

78K0R/KF3

16 位单片机微控制器

μPD78F1152

μPD78F1153

μPD78F1154

μPD78F1155

μPD78F1156

78K0R/KF3 具有片上调试功能。

大规模生产时不要使用本产品，因为在使用片上调试功能后，从 flash 存储器可重写的次数受到限制这一点来看，它的可靠性不能保证。NEC Electronics 不接受对于本产品的任何投诉。

[备忘录]

① 输入引脚处的电压波形

输入噪音或一个反射波引起的波形失真可能导致错误发生。如果由于噪音等的影响使CMOS设备的输入电压范围保持在 V_{IL} （最大值）和 V_{IH} （最小值）之间，设备可能发生错误。在输入电平固定时以及输入电平从 V_{IL} （最大值）过渡到 V_{IH} （最小值）时的传输期间，要防止散射噪声影响设备。

② 未使用的输入引脚的处理

CMOS设备的输入端保持开路可能导致误操作。如果一个输入引脚未被连接，则由于噪音等原因可能会产生内部输入电平，从而导致误操作。CMOS设备的操作特性与Bipolar或NMOS设备不同。CMOS设备的输入电平必须借助上拉或下拉电路固定在高电平或低电平。每一个未使用引脚都应该通过附加电阻连接到 V_{DD} 或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异，必须遵循与设备相关的规定和说明。

③ ESD 防护措施

如果MOS设备周围有强电场，将会击穿氧化栅极，从而影响设备的运行。因此必须采取措施，尽可能防止静电产生。一旦有静电，必须立即释放。对于环境必须有适当的控制。如果空气干燥，应当使用增湿器。建议避免使用容易产生静电的绝缘体。半导体设备的存放和运输必须使用抗静电容器、抗静电屏蔽袋或导电材料容器。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手带以保证良好接地。不能用手直接接触半导体设备。对于装配有半导体设备的PWB板也应采取类似的静电防范措施。

④ 初始化之前的状态

在上电时MOS设备的初始状态是不确定的。在刚刚上电之后，具有复位功能的MOS设备并没有被初始化。因此上电不能保证输出引脚的电平，I/O设置和寄存器的内容。设备在收到复位信号后才进行初始化。具有复位功能的设备在上电后必须立即进行复位操作。

⑤ 电源开关顺序

在一个设备的内部操作和外部接口使用不同的电源的情况下，按照规定，应先在接通内部电源之后再

接通外部电源。当关闭电源时，按照规定，先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒，可能会导致设备的内部组件过电压，产生异常电流，从而引起内部组件的误操作和性能的退化。

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

不要向没有加电的设备输入信号或提供I/O上拉电源。因为输入信号或提供I/O上拉电源将引起电流注入，从而引起设备的误操作，并产生异常电流，从而使内部组件退化。

每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

Windows 和 Windows NT 是 Microsoft Corporation 在美国/或其他国家的注册商标或商标。

PC/AT 是 International Business Machines Corporation 的商标。

EEPROM 是 NEC 电子公司的注册商标。

SuperFlash 是 Silicon Storage Technology, Inc 公司的注册商标。包括美国和日本在内的几个国家。

注意事项：该产品使用的 SuperFlash[®] 技术获得了 Silicon Storage Technology, Inc. 公司的授权。

- 本档所登载的内容有效期截至 2008 年 1 月。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本文件所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

“标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”： 计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”： 运输设备（汽车、火车、船舶等），交通信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”： 航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

(注)

- (1) 本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。
- (2) 本声明中的“本公司产品”是指所有由日本电气电子株式会社开发或制造，或为日本电气电子株式会社（定义如上）开发或制造的产品。

引言

读者对象 本手册适用于那些希望了解 78K0R/KF3 功能，并设计开发应用系统和程序的工程师。
目标产品如下。

78K0R/KF3: μ PD78F1152, 78F1153, 78F1154, 78F1155, 78F1156

目的 本手册用于帮助用户了解下面**组件**中描述的功能。

组件 78K0R/KF3 手册主要分为两个部分：手册和指令（与 78K0R 系列通用）。

78K0R/KF3 用户手册 (本手册)	78K0R 微控制器 用户手册 指令
-----------------------------------	---------------------------------

- 引脚功能
- 内部模块功能
- 中断
- 其它内置外设功能
- 电气特性（目标值）
- CPU 功能
- 指令集
- 指令扩展

手册使用方法 在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

- 如果读者要了解产品功能：
→ 请按**目录**顺序阅读本手册。标记“<R>”为主要修订处。在 PDF 文件中只要将“<R>”复制到“查找”中，就可以很容易找到修订处。
- 如何解释寄存器格式：
→ 括号中的位名在 RA78K0R 中被定义为保留字，并且在 CC78K0R 中直接通过 `#pragma sfr` 伪指令定义为一个特殊功能寄存器（sfr）变量。
- 如果读者希望了解 78K0R 系列指令的详细信息：
→ 可参阅 **78K0R 系列指令用户手册（U17792E）**。

规定	数据规则:	数据的高位部分在左边, 低位部分在右边
	有效低电平表示法:	$\overline{\text{xxx}}$ (在引脚和信号名称上划一条线)
	注:	文中用 注 标注的术语脚注
	注意事项:	需要特别关注的信息
	备注:	补充信息
	数的表示法:	二进制 ... xxxx 或 xxxxB
		十进制 ... xxxx
		十六进制 ... xxxxH

相关文档 本手册中指出的相关文档可能会包括初稿版本。但是并未对初稿版本作特殊标注。

设备文档

文档名称	文档编号
78K0R/KF3 用户手册	本手册
78K0R 微控制器指令用户手册	U17792E
78K0R 微控制器自编程库类型 01 用户手册	U18706E

开发工具的相关文档 (软件) (用户手册)

文档名称	文档编号	
CC78K0R Ver. 1.00 C 编译器	操作篇	U17838E
	语言篇	U17837E
RA78K0R Ver. 1.00 汇编包	操作篇	U17836E
	语言篇	U17835E
SM+ 系统模拟器	操作篇	U18010E
PM+ Ver. 6.20		U17990E
ID78K0R-QB Ver. 3.20 集成调试器	操作	U17839E

开发工具的相关文档 (硬件) (用户手册)

文档名称	文档编号
QB-MINI2 具有编程功能的片上调试仿真器	U18371E
QB-78K0RKX3 在线仿真器	U17866E

与 Flash 存储器编程相关文档

文档名称	文档编号
PG-FP4 Flash 存储器编程器用户手册	U15260E

注意事项 对以上列出的相关文档所做修改恕不另行通知, 在设计时请使用每个文档的最新版本。

其他文档

文档名称	文档编号
半导体选择指南 —产品和封装—	X13769X
半导体设备安装手册	注
NEC 半导体设备质量等级	C11531E
NEC 半导体设备可靠性/质量控制系统	C10983E
半导体设备防静电 ESD 保护指南	C11892E

注 浏览“半导体设备安装手册”网站 (<http://www.necel.com/pkg/en/mount/index.html>)。

注意事项 对以上列出的相关文档进行修改时恕不另行通知，在设计时请使用每个文档的最新版本。

目录

第一章 概述	17
1.1 特性.....	17
1.2 应用.....	18
1.3 订购信息.....	18
1.4 引脚配置（俯视图）.....	19
1.5 78K0R 系列单片机列表.....	21
1.6 框图.....	22
1.7 功能概述.....	23
第二章 引脚功能	25
2.1 引脚功能列表.....	25
2.2 引脚功能描述.....	31
2.2.1 P00~P06（端口 0）.....	31
2.2.2 P10~P17（端口 1）.....	32
2.2.3 P20~P27（端口 2）.....	33
2.2.4 P30~P31（端口 3）.....	33
2.2.5 P40~P47（端口 4）.....	34
2.2.6 P50~P55（端口 5）.....	35
2.2.7 P60~P67（端口 6）.....	36
2.2.8 P70~P77（端口 7）.....	36
2.2.9 P90（端口 9）.....	37
2.2.10 P110, P111（端口 11）.....	37
2.2.11 P120~P124（端口 12）.....	37
2.2.12 P130（端口 13）.....	38
2.2.13 P140~P145（端口 14）.....	38
2.2.14 AVREF0.....	39
2.2.15 AVREF1.....	39
2.2.16 AVSS.....	39
2.2.17 RESET.....	39
2.2.18 REGC.....	40
2.2.19 VDD, EVDD.....	40
2.2.20 VSS, EVSS.....	40
2.2.21 FLMD0.....	40
2.3 引脚 I/O 电路和未使用引脚的推荐连接方式.....	41
第三章 CPU 结构	46
3.1 存储器空间	46
3.1.1 内部程序存储器空间	53
3.1.2 镜像区域	55
3.1.3 内部数据存储器空间	56
3.1.4 特殊功能寄存器（SFR）区域.....	57
3.1.5 扩展特殊功能寄存器（第二 SFR: 第二特殊功能寄存器）区域	57
3.1.6 数据存储器寻址	58
3.2 处理器寄存器.....	63
3.2.1 控制寄存器.....	63
3.2.2 通用寄存器.....	65

3.2.3 ES 和 CS 寄存器	67
3.2.4 特殊功能寄存器 (SFR)	68
3.2.5 扩展特殊功能寄存器 (第二 SFR: 第二特殊功能寄存器)	74
3.3 指令地址寻址	80
3.3.1 相对寻址	80
3.3.2 立即寻址	80
3.3.3 表间接寻址	81
3.3.4 寄存器直接寻址	82
3.4 处理数据地址寻址	83
3.4.1 隐含寻址	83
3.4.2 寄存器寻址	83
3.4.3 直接寻址	84
3.4.4 短直接寻址	85
3.4.5 SFR 寻址	86
3.4.6 寄存器间接寻址	87
3.4.7 基址寻址	88
3.4.8 基址变址寻址	91
3.4.9 堆栈寻址	92
第四章 端口功能	93
4.1 端口功能	93
4.2 端口配置	96
4.2.1 端口 0	97
4.2.2 端口 1	103
4.2.3 端口 2	109
4.2.4 端口 3	110
4.2.5 端口 4	112
4.2.6 端口 5	120
4.2.7 端口 6	122
4.2.8 端口 7	125
4.2.9 端口 9	126
4.2.10 端口 11	127
4.2.11 端口 12	128
4.2.12 端口 13	131
4.2.13 端口 14	132
4.3 控制端口功能的寄存器	136
4.4 端口功能操作	143
4.4.1 写入 I/O 端口	143
4.4.2 从 I/O 端口读取	143
4.4.3 I/O 端口的操作	143
4.4.4 连接到不同的供电电压 (2.5V, 3V) 的外部设备	144
4.5 使用复用功能时端口模式寄存器和输出锁存的设置	146
4.6 端口寄存器 n (Pn) 的 1 位操作指令的注意事项	149
第五章 时钟发生器	150
5.1 时钟发生器的功能	150
5.2 时钟发生器的配置	151
5.3 控制时钟发生器的寄存器	153
5.4 系统时钟振荡器	167

5.4.1 X1 振荡器	167
5.4.2 XT1 振荡器	167
5.4.3 内部高速振荡器	170
5.4.4 内部低速振荡器	170
5.4.5 预分频器	170
5.5 时钟发生器操作	171
5.6 控制时钟	175
5.6.1 控制高速系统时钟的示例	175
5.6.2 控制内部高速振荡时钟的示例	178
5.6.3 控制副系统时钟示例	180
5.6.4 内部低速振荡时钟的控制示例	182
6.5.5 CPU 时钟状态迁移图	183
5.6.6 CPU 时钟切换之前的条件与切换之后的处理	188
5.6.7 CPU 时钟和主系统时钟切换所需的时间	189
5.6.8 时钟振荡停止前的条件	190
第六章 定时器阵列单元	191
6.1 定时器阵列单元的功能	191
6.1.1 独立操作时每通道的功能	191
6.1.2 和其他通道操作时每个通道的功能	192
6.1.3 LIN 总线支持功能（仅限通道 7）	192
6.2 定时器阵列单元的配置	193
6.3 控制定时器阵列单元的寄存器	198
6.4 通道输出（TO0n 引脚）控制	219
6.4.1 TO0n 引脚输出电路配置	219
6.4.2 TO0n 引脚输出设置	220
6.4.3 通道输出操作的注意事项	220
6.4.4 TO0n 位的集中操作	224
6.4.5 定时器中断和操作开始时 TO0n 引脚输出	225
6.5 通道输入（TI0n 引脚）控制	226
6.5.1 TI0n 沿检测电路	226
6.6 定时器阵列单元的基本功能	227
6.6.1 独立操作功能和组合操作功能的概述	227
6.6.2 组合操作功能的基本规则	227
6.6.3 组合操作功能的基本规则应用范围	228
6.7 定时器阵列单元作为通道的操作	229
6.7.1 作为间隔定时器/方波输出操作	229
6.7.2 作为外部事件计数器操作	233
6.7.3 用作分频操作	236
6.7.4 作为输入脉冲间隔测量操作	240
6.7.5 用作输入信号高/低电平宽度测量操作	244
6.8 定时器阵列单元的多通道操作	248
6.8.1 作为 PWM 功能操作	248
6.8.2 作为单脉冲输出功能操作	255
6.8.3 作为多路 PWM 输出功能操作	262
第七章 实时计数器	269
7.1 实时计数器的功能	269
7.2 实时计数器的配置	269

7.3 控制实时计数器的寄存器.....	271
7.4 实时计数器操作.....	283
7.4.1 启动实时计数器的操作.....	283
7.4.2 读/写实时计数器.....	284
7.4.3 设置实时计数器报警.....	286
第八章 看门狗定时器.....	287
8.1 看门狗定时器的功能.....	287
8.2 看门狗定时器的配置.....	288
8.3 控制看门狗定时器的寄存器.....	289
8.4 看门狗定时器的操作.....	290
8.4.1 看门狗定时器的控制操作.....	290
8.4.2 看门狗定时器溢出时间的设置.....	291
8.4.3 看门狗定时器窗口打开比例的设置.....	292
8.4.4 设置看门狗定时器间隔中断.....	293
第九章 时钟输出/蜂鸣器输出控制器.....	294
9.1 时钟输出/蜂鸣器输出控制器的功能.....	294
9.2 时钟输出/蜂鸣器输出控制器的配置.....	295
9.3 控制时钟输出/蜂鸣器输出控制器的寄存器.....	295
9.4 时钟输出/蜂鸣器输出控制器的操作.....	297
9.4.1 作为输出引脚操作.....	297
第十章 A/D 转换器.....	298
10.1 A/D 转换器的功能.....	298
10.2 A/D 转换器的配置.....	299
10.3 A/D 转换器使用的寄存器.....	301
10.4 A/D 转换器操作.....	309
10.4.1 A/D 转换器的基本操作.....	309
10.4.2 输入电压和转换结果.....	311
10.4.3 A/D 转换器操作模式.....	312
10.5 A/D 转换器特征表的阅读方法.....	314
10.6 A/D 转换器的注意事项.....	316
第十一章 D/A 转换器.....	320
11.1 D/A 转换器的功能.....	320
11.2 D/A 转换器的配置.....	320
11.3 D/A 转换器中使用的寄存器.....	321
11.4 D/A 转换器的操作.....	323
11.4.1 在普通模式下操作.....	323
11.4.2 在实时输出模式下操作.....	323
11.4.3 注意事项.....	324
第十二章 串行阵列单元.....	325
12.1 串行阵列单元的功能.....	325
12.1.1 3线串行输入/输出 (CSI00, CSI01, CSI10, CSI20).....	325
12.1.2 UART (UART0, UART1, UART2, UART3).....	326

12.1.3 简易 I2C (IIC10, IIC20)	326
12.2 串行阵列单元的配置	327
12.3 控制串行阵列单元的寄存器	332
12.4 操作停止模式	354
12.4.1 按照单元停止操作	354
12.4.2 按照通道停止操作	355
12.5 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的操作	357
12.5.1 主机发送	358
12.5.2 主机接收	367
12.5.3 主机发送/接收	373
12.5.4 从机发送	381
12.5.5 从机接收	390
12.5.6 从机发送/接收	396
12.5.7 计算发送时钟的频率	405
12.6 UART (UART0, UART1, UART2, UART3) 通信的操作	407
12.6.1 UART 发送	408
12.6.2 UART 接收	418
12.6.3 LIN 发送	425
12.6.4 LIN 接收	428
12.6.5 计算波特率	433
12.7 简易 I2C (IIC10, IIC20) 通信操作	437
12.7.1 地址区域发送	438
12.7.2 数据发送	443
12.7.3 数据接收	446
12.7.4 产生停止条件	449
12.7.5 计算发送速率	450
12.8 在错误情况下的处理过程	453
12.9 寄存器设置和引脚之间的关系	455
第十三章 串行接口 IIC0	462
13.1 串行接口 IIC0 的功能	462
13.2 串行接口 IIC0 的配置	465
13.3 控制串行接口 IIC0 的寄存器	468
13.4 I2C 总线模式功能	480
13.4.1 引脚配置	480
13.5 I2C 总线定义和控制方法	481
13.5.1 起始条件	481
13.5.2 地址	482
13.5.3 发送方向规范	482
13.5.4 发送时钟设置方法	483
13.5.5 应答信号 (ACK)	484
13.5.6 停止条件	486
13.5.7 等待	487
13.5.8 取消等待	489
13.5.9 中断请求 (INTIIC0) 产生时序和等待控制	490
13.5.10 地址匹配的检测方法	491
13.5.11 错误检测	491
13.5.12 扩展码	491
13.5.13 仲裁	492

13.5.14 唤醒功能	493
13.5.15 通信预约	494
13.5.16 注意事项	498
13.5.17 通信操作	499
13.5.18 产生 I2C 中断请求 (INTIICO) 的时序	507
13.6 时序图	528
第十四章 乘法器	535
14.1 乘法器的功能	535
14.2 乘法器的配置	536
14.3 乘法器的操作	537
第十五章 DMA 控制器	538
15.1 DMA 控制器	538
15.2 DMA 控制器的配置	539
15.3 控制 DMA 控制器的寄存器	542
15.4 DMA 控制器的操作	545
15.4.1 操作顺序	545
15.4.2 传输模式	546
15.4.3 DMA 传输中止	546
15.5 DMA 控制器设置的示例	547
15.5.1 CSI 连续发送	547
15.5.2 A/D 转换结果的连续捕获	549
15.5.3 UART 连续接收 + ACK 发送	551
15.5.4 通过 DWAITn 保持 DMA 传输挂起	553
15.5.5 软件强行中止	554
15.6 使用 DMA 控制器的注意事项	555
第十六章 中断功能	557
16.1 中断功能的类型	557
16.2 中断源及配置	558
16.3 控制中断功能的寄存器	561
16.4 中断服务操作	571
16.4.1 可屏蔽的中断响应	571
16.4.2 软件中断请求响应	573
16.4.3 多重中断嵌套	574
16.4.4 保持中断请求	577
第十七章 按键中断功能	578
17.1 按键中断的功能	578
17.2 按键中断的配置	578
17.3 控制按键中断的寄存器	579
第十八章 待机功能	580
18.1 待机功能及配置	580
18.1.1 待机功能	580
18.1.2 控制待机功能的寄存器	580

18.2 待机功能的操作	583
18.2.1 HALT 模式	583
18.2.2 STOP 模式	588
第十九章 复位功能	594
19.1 确认复位源的寄存器	602
第二十章 上电清零电路	603
20.1 上电清零电路的功能	603
20.2 上电清零电路的配置	604
20.3 上电清零电路的操作	604
20.4 上电清零电路使用注意事项	607
第二十一章 低电压检测电路	609
21.1 低电压检测电路的功能	609
21.2 低电压检测电路的配置	610
21.3 控制低电压检测电路的寄存器	610
21.4 低电压检测器的操作	615
21.4.1 用作复位时	616
21.4.2 用作中断时	622
21.5 低电压检测电路的注意事项	628
第二十二章 稳压器	632
22.1 稳压器概述	632
22.2 控制稳压器的寄存器	632
第二十三章 选项字节	634
23.1 选项字节的功能	634
23.1.1 用户选项字节 (000C0H~ 000C2H/010C0H ~010C2H)	634
23.1.2 片上调试选项字节 (000C3H/ 010C3H)	635
23.2 用户选项字节的格式	635
23.3 片上调试选项字节的格式	637
23.4 选项字节的设置	637
第二十四章 FLASH 存储器	638
24.1 用 Flash 存储器编程器写入	638
24.2 编程环境	640
24.3 通信模式	640
24.4 On Board 方式的引脚连接	641
24.4.1 FLMD0 引脚	641
24.4.2 TOOL0 引脚	642
24.4.3 RESET 引脚	642
24.4.4 端口引脚	643
24.4.5 REGC 引脚	643
24.4.6 X1 和 X2 引脚	643
24.4.7 供电电压	643
24.5 控制 Flash 存储器的寄存器	643

24.6 编程方法.....	644
24.6.1 控制 flash 存储器.....	644
24.6.2 Flash 存储器编程模式.....	644
24.6.3 选择通信模式.....	645
24.6.4 通信命令.....	645
24.7 安全设置.....	647
24.8 通过自编程进行 Flash 存储器编程.....	649
24.8.1 引导交换功能.....	651
24.8.2 Flash 防护窗口功能.....	653
第二十五章 片上调试功能.....	654
25.1 连接 QB-MINI2 到 78K0R/KF3.....	654
25.2 片上调试安全 ID.....	655
25.3 用户资源的保留.....	655
第二十六章 BCD 修正电路.....	657
26.1 BCD 修正电路功能.....	657
26.2 BCD 修正电路使用的寄存器.....	657
26.3 BCD 修正电路操作.....	658
第二十七章 指令集.....	660
27.1 操作列表使用规则.....	660
27.1.1 操作数标识符和标识方法.....	660
27.1.2 操作栏描述.....	661
27.1.3 标志操作栏的描述.....	662
27.1.4 前缀指令.....	662
27.2 操作列表.....	663
第二十八章 电气特性.....	680
第二十九章 封装图.....	726
附录 A 开发工具.....	728
A.1 软件包.....	731
A.2 语言处理软件.....	731
A.3 控制软件.....	732
A.4 Flash 存储器编程工具.....	732
A.4.1 当使用 flash 存储器编程器 FG-FP5, FL-PR5, FG-FP4 和 FL-PR4.....	732
A.4.2 当使用具有编程功能的片上调试仿真器 QB-MINI2.....	733
A.5 调试工具（硬件）.....	734
A.5.1 当使用在线仿真器 QB-78K0RKX3.....	734
A.5.2 当使用具有编程功能的片上调试仿真器 QB-MINI2 时.....	735
A.6 调试工具（软件）.....	735
附录 B 修订历史.....	736
B.1 本版中的主要修订之处.....	736
B.2 以往版本的修订历史.....	740

第一章 概述

1.1 特性

- 指令最短执行时间可以改变，范围在高速（0.05 μs ：@高速系统时钟的操作频率为 20 MHz）和超低速（61 μs ：@子系统时钟的操作频率为 32.768 kHz）之间
- 通用寄存器：8 位 \times 32 寄存器（8 位 \times 8 寄存器 \times 4 组）
- ROM, RAM 容量

产品型号	程序存储器 (ROM)		数据存储器 (RAM)
	项目		
$\mu\text{PD78F1152}$	Flash 存储器	64 KB	4 KB
$\mu\text{PD78F1153}$		96 KB	6 KB
$\mu\text{PD78F1154}$		128 KB	8 KB
$\mu\text{PD78F1155}$		192 KB	10 KB
$\mu\text{PD78F1156}$		256 KB	12 KB

- 内置单电源 flash 存储器（芯片擦除/块擦除/禁止写入功能）
- 自编程（具有引导交换功能/flash 屏蔽窗口功能）
- 片上调试功能
- 内置上电清零电路（POC）和低电压检测电路（LVI）
- 内置看门狗定时器（通过片上内部低速振荡时钟操作）
- 内置乘法器（16 位 \times 16 位）
- 内置按键中断功能
- 内置时钟输出/蜂鸣器输出控制器
- 片上 BCD 调整
- I/O 端口：70（N 沟道开漏：4）
- 定时器：10 通道
 - 16 位定时器：8 通道
 - 看门狗定时器：1 通道
 - 实时计数器：1 通道
- 串行接口
 - CSI：2 通道/UART：1 通道
 - CSI：1 通道/UART：1 通道/简易 I²C：1 通道
 - CSI：1 通道/UART：1 通道/简易 I²C：1 通道
 - UART（支持 LIN 总线）：1 通道
 - I²C：1 通道
- 10 位分辨率 A/D 转换器（ $AV_{REF0} = 2.3\sim 5.5\text{ V}$ ）：8 通道
- 8 位分辨率 D/A 转换器（ $AV_{REF1} = 1.8\sim 5.5\text{ V}$ ）：2 通道
- 供电电压： $V_{DD} = 1.8\sim 5.5\text{ V}$
- 工作环境温度： $T_A = -40\sim +85^\circ\text{C}$

1.2 应用

- 家用电器
 - 激光打印机马达
 - 衣物清洗设备
 - 空调设备
 - 冰箱
- 家庭音响系统
- 数码相机, 数码摄像机

1.3 订购信息

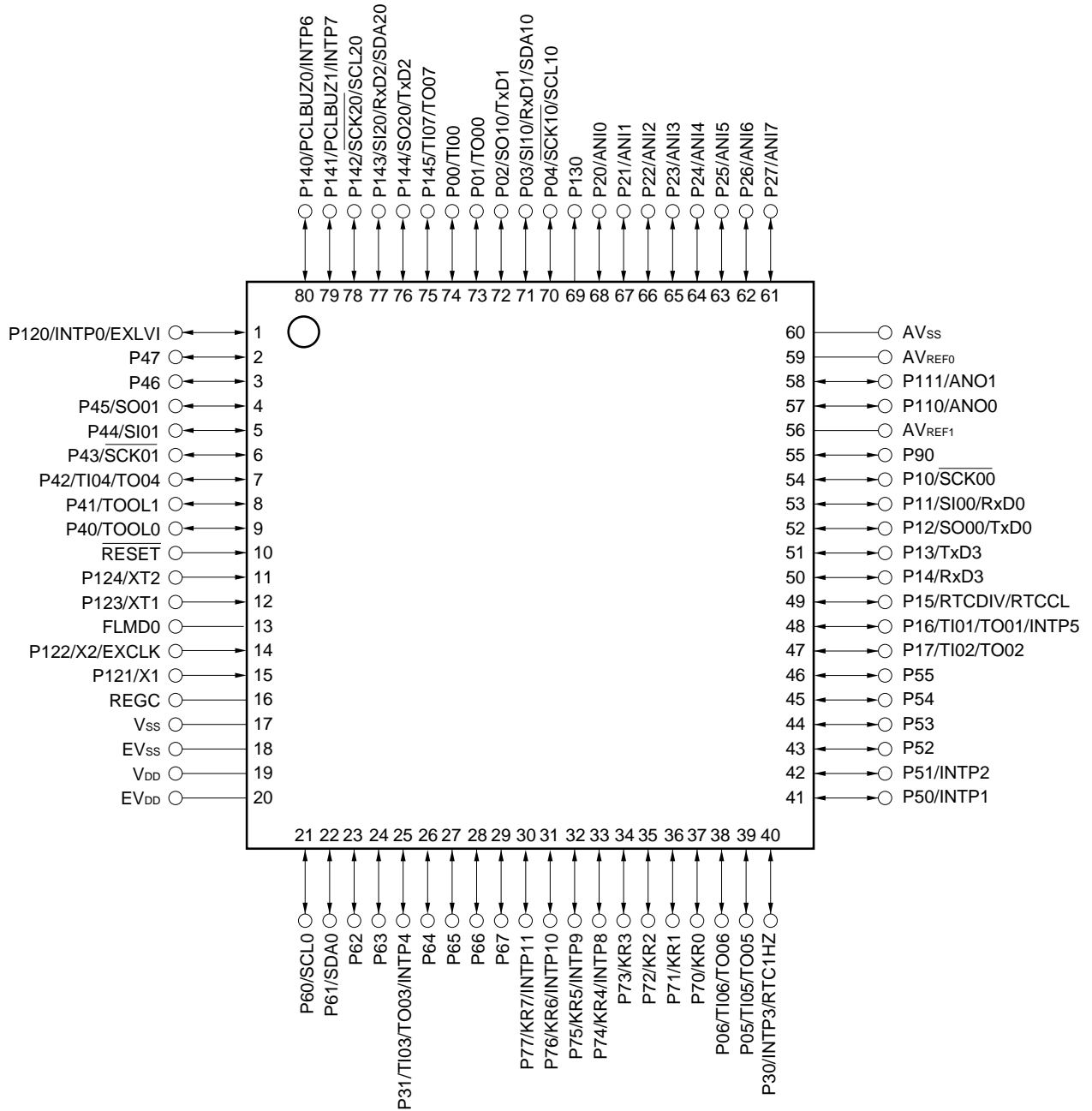
- **Flash 存储器版本**

产品型号	封装
μ PD78F1152GC-GAD-AX	80-引脚塑封 LQFP (14 × 14)
μ PD78F1153GC-GAD-AX	80-引脚塑封 LQFP (14 × 14)
μ PD78F1154GC-GAD-AX	80-引脚塑封 LQFP (14 × 14)
μ PD78F1155GC-GAD-AX	80-引脚塑封 LQFP (14 × 14)
μ PD78F1156GC-GAD-AX	80-引脚塑封 LQFP (14 × 14)
μ PD78F1152GK-GAK-AX	80-引脚塑封 LQFP (密脚距) (12 × 12)
μ PD78F1153GK-GAK-AX	80-引脚塑封 LQFP (密脚距) (12 × 12)
μ PD78F1154GK-GAK-AX	80-引脚塑封 LQFP (密脚距) (12 × 12)
μ PD78F1155GK-GAK-AX	80-引脚塑封 LQFP (密脚距) (12 × 12)
μ PD78F1156GK-GAK-AX	80-引脚塑封 LQFP (密脚距) (12 × 12)

注意事项 **78K0R/KF3** 具有片上调试功能。大规模生产时不要使用本产品，因为在使用片上调试功能后，从 flash 存储器可重写的次数受到限制这一点来看，它的可靠性不能保证。**NEC Electronics** 不接受对于本产品使用片上调试功能后的任何投诉。

1.4 引脚配置（俯视图）

- 80-引脚塑封 LQFP (14 × 14)
- 80-引脚塑封 LQFP (密脚距) (12 × 12)



- 注意事项**
1. 使 AV_{SS} 的电压与 EV_{SS} 和 V_{SS} 相同。
 2. 使 EV_{DD} 的电压与 V_{DD} 相同。
 3. 通过一个电容 (0.47~1 μF) 将 REGC 引脚连接到 V_{SS}。

<R>

备注 应用单片机时，必须减少其内部产生的噪声，推荐给两个 EV_{DD} 引脚提供独立的电源，并将两个 EV_{SS} 引脚分别连接到地线。

引脚标识

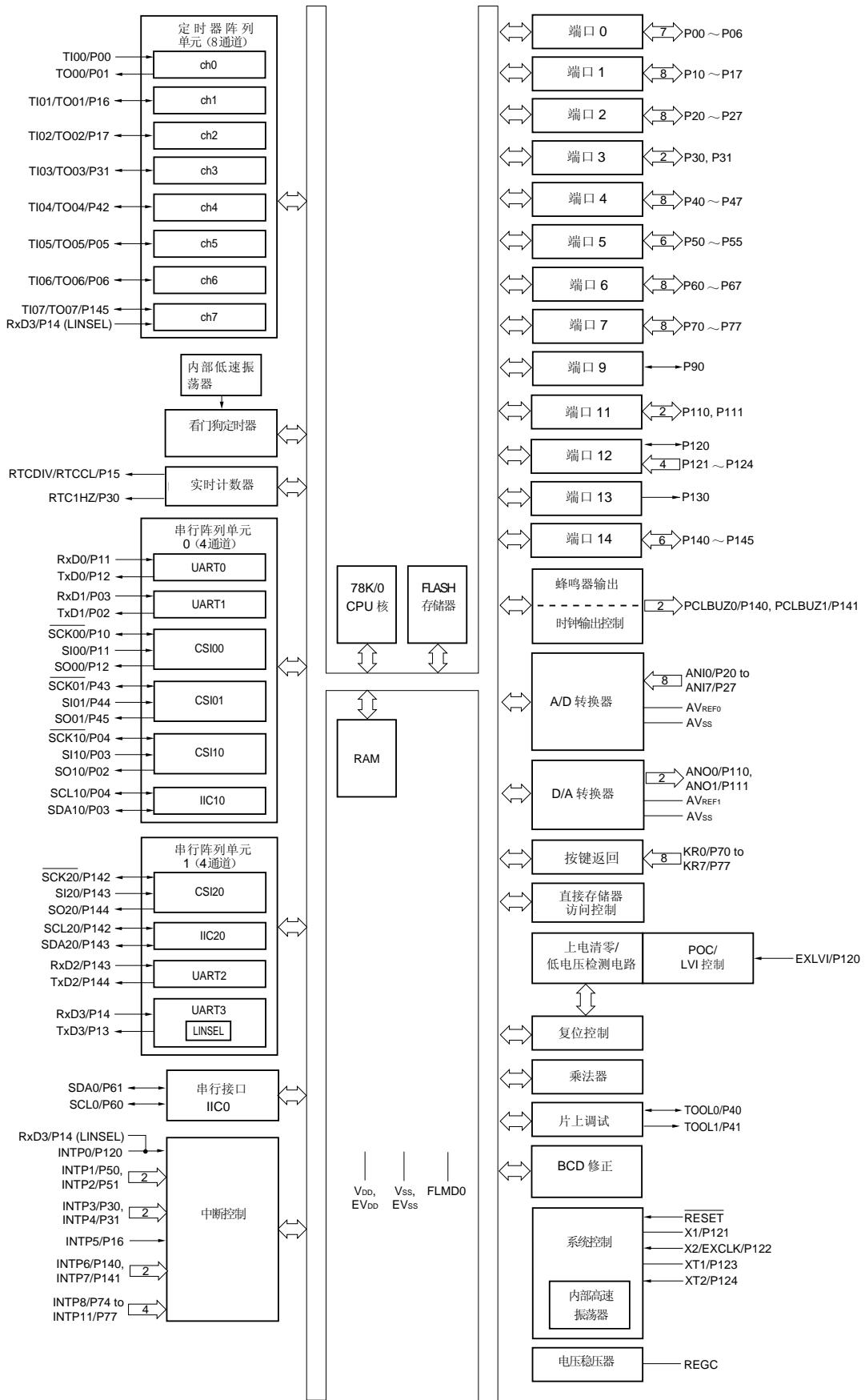
AN10~AN17:	模拟 输入	REGC:	调节器电容
ANO0, ANO1:	模拟 输出	RESET:	复位
AVREF0, AVREF1:	模拟参考电压	RTC1HZ:	实时计数器修正时钟 (1 Hz) 输出
AVSS :	模拟 地	RTCCL:	实时计数器时钟 (32 kHz 原始振荡) 输出
EVDD:	端口的供电电源	RTCDIV:	实时计数器时钟 (32 kHz 分频) 输出
EVSS:	端口的地	RxD0~RxD3:	接收数据
EXCLK:	外部时钟输入 (主系统时钟)	$\overline{\text{SCK00}}, \overline{\text{SCK01}},$ $\overline{\text{SCK10}}, \overline{\text{SCK20}}:$	串行时钟输入/输出
EXLVI:	为低电压检测电路提供的外部电压输入	SCL0, SCL10, SCL20:	串行时钟输入/输出
FLMD0:	Flash 编程模式	SDA0, SDA10, SDA20:	串行数据输入/输出
INTP0~INTP11:	外部中断输入	SI00, SI01,	
KR0~KR7:	按键返回	SI10, SI20:	串行数据输入
P00~P06:	端口 0	SO00, SO01,	
P10~P17:	端口 1	SO10, SO20:	串行数据输出
P20~P27:	端口 2	TI00~TI07:	定时器 输入
P30, P31:	端口 3	TO00~TO07:	定时器 输出
P40~P47:	端口 4	TOOL0:	对工具的数据输入/输出
P50~P55:	端口 5	TOOL1:	对工具的时钟 输出
P60~P67:	端口 6	TxD0~TxD3:	发送数据
P70~P77:	端口 7	VDD:	电源电压
P90:	端口 9	VSS:	地
P110, P111:	端口 11	X1, X2:	晶体振荡器 (主系统时钟)
P120~P124:	端口 12	XT1, XT2:	晶体振荡器 (子系统时钟)
P130:	端口 13		
P140~P145:	端口 14		
PCLBUZ0, PCLBUZ1:	可编程时钟 输出/ 蜂鸣器输出		

1.5 78K0R 系列单片机列表

ROM	RAM	78K0R/KE3	78K0R/KF3	78K0R/KG3	78K0R/KH3	78K0R/KJ3
		64 引脚	80 引脚	100 引脚	128 引脚	144 引脚
512 KB	30 KB	–	–	μPD78F1168 ^注	μPD78F1178 ^注	μPD78F1188 ^注
384 KB	24 KB	–	–	μPD78F1167 ^注	μPD78F1177 ^注	μPD78F1187 ^注
256 KB	12 KB	μPD78F1146	μPD78F1156	μPD78F1166	μPD78F1176 ^注	μPD78F1186 ^注
192 KB	10 KB	μPD78F1145	μPD78F1155	μPD78F1165	μPD78F1175 ^注	μPD78F1185 ^注
128 KB	8 KB	μPD78F1144	μPD78F1154	μPD78F1164	μPD78F1174 ^注	μPD78F1184 ^注
96 KB	6 KB	μPD78F1143	μPD78F1153	μPD78F1163	–	–
64 KB	4 KB	μPD78F1142	μPD78F1152	μPD78F1162	–	–

注 开发中

1.6 框图



1.7 功能概述

(1/2)

项目		μ PD78F1152	μ PD78F1153	μ PD78F1154	μ PD78F1155	μ PD78F1156
内部存储器	Flash 存储器 (支持自编程)	64 KB	96 KB	128 KB	192 KB	256 KB
	RAM	4 KB	6 KB	8 KB	10 KB	12 KB
存储器空间		1 MB				
主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体/陶瓷) 振荡, 外部主系统时钟输入 (EXCLK) 2~20 MHz: $V_{DD} = 2.7\sim 5.5$ V, 2~5 MHz: $V_{DD} = 1.8\sim 5.5$ V				
	内部高速振荡时钟	内部振荡 8 MHz (TYP.): $V_{DD} = 1.8\sim 5.5$ V				
副系统时钟 (振荡频率)	XT1 (晶体) 振荡 32.768 kHz (TYP.): $V_{DD} = 1.8\sim 5.5$ V					
内部低速振荡时钟 (WDT)		内部振荡 240 kHz (TYP.): $V_{DD} = 1.8\sim 5.5$ V				
通用寄存器		8 位 \times 32 寄存器 (8 位 \times 8 寄存器 \times 4 组)				
最小指令执行时间		0.05 μ s (高速系统时钟: $f_{MX} = 20$ MHz 操作)				
		0.125 μ s (内部高速振荡时钟: $f_{IH} = 8$ MHz (TYP.) 操作)				
		61 μ s (副系统时钟: $f_{SUB} = 32.768$ kHz 操作)				
指令集		<ul style="list-style-type: none"> • 8 位操作, 16 位操作 • 乘法 (16 位 \times 16 位) • 位操作 (Set, reset, test, 和 Boolean 操作) 等 				
I/O 端口		总计: 70 CMOS I/O: 61 CMOS 输入: 4 CMOS 输出: 1 N沟道开接 I/O (6 V 耐压): 4				
定时器		<ul style="list-style-type: none"> • 16 位定时器: 8 通道 • 看门狗定时器: 1 通道 • 实时计数器: 1 通道 				
		定时器输出	8 (PWM 输出: 7)			
		RTC 输出	2 <ul style="list-style-type: none"> • 1 Hz (副系统时钟: $f_{SUB} = 32.768$ kHz) • 512 Hz 或 16.384 kHz 或 32.768 kHz (副系统时钟: $f_{SUB} = 32.768$ kHz) 			
时钟输出/蜂鸣器输出		2 <ul style="list-style-type: none"> • 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (外围硬件时钟: $f_{MAIN} = 20$ MHz 操作) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (副系统时钟: $f_{SUB} = 32.768$ kHz 操作) 				
A/D 转换器		10 位分辨率 \times 8 通道 ($AV_{REF0} = 2.3\sim 5.5$ V)				
D/A 转换器		8 位分辨率 \times 2 通道 ($AV_{REF1} = 1.8\sim 5.5$ V)				

项目	μ PD78F1152	μ PD78F1153	μ PD78F1154	μ PD78F1155	μ PD78F1156
串行接口	<ul style="list-style-type: none"> • UART 支持 LIN 总线：1 通道 • CSI：2 通道/UART：1 通道 • CSI：1 通道/UART：1 通道/简易 I²C：1 通道 • CSI：1 通道/UART：1 通道/简易 I²C：1 通道 • I²C 总线：1 通道 				
乘法器	16 位 × 16 位 = 32 位				
DMA 控制器	2 通道				
向量中断源	内部	28			
	外部	13			
按键中断	通过按键输入引脚（KR0~KR7）的下降沿检测发生的按键中断（INTKR）。				
复位	<ul style="list-style-type: none"> • 由 $\overline{\text{RESET}}$ 引脚引起的复位 • 由看门狗定时器引起的内部复位 • 由上电清零引起的内部复位 • 由低电压检测电路引起的内部复位 • 由非法指令执行引起的内部复位^{注 2} 				
片上调试功能	提供				
供电电压	$V_{DD} = 1.8 \sim 5.5 \text{ V}$				
工作环境温度	$T_A = -40 \sim +85^\circ\text{C}$				
封装	80 引脚塑封 LQFP（14 × 14）（0.65 mm 间距）				
	80 引脚塑封 LQFP（密脚距）（12 × 12）（0.5 mm 间距）				

注

当执行指令码 FFH 时，会产生非法指令。

由执行非法指令引起的复位不能通过在线仿真器或片上调试仿真器产生。

第二章 引脚功能

2.1 引脚功能列表

有四种类型的引脚 I/O 缓冲式供电电源：AV_{REF0}，AV_{REF1}，EV_{DD}，和 V_{DD}。下表显示了这些供电电源与引脚之间的关系。

表 2-1. 引脚 I/O 缓冲式供电电源

供电电源	对应的引脚
AV _{REF0}	P20~P27
AV _{REF1}	P110, P111
EV _{DD}	<ul style="list-style-type: none">• 除 P20~P27, P110, P111, 和 P121~P124 以外的端口引脚• RESET 引脚和 FLMD0 引脚
V _{DD}	<ul style="list-style-type: none">• P121~P124• 端口引脚以外的引脚（除 RESET 引脚和 FLMD0 引脚之外）

(1) 端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 7 位 I/O 端口。 P03 和 P04 的输入可被设置到 TTL 缓冲器。 P02~P04 的输出可设置到 N-ch 漏极开路输出 (V _{DD} 耐压)。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05				TI05/TO05
P06				TI06/TO06
P10	I/O	端口 1。 8 位 I/O 端口。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	$\overline{\text{SCK00}}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20~P27	I/O	端口 2。 8 位 I/O 端口。 可以位操作选择输入/输出模式。	数字输入端口	ANI0~ANI7
P30	I/O	端口 3。 2 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40 ^注	I/O	端口 4。 8 位 I/O 端口。 P43 和 P44 的输入可被设置到 TTL 缓冲器。 P43 和 P45 的输出可设置到 N-ch 漏极开路输出 (V _{DD} 耐压) 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				$\overline{\text{SCK01}}$
P44				SI01
P45				SO01
P46				-
P47				-
P50	I/O	端口 5。 6 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	INTP1
P51				INTP2
P52				-
P53				-
P54				-
P55				-

注 如果通过选项字节允许片上调试, 请务必确保对 P40/TOOL0 引脚外部上拉 (参见 2.2.5 P40~P47 (端口 4) 中的**注意事项**)。

(1) 端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
P60	I/O	端口 6。 8 位 I/O 端口。 P60~P63 的输出可设置为 N-ch 漏极开路输出 (6 V 耐压)。 可以位操作选择输入/输出模式。 只有 P64~P67 可以通过软件设置, 定义是否使用内置上拉电阻。	输入端口	SCL0
P61				SDA0
P62				-
P63				-
P64				-
P65				-
P66				-
P67				-
P70~P73	I/O	端口 7。 8 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	KR0~KR3
P74~P77				KR4/INTP8 ~ KR7/INTP11
P90	I/O	端口 9。 1 位 I/O 端口。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	-
P110	I/O	端口 11。 2 位 I/O 端口。 可以位操作选择输入/输出模式。	输入端口	ANO0
P111				ANO1
P120	I/O	端口 12。 1 位 I/O 端口和 4 位输入端口。 仅有 P120 可以通过软件设置, 定义是否使用内置上拉电阻。	输入端口	INTP0/EXLVI
P121				X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	输出	端口 13, 1 位输出端口。	输出端口	-
P140	I/O	端口 14。 6 位 I/O 端口。 P142 和 P143 的输入可被设置到 TTL 缓冲器。 P142~P144 的输出可设置为 N-ch 漏极开路输出 (V _{DD} 耐压)。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07

(2) 非端口功能 (1/3)

功能名称	I/O	功能	复位后	复用功能
ANI0~ANI7	输入	A/D 转换器模拟输入	数字输入端口	P20~P27
ANO0	输出	D/A 转换器模拟输出	输入端口	P110
ANO1	输出	D/A 转换器模拟输出	输入端口	P111
EXLVI	输入	为外部低电压检测提供的电压输入	输入端口	P120/INTP0
INTP0	输入	定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入	输入端口	P120/EXLVI
INTP1				P50
INTP2				P51
INTP3				P30/RTC1HZ
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP6				P140/PCLBUZ0
INTP7				P141/PCLBUZ1
INTP8				P74/KR4~P77/KR7
INTP9				
INTP10				
INTP11				
KR0~KR3	输入	按键中断输入	输入端口	P70~P73
KR4~KR7				P74/INTP8~ P77/INTP11
PCLBUZ0	输出	时钟输出/蜂鸣器 输出	输入端口	P140/INTP6
PCLBUZ1				P141/INTP7
REGC	-	连接调节器输出（2.5 V）稳定电容用于内部操作 通过一个电容（0.47~1 μ F）连接到 V _{SS} 。	-	-
<R> RTCDIV	输出	实时计数器时钟（32 kHz 分频）输出	输入端口	P15/RTCCL
RTCCL	输出	实时计数器时钟（32 kHz 原始振荡）输出	输入端口	P15/RTCDIV
RTC1HZ	输出	实时计数器修正时钟（1 Hz）输出	输入端口	P30/INTP3
RESET	输入	系统复位输入	-	-
RxD0	输入	串行数据输入到 UART0	输入端口	P11/SI00
RxD1	输入	串行数据输入到 UART1	输入端口	P03/SI10/SDA10
RxD2	输入	串行数据输入到 UART2	输入端口	P143/SI20/SDA20
RxD3	输入	串行数据输入到 UART3	输入端口	P14
SCK00	I/O	CSI00, CSI01, CSI10, CSI11, CSI20 和 CSI21 的时钟输入/输出	输入端口	P10
SCK01				P43
SCK10				P04/SCL10
SCK20				P142/SCL20

(2) 非端口功能 (2/3)

功能名称	I/O	功能	复位后	复用功能
SCL0	I/O	I ² C 的时钟输入/输出	输入端口	P60
SCL10	I/O	简易 I ² C 的时钟输入/输出	输入端口	P04/SCK10
SCL20				P142/SCK20
SDA0	I/O	I ² C 的串行数据 I/O	输入端口	P61
SDA10		简易 I ² C 的串行数据 I/O	输入端口	P03/SI10/RxD1
SDA20			输入端口	P143/SI20/RxD2
SI00	输入	串行数据输入 到 CSI00, CSI01, CSI10 和 CSI20	输入端口	P11/RxD0
SI01				P44
SI10				P03/RxD1/SDA10
SI20				P143/RxD2/SDA20
SO00	输出	CSI00, CSI01, CSI10 和 CSI20 的串行数据输出	输入端口	P12/TxD0
SO01				P45
SO10				P02/TxD1
SO20				P144/TxD2
TI00	输入	外部计数时钟输入到 16 位定时器 00	输入端口	P00
TI01		外部计数时钟输入到 16 位定时器 01		P16/TO01/INTP5
TI02		外部计数时钟输入到 16 位定时器 02		P17/TO02
TI03		外部计数时钟输入到 16 位定时器 03		P31/TO03/INTP4
TI04		外部计数时钟输入到 16 位定时器 04		P42/TO04
TI05		外部计数时钟输入到 16 位定时器 05		P05/TO05
TI06		外部计数时钟输入到 16 位定时器 06		P06/TO06
TI07		外部计数时钟输入到 16 位定时器 07		P145/TO07
TO00	输出	16 位定时器 00 输出	输入端口	P01
TO01		16 位定时器 01 输出		P16/TI01/INTP5
TO02		16 位定时器 02 输出		P17/TI02
TO03		16 位定时器 03 输出		P31/TI03/INTP4
TO04		16 位定时器 04 输出		P42/TI04
TO05		16 位定时器 05 输出		P05/TI05
TO06		16 位定时器 06 输出		P06/TI06
TO07		16 位定时器 07 输出		P145/TI07
TxD0	输出	UART0 的串行数据输出	输入端口	P12/SO00
TxD1		UART1 的串行数据输出		P02/SO10
TxD2		UART2 的串行数据输出		P144/SO20
TxD3		UART3 的串行数据输出		P13
X1	-	连接主系统时钟振荡器	输入端口	P121
X2	-		输入端口	P122/EXCLK
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2
XT1	-	连接副系统时钟振荡器	输入端口	P123
XT2	-		输入端口	P124

(2) 非端口功能 (3/3)

功能名称	I/O	功能	复位后	复用功能
V _{DD}	-	正向供电电源 (P121 ~ P124 和端口除外)	-	-
EV _{DD}	-	端口的正向供电电源 (P20~P27, P110, P111, P121~P124 除外)	-	-
AV _{REF0}	-	<ul style="list-style-type: none"> • A/D 转换器参考电压输入 • P20~P27 和 A/D 转换器的正向供电电源 	-	-
AV _{REF1}	-	<ul style="list-style-type: none"> • D/A 转换器参考电压输入 • P110, P111, D/A 转换器的正向供电电源 	-	-
VSS	-	地电位 (P121~P124 和其他端口除外)	-	-
EV _{SS}	-	端口地电位 (P20~P27, P110, P111 和 P121~P124 除外)	-	-
AV _{SS}	-	A/D 转换器, D/A 转换器, P20~P27, P110 和 P111 的地电位	-	-
FLMD0	-	Flash 存储器编程模式设置	-	-
TOOL0	I/O	Flash 存储器编程器/调试器的数据 I/O	输入端口	P40
TOOL1	输出	调试器的时钟输出	输入端口	P41

2.2 引脚功能描述

2.2.1 P00~P06（端口 0）

P00~P06 功能作为 7 位 I/O 端口。这些引脚也可用于定时器 I/O、串行接口数据 I/O、和时钟 I/O。

使用端口输入模式寄存器 0（PIM0），可以按位指定 P03 和 P04 引脚的输入通过普通输入缓冲器或 TTL 输入缓冲器。

使用端口输出模式寄存器 0（POM0），P02~P04 引脚输出可按位指定为普通 CMOS 输出或 N-ch 漏极开路输出（V_{DD} 耐压）。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P00~P06 功能作为 7 位 I/O 端口。通过端口模式寄存器 0（PM0），P00~P06 可按位设置为输入端口或输出端口。由上拉电阻选择寄存器 0（PU0）指定内置上拉电阻的使用。

(2) 控制模式

P00~P06 可用于定时器 I/O、串行接口数据 I/O、和时钟 I/O。

(a) T100, T105, T106

这些引脚用于输入 16 位定时器 00, 05, 06 的外部计数时钟/捕捉触发。

(b) T000, T005, T006

这些是 16 位定时器 00, 05, 06 的输出引脚。

(c) S110

这是串行接口 CSI10 的串行数据输入引脚。

(d) S010

这是串行接口 CSI10 的串行数据输出引脚。

(e) $\overline{\text{SCK10}}$

这是串行接口 CSI10 的串行时钟 I/O 引脚。

(f) TxD1

这是串行接口 UART1 的串行数据输出引脚。

(g) RxD1

这是串行接口 UART1 的串行数据输入引脚。

(h) SDA10

这是简易 I²C 串行接口的串行数据 I/O 引脚。

(i) SCL10

这是简易 I²C 串行接口的串行时钟 I/O 引脚。

注意事项 要使用 P02/SO10/TxD1 和 P04/ $\overline{\text{SCK10}}$ /SCL10 作为通用端口，将串行通信操作设置寄存器 02 (SCR02) 设置为默认状态 (0087H)。另外，将端口输出模式寄存器 0 (POM0) 清除为 00H。

2.2.2 P10~P17 (端口 1)

P10~P17 功能作为 8 位 I/O 端口。这些引脚也可作为外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O、和实时计数器时钟输出。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P10~P17 功能作为 8 位 I/O 端口。通过端口模式寄存器 1 (PM1)，P10~P17 可按位设置为输入或输出端口。由上拉电阻选择寄存器 1 (PU1) 指定内置上拉电阻的使用。

(2) 控制模式

P10~P17 可作为外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O、和实时计数器时钟输出。

(a) SI00

这是串行接口 CSI10 的串行数据输入引脚。

(b) SO00

这是串行接口 CSI10 的串行数据输出引脚。

(c) $\overline{\text{SCK00}}$

这是串行接口 CSI10 的串行时钟 I/O 引脚。

(d) RxD0

这是串行接口 UART0 的串行数据输入引脚。

(e) RxD3

这是串行接口 UART3 的串行数据输入引脚。

(f) TxD0

这是串行接口 UART0 的串行数据输出引脚。

(g) TxD3

这是串行接口 UART3 的串行数据输出引脚。

(h) TI01, TI02

此引脚用于输入 16 位定时器 01 和 02 的外部计数时钟/捕捉触发。

(i) TO01, TO02

这些是 16 位定时器 01 和 02 的输出引脚。

(j) INTP5

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(k) RTCDIV

这是实时计数器时钟（32 kHz，分频）输出引脚。

(l) RTCCL

这是实时计数器时钟（32 kHz,原始振荡）输出引脚。

- 注意事项**
1. 要将 P10/SCK00 和 P12/SO00/TxD0 作为通用端口使用，将串行通信操作设置寄存器 00（SCR00）设置到默认状态（0087H）。
 2. 不要同时允许 RTCCL 和 RTCDIV 输出。

2.2.3 P20~P27（端口 2）

P20~P27 功能作为 8 位 I/O 端口。这些引脚也可作为 A/D 转换器模拟输入功能使用。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P20~P27 功能作为 8 位 I/O 端口。通过端口模式寄存器 2（PM2），P20~P27 可按位设置为输入或输出端口。

(2) 控制模式

P20~P27 作为 A/D 转换器模拟输入引脚（ANI0~ANI7）。当使用这些引脚作为模拟输入引脚时，参见 10.6 (5) ANI0/P20~ANI7/P27。

注意事项 ANI0/P20~ANI7/P27 在复位释放后被设置为数字输入（通用端口）。

2.2.4 P30~P31（端口 3）

P30~P31 是 2 位 I/O 端口。P30 和 P31 引脚也可作为外部中断请求输入、定时器 I/O 和实时计数器修正时钟输出操作。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P30 和 P31 是 8 位 I/O 端口。通过端口模式寄存器 3（PM3），P30 和 P31 可按位设置为输入或输出端口。由上拉电阻选择寄存器 3（PU3）指定内置上拉电阻的使用。

(2) 控制模式

P30 和 P31 功能是外部中断请求输入、定时器 I/O 实时计数器修正时钟输出操作。

(a) INTP3, INTP4

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) TI03

此引脚用于输入 16 位定时器 03 的外部计数时钟/捕捉触发。

(c) TO03

这是 16 位定时器 03 的输出引脚。

(d) RTC1HZ

这是实时计数器修正时钟（1 Hz）输出引脚。

2.2.5 P40~P47（端口 4）

P40~P47 是 8 位 I/O 端口。这些引脚可作为串行接口数据 I/O、时钟 I/O、flash 存储器编程器/调试器的数据 I/O、时钟输出和定时器 I/O 操作。

使用端口输入模式寄存器 4（PIM4），可以按位指定 P43 和 P44 引脚的输入通过普通输入缓冲器或 TTL 输入缓冲器。

使用端口输出模式寄存器 4（POM4），P43 和 P45 引脚输出可按位指定为普通 CMOS 输出或 N-ch 漏极开路输出（V_{DD} 耐压）。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P40~P47 是 8 位 I/O 端口。通过端口模式寄存器 4（PM4），P40~P47 可按位设置为输入或输出端口。由上拉电阻选择寄存器 4（PU4）指定内置上拉电阻的使用。

当允许片上调试（通过选项字节）时确保连接一个外部上拉电阻到 P40。

(2) 控制模式

P40~P47 可作为串行接口数据 I/O、时钟 I/O、flash 存储器编程器/调试器的数据 I/O、时钟输出和定时器 I/O 操作。

(a) TOOL0

这是 flash 存储器编程器/调试器的数据 I/O。

当允许片上调试（下拉禁止）时，请务必确保将此引脚加外部上拉。

(b) TOOL1

这是为调试器输出时钟的引脚。

当使用片上调试功能时，通过调试器模式设置 P41/TOOL1 引脚可用作如下操作。

1 线模式：可用作端口（P41）。

2 线模式：可用作 TOOL1 引脚，并且不能作为端口（P41）使用。

(c) T104

此引脚用于输入 16 位定时器 04 的外部计数时钟/捕捉触发。

(d) T004

这是 16 位定时器 04 的输出引脚。

(e) SCK01

这是串行接口 CSI01 的串行时钟 I/O 引脚。

(f) SI01

这是串行接口 CSI01 的串行数据输入引脚。

(g) SO01

这是串行接口 CSI01 的串行数据输出引脚。

注意事项 P40/TOOL0 引脚功能可以是下面 (a) 到 (c) 中的描述。

在 (b) 或 (c) 的情况下，确保指定的条件。

(a) 在普通操作模式下，或者通过选项字节（000C3H）禁止片上调试（OCDENSET = 0）时

=> 此引脚用作端口引脚（P40）。

(b) 在普通操作模式下，或者通过选项字节（000C3H）允许片上调试（OCDENSET = 1）时

=> 通过一个外接电阻将此引脚连接到 EV_{DD}，并且在复位释放前此引脚一直输入高电平。

(c) 当使用片上调试功能时，或在 flash 存储器编程器写入模式下时

=> 此引脚作为 TOOL0 使用。

直接将此引脚连接到片上调试仿真器或 flash 存储器编程器，或通过一个外接电阻连接到 EV_{DD} 上拉。

2.2.6 P50~P55（端口 5）

P50~P55 是 6 位 I/O 端口。这些引脚也可作为外部中断请求输入使用。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P50~P55 是 6 位 I/O 端口。通过端口模式寄存器 5（PM5），P50~P55 可按位设置为输入或输出端口。由上拉电阻选择寄存器 5（PU5）指定内置上拉电阻的使用。

(2) 控制模式

P50~P55 作为外部中断请求输入使用。

(a) INTP1, INTP2

这些是可以指定有效输入边沿的外部中断请求输入（上升沿，下降沿，或上升和下降双沿）引脚。

2.2.7 P60~P67（端口 6）

P60~P67 是 8 位 I/O 端口。这些引脚可作为串行接口数据 I/O、时钟 I/O 使用。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P60~P67 是 8 位 I/O 端口。通过端口模式寄存器 6（PM6），P60~P67 可按位设置为输入或输出端口。只有 P64~P67 可以由上拉电阻选择寄存器 6（PU6）指定内置上拉电阻的使用。

P60~P63 的输出是 N-ch 漏极开路输出（6 V 耐压）。

(2) 控制模式

P60~P67 可作为串行接口数据 I/O 和时钟 I/O 操作。

(a) SDA0

这是串行接口 IIC0 的串行数据 I/O 引脚。

(b) SCL0

这是串行接口 IIC0 的串行时钟 I/O 引脚。

2.2.8 P70~P77（端口 7）

P70~P77 是 8 位 I/O 端口。这些引脚也可作为按键中断输入、和外部中断请求输入操作。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P70~P77 是 8 位 I/O 端口。通过端口模式寄存器 7（PM7），P70~P77 可按位设置为输入或输出端口。由上拉电阻选择寄存器 7（PU7）指定内置上拉电阻的使用。

(2) 控制模式

P70~P77 的功能是按键中断输入、和外部中断请求输入。

(a) KR0~KR7

这些是按键中断输入引脚

(b) INTP8~INTP11

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

2.2.9 P90 (端口 9)

P90 是 1 位 I/O 端口。

可以用 1 位单元指定。

(1) 端口模式

P90 是 1 位 I/O 端口。通过端口模式寄存器 9 (PM9)，P90 可设置为输入或输出端口。由上拉电阻选择寄存器 9 (PU9) 指定内置上拉电阻的使用。

2.2.10 P110, P111 (端口 11)

P110, P111 是 2 位 I/O 端口。P110 和 P111 这些引脚作为 D/A 转换器模拟输出。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P110 和 P111 是 2 位 I/O 端口。通过端口模式寄存器 11 (PM11)，P110 和 P111 可按位设置为输入或输出端口。

(2) 控制模式

P110 和 P111 作为 D/A 转换器模拟输出引脚 (ANO0, ANO1)。当使用这些引脚作为模拟输入引脚时，参见 11.4.3 注意事项。

2.2.11 P120~P124 (端口 12)

P120 是 1 位 I/O 端口，P121~P124 是 4 位输入端口。这些引脚也可作为外部中断请求输入、外部低电压检测的电压输入，为主系统时钟连接电阻，为副系统时钟连接电阻，主系统时钟的外部时钟输入使用。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P120 是 1 位 I/O 端口。通过端口模式寄存器 12 (PM12)，P120 可设置为输入或输出端口。由上拉电阻选择寄存器 12 (PU12) 指定内置上拉电阻的使用。

P121~P124 是 4 位输入端口。

(2) 控制模式

P120~P124 作为外部中断请求输入、外部低电压检测的电压输入、为主系统时钟连接电阻、为副系统时钟连接电阻、和主系统时钟的外部时钟输入使用。

(a) INTP0

可定义有效沿 (上升沿、下降沿，或兼有上升沿和下降沿)，用于外部中断请求输入。

(b) EXLVI

这是外部低电压检测的电压输入引脚。

(c) X1, X2

这些引脚为主系统时钟连接电阻。

(d) EXCLK

为主系统时钟连接外部时钟输入。

(e) XT1, XT2

为副系统时钟连接电阻。

2.2.12 P130 (端口 13)

P130 是 1 位输出端口。

备注 当设备复位时, P130 输出低电平。因此, 在设备复位前 P130 全输出高电平, P130 的输出信号可用作 CPU 的伪复位信号 (参见 4.2.12 端口 13)

2.2.13 P140~P145 (端口 14)

P140~P145 是 6 位 I/O 端口。这些引脚可作为定时器 I/O、外部中断请求输入、时钟/蜂鸣器输出、串行接口数据 I/O 和时钟 I/O 使用。

使用端口输入模式寄存器 14 (PIM14), 通过普通缓冲器或 TTL 缓冲器可按位指定 P142 和 P143 引脚输入。

使用端口输出模式寄存器 14 (POM14), P142~P144 引脚输出可按位指定为普通 CMOS 输出或 N-ch 漏极开路输出 (V_{DD} 耐压)。

以下操作模式可以用 1 位单元指定。

(1) 端口模式

P140~P145 是 6 位 I/O 端口。通过端口模式寄存器 14 (PM14), P140~P145 可按位设置为输入或输出端口。由上拉电阻选择寄存器 14 (PU14) 指定内置上拉电阻的使用。

(2) 控制模式

P140~P145 是定时器 I/O、外部中断请求输入、时钟/蜂鸣器输出、串行接口数据 I/O 和时钟 I/O。

(a) INTP6, INTP7

可定义有效沿 (上升沿、下降沿, 或兼有上升沿和下降沿), 用于外部中断请求输入。

(b) PCLBUZ0, PCLBUZ1

这些是时钟输出/蜂鸣器输出引脚。

(c) TI07

此引脚用于输入 16 位定时器 07 的外部计数时钟/捕捉触发。

(d) TO07

这是 16 位定时器 07 的输出引脚。

(e) SI20

这是串行接口 CSI20 的串行数据输入引脚。

(f) SO20

这是串行接口 CSI20 的串行数据输出引脚。

(g) $\overline{\text{SCK20}}$

这是串行接口 CSI20 的串行时钟 I/O 引脚。

(h) TxD2

这是串行接口 UART2 的串行数据输出引脚。

(i) RxD2

这是串行接口 UART2 的串行数据输入引脚。

(j) SDA20

这是简易 I²C 串行接口的串行数据 I/O 引脚。

(k) SCL20

这是简易 I²C 串行接口的串行时钟 I/O 引脚。

2.2.14 AVREF0

用于 A/D 转换器参考电压输入和 P20~P27 及 A/D 转换器的正向电源供电。

当端口 2 的所有引脚用作模拟端口引脚时，确保 $2.3\text{ V} \leq \text{AVREF0} \leq \text{VDD}$ 。当端口 2 中一个或多个引脚用作数字端口引脚或不使用 A/D 转换器时，确保 AVREF0 的电压与 EVDD 或 VDD 相同。

2.2.15 AVREF1

用于 D/A 转换器参考电压输入和 P110、P111 及 D/A 转换器的正向电源供电。

当端口 11 的所有引脚用作模拟端口引脚时，确保 $1.8\text{ V} \leq \text{AVREF1} \leq \text{VDD}$ 。当端口 11 中一个或多个引脚用作数字端口引脚或不使用 D/A 转换器时，确保 AVREF1 的电压与 EVDD 或 VDD 相同。

2.2.16 AVss

作为 A/D 转换器，D/A 转换器，P20~P27、P110 和 P111 的地引脚。即使在不使用 A/D 转换器和 D/A 转换器时，该引脚电平也始终应与 EVss 和 Vss 相同。

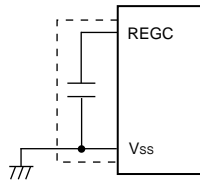
2.2.17 $\overline{\text{RESET}}$

这是低有效系统复位输入引脚。

当外部复位引脚不使用时，直接连接此引脚到 VDD，或经过一个电阻连接到 VDD。

2.2.18 REGC

<R> 用于内部操作的稳压器输出（2.5 V）稳定电容的连接。通过一个电容器（ $0.47\ \mu\text{F} \sim 1\ \mu\text{F}$ ）将此引脚连接到 V_{SS} 。但是，当进入 STOP 模式，因为内部高速振荡时钟和外部主系统时钟的操作，推荐 $0.47\ \mu\text{F}$ 使用性能特征优秀的电容器，因为它用于稳定内部电压。



注意事项 在上图虚线部分，使用的连接应该尽可能短。

2.2.19 V_{DD} , EV_{DD}

V_{DD} 是为 P121 ~ P124 与其他端口提供正向电源的引脚。

EV_{DD} 为端口的正向供电电源引脚，但 P20~P27, P110, P111, 和 P121~P124 除外。

2.2.20 V_{SS} , EV_{SS}

V_{SS} 是为 P121 ~ P124 与其他端口提供接地的引脚。

EV_{SS} 为端口的地引脚，但 P20~P27, P110, P111, 和 P121~P124 除外。

2.2.21 FLMD0

此引脚用于设置 flash 存储器编程模式。

按下列过程执行操作。

(a) 在普通操作模式下

普通操作期间，此引脚推荐开路。

复位释放前 FLMD0 引脚必须保持 V_{SS} 电平，但是不要外部下拉，因为通过复位该引脚已经被内部下拉。

通过背景事件控制寄存器（BECTL）的第 7 位（FLMDPUP）选择下拉（也就是，FLMDPUP = “0”，默认值）（参见 24.5 (1) 背景事件控制寄存器）。如果需要外部下拉，请使用 $200\ \text{k}\Omega$ 以内的电阻将其下拉。

自编程功能和使用编程器对 flash 存储器重写功能可以通过硬件来禁止。该引脚直接连接到 V_{SS} 即可。

(b) 在自编程模式下

使用自编程功能期间，此引脚推荐开路。如果需要外部下拉，请使用 $100\ \text{k}\Omega \sim 200\ \text{k}\Omega$ 的电阻将其下拉。

在自编程模式中，对该引脚的设置改为在自编程库中上拉。

(c) Flash 存储器编程器写入期间

当通过 flash 存储器编程器写入数据时，将此引脚直接连接到 flash 存储器编程器。这样会写入 V_{DD} 电平电压到 FLMD0 引脚。

因为复位已经对该引脚内部下拉，FLMD0 引脚无需外部下拉。要外部下拉，使用 $1\ \text{k}\Omega \sim 200\ \text{k}\Omega$ 的电阻。

2.3 引脚I/O 电路和未使用引脚的推荐连接方式

表 2-2 为引脚 I/O 电路类型和未使用引脚的建议连接方式。

表 2-2. 未使用引脚的连接 (1/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式	
P00/TI00	8-R	I/O	输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路。	
P01/TO00	5-AG			
P02/SO10/TxD1				
P03/SI10/RxD1/SDA10	5-AN			
P04/SCK10/SCL10				
P05/TI05/TO05	8-R			
P06/TI06/TO06				
P10/SCK00				
P11/SI00/RxD0				
P12/SO00/TxD0	5-AG			
P13/TxD3				
P14/RxD3	8-R			
P15/RTCDIV/RTCCCL	5-AG			
P16/TI01/TO01/INTP5	8-R			
P17/TI02/TO02				
P20/ANI0 ~ P27/ANI7 ^注	11-G			输入： 通过电阻分别连接到 AV _{REF0} 或 AV _{SS} 。 输出： 悬空开路
P30/RTC1HZ/INTP3	8-R			输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空 开路
P31/TI03/TO03/INTP4				
P40/TOOL0		<当允许片上调试时> 上拉此引脚（禁止下拉）。 <当禁止片上调试时> 输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路		
P41/TOOL1	5-AG	输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路		
P42/TI04/TO04	8-R			
P43/SCK01	5-AN			
P44/SI01				
P45/SO01	5-AG			
P46	8-R			
P47				

注 P20/ANI0~P27/ANI7 复位释放后被设置为数字输入端口。

表 2-2. 未使用引脚的连接 (2/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式	
P50/INTP1	8-R	I/O	输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路。	
P51/INTP2				
P52 ~ P55	5-AG			
P60/SCL0	13-R		输入： 连接到 EV _{SS} 。 输出： 设置端口输出锁存为 0，通过低电压输出使这些引脚悬空。	
P61/SDA0				
P62, P63	13-P			
P64 ~ P67	5-AG			
P70/KR0 ~ P73/KR3	8-R		输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路。	
P74/KR4/INTP8 ~ P77/KR7/INTP11				
P90				
P110/ANO0, P111/ANO1	12-G		输入： 通过电阻分别连接到 AV _{REF1} 或 AV _{SS} 。 输出： 悬空开路。	
P120/INTP0/EXLVI	8-R	输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路。		
<R> P121/X1 ^注	37-B	输入	通过电阻分别连接到 V _{DD} 或 V _{SS} 。	
				P122/X2/EXCLK ^注
				P123/XT1 ^注
				P124/XT2 ^注
P130	3-C	输出	悬空开路。	
P140/PCLBUZ0/INTP6	8-R	I/O	输入： 通过电阻分别连接到 EV _{DD} 或 EV _{SS} 。 输出： 悬空开路。	
P141/PCLBUZ1/INTP7				
P142/SCK20/SCL20	5-AN			
P143/SI20/RxD2/SDA20				
P144/SO20/TxD2	5-AG			
P145/TI07/TO07	8-R			
AV _{REF0}	-		-	<当 P20~P27 中一个以上设置为数字端口时> 使此引脚电压与 EV _{DD} 或 V _{DD} 相同。 <当 P20~P27 都设置为模拟端口时> 确保 2.3 V ≤ AV _{REF0} ≤ V _{DD} 。

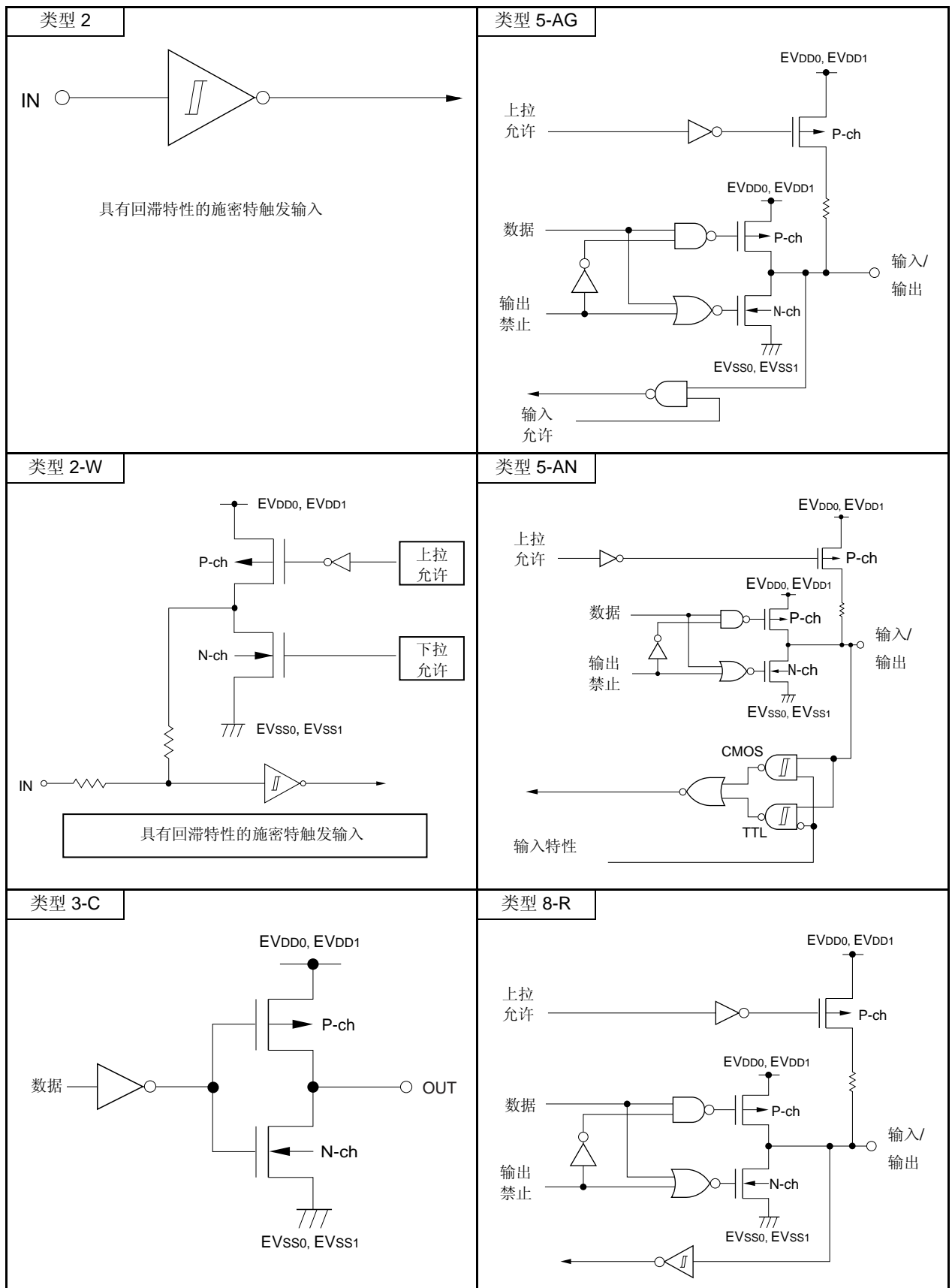
注 当这些引脚不使用时，使用上面推荐输入端口模式的连接（参见图 5-2 时钟操作模式控制寄存器（CMC）的格式）。

表 2-2. 未使用引脚的连接 (3/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
AV _{REF1}	-	-	<当 P110 或 P111 设置为数字端口时> 保证此引脚电压与 EV _{DD0} 或 V _{DD} 。 <当 P110 和 P111 均被设置为模拟端口时> 确保 $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ 。
AV _{SS}	-	-	保证此引脚电压与 EV _{SS} 或 V _{SS} 相同。
FLMD0	2-W	-	悬空或经过一个大于 100 kΩ 的电阻连接到 V _{SS} 。
RESET	2	输入	直接连接到 V _{DD} ，或经过一个电阻连接到 V _{DD} 。
REGC	-	-	通过一个电容 (0.47~1 μF) 连接到 V _{SS} 。

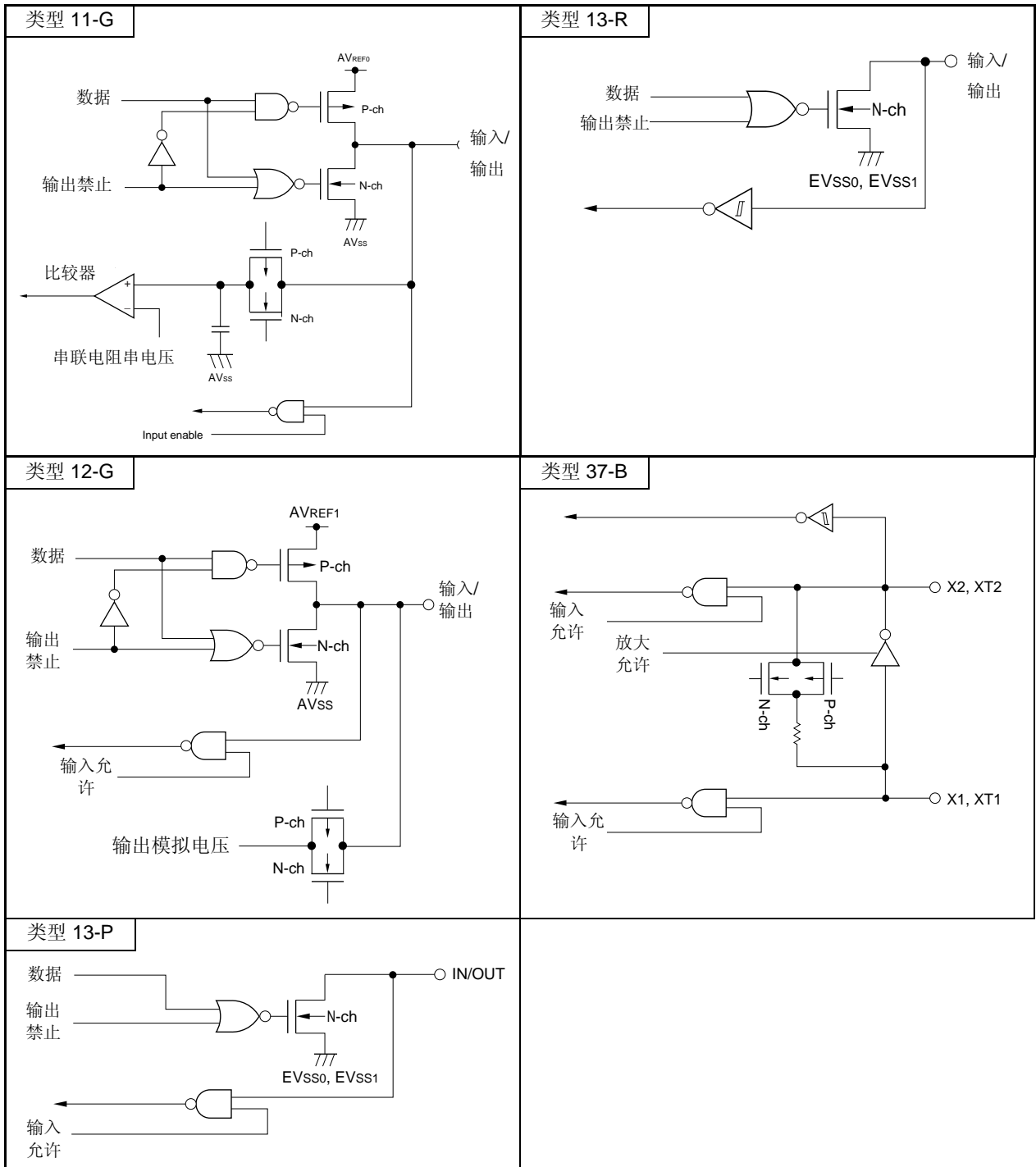
<R>

图 2-1. 引脚 I/O 电路列表 (1/2)



<R>

图 2-1. 引脚 I/O 电路列表 (2/2)

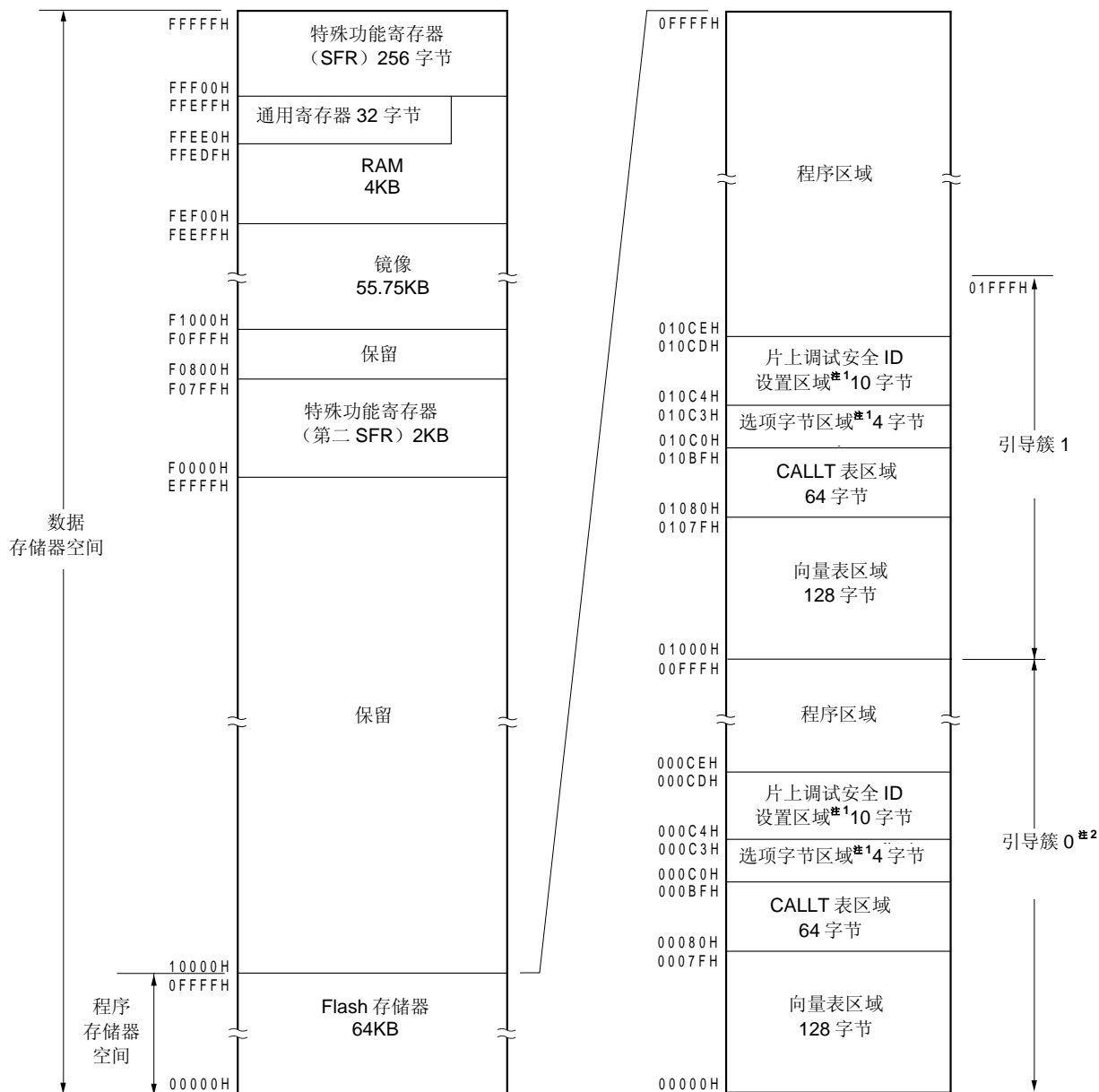


第三章 CPU 结构

3.1 存储器空间

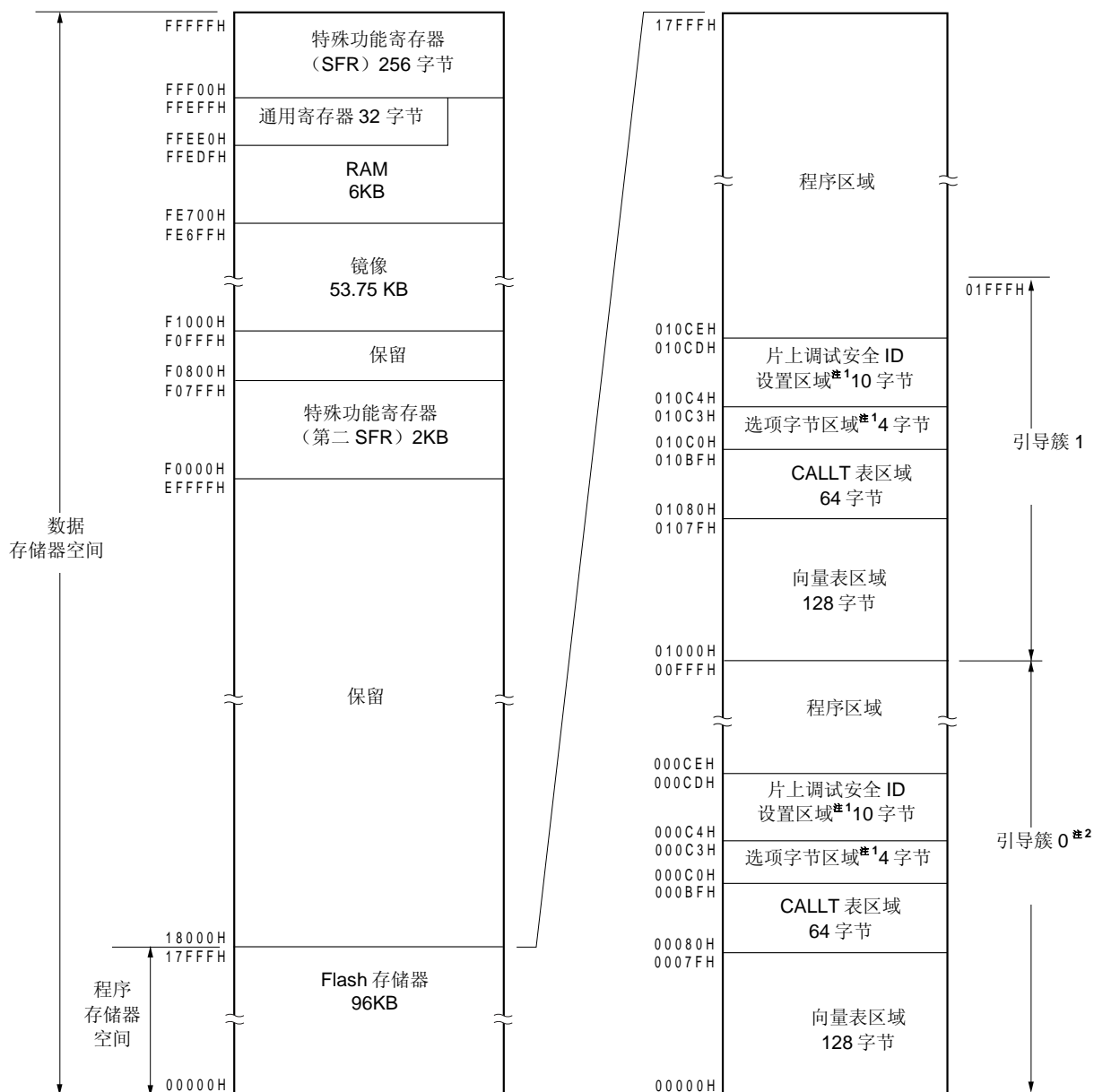
78K0R/KF3 产品可以访问大小为 1MB 的存储器空间。图 3-1 到 3-5 显示了存储空间映射图。

图 3-1. 存储器映射图 (μ PD78F1152)



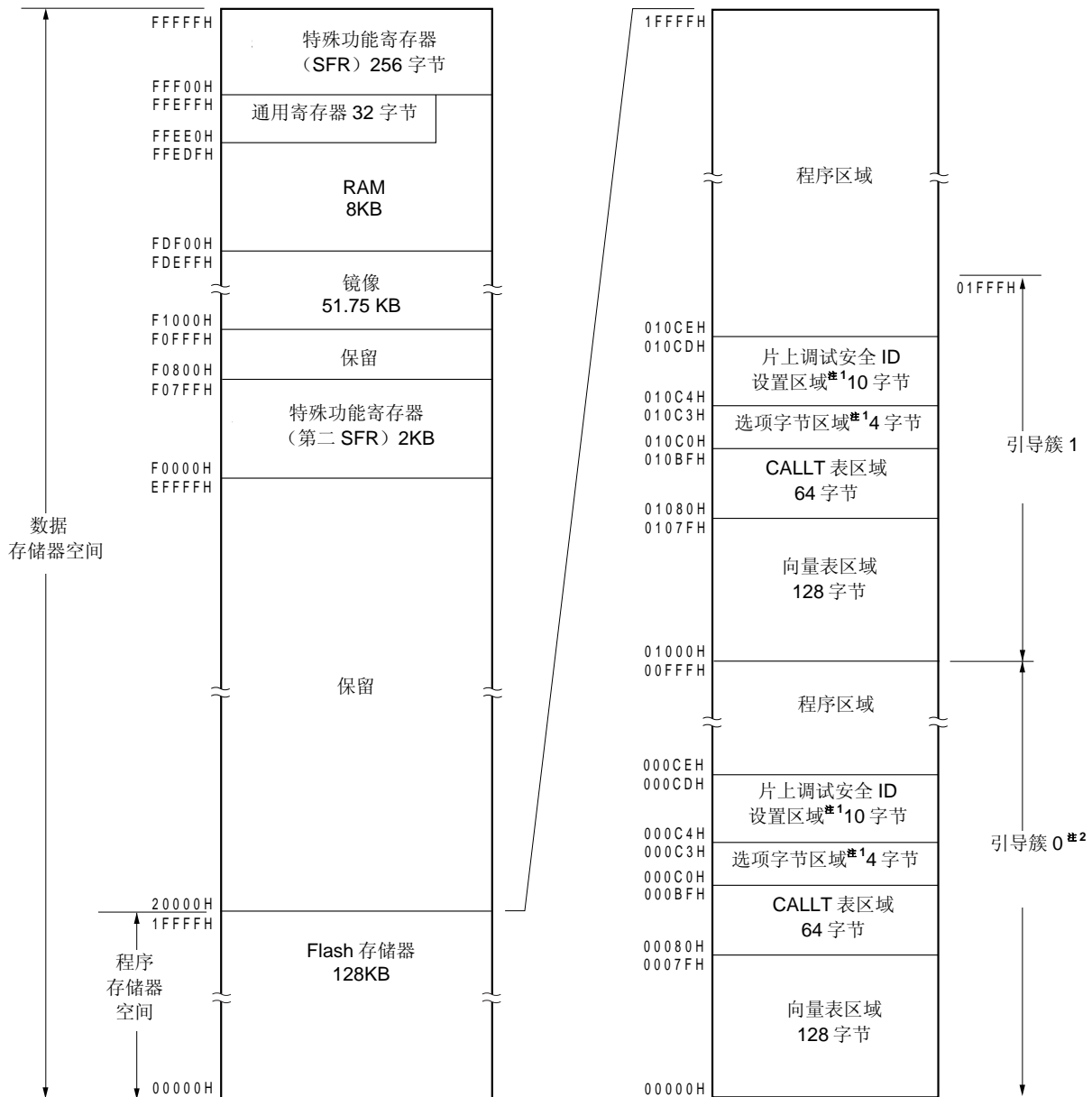
- 注 1. 当不使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H，片上调试安全 ID 为 000C4H ~ 000CDH。
 当使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H 和 010C0H ~ 010C3H，片上调试安全 ID 为 000C4H ~ 000CDH 和 010C4H ~ 010CDH。
2. 根据安全设置可以禁止写入引导簇 0（参见 24.7 安全设置）。

图 3-2. 存储器映射图 (μPD78F1153)



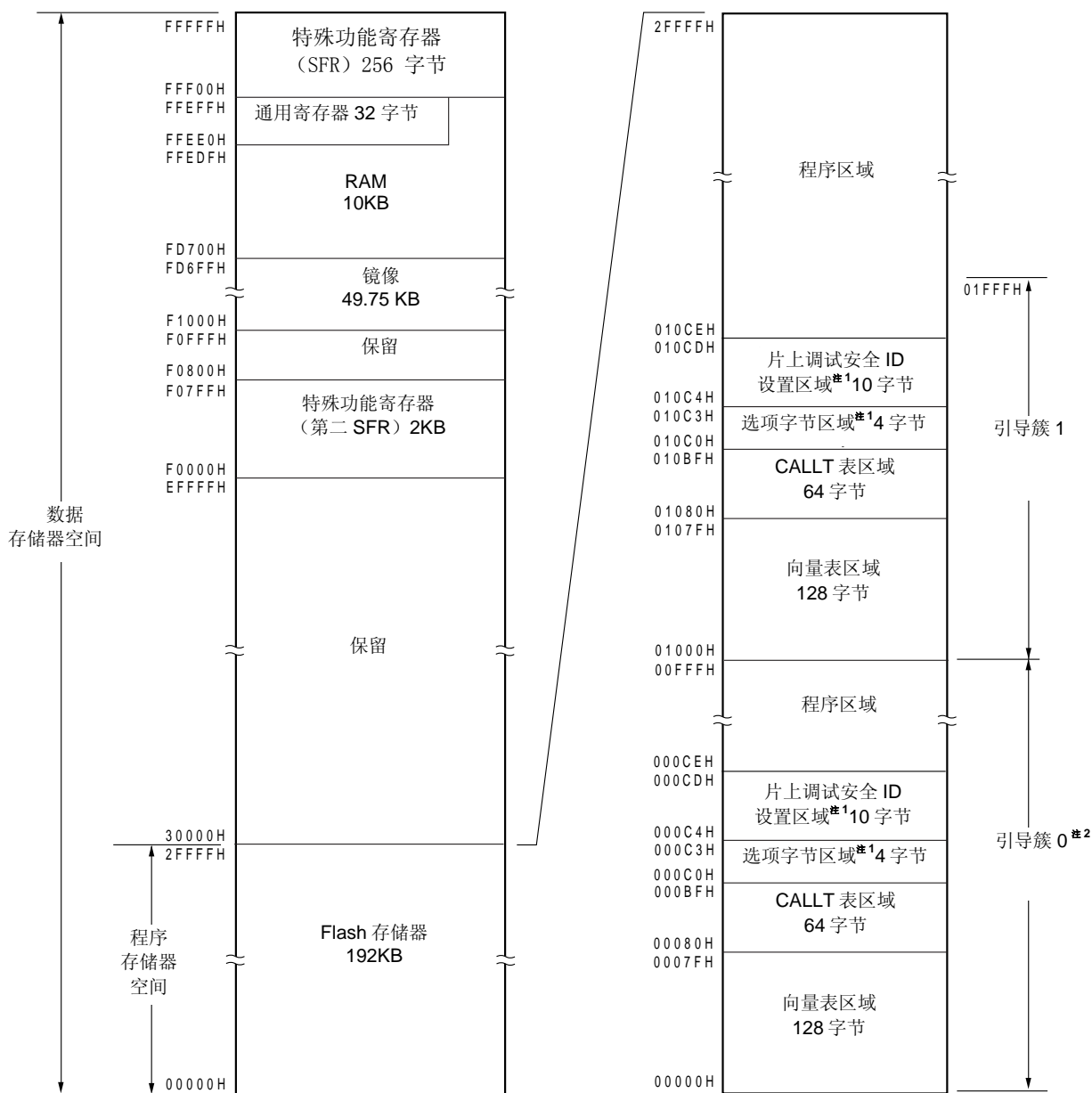
- 注 1. 当不使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H，片上调试安全 ID 为 000C4H ~ 000CDH。
 当使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H 和 010C0H ~ 010C3H，片上调试安全 ID 为 000C4H ~ 000CDH 和 010C4H ~ 010CDH。
2. 根据安全设置可以禁止写入引导簇 0（参见 24.7 安全设置）。

图 3-3. 存储器映射图 (μPD78F1154)



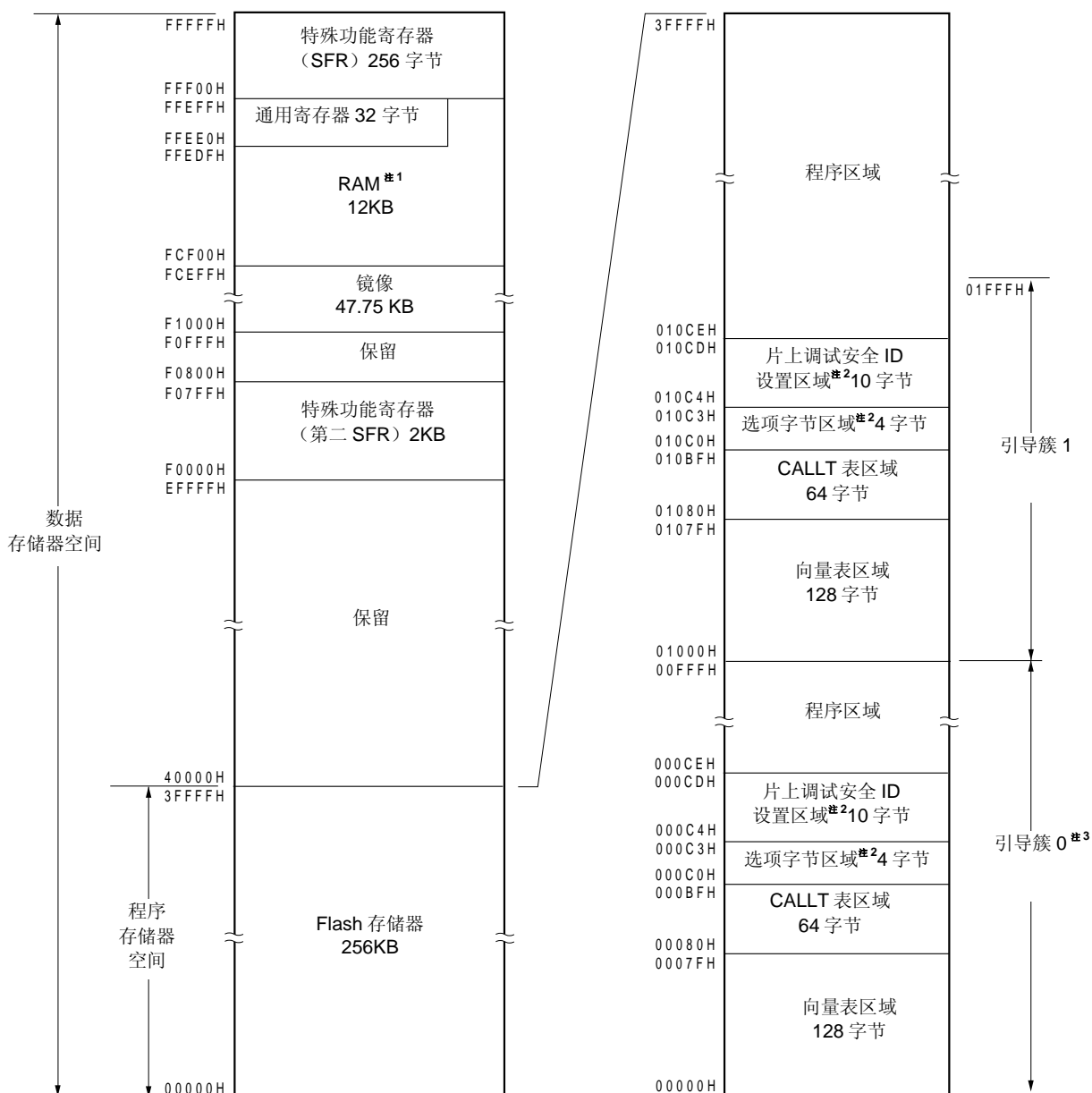
- 注 1. 当不使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H，片上调试安全 ID 为 000C4H ~ 000CDH。
 当使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H 和 010C0H ~ 010C3H，片上调试安全 ID 为 000C4H ~ 000CDH 和 010C4H ~ 010CDH。
2. 根据安全设置可以禁止写入引导簇 0 (参见 24.7 安全设置)。

图 3-4. 存储器映射图 (μPD78F1155)



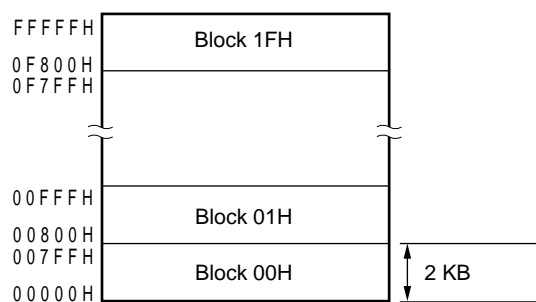
- 注 1. 当不使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H，片上调试安全 ID 为 000C4H ~ 000CDH。
 当使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H 和 010C0H ~ 010C3H，片上调试安全 ID 为 000C4H ~ 000CDH 和 010C4H ~ 010CDH。
2. 根据安全设置可以禁止写入引导簇 0（参见 24.7 安全设置）。

图 3-5. 存储器映射图 (μPD78F1156)



- 注
1. 使用自编程功能时，区域 F8700H~F8EFFH 禁止使用。因为此区域用于自编程库。
 2. 当不使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H，片上调试安全 ID 为 000C4H ~ 000CDH。
当使用引导交换功能的时候：设置选项字节区域：000C0H ~ 000C3H 和 010C0H ~ 010C3H，片上调试安全 ID 为 000C4H ~ 000CDH 和 010C4H ~ 010CDH。
 3. 根据安全设置可以禁止写入引导簇 0（参见 24.7 安全设置）。

备注 Flash 存储器分为块 (block) (1 块 (block) = 2 KB)。如需了解地址值和块编号, 参见表 3-1 **Flash 存储器的地址值和块 (block) 编号之间的关系。**



Flash 存储器的地址值和块 (block) 编号之间的关系如下所示。

表 3-1. Flash 存储器的地址值和块 (block) 编号之间的关系 (1/2)

地址值	块 (block) 编号	地址值	块 (block) 编号	地址值	块 (block) 编号	地址值	块 (block) 编号
00000H~007FFH	00H	10000H~107FFH	20H	20000H~207FFH	40H	30000H~307FFH	60H
00800H~00FFFH	01H	10800H~10FFFH	21H	20800H~20FFFH	41H	30800H~30FFFH	61H
01000H~017FFH	02H	11000H~117FFH	22H	21000H~217FFH	42H	31000H~317FFH	62H
01800H~01FFFH	03H	11800H~11FFFH	23H	21800H~21FFFH	43H	31800H~31FFFH	63H
02000H~027FFH	04H	12000H~127FFH	24H	22000H~227FFH	44H	32000H~327FFH	64H
02800H~02FFFH	05H	12800H~12FFFH	25H	22800H~22FFFH	45H	32800H~32FFFH	65H
03000H~037FFH	06H	13000H~137FFH	26H	23000H~237FFH	46H	33000H~337FFH	66H
03800H~03FFFH	07H	13800H~13FFFH	27H	23800H~23FFFH	47H	33800H~33FFFH	67H
04000H~047FFH	08H	14000H~147FFH	28H	24000H~247FFH	48H	34000H~347FFH	68H
04800H~04FFFH	09H	14800H~14FFFH	29H	24800H~24FFFH	49H	34800H~34FFFH	69H
05000H~057FFH	0AH	15000H~157FFH	2AH	25000H~257FFH	4AH	35000H~357FFH	6AH
05800H~05FFFH	0BH	15800H~15FFFH	2BH	25800H~25FFFH	4BH	35800H~35FFFH	6BH
06000H~067FFH	0CH	16000H~167FFH	2CH	26000H~267FFH	4CH	36000H~367FFH	6CH
06800H~06FFFH	0DH	16800H~16FFFH	2DH	26800H~26FFFH	4DH	36800H~36FFFH	6DH
07000H~077FFH	0EH	17000H~177FFH	2EH	27000H~277FFH	4EH	37000H~377FFH	6EH
07800H~07FFFH	0FH	17800H~17FFFH	2FH	27800H~27FFFH	4FH	37800H~37FFFH	6FH
08000H~087FFH	10H	18000H~187FFH	30H	28000H~287FFH	50H	38000H~387FFH	70H
08800H~08FFFH	11H	18800H~18FFFH	31H	28800H~28FFFH	51H	38800H~38FFFH	71H
09000H~097FFH	12H	19000H~197FFH	32H	29000H~297FFH	52H	39000H~397FFH	72H
09800H~09FFFH	13H	19800H~19FFFH	33H	29800H~29FFFH	53H	39800H~39FFFH	73H
0A000H~0A7FFH	14H	1A000H~1A7FFH	34H	2A000H~2A7FFH	54H	3A000H~3A7FFH	74H
0A800H~0AFFFH	15H	1A800H~1AFFFH	35H	2A800H~2AFFFH	55H	3A800H~3AFFFH	75H
0B000H~0B7FFH	16H	1B000H~1B7FFH	36H	2B000H~2B7FFH	56H	3B000H~3B7FFH	76H
0B800H~0BFFFH	17H	1B800H~1BFFFH	37H	2B800H~2BFFFH	57H	3B800H~3BFFFH	77H
0C000H~0C7FFH	18H	1C000H~1C7FFH	38H	2C000H~2C7FFH	58H	3C000H~3C7FFH	78H
0C800H~0CFFFH	19H	1C800H~1CFFFH	39H	2C800H~2CFFFH	59H	3C800H~3CFFFH	79H
0D000H~0D7FFH	1AH	1D000H~1D7FFH	3AH	2D000H~2D7FFH	5AH	3D000H~3D7FFH	7AH
0D800H~0DFFFH	1BH	1D800H~1DFFFH	3BH	2D800H~2DFFFH	5BH	3D800H~3DFFFH	7BH
0E000H~0E7FFH	1CH	1E000H~1E7FFH	3CH	2E000H~2E7FFH	5CH	3E000H~3E7FFH	7CH
0E800H~0EFFFH	1DH	1E800H~1EFFFH	3DH	2E800H~2EFFFH	5DH	3E800H~3EFFFH	7DH
0F000H~0F7FFH	1EH	1F000H~1F7FFH	3EH	2F000H~2F7FFH	5EH	3F000H~3F7FFH	7EH
0F800H~0FFFFH	1FH	1F800H~1FFFFH	3FH	2F800H~2FFFFH	5FH	3F800H~3FFFFH	7FH

备注 μPD78F1152: 块 (block) 编号 00H~1FH
 μPD78F1153: 块 (block) 编号 00H~2FH
 μPD78F1154: 块 (block) 编号 00H~3FH
 μPD78F1155: 块 (block) 编号 00H~5FH
 μPD78F1156: 块 (block) 编号 00H~7FH

3.1.1 内部程序存储器空间

内部程序存储空间用于存储程序和表数据，一般通过程序计数器（PC）来寻址。

78K0R/KF3 产品内部 ROM（Flash 存储器）的情况如下表所示。

表 3-2. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μ PD78F1152	Flash 存储器	65536 \times 8 位 (00000H~0FFFFH)
μ PD78F1153		98303 \times 8 位 (00000H~17FFFH)
μ PD78F1154		131071 \times 8 位 (00000H~1FFFFH)
μ PD78F1155		196607 \times 8 位 (00000H~2FFFFH)
μ PD78F1156		262143 \times 8 位 (00000H~3FFFFH)

内部程序存储空间主要分为以下几个区域。

(1) 向量表区域

00000H~0007FH 共 128 字节作为向量表区域。在向量表中存放的内容是根据复位信号或每个中断请求的产生进行跳转的程序的起始地址。

在 16 位地址中，低 8 位存储在偶地址，高 8 位存储在奇地址。

表 3-3. 向量表

向量表地址	中断源	向量表地址	中断源
00000H	RESET 输入, POC, LVI, WDT, TRAP	0002CH	INTTM00
		0002EH	INTTM01
00004H	INTWDTI	00030H	INTTM02
00006H	INTLVI	00032H	INTTM03
00008H	INTP0	00034H	INTAD
0000AH	INTP1	00036H	INTRTC
0000CH	INTP2	00038H	INTRTCI
0000EH	INTP3	0003AH	INTKR
00010H	INTP4	0003CH	INTST2/INTCSI20/INTIIC20
00012H	INTP5	0003EH	INTSR2
00014H	INTST3	00040H	INTSRE2
00016H	INTSR3	00042H	INTTM04
00018H	INTSRE3	00044H	INTTM05
0001AH	INTDMA0	00046H	INTTM06
0001CH	INTDMA1	00048H	INTTM07
0001EH	INTST0/INTCSI00	0004AH	INTP6
00020H	INTSR0/INTCSI01	0004CH	INTP7
00022H	INTSRE0	0004EH	INTP8
00024H	INTST1/INTCSI10/INTIIC10	00050H	INTP9
00026H	INTSR1	00052H	INTP10
00028H	INTSRE1	00054H	INTP11
0002AH	INTIIC0	0007EH	BRK

(2) CALLT 指令表区域

00080H~000BFH 共 64 字节的区域, 可存放 2 字节调用指令 (CALLT) 的子程序入口地址。设置子程序入口地址的值在 00000H~0FFFFH 范围内 (因为地址码为 2 字节)。

要使用引导交换功能, 设置 CALLT 指令表也在 01080H~010BFH。

(3) 选项字节区域

000C0H~000C3H 的 4 字节的区域可以用作选项字节区域。当使用引导交换功能时, 在 010C0H~010C3H 设置选项字节。详细信息参见第二十三章 选项字节。

(4) 片上调试安全 ID 设置区域

000C4H~000CDH 和 010C4H~010CDH 各 10 字节区域可用作片上调试安全 ID 设置区域。不使用引导交换功能时, 在 000C4H~000CDH 区域设置片上调试安全 ID; 而当使用引导交换功能时, 在 000C4H~000CDH 和 010C4H~010CDH 区域设置。详细情况请参见第二十五章 片上调试功能。

3.1.2 镜像区域

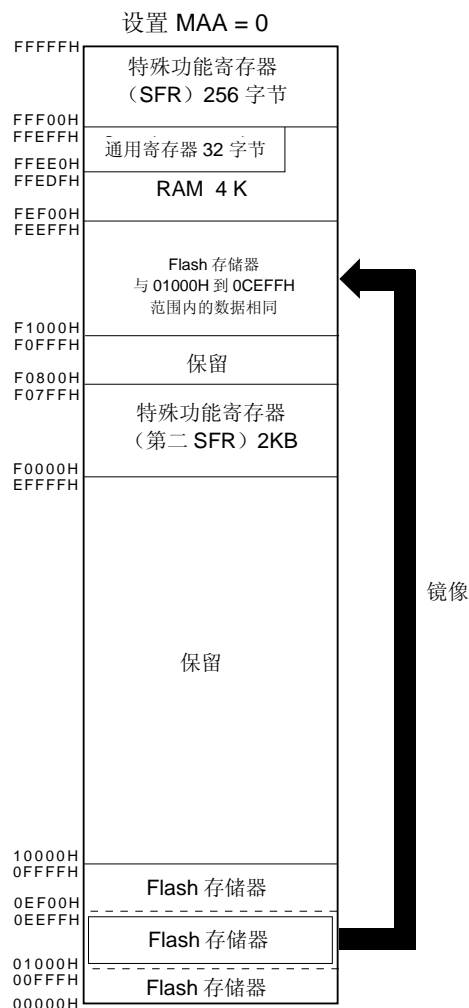
μ PD78F1152 将 00000H~0FFFFH 的数据 flash 区域镜像到 F0000H~FFFFFFH。 μ PD78F1153, 78F1154, 78F1155 和 78F1156 将 00000H~0FFFFH 或 10000H~1FFFFH 的数据 flash 区域镜像到 F0000H~FFFFFFH（数据 flash 区域的镜像通过处理器模式控制寄存器（PMC）来设置）。

通过从 F0000H~FFFFFFH 读数据，数据 flash 的内容可以用更短的代码读取。但是数据 flash 区域不能镜像到 SFR，扩展 SFR，RAM，和禁止使用的区域。

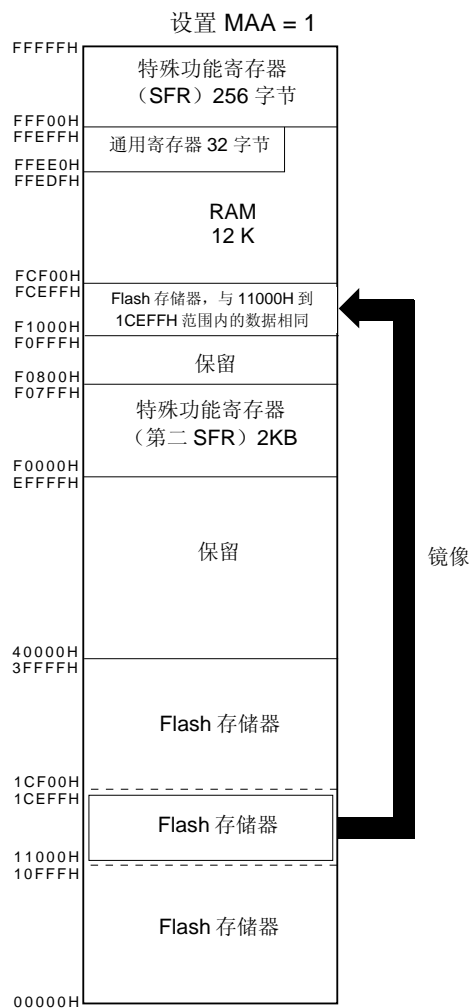
镜像区域只能读取，并且不能从此区域中获取指令。

如下为示例。

示例 1 μ PD78F1152
(Flash 存储器: 64 KB, RAM: 4 KB)



示例 2 μ PD78F1156
(Flash 存储器: 256 KB, RAM: 12 KB)



备注 MAA: 处理器模式控制寄存器（PMC）的第 0 位。

PMC 寄存器描述如下。

• 处理器模式控制寄存器 (PMC)

此寄存器选择 Flash 存储器内将要被镜像到区域 F0000H~FFFFFFH 的空间。

PMC 只能由 1 位 或 8 位存储器操作指令设置。

复位信号的生成会将此寄存器设置为 00H。

图 3-6.处理器模式控制寄存器 (PMC) 的配置格式

地址: FFFFEH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	选择Flash存储器内将要被镜像到区域F0000H~FFFFFFH的空间
0	00000H~0FFFFH镜像到 F0000H~FFFFFFH
1	10000H~1FFFFH 镜像到 F0000H~FFFFFFH

- 注意事项
1. 在初始化设置期间, 只能设置一次 PMC, 并要在操作 DMA 控制器之前。除了在初始化设置期间外, 禁止重写 PMC。
 2. 设置 PMC 后, 等待至少一个指令周期才能访问镜像区域。
 3. 当使用 μ PD78F1152 时, 确保江该寄存器的第 0 位 (MAA) 设置为 0。

3.1.3 内部数据存储器空间

78K0R/KF3 产品包括如下 RAM。

表 3-4. 内部 RAM 容量

产品型号	内部 RAM
μ PD78F1152	4096 \times 8 位 (FDF00H~FFEFFFH)
μ PD78F1153	6144 \times 8 位 (FD700H~FFEFFFH)
μ PD78F1154	8192 \times 8 位 (FCF00H~FFEFFFH)
μ PD78F1155	10240 \times 8 位 (F9F00H~FFEFFFH)
μ PD78F1156	12288 \times 8 位 (F8700H~FFEFFFH)

32 字节的区域 FFEE0H~FFEFFFH 分配了四组通用寄存器组, 每个寄存器组包括 8 个 8 位寄存器。

该区域可用作写入和执行指令的程序区域。但是, 在通用寄存器中禁止执行指令。

内部高速 RAM 可用作堆栈存储器。

- 注意事项 当使用自编程功能时, 区域 FFE20H~FFEDFH 和 FCF00H~FD6FFFH (μ PD78F1156) 不能用作堆栈存储器。

3.1.4 特殊功能寄存器（SFR）区域

片上外围硬件特殊功能寄存器（SFRs）存放在区域 FFF00H～FFFFFH（参见 3.2.4 特殊功能寄存器（SFRs）的表 3-5 特殊功能寄存器列表）内。

注意事项 不要访问那些未分配特殊功能寄存器的地址区域。

3.1.5 扩展特殊功能寄存器（第二 SFR：第二特殊功能寄存器）区域

片上外围硬件特殊功能寄存器（第二 SFRs）被分配在 F0000H～F07FFH 区域（参见 3.2.5 扩展特殊功能寄存器（第二 SFR：第二特殊功能寄存器）中的表 3-6）内。

SFR 区域（从 FFF00H 到 FFFFFH）之外的 SFR 分配到此扩展 SFR 空间。但是，扩展 SFR 区域访问指令长度比 SFR 区域多 1 个字节。

注意事项 不要访问那些未分配特殊功能寄存器的地址区域。

3.1.6 数据存储寻址

寻址是定位地址的方式，需要定位的地址包括下一条执行指令的地址，或者与指令执行相关的存储器地址或寄存器地址。

基于可操作性和其他考虑，在 78K0R/KF3 中提供了几种用于与指令执行相关的存储器寻址方式。因为有专门的数据存储区域，故可使用一些特殊的寻址方式，具有特殊功能寄存器（SFR）和通用寄存器的功能。图 3-7 到 3-11 显示了数据存储空间与寻址方式的对应关系。

图 3-7. 数据存储与寻址方式的对应关系 (μPD78F1152)

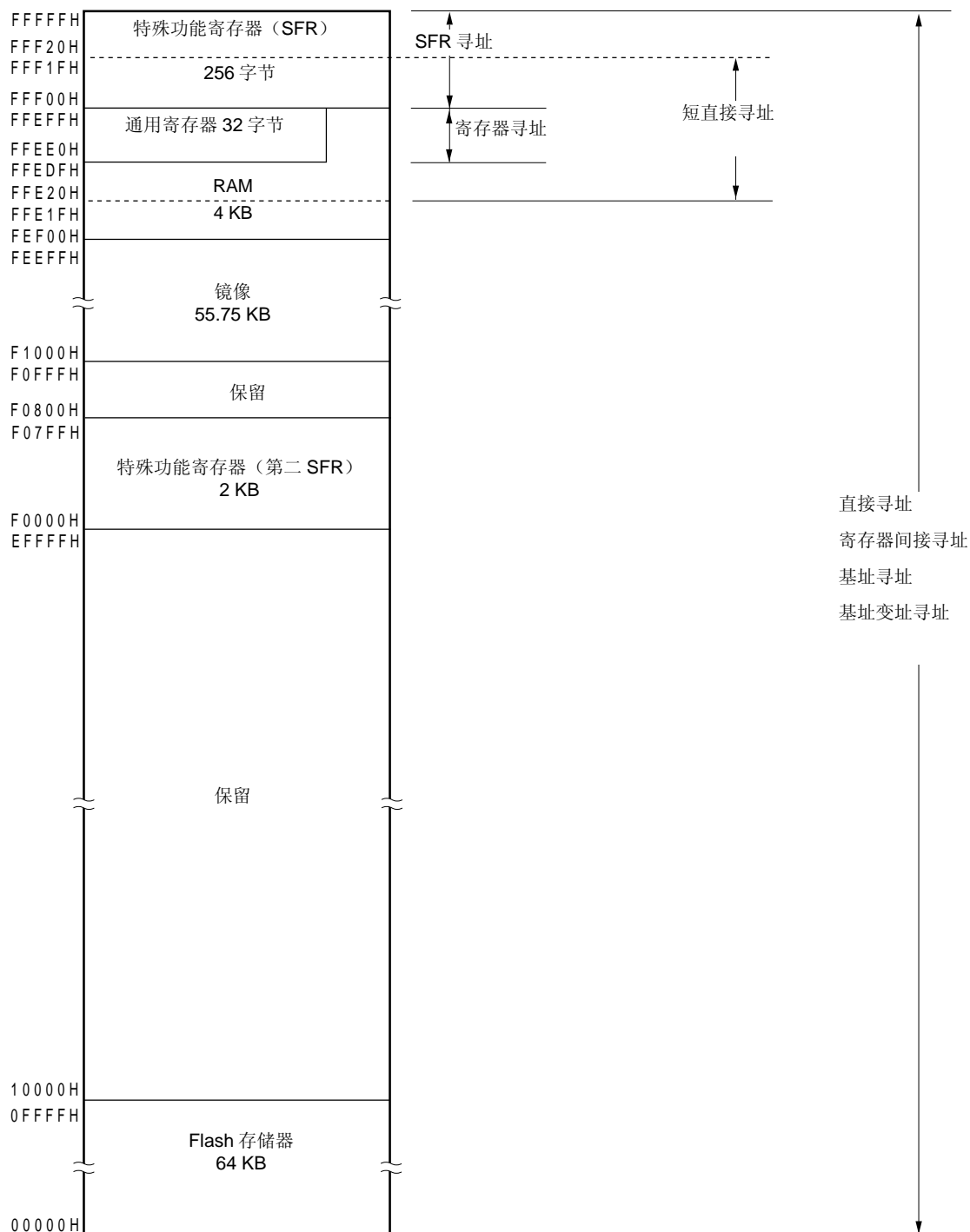


图 3-8. 数据存储与寻址方式的对应关系 (μPD78F1153)

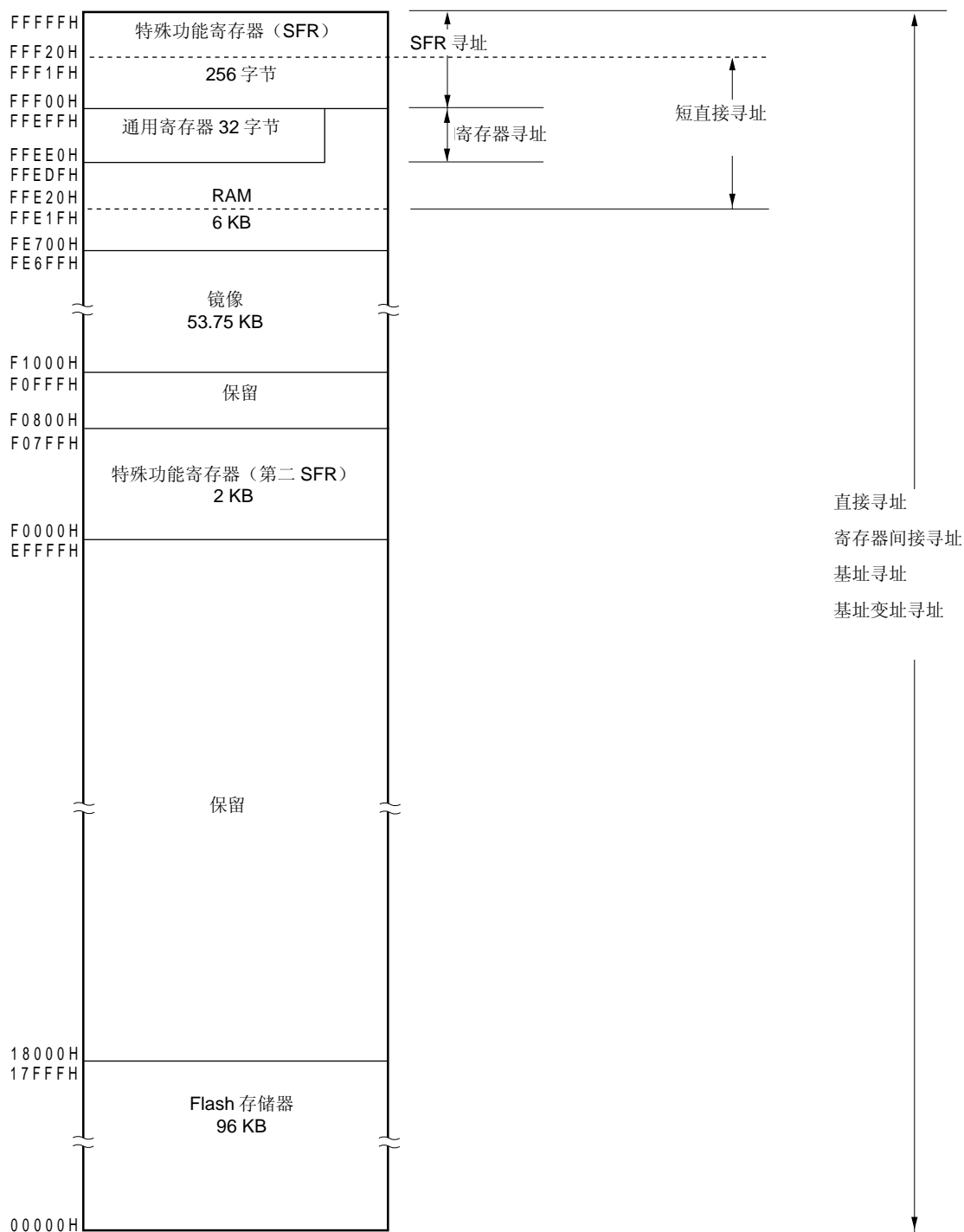


图 3-9. 数据存储与寻址方式的对应关系 (μPD78F1154)

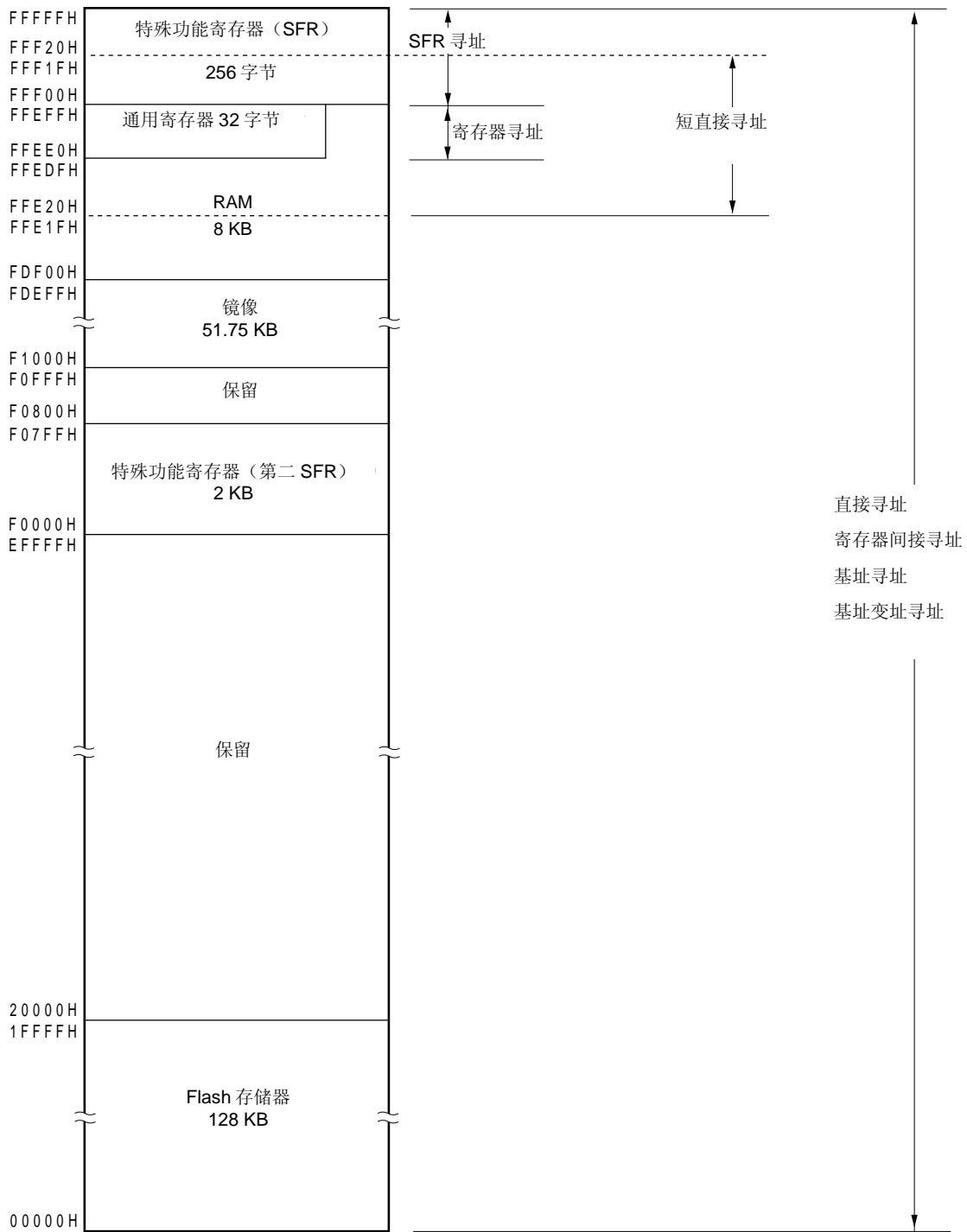


图 3-10. 数据存储器与寻址方式的对应关系 (μPD78F1155)

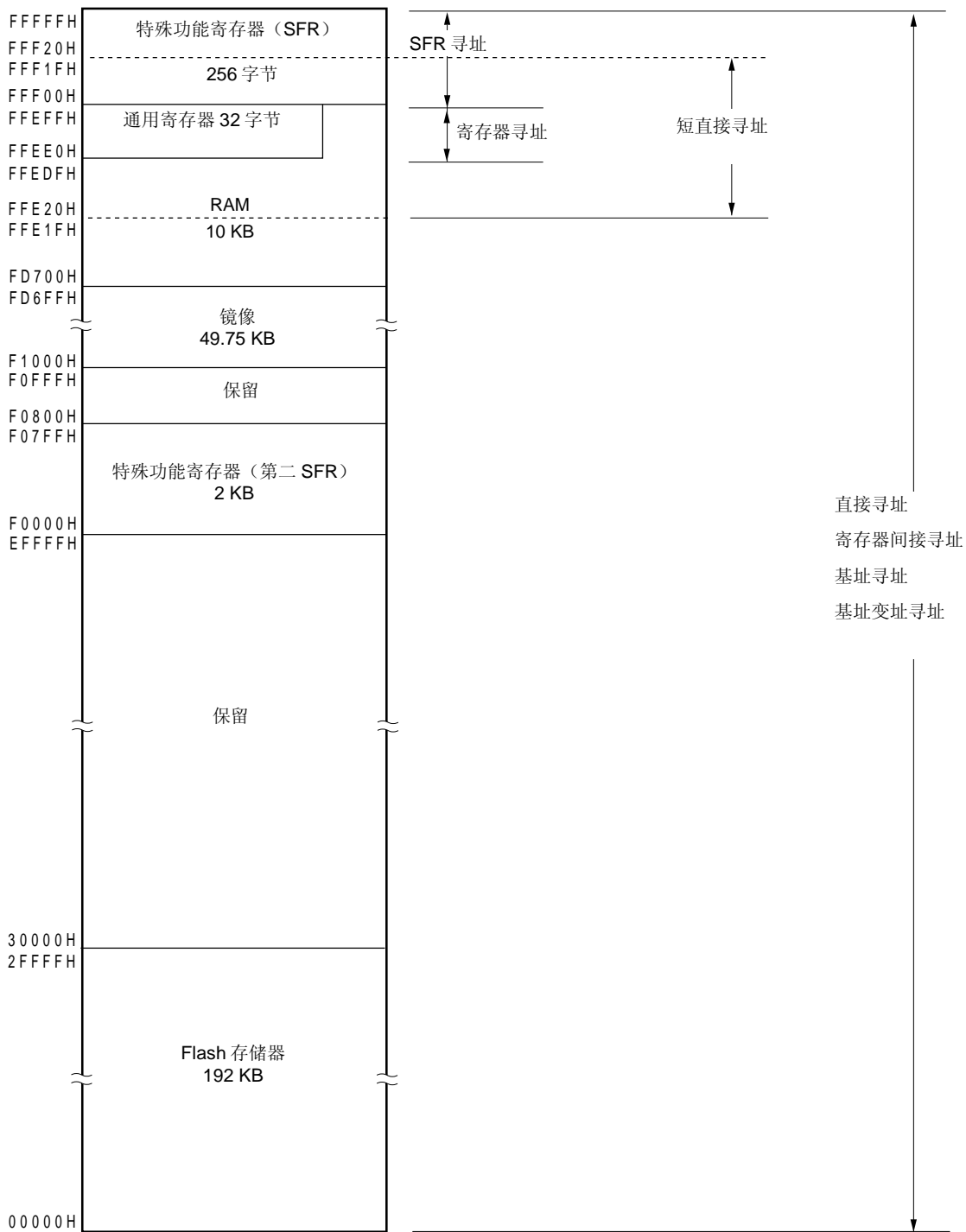
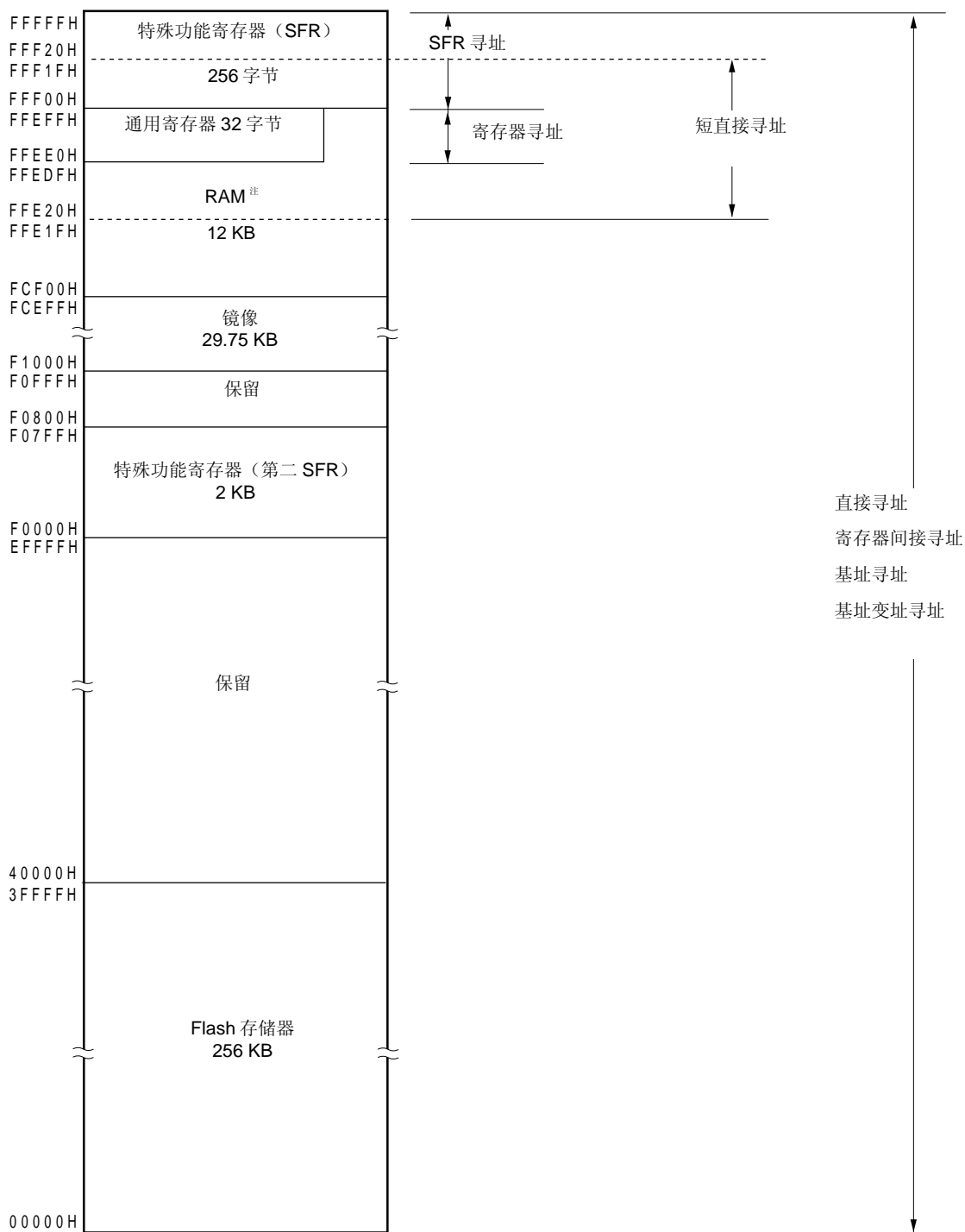


图 3-11. 数据存储器与寻址方式的对应关系 (μPD78F1156)



注 当使用自编程功能时，区域 FCF00H~FD6FFH 禁止使用。因为此区域用于自编程库。

3.2 处理器寄存器

78K0R/KF3 产品包含以下几种处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序执行的顺序、状态和堆栈空间。程序计数器（PC）、程序状态字（PSW）和堆栈指针寄存器（SP）都属于控制寄存器。

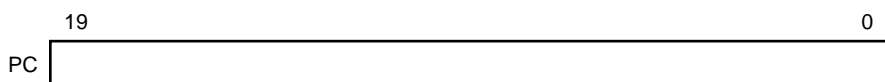
(1) 程序计数器（PC）

程序计数器是一个 20 位寄存器，用于存放下一条即将要执行的指令的地址。

在正常情况下，根据获取的指令字节数，程序计数器（PC）的值会自动累加。当执行分支指令时，则设置立即数和寄存器内容。

复位信号的产生将复位向量表中地址为 0000H 和 0001H 中的值赋给程序计数器。

图 3-12. 程序计数器的格式



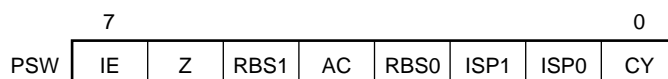
(2) 程序状态字（PSW）

程序状态字（PSW）是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行设置或复位。

根据中断请求的产生或执行 PUSH PSW 指令，程序状态字的内容自动入栈；通过执行 RETB, RETI 和 POP PSW 指令，程序状态字的值自动恢复。

复位信号的产生将程序状态字的内容设置为 06H。

图 3-13. 程序状态字的格式



(a) 中断允许标志（IE）

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示不允许中断（DI），即禁止所有可屏蔽中断请求。

当 IE 为 1 时，表示允许中断（EI），通过优先服务标志（ISP1, ISP0）、用于各种中断源的中断屏蔽标志以及优先级规定标志来完成响应中断请求的控制。

当执行 DI 指令或中断请求得到响应时，该标志复位（0）；当执行 EI 指令时，该标志设置为 1。

(b) 零标志（Z）

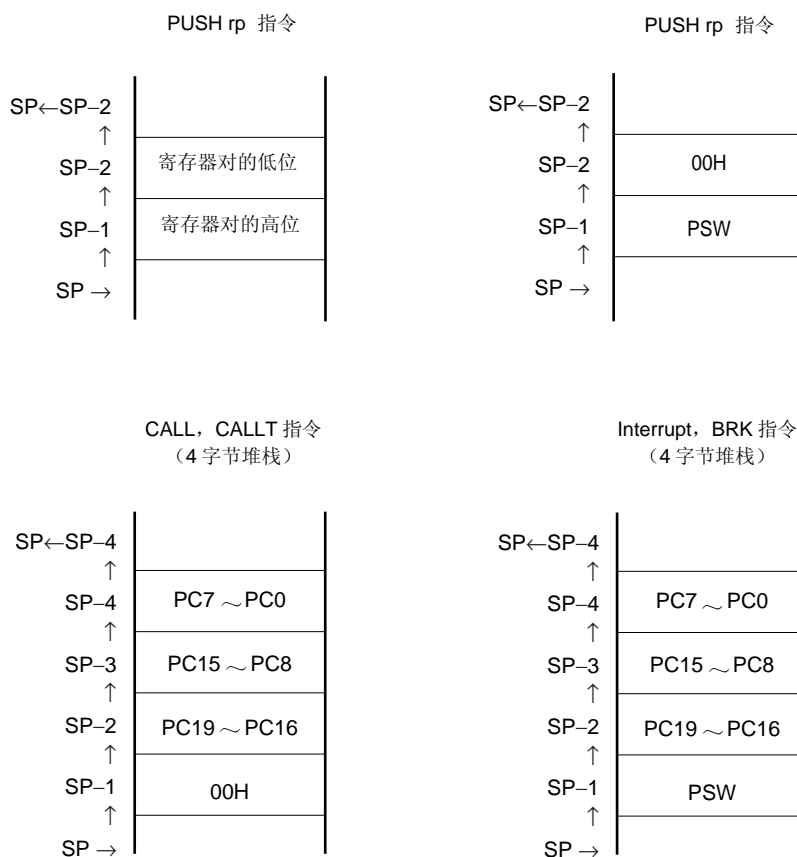
当操作结果为 0 时，该标志置位（1），其他所有情况下都重置（0）。

(c) 寄存器组选择标志（RBS0, RBS1）

寄存器组选择标志有两位，用于选择四组寄存器中的其中一组。

标志位中存储的信息用来指明执行 SEL RBN 指令时所选择的寄存器组。

图 3-15. 将数据存入堆栈



3.2.2 通用寄存器

通用寄存器映射到数据存储器的特定地址空间 FFEE0H~ FFEFFH。通用寄存器共有四组，每一组由 8 个 8 位寄存器 (X, A, C, B, E, D, L 和 H) 组成。

此外每个寄存器可作为一个 8 位寄存器使用，两个成对的 8 位寄存器可作为一个 16 位寄存器 (AX, BC, DE 和 HL) 使用。

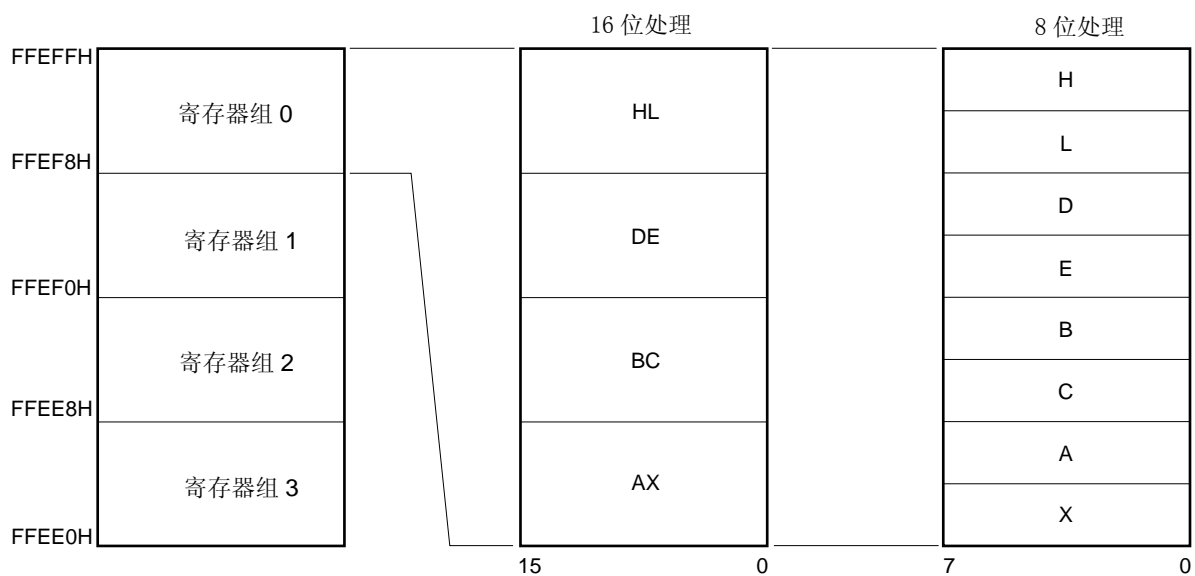
描述通用寄存器时，可以使用功能名称 (X, A, C, B, E, D, L, H 及 AX, BC, DE, HL) 或绝对名称 (R0 ~ R7, RP0 ~ RP3)。

用于指令执行的寄存器组由 CPU 控制指令 (SEL Rbn) 来设置。由于 4 个寄存器组的结构，通过一个用于正常处理的寄存器和另一个用于中断处理的寄存器之间的切换，可以创建一个高效率的程序。

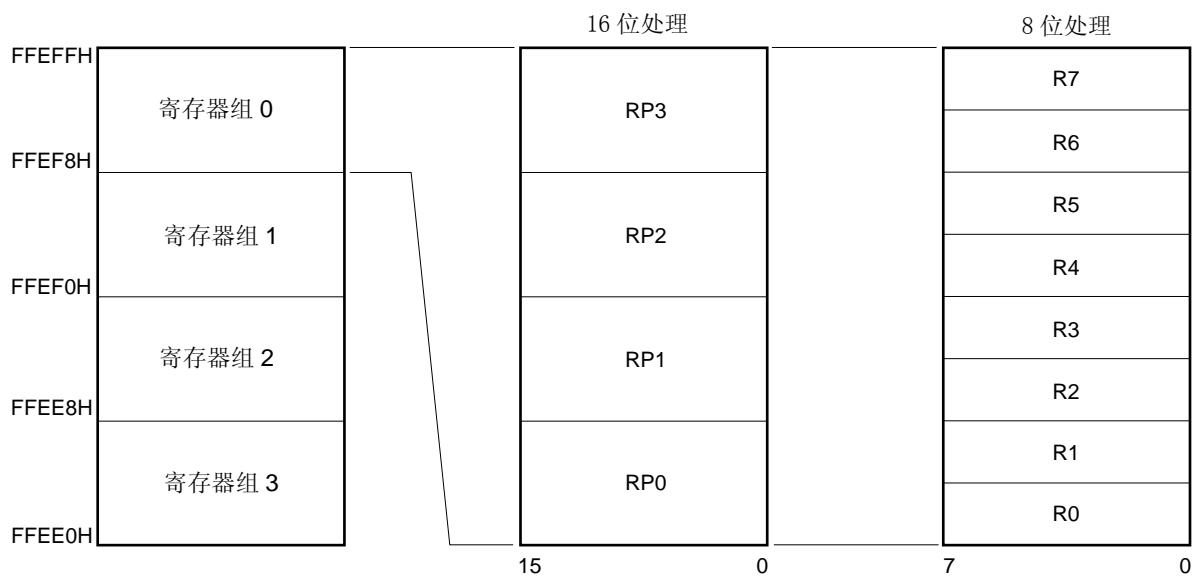
注意事项 禁止使用通用寄存器 (FFEE0H~FFEFFH) 空间提取指令或作为堆栈区域。

图 3-16. 通用寄存器的配置

(a) 功能名称



(b) 绝对名称



3.2.3 ES 和 CS 寄存器

ES 寄存器用于数据访问，CS 寄存器用于在执行跳转指令时指定高位地址。

ES 寄存器复位后的默认值为 0FH，CS 寄存器为 00H。

图 3-17. ES 和 CS 寄存器的配置

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0
CS	0	0	0	0	CS3	CP2	CP1	CP0

3.2.4 特殊功能寄存器 (SFR)

与通用寄存器不同，每个特殊功能寄存器都有特定的功能。

特殊寄存器指定在 FFF00H 到 FFFFFH 区域。

特殊功能寄存器可以像通用寄存器那样使用运算指令、传送指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是1位、8位和16位。

每种位单元操作的描述如下。

- 1 位操作
1 位操作指令的操作数 (`sfr.bit`) 被描述为汇编器的保留符号。该操作也可由一个地址来定义。
- 8 位操作
8 位操作指令的操作数 (`sfr`) 被描述为汇编器的保留符号。该操作也可由一个地址来定义。
- 16 位操作
16 位操作指令的操作数 (`sfrp`) 被描述为汇编器的保留符号。寻址时必须为一个偶地址。

表 3-5 为特殊功能寄存器列表。表中术语的含义如下。

- 符号
表示特殊功能寄存器地址的符号，在 RA78K0R 中是保留字，也可以使用 CC78K0R 中的 `#pragma sfr` 伪指令定义一个 `sfr` 变量。在使用 RA78K0R, ID78K0R-QB 和 SM+ for 78K0R 时，可以将符号作为指令操作数执行写操作。
- R/W
表示特殊功能寄存器可读或可写。
R/W: 可读/写
R: 只读
W: 只写
- 可操作的位单元
“√”表示可操作的位单元（1, 8, 或 16）。“-”表示不可操作的位单元。
- 复位后
表示复位信号产生后每个寄存器的状态

注意事项 不要访问未分配扩展 SFRs 的地址。

备注 关于扩展 SFRs（第二 SFRs），参见 3.2.5 扩展特殊功能寄存器（第二 SFRs：第二特殊功能寄存器）。

表 3-5. SFR 列表 (1/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFF00H	端口寄存器 0	P0		R/W	√	√	-	00H
FFF01H	端口寄存器 1	P1		R/W	√	√	-	00H
FFF02H	端口寄存器 2	P2		R/W	√	√	-	00H
FFF03H	端口寄存器 3	P3		R/W	√	√	-	00H
FFF04H	端口寄存器 4	P4		R/W	√	√	-	00H
FFF05H	端口寄存器 5	P5		R/W	√	√	-	00H
FFF06H	端口寄存器 6	P6		R/W	√	√	-	00H
FFF07H	端口寄存器 7	P7		R/W	√	√	-	00H
FFF09H	端口寄存器 9	P9		R/W	√	√	-	00H
FFF0BH	端口寄存器 11	P11		R/W	√	√	-	00H
FFF0CH	端口寄存器 12	P12		R/W	√	√	-	00H
FFF0DH	端口寄存器 13	P13		R/W	√	√	-	00H
FFF0EH	端口寄存器 14	P14		R/W	√	√	-	00H
FFF10H	串行数据寄存器 00	TxD0/ SIO00	SDR00	R/W	-	√	√	0000H
FFF11H		-			-	-		
FFF12H	串行数据寄存器 01	RxD0	SDR01	R/W	-	√	√	0000H
FFF13H		-			-	-		
FFF14H	串行数据寄存器 12	TxD3	SDR12	R/W	-	√	√	0000H
FFF15H		-			-	-		
FFF16H	串行数据寄存器 13	RxD3	SDR13	R/W	-	√	√	0000H
FFF17H		-			-	-		
FFF18H	定时器数据寄存器 00	TDR00		R/W	-	-	√	0000H
FFF19H					-	-	-	
FFF1AH	定时器数据寄存器 01	TDR01		R/W	-	-	√	0000H
FFF1BH					-	-	-	
FFF1CH	8 位 D/A 转换值设置寄存器 0	DACS0		R/W	√	√	-	00H
FFF1DH	8 位 D/A 转换值设置寄存器 1	DACS1		R/W	√	√	-	00H
FFF1EH	10 位 A/D 转换结果寄存器	ADCR		R	-	-	√	0000H
FFF1FH	8 位 A/D 转换结果寄存器	ADCRH		R	-	√	-	00H
FFF20H	端口模式寄存器 0	PM0		R/W	√	√	-	FFH
FFF21H	端口模式寄存器 1	PM1		R/W	√	√	-	FFH
FFF22H	端口模式寄存器 2	PM2		R/W	√	√	-	FFH
FFF23H	端口模式寄存器 3	PM3		R/W	√	√	-	FFH
FFF24H	端口模式寄存器 4	PM4		R/W	√	√	-	FFH
FFF25H	端口模式寄存器 5	PM5		R/W	√	√	-	FFH
FFF26H	端口模式寄存器 6	PM6		R/W	√	√	-	FFH
FFF27H	端口模式寄存器 7	PM7		R/W	√	√	-	FFH
FFF29H	端口模式寄存器 9	PM9		R/W	√	√	-	FFH
FFF2BH	端口模式寄存器 11	PM11		R/W	√	√	-	FFH
FFF2CH	端口模式寄存器 12	PM12		R/W	√	√	-	FFH
FFF2DH	端口模式寄存器 13	PM13		R/W	√	√	-	FEH
FFF2EH	端口模式寄存器 14	PM14		R/W	√	√	-	FFH

表 3-5. SFR 列表 (2/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFF30H	A/D 转换器模式寄存器	ADM		R/W	√	√	–	00H
FFF31H	模拟输入通道指定寄存器	ADS		R/W	√	√	–	00H
FFF32H	D/A 转换器模式寄存器	DAM		R/W	√	√	–	00H
FFF37H	按键返回模式寄存器	KRM		R/W	√	√	–	00H
FFF38H	外部中断上升沿允许寄存器 0	EGP0		R/W	√	√	–	00H
FFF39H	外部中断下降沿允许寄存器 0	EGN0		R/W	√	√	–	00H
FFF3AH	外部中断上升沿允许寄存器 1	EGP1		R/W	√	√	–	00H
FFF3BH	外部中断下降沿允许寄存器 1	EGN1		R/W	√	√	–	00H
FFF3CH	输入切换控制寄存器	ISC		R/W	√	√	–	00H
FFF3EH	定时器输入选择寄存器 0	TIS0		R/W	√	√	–	00H
FFF44H	串行数据寄存器 02	TxD1/SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	串行数据寄存器 03	RxD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	串行数据寄存器 10	TxD2/SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	串行数据寄存器 11	RxD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IIC 移位寄存器 0	IIC0		R/W	–	√	–	00H
FFF51H	IIC 标志寄存器 0	IICF0		R/W	√	√	–	00H
FFF52H	IIC 控制寄存器 0	IICC0		R/W	√	√	–	00H
FFF53H	IIC slave 地址寄存器 0	SVA0		R/W	–	√	–	00H
FFF54H	IIC 时钟选择寄存器 0	IICCL0		R/W	√	√	–	00H
FFF55H	IIC 功能扩展寄存器 0	IICX0		R/W	√	√	–	00H
FFF56H	IIC 状态寄存器 0	IICCS0		R	√	√	–	00H
FFF64H	定时器数据寄存器 02	TDR02		R/W	–	–	√	0000H
FFF65H					–	–	–	
FFF66H	定时器数据寄存器 03	TDR03		R/W	–	–	√	0000H
FFF67H					–	–	–	
FFF68H	定时器数据寄存器 04	TDR04		R/W	–	–	√	0000H
FFF69H					–	–	–	
FFF6AH	定时器数据寄存器 05	TDR05		R/W	–	–	√	0000H
FFF6BH					–	–	–	
FFF6CH	定时器数据寄存器 06	TDR06		R/W	–	–	√	0000H
FFF6DH					–	–	–	
FFF6EH	定时器数据寄存器 07	TDR07		R/W	–	–	√	0000H
FFF6FH					–	–	–	

表 3-5. SFR 列表 (3/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
FFF90H	副计数寄存器	RSUBC	R	-	-	√	0000H
FFF91H							
FFF92H	秒计数寄存器	SEC	R/W	-	√	-	00H
FFF93H	分钟计数寄存器	MIN	R/W	-	√	-	00H
FFF94H	小时计数寄存器	HOUR	R/W	-	√	-	12H ^{#1}
FFF95H	星期计数寄存器	WEEK	R/W	-	√	-	00H
FFF96H	日计数寄存器	DAY	R/W	-	√	-	01H
FFF97H	月计数寄存器	MONTH	R/W	-	√	-	01H
FFF98H	年计数寄存器	YEAR	R/W	-	√	-	00H
FFF99H	钟表错误修正寄存器	SUBCUD	R/W	-	√	-	00H
FFF9AH	分钟报警寄存器	ALARMWWM	R/W	-	√	-	00H
FFF9BH	小时报警寄存器	ALARMWH	R/W	-	√	-	12H
FFF9CH	星期报警寄存器	ALARMWW	R/W	-	√	-	00H
FFF9DH	实时计数器控制寄存器 0	RTCC0	R/W	√	√	-	00H
FFF9EH	实时计数器控制寄存器 1	RTCC1	R/W	√	√	-	00H
FFF9FH	实时计数器控制寄存器 2	RTCC2	R/W	√	√	-	00H
FFFA0H	时钟操作模式控制寄存器	CMC	R/W	-	√	-	00H
FFFA1H	时钟操作状态控制寄存器	CSC	R/W	√	√	-	C0H
FFFA2H	振荡稳定时间计数器状态寄存器	OSTC	R	√	√	-	00H
FFFA3H	振荡稳定时间选择寄存器	OSTS	R/W	-	√	-	07H
FFFA4H	系统时钟控制寄存器	CKC	R/W	√	√	-	09H
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	√	√	-	00H
FFFA6H	时钟输出选择寄存器 1	CKS1	R/W	√	√	-	00H
FFFA8H	复位控制标志寄存器	RESF	R	-	√	-	00H ^{#2}
FFFA9H	低电压检测寄存器	LVIM	R/W	√	√	-	00H ^{#3}
FFFAAH	低电压检测电平选择寄存器	LVIS	R/W	√	√	-	0EH ^{#4}
FFFABH	看门狗定时器允许寄存器	WDTE	R/W	-	√	-	1A/9A ^{#5}
<R> FFFACH	-	TTBLH ^{#6}	-	-	-	-	未定义
FFFADH							
<R> FFFAEH	-	TTBLL ^{#6}	-	-	-	-	未定义
FFF AFH							

注 1. 如果 AMPM 位 (RTCC0 寄存器的第 3 位) 复位后设置为 1, 此寄存器的值为 00H。

2. RESF 的复位值根据复位源不同而不同。

3. LVIM 的复位值根据复位源和选项字节的设置的不同而不同。

4. LVIS 的复位值根据复位源不同而不同。

5. WDTE 的复位值由选项字节的设置决定。

6. 该 SFR 不能由用户使用, 所以不要直接对其操作。

表 3-5. SFR 列表 (4/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFFB0H	DMA SFR 地址寄存器 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR 地址寄存器 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM 地址寄存器 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM 地址寄存器 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM 地址寄存器 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM 地址寄存器 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA 字节计数寄存器 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA 字节计数寄存器 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA 字节计数寄存器 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA 字节计数寄存器 1H	DBC1H		R/W	–	√		00H
FFFB AH	DMA 模式控制寄存器 0	DMC0		R/W	√	√	–	00H
FFFB BH	DMA 模式控制寄存器 1	DMC1		R/W	√	√	–	00H
FFFB CH	DMA 操作控制寄存器 0	DRC0		R/W	√	√	–	00H
FFFB DH	DMA 操作控制寄存器 1	DRC1		R/W	√	√	–	00H
FFFB EH	背景事件控制 寄存器	BECTL		R/W	√	√	–	00H
<R> FFFC0H	–	PFCMD [※]		–	–	–	–	未定义
<R> FFFC2H	–	PFS [※]		–	–	–	–	未定义
<R> FFFC4H	–	FLPMC [※]		–	–	–	–	未定义
FFFD0H	中断请求标志寄存器 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	中断请求标志寄存器 2H	IF2H		R/W	√	√		00H
FFFD4H	中断屏蔽标志寄存器 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	中断屏蔽标志寄存器 2H	MK2H		R/W	√	√		FFH
FFFD8H	优先级指定标志寄存器 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	优先级指定标志寄存器 02H	PR02H		R/W	√	√		FFH
FFFDCH	优先级指定标志寄存器 12L	PR12L	PR12	R/W	√	√	√	FFH
FFDDH	优先级指定标志寄存器 12H	PR12H		R/W	√	√		FFH
FFFE0H	中断请求标志寄存器 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	中断请求标志寄存器 0H	IF0H		R/W	√	√		00H
FFFE2H	中断请求标志寄存器 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	中断请求标志寄存器 1H	IF1H		R/W	√	√		00H
FFFE4H	中断屏蔽标志寄存器 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	中断屏蔽标志寄存器 0H	MK0H		R/W	√	√		FFH
FFFE6H	中断屏蔽标志寄存器 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	中断屏蔽标志寄存器 1H	MK1H		R/W	√	√		FFH
FFFE8H	优先级指定标志寄存器 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	优先级指定标志寄存器 00H	PR00H		R/W	√	√		FFH
FFFE AH	优先级指定标志寄存器 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFE BH	优先级指定标志寄存器 01H	PR01H		R/W	√	√		FFH
FFFE CH	优先级指定标志寄存器 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFE DH	优先级指定标志寄存器 10H	PR10H		R/W	√	√		FFH

<R> 注 不要直接操作该 SFR，因为它在自编程库中被使用。

表 3-5. SFR 列表 (5/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFFEEH	优先级指定标志寄存器 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	优先级指定标志寄存器 11H	PR11H		R/W	√	√		FFH
FFFF0H	乘法输入数据寄存器 A	MULA		R/W	-	-	√	0000H
FFFF1H								
FFFF2H	乘法输入数据寄存器 B	MULB		R/W	-	-	√	0000H
FFFF3H								
FFFF4H	高位乘法结果存储寄存器	MULOH		R	-	-	√	0000H
FFFF5H								
FFFF6H	低位乘法结果存储寄存器	MULOL		R	-	-	√	0000H
FFFF7H								
FFFEH	处理器模式控制寄存器	PMC		R/W	√	√	-	00H

备注 关于扩展 SFR (第二 SFR)，参见表 3-6 扩展 SFR (第二 SFR) 列表。

3.2.5 扩展特殊功能寄存器（第二SFR：第二特殊功能寄存器）

与通用寄存器不同，每个扩展 SFR（第二 SFR）都有特殊的功能。

扩展 SFR 分配在 F0000H 到 F07FFH 区域内。分配在 SFR 区域（FFF00H~FFFFFH）以外的 SFR 指定在此区域。访问扩展 SFR 区域的指令，比访问 SFR 区域的指令长 1 字节。

扩展 SFR 可像通用寄存器那样用运算指令、传送指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是 1 位、8 位和 16 位。

每种位单元操作的描述如下。

- 1 位操作
1 位操作指令的操作数 (!addr16.bit) 被描述为汇编器的保留符号。该操作也可由一个地址来定义。
- 8 位操作
8 位操作指令的操作数 (!addr16) 被描述为汇编器的保留符号。该操作也可由一个地址来定义。
- 16 位操作
16 位操作指令的操作数 (!addr16) 被描述为汇编器的保留符号。寻址时必须为一个偶地址。

表 3-6 为扩展 SFR 列表。表中术语的含义如下。

- 符号
表示扩展 SFR 地址的符号，在 RA78K0R 中是保留字，并使用 CC78K0R 中的 #pragma sfr 伪指令定义为一个 sfr 变量。在使用 RA78K0R, ID78K0R-QB 和 SM+ for 78K0R 时，可以将符号作为指令操作数执行写操作。
- R/W
表示扩展 SFR 可读或可写。
R/W: 可读/写
R: 只读
W: 只写
- 可操作的位单元
“√”表示可操作的位单元（1, 8 或 16）。“-”表示不可操作的位单元。
- 复位后
表示复位后每个寄存器的状态。

注意事项 不要访问未分配扩展 SFR 的地址。

备注 对于 SFR 区域的 SFR，参见 3.2.4 特殊功能寄存器（SFRs）。

表 3-6. 扩展 SFR (第二 SFR) 列表 (1/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后	
				1 位	8 位	16 位		
F0017H	A/D 端口配置寄存器	ADPC	R/W	–	√	–	10H	
F0030H	上拉电阻选项寄存器 0	PU0	R/W	√	√	–	00H	
F0031H	上拉电阻选项寄存器 1	PU1	R/W	√	√	–	00H	
F0033H	上拉电阻选项寄存器 3	PU3	R/W	√	√	–	00H	
F0034H	上拉电阻选项寄存器 4	PU4	R/W	√	√	–	00H	
F0035H	上拉电阻选项寄存器 5	PU5	R/W	√	√	–	00H	
F0037H	上拉电阻选项寄存器 7	PU7	R/W	√	√	–	00H	
F0039H	上拉电阻选项寄存器 9	PU9	R/W	√	√	–	00H	
F003CH	上拉电阻选项寄存器 12	PU12	R/W	√	√	–	00H	
F003EH	上拉电阻选项寄存器 14	PU14	R/W	√	√	–	00H	
F0040H	端口输入模式寄存器 0	PIM0	R/W	√	√	–	00H	
F0044H	端口输入模式寄存器 4	PIM4	R/W	√	√	–	00H	
F004CH	端口输入模式寄存器 12	PIM12	R/W	√	√	–	00H	
F004EH	端口输入模式寄存器 14	PIM14	R/W	√	√	–	00H	
F0050H	端口输出模式寄存器 0	POM0	R/W	√	√	–	00H	
F0054H	端口输出模式寄存器 4	POM4	R/W	√	√	–	00H	
F005CH	端口输出模式寄存器 12	POM12	R/W	√	√	–	00H	
F005EH	端口输出模式寄存器 14	POM14	R/W	√	√	–	00H	
F0060H	噪声滤波器允许寄存器 0	NFEN0	R/W	√	√	–	00H	
F0061H	噪声滤波器允许寄存器 1	NFEN1	R/W	√	√	–	00H	
F0064H	上拉电阻选项寄存器 16	PU16	R/W	√	√	–	00H	
F00F0H	外围设备 允许寄存器 0	PER0	R/W	√	√	–	00H	
F00F2H	内部高速振荡器调整寄存器	HIOTRM	R/W	–	√	–	10H	
F00F3H	操作速度模式控制寄存器	OSMC	R/W	–	√	–	00H	
F00F4H	调节器模式控制寄存器	RMC	R/W	–	√	–	00H	
<R> F00FEH	BCD 调整结果寄存器	BCDADJ	R	–	√	–	Undefined	
F0100H	串行 状态 寄存器 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–			
F0102H	串行 状态 寄存器 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–			
F0104H	串行 状态 寄存器 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–			
F0106H	串行 状态 寄存器 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–			
F0108H	串行 标志 清零触发寄存器 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–			
F010AH	串行 标志 清零触发寄存器 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–			
F010CH	串行 标志 清零触发寄存器 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–			
F010EH	串行 标志 清零触发寄存器 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–			

表 3-6. 扩展 SFR (第二 SFR) 列表 (2/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
F0110H	串行模式寄存器 00	SMR00		R/W	-	-	√	0020H
F0111H								
F0112H	串行模式寄存器 01	SMR01		R/W	-	-	√	0020H
F0113H								
F0114H	串行模式寄存器 02	SMR02		R/W	-	-	√	0020H
F0115H								
F0116H	串行模式寄存器 03	SMR03		R/W	-	-	√	0020H
F0117H								
F0118H	串行通信操作设置寄存器 00	SCR00		R/W	-	-	√	0087H
F0119H								
F011AH	串行通信操作设置寄存器 01	SCR01		R/W	-	-	√	0087H
F011BH								
F011CH	串行通信操作设置寄存器 02	SCR02		R/W	-	-	√	0087H
F011DH								
F011EH	串行通信操作设置寄存器 03	SCR03		R/W	-	-	√	0087H
F011FH								
F0120H	串行通道允许状态 寄存器 0	SE0L	SE0	R	√	√	√	0000H
F0121H		-			-	-		
F0122H	串行通道开始触发寄存器 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		-			-	-		
F0124H	串行通道停止触发寄存器 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		-			-	-		
F0126H	串行 时钟选择寄存器 0	SPS0L	SPS0	R/W	-	√	√	0000H
F0127H		-			-	-		
F0128H	串行输出寄存器 0	SO0		R/W	-	-	√	0F0FH
F0129H								
F012AH	串行输出允许寄存器 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		-			-	-		
<R> F0134H	串行输出电平 寄存器 0	SOL0L	SOL0	R/W	-	√	√	0000H
<R> F0135H		-			-	-		
F0140H	串行 状态 寄存器 10	SSR10L	SSR10	R	-	√	√	0000H
F0141H		-			-	-		
F0142H	串行 状态 寄存器 11	SSR11L	SSR11	R	-	√	√	0000H
F0143H		-			-	-		
F0144H	串行 状态 寄存器 12	SSR12L	SSR12	R	-	√	√	0000H
F0145H		-			-	-		
F0146H	串行 状态 寄存器 13	SSR13L	SSR13	R	-	√	√	0000H
F0147H		-			-	-		
F0148H	串行 标志 清零触发寄存器 10	SIR10L	SIR10	R/W	-	√	√	0000H
F0149H		-			-	-		
F014AH	串行 标志 清零触发寄存器 11	SIR11L	SIR11	R/W	-	√	√	0000H
F014BH		-			-	-		

表 3-6. 扩展 SFR (第二 SFR) 列表 (3/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
F014CH	串行标志 清零触发寄存器 12	SIR12L	SIR12	R/W	-	√	√	0000H
F014DH		-			-			
F014EH	串行标志 清零触发寄存器 13	SIR13L	SIR13	R/W	-	√	√	0000H
F014FH		-			-			
F0150H	串行模式寄存器 10	SMR10		R/W	-	-	√	0020H
F0151H								
F0152H	串行模式寄存器 11	SMR11		R/W	-	-	√	0020H
F0153H								
F0154H	串行模式寄存器 12	SMR12		R/W	-	-	√	0020H
F0155H								
F0156H	串行模式寄存器 13	SMR13		R/W	-	-	√	0020H
F0157H								
F0158H	串行通信操作设置寄存器 10	SCR10		R/W	-	-	√	0087H
F0159H								
F015AH	串行通信操作设置寄存器 11	SCR11		R/W	-	-	√	0087H
F015BH								
F015CH	串行通信操作设置寄存器 12	SCR12		R/W	-	-	√	0087H
F015DH								
F015EH	串行通信操作设置寄存器 13	SCR13		R/W	-	-	√	0087H
F015FH								
F0160H	串行通道允许状态寄存器 1	SE1L	SE1	R	√	√	√	0000H
F0161H		-			-			
F0162H	串行通道开始触发寄存器 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		-			-			
F0164H	串行通道停止触发寄存器 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		-			-			
F0166H	串行时钟选择寄存器 1	SPS1L	SPS1	R/W	-	√	√	0000H
F0167H		-			-			
F0168H	串行输出寄存器 1	SO1		R/W	-	-	√	0F0FH
F0169H								
F016AH	串行输出允许寄存器 1	SOE1L	SOE1L	R/W	√	√	√	0000H
F016BH		-			-			
F0174H	串行输出电平寄存器 1	SOL1L	SOL1	R/W	-	√	√	0000H
F0175H		-			-			
F0180H	定时器计数器寄存器 00	TCR00		R	-	-	√	FFFFH
F0181H								
F0182H	定时器计数器寄存器 01	TCR01		R	-	-	√	FFFFH
F0183H								
F0184H	定时器计数器寄存器 02	TCR02		R	-	-	√	FFFFH
F0185H								
F0186H	定时器计数器寄存器 03	TCR03		R	-	-	√	FFFFH
F0187H								

<R>
<R>

表 3-6. 扩展 SFR (第二 SFR) 列表 (4/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
F0188H	定时器计数器寄存器 04	TCR04	R	-	-	√	FFFFH	
F0189H								
F018AH	定时器计数器寄存器 05	TCR05	R	-	-	√	FFFFH	
F018BH								
F018CH	定时器计数器寄存器 06	TCR06	R	-	-	√	FFFFH	
F018DH								
F018EH	定时器计数器寄存器 07	TCR07	R	-	-	√	FFFFH	
F018FH								
F0190H	定时器模式寄存器 00	TMR00	R/W	-	-	√	0000H	
F0191H								
F0192H	定时器模式寄存器 01	TMR01	R/W	-	-	√	0000H	
F0193H								
F0194H	定时器模式寄存器 02	TMR02	R/W	-	-	√	0000H	
F0195H								
F0196H	定时器模式寄存器 03	TMR03	R/W	-	-	√	0000H	
F0197H								
F0198H	定时器模式寄存器 04	TMR04	R/W	-	-	√	0000H	
F0199H								
F019AH	定时器模式寄存器 05	TMR05	R/W	-	-	√	0000H	
F019BH								
F019CH	定时器模式寄存器 06	TMR06	R/W	-	-	√	0000H	
F019DH								
F019EH	定时器模式寄存器 07	TMR07	R/W	-	-	√	0000H	
F019FH								
F01A0H	定时器状态寄存器 00	TSR00L	TSR00	R	-	√	√	0000H
F01A1H		-			-			
F01A2H	定时器状态寄存器 01	TSR01L	TSR01	R	-	√	√	0000H
F01A3H		-			-			
F01A4H	定时器状态寄存器 02	TSR02L	TSR02	R	-	√	√	0000H
F01A5H		-			-			
F01A6H	定时器状态寄存器 03	TSR03L	TSR03	R	-	√	√	0000H
F01A7H		-			-			
F01A8H	定时器状态寄存器 04	TSR04L	TSR04	R	-	√	√	0000H
F01A9H		-			-			
F01AAH	定时器状态寄存器 05	TSR05L	TSR05	R	-	√	√	0000H
F01ABH		-			-			
F01ACH	定时器状态寄存器 06	TSR06L	TSR06	R	-	√	√	0000H
F01ADH		-			-			
F01AEH	定时器状态寄存器 07	TSR07L	TSR07	R	-	√	√	0000H
F01AFH		-			-			

表 3-6. 扩展 SFR (第二 SFR) 列表 (5/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
F01B0H	定时器通道允许状态 寄存器 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–			
F01B2H	定时器通道开始触发寄存器 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–			
F01B4H	定时器通道停止触发寄存器 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–			
F01B6H	定时器时钟选择寄存器 0	TPS0L	TPS0	R/W	–	√	√	0000H
F01B7H		–			–			
F01B8H	定时器输出寄存器 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–			
F01BAH	定时器输出允许寄存器 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–			
F01BCH	定时器输出电平寄存器 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–			
F01BEH	定时器 输出模式寄存器 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–			

备注 关于 SFR 区域中的 SFR, 参见表 3-5 SFR 列表。

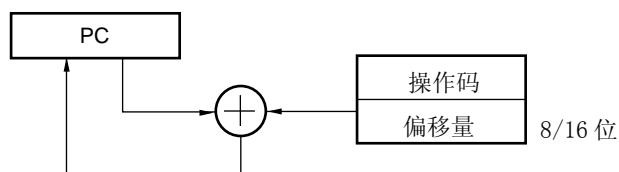
3.3 指令地址寻址

3.3.1 相对寻址

[功能]

相对寻址结果存储在程序计数器 (PC)，程序计数器 (PC) 的值 (下一条指令的开始地址) 加上包括在指令字 (带符号的补码数据: -128~ +127 或 -32768~+32767) 中的偏移量的值, 并指定分支跳转目标的程序地址。相对寻址仅使用于分支跳转指令。

图 3-18. 相对寻址的概述



3.3.2 立即寻址

[功能]

将指令中的立即数赋给程序计数器 (PC)，然后转向该地址。

对于立即寻址, CALL !!addr20 或 BR !!addr20 用作指定 20 位地址, CALL !addr16 或 BR !addr16 用作指定 16 位地址。当指定 16 位地址时高 4 位设置为 0000。

图 3-19. CALL !!addr20/BR !!addr20 的例子

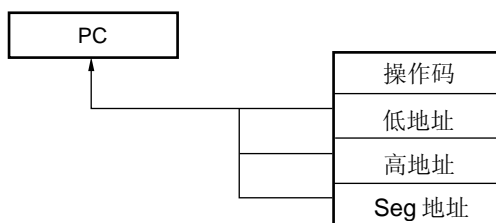
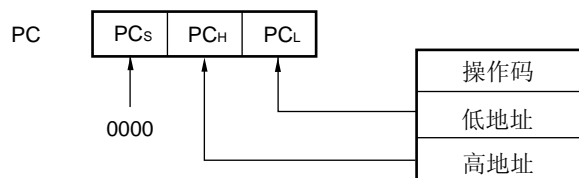


图 3-20. CALL !addr16/BR !addr16 的例子

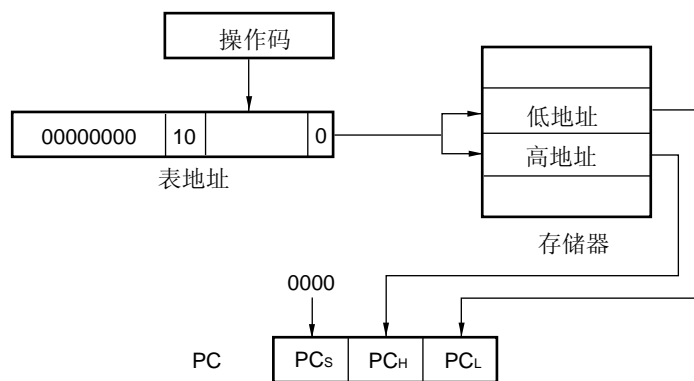


3.3.3 表间接寻址

[功能]

通过指令字中的指令码低 5 位的立即数在 CALLT 表区域 (0080H~ 00BFH) 指定表地址，将表地址和下一个地址作为 16 位数存储在程序计数器 (PC) 中，并指定程序地址。表间接寻址仅适用于 CALLT 指令。在 78K0R 微控制器种，分支跳转范围仅允许在从 00000H 到 0FFFFH 的 64 KB 空间中。

图 3-21. 表间接寻址的概述

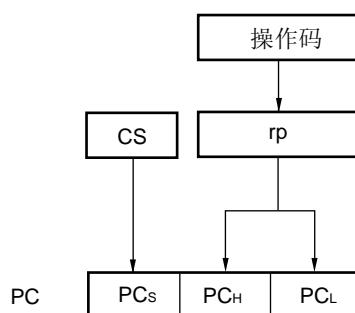


3.3.4 寄存器直接寻址

[功能]

寄存器直接寻址结果存储在程序计数器（PC）中，通过指令字的 20 位数据将当前寄存器组的通用寄存器对（AX/BC/DE/HL）和当前寄存器组的 CS 寄存器的内容赋给程序计数器（PC），指定程序地址。寄存器直接寻址仅适用于 CALL AX, BC, DE, HL 和 BR AX 指令。

图 3-22. 寄存器直接寻址概况



3.4 处理数据地址寻址

3.4.1 隐含寻址

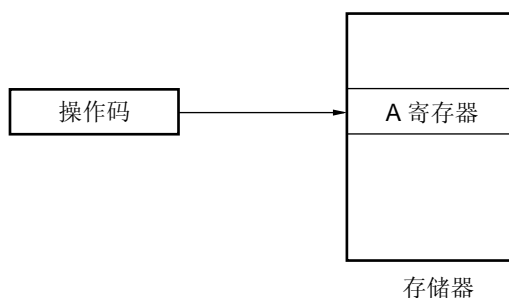
[功能]

访问那些具有特殊功能的寄存器（比如累加器）的指令由指令字直接指定，在指令字的中无需使用任何寄存器指定字段。

[操作数格式]

由于指令自动采用隐含寻址方式，所以没有特定的操作数格式。
 隐含寻址仅适用于 MULU X。

图 3-23. 隐含寻址概况



3.4.2 寄存器寻址

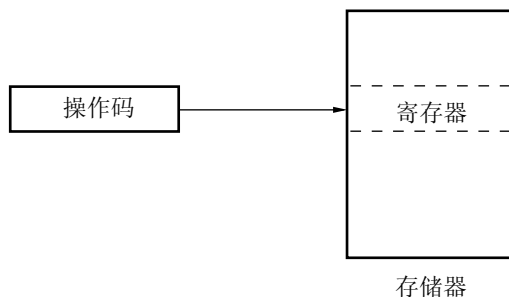
[功能]

寄存器寻址方式将通用寄存器作为操作数进行访问。3 位的指令字用于选择 8 位寄存器，2 位的指令字用于选择 16 位寄存器。

[操作数格式]

标识符	描述
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

图 3-24. 寄存器寻址概况



3.4.3 直接寻址

[功能]

将指令字中的立即数作为操作数地址进行直接寻址操作。

[操作数格式]

标识符	描述
ADDR16	标号或16位立即数（仅适用于从F0000H到FFFFFH的空间）
ES: ADDR16	标号或16位立即数（通过ES寄存器指定高4位地址）

图 3-25. ADDR16 的示例

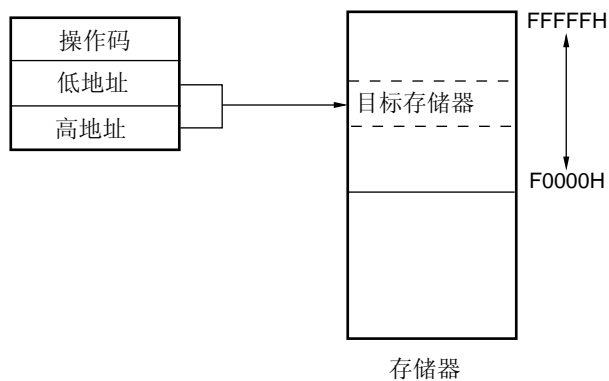
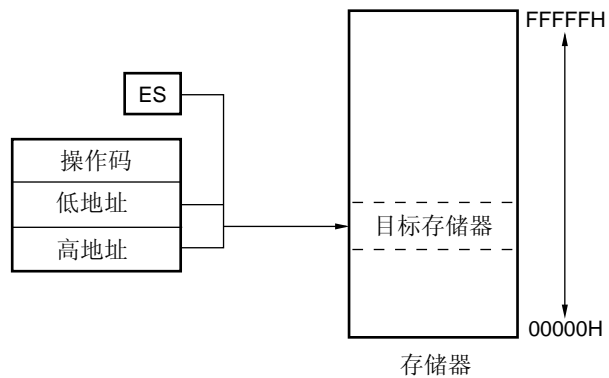


图 3-26. ES: ADDR16 的示例



3.4.4 短直接寻址

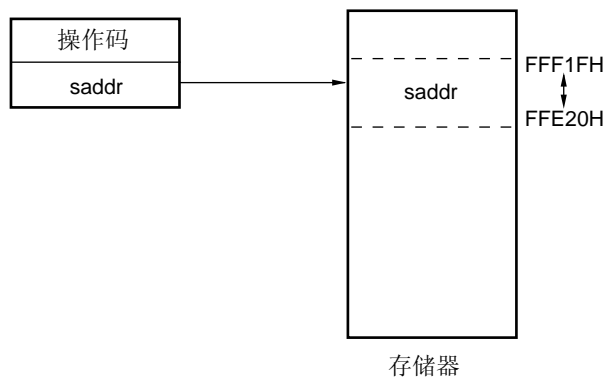
[功能]

用指令字中的 8 位数据直接指定目标地址。这种寻址仅适用于从 FFE20H 到 FFF1FH 的空间。

[操作数格式]

标识符	描述
SADDR	标号, FFE20H 到 FFF1FH 的立即数或 0FE20H 到 0FF1FH 的立即数 (仅适用于从 FFE20H 到 FFF1FH 的空间)
SADDRP	标号, FFE20H 到 FFF1FH 的立即数或 0FE20H 到 0FF1FH 的立即数 (仅偶地址) (仅适用于从 FFE20H 到 FFF1FH 的空间)

图 3-27. 短直接寻址概况



备注 SADDR 和 SADDRP 以 16 位立即数 (实际地址的高 4 位忽略) 表示地址 FE20H 到 FF1FH 的值, 以 20 位立即数表示地址 FFE20H 到 FFF1FH 的值。
不管是否使用 SADDR 或 SADDRP, 从 FFE20H 到 FFF1FH 空间中的地址被指定为存储器。

3.4.5 SFR 寻址

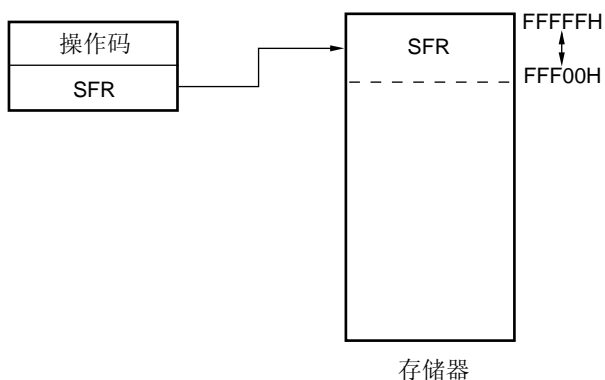
[功能]

用指令字中的 8 位数据直接指定目标 SFR 地址。这种寻址仅适用于从 FFF00H 到 FFFFFH 的空间。

[操作数格式]

标识符	描述
SFR	SFR 名称
SFRP	16位操作SFR 名称（仅偶地址）

图 3-28. SFR 寻址概况



3.4.6 寄存器间接寻址

[功能]

寄存器间接寻址直接使用寄存器对的内容指定目标地址，该寄存器对由指令字中的操作数地址指定。

[操作数格式]

标识符	描述
-	[DE], [HL] (仅适用于从F0000H 到 FFFFFH 的空间)
-	ES: [DE], ES: [HL] (通过ES寄存器指定高4位地址)

图 3-29. [DE], [HL]的示例

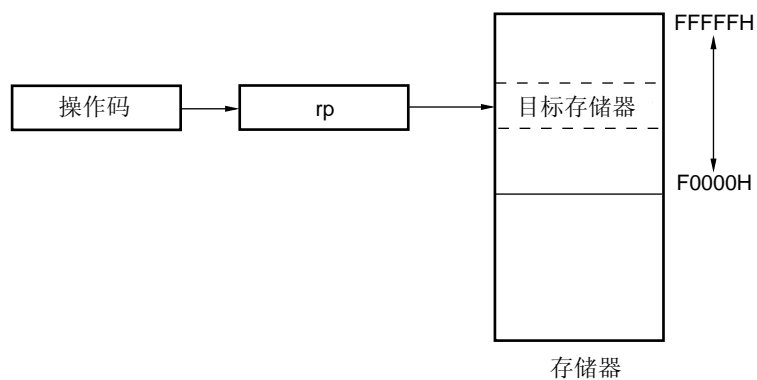
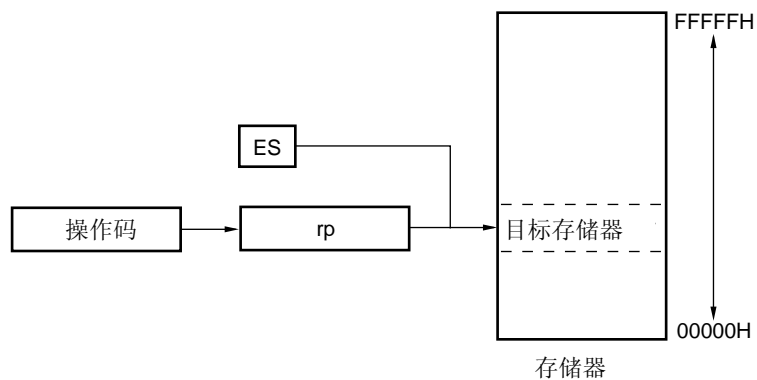


图 3-30. ES: [DE], ES: [HL]的示例



3.4.7 基址寻址

[功能]

由指令字指定寄存器对的值作为基址，8 位立即数或 16 位立即数作为偏移量。相加结果用于指定目标地址。

[操作数格式]

标识符	描述
-	[HL + byte], [DE + byte], [SP + byte] (仅适用于从F0000H 到 FFFFFH 的空间)
-	word[B], word[C] (仅适用于从F0000H 到 FFFFFH 的空间)
-	word[BC] (仅适用于从F0000H 到 FFFFFH 的空间)
-	ES: [HL + byte], ES: [DE + byte] (通过ES寄存器指定高4位地址)
-	ES: word[B], ES: word[C] (通过ES寄存器指定高4位地址)
-	ES: word[BC] (通过ES寄存器指定高4位地址)

图 3-31. [SP+byte]的示例

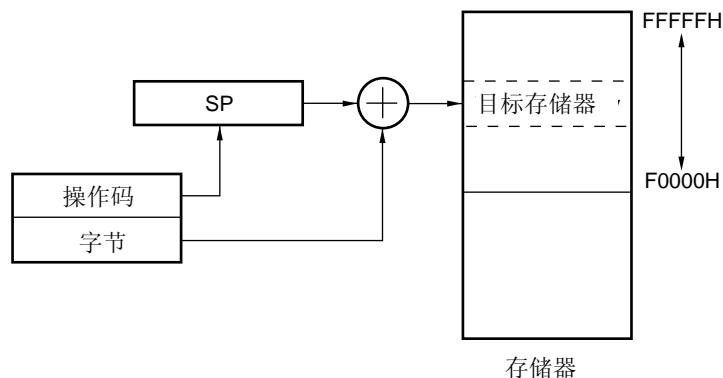


图 3-32. [HL + byte], [DE + byte]的示例

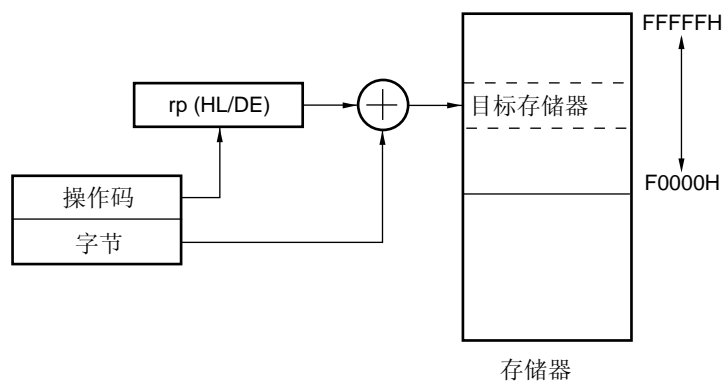


图 3-33. word[B], word[C]的示例

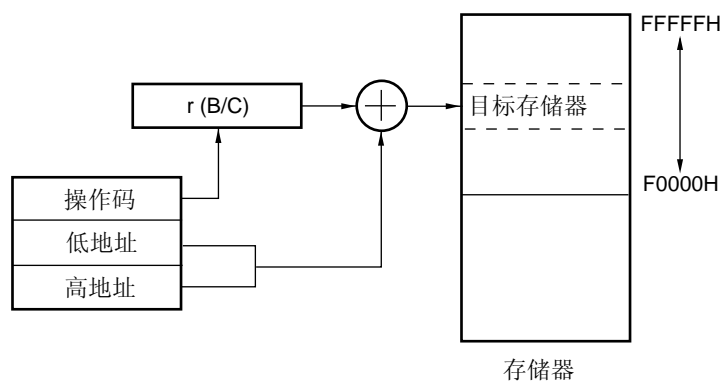


图 3-34. word[BC]的示例

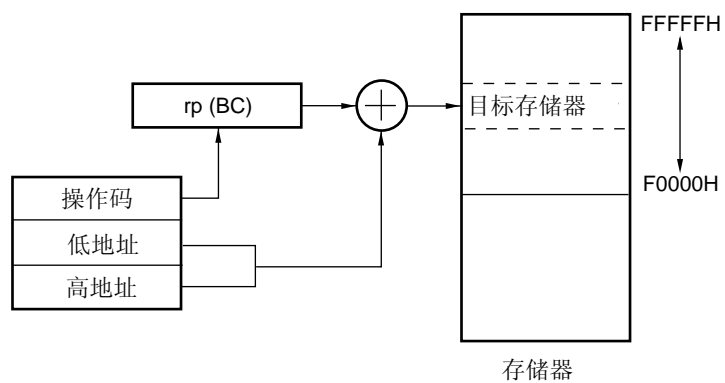


图 3-35. ES: [HL + byte], ES: [DE + byte]的示例

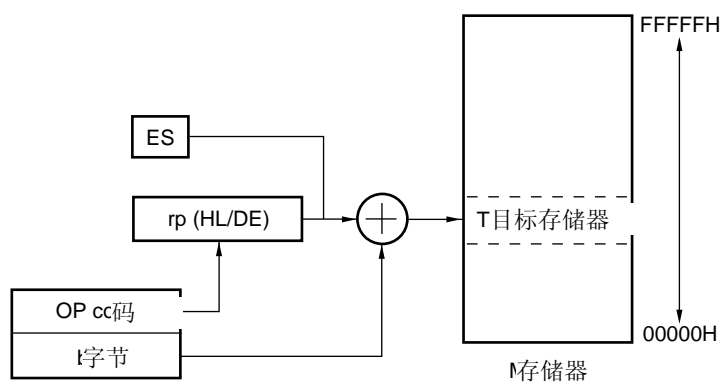


图 3-36. ES: word[B], ES: word[C] 的示例

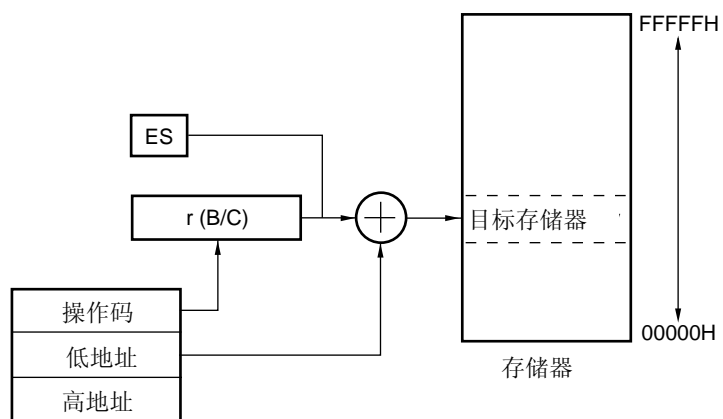
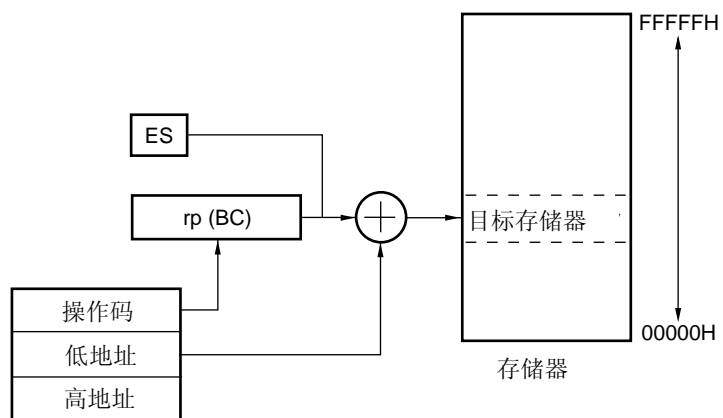


图 3-37. ES: word[BC] 的示例



3.4.8 基址变址寻址

[功能]

由指令字指定寄存器对的值作为基址，将作为偏移量的 B 寄存器或 C 寄存器的内容加到 HL 寄存器中，相加结果用于指定目标地址。

[操作数格式]

标识符	描述
-	[HL+B], [HL+C] (仅适用于从F0000H到FFFFFH的空间)
-	ES: [HL+B], ES: [HL+C] (通过ES寄存器指定高4位地址)

图 3-38. [HL+B], [HL+C] 的示例

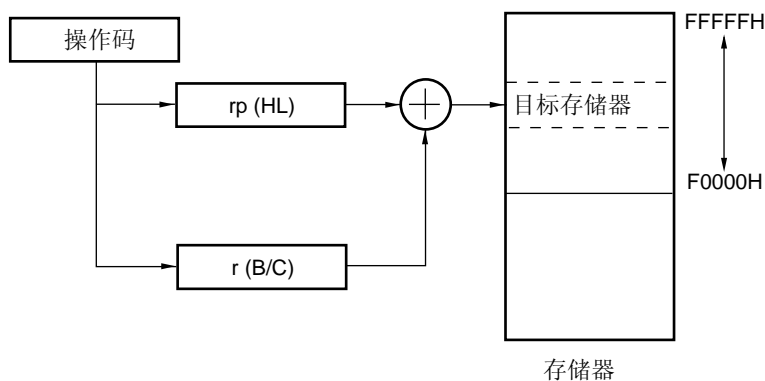
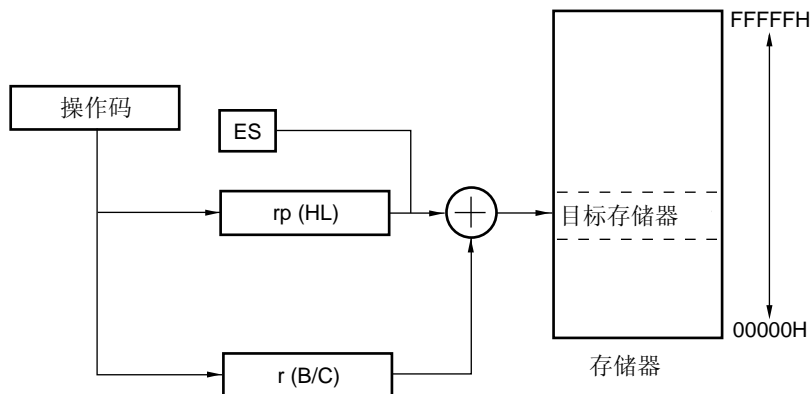


图 3-39. ES: [HL+B], ES: [HL+C] 的示例



3.4.9 堆栈寻址

[功能]

根据堆栈指针（SP）的内容对堆栈区域进行间接寻址。当执行 PUSH, POP, 子程序调用, 和返回指令时, 或者产生中断请求时保存或恢复寄存器操作时, 将自动采用这种寻址方式。

该方式仅对内部 RAM 区域进行寻址。

[操作数格式]

标识符	描述
-	PUSH AX/BC/DE/HL POP AX/BC/DE/HL CALL/CALLT RET BRK RETB (产生中断请求) RETI

第四章 端口功能

4.1 端口功能

有四种类型的引脚 I/O 缓冲电源： AV_{REF0} ， AV_{REF1} ， EV_{DD} 和 V_{DD} 。这些电源和引脚之间的关系显示如下。

表 4-1. 引脚 I/O 缓冲器电源

供电电源	对应的引脚
AV_{REF0}	P20~P27
AV_{REF1}	P110, P111
EV_{DD}	<ul style="list-style-type: none"> 除 P20~P27, P110, P111 和 P121~P124 以外的端口引脚 \overline{RESET} 引脚和 FLMD0 引脚
V_{DD}	<ul style="list-style-type: none"> P121~P124 端口引脚以外的引脚（除 \overline{RESET} 引脚和 FLMD0 引脚之外）

78K0R/KF3 产品中提供的端口如图 4-1 所示，这些端口可以支持多种控制操作。每个端口的功能如表 4-2 所示。除了作为数字 I/O 端口功能，这些端口还有几个复用功能。如需了解这些端口复用功能的详细信息，请参见第二章 引脚功能。

图 4-1. 端口类型

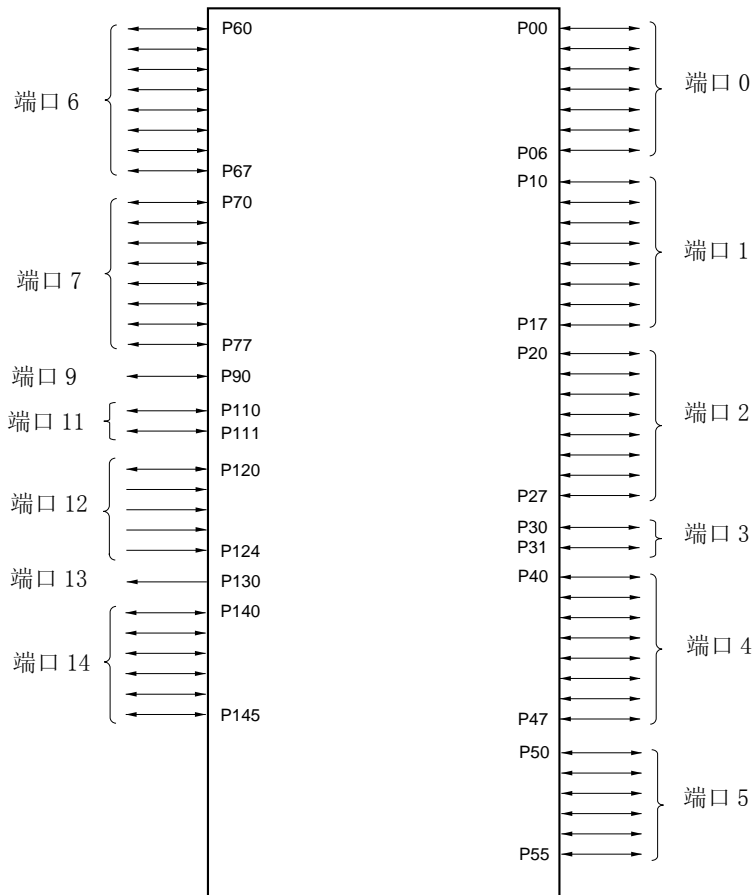


表 4-2. 端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 7 位 I/O 端口。 P03 和 P04 的输入可被设置到 TTL 缓冲器。 P02~P04 的输出可设置到 N-ch 漏极开路输出 (V _{DD} 耐压)。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05				TI05/TO05
P06				TI06/TO06
P10	I/O	端口 1。 8 位 I/O 端口。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	$\overline{\text{SCK00}}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20~P27	I/O	端口 2。 8 位 I/O 端口。 可以位操作选择输入/输出模式。	数字输入端口	ANI0~ANI7
P30	I/O	端口 3。 2 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40 ^注	I/O	端口 4。 8 位 I/O 端口。 P43 和 P44 的输入可被设置到 TTL 缓冲器。 P43 和 P45 的输出可设置到 N-ch 漏极开路输出 (V _{DD} 耐压) 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				$\overline{\text{SCK01}}$
P44				SI01
P45				SO01
P46				-
P47				-
P50	I/O	端口 5。 6 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	INTP1
P51				INTP2
P52				-
P53				-
P54				-
P55				-

注 如果通过选项字节允许片上调试, 请务必确保对 P40/TOOL0 引脚外部上拉 (参见 2.2.5 P40~P47 (端口 4) 中的**注意事项**)。

表 4-2. 端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
P60	I/O	端口 6。 8 位 I/O 端口。 P60~P63 的输出可设置为 N-ch 漏极开路输出 (6 V 耐压)。 可以位操作选择输入/输出模式。 只有 P64~P67 可以通过软件设置, 定义是否使用内置上拉电阻。	输入端口	SCL0
P61				SDA0
P62				-
P63				-
P64				-
P65				-
P66				-
P67				-
P70~P73	I/O	端口 7。 8 位 I/O 端口。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	KR0~KR3
P74~P77				KR4/INTP8 ~ KR7/INTP11
P90	I/O	端口 9。 1 位 I/O 端口。 可以位 (bit) 操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	-
P110	I/O	端口 11。 2 位 I/O 端口。 可以位操作选择输入/输出模式。	输入端口	ANO0
P111				ANO1
P120	I/O	端口 12。 1 位 I/O 端口和 4 位输入端口。 仅有 P120 可以通过软件设置, 定义是否使用内置上拉电阻。	输入端口	INTP0/EXLVI
P121				X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	输出	端口 13, 1 位输出端口。	输出端口	-
P140	I/O	端口 14。 6 位 I/O 端口。 P142 和 P143 的输入可被设置到 TTL 缓冲器。 P142~P144 的输出可设置为 N-ch 漏极开路输出 (V _{DD} 耐压)。 可以位操作选择输入/输出模式。 通过软件设置, 可以定义是否使用内置上拉电阻。	输入端口	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07

4.2 端口配置

端口包括如下硬件。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0~PM7, PM9, PM11, PM12, PM14) 端口寄存器 (P0~P7, P9, P11, P14) 上拉电阻选项寄存器 (PU0, PU1, PU3~PU7, PU9, PU12, PU14) 端口输入模式寄存器 (PIM0, PIM4, PIM14) 端口输出模式寄存器 (POM0, POM4, POM14) A/D 端口配置寄存器 (ADPC)
端口	总计: 70 (CMOS I/O: 61, CMOS 输入: 4, CMOS 输出: 1, N-ch 漏极开路 I/O: 4)
上拉电阻	总计: 51

4.2.1 端口 0

端口 0 是具有输出锁存的 7 位 I/O 端口。通过使用端口模式寄存器 0 (PM0)，可以按位设置端口 0 为输入模式或输出模式。如果 P00 ~ P06 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 0 (PU0) 以 1 位单元的方式指定。

通过使用端口输入模式寄存器 0 (PIM0) 以 1 位单元的方式可指定 P03 和 P04 引脚通过普通缓冲器或 TTL 缓冲器输入。

通过端口输出模式寄存器 0 (POM0) 以 1 位为单元的方式可指定 P02~P04 引脚的输出为 N-ch 漏极开路输出 (V_{DD} 耐压)。

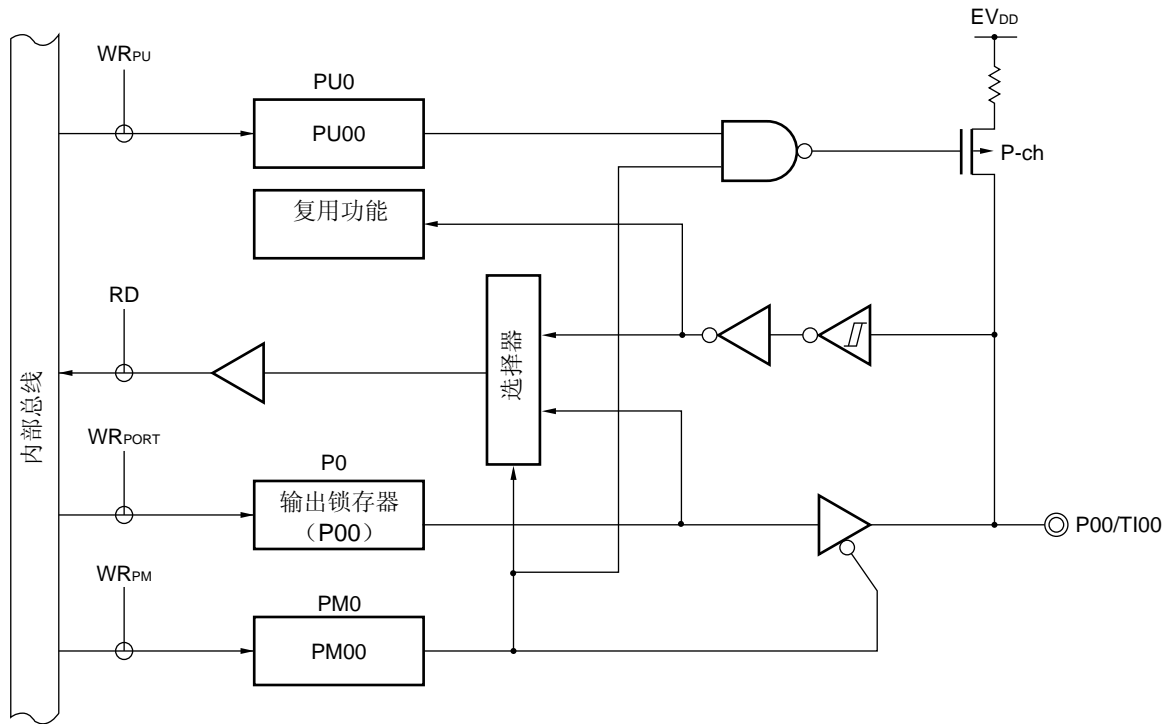
此端口也可用于定时器 I/O、串行接口数据 I/O 和时钟 I/O。

复位信号的发生将设置端口 0 为输入模式。

图 4-2~4-6 所示为端口 0 的框图。

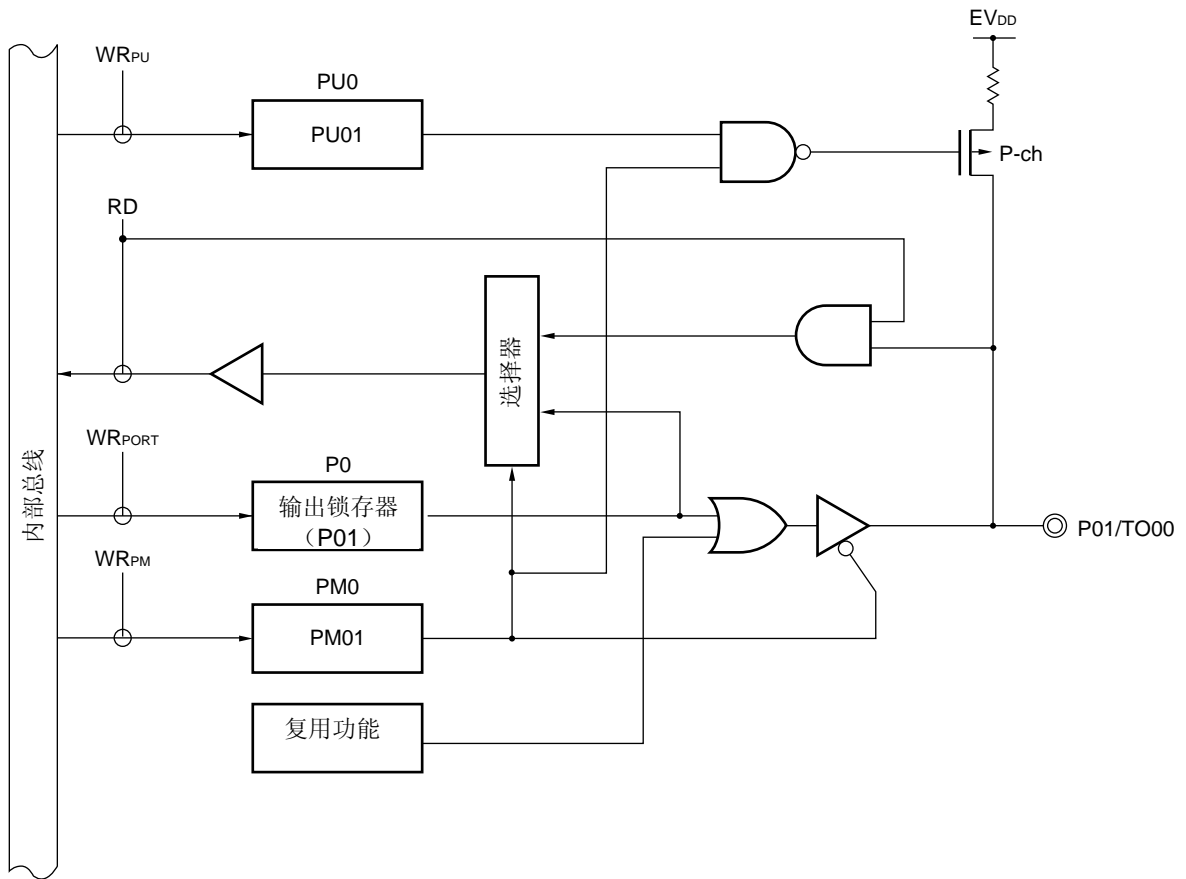
- <R> 注意事项 1. 要将 P01/TO00, P05/TI05/TO05, P06/TI06/TO06 作为通用端口使用, 设置定时器输出寄存器 0 (TO0) 的第 0, 5 和第 6 位 (TO00, TO05, TO06) 和定时器输出允许寄存器 0 (TOE0) 的第 0, 5 和第 6 位 (TOE00, TOE05, TOE06) 为“0”, 这也就是这些寄存器的默认值。
- <R> 2. 要将 P02/SO10/TxD1, P03/SI10/RxD1/SDA10 或 P04/SCK10/SCL10 作为通用端口使用, 注意串行阵列单元 0 的设置。关于更多细节, 参见表 12-7 寄存器设置和引脚之间的关系 (单元 0 的通道 2: CSI10, UART1 发送, IIC10) 和表 12-8 寄存器设置和引脚之间的关系 (单元 0 的通道 3: UART1 接收)。

图 4-2. P00 的框图



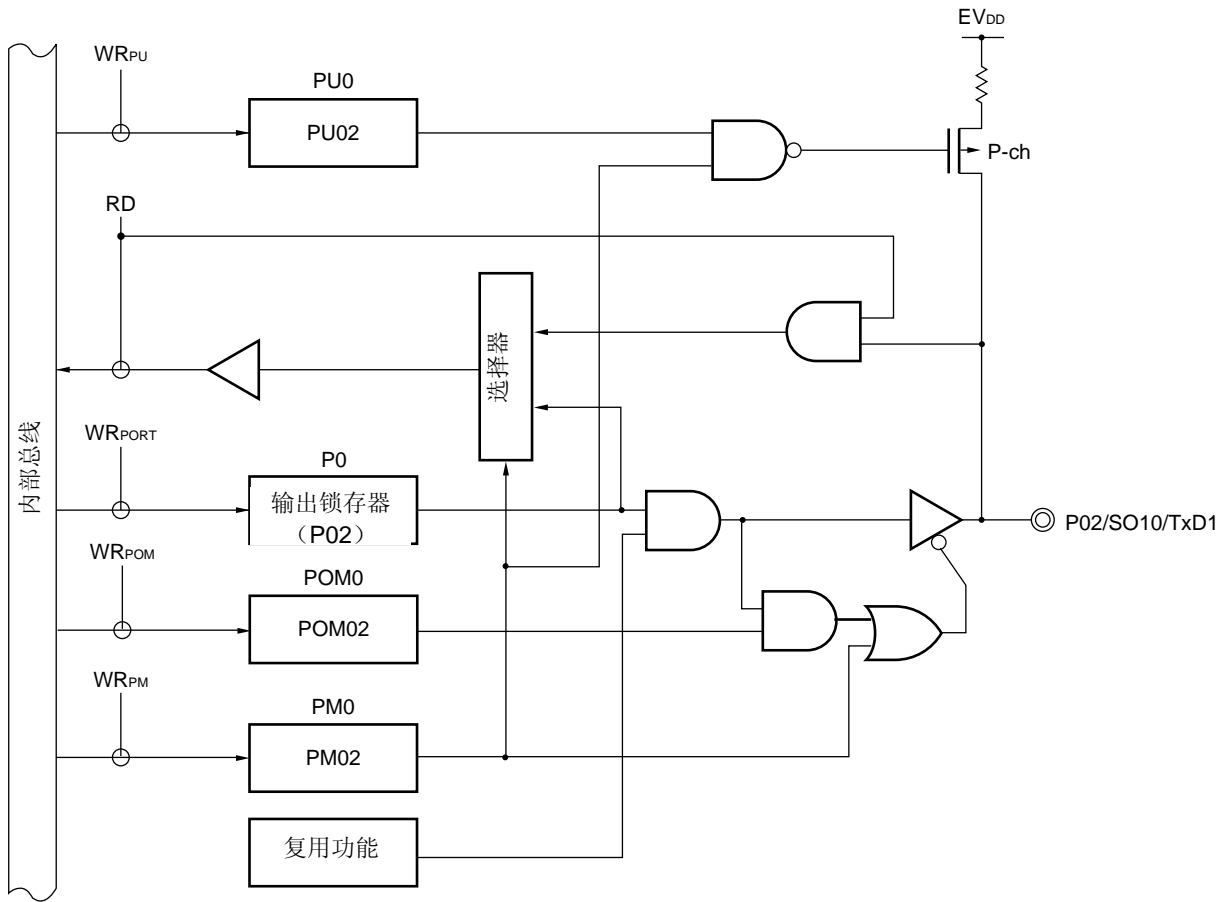
- P0: 端口寄存器 0
- PU0: 上拉电阻选项寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

图 4-3. P01 的框图



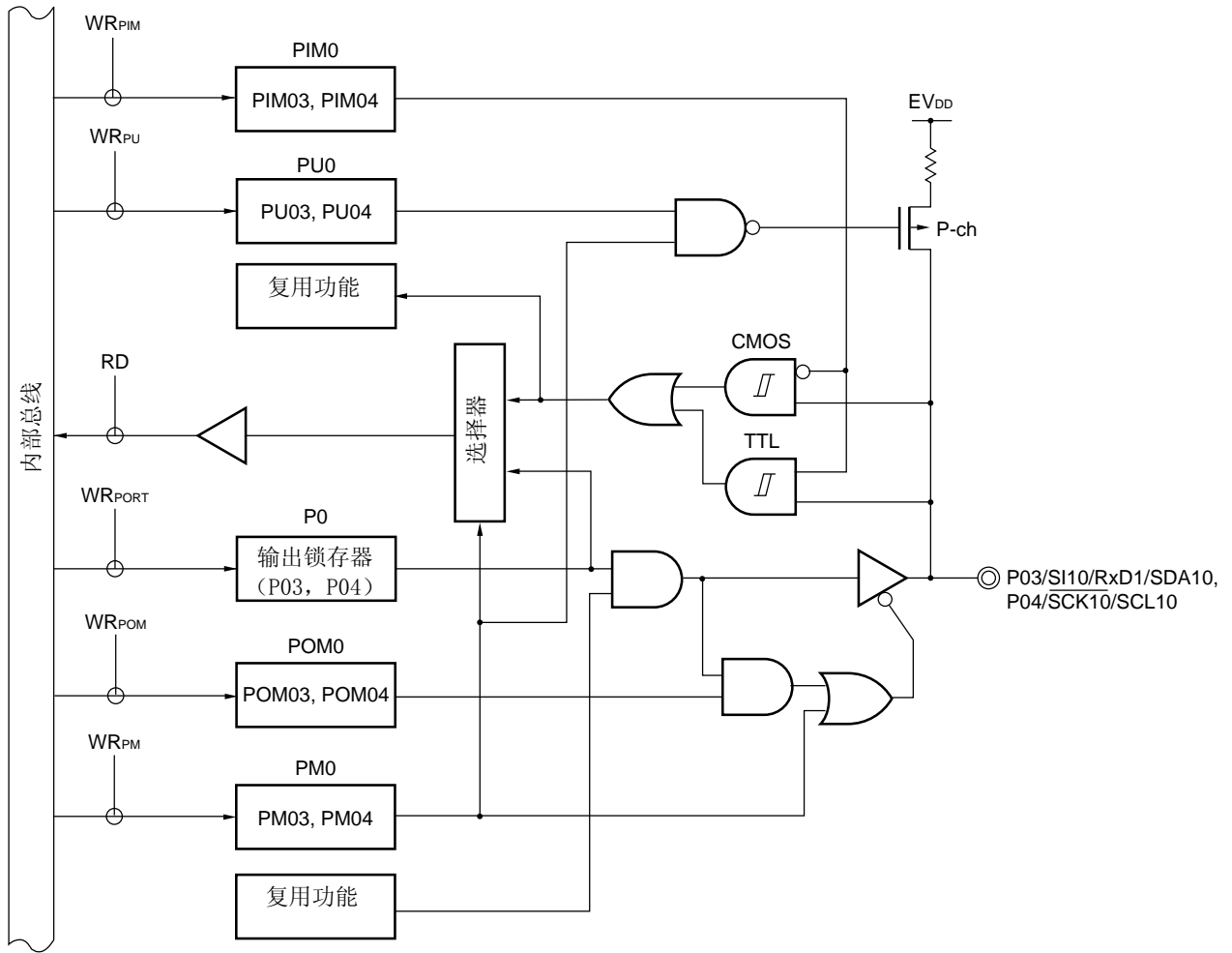
- P0: 端口寄存器 0
- PU0: 上拉电阻选项寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

图 4-4. P02 的框图



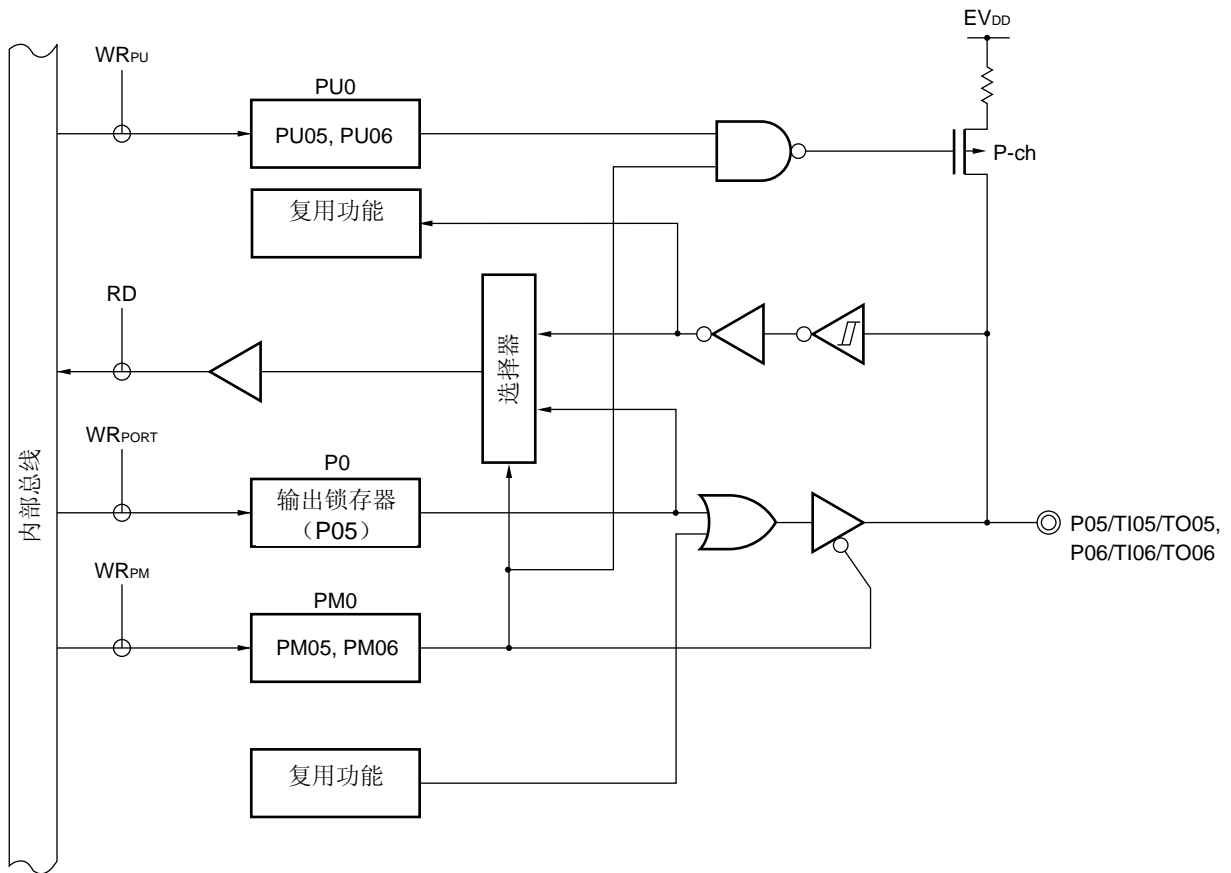
- P0: 端口寄存器 0
- PU0: 上拉电阻选项寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

图 4-5. P03 和 P04 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选项寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

图 4-6. P05 和 P06 的框图



- P0: 端口寄存器 0
- PU0: 上拉电阻选项寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

4.2.2 端口 1

端口 1 是具有输出锁存的 8 位 I/O 端口。通过使用端口模式寄存器 1 (PM1)，可以按位设置端口 1 为输入模式或输出模式。如果 P10 ~ P17 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 1 (PU1) 以 1 位单元的方式指定。

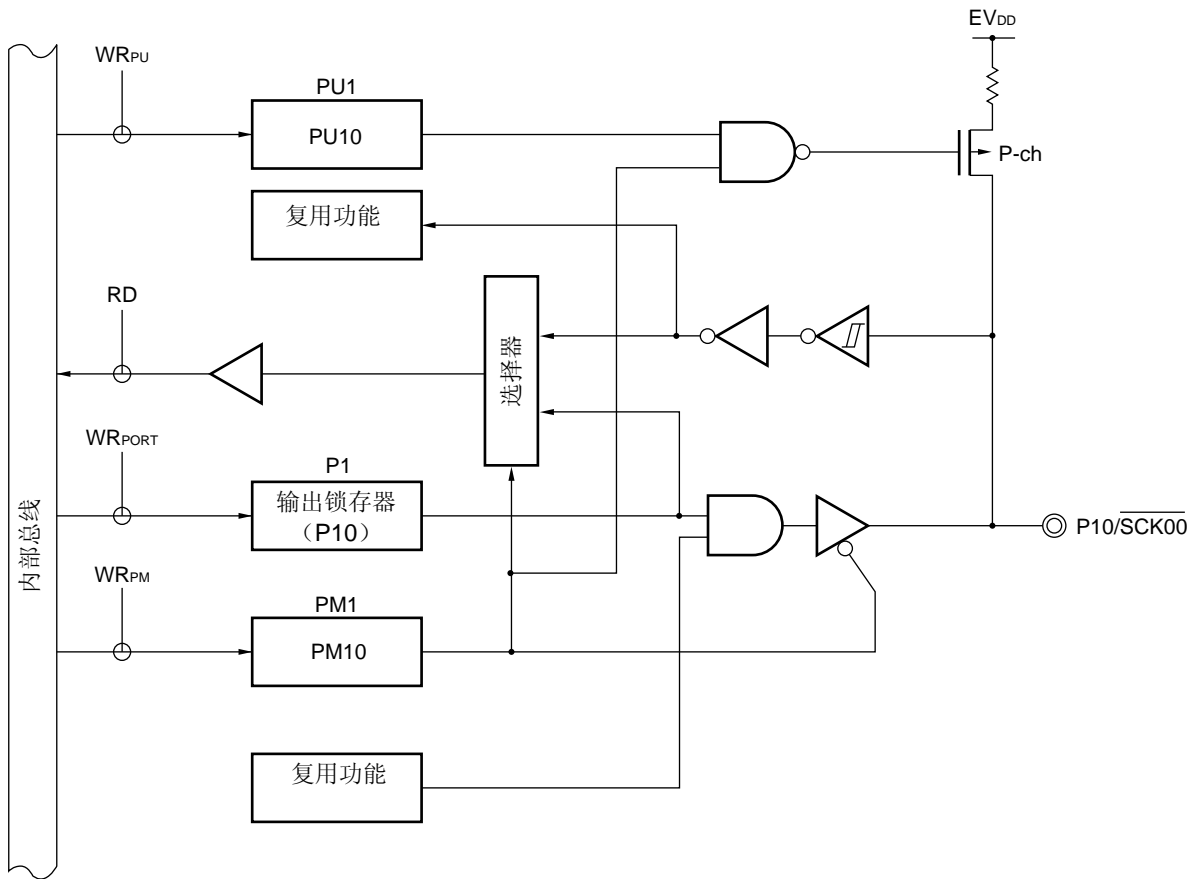
此端口也可用于外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O、和实时计数器时钟输出。

复位信号发生设置端口 1 为输入模式。

图 4-7~4-11 所示为端口 1 的框图。

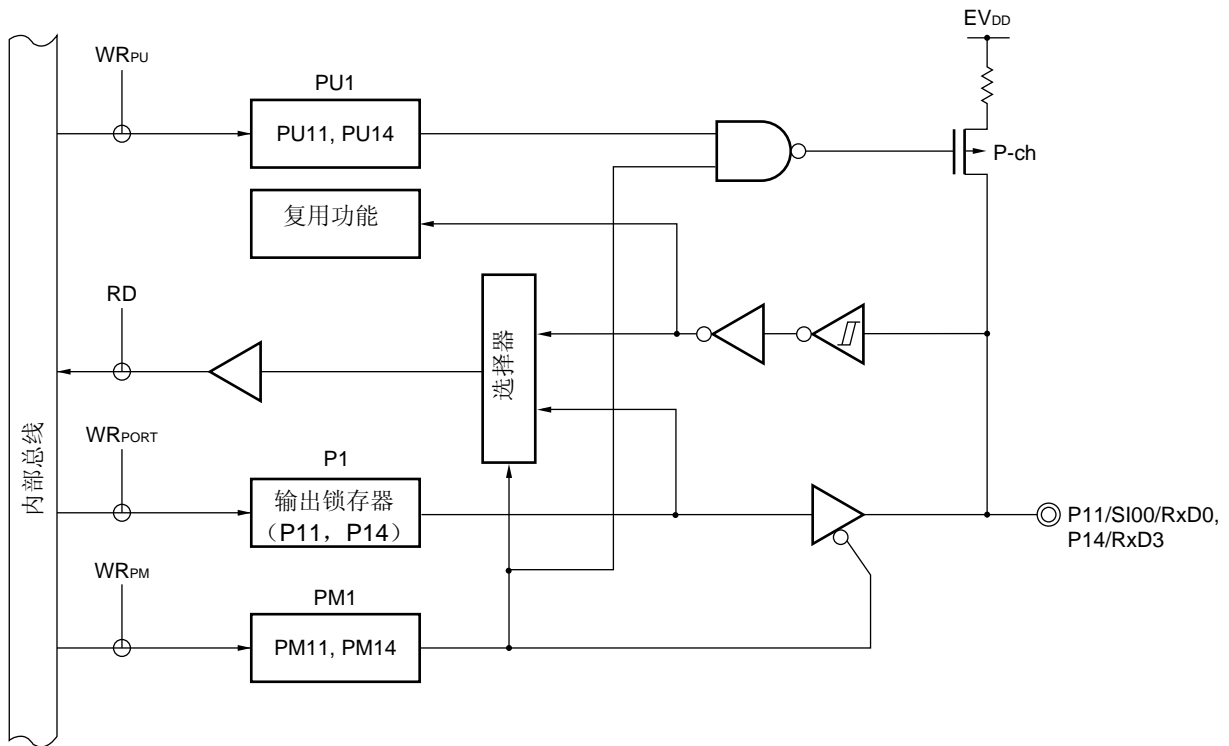
- <R> **注意事项**
1. 要将 P10/SCK00, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3 或 P14/RxD3 作为通用端口使用, 注意串行阵列单元 0 的设置。关于更多细节, 参见表 12-5 寄存器设置和引脚之间的关系 (单元 0 的通道 0: CSI00, UART0 发送) 和表 12-6 寄存器设置和引脚之间的关系 (单元 0 的通道 1: UART0 接收), 表 12-11 寄存器设置和引脚之间的关系 (单元 1 的通道 2: UART3 发送) 和表 12-12 寄存器设置和引脚之间的关系 (单元 1 的通道 3: UART3 接收)
 2. 要将 P16/TI01/TO01/INTP5 或 P17/TI02/TO02 作为通用端口使用, 设置定时器输出寄存器 0 (TO0) 的第 1 位和第 2 位 (TO01, TO02) 和定时器输出允许寄存器 0 (TOE0) 的第 1 位和第 2 位 (TOE00, TOE02) 为“0”, 这也就是这些寄存器的默认值。
 3. 要将 P15/RTCDIV/RTCCL 作为通用端口使用, 设置实时计数器控制寄存器 0 (RTCC0) 的第 4 位 (RCLOE0) 和实时计数器控制寄存器 2 (RTCC2) 的第 6 位 (RCLOE2) 为“0”, 这也就是这些寄存器的默认值。

图 4-7. P10 的框图



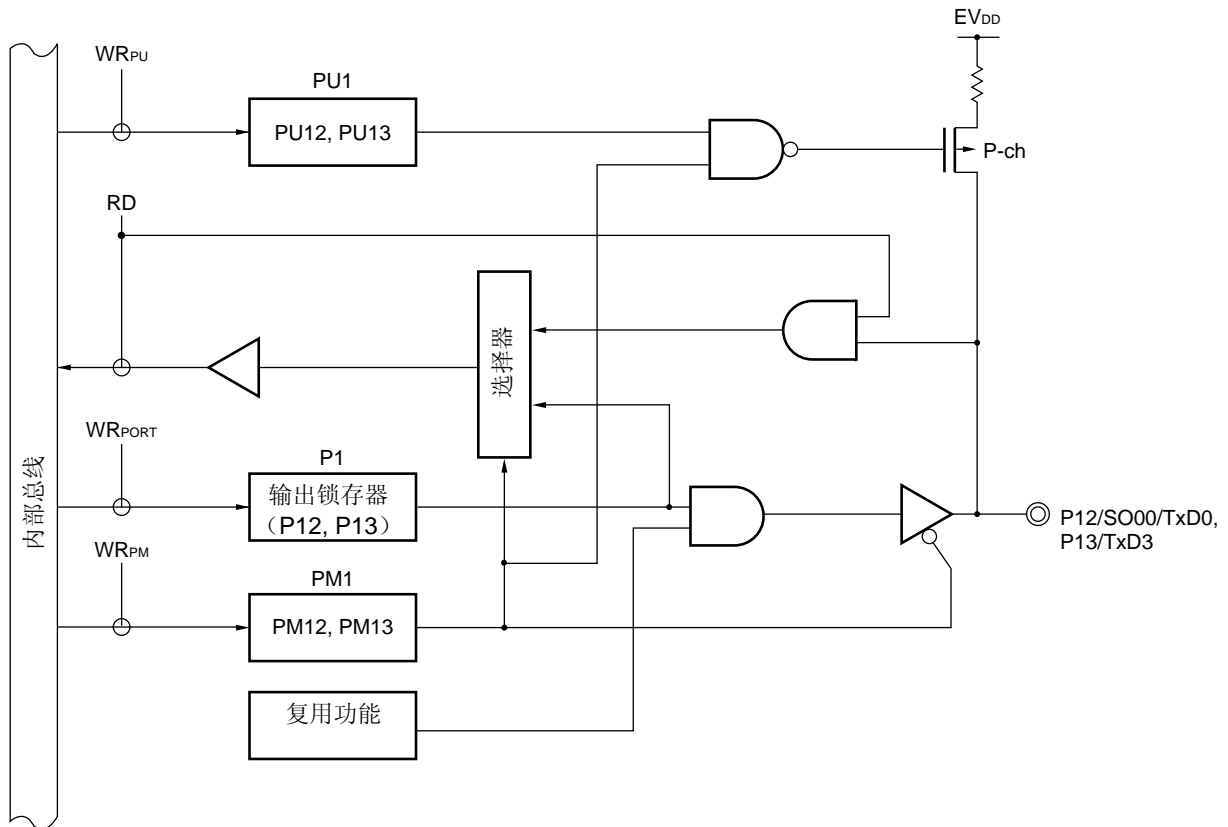
- P1: 端口寄存器 1
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-8. P11 和 P14 的框图



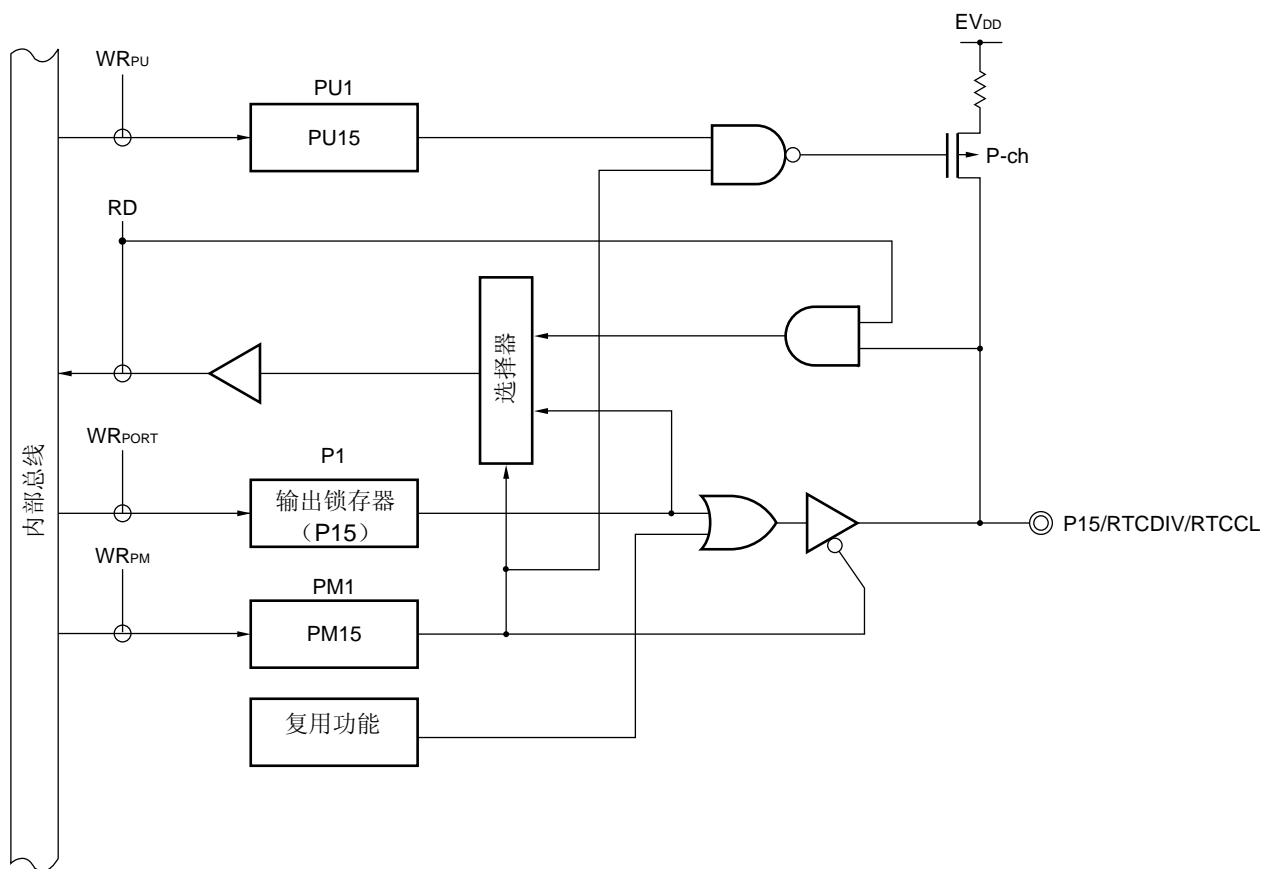
- P1: 端口寄存器 1
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-9. P12 和 P13 的框图



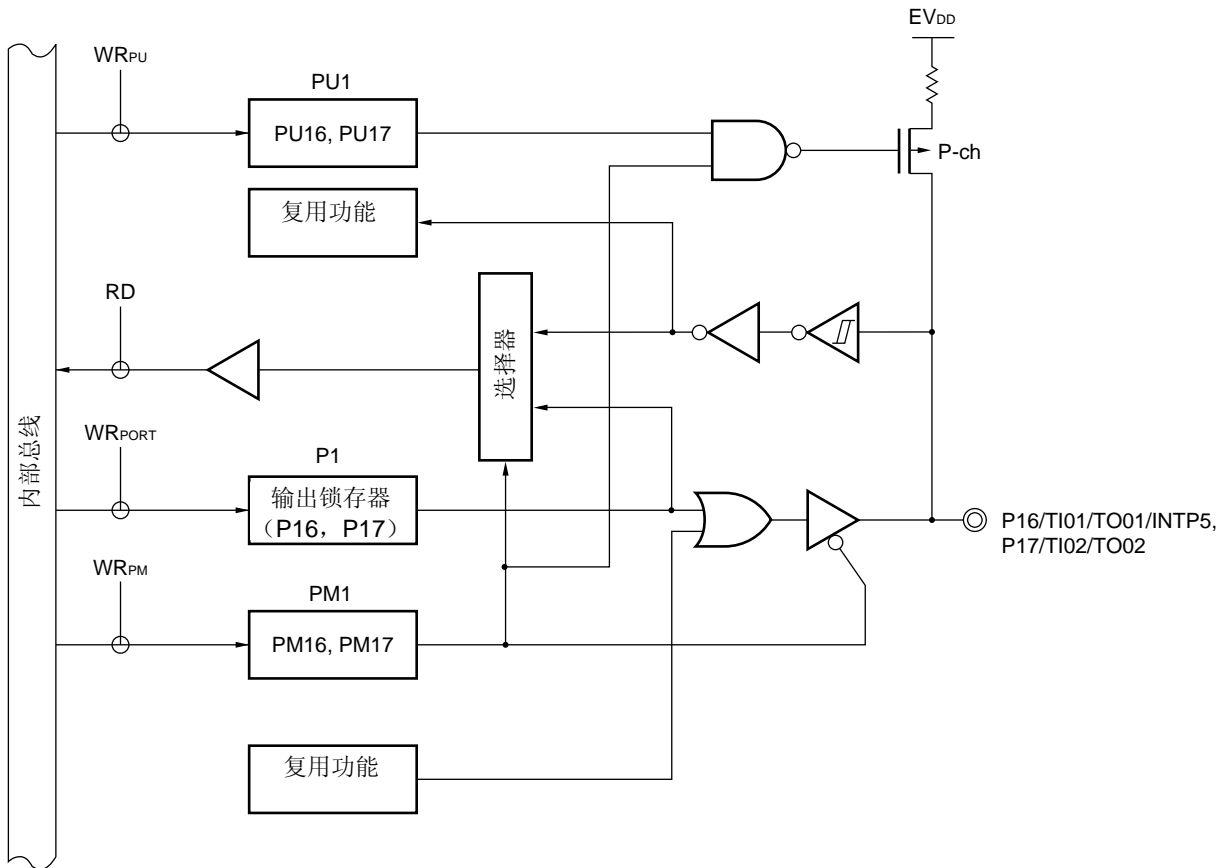
- P1: 端口寄存器 1
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-10. P15 的框图



- P1: 端口寄存器 1
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-11. P16 和 P17 的框图



- P1: 端口寄存器 1
- PU1: 上拉电阻选项寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

4.2.3 端口 2

端口 2 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 2 (PM2)，可以按位设置端口 2 为输入模式或输出模式。

此端口也可用于 A/D 转换器 模拟输入。

如果要将 P20/AN10 ~ P27/AN17 作为数字输入引脚，应通过使用 A/D 端口配置寄存器 (ADPC) 将这些引脚设置为数字 I/O 模式，并通过使用 PM2 设置为输入模式。从低位端开始使用这些引脚。

如果要将 P20/AN10 ~ P27/AN17 作为数字输出引脚，应通过使用 A/D 端口配置寄存器 (ADPC) 将这些引脚设置为数字 I/O 模式，并通过使用 PM2 设置为输出模式。

如果要将 P20/AN10~P27/AN17 作为模拟输入引脚，通过 A/D 端口配置寄存器 (ADPC) 和使用 PM2 在输入模式下设置为模拟输入模式。从高位端开始使用这些引脚。

表 4-4. P20/AN10~P27/AN17 引脚的设置功能

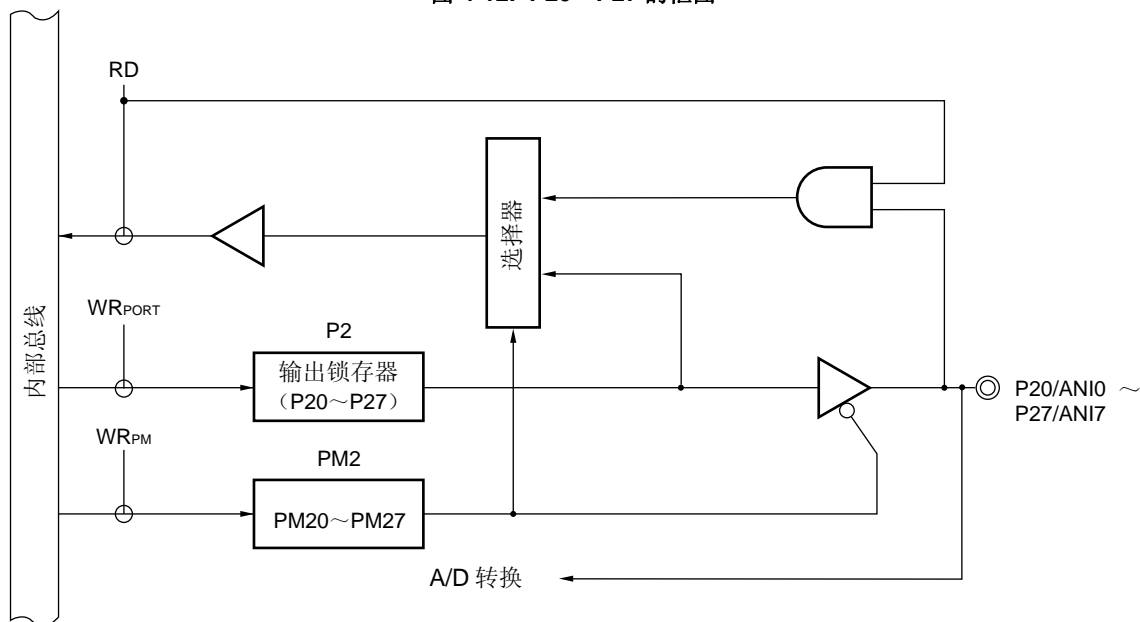
ADPC	PM2	ADS	P20/AN10~P27/AN17 引脚
数字 I/O 选择	输入模式	-	数字输入
	输出模式	-	数字输出
模拟输入 选择	输入模式	选择 ANI.	模拟输入 (进行转换)
		不要选择 ANI.	模拟输入 (不进行转换)
	输出模式	选择 ANI.	设置禁止
		不要选择 ANI.	

