

# R7F0C01072DNP、 R7F0C010B2DFP-C

用户手册 硬件篇

16位单芯片微控制器

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。  
请通过瑞萨电子的主页确认发布的最新信息。

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## 注意事项

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，瑞萨电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，瑞萨电子已尽量做到合理注意，但是，瑞萨电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，瑞萨电子不承担任何责任。
3. 对于因使用本文档中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，瑞萨电子不承担任何责任。本文档所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 用户不得更改、修改、复制或以其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或以其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
5. 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：

标准等级： 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。

高质量等级： 运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。

瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。

6. 使用本文档中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机软件单独进行评估，所以请用用户自行对最终产品或系统进行安全评估。
8. 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相应法律法规（包括但不限于《欧盟 RoHS 指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文档中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的任何目的（如大规模杀伤性武器的开发等）。在将本文档中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
10. 向第三方分销或处分产品或者以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文档规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
11. 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文档。
12. 如果对本文档所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何其他疑问，请向瑞萨电子的营业部门咨询。

（注 1） 瑞萨电子：在本文档中指瑞萨电子株式会社及其控股子公司。

（注 2） 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。

## 关于 CMOS 器件的注意事项

### ① 输入引脚处的施加电压波形

输入噪声或由反射波引起的波形失真可能导致故障发生。如果由于噪声等影响，使CMOS器件的输入电压范围处于在 $V_{IL}$  (MAX) 和 $V_{IH}$  (MIN) 之间，器件可能发生故障。在输入电平固定时以及输入电平从 $V_{IL}$  (MAX) 到 $V_{IH}$  (MIN) 的过渡期间，要谨防颤振噪声进入器件。

### ② 未使用的输入引脚的处理

CMOS器件上未连接的输入端可能是故障源。如果一个输入引脚未被连接，则由于噪声等原因可能会产生内部输入电平，从而导致故障。CMOS器件的工作方式与双极性或NMOS器件不同。CMOS器件的输入电平必须借助上拉或下拉电路固定于高电平或低电平。每一个未使用引脚只要有可能成为输出引脚时，都应该通过附加电阻连接到 $V_{DD}$ 或GND。对未使用引脚的处理因器件而不同，必须遵循与器件相关的规格和说明。

### ③ ESD 防护措施

如果MOS器件周围有强电场，将会击穿氧化栅极，降低器件的工作性能。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。环境必须控制适当。如果空气干燥，应当使用加湿器。建议避免使用容易产生静电的绝缘体。半导体器件的存放和运输必须使用抗静电容器、静电屏蔽袋或导电材料包装。所有包括工作台和工作面的测试和测量工具必须良好接地。操作员应当佩戴手腕带以保证良好接地。不能用手直接接触半导体器件。对装配有半导体器件的PW板也应采取类似的静电防范措施。

### ④ 初始化之前的状态

上电并不一定定义MOS器件的初始状态。刚接通电源时，具有复位功能的MOS器件并没有被初始化。因此上电不能保证输出引脚的电平、输入/输出设置和寄存器的内容。器件在收到复位信号后才进行初始化。具有复位功能的器件在上电后必须立即进行复位操作。

### ⑤ 电源上电 / 断电序列

器件内部工作和外部接口使用不同电源的情况下，原则上应先在接通内部电源之后再接通外部电源。当关闭电源时，原则上先关闭外部电源再关闭内部电源。如果电源开关顺序相反，可能会对器件的内部元件施加电压，从而由于异常电流的流过而造成故障和降低元件的性能。须视具体器件和支配器件的相关规格来单独决定正确的上电/断电序列。

### ⑥ 断电状态期间的信号输入

不要在器件断电时输入信号或输入/输出上拉电源。因为输入信号或提供输入/输出上拉电源将引起电流注入，从而引起器件的误操作，并且此时流过器件的异常电流引起内部元件性能劣化。须视具体器件和支配器件的相关规格来单独决定断电状态期间的信号输入。

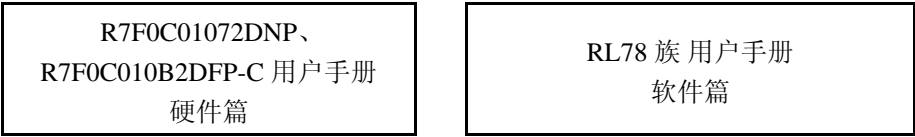
# 本手册的使用方法

对 象      本手册是以理解 R7F0C01072DNP、R7F0C010B2DFP-C 的功能并且设计和开发其应用系统和程序的用户工程师为对象。  
对象产品如下：

- 24 引脚：R7F0C01072DNP
- 32 引脚：R7F0C010B2DFP-C

目 的      本手册以帮助用户理解以下构成所示的功能为目的。

构 成      R7F0C01072DNP、R7F0C010B2DFP-C 的手册分为本手册和指令篇（RL78 族共通）共 2 册。



- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• 引脚功能</li><li>• 内部块功能</li><li>• 中断</li><li>• 其他的内部外围功能</li><li>• 电特性</li></ul> | <ul style="list-style-type: none"><li>• CPU 功能</li><li>• 指令集</li><li>• 指令的说明</li></ul> |
|---|--|

阅读方法      本手册的读者必须具备电气、逻辑电路和微控制器的基础知识。

- 要理解全部功能时  
→ 按照目录的顺序阅读本手册。
- 寄存器格式的阅读方法  
→ 关于方框（□）内的位号，其位的名称在汇编程序中被定义为保留字，而在编译程序中被 #pragma sfr 指令定义为 sfr 变量。
- 要详细了解 R7F0C01072DNP、R7F0C010B2DFP-C 微控制器的指令功能时  
→ 请参照另一册《RL78 族 用户手册 软件篇》（R01US0015E）。

- 凡例
- 数据的表示

有效低电平的表示

注

注意

备注

数字的表示
- : 左侧为高位，右侧为低位。

: xxx （在引脚或者信号名称上有上划线）

: 正文中加注的说明

: 阅读时需要特别注意的内容

: 征文的补充说明

: 二进制数 ..... xxxx 或者 xxxxB  
十进制数 ..... xxxx  
十六进制数 ..... xxxxH

相关资料 相关资料可能是暂定版，但是没有“暂定”的表示。

芯片的相关资料

资料名称	资料号
R7F0C01072DNP、R7F0C010B2DFP-C 用户手册 硬件篇	本手册
RL78 族 用户手册 指令篇	R01US0015E

闪存编程器的相关资料 ( 用户手册 )

资料名称	资料号
PG-FP5 闪存编程器 用户手册	R20UT0008E

注意 上述的相关资料可能会有改动，恕不另行通知。设计时请使用最新版本的资料。

其他资料

资料名称	资料号
瑞萨 单片机综合目录	R01CS0001E
半导体封装 安装手册	注
静电放电 (ESD) 损坏对策指南	C11892E

注 请参阅“半导体封装安装手册”网页。  
(<http://www.renesas.com/products/package/manual/index.jsp>)

注意 上述的相关资料可能会有改动，恕不另行通知。设计时请使用最新版本的资料。

# 目 录

第 1 章	概述 .....	1
1.1	特点 .....	1
1.2	产品型号一览表 .....	2
1.3	引脚连接图 (Top View) .....	3
1.3.1	24 引脚产品 .....	3
1.3.2	32 引脚产品 .....	4
1.4	引脚名 .....	5
1.5	框图 .....	6
1.5.1	24 引脚产品 .....	6
1.5.2	32 引脚产品 .....	7
1.6	功能概要 .....	8
第 2 章	引脚功能 .....	10
2.1	引脚功能一览表 .....	10
2.1.1	24 引脚产品 .....	10
2.1.2	32 引脚产品 .....	11
2.2	端口以外的引脚 .....	12
2.2.1	各产品配置的功能 .....	12
2.2.2	功能说明 .....	13
2.3	引脚功能的说明 .....	15
2.3.1	P10 ~ P17 (Port 1) .....	15
2.3.2	P20 ~ P27 (Port 2) .....	16
2.3.3	P30 ~ P35 (Port 3) .....	17
2.3.4	P40 (Port 4) .....	18
2.3.5	P60、P61 (Port 6) .....	18
2.3.6	P121、P122 (Port 12) .....	19
2.3.7	P137 (Port 13) .....	19
2.3.8	$V_{DD}$ 、 $V_{SS}$ .....	20
2.3.9	RESET .....	20
2.3.10	REGC .....	20
2.4	未使用引脚的处理 .....	21
第 3 章	CPU 体系结构 .....	24
3.1	存储空间 .....	24
3.1.1	内部程序存储空间 .....	28
3.1.2	镜像区 .....	30
3.1.3	内部数据存储空间 .....	32
3.1.4	特殊功能寄存器 (SFR: Special Function Register) 的区域 .....	32
3.1.5	扩展特殊功能寄存器 (2nd SFR: 2nd Special Function Register) 的区域 .....	32
3.1.6	数据存储器的寻址 .....	32
3.2	处理器的寄存器 .....	34
3.2.1	控制寄存器 .....	34
3.2.2	通用寄存器 .....	36
3.2.3	ES 寄存器和 CS 寄存器 .....	37
3.2.4	特殊功能寄存器 (SFR: Special Function Register) .....	38
3.2.5	扩展特殊功能寄存器 (2nd SFR: 2nd Special Function Register) .....	42
3.3	指令地址的寻址 .....	47
3.3.1	相对寻址 .....	47
3.3.2	立即寻址 .....	47



3.3.3	表间接寻址 .....	48
3.3.4	寄存器直接寻址 .....	48
3.4	处理数据地址的寻址 .....	49
3.4.1	隐含寻址 .....	49
3.4.2	寄存器寻址 .....	49
3.4.3	直接寻址 .....	50
3.4.4	短直接寻址 .....	51
3.4.5	SFR 寻址 .....	52
3.4.6	寄存器间接寻址 .....	53
3.4.7	基址寻址 .....	54
3.4.8	基址变址寻址 .....	58
3.4.9	堆栈寻址 .....	59
<b>第 4 章</b>	<b>端口功能 .....</b>	<b>63</b>
4.1	端口功能 .....	63
4.2	端口结构 .....	63
4.2.1	端口 1 .....	64
4.2.2	端口 2 .....	69
4.2.3	端口 3 .....	73
4.2.4	端口 4 .....	79
4.2.5	端口 6 .....	81
4.2.6	端口 12 .....	83
4.2.7	端口 13 .....	84
4.3	控制端口功能的寄存器 .....	85
4.3.1	端口模式寄存器 (PMxx) .....	87
4.3.2	端口寄存器 (Pxx) .....	88
4.3.3	上拉电阻选择寄存器 (PUxx) .....	89
4.3.4	端口模式控制寄存器 1 (PMC1) (只限于 24 引脚产品) .....	90
4.3.5	A/D 端口配置寄存器 (ADPC) .....	90
4.4	端口功能的运行 .....	91
4.4.1	输入 / 输出端口的写操作 .....	91
4.4.2	输入 / 输出端口的读操作 .....	91
4.4.3	输入 / 输出端口的运算 .....	91
4.5	使用复用功能时的端口相关寄存器的设定 .....	92
4.6	使用端口功能时的注意事项 .....	95
4.6.1	有关对端口寄存器 n (Pn) 的位存储器操作指令的注意事项 .....	95
4.6.2	设定引脚时的注意事项 .....	96
<b>第 5 章</b>	<b>时钟发生电路 .....</b>	<b>97</b>
5.1	时钟发生电路的功能 .....	97
5.2	时钟发生电路的结构 .....	98
5.3	控制时钟发生电路的寄存器 .....	100
5.3.1	时钟运行模式控制寄存器 (CMC) .....	101
5.3.2	系统时钟控制寄存器 (CKC) .....	102
5.3.3	时钟运行状态控制寄存器 (CSC) .....	103
5.3.4	振荡稳定时间计数器的状态寄存器 (OSTC) .....	104
5.3.5	振荡稳定时间选择寄存器 (OSTS) .....	106
5.3.6	外围允许寄存器 0、1 (PER0、PER1) .....	107
5.3.7	高速内部振荡器的频率选择寄存器 (HOCODIV) .....	109
5.3.8	高速内部振荡器的微调寄存器 (HIOTRM) .....	110
5.4	系统时钟振荡电路 .....	111
5.4.1	X1 振荡电路 .....	111

5.4.2	高速内部振荡器 .....	114
5.4.3	低速内部振荡器 .....	114
5.5	时钟发生电路的运行 .....	115
5.6	时钟控制 .....	117
5.6.1	高速内部振荡器的设定例子 .....	117
5.6.2	X1 振荡电路的设定例子 .....	118
5.6.3	CPU 时钟的状态转移图 .....	119
5.6.4	CPU 时钟转移前的条件和转移后的处理 .....	123
5.6.5	CPU 时钟和系统时钟的切换所需时间 .....	124
5.6.6	时钟振荡停止前的条件 .....	124
5.7	谐振器和振荡电路常数 .....	125
<b>第 6 章</b>	<b>定时器阵列单元 .....</b>	<b>126</b>
6.1	定时器阵列单元的功能 .....	127
6.1.1	独立通道运行功能 .....	127
6.1.2	多通道联动运行功能 .....	128
6.1.3	8 位定时器运行功能（只限于通道 1 和通道 3） .....	129
6.2	定时器阵列单元的结构 .....	130
6.2.1	定时器计数寄存器 mn（TCRmn） .....	134
6.2.2	定时器数据寄存器 mn（TDRmn） .....	136
6.3	控制定时器阵列单元的寄存器 .....	137
6.3.1	外围允许寄存器 0（PER0） .....	138
6.3.2	定时器时钟选择寄存器 m（TPSm） .....	139
6.3.3	定时器模式寄存器 mn（TMRmn） .....	142
6.3.4	定时器状态寄存器 mn（TSRmn） .....	147
6.3.5	定时器通道允许状态寄存器 m（TEm） .....	148
6.3.6	定时器通道开始寄存器 m（TSM） .....	149
6.3.7	定时器通道停止寄存器 m（TTm） .....	150
6.3.8	定时器输入选择寄存器 0（TIS0） .....	151
6.3.9	定时器输出允许寄存器 m（TOEm） .....	152
6.3.10	定时器输出寄存器 m（TOM） .....	153
6.3.11	定时器输出电平寄存器 m（TOLm） .....	154
6.3.12	定时器输出模式寄存器 m（TOMm） .....	155
6.3.13	噪声滤波器允许寄存器 1（NFEN1） .....	156
6.3.14	端口模式寄存器 1、3（PM1、PM3） .....	157
6.4	定时器阵列单元的基本规则 .....	158
6.4.1	多通道联动运行功能的基本规则 .....	158
6.4.2	8 位定时器运行功能的基本规则（只限于通道 1 和通道 3） .....	160
6.5	计数器的运行 .....	161
6.5.1	计数时钟（fTCLK） .....	161
6.5.2	计数器的开始时序 .....	163
6.5.3	计数器的运行 .....	164
6.6	通道输出（TOMn 引脚）的控制 .....	169
6.6.1	TOMn 引脚输出电路的结构 .....	169
6.6.2	TOMn 引脚的输出设定 .....	170
6.6.3	通道输出运行的注意事项 .....	171
6.6.4	TOMn 位的一次性操作 .....	175
6.6.5	有关开始计数时的定时器中断和 TOMn 引脚输出 .....	176
6.7	定时器输入（TIMn）的控制 .....	177
6.7.1	TIMn 引脚输入电路的结构 .....	177
6.7.2	噪声滤波器 .....	177
6.7.3	操作通道输入时的注意事项 .....	178

6.8	定时器阵列单元的独立通道运行功能 .....	179
6.8.1	作为间隔定时器 / 方波输出的运行 .....	179
6.8.2	作为外部事件计数器的运行 .....	184
6.8.3	作为输入脉冲间隔测量的运行 .....	188
6.8.4	作为输入信号高低电平宽度测量的运行 .....	192
6.8.5	作为延迟计数器的运行 .....	196
6.9	定时器阵列单元的多通道联动运行功能 .....	200
6.9.1	作为单触发脉冲输出功能的运行 .....	200
6.9.2	作为 PWM 功能的运行 .....	207
6.9.3	作为多重 PWM 输出功能的运行 .....	214
6.10	使用定时器阵列单元时的注意事项 .....	222
6.10.1	使用定时器输出时的注意事项 .....	222
<b>第 7 章</b>	<b>时钟输出 / 蜂鸣器输出控制电路 .....</b>	<b>223</b>
7.1	时钟输出 / 蜂鸣器输出控制电路的功能 .....	223
7.2	时钟输出 / 蜂鸣器输出控制电路的结构 .....	224
7.3	控制时钟输出 / 蜂鸣器输出控制电路的寄存器 .....	225
7.3.1	时钟输出选择寄存器 n (CKSn) .....	225
7.3.2	控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器 .....	226
7.4	时钟输出 / 蜂鸣器输出控制电路的运行 .....	227
7.4.1	输出引脚的运行 .....	227
7.5	时钟输出 / 蜂鸣器输出控制电路的注意事项 .....	227
<b>第 8 章</b>	<b>看门狗定时器 .....</b>	<b>228</b>
8.1	看门狗定时器的功能 .....	228
8.2	看门狗定时器的结构 .....	229
8.3	控制看门狗定时器的寄存器 .....	230
8.3.1	看门狗定时器的允许寄存器 (WDTE) .....	230
8.4	看门狗定时器的运行 .....	231
8.4.1	看门狗定时器的运行控制 .....	231
8.4.2	看门狗定时器上溢时间的设定 .....	232
8.4.3	看门狗定时器窗口打开期间的设定 .....	233
8.4.4	看门狗定时器间隔中断的设定 .....	234
<b>第 9 章</b>	<b>A/D 转换器 .....</b>	<b>235</b>
9.1	A/D 转换器的功能 .....	235
9.2	A/D 转换器的结构 .....	237
9.3	控制 A/D 转换器的寄存器 .....	239
9.3.1	外围允许寄存器 0 (PER0) .....	240
9.3.2	A/D 转换器的模式寄存器 0 (ADM0) .....	241
9.3.3	A/D 转换器的模式寄存器 1 (ADM1) .....	249
9.3.4	A/D 转换器的模式寄存器 2 (ADM2) .....	250
9.3.5	12 位 A/D 转换结果寄存器 (ADCR) .....	252
9.3.6	8 位 A/D 转换结果寄存器 (ADCRH) .....	253
9.3.7	模拟输入通道指定寄存器 (ADS) .....	254
9.3.8	转换结果比较上限值设定寄存器 (ADUL) .....	256
9.3.9	转换结果比较下限值设定寄存器 (ADLL) .....	256
9.3.10	A/D 测试寄存器 (ADTES) .....	257
9.3.11	控制模拟输入引脚端口功能的寄存器 .....	257
9.4	A/D 转换器的转换运行 .....	258
9.5	输入电压和转换结果 .....	260
9.6	A/D 转换器的运行模式 .....	261

9.6.1	软件触发模式（选择模式、连续转换模式） .....	261
9.6.2	软件触发模式（选择模式、单次转换模式） .....	262
9.6.3	软件触发模式（扫描模式、连续转换模式） .....	263
9.6.4	软件触发模式（扫描模式、单次转换模式） .....	264
9.6.5	硬件触发无等待模式（选择模式、连续转换模式） .....	265
9.6.6	硬件触发无等待模式（选择模式、单次转换模式） .....	266
9.6.7	硬件触发无等待模式（扫描模式、连续转换模式） .....	267
9.6.8	硬件触发无等待模式（扫描模式、单次转换模式） .....	268
9.6.9	硬件触发等待模式（选择模式、连续转换模式） .....	269
9.6.10	硬件触发等待模式（选择模式、单次转换模式） .....	270
9.6.11	硬件触发等待模式（扫描模式、连续转换模式） .....	271
9.6.12	硬件触发等待模式（扫描模式、单次转换模式） .....	272
9.7	A/D 转换器的设定流程图 .....	273
9.7.1	软件触发模式的设定 .....	273
9.7.2	硬件触发无等待模式的设定 .....	274
9.7.3	硬件触发等待模式的设定 .....	275
9.7.4	选择温度传感器的输出电压 / 内部基准电压时的设定 （以软件触发模式、单次转换模式为例） .....	276
9.7.5	测试模式的设定 .....	277
9.8	SNOOZE 模式功能 .....	278
9.9	A/D 转换器特性表的阅读方法 .....	282
9.10	A/D 转换器的注意事项 .....	285
第 10 章	D/A 转换器 .....	289
10.1	D/A 转换器的功能 .....	289
10.2	D/A 转换器的结构 .....	289
10.3	D/A 转换器使用的寄存器 .....	291
10.3.1	A/D 端口配置寄存器（ADPC） .....	291
10.3.2	外围允许寄存器 1（PER1） .....	292
10.3.3	D/A 转换器的模式寄存器（DAM） .....	293
10.3.4	D/A 转换值设定寄存器 i（DACSi）（i=0、1） .....	293
10.3.5	端口模式寄存器 2（PM2） .....	294
10.4	D/A 转换器的运行 .....	295
10.4.1	通常模式中的运行 .....	295
10.4.2	实时输出模式中的运行 .....	296
10.5	使用 D/A 转换器时的注意事项 .....	297
第 11 章	串行阵列单元 .....	298
11.1	串行阵列单元的功能 .....	298
11.1.1	3 线串行 I/O（CSI00） .....	298
11.1.2	UART（UART0） .....	299
11.2	串行阵列单元的结构 .....	300
11.2.1	移位寄存器 .....	302
11.2.2	串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位 .....	302
11.3	控制串行阵列单元的寄存器 .....	304
11.3.1	外围允许寄存器 0（PER0） .....	305
11.3.2	串行时钟选择寄存器 m（SPSm） .....	306
11.3.3	串行模式寄存器 mn（SMRmn） .....	307
11.3.4	串行通信运行设定寄存器 mn（SCRmn） .....	308
11.3.5	串行数据寄存器 mn（SDRmn）的高 7 位 .....	311
11.3.6	串行标志清除触发寄存器 mn（SIRmn） .....	312
11.3.7	串行状态寄存器 mn（SSRmn） .....	313

11.3.8	串行通道开始寄存器 m (SSm) .....	315
11.3.9	串行通道停止寄存器 m (STm) .....	316
11.3.10	串行通道允许状态寄存器 m (SEm) .....	317
11.3.11	串行输出允许寄存器 m (SOEm) .....	318
11.3.12	串行输出寄存器 m (SOM) .....	319
11.3.13	串行输出电平寄存器 m (SOLm) .....	320
11.3.14	串行待机控制寄存器 m (SSCm) .....	321
11.3.15	输入切换控制寄存器 (ISC) .....	322
11.3.16	噪声滤波器允许寄存器 0 (NFEN0) .....	323
11.3.17	端口模式寄存器 3 (PM3) .....	324
11.4	运行停止模式 .....	325
11.4.1	以单元为单位停止运行的情况 .....	325
11.4.2	按通道停止运行的情况 .....	326
11.5	3 线串行 I/O (CSI00) 通信的运行 .....	327
11.5.1	主控发送 .....	328
11.5.2	主控接收 .....	336
11.5.3	主控的发送和接收 .....	344
11.5.4	从属发送 .....	352
11.5.5	从属接收 .....	360
11.5.6	从属发送和接收 .....	366
11.5.7	SNOOZE 模式功能 (CSI00) .....	374
11.5.8	传送时钟频率的计算 .....	378
11.5.9	在 3 线串行 I/O (CSI00) 通信过程中发生错误时的处理步骤 .....	380
11.6	从属选择输入功能的时钟同步串行通信的运行 .....	381
11.6.1	从属发送 .....	384
11.6.2	从属接收 .....	394
11.6.3	从属的发送和接收 .....	401
11.6.4	传送时钟频率的计算 .....	411
11.6.5	在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤 .....	412
11.7	UART (UART0) 通信的运行 .....	413
11.7.1	UART 发送 .....	414
11.7.2	UART 接收 .....	423
11.7.3	SNOOZE 模式功能 .....	430
11.7.4	波特率的计算 .....	436
11.7.5	在 UART (UART0) 通信过程中发生错误时的处理步骤 .....	440
第 12 章	串行接口 IICA .....	441
12.1	串行接口 IICA 的功能 .....	441
12.2	串行接口 IICA 的结构 .....	444
12.3	控制串行接口 IICA 的寄存器 .....	447
12.3.1	外围允许寄存器 0 (PER0) .....	447
12.3.2	IICA 控制寄存器 n0 (IICCTLn0) .....	448
12.3.3	IICA 状态寄存器 n (IICSn) .....	453
12.3.4	IICA 标志寄存器 n (IICFn) .....	455
12.3.5	IICA 控制寄存器 n1 (IICCTLn1) .....	457
12.3.6	IICA 低电平宽度设定寄存器 n (IICWLn) .....	459
12.3.7	IICA 高电平宽度设定寄存器 n (IICWHn) .....	459
12.3.8	端口模式寄存器 6 (PM6) .....	460
12.4	I <sup>2</sup> C 总线模式的功能 .....	461
12.4.1	引脚结构 .....	461
12.4.2	通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法 .....	462
12.5	I <sup>2</sup> C 总线的定义和控制方法 .....	463

12.5.1	开始条件 .....	463
12.5.2	地址 .....	464
12.5.3	传送方向的指定 .....	464
12.5.4	应答 (ACK) .....	465
12.5.5	停止条件 .....	466
12.5.6	等待 .....	467
12.5.7	等待的解除方法 .....	469
12.5.8	中断请求 (INTIICAn) 的产生时序和等待控制 .....	470
12.5.9	地址匹配的检测方法 .....	471
12.5.10	错误的检测 .....	471
12.5.11	扩展码 .....	472
12.5.12	仲裁 .....	473
12.5.13	唤醒功能 .....	475
12.5.14	通信预约 .....	478
12.5.15	其他注意事项 .....	482
12.5.16	通信运行 .....	483
12.5.17	I <sup>2</sup> C 中断请求 (INTIICAn) 的产生时序 .....	491
12.6	时序图 .....	512
<b>第 13 章</b>	<b>DMA 控制器 .....</b>	<b>527</b>
13.1	DMA 控制器的功能 .....	527
13.2	DMA 控制器的结构 .....	528
13.2.1	DMA SFR 地址寄存器 n (DSAn) .....	528
13.2.2	DMA RAM 地址寄存器 n (DRAn) .....	529
13.2.3	DMA 字节计数寄存器 n (DBCn) .....	530
13.3	控制 DMA 控制器的寄存器 .....	531
13.3.1	DMA 模式控制寄存器 n (DMCn) .....	531
13.3.2	DMA 运行控制寄存器 n (DRCn) .....	533
13.4	DMA 控制器的运行 .....	534
13.4.1	运行步骤 .....	534
13.4.2	传送模式 .....	535
13.4.3	DMA 传送的结束 .....	535
13.5	DMA 控制器的设定例子 .....	536
13.5.1	CSI 连续发送 .....	536
13.5.2	A/D 转换结果的连续读取 .....	538
13.5.3	UART 连续接收 +ACK 发送 .....	540
13.5.4	通过 DWAITn 位保留 DMA 传送 .....	541
13.5.5	通过软件强制结束 .....	542
13.6	DMA 控制器的注意事项 .....	544
<b>第 14 章</b>	<b>事件链接控制器 (ELC) .....</b>	<b>546</b>
14.1	ELC 的功能 .....	546
14.2	ELC 的结构 .....	546
14.3	控制 ELC 的寄存器 .....	547
14.3.1	事件输出目标选择寄存器 n (ELSELRn) (n=00 ~ 09) .....	547
14.4	运行说明 .....	549
<b>第 15 章</b>	<b>中断功能 .....</b>	<b>550</b>
15.1	中断功能的种类 .....	550
15.2	中断源和结构 .....	550
15.3	控制中断功能的寄存器 .....	554
15.3.1	中断请求标志寄存器 (IF0L、IF0H、IF1L) .....	556

15.3.2	中断屏蔽标志寄存器 (MK0L、MK0H、MK1L) .....	557
15.3.3	优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR10L、PR10H、PR11L) .....	557
15.3.4	外部中断上升沿允许寄存器 (EGP0) 和外部中断下降沿允许寄存器 (EGN0) .....	559
15.3.5	程序状态字 (PSW) .....	560
15.4	中断处理的操作 .....	561
15.4.1	可屏蔽中断请求的接受 .....	561
15.4.2	软件中断请求的接受 .....	564
15.4.3	多重中断处理 .....	564
15.4.4	中断请求的保留 .....	568
第 16 章	待机功能 .....	569
16.1	待机功能 .....	569
16.2	控制待机功能的寄存器 .....	570
16.2.1	振荡稳定时间计数器的状态寄存器 (OSTC) .....	571
16.2.2	振荡稳定时间选择寄存器 (OSTS) .....	572
16.3	待机功能的运行 .....	573
16.3.1	HALT 模式 .....	573
16.3.2	STOP 模式 .....	577
16.3.3	SNOOZE 模式 .....	582
第 17 章	复位功能 .....	584
17.1	复位时序 .....	586
17.2	确认复位源的寄存器 .....	592
17.2.1	复位控制标志寄存器 (RESF) .....	592
第 18 章	上电复位电路 .....	595
18.1	上电复位电路的功能 .....	595
18.2	上电复位电路的结构 .....	595
18.3	上电复位电路的运行 .....	596
18.4	上电复位电路的注意事项 .....	599
第 19 章	电压检测电路 .....	601
19.1	电压检测电路的功能 .....	601
19.2	电压检测电路的结构 .....	602
19.3	控制电压检测电路的寄存器 .....	603
19.3.1	电压检测寄存器 (LVIM) .....	603
19.3.2	电压检测电平寄存器 (LVIS) .....	604
19.4	电压检测电路的运行 .....	606
19.4.1	用作复位模式时的设定 .....	606
19.4.2	用作中断模式时的设定 .....	608
19.4.3	用作中断 & 复位模式时的设定 .....	610
19.5	电压检测电路的注意事项 .....	616
第 20 章	安全功能 .....	618
20.1	安全功能的概要 .....	618
20.2	安全功能使用的寄存器 .....	619
20.3	闪存 CRC 运算功能 (高速 CRC) 的运行 .....	619
20.3.1	闪存 CRC 控制寄存器 (CRC0CTL) .....	620
20.3.2	闪存 CRC 运算结果寄存器 (PGCRCL) .....	620
20.3.3	操作流程 .....	621
20.4	CRC 运算功能 (通用 CRC) .....	622
20.4.1	CRC 输入寄存器 (CRCIN) .....	622

20.4.2	CRC 数据寄存器 (CRCD)	623
20.4.3	操作流程	623
20.5	RAM 奇偶校验错误检测功能	624
20.5.1	RAM 奇偶校验错误控制寄存器 (RPECTL)	624
20.6	RAM 保护功能	626
20.6.1	非法存储器存取检测控制寄存器 (IAWCTL)	626
20.7	SFR 保护功能	627
20.7.1	非法存储器存取检测控制寄存器 (IAWCTL)	627
20.8	非法存储器存取检测功能	628
20.8.1	非法存储器存取检测控制寄存器 (IAWCTL)	629
20.9	频率检测功能	630
20.9.1	定时器输入选择寄存器 0 (TIS0)	631
20.10	A/D 测试功能	632
20.10.1	A/D 测试寄存器 (ADTES)	633
20.10.2	模拟输入通道指定寄存器 (ADS)	634
第 21 章	稳压器	635
21.1	稳压器的概要	635
第 22 章	选项字节	636
22.1	选项字节的功能	636
22.1.1	用户选项字节 (000C0H ~ 000C2H)	636
22.1.2	片上调试选项字节 (000C3H)	637
22.2	用户选项字节的格式	638
22.3	片上调试选项字节的格式	641
22.4	选项字节的设定	642
第 23 章	闪存	643
23.1	使用闪存编程器的串行编程	644
23.1.1	编程环境	646
23.1.2	通信方式	646
23.2	使用外部器件 (内置 UART) 的串行编程	647
23.2.1	编程环境	647
23.2.2	通信方式	648
23.3	电路板上的引脚处理	649
23.3.1	P40/TOOL0 引脚	649
23.3.2	RESET 引脚	649
23.3.3	端口引脚	650
23.3.4	REGC 引脚	650
23.3.5	X1 引脚和 X2 引脚	650
23.3.6	电源	650
23.4	数据闪存	651
23.4.1	数据闪存的概要	651
23.4.2	控制数据闪存的寄存器	652
23.4.3	数据闪存的存取步骤	653
23.5	编程方法	654
23.5.1	闪存的控制	654
23.5.2	闪存编程模式	655
23.5.3	通信方式	656
23.5.4	通信命令	657
23.5.5	特征数据的说明	658
23.6	安全设定	659



23.7 通过自编程进行的闪存编程 .....	661
23.7.1 闪存屏蔽窗口功能 .....	664
23.8 使用 PG-FP5 时的各命令处理时间（参考值） .....	665
<b>第 24 章 片上调试功能 .....</b>	<b>666</b>
24.1 E1 片上调试仿真器和 R7F0C010 的连接 .....	666
24.2 片上调试安全 ID .....	667
24.3 用户资源的确保 .....	667
<b>第 25 章 十进制校正（BCD）电路 .....</b>	<b>669</b>
25.1 十进制校正电路的功能 .....	669
25.2 十进制校正电路使用的寄存器 .....	669
25.2.1 BCD 校正结果寄存器（BCDADJ） .....	669
25.3 十进制校正电路的运行 .....	670
<b>第 26 章 指令集的概要 .....</b>	<b>672</b>
26.1 凡例 .....	672
26.1.1 操作数的表现形式和记述方法 .....	672
26.1.2 操作栏的说明 .....	673
26.1.3 标志栏的说明 .....	674
26.1.4 PREFIX 指令 .....	674
26.2 操作一览表 .....	675
<b>第 27 章 电特性 .....</b>	<b>693</b>
27.1 绝对最大额定值 .....	694
27.2 振荡电路特性 .....	696
27.2.1 X1 振荡电路特性 .....	696
27.2.2 内部振荡器特性 .....	696
27.3 DC 特性 .....	697
27.3.1 引脚特性 .....	697
27.3.2 电源电流特性 .....	700
27.4 AC 特性 .....	703
27.5 外围功能特性 .....	707
27.5.1 串行阵列单元 .....	707
27.5.2 串行接口 IICA .....	713
27.5.3 片上调试（UART） .....	714
27.6 模拟特性 .....	714
27.6.1 A/D 转换器特性 .....	714
27.6.2 温度传感器 / 内部基准电压特性 .....	717
27.6.3 D/A 转换器 .....	717
27.6.4 POR 电路特性 .....	717
27.6.5 LVD 电路特性 .....	718
27.7 STOP 模式中的数据存储器器的低电源电压数据保持特性 .....	719
27.8 闪存编程特性 .....	719
27.9 闪存编程模式的进入时序 .....	720
<b>第 28 章 封装尺寸图 .....</b>	<b>721</b>
28.1 24 引脚产品 .....	721
28.2 32 引脚产品 .....	722
<b>附录 A 修订记录 .....</b>	<b>723</b>

## 第 1 章 概述

### 1.1 特点

- 能更改高速（0.03125 $\mu$ s: 高速内部振荡器时钟 32MHz 运行）到低速（1.0 $\mu$ s: 高速内部振荡器时钟 1MHz 运行）的最短指令执行时间。
- 通用寄存器: 8 位 $\times$ 32 个寄存器（8 位 $\times$ 8 个寄存器 $\times$ 4 组）
- ROM: 16KB, RAM: 1.5KB, 数据闪存: 2KB
- 内置高速内部振荡器时钟。
  - 可选择 32MHz(TYP.)、24MHz(TYP.)、16MHz(TYP.)、12MHz(TYP.)、8MHz(TYP.)、4MHz(TYP.) 或者 1MHz(TYP.)。
- 内置单电源闪存（有块擦除/编程禁止功能）。
- 支持自编程功能。
- 内置片上调试功能。
- 内置上电复位（POR）电路和电压检测电路（LVD）。
- 内置看门狗定时器（能以专用的低速内部振荡器时钟运行）。
- 内置时钟输出/蜂鸣器输出的控制电路。
- 内置十进制校正（BCD）电路。
- I/O 端口: 26 个或者 28 个
- 定时器
  - 16 位定时器 TAU : 4 个通道
  - 看门狗定时器 : 1 个通道
- 串行接口
  - CSI
  - UART
  - I<sup>2</sup>C
- 8/12 位分辨率 A/D 转换器（V<sub>DD</sub>=2.7~3.6V）: 6 个或者 8 个通道
- 待机功能: HALT、STOP、SNOOZE 模式
- 内置 D/A 转换器。
- DMA 控制器: 2 个通道
- 内置事件链接控制器（ELC）。
- 电源电压: V<sub>DD</sub>=2.7~3.6V
- 工作环境温度: T<sub>A</sub>=-40~+85°C

