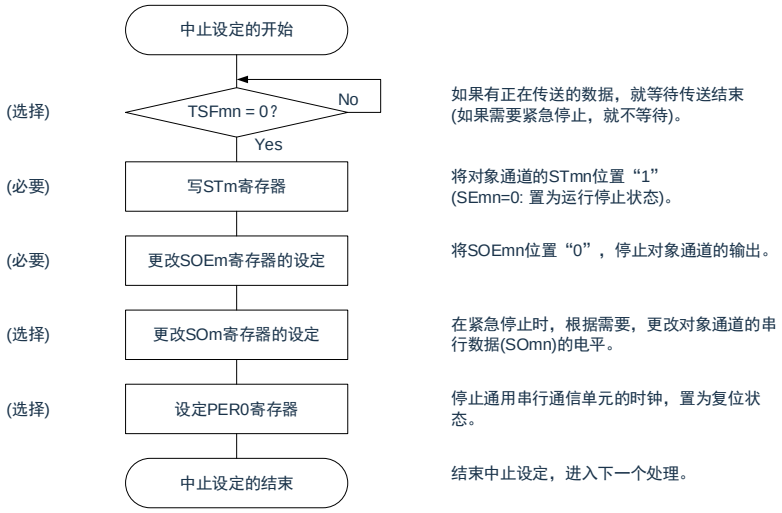
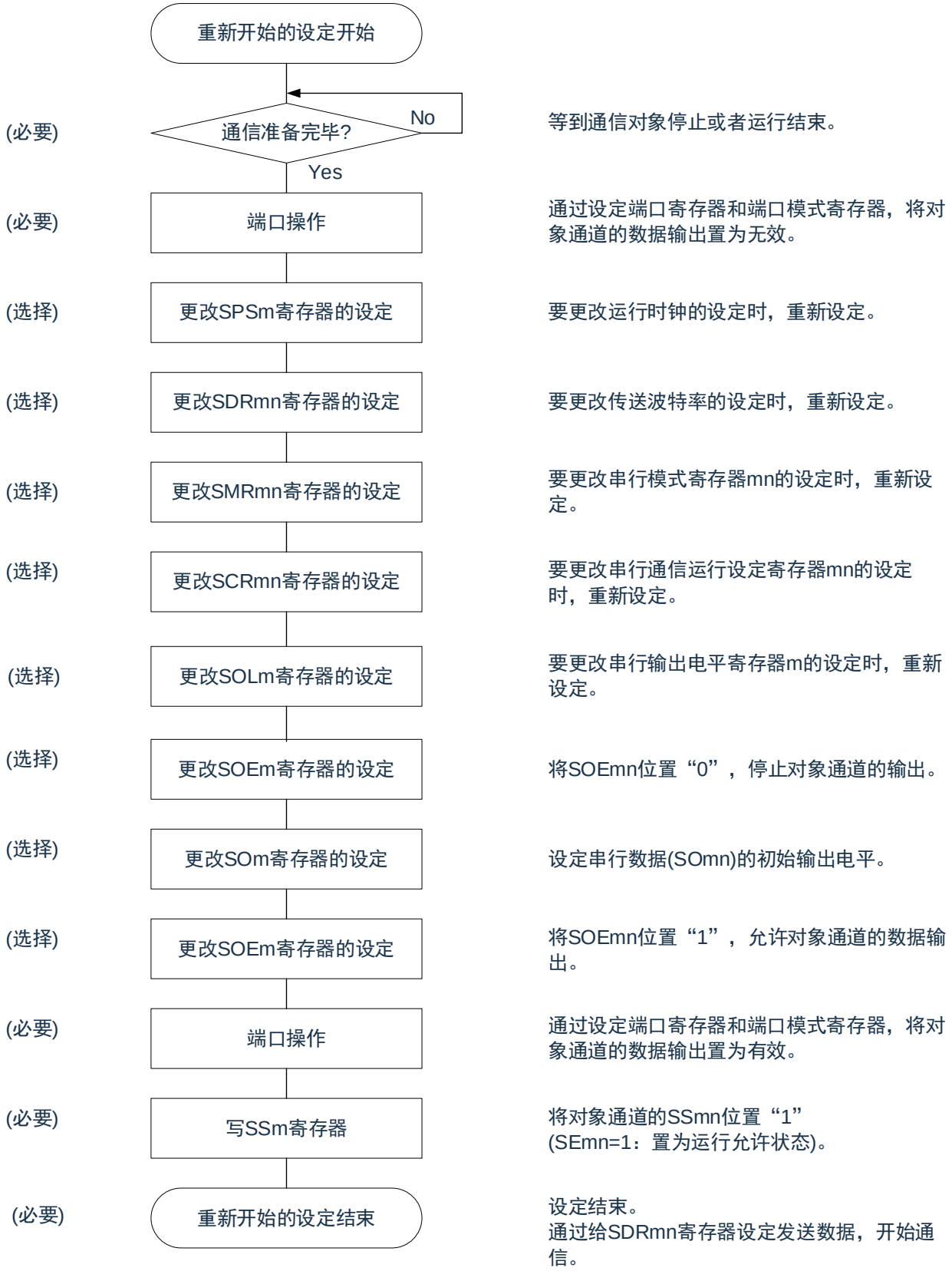


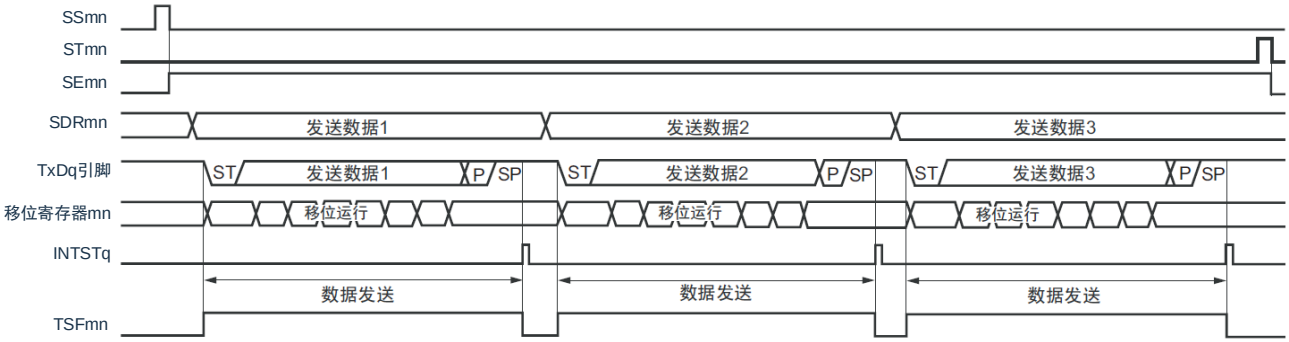
图 16-99：重新开始 UART 发送的设定步骤



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

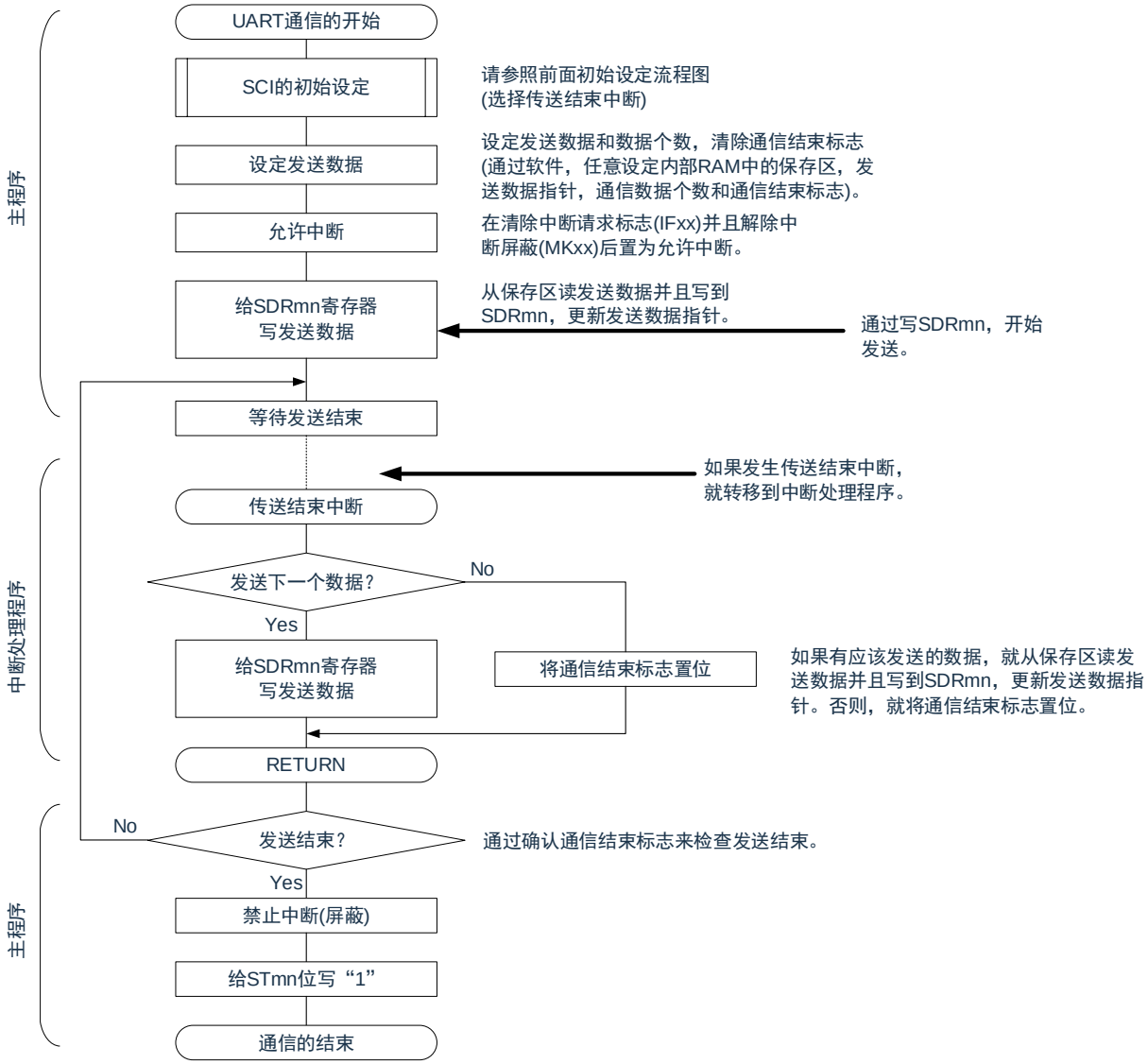
(3) 处理流程（单次发送模式）

图 16-100：UART 发送（单次发送模式）的时序图



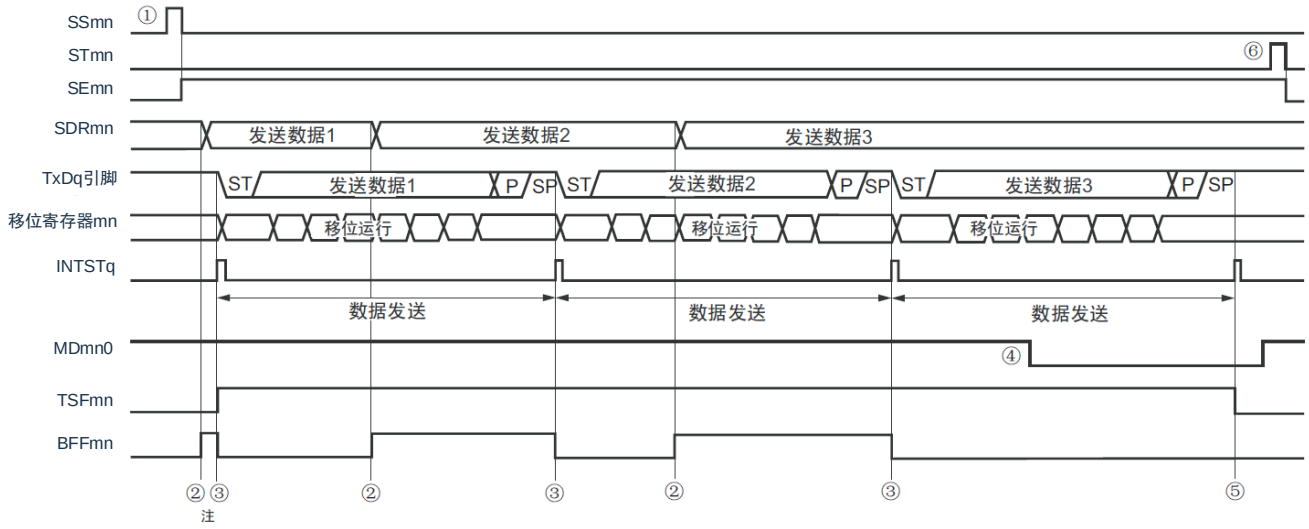
备注：m：单元号（m=0、1、2）n：通道号（n=0）q：UART 号（q=0~2）

图 16-101：UART 发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 16-102：UART 发送（连续发送模式）的时序图

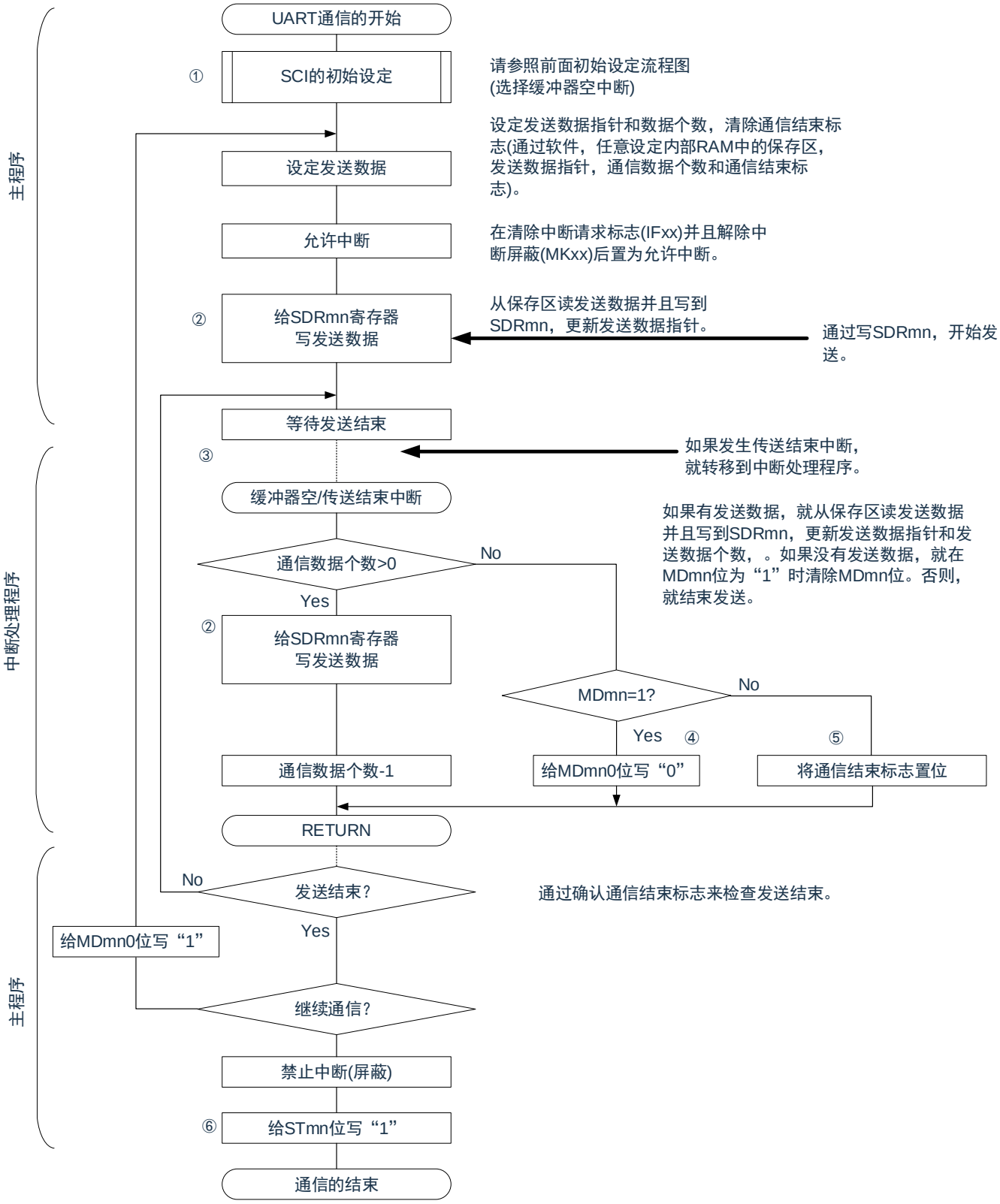


注：如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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

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

图 16-103：UART 发送（连续发送模式）的流程图



备注：图中的①~⑥对应“图 16-102 UART 发送（连续发送模式）的时序图”中的①~⑥。

16.7.2 UART接收

UART 接收是本产品微控制器其他设备异步接收数据的运行。

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

UART	UART0	UART1	UART2
对象通道	SCI0 的通道 1	SCI1 的通道 1	SCI2 的通道 1
使用的引脚	RxD0	RxD1	RxD2
中断	INTSR0	INTSR1	INTSR2
	只限于传送结束中断（禁止设定缓冲器空中断）。		
错误检测标志	• 帧错误检测标志（FEFmn） • 奇偶校验错误检测标志（PEFmn） • 溢出错误检测标志（OVFmn）		
传送数据长度	7 位、8 位、9 位或者 16 位		
传送速率	Max.F_{MCK}/6[bps]（SDRmn[15:9]≥2）、 Min.F_{CLK}/(2 x 2^{15} x 128)[bps]		
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： • 无奇偶校验位（无奇偶校验）。 • 附加零校验（无奇偶校验）。 • 偶校验 • 奇校验		
停止位	附加 1 位。		
数据方向	MSB 优先或者 LSB 优先		

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

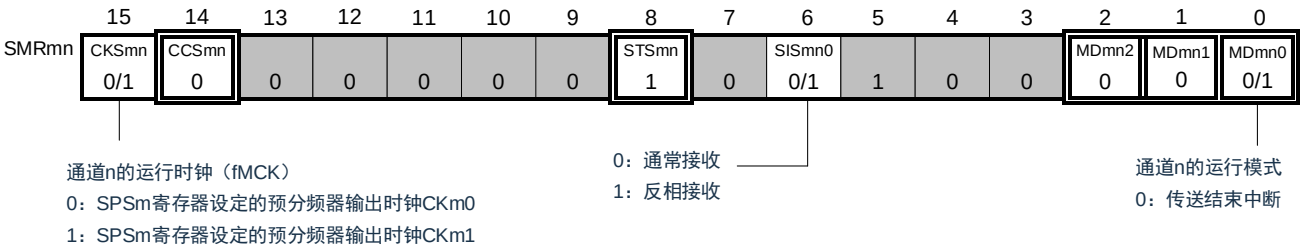
备注：

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

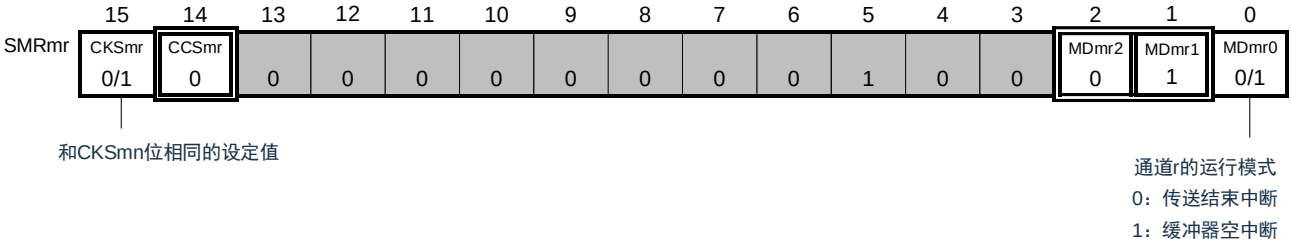
(1) 寄存器的设定

图 16-104：UART（UART0~UART2）的 UART 接收时的寄存器设定内容例子(1/2)

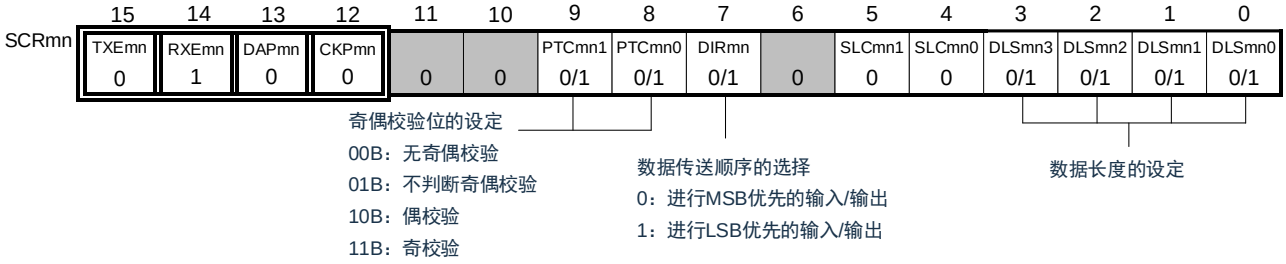
(a) 串行模式寄存器mn(SMRmn)



(b) 串行模式寄存器mn(SMRmr)

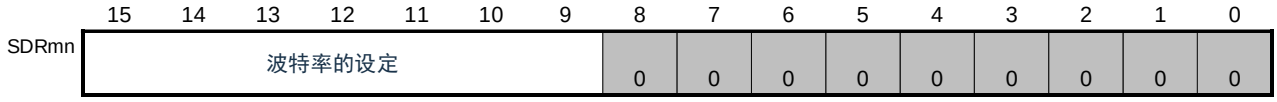


(c) 串行通信运行设定寄存器mn(SCRmn)

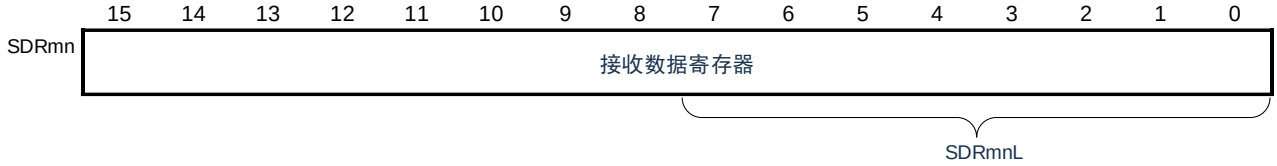


(d) 串行数据寄存器mn(SDRmn)

(1) 运行停止（SEmn=0）时



(2) 运行期间（SEmn=1）（低8位：SDRmnL）



注意：在 UART 接收时，还必须设定与通道 n 成对的通道 r 的 SMRmr 寄存器。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=1）
   r：通道号（r=n-1）q：UART 号（q=0~2）
2. □：在 UART 接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

图 16-104：UART（UART0~UART2）的 UART 接收时的寄存器设定内容例子(2/2)

(e) 串行输出寄存器m(SOm) …...在此模式中不使用。

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

备注：

1. m：单元号（m=0、1、2）
2. □：在 UART 接收模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 16-105：UART 接收的初始设定步骤

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

图 16-106：UART 接收的中止步骤

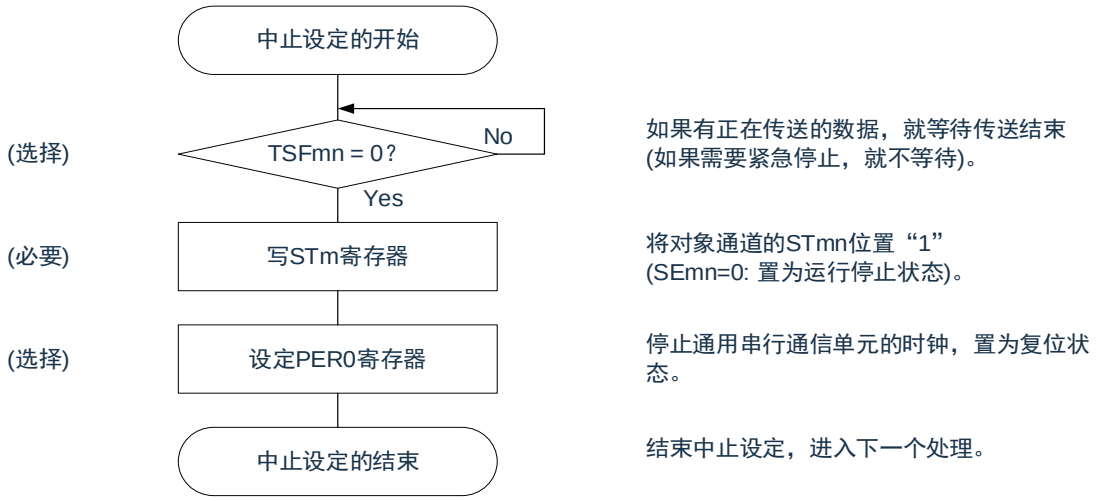


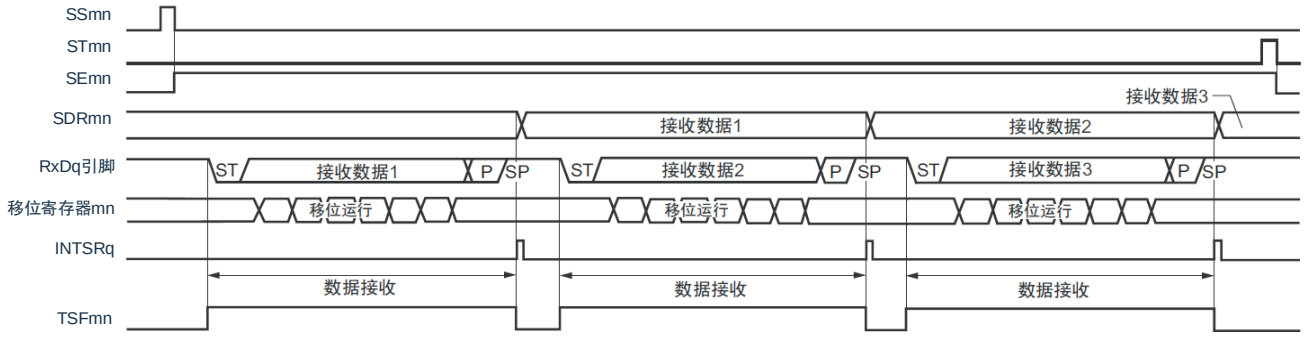
图 16-107：重新开始 UART 接收的设定步骤

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

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

(3) 处理流程

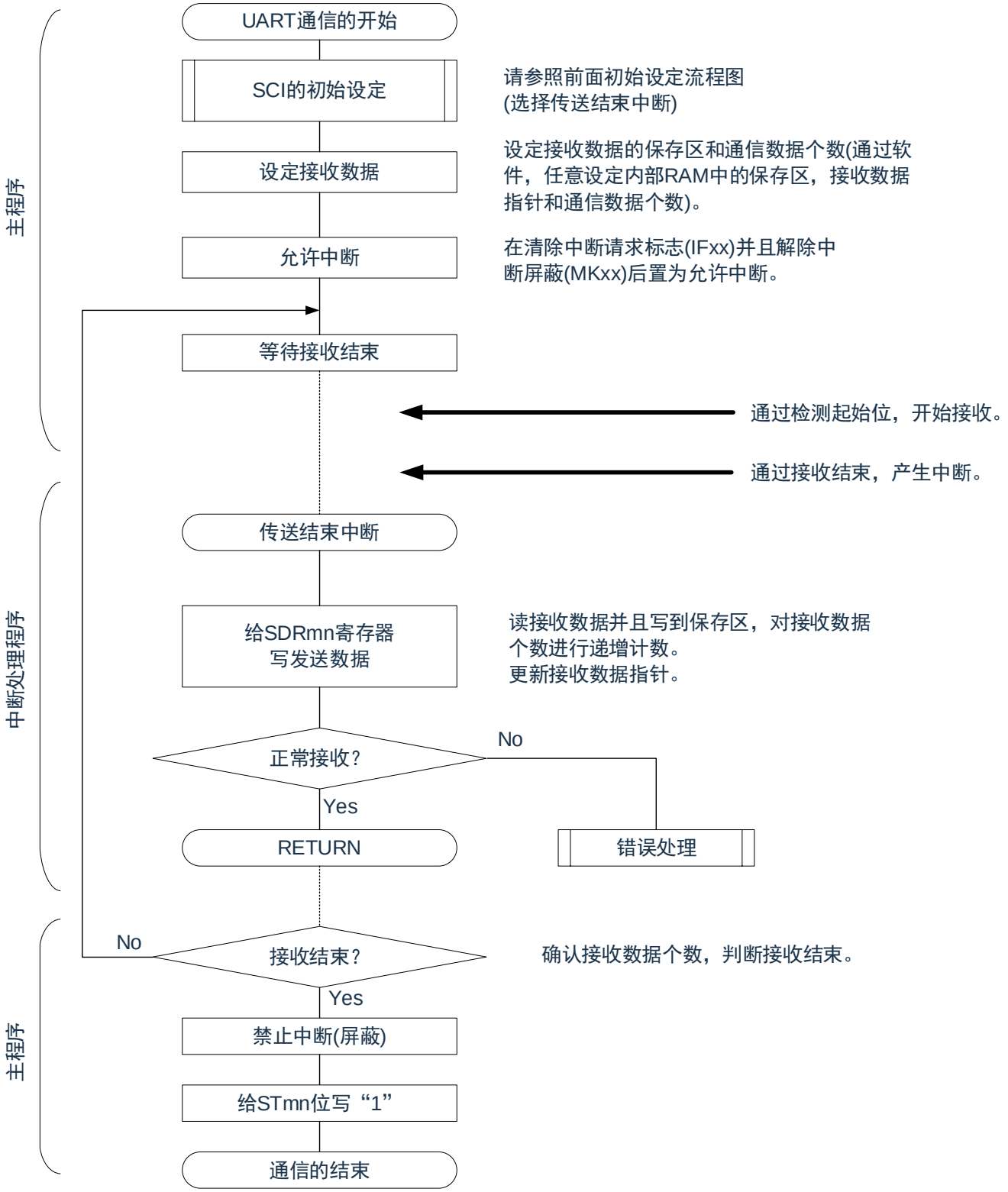
图 16-108：UART 接收的时序图



备注：m：单元号（m=0、1、2）n：通道号（n=1）

R：通道号（r=n-1）q：UART 号（q=0~2）

图 16-109：UART 接收的流程图



16.7.3 低功耗UART模式功能

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

要在低功耗 UART 模式中使用 UARTq 时，必须在转移到深度睡眠模式前进行以下的设定 （参照 “图 16-111 和图 16-113 低功耗 UART 模式运行的流程图 ”）。

- 在低功耗 UART 模式中，需要更改 UART 接收波特率的设定 （更改为和通常运行时不同的值）。必须参照表 16-4，设定 SPSm 寄存器和 SDRmn 寄存器[15:9]。
- 设定 EOCmn 位和 SSECmn 位。能设定允许或者停止在发生通信错误时产生错误中断 。
- 必须在即将要转移到深度睡眠模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。在初始设定结束后，将串行通道开始寄存器 m （SSm）的 SSm1 位置“1”。
- 如果在转移到深度睡眠模式后检测到 RxDq 输入的起始位，UARTq 就开始接收。

注意：

1. 只有在选择高速内部振荡器时钟（F_{IH}）作为 F_{CLK} 时才能使用低功耗 UART 模式。
2. 低功耗 UART 模式中的传送速率只为 4800bps。
3. 如果将 SWCm 位置 “1”，就只能在深度睡眠模式中开始接收时使用 UARTq。当其他低功耗 UART 模式的功能和中断同时使用并且在以下非深度睡眠模式的状态下开始接收时，可能无法正常接收数据而发生帧错误或者奇偶校验错误。
   - 在将 SWCm 位置 “1” 后并且在转移到深度睡眠模式前开始接收的情况
   - 在其他低功耗 UART 模式中开始接收的情况
   - 在通过中断等从深度睡眠模式返回到通常运行后并且在将 SWCm 位置 “0” 前开始接收的情况
4. 如果将 SSECm 位置 “1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFmn、FEFmn、OVFmn 标志置位。因此，在 SSECm 位为 “1” 的情况下使用时，必须在将 SWC0 位置 “1” 前清除 PEFmn、FEFmn、OVFmn 标志并且读 SDRm1 寄存器的 bit7 ~ 0 （RxDq）。
5. 通过检测 RxDq 引脚的有效边沿，转移到低功耗 UART 模式。
   如果接受到无法检测到输入起始位的短脉冲，就可能不开始 UART 接收而继续保持低功耗 UART 模式。此时，可能在下一次的 UART 接收过程中无法正常接收数据而发生帧错误或者奇偶校验错误。

表 16-4：低功耗 UART 模式中 UART 接收波特率的设定

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

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

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

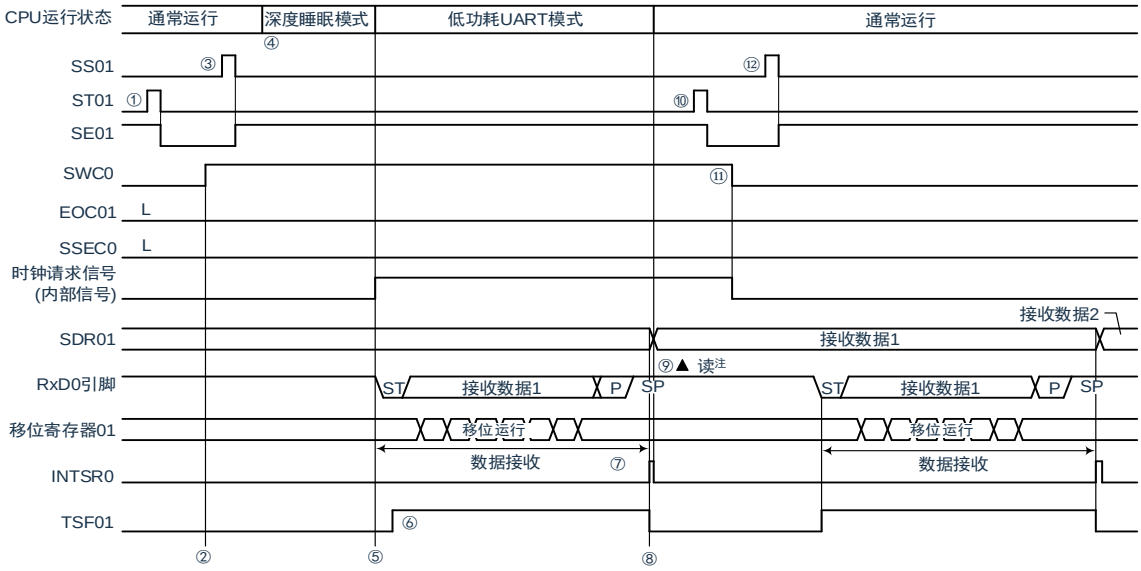
备注：最大容许值和最小容许值是 UART 接收时的波特率的容许值。

必须将发送方的波特率设定在此范围内。

(1) 低功耗 UART 模式运行 （EOCm1=0、 SSECm=0/1）

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

图 16-110：低功耗 UART 模式运行（EOCm1=0、SSECm=0/1）时序图

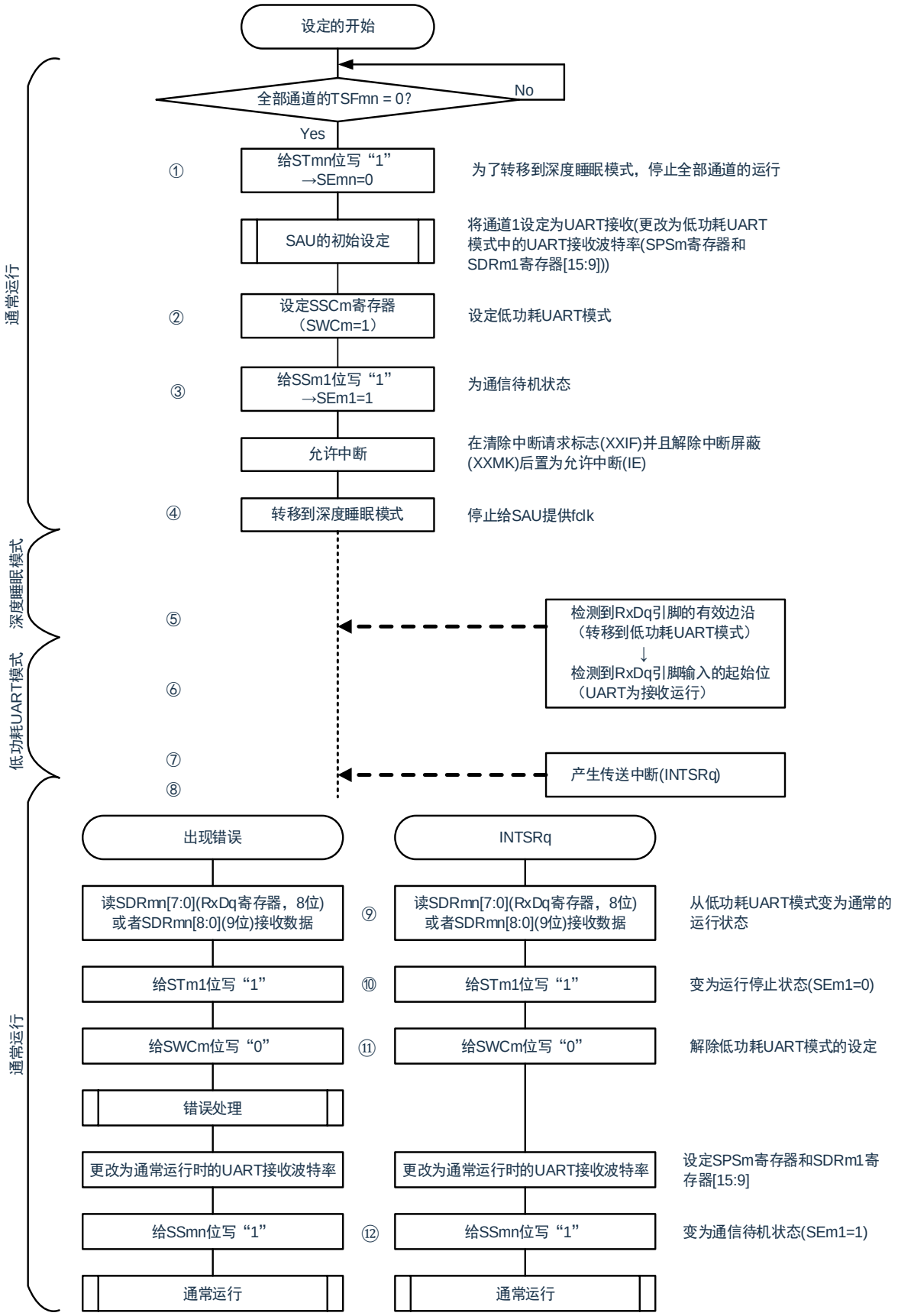


注：必须在 SWCm 位为 “1” 的状态下读接收数据。

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

备注：图中的①~⑫对应 “ 图 16-111：低功耗 UART 模式运行（EOCm1=0、SSECm=0/1）的流程图 ” 中的①~⑫。

图 16-111：低功耗 UART 模式运行（EOCm1=0、SSECm=0/1）的流程图

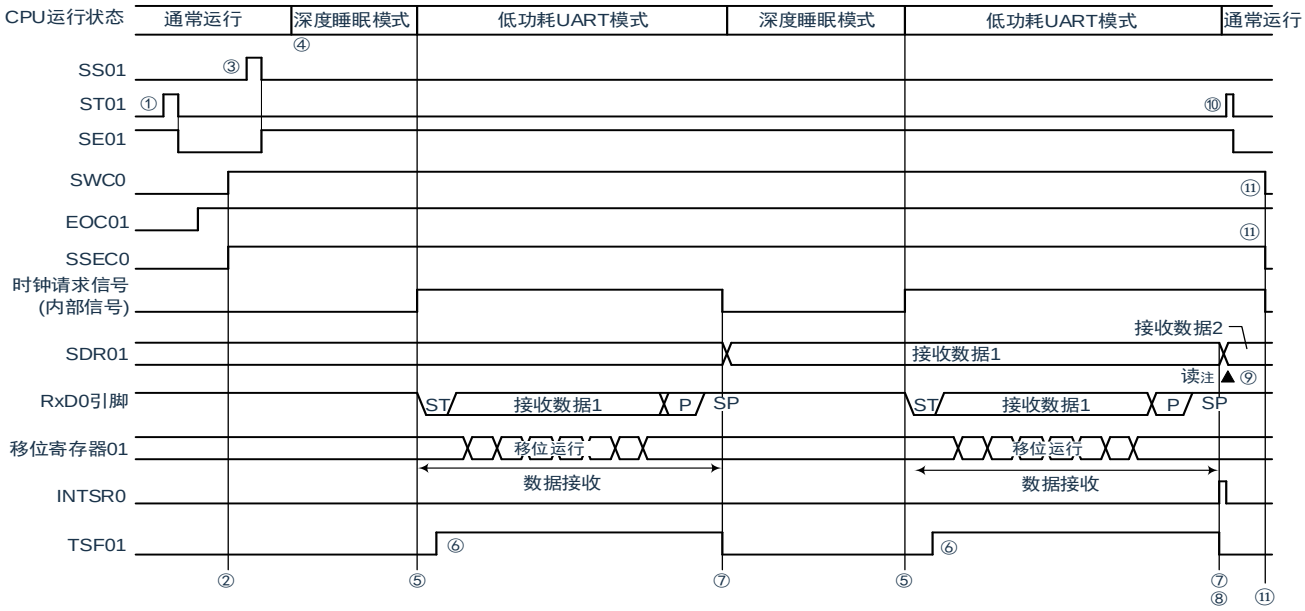


备注：图中的①~⑫ 对应“图 16-110：低功耗 UART 模式运行（EOCm1=0、SSECm=0/1）时序图”中的①~⑫。

(2) 低功耗 UART 模式运行（EOCm1=1、 SSECm=1）

因为 EOCm1 位为 “1”，所以发生通信错误不产生错误中断。

图 16-112：低功耗 UART 模式运行（EOCm1=1、SSECm=1）的时序图



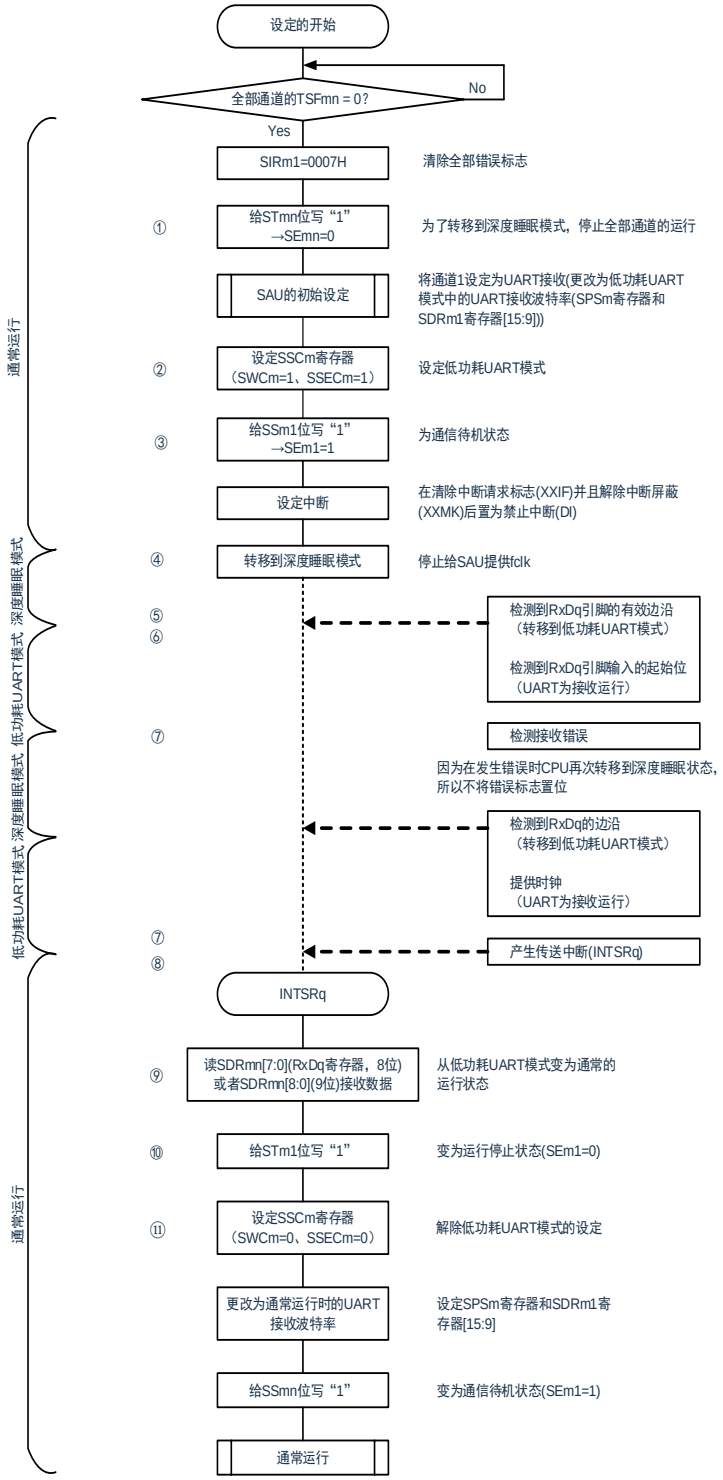
注：必须在 SWCm 位为 “1” 的状态下读接收数据。

注意：

1. 在向低功耗 UART 模式转移前或者在低功耗 UART 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除低功耗 UART 模式）。
2. 如果 SSECm 位为 “1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位。因此，在 SSECm 位为 “1” 的情况下使用时，必须在将 SWCm 位置 “1” 前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0]（RXDq 寄存器，8 位）或者 SDRm1[8:0]（9 位）。

备注：图中的①~⑪对应 “图 16-113：低功耗 UART 模式运行（EOCm1=1、SSECm=1）的流程图” 中的①~⑪

图 16-113：低功耗 UART 模式运行（EOCm1=1、SSECm=1）的流程图



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

备注：图中的①~⑪对应 “图 16-112：低功耗 UART 模式运行（EOCm1=1、SSECm=1）的时序图” 中的①~⑪。

16.7.4 波特率的计算

(1) 波特率的计算式

UART（UART0~UART2）通信的波特率能用以下计算式进行计算：

（波特率）={对象通道的运行时钟（F_{MCK}）频率}÷(SDRmn[15:9]+1)÷2[bps]

注意：禁止将串行数据寄存器 mn（SDRmn）的 SDRmn[15:9]设定为“0000000B”和“0000001B”。

备注：

1. 因为在使用 UART 时 SDRmn[15:9]的值为 SDRmn 寄存器的 bit15~9 的值（0000010B~1111111B），所以为 2~127。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）
   运行时钟（F_{MCK}）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn 位）。

表 16-5：UART 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟（F_{MCK}）注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		F_{CLK}=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 fCLK 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注：

1. X：忽略
2. m：单元号（m=0、1、2）n：通道号（n=0、1）

(2) 发送时的波特率误差

UART（UART0~UART2）通信发送时的波特率误差能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

（波特率误差）=（波特率的计算值）÷（目标波特率的值）×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）n：通道号（n=0）

(3) 接收时的波特率容许范围

UART（UART0~UART2）通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

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

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

Brate：接收方的波特率的计算值（参照“波特率的计算式”）

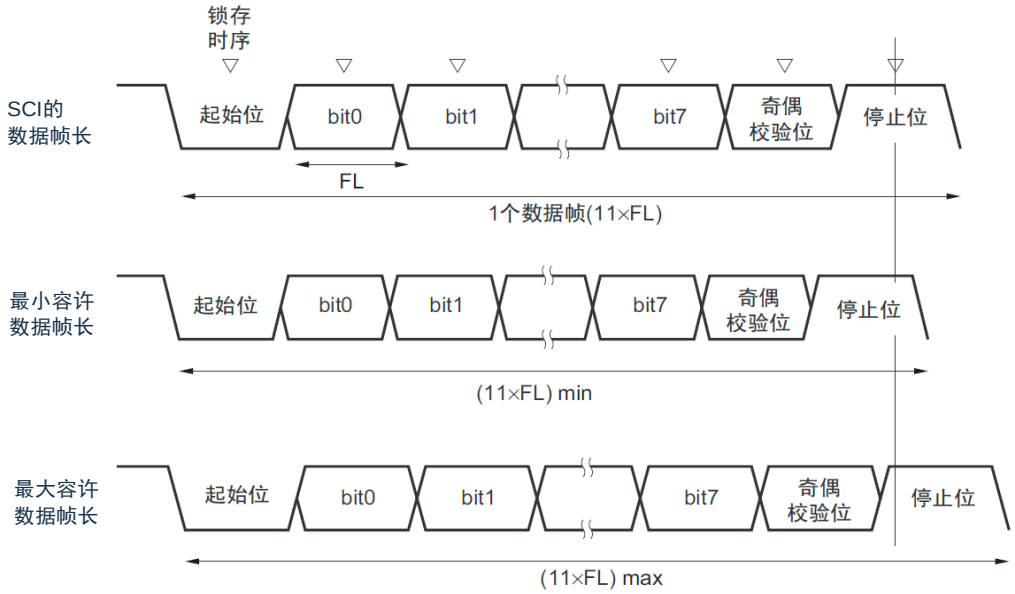
k：SDRmn[15:9]+1

Nfr：1 个数据的帧长[位]

=（起始位）+（数据长度）+（奇偶校验位）+（停止位）

备注：m：单元号（m=0、1、2） n：通道号（n=1）

图 16-114：接收时的波特率容许范围（1 个数据的帧长=11 位的情况）



如图 16-114 所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器 mn（SDRmn）的 bit15~9 设定的分频比。如果最后的数据（停止位）能赶上此锁存时序，就能正常接收。

16.7.5 在UART（UART0~UART2）通信过程中发生错误时的处理步骤

在 UART（UART0~UART2）通信过程中发生错误时的处理步骤如图 16-115 和图 16-116 所示。

图 16-115：发生奇偶校验错误或者溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SDIRmn 寄存器，只能清除读操作时的错误。

图 16-116：发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器 mn（SIRmn）。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 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）n：通道号（n=0、1）

16.8 LIN通信的运行

16.8.1 LIN发送

在 UART 发送中，UART0 支持 LIN 通信。

LIN 发送使用单元 0 的通道 0。

UART	UART0	UART1	UART2
LIN 通信的支持	能	不能	不能
对象通道	SCI0 的通道 0	—	—
使用的引脚	TxD0	—	—
中断	INTST0	—	—
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	无		
传送数据长度	8 位		
传送速率[注]	Max.F_{MCK}/6[bps]（SDR00[15:9]≥2）、Min.F_{CLK}/(2 x 2^{15} x 128)[bps]		
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	无奇偶校验位。		
停止位	附加 1 位。		
数据方向	LSB 优先		

注：必须在满足此条件并且满足电特性的外围功能特性（参照数据手册）的范围内使用，并且在 LIN 通信中经常使用 2.4/9.6/19.2kbps。

备注：F_{MCK}：对象通道的运行时钟频率 F_{CLK}：系统时钟频率

LIN 是 Local Interconnect Network 的简称，是为降低汽车网络成本的低速（1~20kbps）串行通信协议。LIN 通信是单主控通信，一个主控设备最多可连接 15 台从属设备。

LIN 从属设备用于开关、传动装置和传感器等的控制，这些装置通过 LIN 连接到主控设备。

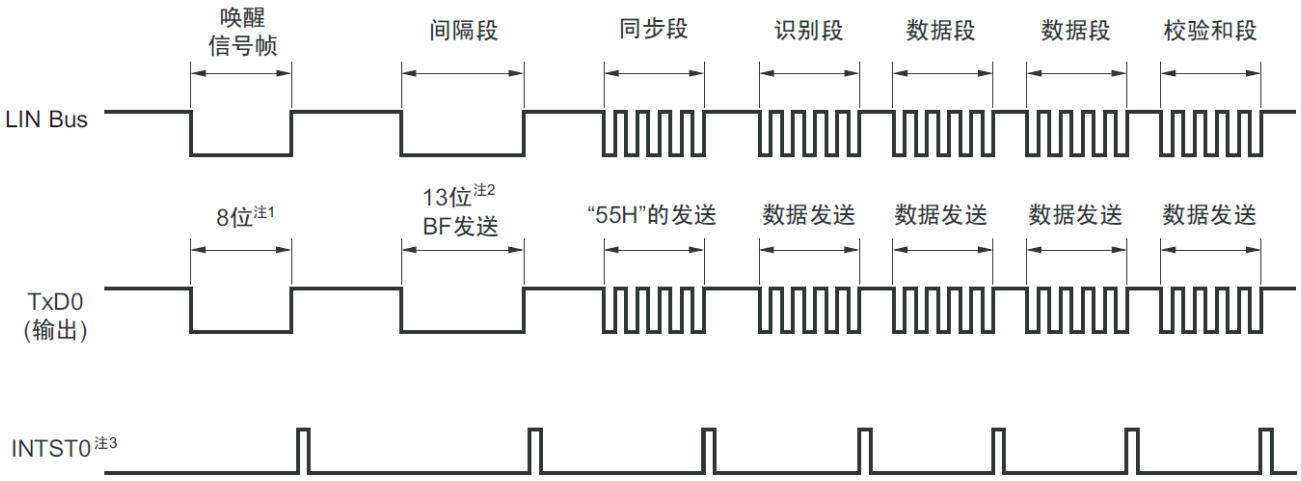
LIN 主控一般连接 CAN（Controller Area Network）等的网络。

LIN 总线是单线方式的总线，通过符合 ISDO9141 的收发器连接各节点。

根据 LIN 协议，主控设备发送附加波特率信息的帧，从属设备接收此帧并且校正与主控设备的波特率误差。因此，如果从属设备的波特率误差不大于±15%，就能进行通信。

LIN 的发送操作的概要如图 14-113 所示。

图 16-117：LIN 的发送操作



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

注 2：间隔段规定为 13 位宽的低电平输出，因此假设主传送使用的波特率为 N[bps]，间隔段使用的波特率如下：

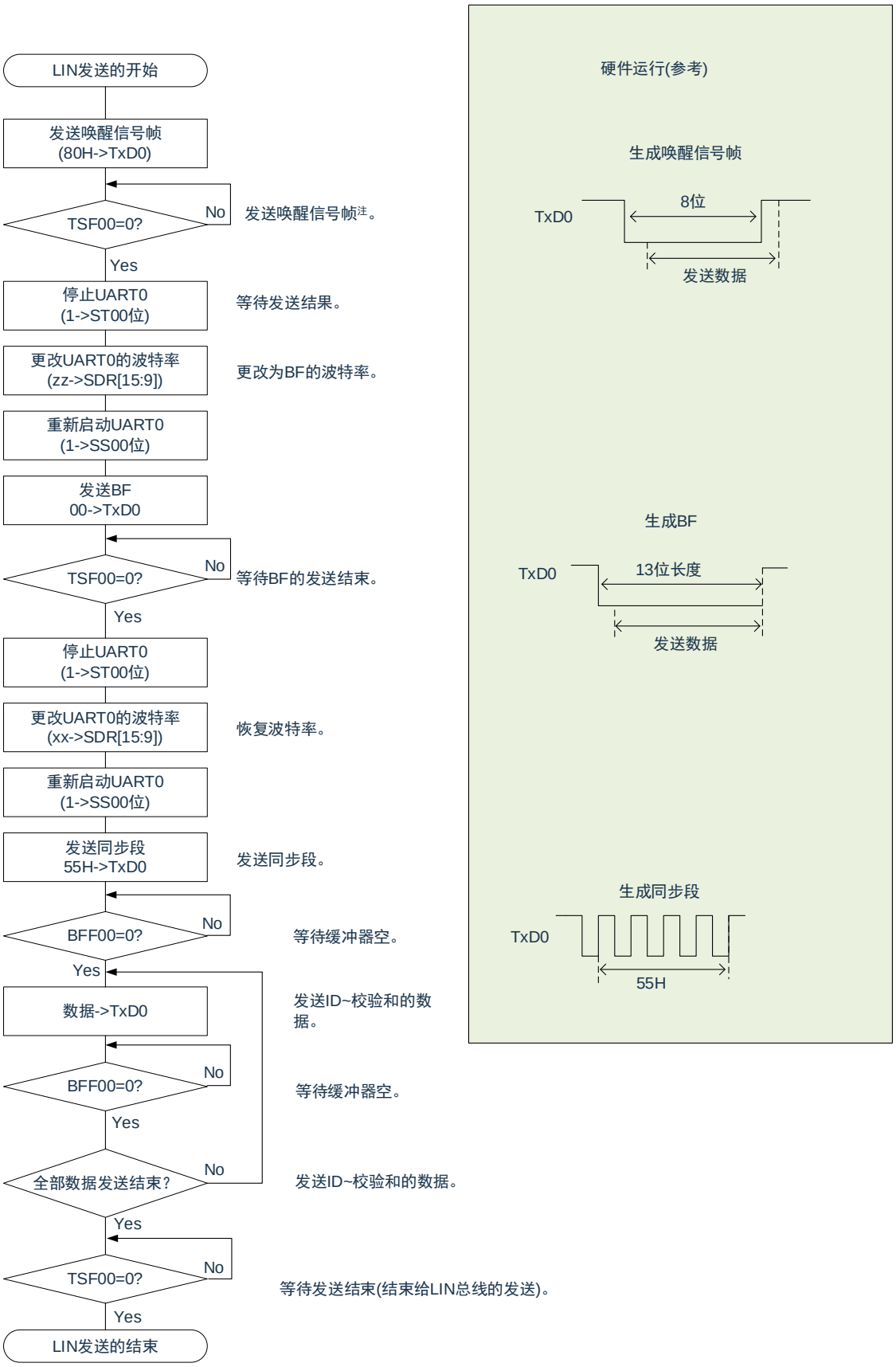
(间隔段的波特率)=9/13×N

通过此波特率发送“00H”的数据，生成间隔段。

注 3：在各数据发送结束时输出 INTST0，而且在 BF 发送时也输出 INTST0。

备注：由软件控制各段间的间隔。

图 16-118：LIN 发送的流程图



注：只限于从 LIN-bus 睡眠状态启动的情况。

备注：这是从结束 UART 的初始设定并且允许从属发送开始的流程。

16.8.2 LIN接收

在 UART 接收中，UART0 支持 LIN 通信。

LIN 接收使用单元 0 的通道 1。

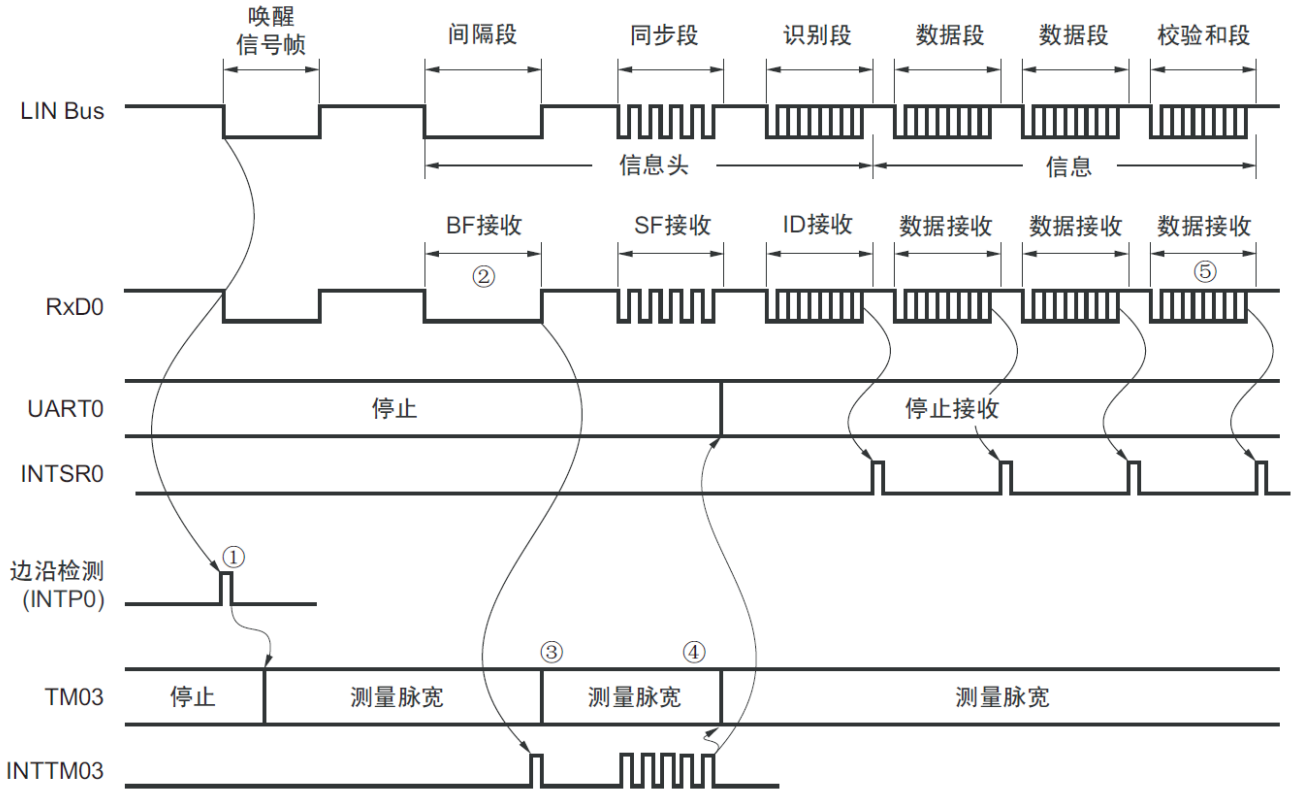
UART	UART0	UART1	UART2
LIN通信的支持	能	不能	不能
对象通道	SCI0的通道1	—	—
使用的引脚	RxD0	—	—
中断	INTSR0	—	—
	只限于传送结束中断（禁止设定缓冲器空中断）。		
错误检测标志	• 帧错误检测标志（FEF01） • 溢出错误检测标志（OVF01）		
传送数据长度	8位		
传送速率[注]	Max.F_{MCK}/6[bps]（SDR01[15:9]≥2）、Min.F_{CLK}/(2 x2^{15}x 128)[bps]		
数据相位	正相输出（默认值：高电平）。反相输出（默认值：低电平）。		
奇偶校验位	无奇偶校验位（不进行奇偶校验）。		
停止位	附加1位。		
数据方向	LSB优先		

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

备注：F_{MCK}：对象通道的运行时钟频率 F_{CLK}：系统时钟频率

LIN 的接收操作的概要如图 14-115 所示。

图 16-119：LIN 的接收操作



信号处理的流程如下：

① 通过检测引脚的中断边沿（INTP0）来检测唤醒信号。当检测到唤醒信号时，为了测量BF的低电平宽度，将TM03设定为测量脉宽，然后进入BF接收等待状态。

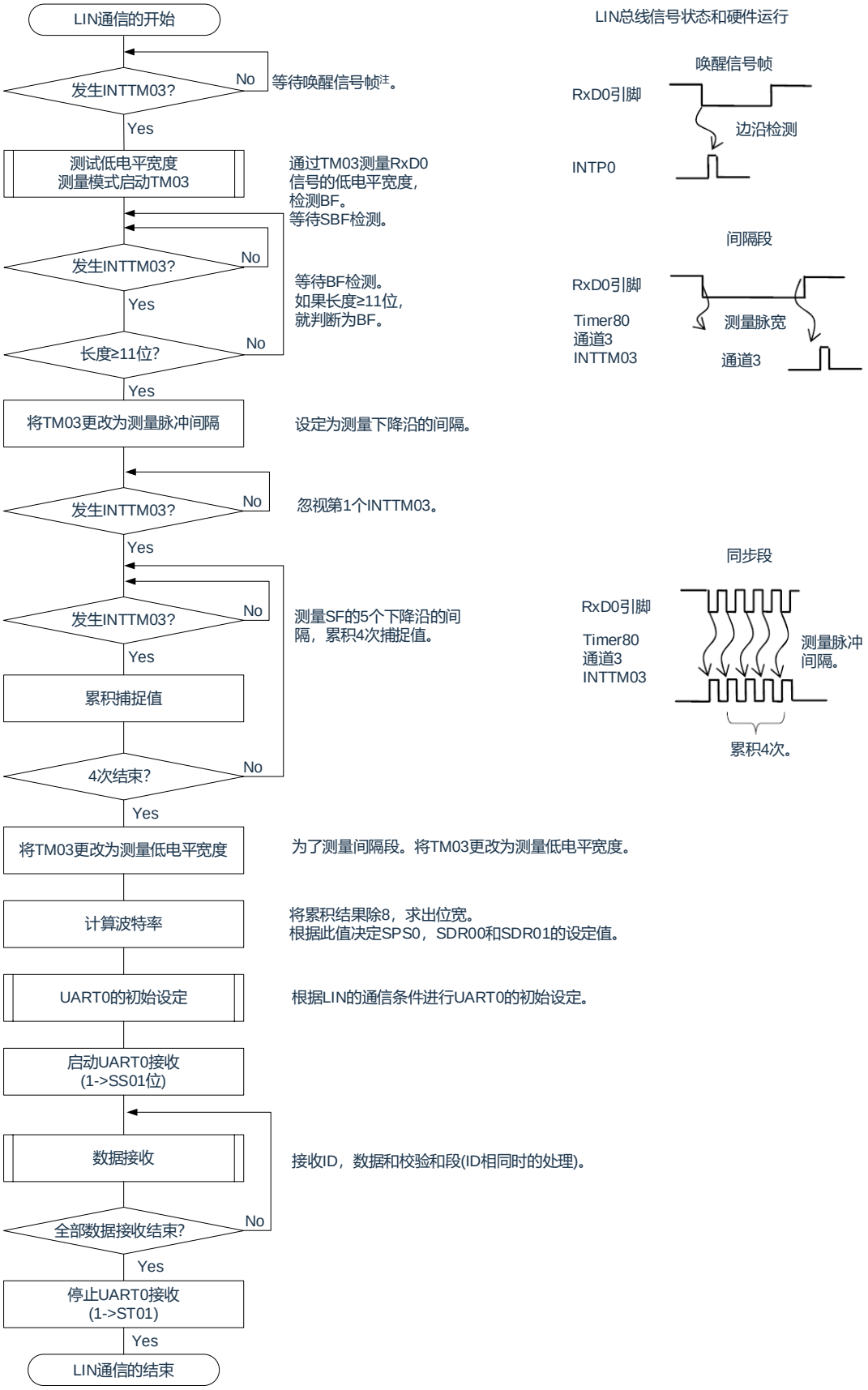
② 如果检测到BF的下降沿，TM03就开始测量低电平宽度，并且在BF的上升沿进行捕捉。根据捕捉到的值判断是否为BF信号。

③ 当BF接收正常结束时，必须将TM03设定为测量脉冲间隔，并且测量4次同步段的RxD0信号下降沿的间隔（参照“5.8.4 作为输入脉冲间隔测量的运行”）。

④ 根据同步段（SF）的位间隔计算波特率误差。然后，必须在暂停UART0运行后调整（重新设定）波特率。

⑤ 必须通过软件区分校验和段。还必须通过软件在接收校验和段后对UART0进行初始化并且再次设定为BF接收等待状态。

图 16-120：LIN 接收的流程图



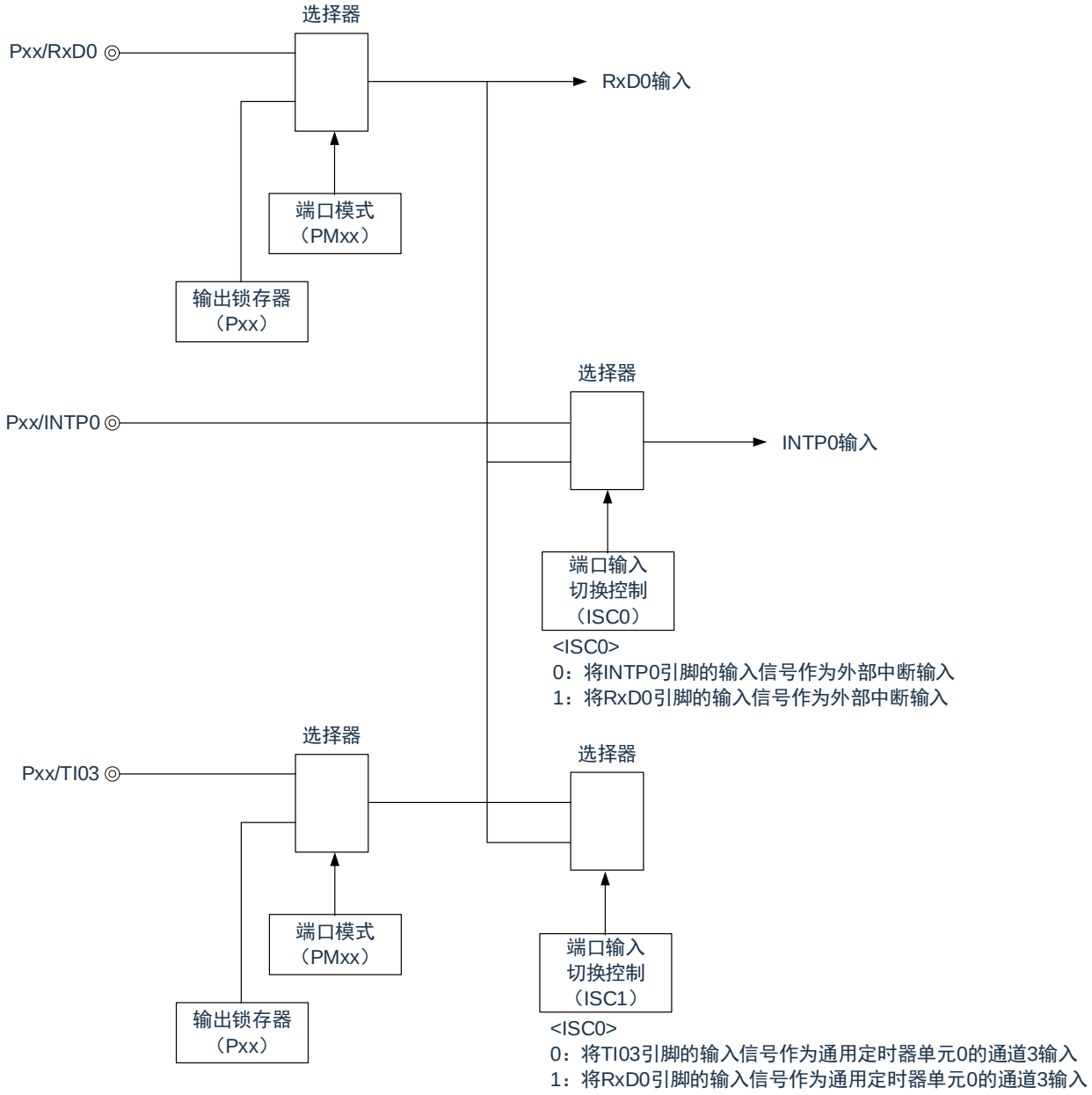
注：只在睡眠状态下才需要。

用于 LIN 接收操作的端口结构图如图图 16-121 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 16-121：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）

16.9 简易I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的运行

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

对于开始条件和停止条件，必须遵守 AC 规格，在操作控制寄存器的同时通过软件进行处理。

[数据的发送和接收]

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

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误

※[简易 I²C 不支持的功能]

- 从属发送、从属接收
- 多主控功能（仲裁失败检测功能）
- 等待检测功能

注：在接收最后的数据时，如果给 SDOEmn 位（SDOEm 寄存器）写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“16.9.3 (2)处理流程”。

备注：m：单元号（m=0、1、2）n：通道号（n=0、1）

SCI0 的通道 0~1、SCI1 的通道 0~1 以及 SCI2 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 16.9.1）
- 数据发送（参照 16.9.2）
- 数据接收（参照 16.9.3）
- 停止条件的产生（参照 16.9.4）

16.9.1 地址段发送

地址段发送是为了特别指定传送对象（从属设备）而在 I²C 通信时最先进行的发送运行。在产生开始条件后，将地址（7 位）和传送方向（1 位）作为 1 帧进行发送。

简易 I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCL00 SDA00 注 1	SCL01 SDA01 注 1	SCL10 SDA10 注 1	SCL11 SDA11 注 1	SCL20 SDA20 注 1	SCL21 SDA21 注 1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK 错误检测标志（PEFmn）					
传送数据长度	8 位（将高 7 位作为地址并且将低 1 位作为 R/W 控制进行发送）					
传送速率注 2	Max.F_{MCK}/4[Hz]（SDRmn[15:9]≥1）F_{MCK}：对象通道的运行时钟频率但是，必须在 I²C 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“”。

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、2）n：通道号（n=0、1）

(1) 寄存器的设定

图 16-122：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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 注1	0	SISmn0 0 注1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

通道n的运行时钟（fMCK）
0：SPSm寄存器设定的预分频器输出时钟CKm0
1：SPSm寄存器设定的预分频器输出时钟CKm1

通道n的中断源
0：传送结束中断
1：缓冲器空中断

(b) 串行通信运行设定寄存器mn(SCRmn)

	15	14	13	12	11	10	15	14	13	6	15	14	13	12		
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 注2	SLCmn0 1	DLSmn3 0	DLSmn2 1	DLSmn1 1 注3	DLSmn0 1

奇偶校验位位的设定
00B：无奇偶校验

停止位的设定
01B：附加1位(ACK)

(c) 串行数据寄存器mn(SDRmn)　(低8位：SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定							0	发送数据的设定（地址+R/W）							

SIOr

(d) 串行输出寄存器m(SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

通过操作Somn位来产生开始条件。

(e) 串行输出允许寄存器m(SOEm)

	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 0/1	SOEm0 0/1

在产生开始条件前SOEmn位为“0”，在产生开始条件后SOEmn位为“1”。

(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 0/1	SSm0 0/1

注 1：只限于 SMR01、SMR11、SMR21。

注 2：只限于 SCR00、SCR10、SCR20。

注 3：只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）
2. □：在 IIC 模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

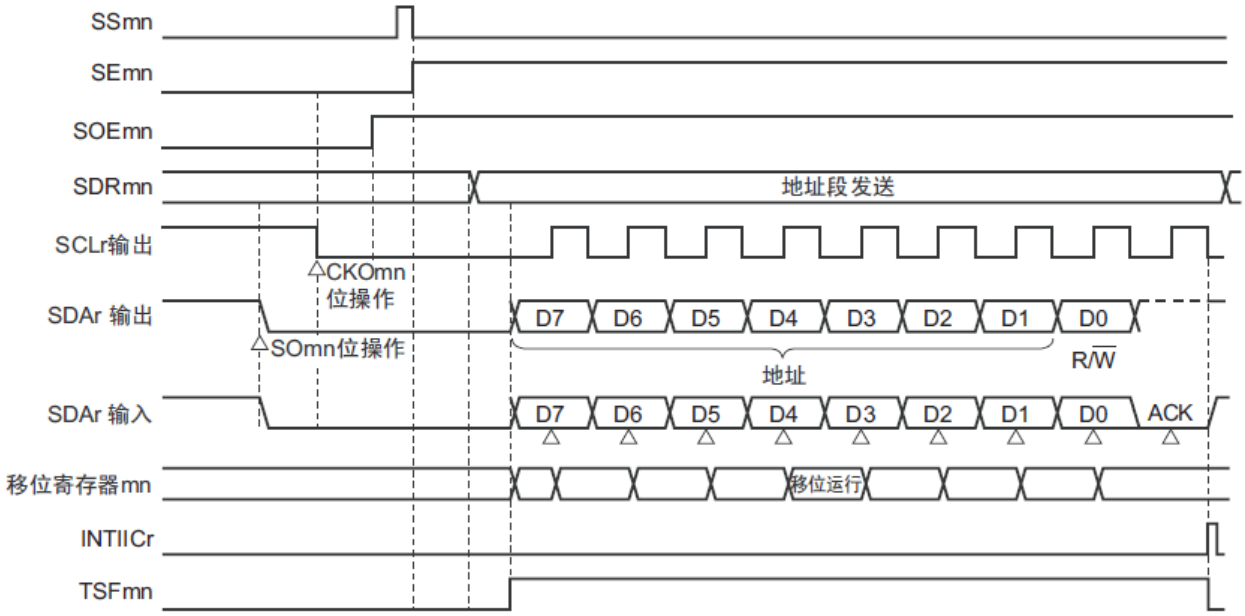
(2) 操作步骤

图 16-123：地址段发送的初始设定步骤

备注：在初始设定结束时，简易 I^2C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）为禁止输出并且处于运行停止状态。

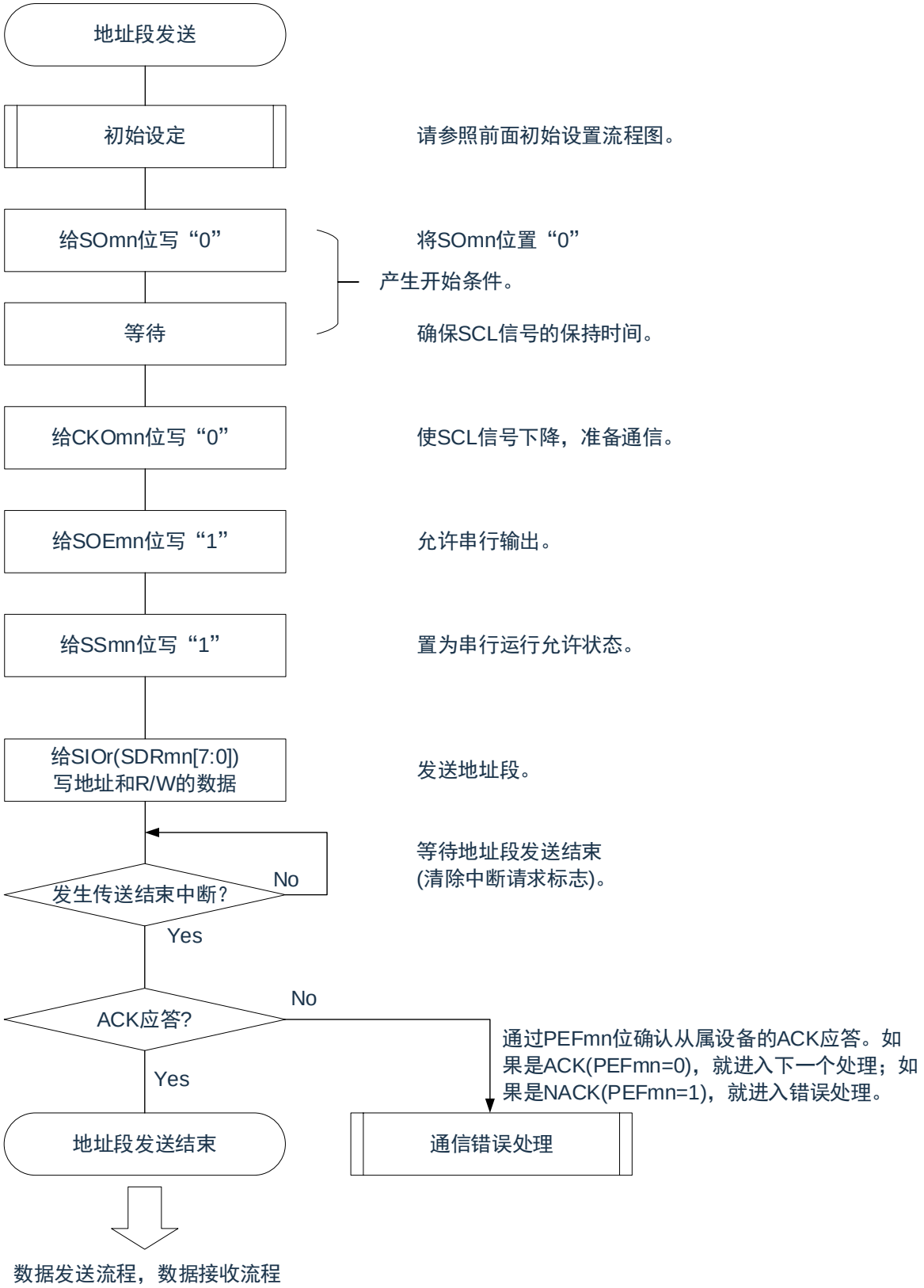
(3) 处理流程

图 16-124：地址段发送的时序图



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）

图 16-125：地址段发送的流程图



16.9.2 数据发送

数据发送是在发送地址段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

简易 I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCL00 SDA00 注 1	SCL01 SDA01 注 1	SCL10 SDA10 注 1	SCL11 SDA11 注 1	SCL20 SDA20 注 1	SCL21 SDA21 注 1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	ACK 错误标志（PEFmn）					
传送数据长度	8 位					
传送速率注 2	Max.F_{MCK}/4[Hz]（SDRmn[15:9]≥1）F_{MCK}：对象通道的运行时钟频率但是，必须在 I²C 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）。