当复位信号发生时，所有 P20/AN10~P27/AN17 均设置为数字输入模式。

图 4-12 所示为端口 2 的框图。

注意事项 当端口 2 作为数字端口时，AVREF0 引脚的电压要与 VDD 引脚相同。

图 4-12. P20~P27 的框图



P2: 端口寄存器 2
 PM2: 端口模式寄存器 2
 RD: 读信号
 WR_{xx}: 写信号

4.2.4 端口 3

端口 3 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 3 (PM3)，可以按位设置端口 3 为输入模式或输出模式。如果 P30 和 P31 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 3 (PU3) 以 1 位单元的方式指定。

此端口也可用于外部中断请求输入、定时器 I/O、和实时计数器修正时钟输出。

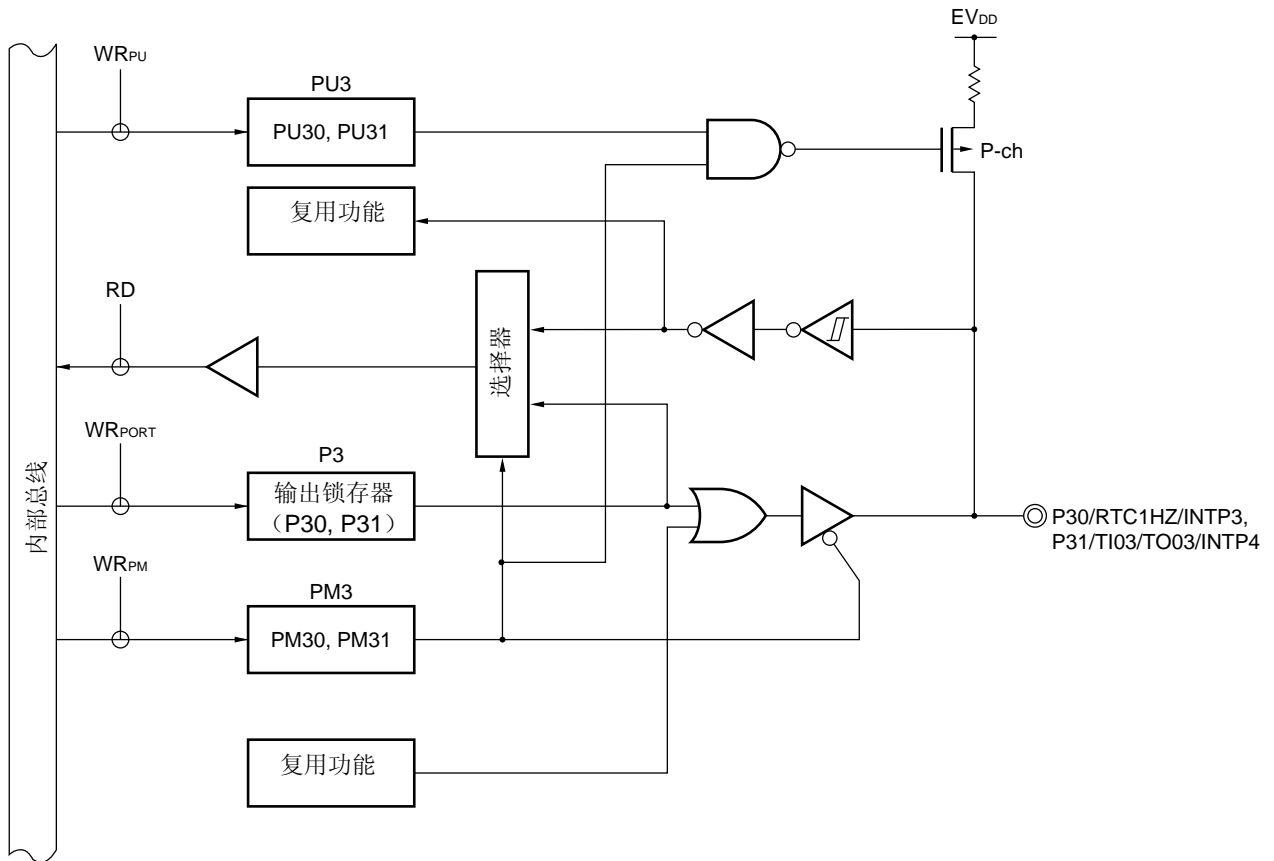
复位信号的产生将会设置端口 3 为输入模式。

图 4-13 所示为端口 3 的框图。

<R> 注意事项 1. 要将 P31/TI03/TO03/INTP4 作为通用端口使用，设置定时器输出寄存器 0 (TO0) 的第 3 位 (TO03) 和定时器输出允许寄存器 0 (TOE0) 的第 3 位 (TOE03) 为“0”，这也就是这些寄存器的默认值。

<R> 2. 要将 P30/RTC1HZ/INTP3 作为通用端口使用，设置实时计数器控制寄存器 0 (RTCC0) 的第 5 位 (RCLOE1) 为“0”，这也就是该寄存器的默认值。

图 4-13. P30 和 P31 的框图



- P3: 端口寄存器 3
- PU3: 上拉电阻选项寄存器 3
- PM3: 端口模式寄存器 3
- RD: 读信号
- WR_{xx}: 写信号

4.2.5 端口 4

端口 4 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 4 (PM4)，可以按位设置端口 4 为输入模式或输出模式。如果 P40 ~ P47 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 4 (PU4) 以 1 位单元的方式指定。

通过使用端口输入模式寄存器 4 (PIM4) 以 1 位单元的方式可指定 P43 和 P44 引脚通过普通缓冲器或 TTL 缓冲器输入。

通过端口输出模式寄存器 4 (POM4) 以 1 位为单元的方式可指定 P43~P45 引脚的输出为 N-ch 漏极开路输出 (V_{DD} 耐压)。

此端口也可用于串行接口数据 I/O、时钟 I/O、flash 存储器编程器/调试器数据 I/O、时钟输出、和定时器 I/O。

复位信号发生设置端口 4 为输入模式。

图 4-14~4-20 所示为端口 4 的框图。

注 连接工具时，P40 和 P41 引脚不能连接上拉电阻。

注意事项 1. 连接工具时，P40 引脚不能作为端口引脚使用。

使用片上调试功能时，通过调试器上的模式设置 P41 引脚可作如下用途。

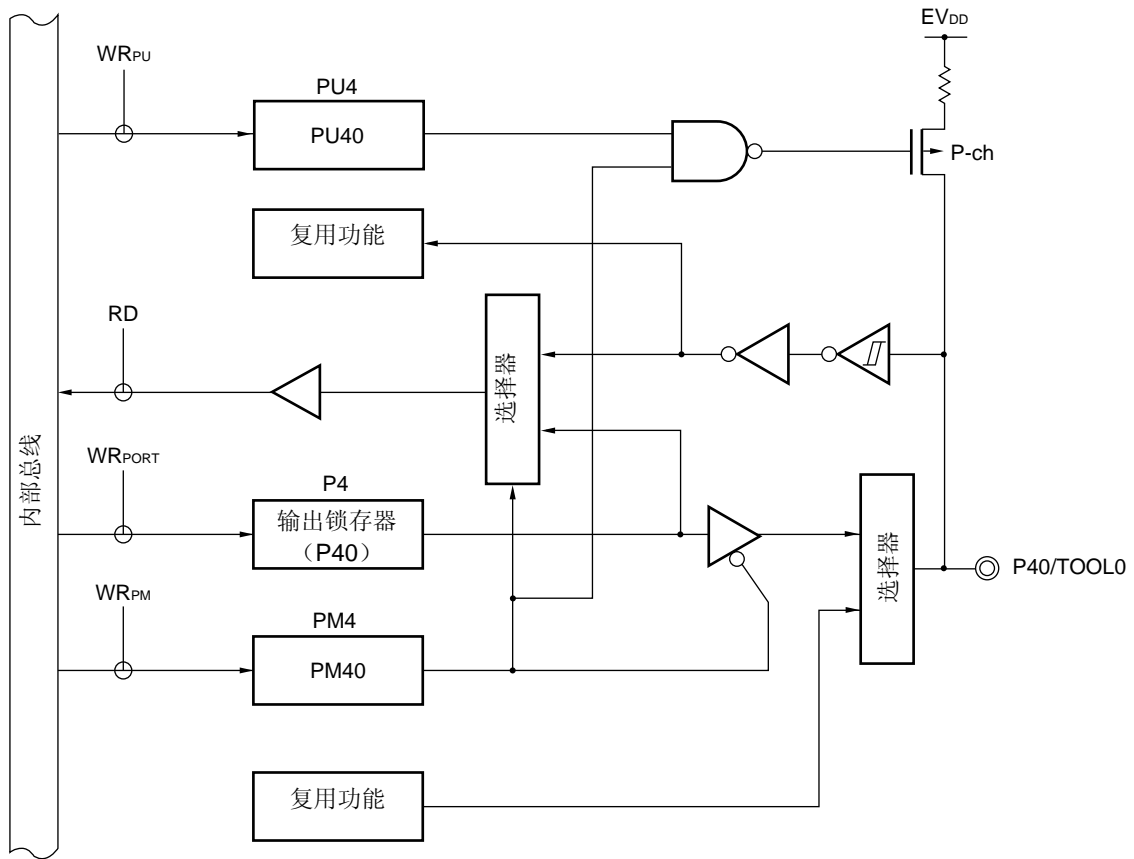
1 线模式：可用作端口 (P41)。

2 线模式：用作 $\overline{\text{TOOL1}}$ 引脚并且不能用作端口 (P41)。

<R> 2. 要将 P43/ $\overline{\text{SCK01}}$ ，P44/SI01 或 P45/SO011 用作通用端口，注意串行阵列单元 0 的设置。关于更多细节，参见表 12-5 寄存器设置和引脚之间的关系 (单元 0 的通道 0: CSI01, UART0 发送)。

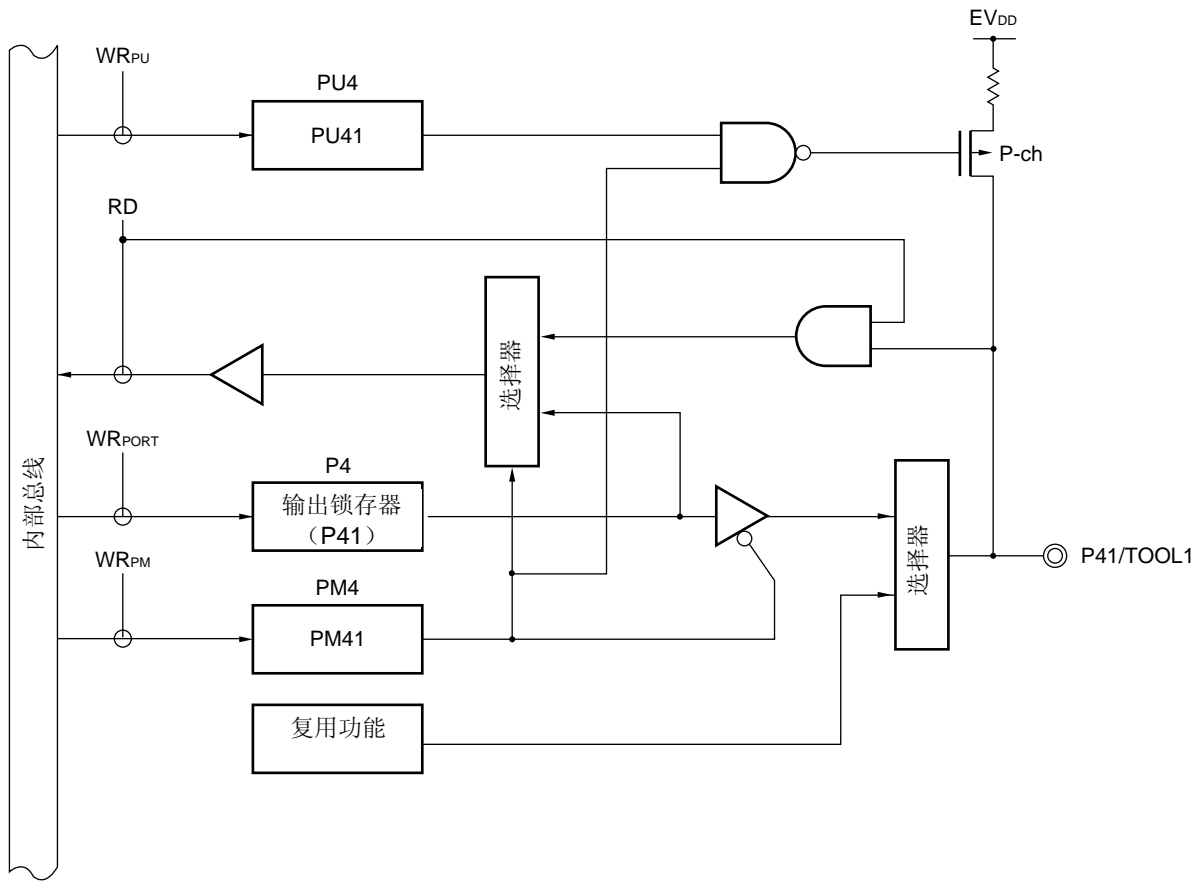
<R> 3. 要将 P42/TI04/TO04 作为通用端口使用，将定时器输出寄存器 0 (TO0) 的第 4 位 (TO04) 和定时器输出允许寄存器 0 (TOE0) 的第 4 位 (TOE04) 设置为“0”，这也就是这些寄存器的默认值。

图 4-14. P40 的框图



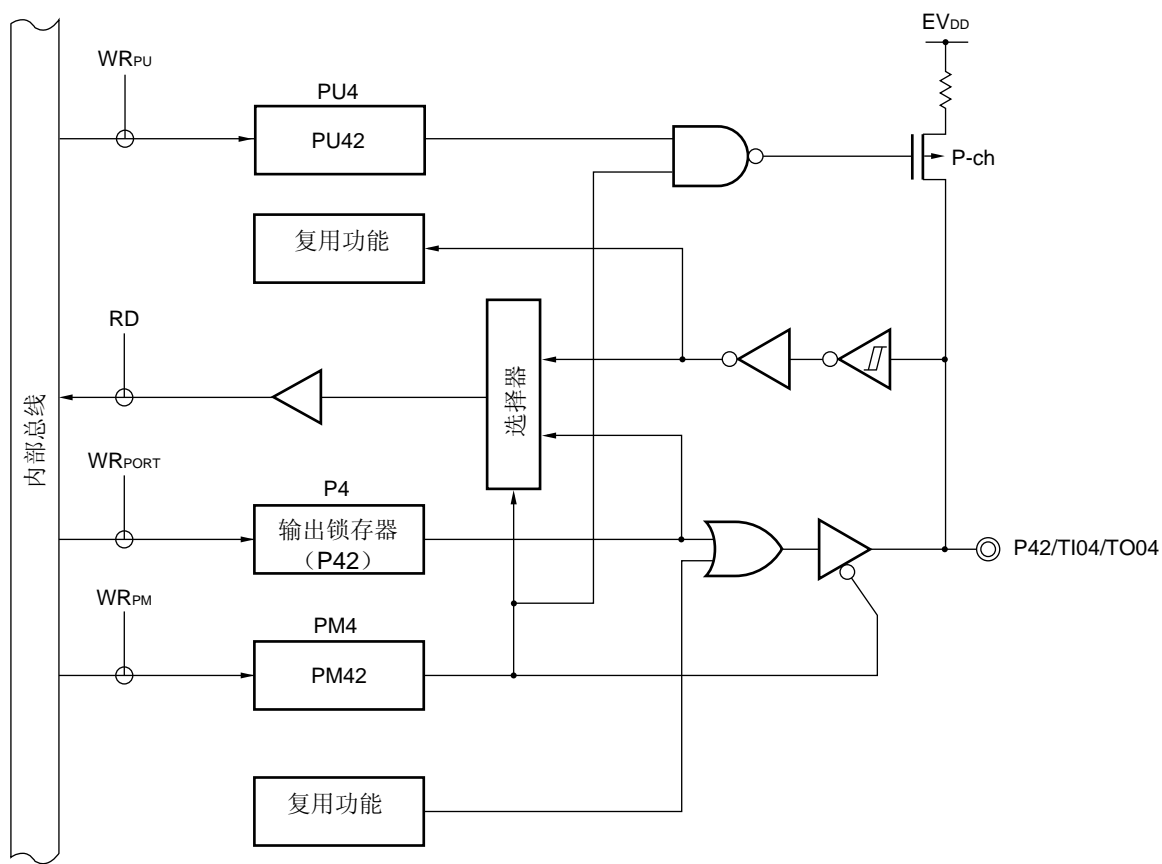
- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-15. P41 的框图



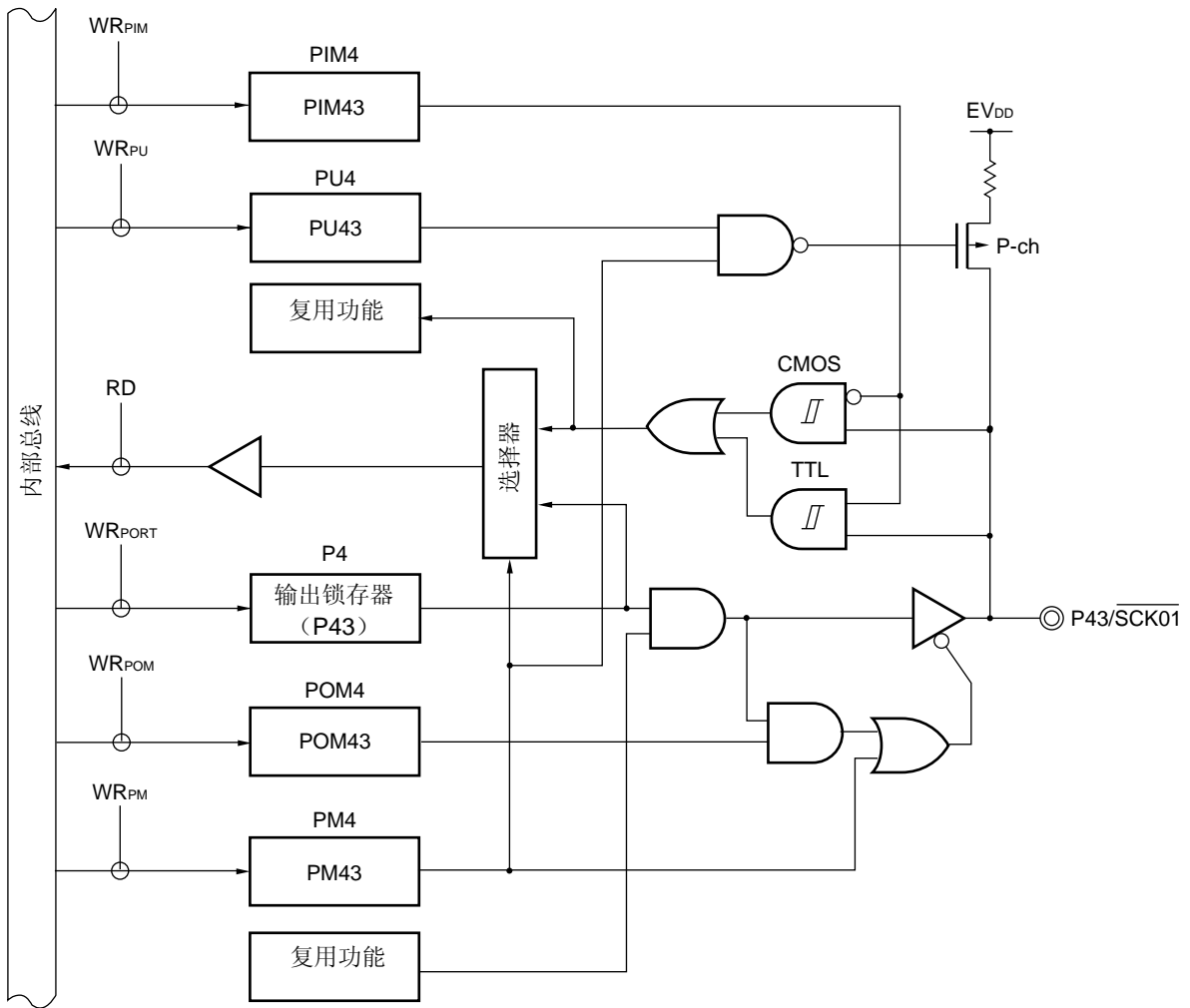
- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-16. P42 的框图



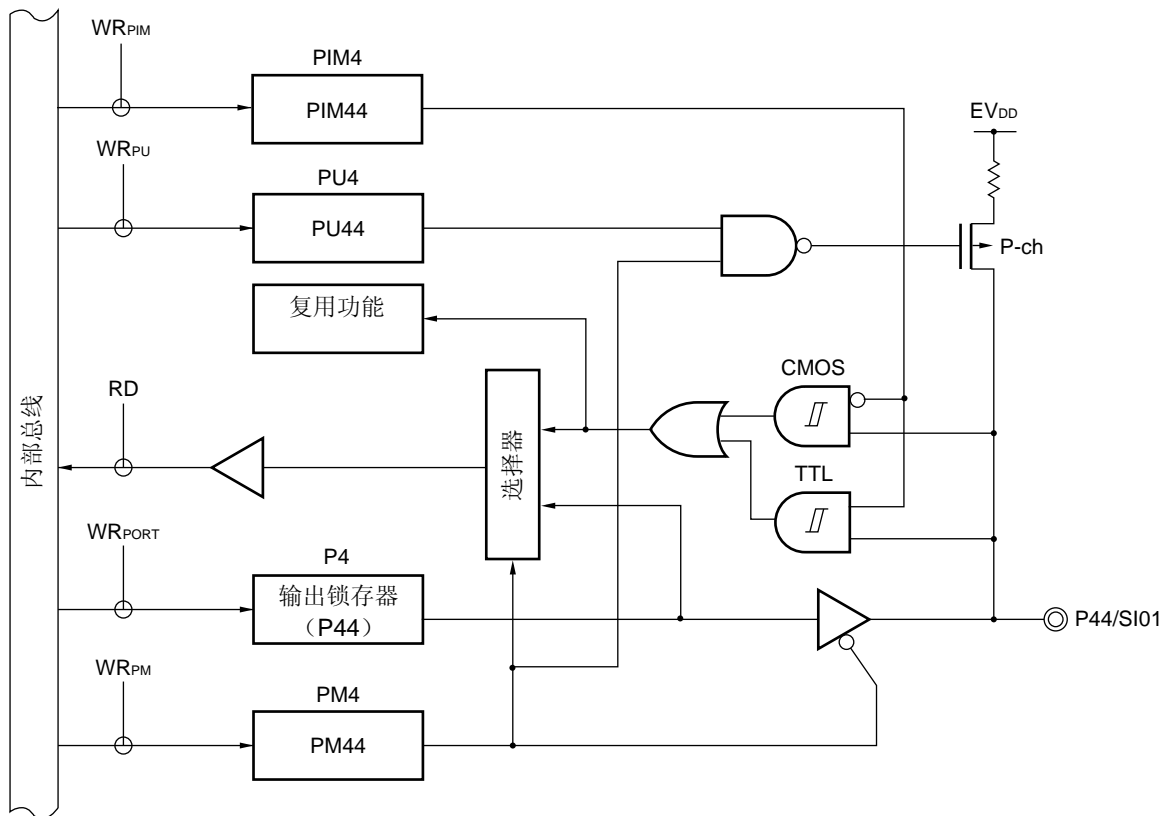
- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-17. P43 的框图



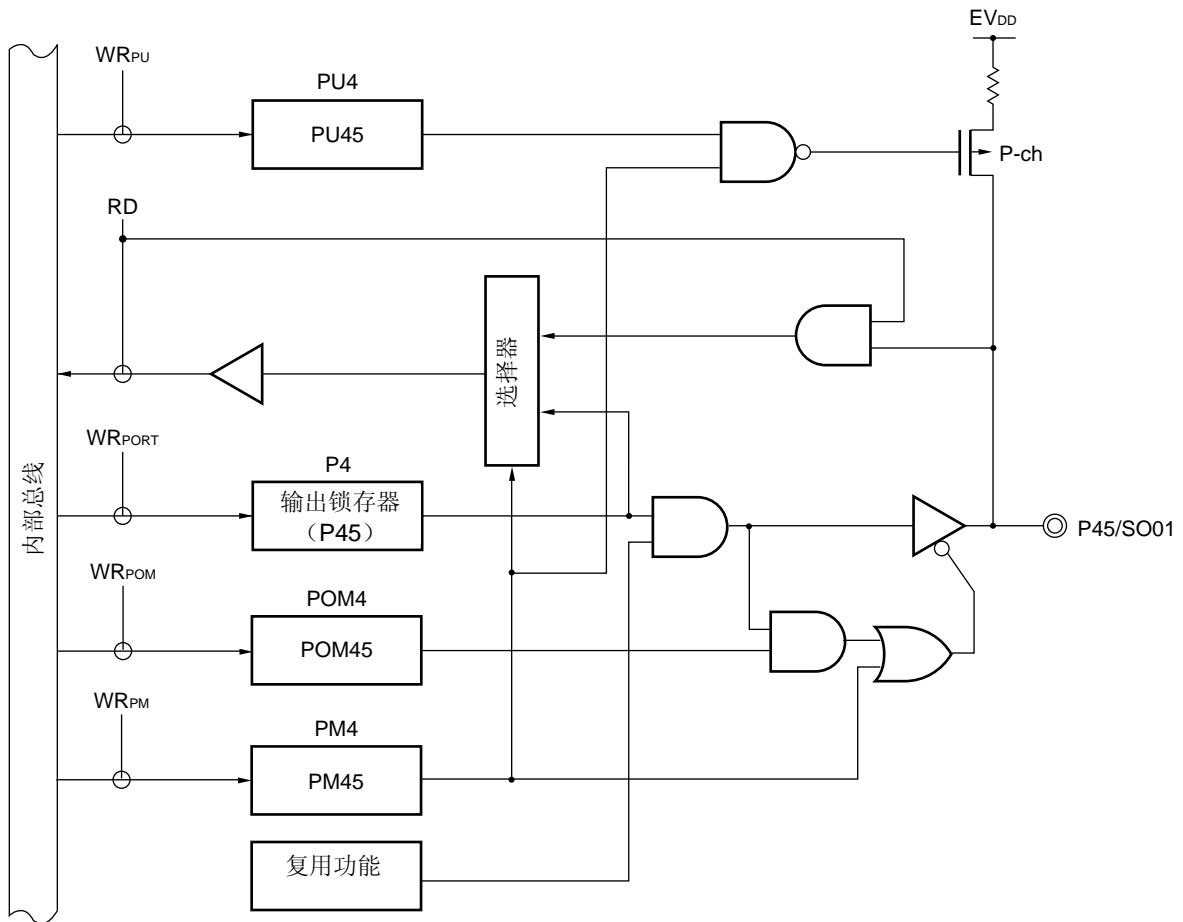
- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-18. P44 的框图



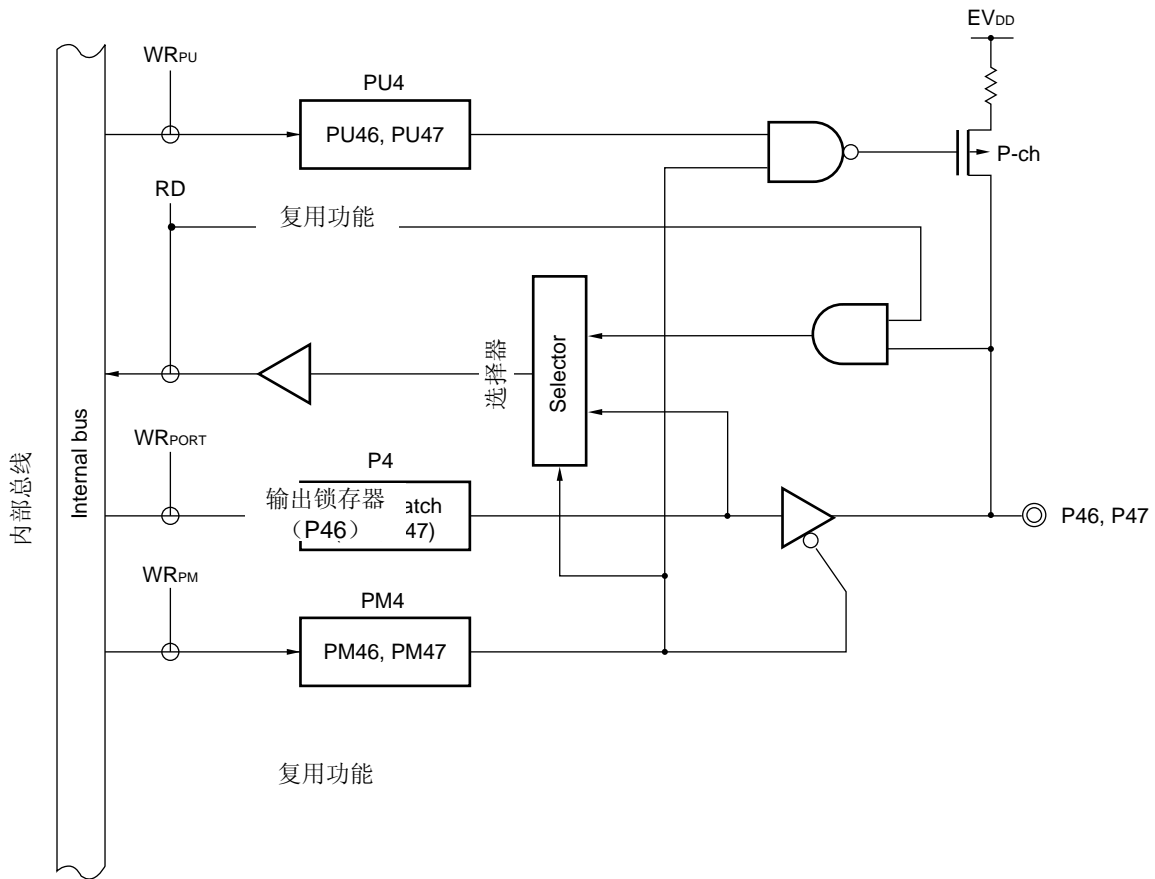
- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-19. P45 的框图



- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

图 4-20. P46 和 P47 的框图



- P4: 端口寄存器 4
- PU4: 上拉电阻选项寄存器 4
- PM4: 端口模式寄存器 4
- RD: 读信号
- WR_{xx}: 写信号

4.2.6 端口 5

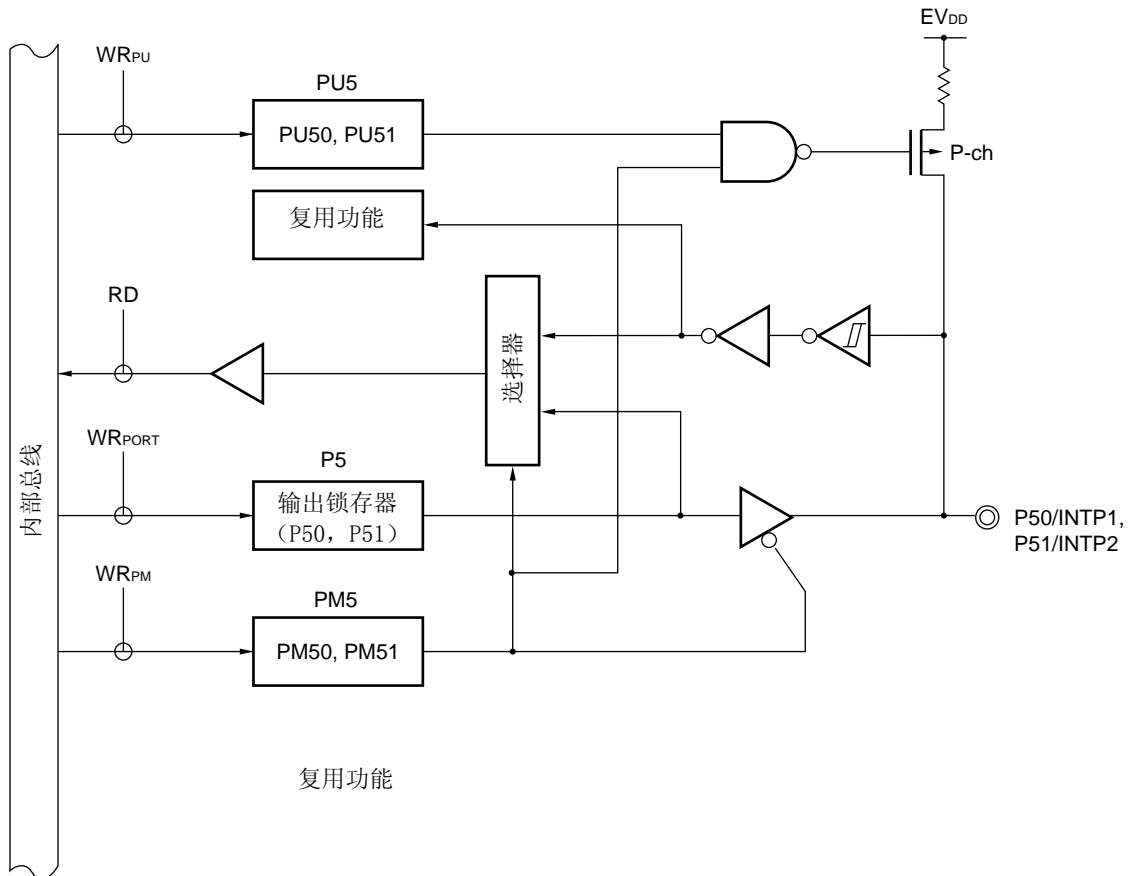
端口 5 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 5 (PM5)，可以按位设置端口 5 为输入模式或输出模式。当 P50 ~ P55 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 5 (PU5) 以 1 位单元的方式指定。

此端口也可用于外部中断请求输入。

复位信号发生设置端口 5 为输入模式。

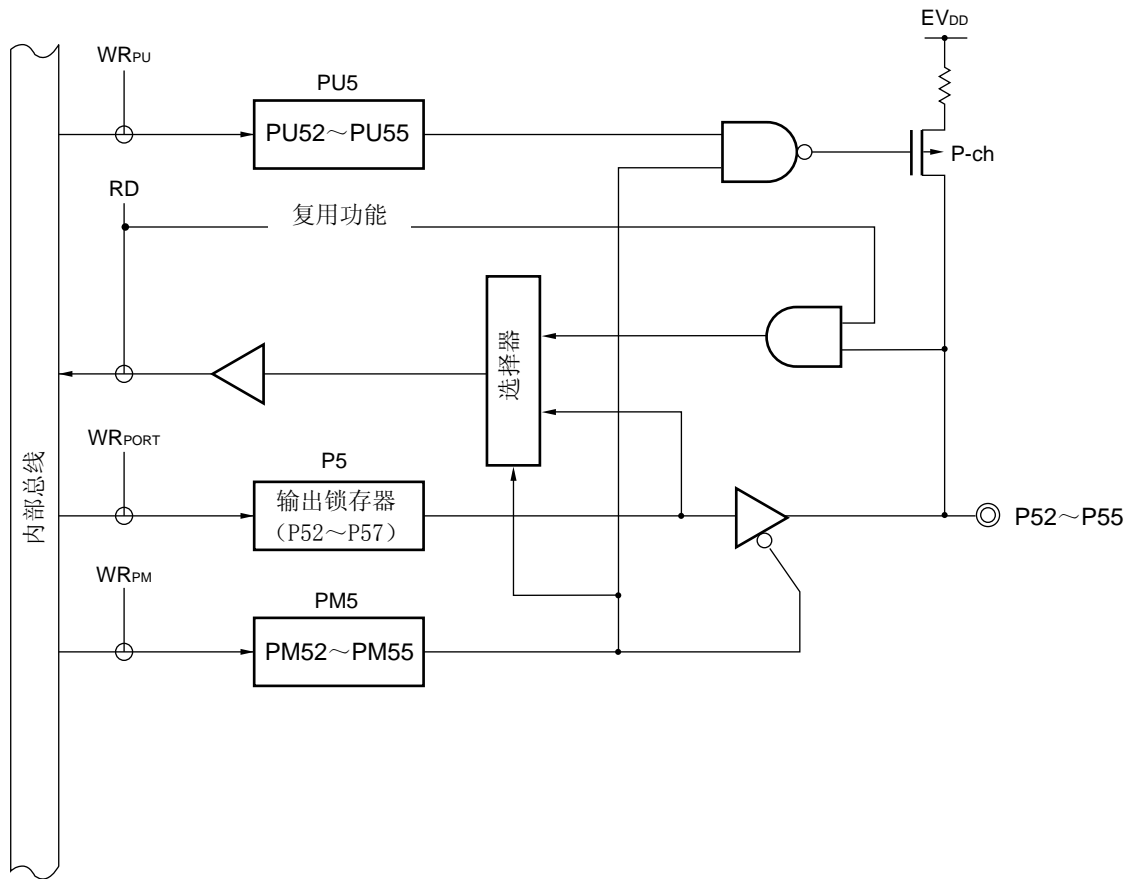
图 4-21 和 4-22 所示为端口 5 的框图。

图 4-21. P50 和 P51 的框图



- P5: 端口寄存器 5
- PU5: 上拉电阻选项寄存器 5
- PM5: 端口模式寄存器 5
- RD: 读信号
- WR_{xx}: 写信号
- EXEN: 存储器扩展模式寄存器 (MEM) 的第 7 位

图 4-22. P52~P55 的框图



- P5: 端口寄存器 5
- PU5: 上拉电阻选项寄存器 5
- PM5: 端口模式寄存器 5
- RD: 读信号
- WR_{xx}: 写信号
- EXEN: 存储器扩展模式寄存器 (MEM) 的第 7 位

4.2.7 端口 6

端口 6 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 6 (PM6)，可以按位设置端口 6 为输入模式或输出模式。当 P64~P67 引脚作为输入端口使用时，可由上拉电阻选项寄存器 6 (PU6) 按位设定片内上拉电阻的使用。

P60~P63 引脚的输出是 N-ch 漏极开路输出 (6 V 耐压)。

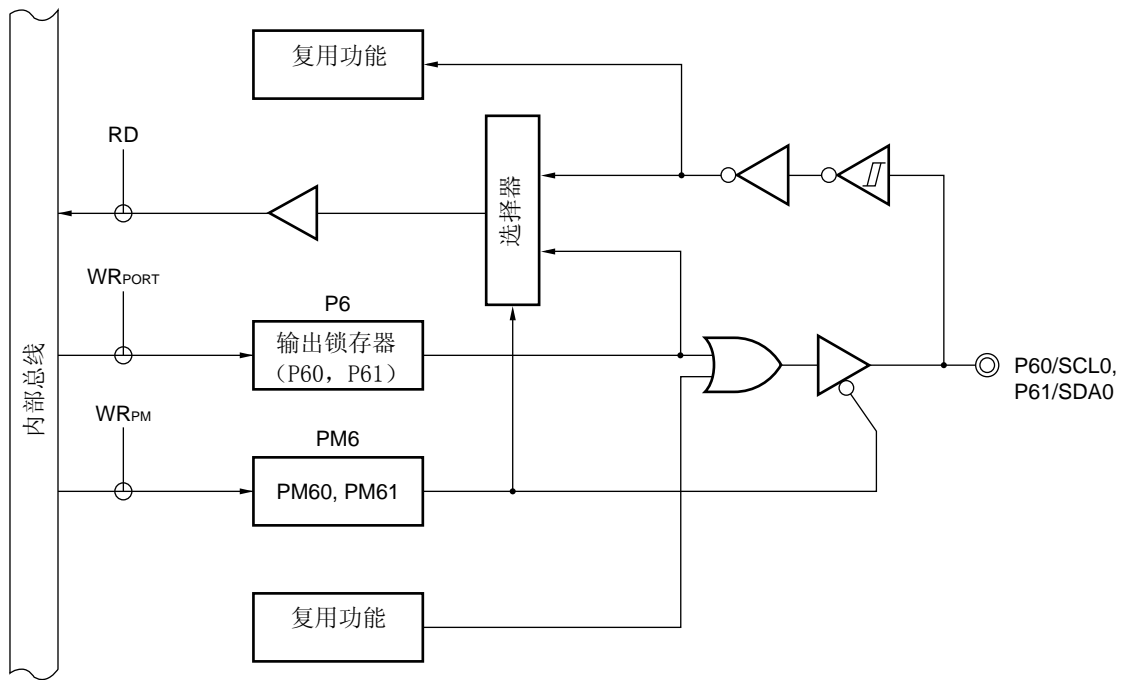
此端口也可用于串行接口数据 I/O 和时钟 I/O。

复位信号发生设置端口 6 为输入模式。

图 4-23~4-25 所示为端口 6 的框图。

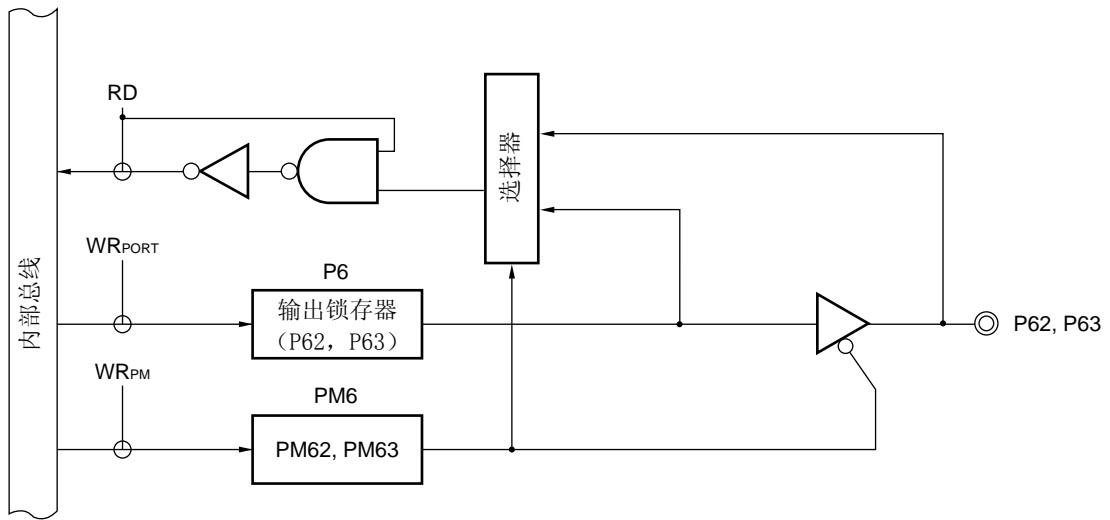
<R> **注意事项** 当 P60/SCL0 or P61/SDA0 用作通用端口时，停止串行接口 IIC0 的操作。

图 4-23. P60 和 P61 的框图



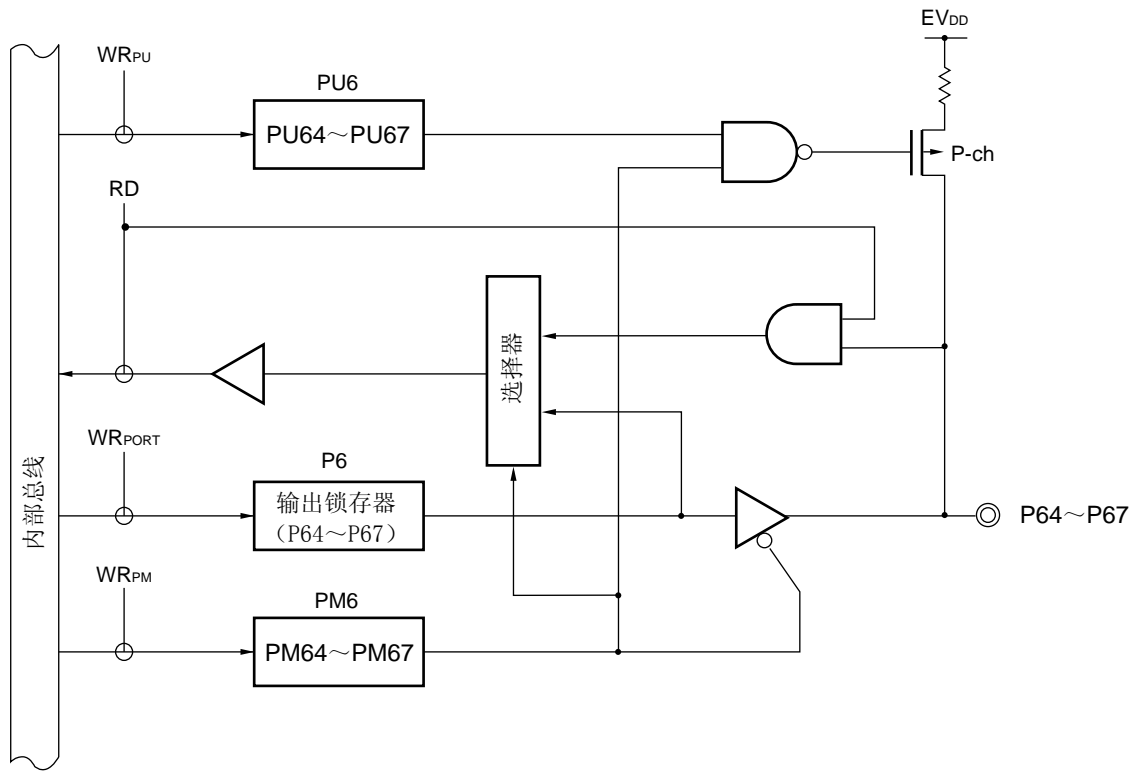
- P6: 端口寄存器 6
- PM6: 端口模式寄存器 6
- RD: 读信号
- WR_{xx}: 写信号

图 4-24. P62 和 P63 的框图



- P6: 端口寄存器 6
- PM6: 端口模式寄存器 6
- RD: 读信号
- WR_{xx}: 写信号

图 4-25. P64~P67 的框图



- P6: 端口寄存器 6
- PU6: 上拉电阻选项寄存器 6
- PM6: 端口模式寄存器 6
- RD: 读信号
- WR_{xx}: 写信号

4.2.8 端口 7

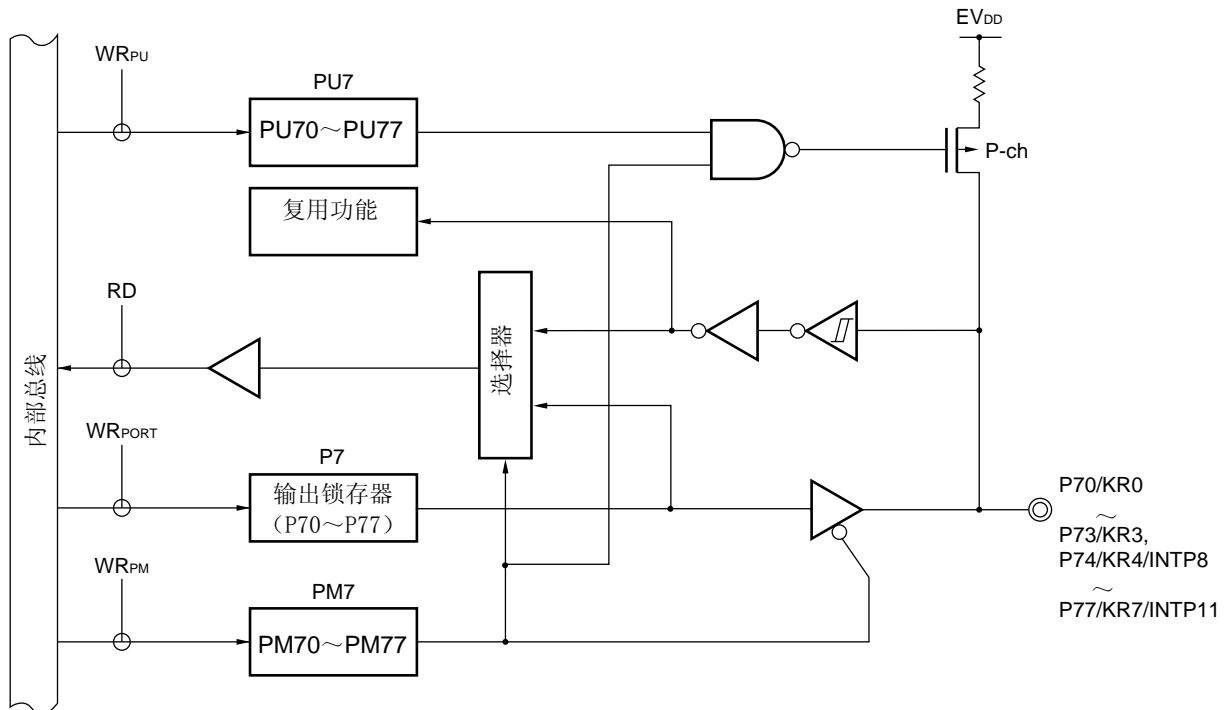
端口 7 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 7 (PM7)，可以按位设置端口 7 为输入模式或输出模式。作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 7 (PU7) 以 1 位单元的方式指定。

此端口也可用作按键返回输入、和中断请求输入。

复位信号发生设置端口 7 为输入模式。

图 4-26 所示为端口 7 的框图。

图 4-26. P70~P77 的框图



- P7: 端口寄存器 7
- PU7: 上拉电阻选项寄存器 7
- PM7: 端口模式寄存器 7
- RD: 读信号
- WR_{xx}: 写信号

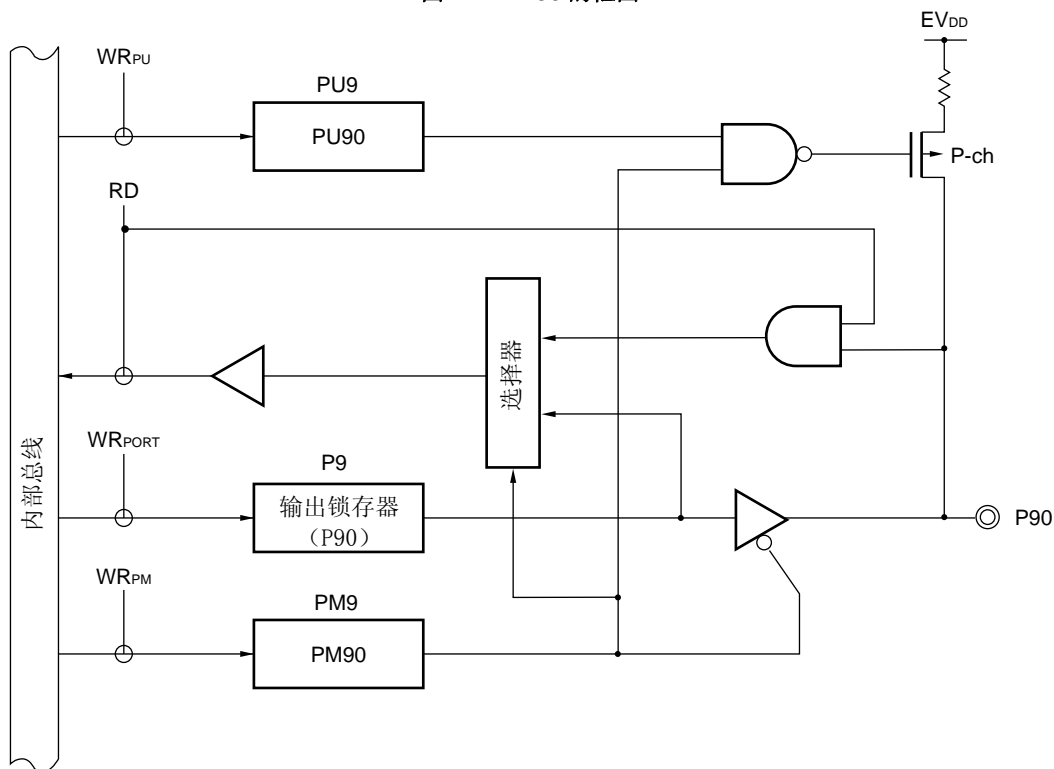
4.2.9 端口 9

端口 9 是具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 9 (PM9)，可以设置端口 9 为输入模式或输出模式。P90 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 9 (PU9) 以 1 位单元的方式指定。

复位信号发生设置端口 9 为输入模式。

图 4-27 所示为端口 9 的框图。

图 4-27. P90 的框图



- P9: 端口寄存器 9
- PU9: 上拉电阻选项寄存器 9
- PM9: 端口模式寄存器 9
- RD: 读信号
- WR_{xx}: 写信号

4.2.10 端口 11

端口 11 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 11 (PM11)，可以按位设置端口 11 为输入模式或输出模式。

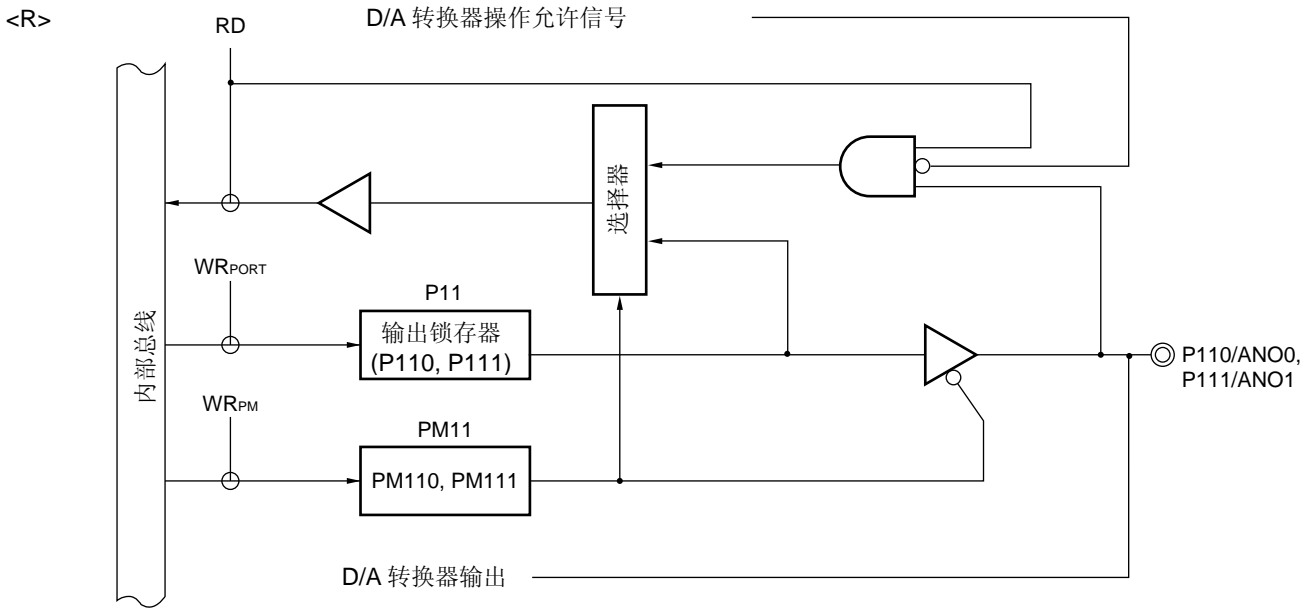
此端口也可用作 D/A 转换器模拟输出。

复位信号发生设置端口 11 为输入模式。

图 4-28 所示为端口 11 的框图。

注意事项 当端口 11 用作数字端口时，确保 AVREF1 引脚与 VDD 引脚电平相同。

图 4-28. P110 和 P111 的框图



- P11: 端口寄存器 11
- PM11: 端口模式寄存器 11
- RD: 读信号
- WR_{xx}: 写信号

4.2.11 端口 12

P120 是具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 12 (PM12)，可以设置端口 12 为输入模式或输出模式。当用作输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 12 (PU12) 指定。

P121~P124 是 4 位输入端口。

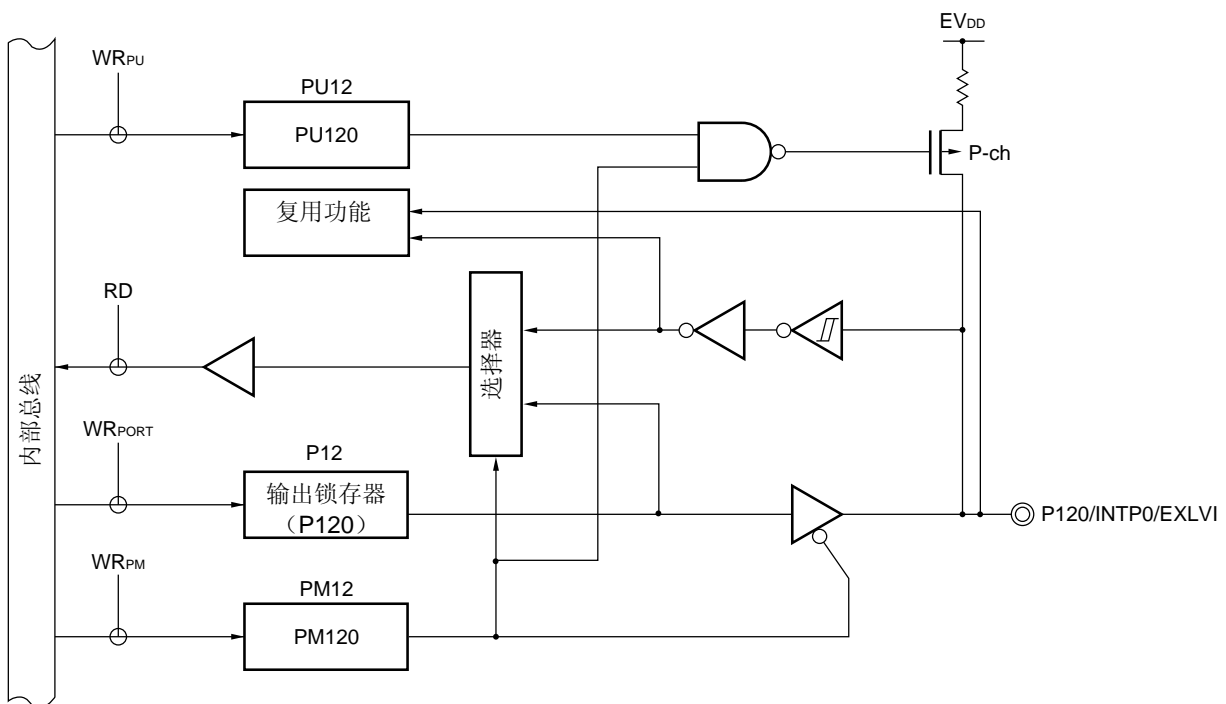
此端口也可用作外部中断请求输入、外部低电压检测的电压输入、为主系统时钟连接振荡器、为副系统时钟连接振荡器、和主系统时钟的外部时钟输入。

复位信号发生设置端口 12 为输入模式。

图 4-29~图 4-31 所示为端口 12 的框图。

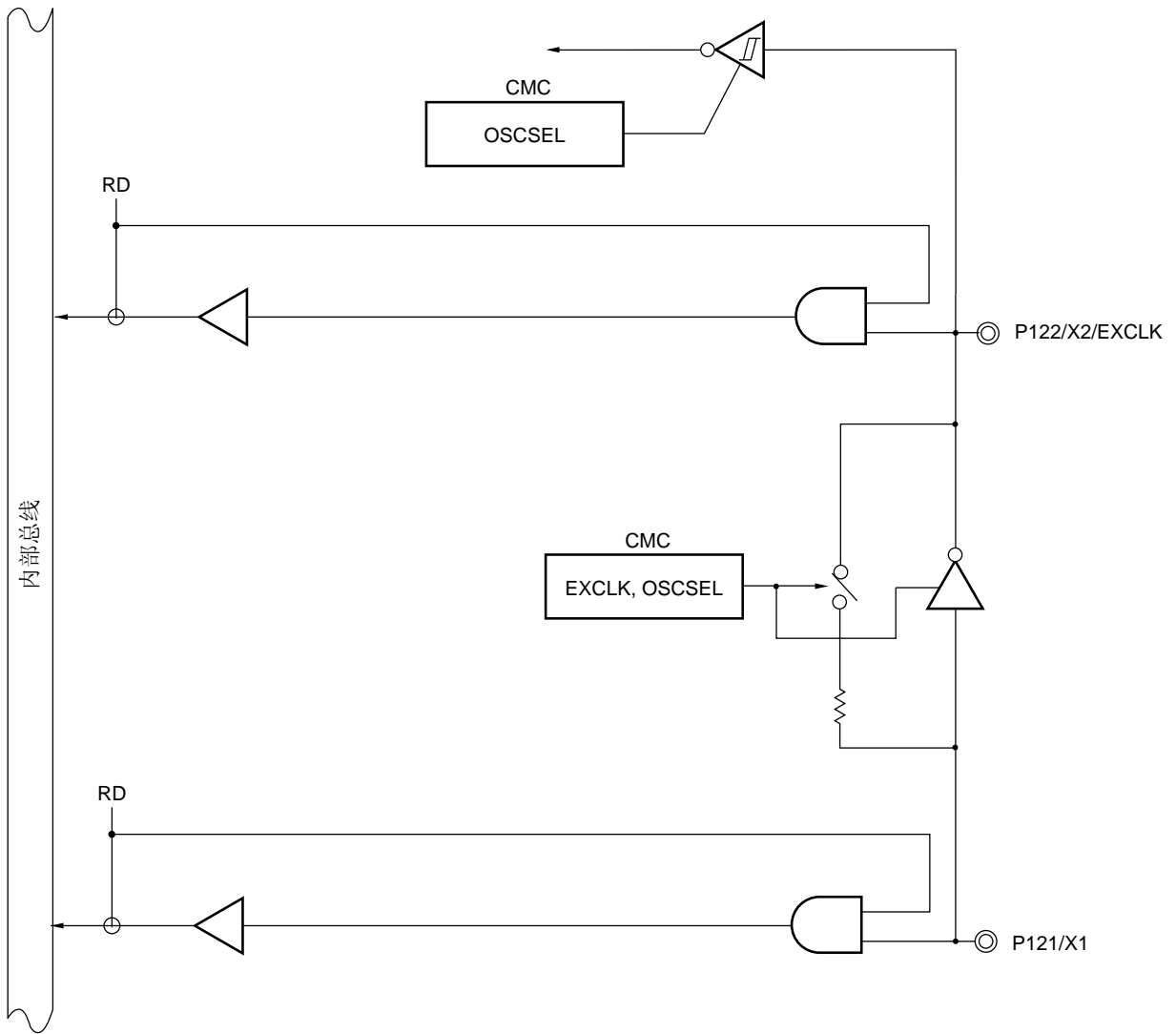
注意事项 在复位释放后 P121~P124 的功能设置只能设置一次。端口一旦设置为连接振荡器，除非执行复位，否则不能用作输入端口。

图 4-29. P120 的框图



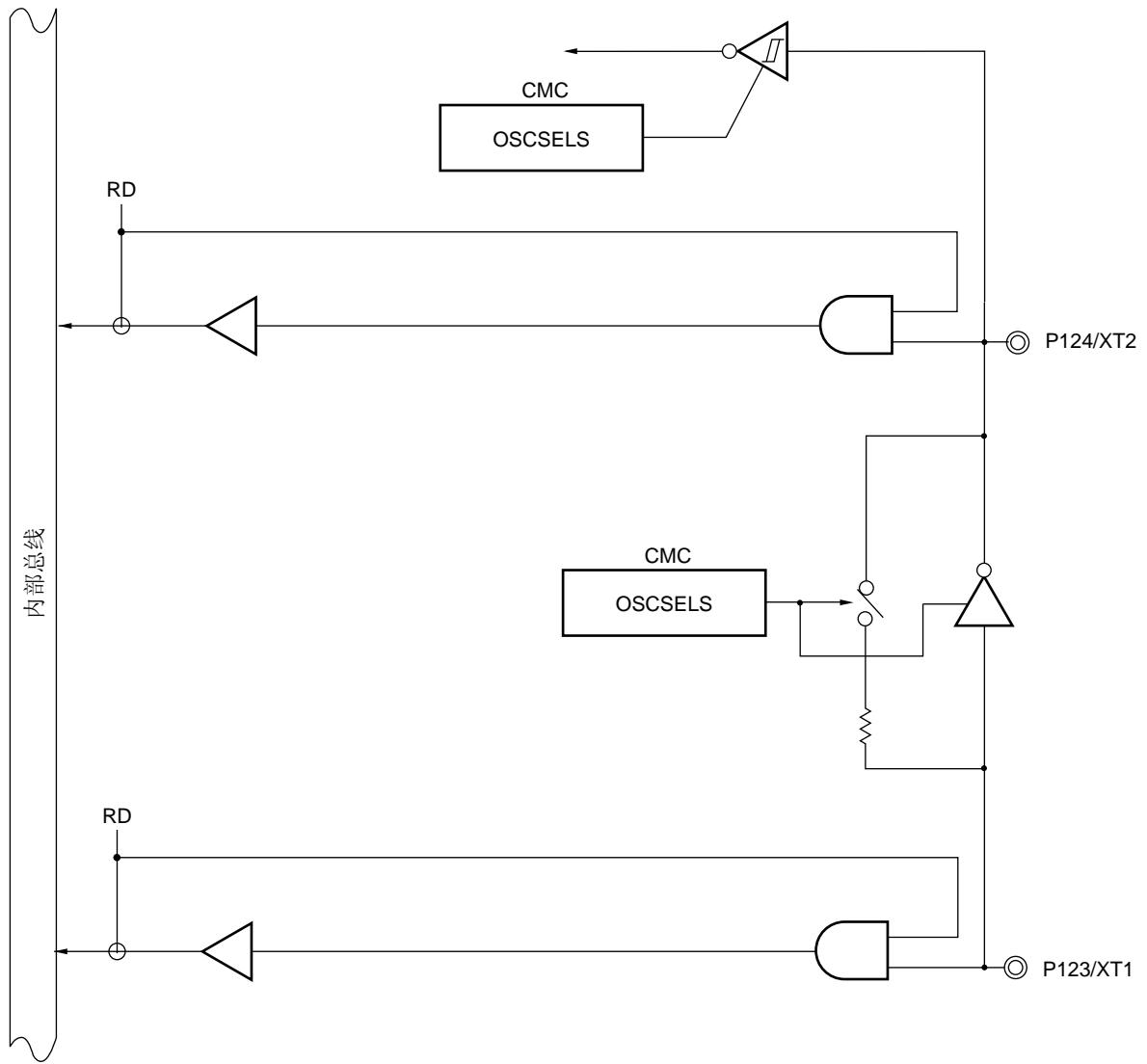
- P12: 端口寄存器 12
- PU12: 上拉电阻选项寄存器 12
- PM12: 端口模式寄存器 12
- RD: 读信号
- WR_{xx}: 写信号

图 4-30. P121 和 P122 的框图



CMC: 时钟操作模式控制寄存器
RD: 读信号

图 4-31. P123 和 P124 的框图



CMC: 时钟操作模式控制寄存器
RD: 读信号

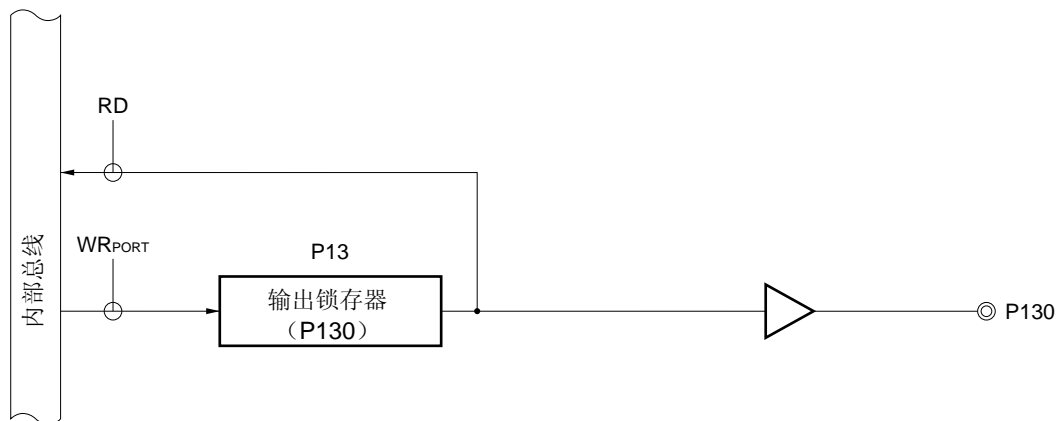
<R>

4.2.12 端口 13

P130 是具有输出锁存功能的 1 位输出端口。

图 4-32 所示为端口 13 的框图。

图 4-32. P130 的框图

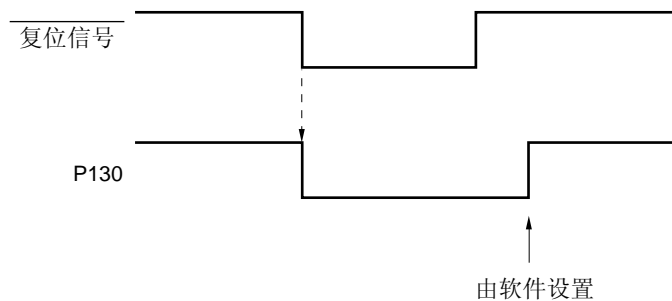


P13: 端口寄存器 13

RD: 读信号

WR_{xx}: 写信号

备注 复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号可作为伪输出 (dummy-output) 的 CPU 复位信号。



4.2.13 端口 14

端口 14 是具有输出锁存功能的 6 位 I/O 端口。通过使用端口模式寄存器 14 (PM14)，可以按位设置端口 14 为输入模式或输出模式。当 P140~P145 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 14 (PU14) 以 1 位单元的方式指定。

通过使用端口输入模式寄存器 14 (PIM14) 以 1 位单元的方式可指定 P142 和 P143 引脚通过普通缓冲器或 TTL 缓冲器输入。

通过端口输出模式寄存器 14 (POM14) 以 1 位为单元的方式可指定 P142~P144 引脚的输出为 N-ch 漏极开路输出 (V_{DD} 耐压)。

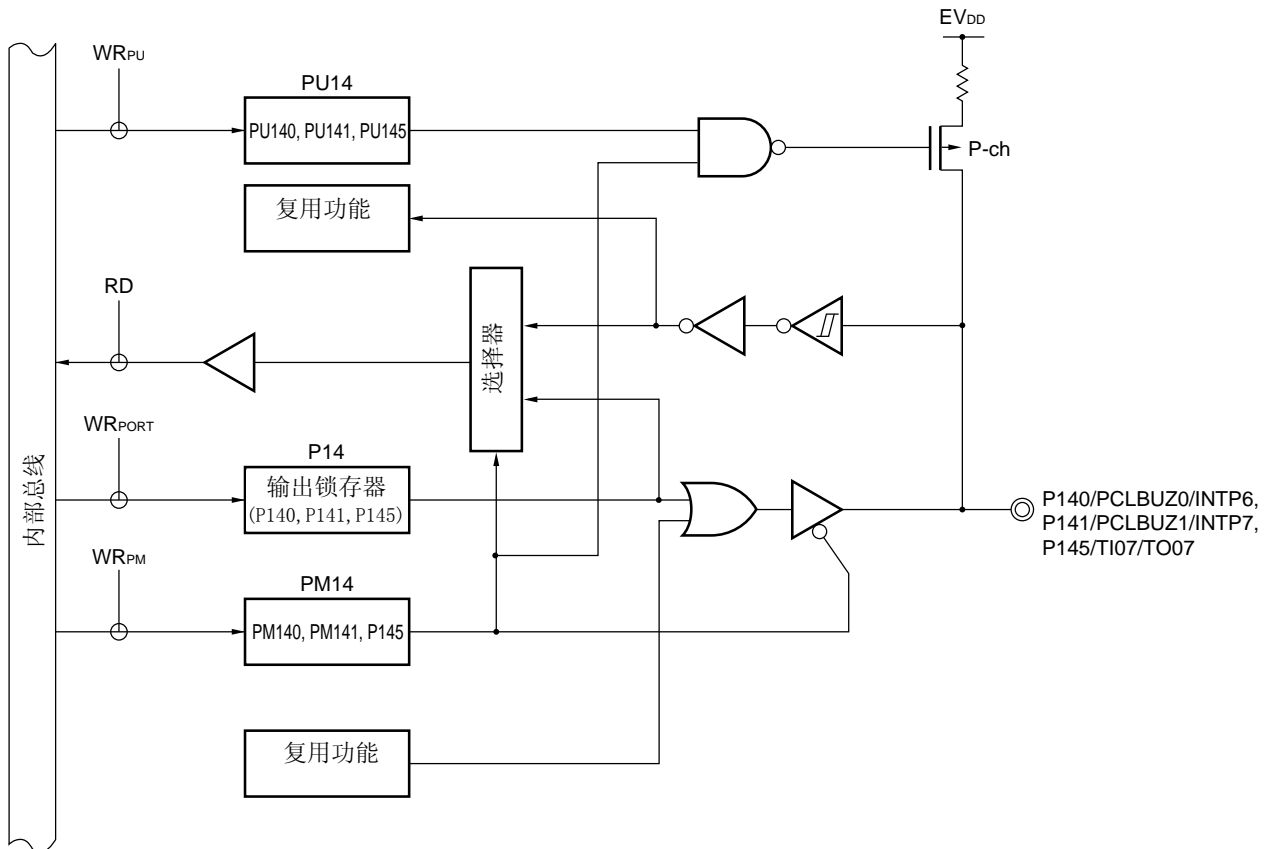
此端口也可用作定时器 I/O、外部中断请求输入、时钟/蜂鸣器输出、串行接口数据 I/O、和时钟 I/O。

复位信号发生设置端口 14 为输入模式。

图 4-33~4-35 所示为端口 14 的框图。

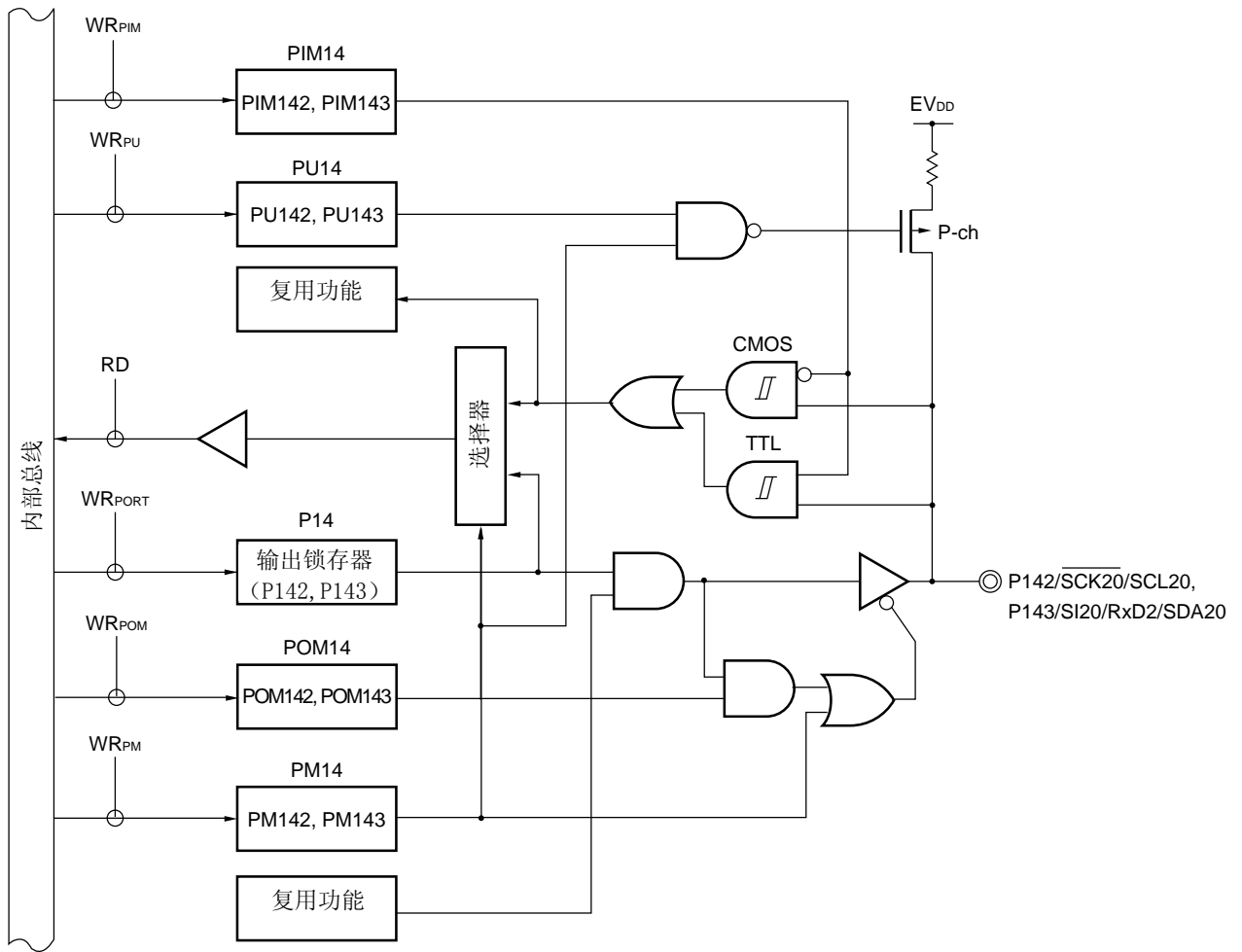
- <R> **注意事项**
1. 要将 P142/ $\overline{\text{SCK20}}$ /SCL20, P143/SI20/RxD2/SDA20 或 P144/SO20/TxD2 作为通用端口使用，注意串行阵列单元 1 的设置。关于更多细节，参见表 12-9 寄存器设置和引脚之间的关系（单元 0 的通道 1: CSI20, UART2 发送, IIC20）和表 12-10 寄存器设置和引脚之间的关系（单元 1 的通道 1: UART2 接收）。
 2. 要将 P145/TI07/TO07 作为通用端口使用，将定时器输出寄存器 0 (TO0) 的第 7 位 (TO07) 和定时器输出允许寄存器 0 (TOE0) 的第 7 位 (TOE07) 设置为“0”，这也就是这些寄存器的默认值。
 3. 要将 P140/PCLBUZ0/INTP6 或 P141/PCLBUZ1/INTP7 作为通用端口使用，设置时钟输出选择寄存器 0 和 1 (CKS0, CKS1) 的第 7 位为“0”，这也就是这些寄存器的默认值。

图 4-33. P140, P141, 和 P145 的框图



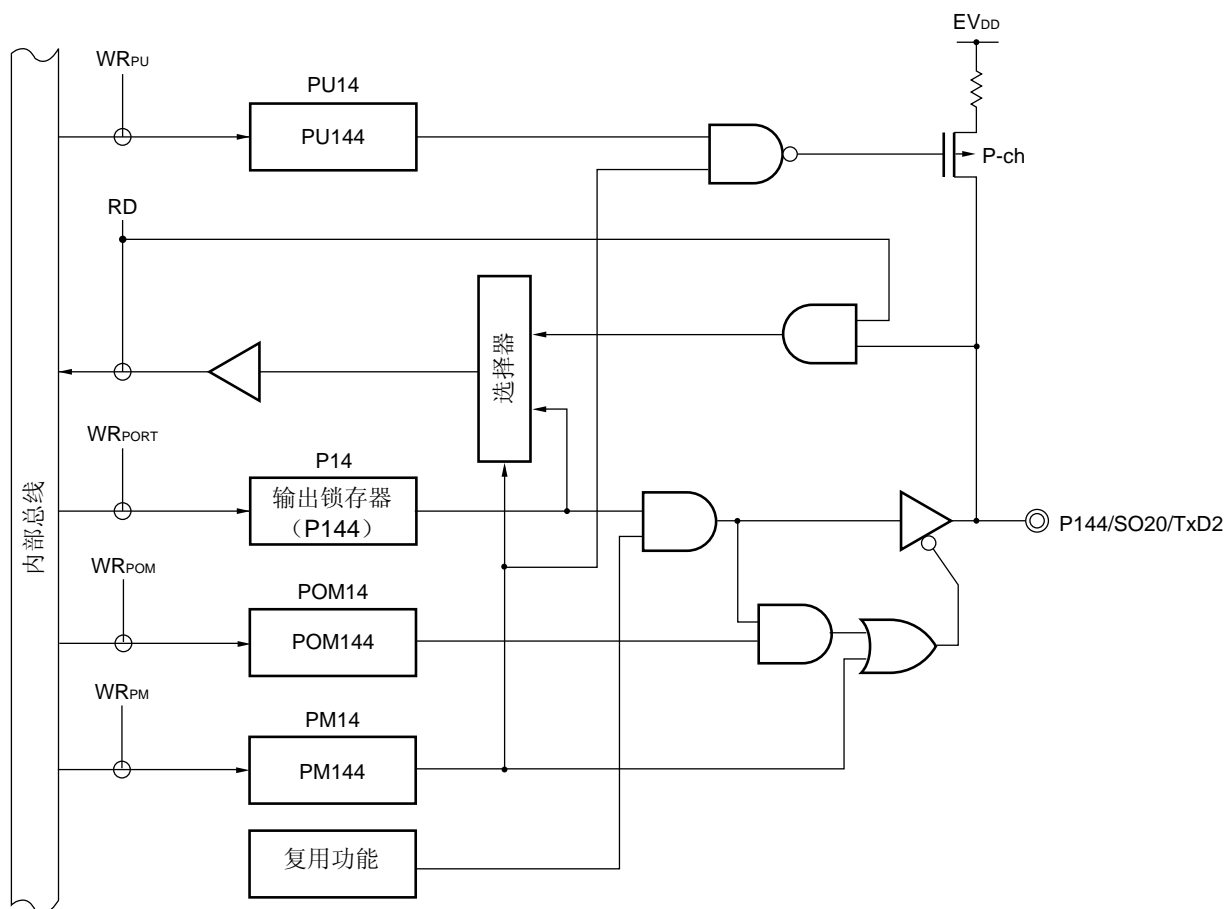
- P14: 端口寄存器 14
- PU14: 上拉电阻选项寄存器 14
- PM14: 端口模式寄存器 14
- RD: 读信号
- WR_{xx}: 写信号

图 4-34. P142 和 P143 的框图



- P14: 端口寄存器 14
- PU14: 上拉电阻选项寄存器 14
- PM14: 端口模式寄存器 14
- RD: 读信号
- WR_{xx}: 写信号

图 4-35. P14 的框图



- P14: 端口寄存器 14
- PU14: 上拉电阻选项寄存器 14
- PM14: 端口模式寄存器 14
- RD: 读信号
- WR_{xx}: 写信号

4.3 控制端口功能的寄存器

端口功能由如下六种类型寄存器控制。

- 端口模式寄存器 (PM0~PM7, PM9, PM11, PM12, PM14)
- 端口寄存器 (P0~P7, P9, P11~P14)
- 上拉电阻选项寄存器 (PU0, PU1, PU3~PU7, PU9, PU12, PU14)
- 端口输入模式寄存器 (PIM0, PIM4, PIM14)
- 端口输出模式寄存器 (POM0, POM4, POM14)
- A/D 端口配置 寄存器 (ADPC)

(1) 端口模式寄存器 (PM0~PM7, PM9, PM11, PM12, PM14)

这类寄存器以 1 位单元的方式定义端口的输入模式或输出模式。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器的内容设置为 FFH。

当端口引脚被用作复用功能时, 需要参考 4.5 使用复用功能时端口模式寄存器和输出锁存器的设置, 对端口模式寄存器进行设置。

图 4-36. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM9	1	1	1	1	1	1	1	PM90	FFF29H	FFH	R/W
PM11	1	1	1	1	1	1	PM111	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PMmn	Pmn 引脚 I/O 模式选择 (m = 0~7, 9, 11, 12, 14; n = 0~7)										
0	输出模式 (输出缓冲器打开)										
1	输入模式 (输出缓冲器关闭)										

注意事项 确保设置PM0的第7位，PM3的第2~7位，PM5的第6~7位，PM9的第1~7位，PM11的第2~7位，PM12的第1~7位，PM14的第6和7位为“1”。

(2) 端口寄存器 (P0~P7, P9, P11~P14)

这类寄存器用于设置芯片端口要输出的数据。

如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值^註。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将寄存器清零 (00H)。

注 P2 设置为 A/D 转换器的模拟输入时，或 P11 设置为 D/A 转换器的模拟输出时，如果端口在输入模式期间读取，它一直为 0，并且无引脚电平读出。

图 4-37. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (输出锁存)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (输出锁存)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (输出锁存)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (输出锁存)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (输出锁存)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	FFF05H	00H (输出锁存)	R/W
P6	P67	P66	P65	P64	P63	P62	P61	P60	FFF06H	00H (输出锁存)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (输出锁存)	R/W
P9	0	0	0	0	0	0	0	P90	FFF09H	00H (输出锁存)	R/W
P11	0	0	0	0	0	0	P111	P110	FFF0BH	00H (输出锁存)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	未定义	R/W [※]
P13	0	0	0	0	0	0	0	P130	FFF0DH	00H (输出锁存)	R/W
P14	0	0	P145	P144	P143	P142	P141	P140	FFF0EH	00H (输出锁存)	R/W

Pmn	m = 0~7, 9, 11~14; n = 0~7	
	输出数据控制 (输出模式)	输入数据读 (输入模式)
0	输出 0	输入低电平
1	输出 1	输入高电平

注 P121~P124 只读。

(3) 上拉电阻选项寄存器 (PU0, PU1, PU3~PU7, PU9, PU12, PU14)

这类寄存器指定是否使用 P00~P06, P10~P17, P30, P31, P40~P47, P50~P55, P64~P67, P70~P77, P90, P120 或 P140~P145 的片内上拉电阻。内部上拉电阻可以以 1 位单元的方式用于那些已设置为输入模式的引脚, 并且这些引脚已通过 PU0, PU1, PU3~PU7, PU9, PU12 和 PU14 指定为使用内部上拉电阻。不论是否设置了 PU0, PU1, PU3~PU7, PU9, PU12 和 PU14, 设置为输出模式的引脚和用作复用功能的输出引脚都不会连接内部上拉电阻。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 (00H)。

图 4-38 上拉电阻选项寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU9	0	0	0	0	0	0	0	PU90	F0039H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W
PUmn	Pmn 引脚 片内上拉电阻 选择 (m = 0, 1, 3~7, 9, 12, 14; n = 0~7)										
0	片内上拉电阻不连接										
1	片内上拉电阻 连接										

(4) 端口输入模式寄存器 (PIM0, PIM4, PIM14)

这些寄存器按位设置 P03, P04, P43, P44, P142 和 P143 的输入缓冲器。

<R>

在和不同电压的外部设备进行串行通讯中可以选择 TTL 输入缓冲。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 (00H)。

图 4-39. 端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIM0	0	0	0	PIM04	PIM03	0	0	0	F0040H	00H	R/W
PIM4	0	0	0	PIM44	PIM43	0	0	0	F0044H	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W
PIMmn	Pmn 引脚输入 缓冲器 选择 (m = 0, 4, 14; n = 2~4)										
0	普通缓冲器										
1	TTL 缓冲器										

(5) 端口输出模式寄存器 (POM0, POM4, POM14)

这些寄存器按位设置 P02~P04, P43, P45, 或 P142~P144 的输出模式。

<R>

在和不同电压的外部设备进行串行通讯中可以选择 N-ch 开漏输出 (V_{DD} 耐压) 模式, 对于 SDA10 和 SDA20 引脚, 在和相同电压的外部设备进行简单 I²C 通讯中可以选择 N-ch 开漏输出 (V_{DD} 耐压) 模式。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 (00H)。

图 4-54. 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM4	0	0	POM45	0	POM43	0	0	0	F0054H	00H	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W
POMmn	Pmn 引脚输出模式 选择 (m = 0, 4, 14; n = 2~5)										
0	普通模式										
1	N-ch 漏极开路 (V _{DD} 耐压) 模式										

(6) A/D 端口配置 寄存器 (ADPC)

此寄存器切换 P20/ANI0~P27/ANI7 引脚到数字 I/O 端口或 A/D 转换器的模拟输入。

ADPC 可以由 1 位或 8 位存储器操作指令设置。

复位信号发生设置此寄存器为 10H。

图 4-55. A/D 端口配置 寄存器 (ADPC) 的格式

地址: F0017H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	模拟输入 (A) / 数字 I/O (D) 切换							
					ANI7/ P27	ANI6/ P26	ANI5/ P25	ANI4/ P24	ANI3/ P23	ANI2/ P22	ANI1/ P21	ANI0/ P20
0	0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D
其他值				设置禁止								

- 注意事项**
1. 通过使用端口模式寄存器 2 (PM2) 将用作 A/D 转换的通道设置为输入模式。
 2. 不要使用模拟输入通道选择寄存器 (ADS) 设置已经由 ADPC 设置为数字 I/O 的引脚。
 3. 当 ANI0/P20~ANI7/P27 所有引脚都被用作数字 I/O (D) 端口时, 由 ADPC4~ADPC0 可以设置为 01000 或 10000。

4.4 端口功能操作

对输入模式和输出模式的端口操作是不同的，具体如下所示。

4.4.1 写入 I/O 端口

(1) 输出模式

使用传送指令对输出锁存器进行写操作，输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号产生时输出锁存器的内容被清零。

(2) 输入模式

使用传送指令对输出锁存器进行写操作，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号产生时输出锁存器的内容被清零。

4.4.2 从 I/O 端口读取

(1) 输出模式

使用传送指令读取的是输出锁存器的内容。输出锁存器的内容不会改变。

(2) 输入模式

使用传送指令读取的是引脚状态。输出锁存器的内容不会改变。

4.4.3 I/O 端口的操作

(1) 输出模式

对输出锁存器执行一个操作时，操作结果写入输出锁存器。而输出锁存器的内容则从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号的产生时输出锁存器的内容被清零。

(2) 输入模式

读取引脚电平，对它的内容执行操作，操作的结果写入输出锁存器，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

当复位信号的产生时输出锁存器的内容被清零。

4.4.4 连接到不同的供电电压（2.5V，3V）的外部设备

当端口 0，4 和 14 部分引脚工作在 $V_{DD} = 4.0\text{ V} \sim 5.5\text{ V}$ ，与可以工作在 2.5V 或 3 V 供电电压的外部设备进行 I/O 连接。

关于输入，CMOS/TTL 可通过端口输入模式寄存器（PIM0，PIM4，PIM14）按位切换。

此外，关于输出，通过端口输出模式寄存器（POM0，POM4，POM14）切换输出缓冲器为 N-ch 漏极开路（ V_{DD} 耐压），即可支持不同的供电电压。

（1）使用 UART1，UART2，CSI01，CSI10 和 CSI20 功能的 I/O 引脚时的设置过程

（a）用作 2.5V 或 3V 输入端口

- <1> 复位释放后，端口模式为输入模式（Hi-Z）。
- <2> 如果需要上拉，使用外部上拉引脚（片内上拉电阻不能使用）。

UART1:	P03
UART2:	P143
CSI01:	P43, P44
CSI10:	P03, P04
CSI20:	P142, P143

- <3> 设置 PIMn 寄存器的相关位为 1，以切换到 TTL 输入缓冲器。
- <4> V_{IH}/V_{IL} 操作在 2.5V 或 3 V 工作电压。

（b）用作 2.5V 或 3 V 输出端口

- <1> 复位释放后，端口模式为输入模式（Hi-Z）。
- <2> 需要使用外部上拉引脚（片内上拉电阻不能使用）。

UART1:	P02
UART2:	P144
CSI01:	P43, P45
CSI10:	P02, P04
CSI20:	P142, P144

- <3> 设置相关端口的输出锁存为 1。
- <4> 设置 POMn 寄存器的相关位为 1 以设置为 N-ch 漏极开路输出（ V_{DD} 耐压）模式。
- <5> 通过操作 PMn 寄存器设置输出模式。
同时，输出数据为高电平，因此引脚为 Hi-Z 状态。
- <6> 根据串行阵列单元的工作状态，只在低电平时才能操作。

备注 n = 0, 4, 14

(2) 当使用简单 IIC10 和 IIC20 功能的 I/O 引脚设置过程

- <1> 复位释放后，端口模式为输入模式（Hi-Z）。
- <2> 需要使用外部上拉引脚（片内上拉电阻不能使用）。

简单 IIC10: P03, P04
简单 IIC20: P142, P143

- <3> 设置相关端口的输出锁存为 1。
- <4> 设置 POMn 寄存器的相关位为 1 以设置为 N-ch 漏极开路 输出（VDD 耐压）模式。
- <5> 通过操作 PMn 寄存器相关位设置输出模式（在输出模式下可以是数据 I/O）。
同时，输出 数据为高电平，因此引脚为 Hi-Z 状态。
- <6> 允许 串行阵列单元的操作，并且设置为简易 IIC 模式。

备注 n = 0, 14

4.5 使用复用功能时端口模式寄存器和输出锁存的设置

要使用端口的复用功能，按下表 4-5 设置端口模式寄存器和输出锁存器。

表 4-5. 使用复用功能时端口模式寄存器和输出锁存的设置 (1/3)

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P00	TI00	输入	1	×
P01	TO00	输出	0	0
P02	SO10	输出	0	1
	TxD1	输出	0	1
P03	SI10	输入	1	×
	RxD1	输入	1	×
	SDA10	I/O	0	1
P04	$\overline{\text{SCK10}}$	输入	1	×
		输出	0	1
	SCL10	I/O	0	1
P05	TI05	输入	1	×
	TO05	输出	0	0
P06	TI06	输入	1	×
	TO06	输出	0	0
P10	$\overline{\text{SCK00}}$	输入	1	×
		输出	0	1
P11	SI00	输入	1	×
	RxD0	输入	1	×
P12	SO00	输出	0	1
	TxD0	输出	0	1
P13	TxD3	输出	0	1
P14	RxD3	输入	1	×
P15	RTCDIV	输出	0	0
	RTCCL	输出	0	0
P16	TI01	输入	1	×

备注 ×: 不必考虑
 PM_{xx}: 端口模式寄存器
 P_{xx}: 端口输出锁存

表 4-6. 使用复用 功能时端口模式寄存器和输出锁存的设置 (2/3)

引脚名称	复用 功能		PMxx	Pxx
	功能名称	I/O		
P17	TI02	输入	1	×
	TO02	输出	0	0
P20~P27 注 1	ANI0~ANI7 注 1	输入	1	×
P30	RTC1HZ	输出	0	0
	INTP3	输入	1	×
P31	TI03	输入	1	×
	TO03	输出	0	0
	INTP4	输入	1	×
P40	TOOL0	I/O	×	×
P41	TOOL1	输出	×	×
P42	TI04	输入	1	×
	TO04	输出	0	0
P43	SCK01	输入	1	×
		输出	0	1
P44	SI01	输入	1	×
P45	SO01	输出	0	1
P50	INTP1	输入	1	×
P51	INTP2	输入	1	×
P60	SCL0	I/O	0	0
P61	SDA0	I/O	0	0
P70~P73	KR0~KR3	输入	1	×
P74~P77	INTP8~INTP11	输入	1	
	KR4~KR7	输入	1	×
P110, P111	ANO0, ANO1 注 2	输出	1	×
P120	INTP0	输入	1	×
	EXLVI	输入	1	×
P140	PCLBUZ0	输出	0	0
	INTP6	输入	1	×
P141	PCLBUZ1	输出	0	0
	INTP7	输入	1	×
P142	SCK20	输入	1	×
		输出	0	1
	SCL20	I/O	0	1

备注 ×: 不必考虑
PMxx: 端口模式寄存器
Pxx: 端口输出锁存

(注 1 和 2 列在下页。)

表 4-6. 使用复用功能时端口模式寄存器和输出锁存的设置 (3/3)

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P143	SI20	输入	1	×
	RxD2	输入	1	×
	SDA20	I/O	0	1
P144	SO20	输出	0	1
	TxD2	输出	0	1
P145	TI07	输入	1	×
	TO07	输出	0	0

备注 ×: 不必考虑
PM_{xx}: 端口模式寄存器
P_{xx}: 端口输出锁存

- 注**
1. ANI0/P20~ANI7/P27 引脚的功能可通过 A/D 端口配置寄存器 (ADPC), 模拟输入通道选择寄存器 (ADS) 和 PM2 来选择。
 2. D/A 转换器操作允许时 (DACE_n = 1), ANO_n 的功能自动选择。而且, 设置端口模式寄存器 11 在输入模式 (PM11_n = 1)。

表 4-6. ANI0/P20~ANI7/P27 引脚的设置功能

ADPC	PM2	ADS	ANI0/P20~ANI7/P27 引脚
数字 I/O 选择	输入模式	-	数字输入
	输出模式	-	数字输出
模拟输入 选择	输入模式	选择 ANI.	模拟输入 (进行转换)
		不要选择 ANI.	模拟输入 (不进行转换)
	输出模式	选择 ANI.	设置禁止
		不要选择 ANI.	

4.6 端口寄存器n (Pn) 的1位操作指令的注意事项

当对一个端口执行1位操作指令，该端口既有输入功能又有输出功能，除了写入目标位，没被操作的输入端口的输出锁存值可能会被改变。

所以，建议在控制一个端口从输入模式切换到输出模式时，要重写输出锁存。

<举例> 当 P10 是输出端口时，P11 ~ P17 是输入端口（所有引脚状态是高电平），端口 1 的端口锁存值是 00H，如果通过 1 位操作指令把输出端口 P10 的输出从低电平变化到高电平，输出端口 1 的输出锁存值变为 FFH。

解释： 写操作的目标是输出锁存，从 PMnm 位是 1 的 Pn 寄存器读出的是引脚状态。
在 78K0R/KF3 中按照如下顺序执行 1 位操作指令。

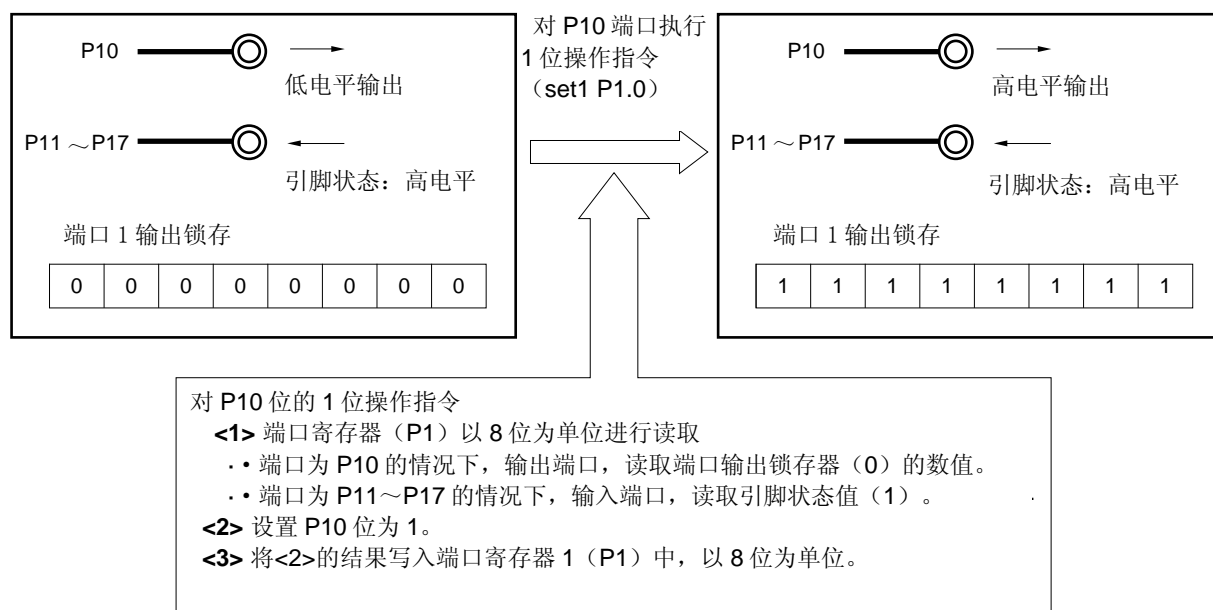
- <1> 以 8 位单元读出 Pn 寄存器。
- <2> 目标的一位被操作。
- <3> 以 8 位单元写 Pn 寄存器。

在步骤<1>中，输出端口 P10 的输出缓存值（0）被读出，然而输入端口 P11 ~ P17 的引脚状态也被读出。如果 P11 ~ P17 的引脚状态在此时是高电平，读出的值为 FEH。

通过<2>中的操作把值改变为 FFH。

FFH 通过<3>中的操作写入输出缓存。

图 4-42. 位操作指令 (P10)



第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器用于产生时钟，并提供给 CPU 和外部硬件设备。

可以使用以下三种系统时钟和时钟振荡器。

(1) 主系统时钟

<1> X1 振荡器

连接一个振荡器到 X1 和 X2，该振荡电路产生 $f_x = 2 \sim 20\text{MHz}$ 的时钟。

通过执行 STOP 指令或设置 MSTOP（时钟操作状态控制寄存器（CSC）的第 7 位），可以停止振荡。

<2> 内部高速振荡器

这个振荡电路产生一个 $f_{IH} = 8\text{MHz}$ （典型值）的时钟。复位释放后，CPU 总是使用这个内部高速振荡时钟开始进行操作。可以通过执行 STOP 指令或者设置 HIOSTOP（CSC 的第 0 位）停止其振荡。

外部主系统时钟（ $f_{EXCLK} = 2 \sim 20\text{MHz}$ ）也可以通过 EXCLK/X2/P122 引脚提供。可以通过执行 STOP 指令或者设置 MSTOP 禁止外部主系统时钟输入。

可以通过设置 MCM0（系统时钟控制寄存器（CKC）的第 4 位）来选择内部高速系统时钟（X1 时钟，或者外部主系统时钟）或内部高速振荡时钟，作为主系统时钟。

(2) 副系统时钟

• XT1 时钟振荡器

通过在 XT1 和 XT2 之间连接一个 32.768kHz 的振荡器，该电路以 $f_{XT} = 32.768\text{kHz}$ 的频率进行振荡。通过设置 XTSTOP（CSC 的第 6 位），可以停止振荡。

备注

f_x : X1 时钟振荡频率
 f_{IH} : 内部高速振荡时钟频率
 f_{EX} : 外部主系统时钟频率
 f_{SUB} : 副系统时钟频率

(3) 内部低速振荡时钟（用于看门狗定时器的时钟）

• 内部低速振荡器

此电路振荡时钟 $f_{IL} = 240\text{kHz}$ （典型值）。

内部低速振荡时钟不能用作 CPU 时钟。唯一使用内部低速振荡时钟操作的硬件就是看门狗定时器。

当看门狗定时器停止时振荡停止。

备注

1. f_{IL} : 内部低速振荡时钟频率
2. 如下情况看门狗定时器停止。
 - 当选项字节（000C0H）的第 4 位（WDTON）= 0
 - 当选项字节（000C0H）的第 4 位（WDTON）= 1 且第 0 位（WDSTBYON）= 0 时，执行 HALT 或 STOP 指令

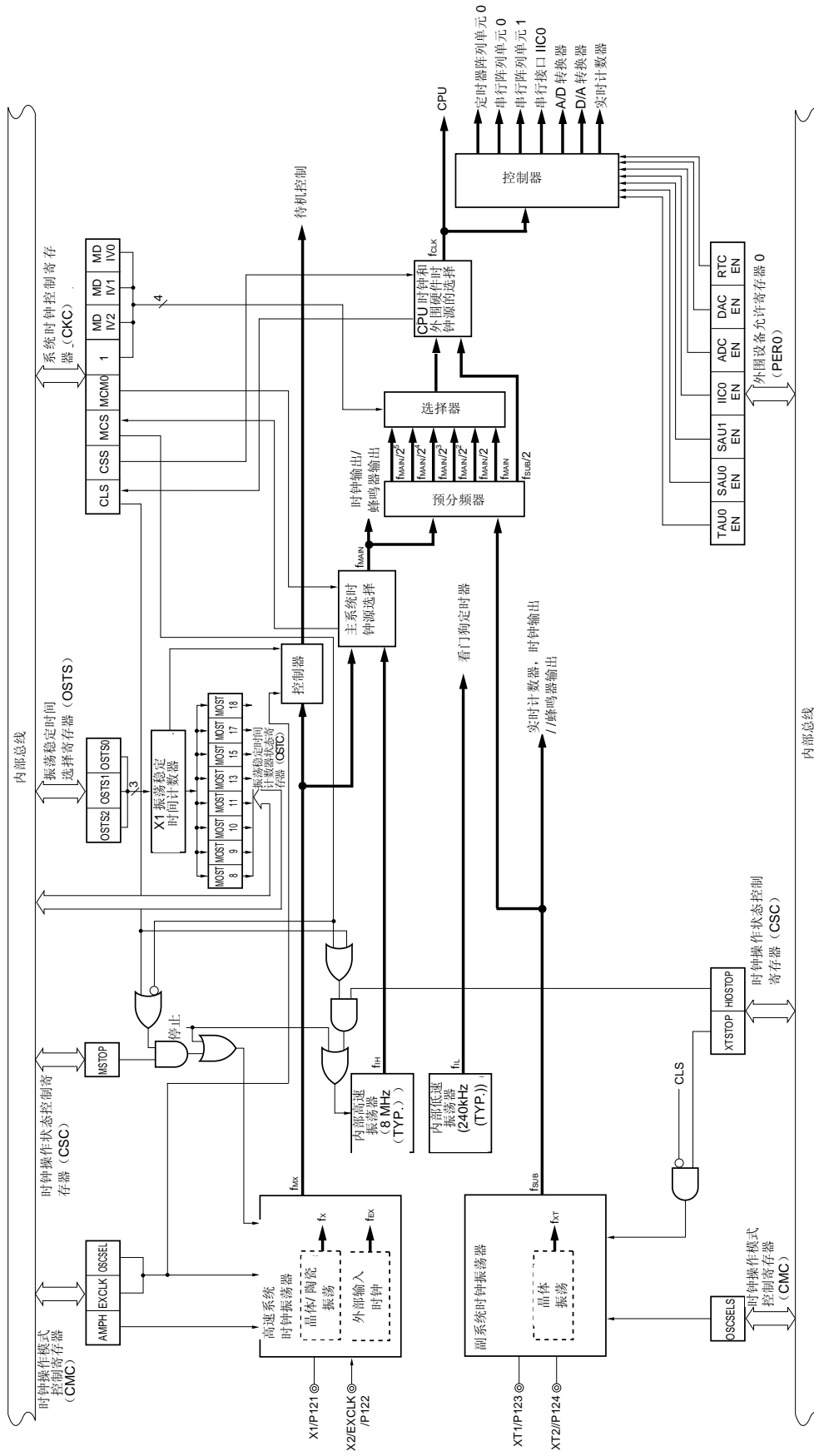
5.2 时钟发生器的配置

时钟发生器包括如下硬件。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	时钟操作模式控制寄存器 (CMC) 时钟操作状态控制寄存器 (CSC) 振荡稳定时间计数器状态寄存器 (OSTC) 振荡稳定时间选择寄存器 (OSTS) 系统时钟控制寄存器 (CKC) 外围设备允许寄存器 0 (PER0) 操作速度模式控制寄存器 (OSMC) 内部高速振荡器微调寄存器 (HIOTRM)
振荡器	X1 振荡器 XT1 振荡器 内部高速振荡器 内部低速振荡器

图 5-1. 时钟发生器的框图



备注	f_X :	X1 时钟振荡频率
	f_{IH} :	内部高速振荡时钟频率
	f_{EX} :	外部主系统时钟频率
	f_{MX} :	高速系统时钟频率
	f_{MAIN} :	主系统时钟频率
	f_{XT} :	XT1 时钟振荡频率
	f_{SUB} :	副系统时钟频率
	f_{CLK} :	CPU/外围硬件时钟频率
	f_{IL} :	内部低速振荡时钟频率

5.3 控制时钟发生器的寄存器

如下八个寄存器用于控制时钟发生器。

- 时钟操作模式控制寄存器 (CMC)
- 时钟操作状态控制寄存器 (CSC)
- 振荡稳定时间计数器状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)
- 系统时钟控制寄存器 (CKC)
- 外围设备允许寄存器 0 (PER0)
- 操作速度模式控制寄存器 (OSMC)
- 内部高速振荡器 微调寄存器 (HIOTRM)

(1) 时钟操作模式控制寄存器 (CMC)

此寄存器用于设置 X1/P121, X2/EXCLK/P122, XT1/P123, 和 XT2/P124 引脚的操作模式, 并选择振荡器频率范围。

复位释放后 CMC 只能由 8 位存储器操作指令写入。此寄存器可由 1 位或 8 位存储器操作指令读取。复位信号产生将此寄存器清为 00H。

图 5-2. 时钟操作模式控制寄存器 (CMC) 的格式

地址: FFFA0H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
EXCLK	OSCSEL	高速系统时钟引脚操作模式		X1/P121 引脚		X2/EXCLK/P122 引脚		
0	0	输入端口模式		输入端口				
0	1	X1 振荡模式		晶体/陶瓷振荡器的连接				
1	0	输入端口模式		输入端口				
1	1	外部时钟输入模式		输入端口		外部时钟输入		
OSCSELS	副系统时钟引脚操作模式			XT1/P123 引脚		XT2/P124 引脚		
0	输入端口模式			输入端口				
0	XT1 振荡模式			晶体振荡器连接				
AMPH	高速系统 时钟振荡频率的控制							
0	$2 \text{ MHz} \leq f_{\text{MX}} \leq 10 \text{ MHz}$							
1	$10 \text{ MHz} < f_{\text{MX}} \leq 20 \text{ MHz}$							

- 注意事项**
1. CMC 在复位释放后只能写入一次, 通过 8 位存储器操作指令写入。
 2. 复位释放后, 在通过时钟操作状态控制寄存器 (CSC) 设置 X1 或 XT1 振荡开始之前, 设置 CMC。
 3. 如果 X1 时钟振荡频率超过 10 MHz, 请确保设置 AMPH 为 1。
 4. 即使寄存器曾为默认值, 为防止程序循环中的错误操作, 建议复位释放后将 CMC 置为默认值 (00H)。

备注 f_{MX} : 高速系统时钟频率

(2) 时钟操作状态控制寄存器 (CSC)

此寄存器用于控制高速系统时钟、内部高速振荡时钟、和副系统时钟（除内部低速振荡时钟）的操作。

CSC 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生会将此寄存器清为 C0H。

图 5-3. 时钟操作状态控制寄存器 (CSC) 的格式

地址: FFFA1H 复位后: C0H R/W

符号	<7>	<6>	5	4	3	2	1	<0>
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
MSTOP	高速系统时钟操作的控制							
	X1 振荡模式		外部时钟输入模式			输入端口模式		
0	X1 振荡器工作		EXCLK 引脚的外部时钟有效			-		
1	X1 振荡器停止		EXCLK 引脚的外部时钟无效			-		
XTSTOP	副系统时钟操作控制							
	XT1 振荡模式				输入端口模式			
0	XT1 振荡器工作				-			
1	XT1 振荡器停止				-			
HIOSTOP	内部高速振荡器时钟操作的控制							
0	内部高速振荡器工作							
1	内部高速振荡器停止							

- 注意事项**
1. 复位释放后，在由 MSTOP 设置的 X1 启动振荡之前、或由 XTSTOP 设置的 XT1 启动振荡之前，设置时钟操作模式控制寄存器 (CMC)。
 2. 要让 MSTOP 设置的 X1 开始振荡，使用振荡稳定时间计数状态寄存器 (OSTC) 检查 X1 时钟的振荡稳定时间。
 3. 不要停止由 OSC 寄存器为 CPU 外围硬件时钟 (fCLK) 选择的时钟。

注意事项 4. 停止时钟振荡（外部时钟输入无效）的寄存器标志设置 和时钟振荡停止条件如下所示。

表 5-2. 停止时钟振荡的条件和标志设置

时钟	停止时钟的条件 (外部时钟输入无效)	CSC 寄存器标志 的设置
X1 时钟 外部主系统时钟	<ul style="list-style-type: none"> • CLS = 0 和 MCS = 0 • CLS = 1 (CPU 和外围硬件时钟工作在除高速系统时钟以外的时钟。) 	MSTOP = 1
副系统时钟	<ul style="list-style-type: none"> • CLS = 0 (CPU 和 外围硬件时钟工作在除副系统时钟以外的时钟。) 	XTSTOP = 1
内部高速振荡时钟	<ul style="list-style-type: none"> • CLS = 0 和 MCS = 1 • CLS = 1 (CPU 和外围硬件时钟工作在除内部高速振荡器时钟以外的时钟。) 	HIOSTOP = 1

(3) 振荡稳定时间计数器状态寄存器 (OSTC)

此寄存器用来表示 X1 时钟振荡稳定时间计数器的计数状态。

X1 时钟振荡稳定时间在如下情况可以被检测，

- 内部高速振荡时钟或副系统时钟用作 CPU 时钟时，如果 X1 时钟开始振荡。
- 当内部高速振荡时钟作为 CPU 时钟并且 X1 时钟在振荡时，如果进入 STOP 模式然后释放。

OSTC 可由 1 位或 8 位存储器操作指令读取。

复位信号产生时，STOP 指令并且 MSTOP (CSC 寄存器的第 7 位) = 1 会将 OSTC 清零为 00H。

备注 振荡稳定时间计数器在如下情况开始计数。

- 当 X1 时钟开始振荡时 (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- 当 STOP 模式释放时

图 5-4. 振荡稳定时间计数器状态寄存器 (OSTC) 的格式

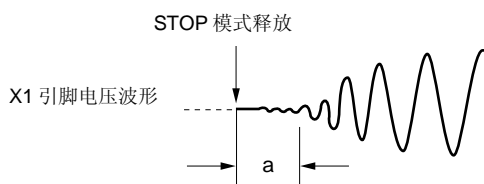
地址: FFFA2H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	振荡稳定时间状态		
								$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$	
0	0	0	0	0	0	0	0	$2^8/f_x \text{ max.}$	25.6 $\mu\text{s max.}$	12.8 $\mu\text{s max.}$
1	0	0	0	0	0	0	0	$2^8/f_x \text{ min.}$	25.6 $\mu\text{s min.}$	12.8 $\mu\text{s min.}$
1	1	0	0	0	0	0	0	$2^9/f_x \text{ min.}$	51.2 $\mu\text{s min.}$	25.6 $\mu\text{s min.}$
1	1	1	0	0	0	0	0	$2^{10}/f_x \text{ min.}$	102.4 $\mu\text{s min.}$	51.2 $\mu\text{s min.}$
1	1	1	1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s min.}$	102.4 $\mu\text{s min.}$
1	1	1	1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s min.}$	409.6 $\mu\text{s min.}$
1	1	1	1	1	1	0	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_x \text{ min.}$	13.11 ms min.	5.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_x \text{ min.}$	26.21 ms min.	13.11 ms min.

- 注意事项
1. 经过上面的时间后, 这些位从 **MOST8** 按顺序设为 **1**, 并且保持为 **1**。
 2. 振荡稳定时间计数器已经达到由 **OSTS** 设置的振荡稳定时间。
在如下情况中, 设置 **OSTS** 的振荡稳定时间的值应该大于由 **OSTC** 寄存器振荡开始后检查的计数值。
 - 内部高速振荡时钟或副系统时钟作为 CPU 时钟时, 如果 X1 时钟开始振荡。
 - 当内部高速振荡时钟作为 CPU 时钟并且 X1 时钟在振荡时, 如果进入 **STOP** 模式然后释放。

(注, 而且由 **OSTS** 设置的振荡稳定时间在 **STOP** 模式释放后设置到 **OSTC**)
 3. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间。(下图“a”表示的部分)



备注 fx: X1 时钟振荡频率

(4) 振荡稳定时间选择寄存器 (OSTS)

此寄存器用于在 STOP 模式释放时，选择 X1 时钟振荡稳定等待时间。

当选择 X1 时钟作为 CPU 时钟时，STOP 模式释放后，操作自动等待的时间就是 OSTS 设置的时间。

当选择内部高速振荡时钟作为 CPU 时钟时，通过 OSTC 来确定在 STOP 模式释放后需要等待的振荡稳定时间已经到达。振荡稳定时间的检查要持续到 OSTC 设置的时间。

OSTS 可由 8 位存储器操作指令设置。

复位信号的产生会将 OSTS 置为 07H。

图 5-5. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFFA3H 复位后: 07H R/W

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

OSTS2	OSTS1	OSTS0		振荡稳定时间选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^8/f_x$	25.6 μs	设置禁止
0	0	1	$2^9/f_x$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_x$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_x$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_x$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	5.55 ms
1	1	1	$2^{19}/f_x$	26.21 ms	13.11 ms

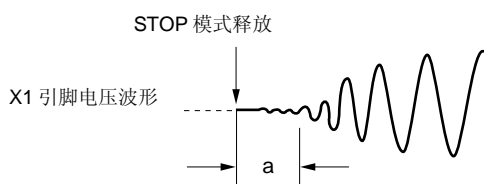
- 注意事项
1. 当 X1 时钟用作 CPU 时钟使用时, 若要设置 STOP 模式, 则必须在执行 STOP 指令之前设置 OSTS。
 2. 禁止设置 振荡稳定时间 小于 20 μs 。
 3. 要改变 OSTS 寄存器的设置, 必须确保 OSTC 寄存器的计数操作已经完成。
 4. 在 X1 时钟振荡稳定时间期间不要改变 OSTS 寄存器的值。
 5. 振荡稳定时间计数器应该达到由 OSTS 设置的振荡稳定时间。

在如下情况中, 设置 OSTS 的振荡稳定时间的值应该大于由 OSTC 寄存器振荡开始后检查的计数值。

- 内部高速振荡时钟或副系统时钟作为 CPU 时钟时, 如果 X1 时钟开始振荡。
- 当内部高速振荡时钟作为 CPU 时钟并且 X1 时钟在振荡时, 如果进入 STOP 模式然后释放。

(注, 而且由 OSTS 设置的振荡稳定时间在 STOP 模式释放后设置到 OSTC)

6. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间 (下图 “a” 表示的部分)。



备注 fx: X1 时钟振荡频率

(5) 系统时钟控制寄存器 (CKC)

此寄存器用于选择 CPU/ 外围硬件时钟和分频率。

CKC 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 09H。

图 5-6. 系统时钟控制寄存器 (CKC) 的格式

地址: FFFA4H 复位后: 09H R/W^{#1}

符号	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	1	MDIV2	MDIV1	MDIV0

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})

MCS	主系统时钟 (f _{MAIN}) 的状态
0	内部高速振荡时钟 (f _{IH})
1	高速系统时钟 (f _{MX})

CSS	MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟 (f _{CLK}) 的选择
0	0	0	0	0	f _{IH}
		0	0	1	f _{IH} /2 (默认)
		0	1	0	f _{IH} /2 ²
		0	1	1	f _{IH} /2 ³
		1	0	0	f _{IH} /2 ⁴
		1	0	1	f _{IH} /2 ⁵
0	1	0	0	0	f _{MX}
		0	0	1	f _{MX} /2
		0	1	0	f _{MX} /2 ²
		0	1	1	f _{MX} /2 ³
		1	0	0	f _{MX} /2 ⁴
		1	0	1	f _{MX} /2 ⁵ 注2
1注3	×注3	×	×	×	f _{SUB} /2
其他值					设置禁止

- 注
1. 第 7 和 5 位只读。
 2. f_{MX} < 4 MHz 时设置禁止。
 3. 当 CSS 设置为 1 时, 禁止改变 MCM0 位的值。

- 备注
1. f_{IH}: 内部高速振荡时钟频率
f_{MX}: 高速系统时钟频率
f_{SUB}: 副系统时钟频率
 2. ×: 不必考虑

(注意事项 1 ~ 3 在下页列出)

- <R>
- 注意事项 1. 第 3 位必须设置为 1。
2. 由 CSS、MCM0 和 MDIV2 ~ MDIV0 设置的时钟供给 CPU 和外围硬件。如果 CPU 时钟改变，因此供给外围硬件（除实时计数器、时钟输出/蜂鸣器输出和看门狗定时器外）的时钟也同时改变。因此，当 CPU/ 外围操作硬件时钟改变时，停止每个外围设备的功能。
3. 如果副系统时钟用作外围硬件时钟，A/D 转换器和 IIC0 的操作不保证。外围硬件的操作特性，请参考描述外围硬件的章节，如第二十八章 电气特性。

在 78K0R/KF3 中执行速度最快的指令，其执行时间在 1 个 CPU 时钟以内。CPU 时钟 (f_{CPU}) 与指令最短执行时间的关系如表 5-3 所示。

表 5-3. CPU 时钟和最小指令执行时间之间的关系

CPU 时钟 (由 MDIV2~ MDIV0 位设置的 值)	最小指令执行时间: $1/f_{CLK}$			
	主系统时钟 (CSS = 0)			副系统时钟 (CSS = 1)
	高速系统时钟 (MCM0 = 1)		内部高速振荡时钟 (MCM0 = 0)	
	运行于 10 MHz	运行于 20 MHz	运行于 8 MHz (典型值)	运行于 32.768 kHz
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (典型值)	—
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (典型值) (默认)	—
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (典型值)	—
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (典型值)	—
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (典型值)	—
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (典型值)	—
$f_{SUB}/2$	—		—	61 μs

备注 f_{MAIN} : 主系统时钟频率 (f_{IH} 或 f_{MX})
 f_{SUB} : 副系统时钟频率

(6) 外围设备允许寄存器 0 (PER0)

这些寄存器用于外围硬件模块的允许和禁止。为了降低功耗和噪声，停止提供时钟给未工作的硬件。

PER0 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 00H。

图 5-7. 外围设备允许寄存器的格式 (1/2)

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	实时计数器 (RTC) 输入时钟 [#] 的控制
0	停止供给输入时钟。 • 用于实时计数器 (RTC) 的 SFR 不能被写入 (可读)。 • 实时计数器 (RTC) 的操作继续。
1	供给输入时钟。 • 用于实时计数器 (RTC) 的 SFR 可以读写。

DACEN	D/A 转换器输入时钟的控制
0	停止供给输入时钟。 • 用于 D/A 转换器的 SFR 不能写入。 • D/A 转换器 在复位状态。
1	供给输入时钟。 • 用于 D/A 转换器的 SFR 可以读写。

ADCEN	A/D 转换器输入时钟的控制
0	停止供给输入时钟。 • 用于 A/D 转换器的 SFR 不能写入。 • A/D 转换器 在复位状态。
1	供给输入时钟。 • 用于 A/D 转换器的 SFR 可以读写。

IIC0EN	串行接口 IIC0 输入时钟的控制
0	停止供给输入时钟。 • 用于串行接口 IIC0 的 SFR 不能写入。 • 串行接口 IIC0 在复位状态。
1	供给输入时钟。 • 用于串行接口 IIC0 的 SFR 可以读写。

注 当从 CPU 访问实时计数器 (RTC) 使用的寄存器时，可通过 RTCEN 控制输入时钟。RTCEN 不能控制供给 RTC 的操作时钟 (f_{SUB})。

注意事项 确保 PER0 的第 1 位为 0。

图 5-7. 外围设备允许寄存器的格式 (2/2)

SAU1EN	串行阵列单元 1 输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 由串行阵列单元 1 使用的 SFR 不能写入。 • 串行阵列单元 1 在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 由串行阵列单元 1 使用的 SFR 可以读写。

SAU0EN	串行阵列单元 0 输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 由串行阵列单元 0 使用的 SFR 不能写入。 • 串行阵列单元 0 在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 由串行阵列单元 0 使用的 SFR 可以读写。

TAU0EN	定时器阵列单元输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 由定时器阵列单元使用的 SFR 不能写入。 • 定时器阵列单元在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 由定时器阵列单元使用的 SFR 可以读写。

注意事项 确保 PER0 的第 1 位为 0。

(7) 操作速度模式控制寄存器 (OSMC)

此寄存器用于控制 flash 存储器高速操作时的加速电路。

如果微控制器运行于小于 10 MHz 的低速系统时钟，通过设置此寄存器到默认值 (00H) 可以降低功耗。

OSMC 可由 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 00H。

图 5-8. 操作速度模式控制寄存器 (OSMC) 的格式

地址: F00F3H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
OSMC	0	0	0	0	0	0	0	FSEL

FSEL	fCLK 频率选择
0	操作频率小于 10 MHz (默认)。
1	操作频率大于 10 MHz。

- 注意事项**
1. 当复位释放后，OSMC 只能写入一次，通过 8 位存储器操作指令写入。
 2. 在下面两个操作前写入“1”到 FSEL。
 - 在 fCLK 时钟分频为某个时钟频率 (f_{IH} 除外) 之前改变时钟。
 - 操作 DMA 控制器。
 3. 当“1”写入到 FSEL 标志时，CPU 等待。
 当 fCLK = f_{IH} 时，等待时间为 15 μs ~ 20 μs (目标值)；当 fCLK = f_{IH}/2 时，30 μs ~ 40 μs (目标值)。
 但是，即使在 CPU 等待时，f_x 的振荡稳定时间仍然会继续计数。
 4. 要使 fCLK 增加到 10 MHz 以上，设置 FSEL 为“1”，然后在两个周期之后改变 fCLK。
 5. 即使当 FSEL = 1 设置，系统时钟可以在 10 MHz 以下的频率内操作。
 即使当 FSEL 设置为“1”，但是需要在 V_{DD} ≥ 2.25 V 的情况下进行。
 即使当 FSEL 设置为“1”，确保在如下时间点 V_{DD} ≥ 2.25 V，即使 fCLK 已经被分频。
 - 当从 STOP 模式释放 f_{IH} 或 f_{EX} 选择作为 fCLK。
 - 当 fCLK 从 f_{SUB} 切换到 f_{MAIN}。

<R>

(8) 内部高速振荡器微调寄存器 (HIOTRM)

此寄存器用于调整内部高速振荡器的精确度。

<R> 通过副系统时钟的内部高速振荡器频率的自测量，此时副系统时钟使用晶体振荡器，定时器可以使用高精度外部时钟输入（实时计数器或定时器阵列单元）等等，该寄存器可以调整精度。

HIOTRM 可由 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 10H。

<R> **注意事项** 在精度调整后，如果温度和 V_{DD} 引脚电压改变，频率可能也会和预期不同。

此外，如果 HIOTRM 寄存器设置为初始值 (10H) 以外的其他任何值，内部高速振荡时钟的振荡精度可能超出 $8\text{ MHz} \pm 5\%$ 。这决定于随后的温度和 V_{DD} 电压变化情况，或 HIOTRM 寄存器的设置。当温度和 V_{DD} 电压改变，在得到需要的频率精度之前必须进行精度调整。

<R>

图 5-9. 内部高速振荡器微调寄存器 (HIOTRM) 的格式

地址: F00F2H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
HIOTRM	0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0

TTRM4	TTRM3	TTRM2	TTRM1	TTRM0	时钟修正值 (目标) ($2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)		
					MIN.	典型值	MAX.
0	0	0	0	0	-5.54%	-4.88%	-4.02%
0	0	0	0	1	-5.28%	-4.62%	-3.76%
0	0	0	1	0	-4.99%	-4.33%	-3.47%
0	0	0	1	1	-4.69%	-4.03%	-3.17%
0	0	1	0	0	-4.39%	-3.73%	-2.87%
0	0	1	0	1	-4.09%	-3.43%	-2.57%
0	0	1	1	0	-3.79%	-3.13%	-2.27%
0	0	1	1	1	-3.49%	-2.83%	-1.97%
0	1	0	0	0	-3.19%	-2.53%	-1.67%
0	1	0	0	1	-2.88%	-2.22%	-1.36%
0	1	0	1	0	-2.23%	-1.91%	-1.31%
0	1	0	1	1	-1.92%	-1.60%	-1.28%
0	1	1	0	0	-1.60%	-1.28%	-0.96%
0	1	1	0	1	-1.28%	-0.96%	-0.64%
0	1	1	1	0	-0.96%	-0.64%	-0.32%
0	1	1	1	1	-0.64%	-0.32%	±0%
1	0	0	0	0	±0% (默认)		
1	0	0	0	1	±0%	+0.32%	+0.64%
1	0	0	1	0	+0.33%	+0.65%	+0.97%
1	0	0	1	1	+0.66%	+0.98%	+1.30%
1	0	1	0	0	+0.99%	+1.31%	+1.63%
1	0	1	0	1	+1.32%	+1.64%	+1.96%
1	0	1	1	0	+1.38%	+1.98%	+2.30%
1	0	1	1	1	+1.46%	+2.32%	+2.98%
1	1	0	0	0	+1.80%	+2.66%	+3.32%
1	1	0	0	1	+2.14%	+3.00%	+3.66%
1	1	0	1	0	+2.48%	+3.34%	+4.00%
1	1	0	1	1	+2.83%	+3.69%	+4.35%
1	1	1	0	0	+3.18%	+4.04%	+4.70%
1	1	1	0	1	+3.53%	+4.39%	+5.05%
1	1	1	1	0	+3.88%	+4.74%	+5.40%
1	1	1	1	1	+4.24%	+5.10%	+5.76%

<R> 注意事项 HIOTRM 寄存器指定值增加, 内部高速振荡频率速度增快; 指定值减小则速度降低。不会发生值增加频率速度降低, 或值减小速度增加的情况。

5.4 系统时钟振荡器

5.4.1 X1 振荡器

X1 振荡器采用晶体振荡器或陶瓷振荡器（2 ~ 20MHz），连接到 X1 和 X2 引脚。

同样可以输入外部时钟。EXCLK 引脚输入时钟信号。

要使用 X1 振荡器，按如下方式设置时钟操作模式控制寄存器（CMC）的第 7 和 6 位（EXCLK, OSCSEL）。

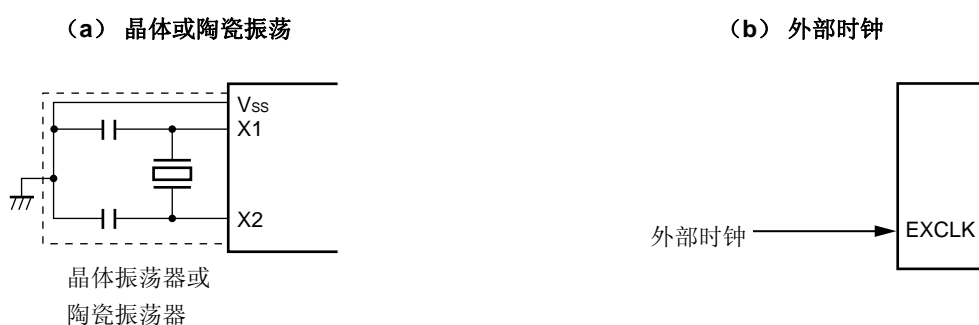
- 晶体或陶瓷振荡： $EXCLK, OSCSEL = 0, 1$
- 外部时钟输入： $EXCLK, OSCSEL = 1, 1$

当 X1 振荡器不使用时，设置输入端口模式（EXCLK, OSCSEL = 0, 0）。

当引脚既不作振荡信号输入，也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-10 为 X1 振荡器的外部电路示例。

图 5-10. X1 振荡器的外部电路的示例



注意事项在下页中列出。

5.4.2 XT1 振荡器

XT1 振荡器采用晶体振荡器（标准值为：32.768 kHz），连接到 XT1 和 XT2 引脚。

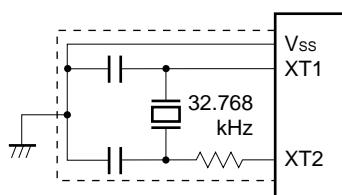
要使用 XT1 振荡器，设置时钟操作模式控制寄存器（CMC）的第 4 位（OSCSELS）为 1。

当 XT1 振荡器不使用时，设置输入端口模式（OSCSELS = 0）。

当引脚既不作振荡信号输入，也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-11 为 XT1 振荡器的外部电路示例。

图 5-11. XT1 振荡器（晶体振荡）的外部电路示例



注意事项在下页中列出。

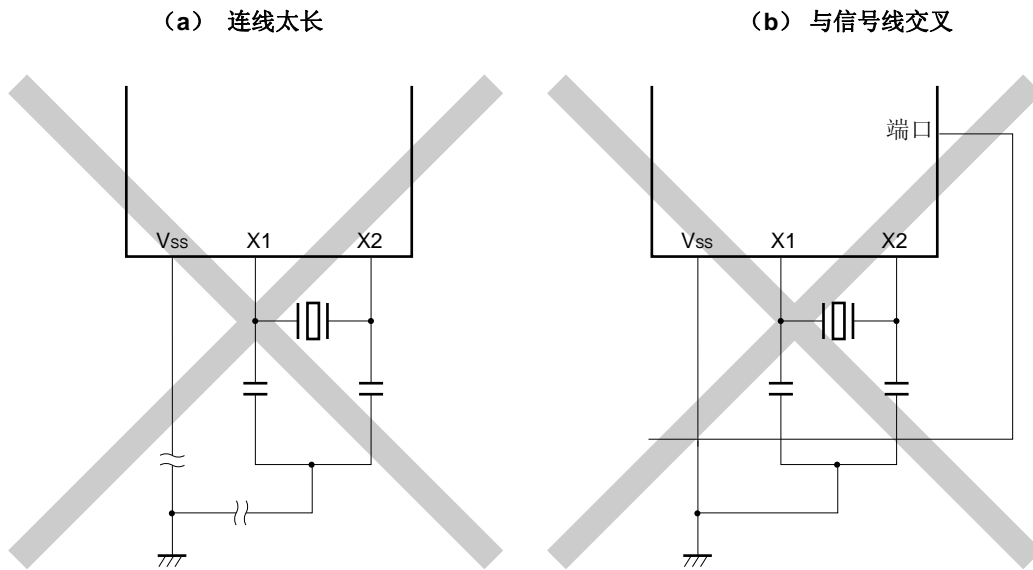
注意事项 1. 在使用 X1 振荡器和 XT1 振荡器时，图 5-10 和图 5-11 中被虚线包围的部分的连线应按照如下连接方法进行，以防止连接线的线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
不要将电容的地信号在大电流接地模式中接入。
- 不要从振荡器获取信号。

注意 XT1 振荡器被设计成低振幅电路，以降低功耗。

图 5-12 为不正确的振荡器连接示例。

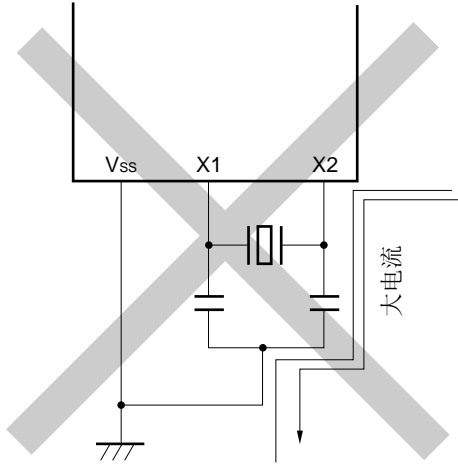
图 5-12. 不正确的振荡器连接示例 (1/2)



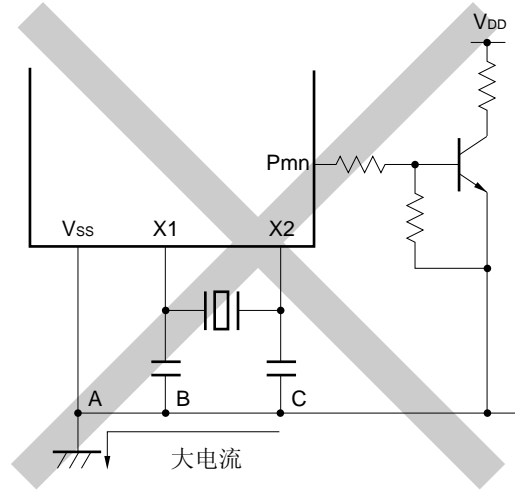
备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也接在 XT2 这边。

图 5-12. 不正确的振荡器连接示例 (2/2)

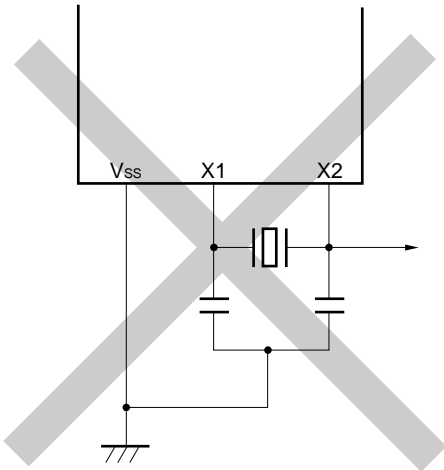
(c) 连接线周围有变化大的电流



(d) 大电流经过振荡器的地线
(A、B、C 三点电势波动)



(e) 获取信号



备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也接在 XT2 这边。

注意事项. 当 X2 和 XT1 并行连接时，X2 的串扰噪音会叠加到 XT1，从而导致错误操作。

5.4.3 内部高速振荡器

78K0R/KF3 产品中包含内部高速振荡器（8 MHz（典型值））。可以通过时钟操作状态控制寄存器（CSC）的第 0 位（HIOSTOP）控制振荡。

复位释放后，内部高速振荡器自动开始振荡。

5.4.4 内部低速振荡器

78K0R/KF3 产品中包含内部低速振荡器。

内部低速振荡时钟只作为看门狗定时器的时钟使用。内部低速振荡时钟不能用作 CPU 时钟。

复位释放后，内部低速振荡器自动产生振荡，同时如果使用选项字节允许看门狗定时器操作，则可以驱动看门狗定时器（240 kHz（典型值））。

只有当看门狗定时器停止时，内部低速振荡器才会停止振荡。当看门狗定时器工作时，即使程序循环，内部低速振荡时钟不会停止。

5.4.5 预分频器

通过对主系统时钟和副系统时钟进行分频，预分频器可以产生多种 CPU/外围硬件时钟。

5.5 时钟发生器操作

时钟发生器用于产生以下几种时钟，并控制 CPU 的操作模式，比如待机模式（参见图 5-1）。

- 主系统时钟 f_{MAIN}
 - 高速系统时钟 f_{MX}
 - X1 时钟 f_x
 - 外部主系统时钟 f_{EX}
 - 内部高速振荡时钟 f_{IH}
- 副系统时钟 f_{SUB}
- 内部低速振荡时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

在 78K0R/KF3 中，当复位释放后内部高速振荡器开始输出时，CPU 开始操作，因此具有以下特点。

(1) 增强安全功能

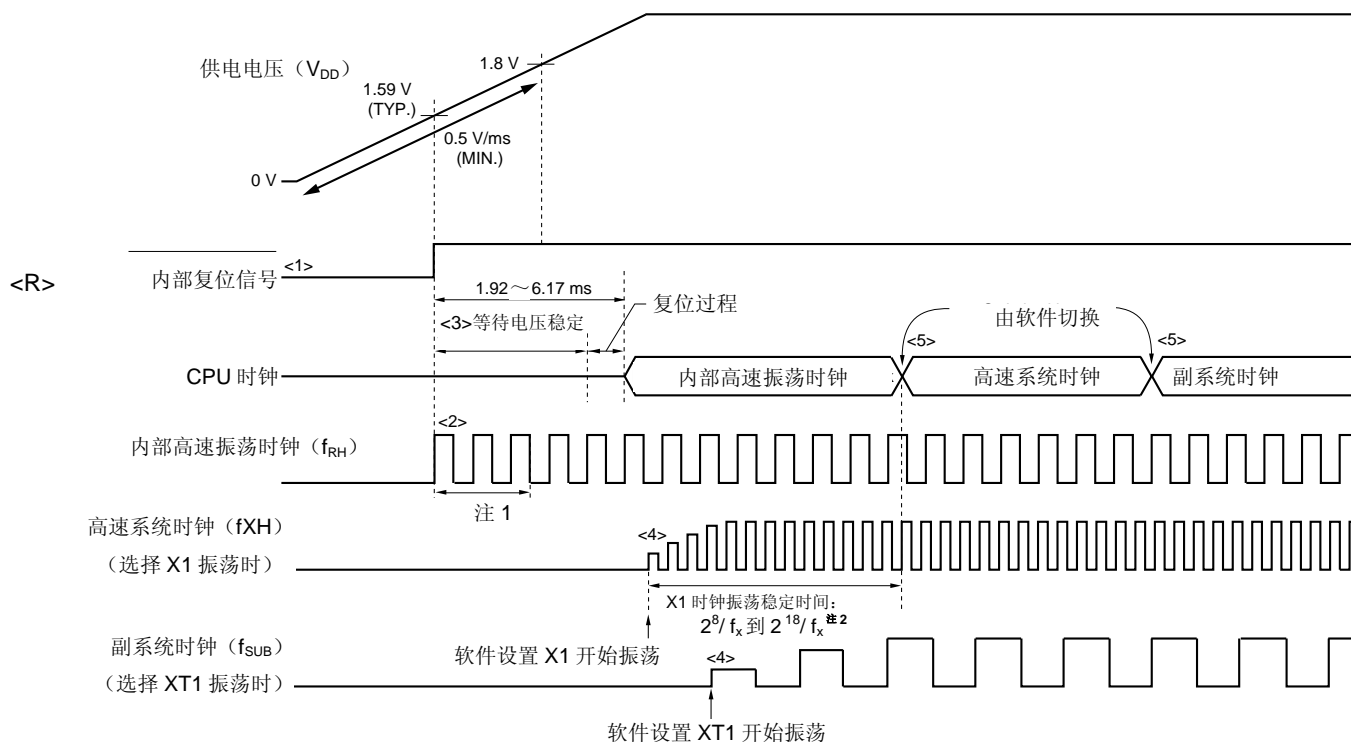
如果 X1 时钟被默认设置为 CPU 时钟，在 X1 时钟遭到损坏或连接错误时设备不能操作，因此复位释放后也不能操作。但是，如果 CPU 的初始时钟是内部高速振荡时钟，则在复位释放后，由内部高速振荡时钟启动设备。这样在异常情况下，可以由软件确认复位源，系统只需执行很少安全处理，便可以确保系统安全关闭。

(2) 改善性能

由于 CPU 可以在不必等待 X1 时钟振荡稳定时间情况下就启动，所以总的性能得到了改善。

图 5-13 和图 5-14 为上电时时钟发生器的操作图。

图 5-13. 供电电源打开时 时钟发生器的操作（当设置 LVI 默认启动功能停止（选项字节：LVIOFF = 1）时）



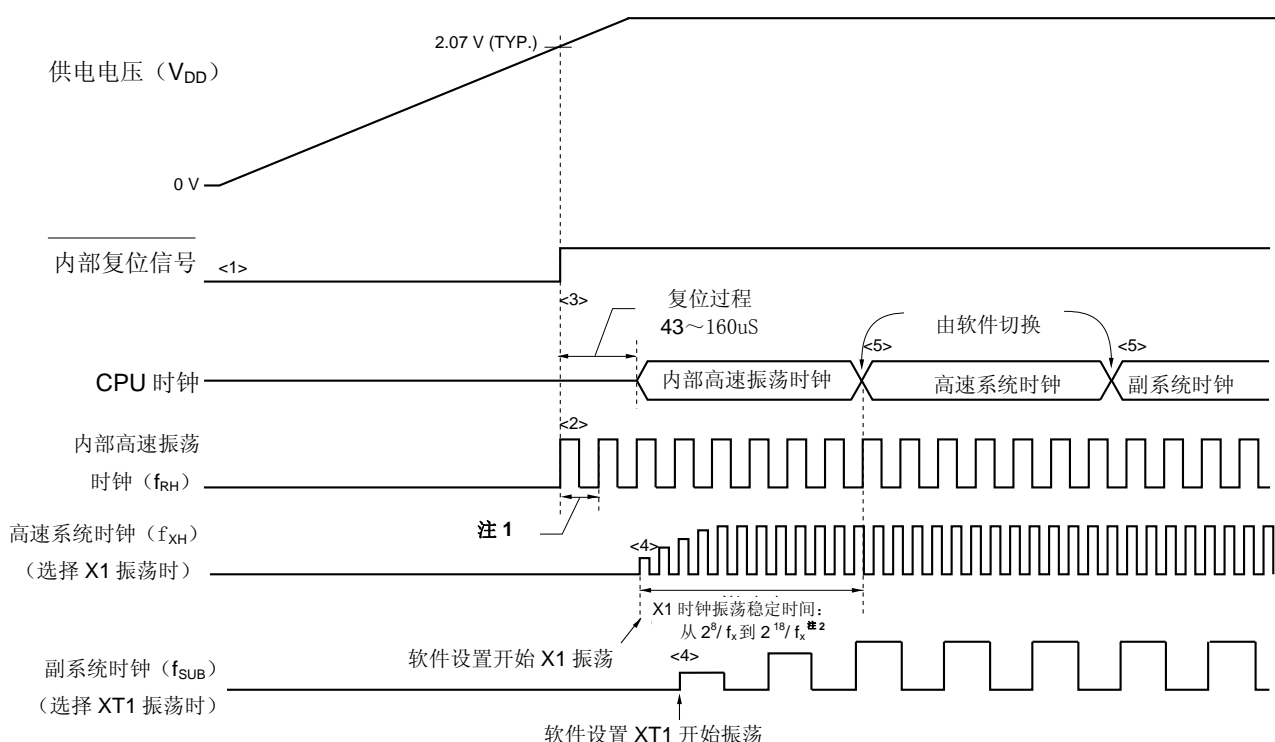
- <1> 当电源开启时，通过上电清零（POC）电路产生一个内部复位信号。
- <2> 当供电电压超过 1.59 V（典型值），复位释放并且内部高速振荡器自动开始振荡。
- <3> 当供电电压上升斜率为 0.5 V/ms（MAX.）时，复位释放并且在经历了供电电压和稳压器的稳定时间后，CPU 开始使用内部高速振荡时钟，然后进行复位处理。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始（参见 5.6.1 高速系统时钟控制示例中的（1）和 5.6.3 副系统时钟控制示例中的（1））。
- <5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换（参见 5.6.1 高速系统时钟控制示例中的（3）和 5.6.3 副系统时钟控制示例中的（3））。

- 注**
- 内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。
 - 释放复位后（如上图所示）或在 CPU 使用内部高速振荡时钟情况下释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器（OSTC）来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟（X1 振荡），则可使用振荡稳定时间选择寄存器（OSTS）设置释放 STOP 模式时的振荡稳定时间。

- 注意事项**
1. 在供电电压达到 1.8V 之前，如果电压上升斜率小于 0.5 V/ms (MAX.)，则输入一个低电平到电源的 **RESET** 引脚直至电压达到 1.8V，或者通过使用选项字节 (**POCMODE = 1**) 设置 2.7V/1.59V POC 模式 (见图 5-14)。通过这种方式，CPU 的操作时序与 <2> 及由 **RESET** 引脚释放复位后的时序相同，如图 5-13 所示。
 2. 当使用 **EXCLK** 引脚的外部时钟输入时，不需要等待振荡稳定时间。

备注 当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 **STOP** 指令来停止 (参见 5.6.1 高速系统时钟控制示例中的 (4) 和 5.6.3 副系统时钟控制示例中的 (4))。

图 5-14. 供电电源打开时时钟发生器的操作 (当设置 LVI 默认启动功能允许 (选项字节: **LVIOFF = 0**) 时)



- <1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 2.07 V (典型值)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始 (参见 5.6.1 高速系统时钟控制示例中的 (1) 和 5.6.3 副系统时钟控制示例中的 (1))。
- <5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换 (参见 5.6.1 高速系统时钟控制示例中的 (3) 和 5.6.3 副系统时钟控制示例中的 (3))。

- 注**
1. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 2. 释放复位后（如上图所示）或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器（OSTC）来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟（X1 振荡），则可使用振荡稳定时间选择寄存器（OSTS）设置释放 STOP 模式时的振荡稳定时间。
- 注意事项**
1. 在供电电压达到 1.59V（典型值）后，必需有电压振荡稳定时间。如果在供电电源振荡稳定时间内电源电压从 1.59 V（典型值）上升到 2.07 V（典型值），在复位处理前自动产生供电电源振荡稳定时间。
 2. 当使用 EXCLK 引脚的外部时钟输入时，不需要等待振荡稳定时间。
- 备注**
- 当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止（参见 5.6.1 高速系统时钟控制示例中的（4），5.6.2 内部高速振荡时钟控制示例中的（3）和 5.6.3 副系统时钟控制示例中的（4））。

5.6 控制时钟

5.6.1 控制高速系统时钟的示例

如下两种高速系统时钟可用。

- X1 时钟：晶体/陶瓷振荡器连接到 X1 和 X2 引脚。
- 外部主系统时钟：外部时钟输入到 EXCLK 引脚。

不使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚可用作 I/O 端口引脚。

注意事项 X1/P121 和 X2/EXCLK/P122 引脚在复位释放后是输入端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) X1 时钟振荡时
- (2) 当使用外部主系统时钟
- (3) 当使用高速系统时钟作为 CPU/ 外围硬件时钟
- (4) 停止高速系统时钟时

(1) X1 时钟振荡时设置过程示例

<1> 设置 P121/X1 和 P122/X2/EXCLK 引脚并设置振荡频率（CMC 寄存器）

- $2 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	0

- $10 \text{ MHz} < f_x \leq 20 \text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	1

- 备注**
1. f_x : X1 时钟振荡频率
 2. P123/XT1 和 P124/XT2 引脚的设置，参见 5.6.3 控制副系统时钟的示例。

<2> 控制 X1 时钟振荡（MOC 寄存器）
若 MSTOP 被清零，则 X1 振荡器开始振荡。

<3> 等待 X1 时钟振荡稳定
检测 OSTC 寄存器并等待所需的时间。
在等待时间内，其它软件处理过程的执行可以使用内部高速振荡时钟。

- 注意事项**
1. 当复位释放后，CMC 寄存器只能写入一次，通过 8 位存储器操作指令写入。
因此，需要同时设置 OSCSELS 位的值。关于 OSCSELS 位，参见 5.6.3 副系统时钟控制的示例。
 2. 在供电电压达到时钟需要的工作电压后，设置 X1 时钟（参见第二十八章 电气特性）。

(2) 使用外部主系统时钟时设置过程示例

<1> 设置 P121/X1 和 P122/X2/EXCLK 引脚 (CMC 寄存器)

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
1	1	0	0/1	0	0	0	×

备注 1. ×: 不必考虑
 2. P123/XT1 和 P124/XT2 引脚的设置, 参见 5.6.3 (1) 当副系统时钟振荡时的设置过程的示例。

<2> 控制外部主系统时钟输入 (CSC 寄存器)
 MSTOP 清为 0 时, 外部主系统时钟 的输入允许。

注意事项 1. 当复位释放后, CMC 寄存器只能写入一次, 通过 8 位存储器操作指令写入。因此, 需要同时设置 OSCSELS 位的值。关于 OSCSELS 位, 参见 5.6.3 副系统时钟控制的示例。
 2. 在供电电压达到时钟需要的工作电压后, 设置外部主系统时钟 (参见 第二十八章 电气特性)。

(3) 使用高速系统时钟作为 CPU/ 外部硬件时钟时设置过程示例

<1> 设置高速系统 时钟振荡^注
 (参见 5.6.1 (1) X1 时钟振荡时设置过程示例和 (2) 使用外部主系统时钟时设置过程示例。)

注 当高速系统时钟已经运行时, 不需要设置<1>的步骤。

<2> 设置高速系统时钟作为 CPU/ 外围硬件时钟的源时钟, 并设置时钟的分频 (CKC 寄存器)

MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟 (f _{CLK}) 的选择
1	0	0	0	f _{MX}
	0	0	1	f _{MX} /2
	0	1	0	f _{MX} /2 ²
	0	1	1	f _{MX} /2 ³
	1	0	0	f _{MX} /2 ⁴
	1	0	1	f _{MX} /2 ⁵ ^注

注 f_{MX} < 4 MHz 时禁止设置。

<3> 如果外围硬件模块不使用，可以停止供给硬件模块的输入时钟。

(PER0 寄存器)

RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
xxxEN	输入时钟控制						
0	停止供给输入时钟。						
1	供给输入时钟。						

注意事项 确保 PER0 的第 1 位为 0。

备注	RTCEN:	实时计数器输入时钟的控制
	DACEN:	D/A 转换器输入时钟的控制
	ADCEN:	A/D 转换器输入时钟的控制
	IIC0EN:	串行接口 IIC0 输入时钟的控制
	SAU1EN:	串行阵列单元 1 输入时钟的控制
	SAU0EN:	串行阵列单元 0 输入时钟的控制
	TAU0EN:	定时器阵列单元的输入时钟的控制

(4) 停止高速系统时钟时设置过程示例

在如下两种情况下，高速系统时钟停止（如果使用外部时钟，禁止时钟输入）。

- 执行 STOP 指令
- 设置 MSTOP 为 1

(a) 执行 STOP 指令

<1> 设置停止外围硬件

停止那些在 STOP 模式下无法使用的外部硬件（需要了解在 STOP 模式下无法使用的外部硬件，可参见第十八章 待机功能）

<2> STOP 模式释放后设置 X1 时钟振荡稳定时间

当 CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTs。

<3> 执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且 X1 振荡停止（禁止外部时钟输入）。

(b) 通过将 **MSTOP** 设置为 1 来停止 X1 振荡（禁止外部时钟输入）

<1> 确认 CPU 时钟状态（CKC 寄存器）

根据 CLS 和 MCS 确认 CPU 的操作时钟不是高速系统时钟。

当 CLS = 0 且 MCS = 1 时，CPU 采用高速系统时钟，因此将 CPU 时钟切换为副系统时钟或者内部高速振荡时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> X1 时钟振荡重启后设置 X1 时钟振荡的稳定时间[※]

X1 时钟振荡重启后，在设置 **MSTOP** 为"1"之前，设置 **OSTS** 寄存器的值，该值应该大于由 **OSTS** 寄存器确定的计数值。

<3> 停止高速系统时钟（CSC 寄存器）

MSTOP 设置为 1 时，X1 振荡停止（外部时钟的输入禁止）。

注 当高速系统时钟在 X1 振荡模式下，此设置需要恢复 X1 时钟振荡。
外部时钟输入模式中不需要此项设置。

注意事项 设置 **MSTOP** 为 1 时一定要确保 **MCS = 0** 或 **CLS = 1**。另外，停止工作在高速系统时钟的外围硬件。

5.6.2 控制内部高速振荡时钟的示例

下面描述的示例是以下情况时的设置过程示例。

- (1) 当重启内部高速振荡时钟的振荡时
- (2) 当使用内部高速振荡时钟作为 CPU/ 外围硬件时钟时
- (3) 当停止内部高速振荡时钟时

(1) 内部高速振荡时钟重新振荡时设置过程示例[※]

<1> 内部高速振荡时钟重新振荡的设置（CSC 寄存器）

当 **HIOSTOP** 清为 0 时，内部高速振荡时钟重新振荡。

注 复位释放后，内部高速振荡器自动开始振荡，并且内部高速振荡时钟作为 CPU/ 外围硬件时钟使用。

(2) 内部高速振荡时钟作为 CPU/ 外围硬件时钟的设置过程示例

<1> 内部高速振荡时钟重新振荡[※]

（参见 5.6.2 (1) 当重启内部高速振荡时钟时设置过程示例）。

注 当内部高速振荡时钟已经运行时，不需要设置<1>。

<2> 设置内部高速振荡时钟作为 CPU/外围硬件时钟的源时钟，并设置时钟（CKC 寄存器）的分频系数

MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟 (f _{CLK}) 的选择
0	0	0	0	f _H
	0	0	1	f _H /2
	0	1	0	f _H /2 ²
	0	1	1	f _H /2 ³
	1	0	0	f _H /2 ⁴
	1	0	1	f _H /2 ⁵

注意事项 如果在重启内部高速振荡时钟后将 CPU/外围硬件时钟从高速系统时钟切换到内部高速振荡时钟，至少经过 10 μs 之后才可以进行该切换操作。

如果在内部高速振荡时钟重启后立即切换，则内部高速振荡的精度在 10 μs 内无法保证。

(3) 停止内部高速振荡时钟时设置过程示例

可以用如下两种方式停止内部高速振荡时钟。

- 执行 STOP 指令
- 设置 HIOSTOP 为 1

(a) 执行 STOP 指令

<1> 外围硬件的设置

停止那些不能在 STOP 模式下使用的外部硬件（如需了解在 STOP 模式下无法使用的外部硬件，可参见第十八章 待机功能）。

<2> 释放待机模式后 X1 时钟振荡稳定时间的设置

CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTs。

<3> 执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且停止内部高速振荡时钟。

(b) 通过设置 HIOSTOP 为 1 来停止内部高速振荡时钟

<1> 确定 CPU 时钟状态（CKC 寄存器）

根据 CLS 和 MCS 确认 CPU 的操作时钟不是内部高速振荡时钟。

CLS = 0 和 MCS = 0 时，CPU 采用内部高速振荡时钟，因此将 CPU 时钟切换为高速系统时钟或副系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	x	副系统时钟

<2> 停止内部高速振荡时钟（CSC 寄存器）

当 HIOSTOP 设置为 1 时，内部高速振荡时钟停止。

注意事项 当 HIOSTOP = 1 时，必须确认 MCS = 1 或 CLS = 1。此外，必须停止正在使用内部高速振荡时钟的外部硬件。

5.6.3 控制副系统时钟示例

通过连接晶体振荡器到 XT1 和 XT2 引脚，副系统时钟可振荡。

当不使用副系统时钟时，XT1/P123 和 XT2/P124 引脚可用作输入端口引脚。

注意事项 XT1/P123 和 XT2/P124 引脚在复位释放后是输入端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) XT1 时钟振荡
- (2) 使用副系统时钟作为 CPU 时钟
- (3) 停止副系统时钟

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件（除实时计数器、时钟输出/蜂鸣器输出和看门狗定时器外）。此时，不保证 A/D 转换器和 IIC0 的操作。外围硬件的操作特性，参见各章节中的外围硬件描述和第二十八章 电气特性。

(1) 副系统时钟振荡的设置示例

<1> 设置 P123/XT1 和 P124/XT2 引脚（CMC 寄存器）

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0/1	0/1	0	1	0	0	0	×

- 备注**
1. ×：不必考虑
 2. P121/X1 和 P122/X2 引脚的设置，参见 5.6.1 高速系统时钟控制示例。

<2> 副系统时钟（CSC 寄存器）振荡控制

如果 XTSTOP 清为 0，XT1 振荡器开始振荡。

<3> 等待副系统时钟振荡稳定

通过软件等待副系统时钟的振荡稳定时间，使用定时器功能。

注意事项 当复位释放后，CMC 寄存器只能写入一次，通过 8 位存储器操作指令写入。

因此，需要同时设置 EXCLK 和 OSCSEL 位的值。关于 EXCLK 和 OSCSEL 位，参见 5.6.1

- (1) X1 时钟振荡设置过程示例或 5.6.1 (2) 使用外部主系统时钟时设置过程示例。

(2) 使用副系统时钟作为 CPU 时钟时设置过程示例

<1> 设置副系统时钟振荡[‡]

(参见 5.6.3 (1) 副系统时钟振荡设置过程示例。)

注 当副系统时钟已经运行时，不需要设置步骤<1>。

<2> 设置副系统时钟作为 CPU 时钟 (CKC 寄存器) 的源时钟

CSS	CPU/外围硬件时钟 (fCLK) 的选择
1	f _{SUB} /2

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件（除实时计数器，时钟输出/蜂鸣器输出，和看门狗定时器外）。此时，不保证 A/D 转换器和 IIC0 的操作。外围硬件的操作特性，参见各章节中的外围硬件描述和第二十八章 电气特性

(3) 停止副系统时钟时设置过程示例

<1> 确定 CPU 时钟状态 (CKC 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟。

当 CLS = 1 时，CPU 采用副系统时钟，因此将 CPU 时钟切换为内部高速振荡时钟或高速系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> 停止副系统时钟 (CSC 寄存器)

XTSTOP 设置为 1 时，停止 XT1 振荡。

注意事项 1. 当设置 XTSTOP 为 1 时确保 CLS = 0。另外，停止那些运行于副系统时钟的外围硬件。
2. 副系统时钟振荡不能通过使用 STOP 指令来停止。

5.5.4 内部低速振荡时钟的控制示例

内部低速振荡时钟不能用作 CPU 时钟。只能用作看门狗定时器时钟。

复位释放后内部低速振荡器自动开始振荡，并且如果通过选项字节允许看门狗定时器操作，则驱动看门狗定时器（240 kHz（典型值））。

除非看门狗定时器停止，否则内部低速振荡器会继续振荡。当看门狗定时器操作时，即使程序循环的情况下，内部低速振荡时钟也不停止。

(1) 停止内部低速振荡时钟时设置过程示例

内部低速振荡时钟在下面两种情况下可被停止。

- 通过选项字节（000C0H 的第 0 位（WDSTBYON）= 0）在 HALT/STOP 模式下停止看门狗定时器，并且执行 HALT 或 STOP 指令。
- 通过选项字节（000C0H 的第 4 位（WDTON）= 0）停止看门狗定时器。

(2) 内部低速振荡时钟重新振荡时设置过程示例

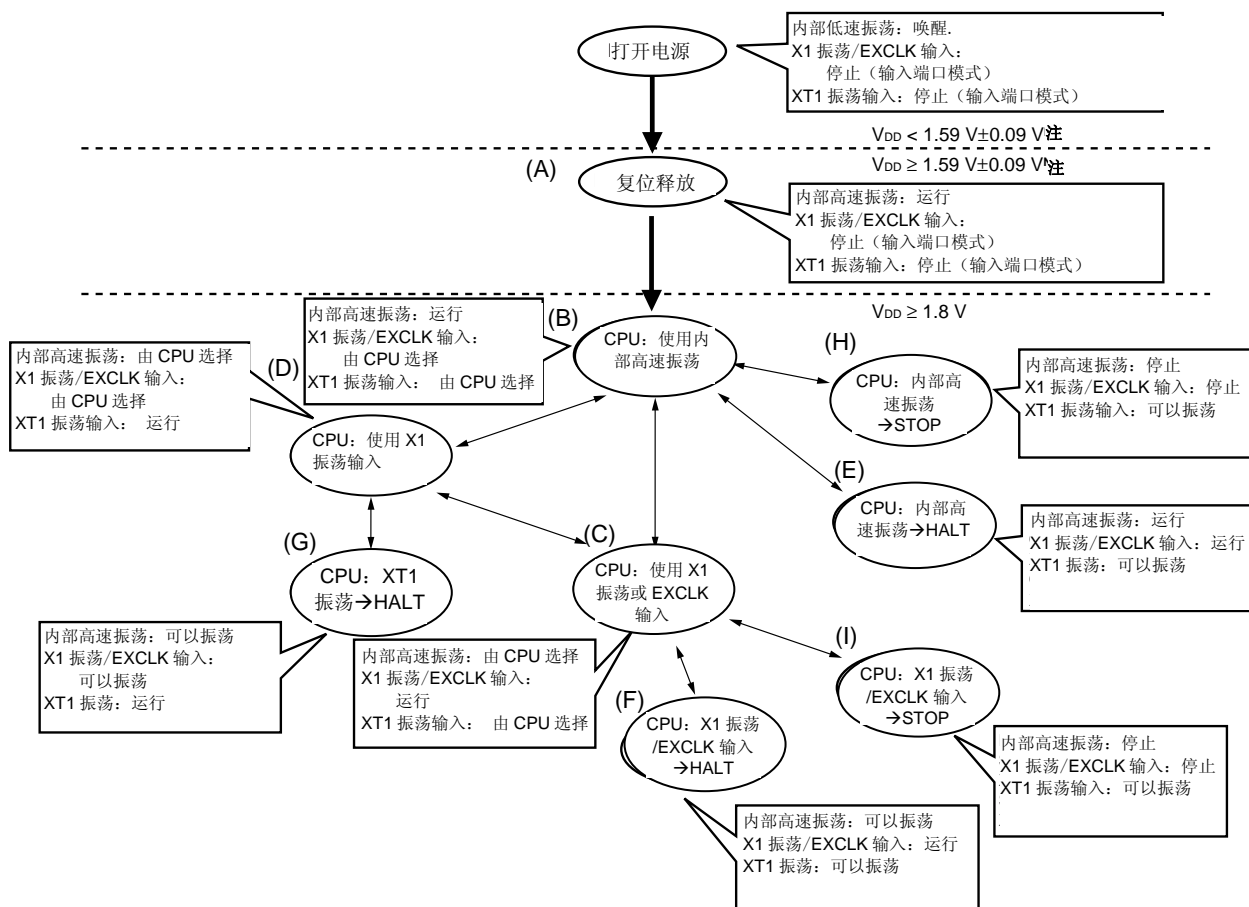
如下情况内部低速振荡时钟可被重启。

- 释放 HALT 或 STOP 模式
（只有当通过选项字节（000C0H 的第 0 位（WDSTBYON）= 0）在 HALT/STOP 模式下停止看门狗定时器，以及执行 HALT 或 STOP 指令使看门狗定时器停止时）。

6.5.5 CPU 时钟状态迁移图

图 5-15 显示了该产品 CPU 时钟状态迁移图。

图 5-15. CPU 时钟状态迁移图



注 初值可能会改变。

备注 如果低电压检测电路 (LVI) 通过选项字节默认设置为开启, 直到电源电压 (V_{DD}) 超过 $2.07 V \pm 0.2 V$ 复位信号才会被释放^注。

在复位处理后, 状态将会迁移到上图的 (B)。

表 5-4 显示了 CPU 时钟的切换过程与 SFR 寄存器设置示例。

表 5-4. CPU 时钟切换与 SFR 寄存器设置示例 (1/4)

(1) 复位释放后 (A) CPU 使用内部高速振荡时钟 (B)

状态迁移	SFR 寄存器设置
(A) → (B)	SFR 寄存器不必设置 (复位释放后默认状态)。

(2) 复位释放后 (A) CPU 使用高速系统时钟 (C)

(复位释放后 CPU 立即使用内部高速振荡时钟 (B)。。)

(SFR 寄存器设置序列)

设置 SFR 寄存器的标志 状态迁移	CMC 寄存器 ^{注1}			CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSSEL	AMPH	MSTOP	FSEL		MCM0
(A) → (B) → (C) (X1 时钟: 2 MHz ≤ f _x ≤ 10 MHz)	0	1	0	0	0	必须检查	1
(A) → (B) → (C) (X1 时钟: 10 MHz < f _x ≤ 20 MHz)	0	1	1	0	1 ^{注2}	必须检查	1
(A) → (B) → (C) (外部主时钟)	1	1	0/1	0	0/1	不能检查	1

注 1. 当复位释放后, CMC 和 OSMC 寄存器只能写入一次, 通过 8 位存储器操作指令写入。

2. 当 f_{CLK} > 10 MHz 时, FSEL = 1

如果选择分频时钟并且 f_{CLK} ≤ 10 MHz, 即使 f_x > 10 MHz 也可使用 FSEL = 0。

注意事项 供电电压达到所选时钟的可操作电压后, 设置时钟 (参见 第二十八章 电气特性)。

(3) 复位释放后 (A) CPU 使用副系统时钟 (D)

(复位释放后 CPU 立即使用内部高速振荡时钟 (B)。。)

(SFR 寄存器设置序列)

设置 SFR 寄存器的标志 状态迁移	CMC 寄存器 ^注	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCSLS	XTSTOP		CSS
(A) → (B) → (D)	1	0	需要	1

注 当复位释放后, CMC 寄存器只能写入一次, 通过 8 位存储器操作指令写入。

备注 表 5-4 中的 (A) ~ (I) 对应图 5-15 中的 (A) ~ (I)。

表 5-4. CPU 时钟切换与 SFR 寄存器设置示例 (2/4)

(4) CPU 时钟从内部高速振荡时钟 (B) 切换到高速系统时钟 (C)

(SFR 寄存器设置序列) →

状态迁移 / SFR 寄存器的设置标志	CMC 寄存器 ^{注1}			OSTS 寄存器	CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 时钟: 2 MHz ≤ f _X ≤ 10 MHz)	0	1	0	注 2	0	0	必须检查	1
(B) → (C) (X1 时钟: 10 MHz < f _X ≤ 20 MHz)	0	1	1	注 2	0	1 ^{注3}	必须检查	1
(B) → (C) (外部主时钟)	1	1	0/1	注 2	0	0/1	不能检查	1

如果这些寄存器已经设置则不需要

如果 CPU 使用高速系统时钟则不必要

- 注**
1. 当复位释放后, CMC 和 OSMC 寄存器只能写入一次, 如果已经设置了该项, 则无需再设置。
 2. 设置振荡稳定时间如下。
 - 需要的 OSTC 振荡稳定时间 ≤ 由 OSTS 设置的振荡稳定时间
 3. 当 f_{CLK} > 10 MHz, FSEL = 1
如果选择分频时钟并且 f_{CLK} ≤ 10 MHz, 即使 f_X > 10 MHz 也可使用 FSEL = 0。

注意事项 供电电压达到所选时钟的可操作电压后, 设置时钟 (参见 第二十八章 电气特性)。

(5) CPU 时钟从内部高速振荡时钟 (B) 切换到副系统时钟 (D)

(SFR 寄存器设置序列) →

状态迁移 / 设置 SFR 寄存器标志	CMC 寄存器 ^注	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCSLS	XTSTOP		CSS
(B) → (D)	1	0	需要	1

如果 CPU 工作在副系统时钟, 则不需要

注 当复位释放后, CMC 寄存器只能写入一次, 通过 8 位存储器操作指令写入。

备注 表 5-4 中的 (A) ~ (I) 对应图 5-15 中的 (A) ~ (I)。

表 5-4. CPU 时钟切换与 SFR 寄存器设置示例 (3/4)

(6) CPU 时钟从高速系统时钟 (C) 切换到内部高速振荡时钟 (B)

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志 状态迁移	CSC 寄存器	振荡精确稳定时间	CKC 寄存器
	HIOSTOP		MCM0
(C) → (B)	0	10 μs	0

如果 CPU 工作在内部高速时钟, 则不需要

(7) CPU 时钟从高速系统时钟 (C) 切换到副系统时钟 (D)

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^注	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCSELS	XTSTOP		CSS
(C) → (D)	1	0	需要	1

如果 CPU 使用副系统时钟, 则不必要

注 当复位释放后, CMC 寄存器只能写入一次, 通过 8 位存储器操作指令写入。

(8) CPU 时钟从副系统时钟 (D) 切换到内部高速振荡时钟 (B)

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志 状态迁移	CSC 寄存器	CKC 寄存器	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

如果 CPU 使用内部高速振荡时钟, 则不必要

如果这些寄存器已经设置, 则不需要

表 5-4. CPU 时钟切换与 SFR 寄存器设置示例 (4/4)

(9) CPU 时钟从副系统时钟 (D) 切换到高速系统时钟 (C)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^{#1}			OSTS 寄存器	CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器	
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 时钟: 2 MHz ≤ f _x ≤ 10 MHz)	0	1	0	注 2	0	0	必须检查	1	0
(D) → (C) (X1 时钟: 10 MHz < f _x ≤ 20 MHz)	0	1	1	注 2	0	1 ^{#3}	必须检查	1	0
(D) → (C) (外部主时钟)	1	1	0/1	注 2	0	0/1	不能检查	1	0

如果这些寄存器已经设置, 则不需要 如果 CPU 使用高速系统时钟, 则不必要 如果这些寄存器已经设置, 则不需要

- 注
1. 当复位释放后, CMC 和 OSMC 寄存器只能写入一次, 如果已经设置了该项, 则无需再设置。
 2. 设置振荡稳定时间如下。
 - 需要的 OSTC 振荡稳定时间 ≤ 由 OSTS 设置的振荡稳定时间
 3. 当 f_{CLK} > 10 MHz, FSEL = 1
如果选择分频时钟并且 f_{CLK} ≤ 10 MHz, 即使 f_x > 10 MHz 也可使用 FSEL = 0。

注意事项 供电电压达到所选时钟的可操作电压后, 设置时钟 (参见 第二十八章 电气特性)。

备注 表 5-4 中的 (A) ~ (I) 对应图 5-15 中的 (A) ~ (I)。

- (10)
- CPU 使用内部高速振荡时钟 (B) 时 HALT 模式 (E) 的设置。
 - CPU 使用高速系统时钟 (C) 时 HALT 模式 (F) 的设置。
 - CPU 使用副系统时钟 (D) 时 HALT 模式 (G) 的设置。

状态迁移	设置
(B) → (E) (C) → (F) (D) → (G)	执行 HALT 指令

- (11)
- CPU 使用内部高速振荡时钟 (B) 时 STOP 模式 (H) 的设置。
 - CPU 使用高速系统时钟 (C) 时 STOP 模式 (I) 的设置。

(设置顺序)

状态迁移		设置		
(B) → (H)	X1 停止	停止那些不能在 STOP 模式下使用的外部功能	-	执行 STOP 指令
	X1 振荡		设置 OSTS 寄存器	
(C) → (I)				

备注 表 5-4 中的 (A) ~ (I) 对应图 5-15 中的 (A) ~ (I)。

5.6.6 CPU时钟切换之前的条件与切换之后的处理

CPU 时钟切换之前的条件与切换之后的处理显示如下。

表 5-5. 切换 CPU 时钟

CPU 时钟		切换之前的条件	切换后的处理
切换前	切换后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间	停止内部高速振荡器 (HIOSTOP = 1) 可以降低操作电流。
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	副系统时钟	X1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	
X1 时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	X1 振荡停止 (MSTOP = 1) .
	外部主系统时钟	该切换不允许 (要切换时钟, 执行复位后再设置它)	-
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	X1 振荡停止 (MSTOP = 1) .
外部主系统时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	外部主系统时钟输入禁止 (MSTOP = 1) .
	X1 时钟	切换不允许 (要切换时钟, 执行复位后再设置它)	-
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	外部主系统时钟输入禁止 (MSTOP = 1) .
副系统时钟	内部高速振荡时钟	内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟 • HIOSTOP = 0, MCS = 0	XT1 振荡可以停止 (XTSTOP = 1)
	X1 时钟	X1 振荡稳定且选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间 • MCS = 1	
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 并选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	

5.6.7 CPU时钟和主系统时钟切换所需的时间

通过设置系统时钟控制寄存器（CKC）的第 0 位到第 2 位、第 4 位以及第 6 位（MDIV0~ MDIV2, MCM0, CSS），可以切换 CPU 时钟（为主系统时钟或副系统时钟），可以切换主系统时钟（为内部高速振荡时钟或高速系统时钟），同时可以改变主系统时钟的分频系数。

修改 CKC 后，实际的切换操作不会立即执行；会使用切换前的时钟继续操作几个时钟（见表 5-6 ~ 表 5-9）。

可以通过 CKC 寄存器的第 7 位（CLS）来确定 CPU 使用主系统时钟还是副系统时钟。可以通过 CKC 寄存器的第 5 位（MCS）来确定主系统时钟是高速系统时钟 还是内部高速振荡时钟。

当切换 CPU 时钟时，外围硬件时钟也同样被切换。

表 5-6. 主系统时钟切换需要的最长时间

时钟 A	切换方法	时钟 B	类型
f _{IH}	↔	f _{MX}	类型 2（参见表 5-8）
f _{MAIN}	↔	f _{SUB}	类型 3（参见表 5-9）
f _{MAIN}	↔ (改变分频系数)	f _{MAIN}	类型 1（参见表 5-7）
f _{SUB}	↔ (改变分频系数)	f _{SUB}	类型 1（参见表 5-7）

表 5-7. 类型 1 中需要的最大时钟数量

切换前的设置值	切换后的设置值	
	时钟 A	时钟 B
时钟 A		1 + f _A /f _B 时钟
时钟 B	1 + f _B /f _A 时钟	

表 5-8. 类型 2 中需要的最大时钟数量

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}		1 + f _{MX} /f _{IH} 时钟
	f _{MX} < f _{IH}		2 f _{IH} /f _{MX} 时钟
1 (f _{MAIN} = f _{MX})	f _{MX} > f _{IH}	2 f _{MX} /f _{IH} 时钟	
	f _{MX} < f _{IH}	1 + f _{MX} /f _{IH} 时钟	

（备注在下页列出）

表 5-9. 类型 3 需要的最大时钟数量

切换前的设置值		切换后的设置值	
CSS		CSS	
		0 (fCLK = fMAIN)	1 (fCLK = fSUB)
0 (fCLK = fMAIN)	fMAIN < fSUB		2 + fMAIN/fSUB 时钟
	fMAIN > fSUB		1 + 2 fMAIN/fSUB 时钟
1 (fCLK = fSUB)	fMAIN < fSUB	1 + 2 fSUB/f _{主时钟}	
	fMAIN > fSUB	2 + fSUB/f _{主时钟}	

- 备注**
1. 表 5-7 ~ 表 5-9 中列出的时钟数是切换前的 CPU 时钟数量。
 2. 计算表 5-7 ~ 表 5-9 的时钟数量时，舍去小数部分。

示例： 将主系统时钟从内部高速振荡时钟切换到高速系统时钟（@振荡频率 f_{RH} = 8 MHz, f_{XH} = 10 MHz）

$$1 + f_{RH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2 \text{ 时钟}$$

5.6.8 时钟振荡停止前的条件

下表列出了停止钟振荡的寄存器标志位设置（禁止外部时钟输入）和时钟振荡停止前的条件。

表 5-10. 时钟振荡停止前的条件及标志位设置

时钟	时钟振荡停止前的条件 (禁止外部时钟输入)	SFR 寄存器的标志位 设置
内部高速振荡时钟	MCS = 1 或 CLS = 1 (CPU 使用的操作时钟不是内部高速振荡时钟)	HIOSTOP = 1
X1 时钟	MCS = 0 或 CLS = 1 (CPU 使用的操作时钟不是高速系统时钟)	MSTOP = 1
外部主系统时钟		
副系统时钟	CLS = 0 (CPU 使用的操作时钟不是副系统时钟)	XTSTOP = 1

第六章 定时器阵列单元

每个定时器阵列单元有 8 个 16 位定时器。每个 16 位定时器叫一个通道，并且可以作为一个独立定时器。另外，2 个或更多“通道”可以组合使用，用作高精度定时器。

独立操作功能	组合操作功能
<ul style="list-style-type: none">• 间隔定时器• 方波输出• 外部事件计数器• 分频功能• 输入脉冲间隔测量• 输入信号的高/低电平宽度测量	<ul style="list-style-type: none">• PWM 输出• 单脉冲输出• 多路 PWM 输出

通道 7 与串行阵列单元 1 的 UART3 可以组合实现 LIN 总线接收处理。

6.1 定时器阵列单元的功能

定时器阵列单元有如下功能。

6.1.1 独立操作时每通道的功能

无论其他通道的操作模式任何通道可使用独立操作功能（详细情况，参考 6.6.1 独立操作功能和组合操作功能的概述）。

(1) 间隔定时器

一个单元的每个定时器可用作产生固定间隔的中断（INTTM0n）的参考定时器。

(2) 方波输出

每次执行反转操作，产生 INTTM0n，并且从定时器输出引脚（TO0n）输出占空比为 50% 的方波。

(3) 外部事件计数器

从定时器输入引脚（TI0n）输入的信号有效沿的数量达到指定数量时，一个单元的每个定时器都可用作产生中断的事件计数器。

(4) 分频功能

从定时器输入引脚（TI0n）输入的时钟被分频，并且从输出引脚（TO0n）输出。

(5) 输入脉冲间隔测量

在定时器输入引脚（TI0n）输入脉冲信号的有效沿开始计数。定时器的计数值在下一个脉冲有效沿被捕捉。这样，可以测量输入脉冲的间隔。

(6) 输入信号高/低电平宽度测量

由输入到定时器输入引脚（TI0n）的信号的单侧边沿开始计数，并且在其他沿捕捉计数值。这样，可以测量输入信号的高电平或低电平宽度。

备注 n: 通道编号 (n = 0 ~ 7)

6.1.2 和其他通道操作时每个通道的功能

组合操作功能是主通道（控制周期的主要参考定时器）和从通道（随主通道操作的定时器）组合一起操作（更多细节，参考 6.6.1 独立操作功能和组合操作功能的概述）。

(1) PWM（脉冲宽度调制器）输出

两个通道组合用来产生指定周期和指定占空比的脉冲。

(2) 单脉冲输出

两个通道用作设置产生指定延迟时间和指定脉冲宽度的单脉冲。

(3) 多路 PWM（脉冲宽度调制器）输出

通过扩展 PWM 功能，使用一个主通道和两个以上从通道，可以产生七种类型的 PWM 信号，并可以指定周期和占空比。

6.1.3 LIN总线支持功能（仅限通道 7）

(1) 唤醒信号的检测

定时器在输入到 UART3 的串行数据输入引脚（RxD3）的信号的下沿开始计数。这样，低电平宽度可以测量。如果低电平宽度大于指定值，则认为是一个唤醒信号。

(2) 同步中断域检测

在检测到唤醒信号后，定时器在输入到 UART3 的串行数据输入引脚（RxD3）的信号的下沿开始计数，并且在上升沿处捕捉定时器计数值。这样，可以测量低电平宽度。如果低电平宽度大于指定值，则认为同步中断域。

(3) 同步域的脉冲宽度测量

检测到同步中断域后，在输入到 UART3 的串行数据输入引脚（RxD3）的信号的低电平宽度和高电平宽度可测量到。这样通过测量到的同步域的位间隔，可计算出波特率。

6.2 定时器阵列单元的配置

定时器阵列单元包括如下硬件。

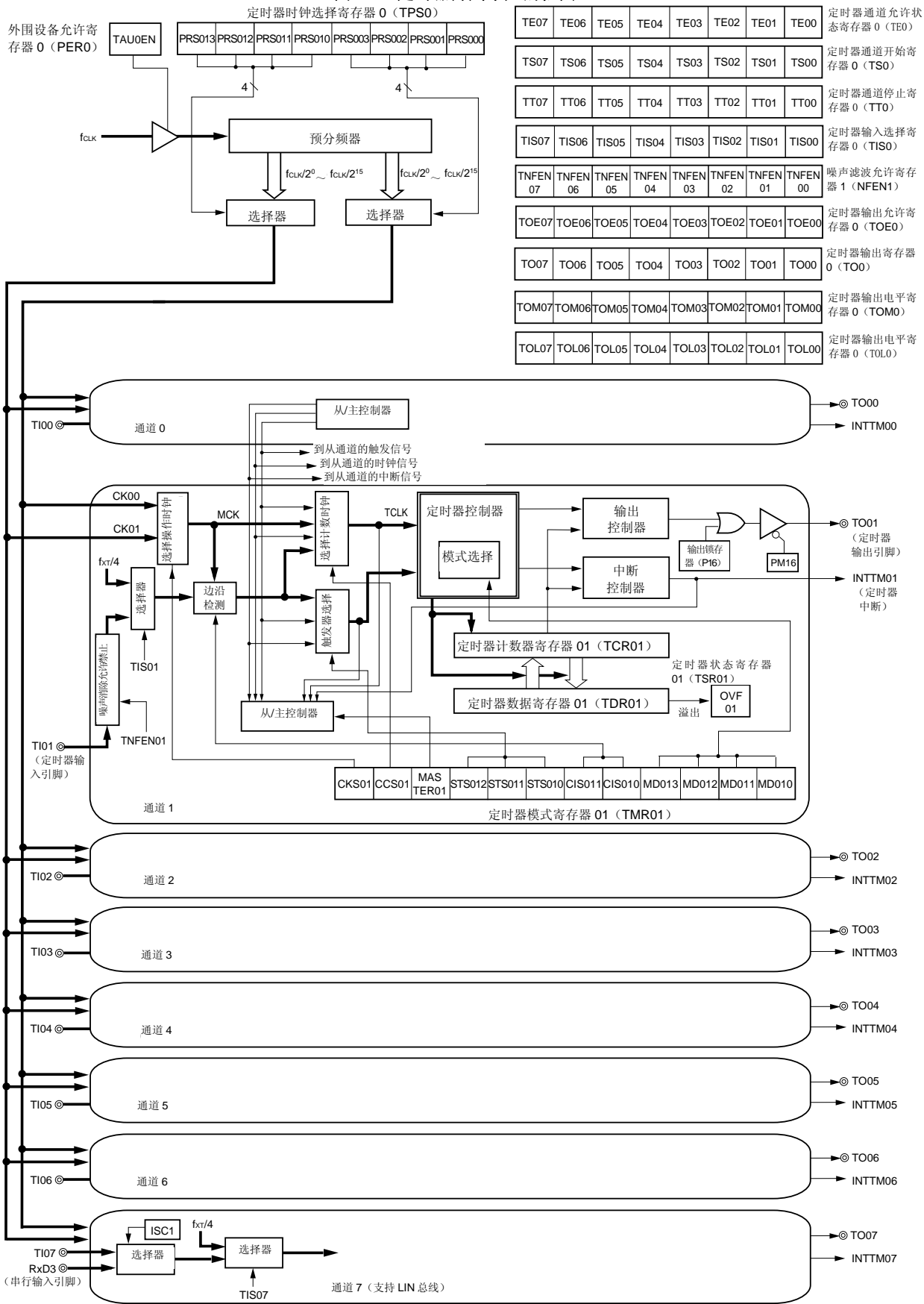
表 6-1. 定时器阵列单元的配置

项目	配置
定时器/计数器	定时器 计数器 寄存器 0n (TCR0n)
寄存器	定时器数据寄存器 0n (TDR0n)
定时器输入	TI00~TI07 引脚, RxD3 引脚 (对于 LIN 总线)
定时器输出	TO00~TO07 引脚, 输出控制器
控制寄存器	<p><单位设置 block 的寄存器></p> <ul style="list-style-type: none"> • 外围设备允许寄存器 0 (PER0) • 定时器时钟选择寄存器 0 (TPS0) • 定时器通道允许状态寄存器 0 (TE0) • 定时器通道启动寄存器 0 (TS0) • 定时器通道停止寄存器 0 (TT0) • 定时器输入选择寄存器 0 (TIS0) • 定时器输出允许寄存器 0 (TOE0) • 定时器输出寄存器 0 (TO0) • 定时器输出电平寄存器 0 (TOL0) • 定时器输出模式寄存器 0 (TOM0) <p><每个通道的寄存器></p> <ul style="list-style-type: none"> • 定时器模式寄存器 0n (TMR0n) • 定时器状态寄存器 0n (TSR0n) • 输入切换控制寄存器 (ISC) (仅对通道 7 有效) • 噪声滤波允许寄存器 1 (NFEN1) • 端口模式寄存器 0, 1, 3, 4, 14 (PM0, PM1, PM3, PM4, PM14) • 端口寄存器 0, 1, 3, 4, 14 (P0, P1, P3, P4, P14)