备注 内置的功能因产品而不同。请参照“1.6 功能概要”。

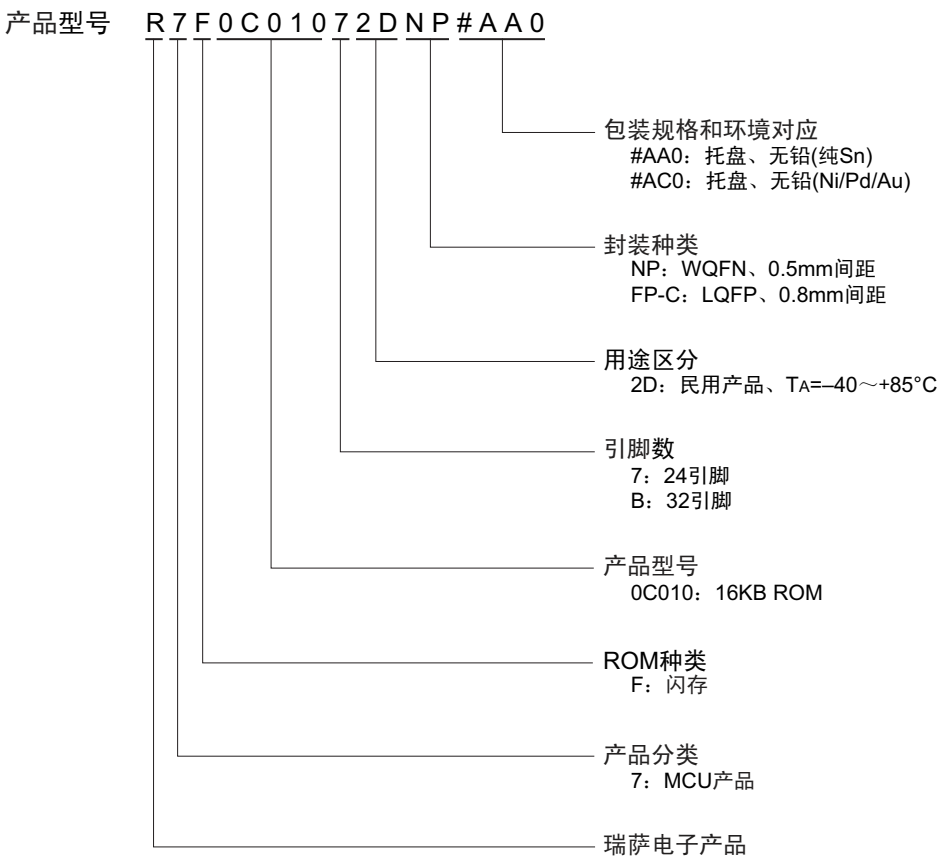
- ROM、RAM 容量

闪存 ROM	数据闪存	RAM	24 引脚	32 引脚
16KB	2KB	1.5KB	R7F0C01072DNP	R7F0C010B2DFP-C

注 在使用自编程功能和数据闪存功能时，大约为 0.5KB（详细内容请参照“第 3 章”）。

1.2 产品型号一览表

图 1-1 产品型号、存储容量和封装

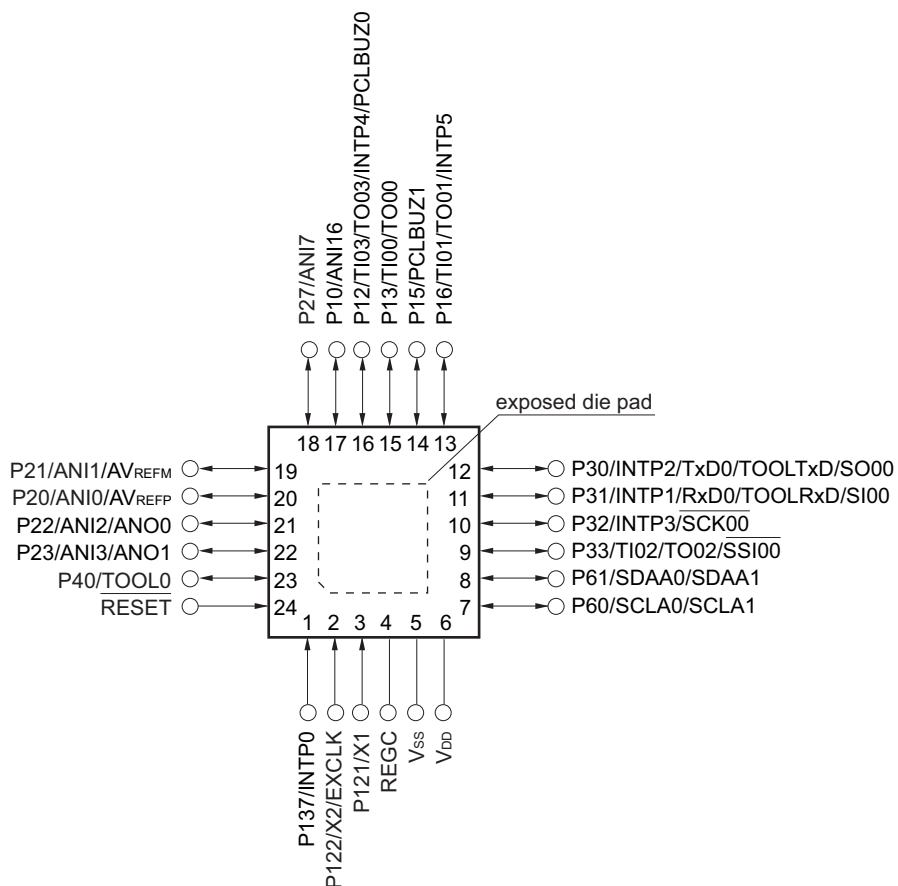


引脚数	封装	数据闪存	包装规格和环境对应	订购产品型号
24 引脚	24 引脚塑封 WQFN （4×4）	2KB	托盘、无铅 （纯 Sn）	R7F0C01072DNP#AA0
			托盘、无铅 （Ni/Pd/Au）	R7F0C01072DNP#AC0
32 引脚	32 引脚塑封 LQFP （7×7）		托盘、无铅 （纯 Sn）	R7F0C010B2DFP-C#AA0
			托盘、无铅 （Ni/Pd/Au）	R7F0C010B2DFP-C#AC0

### 1.3 引脚连接图 (Top View)

#### 1.3.1 24 引脚产品

- 24 引脚塑封 WQFN (4×4)

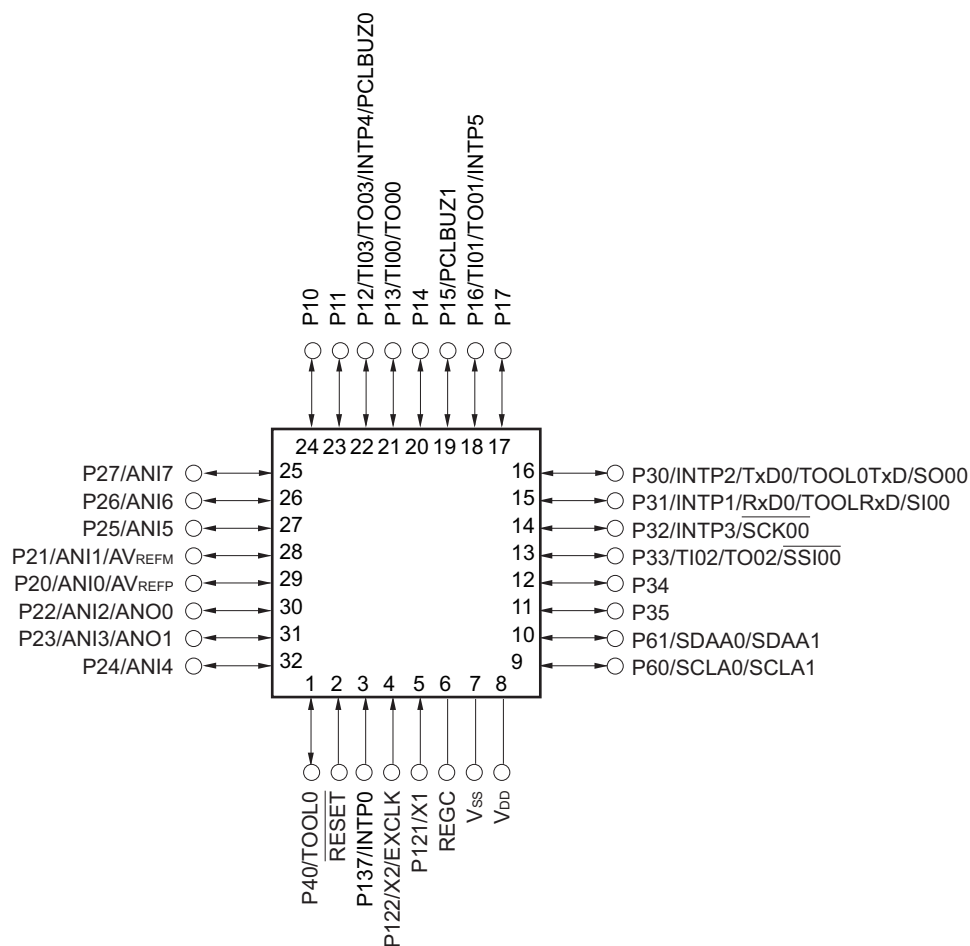


**注意** 必须通过电容器 (0.47 ~ 1μF) 将 REGC 引脚连接 V<sub>SS</sub> 引脚。

**备注** 有关引脚名, 请参照“1.4 引脚名”。

## 1.3.2 32 引脚产品

- 32 引脚塑封 LQFP (7×7)



**注意** 必须通过电容器 (0.47 ~ 1μF) 将 REGC 引脚连接 V<sub>SS</sub> 引脚。

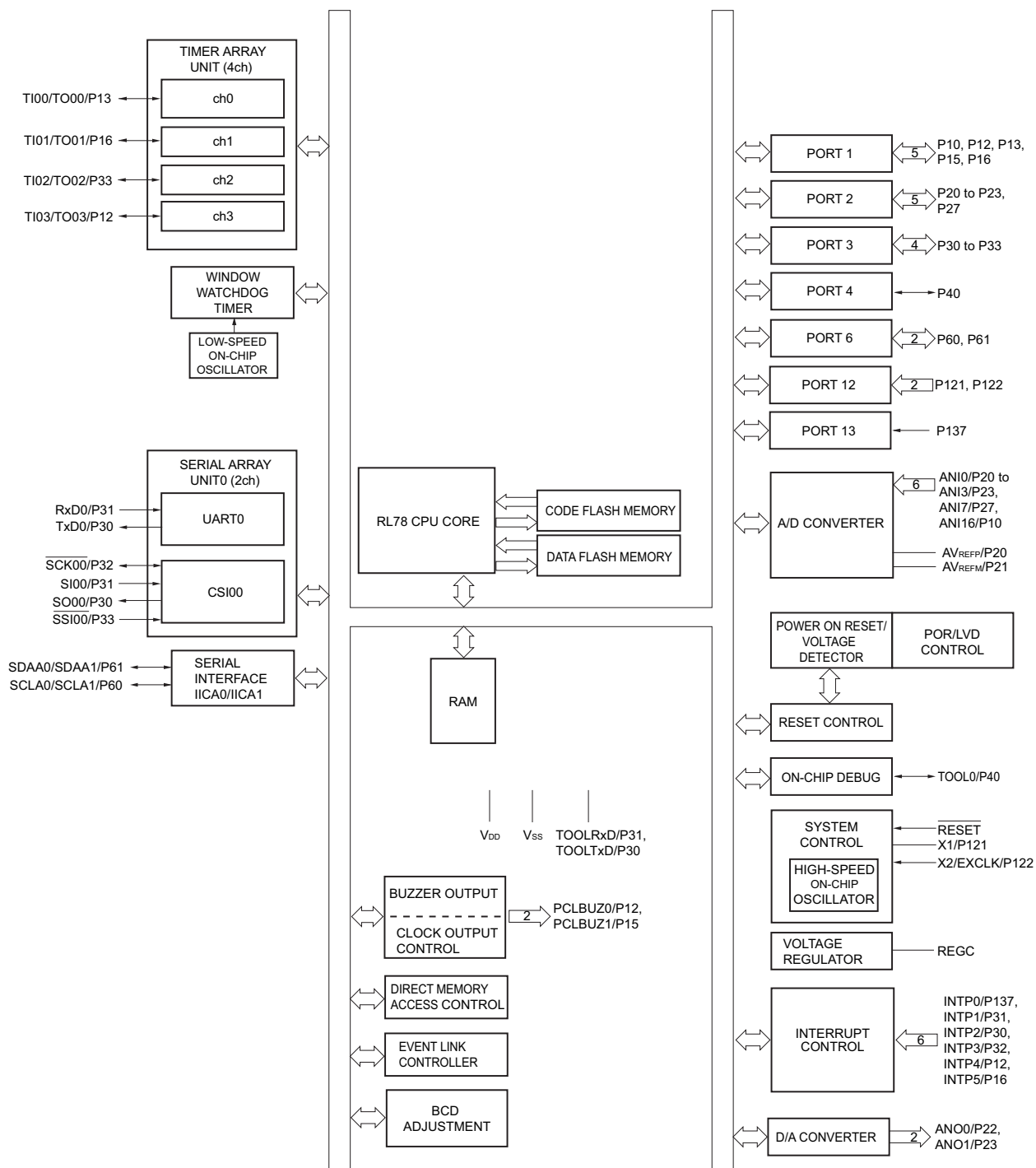
**备注** 有关引脚名, 请参照“1.4 引脚名”。

## 1.4 引脚名

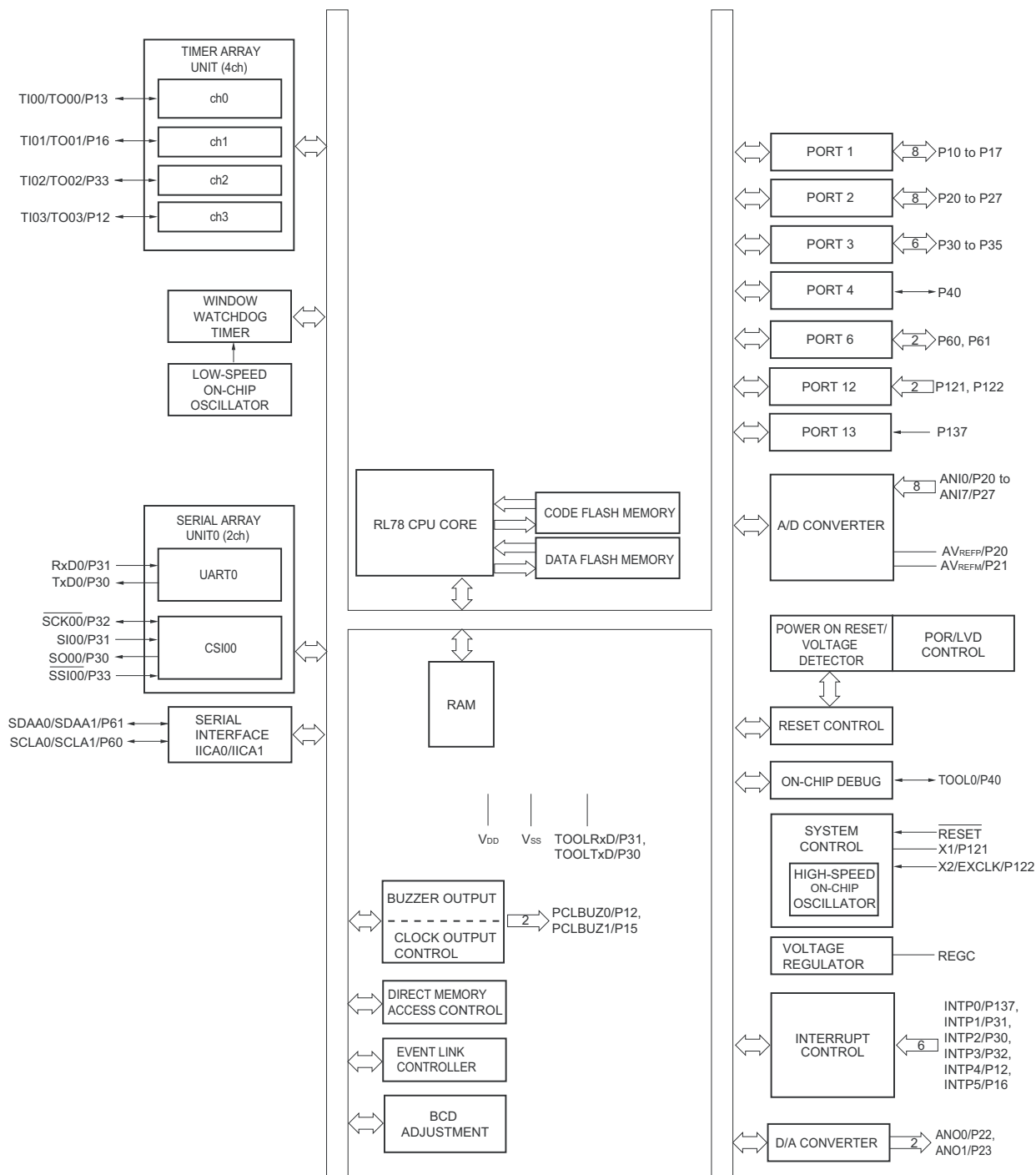
ANI0 ~ ANI7、ANI16	: Analog Input	RxD0	: Receive Data
ANO0、ANO1	: Analog Output	$\overline{\text{SCK00}}$	: Serial Clock Input/Output
AVREFM	: Analog Reference Voltage Minus	SCLA0、SCLA1	: Serial Clock Input/Output
AVREFP	: Analog Reference Voltage Plus	SDAA0、SDAA1	: Serial Data Input/Output
EXCLK	: External Clock Input (Main System Clock)	SI00	: Serial Data Input
		SO00	: Serial Data Output
		$\overline{\text{SSI00}}$	: Serial Interface Chip Select Input
		TI00 ~ TI03	: Timer Input
INTP0 ~ INTP5	: External Interrupt Input	TO00 ~ TO03	: Timer Output
P10 ~ P17	: Port 1	TOOL0	: Data Input/Output for Tool
P20 ~ P27	: Port 2	TOOLRxD、TOOLTxD	: Data Input/Output for External Device
P30 ~ P35	: Port 3	TxD0	: Transmit Data
P40	: Port 4	V <sub>DD</sub>	: Power Supply
P60、P61	: Port 6	V <sub>SS</sub>	: Ground
P121、P122	: Port 12	X1、X2	: Crystal Oscillator (Main System Clock)
P137	: Port 13		
PCLBUZ0、PCLBUZ1	: Programmable Clock Output/ Buzzer Output		
REGC	: Regulator Capacitance		
$\overline{\text{RESET}}$	: Reset		

## 1.5 框图

## 1.5.1 24 引脚产品



## 1.5.2 32 引脚产品





## 1.6 功能概要

(1/2)

项目		24 引脚	32 引脚
		R7F0C01072DNP	R7F0C010B2DFP-C
代码闪存		16KB	
数据闪存		2KB	
RAM		1.5KB 注	
存储空间		1M 字节	
主系统 时钟	高速系统时钟	X1（晶体 / 陶瓷）振荡、外部主系统时钟输入（EXCLK） 1 ~ 20MHz: V <sub>DD</sub> =2.7 ~ 3.6V	
	高速内部振荡器 时钟（f <sub>IH</sub> ）	高速运行: 32MHz（V <sub>DD</sub> =2.7 ~ 3.6V）	
低速内部振荡器时钟		15kHz(TYP.): V <sub>DD</sub> =2.7 ~ 3.6V	
通用寄存器		8 位 ×32 个寄存器（8 位 ×8 个寄存器 ×4 组）	
最短指令执行时间		0.03125μs（高速内部振荡器时钟: f <sub>IH</sub> =32MHz 运行时）	
		0.05μs（高速系统时钟: f <sub>MX</sub> =20MHz 运行时）	
指令集		• 数据传送（8/16 位） • 加减 / 逻辑运算（8/16 位） • 乘法运算（8 位 ×8 位） • 循环、桶式移位、位操作（置位、复位、测试和布尔运算）等	
I/O 端口	合计	20	28
	CMOS 输入 / 输出	15	23
	CMOS 输入	3	3
	N 沟道漏极开路 输入 / 输出 （6V 耐压）	2	2
定时器	16 位定时器	4 个通道（TAU）	
	看门狗定时器	1 个通道	
	定时器输出	4 个 PWM 输出: 3 个	
时钟输出 / 蜂鸣器输出		2 个	
		• 2.44kHz、4.88kHz、9.76kHz、1.25MHz、2.5MHz、5MHz、10MHz （主系统时钟: f <sub>MAIN</sub> =20MHz 运行时）	
8/12 位分辨率 A/D 转换器		6 个通道	8 个通道
D/A 转换器		2 个通道	
串行接口		• CSI: 1 个通道 /UART: 1 个通道	
	I <sup>2</sup> C 总线	1 个通道（2 个从属地址）	
DMA 控制器		2 个通道	
事件链接控制器（ELC）		事件输入: 10 个, 事件触发输出: 3 个	
向量	内部	12	
中断源	外部	6	

注 在使用自编程功能和数据闪存功能时, 大约为 0.5KB（详细内容请参照“第 3 章”）。

(2/2)

项目	24 引脚	32 引脚
	R7F0C01072DNP	R7F0C010B2DFP-C
复位	<ul style="list-style-type: none"> <li>• 通过 RESET 引脚进行的复位</li> <li>• 通过看门狗定时器进行的内部复位</li> <li>• 通过上电复位进行的内部复位</li> <li>• 通过电压检测电路进行的内部复位</li> <li>• 因执行非法指令而产生的内部复位<sup>注</sup></li> <li>• 因 RAM 奇偶校验错误而产生的内部复位</li> <li>• 因存取非法存储器而产生的内部复位</li> </ul>	
上电复位电路	<ul style="list-style-type: none"> <li>• 上电复位: 1.51±0.03V</li> <li>• 断电复位: 1.50±0.03V</li> </ul>	
电压检测电路	2.75V ~ 3.13V (4 种)	
片上调试功能	有	
电源电压	V <sub>DD</sub> =2.7 ~ 3.6V	
工作环境温度	T <sub>A</sub> =-40 ~ +85°C	

注 当执行指令码 FFH 时，发生非法指令错误。  
 在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

## 第 2 章 引脚功能

### 2.1 引脚功能一览表

全部产品引脚的输入 / 输出缓冲器电源为单电源。

各端口设定的输入 / 输出、缓冲器和上拉电阻也对复用功能有效。

表 2-1 各引脚的输入 / 输出缓冲器电源

电源	对应的引脚
V <sub>DD</sub>	全部引脚

#### 2.1.1 24 引脚产品

功能名称	输入 / 输出	功能	复位后	复用功能
P10	输入 / 输出	端口 1 5 位输入 / 输出端口 能以位为单位指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	ANI16
P12				TI03/TO03/INTP4/ PCLBUZ0
P13				TI00/TO00
P15				PCLBUZ1
P16				TI01/TO01/INTP5
P20	输入 / 输出	端口 2 5 位输入 / 输出端口 能以位为单位指定输入或者输出。	模拟输入 端口	ANI0/AV <sub>REFP</sub>
P21				ANI1/AV <sub>REFM</sub>
P22				ANI2/ANO0
P23				ANI3/ANO1
P27				ANI7
P30	输入 / 输出	端口 3 4 位输入 / 输出端口 能以位为单位指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	INTP2/TxD0/ TOOLTxD/SO00
P31				INTP1/RxD0/ TOOLRxD/SI00
P32				INTP3/ $\overline{\text{SCK00}}$
P33				TI02/TO02/SSI00
P40	输入 / 输出	端口 4 1 位输入 / 输出端口 能指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	TOOL0
P60	输入 / 输出	端口 6 2 位输入 / 输出端口 P60 和 P61 的输出为 N 沟道漏极开路输出 (6V 耐压)。 能以位为单位指定输入或者输出。	输入端口	SCLA0/SCLA1
P61				SDAA0/SDAA1
P121	输入	端口 12 2 位输入端口	输入端口	X1
P122				X2/EXCLK
P137	输入	端口 13 1 位输入专用端口	输入端口	INTP0

## 2.1.2 32 引脚产品

功能名称	输入 / 输出	功能	复位后	复用功能
P10	输入 / 输出	端口 1 8 位输入 / 输出端口 能以位为单位指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	—
P11				—
P12				TI03/TO03/INTP4/ PCLBUZ0
P13				TI00/TO00
P14				—
P15				PCLBUZ1
P16				TI01/TO01/INTP5
P17				—
P20	输入 / 输出	端口 2 8 位输入 / 输出端口 能以位为单位指定输入或者输出。	模拟输入 端口	ANI0/AV <sub>REFP</sub>
P21				ANI1/AV <sub>REFM</sub>
P22				ANI2/ANO0
P23				ANI3/ANO1
P24				ANI4
P25				ANI5
P26				ANI6
P27				ANI7
P30	输入 / 输出	端口 3 6 位输入 / 输出端口 能以位为单位指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	INTP2/TxD0/ TOOLTxD/SO00
P31				INTP1/RxD0/ TOOLRxD/SI00
P32				INTP3/ $\overline{\text{SCK00}}$
P33				TI02/TO02/SSI00
P34				—
P35				—
P40	输入 / 输出	端口 4 1 位输入 / 输出端口 能指定输入或者输出。 能通过软件的设定, 使用内部上拉电阻。	输入端口	TOOL0
P60	输入 / 输出	端口 6 2 位输入 / 输出端口 P60 和 P61 的输出为 N 沟道漏极开路输出 (6V 耐压)。 能以位为单位指定输入或者输出。	输入端口	SCLA0/SCLA1
P61				SDAA0/SDAA1
P121	输入	端口 12 2 位输入端口	输入端口	X1
P122				X2/EXCLK
P137	输入	端口 13 1 位输入专用端口	输入端口	INTP0

## 2.2 端口以外的引脚

### 2.2.1 各产品配置的功能

功能名称	32-pin	24-pin
ANI0	○	○
ANI1	○	○
ANI2	○	○
ANI3	○	○
ANI4	○	—
ANI5	○	—
ANI6	○	—
ANI7	○	○
ANI16	—	○
ANO0	○	○
ANO1	○	○
INTP0	○	○
INTP1	○	○
INTP2	○	○
INTP3	○	○
INTP4	○	○
INTP5	○	○
PCLBUZ0	○	○
PCLBUZ1	○	○
REGC	○	○
RESET	○	○
RxD0	○	○
SCK00	○	○
SCLA0	○	○
SCLA1	○	○
SDAA0	○	○
SDAA1	○	○
SI00	○	○
SO00	○	○
SSI00	○	○
TI00	○	○
TI01	○	○
TI02	○	○
TI03	○	○
TO00	○	○
TO01	○	○
TO02	○	○
TO03	○	○
TxD0	○	○

功能名称	32-pin	24-pin
X1	○	○
X2	○	○
EXCLK	○	○
V <sub>DD</sub>	○	○
V <sub>SS</sub>	○	○
TOOLRxD	○	○
TOOLTxD	○	○
TOOL0	○	○

## 2.2.2 功能说明

(1/2)

功能名称	输入 / 输出	功能
ANI0	输入	A/D 转换器的模拟输入
ANI1		
ANI2		
ANI3		
ANI4		
ANI5		
ANI6		
ANI7		
ANI16		
ANO0	输出	D/A 转换器输出
ANO1		
INTP0	输入	外部中断输入
INTP1		
INTP2		
INTP3		
INTP4		
INTP5		
PCLBUZ0	输出	时钟输出 / 蜂鸣器输出
PCLBUZ1		
REGC	—	连接用于内部工作的稳压器输出稳定电容器。 必须通过电容器 (0.47 ~ 1μF) 连接 V <sub>SS</sub> 。
RESET	输入	外部复位输入
RxD0	输入	UART0 的串行数据输入
SCK00	输入 / 输出	CSI00 的时钟输入 / 输出
SCLA0	输入 / 输出	I <sup>2</sup> C 的时钟输入 / 输出
SCLA1		
SDAA0	输入 / 输出	I <sup>2</sup> C 的串行数据输入 / 输出
SDAA1		
SI00	输入	CSI00 的串行数据输入
SO00	输出	CSI00 的串行数据输出
SSI00	输入	CSI00 的芯片选择输入
TI00	输入	16 位定时器 00 的外部计数时钟输入
TI01		16 位定时器 01 的外部计数时钟输入
TI02		16 位定时器 02 的外部计数时钟输入
TI03		16 位定时器 03 的外部计数时钟输入
TO00	输出	16 位定时器 00 的输出
TO01		16 位定时器 01 的输出
TO02		16 位定时器 02 的输出
TO03		16 位定时器 03 的输出
TxD0	输出	UART0 的串行数据输出

(2/2)

功能名称	输入 / 输出	功能
X1	—	连接用于主系统时钟的谐振器。
X2	—	
EXCLK	输入	主系统时钟的外部时钟输入
V <sub>DD</sub>	—	全部引脚的正电源
V <sub>SS</sub>	—	全部引脚的接地电位
TOOLRxD	输入	用于在闪存编程时连接外部设备的 UART 接收引脚
TOOLTxD	输出	用于在闪存编程时连接外部设备的 UART 发送引脚
TOOL0	输入 / 输出	用于闪存编程器 / 调试器的数据输入 / 输出

注意 解除复位时的 P40/TOOL0 和运行模式的关系如下所示。

表 2-2 解除复位时的 P40/TOOL0 和运行模式的关系

P40/TOOL0	运行模式
V <sub>DD</sub>	通常运行模式
0V	闪存编程模式

详细内容请参照“23.5 编程方法”。

备注 作为噪声和锁定的对策，必须在 V<sub>DD</sub>-V<sub>SS</sub> 之间以最短的距离并且用较粗的布线连接旁路电容器（0.1μF 左右）。

## 2.3 引脚功能的说明

备注 配置的引脚因产品而不同。请参照“1.3 引脚连接图（Top View）”和“2.1 引脚功能一览表”。

### 2.3.1 P10 ~ P17（Port 1）

这是输入 / 输出端口。除了输入 / 输出端口以外，还有定时器的输入 / 输出、外部中断的请求输入和时钟 / 蜂鸣器输出功能。

能通过设定上拉电阻选择寄存器 1（PU1），使用内部上拉电阻。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。能通过设定端口模式寄存器 1（PM1），以位为单位指定为输入端口或者输出端口。

#### (2) 控制模式

用作定时器的输入 / 输出、外部中断的请求输入和时钟 / 蜂鸣器的输出。

##### (a) TI00、TI01、TI03

这些是 16 位定时器 00、01、03 的外部计数时钟 / 捕捉触发的输入引脚。

##### (b) TO00、TO01、TO03

这些是 16 位定时器 00、01、03 的定时器输出引脚。

##### (c) INTP4、INTP5

这些是能指定有效边沿（上升沿、下降沿、上升和下降的双边沿）的外部中断的请求输入引脚。

##### (d) PCLBUZ0、PCLBUZ1

这些是时钟 / 蜂鸣器的输出引脚。

##### (e) ANI16（只限于 24 引脚产品）

用作 A/D 转换器的模拟输入引脚（ANI16）。请参照“9.10(5) 模拟输入（ANIn）引脚”。



### 2.3.2 P20 ~ P27 (Port 2)

这是输入 / 输出端口。除了输入 / 输出端口以外，还有 A/D 转换器的模拟输入、A/D 转换器的基准电位输入和 D/A 转换器输出功能。

通过 A/D 端口配置寄存器 (ADPC) 将各引脚设定为数字或者模拟。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。能通过设定端口模式寄存器 2 (PM2)，以位为单位指定为输入端口或者输出端口。

#### (2) 控制模式

用作 A/D 转换器的模拟输入、A/D 转换器的基准电位输入和 D/A 转换器的输出。

##### (a) ANI0 ~ ANI7

用作 A/D 转换器的模拟输入引脚 (ANI0 ~ ANI7)。请参照“9.10(5) 模拟输入 (ANIn) 引脚”。

##### (b) $AV_{REFP}$

这是 A/D 转换器的基准电位 (+) 的输入引脚。

##### (c) $AV_{REFM}$

这是 A/D 转换器的基准电位 (-) 的输入引脚。

##### (d) ANO0、ANO1

这些是 D/A 转换器的输出引脚。

### 2.3.3 P30 ~ P35 (Port 3)

这是输入 / 输出端口。除了输入 / 输出端口以外，还有外部中断的请求输入、串行接口的数据输入 / 输出、时钟的输入 / 输出、芯片的选择输入、编程 UART 的输入 / 输出和定时器的输入 / 输出功能。

能通过设定上拉电阻选择寄存器 3 (PU3)，使用内部上拉电阻。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。能通过设定端口模式寄存器 3 (PM3)，以位为单位指定为输入端口或者输出端口。

#### (2) 控制模式

用作外部中断的请求输入、串行接口的数据输入 / 输出、时钟的输入 / 输出、芯片的选择输入、编程 UART 的输入 / 输出和定时器的输入 / 输出。

##### (a) INTP1、INTP2、INTP3

这些是能指定有效边沿（上升沿、下降沿、上升和下降的双边沿）的外部中断的请求输入引脚。

##### (b) TxD0

这是串行接口 UART0 的串行数据输出引脚。

##### (c) RxD0

这是串行接口 UART0 的串行数据输入引脚。

##### (d) SI00

这是串行接口 CSI00 的串行数据输入引脚。

##### (e) SO00

这是串行接口 CSI00 的串行数据输出引脚。

##### (f) $\overline{\text{SCK00}}$

这是串行接口 CSI00 的串行时钟输入 / 输出引脚。

##### (g) $\overline{\text{SSI00}}$

这是串行接口 CSI00 的芯片选择输入引脚。

##### (h) TI02

这是 16 位定时器 02 的外部计数时钟 / 捕捉触发的输入引脚。

##### (i) TO02

这是 16 位定时器 02 的定时器输出引脚。

##### (j) TOOLTxD

这是用于在闪存编程时连接外部设备的 UART 串行数据输出引脚。

##### (k) TOOLRxD

这是用于在闪存编程时连接外部设备的 UART 串行数据输入引脚。

### 2.3.4 P40 (Port 4)

这是输入 / 输出端口。除了输入 / 输出端口以外，还有用于闪存编程器 / 调试器的数据输入 / 输出功能。

能通过设定上拉电阻选择寄存器 4 (PU4)，使用内部上拉电阻。

当 P40 为允许片上调试（通过选项字节进行设定）时，必须外接上拉电阻。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。能通过设定端口模式寄存器 4 (PM4)，以位为单位指定为输入端口或者输出端口。

#### (2) 控制模式

用作闪存编程器 / 调试器的数据输入 / 输出。

#### (a) TOOL0

这是用于闪存编程器 / 调试器的数据输入 / 输出引脚。

当允许片上调试时，必须在外部进行上拉（禁止下拉）。

**注意** 解除复位时的 P40/TOOL0 和运行模式的关系如下所示。

表 2-3 解除复位时的 P40/TOOL0 和运行模式的关系

P40/TOOL0	运行模式
V <sub>DD</sub>	通常运行模式
0V	闪存编程模式

详细内容请参照“23.5 编程方法”。

### 2.3.5 P60、P61 (Port 6)

这是输入 / 输出端口。除了输入 / 输出端口以外，还有串行接口的数据输入 / 输出和时钟输入 / 输出功能。

P60 引脚和 P61 引脚的输出为 N 沟道漏极开路输出（6V 耐压）。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。能通过设定端口模式寄存器 6 (PM6)，以位为单位指定为输入端口或者输出端口。