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、2）n：通道号（n=0、1）

(1) 寄存器的设定

图 16-126：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
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 注1	0	SISmn0 0 注1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) 串行通信运行设定寄存器mn(SCRmn)…...在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	15	14	13	6	15	14	13	12		
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 注2	SLCmn0 1	DLSmn3 0	DLSmn2 1	DLSmn1 1 注3	DLSmn0 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不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定 注4							0	发送数据的设定							

SIOr

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 注5	CKOm0 0/1 注5	0	0	0	0	0	0	SOm1 0/1 注5	SOm0 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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 1	SOEm0 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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

注 1：只限于 SMR01、SMR11、SMR21 寄存器。

注 2：只限于 SCR00、SCR10、SCR20 寄存器。

注 3：只限于 SCR00、SCR01、SCR10、SCR11、SCR20、SCR21 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：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）
2. □：在 IIC 模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

(2) 处理流程

图 16-127：数据发送的时序图

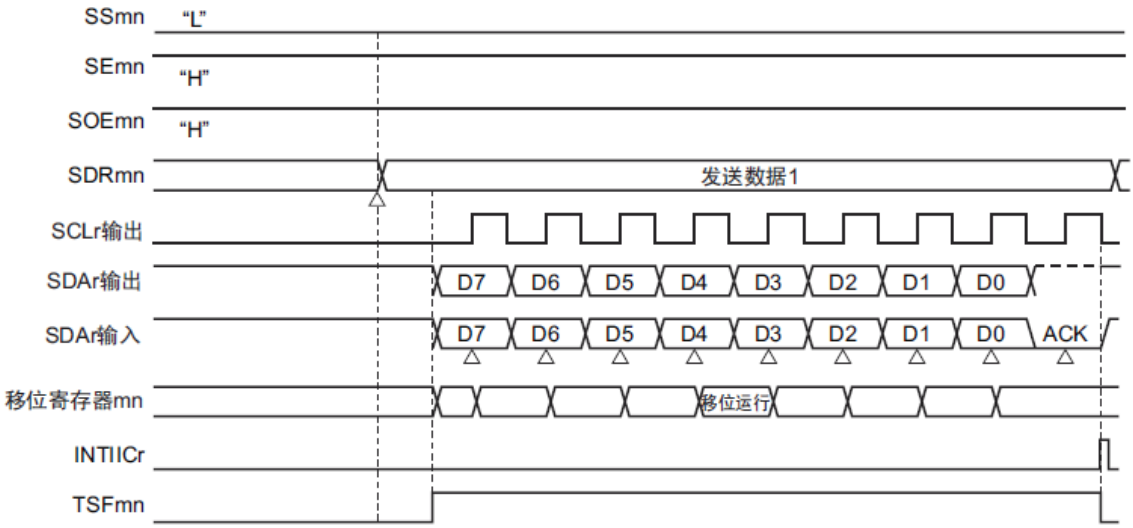
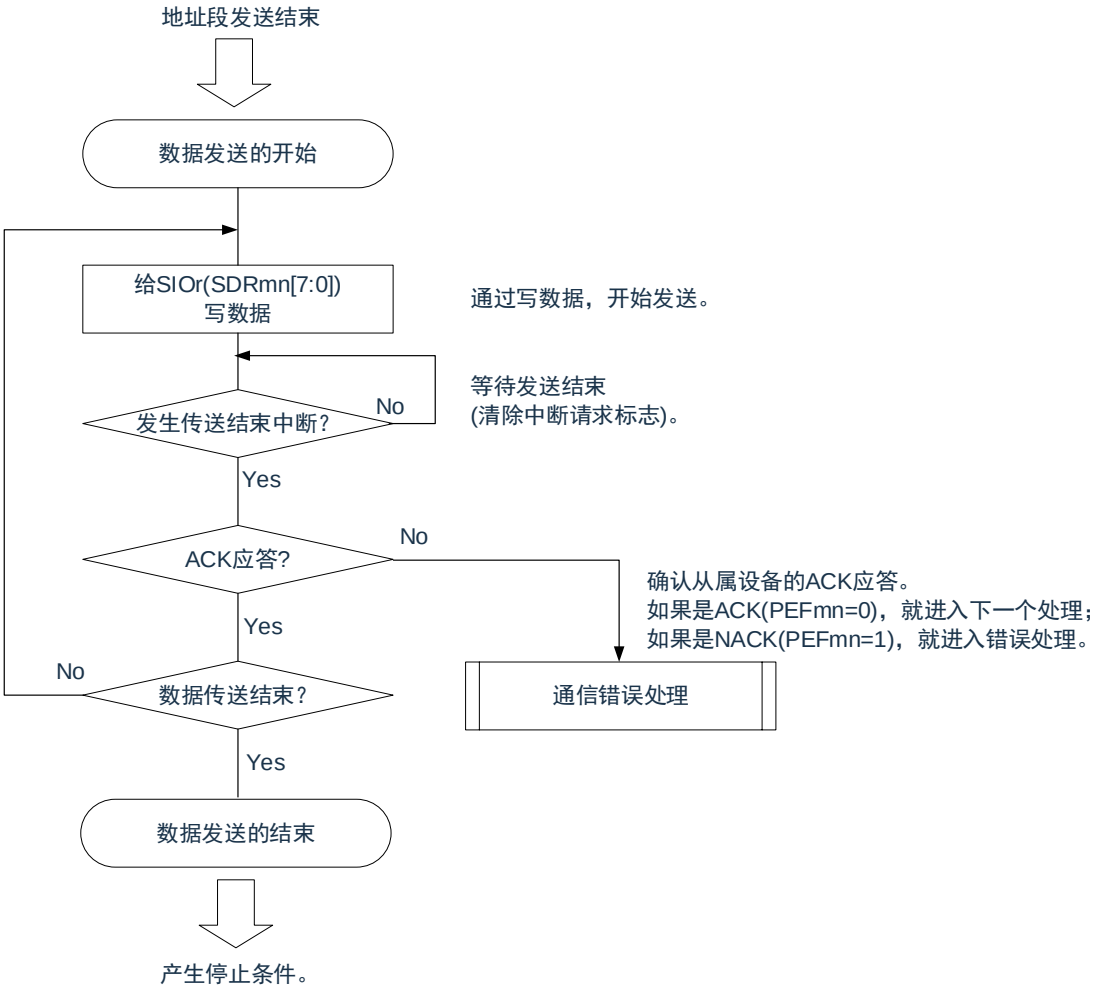


图 16-128：数据发送的流程图



16.9.3 数据接收

数据接收是在发送地址段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

简易 I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
对象通道	SCI0 的通道 0	SCI0 的通道 1	SCI1 的通道 0	SCI1 的通道 1	SCI2 的通道 0	SCI2 的通道 1
使用的引脚	SCL00 SDA00 注1	SCL01 SDA01 注1	SCL10 SDA10 注1	SCL11 SDA11 注1	SCL20 SDA20 注1	SCL21 SDA21 注1
中断	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	只限于传送结束中断（不能选择缓冲器空中断）。					
错误检测标志	只有溢出错误检测标志（OVFmn）。					
传送数据长度	8 位					
传送速率注2	Max.F_{MCK}/4[Hz]（SDRmn[15:9]≥1）F_{MCK}：对象通道的运行时钟频率但是，必须在 I²C 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）。

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、2）n：通道号（n=0、1）

(1) 寄存器的设定

图 16-129：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 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 注1	0	SISmn0 0 注1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 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	15	14	13	6	15	14	13	12		
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 注2	SLCmn0 1	DLSmn3 0	DLSmn2 1	DLSmn1 1 注3	DLSmn0 1

(c) 串行数据寄存器mn(SDRmn)　　(低8位：SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	波特率的设定 注4							0	虚拟发送数据的设定（FFH）							

SIOr

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 注5	CKOm0 0/1 注5	0	0	0	0	0	0	SOm1 0/1 注5	SOm0 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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

注 1：只限于 SMR01、SMR11、SMR21 寄存器。

注 2：只限于 SCR00、SCR10、SCR20 寄存器。

注 3：只限于 SCR00、SCR01、SCR10、SCR11、SCR20、SCR21 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。

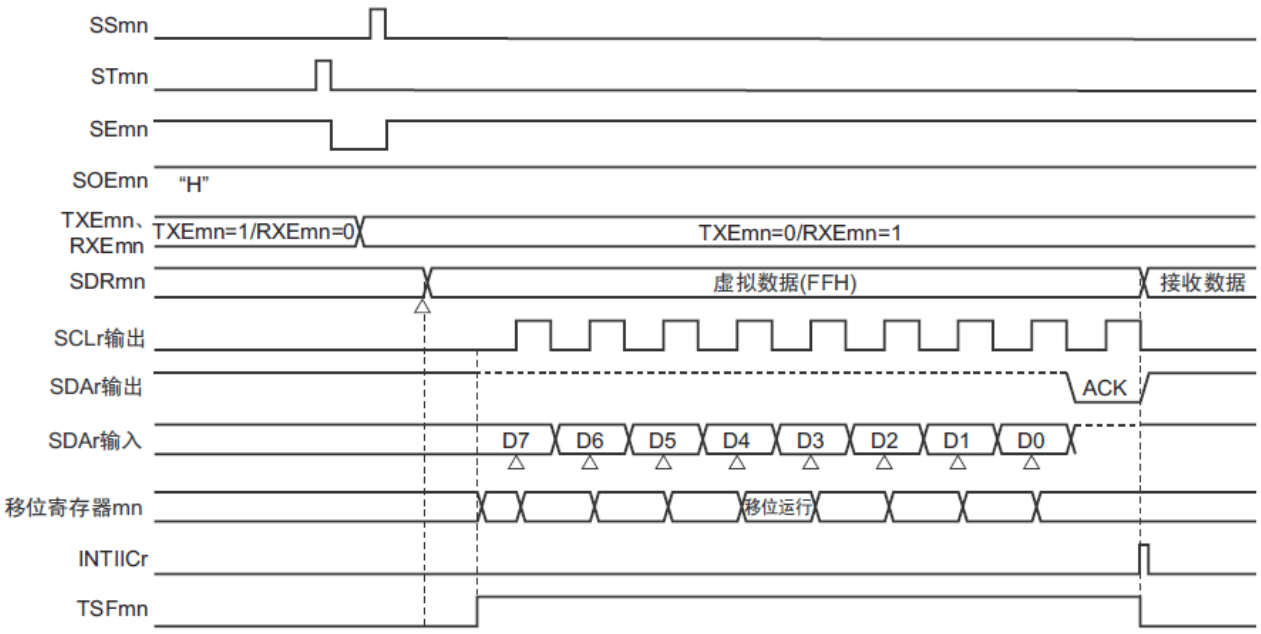
备注：

1. m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）
2. □：在 IIC 模式中为固定设定。■：不能设定（设定初始值）。
   ×：这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。
   0/1：根据用户的用途置“0”或者“1”。

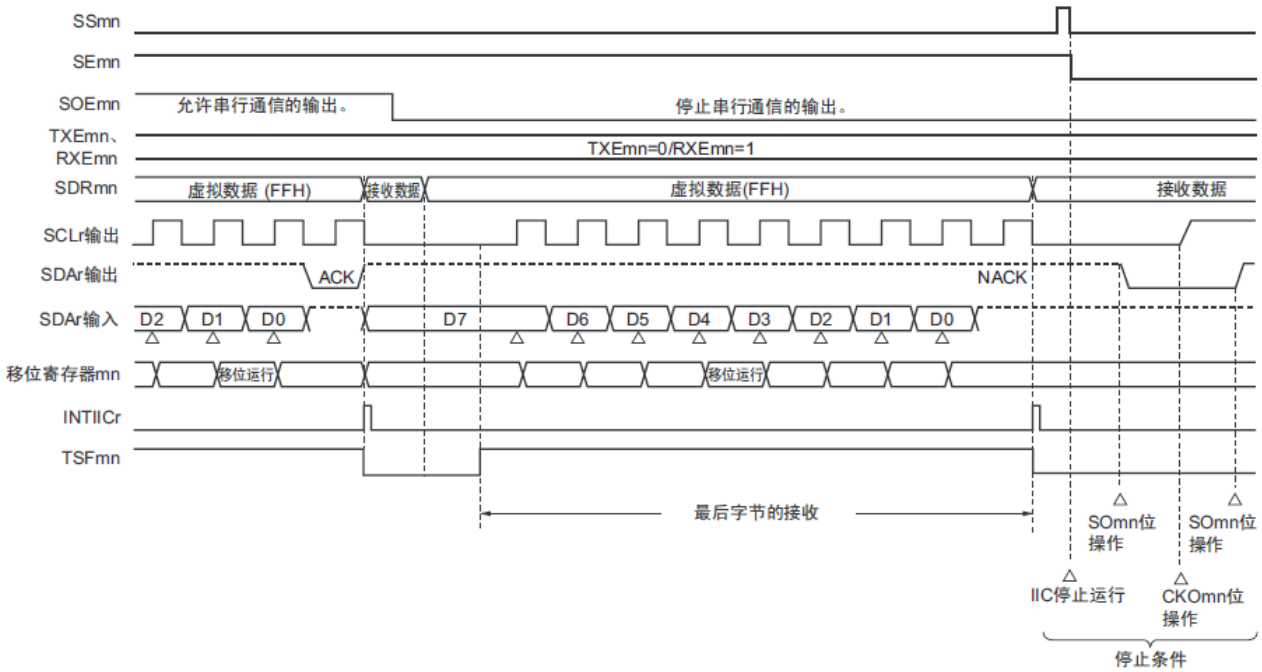
(2) 处理流程

图 16-130：数据接收的时序图

(a) 开始接收数据的情况

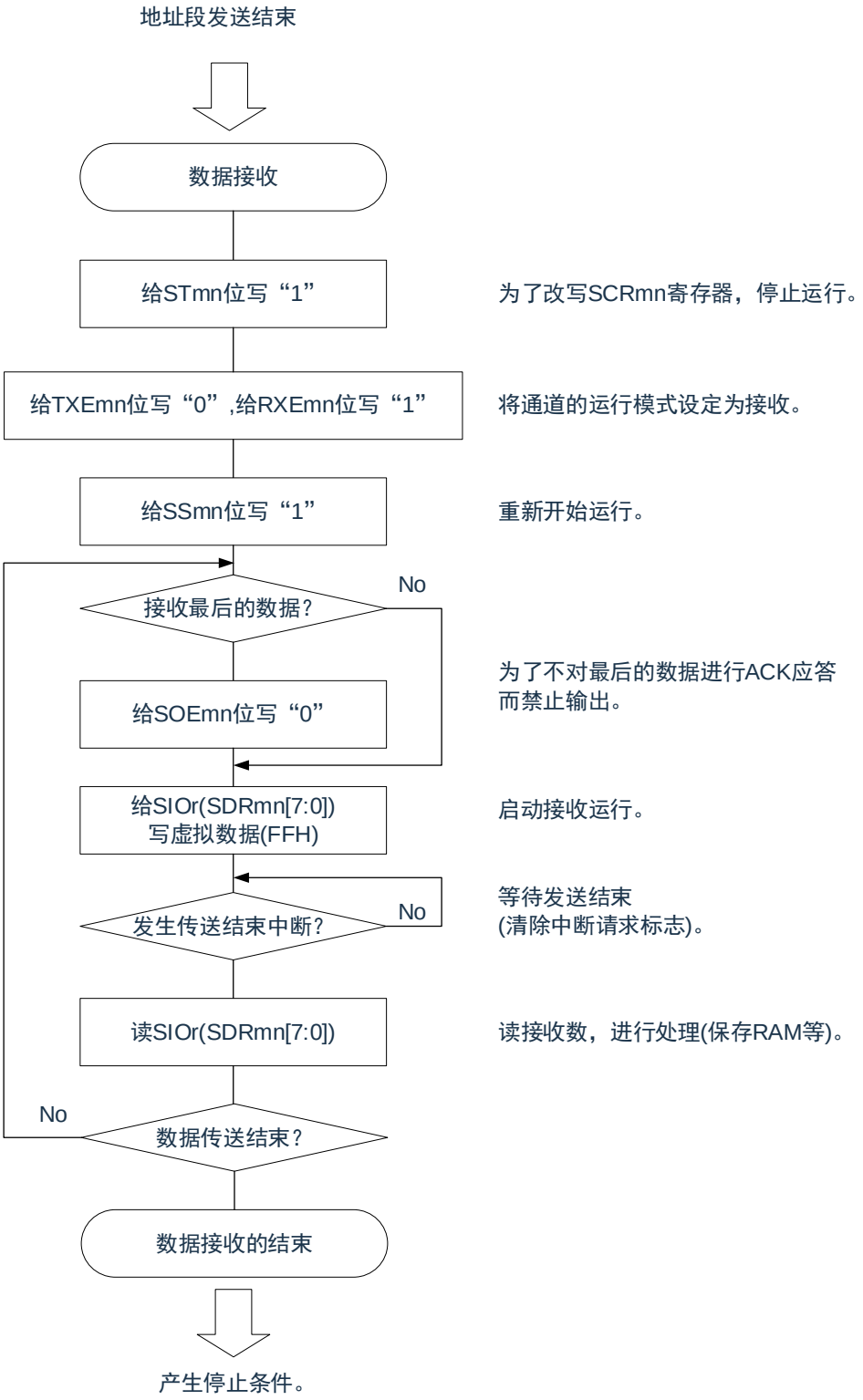


(b) 接收最后数据的情况



备注：m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）

图 16-131：数据接收的流程图



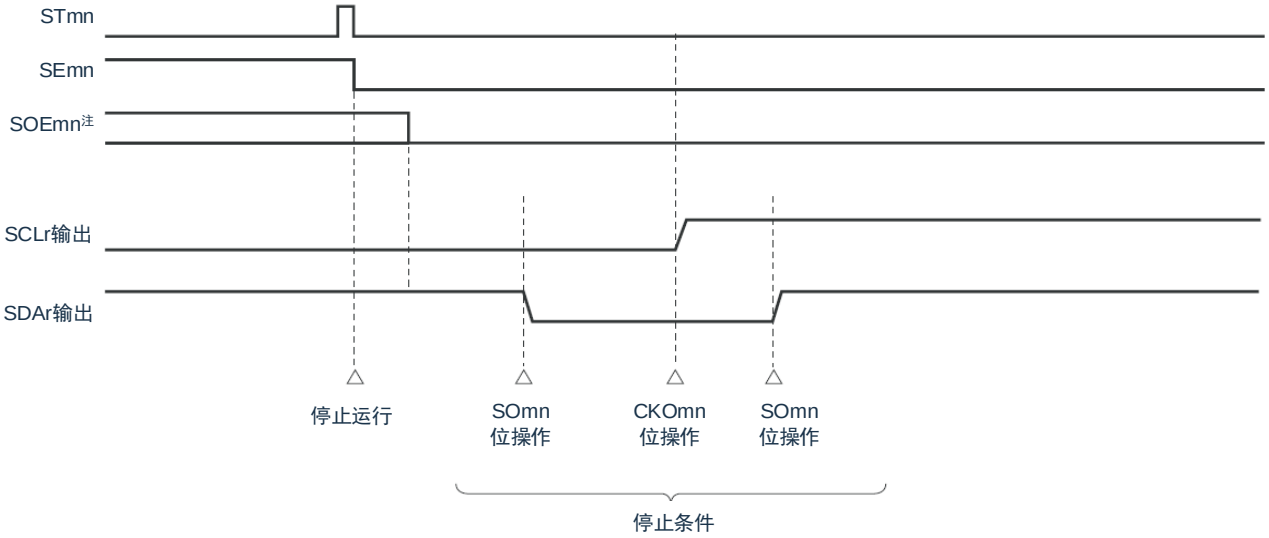
注意：在接收最后的数据时不输出 ACK（NACK）。此后，先通过将串行通道停止寄存器 m（STm）的 STmn 位置“1”来停止运行，然后产生停止条件来结束通信。

16.9.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。

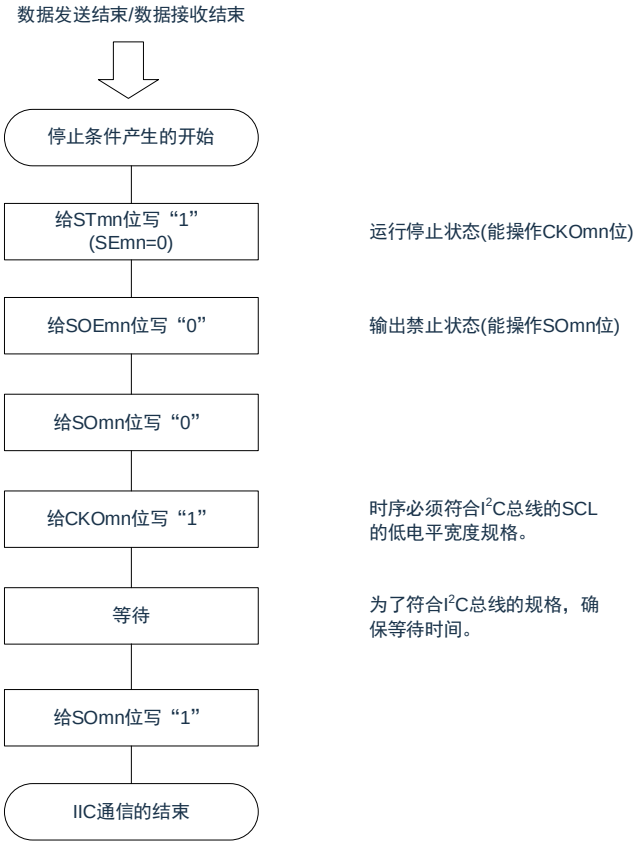
(1) 处理流程

图 16-132：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 16-133：产生停止条件的流程图



16.9.5 传送速率的计算

简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信的传送速率能用以下计算式进行计算。

（传送速率）={对象通道的运行时钟（F_{MCK}）频率}÷(SDRmn[15:9]+1)÷2

注意：禁止将 SDRmn[15:9]设定为“0000000B”，SDRmn[15:9]的设定值必须大于等于“0000001B”。简易 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。

备注：

1. 因为 SDRmn[15:9]的值为串行数据寄存器（SDRmn）的 bit15~9 的值（0000001B~1111111B），所以为 1~127。
2. m：单元号（m=0、1、2）n：通道号（n=0、1）

运行时钟（F_{MCK}）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn 位）

表 16-6：简易 I²C 运行时钟的选择

SMRmn 寄存器	SPSm寄存器								运行时钟（F_{MCK}）注	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		F_{CLK}=32MHz运行时
0	X	X	X	X	0	0	0	0	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：要更改被选择为 F_{CLK} 的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止通用串行通信单元（SCI）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注：

1. X：忽略
2. m：单元号（m=0、1、2）n：通道号（n=0、1）

F_{MCK}=F_{CLK}=32MHz 时的 I²C 传送速率的设定例子如下所示：

I²C 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。

16.9.6 在简易I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信过程中发生错误时的处理步骤

在简易 I²C（IIC00、IIC01、IIC10、IIC11、IIC20、IIC21）通信过程中发生错误时的处理步骤如图 15-134 和图 16-135 所示。

图 15-134：发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SDIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SDIRmn 寄存器，只能清除读操作时的错误。

图 16-135：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。	

注：m：单元号（m=0、1、2）n：通道号（n=0、1）r：IIC 号（r=00、01、10、11、20、21）

第17章 串行接口SPI

17.1 串行接口SPI的功能

本产品搭载一个串行接口 SPI，有以下 2 种模式。

(1) 运行停止模式

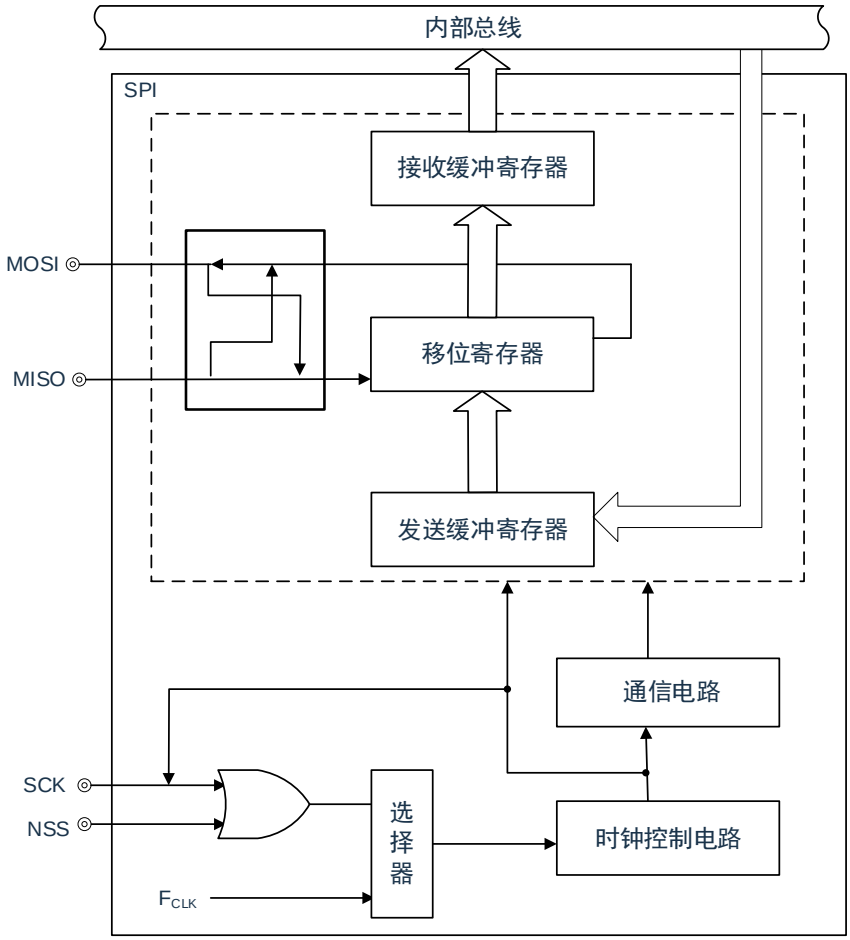
这是用于不进行串行传送时的模式，能降低功耗。

(2) 3-wire 串行 I/O 模式

此模式通过串行时钟（SCK）和串行数据总线（MISO 和 MOSI）的 3 条线，与多个设备进行 8 位或 16 位数据传送。

17.2 串行接口SPI的结构

图 17-1：串行接口 SPI 的框图



17.3 控制串行接口SPI的寄存器

通过以下寄存器控制串行接口 SPI。

- 外围允许寄存器1（PER1）
- 串行操作模式寄存器（SPIM）
- 串行时钟选择寄存器（SPIC）
- 发送缓冲寄存器（SDRO）
- 接收缓冲寄存器（SDRI）

17.3.1 外围允许寄存器1（PER1）

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

要使用 SPI 功能时，必须将 SPIHSOEN 置“1”。

详细请参见“4.3.6 外围允许寄存器 1（PER1）”

17.3.2 SPI操作模式寄存器（SPIM）

SPIM 用于选择操作模式并控制操作的允许或禁止。可由 8 位存储操作指令设置 SPIM。

产生复位信号将该寄存器清除为 00H。

图 17-2：模式控制寄存器（SPIM）的格式

地址：0x40047800　　复位后：00H　　R/W [注1]

符号	7	6	5	4	3	2	1	0
SPIM	SPIE	TRMD	NSSE	DIR	INTMD	DLS	RECMD	-

SPIE	SPI运行的允许
0	停止运行。
1	允许运行。

TRMD[注3]	发送/接收模式控制
0	接收模式
1	发送/接收模式

NSSE[注4]	NSS引脚使用选择
0	未使用NSS引脚
1	使用NSS引脚

DIR	数据传送顺序选择
0	进行MSB优先的输入/输出。
1	进行LSB优先的输入/输出。

INTMD	中断源选择
0	传送结束中断
1	发送缓冲器空中断

DLS	数据长度的设定
0	8位数据长度
1	16位数据长度

RECMD	接收模式的模式选择
0	单次接收
1	连续接收

注 1：SPTF=1（串行通信期间）时，禁止重写 TRMD,DIR,NSSE。

注 2：TRMD 为 0 时 MO 或 SO 输出固定为低电平。

注 3：将该位置为 1 之前，将 NSS 引脚输入电平固定为 0 或者 1。

17.3.3 SPI时钟选择寄存器（SPIC）

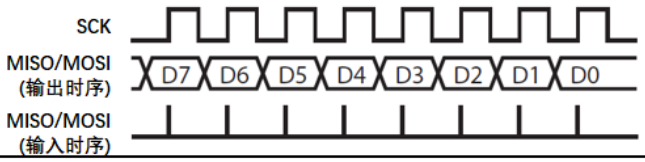
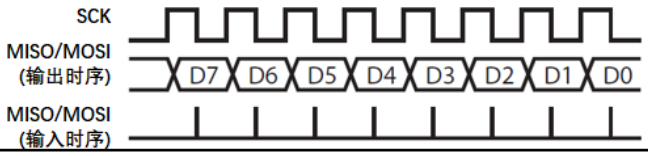
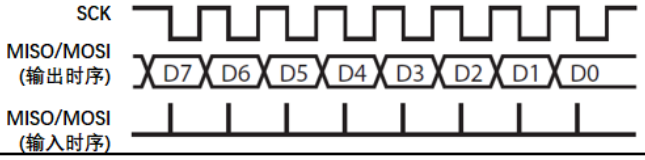
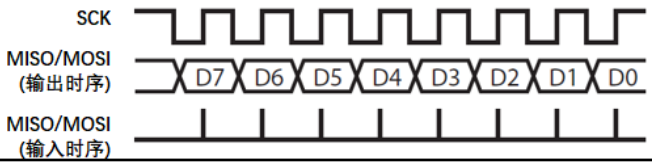
该寄存器指定数据发送/接收的时序，并设置串行时钟。SPIC 可由 8 位存储操作指令设置。

产生复位信号将该寄存器清除为 01H。

图 17-3：时钟选择寄存器（SPIC）的格式

地址：0x40047804　　复位后：01H　　R/W

符号	7	6	5	4	3	2	1	0
SPIC	0	0	0	CPOL	CPHA	CKS2	CKS1	CKS0

CPOL	CPHA	数据发送/接收时序的指定
0	0	
0	1	
1	0	
1	1	

CKS2	CKS1	CKS0	SPI 串行时钟选择	模式
0	0	0	F_{CLK}	主机模式
0	0	1	$F_{CLK}/2$	
0	1	0	$F_{CLK}/2^2$	
0	1	1	$F_{CLK}/2^3$	
1	0	0	$F_{CLK}/2^4$	
1	0	1	$F_{CLK}/2^5$	
1	1	0	$F_{CLK}/2^6$	
1	1	1	从 SCK 输入的外部时钟	从机模式

注意：

1. SPIE=1（操作使能）时禁止写入 SPIC。
2. 复位后数据时钟的相位类型为类型 1。

17.3.4 SPI状态寄存器（SPIS）

SPIT 寄存器用于确认 SPI 的通信状态。可由 8 位存储操作指令读取 SPIS。

产生复位信号将该寄存器清除为 00H。

图 17-4：状态寄存器（SPIS）的格式

地址：0x40047810　　复位后：00H　　R

符号	7	6	5	4	3	2	1	0
SPIS	-	-	-	-	-	-	SDRIF	SPTF

SDRIF	接收缓冲器非空标志位
0	接收缓存里没有新接收到的有效数据
1	接收缓存里有接收到的有效数据。在读取寄存器SDRIF时，该位清0

SPTF[注1]	通信状态标志位
0	通信停止
1	通信正在进行中

注 1：SPTF=1（串行通信期间）时，禁止重写 TRMD,DIR,NSSE。

17.3.5 发送缓冲寄存器（SDRO）

该寄存器设置发送数据。

当将串行操作模式寄存器（SPIM）的位 7（SPIE）和位 6（TRMD）置为 1 时，通过将数据写入 SDRO 开始发送/接收。

串行 I/O 移位寄存器把 SDRO 中的数据从并行数据转换成串行数据，并输出到串行输出引脚。

可用 8 位或 16 位存储操作指令写入或读取 SDRO。

产生复位信号将该寄存器清除为 0000H。

图 17-5：发送缓冲寄存器（SDRO）的格式

地址：0x40047808　　复位后：0000H　　R/W

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

17.3.6 接收缓冲寄存器（SDRI）

该寄存器存储接收到的数据。

如果将串行操作模式寄存器（SPIM）的位 6（TRMD）置为 0，则通过从 SDRI 中读取数据开始接收。

接收期间，将数据从串行输入引脚读入 SDRI 中。

可用 8 位或 16 位存储操作指令读取 SDRI。

产生复位信号将该寄存器清除为 0000H。

图 17-6：接收缓冲寄存器（SDRI）的格式

地址：0x4004780C　　复位后：0000H　　R

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

17.4 串行接口SPI的操作

3 线串行 I/O 模式中，数据以 8 位或 16 位为单位发送或接收。数据各位的发送或接收与串行时钟同步。

开始通信后，SPIT 的位 0（SPTF）被置为 1。当数据的通信已完成时，设置通信完成中断请求标志（SPIIF），并将 SPTF 清除为 0。然后使能下一次通信。

注意：

1. SPTF=1（串行通信期间）时，禁止访问控制寄存器和数据寄存器。
2. 必须在满足 SCLK 周期时间（T_{KCY}）特性的范围内使用。详细内容请参照数据手册。

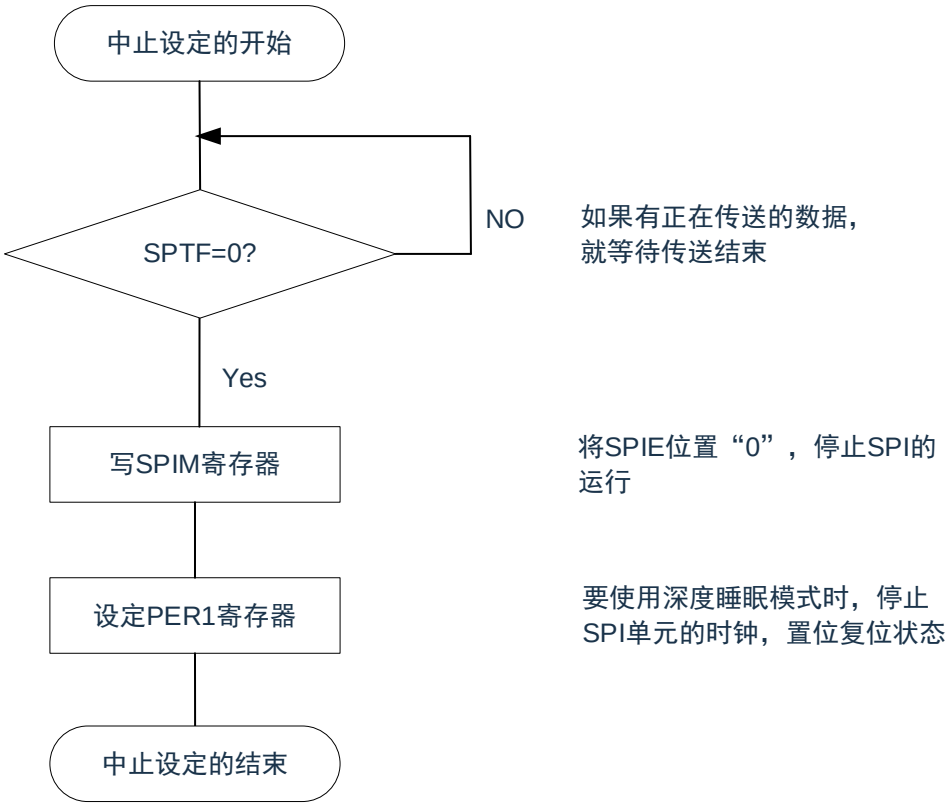
17.4.1 主控的发送和接收

如果串行操作模式寄存器（SPIM）的位 6（TRMD）为 1，则可以发送或接收数据。当将某个值写入发送缓冲寄存器（SDRO）时，开始发送/接收。

(1) 操作步骤

图 17-7：主控发送/接收的初始设置步骤

图 17-8：主控发送/接收的中止步骤



(2) 处理流程

图 17-9：发送/接收时序(单次发送模式)的时序图（INTMD=0,CPHA=1、CPOL=1）

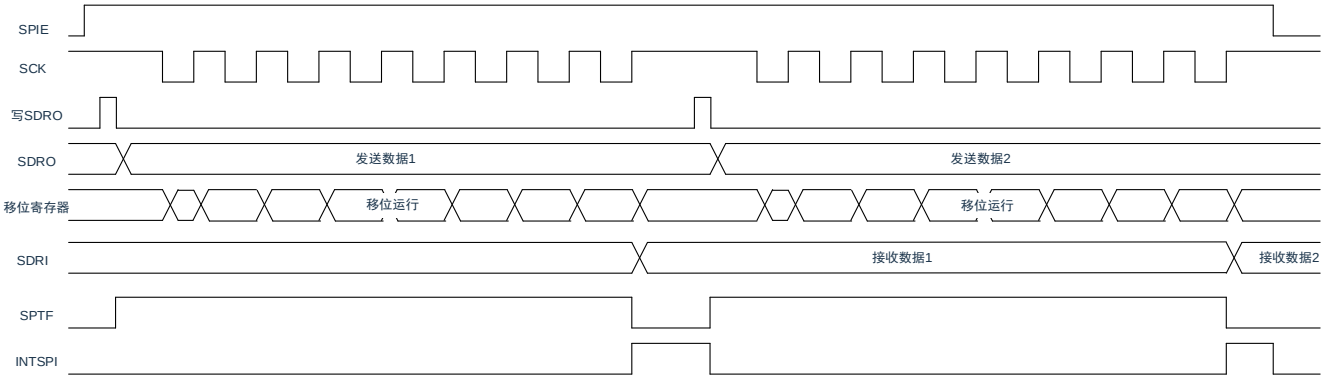
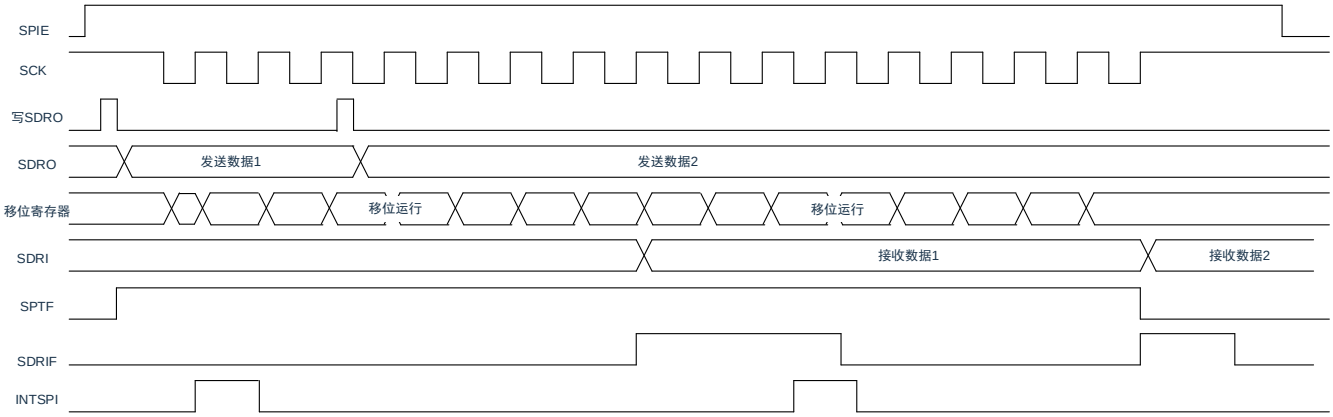


图 17-10：发送/接收时序(连续发送模式)的时序图（INTMD=1,CPHA=1、CPOL=1）



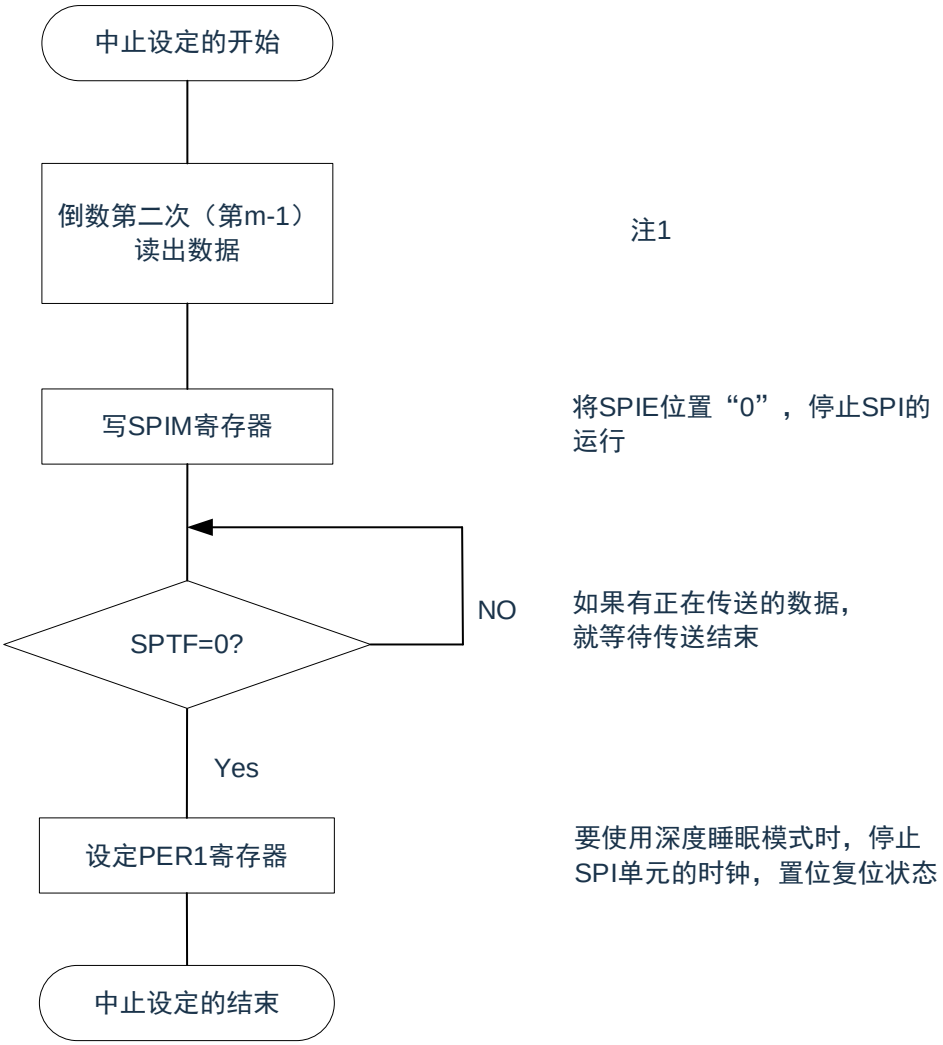
17.4.2 主控的接收

如果串行操作模式寄存器（SPIM）的位 6（TRMD）为 0，则可以只接收数据。当从接收缓冲寄存器（SDRI）中读取数据时，开始接收。

(1) 操作步骤

图 17-11：主控接收的初始设置步骤

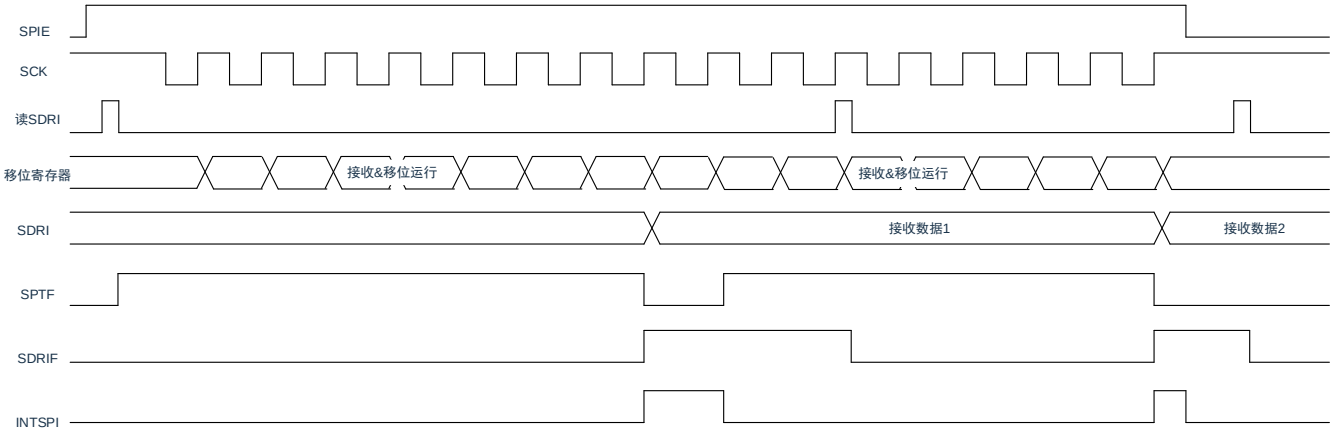
图 17-12：主控接收的中止步骤



注 1：只接收模式下，SPI 传输由读取 SDRI 寄存器的值触发。如果不及时中止 SPI 的动作，可能会在最后一次读取 SDRI 之后有一次冗余的传输。如果想避免最后一次冗余的传输，可以在倒数第二次读出数据之后，等待一个 SCK 周期后关闭 SPIE。SPI 的传输将在最后一个数据传输完成后中止。

(2) 处理流程

图 17-13：接收时序的时序图（CPHA=1、CPOL=1）



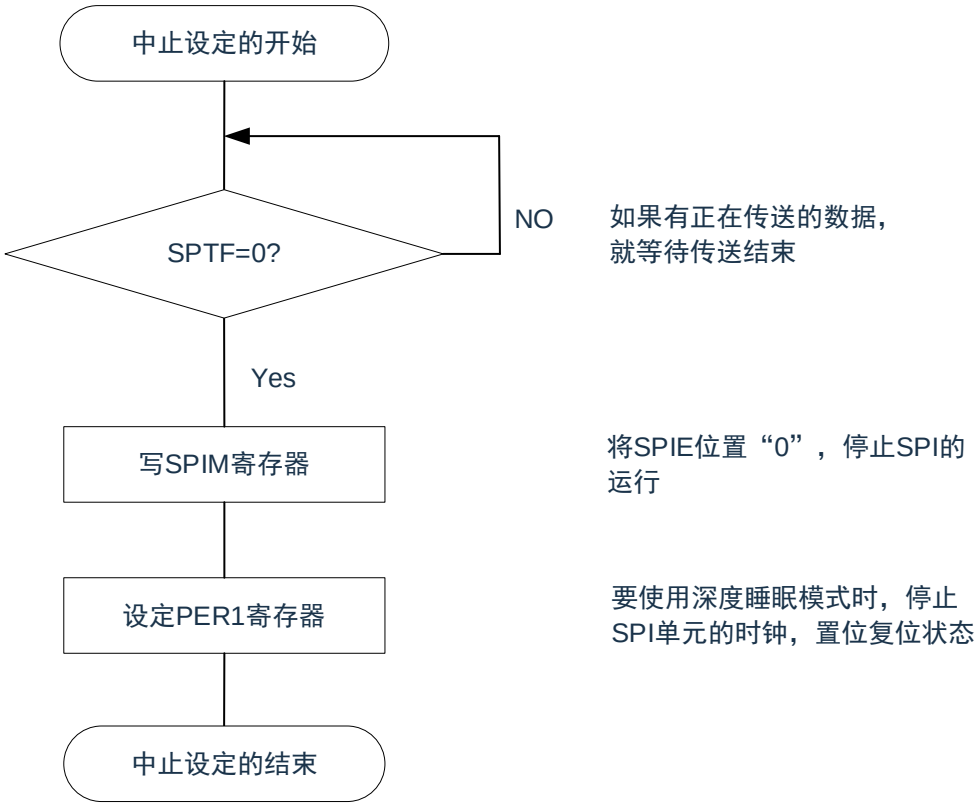
17.4.3 从属的发送和接收

如果串行时钟选择寄存器（SPIC）的 CKS2-0 位选择从机模式，串行操作模式寄存器（SPIM）的位 6（TRMD）为 1，则进入从机发送/接收模式。当将某个值写入发送缓冲寄存器（SDRO）后，等待主控设备的时钟，开始发送/接收。

(1) 操作步骤

图 17-14：从属发送/接收的初始设置步骤

图 17-15：从属发送/接收的中止步骤



(2) 处理流程

图 17-16：发送/接收时序(单次发送模式)的时序图（INTMD=0,CPHA=1、CPOL=1）

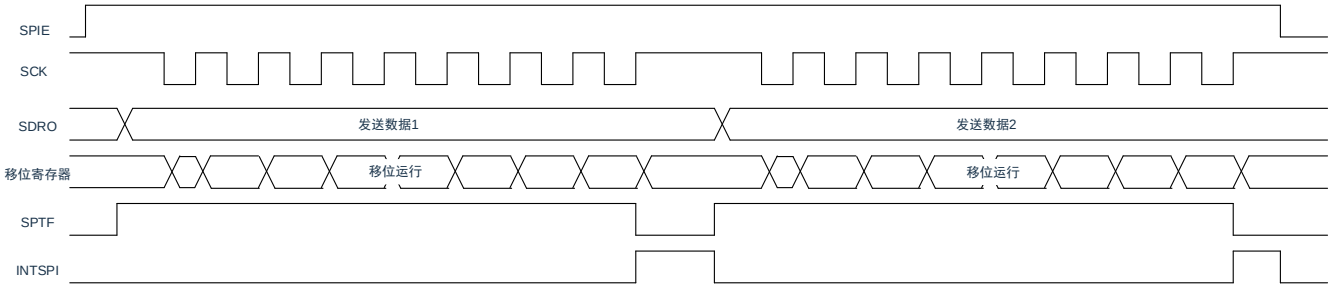
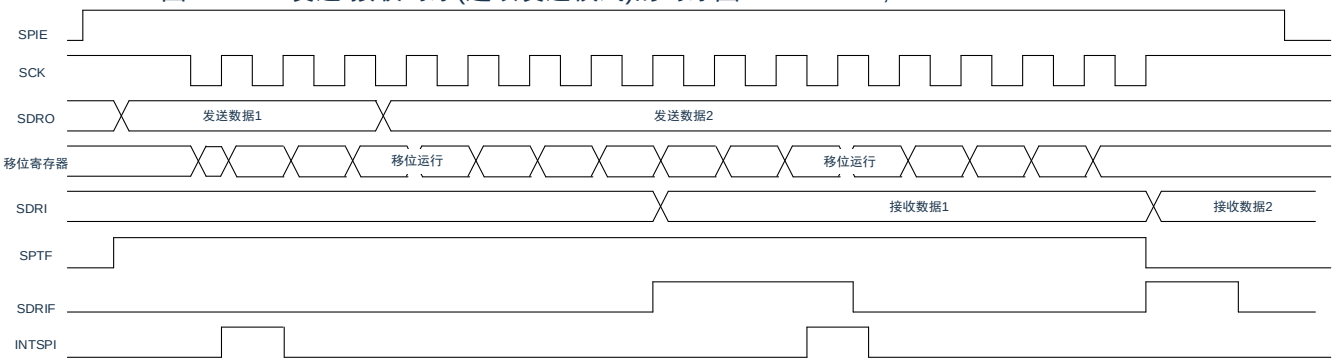


图 17-17：发送/接收时序(连续发送模式)的时序图（INTMD=1, CPHA=1、CPOL=1）



17.4.4 从属的接收

如果串行时钟选择寄存器（SPIC）的 CKS2-0n 位选择从机模式，串行操作模式寄存器（SPIM）的位 6（TRMD）为 0，则进入从机接收模式。当从接收缓冲寄存器（SDRI）中读取数据时，等待主控设备的时钟，开始接收。

(1) 操作步骤

图 17-18：从属接收的初始设置步骤

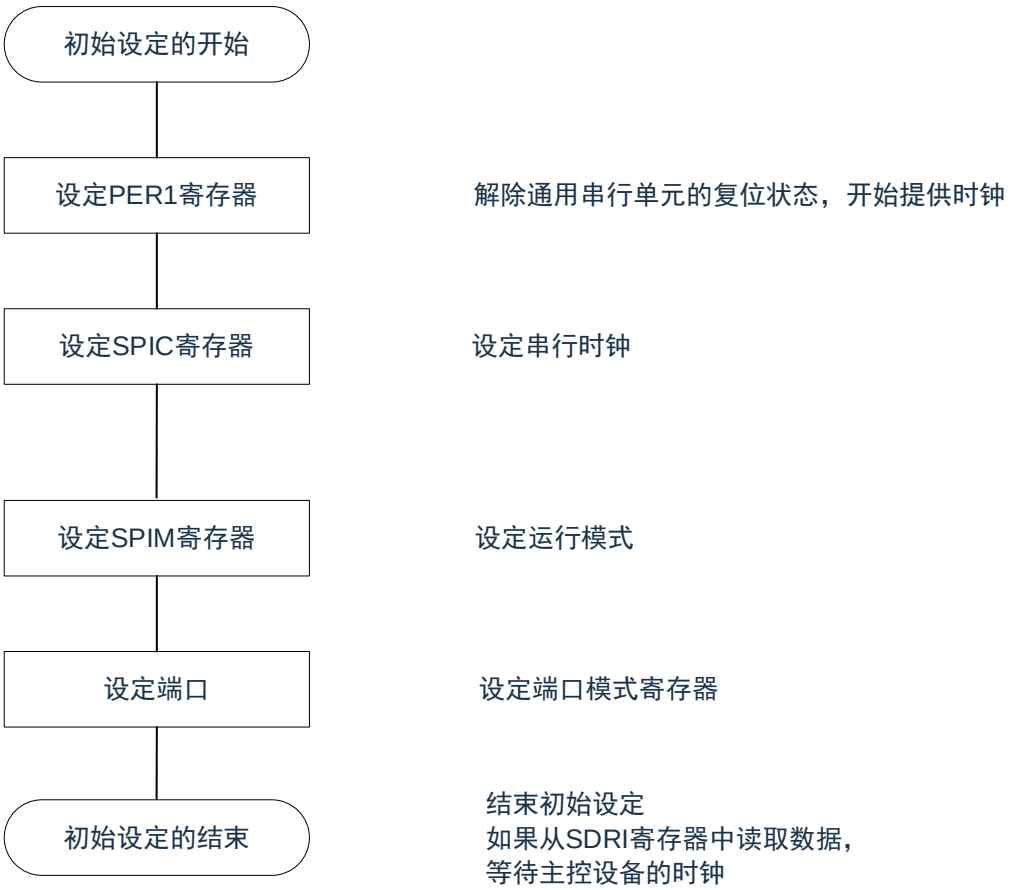
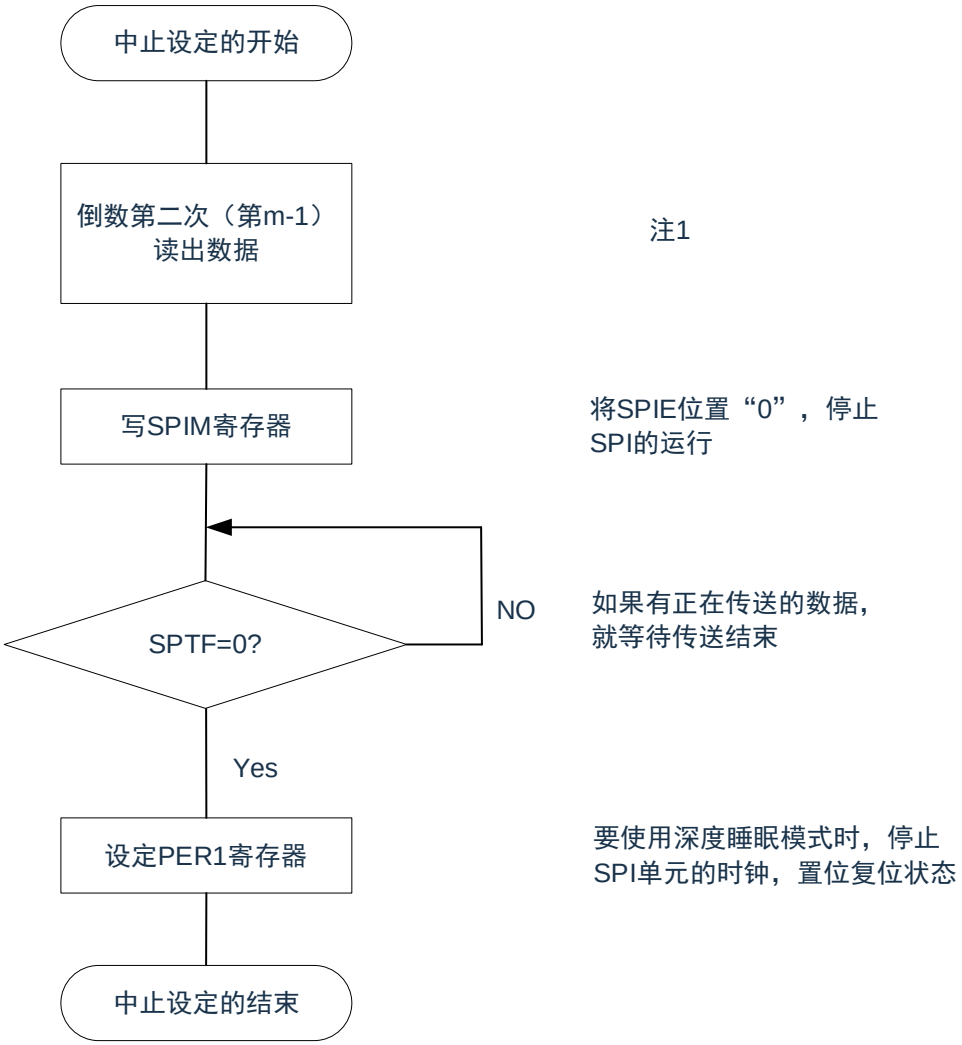


图 17-19：从属接收的中止步骤

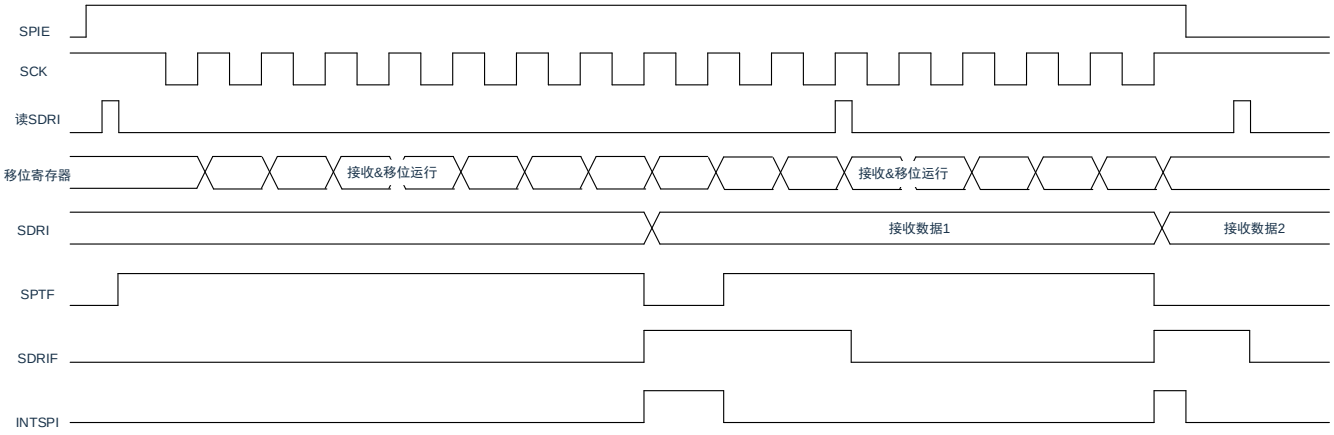


注 1：只接收模式下，SPI 传输由读取 SDRI 寄存器的值触发。如果不及时中止 SPI 的动作，可能会在最后一次读取 SDRI 之后有一次冗余的传输。

如果想避免最后一次冗余的传输，可以在倒数第二次读出数据之后，等待一个 SCK 周期后关闭 SPIE。SPI 的传输将在最后一个数据传输完成后中止。

(2) 处理流程

图 17-20：接收时序的时序图（CPHA=1、CPOL=1）



第18章 串行接口IICA

18.1 串行接口IICA的功能

本产品搭载 1 个串行接口 IICA0，有以下 3 种模式。

18.1.1 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

18.1.2 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 I^2C 总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的 I^2C 总线控制部分。

因为串行接口 IICA 的 SCLAn 引脚和 SDAAn 引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。睡眠模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除深度睡眠模式。通过 IICA 控制寄存器 n1（IICCTLn1）的 WUPn 位进行设定。

串行接口 IICA 的框图如图 18-1 所示。

备注：n=0。

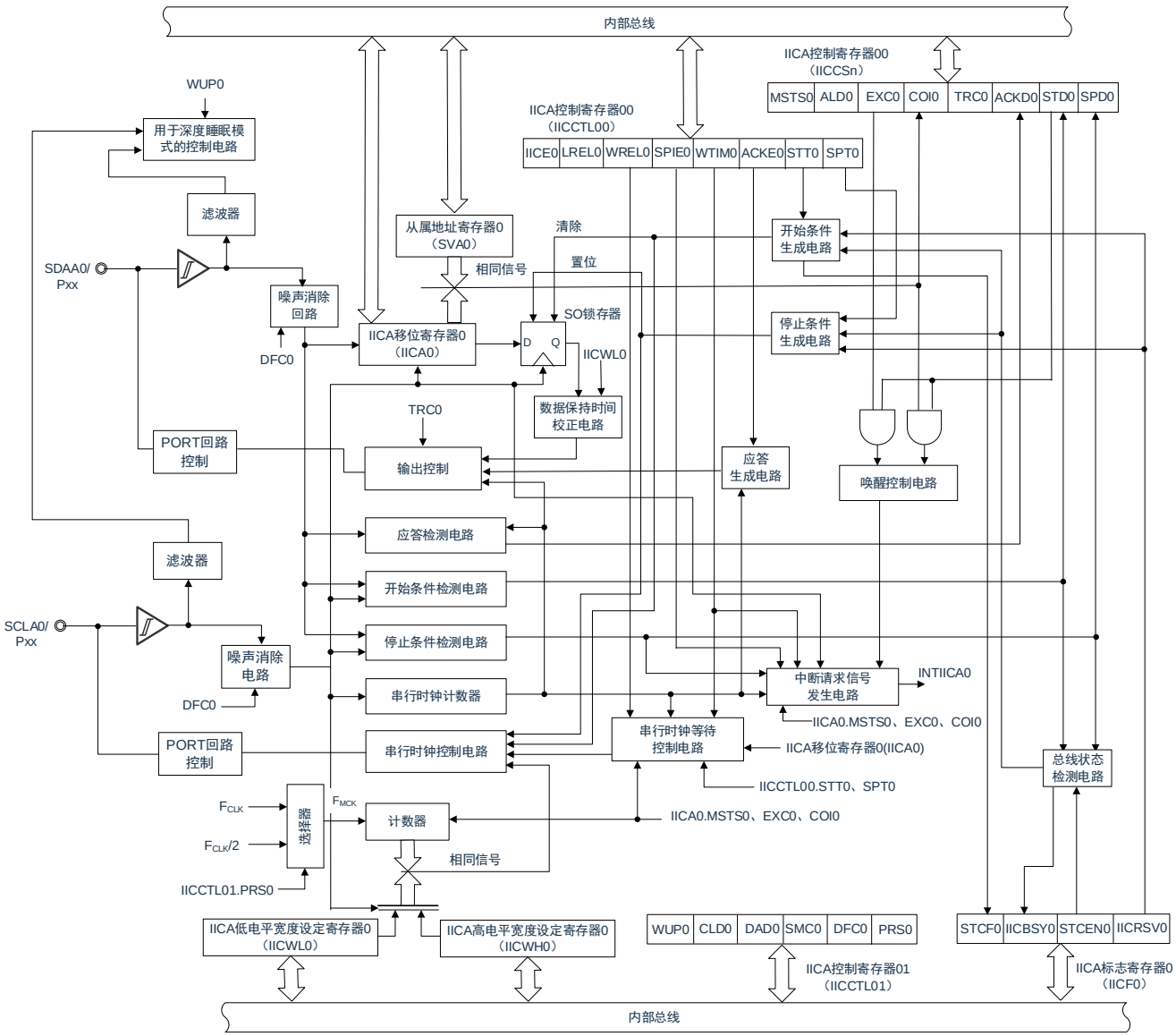
18.1.3 唤醒模式

在深度睡眠模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除深度睡眠模式。通过 IICA 控制寄存器 n1（IICCTLn1）的 WUPn 位进行设置。

串行接口 IICA 的框图如图 18-1 所示。

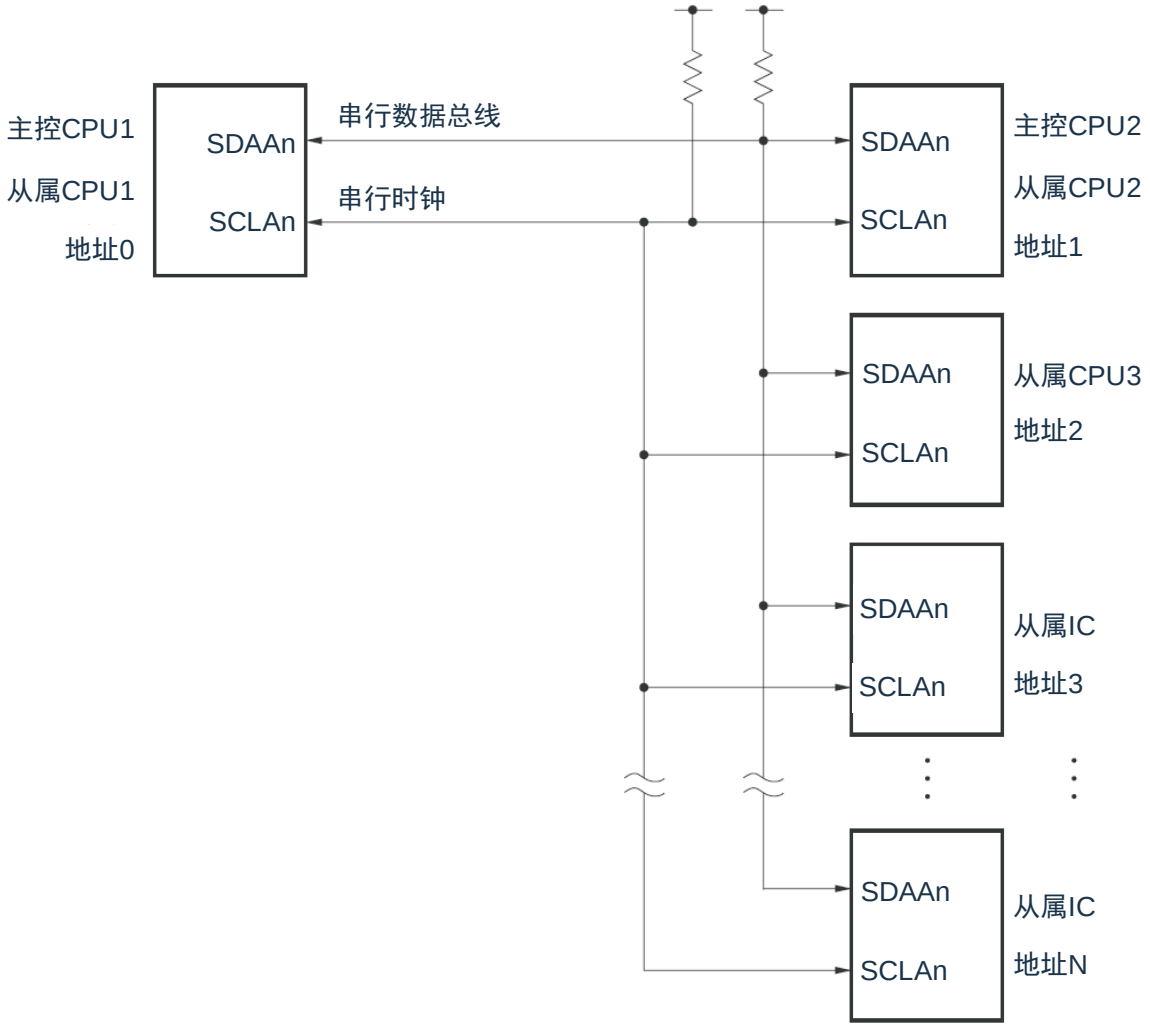
备注：n=0

图 18-1：串行接口 IICA 的框图



串行总线的结构例子如图 18-2 所示。

图 18-2：I2C 总线的串行总线结构例子



备注：n=0。

18.2 串行接口IICA的结构

串行接口 IICA 由以下硬件构成。

表 18-1：串行接口 IICA 的结构

项目	结构
寄存器	IICA移位寄存器n（IICAn） 从属地址寄存器n（SVAn）
控制寄存器	外围允许寄存器0（PER0） IICA控制寄存器n0（IICCTLn0） IICA状态寄存器n（IICSn） IICA标志寄存器n（IICFn） IICA控制寄存器n1（IICCTLn1） IICA 低电平宽度设定寄存器 n（IICWLn）IICA高电平宽度设定寄存器 n（IICWHn）端口模式寄存器（PMxx） 端口模式控制寄存器（PMCxx） 端口复用功能配置寄存器（PxxCFG）

备注：

1. n=0。
2. 本产品可将 IICA 输入/输出引脚功能复用到多个端口。当某端口被配置为 IICA 引脚的复用功能后，该端口的 N 沟道漏极开路输出（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。

寄存器列表：

基地址	偏移地址	寄存器名称	R/W	复位值
0x40047000	0x000	IICCTL00	R/W	00H
	0x001	IICCTL01	R/W	00H
	0x002	IICWL0	R/W	FFH
	0x003	IICWH0	R/W	FFH
	0x004	SVA0	R/W	00H
	0x005	IICSE0	R	00H
	0x008	IICA0	R/W	00H
	0x009	IICS0	R	00H
	0x00A	IICF0	R/W	00H

18.2.1 IICA移位寄存器n（IICAn）

IICAn 寄存器是与串行时钟同步进行 8 位串行数据和 8 位并行数据相互转换的寄存器，用于发送和接收。能通过读写 IICAn 寄存器来控制实际的发送和接收。

在等待期间，通过写 IICAn 寄存器来解除等待，开始传送数据。通过 8 位存储器操作指令设定 IICAn 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 18-3：IICAn 移位寄存器 n（IICAn）的格式

复位后：00H，R/W

符号	7	6	5	4	3	2	1	0
IICAn								

注意：

1. 在数据传送过程中，不能给 IICAn 寄存器写数据。
2. 只能在等待期间读写 IICAn 寄存器。除了等待期间以外，禁止在通信状态下存取 IICAn 寄存器。但是，在主控设备的情况下，能在将通信触发位（STTn）置“1”后写一次 IICAn 寄存器。
3. 当预约通信时，必须在检测到由停止条件产生的中断后给 IICAn 寄存器写数据。

备注：n=0。

18.2.2 从属地址寄存器n（SVAn）

这是在用作从属设备时保存 7 位本地站地址{A6,A5,A4,A3,A2,A1,A0}的寄存器。

通过 8 位存储器操作指令设定 SVAn 寄存器。但是，在 STDn 位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 18-4：从属地址寄存器 n（SVAn）的格式

复位后：00H，R/W

符号	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0[注]

注：bit0 固定为“0”。

18.2.3 SO锁存器

SO 锁存器保持 SDAAn 引脚的输出电平。

18.2.4 唤醒控制电路

当设定在从属地址寄存器 n（SVAn）的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求（INTIICAn）。

18.2.5 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了 8 位数据的发送和接收。

18.2.6 中断请求信号发生电路

此电路控制产生中断请求信号（INTIICAn）。由以下 2 种触发产生 I^2C 中断请求。

- 第 8 个或者第 9 个串行时钟的下降（通过 WTIMn 位进行设定）
- 因检测到停止条件而产生中断请求（通过 SPIEn 位进行设定）。

备注：WTIMn 位：IICA 控制寄存器 n0（IICCTLn0）的 bit3
SPIEn 位：IICA 控制寄存器 n0（IICCTLn0）的 bit4

18.2.7 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到 SCLAn 引脚的时钟。

18.2.8 串行时钟等待控制电路

此电路控制等待时序。

18.2.9 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

18.2.10 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

18.2.11 开始条件生成电路

如果将 STTn 位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下（IICRSVn 位=1）并且没有释放总线（IICBSYn 位=1）时，忽视开始条件请求并且将 STCFn 位置“1”。

18.2.12 停止条件生成电路

如果将 SPTn 位置“1”，此电路就生成停止条件。

18.2.13 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。但是，在刚运行时不能立即检测总线状态，因此必须通过 STCENn 位设定总线状态检测电路的初始状态。

备注：

1. STTn 位：IICA 控制寄存器 n0（IICCTLn0）的 bit1
   SPTn 位：IICA 控制寄存器 n0（IICCTLn0）的 bit0
   IICRSVn 位：IICA 标志寄存器 n（IICFn）的 bit0
   IICBSYn 位：IICA 标志寄存器 n（IICFn）的 bit6STCFn 位：IICA 标志寄存器 n（IICFn）的 bit7
   STCENn 位：IICA 标志寄存器 n（IICFn）的 bit1
2. n=0

18.3 控制串行接口IICA的寄存器

通过以下几种寄存器控制串行接口 IICA。

- 外围允许寄存器 0（PER0）
- IICA控制寄存器n0（IICCTLn0）
- IICA标志寄存器n（IICFn）
- IICA状态寄存器n（IICSn）
- IICA控制寄存器n1（IICCTLn1）
- IICA 低电平宽度设定寄存器 n（IICWLn）
- IICA 高电平宽度设定寄存器 n（IICWHn）

备注：n=0

18.3.1 外围允许寄存器0（PER0）

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

要使用串行接口 IICA0 时，必须将 bit6（IICA0EN）置“1”。

通过 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 18-5：外围允许寄存器 A（PER0）的格式

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

地址：40020424H复位后：00000000H R/W

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

注意：要设定串行接口 IICA 时，必须先在 IICAEN 位为“1”的状态下设定以下的寄存器。当 IICAEN 位为“0”时，串行接口 IICA 的控制寄存器的值为初始值，忽视写操作（端口复用功能配置寄存器（PxxCFG）、端口模式寄存器（PMxx）和端口模式控制寄存器（PMCxx）除外）。

- IICA 控制寄存器 n0（IICCTLn0）
- IICA 标志寄存器 n（IICFn）
- IICA 状态寄存器 n（IICSn）
- IICA 控制寄存器 n1（IICCTLn1）
- IICA 低电平宽度设定寄存器 n（IICWLn）
- IICA 高电平宽度设定寄存器 n（IICWHn）

18.3.2 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”。

备注：n=0。

图 18-6：IICA 控制寄存器 n0（IICCTLn0）的格式(1/4)

复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	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。
必须在SCLAn线和SDAAn线为高电平的状态下将此位置“1”。	
清除条件（IICEn=0）	置位条件（IICEn=1）
• 通过指令清除。 • 当复位时	• 通过指令置位。

LRELn[注2,3]	通信的退出
0	通常运行
1	退出当前的通信，进入待机状态。执行后自动清“0”。 在接收到与本地站无关的扩展码等情况下使用。 SCLAn线和SDAAn线变为高阻状态。 IICA控制寄存器n0（IICCTLn0）和IICA状态寄存器n（IICSn）中的以下标志被清“0”： •STTn•SPTn•MSTSn•EXCn•COIn•TRCn•ACKDn•STDn
变为退出通信的待机状态，保持到满足以下的通信参加条件为止。 • 在检测到停止条件后作为主控设备启动。 • 在检测到开始条件后地址匹配或者接收到扩展码。	
清除条件（LRELn=0）	置位条件（LRELn=1）
• 在执行后自动清除。 • 当复位时	• 通过指令置位。

WRELn[注2,3]	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下（TRCn=1）的第9个时钟等待期间将WRELn位（解除等待）置位，SDAAn线就变为高阻抗状态（TRCn=0）。	
清除条件（WRELn=0）	置位条件（WRELn=1）
• 在执行后自动清除。 • 当复位时	• 通过指令置位。

注 1：对 IICA 移位寄存器 n（IICAn）、IICA 标志寄存器 n（IICFn）的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1（IICCTLn1）的 CLDn 位和 DADn 位进行复位。

注 2：在 IICEn 位为“0”的状态下，此位的信号无效。

注 3：LRELn 位和 WRELn 位的读取值总是“0”。

注意：如果在 SCLAn 线为高电平、SDAAn 线为低电平并且数字滤波器为 ON（IICCTLn1 寄存器的 DFCn=1）时允许 I²C 运行（IICEn=1），就立即检测开始条件。此时，必须在允许 I²C 运行（IICEn=1）后连续通过位存储器操作指令将 LRELn 位置“1”。

备注：n=0

图 18-6：IICA 控制寄存器 n0（IICCTLn0）的格式(2/4)

SPIEn[注1]	允许或者禁止停止条件检测产生的中断请求
0	禁止
1	允许
当IICA控制寄存器n1（IICCTLn1）的WUPn位为“1”时，即使将SPIEn位置“1”也不产生停止条件中断。	
清除条件（SPIEn=0）	置位条件（SPIEn=1）
• 通过指令清除。 • 当复位时	• 通过指令置位。

WTIMn[注1]	等待和中断请求的控制
0	在第8个时钟的下降沿产生中断请求信号。 主控设备：在输出8个时钟后，将时钟输出置为低电平进行等待。 从属设备：在输入8个时钟后，将时钟置为低电平，然后等待主控设备。
1	在第9个时钟的下降沿产生中断请求信号。 主控设备：在输出9个时钟后，将时钟输出置为低电平进行等待。 从属设备：在输入9个时钟后，将时钟置为低电平，然后等待主控设备。
在地址传送期间，与此位的设定无关，在第9个时钟的下降沿产生中断；在地址传送结束后，此位的设定有效。主控设备在地址传送期间的第9个时钟下降沿进入等待状态。接收到本地站地址的从属设备在产生应答（ACK）后的第9个时钟下降沿进入等待状态，但是接收到扩展码的从属设备在第8个时钟下降沿进入等待状态。	
清除条件（WTIMn=0）	置位条件（WTIMn=1）
• 通过指令清除。 • 当复位时	• 通过指令置位。

ACKEn[注1,2]	应答控制
0	禁止应答。
1	允许应答。在第9个时钟期间将SDAAn线置为低电平。
清除条件（ACKEn=0）	置位条件（ACKEn=1）
• 通过指令清除。 • 当复位时	• 通过指令置位。

注 1：在 IICEn 位为“0”的状态下，此位的信号无效。必须在此期间设定此位。

注 2：在地址传送过程中不是扩展码时，设定值无效。当为从属设备并且地址匹配时，与设定值无关而生成应答。

备注：n=0

图 18-6：IICA 控制寄存器 n0（IICCTLn0）的格式(3/4)

STTn[注1,2]	开始条件的触发
0	不生成开始条件。
1	当总线被释放时（待机状态，IICBSYn位为“0”）：如果将此位置“1”，就生成开始条件（作为主控设备的启动）。当第三方正在通信时： • 允许通信预约功能的情况（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 • 禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除STTn位并且将STTn清除标志（STCFn）置“1”，不生成开始条件。等待状态（主控设备）： 在解除等待后生成重新开始条件。

有关置位时序的注意事项：
- 主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。
- 主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第9个时钟后的等待期间将此位置“1”。
- 禁止与停止条件的触发（SPTn）同时置“1”。
- 在将STTn位置“1”后，禁止在满足清除条件前再次将此位“1”。

清除条件（STTn=0）	置位条件（STTn=1）
• 在禁止通信预约的状态下将STTn位置“1”。 • 在仲裁失败时 • 主控设备生成开始条件。 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 通过指令置位。

注 1：在 IICEn 位为“0”的状态下，此位的信号无效。

注 2：STTn 位的读取值总是“0”。

备注：

1. 如果在设定数据后读 bit1（STTn），此位就变为“0”。
2. IICRSVn：IICA 标志寄存器 n（IICFn）的 bit0
   STCFn：IICA 标志寄存器 n（IICFn）的 bit7
3. n=0

图 18-6：IICA 控制寄存器 n0（IICCTLn0）的格式(4/4)