备注 n: 通道编号 (n = 0~7)

图 6-1 所示为框图。

图 6-1. 定时器阵列单元的框图



TCR0n 寄存器 读取值根据如下操作模式改变和操作状态的不同而不同。

表 6-2. TCR0n 寄存器在不同操作模式下的读取值

操作模式	计数模式	TCR0n 寄存器读取值 ^注			
		复位后操作模式改变	计数操作终止后操作模式改变 (TT0n = 1)	计数操作终止后重启 (TT0n = 1)	单计数后启动触发期间等待状态
间隔定时器 模式	计数减	FFFFH	不确定	操作终止时的值	-
捕捉模式	计数加	0000H	不确定	操作终止时的值	-
事件计数器 模式	计数减	FFFFH	不确定	操作终止时的值	-
单计数模式	计数减	FFFFH	不确定	操作终止时的值	FFFFH
捕捉 & 单计数模式	计数加	0000H	不确定	操作终止时的值	TDR0n 寄存器的捕捉值 + 1

注 当 TE0n = 0 时 TS0n 被设置为"1"后 TCR0n 寄存器的读取值为所示。读取值保存在 TCR0n 寄存器知道计数操作开始。

备注 n = 0~7

(2) 定时器数据寄存器 0n (TDR0n)

此寄存器为 16 位寄存器，可选择捕捉功能和比较功能。

通过使用 TMR0n 的 MD0n3~MD0n0 位选择操作模式可切换捕捉或比较功能。

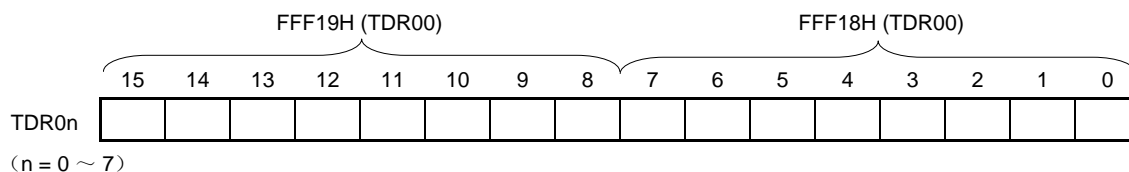
TDR0n 的值任何时间都可改变。

此寄存器可以 16 位为单位读或写。

复位信号产生将寄存器清为 0000H。

图 6-3. 定时器数据寄存器 0n (TDR0n) 的格式

地址: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), 复位后: 0000H R/W
FFF64H, FFF65H (TDR02) ~ FFF6EH, FFF6FH (TDR07)

**(i) TDR0n 用作比较寄存器时**

从设置到 TDR0n 的值开始向下计数。当计数值达到 0000H 时，产生一个中断信号 (INTTM0n)。

TDR0n 的值保持到重写。

注意事项 当设置为比较功能时，即使捕捉出发输入 TDR0n 也不能执行捕捉操作。

(ii) TDR0n 用作捕捉寄存器时

当捕捉出发输入时 TCR0n 的计数值被捕捉到 TDR0n。

TI0n 引脚的有效沿可选择作为捕捉触发。此选择由 TMR0n 进行。

备注 n = 0 ~ 7

6.3 控制定时器阵列单元的寄存器

定时器阵列单元由如下寄存器控制。

- 外围设备允许寄存器 0 (PER0)
- 定时器时钟选择寄存器 0 (TPS0)
- 定时器模式寄存器 0n (TMR0n)
- 定时器状态寄存器 0n (TSR0n)
- 定时器通道允许状态寄存器 0 (TE0)
- 定时器通道启动寄存器 0 (TS0)
- 定时器通道停止寄存器 0 (TT0)
- 定时器输入选择寄存器 0 (TIS0)
- 定时器输出 允许寄存器 0 (TOE0)
- 定时器输出 寄存器 0 (TO0)
- 定时器输出电平寄存器 0 (TOL0)
- 定时器输出模式寄存器 0 (TOM0)
- 输入切换控制寄存器 (ISC)
- 噪声滤波允许寄存器 1 (NFEN1)
- 端口模式寄存器 0, 1, 3, 4, 14 (PM0, PM1, PM3, PM4, PM14)
- 端口寄存器 0, 1, 3, 4, 14 (P0, P1, P3, P4, P14)

备注 n = 0 ~ 7

(1) 外围设备允许寄存器 0 (PER0)

PER0 用于每个外围模块的允许或禁止使用。为了降低功耗和噪声对于不使用的外围模块停止供给时钟。

使用定时器阵列单元时，确保设置此寄存器的第 0 位 (TAU0EN) 为 1。

PER0 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 00H。

注意事项 1. 设置定时器阵列单元时，首先确保设置 TAU0EN 为 1。如果 TAU0EN = 0，定时器阵列单元的控制寄存器的写入被忽略，并且所有读取的值都是默认值。

2. 确保将 PER0 寄存器的第 1 位清为 0。

图 6-4. 外围设备允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	定时器阵列单元输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> 由定时器阵列单元使用的 SFR 不能写入。 定时器阵列单元在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> 由定时器阵列单元使用的 SFR 可读/写。

(2) 定时器时钟选择寄存器 0 (TPS0)

TPS0 是 16 位寄存器，用于通常供给每个通道的操作时钟 (CK00, CK01) 的两种类型的选择。CK01 由 TPS0 的第 7~4 位选择，CK00 通过将第 3 位设置为 0 选择。

定时器操作期间只有在如下情况下可以重写 TPS0。

PRS000~ PRS003 位重写: 只有在操作停止状态 (TE0n = 0) 下所有通道设置为 CKS0n = 0 时可能

PRS010 ~ PRS013 位重写: 只有在操作停止状态 (TE0n = 0) 下所有通道设置为 CKS0n = 1 时可能

TPS0 可由 16 位存储器操作指令设置。

TPS0 的低 8 位和 TPS0L 可以由 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-5. 定时器时钟选择寄存器 0 (TPS0) 的格式

地址: F01B6H, F01B7H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0m3	PRS 0m2	PRS 0m1	PRS 0m0		操作时钟 (CK0m) ^注 的选择			
					f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	f _{CLK} /2 ⁷	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	f _{CLK} /2 ¹²	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	f _{CLK} /2 ¹³	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	f _{CLK} /2 ¹⁴	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	f _{CLK} /2 ¹⁵	61 Hz	153 Hz	305 Hz	610 Hz

<R> 注 改变为 f_{CLK} (由系统时钟控制寄存器 (CKC) 的值改变) 选择的时钟时, 停止定时器阵列单元 (TT0 = 00FFH)。

注意事项 确保将第 15~ 8 位清为“0”。

备注 1. f_{CLK}: CPU/外围硬件时钟频率
2. m = 0, 1 n = 0~7

(3) 定时器模式寄存器 0n (TMR0n)

TMR0n 设置通道 n 的操作模式。它用于选择操作时钟 (MCK)，计数时钟，定时器作为主或从操作，开始触发和捕捉出发，定时器输入的有效沿，和操作模式 (间隔，捕捉，事件计数器，单计数，或捕捉&单计数)。

当寄存器在操作时 (when TE0 = 1) 禁止重写 TMR0n。但是，第 7 和 6 位 (CIS0n1, CIS0n0) 即使当寄存器一些功能操作时也可以被重写 (TE0 = 1) (详细情况，参考 6.7 作为独立通道的定时器阵列单元的操作和 6.8 定时器阵列单元的复用通道的操作)。

TMR0n 可由 16 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-6. 定时器模式寄存器 0n (TMR0n) 的格式 (1/3)

地址: F0190H, F0191H (TMR00) ~ F019EH, F019FH (TMR07) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CKS 0n	通道 n 的操作时钟 (MCK) 的选择
0	由 PRS 寄存器设置时钟 CK00 操作
1	由 PRS 寄存器设置时钟 CK01 操作
操作时钟 MCK 用于沿检测器。根据 CCS0n 位的设置，一个计数时钟 (TCLK) 产生根据 CCS0n 位的设置。	

CCS 0n	通道 n 的计数时钟 (TCLK) 选择
0	操作时钟 MCK 由 CKS0n 位指定
1	从 TI0n 引脚输入的输入信号的有效沿
计数时钟 TCLK 用于定时器/计数器，输出控制器，和中断控制器。	

MASTER0n	为通道 n 选择独立操作功能或在组合操作功能中作为从通道 / 在组合操作功能中作为主通道
0	独立操作功能或在组合操作功能中作为从通道。
1	组合操作功能中作为主通道操作。
只有偶数通道可以被设置为主通道 (MASTER0n = 1)。奇数通道一定要设置为从通道使用 (MASTER0n = 0)。对于作为独立操作功能使用的通道，MASTER0n 清为 0。	

注意事项 确保设置第 14, 13, 5,和 4 位为“0”。

备注 n = 0~7

图 6-6. 定时器模式寄存器 0n (TMR0n) 的格式 (2/3)

地址: F0190H, F0191H (TMR00) ~ F019EH, F019FH (TMR07) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

STS 0n2	STS 0n1	STS 0n0	通道 n 的开始触发或捕捉触发
0	0	0	只有软件触发开始有效 (其他触发源不选择)。
0	0	1	TI0n 引脚输入 的有效沿用作开始触发和捕捉触发。
0	1	0	TI0n 引脚输入 的双沿用作开始触发和捕捉触发。
1	0	0	使用主通道的中断信号 (组合操作功能中当通道用作从通道时)。
其他值			禁止设置

CIS 0n1	CIS 0n0	TI0n 引脚输入有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	双沿 (当低电平宽度被测量时) 开始触发: 下降沿, 捕捉触发: 上升沿
1	1	双沿 (当高电平宽度被测量时) 开始触发: 上升沿, 捕捉触发: 下降沿
当 STS0n2 ~ STS0n0 位的值不是 010B 时如果指定双沿, 设置 CIS0n1 ~ CIS0n0 位为 10B。		

备注 n = 0~7

图 6-6. 定时器模式寄存器 0n (TMR0n) 的格式 (3/3)

地址: F0190H, F0191H (TMR00) - F019EH, F019FH (TMR07) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

MD 0n3	MD 0n2	MD 0n1	MD 0n0	通道 n 的操作模式	TCR 的计数操作	独立操作
0	0	0	1/0	间隔定时器模式	向下计数	可能
0	1	0	1/0	捕捉模式	向上计数	可能
0	1	1	0	事件计数器模式	向下计数	可能
1	0	0	1/0	单计数模式	向下计数	不可能
1	1	0	0	捕捉 & 单计数模式	向上计数	可能
其他值				设置禁止		
MD0n0 位的操作根据每种操作模式的不同而不同 (参见下表)。						

操作模式 (由 MD0n3~MD0n1 位设置的值 (参见上表))	MD 0n0	开始计数和中断的设置
• 间隔定时器模式 (0, 0, 0)	0	当计数开始时定时器 中断不产生 (定时器输出也不改变)。
• 捕捉模式 (0, 1, 0)	1	当计数开始时定时器 中断产生 (定时器输出也改变)。
• Event 计数器模式 (0, 1, 1)	0	当计数开始时定时器 中断不产生 (定时器输出也不改变)。
• 单计数模式 (1, 0, 0)	0	在计数操作期间开始触发无效。 同时, 中断也不产生。
	1	在计数操作期间开始触发有效 ^注 。 同时, 中断也产生。
• 捕捉 & 单计数模式 (1, 1, 0)	0	当计数开始时定时器 中断不产生 (定时器输出也不改变)。 在计数操作期间开始触发无效。 同时, 中断也不产生。
其他值		设置禁止

注 如果操作期间开始触发 (TS0n = 1) 被发出, 计数器被清零, 中断产生, 开始重新计数。

备注 n = 0~7

(4) 定时器状态寄存器 0n (TSR0n)

TSR0n 指示通道 n 的计数器的溢出状态。

TSR0n 只有在捕捉模式 (MD0n3~MD0n1 = 010B) 和捕捉& 单计数模式 (MD0n3~ MD0n1 = 110B) 模式下有效。在其他模式下不进行设置。对于 OVF 位在每种操作模式下的操作和设置/清零条件参见表 6-3。

TSR0n 可由 16 位存储器操作指令读取。

TSR0n 的低 8 位和 TSR0nL 可以由 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-7. 定时器状态寄存器 0n (TSR0n) 的格式

地址: F01A0H, F01A1H (TSR00) ~F01AEH, F01AFH (TSR07) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	通道 n 的计数器溢出状态
0	溢出不发生。
1	溢出发生。
OVF = 1 时, 如果当下一个被捕捉到的值没有溢出, 此标志被清零 (OVF = 0)。	

<R>

表 6-3. 在每种操作模式下 OVF 位操作和设置/清零条件

定时器操作模式	OVF	置位/清零条件
• 捕捉模式	清零	当捕捉时没有溢出发生
• 捕捉& 单计数模式	置位	当捕捉时溢出已经发生
• 间隔定时器模式	清零	— (使用禁止, 不置位也不清零)
• 事件计数器模式	置位	
• 单计数模式		

<R>

备注 OVF 位在计数器溢出以后不立即改变, 而是在随后的捕捉时改变。

(5) 定时器通道允许状态寄存器 0 (TE0)

TE0 用于允许或停止每个通道的定时器操作。

定时器通道启动寄存器 0 (TS0) 的一个位被设置为 1 时，此寄存器的相应位被设置为 1。定时器通道停止寄存器 0 (TT0) 的位设置为 1 时，此寄存器的相应位被清为 0。

TE0 可由 16 位存储器操作指令读取。

TE0 的低 8 位和 TEOL 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-8. 定时器通道允许状态寄存器 0 (TE0) 的格式

地址: F01B0H, F01B1H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TE0n	通道 n 的操作允许/停止状态的指示
0	操作停止
1	操作允许

备注 n = 0 ~ 7

(6) 定时器通道启动寄存器 0 (TS0)

TS0 是用于清零定时器计数器 (TCR0n) 和启动每个通道的计数操作的触发寄存器。

此寄存器的一个位 (TS0n) 设置为 1 时, 定时器通道允许状态寄存器 0 (TE0) 的相应位 (TE0n) 设置为 1。TE0n = 1 时 TS0n 是触发位并且立即清零。

TS0 可由 16 位存储器操作指令设置。

TS0 的低 8 位和 TS0L 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-9. 定时器通道启动寄存器 0 (TS0) 的格式

地址: F01B2H, F01B3H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TS0n	通道 n 的操作允许 (开始) 触发
0	无触发操作
1	TE0n 设置为 1 并且计数操作允许。 TCR0n 在计数操作允许状态的计数操作根据每个操作模式的不同而不同 (参见表 6-4)。

注意事项 确保将第 15~8 位为“0”。

备注 1. 读 TS0 寄存器时, 一直读取 0。

2. n = 0~7

表 6-4. 从计数操作允许状态到 TCR0n 计数开始的操作 (1/2)

定时器操作模式	当 TS0n = 1 时的操作
• 间隔定时器 模式	从开始触发检测 (TS0n=1) 直到计数时钟发生没有操作执行。 第一个计数时钟将 TDR0n 的值加载到 TCR0n, 后面的计数时钟执行向下计数操作 (参见 6.3 (6) (a) 间隔定时器 模式下的开始时序)。
• 事件计数器 模式	TS0n 位写入 1, 加载 TDR0n 的值到 TCR0n。 后面的计数时钟执行向下计数操作。 外部触发检测由 STS0n2~ STS0n0 位检测在 TMR0n 寄存器中不开始计数操作 (参见 6.3 (6) (b) 事件 计数器 模式下的开始时序)。
• 捕捉模式	从开始触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 0000H 到 TCR0n, 后面的计数时钟执行向上计数操作 (参见 6.3 (6) (c) 捕捉模式下的开始时序)。

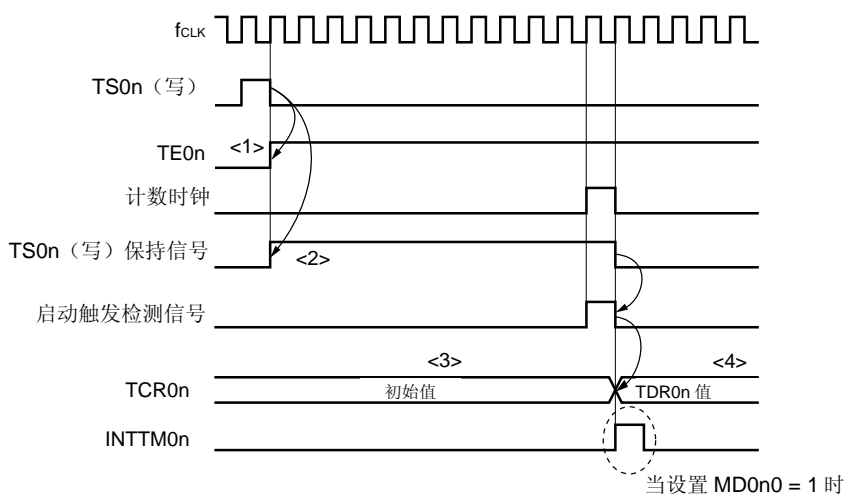
表 6-4. 从计数操作允许状态到 TCR0n 计数开始的操作 (2/2)

定时器操作模式	设置 TS0n = 1 时的操作
• 单计数 模式	TS0n = 0 时, 写入 1 到 TS0n 位设置开始触发等待状态。 从开始触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 TDR0n 的值到 TCR0n, 后面的计数时钟执行向下计数操作 (参见 6.3 (6) (d) 单计数 模式下的开始时序)。
• 捕捉& 单计数 模式	TS0n = 0 时, 写入 1 到 TS0n 位设置开始触发等待状态。 从开始触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 0000H 到 TCR0n, 后面的计数时钟执行向上计数操作 (参见 6.3 (6) (e) 捕捉& 单计数 模式下的开始时序)。

(a) 间隔定时器模式下的开始时序

- <1> 写入 1 到 TS0n 设置 TE0n = 1
- <2> TS0n 的写入数据一直保持到计数时钟产生。
- <3> TCR0n 的初始值一直保持到计数时钟产生。
- <4> 计数时钟产生, “TDR0n 值”被加载到 TCR0n 并且计数开始。

图 6-10. 开始时序 (在间隔定时器模式下)

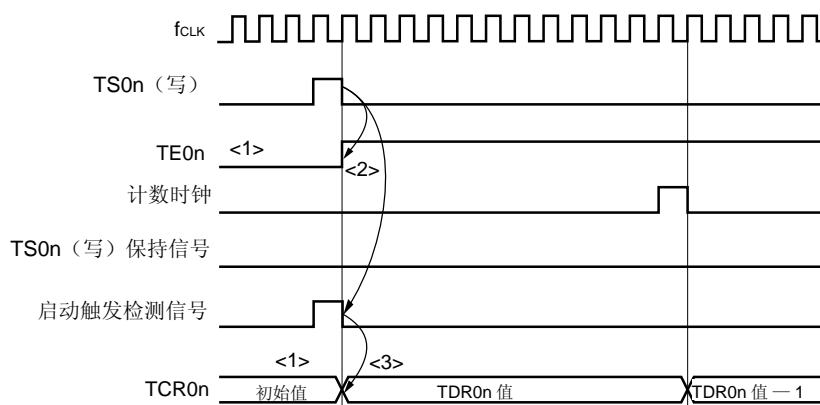


注意事项 TS0n 写入后在计数时钟的第一个周期操作中, 由于直到产生计数时钟开始计数, 产生最大一个时钟的误差。当在计数开始时序的信息需要时, 通过设置 MD0n0 = 1 可在计数开始产生中断。

(b) 事件计数器模式下的开始时序

- <1> TE0n 设置为 0 时, TCR0n 保持初始值。
- <2> 写入 1 到 TS0n 设置 TE0n 为 1。
- <3> 一旦 TS0n 写入 1 并且设置 TE0n 为 1, "TDR0n 值"就被加载到 TCR0n 开始计数。
- <4> 然后, TCR0n 的值根据计数时钟向下计数。

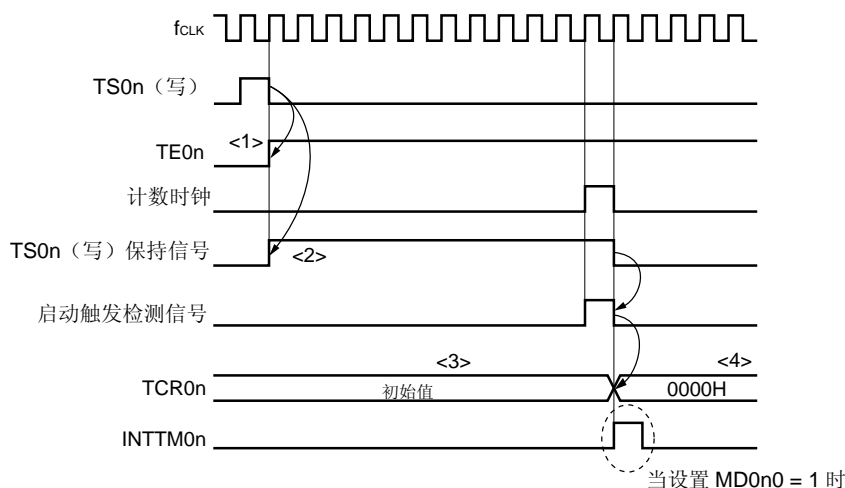
图 6-11. 开始时序 (事件计数器模式下)



(c) 捕捉模式下的开始时序

- <1> 写入 1 到 TS0n 设置 TE0n = 1
- <2> TS0n 的写入数据保持到计数时钟发生。
- <3> TCR0n 初始值保持到计数时钟发生。
- <4> 计数时钟产生时, 0000H 加载到 TCR0n 并且计数开始。

图 6-12. 开始时序 (捕捉模式下)

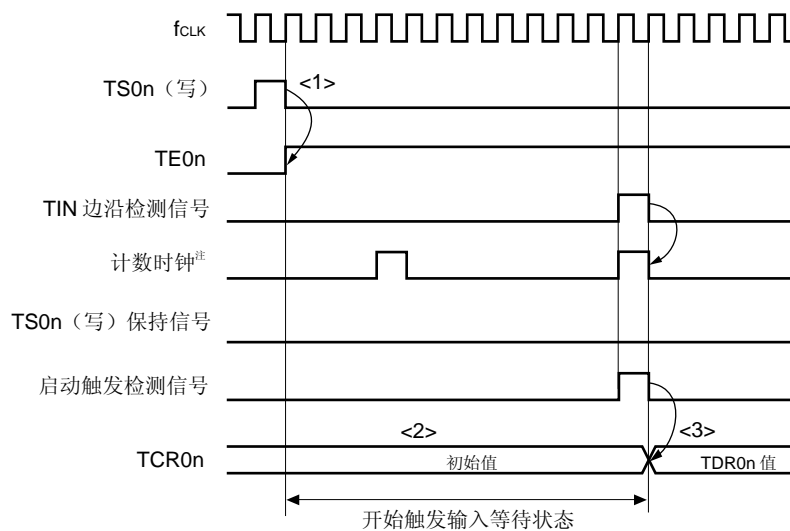


注意事项 TS0n 写入后在计数时钟的第一个周期操作中, 由于直到产生计数时钟开始计数, 产生最大一个时钟的错误。当在计数开始时序的信息需要时, 通过设置 MD0n0 = 1 可在计数开始产生中断。

(d) 单计数模式下的开始时序

- <1> 写入 1 到 TS0n 设置 TE0n = 1
- <2> 进入开始触发输入等待状态，TCR0n 保持初始值。
- <3> 开始触发检测，“TDR0n 值”被加载到 TCR0n 并且计数开始。

图 6-13. 开始时序（单计数模式下）



注 当设置单计数模式时，操作时钟（MCK）被选择作为计数时钟（CCS0n = 0）。

注意事项 因为操作在开始触发检测时开始，产生一个输入信号采样误差（使用 TI0n 时误差为一个计数时钟）。

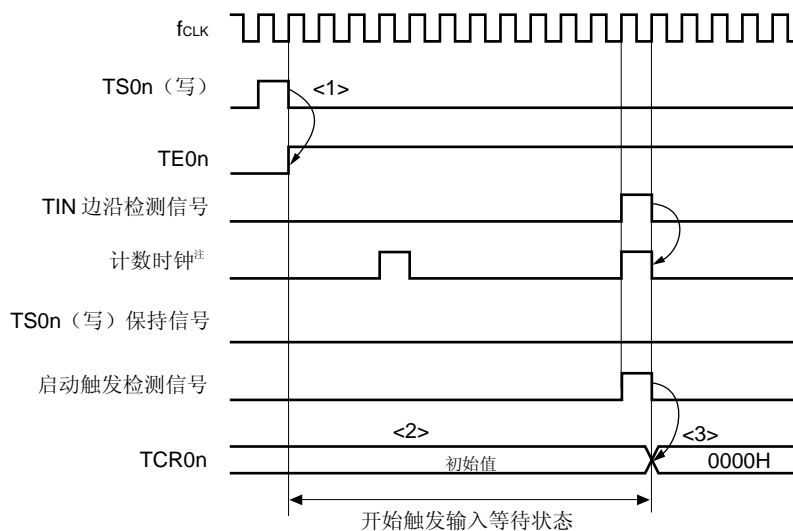
(e) 捕捉& 单计数 模式下的开始时序

<1>写入 1 到 TS0n 设置 TE0n = 1

<2>进入开始触发输入等待状态，TCR0n 保持初始值。

<3>开始触发检测，0000H 被加载到 TCR0n 并且计数开始。

图 6-14. 开始时序 (在捕捉& 单计数 模式下)



注 当设置捕捉& 单计数模式时，操作时钟 (MCK) 被选择作为计数时钟 (CCS0n = 0)。

注意事项 因为操作在开始触发检测时开始，产生一个输入信号采样误差 (使用 **TIO_n** 时误差为一个计数时钟)。

(7) 定时器通道停止寄存器 0 (TT0)

TT0 是触发寄存器，用于每个通道清零定时器计数器 (TCR0n) 可启动计数操作。

当此寄存器的一个位 (TT0n) 设置为 1 时，定时器通道允许状态寄存器 0 (TE0) 的相应位 (TE0n) 清为 0。TT0n 是触发位，并且当 TE0n = 0 时立即清零。

TT0 可由 16 位存储器操作指令设置。

TT0 的低 8 位和 TT0L 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-15. 定时器通道停止寄存器 0 (TT0) 的格式

地址: F01B4H, F01B5H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

TT0n	通道 n 的操作停止触发
0	无触发操作
1	操作停止 (停止触发产生)。

注意事项 务必将第 15~ 8 位清为“0”。

- 备注**
1. 当读取 TT0 寄存器时，只读取 0。
 2. n = 0~7

(8) 定时器输入选择寄存器 0 (TIS0)

TIS0 用于选择每个通道的信号输入到定时器输入引脚 (TI0n) 或副系统时钟由 4 分频 (fxt/4) 有效。

复位信号产生将此寄存器清为 00H。

TIS0 由 1 位或 8 位存储器操作指令设置。

图 6-16. 定时器输入选择寄存器 0 (TIS0) 的格式

地址: FFF3EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TIS0n	通道 n 使用定时器输入/副系统时钟的选择
0	定时器输入引脚 (TI0n) 的输入信号
1	副系统时钟 4 分频 (fxt/4)

<R> **注意事项** 当使用 LIN 总线通讯功能时，通过将 ISC1 (输入切换控制寄存器 (ISC) 的第 1 位) 设置为 1 且将 TIS07 设置为 0，来选择 RxD3 引脚的输入信号。

(9) 定时器输出允许寄存器 0 (TOE0)

TOE0 用于允许或禁止每个通道的定时器输出。

允许定时器输出的通道 n 不能用软件重写定时器输出寄存器 (TO0) 的 $TO0n$ 位的值, 通过从定时器输出引脚 ($TO0n$) 输出的计数值设置定时器输出功能。

TOE0 可由 16 位存储器操作指令设置。

TOE0 的低 8 位和 TOE0L 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-17. 定时器输出允许寄存器 0 (TOE0) 的格式

地址: F01BAH, F01BBH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

TOE 0n	通道 n 的定时器输出允许/禁止
0	TO0n 由计数操作停止操作 (定时器通道输出位)。 允许写入到 TO0n 位。 TO0n 引脚作为数据输出, 并且输出的电平设置到 TO0n 位。 TO0n 引脚的输出电平可由软件修改。
1	TO0n 由计数操作允许操作 (定时器通道输出位)。 禁止写入到 TO0n 位 (写入忽略)。 TO0n 引脚作为定时器输出, 根据定时器操作设置或复位 TOE0n。 TO0n 引脚根据定时器操作输出方波或 PWM。

注意事项 务必将第 15~ 8 位清为“0”。

备注 $n = 0 \sim 7$

(10) 定时器输出 寄存器 0 (TO0)

TO0 是每个通道的定时器输出的缓冲寄存器。

此寄存器中每个位的值从通道的定时器输出引脚 (TO0n) 输出。

只有当定时器输出禁止 (TOE0n = 0) 时此寄存器可以通过软件重写。当定时器输出允许时 (TOE0n = 1)，通过软件重写此寄存器被忽略，只有当定时器操作时值改变。

<R> 要使用 P01/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P05/TO05, P06/TO06, 或 P145/TO07 引脚作为端口功能引脚, 设置 TO0n 中的对应位为 “0”。

TO0 可由 16 位存储器操作指令设置。

TO0 的低 8 位和 TOOL 可以由 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-18. 定时器输出寄存器 0 (TO0) 的格式

地址: F01B8H, F01B9H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO0 7	TO0 6	TO0 5	TO0 4	TO0 3	TO0 2	TO0 1	TO0 0

TO0 n	通道 n 的定时器输出
0	定时器输出值为 “0”。
1	定时器输出值为 “1”。

注意事项 务必将第 15~ 8 位清为 “0”。

备注 n = 0~7

(11) 定时器输出 电平寄存器 0 (TOL0)

TOL0 是用于控制每个通道定时器输出电平的寄存器。

当在组合操作模式 ($TOM0n = 1$) 中允许定时器输出 ($TOE0n = 1$) 时, 通过此寄存器设置通道 n 反向输出。在反转模式 ($TOM0n = 0$) 下, 此寄存器设置无效。

TOL0 可由 16 位存储器操作指令设置。

TOL0 的低 8 位和 TOL0L 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-19. 定时器输出电平寄存器 0 (TOL0) 的格式

地址: F01BCH, F01BDH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL 07	TOL 06	TOL 05	TOL 04	TOL 03	TOL 02	TOL 01	TOL 00

TOL 0n	通道 n 的定时器输出电平控制														
0	正向逻辑输出 (高有效)														
1	反向输出 (低有效)														

注意事项 务必将第 15~ 8 位清为“0”。

- 备注**
1. 如果此寄存器的值在定时器操作期间被重写, 当定时器输出信号变为下一个, 反转定时器输出, 在寄存器值重写后立即代替。
 2. $n = 0 \sim 7$

(12) 定时器输出模式寄存器 0 (TOM0)

TOM0 用于每个通道的定时器输出模式控制。

<R> 当某个通道用作独立操作功能时，将该通道对应的位设置为“0”。

<R> 当某个通道用作组合操作功能时（PWM 输出，单脉冲输出，或多路 PWM 输出），将主通道对应的位设置为“0”，并将从通道对应的位设置为“1”。

当通道用于组合功能时（PWM 输出，单脉冲输出，或多路 PWM 输出），设置从通道的相应位为 1。

通过此寄存器进行的通道 n 的设置反射到当定时器输出信号设置或定时器输出允许复位时的时序上。

TOM0 可由 16 位存储器操作指令设置。

TOM0 的低 8 位和 TOM0L 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器清为 0000H。

图 6-20. 定时器输出模式寄存器 0 (TOM0) 的格式

地址: F01BEH, F01BFH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	TOM 00
TOM 0n	通道 n 的定时器输出模式控制															
0	反转操作模式（由定时器中断请求信号（INTTM0n）产生反转输出）															
1	组合操作模式（由主通道的定时器中断请求信号（INITTM0n）设置输出，由从通道的定时器中断请求信号（INITTM0m）复位输出）															

注意事项 务必将第 15~8 位清为“0”。

<R> **备注** n: 通道编号, m: 从通道编号
n = 0~7 (主通道的 n = 0, 2, 4, 6)
n < m ≤ 7 (m 是大于 n 的连续整数)

(13) 输入切换控制寄存器 (ISC)

ISC 用于使用通道 7 和串行阵列单元 1 联合实现 LIN 总线通讯操作。

当此寄存器的第 1 位设置为 1 时，串行数据输入引脚 (RxD3) 的输入信号被选择作为定时器输入信号。

ISC 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器 设为 00H。

图 6-21. 输入切换控制寄存器 (ISC) 的格式

地址: FFF3CH 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	切换定时器阵列单元的通道 7 输入
0	Ti07 引脚的输入信号作为定时器输入 (普通操作)。
1	RxD3 引脚的输入信号用作定时器输入 (唤醒信号检测)。

ISC0	切换外部中断 (INTP0) 输入
0	INTP0 引脚的输入信号作为外部中断 (普通操作)。
1	RxD3 引脚的输入信号作为外部中断 (测量同步中断域和同步域的脉冲宽度)。

注意事项 务必将第 7~2 位清为“0”。

<R> **备注** 使用 LIN 总线通信功能时，通过设置 ISC1 为 1 且设置 TIS07 (定时器输入选择寄存器 (TIS0) 的第 7 位) 为“0”，来选择 RxD3 引脚的输入信号。

(14) 噪声滤波器允许寄存器 1 (NFEN1)

NFEN1 用来设置噪声滤波器是否可用于每通道的定时器输入信号。

引脚需要消除噪声时通过设置相应位到 1 允许噪声滤波器。

当噪声滤波器 开时，执行相等检测和与 CPU/外围硬件时钟 (fCLK) 的 2 个时钟同步。当噪声滤波器关闭时，只执行与 CPU/外围硬件时钟 (fCLK) 同步。

NFEN1 可以由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器 设为 00H。

图 6-22. 噪声滤波器允许寄存器 1 (NFEN1) 的格式

地址: F0061H 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	允许/禁止使用 TI07/TO07/P145 引脚或 RxD3/P14 引脚输入信号 ^注 的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN06	允许/禁止使用 TI06/TO06/P06 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN05	允许/禁止使用 TI05/TO05/P05 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN04	允许/禁止使用 TI04/TO04/P42 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN03	允许/禁止使用 TI03/TO03/INTP4/P31 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN02	允许/禁止使用 TI02/TO02/P17 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN01	允许/禁止使用 TI01/TO01/INTP5/P16 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							
TNFEN00	允许/禁止使用 TI00/P00 引脚输入信号的噪声滤波器							
0	噪声滤波器关							
1	噪声滤波器开							

注 可通过设置 ISC 寄存器的 ISC1 切换可应用的引脚。

ISC1 = 0: 可选择是否使用 TI07 引脚的噪声滤波器。

ISC1 = 1: 可选择是否使用 RxD3 引脚的噪声滤波器。

(15) 端口模式寄存器 0, 1, 3, 4, 14 (PM0, PM1, PM3, PM4, PM14)

这些寄存器按位设置端口 0, 1, 3, 4 和 14 的输入/输出。

使用 P01/TO00, P05/TO05/TI05, P06/TO06/TI06, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04 和 P145/TO07/TI07 引脚作为定时器输出时, 设置 PM01, PM05, PM06, PM16, PM17, PM31, PM42 和 PM145 和输出 P01, P05, P06, P16, P17, P31, P42 和 P145 的输出锁存为 0。

使用 P00/TI00, P05/TO05/TI05, P06/TO06/TI06, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04 和 P145/TO07/TI07 引脚作为定时器输入时, 设置 PM00, PM05, PM06, PM16, PM17, PM31, PM42 和 PM145 为 1。此时 P00, P05, P06, P16, P17, P31, P42 和 P145 的输出锁存可以为 0 或 1。

PM0, PM1, PM3, PM4 和 PM14 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将这些寄存器设置为 FFH。

图 6-23. 端口模式寄存器 0, 1, 3, 4, 和 14 (PM0, PM1, PM3, PM4, PM14) 的格式

地址: FFF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

地址: FFF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

地址: FFF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

地址: FFF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

地址: FFF2EH 复位后: FFH R/W

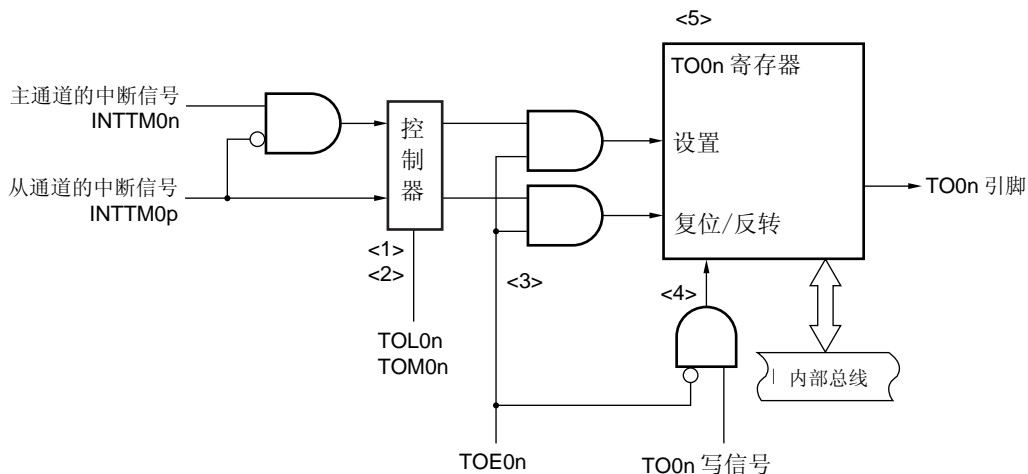
符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn 引脚 I/O 模式选择 (m = 0, 1, 3, 4, 14; n = 0~7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

6.4 通道输出（TO0n 引脚）控制

6.4.1 TO0n 引脚输出电路配置

图 6-24. 输出电路配置



TO0n 引脚输出电路描述如下。

- <1> TOM0n = 0（反转模式）时，忽略 TOL0n 寄存器的设置值并且只有 INTTM0p（从通道定时器中断）被发送到 TO0n 寄存器。
- <2> TOM0n = 1（组合操作模式）时，INTTM0n（主通道定时器中断）和 INTTM0p（从通道定时器中断）均被发送到 TO0n 寄存器。
这时，TOL0n 寄存器有效，并且下面为信号的控制：

TOL0n = 0 时： 正向操作（INTTM0 → 设置，INTTM0p → 复位）
TOL0n = 1 时： 反向操作（INTTM0 → 复位，INTTM0p → 设置）

INTTM0n 和 INTTM0p 同时产生时，（0%输出 PWM），INTTM0p（复位信号）优先，并且 INTTM0n（设置信号）被屏蔽。

- <3> TOE0n = 1 时，INTTM0n（主通道定时器中断）和 INTTM0p（从通道定时器中断）被发送到 TO0n 寄存器。 写入到 TO0n 寄存器（TO0n 写入信号）无效。
TOE0n = 1 时，TO0n 引脚输出出中断信号外不改变。
要初始化 TO0n 引脚输出电平，需要设置 TOE0n = 0 并且给 TO0n 写入一个值。
- <4> TOE0n = 0 时，写入 TO0n 位到触发通道（TO0n 信号）有效。TOE0n = 0 时，INTTM0n（主通道定时器中断）和 INTTM0p（从通道定时器中断）都不发送到 TO0n 寄存器。
- <5> TO0n 寄存器一直可读，TO0n 引脚输出电平可检测。

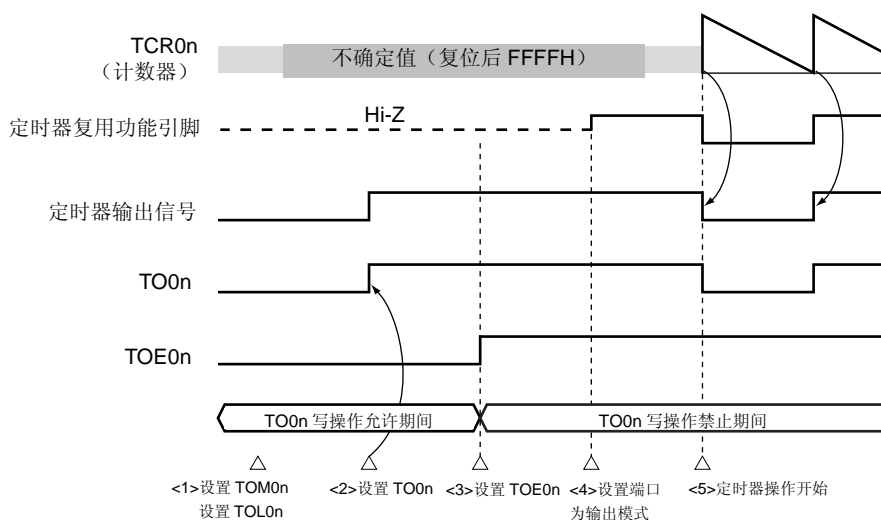
备注

1. n = 0~7（主通道 n = 0, 2, 4, 或 6）
2. p = n + 1, n + 2, n + 3 ...（p ≤ 7）

6.4.2 TO0n引脚输出设置

下图所示为 TO0n 输出引脚从初始设置到定时器操作开始的发送过程和状态。

图 6-25. 从定时器输出设置到操作开始的状态发送



<1> 定时器输出的操作模式设置。

- TOM0n 位 (0: 反转模式, 1: 组合操作模式)
- TOL0n 位 (0: 正向输出, 1: 反向输出)

<2> 定时器输出信号通过设置 TO0n 设置到初始状态。

<3> 定时器输出操作通过写入 1 到 TOE0n 允许 (写入 TO0n 禁止)。

<4> 端口 I/O 设置为输出 (参见 6.3 (15) 端口模式寄存器 0, 1, 3, 4, 14)。

<5> 定时器操作允许 (TS0n = 1)。

备注 n = 0~7

6.4.3 通道输出操作的注意事项

(1) 定时器操作期间改变寄存器 TO0, TOE0, TOL0 和 TOM0 中设置的值

因为定时器操作 (TCR0n 和 TDR0n 的操作) 与 TO0n 输出电路独立并且改变设置在 TO0, TOE0, TOL0, 和 TOM0 的值不影响定时器操作, 所以在定时器操作期间这些值可以被改变。

当 TOE0, TOL0, 和 TOM0 (除 TO0 外) 中设置的值改变为关闭定时器中断 (INTTM0n), 根据产生定时器中断 (INTTM0n) 信号之前或之后值是否改变, 输出到 TO0 引脚的波形可能会不同。

备注 n = 0~7

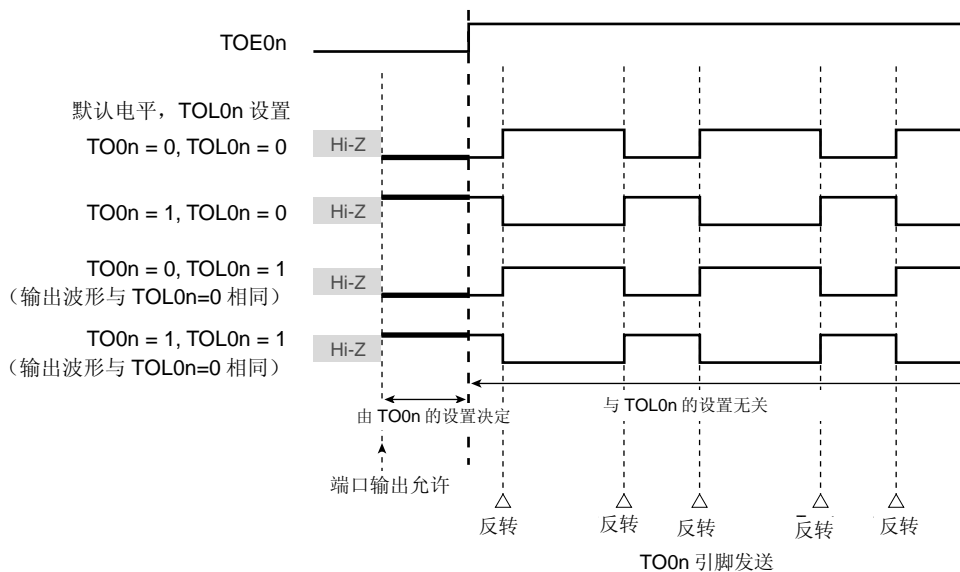
(2) TO0n 引脚的默认电平和定时器操作开始后输出电平

在改变默认电平之后，在允许端口输出和设置 TOE0n = 1 之前，当在 TOE0n = 0 的状态写入完成时 TO0n 引脚输出电平的转换如下图所示。

(a) 设置 TOM0n = 0 当操作开始时（反转输出）

TOM0n = 0 时 TOL0n 设置无效。当定时器操作在设置默认电平后开始，出发信号产生并且 O0n 引脚的输出电平被反转。

图 6-26. 反转输出时 TO0n 输出 状态 (TOM0n = 0)

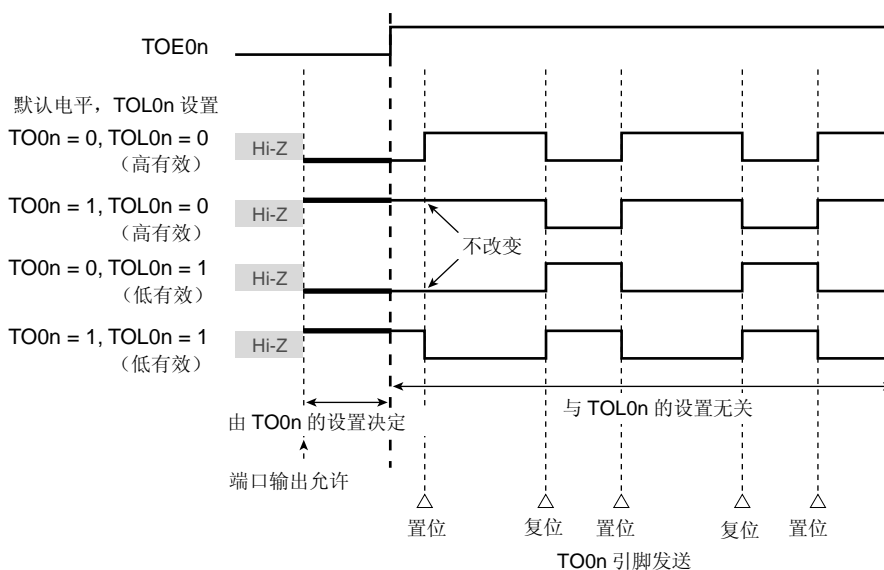


- 备注
1. 反转：TO0n 引脚输出状态反向
 2. n = 0~7

(b) 设置 $TOM0n = 1$ 当操作开始时 (组合操作模式 (PWM 输出))

$TOM0n = 1$ 时, 有效电平由 $TOL0n$ 设置。

图 6-27. PWM 输出时 $TO0n$ 引脚输出状态 ($TOM0n = 1$)



- 备注**
1. 设置: $TO0n$ 引脚的输出信号从无效电平变为有效电平。
复位: $TO0n$ 引脚的输出信号从有效电平变为无效电平。
 2. $n = 0 \sim 7$

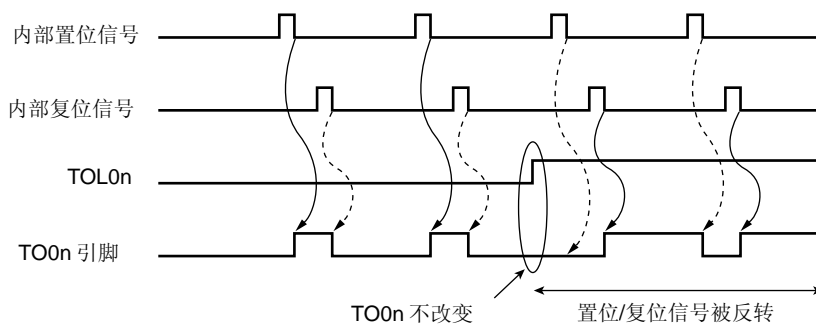
(3) $TO0n$ 引脚在组合操作模式 ($TOM0n = 1$) 下的操作

(a) 当 $TOL0n$ 设置在定时器操作期间已经被改变

当 $TOL0n$ 设置在定时器操作期间已经被改变时, 当 $TO0n$ 改变条件的时序发生时设置变为有效。重写 $TOL0n$ 不改变 $TO0n$ 的输出电平。

下图所示为 $TOL0n$ 的值在定时器操作期间 ($TOM0n = 1$) 被改变时的操作。

图 6-28. 当 $TOL0n$ 在定时器操作期间被改变时的操作



- 备注**
1. 设置: $TO0n$ 引脚的输出信号从无效电平变为有效电平。
复位: $TO0n$ 引脚的输出信号从有效电平变为无效电平。
 2. $n = 0 \sim 7$

(b) 设置/复位时序

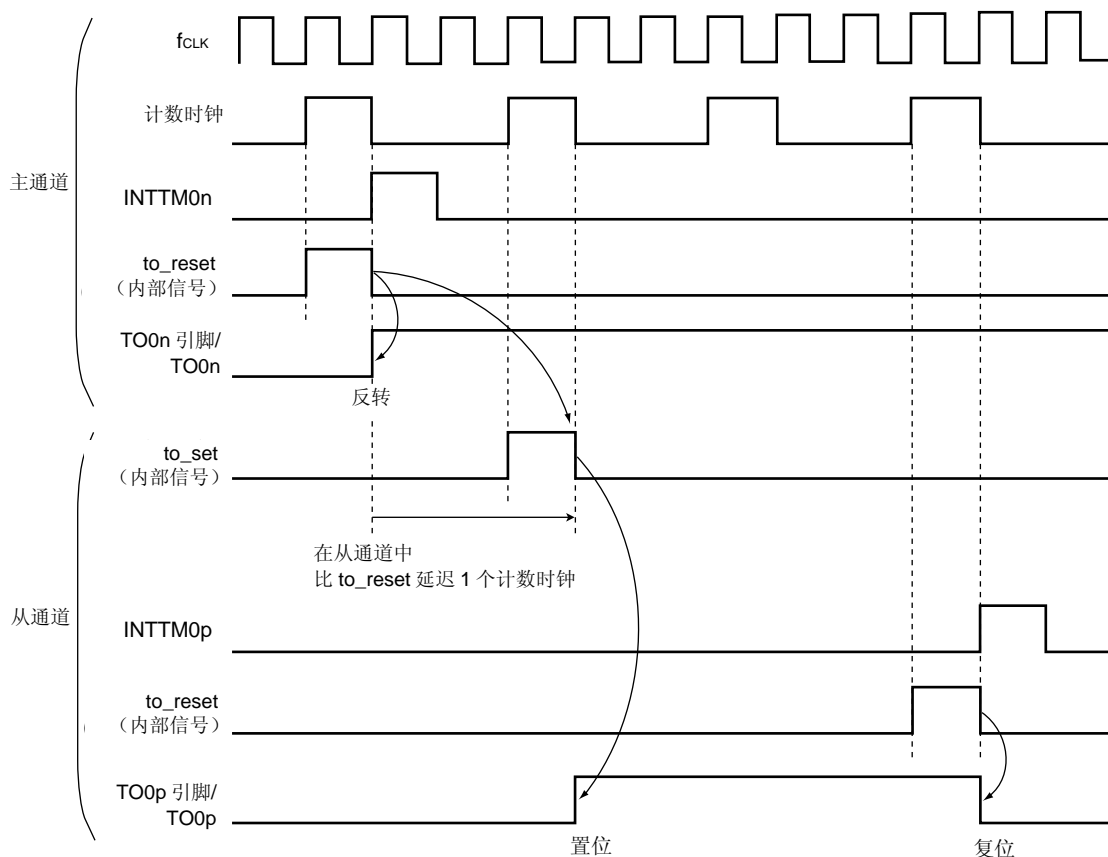
在 PWM 输出时要实现 0%/100%输出，主通道定时器中断（INTTM0n）发生时 TO0n 引脚/TO0n 设置时序由从通道定时器中断（INTTM0p）延迟一个计数时钟。

如果设置条件和复位条件同时发生，后者具有较高的优先级。

图 6-29 所示为主/从通道设置如下时设置/复位操作状态。

主通道: TOE0n = 1, TOM0n = 0, TOL0n = 0
从通道: TOE0p = 1, TOM0p = 1, TOL0p = 0

图 6-29. 设置/复位时序操作状态

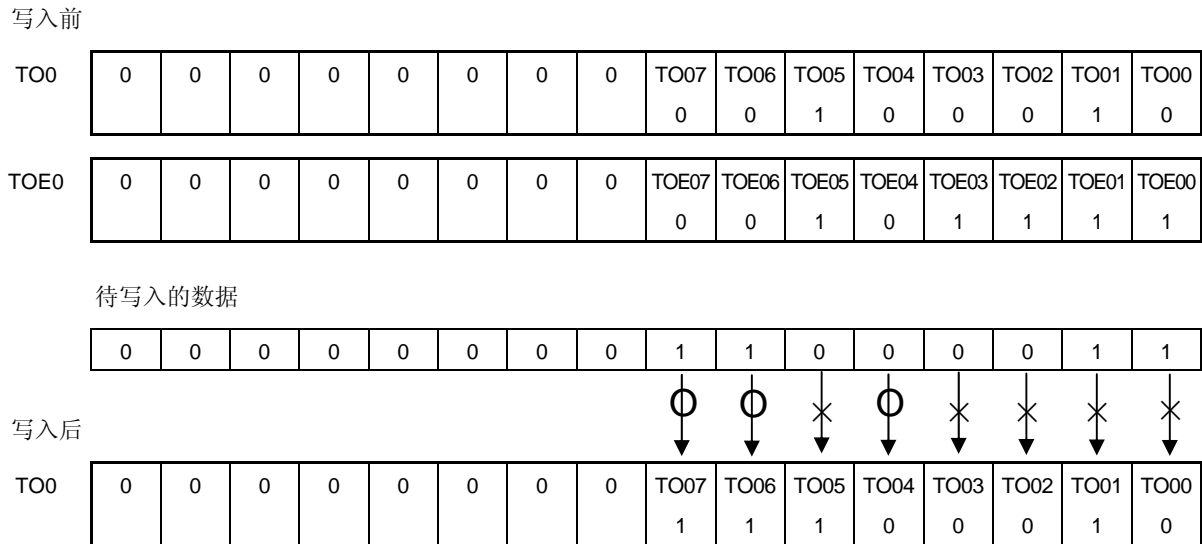


- 备注**
1. to_reset: TO0n 引脚复位/反转信号
to_set: TO0n 引脚设置信号
 2. n = 0~7 (当 n = 0, 2, 4, 或 6 对于主通道)
 3. p = n+1, n+2, n+3 ... (p ≤ 7)

6.4.4 TO0n位的集中操作

在 TO0 寄存器中，所有通道的设置位可在一个寄存器中定位，与 TS0 寄存器（通道开始触发器）相同。因此所有通道的 TO0n 可集中操作。只有指定位可通过设置相应 TOE0n = 0 操作到目标 TO0n（通道输出）。

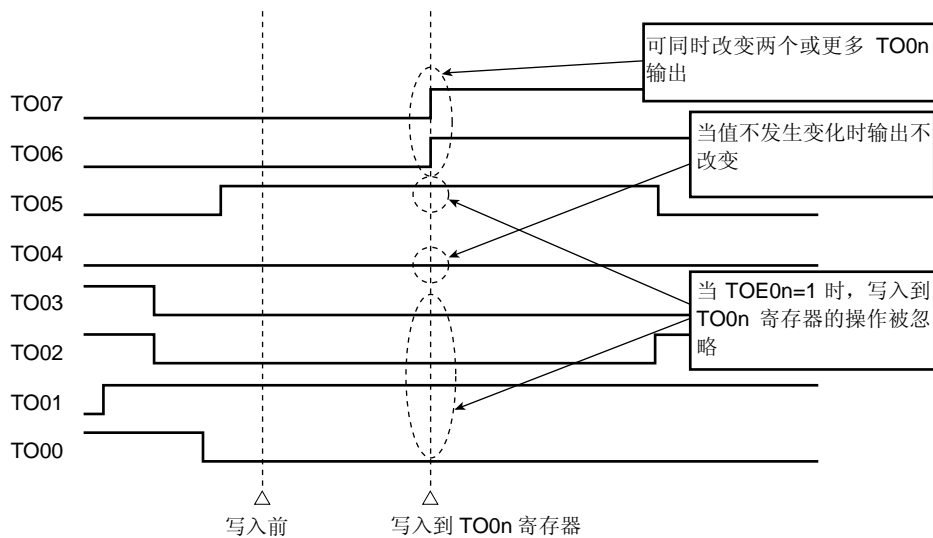
图 6-30. TO0n 位集中操作的示例



TOE0n = 0 只写入 TO0n 位，TOE0n = 1 写入 TO0n 被忽略。

TO0n（通道输出），设置为 TOE0n = 1，不被写操作影响。即使对 TO0n 进行写操作，将其忽略并且又定时器操作改变的输出正常执行。

图 6-31. 通过集中操作 Toon 位 TO0n 引脚状态



(注意事项和备注在下页中给出。)

注意事项 当 $TOE0n = 1$ 时，即使各定时器内容写入 $TO0n$ 的定时器中断（ $INTTM0n$ ）， $TO0n$ 引脚输出正常。

备注 $n = 0 \sim 7$

6.4.5 定时器中断和操作开始时 $TO0n$ 引脚输出

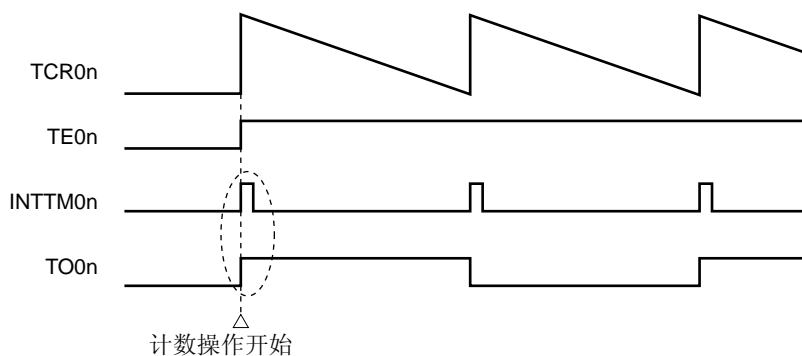
在间隔定时器模式或捕捉模式， $TMR0n$ 寄存器中的 $MD0n0$ 位设置是否在计数开始时产生定时器中断。

当 $MD0n0$ 设置为 1 时，计数操作开始时可通过定时器中断（ $INTTM0n$ ）产生被识别。

在其他模式下，在计数操作开始的定时器中断和 $TO0n$ 输出都不被控制。

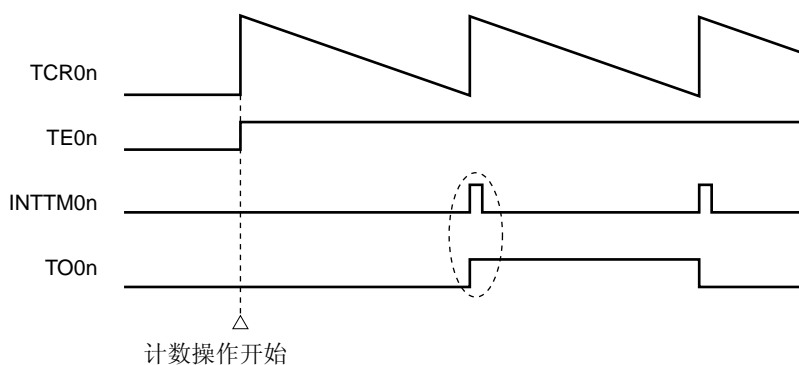
图 6-32 和 6-33 所示为设置为间隔定时器模式（ $TOE0n = 1$ ， $TOM0n = 0$ ）时的操作示例。

图 6-32. 当 $MD0n0$ 设置为 1



$MD0n0$ 设置为 1 时，定时器中断（ $INTTM0n$ ）在计数操作开始时输出，并且 $TO0n$ 执行反转操作。

图 6-33. 当 $MD0n0$ 设置为 0



$MD0n0$ 设置为 0 时，定时器中断（ $INTTM0n$ ）在计数操作开始时不输出，并且 $TO0n$ 也不改变。计数一个周期后， $INTTM0n$ 输出并且 $TO0n$ 执行反转操作。

备注 $n = 0 \sim 7$

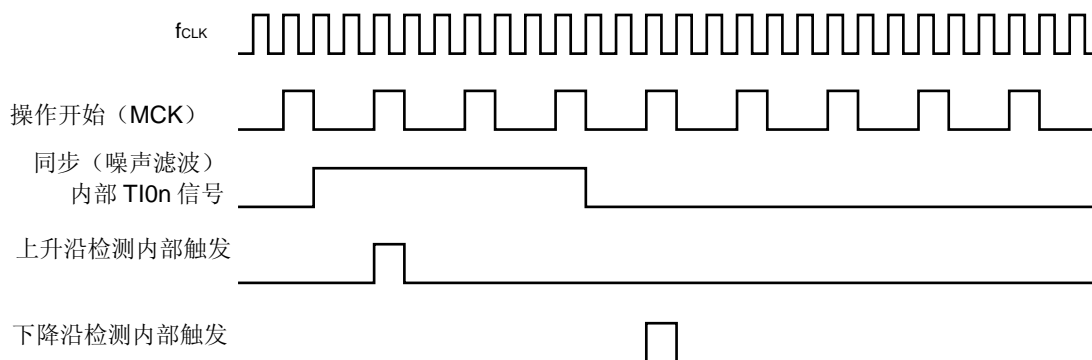
6.5 通道输入（TI0n 引脚）控制

<R> 6.5.1 TI0n 边沿检测电路

(1) 边沿检测基本操作时序

沿检测电路采样执行与操作时钟（MCK）一致。

图 6-34. 边沿检测基本操作时序



备注 n = 0~7

6.6 定时器阵列单元的基本功能

6.6.1 独立操作功能和组合操作功能的概述

定时器阵列单元由几个通道组成，具有独立操作功能允许每个通道独立操作，和使用两个或更多通道组合在一起的组合操作功能。

任意通道可使用独立操作功能，不论其他通道的操作模式。

组合操作功能通过将主通道（主要计数周期的参考定时器）和一个从通道（与主通道操作一直的定时器）组合在一起实现，使用此功能时必须遵守下面几条规则。

6.6.2 组合操作功能的基本规则

使用组合操作功能的基本规则如下所示。

- (1) 只有偶数通道（通道 0, 2, 4, 等）可设置为主通道。
- (2) 除通道 0 以外的任一通道都可设置为从通道。
- (3) 从通道必须低于主通道。
例：如果通道 2 设置为主通道，通道 3 以后的通道（通道 3, 4, 5, 等）可设置为从通道。
- (4) 一个主通道可以设置两个或更多从通道。
- (5) 当使用两个以上主通道时，与从通道之间有主通道的从通道不能被设置。
例：如果通道 0 和 4 设置为主通道，通道 1 到 3 可被设置为主通道 0 的从通道。通道 5~7 不能设置作为主通道 0 的从通道。
- (6) 从通道与主通道组合时，从通道的操作时钟必须与主通道相同。与主通道一起操作的从通道的 CKS 位（TMR0n 寄存器的第 15 位）必须与主通道的相应位的值相同。
- (7) 主通道可发送 INTTM0n（中断），启动软件触发和低通道计数时钟。
- (8) 从通道可使用 INTTM0n（中断），启动软件触发，和主通道的计数时钟，但是不能发送自己的 INTTM0n（中断），启动软件触发，和计数时钟到低通道。
- (9) 主通道不能使用来自高位主通道 INTTM0n（中断），启动软件触发，和计数时钟。
- (10) 要同时启动组合操作的通道，组合通道的 TS0n 位必须同时设置。
- (11) 在计数操作期间，所有通道的 TS0n 位组合操作或只有主通道可设置。只有从通道的 TS0n 则不能设置。
- (12) 要同时停止组合的通道，组合通道的 TT0n 位必须同时设置。

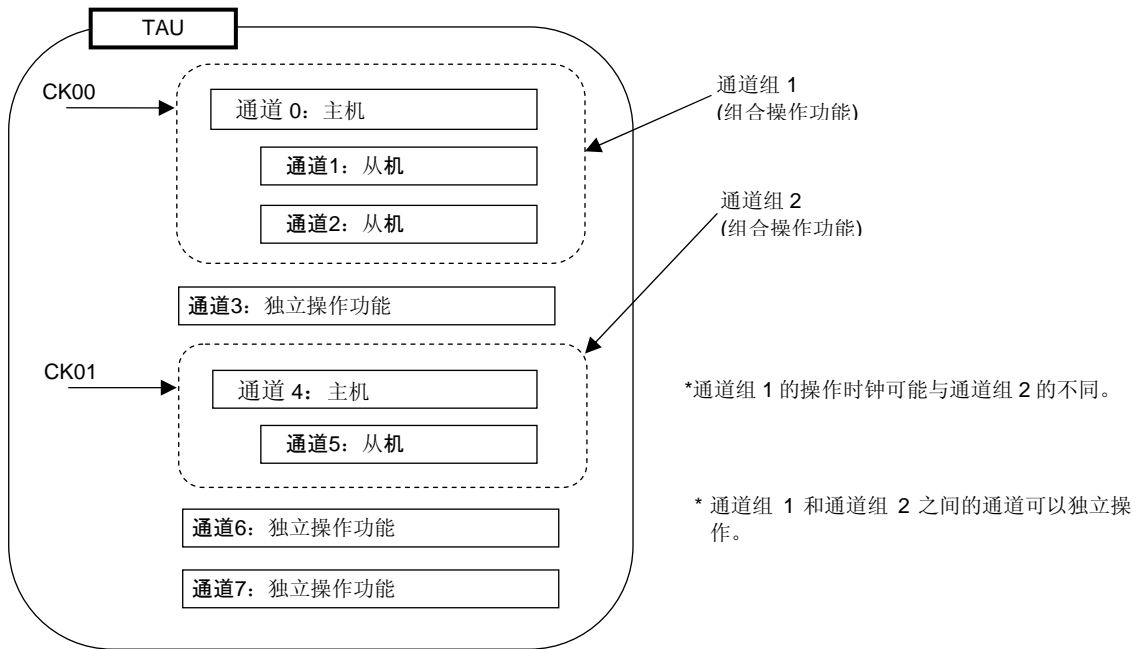
备注 n = 0~7

6.6.3 组合操作功能的基本规则应用范围

组合操作功能的规则应用于一个通道组（一个主通道和从通道组成一个组合操作功能）。

如果被指定的两个或更多通道组不组合操作，则 **6.6.2 组合操作功能的基本规则** 中的组合操作功能的基本规则不适用于该通道组。

例



6.7 定时器阵列单元作为通道的操作

6.7.1 作为间隔定时器/方波输出操作

(1) 间隔定时器

定时器阵列单元可用作产生固定间隔的 INTTM0n（定时器中断）的参考定时器。

可通过下面的表达式计算出中断产生周期。

$$\text{INTTM0n (定时器中断) 的产生周期} = \text{计数时钟周期} \times (\text{TDR0n 的设置值} + 1)$$

(2) 作为方波输出操作

一旦产生 INTTM0n，TO0n 就执行反转操作，并且输出一个占空比为 50% 的方波。

TO0n 输出方波频率和周期可以通过下面的表达式计算出。

$$\bullet \text{ 从 TO0n 方波输出的周期} = \text{计数时钟周期} \times (\text{TDR0n 的设置值} + 1) \times 2$$

$$\bullet \text{ 从 TO0n 方波输出的频率} = \text{计数时钟频率} / \{ (\text{TDR0n 的设置值} + 1) \times 2 \}$$

TCR0n 计数器在间隔定时器模式作为向下计数器操作。

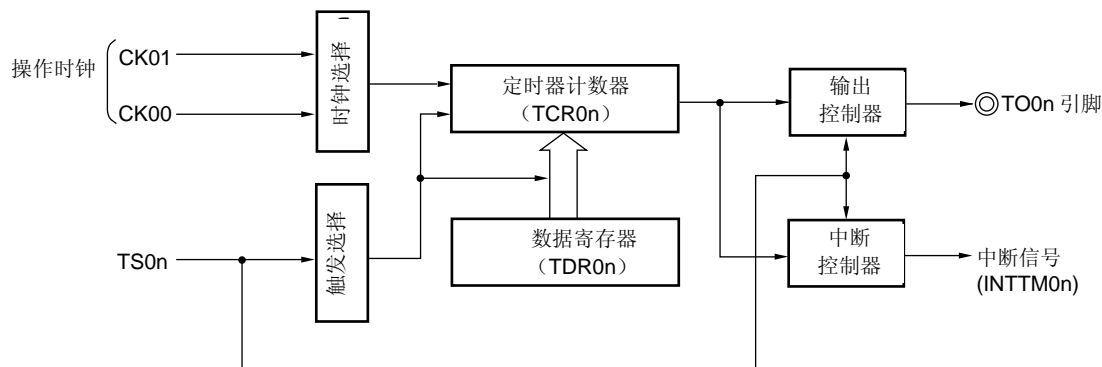
通道启动触发位 (TS0n) 设置为 1 后，TCR0n 在第一个计数时钟加载 TDR0n 的值。如果 TMR0n = 0 的 MD0n0 这时，INTTM0n 不输出并且 TO0n 不反转。如 TMR0n = 1 的 MD0n0，INTTM0n 输出并且 TO0n 反转。

之后，TCR0n 与计数时钟同步向下计数。

当 TCR0n = 0000H 时，INTTM0n 输出 并且 TO0n 在下一个计数时钟反转。同时，TCR0n 再次加载 TDR0n 的值。之后，重复相同的操作。

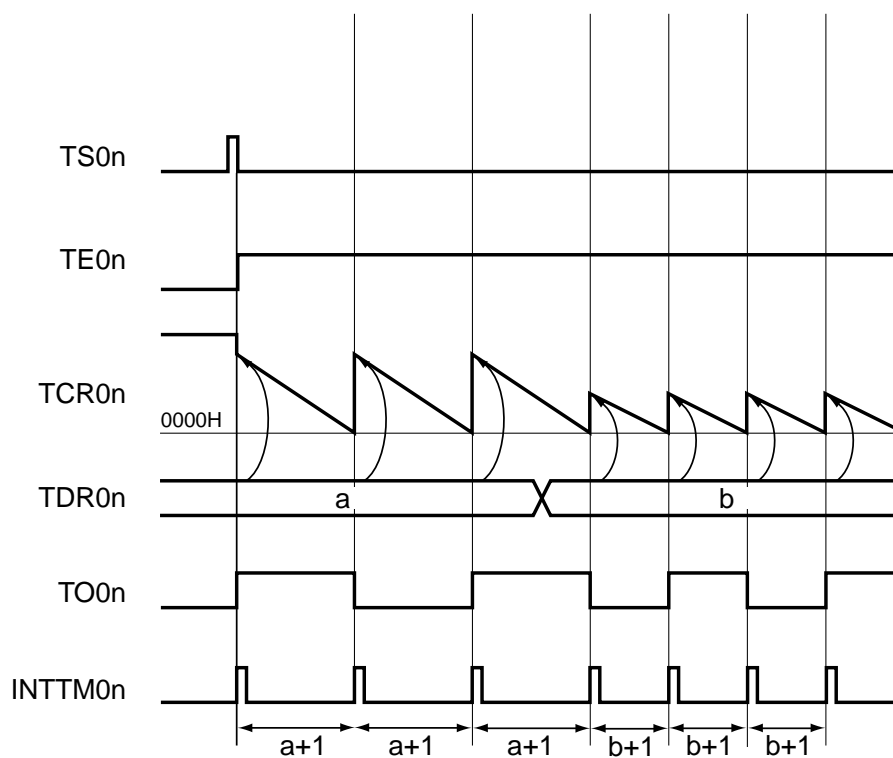
TDR0n 任何时候都可重写。TDR0n 的新值从下个周期开始有效。

图 6-35. 作为间隔定时器/方波输出操作的框图



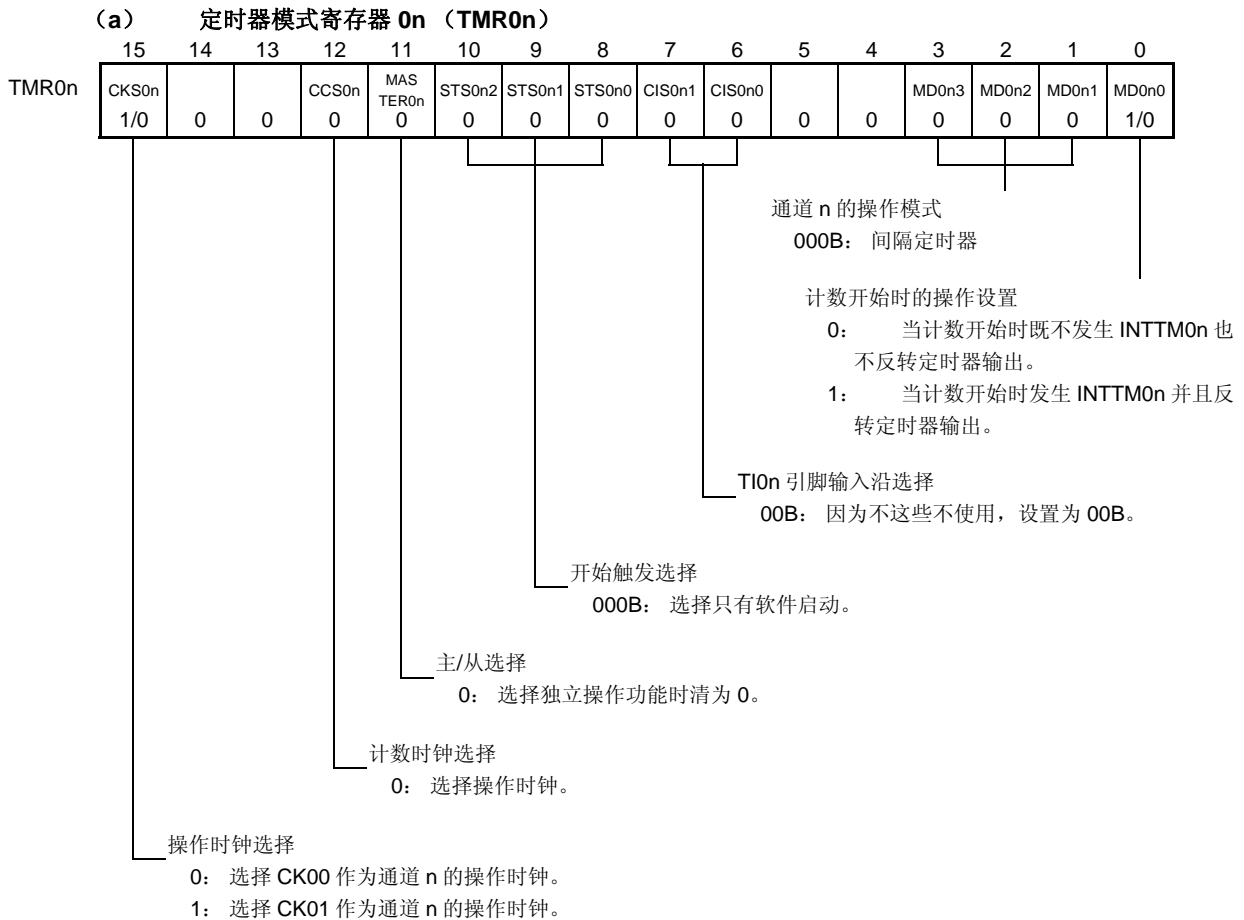
备注 n = 0~7

图 6-36. 作为间隔定时器/方波输出 ($MD0n0 = 1$) 操作的基本时序示例

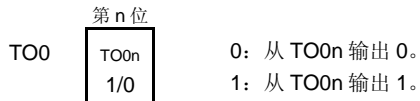


备注 $n = 0 \sim 7$

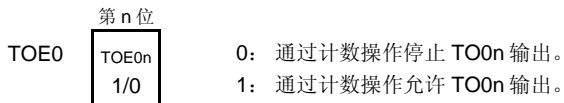
图 6-37. 间隔定时器/方波输出 操作期间寄存器设置内容的示例



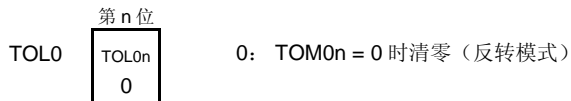
(b) 定时器输出 寄存器 0 (TO0)



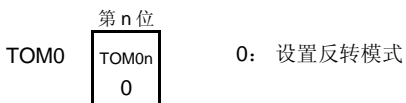
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



备注 n = 0~7

图 6-38. 间隔定时器/方波输出功能的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道默认设置	设置 TMR0n 寄存器 (决定通道的操作模式)。 设置间隔 (周期) 值到 TDR0n 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	要使用 TO0n 输出 将 TOM0 寄存器的 TOM0n 位清为 0 (反转模式)。 将 TOL0n 位清为 0。 设置 TO0n 位并且决定 TO0n 输出的默认值。 设置 TOE0n 为 1 并且允许 TO0n 的操作。 将端口寄存器和端口模式寄存器清为 0。	TO0n 引脚为 Hi-Z 输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TO0n 默认设置电平。 因为通道停止操作 TO0n 不改变。 TO0n 引脚输出 TO0n 设置电平。
操作开始	设置 TOE0n 为 1 (只当操作恢复时)。 设置 TS0n 位为 1。 由于是触发位, TS0n 位自动恢复为 0。	TE0n = 1, 并且计数操作开始。 在计数时钟输入时 TDR0n 的值被加载到 TCR0n。INTTM0n 产生, 并且如果 TMR0n 寄存器的 MD0n0 位为 1 时 TO0n 执行反转操作。
操作期间	TMR0n, TOM0, 和 TOL0 寄存器的设置值不能改变。 TDR0n 寄存器的设置值不能改变。 TCR0n 寄存器总是可读。 TSR0n 寄存器不使用。 TO0 和 TOE0 寄存器的设置值可以改变。	计数器 (TCR0n) 向下计数。当计数值达到 0000H 时, TDR0n 的值被再次加载到 TCR0n, 并且计数操作继续。通过检测 TCR0n=0000H, INTTM0n 发生, 并且执行反转操作。之后, 重复上面的操作。
操作停止	TT0n 位设置为 1。 由于是触发位, TT0n 位自动恢复为 0。 TOE0n 清为 0 并且值设置到 TO0 寄存器。	TE0n = 0, 并且计数操作停止。 TCR0n 保持计数值并停止。 TO0n 输出没有被初始化但保持当前状态。 TO0n 引脚输出 TO0n 设置电平。
TAU 停止	要保持 TO0n 引脚输出电平 在要保存的值设置到端口寄存器后将 TO0n 位置 0。	TO0n 引脚输出电平由端口功能保持。
	不需要保持 TO0n 引脚输出电平时, 将端口模式寄存器切换到输入模式。	TO0n 引脚输出电平变为 Hi-Z 输出状态。
	PER0 寄存器的 TAU0EN 位被清为 0。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TO0n 位被清零, TO0n 引脚设置为端口模式。)

操作恢复。

备注 n = 0~7

6.7.2 作为外部事件计数器操作

定时器阵列单元可用作外部事件计数器，对有效输入沿（外部事件）的数目进行计数，在 TI0n 引脚检测。当达到指定计数值时，事件计数器产生一个中断。指定的计数号可通过下面的表达式计算得出。

$$\text{指定的计数序号} = \text{TDR0n 的设置值} + 1$$

在事件计数器模式下 TCR0n 作为向下计数器操作。

通道启动触发位（TS0n）设置为 1 时，TCR0n 加载 TDR0n 的值。

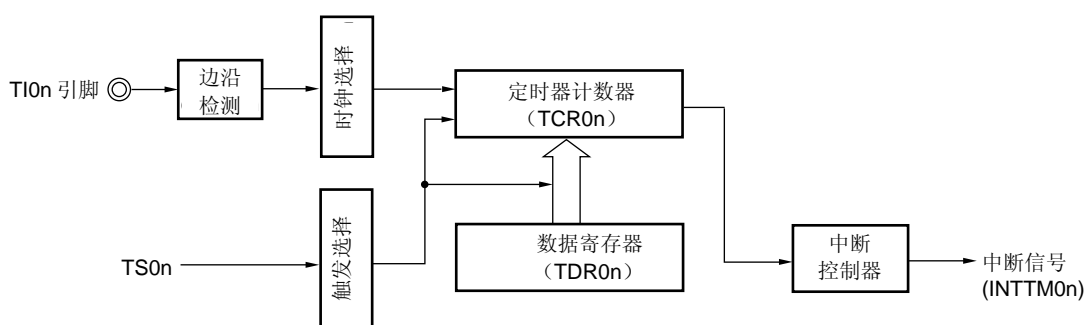
每次 TI0n 引脚的有效输入沿被检测到时 TCR0n 向下计数。当 TCR0n = 0000H 时，TDR0n 的值再次加载到 TCR0n，并且输出 INTTM0n。

之后，重复上面的操作。

TO0n 不能使用，因为它的波形根据外部事件决定，并且不规则。

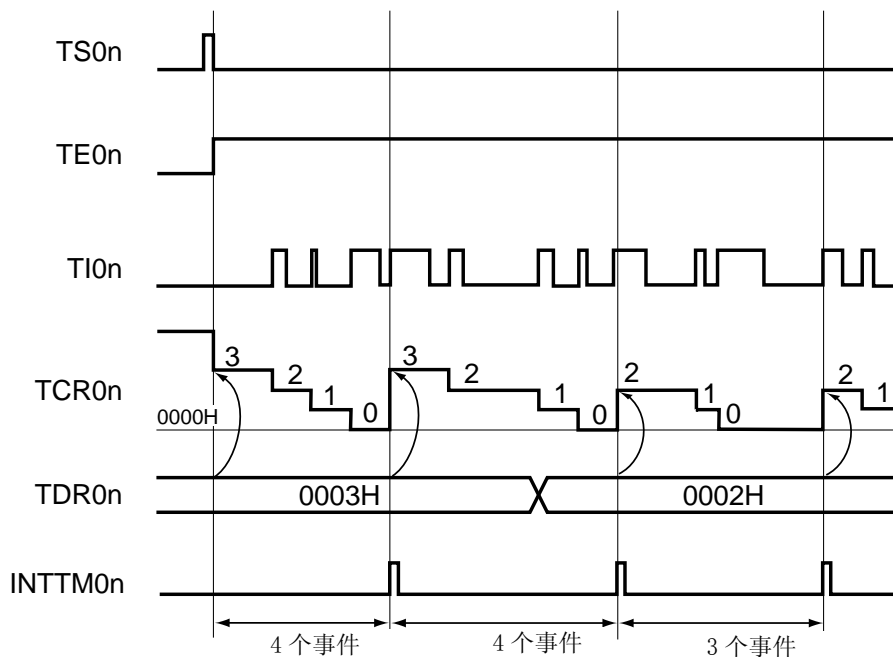
TDR0n 可在任何时间重写。在下一个计数周期 TDR0n 的新值有效。

图 6-39. 作为外部事件计数器操作的框图



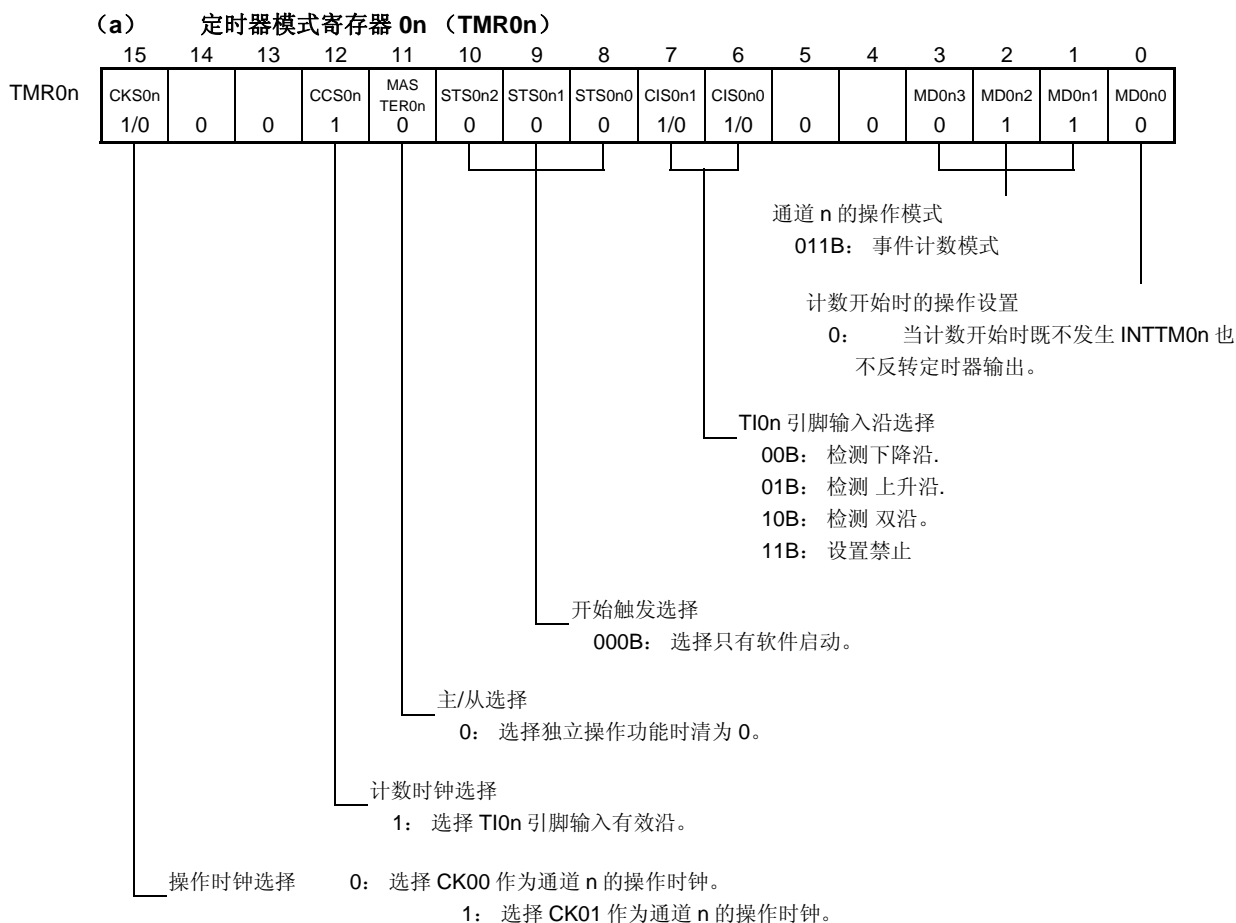
备注 n = 0~7

图 6-40. 作为外部事件计数器操作的基本时序示例

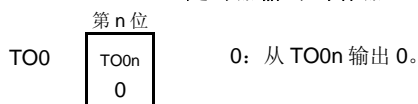


备注 n = 0~7

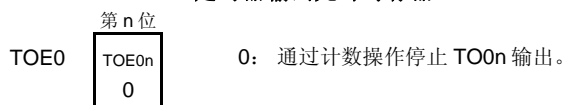
图 6-41. 在外部事件计数器模式下寄存器设置的示例



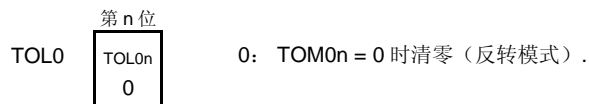
(b) 定时器输出寄存器 0 (TO0)



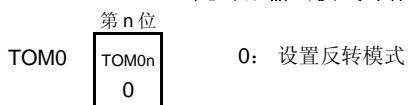
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



备注 n = 0~7

图 6-42. 使用外部事件计数器功能时的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	
通道默认设置	设置 TMR0n 寄存器 (决定通道的操作模式)。 设置计数到 TDR0n 寄存器的数。 将 TOE0 寄存器的 TOE0n 位清为 0。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置 TS0n 位为 1。 由于是触发位, TS0n 位自动恢复为 0。	TE0n = 1, 并且计数操作开始。 TDR0n 的值加载到 TCR0n 并且 TI0n 引脚输入沿的检测等待。
操作期间	TDR0n 寄存器的设置值不能改变。 TCR0n 寄存器总是可读。 TSR0n 寄存器不使用。 TMR0n, TOM0, TOL0, TO0, 和 TOE0 寄存器的设置值不能改变。	计数器 (TCR0n) 在每次检测到 TI0n 引脚输入沿时向下计数。当计数值达到 0000H, TDR0n 的值被再次加载到 TCR0n, 并且计数操作继续。通过检测 TCR0n=0000H, INTTM0n 发生。 之后, 重复上面的操作。
操作停止	TT0n 位设置为 1。 由于是触发位, TT0n 位自动恢复为 0。	TE0n = 0, 并且计数操作停止。 TCR0n 保持计数值并停止。
TAU 停止	PER0 寄存器的 TAU0EN 位被清为 0。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

操作恢复。

备注 n = 0~7

6.7.3 用作分频操作

The 定时器阵列单元可用作分频器，分频输入到 TI0n 引脚和从 TO0n 输出的时钟。
从 TO0n 输出的时钟分频可由下面的表达式计算出。

- 当选择上升沿/下降沿时：
分频时钟频率=输入时钟频率/{ (TDR0n 的设置值 + 1) × 2}
- 当选择双沿时：
分频时钟频率 \cong 输入时钟频率 / (TDR0n 的设置值 + 1)

在间隔定时器 模式下 TCR0n 作为向下计数器操作。

通道启动触发位 (TS0n) 设置为 1 后，当 TI0n 有效沿检测到时 TCR0n 加载 TDR0n 的值。如果此时 TMR0n 的 MD0n0 = 0，INTTM0n 不输出并且 TO0n 不反转。如果 TMR0n 的 MD0n0 = 1，INTTM0n 输出并且 TO0n 反转。

之后，TCR0n 在 TI0n 有效沿向下计数。当 TCR0n = 0000H 时，反转 TO0n。同时，TCR0n 再次加载 TDR0n 的值，继续计数。

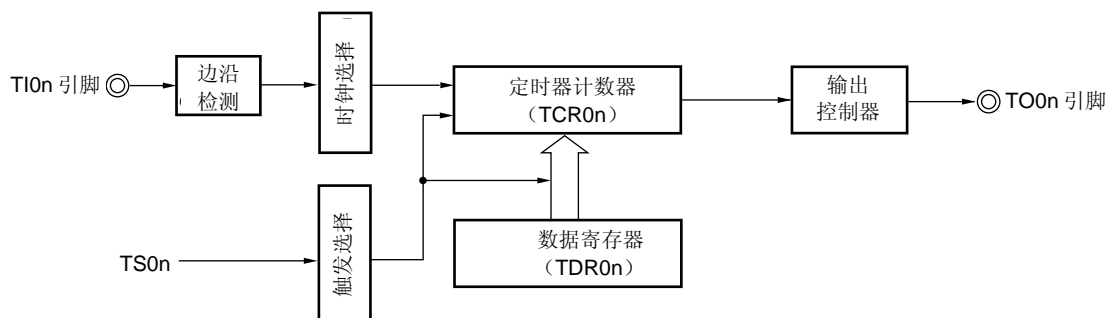
如果选择 TI0n 的双沿检测，输入时钟的占空比误差影响 TO0n 输出的分频时钟周期。

TO0n 输出时钟周期包括操作时钟一个周期的采样误差。

$$\text{TO0n 输出 时钟周期} = \text{理想 TO0n 输出时钟周期} \pm \text{操作时钟周期 (误差)}$$

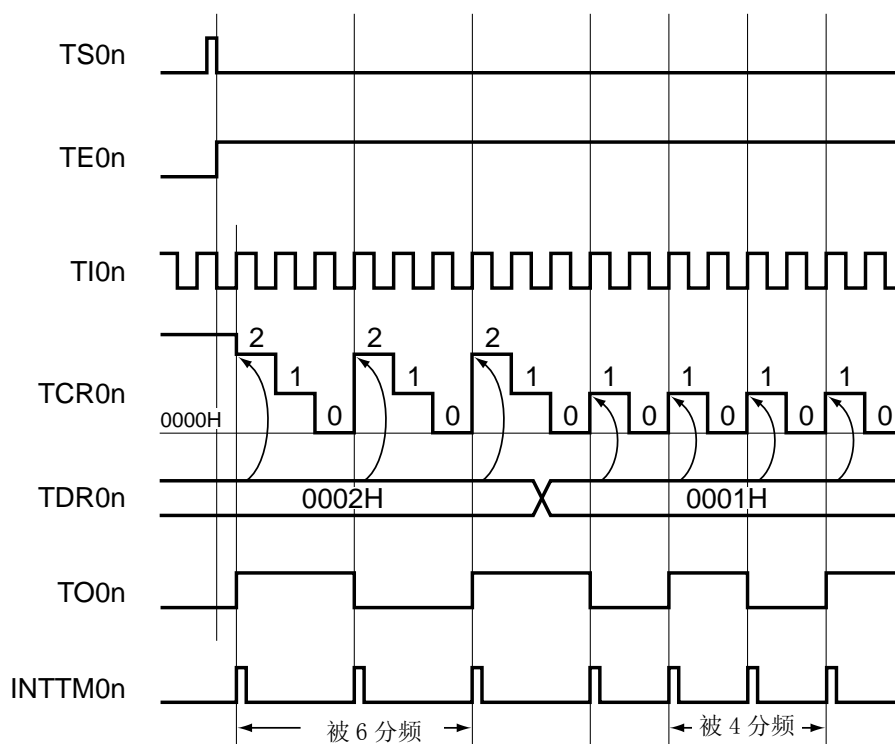
TDR0n 在任何时间可以重写。在下一个计数周期 TDR0n 的新值有效。

图 6-43. 用作分频操作的框图



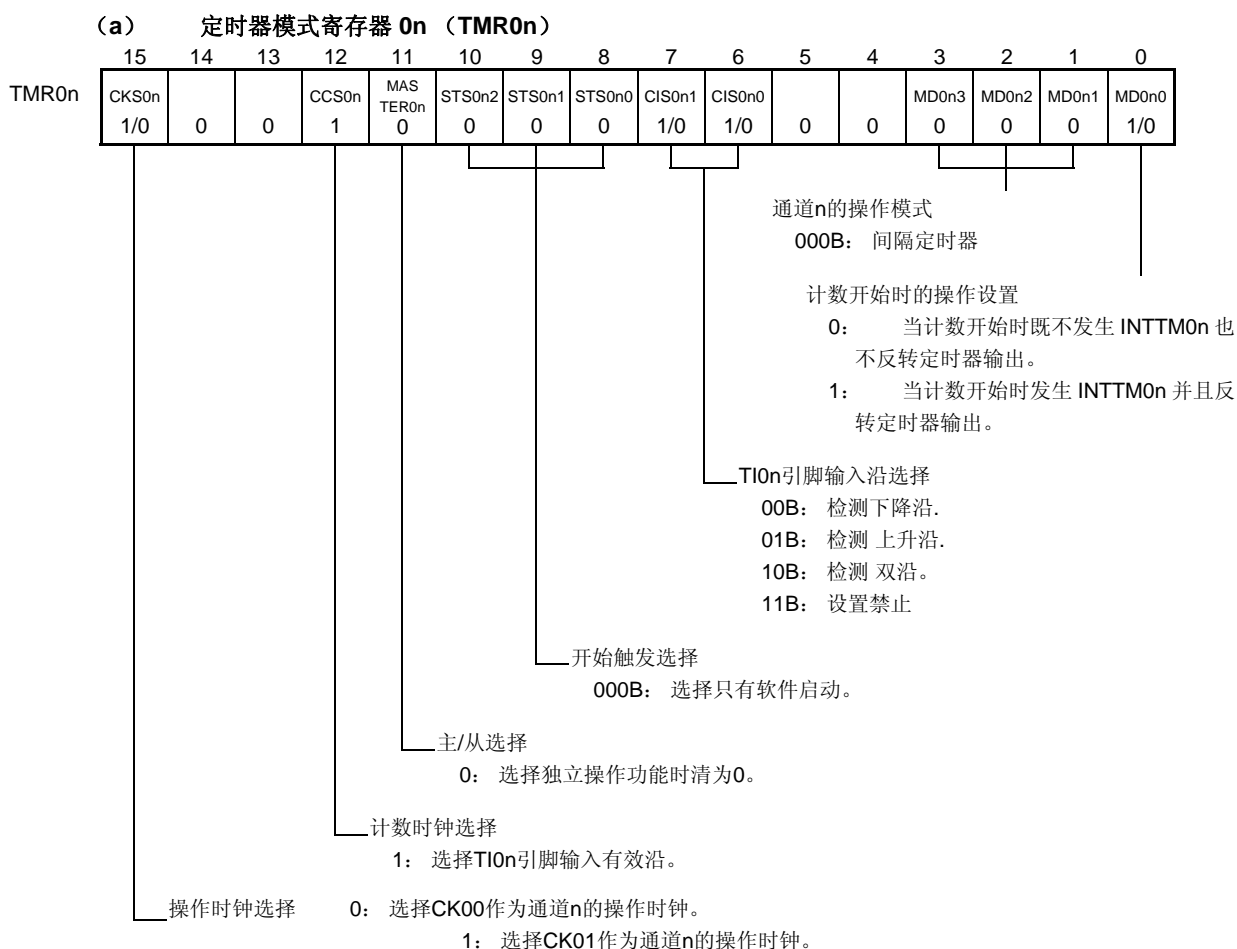
备注 n = 0~7

图 6-44. 作为分频器 (MD0n0 = 1) 操作的基本时序示例



备注 n = 0~7

图 6-45.作为分频器使用时寄存器设置的示例



(b) 定时器输出 寄存器 0 (TO0)

	第 n 位								
TO0	TO0n	1/0	0:	从 TO0n 输出 0。	1:	从 TO0n 输出 1。			

(c) 定时器输出允许寄存器 0 (TOE0)

	第 n 位								
TOE0	TOE0n	1/0	0:	通过计数操作停止 TO0n 输出。	1:	通过计数操作允许 TO0n 输出。			

(d) 定时器输出电平寄存器 0 (TOL0)

	第 n 位								
TOL0	TOL0n	0	0:	TOM0n = 0 时清零 (反转模式)					

(e) 定时器输出模式寄存器 0 (TOM0)

	第 n 位								
TOM0	TOM0n	0	0:	设置反转模式					

备注 n = 0~7

图 6-46. 当使用分频功能时的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。 设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道默认设置	设置 TMR0n 寄存器 (决定通道的操作模式)。 设置间隔 (周期) 值到 TDR0n 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	将 TOM0 寄存器的 TOM0n 位清为 0 (反转模式)。 将 TOL0n 位清为 0。 设置 TO0n 位并且决定 TO0n 输出的默认值。 设置 TOE0n 为 1 并且允许 TO0n 的操作。 将端口寄存器和端口模式寄存器清为 0。	TO0n 引脚为 Hi-Z 输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为 0 时, TO0n 默认设置电平输出。 因为通道停止操作 TO0n 不改变。 TO0n 引脚输出 TO0n 设置电平。
操作开始	设置 TOE0n 为 1 (仅当操作恢复时)。 设置 TS0n 位为 1。 由于是触发位, TS0n 位自动恢复为 0。	TE0n = 1, 并且计数操作开始。 在计数时钟输入时 TDR0n 的值被加载到 TCR0n。INTTM0n 产生, 并且如果 TMR0n 寄存器的 MD0n0 位为 1 时 TO0n 执行反转操作。
	操作期间	计数器 (TCR0n) 向下计数。当计数值达到 0000H 时, TDR0n 的值被再次加载到 TCR0n, 并且计数操作继续。通过检测 TCR0n = 0000H, INTTM0n 产生并且 TO0n 执行反转操作。之后, 重复上面的操作。
操作停止	TT0n 位设置为 1。 由于是触发位, TT0n 位自动恢复为 0。	TE0n = 0, 并且计数操作停止。 TCR0n 保持计数值并停止。 TO0n 输出没有被初始化但保持当前状态。
	TOE0n 被清零 (0) 并且值被设置到 TO0 寄存器。	TO0n 引脚输出 TO0n 设置电平。
TAU 停止	要保持 TO0n 引脚输出电平 在值被保存到端口寄存器后清零 TO0n 位为 0。 当不需要保存 TO0n 引脚输出电平时 切换模式寄存器到输入模式。	TO0n 引脚输出电平由端口功能保持。 TO0n 引脚输出电平变为 Hi-Z 输出状态。
	PER0 寄存器的 TAU0EN 位被清为 0。	掉电状态 所有电路被初始化并且每个通道的 SFR 也被初始化。 (TO0n 位清为 0, 并且 TO0n 引脚设置为模式)。

操作恢复。

备注 n = 0~7

6.7.4 作为输入脉冲间隔测量操作

在 $TI0n$ 有效沿计数值可被捕捉，输入到 $TI0n$ 的脉冲间隔可被测量。
脉冲间隔可通过下面的公式计算。

$$TI0n \text{ 输入脉冲间隔} = \text{计数时钟周期} \times \left((1000H \times \text{TSR0n: OVF}) + (\text{TDR0n 的捕捉值} + 1) \right)$$

注意事项 通过 $TMR0n$ 寄存器的 $CKS0n$ 位选择的操作时钟 $TI0n$ 引脚输入被采样，因此错误与发生的操作时钟数相等。

$TCR0n$ 在捕捉模式下作为向上计数器操作。

当通道启动触发器 ($TS0n$) 设置为 1 时, $TCR0n$ 从 0000H 开始与计数时钟同步向上计数。

当 $TI0n$ 引脚输入的有效沿被检测到时, 计数值被发送 (捕捉) 到 $TDR0n$, 同时计数器 ($TCR0n$) 被清为 0000H, 并且 $INTTM0n$ 输出。如果此时计数器溢出, $TSR0n$ 寄存器的 OVF 位设置为 1。如果计数器不溢出, OVF 位被清零。之后, 重复上面的操作。

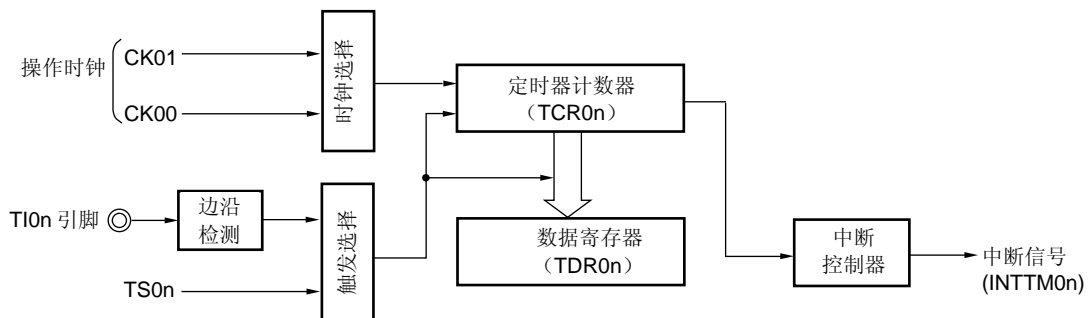
计数值一旦被捕捉到 $TDR0n$ 寄存器, $TSR0n$ 寄存器的 OVF 位就根据测量期间计数器是否溢出更新。因此, 可以检测到捕捉值的溢出状态。

如果计数器在两个以上周期达到满计数, 判断为一个溢出发生, $TSR0n$ 寄存器的 OVF 位设置为 1。因此 OVF 位作为累计标志, 如果一个溢出发生一次以上, 不能测量到正确的间隔值。

通过作为开始触发和捕捉触发的 $TI0n$ 的有效沿设置 $TMR0n$ 寄存器的 $STS0n2 \sim STS0n0$ 位为 001B。

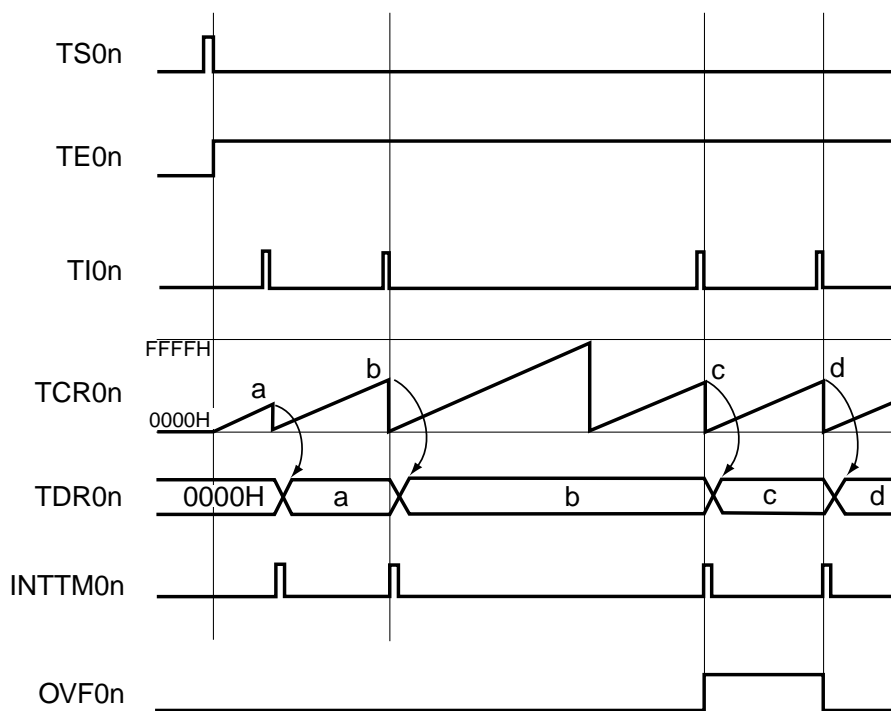
当 $TE0n = 1$, 代替 $TI0n$ 引脚输入, 软件操作 ($TS0n = 1$) 可作为捕捉触发。

图 6-47. 作为输入脉冲间隔测量操作的框图



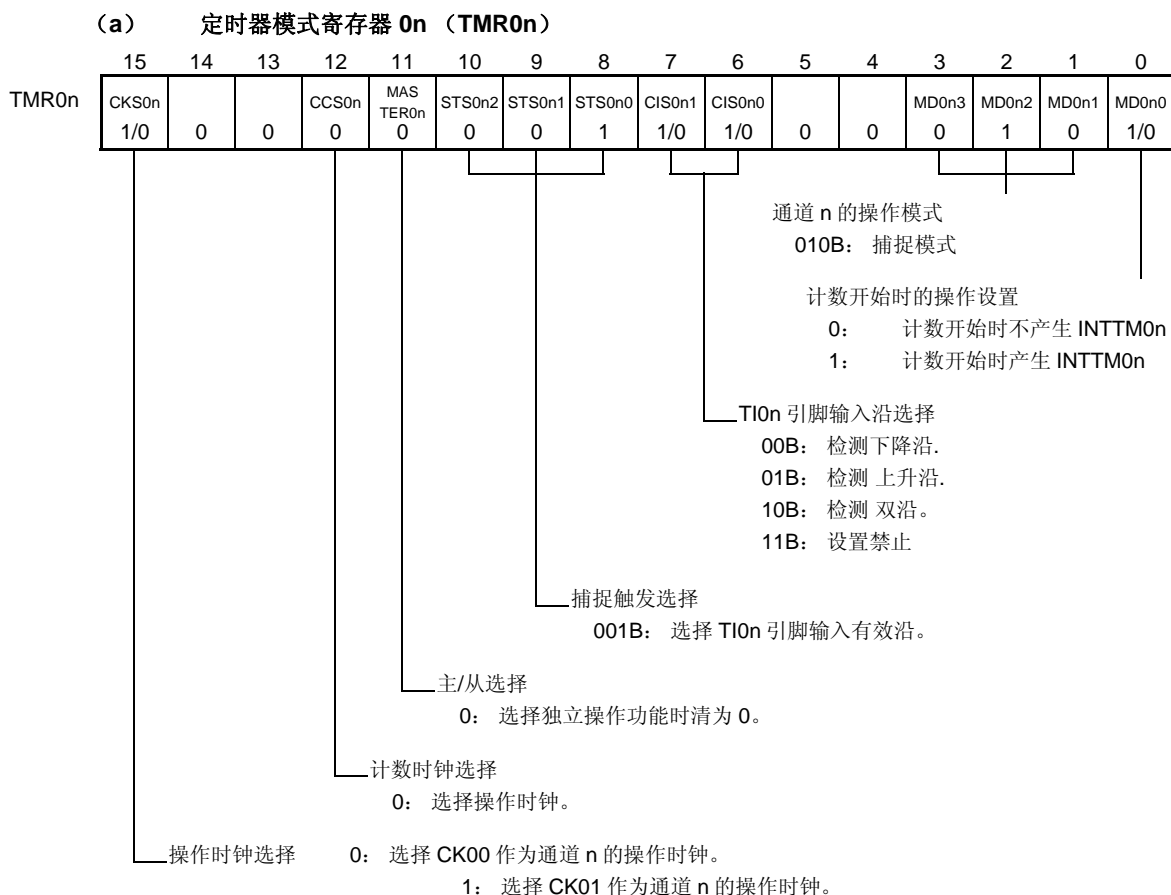
备注 $n = 0 \sim 7$

图 6-48. 作为输入脉冲间隔测量 (MD0n0 = 0) 操作的基本时序示例



备注 n = 0~7

图 6-49. 测量输入 脉冲间隔的寄存器设置的示例



(b) 定时器输出 寄存器 0 (TO0)

第 n 位
TO0n
0
0: 从 TO0n 输出 0。

(c) 定时器输出允许寄存器 0 (TOE0)

第 n 位
TOE0n
0
0: 通过计数操作停止 TO0n 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

第 n 位
TOL0n
0
0: TOM0n = 0 时清零 (反转模式)。

(e) 定时器输出模式寄存器 0 (TOM0)

第 n 位
TOM0n
0
0: 设置反转模式

备注 n = 0~7

图 6-50. 当使用输入脉冲间隔测量功能时的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	
通道默认设置	设置 TMR0n 寄存器 (决定通道的操作模式)。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置 TS0n 位为 1。 由于是触发位, TS0n 位自动恢复为 0。	TE0n = 1, 并且计数操作开始。 当计数时钟输入时 TCR0n 被清为 0000H。 当 TMR0n 寄存器的 MD0n0 位为 1 时, INTTM0n 发生。
操作期间	只有 TMR0n 寄存器的 CIS0n1 和 CIS0n0 的设置值可修改。 TDR0n 寄存器总是可读。 TCR0n 寄存器总是可读。 TSR0n 寄存器可读 TOM0, TOL0, TO0, 和 TOE0 寄存器的设置值不能修改。	计数器 (TCRn) 从 0000H 开始向上计数。当检测到 TI0n 引脚输入有效沿时, 计数值发送 (捕捉) 到 TDR0n。此时, TCR0n 清为 0000H, 并且 INTTM0n 信号发生。如果此时发生溢出, TSR0n 寄存器的 OVF 位被设置, 如果溢出不发生, OVF 位清零。之后, 重复上面的操作。
操作停止	TT0n 位设置为 1。 由于是触发位, TT0n 位自动恢复为 0。	TE0n = 0, 并且计数操作停止。 TCR0n 保持计数值并停止。 TSR0n 寄存器的 OVF 位不保存。
TAU 停止	PER0 寄存器的 TAU0EN 位被清为 0。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

操作恢复。

备注 n = 0~7

6.7.5 用作输入信号高/低电平宽度测量操作

通过在 TIO_n 的一个沿开始计数并且在另一个沿捕捉计数的序号，可以测量 TIO_n 的信号宽度（高电平宽度/低电平宽度）。TIO_n 的信号宽度可以由下面的公式计算出。

$$\text{TIO}_n \text{ 输入的信号宽度} = \text{计数时钟周期} \times ((10000\text{H} \times \text{TSR}_n: \text{OVF}) + (\text{TDR}_n \text{ 的捕捉值} + 1))$$

注意事项 通过 TMR_{0n} 寄存器的 CKS_{0n} 位选择的操作时钟 TIO_n 引脚输入被采样，因此误差与发生的操作时钟数相等。

TCR_{0n} 在捕捉&单计数模式下作为向上计数器。

当通道启动触发器（TS_{0n}）设置为 1，TE_{0n} 设置为 1 并且 TIO_n 引脚开始沿检测等待状态被设置。

当检测到 TIO_n 开始有效沿（当测量高电平宽度时 TIO_n 的上升沿），计数器与计数时钟同步向上计数。之后当检测到有效捕捉沿（当测量高电平宽度时 TIO_n 的下降沿），计数值发送到 TDR_{0n}，同时，INTTM_{0n} 输出。如果计数器在此时溢出，TSR_{0n} 寄存器的 OVF 位设置为 1。如果计数器没有溢出，OVF 位清零。当值为“发送到 TDR_{0n} 的值+1”时 TCR_{0n} 停止，设置 TIO_n 引脚开始沿检测等待状态。之后，重复上面的操作。

计数值一旦被捕捉到 TDR_{0n} 寄存器，TSR_{0n} 寄存器的 OVF 位就根据测量期间计数器是否溢出更新。因此，可以检测到捕捉值的溢出状态。

如果计数器在两个以上周期达到满计数，判断为一个溢出发生，TSR_{0n} 寄存器的 OVF 位设置为 1。因此 OVF 位作为累计标志，如果一个溢出发生一次以上，不能测量到正确的间隔值。

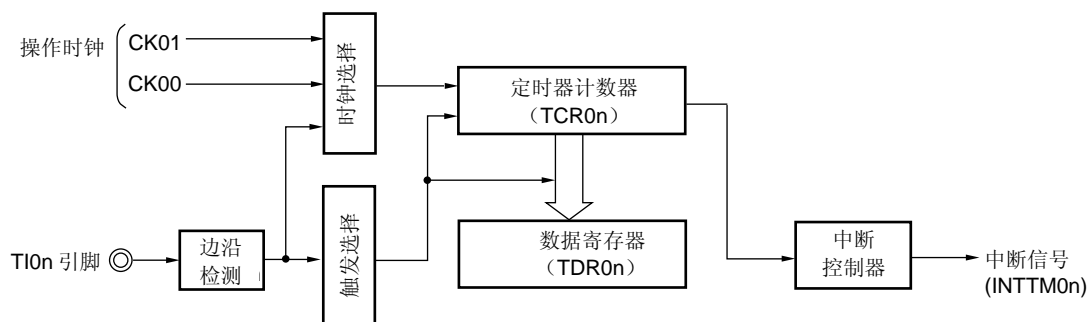
通过 TMR_{0n} 寄存器的 CIS_{0n1} 和 CIS_{0n0} 位可以选择测量 TIO_n 引脚的高电平宽度或低电平宽度。

因为用于测量 TIO_n 引脚输入的信号宽度的功能，当 TE_{0n} 为 1 时 TS_{0n} 不能设置为 1。

TMR_{0n} 的 CIS_{0n1}, CIS_{0n0} = 10B: 测量低电平宽度。

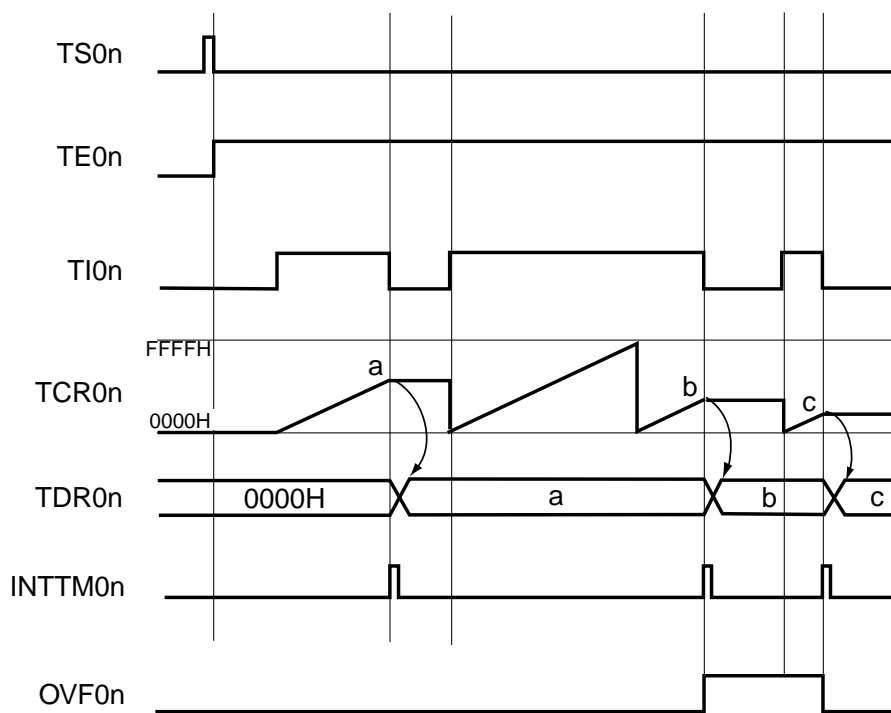
TMR_{0n} 的 CIS_{0n1}, CIS_{0n0} = 11B: 测量高电平宽度。

图 6-51.作为输入信号高/低电平宽度测量操作的框图



备注 n = 0~7

图 6-52. 作为输入信号高/低电平宽度测量操作的基本时序示例



备注 n = 0~7

图 6-54. 使用输入信号高/低电平宽度测量功能的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。 设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道默认设置	设置 TMR0n 寄存器 (决定通道的操作模式)。 TOE0n 清为 0 并且停止 TO0n 的操作。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置 TS0n 位为 1。 由于是触发位, TS0n 位自动恢复为 0。	TE0n = 1, 设置 TI0n 引脚开始沿检测等待状态。
	检测 TI0n 引脚输入计数开始有效沿。	TCR0n 清为 0000H 并且开始向上计数。
操作期间	TDR0n 寄存器的设置值不能改变。 TCR0n 寄存器总是可读。 TSR0n 寄存器不使用。 TMR0n, TOM0, TOL0, TO0, 和 TOE0 寄存器的设置值不能改变。	当 TI0n 引脚开始沿被检测到时, 计数器 (TCRn) 从 0000H 开始向上计数。如果检测到 TI0n 引脚的一个捕捉沿, 计数值发送到 TDR0n, 产生 INTTM0n。 如果此时发生溢出, 设置寄存器的 OVF 位; 如果没有溢出, OVF 位清零。直到下一个 TI0n 引脚开始沿被检测到 TCR0n 停止计数操作。
操作停止	TT0n 位设置为 1。 因为是触发位 TT0n 位自动返回 0。	TE0n = 0, 并且计数操作停止。 TCR0n 保持计数值并停止。 TSR0n 寄存器的 OVF 位也保持。
TAU 停止	PER0 寄存器的 TAU0EN 位清为 0。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

操作恢复。

备注 n = 0~7

6.8 定时器阵列单元的多通道操作

6.8.1 作为PWM 功能操作

两个通道可用做设置产生任意周期和占空比的脉冲。
输出 脉冲的周期和占空比可通过下面的公式计算出。

<p>脉冲周期= { TDR0n (主设备) 的设置值 + 1} × 计数时钟周期</p> <p>占空比 [%] = { DR0m (从设备) 的设置值} / { TDR0n (主设备) 的设置值 + 1} × 100</p> <p>0%输出: TDR0m (从设备) 的设置值 = 0000H</p> <p>100%输出: TDR0m (从设备) 的设置值 ≥ { TDR0n (主设备) 的设置值 + 1}</p>

备注 如果 TDR0m (从设备) 的设置值 > (TDR0n 的设置值 (主设备) + 1) 占空比超出 100%。作为 100%输出。

主通道操作在间隔定时器模式和计数周期。当通道启动触发器 (TS0n) 设置为 1 时, INTTM0n 输出。TCR0n 从 TDR0n 的加载值开始向下计数, 与计数时钟同步。当 TCR0n = 0000H 时, INTTM0n 输出。TCR0n 再次加载 TDR0n 的值。之后, 重复上面的操作。

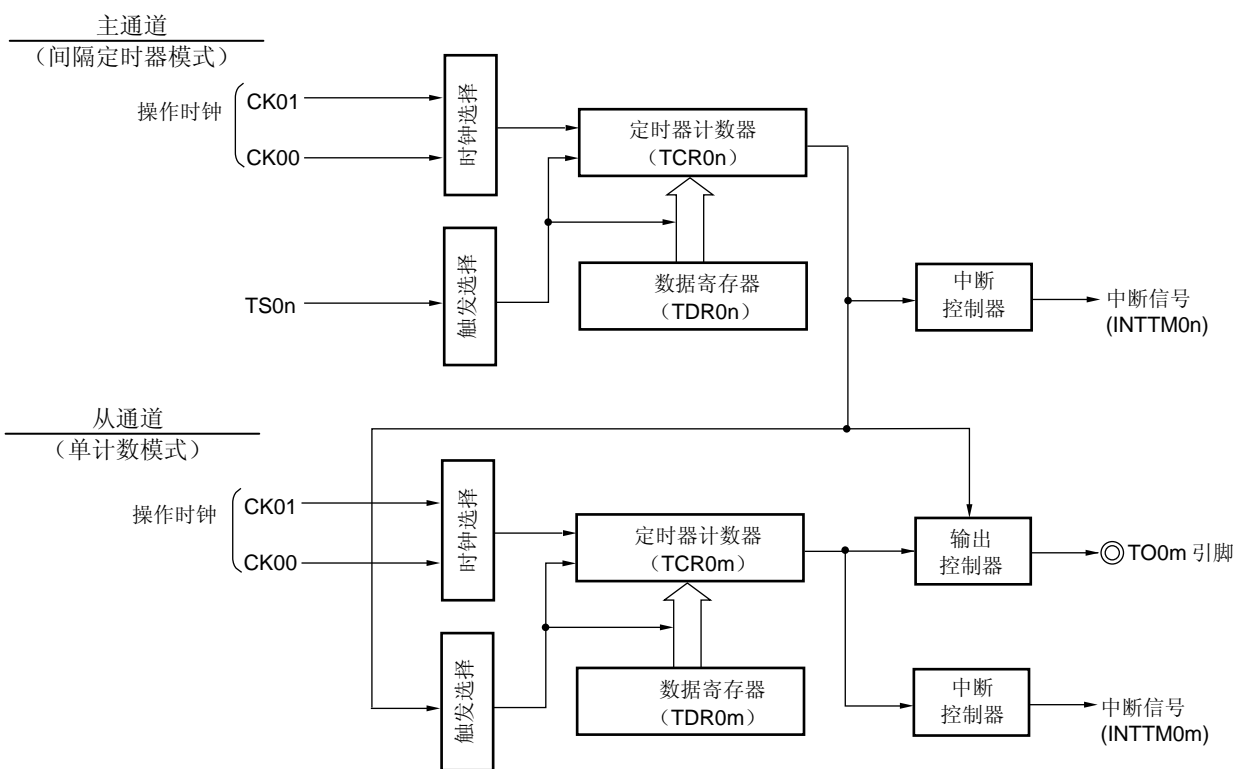
从通道的 TCR0m 操作在单计数模式下, 计数占空比, 和从 TO0m 引脚输出的 PWM 波形。从通道的 TCR0m 加载 TDR0m 的值, 通过主通道的 INTTM0n 作为开始触发, 并且停止计数知道下一个开始触发 (主通道的 INTTM0n) 输入。

TO0m 的输出电平在 INTTM0n 从主通道发生或变为有效的一个计数时钟, 当 TCR0m = 0000H 时无效。

注意事项 要重写主通道的 TDR0n 和从通道的 TDR0m, 需要两次写入访问。TDR0n 和 TDR0m 的值加载到 TCR0n 和 TRC0m 的时序在主通道的 INTTM0n 发生时。因此, 当修改分别主通道的 INTTM0n 发生之前和之后执行时, TO0m 引脚不能输出预期的波形。要修改主通道的 TDR0n 和从通道的 TDR0m, 必须在 INTTM0n 从主通道产生后立即修改寄存器。

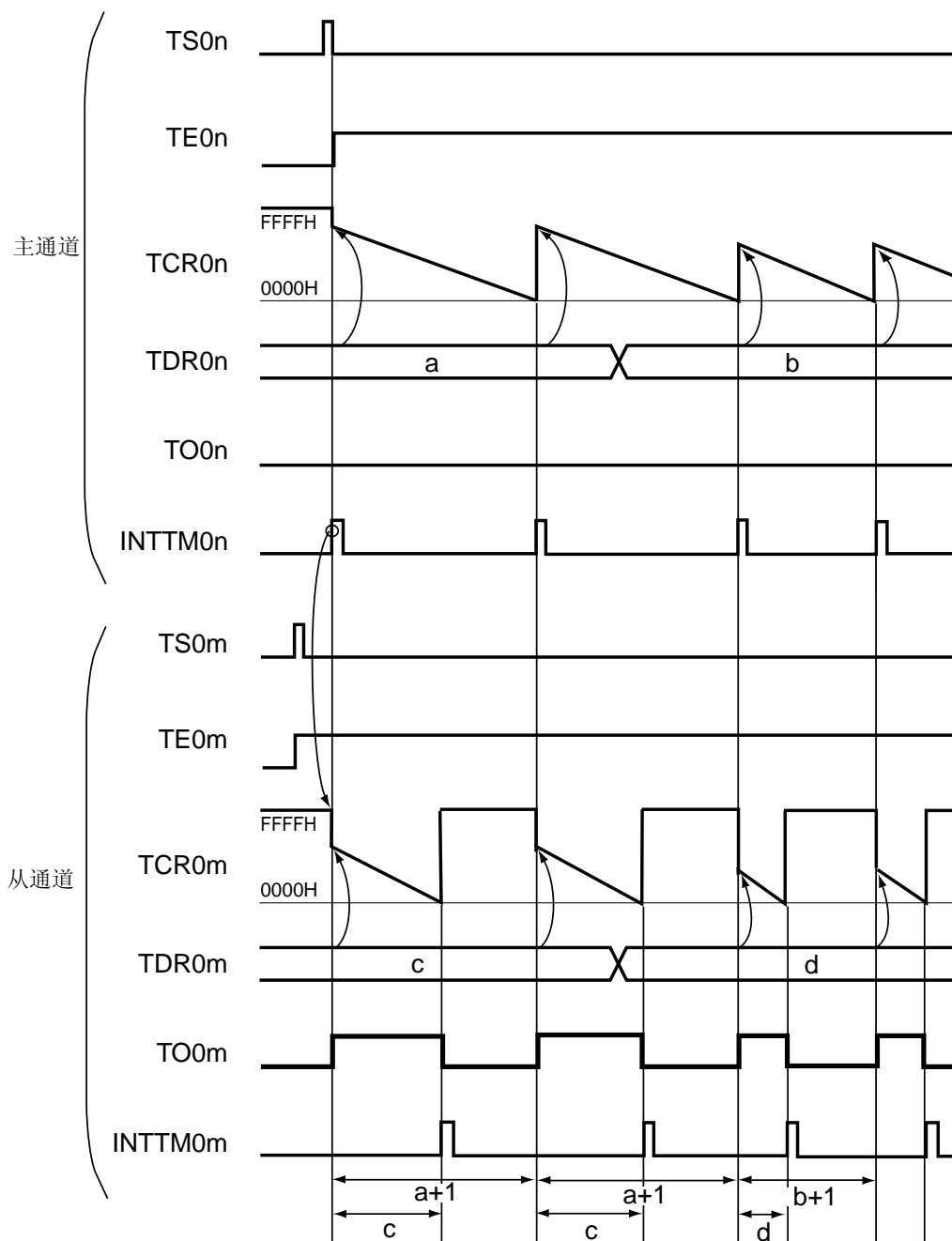
备注 n = 0, 2, 4, 6
m = n + 1

图 6-55. 作为 PWM 功能操作的框图



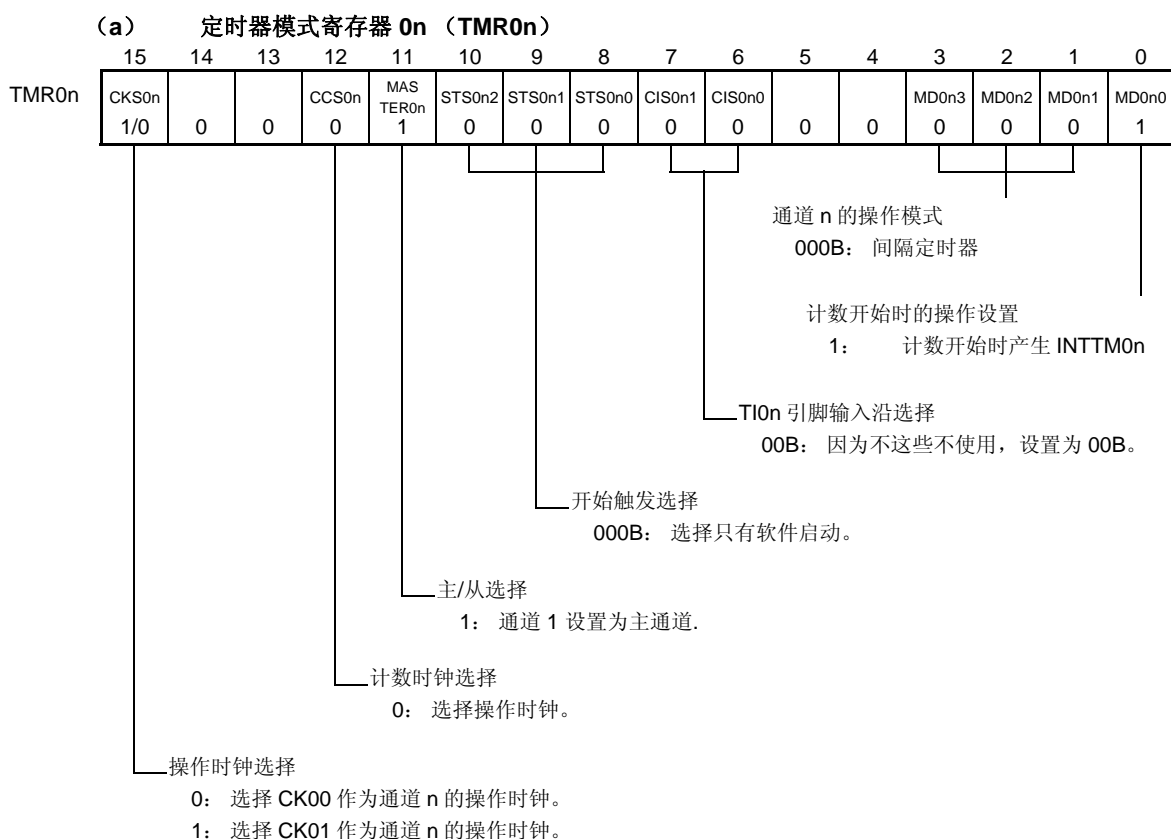
备注 $n = 0, 2, 4, 6$
 $m = n + 1$

图 6-56. 作为 PWM 功能操作的基本时序示例



备注 n = 0, 2, 4, 6
m = n + 1

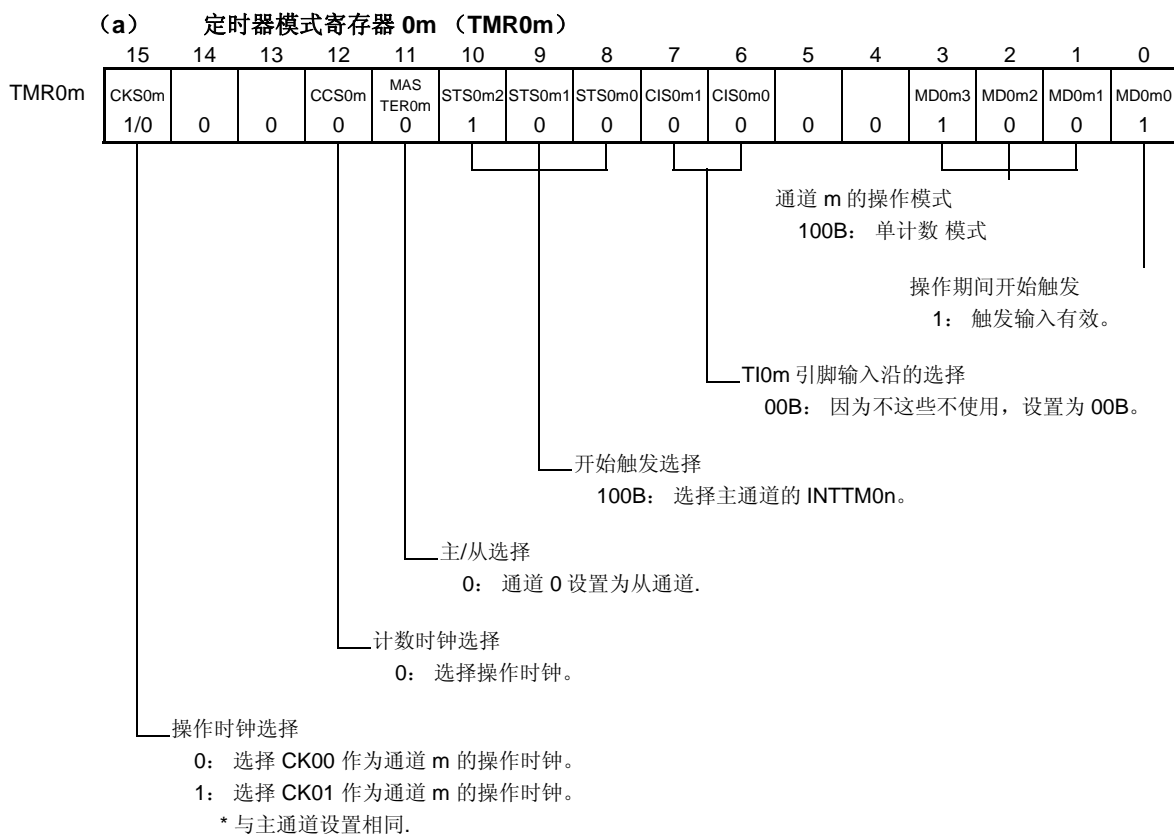
图 6-57. 当作为 PWM 功能（主通道）使用时寄存器设置的示例



- (b) 定时器输出寄存器 0 (TO0)
- | | | | |
|-----|-------|---|-----------------|
| | 第 n 位 | | |
| TO0 | TO0n | 0 | 0: 从 TO0n 输出 0。 |
- (c) 定时器输出允许寄存器 0 (TOE0)
- | | | | |
|------|-------|---|----------------------|
| | 第 n 位 | | |
| TOE0 | TOE0n | 0 | 0: 通过计数操作停止 TO0n 输出。 |
- (d) 定时器输出电平寄存器 0 (TOL0)
- | | | | |
|------|-------|---|--------------------------|
| | 第 n 位 | | |
| TOL0 | TOL0n | 0 | 0: TOM0n = 0 时清零 (反转模式)。 |
- (e) 定时器输出模式寄存器 0 (TOM0)
- | | | | |
|------|-------|---|-----------|
| | 第 n 位 | | |
| TOM0 | TOM0n | 0 | 0: 设置反转模式 |

备注 n = 0, 2, 4, 6

图 6-58. 当作为 PWM 功能（从通道）使用时寄存器设置的示例



(b) 定时器输出 寄存器 0 (TO0)

	第 m 位								
TO0	TO0m			0: 同 TO0m 输出 0。					
	1/0			1: 同 TO0m 输出 1。					

(c) 定时器输出允许寄存器 0 (TOE0)

	第 m 位								
TOE0	TOE0m			0: 通过计数操作停止 TO0m 输出。					
	1/0			1: 通过计数操作允许 TO0m 输出。					

(d) 定时器输出电平寄存器 0 (TOL0)

	第 m 位								
TOL0	TOL0m			0: 正向逻辑输出 (高有效)					
	1/0			1: 反向输出 (低有效)					

(e) 定时器输出模式寄存器 0 (TOM0)

	第 m 位								
TOM0	TOM0m			1: 设置组合操作模式。					
	1								

备注 n = 0, 2, 4, 6
m = n + 1

图 6-59. 当用作 PWM 功能时的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。 设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道 默认设置	设置两个通道的 TMR0n 和 TMR0m 寄存器要被使用 (决定通道的操作模式)。 一个间隔 (周期) 值设置到主通道的 TDR0n 寄存器, 占空比设置到从通道的 TDR0m 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 TOM0 寄存器的 TOM0m 位设置为 1 (组合操作模式)。 设置 TOL0m 位 设置 TO0m 位并决定 TO0m 输出的默认电平。 设置 TOE0m 为 1 并允许 TO0m 操作。 将端口寄存器和端口模式寄存器清为 0。	TO0n 引脚为 Hi-Z 输出状态。 当模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TO0n 默认设置电平。 因为通道停止操作 TO0m 不能修改。 TO0m 引脚输出 TO0m 的设置电平。

备注 n = 0, 2, 4, 6
m = n + 1

图 6-59. 当用作 PWM 功能时的操作过程 (2/2)

	软件操作	硬件状态	
操作恢复。	操作开始	<p>设置 TOE0m (从通道) 为 1 (仅当恢复操作时)。 TS0 寄存器的 TS0n (主通道) 和 TS0m (从通道) 位同时设置为 1。 因为是触发位 TS0n 和 TS0m 位自动返回为 0。</p>	<p>TE0n = 1, TE0m = 1 当主通道开始计数, 产生 INTTM0n。由此中断触发的, 从通道也开始计数。</p>
	操作期间	<p>TMR0n 和 TMR0m 寄存器的设置值不能修改。 TDR0n 和 TDR0m 寄存器的设置值在主通道的 INTTM0n 产生后可被修改。 TCR0n 和 TCR0m 寄存器一直可度。 TSR0n 和 TSR0m 寄存器不可用。 TOL0, TO0, 和 TOE0 寄存器的设置值不能修改。</p>	<p>主通道的计数器加载 TDR0n 的值到 TCR0n, 并向下计数。当计数值达到 TCR0n = 0000H, 产生 INTTM0n 输出。同时, TDR0n 寄存器的值加载到 TCR0n, 计数器开始再次向下计数。 在从通道, TDR0m 的值加载到 TCR0m, 由主通道的 INTTM0n 触发, 计数器开始向下计数。The 输出 level of TO0m 的输出电平在 INTTM0n 从主通道输出后的一个计数时钟有效。当 TCR0m = 0000H 时无效, 并且计数操作停止。 之后, 重复上面的操作。</p>
	操作停止	<p>TT0n (主通道) TT0m (从通道) 同时设置为 1。 因为是触发位 TT0n 和 TT0m 位自动返回为 0。</p>	<p>TE0n, TE0m = 0, 并且计数操作停止。 TCR0n 和 TCR0m 保持计数值并停止。 TO0m 输出 不被初始化而保持当前状态。</p>
	TAU 停止	<p>从通道的 TOE0m 清为 0, 值设置到 TO0m 寄存器。</p> <p>要保持 TO0m 引脚输出电平 TO0m 位在值保存到设置的端口寄存器后清为 0。 当不需要保持 TO0m 引脚输出电平时 切换端口模式寄存器到输入模式。</p> <p>PER0 寄存器的 TAU0EN 位被清为 0。</p>	<p>TO0m 引脚输出 TO0n 的设置电平。</p> <p>TO0m 引脚输出电平通过端口功能保持。</p> <p>TO0m 引脚输出电平变为 Hi-Z 输出状态。</p> <p>掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TO0m 位清为 0, TO0m 引脚设置为端口模式。)</p>

备注 n = 0, 2, 4, 6
m = n + 1

6.8.2 作为单脉冲输出功能操作

通过设置的两个通道，从输入到 TIO_n 引脚信号可产生任意延迟脉冲宽度的单脉冲。延迟时间和脉冲宽度可通过下面的公式计算出。

$$\begin{aligned} \text{延迟时间} &= \{ \text{TDR0n (主通道) 的设置值} + 2 \} \times \text{计数时钟周期} \\ \text{脉冲宽度} &= \{ \text{TDR0m (从通道) 的设置值} \} \times \text{计数时钟周期} \end{aligned}$$

主通道操作在单计数模式下，对延迟计数。主通道的 $TCR0_n$ 在检测到开始触发时开始操作， $TDR0_n$ 的值赋给 $TCR0_n$ 。 $TCR0_n$ 与计数时钟同步，由加载的 $TDR0_n$ 值开始递减计数。当 $TCR0_n = 0000H$ 时，输出 $INTTM0_n$ 并且停止计数直到下一个开始触发被检测。

从通道操作在单计数模式下，对脉冲宽度计数。通过主通道的 $INTTM0_n$ 作为开始触发，从通道的 $TCR0_m$ 开始计数，并加载 $TDR0_m$ 值。 $TCR0_m$ 与计数时钟同步，由加载的 $TDR0_m$ 值开始递减计数。当 $TCR0_m = 0000H$ 时，输出 $INTTM0_m$ 并停止计数直到下一个开始触发（主通道的 $INTTM0_n$ ）被检测到。 $TOOp$ 的输出电平在 $INTTM0_n$ 从主通道产生后的一个计数时钟后变为有效， $TCR0_m = 0000H$ 时无效。

代替 TIO_n 引脚输入，单脉冲输出也可通过软件操作（ $TS0_n = 1$ ）作为触发。

注意事项 主通道的 $TDR0_n$ 的时序与从通道的 $TDR0_m$ 不同。如果 $TDR0_n$ 和 $TDR0_m$ 在修改操作期间，则输出一个非法波形。必须在修改通道的 $INTTM0_n$ 发生后修改 $TDR0_n$ 和 $TDR0_m$ 。

备注 $n = 0, 2, 4, 6$
 $m = n + 1$

图 6-60. 作为单脉冲输出功能操作的框图

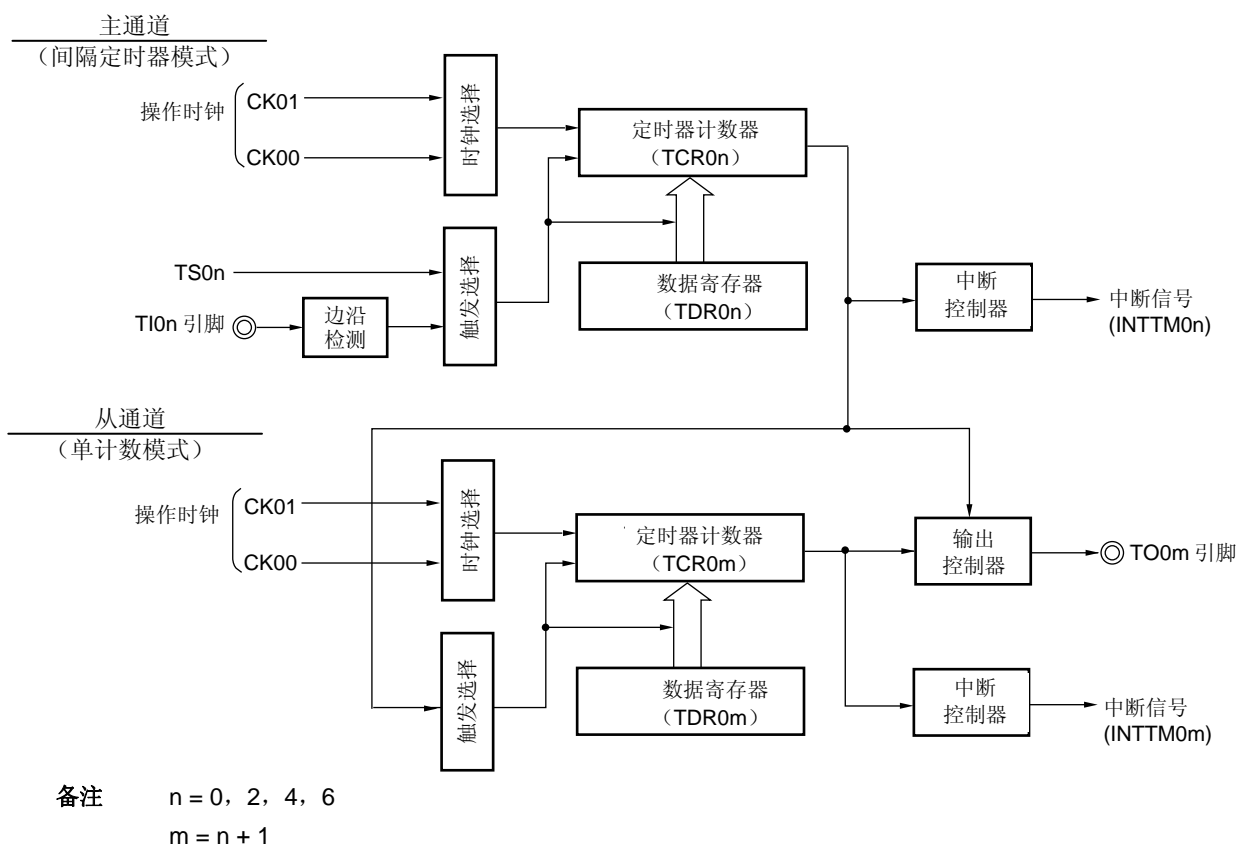
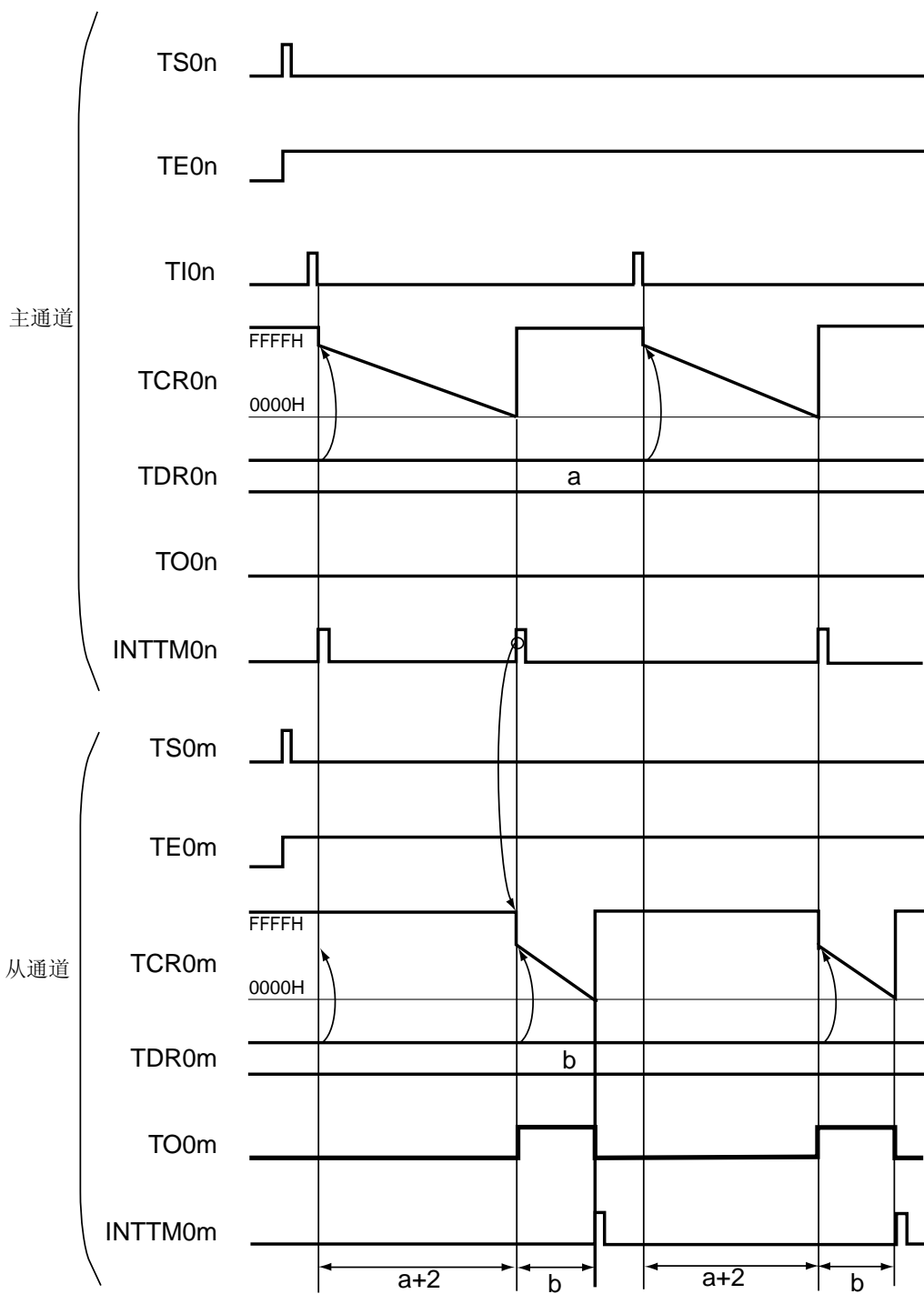
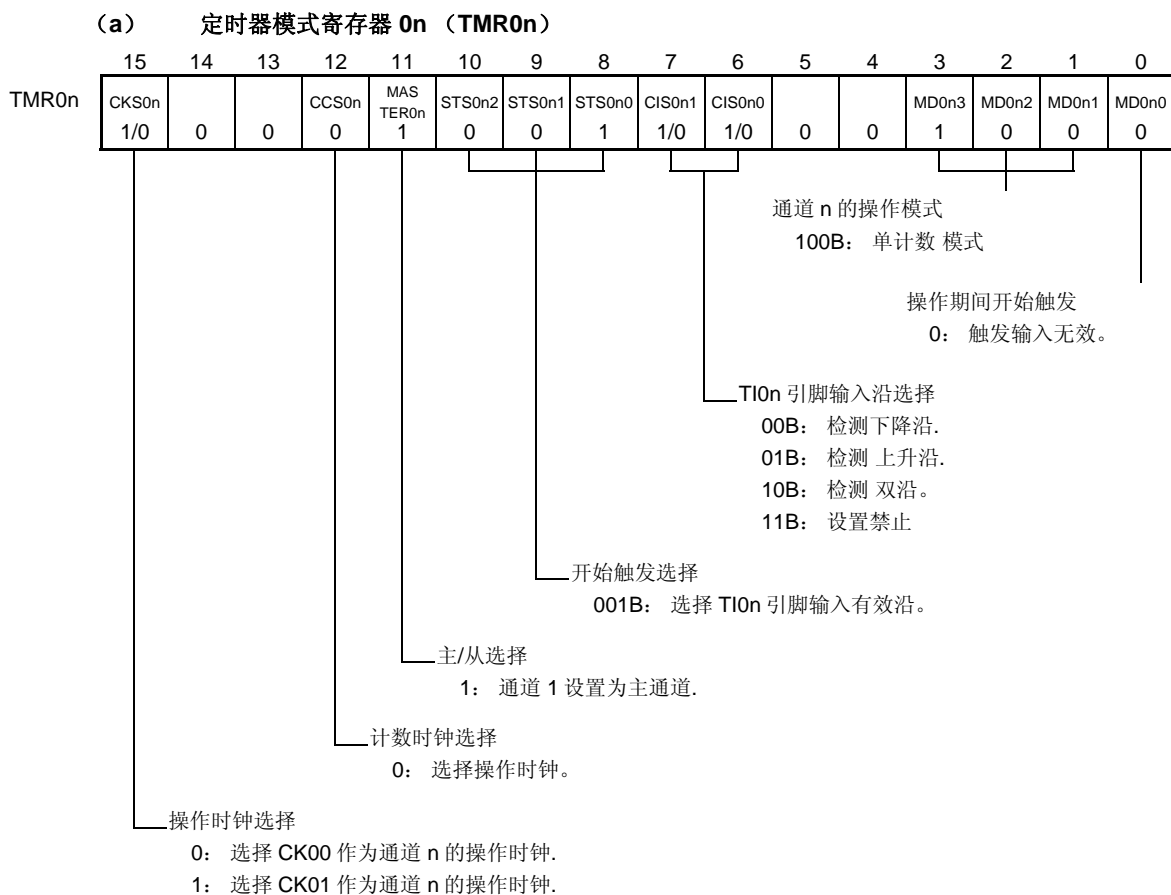


图 6-61. 作为单脉冲输出功能操作的基本时序示例



备注 $n = 0, 2, 4, 6$
 $m = n + 1$

图 6-62. 作为单脉冲输出功能操作的寄存器设置示例（主通道）



(b) 定时器输出 寄存器 0 (TO0)

第 n 位
TO0 TO0n
0 0: 从 TO0n 输出 0。

(c) 定时器输出允许寄存器 0 (TOE0)

第 n 位
TOE0 TOE0n
0 0: 通过计数操作停止 TO0n 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

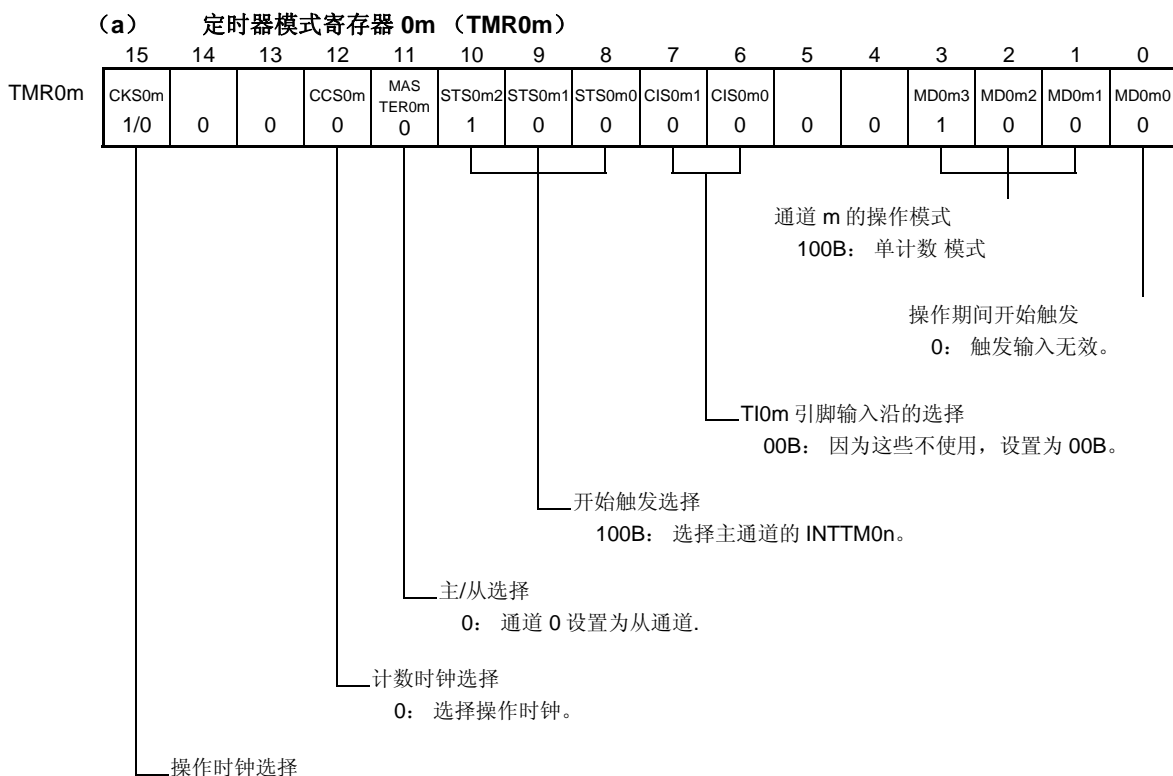
第 n 位
TOL0 TOL0n
0 0: TOM0n = 0 时清零 (反转模式)。

(e) 定时器输出模式寄存器 0 (TOM0)

第 n 位
TOM0 TOM0n
0 0: 设置反转模式

备注 n = 0, 2, 4, 6

图 6-63. 作为单脉冲输出功能操作的寄存器设置示例（从通道）



(b) 定时器输出 寄存器 0 (TO0)

	第 m 位		
TO0	TO0m	1/0	0: 从 TO0m 输出 0。 1: 从 TO0m 输出 1。

(c) 定时器输出允许寄存器 0 (TOE0)

	第 m 位		
TOE0	TOE0m	1/0	0: 通过计数操作停止 TO0m 输出。 1: 通过计数操作允许 TO0m 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

	第 m 位		
TOL0	TOL0m	1/0	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)

(e) 定时器输出模式寄存器 0 (TOM0)

	第 m 位		
TOM0	TOM0m	1	1: 设置组合操作模式。

备注 n = 0, 2, 4, 6
m = n + 1

图 6-64. 单脉冲输出功能的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	
通道 默认设置	设置要使用的两个通道的 TMR0n 和 TMR0m 寄存器 (决定通道的操作模式)。 输出延迟设置到主通道的 TDR0n 寄存器, 脉冲宽度设置到从通道的 TDR0m 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 TOM0 寄存器的 TOM0m 位设置为 1 (组合操作模式)。 设置 TOL0m 位。 设置 TO0m 位并决定 TO0m 输出的默认电平。 设置 TOE0m 为 1, 允许 TO0m 操作。 将端口寄存器和端口模式寄存器清为 0。	TO0n 引脚为 Hi-Z 输出状态。 当模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TO0n 默认设置电平。 因为通道停止操作 TO0m 不能修改。 TO0m 引脚输出 TO0m 的设置电平。

备注 n = 0, 2, 4, 6
 m = n + 1

图 6-64. 单脉冲输出功能的操作过程 (2/2)

	软件操作	硬件状态
操作开始	设置 TOE0m (从通道) 为 1 (仅当操作恢复时). TS0 寄存器的 TS0n (主通道) 和 TS0m (从通道) 位同时设置为 1。 因为触发位 TS0n 和 TS0m 位自动返回为 0。	TE0n 和 TE0m 设置为 1, 主通道进入 TI0n 输入沿检测等待状态。 计数器停止操作。
	检测 TI0n 引脚输入的主通道有效沿。	主通道开始计数。
操作期间	TMR0n 寄存器的 CISn1 和 CISn0 位设置值能修改。 TMR0m, TDR0n, TDR0m, 和 TOM0 寄存器的设置值不能修改。 TCR0n 和 TCR0m 寄存器一直可读。 TSR0n 和 TSR0m 寄存器不可用。 TOL0, TO0, 和 TOE0 寄存器的设置值能修改。	主通道的计数器加载 TDR0n 的值到 TCR0n, 并向下计数。当计数值达到 TCR0n = 0000H, 产生 INTTM0n 输出。同时, TDR0n 寄存器的值加载到 TCR0n, 计数器开始再次向下计数。 在从通道, TDR0m 的值加载到 TCR0m, 由主通道的 INTTM0n 触发, 计数器开始向下计数。TO0m 的输出电平在 INTTM0n 从主通道输出后的一个计数时钟有效。当 TCR0m = 0000H 时无效, 并且计数操作停止。之后, 重复上面的操作。
操作停止	TT0n (主通道) 和 TT0m (从通道) 位同时设置为 1。 因为触发位 TT0n 和 TT0m 位自动返回为 0。	TE0n, TE0m = 0, 并且计数操作停止。 TCR0n 和 TCR0m 保持计数值并停止。 TO0m 输出不被初始化而保持当前状态。
	从通道的 TOE0m 清为 0, 值设置到 TO0 寄存器。	TO0m 引脚输出 TO0n 的设置电平。
TAU 停止	要保持 TO0m 引脚输出电平 TO0m 位在值保存到设置的端口寄存器后清为 0。 当不需要保持 TO0m 引脚输出电平时, 切换端口模式寄存器到输入模式。	TO0m 引脚输出电平通过端口功能保持。 TO0m 引脚输出电平变为 Hi-Z 输出状态。
	PER0 寄存器的 TAU0EN 位被清为 0。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TO0m 位清为 0, TO0m 引脚设置为端口模式。)

操作恢复。

备注 n = 0, 2, 4, 6
m = n + 1

6.8.3 作为多路PWM输出功能操作

通过执行 PWM 功能和两个以上通道，可产生一些 PWM 输出信号。

例如，当使用两个从通道时，输出 脉冲的周期和占空比可通过下面的公式计算出。

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDR0n (主通道) 的设置值} + 1 \} \times \text{计数时钟周期} \\ \text{占空比 1 [\%]} &= \{ \text{TDR0m (从通道 1) 的设置值} \} / \{ \text{TDR0n (主通道) 的设置值} + 1 \} \times 100 \\ \text{占空比 2 [\%]} &= \{ \text{TDR0m (从通道 2) 的设置值} \} / \{ \text{TDR0n (主通道) 的设置值} + 1 \} \times 100 \end{aligned}$$

备注 如果 TDR0p (从通道 1) 的设置值 > (TDR0n 的设置值 (主通道) + 1) 或如果 TDR0q (从通道 2) 的设置值 > (TDR0n 的设置值 (主通道) + 1) 占空比超出 100%。作为 100% 输出。

主通道的 TCR0n 操作在间隔定时器模式下，对周期计数。

从通道 1 的 TCR0p 操作在单计数模式下，对占空比计数，从 TO0p 引脚输出输出 PWM 波形。TDR0p 的值赋给 TCR0p，主通道的 INTTM0n 作为开始触发，向下计数开始。当 TCR0p = 0000H 时，TCR0p 输出 INTTM0p，并且停止计数直到下一个开始触发 (主通道的 INTTM0n) 被输入。TO0p 的输出电平在 INTTM0n 从主通道产生后的一个计数时钟后变为有效，TCR0p = 0000H 时无效。

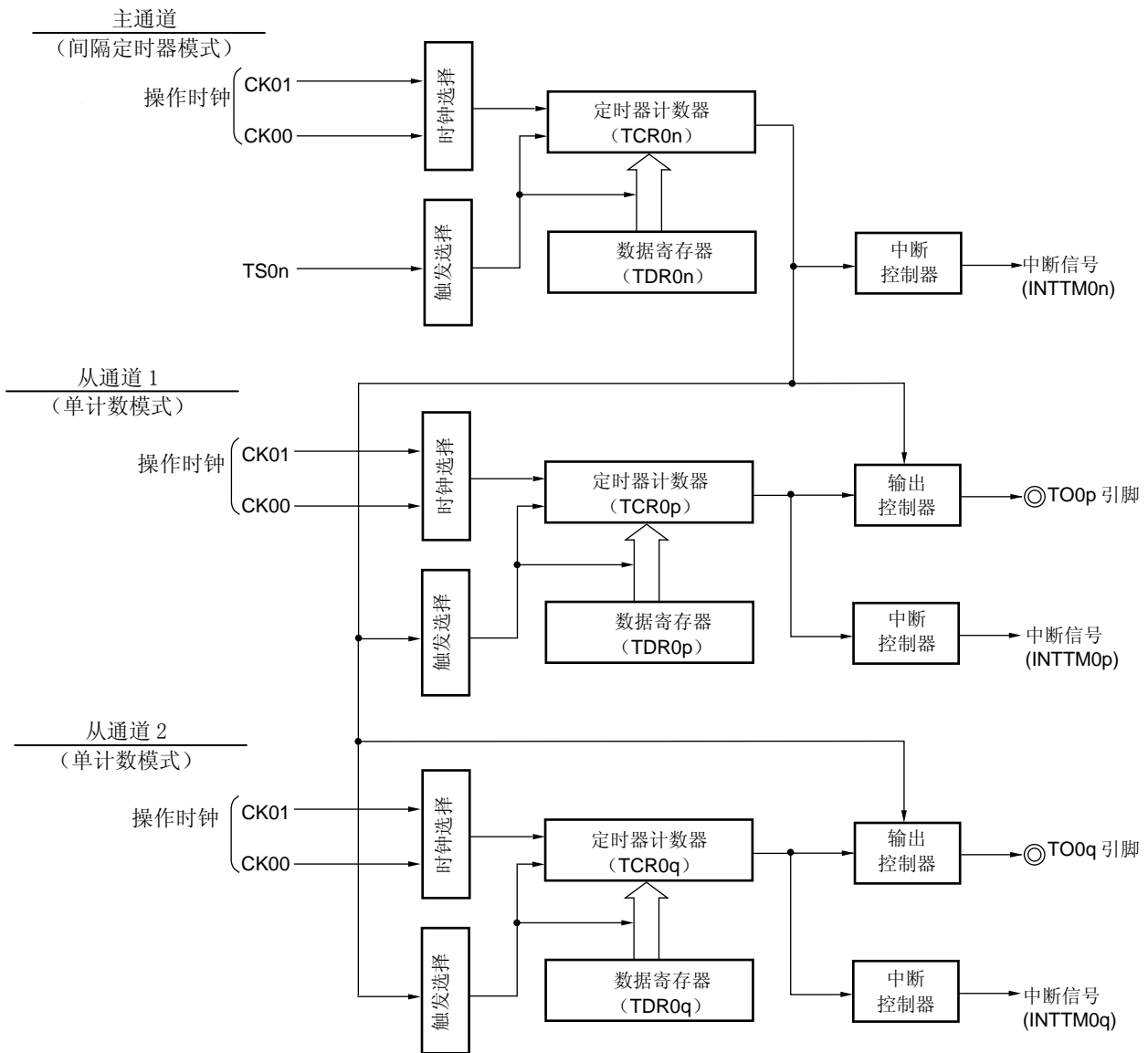
与从通道 1 的 TCR0p 相同，从通道 2 的 TCR0q 操作在单计数模式下，对占空比计数，从 TO0q 引脚输出输出 PWM 波形。TDR0q 的值赋给 TCR0q，主通道的 INTTM0n 作为开始触发，向下计数开始。当 TCR0q = 0000H 时，TCR0q 输出 INTTM0q，并且停止计数直到下一个开始触发 (主通道的 INTTM0n) 被输入。TO0q 的输出电平在 INTTM0n 从主通道产生后的一个计数时钟后变为有效，TCR0q = 0000H 时无效。

当通道 0 用作上面的主通道，最多可同时输出七种类型 PWM 信号。

注意事项 要修改主通道的 TDR0n 和从通道 1 的 TDR0p，写访问必须至少执行两次。因为在主通道的 INTTM0n 产生后 TDR0n 和 TDR0p 的值赋给 TCR0n 和 TCR0p，如果修改分别在主通道的 INTTM0n 发生之前和之后执行，TO0p 引脚可能不能输出预期的波形。要重写主通道的 TDR0n 和从通道的 TDR0p，必须在主通道的 INTTM0n 发生之后立即修改寄存器 (这也适用于从通道 2 的 TDR0q)。

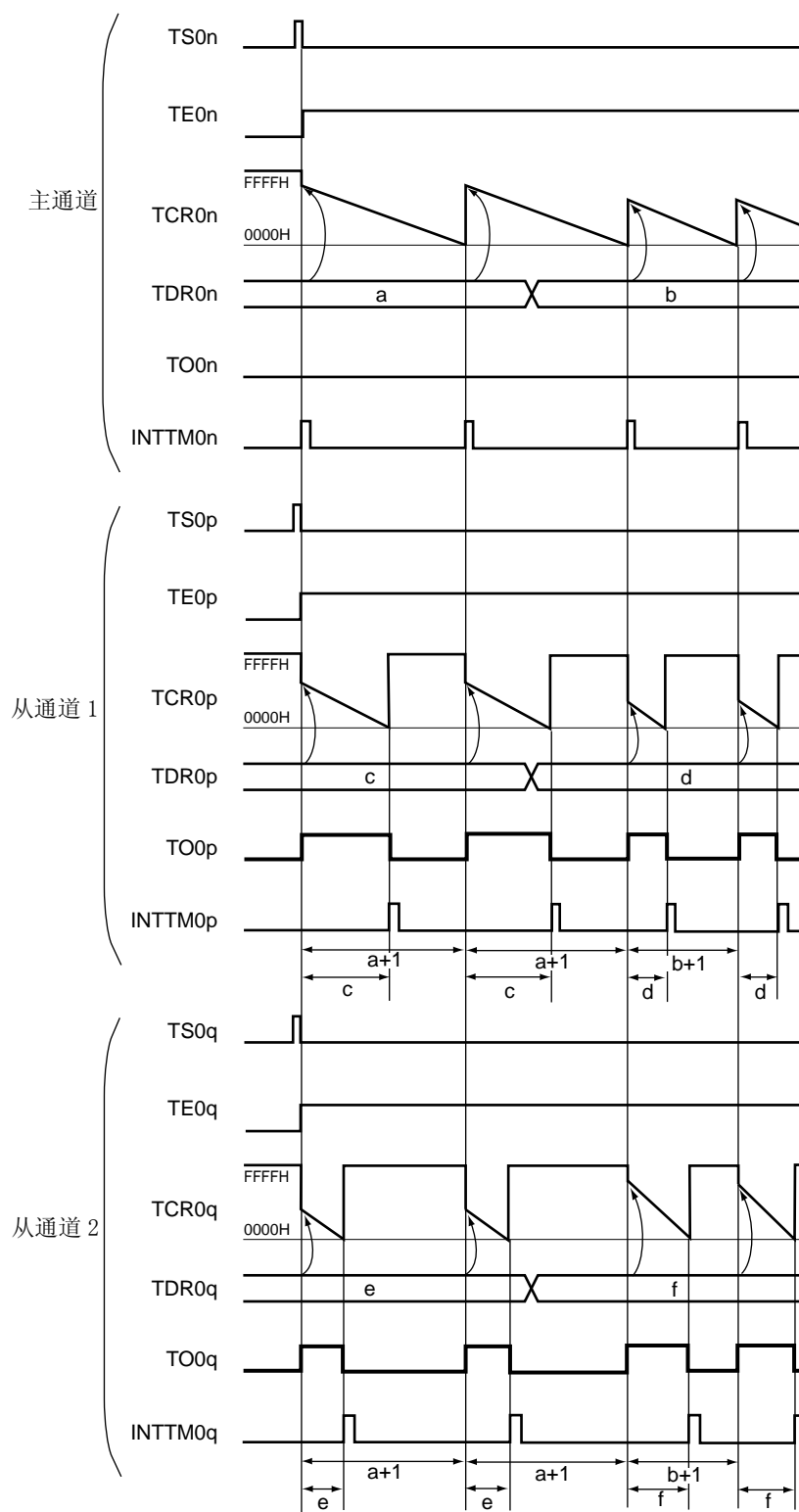
备注 1. $n = 0, 2, 4$
 $n < p < q \leq 7$
 p 和 q 是 n 后面的连续整数 ($p = n + 1, q = n + 2$)

图 6-65. 作为多路 PWM 输出功能操作的框图（输出两种类型 PWM）



备注 1. $n = 0, 2, 4$
 2. $p = n + 1$
 $q = n + 2$

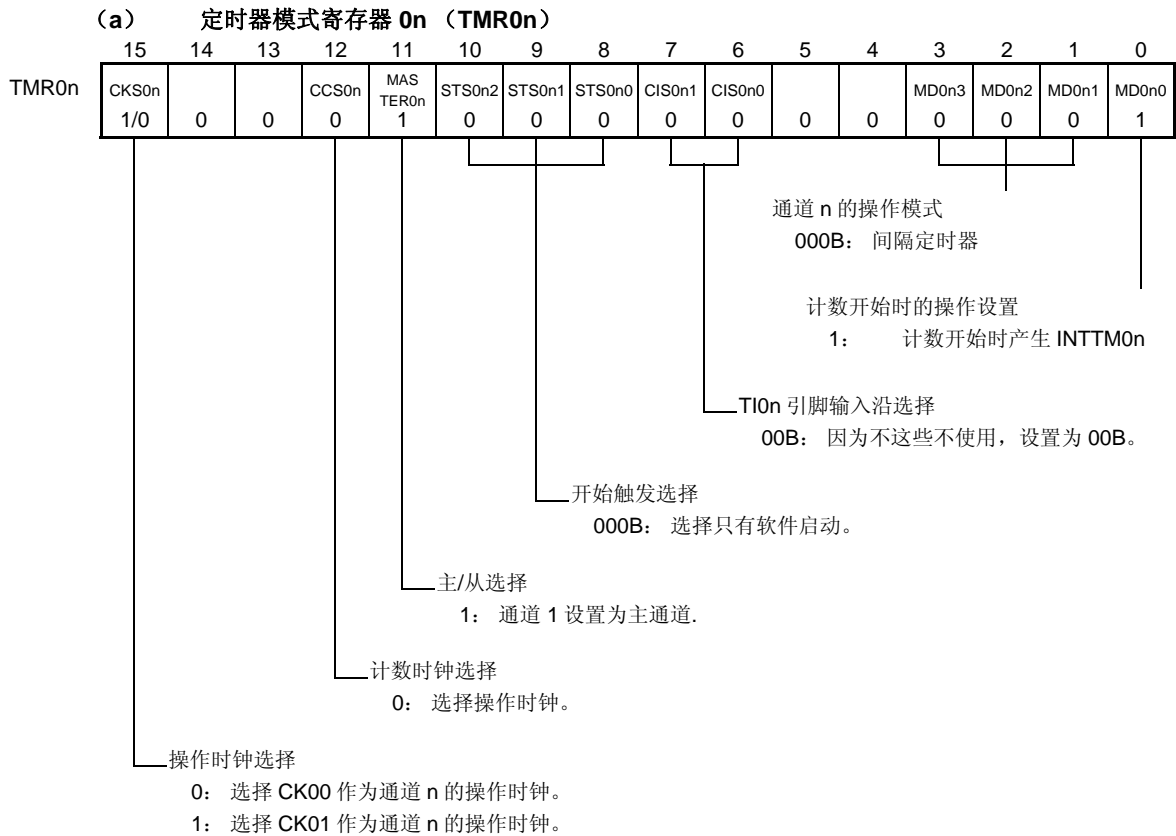
图 6-66. 作为多路 PWM 输出功能（输出两种类型 PWM）操作的基本时序示例



备注

1. $n = 0, 2, 4$
2. $p = n + 1$
- $q = n + 2$

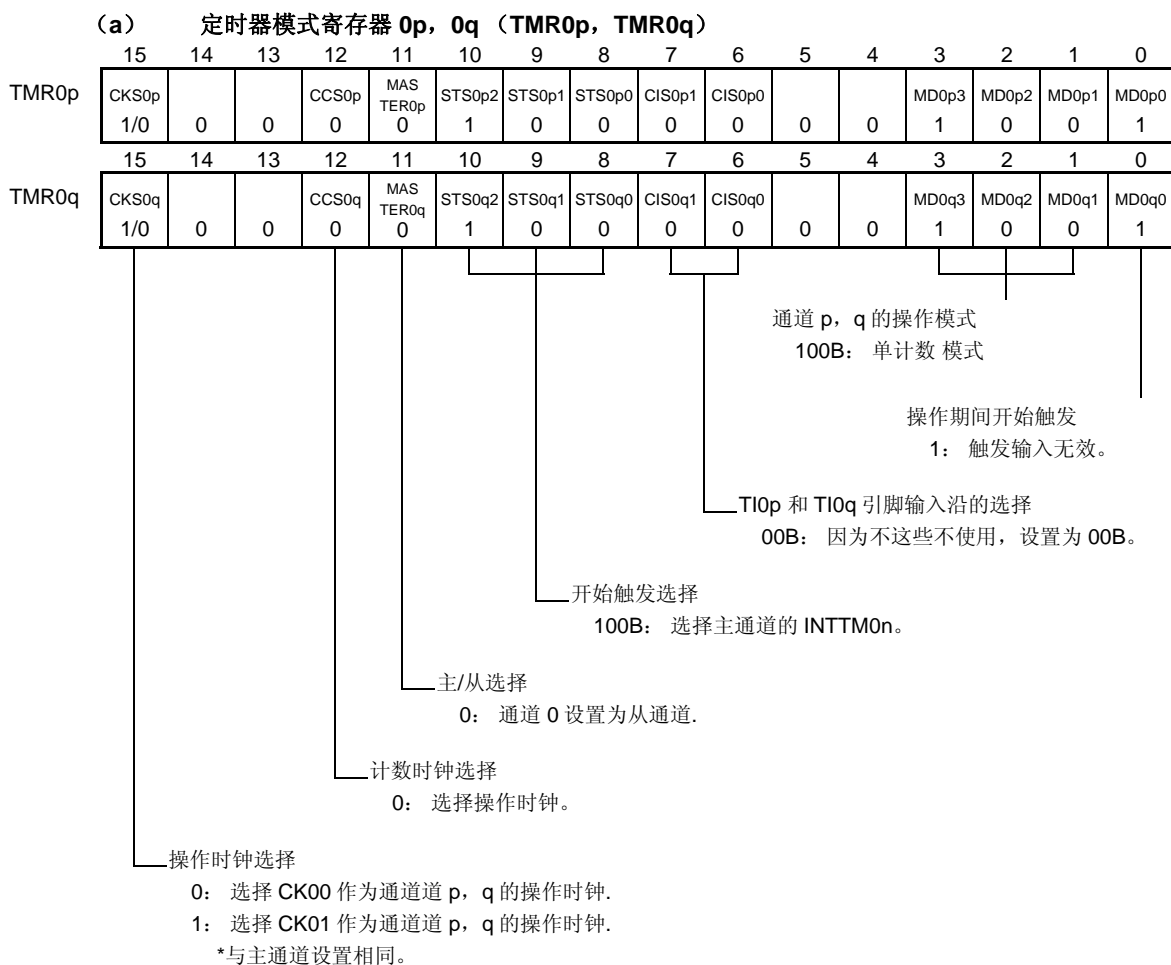
图 6-67. 当使用多路 PWM 输出 功能（主通道）时寄存器设置的示例



- (b) 定时器输出 寄存器 0 (TO0)
- | | | |
|-----|--------------------|-----------------|
| TO0 | 第 n 位
TO0n
0 | 0: 从 TO0n 输出 0。 |
|-----|--------------------|-----------------|
- (c) 定时器输出允许寄存器 0 (TOE0)
- | | | |
|------|---------------------|----------------------|
| TOE0 | 第 n 位
TOE0n
0 | 0: 通过计数操作停止 TO0n 输出。 |
|------|---------------------|----------------------|
- (d) 定时器输出电平寄存器 0 (TOL0)
- | | | |
|------|---------------------|--------------------------|
| TOL0 | 第 n 位
TOL0n
0 | 0: TOM0n = 0 时清零 (反转模式)。 |
|------|---------------------|--------------------------|
- (e) 定时器输出模式寄存器 0 (TOM0)
- | | | |
|------|---------------------|-----------|
| TOM0 | 第 n 位
TOM0n
0 | 0: 设置反转模式 |
|------|---------------------|-----------|

备注 n = 0, 2, 4

图 6-68. 当使用多路 PWM 输出 功能（从通道）时寄存器设置的示例（输出两种类型 PWM）



(b) 定时器输出 寄存器 0 (TO0)

	第 q 位	第 p 位	
TO0	TO0q	TO0p	0: 从 TO0p 或 TO0q 输出 0. 1: 从 TO0p 或 TO0q 输出 1.
	1/0	1/0	

(c) 定时器输出允许寄存器 0 (TOE0)

	第 q 位	第 p 位	
TOE0	TOE0q	TOE0p	0: 通过计数操作停止 TO0p 或 TO0q 输出。 1: 通过计数操作允许 TO0p 或 TO0q 输出。
	1/0	1/0	

(d) 定时器输出电平寄存器 0 (TOL0)

	第 q 位	第 p 位	
TOL0	TOL0q	TOL0p	0: 正向逻辑输出（高有效） 1: 反向输出（低有效）
	1/0	1/0	

(e) 定时器输出模式寄存器 0 (TOM0)

	第 q 位	第 p 位	
TOM0	TOM0q	TOM0p	1: 设置组合操作模式。
	1	1	

备注 n = 0, 2, 4; p = n+1; q = n+2

图 6-69. 当用作多路 PWM 输出功能时的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER0 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 决定 CK00 和 CK01 的时钟频率。	
通道默认设置	设置每个通道要使用的 TMR0n, TMR0p, 和 TMR0q 寄存器通道 (决定通道的操作模式)。 间隔 (周期) 值设置到主通道的 TDR0n 寄存器, 占空比设置到从通道的 TDR0m 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 TOM0 寄存器的 TOM0m 位设置为 1 (组合操作模式)。 将 TOL0p 和 TOL0q 位清为 0。 设置 TO0p 和 TO0q 位并决定 TO0p 和 TO0q 输出的默认电平。 设置 TOE0p 或 TOE0q 为 1, 并且允许 TO0m 的操作。 将端口寄存器和端口模式寄存器清为 0。	TO0n 引脚为 Hi-Z 输出状态。 当模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TO0p 和 TO0q 默认设置电平。 因为通道停止操作, TO0p 或 TO0q 不改变。 TO0p 和 TO0q 引脚输出 TO0p 和 TO0q 设置电平。

- 备注
1. n = 0, 2, 4
 2. p = n + 1; q = n + 2

图 6-69. 当用作多路 PWM 输出功能时的操作过程 (2/2)

	软件操作	硬件状态	
操作恢复。	操作开始	<p>设置 TOE0p 和 TOE0q (从通道) 为 1 (仅当恢复操作时)</p> <p>TS0 寄存器的 TS0n 位 (主通道), 和 TS0p 和 TS0q (从通道) 位 同时设置为 1。</p> <p>因为是触发位, TS0n, TS0p, 和 TS0q 位自动返回为 0。</p>	<p>TE0n = 1, TE0p, TE0q = 1</p> <p>当主通道开始计数, 产生 INTTM0n。由此中断触发的, 从通道也开始计数。</p>
	操作期间	<p>TMR0n, TMR0p, TMR0q, TOM0, 和 TOE0 寄存器的设置值不能修改。</p> <p>TDR0n, TDR0p, 和 TDR0q 寄存器的设置值在主通道的 INTTM0n 产生后可被修改。</p> <p>TCR0n, TCR0p, 和 TCR0q 寄存器一直可读。</p> <p>TSR0n, TSR0p, 和 TSR0q 寄存器不可用。</p> <p>TOM0, TOL0, TO0, 和 TOE0 寄存器的设置值不能修改。</p>	<p>主通道的计数器加载 TDR0n 的值到 TCR0n, 并向下计数。当计数值达到 TCR0n = 0000H, 产生 INTTM0n 输出。同时, TDR0n 寄存器的值加载到 TCR0n, 计数器开始再次向下计数。</p> <p>在从通道 1, TDR0p 的值发送到 TCR0p, 由主通道的 INTTM0n 触发, 计数器开始向下计数。TO0p 的输出电平在 INTTM0n 从主通道输出后的一个计数时钟有效。当 TCR0p = 0000H 时无效, 并且计数操作停止。</p> <p>在从通道 2, TDR0q 的值发送到 TCR0q, 由主通道的 INTTM0n 触发, 计数器开始向下计数。TO0q 的输出电平在 INTTM0n 从主通道输出后的一个计数时钟有效。当 TCR0q = 0000H 时无效, 并且计数操作停止。</p> <p>之后, 重复上面的操作。</p>
	操作停止	<p>TT0n 位 (主通道), TT0p, 和 TT0q (从通道) 位同时设置为 1。</p> <p>因为是触发位 TT0n, TT0p, 和 TT0q 位自动返回为 0。</p>	<p>TE0n, TE0p, TE0q = 0, 并且计数操作停止。</p> <p>TCR0n, TCR0p 和 TCR0q 保持计数值并停止。</p> <p>TO0p 和 TO0q 输出不被初始化而保持当前状态。</p>
		<p>从通道的 TOE0p 或 TOE0q 清为 0, 值设置到 TO0p 和 TO0q 寄存器。</p>	<p>TO0p 和 TO0q 引脚输出 TO0p 和 TO0q 的设置电平。</p>
	TAU 停止	<p>要保持 TO0m 引脚输出电平</p> <p>TO0m 位在值保存到设置的端口寄存器后清为 0。</p> <p>当不需要保持 TO0m 引脚输出电平时</p> <p>切换端口模式寄存器到输入模式。</p>	<p>TO0p 和 TO0q 引脚输出电平通过端口功能保持。</p>
	<p>PER0 寄存器的 TAU0EN 位被清为 0。</p>	<p>TO0p 和 TO0q 引脚输出电平变为 Hi-Z 输出状态。</p> <p>掉电状态</p> <p>所有电路被初始化, 并且每个通道的 SFR 也被初始化。(TO0p 和 TO0q 位清为 0, TO0p 和 TO0q 引脚设置为端口模式.)</p>	

备注

1. n = 0, 2, 4
2. p = n + 1; q = n + 2

第七章 实时计数器

7.1 实时计数器的功能

实时计数器有如下特点。

- 具有年、月、星期、天、小时、分钟和秒的计数器，可以共计 99 年。
- 固定周期中断功能（周期：1 月到 0.5 秒）
- 报警中断功能（报警：星期、小时、分钟）
- 间隔中断功能
- 1 Hz 的引脚输出功能
- 512 Hz 或 16.384 kHz 或 32.768 kHz 的引脚输出功能

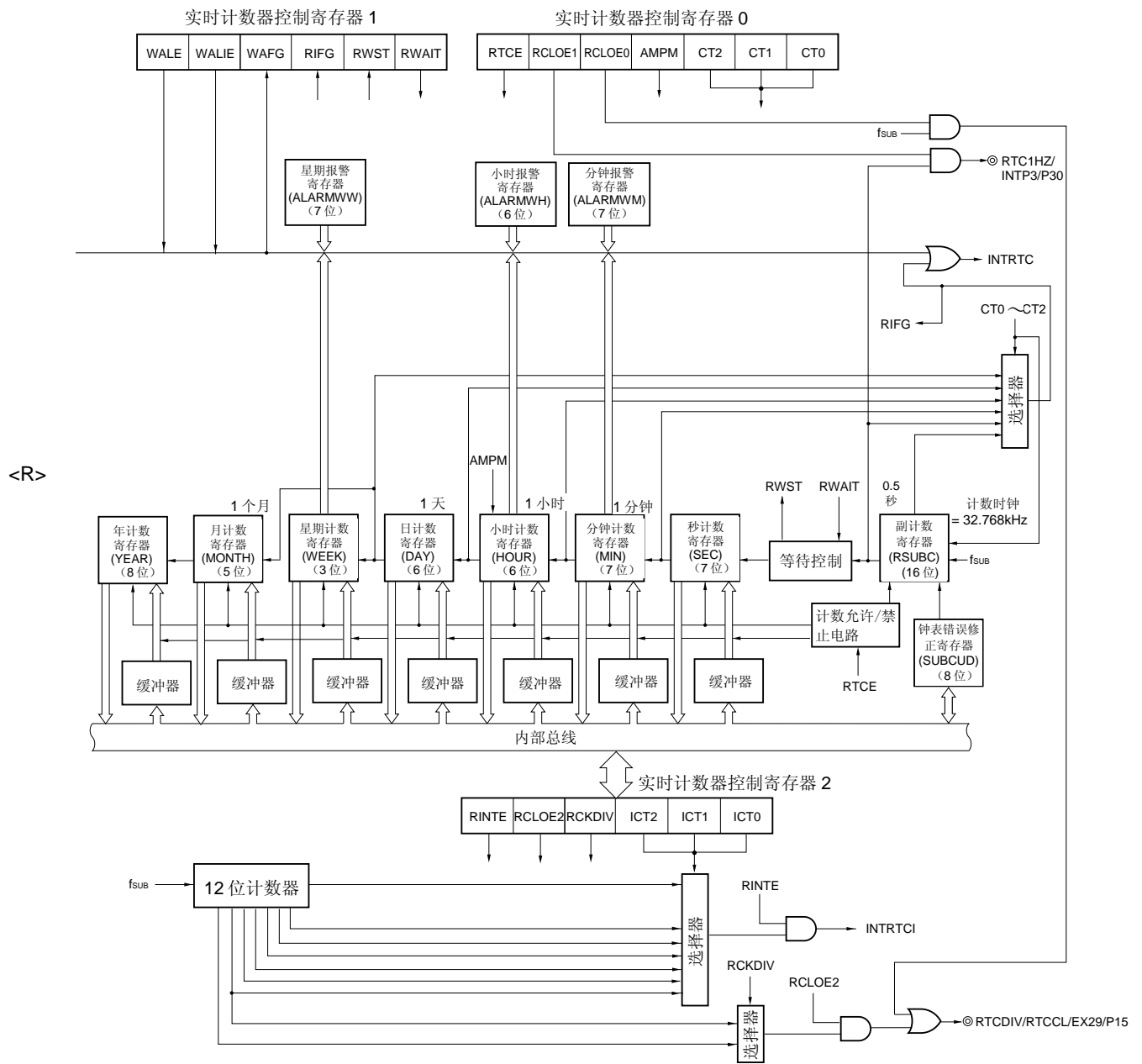
7.2 实时计数器的配置

实时计数器包括如下硬件。

表 7-1. 实时计数器的配置

项目	配置
控制寄存器	外围设备允许寄存器 0 (PER0)
	实时计数器控制寄存器 0 (RTCC0)
	实时计数器控制寄存器 1 (RTCC1)
	实时计数器控制寄存器 2 (RTCC2)
	副计数器寄存器 (RSUBC)
	秒计数器寄存器 (SEC)
	分计数器寄存器 (MIN)
	小时计数器寄存器 (HOUR)
	日期计数器寄存器 (DAY)
	星期计数器寄存器 (WEEK)
	月计数器寄存器 (MONTH)
	年计数器寄存器 (YEAR)
	钟表错误修正寄存器 (SUBCUD)
	分钟报警寄存器 (ALARMWM)
	小时报警寄存器 (ALARMWH)
	星期报警寄存器 (ALARMWW)

图 7-1. 实时计数器的框图



7.3 控制实时计数器的寄存器

以下 16 个寄存器用于控制实时计数器。

(1) 外围设备允许寄存器 0 (PER0)

PER0 用于允许或禁止每个外围硬件模块的使用。为了降低功耗和噪声，对于不使用的硬件模块停止提供时钟。

使用实时计数器时，请确保该寄存器的第 7 位 (RTCEN) 为 1。

PER0 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将寄存器清为 00H。

图 7-2. 外围设备允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	实时计数器 (RTC) 输入时钟的控制 ^注
0	停止提供输入时钟。 <ul style="list-style-type: none"> 由实时计数器 (RTC) 使用的 SFR 不能写入。 实时计数器 (RTC) 在复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> 由实时计数器 (RTC) 使用的 SFR 可读/写。

注 当从 CPU 访问实时计数器 (RTC) 使用的寄存器时，可通过 RTCEN 控制输入时钟。RTCEN 不能控制供给 RTC 的操作时钟 (f_{SUB})。

- <R>
- 注意事项 1.** 使用实时计数器时，首先确保设置 RTCEN 为 1。如果 RTCEN = 0，写入实时计数器的控制寄存器被忽略，并且，即使读取寄存器，只能读到默认值。
 - 2.** 确保将 PER0 寄存器的第 1 位清 0。

(2) 实时计数器控制寄存器 0 (RTCC0)

RTCC0 寄存器是 8 位寄存器，它用于启动或停止实时计数器操作、控制 RTCCL 和 RTC1HZ 引脚、设置 12 或 24 小时系统和固定周期中断功能。

RTCC0 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生对寄存器清零 (00H)。

图 7-3. 实时计数器控制寄存器 0 (RTCC0) 的格式

地址: FFF9DH 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	实时计数器操作控制
0	停止计数器操作。
1	启动计数器操作。

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

RCLOE0 [※]	RTCCL 引脚输出控制
0	禁止 RTCCL 引脚的输出 (32 kHz)。
1	允许 RTCCL 引脚的输出 (32 kHz)。

AMPM	选择12-/24小时系统
0	12 小时系统 (a.m. 和 p.m. 显示)
1	24 小时系统
<ul style="list-style-type: none"> • 改变 AMPM 的值, 要将 RWAIT (RTCC1 的第 0 位) 置为 1, 并重置小时计数寄存器 (HOUR)。 • 表 7-2 表示显示的时间数字显示。 	

CT2	CT1	CT0	选择固定周期中断 (INTRTC)
0	0	0	不用作固定周期中断功能
0	0	1	每 0.5 s 一次 (与秒累加同步)
0	1	0	每 1 s 一次 (与秒累加相同)
0	1	1	每 1 m 一次 (每分钟的 00 秒)
1	0	0	每 1 小时一次 (每小时的 00 分钟 00 秒)
1	0	1	每 1 天一次 (每天的 00 小时 00 分钟 00 秒)
1	1	x	每 1 月一次 (每月的第 1 天 a.m 00 小时 00 分 00 秒)
CT2 ~ CT0 值改变后, 将中断请求清零。			

注 不能同时使能 RCLOE0 和 RCLOE2。

注意事项 当 RTCE = 1 时, 如果改变了 RCLOE0 和 RCLOE1, 32 kHz 和 1 Hz 输出信号可能产生一个窄脉冲。

备注 x: 不必考虑

表 7-2. 时间数字显示

24 小时系统	12 小时系统	24 小时系统	12 小时系统
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

(3) 实时计数器控制寄存器 1 (RTCC1)

RTCC1 寄存器是 8 位寄存器，它用于控制报警中断功能和计数器等待时间。

RTCC1 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生对寄存器清零 (00H)。

图 7-4. 实时计数器控制寄存器 1 (RTCC1) 的格式 (1/2)

地址: FFF9EH 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	报警操作控制
0	相等操作无效
1	相等操作有效
设置报警寄存器 (RTCC1 的 WALIE 标志, ALARMWM 寄存器, ALARMWH 寄存器和 ALARMWW 寄存器), 禁止 WALE (将其清零)。	

WALIE	控制报警中断 (INTRTC) 功能操作
0	报警相等时不产生中断。
1	报警相等时产生中断。

WAFG	报警检测状态标志
0	报警不匹配
1	检测报警匹配
此状态标志表示检测到报警匹配。仅当 WALE = 1, 且在检测到报警匹配的一个时钟 (32 kHz) 后设置“1”才有效。此标志在写入“0”后清零, 写入“1”无效。	

图 7-4. 实时计数器控制寄存器 1 (RTCC1) 的格式 (2/2)

RIFG	固定周期中断状态标志
0	不产生固定周期中断。
1	产生固定周期中断。
此标志显示固定周期中断产生状态。当固定周期中断产生，该标志置“1”。 此标志在写入“0”后被清零，写入“1”是无效操作。	

RWST	实时计数器等待状态标志
0	计数器操作。
1	该模式可以读或写计数器值。
此状态标志表示对 RWAIT 的设置是否有效。 读或写计数器值前，确认此标志值为 1。	

RWAIT	等待控制实时计数器
0	设置计数器操作。
1	停止 SEC ~ YEAR 计数器。该模式可以读或写计数器值。
该位控制计数器操作。 确保写入“1”，读或写计数器值。 因为 RSUBC 操作继续，在 1 秒内完成读或写，并将此位清零。 当 RWAIT = 1 时，它占用 1 个时钟（32kHz）直到计数器值被读出或写入。 当 RWAIT = 1 时，如果 RSUBC 溢出，在 RWAIT = 0 之后该值递增。如果写入秒计数寄存器，因为 RSUBC 被清零，它将不能递增。	

注意事项 如果通过 1 位操作指令写入 WAFG 标志，RIFG 标志可能会清零。因此，必须使用 8 位操作指令来执行 WAFG 标志写入，并且同时将 RIFG 标志置 1 使写入无效。同样，使用 8 位操作指令来执行 RIFG 标志写入，并且将 WAFG 标志置 1。

备注 固定周期中断和报警匹配中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，在 INTRTC 发生的时刻，通过检查固定周期中断状态标志 (RIFG) 和报警检测状态标志 (WAFG)，来判断中断的正确产生。

(4) 实时计数器控制寄存器 2 (RTCC2)

RTCC2 寄存器是 8 位寄存器，它用于控制间隔中断功能和 RTCDIV 引脚。

RTCC2 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生对寄存器清零 (00H)。

图 7-5. 实时计数器控制寄存器 2 (RTCC2) 的格式

地址: FFF9FH 复位后: 00H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	选择间隔中断 (INTRTCI)
0	x	x	x	不产生间隔中断
1	0	0	0	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	x	$2^{12}/f_{XT}$ (125 ms)

当 RINTE = 0 时，更改 ICT2、ICT1 和 ICT0。

RCLOE2 ^注	RTCDIV 引脚输出控制
0	禁止 RTCDIV 引脚输出。
1	允许 RTCDIV 引脚输出。

RCKDIV	选择 RTCDIV 引脚输出频率
0	RTCDIV 引脚输出 512 Hz.
1	RTCDIV 引脚输出 16.384 kHz.