#### (2) 控制模式

用作串行接口的数据输入 / 输出、时钟输入 / 输出、芯片选择输入和定时器的输入 / 输出。

#### (a) SCLA0、SCLA1

这些是串行接口 IICA0、IICA1 的串行时钟输入 / 输出引脚。

#### (b) SDAA0、SDAA1

这些是串行接口 IICA0、IICA1 的串行数据输入 / 输出引脚。

### 2.3.6 P121、P122 (Port 12)

这是输入 / 输出端口。除了输入 / 输出端口以外，还有主系统时钟的谐振器连接和主系统时钟的外部时钟输入功能。

能以位为单位指定以下模式。

#### (1) 端口模式

用作输入 / 输出端口。

#### (2) 控制模式

用作主系统时钟的谐振器连接和主系统时钟的外部时钟输入。

##### (a) X1、X2

这些是主系统时钟的谐振器连接引脚。

##### (b) EXCLK

这是主系统时钟的外部时钟输入引脚。

### 2.3.7 P137 (Port 13)

P137 是输入端口。除了输入端口以外，还有外部中断的请求输入功能。

#### (1) 端口模式

P137 用作输入端口。

#### (2) 控制模式

用作外部中断的请求输入引脚。

##### (a) INTP0

这是能指定有效边沿（上升沿、下降沿、上升和下降的双边沿）的外部中断的请求输入引脚。

### 2.3.8 $V_{DD}$ 、 $V_{SS}$

#### (1) $V_{DD}$

这是提供正电源的引脚。

#### (2) $V_{SS}$

这是接地电位引脚。

**备注** 作为噪声和锁定的对策，必须在  $V_{DD}$ - $V_{SS}$  之间以最短的距离并且用较粗的布线连接旁路电容器（0.1 $\mu$ F 左右）。

### 2.3.9 $\overline{\text{RESET}}$

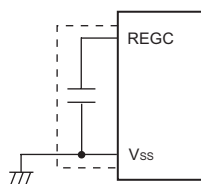
这是低电平有效的系统复位输入引脚。

当不使用外部复位时，必须直接或者通过电阻连接  $V_{DD}$ 。

当使用外部复位时，必须以  $V_{DD}$  为基准进行设计。

### 2.3.10 REGC

这是用于内部工作的稳压器输出稳定电容器的连接引脚。必须通过电容器（0.47 ~ 1 $\mu$ F）连接  $V_{SS}$ 。为了稳定内部电压，必须使用特性好的电容器。



**注意** 必须尽量缩短上图虚线部分的布线。

## 2.4 未使用引脚的处理

未使用引脚的处理如表 2-4 所示。

表 2-4 各未使用引脚的处理

引脚名	输入 / 输出电路类型	输入 / 输出	未使用时的推荐连接方法
P10/ANI16 注 1	11-U	输入 / 输出	输入：必须单独通过电阻连接 V <sub>DD</sub> 或者 V <sub>SS</sub> 。 输出：必须置为开路。
P11 注 2	8-R		
P12/TI03/TO03/INTP4/ PCLBUZ0			
P13/TI00/TO00			
P14 注 2			
P15/PCLBUZ1			
P16/TI01/TO01/INTP5			
P17 注 2			
P20/ANI0/AV <sub>REFP</sub>			
P21/ANI1/AV <sub>REFM</sub>			
P22/ANI2/ANO0	44		
P23/ANI3/ANO1			
P24/ANI4 注 2	11-G		
P25/ANI5 注 2			
P26/ANI6 注 2			
P27/ANI7			
P30/INTP2/TxD0 /TOOLTxD/SO00	8-R		
P31/INTP1/RxD0/ TOOLRxD/SI00			
P32/INTP3/SCK00			
P33/TI02/TO02/SSI00			
P34 注 2			
P35 注 2			
P40/TOOL0		输入 / 输出	
P60/SCLA0/SCLA1			13-R
P61/SDAA0/SDAA1			
P121/X1	37-C	输入	必须单独通过电阻连接 V <sub>DD</sub> 或者 V <sub>SS</sub> 。
P122/X2/EXCLK			
P137/INTP0	2	输入	必须单独通过电阻连接 V <sub>DD</sub> 或者 V <sub>SS</sub> 。
RESET	2	输入	必须直接或者通过电阻连接 V <sub>DD</sub> 。
REGC	—	—	必须通过电容器（0.47 ~ 1μF）连接 V <sub>SS</sub> 。

注 1. ANI16 只限于 24 引脚产品。

2. 只限于 32 引脚产品。

图 2-1 引脚的输入 / 输出电路一览表 (1/2)

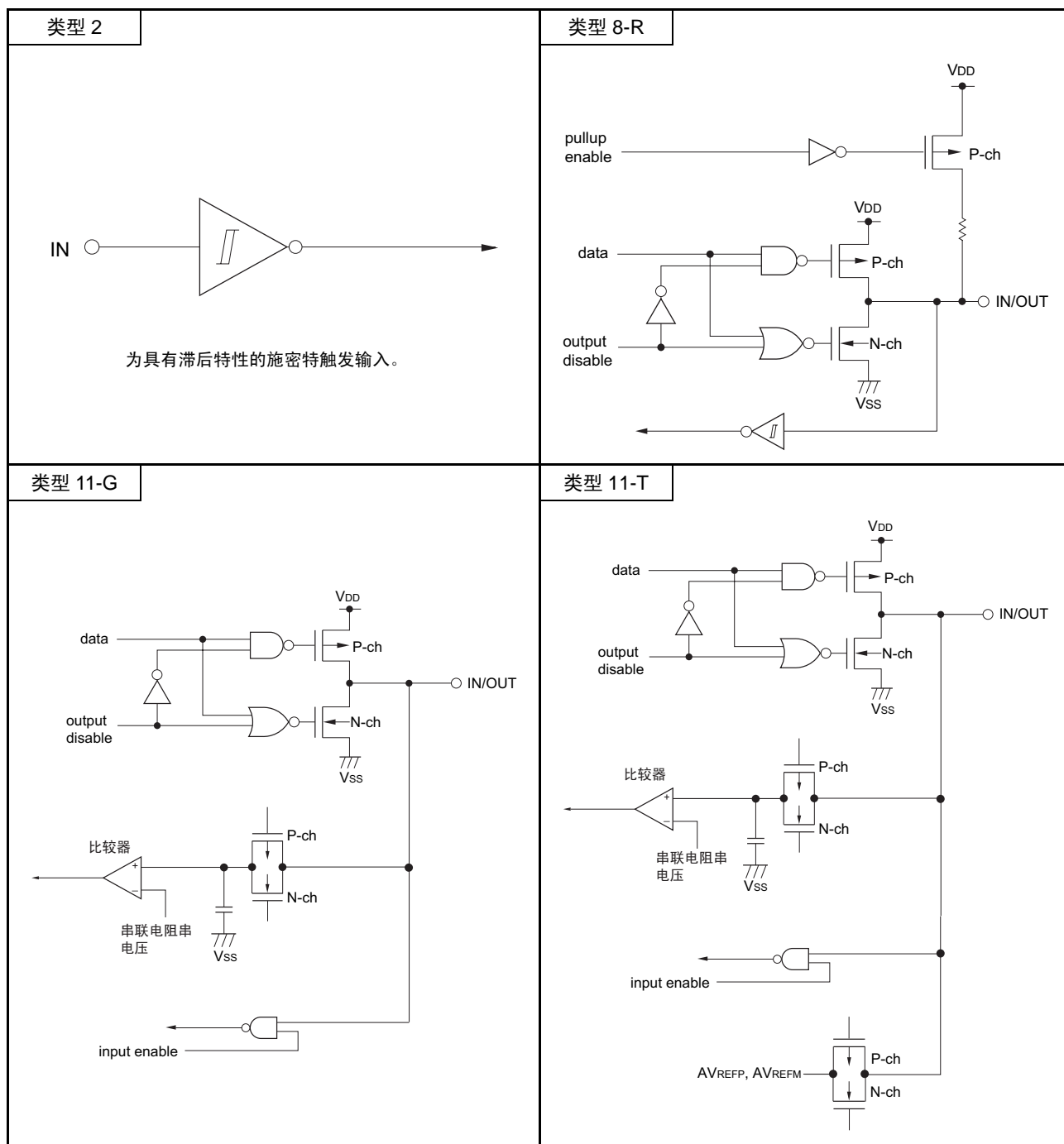
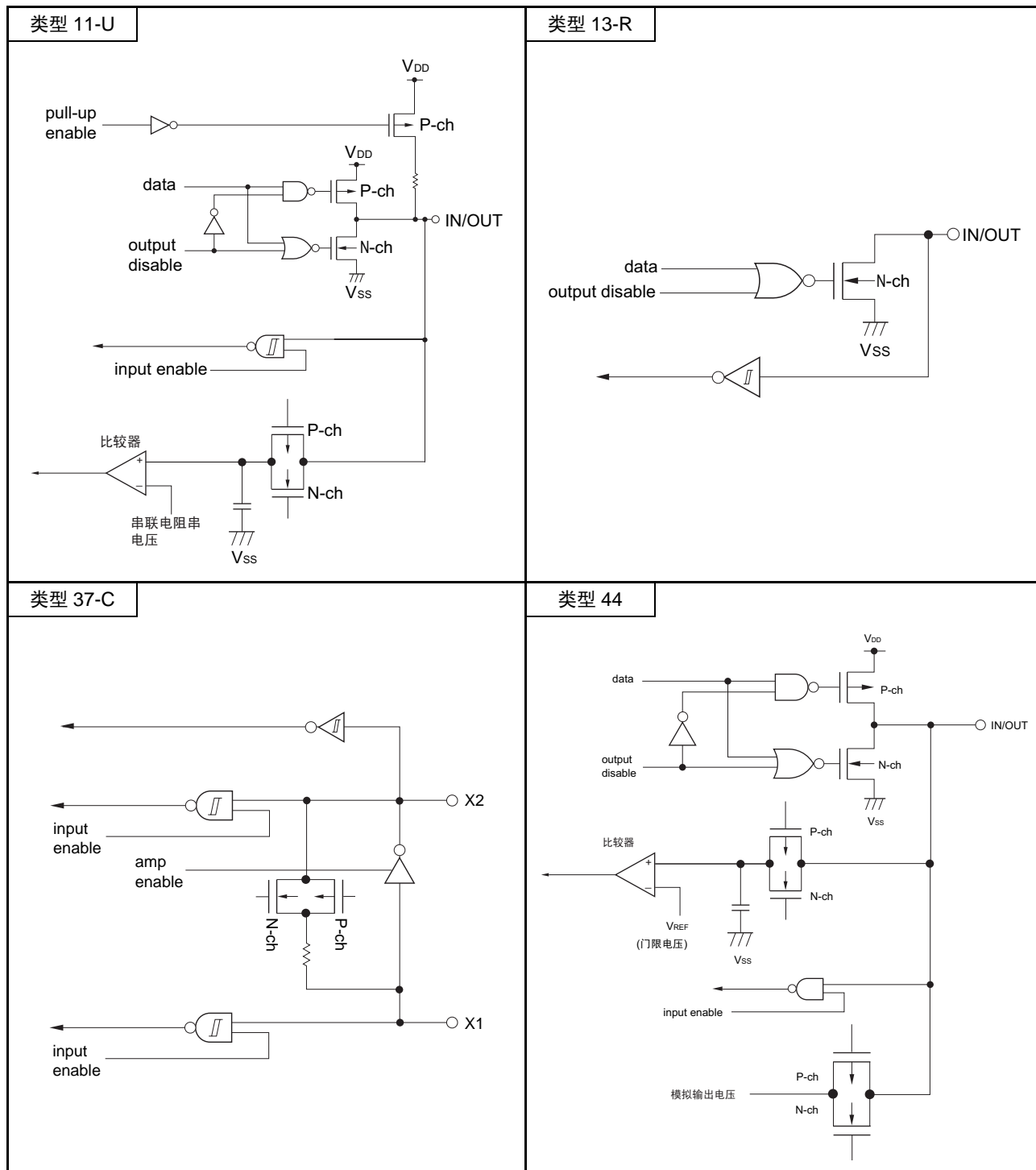


图 2-1 引脚的输入 / 输出电路一览表 (2/2)



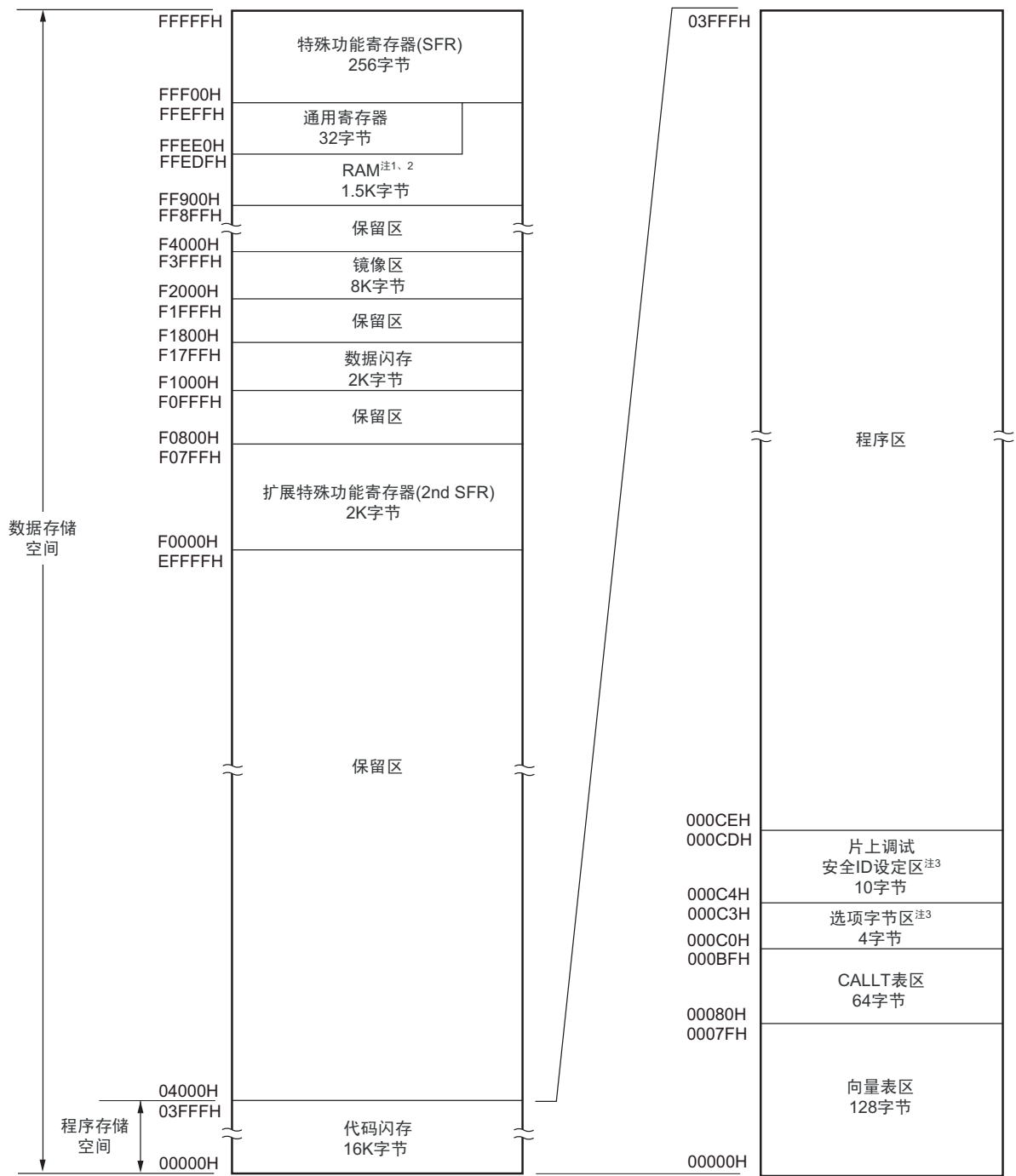


## 第 3 章 CPU 体系结构

### 3.1 存储空间

R7F0C010 能存取 1M 字节的地址空间。存储器映像如图 3-1 所示。

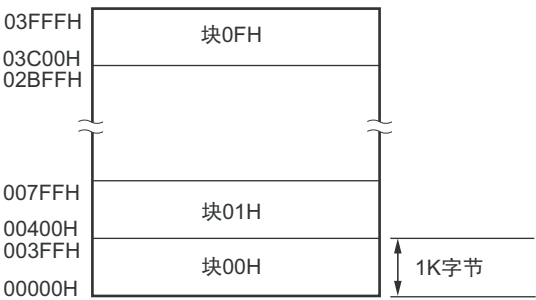
图 3-1 存储器映像



- 注 1. 在自编程时，不能将堆栈数据缓冲器和向量中断处理的转移目标分配到 FFE20H ~ FFEFFH 和 FF900H ~ FFC80H 的区域。
2. 能从除了通用寄存器以外的 RAM 区执行指令。
3. 给 000C0H ~ 000C3H 设定选项字节，并且给 000C4H ~ 000CDH 设定片上调试安全 ID。

注意 在允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区+10 字节”的区域进行初始化。通过产生复位，进入允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的状态。详细内容请参照“20.5 RAM 奇偶校验错误检测功能”。

备注 闪存分为多个块（1 块 =1K 字节）。有关地址值和块号，请参照“表 3-1 闪存的地址值和块号的对应”。



闪存地址值和块号的对应如下所示。

表 3-1 闪存地址值和块号的对应

地址值	块号
00000H ~ 003FFH	00H
00400H ~ 007FFH	01H
00800H ~ 00BFFH	02H
00C00H ~ 00FFFH	03H
01000H ~ 013FFH	04H
01400H ~ 017FFH	05H
01800H ~ 01BFFH	06H
01C00H ~ 01FFFH	07H
02000H ~ 023FFH	08H
02400H ~ 027FFH	09H
02800H ~ 02BFFH	0AH
02C00H ~ 02FFFH	0BH
03000H ~ 033FFH	0CH
03400H ~ 037FFH	0DH
03800H ~ 03BFFH	0EH
03C00H ~ 03FFFH	0FH

### 3.1.1 内部程序存储空间

内部程序存储空间保存程序和表数据，R7F0C010 内置的 ROM（闪存）如下所示。

表 3-2 内部 ROM 容量

产品	内部 ROM	
	构造	容量
R7F0C010	闪存	16384×8 位（00000H ~ 03FFFH）

内部程序的存储空间分为以下区域。

#### (1) 向量表区

将 00000H ~ 0007FH 的 128 字节区域保留为向量表区，向量表区保存在复位或者产生各中断请求时需要转移的程序起始地址。另外，因为向量码为 2 字节，所以中断的转移目标地址为 00000H ~ 0FFFFH 的 64K 地址。

偶数地址保存 16 位地址中的低 8 位，奇数地址保存 16 位地址中的高 8 位。

表 3-3 向量表

向量表地址	中断源
0000H	RESET、POR、LVD、WDT、TRAP、IAW、RPE
0004H	INTWDTI
0006H	INTLVI
0008H	INTP0
000AH	INTP1
000CH	INTP2
000EH	INTP3
0010H	INTP4
0012H	INTP5
0014H	INTAD
0016H	INTIICA0
0018H	INTFL
001AH	INTDMA0
001CH	INTDMA1
001EH	INTST0/INTCSI00
0020H	INTSR0
0022H	INTSRE0
	INTTM01H
0028H	INTTM03H
002AH	INTIICA1
002CH	INTTM00
002EH	INTTM01
0030H	INTTM02
0032H	INTTM03

## (2) CALLT 指令表区

00080H ~ 000BFH 的 64 字节区域能保存 2 字节调用指令（CALLT）的子程序入口地址。必须给予程序入口地址设定 00000H ~ 0FFFFH 内的值（因为地址码为 2 字节）。

## (3) 选项字节区

000C0H ~ 000C3H 的 4 字节区域用作选项字节区。详细内容请参照“第 22 章 选项字节”。

## (4) 片上调试安全 ID 设定区

000C4H ~ 000CDH 的 10 字节区域用作片上调试安全 ID 设定区。详细内容请参照“第 24 章 片上调试功能”。

3.1.2 镜像区

R7F0C010 将 02000H ~ 03FFFH 的代码闪存区镜像到 F2000H ~ F3FFFH。

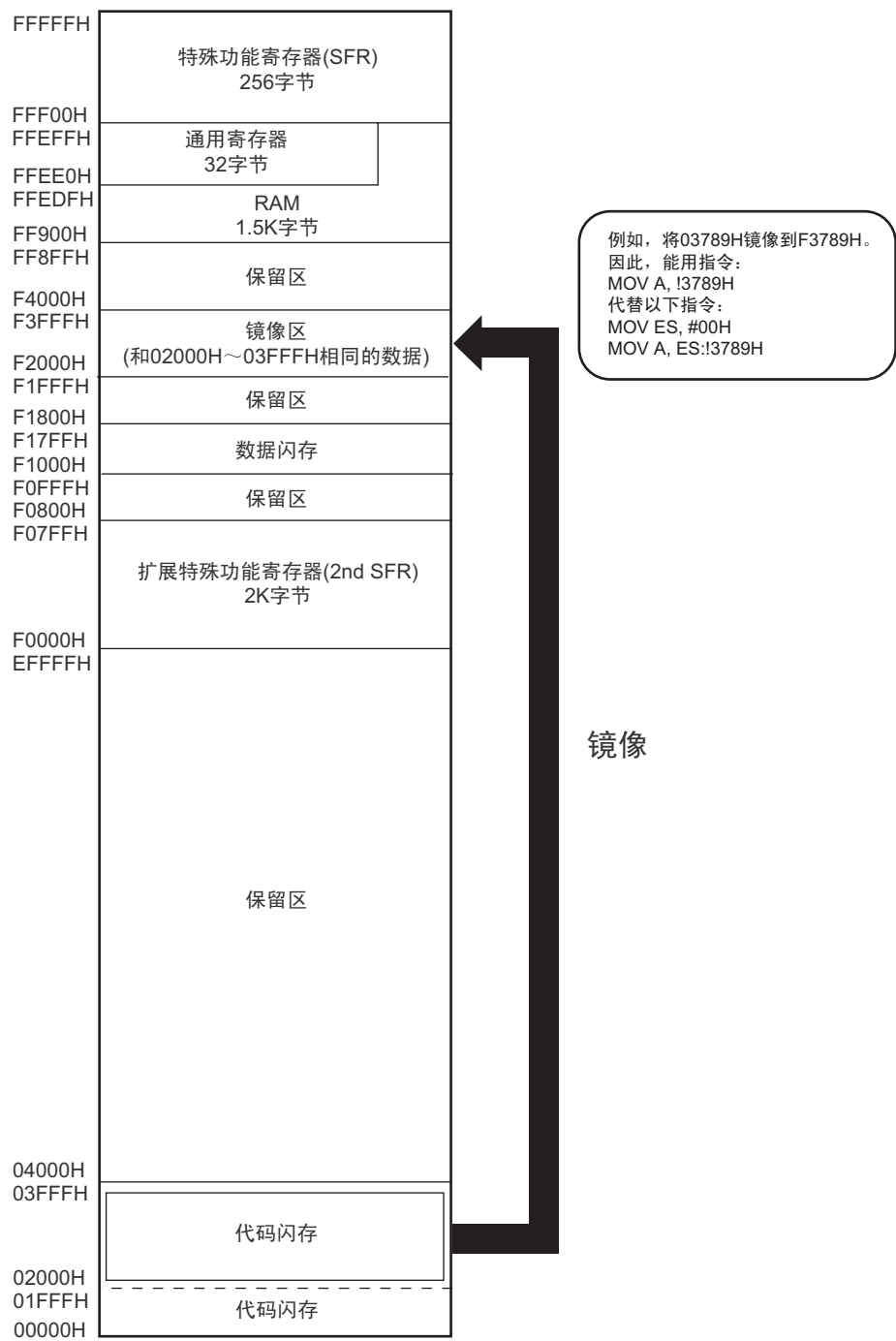
能通过从镜像目标的 F2000H ~ F3FFFH 读数据来使用操作数中不持有 ES 寄存器的指令，因此能用短代码读代码闪存的内容。但是，不能将代码闪存区镜像到 SFR、扩展 SFR、RAM 区以及保留区。

有关各产品的镜像区，请参照“3.1 存储空间”。

镜像区为只读区，不能从此区域取指令。

例子如下所示。

例 R7F0C010（闪存为 16K 字节，RAM 为 1.5K 字节）



- **处理器模式控制寄存器（PMC）**  
这是设定要镜像到F2000H～F3FFFFH的闪存空间的寄存器。  
通过1位或者8位存储器操作指令设定PMC寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 3-2 处理器模式控制寄存器 (PMC) 的格式

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	要镜像到 F2000H ~ F3FFFH 的闪存空间的设定
0	将 02000H ~ 03FFFH 镜像到 F2000H ~ F3FFFH。
1	禁止设定。

注意 1. bit0 (MAA) 必须为“0” (初始值)。

2. 在设定 PMC 寄存器后，必须至少在等待 1 条指令后存取镜像区。



### 3.1.3 内部数据存储空间

R7F0C010 内置以下 RAM。

表 3-4 内部 RAM 容量

产品	内部 RAM
R7F0C010	1536×8 位 (FF900H ~ FFEFFH)

内部 RAM 除了能用作数据区以外，还能作为程序区写指令并且执行指令（不能在分配通用寄存器的区域执行指令）。给内部 RAM 区 FFEE0H ~ FFEFFH 的 32 字节区域分配了以 8 个 8 位寄存器为 1 组的 4 组通用寄存器。

另外，堆栈存储器使用内部 RAM。

注意 1. 不能将分配通用寄存器（FFEE0H ~ FFEFFH）的空间用于取指令和堆栈区。

2. 在自编程或者改写数据闪存时，不能给以下产品的 RAM 区分配各库使用的堆栈和数据缓冲器以及用于 DMA 传送的 RAM 地址。有关详细内容，请确认《RL78 Family Flash Self Programming Library Type01 User's Manual》和《RL78 Family Data Flash Library Type04 User's Manual》。

R7F0C010 : FFE20H ~ FFEFFH

3. 因为以下产品的 RAM 区用于自编程库和数据闪存库，所以禁止使用（参照“图 3-1 存储器映像”）：

R7F0C010 : FF900H ~ FFC80H

### 3.1.4 特殊功能寄存器（SFR: Special Function Register）的区域

内部外围硬件的特殊功能寄存器（SFR）分配在 FFF00H ~ FFFFFH 的区域（参照“3.2.4 特殊功能寄存器（SFR: Special Function Register）”的表 3-5）。

注意 不能存取未分配 SFR 的地址。

### 3.1.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）的区域

内部外围硬件的扩展特殊功能寄存器（2nd SFR）分配在 F0000H ~ F07FFH 的区域（参照“3.2.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）”的表 3-6）。

在此区域中分配了 SFR 区（FFF00H ~ FFFFFH）以外的 SFR，但是扩展 SFR 区的存取指令比 SFR 区长 1 字节。

注意 不能存取未分配扩展 SFR 的地址。

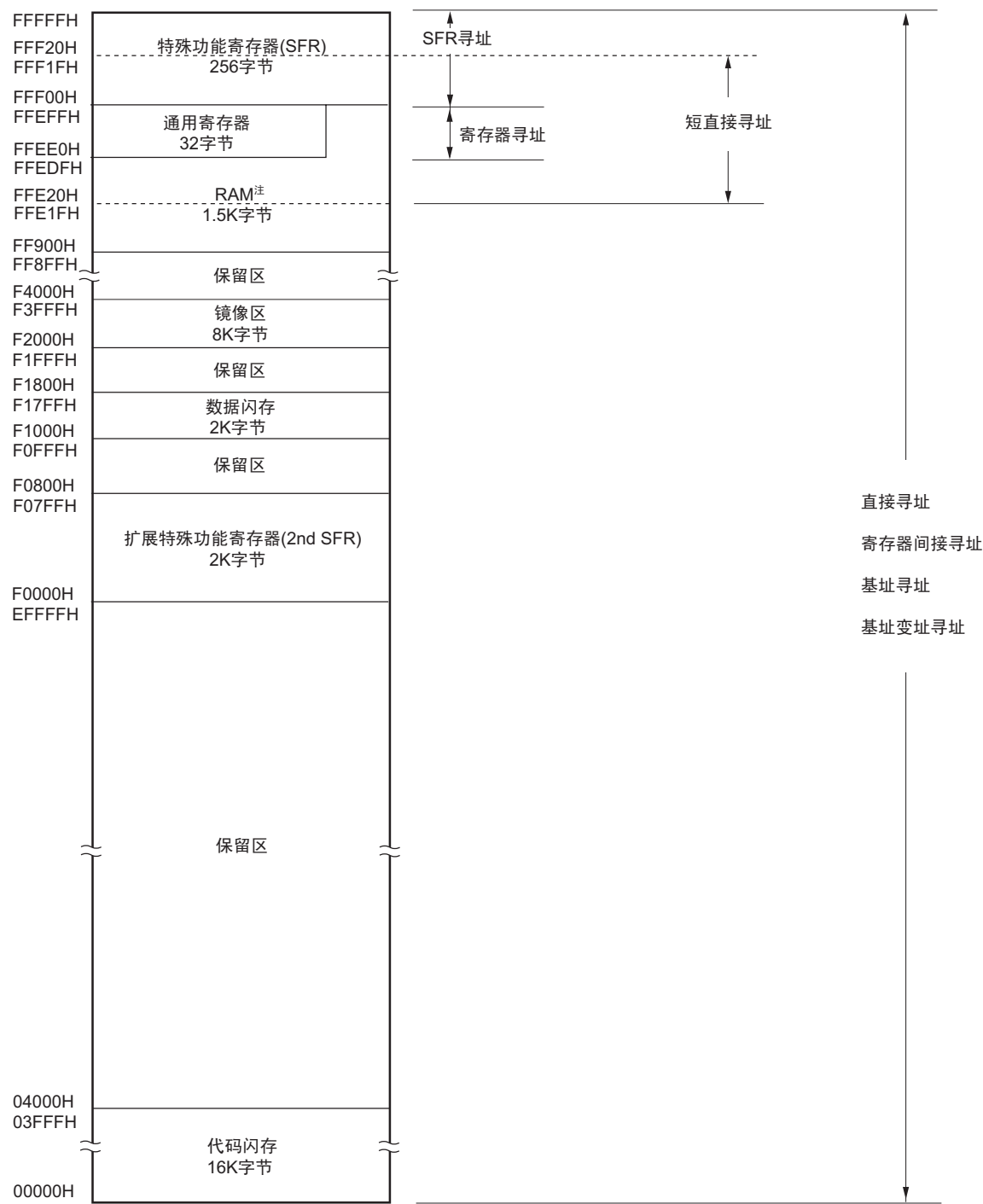
### 3.1.6 数据存储器的寻址

所谓寻址，是指定下一次要执行的指令地址以及指令执行操作对象的寄存器或者存储器等地址的方法。

对于指令执行操作对象的存储器的寻址，考虑到可操作性等，R7F0C010 提供了丰富的寻址方式。尤其是能根据特殊功能寄存器（SFR）和通用寄存器等各种功能进行特殊的寻址。数据存储器和寻址的对应如图 3-3 所示。

有关各寻址的详细内容，请参照“3.4 处理数据地址的寻址”。

图 3-3 数据存储器和寻址的对应



注 在使用自编程功能和数据闪存功能时，此区域用于自编程库，因此禁止使用 FFE20H ~ FFEFFH 和 FF900H ~ FFC80H 的区域。

## 3.2 处理器的寄存器

R7F0C010 内置以下处理器的寄存器。

### 3.2.1 控制寄存器

这是具有控制程序顺序、状态和堆栈存储器等专用功能的寄存器。在控制寄存器中有程序计数器（PC）、程序状态字（PSW）和堆栈指针（SP）。

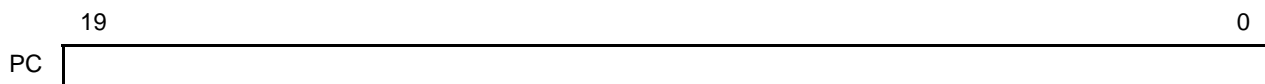
#### (1) 程序计数器（PC）

程序计数器是保存下一次要执行的程序地址信息的 20 位寄存器。

在通常运行时，根据预取的指令码字节数自动进行递增。在执行转移指令时，设定立即数或者寄存器的内容。

在产生复位信号后，给程序计数器的低 16 位设定地址 0000H 和 0001H 的复位向量表的值，并且将程序计数器的高 4 位清“0000”。

图 3-4 程序计数器的结构



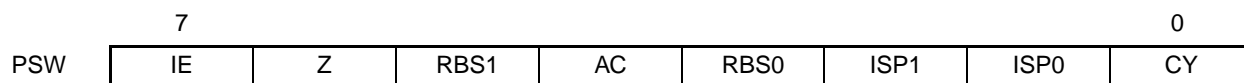
#### (2) 程序状态字（PSW）

程序状态字是由各种标志组成的 8 位寄存器，通过执行指令对这些标识进行置位和清除。

在接受向量中断请求以及执行 PUSH PSW 指令时，将程序状态字的内容保存到堆栈区，而在执行 RETB 指令、RETI 指令或者 POP PSW 指令时恢复程序状态字的内容。

在产生复位信号后，PSW 的值变为“06H”。

图 3-5 程序状态字的结构



#### (a) 中断允许标志（IE）

这是控制 CPU 的中断请求接受运行的标志。

当 IE 位是“0”时，为中断禁止（DI）状态，禁止全部可屏蔽中断。

当 IE 位是“1”时，为中断允许（EI）状态，通过服务优先级标志（ISP1、ISP0）、各中断源的中断屏蔽标志和优先级指定标志进行可屏蔽中断请求的接受控制。

通过执行 DI 指令或者接受中断，将此标志清“0”；通过执行 EI 指令，将此标志置“1”。

#### (b) 零标志（Z）

当运算或者比较结果为零或者相等时，将此标志置“1”。否则，将此标志清“0”。

## (c) 寄存器组选择标志 (RBS0、RBS1)

这是从 4 组寄存器中选择 1 组的 2 位标志。

此标志保存通过执行 SEL RBn 指令所选寄存器组的 2 位信息。

## (d) 辅助进位标志 (AC)

当运算结果在 bit3 产生进位或者借位时，将此标志置“1”。否则，将此标志清“0”。

## (e) 服务优先级标志 (ISP1、ISP0)

这是管理能接受的可屏蔽向量中断优先级的标志。禁止接受优先级指定标志寄存器 (PRn0L、PRn0H、PRn1L) (参照 15.3.3) 指定的低于 ISP0 标志值和 ISP1 标志值的向量中断请求。另外，实际上根据中断允许标志 (IE) 的状态控制是否接受向量中断请求。

备注 n=0、1

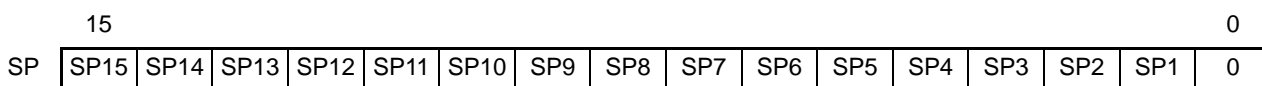
## (f) 进位标志 (CY)

这是在执行加减运算指令时保存上溢和下溢的标志。另外，在执行循环指令时保存移出的值，并且在执行位运算指令时用作位累加器。

## (3) 堆栈指针 (SP)

这是保存存储器堆栈区起始地址的 16 位寄存器。只有内部 RAM 区才能设定为堆栈区。

图 3-6 堆栈指针的结构



在通过堆栈指针进行堆栈寻址的过程中，SP 在写堆栈存储器（压栈）时先递减，而在读堆栈存储器（退栈）后递增。

注意 1. 在产生复位信号后，SP 的内容变为不定值，因此必须在使用堆栈前对 SP 进行初始化。

2. 不能将分配通用寄存器 (FFEE0H ~ FFEFFH) 的空间用于取指令和堆栈区。

3. 在进行自编程或者改写数据闪存时，不能给以下产品的 RAM 区分配各库使用的堆栈和数据缓冲器以及用于 DMA 传送的 RAM 地址。有关详细内容，请确认《RL78 Family Flash Self Programming Library Type01 User's Manual》和《RL78 Family Data Flash Library Type04 User's Manual》。