SPTn[注]	停止条件的触发
0	不生成停止条件。
1	生成停止条件（作为主控设备的传送结束）。

有关置位时序的注意事项：
- 主控接收：禁止在传送过程中将此位置“1”。只有在将ACKEn位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。
- 主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第9个时钟后的等待期间将此位置“1”。
- 禁止与开始条件的触发（STTn）同时置“1”。
- 只有在主控设备的情况下才能将SPTn位置“1”。
- 在WTIMn位为“0”时，必须注意：如果在输出8个时钟后的等待期间将SPTn位置“1”，就在解除等待后的第9个时钟的高电平期间生成停止条件。必须在输出8个时钟后的等待期间将WTIMn位从“0”置为“1”并且在输出第9个时钟后的等待期间将SPTn位置“1”。
- 在将SPTn位置“1”后，禁止在满足清除条件前再次将此位置“1”。

清除条件（SPTn=0）	置位条件（SPTn=1）
• 当仲裁失败时 • 在检测到停止条件后自动清除。 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 通过指令置位。

注：SPTn 位的读取值总是“0”。

注意：在 IICA 状态寄存器 n（IICSn）的 bit3（TRCn）为“1”（发送状态）时，如果在第 9 个时钟将 IICCTLn0 寄存器的 bit5（WRELn）置“1”来解除等待，就在清除 TRCn 位（接收状态）后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1”（发送状态）时的等待解除。

备注：n=0

18.3.3 IICA状态寄存器n（IICSn）

这是表示 I²C 状态的寄存器。

只有在 STTn 位为“1”并且等待期间，才能 8 位存储器操作指令读 IICSn 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

注意：在深度睡眠模式中允许地址匹配唤醒功能（WUPn=1）状态下，禁止读 IICSn 寄存器。在 WUPn 位为“1”的状态下，与 INTIICAn 中断请求无关，如果将 WUPn 位从“1”改为“0”（停止唤醒运行），就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许（SPIEn=1）因检测到停止条件而产生的中断，并且在检测到中断后读 IICSn 寄存器。

备注：STTn：IICA 控制寄存器 n0（IICCTLn0）的 bit1

WUPn：IICA 控制寄存器 n1（IICCTLn1）的 bit7

图 18-7：IICA 状态寄存器 n（IICSn）的格式(1/3)

复位后：00H　　R

符号	7	6	5	4	3	2	1	0
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	主控状态的确认标志
0	从属状态或者通信待机状态
1	主控通信状态

清除条件（MSTSn=0）	置位条件（MSTSn=1）
• 当检测到停止条件时 • 当ALDn位为“1”（仲裁失败）时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当生成开始条件时

ALDn	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MSTSn位。

清除条件（ALDn=0）	置位条件（ALDn=1）
• 在读IICSn寄存器后自动清除注。 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当仲裁失败时

注：即使对 IICSn 寄存器以外的位执行位存储器操作指令，也清除此位。因此，在使用 ALDn 位时，必须在读其他位前先读 ALDn 位的数据。

备注：

1. LRELn：IICA 控制寄存器 n0（IICCTLn0）的 bit6

   IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7

2. n=0

图 18-7：IICA 状态寄存器 n（IICSn）的格式(2/3)

EXCn	扩展码的接收检测
0	未接收到扩展码。
1	接收到扩展码。

清除条件（EXCn=0）	置位条件（EXCn=1）
• 当检测到开始条件时 • 当检测到停止条件时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当接收的地址数据的高4位为“0000”或者“1111”时（在第8个时钟的上升沿置位）

COIn	地址匹配的检测
0	地址不同。
1	地址相同。

清除条件（COIn=0）	置位条件（COIn=1）
• 当检测到开始条件时 • 当检测到停止条件时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当接收地址和本地站地址（从属地址寄存器n（SVAn））相同时（在第8个时钟的上升沿置位）

TRCn	发送/接收的状态检测
0	处于接收状态（发送状态除外）。将SDAAn线置为高阻抗。
1	处于发送状态。设定为能将SOn锁存器的值输出到SDAAn线（在第1字节的第9个时钟的下降沿以后有效）。

清除条件（TRCn=0）	置位条件（TRCn=1）
＜主控设备和从属设备＞ • 当检测到停止条件时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 因WRELn位为“1”（解除等待）而进行的清除注 • 当ALDn位从“0”变为“1”（仲裁失败）时 • 当复位时 • 不参加通信的情况（MSTSn、EXCn、COIn=0） ＜主控设备＞ • 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	＜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

注：在 IICA 状态寄存器 n（IICSn）的 bit3（TRCn）为“1”（发送状态）时，如果在第 9 个时钟将 IICA 控制寄存器 n0（IICCTLn0）的 bit5（WRELn）置“1”来解除等待，就在清除 TRCn 位（接收状态）后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1”（发送状态）时的等待解除。

备注：

1. LRELn：IICA 控制寄存器 n0（IICCTLn0）的 bit6
   IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7
2. n=0

图 18-7：IICA 状态寄存器 n（IICSn）的格式(3/3)

ACKDn	应答（ACK）的检测
0	未检测到应答。
1	检测到应答。

清除条件（ACKDn=0）	置位条件（ACKDn=1）
• 当检测到停止条件时 • 当下一个字节的第1个时钟上升时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 在SCLAn线的第9个时钟上升沿将SDAAn线置为低电平时

STDn	开始条件的检测
0	未检测到开始条件。
1	检测到开始条件，表示处于地址传送期间。

清除条件（STDn=0）	置位条件（STDn=1）
• 当检测到停止条件时 • 在地址传送后的下一个字节的第1个时钟上升时 • 因LRELn位为“1”（退出通信）而进行的清除 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当检测到开始条件时

SPDn	停止条件的检测
0	未检测到停止条件。
1	检测到停止条件，主控设备结束通信并且已释放总线。

清除条件（SPDn=0）	置位条件（SPDn=1）
• 在将此位置位后，在检测到开始条件后的地址传送字节的第1个时钟上升时 • 当WUPn位从“1”变为“0”时 • 当IICEn位从“1”变为“0”（停止运行）时 • 当复位时	• 当检测到停止条件时

备注：

1. LRELn：IICA 控制寄存器 n0（IICCTLn0）的 bit6

   IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7

2. n=0

18.3.4 IICA标志寄存器n（IICFn）

这是设定 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”。

图 18-8：IICA 标志寄存器 n（IICFn）的格式

复位后：00H　　R/W 注

符号	7	6	5	4	3	2	1	0
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn清除标志
0	发行开始条件。
1	无法发行开始条件而清除STTn标志。

清除条件（STCFn=0）	置位条件（STCFn=1）
• 因STTn位为“1”而进行的清除 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 在设定为禁止通信预约（IICRSVn=1）的状态下无法发行开始条件而将STTn位清“0”时

IICBSYn	I²C总线状态标志
0	总线释放状态（STCENn=1时的通信初始状态）
1	总线通信状态（STCENn=0时的通信初始状态）

清除条件（IICBSYn=0）	置位条件（IICBSYn=1）
• 当检测到停止条件时 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 当检测到开始条件时 • STCENn位为“0”时的IICEn位的置位

STCENn	初始开始允许触发
0	在允许运行（IICEn=1）后，通过检测停止条件来允许生成开始条件。
1	在允许运行（IICEn=1）后，不检测停止条件而允许生成开始条件。

清除条件（STCENn=0）	置位条件（STCENn=1）
• 通过指令清除。 • 当检测到开始条件时 • 当复位时	• 通过指令置位。

IICRSVn	通信预约功能禁止位
0	允许通信预约。
1	禁止通信预约。

清除条件（IICRSVn=0）	置位条件（IICRSVn=1）
• 通过指令清除。 • 当复位时	• 通过指令置位。

注：bit6 和 bit7 是只读位。

注意：

1. 只有在停止运行（IICEn=0）时才能写 STCENn 位。
2. 如果 STCENn 位为“1”，就与实际的总线状态无关而认为总线为释放状态（IICBSYn=0），因此为了避免在发行第 1 个开始条件（STTn=1）时破坏其他通信，需要确认没有正在通信的第三方。
3. 只有在停止运行（IICEn=0）时才能写 IICRSVn。

备注：

1. STTn：IICA 控制寄存器 n0（IICCTLn0）的 bit1
2. IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7

18.3.5 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”。

图 18-9：IICA 控制寄存器 n1（IICCTLn1）的格式(1/2)

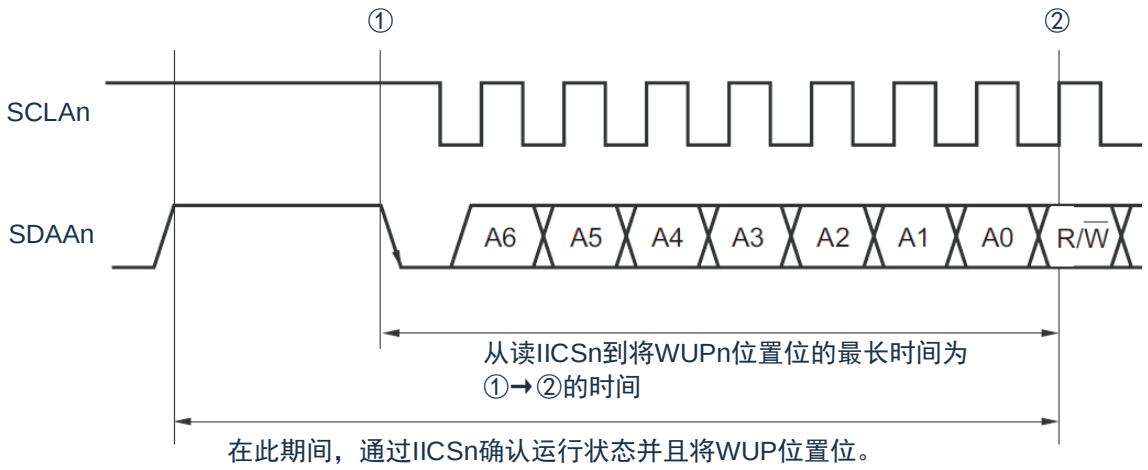
复位后：00H　　R/W 注 1

符号	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	地址匹配唤醒的控制
0	在深度睡眠模式中，停止地址匹配唤醒功能的运行。
1	在深度睡眠模式中，允许地址匹配唤醒功能的运行。
要通过将WUPn位置“1”来转移到深度睡眠模式时，必须在将WUPn位置“1”后至少经过3个F_{MCK}时钟，然后执行深度睡眠指令（参照“图14-28将WUPn位置“1”时的流程”）。在地址匹配或者接收到扩展码后，必须将WUPn位清“0”。能通过将WUPn位清“0”来参加后续的通信（需要在将WUPn位清“0”后解除等待以及写发送数据）。 在WUPn位为“1”的状态下，地址匹配或者接收到扩展码时的中断时序与WUPn位为“0”时的中断时序相同（根据时钟产生采样误差的延迟差）。另外，当WUPn位为“1”时，即使将SPIEn位置“1”也不产生停止条件中断。	
清除条件（WUPn=0）	置位条件（WUPn=1）
• 通过指令清除（在地址匹配或者接收到扩展码后）。	• 通过指令置位（MSTSn=0、EXCn=0、COIn=0并且STDn=0（不参加通信））注2。

注 1：bit4 和 bit5 是只读位。

注 2：在以下所示的期间，需要确认 IICA 状态寄存器 n（IICSn）的状态并且将其置位。



备注：n=0

图 18-9：IICA 控制寄存器 n1（IICCTLn1）的格式(2/2)

CLDn	SCLAn引脚的电平检测（只在IICEn位为“1”时有效）
0	检测到SCLAn引脚为低电平。
1	检测到SCLAn引脚为高电平。
清除条件（CLDn=0）	置位条件（CLDn=1）
• 当SCLAn引脚为低电平时 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 当SCLAn引脚为高电平时

DADn	SDAAn引脚的电平检测（只在IICEn位为“1”时有效）
0	检测到SDAAn引脚为低电平。
1	检测到SDAAn引脚为高电平。
清除条件（DADn=0）	置位条件（DADn=1）
• 当SDAAn引脚为低电平时 • 当IICEn位为“0”（停止运行）时 • 当复位时	• 当SDAAn引脚为高电平时

SMCn	运行模式的切换
0	在标准模式中运行（最大传送速率：100kbps）。
1	在快速模式（最大传送速率：400kbps）或者增强型快速模式（最大传送速率：1Mbps）中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器OFF
1	数字滤波器ON
必须在快速模式或者增强型快速模式中使用数字滤波器。数字滤波器用于消除噪声。 无论是将DFCn位置“1”还是清“0”，传送时钟都不变。	

PRSn	运行时钟（F_{MCK}）的控制
0	选择F_{CLK}（1MHz≤F_{CLK}≤20MHz）。
1	选择F_{CLK}/2（20MHz＜F_{CLK}）

注意：

1. IICA 运行时钟（F_{MCK}）的最大工作频率为 20MHz(Max.)。只有在 F_{CLK} 超过 20MHz 时才必须将 IICA 控制寄存器 n1（IICCTLn1）的 bit0（PRSn）置“1”。
2. 在设定传送时钟的情况下，必须注意 fCLK 的最小工作频率。串行接口 IICA 的 F_{CLK} 最小工作频率取决于运行模式。

   快速模式：F_{CLK}=3.5MHz(Min.)

   增强型快速模式：F_{CLK}=10MHz(Min.)

   标准模式：F_{CLK}=1MHz(Min.)

备注：

1. IICEn：IICA 控制寄存器 n0（IICCTLn0）的 bit7
2. n=0

18.3.6 IICA低电平宽度设定寄存器n（IICWLn）

此寄存器控制串行接口 IICA 输出的 SCLAn 引脚信号低电平宽度（T_{LOW}）和 SDAAn 引脚信号。

通过 8 位存储器操作指令设定 IICWLn 寄存器。

必须在禁止 I^2C 运行（IICA 控制寄存器 n0（IICCTLn0）的 bit7（IICEn）=0）时设定 IICWLn 寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

有关 IICWLn 寄存器的设定方法，请参照“18.4.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法”。

数据保持时间为 IICWLn 所设时间的 1/4。

图 18-10：IICA 低电平宽度设定寄存器 n（IICWLn）的格式

复位后：FFH　　R/W

符号	7	6	5	4	3	2	1	0
IICWLn								

18.3.7 IICA高电平宽度设定寄存器n（IICWHn）

此寄存器控制串行接口 IICA 输出的 SCLAn 引脚信号高电平宽度和 SDAAn 引脚信号。通过 8 位存储器操作指令设定 IICWHn 寄存器。

必须在禁止 I^2C 运行（IICA 控制寄存器 n0（IICCTLn0）的 bit7（IICEn）=0）时设定 IICWHn 寄存器。在产生复位信号后，此寄存器的值变为“FFH”。

图 18-11：IICA 高电平宽度设定寄存器 n（IICWHn）的格式

复位后：FFH　　R/W

符号	7	6	5	4	3	2	1	0
IICWHn								

备注：

1. 有关主控方传送时钟的设定方法，请参照 18.4.2(1)；有关从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法，请参照 18.4.2(2)。
2. n=0

18.3.8 控制IICA引脚端口功能的寄存器

本产品可将 IICAn 的引脚功能复用到多个端口。

通过设置端口复用功能配置寄存器（SCLAnPCFG 和 SDAAnPCFG）可将 SCALn 引脚和 SDAAn 引脚分别配置到端口。（n=0）

将这两个端口对应的端口模式控制寄存器（PMCxx）的位和端口模式寄存器（PMxx）的位置“0”。

这两个端口被配置为 IICA 引脚的复用功能后，端口的 N 沟道漏极开路输出（VDD/EVDD 耐压）模式由设计保证自动打开，即 POMxx 寄存器不需要用户设置。

详细的设置方法参见“第 2 章 端口功能”。

18.4 I2C总线模式的功能

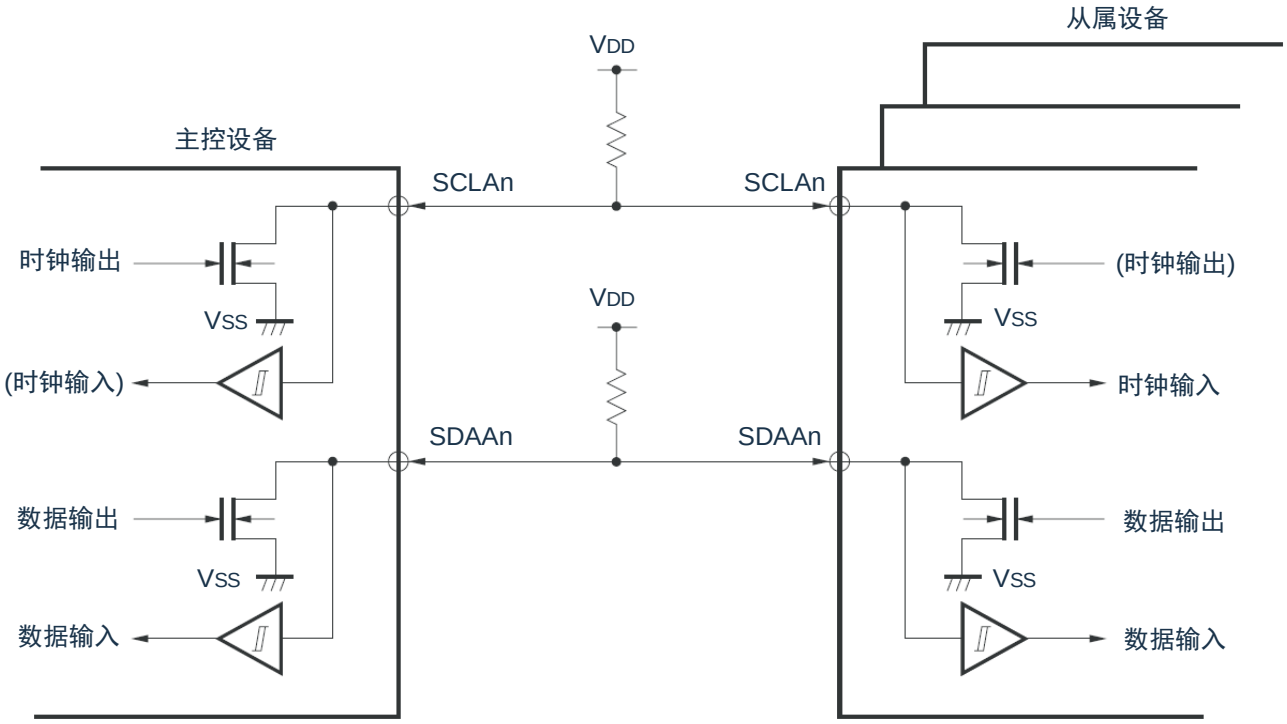
18.4.1 引脚结构

串行时钟引脚（SCLAn）和串行数据总线引脚（SDAAn）的结构如下。

1) SCLAn……串行时钟的输入/输出引脚
   主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。
2) SDAAn……串行数据的输入/输出复用引脚
   主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

因为串行时钟线和串行数据总线的输出为 N 沟道漏极开路输出，所以需要外接上拉电阻。

图 18-12：引脚结构图



备注：n=0

18.4.2 通过IICWLn寄存器和IICWHn寄存器设定传送时钟的方法

(1) 主控方传送时钟的设定方法

$$传送时钟 = \frac{F_{MCK}}{IICWL + IICWH + F_{MCK}(T_R+T_F)}$$

此时，IICWLn 寄存器和 IICWHn 寄存器的最佳设定值如下：
（全部设定值的小数部分都舍入）

- 快速模式

  $IICWLn=\frac{0.52}{传送时钟} \times F_{MCK}$

  $IICWHn=(\frac{0.48}{传送时钟} \times T_R-T_F) \times F_{MCK}$

- 标准模式

  $IICWLn=\frac{0.47}{传送时钟} \times F_{MCK}$

  $IICWHn=(\frac{0.53}{传送时钟} \times T_R-T_F) \times F_{MCK}$

- 增强型快速模式

  $IICWLn=\frac{0.50}{传送时钟} \times F_{MCK}$

  $IICWHn=(\frac{0.50}{传送时钟} \times T_R-T_F) \times F_{MCK}$

(2) 从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法
（全部设定值的小数部分都舍入）

- 快速模式

  $IICWLn=1.3us \times F_{MCK}$

  $IICWHn=(1.2us-T_R-T_F) \times F_{MCK}$

- 标准模式

  $IICWLn=4.7us \times F_{MCK}$

  $IICWHn=(5.3us-T_R-T_F) \times F_{MCK}$

- 增强型快速模式

  $IICWLn=0.50us \times F_{MCK}$

  $IICWHn=(0.50us-T_R-T_F) \times F_{MCK}$

注意：

1. IICA 运行时钟（F_{MCK}）的最大工作频率为 20MHz(Max.)。只有在 F_{CLK} 超过 20MHz 时才必须将 IICA 控制寄存器 n1（IICCTLn1）的 bit0（PRSn）置“1”。
2. 在设定传送时钟的情况下，必须注意 F_{CLK} 的最小工作频率。串行接口 IICA 的 F_{CLK} 最小工作频率取决于运行模式。

   快速模式：F_{CLK}=3.5MHz(Min.)

   增强型快速模式：F_{CLK}=10MHz(Min.)

   标准模式：F_{CLK}=1MHz(Min.)

备注：

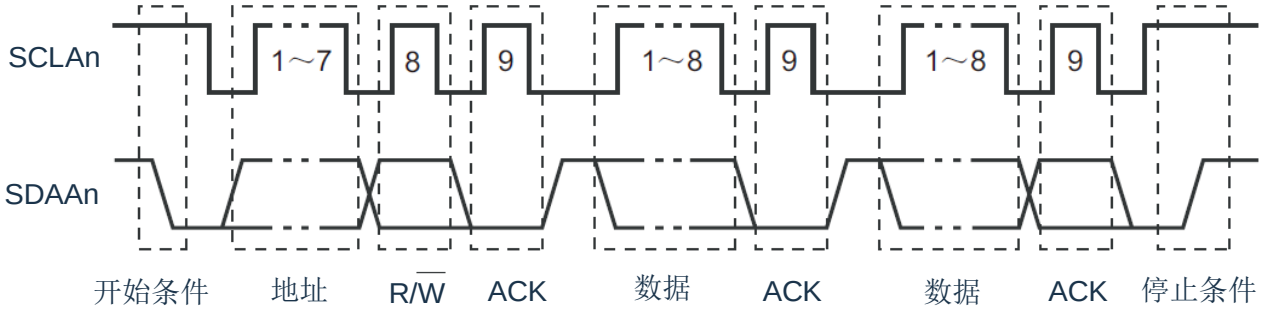
1. 因为 SDAAn 信号和 SCLAn 信号的上升时间（T_R）和下降时间（T_F）因上拉电阻和布线电容而不同，所以必须各自计算。
2. IICWLn：IICA 低电平宽度设定寄存器 n
   IICWHn：IICA 高电平宽度设定寄存器 n
   T_F：SDAAn 信号和 SCLAn 信号的下降时间
   T_R：SDAAn 信号和 SCLAn 信号的上升时间
   F_{MCK}：IICA 运行时钟频率
3. n=0

18.5 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 18-13：I²C 总线的串行数据传送时序



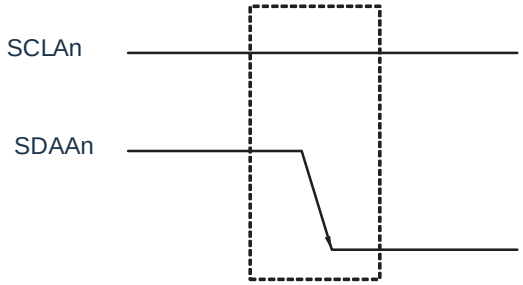
主控设备生成开始条件、从属地址和停止条件。

主控设备和从属设备都能生成应答（ACK）（在一般情况下，接收方输出 8 位数据）。主控设备连续输出串行时钟（SCLAn）。但是，从属设备能延长 SCLAn 引脚的低电平期间并且插入等待。

18.5.1 开始条件

在 SCLAn 引脚为高电平时，如果 SDAAn 引脚从高电平变为低电平，就生成开始条件。SCLAn 引脚和 SDAAn 引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。在用作从属设备时，能检测到开始条件。

图 18-14：开始条件



在检测到停止条件（SPDn：IICA 状态寄存器 n（IICSn）的 bit0=1）的状态下，如果将 IICA 控制寄存器 n0（IICCTLn0）的 bit1（STTn）置“1”，输出开始条件。如果检测到开始条件，就将 IICSn 寄存器的 bit1（STDn）置“1”。

备注：n=0

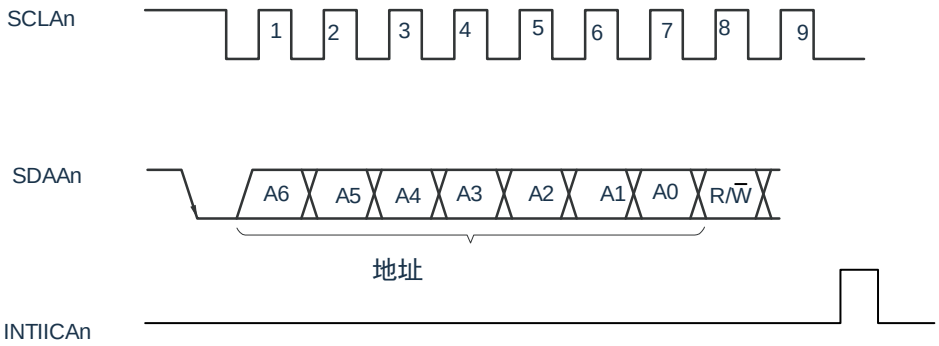
18.5.2 地址

开始条件的后续 7 位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的 7 位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查 7 位数据是否和从属地址寄存器 n（SVAn）的内容相同。此时，如果 7 位数据和 SVAn 寄存器的值相同，该从属设备就被选中，在主控设备生成开始条件或者停止条件前，与主控设备进行通信。

图 18-15：地址



注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

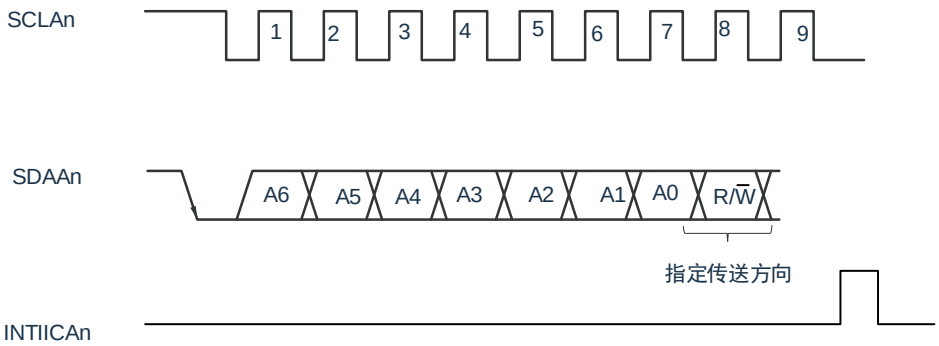
如果将从属地址和“18.5.3 传送方向的指定”中说明的传送方向构成的 8 位数据写到 IICA 移位寄存器 n（IICAn），就输出地址。接收到的地址被写到 IICAn 寄存器。从属地址分配在 IICAn 寄存器的高 7 位。

18.5.3 传送方向的指定

主控设备在 7 位地址之后发送 1 位指定传送方向的数据。

当此传送方向指定位为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定位为“1”时，表示主控设备从从属设备接收数据。

图 18-16：传送方向的指定



注：如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

备注：n=0

18.5.4 应答（ACK）

能通过应答（ACK）确认发送方和接收方的串行数据状态。接收方在每次接收到 8 位数据时返回应答。

通常，发送方在发送 8 位数据后接收应答。当接收方返回应答时，认为已正常接收，继续处理。能通过 IICA 状态寄存器 n（IICSn）的 bit2（ACKDn）确认应答的检测。在主控设备为接收状态下接收到最后的数据时，不返回应答而生成停止条件。在从属设备接收数据后不返回应答时，主控设备输出停止条件或者重新开始条件，中止发送。不返回应答的原因如下：

① 没有正常接收。

② 已结束最后数据的接收。

③ 不存在地址指定的接收方。

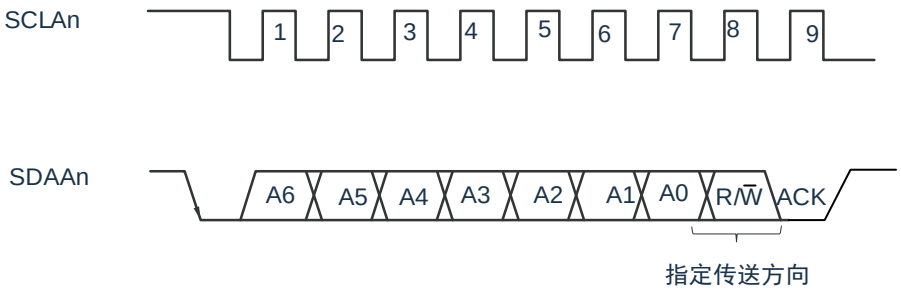
接收方在第 9 个时钟将 SDAAn 线置为低电平，生成应答（正常接收）。

通过将 IICA 控制寄存器 n0（IICCTLn0）的 bit2（ACKEn）置“1”，变为能自动生成应答的状态。通过 7 位地址信息后续的第 8 位数据设定 IICSn 寄存器的 bit3（TRCn）。在接收（TRCn=0）的情况下，通常必须将 ACKEn 位置“1”。

在从属接收运行过程中（TRCn=0）不能接收数据或者不需要下一个数据时，必须将 ACKEn 位清“0”，通知主控方不能接收数据。

在主控接收运行过程中（TRCn=0）不需要下一个数据时，为了不生成应答，必须将 ACKEn 位清“0”，通知从属发送方数据的结束（停止发送）。

图 18-17：应答



当接收到本地站的地址时，与 ACKEn 位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答（NACK）。

在接收到扩展码时，通过事先将 ACKEn 位置“1”，生成应答。接收数据时的应答生成方法因等待时序的设定而不同，如下所示。

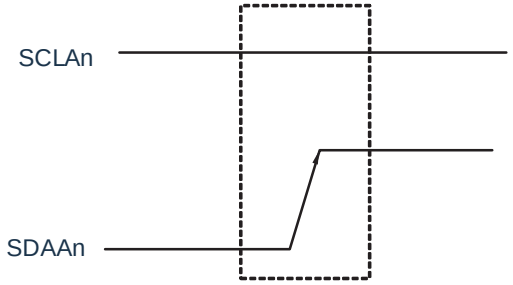
- 当选择 8 个时钟的等待时（IICCTLn0 寄存器的 bit3（WTIMn）=0）：通过在解除等待前将 ACKEn 位置“1”，与 SCLAn 引脚的第 8 个时钟下降沿同步生成应答。
- 当选择 9 个时钟的等待时（IICCTLn0 寄存器的 bit3（WTIMn）=1）：通过事先将 ACKEn 位置“1”，生成应答。

备注：n=0

18.5.5 停止条件

在 SCLAn 引脚为高电平时，如果 SDAAn 引脚从低电平变为高电平，就生成停止条件。停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图 18-18：停止条件



如果将 IICA 控制寄存器 n0（IICCTLn0）的 bit0（SPTn）置“1”，就生成停止条件。如果检测到停止条件，就将 IICA 状态寄存器 n（IICSn）的 bit0（SPDn）置“1”，并且在 IICCTLn0 寄存器的 bit4（SPIEn）为“1”时产生 INTIICAn。

备注：n=0

18.5.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送/接收（等待状态）。

通过将 SCLAn 引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图 18-19：等待(1/2)

(1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况

（主控设备：发送，从属设备：接收，ACKEn=1）

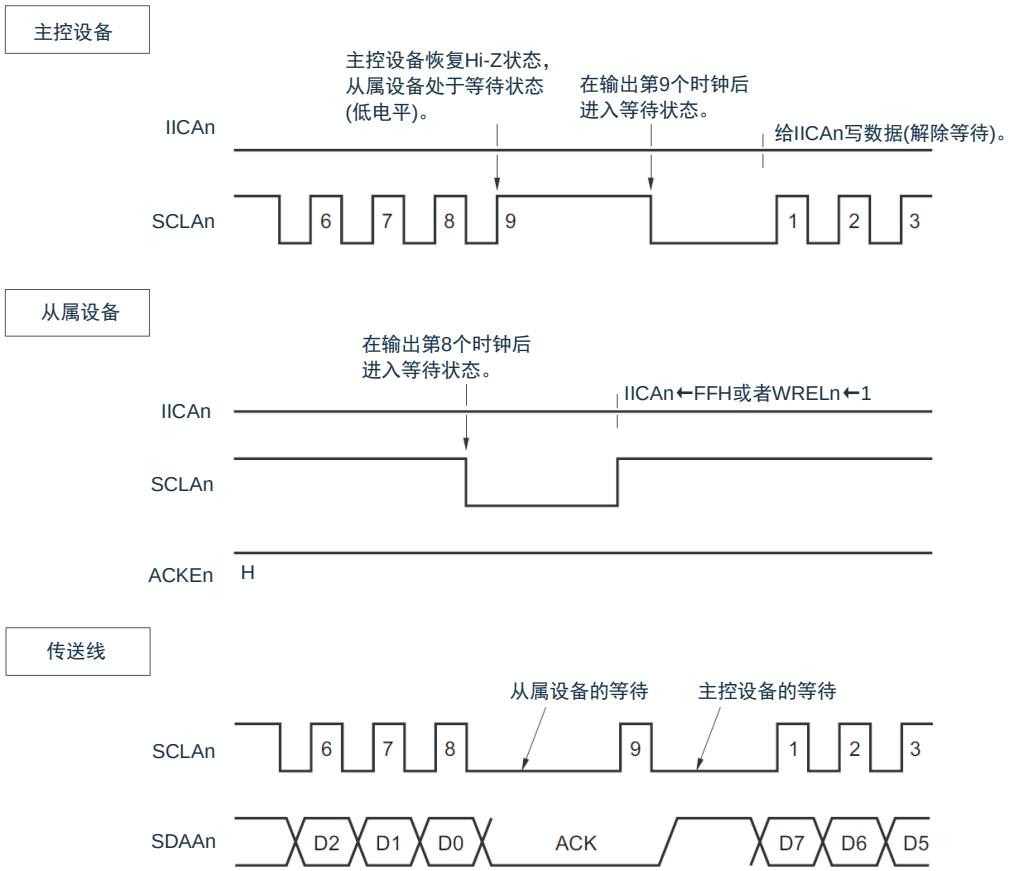
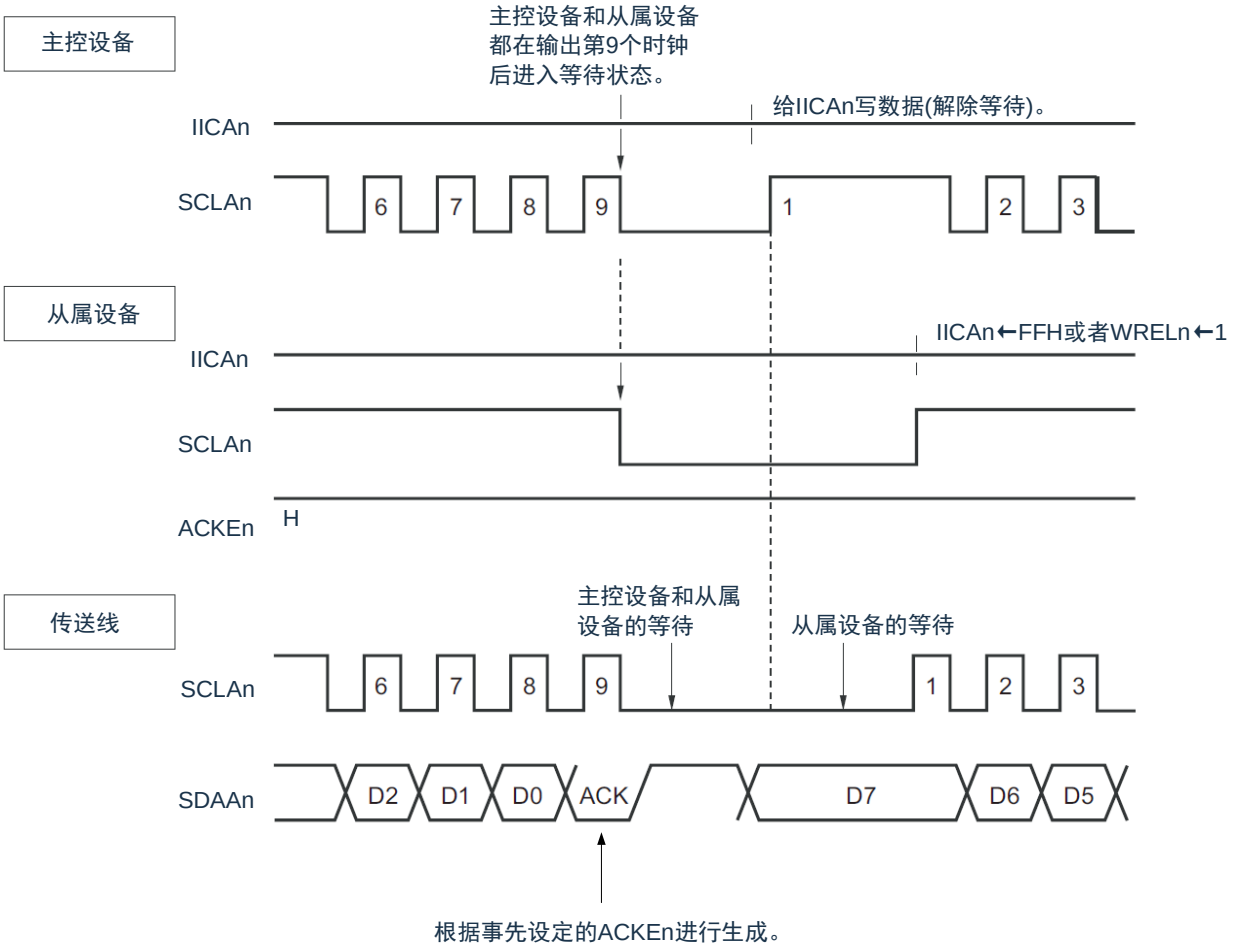


图 18-19：等待(2/2)

(2) 主控设备和从属设备都为 9 个时钟等待的情况

（主控设备：发送，从属设备：接收，ACKEn=1）



备注：ACKEn：IICA 控制寄存器 n0（IICCTLn0）的 bit2

WRELn：IICA 控制寄存器 n0（IICCTLn0）的 bit5

通过设定 IICA 控制寄存器 n0（IICCTLn0）的 bit3（WTIMn）自动产生等待状态。通常，在接收方，如果 IICCTLn0 寄存器的 bit5（WRELn）为“1”或者给 IICA 移位寄存器 n（IICAn）写“FFH”，就解除等待；在发送方，如果给 IICAn 寄存器写数据，就解除等待。主控设备还能通过以下方法解除等待：

- 将IICCTLn0寄存器的bit1（STTn）置“1”。
- 将IICCTLn0寄存器的bit0（SPTn）置“1”。

备注：n=0

18.5.7 等待的解除方法

在一般情况下，I²C 能通过以下的处理来解除等待。

- 给 IICA 移位寄存器 n（IICAn）写数据。
- 将 IICA 控制寄存器 n0（IICCTLn0）的 bit5（WRELn）置位（解除等待）。
- 将 IICCTLn0 寄存器的 bit1（STTn）置位（生成开始条件）[注]。
- 将 IICCTLn0 寄存器的 bit0（SPTn）置位（生成停止条件）[注]。

注：只限于主控设备。

如果执行了这些等待的解除处理，I²C 就解除等待，重新开始通信。要在解除等待后发送数据（包括地址）时，必须给 IICAn 寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将 IICCTLn0 寄存器的 bit5（WRELn）置“1”。要在解除等待后生成重新开始条件时，必须将 IICCTLn0 寄存器的 bit1（STTn）置“1”。要在解除等待后生成停止条件时，必须将 IICCTLn0 寄存器的 bit0（SPTn）置“1”。对于一次等待只能执行一次解除处理。

例如，如果在通过将 WRELn 位置“1”来解除等待后给 IICAn 寄存器写数据，SDAAn 线的变化时序与 IICAn 寄存器的写时序就可能发生冲突，导致将错误的值输出到 SDAAn 线。除了这些处理以外，在中途中止通信的情况下，如果将 IICEn 位清“0”，就停止通信，因此能解除等待。在 I²C 总线状态因噪声而被死锁的情况下，如果将 IICCTLn0 寄存器的 bit6（LRELn）置“1”，就退出通信，因此能解除等待。

注意：如果在 WUPn 位为“1”时执行等待的解除处理，就不解除等待。

备注：n=0

18.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设定 IICA 控制寄存器 n0（IICCTLn0）的 bit3（WTIMn），在表 18-2 所示的时序产生 INTIICAn 并且进行等待控制。

表 18-2：INTIICAn 的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9[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。

备注：表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属运行：与 WTIMn 位无关，根据上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控运行：与 WTIMn 位无关，在第 9 个时钟的下降沿产生中断和等待的时序。

(2) 数据接收

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

(3) 数据发送

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

备注：n=0

(4) 等待的解除方法

等待的解除方法有以下 4 种：

- 给 IICA 移位寄存器 n（IICAn）写数据。
- 将 IICA 控制寄存器 n0（IICCTLn0）的 bit5（WRELn）置位（解除等待）。
- 将 IICCTLn0 寄存器的 bit1（STTn）置位（生成开始条件）[注]。
- 将 IICCTLn0 寄存器的 bit0（SPTn）置位（生成停止条件）[注]。

注：只限于主控设备。

当选择 8 个时钟的等待（WTIMn=0）时，需要在解除等待前决定是否生成应答。

(5) 停止条件的检测

如果检测到停止条件，就产生 INTIICAn（只限于 SPIEn=1 的情况）。

18.5.9 地址匹配的检测方法

在 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。

18.5.10 错误的检测

在 I²C 总线模式中，因为发送过程中的串行数据总线（SDAAn）的状态被取到发送器件的 IICA 移位寄存器 n（IICAn），所以能通过将开始发送前和发送结束后的 IICA 数据进行比较来检测发送错误。此时，如果 2 个数据不同，就判断为发生了发送错误。

备注：n=0

18.5.11 扩展码

(1) 当接收地址的高4位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志（EXCn）置“1”，并且在第8个时钟的下降沿产生中断请求（INTIICAn）。
不影响保存在从属地址寄存器n（SVAn）的本地站地址。

(2) 当SVAn寄存器的设定值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求（INTIICAn）。
- 高4位数据相同：EXCn=1
- 7位数据相同：COIn=1

备注：EXCn：IICA 状态寄存器 n（IICSn）的 bit5
COIn：IICA 状态寄存器 n（IICSn）的 bit4

(3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将IICA控制寄存器n0（IICCTLn0）的bit6（LRELn）置“1”，进入下一次通信的待机状态。

表 18-3：主要扩展码的位定义

从属地址	R/W位	说明
0000000	0	全呼地址
11110xx	0	10位从属地址的指定（地址认证时）
11110xx	1	10位从属地址的指定（在地址相同后发行读命令时）

备注：
1. 有关上述以外的扩展码，请参照 NXP 公司发行的 I^2C 总线规格书。
2. n=0

18.5.12 仲裁

当多个主控设备同时生成开始条件时（在 STDn 位变为“1”前将 STTn 位置“1”的情况），边调整时钟边进行主控设备的通信，直到数据不同为止。此运行称为仲裁。

在仲裁失败时，仲裁失败的主控设备将 IICA 状态寄存器 n（IICSn）的仲裁失败标志（ALDn）置“1”，并且将 SCLAn 线和 SDAAn 线都置为高阻抗状态，释放总线。

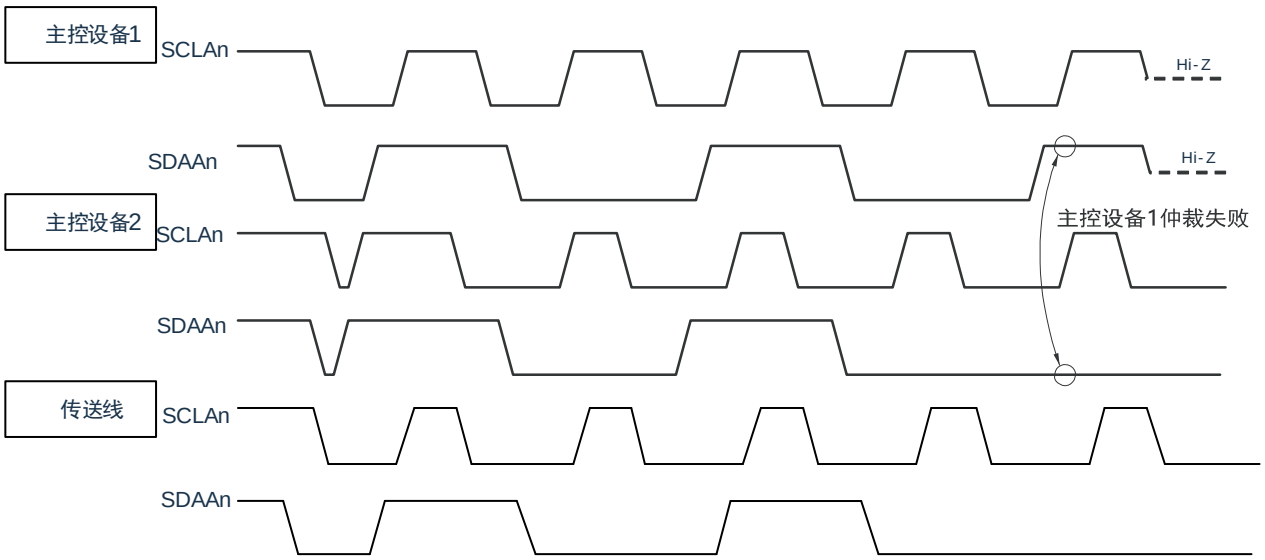
在发生下一次中断请求时（例如：在第 8 或者第 9 个时钟检测到停止条件），用软件通过 ALDn 位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“14.5.8 中断请求（INTIICAn）的产生时序和等待控制”。

备注：STDn：IICA 状态寄存器 n（IICSn）的 bit1

STTn：IICA 控制寄存器 n0（IICCTLn0）的 bit1

图 18-20：仲裁时序例子



备注：n=0

表 18-4：发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第8或者第9个时钟的下降沿[注1]
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时（SPIEn=1）[注2]
想要生成重新开始条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿[注1]
想要生成重新开始条件，但是检测到停止条件。	在生成停止条件时（SPIEn=1）[注2]
想要生成停止条件，但是数据为低电平。	在字节传送后的第8或者第9个时钟的下降沿[注1]
想要生成重新开始条件，但是SCLAn为低电平。	

注 1：当 WTIMn 位（IICA 控制寄存器 n0（IICCTLn0）的 bit3）为“1”时，在第 9 个时钟的下降沿产生中断请求；当 WTIMn 位为“0”并且接收到扩展码的从属地址时，在第 8 个时钟的下降沿产生中断请求。

注 2：当有可能发生仲裁时，必须在主控运行时将 SPIEn 位置“1”。

备注：

1. SPIEn：IICA 控制寄存器 n0（IICCTLn0）的 bit4
2. n=0

18.5.13 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 18-21 所示，通过地址匹配将 WUPn 位置“0”时的流程如图 18-22 所示。

图 18-21：将 WUPn 位置“1”时的流程

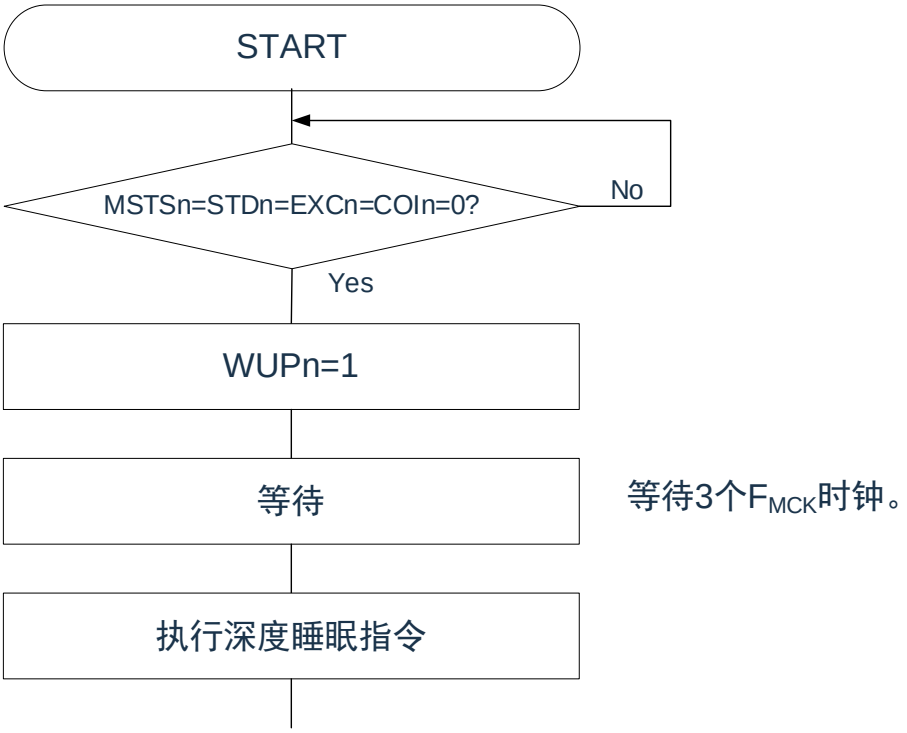
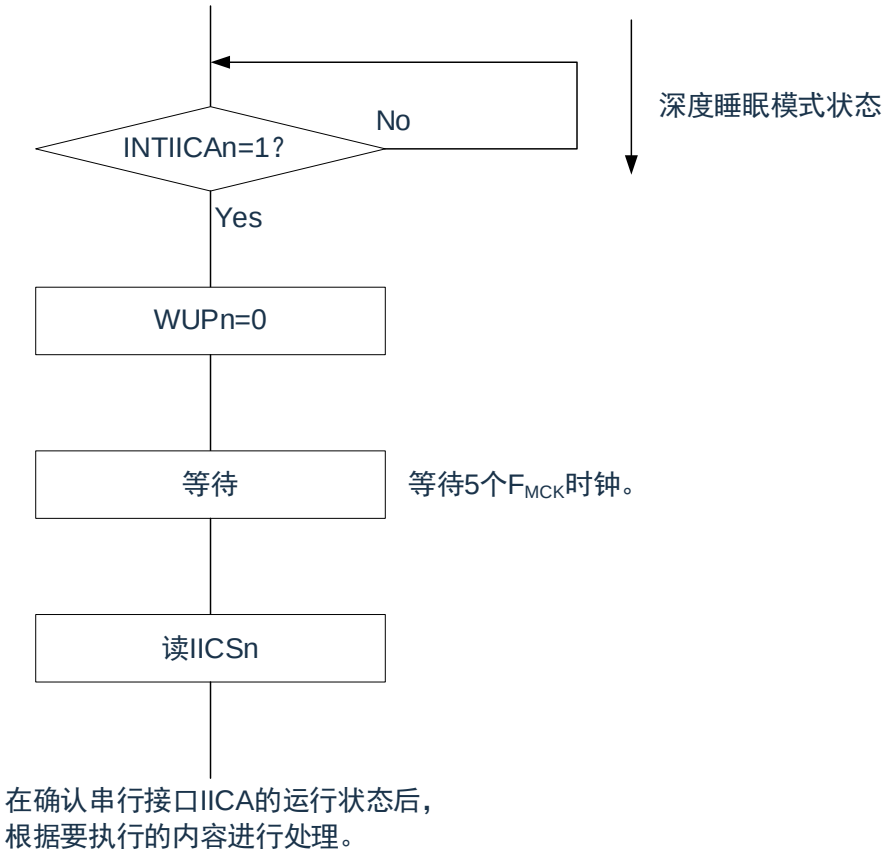


图 18-22：通过地址匹配将 WUPn 位置“0”时的流程（包括接收扩展码）



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。

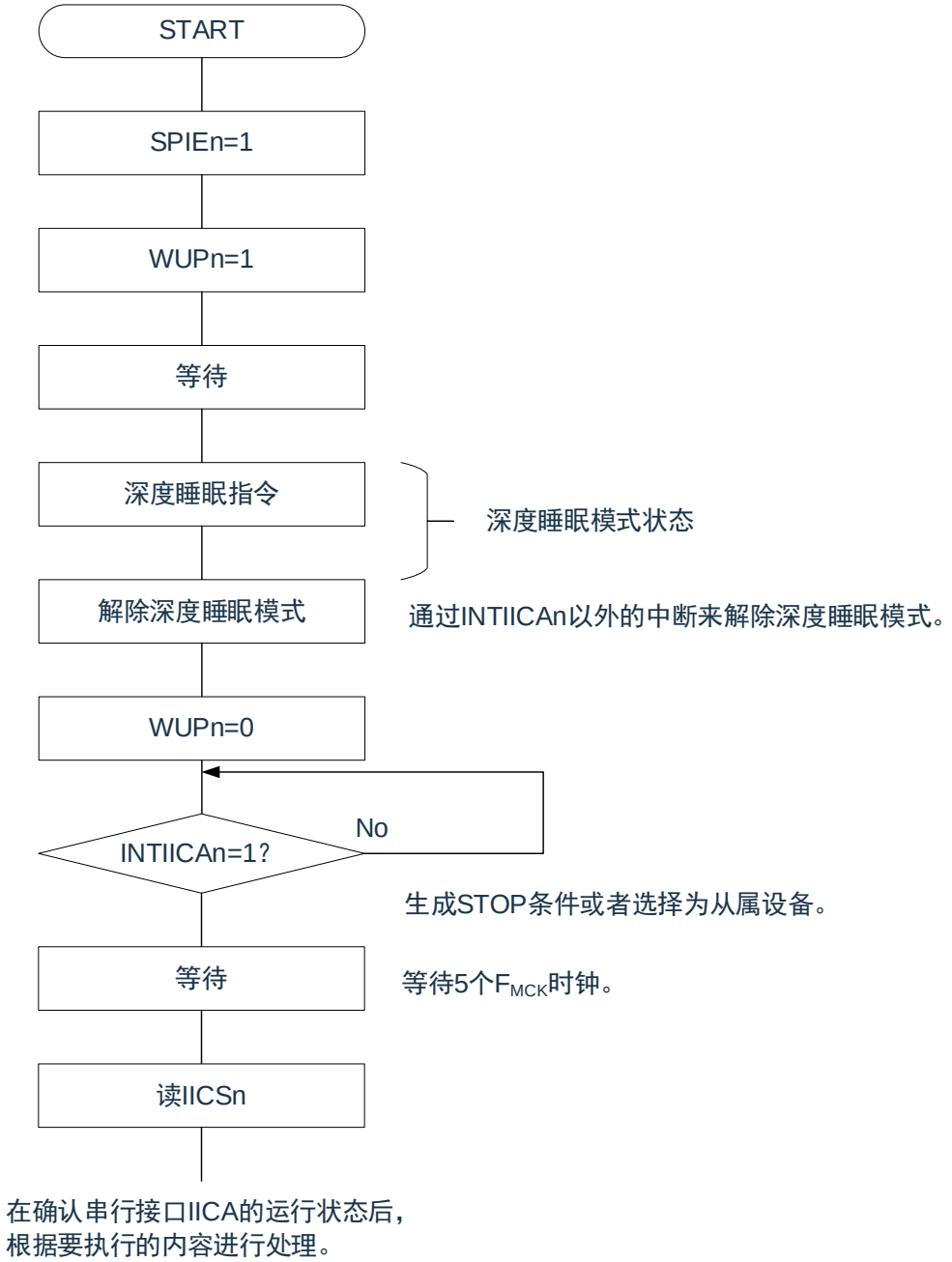
- 下一次 IIC 通信为主控设备运行的情况：图 18-23 的流程
- 下一次 IIC 通信为从属设备运行的情况：

  通过 INTIICAn 中断返回的情况：和图 18-22 的流程相同。

  通过 INTIICAn 中断以外的中断返回的情况：必须在产生 INTIICAn 中断前保持 WUPn 位为“1”的状态继续运行。

备注：n=0

图 18-23：在通过 INTIICAn 以外的中断来解除深度睡眠模式后作为主控设备运行的情况



备注：n=0

18.5.14 通信预约

(1) 允许通信预约功能的情况（IICA 标志寄存器 n（IICFn）的 bit0（IICRSVn）=0）

要在不加入总线的状态下进行下一次主控通信时，能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下 2 种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICA 控制寄存器 n0（IICCTLn0）的 bit6（LRELn）置“1”，退出通信后释放了总线）

如果在不加入总线的状态下将 IICCTLn0 寄存器的 bit1（STTn）置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将 IICCTLn0 寄存器的 bit4（SPIEn）置“1”，在通过产生的中断请求信号（INTIICAn）检测到总线的释放（检测到停止条件）后，如果给 IICA 移位寄存器 n（IICAn）写地址，就自动作为主控设备开始通信。在检测到停止条件前，给 IICAn 寄存器写的数据无效。

当将 STTn 位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时……生成开始条件
- 总线未处于释放状态（待机状态）时……通信预约

在将 STTn 位置“1”并且经过等待时间后，通过 MSTSn 位（IICA 状态寄存器 n（IICSn）的 bit7）确认是否作为通信预约运行。

必须通过软件确保以下计算式计算的等待时间：

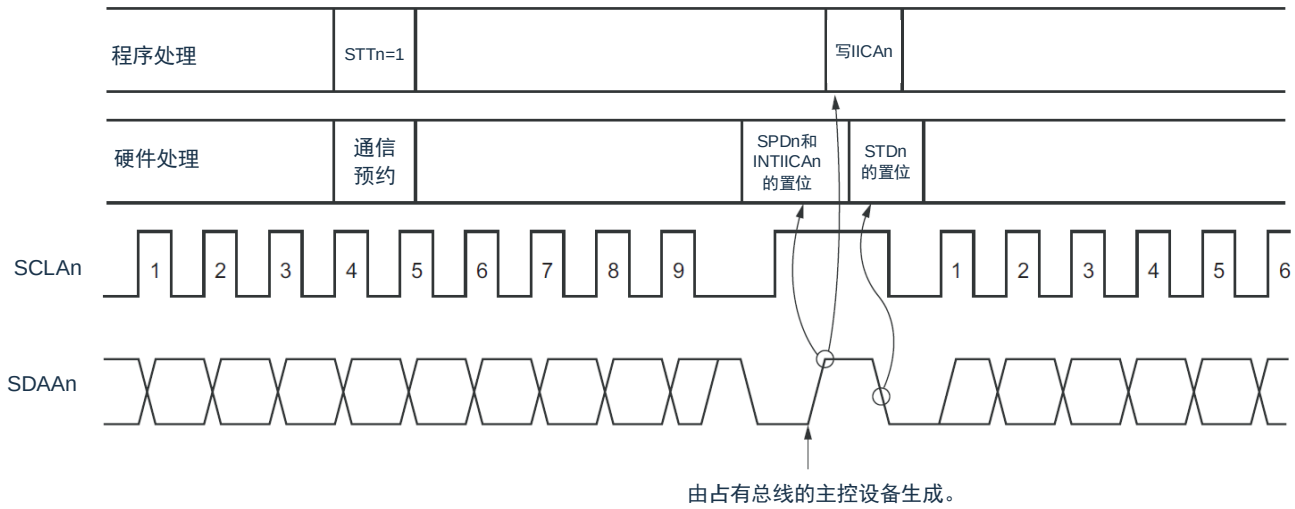
从将STTn 位置“1” 到确认MSTSn 标志为止的等待时间： (IICWLn 的设定值+ IICWHn 的设定值+ 4) / F_{MCK} + T_F × 2

备注：

1. IICWLn：IICA 低电平宽度设定寄存器 n
   IICWHn：IICA 高电平宽度设定寄存器 n
   T_F：SDAAn 信号和 SCLAn 信号的下降时间
   F_{MCK}：IICA 运行时钟频率
2. n=0

通信预约的时序如下图所示。

图 18-24：通信预约的时序



备注：IICAn：IICA 移位寄存器 n

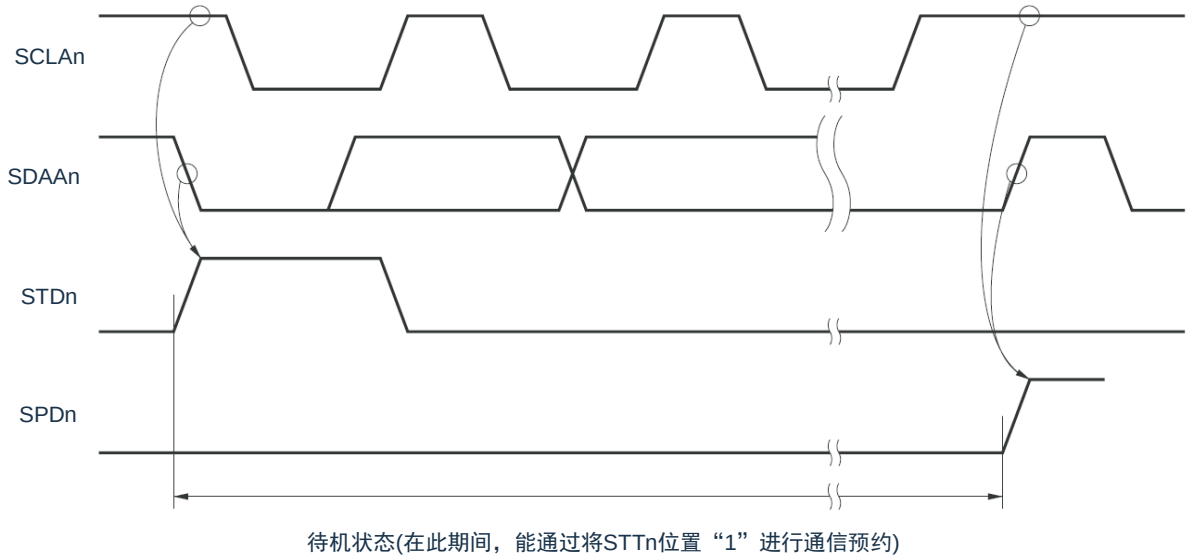
STTn：IICA 控制寄存器 n0（IICCTLn0）的 bit1

STDn：IICA 状态寄存器 n（IICSn）的 bit1

SPDn：IICA 状态寄存器 n（IICSn）的 bit0

通过图 18-25 所示的时序接受通信预约。在 IICA 状态寄存器 n（IICSn）的 bit1（STDn）变为“1”后并且在检测到停止条件前，将 IICA 控制寄存器 n0（IICCTLn0）的 bit1（STTn）置“1”进行通信预约。

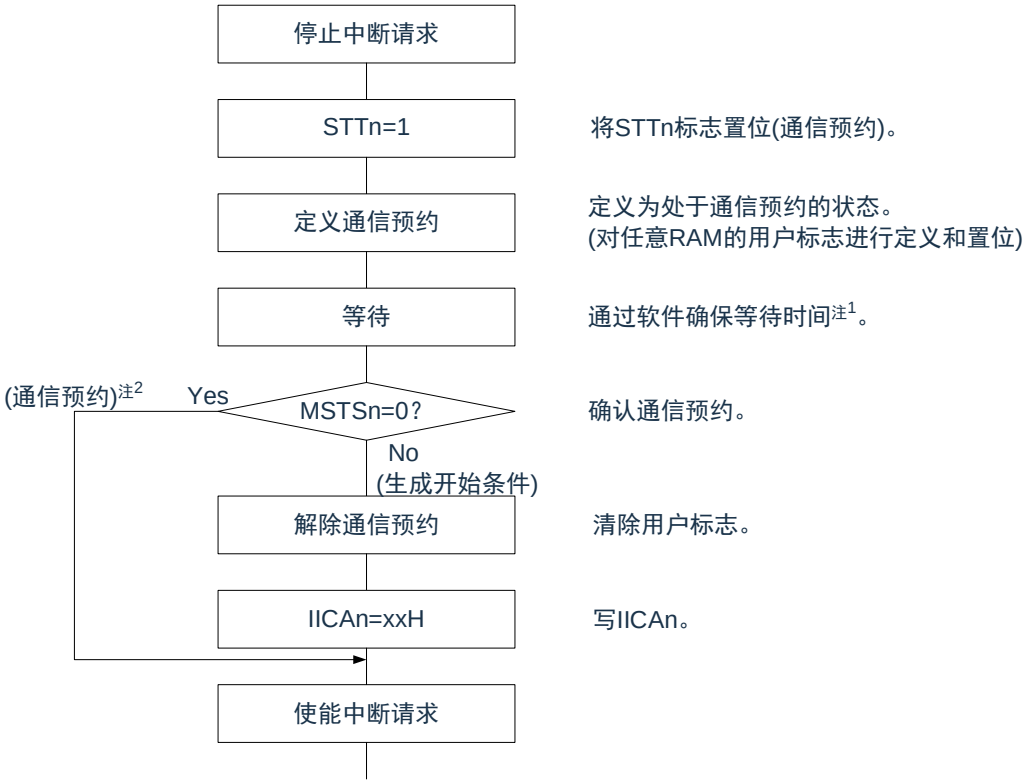
图 18-25：通信预约的接受时序



备注：n=0

通信预约的步骤如**错误!书签自引用无效。**所示。

图 18-26：通信预约的步骤



注 1：等待时间如下：(IICWLn 的设定值+IICWHn 的设定值+4)/F_{MCK}+T_F×2

注 2：在通信预约运行时，通过停止条件中断请求来写 IICA 移位寄存器 n（IICAn）。

备注：

1. STTn：IICA 控制寄存器 n0（IICCTLn0）的 bit1
   MSTSn：IICA 状态寄存器 n（IICSn）的 bit7
   IICAn：IICA 移位寄存器 n
   IICWLn：IICA 低电平宽度设定寄存器 n
   IICWHn：IICA 高电平宽度设定寄存器 n
   T_F：SDAAn 信号和 SCLAn 信号的下降时间
   F_{MCK}：IICA 运行时钟频率
2. n=0

(2) 禁止通信预约功能的情况（IICA 标志寄存器 n（IICFn）的 bit0（IICRSVn）=1）

在总线通信过程中，如果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：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICCTLn0 寄存器的 bit6（LRELn）置“1”，退出通信后释放了总线）

能通过 STCFn（IICFn 寄存器的 bit7）来确认是生成了开始条件还是拒绝了请求。因为从 STTn 位为“1”到将 STCFn 位置“1”为止需要 5 个 f_{MCK} 时钟的时间，所以必须通过软件确保此时间。

备注：n=0

18.5.15 其他注意事项

(1) STCENn位为“0”的情况

在刚允许I²C运行（IICEn=1）后，与实际的总线状态无关而视为通信状态（IICBSYn=1）。要在没有检测到停止条件的状态下进行主控通信时，必须先生成停止条件，在释放总线后进行主控通信。对于多主控，在总线未释放（未检测到停止条件）的状态下不能进行主控通信。按照以下顺序生成停止条件：

① 设定IICA控制寄存器n1（IICCTLn1）。

② 将IICA控制寄存器n0（IICCTLn0）的bit7（IICEn）置“1”。

③ 将IICCTLn0寄存器的bit0（SPTn）置“1”。

(2) STCENn位为“1”的情况

在刚允许I²C运行（IICEn=1）后，与实际的总线状态无关而视为释放状态（IICBSYn=0）。因此在生成第1个开始条件（STTn=1）时，为了不破坏其他的通信，需要确认总线已被释放。

(3) 正在和其他设备进行I²C通信的情况

在SDAAn引脚为低电平并且SCLAn引脚为高电平时，如果允许I²C运行而且中途参加通信，I²C的宏就视为SDAAn引脚从高电平变为低电平（检测到开始条件）。如果此时总线上的值是能识别为扩展码的值，就返回应答而妨碍和其他设备的I²C通信。为了避免这种情况，必须按照以下顺序启动I²C：

① 将IICCTLn0寄存器的bit4（SPIEn）清“0”，禁止在检测到停止条件时产生中断请求信号（INTIICAn）

② 将IICCTLn0寄存器的bit7（IICEn）置“1”，允许I²C运行。

③ 等待检测到开始条件。

④ 在返回应答前（将IICEn位置“1”后的4~72个F_{MCK}时钟内）将IICCTLn0寄存器的bit6（LRELn）置“1”，强制将检测置为无效。

(4) 在将STTn位和SPTn位（IICCTLn0寄存器的bit1和bit0）置位后，禁止清“0”前的再置位。

(5) 如果进行了通信预约，就必须将SPIEn位（IICCTLn0寄存器的bit4）置“1”，在检测到停止条件时产生中断请求。在产生中断请求后，通过给IICA移位寄存器n（IICAn）写通信数据，开始传送。如果在检测到停止条件时不发生中断，就因在开始通信时不产生中断请求而停止在等待状态。但是，当通过软件检测MSTSn位（IICA状态寄存器n（IICSn）的bit7）时，不需要将SPIEn位置“1”。

备注：n=0

18.5.16 通信运行

在此通过流程图表示以下 3 个运行步骤。

(1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在I^2C总线的多主控系统中，只根据I^2C总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1帧）为高电平，就将总线作为释放状态而参加通信。此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

用作I^2C总线从属设备的例子如下所示。

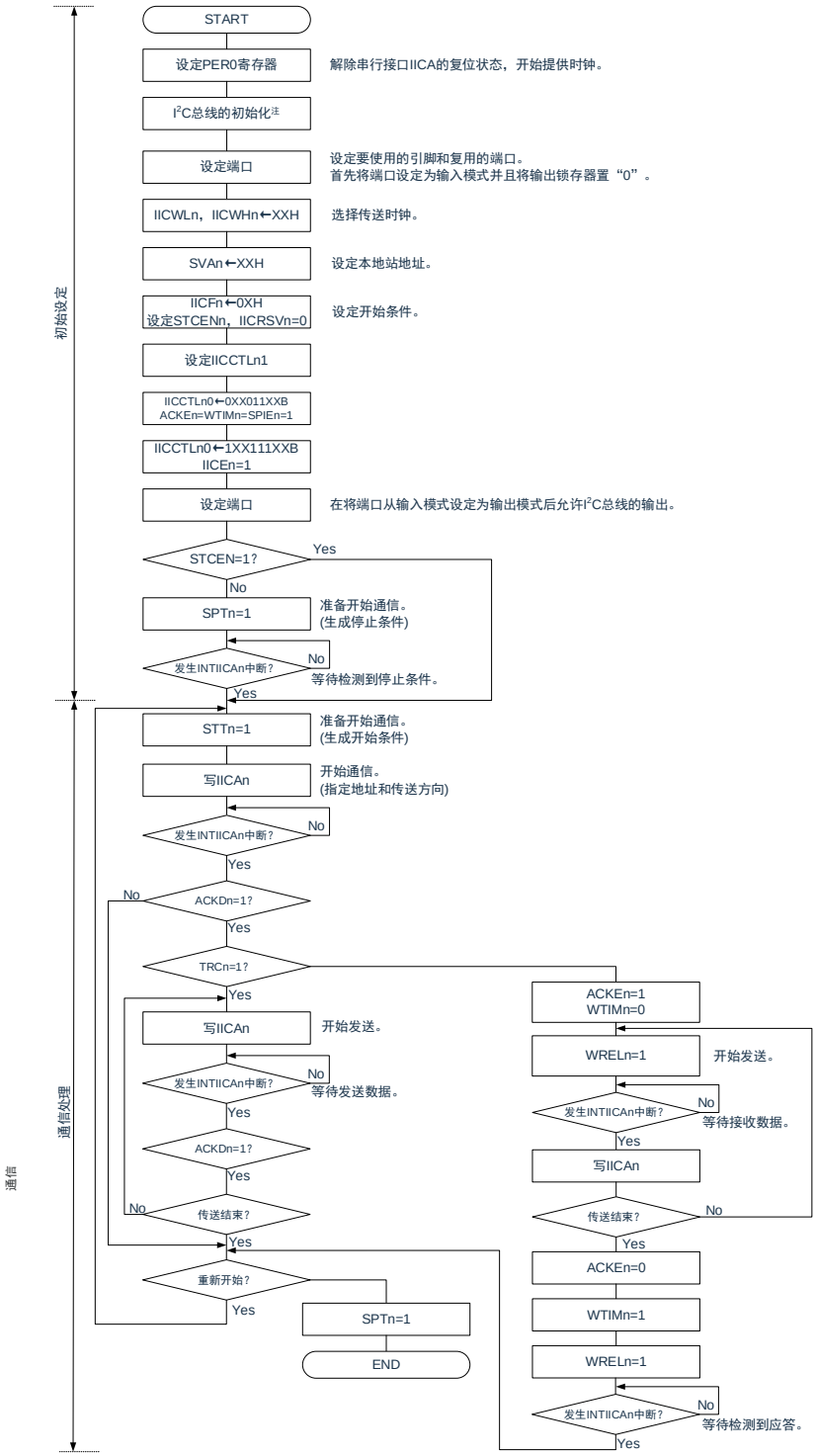
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待INTIICAn中断的发生。如果发生INTIICAn中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注：n=0

(1) 单主控系统的主控运行

图 18-27：单主控系统的主控运行



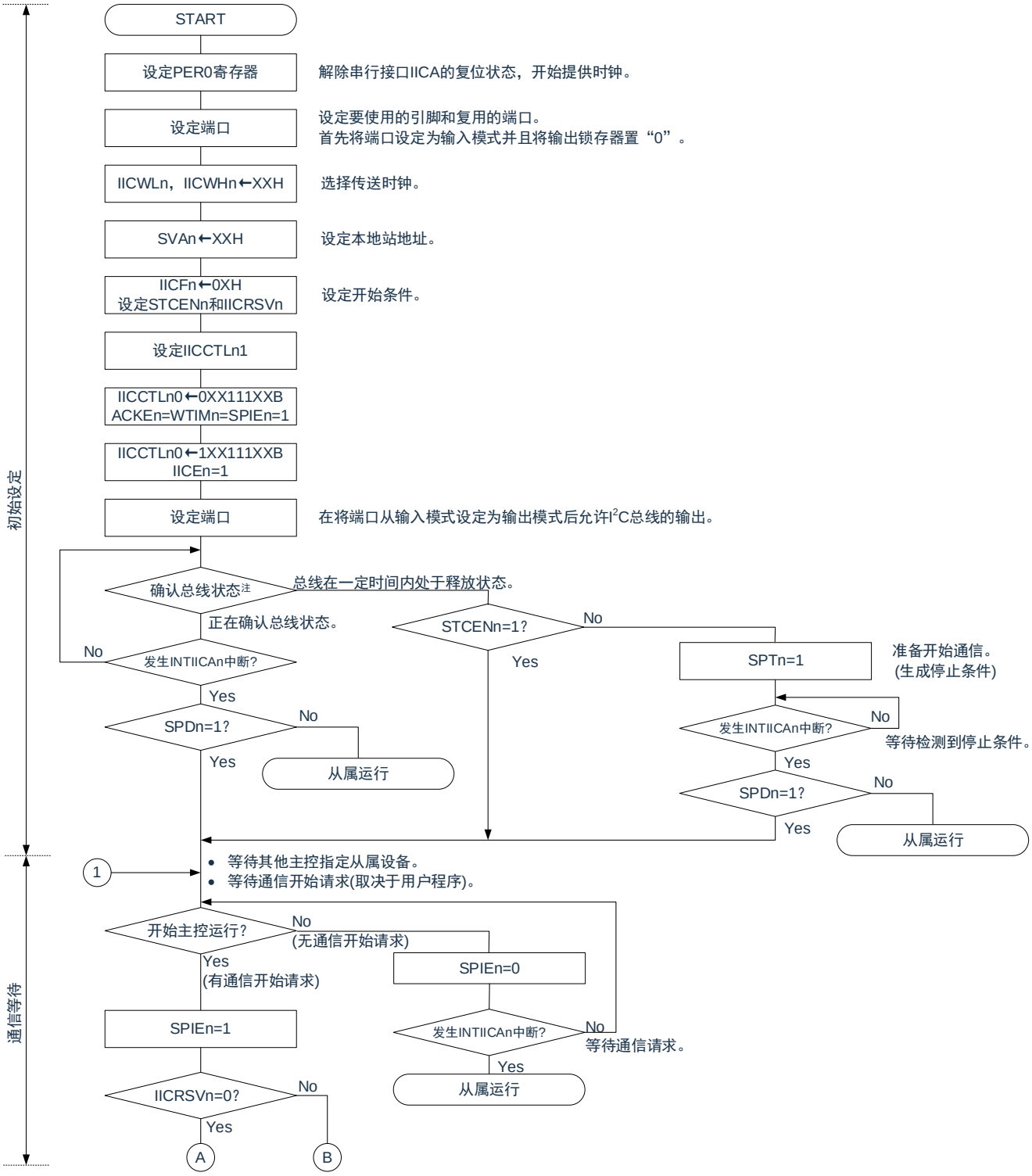
注：必须根据通信中的产品的规格，释放 I²C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。例如，如果 EEPROM 处于给 SDAAn 引脚输出低电平的状态，就必须将 SCLAn 引脚设定为输出端口，并且在 SDAAn 引脚固定为高电平前从输出端口输出时钟脉冲。

备注：

1. 发送和接收的格式必须符合通信中的产品的规格。
2. n=0。

(2) 多主控系统的主控运行

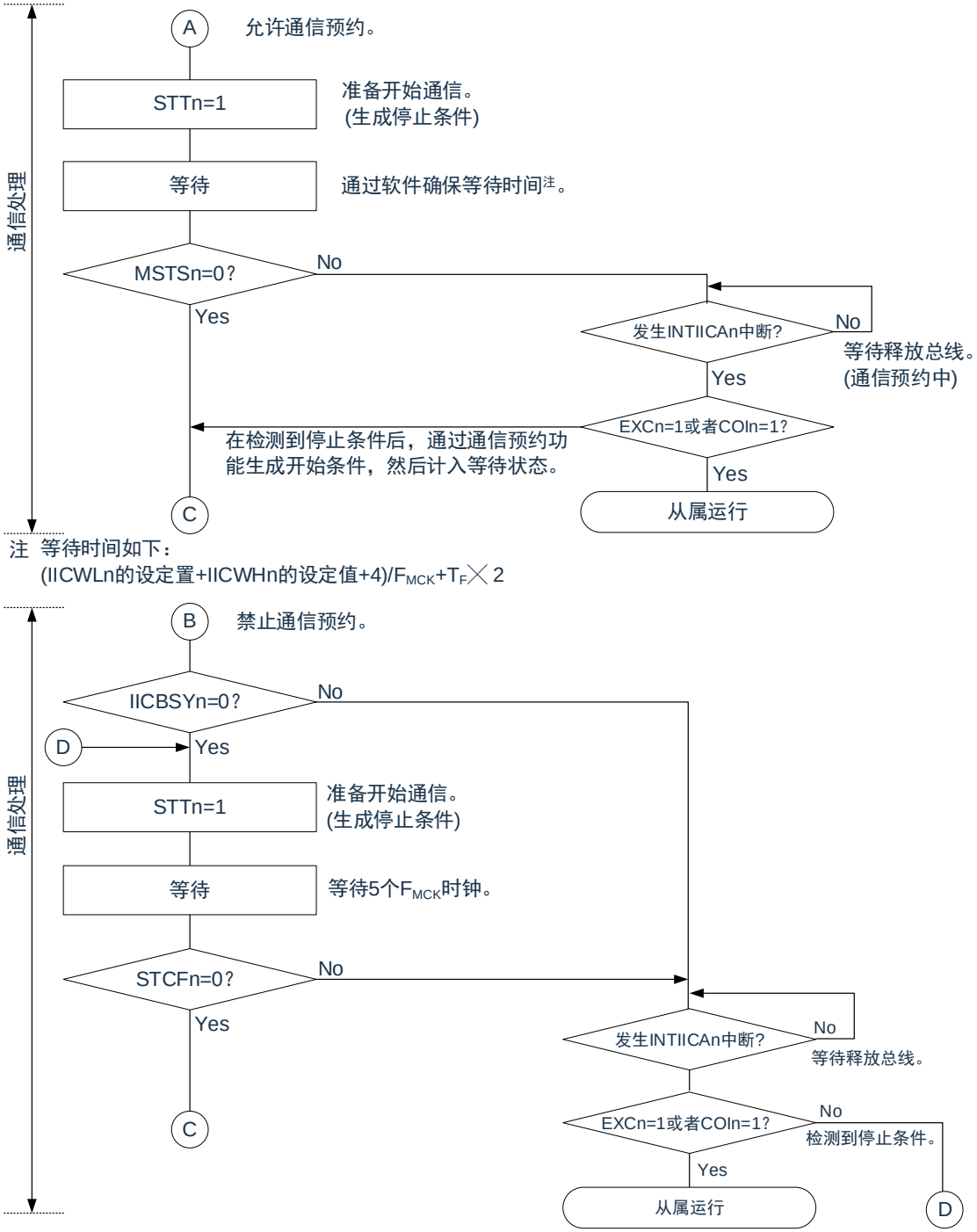
图 18-28：多主控系统的主控运行(1/3)