注 不能同时允许 RCLOE0 和 RCLOE2。

注意事项 当 RTCDIV 引脚输出停止，在最多 2 个 f_{XT} 时钟之后继续输出，并进入低电平。输出 512 Hz 时，当进入高电平之后立刻停止输出，可能产生一个脉冲，该脉冲宽度至少 1 个 f_{XT} 时钟。

(5) 副计数寄存器 (RSUBC)

RSUB 寄存器是 16 位寄存器，是实时计数器 1 秒的计数标准时间。它可置 0000H ~ 7FFFH 的值，使用 32.768 kHz 时钟来计数 1 秒钟。

RSUBC 可以 16 位存储器操作指令设置。

复位信号的产生对寄存器清零 (0000H)。

- 注意事项**
1. 当使用 SUBCU 寄存器进行纠正，值可能会大于 8000H。
 2. 由写入秒计数寄存器产生的复位，同样会将该寄存器清零。
 3. 在操作期间读取该寄存器，因为读取的值是不断变化的，所以不能保证读出值。

图 7-6. 副计数寄存器 (RSUBC) 的格式

地址: FFF90H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

地址: FFF91H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) 秒计数寄存器 (SEC)

SEC 寄存器是 8 位寄存器，为 0~59 (十进制) 值和表示秒的计数值。

当副计数器溢出它将开始递增计数。

要将数据写入该寄存器时，在最多 2 个时钟 (32.768kHz) 后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 0~59 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

SEC 可以通过 8 位存储器操作指令设置。

复位信号的产生对寄存器清零 (00H)。

图 7-7. 秒计数寄存器 (SEC) 的格式

地址: FFF92H 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

(7) 分钟计数寄存器 (MIN)

MIN 寄存器是 8 位寄存器，为 0~59（十进制）值和表示分钟的计数值。

当秒计数器溢出，它将开始计数。

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 0~59 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

MIN 可以通过 8 位存储器操作指令设置。

复位信号的产生对寄存器清零（00H）。

图 7-8. 分钟计数寄存器的格式 (MIN)

地址：FFF93H 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

(8) 小时计数寄存器 (HOUR)

HOUR 寄存器是 8 位寄存器，为 0~23 或 1~12（十进制）值和显示计数的小时值。

当分钟计数器溢出它将开始计数。

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 00~23、01~12 或 21~23 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

HOUR 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 12H。

如果当 AMPM 位（RTCC0 寄存器的第 3 位）置为 1，复位后该寄存器的值清零（00H）。

图 7-9. 小时计数寄存器 (HOUR) 的格式

地址：FFF94H 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

注意事项 当 AMPM = 0（如果选择 12 小时系统），HOUR 的第 5 位（HOUR20）显示 AM（0）/PM（1）。

(9) 日期计数寄存器 (DAY)

HOUR 寄存器是 8 位寄存器，为 31（十进制）值并表示天数的计数值。

当小时计数器溢出，它将开始计数。

计数器计数如下。

- 01 ~ 31（一月、三月、五月、七月、八月、十月、十二月）
- 01 ~ 30（四月、六月、九月、十一月）
- 01 ~ 29（二月、闰年）
- 01 ~ 28（二月、正常年）

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 00~31 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

DAY 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 01H。

图 7-10. 日期计数寄存器 (DAY) 的格式

地址: FFF96H 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

(10) 星期计数寄存器 (WEEK)

WEEK 寄存器是 8 位寄存器，为 0~6（十进制）值和显示计数的星期值。

当日计数器溢出，它将开始计数。

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 00~06 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

WEEK 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 00H。

图 7-11. 星期计数寄存器 (WEEK) 的格式

地址: FFF95H 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

(11) 月计数寄存器 (MONTH)

MONTH 寄存器是 8 位寄存器，为 1~12（十进制）值和显示计数的月值。

当日计数器溢出，它将开始计数。

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 01~12 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

MONTH 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 01H。

图 7-12. 月计数寄存器 (MONTH) 的格式

地址: FFF97H 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

(12) 年计数寄存器 (YEAR)

YEA 寄存器是 8 位寄存器，为 0~99（十进制）值和显示计数的月值。

当月计数器溢出，它将开始计数。

计数器计数如下。

00, 04, 08, ..., 92 和 96 显示闰年

要将数据写入该寄存器时，在最多 2 个时钟（32.768kHz）后，该数据被写入缓冲器然后写入计数器。以 BCD 码设置 00~99 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。

YEAR 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 00H。

图 7-13. 年计数寄存器 (YEAR) 的格式

地址: FFF98H 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

(13) 钟表错误修正寄存器 (SUBCUD)

该寄存器用作纠正副计数器 (RSUBC) 的计数值。

SUBCUD 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 00H。

图 7-14. 钟表错误修正寄存器 (SUBCUD) 的格式

地址: FFF99H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	设置钟表纠错时间
0	当秒数字为 00, 20 或 40 时纠正钟表错误。
1	仅当秒数字为 00 时纠正钟表错误。

F6	设置钟表纠错方法
0	增加的数字为 $\{ (F5, F4, F3, F2, F1, F0) - 1 \} \times 2$
1	减小的数字为 $\{ \langle /F5, /F4, /F3, /F2, /F1, /F0 \rangle + 1 \} \times 2$
当 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *) 时, 钟表错误不能修正。 /F5 ~ /F0 对应位反向值 (当 111100 时为 000011)。	

(14) 分钟报警寄存器 (ALARMWM)

该寄存器用于设置分钟报警。

ALARMWM 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 00H。

注意事项 以 BCD 码设置 00 ~ 59 十进制数到该寄存器。如果设置的值超出范围, 不能检测报警。

图 7-15. 分钟报警寄存器 (ALARMWM) 的格式

地址: FFF9AH 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

(15) 小时报警寄存器 (ALARMWH)

该寄存器用于设置小时报警。

ALARMWH 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 12H。

<R>

但是如果当 AMPM 位 (RTCC0 寄存器的第 3 位) 置为 1, 复位后该寄存器的值清零 (00H)。

注意事项 以 BCD 码设置 00 ~ 23、01 ~ 12 或 21 ~ 32 十进制数到该寄存器。如果设置的值超出范围, 不能检测报警。

图 7-16. 小时报警寄存器 (ALARMWH) 的格式

地址: FFF9BH 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

注意事项 如果 AMPM = 0 (选择 12 小时系统), ALARMWH 的第 5 位 (WH20) 显示 AM (0) /PM (1)。

(16) 星期报警寄存器 (ALARMWW)

该寄存器用于设置报警数据。

ALARMWW 可以通过 8 位存储器操作指令设置。

复位信号的产生将寄存器置为 00H。

注意事项 以 BCD 码设置 00 ~ 23、01 ~ 12 或 21 ~ 32 十进制数到该寄存器。如果设置的值超出范围，不能检测报警。

图 7-17. 星期报警寄存器 (ALARMWW) 的格式

地址: FFF9CH 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

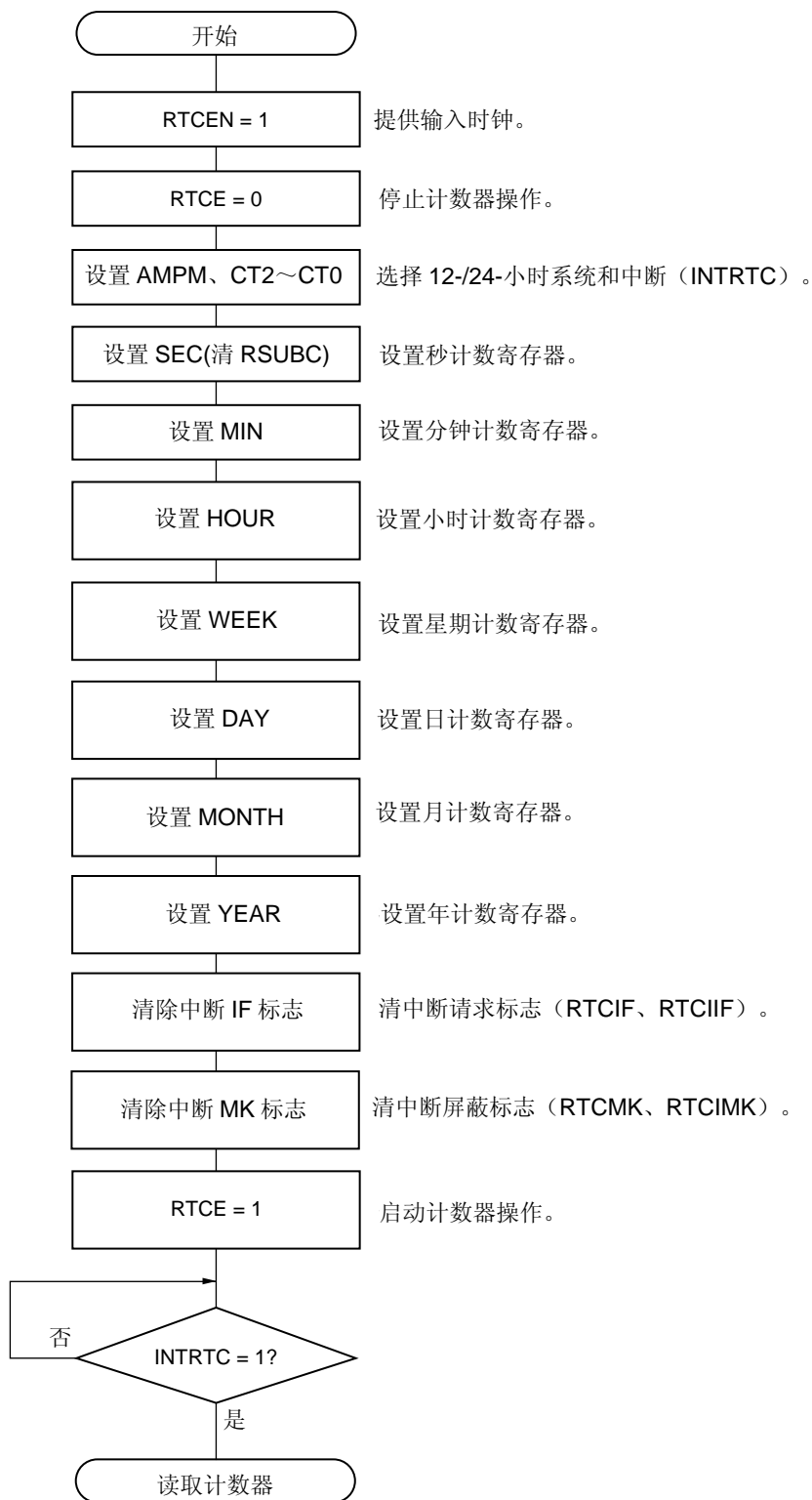
以下是设置报警的示例。

报警的时间	Day							12 小时显示				24 小时显示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	小时 10	小时 1	分钟 10	分钟 1	小时 10	小时 1	分钟 10	分钟 1
	W	W	W	W	W	W	W								
每天 0: 00 a.m.	0	1	2	3	4	5	6								
每天 1: 30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天 11: 59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一到星期五 0: 00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 1: 30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期 五 11: 59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.4 实时计数器操作

7.4.1 启动实时计数器的操作

图 7-18. 启动实时计数器的操作流程



<R> 注 当副系统时钟振荡稳定后，首先设置 RTCEN 为 1。

7.4.2 读/写实时计数器

当 RWAIT = 1 时，读出或写入该计数器。

图 7-19. 读出实时计数器的操作流程



注 设置 STOP 模式前，必须置 RWST = 0。

注意事项 在 1 秒内完成连续操作，置 RWAIT 为 1 到清 RWAIT 为 0。

备注 SEC, MIN, HOUR, WEEK, DAY, MONTH 和 YEAR 将顺次读出。
不用设置所有的寄存器，仅部分寄存器可能被读出。

图 7-20. 写入实时计数器的操作流程



注 设置 STOP 模式前，必须置 RWST = 0。

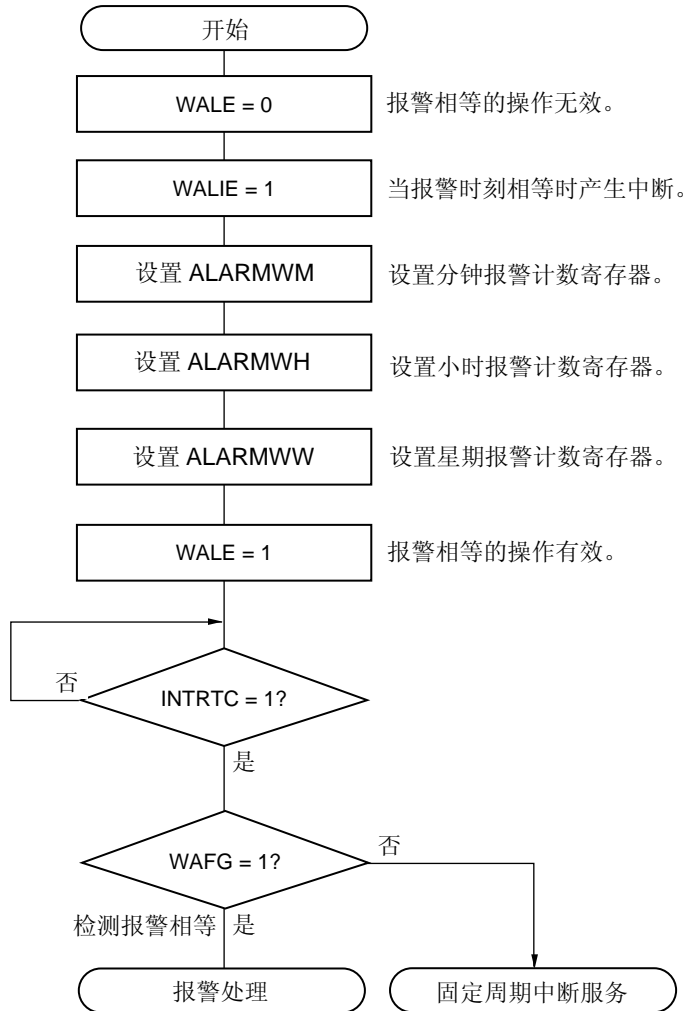
注意事项 在 1 秒内完成连续操作，置 RWAIT 为 1 到清 RWAIT 为 0。

备注 SEC, MIN, HOUR, WEEK, DAY, MONTH 和 YEAR 将顺次读出。
不用设置所有的寄存器，仅部分寄存器可能被读出。

7.4.3 设置实时计数器报警

当 WALE = 0 时，设置报警时间。

图 7-21. 报警设置流程图



备注

1. 可以按照任意顺序写入 ALARMWMM, ALARMWH 和 ALARMWW 寄存器。
2. 固定周期中断和报警匹配中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，在 INTRTC 发生的时刻，通过检查固定周期中断状态标志 (RIFG) 和报警检测状态标志 (WAFG)，来判断中断的正确产生。

第八章 看门狗定时器

8.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟。

看门狗定时器用于检测不期望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。

出现以下情况时认为检测到程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器允许寄存器 (WDTE) 执行 1 位操作指令。
- 如果将 “ACH” 以外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE

当由看门狗定时器产生复位时，复位控制标志寄存器 (RESF) 的第 4 位 (WDTRF) 置 1。需要了解 RESF 的详细信息，可以参阅 **第十九章 复位功能**。

当达到溢出时间的 75% 时，可以产生一个间隔中断。

8.2 看门狗定时器的配置

看门狗定时器包括如下硬件。

表 8-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器允许寄存器 (WDTE)

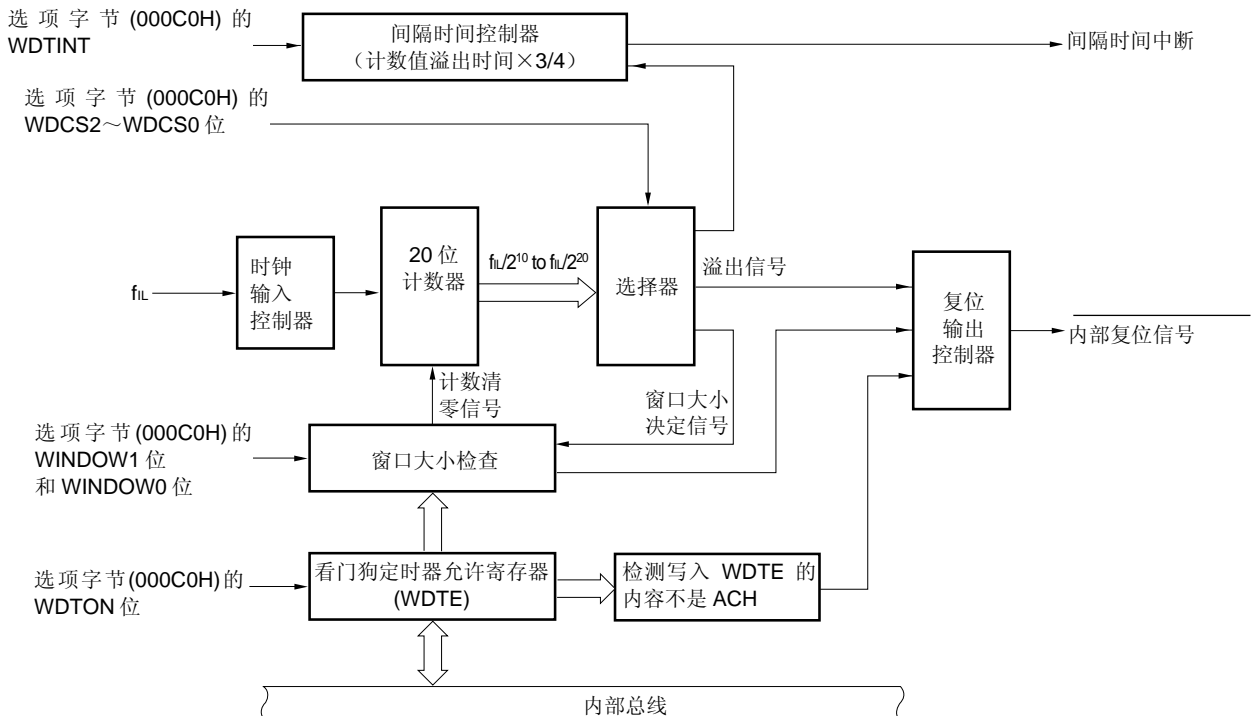
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开比例。

表 8-2. 选项字节和看门狗定时器的设置

看门狗定时器的设置	选项字节 (000C0H)
看门狗定时器间隔中断	第 7 位 (WDTINT)
窗口打开比例	第 6 和 5 位 (WINDOW1, WINDOW0)
控制看门狗定时器的计数器操作	第 4 位 (WDTON)
看门狗定时器的溢出时间	第 3 ~ 1 位 (WDCS2 ~ WDCS0)
控制看门狗定时器的计数操作 (在 HALT/STOP 模式)	第 0 位 (WDSTBYON)

备注 需要了解选项字节的内容, 可参见 第二十三章 选项字节。

图 8-1. 看门狗定时器的框图



8.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器允许寄存器（WDTE）控制。

(1) 看门狗定时器允许寄存器（WDTE）

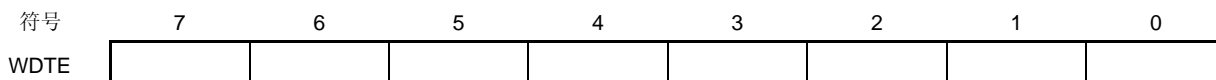
将“ACH”写入 WDTE 可以对看门狗定时器计数器清零，并开始再次计数操作。

可以通过 8 位存储器操作指令设置该寄存器。

复位信号的产生将该寄存器设置为 9AH 或 1AH^注。

图 8-2. 看门狗定时器允许寄存器（WDTE）的格式

地址：FFFABH 复位后：9AH/1AH^注 R/W



注 WDTE 复位后的值会根据选项字节（000C0H） WDTON 位的设置值而变化。如果要对看门狗定时器进行操作，设置 WDTON = 1。

WDTON 设置值	WDTE 复位值
0（禁止看门狗定时器计数的操作）	1AH
1（允许看门狗定时器计数的操作）	9AH

- <R> 注意事项
1. 如果写入 WDTE 的值不是 ACH，将产生一个内部复位信号。
 2. 如果使用 1 位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。
 3. 从 WDTE 读取的值为 9AH/1AH（与写入值（ACH）不同）。

8.4 看门狗定时器的操作

8.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，由选项字节（000C0H）指定其操作。

- 通过设置选项字节（000C0H）的第 4 位（WDTON）=1，允许看门狗定时器的计数操作（复位释放后计数器开始计数）（如需了解详细信息，参见 第二十三章）。

WDTON	看门狗定时器计数器
0	禁止计数器操作（复位后停止计数）
1	允许计数器操作（复位后开始计数）

- 通过使用选项字节（000C0H）的第 3 位～第 1 位（WDCS2 ~ WDCS0）设置溢出时间（如需了解详细信息，参见 8.4.2 和 第二十三章）。
 - 通过使用选项字节（000C0H）的第 6 位和第 5 位（WINDOW1 和 WINDOW0）设置窗口打开比例（如需了解详细信息，参见 8.4.3 和 第二十三章）。
- 复位释放后，看门狗定时器开始计数。
 - 在看门狗定时器开始计数后且在达到选项字节设置的溢出时间之前，将“ACH”写入 WDTE，看门狗定时器将被清零，并开始再次计数。
 - 此后，在复位释放后的窗口打开期间重复写入 WDTE。如果在窗口关闭期间写 WDTE，则会产生内部复位信号。
 - 如果溢出时间已到，却没有“ACH”写入 WDTE，则会产生内部复位信号。
出现以下情况时会产生内部复位信号。
 - 如果对看门狗定时器允许寄存器（WDTE）执行位操作指令时。
 - 如果将“ACH”以外的数据写入 WDTE。

- 注意事项**
- 复位释放后对 WDTE 进行第一次写入操作，将清零看门狗定时器，不论窗口打开时间如何，只要这一操作是在到达溢出时间之前进行的，看门狗定时器都会开始再次计数。
 - 如果将“ACH”写入 WDTE 来对看门狗定时器清零，则实际溢出时间可能与由选项字节设置的溢出时间不同，最高可达 $2/f_{RL}$ 秒。
 - 在计数值即将溢出之前也可以将看门狗定时器清零。

<例如> 当溢出时间设置为 $2^{10}/f_{IL}$ ，在计数值小于 3FH 之前写入“ACH”都是有效操作。

注意事项 4. 看门狗定时器在 HALT 和 STOP 模式下的操作会根据选项字节（000C0H）第 0 位（WDSTBYON）的设置值的不同而有所差异，如下所示。

	WDSTBYON = 0	WDSTBYON = 1
HALT 模式	看门狗定时器操作停止	看门狗定时器操作继续
STOP 模式		

如果 WDSTBYON = 0，则在 HALT 或 STOP 模式释放后看门狗定时器恢复计数。此时，不对计数器清零，而从计数器停止时的值开始计数。

释放 STOP 模式后使用 X1 振荡时钟，经过振荡稳定时间后 CPU 开始操作。

因此，如果 STOP 模式释放至看门狗定时器溢出之间的周期较短，在等待振荡稳定期间发生一个溢出，将会引起复位。

于是，当使用 X1 振荡时钟时，或者当释放 STOP 后看门狗定时器由间隔中断进行清零时，设置溢出时间要考虑振荡稳定时间。

5. 在 flash 存储器自编程和 EEPROMTM 模拟期间看门狗定时器继续运行。在处理期间，中断响应会延迟。设置溢出时间和窗口大小时，应考虑延迟时间。

8.4.2 看门狗定时器溢出时间的设置

通过使用选项字节（000C0H）的第 3 位~第 1 位（WDCS2 ~ WDCS0），设置看门狗定时器的溢出时间。

如果出现溢出，则将产生内部复位信号。在到达溢出时间之前且在窗口打开期间内将“ACH”写入 WDTE，可对当前计数值清零，且看门狗定时器开始再次计数。

溢出时间设置如下所示。

表 8-3. 看门狗定时器溢出时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{15}/f_{IL}$ (124.12 ms)
1	0	1	$2^{17}/f_{IL}$ (496.48 ms)
1	1	0	$2^{18}/f_{IL}$ (992.97 ms)
1	1	1	$2^{20}/f_{IL}$ (3971.88 ms)

注意事项 在 flash 存储器自编程和 EEPROM 模拟期间看门狗定时器继续运行。在处理期间，中断响应会延迟。设置溢出时间和窗口大小时，应考虑延迟时间。

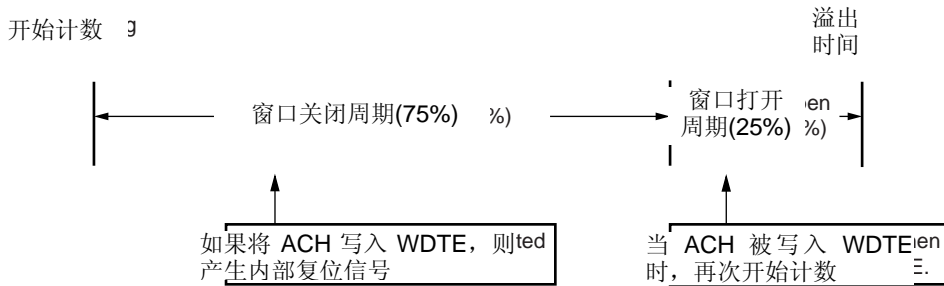
备注 1. f_{IL} : 内部低速振荡时钟频率
2. (): $f_{IL} = 264 \text{ kHz (MAX.)}$

8.4.3 看门狗定时器窗口打开比例的设置

通过使用选项字节（000C0H）的第 6 位和第 5 位（WINDOW1， WINDOW0），设置看门狗定时器的窗口打开比例。窗口使用情况描述如下。

- 如果在窗口打开期间将“ACH”写入 WDTE，则对看门狗定时器清零并再次开始计数。
- 在窗口关闭期间即使将“ACH”写入 WDTE，也会检测到异常情况，并产生内部复位信号。

例：如果窗口打开比例为 25%



注意事项 复位释放后对 WDTE 进行第一次写入操作，将清零看门狗定时器，不论窗口打开时间如何，只要这一操作是在到达溢出时间之前进行的，看门狗定时器都会开始再次计数。

窗口打开比例设置如下所示。

表 8-4. 看门狗定时器窗口打开比例的设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开比例
0	0	25%
0	1	50%
1	0	75%
1	1	100%

- 注意事项**
1. 在 flash 存储器自编程和 EEPROM 模拟期间看门狗定时器继续运行。在处理期间，中断响应会延迟。设置溢出时间和窗口大小时，应考虑延迟时间。
 2. 当选项字节（000C0H）的第 0 位 WDSTBYON）= 0 时，无论 WINDOW1 和 WINDOW0 为何值，窗口打开比例为 100%。
 3. 如果看门狗定时器符合下面任一条件，不要设置窗口打开比例为 25%。
 - 当使用的供电电压（V_{DD}）低于 2.7 V 时。
 - 当通过 STOP 模式或软件停止所有主系统时钟（内部高速振荡时钟，X1 时钟，和外部主系统时钟）时。

<R>

备注 如果将溢出时间设置为 $2^{10}/f_{RL}$ ，则窗口关闭时间和打开时间如下所示。

	窗口打开比例的设置			
	25%	50%	75%	100%
窗口关闭时间	0 ~ 3.56 ms	0 ~ 2.37 ms	0 ~ 0.119 ms	无
窗口打开时间	3.56 ~ 3.88 ms	2.37 ~ 3.88 ms	0.119 ~ 3.88 ms	0 ~ 3.88 ms

<当窗口打开比例为 25%>

- 溢出时间：
 $2^{10}/f_{IL}$ (MAX.) = $2^{10}/264$ kHz (MAX.) = 3.88 ms
- 窗口关闭时间：
 $0 \sim 2^{10}/f_{IL}$ (MIN.) \times (1 - 0.25) = $0 \sim 2^{10}/216$ kHz (MIN.) \times 0.75 = 0 ~ 3.56 ms
- 窗口打开时间：
 $2^{10}/f_{IL}$ (MIN.) \times (1 - 0.25) $\sim 2^{10}/f_{IL}$ (MAX.) =
 $2^{10}/216$ kHz (MIN.) \times 0.75 $\sim 2^{10}/264$ kHz (MAX.) =
3.56 ~ 3.88 ms

8.4.4 设置看门狗定时器间隔中断

根据选项字节 (00C0H) 第 7 位 (WDTINT) 的设置，当达到溢出时间的 75% 时可产生一个间隔中断 (INTWDTI)。

表 8-5. 看门狗定时器间隔中断的设置

WDTINT	看门狗定时器间隔中断的使用
0	使用间隔中断。
1	达到溢出时间的 75% 时产生间隔中断。

注意事项 在 STOP 模式释放后使用 X1 振荡时钟时，经过振荡稳定时间后，CPU 开始工作。因此，如果 STOP 模式释放至看门狗定时器溢出之间的周期较短，在等待振荡稳定期间发生一个溢出，将会引起复位。
于是，当使用 X1 振荡时钟时，或者当释放 STOP 后看门狗定时器由间隔中断进行清零时，设置溢出时间要考虑振荡稳定时间。

备注 在产生 INTWDTI 后看门狗定时器继续计数（直到 ACH 写入到 WDTE 寄存器）。如果在溢出时间前没有“ACH”写入 WDTE 寄存器，产生一个内部复位信号。

第九章 时钟输出/蜂鸣器输出控制器

9.1 时钟输出/蜂鸣器输出控制器的功能

时钟输出控制器用于遥控发送期间的载波输出，以及供给微控制器的外围芯片的输出时钟。

蜂鸣器输出用于蜂鸣器频率的方波输出。

引脚可输出时钟或蜂鸣器声音。

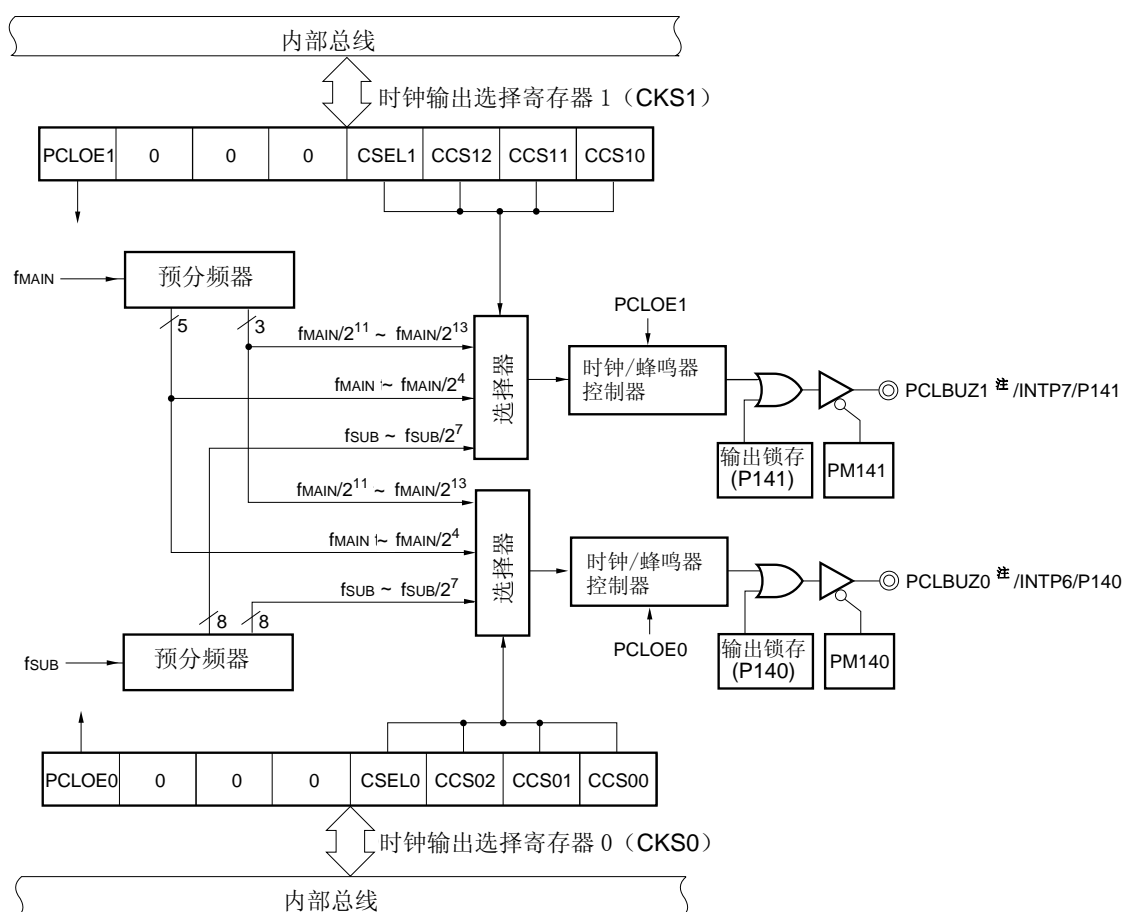
有两个输出引脚可用，PCLBUZ0 和 PCLBUZ1。

PCLBUZ0 输出由时钟输出选择寄存器 0 (CKS0) 选择的时钟。

PCLBUZ1 输出由时钟输出选择寄存器 1 (CKS1) 选择的时钟。

图 9-1 所示为时钟输出/蜂鸣器输出控制器的框图。

图 9-1. 时钟输出/蜂鸣器输出控制器的框图



注 在 $2.7\text{ V} \leq V_{DD}$ 时，PCLBUZ0 和 PCLBUZ1 引脚可输出的时钟最高为 10 MHz。当 $V_{DD} < 2.7\text{ V}$ 时，设置的输出时钟不能超过 5 MHz。

9.2 时钟输出/蜂鸣器输出控制器的配置

时钟输出/蜂鸣器输出控制器包括如下硬件。

表 9-1. 时钟输出/蜂鸣器输出控制器的配置

项目	配置
控制寄存器	时钟输出选择寄存器 0, 1 (CKS0, CKS1) 端口模式寄存器 14 (PM14) 端口寄存器 14 (P14)

9.3 控制时钟输出/蜂鸣器输出控制器的寄存器

如下两个时钟输出/蜂鸣器输出控制器。

- 时钟输出选择寄存器 0, 1 (CKS0, CSK1)
- 端口模式寄存器 14 (PM14)

(1) 时钟输出选择寄存器 0, 1 (CKS0, CKS1)

这些寄存器用于设置时钟输出或蜂鸣器频率输出引脚 (PCLBUZ0/PCLBUZ1) 的输出允许/禁止, 并设置输出时钟。

通过 CKS0 选择要从 PCLBUZ0 输出的时钟。

通过 CKS1 选择要从 PCLBUZ1 输出的时钟。

CKS0 和 CKS1 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器清零 (00H)。

图 9-2. 时钟输出选择寄存器 n (CKSn) 的格式

地址: FFFA5H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn 输出允许/禁止选择
0	输出禁止 (默认)
1	输出允许

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn 输出时钟选择			
				$f_{MAIN} =$ 5 MHz	$f_{MAIN} =$ 10 MHz	$f_{MAIN} =$ 20 MHz	
0	0	0	0	f_{MAIN}	5 MHz	10 MHz [‡]	禁止设置 [‡]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz [‡]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f_{SUB}	32.768 kHz		
1	0	0	1	$f_{SUB}/2$	16.384 kHz		
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz		
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz		
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz		
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz		
1	1	1	0	$f_{SUB}/2^6$	512 Hz		
1	1	1	1	$f_{SUB}/2^7$	256 Hz		

注 在 $2.7\text{ V} \leq V_{DD}$ 时, 输出的时钟最高为 10 MHz。当 $V_{DD} < 2.7\text{ V}$ 时, 设置的输出时钟不能超过 5 MHz。

- 注意事项
1. 禁止时钟输出 (PCLOEn = 0) 后, 改变输出时钟。
 2. 如果在时钟输出 (PCLOEn = 1) 期间选择的时钟 (f_{MAIN} 或 f_{SUB}) 停止, 则输出不确定。

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

(2) 端口模式寄存器 14 (PM14)

该寄存器用于按位选择设置端口 14 的输入/输出操作模式。

当使用 P140/INTP6/PCLBUZ0 和 P141/INTP7/PCLBUZ1 引脚作为时钟输出/蜂鸣器输出时，对 PM140 和 PM141 清零，并且 P140 和 P141 的输出锁存器也为 0。

PM14 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器设置为 FFH。

图 9-3. 端口模式寄存器 14 (PM14) 的格式

地址: FFF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PM14n	P14n 引脚 I/O 模式选择 (n = 0 ~ 5)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

9.4 时钟输出/蜂鸣器输出控制器的操作

一个引脚可用于输出时钟或输出蜂鸣器声音。

有两个输出引脚可用，PCLBUZ0 和 PCLBUZ1。

通过时钟输出选择寄存器 0 (CKS0) 选择要从 PCLBUZ0 输出的时钟/蜂鸣器。

通过时钟输出选择寄存器 1 (CKS1) 选择要从 PCLBUZ1 输出的时钟/蜂鸣器。

9.4.1 作为输出引脚操作

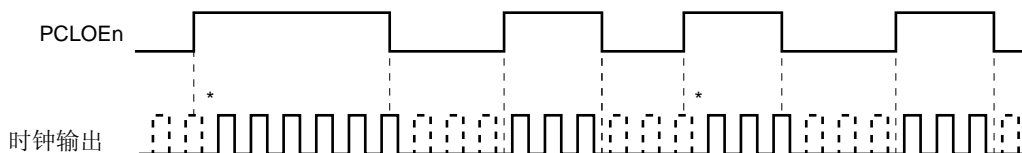
PCLBUZn 的输出按照以下过程进行。

<1> 通过时钟输出选择寄存器 (CKSn) 的第 0~3 位 (CCSn0 ~ CCSn2, CSELn) 选择 PCLBUZn 引脚的输出频率 (输出为禁止状态)。

<2> 设置 CKSn 的第 7 位 (PCLOEn) 为 1，允许时钟/蜂鸣器输出。

备注 当用于输出时钟，以及时钟输出允许或禁止时，该控制器的设计不会输出窄脉冲。如图 10-4 中所示，确保从时钟的低周期 (图中* 所标记) 开始输出。当停止输出时，在时钟高电平周期后执行。

图 9-4. 遥控输出应用示例



备注 n = 0, 1

第十章 A/D 转换器

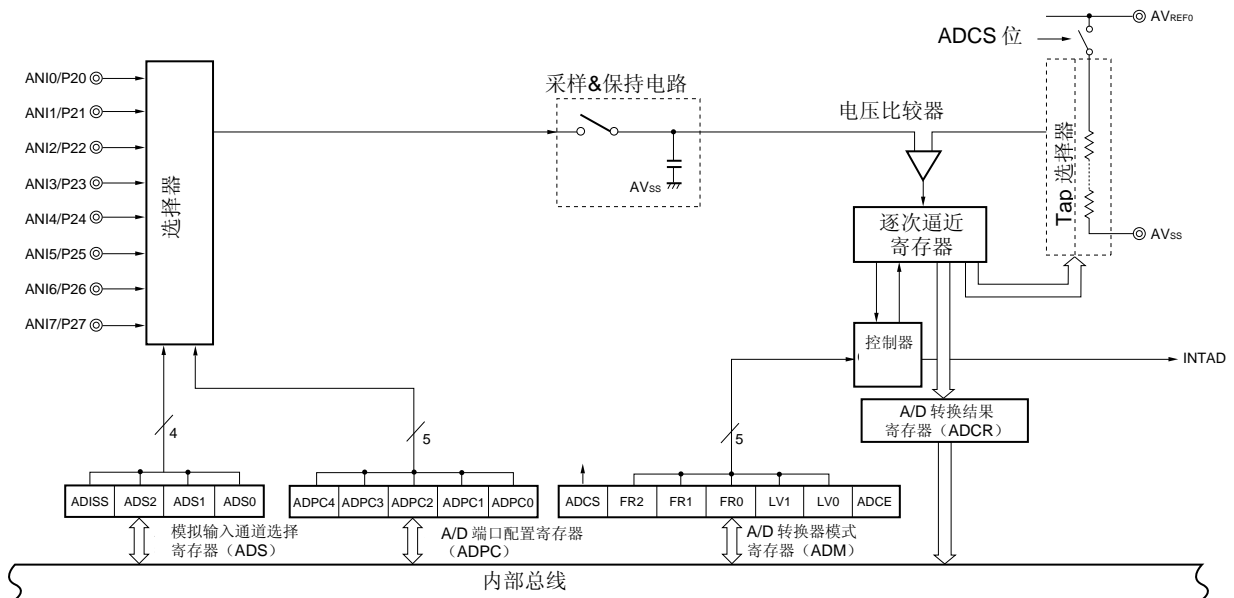
10.1 A/D转换器的功能

A/D 转换器用于将模拟输入信号转换为数字值，最多有 8 个通道（ANI0 ~ ANI7）组成，具有 10 位分辨率。
A/D 转换器有如下功能。

- 10 位分辨率 A/D 转换

从模拟输入 ANI0~ANI7 中选择一个通道，重复执行 10 位分辨率 A/D 转换。每次 A/D 转换结束，都可以产生一个中断请求（INTAD）。

图 10-1. A/D 转换器的框图



10.2 A/D转换器的配置

A/D 转换器包括以下硬件。

(1) AN10~ AN17 引脚

这些是 8 通道 A/D 转换器的模拟输入引脚。用于输入模拟信号，以便转换成数字信号。如果引脚没有选中用作模拟输入，则可以用作 I/O 端口引脚。

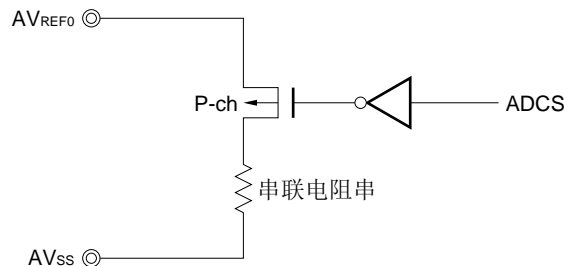
(2) 采样&保持电路

采样&保持电路在 A/D 转换器启动时对选择器选中的模拟输入引脚的输入电压进行采样，并在 A/D 转换期间保持采样到的电压值。

(3) 串联电阻串

串联电阻串用于连接 AV_{REF0} 和 AV_{SS} ，并产生一个电压，与采样到的电压值进行比较。

图 10-2. 串联电阻串的电路配置



(4) 电压比较器

电压比较器用于比较采样到的电压值与串联电阻串的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器用于转换电压比较器所比较的结果，并从最高有效位 (MSB) 开始转换结果。

当电压被转换成数字值，并写入最低有效位 (LSB) 时 (A/D 转换结束)，将 SAR 的内容传送到 A/D 转换结果寄存器 (ADCR) 中。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束时，将 A/D 转换结果从逐次逼近寄存器 (SAR) 加载至该寄存器中。并由该寄存器将转换结果保存在它的高 10 位中 (低 6 位恒为 0)。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束时，将 A/D 转换结果从逐次逼近寄存器加载至该寄存器中。并将转换结果的高 8 位保存在 ADCRH 中。

(8) 控制器

该电路用于控制模拟输入信号被转换成数字信号的转换时间，并启动和停止转换操作。当 A/D 转换结束时，该控制器产生中断 INTAD。

(9) AVREF0 引脚

该引脚为 A/D 转换器输入一个模拟供电/参考电压。当端口 2 的所有引脚被用作模拟端口引脚时，要使 AVREF0 电压在 $2.3\text{ V} \leq \text{AVREF0} \leq \text{VDD}$ 。当端口 2 的一个或多个引脚用作数字端口引脚，或不使用 A/D 转换器时，保证 AVREF0 电压与 EVDD 或 VDD 一致。

基于 AVREF0 和 AVSS 之间的电压，将输入到 ANI0 ~ ANI7 的信号转换成数字信号。

(10) AVSS 引脚

这是 A/D 转换器的信号地引脚。不使用 A/D 转换器时，此引脚的电压与 EVSS 和 VSS 引脚保持一致。

(11) A/D 转换器模式寄存器 (ADM)

该寄存器用于设置模拟输入信号的转换时间，并启动或停止转换操作。

(12) A/D 端口配置寄存器 (ADPC)

该寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换为 A/D 转换器的模拟输入，或端口的数字 I/O 输入。

(13) 模拟输入通道指定寄存器 (ADS)

该寄存器用来选择输入模拟电压（将被转换成数字信号）的端口。

(14) 端口模式寄存器 2 (PM2)

此寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换为输入或输出。

10.3 A/D转换器使用的寄存器

A/D 转换器 使用下面七个寄存器。

- 外围设备允许寄存器 0 (PER0)
- A/D 转换器模式寄存器 (ADM)
- A/D 端口配置寄存器 (ADPC)
- 模拟输入通道指定寄存器 (ADS)
- 端口模式寄存器 2 (PM2)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)

(1) 外围设备允许寄存器 0 (PER0)

PER0 用于允许或禁止每个外围硬件模块操作。为了降低功耗和噪声，对于不使用的硬件模块停止提供时钟。

使用 A/D 转换器时，确保设置此寄存器的第 5 位 (ADCEN) 为 1。

PER0 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将该寄存器清零 (00H)。

- 注意事项 1.** 设置 A/D 转换器时，首先确保设置 ADCEN 为 1。如果 ADCEN = 0，写入到 A/D 转换器的控制寄存器被忽略，即使读取寄存器，只能读到默认值。
- 2.** 确保将 PER0 的第 1 位清 0。

图 10-3. 外围设备允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	A/D 转换器 输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 不能写入被 A/D 转换器使用的 SFR。 • A/D 转换器在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 可读/写被 A/D 转换器使用的 SFR。

(2) A/D 转换器模式寄存器 (ADM)

该寄存器设置模拟输入的 A/D 转换时间，并启动/停止转换。

ADM 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将该寄存器清零 (00H)。

图 10-4. A/D 转换器模式寄存器 (ADM) 的格式

地址: FFF30H 复位后: 00H R/W

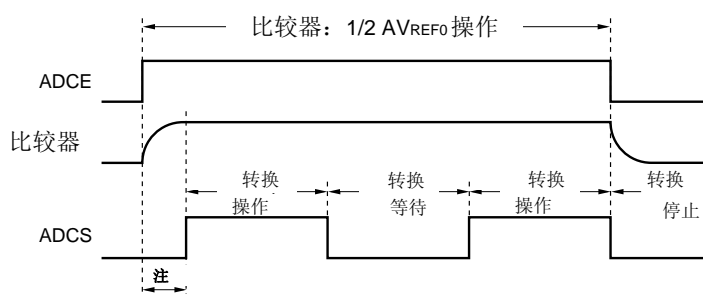
符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2 ^{注1}	FR1 ^{注1}	FR0 ^{注1}	LV1 ^{注1}	LV0 ^{注1}	ADCE
ADCS	A/D 转换操作控制							
0	停止转换操作。							
1	允许转换操作							
ADCE	比较器操作控制 ^{注2}							
0	停止比较器的操作							
1	允许比较器的操作 (比较器: $1/2AV_{REF0}$)							

- 注
1. FR2 ~FR0, LV1, LV0, 和 A/D 转换的详细信息, 参见表 10-2 A/D 转换时间选择。
 2. 比较器的操作是由 ADCS 和 ADCE 控制的, 从操作开始到操作稳定需要 $1\mu\text{s}$ 。因此, 当 ADCE 被设置为 1 后, 至少经过 $1\mu\text{s}$ 的时间, 再将 ADCS 置 1 时, 此时的转换结果优于第一次的转换结果。否则, 忽略第一次转换的数据。

表 10-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流功耗)
0	1	转换等待模式 (比较器: $1/2AV_{REF0}$ 操作, 只有比较器产生功耗)
1	0	禁止设置
1	1	转换模式 (比较器: $1/2AV_{REF0}$ 操作)

图 10-5. 使用比较器时的时序图



注 从 ADCE 位上升到 ADCS 位下降所需时间必须至少为 $1\mu\text{s}$, 以便稳定内部电路。

注意事项 在写入不同数据修改 FR0 ~ FR2 位、LV1 和 LV0 位之前, A/D 转换必须停止。

表 10-2. A/D 转换时间选择

(1) $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$

A/D 转换器模式寄存器 (ADM)					转换时间选择			转换时钟 (f _{AD})	
FR2	FR1	FR0	LV1	LV0	f _{CLK} = 2 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz		
0	0	0	0	0	264/f _{CLK}	禁止设置	26.4 μs	13.2 μs	f _{CLK} /12
0	0	1	0	0	176/f _{CLK}		17.6 μs	8.8 μs [‡]	f _{CLK} /8
0	1	0	0	0	132/f _{CLK}		13.2 μs	6.6 μs [‡]	f _{CLK} /6
0	1	1	0	0	88/f _{CLK}		8.8 μs [‡]	禁止设置	f _{CLK} /4
1	0	0	0	0	66/f _{CLK}	33.0 μs	6.6 μs [‡]		f _{CLK} /3
1	0	1	0	0	44/f _{CLK}	22.0 μs	禁止设置		f _{CLK} /2
1	1	1	0	0	22/f _{CLK}	11.0 μs [‡]		f _{CLK}	
其他值					禁止设置				

注 只有当 $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ 时可设置。

注意事项 根据以下条件设置转换时间。

- $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$: f_{AD} = 0.6~3.6 MHz
- $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$: f_{AD} = 0.6~1.8 MHz

(2) $2.3\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$

A/D 转换器模式寄存器 (ADM)					转换时间选择		转换时钟 (f _{AD})	
FR2	FR1	FR0	LV1	LV0	f _{CLK} = 2 MHz	f _{CLK} = 5 MHz		
0	0	0	0	1	480/f _{CLK}	禁止设置	禁止设置	f _{CLK} /12
0	0	1	0	1	320/f _{CLK}		64.0 μs ^{‡1}	f _{CLK} /8
0	1	0	0	1	240/f _{CLK}		48.0 μs ^{‡1}	f _{CLK} /6
0	1	1	0	1	160/f _{CLK}		32.0 μs	f _{CLK} /4
1	0	0	0	1	120/f _{CLK}	60.0 μs	24.0 μs ^{‡2}	f _{CLK} /3
1	0	1	0	1	80/f _{CLK}	40.0 μs	16.0 μs ^{‡3}	f _{CLK} /2
1	1	1	0	1	40/f _{CLK}	20.0 μs ^{‡3}	禁止设置	f _{CLK}
其他值					禁止设置			

注 1. 只有当 $2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$ 时可以设置。

2. 只有当 $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ 时可以设置。

3. 只有当 $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ 时可以设置。

注意事项 1. 根据以下条件设置转换时间。

- $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$: f_{AD} = 1.2 ~ 3.6 MHz
- $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$: f_{AD} = 1.2 ~ 1.8 MHz
- $2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$: f_{AD} = 0.6 ~ 1.44 MHz

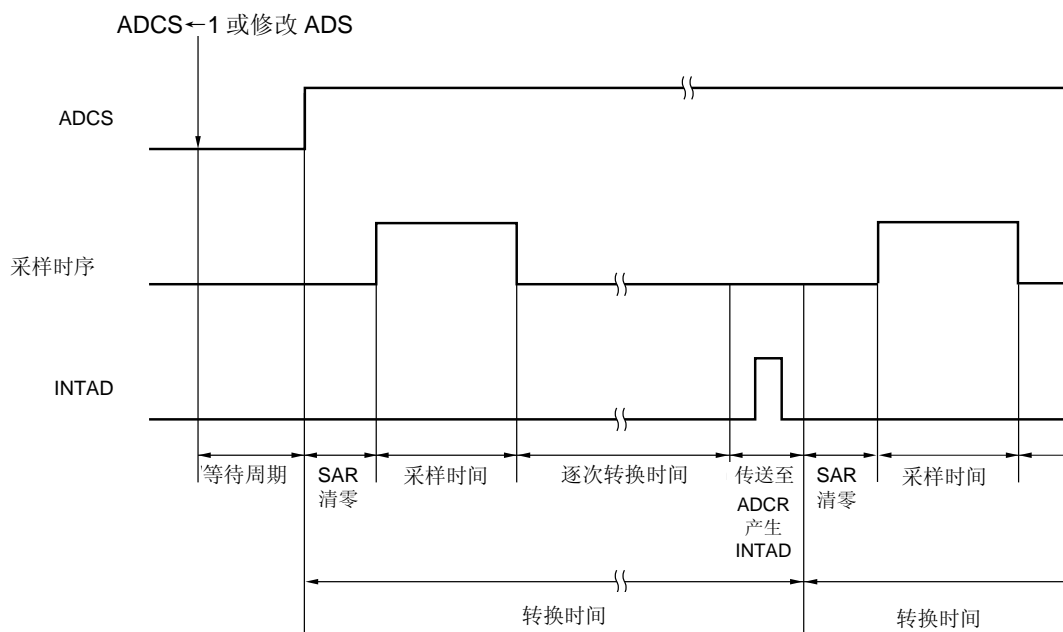
2. 在写入不同的数据修改 FR0 ~ FR2 位、LV1 和 LV0 位之前, A/D 转换必须停止 (ADCS = 0)。

3. 当 $2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$ 时, 修改 LV1 和 LV0 的默认值。

4. 上述的转换时间不包括时钟频率误差。在选择转换时间的时候, 要将时钟频率误差考虑进去。

备注 f_{CLK}: CPU/外围硬件时钟频率

图 10-6. A/D 转换器采样和 A/D 转换时序



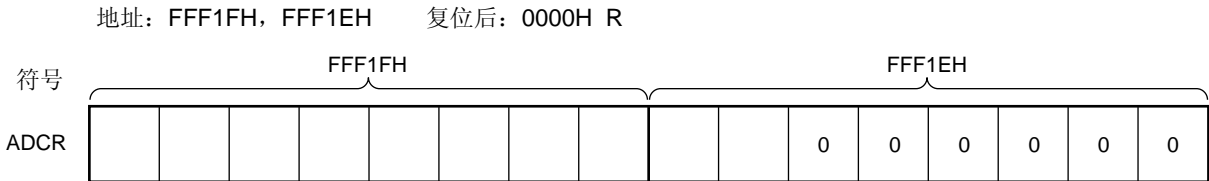
(3) 10 位 A/D 转换结果寄存器 (ADCR)

这是一个 16 位寄存器，用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束时，将转换结果从逐次逼近寄存器传送至 ADCR 中。转换结果的高 8 位存储在 FFF1FH 中，而转换结果的低 2 位存储在 FFF1EH 的高 2 位中。

ADCR 可由 16 位存储器操作指令读取。

复位信号产生将该寄存器清零 (0000H)。

图 10-7. 10 位 A/D 转换结果寄存器 (ADCR) 的格式



注意事项 在对 A/D 转换器模式寄存器 (ADM) 和模拟输入通道选择寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 赋值时，ADCR 的内容可能是不确定的。转换结束后先读取转换结果，再对 ADM, ADS 和 ADPC 赋值。如果没有使用上述时序，可能会读取到不正确的转换结果。

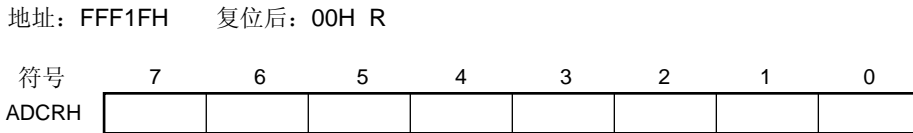
(4) 8 位 A/D 转换结果寄存器 (ADCRH)

这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。

ADCRH 可由 8 位存储器操作指令读取。

复位信号产生将该寄存器清零 (00H)。

图 10-8. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式



注意事项 在对 A/D 转换器模式寄存器 (ADM) 和模拟输入通道选择寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 赋值时，ADCR 的内容可能是不确定的。转换结束后先读取转换结果，再对 ADM, ADS 和 ADPC 赋值。如果没有使用上述时序，可能会读取到不正确的转换结果。

(5) 模拟输入通道指定寄存器 (ADS)

该寄存器用来选择进行 A/D 转换模拟电压的输入通道。

ADS 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将该寄存器清零 (00H)。

图 10-9. 模拟输入通道指定寄存器 (ADS) 的格式

地址: FFF31H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	0	0	ADS2	ADS1	ADS0

ADISS	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	ANI0	P20/ANI0 引脚
0	0	0	1	ANI1	P21/ANI1 引脚
×	0	1	0	ANI2	P22/ANI2 引脚
×	0	1	1	ANI3	P23/ANI3 引脚
×	1	0	0	ANI4	P24/ANI4 引脚
×	1	0	1	ANI5	P25/ANI5 引脚
×	1	1	0	ANI6	P26/ANI6 引脚
×	1	1	1	ANI7	P27/ANI7 引脚

注意事项 1. 第 3~6 位必须清零。

2 使用端口模式寄存器 2 (PM2) 将用于 A/D 转换通道的端口设置为输入模式。

3. 不要通过 ADPC 和 ADS 设置数字 I/O 引脚。

备注 ×: 不必考虑

(6) A/D 端口配置寄存器 (ADPC)

这个寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换为 A/D 转换器的模拟输入或是数字 I/O 端口。

ADPC 可由 8 位存储器操作指令设置。

复位信号产生将该寄存器清为 10H。

图 10-10. A/D 端口配置寄存器 (ADPC) 的格式

地址: F0017H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADP C4	ADP C3	ADP C2	ADP C1	ADP C0	模拟输入 (A) / 数字 I/O (D) 切换							
					ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D
其他值					禁止设置							

- 注意事项
1. 用端口模式寄存器 2 (PM2) 设置用于 A/D 转换通道的端口为输入模式。
 2. 不要通过 ADPC 以及 ADS 设置数字 I/O。
 3. 当 ANI0/P20 ~ ANI7/P27 都用作数字 I/O (D)，ADPC4~ADPC0 可以设置为 01000 或 10000 均可。

(7) 端口模式寄存器 2 (PM2)

当使用 ANI0/P20~ANI7/P27 引脚用于模拟输入端口时，设置 PM20~PM27 为 1。P20~P27 的输出锁存此时可能为 0 或 1。

如果 PM20~PM27 设置为 0，则他们不能用作模拟输入端口引脚。

PM2 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将这些寄存器设置为 FFH。

注意事项 如果引脚设置为模拟输入端口，引脚只能读到“0”，而读不到引脚电平。

图 10-11. 端口模式寄存器 2 (PM2) 的格式

地址: FFF22H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	P2n 引脚 I/O 模式选择 (n = 0~7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器打关)

根据 ADPC, ADS 和 PM2 的设置, ANI0/P20~ANI7/P27 引脚功能如下所示。

表 10-3. ANI0/P20~ANI7/P27 引脚的设置功能

ADPC	PM2	ADS	ANI0/P20~ANI7/P27 引脚
数字 I/O 选择	输入模式	-	数字输入
	输出模式	-	数字输出
模拟输入选择	输入模式	选择 ANI.	模拟输入 (被转换)
		不要选择 ANI.	模拟输入 (不被转换)
	输出模式	选择 ANI.	禁止设置
		不要选择 ANI.	

10.4 A/D转换器操作

10.4.1 A/D转换器的基本操作

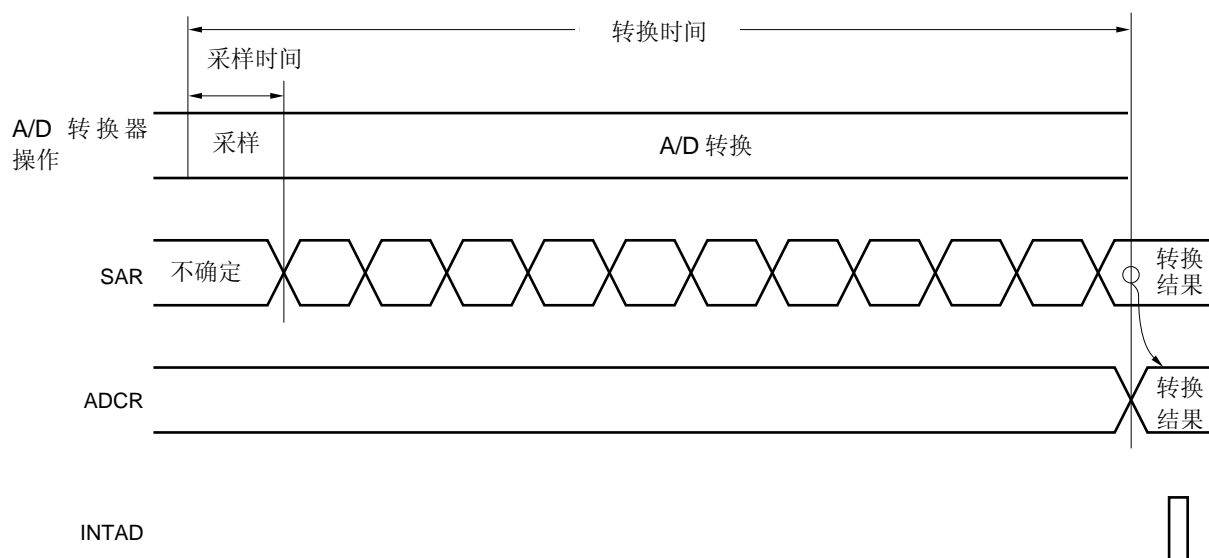
- <1> 把外围设备允许寄存器 0 (PER0) 的第 5 位 (ADCEN) 置 1, 启动 A/D 转换器输入时钟供给。
- <2> 把 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 置 1, 启动比较器的操作。
- <3> 通过 A/D 端口配置寄存器 (ADPC) 将模拟输入的通道设置为 A/D 转换模式, 并且通过端口模式寄存器 2 (PM2) 设置为输入模式。
- <4> 通过 ADM 的第 5~1 位 (FR2~FR0, LV1, 和 LV0) 设置 A/D 转换时间。
- <5> 使用模拟输入通道指定寄存器 (ADS) 为 A/D 转换选择一个通道。
- <6> 通过设置 ADM 的第 7 位 (ADCS) 为 1 开始转换操作。
(<7>~<13> 操作由硬件执行。)
- <7> 输入到所选择的模拟输入通道的电压由采样&保持电路采样。
- <8> 在经过一段时间的采样后, 采样&保持电路处于保持状态, 且在 A/D 转换操作结束前一直保持采样电压。
- <9> 设置逐次逼近寄存器 (SAR) 的第 9 位。通过分接选择器将串联电阻串的分接电压置为 $(1/2) AV_{REF0}$ 。
- <10> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入电压高于 $(1/2) AV_{REF}$, 则 SAR 的 MSB = 1; 如果模拟输入电压低于 $(1/2) AV_{REF}$, 则 SAR 的 MSB = 0。
- <11> 接下来, SAR 的第 8 位自动置 1, 并进入下一个比较过程。根据第 9 位的预置值选择串联电阻串的分接电压, 具体描述如下
 - 第 9 位 = 1: $(3/4) AV_{REF0}$
 - 第 9 位 = 0: $(1/4) AV_{REF0}$
 比较分接电压与采样电压, 并设置 SAR 的第 8 位, 如下所示。
 - 模拟输入电压 \geq 分接电压: 第 8 位 = 1
 - 模拟输入电压 $<$ 分接电压: 第 8 位 = 0
- <12> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <13> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器 (ADCR, ADCRH) 中, 并锁存。
同时也会产生 A/D 转换结束中断请求 (INTAD)。
- <14> 重复步骤<7>~<13>, 直到 ADCS 被清为 0。
将 ADCS 清零, 以停止 A/D 转换器操作。
当 ADCE = 1 时, 若要重新启动 A/D 转换操作, 应从步骤<6>开始。当 ADCE = 0 时, 若要再次启动 A/D 转换操作, 设置 ADCE=1, 等待至少 $1 \mu s$, 然后从步骤<6>开始操作。如要改变 A/D 转换的通道, 则从步骤<5>开始。

注意事项 必须确保 <2> ~ <6> 的操作时间至少为 $1 \mu s$ 。

备注 有两种类型的 A/D 转换结果寄存器可以使用。

- ADCR (16 位): 存储 10 位 A/D 转换值
- ADCRH (8 位): 存储 8 位 A/D 转换值

图 10-12. A/D 转换器的基本操作



连续执行 A/D 转换操作，直到用软件将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 复位 (0)。

在 A/D 转换期间，如果对模拟输入通道选择寄存器 (ADS) 进行写入操作，则转换操作被初始化，并且若 ADCS 被设置为 1，则转换操作重新开始。

复位信号的产生会将 A/D 转换结果寄存器 (ADCR, ADCRH) 的内容设置为 0000H 或 00H。

10.4.2 输入电压和转换结果

输入到模拟输入引脚（AN10~AN17）的模拟输入电压与理论上的 A/D 转换结果（存储在 10 位 A/D 转换结果寄存器（ADCR）中）之间的关系表示如下。

$$SAR = INT \left(\frac{V_{AIN}}{AV_{REF0}} \times 1024 + 0.5 \right)$$

$$ADCR = SAR \times 64$$

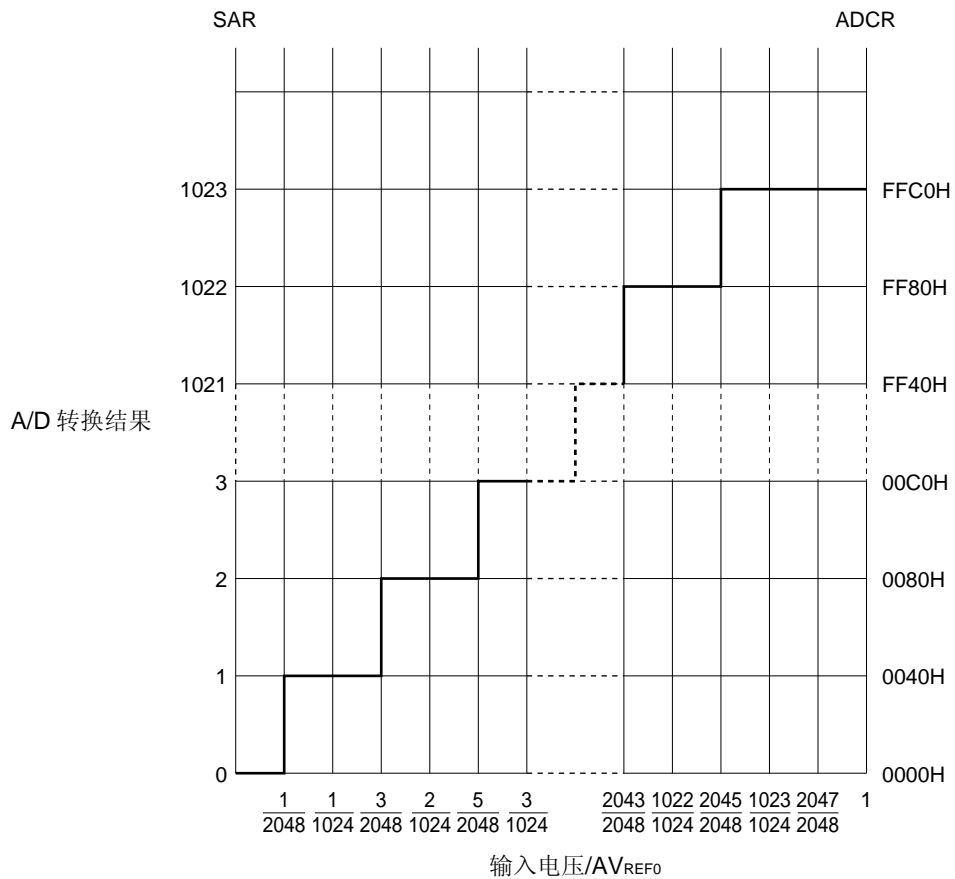
或

$$\left(\frac{ADCR}{64} - 0.5 \right) \times \frac{AV_{REF0}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5 \right) \times \frac{AV_{REF0}}{1024}$$

- 其中，INT（）： 该函数返回括号中值的整数部分。
 V_{AIN}： 模拟输入电压
 AV_{REF0}： AV_{REF0} 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

图 11-13 显示模拟输入电压与 A/D 转换结果之间的关系。

图 10-13. 模拟输入电压与 A/D 转换结果之间的关系



10.4.3 A/D转换器 操作模式

A/D 转换器的操作模式为选择模式。由模拟输入通道选择寄存器 (ADS) 从 ANI0~ANI7 中选择一个模拟输入通道, 并执行 A/D 转换。

(1) A/D 转换操作

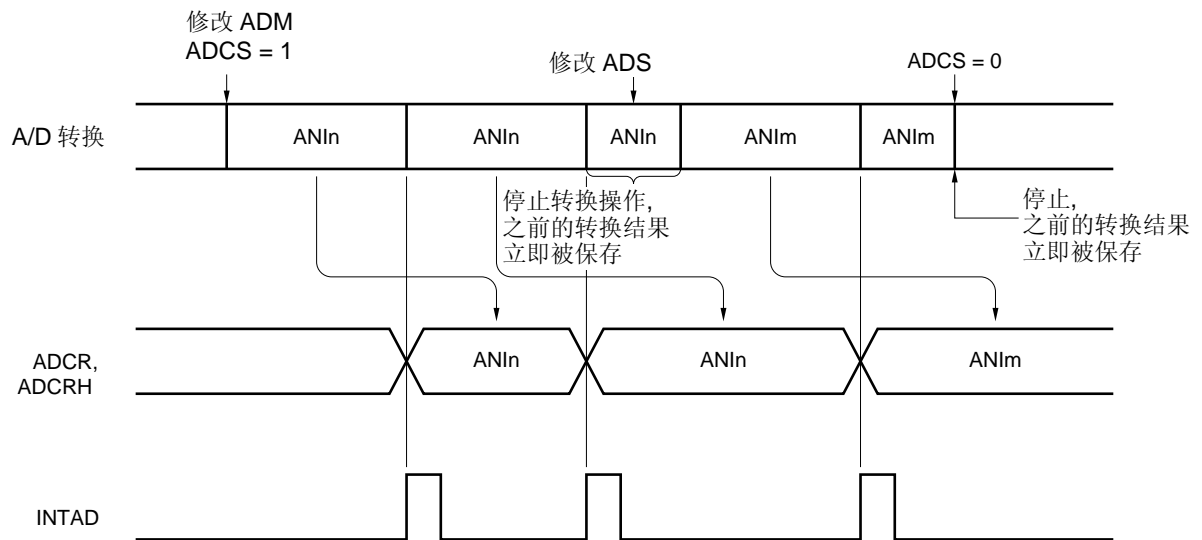
通过将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 置 1, 可以启动对电压的 A/D 转换操作, 该电压由模拟输入通道选择寄存器 (ADS) 选择的模拟输入引脚输入。

当 A/D 转换完成时, 将 A/D 转换结果存于 A/D 转换结果寄存器 (ADCR) 中, 并产生一个中断请求信号 (INTAD)。当一个 A/D 转换结束时, 下一个 A/D 转换操作将立即开始。

如果在 A/D 转换期间 ADS 的值被重写, 则停止当前正在执行的 A/D 转换操作, 重新开始。

如果在 A/D 转换期间 ADCS 被置 0, 则 A/D 转换立即停止。此时, 立即保存之前的转换结果。

图 10-14. A/D 转换操作



- 备注
1. $n = 0 \sim 7$
 2. $m = 0 \sim 7$

设置方法如下。

- <1> 设置外围设备允许寄存器 0 (PER0) 的第 5 位 (ADCEN) 为 1。
- <2> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1。
- <3> 通过 A/D 端口配置寄存器 (ADPC) 的第 4~0 位 (ADPC4~ADPC0) 和端口模式寄存器 2 (PM2) 的第 7~0 位 (PM27~PM20) 设置用于模拟输入模式的通道。
- <4> 通过 ADM 的第 5~1 位 (FR2~FR0, LV1, 和 LV0) 选择转换时间。
- <5> 使用模拟输入通道选择寄存器 (ADS) 的第 7 和 2~0 位 (ADISS, ADS2~ADS0) 选择通道。
- <6> 将 ADM 的第 7 位 (ADCS) 置 1, 启动 A/D 转换。
- <7> 当 A/D 转换结束, 产生一个中断请求信号 (INTAD)。
- <8> 将 A/D 转换数据传送至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。

<改变通道>

- <9> 根据 ADS 的第 7 和 2~0 位 (ADISS, ADS2~ADS0) 改变通道来启动 A/D 转换。
- <10> 当 A/D 转换结束, 产生一个中断请求信号 (INTAD)。
- <11> 将 A/D 转换数据传送至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。

<完成 A/D 转换>

- <12> ADCS 清为 0。
- <13> ADCE 清为 0。
- <14> 外围设备允许寄存器 0 (PER0) 的第 5 位 (ADCEN) 清为 0。

- 注意事项**
1. 必须确保步骤 <2> ~ <6> 之间的操作时间至少为 1 μ s。
 2. <2> 可以在 <3> 和 <5> 之间进行。
 3. <2> 可以被省略。但在这种情况中 (<6> 之后) 忽略第一次转换结果。
 4. 步骤 <7> ~ <10> 所经历的时间不同于使用 ADM 的第 5~1 位 (FR2 ~ FR0, LV1, LV0) 设置的转换时间。步骤 <9> ~ <10> 所经历的时间为 FR2 ~ FR0、LV1, LV0 设置的转换时间。

10.5 A/D转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1LSB（最低有效位）。对于满度的 1LSB 的百分比用 %FSR（满度范围）表示。

当分辨率为 10 位时 1LSB 表示如下。

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\% \text{FSR} \end{aligned}$$

精确度与分辨率无关，而由总误差决定。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零标度误差、满量程误差、积分线性误差和微分线性误差等组合起来表示总误差。

注意量化误差不属于特征表中总误差的范围。

(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。

注意量化误差不属于特征表中总误差、零标度误差、满量程误差、积分线性误差和微分线性误差的范围。

图 10-15. 总误差

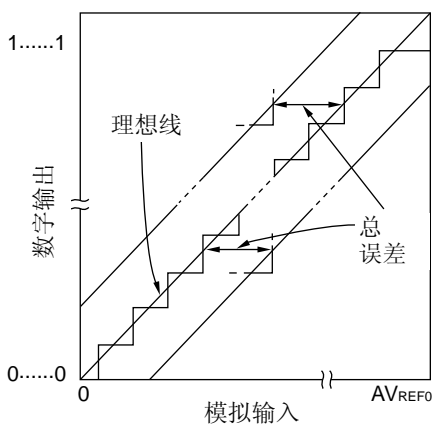
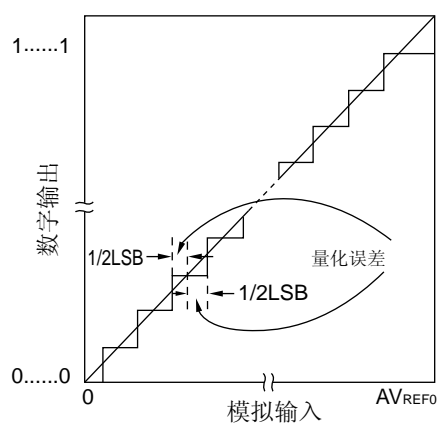


图 10-16. 量化误差



(4) 零标度误差

零标度误差表示当数字输出范围在 0.....000 ~ 0.....001 之间时模拟输入电压的实际测量值与理论值 ($1/2\text{LSB}$) 之间的误差。

如果实际测量值大于理论值，零标度误差表示当数字输出范围在 0.....001 ~ 0.....010 之间时模拟输入电压的实际测量值与理论值 ($3/2\text{LSB}$) 之间的误差。

(5) 满量程误差

满量程误差表示当数字输出范围在 $1\dots\dots110 \sim 1\dots\dots111$ 之间时模拟输入电压的实际测量值与理论值（满量程 - $3/2\text{LSB}$ ）之间的误差。

(6) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零标度误差和满量程误差均为 0 时实际测量值与理想直线之间误差的最大值。

(7) 微分线性误差

当代码输出的理想宽度为 1LSB 时，微分线性误差表示实际测量值与理想值之间的差距。

图 10-17. 零标度误差

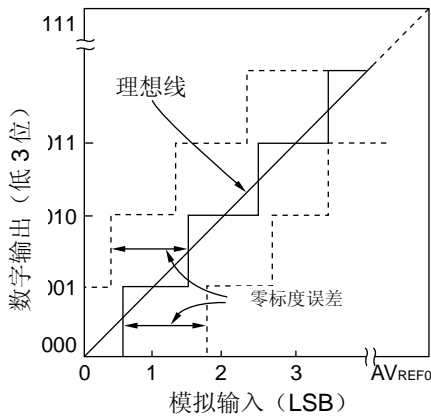


图 10-18. 满量程误差

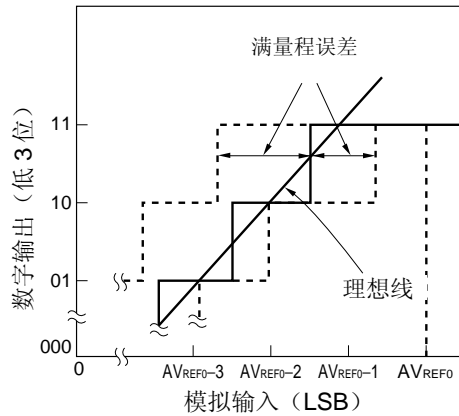


图 10-19. 积分线性误差

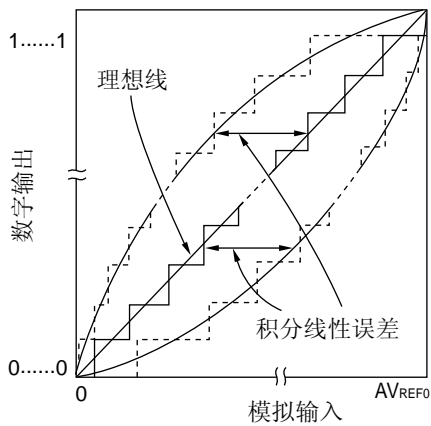
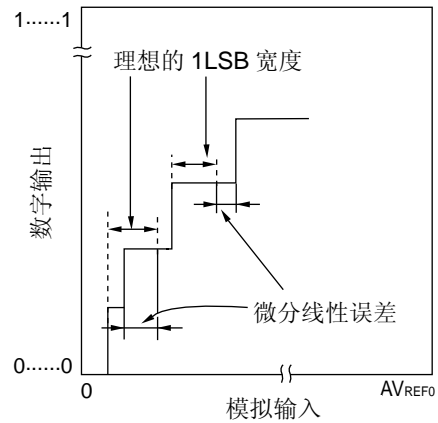


图 10-20. 微分线性误差

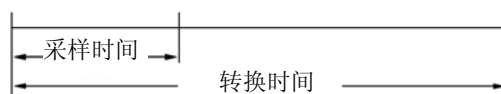


(8) 转换时间

转换时间表示从开始采样到获得数字输出所经历的时间。
采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示打开模拟电压的模拟开关到模拟电压被采样&保持电路采样所需的时间。



10.6 A/D转换器的注意事项

(1) STOP 模式中的操作电流

在 STOP 模式中 A/D 转换器停止操作。此时，将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零，可以降低操作电流。

若要从待机状态重新启动，将中断请求标志寄存器 1L (IF1L) 的第 0 位 (ADIF) 清零 (0)，然后开始操作。

(2) AN10~AN17 的输入范围

观察 AN10~AN17 输入电压的额定范围。如果输入到模拟输入通道的电压大于等于 AV_{REF} ，或者小于等于 AV_{SS} (即使在绝对最大额定范围之内)，则该通道的转换值不确定。此外，其它通道的转换值也可能会受影响。

(3) 冲突操作

<1> 转换结束后，通过指令对 A/D 转换结果寄存器 (ADCR, ADCRH) 的写操作和 ADCR 或 ADCRH 的读操作之间的冲突。

ADCR 或 ADCRH 读操作的优先级高。在执行读操作后，才将新的转换结果写入 ADCR 或 ADCRH。

<2> 转换结束后，ADCR 或 ADCRH 的写操作、A/D 转换器模式寄存器 (ADM) 的写操作以及模拟输入通道选择寄存器 (ADS) 或 A/D 端口配置寄存器 (ADPC) 的写操作之间的冲突。

ADM、ADS 或 ADPC 的写操作的优先级高。不执行 ADCR 或 ADCRH 的写操作，也不会产生转换结束中断信号 (INTAD)。

(4) 解决噪音问题的方法

为了保持 10 位分辨率，必须注意输入到 AV_{REF0} 引脚和 AN10~AN17 引脚的噪音。

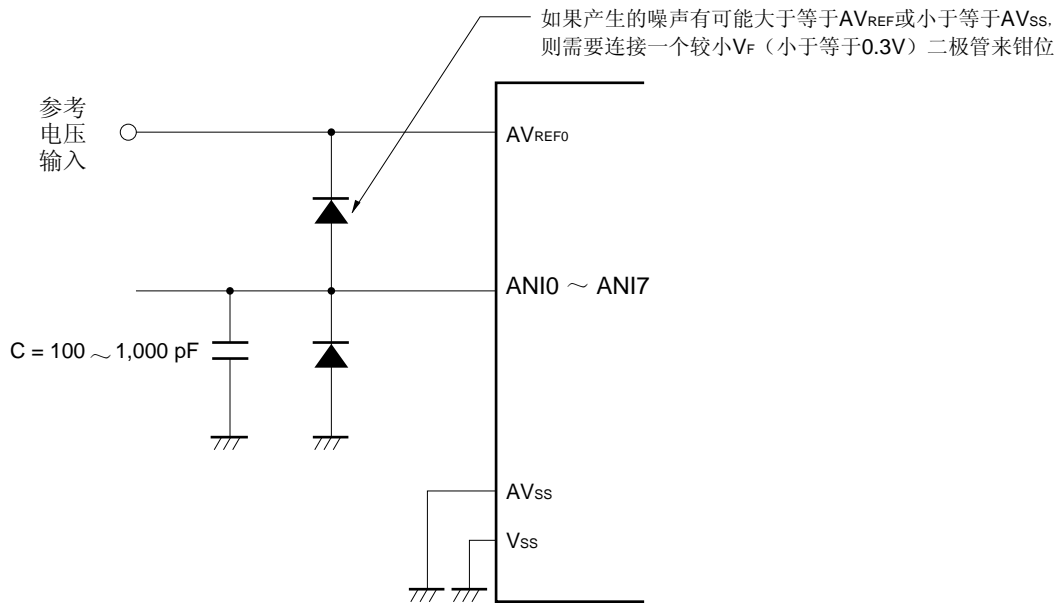
<1> 连接一个电容到电源上，该电容的等效电阻要低，且频率响应特征要好。

<2> 模拟输入源的输出阻抗越大，干扰就越大。为了降低噪音，建议按图 10-21 所示连接外部 C。

<3> 在转换过程中不要切换引脚。

<4> 如果在转换开始后立即设置 HALT 模式，则可以改善精度。

图 10-21. 模拟输入引脚连接

**(5) ANI0/P20~ANI7/P27**

<1> 模拟输入引脚 (ANI0~ANI7) 也可用作输入端口引脚 (P20~P27)。

当选择 ANI0 ~ ANI7 中的任意一个通道执行 A/D 转换时，转换过程中不要访问 P20~P27；否则转换分辨率可能会降低。建议从 ANI0/P20 (离 AVREF0 最远) 开始，选择引脚用作 P20~P27 数字 I/O 端口模式。

<2> 如果正在进行 A/D 转换的引脚的相邻引脚上有数字脉冲，则由于噪音耦合，可能得不到预期的 A/D 转换值。因此在进行 A/D 转换时不要在相邻引脚引入脉冲。

(6) ANI0~ANI7 引脚的输入阻抗

采样期间 A/D 转换器对采样电容充电，以便进行采样。

因此当不进行采样时仅有漏电流经过，而在采样期间则有为电容充电的电流，因此根据是否进行采样，输入阻抗会波动，该波动还有其他影响因素。

为了使采样有效，建议模拟输入源的输出阻抗小于等于 10 kΩ，并将一个 100 pF 左右的电容连接到 ANI0 ~ ANI7 引脚上 (参见图 10-21)。

(7) AVREF0 引脚输入阻抗

在 AVREF0 与 AVSS 引脚之间连接几十千欧的串联电阻串。

因此，如果参考电压源的输出阻抗很高，相当于一系列电阻连接到 AVREF0 与 AVSS 引脚之间的串联电阻串，会导致较大的参考电压误差。

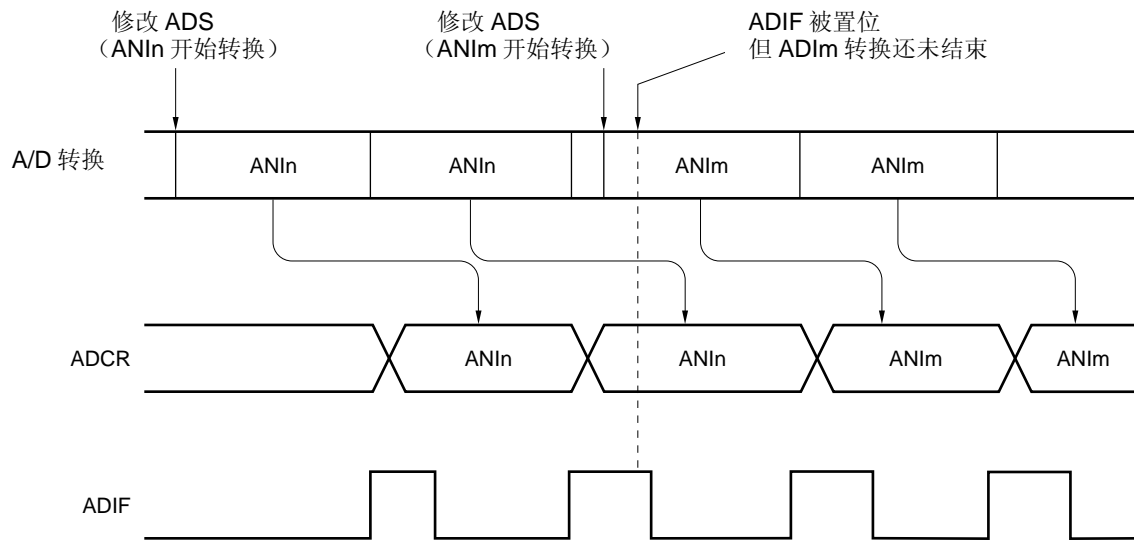
(8) 中断请求标志 (ADIF)

即使模拟输入通道选择寄存器 (ADS) 的值被修改, 中断请求标志 (ADIF) 也不会被清零。

因此, 如果在 A/D 转换期间有模拟输入引脚发生变化, 模拟输入通道的 A/D 转换结果和改变前的模拟通道的 ADIF 都可能在重写 ADS 之前被设置。此时需要注意的是, 当 ADS 修改后立即读取 ADIF 时, 即使修改后的模拟输入的 A/D 转换尚未结束, 也会设置 ADIF。

当 A/D 转换停止后又重新开始时, 在启动前先对 ADIF 清零。

图 10-22. A/D 转换结束中断请求产生的时序



- 备注
1. $n = 0 \sim 7$
 2. $m = 0 \sim 7$

(9) A/D 转换刚开始时的转换结果

在 A/D 转换开始后, 若在 ADCE 置 1 后的 $1 \mu\text{s}$ 内对 ADCS 置 1, 或者 ADCE=0 时 ADCS 置 1, 那么第一次的 A/D 转换值可能不在额定范围内。可采取措施, 如轮询 A/D 转换结束中断请求 (INTAD), 并删除第 1 次转换结果。

(10) A/D 转换结果寄存器 (ADCR, ADCRH) 读操作

当对 A/D 转换器模式寄存器 (ADM)、模拟输入通道选择寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 进行写入操作时, ADCR 和 ADCRH 的内容可能不确定。转换结束后要读取转换结果, 应该在 ADM、ADS 和 ADPC 写入操作前进行。如果采用与上述不同的时序操作, 可能会读取到不正确的转换结果。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 10-23. ANIn 引脚的内部等效电路

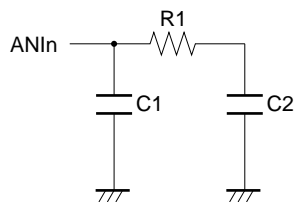


表 10-4. 等效电路的电阻和电容（参考值）

AV_{REF0}	R1	C1	C2
$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	8.1 k Ω	8 pF	5 pF
$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	31 k Ω	8 pF	5 pF
$2.3\text{ V} \leq V_{DD} < 2.7\text{ V}$	381 k Ω	8 pF	5 pF

- 备注
1. 表 10-4 所显示的电阻和电容值不是保证值。
 2. $n = 0 \sim 7$

第十一章 D/A 转换器

11.1 D/A 转换器的功能

D/A 转换器用于将输入的数字信号转换成模拟信号，有 8 位分辨率。它可以控制两个通道（ANO0 和 ANO1）的输出模拟信号。D/A 转换器 具有如下特性。

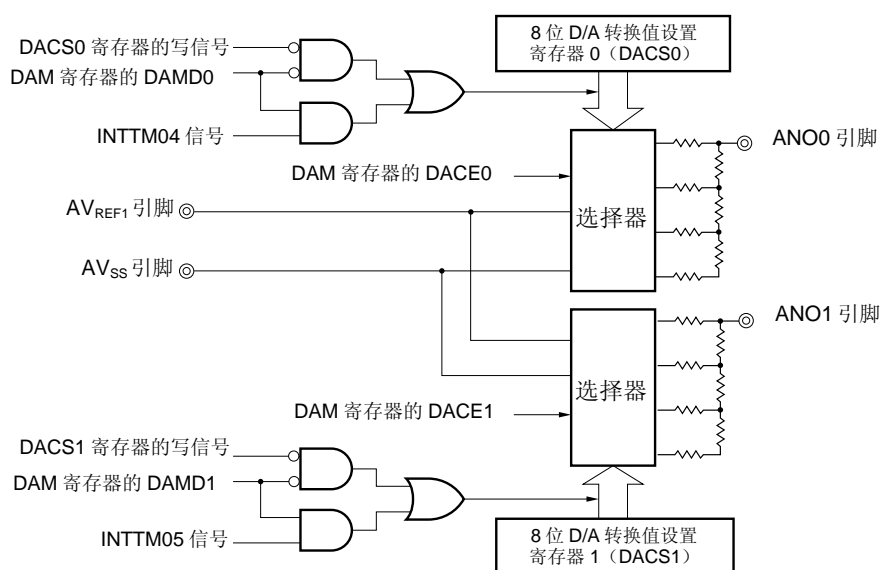
- 8 位分辨率 × 2 通道
- R-2R 梯形法
- 输出模拟电压： $AV_{REF1} \times m/256$ （ AV_{REF1} ：D/A 转换器的参考电压， m ：设置到 $DACS_n$ 寄存器的值）
- 操作模式：普通模式/实时输出模式

备注 $n = 0, 1$

11.2 D/A 转换器的配置

D/A 转换器的配置如下所示。

图 11-1. D/A 转换器的框图



- 备注
1. INTTM04 和 INTTM05 是定时器触发信号（从定时器通道 4 和 5 产生的中断信号），在实时输出模式下使用。
 2. D/A 转换器的通道 0 和通道 1 共用 AV_{REF1} 引脚。
 3. D/A 转换器的通道 0 和通道 1 共用 AV_{SS} 引脚。 AV_{SS} 也由 D/A 转换器共享。

D/A 转换器包括以下硬件

表 11-1. D/A 转换器的配置

项目	配置
控制寄存器	外围设备允许寄存器 0 (PER0) D/A 转换器模式寄存器 (DAM) 8 位 D/A 转换值设置寄存器 0, 1 (DACS0, DACS1)

11.3 D/A转换器中使用的寄存器

D/A 转换器使用下面三个寄存器。

- 外围设备允许寄存器 0 (PER0)
- D/A 转换器模式寄存器 (DAM)
- 8 位 D/A 转换值设置寄存器 0, 1 (DACS0, DACS1)

(1) 外围设备允许寄存器 0 (PER0)

PER0 用于允许或禁止每个外围硬件模块操作。为了降低功耗和噪声，对于不使用的硬件模块停止提供时钟。

使用 D/A 转换器时，确保设置此寄存器的第 6 位 (DACEN) 为 1。

PER0 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将该寄存器清零 (00H)。

注意事项 1. 设置 D/A 转换器时，首先务必设置 DACEN 为 1。如果 DACEN = 0，写入到 D/A 转换器的控制寄存器被忽略，即使读取寄存器，只能读到默认值。

2. 确保将 PER0 的第 1 位清 0。

图 11-2. 外围设备允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

DACEN	D/A 转换器 输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 不能写入被 D/A 转换器使用的 SFR。 • D/A 转换器在复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 可读/写被 D/A 转换器使用的 SFR。

(2) D/A 转换器模式寄存器 (DAM)

此寄存器控制 D/A 转换器的操作。

DAM 可以通过 1 位或 8 位存储器操作指令设置。

复位信号产生将该寄存器清零 (00H)。

图 11-3. D/A 转换器模式寄存器 (DAM) 的格式

地址: FFF32H 复位后: 00H R/W

符号	7	6	<5>	<4>	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEn	D/A 转换操作的控制 (n = 0, 1)	
0	停止转换操作	
1	允许转换操作	

DAMDn	D/A 转换器操作模式的选择 (n = 0, 1)	
0	普通模式	
1	实时输出模式	

(3) 8 位 D/A 转换值设置寄存器 0 和 1 (DACS0, DACS1)

这些寄存器用于设置要输出到 ANO0 和 ANO1 引脚的模拟电压值。

DACS0 和 DACS1 可以通过 8 位存储器操作指令读取。

复位信号产生将该寄存器清零 (00H)。

图 11-4. 8 位 D/A 转换值设置寄存器 0 和 1 (DACS0, DACS1) 的格式

地址: FFF1CH, FFF1DH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DACS _n	DACS _n 7	DACS _n 6	DACS _n 5	DACS _n 4	DACS _n 3	DACS _n 2	DACS _n 1	DACS _n 0

备注 n = 0, 1

11.4 D/A转换器的操作

11.4.1 在普通模式下操作

对 DACSn 寄存器进行写入操作，可以触发执行 D/A 转换。设置方法描述如下。

- <1> 设置 DAM 寄存器的 DAMDn 位为 0（普通模式）。
- <2> 设置要从 ANOn 引脚输出的模拟电压值到 DACSn 寄存器。
上面的步骤 <1>和<2> 是初始化设置。
- <3> 设置 DAM 寄存器的 DACE n 位为 1（D/A 转换允许）。
当此设置执行时 D/A 转换开始，经过稳定时间后，模拟电压被输出到 ANOn 引脚。
- <4> 要执行后面的 D/A 转换，写入到 DACSn 寄存器。
保存前一个 D/A 转换结果，直到执行下一个 D/A 转换。
当 DAM 寄存器的 DACE n 位设置为 0（D/A 转换操作停止）时，D/A 转换停止。当 PM11 寄存器的 PM11n 位= 1（输入模式）时，ANOn 引脚为高阻抗状态。当 PM11n 位 = 0（输出模式）时，ANOn 引脚输出 P11 寄存器的设置值。

- 备注**
- 1. 复用功能引脚设置的详细情况，参见表 4-6 使用端口引脚作为复用功能引脚。
 - 2. $n = 0, 1$

11.4.2 在实时输出模式下操作

使用定时器通道 4 和定时器通道 5 的中断请求信号（INTTM04 和 INTTM05）作为触发执行 D/A 转换。设置方法描述如下。

- <1> 设置 DAM 寄存器的 DAMDn 位为 1（实时输出模式）。
- <2> 设置要从 ANOn 引脚输出的模拟电压值到 DACSn 寄存器。
- <3> 设置 DAM 寄存器的 DACE n 位为 1（D/A 转换允许）。
上面的步骤 <1>到<3> 是初始化设置。
- <4> 操作定时器通道 4 和定时器通道 5。
- <5> 当 INTTM04 和 INTTM05 信号发生时 D/A 转换开始，经过稳定时间后，模拟电压被输出到 ANOn 引脚。
- <6> 之后，每次 INTTM04 和 INTTM05 产生，设置在 DACSn 寄存器中的值被输出。
在下一个 D/A 转换开始（INTTM04 和 INTTM05 信号产生）前，设置要从 ANOn 引脚输出的模拟电压值到 DACSn 寄存器。
当 DAM 寄存器的 DACE n 位设置为 0（D/A 转换操作停止）时，D/A 转换停止。当 PM11 寄存器的 PM11n 位= 1（输入模式）时，ANOn 引脚为高阻抗状态，当 PM11n 位 = 0（输出模式）时，ANOn 引脚输出 P11 寄存器的设置值。

- 备注**
- 1. ANO0 和 ANO1 引脚的输出值直到上面的步骤<5>之前不确定。
 - 2. 对于 ANO0 和 ANO1 引脚在 HALT 和 STOP 模式下的输出值，参见第十八章 待机功能。
 - 3. 复用功能引脚设置的详细情况，参见表 4-6 使用端口引脚作为复用功能引脚。
 - 4. $n = 0, 1$

11.4.3 注意事项

当使用 78K0R/KF3 的 D/A 转换器时注意下面的注意事项。

- (1) 在 D/A 转换期间不能使用数字端口 I/O 功能，也就是 ANO0 和 ANO1 引脚的复用功能。
在 D/A 转换期间读取 P11 寄存器时，在输入模式下读取到 0，在输出模式下读取为 P11 寄存器的设置值。如果设置为数字输出模式，则没有输出数据输出到引脚。
- (2) 在 D/A 转换期间不要读/写 P11 寄存器，也不要改变 PM11 寄存器的设置（否则转换精度可能会降低）。
- (3) 建议 ANO0 和 ANO1 两个引脚同使用作模拟输出引脚或数字 I/O 引脚，也就是说，这两个通道作为相同功能使用（如果这两个引脚用作不同的应用功能，转换精度可能会降低）。
- (4) 在实时输出模式下，在定时器触发产生前设置 DACSn 寄存器值。另外，当触发信号输出时，不要改变 DACSn 寄存器的值。
- (5) 在改变操作模式前，务必要把 DAM 寄存器的 DACEn 位清 0（D/A 转换停止）。
- (6) 当使用的端口功能为 ANO0 或 ANO1 引脚时，端口输入的电平应该尽量平稳。
- (7) 提供给 AVREF1 的电压时序与 AVREF0（A/D 转换器参考电压）相同。
- (8) 因为 D/A 转换器在 STOP 模式下停止操作，ANO0 和 ANO1 引脚变为高阻抗状态，可以降低功耗。
在除 STOP 模式以外的待机模式，操作会继续。如需降低功耗，DAM 寄存器的 DACEn 位清 0（D/A 转换器停止）。
- (9) 因为 D/A 转换器的输出阻抗为高阻抗，无法从 ANOn 引脚（ $n = 0, 1$ ）获得电流。当负载的输入阻抗为低时，在负载和 ANOn 引脚之间放置一个跟随放大器，线的长度尽可能短（针对高阻抗）。如果线太长，可以采取必要的措施，如用接地模式将该线环绕起来。

第十二章 串行阵列单元

串行阵列单元每个单元有四个串行通道，并且可以将两个或更多不同的串行接口（3 线串行（CSI）、UART 和简易 I²C）组合起来使用。

78K0R/KF3 可以支持每个通道的功能如下所示（单元 1 的通道 2 和 3 专用于 UART3（支持 LIN 总线））。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	-
	1	CSI01		-
	2	CSI10	UART1	IIC10
	3			
1	0	CSI20	UART2	IIC20
	1			
	2	-	UART3（支持 LIN 总线）	-
	3	-		-

（组合的举例） 当单元 0 的通道 0 和通道 1 用作“UART0”时，不能使用 CSI00 和 CSI01，但是 CSI10、UART1 或 IIC10 可以使用。

12.1 串行阵列单元的功能

78K0R/KF3 支持的每个串行接口有以下特征。

12.1.1 3 线串行输入/输出（CSI00, CSI01, CSI10, CSI20）

这是一个使用 3 线进行通讯的功能，使用时钟信号来同步：串行时钟（SCK）线和串行数据（SI 和 SO）线。

[数据发送/接收]

- 7 或 8 位的数据长度
- 发送/接收数据的相位控制
- 可以选择 MSB/LSB 先行
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- 输入/输出时钟的相位控制
- 通过每个通道的预分频器和内部计数器来设置发送周期

[中断功能]

- 发送结束中断/缓冲区为空中断

[错误检测标志]

- 溢出错误

12.1.2 UART (UART0, UART1, UART2, UART3)

这是一个使用 2 线进行通讯的功能，使用启动-停止信号来同步：串行数据发送 (TxD) 和串行数据接收 (RxD) 线。它与通信对象异步发送或接收数据（通过使用一个内部波特率）。全双工 UART 通信可以通过使用 2 个通道来实现，一个专用于发送（偶数编号通道），另一个专用于接收（奇数编号通道）。

[数据发送/接收]

- 5、7 或 8 位的数据长度
- 可以选择 MSB/LSB 先行
- 发送/接收数据的电平设置和相反的选择
- 奇偶位附加和奇偶校验功能
- 附加停止位

[中断功能]

- 发送结束中断/缓冲区为空中断
- 在帧错误、奇偶校验错误或溢出错误情况下的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误或溢出错误

LIN 总线可以在 UART3（单元 1 的 2 和 3 通道）上使用

[LIN 总线功能]

- 唤醒信号检测
- 同步间隔区域 (SBF) 检测
- 同步区域测量，波特率计算

使用外部中断 (INTP0) 或定时器阵列单元 (TAU)。

12.1.3 简易 I²C (IIC10, IIC20)

这是一个使用 2 线与两个或更多设备进行同步通信的功能：串行时钟 (SCL) 和串行数据 (SDA)。这个简易 I²C 适用于与一个设备单独通信，例如 EEPROM、flash 存储器或模/数转换器。因此，它只能用作主机方面而不能检测等待状态。

使用软件，也就是操作那些控制寄存器，来确保启动和停止情况下遵守 AC 规范。

[数据发送/接收]

- 主机发送、主机接收（只有单个主机的主机功能）
- ACK 输出和 ACK 检测功能
- 8 位的数据长度（当传送地址时，通过高 7 位指定地址，最低位用作读/写控制。）
- 启动条件和停止条件的手动产生

[中断功能]

- 发送结束中断

[错误检测标志]

- 奇偶校验错误 (ACK 错误)

* [简易 I²C 不支持的功能]

- 从机发送、从机接收
- 仲裁失利检测功能
- 等待检测功能

备注 要使用全功能的 I²C 总线，见第十三章 串行接口 IIC0。

12.2 串行阵列单元的配置

串行阵列单元包含以下硬件。

表 12-1. 串行阵列单元的配置

项目	配置
移位寄存器	8 位
缓冲寄存器	串行数据寄存器 mn (SDRmn) ^注 的低 8 位
串行时钟 I/O	SCK00, SCK01, SCK10, SCK20 引脚 (对于 3 线串行 I/O), SCL10, SCL20 引脚 (对于简易 I ² C)
串行数据输入	SI00, SI01, SI10, SI20 引脚 (对于 3 线串行 I/O), RxD0, RxD1, RxD2 引脚 (对于 UART), RxD3 引脚 (对于 UART 支持的 LIN 总线)
串行数据输出	SO00, SO01, SO10, SO11, SO20, SO21 引脚 (对于 3 线串行 I/O), TxD0, TxD1, TxD2 引脚 (对于 UART), TxD3 引脚 (对于支持 LIN 总线的 UART), 输出控制器
串行数据输入/输出	SDA10, SDA20 引脚 (对于简易 I ² C)
控制寄存器	<p><单元设置块的寄存器></p> <ul style="list-style-type: none"> • 外围允许寄存器 0 (PER0) • 串行时钟选择寄存器 m (SPSm) • 串行通道允许状态寄存器 m (SEm) • 串行通道启动寄存器 m (SSm) • 串行通道停止寄存器 m (STm) • 串行输出允许寄存器 m (SOEm) • 串行输出寄存器 m (SOM) • 串行输出电平寄存器 m (SOLm) • 输入切换控制寄存器 (ISC) • 噪声滤波器允许寄存器 0 (NFEN0) <p><每个通道的寄存器></p> <ul style="list-style-type: none"> • 串行数据寄存器 mn (SDRmn) • 串行模式寄存器 mn (SMRmn) • 串行通信操作设置寄存器 mn (SCRmn) • 串行状态寄存器 mn (SSRmn) • 串行标志清除触发寄存器 mn (SIRmn) • 端口输入模式寄存器 0, 4, 14 (PIM0, PIM4, PIM14) • 端口输出模式寄存器 0, 4, 14 (POM0, POM4, POM14) • 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14) • 端口寄存器 0, 1, 4, 14 (P0, P1, P4, P14)

注 根据通信模式的不同，串行数据寄存器 mn (SDRmn) 的低 8 位可以使用以下 SFR 来读取或写入，

- CSIp 通信 ... SIOp (CSIp 数据寄存器)
- UARTq 接收 ... RXDq (UARTq 接收数据寄存器)
- UARTq 发送 ... TXDq (UARTq 发送数据寄存器)
- IICr 通信 ... SIOr (IICr 数据寄存器)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3),
p: CSI 编号 (p = 00, 01, 10, 20), q: UART 编号 (q = 0 ~ 3), r: IIC 编号 (r = 10, 20)

图 12-1 表示串行阵列单元 0 的框图。

图 12-1. 串行阵列单元 0 的框图

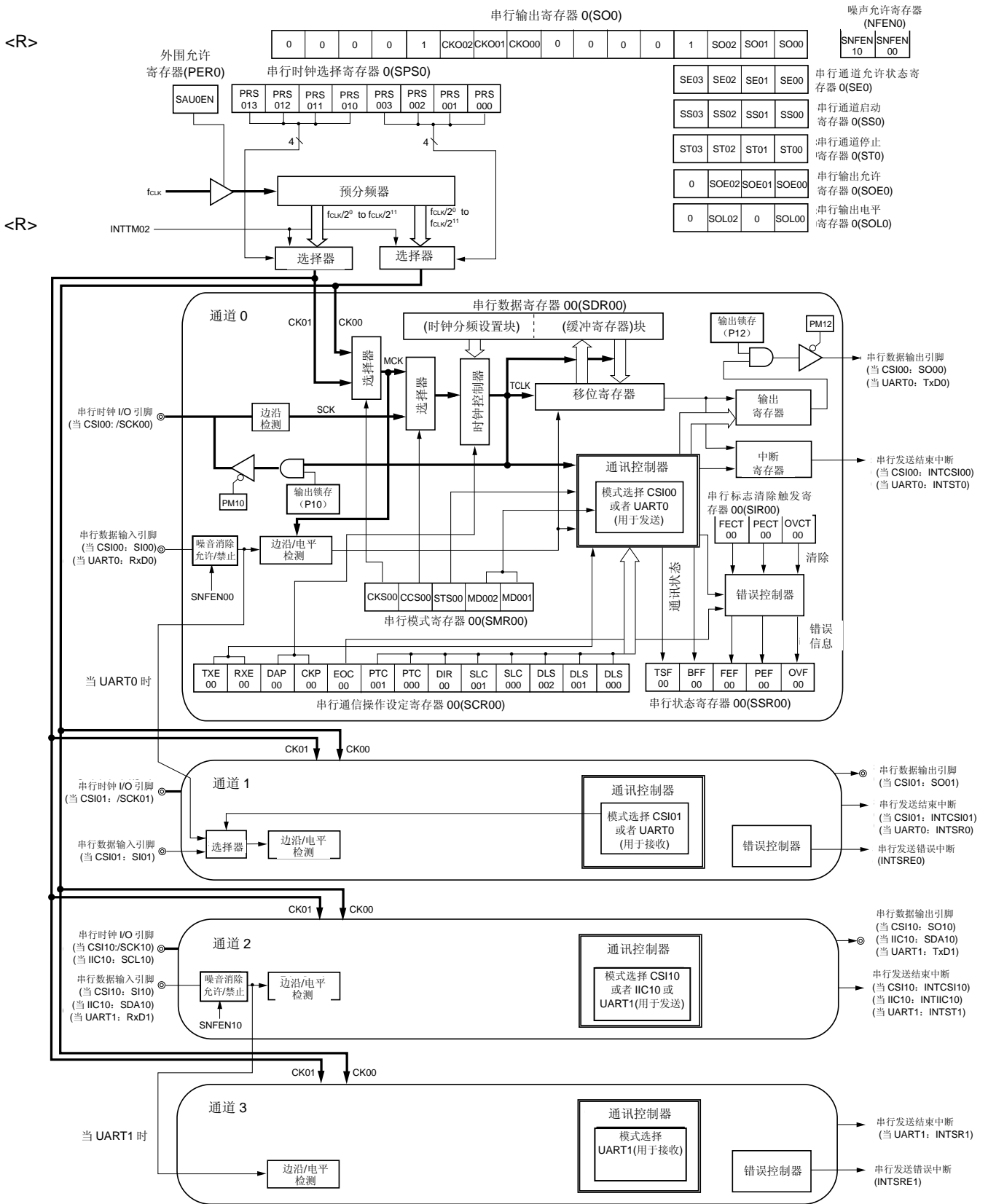
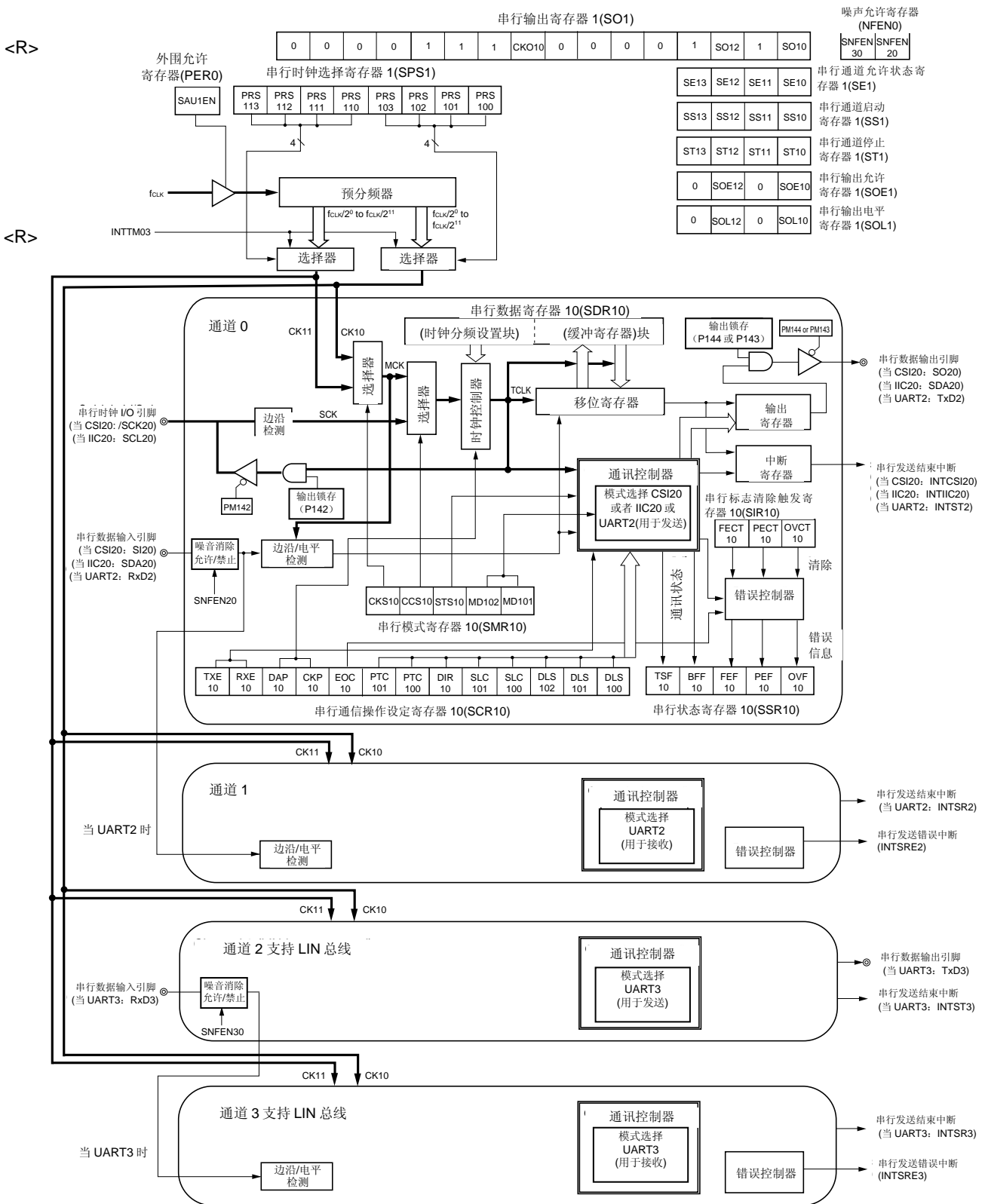


图 12-2 表示串行阵列单元 1 的框图。

图 12-2. 串行阵列单元 1 的框图

<R>



<R>

<R>

(1) 移位寄存器

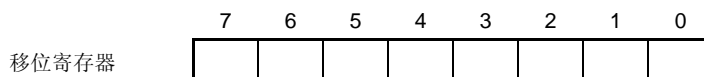
这是一个 8 位寄存器，可以将并行数据转换到串行数据，或将串行数据转换到并行数据。

接收过程中，它将输入到串行引脚的数据转换为并行数据。

当数据被发送时，被设置到该寄存器的值被转换为串行数据从串行输出引脚输出。

移位寄存器不能由程序直接操作。

要读取或写入移位寄存器，使用串行数据寄存器 mn (SDRmn) 的低 8 位。



(2) 串行数据寄存器 mn (SDRmn) 的低 8 位

SDRmn 是通道 n 的发送/接收寄存器 (16 位)。第 7 位到第 0 位作为发送/接收缓冲寄存器，位 15 到 9 被用作设置操作时钟 (MCK) 分频系数的寄存器。

当接收数据时，移位寄存器转换得到的并行数据被保存到低 8 位。当发送数据时，要被传送到移位寄存器的数据存放到低 8 位。

根据 SCRmn 寄存器的位 0~2 (DLSmn0~DLSmn2) 的设置，保存到低 8 位的数据如下所示，与数据的输出顺序无关。

- 5 位的数据长度 (保存在 SDRmn 寄存器的第 0~4 位中) (只有在 UART 模式下可以设置)
- 7 位的数据长度 (保存在 SDRmn 寄存器的第 0~6 位中)
- 8 位的数据长度 (保存在 SDRmn 寄存器的第 0~7 位中)

SDRmn 可以以 16 位为单位来读取或写入。

根据通信模式不同，SDRmn 的低 8 位可以通过下面的 SFR 来读取或写入^注：

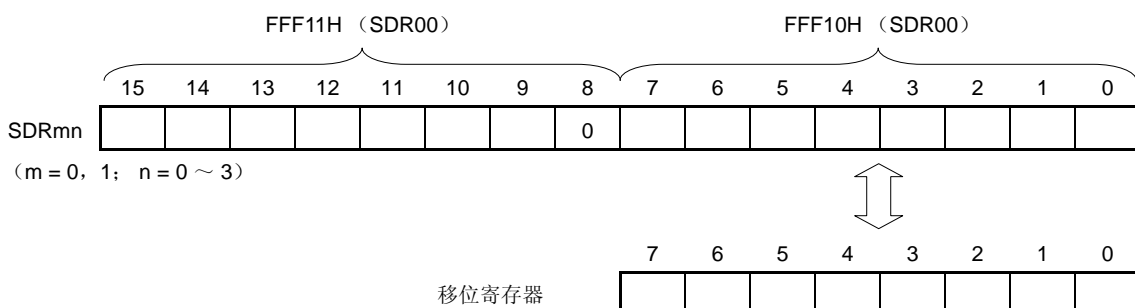
- CSIp 通信 ... SIOp (CSIp 数据寄存器)
 - UARTq 接收 ... RXDq (UARTq 接收数据寄存器)
 - UARTq 发送 ... TXDq (UARTq 发送数据寄存器)
 - IICr 通信 ... SIOr (IICr 数据寄存器)
- 注** 当操作被停止 (SEmn = 0) 时，禁止以 8 位单元写入。

复位信号会清除该寄存器为 0000H。

- 备注**
1. 数据被接收后，“0”被保存到超出数据长度的的位部分中的第 0~7 位。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3),
p: CSI 编号 (p = 00, 01, 10, 20), q: UART 编号 (q = 0 ~ 3), r: IIC 编号 (r = 10, 20)

图 12-3. 串行数据寄存器 mn (SDRmn) 的格式

地址: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), 复位后: 0000H R/W
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



注意事项 确保清除第 8 到 0 位。

- 备注**
1. 关于 SDRmn 的高 7 位的功能，见 12.3 控制串行阵列单元的寄存器。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3),
 p: CSI 编号 (p = 00, 01, 10, 20), q: UART 编号 (q = 0~3), r: IIC 编号 (r = 10, 20)

12.3 控制串行阵列单元的寄存器

串行阵列单元被以下寄存器控制。

- 外围允许寄存器 (PER0)
- 串行时钟选择寄存器 m (SPSm)
- 串行模式寄存器 mn (SMRmn)
- 串行通信操作设置寄存器 mn (SCRmn)
- 串行数据寄存器 mn (SDRmn)
- 串行状态寄存器 mn (SSRmn)
- 串行标志清除触发寄存器 mn (SIRmn)
- 串行通道允许状态寄存器 m (SEm)
- 串行通道启动寄存器 m (SSm)
- 串行通道停止寄存器 m (STm)
- 串行输出允许寄存器 m (SOEm)
- 串行输出电平寄存器 m (SOLm)
- 串行输出寄存器 m (SOM)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器 0 (NFEN0)
- 端口输入模式寄存器 0, 4, 14 (PIM0, PIM4, PIM14)
- 端口输出模式寄存器 0, 4, 14 (POM0, POM4, POM14)
- 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14)
- 端口寄存器 0, 1, 4, 14 (P0, P1, P4, P14)

备注 m: 单元编号 (m = 0, 1)
 n: 通道编号 (n = 0 ~ 3)

(1) 外围允许寄存器 0 (PER0)

PER0 用来允许或禁止每个外围硬件模块的使用。为了降低功耗和噪声，对于不使用的硬件模块停止提供时钟。

当使用串行阵列单元 0 时，确保设置该寄存器的位 2 (SAU0EN) 为 1。

当使用串行阵列单元 1 时，确保设置该寄存器的位 3 (SAU1EN) 为 1。

PER0 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号清除该寄存器为 00H。

图 12-4. 外围允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	串行阵列单元m输入时钟的控制
0	停止供给输入时钟。 <ul style="list-style-type: none"> • 串行阵列单元使用的SFR不能被写入。 • 串行阵列单元m处于复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> • 串行阵列单元m使用的SFR可以被读取/写入。

- 注意事项**
1. 当设置串行阵列单元 m 时，确保首先设置 SAUmEN 为 1。如果 SAUmEN = 0，对串行阵列单元 m 的控制寄存器的写入被忽略。如果寄存器被读取，只能得到默认值（输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器 (NFEN0)、端口输入模式寄存器 (PIM0)、端口输出模式寄存器 (POM0)、端口模式寄存器 (PM0, PM1) 和端口寄存器 (P0, P1) 除外)。
 2. 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。
 3. 确保将 PER0 寄存器的第 0 位清 0。

备注 m: 单元编号 (m = 0, 1)

(2) 串行时钟选择寄存器 m (SPSm)

SPSm 是用来选择提供给每个通道的两种类型操作时钟 (CKm0, CKm1) 的 16 位寄存器。通过 SPSm 的第 7 位到第 4 位选择 CKm1，通过 SPSm 的第 3 位到第 0 位选择 CKm0。

当寄存器在操作 (当 SEMn = 1 时) 中时，禁止修改 SPSm。

SPSm 可以通过 16 位存储器操作指令来设置。

通过 8 位存储器操作指令操作 SPSmL 也可以设置 SPSm 的低 8 位。

复位信号清除该寄存器为 0000H。

图 12-5. 串行时钟选择寄存器 m (SPSm) 的格式

地址: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 复位后: 0000H R/W

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

PRS mp3	PRS mp2	PRS mp1	PRS mp0	操作时钟的选择 (CKmp) ^{注1}	f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
					0	0	0	0
0	0	0	1	f _{CLK} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	f _{CLK} /2 ⁷	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{注2}				
其他				禁止设置				

<R>

- <R> 注 1. 当更改 f_{CLK} 的时钟时 (通过更改系统时钟控制寄存器 (CKC) 的值), 应该在串行阵列单元 (SAU) 的操作停止 (STm = 000FH) 后再进行。当选择 INTTM02 和 INTTM03 作为操作时钟时, 也要停止定时器阵列单元 (TAU) (TT0 = 00FFH)。
- <R> 2. 通过设置 TAU 的 TIS0 寄存器的 TIS02 (如果 m = 0) 和 TIS03 (如果 m = 1) 位为 1、选择 f_{SUB}/4 作为输入时钟, 并使用 SPSm 寄存器选择 INTTM02 和 INTTM03, SAU 可以在副系统时钟固定分频情况下工作, 而与 f_{CLK} 的频率 (主系统时钟、副系统时钟) 无关。但是, 当更改 f_{CLK} 时, 按照上面注 1 的描述, SAU 和 TAU 必须停止。

- 注意事项**
1. 确保清除位 15 到 8 为“0”。
 2. 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

- 备注**
1. f_{CLK}: CPU/外围硬件时钟频率
f_{SUB}: 副系统时钟频率
 2. m: 单元编号 (m = 0, 1), p = 0, 1

(3) 串行模式寄存器 mn (SMRmn)

SMRmn 是设置通道 n 工作模式的寄存器。它也被用来选择操作时钟 (MCK)、指定串行时钟 (SCK) 是否可以输入、设置作为启动触发器、作为一种操作模式 (CSI, UART 或 I²C) 或作为中断源。只有在 UART 模式下, 该寄存器可以用来反转接收数据的电平。

当寄存器在操作 (当 SEMn = 1 时) 时, 禁止修改 SMRmn。但是, 在操作过程中 MDmn0 位可以被修改。

SMRmn 可以通过 16 位存储器操作指令来设置。

复位信号设置该寄存器为 0020H。

图 12-6. 串行模式寄存器 mn (SMRmn) 的格式 (1/2)

地址: F0110H, F0111H (SMR00) ~ F0116H, F0117H (SMR03), 复位后: 0020H R/W
 F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),
 F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

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	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	通道n的操作时钟 (MCK) 的选择
0	通过PRS寄存器设置的预分频器输出时钟CKm0
1	通过PRS寄存器设置的预分频器输出时钟CKm1
边沿检测器使用操作时钟MCK。此外, 根据CCSmn位和SDRmn寄存器高7位的设置, 产生发送时钟 (TCLK)。	

CCS mn	通道n的发送时钟 (TCLK) 选择
0	通过CKSmn位指定的分频操作时钟MCK
1	从SCK引脚输入的时钟 (在CSI模式下的从机发送)
发送时钟TCLK被用于移位寄存器、通信控制器、输出控制器、中断控制器和错误控制器。当CCSmn = 0时, MCK的分频系数由SDRmn寄存器的高7位设置。	

STS mn	启动触发源的选择
0	只有软件触发有效 (CSI、UART发送和简易I ² C时选择)。
1	RxD引脚的有效沿 (UART接收时选择)
在SSm寄存器被设置为 1 后, 当以上源条件满足时, 发送被启动。	

注意事项 确保清除位 13~9、7、4 和 3~0 为“0”。确保设置第 5 位为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3)

图 12-6. 串行模式寄存器 mn (SMRmn) 的格式 (2/2)

地址: F0110H, F0111H (SMR00) ~ F0116H, F0117H (SMR03), 复位后: 0020H R/W
 F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),
 F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

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	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	在UART模式下, 通道n 的接收数据的电平的反转控制	
0	下降沿被检测为起始位。 输入通信数据和捕获数据相同。	
1	上升沿被检测为起始位。 反转输入通信数据然后捕获。	

MD mn2	MD mn1	通道n 的工作模式设置
0	0	CSI模式
0	1	UART模式
1	0	简易I ² C模式
1	1	禁止设置

MD mn0	通道n的中断源选择	
0	发送结束中断	
1	缓冲区为空中断	
对于连续发送, 当SDRmn数据发送完后, 通过设置MDmn0为1, 写入下一个发送数据。		

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3)

(4) 串行通信操作设置寄存器 mn (SCRmn)

SCRmn 是通道 n 的通信操作设置寄存器。它被用来设置数据发送/接收模式、数据和时钟的相位、是否屏蔽错误信号、奇偶位、起始位、停止位和数据长度。

当寄存器在操作 (当 SE_{mn} = 1 时) 时, 修改 SCRmn 被禁止。

SCRmn 可以通过 16 位存储器操作指令来设置。

复位信号设置该寄存器为 0087H。

图 12-7. 串行通信操作设置寄存器 mn (SCRmn) 的格式 (1/3)

地址: F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03), 复位后: 0087H R/W
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

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

TXE mn	RXE mn	通道n的工作模式设置
0	0	不启动通信。
0	1	只接收
1	0	只发送
1	1	发送/接收

DAP mn	CKP mn	在CSI模式下数据和时钟相位的选择
0	0	
0	1	
1	0	
1	1	

在UART模式和简易I²C模式下, 确保设置 DAPmn, CKPmn = 0, 0。

注意事项 确保清除第 3、6 和 11 位为“0”。确保设置第 2 位为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3),
 p: CSI 编号 (p = 00, 01, 10, 20)

图 12-7. 串行通信操作设置寄存器 mn (SCRmn) 的格式 (2/3)

地址: F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03), 复位后: 0087H R/W
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

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

EOC mn	错误中断信号屏蔽的选择 (INTSREx (x = 0~3))	
0	屏蔽错误中断INTSREx (不屏蔽INTSRx)。	
1	允许错误中断INTSREx的产生 (如果错误发生, 屏蔽INTSRx)。	
在CSI模式、简易I ² C模式下以及在UART发送*过程中, 设置EOCmn = 0。 在UART接收过程中, 设置EOCmn = 1。		

PTC mn1	PTC mn0	在UART模式中, 奇偶位的设置	
		发送	接收
0	0	不输出奇偶校验位。	无奇偶校验位的接收
0	1	输出0奇偶校验位。	没有奇偶校验位判断
1	0	输出偶校验位。	判断为偶校验位。
1	1	输出奇校验位。	判断为奇校验位。
在CSI模式和简易I ² C模式下, 确保设置 PTCmn1, PTCmn0 = 0, 0。			

DIR mn	在CSI和UART模式下, 数据发送顺序的选择	
0	输入/输出数据MSB先行。	
1	输入/输出数据LSB先行。	
在简易I ² C模式下, 确保清除DIRmn = 0。		

SLC mn1	SLC mn0	UART模式下的停止位设置	
0	0	没有停止位。	
0	1	停止位长度 = 1 位	
1	0	停止位长度 = 2 位	
1	1	禁止设置	
当选择发送结束中断时, 在所有停止位被完全发送后, 产生中断。 在UART接收过程中以及简易I ² C模式下, 设置 1 位停止位 (SLCmn1, SLCmn0 = 0, 1) 。 在CSI模式下, 设置没有停止位 (SLCmn1, SLCmn0 = 0, 0)。			

<R> 注 当不在 EOC01 = 0 的情况下使用 CSI01 时, 可能产生错误中断 INTSRE0。

注意事项 确保清除第位 3、6 和 11 位为“0”。确保设置第 2 位为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

图 12-7. 串行通信操作设置寄存器 mn (SCRmn) 的格式 (3/3)

地址: F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03), 复位后: 0087H R/W
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

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

DLS mn2	DLS mn1	DLS mn0	在CSI和UART模式下, 数据长度的选择
1	0	0	5位的数据长度 (保存在SDRmn寄存器的第0位到第4位) (只有在UART模式下可以选择)
1	1	0	7位的数据长度 (保存在SDRmn寄存器的第0位到第6位中)
1	1	1	8位的数据长度 (保存在SDRmn寄存器的第0位到第7位中)
其他			禁止设置
在简易I ² C模式下, 确保设置DLSmn0 = 1。			

注意事项 确保清除位 3、6 和 11 为“0”。确保设置位 2 为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

(5) 串行数据寄存器 (SDRmn) 的高 7 位

SDRmn 是通道 n 的发送/接收数据寄存器 (16 位)。第 7 位到第 0 位用作发送/接收缓冲寄存器, 第 15 位到第 9 位用来设置操作时钟 (MCK) 的分频系数。如果串行模式寄存器 (SMRmn) 的 CCSmn 位被清除为 0, SDRmn 的高 7 位设置的操作时钟分频后被设置为发送时钟。

关于 SDRmn 的低 8 位的功能, 见 12.2 串行阵列单元的配置。

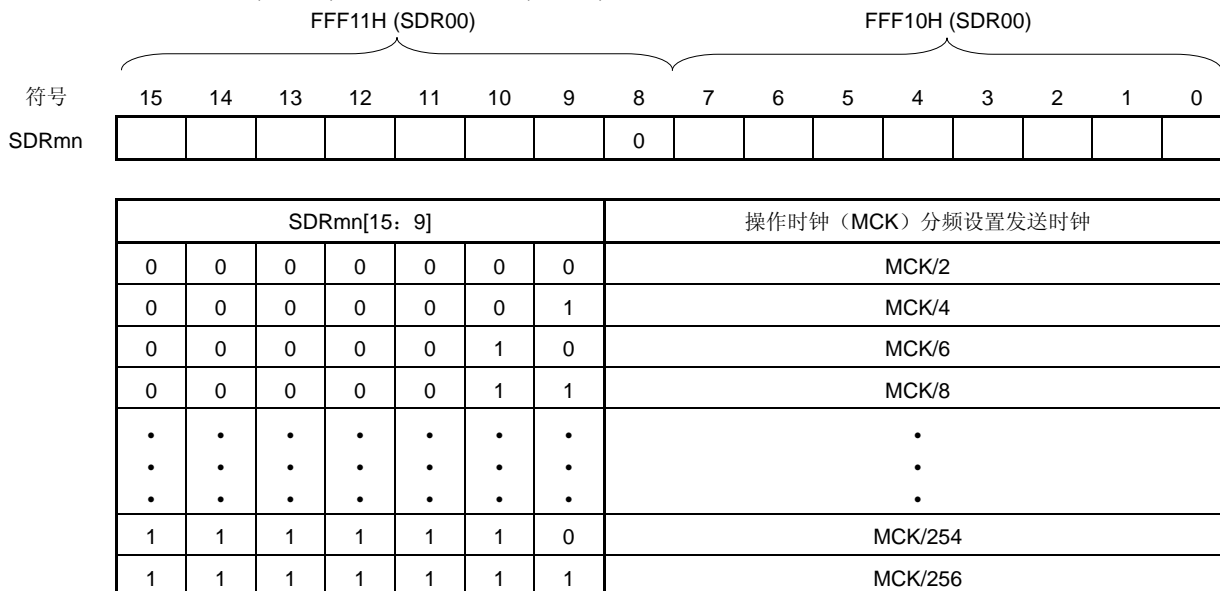
SDRmn 可以以 16 位为单位被读取或写入。

但是, 只有当操作被停止 (SEmn = 0) 时, 高 7 位可以被写入或读取。工作过程中 (SEmn = 1), 只能写入 SDRmn 的低 8 位。在操作过程中读取 SDRmn 时, 总是读出 0。

复位信号清除该寄存器为 0000H。

图 12-8. 串行数据寄存器 mn (SDRmn) 的格式

地址: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), 复位后: 0000H R/W
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



- 注意事项
1. 确保清除第 8 位为“0”。
 2. 当使用 UART 时, SDRmn[15: 9] = (0000000B, 0000001B) 被禁止设置。

- 备注
1. 关于 SDRmn 的低 8 位的功能, 见 12.2 串行阵列单元的配置。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

(6) 串行状态寄存器 mn (SSRmn)

SSRmn 是指通道 n 的通信状态和错误状态的寄存器。该寄存器可以表示帧错误、奇偶校验错误和溢出错误。

SSRmn 可以通过 16 位存储器操作指令来读取。

可以通过 8 位存储器操作指令操作 SSRmnL 来设置的 SSRmn 低 8 位。

复位信号清除该寄存器为 0000H。

图 12-9. 串行状态寄存器 mn (SSRmn) 的格式 (1/2)

地址: F0100H, F0101H (SSR00) ~ F0106H, F0107H (SSR03), 复位后: 0000H R
 F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),
 F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

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	PEF mn	OVF mn

TSF mn	通道n的通信状态指示标志
0	通信没有执行。
1	通信被执行。
因为这个标志是个更新标志, 当通信操作完成时, 它会自动被清除。当STmn/SSmn位被设置为1时, 这个标志也会被清除。	

BFF mn	通道n的缓冲寄存器状态指示标志
0	有效数据未被保存在SDRmn寄存器中。
1	有效数据被保存在SDRmn寄存器中。
这个标志是个更新标志。当从SDRmn寄存器到移位寄存器的发送完成时, 它会自动被清除。接收过程中, 当从SDRmn寄存器读取数据时, 它会自动被清除。当STmn/SSmn位被设置为1时, 这个标志也会被清除。 当SCRmn寄存器的TXEmn位= 1 (每种通信模式下的发送或接收模式) 时, 如果发送数据被写入SDRmn寄存器, 这个标志被自动置位。当SCRmn寄存器的RXEmn位= 1 (每种通信模式下的发送或接收模式) 时, 如果接收数据被保存到SDRmn寄存器, 这个标志也会被自动置位。在接收错误情况下, 它也会被自动置位。 当BFFmn = 1时, 如果数据被写入SDRmn寄存器, 保存在寄存器中的发送/接收数据被放弃, 并且会检测到溢出错误 (OVFmn = 1)。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

图 12-9. 串行状态寄存器 mn (SSRmn) 的格式 (2/2)

地址: F0100H, F0101H (SSR00) ~ F0106H, F0107H (SSR03), 复位后: 0000H R
 F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),
 F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

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	PEF mn	OVF mn

FEF mn	通道n的帧错误检测标志
0	错误没有发生。
1	在UART接收过程中, 一个帧错误发生。 <帧错误原因> 如果在UART接收完成时没有检测到停止位, 会发生帧错误。
这是一个累积标志, 直到SIRmn寄存器的FECTmn位写入“1”之前, 它不会被清除。	

PEF mn	通道n的奇偶校验错误检测标志
0	错误没有发生。
1	在UART接收过程中, 发生奇偶校验错误, 或者在I ² C发送过程中, ACK没有被检测。 <奇偶校验错误原因> <ul style="list-style-type: none"> 如果在UART接收完成时, 发送数据的奇偶校验位与接受到的奇偶校验位不匹配, 发生奇偶校验错误。 在I²C发送过程中, 如果在ACK接收时序中从机没有返回ACK信号, 没有检测到ACK。
这是一个累积标志, 直到SIRmn寄存器的PECTmn位写入“1”之前, 它不会被清除。	

OVF mn	通道n的溢出错误检测标志
0	没有错误发生。
1	发生溢出错误。 <溢出错误的原因> <ul style="list-style-type: none"> 保存在SDRmn寄存器中的接收数据没有被读取, 并且发送数据被写入或者下一个接收数据已经被写入。 在CSI模式下, 对于从机发送或接收, 发送数据没有准备好。
这是一个累积标志, 直到SIRmn寄存器的OVCTmn位写入“1”之前, 它不会被清除。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

(7) 串行标志清除触发寄存器 mn (SIRmn)

SIRmn 是个触发寄存器，用于通道 n 清除每个错误标志。

当该寄存器的每一位 (FECTmn, PECTmn, OVCTmn) 被设置为 1 时，串行状态寄存器 mn 的对应位 (FEFmn, PEFmn, OVFmn) 被清除为 0。因为 SIRmn 是个触发寄存器，当 SSRmn 的对应位被清除时，它随即被清除。

SIRmn 可以通过 16 位存储器操作指令来设置。

可以通过 8 位存储器操作指令操作 SIRmnL 来设置 SIRmn 的低 8 位。

复位信号清除该寄存器为 0000H。

图 12-10. 串行标志清除触发寄存器 mn (SIRmn) 的格式

地址: F0108H, F0109H (SIR00) ~ F010EH, F010FH (SIR03), 复位后: 0000H R/W
 F0148H, F0149H (SIR10), F014AH, F014BH (SIR11),
 F014CH, F014DH (SIR12), F014EH, F014FH (SIR13)

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	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	清除通道n的帧错误的触发器
0	无触发器操作
1	清除SSRmn 寄存器的FEFmn位为0。

PEC Tmn	清除通道n的奇偶校验错误标志的触发器
0	无触发操作
1	清除SSRmn 寄存器的PEFmn位为0。

OVC Tmn	通道n的溢出错误标志的清除触发器
0	无触发操作
1	清除SSRmn 寄存器的OVFmn位为0。

注意事项 确保清除第 15~3 位为“0”。

- 备注
1. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)
 2. 当读取 SIRmn 寄存器时，总是读到 0000H。

(8) 串行通道允许状态寄存器 m (SEm)

SEm 表示每个通道的数据发送/接收操作是被允许还是被停止。

当串行通道启动寄存器 0 (SSm) 的某位写入“1”时，该寄存器中的对应位被设置为 1。当串行通道停止寄存器 0 (STm) 的某位写入“1”时，该寄存器中的对应位被设置为 0。

允许操作的通道 n 不能通过软件重写下面描述的串行输出寄存器 m (SOM) 的 CKOm 的值，并且通信操作的对应值从串行时钟引脚输出。

对于停止操作的通道 n，可以通过软件设定 SOM 寄存器的 CKOm 的值，并且从串行时钟引脚输出它的值。这样，任何波形都可以通过软件产生，例如起始条件/停止条件。

SEm 可以通过 16 位存储器操作指令来读取。

可以通过 1 位或者 8 位存储器操作指令操作 SEmnL 来设定 SEmn 的低 8 位。

复位后该寄存器的值为 0000H。

图 12-11. 串行通道允许状态寄存器 m (SEm) 的格式

地址: F0120H, F0121H (SE0), F0160H, F0161H (SE1) 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	SEm	SEm	SEm	SEm
													3	2	1	0

SEm n	通道n的操作允许/停止状态指示
0	操作停止（停止时，控制寄存器和移位寄存器的值以及串行时钟输入/输出引脚、串行数据输出引脚、FEF、PEF和OVF的状态被保持 [※] ）。
1	操作被允许。

注 SSRmn 寄存器的第 6 位和第 5 位 (TSFmn, BFFmn) 被清除。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)

(9) 串行通道启动寄存器 m (SSm)

SSm 是用来启动每个通道的通信/计数的触发寄存器。

当该寄存器 (SSmn) 中的某位被写入“1”时，串行通道允许状态寄存器 (SEmn) 中的对应位被设置为 1。因为 SSmn 是个触发位，当 SEmn = 1 时，它随即被清除。

SSm 可以通过 16 位存储器操作指令来设置。

可以通过 1 位或 8 位存储器操作指令操作 SSmL 来设置 SSm 的低 8 位。

复位信号清除该寄存器为 0000H。

图 12-12. 串行通道启动寄存器 m (SSm) 的格式

地址: F0122H, F0123H (SS0), F0162H, F0163H (SS1) 复位后: 0000H R/W

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

SSmn	通道n的操作启动触发
0	无触发操作
1	设置SEmn为1，并且进入通信等待状态（如果通信操作已经被执行，操作被停止，并且启动情况被等待）。

注意事项 确保清除位 15~4 为“0”。

- 备注**
1. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)
 2. 当读取 SSm 寄存器时，总是读到 0000H。

(10) 串行通道停止寄存器 m (STm)

STm 是用来停止每个通道的通信/计数触发寄存器。

当该寄存器 (STmn) 中的某位被写入“1”时，串行通道允许状态寄存器 (SEmn) 中的对应位被清除为 0。因为 STmn 是个触发位，当 SEmn = 0 时，它随即被清除。

STm 可以通过 16 位存储器操作指令来设置。

可以通过 1 位或 8 位存储器操作指令操作 STmL 来设置 STm 的低 8 位。

复位信号清除该寄存器为 0000H。

图 12-13. 串行通道停止寄存器 m (STm) 的格式

地址: F0124H, F0125H (ST0), F0164H, F0165H (ST1) 复位后: 0000H R/W

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

STm n	通道n的操作停止触发
0	无触发操作
1	清除SEmn为0，并停止通信操作。 (停止时，控制寄存器和移位寄存器的值以及串行时钟输入/输出引脚、串行数据输出引脚、FEF、PEF和OVF的状态被保持*)

注 SSRmn 寄存器的第 6 位和第 5 位 (TSFmn, BFFmn) 被清除。

注意事项 确保清除位 15~4 为“0”。

备注 1. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)
 2. 当读取 STm 寄存器时，总是读到 0000H。

(11) 串行输出允许寄存器 m (SOEm)

SOEm 是用来允许或停止每个通道的串行通信输出的寄存器。

允许操作的通道 n 不能通过软件重写下面描述的串行输出寄存器 m (SOM) 的 SOMn 的值，并且通信操作的对应值从串行时钟引脚输出。

对于停止操作的通道 n，可以通过软件设定 SOM 寄存器的 SOMn 的值，并且从串行时钟引脚输出它的值。

这样，任何波形都可以通过软件产生，例如起始条件/停止条件。

SOEm 可以通过 16 位存储器操作指令来设置。

可以通过 1 位或 8 位存储器操作指令操作 SOEmL 来设置 SOEm 的低 8 位。

复位信号清除该寄存器为 0000H。

<R>

图 12-14. 串行输出允许寄存器 m (SOEm) 的格式

地址: F012AH, F012BH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 02	SOE 01	SOE 00

地址: F016AH, F016BH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 12	0	SOE 10

SOE mn	通道n的串行输出允许/禁止
0	停止串行通信操作的输出。
1	允许串行通信操作的输出。

注意事项 确保清除 SOE0 的第 15~3 位和 SOE1 的第 15~3 位和第 1 位为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)
mn = 00 ~ 02, 10, 12

(12) 串行输出寄存器 m (SOm)

SOm 是每个通道的串行输出的缓冲寄存器。

该寄存器第 n 位的值从通道 n 的串行数据输出引脚被输出。

该寄存器第 (n + 8) 位的值从通道 n 的串行时钟输出引脚被输出。

只有当串行输出被禁止 (SOEmn = 0) 时, 可以通过软件写入该寄存器的 SOmn。当串行输出被允许 (SOEmn = 1) 时, 通过软件进行的修改被忽略, 并且寄存器的值只能通过串行通信操作来更改。

只有当串行操作被停止 (SEmn = 0) 时, 可以通过软件写入该寄存器的 CKOmn。当串行操作被允许 (SEmn = 1) 时, 通过软件进行的修改被忽略, 并且 CKOmn 的值只能通过串行通信操作来更改。

<R> 要使用 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/SCK10/SCL10, P10/SCK00, P12/SO00/TxD0, P13/TxD3, P43/SCK01, P45/SO01, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2 或 P144/SO20/TxD2 引脚作为端口功能引脚, 设置对应的 CKOmn 位和 SOmn 位为“1”。

SOm 通过 16 位存储器操作指令来设置。

复位信号清除该寄存器为 0F0FH。

<R> 图 12-15. 串行输出寄存器 m (SOm) 的格式

地址: F0128H, F0129H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	CKO 01	CKO 00	0	0	0	0	1	SO 02	SO 01	SO 00

地址: F0168H, F0169H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO 10	0	0	0	0	1	SO 12	1	SO 10

CKO mn	通道n的串行时钟输出
0	串行时钟输出值为“0”。
1	串行时钟输出值为“1”。

SO mn	通道n的串行数据输出
0	串行数据输出值为“0”。
1	串行数据输出值为“1”。

注意事项 确保设置 SO0 的第 11 位和第 3 位为“1”, 设置 SO1 的第 11~9 位、第 3 位和第 1 位为“1”。同时确保清除 SOm 的第 15~12 和 7~4 位为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3)
mn = 00~02, 10, 12

(13) 串行输出电平寄存器 m (SOLm)

SOLm 是用来设置每个通道数据输出电平反转的寄存器。

该寄存器只能在 UART 模式下设置。在 CSI 模式和简易 I²C 模式下，确保设置为 0000H。

只有当串行输出允许 (SOEmn = 1) 时，使用该寄存器将通道 n 的反转数据输出到引脚上。当串行输出被禁止 (SOEmn = 0) 时，SOmn 位的值不做反转直接输出。

当寄存器在操作中 (当 SEmn = 1 时) 时，禁止修改 SOLm。

SOLm 可以通过 16 位存储器操作指令来设置。

可以通过 8 位存储器操作指令操作 SOLmL 来设置 SOLm 的低 8 位。

复位信号清除该寄存器为 0000H。

图 12-16. 串行输出电平寄存器 m (SOLm) 的格式

地址: F0134H, F0135H (SOL0), F0174H, F0175H (SOL1) 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	SOL m2	0	SOL m0

SOL mn	在UART模式下，选择通道n的发送数据的电平的反转
0	通信数据直接被输出。
1	通信数据被反转后输出。

注意事项 确保清除第 15~4、3 和 1 位为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2)

(14) 输入切换控制寄存器 (ISC)

在 UART3 上使用外部中断和定时器阵列单元，ISC 可以用来实现 LIN 总线通信操作。

当第 0 位被设置为 1 时，串行数据输入 (RxD3) 引脚的输入信号被选择为外部中断 (INTP0)，该中断可以用来检测唤醒信号。

当第 1 位被设置为 1 时，串行数据输入 (RxD3) 引脚的输入信号被选择为定时器输入，所以，同步间隔区域和同步区域的脉冲宽度可以由定时器测量。

ISC 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号清除该寄存器为 00H。

图 12-17. 输入切换控制寄存器 (ISC) 的格式

地址: FFF3CH 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	切换定时器阵列单元的通道 7 输入
0	使用TI07 引脚的输入信号作为定时器输入（正常操作）。
1	RxD3 引脚的输入信号被用作定时器输入（唤醒信号检测）。

ISC0	切换外部中断 (INTP0) 输入
0	使用INTP0 引脚的输入信号作为外部中断（正常操作）。
1	使用RxD3引脚的输入信号作为外部中断 (用来测量同步间隔区域和同步区域的脉冲宽度)。

注意事项 确保清除第 7 ~ 2 位为“0”。

(15) 噪声滤波器允许寄存器 0 (NFEN0)

NFEN0 用来设置每个通道串行数据输入引脚的输入信号是否使用噪声滤波器。

通过清除该寄存器的对应位为 0，禁止 CSI 或简易 I²C 通信引脚的噪声滤波器。

通过设置该寄存器的对应位为 1，允许 UART 通信引脚的噪声滤波器。

当允许适用噪声滤波器时，CPU/外围操作时钟 (f_{CLK}) 由 2 时钟匹配检测进行同步。

NFEN0 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号清除该寄存器为 00H。

图 12-18. 噪声滤波器允许寄存器 0 (NFEN0) 的格式

地址: F0060H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	RxD3/P14引脚的噪声滤波器的使用
0	噪声滤波器关
1	噪声滤波器开
设置SNFEN30为1来使用RxD3引脚。 清除SNFEN30为0来使用P14引脚。	

SNFEN20	RxD2/SDA20/SI20/P143引脚的噪声滤波器的使用
0	噪声滤波器关
1	噪声滤波器开
设置SNFEN20为1来使用RxD2引脚。 清除SNFEN20为0来使用SDA20、SI20和P143引脚。	

SNFEN10	RxD1/SDA10/SI10/P03引脚的噪声滤波器的使用
0	噪声滤波器关
1	噪声滤波器开
设置SNFEN10为1来使用RxD1引脚。 清除SNFEN10为0来使用SDA10、SI10和P03引脚。	

SNFEN00	Use of noise filter of RxD0/SI00/P11 pin
0	噪声滤波器关
1	噪声滤波器开
设置SNFEN00为1来使用RxD0引脚。 清除SNFEN00为0来使用P11引脚。	

注意事项 确保清除位 7、5、3 和 1 为“0”。

(16) 端口输入模式寄存器 0, 4, 14 (PIM0, PIM4, PIM14)

这些寄存器以 1 位为单位设置端口 0、4 和 14 的输入缓冲。

PIM0、PIM4、和 PIM14 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号清除这些寄存器为 00H。

图 12-19. 端口输入模式寄存器 0、4、和 14 (PIM0, PIM4, PIM14) 的格式

地址: F0040H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM0	0	0	0	PIM04	PIM03	0	0	0

地址: F0044H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM4	0	0	0	PIM44	PIM43	0	0	0

地址: F004EH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM14	0	0	0	0	PIM143	PIM142	0	0

PIMmn	Pmn引脚输入缓冲选择 (m = 0, 4, 14; n = 2~4)						
0	正常输入缓冲						
1	TTL输入缓冲						

(17) 端口输出模式寄存器 0, 4, 14 (POM0, POM4, POM14)

这些寄存器以 1 位为单位设置端口 0、4 和 14 的输出模式。

POM0、POM4 和 POM14 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号清除这些寄存器为 00H。

图 12-20. 端口输出模式寄存器 0、4、和 14 (POM0, POM4, POM14) 的格式

地址: F0050H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM0	0	0	0	POM04	POM03	POM02	0	0

地址: F0054H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM4	0	0	POM45	0	POM43	0	0	0

地址: F005EH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM14	0	0	0	POM144	POM143	POM142	0	0

POMmn	Pmn引脚输出缓冲选择 (m = 0, 4, 14; n = 2~5)						
0	正常输出模式。						
1	N-ch 开漏输出 (V _{DD} 耐压) 模式						

(18) 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14)

这些寄存器以 1 位为单位设置端口 0、1、4 和 14 的输入/输出。

当使用 P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/SCK10/SCL10, P10/SCK00, P12/SO00/TxD0, P13/TxD3, P43/SCK01, P45/SO01, P142/SCK20/SCL20, P143/SI20/RxD2/SDA20 和 P144/SO20/TxD2 引脚作为串行数据输出或串行时钟输出时, 清除 PM02~PM04, PM10, PM12, PM13, PM43, PM45 和 PM142~PM144 位为 0, 并且设置 P02~P04, P10, P12, P13, P43, P45 和 P142~P144 的输出锁存为 1。

当使用 P03/SI10/RxD1/SDA10, P04/SCK10/SCL10, P10/SCK00, P11/SI00/RxD0, P14/RxD3, P43/SCK01, P44/SI01, P142/SCK20/SCL20 和 P143/SI20/RxD2/SDA20 引脚作为串行数据输入或串行时钟输入时, 设置 PM03, PM04, PM10, PM11, PM14, PM43, PM44, PM142 和 PM143 位为 1。此时, P03, P04, P10, P11, P14, P43, P44, P142 和 P143 的输出锁存可能是 0 或 1。

PM0, PM1, PM4 和 PM14 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号设置这些寄存器为 FFH。

图 12-21. 端口模式寄存器 0、1、4 和 14 (PM0, PM1, PM4, PM14) 的格式

地址: FFF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

地址: FFF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

地址: FFF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

地址: FFF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn引脚输入/输出模式选择 (m = 0, 1, 4, 14; n = 0~7)
0	输出模式 (输出缓冲开)
1	输入模式 (输出缓冲关)

<R> 12.4 操作停止模式

串行阵列单元的每个串行接口都有操作停止模式。

在这种模式下，串行通信不能执行，因此可以降低功耗。

此外，在这种模式下，P02/SO10/TxD1，P03/SI10/SDA10/RxD1，P04/SCK10/SCL10，P10/SCK00，P11/SI00/RxD0，P12/SO00/TxD0，P13/TxD3，P14/RxD3，P43/SCK01，P44/SI01，P45/SO01，P142/SCK20/SCL20，P143/SI20/SDA20/RxD2 或 P144/SO20/TxD2 引脚可以被用作普通端口引脚。

12.4.1 按照单元停止操作

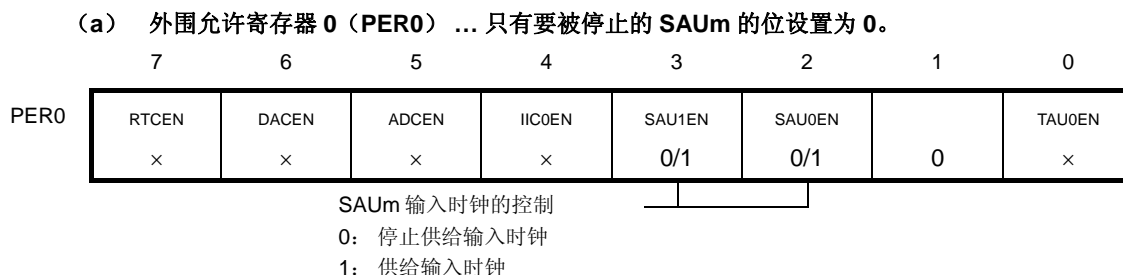
按照单元停止操作的可以通过外围允许寄存器 0 (PER0) 来设置。

PER0 被用作允许或禁止每个外围硬件模块的使用。为了降低功耗和噪声，对于未使用的硬件模块停止供给时钟。

要停止串行阵列单元 0 的操作，设置第 2 位 (SAU0EN) 为 0。

要停止串行阵列单元 1 的操作，设置第 3 位 (SAU1EN) 为 0。

图 12-22. 当按照单元停止操作时，外围允许寄存器 0 (PER0) 的设置



注意事项 1. 如果 SAUmEN = 0，写入串行阵列单元 m 的控制寄存器的操作无效。如果读取寄存器，只能得到默认值（输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器 (NFEN0)、端口输入模式寄存器 (PIM0)、端口输出模式寄存器 (POM0)、端口模式寄存器 (PM0, PM1) 和端口寄存器 (P0, P1) 除外）。

2. 确保将 PER0 寄存器的第 1 位清 0。

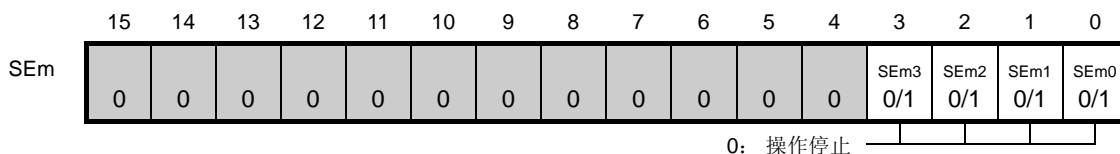
备注 m: 单元编号 (m = 0, 1) : 禁止设置 (由硬件固定)
 x: 串行阵列单元不使用的位 (依赖于其它外围功能的设置)
 0/1: 根据用户的使用来设置为 0 或 1

12.4.2 按照通道停止操作

使用以下每个寄存器，按照通道的操作的停止被设置。

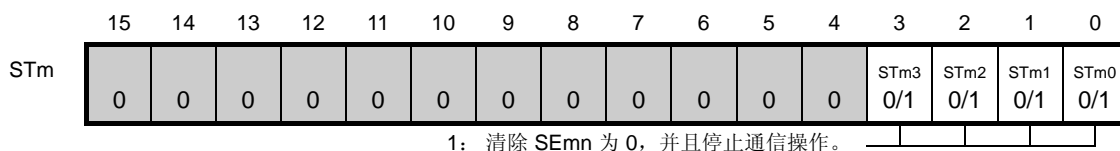
图 12-23. 当按照通道停止操作时，每个寄存器的设置 (1/2)

(a) 串行通道允许状态寄存器 m (SEm) ... 该寄存器表示每个通道的数据发送/接收操作被允许还是被停止。



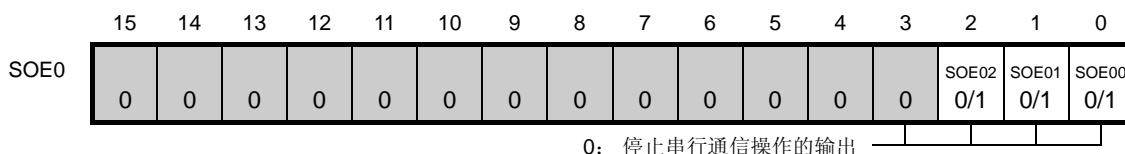
* Sem 寄存器是一个只读寄存器，可以使用 STm 寄存器来停止操作。
对于操作被停止的某个通道，SOM 寄存器的 CKOm 的值可以通过软件设置。

(b) 串行通道停止寄存器 m (STm) ... 该寄存器是用来按照通道停止通信/计数的触发寄存器

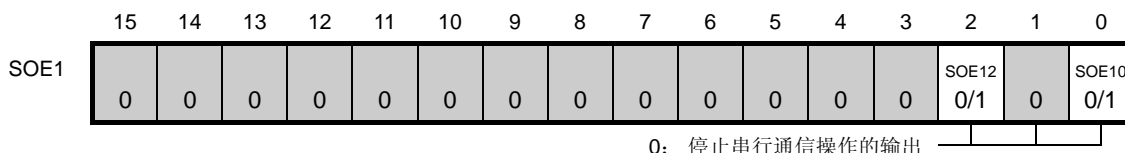


* 因为 STmn 是个触发位，当 SEmn = 0 时，它立即被清除。

(c) 串行输出允许寄存器 m (SOEm) ... 该寄存器是用来允许或停止每个通道的串行通信操作的寄存器。



* 对于串行输出被停止的通道 n，SO0 寄存器的 SO0n 值可以通过软件设置。

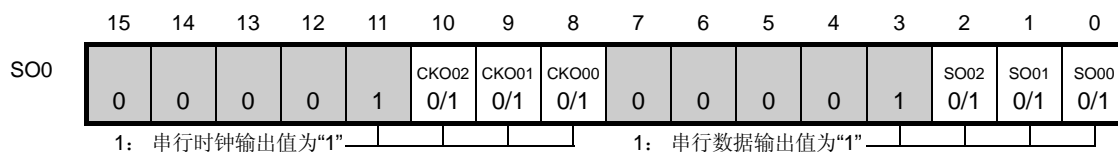


*对于串行输出被停止的通道 n，SO1 寄存器的 SO1n 值可以通过软件设置。

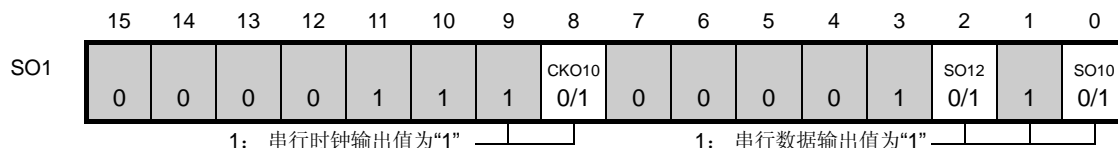
备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3)
: 设置无效 (由硬件固定), 0/1: 根据用户的使用来设置为 0 或 1

图 12-23. 当按照通道停止操作时，每个寄存器的设置 (2/2)

(d) 串行输出寄存器 m (SOm) ...该寄存器是每个通道的串行输出的缓冲寄存器。



* 当使用对应于每个通道的引脚作为端口功能引脚时，设置对应的 CKO0n 和 SO0n 位为“1”。



* 当每个通道的对应引脚用作端口功能引脚时，设置对应的 CKO10 和 SO1n 位为“1”。

备注 m: 单元编号 (m = 0, 1) , n: 通道编号 (n = 0~3)
 □ : 设置无效 (由硬件固定), 0/1: 根据用户的使用来设置为 0 或 1

12.5 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的操作

这是一个使用 3 线进行通讯的功能，使用时钟信号来同步：串行时钟 (SCK) 线和串行数据 (SI 和 SO) 线。

[数据发送/接收]

- 7 或 8 位的数据长度
- 发送/接收数据的相位控制
- 可以选择 MSB/LSB 先行
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- I/O 时钟的相位控制
- 通过每个通道的预分频器和内部计数器来设置发送周期

[中断功能]

- 发送结束中断/缓冲区为空中断

[错误检测标志]

- 溢出错误

SAU0 的通道 0~2 以及 SAU1 的通道 0 支持 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20)。

单元	通道	用作CSI	用作UART	用作简化的I ² C
0	0	CSI00	UART0	-
	1	CSI01		-
	2	CSI10	UART1	IIC10
	3	-		-
1	0	CSI20	UART2	IIC20
	1	-	UART3 (支持LIN总线)	-
	2	-		-
	3	-		-

3 线串行 I/O (CSI00, CSI01, CIS10, CIS11, CSI20, CIS21) 执行下面 6 种类型的通信操作。

- 主机发送 (见 12.5.1.)
- 主机接收 (见 12.5.2.)
- 主机发送/接收 (见 12.5.3.)
- 从机发送 (见 12.5.4.)
- 从机接收 (见 12.5.5.)
- 从机发送/接收 (见 12.5.6.)

12.5.1 主机发送

主机发送是由 78K0R/KF3 输出发送时钟并将数据发送到另一个设备。

<R>

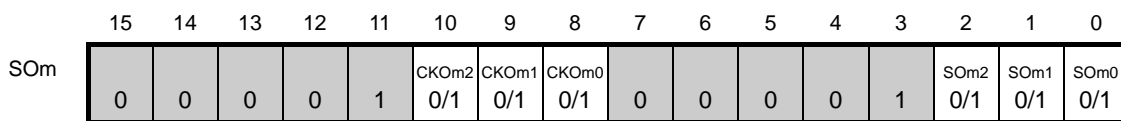
3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择发送结束中断（在单个发送模式下）或者缓冲区为空中断（在连续发送模式下）。			
错误检测标志	无			
发送数据长度	7或8位			
发送速率	最大 $f_{CLK}/4$ [MHz], 最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] [#] f_{CLK} : 系统时钟频率			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 从串行时钟开始工作时, 开始输出数据。 • DAPmn = 1: 在串行时钟开始工作的半个时钟前, 开始输出数据。 			
时钟相位	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB先行			

<R> 注 操作时需要满足以上条件, 且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

(1) 寄存器设置

图 12-24. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 主机发送寄存器内容的举例

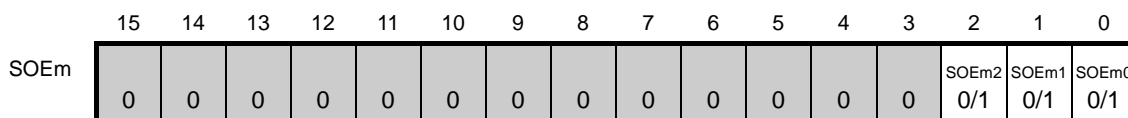
(a) 串行输出寄存器 m (SOm) ... 只设置目标通道的对应位。



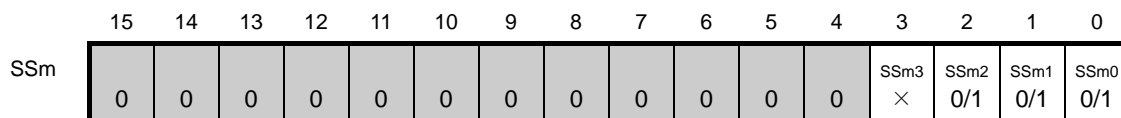
<R>

如果数据相位是正向的 (CKPmn = 0)，当这些位为 1 时，通信启动。如果相位是反向的 (CKPmn = 1)，当这些位为 0 时，通信启动。

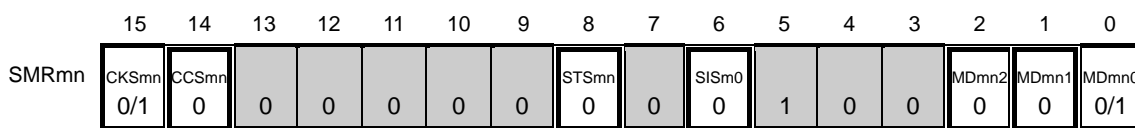
(b) 串行输出使能寄存器 m (SOEm) ... 只设置目标通道的对应位为 1。



(c) 串行通道启动寄存器 m (SSm) ... 只设置目标通道的对应位为 1。

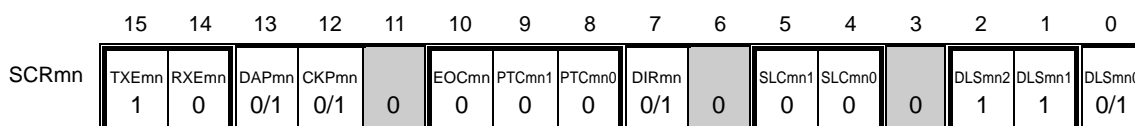


(d) 串行模式寄存器 mn (SMRmn)

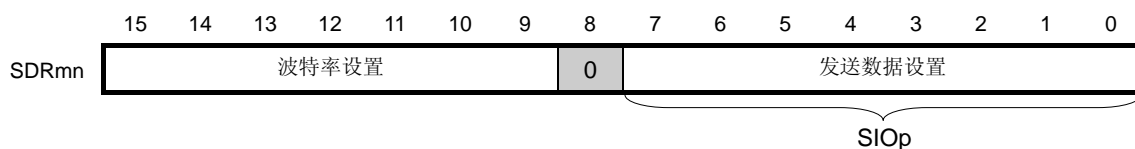


通道 n 的工作模式
0: 发送结束中断
1: 缓冲区为空中断

(e) 串行通信操作设置寄存器 mn (SCRmn)



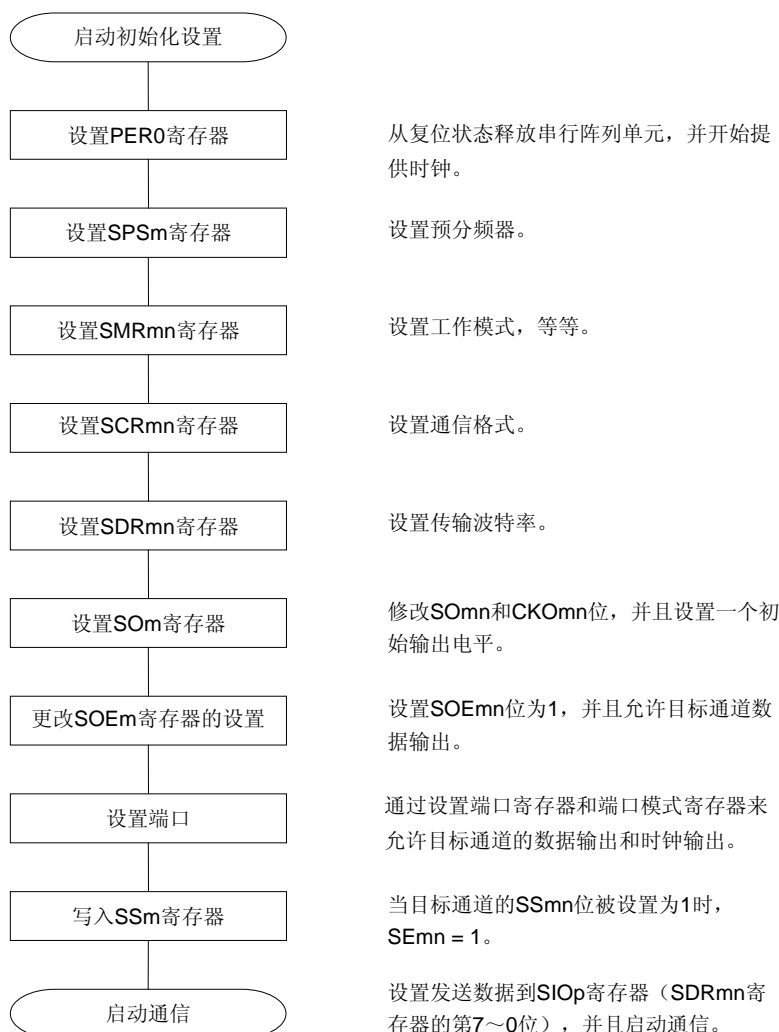
(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)
 □: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

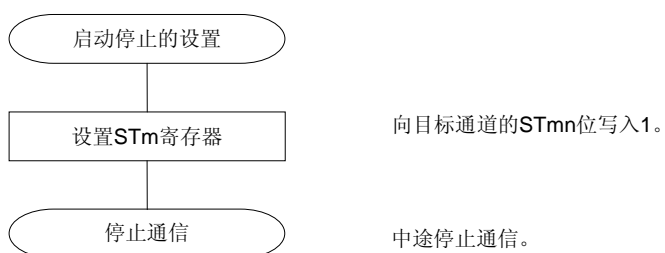
(2) 操作过程

图 12-25. 主机发送的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-26. 停止主机发送的过程



- 备注**
1. 即使停止通信后，引脚的电平仍然会保持。要恢复操作，需重置 SOm 寄存器（见图 12-27 恢复主机发送的过程）。
 2. p: CSI 编号 (p = 00, 01, 10, 20)

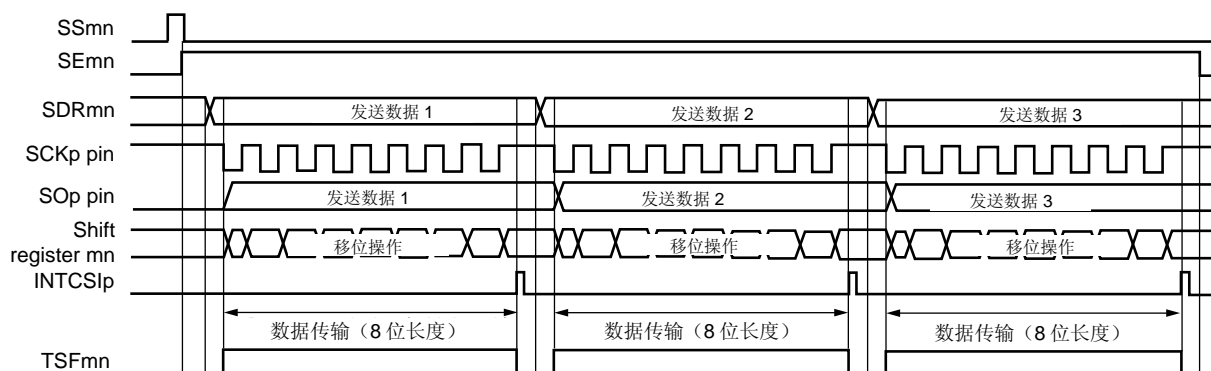
图 12-27 恢复主机发送的过程



<R>

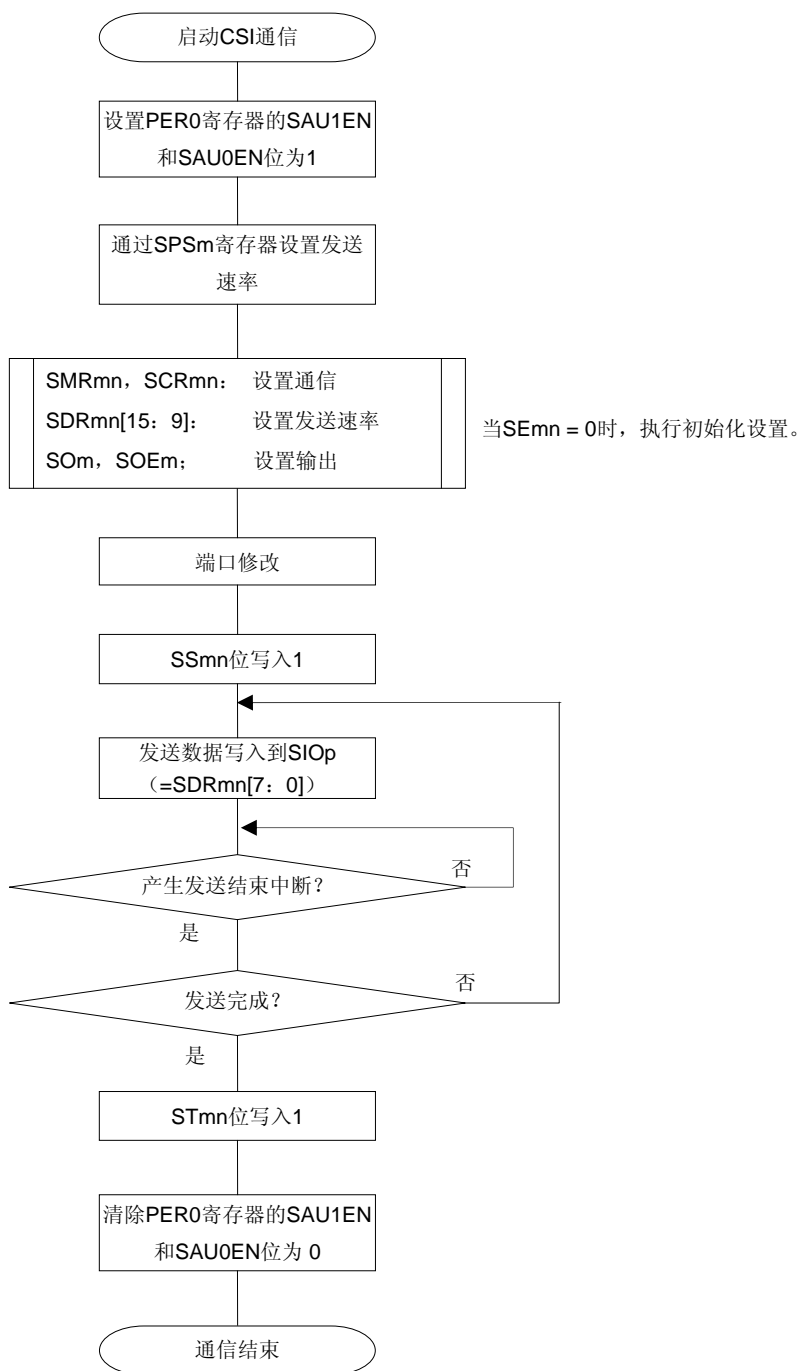
(3) 处理流程 (在单个发送模式下)

图 12-28. 主机发送的时序图 (在单个发送模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

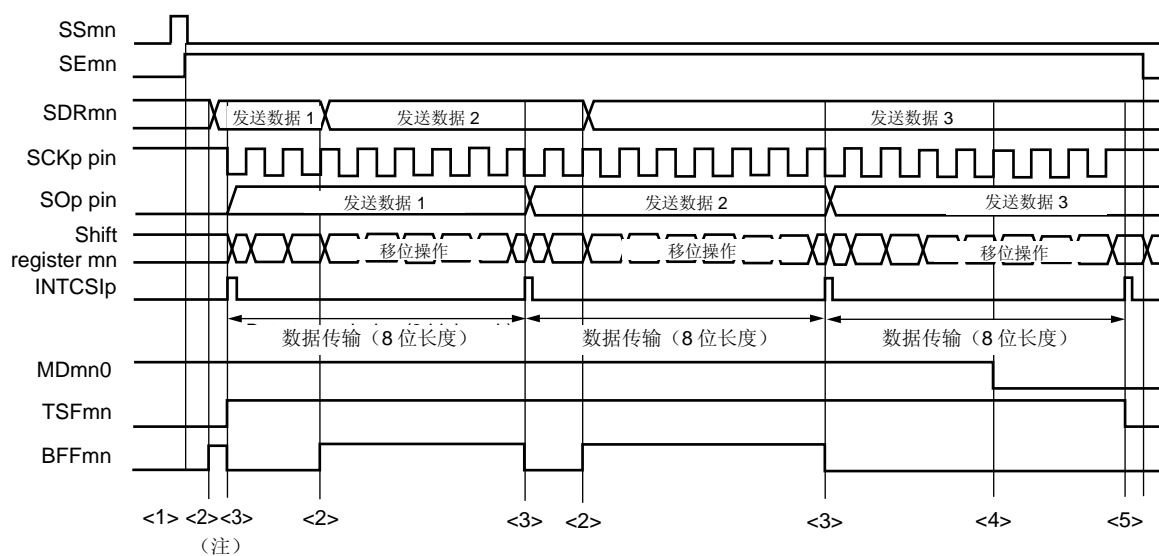
图 12-29. 主机发送的流程图（在单个发送模式下）



注意事项 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

图 12-30. 主机发送的时序图（在连续发送模式下）

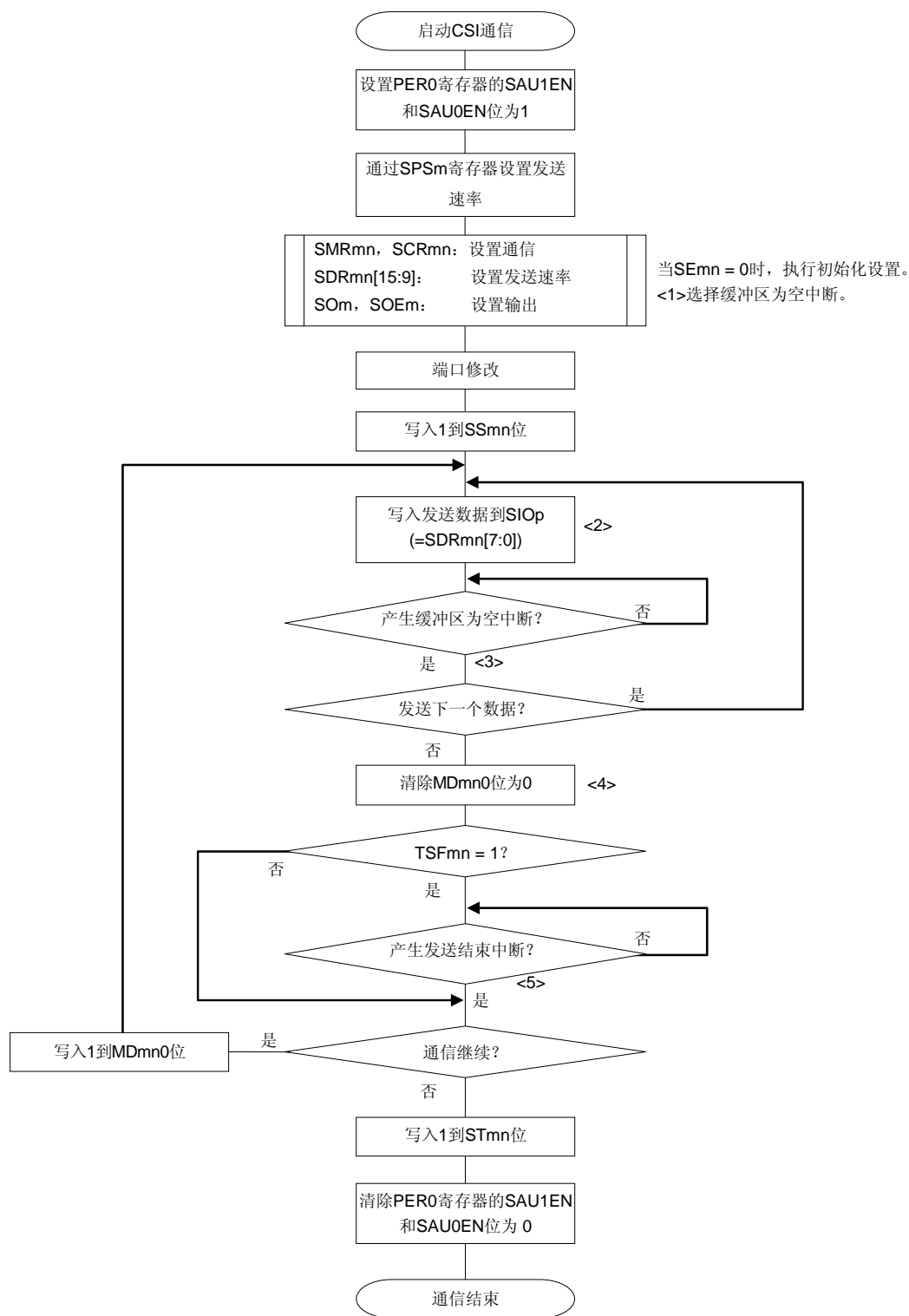


注 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据被覆盖。

注意事项 即使在操作过程中，MDmn0 位也可以被修改。但是必需在最后一位的发送开始之前修改它，这样，它将在最后发送数据的发送结束中断前被修改。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

图 12-31. 主机发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>到<5>对应图 12-30 主机发送的时序图（在连续发送模式下）中的<1>到<5>。

12.5.2 主机接收

主机接收是 78K0R/KF3 输出发送时钟并接收其它设备的数据。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
使用的引脚	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK20, SI20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	只有发送结束中断（禁止缓冲区为空中断。）			
错误检测标志	只有溢出错误检测标志（OVFmn）			
发送数据长度	7或8位			
发送速率	最大 $f_{CLK}/4$ [MHz], 最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] ^注 f_{CLK} : 系统时钟频率			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 从串行时钟开始工作时, 开始输出数据。 • DAPmn = 1: 在串行时钟开始工作的半个时钟前, 开始输出数据。 			
时钟相位e	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB先行			

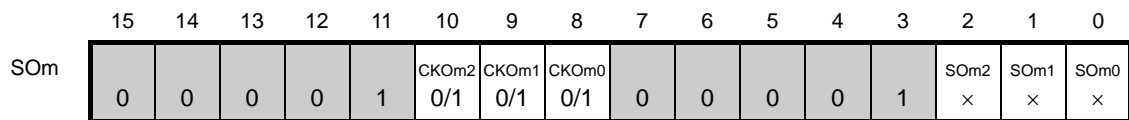
<R>

<R> 注 操作时需要满足以上条件, 且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

(1) 寄存器设置

图 12-32. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 主机接收寄存器内容的举例

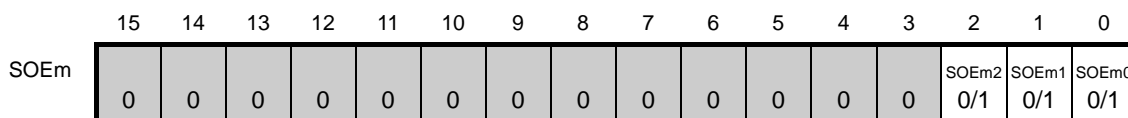
(a) 串行输出寄存器 m (SOm) ...只设置目标通道的对应位。



<R>

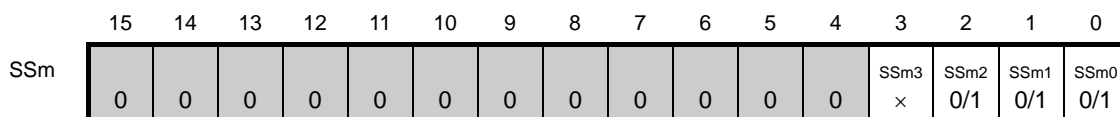
如果数据相位是正向的 (CKPmn = 0)，当这些位为 1 时，通信启动。如果相位是反向的 (CKPmn = 1)，当这些位为 0 时，通信启动。

(b) 串行输出使能寄存器 m (SOEm)

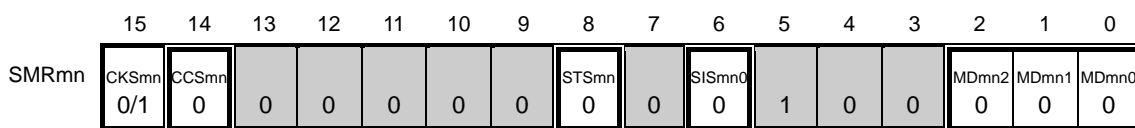


在 CSI 主机接收中使用，设置这些位为 0。

(c) 串行通道启动寄存器 m (SSm) ...只设置目标通道的对应位为 1。

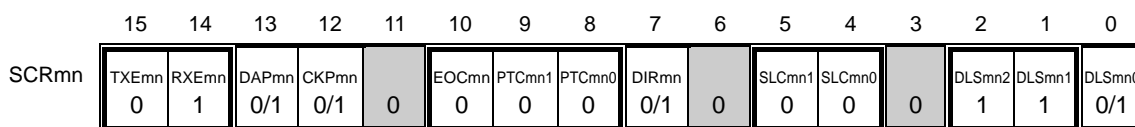


(d) 串行模式寄存器 mn (SMRmn)

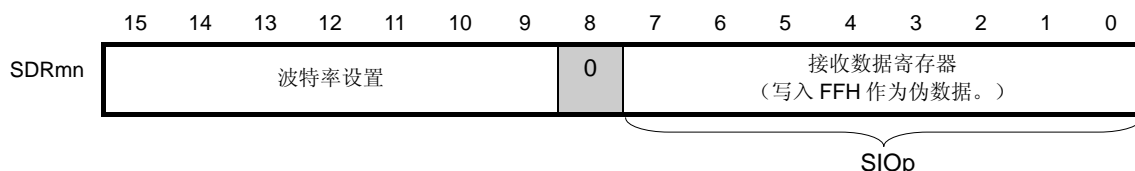


通道 n 的工作模式
0: 发送结束中断

(e) 串行通信操作设置寄存器 mn (SCRmn)



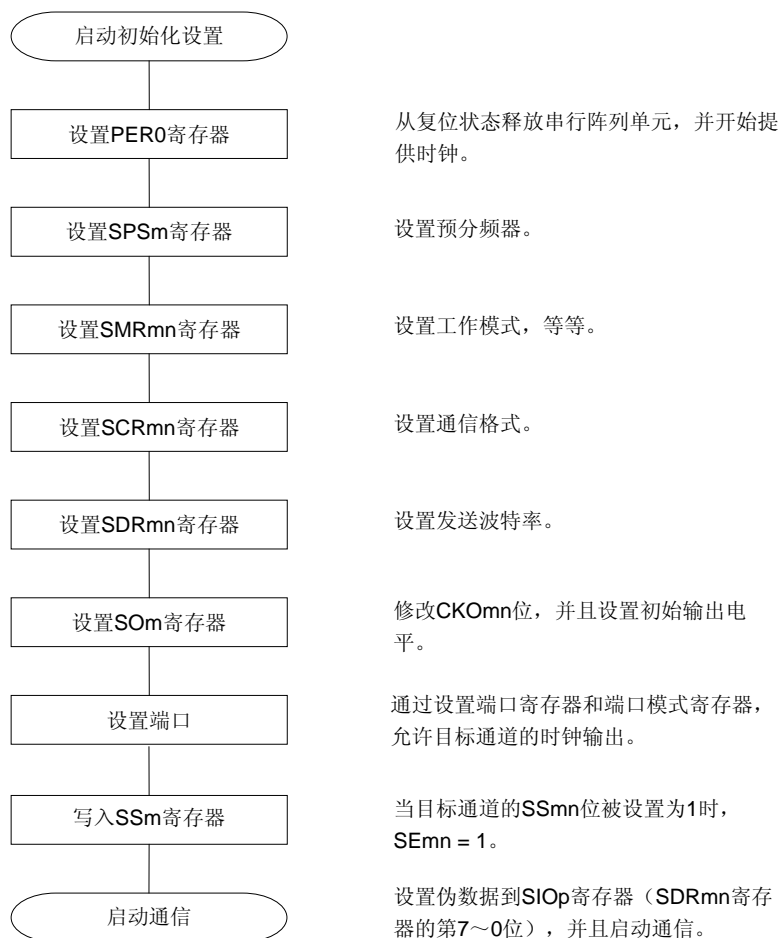
(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)
 □: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

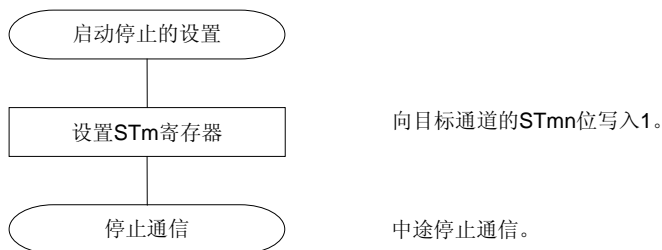
(2) 操作过程

图 12-33. 主机接收的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-34. 停止主机接收的过程



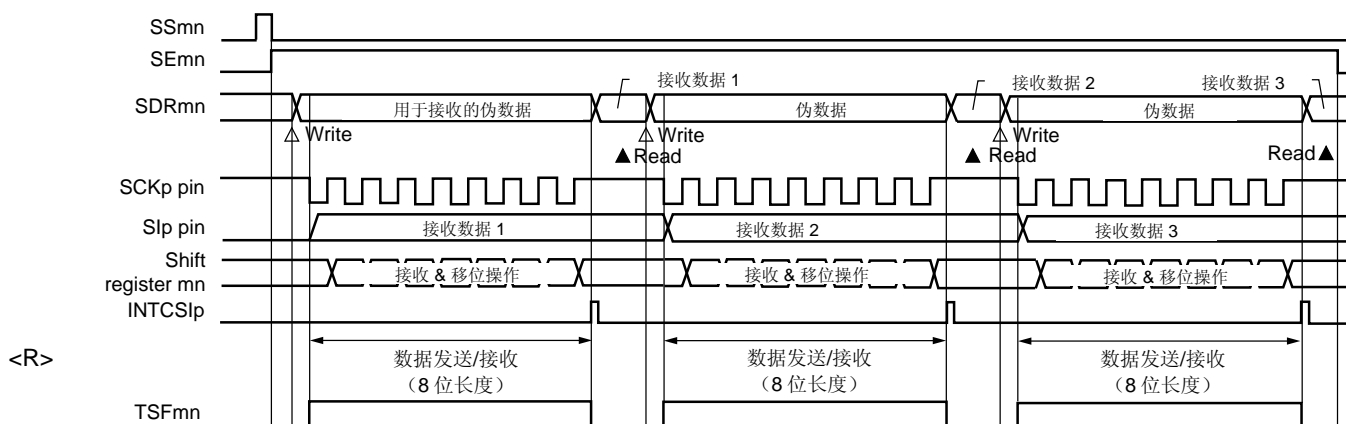
备注 即使在通信停止后，引脚的电平仍然会保持。要恢复操作，重新设置 SOMn 寄存器（见图 12-35 恢复主机接收的过程）。

图 12-35. 恢复主机接收的过程



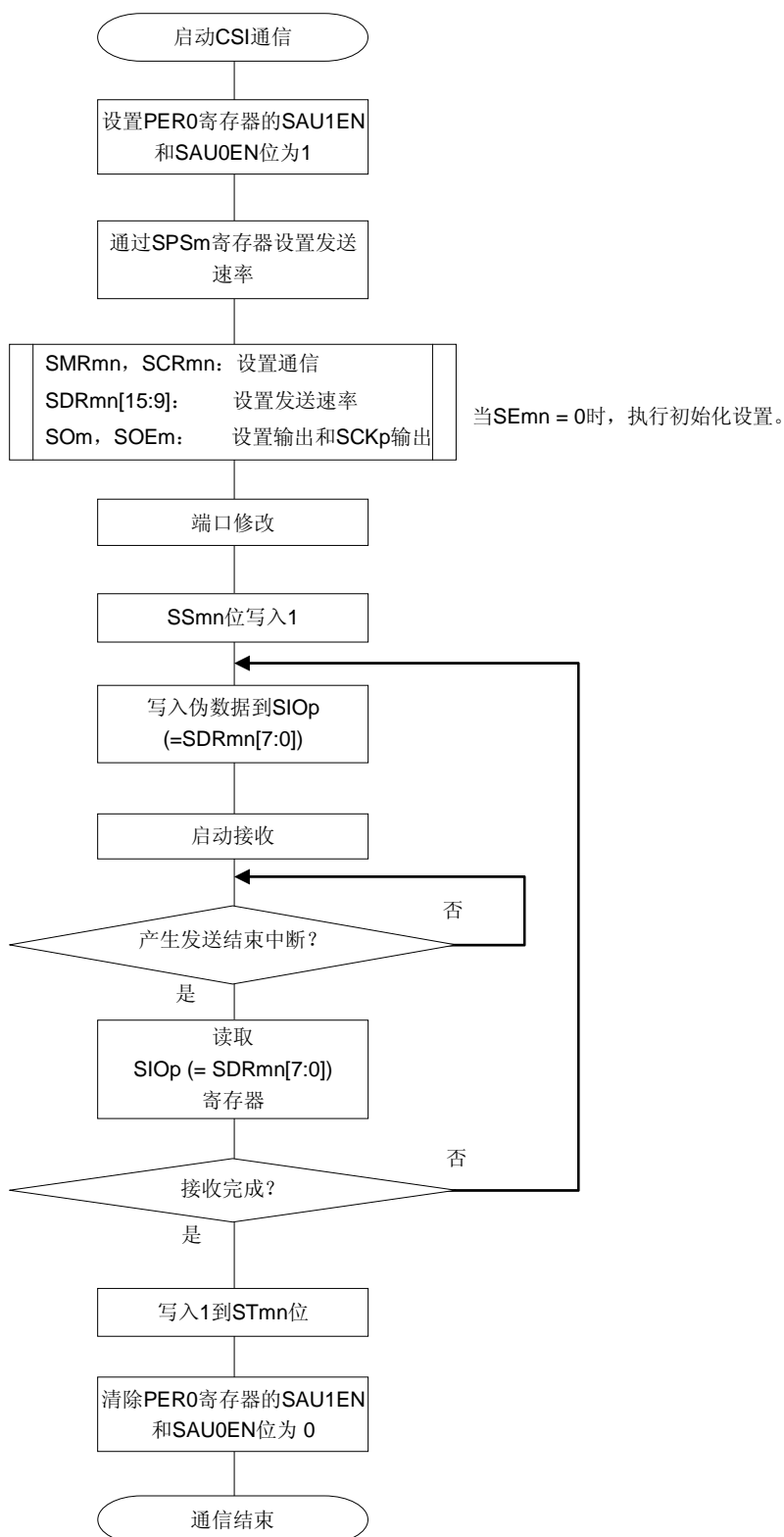
(3) 处理流程 (在单个接收模式下)

图 12-36. 主机接收的时序图 (在单个接收模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

图 12-37. 主机接收的流程图（在单个接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

12.5.3 主机发送/接收

主机发送/接收是由 78K0R/KF3 输出发送时钟，并且发送数据到其它设备，或从其他设备接收数据。

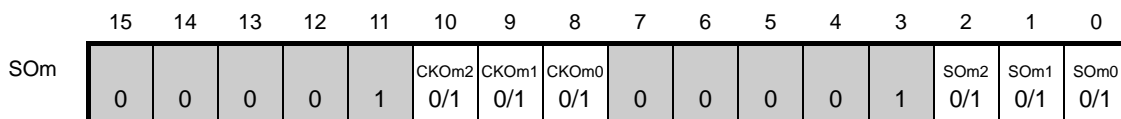
3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK20, SI20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择发送结束中断（在单个发送模式下）或者缓冲区为空中断（在连续发送模式下）。			
错误检测标志	只有溢出错误检测标志（OVFmn）			
发送数据长度	7或8位			
<R> 发送速率	最大 $f_{CLK}/4$ [MHz], 最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] [*] f_{CLK} : 系统时钟频率			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 从串行时钟开始工作时，数据I/O启动。 • DAPmn = 1: 在串行时钟开始工作的半个时钟前，数据I/O启动。 			
时钟相位	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB先行			

<R> 注 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

(1) 寄存器设置

图 12-38. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 主机发送/接收寄存器内容的举例

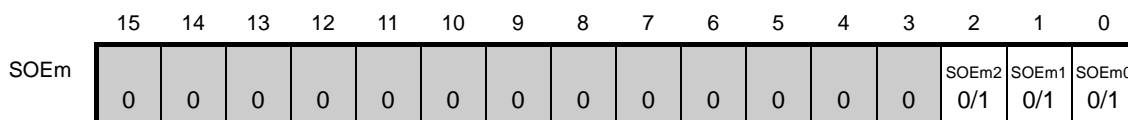
(a) 串行输出寄存器 m (SOm) ...只设置目标通道的对应位。



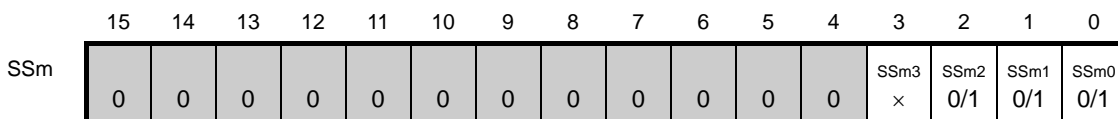
<R>

如果数据相位是正向的 (CKPmn = 0)，当这些位为 1 时，通信启动。如果相位是反向的 (CKPmn = 1)，当这些位为 0 时，通信启动。

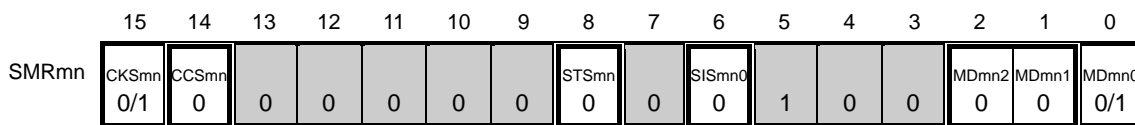
(b) 串行输出使能寄存器 m (SOEm) ... 只设置目标通道的对应位为 1。



(c) 串行通道启动寄存器 m (SSm) ...只设置目标通道的对应位为 1。



(d) 串行模式寄存器 mn (SMRmn)

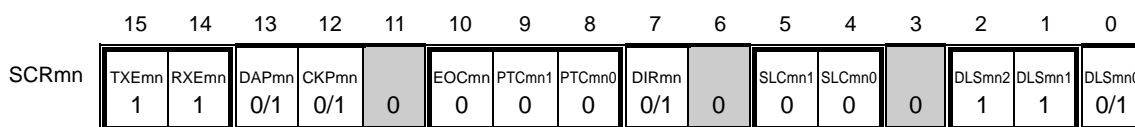


通道 n 的工作模式

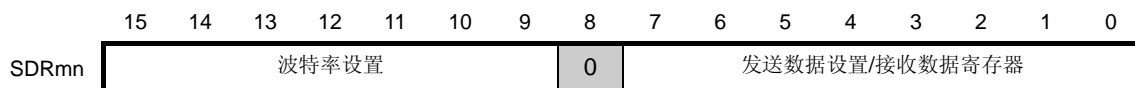
0: 发送结束中断

1: 缓冲区为空中断

(e) 串行通信操作设置寄存器 mn (SCRmn)



(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



SIOp

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

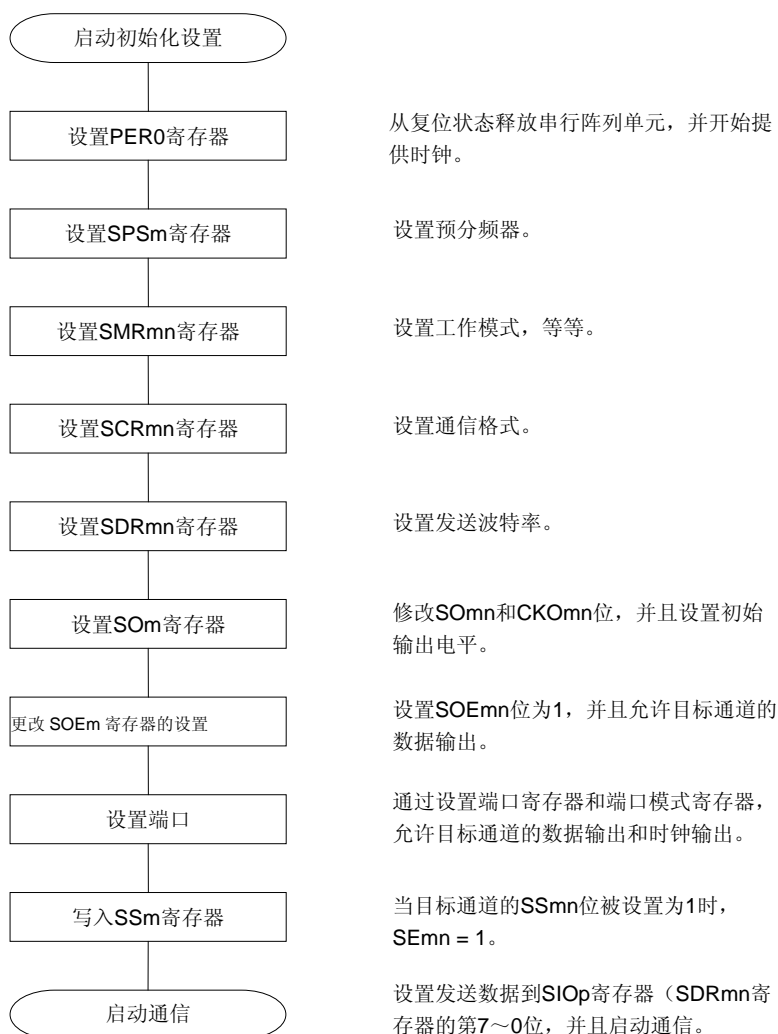
□: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)

x: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)

0/1: 根据用户的使用设置为 0 或 1。

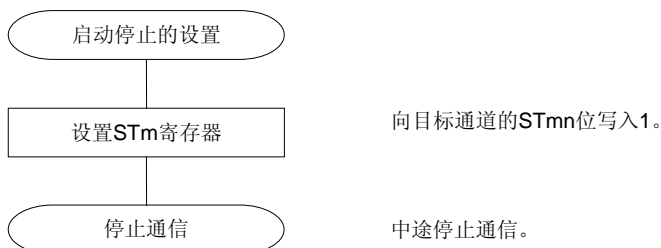
(2) 操作过程

图 12-39. 主机发送/接收的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-40. 停止主机发送/接收的过程



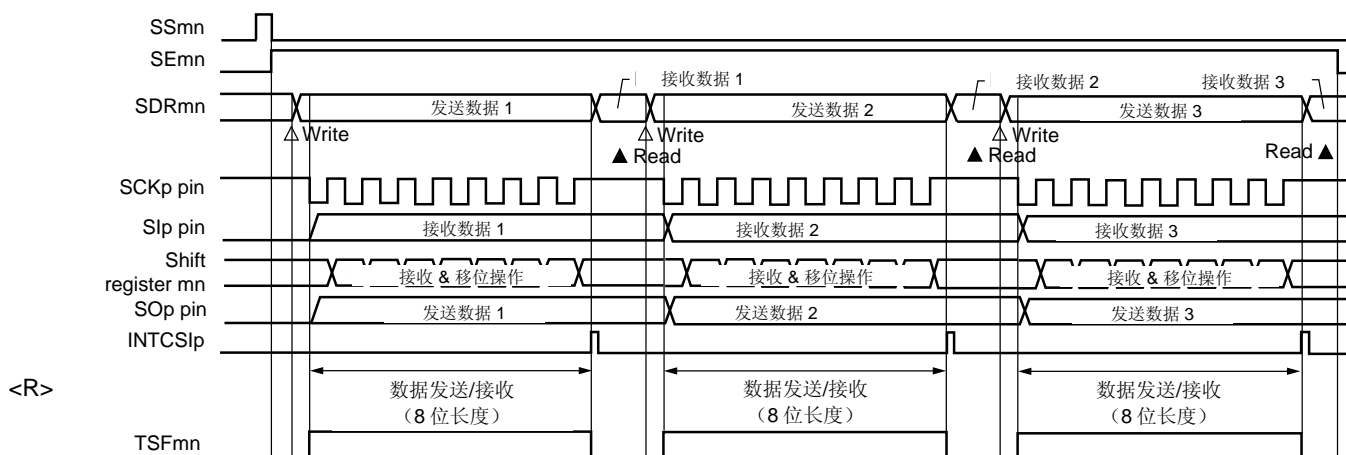
备注 即使在通信停止后，引脚的电平仍然会保持。要恢复操作，重新设置 SOm 寄存器（见图 12-41 恢复主机发送/接收的过程）。

图 12-41. 恢复主机发送/接收的过程



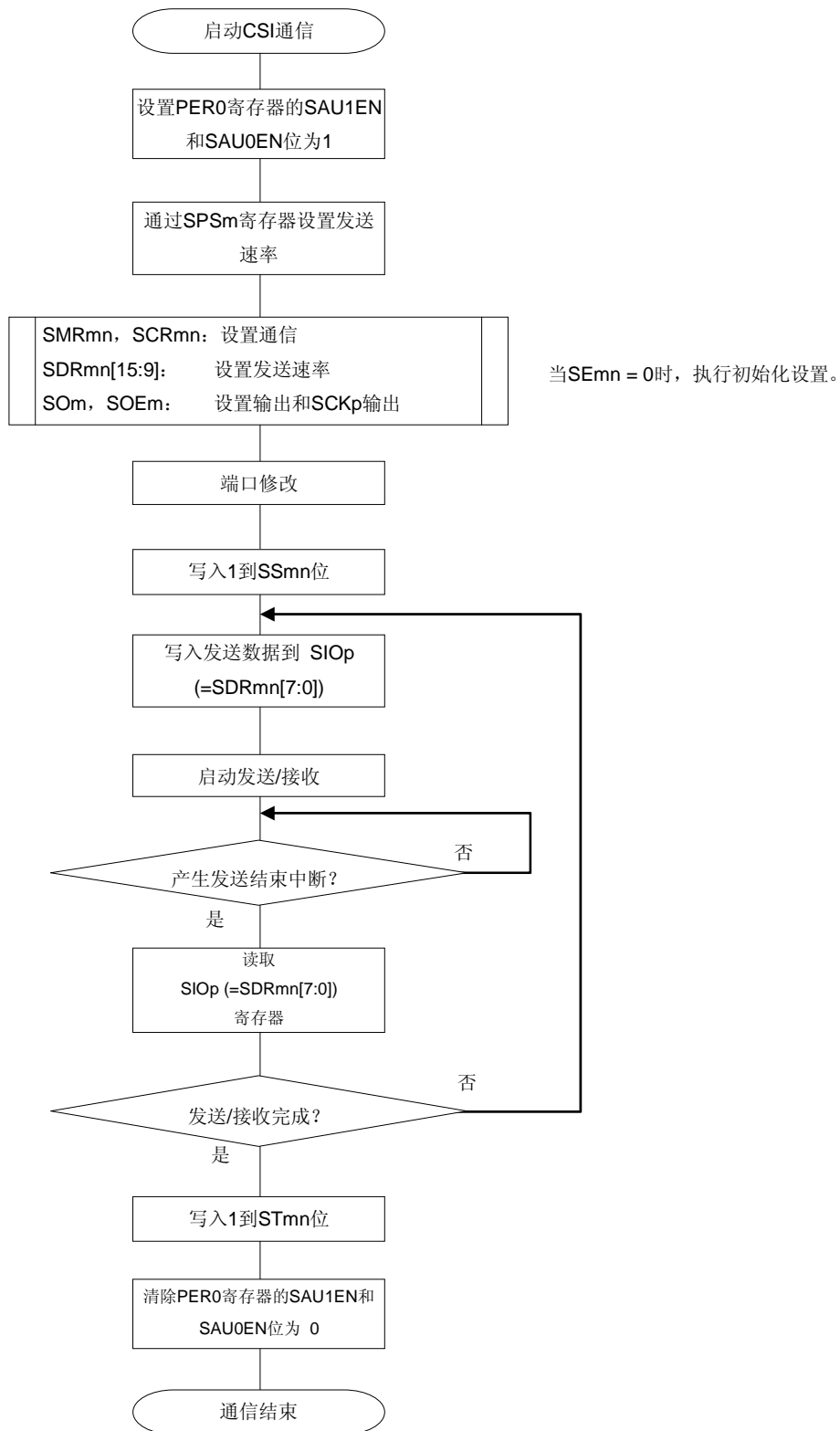
(3) 处理流程 (在单个发送/接收模式下)

图 12-42. 主机发送/接收的时序图 (在单个发送/接收模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

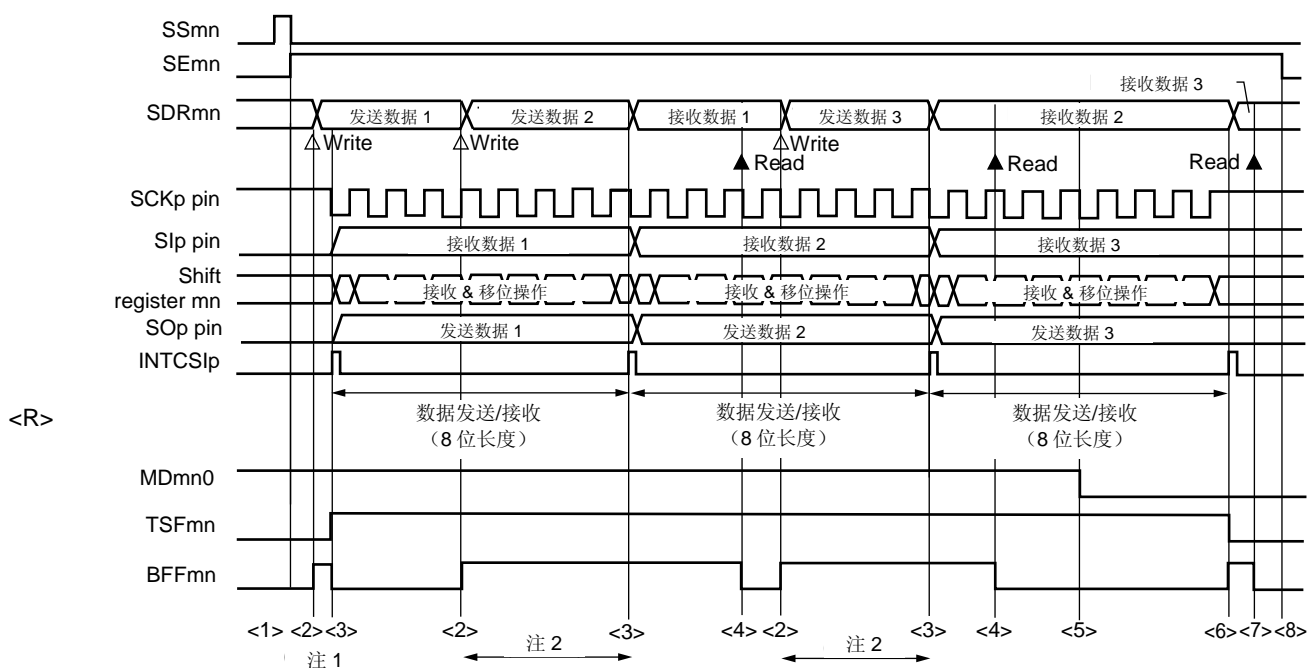
图 12-43. 主机发送/接收的流程图（在单个发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程 (在连续发送/接收模式下)

图 12-44. 主机发送/接收的时序图 (在连续发送/接收模式下)

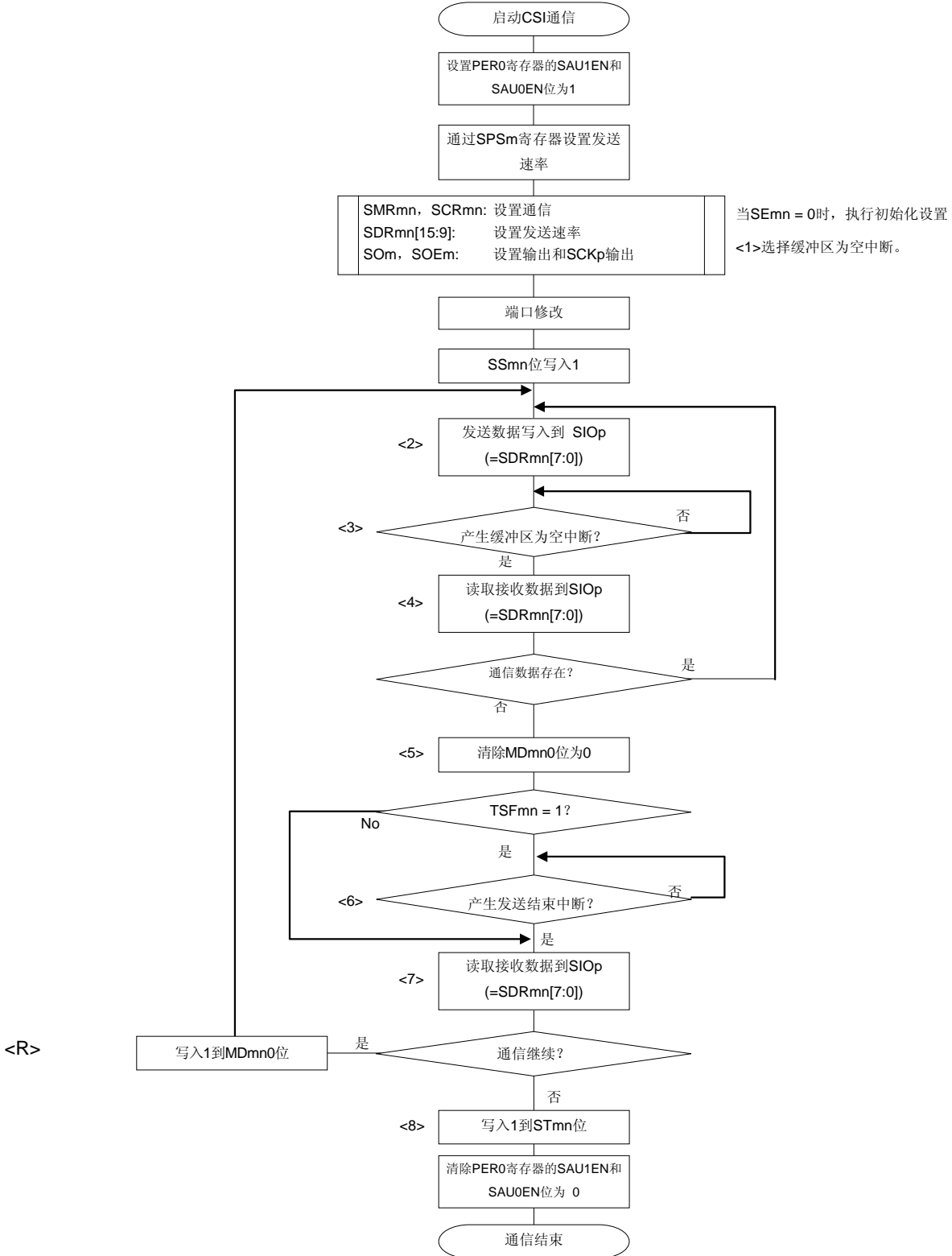


- 注 1. 在 BFFmn = 1 的情况下, 当发送数据写入 SDRmn 寄存器时, 发送数据被覆盖。
 注 2. 在这个周期内, 可以通过读取 SDRmn 寄存器来读出发送数据。此时, 发送操作不受影响。

注意事项 即使在操作过程中, MDmn0 位也可以被修改。
 但是, 在最后一位的发送开始之前修改它, 于是它将在最后发送数据的发送结束中断前被修改。

- 备注** 1. 上图中的 <1> 到 <8> 对应于图 12-45 主机发送/接收的流程图 (在连续发送/接收模式下) 中的 <1> 到 <8>。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

图 12-45. 主机发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。
备注 上图中的<1> 到 <8>对应于图 12-44 主机发送/接收的时序图（在连续发送/接收模式下）中的<1> 到 <8>。

12.5.4 从机发送

从机发送是 78K0R/KF3 发送数据到另一个设备，发送时钟由通讯对象输入。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择发送结束中断（在单个发送模式下）或者缓冲区为空中断（在连续发送模式下）。			
错误检测标志	只有溢出错误检测标志（OVFmn）			
发送数据长度	7或8位			
<R> 发送速率	f _{CLK} /6 [MHz]和f _{MCK} /2 [MHz]中较小的一个作为最大发送速率 ^{注1, 2} 。			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 从串行时钟开始工作时，开始输出数据。 • DAPmn = 1: 在串行时钟开始工作的半个时钟前，开始输出数据。 			
时钟相位	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB 先行			

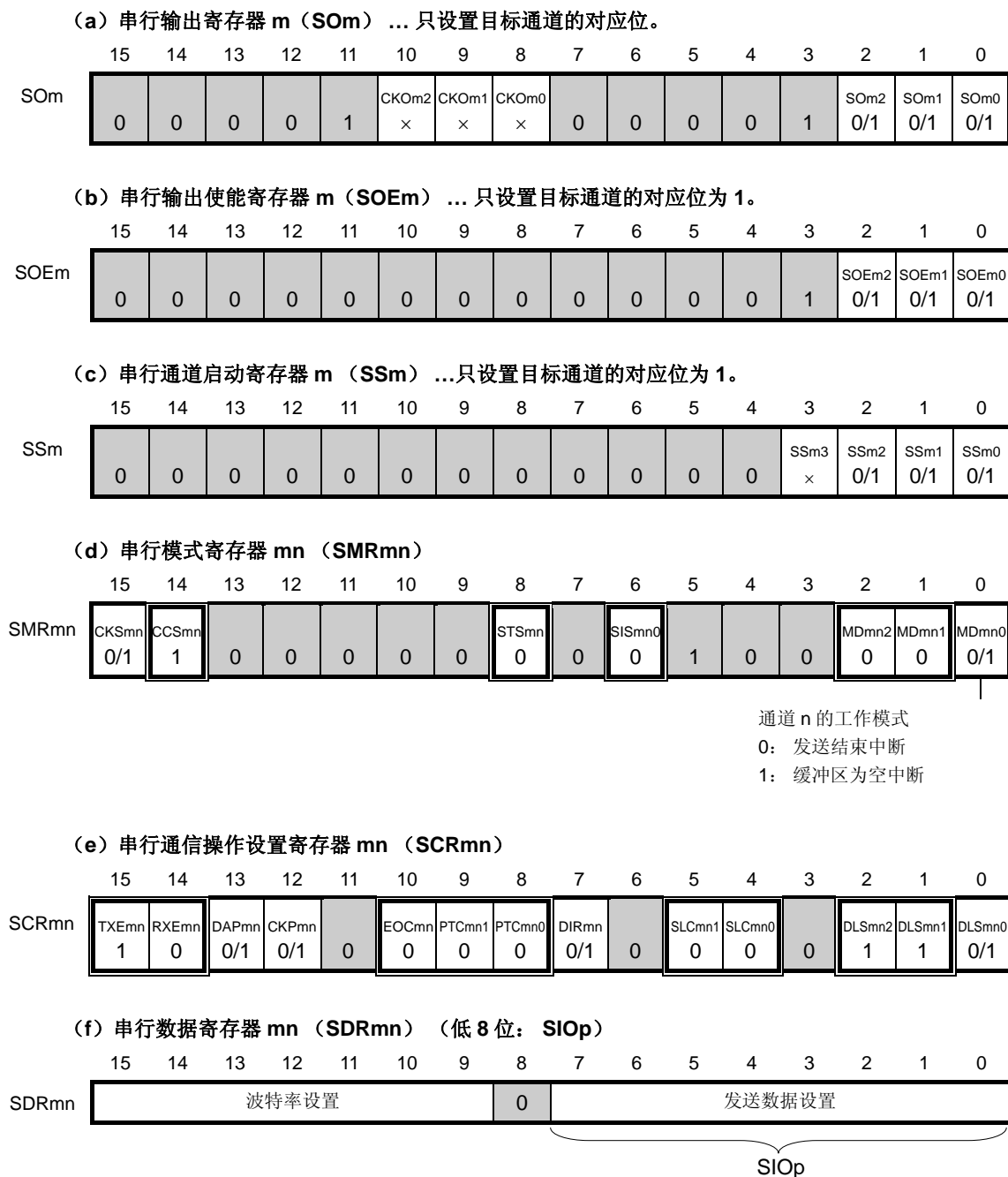
<R> 注 1. 因为输入到引脚 SCK00、SCK01、SCK10 和 SCK20 的外部串行时钟被内部采样后使用，最快波特率就是 f_{CLK}/6 [MHz]和 f_{MCK}/2 [MHz]之中较小的那个。

<R> 2. 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK}: 目标通道的操作时钟（MCK）频率
f_{CLK}: 系统时钟频率

(1) 寄存器设置

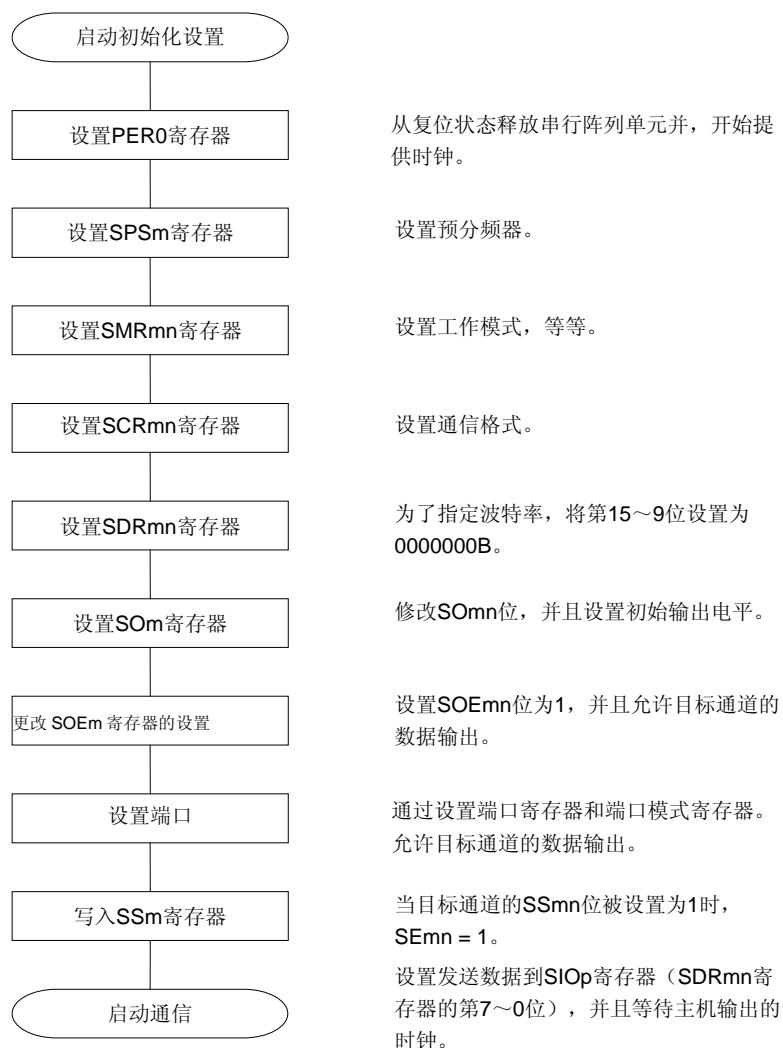
图 12-46. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 从机发送寄存器内容的举例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)
 □: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

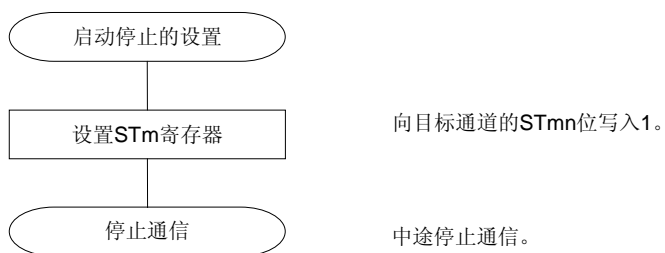
(2) 操作过程

图 12-47. 从机发送的初始化设置过程



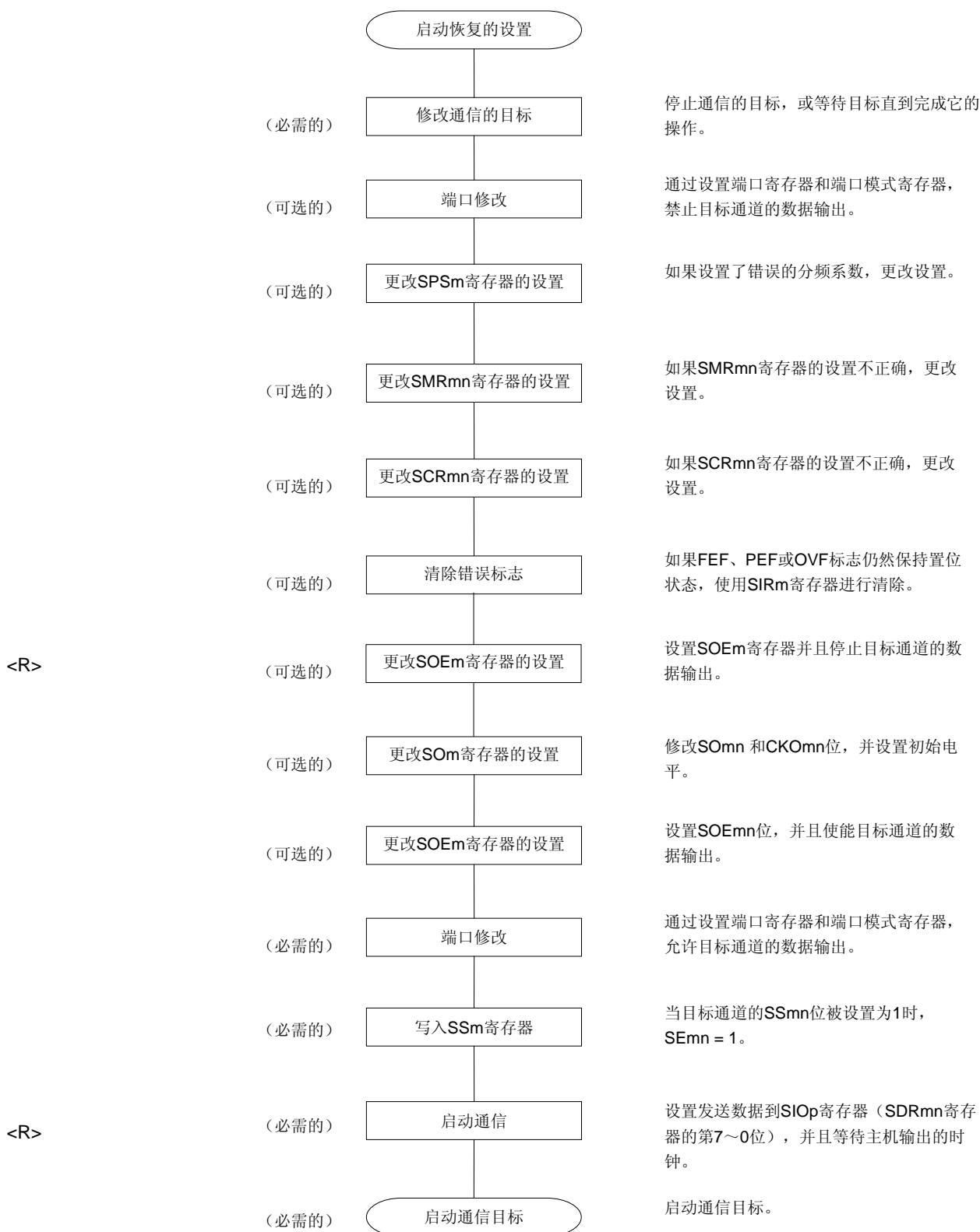
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-48. 停止从机发送的过程



备注 即使在通信停止后，引脚的电平仍然会保持。要恢复操作，重新设置 S0m 寄存器（见图 12-49 恢复从机发送的过程）。

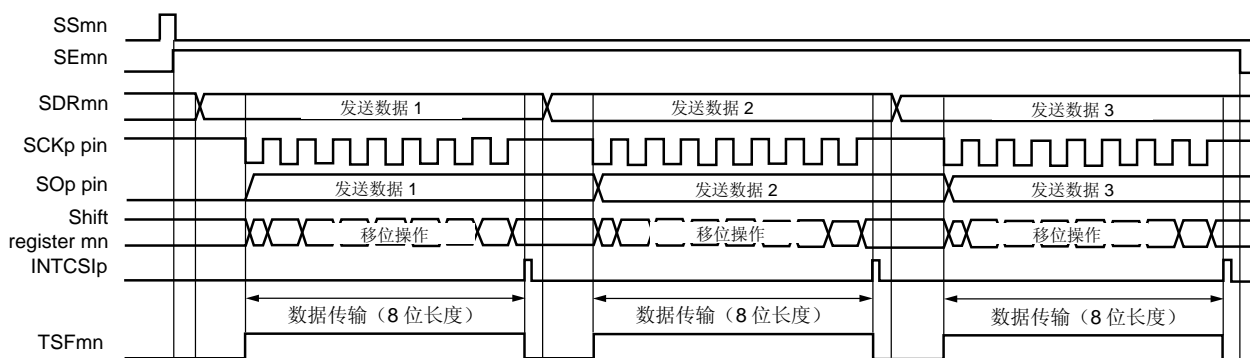
图 12-49. 恢复从机发送的过程



(3) 处理流程 (在单个发送模式下)