R7F0C010 : FFE20H ~ FFEFFH

4. 因为以下产品的 RAM 区用于自编程库和数据闪存库，所以禁止使用（参照“图 3-1 存储器映像”）：

R7F0C010 : FF900H ~ FFC80H

3.2.2 通用寄存器

通用寄存器被映像到数据存储器的特定地址（FFEE0H ~ FFEFFH），由 8 个 8 位寄存器（X、A、C、B、E、D、L、H）为 1 组的 4 组寄存器构成。

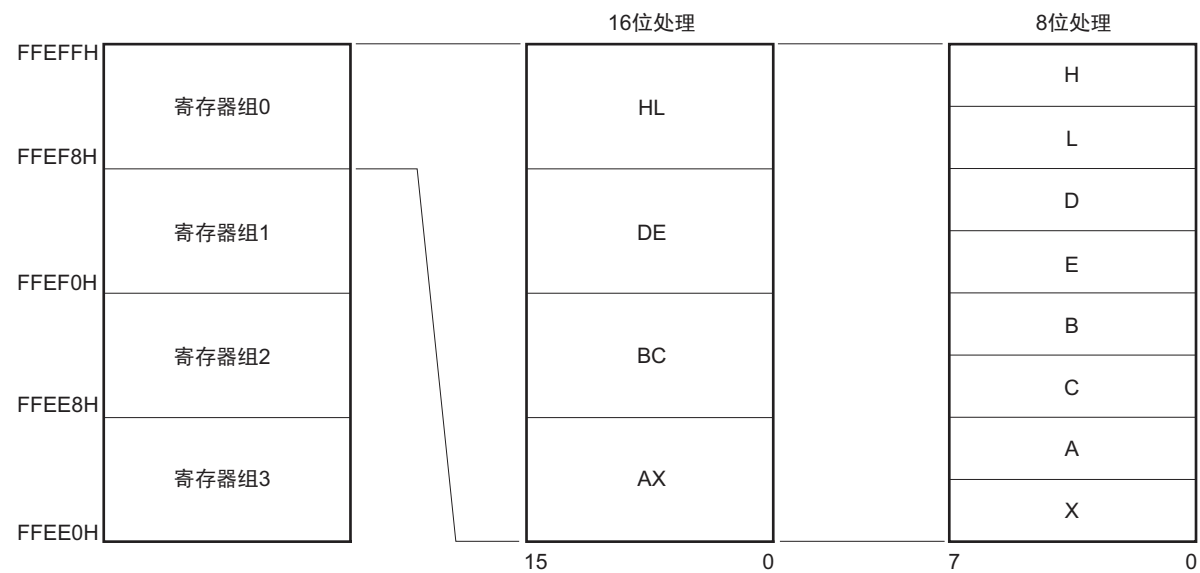
各寄存器除了能分别用作 8 位寄存器以外，还能将 2 个 8 位寄存器成对用作 1 个 16 位寄存器（AX、BC、DE、HL）。

通过 CPU 控制指令（SEL RBn）设定执行指令时使用的寄存器组。因为结构为 4 个寄存器组，所以能对通常处理所用寄存器和中断处理所用寄存器进行寄存器组的切换，建立高效率的程序。

注意 禁止将通用寄存器（FFEE0H ~ FFEFFH）的空间用于取指令和堆栈区。

图 3-7 通用寄存器的结构

(a) 功能名称



### 3.2.3 ES 寄存器和 CS 寄存器

能通过 ES 寄存器和 CS 寄存器（寄存器直接寻址）分别指定存取数据和执行转移指令时的高位地址。ES 寄存器的复位后的初始值为“0FH”，CS 寄存器的复位后的初始值为“00H”。

图 3-8 ES/CS 寄存器的结构

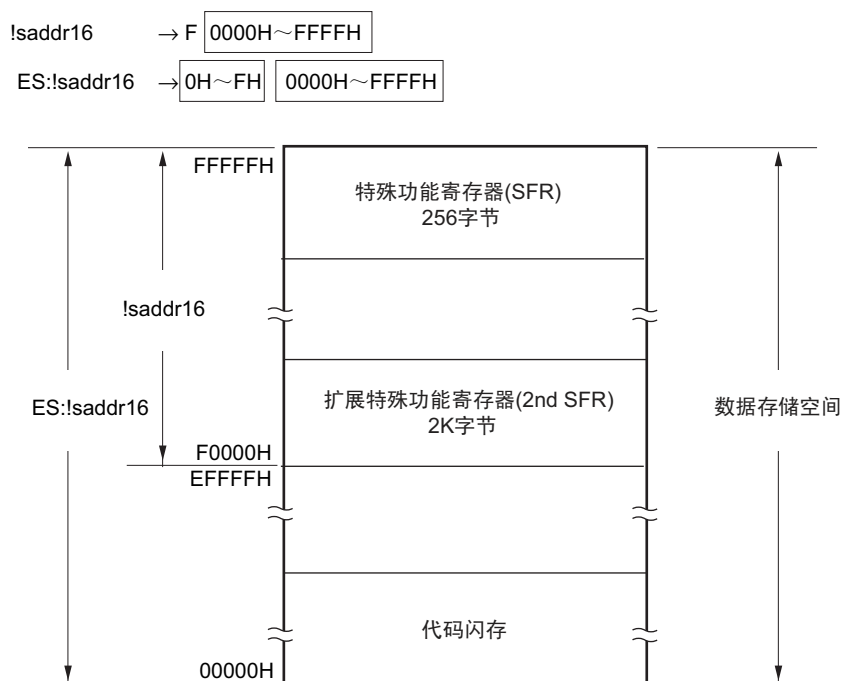
	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CP2	CP1	CP0

能通过 16 位地址进行存取的数据区是 F0000H ~ FFFFFH 的 64K 字节空间，但是如果附加“ES:”，就能扩展到 00000H ~ FFFFFH 的 1M 字节空间。

图 3-9 数据存取区的扩展



### 3.2.4 特殊功能寄存器（SFR：Special Function Register）

SFR 是和通用寄存器不同的并且分别具有特殊功能的寄存器。

SFR 空间分配在 FFF00H ~ FFFFFH 的区域。

和通用寄存器一样，能通过运算指令、传送指令和位操作指令来操作 SFR。能操作的位单位（1、8、16）因各 SFR 而不同。

各操作位单位的指定方法如下所示。

- 位操作  
给位操作指令的操作数（sfr.bit）进行以下的记述：  
已定义位名的情况：<位名>  
没有定义位名的情况：<寄存器名>.<位号>或者<地址>.<位号>
- 8位操作  
给8位操作指令的操作数（sfr）记述汇编程序定义的符号，也能指定地址。
- 16位操作  
给16位操作指令的操作数（sfrp）记述汇编程序定义的符号。当指定地址时，必须记述偶数地址。

SFR 一览表如表 3-5 所示。表中的项目的含义如下所示。

- 符号  
这是表示特殊功能寄存器地址的符号。在汇编程序中为保留字，在编译程序中通过 #pragma sfr 指令定义为 sfr 变量。在使用汇编程序、调试程序和仿真程序时，能记述为指令的操作数。
- R/W  
表示能否读（Read）写（Write）相应的特殊功能寄存器。  
R/W：可读写  
R：只能读  
W：只能写
- 可操作的位单位  
“○”表示能操作的位单位（1、8、16）。“—”表示不能操作的位单位。
- 复位后  
表示产生复位信号后的各寄存器的状态。

**注意** 不能存取未分配 SFR 的地址。

**备注** 有关扩展 SFR（2nd SFR），请参照“3.2.5 扩展特殊功能寄存器（2nd SFR：2nd Special Function Register）”。

表 3-5 特殊功能寄存器（SFR）一览表 (1/3)

地址	特殊功能寄存器（SFR）名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
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
FFF06H	端口寄存器 6	P6		R/W	○	○	—	00H
FFF0CH	端口寄存器 12	P12		R/W	○	○	—	不定值
FFF0DH	端口寄存器 13	P13		R/W	○	○	—	不定值
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/ SIO01	SDR01	R/W	—	○	○	0000H
FFF13H		—			—	—		
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	TDR01L	TDR01	R/W	—	○	○	00H
FFF1BH		TDR01H			—	○		00H
FFF1EH	12 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
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
FFF26H	端口模式寄存器 6	PM6		R/W	○	○	—	FFH
FFF30H	A/D 转换器的模式寄存器 0	ADM0		R/W	○	○	—	00H
FFF31H	模拟输入通道指定寄存器	ADS		R/W	○	○	—	00H
FFF32H	A/D 转换器的模式寄存器 1	ADM1		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
FFF50H	IICA 移位寄存器 0	IICA0		R/W	—	○	—	00H
FFF51H	IICA 状态寄存器 0	IICS0		R	○	○	—	00H
FFF52H	IICA 标志寄存器 0	IICF0		R/W	○	○	—	00H
FFF54H	IICA 移位寄存器 1	IICA1		R/W	—	○	—	00H
FFF55H	IICA 状态寄存器 1	IICS1		R	○	○	—	00H
FFF56H	IICA 标志寄存器 1	IICF1		R/W	○	○	—	00H
FFF58H	D/A 转换值设定寄存器 0	DACS0		R/W	—	—	○	0000H
FFF59H								
FFF5AH	D/A 转换值设定寄存器 1	DACS1		R/W	—	—	○	0000H
FFF5BH								
FFF5CH	D/A 转换器的模式寄存器	DAM		R/W	○	○	—	00H



表 3-5 特殊功能寄存器（SFR）一览表 (2/3)

地址	特殊功能寄存器（SFR）名称	符号	R/W	可操作位的范围			复位后
				1 位	8 位	16 位	
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	TDR03L	R/W	—	○	○	00H
FFF67H		TDR03H		—	○	○	00H
FFFA0H	时钟运行模式控制寄存器	CMC	R/W	—	○	—	00H
FFFA1H	时钟运行状态控制寄存器	CSC	R/W	○	○	—	C0H
FFFA2H	振荡稳定时间计数器的状态寄存器	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	○	○	—	00H
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	—	○	—	不定值 <sup>注1</sup>
FFFA9H	电压检测寄存器	LVIM	R/W	○	○	—	00H <sup>注2</sup>
FFFAAH	电压检测电平寄存器	LVIS	R/W	○	○	—	00H/01H/ 81H <sup>注3</sup>
FFFABH	看门狗定时器允许寄存器	WDTE	R/W	—	○	—	1AH/9AH <sup>注4</sup>
FFFACH	CRC 输入寄存器	CRCIN	R/W	—	○	—	00H
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	—	○	○	00H
FFFB3H	DMA RAM 地址寄存器 0H	DRA0H		—	○	○	00H
FFFB4H	DMA RAM 地址寄存器 1L	DRA1L	DRA1	—	○	○	00H
FFFB5H	DMA RAM 地址寄存器 1H	DRA1H		—	○	○	00H
FFFB6H	DMA 字节计数寄存器 0L	DBC0L	DBC0	—	○	○	00H
FFFB7H	DMA 字节计数寄存器 0H	DBC0H		—	○	○	00H
FFFB8H	DMA 字节计数寄存器 1L	DBC1L	DBC1	—	○	○	00H
FFFB9H	DMA 字节计数寄存器 1H	DBC1H		—	○	○	00H
FFFBABH	DMA 模式控制寄存器 0	DMC0	R/W	○	○	—	00H
FFFBABH	DMA 模式控制寄存器 1	DMC1	R/W	○	○	—	00H
FFFBCH	DMA 运行控制寄存器 0	DRC0	R/W	○	○	—	00H
FFFBDBH	DMA 运行控制寄存器 1	DRC1	R/W	○	○	—	00H

- 注 1. RESF 寄存器的复位值因复位源而变。
- 注 2. LVIM 寄存器的复位值因复位源而变。
- 注 3. LVIS 寄存器的复位值因复位源和选项字节的设定而变。
- 注 4. WDTE 寄存器的复位值取决于选项字节的设定。

表 3-5 特殊功能寄存器（SFR）一览表 (3/3)

地址	特殊功能寄存器（SFR）名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
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		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		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
FFFEAH	优先级指定标志寄存器 01L	PR01L		R/W	○	○	—	FFH
FFFECH	优先级指定标志寄存器 10L	PR10L	PR10	R/W	○	○	○	FFH
FF FEDH	优先级指定标志寄存器 10H	PR10H		R/W	○	○		FFH
FFFE EH	优先级指定标志寄存器 11L	PR11L		R/W	○	○	—	FFH
FFFFEH	处理器模式控制寄存器	PMC		R/W	○	○	—	00H

备注 有关扩展 SFR（2nd SFR），请参照“表 3-6 扩展特殊功能寄存器（2nd SFR）一览表”。

### 3.2.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）

扩展 SFR（2nd SFR）是和通用寄存器不同的并且分别具有特殊功能的寄存器。

扩展 SFR 空间分配在 F0000H ~ F07FFH 的区域。在此区域中分配了 SFR 区（FFF00H ~ FFFFFH）以外的 SFR，但是扩展 SFR 区的存取指令比 SFR 区长 1 字节。

和通用寄存器一样，能通过运算指令、传送指令和位操作指令来操作扩展 SFR。能操作的位单位（1、8、16）因各扩展 SFR 而不同。

各操作位单位的指定方法如下所示。

- 位操作  
给位操作指令的操作数（!addr16.bit）进行以下的记述：  
已定义位名的情况：<位名>  
没有定义位名的情况：<寄存器名>.<位号>或者<地址>.<位号>
- 8位操作  
给8位操作指令的操作数（!addr16）记述汇编程序定义的符号，也能指定地址。
- 16位操作  
给16位操作指令的操作数（!addr16）记述汇编程序定义的符号。当指定地址时，必须记述偶数地址。

扩展 SFR 一览表如表 3-6 所示。表中的项目的含义如下所示。

- 符号  
这是表示扩展 SFR 地址的符号。在汇编程序中为保留字，在编译程序中通过 #pragma sfr 指令定义为 sfr 变量。在使用汇编程序、调试程序和仿真程序时，能记述为指令的操作数。
- R/W  
表示能否读（Read）写（Write）相应的扩展 SFR。  
R/W：可读写  
R：只能读  
W：只能写
- 可操作的位单位  
“○”表示能操作的位单位（1、8、16）。“—”表示不能操作的位单位。
- 复位后  
表示产生复位信号后的各寄存器的状态。

**注意** 不能存取未分配扩展 SFR（2nd SFR）的地址。

**备注** 有关 SFR 区的 SFR，请参照“3.2.4 特殊功能寄存器（SFR: Special Function Register）”。

表 3-6 扩展特殊功能寄存器（2nd SFR）一览表 (1/4)

地址	扩展特殊功能寄存器 （2nd SFR）名称	符号	R/W	可操作位的范围			复位后
				1 位	8 位	16 位	
F0010H	A/D 转换器的模式寄存器 2	ADM2	R/W	○	○	—	00H
F0011H	转换结果比较上限值设定寄存器	ADUL	R/W	—	○	—	FFH
F0012H	转换结果比较下限值设定寄存器	ADLL	R/W	—	○	—	00H
F0013H	A/D 测试寄存器	ADTES	R/W	—	○	—	00H
F0031H	上拉电阻选择寄存器 1	PU1	R/W	○	○	—	00H
F0033H	上拉电阻选择寄存器 3	PU3	R/W	○	○	—	00H
F0034H	上拉电阻选择寄存器 4	PU4	R/W	○	○	—	01H
F0061H	端口模式控制寄存器 1	PMC1	R/W	○	○	—	FFH
F0070H	噪声滤波器允许寄存器 0	NFEN0	R/W	○	○	—	00H
F0071H	噪声滤波器允许寄存器 1	NFEN1	R/W	○	○	—	00H
F0073H	输入切换控制寄存器	ISC	R/W	○	○	—	00H
F0074H	定时器输入选择寄存器 0	TIS0	R/W	—	○	—	00H
F0076H	A/D 端口配置寄存器	ADPC	R/W	—	○	—	00H
F0078H	非法存储器存取检测控制寄存器	IAWCTL	R/W	—	○	—	00H
F007AH	外围允许寄存器 1	PER1	R/W	○	○	—	00H
F0090H	数据闪存控制寄存器	DFLCTL	R/W	○	○	—	00H
F00A0H	高速内部振荡器的微调寄存器	HIOTRM	R/W	—	○	—	不定值 <sup>注</sup>
F00A8H	高速内部振荡器的频率选择寄存器	HOCODIV	R/W	—	○	—	不定值
F00F0H	外围允许寄存器 0	PER0	R/W	○	○	—	00H
F00F5H	RAM 奇偶校验错误控制寄存器	RPECTL	R/W	○	○	—	00H
F00FEH	BCD 校正结果寄存器	BCDADJ	R	—	○	—	不定值
F0100H	串行状态寄存器 00	SSR00L	SSR00	—	○	○	0000H
F0101H		—					
F0102H	串行状态寄存器 01	SSR01L	SSR01	—	○	○	0000H
F0103H		—					
F0108H	串行标志清除触发寄存器 00	SIR00L	SIR00	—	○	○	0000H
F0109H		—					
F010AH	串行标志清除触发寄存器 01	SIR01L	SIR01	—	○	○	0000H
F010BH		—					

注 复位值是产品发货时的调整值。

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (2/4)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			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								
F0118H	串行通信运行设定寄存器 00	SCR00		R/W	—	—	○	0087H
F0119H								
F011AH	串行通信运行设定寄存器 01	SCR01		R/W	—	—	○	0087H
F011BH								
F0120H	串行通道允许状态寄存器 0	SE0L	SE0	R	○	○	○	0000H
F0121H		—			—	—		
F0122H	串行通道开始寄存器 0	SS0L	SS0	R/W	○	○	○	0000H
F0123H		—			—	—		
F0124H	串行通道停止寄存器 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	—	—	○	0303H
F0129H								
F012AH	串行输出允许寄存器 0	SOE0L	SOE0	R/W	○	○	○	0000H
F012BH		—			—	—		
F0134H	串行输出电平寄存器 0	SOL0L	SOL0	R/W	—	○	○	0000H
F0135H		—			—	—		
F0138H	串行待机控制寄存器 0	SSC0L	SSC0	R/W	—	○	○	0000H
		—			—	—		
F0180H	定时器计数寄存器 00	TCR00		R	—	—	○	FFFFH
F0181H								
F0182H	定时器计数寄存器 01	TCR01		R	—	—	○	FFFFH
F0183H								
F0184H	定时器计数寄存器 02	TCR02		R	—	—	○	FFFFH
F0185H								
F0186H	定时器计数寄存器 03	TCR03		R	—	—	○	FFFFH
F0187H								
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								

表 3-6 扩展特殊功能寄存器 (2nd SFR) 一览表 (3/4)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
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		—			—			
F01B0H	定时器通道允许状态寄存器 0	TE0L	TE0	R	○	○	○	0000H
F01B1H		—			—			
F01B2H	定时器通道开始寄存器 0	TS0L	TS0	R/W	○	○	○	0000H
F01B3H		—			—			
F01B4H	定时器通道停止寄存器 0	TT0L	TT0	R/W	○	○	○	0000H
F01B5H		—			—			
F01B6H	定时器时钟选择寄存器 0	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		—			—			
F0230H	IICA 控制寄存器 00	IICCTL00		R/W	○	○	—	00H
F0231H	IICA 控制寄存器 01	IICCTL01		R/W	○	○	—	00H
F0232H	IICA 低电平宽度设定寄存器 0	IICWL0		R/W	—	○	—	FFH
F0233H	IICA 高电平宽度设定寄存器 0	IICWH0		R/W	—	○	—	FFH
F0234H	从属地址寄存器 0	SVA0		R/W	—	○	—	00H
F0238H	IICA 控制寄存器 10	IICCTL10		R/W	○	○	—	00H
F0239H	IICA 控制寄存器 11	IICCTL11		R/W	○	○	—	00H
F023AH	IICA 低电平宽度设定寄存器 1	IICWL1		R/W	—	○	—	FFH
F023BH	IICA 高电平宽度设定寄存器 1	IICWH1		R/W	—	○	—	FFH
F023CH	从属地址寄存器 1	SVA1		R/W	—	○	—	00H
F02F0H	闪存 CRC 控制寄存器	CRC0CTL		R/W	○	○	—	00H
F02F2H	闪存 CRC 运算结果寄存器	PGCRCL		R/W	—	—	○	0000H
F02FAH	CRC 数据寄存器	CRCD		R/W	—	—	○	0000H

表 3-6 扩展特殊功能寄存器（2nd SFR）一览表 (4/4)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号	R/W	可操作位的范围			复位后
				1 位	8 位	16 位	
F0300H	事件链接设定寄存器 00	ELSELR00	R/W	○	○	—	00H
F0301H	事件链接设定寄存器 01	ELSELR01	R/W	○	○	—	00H
F0302H	事件链接设定寄存器 02	ELSELR02	R/W	○	○	—	00H
F0303H	事件链接设定寄存器 03	ELSELR03	R/W	○	○	—	00H
F0304H	事件链接设定寄存器 04	ELSELR04	R/W	○	○	—	00H
F0305H	事件链接设定寄存器 05	ELSELR05	R/W	○	○	—	00H
F0306H	事件链接设定寄存器 06	ELSELR06	R/W	○	○	—	00H
F0307H	事件链接设定寄存器 07	ELSELR07	R/W	○	○	—	00H
F0308H	事件链接设定寄存器 08	ELSELR08	R/W	○	○	—	00H
F0309H	事件链接设定寄存器 09	ELSELR09	R/W	○	○	—	00H

备注 有关 SFR 区的 SFR，请参照“表 3-5 特殊功能寄存器（SFR）一览表”。

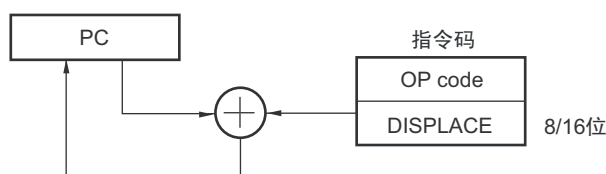
### 3.3 指令地址的寻址

#### 3.3.1 相对寻址

##### 【功能】

相对寻址将指令码中的位移量（带符号的补码数据：-128 ~ +127 或者 -32768 ~ +32767）加上程序计数器（PC）的值（下一条指令的起始地址），结果保存在程序计数器（PC）并且指定转移目标的程序地址。相对寻址只适用于转移指令。

图 3-10 相对寻址的概要



#### 3.3.2 立即寻址

##### 【功能】

立即寻址将指令码中的立即数保存到程序计数器，指定转移目标的程序地址。

在立即寻址中有指定 20 位地址的 CALL !!addr20/BR !!addr20 和指定 16 位地址的 CALL !addr16/BR !addr16。当指定 16 位地址时，将高 4 位置“0000”。

图 3-11 CALL !!addr20/BR !!addr20 的例子

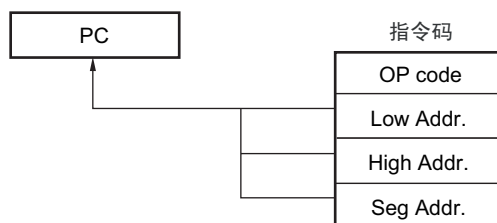
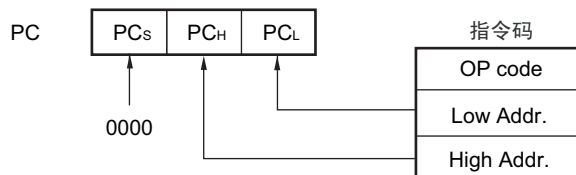


图 3-12 CALL !addr16/BR !addr16 的例子





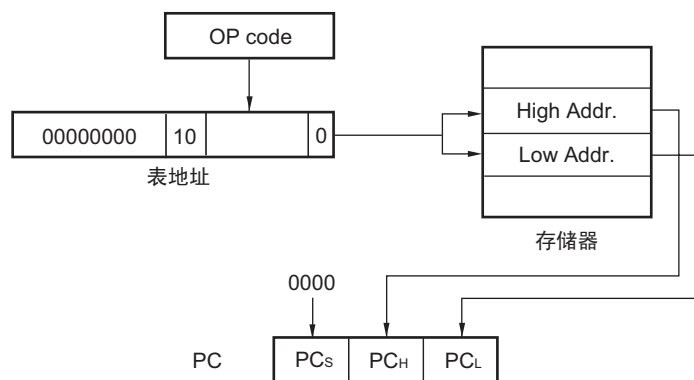
### 3.3.3 表间接寻址

#### 【功能】

表间接寻址通过指令码中的 5 位立即数指定 CALLT 表区（0080H ~ 00BFH）中的表地址，将此内容和其后续的地址内容作为 16 位数据保存到程序计数器（PC），指定程序地址。表间接寻址只适用于 CALLT 指令。

RL78 微控制器只能在 00000H ~ 0FFFFH 的 64K 字节空间中进行转移。

图 3-13 表间接寻址的概要

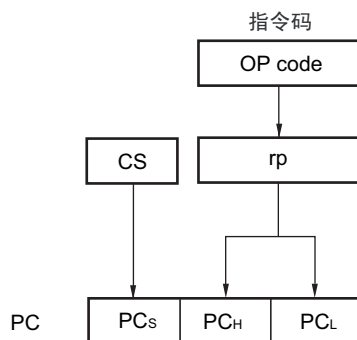


### 3.3.4 寄存器直接寻址

#### 【功能】

寄存器直接寻址将指令码指定的当前寄存器组的通用寄存器对（AX/BC/DE/HL）和 CS 寄存器的内容作为 20 位数据保存到程序计数器（PC），指定程序地址。寄存器直接寻址只适用于 CALL AX/BC/DE/HL 和 BR AX 指令。

图 3-14 寄存器直接寻址的概要



3.4 处理数据地址的寻址

3.4.1 隐含寻址

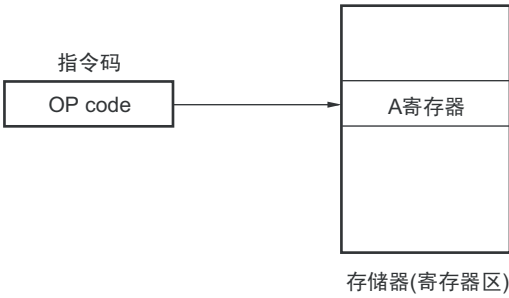
【功能】

对于具有累加器等特殊功能寄存器的存取指令，在指令码中没有寄存器指定字段，而通过指令码直接指定。

【操作数形式】

隐含寻址只适用于 MULU X 指令。

图 3-15 隐含寻址的概要



3.4.2 寄存器寻址

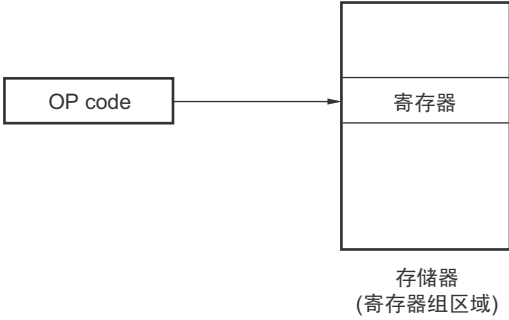
【功能】

寄存器寻址是将通用寄存器作为操作数进行存取的寻址方式。当指定 8 位寄存器时，通过指令码中的 3 位选择寄存器；当指定 16 位寄存器时，通过指令码中的 2 位选择寄存器。

【操作数形式】

表现形式	记述方法
r	X、A、C、B、E、D、L、H
rp	AX、BC、DE、HL

图 3-16 寄存器寻址的概要



3.4.3 直接寻址

【功能】

直接寻址是以指令码中的立即数为操作数地址来直接指定对象地址的寻址方式。

【操作数形式】

表现形式	记述方法
!addr16	标号或者 16 位立即数 (只能指定 F0000H ~ FFFFFH 的空间)
ES:!addr16	标号或者 16 位立即数 (通过 ES 寄存器指定高 4 位地址)

图 3-17 !addr16 的例子

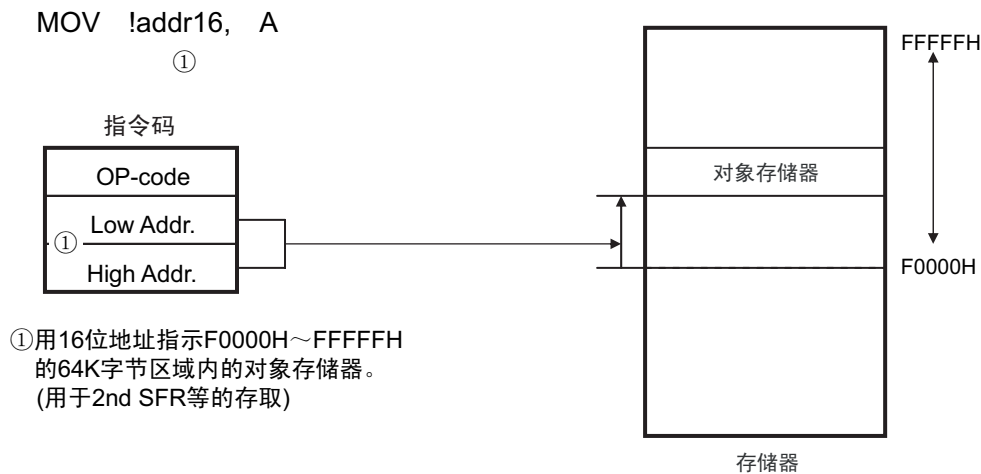
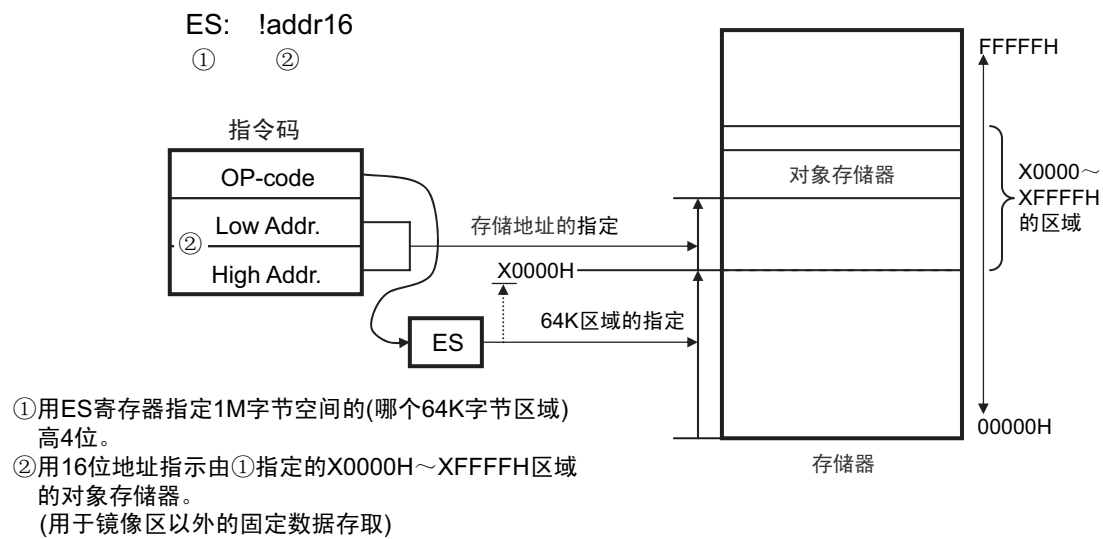


图 3-18 ES:!addr16 的例子



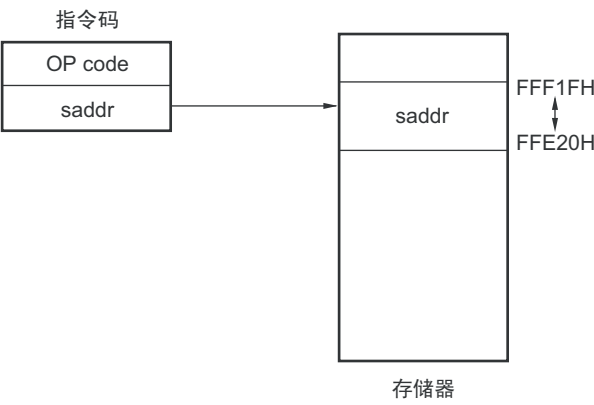
3.4.4 短直接寻址

**【功能】**  
短直接寻址是通过指令码中的8位数据直接指定对象地址的寻址方式。此寻址方式只适用于FFE20H～FFF1FH的空间。

**【操作数形式】**

表现形式	记述方法
SADDR	标号、 FFE20H ～ FFF1FH 的立即数
SADDRP	标号、 FFE20H ～ FFF1FH 的立即数 （只限于偶数地址）

图 3-19 短直接寻址的概要



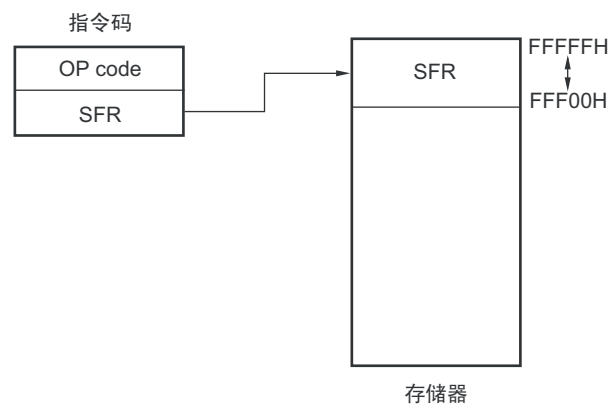
**备注** SADDR 和 SADDRP 能通过 16 位立即数（省略了实际地址的高 4 位）记述 FE20H ～ FF1FH 的值，并且还能通过 20 位立即数记述 FFE20H ～ FFF1FH 的值。  
但是，无论用哪种形式，都指定存储器的FFE20H～FFF1FH空间地址。

3.4.5 SFR 寻址

- 【功能】
- SFR 寻址是通过指令码中的 8 位数据直接指定对象 SFR 地址的寻址方式。此寻址方式只适用于 FFF00H ~ FFFFFH 的空间。
- 【操作数形式】

表现形式	记述方法
SFR	SFR 寄存器名
SFRP	16 位可操作的 SFR 寄存器名（偶数地址）

图 3-20 SFR 寻址的概要



3.4.6 寄存器间接寻址

【功能】

寄存器间接寻址以指令码指定的寄存器对的内容为操作数地址，指定对象地址。

【操作数形式】

表现形式	记述方法
—	[DE]、[HL] (只能指定 F0000H ~ FFFFFH 的空间)
—	ES:[DE]、ES:[HL] (通过 ES 寄存器指定高 4 位地址)

图 3-21 [DE]、[HL] 的例子

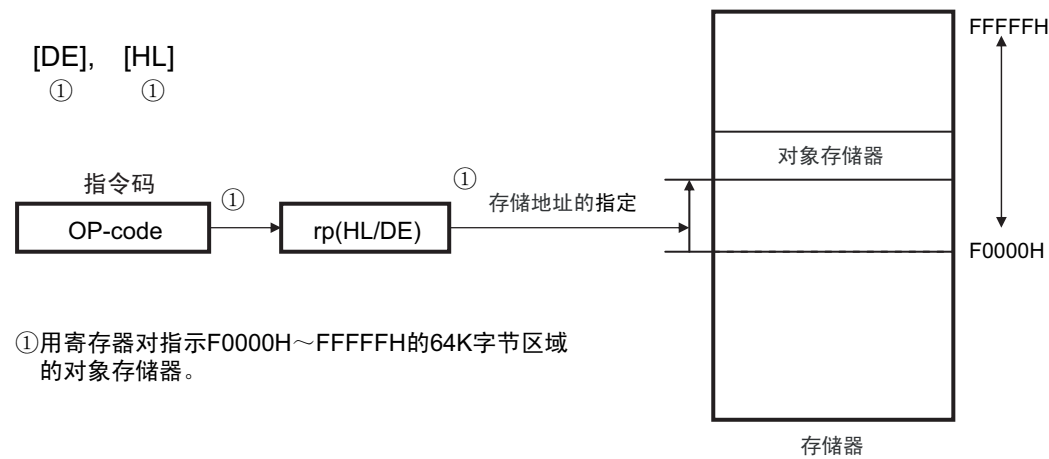
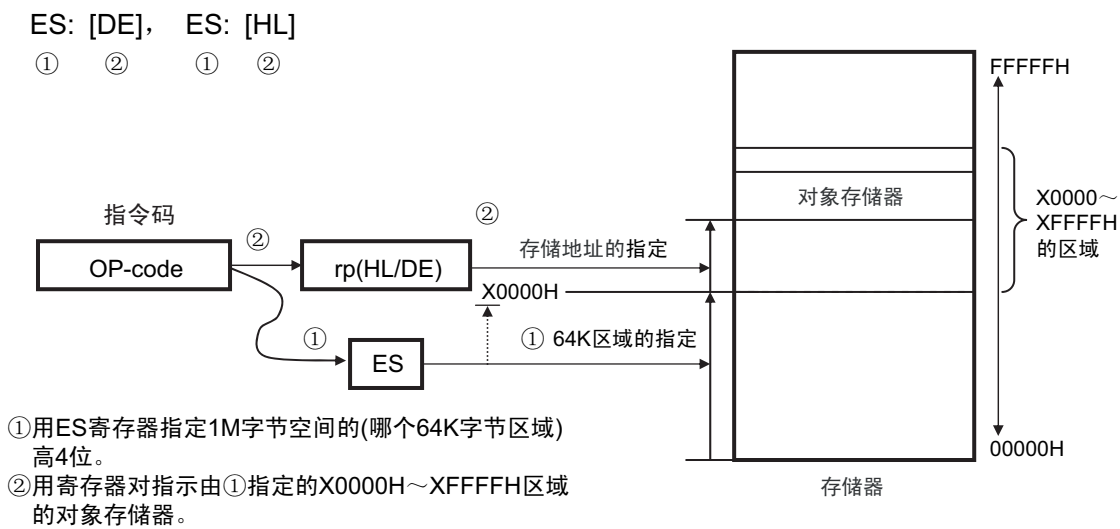