注：必须确认总线在一定时间内（例如，1 帧）处于释放状态（CLDn 位=1、DADn 位=1）。当 SDAAn 引脚固定为低电平时，必须根据通信中的产品的规格，判断是否释放 I²C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。

备注：n=0

图 18-28：多主控系统的主控运行(2/3)

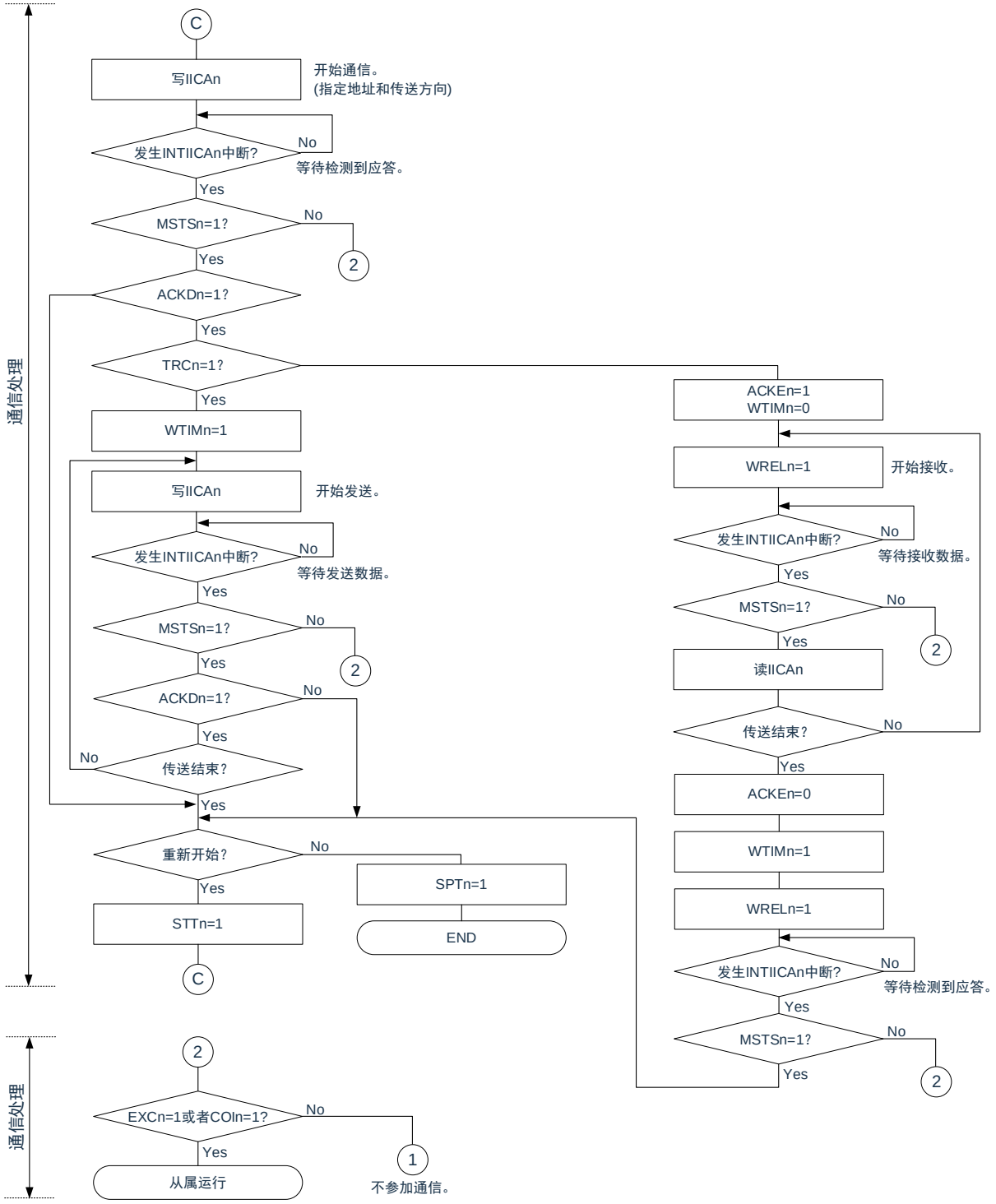


注 等待时间如下：
(IICWLn的设定置+IICWHn的设定值+4)/F_{MCK}+T_F×2

备注：

1. IICWLn：IICA 低电平宽度设定寄存器 n
   IICWHn：IICA 高电平宽度设定寄存器 n
   T_F：SDAAn 信号和 SCLAn 信号的下降时间
   F_{MCK}：IICA 运行时钟频率
2. n=0

图 18-28：多主控系统的主控运行(3/3)



备注：

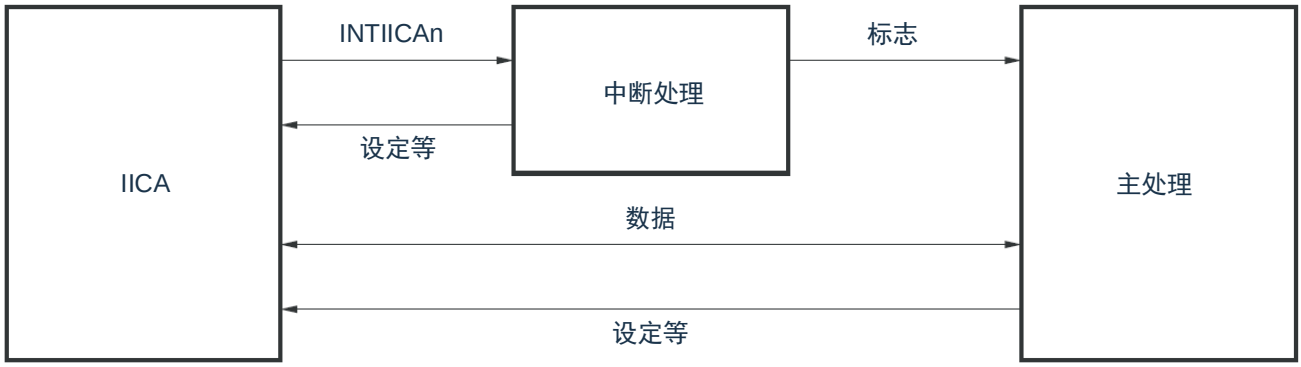
1. 传送和接收的格式必须符合通信中的产品的规格。
2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIICAn 中断时读 MSTSn 位，确认仲裁结果。
3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIICAn 中断时通过 IICA 状态寄存器 n（IICSn）和 IICA 标志寄存器 n（IICFn）确认状态，决定以后的处理。
4. n=0

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过INTIICAn中断进行处理（需要对通信中的停止条件检测等的运行状态进行很大的变更处理）。

在此说明中，假设数据通信不支持扩展码，INTIICAn中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下 3 个标志并且代替 INTIICAn 将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下2种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态（有效地址的检测~停止条件的检测，未检测到主控设备的应答，地址不同）

② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和 INTIICAn 中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第 1 个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第 1 个数据（地址匹配被解释为下一次数据请求）。

③ 通信方向标志

此标志表示通信方向，和 TRCn 位的值相同。

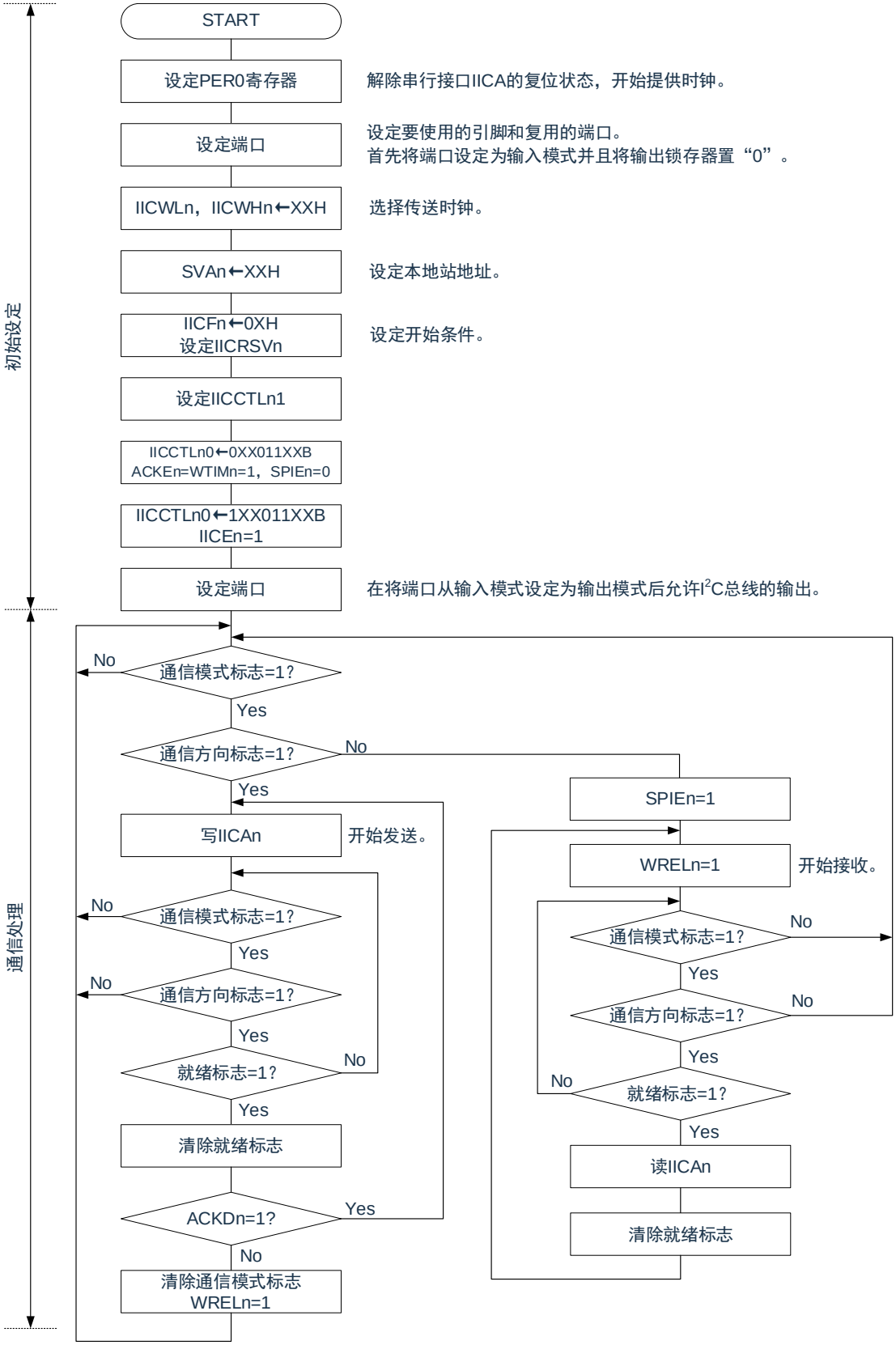
备注：n=0

从属运行的主处理部的运行如下所示。

启动串行接口 IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图 18-29：从属运行步骤(1)



备注：

1. 传送和接收的格式必须符合通信中的产品的规格。
2. n=0

从属设备通过 INTIICAn 中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过 INTIICAn 中断确认状态并且进行以下处理。

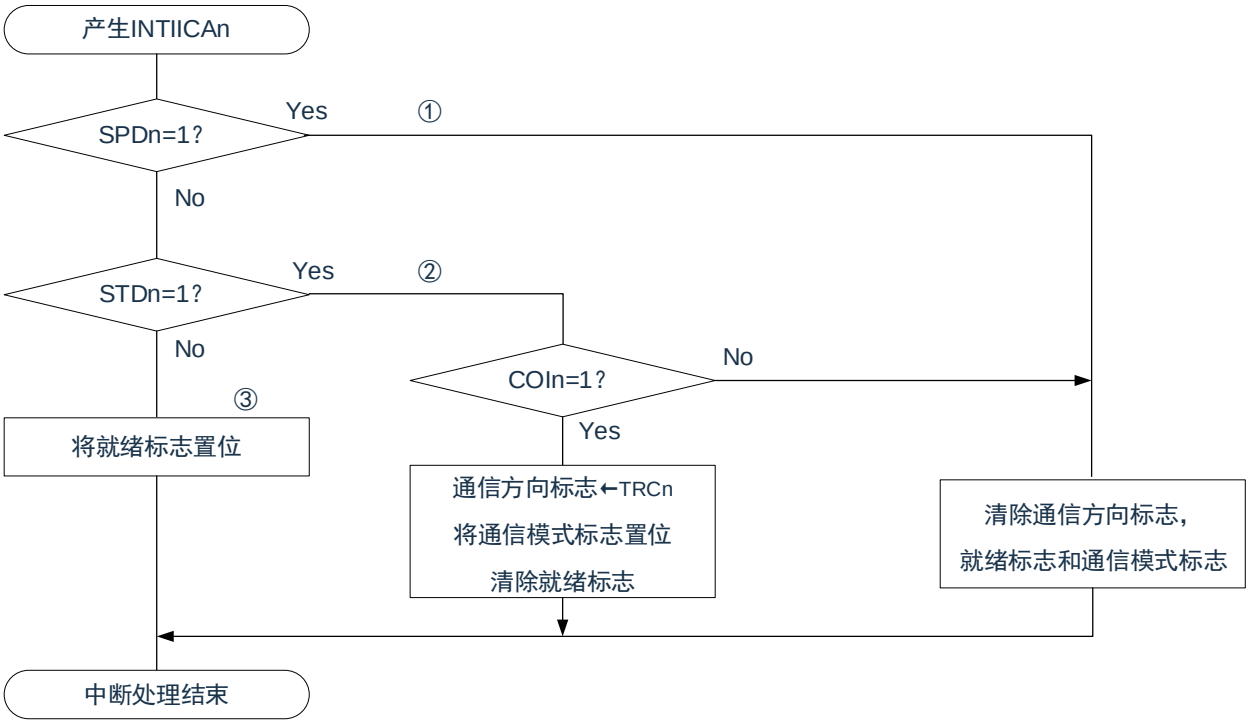
① 如果生成停止条件，就结束通信。

② 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设定为通信模式并且解除等待，然后从中断返回（清除就绪标志）。

③ 当发送和接收数据时，只要将就绪标志置位，I^2C总线就保持等待状态并且从中断返回。

备注：上述的①~③对应“图 18-30：从属运行步骤(2)”的①~③。

图 18-30：从属运行步骤(2)



备注：n=0

18.5.17 I²C中断请求（INTIICAn）的产生时序

数据的发送和接收时序、INTIICAn 中断请求信号的产生时序以及产生 INTIICAn 信号时的 IICA 状态寄存器 n（IICSn）的值如下所示。

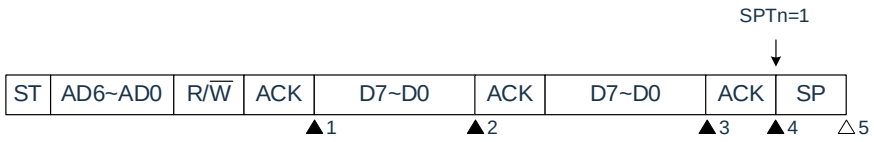
备注：

1. ST：开始条件
   AD6~AD0：地址
   R/W：传送方向的指定
   ACK：应答
   D7~D0：数据
   SP：停止条件
2. n=0

(1) 主控运行

(a) Start~Address~Data~Data~Stop（发送和接收）

(i) WTIMn=0 的情况

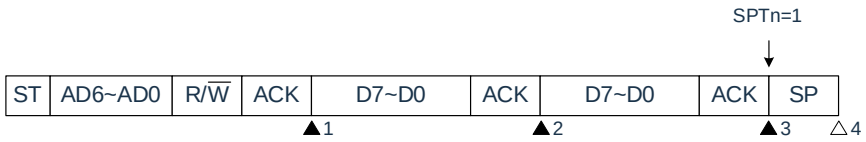


▲1:IICSn=1000X110B
▲2:IICSn=1000X000B
▲3:IICSn=1000X000B(将WTIMn位置“1”)注
▲4:IICSn=1000XX00B(将SPTn位置“1”)
△5:IICSn=00000001B

注　为了生成停止条件，必须将WTIMn位置“1”并且更改INTIICAn中断请求信号的产生时序。

备注　▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

(ii) WTIMn=1 的情况



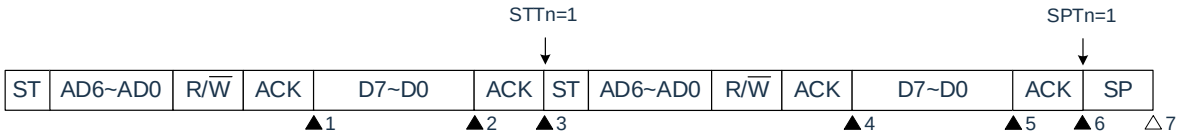
▲1:IICSn=1000X110B
▲2:IICSn=1000X100B
▲3:IICSn=1000XX00B(将SPTn位置“1”)
△4:IICSn=00000001B

备注　▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

备注：n=0

(b) Start~Address~Data~Start~Address~Data~Stop（重新开始）

(i) WTIMn=0 的情况

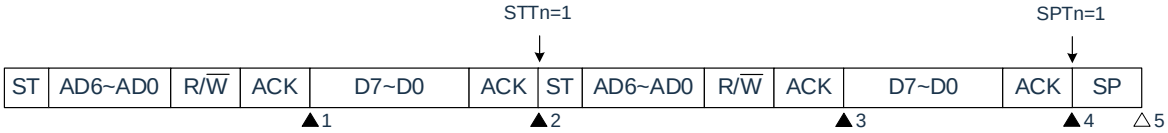


▲1:IICSn=1000X110B
▲2:IICSn=1000X000B(将WTIMn位置“1”)注1
▲3:IICSn=1000XX00B(将WTIMn位清“0”注2并且将STTn位置“1”)
▲4:IICSn=1000X110B
▲5:IICSn=1000X000B(将WTIMn位置“1”)注3
▲6:IICSn=1000XX00B(将SPTn位置“1”)
△7:IICSn=00000001B

注1. 为了生成开始条件，必须将WTIMn位置“1”并且更改INTIICAn中断请求信号的产生时序。
2. 为了恢复原来的设定，必须将WTIMn位清“0”.
3. 为了生成停止条件，必须将WTIMn位置“1”并且更改INTIICAn中断请求信号的产生时序。

备注 ▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

(ii) WTIMn=1 的情况



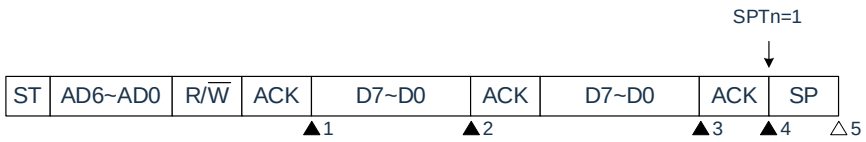
▲1:IICSn=1000X110B
▲2:IICSn=1000XX00B(将STTn位置“1”)
▲3:IICSn=1000X110B
▲4:IICSn=1000XX00B(将SPTn位置“1”)
△5:IICSn=00000001B

备注 ▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

备注：n=0

(c) Start~Code~Data~Data~Stop（发送扩展码）

(i) WTIMn=0 的情况

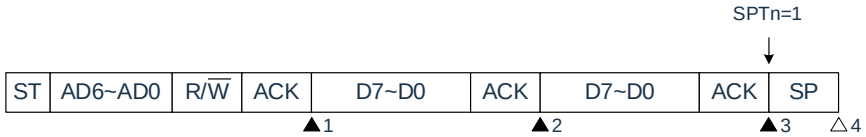


▲1:IICSn=1010X110B
▲2:IICSn=1010X000B
▲3:IICSn=1010X000B(将WTIMn位置“1”)注
▲4:IICSn=1010XX00B(将SPTn位置“1”)
△5:IICSn=00000001B

注　为了生成停止条件，必须将WTIMn位置“1”并且更改INTIICAn中断请求信号的产生时序。

备注　▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

(ii) WTIMn=1 的情况



▲1:IICSn=1010X110B
▲2:IICSn=1010X100B
▲3:IICSn=1010XX00B(将SPTn位置“1”)
△4:IICSn=00000001B

备注　▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

备注：n=0

(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

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1	▲2		▲3	△4

▲1:IICSn=0001X110B
▲2:IICSn=0001X000B
▲3:IICSn=0001X000B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1		▲2	▲3	△4

▲1:IICSn=0001X110B
▲2:IICSn=0001X100B
▲3:IICSn=0001XX00B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(b) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1	▲2					▲3	▲4	△5

▲1:IICSn=0001X110B
▲2:IICSn=0001X000B
▲3:IICSn=0001X110B
▲4:IICSn=0001X000B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后SVAn相同）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1		▲2				▲3		▲4 △5

▲1:IICSn=0001X110B
▲2:IICSn=0001XX00B
▲3:IICSn=0001X110B
▲4:IICSn=0001XX00B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(c) Start~Address~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1	▲2				▲3		▲4	△5

▲1:IICSn=0001X110B
▲2:IICSn=0001X000B
▲3:IICSn=0010X010B
▲4:IICSn=0010X000B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后地址不同（扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1		▲2			▲3	▲4	▲5	△6

▲1:IICSn=0001X110B
▲2:IICSn=0001XX00B
▲3:IICSn=0010X010B
▲4:IICSn=0010X110B
▲5:IICSn=0010XX00B
△6:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(d) Start~Address~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（非扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1	▲2					▲3		△4

▲1:IICSn=0001X110B
▲2:IICSn=0001X000B
▲3:IICSn=00000110B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后地址不同（非扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1		▲2				▲3		△4

▲1:IICSn=0001X110B
▲2:IICSn=0001XX00B
▲3:IICSn=00000110B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(3) 从属运行（接收扩展码的情况）

在接收扩展码时，始终参加通信。

(a) Start~Code~Data~Data~Stop

(i) WTIMn=0的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1		▲2		▲3	△4

▲1:IICSn=0010X010B
▲2:IICSn=0010X000B
▲3:IICSn=0010X000B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3	▲4	△5

▲1:IICSn=0010X010B
▲2:IICSn=0010X110B
▲3:IICSn=0010X100B
▲4:IICSn=0010XX00B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(b) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后SVAn相同）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1		▲2					▲3	▲4	△5

▲1:IICSn=0010X010B
▲2:IICSn=0010X000B
▲3:IICSn=0001X110B
▲4:IICSn=0001X000B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后SVAn相同）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3				▲4	▲5	△6

▲1:IICSn=0010X010B
▲2:IICSn=0010X110B
▲3:IICSn=0010XX00B
▲4:IICSn=0001X110B
▲5:IICSn=0001XX00B
△6:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(c) Start~Code~Data~Start~Code~Data~Stop

(i) WTIMn=0的情况（在重新开始后接收扩展码）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1:IICSn=0010X010B
▲2:IICSn=0010X000B
▲3:IICSn=0010X010B
▲4:IICSn=0010X000B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后接收扩展码）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3			▲4	▲5	▲6	△7

▲1:IICSn=0010X010B
▲2:IICSn=0010X110B
▲3:IICSn=0010XX00B
▲4:IICSn=0010X010B
▲5:IICSn=0010X110B
▲6:IICSn=0010XX00B
△7:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(d) Start~Code~Data~Start~Address~Data~Stop

(i) WTIMn=0的情况（在重新开始后地址不同（非扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1		▲2					▲3		△4

▲1:IICSn=0010X010B
▲2:IICSn=0010X000B
▲3:IICSn=00000X10B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况（在重新开始后地址不同（非扩展码））

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
			▲1	▲2		▲3				▲4		△5

▲1:IICSn=0010X010B
▲2:IICSn=0010X110B
▲3:IICSn=0010XX00B
▲4:IICSn=00000X10B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(4) 不参加通信的运行

(a) Start~Code~Data~Data~Stop

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
								△1

△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

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1	▲2		▲3	△4

▲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

(ii) WTIMn=1的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1		▲2	▲3	△4

▲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

备注：n=0

(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIMn=0 的情况

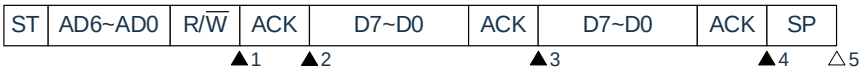
▲1:IICSn=0110X010B
▲2:IICSn=0010X000B
▲3:IICSn=0010X000B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1 的情况



▲1:IICSn=0110X010B
▲2:IICSn=0010X110B
▲3:IICSn=0010X100B
▲4:IICSn=0010XX00B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

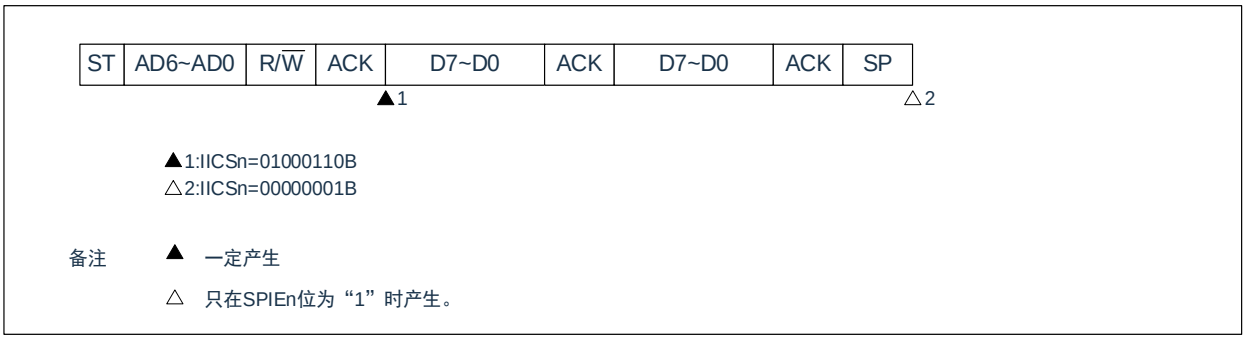
X 任意

备注：n=0

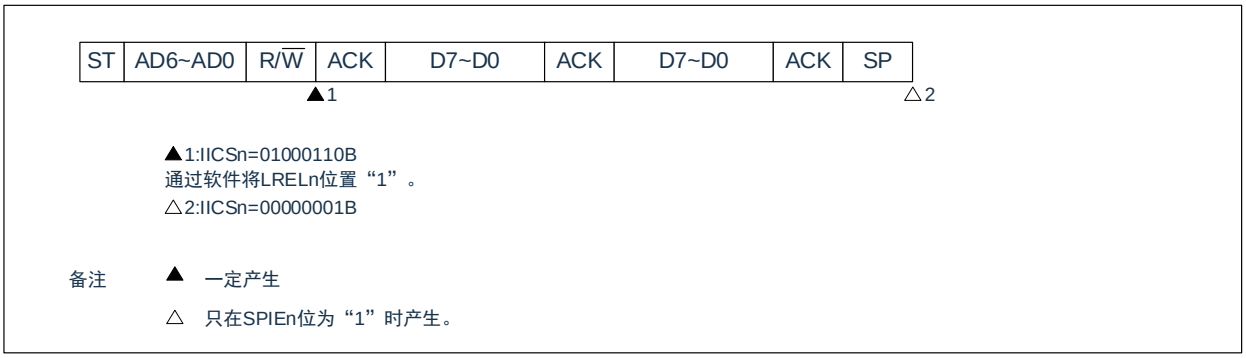
(6) 仲裁失败的运行（在仲裁失败后不参加通信）

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 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）



(b) 在发送扩展码的过程中仲裁失败的情况



备注：n=0

(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1	▲2			△3

▲1:IICSn=10001110B
▲2:IICSn=01000000B
△3:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	D7~D0	ACK	SP
				▲1		▲2		△3

▲1:IICSn=10001110B
▲2:IICSn=01000100B
△3:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

备注：n=0

(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码（例如，SVAn 不同）

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1						▲2		△3

▲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

(ii) 扩展码

ST	AD6~AD0	R/$\overline{W}$	ACK	D7~Dm	ACK	ST	AD6~AD0	R/$\overline{W}$	ACK	D7~D0	ACK	SP
				▲1					▲2			△3

▲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

备注：n=0

(e) 在传送数据时因停止条件而仲裁失败的情况

▲1:IICSn=10000110B
△2:IICSn=01000001B

备注 ▲ 一定产生

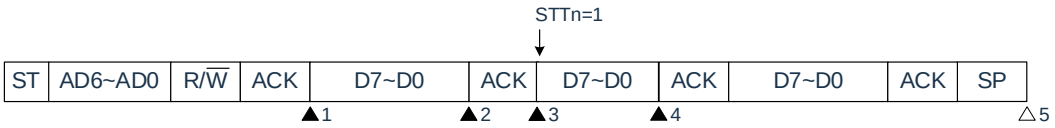
△ 只在SPIEn位为“1”时产生。

m=0~6

备注：n=0

(f) 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



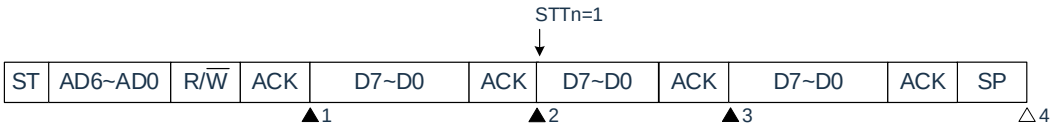
▲1:IICSn=1000X110B
▲2:IICSn=1000X000B(将WTIMn位置“1”)
▲3:IICSn=1000X100B(将WTIMn位清“0”)
▲4:IICSn=01000000B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况



▲1:IICSn=1000X110B
▲2:IICSn=1000X100B(将STTn位置“1”)
▲3:IICSn=01000100B
△4:IICSn=00000001B

备注 ▲ 一定产生

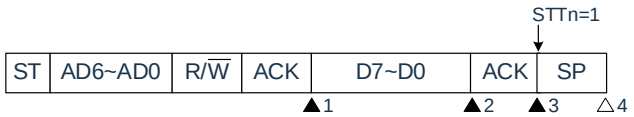
△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

(g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

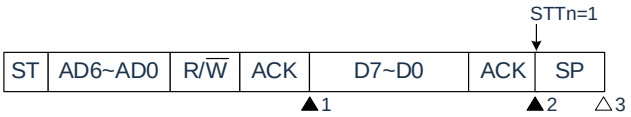
(i) WTIMn=0的情况



▲1:IICSn=1000X110B
▲2:IICSn=1000X000B(将WTIMn位置“1”)
▲3:IICSn=1000XX00B(将STTn位置“1”)
△4:IICSn=01000001B

备注 ▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

(ii) WTIMn=1的情况



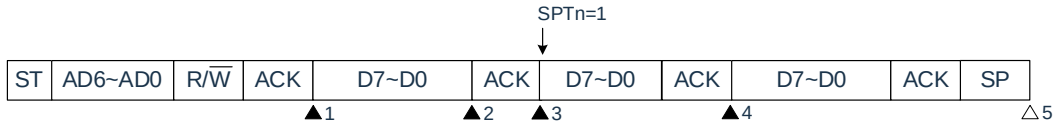
▲1:IICSn=1000X110B
▲2:IICSn=1000XX00B(将STTn位置“1”)
△3:IICSn=01000001B

备注 ▲ 一定产生
△ 只在SPIEn位为“1”时产生。
X 任意

备注：n=0

(h) 在想要生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



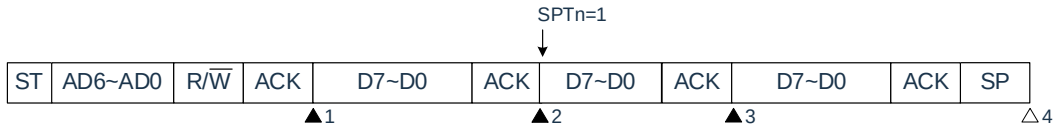
▲1:IICSn=1000X110B
▲2:IICSn=1000X000B(将WTIMn位置“1”)
▲3:IICSn=1000X100B(将WTIMn位清“0”)
▲4:IICSn=01000100B
△5:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

(ii) WTIMn=1的情况



▲1:IICSn=1000X110B
▲2:IICSn=1000X100B(将SPTn位置“1”)
▲3:IICSn=01000100B
△4:IICSn=00000001B

备注 ▲ 一定产生

△ 只在SPIEn位为“1”时产生。

X 任意

备注：n=0

18.6 时序图

在 I²C 总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。主控设备在从属设备地址之后发送表示数据传送方向的 TRCn 位（IICA 状态寄存器 n（IICSn）的 bit3），开始与从属设备进行串行通信。数据通信的时序图如图 18-31 和图 18-32 所示。

与串行时钟（SCLAn）的下降沿同步进行 IICA 移位寄存器 n（IICAn）的移位，并且将发送数据传送到 SO 锁存器，以 MSB 优先从 SDAAn 引脚输出数据。

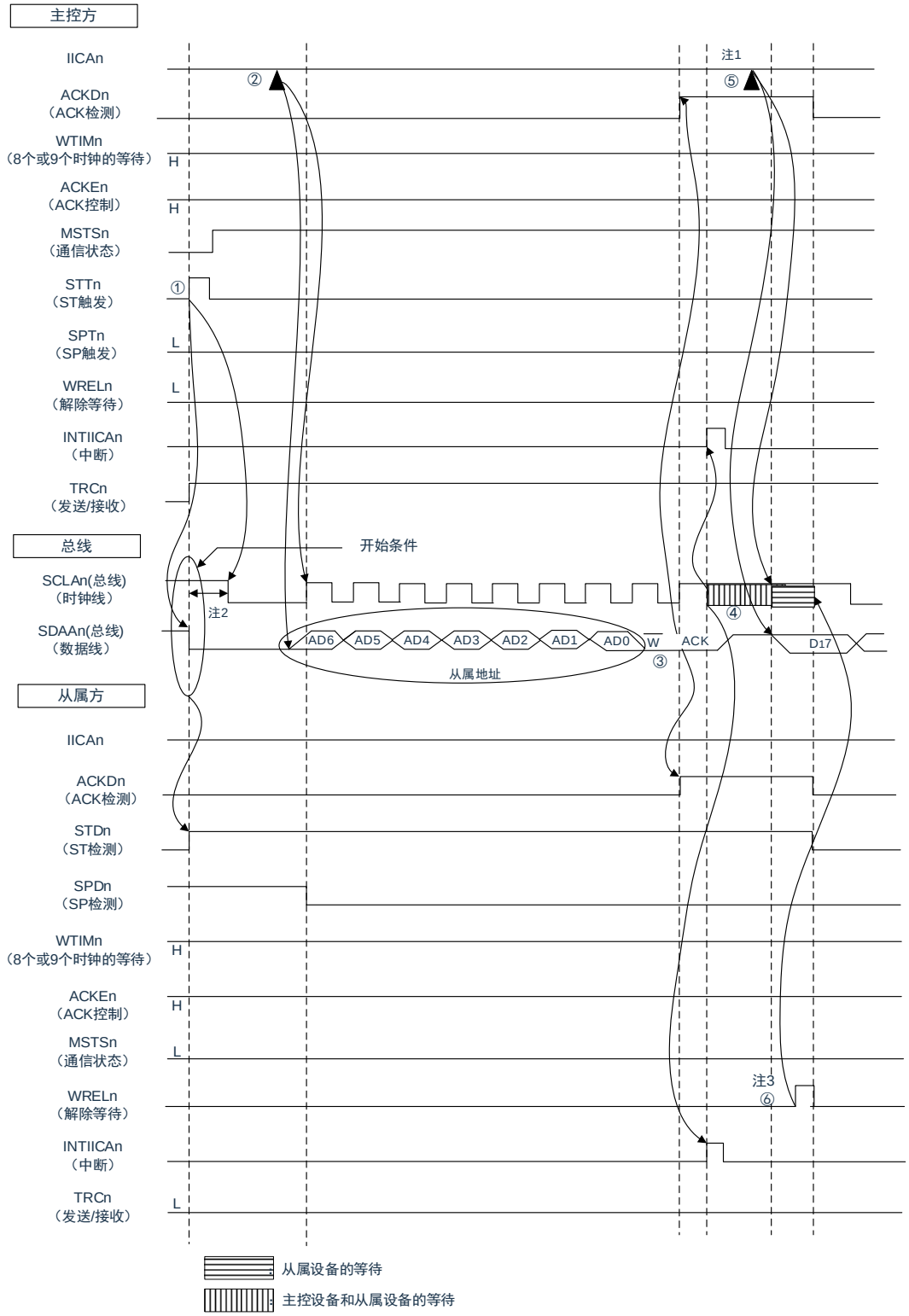
在 SCLAn 的上升沿将 SDAAn 引脚输入的数据取到 IICAn。

备注：n=0

图 18-31：主控设备→从属设备的通信例子

（主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待）(1/4)

(1) 开始条件~地址~数据



注 1：要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2：从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设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 18-31 的“(1)开始条件~地址~数据”的①~⑥的说明如下：

(7) 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过SCLAn=1使SDAAn从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTSn=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。

②如果主控方给IICA移位寄存器n（IICAn）写地址+W（发送），就发送从属地址。

③在从属方，如果接收地址和本地站地址（SVAn的值）相同[注]，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

④主控方在第9个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）[注]。

⑤主控方给IICAn寄存器写发送数据，解除主控方的等待。

⑥如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

注：如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

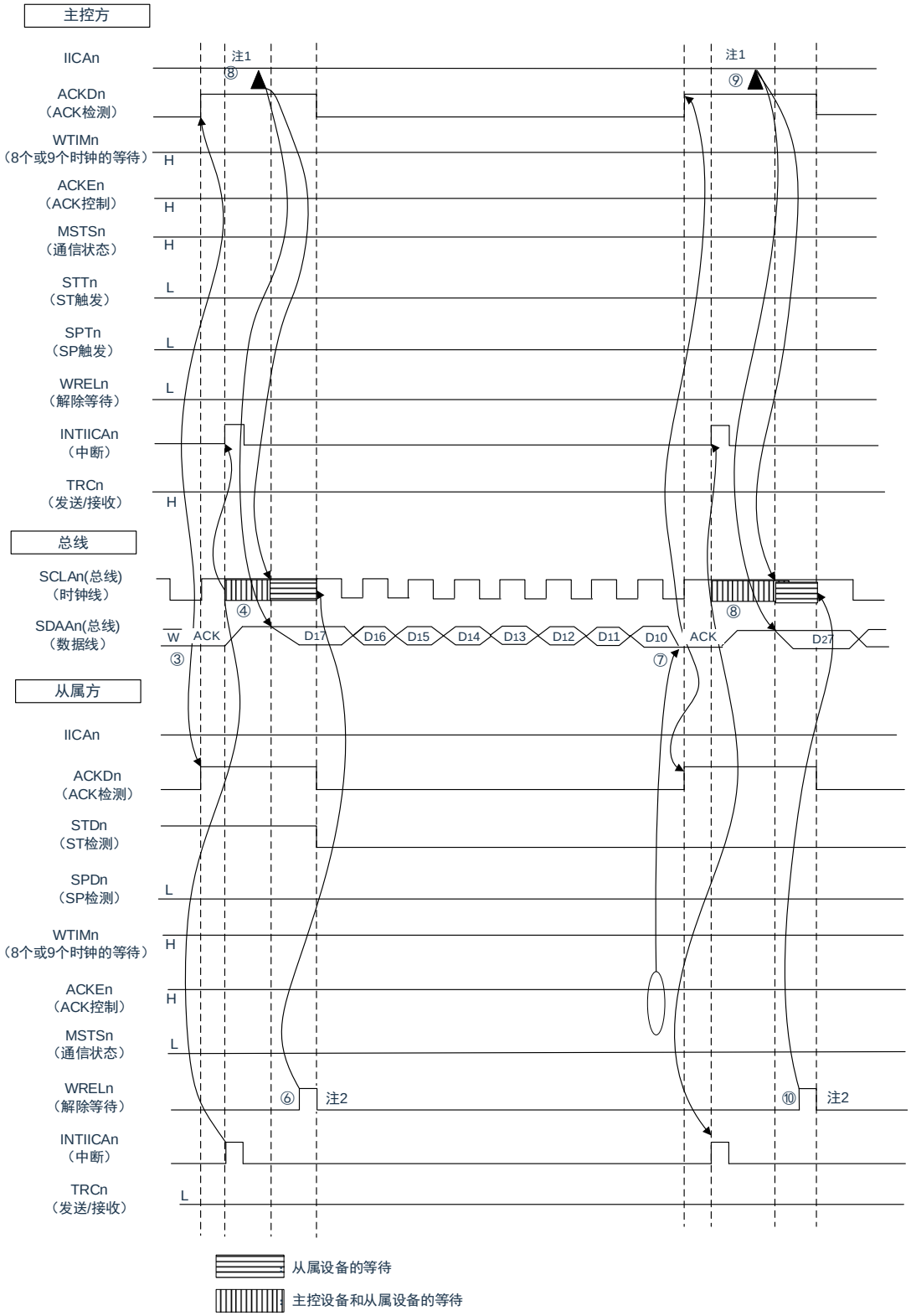
备注：

1. 图 18-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。
   图 18-31 的“(1)开始条件~地址~数据”说明步骤①~⑥。
   图 18-31 的“(2)地址~数据~数据”说明步骤③~⑩。
   图 18-31 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图 18-31：主控设备→从属设备的通信例子

（主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待）(2/4)

(2) 地址~数据~数据



注 1：要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2：要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

图 18-31 的“(2)地址~数据~数据”的③~⑩的说明如下：

③在从属方，如果接收地址和本地站地址（SVAn 的值）相同[注]，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

④主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）[注]。

⑤主控方给 IICA 移位寄存器 n（IICAn）写发送数据，解除主控方的等待。

⑥如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

⑦在数据传送结束后，因为从属方的 ACKEn 位为“1”，所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

⑧主控方和从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

⑨主控方给 IICAn 寄存器写发送数据，解除主控方的等待。

⑩如果从属方读接收数据并且解除等待（WRELn=1），主控方就开始给从属方传送数据

注：如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

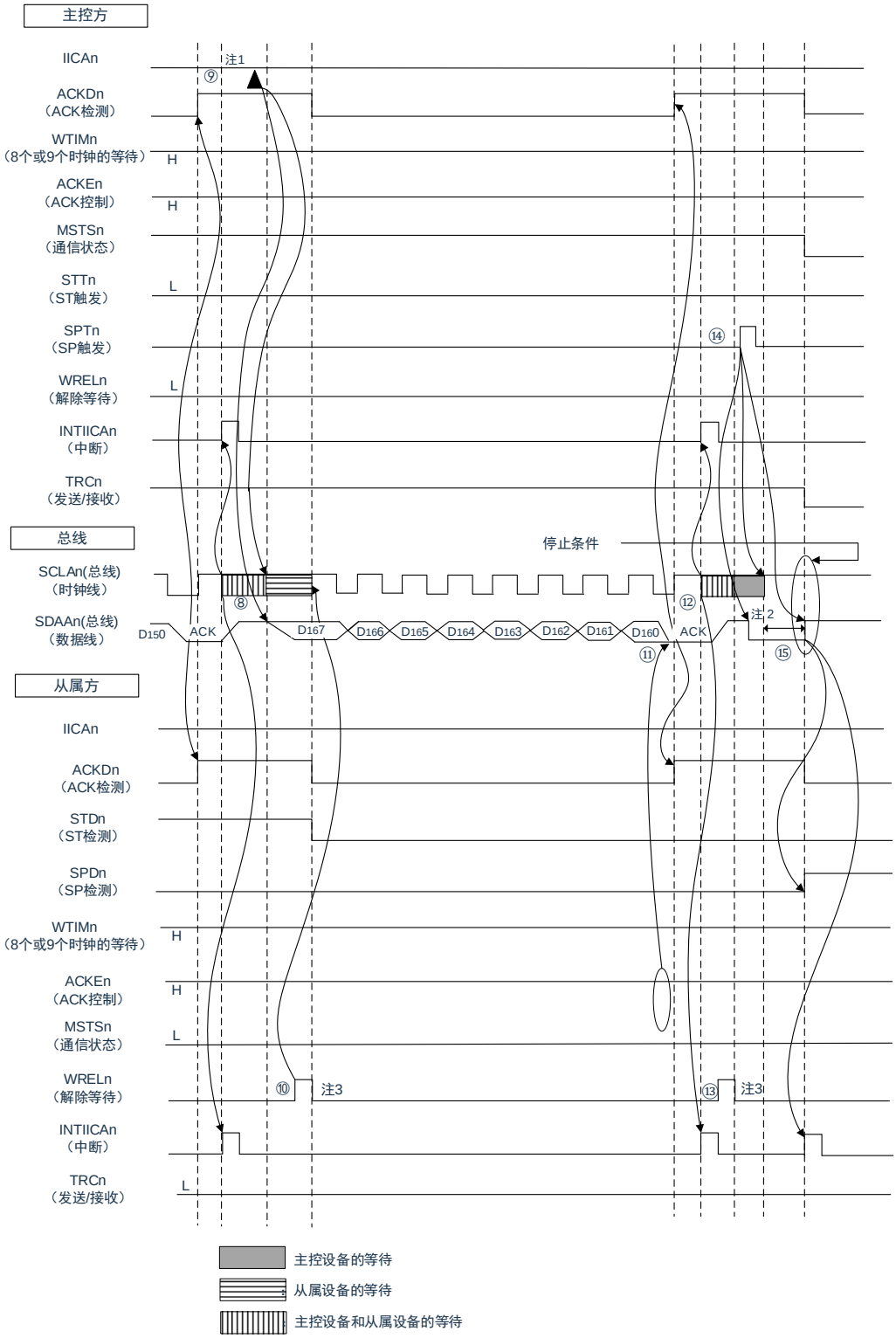
备注：

1. 图 18-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。
   图 18-31 的“(1)开始条件~地址~数据”说明步骤①~⑥。
   图 18-31 的“(2)地址~数据~数据”说明步骤③~⑩。
   图 18-31 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图 18-31：主控设备→从属设备的通信例子

（主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待）(3/4)

(3) 数据~数据~停止条件



注 1：要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

注 2：在发行停止条件后，从 SCLAn 引脚信号上升到生成停止条件的时间，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 18-31 的“(3)数据~数据~停止条件”的⑦~⑮的说明如下：

⑦在数据传送结束后，因为从属方的ACKEn位为“1”，所以通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

⑧主控方和从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

⑨主控方给IICA移位寄存器n（IICAn）写发送数据，解除主控方的等待。

⑩如果从属方读接收数据并且解除等待（WRELn=1），主控方就开始给从属方传送数据

⑪在数据传送结束后，从属方（ACKEn=1）通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

⑫主控方和从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

⑬从属方读接收数据，解除等待（WRELn=1）。

⑭如果在主控方将停止条件触发置位（SPTn=1），就清除总线数据线（SDAAn=0）并且将总线时钟线置位（SCLAn=1），在经过停止条件的准备时间后将总线数据线置位（SDAAn=1），生成停止条件（通过SCLAn=1使SDAAn从“0”变为“1”）。

⑮如果生成停止条件，从属方就检测到停止条件并且产生中断（INTIICAn：停止条件中断）。

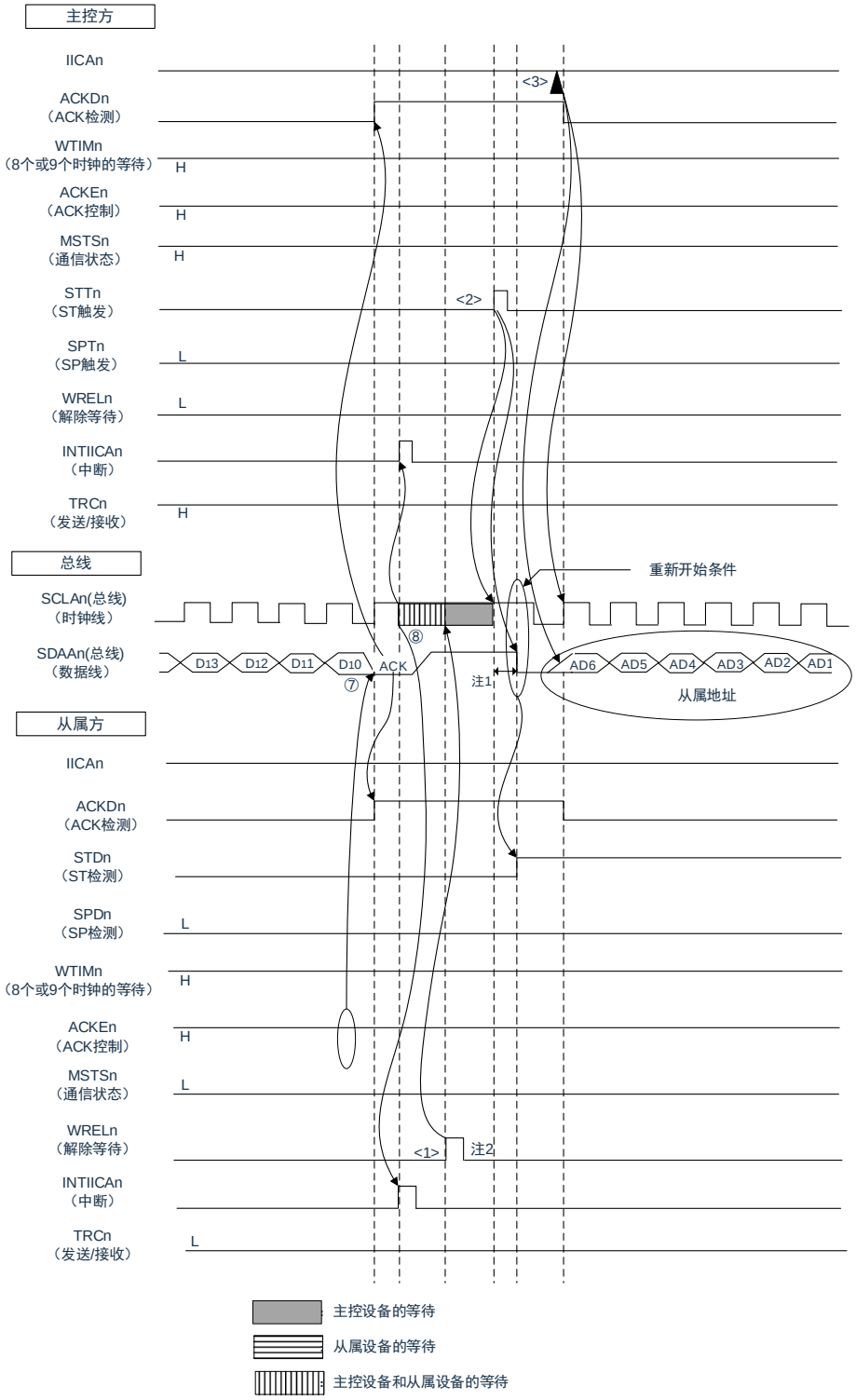
备注：

1. 图 18-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。
   图 18-31 的“(1)开始条件~地址~数据”说明步骤①~⑥。
   图 18-31 的“(2)地址~数据~数据”说明步骤③~⑩。
   图 18-31 的“(3)数据~数据~停止条件”说明步骤⑦~⑮。
2. n=0

图 18-31：主控设备→从属设备的通信例子

（主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待）(4/4)

(4) 数据~重新开始条件~地址



注 1：在发行重新开始条件后，从 SCLAn 引脚信号上升到生成开始条件的时间，在设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。

图 17-31 的“(4)数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行＜1＞~＜3＞，从而返回到步骤③的数据发送步骤。

⑦在数据传送结束后，因为从属方的 ACKEn 位为“1”，所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

⑧主控方和从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

(8) 从属方读接收数据，解除等待（WRELn=1）。

＜2＞如果在主控方再次将开始条件触发置位（STTn=1），总线时钟线就上升（SCLAn=1），而且在经过重新开始条件的准备时间后总线数据线下降（SDAAn=0），生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。然后，如果检测到开始条件，就在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。

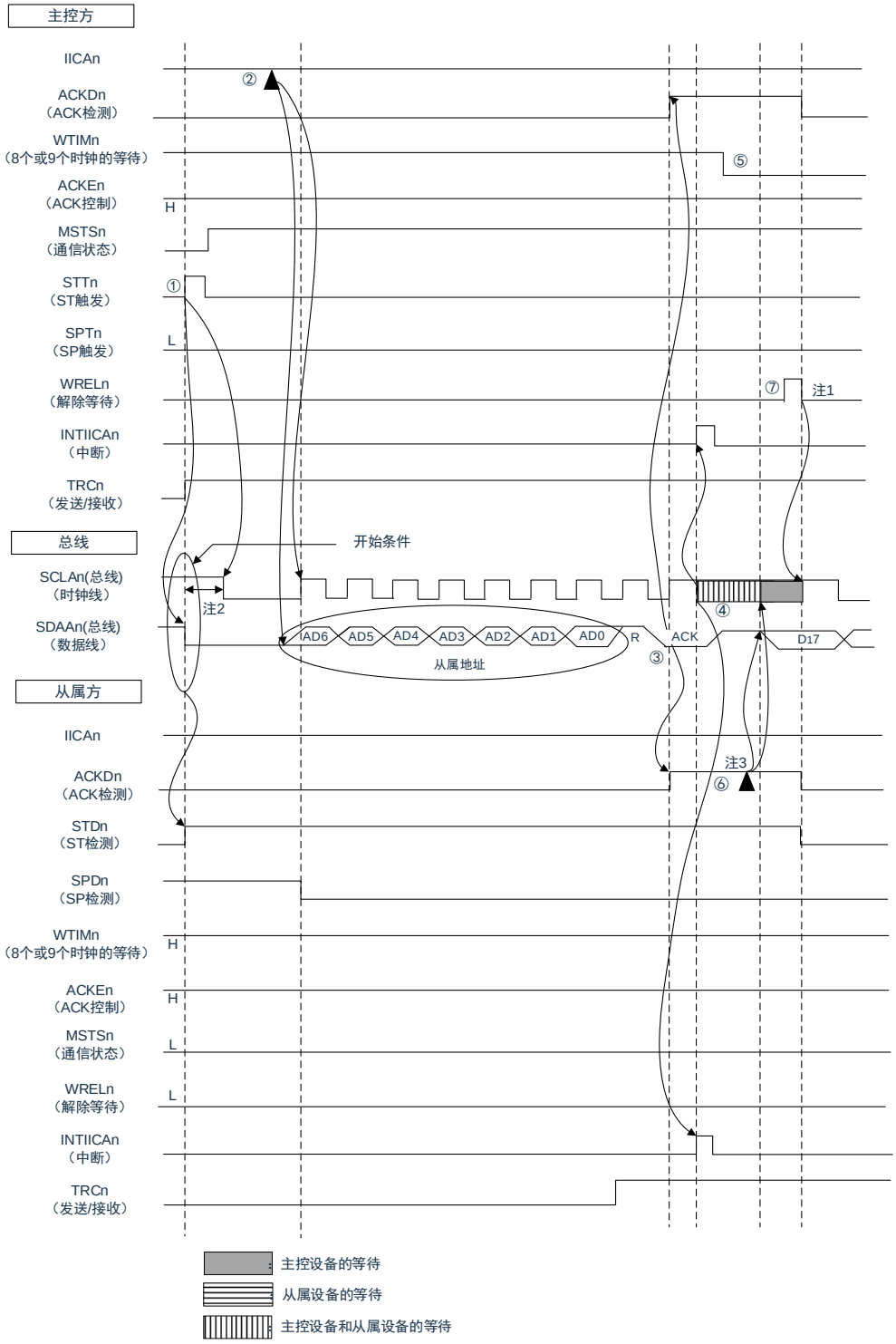
＜3＞如果主控方给 IICA 移位寄存器 n（IICAn）写地址+R/W（发送），就发送从属地址。

备注：n=0

图 18-32：从属设备→主控设备的通信例子

（主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待）(1/3)

(1) 开始条件~地址~数据



注 1：要解除主控方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2：从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设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 18-32 的“(1)开始条件~地址~数据”的①~⑦的说明如下：

①如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过SCLAn=1使SDAAn从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTSn=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。

②如果主控方给IICA移位寄存器n（IICAn）写地址+R（接收），就发送从属地址。

③在从属方，如果接收地址和本地站地址（SVAn的值）相同[注]，就通过硬件给主控方发送ACK。主控方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

④主控方在第9个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）[注]。

⑤主控方将等待时序改为第8个时钟（WTIMn=0）。

⑥从属方给IICAn寄存器写发送数据，解除从属方的等待。

⑦主控方解除等待（WRELn=1），开始来自从属设备的数据传送。

注：如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

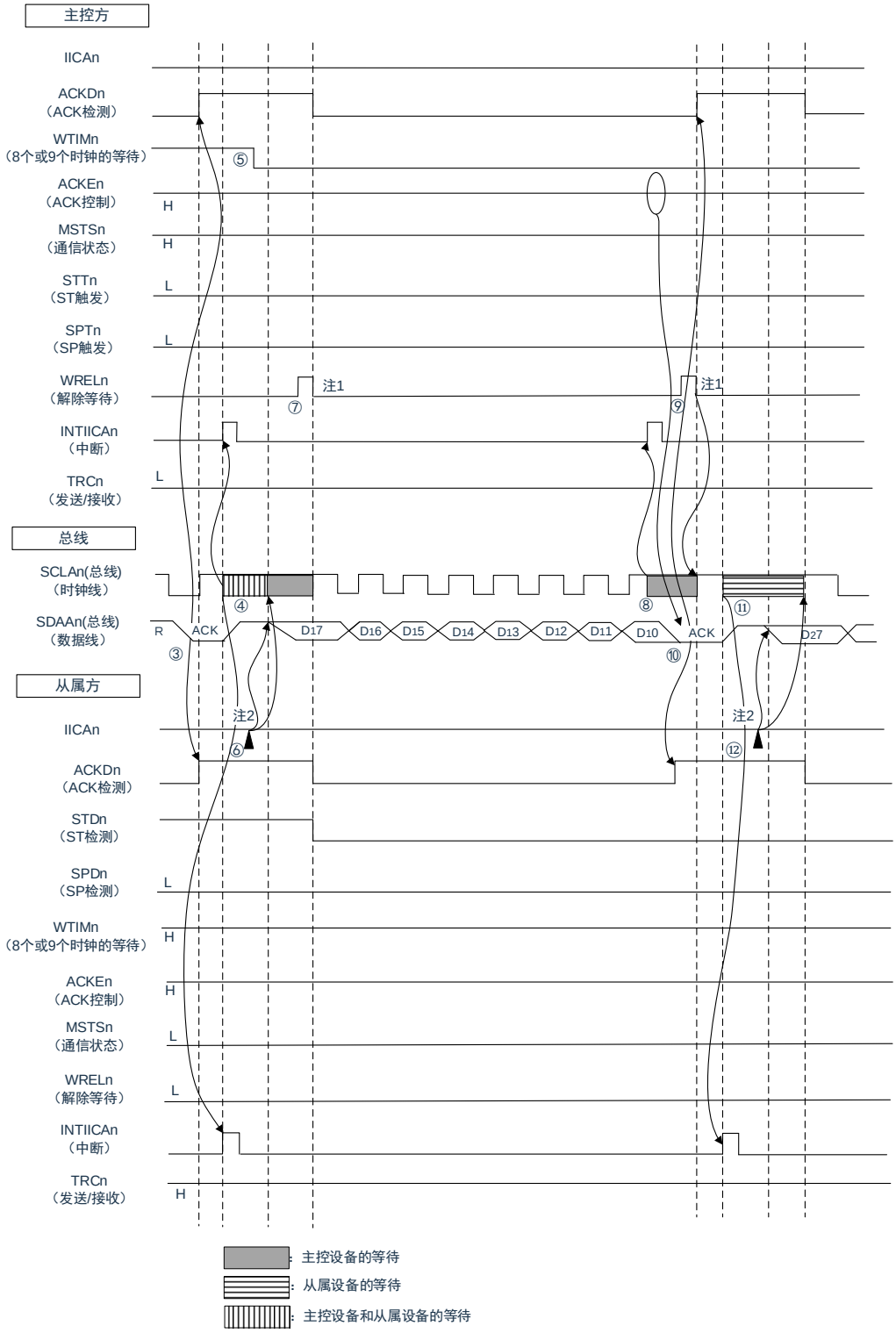
备注：

1. 图 18-32 的①~⑲是通过 I²C 总线进行数据通信的一系列运行步骤。
   图 18-32 的“(1)开始条件~地址~数据”说明步骤①~⑦。
   图 18-32 的“(2)地址~数据~数据”说明步骤③~⑫。
   图 18-32 的“(3)数据~数据~停止条件”说明步骤⑧~⑲。
2. n=0

图 18-32：从属设备→主控设备的通信例子

（主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待）(2/3)

(2) 地址~数据~数据



注 1：要解除主控方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2：要解除从属方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

图 18-32 的“(2)地址~数据~数据”的③~⑫的说明如下：

③在从属方，如果接收地址和本地站地址（SVAn 的值）相同[注]，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

④主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）[注]。

⑤主控方将等待时序改为第 8 个时钟（WTIMn=0）。

⑥从属方给 IICA 移位寄存器 n（IICAn）写发送数据，解除从属方的等待。

⑦主控方解除等待（WRELn=1），开始来自从属设备的数据传送。

⑧主控方在第 8 个时钟的下降沿进入等待状态（SCLAn=0），并产生中断（INTIICAn：传送结束中断）。因为主控方的 ACKEn 位为“1”，所以通过硬件给从属方发送 ACK。

⑨主控方读接收数据，解除等待（WRELn=1）。

⑩从属方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。

⑪从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）

⑫如果从属方给 IICAn 寄存器写发送数据，就解除从属方的等待，开始从属方到主控方的数据传送。

注：如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

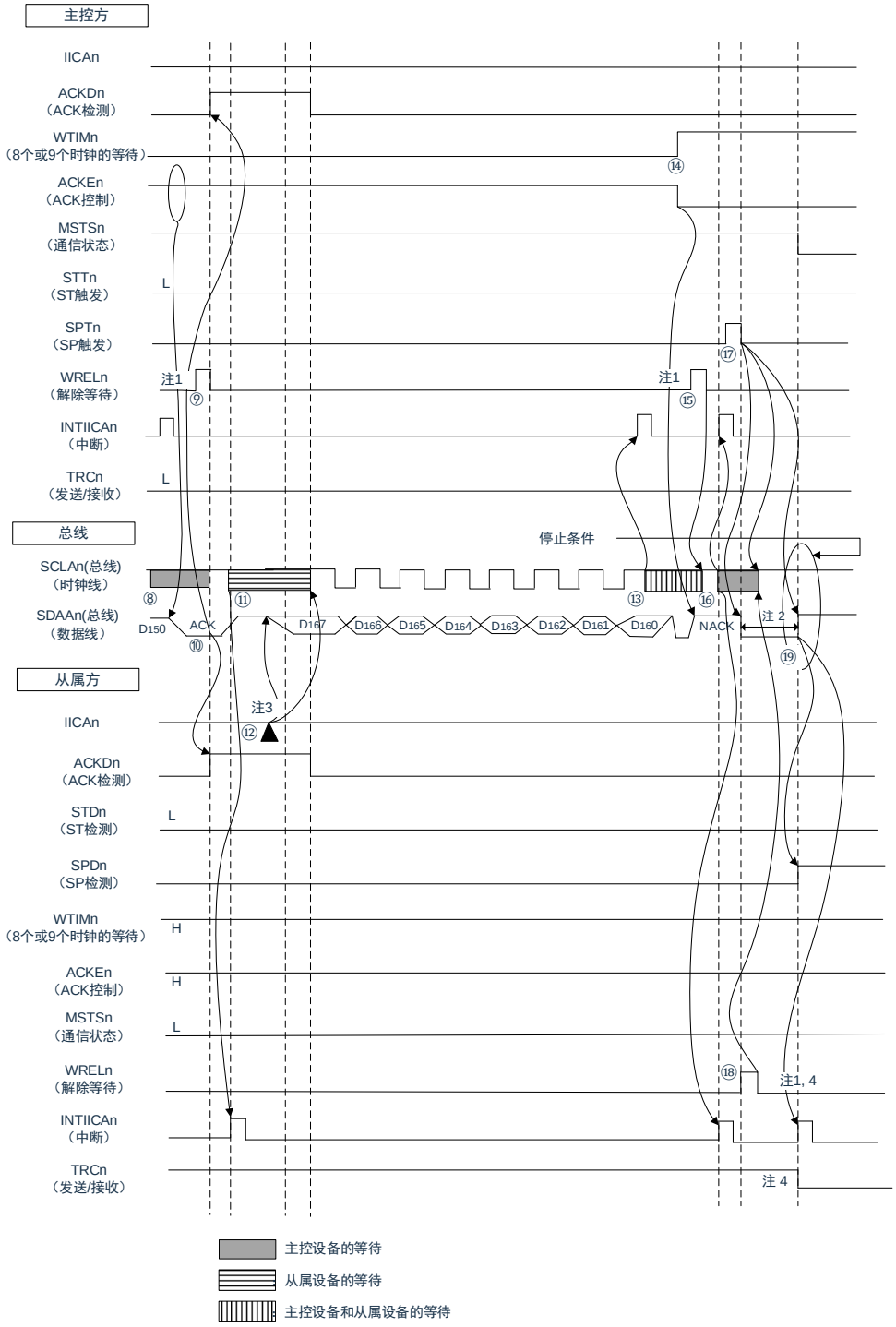
备注：

1. 图 18-32 的①~⑲是通过 I2C 总线进行数据通信的一系列运行步骤。
   图 18-32 的“(1)开始条件~地址~数据”说明步骤①~⑦。
   图 18-32 的“(2)地址~数据~数据”说明步骤③~⑫。
   图 18-32 的“(3)数据~数据~停止条件”说明步骤⑧~⑲。
2. n=0

图 18-32：从属设备→主控设备的通信例子

（主控设备：选择 8 个→9 个时钟的等待，从属设备：选择 9 个时钟的等待）(3/3)

(3) 数据~数据~停止条件



注 1：要解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

注 2：在发行停止条件后，从 SCLAn 引脚信号上升到生成停止条件的时间，在设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 18-32 的“(3)数据~数据~停止条件”的⑧~⑲的说明如下：

⑧主控方在第8个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）。因为主控方的ACKEn位为“0”，所以通过硬件给从属方发送ACK。

⑨主控方读接收数据，解除等待（WRELn=1）。

⑩从属方在第9个时钟的上升沿检测到ACK（ACKDn=1）。

⑪从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）

⑫如果从属方给IICA移位寄存器n（IICAn）写发送数据，就解除从属方的等待，开始从属方到主控方的数据传送。

⑬主控方在第8个时钟的下降沿产生中断（INTIICAn：传送结束中断），并且进入等待状态（SCLAn=0）。因为进行ACK控制（ACKEn=1），所以此阶段的总线数据线变为低电平（SDAAn=0）。

⑭主控方设定为NACK应答（ACKEn=0），并且将等待时序改为第9个时钟（WTIMn=1）。如果主控方解除等待（WRELn=1），从属方就在第9个时钟的上升沿检测到NACK（ACKDn=0）。

⑮主控方和从属方在第9个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。

⑯如果主控方发行停止条件（SPTn=1），就清除总线数据线（SDAAn=0），并且解除主控方的等待。此后，主控方处于待机状态，直到将总线时钟线置位（SCLAn=1）为止。

⑰从属方在确认NACK后停止发送，为了结束通信，解除等待（WRELn=1）。如果解除从属方的等待，就将总线时钟线置位（SCLAn=1）。

⑱如果主控方确认到总线时钟线被置位（SCLAn=1），就在经过停止条件准备时间后将总线数据线置位

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19章 IrDA

IrDA 通过与通用串行通信单元（SCI）合作，实现发送和接收符合 IrDA（InfraredDataAssociation（红外线数据协会））1.0 协议的 IrDA 通信波形。

19.1 IrDA的功能

如果通过 IRCR 寄存器的 IRE 位将 IrDA 功能置为有效，通用串行通信单元 1（UART1）的 TxD1 信号和 RxD1 信号就能对符合 IrDA1.0 协议的波形进行编码或者解码（IrTxD/IrRxD 引脚），之后通过连接红外线发送/接收的发送器或者接收器，实现支持 IrDA1.0 协议的红外线发送和接收。

在支持 IrDA1.0 协议的系统中，以 9600bps 的传送速率开始通信后，可根据需要改变传送速率。IrDA 没有内置自动改变传送速率的功能，所以必须通过软件更改设定以改变传送速率。

在选择高速内部振荡器（F_{IH}=24、12、6、3MHz）时，能设定以下的波特率。

- 115.2kbps、57.6kbps、38.4kbps、19.2kbps、9600bps、2400bps

IrDA 与 SCI 的合作示意框图如图 19-1 所示。

图 19-1：IrDA 与 SCI 的合作示意框图

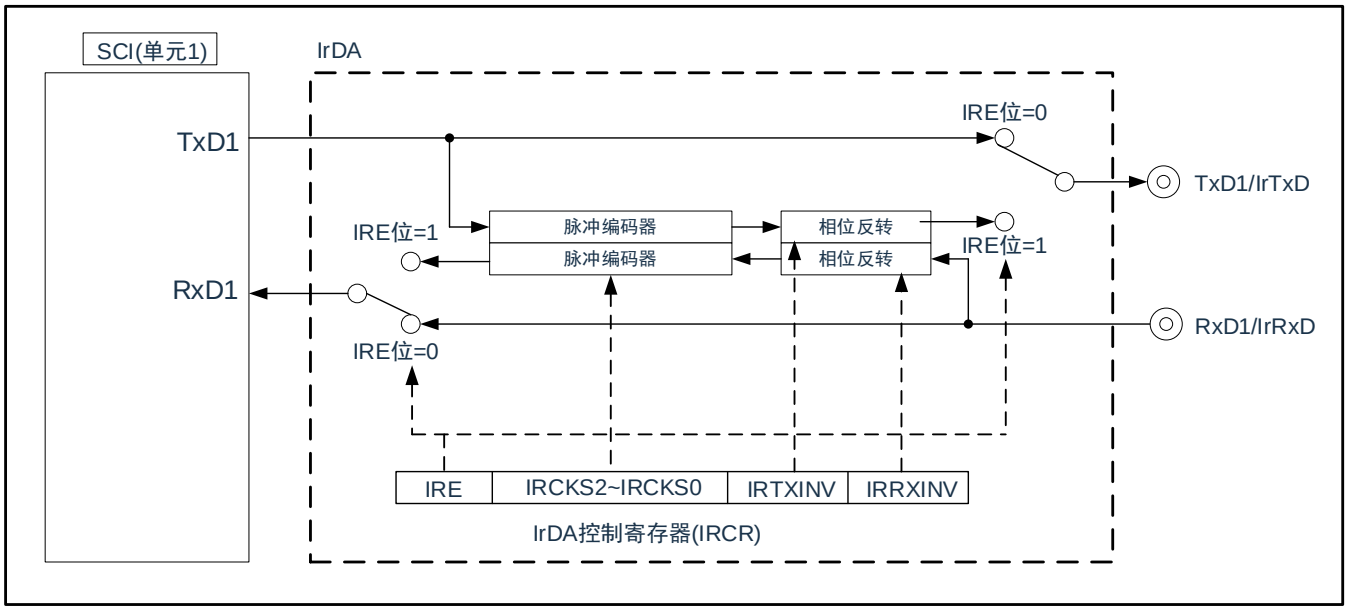


表 19-1：IrDA 的引脚结构

引脚名称	输入/输出	功能
IrTxD	输出	发送数据的输出引脚
IrRxD	输入	接收数据的输入引脚

19.2 控制IrDA的寄存器

通过以下寄存器控制 IrDA 功能。

- 外围允许寄存器 0（PER0）
- IrDA 控制寄存器（IRCR）

19.2.1 外围允许寄存器0（PER0）

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

要使用 IrDA 时，必须将 PER0.bit5（IRDAEN）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”。

图 19-2：外围允许寄存器 0（PER0）的格式

地址：40020420H 复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
PER0	RTCEN	IICAEN	IRDAEN	SCI2EN	SCI1EN	SCI0EN	TMAEN	TM80EN

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

注意：在设定 IrDA 时，必须先将 IRDAEN 位置“1”。当 IRDAEN 位为“0”时，忽视 IrDA 的控制寄存器的写操作，而且读取值全部为初始值。

19.2.2 IrDA控制寄存器（IRCR）

这是控制 IrDA 功能的寄存器。进行接收数据和发送数据的极性切换、IrDA 的时钟选择、以及串行输入/输出引脚功能（通常的串行功能和 IrDA 功能）切换的选择。通过 8 位存储器操作指令设定 IRCR 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 19-3：IrDA 控制寄存器（IRCR）的格式

地址：40048000H　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
IRCR	IRE	IRCKS2	IRCKS1	IRCKS0	IRTXINV	IRRXINV	0	0

IRE	IrDA的允许
0	串行输入/输出引脚用作通常的串行功能
1	串行输入/输出引脚用作IrDA功能

IRCKS2	IRCKS1	IRCKS0	IrDA的时钟选择
0	0	0	B3/16（B=位速率）
0	0	1	$F_{CLK}/2$
0	1	0	$F_{CLK}/4$
0	1	1	$F_{CLK}/8$
1	0	0	$F_{CLK}/16$
1	0	1	$F_{CLK}/32$
1	1	0	$F_{CLK}/64$
1	1	1	禁止设定

IRTXINV	IrTxD数据的极性切换
0	对发送数据进行IrTxD输出
1	反转发送数据后进行IrTxD输出

IRRXINV	IrRxD数据的极性切换
0	将IrRxD引脚的输入数据用作接收数据
1	将反转IrRxD引脚的输入数据后的数据用作接收数据

注意：

1. 必须将 bit1 和 bit0 置“0”。
2. 只有在 IRE 位为“0”时，才能设定 IRCKS[2:0]位、IRTXINV 位和 IRRXINV 位。

19.3 IrDA的运行

19.3.1 IrDA通信的操作步骤

(1) IrDA 通信的初始设定流程

按照以下步骤进行 IrDA 的初始设定。

(a) 将 PER0 寄存器的 IRDAEN 位置“1”。

(b) 设定 IRCR 寄存器。

(c) 设定 SCI 的相关寄存器（参照 UART 模式的设定步骤）。

(2) IrDA 通信的停止流程

(a) 通过设定端口寄存器和端口模式寄存器，设定 IrDA 通信停止后的 IrTxD 引脚状态。

备注：在通过步骤 3 进行 IrDA 复位时，IrTxD 引脚有可能因切换为通常的串行接口 UART 的数据输出，而改变输出状态。

- 从 IrTxD 引脚输出低电平的情况
  将端口寄存器置“0”。在进行此设定后，IrTxD 引脚立即固定为低电平。
- 从 IrTxD 引脚输出高电平的情况
  将端口寄存器置“1”。通过此设定，在步骤 3 的 IrDA 复位后，IrTxD 引脚立即固定为高电平。
- 将 IrTxD 引脚置为 Hi-Z 状态的情况
  将端口模式寄存器置“1”。在进行此设定后，IrTxD 引脚立即变为 Hi-Z 状态。

(b) 将 STm 寄存器（SCI 的相关寄存器）的 STm0 位和 STm1 位置“1”（停止 SCI 的通道 0 和通道 1 的运行）。

(c) 3.将 PER0 寄存器的 IRDAEN 位置“0”，进行 IrDA 复位。

不能在上述步骤以外的情况下将 STm 寄存器的 STm0 位和 STm1 位置“1”，或者将 IrDA 的 IRE 位置“0”。

(3) 发送 IrDA 帧错误时的步骤

在 IrDA 通信过程中发生帧错误时，为了设定为可接收后续数据的状态，必须进行以下设定。

(a) 将 SCI 的 STm 寄存器的 STm1 位置“1”（停止 SCI 的通道 1 的运行）。

(b) 将 SCI 的 SSm 寄存器的 SSm1 位置“1”（开始 SCI 的通道 1 的运行）。

备注：m：单元号（m=0）

有关 SCI 的帧错误处理，请参照“第 16 章 通用串行通信单元”。

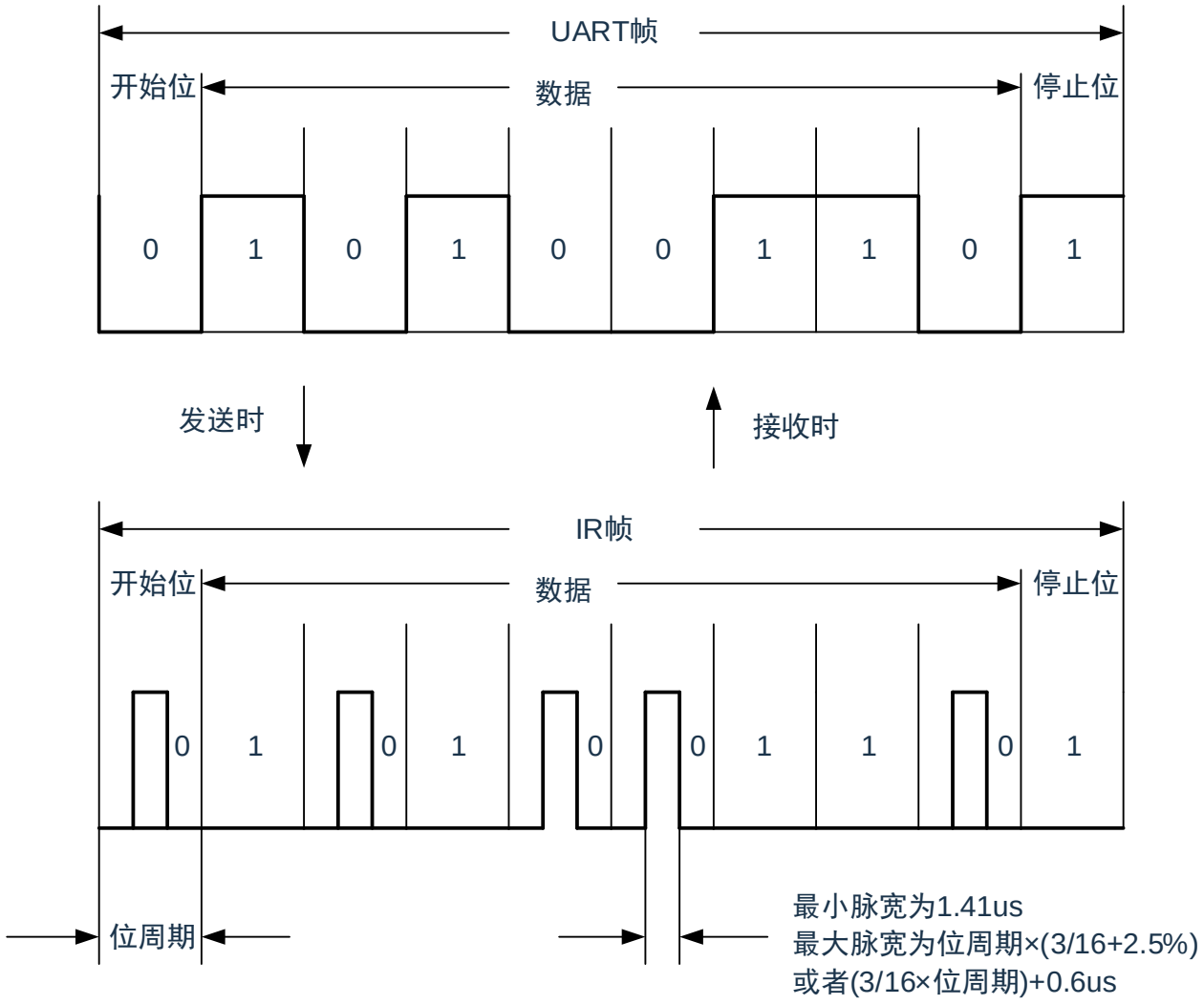
19.3.2 发送

在发送时，来自 SCI 的输出信号（UART 帧）通过 IrDA 转换为 IR 帧（参照图 19-4）。

在 IRTXINV 位为“0”并且串行数据为“0”时，输出位周期（1 位位宽期间）×3/16 的高电平脉冲（初始值）。另外，根据 IRCKS2~IRCKS0 位的设定值可更改高电平脉宽。按标准，规定高电平脉宽最小为 1.41us，最大为(3/16+2.5%)×位周期，或者(3/16×位周期)+0.6us。

在 CPU 或者外围硬件时钟（F_{CLK}）为 24MHz 时，能设定的最小高电平脉宽为 1.5us（满足上述规定的高电平脉宽不小于 1.41us 的条件）。另外，在串行数据为“1”时，不输出脉冲。