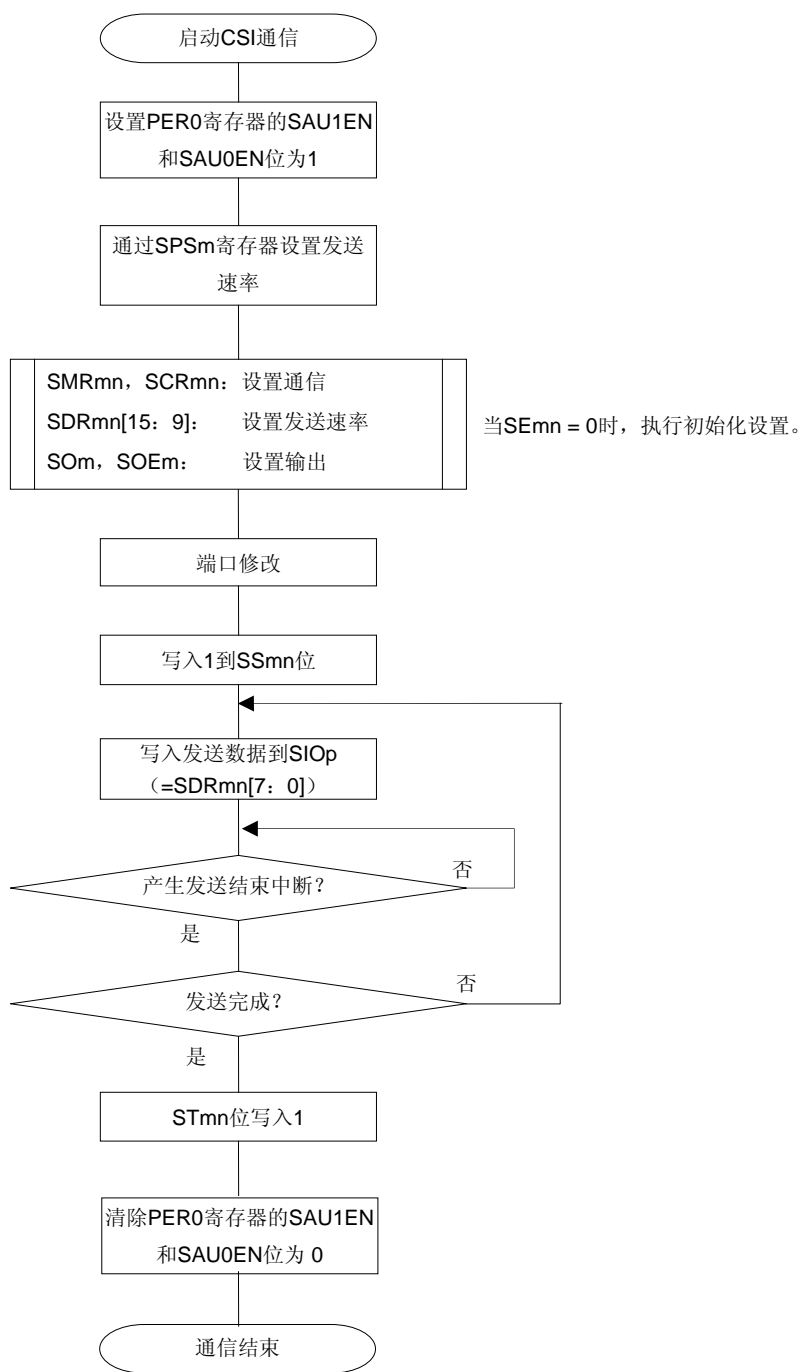
<R>

图 12-50. 从机发送的时序图 (在单个发送模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

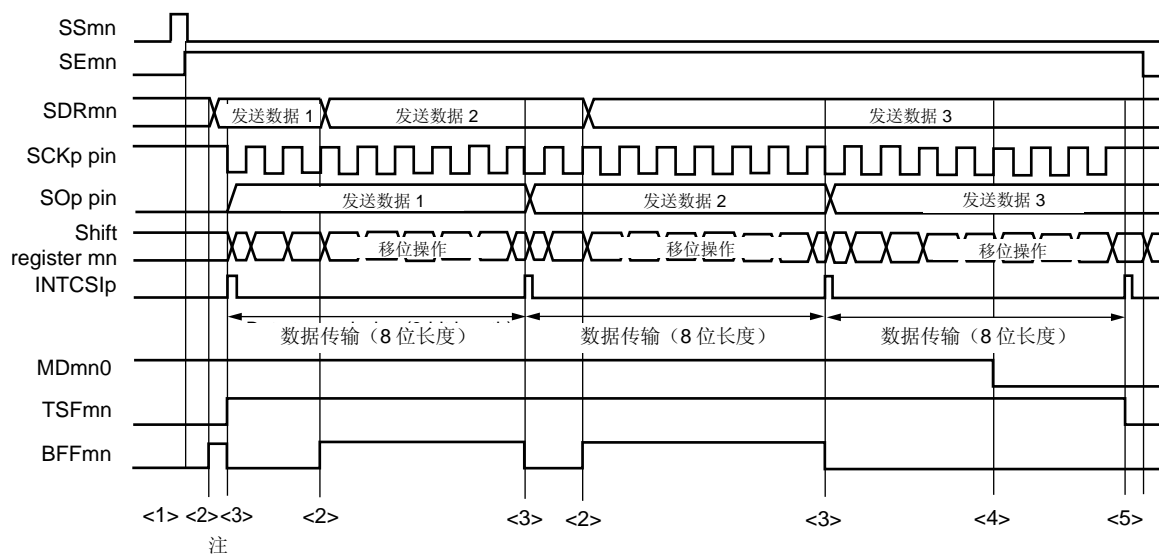
图 12-51. 从机发送的流程图（在单个发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

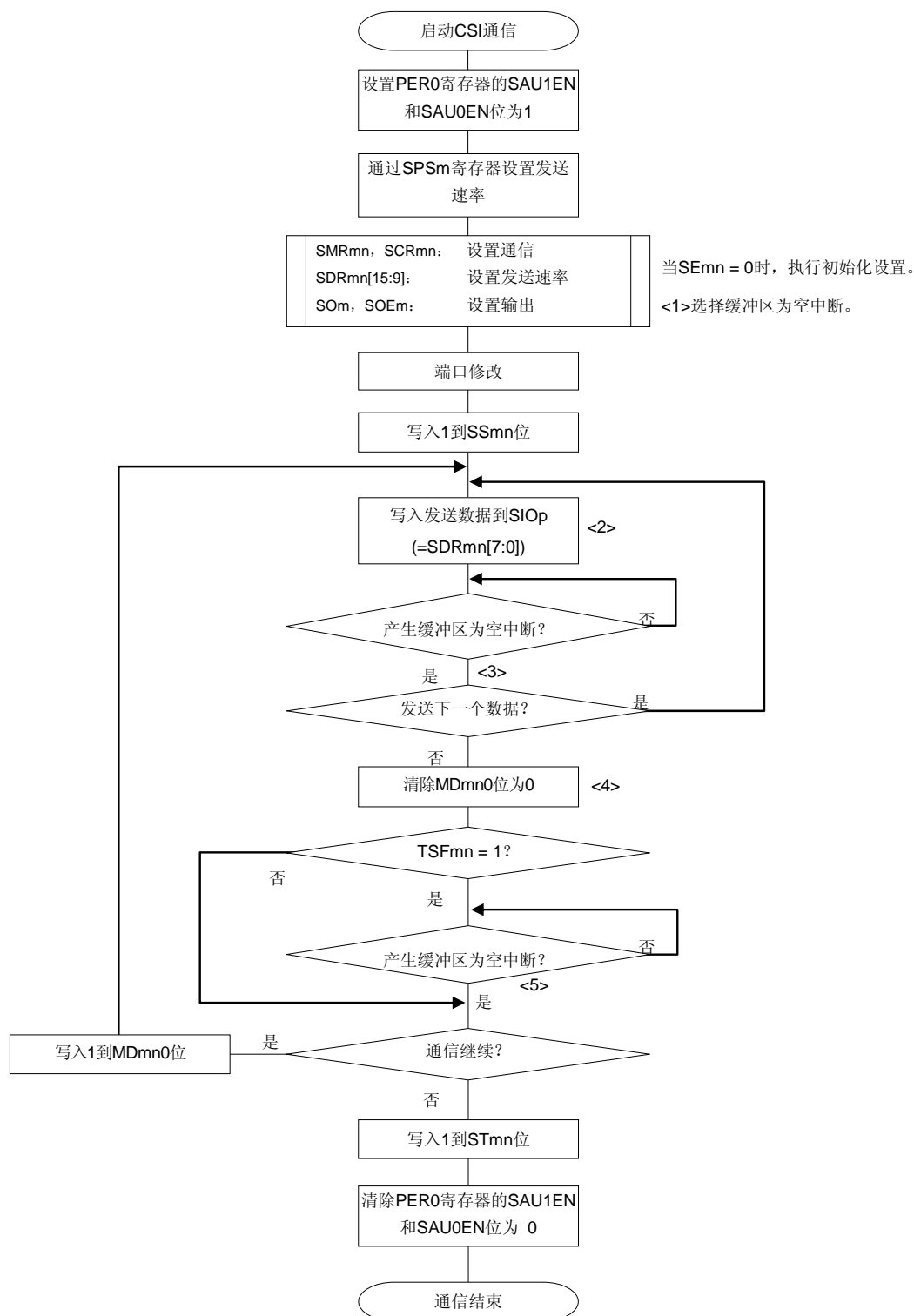
图 12-52. 从机发送的时序图（在连续发送模式下）



注 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据被覆盖。

注意事项 即使在操作过程中，MDmn0 位可以被修改。但是，请在最后一位的发送开始之前修改它

图 12-53. 从机发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>到<5>对应图 12-52 从机发送的时序图（在连续发送模式下）中的<1>到<5>。

12.5.5 从机接收

从机接收是 78K0R/KF3 接收另一个设备的数据，发送时钟由通讯对象输入。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK20, SI20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	只有发送结束中断（禁止设置缓冲区为空中断。）			
错误检测标志	只有溢出错误检测标志（OVFmn）			
发送数据长度	7或8位			
发送速率	f _{CLK} /6 [MHz]和f _{MCK} /2 [MHz]中较小的一个作为最大发送速率 ^{#1,2} 。			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 数据输入从串行时钟开始工作时开始。 • DAPmn = 1: 数据输入在串行时钟开始工作前半个时钟开始。 			
时钟相位	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB先行			

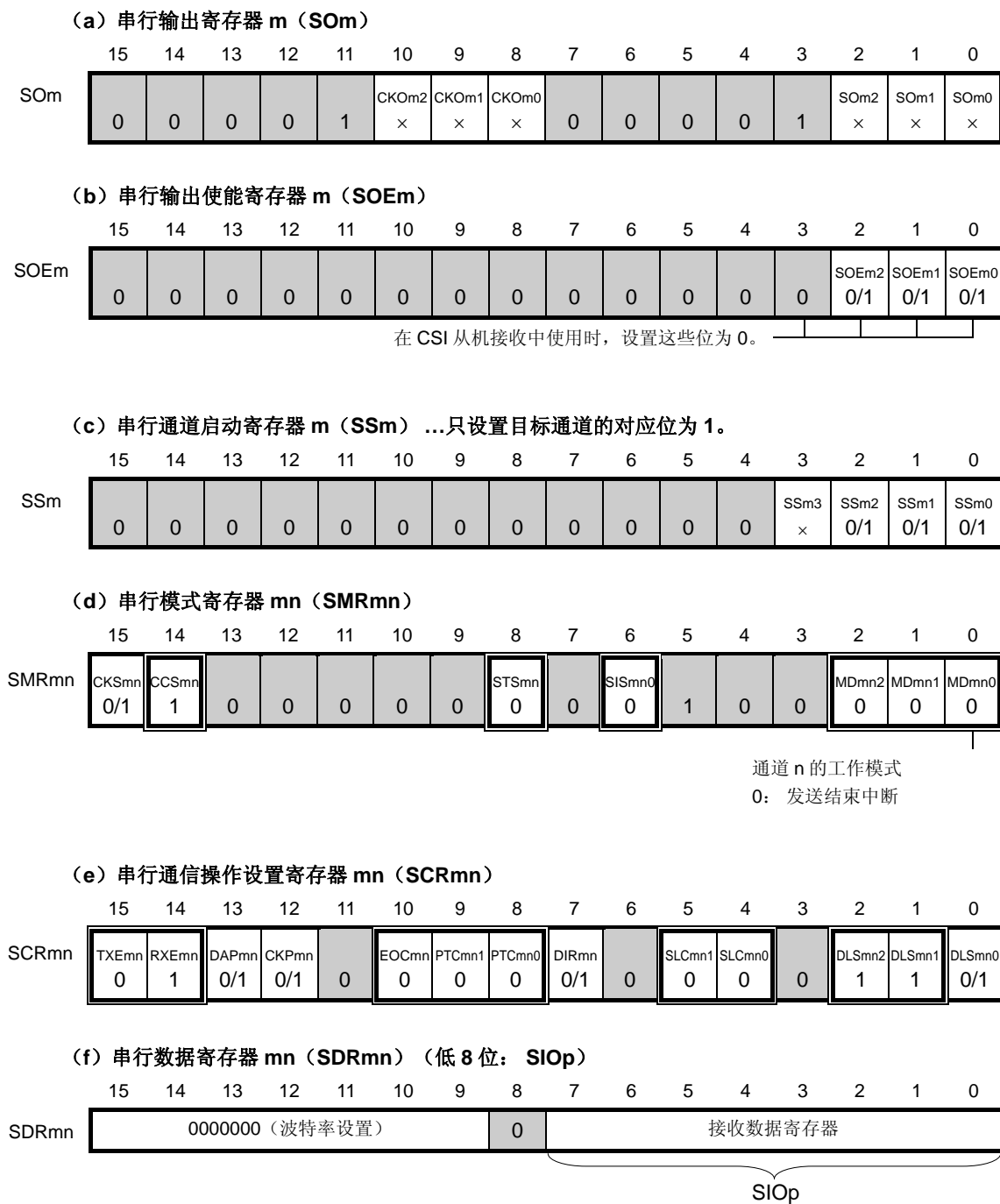
<R> 注 1. 因为输入到引脚 SCK00、SCK01、SCK10 和 SCK20 的外部串行时钟被内部采样后使用，最快波特率就是 f_{CLK}/6 [MHz]和 f_{MCK}/2 [MHz]之中较小的那个。

<R> 2. 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK}: 目标通道的操作时钟（MCK）频率
 f_{CLK}: 系统时钟频率

(1) 寄存器设置

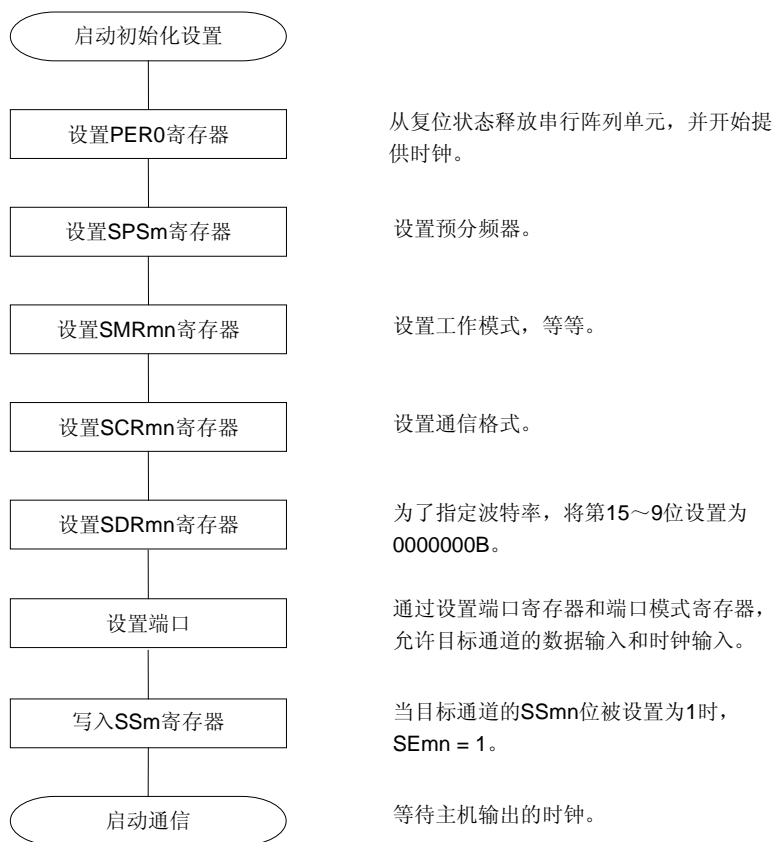
图 12-54. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 从机接收寄存器内容的举例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)
 □: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

(2) 操作过程

图 12-55. 主接收的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-56. 停止从机接收的过程

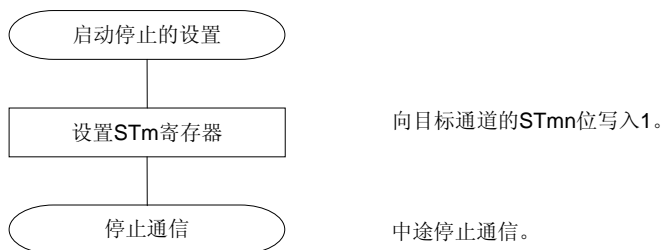


图 12-57. 恢复从机接收的过程

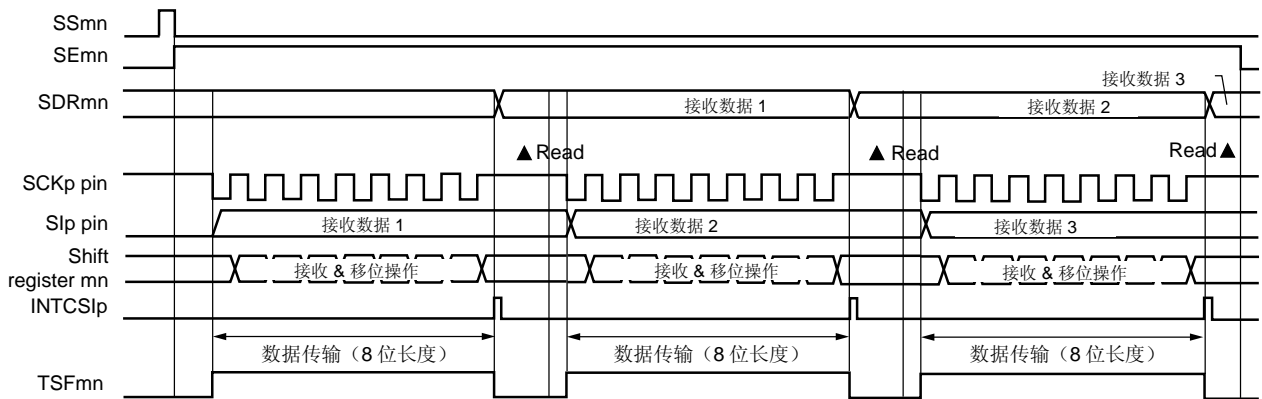


<R>

(3) 处理流程 (在单个接收模式下)

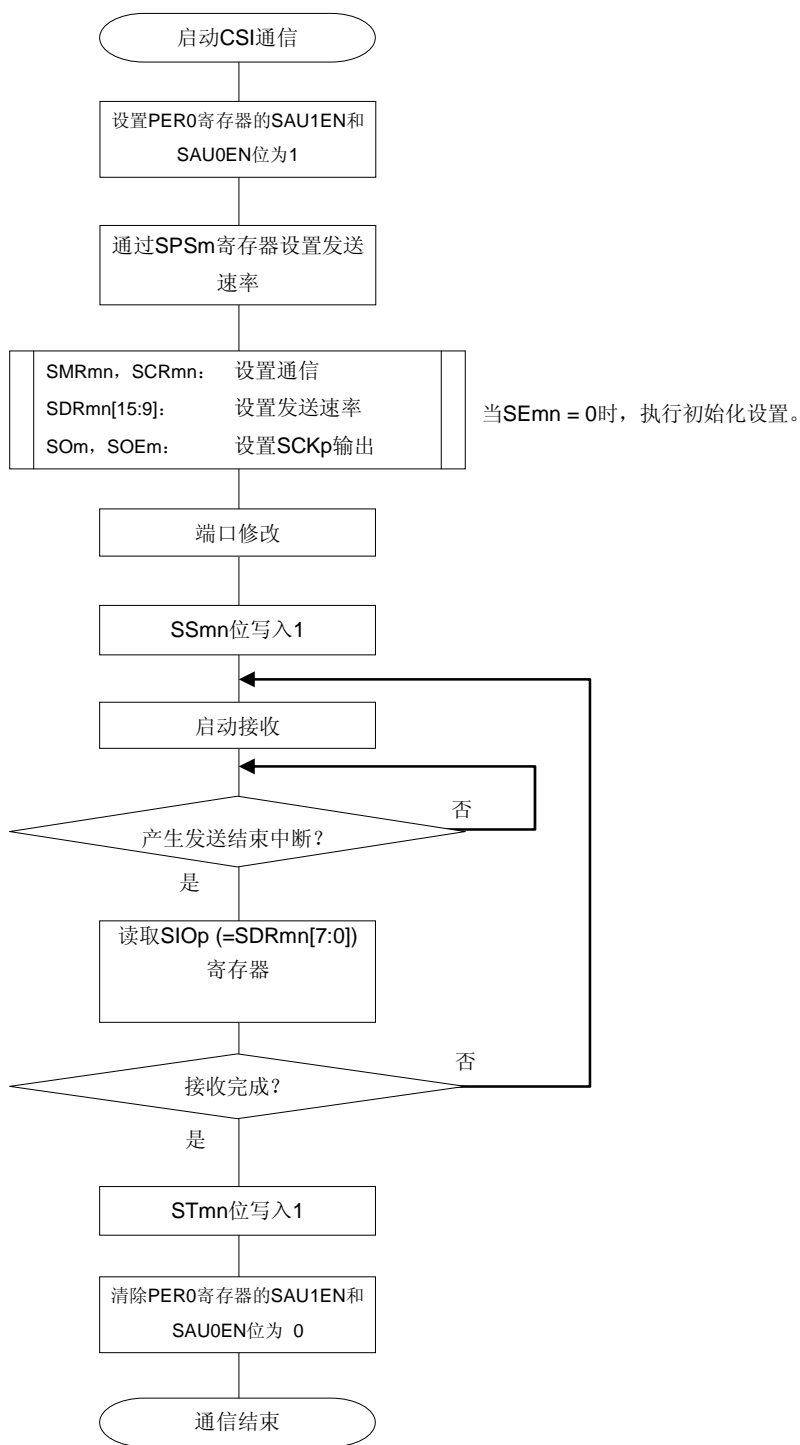
<R>

图 12-58. 从机接收的时序图 (在单个接收模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

图 12-59. 从机接收的流程图（在单个接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

12.5.6 从机发送/接收

从机发送是 78K0R/KF3 发送数据到另一个设备或从另一个设备接收数据，发送时钟由通讯对象输入。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用的引脚	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK20, SI20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择发送结束中断（在单个发送模式下）或者缓冲区为空中断（在连续发送模式下）。			
错误检测标志	只有溢出错误检测标志（OVFmn）			
发送数据长度	7或8位			
发送速率	f _{CLK} /6 [MHz]和f _{MCK} /2 [MHz]中较小的一个作为最大发送速率 ^{#1, 2} 。			
数据相位	通过DAPmn位选择 <ul style="list-style-type: none"> • DAPmn = 0: 从串行时钟开始工作时，数据I/O启动。 • DAPmn = 1: 在串行时钟开始工作的半个时钟前，数据I/O启动。 			
时钟相位	通过CKPmn位选择 <ul style="list-style-type: none"> • CKPmn = 0: 正向 • CKPmn = 1: 反向 			
数据方向	MSB 或 LSB 先行			

<R>

- 注 1. 因为输入到引脚 SCK00、SCK01、SCK10 和 SCK20 的外部串行时钟被内部采样后使用，最快波特率是 f_{CLK}/6 [MHz]和 f_{MCK}/2 [MHz]之中较小的那个。
- 注 2. 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

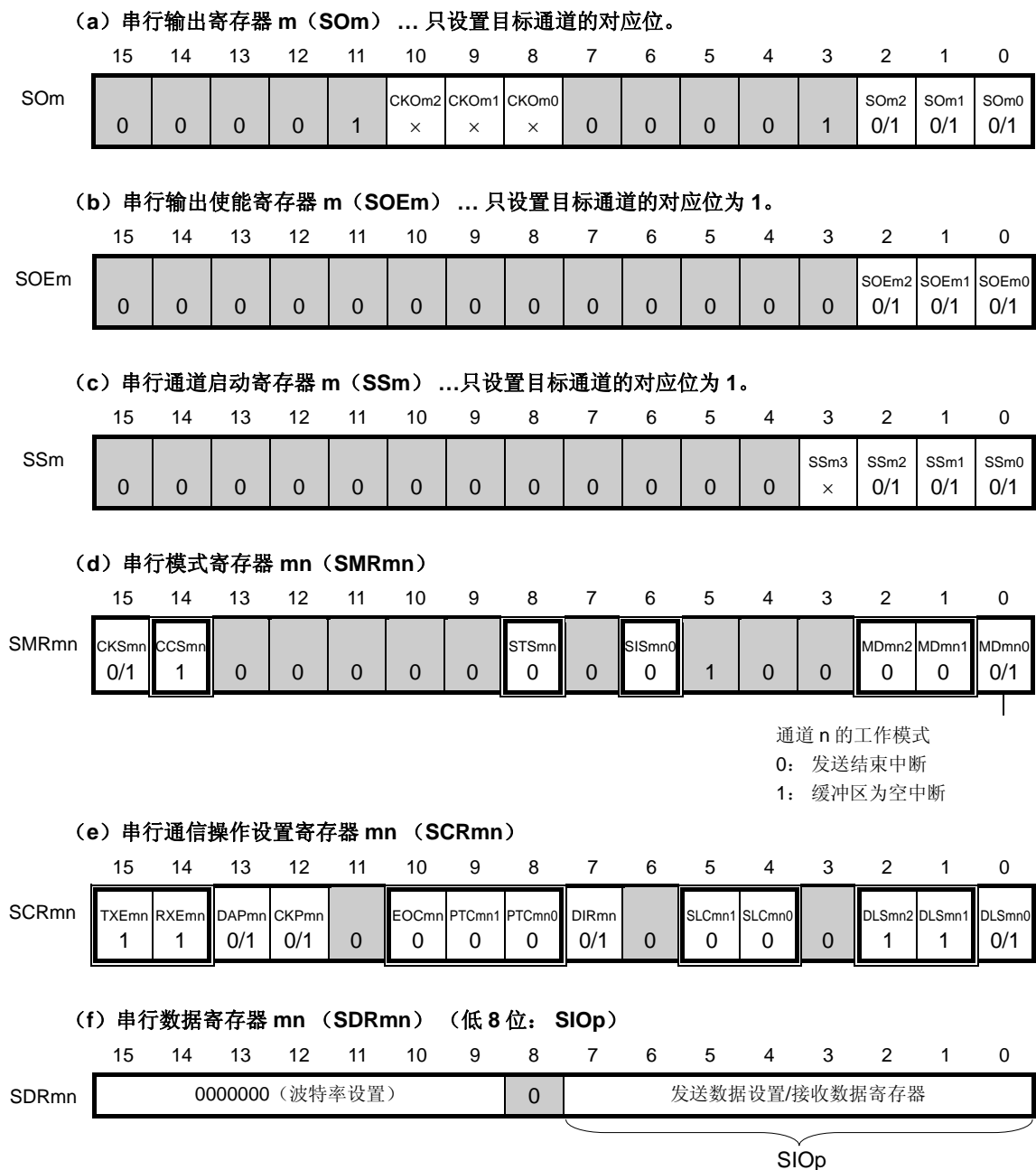
<R>

<R>

备注 f_{MCK}: 目标通道的操作时钟（MCK）频率
 f_{CLK}: 系统时钟频率

(1) 寄存器设置

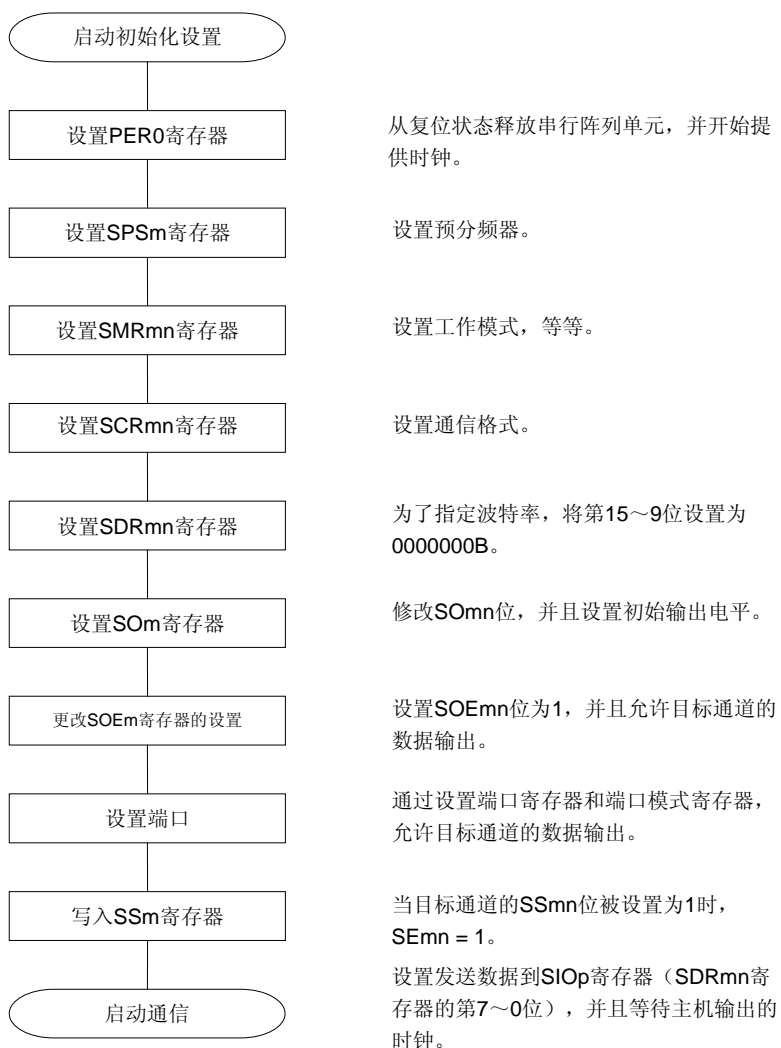
图 12-60. 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 主机发送/接收寄存器内容的举例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)
 □: 在 CSI 主机发送模式下设置固定, ■: 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

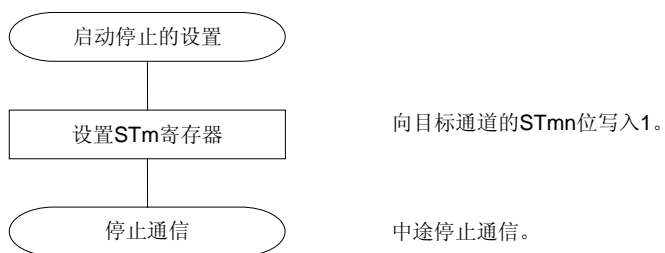
(2) 操作过程

图 12-61. 主机发送/接收的初始化设置过程



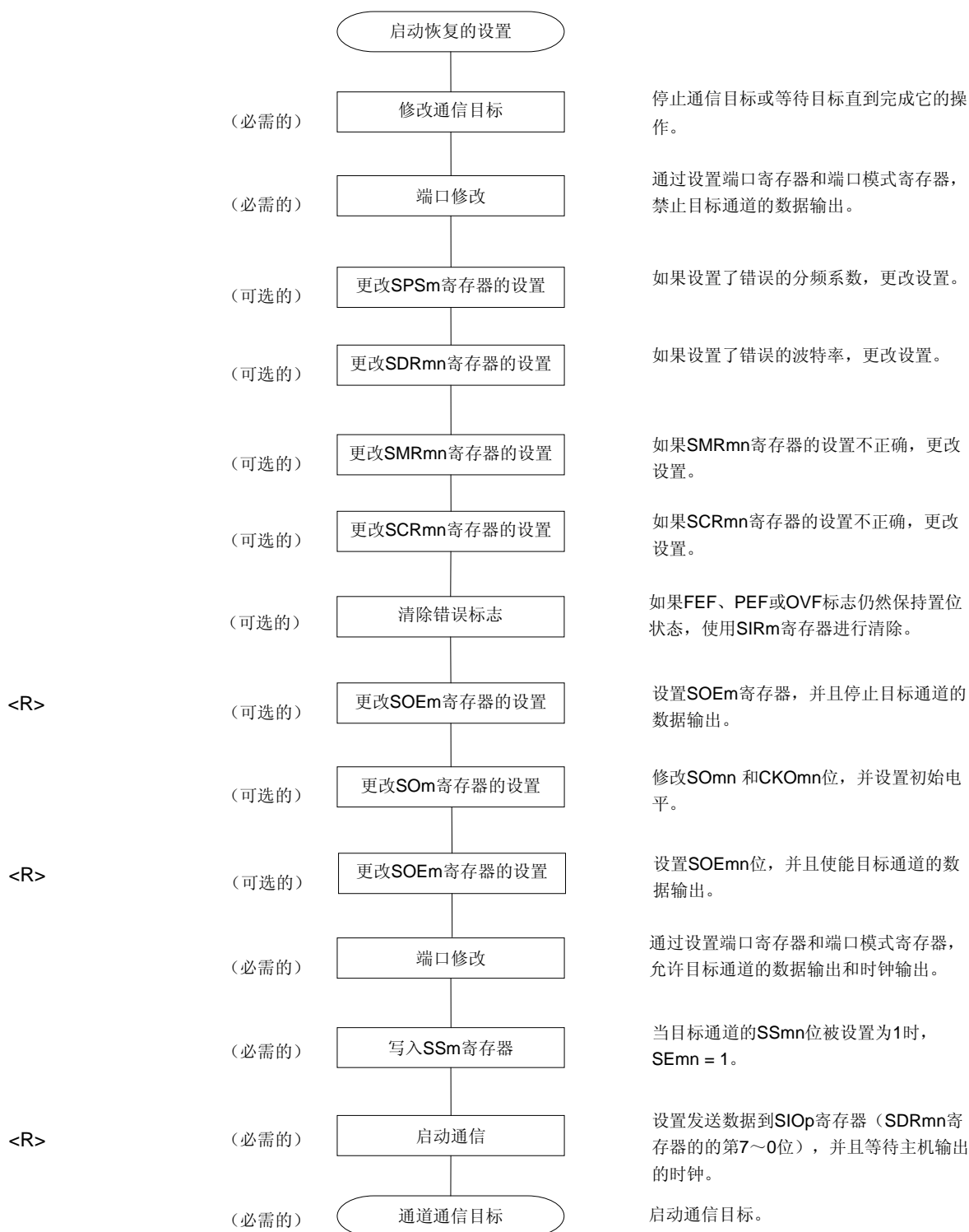
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-62. 停止主机发送/接收的过程



备注 即使在通信停止后，引脚的电平仍然会保持。要恢复操作，重新设置 S0m 寄存器（见图 12-63 恢复从机发送/接收的过程）。

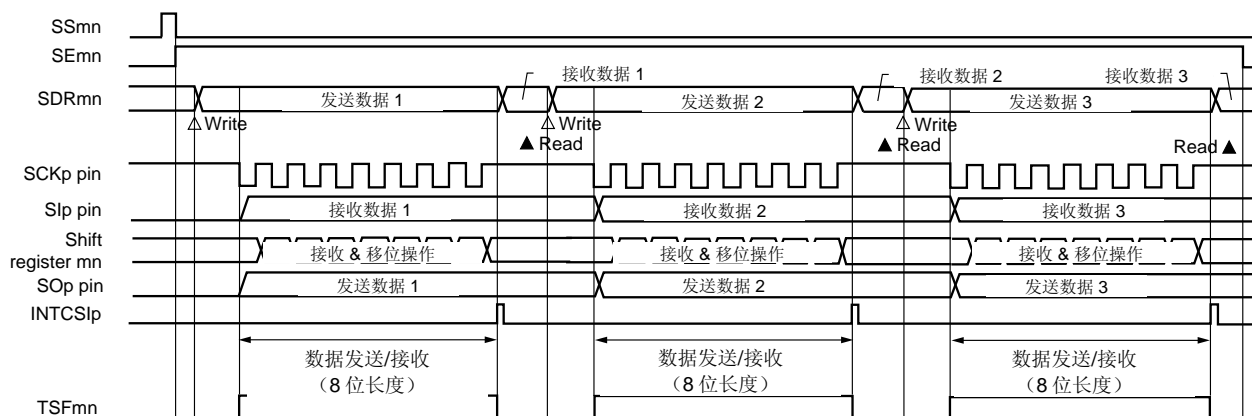
图 12-63. 恢复从机发送/接收的过程



(3) 处理流程 (在单个发送/接收模式下)

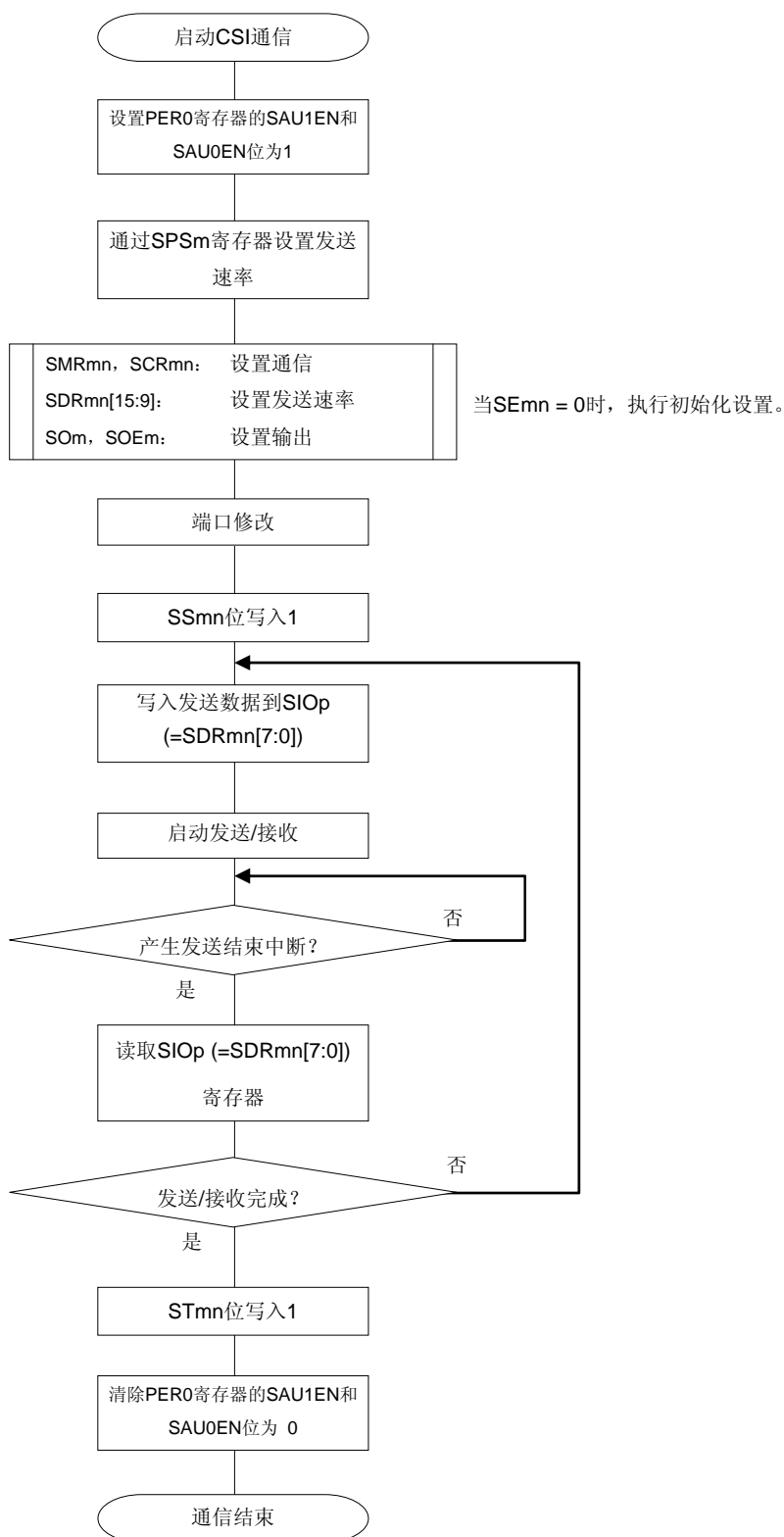
<R>

图 12-64. 从机发送/接收的时序图 (在单个发送/接收模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

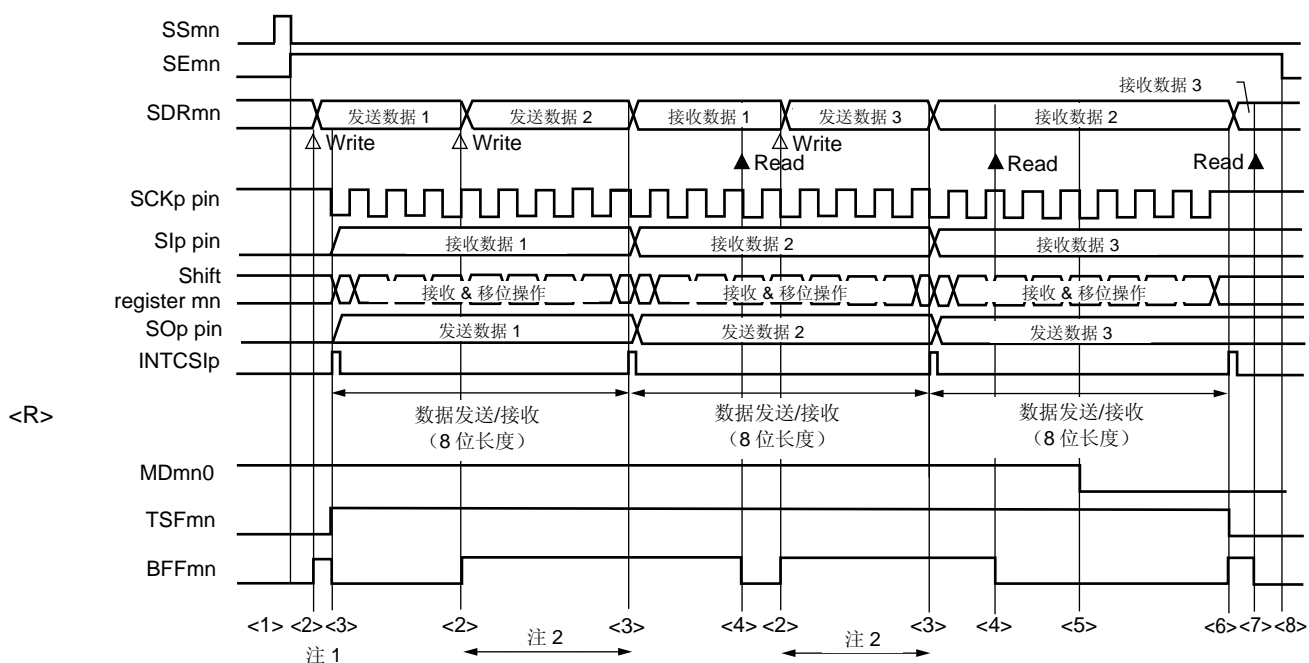
图 12-65. 从机发送/接收的流程图（在单个发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送/接收模式下）

图 12-66. 从机发送/接收的时序图（在连续发送/接收模式下）

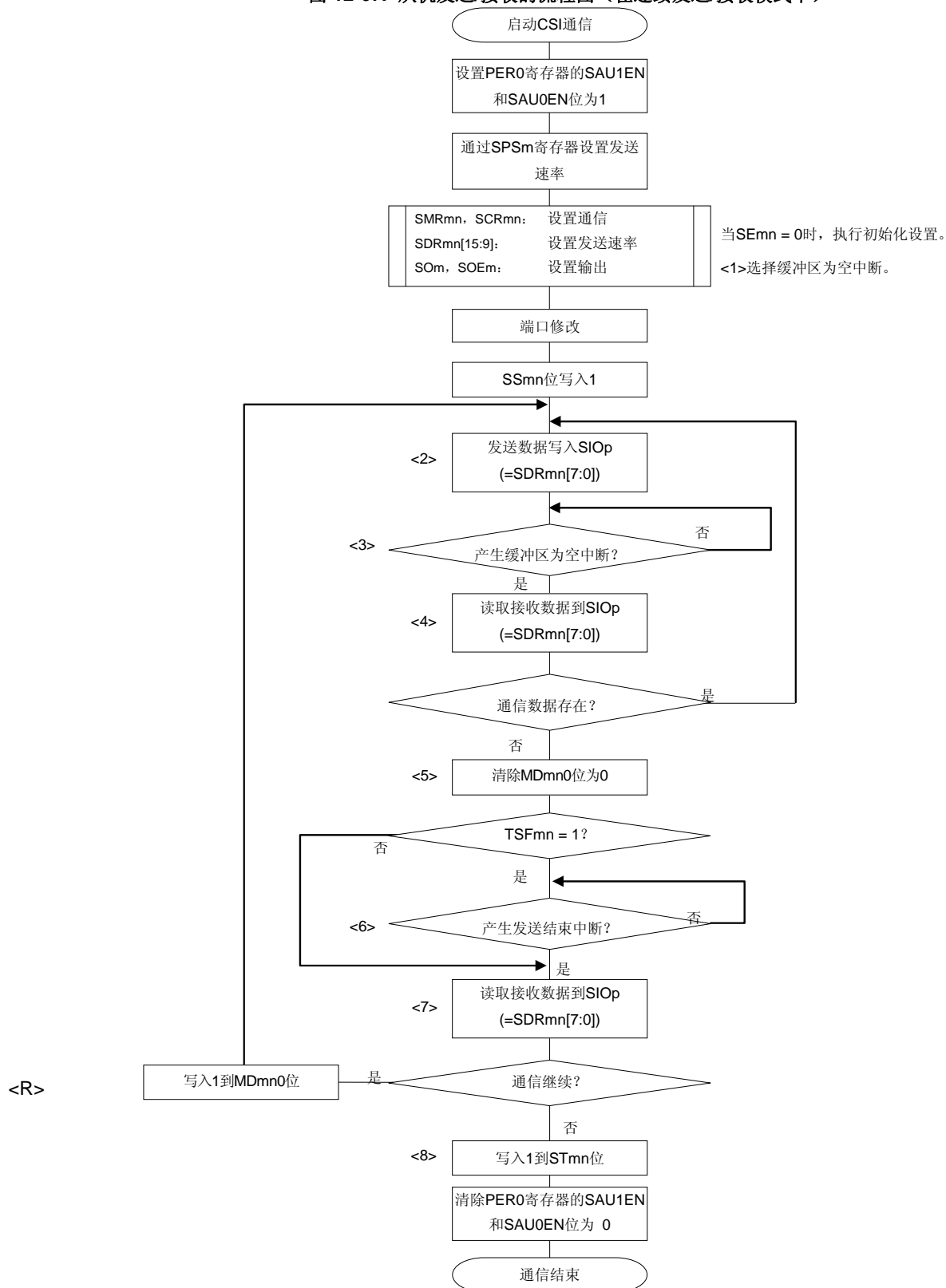


- 注 1. 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据被覆盖。
 注 2. 在这个周期内，可以通过读取 SDRmn 寄存器读出发送数据。此时，发送操作不受影响。

注意事项 在操作过程中，MDmn0 位可以被修改。
 但是，在最后一位的发送开始之前修改它，于是它将在最后发送数据的发送结束中断前被修改。

- 备注**
1. 上图中的<1> 到 <8>对应于图 12-67 从机发送/接收的流程图（在连续发送/接收模式下）中的 <1> 到 <8>。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2), p: CSI 编号 (p = 00, 01, 10, 20)

图 12-67. 从机发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 上图中的<1>~<8>对应于图 12-66 从机发送/接收的时序图（在连续发送/接收模式下）中的<1>~<8>。

12.5.7 计算发送时钟的频率

3线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的发送时钟频率可以通过下列表达式来计算。

(1) 主机

$$\text{(发送时钟频率)} = \{\text{目标通道的操作时钟 (MCK) 频率}\} \div (\text{SDRmn}[15: 9] + 1) \div 2 \text{ [Hz]}$$

(2) 从机

$$\text{(发送时钟频率)} = \{\text{主方提供的串行时钟 (SCK) 频率}\}^{\#} \text{ [Hz]}$$

<R>

注 允许的最大频率是 $f_{CLK}/6$ [MHz]和 $f_{MCK}/2$ [MHz]中较小的那个。

备注 1. SDRmn[15: 9]的值是 SDRmn 寄存器的第 15~9 位 (0000000B 到 1111111B)，所以它的取值是 0 到 127。

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2)

操作时钟 (MCK) 由串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的第 15 位来确定。

表 12-2. 操作时钟选择

SMRmn 寄存器	SPSm 寄存器								操作时钟 (MCK) ^{注1}	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
									INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{注2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
									INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{注2}	
其他									禁止设置	

<R>

<R>

<R> 注 1. 当更改 f_{CLK} 使用的时钟时（通过更改系统时钟控制寄存器（CKC）的值），在串行阵列单元（SAU）的操作停止（STm = 000FH）后再进行。当选择 INTTM02 和 INTTM03 作为操作时钟时，也要停止定时器阵列单元（TAU）（TT0 = 00FFH）。

<R> 2. 通过设置 TAU 的 TIS0 寄存器的 TIS02（如果 m = 0）和 TIS03（如果 m = 1）位为 1、选择 f_{SUB}/4 作为输入时钟，并使用 SPSm 寄存器选择 INTTM02 和 INTTM03，SAU 可以在副系统时钟固定分频情况下工作，而与 f_{CLK} 的频率（主系统时钟、副系统时钟）无关。但是，当更改 f_{CLK} 时，按照上面注 1 的描述，SAU 和 TAU 必须停止。

备注 1. X: 和此位无关

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~2)

12.6 UART (UART0, UART1, UART2, UART3) 通信的操作

这是一个使用 2 线进行通讯的功能，使用启动-停止信号来同步：串行数据发送 (TxD) 和串行数据接收 (RxD) 线。它与通信对象异步发送或接收数据 (通过使用一个内部波特率)。全双工 UART 通信可以通过使用 2 个通道来实现，一个专用于发送 (偶数编号通道)，另一个专用于接收 (奇数编号通道)。

[数据发送/接收]

- 5、7 或 8 位的数据长度
- 选择 MSB/LSB 先行
- 发送/接收数据的电平设置和反转选择
- 奇偶校验位附加和奇偶校验功能
- 停止位附加

[中断功能]

- 发送结束中断/缓冲区为空中断
- 错误中断可以是帧错误、奇偶校验错误或溢出错误

[错误检测标志]

- 帧错误、奇偶校验错误或溢出错误

UART3 (单元 1 的 2 和 3 通道) 支持 LIN 总线

[LIN 总线功能]

- 唤醒信号检测
- 同步间隔区域 (SBF) 检测
- 同步区域测量, 波特率计算

} 使用外部中断 (INTP0) 或定时器阵列单元 (TAU)。

UART0 使用 SAU0 的通道 0 和 1。

UART1 使用 SAU0 的通道 2 和 3。

UART2 使用 SAU1 的通道 0 和 1。

UART3 使用 SAU1 的通道 2 和 3。

单元	通道	用作CSI	用作UART	用作简易I ² C
0	0	CSI00	UART0	-
	1	CSI01		-
	2	CSI10	UART1	IIC10
	3	-		-
1	0	CSI20	UART2	IIC20
	1	-		-
	2	-	UART3 (支持LIN总线)	-
	3	-		-

UART 执行以下 4 种类型的通信操作。

- UART 发送 (见 12.6.1.)
- UART 接收 (见 12.6.2.)
- LIN 发送 (只有 UART3) (见 12.6.3.)
- LIN 接收 (只有 UART 3) (见 12.6.4.)

12.6.1 UART 发送

UART 发送是从 78K0R/KF3 异步（启动-停止同步）发送数据到另一个设备的操作。

在用作 UART 的两个通道中，偶数通道被用作 UART 发送。

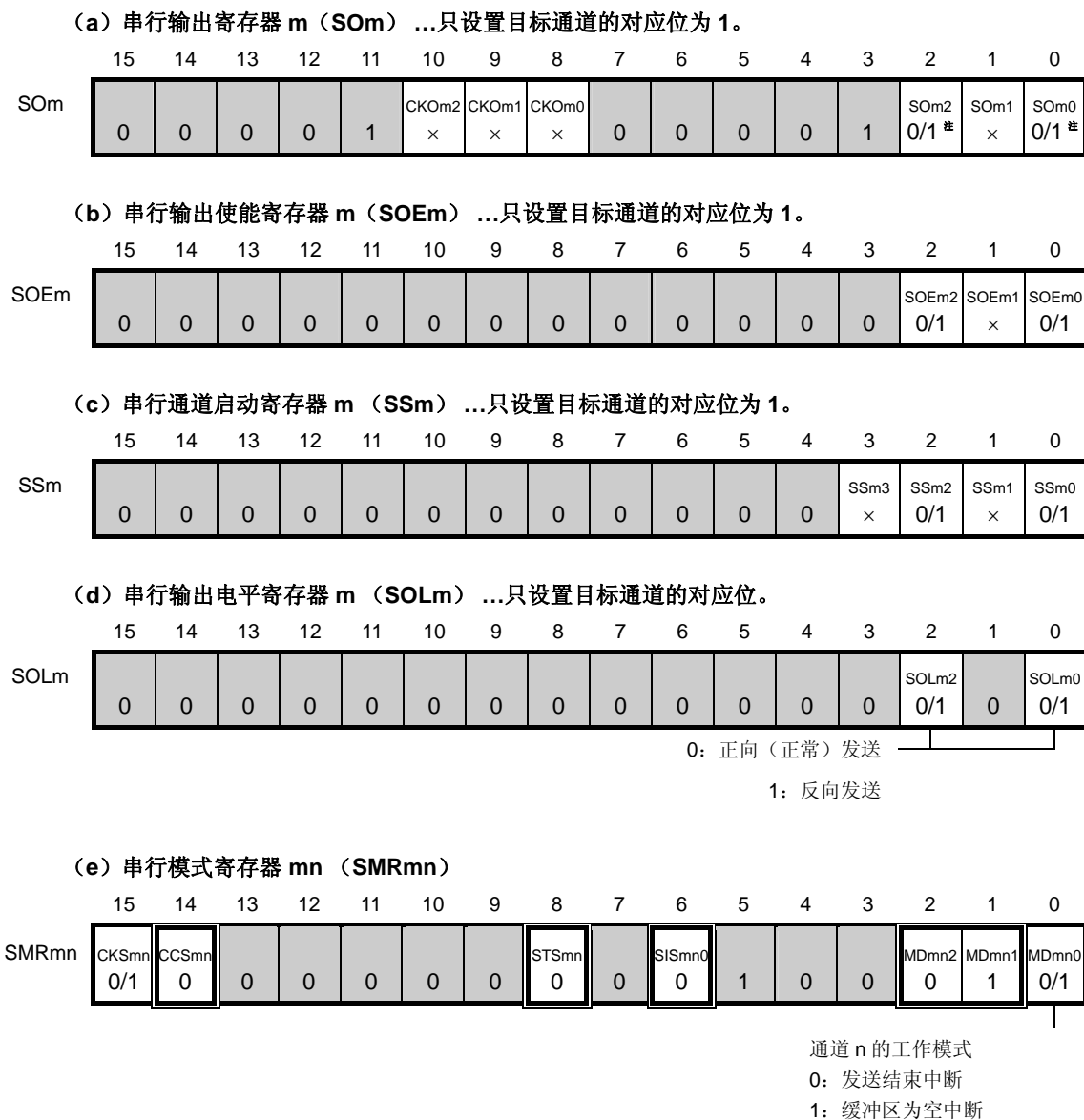
UART	UART0	UART1	UART2	UART3
目标通道	SAU0的通道0	SAU0的通道2	SAU1的通道0	SAU1的通道2
使用的引脚	TxD0	TxD1	TxD2	TxD3
中断	INTST0	INTST1	INTST2	INTST3
	可以选择发送结束中断（在单个发送模式下）或者缓冲区为空中断（在连续发送模式下）。			
错误检测标志	无			
发送数据长度	5、7或8位			
发送速率	最大 $f_{MCK}/6$ [bps]（SDRmn [15: 9] = 2 或更大），最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [‡]			
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）			
奇偶校验位	可以选择以下情况 <ul style="list-style-type: none"> • 无奇偶校验位 • 附加0校验 • 附加偶校验 • 附加奇校验 			
停止位	可以选择以下情况 <ul style="list-style-type: none"> • 附加1位 • 附加2位 			
数据方向	MSB 或 LSB先行			

注 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK} : 目标通道的操作时钟（MCK）频率
 f_{CLK} : 系统时钟频率

(1) 寄存器设置

图 12-68. UART (UART0, UART1, UART2, UART3) 发送寄存器内容的举例 (1/2)

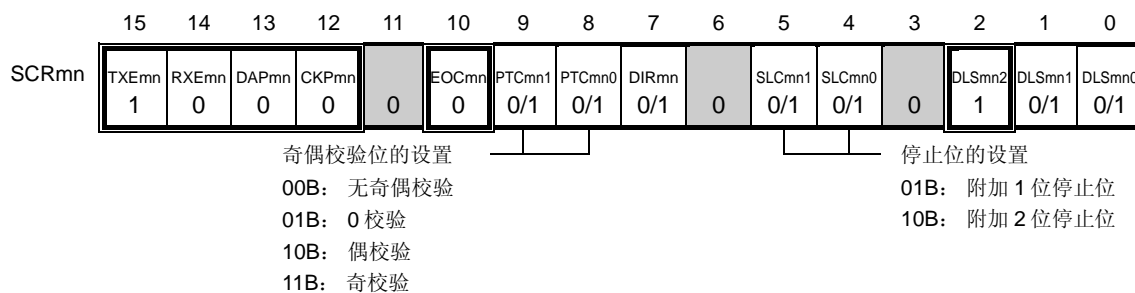


注 在启动发送前，当目标通道的 SOLmn 位被设置为 0 时，确认该位设置为 1，并且当目标通道的 SOLmn 位被设置为 1 时，确认该位设置为 0。这个值会根据通信过程中的通信数据而改变。

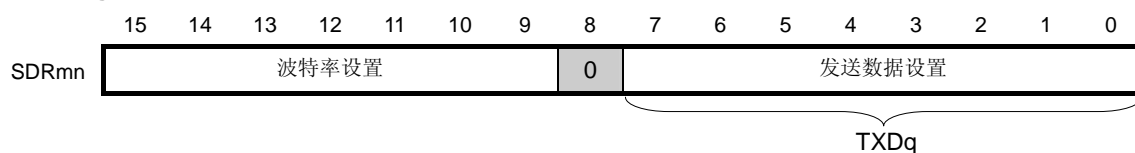
备注 m: 单元编号 (m=0, 1), n: 通道编号 (n=0, 2), q: UART 编号 (q=0~3)
 □: 在 UART 发送模式下设置固定, ■: 设置禁止 (置为初始值)
 x: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

图 12-68. UART (UART0, UART1, UART2, UART3) 发送寄存器内容的举例 (2/2)

(f) 串行通信操作设置寄存器 mn (SCRmn)



(g) 串行数据寄存器 mn (SDRmn) (低 8 位: TXDq)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), q: UART 编号 (q = 0~3)

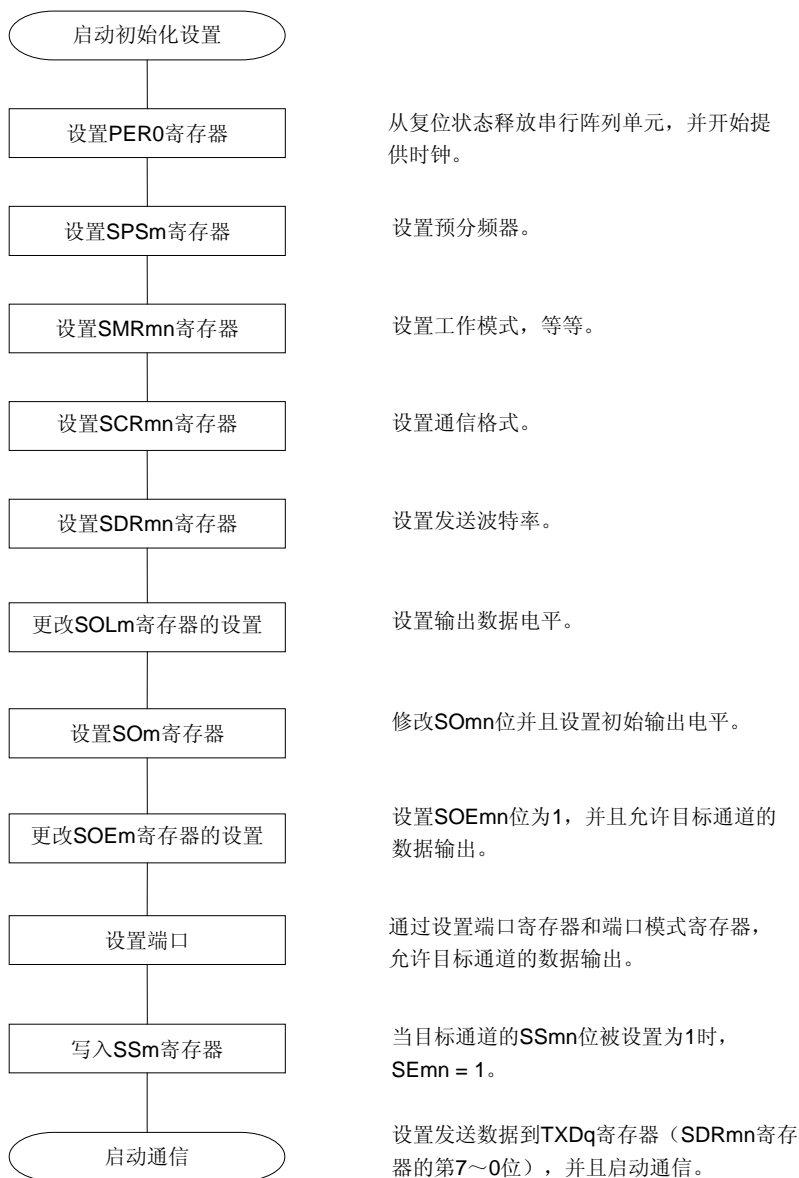
 □: 在 UART 发送模式下设置固定, ■: 设置禁止 (置为初始值)

 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)

 0/1: 根据用户的使用设置为 0 或 1

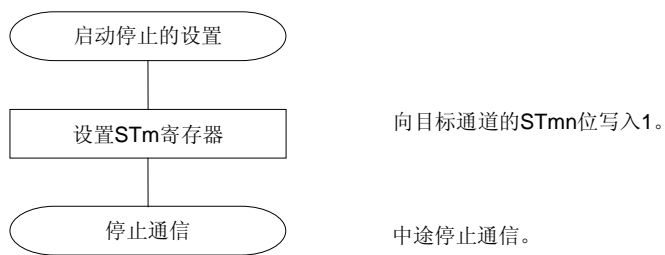
(2) 操作过程

图 12-69. UART 发送的初始化设置过程



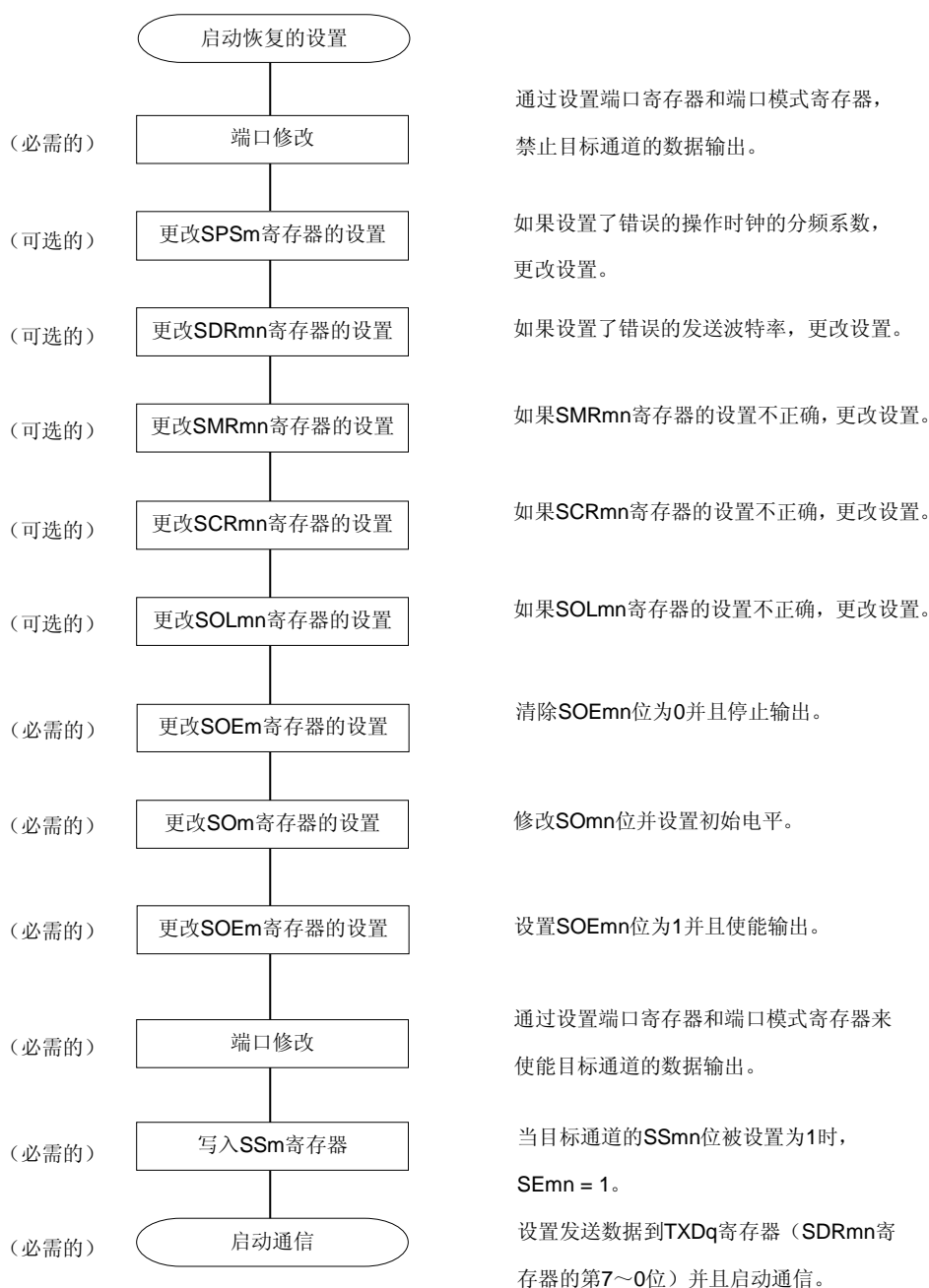
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-70. 停止 UART 发送的过程



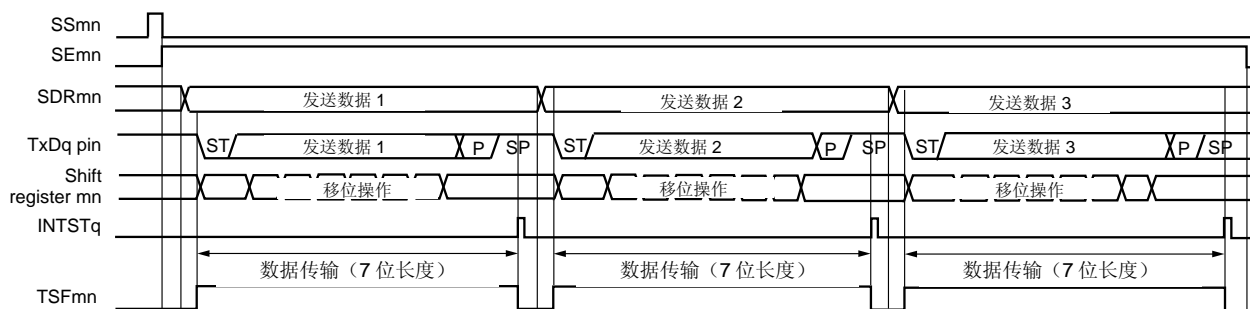
备注 即使在通信停止后，引脚的电平仍然会保持。要恢复操作，重新设置 S0m 寄存器（见图 12-71 恢复 UART 发送的过程）。

图 12-71. 恢复 UART 发送的过程



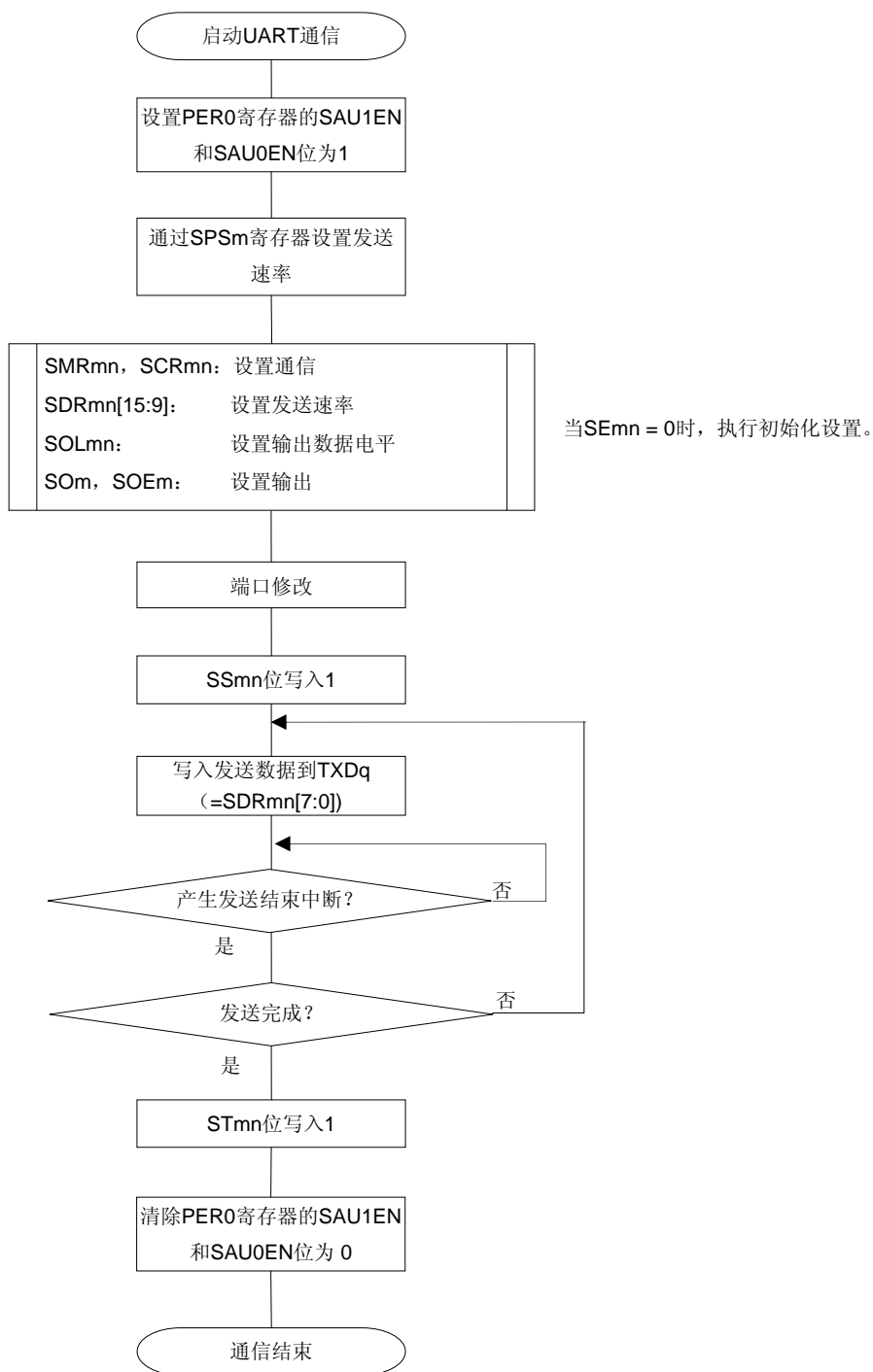
(3) 处理流程 (在单个发送模式下)

图 12-72. UART 发送的时序图 (在单个发送模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), q: UART 编号 (q = 0~3)

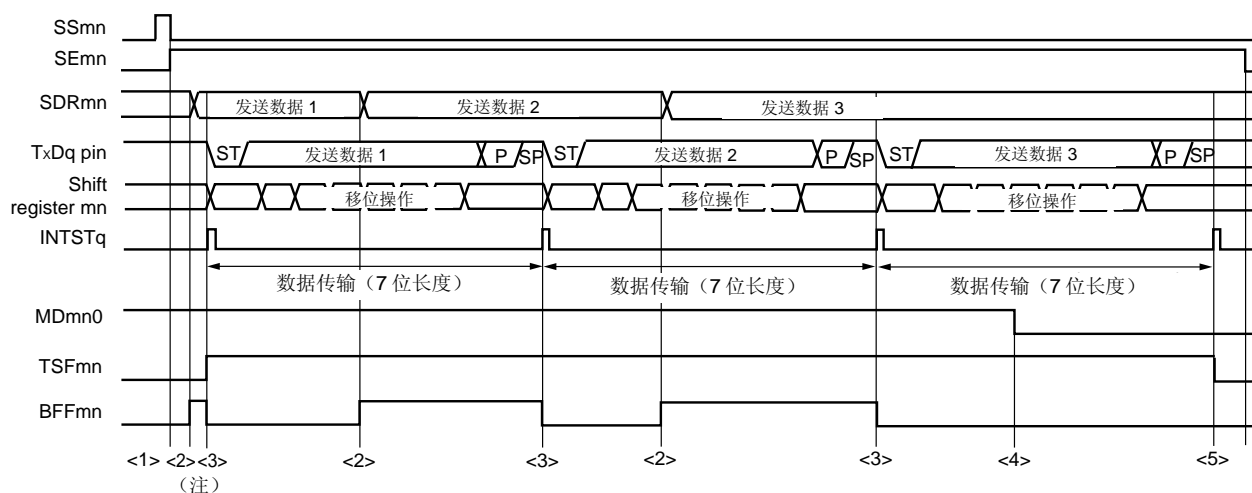
图 12-73. UART 发送的流程图（在单个发送模式下）



注意事项 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程 (在连续发送模式下)

图 12-74. UART 发送的时序图 (在连续发送模式下)

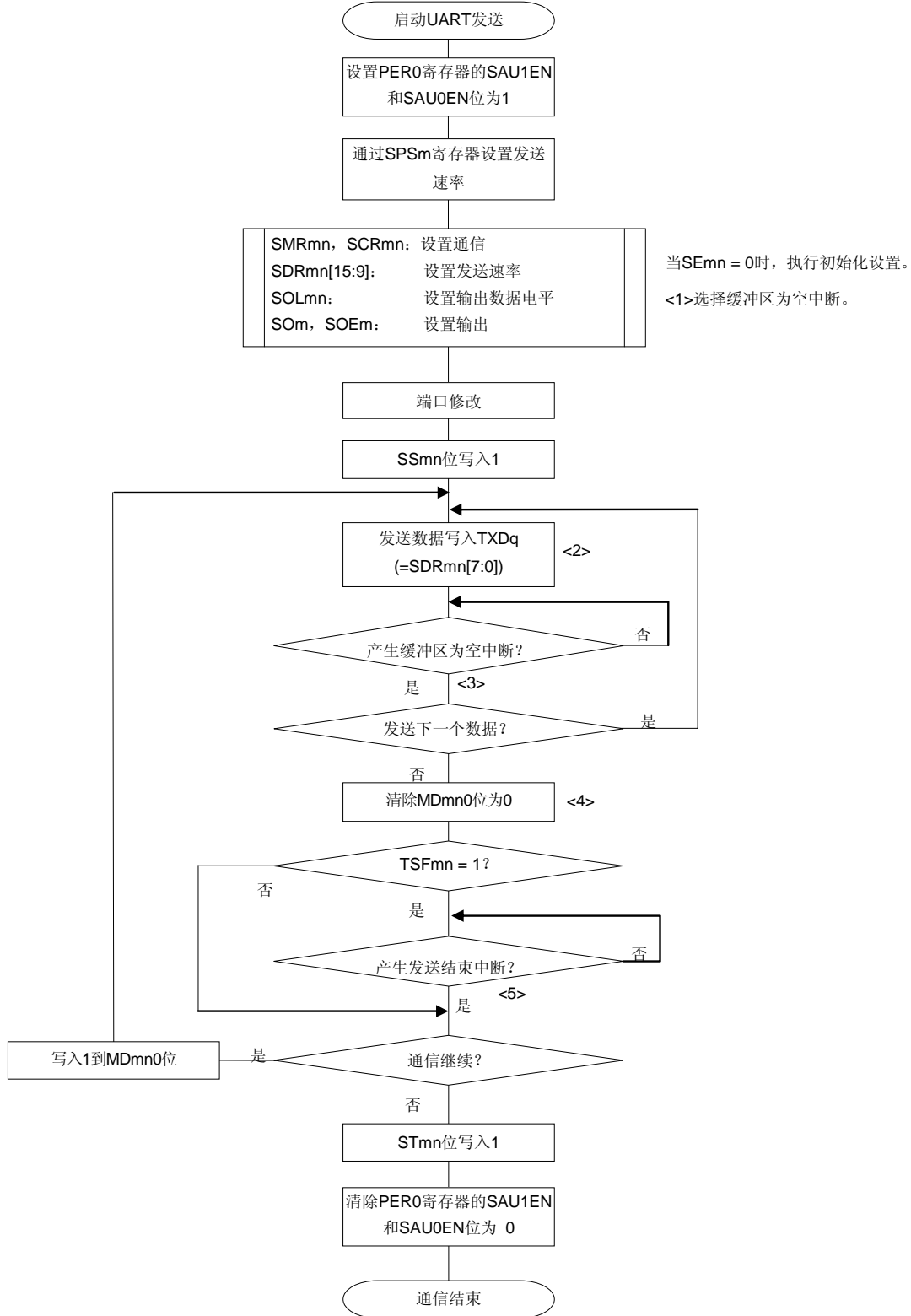


注 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据被覆盖。

注意事项 在操作过程中，MDmn0 位可以被修改。
但是，在最后一位的发送开始之前修改它，于是它将在最后发送数据的发送结束中断前被修改。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), q: UART 编号 (q = 0~3)

图 12-75. UART 发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 上图中的<1>到<5>对应图 12-74 UART 发送的时序图（在连续发送模式下）中的<1>到<5>。

12.6.2 UART接收

UART 接收是 78K0R/KF3 从其它设备异步接收数据的操作（启动-停止同步）。

用作 UART 的两个通道中的奇数通道被用来进行 UART 接收。

UART	UART0	UART1	UART2	UART3
目标通道	SAU0的通道1	SAU0的通道3	SAU1的通道1	SAU1的通道3
使用的引脚	RxD0	RxD1	RxD2	RxD3
中断	INTSR0	INTSR1	INTSR2	INTSR3
	只有发送结束中断（禁止设置缓冲区为空中断。）			
错误中断	INTSRE0	INTSRE1	INTSRE2	INTSRE3
错误检测标志	<ul style="list-style-type: none"> • 帧错误检测标志（FEFmn） • 奇偶校验错误检测标志（PEFmn） • 溢出错误检测标志（OVFmn） 			
<R> 发送数据长度	5、7或8位			
<R> 发送速率	最大 $f_{MCK}/6$ [bps]（SDRmn [15: 9] = 2或更大），最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] ^注			
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）			
奇偶校验位	可以选择以下情况 <ul style="list-style-type: none"> • 无奇偶校验位（无奇偶校验） • 附加0校验（无奇偶校验） • 附加偶校验 • 附加奇校验 			
停止位	附加1位停止位			
数据方向	MSB 或 LSB先行			

<R> 注 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK} : 目标通道的操作时钟（MCK）频率
 f_{CLK} : 系统时钟频率

(1) 寄存器设置

图 12-76. UART (UART0, UART1, UART2, UART3) 接收寄存器内容的举例 (1/2)

(a) 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					1	CKOm2	CKOm1	CKOm0					1	SOM2	SOM1	SOM0
	0	0	0	0	1	x	x	x	0	0	0	0	1	x	x	x

(b) 串行输出使能寄存器 m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm														SOEm2	SOEm1	SOEm0
<R>	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0/1	x

当用于 UART 接收时，设置这个位为 0

(c) 串行通道启动寄存器 m (SSm) ...只设置目标通道的对应位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SSm														SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	x	0/1	x

(d) 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	1	0	0/1	1	0	0	0	1	0

0: 正向 (正常) 发送
1: 反向发送

通道 n 的工作模式
0: 发送结束中断

(e) 串行模式寄存器 mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr	CCSmr						STSmr		SISmr0				MDmr2	MDmr1	MDmr0
<R>	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0/1

与 CKSmn 的设置相同

通道 r 的工作模式
0: 发送结束中断
1: 缓冲区为空中断

(f) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	0	1	0	0	0	1	0/1	0/1	0/1	0	0	1	0	1	0/1	0/1

注意事项 对于 UART 接收，确认设置与通道 n 成对使用的通道 r 的 SMRmr。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), r: 通道编号 (r = n - 1),

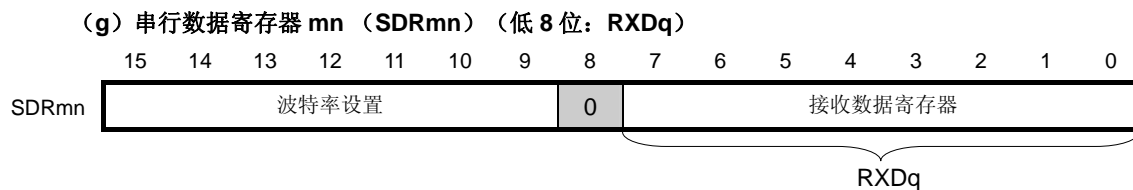
q: UART 编号 (q = 0~3)

□: 在 UART 接收模式下设置固定, ■: 设置禁止 (置为初始值)

x: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)

0/1: 根据用户的使用设置为 0 或 1

图 12-76. UART (UART0, UART1, UART2, UART3) 接收寄存器内容的举例 (2/2)



注意事项 对于 UART 接收，确保设置通道 r 的 SMRmr，通道 r 是与通道 n 成对使用的偶数通道。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), r: 通道编号 (r = n - 1),
q: UART 编号 (q = 0~3)

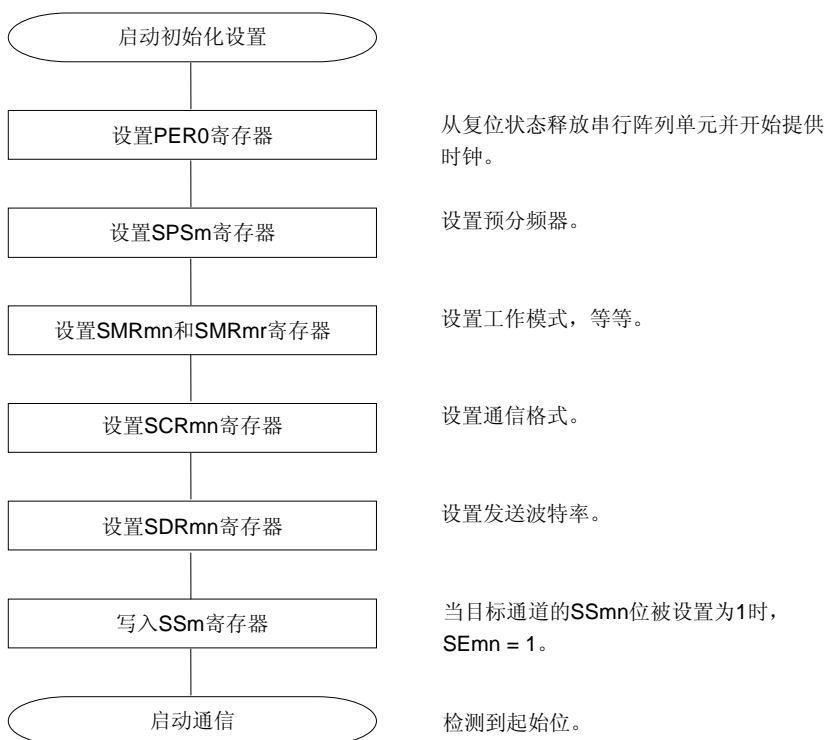
: 在 UART 发送模式下设置固定, : 设置禁止 (置为初始值)

x: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)

0/1: 根据用户的使用设置为 0 或 1

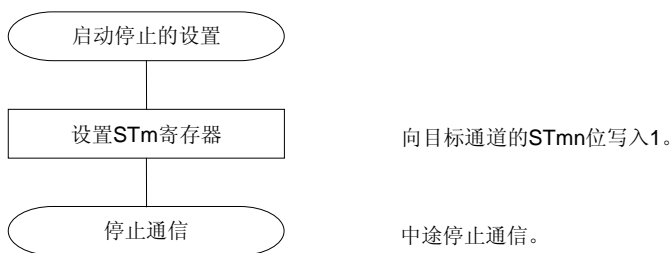
(2) 操作过程

图 12-77. UART 接收的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 12-78. 停止 UART 接收过程



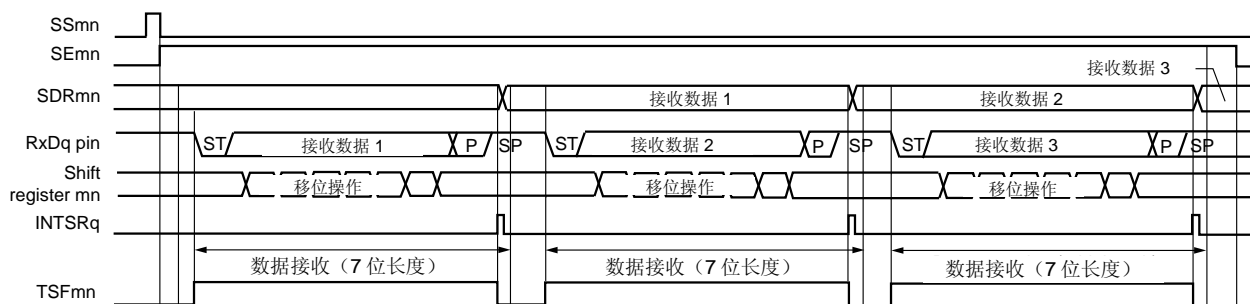
<R>

图 12-79. 重新启动 UART 接收的过程



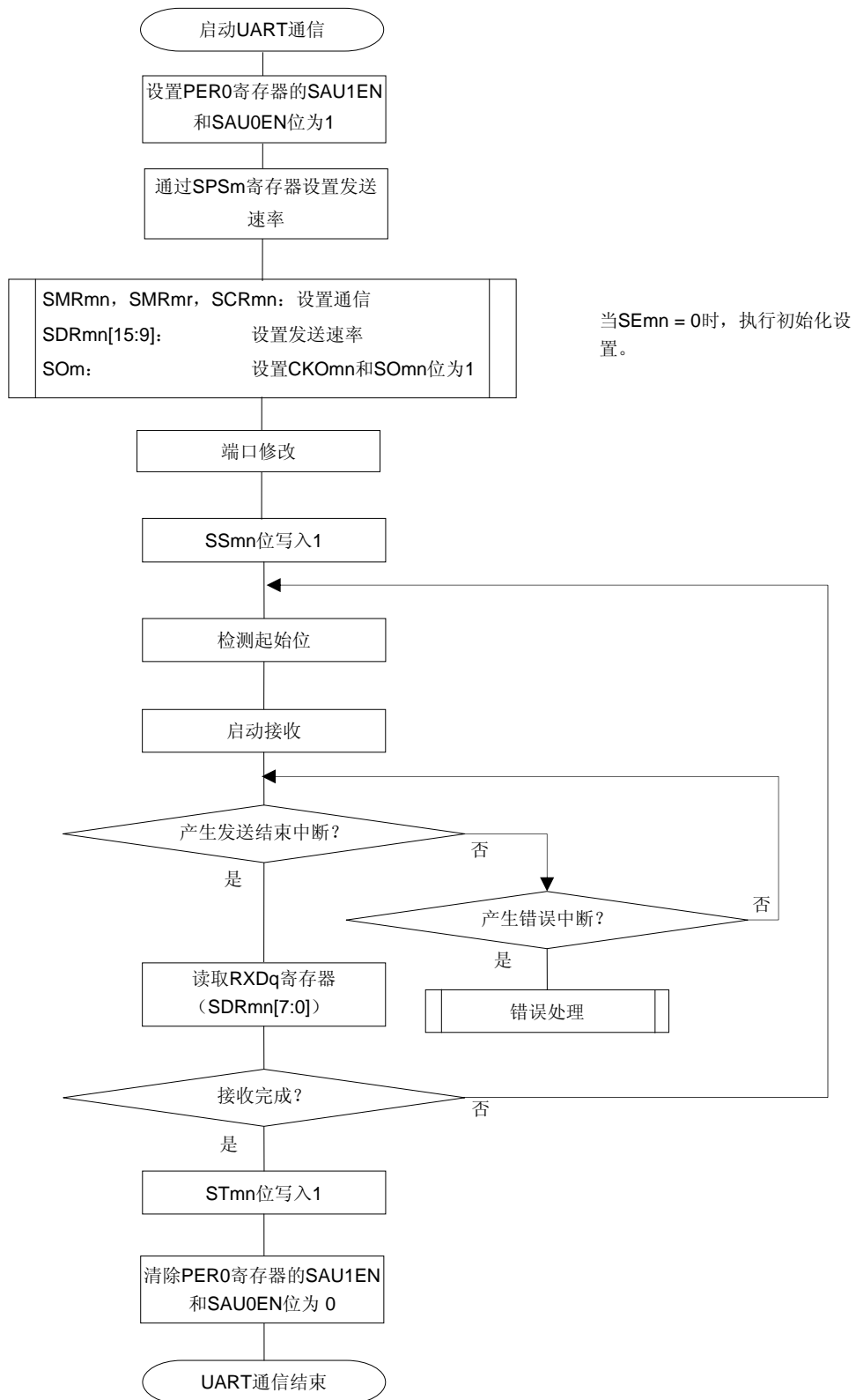
(3) 处理流程

图 12-80. UART 接收的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), q: UART 编号 (q = 0~3)

图 12-81. UART 接收的流程图



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

12.6.3 LIN发送

在 UART 发送中，UART3 支持 LIN 通信。

LIN 发送使用单元 1（SAU1）的通道 2。

UART	UART0	UART1	UART2	UART3
LIN通信的支持	不支持	不支持	不支持	支持
目标通道	-	-	-	SAU1的通道2
使用的引脚	-	-	-	TxD3
中断	-	-	-	INTST3
	可以选择发送结束中断（在单个发送模式下）或缓冲区为空中断（在连续发送模式下）。			
错误检测标志	无			
发送数据长度	8 位			
发送速率	最大 $f_{MCK}/6$ [bps]（ $SDR_{mn}[15:9] = 2$ 或更大），最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [#]			
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）			
奇偶校验位	可以选择以下情况 <ul style="list-style-type: none"> • 无奇偶校验位 • 附加0奇偶检验 • 附加偶校验 • 附加奇校验 			
停止位	可以选择以下情况 <ul style="list-style-type: none"> • 附加1位 • 附加2位 			
数据方向	MSB 或 LSB先行			

<R> 注 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK} : 目标通道的操作时钟（MCK）频率
 f_{CLK} : 系统时钟频率

LIN 就是本地互连网络（Local Interconnect Network），并且是低速（1 到 20kbps）串行通信协议，用来减少汽车网络成本。

LIN 的通信是单个主机通信，并且一个主机最多可以连接 15 个从机。

从机被用来控制开关、制动器和传感器，它们通过 LIN 连接到主机。

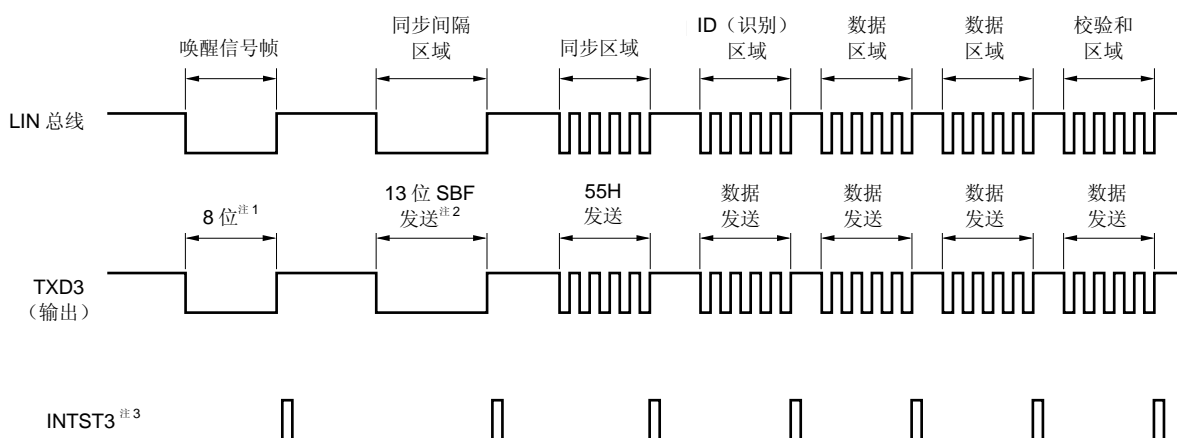
通常，主机被连接到一个网络上，例如 CAN（控制器区域网络）。

LIN 总线是个单线总线，通过收发器连接到它上面的节点要符合 ISO9141 标准

根据 LIN 的协议，主机发送一帧数据时会附加波特率信息。从机接收这一帧数据并修正自己与主机的波特率误差。如果从机的波特率误差在 ±15%内，通信可以建立。

图 12-82 概述了 LIN 的发送操作。

图 12-82. LIN 的发送操作

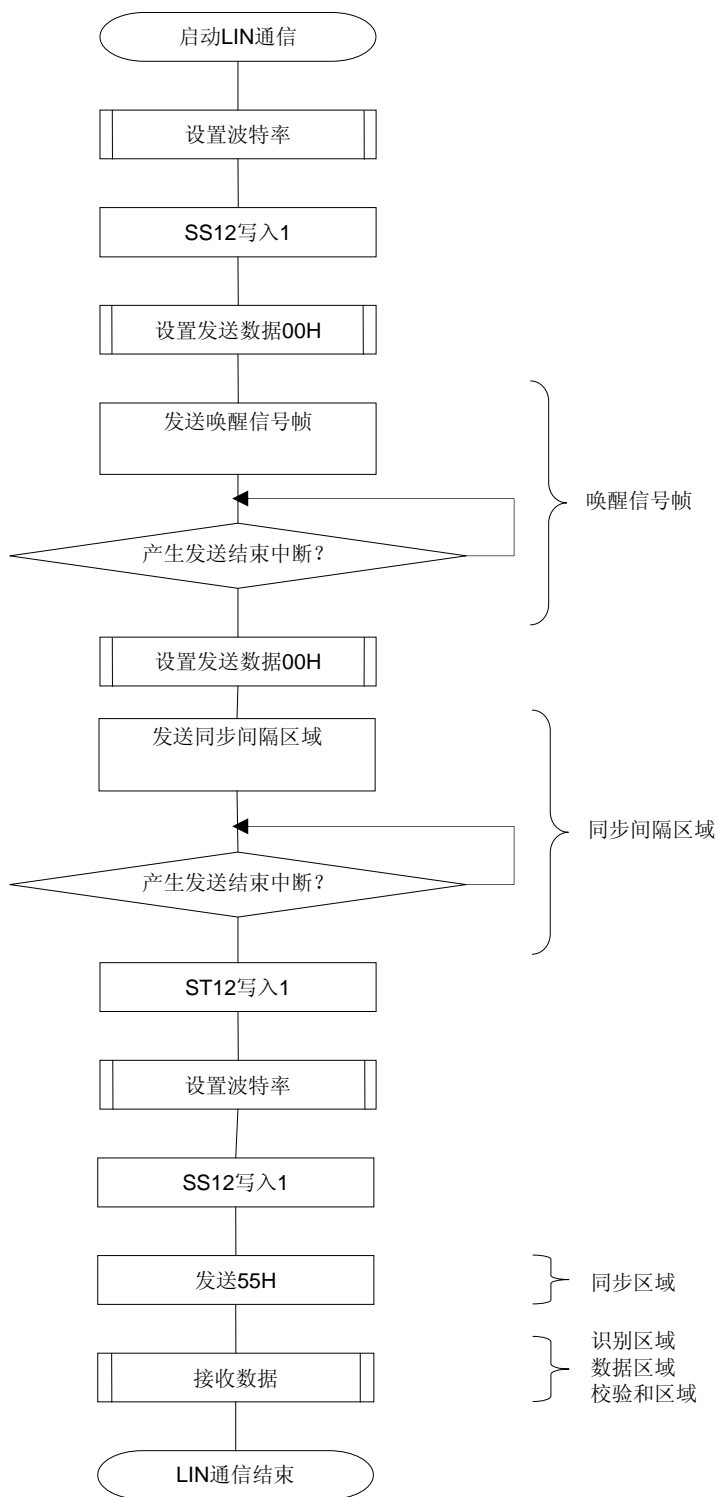


- 注
1. 设置波特率是为了满足唤醒信号的标准，并且数据 00H 被发送。
 2. 同步间隔区域被定义为有 13 位宽度，并输出低电平。当主发送的波特率为 N [bps]时，同步间隔区域的波特率按照下面计算。

$$\text{(同步间隔区域的波特率)} = 8/13 \times N$$
 以这个波特率发送数据 00H，就产生一个同步间隔区域。
 3. INTST3 在发送完成时被输出。当执行 SBF 发送时，INTST3 也会被输出。

备注 区域之间的间隔由软件控制。

图 12-83. LIN 发送的流程图



12.6.4 LIN接收

在 UART 接收中，UART3 支持 LIN 通信。

LIN 接收使用单元 1 (SAU1) 的通道 3。

UART	UART0	UART1	UART2	UART3
LIN通信的支持	不支持	不支持	不支持	支持
目标通道	-	-	-	SAU1的通道3
使用的引脚	-	-	-	RxD3
中断	-	-	-	INTSR3
	只有发送结束中断（禁止设置缓冲区为空中断。）			
错误中断	-	-	-	INTSRE3
错误检测标志	<ul style="list-style-type: none"> • 帧错误检测标志 (FEFmn) • 奇偶校验错误检测标志 (PEFmn) • 溢出错误检测标志 (OVFmn) 			
发送数据长度	8位			
发送速率	最大 $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2或更大)，最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [‡]			
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）			
奇偶校验位	可以选择以下情况 <ul style="list-style-type: none"> • 无奇偶校验位 • 附加0奇偶校验 • 附加偶校验 • 附加奇校验 			
停止位	可以选择以下情况 <ul style="list-style-type: none"> • 附加1位停止位 • 附加2位停止位 			
数据方向	MSB 或 LSB先行			

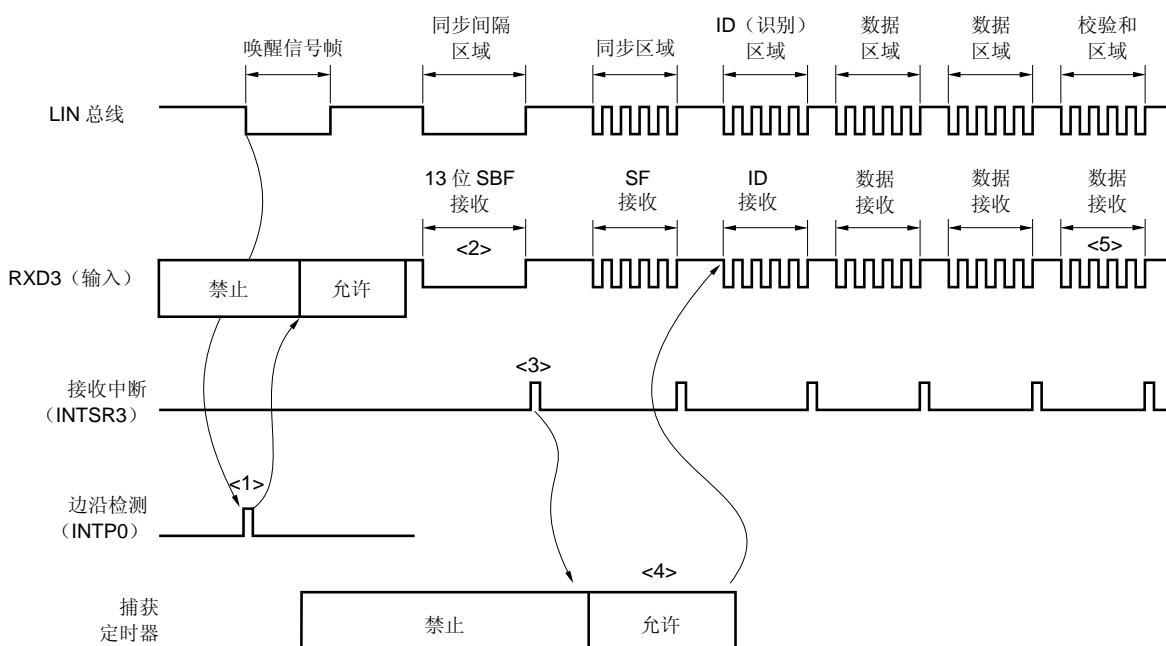
<R>
<R>

注 操作时需要满足以上条件，且在电气规范的 AC 特性（见 第二十八章 电气规范）范围内使用。

备注 f_{MCK} : 目标通道的操作时钟 (MCK) 频率
 f_{CLK} : 系统时钟频率

图 12-84 概述 LIN 的接收操作。

图 12-84. LIN 的接收操作



这是信号处理的流程。

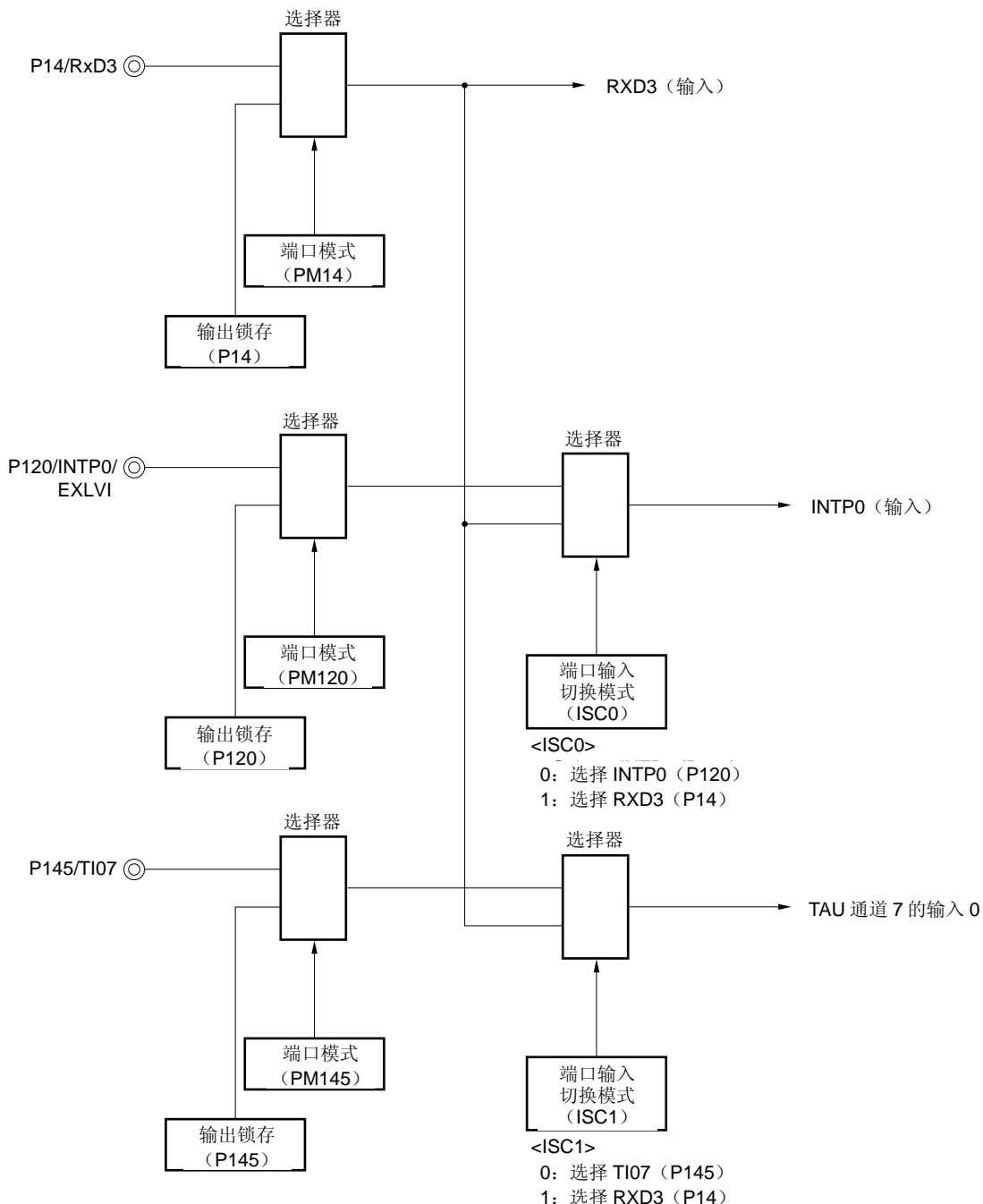
- <1> 在引脚上检测到中断边沿 (INTPO) 发生时, 认为是唤醒信号。当检测唤醒信号时, 允许 UART3 的接收 (RXE13 = 1), 并且等待 SBF 接收。
- <2> 当检测到 SBF 的起始位时, 按照设置的波特率启动接收, 并且串行数据按顺序被保存到 RXD3 寄存器 (= 串行数据寄存器 13 (SDR13) 的第 7~0 位)。当检测到停止位时, 产生接收结束中断请求 (INTSR3)。当 11 位或更多的低电平数据被检测为 SBF 时, 可以认为 SBF 接收已经正确完成。如果检测到的 SBF 是少于 11 位的低电平数据, 可以认为发生 SBF 接收错误, 并且系统返回 SBF 接收等待状态。
- <3> 当 SBF 接收正确完成时, 启动定时器阵列单元的通道 7 并测量同步区域的位间隔 (脉冲宽度) (见 6.7.5 用作输入信号高/低电平宽度测量操作)。
- <4> 从同步区域 (SF) 的位间隔计算波特率误差。停止 UART3 一次, 并调整 (重新设置) 波特率。
- <5> 校验和区域应该由软件来区别。此外, 校验和被接收后的 UART3 初始化处理也由软件执行, SBF 接收等待处理也要通过软件执行。

图 12-85 显示 LIN 接收端口的配置。

检测外部中断 (INTP0) 的边沿, 可以接收到从 LIN 主机发送的唤醒信号。通过使用定时器阵列单元 (TAU) 的外部事件捕获操作, 主机发送的同步区域长度可以被测量, 并用来计算波特率误差。

通过控制端口输入 (ISC0/ISC1) 的切换, 从用于接收的端口 (RXD3) 输入的输入源可以输入到外部中断引脚 (INTP0) 和定时器阵列单元 (TAU)。

图 12-85. 操作 LIN 接收的端口配置



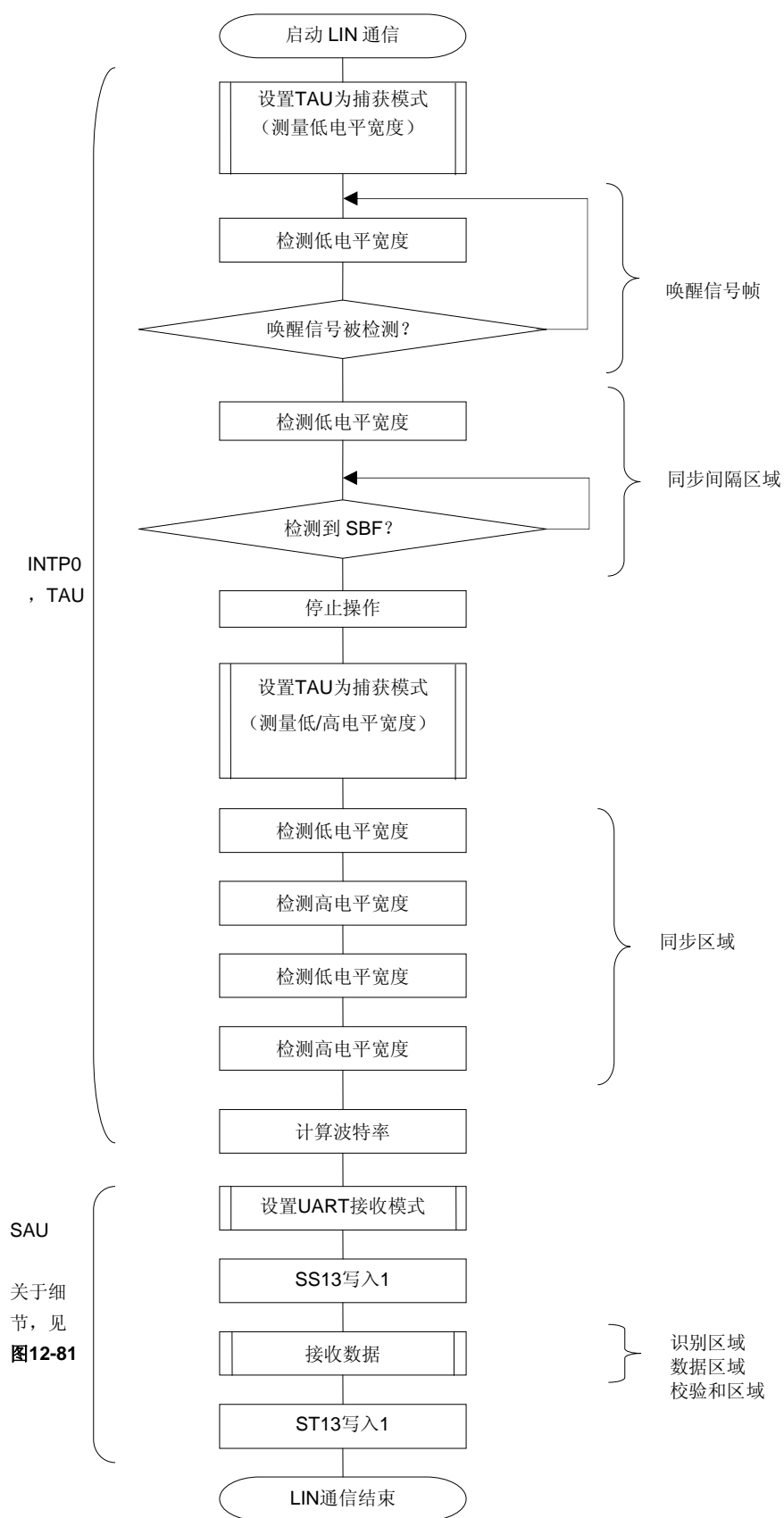
备注 INSC0, INSC1: 输入切换控制寄存器 (ISC) 的第 0 位和第 1 位 (见图 12-17。)

用作 LIN 通信的外围功能如下所示。

<使用的外围功能>

- 外部中断 (INTP0)：唤醒信号检测
用途：检测唤醒信号的边沿，并启动通信
- 定时器阵列单元 (TAU) 的通道 7：波特率误差检测
用途：检测同步区域 (SF) 的长度，并且除以位数来检测误差（输入到 RXD3 的边沿间隔在捕获模式下测量）。
- 串行阵列单元 (SAU1) 的通道 2 和 3 (UART3)

图 12-86. LIN 接收的流程图



12.6.5 计算波特率

(1) 波特率计算表达式

UART (UART0, UART1, UART2, UART3) 通信的波特率可以按照下列表达式来计算。

$$\text{(波特率)} = \{\text{目标通道的操作时钟 (MCK) 频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [bps]}$$

注意事项 禁止设置 **SDRmn [15:9] = (0000000B, 0000001B)**。

- 备注**
1. 当使用 UART 时, SDRmn[15:9]的值是 SDRmn 寄存器的位 15 到 9 的值 (0000010B 到 1111111B), 因此它的取值为 2 到 127。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3)

操作时钟 (MCK) 由串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的第 15 位 (CKSmn) 来确定。

表 12-3. 操作时钟选择

SMRmn 寄存器	SPSm 寄存器								操作时钟 (MCK) ^{注1}	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
除上面以外									禁止设置	

<R>

<R>

<R> 注 1. 当更改 f_{CLK} 使用的时钟时 (通过更改系统时钟控制寄存器 (CKC) 的值), 在串行阵列单元 (SAU) 的操作停止 (STm = 000FH) 后再进行。当选择 INTTM02 和 INTTM03 作为操作时钟时, 也要停止定时器阵列单元 (TAU) (TT0 = 00FFH)。

<R> 2. 通过设置 TAU 的 TIS0 寄存器的 TIS02 (如果 m = 0) 和 TIS03 (如果 m = 1) 位为 1、选择 f_{SUB}/4 作为输入时钟, 并使用 SPSm 寄存器选择 INTTM02 和 INTTM03, SAU 可以在副系统时钟固定分频情况下工作, 而与 f_{CLK} 的频率 (主系统时钟、副系统时钟) 无关。但是, 当更改 f_{CLK} 时, 按照上面注 1 的描述, SAU 和 TAU 必须停止。

备注 1. X: 和此位无关

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0~3)

(2) 发送过程中的波特率误差

发送过程中的 UART (UART0, UART1, UART2, UART3) 的波特率误差可以按照下列表达式来计算。确认发送端的波特率在接收端允许的波特率范围内。

$$\text{(波特率)} = \text{(计算的波特率值)} \div \text{(目标波特率)} \times 100 - 100 \text{ [%]}$$

这是在 $f_{CLK} = 20 \text{ MHz}$ 时设置 UART 波特率的一个示例。

UART波特率 (目标波特率)	$f_{CLK} = 20 \text{ MHz}$			
	操作时钟 (MCK)	SDRmn[15:9]	计算得到的波特率	与目标波特率的误差
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

(3) 接收允许的波特率范围

在 UART (UART0, UART1, UART2, UART3) 通信过程中, 接收允许的波特率范围可以按照下列表达式来计算。确保发送端的波特率在接收端允许的波特率范围内。

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

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

Brate: 接收端计算得到的波特率值 (见 12.6.5 (1) 波特率计算表达式。)

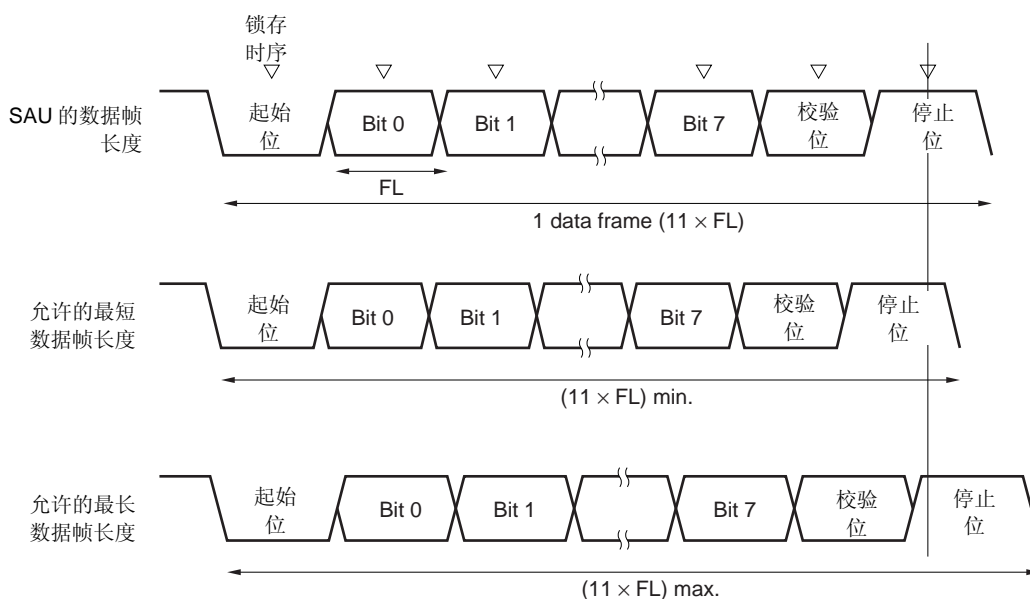
k: SDRmn[15:9] + 1

Nfr: 1 个数据帧长度[位]

= (起始位) + (数据长度) + (奇偶校验位) + (停止位)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3)

图 12-87. 接收允许的波特率范围 (1 个数据帧长度 = 11 位)



如图 12-87 所示, 在检测到起始位后, 接收数据的锁存时序由串行时间寄存器 mn (SDRmn) 的第 15~9 位设置的分频系数来确定。如果在锁存时序之前接收到最后一个数据 (停止位), 数据可以正确接收。

12.7 简易I²C (IIC10, IIC20) 通信操作

这是一个使用 2 线与两个或更多设备进行同步通信的功能：串行时钟 (SCL) 和串行数据 (SDA)。这个简易 I²C 功能适用于与一个设备单独通信，例如 EEPROM、flash 存储器或模/数转换器。因此，它只能用作主机方而不能检测等待状态。使用软件，也就是操作那些控制寄存器，来确保启动和停止情况下遵守 AC 规范。

[数据发送/接收]

- 主机发送、主机接收（只有单个主机的主机功能）
- ACK 输出和 ACK 检测功能
- 8 位的数据长度（当传送地址时，通过高 7 位指定地址，最低位用作读/写控制。）
- 启动条件和停止条件的手动产生

[中断功能]

- 发送结束中断

[错误检测标志]

- 奇偶校验错误 (ACK 错误)

* [简易 I²C 不支持的功能]

- 从机发送、从机接收
- 仲裁失利检测功能
- 等待检测功能

备注 要使用全功能的 I²C 总线，见第十三章 串行接口 IIC0。

SAU0 的通道 2 以及 SAU1 的通道 0 支持 I²C (IIC10, IIC20) 功能。

单元	通道	用作CSI	用作UART	用作简易I ² C
0	0	CSI00	UART0	-
	1	CSI01		-
	2	CSI10	UART1	IIC10
	3	-		-
1	0	CSI20	UART2	IIC20
	1	-		-
	2	-	UART3 (支持LIN总线)	-
	3	-		-

简易 I²C (IIC10, IIC20) 执行下面 4 种类型的通信操作。

- 地址域发送 (见 12.7.1.)
- 数据发送 (见 12.7.2.)
- 数据接收 (见 12.7.3.)
- 停止条件产生 (见 12.7.4.)

12.7.1 地址域发送

地址域发送是在 I²C 通信中首先执行的操作，用来确定发送的目标（从机）。在产生启动条件后，地址（7 位）和发送方向（1 位）在一帧数据中发送。

简易 I ² C	IIC10	IIC20
目标通道	SAU0的通道2	SAU1的通道0
使用的引脚	SCL10, SDA10	SCL20, SDA20
中断	INTIIC10	INTIIC20
	只有发送结束中断（禁止设置缓冲区为空中断。）	
错误检测标志	奇偶校验错误检测标志（PEFmn）	
发送数据长度	8位（指定高7位为地址，最低位为R/W控制）	
发送速率	最大f _{CLK} /4 MHz f _{CLK} : 系统时钟频率 但是，在I ² C的每种模式下，必须满足以下条件。 <ul style="list-style-type: none"> • 最大400 kHz（优先模式） • 最大100 kHz（标准模式） 	
数据电平	正向输出（默认：高电平）	
奇偶校验位	无奇偶校验位	
停止位	附加1位（用于ACK接收时序）	
数据方向	MSB 先行	

(2) 操作过程

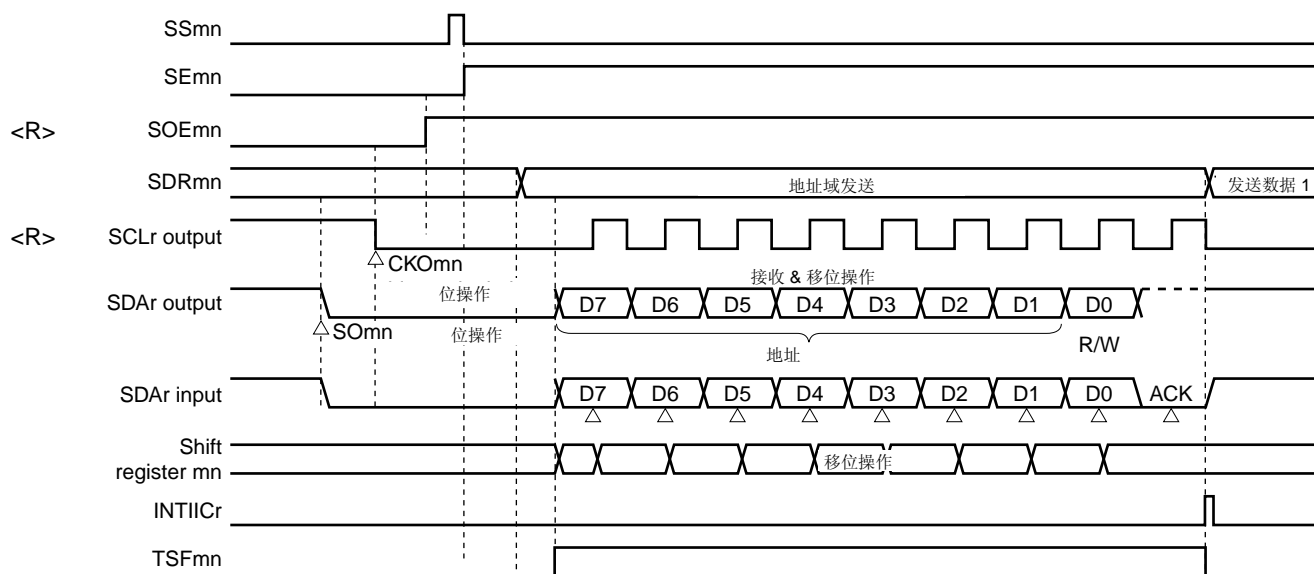
图 12-89. 地址域发送的初始化设置过程



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

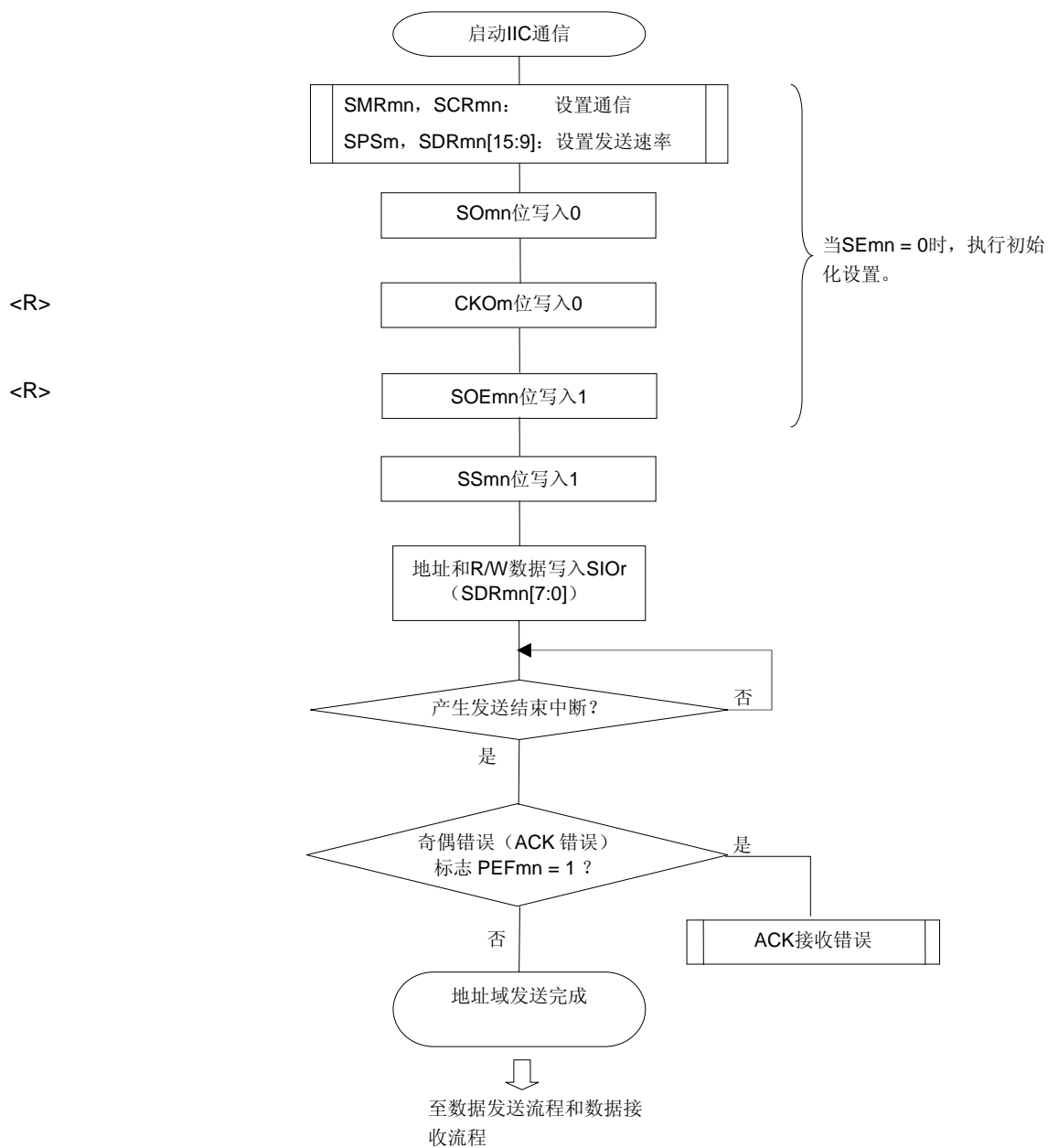
(3) 处理流程

图 12-90. 地址域发送的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)

图 12-91. 地址域发送的流程图



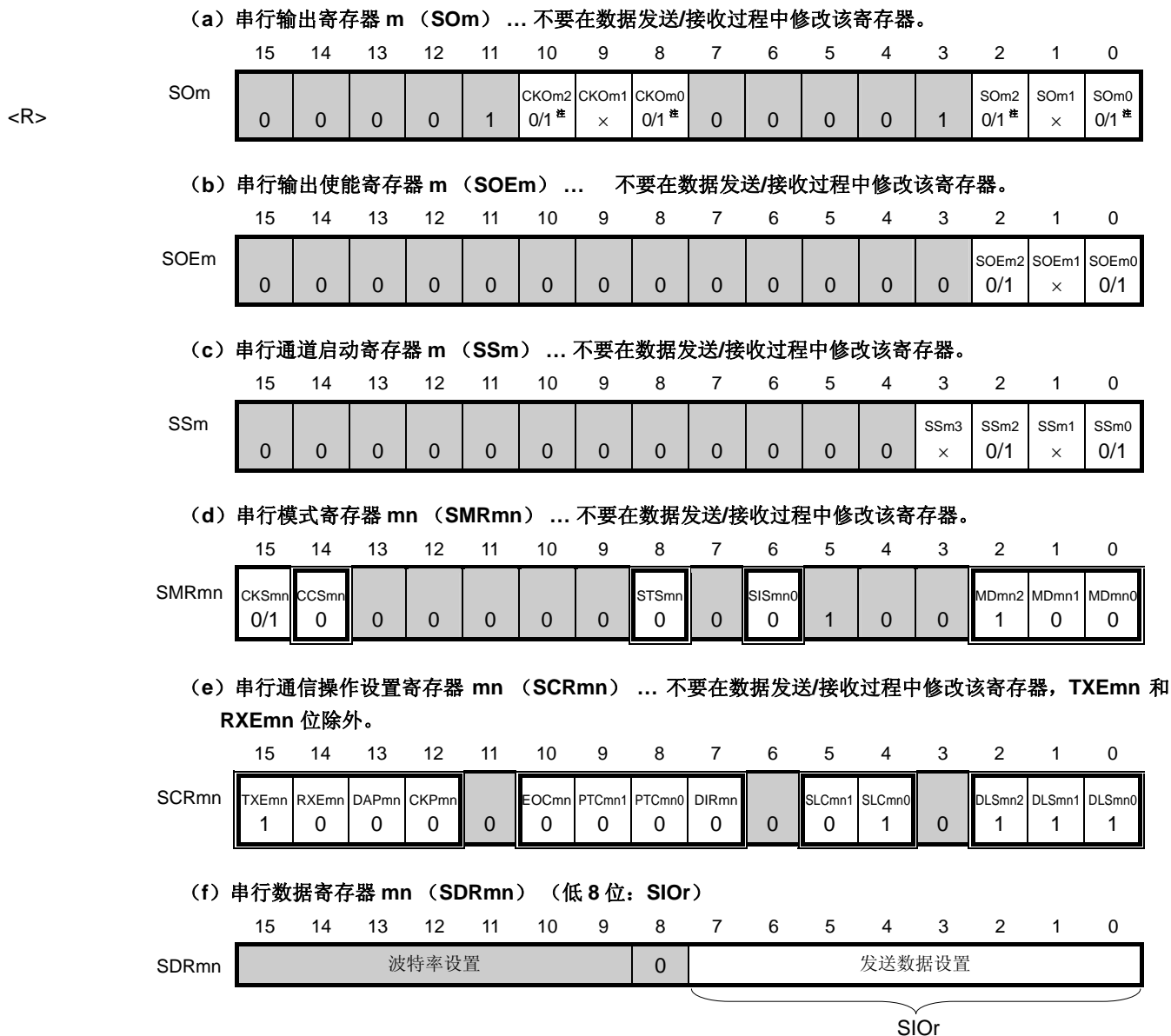
12.7.2 数据发送

数据发送是在发送地址域之后将数据发送到发送目标（从机）的操作。在所有数据被发送到从机后，产生一个停止条件，并且释放总线。

简易I ² C	IIC10	IIC20
目标通道	SAU0的通道2	SAU1的通道0
使用的引脚	SCL10, SDA10	SCL20, SDA20
中断	INTIIC10	INTIIC20
	只有发送结束中断（禁止设置缓冲区为空中断。）	
错误检测标志	奇偶校验错误检测标志（PEFmn）	
发送数据长度	8位	
发送速率	最大f _{CLK} /4 MHz f _{CLK} : 系统时钟频率 但是，在I ² C的每种模式下，必须满足以下条件。 <ul style="list-style-type: none"> • 最大400 kHz（优先模式） • 最大100 kHz（标准模式） 	
数据电平	正向输出（默认：高电平）	
奇偶校验位	无奇偶校验位	
停止位	附加1位（用于ACK接收时序）	
数据方向	MSB 先行	

(1) 寄存器设置

图 12-92. 简易 I²C (IIC10, IIC20) 数据发送寄存器内容的举例

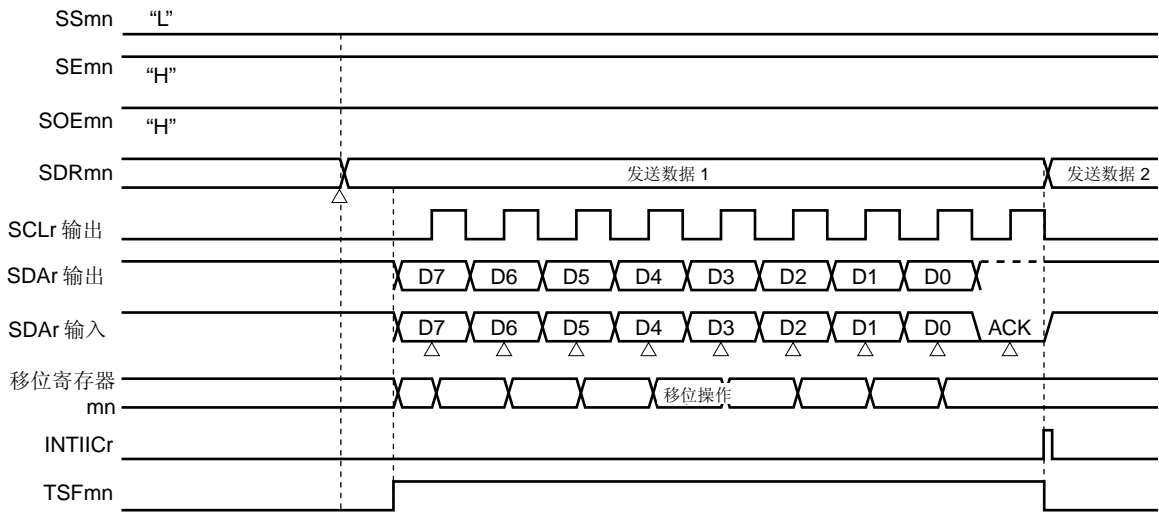


注 值根据通信操作过程中的通信数据而改变。

- 备注
- m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)
 - : 在 IIC 模式下设置固定, : 设置禁止 (置为初始值)
 - ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 - 0/1: 根据用户的使用设置为 0 或 1

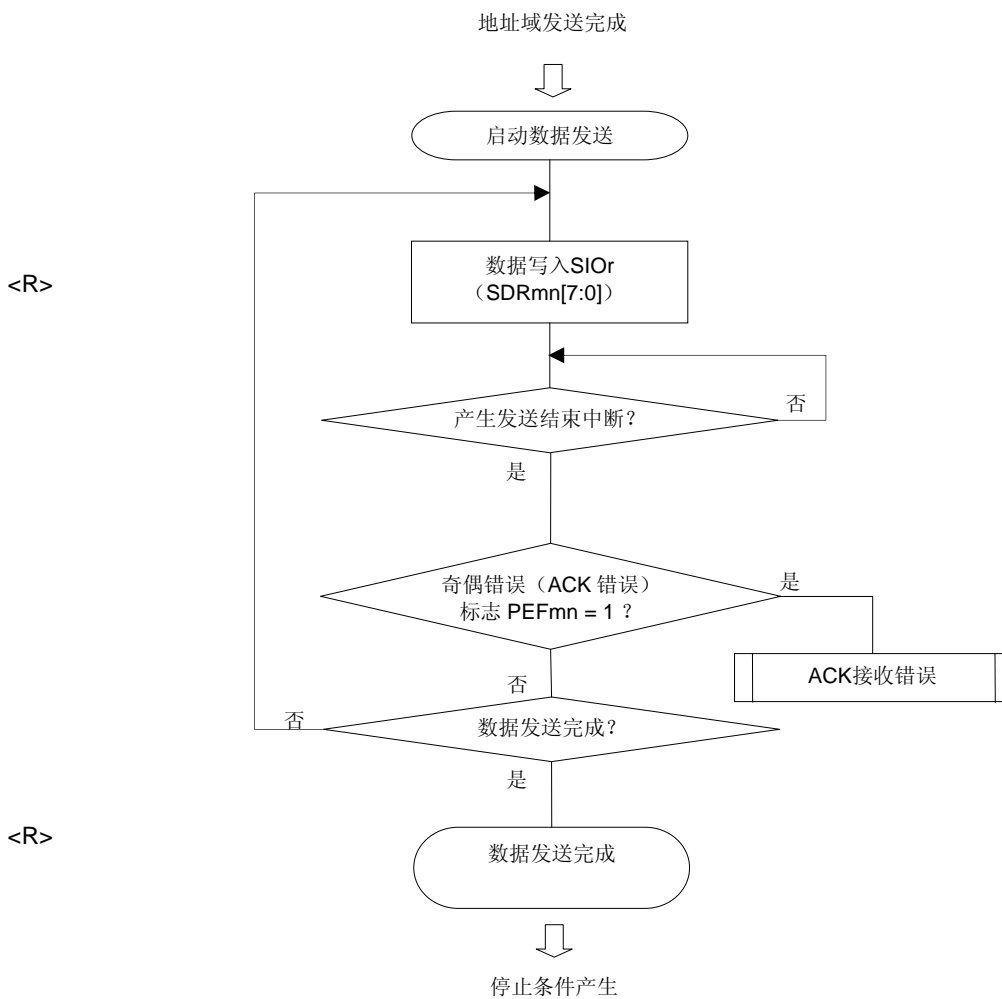
(2) 处理流程

图 12-93. 数据发送的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)

图 12-94. 数据发送的流程图



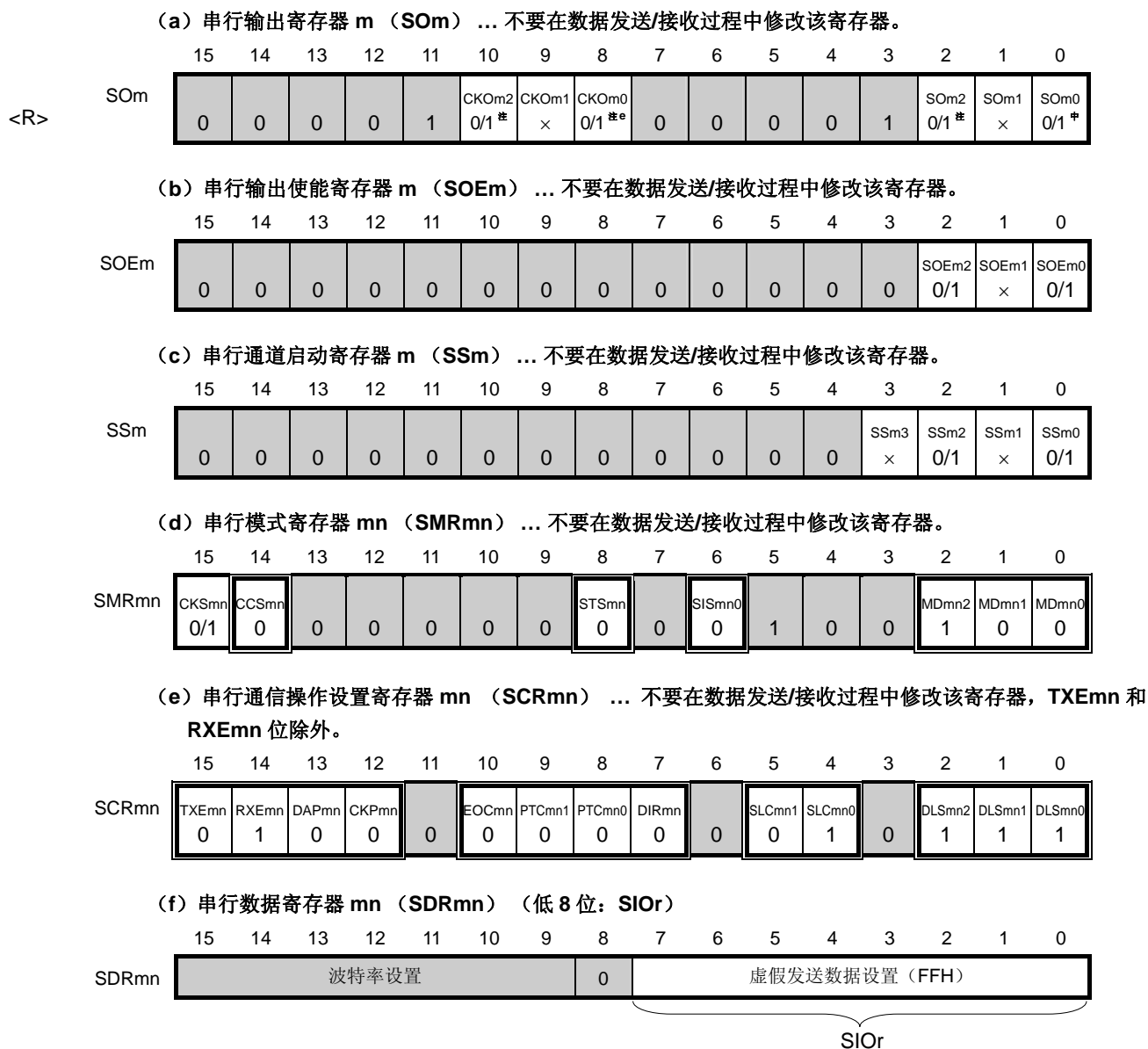
12.7.3 数据接收

数据接收是一个在地址域发送结束后将数据接收到发送目标（从机）的操作。在所有数据被接收到从机后，产生一个停止条件，并且释放总线。

简易I ² C	IIC10	IIC20
目标通道	SAU0的通道2	SAU1的通道0
使用的引脚	SCL10, SDA10	SCL20, SDA20
中断	INTIIC10	INTIIC20
	只有发送结束中断（禁止设置缓冲区为空中断）	
错误检测标志	无	
发送数据长度	8位	
发送速率	最大 $f_{CLK}/4$ MHz f_{CLK} : 系统时钟频率 但是，在I ² C的每种模式下，必须满足以下条件。 <ul style="list-style-type: none"> • 最大400 kHz（优先模式） • 最大100 kHz（标准模式） 	
数据电平	正向输出（默认：高电平）	
奇偶校验位	无奇偶校验位	
停止位	附加1位（用于ACK接收时序）	
数据方向	MSB 先行	

(1) 寄存器设置

图 12-95. 简易 I²C (IIC10, IIC20) 的数据接收的寄存器内容举例



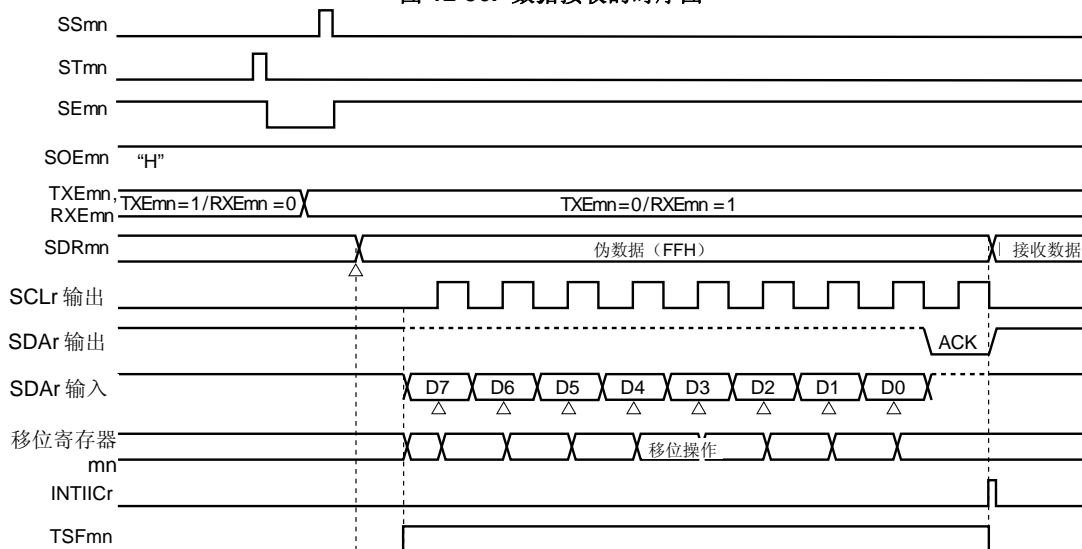
<R> 注 该值由通信操作过程中的通信数据决定。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)
: 在 IIC 模式下设置固定, : 设置禁止 (置为初始值)
 ×: 该模式下不能使用的位 (在任何模式下都未使用, 则置为初始值)
 0/1: 根据用户的使用设置为 0 或 1

(2) 处理流程

<R>

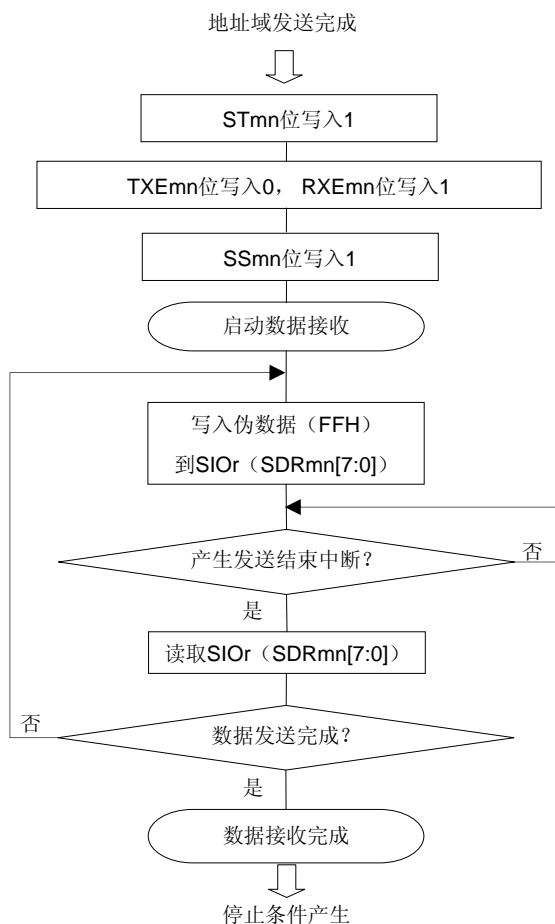
图 12-96. 数据接收的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)

<R>

图 12-97. 数据接收的流程图



<R>

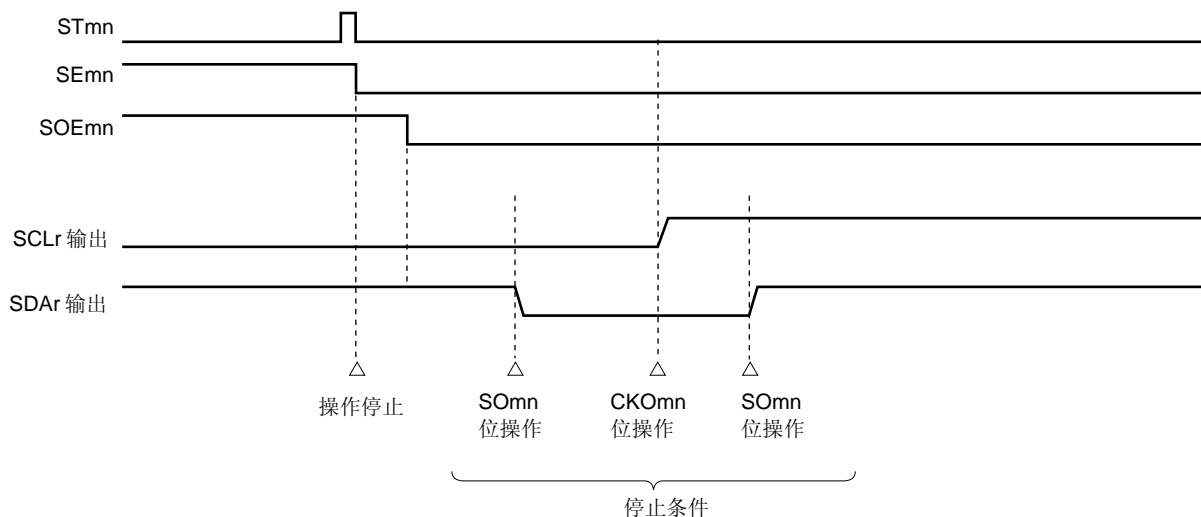
注意事项 当接收最后一个数据时, 会输出 ACK。通过设置“1”到 STmn 位, 停止操作并且产生一个停止条件, 完成通信。

12.7.4 产生停止条件

在所有数据被发送或者被接收后，产生一个停止条件，并且释放总线。

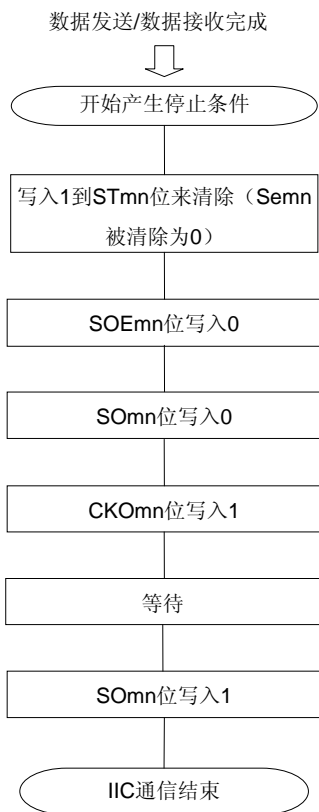
(1) 处理流程

图 12-98. 停止条件产生的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), r: IIC 编号 (r = 10, 20)

图 12-99. 停止条件产生的流程图



保留一段等待时间，这样可以满足从机的I²C 规范。

<R>

12.7.5 计算发送速率

简易 I²C (IIC10, IIC20) 通信的发送速率可以按照下列表达式来计算。

$$\text{(发送速率)} = \{ \text{目标通道的操作时钟 (MCK) 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2$$

- 备注**
1. SDRmn[15:9]的值是 SDRmn 寄存器的位 15 到 9 的值 (0000000B 到 1111111B)，因此它的取值为 0 到 127。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2)

操作时钟 (MCK) 由串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的第 15 位 (CKSmn) 来确定。

表 12-4. 操作时钟选择

SMRmn 寄存器	SPSm 寄存器								操作时钟 (MCK) ^{注1}	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
								INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{注2}		
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
								INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{注2}		
除上面以外									禁止设置	

<R>

<R>

注 1. 当更改 f_{CLK} 使用的时钟时（通过更改系统时钟控制寄存器（CKC）的值），在串行阵列单元（SAU）的操作停止（STm = 000FH）后再进行。当选择 INTTM02 和 INTTM03 作为操作时钟时，也要停止定时器阵列单元（TAU）（TT0 = 00FFH）。

3. 通过设置 TAU 的 TIS0 寄存器的 TIS02（如果 m = 0）和 TIS03（如果 m = 1）位为 1、选择 f_{SUB}/4 作为输入时钟，并使用 SPSm 寄存器选择 INTTM02 和 INTTM03，SAU 可以在副系统时钟固定分频情况下工作，而与 f_{CLK} 的频率（主系统时钟、副系统时钟）无关。但是，当更改 f_{CLK} 时，按照上面注 1 的描述，SAU 和 TAU 必须停止。

备注 1. X: 和此位无关

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2)

这是在 $MCK = f_{CLK} = 20 \text{ MHz}$ 时，设置 IIC 发送速率的一个示例。

IIC发送模式 (期望的发送速率)	$f_{CLK} = 20 \text{ MHz}$			
	操作时钟 (MCK)	SDRmn[15:9]	计算得到的发送速率	与期望的发送速率的误差
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

12.8 在错误情况下的处理过程

图 12-100 ~ 12-102 中描述了发生每种类型的错误时需要遵循的处理过程。

图 12-100. 在奇偶校验错误或溢出错误情况下的处理过程

软件操作	硬件状态	备注
读取SDRmn寄存器。	►BFF = 0, 并且允许通道n接收数据。	这用来防止溢出错误, 如果在错误处理过程中完成下一个接收。
读取SSRmn寄存器。		识别错误类型, 并且用读取获得的值来清除错误标志。
写入SIRmn寄存器。	►错误标志被清除。	将从SSRmn寄存器中读出的值不经修改直接写入SIRmn寄存器, 错误只能在读取过程中被清除。

图 12-101. 在帧错误情况下的处理过程

软件操作	硬件状态	备注
读取SDRmn寄存器。	►BFF = 0, 并且允许通道n接收数据。	这用来防止溢出错误, 如果在错误处理过程中完成下一个接收。
读取SSRmn寄存器。		识别错误类型, 并且用读取获得的值来清除错误标志。
写入SIRmn寄存器。	►错误标志被清除。	将从SSRmn寄存器中读出的值不经修改直接写入SIRmn寄存器, 错误只能在读取过程中被清除。
设置STmn位为1。	►SEmn = 0, 并且通道n停止操作。	
与其他通信方同步		重新与其他通信方同步, 并且恢复通信, 因为认为由于起始位错位导致帧错误发生。
设置SSmn位为1。	►SEmn = 1, 并且允许通道n操作。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2)

图 12-102. 在简易 I²C 模式下奇偶校验错误 (ACK 错误) 的情况下的处理过程

软件操作	硬件状态	备注
读取SDRmn寄存器。	► BFF = 0, 并且允许通道n接收数据。	这用来防止溢出错误, 如果在错误处理过程中完成下一个接收。
读取SSRmn寄存器。		识别错误类型, 并且用读取获得的值来清除错误标志。
写入SIRmn寄存器。	► 错误标志被清除。	将从SSRmn寄存器中读出的值不经修改直接写入SIRmn寄存器, 错误只能在读取过程中被清除。
设置STMn位为1。	► SEMn = 0, 并且通道n停止操作。	从机没有准备好接收, 因为没有返回ACK。因此, 创建一个停止条件, 释放总线, 并且再次从启动条件开始通信。或者, 产生一个重启条件, 并且发送过程可以有地址发送开始重新完成。
创建停止条件。		
创建启动条件。		
设置SSmn位为1。	► SEMn = 1, 并且允许通道n操作。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10

<R> 12.9 寄存器设置和引脚之间的关系

表 12-5 ~ 表 12-12 是串行阵列单元 0 和 1 的每个通道的寄存器设置和引脚之间的关系。

表 12-5. 寄存器设置和引脚之间的关系 (单元 0 的通道 0: CSI00, UART0 发送)

SE 00 注1	MD 002	MD 001	SOE 00	SO 00	CKO 00	TXE 00	RXE 00	PM 10	P10	PM 11 注2	P11 注2	PM 12	P12	工作模式	引脚功能		
															SCK00/ P10	SI00/ RxD0/P11 注2	SO00/ TxD0/P12
0	0	0	0	1	1	0	0	× 注3	× 注3	× 注3	× 注3	× 注3	× 注3	操作停止模式	P10	P11	P12
																P11/RxD0	
1	0	0	0	1	1	0	1	1	×	1	×	× 注3	× 注3	CSI00 从机接收	$\overline{\text{SCK00}}$ (输入)	SI00	P12
														CSI00 从机发送	$\overline{\text{SCK00}}$ (输入)	P11	SO00
														CSI00 从机发送/ 接收	$\overline{\text{SCK00}}$ (输入)	SI00	SO00
														CSI00 主机接收	$\overline{\text{SCK00}}$ (输出)	SI00	P12
														CSI00 主机发送	$\overline{\text{SCK00}}$ (输出)	P11	SO00
														CSI00 主机发送/ 接收	$\overline{\text{SCK00}}$ (输出)	SI00	SO00
														UART0 发送 ^{注5}	P10	P11/RxD0	TxD0

- 注
1. SE0 寄存器是只读状态寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 1 被设置为 UART0 接收时，该引脚变为 RxD0 功能引脚（参阅表 12-6）。在这种情况下，必须为单元 0 的通道 0 选择操作停止模式或 UART0 发送模式。
 3. 该引脚可以被设置为端口功能引脚。
 4. 根据通信操作，可以是 0 或 1。关于细节，参阅 12.3 (12) 串行输出寄存器 m (SOM)。
 5. 当成对使用 UART0 发送和接收时，设置单元 0 的通道 1 为 UART0 接收（参阅表 12-6）。

备注 X: 和此位无关

表 12-6. 寄存器设置和引脚之间的关系 (单元 0 的通道 1: CSI01, UART0 接收)

SE 01 注1	MD 012	MD 011	SOE 01	SO01	CKO 01	TXE 01	RXE 01	PM 43	P43	PM44	P44	PM 45	P45	PM 11 注2	P11 注2	工作模式	引脚功能			
																	SCK01/ P43	SI01/ P44	SO01/ P45	SI00/ RxD0/ P11注2
0	0	0	0	1	1	0	0	×	×	×	×	×	×	×	×	操作停止 模式	P43	P44	P45	SI00/ P11
	0	1						注3	注3	注3	注3	注3	注3	注3	注3					
1	0	0	0	1	1	0	1	1	×	1	×	×	×	×	×	CSI01 从 机接收	SCK01 (输入)	SI01	P45	SI00/ P11
			1	0/1 注4	1	1	0	1	×	×	×	0	1	×	×	CSI01 从 机发送	SCK01 (输入)	P44	SO01	SI00/ P11
			1	0/1 注4	1	1	1	1	×	1	×	0	1	×	×	CSI01 从 机发送/接 收	SCK01 (输入)	SI01	SO01	SI00/ P11
			0	1	0/1 注4	0	1	0	1	1	×	×	×	×	×	CSI01 主 机接收	SCK01 (输出)	SI01	P45	SI00/ P11
			1	0/1 注4	0/1 注4	1	0	0	1	×	×	0	1	×	×	CSI01 主 机发送	SCK01 (输出)	P44	SO01	SI00/ P11
			1	0/1 注4	0/1 注4	1	1	0	1	1	×	0	1	×	×	CSI01 主 机发送/接 收	SCK01 (输出)	SI01	SO01	SI00/ P11
0	1	0	1	1	0	1	×	×	×	×	×	×	1	×	UART0 接收注5, 6	P43	P44	P45	RxD0	

- 注
- SE0 寄存器是只读状态寄存器，使用 SS0 和 ST0 寄存器来设置它。
 - 当单元 0 的通道 1 被设置为 UART0 接收时，该引脚变为 RxD0 功能引脚。在这种情况下，必须为单元 0 的通道 0 选择操作停止模式或 UART0 发送（参阅表 12-5）。
当单元 0 的通道 0 被设置为 CSI00 时，该引脚不能被用作 RxD0 功能引脚。在这种情况下，设置单元 0 的通道 1 为操作停止模式或 CSI01。
 - 该引脚可以被设置为端口功能引脚。
 - 根据通信操作，可以是 0 或 1。关于细节，参阅 12.3 (12) 串行输出寄存器 m (S0m)。
 - 当成对使用 UART0 发送和接收时，设置单元 0 的通道 0 为 UART0 发送（参阅表 12-5）。
 - 在 UART0 接收过程中，必须设置单元 0 的通道 0 的 SMR00 寄存器。关于细节，参阅 12.5.2 (1) 寄存器设置。

备注 X: 和此位无关

表 12-7. 寄存器设置和引脚之间的关系（单元 0 的通道 2: CSI10, UART1 发送, IIC10）

SE 02 注1	MD 022	MD 021	SOE 02	SO 02	CKO 02	TXE 02	RXE 02	PM 04	P04	PM03 注2	P03 注2	PM02	P02	工作模式	引脚功能				
															SCK10/ SCL10/P04	SI10/SDA10/ RxD1/P03注2	SO10/ TxD1/P02		
0	0	0	0	1	1	0	0	× 注3	× 注3	× 注3	× 注3	× 注3	× 注3	操作停止模式	P04	P03	P02		
	0	1														P03/RxD1			
	1	0														P03			
1	0	0	0	1	1	0	1	1	×	1	×	× 注3	× 注3	CSI10 从机接收	SCK10 (输入)	SI10	P02		
			1	0/1 注4	1	1	0	1	×	×	注3	注3	0	1	CSI10 从机发送	SCK10 (输入)	P03	SO10	
			1	0/1 注4	1	1	1	1	×	1	×	0	1	CSI10 从机发送/接收	SCK10 (输入)	SI10	SO10		
			0	1	0/1 注4	0	1	0	1	1	×	× 注3	× 注3	CSI10 主机接收	SCK10 (输出)	SI10	P02		
			1	0/1 注4	0/1 注4	1	0	0	1	×	× 注3	× 注3	0	1	CSI10 主机发送	SCK10 (输出)	P03	SO10	
			1	0/1 注4	0/1 注4	1	1	0	1	1	×	0	1	CSI10 主机发送/接收	SCK10 (输出)	SI10	SO10		
	0	1	1	0/1 注4	1	1	0	×	× 注3	× 注3	× 注3	× 注3	0	1	UART1 发送注5	P04	P03/RxD1	TxD1	
0	1	0	0	0/1 注6	0/1 注6	0	0	0	1	0	1	× 注3	× 注3	IIC10 启动条件	SCL10	SDA10	P02		
						1	0												
						0	1												
			1	1	0/1 注4	0/1 注4	1	0	0	1	0	1	×	× 注3	× 注3	IIC10 地址域发送	SCL10	SDA10	P02
				1	0/1 注4	0/1 注4	1	0	0	1	0	1	×	× 注3	× 注3	IIC10 数据发送	SCL10	SDA10	P02
0	1	0	0	0/1 注7	0/1 注7	0	0	0	1	0	1	× 注3	× 注3	IIC10 停止条件	SCL10	SDA10	P02		
						1	0												
						0	1												

- 注
1. SE0 寄存器是只读状态寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 3 被设置为 UART1 接收时，该引脚变为 RxD1 功能引脚（参阅表 12-8）。在这种情况下，必须为单元 0 的通道 2 选择操作停止模式或 UART1 发送。
 3. 该引脚可以被设置为端口功能引脚。
 4. 根据通信操作，可以是 0 或 1。关于细节，参阅 12.3 (12) 串行输出寄存器 m (SOm)。
 5. 当成对使用 UART1 发送和接收时，设置单元 0 的通道 3 为 UART1 接收（参阅表 12-8）。
 6. 在产生启动条件前，设置 CKO02 位为 1。当产生启动条件时，将 SO02 位从 1 清除到 0。
 7. 在产生停止条件前，设置 CKO02 位为 1。当产生停止条件时，将 SO02 位从 0 设置为 1。

备注 X: 和此位无关

表 12-8. 寄存器设置和引脚之间的关系 (单元 0 的通道 3: UART1 接收)

SE03 ^{注1}	MD032	MD031	TXE03	RXE03	PM03 ^{注2}	P03 ^{注2}	工作模式	引脚功能
								SI10/SDA10/RxD1/P03 ^{注2}
0	0	1	0	0	× ^{注3}	× ^{注3}	操作停止模式	SI10/SDA10/P03 ^{注2}
1	0	1	0	1	1	×	UART1 接收 ^{注4,5}	RxD1

- 注**
1. SE0 寄存器是只读状态寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 3 被设置为 UART1 接收时，该引脚变为 RxD1 功能引脚。在这种情况下，必须为单元 0 的通道 2 选择操作停止模式或 UART1 发送（参阅表 12-7）。
当单元 0 的通道 2 被设置为 CSI10 或 IIC10 时，该引脚不能被用作 RxD1 功能引脚。在这种情况下，设置单元 0 的通道 3 为操作停止模式。
 3. 该引脚可以被设置为端口功能引脚。
 4. 当成对使用 UART1 发送和接收时，设置单元 0 的通道 2 为 UART1 发送（参阅表 12-7）。
 5. 在 UART1 接收过程中，必须设置单元 0 的通道 2 的 SMR02 寄存器。关于细节，参阅 12.5.2 (1) 寄存器设置。

备注 X: 和此位无关

表 12-9. 寄存器设置和引脚之间的关系（单元 1 的通道 0: CSI20, UART2 发送, IIC20）

SE 10 注1	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 142	P142	PM 143 注2	P143 注2	PM 144	P144	工作模式	引脚功能		
															SCK20/ SCL20/P14 2	SI20/SDA20/ RxD2/P143 注2	SO20/ TxD2/ P144
0	0	0	0	1	1	0	0	× 注3	× 注3	× 注3	× 注3	× 注3	× 注3	操作停止模式	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	×	1	×	× 注3	× 注3	CSI20 从机接收	SCK20 (输入)	SI20	P144
			1	0/1 注4	1	1	0	1	×	× 注3	× 注3	0	1	CSI20 从机发送	SCK20 (输入)	P143	SO20
			1	0/1 注4	1	1	1	1	×	1	×	0	1	CSI20 从机发送/接收	SCK20 (输入)	SI20	SO20
			0	1	0/1 注4	0	1	0	1	1	×	× 注3	× 注3	CSI20 主机接收	SCK20 (输出)	SI20	P144
			1	0/1 注4	0/1 注4	1	0	0	1	× 注3	× 注3	0	1	CSI20 主机发送	SCK20 (输出)	P143	SO20
			1	0/1 注4	0/1 注4	1	1	0	1	1	×	0	1	CSI20 主机发送/接收	SCK20 (输出)	SI20	SO20
	0	1	1	0/1 注4	1	1	0	× 注3	× 注3	× 注3	× 注3	0	1	UART2 发送 注5	P142	P143/RxD2	TxD2
0	1	0	0	0/1 注6	0/1 注6	0	0	0	1	0	1	× 注3	× 注3	IIC20 启动条件	SCL20	SDA20	P144
1						0											
0						1											
1			1	0/1 注4	0/1 注4	1	0	0	1	0	1	× 注3	× 注3	IIC20 地址域发送	SCL20	SDA20	P144
1	1	0	1	0/1 注4	0/1 注4	1	0	0	1	0	1	× 注3	× 注3	IIC20 数据 发送	SCL20	SDA20	P144
			1	0/1 注4	0/1 注4	0	1	0	1	0	1	× 注3	× 注3	IIC20 数据 接收	SCL20	SDA20	P144
0	1	0	0	0/1 注7	0/1 注7	0	0	0	1	0	1	× 注3	× 注3	IIC20 停止条件	SCL20	SDA20	P144
						1	0										
						0	1										

- 注
- SE1 寄存器是只读状态寄存器，使用 SS1 和 ST1 寄存器来设置它。
 - 当单元 1 的通道 1 被设置为 UART2 接收时，该引脚变为 RxD2 功能引脚（参阅表 12-10）。在这种情况下，必须为单元 1 的通道 0 选择操作停止模式或 UART2 发送。
 - 该引脚可以被设置为端口功能引脚。
 - 根据通信操作，可以是 0 或 1。关于细节，参阅 12.3 (12) 串行输出寄存器 m (SOm)。
 - 当成对使用 UART2 发送和接收时，设置单元 1 的通道 1 为 UART2 接收（参阅表 12-10）。
 - 在产生启动条件前，设置 CKO10 位为 1。当产生启动条件时，将 SO10 位从 1 清除到 0。
 - 在产生停止条件前，设置 CKO10 位为 1。当产生停止条件时，将 SO10 位从 0 设置为 1。

备注 X: 和此位无关

表 12-10. 寄存器设置和引脚之间的关系 (单元 1 的通道 1: CSI21, UART2 接收, IIC21)

SE11 ^{注1}	MD112	MD111	TXE11	RXE11	PM143 ^{注2}	P143 ^{注2}	工作模式	引脚功能
								SI20/SDA20/RxD2/P143 ^{注2}
0	0	1	0	0	×	×	操作停止模式	SI20/SDA20/P143
1	0	1	0	1	1	×	UART2 接收 ^{注4,5}	RxD2

- 注**
1. SE1 寄存器是只读状态寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 当单元 1 的通道 1 被设置为 UART2 接收时，该引脚变为 RxD2 功能引脚。在这种情况下，必须设置单元 1 的通道 0 为操作停止模式或 UART2 发送（参阅表 12-9）。
当单元 1 的通道 0 被设置为 CSI20 或 IIC20 时，该引脚不能被用作 RxD2 功能引脚。在这种情况下，设置单元 1 的通道 2 为操作停止模式。
 3. 该引脚可以被设置为端口功能引脚。
 4. 当成对使用 UART2 发送和接收时，设置单元 1 的通道 0 为 UART2 发送（参阅表 12-9）。
 5. 在 UART2 接收过程中，单元 1 的通道 0 的 SMR10 寄存器也必须被设置。关于细节，参阅 12.5.2 (1) 寄存器设置。

备注 ×: 和此位无关

表 12-11. 寄存器设置和引脚之间的关系（单元 1 的通道 2: UART3 发送）

SE12 注1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13	P13	工作模式	引脚功能
										TxD3/P13
0	0	1	0	1	0	0	×	×	操作停止模式	P13
1	0	1	1	0/1 注3	1	0	0	1	UART3 发送注4	TxD3

- 注
1. SE1 寄存器是只读状态寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 该引脚可以被设置为端口功能引脚。
 3. 根据通信操作，可以是 0 或 1。关于细节，参阅 12.3（12） 串行输出寄存器 m（S0m）。
 4. 当成对使用 UART3 发送和接收时，设置单元 1 的通道 3 为 UART3 接收（参阅表 12-12）。

备注 X: 和此位无关

表 12-12. 寄存器设置和引脚之间的关系（单元 1 的通道 3: UART3 接收）

SE13 注1	MD132	MD131	TXE13	RXE13	PM14	P14	工作模式	引脚功能
								RxD3/P14
0	0	1	0	0	×	×	操作停止模式	P14
1	0	1	0	1	1	×	UART3 接收注3, 4	RxD3

- 注
1. SE1 寄存器是只读状态寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 该引脚可以被设置为端口功能引脚。
 3. 当成对使用 UART3 发送和接收时，设置单元 1 的通道 2 为 UART3 发送（参阅表 12-11）
 4. 在 UART3 接收过程中，单元 1 的通道 2 的 SMR12 寄存器也必须被设置。关于细节，参阅 12.5.2（1） 寄存器设置。

备注 X: 和此位无关

第十三章 串行接口 IIC0

13.1 串行接口 IIC0 的功能

串行接口 IIC0 有如下两种模式。

(1) 操作停止模式

在不进行串行传送时，可采用此模式，以降低功耗。

(2) I²C 总线模式（支持多个主设备）

该模式采用两条线（一条串行时钟（SCL0）线和一条串行数据总线（SDA0）），在多个设备之间的进行 8 位数据传送。

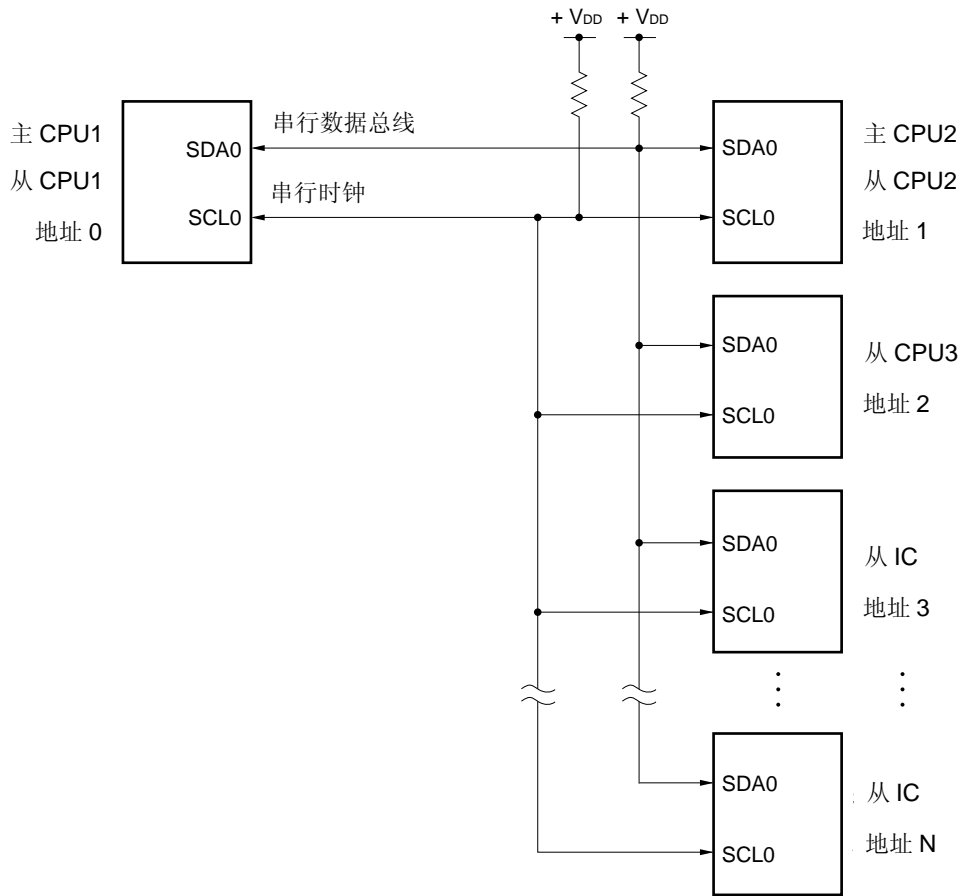
该模式遵循 I²C 总线格式并且主设备能够通过串行数据总线，产生“起始条件”，“地址”，“发送方向规范”，“数据”和“停止条件”等信号传送至从设备。从设备能够通过硬件自动检测这些接收到的状态和数据。这个功能可以简化控制 I²C 总线的应用程序。

由于 SCL0 和 SDA0 引脚是漏极开路输出，IIC0 总线的串行时钟线和串行数据总线需要连接上拉电阻。

图 13-1 所示为串行接口 IIC0 的框图。

图 13-2 显示了串行总线配置示例。

图 13-2. 使用 I²C 总线的串行总线配置示例



13.2 串行接口 IIC0 的配置

串行接口 IIC0 包括如下硬件

表 13-1. 串行接口 IIC0 的配置

项目	配置
寄存器	IIC 移位寄存器 0 (IIC0) 从设备地址寄存器 0 (SVA0)
控制寄存器	外围设备允许寄存器 0 (PER0) IIC 控制寄存器 0 (IICCO) IIC 状态寄存器 0 (IICSO) IIC 标志寄存器 0 (IICFO) IIC 时钟选择寄存器 0 (IICCL0) IIC 功能扩展寄存器 0 (IICX0) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

(1) IIC 移位寄存器 0 (IIC0)

IIC0 用于 8 位串行数据和 8 位并行数据的相互转换，并且和串行时钟同步。IIC0 可以用来发送和接收数据。

对 IIC0 的读写入操作可以控制实际的发送和接收操作。

在等待期间对 IIC0 写入数据，可以取消等待状态并启动数据传送操作。

IIC0 可以通过 8 位存储器操作指令设置。

复位信号的产生将 IIC0 清零 (00H)。

图 13-3. IIC 移位寄存器 0 的格式 (IIC0)

地址: FFF50H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IIC0								

注意事项 1. 在数据传送期间不要对 IIC0 写入数据。

2. 仅在等待期间时才能对 IIC0 进行读写入操作。除等待期间外，禁止在其他的通信状态下访问 IIC0。然而，作为主设备时，在通信触发位 (STT0) 被设为 1 之后，IIC0 只能写入一次。

(2) 从设备地址寄存器 0 (SVA0)

该寄存器用于存储从设备模式下的本地地址。

SVA0 可以通过 8 位存储器操作指令设置。

但是当 STD0=1 时，禁止重写该寄存器（当检测到起始条件时）。

复位信号的产生将 SVA0 清零 (00H)。

图 13-4. 从地址寄存器 0 的格式 (SVA0)

地址: FFF53H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SVA0								0 ^注

注 第 0 位恒为 0。

(3) SO 锁存器

SO 锁存器用来保持 SDA0 引脚的输出电平。

(4) 唤醒控制器

当该寄存器接收的地址与从设备地址寄存器 0 (SVA0) 设置的地址相匹配, 或者当接收到一个扩展代码时, 这个电路将产生一个中断请求信号 (INTIIC0)。

(5) 预分频器

用于选择采样时钟。

(6) 串行时钟计数器

该计数器对发送/接收操作期间输出或输入的串行时钟进行计数, 同时用于验证 8 位数据是否已发送或接收。

(7) 中断请求信号发生器

此电路用于控制中断请求信号 (INTIIC0) 的产生。

一个 I²C 中断请求由下列两个触发条件产生:

- 串行时钟的第 8 或第 9 个下降沿 (通过 WTIMO 位设置)
- 当检测到停止条件 (通过 SPIE0 位设置) 时产生中断请求

备注 WTIMO 位: IIC 控制寄存器 0 (IICC0) 的第 3 位
SPIE0 位: IIC 控制寄存器 0 (IICC0) 的第 4 位

(8) 串行时钟控制器

在主设备模式下, 此电路通过 SCL0 引脚, 从采样时钟产生时钟输出。

(9) 串行时钟等待控制器

用于控制等待时序。

(10) ACK 信号发生器, 停止条件检测器, 起始条件检测器和 ACK 信号检测器

这些电路用于产生和检测每种状态。

(11) 数据保持时间修正电路

此电路根据串行时钟的下降沿产生数据的保持时间。

(12) 起始条件发生器

当 STT0 位为 1 时, 该电路产生一个起始条件信号。

但是, 在禁止通信预约的状态下 (IICRSV 位 = 1), 只要总线尚未被释放 (IICBSY 位 = 1), 起始条件请求将被忽略并且 STCF 被置为 1。

(13) 停止条件发生器

当 SPT0 位为 1 时, 此电路产生一个停止条件信号。

(14) 总线状态检测器

此电路通过检测起始条件和停止条件来确定总线是否被释放。

然而，如果操作后总线状态无法被立即检测到，则初始状态由 STCEN 位来设置。

备注	STT0 位:	IIC 控制寄存器 0 (IICC0) 的第 1 位
	SPT0 位:	IIC 控制寄存器 0 (IICC0) 的第 0 位
	IICRSV 位:	IIC 标志寄存器 0 (IICF0) 的第 0 位
	IICBSY 位:	IIC 标志寄存器 0 (IICF0) 的第 6 位
	STCF 位:	IIC 标志寄存器 0 (IICF0) 的第 7 位
	STCEN 位:	IIC 标志寄存器 0 (IICF0) 的第 1 位

13.3 控制串行接口 IIC0 的寄存器

串行接口 IIC0 由下列 8 个寄存器控制。

- 外围设备允许寄存器 0 (PER0)
- IIC 控制寄存器 0 (IICC0)
- IIC 标志寄存器 0 (IICF0)
- IIC 状态寄存器 0 (IICS0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

(1) 外围设备允许寄存器 0 (PER0)

PER0 用于允许或禁止每个外围硬件模块。当外围硬件不使用时，停止提供时钟，以降低功耗和噪声。

当使用串行接口 IIC0 时，此寄存器的第 4 位 (IIC0EN) 必须置为 1。

PER0 可以通过 1 位或 8 位存储器操作指令设置。

复位信号发生将此寄存器清为 00H。

注意事项 1. 设置串行接口 IIC0 时，IIC0EN 必须首先设置为 1。如果 IIC0EN = 0，写入串行接口 IIC0 控制寄存器被忽略，即使读取，只能读到默认值。

2. 确保 PER0 寄存器的第 1 位清 0。

图 13-5. 外围设备允许寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN
IIC0EN	控制串行接口 IIC0 输入时钟							
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 由串行接口 IIC0 使用的 SFR 不能写入。 • 串行接口 IIC0 在复位状态。 							
1	提供输入时钟。 <ul style="list-style-type: none"> • 由串行接口 IIC0 使用的 SFR 可以读/写。 							

(2) IIC 控制寄存器 0 (IICC0)

这个寄存器用于设置允许/停止 I²C 的操作、设置等待时间和 I²C 的其他操作。

IICC0 可以通过 1 位或 8 位存储器操作指令进行设置。但在 IICE0 位 = 0 或者在等待期间，需要设置 SPIE0、WTIM0 和 ACKE0 位。当 IICE0 位从“0”变为“1”时，这些位可以同时被设置。

复位信号的产生将 IICC0 清零 (00H)。

图 13-6. IIC 控制寄存器 0 (IICC0) 的格式 (1/4)

地址: FFF52H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I ² C 操作允许
0	停止操作。复位 IIC 状态寄存器 0 (IICS0) ^{注1} 。停止内部操作。
1	允许操作
当 SCL0 和 SDA0 线是高电平时, 必须将该位设置为 1。	
清零条件 (IICE0 = 0)	设置条件 (IICE0 = 1)
<ul style="list-style-type: none"> 通过指令清零 复位 	<ul style="list-style-type: none"> 通过指令设置

LRELO ^{注2}	退出通信
0	正常操作
1	此时从当前的通信模式退出, 进入待机模式。当执行此操作后, 设置的值自动清零。 适用于那些已接受了本地无关扩展码的情况。 SCL0 和 SDA0 线被设置为高阻抗状态。 下列 IIC 控制寄存器 0 (IICC0) 和 IIC 状态寄存器 0 (IICS0) 的标志位被清零。 •STT0 •SPT0 •MSTS0 •EXC0 •COI0 •TRC0 •ACKD0 •STD0
从通信模式退出后一直保持待机模式, 直到发生下列通信入口条件。	
<ul style="list-style-type: none"> 检测到停止条件后, 以主设备模式重新启动。 开始条件后, 接收到地址匹配或接收到扩展码。 	
清零条件 (LRELO = 0)	设置条件 (LRELO = 1)
<ul style="list-style-type: none"> 执行后自动清零 复位 	<ul style="list-style-type: none"> 通过指令设置

WRELO ^{注2}	等待取消
0	不取消等待
1	取消等待。在等待取消后, 此设置被自动清除。
在发送状态 (TRC0 = 1) 第 9 个时钟脉冲的等待期间设置 WRELO (等待取消) 时, SDA0 将进入高阻抗状态 (TRC0 = 0)。	
清零条件 (WRELO = 0)	设置条件 (WRELO = 1)
<ul style="list-style-type: none"> 执行后自动清零 复位 	<ul style="list-style-type: none"> 通过指令设置

- 注
1. 对 IICS0 寄存器、IICF0 寄存器的 STCF0 位与 IICBSY 位、和 IICCL0 寄存器的 CLD0 位与 DAD0 位进行复位。
 2. 当 IICE0 = 0 时, 本标志信号无效。

注意事项 允许 I²C 操作 (IICE0 = 1) 后, 在 SCL0 为高电平并且 SDA0 为低电平时, 立即检测到起始条件。允许 I²C 操作 (IICE0 = 1) 后, 立即通过 1 位存储器操作指令设置 LRELO (1)。

图 13-6. IIC 控制寄存器 0 (IICC0) 的格式 (2/4)

SPIE0 ^{注1}	当检测到停止条件时，允许/禁止产生中断请求	
0	禁止	
1	允许	
清零条件 (SPIE0 = 0)		设置条件 (SPIE0 = 1)
<ul style="list-style-type: none"> • 通过指令清零 • 复位 		<ul style="list-style-type: none"> • 通过指令进行设置

WTIMO ^{注1}	对等待和中断请求产生的控制	
0	在第 8 个时钟的下降沿产生中断请求。 主设备模式：输出 8 个时钟之后，将时钟输出设置为低电平，且设置等待状态。 从设备模式：输入 8 个时钟之后，将时钟设置为低电平，且为主设备设置等待状态。	
1	在第 9 个时钟的下降沿产生中断请求。 主设备模式：输出 9 个时钟之后，将时钟输出设置为低电平，且设置等待状态。 从设备模式：输入 9 个时钟之后，将时钟设置为低电平，且设置主设备等待状态。	
地址传送期间，在第 9 个时钟的下降沿产生中断，而与该位的设置无关。地址传送完成之后该位设置才有效。主设备模式下，在地址传送的第 9 个时钟下降沿插入一个等待信号。对于已接收到本地地址的从设备，在发出应答信号 (ACK) 之后的第 9 个时钟下降沿插入一个等待信号。但是，当从设备已经收到一个扩展码时，将在第 8 个时钟的下降沿插入等待信号。		
清零条件 (WTIMO = 0)		设置条件 (WTIMO = 1)
<ul style="list-style-type: none"> • 通过指令清零 • 复位 		<ul style="list-style-type: none"> • 通过指令进行设置

ACKE0 ^{注1, 2}	应答控制	
0	禁止应答	
1	允许应答。在第九个时钟周期，SDA0 设置为低电平。	
清零条件 (ACKE0 = 0)		设置条件 (ACKE0 = 1)
<ul style="list-style-type: none"> • 通过指令进行清零 • 复位 		<ul style="list-style-type: none"> • 通过指令进行设置

- 注
1. IICE0 = 0 时，此标志信号无效。
 2. 在地址传送期间，或者编码为非扩展码时，设置的值无效。
在从设备模式下并且地址匹配时，无论设置何值都将产生应答信号。

图 13-6. IIC 控制寄存器 0 (IICC0) 的格式 (3/4)

STT0 ^注	起始条件触发	
0	不产生起始条件	
1	总线释放时 (STOP 模式下) : 产生一个起始条件 (作为主设备启动)。SCL0 处于高电平时, SDA0 从高电平变为低电平时产生起始条件。然后, 经过固定的额定时间后, SCL0 变为低电平 (等待状态)。 有第三方通信时: <ul style="list-style-type: none"> 当允许通信预约时 (IICRSV = 0) 该位作为起始条件预约标志。设置为 1 时, 在总线释放之后自动产生一个起始条件。 当禁止通信预约时 (IICRSV = 1) 将 STCF 设置为 1, 且 STT0 被清零。不产生起始条件。 处于等待状态 (主设备模式) : 退出等待状态之后, 产生一个重复起始条件。	
设置时序的注意事项 <ul style="list-style-type: none"> 主设备接收: 传送期间不能设置为 1。仅在 ACKE0 被清零且从设备已经知道是最终接收时的等待期间, 才能设为 1。 主设备发送: 应答期间内, 不能正常产生起始条件。在第 9 个时钟输出后的等待期间可以设置为 1。 不能和 SPT0 同时设为 1。 在 STT0 被清零前设置 STT0 为 1 然后再次设置该位, 这种操作是被禁止的。 		
清零条件 (STT0 = 0)		设置条件 (STT0 = 1)
<ul style="list-style-type: none"> 禁止通信预约时, 通过将 SST0 设置为 1 进行清零 由仲裁失利清零 通过主设备产生起始条件后进行清零 由 LREL0=1 (退出通信) 清零 当 IICE0 = 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 通过指令进行设置

注 IICE0 = 0 时, 该标志信号无效。

备注

- 第 1 位 (STT0) 在数据设置后再读取时将会变为 0。
- IICRSV: IIC 标志寄存器 (IICF0) 的第 0 位。
STCF: IIC 标志寄存器 (IICF0) 的第 7 位。

图 13-6. IIC 控制寄存器 0 (IICC0) 的格式 (4/4)

SPT0	停止条件触发器				
0	不产生停止条件				
1	产生停止条件 (结束主设备传送) SDA0 变为低电平后, 设置 SCL0 为高电平或等待其变为高电平。然后, 经过固定的额定时间后, SDA0 从低电平变为高电平, 则产生一个停止条件。				
设置时序注意事项 <ul style="list-style-type: none"> 主设备接收: 传送期间不能设为 1。 仅在 ACKE0 被清零且从设备已经知道是最终接收时的等待期间, 才能设为 1。 主设备发送: 应答期间内, 不能正常产生停止条件。在第 9 个时钟输出后的等待期间可以设为 1。 不能和 STT0 同时设置为 1 仅在主设备模式下, SPT0 才能设置为 1 WTIMO 清零后, 如果在输出 8 个时钟之后的等待期间将 SPT0 设为 1, 那么注意, 在第 9 个时钟的高电平期间将产生一个停止条件。在输出 8 个时钟后的等待期间 WTIMO 应该从 0 变为 1, 并且在输出第 9 个时钟后的等待期间 SPT0 应被设置为 1。 禁止在 SPT0 被清零前重复设置为 1。 					
<table border="1" style="width: 100%;"> <thead> <tr> <th>清零条件 (SPT0 = 0)</th> <th>设置条件 (SPT0 = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> 由仲裁失利清零 检测到停止条件后自动清零 可由 LRELO=1 (退出通信) 清零 IICE0 = 0 时 (操作停止) 复位 </td> <td> <ul style="list-style-type: none"> 通过指令进行设置 </td> </tr> </tbody> </table>		清零条件 (SPT0 = 0)	设置条件 (SPT0 = 1)	<ul style="list-style-type: none"> 由仲裁失利清零 检测到停止条件后自动清零 可由 LRELO=1 (退出通信) 清零 IICE0 = 0 时 (操作停止) 复位 	<ul style="list-style-type: none"> 通过指令进行设置
清零条件 (SPT0 = 0)	设置条件 (SPT0 = 1)				
<ul style="list-style-type: none"> 由仲裁失利清零 检测到停止条件后自动清零 可由 LRELO=1 (退出通信) 清零 IICE0 = 0 时 (操作停止) 复位 	<ul style="list-style-type: none"> 通过指令进行设置 				

注 仅在主设备模式下, SPT0 可以设置为 1。但是, 必须将 SPT0 设置为 1, 而且在切换到允许操作状态后检测到第 1 个停止条件之前产生一个停止条件, 。

注意事项 当 IIC 状态寄存器 0 (IICS0) 的第 3 位 (TRC0) 被设为 1, WRELO 在第 9 个时钟设置为 1, 并取消等待, 然后 TRC0 被清零, SDA0 线被设为高阻抗状态。

备注 第 0 位 (SPT0) 在数据设置后再次读取, 将会变为 0。

(3) IIC 状态寄存器 0 (IICS0)

此寄存器用于表示 I²C 的状态。

仅当 STT0 = 1 且处于等待期间，才能通过 1 位或 8 位存储器操作指令读取 IICS0。

复位信号的产生将 IICS0 清零 (00H)。

图 13-7. IIC 状态寄存器 0 (IICS0) 的格式 (IICS0) (1/3)

地址: FFF56H 复位后: 00H R

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	主设备状态	
0	从设备状态或通信待机状态	
1	主设备通信状态	
清零条件 (MSTS0 = 0)		设置条件 (MSTS0 = 1)
<ul style="list-style-type: none"> 检测到停止条件时 ALD0 = 1 时 (仲裁失利) 由 LREL0 = 1 (退出通信) 清零 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 起始条件产生时
ALD0	仲裁失利的检测	
0	此状态表示没有仲裁, 或仲裁结果为“获胜”。	
1	此状态表示仲裁结果为“失利”。MSTS0 被清零。	
清零条件 (ALD0 = 0)		设置条件 (ALD0 = 1)
<ul style="list-style-type: none"> 读取 IICS0 之后, 自动清零[※] IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 仲裁结果为“失败”时。
EXC0	扩展码接收的检测	
0	未接收到扩展码	
1	接收到扩展码	
清零条件 (EXC0 = 0)		设置条件 (EXC0 = 1)
<ul style="list-style-type: none"> 检测到起始条件时 检测到停止条件时 由 LREL0 = 1 (退出通信) 清零 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 当接收到的地址数据的高 4 位为“0000”或“1111”时 (在第 8 个时钟的上升沿设置)

注 当使用 1 位存储器操作指令对 IICS0 之外的某位进行操作, 此寄存器也会清零。因此当使用 ALD0 位时, 应先读取该位数据。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

图 13-7. IIC 状态寄存器 0 (IICS0) 的格式 (2/3)

COI0	地址匹配的检测	
0	地址不匹配	
1	地址匹配	
清零条件 (COI0 = 0)		设置条件 (COI0 = 1)
<ul style="list-style-type: none"> • 检测到起始条件时 • 检测到停止条件时 • 由 LREL0 = 1 (退出通信) 清零 • IICE0 从 1 变为 0 (操作停止) 时 • 复位 		<ul style="list-style-type: none"> • 当接收地址和本地地址 (从设备地址寄存器 0 (SVA0) 的内容) 匹配时 (在第 8 个时钟的上升沿设置)

TRC0	发送/接收状态的检测	
0	接收状态 (非发送状态)。SDA0 线被设为高阻抗状态。	
1	发送状态。允许将 SO0 锁存器中的内容输出到 SDA0 线 (从第 1 个字节第 9 个时钟的下降沿开始才有效)。	
清零条件 (TRC0 = 0)		设置条件 (TRC0 = 1)
<主设备模式和从设备模式> <ul style="list-style-type: none"> • 检测到停止条件时 • 由 LREL0 = 1 (退出通信) 清零 • IICE0 从 1 变为 0 (操作停止) 时 • 由 WREL0 = 1* (取消等待) 清零 • ALD0 从 0 变为 1 (仲裁失利) 时 • 复位 <主设备模式> <ul style="list-style-type: none"> • 当“1”输出到第 1 个字节的 LSB (发送方向规范位) 时 <从设备模式> <ul style="list-style-type: none"> • 检测到起始条件时 • 当“0”输入到第 1 个字节的 LSB (发送方向规范位) 时 <未用于通信时>		<主设备模式> <ul style="list-style-type: none"> • 产生起始条件时 • 当“0”输出到第 1 个字节的 LSB (发送方向规范位) 时 <从设备模式> <ul style="list-style-type: none"> • 当“1”输入到第 1 个字节的 LSB (发送方向规范位) 时

注 当 IIC 状态寄存器 0 (IICS0) 的第 3 位 (TRC0) 为 1 时, 通过在第 9 个时钟将 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WREL0) 设置为 1, 来取消等待状态, TRC0 被清零且 SDA0 线进入高阻抗状态。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位
 IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

图 13-7. IIC 状态寄存器 0 (IICS0) 的格式 (3/3)

ACKD0	应答信号的检测 ($\overline{\text{ACK}}$)	
0	未检测到应答信号	
1	已检测到应答信号	
清零条件 (ACKD0 = 0)		设置条件 (ACKD0 = 1)
<ul style="list-style-type: none"> 检测到停止条件时 在下一个字节的第一个时钟上升沿 由 LREL0 = 1 (退出通信) 清零 当 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 在 SCL0 的第 9 个时钟的上升沿将 SDA0 线设为低电平之后
STD0	起始条件的检测	
0	未检测到起始条件	
1	已检测到起始条件。这表示正处于地址传送阶段。	
清零条件 (STD0 = 0)		设置条件 (STD0 = 1)
<ul style="list-style-type: none"> 检测到停止条件时 在地址发送之后的下一个字节第 1 个时钟的上升沿 由 LREL0 = 1 (退出通信) 清零 当 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 检测到起始条件时
SPD0	停止条件的检测	
0	未检测到停止条件	
1	已检测到停止条件。主设备通信结束，释放总线。	
清零条件 (SPD0 = 0)		设置条件 (SPD0 = 1)
<ul style="list-style-type: none"> 在设置该位后的地址发送字节第 1 个时钟上升沿，并检测到起始条件 当 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 检测到停止条件时

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位

IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(4) IIC 标志寄存器 0 (IICF0)

此寄存器用于设置 I²C 的操作模式，并指示 I²C 总线的状态。

IICF0 可以通过 1 位或 8 位的存储器操作指令进行设置。但是，STCF 和 IICBSY 位是只读的。

IICRSV 位用于允许/禁止通信预约功能。

STCEN 用于设置 IICBSY 的初始值。

仅当 I²C 操作被禁止 (IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) = 0) 时，才可以对 IICRSV 和 STCEN 位进行写入操作。当允许操作时，可以读取 IICF0 寄存器。

复位信号的产生将 IICF0 变为 00H。

图 13-8. IIC 标志寄存器 0 (IICF0) 的格式

地址: FFF51H 复位后: 00H R/W^注

符号	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	清零标志
0	产生起始条件
1	起始条件的产生未成功: 清零 STT0 标志
清零条件 (STCF=0)	清零条件 (STCF=1)
<ul style="list-style-type: none"> • STT0=1 清零 • 当 IICE0=0 (操作停止) • 复位 	<ul style="list-style-type: none"> • 当通讯预约禁止时, 发生起始条件不成功并且 STT0 清零 (IICRSV=1)

IICBSY	总线状态标志
0	总线释放状态 (通讯初始化状态当 STCEN=1)
1	总线通讯状态 (通讯初始化状态当 STCEN=0)
清零条件 (IICBSY = 0)	清零条件 (IICBSY = 1)
<ul style="list-style-type: none"> • 检测到停止条件时 • 当 IICE0 =0 (操作停止) • 复位 	<ul style="list-style-type: none"> • 检测到停止条件时 • 当 STCEN = 0 时 IICE0 的设置

STCEN	初始化起始使能触发
0	操作允许后 (IICE0=1), 检测到停止条件时允许产生起始条件。
1	操作允许后 (IICE0=1), 允许产生起始条件, 而无需检测到停止条件。
清零条件 (STCEN =0)	清零条件 (STCEN =1)
<ul style="list-style-type: none"> • 检测到起始条件时 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

IICRSV	通讯预约功能禁止位
0	允许通讯预约
1	禁止通讯预约
清零条件 (IICRSV =0)	清零条件 (IICRSV =1)
<ul style="list-style-type: none"> • 通过指令清零 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

注 第 6 位和第 7 位是只读的。

- 注意事项**
1. 仅当操作停止 (IICE0 = 0) 时, 才可以对 STCEN 进行写入操作。
 2. 当 STCEN = 1 时不论总线处于何种状态, 总线释放状态 (IICBSY = 0) 都可以识别, 因此当产生第一个起始条件 (STT0 = 1) 时, 必须确保没有第三方通信, 以防该通信被破坏。
 3. 仅当操作停止 (IICE0 = 0) 时, 才可以对 IICRSV 进行写入操作。

备注 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(5) IIC 时钟选择寄存器 0 (IICCL0)

此寄存器用来为 I²C 总线设置传送时钟。

IICCL0 可以通过 1 位或 8 位的存储器操作指令设置。但是, CLD0 和 DAD0 位是只读的。SMC0、CL01、CL00 位和 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 组合设置。(详见 13.5.4 I²C 传送时钟设置方法)

IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 0 时, 设置 IICCL0。

复位信号的产生将 IICCL0 设置为 00H。

图 13-9. IIC 时钟选择寄存器 0 (IICCL0) 的格式

地址: FFF54H 复位后: 00H R/W^注

符号	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	SCL0 引脚电平的检测 (仅当 IICE0 = 1 时有效)
0	SCL0 引脚检测为低电平。
1	SCL0 引脚检测为高电平。
清零条件 (CLD0 = 0)	
设置条件 (CLD0 = 1)	
<ul style="list-style-type: none"> • SCL0 引脚为低电平时 • IICE0 = 0 (操作停止) 时 • 复位 	<ul style="list-style-type: none"> • SCL0 引脚为高电平时

DAD0	SDA0 引脚电平的检测 (仅当 IICE0 = 1 时有效)
0	SDA0 引脚检测为低电平。
1	SDA0 引脚检测为高电平。
清零条件 (DAD0 = 0)	
设置条件 (DAD0 = 1)	
<ul style="list-style-type: none"> • SDA0 引脚为低电平时 • IICE0 = 0 (操作停止) 时 • 复位 	<ul style="list-style-type: none"> • SDA0 引脚为高电平时

SMC0	操作模式切换
0	运行于标准模式
1	运行于高速模式

DFC0	数字滤波器操作控制
0	关闭数字滤波器
1	打开数字滤波器
数字滤波器只能用于高速模式 在高速模式下, 不论 DFC0 位设置 (1) / 清零 (0), 传送时钟不会发生变化。 数字滤波器用于高速模式下的噪声消除。	

注 第 4 位和第 5 位是只读的。

备注 IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(6) IIC 功能扩展寄存器 0 (IICX0)

此寄存器用于设置 I²C 的功能扩展。

IICX0 可以通过 1 位或 8 位存储器操作指令设置。CLX0 位和 IIC 时钟选择寄存器 0 (IICCL0) 的第 3、1 和 0 位 (SMC0、CL01 和 CL00) 组合设置 (详见 13.5.4 I²C 传送时钟设置方法)。

IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 0 时, 设置 IICX0。

复位信号将 IICX0 清零 (00H)。

图 13-10. IIC 功能扩展寄存器 0 (IICX0) 的格式

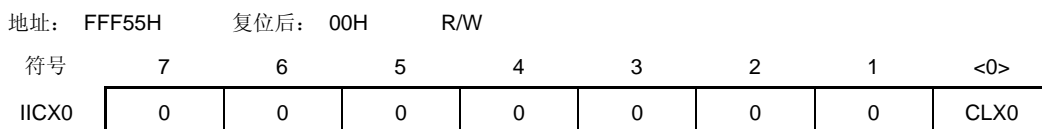


表 13-2. 选择时钟设置

IICX0 第 0 位 CLX0	IICCL0			传送时钟 (f _{CLK} /m)	可设置的选择时钟 (f _{CLK}) 范围	操作模式
	第 3 位 SMC0	第 1 位 CL01	第 0 位 CL00			
0	0	0	0	f _{CLK} /88	4.00 MHz~8.38 MHz	普通模式 (SMC0 位= 0)
0	0	0	1	f _{CLK} /172	8.38 MHz~16.76 MHz	
0	0	1	0	f _{CLK} /344	16.76 MHz~20 MHz	
0	0	1	1	f _{CLK} /44	2.00 MHz~4.19 MHz	
0	1	0	×	f _{CLK} /48	8.00 MHz~16.76 MHz	高速模式 (SMC0 位= 1)
0	1	1	0	f _{CLK} /96	16.00 MHz~20 MHz	
0	1	1	1	f _{CLK} /24	4.00 MHz~8.38 MHz	
1	0	×	×	禁止设置		
1	1	0	×	f _{CLK} /48	8.00 MHz~8.38 MHz	高速模式 (SMC0 位= 1)
1	1	1	0	禁止设置	16.00 MHz~16.76 MHz	
1	1	1	1	f _{CLK} /24	4.00 MHz~4.19 MHz	

注意事项 允许操作 (设置 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 1) 前, 使用 CLX0、SMC0、CL01 和 CL00 决定 I²C 的传送时钟频率。若改变传送时钟频率, 必须先把 IICE0 清零。

- 备注**
1. ×: 不必考虑
 2. f_{CLK}: CPU/外围硬件时钟频率

(7) 端口模式寄存器 6 (PM6)

此寄存器可以按位设置端口 6 的输入/输出操作模式。

当 P60/SCL0 引脚用于时钟 I/O，同时将 P61/SDA0 引脚用于串行数据 I/O 时，将 PM60 和 PM61 清零，且将 P60 和 P61 的输出锁存器清零。

在设置输出模式之前应将 IICE0 (IIC 控制寄存器 0 (IICCO) 的第 7 位) 设为 1，这是因为当 IICE0 为 0 时，P60/SCL0 和 P61/SDA0 引脚输出低电平 (恒定)。

PM6 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将 PM6 设置为 FFH。

图 13-11. 端口模式寄存器 6 (PM6) 的格式

地址: FFF26H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	P6n 引脚 I/O 模式选择 (n=0~7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

13.4 I²C总线模式功能

13.4.1 引脚配置

串行时钟引脚（SCL0）和串行数据总线引脚（SDA0）配置如下。

(1) SCL0.... 此引脚用于串行时钟的输入和输出。

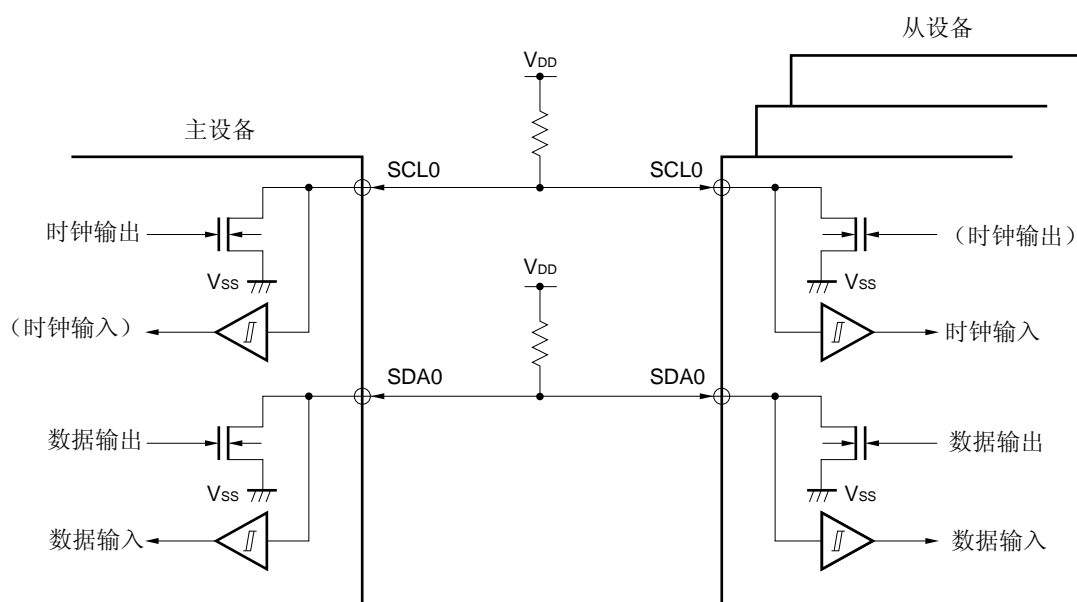
此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特（Schmitt）输入。

(2) SDA0.... 此引脚用于串行数据的输入和输出。

此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特（Schmitt）输入。

由于串行时钟线和串行数据总线的输出都是 N-ch 漏极开路输出，因此需要一个外部上拉电阻。

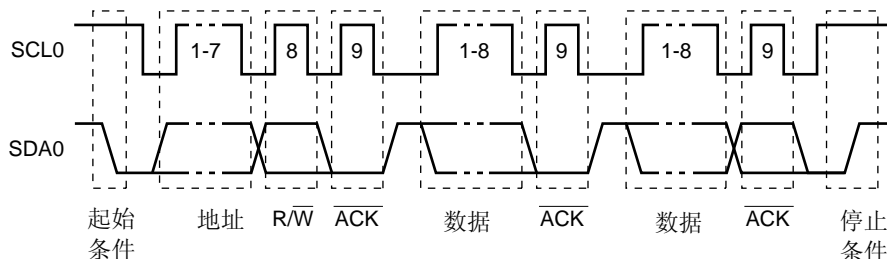
图 13-12. 引脚配置框图



13.5 I²C总线定义和控制方法

以下描述了 I²C 总线的串行数据通信格式和 I²C 总线使用的信号。图 13-13 显示了通过 I²C 串行数据总线发送“起始条件”，“地址”，“数据”和“停止条件”等信号时的传送时序。

图 13-13. I²C 总线串行数据传送时序



主设备产生起始条件、从设备地址和停止条件。

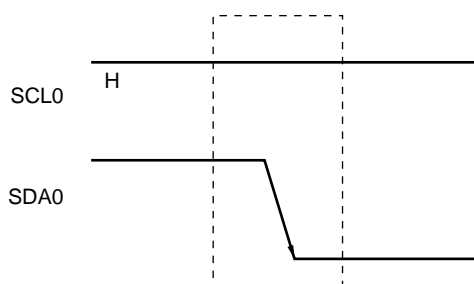
应答信号 ($\overline{\text{ACK}}$) 可以由主设备或者从设备产生 (通常情况下, 由接收 8 位数据的设备输出)。

主设备持续输出串行时钟 (SCL0)。但在从设备中, 可以延长 SCL0 的低电平周期插入等待时间。

13.5.1 起始条件

SCL0 引脚处于高电平, 且 SDA0 引脚从高电平变为低电平时, 产生一个起始条件。当启动串行传送时, SCL0 引脚和 SDA0 引脚的起始条件是由主设备产生并发送给从设备的信号。当使用从设备时, 可以检测到起始条件。

图 13-14. 起始条件



检测到停止条件 (SPD0: IIC 状态寄存器 0 (IICS0) 的第 0 位=1) 后, 当 IIC 控制寄存器 0 (IICC0) 的第 1 位 (STT0) 被设置 (为 1) 时, 输出一个起始条件。检测到起始条件时, IICS0 的第 1 位 (STD0) 被设置 (为 1)。

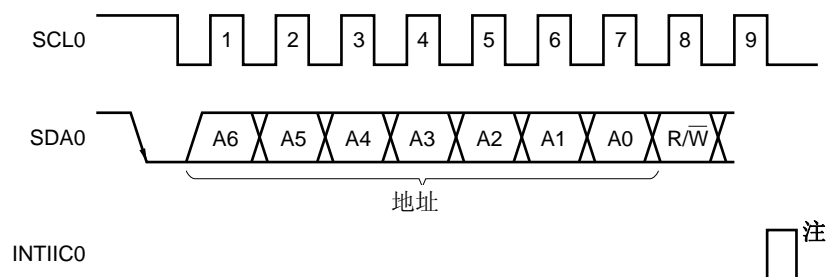
13.5.2 地址

地址由起始条件之后的 7 位数据指定。

一个地址是一个输出的 7 位数据段，用于选择一个通过总线与主设备相连的从设备。因此，相连到总线上的每个从设备必须有一个唯一的地址。

从设备包括具有以下功能的硬件：检测起始条件，检查 7 位地址数据是否与存储在从设备地址寄存器 0（SVA0）中的数据匹配。如果地址数据和 SVA0 的值匹配，则选中该从设备与主设备进行通信，直到主设备产生一个起始条件或停止条件。

图 13-15. 地址



注 如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则不会产生 INTIIC0。

将从设备地址和第 8 位（用于指示传送方向（如 13.5.3 发送方向规范中所描述的））一起写入 IIC 移位寄存器 0（IIC0）并且输出。接收到的地址写入 IIC0。

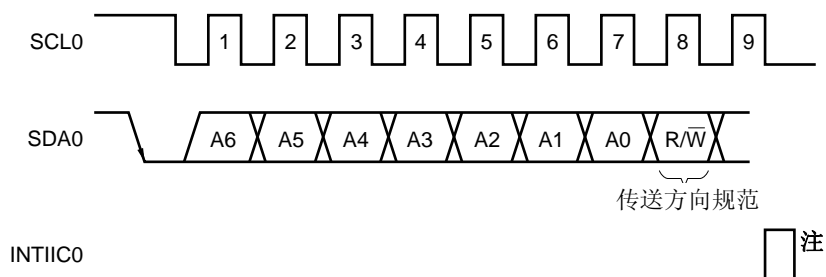
将从设备地址分配给 IIC0 的高 7 位。

13.5.3 发送方向规范

除了 7 位地址数据之外，主设备还发送 1 位数据用于指示传送方向。

当发送方向规范位的值为“0”时，表示主设备正向从设备传送数据。当传送方向指示位的值为“1”时，表示主设备正接收来自从设备的数据。

图 13-16. 发送方向规范



注 如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则不会产生 INTIIC0。

13.5.4 发送时钟设置方法

(1) 主设备的选择时钟设置方法

I²C 发送时钟频率 (f_{SCL}) 可以通过下面的表达式计算得出。

$$f_{SCL} = 1 / (m \times T + t_R + t_F)$$

m = 24, 44, 48, 88, 96, 172, 344 (参见表 13-3 选择时钟设置)

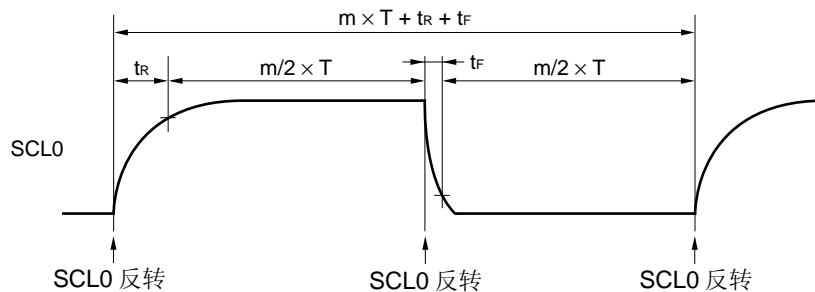
T: 1/f_{CLK}

t_R: SCL0 上升时间

t_F: SCL0 设置时间

例如, 当 f_{CLK} = 4.19 MHz, m = 88, t_R = 200 ns, 和 t_F = 50 ns 时 I²C 发送时钟频率 (f_{SCL}) 使用下面的公式计算。

$$f_{SCL} = 1 / (88 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 47.0 \text{ kHz}$$



通过 IIC 时钟选择寄存器 0 (IICCL0) 的第 3, 1, 和 0 位 (SMC0, CL01, 和 CL00) 和 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 联合设置选择时钟。

(2) 从设备端的选择时钟设置方法

要使用从设备, 根据 f_{CLK} (可设的选择时钟范围) 和 IIC 操作模式 (普通或高速), 按照表 13-3. 选择时钟设置中的定义, 通过 IIC 时钟选择寄存器 0 (IICCL0) 的第 3, 1, 和 0 位 (SMC0, CL01, 和 CL00) 和 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 联合设置选择时钟。

表 13-3. 选择时钟设置

IICX0 第 0 位	IICCL0			发送时钟 (f _{CLK} /m)	可设置的选择时钟 (f _{CLK}) 范围	操作模式
	第 3 位	第 1 位	第 0 位			
CLX0	SMC0	CL01	CL00			
0	0	0	0	f _{CLK} /88	4.00 MHz~8.38 MHz	普通模式 (SMC0 位 = 0)
0	0	0	1	f _{CLK} /172	8.38 MHz~16.76 MHz	
0	0	1	0	f _{CLK} /344	16.76 MHz~20 MHz	
0	0	1	1	f _{CLK} /44	2.00 MHz~4.19 MHz	
0	1	0	×	f _{CLK} /48	8.00 MHz~16.76 MHz	高速模式 (SMC0 位 = 1)
0	1	1	0	f _{CLK} /96	16.00 MHz~20 MHz	
0	1	1	1	f _{CLK} /24	4.00 MHz~8.38 MHz	
1	0	×	×	禁止设置		
1	1	0	×	f _{CLK} /48	8.00 MHz~8.38 MHz	高速模式 (SMC0 位 = 1)
1	1	1	0	禁止设置		
1	1	1	1	f _{CLK} /24	4.00 MHz~4.19 MHz	

注意事项 允许操作（设置 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 1) 前，使用 CLX0、SMC0、CL01 和 CL00 决定 I²C 的传送时钟频率。若要改变传送时钟频率，必须先把 IICE0 清零。

备注

1. ×: 不必考虑
2. f_{CLK}: CPU/外围硬件时钟频率

13.5.5 应答信号 (ACK)

ACK 信号用于检测发送端和接收端的串行数据的状态。

当接收方收到 8 位数据后将返回一个 ACK 信号。

发送方通常在发送 8 位数据之后接收到 ACK 信号。当 ACK 信号从接收方返回时，认为已正确接收并继续之后的处理。是否检测到 ACK 信号可以通过 IIC 状态寄存器 0 (IICS0) 的第 2 位 (ACKD0) 进行检测。

主设备接收到最后一个数据项时，不再返回 ACK 信号而是产生一个停止条件。如果从设备在接收到数据后未返回 ACK 信号，主设备将输出一个停止条件或重启条件并且停止传送。如果未返回 ACK 信号，可能是如下原因。

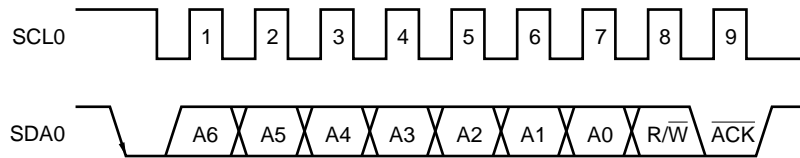
- <1> 接收操作没有正常进行。
- <2> 接收到最后一个数据项。
- <3> 地址指示的接收方不存在。

要产生 ACK 信号，接收方必须在第 9 个时钟时使 SDA0 线变为低电平（表示正常接收）。

可通过设置 IIC 控制寄存器 0 (IICC0) 的第 2 位 (ACKE0) 为 1，自动产生 ACK 信号。IICS0 寄存器的第 3 位 (TRC0) 可以由数据的第 8 位（紧接着 7 位地址信息）设置。通常，设置 ACEK0=1 用于接收 (TRC0 = 0)。

如果一个从设备在接收 (TRC0 = 0) 期间不再收到任何数据或者不再需要下一个数据项，则从设备必须通过将 ACEK0 清零通知主设备，这样它就不会再收到任何数据。

接收 (TRC0 = 0) 期间，当主设备不需要下一个数据项时，必须将 ACEK0 清零，这样就不会产生 ACK。通过这种方式，主设备通知作为发送方的从设备，不再需要其他数据（发送过程将停止）。

图 13-17. $\overline{\text{ACK}}$ 

接收到本地地址时，无论 ACKE0 的值是多少，都将自动产生 $\overline{\text{ACK}}$ 信号。当接收到一个非本地地址时，则不产生 $\overline{\text{ACK}}$ 信号（ NACK ）。

接收到扩展码时，如果 ACKE0 预先设为 1，则将产生 $\overline{\text{ACK}}$ 信号。

接收到数据时，如何产生 $\overline{\text{ACK}}$ 信号，将根据等待时序的设置而变化。

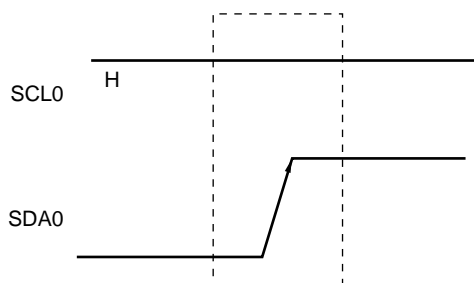
- 当选择 8 时钟等待状态时（ IICC0 寄存器的第 3 位（ WTIM0 ）= 0）：
退出等待状态前将 ACKE0 设为 1，在 SCL0 引脚第 8 个时钟的下降沿产生 $\overline{\text{ACK}}$ 信号。
- 当选择 9 时钟等待状态时（ IICC0 寄存器的第 3 位（ WTIM0 ）= 1）：
通过预先设置 ACKE0 为 1，产生 $\overline{\text{ACK}}$ 信号。