图 3-22 ES:[DE]、ES:[HL] 的例子



## 3.4.7 基址寻址

## 【功能】

基址寻址以指令码指定的寄存器对的内容或者 16 位立即数为基址，以 8 位立即数或者 16 位立即数为偏移量，用基址和偏移量的相加结果指定对象地址。

## 【操作数形式】

表现形式	记述方法
—	[HL+byte]、[DE+byte]、[SP+byte] (只能指定 F0000H ~ FFFFFH 的空间)
—	word[B]、word[C] (只能指定 F0000H ~ FFFFFH 的空间)
—	word[BC] (只能指定 F0000H ~ 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 3-23 [SP+byte] 的例子

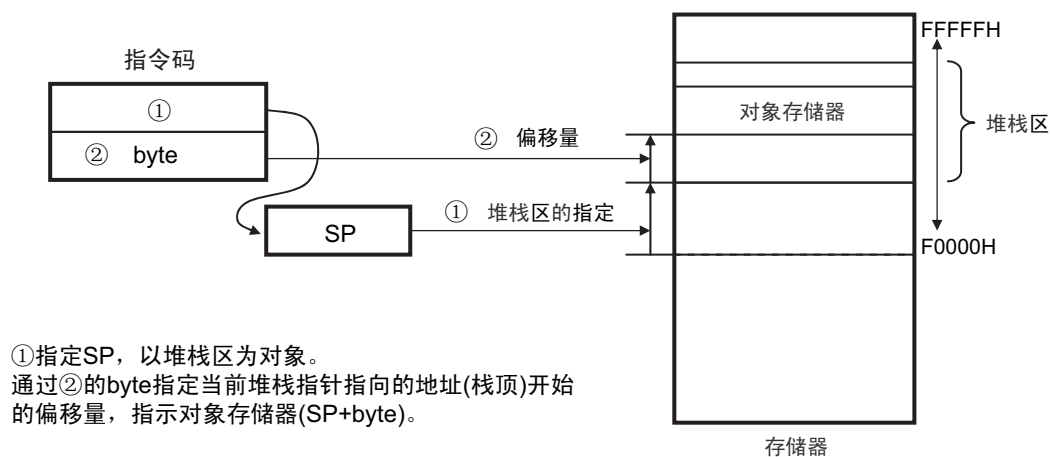


图 3-24 [HL+byte]、[DE+byte] 的例子

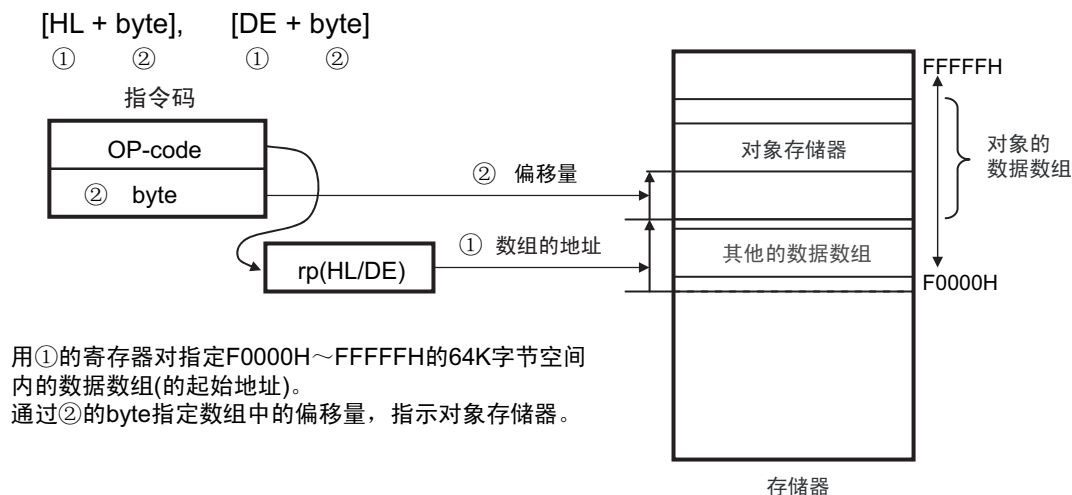


图 3-25 word[B]、word[C] 的例子

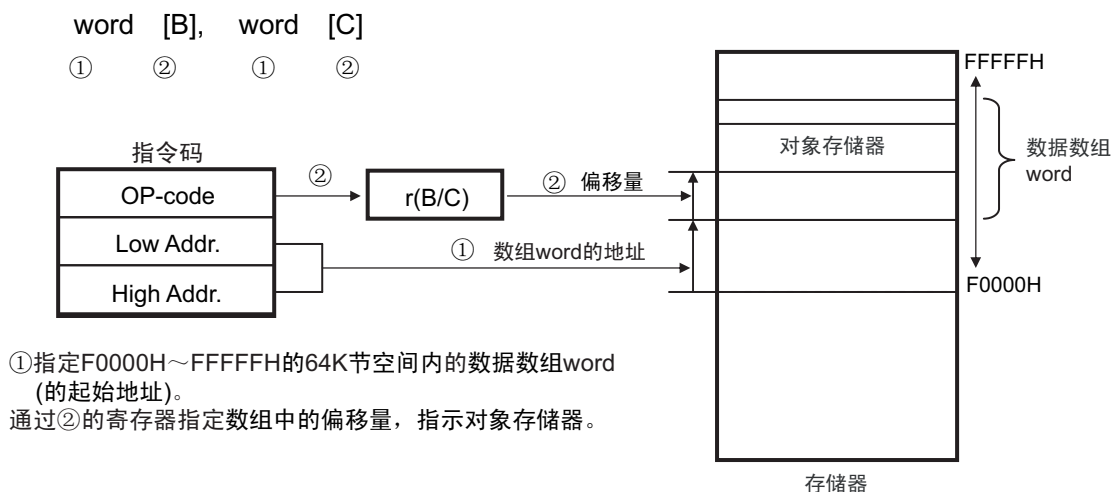


图 3-26 word[BC] 的例子

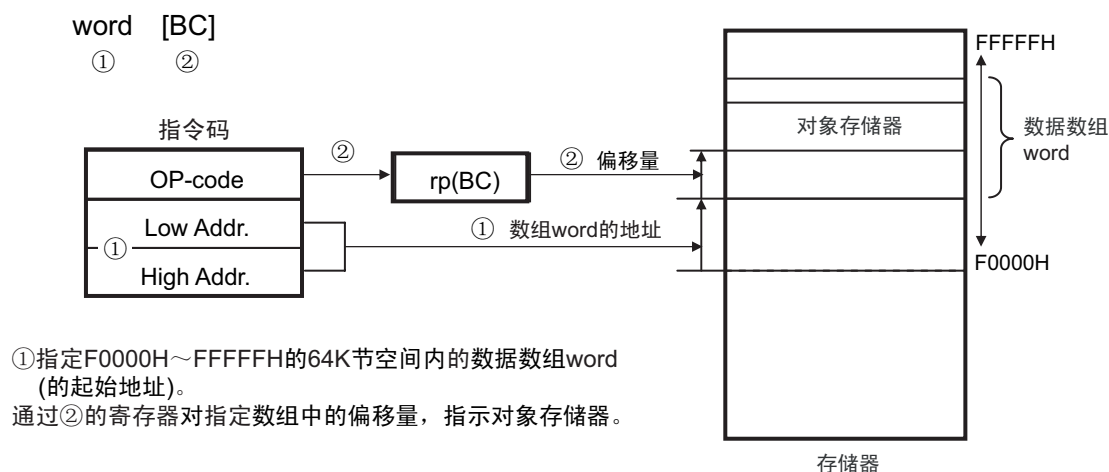




图 3-27 ES:[HL+byte]、ES:[DE+byte] 的例子

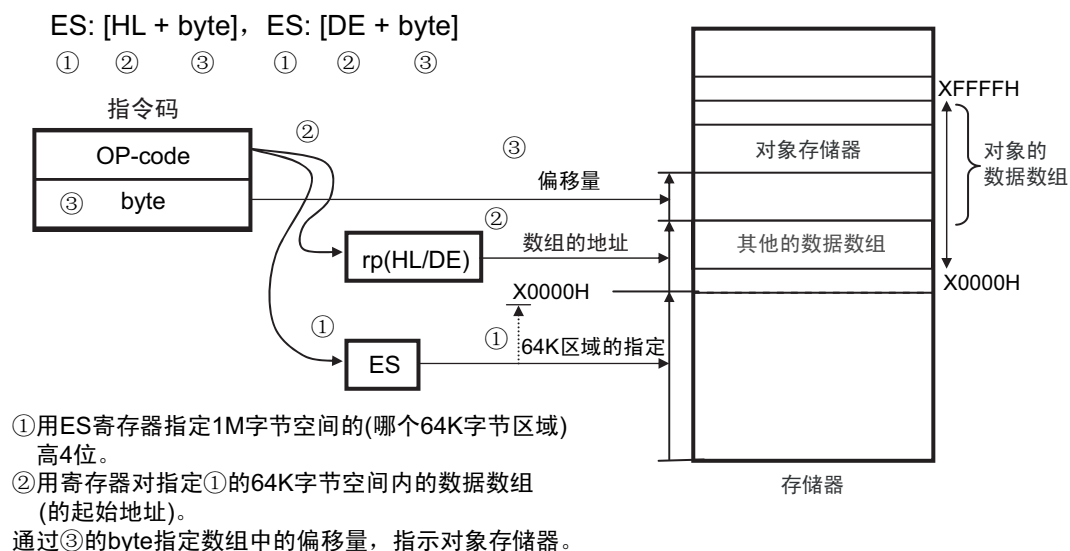


图 3-28 ES:word[B]、ES:word[C] 的例子

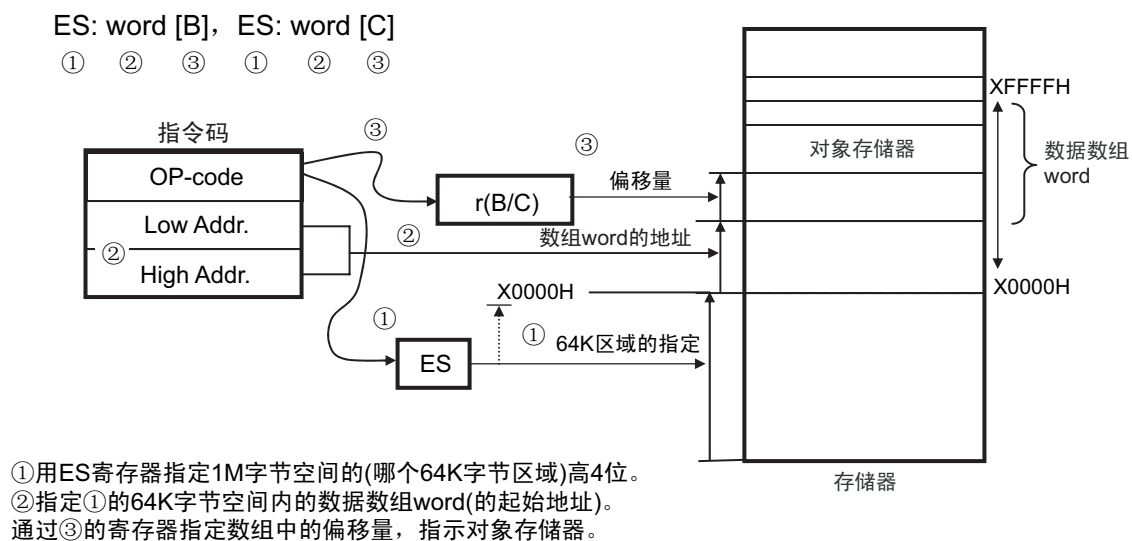
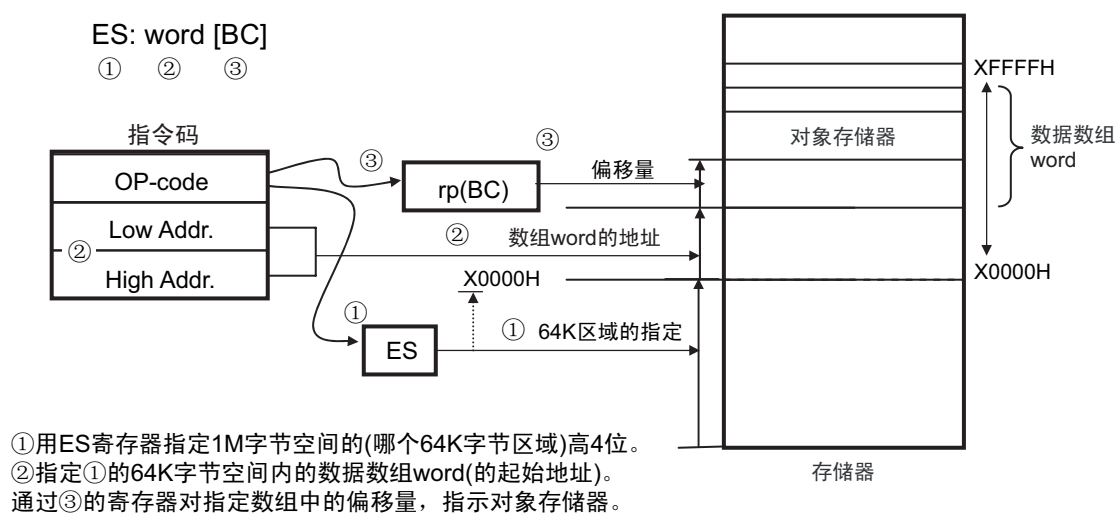


图 3-29 ES:word[BC] 的例子



3.4.8 基址变址寻址

【功能】

基址变址寻址以指令码指定的寄存器对的内容为基址，以指令码指定的 B 寄存器或者 C 寄存器的内容为偏移地址，用基址和偏移地址的相加结果指定对象地址。

【操作数形式】

表现形式	记述方法
—	[HL+B]、[HL+C] (只能指定 F0000H ~ FFFFFH 的空间)
—	ES:[HL+B]、ES:[HL+C] (通过 ES 寄存器指定高 4 位地址)

图 3-30 [HL+B]、[HL+C] 的例子

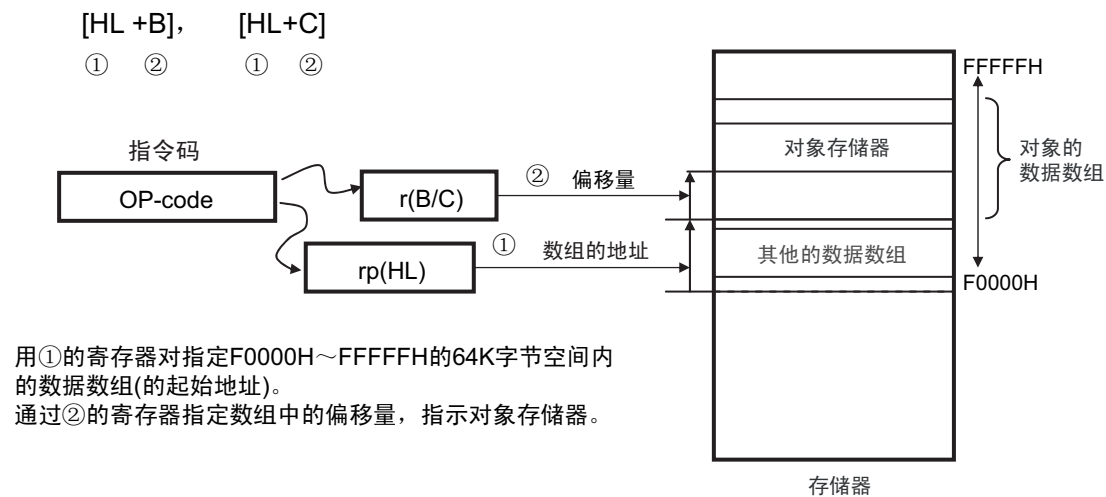
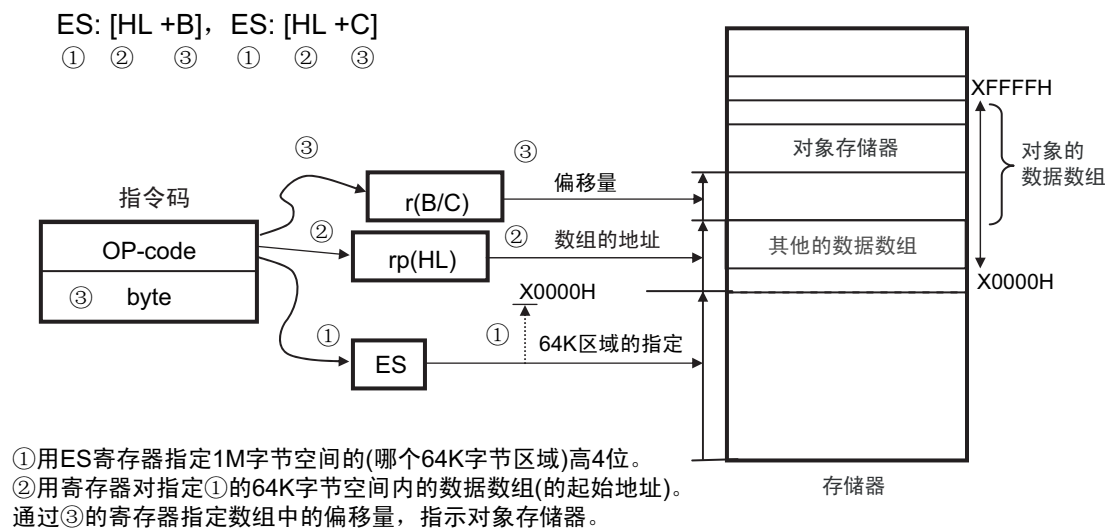


图 3-31 ES:[HL+B]、ES:[HL+C] 的例子



3.4.9 堆栈寻址

【功能】

堆栈寻址是通过堆栈指针（SP）的值间接指定堆栈区域的寻址方式。当执行 PUSH、POP、子程序调用和返回指令时，或者在因产生中断请求而保存 / 恢复寄存器时，自动使用这种寻址方式。

只能将堆栈区设定在内部 RAM 区。

【记述形式】

表现形式	记述方法
—	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (发生中断请求) RETI

根据各堆栈的运行，压栈和退栈的数据如图 3-32 ~ 图 3-37 所示。

图 3-32 PUSH rp 的例子

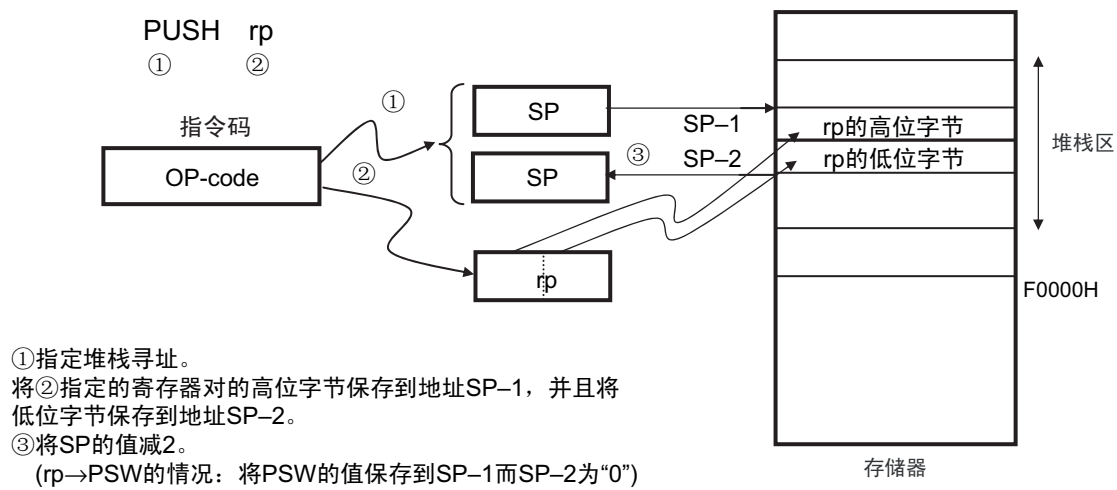


图 3-33 POP 的例子

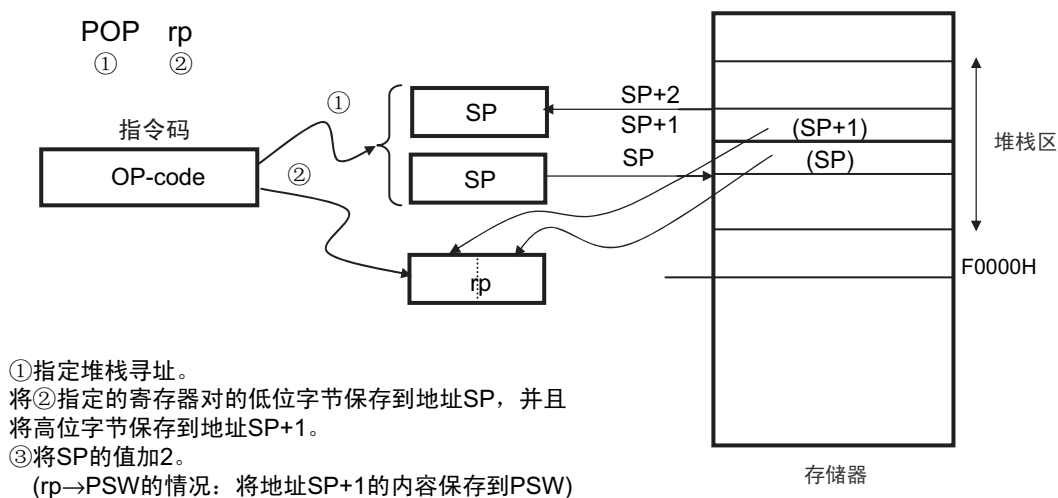


图 3-34 CALL、CALLT 的例子

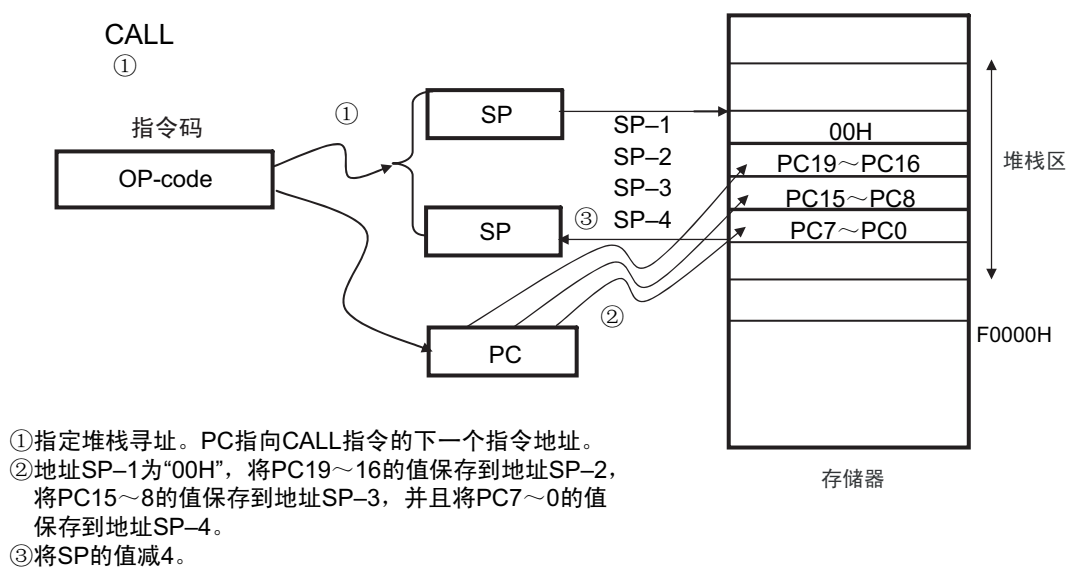


图 3-35 RET 的例子

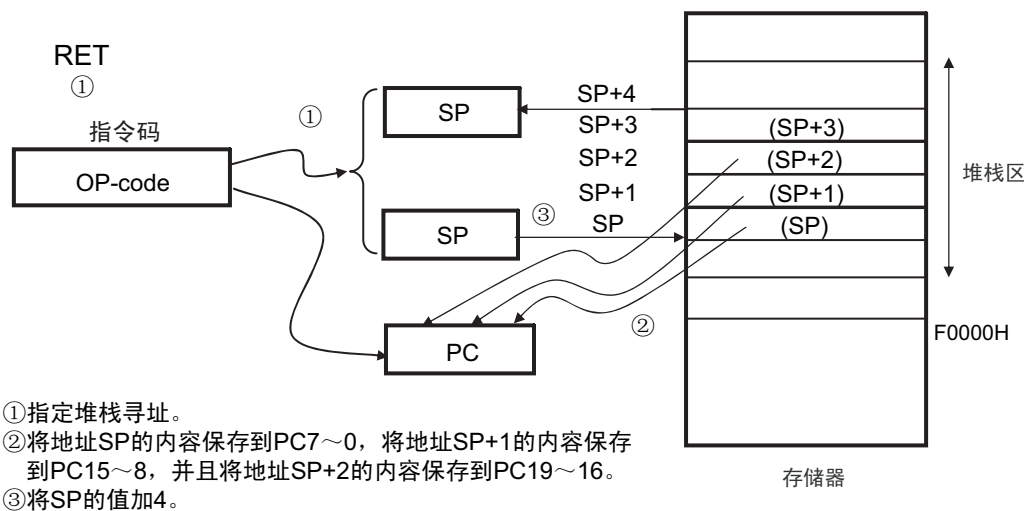


图 3-36 中断、BRK 的例子

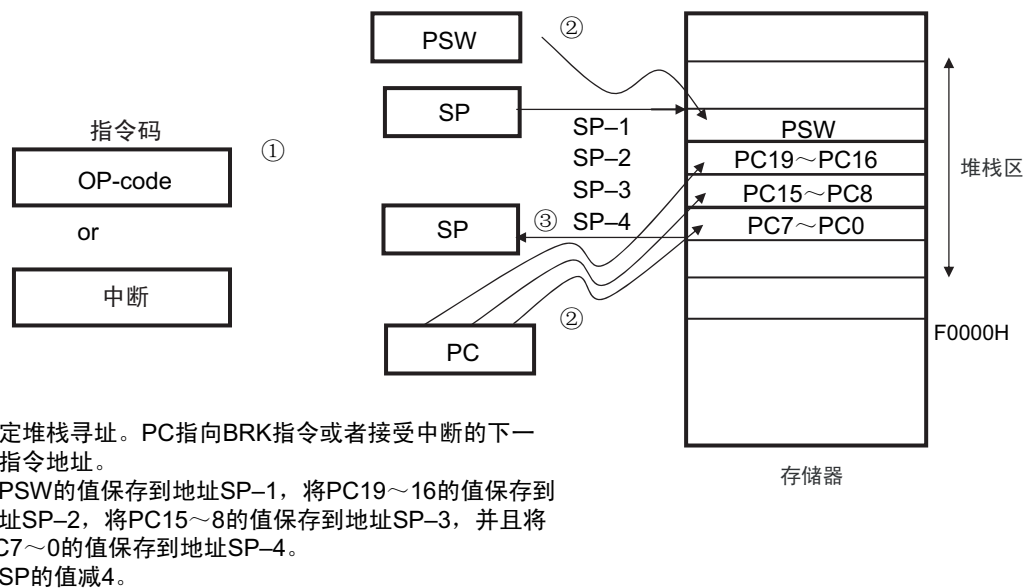
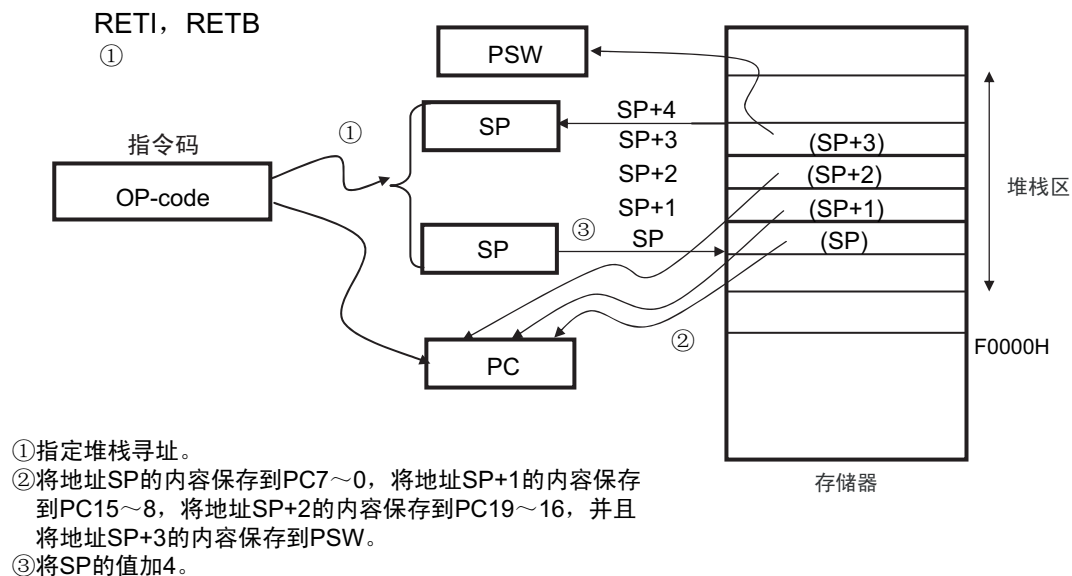


图 3-37 RETI、RETB 的例子



## 第 4 章 端口功能

### 4.1 端口功能

R7F0C010 提供数字输入 / 输出端口，能进行各种各样的控制。

除了作为数字输入 / 输出端口的功能以外，还提供各种复用功能。有关复用功能，请参照“第 2 章 引脚功能”。

### 4.2 端口结构

端口由以下硬件构成。

表 4-1 端口结构

项目	结构
控制寄存器	端口模式寄存器（PM1 ~ PM4、PM6） 端口寄存器（P1 ~ P4、P6、P12、P13） 上拉电阻选择寄存器（PU1、PU3、PU4） 端口模式控制寄存器 1（PMC1） A/D 端口配置寄存器（ADPC）
端口	<ul style="list-style-type: none"> <li>24 引脚产品： 合计：20 个（CMOS 输入 / 输出：15 个，CMOS 输入：3 个，N 沟道漏极开路输入 / 输出：2 个）</li> <li>32 引脚产品： 合计：28 个（CMOS 输入 / 输出：23 个，CMOS 输入：3 个，N 沟道漏极开路输入 / 输出：2 个）</li> </ul>
上拉电阻	<ul style="list-style-type: none"> <li>24 引脚产品：合计：10 个</li> <li>32 引脚产品：合计：15 个</li> </ul>



注意 本章的下述内容主要针对 32 引脚产品进行说明。

#### 4.2.1 端口 1

端口 1 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 1（PM1）以位为单位指定输入模式或者输出模式。当将 P10 ~ P17 引脚用作输入端口时，能通过上拉电阻选择寄存器 1（PU1）以位为单位使用内部上拉电阻。

端口 1 作为复用功能，有定时器的输入 / 输出、外部中断的请求输入和时钟 / 蜂鸣器的输出。

在产生复位信号后，端口 1 变为输入模式。

表 4-2 使用端口 1 时的寄存器设定

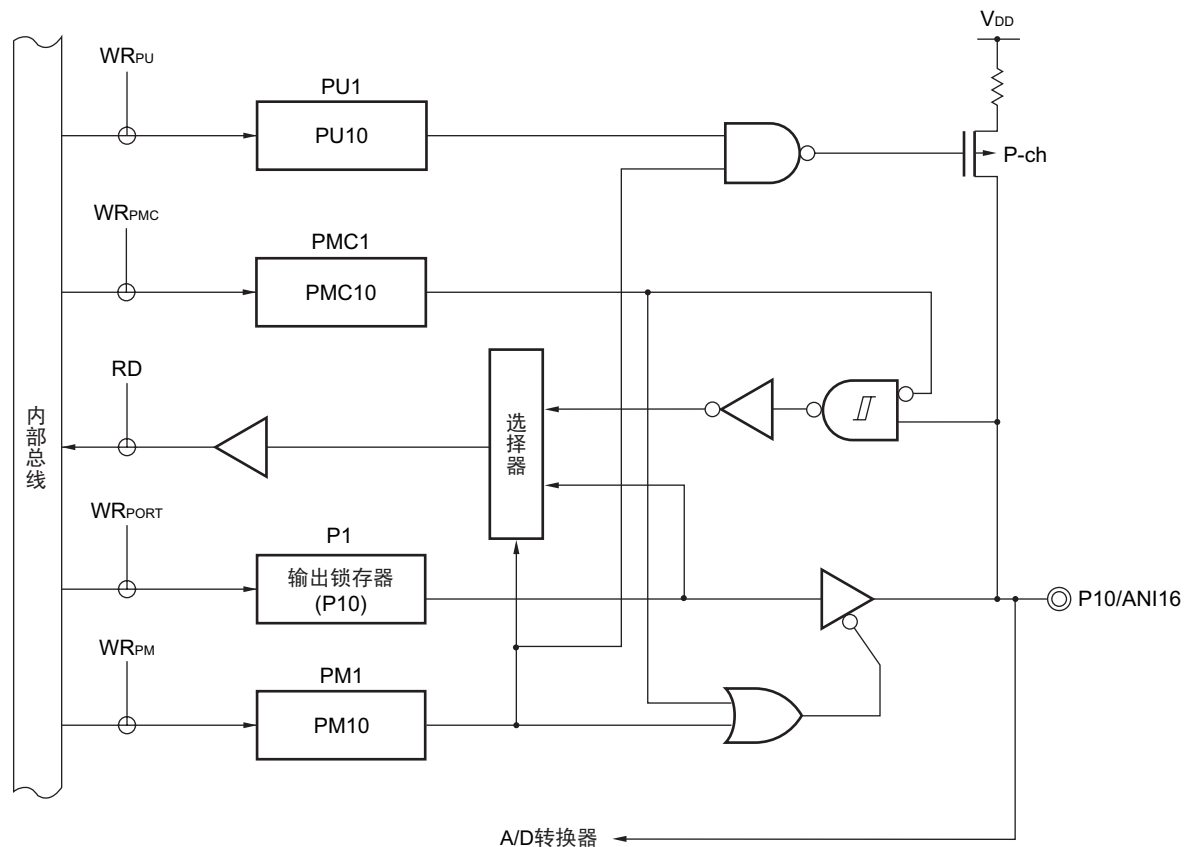
端口	输入 / 输出	PM1x	复用功能的设定	备注
P10	输入	1	×	CMOS 输入
	输出	0	×	CMOS 输出
P11	输入	1	×	CMOS 输入
	输出	0	×	CMOS 输出
P12	输入	1	×	CMOS 输入
	输出	0	PCLBUZ0 输出 =0 注 1 TO03 输出 =0 注 3	CMOS 输出
P13	输入	1	×	CMOS 输入
	输出	0	TO00 输出 =0 注 3	CMOS 输出
P14	输入	1	×	CMOS 输入
	输出	0	×	CMOS 输出
P15	输入	1	×	CMOS 输入
	输出	0	PCLBUZ1 输出 =0 注 2	CMOS 输出
P16	输入	1	×	CMOS 输入
	输出	0	TO01 输出 =0 注 3	CMOS 输出
P17	输入	1	×	CMOS 输入
	输出	0	×	CMOS 输出