图 19-4：IrDA 的发送/接收运行图



19.3.3 接收

接收时，IR 帧的数据通过 IrDA 转换为 UART 帧后，输入到 SCI。在 IRRXINV 位为“0”，并且检测到高电平脉冲时，输出低电平数据。如果在 1 位期间内没有脉冲，就输出高电平数据。必须注意无法识别小于最小脉宽 1.41us 的脉冲。

19.3.4 高电平脉宽的选择

如果发送时的脉宽小于位速率×3/16，可适用的 IRCKS2~IRCKS0 位的设定（最小脉宽）和设定时的高电平脉宽如表 19-2 所示。

表 19-2：IRCKS2~IRCKS0 位的设定值

F_{CLK} [MHz]	项目	＜上段＞位速率[kbps] ＜下段＞位速率＞3/16[us]					
		2.4	9.6	19.2	38.4	57.6	115.2
		78.13	19.53	9.77	4.87	3.26	1.63
1	IRCKS2~IRCKS0	001	001	001	-注1	-注1	-注1
	高电平脉宽[us]	2.00	2.00	2.00	-注1	-注1	-注1
2	IRCKS2~IRCKS0	010	010	010	010	010	-注1
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	-注1
3	IRCKS2~IRCKS0	011	011	011	011	011	-注1
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	-注1
4	IRCKS2~IRCKS0	011	011	011	011	011	000注2
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
6	IRCKS2~IRCKS0	100	100	100	100	100	000注2
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
8	IRCKS2~IRCKS0	100	100	100	100	100	000注2
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
12	IRCKS2~IRCKS0	101	101	101	101	101	000注2
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50
16	IRCKS2~IRCKS0	101	101	101	101	101	000注2
	高电平脉宽[us]	2.00	2.00	2.00	2.00	2.00	1.50
24	IRCKS2~IRCKS0	110	110	110	110	110	000注2
	高电平脉宽[us]	2.67	2.67	2.67	2.67	2.67	1.50

注 1：“-”表示未达到通信标准。
注 2：脉宽不能小于位速率×3/16。

19.4 使用IrDA时的注意事项

1. 能通过外围允许寄存器设定允许或者禁止提供 IrDA 的运行时钟。初始状态为禁止提供时钟，因此无法存取寄存器。在设定寄存器前，必须通过外围允许寄存器设定为允许提供 IrDA 运行时钟的状态。
2. 睡眠模式中，IrDA 功能持续运行。
3. 在 IrDA 通信过程中，禁止使用 SCI 的初始化功能（SS 位=1）。
4. 只有在 IRE 位为“0”时，才能设定 IRCR 寄存器的 IRRXINV 位、IRTXINV 位和 IRCKS[2:0]位。

第20章 LCD控制器/驱动器

20.1 LCD控制器 /驱动器的功能

LCD 控制器 /驱动器的功能如下所示。

(1) 可选择 A 波形或者 B 波形。

(2) LCD 驱动电压生成电路能进行内部升压、电容分割和电阻分割的切换。

(3) 能通过自动读显示数据寄存器进行段信号和公共信号的自动输出。

(4) 能从升压电路运行时生成的 16 种基准电压（调整对比度）中选择。

(5) 能进行 LCD 闪烁显示。

在各显示模式中能显示的最大像素如表 20-1 所示。

表 20-1：最大显示像素

<table>
<tr><th>LCD 驱动器的驱动波形</th><th>LCD 驱动器的驱动电压生成电路</th><th>偏压法</th><th>时间片</th><th>最大显示像素</th></tr>
<tr><td rowspan="10">A 波形</td><td rowspan="6">电阻分割</td><td>-</td><td>静态</td><td>42(42 个段×1 个公共)</td></tr>
<tr><td rowspan="2">1/2</td><td>2</td><td>84(84 个段×2 个公共)</td></tr>
<tr><td>3</td><td rowspan="2">126(42 个段×3 个公共)</td></tr>
<tr><td rowspan="2">1/3</td><td>3</td></tr>
<tr><td>4</td><td>168(42 个段×4 个公共)</td></tr>
<tr><td>1/4</td><td>8</td><td>304(38 个段×8 个公共)</td></tr>
<tr><td rowspan="4">内部升压</td><td rowspan="2">1/3</td><td>3</td><td>126(42 个段×3 个公共)</td></tr>
<tr><td>4</td><td>168(42 个段×4 个公共)</td></tr>
<tr><td rowspan="2">1/4</td><td>6</td><td>240(40 个段×6 个公共)</td></tr>
<tr><td>8</td><td>304(38 个段×8 个公共)</td></tr>
<tr><td rowspan="2">电容分割</td><td rowspan="2">1/3</td><td>3</td><td>126(42 个段×3 个公共)</td></tr>
<tr><td>4</td><td rowspan="2">168(42 个段×4 个公共)</td></tr>
<tr><td rowspan="3">B 波形</td><td rowspan="2">电阻分割，内部升压</td><td>1/3</td><td>4</td></tr>
<tr><td>1/4</td><td>8</td><td>304(38 个段×8 个公共)</td></tr>
<tr><td>电容分割</td><td>1/3</td><td>4</td><td>168(42 个段×4 个公共)</td></tr>
</table>

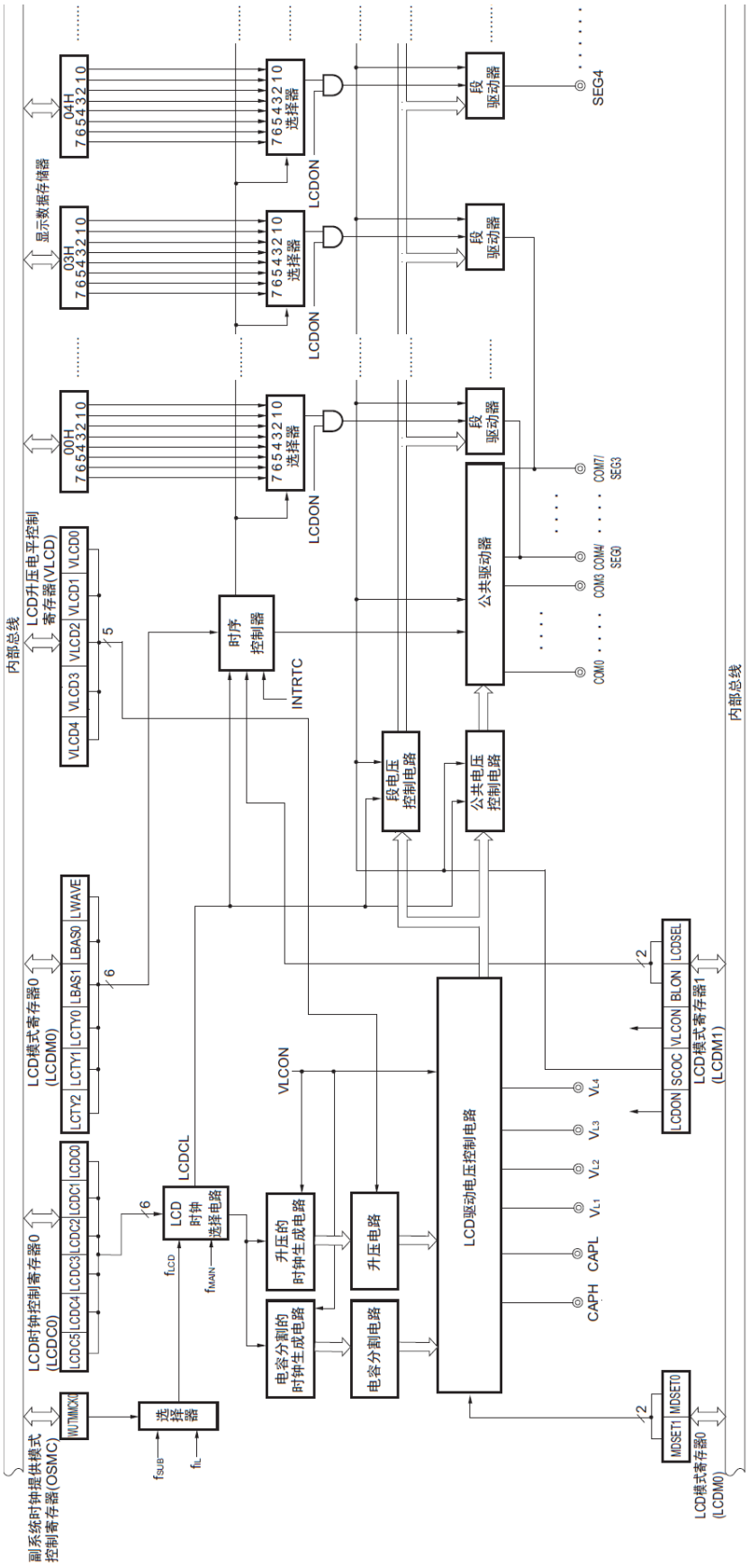
20.2 LCD控制器 /驱动器的结构

LCD控制器 /驱动器由以下硬件构成。

表 20-2：LCD 控制器/驱动器的结构

项目结构	结构
控制寄存器	LCD 模式寄存器 1（LCDM1）
	副系统时钟提供模式控制寄存器（OSMC）
	LCD 时钟控制寄存器 0 （LCDC0）
	LCD 升压电平控制寄存器（VLCD）
	LCD 输入切换控制寄存器（ISCLCD）
	LCD 端口功能寄存器(SEGEN0,SEGEN1,SEGEN2,SEGEN3)

图 20-1：LCD 控制器/驱动器的框图



20.3 控制 LCD控制器 /驱动器的寄存器

通过以下7种寄存器控制 LCD控制器 /驱动器。

- LCD模式寄存器 0（LCDM0）
- LCD模式寄存器 1（LCDM1）
- 副系统时钟提供模式控制寄存器（ OSMC）
- LCD时钟控制寄存器 0（LCDC0）
- LCD升压电平控制寄存器（ VLCD）
- LCD输入切换控制寄存器（ ISCLCD）
- LCD端口功能寄存器（SEGEN0, SEGEN1, SEGEN2, SEGEN3）

20.3.1 LCD模式寄存器 0（LCDM0）

这是设定 LCD运行的寄存器。

通过8位存储器操作指令设定 LCDM0寄存器。

在产生复位信号后，此寄存器的值变为 “00H”。

图 20-2：LCD模式寄存器 0（LCDM0）的格式

地址：0x40049040H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

MDSET1	MDSET0	LCD 驱动电压生成电路的选择
0	0	内部电阻分割方式
0	1	内部升压方式
1	0	电容分割方式
1	1	外部电阻分割方式

LWAVE	LCD 显示波形的选择
0	A 波形
1	B 波形

LDTY2	LDTY1	LDTY0	LCD 显示的时间片选择
0	0	0	静态
0	0	1	2 个时间片
0	1	0	3 个时间片
0	1	1	4 个时间片
1	0	0	6 个时间片
1	0	1	8 个时间片
上述以外			禁止设定

LBAS1	LBAS0	LCD 显示的偏压法选择
0	0	1/2 偏压法
0	1	1/3 偏压法
1	0	1/4 偏压法
1	1	禁止设定

注意：

1. 当 LCDM1 寄存器的 SCOC 位为 “1”时，不能改写 LCDM0 寄存器的值。
2. 当选择静态（LDTY2~ LDTY0=000B）时，必须将 LBAS1 位和 LBAS0 位置初始值（“00B”）。如果设定初始值以外的值，就不保证运行。
3. 只对应表 20-3 所示的显示波形、时间片、偏压法的组合设定。禁止表 20-3 所示以外的组合设定。

表 20-3：显示波形，时间片，偏压法和帧频的组合

显示模式			设定值						驱动电压的生成方式		
显示波形	时间片	偏压法	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0	外部电阻分割	内部分压	电容分割
A 波形	8	1/4	0	1	0	1	1	0	○（24~128Hz）	○（24~64Hz）	X
A 波形	6	1/4	0	1	0	0	1	0	X	○（32~86Hz）	X
A 波形	4	1/3	0	0	1	1	0	1	○（24~128Hz）	○（24~128Hz）	○（24~128Hz）
A 波形	3	1/3	0	0	1	0	0	1	○（32~128Hz）	○（32~128Hz）	○（32~128Hz）
A 波形	3	1/2	0	0	1	0	0	0	○（32~128Hz）	X	X
A 波形	2	1/2	0	0	0	1	0	0	○（24~128Hz）	X	X
A 波形	静态		0	0	0	0	0	0	○（24~128Hz）	X	X
B 波形	8	1/4	1	1	0	1	1	0	○（24~128Hz）	○（24~64Hz）	X
B 波形	4	1/3	1	0	1	1	0	1	○（24~128Hz）	○（24~128Hz）	○（24~128Hz）

备注：○：对应
X：不对应

20.3.2 LCD模式寄存器 1（LCDM1）

此寄存器允许或者禁止显示运行，允许或者停止升压电路和电容分割电路的运行以及设定显示数据区和低电压模式。

通过8位存储器操作指令设定 LCDM1寄存器。

在产生复位信号后，此寄存器的值变为 “00H”。

图 20-3：LCD模式寄存器 1（LCDM1）的格式

地址：0x40049041H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

SCOC	LCDON	LCD 显示的允许或禁止
0	0	将接地电平输出到段引脚或者公共引脚
0	1	
1	0	显示 OFF(段输出全部为非选择信号输出)
1	1	显示 ON

VLCON	升压电路或者电容分割电路的运行允许或者停止
0	停止升压电路或者电容分割电路的运行
1 注1	允许升压电路或者电容分割电路的运行

BLON	LCDSEL	显示数据区的控制
0 注2	0	显示 A 图形区（LCD 显示数据寄存器的低 4 位）的数据
0	1	显示 B 图形区（LCD 显示数据寄存器的高 4 位）的数据
1	0	交替显示 A 图形区和 B 图形区的数据（实时时钟 2（RTC2）的固定周期中断（INTRTC）时序对应的闪烁显示）
1	1	

LCDVLM 注3	升压引脚初始值的切换控制
0	V_{DD} 电压≥ 2.7V 的情况
1	V_{DD} 电压≤ 4.2V 的情况

注 1：在外部电阻分割模式中，禁止设定。

注 2：要选择 F_{IL} 作为 LCD 源时钟（F_{LCD}）时，必须将 BLON 位置“0”。

注 3：这是在使用升压电路时设定 VLx 引脚的初始状态并且缩短升压稳定时间的功能。

在升压开始时的 V_{DD} 电压大于等于 2.7V 的情况下，必须将 LCDVLM 位置“0”；在 V_{DD} 电压小于等于 4.2V 的情况下，必须将 LCDVLM 位置“1”。但是，当 2.7V≤V_{DD}≤4.2V 时，无论 LCDVLM 位为“0”或者 LCDVLM 位为“1”都能运行。

注意：

1. 在使用升压电路时不进行 LCD 显示的情况下，为了降低功耗，必须将 SCOC 位和 VLCON 位都置“0”，并且将 MDSET1 位和 MDSET0 位置“00B”。当 MDSET1 位和 MDSET0 位为“01B” 时，内部基准电压生成内部运行，因此会消耗功率。
2. 在设定外部电阻分割方式（LCDM0 的 MDSET1、MDSET0=00B）或者电容分割方式时（MDSET1、MDSET0=10B）时，必须将 LCDVLM 位置“0”。
3. 当 SCOC 位为“1” 时，不能改写 VLCON 位和 LCDVLM 位。

4. 当选择 8 个时间片的显示模式时，必须将 BLON 位和 LCDSEL 位置“0”。
5. 当使用内部升压方式时，必须在通过 VLCD 寄存器设定基准电压后（当以默认值使用基准电压时，在选择为内部升压方式（LCDM0 寄存器的 MDSET1、MDSET0=01B）后）等待基准电压准备时间（5ms(MIN.)），然后将 VLCON 位置“1”。

20.3.3 副系统时钟提供模式控制寄存器（OSMC）

OSMC 寄存器是通过停止不需要的时钟功能来降低功耗的寄存器。

如果将RTCLPC 位置“1”，就在STOP 模式或者CPU 以副系统时钟运行的HALT 模式中停止给实时时钟、15 位间隔定时器、时钟输出/ 蜂鸣器输出和LCD 控制器/ 驱动器以外的外围功能提供时钟，因此能降低功耗。

能通过OSMC 寄存器选择LCD 控制器/ 驱动器的运行时钟。

详细内容请参考“4.3.7 副系统时钟提供模式控制寄存器（OSMC）”。

20.3.4 LCD 时钟控制寄存器0 （LCDC0）

这是设定 LCD 时钟的寄存器。通过 LCD 时钟和时间片决定帧频。

通过 8 位存储器操作指令设定 LCDC0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-4：LCD时钟控制寄存器（LCDC0）的格式

地址：0x40049042H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
LCDC0	0	0	LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD 时钟（LCDCL）
0	0	0	1	0	0	$F_{SUB}/2^5$ or $F_{IL}/2^5$
0	0	0	1	0	1	$F_{SUB}/2^6$ or $F_{IL}/2^6$
0	0	0	1	1	0	$F_{SUB}/2^7$ or $F_{IL}/2^7$
0	0	0	1	1	1	$F_{SUB}/2^8$ or $F_{IL}/2^8$
0	0	1	0	0	0	$F_{SUB}/2^9$ or $F_{IL}/2^9$
0	0	1	0	0	1	$F_{SUB}/2^{10}$
0	1	0	0	1	1	$F_{MAIN}/2^{10}$
0	1	0	1	0	0	$F_{MAIN}/2^{11}$
0	1	0	1	0	1	$F_{MAIN}/2^{12}$
0	1	0	1	1	0	$F_{MAIN}/2^{13}$
0	1	0	1	1	1	$F_{MAIN}/2^{14}$
0	1	1	0	0	0	$F_{MAIN}/2^{15}$
0	1	1	0	0	1	$F_{MAIN}/2^{16}$
0	1	1	0	1	0	$F_{MAIN}/2^{17}$
0	1	1	0	1	1	$F_{MAIN}/2^{18}$
1	0	1	0	1	1	$F_{MAIN}/2^{19}$
上述以外						禁止设定

注意：

2. 在 LCDM1 寄存器的 SCOC 位为“1” 时，不能设定 LCDC0 寄存器。
3. 必须将 bit6 和 bit7 置“0”。
4. 当设定为内部升压方式或者电容分割方式时，必须对 LCD 时钟（LCDCL）进行以下的设定。
   - 当选择 F_{SUB} 时，不能超过 512Hz。
   - 当选择 F_{IL} 时，不能超过 235Hz。

   详细内容请参照“表 20-3：显示波形，时间片，偏压法和帧频的组合”。

备注：F_{MAIN}：主系统时钟频率

F_{IL}：低速内部振荡器时钟频率

F_{SUB}：副系统时钟频率

20.3.5 LCD 升压电平控制寄存器（VLCD）

这是选择升压电路运行时生成的基准电压（调整对比度）的寄存器。能选择 16 种基准电压。

通过 8 位存储器操作指令设定 VLCD 寄存器。

在产生复位信号后，此寄存器的值变为“04H”。

图 20-5：LCD 升压电平控制寄存器（VLCD）的格式

地址：0x40049043H　　复位后：04H　　R/W

符号	7	6	5	4	3	2	1	0
VLCD	0	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	基准电压的选择（调整对比度）	VL_4 电压	
						1/3 偏压法	1/4 偏压法
0	0	0	1	1	1.00V	3.00V	4.00V
0	0	1	0	0	1.05V	3.15V	4.20V
0	0	1	0	1	1.10V	3.30V	4.40V
0	0	1	1	0	1.15V	3.45V	4.60V
0	0	1	1	1	1.20V	3.60V	4.80V
0	1	0	0	0	1.25V	3.75V	5.00V
0	1	0	0	1	1.30V	3.90V	5.20V
0	1	0	1	0	1.35V	4.05V	禁止设定
0	1	0	1	1	1.40V	4.20V	禁止设定
0	1	1	0	0	1.45V	4.35V	禁止设定
0	1	1	0	1	1.50V	4.50V	禁止设定
0	1	1	1	0	1.55V	4.65V	禁止设定
0	1	1	1	1	1.60V	4.80V	禁止设定
1	0	0	0	0	1.65V	4.95V	禁止设定
1	0	0	0	1	1.70V	5.10V	禁止设定
1	0	0	1	0	1.75V	5.25V	禁止设定
上述以外						禁止设定	

注意：

1. 只有在升压电路运行时，VLCD 寄存器的设定才有效。
2. 必须将 bit5 ~ 7 置“0”。
3. 必须在停止升压电路的运行（VLCON=0）后更改 VLCD 寄存器的值。
4. 当使用内部升压方式时，必须在通过 VLCD 寄存器设定基准电压后（当以默认值使用基准电压时，在选择为内部升压方式（LCDM0 寄存器的 MDSET1、MDSET0=01B）后）等待基准电压准备时间（5ms(MIN.)），然后将 VLCON 位置“1”。
5. 在外部电阻分割方式或者电容分割方式的情况下，VLCD 寄存器必须为初始值“04H”。

20.3.6 LCD 输入切换控制寄存器（ISCLCD）

为了在设定 CAPL、CAPH、VLx 引脚作为 LCD 功能运行的期间防止贯通电流的流入，需要禁止施密特触发缓冲器的输入。

通过 8 位存储器操作指令设定 ISCLCD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-6：LCD 输入切换控制寄存器（ISCLCD）

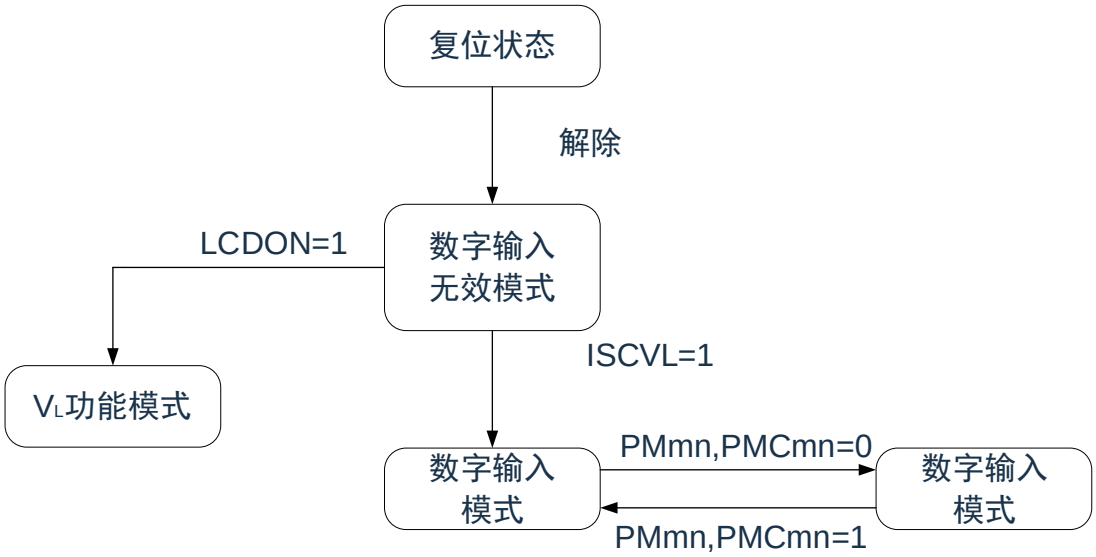
地址：0x40040804H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL	ISCCAP

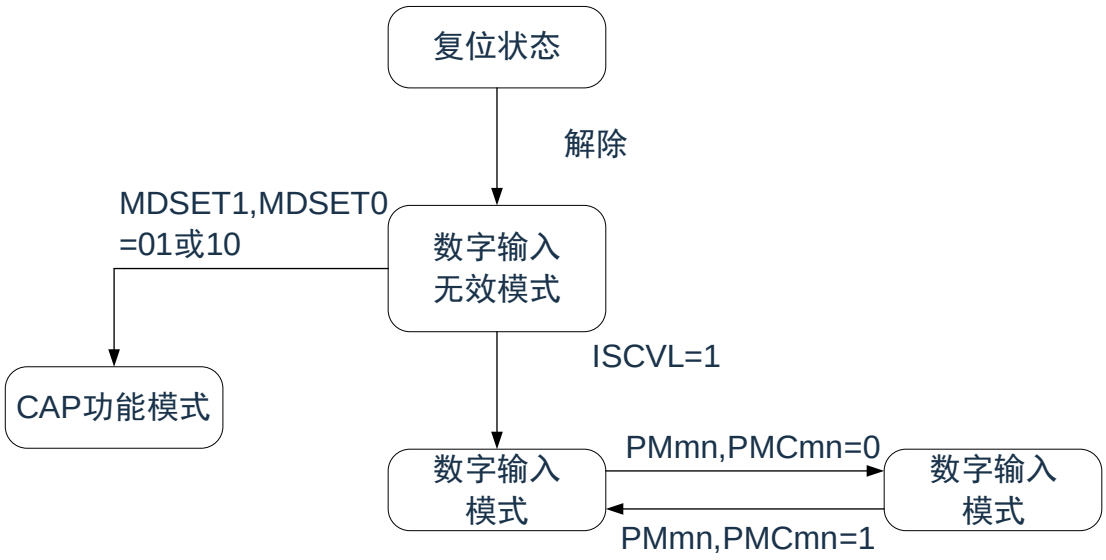
ISCVL	VLx 引脚的施密特触发缓冲器的控制
0	输入无效
1	输入有效

ISCCAP	CAPL 和 CAPH 引脚的施密特触发缓冲器的控制
0	输入无效
1	输入有效

VLx 引脚功能的状态转移如下所示。



CAPx 引脚功能的状态转移如下所示。



20.3.7 LCD 端口功能寄存器

这是设定将 Pmn 引脚用作端口还是用作 LCD 输出的寄存器，

通过 16 位存储器操作指令设定 SEGEN0，SEGEN1，SEGEN2 和 SEGEN3 寄存器。

在产生复位信号后，这些寄存器的值变为“0000H”。

图 20-7：LCD 端口功能寄存器的格式

地址：0x40040806H　复位后：0000H　R/W

符号	15	14	13	12	11	10	9	8
SEGEN0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	COMEN7/ SEGEN3	COMEN6/ SEGEN2	COMEN5/ SEGEN1	COMEN4/ SEGEN0	COMEN3	COMEN2	COMEN1	COMEN0

地址：0x40040808H　复位后：0000H　R/W

符号	15	14	13	12	11	10	9	8
SEGEN1	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
	7	6	5	4	3	2	1	0
	SEGEN7	SEGEN6	SEGEN5	SEGEN4	0	0	0	0

地址：0x4004080AH　复位后：0000H　R/W

符号	15	14	13	12	11	10	9	8
SEGEN2	SEGEN31	SEGEN30	SEGEN29	SEGEN28	SEGEN27	SEGEN26	SEGEN25	SEGEN24
	7	6	5	4	3	2	1	0
	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16

地址：0x4004080CH　复位后：0000H　R/W

符号	15	14	13	12	11	10	9	8
SEGEN3	0	0	0	0	0	0	SEGEN41	SEGEN40
	7	6	5	4	3	2	1	0
	SEGEN39	SEGEN38	SEGEN37	SEGEN36	SEGEN35	SEGEN34	SEGEN33	SEGEN32

COMENm	Pmn 引脚的功能指定
0	用作端口
1	用作 COM 输出

SEGENm	Pmn 引脚的功能指定
0	用作端口
1	用作 SEG 输出

备注：当用作 COM/SEG 输出（SEGENxx=1）时，必须将 PUm 寄存器的 PUmn 位、POMm 寄存器的 POMmn 位、PIMm 寄存器的 PIMmn 位和 Pm 寄存器的 Pmn 位都置“0”。

m=(1-3);

20.4 LCD显示数据寄存器

LCD 显示数据寄存器的映像如表 20-4 所示。能通过更改 LCD 显示数据寄存器的内容，更改 LCD 的显示内容。

表 20-4：LCD 显示数据寄存器的内容和段输出、公共输出的关系(1/2)

(a) 6个时间片和 8个时间片以外（静态、2个时间片、3个时间片、4个时间片）

寄存器名	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
		COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0
SEG0	0x40049000H	SEG0(B 图形区)				SEG0(A 图形区)			
SEG1	0x40049001H	SEG1(B 图形区)				SEG1(A 图形区)			
SEG2	0x40049002H	SEG2(B 图形区)				SEG2(A 图形区)			
SEG3	0x40049003H	SEG3(B 图形区)				SEG3(A 图形区)			
SEG4	0x40049004H	SEG4(B 图形区)				SEG4(A 图形区)			
SEG5	0x40049005H	SEG5(B 图形区)				SEG5(A 图形区)			
SEG6	0x40049006H	SEG6(B 图形区)				SEG6(A 图形区)			
SEG7	0x40049007H	SEG7(B 图形区)				SEG7(A 图形区)			
SEG8	0x40049008H	SEG8(B 图形区)				SEG8(A 图形区)			
SEG9	0x40049009H	SEG9(B 图形区)				SEG9(A 图形区)			
SEG10	0x4004900AH	SEG10(B 图形区)				SEG10(A 图形区)			
SEG11	0x4004900BH	SEG11(B 图形区)				SEG11(A 图形区)			
SEG12	0x4004900CH	SEG12(B 图形区)				SEG12(A 图形区)			
SEG13	0x4004900DH	SEG13(B 图形区)				SEG13(A 图形区)			
SEG14	0x4004900EH	SEG14(B 图形区)				SEG14(A 图形区)			
SEG15	0x4004900FH	SEG15(B 图形区)				SEG15(A 图形区)			
SEG16	0x40049010H	SEG16(B 图形区)				SEG16(A 图形区)			
SEG17	0x40049011H	SEG17(B 图形区)				SEG17(A 图形区)			
SEG18	0x40049012H	SEG18(B 图形区)				SEG18(A 图形区)			
SEG19	0x40049013H	SEG19(B 图形区)				SEG19(A 图形区)			
SEG20	0x40049014H	SEG20(B 图形区)				SEG20(A 图形区)			
SEG21	0x40049015H	SEG21(B 图形区)				SEG21(A 图形区)			
SEG22	0x40049016H	SEG22(B 图形区)				SEG22(A 图形区)			
SEG23	0x40049017H	SEG23(B 图形区)				SEG23(A 图形区)			
SEG24	0x40049018H	SEG24(B 图形区)				SEG24(A 图形区)			
SEG25	0x40049019H	SEG25(B 图形区)				SEG25(A 图形区)			
SEG26	0x4004901AH	SEG26(B 图形区)				SEG26(A 图形区)			
SEG27	0x4004901BH	SEG27(B 图形区)				SEG27(A 图形区)			
SEG32	0x40049020H	SEG32(B 图形区)				SEG32(A 图形区)			
SEG33	0x40049021H	SEG33(B 图形区)				SEG33(A 图形区)			
SEG34	0x40049022H	SEG34(B 图形区)				SEG34(A 图形区)			
SEG35	0x40049023H	SEG35(B 图形区)				SEG35(A 图形区)			
SEG36	0x40049024H	SEG36(B 图形区)				SEG36(A 图形区)			
SEG37	0x40049025H	SEG37(B 图形区)				SEG37(A 图形区)			
SEG38	0x40049026H	SEG38(B 图形区)				SEG38(A 图形区)			
SEG39	0x40049027H	SEG39(B 图形区)				SEG39(A 图形区)			

表 20-4：LCD 显示数据寄存器的内容和段输出、公共输出的关系(2/2)

(b) 6个时间片和8个时间片

寄存器名	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
SEG0	0x40049000H	SEG0 注							
SEG1	0x40049001H	SEG1 注							
SEG2	0x40049002H	SEG2 注							
SEG3	0x40049003H	SEG3 注							
SEG4	0x40049004H	SEG4							
SEG5	0x40049005H	SEG5							
SEG6	0x40049006H	SEG6							
SEG7	0x40049007H	SEG7							
SEG8	0x40049008H	SEG8							
SEG9	0x40049009H	SEG9							
SEG10	0x4004900AH	SEG10							
SEG11	0x4004900BH	SEG11							
SEG12	0x4004900CH	SEG12							
SEG13	0x4004560DH	SEG13							
SEG14	0x4004900EH	SEG14							
SEG15	0x4004900FH	SEG15							
SEG16	0x40049010H	SEG16							
SEG17	0x40049011H	SEG17							
SEG18	0x40049012H	SEG18							
SEG19	0x40049013H	SEG19							
SEG20	0x40049014H	SEG20							
SEG21	0x40049015H	SEG21							
SEG22	0x40049016H	SEG22							
SEG23	0x40049017H	SEG23							
SEG24	0x40049018H	SEG24							
SEG25	0x40049019H	SEG25							
SEG26	0x4004901AH	SEG26							
SEG27	0x4004901BH	SEG27							
SEG32	0x40049020H	SEG32							
SEG33	0x40049021H	SEG33							
SEG34	0x40049022H	SEG34							
SEG35	0x40049023H	SEG35							
SEG36	0x40049024H	SEG36							
SEG37	0x40049025H	SEG37							
SEG38	0x40049026H	SEG38							
SEG39	0x40049027H	SEG39							

注：COM4 ~ COM7 引脚和 SEG0 ~ SEG3 引脚复用。

当用于静态、2 个时间片、3 个时间片或者 4 个时间片时，LCD 显示数据寄存器的各地址的低 4 位为 A 图形区，高 4 位为 B 图形区。

A 图形区的数据和 COM 信号的对应为 bit0->COM0、bit1->COM1、bit2->COM2、bit3->COM3。

B 图形区的数据和 COM 信号的对应为 bit4-＞COM0、bit5-＞COM1、bit6-＞COM2、bit7-＞COM3。

当 BLON 位和 LCDSEL 位都为“0” 时，LCD 显示屏显示 A 图形区的数据；当 BLON 位为“0” 并且 LCDSEL 位为“1” 时， LCD 显示屏显示 B 图形区的数据。

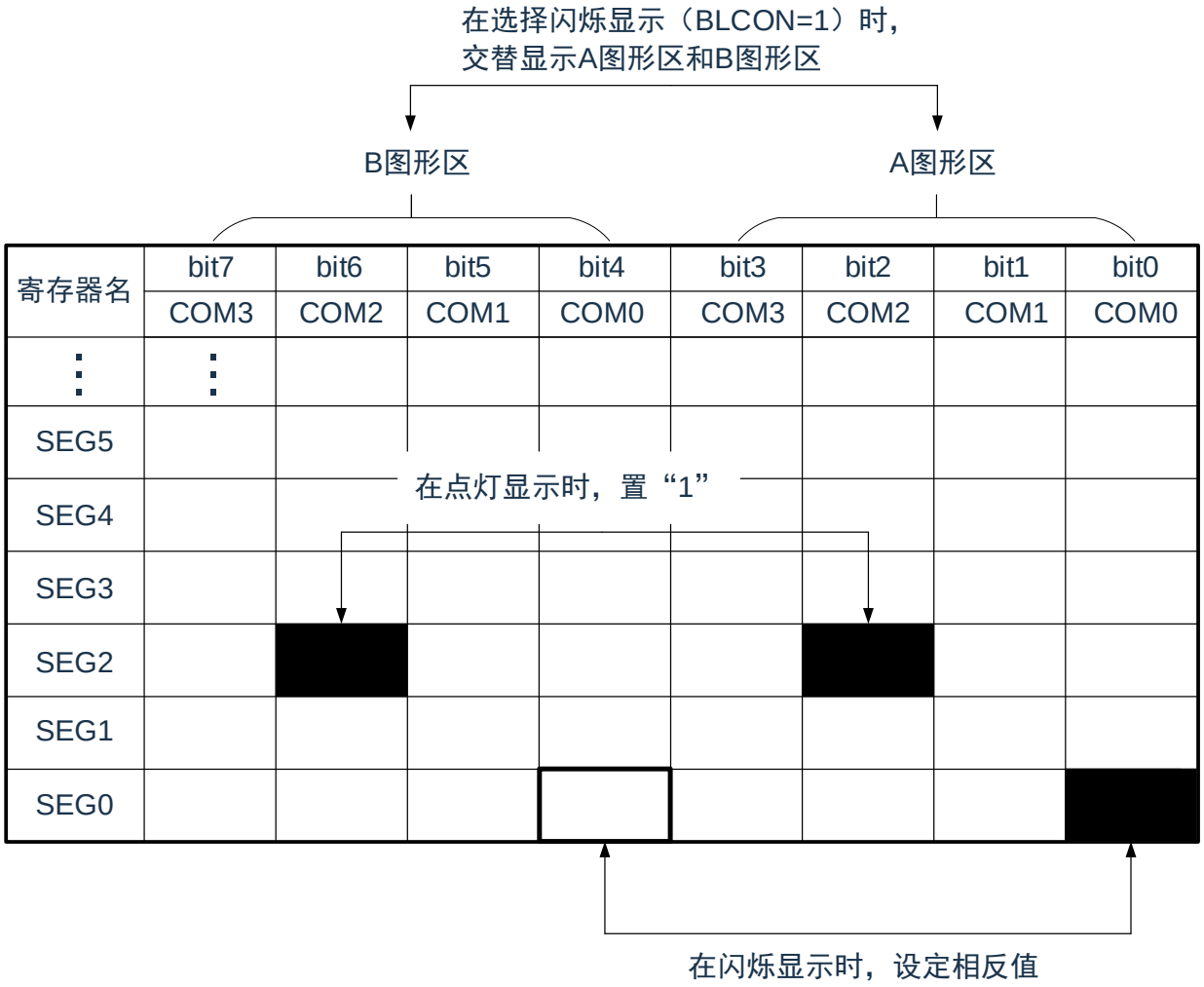
20.5 LCD显示寄存器的选择

当用于静态、2 个时间片、3 个时间片或者 4 个时间片时，能通过设定 BLON 位和 LCDSEL 位，选择以下 3 种 LCD 显示数据寄存器。

- 显示 A 图形区（LCD 显示数据寄存器的低 4 位）的数据。
- 显示 B 图形区（LCD 显示数据寄存器的高 4 位）的数据。
- 交替显示 A 图形区和 B 图形区的数据（实时时钟（RTC）的固定周期中断时序对应的闪烁显示）。

注意：在使用 6 个或者 8 个时间片时，不能选择 LCD 显示数据寄存器（A 图形、B 图形或者闪烁显示）。

图 20-8：图形切换显示时的 LCD 显示寄存器的设定例子



20.5.1 A图形区和 B图形区的数据显示

当 BLON 位和 LCDSEL 位都为 “0”时，将 A 图形区（LCD 显示数据寄存器的低 4 位）的数据作为 LCD 显示寄存器进行输出。

当 BLON 位为“0”并且 LCDSEL 位为“1”时，将 B 图形区（LCD 显示数据寄存器的高 4 位）的数据作为 LCD 显示寄存器进行输出。

有关显示区请参照“20.4 LCD 显示数据寄存器”。

20.5.2 闪烁显示（ A图形区和 B图形区的数据的交替显示）

当 BLON 位为 “1”时，对应实时时钟（RTC）的固定周期中断（ INTRTC）时序，进行 A 图形区和 B 图形区的数据交替显示。有关 RTC 的固定周期中断（ INTRTC、只限于 0.5s 的设定）时序的设定，请参照 “第 7 章 实时时钟”。

当 LCD 闪烁显示时，必须给与 A 图形区的位对应的 B 图形区的位设定反相值（ ex. 将 SEG0 的 bit0 置 “1”，在闪烁显示时将 SEG0 的 bit4 置 “0”）；当 LCD 不闪烁显示时，必须设定相同值（ ex. 将 SEG2 的 bit2 置 “1”，在点灯显示时将 SEG2 的 bit6 置 “1”）。

有关显示区请参照“20.4 LCD 显示数据寄存器”。

显示的切换时序如下所示。

图 20-9：从 A 图形显示到闪烁显示的切换运行

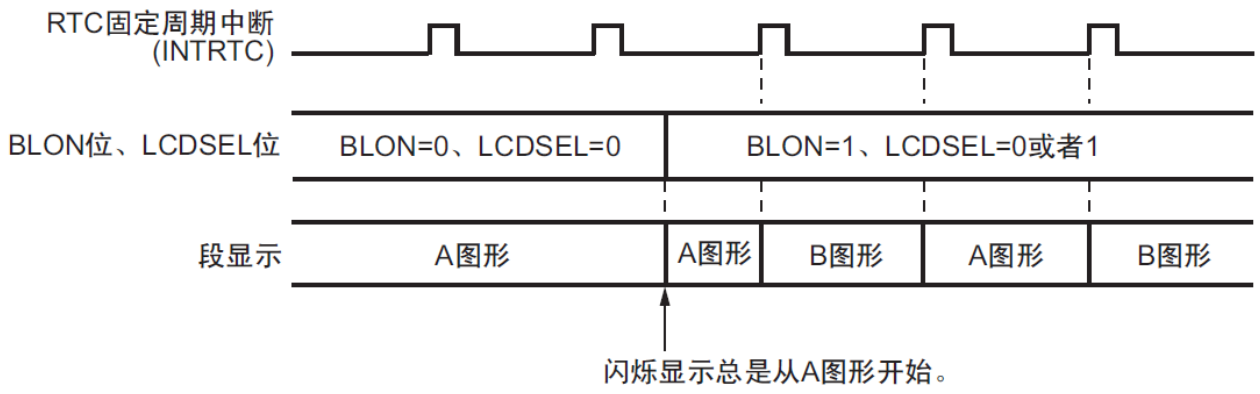
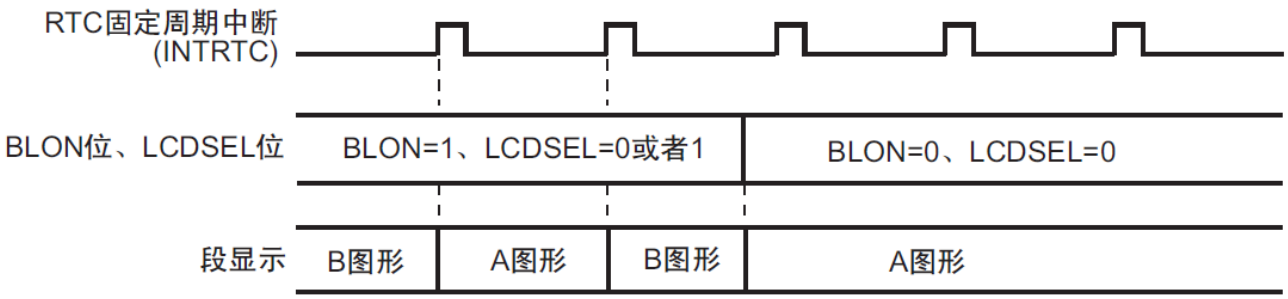


图 20-10：从闪烁显示到 A 图形显示的切换运行



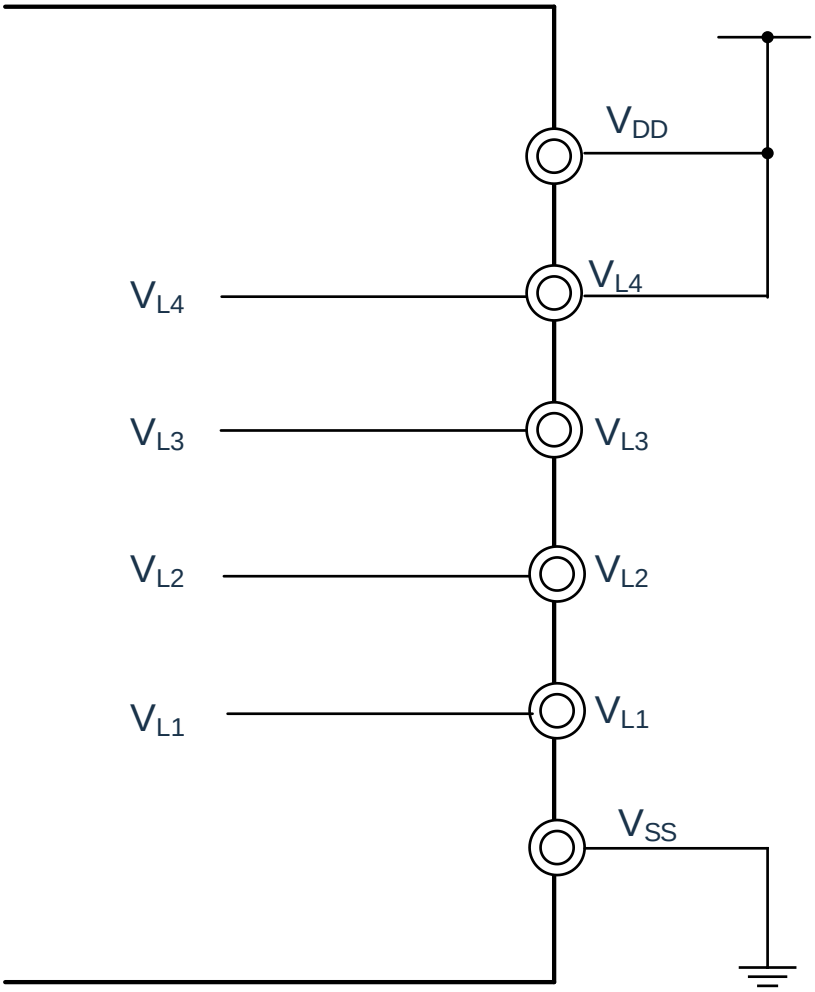
20.6 LCD 驱动电压V_{L1}、V_{L2}、V_{L3}、V_{L4} 的提供

能从内部电阻分割方式，外部电阻分割方式、内部升压方式和电容分割方式中选择 LCD 驱动电源的生成方法。

20.6.1 内部电阻分割方式

芯片内置用于 LCD 驱动电源的内部电阻分压电路，连接的方式如图 20-11 所示。

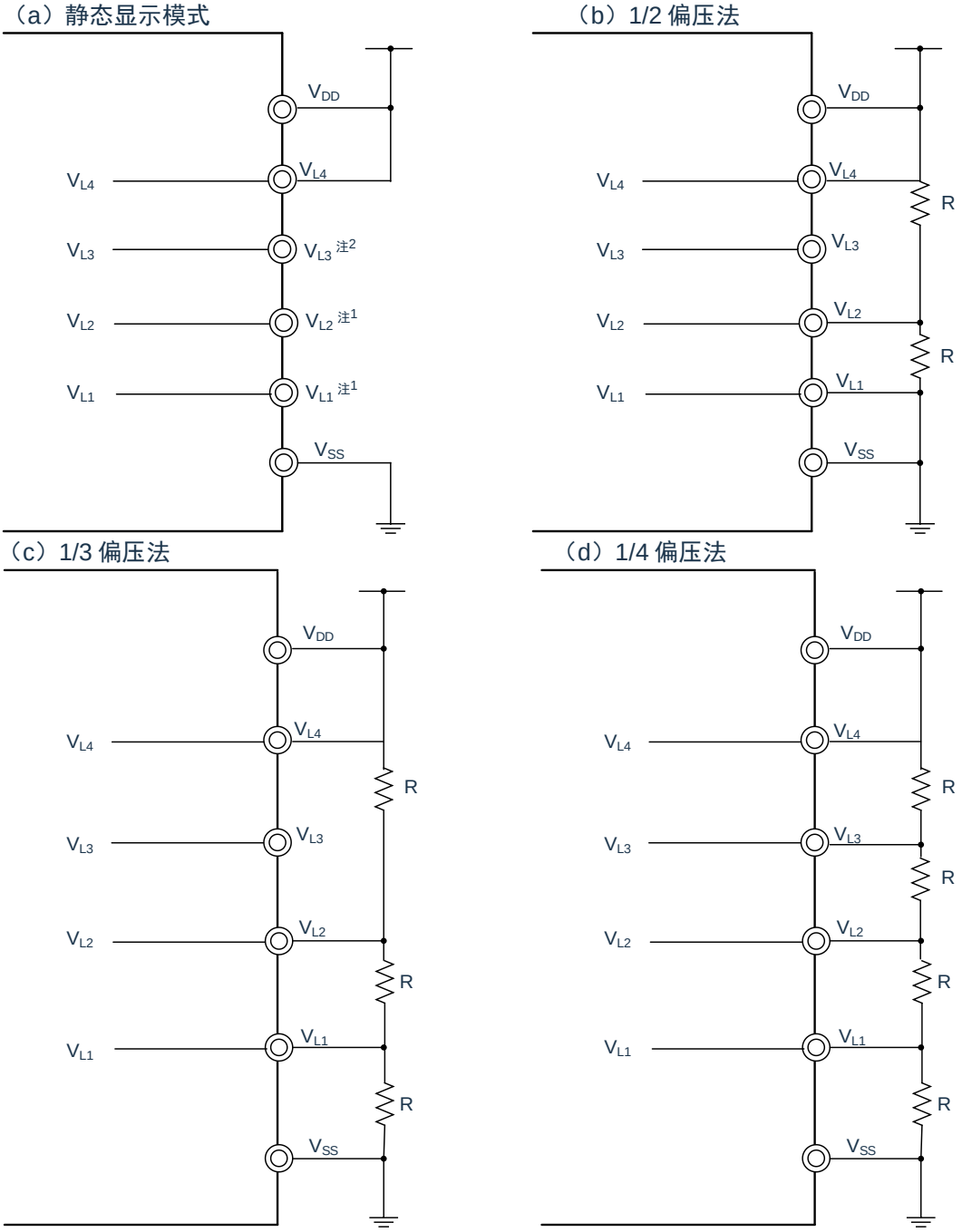
图 20-11：LCD 驱动电源的连接例子（内部电阻分割方式）



20.6.2 外部电阻分割方式

根据各偏压法进行 LCD 驱动电压连接的例子如图 20-12 所示。

图 20-12：LCD 驱动电源的连接例子（外部电阻分割方式）



注 1：必须将 V_{L1} 和 V_{L2} 连接 GND 或者设置为开路。

注 2：V_{L3} 可用作端口。

注意：用于外部电阻分解的电阻 R 的参考值为 10kΩ ~ 1MΩ。要稳定 V_{L1} ~ V_{L4} 引脚的电位时，必须根据需要，在 V_{L1} ~V_{L4} 引脚和 GND 之间连接电容器，电容器的参考值大约为 0.22uF，取决于使用的 LCD 显示屏、段引脚数、公共引脚数、帧频和使用环境。必须在根据系统进行充分评估的基础上调整和决定电容值。

20.6.3 内部升压方式

芯片内置用于 LCD 驱动电源的内部升压电路。通过外接内部升压电路的电容器（0.47uF±30%），生成 LCD 驱动电压。内部升压方式只能使用 1/3 偏压法或者 1/4 偏压法。

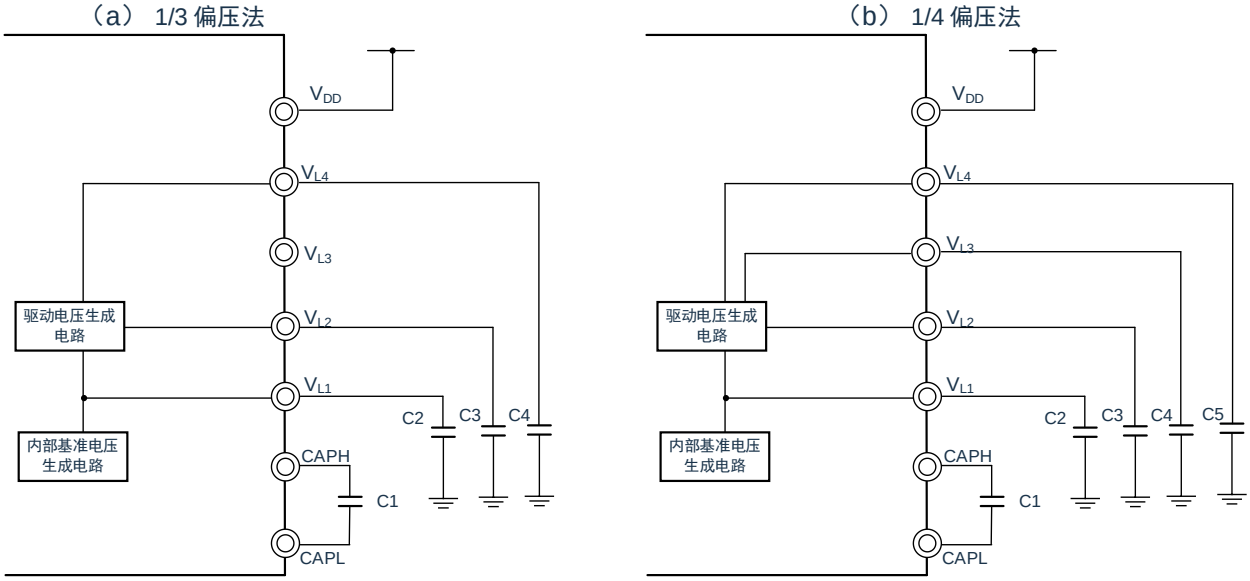
内部升压方式的 LCD 驱动电压和器件本身不是同一个电源，因此与 V_{DD} 的变化无关，能提供固定的电压。

能通过设定 LCD 升压控制寄存器（V_{LCD}）来调整对比度。

表 20-5：LCD驱动电压（内部升压方式）

显示模式 / LCD 驱动电源引脚	1/3 偏压法	1/4 偏压法
V_{L4}	3×V_{L1}	4×V_{L1}
V_{L3}	-	3×V_{L1}
V_{L2}	2×V_{L1}	2×V_{L1}
V_{L1}	LCD 基准电压	LCD 基准电压

图 20-13：LCD 驱动电源的连接例子（内部升压方式）



备注：

1. 必须尽量使用漏电流小的电容器。
2. C1 必须是无极性的电容器。

20.6.4 电容分割方式

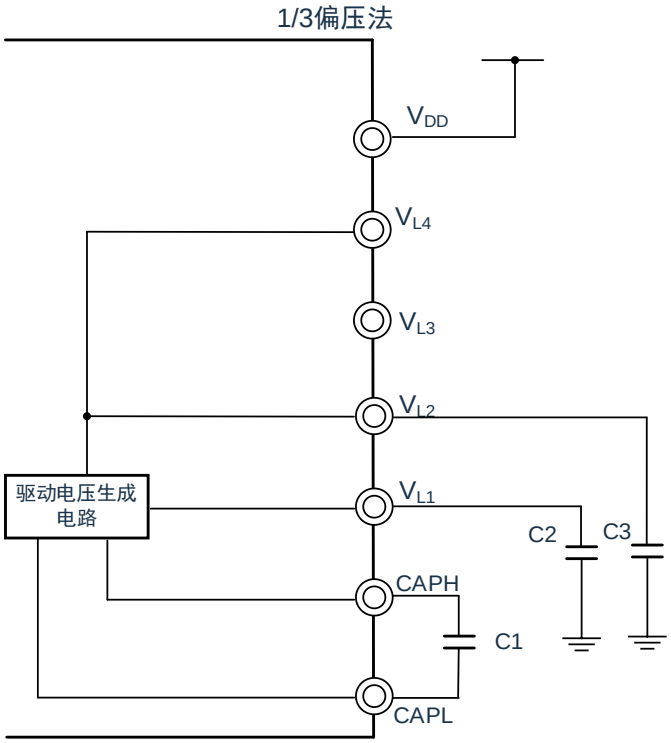
芯片内置用于 LCD 驱动电源的电容分割电路。通过外接电容分割电路的电容器（0.47uF±30%），生成 LCD 驱动电压。电容分割方式只能使用 1/3 偏压法。

和外部电阻分割方式不同，电容分割方式没有电流流过，因此能减小消费电流。

表 20-6：LCD驱动电压（电容分割方式）

显示模式 / LCD 驱动电源引脚	1/3 偏压法
VL4	V_{DD}
VL3	-
VL2	2/3×V_{L4}
VL1	1/3×V_{L4}

图 20-14：LCD 驱动电源的连接例子（电容分割方式）



备注：

1. 必须尽量使用漏电流小的电容器。
2. C1 必须是无极性的电容器。

第21章 增强型DMA

21.1 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 21-1 所示。

表 21-1：DMA 的规格(1/2)

项目		规格
启动源		最大24个启动源
可分配的控制数据		24组
可传送的地址空间	地址空间	全地址范围空间
	源	全地址范围空间可选
	目标	全地址范围空间可选
最大传送次数	正常模式	65535次
	重复模式	65535次
最大传送块大小	正常模式（8位传送）	65535字节
	正常模式（16位传送）	131070字节
	正常模式（32位传送）	262140字节
	重复模式	65535字节
传送单位		8位/16位/32位
传送模式	正常模式	在进行DMACTj寄存器从“1”变为“0”的传送后结束。
	重复模式	在DMACTj寄存器从“1”变为“0”的传送结束后，对重复区的地址进行初始化，在将DMRLDj寄存器的值重新加载到DMACTj寄存器后继续传送。
地址控制	正常模式	固定或者递增
	重复模式	固定或者递增非重复区的地址。
启动源的优先级		参照“表 21-5：DMA 启动源和向量地址“。

表 21-1：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

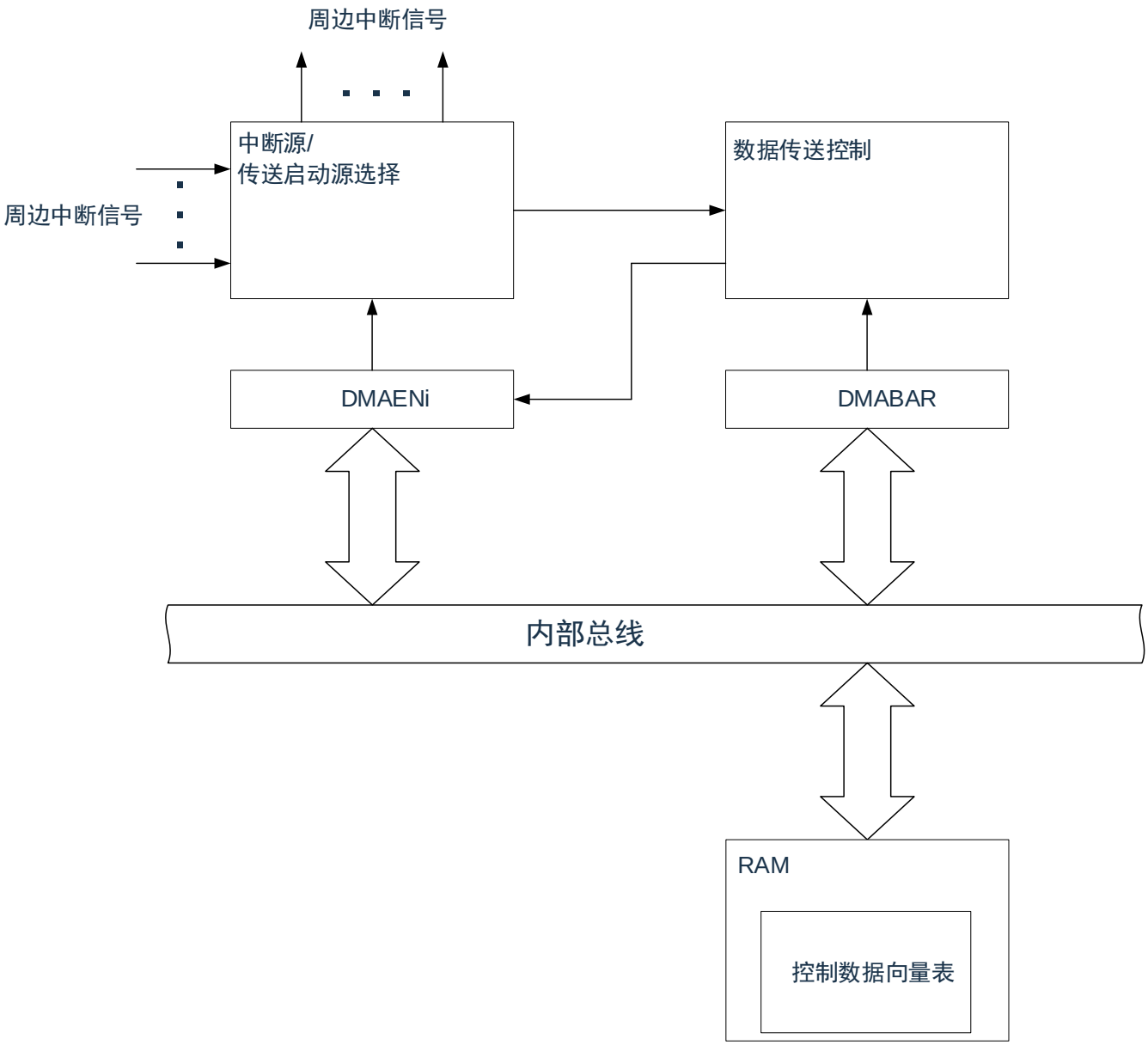
注：在深度睡眠模式，因为闪存停止运行，所以不能作为 DMA 传送源。

备注：i=0~2、j=0~23

21.2 DMA的结构

DMA 的框图如图 21-1 所示

图 21-1：DMA 的框图



21.3 控制DMA的寄存器

控制 DMA 的寄存器如表 21-2 所示。

表 21-2：控制 DMA 的寄存器

寄存器名	符号
外围允许寄存器 1	PER1
DMA 启动允许寄存器 0	DMAEN0
DMA 启动允许寄存器 1	DMAEN1
DMA 启动允许寄存器 2	DMAEN2
DMAENi 置位寄存器 0	DMSET0
DMAENi 置位寄存器 1	DMSET1
DMAENi 置位寄存器 2	DMSET2
DMAENi 复位寄存器 0	DMCLR0
DMAENi 复位寄存器 1	DMCLR1
DMAENi 复位寄存器 2	DMCLR2
DMA 基址寄存器	DMABAR

备注：i=0~2

DMA 的控制数据如表 21-3 所示。

DMA 的控制数据分配在 RAM 的 DMA 控制数据区。通过 DMABAR 寄存器设置 DMA 控制数据区和包含 DMA 向量表区（保存控制数据的起始地址）的 704 字节区域。

表 21-3：DMA 的控制数据

寄存器名	符号
DMA控制寄存器j	DMACRj
DMA块大小寄存器j	DMBLSj
DMA传送次数寄存器j	DMACTj
DMA传送次数重加载寄存器j	DMRLDj
DMA源地址寄存器j	DMSARj
DMA目标地址寄存器j	DMDARj

备注：j=0~23

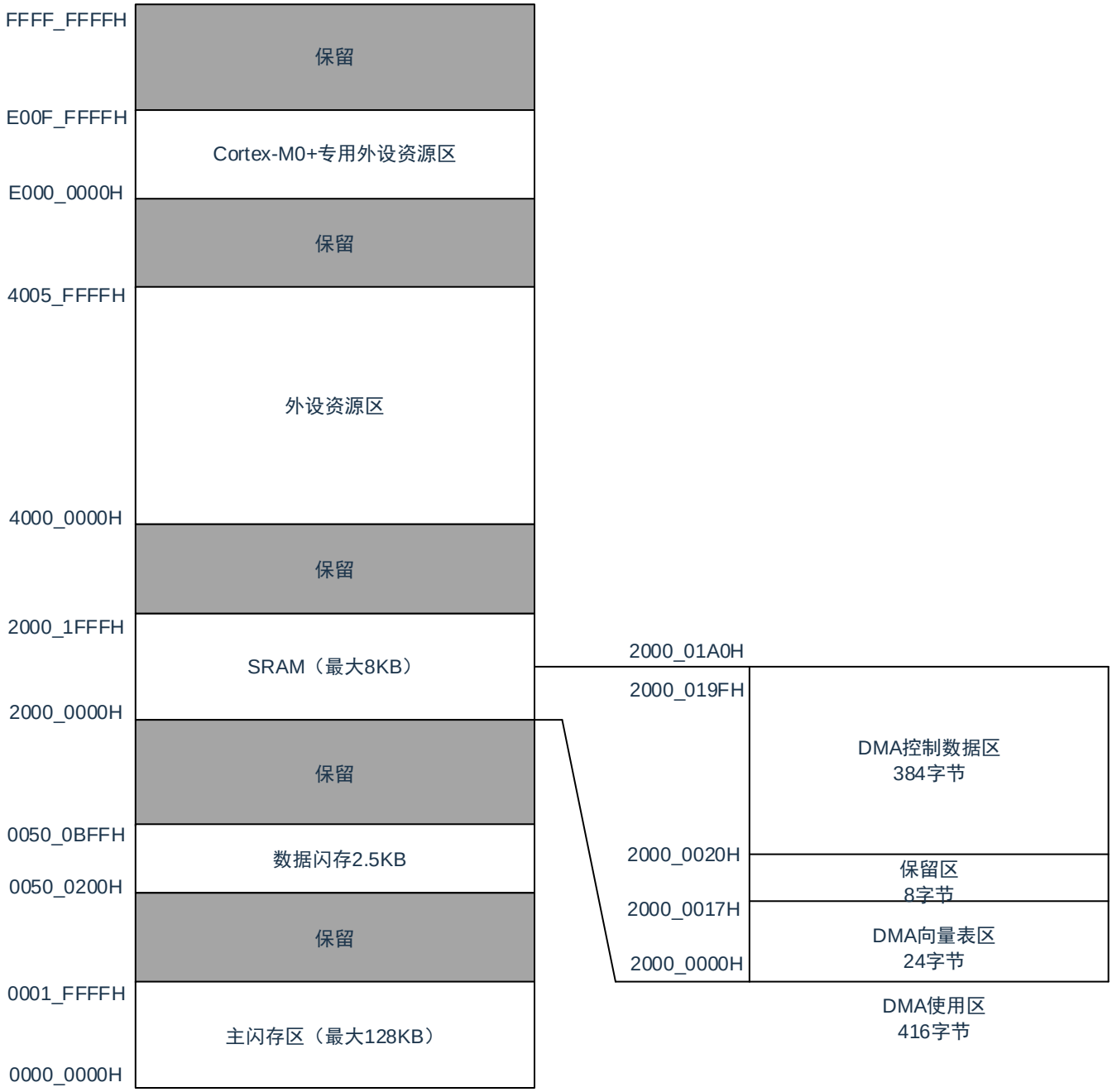
21.3.1 DMA控制数据区和DMA向量表区的分配

通过 DMABAR 寄存器将分配 DMA 的控制数据和向量表的 416 字节区域设置到 RAM 区。

DMABAR 寄存器的设置值为“2000_0000H”时的存储器映像例子如图 21-2 所示。

DMA 控制数据区的 384 字节中 DMA 不使用的空间能用作 RAM。

图 21-2：DMABAR 寄存器的设置值为“2000_0000H”时的存储器映像例子



21.3.2 控制数据的分配

从起始地址开始，按照 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj（j=0~23）寄存器的顺序分配控制数据。

起始地址由 DMABAR 寄存器设置，低 10 位由各启动源分配的向量表分别设置。

控制数据的分配如图 21-3 所示。

注意：

1. 必须在对应的 DMAENi（i=0~2）的 DMAENi0~DMAENi7 位为“0”（禁止启动）时更改 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj 寄存器的数据。
2. 不能通过 DMA 传送进行 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj 和 DMDARj 的存取。

图 21-3：控制数据的分配(DMABAR 设置为 2000_0000H)

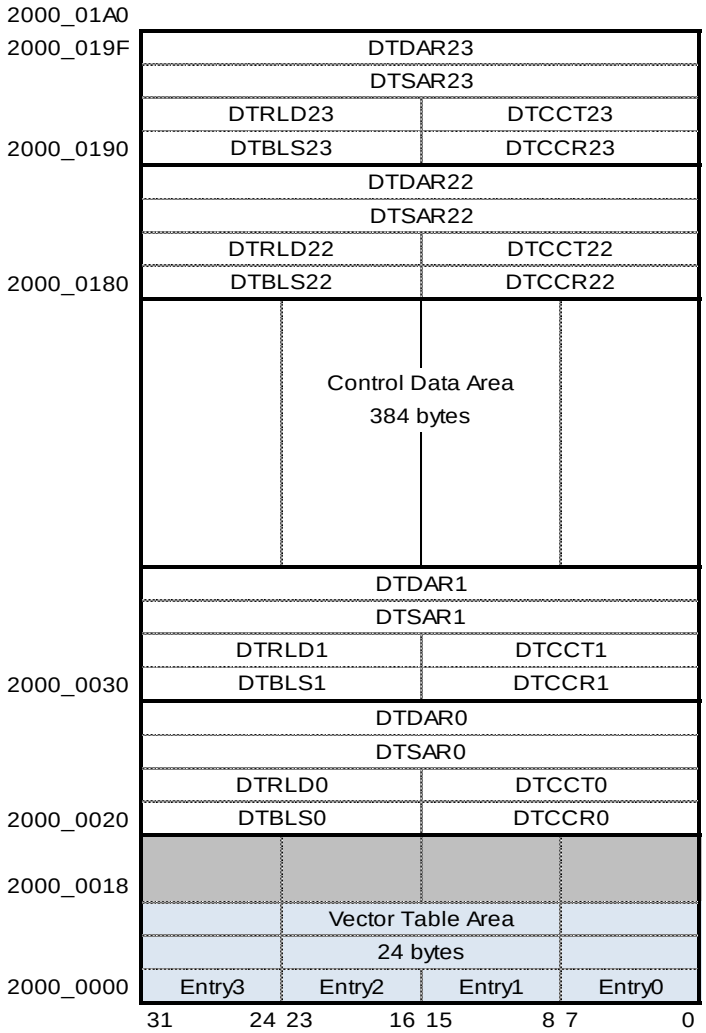


表 21-4：控制数据的起始地址

j	地址
23	baseaddr+190H
22	baseaddr+180H
21	baseaddr+170H
20	baseaddr+160H
19	baseaddr+150H
18	baseaddr+140H
17	baseaddr+130H
16	baseaddr+120H
15	baseaddr+110H
14	baseaddr+100H
13	baseaddr+0F0H
12	baseaddr+0E0H
11	baseaddr+0D0H
10	baseaddr+0C0H
9	baseaddr+0B0H
8	baseaddr+0A0H
7	baseaddr+090H
6	baseaddr+080H
5	baseaddr+070H
4	baseaddr+060H
3	baseaddr+050H
2	baseaddr+040H
1	baseaddr+030H
0	baseaddr+020H

备注：baseaddr：DMABAR 寄存器的设置值

21.3.3 向量表

DMA 一旦启动，就通过从各启动源分配的向量表读取的数据来决定控制数据，读被分配在 DMA 控制数据区的控制数据。

DMA 启动源和向量地址如表 21-5 所示。各启动源的向量表有 1 字节，保存“00H”~“17H”的数据，从 24 组的控制数据中选择 1 组数据。向量地址高 22 位由 DMABAR 寄存器设置，低 10 位被分配了对应启动源的“00H”~“17H”

注意：必须在对应的 DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位为“0”（禁止启动）时更改设置在向量表中的 DMA 控制数据区的起始地址。

图 21-4：控制数据的起始地址和向量表

DMABAR 寄存器的设置值为“2000_0000H”的情况（例）

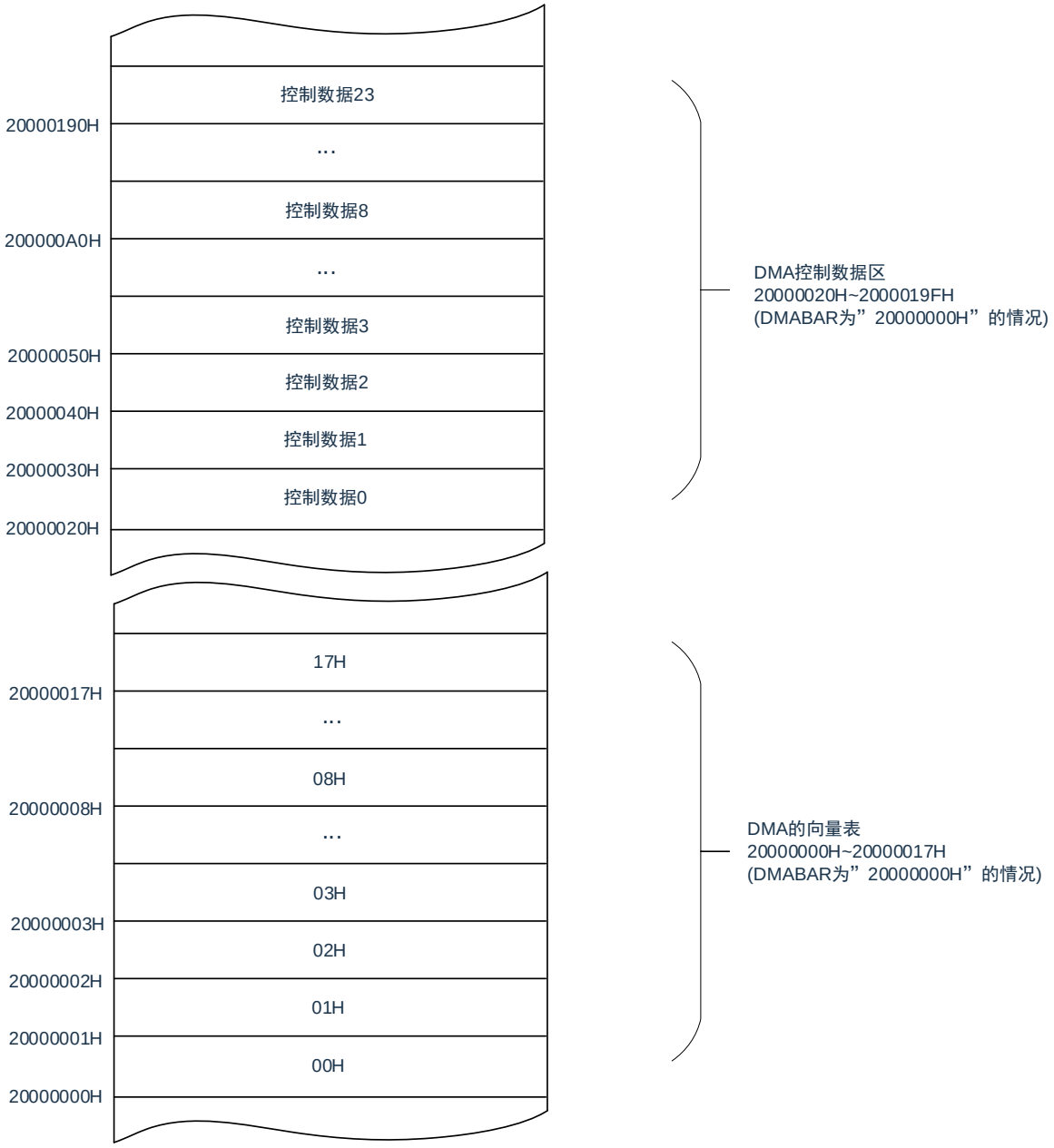


表 21-5：DMA 启动源和向量地址

DMA 启动源（中断请求发生源）	源号	向量地址	优先级
外部中断引脚输入边沿检测0	0	DMABAR 寄存器的设定地址+00H	高
外部中断引脚输入边沿检测1	1	DMABAR 寄存器的设定地址+01H	
外部中断引脚输入边沿检测2	2	DMABAR 寄存器的设定地址+02H	
外部中断引脚输入边沿检测3	3	DMABAR 寄存器的设定地址+03H	
键中断	4	DMABAR 寄存器的设定地址+04H	
A/D转换结束	5	DMABAR 寄存器的设定地址+05H	
UART0发送的传送结束或者缓冲器空中断/SSPI00的传送结束或者缓冲器空中断/IIC00的传送结束	6	DMABAR 寄存器的设定地址+06H	
UART0接收的传送结束/SSPI01的传送结束或者缓冲器空中断/IIC01的传送结束	7	DMABAR 寄存器的设定地址+07H	
UART1发送的传送结束或者缓冲器空中断/SSPI10的传送结束或者缓冲器空中断/IIC10的传送结束	8	DMABAR 寄存器的设定地址+08H	
UART1接收的传送结束/SSPI11的传送结束或者缓冲器空中断/IIC11的传送结束	9	DMABAR 寄存器的设定地址+09H	
UART2发送的传送结束或者缓冲器空中断/SSPI20的传送结束或者缓冲器空中断/IIC20的传送结束	10	DMABAR 寄存器的设定地址+0AH	
UART2接收的传送结束/SSPI21的传送结束或者缓冲器空中断/IIC21的传送结束	11	DMABAR 寄存器的设定地址+0BH	
IICA0通信结束	12	DMABAR 寄存器的设定地址+0CH	
高速SPI通信结束	13	DMABAR 寄存器的设定地址+0DH	
定时器通道00计数结束或捕捉结束	14	DMABAR 寄存器的设定地址+0EH	
定时器通道01计数结束或捕捉结束	15	DMABAR 寄存器的设定地址+0FH	
定时器通道02计数结束或捕捉结束	16	DMABAR 寄存器的设定地址+10H	
定时器通道03计数结束或捕捉结束	17	DMABAR 寄存器的设定地址+11H	
TimerA 下溢	18	DMABAR 寄存器的设定地址+12H	
RTC定周期信号/闹钟一致检出	19	DMABAR 寄存器的设定地址+13H	
间隔定时器中断检出	20	DMABAR 寄存器的设定地址+14H	
比较器检测0	21	DMABAR 寄存器的设定地址+15H	低
比较器检测1	22	DMABAR 寄存器的设定地址+16H	
FLASH编程结束中断	23	DMABAR 寄存器的设定地址+17H	

21.3.4 外围允许寄存器1（PER1）

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

要使用 DMA 时，必须将 bit1（DMAEN）置“1”。

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

详细请参考 4.3.6 外围允许寄存器 0、1（PER0、PER1）

21.3.5 DMA控制寄存器j（DMACRj）（j=0~23）

DMACRj 寄存器控制 DMA 的运行模式。

图 21-5：DMA 控制寄存器 j（DMACRj）的格式

地址：参照“21.3.2：控制数据的分配”。 复位后：不定值 R/W

符号：	15	14	13	12	11	10	9	8
DMACRj	0	0	0	0	0	0	0	FIFO
	7	6	5	4	3	2	1	0
	SZ		RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

FIFO	FIFO块传输控制
0	不是FIFO块传输
1	是FIFO块传输，源地址（SAMOD=0）或目的地址（DAMOD=0）绝对固定

SZ	传送数据长度的选择
00	8位
01	16位
10	32位
11	禁止设置

RPTINT	重复模式中断的允许/禁止
0	禁止发生中断。
1	允许发生中断。
在MODE位为“0”（正常模式）时，RPTINT位的设置无效。	

CHNE	链传送的允许/禁止
0	禁止链传送。
1	允许链传送。

DAMOD	传送目标地址的控制
0	固定
1	递增
在MODE位为“1”（重复模式）并且RPTSEL位为“0”（传送目标为重复区）时，DAMOD位的设置无效。	

SAMOD	传送源地址的控制
0	固定
1	递增
在MODE位为“1”（重复模式）并且RPTSEL位为“1”（传送源为重复区）时，SAMOD位的设置无效。	

RPTSEL	重复区的选择
0	传送目标为重复区。
1	传送源为重复区。
在MODE位为“0”（正常模式）时，RPTSEL位的设置无效。	

MODE	传送模式的选择
0	正常模式
1	重复模式

注意：不能通过 DMA 传送进行 DMACRj 寄存器的存取。

21.3.6 DMA块大小寄存器j（DMBLSj）（j=0~23）

此寄存器设置 1 次启动传送数据的块大小。

图 21-6：DMA 块大小寄存器 j（DMBLSj）的格式

地址：参照“21.3.2 控制数据的分配”。　复位后：不定值　R/W

符号：	15	14	13	12	11	10	9	8
DMBLSj	DMBLSj15	DMBLSj14	DMBLSj13	DMBLSj12	DMBLSj11	DMBLSj10	DMBLSj9	DMBLSj8
	7	6	5	4	3	2	1	0
	DMBLSj7	DMBLSj6	DMBLSj5	DMBLSj4	DMBLSj3	DMBLSj2	DMBLSj1	DMBLSj0

DMBLSj	传送块大小		
	8位传送	16位传送	32位传送
00H	禁止设置	禁止设置	禁止设置
01H	1字节	2字节	4字节
02H	2字节	4字节	8字节
03H	3字节	6字节	12字节
• • •	• • •	• • •	• • •
FDH	253字节	506字节	1012字节
FEH	254字节	508字节	1016字节
FFH	255字节	510字节	1020字节
• • •	• • •	• • •	• • •
FFFFH	65535字节	131070字节	262140字节

注意：不能通过 DMA 传送进行 DMBLSj 寄存器的存取。

21.3.7 DMA传送次数寄存器j（DMACTj）（j=0~23）

此寄存器设置 DMA 的数据传送次数。每当启动 1 次 DMA 传送就减 1。

图 21-7：DMA 传送次数寄存器 j（DMACTj）的格式

地址：参照“21.3.2 控制数据的分配”。　复位后：不定值　R/W

符号：	15	14	13	12	11	10	9	8
DMACTj	DMACTj15	DMACTj14	DMACTj13	DMACTj12	DMACTj11	DMACTj10	DMACTj9	DMACTj8
	7	6	5	4	3	2	1	0
	DMACTj7	DMACTj6	DMACTj5	DMACTj4	DMACTj3	DMACTj2	DMACTj1	DMACTj0