13.5.6 停止条件

SCL0 引脚处于高电平时，将 SDA0 引脚从低电平变为高电平，就可以产生一个停止条件。

停止条件是一个信号，它在串行传送操作结束时由主设备产生并发送给从设备。当作为从设备使用时，可以检测到停止条件。

图 13-18. 停止条件



IIC 控制寄存器 0 (IICC0) 的第 0 位 (SPT0) 设为 1 时，会产生一个停止条件。当检测到停止条件时，IIC 状态寄存器 0 (IICS0) 的第 0 位 (SPD0) 被设为 1，且当 IICC0 的第 4 位 (SPIE0) 设为 1 时产生 INTIIC0。

13.5.7 等待

等待状态用来通知通信另一方，设备（主设备或从设备）正准备发送或者接收数据（即处于等待状态）。

设置 SCL0 引脚为低电平，通知通信另一方设备已进入等待状态。当主设备和从设备的等待状态都被取消时，才能开始下一个数据传送。

图 13-19. 等待 (1/2)

(1) 当主设备有 9 个时钟等待，而从设备有 8 个时钟等待时（主设备发送，从设备接收，同时 ACKE0 = 1）

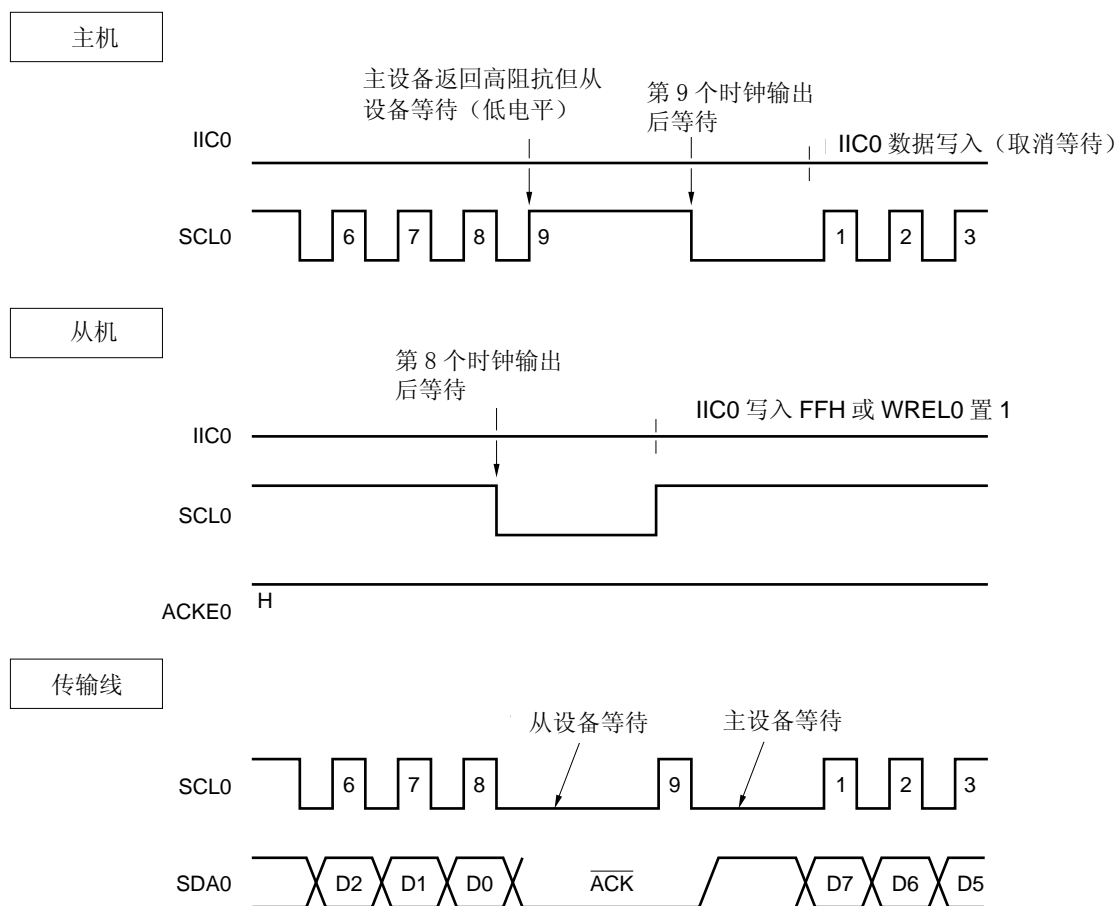
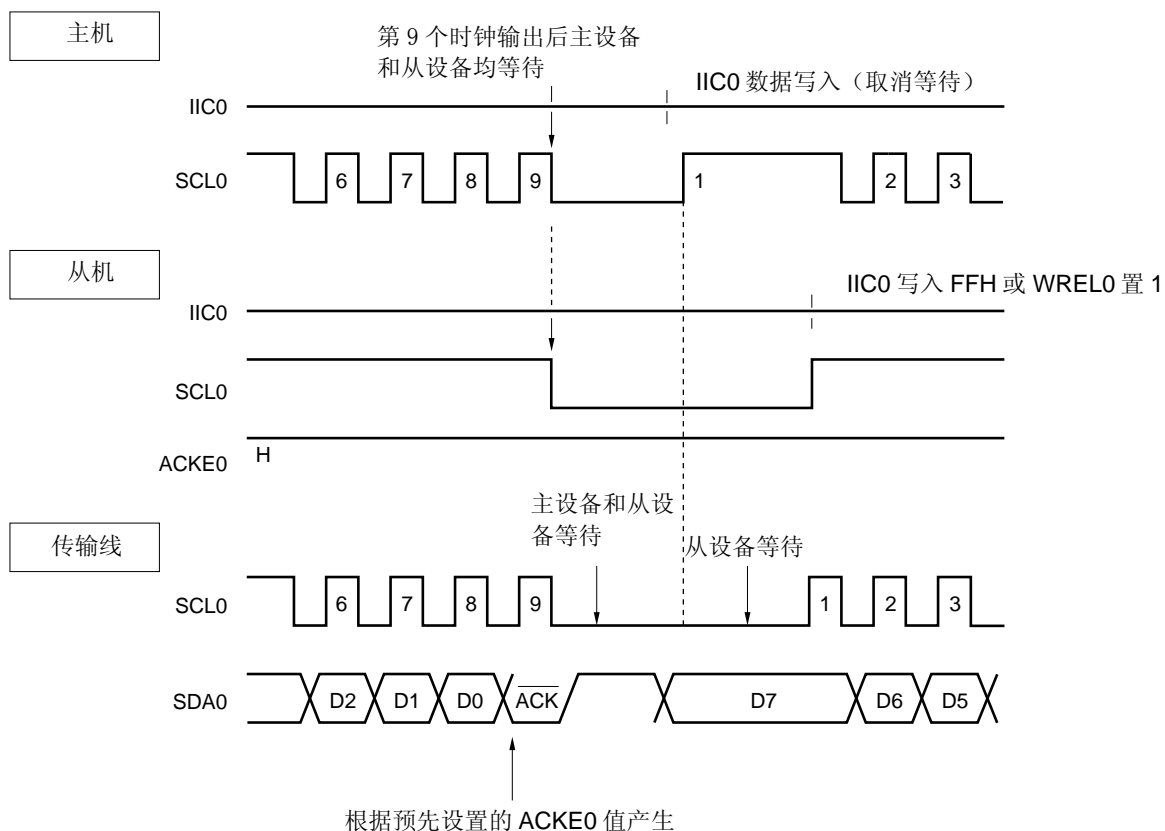


图 13-19. 等待 (2/2)

(2) 当主和从设备都有 9 个时钟后等待时 (主设备发送, 从设备接收, 同时 ACKE0 = 1)



备注 ACKE0: IIC 控制寄存器 0 (IICC0) 的第 2 位
WRELO: IIC 控制寄存器 0 (IICC0) 的第 5 位

根据 IIC 控制寄存器 0 (IICC0) 的第 3 位 (WTIM0) 的设置, 自动产生等待状态。

通常, 当 IICC0 的第 5 位 (WRELO) 设置为 1 时, 或者在将 FFH 写入 IIC 移位寄存器 0 (IIC0) 时, 接收方取消等待状态; 而当数据写入 IIC0 时, 发送方取消等待状态。

主设备也能通过以下任意一种方法取消等待状态。

- 把 IICC0 的第 1 位 (STT0) 设为 1
- 把 IICC0 的第 0 位 (SPT0) 设为 1

13.5.8 取消等待

I²C 通常用如下处理方式取消一个等待状态。

- 把数据写入 IIC 移位寄存器 0 (IIC0)
- 设置 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) (取消等待)
- 设置 IICC0 寄存器的第 1 位 (STT0) (产生起始条件)^注
- 设置 IICC0 寄存器的第 0 位 (SPT0) (产生停止条件)^注

注 仅用于主设备

当执行以上取消等待处理过程时，I²C 取消等待状态且恢复通信。

通过向 IIC0 写数据来取消等待状态并发送数据 (包括地址)。

通过设置 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) 为 1，可在取消等待状态之后接收数据，或完成数据发送操作。

设置 IICC0 的第 1 位 (STT0) 为 1，可在取消一个等待状态之后产生一个重启条件。

设置 IICC0 的第 0 位 (SPT0) 为 1，可在取消一个等待状态之后产生一个停止条件。

对于一个等待状态仅执行一次取消操作。

例如，如果设置 WRELO 为 1 在取消等待状态之后，将数据写入 IIC0，则可能一个不正确的值输出到 SDA0，这是因为改变 SDA0 线的时序和写入 IIC0 的时序相冲突。

除此以外，如果通信已放弃时，IICE0 被零，则停止通信，这样可以取消等待状态。

如果 I²C 总线由于噪音产生死锁，则通过设置 IICC0 的第 6 位 (LRELO) 可以保存通信处理，这样等待状态便可以取消。

13.5.9 中断请求 (INTIIC0) 产生时序和等待控制

IIC 控制寄存器 0 (IICC0) 第 3 位 (WTIMO) 的设置决定 INTIIC0 产生时序和相应的等待控制, 如表 13-4 所示。

表 13-4. INTIIC0 产生时序和等待控制

WTIMO	从设备操作期间			主设备操作期间		
	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. 仅当有地址与从设备地址寄存器 0 (SVA0) 设置的地址匹配时, 在第 9 个时钟的下降沿产生从设备的 INTIIC0 信号和等待周期。
此时, 不论 IICC0 的第 2 位 (ACKE0) 设置为何值, 都会产生 $\overline{\text{ACK}}$ 信号。对于接收到扩展码的从设备, 在第 8 个时钟的下降沿产生 INTIIC0。
但是, 如果重启之后地址不匹配, 将在第 9 个时钟的下降沿产生 INTIIC0, 但不会产生等待周期。
 2. 如果接收地址与从设备地址寄存器 0 (SVA0) 的内容不匹配, 同时也未接收到扩展码, 则 INTIIC0 和等待周期都不会产生。

备注 表中的数字是指串行时钟的时钟信号个数。中断请求和等待控制与这些时钟信号的下降沿同步。

(1) 地址发送/接收期间

- 从设备操作: 中断和等待时序取决于以上注 1、2 中所描述的情况, 与 WTIMO 位无关。
- 主设备操作: 中断和等待时序在第 9 个时钟的下降沿发生, 与 WTIMO 位无关。

(2) 数据接收期间

- 主/从设备操作: 中断和等待时序取决于 WTIMO 位。

(3) 数据发送期间

- 主/从设备操作: 中断和等待时序取决于 WTIMO 位。

(4) 等待取消的方法

四种取消等待方法如下所示。

- 向 IIC 移位寄存器 0 (IIC0) 写入数据
- 设置 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WRELO) (取消等待)
- 设置 IICC0 寄存器的第 1 位 (STT0) (产生起始条件)^注
- 设置 IICC0 寄存器的第 0 位 (SPT0) (产生停止条件)^注

注 仅用于主设备。

当选择等待 8 个时钟时 (WTIMO = 0), 必须在取消等待之前决定是否产生 $\overline{\text{ACK}}$ 信号。

(5) 停止条件的检测

当检测到一个停止条件时 (仅当 SPIE0 = 1), 将会产生 INTIIC0。

13.5.10 地址匹配的检测方法

在 I²C 总线模式下，主设备可以通过发送的从设备地址来选择指定的从设备。

可以通过硬件自动检测到地址匹配。当一个本地地址设置到从设备地址寄存器 0 (SVA0) 中且该地址与主设备发送的从设备地址匹配时，或者接收到扩展码时，都将产生一个中断请求 (INTIIC0)。

13.5.11 错误检测

I²C 总线模式下，在数据发送期间通过发送设备的 IIC 移位寄存器 (IIC0) 捕获串行数据总线 (SDA0) 的状态，因此发送前的 IIC0 数据可以用来和已发送的 IIC0 数据比较，以此来实现对发送错误的检测。当比较值不相等时，可以判断存在一个发送错误。

13.5.12 扩展码

(1) 当接收地址的高 4 位为“0000”或“1111”时，表示接收到扩展码的扩展码接收标志 (EXC0) 被设为 1，并且在第 8 个时钟的下降沿产生中断请求 (INTIIC0)。而存放在从设备地址寄存器 0 (SVA0) 中的本地地址不受影响。

(2) 如果由一个 10 位地址传送设置 SVA0 为“11110xx0”，且主设备传送的值为“11110xx0”，则其结果如下所示。注意在第 8 个时钟的下降沿产生 INTIIC0。

- 数据的高 4 位相等：EXC0 = 1
- 7 位数据相等：COI0 = 1

备注 EXC0: IIC 状态寄存器 0 (IICS0) 的第 5 位
 COI0: IIC 状态寄存器 0 (IICS0) 的第 4 位

(3) 由于中断请求发生之后的处理过程随着扩展码之后的数据而有所不同，因此这样的处理过程由软件来执行。

如果当一个从设备正在操作时接收到扩展码，那么该从设备（即使它的地址不相等）也将参与通信。

例如，接收到扩展码后，如果不希望目标设备作为从设备来操作，则将 IIC 控制寄存器 0 (IICC0) 的第 6 位 (LRELO) 设为 1，为下一个通信操作设置待机模式。

表 13-5. 扩展码位定义

从设备地址	R/W 位	描述
0 0 0 0 0 0 0	0	通用调用地址
0 0 0 0 0 0 0	1	起始字节
0 0 0 0 0 0 1	x	C-总线地址
0 0 0 0 0 1 0	x	为不同总线格式预留的地址
1 1 1 1 0 x x	x	10 位从设备地址规范

13.5.13 仲裁

当多个主设备同时产生起始条件信号时（在 STD0 设为 1 之前将 STT0 设为 1），调整时钟数目执行主设备间的通信，直至数据不再冲突。这种操作称为仲裁。

当有一个主设备仲裁失利时，可通过仲裁失利产生的时序设置 IIC 状态寄存器 0 (IICS0) 中的仲裁失利标志 (ALD0) 为 1，同时 SCL0 和 SDA0 线变为高阻抗，释放总线。

基于下一个中断请求的时序（第 8 或第 9 个时钟，检测到停止条件等）以及由软件设置 ALD0=1 来检测仲裁失利。

有关中断请求时序，详见 13.5.9 I²C 中断请求 (INTIIC0) 产生时序和等待控制。

备注 STD0: IIC 状态寄存器 0 (IICS0) 的第 1 位

STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位

图 13-20. 仲裁时序举例

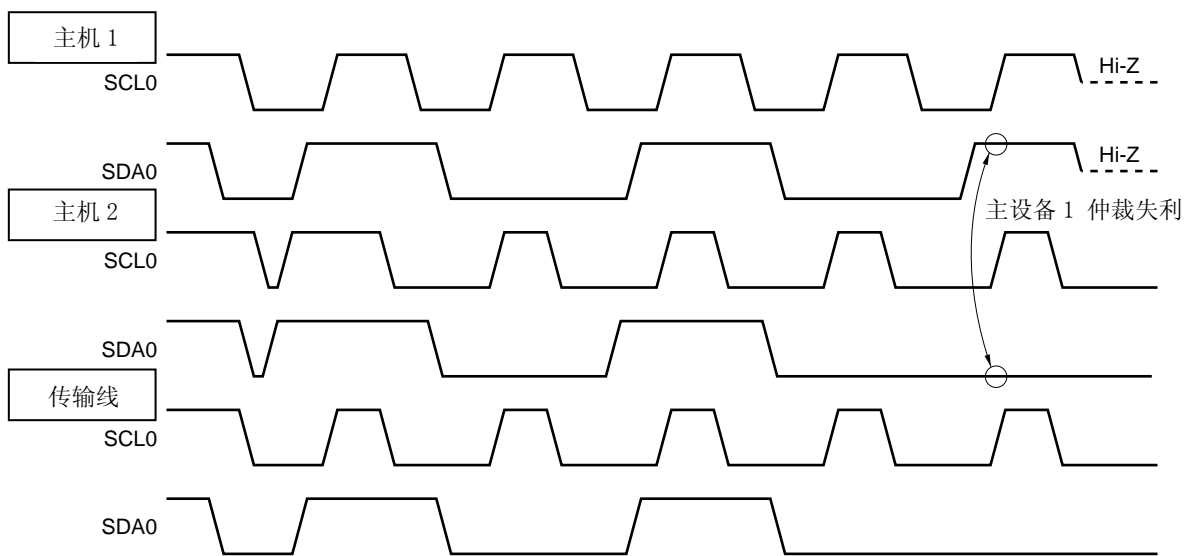


表 13-6. 仲裁期间的状态和中断请求产生时序

仲裁期间的状态	中断请求产生时序
地址发送期间	字节传送后第 8 或第 9 个时钟的下降沿 ^{注1}
地址发送后读/写数据	
扩展码发送期间	
扩展码发送后读/写数据	
数据发送期间	
数据发送之后的 \overline{ACK} 信号传送期间	
数据传送期间检测到重启条件时	
数据传送期间检测到停止条件时	停止条件产生时 (当 SPIE0 = 1) ^{注2}
准备产生重启条件时, 数据为低电平	字节传送后第 8 或第 9 个时钟的下降沿 ^{注1}
准备产生重启条件时, 检测到停止条件	停止条件产生时 (当 SPIE0 = 1) ^{注2}
准备产生停止条件时, 数据为低电平	字节传送后第 8 或第 9 个时钟的下降沿 ^{注1}
准备产生重启条件时, SCL0 为低电平	

- 注 1. 当 WTIMO (IIC 控制寄存器 0 (IICC0) 的第 3 位) = 1 时, 将在第 9 个时钟的下降沿产生一个中断请求。当 WTIMO = 0 且扩展码的从设备地址已被接收时, 将在第 8 个时钟的下降沿产生一个中断请求。
2. 如果可以产生仲裁, 则为主设备操作设置 SPIE0 = 1。

备注 SPIE0: IIC 控制寄存器 0 (IICC0) 的第 4 位

13.5.14 唤醒功能

I²C 总线从设备唤醒功能是, 在接收到本地地址及扩展码时产生一个中断请求信号 (INTIIC0)。

此功能通过地址不匹配时禁止产生不必要的 INTIIC0 信号, 使处理更加有效。

当检测到一个起始条件时, 设置唤醒待机模式。仲裁失利可能将主设备 (已产生起始条件) 转变为从设备, 如果因此而发送地址时, 则该唤醒待机模式有效。

可是, 当检测到停止条件时, 无论是否有唤醒功能, 都设置 IIC 控制寄存器 0 (IICC0) 的第 4 位 (SPIE0), 这将决定允许或禁止中断请求。

13.5.15 通信预约

(1) 允许使用通信预约功能时 (IIC 标志寄存器 0 (IICF0) 的第 0 位 (IICRSV) = 0)

如果要启动当前未连接在总线上的主设备进行通信, 可以使用通信预约功能在总线释放时允许发送一个起始条件信号。有两种无需使用总线的模式。

- 当仲裁结果不是主/从设备操作时
- 当接收到扩展码且从设备操作被禁止时 (不返回 $\overline{\text{ACK}}$ 信号, 且 IIC 控制寄存器 0 (IICC0) 的第 6 位 (LRELO) 被设置为 1 时, 释放总线)。

如果总线未使用时 (检测到停止条件后) IICC0 的第 1 位 (STT0) 被设为 1, 则自动产生一个起始条件并且设置等待状态。

如果在 IICC0 的第 4 位 (SPIE0) 设为 1 之后将地址写入 IIC 移位寄存器 0 (IIC0), 并且通过总线释放 (检测到停止状态) 产生中断请求信号 (INTIIC0) 来检测该地址, 则设备将自动以主设备方式开始通信。在检测到停止条件前写入 IIC0 的数据是无效的。

当 STT0 已被设为 1 时, 操作模式 (作为起始条件或作为通信预约) 取决于总线状态。

- 如果总线已被释放产生一个起始条件
- 如果总线未被释放 (待机模式)通信预约

STT0 设为 1 且经历了等待时间后, 使用 MSTS0 (IIC 状态寄存器 0 (IICS0) 的第 7 位) 检查是否进行通信预约操作。

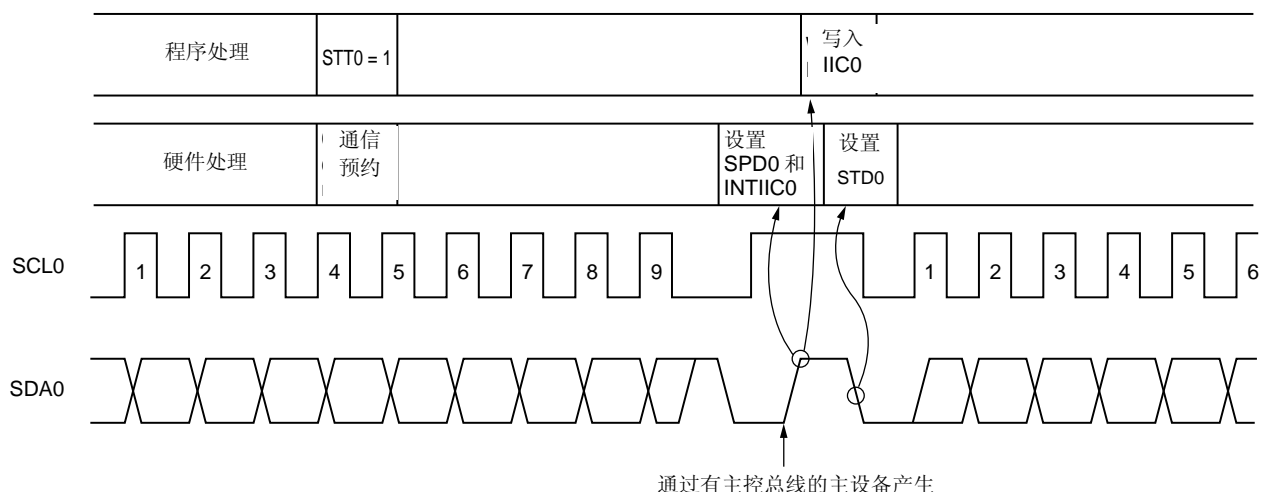
应通过软件设置的等待周期见表 13-7。

表 13-7. 等待周期

CLX0	SMC0	CL01	CL00	等待周期
0	0	0	0	43 个时钟
0	0	0	1	85 个时钟
0	0	1	0	101 个时钟
0	0	1	1	23 个时钟
0	1	0	0	27 个时钟
0	1	0	1	
0	1	1	0	51 个时钟
0	1	1	1	15 个时钟
1	1	0	0	
1	1	0	1	
1	1	1	0	27 个时钟
1	1	1	1	9 个时钟

图 13-21 显示通信预约时序。

图 13-21. 通信预约时序



备注

- IIC0: IIC 移位寄存器 0
- STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
- STD0: IIC 状态寄存器 0 (IICS0) 的第 1 位
- SPD0: IIC 状态寄存器 0 (IICS0) 的第 0 位

通过以下时序接受通信预约。IIC 状态寄存器 0 (IICS0) 的第 1 位 (STD0) 设为 1 之后，可以在检测到停止条件之前设置 IIC 控制寄存器 0 (IICC0) 的第 1 位 (STT0) 为 1，即可进行通信预约。

图 13-22. 接受通信预约的时序

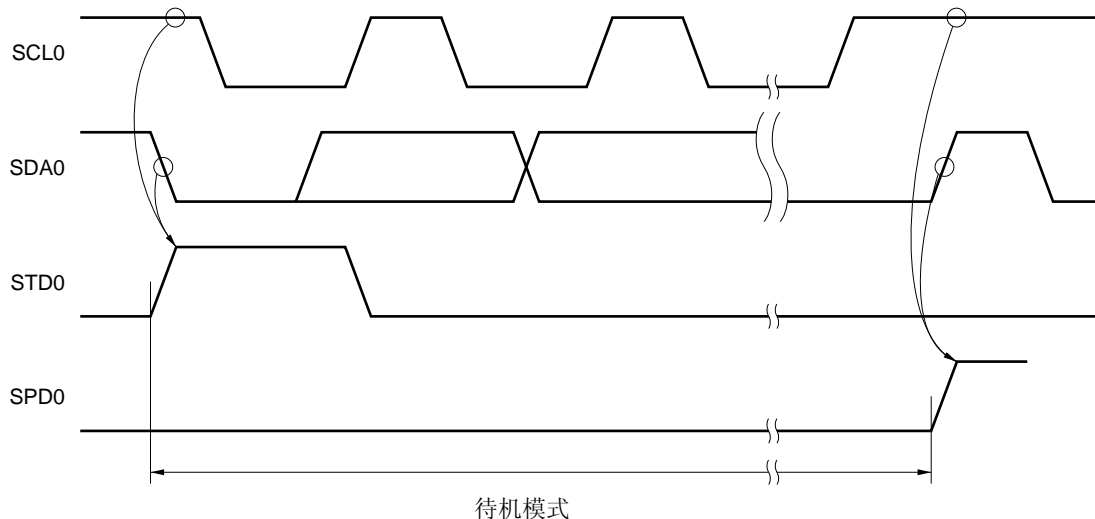
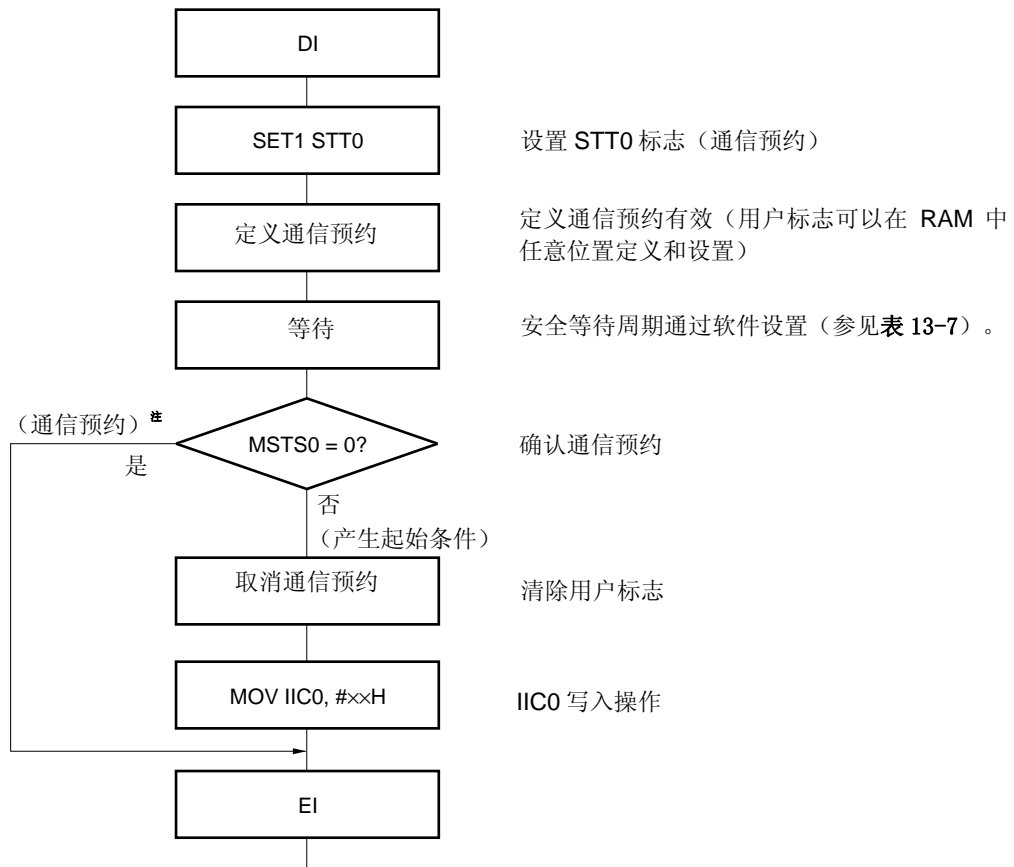


图 13-23 显示通信预约协议。

图 13-23. 通信预约协议



注 当产生一个停止条件中断请求时，通信预约将对 IIC 移位寄存器 0（IIC0）执行写入操作。

备注

STT0: IIC 控制寄存器 0（IICC0）的第 1 位

MSTS0: IIC 状态寄存器 0（IICS0）的第 7 位

IIC0: IIC 移位寄存器 0

(2) 当禁止使用通信预约功能时 (IIC 标志寄存器 0 (IICF0) 的第 0 位 (IICRSV) = 1)

如果在总线通信期间，总线未使用时将 IIC 控制寄存器 0 (IICC0) 的第 1 位 (STT0) 设置为 1，则该请求会被拒绝并且不产生起始条件。下列两种状态包括了总线未使用时的情况。

- 当仲裁结果不是主/从设备操作时
- 接收到一个扩展码且从设备操作被禁止时 (不返回 \overline{ACK} 信号，且 IIC 控制寄存器 0 (IICC0) 的第 6 位 (LREL0) 被设置为 1 时，释放总线)

通过检查 STCF (IICF0 的第 7 位) 来确认是产生了起始条件还是请求被拒绝。设置 STT0 = 1 后，最多需要 5 个时钟直到 STCF 被设置为 1。因此，必须通过软件确保等待时间。

13.5.16 注意事项

- (1) 当 STCEN (IIC 标志寄存器 0 (IICF0) 的第 1 位) = 0 时

在允许 I²C 操作 (IICE0 = 1) 后, 无论总线的实际状态如何, 都可以识别总线通信状态 (IICBSY (IICF0 的第 6 位) = 1)。从某个未检测到停止条件的模式变为主设备通信模式时, 首先产生一个停止条件以释放总线, 然后进行主设备通信。

当使用多个主设备时, 如果总线未被释放 (没有检测到停止条件), 则不能进行主设备通信。

按下列步骤产生停止条件。

- <1> 设置 IIC 时钟选择寄存器 0 (IICCL0)。
- <2> 设置 IIC 控制寄存器 0 (IICC0) 的第 7 位 (IICE0) 为 1。
- <3> 设置 IICC0 的第 0 位 (SPT0) 为 1。

- (2) 当 STCEN = 1

在允许 I²C 操作 (IICE0 = 1) 后, 无论总线的实际状态如何, 都可以识别总线通信状态 (IICBSY (IICF0 的第 6 位) = 1)。要产生第一个起始条件 (STT0 (IIC 控制寄存器 0 (IICC0) 的第 1 位) = 1), 必须确认总线已被释放, 以免干扰其他通信。

- (3) 如果其它 I
- ²
- C 通信已经在进行

在 SDA0 引脚为低电平且 SCL0 引脚为高电平时, 允许 I²C 操作且设备已进行通信, 则 I²C 模块可以识别 SDA0 引脚已经变为低电平 (检测到一个起始条件)。如果此时总线上的值被识别为一个扩展码, 则返回 $\overline{\text{ACK}}$ 信号, 但这会干扰其他 I²C 通信。为了避免这种情况, 可按下列步骤启动 I²C。

- <1> 将 IICC0 的第 4 位 (SPIE0) 清零, 禁止在检测到停止条件时产生中断请求信号 (INTIIC0)。
- <2> 设置 IICC0 的第 7 位 (IICE0) 为 1, 允许 I²C 的操作。
- <3> 等待检测起始条件。
- <4> $\overline{\text{ACK}}$ 信号返回之前 (设置 IICE0 为 1 之后的 4 到 80 个时钟) 设置 IICC 的第 6 位 (LREL0) 为 1, 来强行禁止检测。

- (4) 允许操作之前 (IICE0 = 1), 使用 SMC0, CL01, CL00 (IICL0 的第 3, 1 和 0 位) 和 CLX0 (IICX0 的第 0 位) 联合决定传送时钟频率。要改变传送时钟频率, 必须将 IICE0 清零。

- (5) 在对 STT0 和 SPT0 (IICC0 的第 1 位和第 0 位) 设置之后和清零之前, 禁止再次设置。

- (6) 预约发送时, 设置 SPIE0 (IICL0 的第 4 位) 为 1, 以允许在检测到停止条件时产生中断请求。中断请求产生后, 将通信数据写入 IIC0, 传送过程开始。如果检测到停止条件时不产生中断请求, 设备将处于等待状态, 因为通信开始时不产生中断请求。但是, 由软件检测 MSTS0 (IICS0 的第 7 位) 时不必将 SPIE0 设置为 1。

13.5.17 通信操作

以下显示了三种操作过程和其流程图。

(1) 单主设备系统中的主设备操作

使用 78K0R/KF3 用作单主设备系统中的主设备时，流程图显示如下。

该流程图基本分为初始化设置和通信过程。在程序开始处执行初始化设置。如果需要和从设备通信，准备通信然后执行通信处理。

(2) 多主设备系统中的主设备操作

在 I²C 总线的多主设备系统，当总线参与通信时，不能通过 I²C 总线规范判断总线是否释放。于是，当数据和时钟在某个期间（1 帧）内为高电平时，78K0R/KF3 在总线释放状态时参与通信。

该流程基本分为初始化设置，通信等待和通信处理。这里没有显示当 78K0R/KF3 仲裁失利而作为从设备的处理，只显示作为主设备的处理。在启动时执行初始化设置以参与通信。然后，作为主设备等待通信请求或者作为从设备等待指定。在通信处理中执行实际通信，可以支持和从设备的发送/接收，也支持和其它主设备进行仲裁。

(3) 从设备操作

78K0R/KF3 用作 I²C 总线的从设备的示例如下。

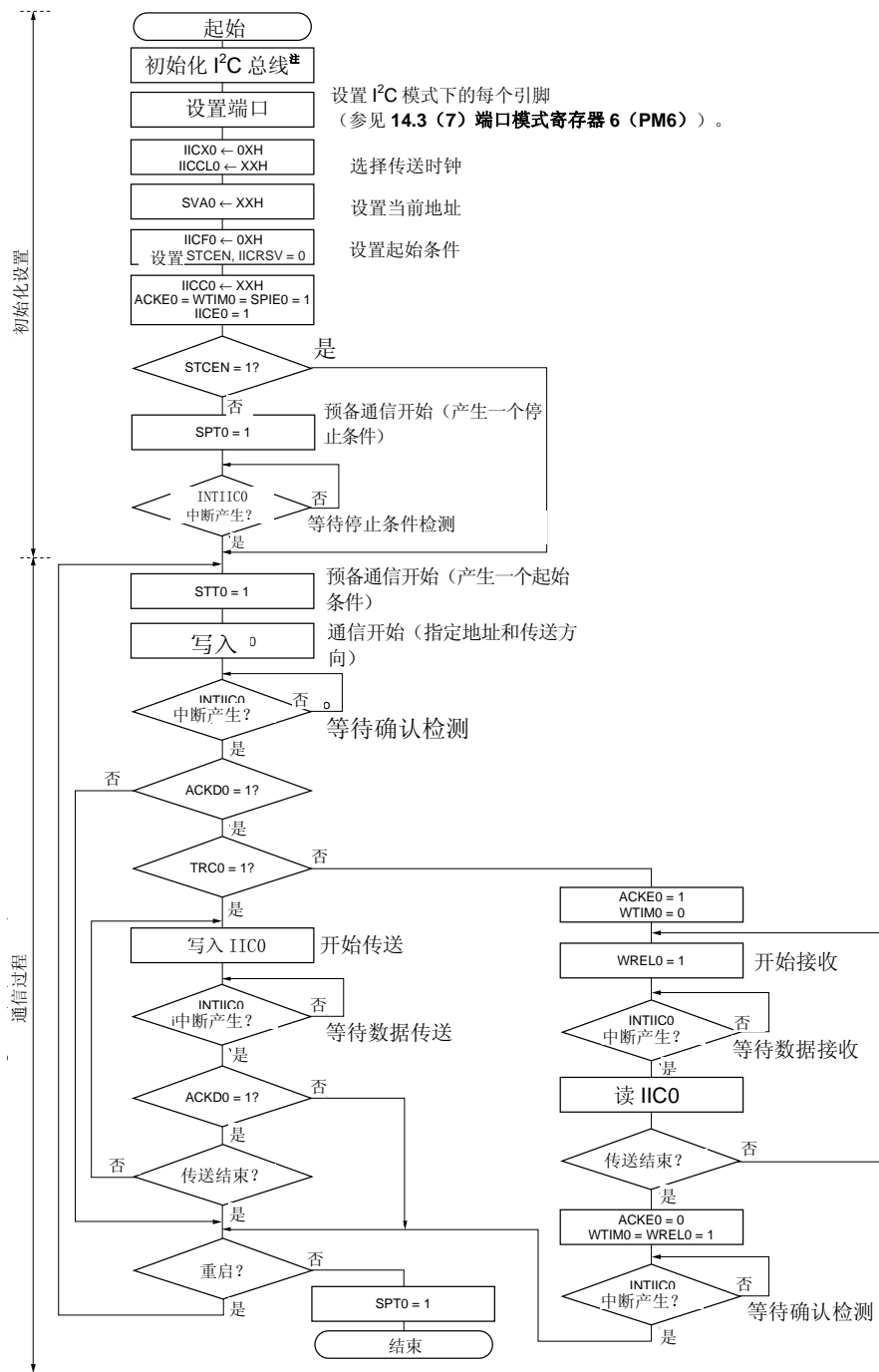
当用作从设备时，由中断开始操作。在启动时执行初始化设置，然后等待 INTIIC0 中断产生（通信等待）。

当产生一个 INTIIC0 中断时，判断通信状态，并且将该判断结果作为标志传递给主程序处理。

通过检查这些标志，可以执行必要的通信处理。

(1) 单主设备系统中的主设备操作

图 13-24. 单主设备系统中的主设备操作

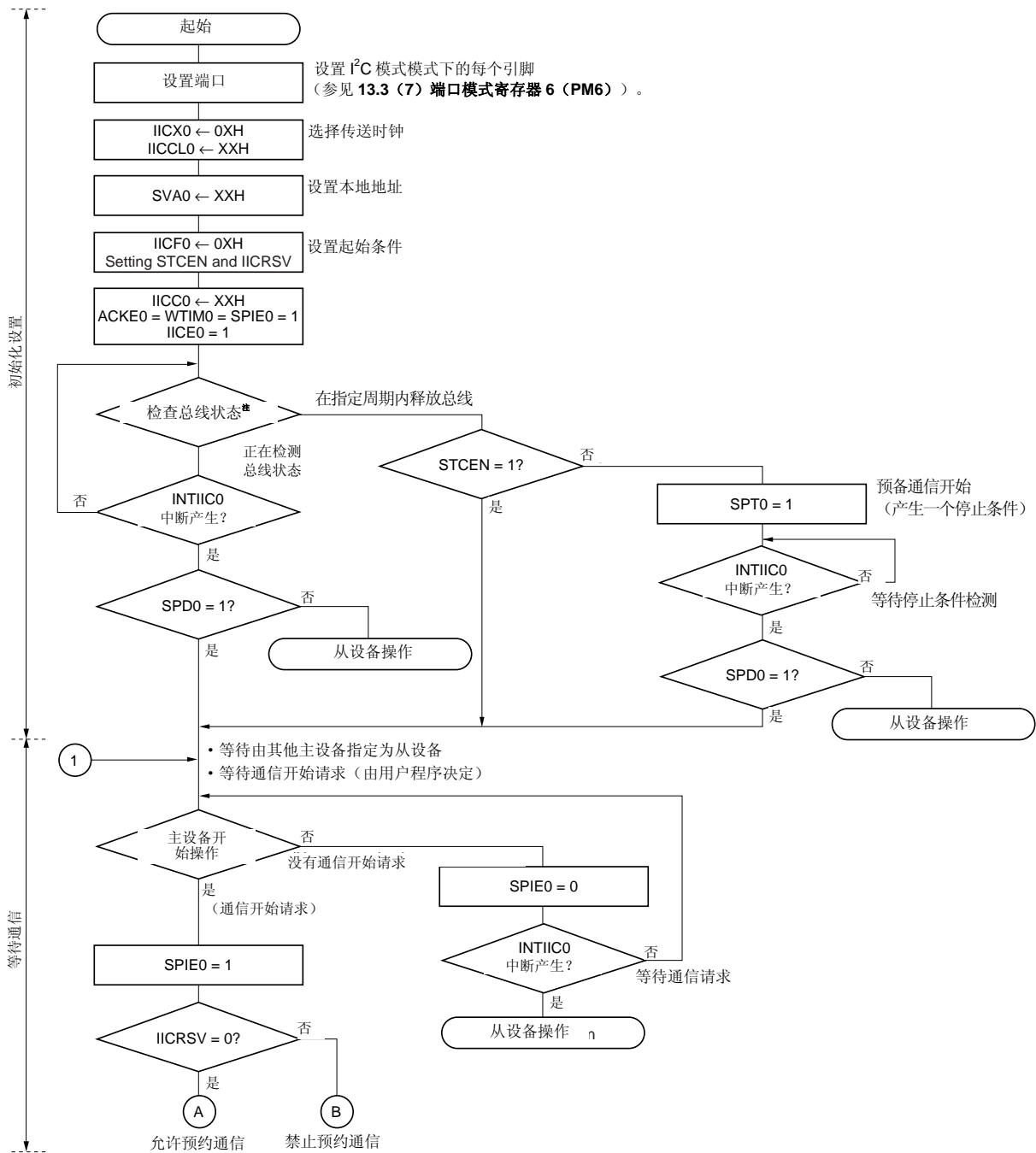


注 根据正在通信的产品规范释放 (SCL0 和 SDA0 引脚 = 高电平) I²C 总线。如果 EEPROM 向 SDA0 引脚输出一个低电平, 例如, 在输出端口模式下设置 SCL0 引脚, 且从输出端口输出一个时钟脉冲直到 SDA0 引脚持续处于高电平。

备注 遵照正在通信的产品规范, 注意发送和接收的格式。

(2) 多主设备系统中的主设备操作

图 13-25. 多主设备系统中的主设备操作 (1/3)



注 确认总线释放 (CLD0 位 = 1, DAD0 位 = 1) 的时间已经超过某段指定的时间 (如, 一帧的周期)。如果 SDA0 引脚持续处于低电平, 可以根据正在使用的产品规范决定是否释放 (SCL0 和 SDA0 引脚 = 高电平) I²C 总线。

图 13-25. 多主设备系统中的主设备操作 (2/3)

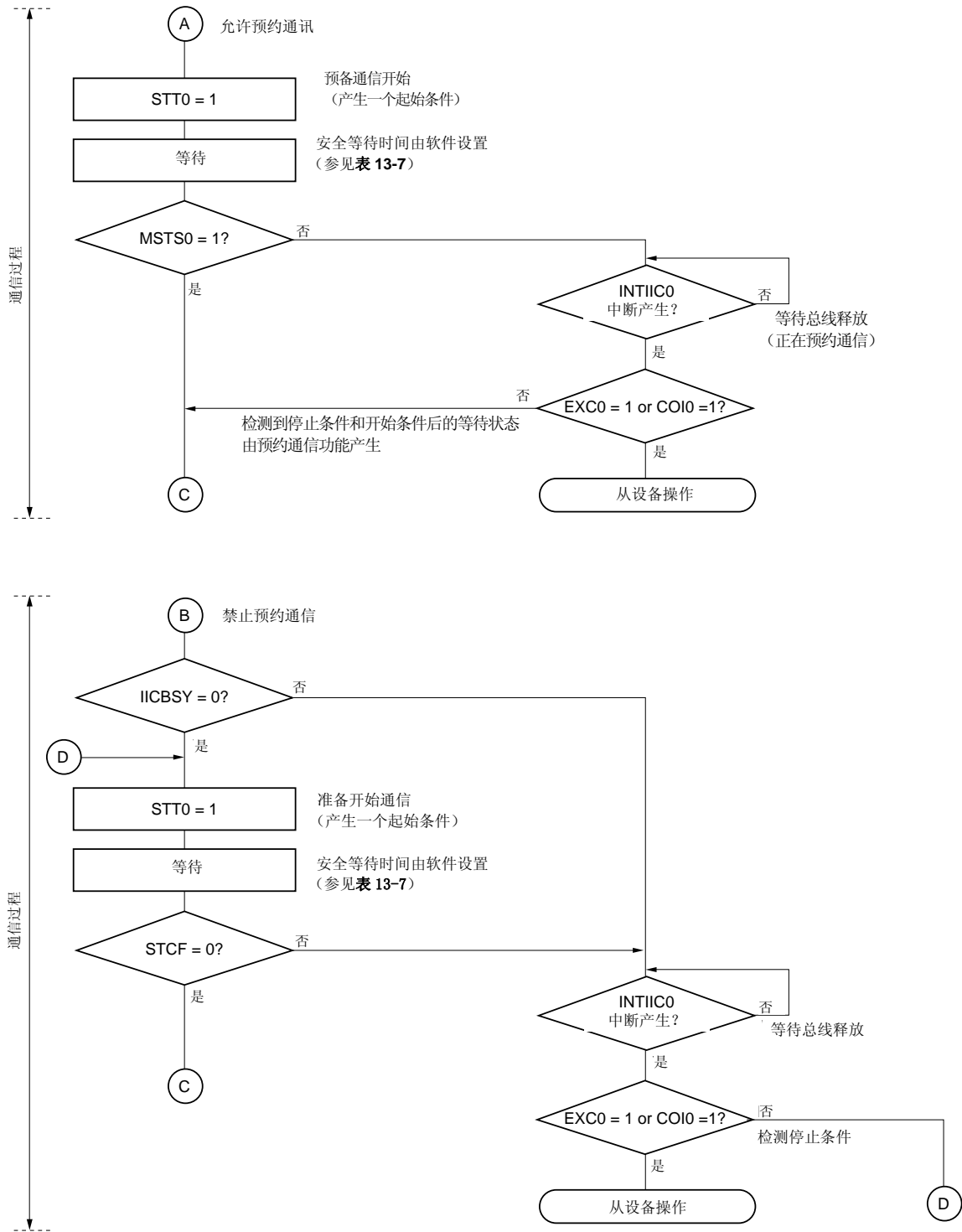
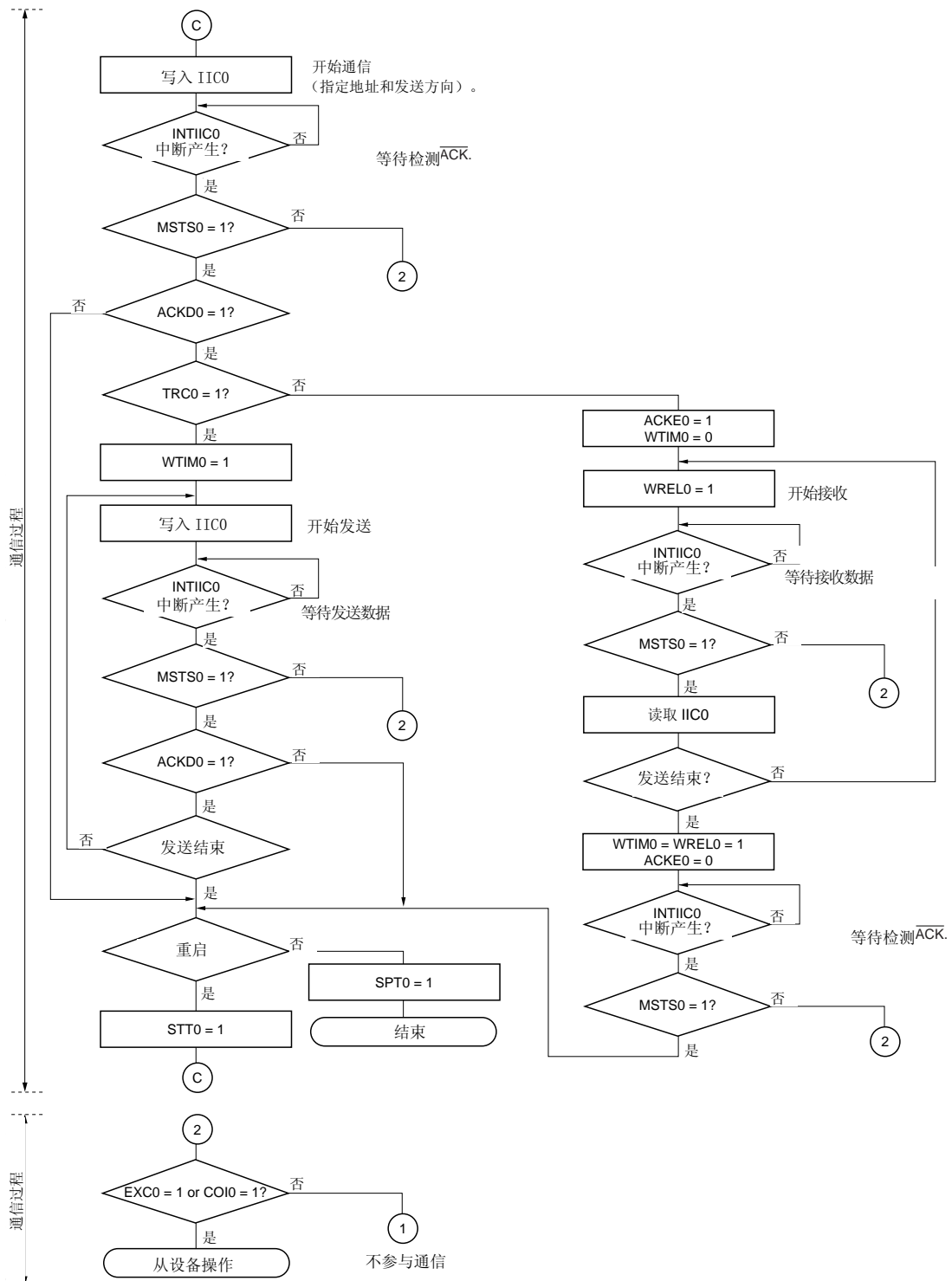


图 13-25. 多主设备系统中的主设备操作 (3/3)



备注

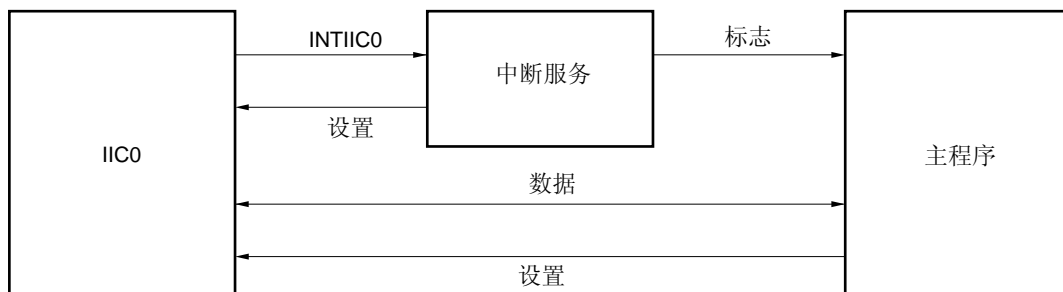
1. 遵照正在通信的产品规范，注意发送和接收的格式。
2. 如果在多主设备系统中作为一个主设备使用时，则应在每次中断 INTIIC0 发生后读取 MSTS0 位来检查仲裁结果。
3. 如果在多主设备系统中作为一个从设备使用时，则通过在每次中断 INTIIC0 发生后使用 IICS0 和 IICF0 寄存器来检查状态，并决定下一步的处理。

(3) 从设备操作

从设备操作过程如下所示。

基本上，从设备操作是由事件驱动的。因此，必须执行 INTIIC0 中断处理过程（该中断处理过程必须从本质上完全改变操作状态，例如通信时的停止条件检测）。

在下面的说明中，假定数据通信不支持扩展码。同时假定 INTIIC0 中断服务只执行状态转换处理，实际数据通信过程由主程序执行。



因此，准备以下 3 个标志，并将它们传递给主程序，而不是传递 INTIIC0，这样可以执行数据通信处理过程。

<1> 通信模式标志

此标志指示下列两种通信状态。

- 清除模式：未进行数据通信的状态
- 通信模式：进行数据通信的状态（从有效地址检测到停止条件检测，不检测来自主设备的 \overline{ACK} 信号，地址不匹配）

<2> 就绪标志

此标志表示允许进行数据通信。它的功能和普通数据通信中的 INTIIC0 中断相同。通过中断服务设置此标志，通过主程序将该标志清零。通信启动时，可以由中断服务将此标志清零。但是，在发送第一个数据时，不能由中断服务设置就绪标志。因此，发送第一个数据时，该标志未清零（地址匹配被认为是请求下一个数据）。

<3> 通信方向标志

此标志表示通信方向。它的值与 TRC0 相同。

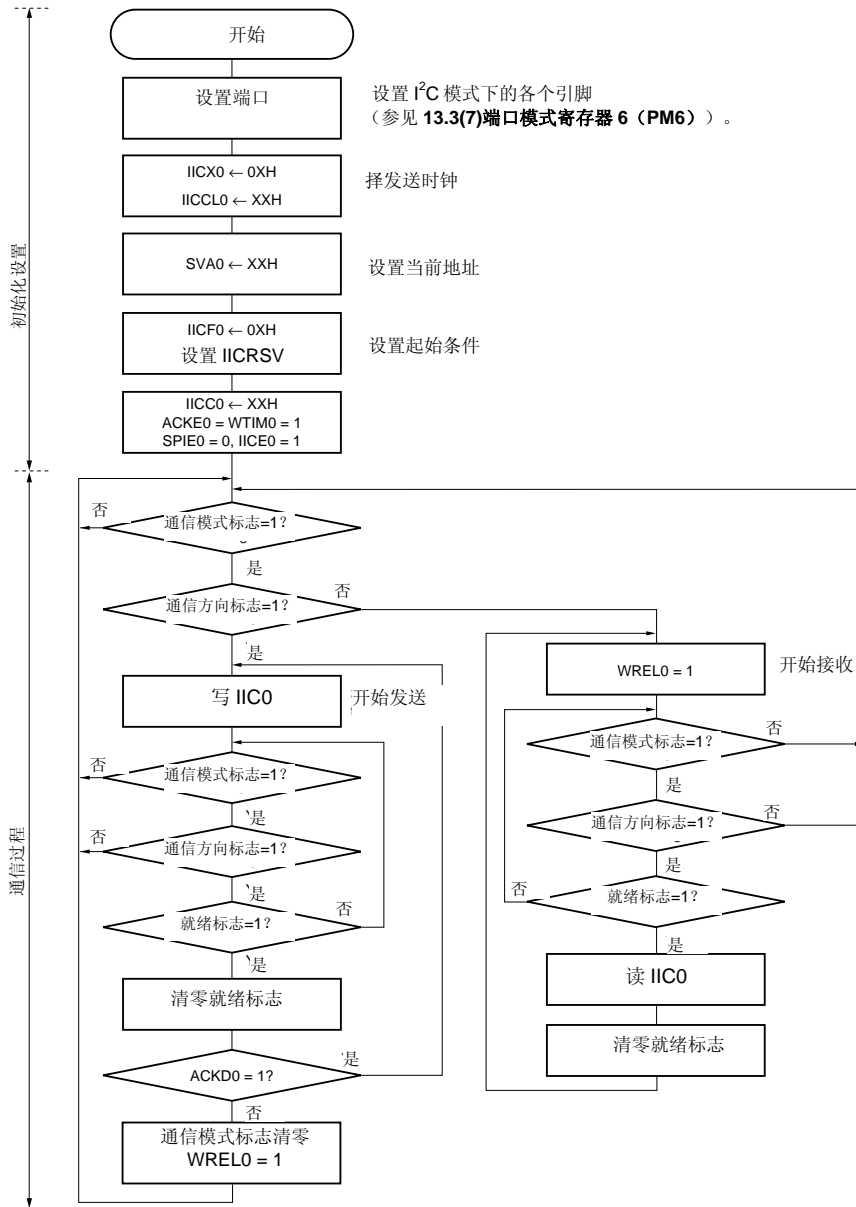
从设备操作的主程序处理过程说明如下。

启动串行接口 IIC0 并等待，直到允许通信。当允许通信时，通过使用通信模式标志和就绪标志来执行通信（由中断来执行停止条件和起始条件的处理。这里，通过这些标志来检查状态）。

重复发送操作直到主设备不再返回 ACK 信号。如果主设备没有返回 ACK 信号，则通信结束。

对于接收操作，则要接收一定数量的数据。通信完成时，ACK 信号不作为下一个数据返回。此后，主设备产生一个停止条件或重启条件。使用这种方法退出通信状态。

图 13-26. 从设备操作流程图 (1)



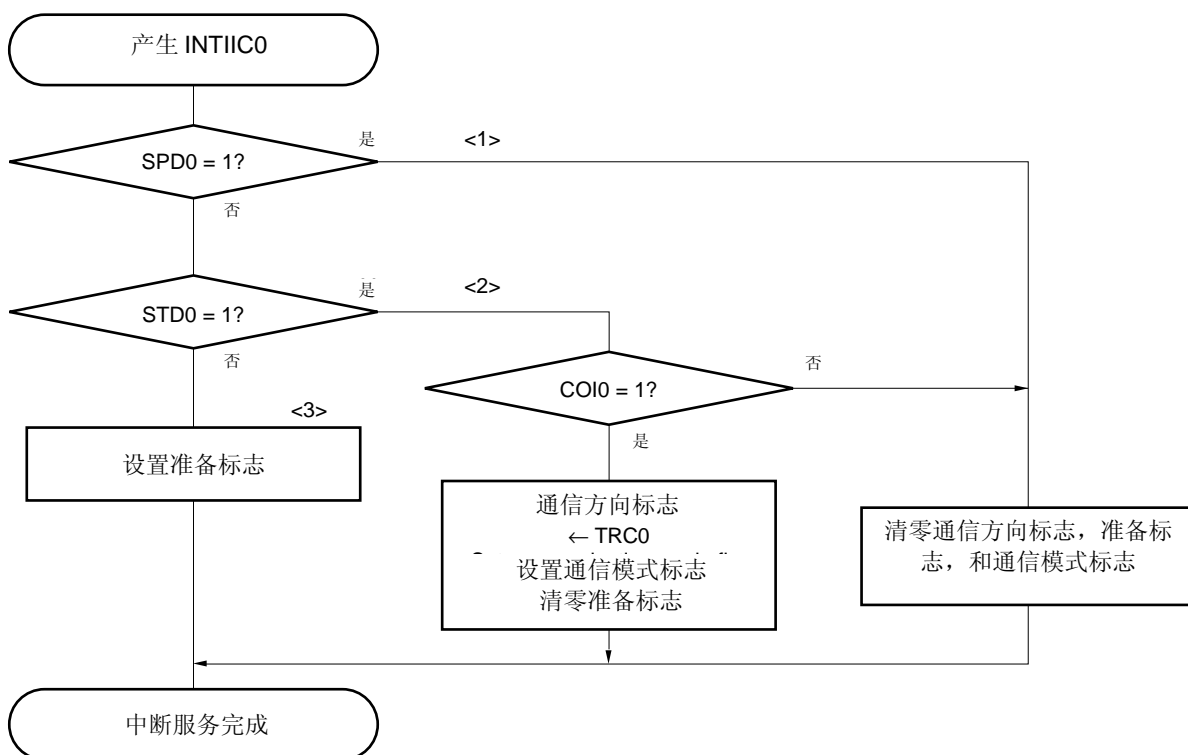
备注 遵照正在通信的产品规范，注意发送和接收的格式。

以下示例说明了用 INTIIC0 中断进行的从设备处理过程（假设该处理中不使用扩展码）。INTIIC0 中断会检查状态，同时执行下列操作。

- <1> 若发出停止条件，则通信停止。
- <2> 若发出起始条件，检查地址，如果地址不匹配，则结束通信。如果地址匹配，则设置通信模式，取消等待，且处理过程从中断返回（就绪标志清零）。
- <3> 对于数据发送/接收，仅需设置就绪标志。处理过程从中断返回，同时 I²C 总线保持等待状态。

备注 以上<1> ~ <3> 与图 13-27 从设备操作流程图中（2）中<1> ~ <3> 相对应。

图 13-27. 从设备操作流程图中（2）



13.5.18 产生I²C中断请求 (INTIIC0) 的时序

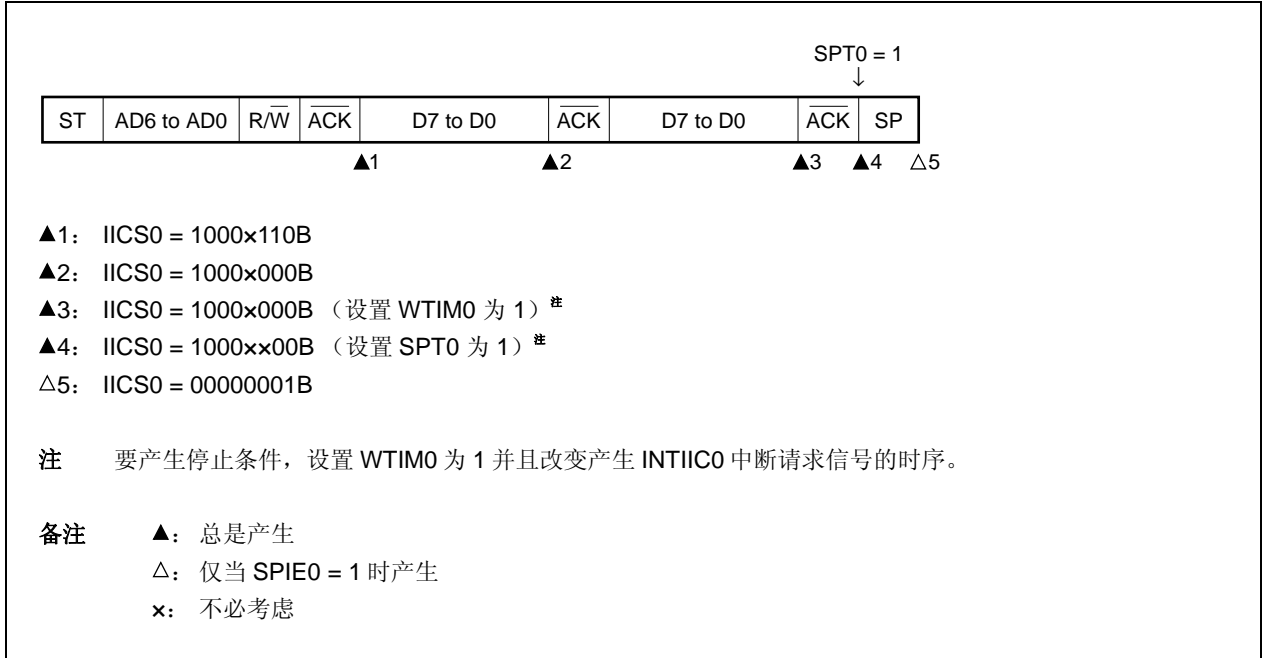
如下所示为发送或者接收数据的时序，产生 INTIIC0 中断请求信号的时序，以及产生 INTIIC0 信号时寄存器 IICS0 的值。

备注	ST:	起始条件
	AD6 ~ AD0:	地址
	R/W:	发送方式规范
	ACK:	应答
	D7 ~ D0:	数据
	SP:	停止条件

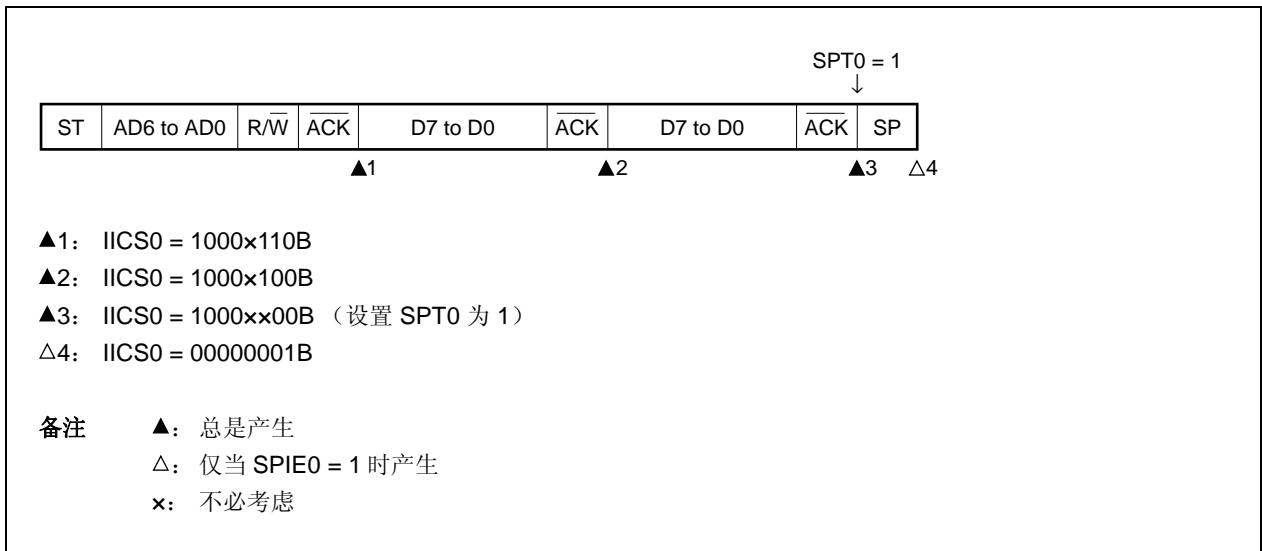
(1) 主设备操作

(a) 起始 ~ 地址 ~ 数据 ~ 数据 ~ 停止 (发送/接收)

(i) 当 **WTIMO = 0**

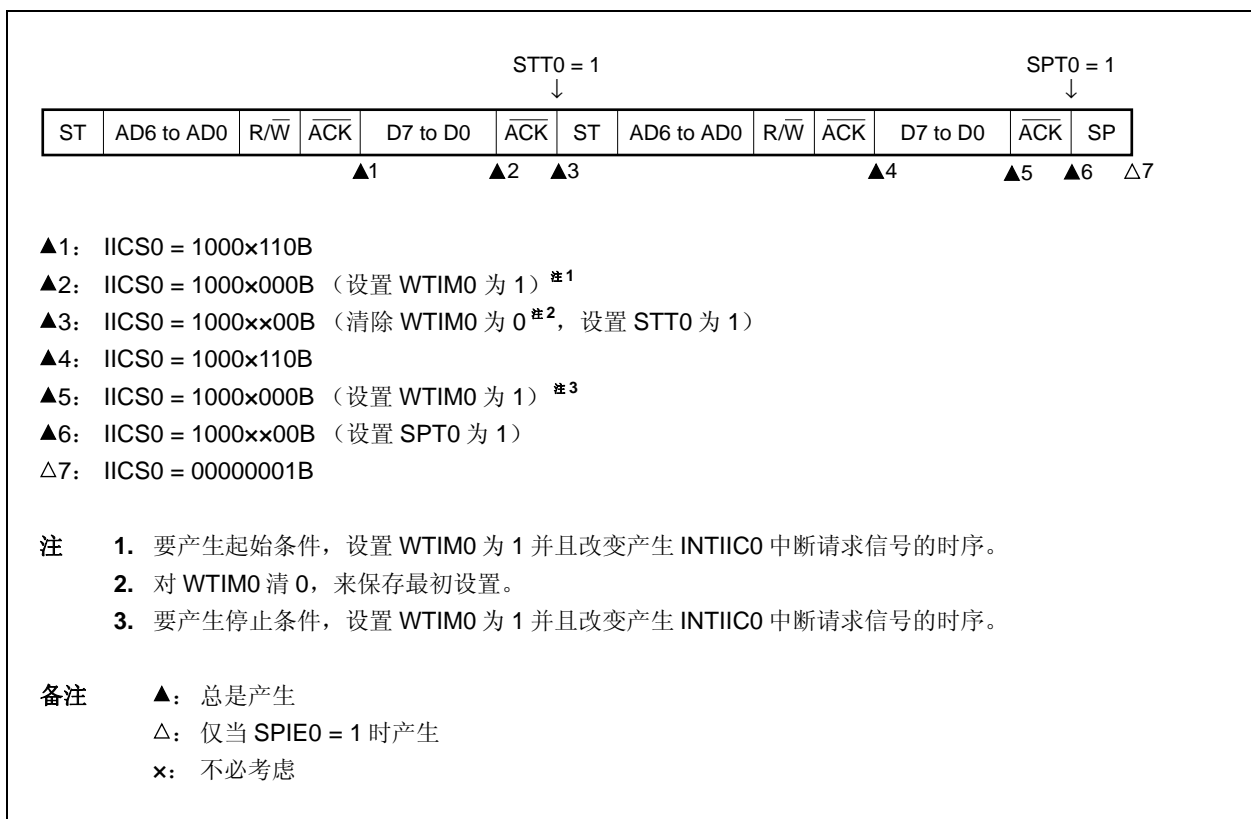


(ii) 当 **WTIMO = 1**

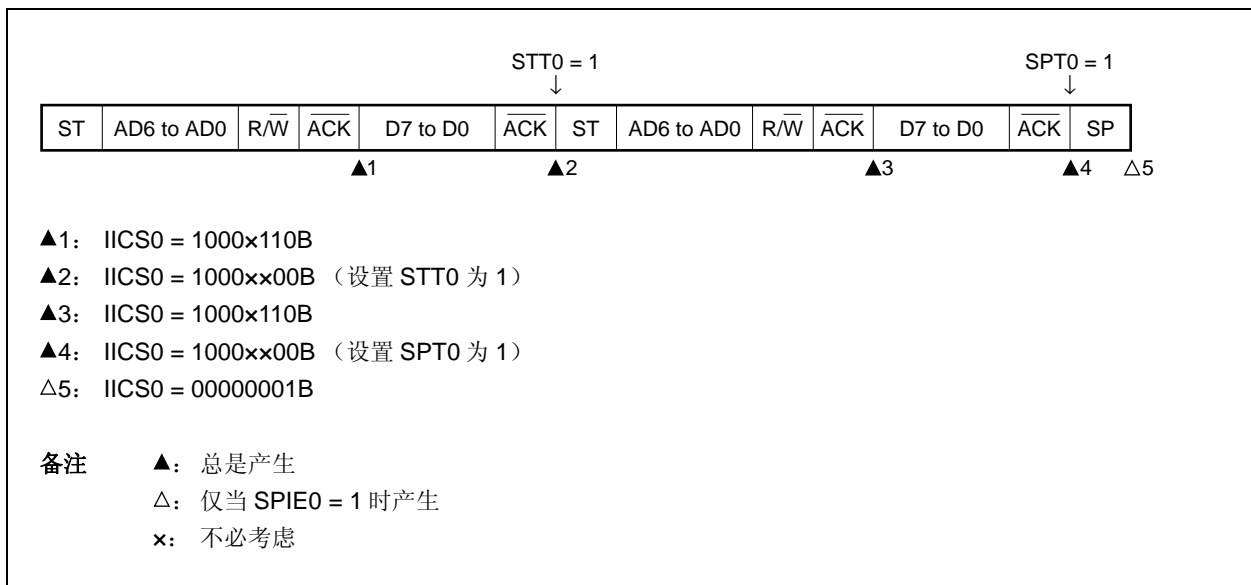


(b) 起始 ~ 地址 ~ 数据 ~ 起始 ~ 地址 ~ 数据 ~ 停止 (重启)

(i) 当 **WTIMO = 0**

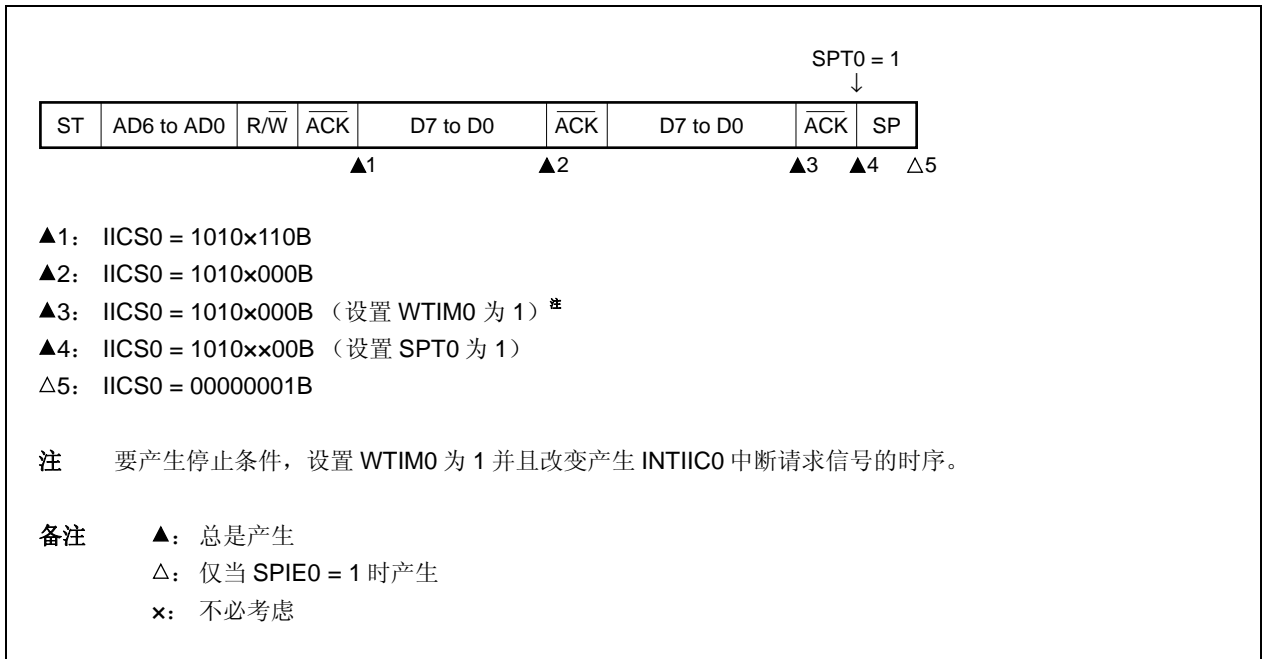


(ii) 当 **WTIMO = 1**

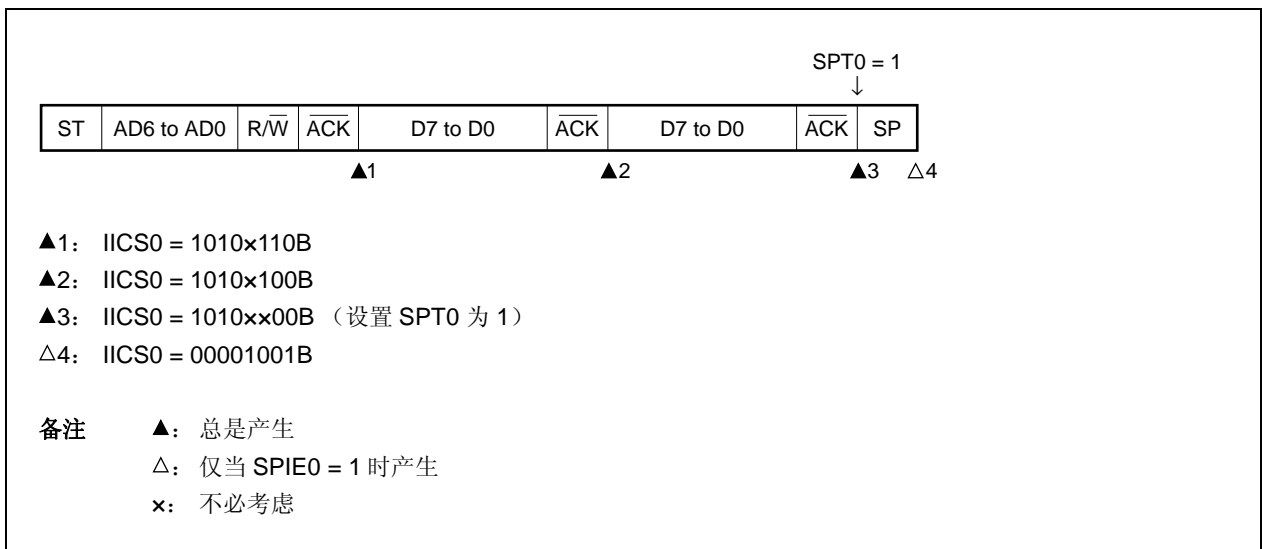


(c) 起始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展码发送)

(i) 当 **WTIM0 = 0**



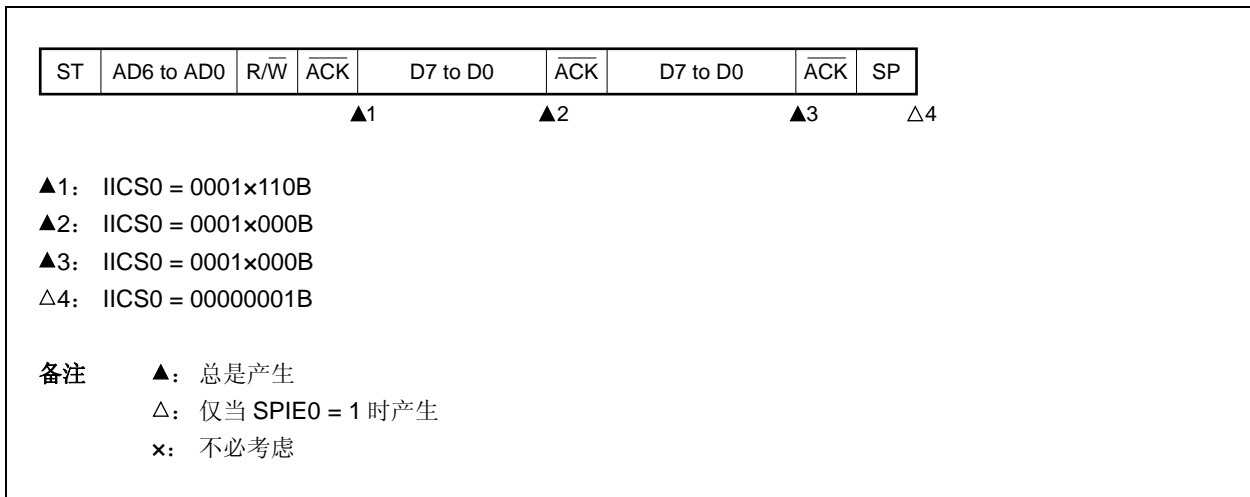
(ii) When **WTIM0 = 1**



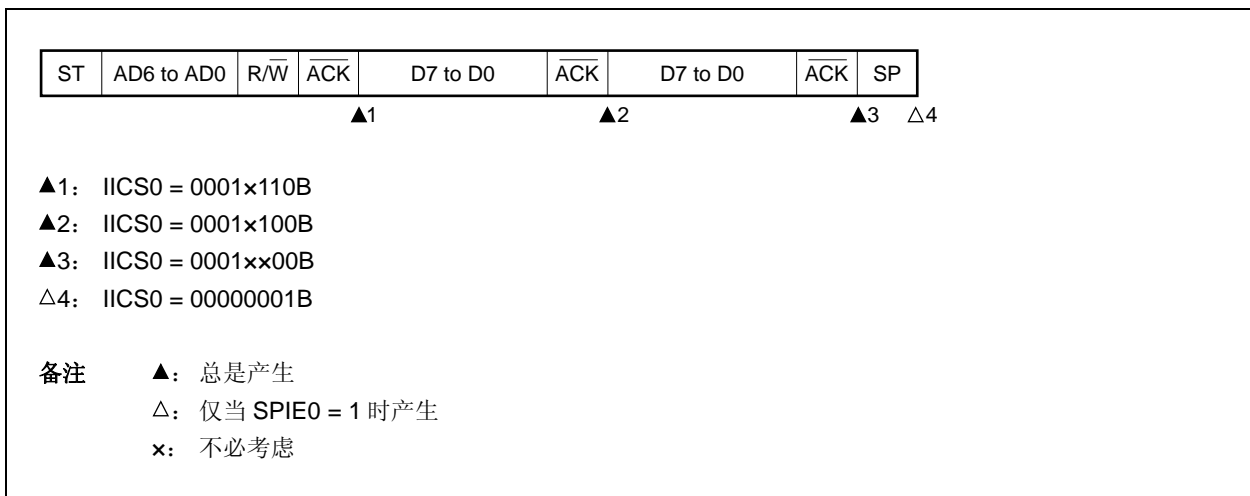
(2) 从设备操作 (从地址数据接收)

(a) 起始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0**

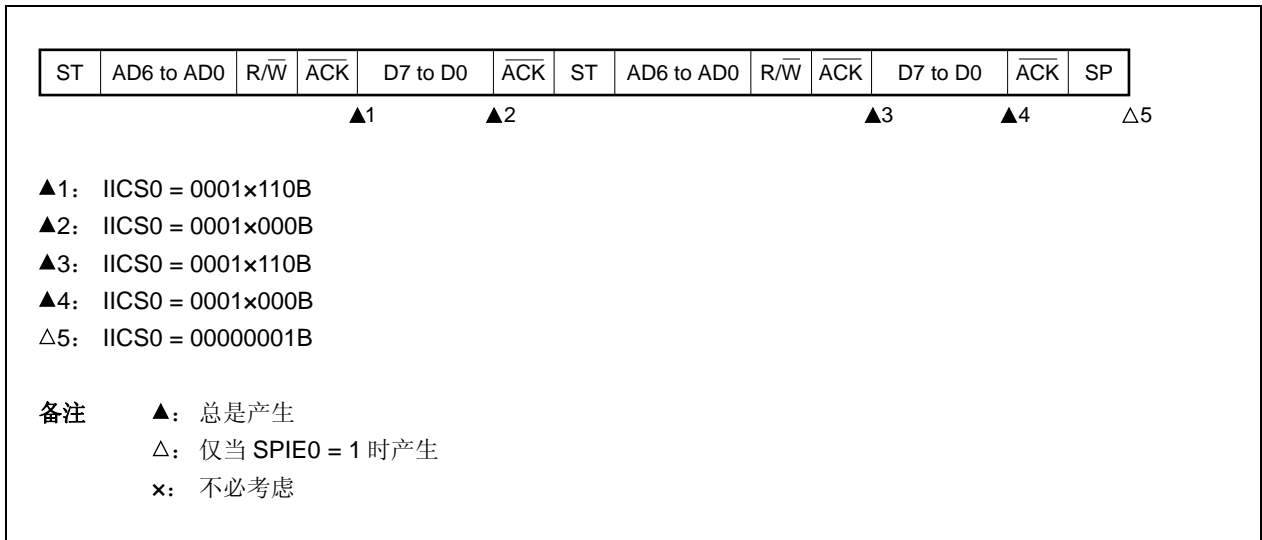


(ii) 当 **WTIMO = 1**

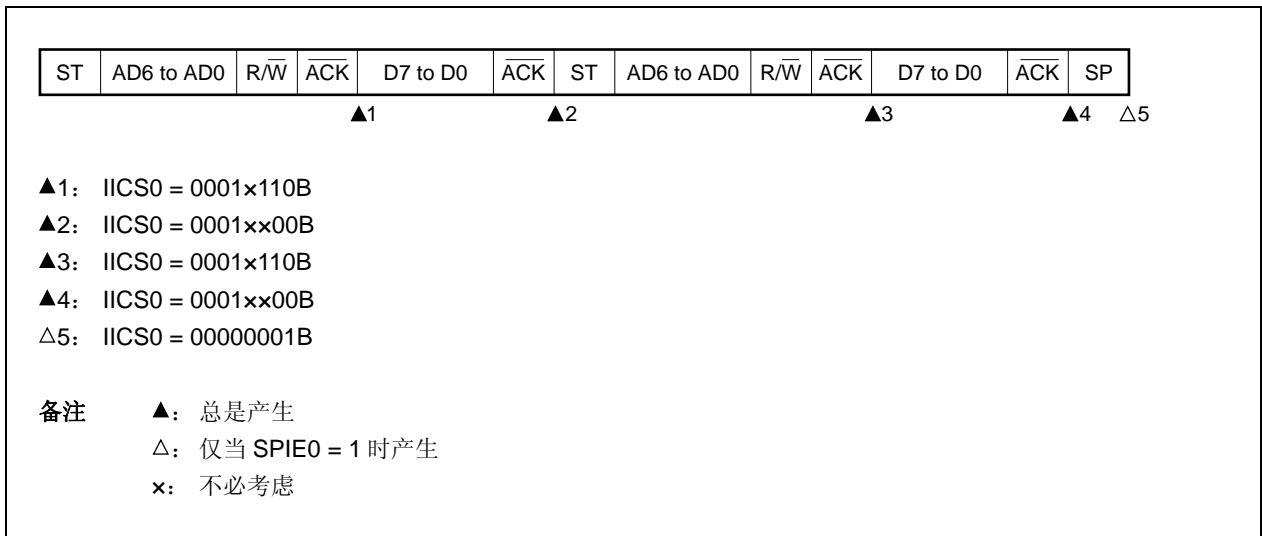


(b) 起始 ~ 地址 ~ 数据 ~ 起始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** (重启后, 和 **SVA0** 匹配)

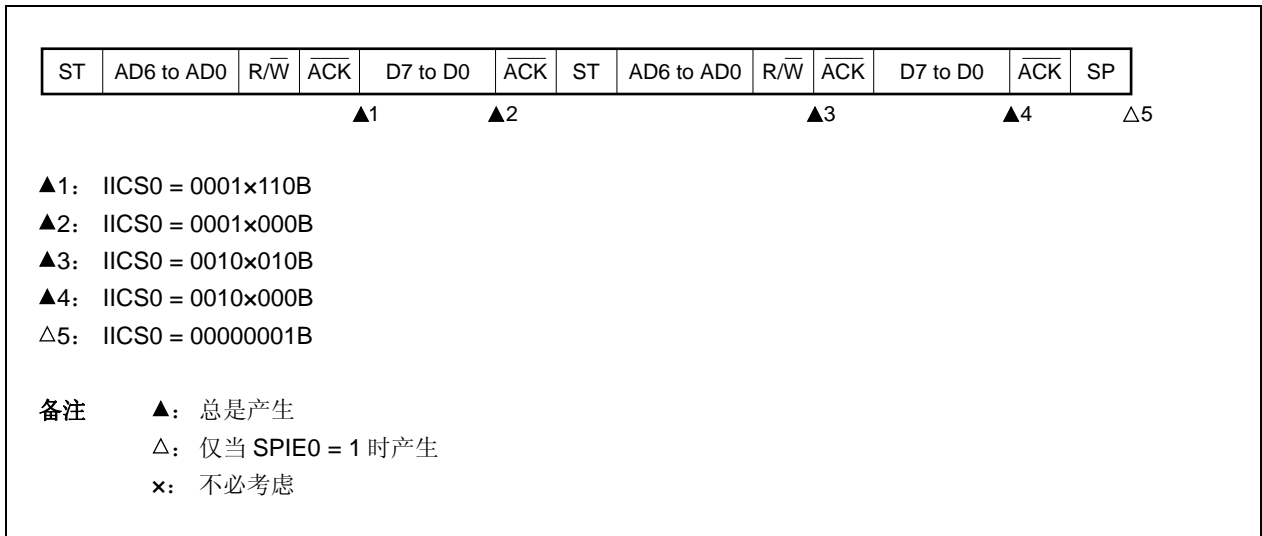


(ii) 当 **WTIMO = 1** (重启后, 和 **SVA0** 匹配)

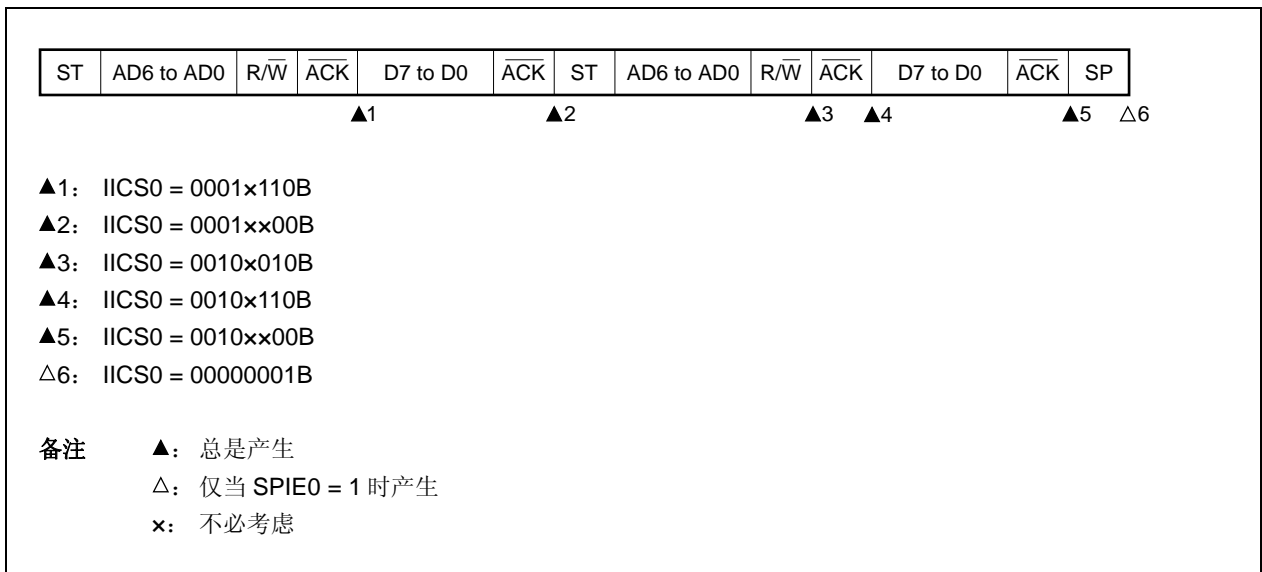


(c) 起始 ~ 地址 ~ 数据 ~ 起始 ~ 代码 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** (重启后, 和地址不匹配相等 (= 扩展码))

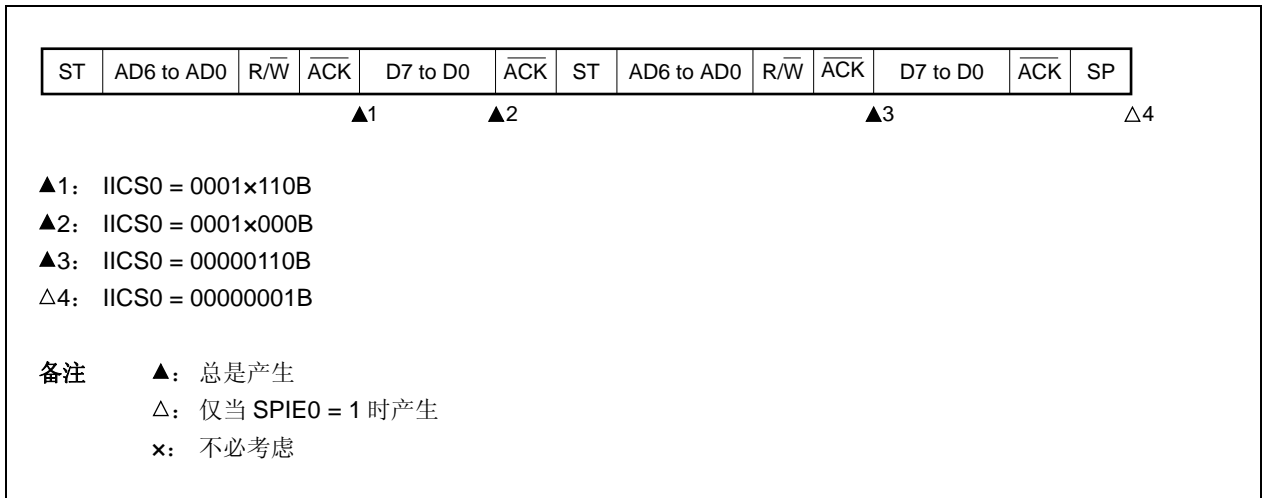


(ii) 当 **WTIMO = 1** (重启后, 和地址不匹配相等 (= 扩展码))

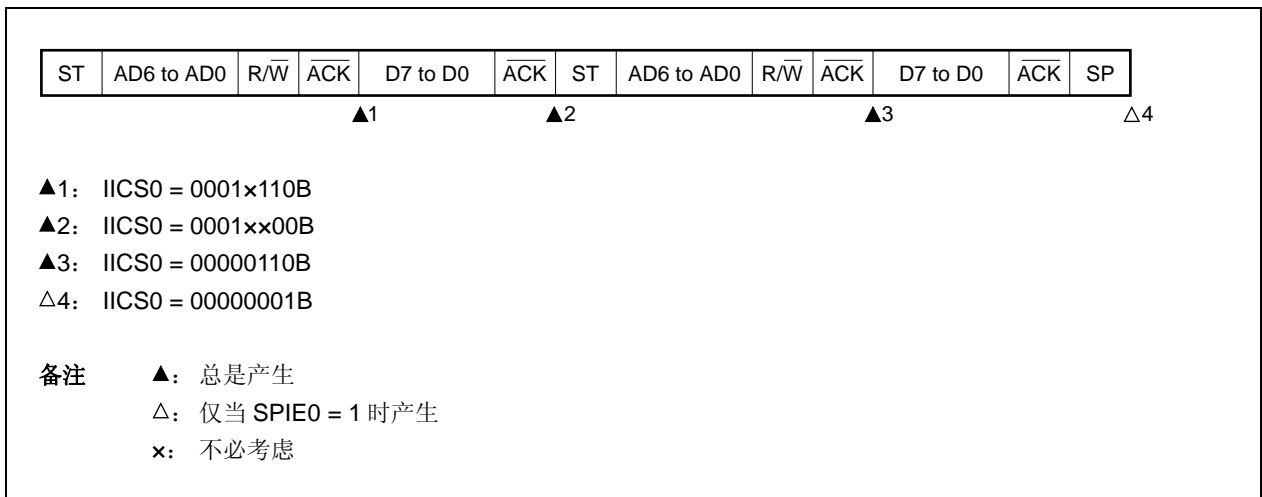


(d) 起始 ~ 地址 ~ 数据 ~ 起始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** (重启后, 和地址不匹配相等 (= 扩展码))



(ii) 当 **WTIMO = 1** (重启后, 和地址不匹配相等 (= 扩展码))

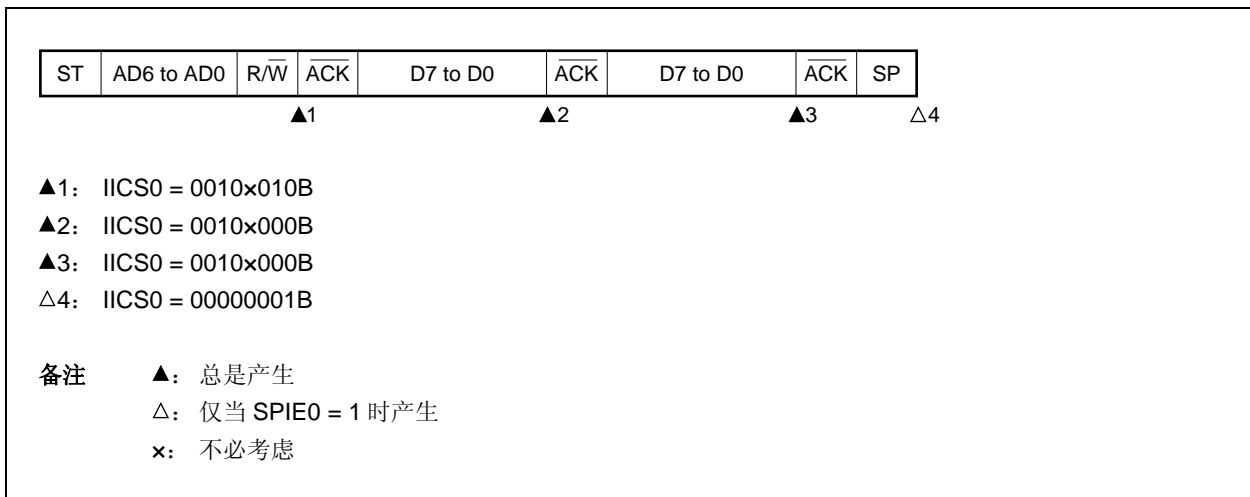


(3) 从设备操作 (当接收扩展码时)

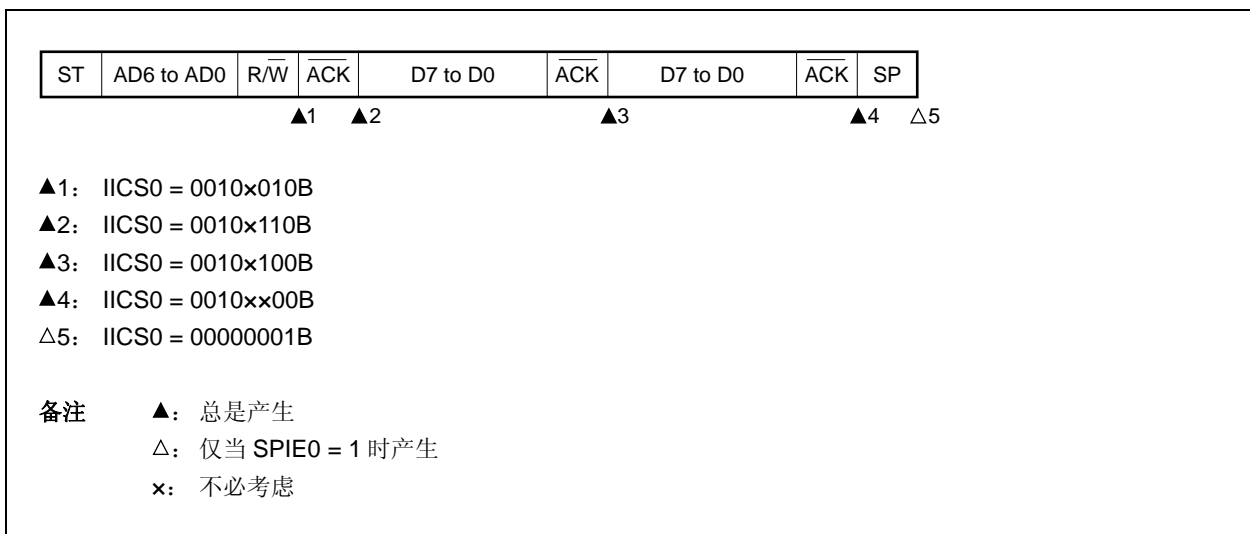
在接收扩展码时, 设备一直参与通信。

(a) 起始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0**

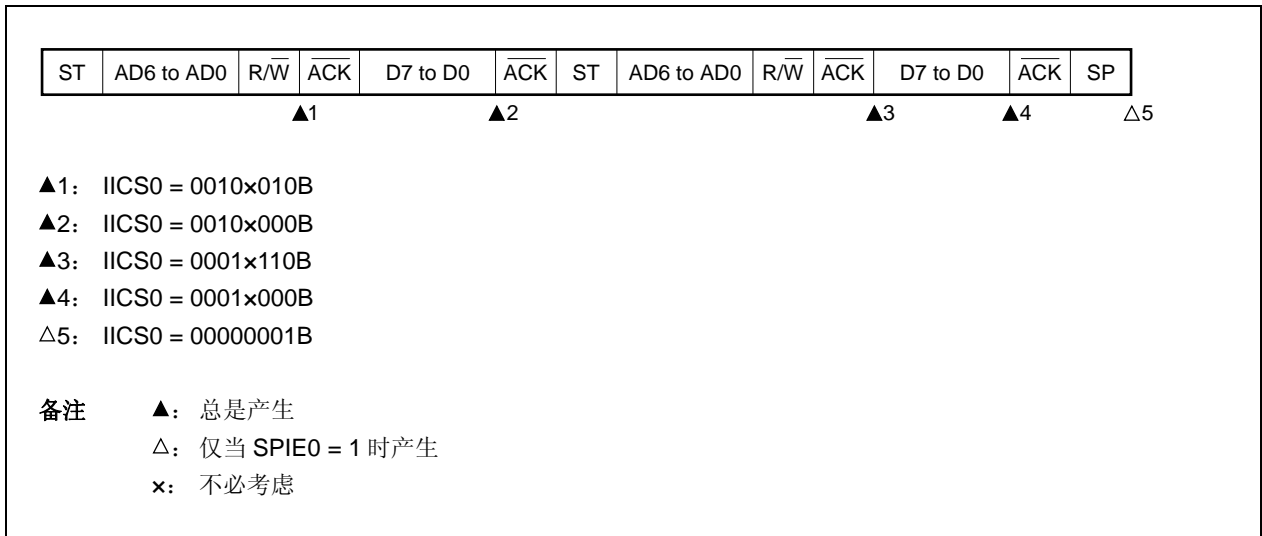


(ii) 当 **WTIMO = 1**

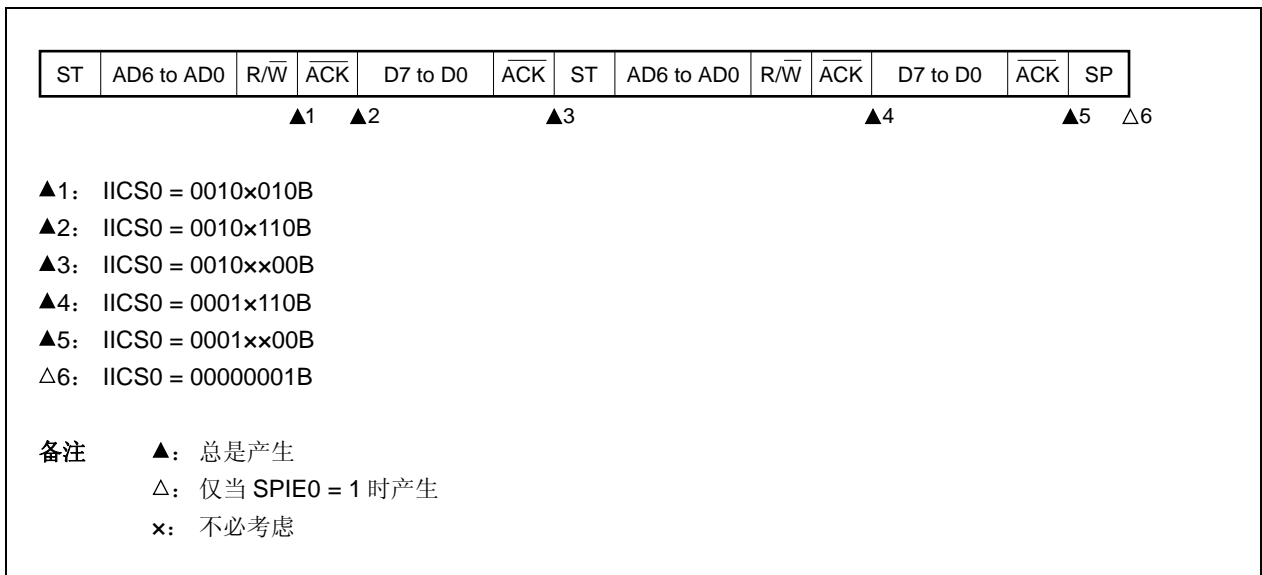


(b) 起始 ~ 代码 ~ 数据 ~ 起始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIM0 = 0** (重启后, 和 **SVA0** 匹配)

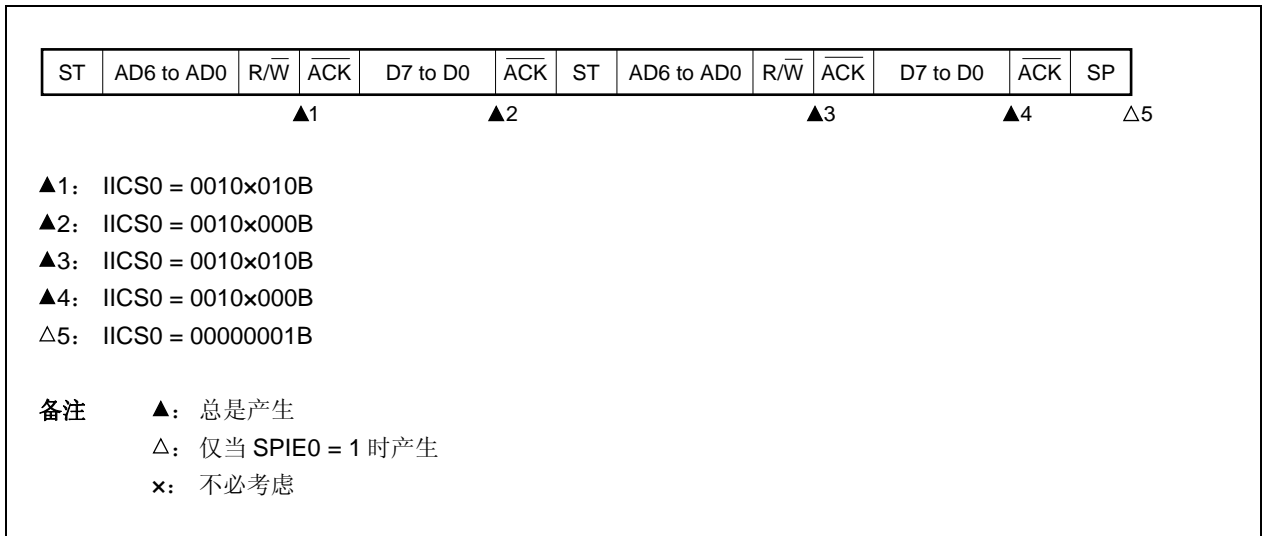


(ii) 当 **WTIM0 = 1** (重启后, 和 **SVA0** 匹配)

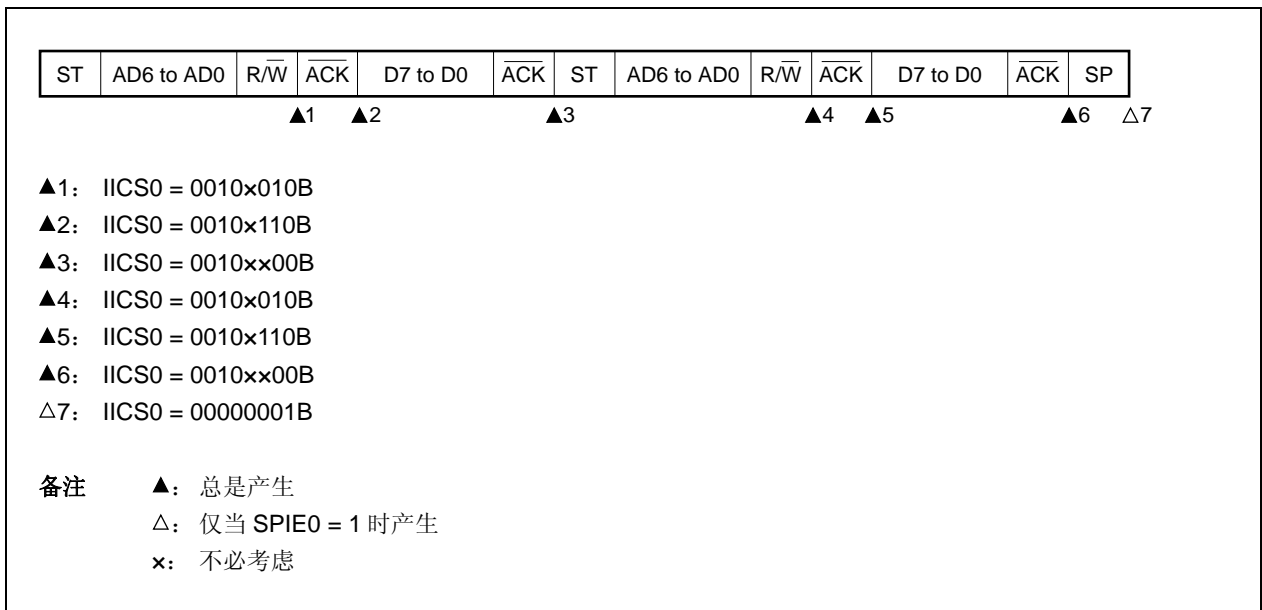


(c) 起始 ~ 代码 ~ 数据 ~ 起始 ~ 代码 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** (复位后, 接收扩展码)

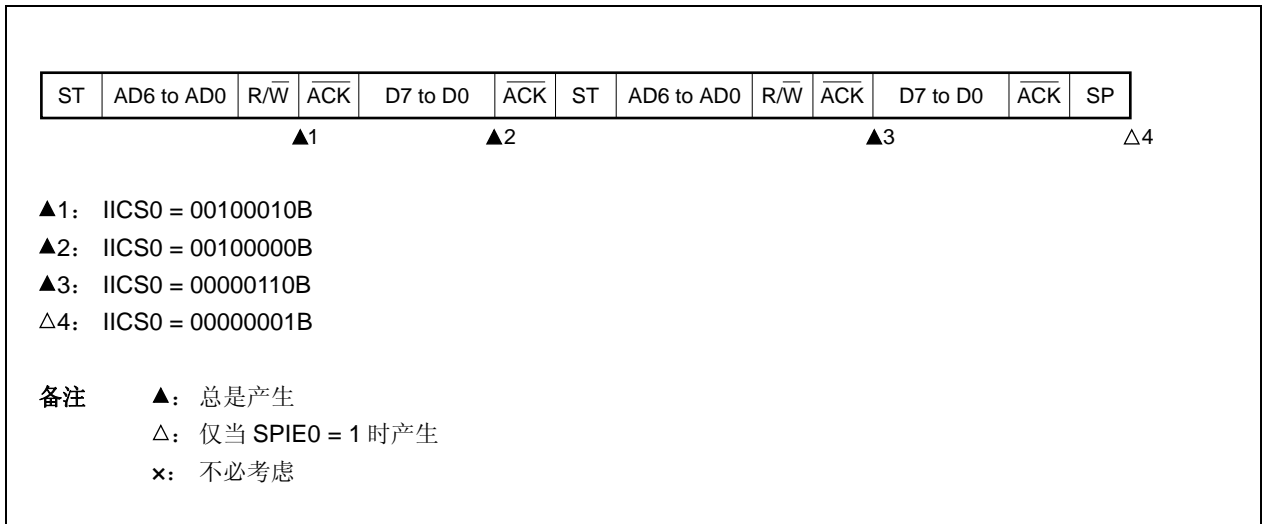


(ii) 当 **WTIMO = 1** (复位后, 接收扩展码)

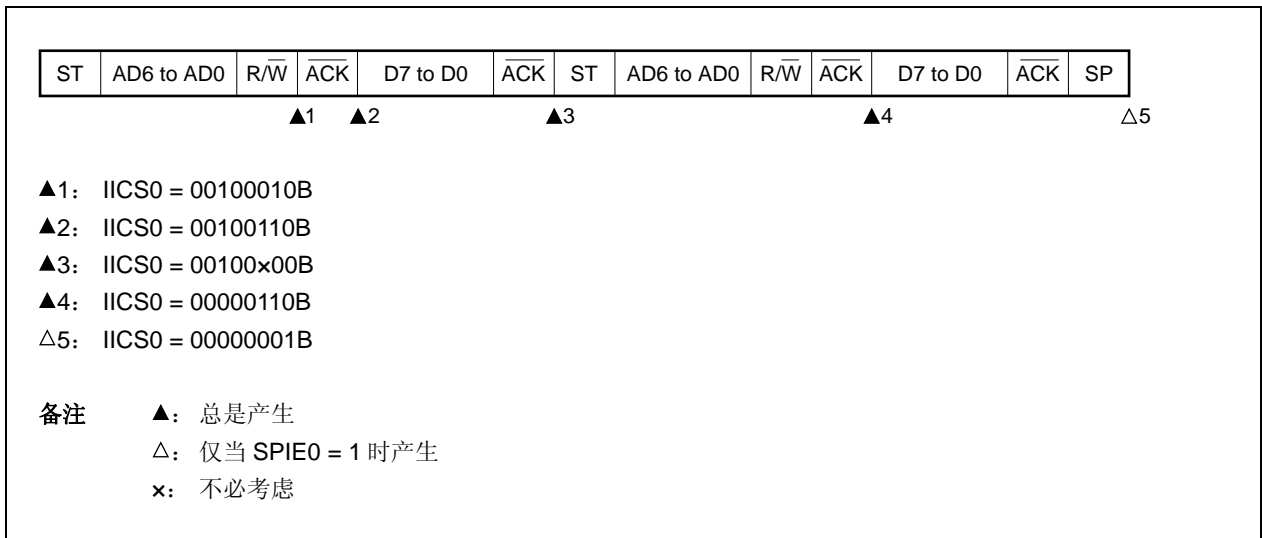


(d) 起始 ~ 代码 ~ 数据 ~ 起始 ~ 地址 ~ 数据 ~ 停止

(i) 当 **WTIMO = 0** (重启后, 和地址不匹配 (= 非扩展码))

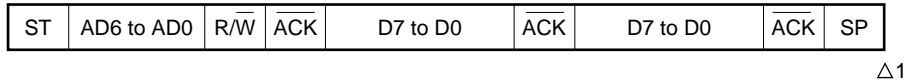


(ii) 当 **WTIMO = 1** (重启后, 和地址不匹配 (= 非扩展码))



(4) 没有通信的操作

(a) 起始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



△1: IICS0 = 00000001B

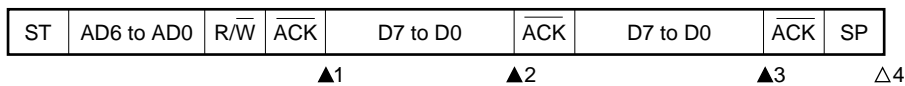
备注 △: 仅当 SPIE0 = 1 时产生

(5) 仲裁失利操作 (仲裁失利后作为从设备操作)

在多主系统中作为主设备, 每次产生中断请求信号 INTIIC0 时, 读取 MSTS0 位来检测仲裁结果。

(a) 在从地址数据发送时产生仲裁失利

(i) 当 WTIMO = 0



▲1: IICS0 = 0101x110B

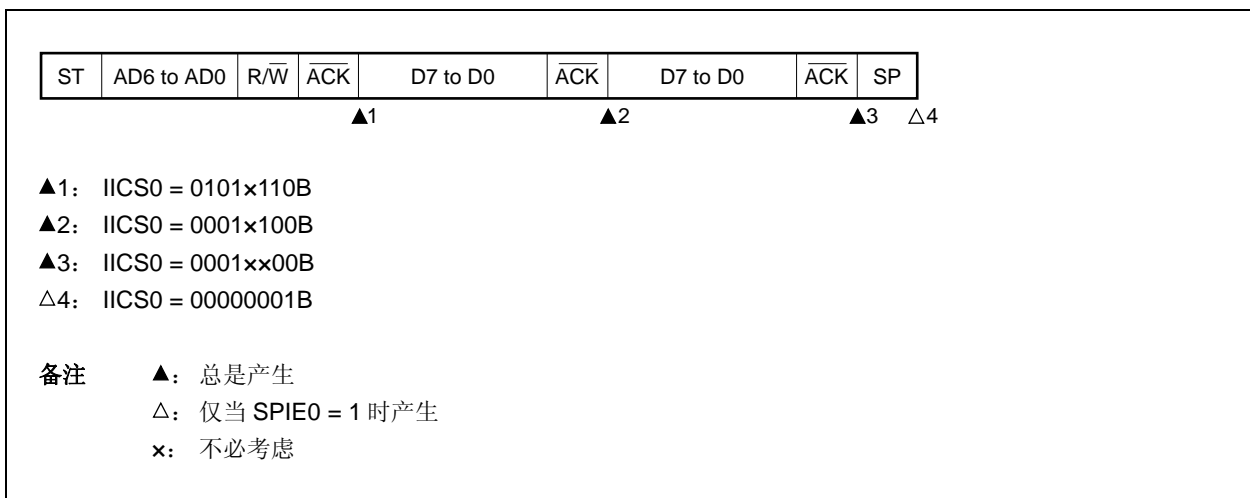
▲2: IICS0 = 0001x000B

▲3: IICS0 = 0001x000B

△4: IICS0 = 00000001B

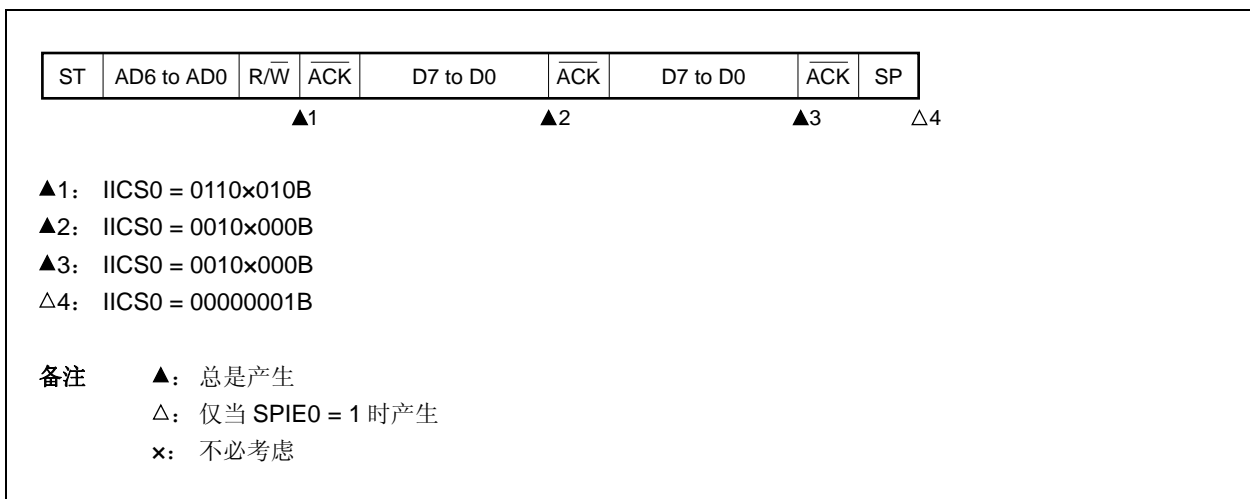
备注 ▲: 总是产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(ii) 当 **WTIM0 = 1**

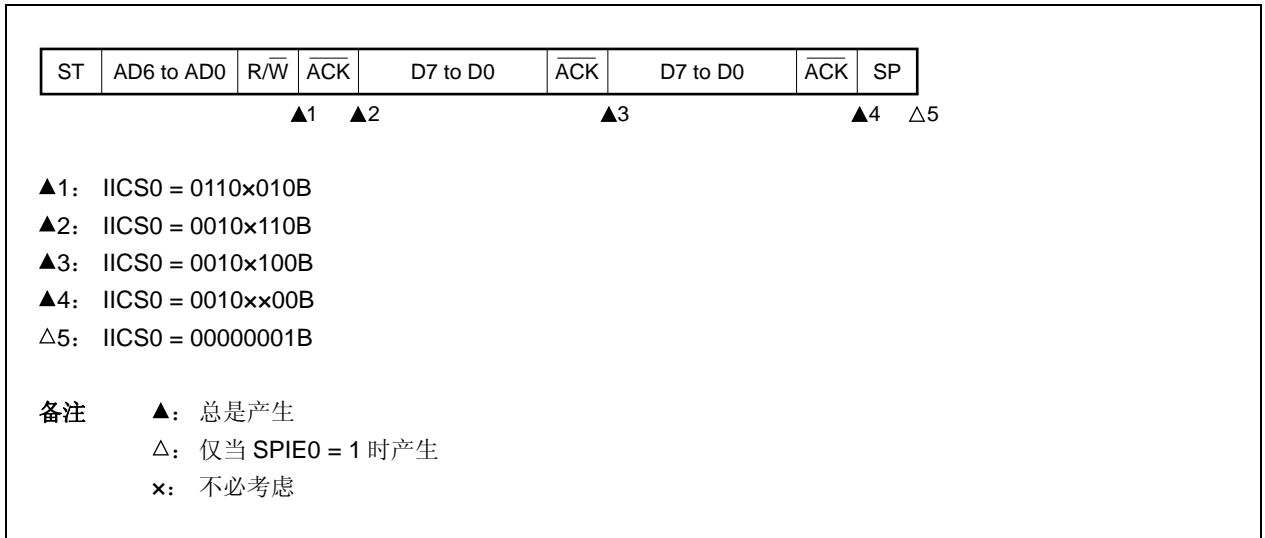


(b) 在扩展码发送时产生仲裁失利

(i) 当 **WTIM0 = 0**



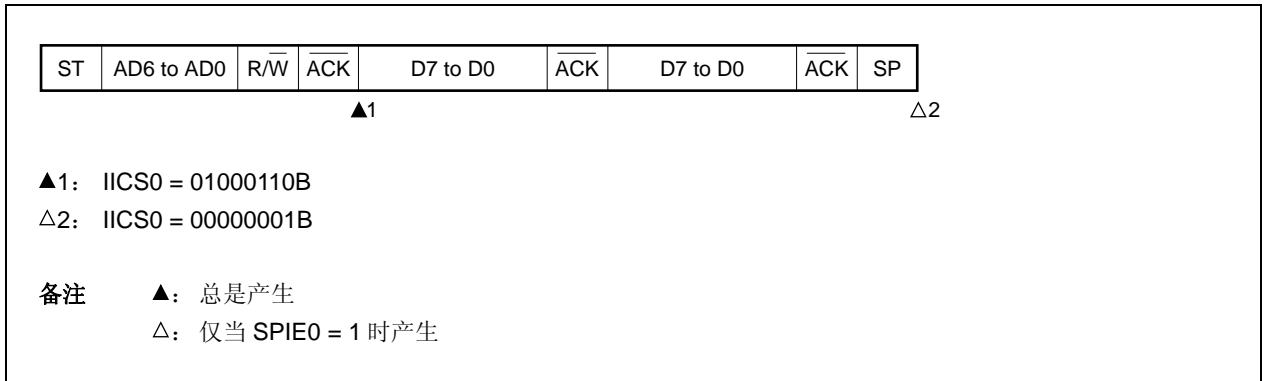
(ii) 当 **WTIMO = 1**



(6) 当发送仲裁失利时的操作（仲裁失利后没有通信）

在多主系统中作为主设备，每次产生中断请求信号 INTIIC0 时，读取 MSTS0 位来检测仲裁结果。

(a) 在从地址数据发送时产生仲裁失利（当 **WTIMO = 1**）



(b) 在扩展码发送时产生仲裁失利

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1					△2

▲1: IICS0 = 0110x010B
 通过软件设置 LREL0 = 1
 △2: IICS0 = 00000001B

备注 ▲: 总是产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(c) 在数据发送时产生仲裁失利

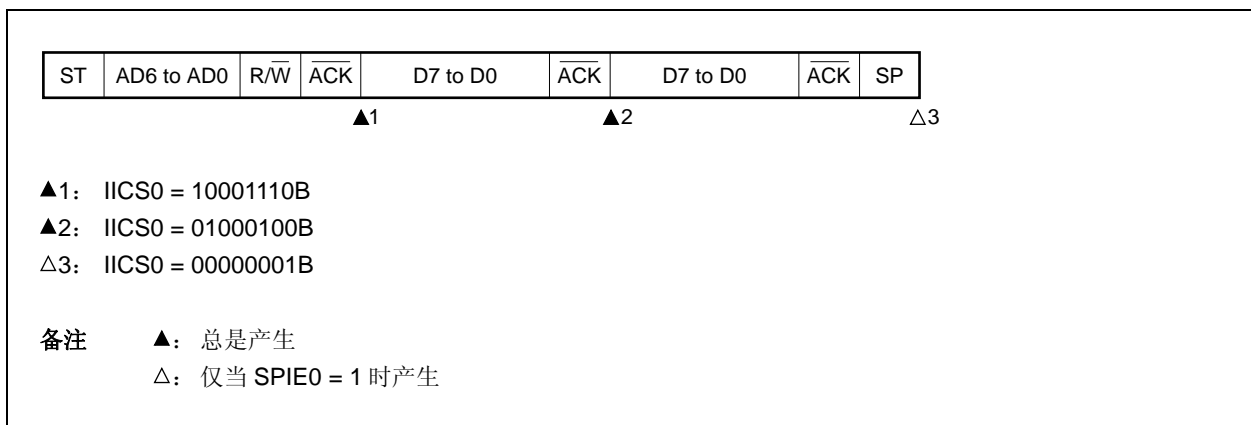
(i) 当 WTIM0 = 0

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2			△3

▲1: IICS0 = 10001110B
 ▲2: IICS0 = 01000000B
 △3: IICS0 = 00000001B

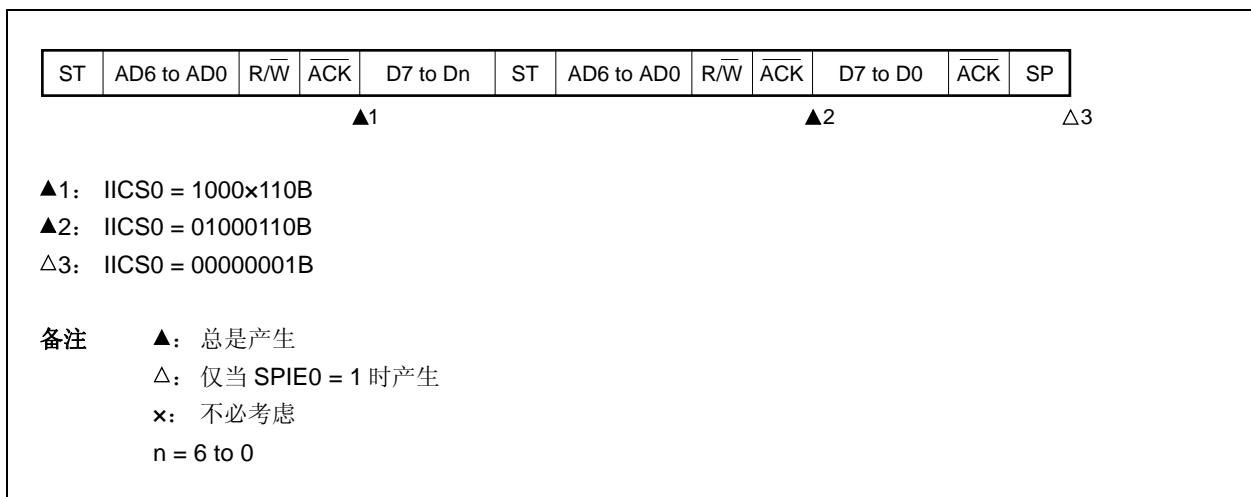
备注 ▲: 总是产生
 △: 仅当 SPIE0 = 1 时产生

(ii) 当 **WTIMO = 1**

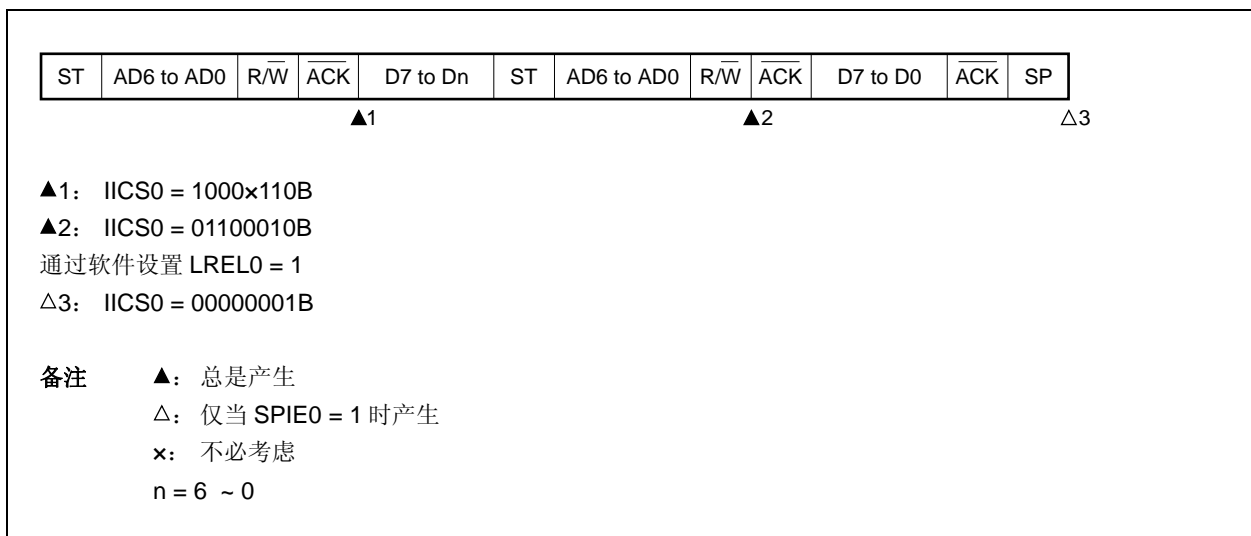


(d) 由于在数据发送时产生重启条件导致仲裁失利

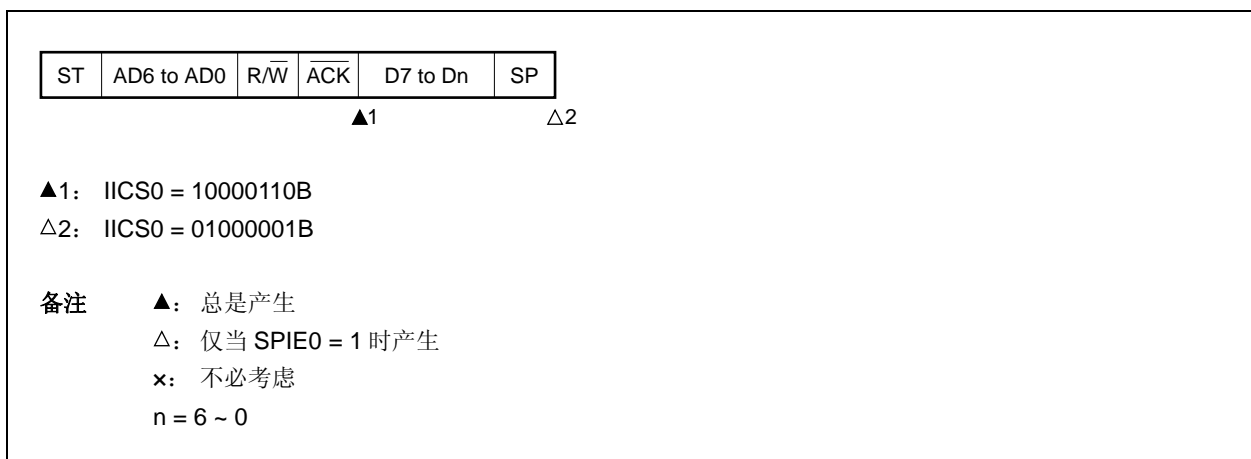
(i) 非扩展码 (例如: 和 **SVA0** 不匹配)



(ii) 扩展码

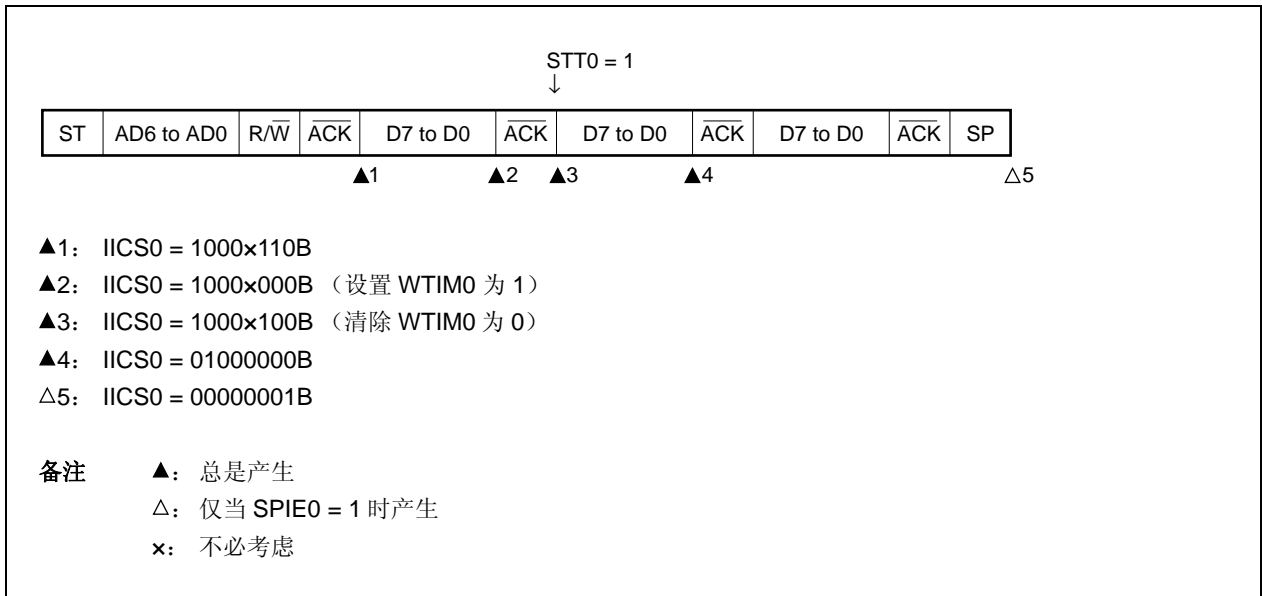


(e) 由于在数据发送时产生停止条件导致仲裁失利

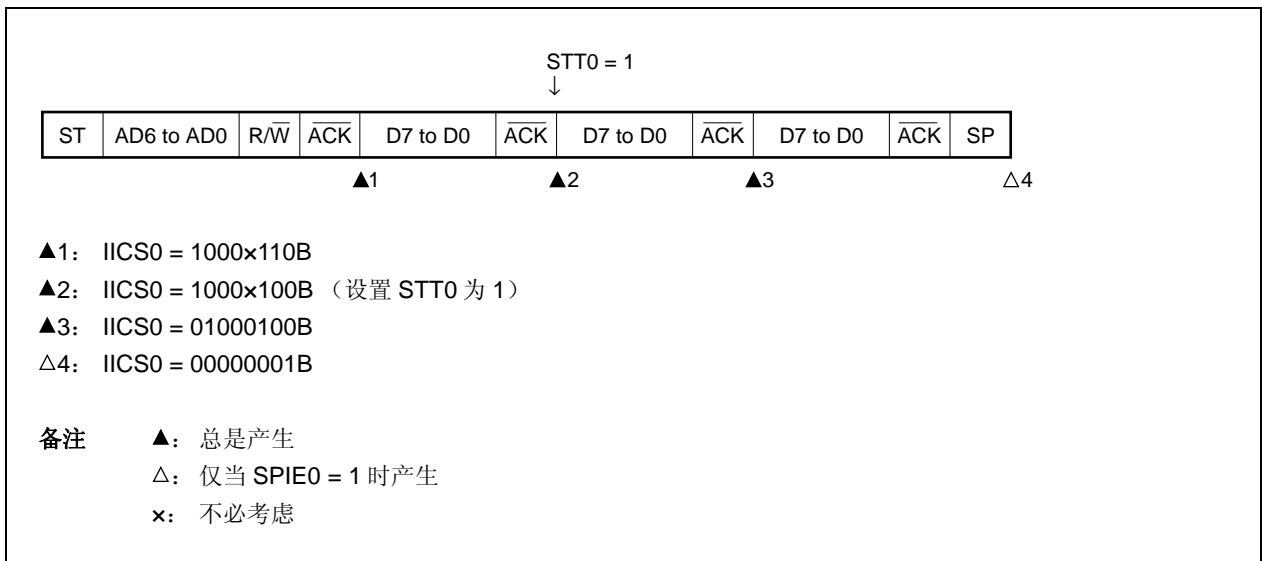


(f) 由于准备产生重启条件时的低电平数据导致仲裁失利

(i) 当 **WTIMO = 0**

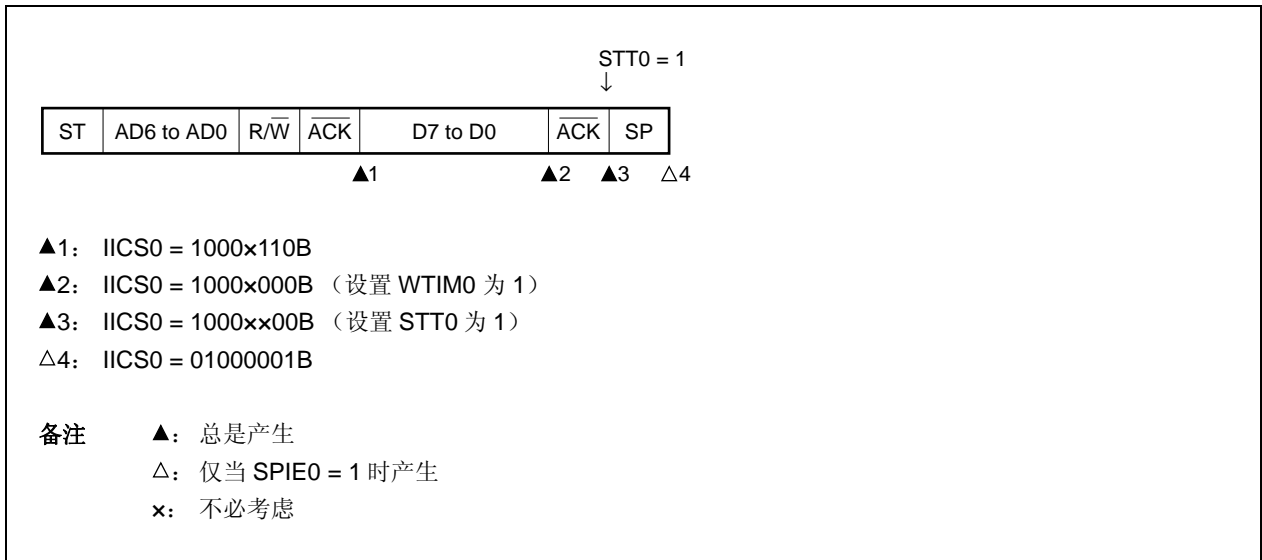


(ii) 当 **WTIMO = 1**

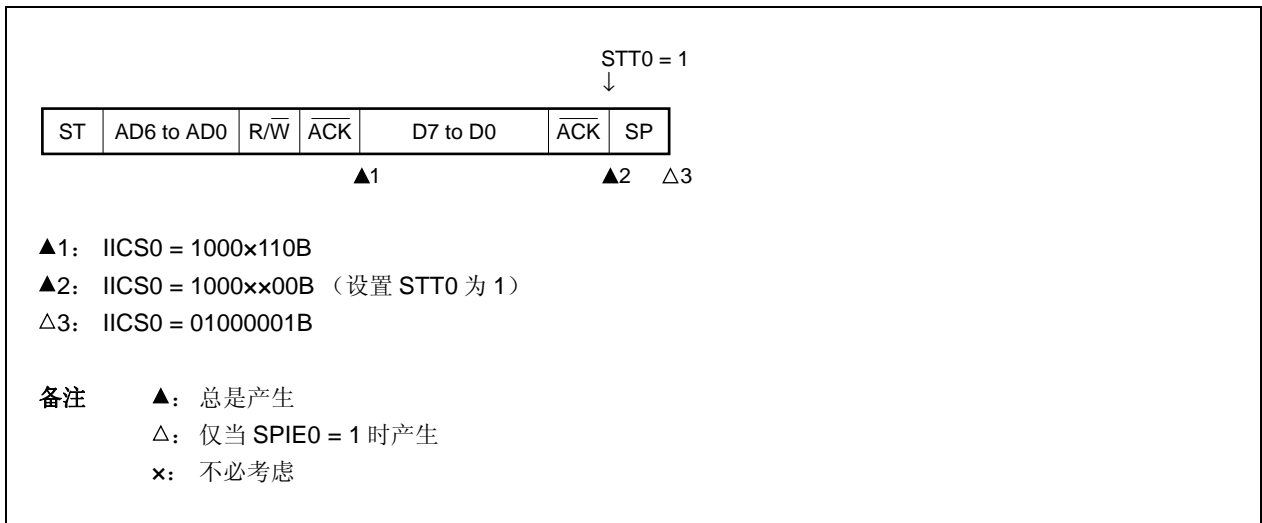


(g) 由于准备产生重启条件时的停止条件导致仲裁失利

(i) 当 **WTIMO = 0**

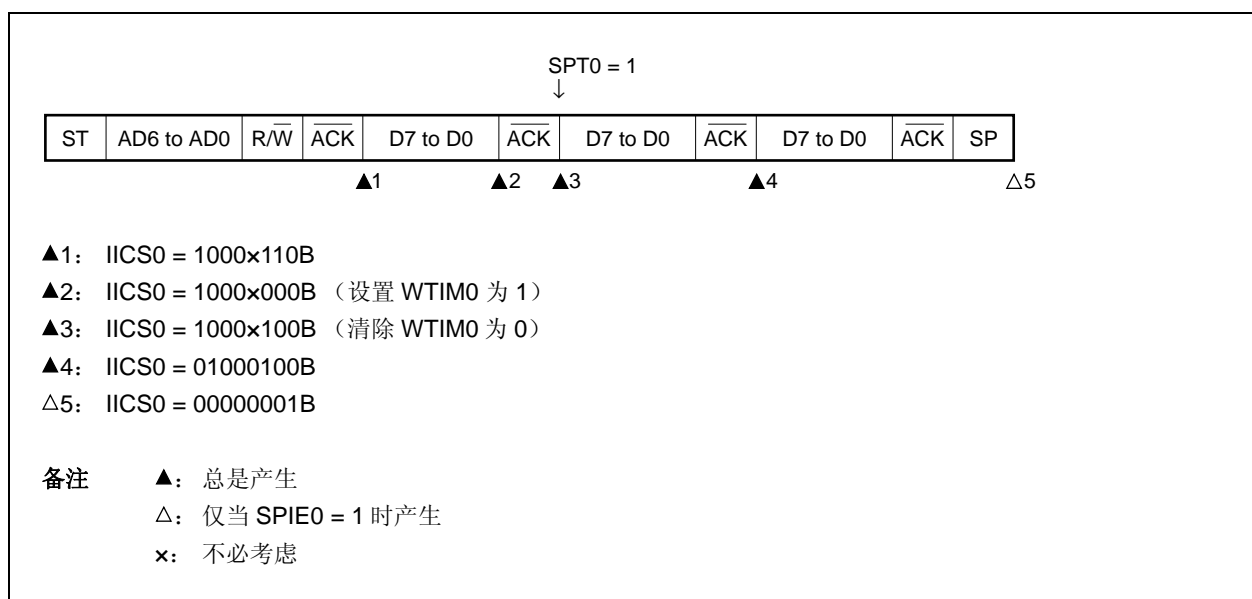


(ii) 当 **WTIMO = 1**

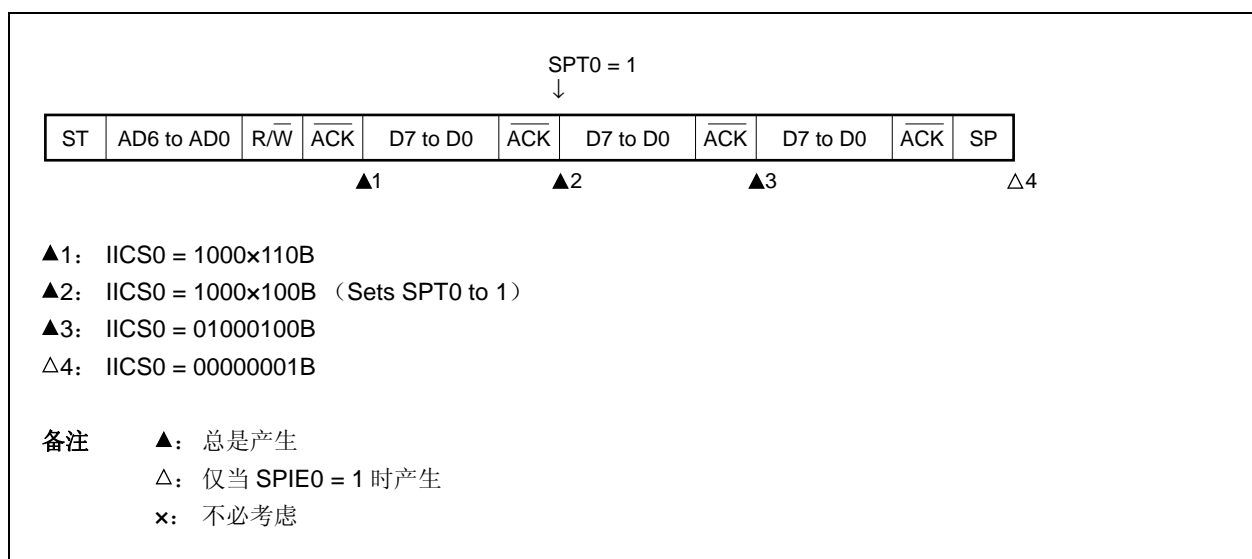


(h) 由于准备产生停止条件时的低电平数据导致产生仲裁失利

(i) 当 **WTIMO = 0**



(ii) 当 **WTIMO = 1**



13.6 时序图

使用 I²C 总线模式时，主设备通过串行总线输出一个地址从若干个从设备中选出一个作为通信对象。

输出从地址后，主设备发送 TRC0 位（IIC 状态寄存器 0（IICS0）的位 3），来指定数据发送方向，然后和从设备开始串行通信。

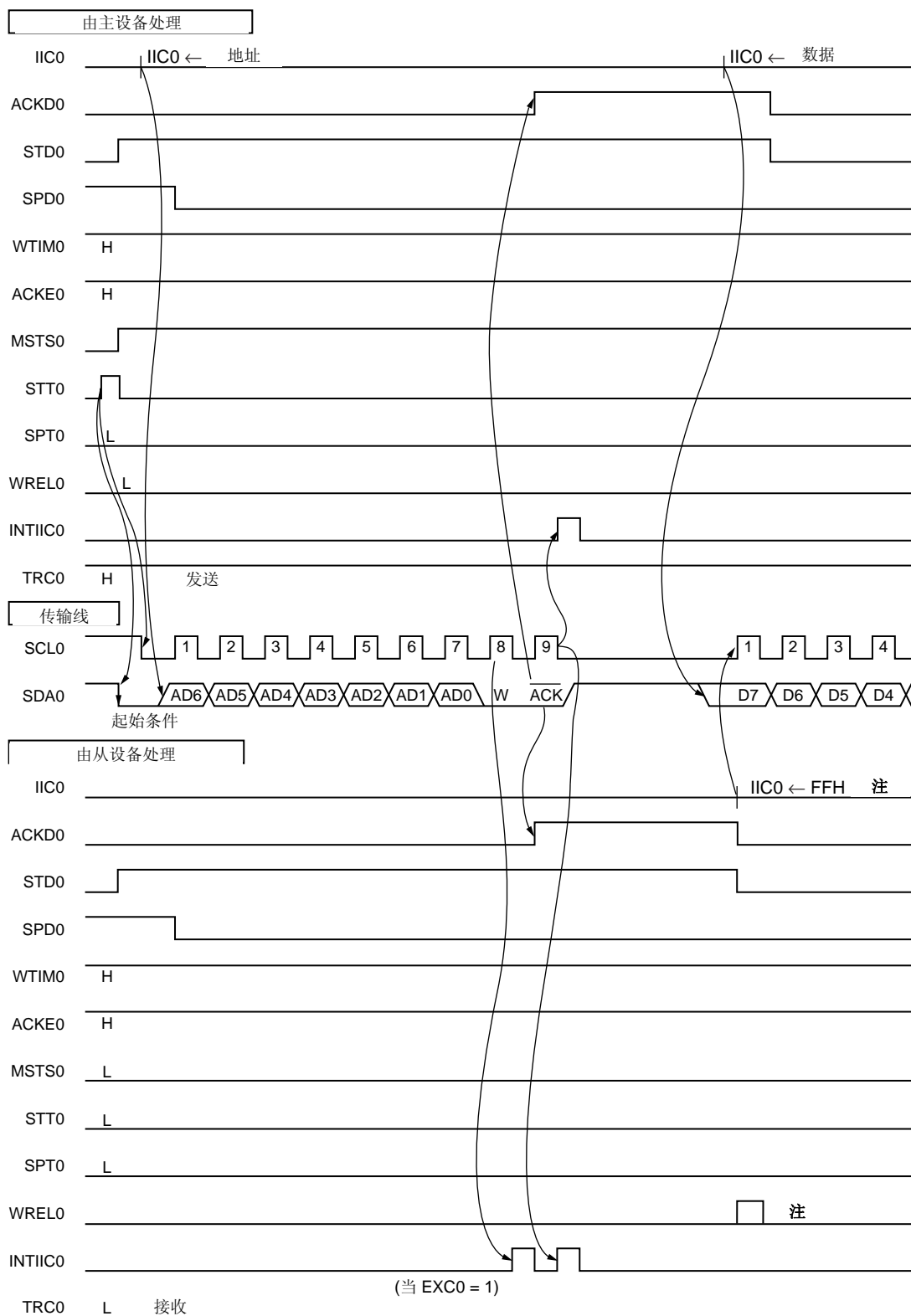
图 13-28 和图 13-29 显示数据通信的时序图。

IIC 移位寄存器 0（IIC0）的移位操作是和串行时钟（SCL0）的下降沿同步的。发送数据到 SO0 锁存并且由 SDA0 引脚输出（MSB 先行）。

通过 SDA0 引脚输入的数据在 SCL0 的上升沿被捕捉到 IIC0。

图 13-28. 主设备到从设备的通信示例
(每个主设备和从设备都选择 9 时钟等待) (1/3)

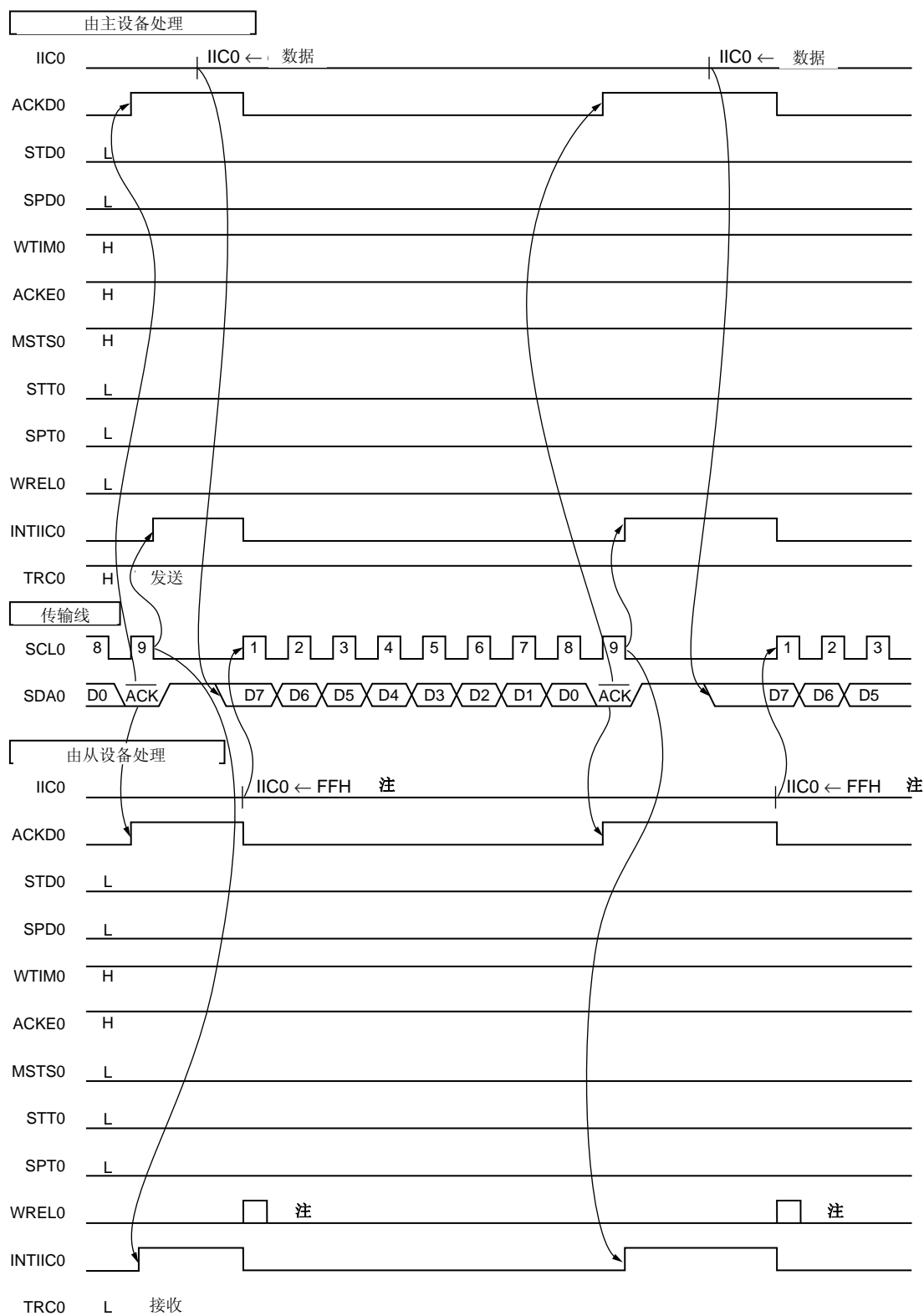
(1) 起始条件 ~ 地址



注 要取消从设备等待，写入“FFH”到 IIC0，或者设置 WRELO。

图 13-28. 主设备到从设备的通信示例
(每个主设备和从设备都选择 9 时钟等待) (2/3)

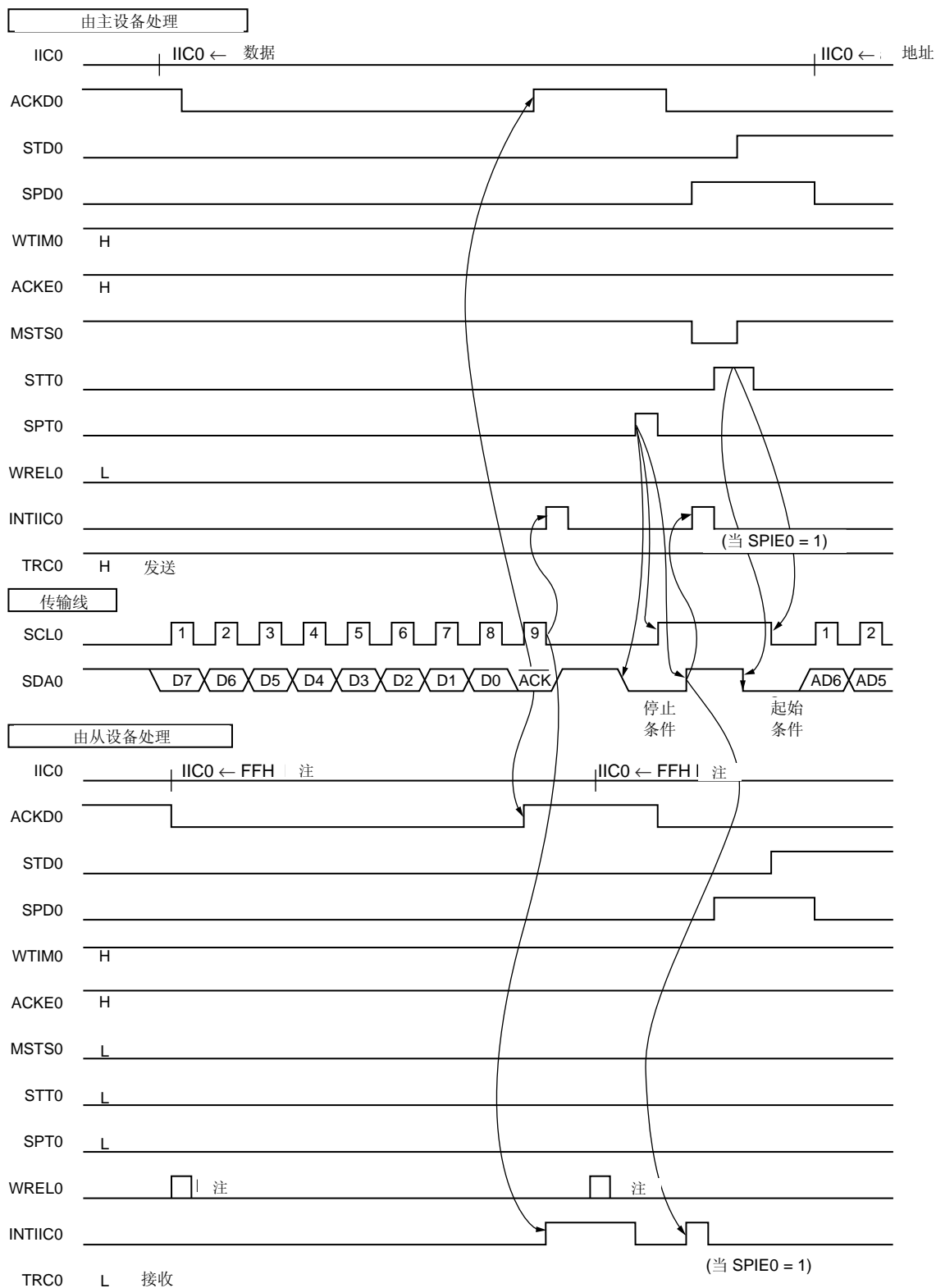
(2) 数据



注 要取消从设备等待，写入“FFH”到 IIC0，或者设置 WREL0。

图 13-28. 主设备到从设备的通信示例
(每个主设备和从设备都选择 9 时钟等待) (3/3)

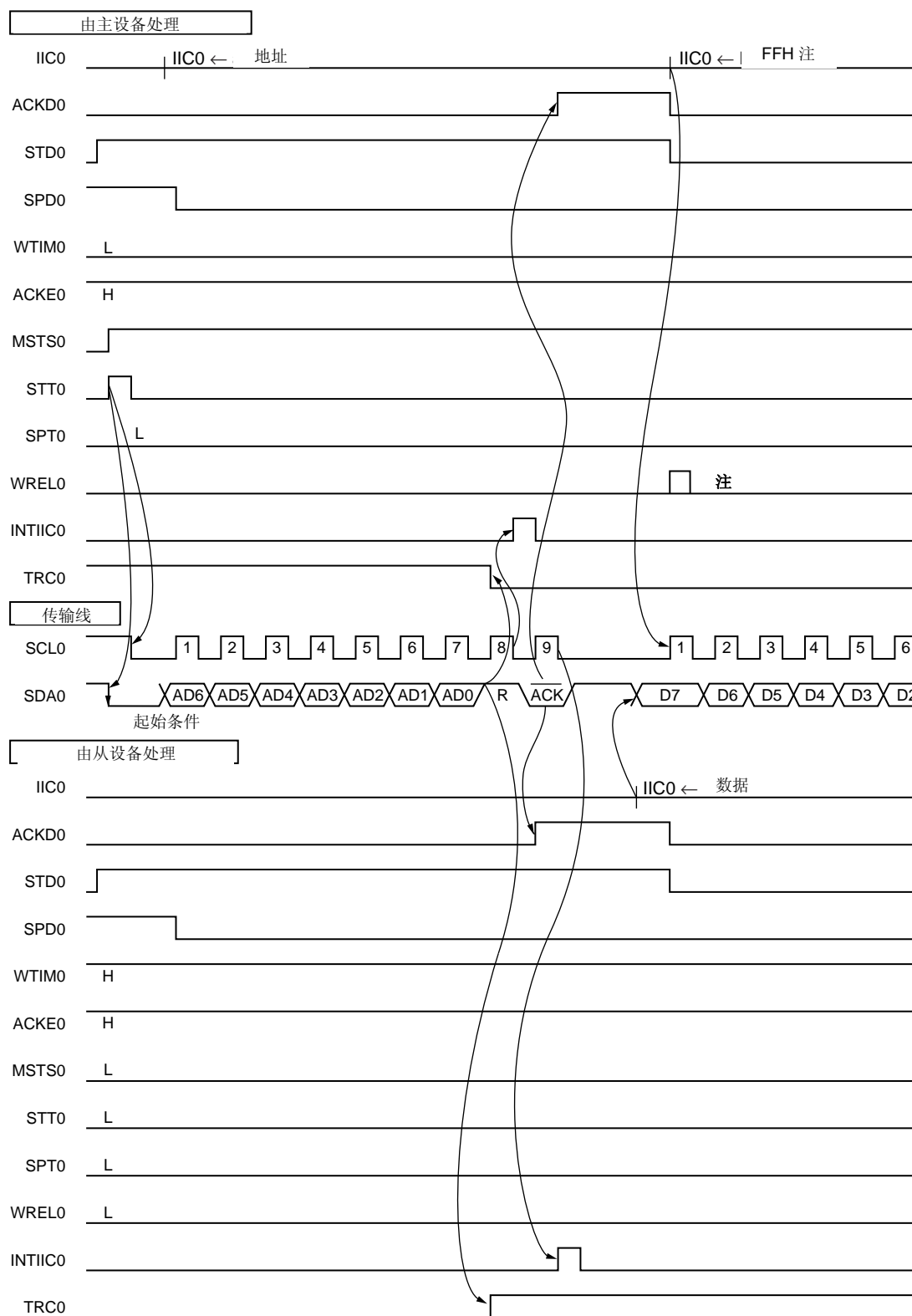
(3) 停止条件



注 要取消从设备等待，写入“FFH”到 IIC0，或者设置 WRELO。

图 13-29.从设备到主设备的通信示例
(主设备选择 8 时钟等待, 从设备选择 9 时钟等待) (1/3)

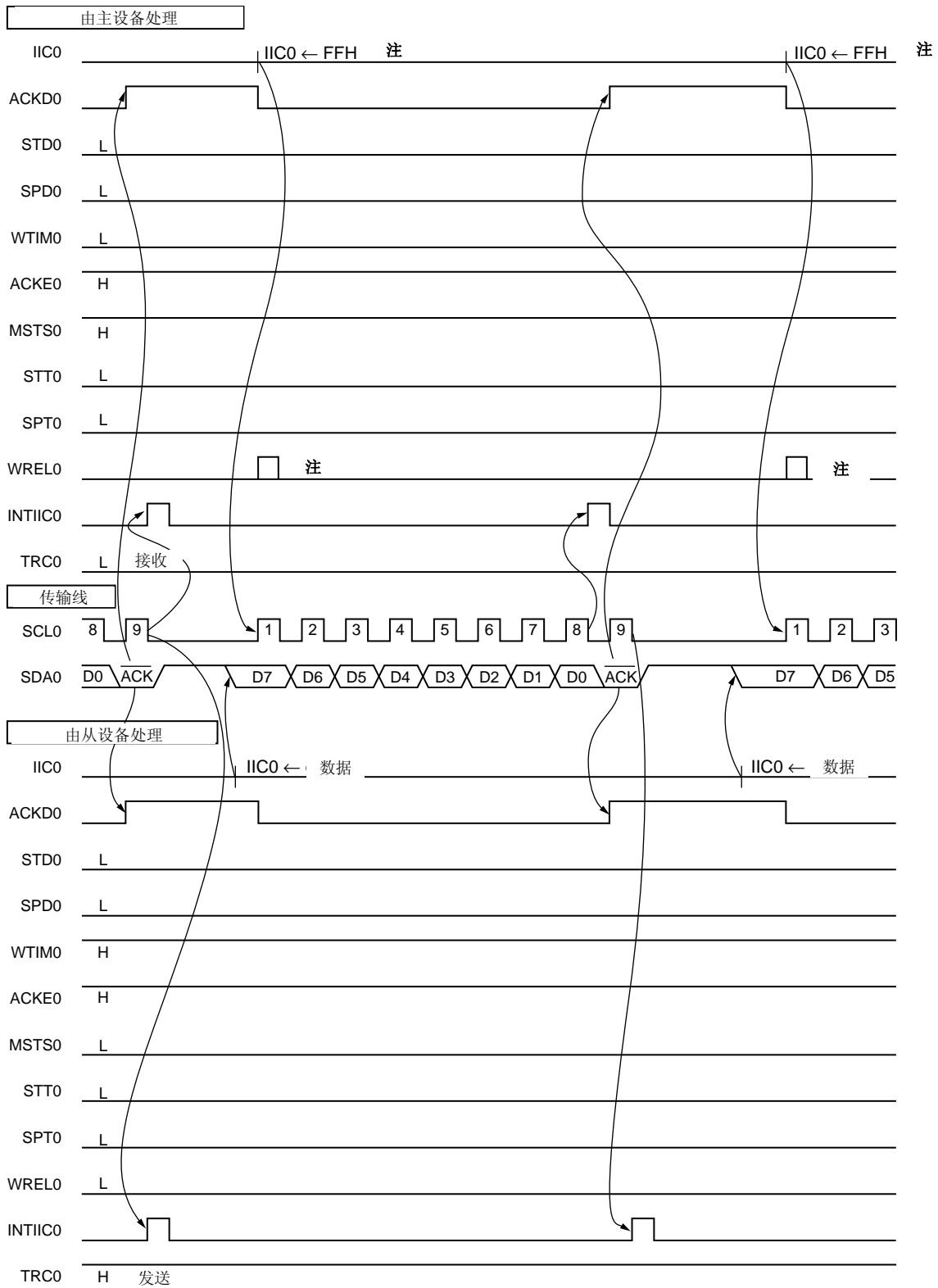
(1) 起始条件 ~ 地址



注 要取消主设备等待, 写入“FFH”到 IIC0, 或者设置 WRELO。

图 13-29.从设备到主设备的通信示例
(主设备选择 8 时钟等待, 从设备选择 9 时钟等待) (2/3)

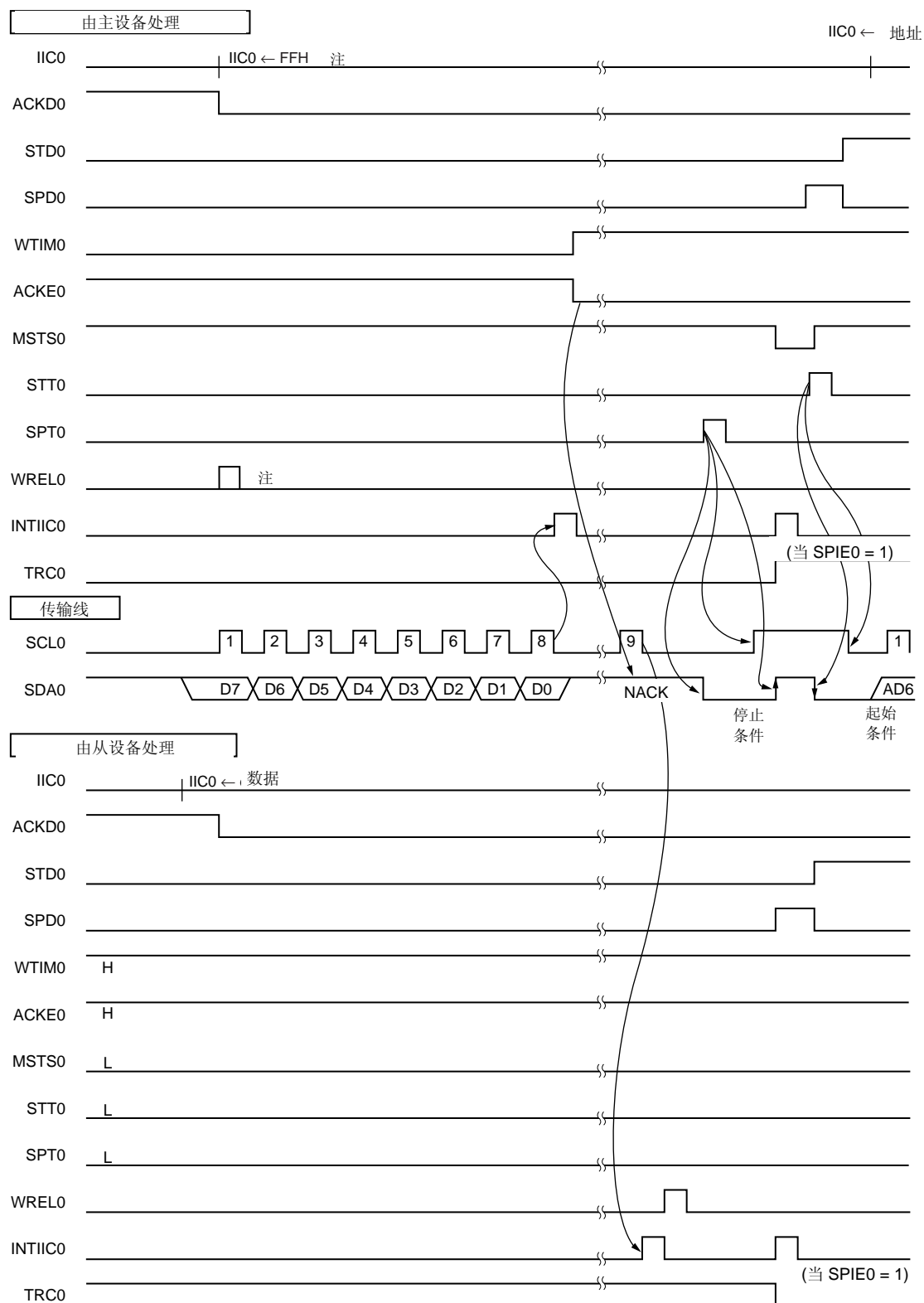
(2) 数据



注 要取消主设备等待, 写入“FFH”到 IIC0, 或者设置 WRELO。

图 13-29.从设备到主设备的通信示例
(主设备选择 8 时钟等待, 从设备选择 9 时钟等待) (3/3)

(3) 停止条件



注 要取消主设备等待, 写入“FFH”到 IIC0, 或者设置 WRELO。

第十四章 乘法器

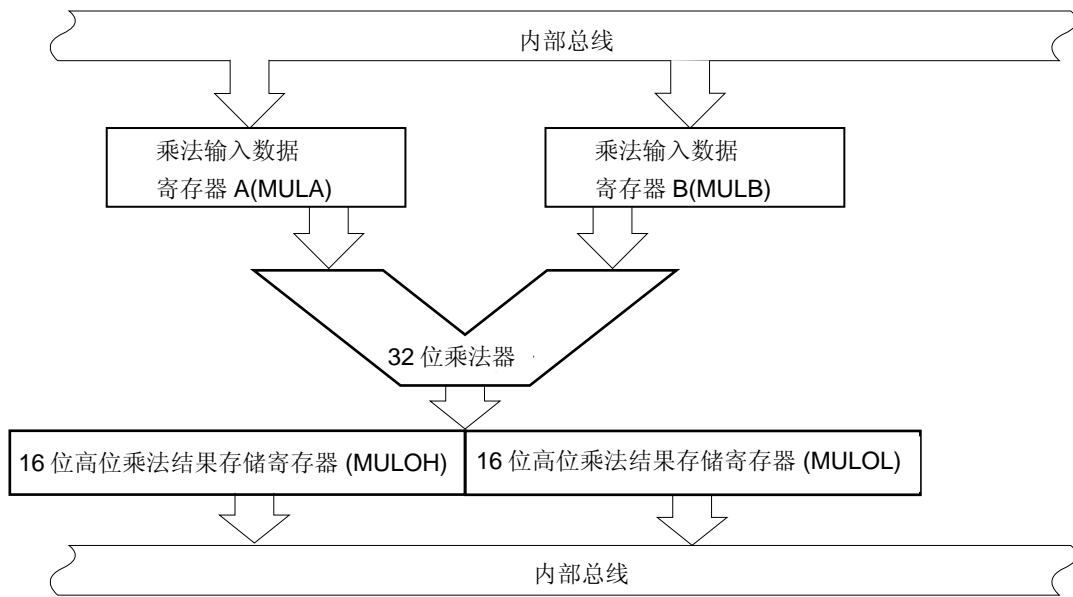
14.1 乘法器的功能

乘法器的功能如下。

- 能够执行 $16 \text{ 位} \times 16 \text{ 位} = 32 \text{ 位}$ 的乘法计算。

图 14-1 显示乘法器的框图。

图 14-1. 乘法器框图



14.2 乘法器的配置

(1) 16 位乘法结果高位存储寄存器和 16 位乘法结果低位存储寄存器 (MULOH, MULOL)

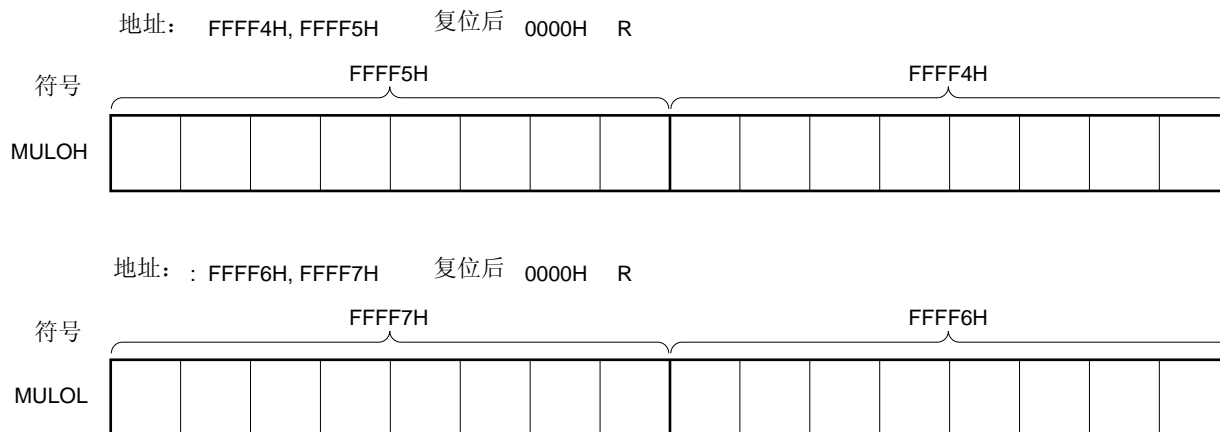
这两个寄存器，MULOH 和 MULOL，用来存储 32 位乘法的结果。乘法结果的高 16 位存储在 MULOH 中，低 16 位存储在 MULOL 中，因此可以存储 32 位乘法结果。

这些寄存器在一个 CPU 时钟之后就保留乘法的结果。

MULOH 和 MULOL 可用 16 位存储器操作指令读取。

复位信号将这些寄存器清零 0000H。

图 14-2. 16 位乘法结果高位存储寄存器和 16 位乘法结果低位存储寄存器 (MULOH, MULOL) 的格式



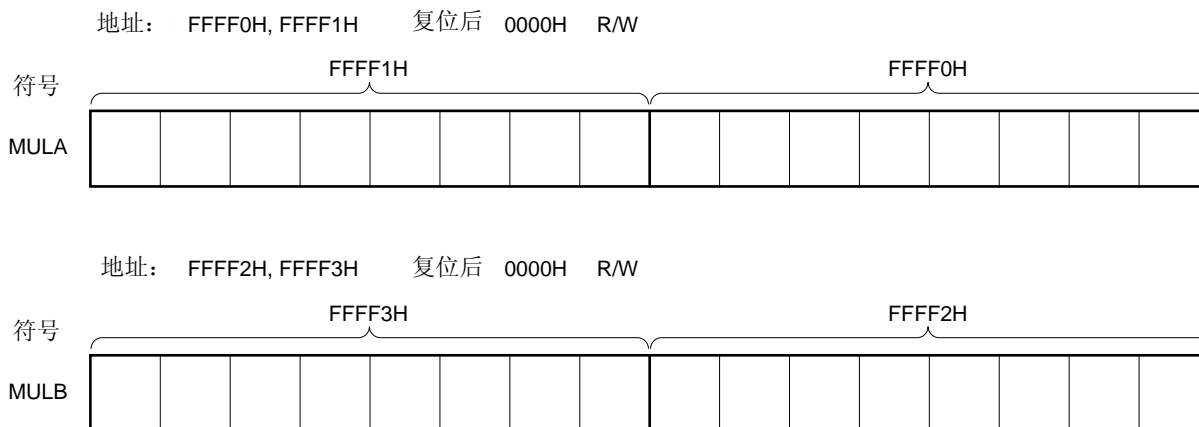
(2) 乘法输入数据寄存器 A, B (MULA, MULB)

这些是 16 位寄存器，存储参与乘法计算的数据。乘法器将 MULA 和 MULB 中的数值相乘。

MULA 和 MULB 可用 16 位存储器操作指令进行设置。

复位信号将这些寄存器清零 0000H。

图 14-3. 乘法输入数据寄存器 A, B (MULA, MULB) 的格式



14.3 乘法器的操作

通过在 MULA 和 MULB 寄存器中存储数据，并在等待 1 个时钟之后，读取 MULOH 和 MULOL 寄存器即可获得乘法结果。即使在 MULA 或 MULB 中固定一个重写另一个的情况下，结果也可以在 1 个时钟或者更多个时钟之后获得。不管 MULOH 或 MULOL 哪个在先，都可以正确读取。

源程序如下。

示例

```
MOVW    MULA, #1234H
MOVW    MULB, #5678H
NOP                                ;等待1个时钟。不一定为NOP
MOVW    AX, MULOH                  ;获得高16位
PUSH    AX
MOVW    AX, MULOL                  ;获得低16位
```

第十五章 DMA控制器

78K0R/KF3 具有内部 DMA（存储器直接访问）控制器。

数据能够在外围硬件比如 DMA、SFRs 和内部 RAM 之间自动传输，而不通过 CPU。

这样，CPU 的正常内部操作和数据传输能够与 SFR 和内部 RAM 之间的传输同时执行，因此能够处理大容量的数据。另外，使用通信、定时器和 A/D 的实时控制也可以实现实时控制。

15.1 DMA 控制器

- DMA 通道个数： 2
- 传输单位： 8 位或 16 位
- 最大传输单元： 1024 次
- 传输类型： 2 周期传输（一次传输在 2 个时钟周期内完成，传输期间 CPU 停止。）
- 传输模式： 单个传输模式
- 传输请求： 可从下列外围硬件中断中选择
 - A/D 转换器
 - 串行接口（CIS00, CSI01, CSI10, UART0, UART1, UART3, 或 IIC10）
 - 定时器（通道 0, 1, 4, 或 5）
- 传输目标： 在 SFR 和内部 RAM 之间

这里是使用 DMA 的示例。

- 串行接口的连续传输
- 模拟数据的批传输
- 固定间隔捕获 A/D 转换结果
- 固定间隔捕获端口数值

15.2 DMA 控制器的配置

DMA 控制器包括下列硬件。

表 15-1. DMA 控制器的配置

项目	配置
地址寄存器	<ul style="list-style-type: none"> • DMA SFR 地址寄存器 0, 1 (DSA0, DSA1) • DMA RAM 地址寄存器 0, 1 (DRA0, DRA1)
计数寄存器	<ul style="list-style-type: none"> • DMA 字节计数寄存器 0, 1 (DBC0, DBC1)
控制寄存器	<ul style="list-style-type: none"> • DMA 模式控制寄存器 0, 1 (DMC0, DMC1) • DMA 操作控制寄存器 0, 1 (DRC0, DRC1)

(1) DMA SFR 地址寄存器 n (DSAn)

该寄存器为 8 位寄存器，用来设置 DMA 通道 n 的传输源或者传输目的地的 SFR 地址。

SFR 地址的低 8 位设置在 FFF00H ~ FFFFFH^注之间。

此寄存器并不会自动增加，而是固定为一个特定的数值。

16 位传输模式中，忽略最低有效位并当作偶地址。

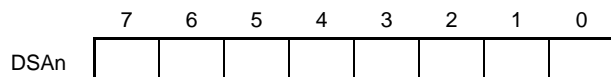
DSAn 可以进行 8 位读写。但是，DMA 传输进行期间不能写入。

复位信号将此寄存器设置为 00H。

注 除了地址 FFFFEH，因为 PMC 寄存器在此存储。

图 15-1. DMA SFR 地址寄存器 n (DSAn) 的格式

地址: FFFB0H (DSA0), FFFB1H (DSA1) 复位后: 00H R/W



备注 n: DMA 通道编号 (n = 0, 1)

15.3 控制DMA 控制器的寄存器

DMA 控制器由下列寄存器控制。

- DMA 模式控制寄存器 n (DMCn)
- DMA 操作控制寄存器 n (DRCn)

备注 n: DMA 通道编号 (n = 0, 1)

(1) DMA 模式控制寄存器 n (DMCn)

DMCn 寄存器用来设置 DMA 通道 n 的传输模式。它用来选择传输方向、数据大小、挂起设置和开始来源。第 7 位 (STGn) 是 DMA 的软件启动触发。

操作期间 (当 DSTn = 1 时), DMCn 的第 6, 5, 和 3 ~ 0 位禁止重写。

DMCn 可以通过 1 位或 8 位存储器操作指令设置。

复位信号将此寄存器设置为 00H。

图 15-4. DMA 模式控制寄存器 n (DMCn) (1/2) 的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn [#]	DMA 传输开始软件触发
0	无触发操作
1	当 DMA 操作允许时 (DENn = 1), DMA 传输开始。
当 DMA 操作允许时 (DENn = 1), 通过在此位写入 1, 开始 DMA 传输。 当读取此位时, 读取结果总是为 0。	

DRSn	选择 DMA 传输方向
0	SFR 到内部 RAM
1	内部 RAM 到 SFR

DSn	制定 DMA 传输的传输数据大小
0	8 位
1	16 位

DWAITn	DMA 传输的挂起
0	DMA 启动请求时, 执行 DMA 传输 (不挂起)。
1	如果有, 挂起 DMA 开始请求。
可以通过对清 DWAITn 数值为 0 来启动挂起的 DMA 传输。 当 DWAITn 数值设置为 1 时, 要花 2 个时钟来真正挂起 DMA 传输。	

注 不管 IFCn0 ~ IFCn3 数值如何, 软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

图 15-4. DMA 模式控制寄存器 n (DMCn) (2/2) 的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	选择 DMA 开始来源 [‡]	
				触发信号	触发内容
0	0	0	0	-	由中断禁止 DMA 传输 (只允许软件触发。)
0	0	1	0	INTTM00	定时器通道 0 中断
0	0	1	1	INTTM01	定时器通道 1 中断
0	1	0	0	INTTM04	定时器通道 4 中断
0	1	0	1	INTTM05	定时器通道 5 中断
0	1	1	0	INTST0/INTCSI00	UART0 发送完成中断/ CSI00 传输完成中断
0	1	1	1	INTSR0/INTCSI01	UART0 接收完成中断/ CSI01 传输完成中断
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1 发送完成中断/ CSI10 传输完成中断/ IIC10 传输完成中断
1	0	0	1	INTSR1	UART1 接收完成中断
1	0	1	0	INTST3	UART3 发送完成中断
1	0	1	1	INTSR3	UART3 接收完成中断
1	1	0	0	INTAD	A/D 转换完成中断
其它				禁止设置	

注 不管 IFCn0 ~ IFCn3 数值为何, 软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

(2) DMA 操作控制寄存器 n (DRCn)

DRCn 寄存器用来禁止或允许 DMA 通道 n 的传输。

操作期间 (当 DSTn = 1 时), 此寄存器的第 7 位 (DENn) 禁止重写。

DRCn 可以通过 1 位或 8 位存储器操作指令设置。

复位信号将此寄存器设置为 00H。

图 15-5. DMA 操作控制寄存器 n (DRCn) 的格式

地址: FFFBCH (DRC0), FFFBDH (DRC1) 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA 操作允许标志
0	禁止 DMA 通道 n 的操作 (停止 DMA 的操作时钟)。
1	允许 DMA 通道 n 的操作。
DMA 操作允许后 (DENn = 1), 当 DSTn = 1 时, DMAC 等待 1 个 DMA 触发。	

DSTn	DMA 传输模式标志
0	DMA 通道 n 的 DMA 传输完成。
1	DMA 通道 n 的 DMA 传输没有完成 (执行中)。
DMA 操作允许后 (DENn = 1), 当 DSTn = 1 时, DMAC 等待 1 个 DMA 触发。 当有软件触发输入, 或由 IFCn3 ~ IFCn0 设置的 (STGn) 启动源触发输入时, DMA 传输开始。 之后, 当 DMA 传输完成时, 此位自动清零 0。 对此位清 0, 将会强制中止正在执行的 DMA 传输。	

注意事项 当 DMA 传输完成时, DSTn 标志自动清零 0。

只有当 DSTn = 0 时, 允许写入 DENn 标志。当没有 DMA 中断 (INTDMA n) 的产生而中止 DMA 传输时, 要设置 DSTn 为 0, 然后设置 DENn 为 0 (详细信息, 参考 15.5.5 软件强行中止)。

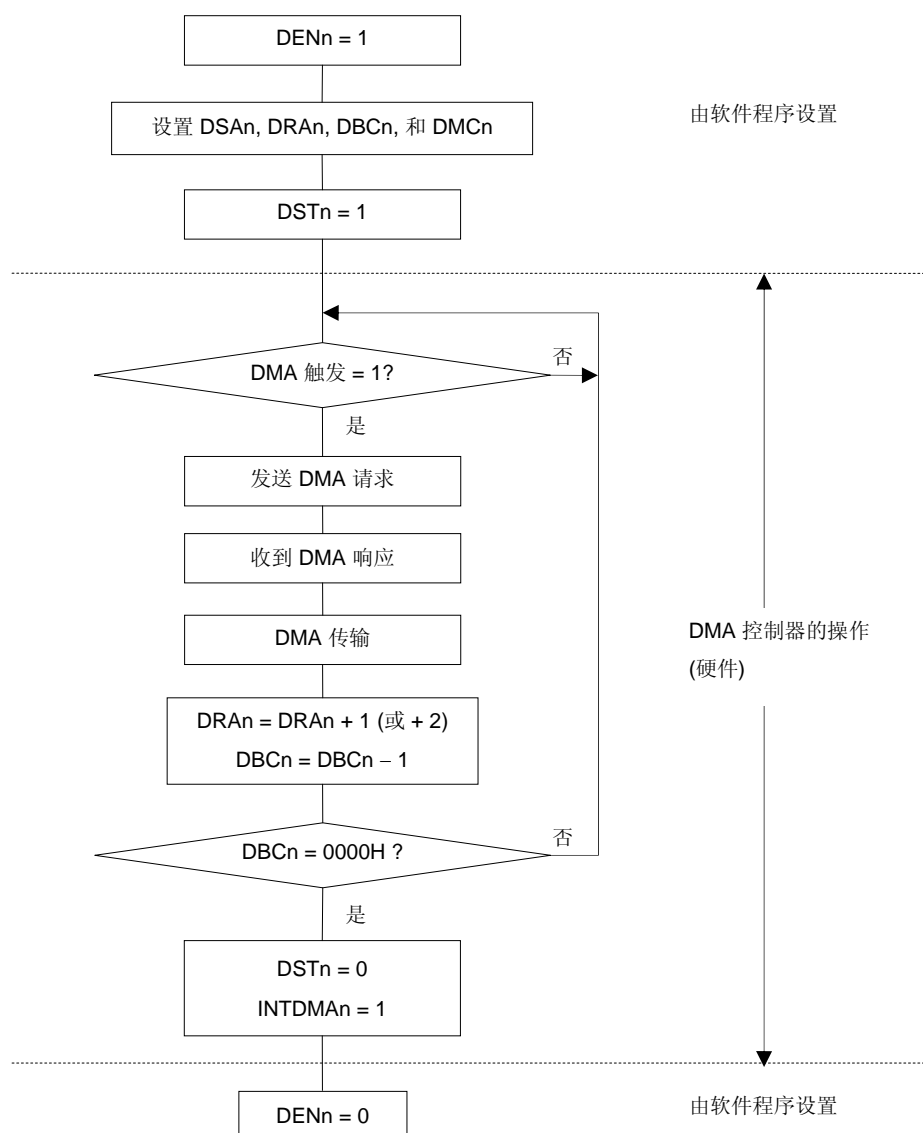
备注 n: DMA 通道编号 (n = 0, 1)

15.4 DMA 控制器的操作

15.4.1 操作顺序

- <1> 当 DENn = 1 时，DMA 控制器允许操作。其它寄存器写入之前，一定要将 DENn 设置为 1。使用 8 位操作指令写入 80H。
- <2> 设置 1 个 SFR 地址、1 个 RAM 地址、传输次数和 DMA 传输模式到 DSA_n，DRA_n，CBC_n，和 DMC_n 寄存器。
- <3> 当 DST_n = 1 时，DMA 控制器等待 DMA 触发。使用 8 位操作指令写入 81H。
- <4> 当有软件触发输入，或由 IFC_n3 ~ IFC_n0 设置的 (STG_n) 启动源触发输入时，DMA 传输开始。
- <5> 当 DBC_n 寄存器中的传输次数变为 0 时，传输完成，并且产生一个中断 (INTDMA_n)，传输自动停止。
- <6> 不使用 DMA 控制器时，将 DENn 清零 0，就可以停止 DMA 控制器的操作。

图 15-6. 操作顺序



备注 n: DMA 通道编号 (n = 0, 1)

15.4.2 传输模式

下面 4 种模式通过使用 DMCn 寄存器的第 6 位和第 5 位 (DRSn 和 DS_n) 选择 DMA 传输的模式。

DRSn	DS _n	DMA 传输模式
0	0	从SFR 传输1字节数据的 (固定地址) 到 RAM (地址每次+1增加)
0	1	从SFR 传输1字节数据的 (固定地址) 到 RAM (地址每次 +2增加)
1	0	从RAM 传输1字节数据的 (地址每次 +1增加) 到 SFR (固定地址)
1	1	从RAM 传输1字节数据的 (地址每次 +2增加) 到 SFR (固定地址)

使用这些传输模式, 最多 1024 个字节数据可以通过串行接口连续传输, A/D 转换的结果可以进行传输, 使用定时器在固定的时间间隔内扫描到的端口数据也可以进行传输。

15.4.3 DMA 传输中止

DBCn = 00H 且 DMA 传输完成时, DSTn 位自动清零 0。产生中断请求 (INTDMAn), 并且传输中止。

当 DSTn 位清零强制中止 DMA 传输时, DBCn 和 DRAn 寄存器保存传输中止时的数值。

如果强行中止传输, 则不产生中断请求 (INTDMAn)。

备注 n: DMA 通道编号 (n = 0, 1)

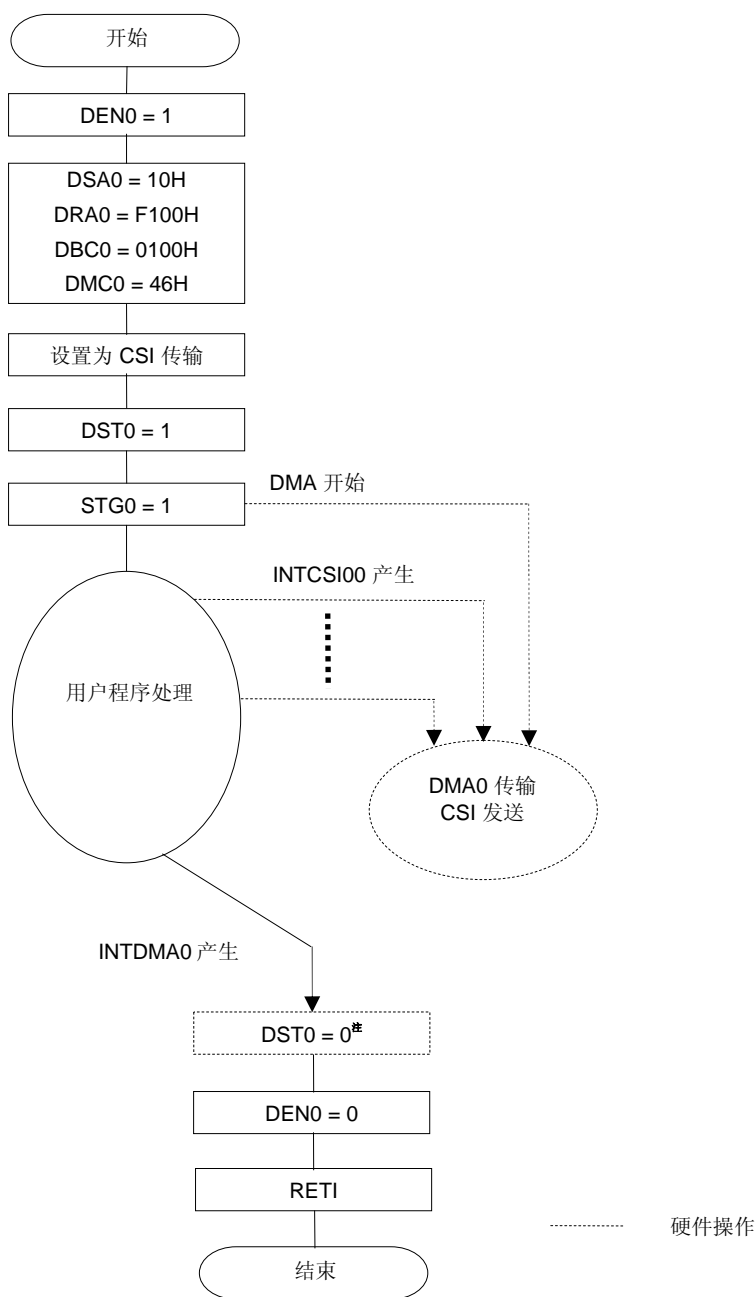
15.5 DMA控制器设置的示例

15.5.1 CSI 连续发送

设置 CSI 连续发送的示例流程图如下所示。

- CSI00 的连续发送
- DMA 通道 0 用作 DMA 传输
- DMA 开始来源: INTCSI00 (软件触发 (STG0) 只用于第一个开始源)
- CSI00 中断由 IFC03 ~ IFC00 (DMC0 寄存器的第 3~0 位) = 0110B 来指定。
- 把 RAM 的 FF100H ~ FF1FFH (256 个字节) 传输到 CSI 发送缓冲器 (SIO00) 的 FFF10H。

图 15-7. CSI 连续发送的设置示例



注 当 DMA 传输完成时，DST0 标志自动清 0。
 只有当 DST0 = 0 时，允许写入 DENO 标志。为了在 DMA0（INTDMA0）中断产生之前中止 DMA 传输，DST0 需要设置为 0，然后把 DENO 也设置为 0（详细信息，参考 15.5.5 软件强行中断）。

连续传输的第一次触发并不由 CSI 中断启动。要由软件触发启动。

第二次及后来的 CSI 发送会自动执行。

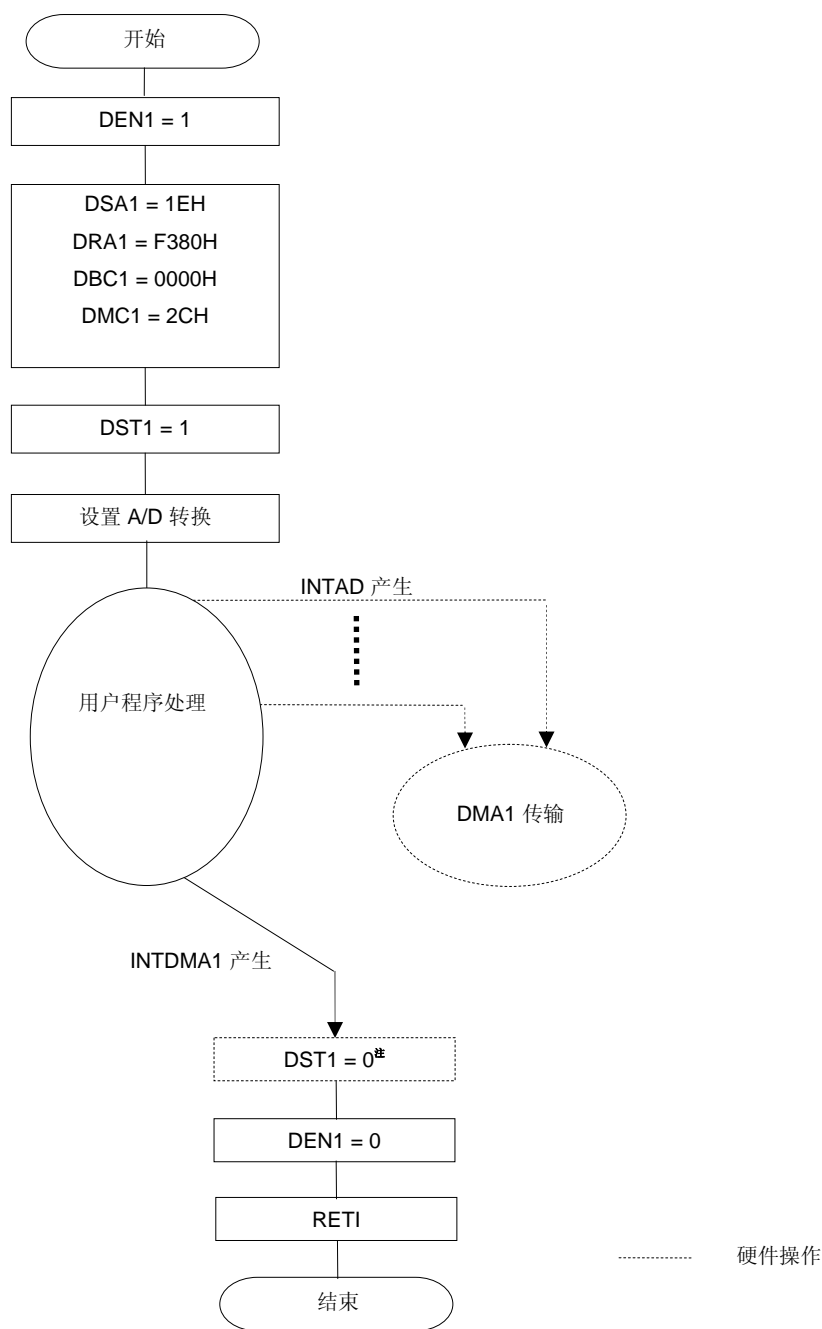
只要最后一个数据写入发送缓冲器，就会产生 DMA 中断（INTDMA0）。此时，CSI 的最后一个数据正在发送。为了再次开始 DMA 传输，需要等待 CSI 的传输完成。

15.5.2 A/D 转换结果的连续捕获

设置 A/D 转换结果连续捕获的示例流程图如下所示。

- A/D 转换结果的连续捕获。
- DMA 通道 1 用于 DMA 传输。
- DMA 开始源: INTAD
- A/D 中断由 IFC03 ~ IFC00 (DMC1 寄存器的位 3~0) = 1100B 来指定。
- 把 10 位 A/D 转换结果寄存器的 FFF1EH 和 FFF1FH (2 个字节) 传输到 RAM 的 FF380H ~ FFB7FH 的 2048 个字节中。

图 15-8. 设置 A/D 转换结果连续捕获的示例



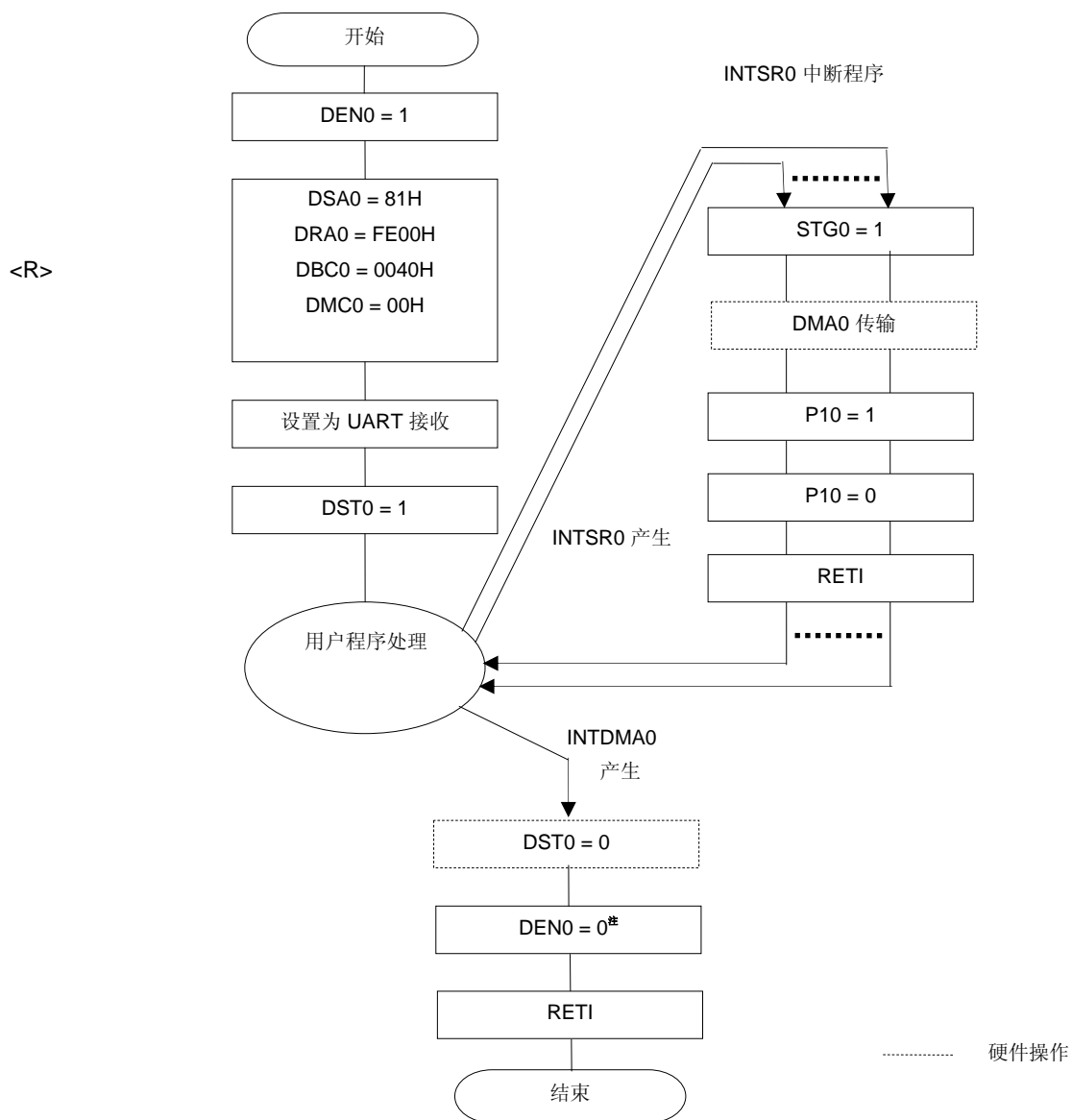
注 当 DMA 传输完成时，DST1 标志自动清 0。
 只有当 DST0 = 0 时，允许写入 DEN0 标志。为了在 DMA0 (INTDMA0) 中断产生之前中止 DMA 传输，DST0 需要设置为 0，然后把 DEN0 也设置为 0 (详细信息，参考 15.5.5 软件强行中断)。

15.5.3 UART 连续接收 + ACK 发送

设置 UART 连续接收 + ACK 发送的示例流程图如下所示。

- 从 UART0 连续接收数据，并在接收完成之后输出 ACK 到 P10。
- DMA 通道 0 用于 DMA 传输。
- DMA 开始源：软件触发（禁止以中断的产生来触发 DMA 传输。）
- 把 UART 接收数据寄存器 0（RxD0）的 FFF12H 传输给 RAM 的 FFE00H ~ FFE3FH 的 64 个字节内。

图 15-9. UART 连续接收 + ACK 发送设置的示例



注 当 DMA 传输完成时，DST0 标志自动清 0。
只有当 DST0 = 0 时，允许写入 DEN0 标志。为了在 DMA0（INTDMA0）中断产生之前中止 DMA 传输，DST0 需要设置为 0，然后把 DEN0 也设置为 0（详细信息，参考 15.5.5 软件强行中断）。

备注 这是一个软件触发用作 DMA 开始源的示例。
如果不发送 ACK，并且数据都是从 UART 中连续接收的，那么在数据接收中 UART 接收结束中断（INTSR0）可以用于启动 DMA。

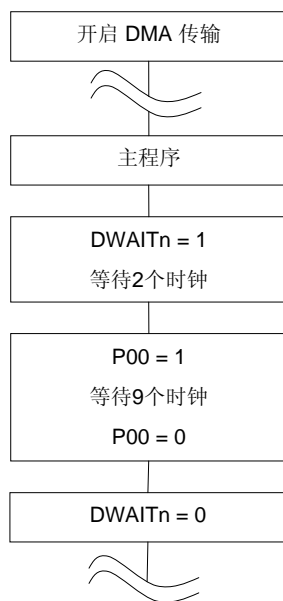
15.5.4 通过DWAITn保持DMA传输挂起

DMA 传输开始时，执行一条指令就进行传输。此时，CPU 的操作停止，并且延时 2 个时钟。如果这样对设定的系统的操作造成了问题，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

例如，为了从 P00 引脚输出一个脉冲，其宽度为操作频率 10 个时钟周期，如果在中途 DMA 传输开始，那么宽度就增加到 12 个时钟周期。这种情况下，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

设置 DWAITn 为 1 后，要花 2 个时钟才能保持 DMA 传输挂起。

图 15-10. 通过 DWAITn 保持 DMA 传输挂起设置的示例



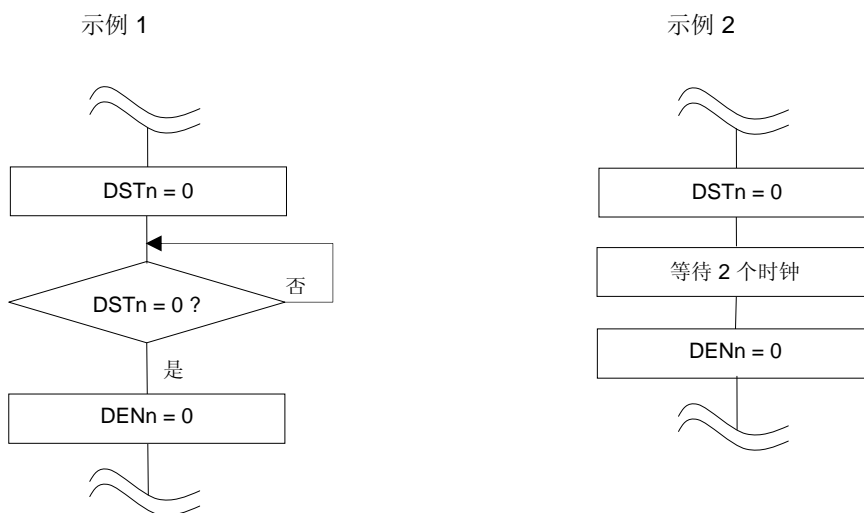
- 备注
1. n: DMA 传输号码 (n = 0, 1)
 2. 1 个时钟: 1/fCLK (fCLK: CPU 时钟)

15.5.5 软件强行中止

DSTn 由软件设置为 0 之后，要花 2 个时钟 DMA 传输才能真正停止，并且 DSTn 位被置为 0。为了在 DMA 的 (INTDMA0) 中断产生之前用软件强行中止 DMA 传输，需要执行下列操作的其中一个。

- 由软件设置 DSTn 为 0（用 8 位操作指令写入 DRCn = 80H），通过查询确认 DSTn 的确已经被清 0，然后设置 DENn 为 0（用 8 位操作指令写入 DRCn = 00H）。
- 由软件设置 DSTn 为 0（用 8 位操作指令写入 DRCn = 80H），然后在 2 个或者更多个时钟之后设置 DENn 为 0（用 8 位操作指令写入 DRCn = 00H）。

图 15-11. DMA 传输的强行中止



- 备注
1. n: DMA 通道编号码 (n = 0, 1)
 2. 1 个时钟: 1/fCLK (fCLK: CPU 时钟)

15.6 使用 DMA 控制器的注意事项

(1) DMA 的优先级

DMA 传输期间，即使产生其它 DMA 通道的请求，也保持挂起。在当前的 DMA 传输完成之后，挂起的 DMA 传输开始。在一段较短的时间内连续产生两个通道的请求，它们就连续传输，并且在完成一个通道的传输之后，就执行另一个 DMA 通道的请求。这种情况下，要在第一个 DMA 传输和随后的 DMA 传输之间执行 1 到 2 条指令。

如果两个 DMA 请求同时产生，那么通道 0 优先于通道 1。

如果 DMA 请求和中断请求同时产生，DMA 传输优先，然后执行中断服务程序。

注 较短的时间是指 8 个或者更少的 CPU 时钟。时钟长度和 DMA 操作之间的关系如下。

- 1 个时钟： 禁止设置 DMA 请求不能接受。
- 2 ~ 4 个时钟： 连续产生请求的那个通道的 DMA 传输执行。
- 5 ~ 8 个时钟： 连续产生请求的那个通道的 DMA 传输执行，还是从其它通道来的 DMA 请求执行取决于 CPU 指令的执行次数。

(2) DMA 响应时间

DMA 传输的响应时间如下。

表 15-2. DMA 传输的响应时间

	最短时间	最长时间
响应时间	4 个时钟	10 个时钟

备注 1 个时钟： $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

但是，下列情况下，DMA 传输可能会有更长的延时。DMA 传输延时的时钟数量根据条件不同而有所差异。

- 由 RAM 执行指令
- 由外部存储器执行指令
- 当访问外部存储器时，如果插入等待周期
- 执行 DMA 挂起指令

(3) 待机模式下的操作

DMA 控制器的待机模式按照下列所示进行。

表 15-3. 待机模式下 DMA 的操作

状态	DMA 操作
HALT 模式	正常操作
STOP 模式	停止操作 如果 DMA 传输和 STOP 指令执行有冲突，那么 DMA 传输就可能要被打断。因此，执行 STOP 之前，要停止 DMA。

(4) DMA 挂起指令

下列指令之后，即使产生 DMA 请求，DMA 传输也要保持挂起。

- CALL !addr16
- CALL &!addr16
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- 涉及寄存器 IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H 和 PSW 的位操作指令，以及操作数包括 ES 寄存器的 8 位操作指令。

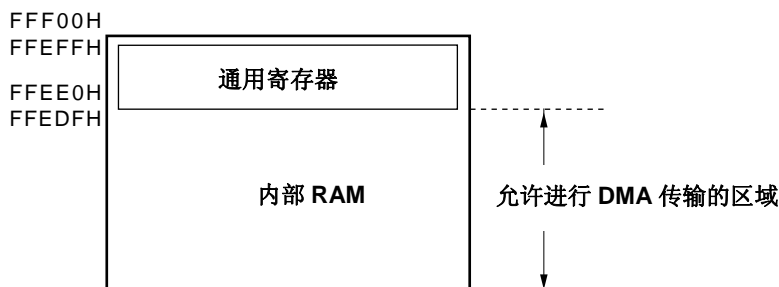
<R>

(5) 如果指定了通用寄存器区域地址或者内部 RAM 之外的区域时的操作

在 DMA 传输期间，由 DRA0n 指示的地址增加。如果地址增加到通用寄存器区域，或者超过了内部 RAM 区域，进行下列操作。

- 在从 SFR 到 RAM 的操作模式下
那个地址的数据丢失。
- 在从 RAM 到 SFR 的操作模式下
传输到 SFR 的数据不确定。

任何一种情况下，都会发生误操作或者对系统造成损坏。因此，一定要确保地址落在通用寄存器之外的内部 RAM 区域中。



第十六章 中断功能

16.1 中断功能的类型

中断功能的类型可分为以下两种。

(1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级规范标志寄存器（PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H）将可屏蔽中断分为四个优先级组。

支持多重中断嵌套，即在低优先级的中断服务子程序中可以被高优先级的中断请求打断。如果同时产生两个或两个以上的中断请求，且具有相同优先级，则根据向量中断服务的优先级进行处理。优先级顺序，请参见表 16-1。

产生一个待机释放信号，并释放 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

外部： 13，内部： 28

(2) 软件中断

这是通过执行 BRK 指令产生的一个向量中断。即使中断被禁止时也可以响应这类中断。软件中断不受中断优先级控制。

16.2 中断源及配置

78K0R/KF3 系列中，共有 42 种中断源，包括可屏蔽的中断和软件中断。另外，还具有 5 种复位源（参见表 16-1）。

表 16-1. 中断源列表（1/2）

中断类型	默认优先级 ^{注1}	中断源		内部/外部	向量表地址	基本配置类型 ^{注2}
		名称	触发器			
可屏蔽	0	INTWDTI	看门狗定时器间隔定时 ^{注3} (溢出时间的 75%)	内部	0004H	(A)
	1	INTLVI	低电压检测 ^{注4}		0006H	
	2	INTP0	引脚输入边沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	5	INTP3			000EH	
	6	INTP4			0010H	
	7	INTP5			0012H	
	8	INTST3	UART3 发送完成	内部	0014H	(A)
	9	INTSR3	UART3 接收完成		0016H	
	10	INTSRE3	UART3 产生接收错误		0018H	
	11	INTDMA0	DMA0 发送完成		001AH	
	12	INTDMA1	DMA1 发送完成		001CH	
	13	INTST0 /INTCSI00	UART0 发送完成/ CSI00 通信完成		001EH	
	14	INTSR0 /INTCSI01	UART0 接收完成/ CSI01 通信完成		0020H	
	15	INTSRE0	CSI00/CSI01/UART0 发生接收错误		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	UART1 发送完成/ CSI10 通信完成/ IIC10 通信完成		0024H	
	17	INTSR1	UART1 接收完成		0026H	
	18	INTSRE1	UART1 发生接收错误		0028H	
	19	INTIIC0	IIC0 通讯完成		002AH	
	20	INTTM00	通道 0 计数或捕获完成		002CH	
	21	INTTM01	通道 1 计数或捕获完成		002EH	
	22	INTTM02	通道 2 计数或捕获完成		内部	
23	INTTM03	通道 3 计数或捕获完成	0032H			

- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 40 的优先级别最低。
 2. 基本配置类型 (A) ~ (C) 与图 16-1 中 (A) ~ (C) 相对应。
 3. 当可选字节 (000C0H) 第 7 位 (WDTINT) 被置 1 时。
 4. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被清零时。