- 注 1. 当将 P12/TI03/TO03/INTP4/PCLBUZ0 用作通用端口时，必须将时钟输出选择寄存器 0（CKS0）的 bit7（RCLOE0）置“0”（初始状态）。
2. 当将 P15/PCLBUZ1 用作通用端口时，必须将时钟输出选择寄存器 1（CKS1）的 bit7（RCLOE1）置“0”（初始状态）。
3. 当将 P12/TI03/TO03/INTP4/PCLBUZ0、P13/TI00/TO00、P16/TI01/TO01/INTP5 用作通用端口时，必须将定时器输出寄存器 0（TO0）的 bit0、1、3（TO00、TO01、TO03）和定时器输出允许寄存器 0（TOE0）的 bit0、1、3（TOE00、TOE01、TOE03）置“0”（初始状态）。

备注 × : 忽略  
PM1x : 端口模式寄存器 1

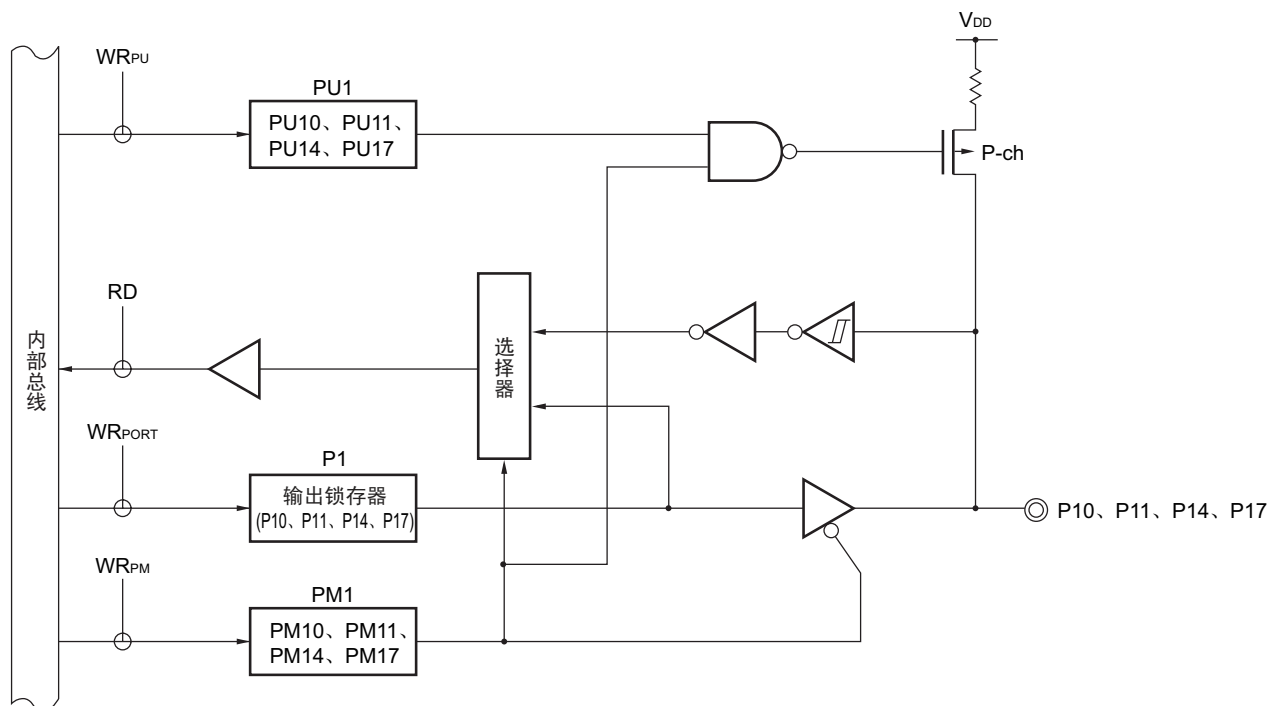
端口 1 的框图如图 4-1 ~ 图 4-4 所示。

图 4-1 P10 的框图（24 引脚产品）



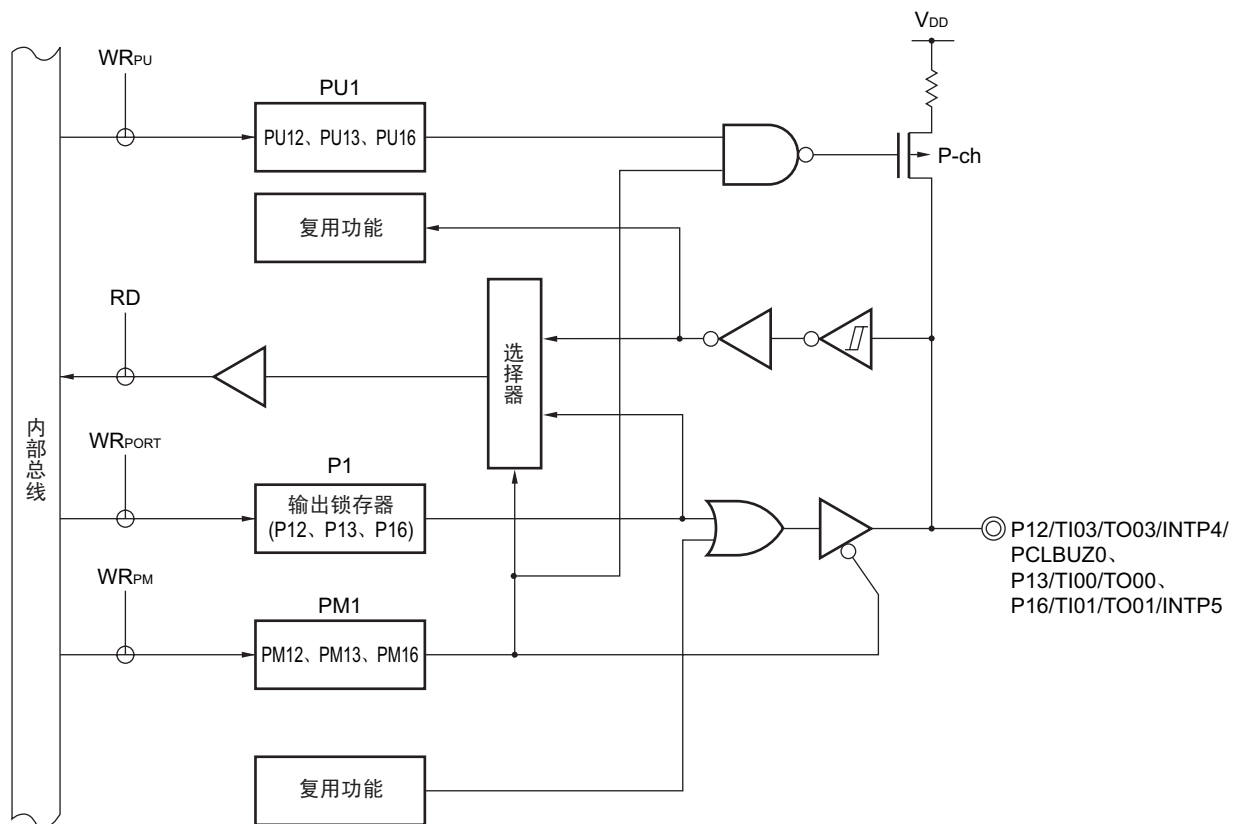
- P1 : 端口寄存器 1  
 PU1 : 上拉电阻选择寄存器 1  
 PM1 : 端口模式寄存器 1  
 PMC1 : 端口模式控制寄存器 1  
 RD : 读信号  
 WRxx : 写信号

图 4-2 P10、P11、P14 和 P17 的框图（32 引脚产品）



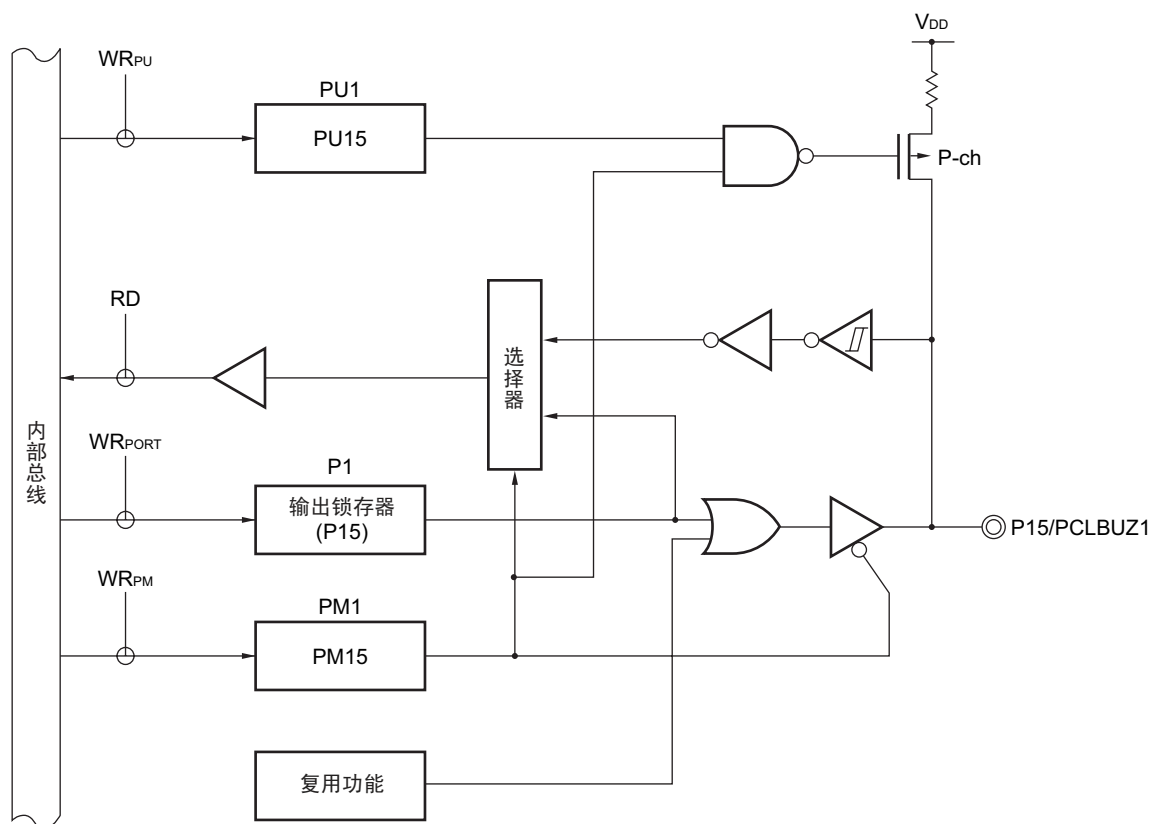
- P1 : 端口寄存器 1  
 PU1 : 上拉电阻选择寄存器 1  
 PM1 : 端口模式寄存器 1  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

图 4-3 P12、P13 和 P16 的框图



- P1 : 端口寄存器 1  
 PU1 : 上拉电阻选择寄存器 1  
 PM1 : 端口模式寄存器 1  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

图 4-4 P15 的框图



- P1 : 端口寄存器 1  
 PU1 : 上拉电阻选择寄存器 1  
 PM1 : 端口模式寄存器 1  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

### 4.2.2 端口 2

端口 2 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 2（PM2）以位为单位指定输入模式或者输出模式。

端口 2 作为复用功能，有 A/D 转换器的模拟输入、A/D 转换器的基准电位（+）输入、A/D 转换器的基准电位（-）输入和 D/A 转换器的输出。

当将 P20/ANI0、P21/ANI1、P22/ANI2/ANO0、P23/ANI3/ANO1、P24/ANI4 ~ P27/ANI7 用作数字输入或者数字输出时，必须按照以下顺序使用：

P27→P26→P25→P21→P20→P22→P23→P24

当将 P20/ANI0、P21/ANI1、P22/ANI2/ANO0、P23/ANI3/ANO1、P24/ANI4 ~ P27/ANI7 用作模拟输入 / 输出时，如果将 P20 和 P21 用作 A/D 转换器的基准电位（+）输入和 A/D 转换器的基准电位（-）输入，就必须从 P20 和 P21 的相邻引脚开始使用。否则，必须按照以下顺序使用：

P24→P23→P22→P20→P21→P25→P26→P27

表 4-3 使用端口 2 时的寄存器设定

引脚名		PM2n	ADPCn	复用功能的设定	备注
名称	输入 / 输出				
P2n	输入	1	1	—	当用作端口时，从高位开始使用。
	输出	0	1		

备注 1. PM2 : 端口模式寄存器 2

ADPC : A/D 端口配置寄存器

2. n=0 ~ 7

表 4-4 P20/ANI0、P21/ANI1、P24/ANI4 ~ P27/ANI7 引脚功能的设定

ADPC 寄存器	PM2 寄存器	ADS 寄存器	P20/ANI0、P21/ANI1、P24/ANI4 ~ P27/ANI7 引脚
数字输入 / 输出的选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入的选择	输入模式	选择 ANI。	模拟输入（转换对象）
		不选择 ANI。	模拟输入（非转换对象）
	输出模式	选择 ANI。	禁止设定。
		不选择 ANI。	

在产生复位信号后，P20/ANI0、P21/ANI1、P24/ANI4 ~ P27/ANI7 变为模拟输入。

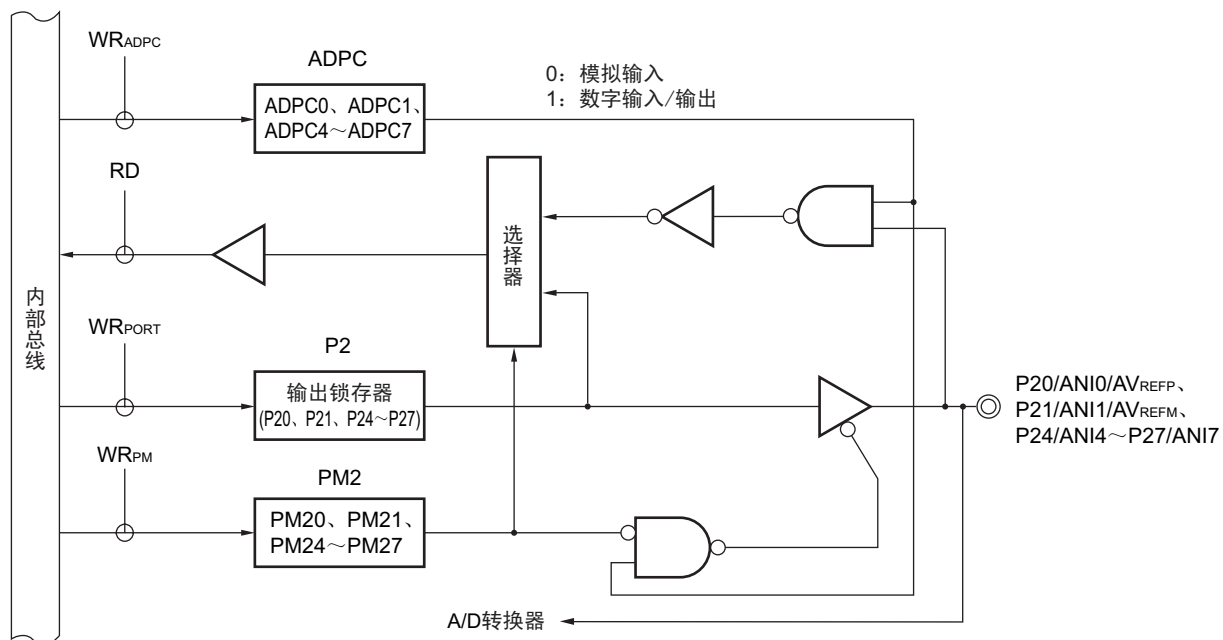
表 4-5 P22/ANI2/ANO0、P23/ANI3/ANO1 引脚功能的设定

ADPC 寄存器	PM2 寄存器	DAM 寄存器	ADS 寄存器	P22/ANI2/ANO0、 P23/ANI3/ANO1 引脚
数字输入 / 输出的选择	输入模式	—	—	数字输入
	输出模式	—	—	数字输出
模拟输入的选择	输入模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	模拟输出
		停止 D/A 转换。	选择 ANI。	模拟输入（转换对象）
			不选择 ANI。	模拟输入（非转换对象）
	输出模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	
		停止 D/A 转换。	选择 ANI。	
			不选择 ANI。	

在产生复位信号后，P22/ANI2/ANO0、P23/ANI3/ANO1 变为模拟输入 / 输出。

例如，32 引脚产品的端口 2 框图如图 4-5 和图 4-6 所示。

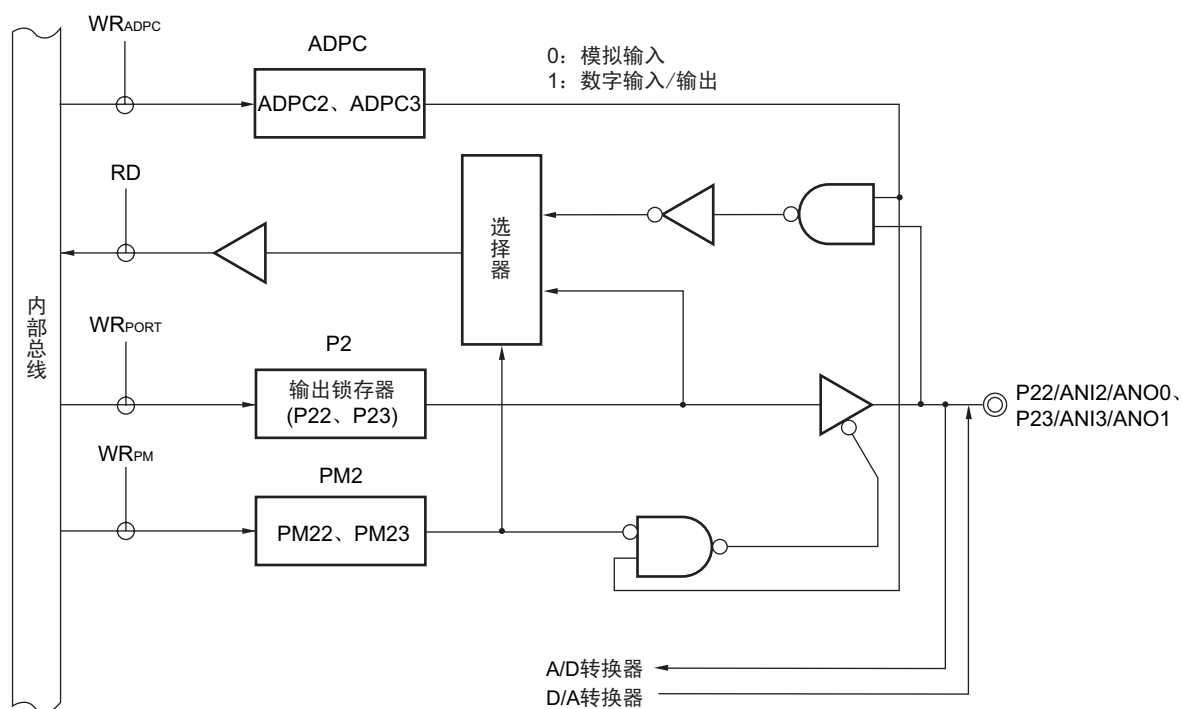
图 4-5 P20、P21 和 P24 ~ P27 的框图



ADPC : A/D 端口配置寄存器  
P2 : 端口寄存器 2  
PM2 : 端口模式寄存器 2  
RD : 读信号  
WR<sub>xx</sub> : 写信号



图 4-6 P22 和 P23 的框图



P2 : 端口寄存器 2  
 PM2 : 端口模式寄存器 2  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

### 4.2.3 端口 3

端口 3 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 3（PM3）以位为单位指定输入模式或者输出模式。当将 P30 ~ P35 引脚用作输入端口时，能通过上拉电阻选择寄存器 3（PU3）以位为单位使用内部上拉电阻。

端口 3 作为复用功能，有外部中断的请求输入、串行接口的数据输入 / 输出、时钟的输入 / 输出、编程 UART 的发送和接收、定时器的输入 / 输出和芯片的选择输入。

在产生复位信号后，P30 ~ P35 变为模拟输入。

表 4-6 使用端口 3 时的寄存器设定

名称	输入 / 输出	PM3x	复用功能的设定	备注
P30	输入	1	×	
	输出	0	SO00/TxD0 输出 =1 注 1	
P31	输入	1	×	
	输出	0	×	
P32	输入	1	×	
	输出	0	SCK00 输出 =1 注 1	
P33	输入	1	×	
	输出	0	TO02 输出 =0 注 2	
P34、P35	输入	1	×	
	输出	0	×	

注 1. 当将 P30/INTP2/TxD0/TOOLTxD/SO00、P32/INTP3/SCK00 用作通用端口时，必须将串行通道允许状态寄存器 0（SE0）的 bit0（SE00）、串行输出寄存器 0（SO0）的 bit0（SO00）和串行输出允许寄存器 0（SOE0）的 bit0（SOE00）置初始值。

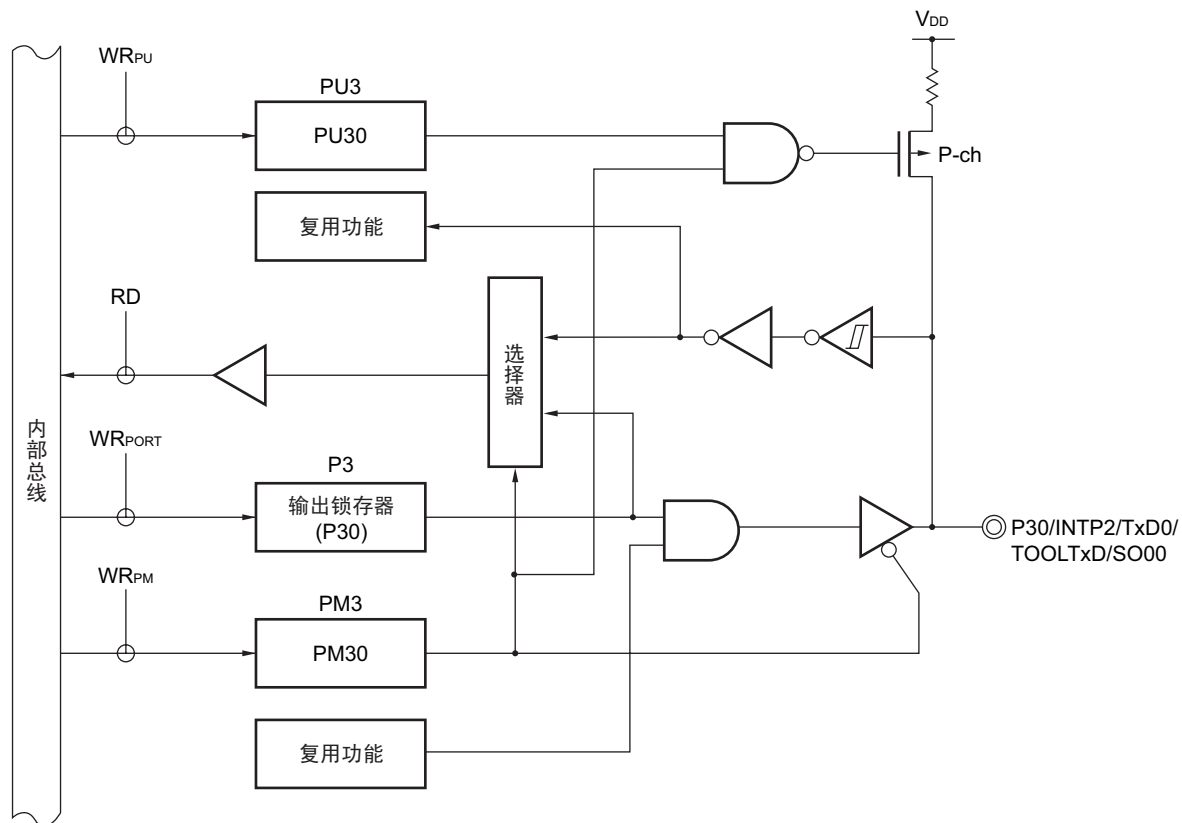
2. 当将 P33/TI02/TO02 用作通用端口时，必须将定时器输出寄存器 0（TO0）的 bit2（TO02）和定时器输出允许寄存器 0（TOE0）的 bit2（TOE02）置“0”（初始状态）。

备注 × : 忽略

PM3x : 端口模式寄存器 3

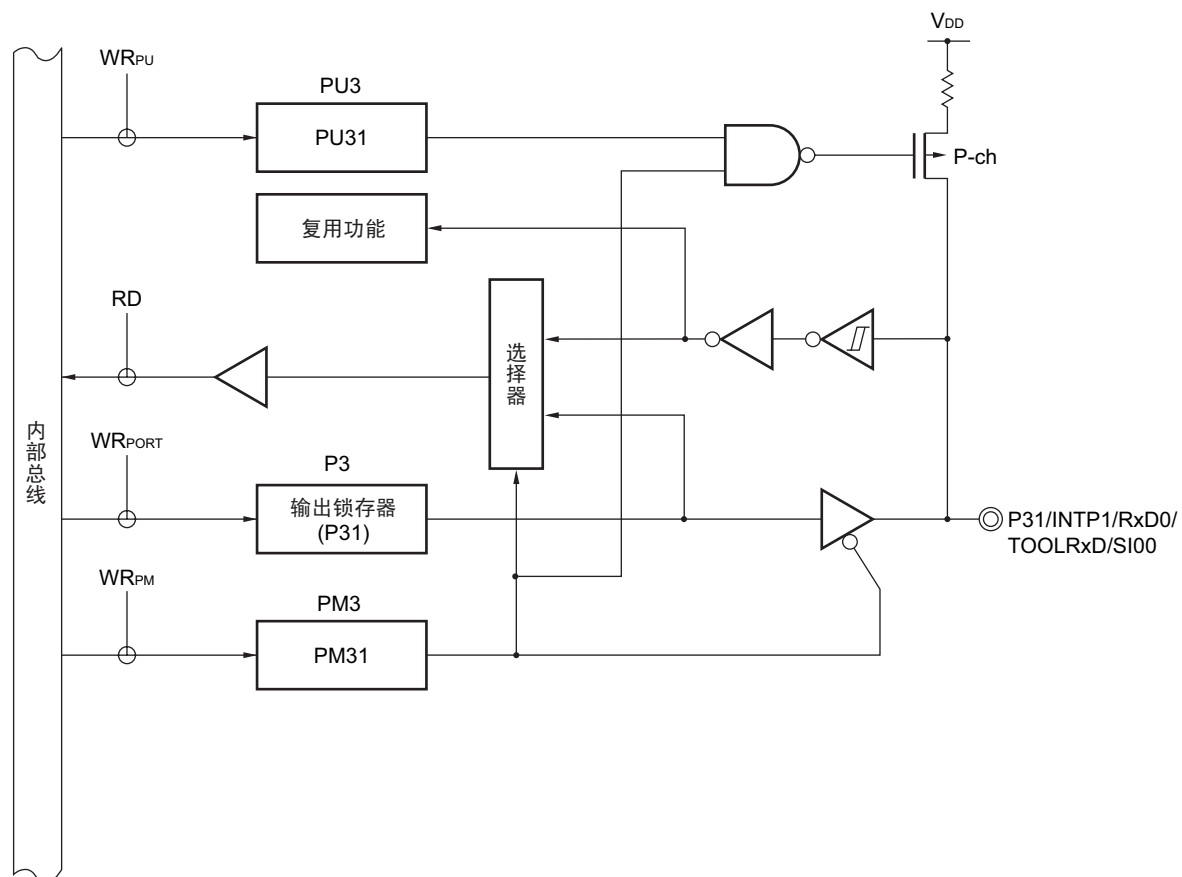
例如，32 引脚产品的端口 3 框图如图 4-7～图 4-11 所示。

图 4-7 P30 的框图



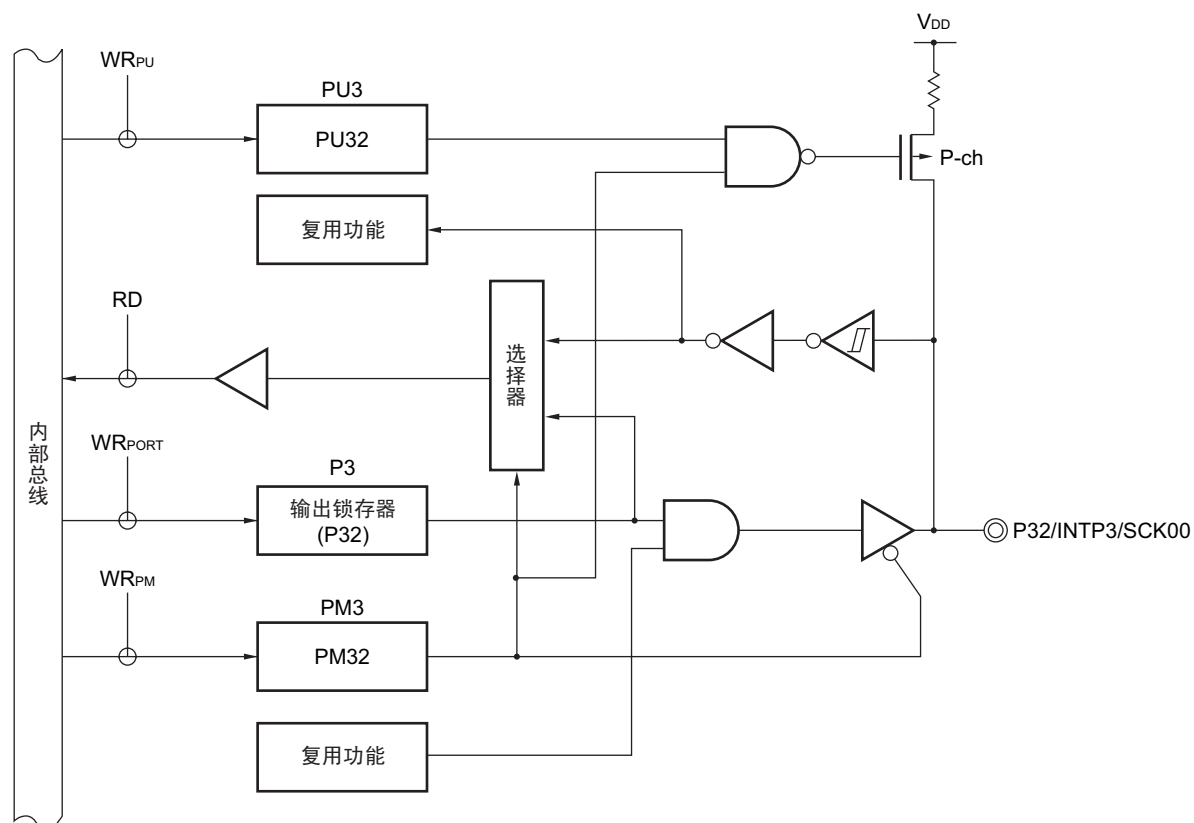
P3	: 端口寄存器 3
PU3	: 上拉电阻选择寄存器 3
PM3	: 端口模式寄存器 3
RD	: 读信号
WR <sub>xx</sub>	: 写信号

图 4-8 P31 的框图



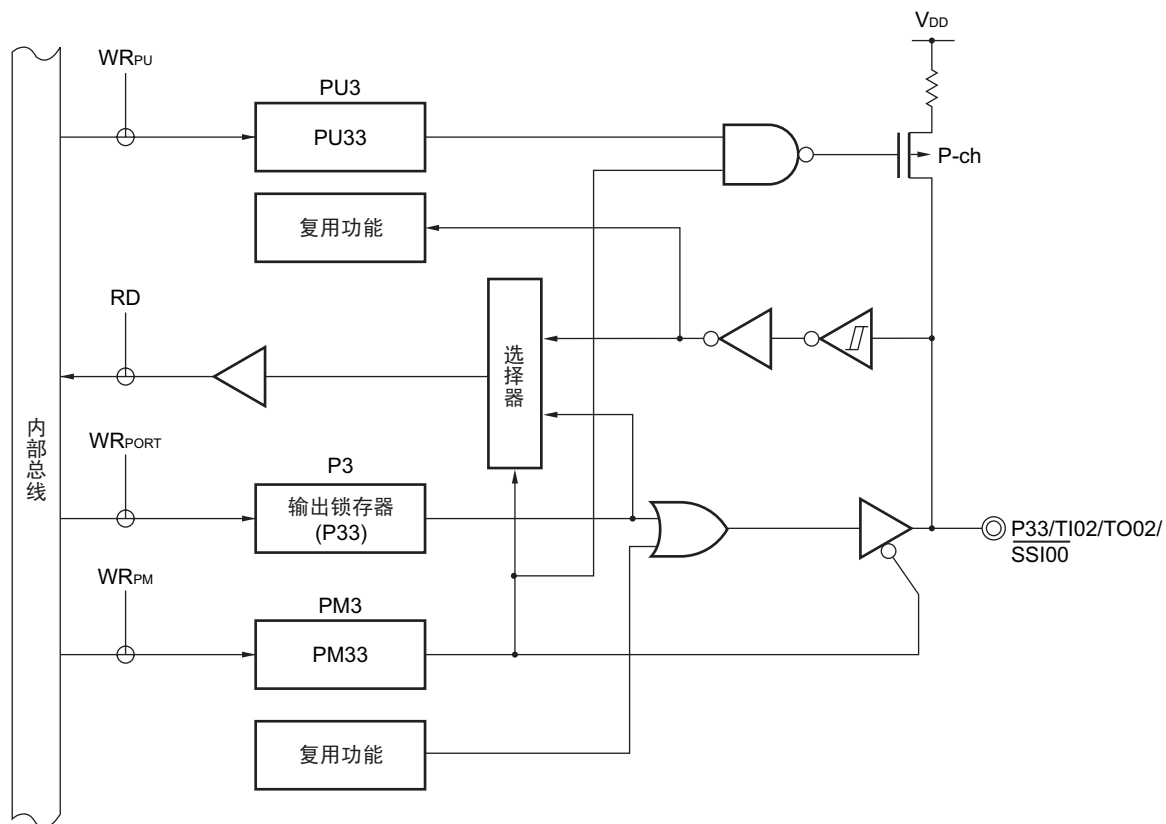
P3	: 端口寄存器 3
PU3	: 上拉电阻选择寄存器 3
PM3	: 端口模式寄存器 3
RD	: 读信号
WR <sub>xx</sub>	: 写信号