DMACTj	传送次数
00H	禁止设置
01H	1次
02H	2次
03H	3次
• • •	• • •
FDH	253次
FEH	254次
FFH	255次
• • •	• • •
FFFFH	65535次

注意：不能通过 DMA 传送进行 DMACTj 寄存器的存取。

21.3.8 DMA传送次数重加载寄存器j（DMRLDj）（j=0~23）

此寄存器设置重复模式中的传送次数寄存器的初始值。在重复模式中，因为将此寄存器的值重新加载到 DMACT 寄存器，所以设置值必须和 DMACT 寄存器的初始值相同。

图 21-8：DMA 传送次数重加载寄存器 j（DMRLDj）的格式

地址：参照“21.3.2 控制数据的分配”。　复位后：不定值　R/W

符号：	15	14	13	12	11	10	9	8
DMRLDj	DMRLDj15	DMRLDj14	DMRLDj13	DMRLDj12	DMRLDj11	DMRLDj10	DMRLDj9	DMRLDj8
	7	6	5	4	3	2	1	0
	DMRLDj7	DMRLDj6	DMRLDj5	DMRLDj4	DMRLDj3	DMRLDj2	DMRLDj1	DMRLDj0

注意：不能通过 DMA 传送进行 DMRLDj 寄存器的存取。

21.3.9 DMA源地址寄存器j（DMSARj）（j=0~23）

此寄存器指定数据传送时的传送源地址。

当 DMACRj 寄存器的 SZ 位为“01”（16 位传送）时，忽视最低位而作为偶地址进行处理。

当 DMACRj 寄存器的 SZ 位为“10”（32 位传送）时，忽视低 2 位而作为 word 地址进行处理。

图 21-9：DMA 源地址寄存器 j（DMSARj）的格式

地址：参照“21.3.2 控制数据的分配”。　复位后：不定值　R/W

符号	31	30	29	28	27	26	25	24
DMSARj	DMSARj31	DMSARj30	DMSARj29	DMSARj28	DMSARj27	DMSARj26	DMSARj25	DMSARj24
	23	22	21	20	19	18	17	16
	DMSARj23	DMSARj22	DMSARj21	DMSARj20	DMSARj19	DMSARj18	DMSARj17	DMSARj16
	15	14	13	12	11	10	9	8
	DMSARj15	DMSARj14	DMSARj13	DMSARj12	DMSARj11	DMSARj10	DMSARj9	DMSARj8
	7	6	5	4	3	2	1	0
	DMSARj7	DMSARj6	DMSARj5	DMSARj4	DMSARj3	DMSARj2	DMSARj1	DMSARj0

注意：不能通过 DMA 传送进行 DMSARj 寄存器的存取。

21.3.10 DMA目标地址寄存器j（DMDARj）（j=0~23）

此寄存器指定数据传送时的传送目标地址。

当 DMACRj 寄存器的 SZ 位为“01”（16 位传送）时，忽视最低位而作为偶地址进行处理。

当 DMACRj 寄存器的 SZ 位为“10”（32 位传送）时，忽视低 2 位而作为 word 地址进行处理。

图 21-10：DMA 目标地址寄存器 j（DMDARj）的格式

地址：参照“21.3.2 控制数据的分配”。 复位后：不定值 R/W

符号	31	30	29	28	27	26	25	24
DMDARj	DMDARj31	DMDARj30	DMDARj29	DMDARj28	DMDARj27	DMDARj26	DMDARj25	DMDARj24
	23	22	21	20	19	18	17	16
	DMDARj23	DMDARj22	DMDARj21	DMDARj20	DMDARj19	DMDARj18	DMDARj17	DMDARj16
	15	14	13	12	11	10	9	8
	DMDARj15	DMDARj14	DMDARj13	DMDARj12	DMDARj11	DMDARj10	DMDARj9	DMDARj8
	7	6	5	4	3	2	1	0
	DMDARj7	DMDARj6	DMDARj5	DMDARj4	DMDARj3	DMDARj2	DMDARj1	DMDARj0

注意：不能通过 DMA 传送进行 DMDARj 寄存器的存取。

21.3.11 DMA启动允许寄存器i（DMAENi）（i=0~2）

这是控制允许或者禁止通过各中断源启动 DMA 的 8 位寄存器。中断源和 DMAENi0~DMAENi7 位的对应如表 21-6 所示。

能通过 8 位存储器操作指令设置 DMAENi 寄存器。

注意：

1. 必须在不产生对应该位的启动源的位置更改 DMAENi0~DMAENi7 位。
2. 不能通过 DMA 传送进行 DMAENi 寄存器的存取。
3. 分配的功能因产品而不同，必须将没有分配功能的位置“0”。

图 21-11：DMA 启动允许寄存器 i（DMAENi）（i=0~2）的格式

地址：40005000H（DMAEN0）、40005001H（DMAEN1）、40005002H（DMAEN2） 复位后：00H

R/W

符号	7	6	5	4	3	2	1	0
DMAENi	DMAENi7	DMAENi6	DMAENi5	DMAENi4	DMAENi3	DMAENi2	DMAENi1	DMAENi0

DMAENi7	DMA启动的允许i7
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi7位变为“0”（禁止启动）。	

DMAENi6	DMA启动的允许i6
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi6位变为“0”（禁止启动）。	

DMAENi5	DMA启动的允许i5
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi5位变为“0”（禁止启动）。	

DMAENi4	DMA启动的允许i4
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi4位变为“0”（禁止启动）。	

DMAENi3	DMA启动的允许i3
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi3位变为“0”（禁止启动）。	

DMAENi2	DMA启动的允许i2
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi2位变为“0”（禁止启动）。	

DMAENi1	DMA启动的允许i1
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi1位变为“0”（禁止启动）。	

DMAENi0	DMA启动的允许i0
0	禁止启动。
1	允许启动。
根据传送结束中断的发生条件，DMAENi0位变为“0”（禁止启动）。	

表 21-6：中断源和 DMAENi0~DMAENi7 位的对应

寄存器	DMAENi7 位	DMAENi6 位	DMAENi5 位	DMAENi4 位	DMAENi3 位	DMAENi2 位	DMAENi1 位	DMAENi0 位
DMAEN0	UART0 接收的传送结束/SSPI01 的传送结束或者缓冲器空中断/IIC01 的传送结束	UART0 发送的传送结束或者缓冲器空中断/SSPI00 的传送结束或者缓冲器空中断/IIC00 的传送结束	A/D 转换结束	键中断	外部中断引脚输入边沿检测 3	外部中断引脚输入边沿检测 2	外部中断引脚输入边沿检测 1	外部中断引脚输入边沿检测 0
DMAEN1	定时器通道 01 计数结束或捕捉结束	定时器通道 00 计数结束或捕捉结束	高速 SPI 通信结束	IICA0 通信结束	UART2 接收的传送结束/SSPI21 的传送结束或者缓冲器空中断/IIC21 的传送结束	UART2 发送的传送结束或者缓冲器空中断/SSPI20 的传送结束或者缓冲器空中断/IIC20 的传送结束	UART1 接收的传送结束/SSPI11 的传送结束或者缓冲器空中断/IIC11 的传送结束	UART1 发送的传送结束或者缓冲器空中断/SSPI10 的传送结束或者缓冲器空中断/IIC10 的传送结束
DMAEN2	FLASH 编程结束中断	比较器检测 1	比较器检测 0	间隔定时器中断检出	RTC 定周期信号/闹钟一致检出	TimerA 下溢	定时器通道 03 计数结束或捕捉结束	定时器通道 02 计数结束或捕捉结束

注意：必须将没有分配功能的位置“0”。

备注：i=0~2

21.3.12 DMA基址寄存器（DMABAR）

这是 32 位寄存器，设置保存 DMA 控制数据区起始地址的向量地址以及 DMA 控制数据区的地址。

注意：

1. 必须在将全部的 DMA 启动源设置为禁止启动的状态下更改 DMABAR 寄存器。
2. 只能改写 1 次 DMABAR 寄存器。
3. 不能通过 DMA 传送进行 DMABAR 寄存器的存取。
4. 有关 DMA 控制数据区和 DMA 向量表区的分配，请参照“21.3.1：DMA 控制数据区和 DMA 向量表区的分配”的注意。
5. 设置该寄存器请保持 1024Byte 对齐，也就是低 10 位设置为零。DMA 硬件忽略低 10 位。
6. 该寄存器只能 WORD 访问，BYTE 和 HALFWORD 访问忽略。

图 21-12：DMA 基址寄存器（DMABAR）的格式

地址：40005008H 复位后：00000000H R/W

符号	31	30	29	28	27	26	25	24
DMABAR	DMABAR31	DMABAR30	DMABAR29	DMABAR28	DMABAR27	DMABAR26	DMABAR25	DMABAR24
	23	22	21	20	19	18	17	16
	DMABAR23	DMABAR22	DMABAR21	DMABAR20	DMABAR19	DMABAR18	DMABAR17	DMABAR16
	15	14	13	12	11	10	9	8
	DMABAR15	DMABAR14	DMABAR13	DMABAR12	DMABAR11	DMABAR10	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

21.3.13 DMAENi置位寄存器i（DMSETi）（i=0~2）

这是 DMA 启动允许寄存器 DMAENi 的置位寄存器，设置相应的位为 1 则可以将 DMAENi 的相应位置位为 1。能通过 8 位存储器操作指令设置 DMSETi 寄存器。

图 21-13：DMAENi 置位寄存器 i（DMSETi）（i=0~2）的格式

地址：40005018H（DMSET0）、40005019H（DMSET1）、4000501AH（DMSET2） 复位后：00H W

符号	7	6	5	4	3	2	1	0
DMSETi	DMSETi7	DMSETi6	DMSETi5	DMSETi4	DMSETi3	DMSETi2	DMSETi1	DMSETi0

DMSETin	DMAENin位置位
0	无操作。
1	将DMAENin位置1。

注：i=0~2，n=0~7

21.3.14 DMAENi复位寄存器i（DMCLRi）（i=0~2）

这是 DMA 启动允许寄存器 DMAENi 的复位寄存器，设置相应的位为 1 则可以将 DMAENi 的相应位复位为 0。能通过 8 位存储器操作指令设置 DMCLRi 寄存器。

图 21-14：DMAENi 复位寄存器 i（DMCLRi）（i=0~2）的格式

地址：40005020H（DMCLR0）、40005021H（DMCLR1）、40005022H（DMCLR2） 复位后：00H W

符号	7	6	5	4	3	2	1	0
DMCLRi	DMCLRi7	DMCLRi6	DMCLRi5	DMCLRi4	DMCLRi3	DMCLRi2	DMCLRi1	DMCLRi0

DMCLRin	DMAENin位清零
0	无操作。
1	将DMAENin位清0。

注：i=0~2，n=0~7

21.4 DMA的运行

DMA 一旦启动，就从 DMA 控制数据区读控制数据，根据此控制数据进行数据传送，并且将数据传送后的控制数据回写到 DMA 控制数据区。能将 24 组控制数据保存到 DMA 控制数据区，并且进行 24 组数据的传送。传送模式有正常模式和重复模式，传送大小有 8 位传送，16 位传送和 32 位传送。在 DMACRj（j=0~23）寄存器的 CHNE 位为“1”（允许链传送）时，通过 1 个启动源读多个控制数据进行连续的数据传送（链传送）。

通过 32 位 DMSARj 寄存器和 32 位 DMDARj 寄存器分别指定传送源地址和传送目标地址。在数据传送后，根据控制数据递增或者固定 DMSARj 寄存器和 DMDARj 寄存器的值。

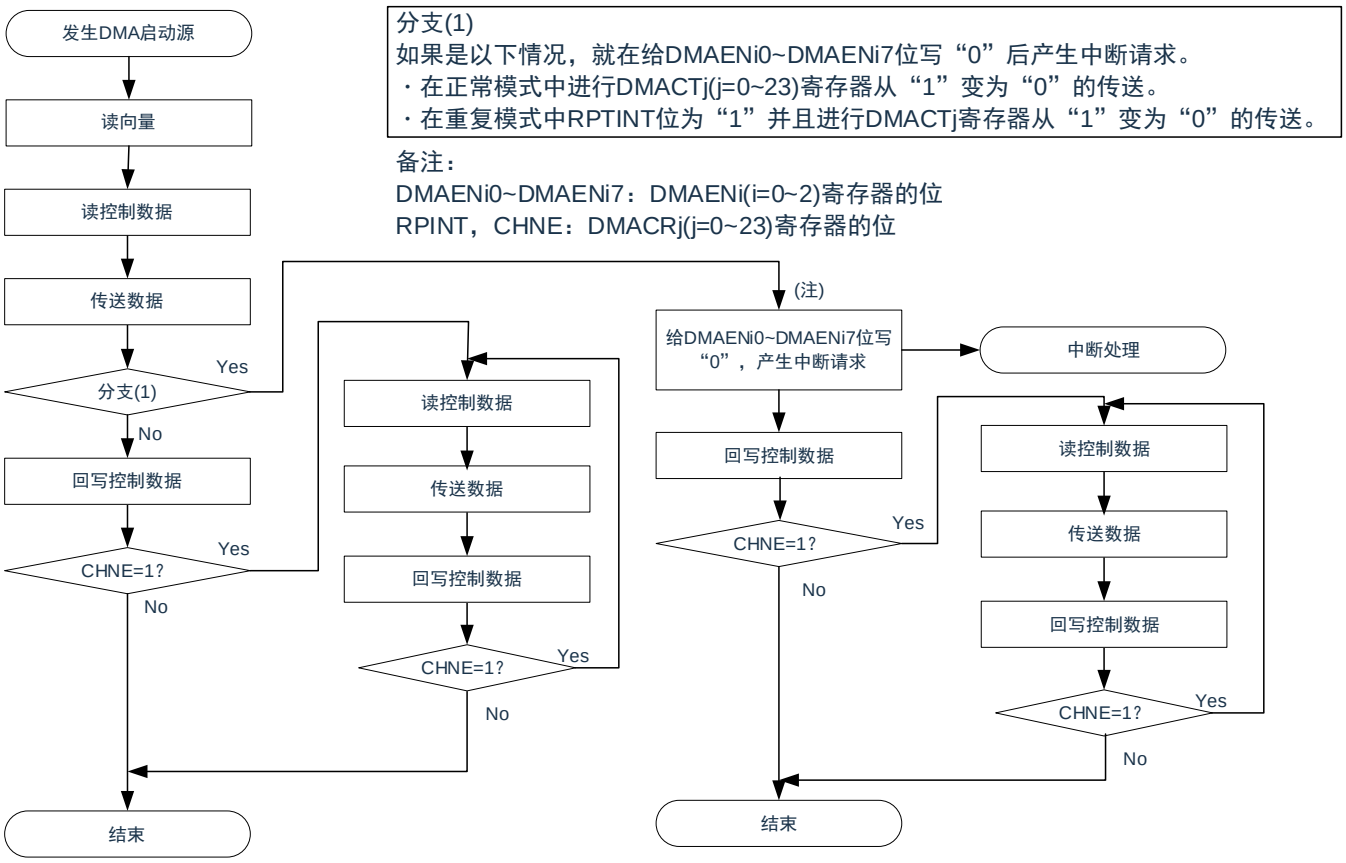
21.4.1 启动源

通过外围功能的中断信号启动 DMA，并且通过 DMAENi（i=0~2）寄存器选择启动 DMA 的中断信号。当数据传送（在进行链传送时，连续进行最初的传送）的设置为下述两种情况时，就在 DMA 运行中将对应的 DMAENi 寄存器的 DMAENi0~DMAENi7 位置“0”（禁止启动）。

- 在正常模式中，进行 DMACTj（j=0~23）寄存器变为“0”的传送。
- 在重复模式中，DMACRj 寄存器的 RPTINT 位为“1”（允许发生中断）并且进行 DMACTj 寄存器变为“0”的传送。

DMA 的内部运行流程图如图 21-15 所示。

图 21-15：DMA 的内部运行流程图



注：在通过允许链传送(CHNE=1)的设置启动的数据传送中，不给 DMAENi0~DMAENi7 位写“0”并且不产生中断请求。

21.4.2 正常模式

在 8 位传送时，1 次启动的传送数据为 1~65535 字节；在 16 位传送时，1 次启动的传送数据为 2~131070 字节；在 32 位传送时，1 次启动的传送数据为 4~262140 字节。传送次数为 1~65535 次。如果进行 DMACTj（j=0~23）寄存器变为“0”的数据传送，就在 DMA 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位置“0”（禁止启动）。

正常模式的寄存器功能和数据传送分别如表 21-7 和图 21-16 所示。

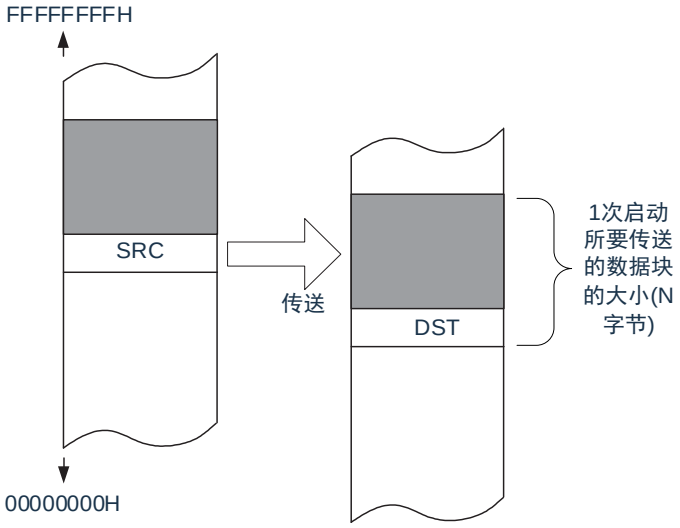
表 21-7：正常模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	不使用[注]。
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

注：当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须进行初始化（00H）

备注：j=0~23

图 21-16：正常模式的数据传送



DMACR寄存器的设置				源地址的控制	目标地址的控制	传送后的源地址	传送后的目标地址
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	固定	固定	SRC	DST
0	1	X	0	递增	固定	SRC+N	DST
1	0	X	0	固定	递增	SRC	DST+N
1	1	X	0	递增	递增	SRC+N	DST+N

备注：DMBLSj 寄存器=N

DMSARj 寄存器=SRC

DMDARj 寄存器=DST

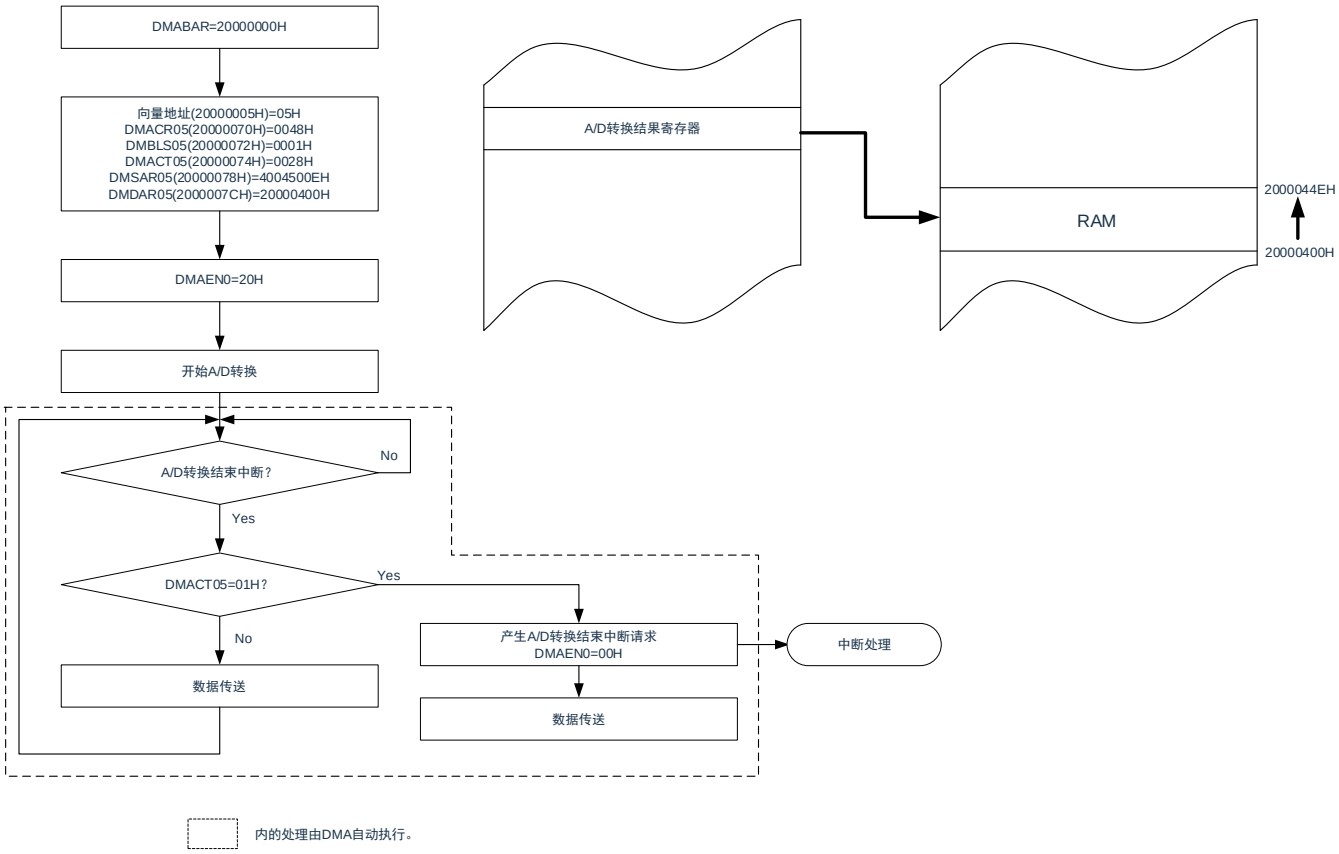
X：“0”或者“1”

(1) 正常模式的使用例子 1：连续取 A/D 转换结果

通过 A/D 转换结束中断启动 DMA CH5，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 2000_0005H，控制数据分配在 2000_0070H~2000_007FH。
- 将 A/D 转换结果寄存器（4004_500EH，4004_500FH）的 2 字节数据传送 40 次到 RAM 的 2000_0400H~2000_044FH 的 80 字节。

图 21-17：正常模式的使用例子 1：连续取 A/D 转换结果



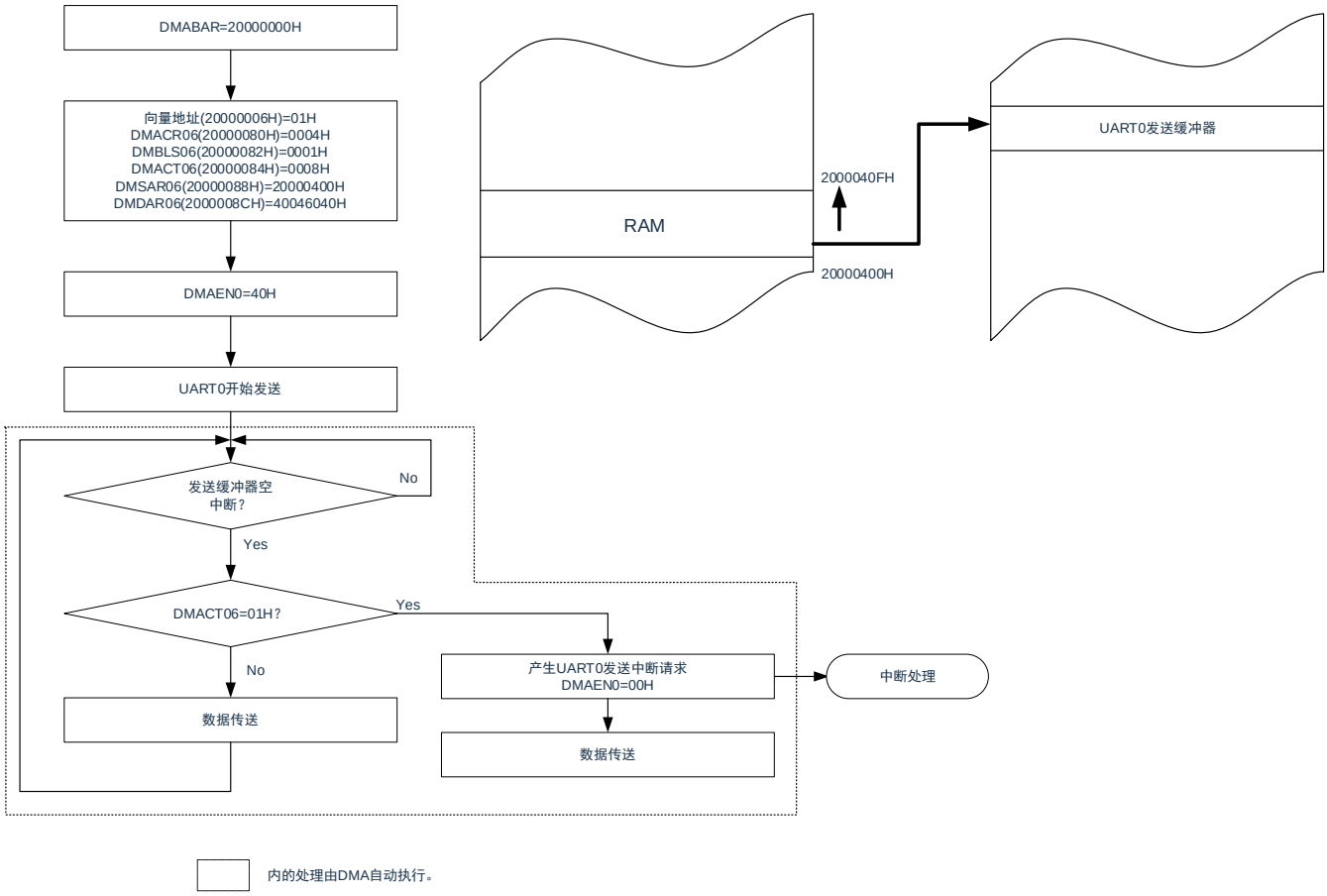
因为是正常模式，所以不使用 DMRLD05 寄存器的值。但是，当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须对 DMRLD05 寄存器进行初始化（0000H）。

(2) 正常模式的使用例子 2：UART0 连续发送

通过 UART0 的发送缓冲器空中断启动 DMA CH6，并且将 RAM 的值传送到 UART0 的发送缓冲器。

- 向量地址分配在 2000_0006H，控制数据分配在 2000080H~2000008FH。
- 将 RAM 的 2000_0400H~2000_0407H 的 8 字节传送到 UART0 的发送缓冲器（4004_6040H）。

图 21-18：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 DMRLD06 寄存器的值。但是，当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，必须对 DMRLD06 寄存器进行初始化（0000H）。

必须通过软件开始第 1 次的 UART0 发送。通过发送缓冲器空中断启动 DMA，然后自动进行第 2 次以后的发送。

21.4.3 重复模式

1 次启动的传送数据为 1~65535 字节。将传送源或者传送目标指定为重复区，传送次数为 1~65535 次。一旦指定次数的传送结束，就对 DMACTj（j=0~23）寄存器以及指定为重复区的地址进行初始化，然后重复进行传送。当 DMACRj 寄存器的 RPTINT 位为“1”（允许发生中断）并且进行 DMACTj 寄存器变为“0”的数据传送时，就在 DMA 运行中向中断控制器产生对应启动源的中断请求，并且将对应的 DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位置“0”（禁止启动）。当 DMACRj 寄存器的 RPTINT 位为“0”（禁止发生中断）时，即使进行 DMACTj 寄存器变为“0”的数据传送，也不产生中断请求，而且 DMAENi0~DMAENi7 位不变为“0”。

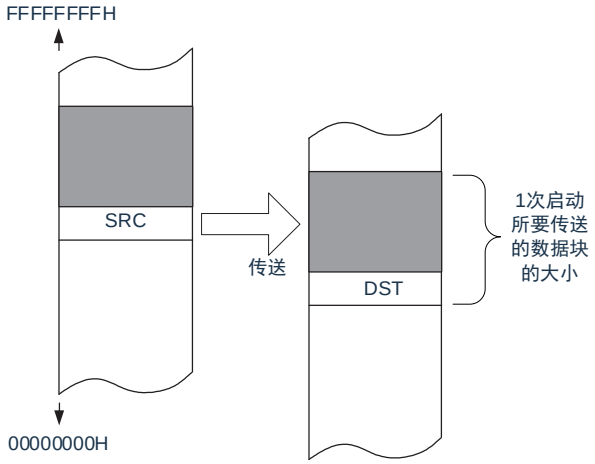
重复模式的寄存器功能和数据传送分别如表 21-8 和图 21-19 所示。

表 21-8：重复模式的寄存器功能

寄存器名	符号	功能
DMA块大小寄存器j	DMBLSj	1次启动所要传送的数据块大小
DMA传送次数寄存器j	DMACTj	数据的传送次数
DMA传送次数重加载寄存器j	DMRLDj	将此寄存器的值重新加载到DMACT寄存器。 （对数据的传送次数进行初始化）
DMA源地址寄存器j	DMSARj	数据的传送源地址
DMA目标地址寄存器j	DMDARj	数据的传送目标地址

备注：j=0~23

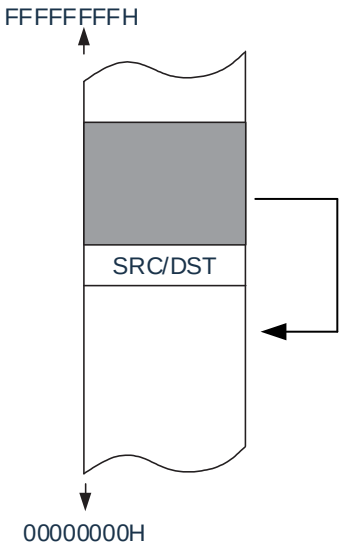
图 21-19：重复模式的数据传送



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”
DMBLSj 寄存器=N
DMACTj 寄存器≠1
DMSARj 寄存器=SRC
DMDARj 寄存器=DST

j=0~23



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”

DMBLSj 寄存器=N

DMACTj 寄存器=1

DMSARj 寄存器=SRC

DMDARj 寄存器=DST

j=0~23

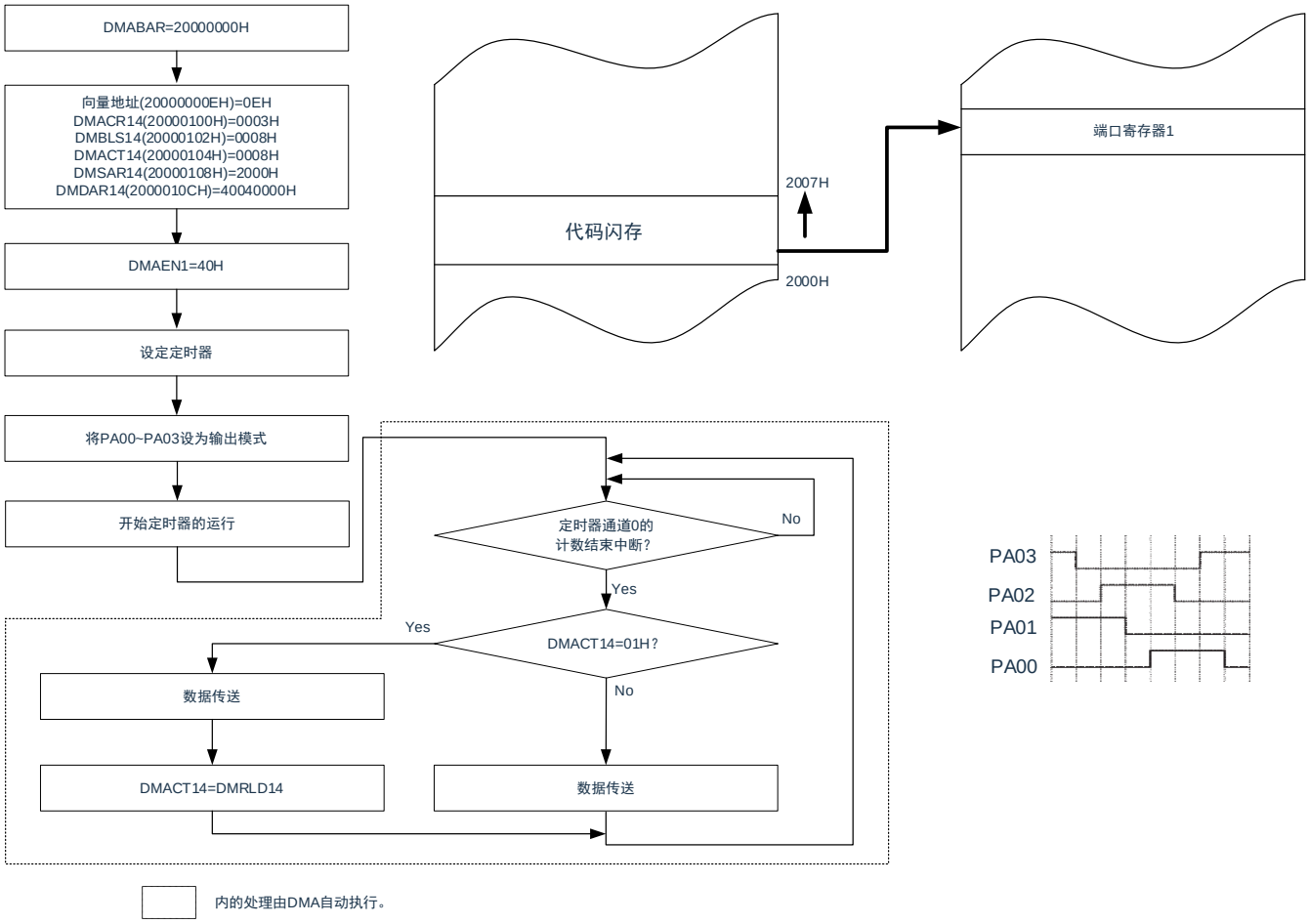
注意：在使用重复模式时，必须将重复区的数据长度设置在 65535 字节以内。

(1) 重复模式的使用例子：使用端口的步进马达控制脉冲输出

使用 Timer8 的通道 0 间隔定时器功能启动 DMA CH14，并且将保存在代码闪存的马达控制脉冲的模式传送到通用端口。

- 向量地址分配在 2000_000EH，控制数据分配在 2000_0100H~2000_010FH。
- 将代码闪存的 02000H~02007H 的 8 字节传送到端口寄存器 A（4004_0000H）。
- 禁止重复模式中断。

图 21-20：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



注意：要停止输出时，必须在停止定时器的运行后清除 DMAEN1 的 bit6。

21.4.4 链传送

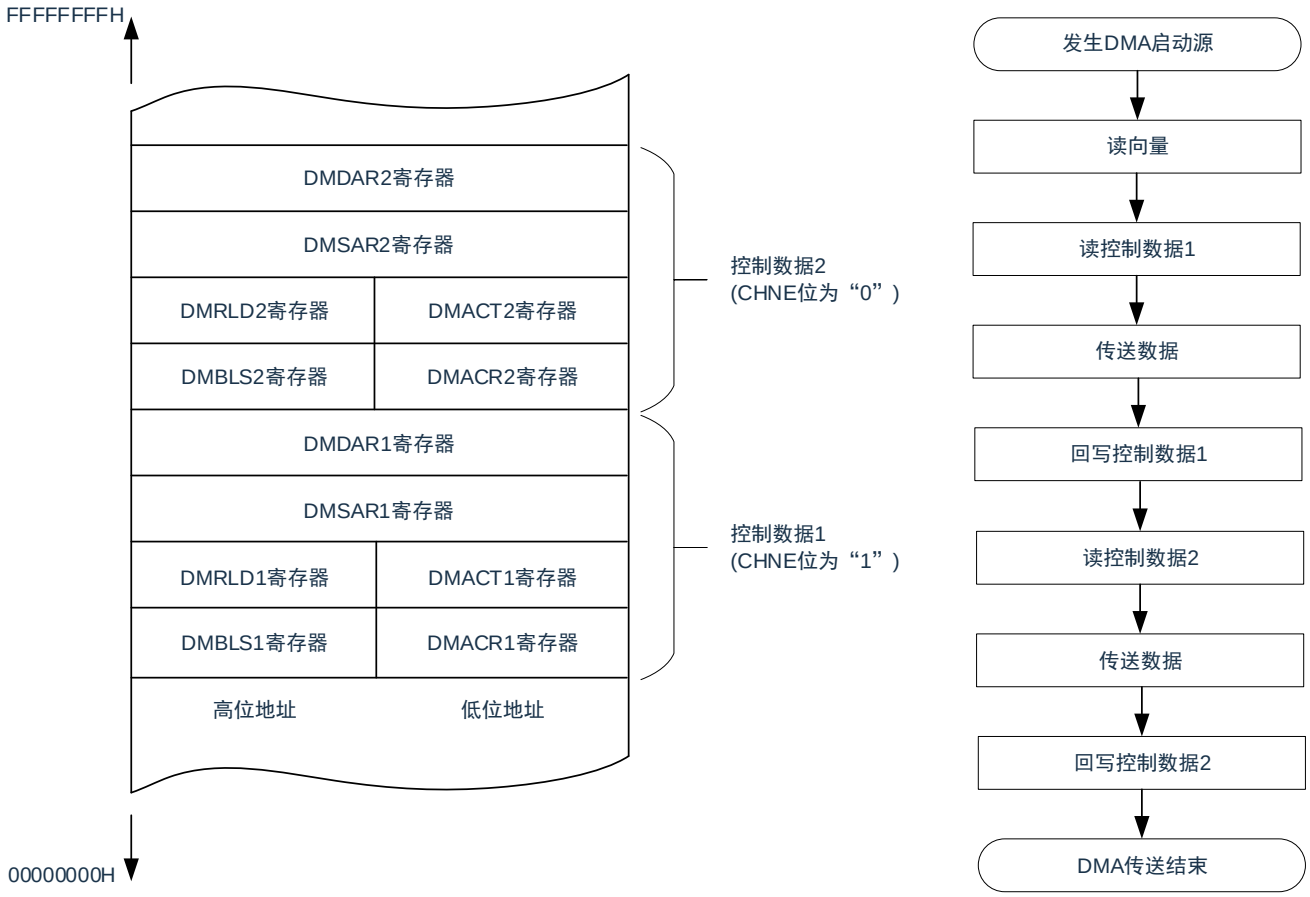
当 DMACRj（j=0~23）寄存器的 CHNE 位为“1（允许链传送）时，能通过 1 个启动源连续进行多个数据的传送

DMA 一旦启动，就通过从启动源对应的向量地址读取的数据来选择控制数据，读被分配在 DMA 控制数据区的控制数据。如果读到的控制数据的 CHNE 位为“1”（允许链传送），就在传送结束后读下一个被分配的控制数据，继续进行传送。重复此操作，直到 CHNE 位为“0”（禁止链传送）的控制数据传送结束为止。

在使用多个控制数据进行链传送时，第一个控制数据设置的传送次数有效，而第 2 个以后处理的控制数据的传送次数无效。

链传送的流程图如图 21-21 所示。

图 21-21：链传送的流程图



注意：

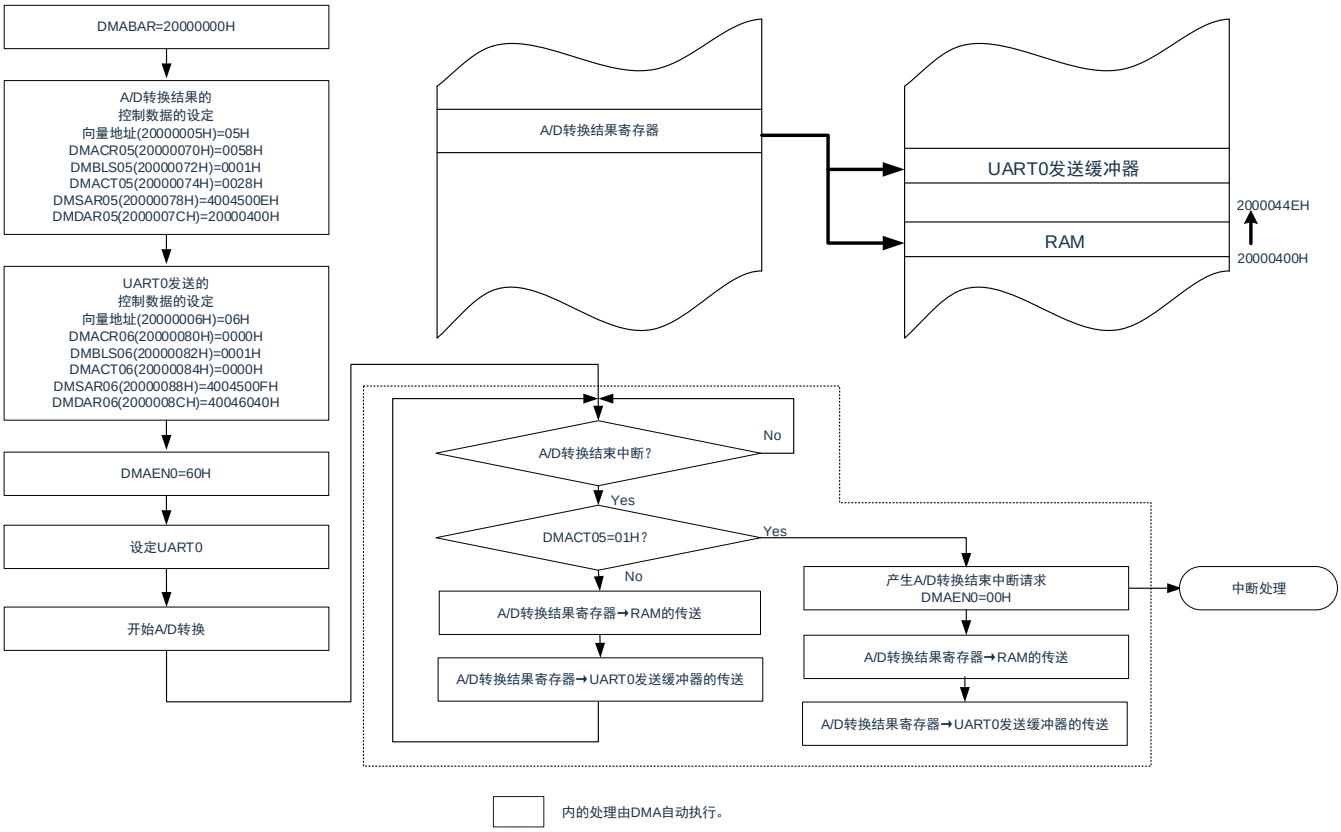
1. 必须将 DMACR23 寄存器的 CHNE 位置“0”（禁止链传送）。
2. 在链传送的第 2 次以后的数据传送时，DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位不变为“0”（禁止 DMA 启动），并且不产生中断请求。

(1) 链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送

通过 A/D 转换结束中断启动 DMA CH5，CH6，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 2000_0005H 和 2000_0006H。

- A/D 转换结果的控制数据分配在 2000_0070H~2000_007FH。
- UART0 发送的控制数据分配在 2000_0080H~2000_008FH。
- 将 A/D 转换结果寄存器（4004500EH，4004500FH）的 2 字节数据传送到 RAM 的 20000400H~2000044FH，并且将 A/D 转换结果寄存器的高位 1 字节（4004500FH）传送到 UART0 的发送缓冲器（40046040H）。

图 21-22：链传送的使用例子：连续取 A/D 转换结果进行 UART0 发送



21.5 使用DMA时的注意事项

21.5.1 DMA控制数据和向量表的设置

- 必须在将全部的 DMA 启动源设置为禁止启动的状态下更改 DMA 基址寄存器（DMABAR）。
- 只能改写 1 次 DMA 基址寄存器（DMABAR）。
- 必须在对应的 DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位为“0”（禁止 DMA 启动）时更改 DMACRj、DMBLSj、DMACTj、DMRLDj、DMSARj、DMDARj 寄存器的数据。
- 必须在对应的 DMAENi（i=0~2）寄存器的 DMAENi0~DMAENi7 位为“0”（禁止 DMA 启动）时更改设置在向量表中的 DMA 控制数据区的起始地址。

21.5.2 DMA控制数据区和DMA向量表区的分配

能分配 DMA 控制数据和向量表的区域因产品和使用条件而不同。

- 堆栈区、DMA 控制数据区和 DMA 向量表区不能重叠。
- 当通过 RAM 奇偶校验错误检测功能允许产生奇偶校验错误复位（RPERDIS=0）时，即使在使用正常模式时也必须对 DMRLD 寄存器进行初始化（0000H）。

21.5.3 DMA的执行时钟数

DMA 启动时的执行情况和所需的时钟数如表 21-9 所示。

表 21-9：DMA 启动时的执行情况和所需的时钟数

读向量	控制数据		读数据	写数据
	读	回写		
1	4	注1	注2	注2

注 1：有关回写控制数据所需的时钟数，请参照“表 21-10：回写控制数据所需的时钟数注 2：有关读写数据所需的时钟数，请参照“表 21-11：读写数据所需的时钟数

表 21-10：回写控制数据所需的时钟数

DMACR寄存器的设置				地址设置		控制寄存器的回写				时钟数
DAMOD	SAMOD	RPTSEL	MODE	源	目标	DMACTj 寄存器	DMRLDj 寄存器	DMSARj 寄存器	DMDARj 寄存器	
0	0	X	0	固定	固定	回写	回写	不回写	不回写	1
0	1	X	0	递增	固定	回写	回写	回写	不回写	2
1	0	X	0	固定	递增	回写	回写	不回写	回写	2
1	1	X	0	递增	递增	回写	回写	回写	回写	3
0	X	1	1	重复区	固定	回写	回写	回写	不回写	2
1	X	1	1		递增	回写	回写	回写	回写	3
X	0	0	1	固定	重复区	回写	回写	不回写	回写	2
X	1	0	1	递增		回写	回写	回写	回写	3

备注：j=0~39，X：“0”或者“1”

表 21-11：读写数据所需的时钟数

执行状态	RAM	代码闪存	数据闪存	特殊功能寄存器（SFR）	扩展特殊功能寄存器（2ndSFR）	
					无等待	等待
读数据	1	2	4	1	1	1+等待数注
写数据	1	—	—	1	1	1+等待数注

21.5.4 DMA的响应时间

DMA 响应时间如表 21-12 所示。DMA 响应时间是指从检测到 DMA 启动源到开始 DMA 传送的时间，不包括 DMA 的执行时钟数。

表 21-12：DMA 的响应时间

	最短时间	最长时间
响应时间	3个时钟	23个时钟

但是，在以下情况下 DMA 的响应可能还会延迟。延迟的时钟数因条件而不同。

- 从内部 RAM 执行指令的情况最长响应时间：20 个时钟

备注：1 个时钟：1/F_{CLK}（F_{CLK}：CPU/外围硬件时钟）

21.5.5 DMA的启动源

- 不能在从输入 DMA 启动源到结束 DMA 传送的期间输入相同的启动源。
- 在产生 DMA 启动源的位置，不能操作该启动源对应的 DMA 启动允许位。
- 如果 DMA 启动源发送竞争，就在 CPU 接受 DMA 传送时判断优先级，决定启动启动源。有关启动源的优先级，请参照“21.3.3：向量表”。
- 如果在以下某个状态下允许 DMA 启动，就开始 DMA 传送，并且在传送结束后产生中断。因此，必须根据需要在确认比较器的监视标志（CnMON）后置为允许 DMA 启动。
  - 设置为通过比较器[注]的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的上升沿产生中断请求（CnEPO=0）而且 IVCMP＞IVREF（或者内部基准电压 1.45V）。
  - 设置为通过比较器的单边沿检测产生中断请求（CnEDG=0）并且通过比较器的下降沿产生中断请求（CnEPO=1）而且 IVCMP＜IVREF（或者内部基准电压 1.45V）。

备注：n=0、1

21.5.6 待机模式中的运行

状态	DMA运行
睡眠模式	能运行（禁止在低功耗RTC模式中运行）。
深度睡眠模式	能接受DMA启动源，并进行DMA传送[注1]

注 1：在深度睡眠模式中，能在检测到 DMA 启动源后进行 DMA 传送，并且在传送结束后返回到深度睡眠模式。但是，因为在深度睡眠模式中代码闪存和数据闪存停止运行，所以不能将闪存设置为传送源。

第22章 联动控制器(EVENTC)

22.1 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。

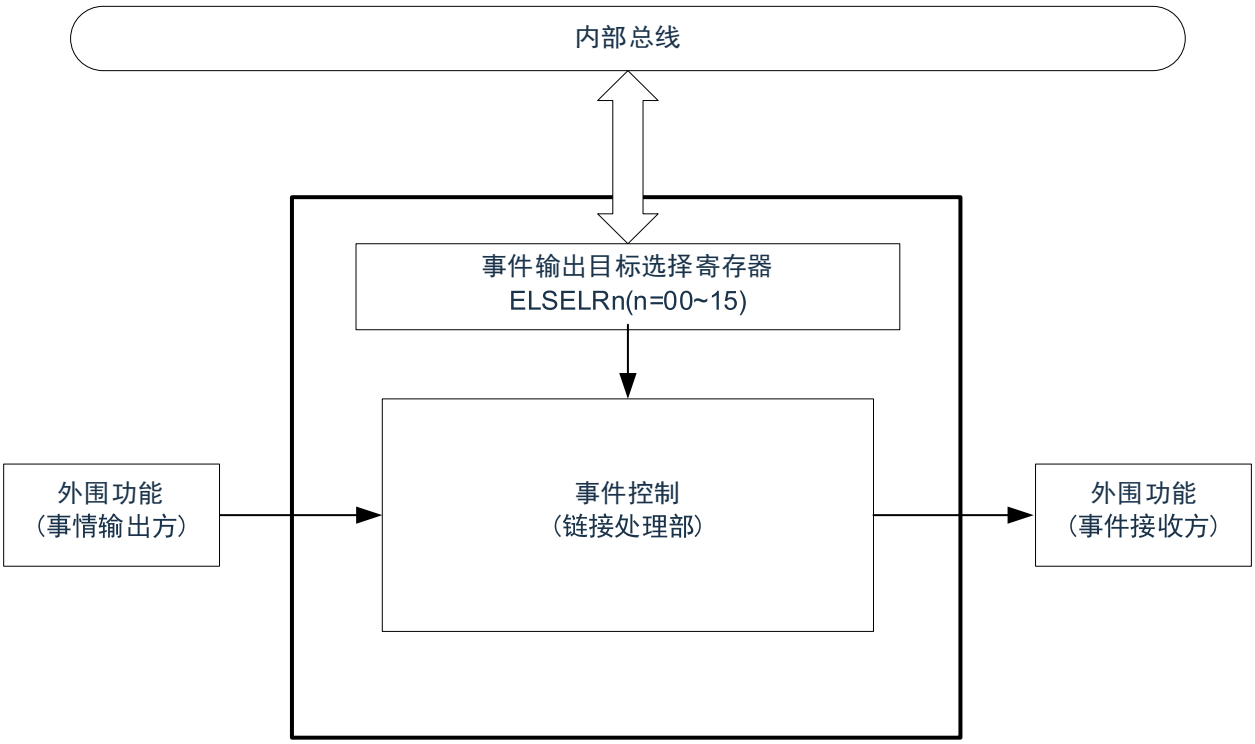
EVENTC 有以下功能：

- 根据产品，能将 16 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 6 种外围功能中的 1 种外围功能运行的启动源。

22.2 EVENTC的结构

EVENTC 的框图如图 22-1 所示。

图 22-1：EVENTC 的框图



22.3 控制寄存器

控制器寄存器如表 22-1 所示。

表 22-1：控制 EVENTC 的寄存器

寄存器名	符号
事件输出目标选择寄存器00	ELSELR00
事件输出目标选择寄存器01	ELSELR01
事件输出目标选择寄存器02	ELSELR02
事件输出目标选择寄存器03	ELSELR03
事件输出目标选择寄存器04	ELSELR04
事件输出目标选择寄存器05	ELSELR05
事件输出目标选择寄存器06	ELSELR06
事件输出目标选择寄存器07	ELSELR07
事件输出目标选择寄存器08	ELSELR08
事件输出目标选择寄存器09	ELSELR09
事件输出目标选择寄存器10	ELSELR10
事件输出目标选择寄存器11	ELSELR11
事件输出目标选择寄存器12	ELSELR12
事件输出目标选择寄存器13	ELSELR13
事件输出目标选择寄存器14	ELSELR14
事件输出目标选择寄存器15	ELSELR15

22.3.1 输出目标选择寄存器n（ELSELRn）（n=00~15）

ELSELRn 寄存器将各事件信号链接到事件接受方外围功能（链接目标外围功能）接受事件时的运行。不能将多个事件输入链接到相同的事件输出目标（事件接受方）。否则，事件接受方外围功能的运行可能不定而无法正常地接受事件信号。另外，不能将事件链接发生源和事件输出目标设置为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设置 ELSELRn 寄存器。

ELSELRn 寄存器（n=00~15）和外围功能的对应如表 22-2 所示，ELSELRn 寄存器（n=00~15）的设置值和链接目标外围功能接受事件时的运行的对应如表 22-3 所示。

图 22-2：事件输出目标选择寄存器 n（ELSELRn）的格式

地址：40041800H（ELSELR00）~4004180FH（ELSELR15） 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn3[注1]	ELSELn2	ELSELn1	ELSELn0	事件链接的选择
0	0	0	0	禁止事件链接。
0	0	0	1	选择所链接的外围功能1的运行[注1]。
0	0	1	0	选择所链接的外围功能2的运行[注1]。
0	0	1	1	选择所链接的外围功能3的运行[注1]。
0	1	0	0	选择所链接的外围功能4的运行[注1]。
0	1	0	1	选择所链接的外围功能5的运行[注1]。
0	1	1	0	选择所链接的外围功能6的运行[注1]。
上述以外				禁止设置。

注 1：请参照“表 22-3：ELSELRn 寄存器（n=00~15）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 22-2：ELSELRn 寄存器（n=00~15）和外围功能的对应

寄存器名	事件发生源（事件输入 n 的输出源）	事件内容
ELSELR00	外部中断边沿检测 0	INTP0
ELSELR01	外部中断边沿检测 1	INTP1
ELSELR02	外部中断边沿检测 2	INTP2
ELSELR03	外部中断边沿检测 3	INTP3
ELSELR04	外部中断边沿检测 4	INTP4
ELSELR05	外部中断边沿检测 5	INTP5
ELSELR06	键返回信号检测	INTKR
ELSELR07	RTC 固定周期/闹钟一致检测	INTRTC
ELSELR08	定时器 A 下溢	INTTMA
ELSELR09	Timer8 通道 00 的计数结束/捕捉结束	INTTM00
ELSELR10	Timer8 通道 01 的计数结束/捕捉结束	INTTM01
ELSELR11	Timer8 通道 02 的计数结束/捕捉结束	INTTM02
ELSELR12	Timer8 通道 03 的计数结束/捕捉结束	INTTM03
ELSELR13	比较器检测 0	INTCMP0
ELSELR14	比较器检测 1	INTCMP1
ELSELR15	停振检测中断	INTOSDC

表 22-3：ELSELRn 寄存器（n=00~15）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	Timer8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	Timer8通道1的定时器输入[注2]	延迟计数器、输入脉冲间隔的测量、外部事件计数器
0100B	4	定时器A	计数源
0101B	5	DA0[注3]	实时输出

注 1：要选择 Timer8 通道 0 的定时器输入作为链接目标外围功能时，必须先通过定时器时钟选择寄存器 0（TPS0）将通道 0 的运行时钟设置为 F_{CLK}，通过噪声滤波器允许寄存器 1（NFEN1）将 TI00 引脚的噪声滤波器置为 OFF（TNFEN00=0），并且通过定时器输入选择寄存器 0（TIS0）将通道 0 使用的定时器输入设置为联动控制器的事件输入信号。

注 2：要选择 Timer8 通道 1 的定时器输入作为链接目标外围功能时，必须先通过定时器时钟选择寄存器 0（TPS0）将通道 1 的运行时钟设置为 F_{CLK}，通过噪声滤波器允许寄存器 1（NFEN1）将 TI01 引脚的噪声滤波器置为 OFF（TNFEN01=0），并且通过定时器输入选择寄存器 0（TIS0）将通道 1 使用的定时器输入设置为 EVENTC 的事件输入信号。

注 3：如果要在 D/A 转换的实时输出模式有效时进入深度睡眠状态，就必须在进入深度睡眠模式之前禁止 EVENTC 的事件链接。

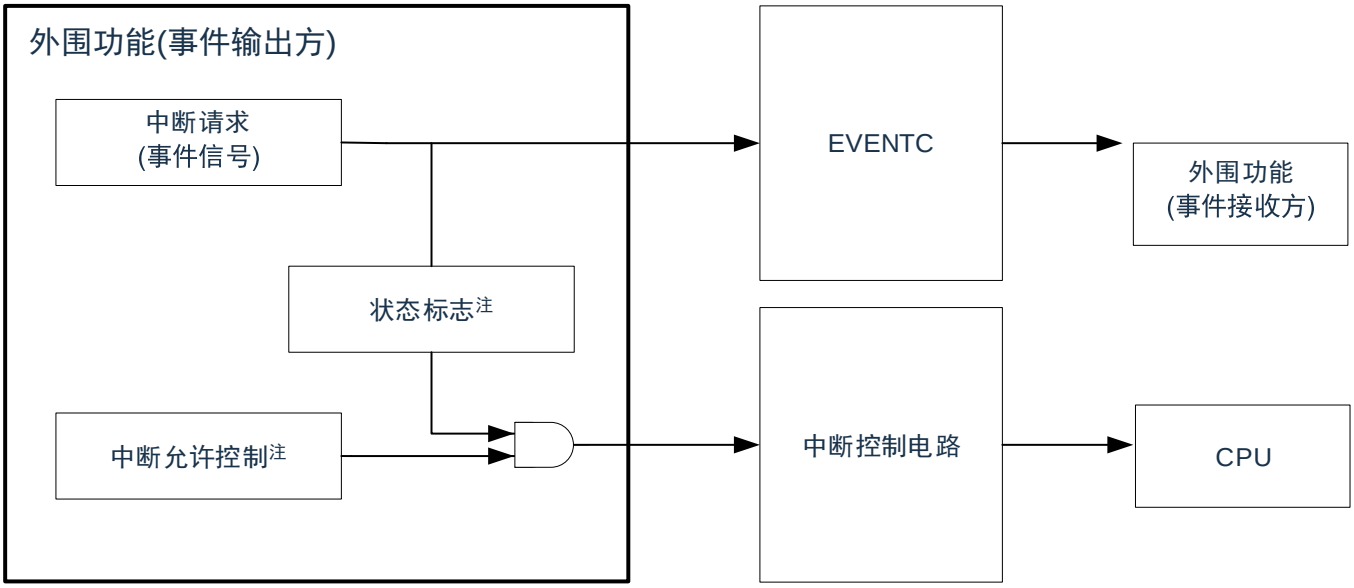
22.4 EVENTC的运行

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 22-3 所示。此图是以有中断请求状态标志和中断允许位（控制允许或者禁止）的外围功能为例的关系。

通过 EVENTC 接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行（参照“表 22-3：ELSELRn 寄存器（n=00~15）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 22-3：中断处理和 EVENTC 的关系



注：有些外围功能没有此功能。

接受事件的外围功能的响应如表 22-4 所示。

表 22-4：接受事件的外围功能的响应

事件接受目标No.	事件链接目标的功能	事件接受后的运行	响应
1	A/D转换器	A/D转换	EVENTC事件直接变为A/D转换的硬件触发。
2	Timer8 通道0的定时器输入	延迟计数器输入脉宽的测量外部事件计数器	在从发生EVENTC事件经过3个或者4个F_{CLK}周期后进行边沿的检测。
3	Timer8 通道1的定时器输入	延迟计数器输入脉宽的测量外部事件计数器	在从发生EVENTC事件经过3个或者4个F_{CLK}周期后进行边沿的检测。
4	定时器A	计数源	EVENTC事件直接变为定时器A的计数源。
5	D/A转换器的通道0	实时输出（通道0）	在从发生EVENTC事件经过2个或者3个F_{CLK}周期后开始通道0的D/A转换。

第23章 中断功能

Cortex-M0+处理器内置了嵌套向量中断控制器(NVIC),支持最多 32 个中断请求(IRQ)输入，以及 1 个不可屏蔽中断(NMI)输入，另外，处理器还支持多个内部异常。

本系统中对 32 个中断请求(IRQ)输入和 1 个不可屏蔽中断(NMI)输入的中断源进行了处理。本用户手册只对本系统中的处理进行了说明，Cortex-M0+处理器内置 NVIC 的功能，请参考 Cortex-M0+处理器的用户手册。

23.1 中断功能的种类

中断功能有以下 2 种。

(1) 可屏蔽中断

这是受屏蔽控制的中断。如果中断屏蔽标志寄存器没有打开，中断请求即使产生，也不会被响应。

可产生待机解除信号，解除深度睡眠模式、睡眠模式。

可屏蔽中断分为外部中断请求和内部中断请求。

(2) 不可屏蔽中断

这是不受屏蔽控制的中断，中断请求一旦产生，CPU 必须响应。

23.2 中断源和结构

中断源列表参照表 23-1。

表 23-1：中断源一览表(1/2)

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		
可屏蔽	0	INTLVI	电压检测[注2]	内部	（A）
	1	INTP0	引脚输入边沿的检测	外部	（B）
	2	INTP1	引脚输入边沿的检测		
	3	INTP2	引脚输入边沿的检测		
	4	INTP3	引脚输入边沿的检测		
	5	INTP4	引脚输入边沿的检测		
	6	INTP5	引脚输入边沿的检测		
	7	INTKR	键中断		
	8	INTRTC	实时时钟的固定周期/闹钟一致检测	内部	（A）
	9	INTIT	间隔信号的检测		
	10	INTST0/ INTCSI00/ INTIIC00	UART0 发送的传送结束或者缓冲器空中断/SSPI00 的传送结束或者缓冲器空中断/IIC00 的传送结束		
	11	INTSR0/ INTCSI01/ INTIIC01	UART0 接收的传送结束/SSPI01 的传送结束或者缓冲器空中断/IIC01 的传送结束		
	12	INTST1/ INTCSI10/ INTIIC10	UART1 发送的传送结束或者缓冲器空中断/SSPI10 的传送结束或者缓冲器空中断/IIC10 的传送结束		
	13	INTSR1/ INTCSI11/ INTIIC11	UART1 接收的传送结束/SSPI11 的传送结束或者缓冲器空中断/IIC11 的传送结束		
	14	INTST2/ INTCSI20/ INTIIC20	UART2 发送的传送结束或者缓冲器空中断/SSPI20 的传送结束或者缓冲器空中断/IIC20 的传送结束		
	15	INTSR2/ INTCSI21/ INTIIC21	UART2 接收的传送结束/SSPI21 的传送结束或者缓冲器空中断/IIC21 的传送结束		

注 1：基本构成类型(A)~(C)分别对应图 23-1 的(A)~(C)。

注 2：这是将电压检测电平寄存器（LVIS）的 bit7（LVIMD）置“0”的情况。

表 23-1 中断源一览表(2/2)

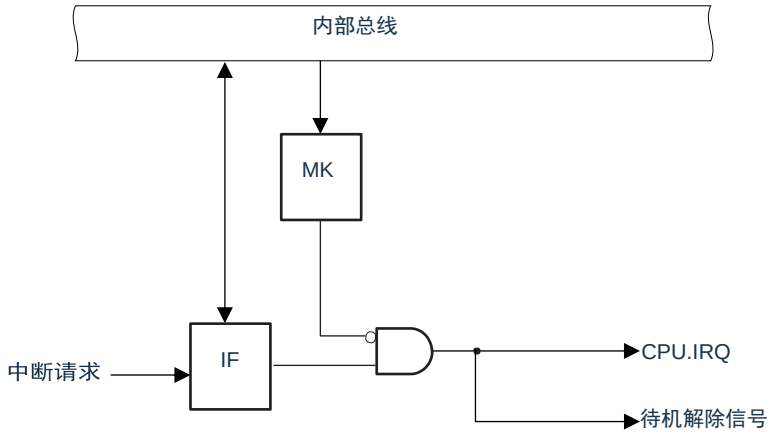
中断处理	中断源编号	中断源		内部/外部	基本结构类型[注1]
		名称	触发		
可屏蔽	16	INTIICA0	IICA0 通信结束	内部	(A)
	17	INTSPI00	SPI 通信结束		
	18	INTTM00	定时器通道 00 的计数结束或捕捉结束		
	19	INTTM01	定时器通道 01 的计数结束或捕捉结束		
	20	INTTM02	定时器通道 02 的计数结束或捕捉结束		
	21	INTTM03	定时器通道 03 的计数结束或捕捉结束		
	22	INTTM04	定时器通道 04 的计数结束或捕捉结束		
	23	INTTM05	定时器通道 05 的计数结束或捕捉结束		
	24	INTTM06	定时器通道 06 的计数结束或捕捉结束		
	25	INTTM07	定时器通道 07 的计数结束或捕捉结束		
	26	INTTMA0	TimerA 下溢		
	27	INTAD	AD 转换结束		
	28	INTCMP0	比较器检测 0		
	29	INTCMP1	比较器检测 1		
	30	INTOSDC	OSDC 停振检测		
	31	INTFL	FLASH 编程结束		
不可屏蔽	-	INTWDT	看门狗定时器间隔中断[注2]	内部	（C）

注 1：基本构成类型(A)~(C)分别对应图 23-1 的(A)~(C)。

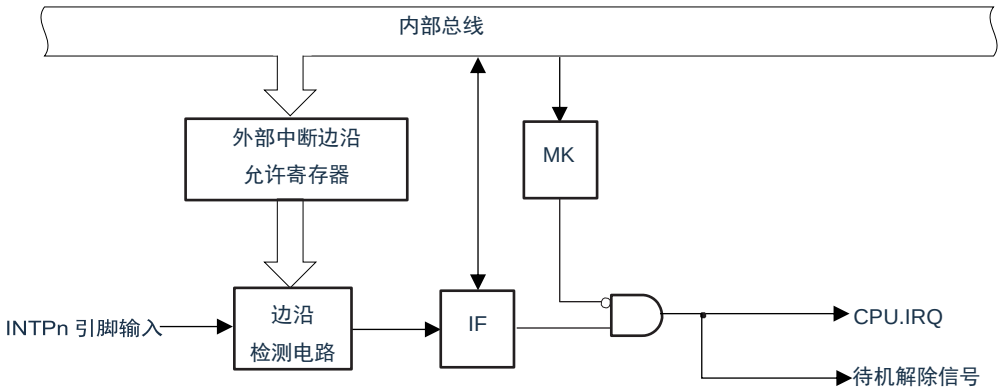
注 2：这是将选项字节（000C0H）的 bit7（WDTINT）置“1”的情况。

图 23-1：中断功能的基本结构

(7) 内部可屏蔽中断

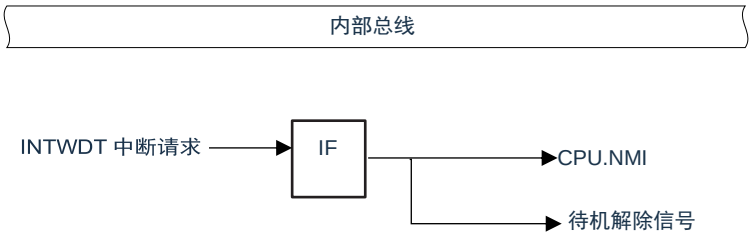


(B)外部可屏蔽中断（INTPn）



备注：n=0~5

I 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。

23.3 控制中断功能的寄存器

通过以下 4 种寄存器控制中断功能。

- 中断请求标志寄存器（IF00~IF31）
- 中断屏蔽标志寄存器（MK00~MK31）
- 外部中断上升沿允许寄存器（EGP0）
- 外部中断下降沿允许寄存器（EGN0）

23.3.1 中断请求标志寄存器（IF00~IF31）

通过发生对应的中断请求或者执行指令，将中断请求标志置“1”。

通过产生复位信号或者执行指令，将中断请求标志清“0”。

通过 8 位存储器操作指令设定 IF00L~IF31L 寄存器

或通过 32 位存储器操作指令设定 IF00~IF31 寄存器。

在产生复位信号后，这些寄存器的值变为“0000_0000H”。

图 23-2：中断请求标志寄存器（IFm）的格式（m=0~31）

地址：IF00：40006000H，IF01：40006004H，IF02：40006008H，IF03：4000600CH
IF04：40006010H，IF05：40006014H，IF06：40006018H，IF07：4000601CH
IF08：40006020H，IF09：40006024H，IF10：40006028H，IF11：4000602CH
IF12：40006030H，IF13：40006034H，IF14：40006038H，IF15：4000603CH
IF16：40006040H，IF17：40006044H，IF18：40006048H，IF19：4000604CH
IF20：40006050H，IF21：40006054H，IF22：40006058H，IF23：4000605CH
IF24：40006060H，IF25：40006064H，IF26：40006068H，IF27：4000606CH
IF28：40006070H，IF29：40006074H，IF30：40006078H，IF31：4000607CH

复位值：0000_0000H　　R/W

	Reserved							
	23	22	21	20	19	18	17	16
	Reserved							
	15	14	13	12	11	10	9	8
	Reserved							
	7	6	5	4	3	2	1	0
IFmL	Reserved							IF

IFmL	编号 0~31 的中断源的中断请求标志
0	不产生中断请求信号。
1	产生中断请求，处于中断请求状态。

注 1：中断源与中断请求标志寄存器的对应关系见表 23-2.

注 2：中断请求标志寄存器与 CPU.IRQ 的对应关系见图 23-4.

23.3.2 中断屏蔽标志寄存器（MK00~MK31）

中断屏蔽标志设定允许或者禁止对应的可屏蔽中断处理。

通过 8 位存储器操作指令设定 MK00L~MK31L 寄存器或通过 32 位存储器操作指令设定 MK00~MK31 寄存器。

在产生复位信号后，这些寄存器的值变为“FFFF_FFFF”。

图 23-3：中断请求屏蔽寄存器（MKm）的格式（m=0~31）

地址：MK00：40006100H，MK01：40006104H，MK02：40006108H，MK03：4000610CH

MK04：40006110H，MK05：40006114H，MK06：40006118H，MK07：4000611CH

MK08：40006120H，MK09：40006124H，MK10：40006128H，MK11：4000612CH

MK12：40006130H，MK13：40006134H，MK14：40006138H，MK15：4000613CH

MK16：40006140H，MK17：40006144H，MK18：40006148H，MK19：4000614CH

MK20：40006150H，MK21：40006154H，MK22：40006158H，MK23：4000615CH

MK24：40006160H，MK25：40006164H，MK26：40006168H，MK27：4000616CH

MK28：40006170H，MK29：40006174H，MK30：40006178H，MK31：4000617CH

复位值：FFFF_FFFFH　　R/W

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

MKmL	编号 0~31 的中断源的中断处理控制[注 1]
0	允许中断处理。
1	禁止中断处理。

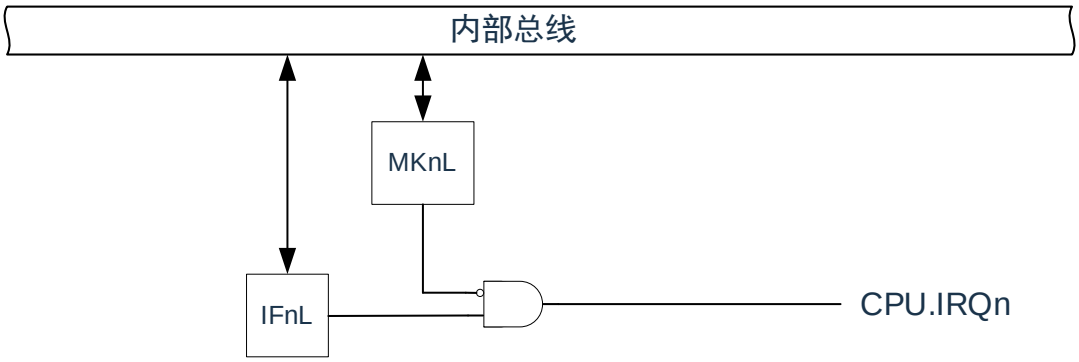
注 1：中断源与中断请求屏蔽寄存器的对应关系见表 23-2.

注 2：中断请求屏蔽寄存器与 CPU.IRQ 的对应关系见图 23-4.

表 23-2：中断源和各标志寄存器的对应关系

编号	中断源	中断请求标志寄存器	中断屏蔽标志寄存器
0	INTLVI	IF00.IFL	MK00.MKL
1	INTP0	IF01.IFL	MK01.MKL
2	INTP1	IF02.IFL	MK02.MKL
3	INTP2	IF03.IFL	MK03.MKL
4	INTP3	IF04.IFL	MK04.MKL
5	INTP4	IF05.IFL	MK05.MKL
6	INTP5	IF06.IFL	MK06.MKL
7	INTKR	IF07.IFL	MK07.MKL
8	INTRTC	IF08.IFL	MK08.MKL
9	INTIT	IF09.IFL	MK09.MKL
10	INTST0/INTCSI00/INTIIC00	IF10.IFL	MK10.MKL
11	INTSR0/INTCSI01/INTIIC01	IF11.IFL	MK11.MKL
12	INTST1/INTCSI10/INTIIC10	IF12.IFL	MK12.MKL
13	INTSR1/INTCSI11/INTIIC11	IF13.IFL	MK13.MKL
14	INTST2/INTCSI20/INTIIC20	IF14.IFL	MK14.MKL
15	INTSR2/INTCSI21/INTIIC21	IF15.IFL	MK15.MKL
16	INTIICA0	IF16.IFL	MK16.MKL
17	INTSPI00	IF17.IFL	MK17.MKL
18	INTTM00	IF18.IFL	MK18.MKL
19	INTTM01	IF19.IFL	MK19.MKL
20	INTTM02	IF20.IFL	MK20.MKL
21	INTTM03	IF21.IFL	MK21.MKL
22	INTTM04	IF22.IFL	MK22.MKL
23	INTTM05	IF23.IFL	MK23.MKL
24	INTTM06	IF24.IFL	MK24.MKL
25	INTTM07	IF25.IFL	MK25.MKL
26	INTTMJ0	IF26.IFL	MK26.MKL
27	INTAD	IF27.IFL	MK27.MKL
28	INTCMP0	IF28.IFL	MK28.MKL
29	INTCMP1	IF29.IFL	MK29.MKL
30	INTOSDC	IF30.IFL	MK30.MKL
31	INTFL	IF31.IFL	MK31.MKL

图 23-4：各标志寄存器与 CPU.IRQ 的关系



23.3.3 外部中断上升沿允许寄存器（EGP0）、外部中断下降沿允许寄存器（EGN0）

这些寄存器设定 INTP0~INTP5 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”。

图 23-5：外部中断上升沿允许寄存器（EGP0）和外部中断下降沿允许寄存器（EGN0）的格式

地址：40041C00H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址：40041C01H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn引脚的有效边沿选择（n=0~5）
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 23-3 所示。

表 23-3：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

注意：如果将外部中断功能使用的输入端口切换到输出模式，就可能检测到有效边沿而产生 INTPn 中断。当切换到输出模式时，必须在禁止检测边沿后（EGPn、EGNn=0、0）将端口模式寄存器（PMxx）置“0”。

备注：

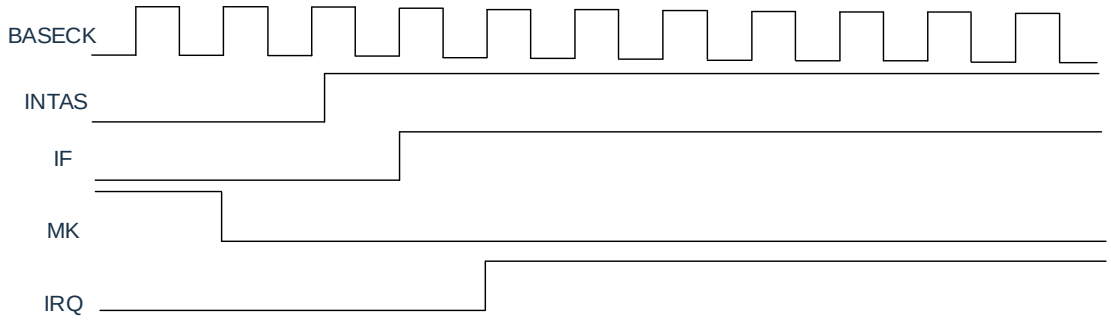
1. 有关边沿检测的端口，请参照“”。
2. n=0~5

23.4 中断处理的操作

23.4.1 可屏蔽中断请求的接受

如果中断请求标志被置“1”并且该中断请求的屏蔽（MK）标志已被清“0”，就进入能接受可屏蔽中断请求的状态，可以将中断请求传递给 NVIC。

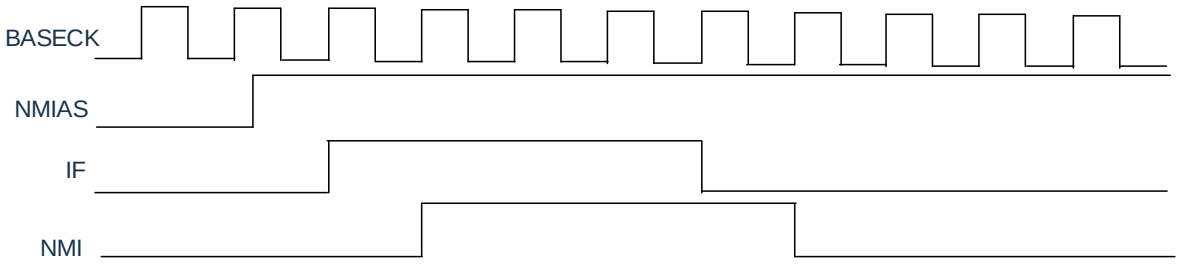
从中断请求标志被置 1，到 CPU 的 IRQ 被置 1，只需要 1 个时钟。



23.4.2 不可屏蔽中断请求的接受

如果产生不可屏蔽中断请求，中断请求标志将会被置“1”，并直接传递给 NVIC。

从中断请求标志被置 1，到 CPU 的 NMI 被置 1，只需要 1 个时钟。



第24章 键中断功能

键中断输入的通道数因产品而不同。

24.1 键中断的功能

能通过给键中断输入引脚（KR0~KR7）输入下降沿，产生键中断（INTKR）。

表 24-1：键中断检测引脚的分配

键中断引脚	键返回模式寄存器（KRM）
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

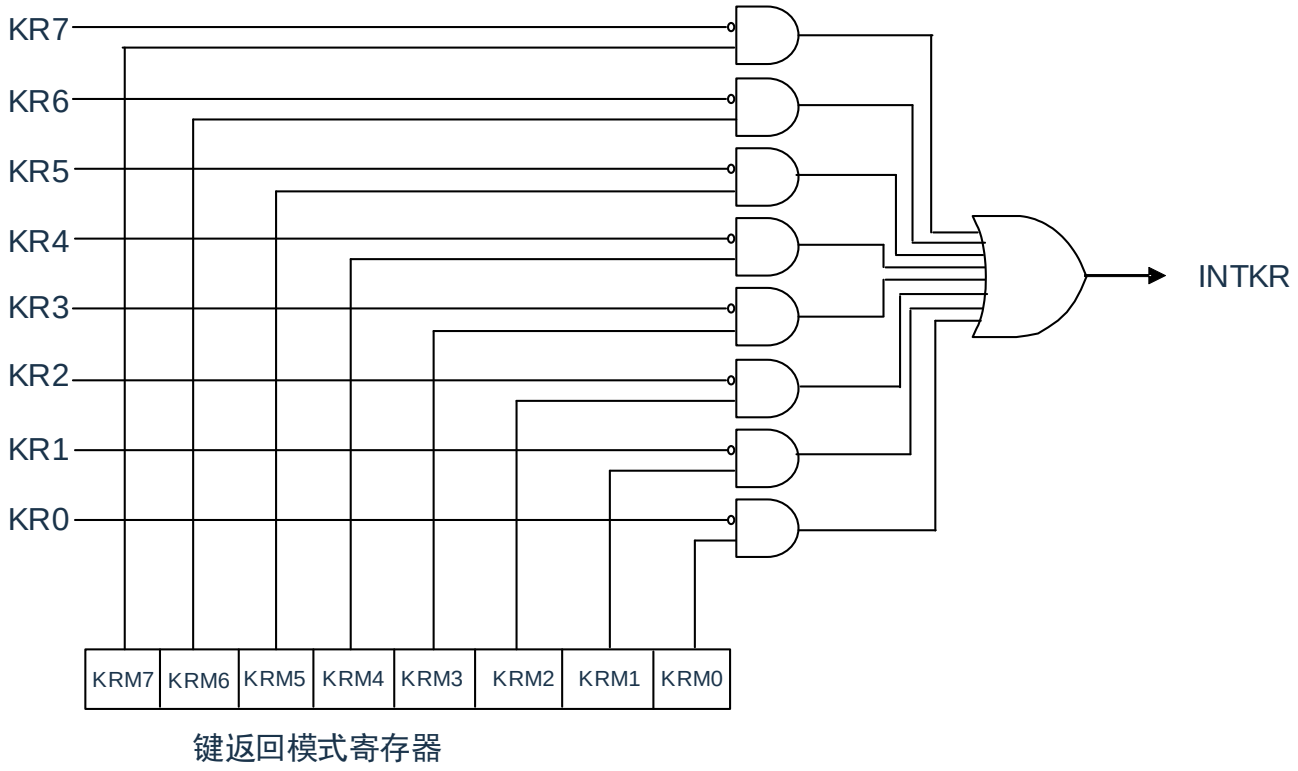
24.2 键中断的结构

键中断由以下硬件构成。

表 24-2：键中断的结构

项目	控制寄存器
控制寄存器	键返回模式寄存器（KRM）端口模式寄存器（PMx） 端口模式控制寄存器（PMCx）

图 24-1：键中断的框图


KR7
KR6
KR5
KR4
KR3
KR2
KR1
KR0
INTKR
KRM7
KRM6
KRM5
KRM4
KRM3
KRM2
KRM1
KRM0
键返回模式寄存器

24.3 控制键中断的寄存器

通过以下寄存器控制键中断功能。

- 键返回模式寄存器（KRM）
- 端口模式寄存器（PMx）

24.3.1 键返回模式寄存器（KRM）

KRM0~KRM7 位控制 KR0~KR7 信号。

通过 8 位存储器操作指令设定 KRM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 24-2：键返回模式寄存器（KRM）的格式

地址：40042001H　　复位后：00H　R/W

符号	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	键中断模式的控制
0	不检测键中断信号。
1	检测键中断信号。

注意：

1. 能通过将键中断输入引脚的上拉电阻寄存器（PUx）的对象位置“1”，使用内部上拉电阻。
2. 如果在给键中断输入引脚输入低电平的状态下将 KRM 寄存器的对象位置位，就产生中断。要想忽视此中断时，必须在通过中断屏蔽标志禁止中断处理后设定 KRM 寄存器。然后，必须在等待键中断输入的低电平宽度（T_{KR}）（请参照数据手册）后清除中断请求标志，允许中断处理。
3. 在键中断模式中未使用的引脚能用作通常的端口。

备注：n=0~7。

第25章 待机功能

25.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下 2 种模式。

(1) 睡眠模式

睡眠模式是停止CPU运行时钟的模式。在设置睡眠模式前，如果高速系统时钟振荡电路、高速内部振荡器或者副系统时钟振荡电路正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到深度睡眠模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

(2) 深度睡眠模式

深度睡眠模式是停止高速系统时钟振荡电路和高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小CPU的工作电流。

因为深度睡眠模式能通过中断请求来解除，所以也能进行间歇运行。但是，在X1时钟的情况下，因为在解除深度睡眠模式时需要确保振荡稳定的等待时间，所以如果需要通过中断请求立即开始处理，就必须选择睡眠模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设置为待机模式前的内容，并且还保持输入/输出端口的输出锁存器和输出缓冲器的状态。

注意：

1. 只有在 CPU 以主系统时钟运行时才能使用深度睡眠模式。当 CPU 以副系统时钟运行时，不能设置为深度睡眠模式。无论 CPU 是以主系统时钟还是以副系统时钟运行，都能使用睡眠模式。
2. 在转移到深度睡眠模式时，必须在停止以主系统时钟运行的外围硬件后执行 WFI 指令。
3. 为了减小 A/D 转换器的工作电流，必须将 A/D 转换器模式寄存器 0（ADM0）的 bit7（ADCS）和 bit0（ADCE）清“0”，在停止 A/D 转换运行后执行 WFI 指令。
4. 能通过选项字节选择在睡眠模式或者深度睡眠模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“第 31 章 选项字节”。

25.2 睡眠模式

25.2.1 睡眠模式的设置

在 SCR 寄存器的 SLEEPDEEP 位为 0 时，执行 WFI 指令，就进入了睡眠模式。在睡眠模式，CPU 停止动作，但是内部寄存器的值仍被保持，周边模块也保持进入睡眠模式之前的状态。周边模块，发振器等在睡眠模式下的状态见表 25-1.