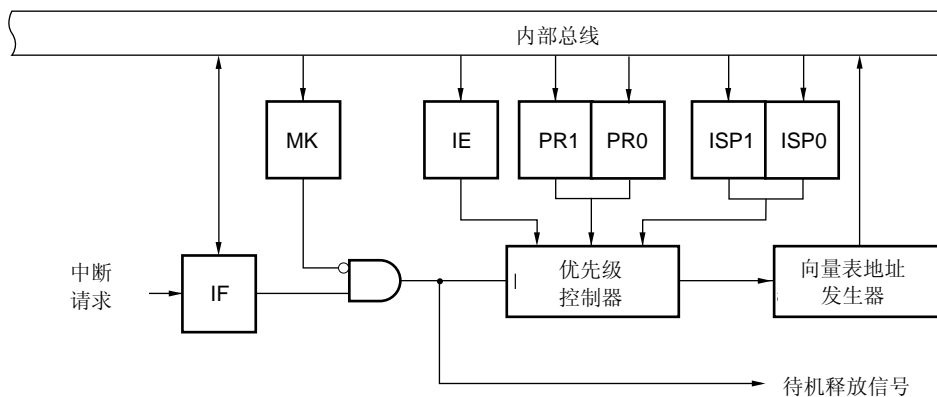
表 16-1. 中断源列表 (2/2)

中断类型	默认优先级 ^{注1}	中断源		内部/外部	向量表地址	基本配置类型 ^{注2}
		名称	触发器			
可屏蔽	24	INTAD	A/D 转换完成	内部	0034H	(A)
	25	INTRTC	实时计数器的固定周期信号/报警匹配检测		0036H	
	26	INTRTCI	实时计数器的间隔信号检测		0038H	
	27	INTKR	按键返回信号检测	外部	003AH	(B)
	28	INTST2 /INTCSI20 /INTIIC20	UART2 发送完成/ CSI20 通信完成/ IIC20 通信完成	内部	003CH	(A)
	29	INTSR2	UART2 接收完成		003EH	
	30	INTSRE2	产生 UART2 接收错误		0040H	
	31	INTTM04	通道 4 计数或捕获完成		0042H	
	32	INTTM05	通道 5 计数或捕获完成		0044H	
	33	INTTM06	通道 6 计数或捕获完成		0046H	
	34	INTTM07	通道 7 计数或捕获完成		0048H	
	35	INTP6	引脚输入边沿检测		外部	
	36	INTP7		004CH		
	37	INTP8		004EH		
	38	INTP9		0050H		
	39	INTP10		0052H		
	40	INTP11		0054H		
软件	-	BRK	执行 BRK 指令	-	007EH	(C)
复位	-	RESET	RESET 引脚输入	-	0000H	-
		POC	上电清零			
		LVI	低电压检测 ^{注3}			
		WDT	看门狗定时器溢出			
		TRAP	执行非法指令 ^{注4}			

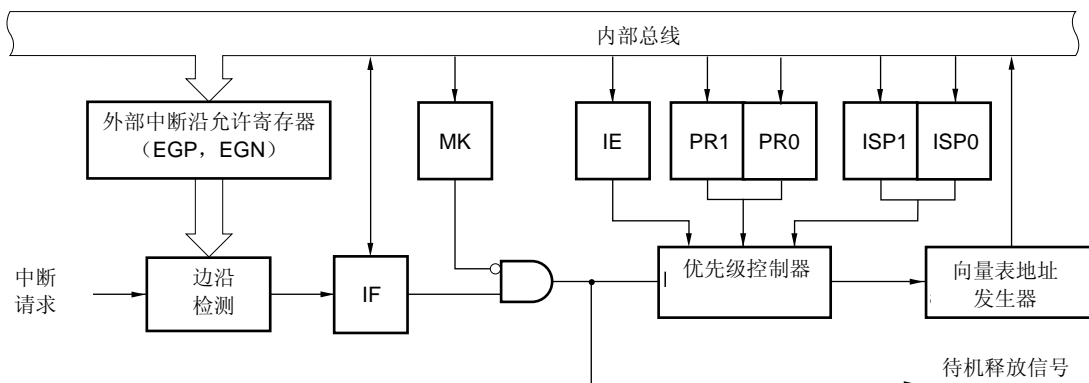
- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级最高，而优先级 40 的优先级最低。
 2. 基本配置类型 (A) ~ (C) 与图 16-1 中 (A) ~ (C) 相对应。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被置 1 时。
 4. 当执行 FFH 中的指令码时。
复位是由非法指令执行引起的，而不是由在线仿真器或者在片调试仿真器的仿真引起。

图 16-1. 中断功能的基本配置

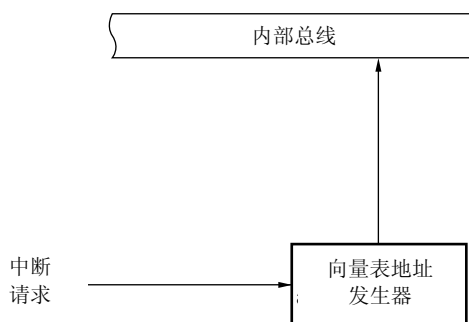
(A) 内部可屏蔽中断



(B) 外部可屏蔽中断



(C) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP0: 正在服务的中断优先级标志 0
- ISP1: 正在服务的中断优先级标志 1
- MK: 中断屏蔽标志
- PR0: 优先级规范标志 0
- PR1: 优先级规范标志 1

16.3 控制中断功能的寄存器

以下 6 种寄存器用于控制中断功能。

- 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- 优先级规范标志寄存器 (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- 外部中断上升沿允许寄存器 (EGP0, EGP1)
- 外部中断下降沿允许寄存器 (EGN0, EGN1)
- 程序状态字 (PSW)

表 16-2 列出了与中断请求源相对应的中断请求标志、中断屏蔽标志和优先级规范标志。

表 16-2. 与中断请求源相对应的标志 (1/2)

中断源	中断请求标志		中断屏蔽标志		优先级规范标志	
		寄存器		寄存器		寄存器
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 ^{注1}	STIF0 ^{注1}		STMK0 ^{注1}		STPR00, STPR10 ^{注1}	
INTCSI00 ^{注1}	CSIIF00 ^{注1}		CSIMK00 ^{注1}		CSIPR000, CSIPR100 ^{注1}	
INTSR0 ^{注2}	SRIF0 ^{注2}		SRMK0 ^{注2}		SRPR00, SRPR10 ^{注2}	
INTCSI01 ^{注2}	CSIIF01 ^{注2}		CSIMK01 ^{注2}		CSIPR001, CSIPR101 ^{注2}	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

- 注
1. 不要同时使用 UART0 和 CSI00，因为它们使用同样的中断请求源标志。如果产生 INTST0 和 INTCSI00 中断源中的某一个，IF1H 的第 5 位设置为 1。MK0H，PR00H，和 PR10H 的第 5 位支持这 3 个中断源。
 2. 不要同时使用 UART0 和 CSI01，因为它们使用同样的中断请求源标志。如果产生 INTSR0 和 INTCSI01 中断源中的某一个，IF0H 的第 6 位设置为 1。MK0H，PR00H，和 PR10H 的第 6 位支持这 3 个中断源。

表 16-2. 与中断请求源相对应的标志 (2/2)

中断源	中断请求标志		中断屏蔽标志		优先级规范标志	
		寄存器		寄存器		寄存器
INTST1 ^{注1}	STIF1 ^{注1}	IF1L	STMK1 ^{注1}	MK1L	STPR01, STPR11 ^{注1}	PR01L, PR11L
INTCSI10 ^{注1}	CSIIF10 ^{注1}		CSIMK10 ^{注1}		CSIPR010, CSIPR110 ^{注1}	
INTIIC10 ^{注1}	IICIF10 ^{注1}		IICMK10 ^{注1}		IICPR010, IICPR110 ^{注1}	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIIC0	IICIF0		IICMK0		IICPR00, IICPR10	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIF		RTCIMK		RTCIPR0, RTCIPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTST2 ^{注2}	STIF2 ^{注2}		STMK2 ^{注2}		STPR02, STPR12 ^{注2}	
INTCSI20 ^{注2}	CSIIF20 ^{注2}		CSIMK20 ^{注2}		CSIPR020, CSIPR120 ^{注2}	
INTIIC20 ^{注2}	IICIF20 ^{注2}		IICMK20 ^{注2}		IICPR020, IICPR120 ^{注2}	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTP10	PIF10		PMK10		PPR010, PPR110	
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H

- 注 1. 不要同时使用 UART1, CSI10, 和 IIC10, 因为它们使用同样的中断请求源标志。如果产生 INTST1, INTCSI10, 和 INTIIC10 中断源中的某一个, IF1L 的第 0 位设置为 1。MK1L, PR01L, 和 PR11L 的第 0 位支持这 3 个中断源。
2. 不要同时使用 UART2, CSI20, 和 IIC20, 因为它们使用同样的中断请求源标志。如果产生 INTST2, INTCSI20, 和 INTIIC20 中断源中的某一个, IF1L 的第 4 位设置为 1。MK1H, PR01 H, 和 PR11 H 的第 4 位支持这 3 个中断源。

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

当产生相关的中断请求或执行指令时，这些中断请求标志被置 1。当在响应中断请求时或产生复位信号时执行指令，这些标志被清零。

当响应中断时，中断请求标志自动清零，然后进入中断服务程序。

可由 1 位或 8 位存储器操作指令设置 IF0L、IF0H、IF1L、IF1H、IF2L 和 IF2H。当 IF0L 与 IF0H、IF1L 与 IF1H 和 IF2L 与 IF2H 组合起来形成 16 位寄存器 IF0、IF1 和 IF2 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器清零 (00H)。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟数量增加 2 个时钟。

图 16-2. 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) 的格式 (1/2)

地址: FFFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF

地址: FFFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	CSIIF01 SRIF0	CSIIF00 STIF0	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

地址: FFFE2H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICIF0	SREIF1	SRIF1	CSIIF10 IICIF10 STIF1

地址: FFFE3H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	SREIF2	SRIF2	CSIIF20 IICIF20 STIF2	KRIF	RTCIIF	RTCIF	ADIF

地址: FFFD0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	PIF6	TMIF07	TMIF06	TMIF05

图 16-2. 中断请求标志寄存器的格式 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

地址: FFFD1H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	0	0	0	0	PIF11

XXIFX	中断请求标志
0	没有产生中断请求信号
1	产生中断请求, 中断请求状态

注意事项 1. 确保要对 IF2H 的第 1 位到第 7 位清零 0。

2. 当退出待机模式后并要使用定时器、串行接口或 A/D 转换器时, 先将中断请求标志清零后再对这些部件进行操作。噪音可能会设置中断请求标志。

3. 当使用中断请求标志寄存器中的标志时, 使用一位存储器操作指令 (CLR1)。当用 C 语言描述时, 由于编译后的汇编指令必须是一位存储器操作指令 (CLR1), 所以应该使用一位操作指令, 如“IF0L.0 = 0;”或“_asm (“clr1 IF0L, 0”);”。

如果一条用 C 语言描述的 8 位存储器操作指令如“IF0L &= 0xfe;”, 则编译后将被转换为三条汇编指令:

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

这种情况下, 当处于“mov a, IF0L”和“mov IF0L, a”之间时, 即使同一中断请求标志寄存器 (IF0L) 的其他请求标志被设置为 1, 则该请求标志也将被“mov IF0L, a”清零。因此在 C 语言中使用 8 位存储器操作指令时必须小心。

(2) 中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断服务。

可由 1 位或 8 位存储器操作指令设置 MK0L、MK0H、MK1L、MK1H、MK2L、和 MK2H。当 MK0L 与 MK0H, MK1L 与 MK1H, 和 MK2L 与 MK2H 组合起来形成 16 位寄存器 MK0、MK1 和 MK2 时, 可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器的内容设置为 FFH。

备注 如果执行写入此寄存器数据的指令, 指令执行的时钟数量增加 2 个时钟。

图 16-3. 中断屏蔽标志寄存器的格式 (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

地址: FFFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

地址: FFFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	CSIMK01 SRMK0	CSIMK00 STMK0	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

地址: FFFE6H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICMK0	SREMK1	SRMK11	CSIMK10 IICMK10 STMK1

地址: FFFE7H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	SREMK2	SRMK2	CSIMK20 IICMK20 STMK2	KRMK	RTCIMK	RTCMK	ADMK

地址: FFFD4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	PMK6	TMMK07	TMMK06	TMMK05

地址: FFFD5H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
MK2H	1	1	1	1	1	1	1	PMK11

XXMKX	中断服务控制						
0	允许中断服务						
1	禁止中断服务						

注意事项 一定要把 MK2H 的第 1 位到第 7 位 设置为 1。

(3) 优先级规范标志寄存器 (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

这些优先级规范标志寄存器用于设置相关的可屏蔽中断优先级次序。

优先级可通过组合使用 PR0xy 和 PR1xy 寄存器来设置 (xy = 0L, 0H, 1L, 1H, 2L 或 2H)。可由 1 位或 8 位存储器操作指令设置 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、和 PR12H。当 PR00L 与 PR00H, PR01L 与 PR01H, PR02L 与 PR02H, PR10L 与 PR10H, PR11L 与 PR11H, 和 PR12L 与 PR12H 组合起来形成 16 位寄存器 PR00, PR01, PR02, PR10, PR11 和 PR12 时, 可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器内容设置为 FFH。

备注 如果执行写入此寄存器数据的指令, 指令执行的时钟数量增加 2 个时钟。

图 16-4. 优先级规范标志寄存器的格式

(PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

地址: FFFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

地址: FFECH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

地址: FFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	CSIPR001 SRPR00	CSIPR000 STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

地址: FFFEDH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	CSIPR101 SRPR10	CSIPR100 STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

地址: FFFEAH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICPR00	SREPR01	SRPR01	CSIPR010 IICPR010 STPR01

地址: FFFEEH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICPR10	SREPR11	SRPR11	CSIPR110 IICPR110 STPR11

图 16-4. 优先级规范标志寄存器的格式

(PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

地址: FFFEBH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	SREPR02	SRPR02	CSIPR020 IICPR020 STPR02	KRPR0	RTCIPR0	RT CPR0	ADPR0

地址: FFFEFH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	SREPR12	SRPR12	CSIPR120 IICPR120 STPR12	KRPR1	RTCIPR1	RT CPR1	ADPR1

地址: FFFD8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

地址: FFFDCH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

地址: FFFD9H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
PR02H	1	1	1	1	1	1	1	PPR011

地址: FFFDDH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	<0>
PR12H	1	1	1	1	1	1	1	PPR111

XXPR1X	XXPR0X	优先级选择
0	0	指定优先级 0 (高优先级)
0	1	指定优先级 1
1	0	指定优先级 2
1	1	指定优先级 3 (低优先级)

注意事项 一定要把 PR02H 和 PR12H 的第 1 位和~第 7 位设置为 1。

(4) 外部中断上升沿允许寄存器 (EGP0, EGP1)

这两个寄存器用于指定 INTP0~INTP11 的有效沿。

可由 1 位或 8 位存储器操作指令设置 EGP0, EGP1, EGN0 和 EGN1。

复位信号的产生将这些寄存器清零 (00H)。

图 16-5.外部中断上升沿允许寄存器 (EGP0, EGP1) 和外部中断下降沿允许寄存器 (EGN0, EGN1) 的格式

地址: FFF38H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FFF39H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

地址: FFF3AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

地址: FFF3BH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn 引脚有效沿的选择 (n = 0 ~ 11)
0	0	禁止脉冲沿检测
0	1	下降沿
1	0	上升沿
1	1	兼有上升沿和下降沿

表 16-3 给出了与 EGPn 和 EGNn 对应的端口。

表 16-3. EGPn 和 EGNn 相关的端口

检测允许寄存器		脉冲沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P46	INTP1
EGP2	EGN2	P47	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7
EGP8	EGN8	P74	INTP8
EGP9	EGN9	P75	INTP9
EGP10	EGN10	P76	INTP10
EGP11	EGN11	P77	INTP11

注意事项 在从外部中断功能切换到端口功能时可能会检测到脉冲沿，因此将 EGPn 和 EGNn 清零可以选择端口模式。

备注 n = 0 ~ 11

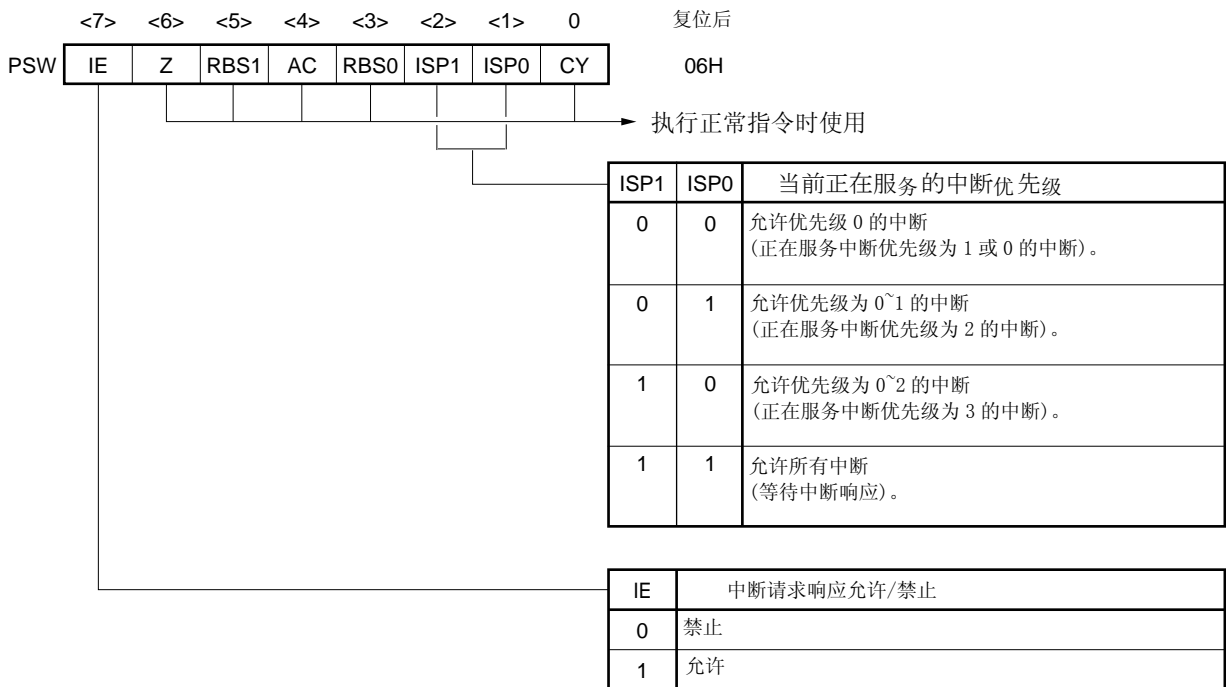
(5) 程序状态字 (PSW)

程序状态字 (PSW) 是用于保存指令执行结果和中断请求当前状态的寄存器。在 PSW 中包含 IE 标志 (设置允许/禁止可屏蔽中断) 和 ISP0 与 ISP1 标志 (控制中断嵌套)。

除了 8 位读/写操作指令外, 还可使用位操作指令和专用指令 (EI 和 DI) 对该寄存器进行操作。在响应向量中断请求时, 如果执行 BRK 指令, 则将 PSW 的内容自动保存到堆栈中, 并且 IE 标志复位为 0。如果响应可屏蔽中断请求, 则将被响应中断的优先级规范标志的内容转移到 ISP0 和 ISP1 标志中。执行 PUSH PSW 指令将 PSW 的内容保存到堆栈中。而执行 RETI、RETB 和 POP PSW 指令可将这些内容从堆栈中恢复。

RESET 输入将 PSW 设置为 06H。

图 16-6. 程序状态字的格式



16.4 中断服务操作

16.4.1 可屏蔽的中断响应

当中断请求标志被置 1，且与该中断请求对应的屏蔽标志（MK）被清零时，这个可屏蔽中断请求可以被响应。如果处于中断允许状态（IE 标志置 1），可以响应这个向量中断请求。但在一个较高优先级中断请求服务期间，不能响应低优先级的中断请求。

从产生一个可屏蔽中断请求到中断服务执行所需要的时间如表 16-4 所示。

中断请求响应时序可参见图 16-8 和 16-9。

表 16-4. 从可屏蔽中断产生到执行中断服务所需要的时间

	最短时间	最长时间 ^注
服务时间	9 个时钟	14 个时钟

注 如果是在 RET 指令执行之前产生一个中断请求，则等待时间会更长。

备注 1 个时钟：1/f_{CLK}（f_{CLK}：CPU 时钟）

如果同时产生两个或两个以上的可屏蔽中断请求，则首先响应优先级规范标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别，则首先响应具有最高默认优先级的中断请求。

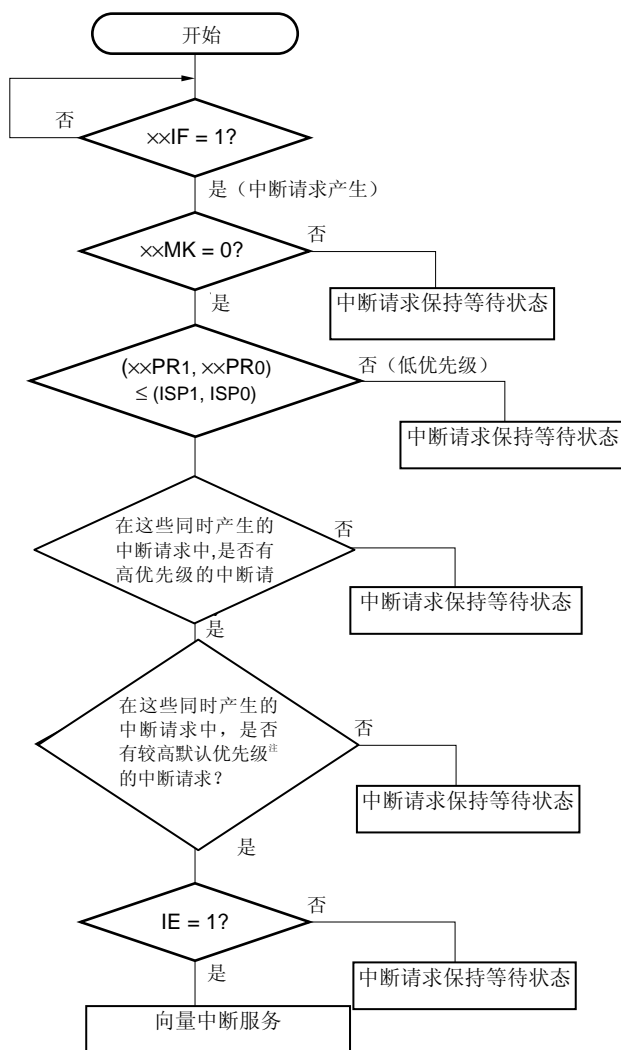
当允许响应中断请求时，处于等待状态的中断请求被响应。

图 16-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求，则将 PSW、PC 的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将与被响应中断的优先级规范标志的内容传送到 ISP1 和 ISP0 中。将中断向量表中的中断请求数据传送到 PC 中，并根据该数据进行跳转。

执行 RETI 指令可以从中断返回。

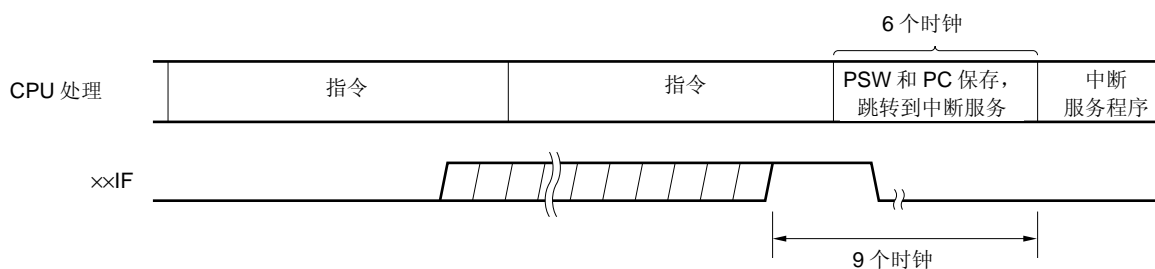
图 16-7. 中断请求响应处理



- xxIF: 中断请求标志
- xxMK: 中断屏蔽标志
- xxPR0: 优先级规范标志 0
- xxPR1: 优先级规范标志 1
- IE: 控制可屏蔽中断请求响应的标志 (1 = 允许、0 = 禁止)
- ISP0, ISP1: 指示正在服务的中断优先级别的标志 (参见图 16-6)

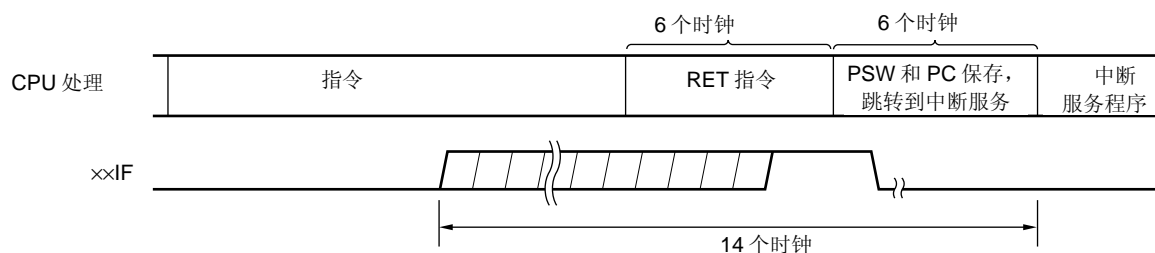
注 关于默认优先级, 参考表 16-1 中断源列表

图 16-8. 中断请求响应时序（最短时间）



备注 1 个时钟: $1/f_{CPU}$ (f_{CPU} : CPU 时钟)

图 16-9. 中断请求响应时序（最长时间）



备注 1 个时钟: $1/f_{CPU}$ (f_{CPU} : CPU 时钟)

16.4.2 软件中断请求响应

执行 BRK 指令就可以响应软件中断。软件中断不能被禁止。

如果响应了一个软件中断请求，则将程序状态字（PSW）和程序计数器（PC）的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将向量表中对应位置（0007EH，0007FH）的内容传送到 PC 中，然后进行跳转。

执行 RETB 指令可以从软件中断返回。

注意事项 从软件中断返回时不要使用 RETI 指令。

16.4.3 多重中断嵌套

在执行一个中断服务程序时，又响应了其他中断，这时就产生了多重中断嵌套。

除非选择允许中断请求响应状态（ $IE = 1$ ），否则不会产生多重中断嵌套。在响应一个中断请求时，禁止响应其它中断请求（ $IE = 0$ ）。因此，如果要允许多重中断嵌套，必须在中断服务期间执行 EI 指令，将 IE 标志置 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定能够成功进行多重中断嵌套，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于多重中断嵌套。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生多重中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于中断处于禁止状态或中断请求的优先级别较低，这些不被允许的中断请求处于挂起状态。若当前的中断服务已结束，则在执行至少一条主程序指令后才可以响应处于挂起状态的中断请求。

表 16-5 显示了允许多重中断嵌套的中断请求之间的关系，图 16-10 为中断嵌套示例。

表 16-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

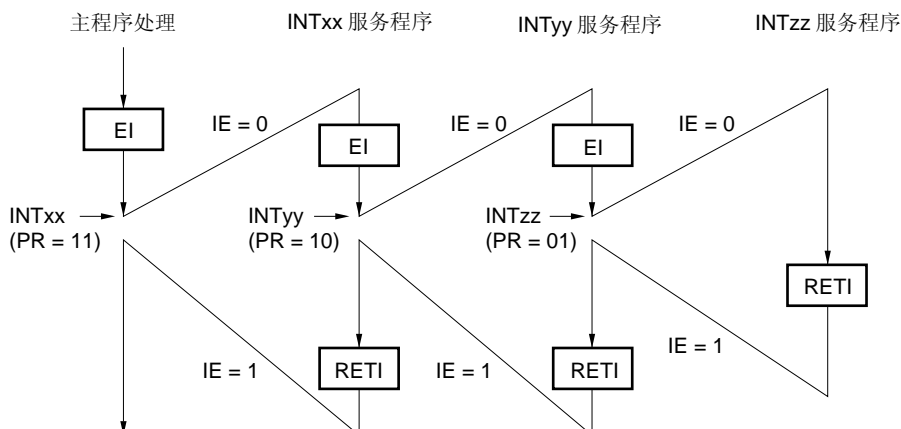
中断嵌套		可屏蔽中断请求								软件中断请求
		优先级 0 (PR = 00)		优先级 1 (PR = 01)		优先级 2 (PR = 10)		优先级 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
正在服务的中断										
可屏蔽的中断	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1 ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
软件中断		○	×	○	×	○	×	○	×	○

备注

- ：允许中断嵌套
- ×：禁止中断嵌套
- ISP0, ISP1, 和 IE 为 PSW 中的标志位。
 ISP1 = 0, ISP0 = 0: 正在服务中断优先级 1 或 0。
 ISP1 = 0, ISP0 = 1: 正在服务中断优先级 2。
 ISP1 = 1, ISP0 = 0: 正在服务中断优先级 3。
 ISP1 = 1, ISP0 = 1: 等待中断响应。
 IE = 0: 禁止响应中断请求。
 IE = 1: 允许响应中断请求。
- PR 为 PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, 和 PR12H 中的标志位。
 PR = 00: 用 $\times\times PR1\times = 0, \times\times PR0\times = 0$ 指定优先级 0 (较高优先级)
 PR = 01: 用 $\times\times PR1\times = 0, \times\times PR0\times = 1$ 指定优先级 1
 PR = 10: 用 $\times\times PR1\times = 1, \times\times PR0\times = 0$ 指定优先级 2
 PR = 11: 用 $\times\times PR1\times = 1, \times\times PR0\times = 1$ 指定优先级 3 (较低优先级)

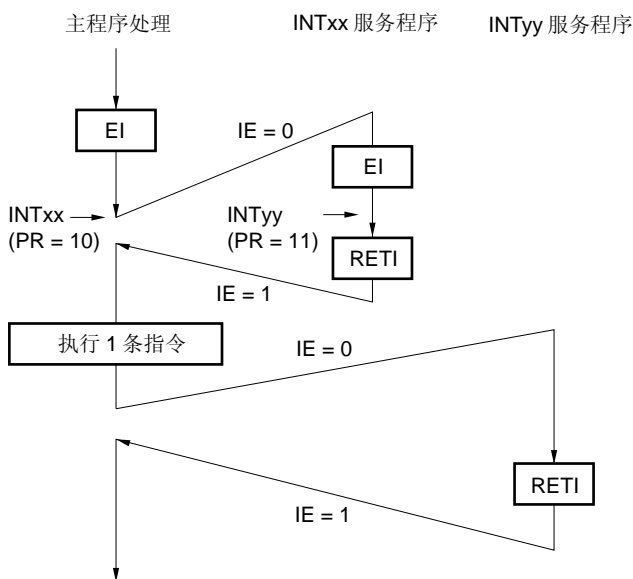
图 16-10. 中断嵌套示例 (1/2)

例 1. 产生两次中断嵌套



在进行 INTxx 中断服务期间，响应了两个中断请求 INTyy 和 INTzz，这时就产生了中断嵌套。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

例 2. 由于优先级控制没有产生中断嵌套

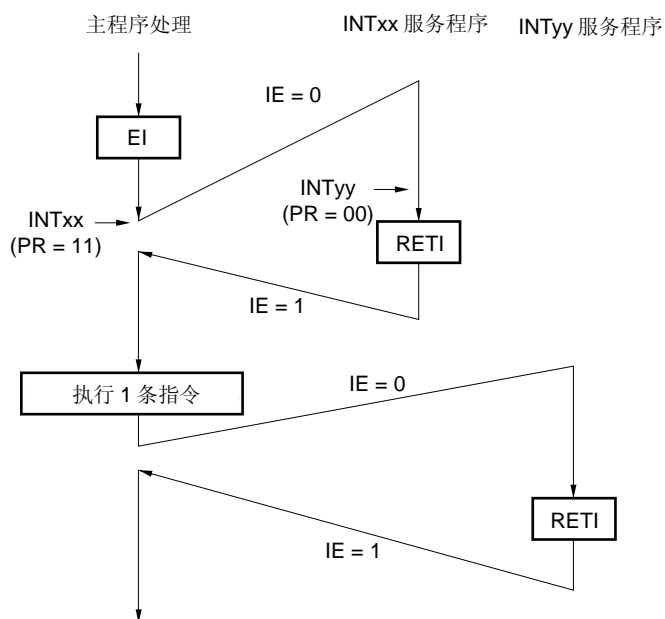


在 INTxx 中断服务期间未响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。中断请求 INTyy 处于挂起等待状态，在执行一条主程序指令后响应该中断请求。

- PR = 00: 用 $\times\times PR1\times = 0$ 、 $\times\times PR0\times = 0$ 指定优先级 0 (较高优先级)
- PR = 01: 用 $\times\times PR1\times = 0$ 、 $\times\times PR0\times = 1$ 指定优先级 1
- PR = 10: 用 $\times\times PR1\times = 1$ 、 $\times\times PR0\times = 0$ 指定优先级 2
- PR = 11: 用 $\times\times PR1\times = 1$ 、 $\times\times PR0\times = 1$ 指定优先级 3 (较低优先级)
- IE = 0: 禁止响应中断请求。
- IE = 1: 允许响应中断请求。

图 16-10. 中断嵌套示例 (2/2)

例 3. 由于不允许中断没有产生中断嵌套



在 INT_{xx} 中断服务期间不允许中断（未执行 EI 指令），因此不能响应中断请求 INT_{yy} ，这样也不会产生中断嵌套。中断请求 INT_{yy} 处于挂起等待状态，在执行一条主程序指令后响应该中断请求。

- PR = 00: 用 $\times\times PR1\times = 0$ 、 $\times\times PR0\times = 0$ 指定优先级 0（较高优先级）
- PR = 01: 用 $\times\times PR1\times = 0$ 、 $\times\times PR0\times = 1$ 指定优先级 1
- PR = 10: 用 $\times\times PR1\times = 1$ 、 $\times\times PR0\times = 0$ 指定优先级 2
- PR = 11: 用 $\times\times PR1\times = 1$ 、 $\times\times PR0\times = 1$ 指定优先级 3（较低优先级）
- IE = 0: 禁止响应中断请求。
- IE = 1: 允许响应中断请求。

16.4.4 保持中断请求

在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令（中断请求保持指令）。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- 操作 IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L 和 PR12H 寄存器的指令。

注意事项 BRK 指令不属于上述列出的中断请求保持指令。但通过执行 BRK 指令激活软件中断，会将 IE 标志清零。因此，即使在执行 BRK 指令期间产生可屏蔽中断请求，该中断请求也不会被响应。

图 16-11 显示了处于等待状态的中断请求时序。

图 16-11. 中断请求保持



- 备注**
1. 指令 N: 中断请求保持指令
 2. 指令 M: 中断请求保持指令之外的指令
 3. xxPR（优先级别）的值不会影响xxIF（中断请求）的操作。

第十七章 按键中断功能

17.1 按键中断的功能

通过设置按键返回模式寄存器（KRM），按键中断输入引脚（KR0~KR7）输入一个下降沿时就会产生按键中断（INTKR）。

表 17-1. 按键中断检测引脚的分配

标志	描述
KRM0	以 1 位单元控制 KR0 信号。
KRM1	以 1 位单元控制 KR1 信号。
KRM2	以 1 位单元控制 KR2 信号。
KRM3	以 1 位单元控制 KR3 信号。
KRM4	以 1 位单元控制 KR4 信号。
KRM5	以 1 位单元控制 KR5 信号。
KRM6	以 1 位单元控制 KR6 信号。
KRM7	以 1 位单元控制 KR7 信号。

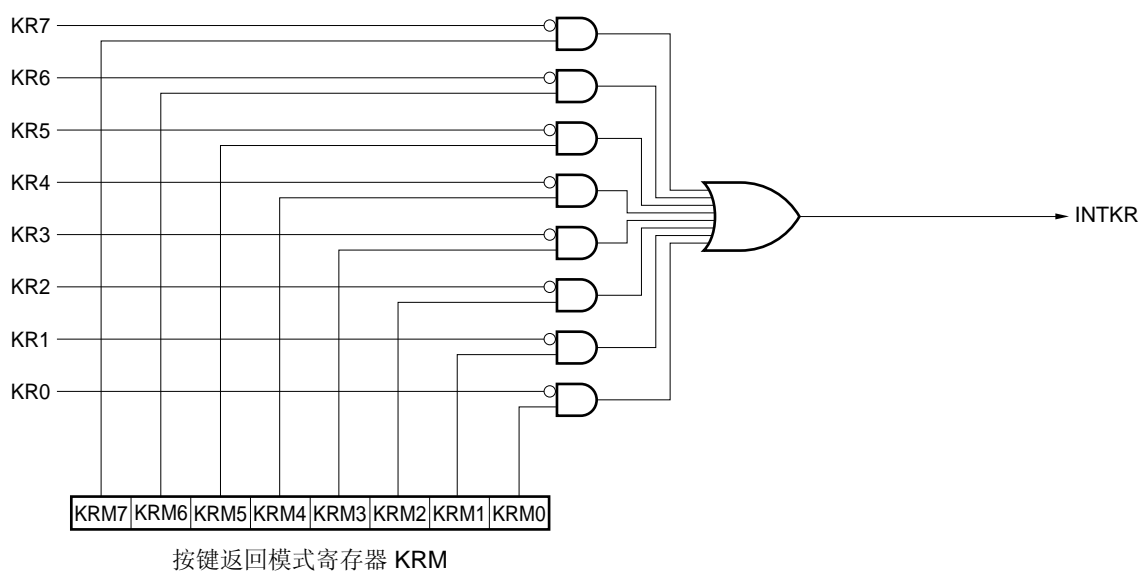
17.2 按键中断的配置

按键中断包括以下硬件。

表 17-2 按键中断的配置

项目	配置
控制寄存器	按键返回模式寄存器（KRM）

图 17-1. 按键中断的框图



17.3 控制按键中断的寄存器

(1) 按键返回模式寄存器 (KRM)

该寄存器使用 KR0 ~ KR7 信号分别控制 KRM0 ~ KRM7 位。

KRM 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将 KRM 清零 (00H)。

图 17-2. 按键返回模式寄存器 (KRM) 的格式

地址: FFF37H 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	按键中断模式控制
0	不检测按键中断信号
1	检测按键中断信号

- 注意事项**
1. 如果 KRM0 ~ KRM7 中任何一位被设置为 1, 则应该将上拉电阻寄存器 7 (PU7) 的第 0 位 ~ 第 7 位 (PU70 ~ PU77) 的相应位设置为 1。
 2. 如果 KRM 发生变化, 则中断请求标志可能被设置。因此先禁止中断, 然后再修改 KRM。先将中断请求标志清零, 再允许中断。
 3. 在按键中断模式中没有被使用的位可用作通用端口。

备注 n = 0 ~ 7

第十八章 待机功能

18.1 待机功能及配置

18.1.1 待机功能

待机功能用于降低系统的工作电流，有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令，设置 HALT 模式。在 HALT 模式中，CPU 操作时钟停止。如果在设置 HALT 模式前，高速系统时钟振荡器、内部高速振荡器、内部低速振荡器或副系统时钟振荡器正在使用，则设置后每种时钟继续振荡。在此模式中，工作电流不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和频繁进行间断的操作非常有效。

(2) STOP 模式

通过执行 STOP 指令，设置 STOP 模式。在 STOP 模式中，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作停止，这样 CPU 的工作电流将会大幅下降。

该模式可以通过中断请求释放，这样可以允许执行间歇性的操作。但是在选择 X1 时钟时，在释放 STOP 模式之后需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

在这两种模式中，寄存器、标志和数据存储器的所有内容将会保持进入待机模式前的内容。I/O 端口输出锁存器和输出缓冲器的状态也将被保持。

- 注意事项**
1. 仅当 CPU 使用主系统时钟时，才能使用 STOP 模式。当 CPU 使用副系统时钟时，不能设置 STOP 模式。当 CPU 使用主系统时钟或副系统时钟时，均可以使用 HALT 模式。
 2. 当切换到 STOP 模式时，在执行 STOP 指令前必须停止那些使用主系统时钟进行操作的外围硬件。
 3. 当使用待机功能时，建议采用以下步骤降低 A/D 转换器的操作电流：首先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零以停止 A/D 转换操作，然后执行 STOP 指令。
 4. 可通过选项字节选择在 HALT 或 STOP 模式下是否停止内部低速振荡器。详细信息，参见第二十三章 选项字节

18.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 振荡稳定时间计数器状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)

备注 有关这些寄存器启动、停止以及时钟选择的情况，可参见第五章 时钟发生器。

(1) 振荡稳定时间计数器的状态寄存器 (OSTC)

该寄存器用于指示 X1 时钟振荡稳定时间计数器的计数状态。

在下列情况下，可以检测 X1 时钟振荡稳定时间。

- 在内部高速振荡时钟或副系统时钟用作 CPU 时钟期间，X1 时钟开始振荡。
- 当 X1 时钟振荡的情况下，内部高速振荡时钟用作 CPU 时钟期间，进入 STOP 模式，然后释放。

OSTC 可以通过 1 位或 8 位存储器操作指令读取。

复位释放 (通过 RESET 输入、POC、LVI、WDT 和执行非法指令)、STOP 指令以及 MSTOP (CSC 寄存器的第 7 位) = 1 都可以将 OSTC 清零 (00H)。

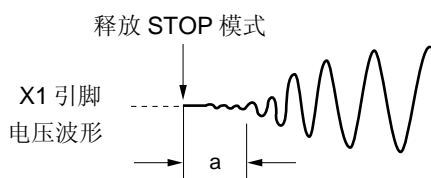
图 18-1. 振荡稳定时间计数器的状态寄存器 (OSTC) 的格式

地址: FFFA2H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	振荡稳定时间的状态		
								fx = 10 MHz	fx = 20 MHz	
0	0	0	0	0	0	0	0	2 ⁸ /fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2 ⁸ /fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2 ⁹ /fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2 ¹⁰ /fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2 ¹¹ /fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2 ¹³ /fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2 ¹⁵ /fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2 ¹⁷ /fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2 ¹⁸ /fx min.	26.21 ms min.	13.11 ms min.

- 注意事项
1. 在经过上述振荡稳定时间后，从 MOST8 起各位被设置为 1，并保持为 1。
 2. 振荡稳定时间计数器的计数达到 OSTC 设置的振荡稳定时间。如果在 CPU 使用内部高速振荡时钟时，进入 STOP 模式后，又释放 STOP 模式，则可以按以下方式设置振荡稳定时间。
 - 预期的 OSTC 振荡稳定时间 ≤ OSTC 设置的振荡稳定时间
 因此需要注意，在释放 STOP 模式后，只有经过 OSTC 设置的振荡稳定时间之后，才能在 OSTC 设置对应状态。
 3. X1 时钟振荡稳定等待时间不包括从释放 STOP 模式到时钟振荡开始之前这段时间。(下图“a”表示的部分)。



备注 fx: X1 时钟振荡频率。

(2) 振荡稳定时间选择寄存器 (OSTS)

该寄存器用于选择释放 STOP 模式后 X1 时钟振荡稳定等待时间。

在释放 STOP 模式后，且 CPU 使用 X1 时钟时，需要等待由 OSTS 设置的振荡稳定时间。

当释放 STOP 模式后，且 CPU 使用内部高速振荡时钟时，可以通过 OSTC 确认是否已经经过预期的振荡稳定时间。可以使用 OSTC 来检测振荡是否稳定，直到达到设置的时钟振荡稳定时间。

OSTS 可以通过 8 位存储器操作指令设置。

复位信号的产生将 OSTS 设置为 07H。

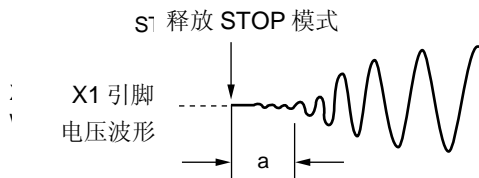
图 18-2. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFFA3H 复位后: 07H R/W

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

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^8/f_x$	25.6 μs	禁止设置
0	0	1	$2^9/f_x$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_x$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_x$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_x$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.21 ms	13.11 ms

- 注意事项
1. 当 X1 时钟用作 CPU 时钟使用时，若要设置 STOP 模式，则必须在执行 STOP 指令之前设置 OSTS。
 2. 禁止设置 振荡稳定时间 小于 20 s。
 3. 要改变 OSTS 寄存器的设置，必须确保 OSTC 寄存器的计数操作已经完成。
 4. 在 X1 时钟振荡稳定时间期间不要改变 OSTS 寄存器的值。
 5. 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。如果在 CPU 使用内部高速振荡时钟时，进入 STOP 模式后，又释放 STOP 模式，则可以按以下方式设置振荡稳定时间。
 - 预期的 OSTC 振荡稳定时间 \leq OSTS 设置的振荡稳定时间
 因此需要注意，在释放 STOP 模式后，只有经过 OSTS 设置的振荡稳定时间之后，才能在 OSTC 设置对应状态。
 6. X1 时钟振荡稳定等待时间不包括从释放 STOP 模式到时钟振荡开始之前这段时间。（下图“a”表示的部分）。



备注 f_x : X1 时钟振荡频率。

18.2 待机功能的操作

18.2.1 HALT模式

(1) HALT 模式

通过执行 HALT 指令，设置 HALT 模式。无论设置之前 CPU 使用的是高速系统时钟、内部高速振荡时钟还是副系统时钟，都可以设置 HALT 模式。

HALT 模式中的操作状态如下所示。

表 18-1. HALT 模式中的操作状态 (1/2)

HALT 模式设置		当 CPU 使用主系统时钟并执行 HALT 指令时		
项目		当 CPU 使用内部高速振荡时钟 (f _H) 时	当 CPU 使用 X1 时钟 (f _X) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
系统时钟		停止 CPU 的时钟		
主系统时钟	f _H	操作继续 (不能停止)	保持设置 HALT 模式前的状态	
	f _X	保持设置 HALT 模式前的状态	操作继续 (不能停止)	不能操作
	f _{EX}		不能操作	操作继续 (不能停止)
副系统时钟	f _{XT}	保持设置 HALT 模式前的状态		
f _{IL}		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置		
		<ul style="list-style-type: none"> • WTON = 0: 停止 • WTON = 1 和 WDSTBYON = 1: 振荡 • WTON = 1 和 WDSTBYON = 0: 停止 		
CPU		操作停止		
Flash 存储器		在低电流消耗模式下可操作		
RAM		操作停止。但是, 当电压比 POC 检测电压高时, 保持 HALT 模式设置之前的状态。		
端口 (锁存器)		保持设置 HALT 模式前的状态		
定时器阵列单元 (TAU)		可操作的		
实时计数器 (RTC)				
看门狗定时器		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置		
		<ul style="list-style-type: none"> • WTON = 0: 停止 • WTON = 1 和 WDSTBYON = 1: 操作 • WTON = 1 和 WDSTBYON = 0: 停止 		
时钟输出/蜂鸣器输出		可以操作		
A/D 转换器				
D/A 转换器				
串行阵列单元 (SAU)				
串行接口 (IIC0)				
乘法器				
DMA 控制器		可以操作		
上电清零 (POC) 功能				
低电压检测功能				
外部中断				

备注

f_H: 内部高速振荡时钟

f_X: X1 时钟

f_{EX}: 外部主系统时钟

f_{XT}: XT1 时钟

f_{IL}: 内部低速振荡时钟

表 18-1. HALT 模式中的操作状态 (2/2)

HALT 模式设置		当 CPU 使用副系统时钟时, 执行 HALT 指令
项目		当 CPU 使用 XT1 时钟 (fxT) 时
系统时钟		停止 CPU 的时钟
主系统时钟	f _H	保持设置 HALT 模式前的状态
	f _x	
	f _{EX}	操作或由外部时钟输入停止
副系统时钟	f _{xT}	操作继续 (不能停止)
f _L		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置 <ul style="list-style-type: none"> • WTON = 0: 停止 • WTON = 1 和 WDSTBYON = 1: 振荡 • WTON = 1 和 WDSTBYON = 0: 停止
CPU		操作停止
Flash 存储器		在低电流消耗模式下可操作
RAM		操作停止。但是, 当电压比 POC 检测电压高时, 保持 HALT 模式设置之前的状态。
端口 (锁存器)		保持设置 HALT 模式前的状态
外部总线接口		操作停止
定时器阵列单元 (TAU)		可操作
实时计数器 (RTC)		
看门狗定时器		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置 <ul style="list-style-type: none"> • WTON = 0: 停止 • WTON = 1 和 WDSTBYON = 1: 操作 • WTON = 1 和 WDSTBYON = 0: 停止
时钟输出/蜂鸣器输出		可操作
A/D 转换器		
D/A 转换器		
串行阵列单元 (SAU)		
串行接口 (IIC0)		
乘法器		
DMA 控制器		可操作
上电清零功能		
低电压检测功能		
外部中断		

备注

- f_H: 内部高速振荡时钟
- f_x: X1 时钟
- f_{EX}: 外部主系统时钟
- f_{xT}: XT1 时钟
- f_L: 内部低速振荡时钟

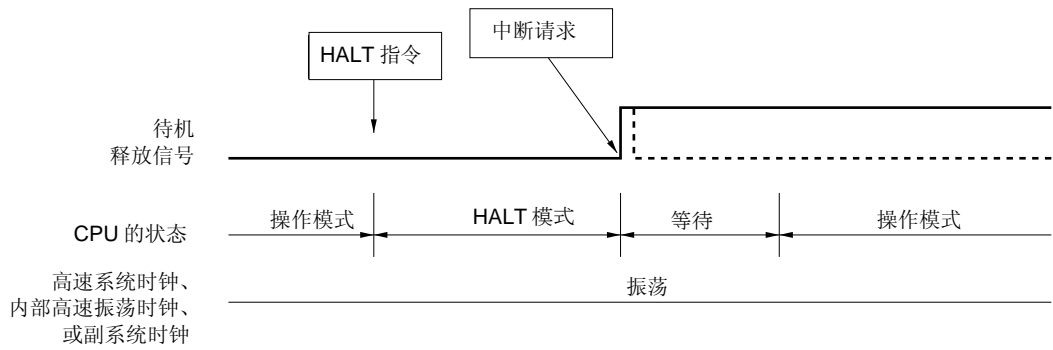
(2) 释放 HALT 模式

以下两种方式可以释放 HALT 模式。

(a) 由未被屏蔽的中断请求释放

当产生一个未被屏蔽的中断时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 18-3. 通过产生中断请求释放 HALT 模式



<R>

注 等待时间如下：

- 当进行向量中断服务时： 10 ~12 个时钟
- 未进行向量中断服务时： 5 ~6 个时钟

备注 虚线表示释放待机模式的中断请求被响应的情况。

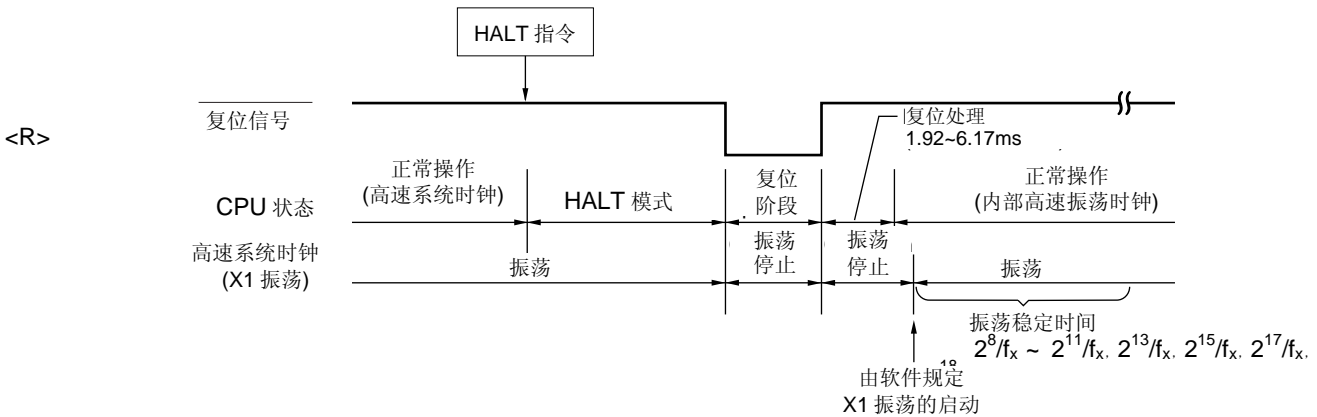
(b) 通过复位信号释放

当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作时，跳转到复位向量地址后，程序开始执行。

<R>

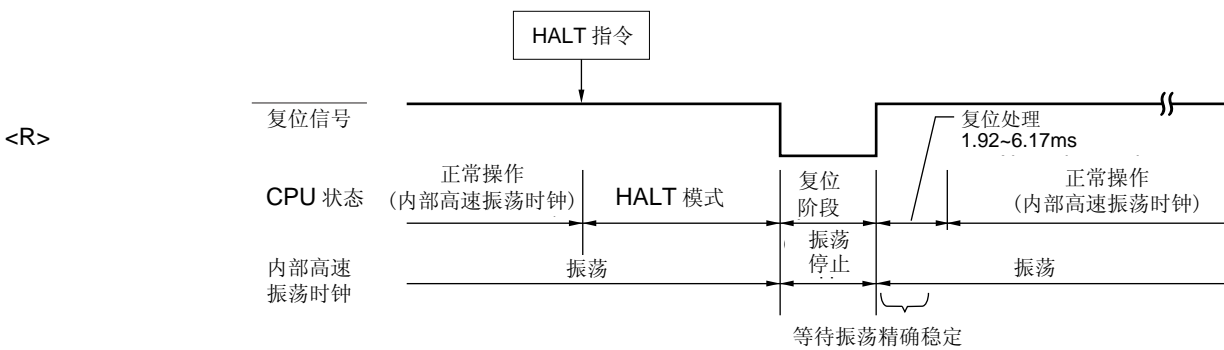
图 18-4. 通过复位释放 HALT 模式

(1) CPU 使用高速系统时钟



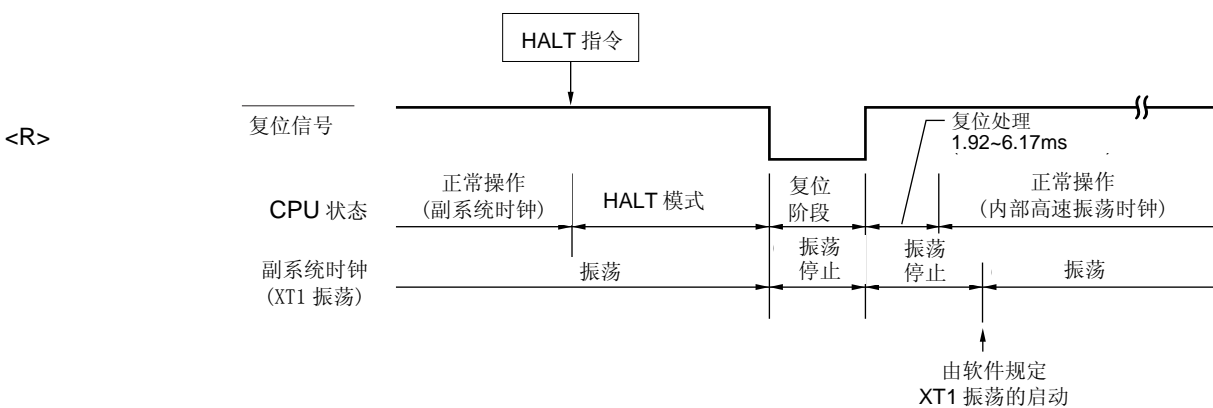
<R>

(2) CPU 使用内部高速振荡时钟



<R>

(3) CPU 使用副系统时钟



<R>

备注 fx: X1 时钟振荡频率

18.2.2 STOP模式

(1) STOP 模式设置及操作状态

执行 STOP 指令，设置 STOP 模式。只有在设置 STOP 模式前 CPU 使用主系统时钟，才可以设置 STOP 模式。

注意事项 由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志被设置且中断屏蔽标志被清零时，则立即释放待机模式。因此，在执行 STOP 指令后，立即将 STOP 模式复位到 HALT 模式，并且在经过 OSTS 设置的等待时间后，系统返回操作模式。

STOP 模式中的操作状态如下所示。

表 18-2. STOP 模式中的操作状态

STOP 模式设置		当 CPU 使用主系统时钟并执行 STOP 指令时		
		当 CPU 使用内部高速振荡时钟 (f _H) 时	当 CPU 使用 X1 时钟 (f _x) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
项目				
系统时钟		停止 CPU 的时钟		
主系统时钟	f _H	停止		
	f _x			
	f _{EX}			
副系统时钟	f _{XT}	保持设置 STOP 模式前的状态		
f _L		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置 ● WTON = 0: 停止 ● WTON = 1 和 WDSTBYON = 1: 振荡 ● WTON = 1 和 WDSTBYON = 0: 停止		
CPU		操作停止		
Flash 存储器		操作停止		
RAM		操作停止。但是, 当电压比 POC 检测电压高时, 保持设置 STOP 模式前的状态。		
端口 (锁存器)		保持设置 STOP 模式前的状态		
定时器阵列单元 (TAU)		操作停止		
实时计数器 (RTC)		保持设置 STOP 模式前的状态		
看门狗定时器		由选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 设置 ● WTON = 0: 停止 ● WTON = 1 和 WDSTBYON = 1: 操作 ● WTON = 1 和 WDSTBYON = 0: 停止		
时钟输出/蜂鸣器输出		只有当副系统时钟作为计数时钟时才可以操作		
A/D 转换器		操作停止		
D/A 转换器		操作停止 (引脚处于高阻状态)		
串行阵列单元 (SAU)		操作停止		
串行接口 (IIC0)				
乘法器				
DMA 控制器				
上电清零功能				
低电压检测功能		可以操作		
外部中断				

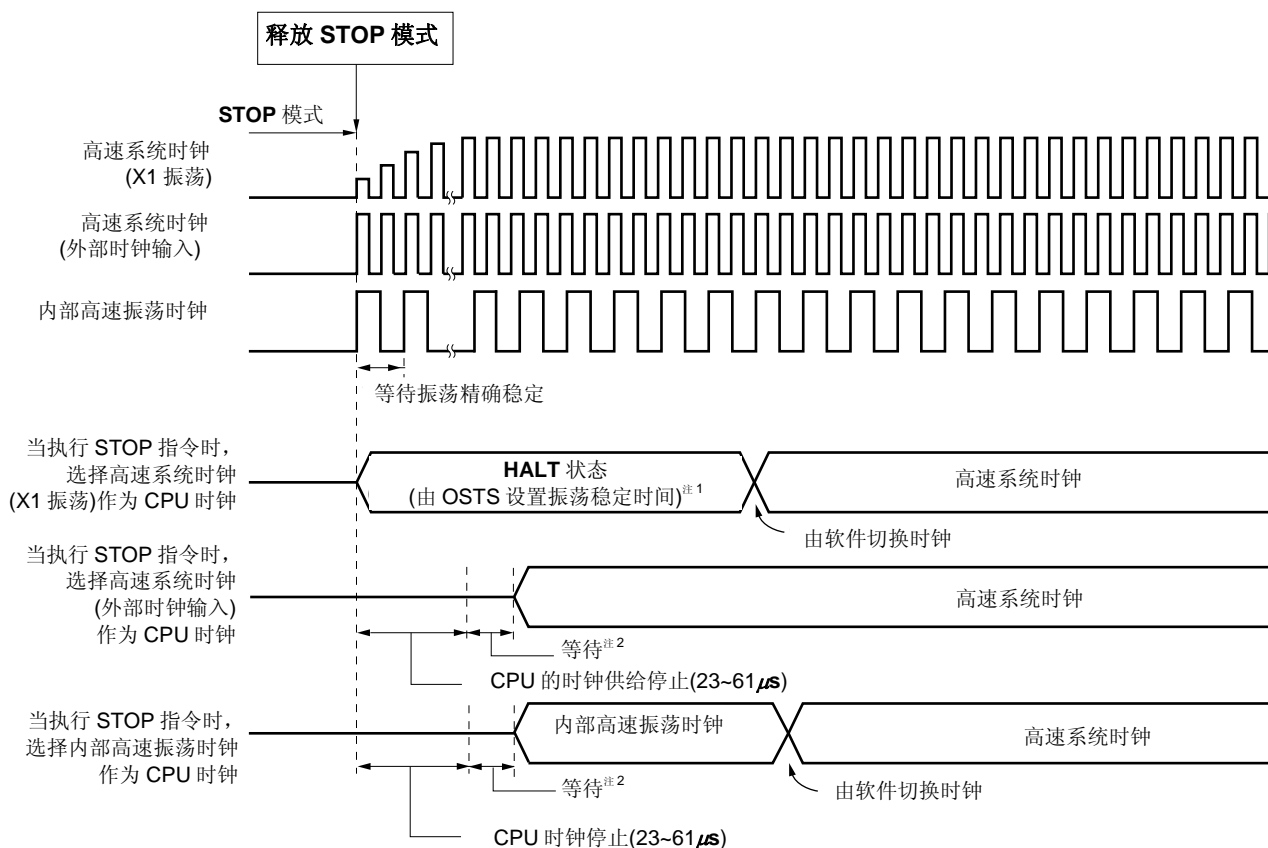
备注

- f_H: 内部高速振荡时钟
- f_x: X1 时钟
- f_{EX}: 外部主系统时钟
- f_{XT}: XT1 时钟
- f_L: 内部低速振荡时钟

- 注意事项
1. 在 STOP 模式释放后，要使用在 STOP 模式下停止操作的外围硬件，以及在 STOP 模式下时钟停止操作的外围硬件，必须重新启动外围硬件。
 2. 要在 STOP 模式下停止内部低速振荡时钟，可使用选项字节停止 HALT/STOP 模式下看门狗定时器操作（000C0H 的第 0 位（WDSTBYON）= 0），然后执行 STOP 指令。
 3. 当 CPU 使用高速系统时钟（X1 振荡）时，为了在 STOP 模式释放后缩短振荡稳定时间，可在执行 STOP 指令前将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后，在将 CPU 时钟从内部高速振荡时钟切换到高速系统时钟（X1 振荡）之前，应使用振荡稳定时间计数器的状态寄存器（OSTC）检测振荡稳定时间。

(2) 释放 STOP 模式

图 18-5. 释放 STOP 模式时的操作时序



注 当 OSTC 设置的振荡稳定时间等于或小于 61μs，HALT 状态最多持续“61μs + 等待时间”。

STOP 模式可通过以下两种方式释放。

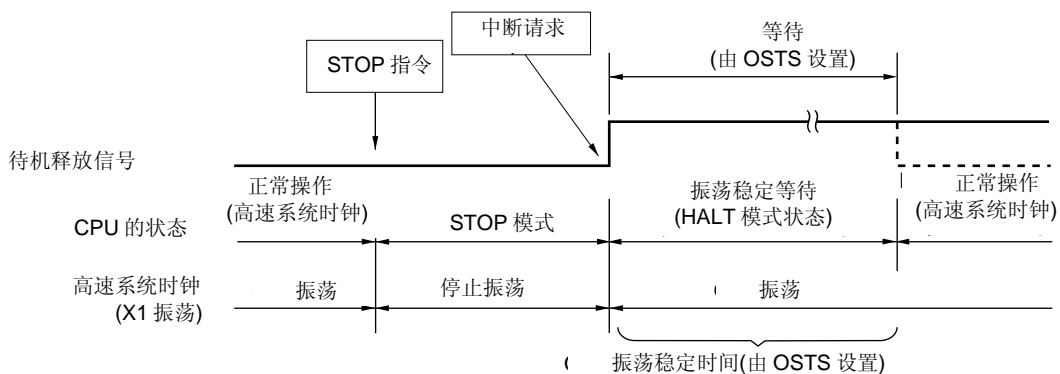
(a) 由未被屏蔽的中断请求释放

当产生一个未被屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

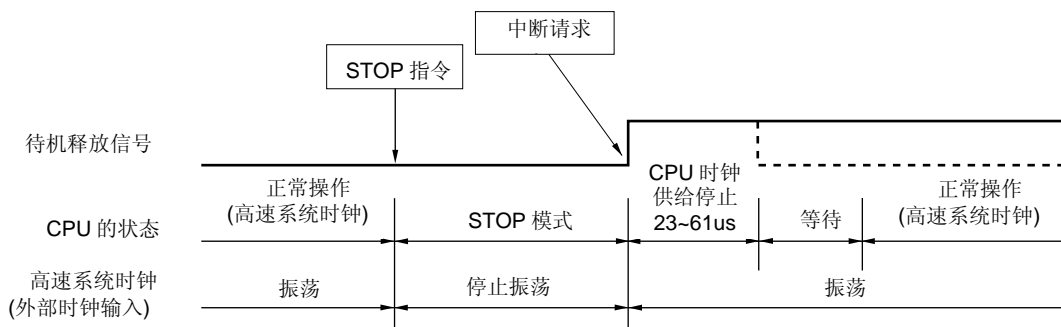
<R>

图 18-6. 通过产生中断请求释放 STOP 模式 (1/2)

(1) 当 CPU 使用高速系统时钟 (X1 振荡器)



(2) 当 CPU 使用高速系统时钟 (外部时钟输入)



<R>

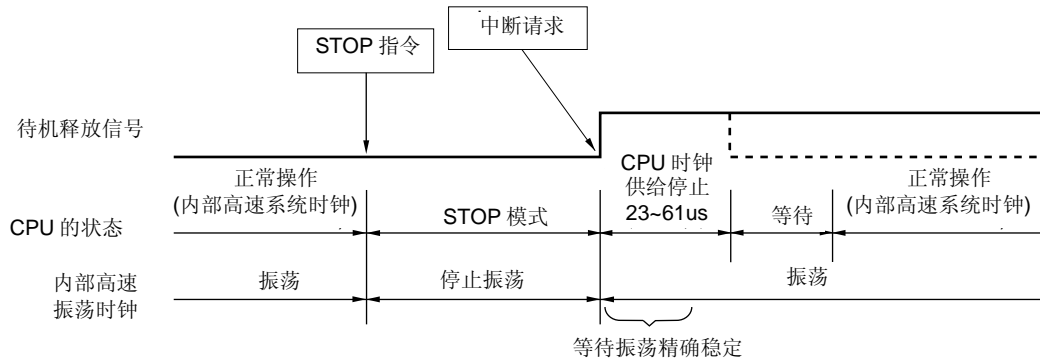
注 等待时间如下：

- 当进行向量中断服务时： 10 ~12 个时钟
- 未进行向量中断服务时： 5 ~6 个时钟

备注 虚线表示释放待机模式的中断请求被响应的情况。

图 18-6. 通过产生中断请求释放 STOP 模式 (2/2)

(3) 当 CPU 使用内部高速振荡时钟



<R>

注 等待时间如下:

- 当进行向量中断服务时: 10 ~ 12 个时钟
- 未进行向量中断服务时: 5 ~ 6 个时钟

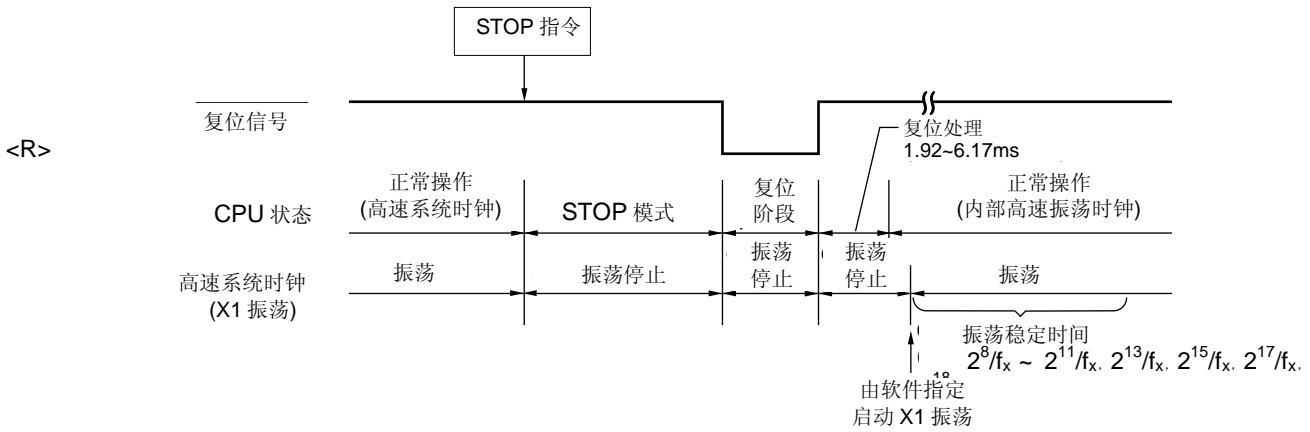
备注 虚线表示释放待机模式的中断请求被响应的情况。

(b) 由复位信号的产生释放

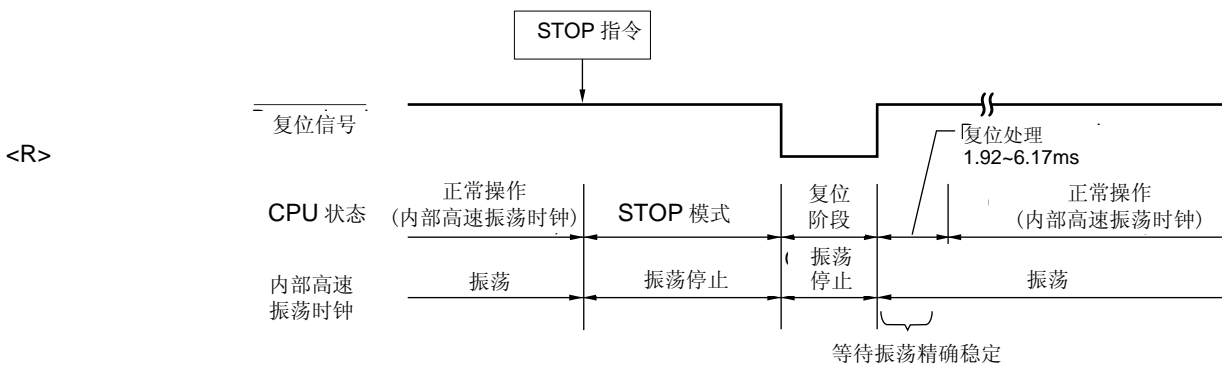
当产生复位信号时，释放 STOP 模式，然后在进行正常复位操作时，跳转到复位向量地址后，程序开始执行。

图 18-7. 由复位释放 STOP 模式

(1) 当 CPU 使用高速系统时钟



(2) 当 CPU 使用内部高速振荡时钟



备注 fx: X1 时钟振荡频率

第十九章 复位功能

以下 5 种操作用于产生复位信号。

- <R>
- (1) 由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号
 - (2) 由看门狗定时器程序循环检测引起的内部复位
 - (3) 通过比较上电清零 (POC) 电路的检测电压和供电电压引起的内部复位
 - (4) 通过比较低电压检测电路 (LVI) 的检测电压和供电电压引起的内部复位
 - (5) 由执行非法指令引起的内部复位^注

外部复位与内部复位在功能上没有什么区别。当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

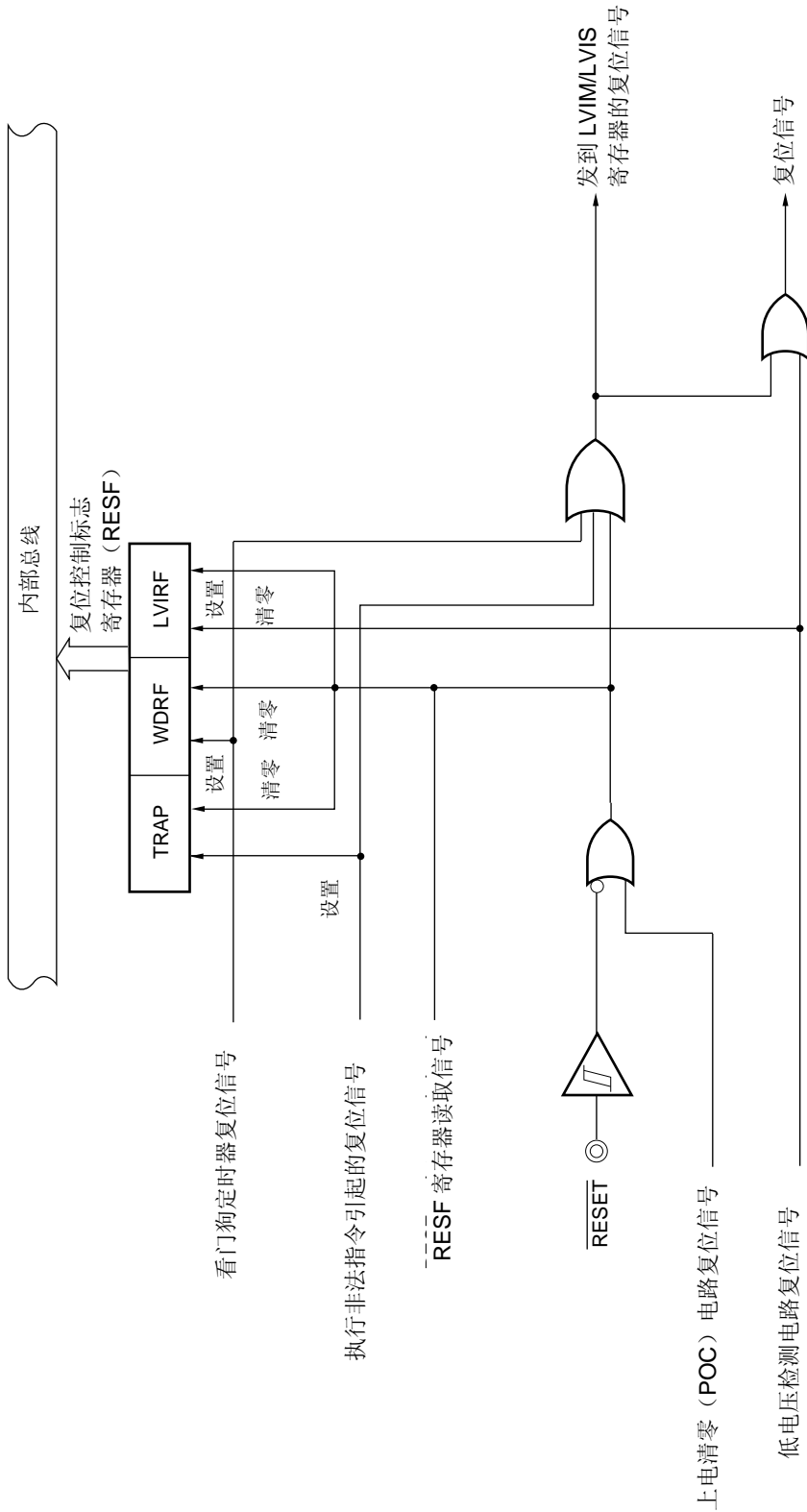
$\overline{\text{RESET}}$ 引脚输入为低电平、看门狗定时器溢出或 POC 和 LVI 电路的电压检测或执行非法指令^注，都可以引起复位。硬件的每项设置状态如表 19-1 和 19-2 所示。在复位信号产生期间或在复位释放后振荡稳定时间内，每个引脚均为高阻抗，但是 P130 除外，它是低电平输出。

当 $\overline{\text{RESET}}$ 引脚输入低电平时，设备被复位。当 $\overline{\text{RESET}}$ 引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序（参见图 19-2 ~ 19-4）。由 POC 和 LVI 电路供电电压检测引起的复位，在复位后当 $V_{DD} \geq V_{POC}$ 或 $V_{DD} \geq V_{LVI}$ 时自动释放，并使用内部高速振荡时钟执行程序（参见第二十章 上电清零电路 和 第二十一章 低电压检测电路）。

注 当执行指令代码 FFH 时，产生非法指令。
非法指令执行引起的复位无法由在线仿真器或者片上调试仿真器的仿真发出。

- 注意事项**
1. 对于外部复位，输入 $\overline{\text{RESET}}$ 引脚的低电平的时间至少应为 10 μs 。
(上电之后，如果外部复位有效，那么供电电压在有效操作电压 ($V_{DD} < 1.8\text{V}$) 之外的时间不计在 10 μs 中。但是在 POC 释放之前，低电平输入可能会继续。)
 2. 复位输入期间，X1 时钟、XT1 时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入无效。
 3. 在通过复位释放 STOP 模式时，复位输入期间保持 STOP 模式下的 RAM 内容。但是，因为 SFR 和第二 SFR 被初始化，所以除了设置为低电平输出的 P130 外，所有端口引脚都变成高阻态。

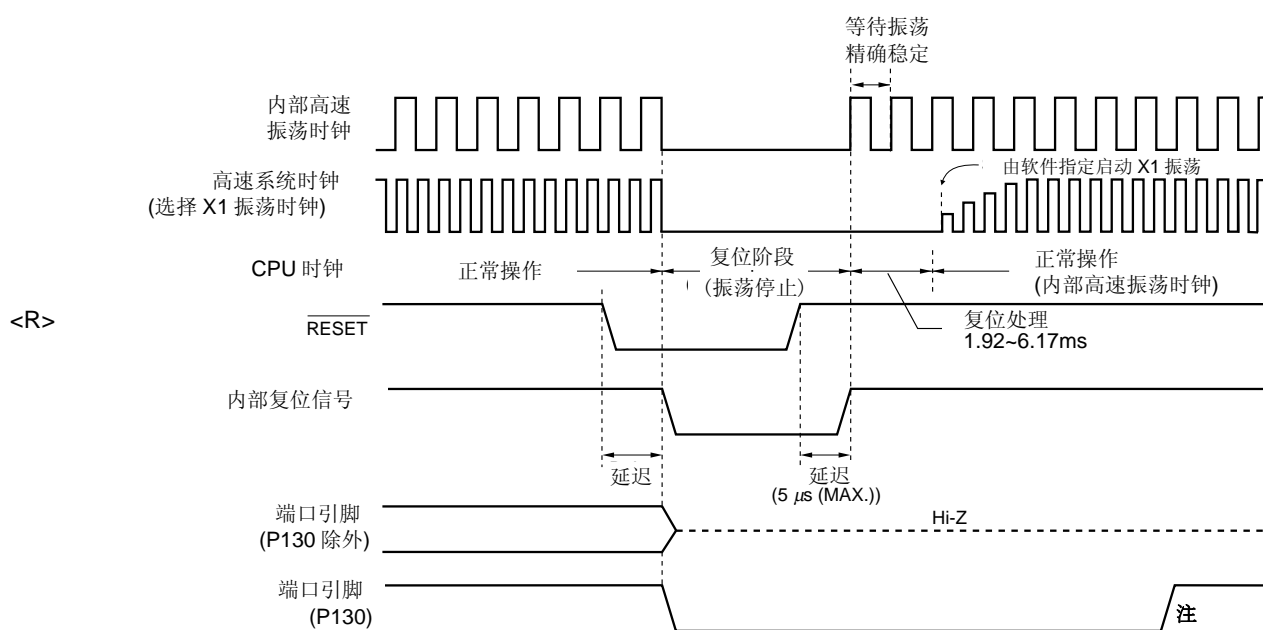
图 19-1. 复位功能的框图



注意事项 LVI 电路内部复位功能不能复位 LVI 电路。

- 备注
- 1. LVIM: 低电压检测寄存器
 - 2. LVIS: 低电压检测电平选择寄存器

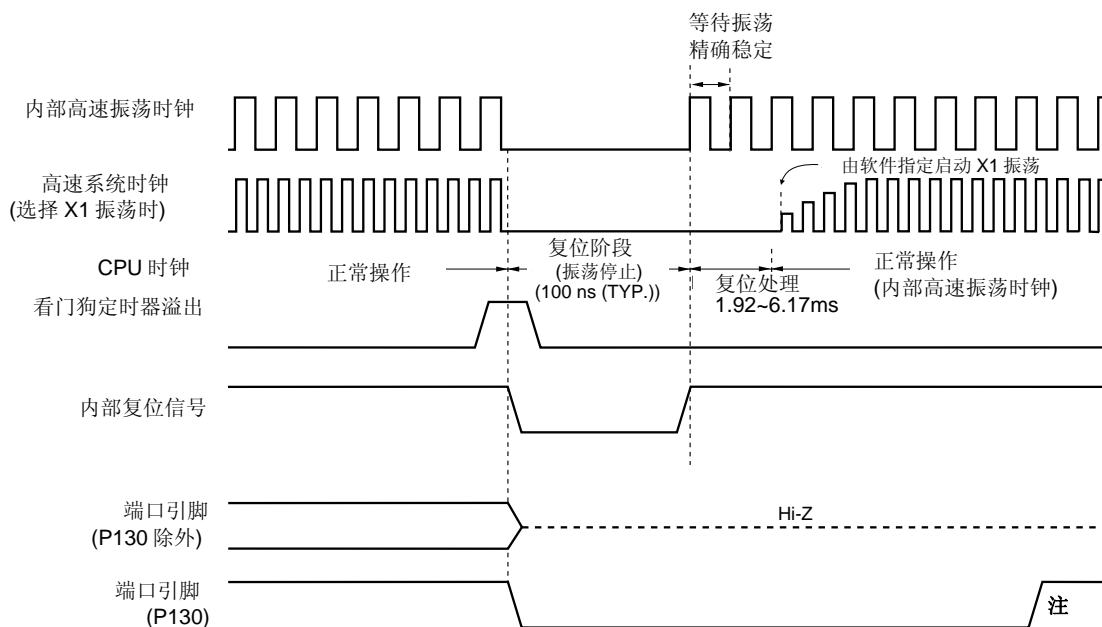
图 19-2. 由 RESET 输入进行复位的时序



注 由软件设置 P130 为高电平输出。

备注 当进行复位时，P130 输出低电平。如果在复位进行之前 P130 设置为输出高电平，那么 P130 的输出信号可以作为 CPU 复位信号的伪输出 (dummy-output)。

图 19-3. 由于看门狗定时器溢出进行复位的时序

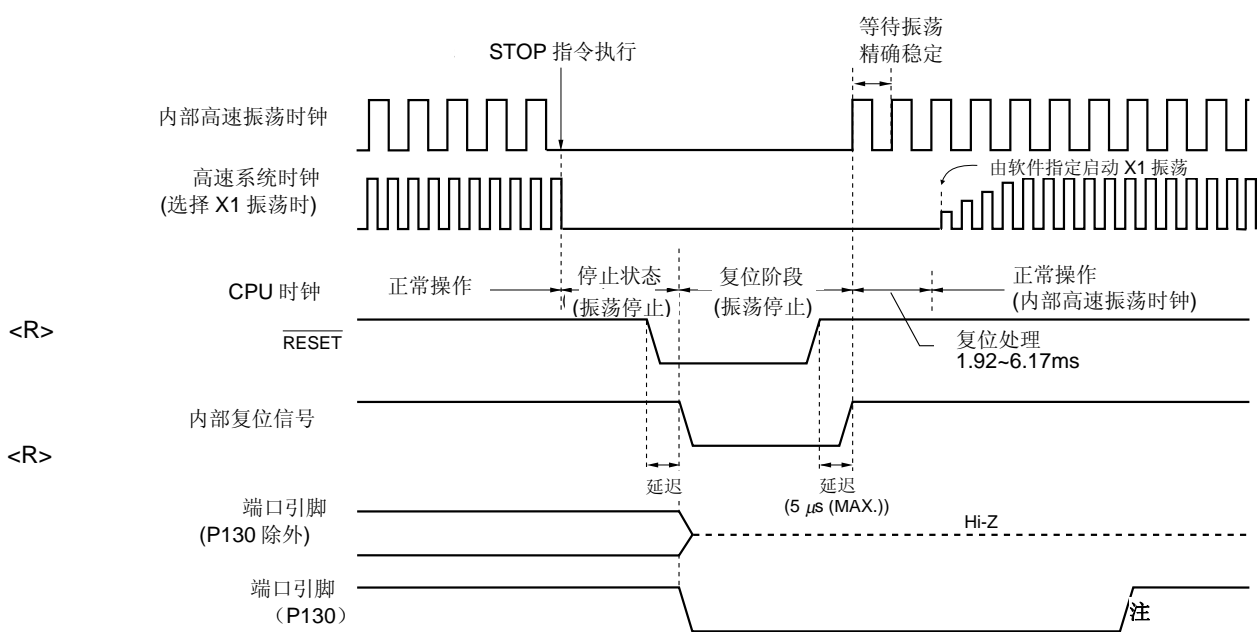


注 由软件设置 P130 为高电平输出。

注意事项 看门狗定时器内部复位会将看门狗定时器复位。

备注 当进行复位时，P130 输出低电平。如果在复位进行之前 P130 设置为输出高电平，那么 P130 的输出信号可以作为 CPU 复位信号的伪输出 (dummy-output)。

图 19-4. 在 STOP 模式下通过 RESET 输入进行复位的时序



注 由软件设置 P130 为高电平输出。

- 备注**
1. 当进行复位时，P130 输出低电平。如果在复位进行之前 P130 设置为输出高电平，那么 P130 的输出信号可以作为 CPU 复位信号的伪输出（dummy-output）。
 2. 关于上电清零电路和低电压检测器的复位时序，请看第二十章 上电清零电路和第二十一章 低电压检测电路。

表 19-1. 复位期间的操作状态

项目	复位期间		
系统时钟	CPU 时钟 停止		
主系统时钟	f _H	操作停止	
	f _X	操作停止 (X1 和 X2 引脚为输入端口模式)	
	f _{EX}	时钟输入无效 (引脚为输入端口模式)	
副系统时钟	f _{XT}	操作停止 (XT1 和 XT2 引脚为输入端口模式)	
f _L	操作停止		
CPU			
Flash 存储器	在低电流消耗模式下可操作		
RAM	操作停止		
端口 (锁存器)	操作停止		
定时器阵列单元 (TAU)			
实时计数器 (RTC)			
钟表定时器			
看门狗定时器			
时钟输出/蜂鸣器输出			
A/D 转换器			
D/A 转换器			
串行阵列单元 (SAU)			
乘法器			
上电清零功能			可以操作
低电压检测功能			操作停止 (但是, 在 LVI 复位时操作继续)
外部中断			操作停止

备注

- f_H: 内部高速振荡时钟
- f_X: X1 振荡时钟
- f_{EX}: 外部主系统时钟
- f_{XT}: XT1 振荡时钟
- f_L: 内部低速振荡时钟

表 19-2. 复位响应后的硬件状态 (1/3)

硬件		复位响应后 ^{#1}
程序计数器 r (PC)		设置复位向量表的内容 (0000H、0001H)
堆栈指针 (SP)		不确定
程序状态字 (PSW)		06H
RAM	数据存储器	不确定 ^{#2}
	通用寄存器	不确定 ^{#2}
端口寄存器 (P0 ~ P7, P11 ~ P14) (输出锁存器)		00H
端口模式寄存器 (PM0 ~ PM7, PM9, PM11, PM12, PM14)		FFH
端口输入模式寄存器 0, 4, 14 (PIM0, PIM4, PIM14)		00H
端口输出模式寄存器 0, 4, 14 (POM0, POM4, POM14)		00H
上拉电阻选项寄存器 (PU0, PU1, PU3 ~ PU5, PU7, PU9, PU12, PU14)		00H
时钟操作模式控制寄存器 (CMC)		00H
时钟操作状态控制寄存器 (CSC)		C0H
处理器模式控制寄存器 (PMC)		00H
系统时钟控制寄存器 (CKC)		09H
振荡稳定时间计数器状态寄存器 (OSTC)		00H
振荡稳定时间选择寄存器 (OSTS)		07H
噪声滤波允许寄存器 0, 1 (NFEN0, NFEN1)		00H
外围设备允许寄存器 0 (PER0)		00H
内部高速振荡器调整寄存器 (HIOTRM)		10H
操作速度模式控制寄存器 (OSMC)		00H
定时器阵列单元 (TAU)	定时器数据寄存器 00, 01, 02, 03, 04, 05, 06, 07 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07)	0000H
	定时器模式寄存器 00, 01, 02, 03, 04, 05, 06, 07 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07)	0000H
	定时器状态寄存器 00, 01, 02, 03, 04, 05, 06, 07 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07)	0000H
	定时输入选择寄存器 0 (TIS0)	00H
	定时器通道计数器寄存器 00, 01, 02, 03, 04, 05, 06, 07 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07)	FFFFH
	定时器通道允许状态寄存器 0 (TE0)	0000H
	定时器通道开始触发寄存器 0 (TS0)	0000H
	定时器通道停止触发寄存器 0 (TT0)	0000H
	定时器时钟选择寄存器 0 (TPS0)	0000H
	定时器通道输出寄存器 0 (TO0)	0000H
	定时器通道输出允许寄存器 0 (TOE0)	0000H
	定时器通道输出电平寄存器 0 (TOL0)	0000H
	定时器通道输出模式寄存器 0 (TOM0)	0000H

- 注
1. 在复位信号产生期间或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 2. 当在待机模式下执行复位时，即使在复位后，预设状态也可以保持。

表 19-2. 复位响应后的硬件状态 (2/3)

硬件	复位响应后 ^{注1}	
实时计数器	副计数寄存器 (RSUBC)	0000H
	秒计数寄存器 (SEC)	00H
	分钟计数寄存器 (MIN)	00H
	小时计数寄存器 (HOUR)	12H
	星期计数寄存器 (WEEK)	00H
	日计数寄存器 (DAY)	01H
	月计数寄存器 (MONTH)	01H
	年计数寄存器 (YEAR)	00H
	钟表错误修正寄存器 (SUBCUD)	00H
	分钟报警寄存器 (ALARMWM)	00H
	小时报警寄存器 (ALARMWH)	12H
	星期报警寄存器 (ALARMWW)	00H
	实时计数器控制寄存器 0 (RTCC0)	00H
	实时计数器控制寄存器 1 (RTCC1)	00H
实时计数器控制寄存器 2 (RTCC2)	00H	
时钟输出/蜂鸣器输出控制器	时钟输出选择寄存器 0, 1 (CKS0, CKS1)	00H
看门狗定时器	允许寄存器 (WDTE)	1AH/9AH ^{注2}
A/D 转换器	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模式输入通道指定寄存器 (ADS)	00H
	A/D 端口配置寄存器 (ADPC)	10H
D/A 转换器	8 位 D/A 转换值设置寄存器 0, 1 (DACS0, DACS1)	00H
	D/A 转换模式寄存器 (DAM)	00H
串行阵列单元 (SAU)	串行数据寄存器 00, 01, 02, 03, 10, 11, 12, 13 (SDR00, SDR01, SDR02, SDR03, SDR10, SDR11, SDR12, SDR13)	0000H
	串行状态寄存器 00, 01, 02, 03, 10, 11, 12, 13 (SSR00, SSR01, SSR02, SSR03, SSR10, SSR11, SSR12, SSR13)	0000H
	串行标志清零触发寄存器 00, 01, 02, 03, 10, 11, 12, 13 (SIR00, SIR01, SIR02, SIR03, SIR10, SIR11, SIR12, SIR13)	0000H
	串行模式寄存器 00, 01, 02, 03, 10, 11, 12, 13 (SMR00, SMR01, SMR02, SMR03, SMR10, SMR11, SMR12, SMR13)	0020H
	串行通信操作设置寄存器 00, 01, 02, 03, 10, 11, 12, 13 (SCR00, SCR01, SCR02, SCR03, SCR10, SCR11, SCR12, SCR13)	0087H
	串行通道允许状态寄存器 0, 1 (SE0, SE1)	0000H
	串行通道开始触发寄存器 0, 1 (SS0, SS1)	0000H
	串行通道停止触发寄存器 0, 1 (ST0, ST1)	0000H
	串行时钟选择寄存器 0, 1 (SPS0, SPS1)	0000H
	串行输出寄存器 0, 1 (SO0, SO1)	0F0FH
	串行输出允许寄存器 0, 1 (SOE0, SOE1)	0000H
	输入切换控制寄存器 (ISC)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 2. WDTE 的复位数值由选项字节的设置来决定。

表 19-2. 复位响应后的硬件状态 (3/3)

硬件		复位响应后 ^{#1}
串行接口 IIC0	移位寄存器 0 (IIC0)	00H
	控制寄存器 0 (IICC0)	00H
	从地址寄存器 0 (SVA0)	00H
	时钟选择寄存器 0 (IICCL0)	00H
	功能扩展寄存器 0 (IICX0)	00H
	状态寄存器 0 (IICS0)	00H
	标志寄存器 0 (IICF0)	00H
乘法器	乘法输入数据寄存器 A (MULA)	0000H
	乘法输入数据寄存器 B (MULB)	0000H
	高位乘法结果存储寄存器 (MULOH)	0000H
	低位乘法结果存储寄存器 (MULOL)	0000H
按键中断	按键返回模式寄存器 (KRM)	00H
复位功能	复位控制标志寄存器 (RESF)	00H ^{#2}
低电压检测器	低电压检测寄存器 (LVIM)	00H ^{#3}
	低电压检测电平选择寄存器 (LVIS)	0EH ^{#2}
DMA 控制器	SFR 地址寄存器 0, 1 (DSA0, DSA1)	00H
	RAM 地址寄存器 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	字节计数寄存器 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	模式控制寄存器 0, 1 (DMC0, DMC1)	00H
	操作控制寄存器 0, 1 (DRC0, DRC1)	00H
中断	请求标志寄存器 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	屏蔽标志寄存器 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	优先级指定标志寄存器 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR10L, PR10H, PR11L, PR11H, PR02L, PR02H, PR12L, PR12H)	FFH
	外部中断上升沿允许寄存器 (EGP0, EGP1)	00H
	外部中断下降沿允许寄存器 (EGN0, EGN1)	00H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 2. 这些值根据复位源的不同而变化。

寄存器 \ 复位源		RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
		RESF	TRAP 位	清零 (0)	清零 (0)	设置 (1)
	WDRF 位			保持	设置 (1)	保持
	LVIRF 位			保持	保持	设置 (1)
LVIS		清除 (0EH)	清除 (0EH)	清除 (0EH)	清除 (0EH)	保持

3. 此数值依据复位源和选项字节的的不同而变化。

19.1 确认复位源的寄存器

78K0R/KF3 中有许多内部复位源。复位控制标志寄存器（RESF）用于存储产生复位请求的复位源。

RESF 可由 8 位存储器操作指令读取。

RESET 输入、由 POC 电路引起的复位输入、以及读取 RESF，都可以将 RESF 清零（00H）。

图 19-5. 复位控制标志寄存器（RESF）的格式

地址： FFFA8H 复位后： 00H^{#1} R

符号	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDRF	0	0	0	LVIRF

TRAP	由执行非法指令引起内部复位请求 ^{#2}
0	不产生内部复位请求，或对 RESF 清零
1	产生内部复位请求

WDRF	看门狗定时器产生的内部复位请求（WDT）
0	不产生内部复位请求，或对 RESF 清零
1	产生内部复位请求

LVIRF	由低电压检测电路（LVI）产生的内部复位请求
0	不产生内部复位请求，或对 RESF 清零
1	产生内部复位请求

- 注
1. 复位后的值依据复位源的不同而变化。
 2. 当执行指令码 FFH 时。
非法指令执行引起的复位无法由在线仿真器或者片上调试仿真器的仿真发出。

- 注意事项
1. 不要用 1 位存储器操作指令读取数据。
 2. 当使用 LVI 默认启动功能（000C1H 的第 0 位（LVIOFF）= 0）时，LVIRF 标志根据上电的波形可能变为 1。

<R>

产生复位请求时的 RESF 状态如表 19-3 所示。

表 19-3. 产生复位请求时的 RESF 状态

复位源 / 标志	RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
TRAP	清零 (0)	清零 (0)	设置 (1)	保持	保持
WDRF			保持	设置 (1)	保持
LVIRF			保持	保持	设置 (1)

第二十章 上电清零电路

20.1 上电清零电路的功能

上电清零电路（POC）有以下功能。

- 在上电时产生内部复位信号。
当供电电压（ V_{DD} ）超过 $1.59\text{ V} \pm 0.09\text{ V}^{\text{注}}$ 时，释放复位信号。

注意事项 如果低电压检测器（LVI）由可选字节的默认值设置为 ON，那么直到供电电压（ V_{DD} ）超过 $2.07\text{ V} \pm 0.2\text{ V}^{\text{注}}$ 才会释放复位信号。

- 比较供电电压（ V_{DD} ）和检测电压（ $V_{POC} = 1.59\text{ V} \pm 0.09\text{ V}^{\text{注}}$ ），并在 $V_{DD} < V_{POC}$ 时产生内部复位信号

注 这些都是初步数据，可能会改变。

注意事项 如果由 POC 电路产生内部复位信号，则将复位控制标志寄存器（RESF）清零（00H）。

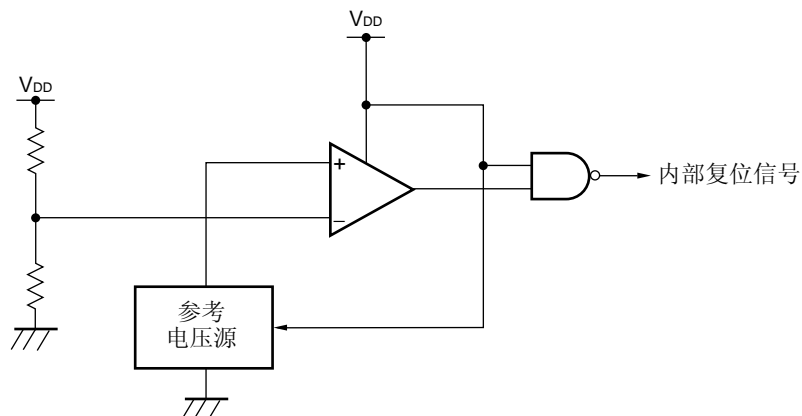
备注 此产品集成了多种可以产生内部复位信号的硬件功能。复位控制标志寄存器 RESF 中有一个标志用于指示复位源，用于指示内部复位信号是看门狗定时器（WDT）、低电压检测电路（LVI）或者执行非法指令产生的。当内部复位信号由 WDT 或 LVI 产生时，不对 RESF 清零（00H），标志位设置为 1。

如需了解 RESF 的详细内容，参见第十九章 复位功能。

20.2 上电清零电路的配置

上电清零电路的框图如图 20-1 所示。

图 20-1. 上电清零电路的框图



20.3 上电清零电路的操作

- 在上电时产生内部复位信号。当供电电压 (V_{DD}) 超过 $1.59\text{ V} \pm 0.09\text{ V}^{\#}$ 时，释放复位信号。

注意事项 如果低电压检测器 (LVI) 由可选字节的默认值设置为 ON，那么直到供电电压 (V_{DD}) 超过 $2.07\text{ V} \pm 0.2\text{ V}^{\#}$ 才会释放复位信号。

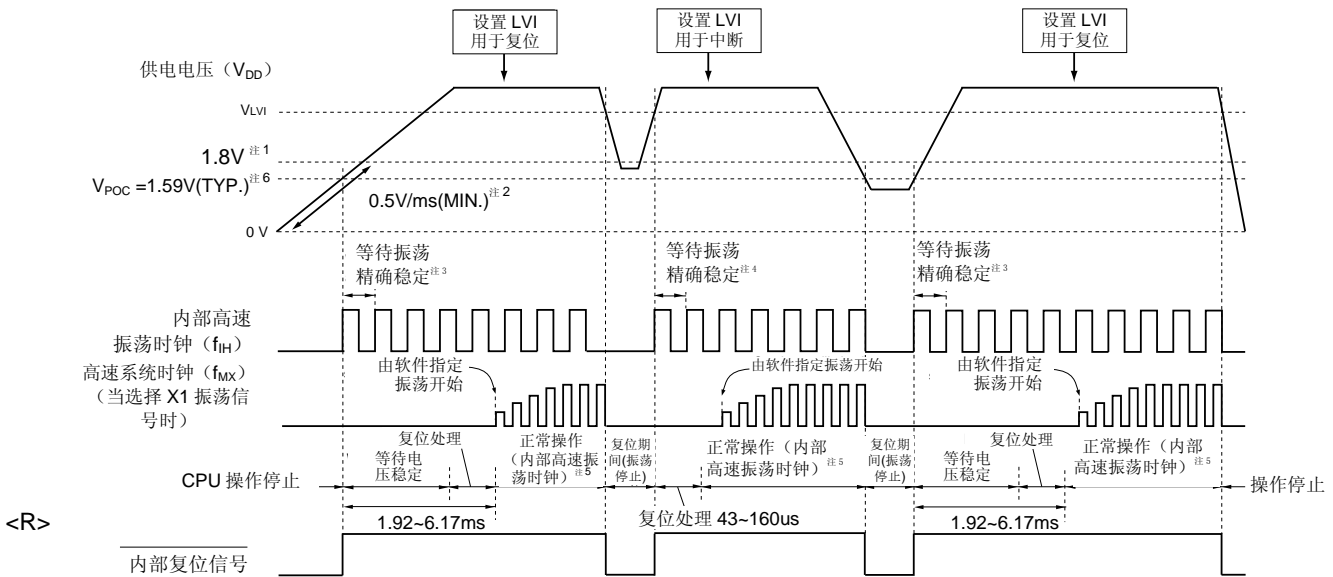
- 比较供电电压 (V_{DD}) 和检测电压 ($V_{POC} = 1.59\text{ V} \pm 0.09\text{ V}^{\#}$)，并在 $V_{DD} < V_{POC}$ 时产生内部复位信号。

注 这些都是初步数据，可能会改变。

由上电清零电路和低电压检测电路产生内部复位信号的时序如下所示。

图 20-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)

(1) 当上电之后 LVI 为 OFF 时 (可选字节: LVIOFF = 1)



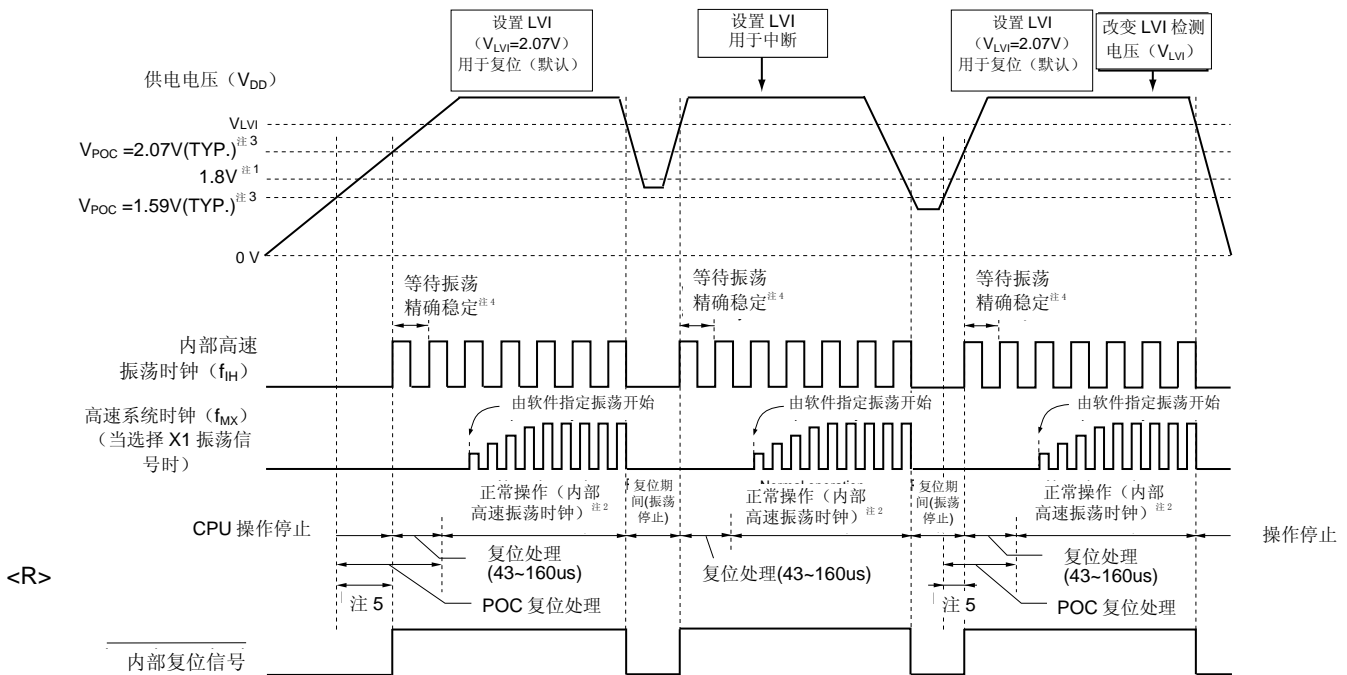
- 注
1. 可靠操作的电压范围: $1.8V \leq V_{DD} \leq 5.5V$ 。为了在供电电压下降时,使低于 $1.8V$ 的状态为复位状态,可以使用低电压检测器的复位功能,或通过 **RESET** 引脚输入低电平。
 2. 如果上电后电压上升到 $1.8V$ 的速率小于 $0.5V/ms$ (MIN.),则在电压达到 $1.8V$ 以前输入一个低电平到 **RESET** 引脚,或者通过使用选项字节 (选项字节: $LVIOFF = 0$) 把 LVI 默认设置为 ON。
 3. 内部电压稳定时间包括内部高速振荡时钟的振荡精度稳定时间。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精度稳定时间。
 5. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟,可以通过 OSTC 确认是否经过了振荡稳定时间。而如果要使用 XT1 时钟,可以通过定时器功能来确认是否经过了振荡稳定时间。
 6. 这些都是初步数据,可能会改变。

注意事项 在释放复位状态后通过软件设置低电压检测电路 (参见第二十一章 低电压检测电路)。

备注 V_{LVI} : LVI 检测电压
 V_{POC} : POC 检测电压

图 20-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (2/2)

(2) 当上电之后 LVI 为 ON 时 (选项字节: LVIOFF = 0)



- 注
1. 可靠操作的电压范围: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。为了在供电电压下降时,使低于 1.8 V 的状态为复位状态,可以使用低电压检测器的复位功能,或通过 **RESET** 引脚输入低电平。
 2. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟,可以通过 **OSTC** 确认是否经过了振荡稳定时间。而如果要使用 XT1 时钟,可以通过定时器功能来确认是否经过了振荡稳定时间。
 3. 这些都是初步数据,可能会变化。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精度稳定时间。
 5. 下面的时间是到达 POC 检测电压 (1.59 V (TYP.)) 与开始正常操作之间需要的时间。
 - 从 1.59 V (TYP.) 到 2.07 V (TYP.) 的时间小于 6.17 ms :
从 1.59 V (TYP.) 到开始正常操作之间要 $1.92 \sim 6.33\text{ ms}$ 的 POC 处理时间。
 - 从 1.59 V (TYP.) 到 2.07 V (TYP.) 的时间大于 6.17 ms :
从 2.07 V (TYP.) 到开始正常操作之间要 $43 \sim 160\text{ us}$ 的复位处理时间。

注意事项 在释放复位状态后通过软件设置低电压检测电路 (参见第二十一章 低电压检测电路)。

备注 V_{LVI} : LVI 检测电压
 V_{POC} : POC 检测电压

20.4 上电清零电路使用注意事项

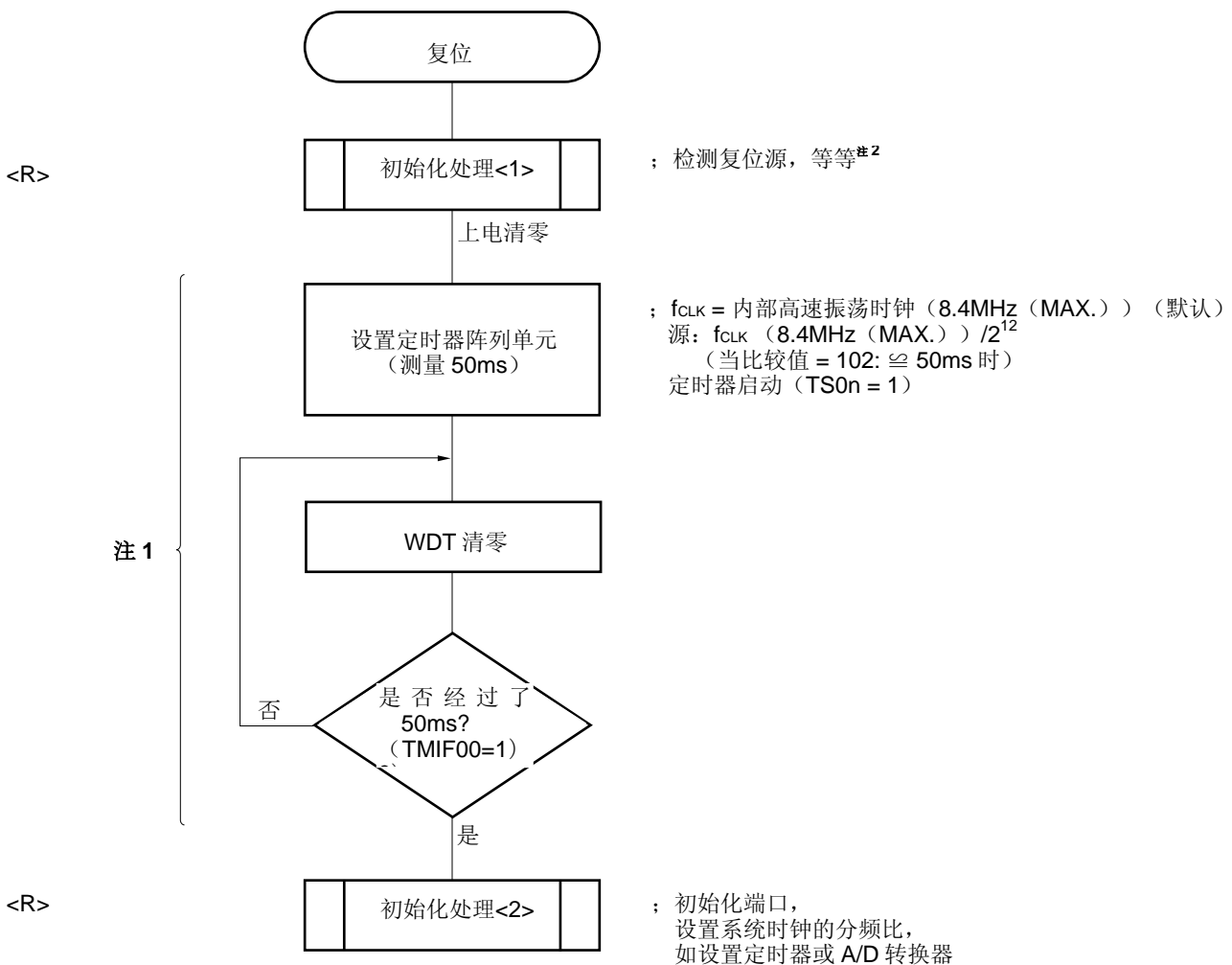
在系统中，如果供电电压（V_{DD}）在接近 POC 检测电压（V_{POC}）时的一段时间内产生波动，则系统可能会反复复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>

在释放复位信号后，通过使用定时器的软件计数器等待系统供电电压的波动周期，然后对端口初始化。

图 20-3. 复位释放后软件处理过程示例（1/2）

- 如果供电电压接近 POC 检测电压，其波动时间 ≤ 50 ms



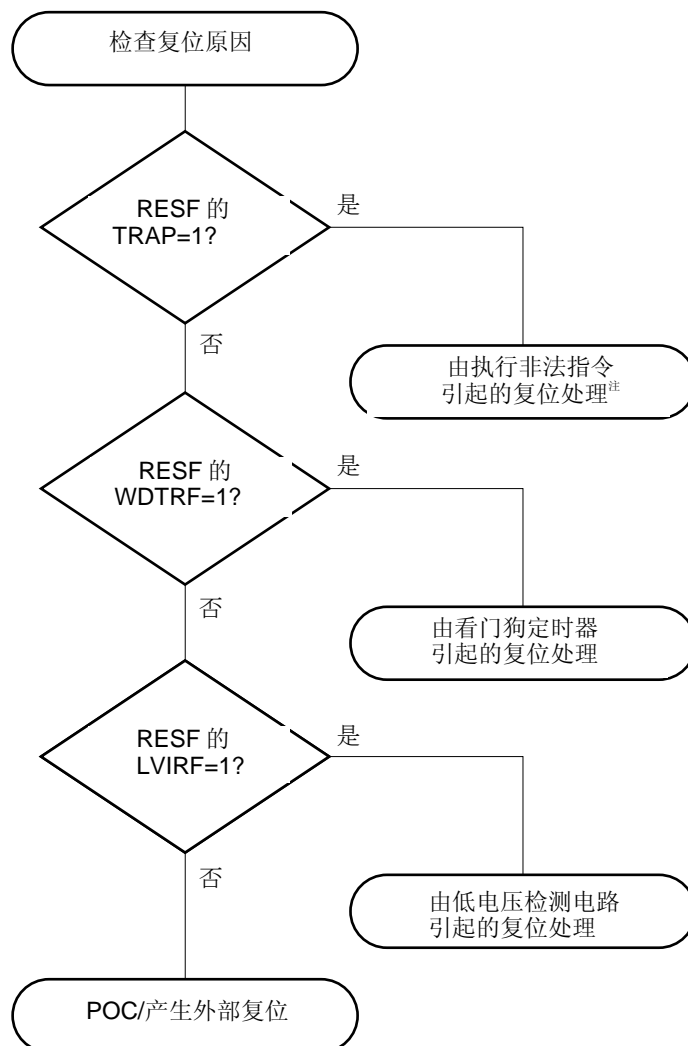
- 注
1. 如果在此期间再次产生复位，则不启动初始化操作<2>。
 2. 流程图显示在下页。

备注 n: 通道号（n = 0~ 7）

图 20-3. 复位释放后软件处理过程示例 (2/2)

• 检测复位原因

<R>



<R>

注 当执行指令代码 FFH 时，产生非法指令。
非法指令执行引起的复位无法由在线仿真器或者片上调试仿真器的仿真发出。

第二十一章 低电压检测电路

21.1 低电压检测电路的功能

低电压检测电路 (LVI) 有以下功能。

- LVI 电路比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，或者比较从外部输入引脚 (EXLVI) 的输入电压与检测电压 ($V_{EXLVI} = 1.21\text{ V} \pm 0.1\text{ V}^{\text{注}}$)，并产生内部复位信号或内部中断信号。
- 低电压检测电路 (LVI) 可由选项字节设置为默认 ON。如果设置为 ON，供电电压从 POC 检测电压或更低电压开始有效，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07\text{ V} \pm 0.2\text{ V}^{\text{注}}$) 时，会产生内部复位信号。之后，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07\text{ V} \pm 0.1\text{ V}^{\text{注}}$) 时，产生内部复位信号。
- 可由软件选择检测供电电压 (V_{DD}) 或外部输入引脚的输入电压 (EXLVI)。
- 可由软件选择复位或中断操作。
- 可通过软件改变供电电压的检测电平 (V_{LVI} , 16 电平)。
- 在 STOP 模式下可以操作。

注 这只是初步数据，可能会改变。

根据软件的设置，可产生以下复位和中断信号。

供电电压 (V_{DD}) 的电平检测选择 (LVISEL = 0)		从外部输入引脚 (EXLVI) 来的输入电压电平检测选择 (LVISEL = 1)	
选择复位 (LVIMD = 1) .	选择中断 (LVIMD = 0) .	选择复位 (LVIMD = 1) .	选择中断 (LVIMD = 0) .
当 $V_{DD} < V_{LVI}$ 时，产生内部复位信号，并且当 $V_{DD} \geq V_{LVI}$ 时，释放复位信号。	当 V_{DD} 降至低于 V_{LVI} ($V_{DD} < V_{LVI}$) 时，或者当 V_{DD} 变为 V_{LVI} 或更高时 ($V_{DD} \geq V_{LVI}$) 时，产生内部中断信号。	当 $EXLVI < V_{EXLVI}$ 时，产生内部复位信号，并且当 $EXLVI \geq V_{EXLVI}$ 时，释放复位信号。	当 EXLVI 降至低于 V_{EXLVI} ($EXLVI < V_{EXLVI}$) 时，或者当 EXLVI 变为 V_{EXLVI} 或更高时 ($EXLVI \geq V_{EXLVI}$)，产生内部中断信号。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的第 2 位

LVIMD: LVIM 的第 1 位

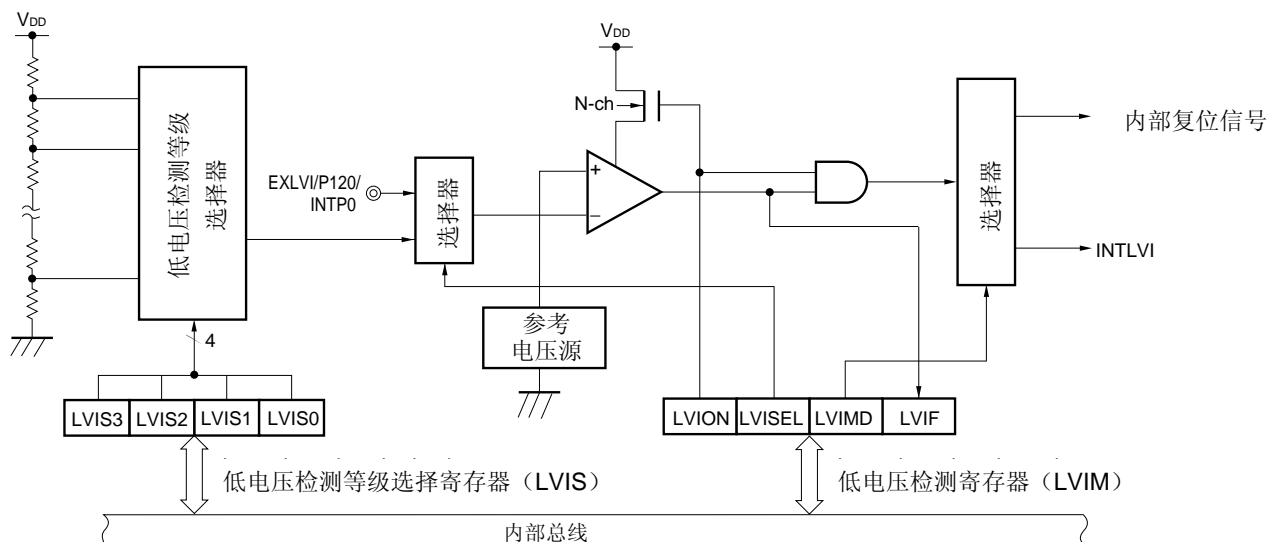
当低电压检测电路运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIM 的第 0 位: LVIF) 得知其是否高于或低于检测电平。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。需要了解 RESF 的详细信息，请参见 **第十九章 复位功能**。

21.2 低电压检测电路的配置

低电压检测电路的框图如图 21-1 所示。

图 21-1. 低电压检测电路的框图



21.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

该寄存器用于设置低电压检测和操作模式。

可以通过 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生将 LVIM 设置为 00H。

图 21-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFFA9H 复位后: 00H^{注1} R/W^{注2}

符号	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION ^{注3, 4}	允许低电压检测操作 n
0	禁止操作
1	允许操作

LVISEL ^{注3}	电压检测的选择
0	供电电压 (V _{DD}) 的检测电平
1	外部输入引脚的输入电压 (EXLVI) 的检测电平

LVIMD	低电压检测操作模式 (中断/复位) 的选择
0	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 V_{DD} 降至低于检测电压 V_{LVI} (V_{DD} < V_{LVI}) 时, 或者当 V_{DD} 变为 V_{LVI} 或更高时 (V_{DD} ≥ V_{LVI}) 时, 产生内部中断信号。 LVISEL = 1: 当从外部输入引脚来的输入电压 EXLVI 降至低于检测电压 V_{EXLVI} (EXLVI < V_{EXLVI}) 时, 或者当 EXLVI 变为 V_{EXLVI} 或更高时 (EXLVI ≥ V_{EXLVI}) 时, 产生内部中断信号。
1	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 V_{DD} < 检测电压 V_{LVI} 时, 产生内部复位信号, 并且当 V_{DD} ≥ V_{LVI} 时, 释放复位信号。 LVISEL = 1: 当从外部输入引脚来的输入电压 EXLVI < 检测电压 V_{EXLVI} 时, 产生内部复位信号, 并且当 EXLVI ≥ V_{EXLVI} 时, 释放复位信号。

LVIF	低电压检测标志
0	<ul style="list-style-type: none"> LVISEL = 0: 供电电压 (V_{DD}) ≥ 检测电压 (V_{LVI}), 或当禁止操作时 LVISEL = 1: 外部输入引脚的输入电压 (EXLVI) ≥ 检测电压 (V_{EXLVI}), 或当禁止操作时
1	<ul style="list-style-type: none"> LVISEL = 0: 供电电压 (V_{DD}) < 检测电压 (V_{LVI}) LVISEL = 1: 外部输入引脚的输入电压 (EXLVI) < 检测电压 (V_{EXLVI})

- 注
- 复位数值根据复位源和选项字节的设置而改变。
LVI 复位时, 该寄存器不清零 (00H)。
如果选项字节 LVIOFF = 0, 当 LVI 之外的复位信号产生时, 它设置为“82H”, 否则如果选项字节 LVIOFF = 1, 设置为“00H”。
 - 第 0 位只读。
 - 在产生 LVI 复位以外的复位信号时, LVION、LVIMD 和 LVISEL 被清零 (0)。在产生 LVI 复位时这些位不被清零。

<R> 注 4. 当 LVION 设置为 1 时，LVI 电路中比较器就开始操作。在 LVION 设置为 1 与使用 LVIF 确认电压之间，需要等待一段时间，该时间段由软件指定如下。

- 操作稳定时间 (10 μ s (MAX.))
- 最小脉冲宽度 (200 μ s (MAX.))
- 检测延迟时间 (200 μ s (MAX.))

在此期间不管电压数值为何，LVIF 的数值可以设置/清零，因此就不能使用。同时，在此期间 LVIIIF 中断请求标志设置为 1。

注意事项 1. 在执行以下任一过程后，停止 LVI。

- 当使用 8 位存储器操作指令时： 写入 00H 到 LVIM。
- 当使用 1 位存储器操作指令时： 对 LVION 清零。

2. 外部输入引脚的输入电压 (EXLVI) 必须满足的条件： EXLVI < VDD。

<R> 3. 当 LVI 用作中断模式 (LVIMD = 0)，并且 LVISEL 设置为 0 时，会产生中断请求信号 (INTLVI)，当供电电压 (VDD) 低于或等于检测电压 (VLVI) (如果 LVISEL = 1，外部输入引脚的输入电压 (EXLVI) 低于或等于检测电压 (VEXLVI)) 时，该中断信号禁止 LVI 操作 (LVION 清零)，并且 LVIIIF 可能会被设置为 1。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器用于选择低电压检测电平。

可以通过 1 位或 8 位存储器操作指令设置该寄存器。

复位信号将 LVIS 设置为 0EH。

图 21-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FFFAAH 复位后: 0EH^{注1} R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V _{LV10} (4.22 ±0.1 V) ^{注2}
0	0	0	1	V _{LV11} (4.07 ±0.1 V) ^{注2}
0	0	1	0	V _{LV12} (3.92 ±0.1 V) ^{注2}
0	0	1	1	V _{LV13} (3.76 ±0.1 V) ^{注2}
0	1	0	0	V _{LV14} (3.61 ±0.1 V) ^{注2}
0	1	0	1	V _{LV15} (3.45 ±0.1 V) ^{注2}
0	1	1	0	V _{LV16} (3.30 ±0.1 V) ^{注2}
0	1	1	1	V _{LV17} (3.15 ±0.1 V) ^{注2}
1	0	0	0	V _{LV18} (2.99 ±0.1 V) ^{注2}
1	0	0	1	V _{LV19} (2.84 ±0.1 V) ^{注2}
1	0	1	0	V _{LV10} (2.68 ±0.1 V) ^{注2}
1	0	1	1	V _{LV11} (2.53 ±0.1 V) ^{注2}
1	1	0	0	V _{LV12} (2.38 ±0.1 V) ^{注2}
1	1	0	1	V _{LV13} (2.22 ±0.1 V) ^{注2}
1	1	1	0	V _{LV14} (2.07 ±0.1 V) ^{注2}
1	1	1	1	V _{LV15} (1.91 ±0.1 V) ^{注2}

- 注**
1. 复位数值根据复位源的不同而变化。
如果是由 LVI 引起的复位, LVIS 的数值并不复位, 而是保留当前值。其它原因引起的复位会将其设置为“0EH”。
 2. 这些均为初步数据, 可能会改变。

注意事项 1. 第 4 位 ~ 第 7 位必须清零。

- <R> 注意事项 2. 按照下列的方法之一，可以改变 LVIS 的数值。
- 当停止 LVI 之后，改变数值
 - <1> 停止 LVI (LVION = 1)。
 - <2> 改变 LVIS 寄存器。
 - <3> 设置用作中断模式 (LVIMD = 0)。
 - <4> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <5> 允许 LVI 操作 (LVION = 0)。
 - <6> 在取消 LVI 中断屏蔽 (LVIMK = 0) 之前，用软件清零。因为当 LVI 允许操作时，可能会设置 LVIIF 标志。
 - 当设置用作中断模式 (LVIMD = 0) 后，改变数值
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置用作中断模式 (LVIMD = 0)。
 - <3> 改变 LVIS 寄存器。
 - <4> 在取消 LVI 中断屏蔽 (LVIMK = 0) 之前，用软件清零。因为当改变 LVIS 寄存器时，可能会设置 LVIIF 标志。
3. 当检测到外部输入引脚的输入电压 (EXLVI) 时，检测电压 (V_{EXLVI}) 恒定。因此，不必设置 LVIS。

(3) 端口模式寄存器 12 (PM12)

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电压输入时，设置 PM120 为 1。此时，P120 的输出锁存器的值可以为 0 或 1。

PM12 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将 PM12 设置为 FFH。

图 21-4. 端口模式寄存器 12 (PM12) 的格式

地址: FFF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 引脚 I/O 模式选择
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

21.4 低电压检测器的操作

低电压检测电路能够用于以下两种模式。

(1) 用于复位 (LVIMD = 1)

- 如果 LVISEL = 0, 比较供电电压 (V_{DD}) 与检测电压 (V_{LVI}), 当 $V_{DD} < V_{LVI}$ 时产生内部复位信号, 当 $V_{DD} \geq V_{LVI}$ 时释放内部复位。
- 如果 LVISEL = 1, 比较外部输入引脚的输入电压 ($EXLVI$) 和检测电压 (V_{EXLVI}), 在 $EXLVI < V_{EXLVI}$ 时产生内部复位信号, 当 $EXLVI \geq V_{EXLVI}$ 时释放内部复位。

备注 低电压检测电路 (LVI) 可由选项字节设置为默认 ON。如果设置为 ON, 供电电压从 POC 检测电压或更低电压开始有效, 当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07\text{ V} \pm 0.2\text{ V}^{\#}$) 时, 会产生内部复位信号。之后, 当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07\text{ V} \pm 0.1\text{ V}^{\#}$) 时, 产生内部复位信号。

(2) 用作中断 (LVIMD = 0)

- 如果 LVISEL = 0, 比较供电电压 (V_{DD}) 与检测电压 (V_{LVI}), 当 V_{DD} 降至比 V_{LVI} 低 ($V_{DD} < V_{LVI}$) 或当 V_{DD} 变为 V_{LVI} 或更高 ($V_{DD} \geq V_{LVI}$) 时, 产生中断信号 (INTLVI)。
- 如果 LVISEL = 1, 比较外部输入引脚的输入电压 ($EXLVI$) 和检测电压 ($V_{EXLVI} = 1.21\text{ V} \pm 0.1\text{ V}^{\#}$), 当 $EXLVI$ 降至比 V_{EXLVI} 低 ($EXLVI < V_{EXLVI}$) 或当 $EXLVI$ 变为 V_{LVI} 或更高 ($EXLVI \geq V_{EXLVI}$) 时, 产生中断信号 (INTLVI)。

注 这只是初步数据, 可能会改变。

当低电压检测电路运行时, 可以通过读取低电压检测标志 (LVIM 的位 0: LVIF) 得知供电电压或者外部输入引脚的输入电压是否高于或低于检测电平。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的位 1。
LVISEL: LVIM 的位 2。

21.4.1 用作复位时

(1) 当检测供电电压 (V_{DD}) 电平时(a) 当设置停止 LVI 默认启动功能 (选项字节: $LVIOFF = 1$) 时

• 启动操作时

- <1> 屏蔽 LVI 中断 ($LVIMK = 1$) .
- <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 (检测供电电压 (V_{DD}) 的电平) (默认值)。
- <3> 使用低电压检测电平选择寄存器 (LVIS) 的第 3 位~第 0 位 (LVIS3 ~ LVIS0) 设置检测电压。
- <4> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。
- <5> 使用软件等待下列几段时间 (总计 $410\mu s$) .
 - 操作稳定时间 ($10\mu s$ (MAX.))
 - 最小脉冲宽度 ($200\mu s$ (MIN.))
 - 检测延迟时间 ($200\mu s$ (MAX.))
- <6> 继续等待, 直至检测到 LVIM 的第 0 位 (LVIF) ($V_{DD} \geq V_{Lvi}$) 。
- <7> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (当检测到设定电平时, 产生复位)。

<R>

图 21-5 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的标号与上面的<1> ~ <7>对应。

注意事项 1. 必须执行过程<1>。当 $LVIMK = 0$ 时, 在过程<4>之后可能会立即产生中断。

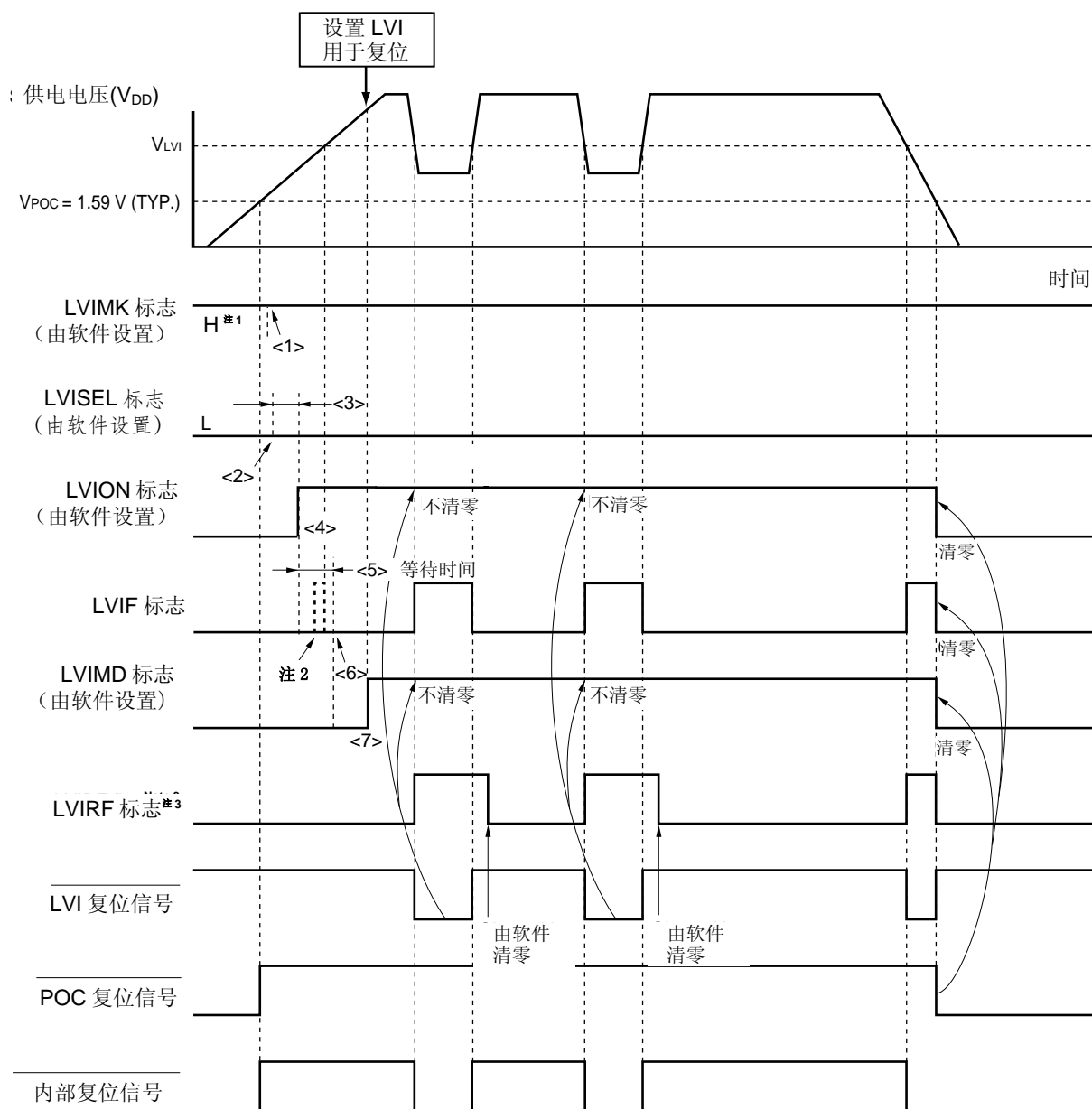
2. 当 $LVIMD=1$ 时, 如果供电电压 (V_{DD}) \geq 检测电压 (V_{Lvi}), 不会产生内部复位信号。

• 当停止操作时

必须执行以下过程的其中之一。

- 当使用 8 位存储器操作指令时:
写入 00H 到 LVIM。
- 当使用 1 位存储器操作指令时:
将 LVIMD 清零, 然后将 LVION 清零。

图 21-5. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 产生的复位信号会将 LVIMK 标志设置为“1”。
 2. 可以设置中断请求标志寄存器的 LVIF 标志和 LVIMD 标志为 (1)。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细内容, 可参见第十九章复位功能

备注 上图 21-5 中的 <1> ~ <7> 对应于 21.4.1 (1) (a) 当设置 LVI 默认启动功能被停止 (选项字节: LVIOFF = 1) 中“启动操作”描述的内容。

(b) 当设置允许 LVI 默认启动功能（选项字节：LVIOFF = 0）时

- 启动操作时

在下列初始设置状态下启动。

- 设置 LVIM 的第 7 位 (LVION) 为 1（允许 LVI 操作）
- 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清 0（检测供电电压 (V_{DD}) 的电平）
- 设置低电压检测电平选择寄存器 (LVIS) 为 0EH（默认数值：V_{LVI} = 2.07 V ± 0.1 V）。
- 设置 LVIM 的第 1 位 (LVIMD) 为 1（当检测到设定电平时产生复位）
- 设置 LVIM 的第 0 位 (LVIF) 为 0（“供电电压 (V_{DD}) ≥ 检测电压 (V_{LVI})”）

<R>

图 21-6 显示了由低电压检测器产生的内部复位信号的时序。

- 当停止操作时

必须执行以下过程的其中之一。

- 当使用 8 位存储器操作指令时：
写入 00H 到 LVIM。
- 当使用 1 位存储器操作指令时：
将 LVIMD 清零，然后将 LVION 清零。

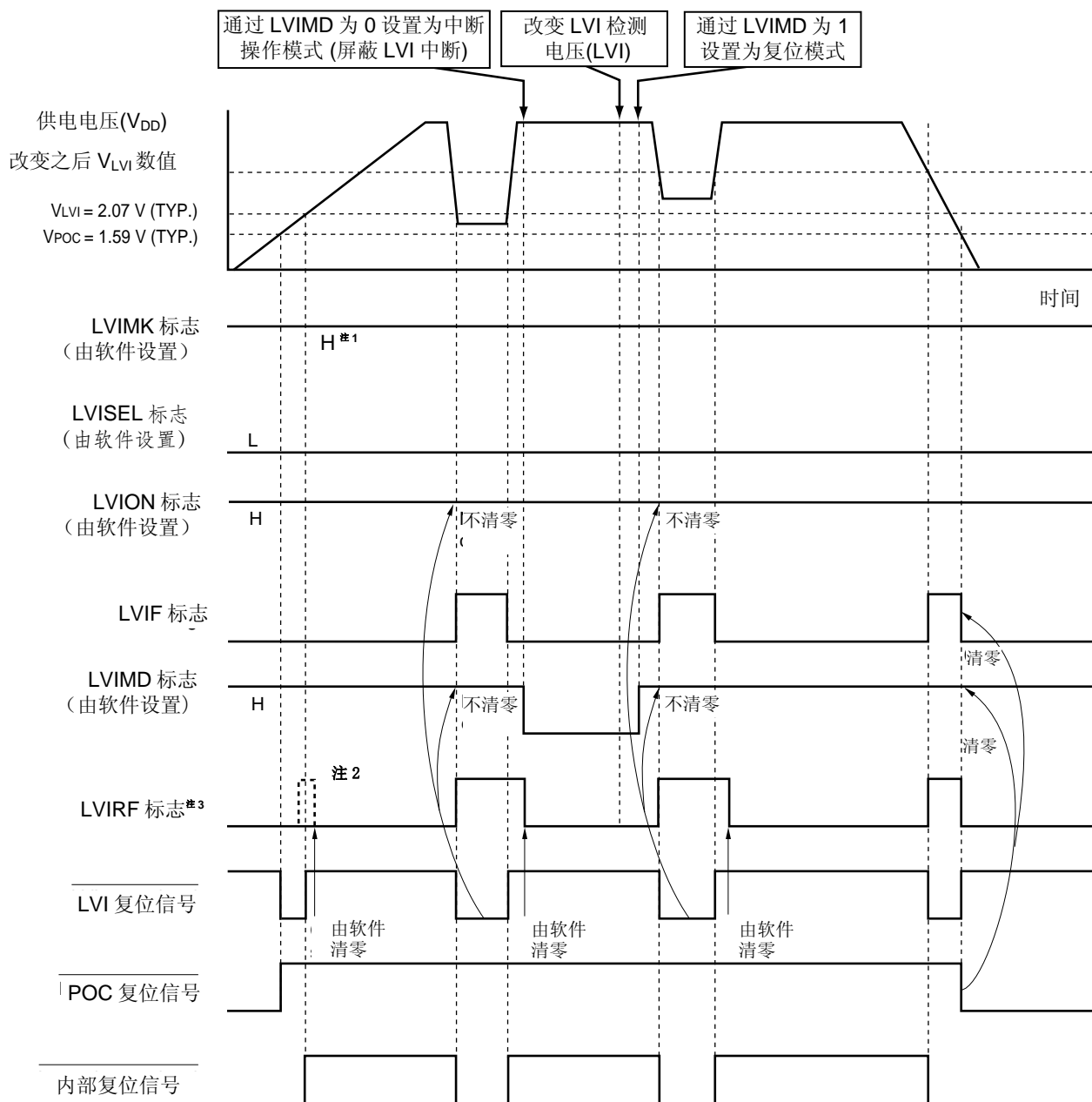
注意事项 即使使用 LVI 默认启动功能时，如果由软件设置 LVI 操作禁止，按照下面所列步骤进行操作：

- 在 LVION = 0 期间，不要进行低电压检测操作。
- 在 LVION = 0 期间，如果产生复位，那么在复位释放后 CPU 开始时，LVION 将被重置为 1。但是，由于 WDT 和执行非法指令引起复位时，将会有一段时间低电压检测功能无法正常操作。
这是因为由 LVI 检测到的脉冲宽度必须小于 200 μ s，在产生复位时，设置 LVION = 1，并且 CPU 无需等待 LVI 的稳定时间即可开始操作。

<R>

<R>

图 21-6. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 0)



注 1. 产生的复位信号会将 LVIMK 标志设置为“1”。

<R>

2. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。

当使用 LVI 的默认启动功能 (000C1H 的第 0 位 (LVIOFF) = 0) 时, LVIRF 标志可能会由于上电波形而从开始就变为 1。

需要了解 RESF 的详细信息, 可参见第十九章 复位功能。

(2) 当检测外部输入引脚 (EXLVI) 的输入电压电平时

• 当启动操作时

<1> 屏蔽 LVI 中断 (LVIMK = 1)。

<2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 1 (检测外部输入引脚 (EXLVI) 的输入电压电平)。

<3> 设置 LVIM 的第 7 位 (LVION) = 1 (允许 LVI 的操作)。

<R>

<4> 使用软件等待下列几段时间 (总计 410 μ s)。

- 操作稳定时间 (10 μ s (MAX.))
- 最小脉冲宽度 (200 μ s (MIN.))
- 检测延迟时间 (200 μ s (MAX.))

<5> 等待直至根据 LVIM 的第 0 位 (LVIF) 检测到 (外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21\text{ V (TYP.)}$))。

<6> 设置 LVIM 的第 1 位 (LVIMD) = 1 (当检测到设定电平时, 产生复位信号)。

图 21-7 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的标号与上面的 <1> ~ <6> 对应。

注意事项 1. 必须执行过程 <1>。当 LVIMK = 0 时, 在过程 <4> 之后可能会立即产生中断。

2. 当 LVIMD=1 时, 如果供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}), 不会产生内部复位信号。

注意事项 1. 必须执行过程 <1>。当 LVIMK = 0 时, 在过程 <3> 之后可能会立即产生中断。

2. 在 LVIMD=1 时, 如果外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21\text{ V (TYP.)}$), 则不会产生内部复位信号。

3. 外部输入引脚的输入电压 (EXLVI) 必须满足的条件: $EXLVI < V_{DD}$ 。

• 当停止操作时

必须执行以下过程的其中之一。

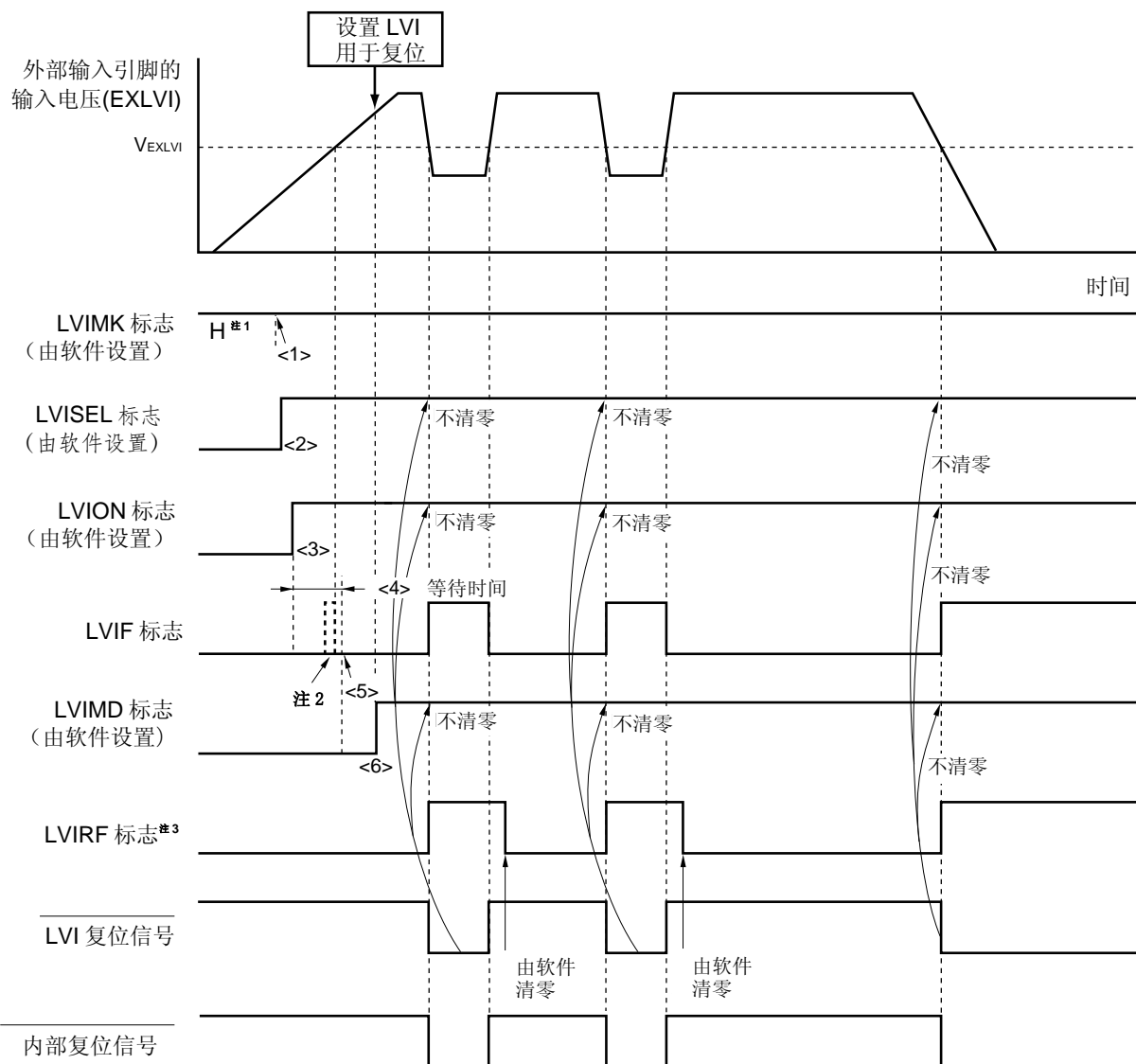
- 当使用 8 位存储器操作指令时:

写入 00H 到 LVIM。

- 当使用 1 位存储器操作指令时:

将 LVIMD 清零, 然后将 LVION 清零。

图 21-7. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 1)



- 注
1. 产生的复位信号会将 LVIMK 标志设置为“1”。
 2. 可以设置中断请求标志寄存器的 LVIIF 标志和 LVIF 标志为 (1)。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细信息, 可参见第十九章复位功能。

<R>

备注 上图 21-7 中的 <1> ~ <6> 对应于 21.4.1 (2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时中“启动操作”描述的内容。

21.4.2 用作中断时

(1) 当检测供电电压 (V_{DD}) 电平时(a) 当设置停止 LVI 默认启动功能 (选项字节: $LVIOFF = 1$) 时

• 当启动操作时

<1> 屏蔽 LVI 中断 ($LVIMK = 1$)。<2> 将低电压检测寄存器 (LVIM) 的第 2 位 ($LVISEL$) 清 (0) (检测供电电压 (V_{DD}) 电平时) (默认值)。把 LVIM 的第 1 位 ($LVIMD$) 清 0 (当检测到设定电平时产生中断信号) (默认值)。

<3> 使用低电压检测电平选择寄存器 (LVIS) 的第 3~0 位设置检测电压。

<4> 设置 LVIM 的第 7 位 ($LVION$) = 1 (允许 LVI 的操作)。<R> <5> 使用软件等待下列几段时间 (总计 $410\mu s$)。• 操作稳定时间 ($10\mu s$ (MAX.))• 最小脉冲宽度 ($200\mu s$ (MIN.))• 检测延迟时间 ($200\mu s$ (MAX.))<6> 当检测 V_{DD} 的下降沿时, 使用 LVIM 的第 0 位 ($LVIF$) 确认 “供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 或者当检测 V_{DD} 的上升沿时, 使用 LVIM 的第 0 位 ($LVIF$) 确认 “供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”。<7> 将 LVI 的中断请求标志 ($LVIIIF$) 清零 (0)。<8> 释放 LVI 的中断屏蔽标志 ($LVIMK$)。

<9> 执行 EI 指令 (当使用向量中断时)。

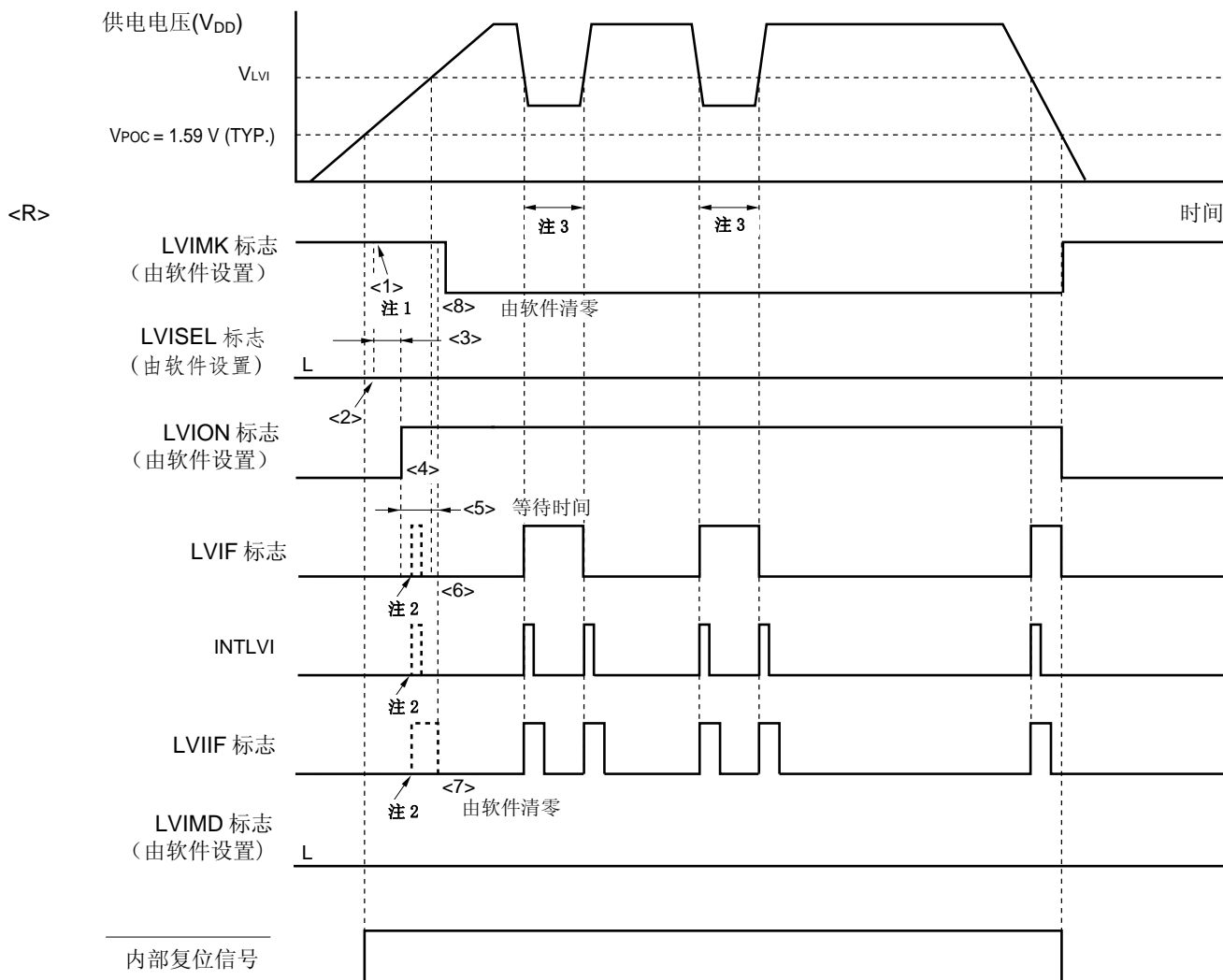
图 21-8 显示了由低电压检测器产生的中断信号的时序。该时序图中的标号与上面的 <1> ~ <8> 对应。

• 当停止操作时

必须执行以下过程的其中之一。

• 当使用 8 位存储器操作指令时:
写入 00H 到 LVIM。• 当使用 1 位存储器操作指令时:
将 LVION 清零。

图 21-8. 低电压检测电路中断信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 产生的复位信号会将 LVIMK 标志设置为“1”。
 2. 产生中断请求信号 (INTLVI)，并且可以设置中断请求标志寄存器的 LVIIIF 标志和 LVIF 标志为 (1)。
 3. 如果当供电电压 (VDD) 低于或者等于检测电压 (VLVI) 时，禁止 LVI 操作，会产生中断请求信号 (INTLVI)，并且 LVIIIF 可能会设置为 1。

备注 上图 21-8 中的<1> ~ <8>对应于 21.4.2 (1) (a) 当设置 LVI 默认启动功能被停止 (选项字节: LVIOFF = 1) 中“启动操作”描述的<1> ~ <8>内容。

(b) 当设置允许 LVI 默认启动功能（选项字节：LVIOFF = 0）时

- 启动操作时

- <R>
- <1> 在下列初始设置状态下开启。
 - 设置 LVIM 的第 7 位 (LVION) 为 1（允许 LVI 操作）
 - 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清 0（检测供电电压的电平 (V_{DD})）
 - 设置低电压检测电平选择寄存器 (LVIS) 为 0EH（默认数值：V_{LVI} = 2.07 V ± 0.1 V）。
 - 设置 LVIM 的第 1 位 (LVIMD) 为 1（当检测到设定电平时产生复位）
 - 设置 LVIM 的第 0 位 (LVIF) 为 0（检测下降沿“供电电压 (V_{DD}) ≥ 检测电压 (V_{LVI})”）
 - <2> 将 LVIM 的第 1 位 (LVIMD) 清零 (0)（当检测到设定电平时产生中断信号）（默认值）。
 - <3> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <4> 执行 EI 指令（当使用向量中断时）。

图 21-9 显示了由低电压检测器产生的中断信号的时序。该时序图中的标号与上面的<1> ~ <3>对应。

- 当停止操作时

必须执行以下过程的其中之一。

- 当使用 8 位存储器操作指令时：
写入 00H 到 LVIM。
- 当使用 1 位存储器操作指令时：
将 LVION 清零 (0)。

注意事项 1. 即使使用 LVI 默认启动功能时，如果由软件设置 LVI 操作禁止，按照下面所列步骤进行操作：

- <R>
- 在 LVION = 0 期间，不要进行低电压检测操作。
 - 在 LVION = 0 期间，如果产生复位，那么在复位释放后 CPU 开始时，LVION 将被重置为 1。但是，由于 WDT 和执行非法指令引起复位时，将会有一段时间低电压检测功能无法正常操作。
这是因为由 LVI 检测到的脉冲宽度必须小于 200μs，在产生复位时，设置 LVION = 1，并且 CPU 无需等待 LVI 的稳定时间即可开始操作。
- <R>
2. 当使用 LVI 的默认启动功能（000C1H 的第 0 位 (LVIOFF) = 0），LVIRF 标志可能会由于上电波形而从开始就变为 1。
需要了解 RESF 的详细信息，可参见第十九章 复位功能。

(2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时

• 当启动操作时

<1> 屏蔽 LVI 中断 (LVIMK = 1)。

<2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为 1 (检测外部输入引脚的输入电压 (EXLVI) 电平)。

将 LVIM 的第 1 位 (LVIMD) 清零 (0) (当检测到设定电平时产生中断信号) (默认值)。

<3> 设置 LVIM 的第 7 位 (LVION) 为 1 (允许 LVI 的操作)。

<R> <4> 使用软件等待下列几段时间 (总计 410 μ s)。

- 操作稳定时间 (10 μ s (MAX.))
- 最小脉冲宽度 (200 μ s (MIN.))
- 检测延迟时间 (200 μ s (MAX.))

<5> 当检测到 EXLVI 的下降沿时, 使用 LVIM 的第 0 位 (LVIF) 确认“外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.))”, 或者当检测 EXLVI 的上升沿时, 使用 LVIM 的第 0 位 (LVIF) 确认“外部输入引脚的输入电压 (EXLVI) $<$ 检测电压 ($V_{EXLVI} = 1.21$ V (TYP.))”。

<6> 将 LVI 的中断请求标志 (LVIF) 清零 (0)。

<7> 释放 LVI 的中断屏蔽标志 (LVIMK)。

<8> 执行 EI 指令 (当使用向量中断时)。

图 21-10 显示了由低电压检测器产生的中断信号的时序。该时序图中的标号与上面的<1> ~ <7>对应。

注意事项 外部输入引脚的输入电压 (EXLVI) 必须满足的条件: $EXLVI < V_{DD}$

• 当停止操作时

必须执行以下过程的其中之一。

• 当使用 8 位存储器操作指令时:

写入 00H 到 LVIM。

• 当使用 1 位存储器操作指令时:

将 LVION 清零 (0)。

21.5 低电压检测电路的注意事项

(1) 当供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 周围频繁波动时采取的方法

在系统中，如果供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。

操作示例 1：用作复位时

系统可能会反复从复位状态进行复位和释放。

在这种情况下，可采用下面的方法任意设置从复位释放到微控制器操作启动所经历的时间。

<方法>

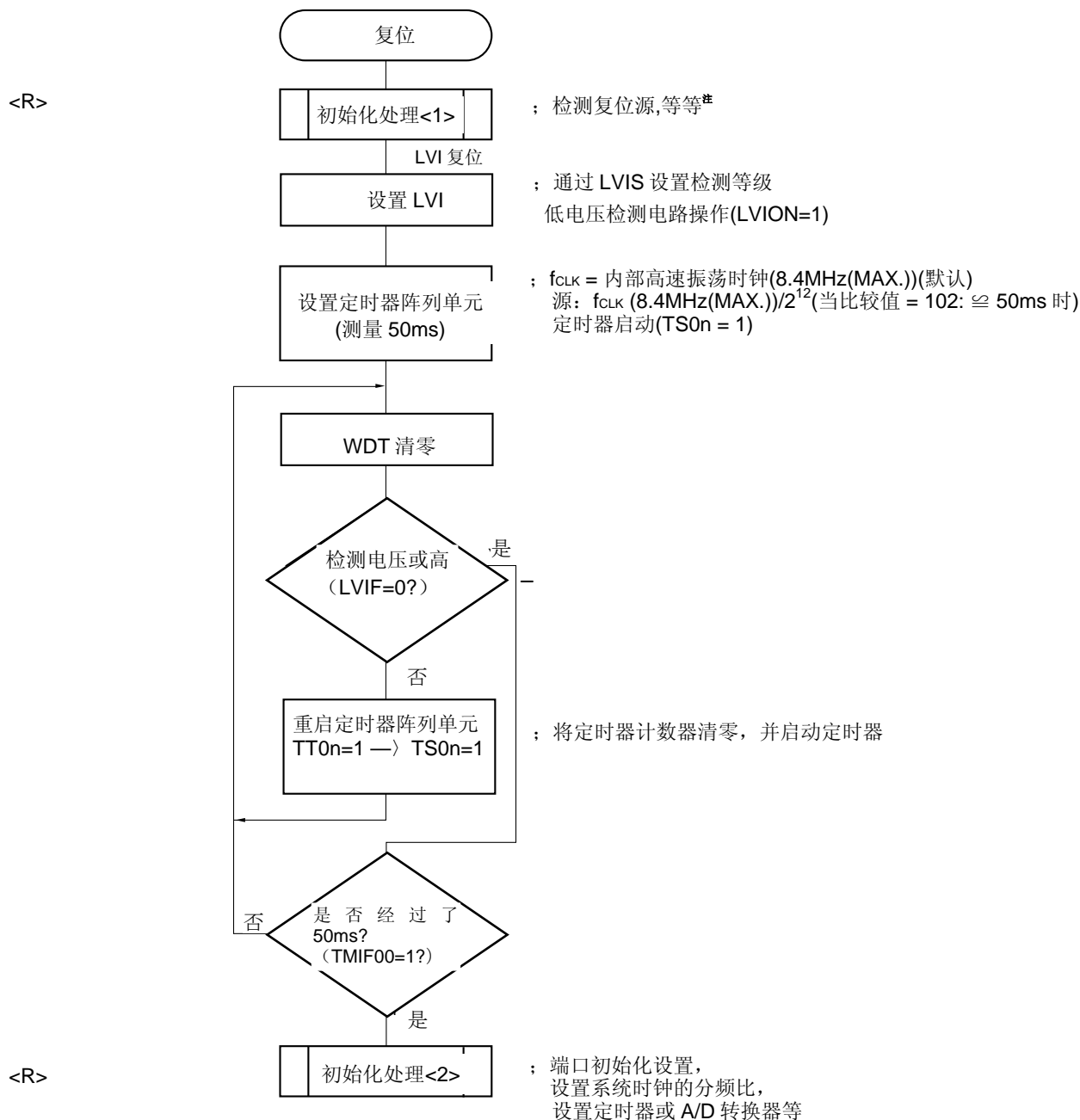
在释放复位信号后，通过软件计数器（使用一个定时器）等待系统的供电电压的波动期，然后对端口初始化（参见图 21-11）。

备注 如果设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为“1”，则对上述方法作如下修改。

- 供电电压 (V_{DD}) → 外部输入引脚的输入电压 (EXLVI)
- 检测电压 (V_{LVI}) → 检测电压 (V_{EXLVI} = 1.21 V)

图 21-11. 复位释放后软件处理过程示例 (1/2)

- 如果供电电压在 LVI 检测电压附近的波动时间小于 50 ms

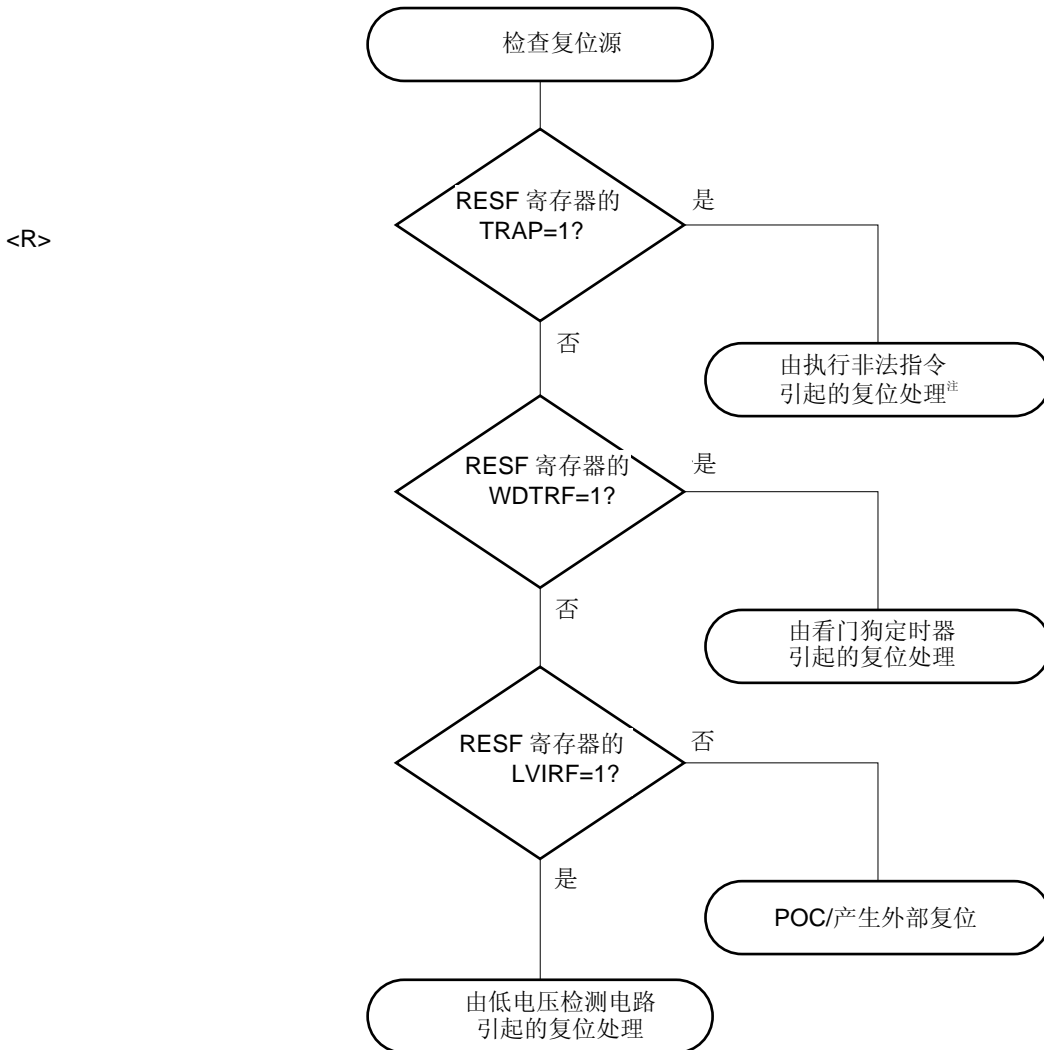


注 流程图在下页显示

- 备注 1. n: 通道编号 (n = 0 ~ 7)
2. 如果设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为“1”, 则对上述方法作如下修改。
- 供电电压 (V_{DD}) → 外部输入引脚的输入电压 (EXLVI)
 - 检测电压 (V_{LVI}) → 检测电压 (V_{EXLVI} = 1.21 V)

图 21-11. 复位释放后软件处理过程示例 (2/2)

- 检查复位源



<R> **注意事项** 当执行指令码 FFH 时。
非法指令执行引起的复位无法由在线仿真器或者片上调试仿真器的仿真发出。

- 备注** 如果设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为“1”，则对上述方法作如下修改。
- 供电电压 (V_{DD}) → 外部输入引脚的输入电压 (EXLVI)
 - 检测电压 (V_{LVI}) → 检测电压 (V_{EXLVI} = 1.21 V)

操作示例 2：用作中断时

系统可能会频繁产生中断。
采取下列方法。

<方法>

在 LVI 中断服务程序中，当检测 V_{DD} 的下降沿时，使用低电压检测寄存器 (LVIM) 的第 0 位 (LVIF) 确认“供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”，或者当检测 V_{DD} 的上升沿时，使用低电压检测寄存器 (LVIM) 的第 0 位 (LVIF) 确认“供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”。将中断请求标志寄存器 0L (IFOL) 的第 1 位 (LVIIF) 清零。

对于供电电压在 LVI 检测电压附近波动时间较长的系统，在等待供电电压波动时间之后采取以上方法。

备注 如果设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 为“1”，则对上述方法作如下修改。

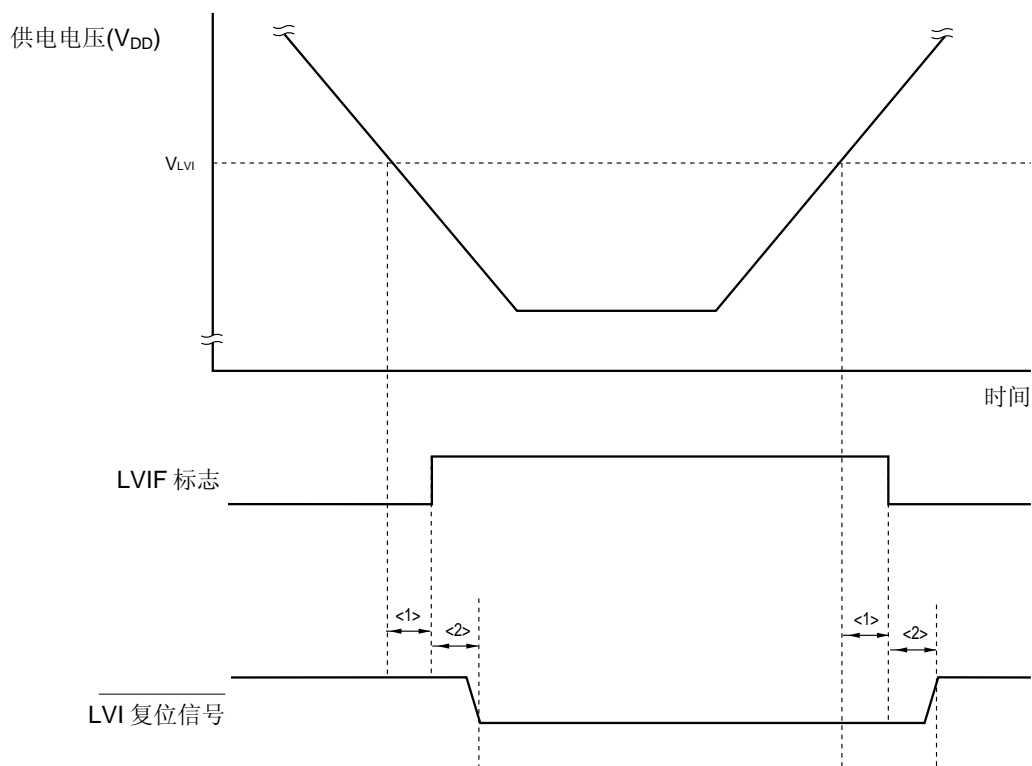
- 供电电压 (V_{DD}) \rightarrow 外部输入引脚的输入电压 ($EXLVI$)
- 检测电压 (V_{LVI}) \rightarrow 检测电压 ($V_{EXLVI} = 1.21 V$)

(2) 从 LVI 复位源产生直到 LVI 复位或释放的延时

从供电电压 (V_{DD}) $<$ LVI 检测电压 (V_{LVI}) 的时刻，直到产生 LVI 复位会有一些延时。

同样，从 LVI 检测电压 (V_{LVI}) \leq 供电电压 (V_{DD}) 的时刻，直到产生 LVI 复位时也会有一些延时（请看图 21-12）。

图 21-12. 从 LVI 复位源产生直到 LVI 复位或释放的延时



- <R> <1> : 最小脉冲宽度 (200 μ s (MIN.))
<2> : 检测延迟时间 (200 μ s (MAX.))

第二十二章 稳压器

22.1 稳压器概述

<R> 78K0R/KF3 包含一个在恒电压下操作该设备的电路。此时，为了稳定稳压器输出电压，通过一个电容（0.47 ~ 1 μF ）将 REGC 引脚连接到 V_{SS} 。但是，在已经进入 STOP 模式时，因为内部高速振荡时钟和外部主系统时钟的操作，推荐使用 0.47 μF 的电容。同时，使用具有良好特性的电容，因为要用它来稳定内部电压。
稳压器的输出电压正常为 2.5 V (TYP.)，在低功耗电流模式下为 1.8 V (TYP.)。

22.2 控制稳压器的寄存器

(1) 稳压器模式控制寄存器 (RMC)

此寄存器设置稳压器的输出电压。

RMC 可以通过 8 位存储器操作指令设置。

复位输入会将此寄存器设置为 00H。

图 22-1. 稳压器模式控制寄存器 (RMC) 的格式

地址: F00F4H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RMC								
RMC[7: 0]	控制稳压器的输出电压							
5AH	固定为低功耗电流模式 (1.8 V)							
00H	根据条件 (参考表 22-1) 在正常模式 (2.5 V) 和低功耗电流模式 (1.8 V) 之间切换							
其它	禁止设置							

- 注意事项
- 只有在低功耗电流模式 (参考表 22-1) 下才可以写入 RMC 寄存器。也就是说，在高速系统时钟 (f_{MX}) 和高速内部振荡时钟 (f_{IH}) 两个都停止时，CPU 使用副系统时钟 (f_{XT}) 期间，才可以重写该寄存器。
 - 当设置固定为低电流损耗模式时，RMC 寄存器在下列情况下可以使用。
 < CPU 使用 X1 时钟时 > $f_{\text{X}} \leq 5 \text{ MHz}$ 并且 $f_{\text{CLK}} \leq 5 \text{ MHz}$
 < 当 CPU 使用高速内部振荡时钟，外部输入时钟或者副系统时钟时 > $f_{\text{CLK}} \leq 5 \text{ MHz}$
 - 在低功耗电流模式下，自编程序禁止。

<R>

表 22-1. 稳压器输出电压条件

模式	输出电压	条件
低功耗电流模式	1.8 V	系统复位期间
		STOP 模式下 (OCD 模式期间除外)
		当高速系统时钟 (f _{MX}) 和 高速内部振荡时钟 (f _{IH}) 两个都停止, CPU 操作使用副系统时钟 (f _{XT})
		当高速系统时钟 (f _{MX}) 和 高速内部振荡时钟 (f _{IH}) 两个在 HALT 模式下都停止时, 设置 CPU 使用副系统时钟 (f _{XT})
正常电流模式	2.5 V	上述之外的其它情况下

第二十三章 选项字节

23.1 选项字节的功能

78K0R/KF3 的 flash 存储器的地址 000C0H~000C3H 组成选项字节区域。

选项字节由用户选项字节（000C0H~000C2H）和片上调试选项字节（000C3H）组成。

当打开电源或从复位状态重启设备时，设备自动引用选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

在自编程期间使用引导交换功能时，将 000C0H~000C3H 切换到 010C0H ~010C3H。因此，事先将 000C0H ~000C3H 和 010C0H ~010C3H 设置为相同的值。

注意事项 000C2H 必须设置为 FFH（当使用引导交换操作时，000C2H/010C2H）。

23.1.1 用户选项字节（000C0H~ 000C2H/010C0H ~010C2H）

（1）000C0H/010C0H

- 看门狗定时器的操作
 - 在 HALT 或 STOP 模式下，操作停止或允许。
- 看门狗定时器的间隔时间的设置
- 看门狗定时器的操作
 - 操作停止或允许。
- 看门狗定时器的窗口打开比例的设置
- 看门狗定时器的间隔中断的设置
 - 使用或不使用

注意事项 当使用引导交换操作时，设置 000C0H 与 010C0H 的值相同，因为 000C0H 要被 010C0H 替换。

（2）000C1H/010C1H

- <R>
- 复位释放时的 LVI 设置（上电时）
 - 在复位释放时 LVI 默认打开或默认关闭（由 $\overline{\text{RESET}}$ 引脚复位，而不是由 LVI, POC, WDT, 或非法指令复位时）。

注意事项 当使用引导交换操作时，设置 000C1H 与 010C1H 的值相同，因为 000C1H 要被 010C1H 替换。

（3）000C2H/010C2H

- 必须设置 FFH，因为这些地址为保留区域。

注意事项 当使用引导交换操作时，010C2H 设置为 FFH，因为 000C2H 要被 010C2H 替换。

23.1.2 片上调试选项字节 (000C3H/ 010C3H)

- 片上调试操作的控制
 - 片上调试操作禁止或允许。
- 当片上调试安全 ID 验证失败时, flash 存储器内的数据处理
 - 如果允许片上调试, 在验证片上调试安全 ID 失败后, 擦除或不擦除 flash 存储器的数据。

注意事项 当使用引导交换功能时, 设置 000C3H 和 010C3H 的值相同, 因为 000C3H 要被 010C3H 替换。

23.2 用户选项字节的格式

用户选项字节的格式如下所示。

图 23-1. 用户选项字节 (000C0H/010C0H) 的格式 (1/2)

地址: 000C0H/010C0H^{#1}

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
看门狗定时器的间隔中断的使用							
0	不使用间隔中断。						
1	当达到溢出时间的 75%时, 产生间隔中断。						
看门狗定时器窗口打开比例 ^{#2}							
WINDOW1	WINDOW0						
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
看门狗定时器计数器的操作控制							
WDTON							
0	计数器操作禁止 (复位后计数停止)						
1	计数器操作允许 (复位后计数开始)						
看门狗定时器溢出时间							
WDCS2	WDCS1	WDCS0					
0	0	0	2 ¹⁰ /f _{IL} (3.88 ms)				
0	0	1	2 ¹¹ /f _{IL} (7.76 ms)				
0	1	0	2 ¹² /f _{IL} (15.52 ms)				
0	1	1	2 ¹³ /f _{IL} (31.03 ms)				
1	0	0	2 ¹⁵ /f _{IL} (124.12 ms)				
1	0	1	2 ¹⁷ /f _{IL} (496.48 ms)				
1	1	0	2 ¹⁸ /f _{IL} (992.97 ms)				
1	1	1	2 ²⁰ /f _{IL} (3971.88 ms)				

图 23-1. 用户选项字节（000C0H/010C0H）的格式（2/2）

地址：000C0H/010C0H^{#1}

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDSTBYON	看门狗定时器 计数器的操作控制（HALT/STOP 模式）						
0	在 HALT/STOP 模式 ^{#2} 下停止计数器操作						
1	在 HALT/STOP 模式下允许计数器操作						

- 注 1. 由于在引导交换时 010C0H 要替换 000C0H，因此 000C0H 的值应该和 010C0H 设置相同的值。
 2. 当 WDSTBYON = 0 时，不论 WINDOW1 和 WINDOW0 为何值，窗口打开比例是 100%。

注意事项 看门狗定时器在 flash 存储器自编程和 EEPROM 模拟期间继续它的操作。处理期间，中断响应时间会延长。设置溢出时间和窗口大小时应考虑到这一延迟情况。

- 备注 1. f_{IL} ：内部低速振荡时钟频率
 2. ()： $f_{IL} = 264 \text{ kHz (MAX.)}$

图 23-2. 选项字节（000C1H/010C1H）的格式

地址：000C1H/010C1H[#]

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	LVI OFF
LVI OFF	上电时 LVI 的设置						
0	复位释放（上电时）时，LVI 默认打开（允许 LVI 默认启动功能）						
1	复位释放（上电时）时，LVI 默认关闭（停止 LVI 默认启动功能）						

<R>
 <R>

注 由于在引导交换时 010C1H 要替换 000C1H，因此 000C1H 的值应该和 010C1H 设置相同的值。

- 注意事项 1. 确保第 7~1 位为“1”。
 2. 即使使用 LVI 默认启动功能时，如果通过软件设置 LVI 操作禁止，则操作如下：
- 在 LVION = 0 期间，不要进行低电压检测操作。
 - 在 LVION = 0 期间，如果产生复位，那么在复位释放后 CPU 开始时，LVION 将被重置为 1。但是，如果是由于 WDT 和执行非法指令引起复位时，将会有一段时间低电压检测功能无法正常操作。
- 这是因为由 LVI 检测到的脉冲宽度必须小于 200 μs ，在产生复位时，设置 LVION = 1，并且 CPU 无需等待 LVI 的稳定时间即可开始操作。

<R>

图 23-3. 选项字节（000C2H/010C2H）的格式

地址：000C2H/010C2H[#]

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

注 必须将 000C2H 设置为 FFH，因为这些地址是保留区域。当使用引导交换操作时，也需要将 010C2H 设置为 FFH，因为 000C2H 要被 010C2H 替换。

23.3 片上调试选项字节的格式

片上调试选项字节的格式如下所示。

图 23-4. 片上调试选项字节（000C3H/010C3H）的格式

地址：000C3H/010C3H[#]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	片上调试操作的控制
0	0	禁止片上调试操作。
0	1	禁止设置。
1	0	如果允许片上调试，在验证片上调试安全 ID 失败后，擦除 flash 存储器的数据。
1	1	如果允许片上调试，在验证片上调试安全 ID 失败后，不擦除 flash 存储器的数据。

注 由于在引导交换时 010C3H 要替换 000C3H，因此要对 000C3H 设置与 010C3H 相同的值。

注意事项 第 7 和 0 位（OCDENSET 和 OCDERSD）只能被指定一个值。
 确保 000010B 的第 6 位设置为 1。

备注 当使用片上调试功能时，第 3 ~1 位的值被重写，因此设置后它的值不确定。
 但是，设置时必须设置默认值（0, 1, 0）到第 3 ~1 位。

23.4 选项字节的设置

通过汇编程序包 RA78K0R 的连接器选项也可以设置用户选项字节和片上调试选项字节。
 如何设置选项字节，参考 RA78K0R 汇编程序包用户手册。

备注 复位处理期间引用选项字节。复位过程的时序，参见第十九章 复位功能。

第二十四章 FLASH存储器

78K0R/KF3 内含 Flash 存储器，可以安装在电路板上实现写入、擦除和覆盖操作。

24.1 用Flash存储器编程器写入

使用专用 Flash 存储器编程器，可以将数据以 on-board 或 off-board 方式写入到 flash 存储器。

(1) On-board 编程

当 78K0R/KF3 系列产品安装到目标系统上以后，Flash 存储器的内容可以被重写。在目标系统中必须安装连接专用 Flash 编程器的连接器。

(2) Off-board 编程

在 78K0R/KF3 系列产品安装到目标系统之前，可使用专用程序适配器（FA 系列）将数据写入 Flash 存储器。

备注 FA 系列是 Naito Denssei Machida Mfg. Co., Ltd 的产品。

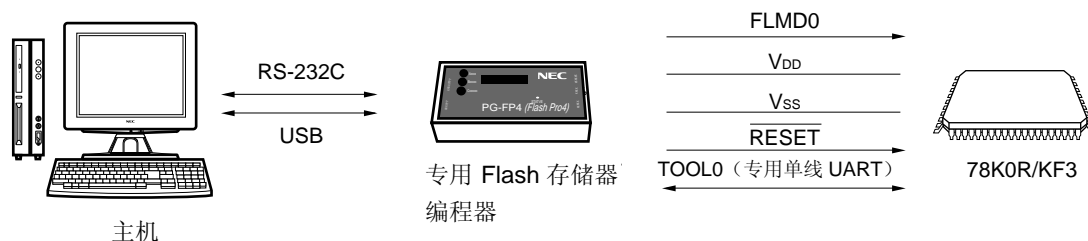
表 24-1. 78K0R/KF3 和专用 Flash 存储器编程器之间的连线

专用 Flash 存储器编程器的引脚配置			引脚名称	引脚编号
信号名称	I/O	引脚功能		
SI/RxD	输入	接收信号	TOOL0/ P40	9
SO/TxD	输出	发送信号		
SCK	输出	发送时钟	-	-
CLK	输出	时钟输出	-	-
/RESET	输出	复位信号	$\overline{\text{RESET}}$	10
FLMD0	输出	模式信号	FLMD0	13
V _{DD}	I/O	V _{DD} 电压产生/ 电压监测	V _{DD}	19
			EV _{DD}	20
			AV _{REF0}	59
			AV _{REF1}	56
GND	-	地	V _{SS}	17
			EV _{SS}	18
			AV _{SS}	60

24.2 编程环境

以下是 78K0R/KF3 Flash 存储器写入所需的编程环境。

图 24-2. Flash 存储器编程环境



需要有一个主机用来控制专用 Flash 编程器。

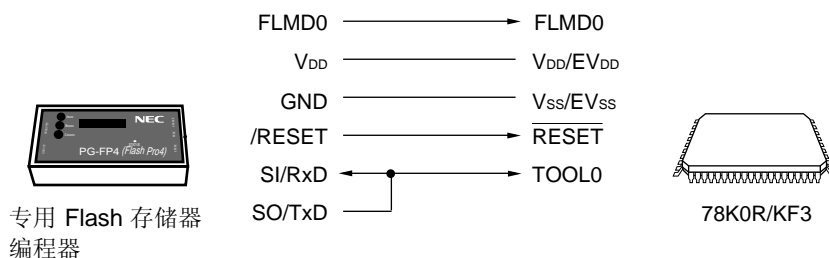
专用 Flash 存储器编程器与 78K0R/KF3 的接口，在 TOOL0 引脚通过专用单线 UART 进行写入和擦除等操作。Flash 存储器 off-board 写操作时必须使用一个专用编程适配器（FA 系列）。

24.3 通信模式

专用 Flash 存储器编程器和 78K0R/KF3 之间的通信由串行通信连接，使用 TOOL0 引脚通过 78K0R/KF3 的专用单线 UART 进行通讯。

发送速率：115,200 bps ~ 1,000,000 bps

图 24-3. 与专用 Flash 存储器编程器通信



当使用专用 Flash 存储器编程器 FlashPro4 时, FlashPro4 为 78K0R/KF3 产生如下信号。详细情况, 参见 FlashPro4 的用户手册。

表 24-2. 引脚连接

FlashPro4			78K0R/KF3	连接
信号名称	I/O	引脚功能	引脚名称	
FLMD0	输出	模式信号	FLMD0	⊙
V _{DD}	I/O	V _{DD} 电压产生/电压监测	V _{DD} , EV _{DD} , AV _{REF0} , AV _{REF1}	⊙
GND	-	地	V _{SS} , EV _{SS} , AV _{SS}	⊙
CLK	输出	时钟输出	-	×
/RESET	输出	复位信号	$\overline{\text{RESET}}$	⊙
SI/RxD	输入	接收信号	TOOL0	⊙
SO/TxD	输出	发送信号		
SCK	输出	发送时钟	-	×

备注 ⊙: 必须连接引脚。
 ×: 引脚不必连接。

24.4 On Board方式的引脚连接

对 Flash 存储器进行 on-board 写入操作时, 目标系统必须有连接专用 Flash 编程器的连接器。电路板上首先要能够提供选择功能, 可以选择正常操作模式或 Flash 存储器编程模式。

当设置 Flash 存储器编程模式时, 那些未用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备在复位后不能立即识别该状态, 则必须对引脚的连接方式进行如下处理。

<R> 24.4.1 FLMD0 引脚

(1) 在 flash 存储器编程模式

当专用 Flash 存储器编程器进行数据写入时, FLMD0 引脚直接连接到专用 Flash 存储器编程器, 这样就可以给 FLMD0 引脚提供 V_{DD} 的写入电压。

在 Flash 存储器编程模式下, FLMD0 引脚不需要外部下拉, 因为它已经通过复位内部下拉。需要外部下拉时, 使用 1 k Ω ~ 200 k Ω 的电阻。

(2) 普通操作模式

在普通操作模式下, 该引脚推荐开路悬空。

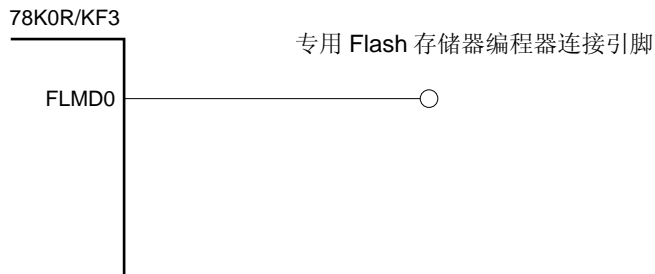
在复位释放前, FLMD0 引脚必须一直保持 V_{SS}, 但是它不需要外部下拉, 因为它已经通过复位内部下拉。但是下拉必须使用背景事件控制寄存器 (BECTL) 的第 7 位保持选中 (参见 24.5 (1) 背景事件控制寄存器)。要使用外部下拉, 使用 200 k Ω 以内的电阻。

使用硬件可以禁止自编程, 也可以禁止使用编程器重写 flash 存储器, 具体做法就是将该引脚直接连接到 V_{SS} 引脚。

(3) 自编程模式

在普通操作模式下，该引脚推荐开路悬空。要进行外部下拉，请使用 100kΩ ~200kΩ 的电阻。在自编程模式下，在自编程库中设置切换到上拉。

图 24-4. FLMD0 引脚连接示例



24.4.2 TOOL0 引脚

在 Flash 存储器编程模式下，该引脚直接连接到专用 Flash 存储器编程器，或通过外部电阻连接 EV_{DD} 进行上拉。

当在普通操作模式下允许片上调试功能时，通过外部电阻连接到 EV_{DD} 对该引脚进行上拉，在复位释放前，确保输入 V_{DD} 到 TOOL0 引脚（禁止上拉此引脚）。

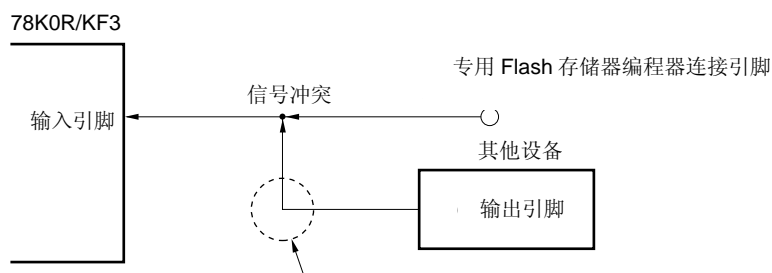
备注 SAU 和 IIC0 引脚不能用于 78K0R/KF3 和专用 Flash 存储器编程器之间的通信，因为使用单线 UART。

24.4.3 RESET 引脚

如果将专用 Flash 编程器的复位信号连接到 RESET 引脚，且该引脚已经连接到板上的复位信号发生器，则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 编程器的复位信号外，不要输入其它信号。

图 24-5. 信号冲突 (RESET 引脚)



在 Flash 存储器编程模式中，另一个设备输出的信号与专用 Flash 存储器编程器的输出信号发生冲突。因此必须隔离另一个设备的信号。

24.4.4 端口引脚

在 Flash 存储器编程模式下，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此，如果连接到端口的设备在复位后不能立即识别端口状态，则必须通过一个电阻将此端口引脚连接到 V_{DD} 或 V_{SS}。

24.4.5 REGC引脚

<R> 与普通操作期间相同，通过一个电容（0.47~ 1 μ F）将 REGC 引脚连接到 GND。然而，在 STOP 模式中，因为内部高速振荡时钟和外部主系统始终的操作，建议为 0.47 μ F。使用特性良好的电容，因为它用于稳定内部电压。

24.4.6 X1 和 X2 引脚

X1 和 X2 的连接状态和普通操作模式相同。

备注 在 Flash 存储器编程模式下，使用内部高速振荡时钟（f_{IH}）。

24.4.7 供电电压

要使用 Flash 编程器输出的供电电压，则将 V_{DD} 引脚与 Flash 编程器的 V_{DD} 相连，将 V_{SS} 引脚与 Flash 编程器的 GND 相连。

要使用板上供电电压，要按正常操作模式连接。

然而，在使用板上供电电压时，如果要使用 Flash 编程器进行电压监视功能，必须将 V_{DD} 和 V_{SS} 引脚分别与 Flash 编程器的 V_{DD} 和 GND 相连接。

其它的供电电压（EV_{DD}，EV_{SS}，AV_{REF0}，AV_{REF1}，和 AV_{SS}）与正常操作模式中的相同。

24.5 控制Flash 存储器的寄存器

(1) 背景事件控制寄存器（BECTL）

即使 FLMD0 引脚不由外部控制，也可以通过 BECTL 寄存器由软件控制设置为自编程模式。

但是，根据 FLMD0 引脚的处理情况，可能无法由软件设置到自编程模式。使用 BECTL 时，建议 FLMD0 引脚悬空开路。当通过外部将其下拉时，使用大于 100 k Ω 的电阻。另外，在普通操作模式下，使用 BECTL 下拉选择。在自编程模式下，在自编程库中设置被切换为上拉。

BECTL 寄存器可以通过 1 位或 8 位存储器操作指令设置。

复位输入设置此寄存器为 00H。

图 24-6. 背景事件控制寄存器（BECTL）的格式

地址：FFFBEH 复位后：00H R/W

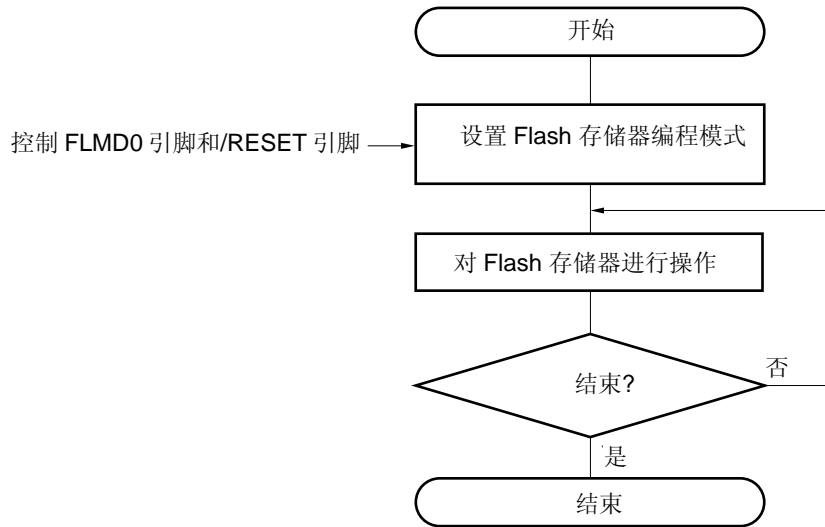
符号	7	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0
FLMDPUP	FLMD0引脚的软件控制							
0	选择下拉							
1	选择上拉							

24.6 编程方法

24.6.1 控制flash 存储器

下图显示了 Flash 存储器操作过程。

图 24-7. Flash 存储器操作过程



24.6.2 Flash 存储器编程模式

要使用专用 Flash 存储器编程器重写 flash 存储器的内容，设置 78K0R/KF3 为 Flash 存储器编程模式。而要设置该模式，必须将 FLMD0 引脚和 TOOL0 引脚连接到 V_{DD}，并对复位信号清零。

当 on-board 写入 Flash 存储器时，使用跳线改变模式。

图 24-8. Flash 存储器编程模式

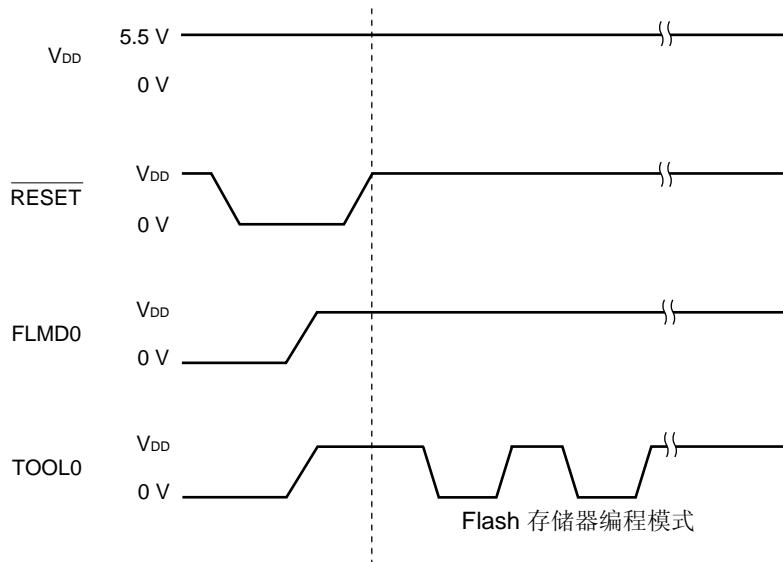


表 24-3. FLMD0 引脚和复位释放后的操作模式之间的关系

FLMD0	操作模式
0	普通操作模式
V _{DD}	Flash 存储器编程模式

24.6.3 选择通信模式

78K0R/KF3 的通信模式如下。

表 24-4. 通信模式

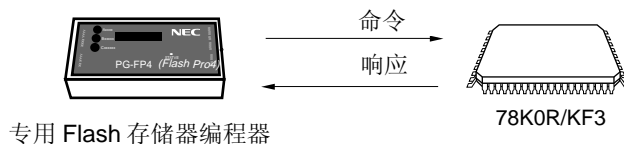
通信模式	标准设置 ^{注1}				使用的引脚
	端口	速度	频率	倍率	
单线模式 (专用单线 UART)	UART-ch0	1 Mbps ^{注2}	—	—	TOOL0

- 注
1. Flash 存储器编程器的 GUI 上的选择标准设置的选项。
 2. 因为波特率误差以外的因素，如信号波形瞬变，也会影响 UART 通信，因此必须象测量波特率误差一样严格测量瞬变。

24.6.4 通信命令

78K0R/KF3 使用命令与专用 Flash 存储器编程器进行通信。从 Flash 存储器编程器发往 78K0R/KF3 的信号称为命令，从 78K0R/KF3 发往专用 Flash 存储器编程器的命令称为响应。

图 24-9. 通信命令



78K0R/KF3 的 Flash 存储器控制命令列表如下。所有这些命令都由编程器发出，78K0R/KF3 根据接收到的命令进行相应的处理。

表 24-5. Flash 存储器控制命令

类别	命令名称	功能
校验	校验	将 flash 存储器指定区域的内容和由编程器发送的数据比较。
擦除	芯片擦除	擦除全部 flash 存储器内容。
	块 (Block) 擦除	擦除 flash 存储器的指定区域。
空白检测	块 (Block) 空白检测	检查 flash 存储器指定块的擦除状态
写入	编程	在 Flash 存储器的指定区域写入数据。
获取信息	硅标记特征	获取 78K0R/KF3 信息 (比如产品型号和 flash 存储器配置)。
	版本获得	获取 78K0R/KF3 固件版本。
	校验和	获得指定区域的校验和数据。
安全	安全设置	设置安全信息。
其他	复位	用于检查通信的同步状态。
	波特率设置	当选择 UART 通信模式时, 设置波特率。

78K0R/KF3 根据专用 Flash 编程器发出的命令返回一个响应。78K0R/KF3 发出的响应列表如下。

表 24-6. 响应名称

响应名称	功能
ACK	响应命令/数据
NAK	响应非法命令/数据

24.7 安全设置

78K0R/KF3 支持安全性功能，该功能禁止修改已经写入 flash 存储器内的用户程序，所以没有授权的话，无法改变程序。

使用安全性设置命令可以执行如下操作。在下次设置编程模式后，安全性设置有效。

- 禁止批量擦除（片擦除）

在 on-board/off-board 编程期间，该项设置 flash 存储器的所有块禁止执行块擦除和批量擦除（片擦除）命令。一旦禁止执行批量擦除（片擦除）命令，所有的禁止设置项（包括禁止批量擦除（片擦除））就不再可以被取消。

注意事项 在对批量擦除进行安全性设置后，不能再对该设备执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写入命令，如果写入的数据和已经存在于 Flash 存储器中的数据不同，则不能写入。

- 禁止块（block）擦除

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器中的指定块（block）进行擦除。但可通过自编程进行块（block）擦除。

- 禁止写入

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器所有块（block）进行写入和块（block）擦除。但可通过自编程进行块（block）擦除。

- 禁止重写引导簇 0

该项设置禁止对 Flash 存储器中的引导簇 0 进行批量擦除（片擦除）、块（block）擦除和写操作。

Flash 存储器在默认出厂设置下，允许使用批量擦除（片擦除）、块（block）擦除、写入和重写引导簇 0 命令。以上安全性设置可以由 on-board/off-board 编程和自编程进行设置。各项安全性设置可以结合使用。

通过执行批量擦除（片擦除）命令，可以取消禁止擦除块（block）和写命令。

表 24-7 显示了当允许使用 78K0R/KF3 安全性功能时，擦除命令和写入命令之间的关系。

备注 要在自编程期间禁止写入和擦除，使用 flash 关闭窗口功能（详细参见 24.8.2）。

表 24-7. 允许安全性功能和命令之间的关系

(1) 在 on-board/off-board 编程期间

有效的安全性设置	执行的命令		
	批量擦除 (片擦除)	Block 擦除	写入
禁止批量擦除 (片擦除)	不能批量擦除	块不能被擦除。	可以执行 ^注
禁止块 (block) 擦除	可以批量擦除		可以执行
禁止写入			不能执行
禁止重写引导簇 0	不能批量擦除	引导簇 0 不能被擦除。	引导簇 0 不能被写入

注 确认没有数据写入待写区域。因为禁止批量擦除 (片擦除) 后, 数据不能被擦除, 所以还有数据没被擦除, 不要写入数据。

(2) 自编程期间

有效的安全性设置	执行命令	
	块擦除	写入
禁止批量擦除 (片擦除)	块可以被擦除。	可以执行。
禁止块 (block) 擦除		
禁止写入		
禁止重写引导簇 0	引导簇 0 不能擦除。	引导簇 0 不能被写入。

备注 要在自编程期间禁止写入和擦除, 使用 flash 封闭窗口功能 (详细参见 24.8.2)。

表 24-8. 在每个编程模式下设置安全性

(1) On-board/off-board 编程

安全	安全性设置	如何禁止安全性设置
禁止批量擦除 (片擦除)	通过专用 Flash 存储器编程器的 GUI 设置, 等。	设置后不能禁止。
禁止块 (block) 擦除		执行批量擦除 (片擦除) 命令。
禁止写入		
禁止重写引导簇 0		设置后不能禁止。

(2) 自编程

安全	安全设置	如何禁止安全设置
禁止批量擦除 (片擦除)	通过信息库设置。	设置后不能禁止。
禁止块 (block) 擦除		在 on-board/off-board 编程期间执行批量擦除 (片擦除) 命令。(在自编程期间不能禁止)
禁止写入		
禁止重写引导簇 0		

24.8 通过自编程进行Flash存储器编程

78K0R/KF3 支持自编程功能，即可以通过用户程序重写 Flash 存储器。由于该功能允许用户应用程序使用 78K0R/KF3 自编程库重写 flash 存储器，因此可用于对程序现场升级。

如果在自编程期间产生一个中断，可以暂时停止自编程操作，去执行中断服务程序。如果在 EI 状态产生可屏蔽中断请求，该请求的跳转直接从自编程库到中断服务程序。在恢复自编程模式后，可以继续自编程操作。但是，中断响应的时间与普通模式下的响应时间不同。

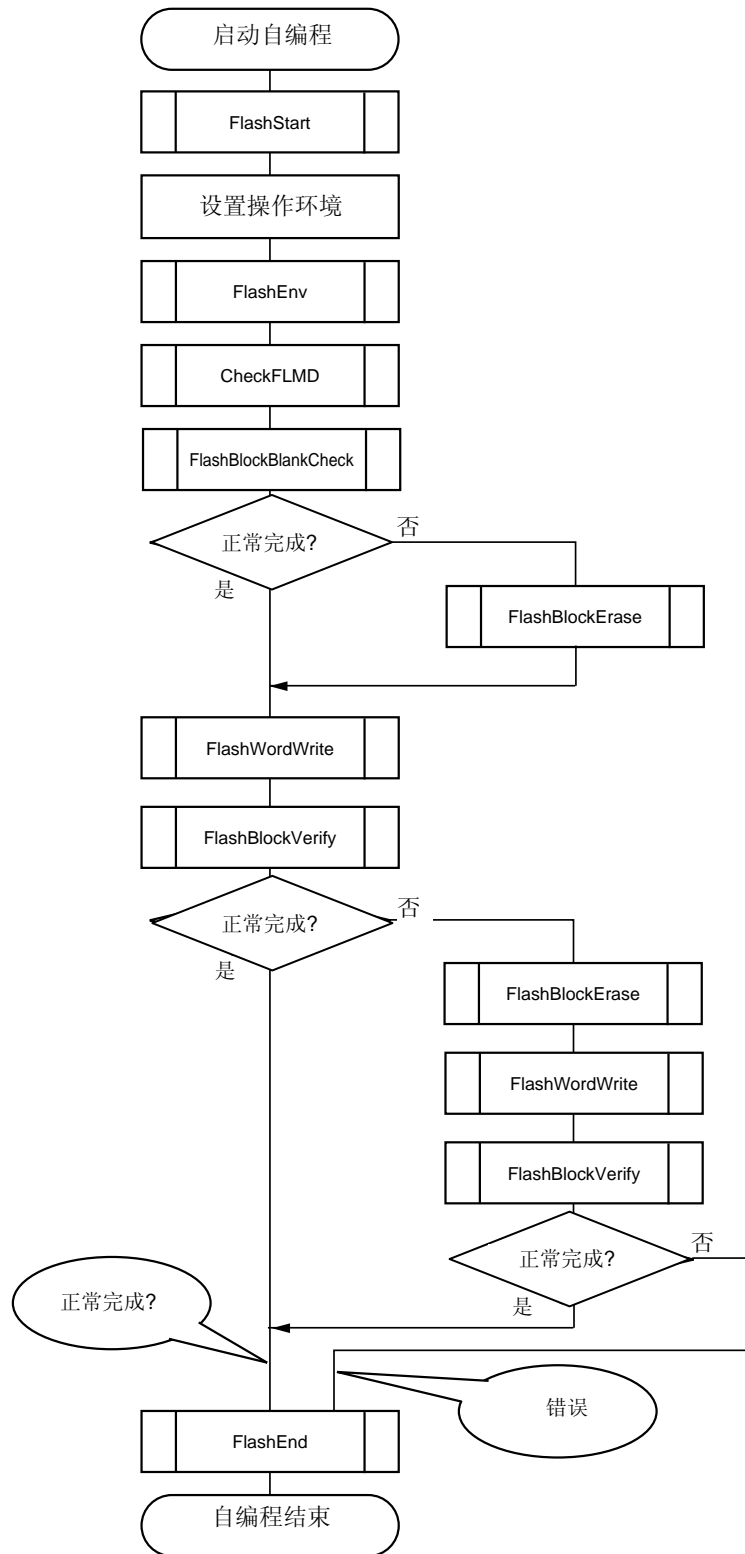
<R> **备注** 如需了解自编程功能和 78K0R/KF3 自编程库的详细信息，请参见 **78K0R 微控制器自编程库类型 01 用户手册 (U18706E)**。

- 注意事项**
1. 当 CPU 使用副系统时钟时，不能使用自编程功能。
 2. 在自编程模式下，调用自编程启动库 (FlashStart)。
 3. 在自编程期间要禁止一个中断，与在普通操作模式下的禁止方法相同，通过执行 DI 指令将 IE 标志清为 (0) 的状态下，执行自编程库。要允许某个中断，通过 EI 指令设置 IE 标志为 (1) 的状态下，将对应的中断屏蔽标志清 (0)，然后执行自编程库。
 4. 在低功耗电流模式下，禁止自编程功能。低功耗电流模式的详细内容，参见第二十二章稳压器。

<R>

下图所示为通过使用自编程库重写 flash 存储器的流程图。

图 24-10. 自编程的流程图（重写 Flash 存储器）



<R> 备注 如需了解自编程库的详细信息，可参考 **78K0R 微控制器自编程库类型 01 用户手册 (U18706E)**。

24.8.1 引导交换功能

如果在自编程期间由于电源失效或其它原因导致重写引导区域的操作失败，则引导区域的数据损坏，不能通过复位或重写来重启程序。

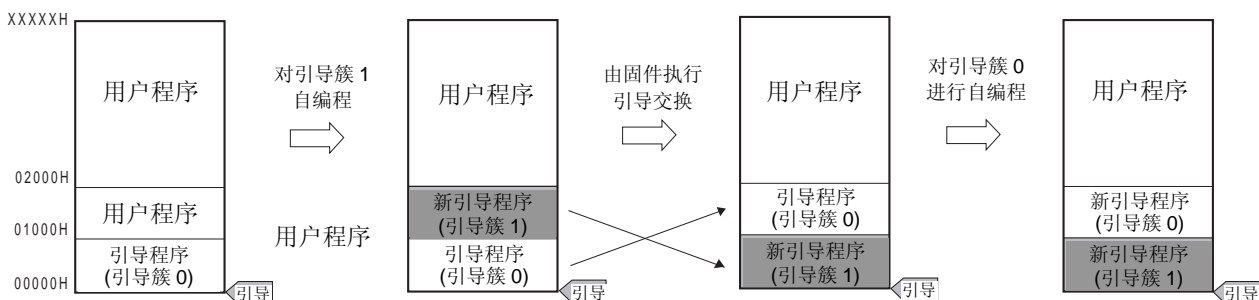
使用引导交换功能可以避免这种情况。

在擦除引导簇 0（这是一个引导程序区域）^{*}之前，先通过自编程将一个新的引导程序写入引导簇 1。当该程序正确写入引导簇 1 后，通过使用 78K0R/KF3 固件的设置信息功能将引导簇 1 与引导簇 0 的内容交换，这样引导簇 1 就用作引导区域。之后对引导程序区域重新初始化，即对引导簇 0 进行擦除或写操作。

这样，即使在重写引导程序区域时出现了电源失效，也能够正确执行程序，这是因为当程序复位并再次启动时，可以从引导交换后的引导簇 1 的引导启动。

注 引导簇是 4 KB 区域，引导簇 0 和 1 通过引导交换功能进行交换。

图 24-11. 引导交换功能

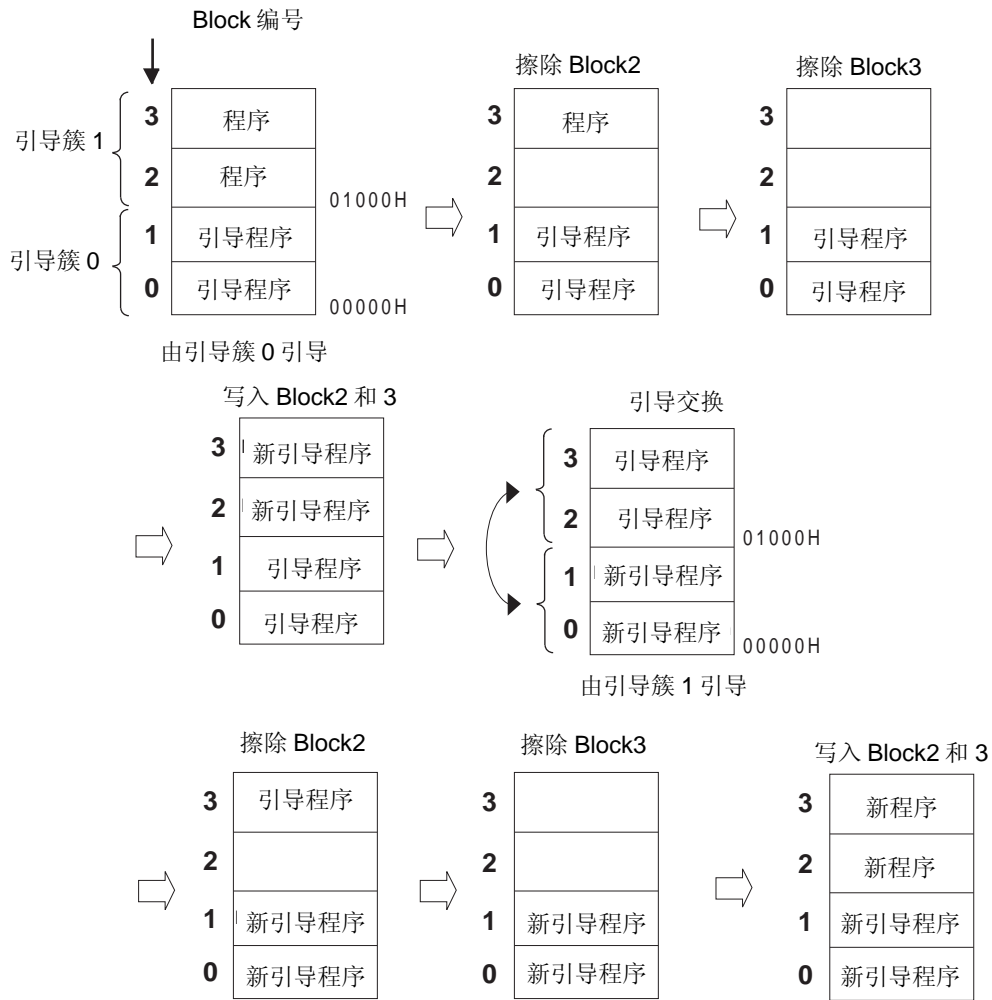


上图的示例如下。

引导簇 0: 引导交换前的引导程序区域

引导簇 1: 引导交换后的引导程序区域

图 24-12. 引导交换执行示例



24.8.2 Flash防护窗口功能

Flash 防护窗口功能是为自编程提供了一种安全功能。

在自编程期间，在指定作为窗口的范围内允许对 flash 存储器进行写入和擦除操作，在指定范围以外禁止写入和擦除在 flash 存储器操作。

在 on-board/off-board 编程和自编程时，可以通过设置和修改来进行扩大或缩小窗口范围。但是，防护功能只在自编程期间有效。在 on-board/off-board 编程时，在防护窗口范围之外，同样允许对 flash 存储器进行写入和擦除操作。

注意事项 如果引导簇 0 的重写禁止区域和 flash 防护窗口范围重叠，则禁止重写引导簇 0 优先。

表 24-9. Flash防护窗口功能设置/修改方法和命令之间的关系

编程条件	窗口范围 设置/修改方法	执行命令	
		块擦除	写入
自编程	通过设置信息库，指定开始和结束块（block）。	只有在窗口范围内允许块擦除。	只有在窗口范围内允许写入。
On-board/Off-board 编程	在专用 Flash 存储器编程器的 GUI 中，指定开始和结束块（block），等。	超出窗口范围之外，也允许块擦除。	超出窗口范围之外，也允许写入。

备注 参见 24.7 安全设置，禁止在 on-board/off-board 编程时写入/擦除。

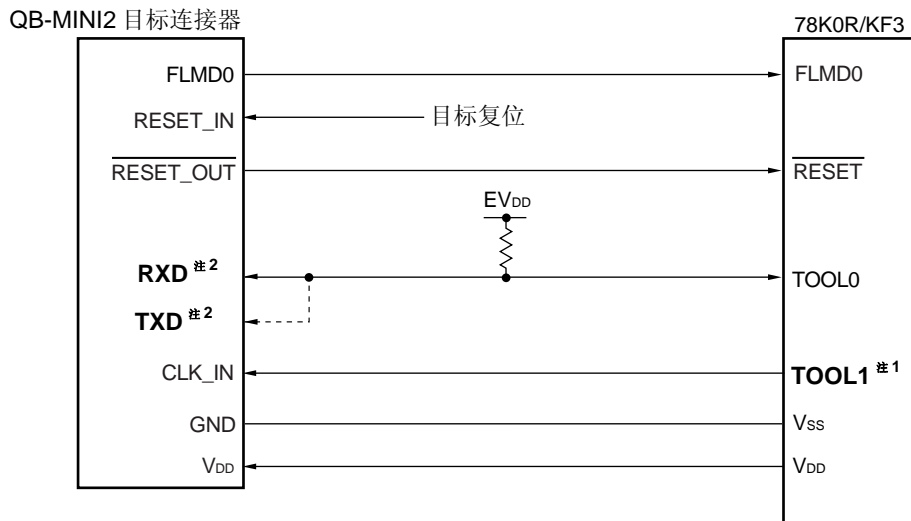
第二十五章 片上调试功能

25.1 连接QB-MINI2 到 78K0R/KF3

78K0R/KF3 通过片上调试仿真器 (QB-MINI2) 与主机通信, 使用 V_{DD} , FLMD0, \overline{RESET} , TOOL0, TOOL1^注 和 V_{SS} 引脚来进行片上调试。

注意事项 78K0R/KF3 具有片上调试功能。由于在使用片上调试功能后产品的稳定性不能得到保证, 因此在给定 flash 存储器可重写次数的情况下不要将该产品用于大规模生产。NEC Electronics 不接受该产品的相关投诉。

图 25-1. QB-MINI2 和 78K0R/KF3 的连接示例



- 注**
1. 在单线模式下通讯不需要连接, 但在 2 线模式下需要连接进行通信。此时, 因为不连接 QB-MINI2 时 TOOL1 引脚未使用, 只需要根据表 2-2 未使用引脚的连接进行必要的连接。
 2. 由于 QB-MINI2 内的 RXD 和 TXD 短接, 因此虚线部分不是必须的。当使用其它 flash 存储器编程器, 在编程器内 RXD 和 TXD 可能不会短接, 在这种情况下, 他们必须在目标系统上短接。