图 4-9 P32 的框图



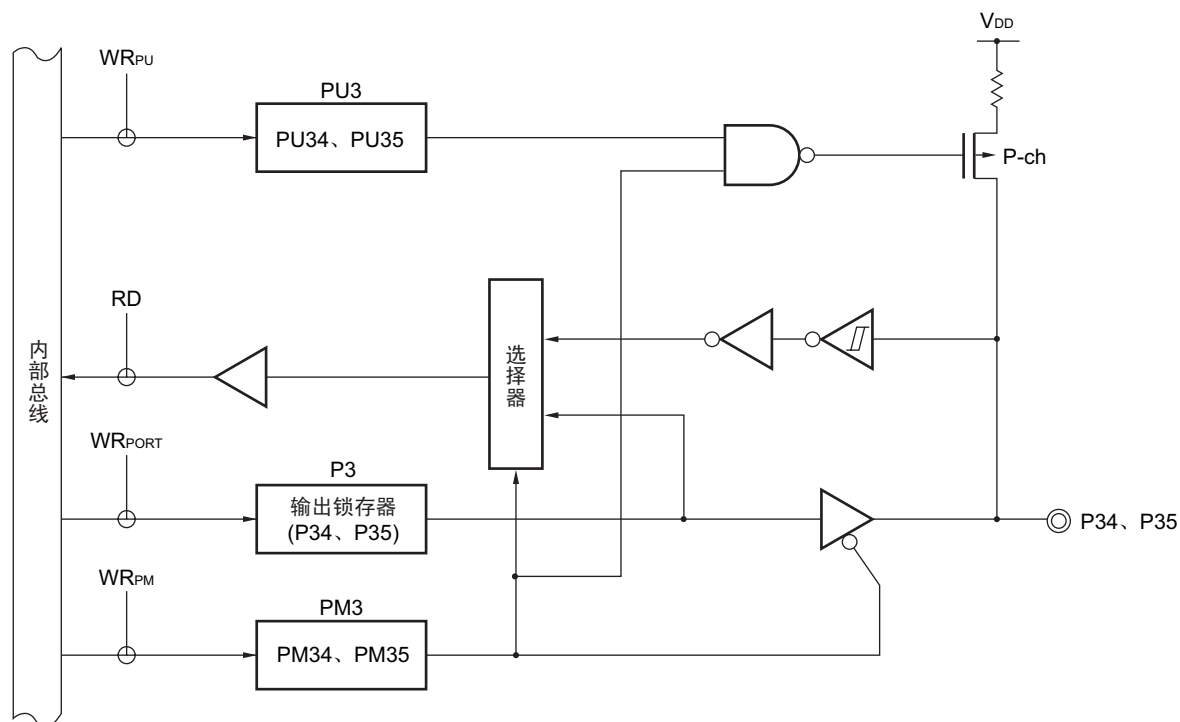
- P3 : 端口寄存器 3  
 PU3 : 上拉电阻选择寄存器 3  
 PM3 : 端口模式寄存器 3  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

图 4-10 P33 的框图



- P3 : 端口寄存器 3  
 PU3 : 上拉电阻选择寄存器 3  
 PM3 : 端口模式寄存器 3  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

图 4-11 P34 和 35 的框图



- P3 : 端口寄存器 3  
 PU3 : 上拉电阻选择寄存器 3  
 PM3 : 端口模式寄存器 3  
 RD : 读信号  
 WR<sub>xx</sub> : 写信号

#### 4.2.4 端口 4

端口 4 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 4（PM4）指定输入模式或者输出模式。当将 P40 引脚用作输入端口时，能通过上拉电阻选择寄存器 4（PU4）以位为单位使用内部上拉电阻。

端口 4 作为复用功能，有用于闪存编程器 / 调试器的数据输入 / 输出。

在产生复位信号后，端口 4 变为输入模式。

表 4-7 使用端口 4 时的寄存器设定

名称	输入 / 输出	PM4x	复用功能的设定	备注
P40	输入	1	×	
	输出	0	×	

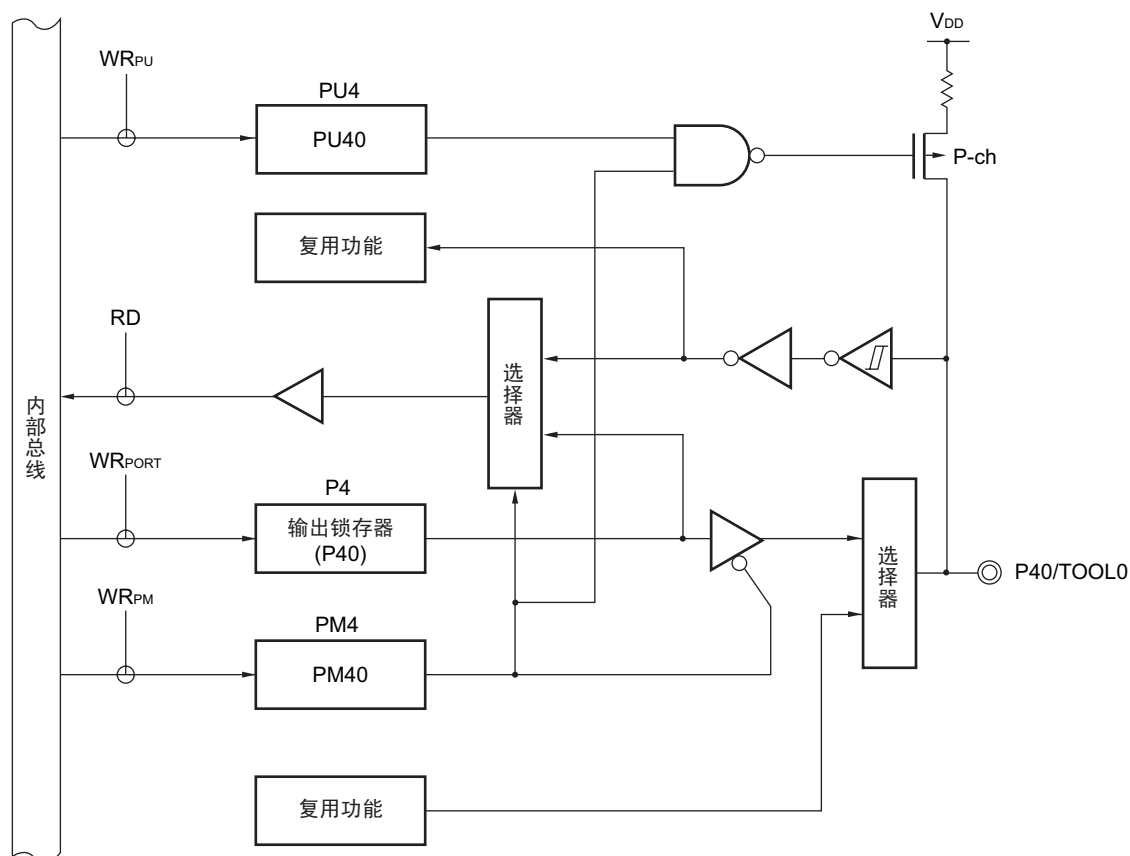
注意 在连接工具时，不能将 P40 引脚用作端口引脚。

备注 × : 忽略  
PM4x : 端口模式寄存器 4



端口 4 的框图如图 4-12 所示。

图 4-12 P40 的框图



- P4 : 端口寄存器 4  
 PU4 : 上拉电阻选择寄存器 4  
 PM4 : 端口模式寄存器 4  
 RD : 读信号  
 WRxx : 写信号

### 4.2.5 端口 6

端口 6 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 6（PM6）以位为单位指定输入模式或者输出模式。

P60 引脚和 P61 引脚的输出为 N 沟道漏极开路输出（6V 耐压）。

端口 6 作为复用功能，有串行接口的数据输入 / 输出和时钟输入 / 输出。

在产生复位信号后，端口 6 变为输入模式。

表 4-8 使用端口 6 时的寄存器设定

名称	输入 / 输出	PM6x	复用功能的设定	备注
P60	输入	1	SCLA0/SCLA1 输出 =1 注	
	输出	0		
P61	输入	1	SDAA0/SDAA1 输出 =1 注	
	输出	0		

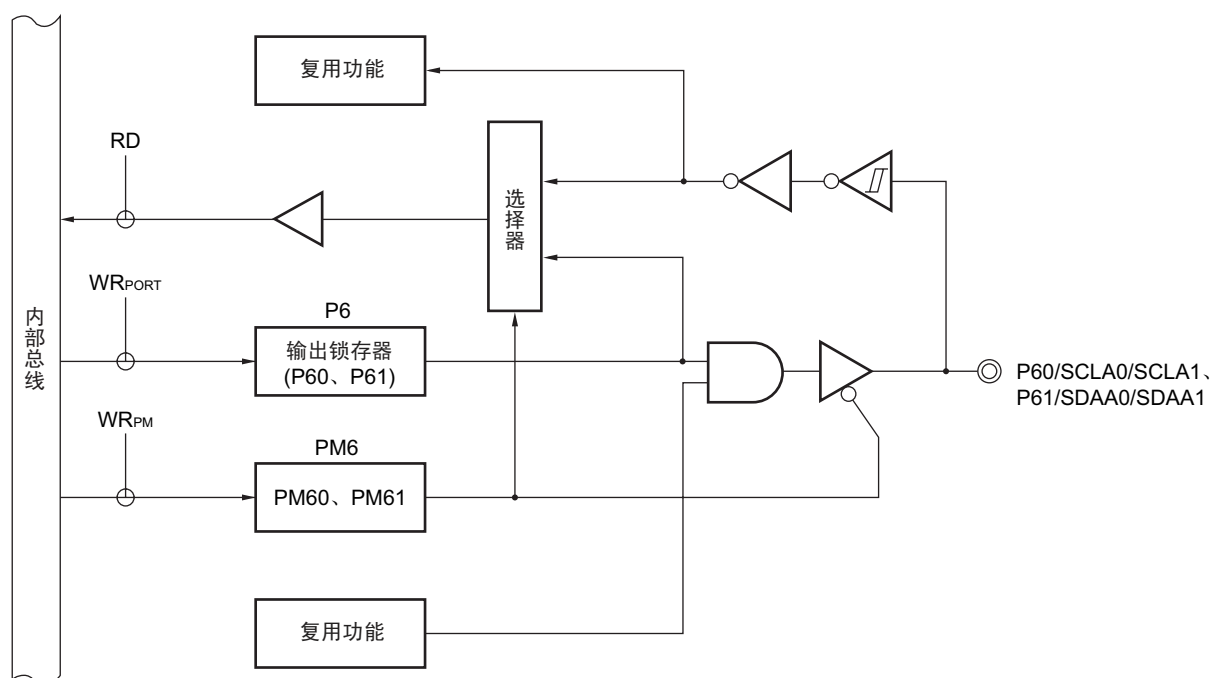
注 当将 P60/SCLA0/SCLA1、P61/SDAA0/SDAA1 用作通用端口时，必须停止对应的串行接口 IICA 的运行。

备注 × : 忽略

PM6x : 端口模式寄存器 6

端口 6 的框图如图 4-13 所示。

图 4-13 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 <sub>xx</sub>	: 写信号

4.2.6 端口 12

P121 和 P122 是 2 位输入专用端口。  
端口 12 作为复用功能，有主系统时钟的谐振器连接和主系统时钟的外部时钟输入。  
在产生复位信号后，端口 12 变为模拟输入。

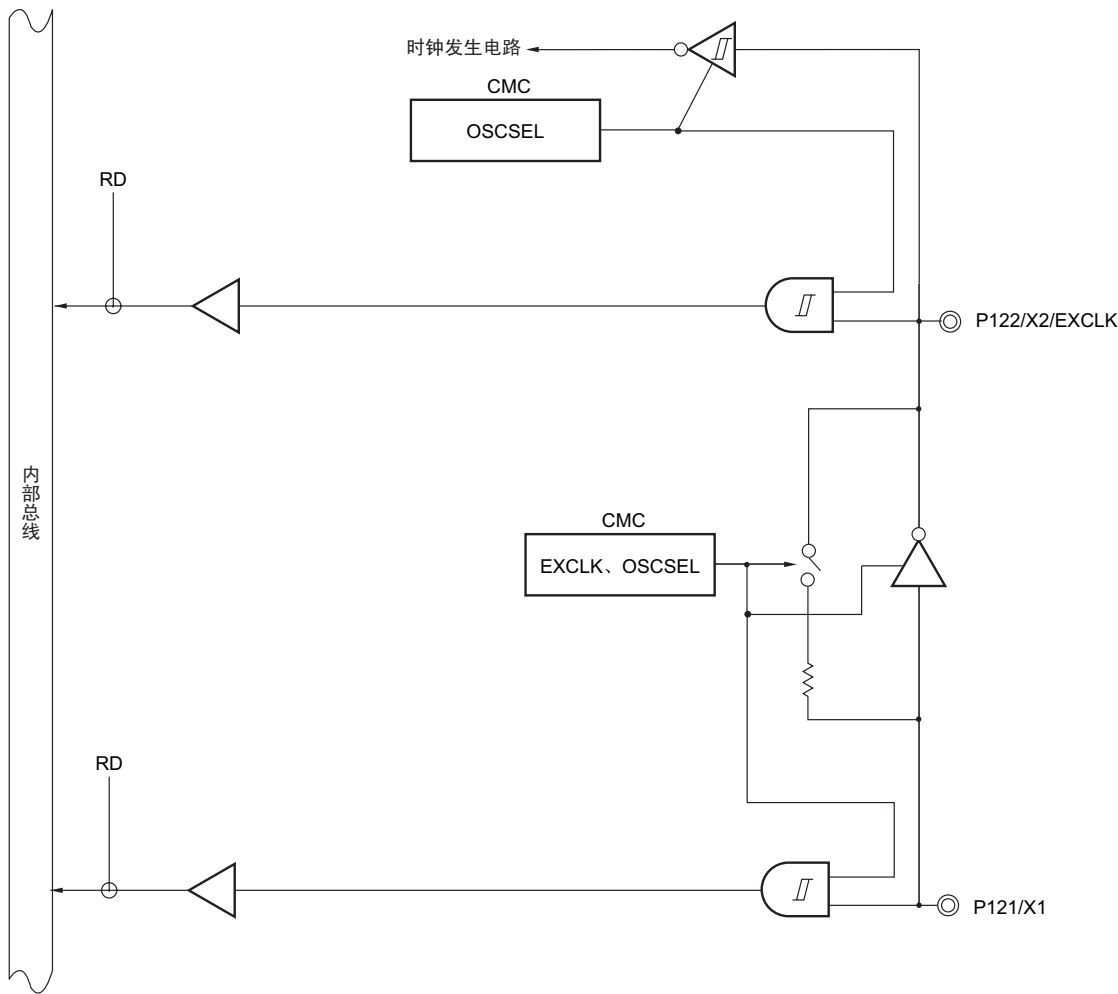
表 4-9 使用端口 12 时的寄存器设定

名称	输入 / 输出	复用功能的设定	备注
P121	输入	CMC 寄存器的 OSCSEL=0 或者 EXCLK=1	
P122	输入	CMC 寄存器的 OSCSEL=0	

注意 只能在解除复位后设定 1 次 P121 和 P122 的功能。设定为连接 X1 振荡或者外部时钟输入的端口在不复位的情况下不能用作输入端口。

端口 12 的框图如图 4-14 所示。

图 4-14 P121 和 P122 的框图



CMC : 时钟运行模式控制寄存器  
RD : 读信号

4.2.7 端口 13

P137 是 1 位输入专用端口，固定为输入模式。  
端口 13 作为复用功能，有外部中断的请求输入。

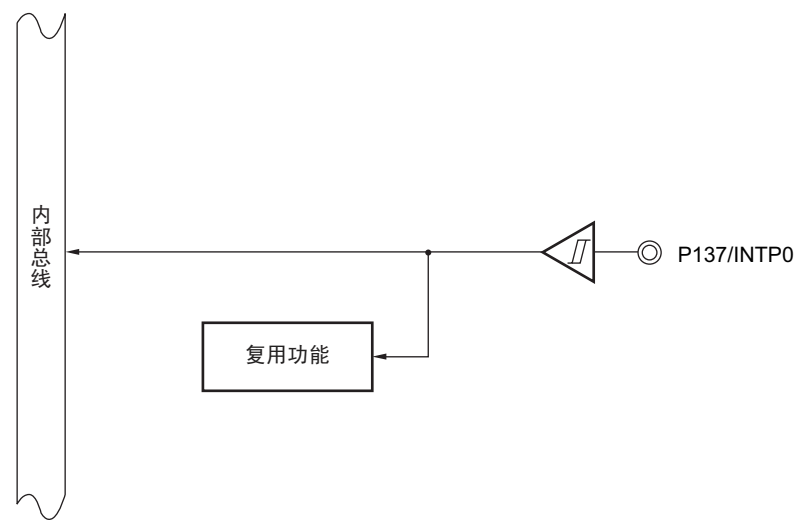
表 4-10 使用端口 13 时的寄存器设定

名称	输入 / 输出	复用功能的设定	备注
P137	输入	×	

备注    ×：忽略

端口 13 的框图如图 4-15 所示。

图 4-15 P137 的框图



### 4.3 控制端口功能的寄存器

通过以下寄存器控制端口。

- 端口模式寄存器 (PMxx)
- 端口寄存器 (Pxx)
- 上拉电阻选择寄存器 (PUxx)
- 端口模式控制寄存器 1 (PMC1)
- A/D 端口配置寄存器 (ADPC)

注意 分配的寄存器和位因产品而不同。有关各产品分配的寄存器和位，请参照表 4-11。必须给未分配的位设定初始值。

表 4-11 各产品分配的 PMxx、Pxx、PUxx、PMC1 寄存器及其位 (1/2)

端口		位名				32 引脚	24 引脚
		PMxx 寄存器	Pxx 寄存器	PUxx 寄存器	PMC1 寄存器注		
端口 1	0	PM10	P10	PU10	PMC10 注	○	○
	1	PM11	P11	PU11	—	○	—
	2	PM12	P12	PU12	—	○	○
	3	PM13	P13	PU13	—	○	○
	4	PM14	P14	PU14	—	○	—
	5	PM15	P15	PU15	—	○	○
	6	PM16	P16	PU16	—	○	○
	7	PM17	P17	PU17	—	○	—
端口 2	0	PM20	P20	—	—	○	○
	1	PM21	P21	—	—	○	○
	2	PM22	P22	—	—	○	○
	3	PM23	P23	—	—	○	○
	4	PM24	P24	—	—	○	—
	5	PM25	P25	—	—	○	—
	6	PM26	P26	—	—	○	—
	7	PM27	P27	—	—	○	○
端口 3	0	PM30	P30	PU30	—	○	○
	1	PM31	P31	PU31	—	○	○
	2	PM32	P32	PU32	—	○	○
	3	PM33	P33	PU33	—	○	○
	4	PM34	P34	PU34	—	○	—
	5	PM35	P35	PU35	—	○	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—

注 只限于 24 引脚产品。

表 4-11 各产品分配的 PMxx、Pxx、PUxx、PMC1 寄存器及其位 (2/2)

端口		位名				32 引脚	24 引脚
		PMxx 寄存器	Pxx 寄存器	PUxx 寄存器	PMC1 寄存器		
端口 4	0	PM40	P40	PU40	—	○	○
	1	—	—	—	—	—	—
	2	—	—	—	—	—	—
	3	—	—	—	—	—	—
	4	—	—	—	—	—	—
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
端口 6	0	PM60	P60	—	—	○	○
	1	PM61	P61	—	—	○	○
	2	—	—	—	—	—	—
	3	—	—	—	—	—	—
	4	—	—	—	—	—	—
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
端口 12	0	—	—	—	—	—	—
	1	—	P121	—	—	○	○
	2	—	P122	—	—	○	○
	3	—	—	—	—	—	—
	4	—	—	—	—	—	—
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	—	—	—	—	—
端口 13	0	—	—	—	—	—	—
	1	—	—	—	—	—	—
	2	—	—	—	—	—	—
	3	—	—	—	—	—	—
	4	—	—	—	—	—	—
	5	—	—	—	—	—	—
	6	—	—	—	—	—	—
	7	—	P137	—	—	○	○

4.3.1 端口模式寄存器（PMxx）

这是以位为单位设定端口输入 / 输出的寄存器。  
通过 1 位或者 8 位存储器操作指令设定端口模式寄存器。  
在产生复位信号后，这些寄存器的值变为“FFH”。  
当将端口引脚用作复用功能的引脚时，必须参照“4.5 使用复用功能时的端口相关寄存器的设定”进行设定。

图 4-16 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
PM1	PM17 注	PM16	PM15	PM14 注	PM13	PM12	PM11 注	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26 注	PM25 注	PM24 注	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	PM35 注	PM34 注	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W

PMmn	PMn 引脚的输入 / 输出模式的选择（m=1 ~ 4、6，n=0 ~ 7）
0	输出模式（输出缓冲器 ON）
1	输入模式（输出缓冲器 OFF）

注 24 引脚产品没有此位。

注意 必须将 PM3 寄存器的 bit6 和 bit7、PM4 寄存器的 bit1 ~ 7 以及 PM6 寄存器的 bit2 ~ 7 置“1”。



4.3.2 端口寄存器（Pxx）

这是设定端口输出锁存器的值的寄存器。

读时，在输入模式中读引脚电平，而在输出模式中读端口的输出锁存器的值注。

通过 1 位或者 8 位存储器操作指令设定端口寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

注 在将 P20 ～ P27 设定为 A/D 转换器的模拟输入功能的情况下，如果在输入模式中读端口，读取值就不是引脚电平而总是“0”。

图 4-17 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
P1	P17 注 3	P16	P15	P14 注 3	P13	P12	P11 注 3	P10	FFF01H	00H（输出锁存器）	R/W
P2	P27	P26 注 3	P25 注 3	P24 注 3	P23	P22	P21	P20	FFF02H	00H（输出锁存器）	R/W
P3	0	0	P35 注 3	P34 注 3	P33	P32	P31	P30	FFF03H	00H（输出锁存器）	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H（输出锁存器）	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H（输出锁存器）	R/W
P12	0	0	0	0	0	P122	P121	0	FFF0CH	不定值	R/W注1
P13	P137	0	0	0	0	0	0	0	FFF0DH	注 2	R/W注1

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

注 1. P121、P122、P137 是只读位。

2. P137：不定值

3. 24 引脚产品没有此位。

备注 m=1 ～ 4、6、12、13，n=0 ～ 7

4.3.3 上拉电阻选择寄存器（PUxx）

这是设定是否使用内部上拉电阻的寄存器。只能对通过上拉电阻选择寄存器指定使用内部上拉电阻的引脚并且设定为输入模式（PMmn=1）的位，以位为单位使用内部上拉电阻。对于设定为输出模式的位，与上拉电阻选择寄存器的设定无关，不连接内部上拉电阻。当用作复用功能的输出引脚或者设定为模拟输入（PMC=1）时也相同。

通过 1 位或者 8 位存储器操作指令设定上拉电阻选择寄存器。  
在产生复位信号后，这些寄存器的值变为“00H”（只有 PU4 为“01H”）。

图 4-18 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU1	PU17 注	PU16	PU15	PU14 注	PU13	PU12	PU11 注	PU10	F0031H	00H	R/W
PU3	0	0	PU35 注	PU34 注	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W

PUmn	Pmn 引脚的内部上拉电阻的选择（m=0、1、4，n=0～7）
0	不连接内部上拉电阻。
1	连接内部上拉电阻。

注 24 引脚产品没有此位。

#### 4.3.4 端口模式控制寄存器 1（PMC1）（只限于 24 引脚产品）

这是以位为单位设定数字输入 / 输出或者模拟输入的寄存器。

通过 1 位或者 8 位存储器操作指令设定端口模式控制寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

图 4-19 端口模式控制寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC1	1	1	1	1	1	1	1	PMC10	F0061H	FFH	R/W

PMC10	P10 引脚的数字输入 / 输出或者模拟输入的选择
0	数字输入 / 输出（模拟输入以外的复用功能）
1	模拟输入

注意 1. 必须通过端口模式寄存器 1（PM1）将 A/D 转换使用的通道设定为输入模式。

2. 对于由 PMC 寄存器设定为数字输入 / 输出的引脚，不能通过模拟输入通道指定寄存器（ADS）进行设定。

3. 必须给未分配的位设定初始值。

#### 4.3.5 A/D 端口配置寄存器（ADPC）

这是将 ANI0/P20 ~ ANI7/P27 引脚切换为 A/D 转换器的模拟输入或者端口的数字输入 / 输出的寄存器。

通过 8 位存储器操作指令设定 ADPC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-20 A/D 端口配置寄存器（ADPC）的格式

地址: F0076H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
ADPC	ADPC7	ADPC6 注	ADPC5 注	ADPC4 注	ADPC3	ADPC2	ADPC1	ADPC0	

ADPCn	P2n/ANIn 的模拟输入（A）和数字输入 / 输出（D）的选择
0	模拟输入（A）（默认值）
1	数字输入 / 输出（D）

注 24 引脚产品没有此位。

注意 1. 对于由 ADPC 寄存器设定为模拟功能的端口，必须通过端口模式寄存器 2（PM2）将其设定为输入模式。

2. 对于由 ADPC 寄存器设定为数字输入 / 输出的引脚，不能通过模拟输入通道指定寄存器（ADS）进行设定。

3. 当使用  $AV_{REFP}$  和  $AV_{REFM}$  时，必须将 ANI0 和 ANI1 设定为模拟输入并且将端口模式寄存器设定为输入模式。

## 4.4 端口功能的运行

如下所示，端口的运行因输入 / 输出模式的设定而不同。

### 4.4.1 输入 / 输出端口的写操作

#### (1) 输出模式的情况

能通过传送指令给输出锁存器写值，并且输出锁存器的内容从引脚输出。

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

#### (2) 输入模式的情况

通过传送指令，给输出锁存器写值，但是因输出缓冲器处于 OFF 状态而引脚的状态不发生变化。因此，能对同时具有输入和输出功能的端口进行字节写。

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

### 4.4.2 输入 / 输出端口的读操作

#### (1) 输出模式的情况

能通过传送指令读输出锁存器的内容，但是输出锁存器的内容不发生变化。

#### (2) 输入模式的情况

能通过传送指令读引脚状态，但是输出锁存器的内容不发生变化。

### 4.4.3 输入 / 输出端口的运算

#### (1) 输出模式的情况

对输出锁存器的内容进行运算，将结果写到输出锁存器，并且输出锁存器的内容从引脚输出。

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

#### (2) 输入模式的情况

读引脚电平并且对其内容进行运算，将运算结果写到输出锁存器，但是因输出缓冲器处于 OFF 状态而引脚的状态不发生变化。因此，能对同时具有输入和输出功能的端口进行字节写。

在产生复位信号时清除输出锁存器的数据。

## 4.5 使用复用功能时的端口相关寄存器的设定

当将端口引脚用作复用功能的引脚时，必须对端口相关寄存器进行如表 4-12 所示的设定。

**注意** 如果给使用的引脚分配了其他复用输出功能，就需要将不使用的复用功能的输出置初始值。有关具体的对象和处理方法，请参照 4.6.2。

表 4-12 使用复用功能时的端口相关寄存器的设定 (1/2)

引脚名	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P12	TI03	输入	1	×
	TO03	输出	0	0
	INTP4	输入	1	×
	PCLBUZ0	输出	0	0
P13	TI00	输入	1	×
	TO00	输出	0	0
P15	PCLBUZ1	输出	0	0
P16	TI01	输入	1	×
	TO01	输出	0	0
	INTP5	输入	1	×
P20 注	ANI0 注	输入	1	×
	AV <sub>REFP</sub> 注	输入	1	×
P21 注	ANI1 注	输入	1	×
	AV <sub>REFM</sub> 注	输入	1	×
P22 注	ANI2 注	输入	1	×
	ANO0	输出	1	×
P23 注	ANI3 注	输入	1	×
	ANO1	输出	1	×
P24 ~ P27 注	ANI4 ~ ANI7 注	输入	1	×

备注 1. × : 忽略

PMxx : 端口模式寄存器

Pxx : 端口输出锁存器

2. 各引脚和其复用功能的关系是 32 引脚产品的情况。当其他产品使用复用功能时，可能分配给不同的引脚，但是 PMxx 和 Pxx 的设定相同。

(注在下页)

表 4-12 使用复用功能时的端口相关寄存器的设定 (2/2)

引脚名	复用功能		PMxx	Pxx
	名称	输入 / 输出		
P30	INTP2	输入	1	×
	TxD0	输出	0	1
	TOOLTxD	输出	0	1
	SO00	输出	0	1
P31	INTP1	输入	1	×
	RxD0	输入	1	×
	TOOLRxD	输入	1	×
	SI00	输入	1	×
P32	INTP3	输入	1	×
	SCK00	输入	1	×
		输出	0	1
P33	TI02	输入	1	×
	TO02	输出	0	0
	SSI00	输入	1	×
P40	TOOL0	输入 / 输出	×	×
P60	SCLA0	输入 / 输出	0	1
	SCLA1	输入 / 输出	0	1
P61	SDAA0	输入 / 输出	0	1
	SDAA1	输入 / 输出	0	1
P137	INTP0	输入	—	×

备注 1. × : 忽略

PMxx : 端口模式寄存器

Pxx : 端口输出锁存器

2. 各引脚和其复用功能的关系是 32 引脚产品的情况。当其他产品使用复用功能时, 可能分配给不同的引脚, 但是 PMxx 和 Pxx 的设定相同。

注 ANI0/P20 ~ ANI7/P27 引脚功能取决于 A/D 端口配置寄存器 (ADPC)、模拟输入通道指定寄存器 (ADS) 以及端口模式寄存器 2 (PM2) 的设定。

表 4-13 ANI0/P20、ANI1/P21、ANI4/P24 ~ ANI7/P27 引脚的功能设定

ADPC 寄存器	PM2 寄存器	ADS 寄存器	ANI0/P20、ANI1/P21、ANI4/P24 ~ ANI7/P27 引脚
数字输入 / 输出的选择	输入模式	×	数字输入
	输出模式	×	数字输出
模拟输入的选择	输入模式	选择 ANI。	模拟输入 (转换对象)
		不选择 ANI。	模拟输入 (非转换对象)
	输出模式	选择 ANI。	禁止设定。
		不选择 ANI。	

备注 ×: 忽略

表 4-14 P22/ANI2/ANO0 引脚和 P23/ANI3/ANO1 引脚的功能设定

ADPC 寄存器	PM2 寄存器	DAM 寄存器	ADS 寄存器	P22/ANI2/ANO0 引脚和 P23/ANI3/ANO1 引脚
数字输入 / 输出的选择	输入模式	—	—	数字输入
	输出模式	—	—	数字输出
模拟输入的选择	输入模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	模拟输出
		停止 D/A 转换。	选择 ANI。	模拟输入（转换对象）
			不选择 ANI。	模拟输入（非转换对象）
	输出模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	
		停止 D/A 转换。	选择 ANI。	
			不选择 ANI。	

24 引脚产品的 ANI16/P10 引脚功能取决于端口模式控制寄存器 1（PMC1）、模拟输入通道指定寄存器（ADS）以及 PM1 寄存器的设定。

## 4.6 使用端口功能时的注意事项

### 4.6.1 有关对端口寄存器 n (Pn) 的位存储器操作指令的注意事项

在对同时具有输入和输出功能的端口执行位存储器操作指令时，除了操作对象的位以外，也可能改写非操作对象的输入端口的输出锁存器的值。

因此，建议在将任意的端口从输入模式切换为输出模式前重新写输出锁存器的值。

<例> 当 P10 为输出端口并且 P11 ~ P17 为输入端口（引脚状态全部为高电平）而且端口 1 的输出锁存器值为“00H”时，如果通过位存储器操作指令将输出端口 P10 的输出从低电平改为高电平，端口 1 的输出锁存器的值就变为“FFH”。

说明：PMnm 位为“1”的端口的 Pn 寄存器的写对象是输出锁存器，而读对象是引脚状态。

在 R7F0C010 内部，位存储器操作指令按照以下顺序执行：

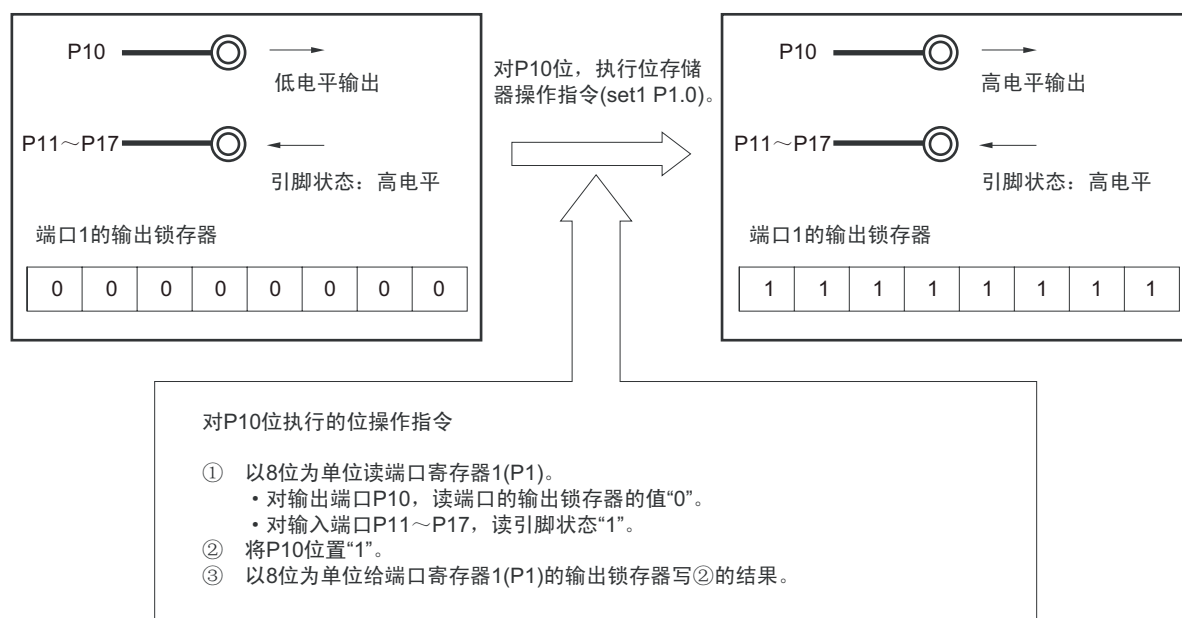
- <1> 以 8 位为单位读 Pn 寄存器。
- <2> 操作 1 位对象位。
- <3> 以 8 位为单位写 Pn 寄存器。

在 <1> 时，读输出端口 P10 的输出锁存器的值“0”，并且读输入端口 P11 ~ P17 的引脚状态。此时，如果 P11 ~ P17 的引脚状态为高电平，读取值就为“FEH”。

通过 <2> 的操作，值变为“FFH”。

通过 <3> 的操作，给输出锁存器写“FFH”。

图 4-21 位存储器操作指令（P10 的情况）





### 4.6.2 设定引脚时的注意事项

如果给使用的输出引脚分配了其他复用输出功能，就需要将不使用的复用功能的输出置初始值（回避输出的冲突）。有关复用输出，请参照“4.5 使用复用功能时的端口相关寄存器的设定”。

另外，对于用作输入的引脚，复用输出功能无效（缓冲器输出为 Hi-Z），因此不需要进行处理。

具体的对象和处理方法如下所示。

表 4-15 不使用的复用功能的处理

对象单元	不使用的复用功能的输出或者输入 / 输出引脚	不使用的复用功能的处理
定时器阵列单元	TO0n	将定时器输出寄存器 0（TO0）的 bit n（TO0n）和定时器输出允许寄存器 0（TOE0）的 bit n（TOE0n）置“0”（初始状态）。
时钟 / 蜂鸣器输出	PCLBUZn	将时钟输出选择寄存器 n（CKSn）的 bit7（PCLOEn）置“0”（初始状态）。
串行阵列单元	SCK00、SCL00、TxD0	将串行通道允许状态寄存器 0（SE0）的 bit0（SE00）、串行输出寄存器 0（SO0）的 bit0（SO00）以及串行输出允许寄存器 0（SOE0）的 bit0（SOE00）置初始值（SO00 为“1”，其他为“0”）。
IICA	SCLAn、SDAAn	将 IICA 设定为运行停止（IICCTLn0 寄存器的 bit7（IICEn）置“0”）。

例 P12/TI03/TO03/INTP4/PCLBUZ0 引脚的情况

#### (1) 用作 PCLBUZ0 输出

P12 : 通过将端口模式寄存器 1 的 PM12 位置“0”，设定为输出。

TI03、INTP4 : 因为是输入，所以不为处理对象。

TO03 : 因为是输出，所以设定为定时器阵列单元 0 的 TO03=TOE03=0。

#### (2) 用作 TO03 输出

P12 : 通过将端口模式寄存器 1 的 PM12 位置“0”，设定为输出。

PCLBUZ0 : 因为是输出，所以设定为时钟 / 蜂鸣器输出的 PCLOE0=0。

TI03 : 因为是输入，所以不为处理对象。

和 P30/INTP2/TxD0/TOOLTxD/SO00 引脚用作 SO00 输出时的 TxD0 相同，无法通过切换模式使相同串行通道内的复用功能有效，因此不为处理对象（如果设定为 CSI（MD002=MD001=0），就不作为 UART 运行，因此 TxD0 输出无效）。