无论设置前的 CPU 时钟是高速系统时钟还是高速内部振荡器时钟或者副系统时钟，都能设置睡眠模式。

注意：当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除睡眠模式。因此，即使在此情况下执行 WFI 指令，也不转移到睡眠模式。

表 25-1：睡眠模式中的运行状态(1/2)

项目 \ 睡眠模式的设置			在CPU以主系统时钟运行的过程中执行WFI指令的情况		
			CPU以高速内部振荡器时钟（F_{IH}）运行	CPU以X1时钟（F_X）运行	CPU以外部主系统时钟（F_{EX}）运行
系统时钟			停止给CPU提供时钟。		
	主系统时钟	F_{IH}	继续运行（不能停止）。	禁止运行。	
		F_X	禁止运行。	继续运行（不能停止）。	不能运行。
		F_{EX}		不能运行。	继续运行（不能停止）。
	副系统时钟	F_{XT}	保持设置为睡眠模式前的状态。		
		F_{EXS}			
	低速内部振荡器时钟	F_{IL}	通过选项字节（000C0H）的bit0（WDSTBYON）和bit4（WDTON）以及副系统时钟提供模式控制寄存器（OSMC）的WUTMMCK0位进行设置。 WUTMMCK0=1：振荡 WUTMMCK0=0并且WDTON=0：停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1：振荡 WUTMMCK0=0、WDTON=1并且WDSTBYON=0：停止		
CPU			停止运行。		
代码闪存					
RAM			停止运行（在执行DMA时，能运行）。		
端口（锁存器）			保持设置为睡眠模式前的状态。		
DIV			能运行。		
Timer8					
实时时钟（RTC）					
15位间隔定时器					
看门狗定时器			参照“第10章 看门狗定时器”。		
定时器A			能运行。		
时钟输出/蜂鸣器输出					
A/D转换器					
D/A转换器					
比较器					
通用串行通信单元（SCI）					
串行接口（IICA）					
aFCAN					
数据传送控制器（DMA）					
联动控制器			能在可运行的功能块之间进行链接。		
上电复位功能			能运行。		
电压检测功能					
外部中断					
键中断功能					
CRC运算功能	高速CRC				
	通用CRC		在RAM区的运算中执行DMA时，能运行。		
RAM奇偶校验功能			在执行DMA时，能运行。		
SFR保护功能					

备注：停止运行：在转移到睡眠模式时自动停止运行。

禁止运行：在转移到睡眠模式前停止运行。

F_{IH}：高速内部振荡器时钟　F_{IL}：低速内部振荡器时钟

F_X：X1 时钟　F_{EX}：外部主系统时钟

F_{XT}：XT1 时钟　F_{EXS}：外部副系统时钟

表 25-1：睡眠模式中的运行状态(2/2)

项目 \ 睡眠模式的设置			在CPU以副系统时钟运行的过程中执行WFI指令的情况	
			CPU以XT1时钟（F_{XT}）运行	CPU以外部副系统时钟（F_{EXS}）运行
系统时钟			停止给CPU提供时钟。	
	主系统时钟	F_{IH}	禁止运行。	
		F_X		
		F_{EX}		
	副系统时钟	F_{XT}	继续运行（不能停止）。	不能运行。
		F_{EXS}	不能运行。	继续运行（不能停止）。
	低速内部振荡器时钟	F_{IL}	通过选项字节（000C0H）的bit0（WDSTBYON）和bit4（WDTON）以及副系统时钟提供模式控制寄存器（OSMC）的WUTMMCK0位进行设置。 • WUTMMCK0=1：振荡 • WUTMMCK0=0并且WDTON=0：停止 • WUTMMCK0=0、WDTON=1并且WDSTBYON=1：振荡 • WUTMMCK0=0、WDTON=1并且WDSTBYON=0：停止	
CPU			停止运行。	
代码闪存				
RAM			停止运行（在执行DMA时，能运行）。	
端口（锁存器）			保持设置为睡眠模式前的状态。	
DIV			当RTCLPC=0时，能运行（否则禁止运行）。	
Time4			当RTCLPC=0时，能运行（否则禁止运行）。	
实时时钟（RTC）			能运行。	
15位间隔定时器				
看门狗定时器			参照“第10章 看门狗定时器”。	
定时器A			当RTCLPC=0时，能运行（否则禁止运行）。	
时钟输出/蜂鸣器输出				
A/D转换器			禁止运行。	
D/A转换器			当RTCLPC=0时，能运行（否则禁止运行）。	
比较器			当比较器的基准电压选择外部输入（IVREFn）时，能运行。	
通用串行通信单元（SCI）			当RTCLPC=0时，能运行（否则禁止运行）。	
串行接口（IICA）			禁止运行。	
aFCAN			当RTCLPC=0时，能运行（否则禁止运行）。	
数据传送控制器（DMA）			当RTCLPC=0时，能运行（否则禁止运行）。	
联动控制器			能在可运行的功能块之间进行链接。	
上电复位功能			能运行。	
电压检测功能				
外部中断				
键中断功能				
CRC运算功能	高速CRC		禁止运行。	
	通用CRC		在RAM区的运算中执行DMA时，能运行。	
RAM奇偶校验错误检测功能			在执行DMA时，能运行。	
SFR保护功能				

备注：停止运行：在转移到睡眠模式时自动停止运行。

禁止运行：在转移到睡眠模式前停止运行。

F_{IH}：高速内部振荡器时钟　F_{IL}：低速内部振荡器时钟

F_X：X1 时钟　F_{EX}：外部主系统时钟

F_{XT}：XT1 时钟　F_{EXS}：外部副系统时钟

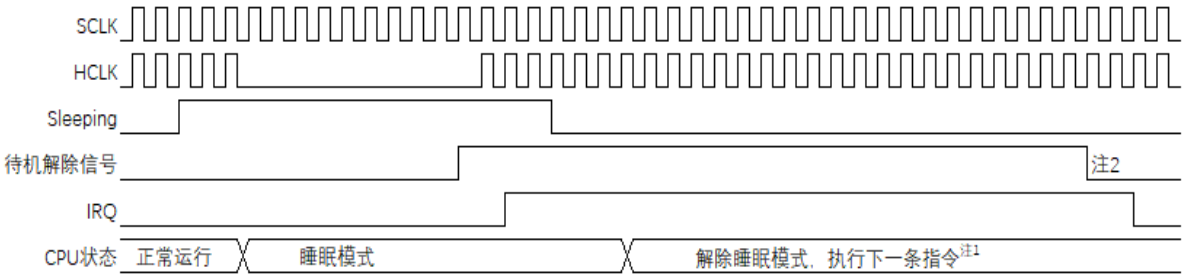
25.2.2 睡眠模式的解除

睡眠模式可以被任意中断以及外部复位端子，POR 复位，低电压检测复位，RAM 奇偶校验错误复位，WDT 复位，软件复位解除。

(1) 通过中断解除

当产生一个未屏蔽的中断，且处于允许接受中断的状态时，睡眠模式就被解除，CPU开始处理中断服务程序。

图 25-1：通过中断请求解除睡眠模式



注 1：从待机解除信号产生到睡眠模式解除，开始执行中断服务程序，需要 16 个时钟。

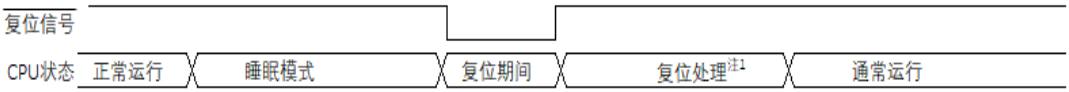
注 2：待机解除信号不能自己清除，必须写寄存器清除。通常是在中断服务程序中写寄存器清除。

注意：进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。

(2) 通过复位解除

当有复位信号产生时，CPU处于复位状态，睡眠模式被解除。和通常的复位相同，在转移到复位向量地址后执行程序。

图 25-2：通过复位解除睡眠模式



注 1：有关复位处理时间，请参照“第 26 章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第 27 章 上电复位电路”。

25.3 深度睡眠模式

25.3.1 深度睡眠模式的设置

在 SCR 寄存器的 SLEEPDEEP 位为 1 时，执行 WFI 指令，就进入了深度睡眠模式。在这个模式，CPU，大多数的周边模块，以及发振器都停止运行。但是，CPU 内部寄存器的值，RAM 数据，周边模块，I/O 的状态被保持。周边模块，发振器在深度睡眠模式的运行状态见表 24-2.

只有在设置前的 CPU 时钟为主系统时钟的情况下才能设置深度睡眠模式。

注意：当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除深度睡眠模式。因此，如果在此情况下执行 WFI 指令，就在一旦进入深度睡眠模式后立即被解除。在执行 WFI 指令并且经过深度睡眠模式解除时间后返回到运行模式。

表 25-2：深度睡眠模式中的运行状态

项目 \ 深度睡眠模式的设置			在CPU以主系统时钟运行的过程中执行WFI指令的情况		
			CPU以高速内部振荡器时钟（F_{IH}）运行	CPU以X1时钟（F_X）运行	CPU以外部主系统时钟（F_{EX}）运行
系统时钟			停止给CPU提供时钟。		
	主系统时钟	F_{IH}	停止		
		F_X			
		F_{EX}			
	副系统时钟	F_{XT}	保持设置为深度睡眠模式前的状态。		
		F_{EXS}			
	低速内部振荡器时钟	F_{IL}	通过选项字节（000C0H）的bit0（WDSTBYON）和bit4（WDTON）以及副系统时钟提供模式控制寄存器（OSMC）的WUTMMCK0位进行设置。 WUTMMCK0=1：振荡 WUTMMCK0=0并且WDTON=0：停止 WUTMMCK0=0、WDTON=1并且WDSTBYON=1：振荡WUTMMCK0=0、WDTON=1并且WDSTBYON=0：停止		
CPU			停止运行。		
代码闪存					
RAM					
端口（锁存器）			保持设置为深度睡眠模式前的状态。		
DIV			禁止运行。		
定时器阵列单元			禁止运行。		
实时时钟（RTC）			能运行。		
15位间隔定时器					
看门狗定时器			参照“第10章 看门狗定时器”。		
定时器A			• 在选择没有TAIO输入滤波器时的事件计数模式中，能运行。 • 在选择副系统时钟作为计数源并且OSMC寄存器的RTCLPC位为“0”时，能运行。 • 在选择低速内部振荡器作为计数源时，能运行。 • 上述以外：禁止运行。		
时钟输出/蜂鸣器输出			在选择副系统时钟作为计数时钟并且RTCLPC位为“0”时，能运行（否则禁止运行）。		
A/D转换器			能进行唤醒。		
D/A转换器			能运行（保持设置为深度睡眠模式前的状态）。		
比较器			能运行（只限于不使用数字滤波器并且比较器的基准电压选择外部输入（IVREFn）的情况）。		
通用串行通信单元（SCI）			只有SSPIp和UARTq才能唤醒。 除了SSPIp和UARTq以外，禁止运行。		
串行阵列单元（IICA）			能通过地址匹配进行唤醒。		
aFCAN			禁止运行。		
数据传送控制器（DMA）			能接受DMA启动源。		
联动控制器			能在可运行的功能块之间进行链接。		
上电复位功能			能运行。		
电压检测功能					
外部中断					
键中断功能					
CRC运算功能	高速CRC		停止运行。		
	通用CRC				
RAM奇偶校验功能					
SFR保护功能					

备注：停止运行：在转移到深度睡眠模式时自动停止运行。

禁止运行：在转移到深度睡眠模式前停止运行。

F_{IH}：高速内部振荡器时钟　F_{IL}：低速内部振荡器时钟

F_X：X1 时钟　F_{EX}：外部主系统时钟

F_{XT}：XT1 时钟　F_{EXS}：外部副系统时钟

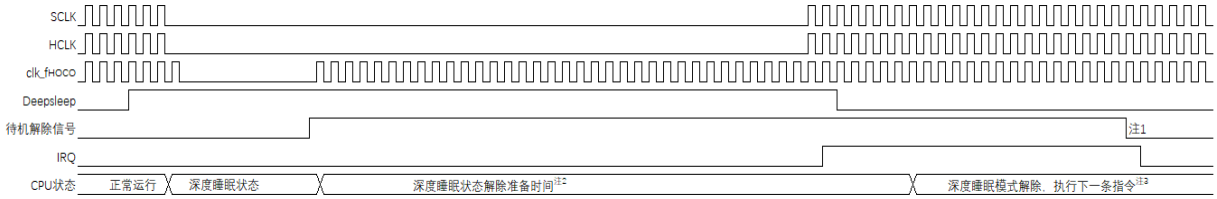
25.3.2 深度睡眠模式的解除

能通过以下 2 种方法解除深度睡眠模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除深度睡眠模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图 25-3：通过中断请求解除深度睡眠模式



注 1：待机解除信号：有关待机解除信号的详细内容，请参照“图 23-1 中断功能的基本结构”。

注 2：深度睡眠状态解除准备时间：

进入深度睡眠模式前 CPU 时钟为高速内部振荡时钟或外部时钟输入时：至少 20us

进入深度睡眠模式前 CPU 时钟为高速系统时钟(X1 振荡)时：

至少 20us 和振荡稳定时间(通过 OSTS 进行设置)中较长的时间

注 3：等待：从 CPU.IRQ 有效到开始执行中断服务程序，需要 14 个时钟。

注意：

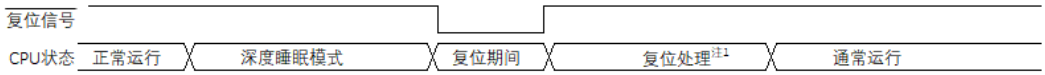
1. 进入睡眠模式前，应只将期待用来解除睡眠模式的中断对应的屏蔽位清零。
2. 在 CPU 以高速系统时钟（X1 振荡）运行并且要缩短深度睡眠模式解除后的振荡稳定时间时，必须在执行 WFI 指令前暂时将 CPU 时钟切换为高速内部振荡器时钟。

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

(b) 通过产生复位信号进行的解除

通过产生复位信号来解除深度睡眠模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图 25-4：通过复位解除深度睡眠模式



注：有关复位处理时间，请参照“第 26 章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第 27 章 上电复位电路”。

第26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) 通过看门狗定时器的程序失控检测产生内部复位。

(3) 通过上电复位(POR)电路的电源电压和检测电压的比较产生内部复位。

(4) 通过电压检测电路(LVD)的电源电压和检测电压的比较产生内部复位。

(5) 因系统复位请求寄存器位(AIRCR.SYSRESETREQ)被置为 1 而产生内部复位。

(6) 因 RAM 奇偶校验错误而产生内部复位。

(7) 因存取非法存储器而产生内部复位。

(8) 停振检测功能选择复位模式且检测到停振而产生内部复位。

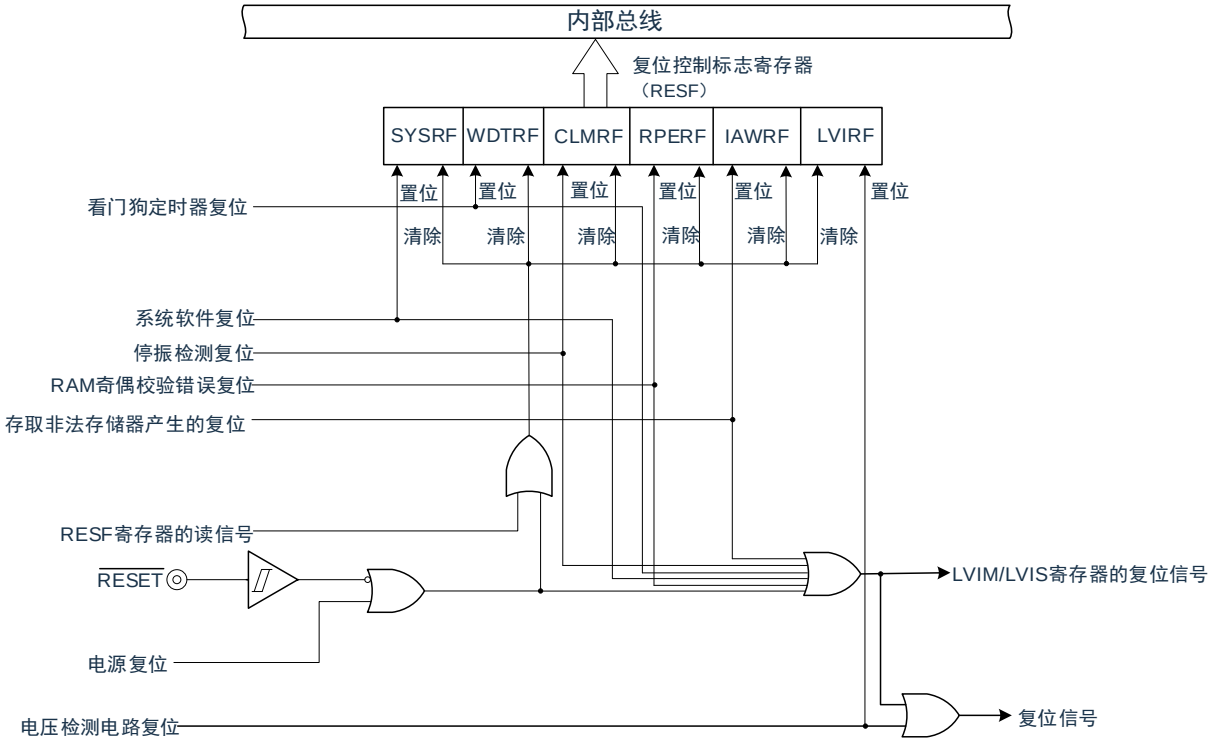
内部复位和外部复位相同，在产生复位信号后，从用户自定义的程序起始地址开始执行程序。

当给 RESETB 引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者系统复位请求位被置位，或者发生 RAM 奇偶检验错误，或者存取非法存储器，或者检测到停振时，产生复位并且各硬件变为如表 25-1 所示的状态。

注意：

1. 在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在 datasheet 的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。
2. 在复位信号发生期间，停止 X1 时钟、XT1 时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟和外部副系统时钟的输入无效。

图 26-1：复位功能的框图



注意：LVD 电路的内部复位不会对 LVD 电路进行复位。

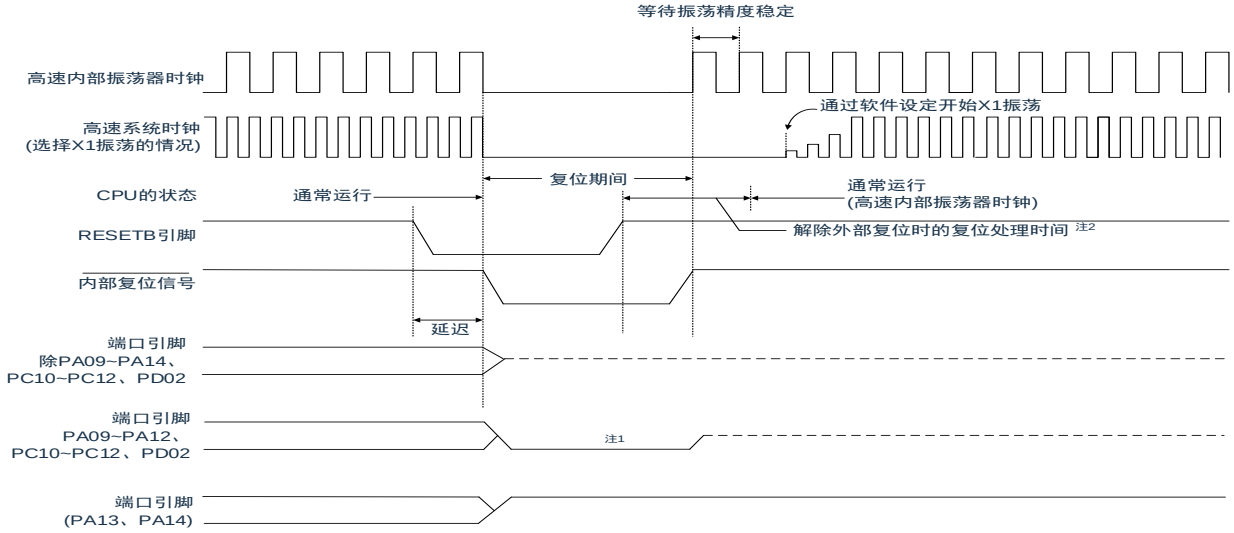
备注：

1. LVIM：电压检测寄存器
2. LVIS：电压检测电平寄存器

26.1 复位时序

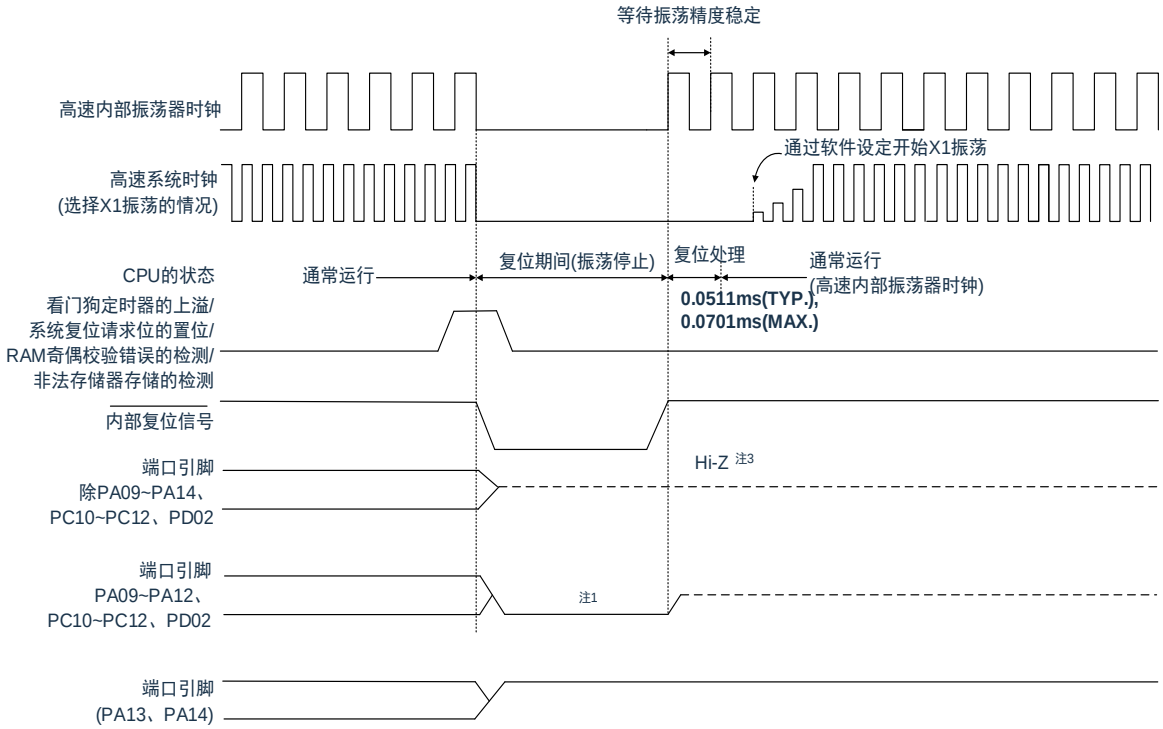
当给 RESETB 引脚输入低电平时，产生复位。然后，如果给 RESETB 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 26-2：RESETB 输入的复位时序



对于因看门狗定时器的上溢、系统复位请求位的置位、RAM 奇偶校验错误的检测或者非法存储器存取的检测或者停振检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 26-3：因看门狗定时器的上溢、系统复位请求位的置位、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位时序



注 1：解除外部复位时的复位处理时间：

解除 POR 后的第 1 次：

0.672ms(TYP.)、0.832ms(MAX.)（使用 LVD 的情况）

0.399ms(TYP.)、0.519ms(MAX.)（不使用 LVD 的情况）

解除 POR 后的第 2 次以后：

0.531ms(TYP.)、0.675ms(MAX.)（使用 LVD 的情况）

0.259ms(TYP.)、0.362ms(MAX.)（不使用 LVD 的情况）

当电源电压上升时，在解除外部复位时的复位处理时间之前需要电压稳定等待时间 0.99ms(TYP.)、2.30ms(MAX.)。

注 2：端口引脚 PA13、PA14 变为以下状态：

- 在外部复位或者 POR 复位的期间为高阻抗。
- 在其他复位期间以及接受复位后为高电平（连接内部上拉电阻）。

注意：看门狗定时器也不例外，在发生内部复位时进行复位。

对于由 POR 电路和 LVD 电路的电压检测产生的复位，如果在复位后满足 $V_{DD} \geqslant V_{POR}$ 或者 $V_{DD} \geqslant V_{LVD}$，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 27 章 上电复位电路”和“第 28 章 电压检测电路”。

备注：V_{POR}：POR 电源电压上升检测电压

V_{LVD}：LVD 检测电压

表 26-1：复位期间的运行状态

<table>
<tr><th colspan="3">项目</th><th>复位期间</th></tr>
<tr><td colspan="3">系统时钟</td><td>停止给CPU提供时钟。</td></tr>
<tr><td rowspan="6"></td><td rowspan="3">主系统时钟</td><td>F_{IH}</td><td>停止运行。</td></tr>
<tr><td>F_X</td><td>停止运行（X1引脚和X2引脚处于输入端口模式）。</td></tr>
<tr><td>F_{EX}</td><td>时钟输入无效（引脚处于输入端口模式）。</td></tr>
<tr><td rowspan="2">副系统时钟</td><td>F_{XT}</td><td>能运行。</td></tr>
<tr><td>F_{EXS}</td><td>时钟输入无效（引脚处于输入端口模式）。</td></tr>
<tr><td>低速内部振荡器时钟</td><td>F_{IL}</td><td>停止运行。</td></tr>
<tr><td colspan="3">CPU</td><td></td></tr>
<tr><td colspan="3">代码闪存</td><td>停止运行。</td></tr>
<tr><td colspan="3">RAM</td><td>停止运行。</td></tr>
<tr><td colspan="3">端口（锁存器）</td><td>高阻抗[注1]</td></tr>
<tr><td colspan="3">DIV</td><td rowspan="3">停止运行。</td></tr>
<tr><td colspan="3">定时器阵列单元</td></tr>
<tr><td colspan="3">定时器A</td></tr>
<tr><td colspan="3">实时时钟（RTC）</td><td>在POR复位时，停止运行。在其他复位时，能运行。</td></tr>
<tr><td colspan="3">15位间隔定时器</td><td rowspan="10">停止运行。</td></tr>
<tr><td colspan="3">看门狗定时器</td></tr>
<tr><td colspan="3">时钟输出/蜂鸣器输出</td></tr>
<tr><td colspan="3">A/D转换器</td></tr>
<tr><td colspan="3">D/A转换器[注1]</td></tr>
<tr><td colspan="3">比较器[注1]</td></tr>
<tr><td colspan="3">通用串行通信单元（SCI）</td></tr>
<tr><td colspan="3">串行接口（IICA）</td></tr>
<tr><td colspan="3">aFCAN</td></tr>
<tr><td colspan="3">数据传送控制器（DMA）</td></tr>
<tr><td colspan="3">上电复位功能</td><td>能进行检测运行。</td></tr>
<tr><td colspan="3">电压检测功能</td><td>在LVD复位时，能运行。在其他复位时，停止运行。</td></tr>
<tr><td colspan="3">外部中断</td><td rowspan="6">停止运行。</td></tr>
<tr><td colspan="3">键中断功能</td></tr>
<tr><td colspan="2" rowspan="2">CRC运算功能</td><td>高速CRC</td></tr>
<tr><td>通用CRC</td></tr>
<tr><td colspan="3">RAM奇偶校验功能</td></tr>
<tr><td colspan="3">SFR保护功能</td></tr>
</table>

备注：F_{IH}：高速内部振荡器时钟　F_X：X1 振荡时钟

F_{EX}：外部主系统时钟　F_{XT}：XT1 振荡时钟

F_{EXS}：外部副系统时钟　F_{IL}：低速内部振荡器时钟

26.2 确认复位源的寄存器

26.2.1 复位控制标志寄存器（RESF）

CMS32H6157 微控制器存在多种内部复位发生源。复位控制标志寄存器（RESF）保存发生复位请求的复位源。能通过 8 位存储器操作指令读 RESF 寄存器。

通过 RESETB 的输入、上电复位（POR）电路的复位和 RESF 寄存器的读取，清除 SYSRF、WDTRF、CLMRF、RPERF、IAWRF、LVIRF 标志。要判断复位源时，必须将 RESF 寄存器的值保存到任意的 RAM，然后通过其 RAM 值进行判断。

图 26-4：复位控制标志寄存器（RESF）的格式

地址：40020440H 复位后：不定值[注1]　　R

符号	7	6	5	4	3	2	1	0
RESF	SYSRF		0	WDTRF	CLMRF	RPERF	IAWRF	LVIRF

SYSRF	系统复位请求位被置位而产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

WDTRF	看门狗定时器（WDT）产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

CLMRF	停振检测功能产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

RPERF	RAM奇偶校验错误产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

IAWRF	存取非法存储器产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

LVIRF	电压检测电路（LVD）产生的内部复位请求
0	没有产生内部复位请求或者清除了RESF寄存器。
1	产生内部复位请求。

注 1：因复位源而不同。请参照表 26-2。

注意：在允许产生 RAM 奇偶校验错误复位(RPERDIS=0)的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区+10 字节”的区域进行初始化。通过产生复位，进入允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的状态。详细内容请参照“28.3.3 RAM 奇偶校验错误检测功能”。

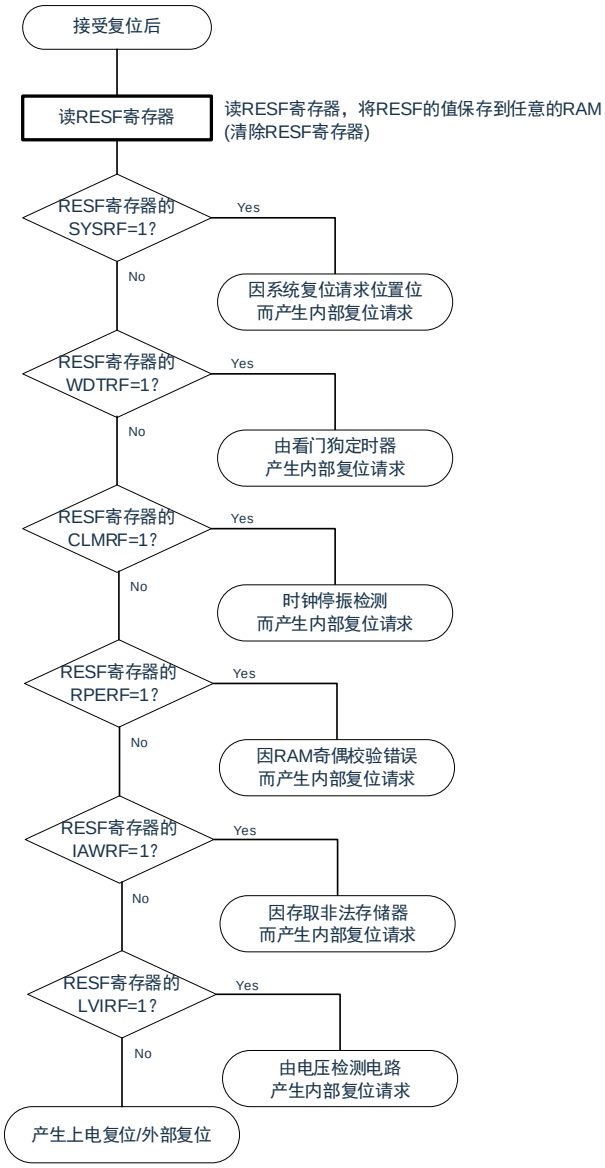
发生复位请求时的 RESF 寄存器状态如表 26-2 所示。

表 26-2：发生复位请求时的 RESF 寄存器状态

<table>
<tr><th>复位源
标志</th><th>RESETB 输入</th><th>POR产生的复位</th><th>系统复位请求位置位产生的复位</th><th>WDT产生的复位</th><th>停振检测产生的复位</th><th>RAM奇偶校验错误产生的复位</th><th>存取非法存储器产生的复位</th><th>LVD产生的复位</th></tr>
<tr><td>SYSRF</td><td rowspan="6">清“0”</td><td rowspan="6">清“0”</td><td>置“1”</td><td>保持</td><td rowspan="2">保持</td><td rowspan="3">保持</td><td rowspan="4">保持</td><td rowspan="5">保持</td></tr>
<tr><td>WDTRF</td><td rowspan="5">保持</td><td>置“1”</td></tr>
<tr><td>CLMRF</td><td rowspan="4">保持</td><td>置“1”</td></tr>
<tr><td>RPERF</td><td rowspan="3">保持</td><td>置“1”</td></tr>
<tr><td>IAWRF</td><td rowspan="2">保持</td><td>置“1”</td></tr>
<tr><td>LVIRF</td><td>保持</td><td>置“1”</td></tr>
</table>

复位源的确认步骤如图所示。

图 26-5：复位源的确认步骤



注意：上述流程是确认步骤的例子。

第27章 上电复位电路

27.1 上电复位电路的功能

上电复位电路（POR）有以下功能。

- 在接通电源时产生内部复位信号。
  如果电源电压（V_{DD}）超过检测电压（V_{POR}），就解除复位。但是，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态。
- 将电源电压（V_{DD}）和检测电压（V_{PDR}）进行比较。当 V_{DD}＜V_{PDR} 时，产生内部复位信号。但是，当电源电压下降时，必须在电源电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注意：当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清“00H”。

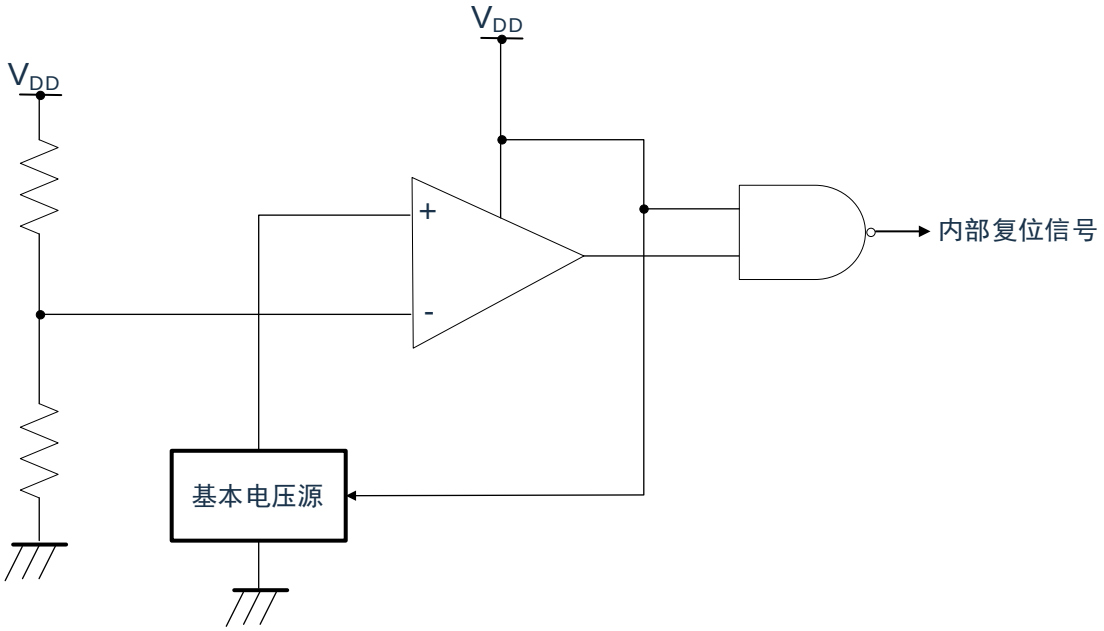
备注：

1. CMS32H6157 内置多个产生内部复位信号的硬件。当由看门狗定时器（WDT）、电压检测（LVD）电路、系统复位请求位置位、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在 RESF 寄存器；当由 WDT、LVD、系统复位请求位的置位、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清“00H”而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 26 章 复位功能”。
2. V_{POR}：POR 电源电压上升检测电压
   V_{PDR}：POR 电源电压下降检测电压
   详细内容请参照数据手册的 POR 电路特性。

27.2 上电复位电路的结构

上电复位电路的框图如图 27-1 所示。

图 27-1：上电复位电路的框图

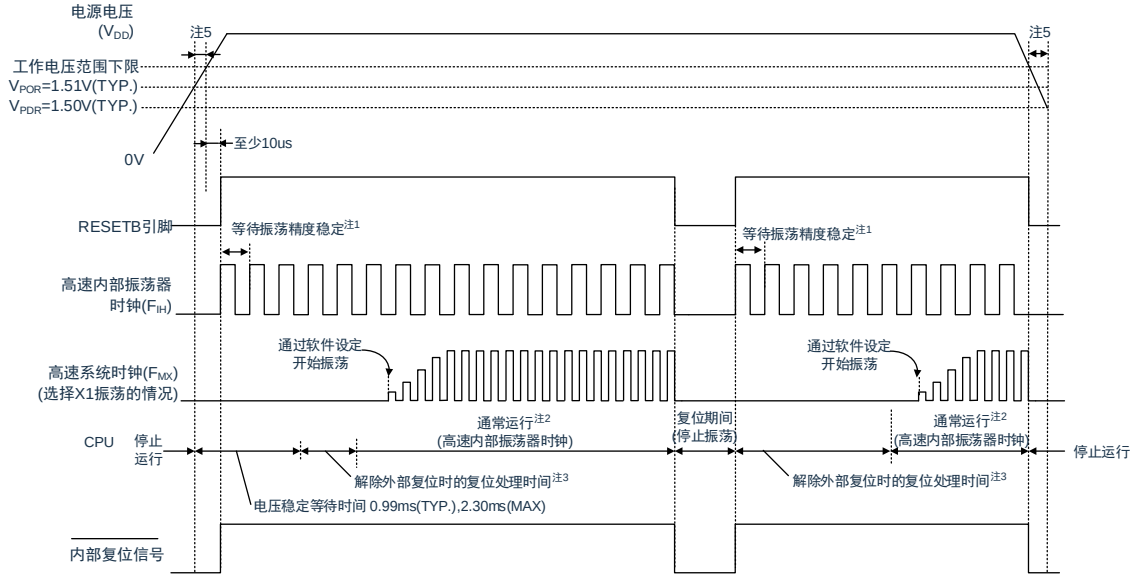


27.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图 27-2：上电复位电路和电压检测电路的内部复位信号的产生时序(1/3)

(1) 使用 RESETB 引脚的外部复位输入的情况



注 1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2：能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注 3：到开始通常运行为止的时间除了达到 V_{POR}（1.51V(TYP.)）后的“电压稳定等待时间”以外，在将 RESETB 信号置为高电平（“1”）后还需要以下的“解除外部复位时的复位处理时间（解除 POR 后的第 1 次）”。解除外部复位时的复位处理时间如下所示：

解除 POR 后的第 1 次：0.672ms(TYP.)、0.832ms(MAX.)（使用 LVD 的情况）

0.399ms(TYP.)、0.519ms(MAX.)（不使用 LVD 的情况）

注 4：解除 POR 后第 2 次以后的解除外部复位时的复位处理时间如下所示：

解除 POR 后的第 2 次以后：0.531ms(TYP.)、0.675ms(MAX.)（使用 LVD 的情况）

0.259ms(TYP.)、0.362ms(MAX.)（不使用 LVD 的情况）

注 5：当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

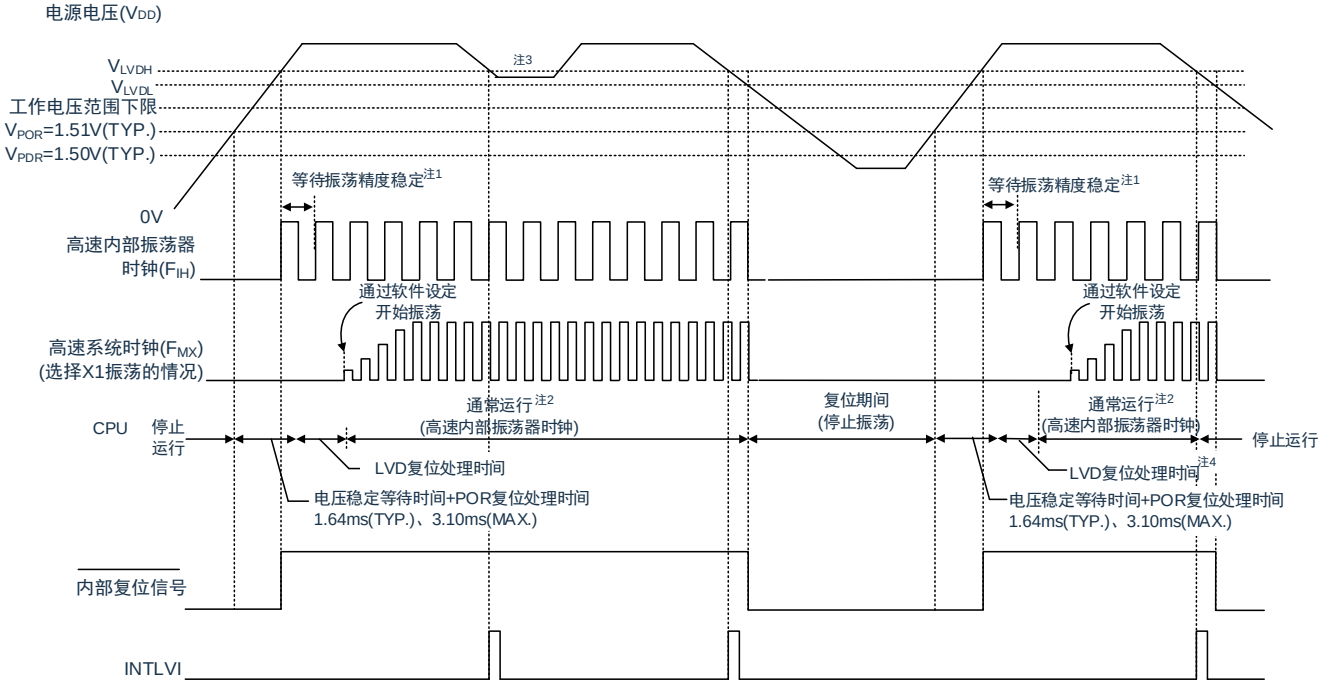
备注：V_{POR}：POR 电源电压上升检测电压

V_{PDR}：POR 电源电压下降检测电压

注意：在 LVD 为 OFF 时，必须使用 RESETB 引脚的外部复位。详细内容请参照“第 28 章 电压检测电路”。

图 27-3：上电复位电路和电压检测电路的内部复位信号的产生时序(2/3)

(2) LVD 为中断&复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、0）



注 1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2：能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注 3：在产生中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的 LVILV 位和 LVIMD 位置“1”。因此，必须考虑可能出现电源电压在不低于低电压检测电压（V_{LVDL}）的状态下恢复到高电压检测电压（V_{LVDH}）或者更高的情况，在产生 INTLVI 后按照“图 28-7：工作电压的确认/复位的设置步骤”和“图 28-8：中断&复位模式的初始设置步骤”进行设置。

注 4：到开始通常运行为止的时间除了达到 V_{POR}（1.51V(TYP.)）后的“电压稳定等待时间+POR 复位处理时间”以外，在达到 LVD 检测电平（V_{LVDH}）后还需要以下的“LVD 复位处理时间”。

LVD 复位处理时间：0ms~0.0701ms(MAX.)

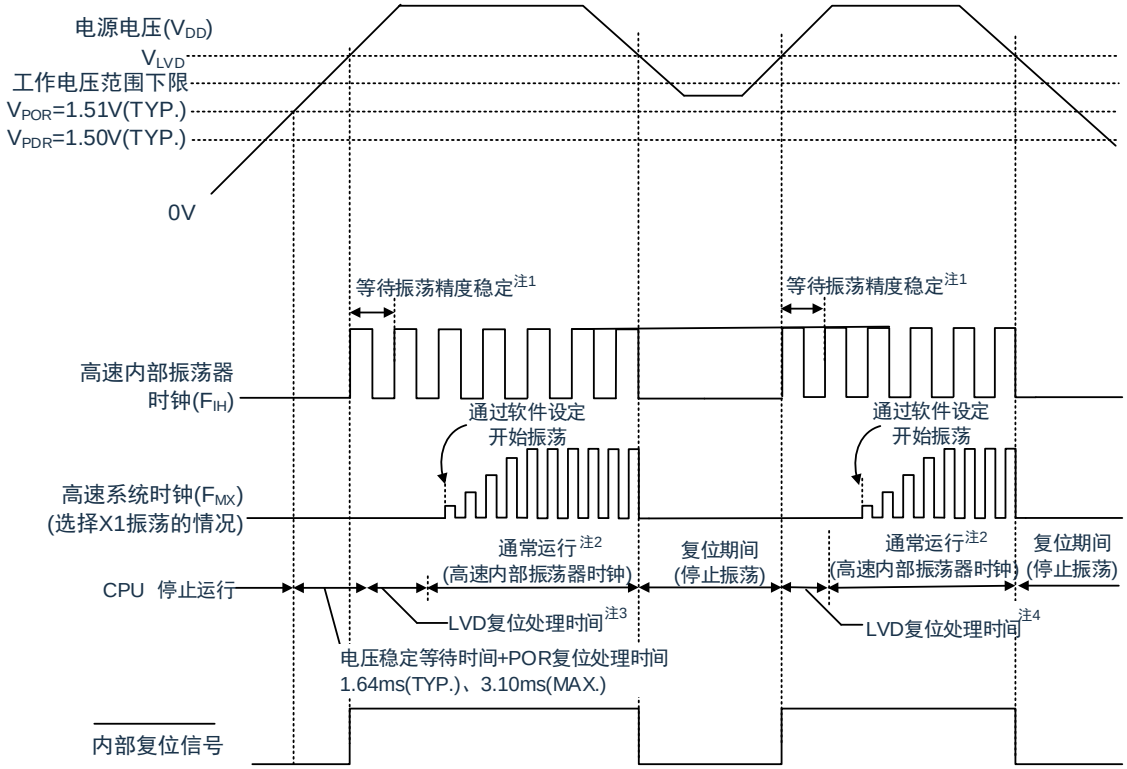
备注：V_{LVDH}、V_{LVDL}：LVD 检测电压

V_{POR}：POR 电源电压上升检测电压

V_{PDR}：POR 电源电压下降检测电压

图 27-4：上电复位电路和电压检测电路的内部复位信号的产生时序(3/3)

(3) LVD 复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、1）



注 1：内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

注 2：能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟或者副系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换；在使用 XT1 时钟的情况下，必须在利用定时器功能等确认振荡稳定时间后进行切换。

注 3：到开始通常运行为止的时间除了达到 V_{POR}（1.51V(TYP.)）后的“电压稳定等待时间+POR 复位处理时间”以外，在达到 LVD 检测电平（V_{LVD}）后还需要以下的“LVD 复位处理时间”。

LVD 复位处理时间：0ms~0.0701ms(MAX.)

注 4：在电源电压下降时，如果只在发生电压检测电路（LVD）的内部复位后恢复电源电压，就在达到 LVD 检测电平（V_{LVD}）后需要以下的“LVD 复位处理时间”。

LVD 复位处理时间：0.0511ms(TYP.)、0.0701ms(MAX.)

备注：

1. V_{LVDH}、V_{LVDL}：LVD 检测电压

   V_{POR}：POR 电源上升检测电压

   V_{PDR}：POR 电源下降检测电压

2. 当选择 LVD 中断模式（选项字节 000C1H 的 LVIMD1、LVIMD0=0、1）时，从接通电源到开始通常运行的时间和“图 26-2(3)LVD 位模式的情况”的“注 3”的时间相同。

第28章 电压检测电路

28.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设置运行模式和检测电压（V_{LVDH}、V_{LVDL}、V_{LVD}）。电压检测（LVD）电路有以下功能。

- 将电源电压（V_{DD}）和检测电压（V_{LVDH}、V_{LVDL}、V_{LVD}）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（V_{LVDH}、V_{LVDL}）能通过选项字节选择 12 种检测电平（参照“第 31 章 选项字节”）。
- 也能在深度睡眠模式中运行。
- 当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设置。

(a) 中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）
通过选项字节 000C1H 选择 2 个检测电压（V_{LVDH}、V_{LVDL}），高电压检测电平（V_{LVDH}）用于解除复位或者产生中断，低电压检测电平（V_{LVDL}）用于产生复位。

(b) 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）
将选项字节 000C1H 选择的 1 个检测电压（V_{LVD}）用于产生或者解除复位。

(c) 中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）
将选项字节 000C1H 选择的 1 个检测电压（V_{LVD}）用于产生中断或者解除复位。在各模式中，产生以下的中断信号和内部复位信号。

中断&复位模式 （LVIMDS1、LVIMDS0=1、0）	复位模式 （LVIMDS1、LVIMDS0=1、1）	中断模式 （LVIMDS1、LVIMDS0=0、1）
在工作电压下降时，当检测到V_{DD}＜V_{LVDH}时，产生中断请求信号；当检测到V_{DD}＜V_{LVDL}时，产生内部复位；当检测到V_{DD}≥V_{LVDH}时，解除内部复位。	当检测到V_{DD}≥V_{LVD}时，解除内部复位；当检测到V_{DD}＜V_{LVD}时，产生内部复位。	在发生复位后，LVD的内部复位状态继续保持到V_{DD}≥V_{LVD}为止。当检测到V_{DD}≥V_{LVD}时，解除LVD的内部复位。在解除LVD的内部复位后，如果检测到V_{DD}＜V_{LVD}或者V_{DD}≥V_{LVD}时，就产生中断请求信号（INTLVI）。

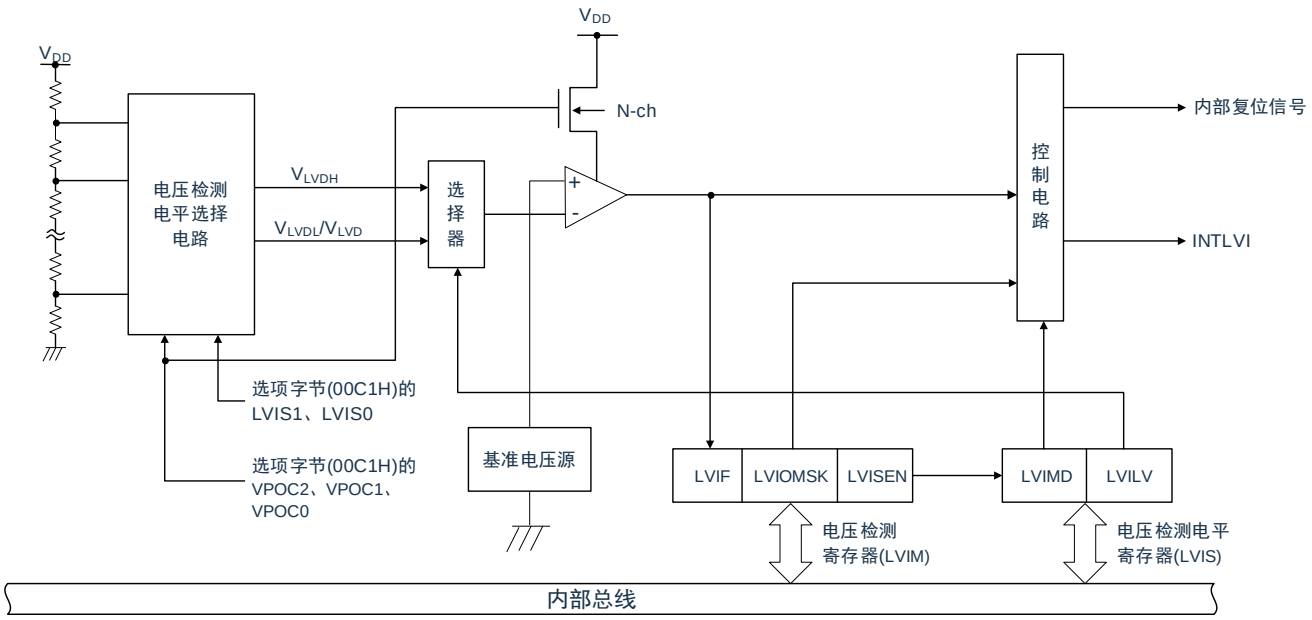
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

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 26 章 复位功能”。

28.2 电压检测电路的结构

电压检测电路的框图如图 28-1 所示。

图 28-1：电压检测电路的框图



28.3 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器（LVIM）
- 电压检测电平寄存器（LVIS）

28.3.1 电压检测寄存器（LVIM）

此寄存器设置允许或者禁止改写电压检测电平寄存器（LVIS），并且确认 LVD 输出的屏蔽状态。通过 8 位存储器操作指令设置 LVIM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 28-2：电压检测寄存器（LVIM）的格式

地址：40020441H 复位后：00H 注1　　R/W 注2

符号	7	6	5	4	3	2	1	0
LVIM	LVISEN注3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN注3	电压检测电平寄存器（LVIS）的允许/禁止改写的设置
0	禁止改写LVIS寄存器（LVIOMSK=0（LVD输出屏蔽无效））。
1	允许改写LVIS寄存器（LVIOMSK=1（LVD输出屏蔽有效））。

LVIOMSK	LVD输出的屏蔽状态标志
0	LVD输出屏蔽无效。
1	LVD输出屏蔽有效注4。

LVIF	电压检测标志
0	电源电压（V_{DD}）≥检测电压（V_{LVD}）或者LVD为OFF。
1	电源电压（V_{DD}）＜检测电压（V_{LVD}）

注 1：复位值因复位源而变。

在 LVD 发生复位时，不对 LVIM 寄存器的值进行复位而保持原来的值；在其他复位时，将 LVISEN 清“0”。

注 2：bit0 和 bit1 是只读位。

注 3：只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能设置，在其他模式中不能更改初始值。

注 4：只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时，LVIOMSK 位才在以下期间自动变为“1”，屏蔽 LVD 产生的复位或者中断。

- LVISEN=1 的期间
- 从发生 LVD 中断开始到 LVD 检测电压稳定为止的等待时间
- 从更改 LVILV 位的值到 LVD 检测电压稳定为止的等待时间

28.3.2 电压检测电平寄存器（LVIS）

这是设置电压检测电平的寄存器。

通过 8 位存储器操作指令设置 LVIS 寄存器。在产生复位信号后，此寄存器的值变为“00H/01H/81H”[注1]。

图 28-3：电压检测电平寄存器（LVIS）的格式

地址：40020442H 复位后：00H/01H/81H [注1]　　R/W

符号	7	6	5	4	3	2	1	0
LVIS	LVIMD[注2]	0	0	0	0	0	0	LVILV[注2]

LVIMD[注2]	电压检测的运行模式
0	中断模式
1	复位模式

LVILV[注2]	LVD检测电平
0	高电压检测电平（V_{LVDH}）
1	低电压检测电平（V_{LVDL}或者V_{LVD}）

注 1：复位值因复位源和选项字节的设置而变。在发生 LVD 复位时，不将此寄存器清“00H”。

在发生 LVD 以外的复位时，此寄存器的值如下：

- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
- 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
- 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H

注 2：只有在选择中断&复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能写“0”。在其他情况下不能设置。在中断&复位模式中，通过产生复位或者中断自动进行值的替换。

注意：

1. 要改写 LVIS 寄存器时，必须按照图 28-7 和图 28-8 的步骤进行。
2. 通过选项字节 000C1H 选择 LVD 的运行模式和各模式的检测电压（V_{LVDH}、V_{LVDL}、V_{LVD}）。用户选项字节（000C1H/010C1H）的格式如表 28-1 所示。有关选项字节的详细内容，请参照“第 31 章 选项字节”。

表 28-1：用户选项字节（000C1H/010C1H）的格式(1/2)

地址：000C1H/010C1H [注]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设置（中断&复位模式）

<table>
<tr><th colspan="3">检测电压</th><th colspan="8">选项字节的设置值</th></tr>
<tr><th colspan="2">V_{LVDH}</th><th>V_{LVDL}</th><th rowspan="2">VPOC2</th><th rowspan="2">VPOC1</th><th rowspan="2">VPOC0</th><th rowspan="2">LVIS1</th><th rowspan="2">LVIS0</th><th colspan="2">模式设置</th></tr>
<tr><th>上升</th><th>下降</th><th>下降</th><th>LVIMDS1</th><th>LVIMDS0</th></tr>
<tr><td>1.77V</td><td>1.73V</td><td rowspan="3">1.63V</td><td rowspan="18">0</td><td rowspan="3">0</td><td rowspan="3">0</td><td>1</td><td>0</td><td rowspan="18">1</td><td rowspan="18">0</td></tr>
<tr><td>1.88V</td><td>1.84V</td><td>0</td><td>1</td></tr>
<tr><td>2.92V</td><td>2.86V</td><td>0</td><td>0</td></tr>
<tr><td>1.98V</td><td>1.94V</td><td rowspan="3">1.84V</td><td rowspan="3">0</td><td rowspan="3">1</td><td>1</td><td>0</td></tr>
<tr><td>2.09V</td><td>2.04V</td><td>0</td><td>1</td></tr>
<tr><td>3.13V</td><td>3.06V</td><td>0</td><td>0</td></tr>
<tr><td>2.61V</td><td>2.55V</td><td rowspan="3">2.45V</td><td rowspan="3">1</td><td rowspan="3">0</td><td>1</td><td>0</td></tr>
<tr><td>2.71V</td><td>2.65V</td><td>0</td><td>1</td></tr>
<tr><td>3.75V</td><td>3.67V</td><td>0</td><td>0</td></tr>
<tr><td>2.92V</td><td>2.86V</td><td rowspan="3">2.75V</td><td rowspan="3">1</td><td rowspan="3">1</td><td>1</td><td>0</td></tr>
<tr><td>3.02V</td><td>2.96V</td><td>0</td><td>1</td></tr>
<tr><td>4.06V</td><td>3.98V</td><td>0</td><td>0</td></tr>
<tr><td colspan="3">—</td><td colspan="8">禁止设置上述以外的值。</td></tr>
</table>

- LVD 的设置（复位模式）

<table>
<tr><th colspan="2">检测电压</th><th colspan="8">选项字节的设置值</th></tr>
<tr><th colspan="2">V_{LVD}</th><th rowspan="2">VPOC2</th><th rowspan="2">VPOC1</th><th rowspan="2">VPOC0</th><th rowspan="2">LVIS1</th><th rowspan="2">LVIS0</th><th colspan="2">模式设置</th></tr>
<tr><th>上升</th><th>下降</th><th>LVIMDS1</th><th>LVIMDS0</th></tr>
<tr><td>1.67V</td><td>1.63V</td><td rowspan="15">0</td><td>0</td><td>0</td><td>1</td><td>1</td><td rowspan="15">1</td><td rowspan="15">1</td></tr>
<tr><td>1.77V</td><td>1.73V</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1.88V</td><td>1.84V</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1.98V</td><td>1.94V</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>2.09V</td><td>2.04V</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>2.50V</td><td>2.45V</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>2.61V</td><td>2.55V</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>2.71V</td><td>2.65V</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>2.81V</td><td>2.75V</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>2.92V</td><td>2.86V</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>3.02V</td><td>2.96V</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>3.13V</td><td>3.06V</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>3.75V</td><td>3.67V</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>4.06V</td><td>3.98V</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td colspan="2">—</td><td colspan="8">禁止设置上述以外的值。</td></tr>
</table>

备注：

1. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
2. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

表 28-1：用户选项字节（000C1H）的格式(2/2)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设置（中断模式）

检测电压		选项字节的设置值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设置	
上升	下降						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	0	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设置上述以外的值。						

- LVD 为 OFF（使用 RESETB 引脚的外部复位）

检测电压		选项字节的设置值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	上升						LVIMDS1	LVIMDS1
—	—	1	×	×	×	×	×	×
—		禁止设置上述以外的值。						

注意：

1. 必须给 bit4 写“1”。
2. 当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H）的设置。

备注：

1. ×：忽略
2. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
3. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

28.4 电压检测电路的运行

28.4.1 用作复位模式时的设置

通过选项字节 000C1H 设置运行模式（复位模式（LVIMDS1、LVIMDS0=1、1））和检测电压（V_{LVD}）。如果设置复位模式，就在以下初始设置的状态下开始运行。

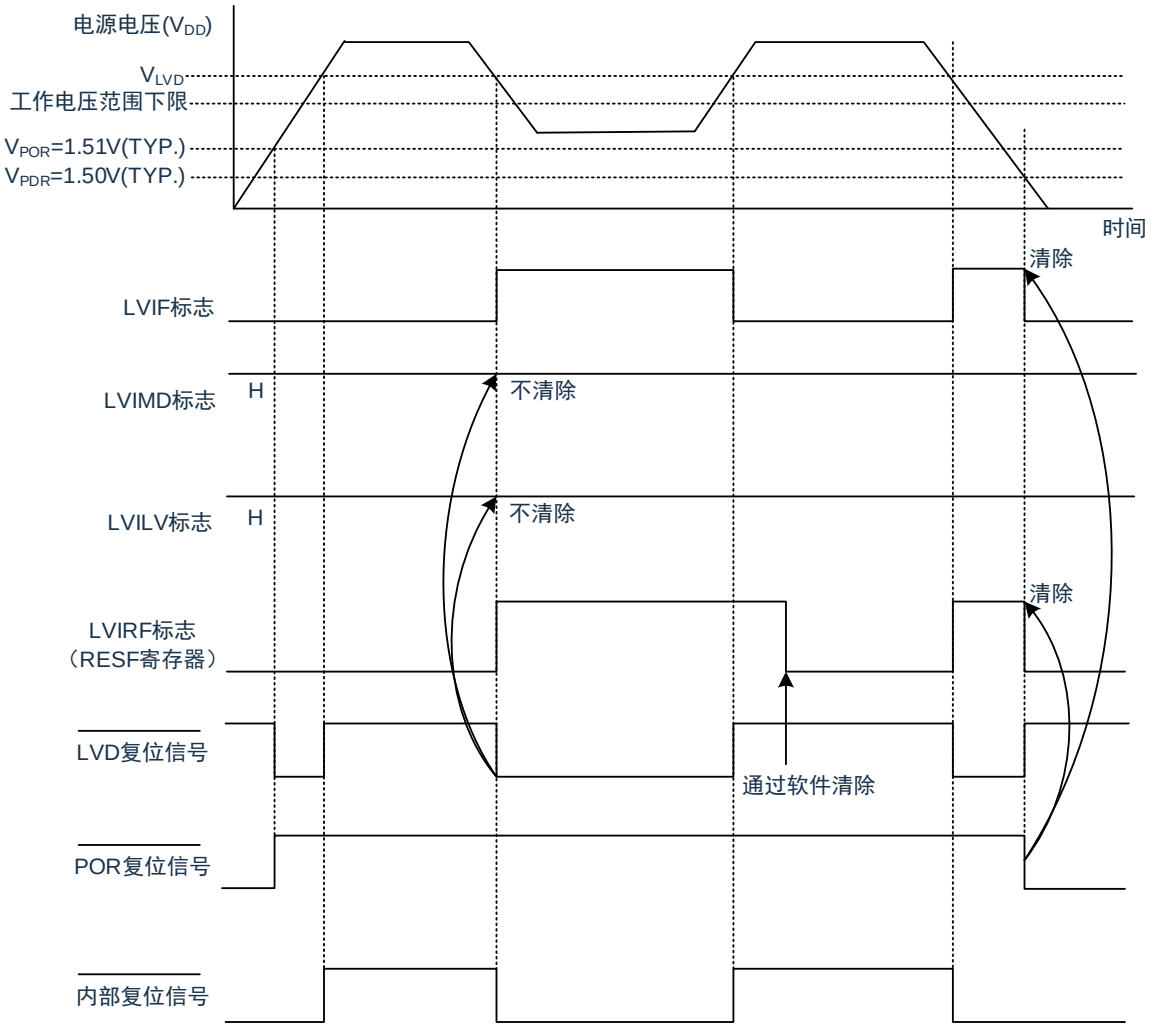
- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。bit7（LVIMD）为“1”（复位模式）。bit0（LVILV）为“1”（电压检测电平：V_{LVD}）。
- LVD 复位模式的运行

当接通电源时，复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）在电源电压（V_{DD}）超过电压检测电平（V_{LVD}）前保持 LVD 的内部复位状态。如电源电压（V_{DD}）超过电压检测电平（V_{LVD}），就解除内部复位。

当工作电压下降时，如果电源电压（V_{DD}）低于电压检测电平（V_{LVD}），就产生 LVD 的内部复位。

LVD 复位模式的内部复位信号的产生时序如图 28-4 所示。

图 28-4：内部复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、1）



备注：V_{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

28.4.2 用作中断模式时的设置

通过选项字节 000C1H 设置运行模式（中断模式（LVIMDS1、LVIMDS0=0、1））和检测电压（V_{LVD}）。如果设置中断模式，就在以下初始设置的状态下开始运行。

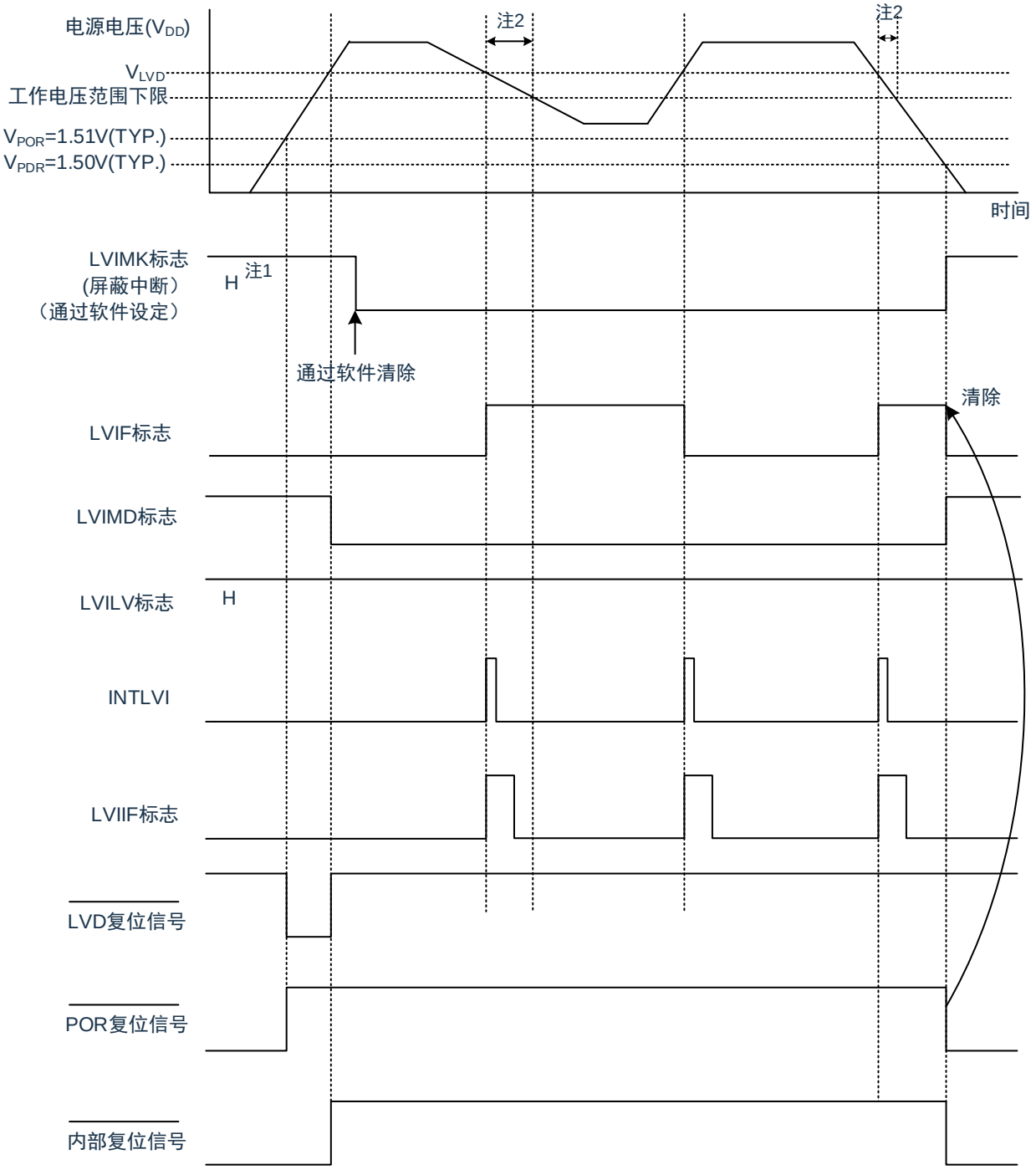
- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“01H”。bit7（LVIMD）为“0”（中断模式）。
  bit0（LVILV）为“1”（电压检测电平：VLVD）。
- LVD 中断模式的运行

在产生复位后，中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（V_{DD}）超过电压检测电平（V_{LVD}）前保持 LVD 的内部复位状态。如果电源电压（V_{DD}）超过电压检测电平（V_{LVD}），就解除 LVD 的内部复位。

在解除 LVD 的内部复位后，如果电源电压（V_{DD}）超过电压检测电平（V_{LVD}），就产生 LVD 的中断请求信号（INTLVI）。当工作电压下降时，必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的产生时序如图 28-5 所示。

图 28-5：中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=0、1）



注 1：在产生复位信号后，LVIMK 标志变为“1”。

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

28.4.3 用作中断&复位模式时的设置

通过选项字节 000C1H 设置运行模式（中断&复位模式（LVIMDS1、LVIMDS0=1、0））和检测电压（V_{LVDH}、V_{LVDL}）。

如果设置中断&复位模式，就在以下初始设置的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。bit7（LVIMD）为“0”（中断模式）。
  bit0（LVILV）为“0”（高电压检测电平：V_{LVDH}）。

LVD 中断&复位模式的运行

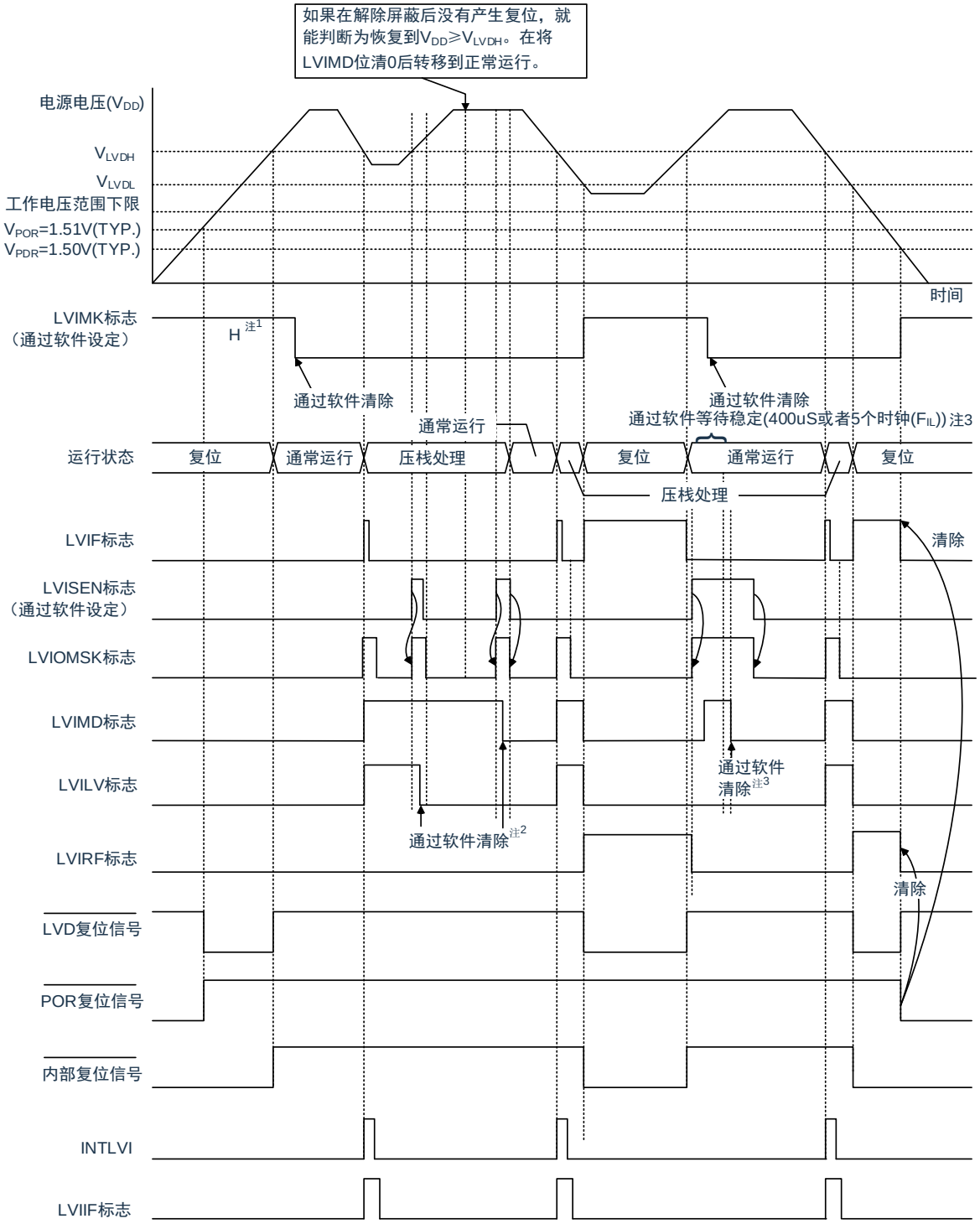
当接通电源时，中断&复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（V_{DD}）超过高电压检测电平（V_{LVDH}）前保持 LVD 的内部复位状态。如果电源电压（V_{DD}）超过高电压检测电平（V_{LVDH}），就解除内部复位。

当工作电压下降时，如果电源电压（V_{DD}）低于高电压检测电平（V_{LVDH}），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（V_{DD}）低于低电压检测电平（V_{LVDL}），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压（V_{DD}）在不低于低电压检测电压（V_{LVDL}）的状态下恢复到高电压检测电压（V_{LVDH}）或者更高，也不产生中断请求信号。

当使用 LVD 中断&复位模式时，必须按照“图 28-7：工作电压的确认/复位的设置步骤”和“图 28-8：中断&复位模式的初始设置步骤”所示的流程图的步骤进行设置。

LVD 中断&复位模式的内部复位信号和中断信号的产生时序如图 28-6 所示。

图 28-6：复位&中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(1/2)