备注 在片上调试时, 推荐 FLMD0 引脚为自编程而悬空。要通过外部下拉, 使用大于 100 k Ω 的电阻。

单线模式 (单线 UART) 使用 TOOL0 引脚, 2 线模式使用 TOOL0 和 TOOL1 引脚, 用于串行通信。flash 存储器编程使用单线模式, 片上调试使用单线模式或 2 线模式。表 25-1 列出了单线模式和 2 线模式之间的区别。

表 25-1. 单线模式和 2 线模式之间的区别列表

通信模式	Flash 存储器编程功能	调试功能
单线模式	可用	<ul style="list-style-type: none"> 不支持伪实时 RAM 监控 (RRM) 功能。 不支持 DMM 功能 (在运行中动态修改存储器)。 调试器的速度比 2 线模式慢 2 到 4 倍。
2 线模式	无	<ul style="list-style-type: none"> 支持伪实时 RAM 监控 (RRM) 功能。 支持 DMM 功能 (在运行中动态修改存储器)。

备注 2 线模式不能用于 flash 编程, 因此, 即使 TOOL1 引脚连接到 QB-MINI2 的 CLK_IN, 写入操作也没有问题, 可以正常执行。

25.2 片上调试安全 ID

在 78K0R/KF3 中, 有一个片上调试操作控制标志存放在 flash 存储器地址 000C3H 中 (参见第二十三章 选项字), 片上调试安全 ID 设置在地址 000C4H~ 000CDH 中, 以防止第三方读取存储器内容。

当使用引导交换功能时, 由于 000C3H、000C4H ~ 000CDH 与 010C3H 以及 010C4H ~ 010CDH 的内容被交换, 因此预先设置的值应该与 010C3H 以及 010C4H ~ 010CDH 的内容相同。

关于片上调试安全 ID, 请参考 **QB-MINI2 有编程功能的片上调试仿真器用户手册 (U18371)**。

表 25-2. 片上调试安全 ID

地址	片上调试安全 ID
000C4H~000CDH	10 个字节长度的任意 ID 代码
010C4H~ 010CDH	

<R> 25.3 用户资源的保留

要执行 78K0R/KF3 和 QB-MINI2 之间的串行通信, 也就是每个调试功能, 必须预先保留存储器空间。如果使用 NEC 电子的汇编器 RA78K0R 或 C 编译器 CC78K0R, 可以通过链接选项进行设置。

(1) 存储器空间的保留

图 25-2 中的阴影部分是保留给调试监控程序的区域, 所以用户程序或数据不能被分配在这些空间。当使用片上调试功能时, 这些空间必须是保留的, 不能被用户程序使用。而且, 这些区域也不能被用户程序修改。

第二十六章 BCD修正电路

26.1 BCD修正电路功能

可通过此电路获得 BCD（二进制码十进数）码的加/减结果和 BCD 码，作为 BCD 码。
十进数修正操作结果可以通过执行加/减指令获得，A 寄存器作为操作数，加/减 BCDADJ 寄存器。

26.2 BCD修正电路使用的寄存器

BCD 修正电路使用如下寄存器。

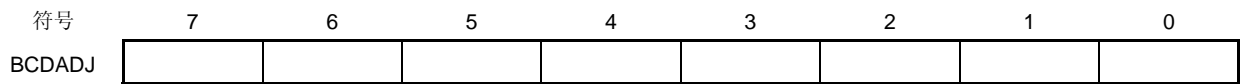
- BCD 修正结果寄存器（BCDADJ）

(1) BCD 修正结果寄存器（BCDADJ）

BCDADJ 寄存器保存修正值，使用 A 寄存器作为操作数，通过加/减指令获得作为 BCD 码的加/减结果。
从 BCDADJ 寄存器读取的值，会根据 A 寄存器的读取值和 CY 与 AC 标志的读取值不同而有所差异。
BCDADJ 可以通过 8 位存储器操作指令读取。
复位输入后此寄存器的值不确定。

图 26-1. BCD 修正结果寄存器（BCDADJ）的格式

地址: F00FEH 复位后: 不确定 R



26.3 BCD 修正电路操作

BCD 修正电路的基本操作如下。

(1) 加指令：使用一个 BCD 码值计算一个 BCD 码值和另一个 BCD 码值相加的结果

- <1> 要执行加法的 BCD 码值存储在 A 寄存器中。
- <2> A 寄存器与第二个操作数（另外一个待执行加法的 BCD 码值）的值按照二进制相加，二进制加法结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。
- <3> A 寄存器（二进制相加结果）的值和 BCDADJ 寄存器（修正值）按照二进制相加，进行十进制数修正，并且修正结果存储在 A 寄存器和 CY 寄存器中。

注意事项 从 BCDADJ 寄存器读取的值，会根据 A 寄存器的读取值和 CY 与 AC 标志的读取值不同而有所差异。因此，在指令<2>之后要执行指令<3>，而不能执行其他指令。要在中断允许状态下执行 BCD 修正，需要在中断函数中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为 BCD 加法的示例。

示例 1: $99 + 89 = 188$

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #99H ; <1>	99H	-	-	-
ADD A, #89H ; <2>	22H	1	1	66H
ADD A, !BCDADJ ; <3>	88H	1	0	-

示例 2: $85 + 15 = 100$

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #85H ; <1>	85H	-	-	-
ADD A, #15H ; <2>	9AH	0	0	06H
ADD A, !BCDADJ ; <3>	00H	1	1	-

示例 3: $80 + 80 = 160$

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #80H ; <1>	80H	-	-	-
ADD A, #80H ; <2>	00H	1	0	60H
ADD A, !BCDADJ ; <3>	60H	1	0	-

(2) 减指令：使用一个 BCD 码值计算一个 BCD 码值和另一个 BCD 码值相减的结果

- <1> 要执行减法的 BCD 码值存储在 A 寄存器中。
- <2> 从 A 寄存器中按照二进制减去第二个操作数（另外一个待执行减法的 BCD 码值），二进制减法结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。
- <3> 从 A 寄存器（二进制相加结果）中按照二进制减去 BCDADJ 寄存器（修正值），进行十进制数的减法修正，并且修正结果存储在 A 寄存器和 CY 寄存器中。

注意事项 从 BCDADJ 寄存器读取的值，会根据 A 寄存器的读取值和 CY 与 AC 标志的读取值不同而有所差异。因此，在指令<2>之后要执行指令<3>，而不能执行其他指令。要在中断允许状态下执行 BCD 修正，需要在中断函数中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为 BCD 减法的示例。

示例：91 - 52 = 39

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #91H ; <1>	91H	-	-	-
SUB A, #52H ; <2>	3FH	0	1	06H
SUB A, !BCDADJ ; <3>	39H	0	0	-

第二十七章 指令集

本章以表格方式列出了 78K0R 微控制器的指令集。如需了解每种指令的操作和操作代码，可参阅另外一个文档 **78K0R 微控制器指令用户手册(U17792E)**。

备注 表 27-5 操作列表中的阴影部分表示操作或指令格式是为 78K0R 微控制器新添加的。

27.1 操作列表使用规则

27.1.1 操作数标识符和标识方法

根据指令操作数标识方法（详情可参见汇编器规范）的规范描述方法，在每种指令的“操作数”栏内列出操作数。如果有两种或两种以上的描述方法，可选其中之一。大写字母和符号 #, !, !!, \$, \$!, [], 和 ES: 都是关键字，必须按其原样书写。每种符号的含义如下所示。

- #: 立即数标识
- !: 16 位绝对地址标识
- !!: 20 位绝对地址标识
- \$: 8 位相对地址标识
- \$!: 16 位相对地址标识
- []: 间接地址标识
- ES: : 扩展地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，注意必须加上符号 #, !, !!, \$, \$!, [], 和 ES: 。

对于操作数寄存器标识符，r 和 rp，功能名称 (X, A, C 等) 或绝对名称 (下表括号中的名称，R0, R1, R2 等) 都可用于描述标识。

表 27-1. 操作数标识符和描述方法

标识符	描述方法
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	特殊功能寄存器符号 (SFR 符号)
sfrp	特殊功能寄存器符号 (16 位可操作表 SFR 符号。仅用偶地址 [Ⓜ])
saddr	FFE20H~FFF1FH 立即数或标号
saddrp	FFE20H~FF1FH 立即数或标号 (仅用偶地址 [Ⓜ])
addr20	00000H~FFFFFH 立即数或标号
addr16	0000H~FFFFH 立即数或标号 (仅用于 16 位可操作寄存器偶地址 [Ⓜ])
addr5	0080H~00BFH 立即数或标号 (仅用偶地址)
word	16 位立即数或标号
byte	8 位立即数或标号
bit	3 位立即数或标号
RBn	RB0~RB3

注 当指定奇地址时，第0位 = 0。

备注 特殊功能寄存器符号，参见表 3-5 SFR 列表 和表 3-6 扩展 SFR (第二 SFR) 列表。

27.1.2 操作栏描述

执行指令时的操作，在操作栏中使用以下符号表示。

表 27-2. 在“操作”栏中的符号

符号	功能
A	A 寄存器; 8 位累加器
X	X 寄存器
B	B 寄存器
C	C 寄存器
D	D 寄存器
E	E 寄存器
H	H 寄存器
L	L 寄存器
ES	ES 寄存器
CS	CS 寄存器
AX	AX 寄存器对; 16 位累加器
BC	BC 寄存器对
DE	DE 寄存器对
HL	HL 寄存器对
PC	程序计数器
SP	堆栈指针
PSW	程序状态字
CY	进位标志
AC	辅助进位标志
Z	零标志
RBS	寄存器组选择标志
IE	中断请求允许标志
()	括号中的地址或寄存器所指的存储单元的内容
X _H , X _L	16 位寄存器: X _H = 高 8 位, X _L = 低 8 位
X _S , X _H , X _L	20 位寄存器: X _S = (第 19~16 位), X _H = (第 15~8 位), X _L = (第 7~0 位)
^	逻辑与(AND)
∨	逻辑或(OR)
⊕	逻辑异或(exclusive OR)
—	数据取反
addr16	16 位立即数
addr20	20 位立即数
jdisp8	带符号的 8 位数据(偏移量)
jdisp16	带符号的 16 位数据(偏移量)

27.1.3 标志操作栏的描述

执行指令时标志值的改变在标志栏中如以下符号所示。

表 27-3. “标志” 栏中的符号

符号	标志值的改变
(空)	不变
0	清为 0
1	设置为 1
×	根据结果进行设置/清零
R	恢复先前保存的值

27.1.4 前缀指令

带“ES:”的指令具有前缀操作码，可以将数据访问区域扩展到 1MB (00000H~FFFFFFH) 空间，通过将 ES 寄存器的值与 F0000H 到 FFFFFH 的 64KB 空间相加。当前缀操作码作为前缀附加到目标指令时，只有紧跟在前缀操作码之后的一条指令执行，其地址与 ES 寄存器的值相加。

表 27-4. 使用前缀操作码的示例

指令	操作码				
	1	2	3	4	5
MOV laddr16, #byte	CFH	laddr16		#byte	-
MOV ES:laddr16, #byte	11H	CFH	laddr16		#byte
MOV A, [HL]	8BH	-	-	-	-
MOV A, ES:[HL]	11H	8BH	-	-	-

注意事项 在执行前缀指令之前，用 MOV ES, A 等设置 ES 寄存器的值。

27.2 操作列表

表 27-5. 操作列表 (1/17)

指令组	助记符	操作数	字节	时钟		操作	标志			
				注 1	注 2		Z	AC	CY	
8 位数据 发送	MOV	r, #byte	2	1	-	r ← byte				
		saddr, #byte	3	1	-	(saddr) ← byte				
		sfr, #byte	3	1	-	sfr ← byte				
		!addr16, #byte	4	1	-	(addr16) ← byte				
		A, r ^{注 3}	1	1	-	A ← r				
		r, A ^{注 3}	1	1	-	r ← A				
		A, saddr	2	1	-	A ← (saddr)				
		saddr, A	2	1	-	(saddr) ← A				
		A, sfr	2	1	-	A ← sfr				
		sfr, A	2	1	-	sfr ← A				
		A, !addr16	3	1	4	A ← (addr16)				
		!addr16, A	3	1	-	(addr16) ← A				
		PSW, #byte	3	3	-	PSW ← byte		x	x	x
		A, PSW	2	1	-	A ← PSW				
		PSW, A	2	3	-	PSW ← A		x	x	x
		ES, #byte	2	1	-	ES ← byte				
		ES, saddr	3	1	-	ES ← (saddr)				
		A, ES	2	1	-	A ← ES				
		ES, A	2	1	-	ES ← A				
		CS, #byte	3	1	-	CS ← byte				
		A, CS	2	1	-	A ← CS				
		CS, A	2	1	-	CS ← A				
		A, [DE]	1	1	4	A ← (DE)				
		[DE], A	1	1	-	(DE) ← A				
		[DE + byte], #byte	3	1	-	(DE + byte) ← byte				
		A, [DE + byte]	2	1	4	A ← (DE + byte)				
		[DE + byte], A	2	1	-	(DE + byte) ← A				
A, [HL]	1	1	4	A ← (HL)						
[HL], A	1	1	-	(HL) ← A						
[HL + byte], #byte	3	1	-	(HL + byte) ← byte						

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 除 r = A 外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (2/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 发送	MOV	A, [HL + byte]	2	1	4	$A \leftarrow (HL + \text{byte})$			
		[HL + byte], A	2	1	-	$(HL + \text{byte}) \leftarrow A$			
		A, [HL + B]	2	1	4	$A \leftarrow (HL + B)$			
		[HL + B], A	2	1	-	$(HL + B) \leftarrow A$			
		A, [HL + C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL + C], A	2	1	-	$(HL + C) \leftarrow A$			
		word[B], #byte	4	1	-	$(B + \text{word}) \leftarrow \text{byte}$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	-	$(B + \text{word}) \leftarrow A$			
		word[C], #byte	4	1	-	$(C + \text{word}) \leftarrow \text{byte}$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	-	$(C + \text{word}) \leftarrow A$			
		word[BC], #byte	4	1	-	$(BC + \text{word}) \leftarrow \text{byte}$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	-	$(BC + \text{word}) \leftarrow A$			
		[SP + byte], #byte	3	1	-	$(SP + \text{byte}) \leftarrow \text{byte}$			
		A, [SP + byte]	2	1	-	$A \leftarrow (SP + \text{byte})$			
		[SP + byte], A	2	1	-	$(SP + \text{byte}) \leftarrow A$			
		B, saddr	2	1	-	$B \leftarrow (\text{saddr})$			
		B, laddr16	3	1	4	$B \leftarrow (\text{addr16})$			
		C, saddr	2	1	-	$C \leftarrow (\text{saddr})$			
		C, laddr16	3	1	4	$C \leftarrow (\text{addr16})$			
		X, saddr	2	1	-	$X \leftarrow (\text{saddr})$			
		X, laddr16	3	1	4	$X \leftarrow (\text{addr16})$			
		ES:laddr16, #byte	5	2	-	$(ES, \text{addr16}) \leftarrow \text{byte}$			
		A, ES:laddr16	4	2	5	$A \leftarrow (ES, \text{addr16})$			
		ES:laddr16, A	4	2	-	$(ES, \text{addr16}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	-	$(ES, DE) \leftarrow A$			
		ES:[DE + byte], #byte	4	2	-	$((ES, DE) + \text{byte}) \leftarrow \text{byte}$			
A, ES:[DE + byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$					
ES:[DE + byte], A	3	2	-	$((ES, DE) + \text{byte}) \leftarrow A$					

- 注 1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟（fcpu）的一个周期。
 2. 时钟周期的数量用于内部 ROM（Flash 存储器）的程序。

表 27-5. 操作列表 (3/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 发送	MOV	A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	-	(ES, HL) ← A			
		ES:[HL + byte], #byte	4	2	-	((ES, HL) + byte) ← byte			
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL + byte], A	3	2	-	((ES, HL) + byte) ← A			
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL + B], A	3	2	-	((ES, HL) + B) ← A			
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL + C], A	3	2	-	((ES, HL) + C) ← A			
		ES:word[B], #byte	5	2	-	((ES, B) + word) ← byte			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	-	((ES, B) + word) ← A			
		ES:word[C], #byte	5	2	-	((ES, C) + word) ← byte			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	-	((ES, C) + word) ← A			
		ES:word[BC], #byte	5	2	-	((ES, BC) + word) ← byte			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	-	((ES, BC) + word) ← A			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
	X, ES:!addr16	4	2	5	X ← (ES, addr16)				
	XCH	A, r ^{注 3}	1 (r = X) 2 (除 r = X 外)	1	-	A ↔ r			
		A, saddr	3	2	-	A ↔ (saddr)			
		A, sfr	3	2	-	A ↔ sfr			
		A, !addr16	4	2	-	A ↔ (addr16)			
		A, [DE]	2	2	-	A ↔ (DE)			
		A, [DE + byte]	3	2	-	A ↔ (DE + byte)			
		A, [HL]	2	2	-	A ↔ (HL)			
		A, [HL + byte]	3	2	-	A ↔ (HL + byte)			
A, [HL + B]		2	2	-	A ↔ (HL + B)				
A, [HL + C]	2	2	-	A ↔ (HL + C)					

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 除 r = A 外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (4/17)

指令组	助记符	操作数	字节	时钟		操作	标志			
				注 1	注 2		Z	AC	CY	
8 位数据发送	XCH	A, ES:!addr16	5	3	-	A \leftrightarrow (ES, addr16)				
		A, ES:[DE]	3	3	-	A \leftrightarrow (ES, DE)				
		A, ES:[DE + byte]	4	3	-	A \leftrightarrow ((ES, DE) + byte)				
		A, ES:[HL]	3	3	-	A \leftrightarrow (ES, HL)				
		A, ES:[HL + byte]	4	3	-	A \leftrightarrow ((ES, HL) + byte)				
		A, ES:[HL + B]	3	3	-	A \leftrightarrow ((ES, HL) + B)				
		A, ES:[HL + C]	3	3	-	A \leftrightarrow ((ES, HL) + C)				
	ONEB	A	1	1	-	A \leftarrow 01H				
		X	1	1	-	X \leftarrow 01H				
		B	1	1	-	B \leftarrow 01H				
		C	1	1	-	C \leftarrow 01H				
		saddr	2	1	-	(saddr) \leftarrow 01H				
		!addr16	3	1	-	(addr16) \leftarrow 01H				
		ES:!addr16	4	2	-	(ES, addr16) \leftarrow 01H				
	CLRB	A	1	1	-	A \leftarrow 00H				
		X	1	1	-	X \leftarrow 00H				
		B	1	1	-	B \leftarrow 00H				
		C	1	1	-	C \leftarrow 00H				
		saddr	2	1	-	(saddr) \leftarrow 00H				
		!addr16	3	1	-	(addr16) \leftarrow 00H				
		ES:!addr16	4	2	-	(ES,addr16) \leftarrow 00H				
	MOVS	[HL + byte], X	3	1	-	(HL + byte) \leftarrow X	x		x	
		ES:[HL + byte], X	4	2	-	(ES, HL + byte) \leftarrow X	x		x	
	16 位数据发送	MOVW	rp, #word	3	1	-	rp \leftarrow word			
			saddrp, #word	4	1	-	(saddrp) \leftarrow word			
			sfrp, #word	4	1	-	sfrp \leftarrow word			
			AX, saddrp	2	1	-	AX \leftarrow (saddrp)			
			saddrp, AX	2	1	-	(saddrp) \leftarrow AX			
AX, sfrp			2	1	-	AX \leftarrow sfrp				
sfrp, AX			2	1	-	sfrp \leftarrow AX				
AX, rp ^{注 3}			1	1	-	AX \leftarrow rp				
rp, AX ^{注 3}			1	1	-	rp \leftarrow AX				

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 除 r = A 外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (5/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数 数据发送	MOVW	AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	-	(addr16) ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	-	(DE) ← AX			
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)			
		[DE + byte], AX	2	1	-	(DE + byte) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	-	(HL) ← AX			
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)			
		[HL + byte], AX	2	1	-	(HL + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	-	(B + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	-	(C + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	-	(BC + word) ← AX			
		AX, [SP + byte]	2	1	-	AX ← (SP + byte)			
		[SP + byte], AX	2	1	-	(SP + byte) ← AX			
		BC, saddrp	2	1	-	BC ← (saddrp)			
		BC, !addr16	3	1	4	BC ← (addr16)			
		DE, saddrp	2	1	-	DE ← (saddrp)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		HL, saddrp	2	1	-	HL ← (saddrp)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	-	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	-	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	-	((ES, DE) + byte) ← AX			
AX, ES:[HL]	2	2	5	AX ← (ES, HL)					
ES:[HL], AX	2	2	-	(ES, HL) ← AX					

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (6/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数 数据发送	MOVW	AX, ES:[HL + byte]	3	2	5	$AX \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], AX	3	2	-	$((ES, HL) + \text{byte}) \leftarrow AX$			
		AX, ES:word[B]	4	2	5	$AX \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], AX	4	2	-	$((ES, B) + \text{word}) \leftarrow AX$			
		AX, ES:word[C]	4	2	5	$AX \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], AX	4	2	-	$((ES, C) + \text{word}) \leftarrow AX$			
		AX, ES:word[BC]	4	2	5	$AX \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], AX	4	2	-	$((ES, BC) + \text{word}) \leftarrow AX$			
		BC, ES:!addr16	4	2	5	$BC \leftarrow (ES, \text{addr16})$			
		DE, ES:!addr16	4	2	5	$DE \leftarrow (ES, \text{addr16})$			
	HL, ES:!addr16	4	2	5	$HL \leftarrow (ES, \text{addr16})$				
	XCHW	AX, rp ^{注 3}	1	1	-	$AX \leftrightarrow rp$			
	ONEW	AX	1	1	-	$AX \leftarrow 0001H$			
		BC	1	1	-	$BC \leftarrow 0001H$			
CLRW	AX	1	1	-	$AX \leftarrow 0000H$				
	BC	1	1	-	$BC \leftarrow 0000H$				
8 位操作	ADD	A, #byte	2	1	-	$A, CY \leftarrow A + \text{byte}$	x	x	x
		saddr, #byte	3	2	-	$(saddr), CY \leftarrow (saddr) + \text{byte}$	x	x	x
		A, r ^{注 4}	2	1	-	$A, CY \leftarrow A + r$	x	x	x
		r, A	2	1	-	$r, CY \leftarrow r + A$	x	x	x
		A, saddr	2	1	-	$A, CY \leftarrow A + (saddr)$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16})$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL)$	x	x	x
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte})$	x	x	x
		A, [HL + B]	2	1	4	$A, CY \leftarrow A + (HL + B)$	x	x	x
		A, [HL + C]	2	1	4	$A, CY \leftarrow A + (HL + C)$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16})$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL)$	x	x	x
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte})$	x	x	x
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B)$	x	x	x
A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C)$	x	x	x		

注 1. 当访问内部 RAM 区域或 SFR 区域时, 或执行没有数据访问的指令时。

2. 当访问程序存储器区域时。

3. 除 $Rp = AX$ 外

4. 除 $r = A$ 外

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。

2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (7/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	ADDC	A, #byte	2	1	-	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	2	-	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r ^{注 3}	2	1	-	A, CY ← A + r + CY	x	x	x
		r, A	2	1	-	r, CY ← r + A + CY	x	x	x
		A, saddr	2	1	-	A, CY ← A + (saddr) + CY	x	x	x
		A, laddr16	3	1	4	A, CY ← A + (addr16) + CY	x	x	x
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	x	x	x
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	x	x	x
		A, ES:laddr16	4	2	5	A, CY ← A + (ES, addr16) + CY	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	x	x	x
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	x	x	x
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	x	x	x
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	x	x	x
		SUB	A, #byte	2	1	-	A, CY ← A - byte	x	x
	saddr, #byte		3	2	-	(saddr), CY ← (saddr) - byte	x	x	x
	A, r ^{注 3}		2	1	-	A, CY ← A - r	x	x	x
	r, A		2	1	-	r, CY ← r - A	x	x	x
	A, saddr		2	1	-	A, CY ← A - (saddr)	x	x	x
	A, laddr16		3	1	4	A, CY ← A - (addr16)	x	x	x
	A, [HL]		1	1	4	A, CY ← A - (HL)	x	x	x
	A, [HL + byte]		2	1	4	A, CY ← A - (HL + byte)	x	x	x
	A, [HL + B]		2	1	4	A, CY ← A - (HL + B)	x	x	x
	A, [HL + C]		2	1	4	A, CY ← A - (HL + C)	x	x	x
	A, ES:laddr16		4	2	5	A, CY ← A - (ES:addr16)	x	x	x
	A, ES:[HL]		2	2	5	A, CY ← A - (ES:HL)	x	x	x

注 1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。

2. 当访问程序存储器区域时。

3. 除 r = A 外

备注

1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。

2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (8/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	SUBC	A, #byte	2	1	-	A, CY ← A - byte - CY	x	x	x
		saddr, #byte	3	2	-	(saddr), CY ← (saddr) - byte - CY	x	x	x
		A, r ^{注 3}	2	1	-	A, CY ← A - r - CY	x	x	x
		r, A	2	1	-	r, CY ← r - A - CY	x	x	x
		A, saddr	2	1	-	A, CY ← A - (saddr) - CY	x	x	x
		A, laddr16	3	1	4	A, CY ← A - (addr16) - CY	x	x	x
		A, [HL]	1	1	4	A, CY ← A - (HL) - CY	x	x	x
		A, [HL + byte]	2	1	4	A, CY ← A - (HL + byte) - CY	x	x	x
		A, [HL + B]	2	1	4	A, CY ← A - (HL + B) - CY	x	x	x
		A, [HL + C]	2	1	4	A, CY ← A - (HL + C) - CY	x	x	x
		A, ES:laddr16	4	2	5	A, CY ← A - (ES:addr16) - CY	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A - (ES:HL) - CY	x	x	x
		A, ES:[HL + byte]	3	2	5	A, CY ← A - ((ES:HL) + byte) - CY	x	x	x
		A, ES:[HL + B]	3	2	5	A, CY ← A - ((ES:HL) + B) - CY	x	x	x
		A, ES:[HL + C]	3	2	5	A, CY ← A - ((ES:HL) + C) - CY	x	x	x
		AND	A, #byte	2	1	-	A ← A ∧ byte	x	
	saddr, #byte		3	2	-	(saddr) ← (saddr) ∧ byte	x		
	A, r ^{注 3}		2	1	-	A ← A ∧ r	x		
	r, A		2	1	-	r ← r ∧ A	x		
	A, saddr		2	1	-	A ← A ∧ (saddr)	x		
	A, laddr16		3	1	4	A ← A ∧ (addr16)	x		
	A, [HL]		1	1	4	A ← A ∧ (HL)	x		
	A, [HL + byte]		2	1	4	A ← A ∧ (HL + byte)	x		
	A, [HL + B]		2	1	4	A ← A ∧ (HL + B)	x		
	A, [HL + C]		2	1	4	A ← A ∧ (HL + C)	x		
	A, ES:laddr16		4	2	5	A ← A ∧ (ES:addr16)	x		
	A, ES:[HL]		2	2	5	A ← A ∧ (ES:HL)	x		
	A, ES:[HL + byte]		3	2	5	A ← A ∧ ((ES:HL) + byte)	x		
	A, ES:[HL + B]		3	2	5	A ← A ∧ ((ES:HL) + B)	x		
	A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	x			

注 1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。

2. 当访问程序存储器区域时。

3. 除 r = A 外

备注

1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。

2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (9/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	OR	A, #byte	2	1	-	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	2	-	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{注 3}	2	1	-	$A \leftarrow A \vee r$	x		
		r, A	2	1	-	$r \leftarrow r \vee A$	x		
		A, saddr	2	1	-	$A \leftarrow A \vee (\text{saddr})$	x		
		A, laddr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	x		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	x		
		A, [HL + byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	x		
		A, [HL + C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	x		
		A, ES:laddr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	x		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	x		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	x		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	x		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	x		
		XOR	A, #byte	2	1	-	$A \leftarrow A \oplus \text{byte}$	x	
	saddr, #byte		3	2	-	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$	x		
	A, r ^{注 3}		2	1	-	$A \leftarrow A \oplus r$	x		
	r, A		2	1	-	$r \leftarrow r \oplus A$	x		
	A, saddr		2	1	-	$A \leftarrow A \oplus (\text{saddr})$	x		
	A, laddr16		3	1	4	$A \leftarrow A \oplus (\text{addr16})$	x		
	A, [HL]		1	1	4	$A \leftarrow A \oplus (\text{HL})$	x		
	A, [HL + byte]		2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{byte})$	x		
	A, [HL + B]		2	1	4	$A \leftarrow A \oplus (\text{HL} + B)$	x		
	A, [HL + C]		2	1	4	$A \leftarrow A \oplus (\text{HL} + C)$	x		
	A, ES:laddr16		4	2	5	$A \leftarrow A \oplus (\text{ES:addr16})$	x		
	A, ES:[HL]		2	2	5	$A \leftarrow A \oplus (\text{ES:HL})$	x		
	A, ES:[HL + byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{byte})$	x			
A, ES:[HL + B]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + B)$	x				
A, ES:[HL + C]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + C)$	x				

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 除 $r = A$ 外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (10/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	CMP	A, #byte	2	1	-	A - byte	x	x	x
		saddr, #byte	3	1	-	(saddr) - byte	x	x	x
		A, r ^{注 3}	2	1	-	A - r	x	x	x
		r, A	2	1	-	r - A	x	x	x
		A, saddr	2	1	-	A - (saddr)	x	x	x
		A, laddr16	3	1	4	A - (addr16)	x	x	x
		A, [HL]	1	1	4	A - (HL)	x	x	x
		A, [HL + byte]	2	1	4	A - (HL + byte)	x	x	x
		A, [HL + B]	2	1	4	A - (HL + B)	x	x	x
		A, [HL + C]	2	1	4	A - (HL + C)	x	x	x
		laddr16, #byte	4	1	4	(addr16) - byte	x	x	x
		A, ES:laddr16	4	2	5	A - (ES:addr16)	x	x	x
		A, ES:[HL]	2	2	5	A - (ES:HL)	x	x	x
		A, ES:[HL + byte]	3	2	5	A - ((ES:HL) + byte)	x	x	x
		A, ES:[HL + B]	3	2	5	A - ((ES:HL) + B)	x	x	x
		A, ES:[HL + C]	3	2	5	A - ((ES:HL) + C)	x	x	x
		ES:laddr16, #byte	5	2	5	(ES:addr16) - byte	x	x	x
	CMP0	A	1	1	-	A - 00H	x	x	x
		X	1	1	-	X - 00H	x	x	x
		B	1	1	-	B - 00H	x	x	x
		C	1	1	-	C - 00H	x	x	x
		saddr	2	1	-	(saddr) - 00H	x	x	x
		laddr16	3	1	4	(addr16) - 00H	x	x	x
		ES:laddr16	4	2	5	(ES:addr16) - 00H	x	x	x
	CMPS	X, [HL + byte]	3	1	4	X - (HL + byte)	x	x	x
		X, ES:[HL + byte]	4	2	5	X - ((ES:HL) + byte)	x	x	x

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 除 r = A 外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (11/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位操作	ADDW	AX, #word	3	1	-	AX, CY ← AX + word	x	x	x
		AX, AX	1	1	-	AX, CY ← AX + AX	x	x	x
		AX, BC	1	1	-	AX, CY ← AX + BC	x	x	x
		AX, DE	1	1	-	AX, CY ← AX + DE	x	x	x
		AX, HL	1	1	-	AX, CY ← AX + HL	x	x	x
		AX, saddrp	2	1	-	AX, CY ← AX + (saddrp)	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	x	x	x
	SUBW	AX, #word	3	1	-	AX, CY ← AX - word	x	x	x
		AX, BC	1	1	-	AX, CY ← AX - BC	x	x	x
		AX, DE	1	1	-	AX, CY ← AX - DE	x	x	x
		AX, HL	1	1	-	AX, CY ← AX - HL	x	x	x
		AX, saddrp	2	1	-	AX, CY ← AX - (saddrp)	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX - (addr16)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX - (HL + byte)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX - (ES:addr16)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX - ((ES:HL) + byte)	x	x	x
		CMPW	AX, #word	3	1	-	AX - word	x	x
	AX, BC		1	1	-	AX - BC	x	x	x
	AX, DE		1	1	-	AX - DE	x	x	x
	AX, HL		1	1	-	AX - HL	x	x	x
	AX, saddrp		2	1	-	AX - (saddrp)	x	x	x
	AX, !addr16		3	1	4	AX - (addr16)	x	x	x
	AX, [HL+byte]		3	1	4	AX - (HL + byte)	x	x	x
	AX, ES:!addr16		4	2	5	AX - (ES:addr16)	x	x	x
AX, ES: [HL+byte]	4		2	5	AX - ((ES:HL) + byte)	x	x	x	
乘法	MULU	X	1	1	-	AX ← A × X			

注 1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (12/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
自加/自减	INC	r	1	1	-	$r \leftarrow r + 1$	×	×	
		saddr	2	2	-	$(saddr) \leftarrow (saddr) + 1$	×	×	
		!addr16	3	2	-	$(addr16) \leftarrow (addr16) + 1$	×	×	
		[HL+byte]	3	2	-	$(HL+byte) \leftarrow (HL+byte) + 1$	×	×	
		ES:!addr16	4	3	-	$(ES, addr16) \leftarrow (ES, addr16) + 1$	×	×	
		ES: [HL+byte]	4	3	-	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte) + 1$	×	×	
	DEC	r	1	1	-	$r \leftarrow r - 1$	×	×	
		saddr	2	2	-	$(saddr) \leftarrow (saddr) - 1$	×	×	
		!addr16	3	2	-	$(addr16) \leftarrow (addr16) - 1$	×	×	
		[HL+byte]	3	2	-	$(HL+byte) \leftarrow (HL+byte) - 1$	×	×	
		ES:!addr16	4	3	-	$(ES, addr16) \leftarrow (ES, addr16) - 1$	×	×	
		ES: [HL+byte]	4	3	-	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte) - 1$	×	×	
	INCW	rp	1	1	-	$rp \leftarrow rp + 1$			
		saddrp	2	2	-	$(saddrp) \leftarrow (saddrp) + 1$			
		!addr16	3	2	-	$(addr16) \leftarrow (addr16) + 1$			
		[HL+byte]	3	2	-	$(HL+byte) \leftarrow (HL+byte) + 1$			
		ES:!addr16	4	3	-	$(ES, addr16) \leftarrow (ES, addr16) + 1$			
		ES: [HL+byte]	4	3	-	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte) + 1$			
	DECW	rp	1	1	-	$rp \leftarrow rp - 1$			
		saddrp	2	2	-	$(saddrp) \leftarrow (saddrp) - 1$			
		!addr16	3	2	-	$(addr16) \leftarrow (addr16) - 1$			
		[HL+byte]	3	2	-	$(HL+byte) \leftarrow (HL+byte) - 1$			
		ES:!addr16	4	3	-	$(ES, addr16) \leftarrow (ES, addr16) - 1$			
		ES: [HL+byte]	4	3	-	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte) - 1$			
移位	SHR	A, cnt	2	1	-	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	-	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	-	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	-	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	-	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	-	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	-	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	-	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
SARW	AX, cnt	2	1	-	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×	

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (13/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
循环移位	ROR	A, 1	2	1	-	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	-	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	-	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	-	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX, 1	2	1	-	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC, 1	2	1	-	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, saddr.bit	3	1	-	$CY \leftarrow (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	-	$CY \leftarrow \text{sfr.bit}$			×
		CY, A.bit	2	1	-	$CY \leftarrow A.\text{bit}$			×
		CY, PSW.bit	3	1	-	$CY \leftarrow \text{PSW.bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow (\text{HL}).\text{bit}$			×
		saddr.bit, CY	3	2	-	$(\text{saddr}).\text{bit} \leftarrow CY$			
		sfr.bit, CY	3	2	-	$\text{sfr.bit} \leftarrow CY$			
		A.bit, CY	2	1	-	$A.\text{bit} \leftarrow CY$			
		PSW.bit, CY	3	4	-	$\text{PSW.bit} \leftarrow CY$	×	×	
		[HL].bit, CY	2	2	-	$(\text{HL}).\text{bit} \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (\text{ES}, \text{HL}).\text{bit}$			×
		ES:[HL].bit, CY	3	3	-	$(\text{ES}, \text{HL}).\text{bit} \leftarrow CY$			
	AND1	CY, saddr.bit	3	1	-	$CY \leftarrow CY \wedge (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	-	$CY \leftarrow CY \wedge \text{sfr.bit}$			×
		CY, A.bit	2	1	-	$CY \leftarrow CY \wedge A.\text{bit}$			×
		CY, PSW.bit	3	1	-	$CY \leftarrow CY \wedge \text{PSW.bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (\text{ES}, \text{HL}).\text{bit}$			×
	OR1	CY, saddr.bit	3	1	-	$CY \leftarrow CY \vee (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	-	$CY \leftarrow CY \vee \text{sfr.bit}$			×
		CY, A.bit	2	1	-	$CY \leftarrow CY \vee A.\text{bit}$			×
		CY, PSW.bit	3	1	-	$CY \leftarrow CY \vee \text{PSW.bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (\text{ES}, \text{HL}).\text{bit}$			×

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (14/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
位操作	XOR1	CY, saddr.bit	3	1	-	$CY \leftarrow CY \oplus (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	-	$CY \leftarrow CY \oplus \text{sfr}.\text{bit}$			×
		CY, A.bit	2	1	-	$CY \leftarrow CY \oplus A.\text{bit}$			×
		CY, PSW.bit	3	1	-	$CY \leftarrow CY \oplus \text{PSW}.\text{bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \oplus (\text{ES}, \text{HL}).\text{bit}$			×
	SET1	saddr.bit	3	2	-	$(\text{saddr}).\text{bit} \leftarrow 1$			
		sfr.bit	3	2	-	$\text{sfr}.\text{bit} \leftarrow 1$			
		A.bit	2	1	-	$A.\text{bit} \leftarrow 1$			
		!addr16.bit	4	2	-	$(\text{addr16}).\text{bit} \leftarrow 1$			
		PSW.bit	3	4	-	$\text{PSW}.\text{bit} \leftarrow 1$	×	×	×
		[HL].bit	2	2	-	$(\text{HL}).\text{bit} \leftarrow 1$			
		ES:!addr16.bit	5	3	-	$(\text{ES}, \text{addr16}).\text{bit} \leftarrow 1$			
		ES:[HL].bit	3	3	-	$(\text{ES}, \text{HL}).\text{bit} \leftarrow 1$			
	CLR1	saddr.bit	3	2	-	$(\text{saddr}.\text{bit}) \leftarrow 0$			
		sfr.bit	3	2	-	$\text{sfr}.\text{bit} \leftarrow 0$			
		A.bit	2	1	-	$A.\text{bit} \leftarrow 0$			
		!addr16.bit	4	2	-	$(\text{addr16}).\text{bit} \leftarrow 0$			
		PSW.bit	3	4	-	$\text{PSW}.\text{bit} \leftarrow 0$	×	×	×
		[HL].bit	2	2	-	$(\text{HL}).\text{bit} \leftarrow 0$			
		ES:!addr16.bit	5	3	-	$(\text{ES}, \text{addr16}).\text{bit} \leftarrow 0$			
		ES:[HL].bit	3	3	-	$(\text{ES}, \text{HL}).\text{bit} \leftarrow 0$			
	SET1	CY	2	1	-	$CY \leftarrow 1$			1
	CLR1	CY	2	1	-	$CY \leftarrow 0$			0
	NOT1	CY	2	1	-	$CY \leftarrow \overline{CY}$			×

- 注 1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟（fcpu）的一个周期。
 2. 时钟周期的数量用于内部 ROM（Flash 存储器）的程序。

表 27-5. 操作列表 (15/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
调用/ 返回	CALL	rp	2	3	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC \leftarrow CS, rp,$ $SP \leftarrow SP - 4$			
		\$!addr20	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow PC + 3 +$ $jdisp16,$ $SP \leftarrow SP - 4$			
		!addr16	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow 0000, addr16,$ $SP \leftarrow SP - 4$			
		!!addr20	4	3	–	$(SP - 2) \leftarrow (PC + 4)_S, (SP - 3) \leftarrow (PC + 4)_H,$ $(SP - 4) \leftarrow (PC + 4)_L, PC \leftarrow addr20,$ $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC_S \leftarrow 0000,$ $PC_H \leftarrow (0000, addr5 + 1),$ $PC_L \leftarrow (0000, addr5),$ $SP \leftarrow SP - 4$			
	BRK	–	2	5	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 2)_S,$ $(SP - 3) \leftarrow (PC + 2)_H, (SP - 4) \leftarrow (PC + 2)_L,$ $PC_S \leftarrow 0000,$ $PC_H \leftarrow (0007FH), PC_L \leftarrow (0007EH),$ $SP \leftarrow SP - 4, IE \leftarrow 0$			
	RET	–	1	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), SP \leftarrow SP + 4$			
RETI	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R	
RETB	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R	

- 注
1. 当访问内部 RAM 区域或 SFR 区域时，或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (16/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
堆栈操作	PUSH	PSW	2	1	-	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2			
		rp	1	1	-	(SP - 1) ← rp _H , (SP - 2) ← rp _L , SP ← SP - 2			
	POP	PSW	2	3	-	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	-	rp _L ← (SP), rp _H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	-	SP ← word			
		SP, AX	2	1	-	SP ← AX			
		AX, SP	2	1	-	AX ← SP			
		HL, SP	3	1	-	HL ← SP			
		BC, SP	3	1	-	BC ← SP			
		DE, SP	3	1	-	DE ← SP			
	ADDW	SP, #byte	2	1	-	SP ← SP + byte			
SUBW	SP, #byte	2	1	-	SP ← SP - byte				
无条件转移	BR	AX	2	3	-	PC ← CS, AX			
		\$addr20	2	3	-	PC ← PC + 2 + jdisp8			
		\$!addr20	3	3	-	PC ← PC + 3 + jdisp16			
		!addr16	3	3	-	PC ← 0000, addr16			
		!!addr20	4	3	-	PC ← addr20			
条件转移	BC	\$addr20	2	2/4 ^{注3}	-	如果 CY = 1, PC ← PC + 2 + jdisp8			
	BNC	\$addr20	2	2/4 ^{注3}	-	如果 CY = 0, PC ← PC + 2 + jdisp8			
	BZ	\$addr20	2	2/4 ^{注3}	-	如果 Z = 1, PC ← PC + 2 + jdisp8			
	BNZ	\$addr20	2	2/4 ^{注3}	-	如果 Z = 0, PC ← PC + 2 + jdisp8			
	BH	\$addr20	3	2/4 ^{注3}	-	如果 (Z ∨ CY) = 0, PC ← PC + 3 + jdisp8			
	BNH	\$addr20	3	2/4 ^{注3}	-	如果 (Z ∨ CY) = 1, PC ← PC + 3 + jdisp8			
	BT	saddr.bit, \$addr20	4	3/5 ^{注3}	-	如果 (saddr).bit = 1, PC ← PC + 4 + jdisp8			
		sfr.bit, \$addr20	4	3/5 ^{注3}	-	如果 sfr.bit = 1, PC ← PC + 4 + jdisp8			
		A.bit, \$addr20	3	3/5 ^{注3}	-	如果 A.bit = 1, PC ← PC + 3 + jdisp8			
		PSW.bit, \$addr20	4	3/5 ^{注3}	-	如果 PSW.bit = 1, PC ← PC + 4 + jdisp8			
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/8	如果 (HL).bit = 1, PC ← PC + 3 + jdisp8			
ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/9	如果 (ES, HL).bit = 1, PC ← PC + 4 + jdisp8					

- 注
1. 当访问内部 RAM 区域或 SFR 区域时, 或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 表示“当不符合条件时/符合条件时”的时钟数量。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

表 27-5. 操作列表 (17/17)

指令组	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
条件转移	BF	saddr.bit, \$addr20	4	3/5 ^{注3}	–	如果 (saddr).bit = 0, PC ← PC + 4 + jdisp8			
		sfr.bit, \$addr20	4	3/5 ^{注3}	–	如果 sfr.bit = 0, PC ← PC + 4 + jdisp8			
		A.bit, \$addr20	3	3/5 ^{注3}	–	如果 A.bit = 0, PC ← PC + 3 + jdisp8			
		PSW.bit, \$addr20	4	3/5 ^{注3}	–	如果 PSW.bit = 0, PC ← PC + 4 + jdisp8			
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/8	如果 (HL).bit = 0, PC ← PC + 3 + jdisp8			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/9	如果 (ES, HL).bit = 0, PC ← PC + 4 + jdisp8			
	BTCLR	saddr.bit, \$addr20	4	3/5 ^{注3}	–	如果 (saddr).bit = 1, 然后重置(saddr).bit, PC ← PC + 4 + jdisp8			
		sfr.bit, \$addr20	4	3/5 ^{注3}	–	如果 sfr.bit = 1, 然后重置 sfr.bit, PC ← PC + 4 + jdisp8			
		A.bit, \$addr20	3	3/5 ^{注3}	–	如果 A.bit = 1, 然后重置 A.bit, PC ← PC + 3 + jdisp8			
		PSW.bit, \$addr20	4	5/7 ^{注3}	–	如果 PSW.bit = 1, 然后重置 PSW.bit, PC ← PC + 4 + jdisp8	×	×	×
		[HL].bit, \$addr20	3	3/5 ^{注3}	–	如果 (HL).bit = 1, 然后重置(HL).bit, PC ← PC + 3 + jdisp8			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	–	如果 (ES, HL).bit = 1, 然后重置(ES, HL).bit, PC ← PC + 4 + jdisp8			
条件跳转	SKC	–	2	1	–	如果 CY = 1, 跳过下一条指令			
	SKNC	–	2	1	–	如果 CY = 0, 跳过下一条指令			
	SKZ	–	2	1	–	如果 Z = 1, 跳过下一条指令			
	SKNZ	–	2	1	–	如果 Z = 0, 跳过下一条指令			
	SKH	–	2	1	–	如果(Z ∨ CY) = 0, 跳过下一条指令			
	SKNH	–	2	1	–	如果(Z ∨ CY) = 1, 跳过下一条指令			
CPU 控制	SEL	Rb _n	2	1	–	RBS[1: 0] ← n			
	NOP	–	1	1	–	无操作			
	EI	–	3	4	–	IE ← 1(允许中断)			
	DI	–	3	4	–	IE ← 0(禁止中断)			
	HALT	–	2	3	–	设置 HALT 模式			
	STOP	–	2	3	–	设置 STOP 模式			

- 注
1. 当访问内部 RAM 区域或 SFR 区域时, 或执行没有数据访问的指令时。
 2. 当访问程序存储器区域时。
 3. 表示“当不符合条件时/符合条件时”的时钟数量。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 时钟周期的数量用于内部 ROM (Flash 存储器) 的程序。

<R>

第二十八章 电气特性

注意事项 **78K0R/KF3** 具有片上调试功能。不要将该产品用于大规模生产，因为在给定闪存重写次数的条件下，使用片上调试功能后，产品的可靠性得不到保证。
使用片上调试功能后，投诉将不被受理。

最大额定值 (TA = 25°C) (1/2)

参数	符号	条件	额定值	单位	
<R>	供电电压	V _{DD}	-0.5~+6.5	V	
		EV _{DD}	-0.5~+6.5	V	
	<R>	V _{SS}	-0.5~+0.3	V	
		EV _{SS}	-0.5~+0.3	V	
		AV _{REF0}	-0.5~V _{DD} +0.3 ^{※1}	V	
		AV _{REF1}	-0.5~V _{DD} +0.3 ^{※1}	V	
		AV _{SS}	-0.5~+0.3	V	
<R>	REGC 引脚输入电压	V _I REGC	REGC -0.3~3.6 以及-0.3~V _{DD} +0.3 ^{※2}	V	
<R>	输入电压	V _{I1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120 ~ P124, P140 ~ P145, EXCLK, /RESET, FLMD0	-0.3 ~ EV _{DD} +0.3 以及-0.3 ~ V _{DD} +0.3 ^{※1}	V
		V _{I2}	P60~P63 (N-ch 漏极开路)	-0.3~+6.5	V
		V _{I3}	P20 ~ P27	-0.3 ~ AV _{REF0} +0.3 以及-0.3 ~ V _{DD} +0.3 ^{※1}	V
		V _{I4}	P110, P111	-0.3 ~ AV _{REF1} +0.3 以及-0.3 ~ V _{DD} +0.3 ^{※1}	V
<R>	输出电压	V _O	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145	-0.3 ~ EV _{DD} +0.3 ^{※1}	V
		V _{O2}	P20 ~ P27	-0.3 ~ AV _{REF0} +0.3	V
		V _{O3}	P110, P111	-0.3 ~ AV _{REF1} +0.3	V
<R>	模拟 输入电压	V _{AN}	ANI0~ANI7	-0.3~AV _{REF0} +0.3 ^{※1} 以及-0.3~V _{DD} +0.3 ^{※1}	V
<R>	模拟 输出电压	V _{AO}	ANO0~ANO1	-0.3~AV _{REF1} +0.3 ^{※1}	V

注

1. 必须小于等于 6.5 V。
2. 通过一个电容 (0.47~1uF) 将 REGC 引脚连接到 V_{SS}。该值控制 REGC 引脚的绝对最大额定值。不要将电压输入这个引脚。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

最大额定值 (TA = 25°C) (2/2)

参数	符号	条件		额定值	单位
输出电流, 高	IOH1	每个引脚	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145,	-10	mA
		所有引脚总和 -80 mA	P00 ~ P04, P40 ~ P47, P120, P130, P140 ~ P145	-25	mA
			P05, P06, P10 ~ P17, P30, P31, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90	-55	mA
	IOH2	每个引脚	P20~P27, P110, P111	-0.5	mA
		所有引脚总和		-2	mA
		输出电流, 低	IOL1	每个引脚	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145
	IOL1	所有引脚总和 200 mA	P00 ~ P04, P40 ~ P47, P120, P130, P140 ~ P145	60	mA
			P05, P06, P10 ~ P17, P30, P31, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90	140	mA
		IOL2	每个引脚	P20~P27, P110, P111	1
所有引脚总和	5		mA		
工作环境温度	TA	在普通操作模式下		-40~+85	°C
		在 flash 存储器编程模式下			
存储温度	Tstg			-65~+150	°C

<R>

注意事项 任何一项参数哪怕是在瞬间超过最大额定值, 都会使产品质量受到影响。也就是说, 最大额定值是产品濒临物理损坏的临界点, 因而, 必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明, 复用功能引脚的特性与那些端口引脚的特性相同。

X1 振荡器特性

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

振荡器	推荐电路	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷振荡器		X1 时钟振荡频率 (fx) 注	2.7 V ≤ VDD ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ VDD < 2.7 V	2.0		5.0	
晶体振荡器		X1 时钟振荡频率 (fx) 注	2.7 V ≤ VDD ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ VDD < 2.7 V	2.0		5.0	

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性**。

注意事项 1. 在使用 X1 时钟时，上图中虚线框内的部分的配线应按照如下布线方法进行布线，以防止连接线的寄生电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 若信号线流经的电流变化较大，则不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 VSS 相同。
- 不要将电容的地信号从流经大电流的接地模式接地。
- 不要从振荡器获取信号。

2. 由于复位后 CPU 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器 (OSTC) 检测 X1 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再决定 OSTC 和振荡稳定时间选择寄存器 (OSTS) 的振荡稳定时间。

备注 对于振荡电路选择和振荡器常数，顾客需要自己评价振荡特性或请振荡器厂商评估。

内部振荡器特性

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

振荡器	参数	条件	MIN.	TYP.	MAX.	单位	
8 MHz 内部振荡器	内部高速振荡时钟频率 (f _H) ^{※1}	2.7 V ≤ V _{DD} ≤ 5.5 V	7.6	8.0	8.4	MHz	
		1.8 V ≤ V _{DD} < 2.7 V	5.0	8.0	8.4	MHz	
240 kHz 内部振荡器	内部低速振荡时钟频率 (f _L)	普通电流模式	2.7 V ≤ V _{DD} ≤ 5.5 V	216	240	264	kHz
			1.8 V ≤ V _{DD} < 2.7 V	192	240	264	kHz
		低功耗电流模式 ^{※2}		192	240	264	kHz

- <R> 注 1. 仅表示当 HIOTRM 寄存器设置为 10H 时的振荡器特性。如需了解指令执行时间，参见 AC 特性。
2. 在如下情况下稳压器输出设置为低功耗电流模式：
- 当 RMC 寄存器设置为 5AH。
 - 系统复位期间。
 - 在 STOP 模式下（除 OCD 模式期间外）。
 - 当 CPU 使用副系统时钟 (f_{X_T}) 时，高速系统时钟 (f_{M_X}) 和高速内部振荡时钟 (f_H) 均停止。
 - 当设置 CPU 使用副系统时钟 (f_{X_T}) 时，在 HALT 模式期间高速系统时钟 (f_{M_X}) 和高速内部振荡时钟 (f_H) 都停止。

备注 普通电流模式和依照稳压器输出电压的低功耗电流模式的详细情况，参见第二十二章 稳压器。

XT1 振荡器特性

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

振荡器	推荐电路	项目	条件	MIN.	TYP.	MAX.	单位
晶体振荡器		XT1 时钟振荡频率 (f_{XT}) ^注		32	32.768	35	kHz

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC** 特性。

注意事项 1. 在使用 **X1** 时钟时，上图中虚线框内的部分的配线应按照如下布线方法进行布线，以防止连接线的寄生电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 若信号线流经的电流变化较大，则不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
- 不要将电容的地信号从流经大电流的接地模式接地。
- 不要从振荡器获取信号。

2. **XT1** 振荡器是一个低振幅电路，用于降低功耗，但由于比 **X1** 振荡器更容易受到噪音干扰，因此在使用 **XT1** 时钟时应特别注意连线方式。

备注 对于振荡电路选择和振荡器常数，顾客需要自己评价振荡特性或请振荡器厂商评估。

DC 特性 (1/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位		
<R>	I_{OH1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145, 中的每个引脚	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-3.0	mA	
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-1.0	mA	
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-1.0	mA	
		P00 ~ P04, P40 ~ P47, P120, P130, P140 ~ P145 的总和 (当占空比 = 70% #2)	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				-20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$				-10.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				-5.0	mA
		P05, P06, P10 ~ P17, P30, P31, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90 的总和 (当占空比 = 70% #2)	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				-30.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$				-19.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				-10.0	mA
		所有引脚总和 (当占空比 = 60% #2)	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				-50.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$				-29.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				-15.0	mA
<R>	I_{OH2}	P20 ~ P27 中的每个引脚	$AV_{REF0} \leq V_{DD}$			-0.1	mA	
		P110, P111 中的每个引脚	$AV_{REF1} \leq V_{DD}$			-0.1	mA	

注 1. 即使电流从 EV_{DD} 引脚流向输出引脚，设备操作时的电流值也可以得到保证。

- <R> 2. 当占空比系数为 60% 或 70% 的条件下的规范指标。
 可通过以下表达式计算已经改变占空比的输出电流值（当占空比从 70% 改变至 n%）。
 • 引脚的总输出电流 = $(I_{OH} \times 0.7) / (n \times 0.01)$
 <示例> $n = 50\%$ 并且 $I_{OH} = 20.0\text{ mA}$
 引脚的总输出电流 = $(-20.0 \times 0.7) / (50 \times 0.01) = -28.0\text{ mA}$
 但是，允许流入单个引脚的电流不会随占空比而变化。高于最大额定值的电流绝对不能流入引脚。

注意事项 P02 ~ P04, P43, P45, 和 P142 ~ P144 在 N-ch 漏极开路模式下不会输出高电平。

备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (2/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
<R>	I_{OL1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145 中的每个引脚	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		0.5	mA
		P60 ~ P63 中的每个引脚	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		15.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		3.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		2.0	mA
		P00 ~ P04, P40 ~ P47, P120, P130, P140 ~ P145 的总和 (当占空比 = 70% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		15.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		9.0	mA
		P05, P06, P10 ~ P17, P30, P31, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90 的总和 (当占空比 = 70% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		45.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		35.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		20.0	mA
		所有引脚总和 (当占空比 = 60% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		65.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		50.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		29.0	mA
<R>	I_{OL2}	P20 ~ P27 中的每个引脚	$AV_{REF0} \leq V_{DD}$		0.4	mA
		P110, P111	$AV_{REF1} \leq V_{DD}$		0.4	mA

注 1. 即使电流从输出引脚流向 EV_{SS} , V_{SS} 或 AV_{SS} 引脚, 设备操作时的电流值也可以得到保证。

2. 当占空比系数为 60% 或 70% 的条件下的规范指标。

可通过以下表达式计算已经改变占空比的输出电流值 (当占空比从 70% 改变至 n%)。

- 引脚的总输出电流 = $(I_{OL} \times 0.7) / (n \times 0.01)$

<示例> $n = 50\%$ 并且 $I_{OL} = 20.0\text{ mA}$

$$\text{引脚的总输出电流} = (20.0 \times 0.7) / (50 \times 0.01) = 28.0\text{ mA}$$

但是, 允许流入单个引脚的电流不会随占空比而变化。高于最大额定值的电流绝对不能流入引脚。

注意事项 P02 ~ P04, P43, P45 和 P142 ~ P144 在 N-ch 漏极开路模式下不会输出高电平。

备注 除非另外说明, 复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (3/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
<R>	V_{IH1}	P01, P02, P12, P13, P15, P41, P45, P52 ~ P55, P64 ~ P67, P90, P121 ~ P124, P144	$0.7V_{DD}$		V_{DD}	V	
	V_{IH2}	P00, P03 ~ P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 ~ P44, P46, P47, P50, P51, P70 ~ P77, P120, P140 ~ P143, P145, EXCLK, $\overline{\text{RESET}}$	普通输入缓冲 $0.8V_{DD}$		V_{DD}	V	
	V_{IH3}	P03, P04, P43, P44, P142, P143	TTL 输入缓冲 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2		V_{DD}	V
			TTL 输入缓冲 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		V_{DD}	V
			TTL 输入缓冲 $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.6		V_{DD}	V
	V_{IH4}	P20~P27	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	$0.7AV_{REF0}$		AV_{REF0}	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$				
<R>	V_{IH5}	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	$0.7AV_{REF1}$		AV_{REF1}	V
			$AV_{REF1} = V_{DD} < 2.7\text{ V}$				
<R>	V_{IH6}	P60~P63	$0.7V_{DD}$		6.0	V	
	V_{IH7}	FLMD0	$0.9V_{DD}^{\#}$		V_{DD}	V	

注 在 flash 存储器编程模式下，必须大于 $0.9V_{DD}$ 。

注意事项 1. 即使在 N-ch 漏极开路模式下，引脚 P02~P04, P43, P45, 和 P142~P144 的 V_{IH} 最大值为 V_{DD} 。

<R> 2. 对于 P122/EXCLK, V_{IH}/V_{IL} 根据输入端口模式或外部时钟模式不同而有所差异。在外部时钟输入模式下，确保满足 EXCLK 的 DC 特性。

备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (4/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位		
<R>	V _{IL1}	P01, P02, P12, P13, P15, P41, P45, P52 ~ P55, P64 ~ P67, P90, P121 ~ P124, P144	0		0.3V _{DD}	V		
	V _{IL2}	P00, P03 ~ P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 ~ P44, P46, P47, P50, P51, P70 ~ P77, P120, P140 ~ P143, P145, EXCLK, $\overline{\text{RESET}}$	普通输入缓冲	0	0.2V _{DD}	V		
	V _{IL3}	P03, P04, P43, P44, P142, P143	TTL 输入缓冲 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V	
			TTL 输入缓冲 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V	
			TTL 输入缓冲 $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		0.2	V	
	<R>	V _{IL4}	P20~P27	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	0		0.3AV _{REF0}	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$					
<R>	V _{IL5}	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	0		0.3AV _{REF1}	V	
$AV_{REF1} = V_{DD} < 2.7\text{ V}$								
	V _{IL6}	P60~P63	0		0.3V _{DD}	V		
	V _{IL7}	FLMD0 ^注	0		0.1V _{DD}	V		

注 当禁止 flash 存储器写入时，将 FLMD0 引脚直接连接到 V_{SS}，并且保持电压低于 0.1V_{DD}。

- 注意事项
- 即使在 N-ch 漏极开路模式下，引脚 P02~P04, P43, P45 和 P142~P144 的 V_{IH} 最大值为 V_{DD}。
 - 对于 P122/EXCLK, V_{IH} 和 V_{IL} 的值根据输入端口模式或外部时钟模式不同而有所差异。在外部时钟输入模式下，确保满足 EXCLK 的 DC 特性。

备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (5/12)

<R> (TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, 1.8 V ≤ AVREF0 ≤ VDD, 1.8 V ≤ AVREF1 ≤ VDD, VSS = EVSS = AVSS = 0 V)

项目	符号	条件	MIN.	TYP.	MAX.	单位		
<R>	VOH1	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145,	4.0 V ≤ VDD ≤ 5.5 V, IOH1 = -3.0 mA	VDD - 0.7			V	
			1.8 V ≤ VDD ≤ 5.5 V, IOH1 = -1.0 mA	VDD - 0.5			V	
	VOH2	P20~P27	AVREF0 ≤ VDD, IOH2 = -0.1 mA	AVREF0 - 0.5			V	
		P110, P111	AVREF1 ≤ VDD, IOH2 = -0.1 mA	AVREF1 - 0.5			V	
<R>	VOL1	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P130, P140 ~ P145,	4.0 V ≤ VDD ≤ 5.5 V, IOL1 = 8.5 mA			0.7	V	
			2.7 V ≤ VDD ≤ 5.5 V, IOL1 = 1.0 mA			0.5	V	
			1.8 V ≤ VDD ≤ 5.5 V, IOL1 = 0.5 mA			0.4	V	
	VOL2	P20~P27	AVREF0 ≤ VDD, IOL2 = 0.4 mA			0.4	V	
				P110, P111	AVREF1 ≤ VDD, IOL2 = 0.4 mA			0.4
	VOL3	P60~P63		4.0 V ≤ VDD ≤ 5.5 V, IOL1 = 15.0 mA			2.0	V
				4.0 V ≤ VDD ≤ 5.5 V, IOL1 = 5.0 mA			0.4	V
				2.7 V ≤ VDD ≤ 5.5 V, IOL1 = 3.0 mA			0.4	V
				1.8 V ≤ VDD ≤ 5.5 V, IOL1 = 2.0 mA			0.4	V

备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (6/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位		
<R>	I _{LIH1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90, P120, P140 ~ P145, FLMD0, /RESET	V _I = V _{DD}		1	μA		
			I _{LIH2}	P20~P27	V _I = AV _{REF0} , 2.7 V ≤ AV _{REF0} ≤ V _{DD}		1	μA
					V _I = AV _{REF0} , AV _{REF0} = V _{DD} < 2.7 V			
			I _{LIH3}	P110, P111	V _I = AV _{REF1} , 2.7 V ≤ AV _{REF1} ≤ V _{DD}		1	μA
V _I = AV _{REF1} , AV _{REF1} = V _{DD} < 2.7 V								
<R>	I _{LIL1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90, P120, P140 ~ P145, FLMD0, /RESET	V _I = V _{SS}		-1	μA		
			I _{LIL2}	P20~P27	V _I = V _{SS} , 2.7 V ≤ AV _{REF0} ≤ V _{DD}		-1	μA
					V _I = V _{SS} , AV _{REF0} = V _{DD} < 2.7 V			
			I _{LIL3}	P110, P111	V _I = V _{SS} , 2.7 V ≤ AV _{REF1} ≤ V _{DD}		-1	μA
V _I = V _{SS} , AV _{REF1} = V _{DD} < 2.7 V								
<R>	I _{LIL4}	P121~P124 (X1, X2, XT1, XT2)	V _I = V _{DD}		1	μA		
			在输入端口 在振荡器连接		10	μA		
<R>	I _{LIL1}	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P60 ~ P67, P70 ~ P77, P90, P120, P140 ~ P145, FLMD0, /RESET	V _I = V _{SS}		-1	μA		
			I _{LIL2}	P20~P27	V _I = V _{SS} , 2.7 V ≤ AV _{REF0} ≤ V _{DD}		-1	μA
					V _I = V _{SS} , AV _{REF0} = V _{DD} < 2.7 V			
			I _{LIL3}	P110, P111	V _I = V _{SS} , 2.7 V ≤ AV _{REF1} ≤ V _{DD}		-1	μA
V _I = V _{SS} , AV _{REF1} = V _{DD} < 2.7 V								
<R>	I _{LIL4}	P121~P124 (X1, X2, XT1, XT2)	V _I = V _{SS}		-1	μA		
			在输入端口 在振荡器连接		-10	μA		

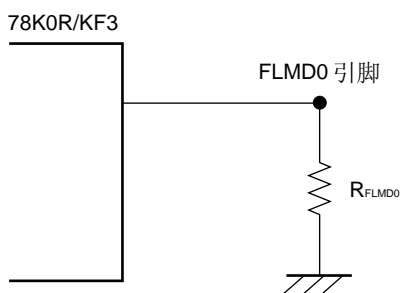
备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (7/12)

<R> (TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, 1.8 V ≤ AVREF0 ≤ VDD, 1.8 V ≤ AVREF1 ≤ VDD, VSS = EVSS = AVSS = 0 V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
片内上拉电阻	R _U	P00 ~ P06, P10 ~ P17, P30, P31, P40 ~ P47, P50 ~ P55, P64 ~ P67, P70 ~ P77, P90, P120, P140 ~ P145 VI = VSS, 在输入端口	10	20	100	kΩ
FLMD0 引脚外部下拉电阻 [※]	R _{FLMD0}	当使用软件允许自编程模式设置时	100			kΩ

<R> **注** 推荐 FLMD0 引脚悬空。如果引脚需要从外部下拉，设置 R_{FLMD0} 最少为 100 kΩ。



备注 除非另外说明，复用功能引脚的特性与那些端口引脚的特性相同。

DC 特性 (8/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8 \text{ V} \leq V_{DD} = EV_{DD} \leq 5.5 \text{ V}$, $1.8 \text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8 \text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$)

<R>

参数	符号	条件		MIN.	TYP.	MAX.	单位		
供电电流	I _{DD1} ^{注1}	操作模式	f _{MX} = 20 MHz ^{注2} , V _{DD} = 5.0 V	方波输入		8.2	12.2	mA	
				振荡器连接		8.5	12.5	mA	
			f _{MX} = 20 MHz ^{注2} , V _{DD} = 3.0 V	方波输入		8.2	12.2	mA	
				振荡器连接		8.5	12.5	mA	
			f _{MX} = 10 MHz ^{注2, 3} , V _{DD} = 5.0 V	方波输入		3.9	6.2	mA	
				振荡器连接		4.0	6.3	mA	
			f _{MX} = 10 MHz ^{注2, 3} , V _{DD} = 3.0 V	方波输入		3.9	6.2	mA	
				振荡器连接		4.0	6.3	mA	
			f _{MX} = 5 MHz ^{注2, 3} , V _{DD} = 3.0 V	普通电流模式	方波输入		2.1	3.0	mA
					振荡器连接		2.2	3.1	mA
			低功耗电流模式 ^{注4}	方波输入		1.5	2.1	mA	
				振荡器连接		1.5	2.1	mA	
			f _{MX} = 5 MHz ^{注2, 3} , V _{DD} = 2.0 V	普通电流模式	方波输入		1.5	2.1	mA
					振荡器连接		1.5	2.1	mA
	低功耗电流模式 ^{注4}	方波输入		1.4	2.0	mA			
		振荡器连接		1.4	2.0	mA			
	f _{IH} = 8 MHz ^{注5}	V _{DD} = 5.0 V		3.5	5.0	mA			
		V _{DD} = 3.0 V		3.5	5.0	mA			

- 注
1. 流入 V_{DD}, EV_{DD}, AV_{REF0}, 和 AV_{REF1} 的总电流, 包括输入引脚的电平恒为 V_{DD} 或 V_{SS} 时的输入漏电流。“MAX.”栏中的值包括外围设备操作电流。但是不包括流经 A/D 转换器, D/A 转换器, LVI 电路, I/O 端口, 和片内上拉/下拉电阻的电流。
 2. 当内部高速振荡器和副系统时钟都停止时。
 3. 当 AMPH (时钟操作模式控制寄存器 (CMC) 的第 0 位) = 0 和 FSEL (操作速度模式控制寄存器 (OSMC) 的第 0 位) = 0 时。
 4. 当 RMC 寄存器被设置为 5AH 时。
 5. 当高速系统时钟和副系统时钟都停止时。

- 备注
1. f_{MX}: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)
 2. f_{IH}: 内部高速振荡时钟频率
 3. 关于稳压器输出电压的普通电流模式和低功耗电流模式的详细情况, 参见第二十二章 稳压器。

DC 特性 (9/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

<R>

参数	符号	条件		MIN.	TYP.	MAX.	单位	
供电电流	$I_{DD1}^{\#1}$	操作 模式	$f_{SUB} = 32.768\text{ kHz}^{\#2}$ $T_A = -40 \sim +70^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		8.0	24.0	μA
				$V_{DD} = 3.0\text{ V}$		8.0	24.0	μA
				$V_{DD} = 2.0\text{ V}$		7.0	21.0	μA
		$f_{SUB} = 32.768\text{ kHz}^{\#2}$ $T_A = -40 \sim +85^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		8.0	31.0	μA	
			$V_{DD} = 3.0\text{ V}$		8.0	31.0	μA	
			$V_{DD} = 2.0\text{ V}$		7.0	28.0	μA	

- 注**
1. 流入 V_{DD} , EV_{DD} , AV_{REF0} , 和 AV_{REF1} 的总电流, 包括输入引脚的电平恒为 V_{DD} 或 V_{SS} 时的输入漏电流。
“MAX.”栏中的值包括外围设备操作电流。但是不包括流经 A/D 转换器, D/A 转换器, LVI 电路, I/O 端口, 和片内上拉/下拉电阻的电流。
 2. 当内部高速振荡器和高速系统时钟都停止时。当看门狗定时器停止时。

备注 f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)

DC 特性 (10/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

<R>

参数	符号	条件		MIN.	TYP.	MAX.	单位		
供电电流	$I_{DD2}^{\#1}$	HALT 模式	$f_{MX} = 20\text{ MHz}^{\#2}$, $V_{DD} = 5.0\text{ V}$	方波输入		1.1	2.7	mA	
				振荡器连接		1.4	3.0	mA	
			$f_{MX} = 20\text{ MHz}^{\#2}$, $V_{DD} = 3.0\text{ V}$	方波输入		1.1	2.7	mA	
				振荡器连接		1.4	3.0	mA	
			$f_{MX} = 10\text{ MHz}^{\#2, 3}$, $V_{DD} = 5.0\text{ V}$	方波输入		0.65	1.4	mA	
				振荡器连接		0.75	1.5	mA	
			$f_{MX} = 10\text{ MHz}^{\#2, 3}$, $V_{DD} = 3.0\text{ V}$	方波输入		0.65	1.4	mA	
				振荡器连接		0.75	1.5	mA	
			$f_{MX} = 5\text{ MHz}^{\#2, 3}$, $V_{DD} = 3.0\text{ V}$	普通电流模式	方波输入		0.39	0.75	mA
					振荡器连接		0.44	0.8	mA
				低功耗电流模式 ^{#4}	方波输入		0.3	0.5	mA
					振荡器连接		0.35	0.55	mA
			$f_{MX} = 5\text{ MHz}^{\#2, 3}$, $V_{DD} = 2.0\text{ V}$	普通电流模式	方波输入		0.3	0.5	mA
					振荡器连接		0.35	0.55	mA
低功耗电流模式 ^{#4}	方波输入			0.3	0.5	mA			
	振荡器连接			0.35	0.55	mA			
$f_{IH} = 8\text{ MHz}^{\#5}$		$V_{DD} = 5.0\text{ V}$		0.45	0.6	mA			
		$V_{DD} = 3.0\text{ V}$		0.45	0.6	mA			

- 注
1. 流入 V_{DD} , EV_{DD} , AV_{REF0} , 和 AV_{REF1} 的总电流, 包括输入引脚的电平恒为 V_{DD} 或 V_{SS} 时的输入漏电流。
“MAX.”栏中的值包括外围设备操作电流。但是不包括流经 A/D 转换器, D/A 转换器, LVI 电路, I/O 端口, 和片内上拉/下拉电阻的电流。在 HALT 指令期间由 flash 存储器执行。
 2. 当内部高速振荡器和副系统时钟都停止时。
 3. 当 AMPH (时钟操作模式控制寄存器 (CMC) 的第 0 位) = 0 和 FSEL (操作速度模式控制寄存器 (OSMC) 的第 0 位) = 0 时。
 4. 当 RMC 寄存器被设置为 5AH 时。
 5. 当高速系统时钟和副系统时钟都停止时。

- 备注
1. f_{MX} : 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)
 2. f_{IH} : 内部高速振荡时钟频率
 3. 关于稳压器输出电压的普通电流模式和低功耗电流模式的详细情况, 参见第二十二章 稳压器。

DC 特性 (11/12)

<R> (TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, 1.8 V ≤ AVREF0 ≤ VDD, 1.8 V ≤ AVREF1 ≤ VDD, VSS = EVSS = AVSS = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位	
供电电流	IDD1 ^{※1}	HALT 模式	fSUB = 32.768 kHz ^{※2} TA = -40~+70 °C	VDD = 5.0 V		2.2	14.0	μA
				VDD = 3.0 V		2.2	14.0	μA
				VDD = 2.0 V		2.1	13.8	μA
		fSUB = 32.768 kHz ^{※2} TA = -40~+85 °C	VDD = 5.0 V		2.2	21.0	μA	
			VDD = 3.0 V		2.2	21.0	μA	
			VDD = 2.0 V		2.1	20.8	μA	
	IDD3 ^{※3}	STOP 模式	TA = -40~+70 °C		1.1	9.0	μA	
			TA = -40~+85 °C		1.1	16.0	μA	

- 注
1. 流入 VDD, EVDD, AVREF0, 和 AVREF1 的总电流, 包括输入引脚的电平恒为 VDD 或 VSS 时的输入漏电流。
“MAX.”栏中的值包括外围设备操作电流。但是不包括流经 A/D 转换器, D/A 转换器, LVI 电路, I/O 端口, 和片内上拉/下拉电阻的电流。在 HALT 指令期间由 flash 存储器执行。
 2. 当内部高速振荡器和高速系统时钟都停止时。当看门狗定时器停止时。
 3. 流入 VDD, EVDD, AVREF0, 和 AVREF1 的总电流, 包括输入引脚的电平恒为 VDD 或 VSS 时的输入漏电流。
当副系统时钟停止时。当看门狗定时器停止时。

备注 fSUB: 副系统时钟频率 (XT1 时钟振荡频率)

DC 特性 (12/12)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

<R>

参数	符号	条件	MIN.	TYP.	MAX.	单位	
<R> RTC 操作电流	I_{RTC} 注 1, 2	$f_{SUB} = 32.768\text{ kHz}$	$V_{DD} = 3.0\text{ V}$		0.2	1.0	μA
			$V_{DD} = 2.0\text{ V}$		0.2	1.0	
<R> 看门狗定时器 操作电流	I_{WDT} 注 2, 3	$f_{IL} = 240\text{ kHz}$		5	10	μA	
<R> A/D 转换器 操作电流	I_{ADC} 注 4	以最大速度转换期间 $2.3\text{ V} \leq AV_{REF0}$		0.86	1.9	mA	
<R> D/A 转换器 操作电流	I_{DAC} 注 5	每一个通道		1.0	2.5	mA	
<R> LVI 操作电流	I_{LVI} 注 6			9	18	μA	

- 注
1. 电流只经过实时计数器 (XT1 振荡器的操作电流除外)。78K0R/KF3 的电流值是典型 (TYP.) 值, 就是在操作模式或 HALT 模式下 I_{DD1} 或 I_{DD2} , 与 I_{RTC} 的典型值的总和。 I_{DD1} 和 I_{DD2} 的 MAX. 值也包括实时计数器操作电流。
 2. 当内部高速振荡器和高速系统时钟都停止时。
 3. 电流只流经看门狗定时器 (包括 240 kHz 内部振荡器的操作电流)。78K0R/KF3 的电流值是 $f_{CLK} = f_{SUB}/2$ 时或当看门狗定时器工作在 STOP 模式下的 I_{DD1} 、 I_{DD2} 或 I_{DD3} 与 I_{WDT} 的总和。
 4. 电流只流经 A/D 转换器 (AV_{REF0} 引脚)。78K0R/KF3 的电流值是当 A/D 转换器在操作模式或 HALT 模式时 I_{DD1} 或 I_{DD2} 与 I_{ADC} 的总和。
 5. 电流只流经 D/A 转换器 (AV_{REF1} 引脚)。78K0R/KF3 的电流值是当 D/A 转换器在操作模式或 HALT 模式时 I_{DD1} 或 I_{DD2} 与 I_{DAC} 的总和。
 6. 电流只流经 LVI 电路。78K0R/KF3 电流值是当 LVI 电路在操作模式、HALT 模式或 STOP 模式时 I_{DD1} 、 I_{DD2} 或 I_{DD3} 与 I_{LVI} 的和。

备注

1. f_{IL} : 内部低速振荡时钟频率
2. f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)
3. f_{CLK} : CPU/ 外围硬件时钟频率

AC 特性

(1) 基本操作 (1/6)

<R> ($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$,
 $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件		MIN.	TYP.	MAX.	单位	
指令周期 (最小指令执行时间)	T_{CY}	主系统时钟 (f_{XP}) 操作	普通电流 模式	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		8	μs
				$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.2		8	μs
			低功耗电流模式		0.2		8	μs
		副系统时钟 (f_{SUB}) 操作		57.2	61	62.5	μs	
		在自编程模 式下	普通电流 模式	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		0.5	μs
<R> 外部主系统时钟频率	f_{EX}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2.0		20.0	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		2.0		5.0	MHz	
<R> 外部主系统时钟输入高电平宽度, 低电平宽度	t_{EXH}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		24			ns	
	t_{EXL}	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		96			ns	
$TI00 \sim TI07$ 输入高电平宽度, 低电平宽度	t_{TIH} , t_{TIL}			$1/f_{MCK} + 10$			ns	
TO00~TO07 输出频率	f_{TO}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
PCLBUZ0/1 输出频率	f_{PCL}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
中断输入高电平宽度, 低电平宽度	t_{INTH} , t_{INTL}			1			μs	
按键中断输入低电平宽度	t_{KR}			250			ns	
<R> $\overline{\text{RESET}}$ 低电平宽度	t_{RSL}			10			μs	

备注

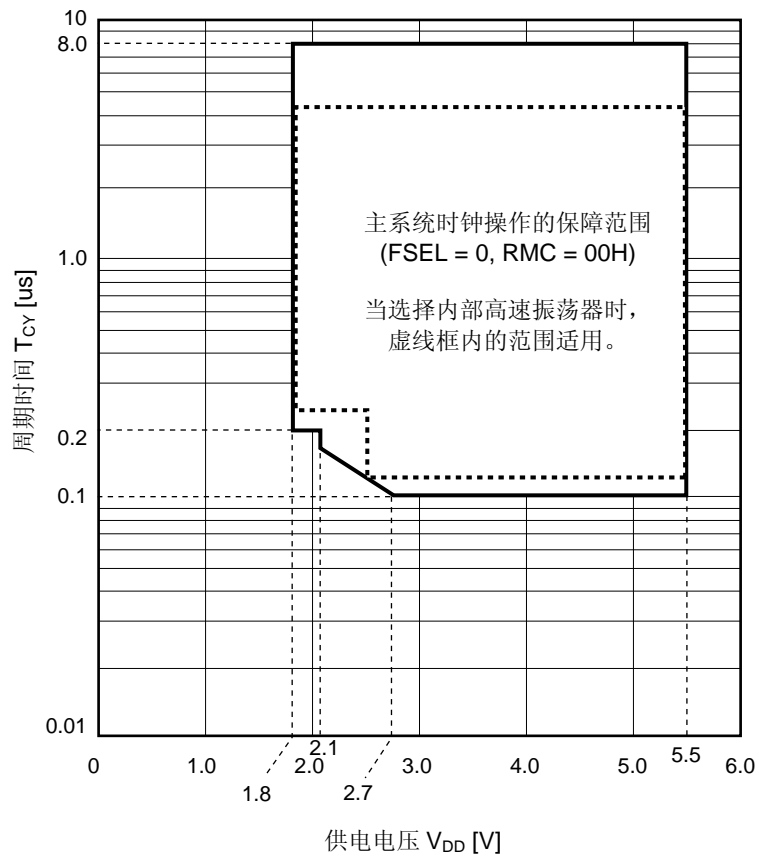
1. f_{MCK} : 定时器阵列单元操作时钟频率

(由 TMR0n 寄存器的 CKS0n 位设置的操作时钟。n: 通道编号 (n = 0 ~ 7))

2. 关于稳压器输出电压的普通电流模式和低功耗电流模式的详细情况, 参见第二十二章 稳压器。

(1) 基本操作 (2/6)

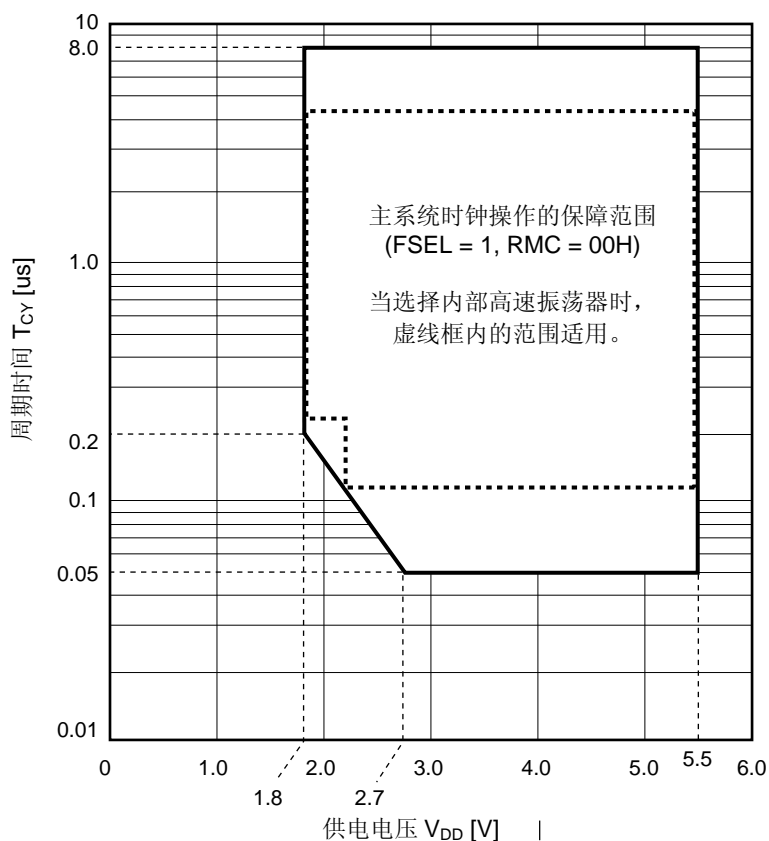
<R> 在主系统时钟操作期间的最短指令执行时间 (FSEL = 0, RMC = 00H)



备注 FSEL: 操作速度模式控制寄存器 (OSMC) 的第 0 位。

(1) 基本操作 (3/6)

<R> 在主系统时钟操作期间的最短指令执行时间 (FSEL = 1, RMC = 00H)



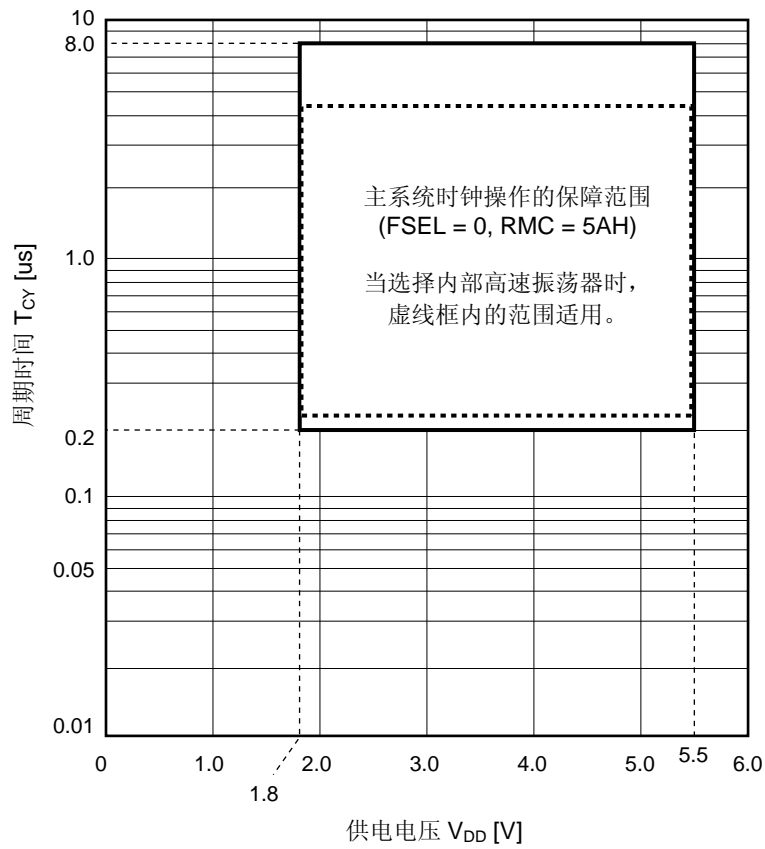
注意 当 V_{DD} 低于 2.25 V 时, 禁止进行下列操作

- 将 FSEL 从 0 修改为 1。
- 当 FSEL 设置为 1 时, 在 f_{EX} 或 f_{IH} 操作期间释放 STOP 模式。
(即使已经进行了分频, 也一定不要执行。STOP 模式可能在 f_x 操作期间释放。)
- 当 FSEL 设置为 1 时, f_{CLK} 的操作从 f_{SUB} 切换到 f_{MAIN} 。
(即使已经进行了分频, 也一定不要执行。)

- 备注**
1. FSEL: 操作速度模式控制寄存器 (OSMC) 的第 0 位。
 2. f_x : X1 时钟振荡频率
 f_{IH} : 内部高速振荡时钟频率
 f_{EX} : 外部主系统时钟频率
 f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率
 f_{CLK} : CPU/ 外围硬件时钟频率

(1) 基本操作 (4/6)

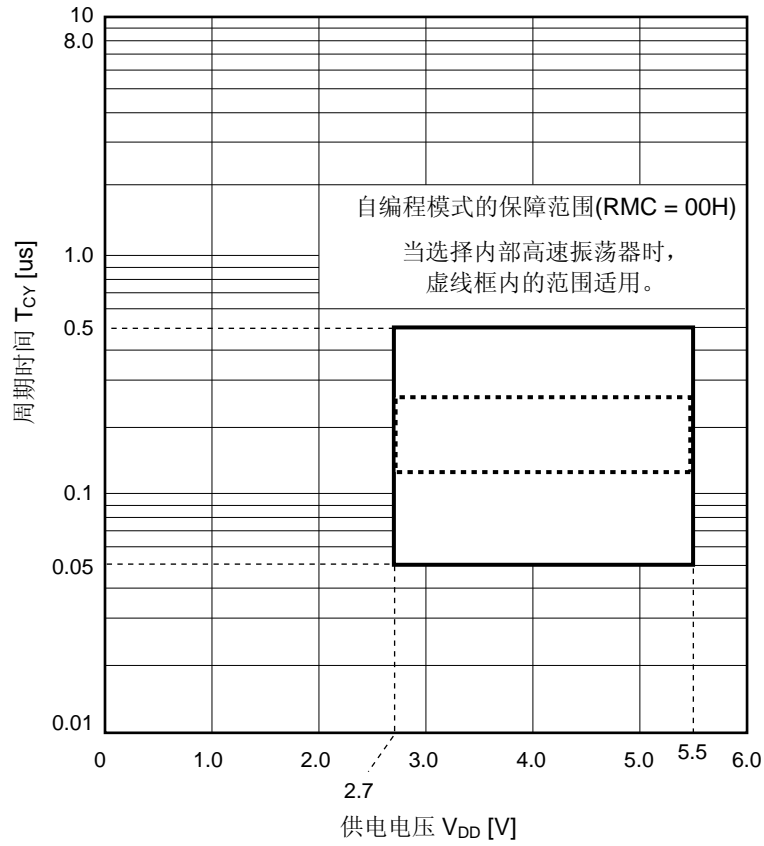
<R> 在主系统时钟操作期间的最短指令执行时间 (FSEL = 0, RMC = 54H)



- 备注
1. FSEL: 操作速度模式控制寄存器 (OSMC) 的第 0 位。
 2. 当 RMC 被设置为 54H 时, 整个电压范围为 5 MHz (MAX.)

(1) 基本操作 (5/6)

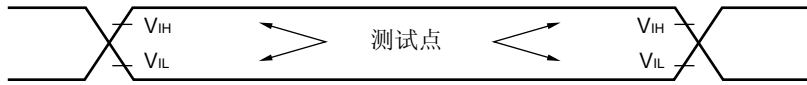
<R> 在自编程模式下的最短指令执行时间 (RMC = 00H)



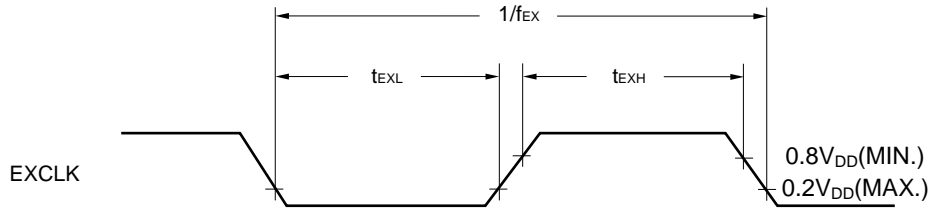
- 备注
1. FSEL: 操作速度模式控制寄存器 (OSMC) 的第 0 位。
 2. 当 RMC 被设置为 54H 或者 CPU 使用副系统时钟时, 自编程功能无法使用。

(1) 基本操作 (6/6)

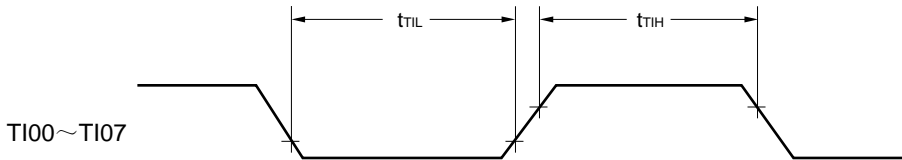
<R> AC 时序测试点



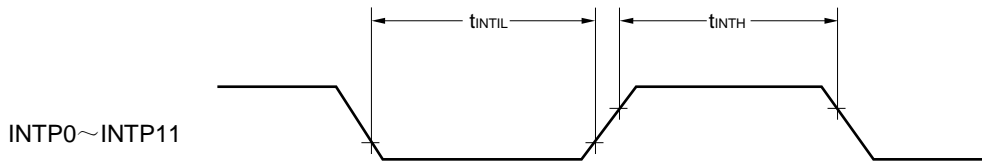
外部主系统时钟时序



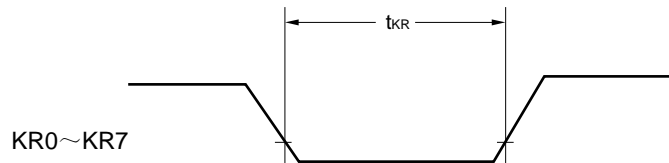
TI 时序



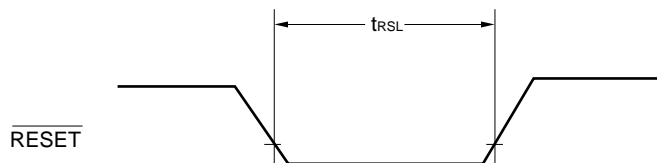
中断请求输入时序



按键中断输入时序



$\overline{\text{RESET}}$ 输入时序

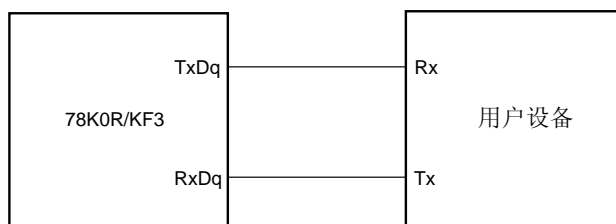


<R> (2) 串行接口：串行阵列单元 (1/17)
 ($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

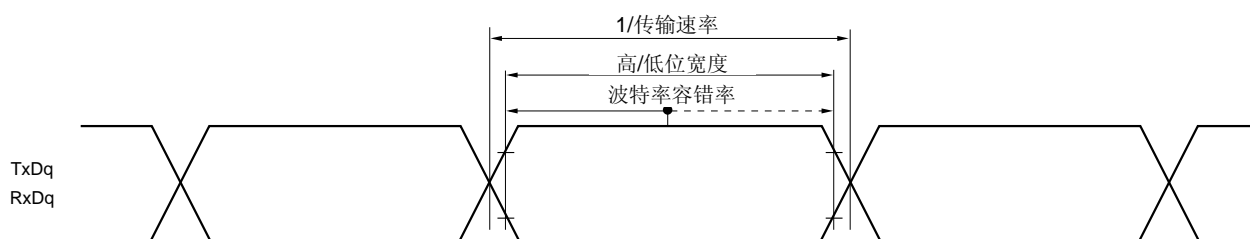
(a) 在通讯期间电位相同 (UART 模式) (专用波特率发生器输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
发送速率					$f_{MCK}/6$	bps
		$f_{CLK} = 20\text{ MHz}$, $f_{MCK} = f_{CLK}$			3.3	Mbps

UART 模式连接图 (在通讯期间电位相同)



UART 模式位宽度 (在通讯期间电位相同) (参考)



注意事项 用 PIMg 和 POMg 寄存器为 RxDi 选择普通输入缓存，为 TxDi 选择普通输出模式。

- 备注
- q: UART 编号 ($q = 0 \sim 3$), g: PIM 和 POM 编号 ($g = 0, 14$),
 i: UART 编号, 可以为其选择在通讯期间使用不同电位 ($i = 1, 2$)
 - f_{MCK} : 串行阵列单元操作时钟频率
 (通过 SMRmn 寄存器的 CKSmn 位设置操作时钟。m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0 \sim 3$))

<R>

(3) 串行接口：串行阵列单元 (2/17)

 $(T_A = -40 \sim +85^\circ\text{C}, 1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}, V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V})$ (b) 在通讯期间电位相同 (CSI 模式) (主设备模式, $\overline{\text{SCKp}}$... 内部时钟输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	200			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	400			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	800			ns
$\overline{\text{SCKp}}$ 高/低电平宽度	$t_{\text{KH1}},$ t_{KL1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{\text{KCY1}}/2 - 20$			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$t_{\text{KCY1}}/2 - 35$			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$t_{\text{KCY1}}/2 - 80$			ns
Slp 建立时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注1}	t_{SIK1}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	70			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	100			ns
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	190			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注2}	t_{KS1}		30			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出 ^{注3} 的延迟时间	t_{KSO1}	$C = 50\text{ pF}$ ^{注4}			40	ns

- 注
1. 当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 0$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 1$ 时。当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 1$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 0$ 时, Slp 建立时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”。
 2. 当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 0$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 1$ 时。当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 1$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 0$ 时, Slp 保持时间变为“从 $\overline{\text{SCKp}}\downarrow$ ”。
 3. 当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 0$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 1$ 。当 $DAP_{mn} = 0$ 且 $CKP_{mn} = 1$ 时, 或当 $DAP_{mn} = 1$ 且 $CKP_{mn} = 0$ 时, 到 SOp 输出的延迟时间变为“从 $\overline{\text{SCKp}}\uparrow$ ”。
 4. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 用 PIMg 和 POMg 寄存器为 SIj 选择普通输入缓存, 为 SOj 和 SCKj 选择普通输出模式。

- 备注
1. p: CSI 编号 (p = 00, 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)
j: CSI 编号, 可以为选择其在通讯期间使用不同电位 (j = 01, 10, 20)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)

<R>

(3) 串行接口：串行阵列单元 (3/17)

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

(c) 在通讯期间电位相同 (CSI 模式) (从设备模式, $\overline{\text{SCKp}}$ 外部时钟输入)

参数	符号	条件	MIN.	TYP.	MAX.	单位	
$\overline{\text{SCKp}}$ 周期时间	tkCY2	16 MHz < fMCK	8/fMCK			ns	
		fMCK ≤ 16 MHz	6/fMCK			ns	
$\overline{\text{SCKp}}$ 高/低电平宽度	tkH2, tkL2		fkcY2/2			ns	
Sip 建立时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注1}	tsIK2		1/fkcY2+80			ns	
Sip 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注2}	tkSI2		50			ns	
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注3}	tkSO2	C = 50 pF ^{注4}	4.0 V ≤ VDD ≤ 5.5 V			1/fMCK+120	ns
			2.7 V ≤ VDD < 4.0 V			1/fMCK+120	ns
			1.8 V ≤ VDD < 2.7 V			1/fMCK+180	ns

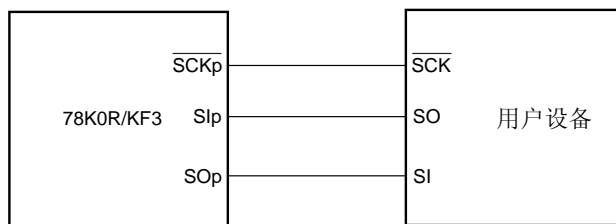
- 注
1. 当 DAPmn = 0 且 CKPmn = 0 时, 或当 DAPmn = 1 且 CKPmn = 1 时。当 DAPmn = 0 且 CKPmn = 1 时, 或当 DAPmn = 1 且 CKPmn = 0 时, Sip 建立时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”。
 2. 当 DAPmn = 0 且 CKPmn = 0 时, 或当 DAPmn = 1 且 CKPmn = 1 时。当 DAPmn = 0 且 CKPmn = 1 时, 或当 DAPmn = 1 且 CKPmn = 0 时, Sip 保持时间变为“从 $\overline{\text{SCKp}}\downarrow$ ”。
 3. 当 DAPmn = 0 且 CKPmn = 0 时, 或当 DAPmn = 1 且 CKPmn = 1。当 DAPmn = 0 且 CKPmn = 1 时, 或当 DAPmn = 1 且 CKPmn = 0 时, 到 SOp 输出的延迟时间变为“从 $\overline{\text{SCKp}}\uparrow$ ”。
 4. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 用 PIMg 和 POMg 寄存器为 SIj 和 $\overline{\text{SCKj}}$ 选择普通输入缓存, 为 SOj 选择普通输出模式。

- 备注
1. p: CSI 编号 (p = 00, 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)
j: CSI 编号, 可以为其选择在通讯期间使用不同电位 (j = 01, 10, 20)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)

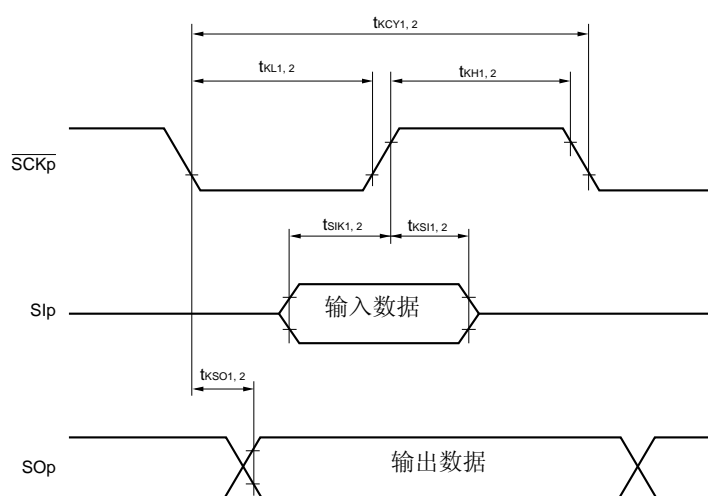
<R> (3) 串行接口：串行阵列单元 (4/17)

CSI 模式连接图 (在通讯期间电位相同)



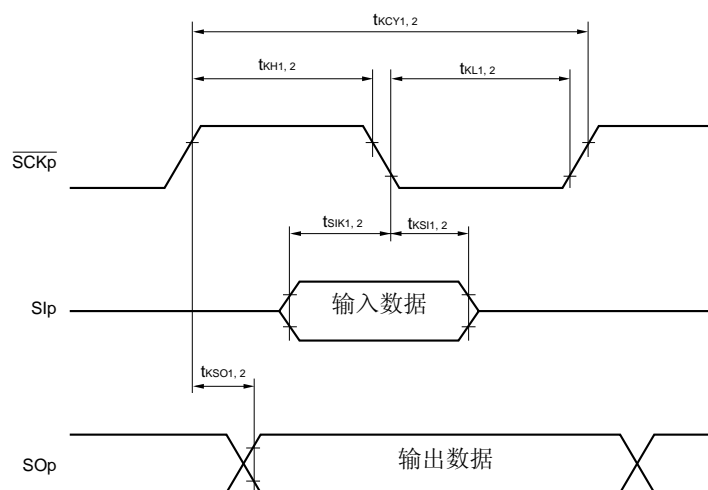
CSI 模式串行传输时序 (在通讯期间电位相同)

(当 DAPmn = 0 且 CKPmn = 0, 或当 DAPmn = 1 且 CKPmn = 1 时)



CSI 模式串行传输时序 (在通讯期间电位相同)

(当 DAPmn = 0 且 CKPmn = 1, 或当 DAPmn = 1 且 CKPmn = 0 时)



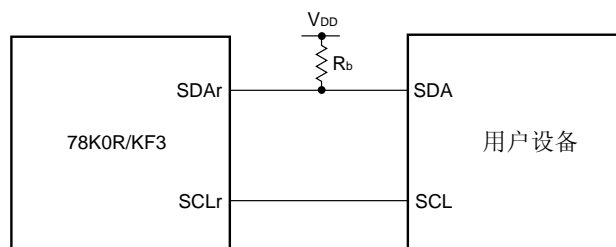
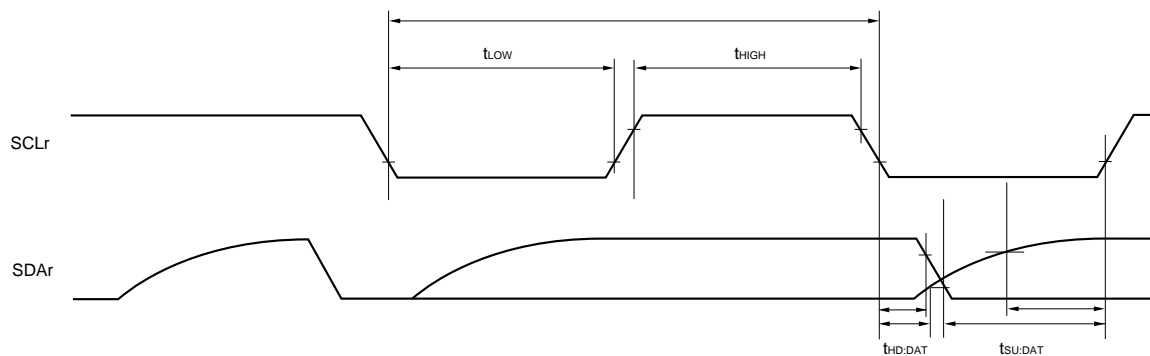
- 备注
1. p: CSI 编号 (p = 00, 01, 10, 20)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)

<R> (3) 串行接口：串行阵列单元 (5/17)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(d) 在通讯期间电位相同 (简易 I²C 模式)

参数	符号	条件	MIN.	MAX.	单位
SCLr 时钟频率	f _{SCL}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$		400	kHz
当 SCLr = "L" 时的保持时间	t _{LOW}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	995		ns
当 SCLr = "H" 时的保持时间	t _{HIGH}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	995		ns
数据建立时间 (接收)	t _{SU:DAT}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	$1/f_{MCK} + 120$		ns
数据保持时间 (发送)	t _{HD:DAT}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	0	160	ns

简易 I²C 模式连接图 (在通讯期间电位相同)简易 I²C 模式串行传输时序 (在通讯期间电位相同)

注意事项 用 PIMg 和 POMg 寄存器为 SDAr 选择普通输入缓存和 N-ch 漏极开路输出 (V_{DD} 耐压)，为 SCLr 选择普通输出模式。

备注

- $R_b[\Omega]$: 通讯线 (SDAr) 上拉阻抗
 $C_b[\text{F}]$: 通讯线 (SCLr, SDAr) 负载电容
- r: IIC 编号 (r = 10, 20), g: PIM 和 POM 编号 (g = 0, 14)
- f_{MCK}: 串行阵列单元操作时钟频率
(通过 SMRmn 寄存器的 CKSmn 位设置操作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2), mn = 02, 10)

<R> (3) 串行接口: 串行阵列单元 (6/17)

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

(e) 在通讯期间电位不同 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) (1/2)

参数	符号	条件		MIN.	TYP.	MAX.	单位
发送速率		接收	4.5 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V			fMCK/6	bps
			fCLK = 20 MHz, fMCK = fCLK			3.3	Mbps
			2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V			fMCK/6	bps
			fCLK = 20 MHz, fMCK = fCLK			3.3	Mbps

注意事项 用 PIMg 和 POMg 寄存器为 RxDq 选择 TTL 输入缓存, 为 TxDq 选择 N-ch 漏极开路输出 (VDD 耐压)。

备注

1. q: UART 编号 (q = 1, 2), g: PIM 和 POM 编号 (g = 0, 14)
2. fMCK: 串行阵列单元操作时钟频率
(通过 SMRmn 寄存器的 CKSmn 位来设置操作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3))
3. 下面的 VIH 和 VIL 是在 UART 模式下使用不同电平通信时的串行阵列单元 AC 特性。
4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V
2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V: VIH = 2.0 V, VIL = 0.5 V
4. UART0 和 UART3 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 UART1 和 UART2。

<R> (3) 串行接口：串行阵列单元 (7/17)

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(e) 在通讯期间电位不同 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) (2/2)

参数	符号	条件		MIN.	TYP.	MAX.	单位
发送速率		接收	$4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$			注 1	
			$f_{CLK} = 16.8\text{ MHz}$, $f_{MCK} = f_{CLK}$, $C_b = 50\text{ pF}$, $R_b = 1.4\text{ k}\Omega$, $V_b = 2.7\text{ V}$			2.8 ^{注2}	Mbps
			$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$			注 3	
			$f_{CLK} = 19.2\text{ MHz}$, $f_{MCK} = f_{CLK}$, $C_b = 50\text{ pF}$, $R_b = 2.7\text{ k}\Omega$, $V_b = 2.3\text{ V}$			1.2 ^{注4}	Mbps

注 1. 下列表达式和 $f_{MCK}/6$ 相比，其中较小的最大传输速率为有效的最大传输速率
当 $4.0\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$ 且 $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ 时，计算传输速率的表达式如下

$$\text{最大传输速率} = \frac{1}{\left\{ -C_b \times R_b \times \ln \left(\frac{2.2}{V_b} \right) \right\} \times 3} \text{ [bps]}$$

$$\text{波特率误差 (理论值)} = \frac{\frac{1}{\text{传输速率} \times 2} - \left\{ -C_b \times R_b \times \ln \left(1 - \frac{2.2}{V_b} \right) \right\}}{\left(\frac{1}{\text{传输速率}} \right) \times \text{已传输的位数量}} \times 100 \text{ [%]}$$

* 该值是传输和发送端之间的相对误差理论值。

2. 该值是满足“条件”栏描述的条件时如何计算的示例。参见上述的注 1 根据客户实际条件来计算最大传输速率。

3. 下列表达式和 $f_{MCK}/6$ 相比，其中较小的最大传输速率为有效的最大传输速率
当 $2.7\text{ V} \leq V_{DD} = EV_{DD} \leq 4.0\text{ V}$ 且 $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ 时，计算传输速率的表达式如下

$$\text{最大传输速率} = \frac{1}{\left\{ -C_b \times R_b \times \ln \left(\frac{2.0}{V_b} \right) \right\} \times 3} \text{ [bps]}$$

$$\text{波特率误差 (理论值)} = \frac{\frac{1}{\text{传输速率} \times 2} - \left\{ -C_b \times R_b \times \ln \left(1 - \frac{2.0}{V_b} \right) \right\}}{\left(\frac{1}{\text{传输速率}} \right) \times \text{已传输的位数量}} \times 100 \text{ [%]}$$

* 该值是传输和发送端之间的相对误差理论值。

4. 该值是满足“条件”栏描述的条件时如何计算的示例。参见上述的注 3 根据客户实际条件来计算最大传输速率。

注意事项 用 PIMg 和 POMg 寄存器为 RxDq 选择 TTL 输入缓存，为 TxDq 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

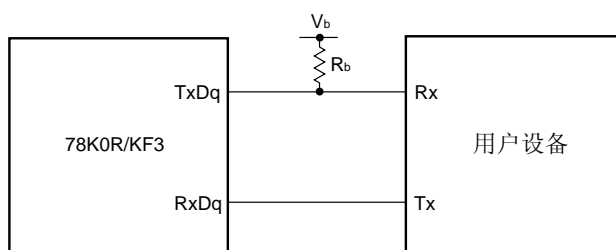
(备注在下一页列出)

<R> (3) 串行接口：串行阵列单元 (8/17)

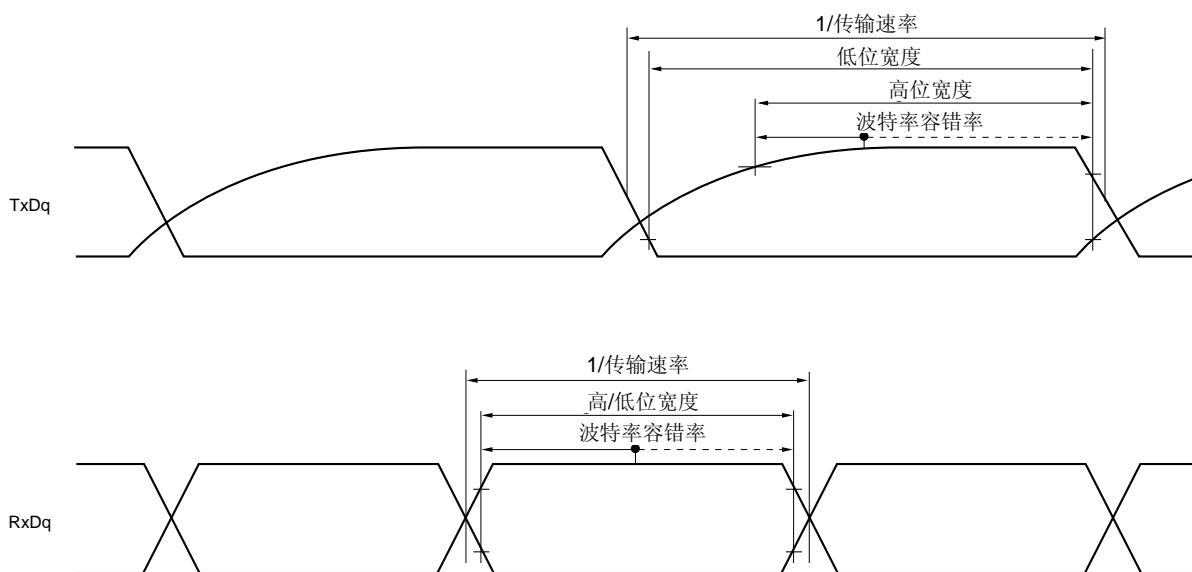
- 备注
1. $R_b[\Omega]$: 通讯线 (TxDq) 上拉阻抗
 $C_b[F]$: 通讯线 (TxDq) 负载电容, $V_b[V]$: 通讯线电压
 2. q: UART 编号 (q = 1, 2), g: PIM 和 POM 编号 (g = 0, 14)
 3. f_{MCK} : 串行阵列单元操作时钟频率
(通过 SMRmn 寄存器的 CKSmn 位来设置操作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 3))
 4. 下面的 V_{IH} 和 V_{IL} 是在 UART 模式下使用不同电平通信时的串行阵列单元 AC 特性。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
 5. UART0 和 UART3 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 UART1 和 UART2。

<R> (3) 串行接口：串行阵列单元 (9/17)

UART 模式连接图 (在通讯期间电位不同)



UART 模式位宽度 (在通讯期间电位不同)



注意事项 用 PIMg 和 POMg 寄存器为 RxDq 选择 TTL 输入缓存，为 TxDq 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

- 备注
1. R_b[Ω]: 通讯线 (TxDq) 上拉阻抗, V_b[V]: 通讯线电压
 2. q: UART 编号 (q = 1, 2), g: PIM 和 POM 编号 (g = 0, 14)
 3. UART0 和 UART3 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 UART1 和 UART2。

<R> (3) 串行接口：串行阵列单元 (10/17)

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

(f) 在通讯期间电位不同 (2.5V, 3V) (CSI 模式) (主设备模式, SCKp... 内部时钟输出) (1/2)

参数	符号	条件	MIN.	TYP.	MAX.	单位
SCKp 周期时间	tkCY1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	500			ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	1000			ns
SCKp 高电平宽度	tkH1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	tkCY1/2 - 120			ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	tkCY1/2 - 275			ns
SCKp 低电平宽度	tkL1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	tkCY1/2 - 20			
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	tkCY1/2 - 35			
Slp 建立时间 (到 SCKp↑) 注1	tSIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	195			ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	380			ns
Slp 保持时间 (从 SCKp↑) 注2	tKSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	30			ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	30			
从 SCKp↓ 到 SOp 输出注3的延迟时间	tkSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ			165	ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ			320	

注 当 DAPmn = 0 且 CKPmn = 0 时, 或当 DAPmn = 1 且 CKPmn = 1 时。

注意事项 用 PIMg 和 POMg 寄存器为 Slp 选择 TTL 输入缓存, 为 SOp 和 SCKp 选择 N-ch 漏极开路输出 (VDD 耐压)。

备注

1. p: CSI 编号 (p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)

3. Rb[Ω]: 通讯线 (SCKp, SOp) 上拉阻抗

Cb[F]: 通讯线 (Slp, SOp, SCKp) 负载电容, Vb[V]: 通讯线电压

4. 下面的 VIH 和 VIL 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性。

4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V

2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V: VIH = 2.0 V, VIL = 0.5 V

5. CSI00 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 CSI01, CSI10, 和 CSI20。

<R> (3) 串行接口：串行阵列单元 (11/17)

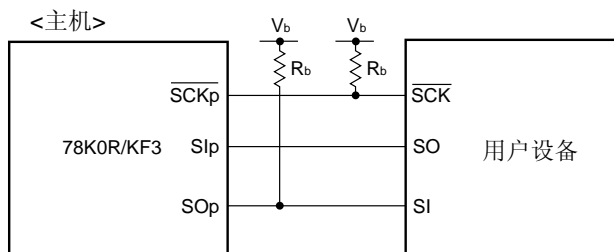
($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(f) 在通讯期间电位不同 (2.5V, 3V) (CSI 模式) (主设备模式, $\overline{\text{SCKp}}$... 内部时钟输出) (1/2)

参数	符号	条件	MIN.	TYP.	MAX.	单位
Slp 建立时间 (到 $\overline{\text{SCKp}}\uparrow$) ^注	tsIK1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	70			ns
		$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$, $2.3\text{ V} \leq V_b < 2.7\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	100			
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^注	tkSI1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	30			ns
		$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$, $2.3\text{ V} \leq V_b < 2.7\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	30			
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^注	tkSO1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 1.4\text{ k}\Omega$			40	ns
		$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$, $2.3\text{ V} \leq V_b < 2.7\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 2.7\text{ k}\Omega$			40	ns

注 当 DAPmn = 0 且 CKPmn = 1 时, 或当 DAPmn = 1 且 CKPmn = 0 时。

CSI 模式连接图 (在通讯期间电位不同)



注意事项 用 PIMg 和 POMg 寄存器为 Slp 选择 TTL 输入缓存, 为 SOp 和 $\overline{\text{SCKp}}$ 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

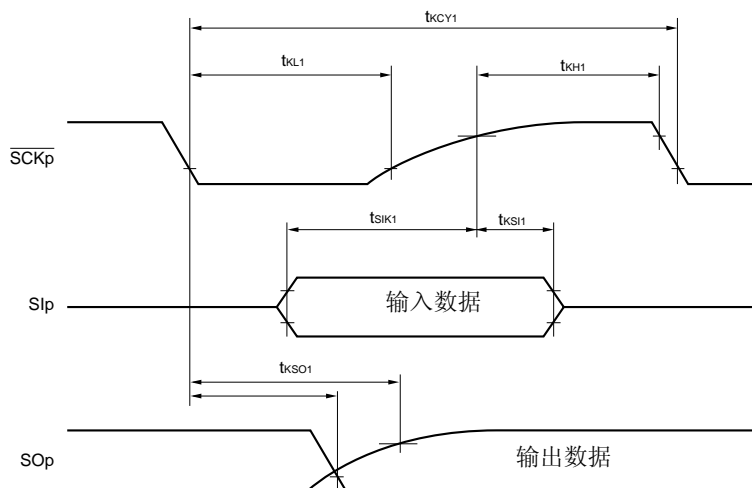
备注

1. p: CSI 编号 (p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)
3. $R_b[\Omega]$: 通讯线 ($\overline{\text{SCKp}}$, SOp) 上拉阻抗
 $C_b[\text{F}]$: 通讯线 (Slp, SOp, $\overline{\text{SCKp}}$) 负载电容, $V_b[\text{V}]$: 通讯线电压
4. 下面的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$: $V_{IH} = 2.2\text{ V}$, $V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$: $V_{IH} = 2.0\text{ V}$, $V_{IL} = 0.5\text{ V}$
5. CSI00 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 CSI01, CSI10, 和 CSI20。

<R> (3) 串行接口：串行阵列单元 (12/17)

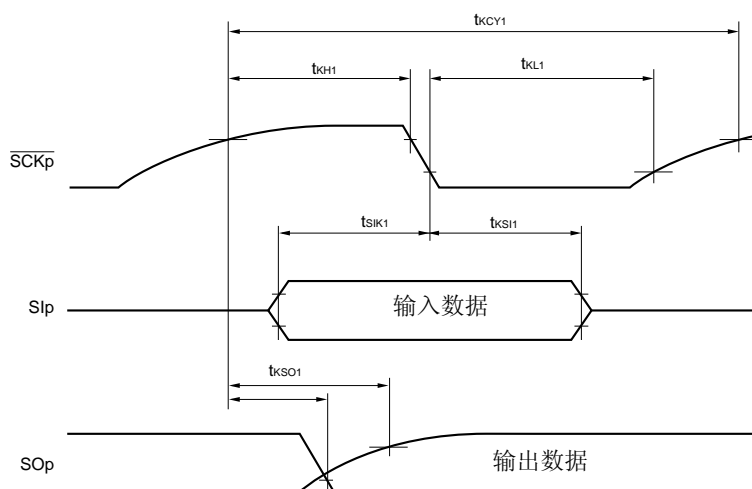
CSI 模式传输时序 (在通讯期间电位不同)

(当 DAPmn = 0 且 CKPmn = 0 时, 或 DAPmn = 1 且 CKPmn = 1)



CSI 模式传输时序 (在通讯期间电位不同)

(当 DAPmn = 0 且 CKPmn = 1 时, 或 DAPmn = 1 且 CKPmn = 0)



注意事项 用 PIMg 和 POMg 寄存器为 Slp 选择 TTL 输入缓存, 为 SOp 和 SCKp 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

备注

1. p: CSI 编号 (p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2)
3. CSI00 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 CSI01, CSI10, 和 CSI20。

<R> (3) 串行接口：串行阵列单元 (13/17)

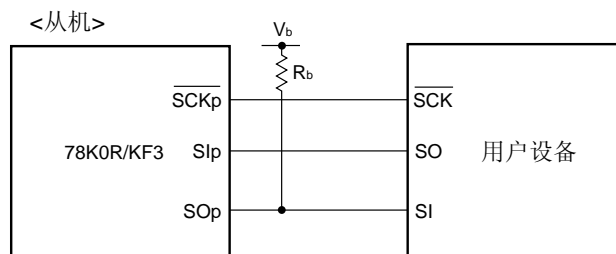
($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(g) 在通讯期间电位不同 (2.5V, 3V) (CSI 模式) (从设备模式, $\overline{\text{SCKp}}$ 外部时钟输入)

参数	符号	条件	MIN.	TYP.	MAX.	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY2}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$	$16.6\text{ MHz} < f_{\text{MCK}}$	$12/f_{\text{MCK}}$		ns
			$12.5\text{ MHz} < f_{\text{MCK}} \leq 16.6\text{ MHz}$	$10/f_{\text{MCK}}$		ns
			$8.3\text{ MHz} < f_{\text{MCK}} \leq 12.5\text{ MHz}$	$8/f_{\text{MCK}}$		ns
			$f_{\text{MCK}} \leq 8.3\text{ MHz}$	$6/f_{\text{MCK}}$		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$	$17.5\text{ MHz} < f_{\text{MCK}}$	$18/f_{\text{MCK}}$		ns
			$15\text{ MHz} < f_{\text{MCK}} \leq 17.5\text{ MHz}$	$16/f_{\text{MCK}}$		ns
			$12.5\text{ MHz} < f_{\text{MCK}} \leq 15\text{ MHz}$	$14/f_{\text{MCK}}$		ns
			$10\text{ MHz} < f_{\text{MCK}} \leq 12.5\text{ MHz}$	$12/f_{\text{MCK}}$		ns
			$7.5\text{ MHz} < f_{\text{MCK}} \leq 10\text{ MHz}$	$10/f_{\text{MCK}}$		ns
			$5\text{ MHz} < f_{\text{MCK}} \leq 7.5\text{ MHz}$	$8/f_{\text{MCK}}$		ns
$f_{\text{MCK}} \leq 5\text{ MHz}$	$6/f_{\text{MCK}}$		ns			
$\overline{\text{SCKp}}$ 高/低电平宽度	t_{KH2} , t_{KL2}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$	$f_{\text{KCY2}}/2 - 20$			ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$	$f_{\text{KCY2}}/2 - 35$			ns
$\overline{\text{Sip}}$ 建立时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注1}	t_{SIK2}		$1/f_{\text{MCK}}+90$			ns
$\overline{\text{Sip}}$ 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注2}	t_{KSI2}		50			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注3}	t_{KSO2}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 1.4\text{ k}\Omega$			$1/f_{\text{MCK}}+245$	ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 50\text{ pF}$, $R_b = 2.7\text{ k}\Omega$			$1/f_{\text{MCK}}+400$	ns

- 注
1. 当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 0$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 1$ 时。当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 1$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 0$ 时, $\overline{\text{Sip}}$ 建立时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”。
 2. 当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 0$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 1$ 时。当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 1$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 0$ 时, $\overline{\text{Sip}}$ 保持时间变为“从 $\overline{\text{SCKp}}\downarrow$ ”。
 3. 当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 0$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 1$ 时。当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 1$ 时, 或当 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 0$ 时, 到 SOp 输出的延迟时间变为“从 $\overline{\text{SCKp}}\uparrow$ ”。

CSI 模式连接图 (在通讯期间电位不同)



(注意事项和备注在下页列出)

<R> (3) 串行接口：串行阵列单元 (14/17)

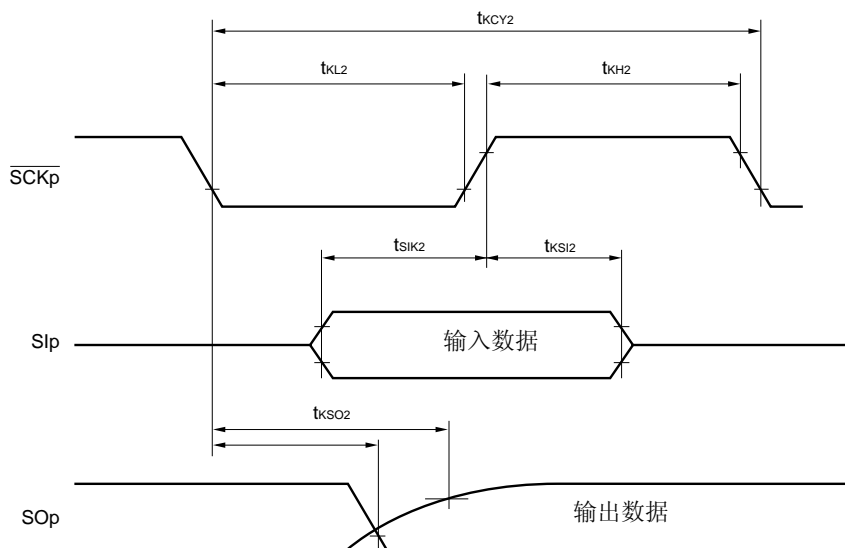
注意事项 用 PIMg 和 POMg 寄存器为 Sip 和 $\overline{\text{SCKp}}$ 选择 TTL 输入缓存，为 SOp 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

- 备注**
1. p: CSI 编号 (p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 4, 14)
 2. $R_b[\Omega]$: 通讯线 ($\overline{\text{SCKp}}$, SOp) 上拉阻抗
 $C_b[F]$: 通讯线 (Sip, SOp, $\overline{\text{SCKp}}$) 负载电容, $V_b[V]$: 通讯线电压
 3. f_{MCK} : 串行阵列单元操作时钟频率
 (通过 SMRmn 寄存器的 CKSmn 位设置操作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2))
 4. 下面的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
 5. CSI00 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 CSI01, CSI10, 和 CSI20。

<R> (3) 串行接口：串行阵列单元 (15/17)

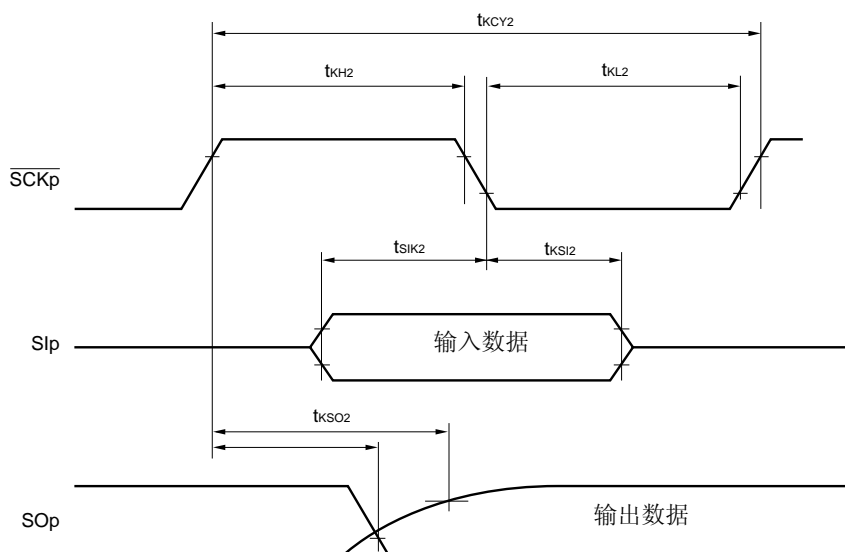
CSI 模式传输时序 (在通讯期间电位不同)

(当 $DAPmn = 0$ 且 $CKPmn = 0$ 时, 或 $DAPmn = 1$ 且 $CKPmn = 1$)



CSI 模式传输时序 (在通讯期间电位不同)

(当 $DAPmn = 0$ 且 $CKPmn = 1$ 时, 或 $DAPmn = 1$ 且 $CKPmn = 0$)



注意事项 用 PIMg 和 POMg 寄存器为 SIp 和 SCKp 选择 TTL 输入缓存, 为 SOp 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

- 备注
1. p: CSI 编号 ($p = 01, 10, 20$), g: PIM 和 POM 编号 ($g = 0, 4, 14$)
 2. m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0 \sim 2$)
 3. CSI00 不能在不同电平下通信。如果需要在不同电平下通信, 请使用 CSI01, CSI10, 和 CSI20。

<R> (3) 串行接口: 串行阵列单元 (16/17)

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

(h) 在通讯期间电位不同 (2.5V, 3V) (简易 I²C 模式)

参数	符号	条件	MIN.	MAX.	单位
SCLr 时钟频率	f _{SCL}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ		400	kHz
		2.7 V ≤ V _{DD} ≤ 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ		400	kHz
当 SCLr = "L" 时的保持时间	t _{LOW}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	1065		ns
		2.7 V ≤ V _{DD} ≤ 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	1065		ns
当 SCLr = "H" 时的保持时间	t _{HIGH}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	445		ns
		2.7 V ≤ V _{DD} ≤ 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	445		ns
数据建立时间 (接收)	t _{SU:DAT}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	1/f _{MCK} +190		ns
		2.7 V ≤ V _{DD} ≤ 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	1/f _{MCK} +190		ns
数据保持时间 (发送)	t _{HD:DAT}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	0	160	ns
		2.7 V ≤ V _{DD} ≤ 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	0	160	ns

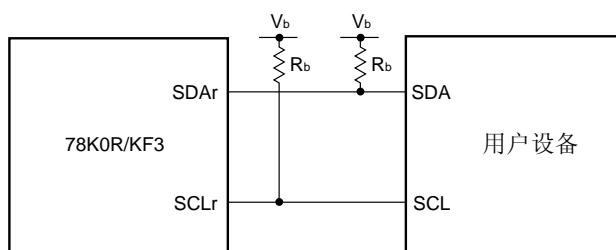
注意事项 用 PIMg 和 POMg 寄存器为 SDAr 选择 TTL 输入缓存和 N-ch 漏极开路输出 (V_{DD} 耐压), 为 SCLr 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

备注

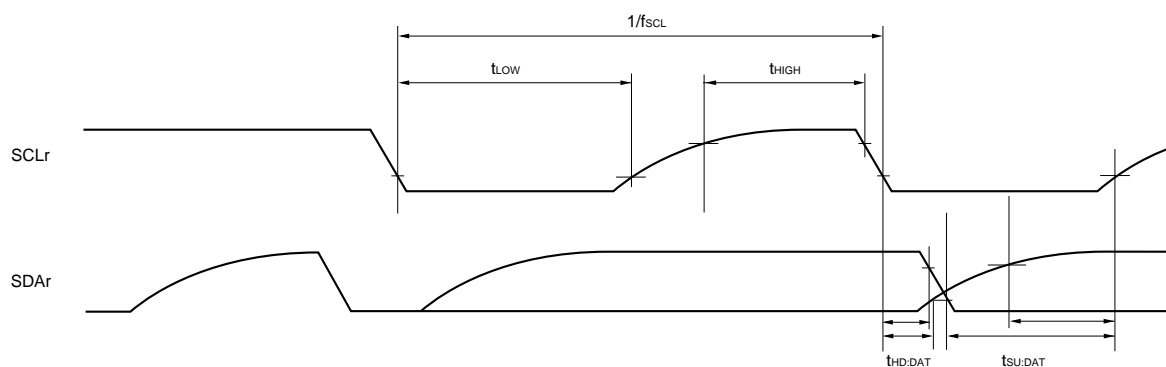
- R_b[Ω]: 通讯线 (SDAr, SCLr) 上拉阻抗
C_b[F]: 通讯线 (SDAr, SCLr) 负载电容, V_b[V]: 通讯线电压
- r: IIC 编号 (r = 10, 20), g: PIM 和 POM 编号 (g = 0, 14)
- f_{MCK}: 串行阵列单元操作时钟频率
(通过 SMRmn 寄存器的 CKSmn 位设置操作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 ~ 2), mn = 02, 10)
- 下面的 V_{IH} 和 V_{IL} 是在 I²C 模式下使用不同电平通信时的串行阵列单元 AC 特性。
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

<R> (3) 串行接口：串行阵列单元 (17/17)

简易 I²C 模式连接图 (在通讯期间电位不同)



简易 I²C 模式连接图 (在通讯期间电位不同)



注意事项 用 PIMg 和 POMg 寄存器为 SDAr 选择 TTL 输入缓存和 N-ch 漏极开路输出 (V_{DD} 耐压)，为 SCLr 选择 N-ch 漏极开路输出 (V_{DD} 耐压)。

- 备注
1. R_b[Ω]: 通讯线 (SDAr, SCLr) 上拉阻抗, V_b[V]: 通讯线电压
 2. r: IIC 编号 (r = 10, 20), g: PIM 和 POM 编号 (g = 0, 14)

(4) 串行接口: IIC0

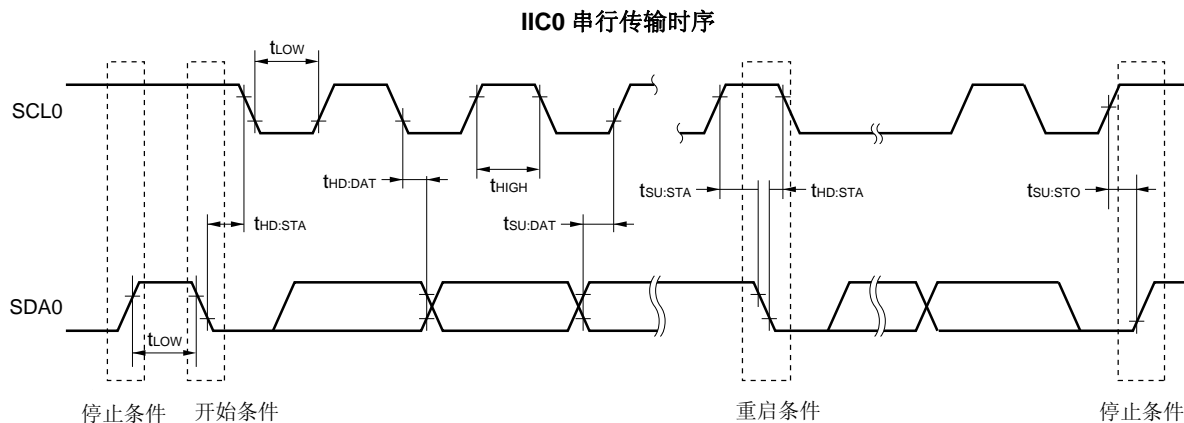
($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(a) IIC0

参数	符号	条件	标准模式		高速模式		单位
			MIN.	MAX.	MIN.	MAX.	
SCL0 时钟频率	f _{SCL}	$6.7\text{ MHz} \leq f_{\text{CLK}}$	0	100	0	400	kHz
		$4.0\text{ MHz} \leq f_{\text{CLK}} < 6.7\text{ MHz}$	0	100	0	340	kHz
		$3.2\text{ MHz} \leq f_{\text{CLK}} < 4.0\text{ MHz}$	0	100	-	-	kHz
		$2.0\text{ MHz} \leq f_{\text{CLK}} < 3.2\text{ MHz}$	0	85	-	-	kHz
重启条件的建立时间 ^{注1}	t _{SU:STA}		4.7		0.6		μs
保持时间	t _{HD:STA}		4.0		0.6		μs
当 SCL0 = "L" 时保持时间	t _{LOW}		4.7		1.3		μs
当 SCL0 = "H" 时保持时间	t _{HIGH}		4.0		0.6		μs
数据建立时间 (接收)	t _{SU:DAT}		250		100		ns
数据保持时间 (发送) ^{注2}	t _{HD:DAT}	CL00 = 1 和 CL01 = 1	0	3.45 ^{注3}	0	0.9 ^{注4}	μs
				5.50 ^{注5}		1.5 ^{注6}	μs
		CL00 = 0 和 CL01 = 0, or CL00 = 1 和 CL01 = 0	0	3.45	0	0.9 ^{注7}	μs
						0.95 ^{注8}	μs
CL00 = 0 和 CL01 = 1	0	3.45	0	0.9	μs		
停止条件的建立时间	t _{SU:STO}		4.0		0.6		μs
总线释放时间	t _{BUF}		4.7		1.3		μs

- 注
1. 当检测到启动/重启条件时, 在这个周期后第一个时钟脉冲产生。
 2. t_{HD:DAT} 的最大值 (MAX.) 在普通发送期间, 等待状态插入在 $\overline{\text{ACK}}$ (响应) 时序。
 3. 当 $3.2\text{ MHz} \leq f_{\text{CLK}} \leq 4.19\text{ MHz}$ 。
 4. 当 $6.7\text{ MHz} \leq f_{\text{CLK}} \leq 8.38\text{ MHz}$ 。
 5. 当 $2.0\text{ MHz} \leq f_{\text{CLK}} < 3.2\text{ MHz}$ 时。此时, SCL0 时钟在 85 kHz 内。
 6. 当 $4.0\text{ MHz} \leq f_{\text{CLK}} < 6.7\text{ MHz}$ 时。此时, SCL0 时钟在 340 kHz 内。
 7. 当 $8.0\text{ MHz} \leq f_{\text{CLK}} \leq 16.76\text{ MHz}$ 。
 8. 当 $7.6\text{ MHz} \leq f_{\text{CLK}} < 8.0\text{ MHz}$ 。

备注 CL00, CL01, DFC0: IIC 时钟选择寄存器 0 (IICCL0) 的第 0, 1 和 2 位



(4) 串行接口：片上调试 (UART)

(TA = -40~+85°C, 1.8 V ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = AVSS = 0 V)

(a) 片上调试 (UART)

参数	符号	条件	MIN.	TYP.	MAX.	单位
发送速率			fCLK/2 ¹²		fCLK/6	bps
		Flash 存储器编程模式			2.66	Mbps
TOOL1 输出频率	f _{TOOL1}	2.7 V ≤ V _{DD} ≤ 5.5 V			10	MHz
		1.8 V ≤ V _{DD} < 2.7 V			2.5	MHz

A/D 转换器特性

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $2.3\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES				10	位
总误差 ^{注1, 2}	AINL	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			± 1.2	%FSR
转换时间	t _{CONV}	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$	6.1		66.6	μs
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$	12.2		66.6	μs
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$	27		66.6	μs
零刻度误差 ^{注1, 2}	E _{ZS}	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			± 0.6	%FSR
满量程误差 ^{注1, 2}	E _{FS}	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			± 0.4	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			± 0.6	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			± 0.6	%FSR
积分非线性误差 ^{注1}	I _{LE}	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			± 2.5	LSB
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			± 4.5	LSB
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			± 6.5	LSB
微分非线性误差 ^{注1}	D _{LE}	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			± 1.5	LSB
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			± 2.0	LSB
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			± 2.0	LSB
模拟输入电压	V _{AIN}	$2.3\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$	AV _{SS}		AV _{REF0}	V

- 注 1. 不包括量化误差 ($\pm 1/2$ LSB)。
2. 此值表示满量程值的比率 (%FSR)。

D/A 转换器特性

($T_A = -40 \sim +85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$, $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

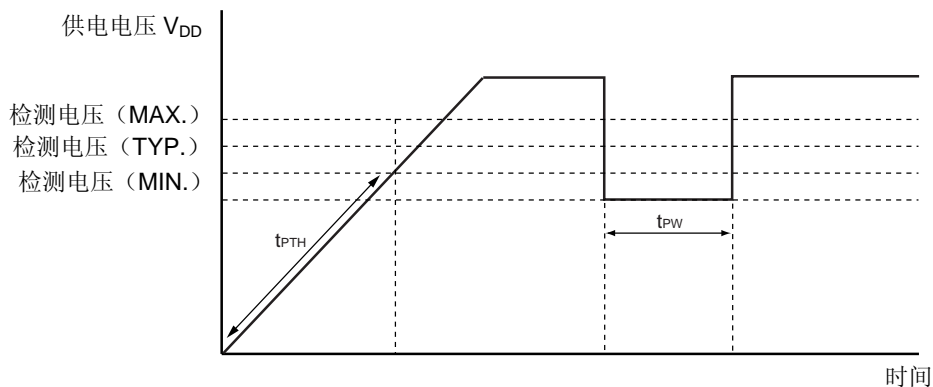
参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES				8	位
总误差	AINL	R _{LOAD} = 2 M Ω			± 1.2	%FSR
		R _{LOAD} = 4 M Ω			± 0.8	%FSR
		R _{LOAD} = 10 M Ω			± 0.6	%FSR
设置时间	t _{SET}	C _{LOAD} = 20 pF	$4.0\text{ V} \leq AV_{REF1} \leq 5.5\text{ V}$		3	μs
			$2.7\text{ V} \leq AV_{REF1} < 4.0\text{ V}$		3	μs
			$1.8\text{ V} \leq AV_{REF1} < 2.7\text{ V}$		6	μs
D/A 输出阻抗值	R _O	D/A 转换器的每个通道		6.4		k Ω

POC 电路特性 (TA = -40~+85°C, VSS = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	VPOC0		1.5	1.59	1.68	V
供电电压上升斜率	tPTH	改变 VDD 的斜率: 0 V → VPOC0	0.5			V/ms
最小脉冲宽度	tpw	当电压下降时	200			μs
检测延迟时间					200	μs

<R>

POC 电路时序



供电电压上升时间 (TA = -40~+85°C, VSS = 0 V)

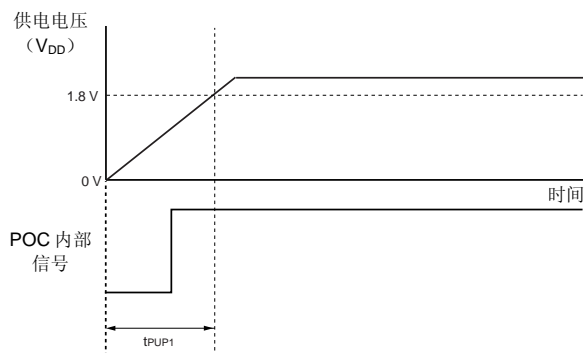
参数	符号	条件	MIN.	TYP.	MAX.	单位
上升到 1.8 V (V _{DD} (MIN.)) 的最大时间 ^注 (V _{DD} : 0 V → 1.8 V)	tPUP1	当不使用 RESET 复位输入时, 设置 LVI 默认启动功能停止 (LVIOFF (选项字节) = 1)。			3.6	ms
上升到 1.8 V (V _{DD} (MIN.)) 的最大时间 (释放 RESET 输入 → V _{DD} : 1.8 V)	tPUP2	当使用 RESET 复位输入时, 设置 LVI 默认启动功能停止 (LVIOFF (选项字节) = 1)。			1.88	ms

注 确保供电电压上升时间小于此值。

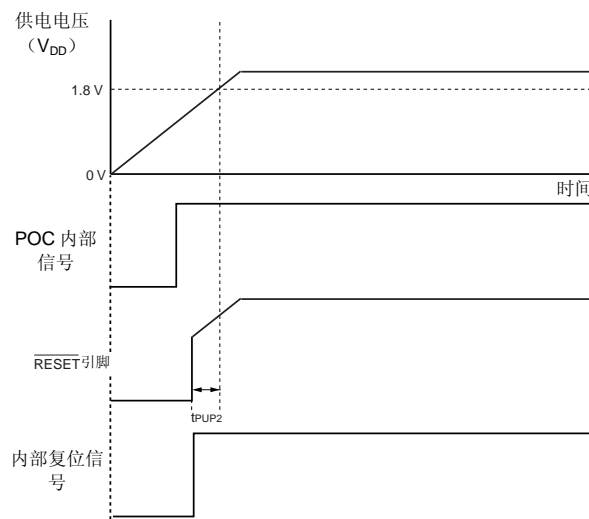
供电电压下降时间时序

<R>

• 当不使用 RESET 引脚输入时



• 当使用 RESET 引脚输入时 (在 POC 释放之后, 通过 RESET 释放外部复位,)



LVI 电路特性 (TA = -40~+85°C, VPOC ≤ VDD = EVDD ≤ 5.5 V, VSS = EVSS = 0 V)

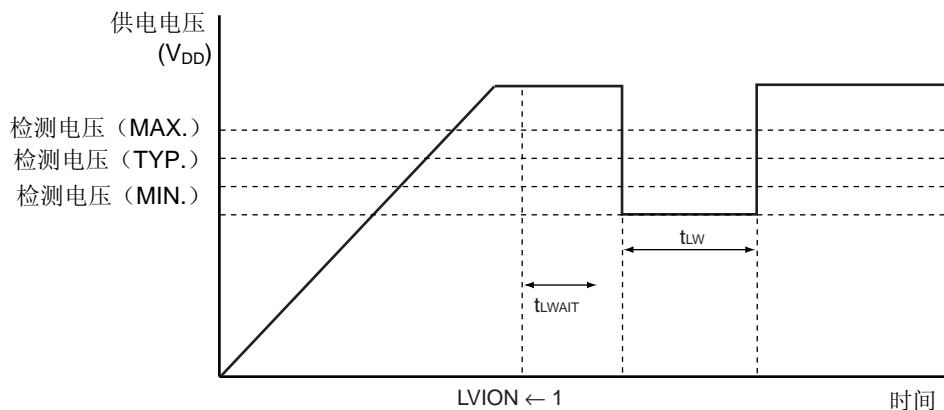
参数	符号	条件	MIN.	TYP.	MAX.	单位	
检测电压	供电电压电平	V _{LV10}		4.12	4.22	4.32	V
		V _{LV11}		3.97	4.07	4.17	V
		V _{LV12}		3.82	3.92	4.02	V
		V _{LV13}		3.66	3.76	3.86	V
		V _{LV14}		3.51	3.61	3.71	V
		V _{LV15}		3.35	3.45	3.55	V
		V _{LV16}		3.20	3.30	3.40	V
		V _{LV17}		3.05	3.15	3.25	V
		V _{LV18}		2.89	2.99	3.09	V
		V _{LV19}		2.74	2.84	2.94	V
		V _{LV110}		2.58	2.68	2.78	V
		V _{LV111}		2.43	2.53	2.63	V
		V _{LV112}		2.28	2.38	2.48	V
		V _{LV113}		2.12	2.22	2.32	V
		V _{LV114}		1.97	2.07	2.17	V
V _{LV115}		1.81	1.91	2.01	V		
外部输入引脚 ^{※1}	V _{EXLVI}	EXLVI < V _{DD} , 1.8 V ≤ V _{DD} ≤ 5.5 V	1.11	1.21	1.31	V	
电源开启时的供电电压	V _{PUPLVI}	当设置 LVI 默认启动功能允许时	1.87	2.07	2.27	V	
最小脉冲宽度	t _{LW}		2.00			μs	
检测延迟时间					200	μs	
操作稳定等待时间 ^{※2}	t _{LWAIT}				10	μs	

<R>

- 注 1. 使用 EXLVI/P120/INTP0 引脚。
 2. 从低电压检测寄存器 (LVIM) 的第 7 位 (LVION) 设置为 1, 到操作稳定所需要的时间。

备注 V_{LV1(n-1)} > V_{LV1n}; n = 1~15

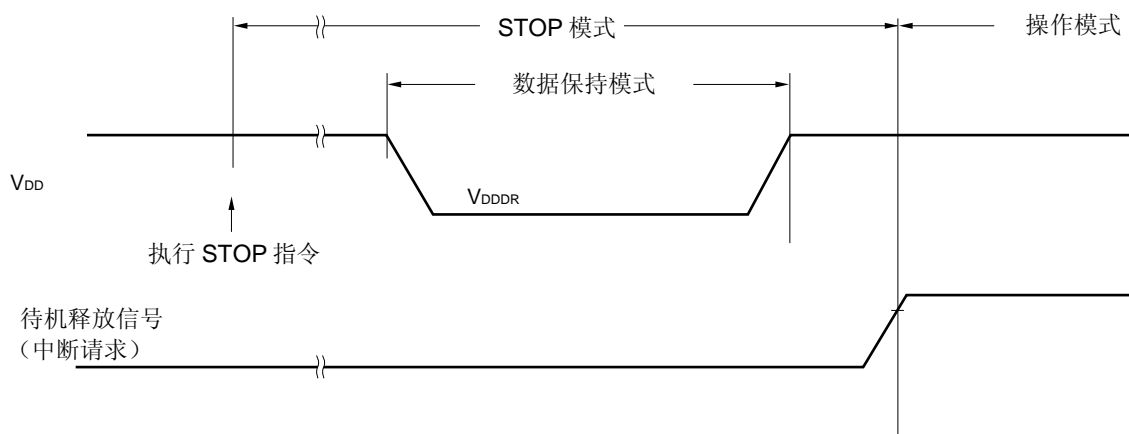
LVI 电路时序



数据存储器 STOP 模式低电压数据保持特性 (TA = -40~+85°C)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持供电电压	V _{DDDR}		1.5 [‡]		5.5	V

注 该值取决于 POC 检测电压。当电压降低时，一直保持该数据直至 POC 复位有效，但当 POC 复位有效时，数据不能保持。



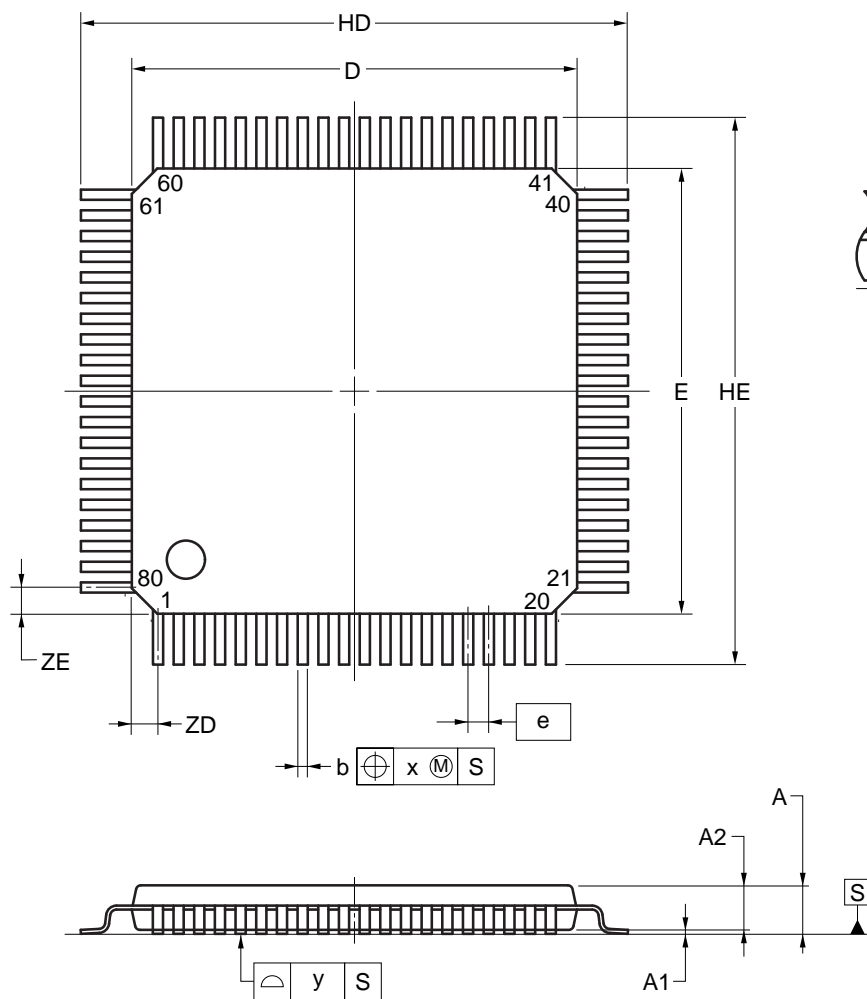
<R> Flash 存储器编程特性

(TA = -40~+85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
V _{DD} 供电电流	I _{DD}	Typ. = 10 MHz, Max. = 20 MHz		6	20	mA
CPU/外围硬件时钟频率	f _{CLK}		2		20	MHz
每片重写次数	C _{erwr}	保存: 15 年 1 次擦除 + 1 次擦除后写入 = 1 次重写 [‡]	100			次

注 在出厂后第一次对产品进行写操作时，“擦除 → 写入”和“只写”都作为一次重写。

80-脚塑封 LQFP (14x14)



引脚详图

(UNIT:mm)

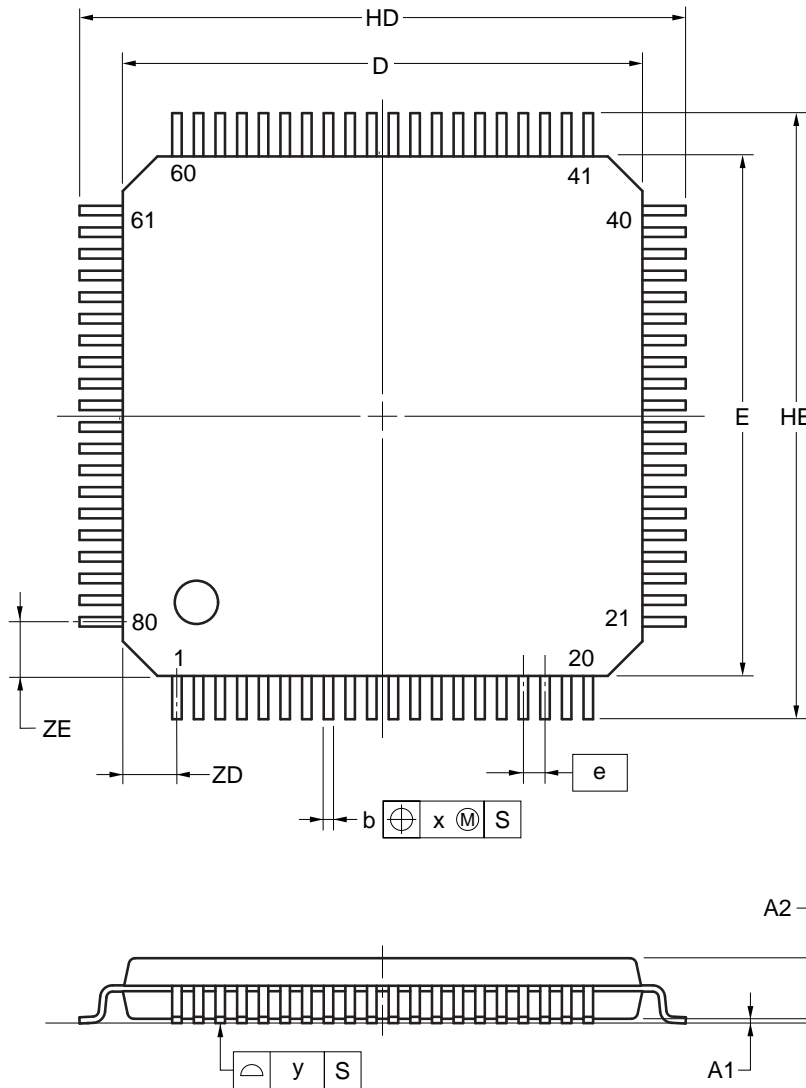
项目	尺寸
D	14.00±0.20
E	14.00±0.20
HD	17.20±0.20
HE	17.20±0.20
A	1.70 MAX.
A1	0.125±0.075
A2	1.40±0.05
A3	0.25
b	0.30 ^{+0.08} _{-0.04}
c	0.125 ^{+0.075} _{-0.025}
L	0.80
Lp	0.886±0.15
L1	1.60±0.20
θ	3° ^{+5°} _{-3°}
e	0.65
x	0.13
y	0.10
ZD	0.825
ZE	0.825

P80GC-65-GAD

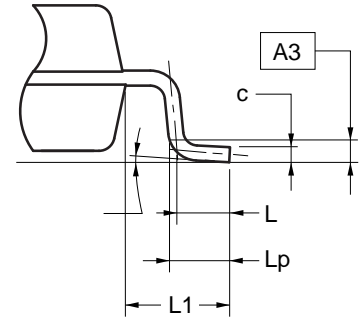
注

在最大使用材料情况下，每条引脚的中心线位于其真实位置的0.13mm内。

80-脚塑封 LQFP(密脚距)(12×12)



引脚详图



(UNIT:mm)

项目	尺寸
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.20 ^{+0.07} _{-0.03}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° ^{+5°} _{-3°}
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P80GK-50-GAK

注

在最大使用材料情况下，每条引脚的中心线位于其真实位置的 0.08mm 内。

© NEC Electronics Corporation 2005

附录A 开发工具

在涉及 78K0R/KF3 的系统开发中可以使用以下开发工具。

图 A-1 显示了开发工具的组成。

- **支持 PC98-NX 系列**

除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

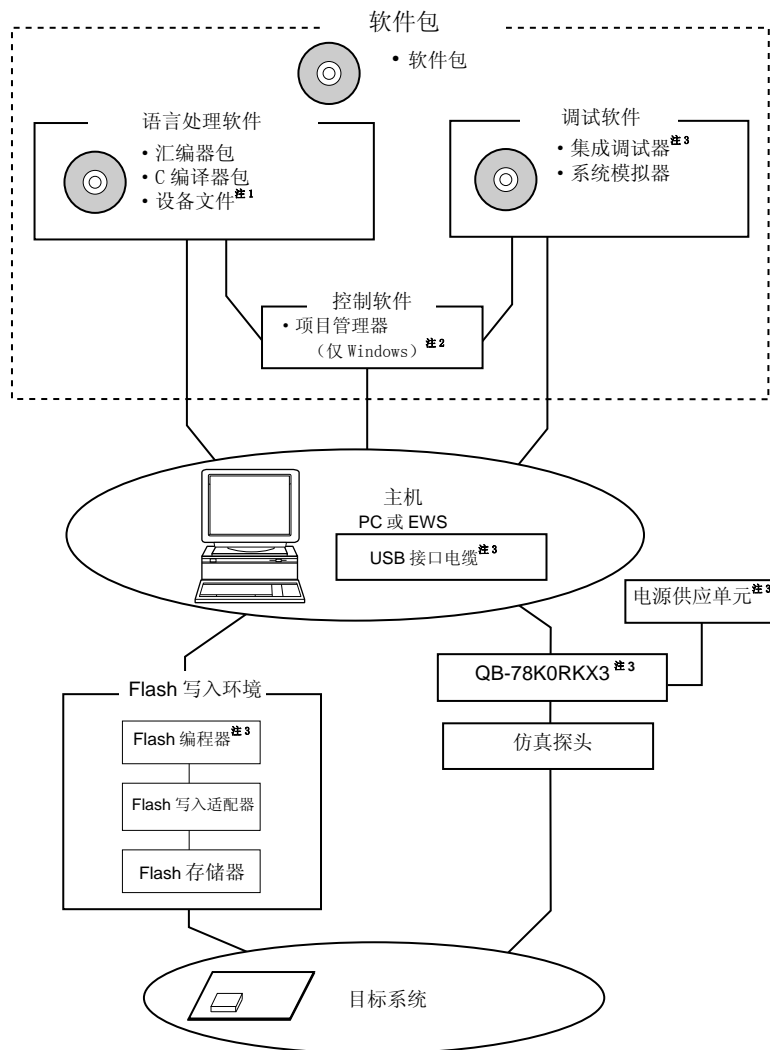
- **Windows™**

除非特别说明，“Windows”是指以下几种操作系统。

- Windows 98
- Windows NT™
- Windows 2000
- Windows XP

图 A-1 开发工具的组成 (1/2)

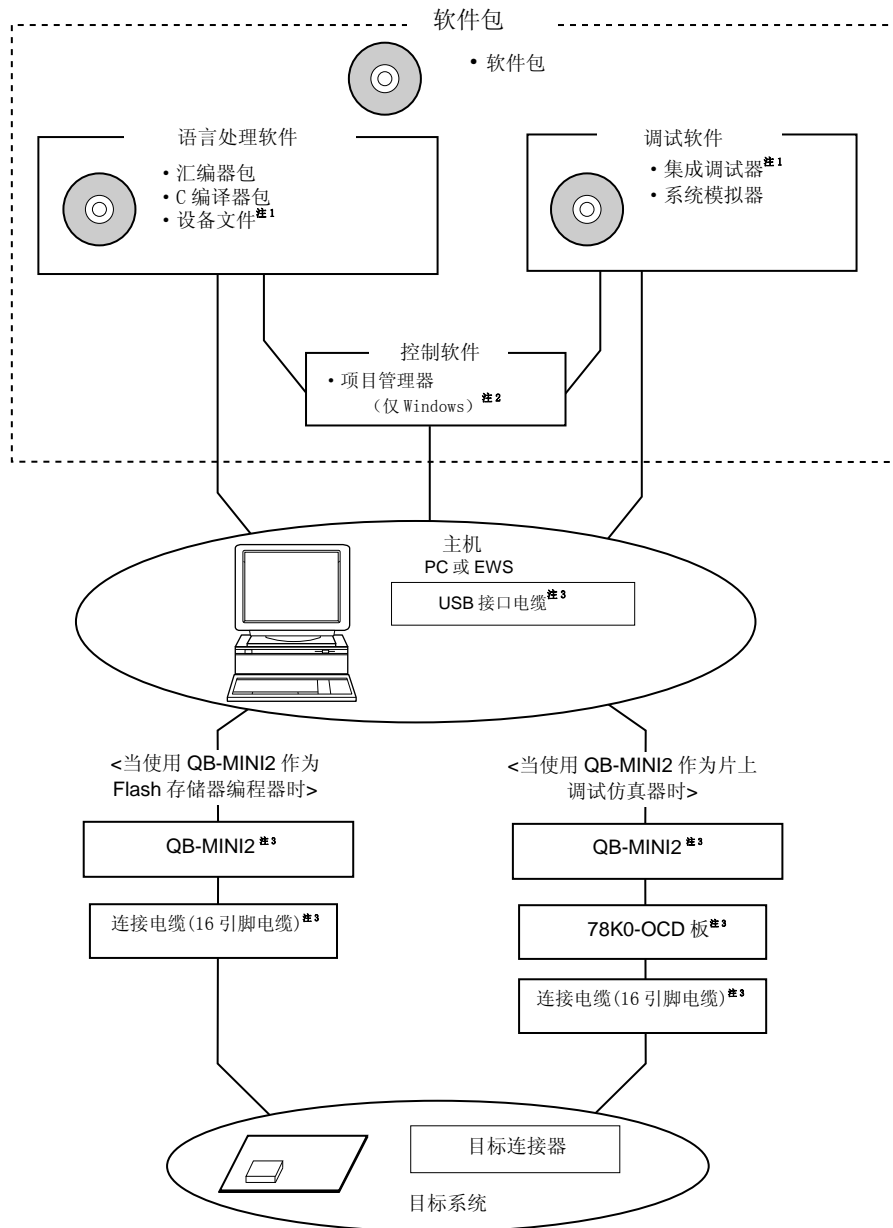
(1) 当使用在线仿真器 QB-78K0RKX3 时



- 注
1. 可从开发工具下载网站 (<http://www.necel.com/micro/ods/eng/index.html>) 下载获取 78K0R/KF3 (DF781188) 的设备文件。
 2. 项目管理器 PM+ 软件包含在汇编编译器中。
PM+ 只能在 Windows 下使用。
 3. 购买在线仿真器 QB-78K0RKX3 时还会附带提供集成调试器 ID78K0R-QB、具有编程功能的片上调试仿真器 QB-MINI2, 电源和 USB 接口电缆。除此之外的其他产品均独立销售。

图 A-1 开发工具的组成 (2/2)

(2) 当使用具有编程功能的片上调试仿真器 QB-MINI2 时



- 注
1. 可从开发工具下载网站 (<http://www.necel.com/micro/ods/eng/index.html>) 下载获取 78K0R/KF3 (DF781188) 的设备文件。
 2. 项目管理器 PM+ 软件包含在汇编编译器中。PM+ 只能在 Windows 下使用。
 3. 购买片上调试仿真器 QB-MINI2 时还附带提供 USB 接口电缆、连接电缆 (10 引脚电缆和 16 引脚电缆)、以及 78K0-OCD 板。其他产品均独立销售。另外, 可以从开发工具下载网站 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>) 下载 QB-MINI2 的操作软件。

A.1 软件包

SP78K0R 78K0R 系列软件包	开发工具（软件包）适用于 78K0R 微控制器。 产品型号： μ SxxxxSP78K0R
------------------------	--

备注 产品型号中的xxxx随主机和使用的操作系统而变化。

μ SxxxxSP78K0R

xxxx	主机	操作系统	存储介质
AB17	PC-9800 系列，	Windows（日文版本）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版本）	

A.2 语言处理软件

RA78K0R 汇编编译器	该汇编编译器将助记符编写的程序转换为微处理器可执行的目标代码。 该汇编编译器可以自动建立符号表，也可以对转移指令进行优化。 该汇编编译器应当和设备文件（DF781188）（独立销售）一起使用。 <在 PC 环境中使用 RA78K0R 的注意事项> 该汇编编译器是基于 D 操作系统的应用程序。它也可以在 Windows 环境下使用，但在 Windows 环境下时必须使用 project manager（包含在汇编编译器软件包中） 产品型号： μ SxxxxRA78K0R
CC78K0R C 编译器	该编译器将 C 程序转换成微控制器可执行的目标代码 该编译器应当和汇编程序、设备文件（二者均独立销售）结合使用。 <PC 环境中使用 CC78K0R 的注意事项> C 编译器是基于 D 操作系统的应用程序。它也可以在 Windows 环境下使用，但在 Windows 环境下时必须使用 project manager（包含在汇编编译器软件包中）。 产品型号： μ SxxxxCC78K0R
DF781188 ^注 设备文件	该文件包含设备特有的信息 该设备文件应当结合工具（RA78K0R， CC78K0R， SM+ for 78K0R， 和 ID78K0R-QB）（均为独立销售）一起使用 相应的操作系统和主机随使用工具而变化 产品型号： μ SxxxxDF781188

注 DF781188 可以与 RA78K0R， CC78K0R， SM+ for 78K0R， 和 ID78K0R-QB 一起使用。可从开发工具下载网站（<http://www.necel.com/micro/ods/eng/index.html>）下载 DF781188 文件。

备注 产品型号中的xxxx随主机和使用的操作系统而变化。

μ SxxxxRA78K0R

μ SxxxxCC78K0R

xxxx	主机	操作系统	存储介质
AB17	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版本)	CD-ROM
BB17		Windows (英文版本)	

μ SxxxxDF781188

xxxx	主机	操作系统	存储介质
AB13	PC-9800 系列, IBM PC/AT 兼容机	Windows (日文版本)	3.5 英寸 2HD FD
BB13		Windows (英文版本)	

A.3 控制软件

PM+ 项目管理器	<p>这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、建立程序和启动调试器，都可以由项目管理器执行</p> <p><注意事项></p> <p>项目管理器 PM+ 包含在汇编编译器软件包 (RA78K0R) 中。</p> <p>仅在 Windows 环境中使用</p>
--------------	--

A.4 Flash 存储器编程工具

<R> A.4.1 当使用Flash存储器编程器FG-FP5, FL-PR5, FG-FP4 和FL-PR4

FL-PR4, PG-FP4, FL-PR5, PG-FP5 Flash 存储器编程器	Flash 存储器编程器，专用于内置有 Flash 存储器的微控制器。
FA-78F1156GC-GAD-RX (支持 RoHS), FA-78F1156GK-GAK-RX (支持 RoHS) Flash 存储器编程适配器	Flash 存储器编程适配器，用于连接到 Flash 存储器编程器使用。 <ul style="list-style-type: none"> FA-78F1156GC-GAD-RX: 80-脚塑封 LQFP (GC-GAD 类型) FA-78F1156GK-GAK-RX: 80-脚塑封 LQFP (GK-GAK 类型)

备注 FL-PR4, FL-PR5, FA-78F1156GC-GAD-RX, 和 FA-78F1156GK-GAK-RX 都是 Naito Densei Machida Mfg. Co., Ltd.的产品。

A.4.2 当使用具有编程功能的片上调试仿真器QB-MINI2

QB-MINI2 具有编程功能的片上调试仿真器	这是具有片上 Flash 存储器的微控制器专用的 Flash 存储器编程器。也可以当作片上调试仿真器使用，用于在开发 78K0R 应用系统时调试硬件和软件。 片上调试仿真器 QB-MINI2 附带提供 USB 接口电缆、连接电缆（10 引脚电缆和 16 引脚电缆）、以及 78K0-OCD 板。要使用 78K0R/KF3，请使用 USB 接口线和 16 引脚连接电缆。
----------------------------	---

备注 可以从开发工具下载页面（<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>）下载QB-MINI2 的操作软件。

A.5 调试工具（硬件）

A.5.1 当使用在线仿真器QB-78K0RKX3

QB-78K0RKX3 [#] 在线仿真器	在使用 78K0R/Kx3 来开发应用系统时，片上调试仿真器用于调试硬件和软件。它支持集成调试器（ID78K0R-QB）。该仿真器与供电电压和仿真探头，以及用于将此仿真器连接到主机的 USB 一起使用。
QB-144-CA-01 引脚检测适配器 (Check pin adapter)	该引脚检测适配器用于示波器等监测波形。
QB-144-EP-02S 仿真探头 (Emulation probe)	这是一种可以灵活类型的仿真探头，一般用于连接在线仿真器和目标系统。
QB-80GC-EA-06T, QB-80GK-EA-06T 交换适配器 (Exchange adapter)	该适配器用于实现从在线仿真器到目标连接器的引脚转换。 • QB-80GC-EA-06T: 80-脚塑封 LQFP (GC-GAD 类型) • QB-80GK-EA-06T: 80-脚塑封 LQFP (GK-GAK 类型)
QB-80GC-YS-01T, QB-80GK-YS-01T 空间适配器 (Space adapter)	该适配器用于调整目标系统和在线仿真器之间的高度。 • QB-80GC-YS-01T: 80-脚塑封 LQFP (GC-GAD 类型) • QB-80GK-YS-01T: 80-脚塑封 LQFP (GK-GAK 类型)
QB-80GC-YQ-01T, QB-80GK-YQ-01T YQ 连接器 (YQ connector)	该连接器用于连接目标连接器和交换适配器。 • QB-80GC-YQ-01T: 80-脚塑封 LQFP (GC-GAD 类型) • QB-80GK-YQ-01T: 80-脚塑封 LQFP (GK-GAK 类型)
QB-80GC-HQ-01T, QB-80GK-HQ-01T 安装适配器 (Mount adapter)	该适配器用于通过插座安装目标设备。 • QB-80GC-HQ-01T: 80-脚塑封 LQFP (GC-GAD 类型) • QB-80GK-HQ-01T: 80-脚塑封 LQFP (GK-GAK 类型)
QB-80GC-NQ-01T, QB-80GK-NQ-01T 目标连接器 (Target connector)	该连接器一般被安装在目标系统上。 • QB-80GC-NQ-01T: 80-脚塑封 LQFP (GC-GAD 类型) • QB-80GK-NQ-01T: 80-脚塑封 LQFP (GK-GAK 类型)

- 备注**
1. 使用 QB-78K0RKX3 时应提供电源、USB 接口电缆。作为控制软件，还提供集成调试器 ID78K0R-QB 和具有编程功能的片上调试仿真器 QB-MINI2。
 2. 封装内容根据产品型号而不同，如下所示。

产品型号	封装内容	在线仿真器	仿真探头	交换适配器	YQ 连接器	目标连接器
QB-78K0RKX3-ZZZ	QB-78K0RKX3	QB-78K0RKX3	无			
QB-78K0RKX3-T80GC			QB-144-EP-02S	QB-80GC-EA-06T	QB-80GC-YQ-01T	QB-80GC-NQ-01T
QB-78K0RKX3-T80GK				QB-80GK-EA-06T	QB-80GK-YQ-01T	QB-80GK-NQ-01T

<R> A.5.2 当使用具有编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 具有编程功能的片上调试仿真器	这是具有片上 Flash 存储器的微控制器专用的 Flash 存储器编程器。也可以当作片上调试仿真器使用，用于在开发 78K0R 应用系统时调试硬件和软件。 片上调试仿真器 QB-MINI2 附带提供 USB 接口电缆、连接电缆（10 引脚电缆和 16 引脚电缆）、以及 78K0-OCD 板。要使用 78K0R/KF3，请使用 USB 接口线和 16 引脚连接电缆。
----------------------------	---

备注 可以从开发工具下载页面（<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>）下载QB-MINI2 的操作软件。

A.6 调试工具（软件）

SM+ for 78K0R 系统模拟器	SM+ for 78K0R 是基于 Windows 的软件。 在主机上模拟目标系统操作时，该仿真器用来执行 C 源程序或汇编程序级的调试操作。 使用 SM+ for 78K0R 可在不依赖硬件开发的基础上进行应用逻辑测试和性能测试，从而提供了较高的开发效率和软件质量。 SM+ for 78K0R 应当结合设备文件（DF781188）（独立销售）使用。 产品型号： μ SxxxxSM781000
ID78K0R-QB 集成调试器	该调试器支持 78K0R 系列微控制器的在线仿真器。ID78K0R-QB 是基于 Windows 的软件。 它改善了 C 兼容的调试功能，并使用集成窗口功能（结合源程序、反汇编显示和跟踪结果的存储区显示）来显示源程序的跟踪结果。它应当结合设备文件（独立销售）一起使用。 产品型号： μ SxxxxID78K0R-QB

备注 产品型号中的xxxx随主机和使用的操作系统而变化。

μ SxxxxSM781000

μ SxxxxID78K0R-QB

xxxx	主机	操作系统	存储介质
AB17	PC-9800 系列，	Windows（日文版本）	CD-ROM
BB17	IBM PC/AT 兼容机	Windows（英文版本）	

附录B 修订历史

B.1 本版中的主要修订之处

(1/4)

页码	描述	分类
通篇	修改 UPD78F1152, 78F1153, 78F1154, 78F1155, 78F1156 的状态, 从开发中改为量产中。	(d)
第二章 引脚功能		
p.25	修改表 2-1. 引脚 I/O 缓冲式供电电源中的 EVDD 和 VDD 对应引脚	(c)
p.40	修改 2.2.21 FLMD0 的描述	(c)
pp.42, 43	修改在表 2-2 未使用引脚的连接中的 37-A 到 37-B, 和 39 到 2-W	(c)
pp.44, 45	修改图 2-1. 引脚 I/O 电路列表中的 37-A 到 37-B, 和 39 到 2-W	(c)
第三章 CPU 结构		
p.66	修改图 3-16. 通用寄存器的配置中的地址	(a)
pp.71, 72	增加表 3-5 特殊功能寄存器列表中的注释和寄存器	(c)
第四章 端口功能		
通篇	在框图中增加 PIM 寄存器和 POM 寄存器	(c)
p.93	修改表 4-1. 引脚 I/O 缓冲器电源中 EVDD 和 VDD 的对应引脚	(c)
p.97	修改 4.2.1 端口 0 中的注意事项 1 和注意事项 2	(c)
p.103	修改 4.2.2 端口 1 中的注意事项 1, 注意事项 2 和注意事项 3	(c)
p.110	修改 4.2.4 端口 3 中的注意事项 1, 并增加注意事项 2	(c)
p.112	修改 4.2.5 端口 4 中的注意事项 2 和注意事项 3	(c)
p.122	在 4.2.7 端口 6 中增加注意事项	(c)
p.132	修改 4.2.13 端口 14 中的注意事项 1 和注意事项 2, 并增加注意事项 3	(c)
p.141	增加 4.3 中 (4) 端口输入模式寄存器 (PIM0, PIM4, PIM14) 和 (5) 端口输出模式寄存器 (POM0, POM4, POM14) 的描述	(c)
第五章 时钟发生器		
p.160	增加图 5-6 系统时钟控制寄存器 (CKC) 的格式中注释 3	(c)
p.164	在图 5-8. 操作速度模式控制寄存器 (OSMC) 的格式中增加注意事项 5	(b)
第六章 定时器阵列单元		
p.194	修改表 6-1. 定时器阵列单元的配置	(a)
p.201	修改图 7-7. 定时器模式寄存器 0n (TMR0n) 的格式 (1/3) 中 MASTER0n 位的描述	(c)
p.211	在图 6-16. 定时器输入选择寄存器 0 (TIS0) 的格式中增加注意事项	(c)
p.213	增加 6.3 (10) 定时器输出寄存器 0 (TO0) 中的描述	(c)
p.215	增加 6.3 (12) 定时器输出模式寄存器 0 (TOM0) 中的描述	(c)
p.216	修改图 6-21. 输入切换控制寄存器 (ISC) 的格式中的备注	(c)
第七章 实时计数器		
p.271	修改图 7-2. 外围设备允许寄存器 0 (PER0) 的格式中的注意事项	(c)
p.281	增加 7.3 (15) 小时报警寄存器 (ALARMWH) 的描述	(c)
p.283	增加图 7-18. 启动实时计数器的操作流程图中的注释	(c)

备注 上表中的修改“分类”如下

- (a) : 错误纠正, (b) : 添加/修改规范, (c) : 添加/修改描述或者注释,
 (d) : 添加/修改封装, 器件编号, 或管理划分, (e) : 添加/修改相关文档

页码	描述	分类
第八章 看门狗定时器		
p.289	修改 8.3 (1) 看门狗定时器允许寄存器 (WDTE) 中的注意事项 1 和注意事项 2	(a)
第十二章 串行阵列单元		
p.328	修改 图 12-1 串行阵列单元 0 的框图	(a)
p.329	修改 图 12-2 串行阵列单元 1 的框图	(a)
p.338	增加图 12-7. 串行通信操作设定寄存器 mn (SCRmn) 的格式 (2/3) 中的注释	(c)
p.348	增加 12.3 (12) 串行输出寄存器 m (SOM) 的描述	(c)
pp.354 至 356	增加 12.4 stop 模式的操作	(c)
pp.359, 368, 374	修改 (a) 串行输出寄存器 m (SOM)	(a)
p.362	修改图 12-27. 恢复主机传输的过程	(c)
p.371	修改图 12-36. 主机接收的时序图 (在单个接收模式下)	(a)
p.376	修改图 12-41. 恢复主机发送/接收的过程	(a)
p.377	修改图 12-42. 主机发送/接收的时序图 (在单个发送/接收模式下)	(a)
p.379	修改图 12-44. 主机发送/接收的时序图 (在连续发送/接收模式下)	(a)
p.380	修改图 12-45. 主机发送/接收的流程图 (在连续发送/接收模式下)	(a)
p.385	修改图 12-49. 恢复从机发送的过程	(a)
p.386	修改图 12-50. 从机发送的时序图 (在单个发送模式下)	(c)
p.393	修改图 12-57. 恢复从机接收的过程	(a)
p.394	修改图 12-58. 从机接收的时序图 (在单个接收模式下)	(a)
p.400	修改图 12-63. 恢复从机发送/接收的过程	(a)
p.401	修改图 12-64. 从机发送/接收的时序图 (在单个发送/接收模式下)	(a)
p.403	修改图 12-66. 从机发送/接收的时序图 (在连续发送/接收模式下)	(a)
p.404	修改图 12-67. 从机发送/接收的流程图 (在连续发送/接收模式下)	(a)
p.418	修改 12.6.2 UART 接收	(a)
p.423	修改图 12-80. UART 接收的时序图	(a)
p.425	修改 12.6.3 LIN 发送	(a)
p.428	修改 12.6.4 LIN 接收	(a)
p.440	修改图 12-89. 地址域发送的初始化设置过程	(a)
p.441	修改图 12-90. 地址域发送的时序图	(a)
p.442	修改图 12-91. 地址域发送的流程图	(a)
p.444	修改图 12-92. 简易 I ² C (IIC10, IIC20) 数据发送寄存器内容的举例, 并增加注释	(a)
p.445	修改图 12-94. 数据发送的流程图	(a)

备注 上表中的修改“分类”如下

- (a) : 错误纠正, (b) : 添加/修改规范, (c) : 添加/修改描述或者注释,
(d) : 添加/修改封装, 器件编号, 或管理划分, (e) : 添加/修改相关文档

页码	描述	分类
第十二章 串行阵列单元 (续)		
p.447	修改图 12-95. 简易 I ² C (IIC10, IIC20) 的数据接收的寄存器内容举例, 并增加注释	(a)
p.448	修改图 12-96. 数据接收的时序图	(c)
p.448	修改图 12-97. 数据接收的流程图, 并增加注意事项	(c)
p.449	修改图 12-99. 停止条件产生的流程图	(c)
pp.455 至 461	增加 12.9 寄存器设置和引脚之间的关系	(c)
第十六章 中断功能		
pp.558, 559	修改表 16-1. 中断源列表	(a)
第十八章 待机功能		
p.586	增加图 18-3. 通过产生中断请求释放 HALT 模式中的注释	(b)
p.590	增加图 18-5. 释放 STOP 模式时 (当产生未屏蔽中断请求时) 的操作时序中的注释	(b)
pp.591, 592	增加图 18-6. 通过产生中断请求释放 STOP 模式中的注释	(b)
第十九章 复位功能		
p.594	修改 (4) 中的描述	(c)
p.596	修改图 19-2 由 /RESET 输入引起的复位时序	(b)
p.597	修改图 19-4 在 STOP 模式下由 /RESET 输入引起的复位时序	(b)
第二十四章 FLASH 存储器		
p.641	修改 24.4.1 FLMD0 引脚	(c)
p.649	修改 24.8 通过自编程进行 Flash 存储器编程中的备注	(e)
p.650	修改图 24-10. 自编程的流程 (重写 Flash 存储器), 并添加备注	(b, e)
第二十六章 BCD 修正电路		
pp.658, 659	增加 26.3 用户资源的保留	(c)
第二十七章 指令集		
p.661	在表 27-2. “操作” 栏中的符号中增加 addr5	(c)
p.677	修改表 27-5. 操作列表 (15/17) 中的 CALLT 操作	(b)
第二十八章 电气特性		
通篇	修改 UPD78F1152, 78F1153, 78F1154, 78F1155, 78F1156 的规范, 从目标规范改为正式规范。	(b)
p.680	最大额定值 <ul style="list-style-type: none"> • 输入电压的修改 • 输出电压的修改 • 修改模拟输出电压 (VAO) 的额定值 	(b)
p.683	修改内部振荡器特性中的注释 1	(c)

备注 上表中的修改“分类”如下

- (a) : 错误纠正, (b) : 添加/修改规范, (c) : 添加/修改描述或者注释,
(d) : 添加/修改封装, 器件编号, 或管理划分, (e) : 添加/修改相关文档

页码	描述	分类
第二十八章 电气特性 (续)		
pp.685 至 690, 692, 694	直流特性 <ul style="list-style-type: none"> • 修改输出电流, 高 (IOH2) 的条件 • 修改输出电流, 低 (IOL2) 的条件 • 修改输入电压, 高 (VIH4) 的条件 • 修改输入电压, 高 (VIH5) 的条件 • 修改输入电压, 高 (VIL4) 的条件 • 修改输入电压, 高 (VIL5) 的条件 • 修改注意事项 2 • 修改输出电压, 高 (VOH2) 的条件 • 修改输出电压, 低 (VOL2) 的条件 • 修改输入漏电流, 高 (LIH2) 的条件 • 修改输入漏电流, 高 (LIH3) 的条件 • 修改输入漏电流, 低 (LIL2) 的条件 • 修改输入漏电流, 低 (LIL3) 的条件 • 修改供应电流 (IDD1), 并增加低功耗电流模式, 注释 4, 和备注 3 • 修改供应电流 (IDD2), 并增加低功耗电流模式, 注释 4, 和备注 3 	(b)
pp.698 至 702	交流特性 <p>(1) 基本操作</p> <ul style="list-style-type: none"> • 在(1) 基本操作中增加主系统时钟操作下的最短指令执行时间和自编程模式下的最短指令执行时间的图 • 修改 AC 时序测试点的标题 	(b)
p.722	修改表中上部分的条件	(a)
p.723	修改供电电压上升时间时序图和图的标题	(c)
附录 A 开发工具		
p.732	修改 A.4.1 当使用 flash 存储器编程器 FG-FP5, FL-PR5, FG-FP4 和 FL-PR4	(b)
p.733	修改 A.4.2 当使用具有编程功能的片上调试仿真器 QB-MINI2	(c)
p.735	修改 A.5.2 当使用具有编程功能的片上调试仿真器 QB-MINI2	(c)

备注 上表中的修改“分类”如下

- (a) : 错误纠正, (b) : 添加/修改规范, (c) : 添加/修改描述或者注释,
(d) : 添加/修改封装, 器件编号, 或管理划分, (e) : 添加/修改相关文档

B.2 以往版本的修订历史

以下是前面版本的修订历史。章节编号表示每个版本的对应章节。

(1/11)

版本	描述	章节
第 2 版	1.1 特性	第一章 概述
	<ul style="list-style-type: none"> • 改变μPD78F1157 和μPD78F1158 的状态为“开发中” • 增加片上 BCD 调整 • 增加 8 位分辨率的 D/A 转换器 	
	增加 1.4 引脚配置（俯视图）中的注意事项 2	
	增加 1.5 78K0R 系列单片机列表	
	在 1.6 框图中增加 BCD 修正电路，并修改外部总线接口内部 I/O 引脚的箭头方向	
	在 1.7 功能概述中改变 μ PD78F1157 和 μ PD78F1158 的状态为“开发中”	
	在 2.1 (2) 非端口功能中修改 EX25, EX26, SO00, SO01, TxD0, 和 TxD3 功能的复用功能	第二章 引脚功能
	在 2.2.5 P40 ~ P47 (port 4) 中增加复用功能描述，并修改注意事项	
	在表 2-2 未使用引脚的连接中增加 I/O 电路类型	
	增加表 2-1 引脚 I/O 电路列表	
	删除第 3 章中的 CALLF 指令描述	第三章 CPU 结构
	修改 3.1 存储器空间中的描述	
	增加图 3-5. 存储器映射图 (μ PD78F1166) 和图 3-13. 数据存储器与寻址方式的对应关系 (μ PD78F1166) 中的注释	
	增加图 3-7. 存储器映射图 (μ PD78F1168) 和图 3-15. 数据存储器与寻址方式的对应关系 (μ PD78F1168) 中的注释	
	修改 3.1.2 镜像区域中的描述，并增加框图示例和 PMC 寄存器的解释	
	修改表 3-5. SFR 列表中的小时计数寄存器和小时报警寄存器的复位值	
	修改表 3-5. SFR 列表中的日计数寄存器和月计数寄存器的复位值	
	修改表 3-5. SFR 列表中的背景事件控制寄存器的复位值	
	在表 3-5. SFR 列表中增加 BCD 调整结果寄存器和注释	
	修改表 3-5. SFR 列表中的高位乘法结果存储寄存器和低位乘法结果存储寄存器的符号	
	在表 3-6. 扩展 SFR (第二 SFR) 列表中增加稳压器模式控制寄存器和 BCD 调整结果寄存器	
	在表 3-6. 扩展 SFR (第二 SFR) 列表中增加如下寄存器的低 8 位的 SFR 名称，并修改它们的 R/W 属性和位操作方式，包括有 SSRmn, SIRmn, SEm, SSm, STm, SPSm, SOEm, SOLm, TCR0n, TSR0n, TE0, TS0, TT0, TPS0, TO0, TOE0, TOL0, 和 TOM0	
	修改表 3-6. 扩展 SFR (第二 SFR) 列表中的串行输出寄存器 0 和串行输出寄存器 1 的复位值	
	修改表 3-6. 扩展 SFR (第二 SFR) 列表中的串行输出允许寄存器 0 和串行输出允许寄存器 1 的复位值	
	修改表 3-6. 扩展 SFR (第二 SFR) 列表中的定时器通道计数寄存器 0n 的 R/W 属性	
	增加 3.3 指令地址寻址	
	增加 3.4 处理数据地址寻址	

版本	描述	章节
第 2 版	增加 4.2.1 端口 0 中的注意事项 1 和注意事项 2	第四章 端口功能
	增加 4.2.2 端口 1 中的注意事项 1 和注意事项 2	
	增加 4.2.4 端口 3 中的注意事项	
	增加 4.2.5 端口 4 中的注意事项 2 和注意事项 3	
	修改图 4-28. P80~ P87 的框图和 图 4-29. P110 和 P111 的框图	
	增加 4.2.12 端口 13 中的注意事项	
	增加 4.2.14 端口 13 中的注意事项 1 和注意事项 2	
	增加图 4-39. 端口输入模式寄存器的格式中的注意事项	
	修改 4.3 (2) 端口寄存器 (P0 ~ P8, P11~ P15) 中的注释	
	增加 4.4.4 连接到不同的供电电压 (3V) 的外部设备	
	增加图 5-3. 存储器扩展模式控制寄存器 (MEM) 的格式中的注释	第五章 外部总线接口
	修改 5.6 (5) ASTB 引脚 和 (6) EX0~EX7, EX8~EX15, EX16~EX23, 和 EX24~EX31 引脚中的描述	
	修改图 5-9. 同步存储器连接的示例和图 5-10. 异步存储器连接的示例	
	增加图 6-3. 时钟操作状态控制寄存器 (CSC) 的格式中的注意事项 3	第六章 时钟发生器
	修改 6.3 (3) 振荡稳定时间计数器状态寄存器 (OSTC) 中的描述	
	修改图 6-4. 振荡稳定时间计数器状态寄存器 (OSTC) 的格式的注意事项 2	
	修改图 6-5. 振荡稳定时间选择寄存器 (OSTS) 的格式的注意事项 5	
	修改图 6-6. 系统时钟控制寄存器 (CKC) 的格式的注意事项 3	
	修改图 6-8. 操作速度模式控制寄存器 (OSMC) 的格式中的注意事项 1 和注意事项 3	
增加图 6-14. 供电电源打开时时钟发生器的操作 (当设置 LVI 默认启动功能允许 (选项字节: LVIOFF = 1) 时) 及其描述		
增加图 6-15. 供电电源打开时时钟发生器的操作 (当设置 LVI 默认启动功能允许 (选项字节: LVIOFF = 0) 时) 及其描述		
修改 6.6.1 (1) X1 时钟振荡时设置过程示例中的注意事项 1		
修改 6.6.1 (2) <2>标题中的寄存器名称		
增加 6.6.1 (4) (b) 中的<2>		
增加 6.6.2 (2) (b) 中的注释		
修改 6.6.3 控制副系统时钟的示例中的注意事项		
修改 6.6.3 (1) 副系统时钟振荡的设置示例中的注意事项		
修改 6.6.3 (2) <2>设置副系统时钟作为 CPU 时钟 (CKC 寄存器) 的源时钟中的位名称		
修改 6.6.3 (2) 使用副系统时钟作为 CPU 时钟时设置过程示例中的注意事项		
修改 6.6.3 (3) <2>标题中的寄存器名称		

版本	描述	章节
第 2 版	增加图 6-16. CPU 时钟状态迁移图中从 (C) 到 (B) 的箭头	第六章 时钟发生器
	修改表 6-4. CPU 时钟切换与 SFR 寄存器设置示例	
	增加表 6-5. 切换 CPU 时钟描述	
	修改 6.6.7 CPU 时钟和主系统时钟切换所需的时间的描述	
	删除表 6-8. 类型 2 中需要的最大时钟数量中的注意事项	
	修改第 7 章中 TIS0n0 和 TIS0n1 位 ~ CIS0n0 和 CIS0n1 位的位名称	第七章 定时器阵列单元
	增加 7.1.1 独立操作时每个通道的功能中的描述	
	增加 7.1.2 和其他通道组合操作时每个通道的功能中的描述	
	增加 7.2 (1) 定时器/计数器 寄存器 0n (TCR0n) 中的描述和表格	
	删除 7.2 (2) 定时器数据寄存器 0n (TDR0n) 中的注意事项	
	在 7.3 控制定时器阵列单元的寄存器中增加 TSR0n, TE0, TS0, TT0, TPS0, TO0, TOE0, TOL0, and TOM0 寄存器低 8 位的 SFR 名称	
	增加 7.3 (2) 定时器时钟选择寄存器 0 (TPS0) 中的描述	
	修改图 7-6. 定时器模式寄存器 m (TMP0n) 的格式中的描述, 并改变设置	
	修改图 7-9. 定时器通道启动寄存器 0 (TS0) 的格式中的 R/W 属性	
	修改图 7-10 定时器通道停止寄存器 0 (TT0) 的格式中的 R/W 属性	
	修改 7.3 (9) 定时器输出 允许寄存器 0 (TOE0) 中的描述	
	修改 7.3 (10) 定时器输出 寄存器 0 (TO0) 中的描述	
	修改 7.3 (11) 定时器输出电平寄存器 0 (TOL0) 中的描述	
	修改图 7-16 输入切换控制寄存器 (ISC) 的格式	
	修改图 7-20 作为间隔定时器/方波输出操作的基本时序示例	
	增加 7.5.4 作为输入脉冲间隔测量操作中的注意事项	
	修改图 7-31 作为输入脉冲间隔测量操作的框图	
	修改第 7 章中 TIS0n0 和 TIS0n1 位 ~ CIS0n0 和 CIS0n1 位的位名称	
增加 7.5.5 用作输入信号高/低电平宽度测量操作中的注意事项		
修改 7.6.1 作为 PWM 功能操作中的描述		
修改 7.6 定时器阵列单元多通道操作中的备注		
修改图 7-43 作为 PWM 功能操作的框图中的描述		
修改图 7-44 作为单脉冲输出功能操作的框图		
修改 7.6.3 作为多路 PWM 输出功能操作中的描述		

版本	描述	章节
第 2 版	修改图 8-2. 外围设备允许寄存器 0 (PER0) 的格式中的注意事项, 并增加备注	第八章 实时计数器
	修改 8.3 (2) 实时计数器控制寄存器 0 (RTCC0) 中的注意事项	
	修改 8.3 (3) 实时计数器控制寄存器 1 (RTCC1) 中的注意事项	
	增加图 8-4 实时计数器控制寄存器 1 (RTCC1) 的格式和图 8-21 报警设置流程图中的备注	
	修改复位值, 并增加 8.3 (8) 小时计数寄存器 (HOUR) 中的描述	
	修改 8.3 (9) 日期计数寄存器 (DAY) 中的复位值	
	修改 8.3 (11) 星期计数寄存器 (WEEK) 中的复位值	
	修改 8.3 (15) 小时报警寄存器 (ALARMWH) 中的复位值	
	增加 11.3 (7) 端口模式寄存器 2 和 15 (PM2, PM15) 中的注意事项	第十一章 A/D 转换器
	在 13.4 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的通信操作至 13.6 简易 I2C (IIC10, IIC20) 的通信操作中增加关于寄存器 PER0 和 SPSm 的注意事项	第十三章 串行阵列单元
	增加 13.3 控制串行阵列单元的寄存器中的 SSRmn, SIRmn, Semn, SSm, STm, SPSm 和 SOLm 寄存器的低 8 位 SFR 名称	
	修改 13.3 控制串行阵列单元的寄存器中的 SIRmn, SSm, 和 STm 寄存器的 R/W 属性	
	修改 13.3 (12) 串行输出寄存器 m (S0m) 中的复位值	
	修改图 13-36 (d) 串行模式寄存器 mn (SMRmn) 中第 0 位的设置	
	删除 13.6 简易 I2C (IIC10, IIC20) 的通信操作中关于溢出错误的描述	
	删除 13.6.1 地址域传输中关于溢出错误的描述	
	删除 13.6.2 数据传输中关于溢出错误的描述	
	删除 13.6.3 数据接收中关于溢出错误的描述	
	增加 14.5.18 发生 I2C 中断请求 (INTIIC0) 的时序	第十四章 串行接口 IIC0
	增加 14.6 时序图	
	修改第 15 章的高位乘法结果存储寄存器和低位乘法结果存储寄存器	第十五章 乘法器
	增加图 15-2 16 位乘法结果高位存储寄存器和 16 位乘法结果低位存储寄存器 (MULOH, MULOL) 的格式	
	增加图 15-3 乘法输入数据寄存器 A, B (MULA, MULB) 的格式	
	增加 16.2 (1) DMA SFR 地址寄存器 n (DSAn) 中的注释	第十六章 DMA 控制器
	增加表 17-2 与中断请求源相对应的标志中的注释	第十七章 中断功能
	修改图 17-2 按键返回模式寄存器 (KRM) 的格式中的 IF2L 寄存器第 0 位至第 2 位的位名称	
	修改图 17-3 按键中断检测引脚的分配中的 MKOL 寄存器第 0 位的位名称	
	修改 17.4.4 保持中断请求	

版本	描述	章节
第 2 版	修改 19.1.2 (1) 振荡稳定时间计数器的状态寄存器 (OSTC) 中的描述	第十九章 待机功能
	修改 19.1.2 (2) 振荡稳定时间选择寄存器 (OSTS) 中的复位值	
	修改图 19-2. 振荡稳定时间选择寄存器 (OSTS) 的格式中的设置	
	修改表 19-1 HALT 模式中的操作状态中关于系统时钟的 f_{IL} 和主系统时钟的 f_x , f_{EX} 描述	
	修改表 19-2 STOP 模式中的操作状态中关于系统时钟的 f_{IL} , RAM 和实时时钟 (RTC) 描述	
	修改图 19-5 释放 STOP 模式时的操作时序	
	修改图 19-6 通过产生中断请求释放 STOP 模式, 并增加 (2) 当 CPU 使用高速系统时钟 (外部时钟输入)	
	在图 20-1 复位功能的框图中增加 RESF 寄存器读取信号	第二十章 复位功能
	在表 20-1 复位期间的操作状态中增加外部总线接口	
	修改表 20-2 复位响应后的硬件状态中实时计数器的小时计数寄存器 (HOUR), 日计数寄存器 (DAY), 月计数寄存器 (MONTH) 和分钟报警寄存器 (ALARMWM) 复位后的状态	
	修改图 21-2 由上电清零电路和低电压检测电路产生内部复位信号的时序, 并增加注释 4	第二十一章 上电清零电路
	增加 22.4.1 用作复位时	第二十二章 低电压检测电路
	增加 22.4.2 用作中断时	
	增加该章内容	第二十三章 稳压器
修改图 24-2 选项字节 (000C1H/010C1H) 的格式中的注意事项	第二十四章 选项字节	
增加 25.5 控制 Flash 存储器的寄存器	第二十五章 FLASH 存储器	
增加该章内容	第二十六章 BCD 修正电路	
增加该章内容	第二十七章 指令集	
DC 特性 <ul style="list-style-type: none"> • 修改输入电压, 高 (V_{IH7}) 的 MIN. 值, 并增加注释 1 • 修改输入电压, 低 (V_{IL5}) 的 MAX. 值 • 修改输入电压, 低 (V_{IL7}) 的 MAX. 值, 并增加注释 2 • 修改输出电压, 高 (V_{OH1}) 的条件 • 修改输出电压, 低 (V_{OL1}, V_{OL3}) 的条件 • 修改输入漏电流, 高 (I_{LH4}) 的条件 • 修改输入漏电流, 低 (I_{LIL4}) 的条件 	第二十八章 电气特性 (目标)	
修改 AC 特性 (1) 基本操作中 AC 时序测量位置的图		

版本	描述	章节
第 2 版	A/D 转换器特性 <ul style="list-style-type: none"> 修改表中上部分的条件 修改差分线形误差 (DLE) 的条件和 MAX. 值 	第二十八章 电气特性 (目标)
	D/A 转换器特性 <ul style="list-style-type: none"> 修改表中上部分的条件 增加 D/A 转换器操作电流 (I_{DAC}) 修改设置时间 (t_{SET}) 的条件 	
	增加该章内容	附录 A 修订历史
第 3 版	删除内部高速振荡时钟的温度修正功能, 并从下列各章中删除温度修正表 H、L 的有关描述 <ul style="list-style-type: none"> 第三章 CPU 结构 第五章 时钟发生器 第十一章 A/D 转换器 第十九章 复位功能 第二十八章 电气特性 (目标) 	通篇
	改变 μ PD78F1152 和 μ PD78F1153 的状态为“开发中”	第一章 概述
	1.1 Feature <ul style="list-style-type: none"> 增加单电源 flash 存储器安全功能 在自编程功能中增加 flash 防护窗口功能 	
	修改图 3-1 存储器映射图 (μ PD78F1152) 至图 3-5 存储器映射图 (μ PD78F1156)	第三章 CPU 结构
	增加 3.1.1 (4) 片上调试安全 ID 设置区域	
	增加 3.1.3 内部数据存储器空间中的注意事项	
	增加 3.2.4 特殊功能寄存器 (SFRs) 中的注意事项	
	修改表 3-5 SFR 列表中的 BCD 调整结果寄存器	
	增加 3.2.5 扩展特殊功能寄存器 (第二 SFR: 第二特殊功能寄存器) 中的注意事项	
	修改图 5-1 时钟发生器的框图	第五章 时钟发生器
	增加图 5-7 外围设备允许寄存器的格式中的注意事项	
	增加 5.3 (7) 操作速度模式控制寄存器 (OSMC) 中的注释 4	
	修改 5.3 (8) 内部高速振荡器微调寄存器 (HIOTRM) 中的描述	
	在图 5-13 供电电源打开时 时钟发生器的操作 (当设置 LVI 默认开始功能停止 (选项字节: LVIOFF = 1) 时) 中增加时钟单元 CPU 操作启动	
	修改图 5-14 供电电源打开时 时钟发生器的操作 (当设置 LVI 默认开始功能允许 (选项字节: LVIOFF = 0) 时)	
	增加 5.6.1 (3) <3> 中的注意事项	
增加 6.3 (1) 外围设备允许寄存器 0 (PER0) 中的注意事项 2	第六章 定时器阵列单元	
修改图 6-6 定时器模式寄存器 0n (TMR0n) 的格式		
增加 6.3 (4) 定时器状态寄存器 0n (TSR0n) 中的描述		
增加表 6-3 在每种操作模式下 OVF 位操作和设置/清零条件		

版本	描述	章节
第3版	增加表 6-4 从计数操作允许状态到 TCR0n 计数开始的操作中的 (a) 至 (e)	第六章 定时器阵列单元
	增加 6.3 (11) 定时器输出电平寄存器 0 (TOL0) 中的描述	
	增加 6.3 (12) 定时器输出模式寄存器 0 (TOM0) 中的描述	
	修改图 6-20 定时器输出模式寄存器 0 (TOM0) 的格式以及其中的备注	
	修改图 6-22 噪声滤波器允许寄存器 1 (NFEN1) 的格式中第 7 位的描述, 并增加注释	
	增加 6.4 通道输出 (TO0n 引脚) 控制	
	增加 6.5 通道输入 (TI0n 引脚) 控制	
	在下列图的标题中增加 MD0n0 位条件	
	<ul style="list-style-type: none"> 图 6-37 作为间隔定时器/方波输出 (MD0n0 = 1) 操作的基本时序示例 图 6-45 作为分频器 (MD0n0 = 1) 操作的基本时序示例 图 6-49 作为输入脉冲间隔测量 (MD0n0 = 0) 操作的基本时序示例 	
	修改 6.7.3 用作分频操作中的描述	
	修改 6.8.3 作为多路 PWM 输出功能操作中的描述	
	修改实时寄存器的清除条件	第七章 实时计数器
	修改图 7-2 外围设备允许寄存器 0 (PER0) 的格式中的描述, 并增加注意事项 1	
	增加图 7-2 外围设备允许寄存器 0 (PER0) 的格式中的注意事项 2	
	增加图 7-4 实时计数器控制寄存器 1 (RTCC1) 的格式中的注意事项	
	增加图 7-5 实时计数器控制寄存器 2 (RTCC2) 的格式中的注意事项 2	
	修改 7.3 (5) 副计数寄存器 (RSUBC) 中的注释 2	
	修改图 7-17 星期报警寄存器 (ALARMWW) 的格式中的位名称	第十一章 A/D 转换器
	增加 10.3 (1) 外围设备允许寄存器 0 (PER0) 中的注意事项 2	
	修改表 10-2 A/D 转换时间选择	第十一章 D/A 转换器
增加 11.3 (1) 外围设备允许寄存器 0 (PER0) 中的注意事项 2		
增加 12.3 (1) 外围设备允许寄存器 0 (PER0) 中的注意事项 3	第十二章 串行阵列单元	
修改图 12-7 串行通信操作设置寄存器 mn (SCRmn) 的格式		
增加 12.3 (13) 串行输出电平寄存器 m (SOLm) 中的描述		
修改图 12-16 串行输出电平寄存器 m (SOLm) 的格式中的第 1 位和第 3 位		
修改 (a) 串行输出寄存器 m (SOM), (d) 串行输出电平寄存器 m (SOLm) 中的设置, 并修改图 12-66 UART (UART0, UART1, UART2, UART3) 发送寄存器内容的举例中的注释		

版本	描述	章节
第3版	修改图 12-74 UART (UART0, UART1, UART2, UART3) 接收寄存器内容的举例中 (b) 串行输出允许寄存器 m (SOEm) 的设置	第十二章 串行阵列单元
	修改图 12-89 地址域发送的流程图	
	修改图 12-92 数据发送的流程图	
	增加 13.3 (1) 外围设备允许寄存器 0 (PER0) 中的注意事项 2	第十三章 串行接口 IIC0
	修改 13.5.4 (2) 从设备端的选择时钟设置方法中的描述	
	增加 15.4.1 操作过程中 <1> 和 <3> 的描述	第十五章 DMA 控制器
	增加 15.5.5 软件强行中止中的描述	
	增加 15.6 (1) DMA 的优先级中的描述, 并增加注释	
	在下列图中增加 reset 处理时间和时钟供给停止时间 • 图 18-4 通过复位释放 HALT 模式 • 图 18-6 通过产生中断请求释放 STOP 模式 • 图 18-7 通过复位释放 STOP 模式	第十八章 待机功能
	修改图 18-5 释放 STOP 模式时的操作时序 (未屏蔽的中断产生请求时)	
	修改图 19-2 由 RESET 输入进行复位的时序	第十九章 复位功能
	修改图 19-3 由于看门狗定时器溢出进行复位的时序	
	修改图 19-4 在 STOP 模式下通过 RESET 输入进行复位的时序	
	增加图 20-2 由上电清零电路和低电压检测电路产生内部复位信号的时序中的 reset 处理时间	第二十章 上电清零电路
	增加 20.4 上电清零电路使用注意事项	
	增加操作稳定时间	第二十一章 低电压检测电路
	修改图 21-3 低电压检测电平选择寄存器 (LVIS) 的格式中的注意事项 2	
	增加 21.5 低电压检测电路的注意事项	
	修改 23.1.1 (2) 000C1H/010C1H 中的描述	第二十三章 选项字节
	修改图 23-2 用户选项字节 (000C1H/010C1H) 的格式	
	修改图 23-4 片上调试选项字节 (000C3H/010C3H) 的格式	
	增加 24.4.1 (3) 在自编程写入时的描述	第二十四章 FLASH 存储器
	增加 24.5 (1) 背景事件控制寄存器 (BECTL) 中的描述	
	增加 24.6 编程方法	
	增加 24.7 安全设置	
	增加 24.8 通过自编程进行 Flash 存储器编程	
	增加该章内容	第二十五章 片上调试功能
	删除 BCD 调整进位寄存器 (BCDCY 位) 等的描述.	第二十六章 BCD 修正电路
	最大额定值 • 增加稳定器电压 (REGC) • 修改输入电压和输出电压	第二十八章 电气特性 (目标)
	增加 XT1 振荡器特性中的 MIN. 值和 MAX. 值	

版本	描述	章节
第 3 版	DC 特性	第二十八章 电气特性 (目标)
	• 修改输出电流, 高 (I _{OH1}) 的条件和 注释 1	
	• 修改输出电流, 低 (I _{OL1}) 的条件和 注释 2	
	• 修改输入电压, 高 (V _{IH2}) 的条件	
	• 修改输入电压, 低 (V _{IL2}) 的条件	
	• 修改输出电压, 低 (V _{OL1}) 的条件	
	• 增加供给电流	
• 增加看门狗定时器操作电流 (I _{WDT})		
• 增加 A/D 转换器操作电流 (I _{ADC})		
• 增加 D/A 转换器操作电流 (I _{DAC})		
• 增加 DMA 控制器操作电流 (I _{DMA})		
• 增加 LVI 操作电流 (I _{LVI})		
修改 A/D 转换器特性中转换时间 (t _{conv}) 的 MIN. 值		
增加 POC 电路特性		
增加供电电压上升时间		
增加 LVI 电路特性		
增加数据存储器 STOP 模式低电压数据保持特性		
重新修订本章	附录 A 开发工具	
第 4 版	删除连接到 REGC 引脚的电容的目标电容值	通篇
	修改 2.2.18 REGC 中的描述	第二章 引脚功能
	修改表 2-2 未使用引脚的连接中的 P60 ~ P64, P110 和 P111	
	修改图 2-1 引脚 I/O 电路列表 (2/2) 中的 12-D ~ 12-G	
	增加 (地址改变) 表 3-6 扩展 SFR (第二 SFR) 列表 (1/5) 中的 BCDADJ 寄存器	第三章 CPU 结构
	修改图 4-6 P05 和 P06 的框图	第四章 端口功能
	修改图 4-28 P110 和 P111 的框图	
	修改图 4-42 位操作指令 (P10)	
	修改图 5-6 系统时钟控制寄存器 (CKC) 的格式中的 注意事项 2	第五章 时钟发生器
	修改 5.3 (8) 内部高速振荡器微调寄存器 (HIOTRM) 中的描述, 并增加 注意事项	
	修改图 5-9 内部高速振荡器微调寄存器 (HIOTRM) 的格式, 并增加 注意事项	
	修改图 5-13 供电电源打开时 时钟发生器的操作 (当设置 LVI 默认启动功能停止 (选项字节: LVIOFF = 1) 时)	
	增加图 6-5 定时器时钟选择寄存器 0 (TPS0) 的格式中的 注释	第六章 定时器阵列单元
	修改表 6-3 在每种操作模式下 OVF 位操作和设置/清零条件, 并增加 备注	
增加图 6-18 定时器输出寄存器 0 (TO0) 的格式中的 注意事项 2		

版本	描述	章节
第 4 版	修改 6.3 (14) 噪声滤波器允许寄存器 1 (NFEN1) 中的描述	第六章
	修改 6.5.1 TI0n 边沿检测电路	定时器阵列单元
	修改图 7-1 实时计数器的框图	第七章 实时计数器
	增加表 8-4 看门狗定时器窗口打开比例的设置中的注意事项 3	第八章 看门狗定时器
	修复 SOEm3 和 SOE11 位设置为 “0”	第十二章 串行阵列单元
	修复 SOm3, SO11, CKOm3, CKO11, 和 CKO12 位设置为 “1”	
	修改备注中的 “设置禁止 (置为初始值)”	
	修改图 12-1 串行阵列单元 0 的框图	
	修改图 12-2 串行阵列单元 1 的框图	
	增加图 12-5 串行时钟选择寄存器 m (SPSm) 的格式中的设置和注释	
	修改图 12-14 串行输出允许寄存器 m (SOEm) 的格式	
	增加 12.3 (12) 串行输出寄存器 m (SOM) 中的描述	
	修改图 12-15 串行输出寄存器 m (SOM) 的格式	
	增加传输速度的注释	
	修改 12.4.4 从机发送中的传输速度和注释	
	修改 12.4.5 从机接收中的传输速度	
	修改 12.4.6 从机发送/接收中的传输速度	
	修改 12.4.7 (2) 中的注释	
	增加表 12-2 操作时钟选择中的设置和注释	
	修改传输速度, 并增加注释	
	修改图 12-74 UART (UART0, UART1, UART2, UART3) 接收寄存器内容的举例	
	修改图 12-77 重新启动 UART 接收的过程	
	增加表 12-3 操作时钟选择中的设置和注释	
	修改图 12-92 数据发送的流程图	
	增加表 12-4 操作时钟选择中的设置和注释	
	修改图 15-9 UART 连续接收 + ACK 发送设置的示例	第十五章
	增加 15.6 (4) DMA 挂起指令中的描述	DMA 控制器
	修改图 18-4 通过复位释放 HALT 模式	第十八章
	修改图 18-7 通过复位释放 STOP 模式	待机功能
	修改图 19-2 由 RESET 输入进行复位的时序中的 reset 处理	第十九章 复位功能
	修改图 19-4 在 STOP 模式下通过 RESET 输入进行复位的时序中的 reset 处理	
	修改图 19-5 复位控制标志寄存器 (RESF) 的格式中的注意事项 2	
	修改图 20-2 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)	第二十章 上电清零电路
修改图 20-2 由上电清零电路和低电压检测电路产生内部复位信号的时序 (2/2), 并增加注释		
修改图 20-3 复位释放后软件处理过程示例		

版本	描述	章节
第 4 版	修改图 21-2 低电压检测寄存器 (LVIM) 的格式中的注释 4, 并增加注意事项 3	第二十一章
	修改图 21-3 低电压检测电平选择寄存器 (LVIS) 的格式中的注意事项 2	低电压检测电路
	修改 21.4.1 (1) (a) 中的<5>	
	修改图 21-5 低电压检测电路内部复位信号产生时序 (位: LVISEL = 0, 选项字节: LVIOFF = 1) 中的注释 2	
	修改 21.4.1 (1) (b) 中的描述和注意事项	
	修改图 21-6 低电压检测电路内部复位信号产生时序 (位: LVISEL = 0, 选项字节: LVIOFF = 0) 及注释	
	修改 21.4.1 (2) 中的<4>	
	修改图 21-7 低电压检测电路内部复位信号产生时序 (位: LVISEL = 1) 中的注释 2	
	修改 21.4.2 (1) 中的<4>	
	增加图 21-8 低电压检测电路中断信号产生时序 (位: LVISEL = 0, 选项字节: LVIOFF = 1) 中的注释 3	
	修改 21.4.2 (1) (b) 中的描述和注意事项	
	修改图 21-9 低电压检测电路中断信号产生时序 (位: LVISEL = 0, 选项字节: LVIOFF = 0), 并增加注释	
	修改 21.4.2 (2) 中的<4>	
	增加图 21-10 低电压检测电路中断信号产生时序 (位: LVISEL = 1) 中的注释 3	
	修改图 21-11 复位释放后软件处理过程示例	
	修改 22.1 稳压器概述	
	增加图 22-1 稳压器模式控制寄存器 (RMC) 的格式中的注释 3	稳压器
修改 23.1.1 (2) 000C1H/010C1H 中的描述	第二十三章	
修改图 23-2 用户选项字节 (000C1H/010C1H) 的格式和注意事项 2	选项字节	
修改 24.4.5 REGC 引脚中的描述	第二十四章	
增加 24.8 通过自编程进行 Flash 存储器编程中的注意事项 4	FLASH 存储器	
增加 25.3 用户资源的保留	第二十五章 片上调试功能	
通篇修改	第二十八章 电气特性 (目标)	

[MEMO]

详细信息请联系:

(中国区)

网址:

<http://www.cn.necel.com/>

<http://www.necel.com/>

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号
量子芯座 7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39
楼 3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230