注 1：在产生复位信号后，LVIMK 标志变为“1”。

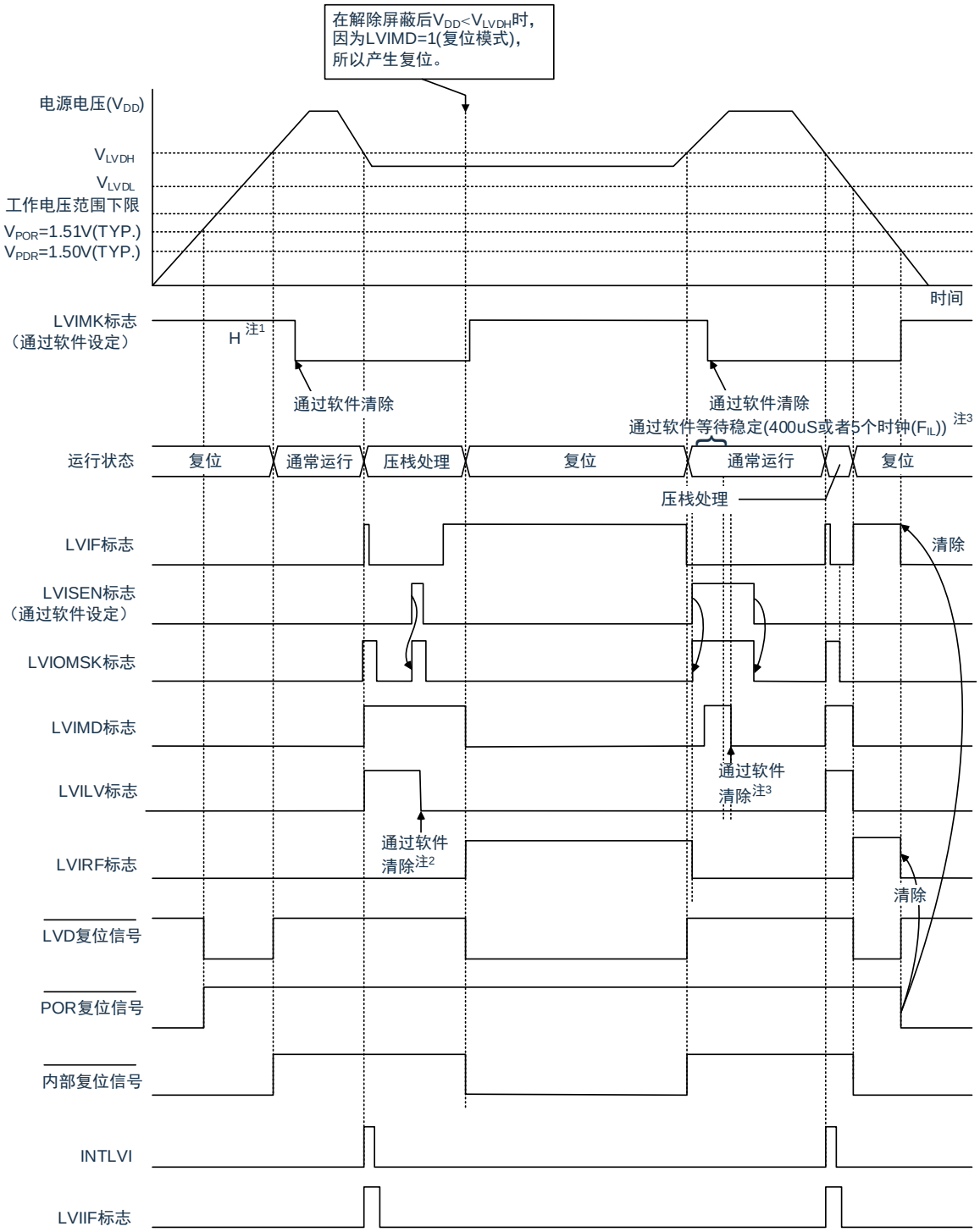
注 2：当使用中断&复位模式时，必须在发生中断后按照“图 28-7：工作电压的确认/复位的设置步骤”进行设置。

注 3：当使用中断&复位模式时，必须在解除复位后按照“图 28-8：中断&复位模式的初始设置步骤”进行设置。

备注：V_{POR}：POR 电源电压上升检测电压

V_{PDR}：POR 电源电压下降检测电压

图 28-6：中断&复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、0）(2/2)



注 1：在产生复位信号后，LVIMK 标志变为“1”。

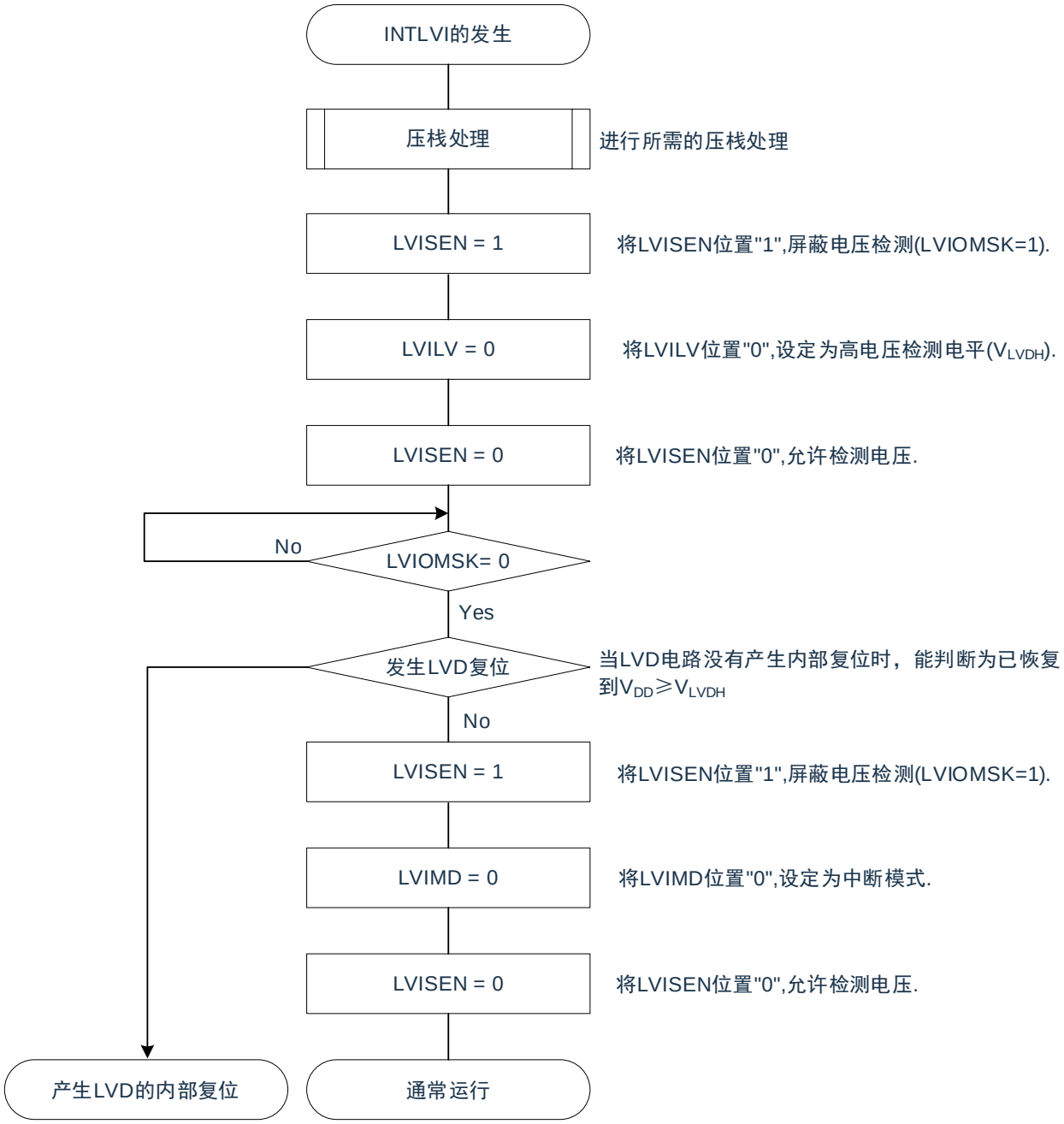
注 2：当使用中断&复位模式时，必须在发生中断后按照“图 28-7：工作电压的确认/复位的设置步骤”进行设置。

注 3：当使用中断&复位模式时，必须在解除复位后按照“图 28-8：中断&复位模式的初始设置步骤”进行设置

备注：V_{POR}：POR 电源电压上升检测电压

V_{PDR}：POR 电源电压下降检测电压

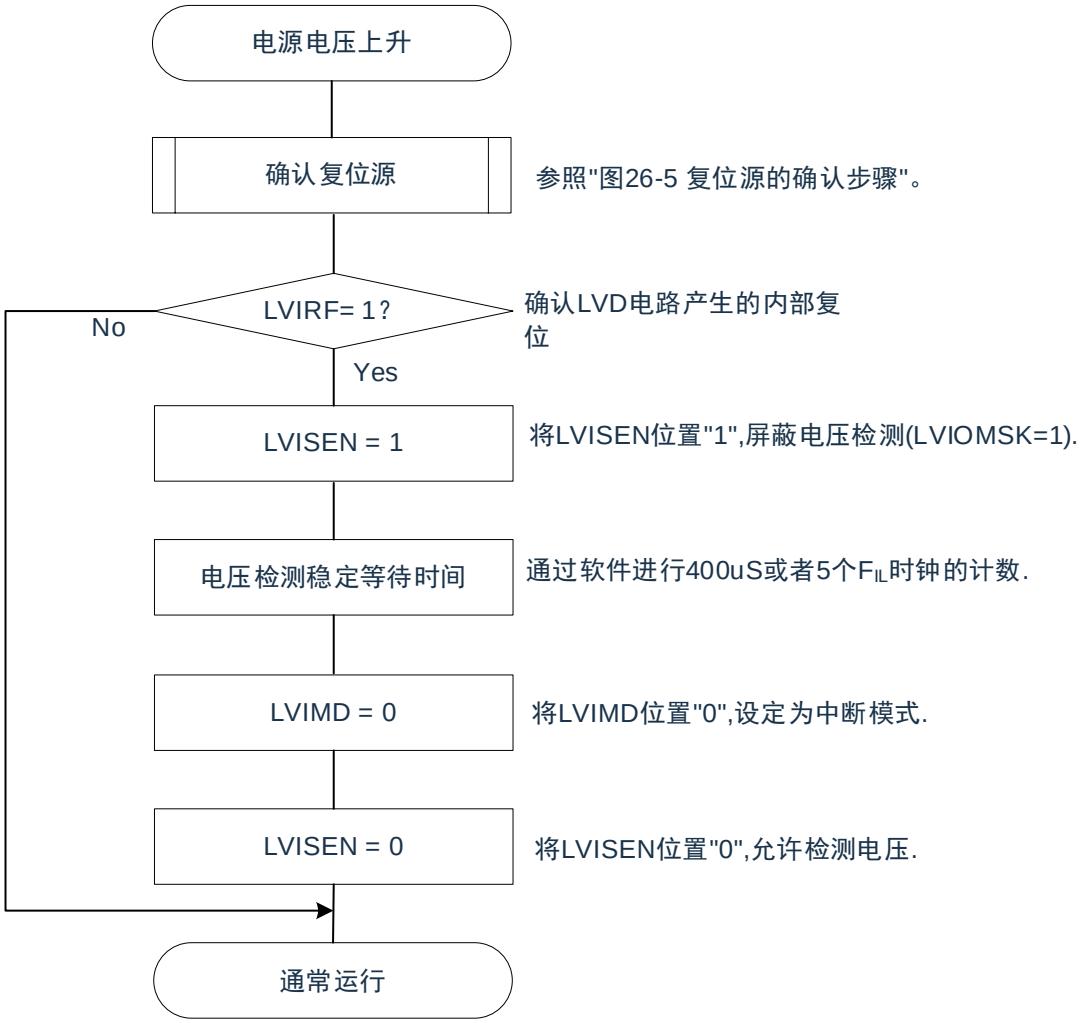
图 28-7：工作电压的确认/复位的设置步骤



如果设置中断&复位模式（LVIMDS1、LVIMDS0=1、0），就在解除 LVD 复位（LVIRF=1）后需要 400us 或者 5 个 F_{IL} 时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时，必须将 LVISEN 位置“1”，屏蔽 LVD 产生的复位或者中断的产生。

中断&复位模式的初始设置步骤如图 28-8 所示。

图 28-8：中断&复位模式的初始设置步骤



备注：F_{IL}：低速内部振荡器时钟频率

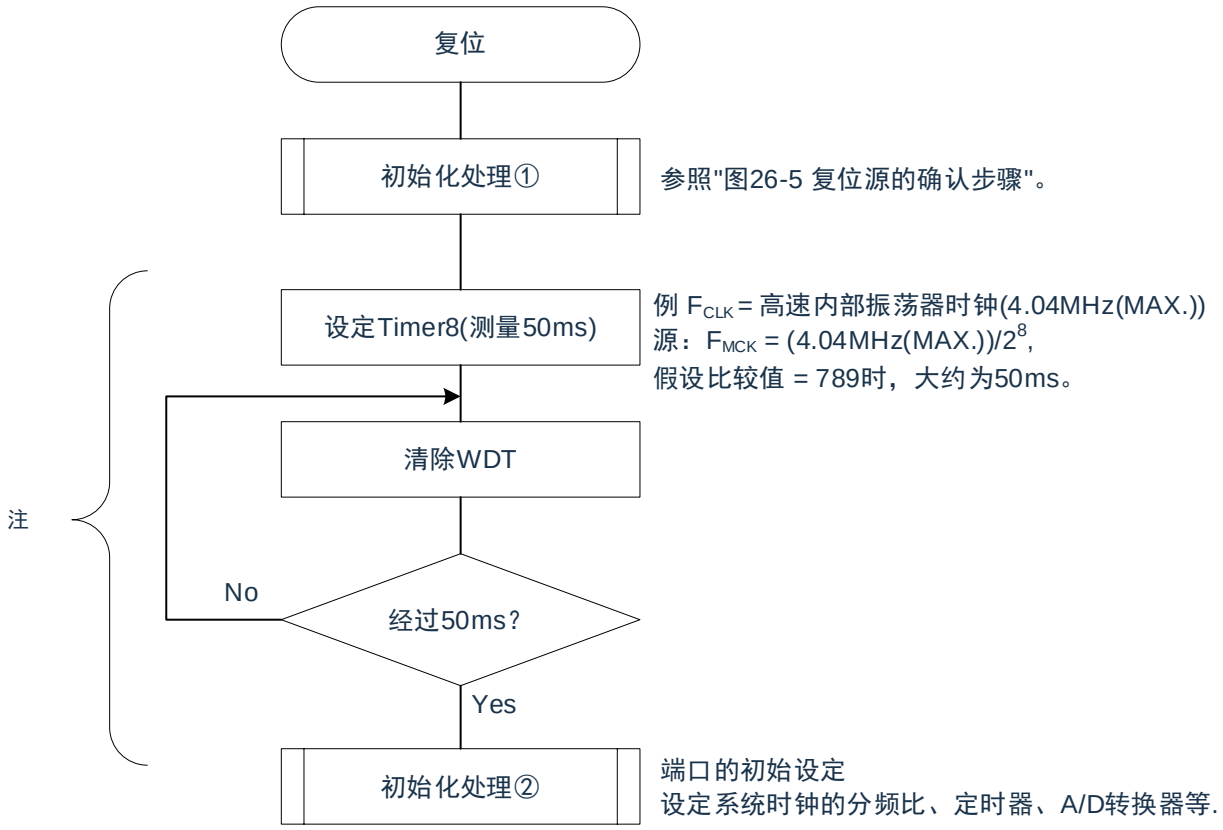
28.5 电压检测电路的注意事项

(1) 有关接通电源时的电压波动

对于电源电压（V_{DD}）在 LVD 检测电压附近发生一定时间波动的系统，有可能重复进入复位状态和复位解除状态。能通过以下的处理，任意设置解除复位到单片机开始运行的时间。

＜处理＞在解除复位后，必须通过使用定时器的软件计数器，在等待各系统不同的电源电压波动时间后进行端口等的初始设置。

图 28-9：LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子

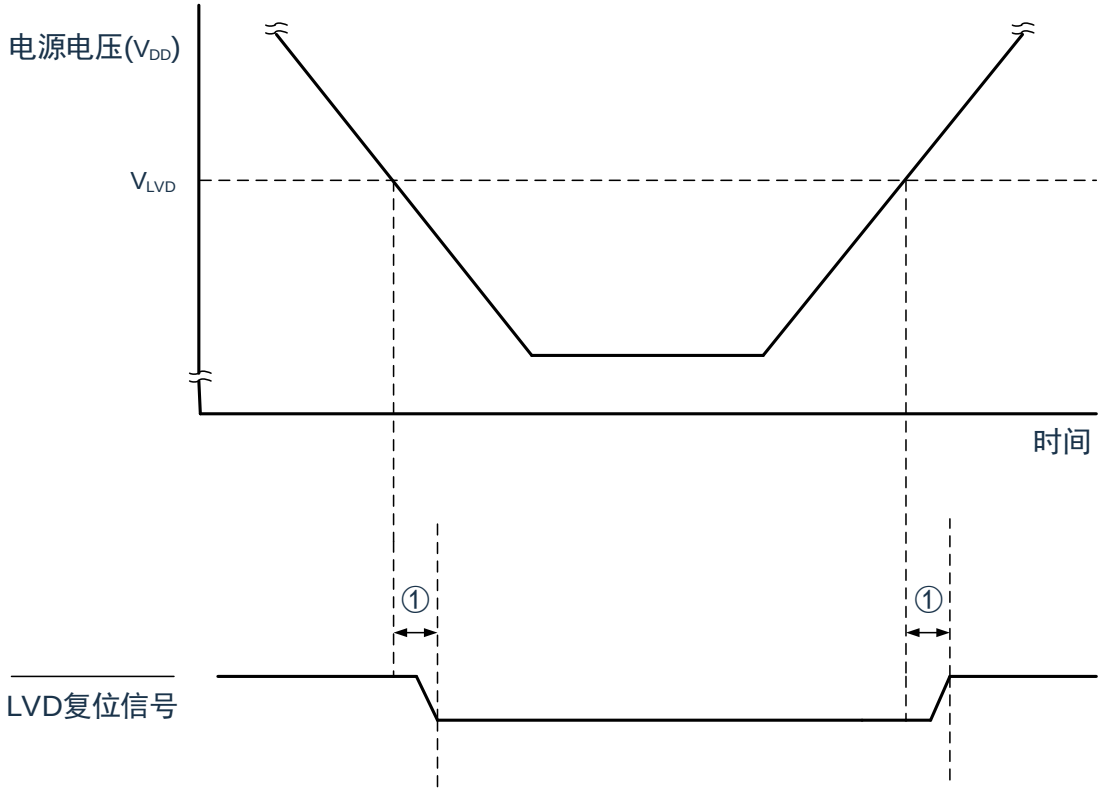


注：如果在此期间再次发生复位，就不转移到初始化处理②。

(2) 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压（V_{DD}）＜LVD 检测电压（V_{LVD}）到产生 LVD 复位为止会发生延迟。同样，从 LVD 检测电压（V_{LVD}）≤电源电压（V_{DD}）到解除 LVD 复位为止也会发生延迟（参照图 28-10）。

图 28-10：从产生 LVD 复位源到产生或者解除 LVD 复位的延迟



备注：①：检测延迟（300us(MAX.)）

(3) 有关将 LVD 置为 OFF 时接通电源的情况

当将 LVD 置为 OFF 时，必须使用 RESETB 引脚的外部复位。

在进行外部复位时，必须至少给 RESETB 引脚输入 10us 的低电平。如果在电源电压上升时进行外部复位，就必须在给 RESETB 引脚输入低电平后接通电源，而且在数据手册的 AC 特性所示的工作电压范围内至少保持 10us 的低电平，然后输入高电平。

(4) 有关将 LVD 置为 OFF 并且设置为 LVD 中断模式时工作电压下降的情况

在将 LVD 置为 OFF 并且设置为 LVD 中断模式的情况下，如果工作电压下降，就必须在工作电压低于数据手册的 AC 特性所示的工作电压范围前，通过深度睡眠模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

第29章 安全功能

29.1 安全功能的概要

为了对应 IEC60730 和 EC61508 安全标准，CMS32H6157 内置以下安全功能。

此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

(1) 闪存CRC运算功能（高速CRC、通用CRC）

通过CRC运算检测闪存的数据错误。能根据不同的用途和使用条件，分别使用以下2个CRC。

- “高速CRC”…在初始化程序中，能停止CPU的运行并且高速检查整个代码闪存区。
- “通用CRC”…在CPU运行中，不限于代码闪存区而能用于多用途的检查。

(2) RAM 奇偶校验错误检测功能

在读RAM数据时，检测奇偶校验错误。

(3) SFR保护功能

防止因CPU失控而改写SFR。

(4) 频率检测功能

能使用通用定时器单元进行CPU/外围硬件时钟频率的自检。

(5) A/D 测试功能

能通过A/D转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、温度传感器输出和内部基准电压输出的A/D转换进行A/D转换器的自检。

(6) 输入/输出端口的数字输出信号电平检测功能

在输入/输出端口为输出模式时，能读引脚的输出电平。

(7) 产品唯一身份标识寄存器（128位）

29.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
• 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）
• RAM奇偶校验错误控制寄存器（RPECTL）	RAM奇偶校验错误检测功能
• 特殊SFR保护控制寄存器（SFRGD）	SFR保护功能
• 定时器输入选择寄存器0（TIS0）	频率检测功能
• A/D测试寄存器（ADTES）	A/D测试功能
• 端口模式选择寄存器（PMS）	输入/输出引脚的数字输出信号电平检测功能

有关各寄存器的内容，在“29.3 安全功能的运行”中进行说明。

29.3 安全功能的运行

29.3.1 闪存CRC运算功能（高速CRC）

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设置（初始化）程序中检查整个代码闪存区。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短（例如，64KB 闪存：512us@32MHz）。

CRC 生成多项式对应 CRC-16-CCITT 的“$X^{16}+X^{12}+X^{5}+1$”。

以 bit31bit0 的 MSB 优先进行运算。

备注：因为通用 CRC 为 LSB 优先，所以运算结果不同。

29.3.1.1 闪存CRC控制寄存器（CRC0CTL）

这是设置高速 CRC 运算器的运行控制和运算范围的寄存器。通过 8 位存储器操作指令设置 CRC0CTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 29-1：闪存 CRC 控制寄存器（CRC0CTL）的格式

地址：40021810H复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	高速CRC运算器的运行控制
0	停止运行。
1	通过执行HALT指令开始运算。

FEA6	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	高速CRC的演算范围
0	0	0	0	0	0	0	00000H ~ 1FFBH(8K-4byte)
0	0	0	0	0	0	1	00000H ~ 3FFBH(16K-4byte)
0	0	0	0	0	1	0	00000H ~ 5FFBH(24K-4byte)
0	0	0	0	0	1	1	00000H ~ 7FFBH(32K-4byte)
0	0	0	0	1	0	0	00000H ~ 9FFBH(40K-4byte)
0	0	0	0	1	0	1	00000H ~ BFFBH(48K-4byte)
0	0	0	0	1	1	0	00000H ~ DFFBH(56K-4byte)
0	0	0	0	1	1	1	00000H ~ FFFBH(64K-4byte)
0	0	0	1	0	0	0	00000H ~ 11FFBH(72K-4byte)
0	0	0	1	0	0	1	00000H ~ 13FFBH(80K-4byte)
0	0	0	1	0	1	0	00000H ~ 15FFBH(88K-4byte)
0	0	0	1	0	1	1	00000H ~ 17FFBH(96K-4byte)
0	0	0	1	1	0	0	00000H ~ 19FFBH(104K-4byte)
0	0	0	1	1	0	1	00000H ~ 1BFFBH(112K-4byte)
0	0	0	1	1	1	0	00000H ~ 1DFFBH(120K-4byte)
0	0	0	1	1	1	1	00000H ~ 1FFFBH(128K-4byte)
1	x	x	x	x	x	x	00000H ~ 1EFFBH(124K-4byte)

备注：必须事先将用于比较的 CRC 运算结果期待值存入闪存的最后 4 字节，因此运算范围为减去 4 字节的范围。

29.3.1.2 闪存CRC运算结果寄存器（PGCRCL）

这是保存高速 CRC 运算结果的寄存器。

通过 16 位存储器操作指令设置 PGCRCL 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 29-2：闪存 CRC 运算结果寄存器（PGCRCL）的格式

地址: 0x40021812 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

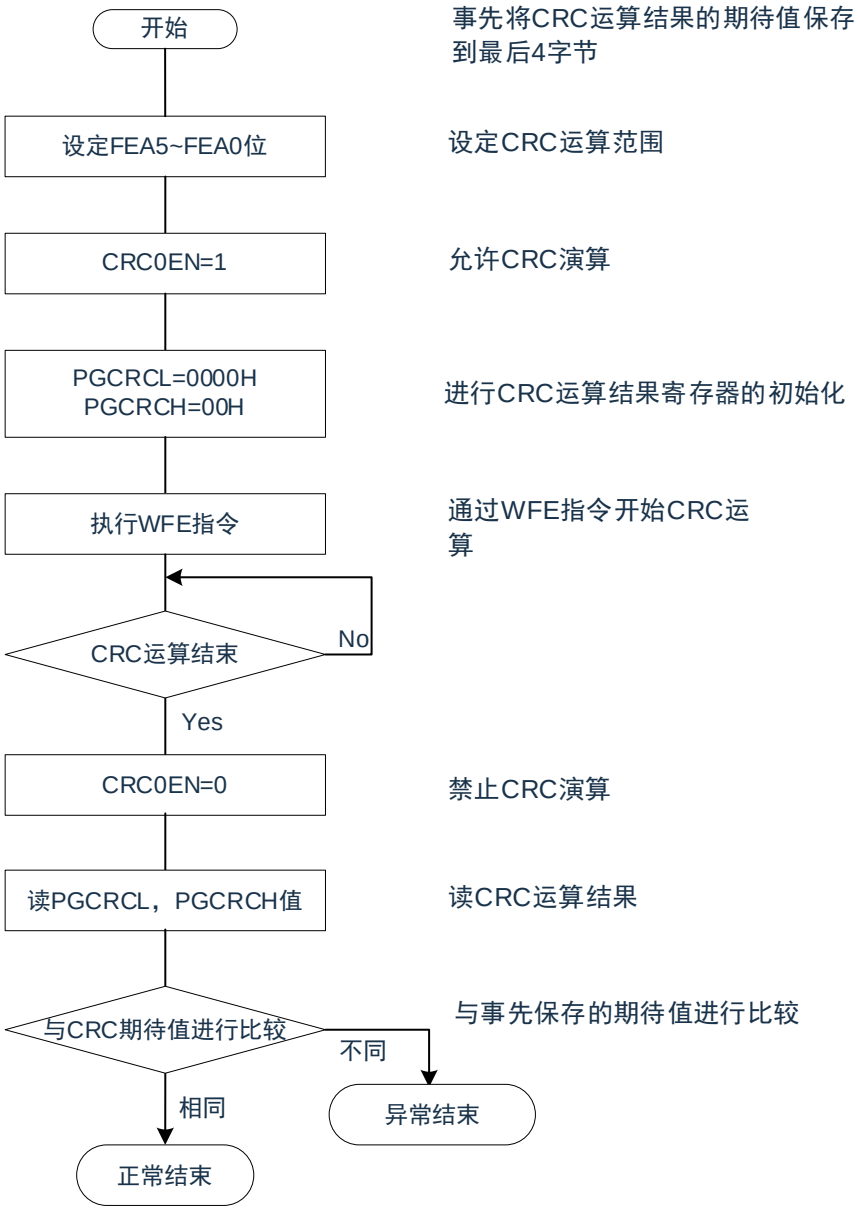
PGCRC15~0	高速CRC的运算结果
0000H~FFFFH	保存高速CRC的运算结果。

注意：只有在 CRC0EN（CRC0CTL 寄存器的 bit7）位为“1”时才能写 PGCRCL 寄存器。

闪存 CRC 运算功能（高速 CRC）的流程图如图 29-3 所示。

＜操作流程＞

图 29-3：闪存 CRC 运算功能（高速 CRC）的流程图



注意：

1. 只以代码闪存为 CRC 运算的对象。
2. 必须将 CRC 运算的期待值保存在代码闪存中的运算范围后的区域。

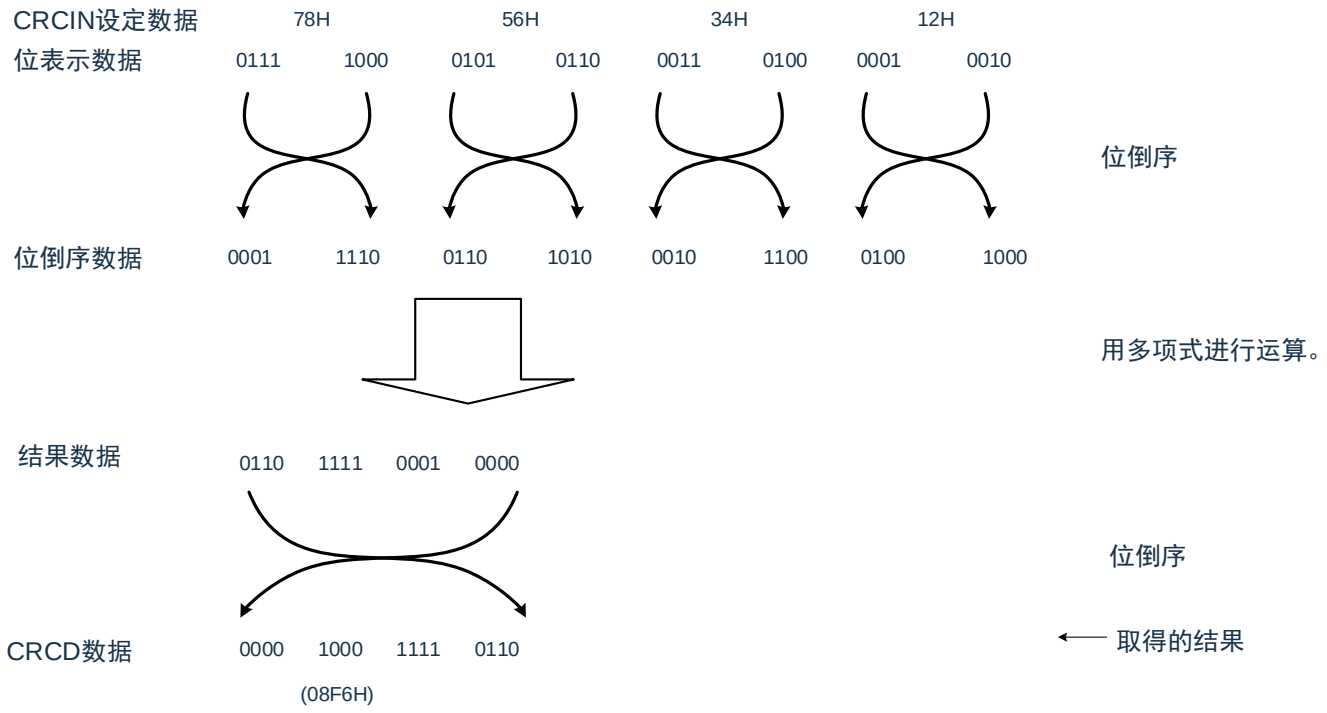
29.3.2 CRC运算功能（通用CRC）

为了必须保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。

此通用 CRC 能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定确认的数据。睡眠模式中的 CRC 运算功能只能在 DMA 传送过程中使用。

在主系统时钟运行模式或者副系统时钟运行模式中，都能使用 CRC 运算功能。

CRC 生成多项式使用 CRC-16-CCITT 的“$X^{16}+X^{12}+X^{5}+1$”。因为考虑到是以 LSB 优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从 LSB 发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给 CRCIN 寄存器写值，从 CRCD 寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行 CRC 运算的结果。



注意：在执行程序的过程中，因为调式程序将软件断点的设置行改写为断点指令，所以如果在 CRC 运算的对象区设置软件断点，CRC 的运算结果就不同。

29.3.2.1CRC输入寄存器（CRCIN）

这是设置通用 CRC 的 CRC 计算数据的 8 位寄存器。能设置的范围为“00H~FFH”。

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

图 29-4：CRC 输入寄存器（CRCIN）的格式

地址：40041400H复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
CRCIN								

bit7~0	功能
00H~FFH	数据输入

29.3.2.2 CRC数据寄存器（CRCD）

这是保存通用 CRC 运算结果的寄存器。能设置的范围为“0000H~FFFFH”。

在写 CRCIN 寄存器后经过 1 个 CPU/外围硬件时钟（F_{CLK}），将 CRC 运算结果保存到 CRCD 寄存器。通过 16 位存储器操作指令设置 CRCD 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 29-5：CRC 数据寄存器（CRCD）的格式

地址：40041402H 复位后：0000H　　R/W

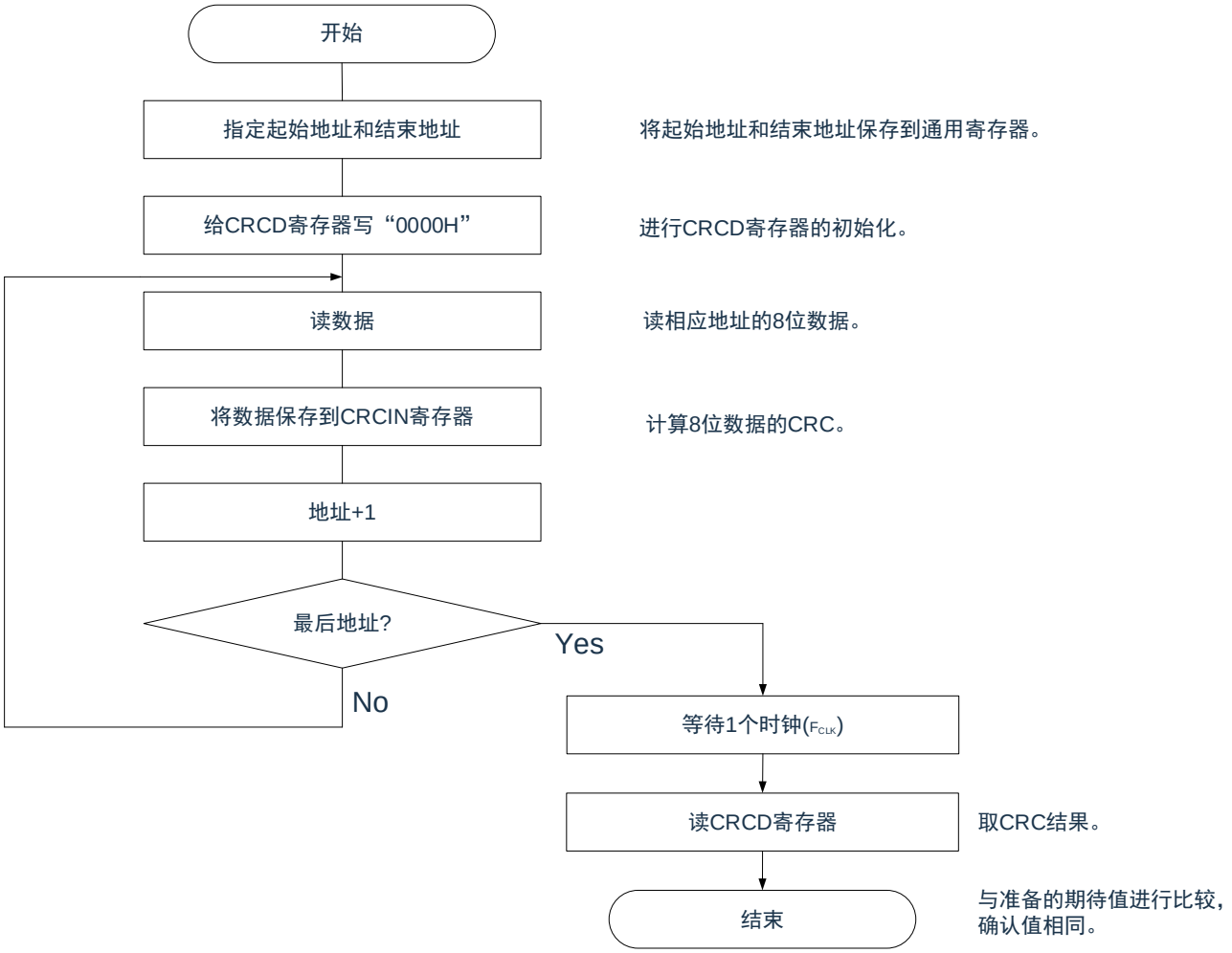
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

注意：

1. 要读 CRCD 寄存器的写入值时，必须在写 CRCIN 寄存器前读 CRCD 寄存器。
2. 如果 CRCD 寄存器的写操作与运算结果的保存发生竞争，就忽视写操作。

＜操作流程＞

图 29-6：CRC 运算功能（通用 CRC）的流程图



29.3.3 RAM奇偶校验错误检测功能

IEC60730 标准要求确认 RAM 数据。因此，CMS32H6157 的 RAM 每 8 位附加 1 位奇偶校验位。RAM 奇偶校验错误检测功能在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位，并且能在发生奇偶校验错误时产生复位。

29.3.3.1 RAM奇偶校验错误控制寄存器（RPECTL）

此寄存器控制奇偶校验的错误确认位和因奇偶校验错误而产生复位。通过 8 位存储器操作指令设置 RPECTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 29-7：RAM 奇偶校验错误控制寄存器（RPECTL）的格式

地址：40020425H 复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	奇偶校验错误复位的屏蔽标志
0	允许产生奇偶校验错误复位。
1	禁止产生奇偶校验错误复位。

RPEF	奇偶校验错误状态标志
0	没有发生奇偶校验错误。
1	发生奇偶校验错误。

注意：在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位。

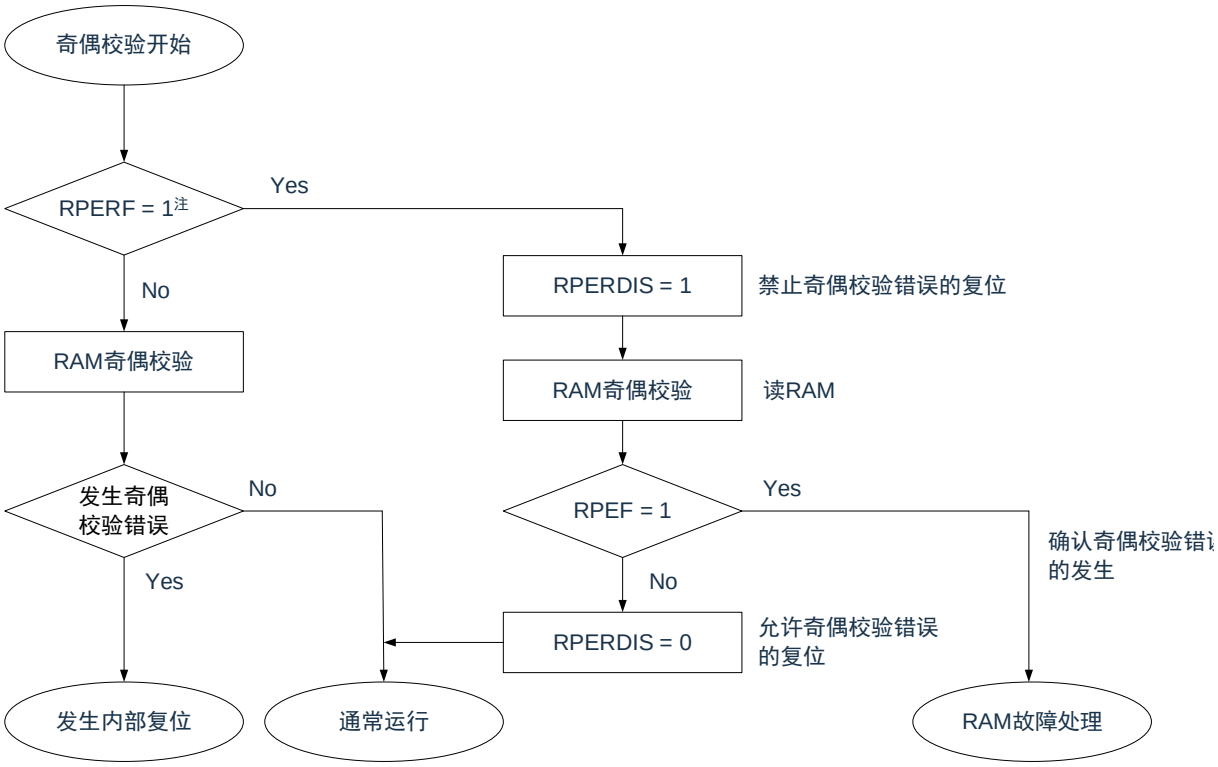
因此，要允许产生 RAM 奇偶校验错误复位（RPERDIS=0）时，必须在存取数据时并且在读数据前对“所用 RAM 区”进行初始化。

因为是流水线运行，所以 CPU 进行预读，有可能因读所用 RAM 区前的未初始化 RAM 区而发生 RAM 奇偶校验错误。因此，要允许产生 RAM 奇偶校验错误复位（RPERDIS=0）时，必须在从 RAM 区执行指令时对“所用 RAM 区+10 字节”的区域进行初始化。

备注：

1. 初始状态为允许产生奇偶校验错误复位（RPERDIS=0）。
2. 即使设置为禁止产生奇偶校验错误复位（RPERDIS=1），也在发生奇偶校验错误时将 RPEF 标志置“1”。如果在 RPEF 位为“1”的状态下设置为允许产生奇偶检验错误复位（RPERDIS=0），就在将 RPERDIS 清“0”时产生奇偶校验错误复位。
3. 因 RAM 奇偶校验错误而将 RPECTL 寄存器的 RPEF 标志置“1”，通过写“0”或者全部复位源将 RPEF 标志清“0”。当 RPEF 标志为“1”时，即使读未发生奇偶校验错误的 RAM，RPEF 标志也保持“1”的状态。
4. RAM 奇偶校验检测的范围不包括通用寄存器。

图 29-8：RAM 奇偶校验的流程



注：有关 RAM 奇偶错误的内部复位的确认，请参照“第 26 章 复位功能”。

29.3.4 SFR保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护重要的 SFR，使其免遭改写。SFR 保护功能用于保护比较器功能、端口功能、中断功能、时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的数据。

如果设置为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

29.3.4.1 SFR保护控制寄存器（SFRGD）

此寄存器控制 SFR 保护功能是否有效。

SFR 保护功能使用 GCOMP 位、GPORT 位、GINT 位和 GCSC 位。

通过 8 位存储器操作指令设置 SFRGD 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 29-9：SFR 保护控制寄存器（SFRGD）的格式

地址：40040C08H　　复位后：00H　　R/W

符号	7	6	5	4	3	2	1	0
SFRGD	0	0	0	0	GCOMP	GPORT	GINT	GCSC

GCOMP	比较器功能的控制寄存器的保护
0	无效。能读写比较器功能的控制寄存器。
1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的SFR]COMPMDR、COMPFIR、COMPOCR、CMPSEL0、CMPSEL1、C0REFS、C1REFS、CMP0HY、CMP1HY、AMPCTL、AMPDAC

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]ISCLCD、SEGn(n=0~3)、PMxx、PUxx、PDxx、POMxx、PMCxx、PSETxx、PCLRxx、PREADxx（xx=A、B、C、D、H）注

GINT	中断功能的寄存器的保护
0	无效。能读写中断功能的控制寄存器。
1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的SFR]IFxx、MKxx、PRxx、EGPx、EGNx

GCSC	时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的保护
0	无效。能读写时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器。
1	有效。时钟控制功能、电压检测电路和RAM奇偶校验错误检测功能的控制寄存器的写操作无效，能读。 [被保护的SFR]CMC、CSC、OSTS、CKC、PERx、OSMC、LVIM、LVIS、RPECTL

注：不保护 Pxx（端口寄存器）。

29.3.5 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

频率检测功能可使用 CPU/外围硬件的时钟频率（F_{CLK}），并且能通过测量 Timer8 的通道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟停止振荡，就不能判断 2 个时钟的比率关系。

＜要比较的时钟＞

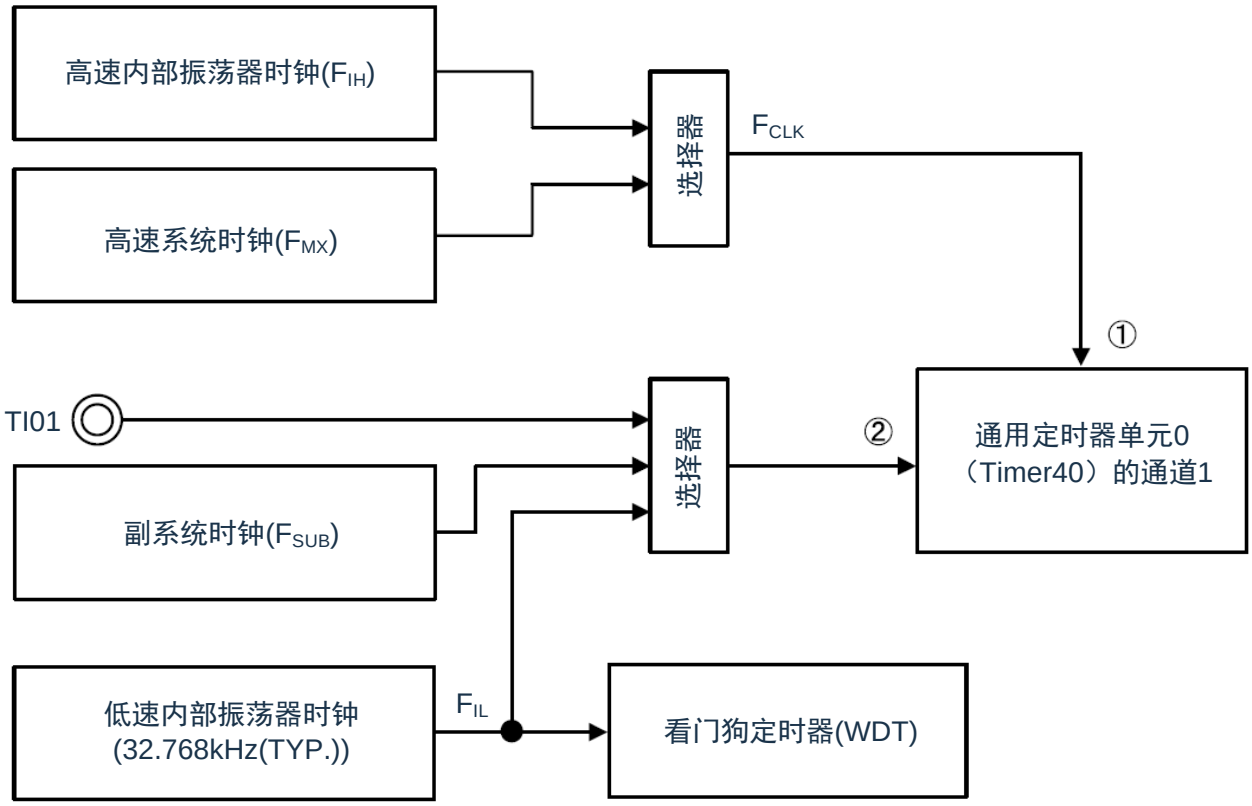
① CPU/外围硬件的时钟频率（F_{CLK}）：

- 高速内部振荡器时钟（F_{IH}）
- 高速系统时钟（F_{MX}）

② Timer8 的通道 1 输入：

- 通道 1 的定时器输入（TI01）
- 低速内部振荡器时钟（F_{IL}：32.768KHz(TYP.)）
- 副系统时钟（F_{SUB}）注

图 29-10：频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。有关输入脉冲间隔的测量方法，请参照“5.8.4 作为输入脉冲间隔测量的运行”。

注：只有内置副系统时钟的产品才能选择。

29.3.5.1 定时器输入选择寄存器0（TIS0）

寄存器说明请参考定时器输入输出选择寄存器。

29.3.6 A/D测试功能

IEC60730 标准要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、温度传感器的输出电压和内部基准电压进行 A/D 转换，确认 A/D 转换器是否正常运行。

能通过以下步骤确认模拟多路转换器：

① 通过ADTES寄存器选择ANIx引脚作为A/D转换对象（ADTES1、ADTES0=0、0）

② 对ANIx引脚进行A/D转换（转换结果1-1）。

③ 通过ADTES寄存器选择A/D转换器的负（-）基准电压作为A/D转换对象（ADTES1、ADTES0=1、0）。

④ 对A/D转换器的负（-）基准电压进行A/D转换（转换结果2-1）。

⑤ 通过ADTES寄存器选择ANIx引脚作为A/D转换对象（ADTES1、ADTES0=0、0）

⑥ 对ANIx引脚进行A/D转换（转换结果1-2）。

⑦ 通过ADTES寄存器选择A/D转换器的正（+）基准电压作为A/D转换对象（ADTES1、ADTES0=1、1）。

⑧ 对A/D转换器的正（+）基准电压进行A/D转换（转换结果2-2）。

⑨ 通过ADTES寄存器选择ANIx引脚作为A/D转换对象（ADTES1、ADTES0=0、0）

⑩ 对ANIx引脚进行A/D转换（转换结果1-3）。

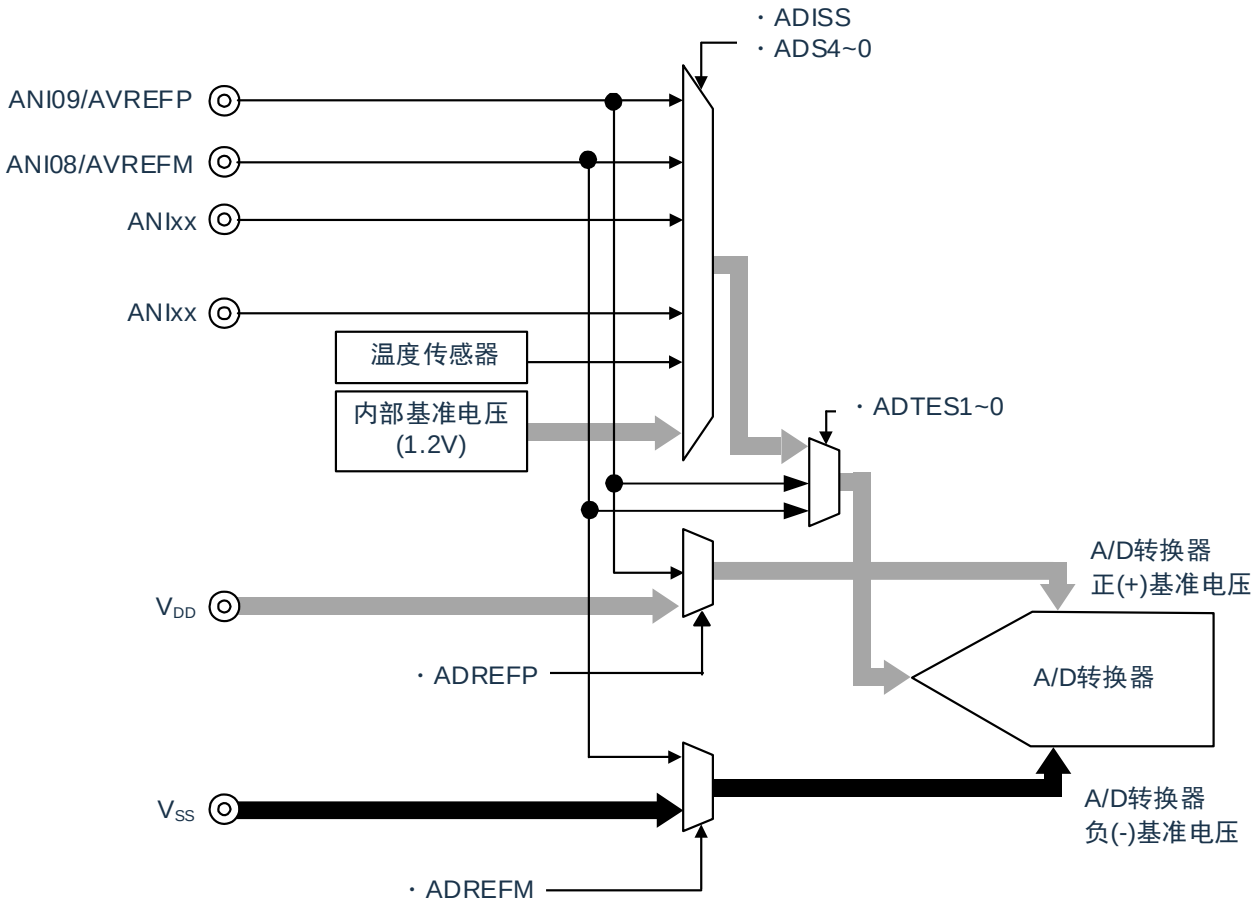
⑪ 确认“转换结果1-1”、“转换结果1-2”和“转换结果1-3”相同。

⑫ 确认“转换结果2-1”的A/D转换结果全部为“0”并且“转换结果2-2”的A/D转换结果全部为“1”。通过以上步骤，能选择模拟多路转换器以及确认布线没有断线。

备注：

1. 在①~⑩的转换过程中，如果模拟输入电压可变，就必须采用其他方法来确认模拟多路转换器。
2. 转换结果含有误差，因此必须在比较转换结果时要适当考虑误差。

图 29-11：A/D 测试功能的结构



29.3.6.1 A/D测试寄存器（ADTES）

此寄存器选择 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANIxx）、温度传感器的输出电压和内部基准电压（1.45V）作为 A/D 转换对象。

当用作 A/D 测试功能时，进行以下的设置：

- 在测量零刻度时，选择负（-）基准电压作为 A/D 转换对象。
- 在测量满刻度时，选择正（+）基准电压作为 A/D 转换对象。

寄存器说明请参考 11.2.10。

29.3.6.2 模拟输入通道指定寄存器（ADS）

此寄存器指定 A/D 转换的模拟电压的输入通道。

要通过 A/D 测试功能测量 ANIxx、温度传感器输出或者内部基准电压（1.45V）时，必须将 A/D 测试寄存器（ADTES）置“00H”。

寄存器说明请参考 11.2.7。

29.3.7 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 BAT32 微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

基地址：0x0050_0E4C

地址偏移：0x00
只读，其值在出厂时编写

U_ID[31:0]

地址偏移：0x28
只读，其值在出厂时编写

U_ID[63:32]

地址偏移：0x2C
只读，其值在出厂时编写

U_ID[95:64]

地址偏移：0x30
只读，其值在出厂时编写

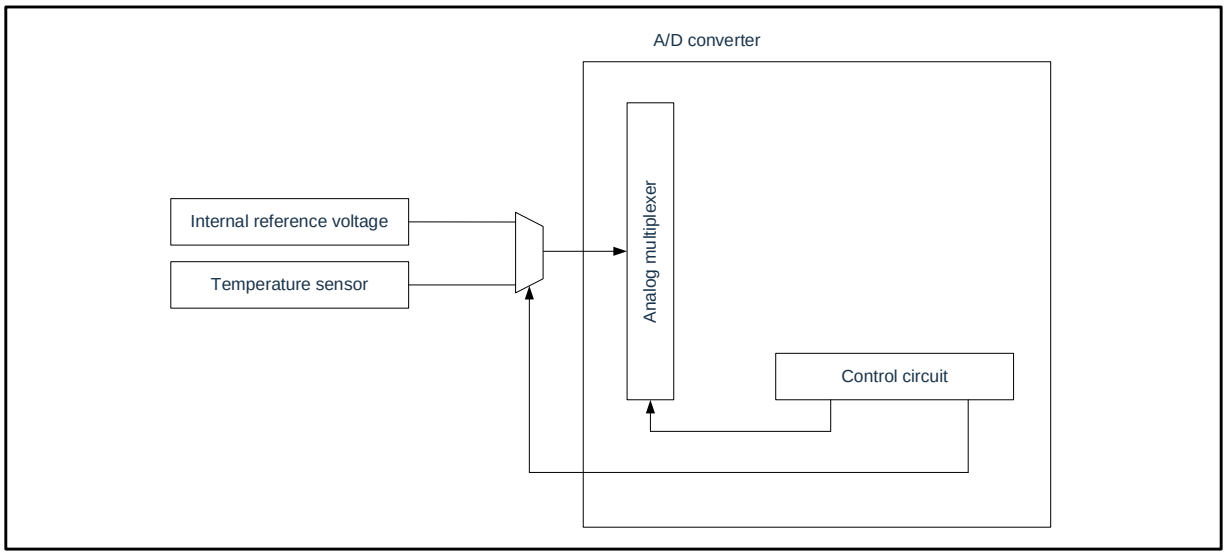
U_ID[127:96]

第30章 温度传感器和内部参考电压

30.1 温度传感器

片上的温度传感器可以对产品的核心温度进行测量和监控，从而保证产品的可靠运行。温度传感器输出的电压与核心温度成正比，并且电压和温度之间是线性关系。其输出电压提供给 ADC 进行转换。图 30-1 显示了温度传感器框图。

图 30-1：温度传感器框图



30.2 温度传感器的寄存器

30.2.1 温度传感器校准数据寄存器 TSN25

地址：0x500C6C

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
TSN25	-	-	-	-	TSN25[11:0]												-	R

只读寄存器，用于记录温度传感器的校准数据 1，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

30.2.2 温度传感器校准数据寄存器 TSN85

地址：0x500C68

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位后	R/W
TSN85	-	-	-	-	TSN85[11:0]												-	R

只读寄存器，用于记录温度传感器的校准数据 2，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

30.3 温度传感器的使用说明

30.3.1 温度传感器的使用原理

温度（T）与传感器电压输出（Vs）成正比，因此温度的计算公式如下：

T = (Vs – V1) / slope + T1

T：测量的温度（˚C）

Vs：温度传感器在温度测量时的输出电压（V）

T1：在第一个点进行实验测量的温度（˚C）

V1：温度传感器测量 T1 时的电压输出（V）

T2：在第二个点进行实验测量的温度（˚C）

V2：温度传感器测量 T2 时的电压输出（V）

Slope：温度传感器的温度斜率(V/°C)，slope = (V2 – V1) / (T2 – T1).

不同传感器的特性不同，因此我们建议测量以下两种不同的样品温度：

1. 使用A/D转换器测量温度传感器在温度T1时输出的电压V1。
2. 使用A/D转换器测量温度传感器在第二个温度T2时输出的电压V2。
3. 从两次结果中计算获得温度斜率(slope = (V2 – V1) / (T2 – T1))
4. 随后，通过将斜率代入温度特性的公式来获得温度(T =(Vs -V1) / slope + T1).

30.3.2 温度传感器的使用方法

方法一：在本产品中，TSN25 寄存器存储了在 T_A=25˚C和 V_{DD}=3.0V 条件下测量的温度传感器的电压转换值（CAL25）。TSN85 寄存器存储了在 T_A=85˚C和 V_{DD}=3.0v 条件下测量的温度传感器的电压转换值（CAL85）。利用这两组值可以计算出温度斜率：

slope = (V2 – V1) / (85 – 25).

V1 = 3.0 × CAL25 / 4096 [V]

V2 = 3.0 × CAL125 / 4096 [V]

利用以上结果，可根据以下公式计算温度：

T = (Vs – V1) / slope + 25 [˚C]

T：测量的温度（˚C）

Vs：使用 A/D 转换器得到的温度传感器在 T 温度时的输出电压（V）

方法二：如果使用“电气特性”中给出的温度斜率，可直接使用以下公式计算测量温度：

T = (Vs – V1) / slope + 25 [˚C]

备注：这种方法产生的温度比方法一测量的精度要低。

30.4 内部参考电压

Band-gap 参考电压(V_{BG})是一个内部的固定参考电压，不受外部供电的影响。V_{BG} 输出可以通过内部连接到 ADC 进行转换，因此 V_{DD} 可以通过 V_{BG} 的 ADC 转换结果计算出来。

由于制程的原因每个芯片的 V_{BG} 稍有不同，因此 V_{DD} 的计算会有一点偏差。本产品在出厂时内置了 V_{DD}=3.0V 时 V_{BG} 的 A/D 转换结果，用户可以通过这个值及当前 V_{BG} 的 A/D 转换结果精准计算出 V_{DD} 的电压值。

30.4.1 VDD校准数据寄存器VDDCDR

地址：0x500C64

符号	15	14	13	12	11 10 9 8 7 6 5 4 3 2 1 0	复位后	R/W
VDDCDR	-	-	-	-	VDDCDR[11:0]	-	R

只读寄存器，用于记录 V_{DD}=3.0V 时 V_{BG} 的 A/D 转换结果，在接通电源或者复位启动时自动载入，每颗芯片有自己的校准数据。

30.4.2 内部参考电压的使用说明

在本产品中，VDDCDR 寄存器存储了在 T_A=T_J=25˚C和 V_{DD}=3.0V 条件下测量的内部参考电压转换值。利用该结果，用户可根据以下公式计算当前 V_{DD} 电压：

V_{DD} = 3.0V*VDDCDR/CALVBG

CALVBG: 使用 A/D 转换器，将 V_{BG} 作为 ADC 输入通道，得到的当前 V_{BG} 转换结果。

第31章 选项字节

31.1 选项字节的功能

CMS32H6157 的闪存 000C0H~000C4H，500004H 为选项字节区。

选项字节由用户选项字节（000C0H~000C2H）和闪存数据保护选项字节（000C3H，500004H）构成。在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

注意：与是否使用各功能无关，必须设定选项字节。

31.1.1 用户选项字节（000C0H~000C2H）

(1) 000C0H

- 看门狗定时器的运行
  - 允许或者禁止计数器的运行。
  - 在睡眠/深度睡眠模式中允许或者停止计数器的运行。
- 看门狗定时器的上溢时间的设定
- 看门狗定时器的窗口打开期间的设定
- 看门狗定时器的间隔中断的设定
  - 使用或者不使用间隔中断。

(2) 000C1H

- LVD 运行模式的设定
  - 中断&复位模式
  - 复位模式
  - 中断模式
  - LVD 为 OFF（使用 RESETB 引脚的外部复位输入）。
- LVD 检测电平（VLVDH、VLVDL、VLVD）的设定

注意：当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过深度睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。

工作电压范围取决于用户选项字节（000C2H）的设定。

(3) 000C2H

- 高速内部振荡器的频率设定
  - 从 1MHz~32MHz 中选择。

31.1.2 闪存数据保护选项字节（000C3H，500004H）

- 片上调试时闪存数据保护的控制

  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。

31.2 用户选项字节的格式

图 31-1：用户选项字节（000C0H）的格式

地址：000C0H

符号	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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

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

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

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 （F_{IL}=36KHz(MAX.)）
0	0	0	$2^6/F_{IL}$（1.8ms）
0	0	1	$2^7/F_{IL}$（3.6ms）
0	1	0	$2^8/F_{IL}$（7.1ms）
0	1	1	$2^9/F_{IL}$（14.2ms）
1	0	0	$2^{11}/F_{IL}$（56.9ms）
1	0	1	$2^{13}/F_{IL}$（227.6ms）
1	1	0	$2^{14}/F_{IL}$（455.1ms）
1	1	1	$2^{16}/F_{IL}$（1820.4ms）

WDSTBYON	看门狗定时器的计数器运行控制（睡眠模式）
0	在睡眠模式中，停止计数器的运行[注2]。
1	在睡眠模式中，允许计数器的运行。

注 1：当 WDSTBYON 位为“0”时，与 WINDOW1 位和 WINDOW0 位的值无关，窗口打开期间为 100%。

备注：F_{IL}：低速内部振荡器的时钟频率。

图 31-2：用户选项字节（000C1H）的格式(1/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断&复位模式）

<table>
<tr><th colspan="3">检测电压</th><th colspan="8">选项字节的设定值</th></tr>
<tr><th colspan="2">V_{LVDH}</th><th>V_{LVDL}</th><th rowspan="2">VPOC2</th><th rowspan="2">VPOC1</th><th rowspan="2">VPOC0</th><th rowspan="2">LVIS1</th><th rowspan="2">LVIS0</th><th colspan="2">模式设定</th></tr>
<tr><th>上升</th><th>下降</th><th>下降</th><th>LVIMDS1</th><th>LVIMDS0</th></tr>
<tr><td>1.98V</td><td>1.94V</td><td rowspan="3">1.84V</td><td rowspan="9">0</td><td rowspan="3">0</td><td rowspan="3">1</td><td>1</td><td>0</td><td rowspan="9">1</td><td rowspan="9">0</td></tr>
<tr><td>2.09V</td><td>2.04V</td><td>0</td><td>1</td></tr>
<tr><td>3.13V</td><td>3.06V</td><td>0</td><td>0</td></tr>
<tr><td>2.61V</td><td>2.55V</td><td rowspan="3">2.45V</td><td rowspan="3">1</td><td rowspan="3">0</td><td>1</td><td>0</td></tr>
<tr><td>2.71V</td><td>2.65V</td><td>0</td><td>1</td></tr>
<tr><td>3.75V</td><td>3.67V</td><td>0</td><td>0</td></tr>
<tr><td>2.92V</td><td>2.86V</td><td rowspan="3">2.75V</td><td rowspan="3">1</td><td rowspan="3">1</td><td>1</td><td>0</td></tr>
<tr><td>3.02V</td><td>2.96V</td><td>0</td><td>1</td></tr>
<tr><td>4.06V</td><td>3.98V</td><td>0</td><td>0</td></tr>
<tr><td colspan="3">—</td><td colspan="8">禁止设定上述以外的值。</td></tr>
</table>

注意：必须给 bit4 写“1”。

备注：

1. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
2. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

图 31-2：用户选项字节（000C1H）的格式(2/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（复位模式）

检测电压		选项字节的设定值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.88V	1.84V	0	0	1	1	1	1	1
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设定上述以外的值。						

注意：必须给 bit4 写“1”。

备注：

1. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
2. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

图 31-2：用户选项字节（000C1H）的格式(3/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断模式）

检测电压		选项字节的设定值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.88V	1.84V	0	0	1	1	1	0	1
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—		禁止设定上述以外的值。						

注意：必须给 bit4 写“1”。

备注：

1. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
2. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

图 31-2：用户选项字节（000C1H）的格式(4/4)

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 为 OFF 时的设定（使用 RESETB 引脚的外部复位输入）

检测电压		选项字节的设定值						
V_{LVDH}							模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值						

注意：

1. 必须给 bit4 写“1”。
2. 当电源电压上升时，必须在电源电压达到数据手册的 AC 特性所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过睡眠模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

备注：

1. ×：忽略
2. 有关 LVD 电路的详细内容，请参照“第 28 章 电压检测电路”。
3. 检测电压是 TYP.值。详细内容请参照数据手册的 LVD 电路特性。

图 31-3：用户选项字节（000C2H）的格式

地址：000C2H

7	6	5	4	3	2	1	0
1	1	1	0	1	FRQSEL2	FRQSEL1	FRQSEL0

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

注意：

1. 必须给 bit7~5 写“1”，bit4 写“0”。
2. 工作频率范围和工作电压范围因闪存的各运行模式而不同。详细内容请参照数据手册的 AC 特性。

31.3 闪存数据保护选项字节的格式

闪存数据保护选项字节的格式如下所示。

图 31-4：闪存数据保护选项字节（000C3H）的格式

地址：000C3H

符号	7	6	5	4	3	2	1	0
	OCDEN[7:0]							

地址：500004H

符号	7	6	5	4	3	2	1	0
	OCDM[7:0]							

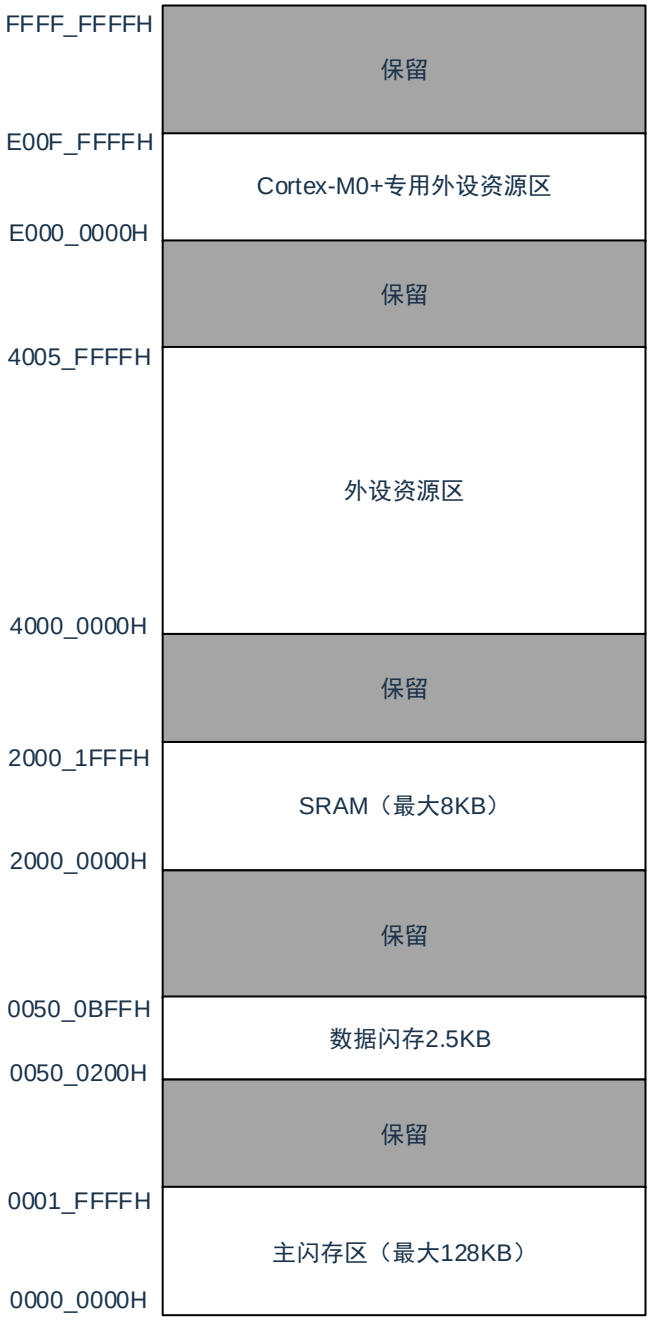
OCDM	OCDEN	闪存数据保护的控制
3C	C3	不允许通过 debugger 对闪存数据进行操作。
3C以外的值	C3	允许通过 debugger 对闪存数据进行 chip 全擦除操作，不允许进行读写操作。
上记以外		允许通过 debugger 对闪存数据进行读出/写入/擦除操作

第32章 FLASH控制

32.1 FLASH控制功能描述

本制品包含一颗 128KByte 容量的 FLASH 存储器，共划分为 256 个 Sector，每个 Sector 容量为 512Byte。可做为程序存储器，数据存储器。本模块支持对该存储器的擦除、编程以及读取操作。此外，本模块支持对 FLASH 存储器擦写的保护，以及控制寄存器的写保护。

32.2 FLASH存储器结构



32.3 控制FLASH的寄存器

控制 FLASH 的寄存器如下所示：

- Flash写保护寄存器(FLPROT)
- Flash操作控制寄存器（FLOPMD1,FLOPMD2）
- Flash擦除模式控制寄存器(FLERMD)
- Flash状态寄存器(FLSTS)
- Flash全片擦除时间控制寄存器(FLCERCNT)
- Flash页擦除时间控制寄存器（FLSERCNT）
- Flash写入时间控制寄存器（FLPROCNT）
- Flash 模式时间控制寄存器

32.3.1 Flash写保护寄存器(FLPROT)

Flash 保护寄存器用于对 FLASH 操作控制寄存器进行保护。

地址：0x40020020　　复位后：00000000H　　R/W

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

WRP	操作寄存器（FLOPMD1/FLOPMD2）写保护
0	不允许改写 FLOPMD1/ FLOPMD2
1	允许改写 FLOPMD1/ FLOPMD2

PRKEY[7:1]	WRP写保护
78h	允许改写 WRP
上记以外	不允许改写 WRP

32.3.2 FLASH操作控制寄存器（FLOPMD1,FLOPMD2）

Flash 操作控制寄存器，用于设定 FLASH 的擦除和写入操作。

地址：0x40020004　　复位后：00000000H　　R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	FLOPMD1[7:0]							

地址：0x40020008　　复位后：00H R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLOPMD2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	FLOPMD2[7:0]							

FLOPMD1	FLOPMD2	操作
55	AA	擦除
AA	55	写入
00	00	读出
上记以外		设定禁止

32.3.3 Flash擦除控制寄存器(FLERMD)

Flash 擦除控制寄存器，用于设定 FLASH 擦除操作的类型。

地址：0x4002000C　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
FLERMD	0	0	0	ERMD1	ERMD0	0	0	0

ERMD1	ERMD0	操作
0	0	sector 擦除
1	0	设定禁止
0	1	chip 擦除[注]
1	1	设定禁止

注：chip 擦除只擦除代码闪存区域，不擦除数据闪存区域。且 chip 擦除不支持硬件校验。

32.3.4 Flash状态寄存器(FLSTS)

通过状态寄存器可以查询 FLASH 控制器的状态。

地址：0x40020000　　复位后：00H R/W

符号	7	6	5	4	3	2	1	0
FLSTS	0	0	0	0	0	0	0	OVF[注]

OVF	FLASH 擦写操作完了标志
0	FLASH 擦写操作未完成
1	FLASH 擦写操作完成

注：OVF 需要软件写”1”进行清除。若不清除，不能进行下一次擦写操作。

EVF	FLASH 擦除硬件校验错误标志
0	FLASH 擦除后，硬件校验没有发生错误
1	FLASH 擦除后，硬件校验发生了错误

注：EVF 需要软件写”1”进行清除。

32.3.5 Flash全片擦除时间控制寄存器(FLCERCNT)

通过 FLCERCNT 寄存器可以设置 FLASH 全片擦除的时间。

地址：0x40020010　　复位后：不定R/W

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

Load	擦除时间设定的选择[注]
0	使用硬件设定的擦除时间
1	使用软件设定的擦除时间（FLCERCNT[9:0]）

注：在主时钟为内部高速 OCO 或者外部输入时钟＜=20M 时，可以使用硬件设定时间，不设定 FLCERCNT。

FLCERCNT[9:0]	软件擦除时间设定
Chip 擦除时间=（CERCNT*2048*Tfclk），需满足＞20ms 的硬件要求	

32.3.6 Flash页擦除时间控制寄存器（FLSERCNT）

通过 FLSERCNT 寄存器可以设置 FLASH 全片擦除的时间。

地址：0x40020014　　复位后：不定R/W

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

Load	擦除时间设定的选择[注]
0	使用硬件设定的擦除时间
1	使用软件设定的擦除时间（FLSERCNT[9:0]）

注：在主时钟为内部高速 OCO 或者外部输入时钟＜=20M 时，可以使用硬件设定时间，不设定 FLSERCNT。

FLSERCNT[9:0]	软件擦除时间设定
sector 擦除时间=（SERCNT*256*Tfclk），需满足＞4ms 的硬件要求	

32.3.7 Flash 写入时间控制寄存器（FLPROCNT）

通过 FLPROCNT 寄存器可以设置 FLASH WORD 写入的时间。

地址：0x4002001C　　复位后：不定 R/W

符号	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLPROCNT	Load1	-	-	-	-	-	-	FLPGSCNT[8:0]								
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Load0	-	-	-	-	-	-	FLPROCNT[8:0]								

Load0	写入时间（Tprog）设定[注]
0	使用硬件设定的写入时间
1	使用软件设定的写入时间（FLPROCNT[9:0]）

注：在主时钟为内部高速 OCO 或者外部输入时钟＜=20M 时，可以使用硬件设定时间，不设定 FLPROCNT。

FLPROCNT[8:0]	软件写入时间设定
写入时间=（PROCNT*4*Tfclk），需满足＞24us 的硬件要求	

Load1	写入动作建立时间（Tpgs）设定[注]
0	使用硬件设定的写入动作建立时间
1	使用软件设定的写入动作建立时间（FLPGSCNT8:0））

注：在主时钟为内部高速 OCO 或者外部输入时钟＜=20M 时，可以使用硬件设定时间，不设定 FLPGSCNT。

FLPGSCNT[8:0]	软件写入动作建立时间设定
写入动作建立时间=（PGSCNT*T_{FCLK}），需满足＞5us 的硬件要求	

32.3.8 Flash擦写保护控制寄存器（FLSECPR）

当 Sector 被保护时，对该 Sector 进行的擦写操作均无效。

地址：0x40020210　　复位后：00000000H　　R/W

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

KEY	寄存器 SECPR 写保护
5AA5F1	允许改写 SECPR[3:0]
上述以外	不允许改写 SECPR[3:0]

SECPR	寄存器 SECPR 写保护
0001	00_0000H~00_0FFFH 的 8 个 sector 不能擦写
0010	00_0000H~00_1FFFH 的 16 个 sector 不能擦写
0011	00_0000H~00_3FFFH 的 32 个 sector 不能擦写
0100	00_0000H~00_7FFFH 的 64 个 sector 不能擦写
0101	00_0000H~00_FFFFH 的 128 个 sector 不能擦写
上述以外	sector 没有保护，允许擦写

32.4 FLASH操作方法

32.4.1 页擦除（sector erase）

sector 擦除，擦除时间由硬件实现，也可以通过 FLSERCNT 配置。操作流程如下：

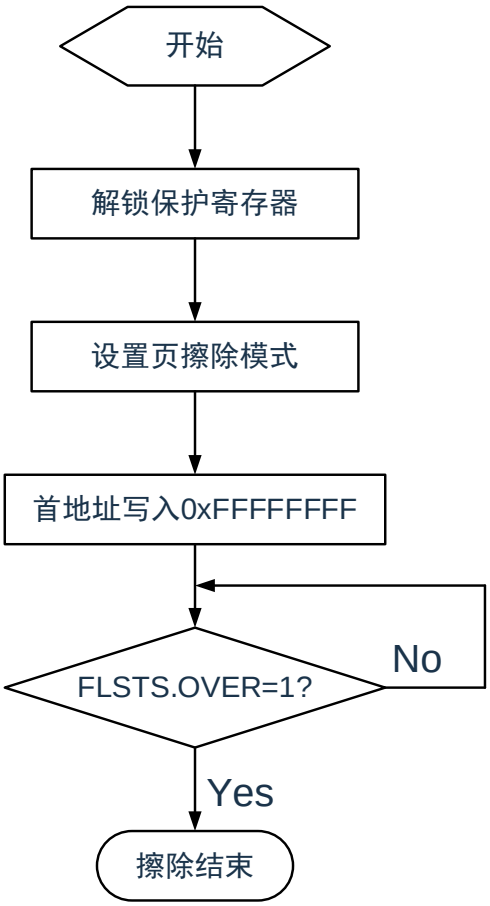
(1) 设置FLERMD.ERMD0为1’b0，选择sector擦除模式；

(2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，

(3) 向擦除目标sector的首地址写入任意数据。例：*（（unsigned long *）0x00000200）=0xffffffff。

(4) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示擦除操作完成。

(5) 进行下次操作前，软件置”1”来清除FLSTS。



32.4.2 全片擦除（chip erase）

chip 擦除，擦除时间由硬件实现，也可以通过 FLCERCNT 配置。操作流程如下：

(1) 设置FLERMD.ERMD0为1’b1，选择chip擦除模式；

(2) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0x55，FLOPMD2设置为0xAA，

(3) 向代码闪存区域的任意地址写入任意数据。

(4) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示擦除操作完成。

(5) 进行下次操作前，软件置”1”来清除FLSTS。

32.4.3 编程（word program）

word 编程，写入时间由硬件实现，也可以通过 PROCNT 配置。操作流程如下：

(1) 设置FLPROT为0xF1，解除FLOPMD的保护。然后将FLOPMD1设置为0xAA，FLOPMD2设置为0x55，

(2) 向目标地址写入相应的数据。

(3) 软件查询状态寄存器FLSTS.OVF,OVF=1时，表示写入操作完成。

(4) 进行下次操作前，软件置”1”来清除FLSTS。

32.5 闪存读取

本设备内置的 FLASH 支持的最快取指频率为 20MHz。当 HCLK 频率超过 20MHz 时，硬件会在 CPU 访问 FLASH 时插入 1 等待周期。

32.6 FLASH操作的注意事项

- FLASH存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。擦写参数的设置可以由硬件实现，也可通过修改参数寄存器进行软件修改；在使用内部高速OCO，MAINOSC/外部输入时钟=20M时，推荐使用硬件设置的擦写参数，无需设置参数寄存器。
- 如果擦写操作从FLASH内执行，则CPU会停止取指，硬件自动等待操作完成后，继续下一条指令。如果该操作从RAM里执行，CPU不会停止取指，当前可以继续下一条指令。
- 在FLASH处于编程操作中时，如果CPU执行进入深睡眠的指令，系统将等待编程动作结束后才会进入深睡眠。

第33章 修改履历

版本号	时间	修改内容
V0.1.0	2023 年 2 月	初始版本
V0.1.1	2023 年 3 月	1) 20.3.2 更正文字错误 2) 修改 32.3.3 Flash 擦除控制寄存器(FLERMD)章节描述 3) 修改 32.3.4 Flash 状态寄存器(FLSTS)章节中寄存器值 4) 修改 32.4.1 页擦除（sector erase）章节内容 5) 修改低速内部振荡器时钟值 6) 修改 15.2.3 章节中模拟输出电压公式
V0.1.2	2023 年 7 月	1) 4.2 章节：图 4-1 中低速内部振荡器频率修改为 32.768KHz（TYP.） 2) 4.3.10 章节：SUBCKSEL 寄存器 bit1 描述变更，“1”选择 32.768K，“0”选择 16.384 3) 31.2 章节：图 31-1 看门狗定时器的上溢时间变更 4) 29.3.5 章节：fIL 修改为 32.768KHz（TYP.） 5) 29.3.7 章节：UID 偏移地址变更 6) 17.3.5/17.4.3/17.3/17.4.1 章节：发送缓冲寄存器名称由 SOTB 修改为 SDRO 7) 17.3.6/17.4.2/17.4.4 章节：接收缓冲寄存器名称由 SIO 修改为 SDRI。与之相关的内容同步修改
V0.2.0	2023 年 10 月	1) 更正标题格式 2) 添加 11.2.2 章节中 ADM0 寄存器 bit 位的说明 3) 更正第 28 章中图 28-1 和图 28-6 的错误及 LVIS 的地址 4) 删除有关 PLL 时钟的描述 5) 更正内容中有误链接