为了省电，建议停止不使用的功能，包括只有输入或者没有输入 / 输出的模块。

## 第 5 章 时钟发生电路

### 5.1 时钟发生电路的功能

时钟发生电路是产生给 CPU 和外围硬件提供时钟的电路。

有以下 2 种系统时钟和时钟振荡电路。

#### (1) 主系统时钟

##### ① X1 振荡电路

能通过给 X1 和 X2 连接谐振器使  $f_X=1\sim 20\text{MHz}$  的时钟振荡，并且能通过执行 STOP 指令或者设定 MSTOP 位（时钟运行状态控制寄存器（CSC）的 bit7）使振荡停止。

##### ② 高速内部振荡器

能通过选项字节（000C2H）从  $f_{IH}=32\text{MHz}$ 、24MHz、16MHz、12MHz、8MHz、4MHz 和 1MHz(TYP.) 中选择频率进行振荡。在解除复位后，CPU 一定以此高速内部振荡器时钟开始运行。能通过执行 STOP 指令或者设定 HIOSTOP 位（CSC 寄存器的 bit0）使振荡停止。

另外，能由 EXCLK/X2/P122 引脚提供外部主系统时钟（ $f_{EX}=1\sim 20\text{MHz}$ ），并且能通过执行 STOP 指令或者设定 MSTOP 位将外部主系统时钟的输入置为无效。

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

能通过高速内部振荡器的频率选择寄存器（HOCODIV）更改选项字节设定的频率。有关频率设定，请参照“图 5-9 高速内部振荡器的频率选择寄存器（HOCODIV）的格式”。

能通过高速内部振荡器设定的振荡频率如下所示（能通过选项字节和高速内部振荡器的频率选择寄存器（HOCODIV）选择的种类）。

电源电压	振荡频率（MHz）						
	1	4	8	12	16	24	32
$2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$	○	○	○	○	○	○	○

#### (2) 低速内部振荡器时钟（低速内部振荡器）

能使  $f_{IL}=15\text{kHz}$ (TYP.) 的时钟振荡。

不能将低速内部振荡器时钟用作 CPU 时钟。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 看门狗定时器

当选项字节（000C0H）的 bit4（WDTON）为“1”时，低速内部振荡器振荡。

但是，在 WDTON 位为“1”并且选项字节（000C0H）的 bit0（WDSTBYON）为“0”时，如果执行 HALT 指令或者 STOP 指令，低速内部振荡器就停止振荡。

备注  $f_X$  : X1 时钟振荡频率  
 $f_{IH}$  : 高速内部振荡器的时钟频率  
 $f_{EX}$  : 外部主系统时钟频率  
 $f_{IL}$  : 低速内部振荡器的时钟频率

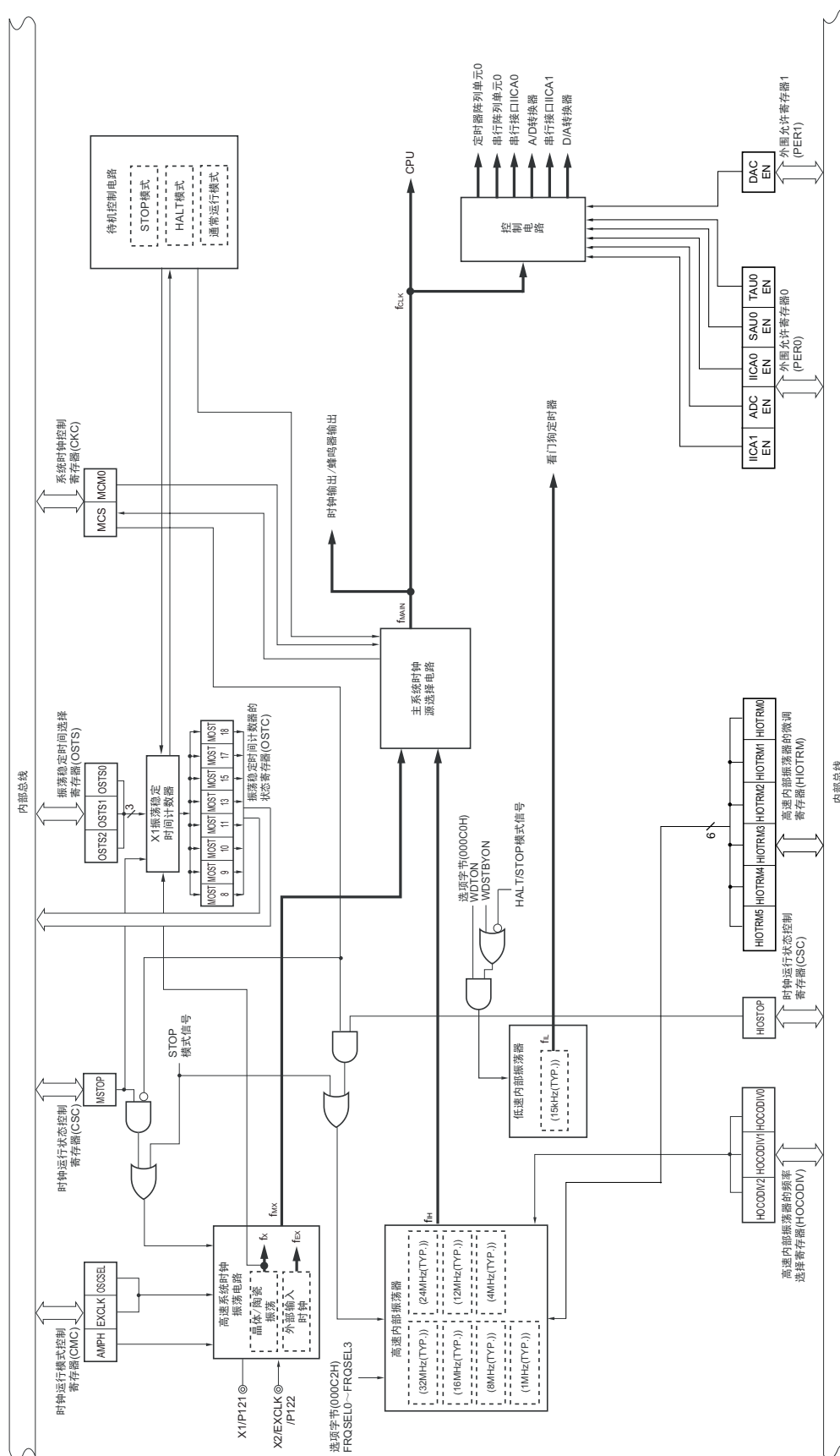
## 5.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表 5-1 时钟发生电路的结构

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

图 5-1 时钟发生电路的框图



备注	$f_X$	: X1 时钟振荡频率
	$f_{IH}$	: 高速内部振荡器的时钟频率
	$f_{EX}$	: 外部主系统时钟频率
	$f_{MX}$	: 高速系统时钟频率
	$f_{MAIN}$	: 主系统时钟频率
	$f_{CLK}$	: CPU/外围硬件的时钟频率
	$f_{IL}$	: 低速内部振荡器的时钟频率

### 5.3 控制时钟发生电路的寄存器

通过以下 9 种寄存器控制时钟发生电路。

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

注意 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

5.3.1 时钟运行模式控制寄存器（CMC）

这是设定 X1/P121 引脚和 X2/EXCLK/P122 引脚的运行模式以及选择振荡电路增益的寄存器。  
在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。能通过 8 位存储器操作指令读此寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 5-2 时钟运行模式控制寄存器（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	0	0	0	0	AMPH

EXCLK	OSCSEL	高速系统时钟 引脚的运行模式	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	连接晶体谐振器或者陶瓷谐振器。	
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

AMPH	X1 时钟振荡频率的控制
0	1MHz ≤ f <sub>X</sub> ≤ 10MHz
1	10MHz < f <sub>X</sub> ≤ 20MHz

- 注意 1. 在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。当以初始值（“00H”）使用 CMC 寄存器时，为了防止程序失控时的误动作（如果误写“00H”以外的值就不能恢复），必须在解除复位后将 CMC 寄存器置“00H”。
2. 在解除复位后并且在通过设定时钟运行状态控制寄存器（CSC）开始 X1 振荡前，必须设定 CMC 寄存器。
3. 必须在解除复位后并且在选择 f<sub>IH</sub> 作为 f<sub>CLK</sub> 的状态（将 f<sub>CLK</sub> 切换为 f<sub>MX</sub> 前的状态）下设定 AMPH 位。
4. 系统时钟的频率上限是 32MHz，但是 X1 振荡电路的频率上限为 20MHz。

备注 f<sub>X</sub>: X1 时钟振荡频率

5.3.2 系统时钟控制寄存器（CKC）

这是选择 CPU/ 外围硬件时钟和主系统时钟的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 CKC 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 5-3 系统时钟控制寄存器（CKC）的格式

地址: FFFA4H

复位后: 00H

R/W 注

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

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

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

注 bit5 是只读位。

备注 f<sub>IH</sub> : 高速内部振荡器的时钟频率  
f<sub>MX</sub> : 高速系统时钟频率  
f<sub>MAIN</sub> : 主系统时钟频率

注意 必须将 bit0 ~ 3、6、7 置“0”。

### 5.3.3 时钟运行状态控制寄存器（CSC）

这是控制高速系统时钟和高速内部振荡器时钟运行的寄存器（低速内部振荡器时钟除外）。

通过 1 位或者 8 位存储器操作指令设定 CSC 寄存器。

在产生复位信号后，此寄存器的值变为“C0H”。

图 5-4 时钟运行状态控制寄存器（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	0	0	0	0	0	0	HIOSTOP

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

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

注意 1. 在解除复位后，必须在设定时钟运行模式控制寄存器（CMC）后设定 CSC 寄存器。

2. 在解除复位后并且在将 MSTOP 位置“0”前，必须设定振荡稳定时间选择寄存器（OSTS）。但是，当以初始值使用 OSTS 寄存器时，不需要设定 OSTS 寄存器。
3. 要通过设定 MSTOP 位开始 X1 振荡时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认 X1 时钟的振荡稳定时间。
4. 不能通过 CSC 寄存器停止被选择为 CPU/ 外围硬件时钟（ $f_{CLK}$ ）的时钟。
5. 有关用于停止时钟振荡（外部时钟输入无效）的寄存器标志设定和停止前的条件，请参照表 5-2。

表 5-2 时钟停止方法

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



### 5.3.4 振荡稳定时间计数器的状态寄存器（OSTC）

这是表示 X1 时钟的振荡稳定时间计数器计数状态的寄存器。

能在以下情况下确认 X1 时钟的振荡稳定时间：

- 当 CPU 时钟为高速内部振荡器时钟并且开始 X1 时钟的振荡时
- 当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 STOP 模式后解除 STOP 模式时

能通过 1 位或者 8 位存储器操作指令读 OSTC 寄存器。

通过复位信号的产生、STOP 指令或者 MSTOP 位（时钟运行状态控制寄存器（CSC）的 bit7）为“1”，此寄存器的值变为“00H”。

备注 在以下情况下，振荡稳定时间计数器开始计数：

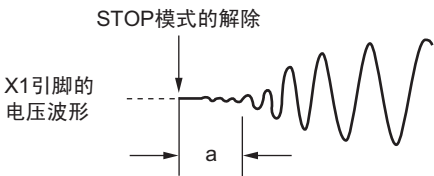
- 当 X1 时钟开始振荡（EXCLK=0→MSTOP=0）时
- 当解除 STOP 模式时

图 5-5 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	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18		

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

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



备注  $f_X$ : X1 时钟振荡频率

5.3.5 振荡稳定时间选择寄存器（OSTS）

这是选择 X1 时钟的振荡稳定时间的寄存器。

如果选择 X1 时钟，就在 X1 振荡电路运行（MSTOP=0）后自动等待 OSTS 寄存器设定的时间。

如果将 CPU 时钟从高速内部振荡器时钟切换到 X1 时钟，或者 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 STOP 模式后解除 STOP 模式，就必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认是否经过振荡稳定时间。能通过 OSTC 寄存器确认 OSTS 寄存器事先设定的时间。

通过 8 位存储器操作指令设定 OSTS 寄存器。

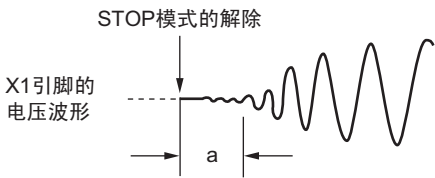
在产生复位信号后，此寄存器的值变为“07H”。

图 5-6 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 <sub>X</sub> =10MHz	f <sub>X</sub> =20MHz
0	0	0	2 <sup>8</sup> /f <sub>X</sub>	25.6μs	12.8μs
0	0	1	2 <sup>9</sup> /f <sub>X</sub>	51.2μs	25.6μs
0	1	0	2 <sup>10</sup> /f <sub>X</sub>	102μs	51.2μs
0	1	1	2 <sup>11</sup> /f <sub>X</sub>	204μs	102μs
1	0	0	2 <sup>13</sup> /f <sub>X</sub>	819μs	409μs
1	0	1	2 <sup>15</sup> /f <sub>X</sub>	3.27ms	1.63ms
1	1	0	2 <sup>17</sup> /f <sub>X</sub>	13.1ms	6.55ms
1	1	1	2 <sup>18</sup> /f <sub>X</sub>	26.2ms	13.1ms

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



备注 f<sub>X</sub>: X1 时钟振荡频率

5.3.6 外围允许寄存器 0、1（PER0、PER1）

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

当使用由此寄存器控制的以下外围功能时，必须在进行外围功能的初始设定前将对应的位置“1”。

- 串行接口 IICA1
- A/D 转换器
- D/A 转换器
- 串行接口 IICA0
- 串行阵列单元 0
- 定时器阵列单元 0

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 5-7 外围允许寄存器 0（PER0）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	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICA1EN	提供串行接口 IICA1 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"><li>• 不能写串行接口 IICA1 使用的 SFR。</li><li>• 串行接口 IICA1 处于复位状态。</li></ul>
1	提供输入时钟。 <ul style="list-style-type: none"><li>• 能读写串行接口 IICA1 使用的 SFR。</li></ul>

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

IICA0EN	提供串行接口 IICA0 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"><li>• 不能写串行接口 IICA0 使用的 SFR。</li><li>• 串行接口 IICA0 处于复位状态。</li></ul>
1	提供输入时钟。 <ul style="list-style-type: none"><li>• 能读写串行接口 IICA0 使用的 SFR。</li></ul>

注意 必须将 bit1、3、7 置“0”。

图 5-7 外围允许寄存器 0 (PER0) 的格式 (2/2)

地址: F00F0H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

SAU0EN	提供串行阵列单元 0 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> <li>不能写串行阵列单元 0 使用的 SFR。</li> <li>串行阵列单元 0 处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写串行阵列单元 0 使用的 SFR。</li> </ul>

TAU0EN	提供定时器阵列单元 0 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> <li>不能写定时器阵列单元 0 使用的 SFR。</li> <li>定时器阵列单元 0 处于复位状态。</li> </ul>
1	提供输入时钟。 <ul style="list-style-type: none"> <li>能读写定时器阵列单元 0 使用的 SFR。</li> </ul>

注意 必须将 bit1、3、7 置“0”。

图 5-8 外围允许寄存器 1 (PER1) 的格式

地址: F007AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER1	DACEN	0	0	0	0	0	0	0

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

注意 必须将 bit0 ~ 6 置“0”。

### 5.3.7 高速内部振荡器的频率选择寄存器（HOCODIV）

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

通过 8 位存储器操作指令设定 HOCODIV 寄存器。

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

图 5-9 高速内部振荡器的频率选择寄存器（HOCODIV）的格式

地址：F00A8H      复位后：选项字节（000C2H）的 FRQSEL2 ~ FRQSEL0 位的设定值      R/W

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

HOCODIV2	HOCODIV1	HOCODIV0	高速内部振荡器时钟频率的选择	
			FRQSEL3 位为“0”	FRQSEL3 位为“1”
0	0	0	24MHz	32MHz
0	0	1	12MHz	16MHz
0	1	0	6MHz	8MHz
0	1	1	3MHz	4MHz
1	0	0	禁止设定。	2MHz
1	0	1	禁止设定。	1MHz
上述以外			禁止设定。	

注意 1. 在更改频率前后，必须将 HOCODIV 寄存器设定在选项字节（000C2H）所设闪存运行模式的可工作电压范围内。

选项字节（000C2H）的值		闪存运行模式	工作频率范围	工作电压范围
CMODE1	CMODE0			
1	0	LS（低速主）模式	1MHz ~ 8MHz	2.7V ~ 3.6V
1	1	HS（高速主）模式	1MHz ~ 32MHz	2.7V ~ 3.6V

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

5.3.8 高速内部振荡器的微调寄存器（HIOTRM）

这是校正高速内部振荡器精度的寄存器。

能使用高精度的外部时钟输入的定时器（定时器阵列单元）等进行高速内部振荡器频率的自测量和精度校正。

通过 8 位存储器操作指令设定 HIOTRM 寄存器。

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

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

图 5-10 高速内部振荡器的微调寄存器（HIOTRM）的格式

地址: F00A0H	复位后: 注		R/W					
符号	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	高速内部振荡器
0	0	0	0	0	0	最低速
0	0	0	0	0	1	↑
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
• • •						
1	1	1	1	1	0	↓
1	1	1	1	1	1	最高速

注 复位值是产品发货时的调整值。

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

2. 有关 HIOTRM 寄存器的使用例子，请参照 RL78 MCU 系列高速内部振荡器时钟频率校正的应用说明（R01AN0464）。

## 5.4 系统时钟振荡电路

### 5.4.1 X1 振荡电路

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

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

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

当不使用 X1 振荡电路时，必须设定为输入端口模式（EXCLK、OSCSEL=0、0）。

而且，当也不用作输入端口时，请参照“表 2-4 各未使用引脚的处理”。

X1 振荡电路的外接电路例子如图 5-11 所示。

图 5-11 X1 振荡电路的外接电路例子

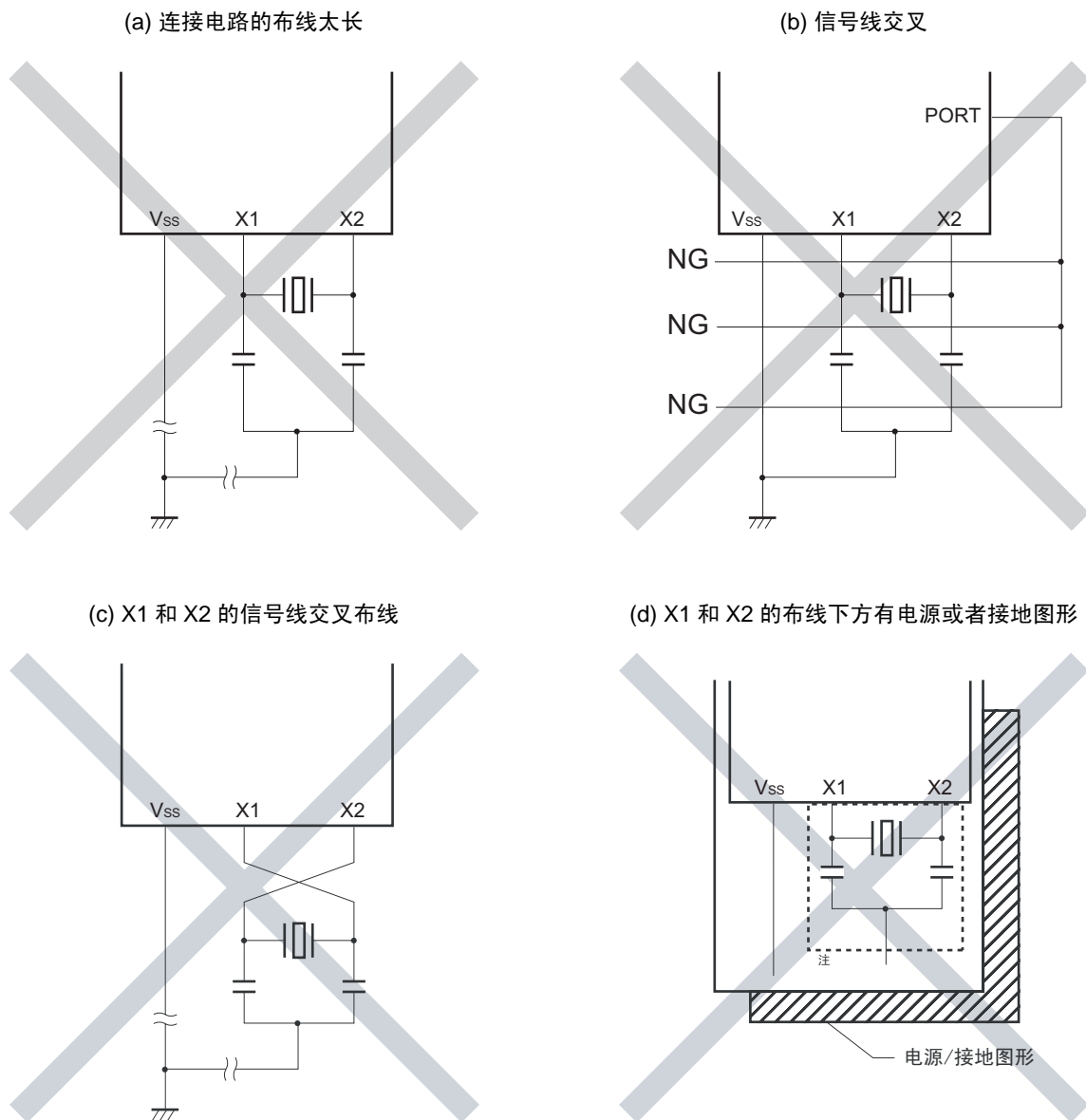


注意事项如下页所示。



不正确的谐振器连接例子如图 5-12 所示。

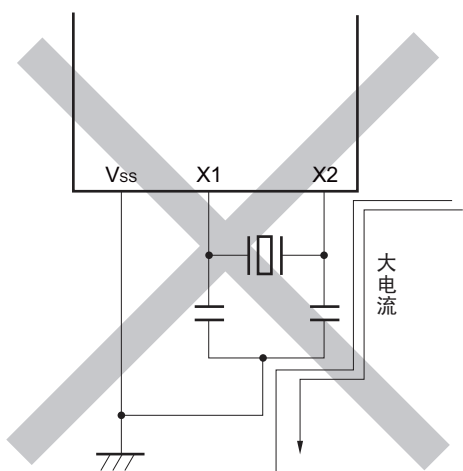
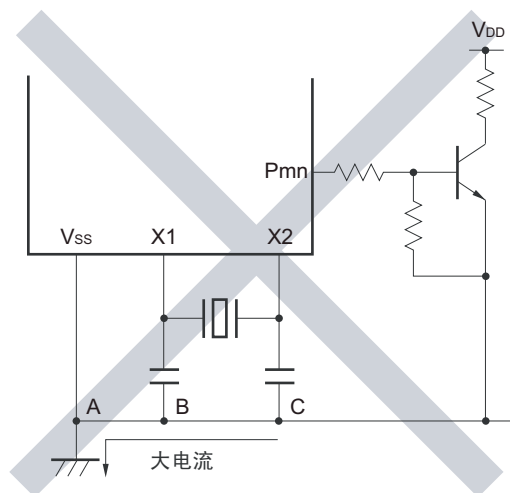
图 5-12 不正确的谐振器连接例子 (1/2)



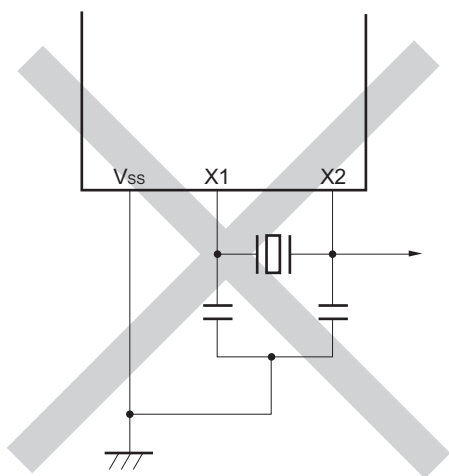
注 在多层板或者双面板中，不能在 X1 引脚、X2 引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

图 5-12 不正确的谐振器连接例子 (2/2)

(e) 有变化的大电流接近信号线

(f) 振荡电路的接地线有电流流过  
(A 点、B 点、C 点的电位发生变化)

(g) 取出信号



### 5.4.2 高速内部振荡器

R7F0C010 内置高速内部振荡器。能通过选项字节（000C2H）从 32MHz、24MHz、16MHz、12MHz、8MHz、4MHz 和 1MHz 中选择频率。能通过时钟运行状态控制寄存器（CSC）的 bit0（HIOSTOP）控制振荡。

在解除复位后，高速内部振荡器自动开始振荡。

### 5.4.3 低速内部振荡器

R7F0C010 内置低速内部振荡器。

低速内部振荡器时钟用作看门狗定时器，但是不能用作 CPU 时钟。

当选项字节（000C0H）的 bit4（WDTON）为“1”时，低速内部振荡器振荡。但是，如果看门狗定时器运行，就在 WDSTBYON 位为“0”并且处于 HALT 模式、STOP 模式或者 SNOOZE 模式时低速内部振荡器停止振荡。在看门狗定时器运行时，即使程序失控，低速内部振荡器时钟也不停止运行。

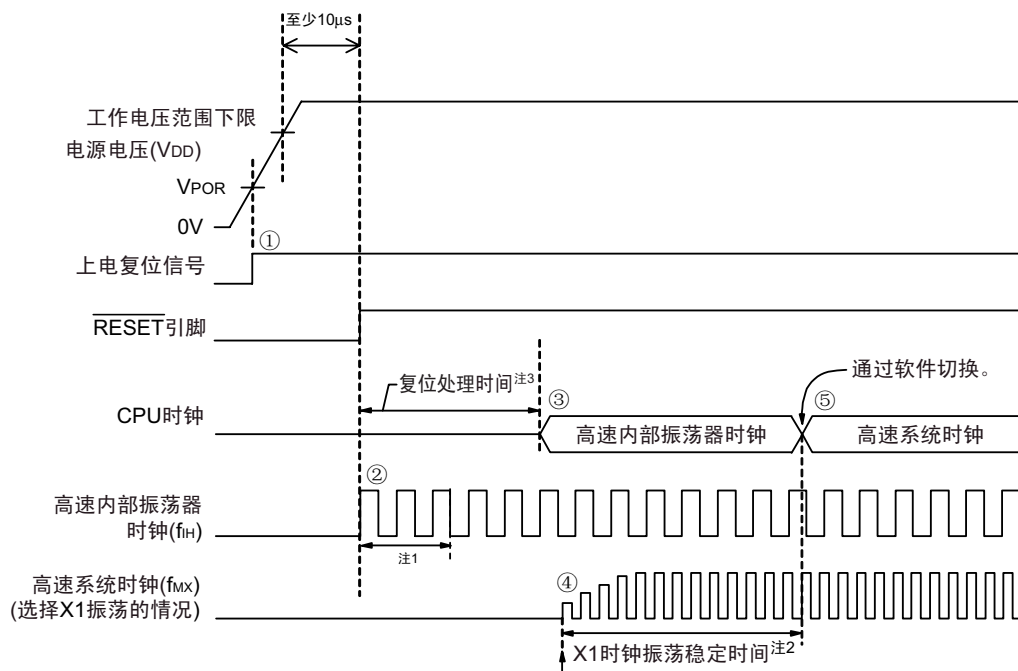
## 5.5 时钟发生电路的运行

时钟发生电路产生以下所示各种时钟，并且控制待机模式等 CPU 的运行模式（参照图 5-1）。

- 主系统时钟  $f_{\text{MAIN}}$ 
  - 高速系统时钟  $f_{\text{MX}}$ 
    - X1 时钟  $f_{\text{X}}$
    - 外部主系统时钟  $f_{\text{EX}}$
  - 高速内部振荡器时钟  $f_{\text{IH}}$
- 低速内部振荡器时钟  $f_{\text{IL}}$
- CPU/外围硬件时钟  $f_{\text{CLK}}$

R7F0C010 在解除复位后，CPU 通过高速内部振荡器的输出开始运行。  
接通电源时的时钟发生电路的运行如图 5-13 所示。

图 5-13 接通电源时的时钟发生电路的运行



- ① 在接通电源后，通过上电复位（POR）电路产生内部复位信号。  
但是，在达到“27.4 AC特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态（上图是使用外部复位时的例子）。
- ② 如果电源电压超过1.51V(TYP.)，就解除复位并且高速内部振荡器自动开始振荡。
- ③ 在解除复位后，进行电压稳定等待和复位处理，然后CPU以高速内部振荡器时钟开始运行。
- ④ 必须通过软件设定X1时钟的开始振荡（参照“5.6.2 X1振荡电路的设定例子”）。
- ⑤ 如果要将CPU时钟切换到X1时钟，就必须在等待时钟振荡稳定后通过软件设定切换（参照“5.6.2 X1振荡电路的设定例子”）。

- 注 1. 复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 当解除复位时，必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认X1时钟的振荡稳定时间。
3. 有关复位处理时间，请参照“第18章 上电复位电路”。

注意 如果使用EXCLK引脚输入的外部时钟，就不需要振荡稳定等待时间。

## 5.6 时钟控制

### 5.6.1 高速内部振荡器的设定例子

在解除复位后，CPU/ 外围硬件时钟 ( $f_{CLK}$ ) 一定以高速内部振荡器时钟运行。能通过选项字节 (000C2H) 的 FRQSEL0 ~ FRQSEL3 位，从 32MHz、24MHz、16MHz、12MHz、8MHz、4MHz 和 1MHz 中选择高速内部振荡器的频率。另外，能通过高速内部振荡器的频率选择寄存器 (HOCODIV) 更改频率。

#### 【选项字节的设定】

地址：000C2H

选项 字节 (000C2H)	7	6	5	4	3	2	1	0
	CMODE1 0/1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	闪存运行模式设定	
1	0	LS (低速主) 模式	$V_{DD}=2.7V \sim 3.6V@1MHz \sim 8MHz$
1	1	HS (高速主) 模式	$V_{DD}=2.7V \sim 3.6V@1MHz \sim 32MHz$
上述以外		禁止设定。	

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的频率
1	0	0	0	32MHz
0	0	0	0	24MHz
1	0	0	1	16MHz
0	0	0	1	12MHz
1	0	1	0	8MHz
1	0	1	1	4MHz
1	1	0	1	1MHz
上述以外				禁止设定。

#### 【高速内部振荡器的频率选择寄存器 (HOCODIV) 的设定】

地址：F00A8H

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

HOCODIV2	HOCODIV1	HOCODIV0	高速内部振荡器时钟频率的选择	
			FRQSEL3 位为“0”	FRQSEL3 位为“1”
0	0	0	24MHz	32MHz
0	0	1	12MHz	16MHz
0	1	0	6MHz	8MHz
0	1	1	3MHz	4MHz
1	0	0	禁止设定。	2MHz
1	0	1	禁止设定。	1MHz
上述以外			禁止设定。	

### 5.6.2 X1 振荡电路的设定例子

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

【寄存器的设定】 必须按照①～⑤的顺序设定寄存器。

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

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSSEL						AMPH 注
	0	1	0	0	0	0	0	0/1

注 当 X1 振荡时钟小于等于 10MHz 时，必须置“0”。

② 通过OSTS寄存器选择解除STOP模式时的X1振荡电路的振荡稳定时间。

例) 要通过10MHz谐振器至少等待102.4 $\mu$ s时，必须设定为以下的值。

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

③ 将CSC寄存器的MSTOP位清“0”，使X1振荡电路开始振荡。

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

④ 通过OSTC寄存器等待X1振荡电路的振荡稳定。

例) 要通过10MHz谐振器至少等待102.4 $\mu$ s时，必须等到各位变为以下的值。

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

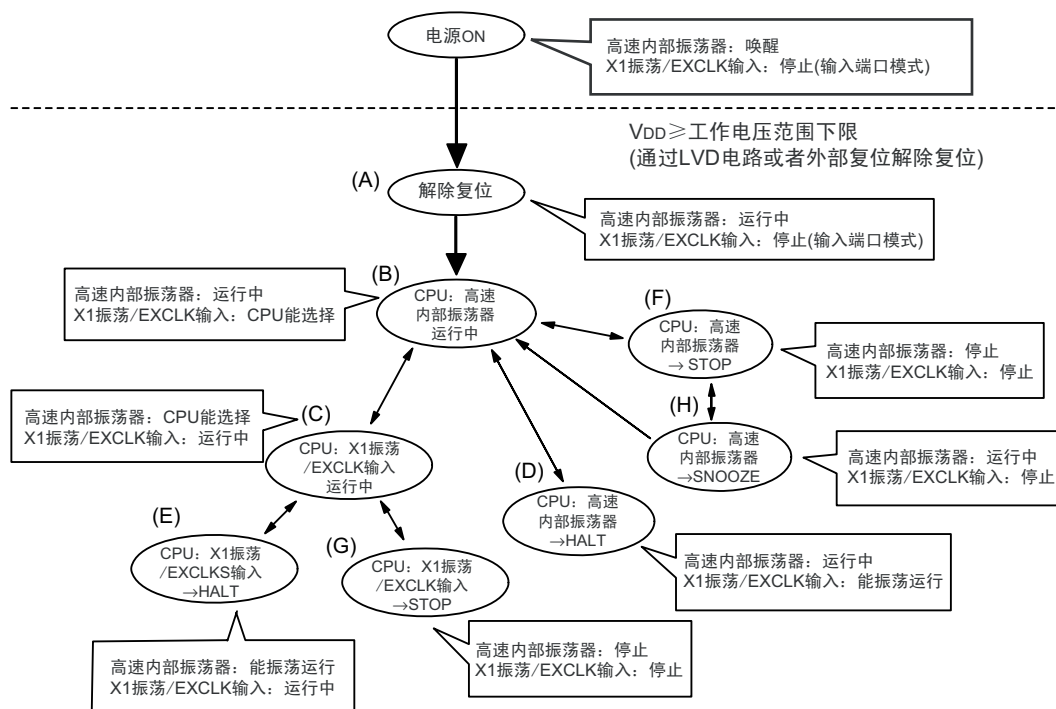
⑤ 通过CKC寄存器的MCM0位将X1振荡时钟设定为CPU/外围硬件时钟。

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

### 5.6.3 CPU 时钟的状态转移图

本产品的 CPU 时钟状态转移图如图 5-14 所示。

图 5-14 CPU 时钟的状态转移图





CPU 时钟的转移和 SFR 寄存器的设定例子等如表 5-3 所示。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (1/3)

(1) 在解除复位(A)后，CPU 转移到高速内部振荡器时钟运行(B)。

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

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

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

- 注 1. 在解除复位后，只能通过 8 位存储器操作指令写 1 次时钟运行模式控制寄存器（CMC）。
2. 必须对振荡稳定时间选择寄存器（OSTS）的振荡稳定时间进行以下的设定：
- 期待的振荡稳定时间计数器的状态寄存器（OSTC）的振荡稳定时间 ≤ OSTS 寄存器设定的振荡稳定时间

注意 必须在电源电压达到设定的时钟可运行电压（参照“第 27 章 电特性”）后设定时钟。

- 备注 1. ×：忽略
2. 表 5-3 的 (A) ~ (H) 对应图 5-14 的 (A) ~ (H)。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (2/3)

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

(SFR 寄存器的设定顺序) →

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

如果已设定就不需要。

在高速系统时钟运行中不需要。

- 注 1. 在解除复位后，只能设定 1 次时钟运行模式控制寄存器（CMC）。如果已设定就不需要。
2. 必须对振荡稳定时间选择寄存器（OSTS）的振荡稳定时间进行以下的设定：
- 期待的振荡稳定时间计数器的状态寄存器（OSTC）的振荡稳定时间 ≤ OSTS 寄存器设定的振荡稳定时间

注意 必须在电源电压达到设定的时钟可运行电压（参照“第 27 章 电特性”）后设定时钟。

- 备注 1. ×：忽略
2. 表 5-3 的 (A) ~ (H) 对应图 5-14 的 (A) ~ (H)。

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

(SFR 寄存器的设定顺序) 

<div>SFR 寄存器的设定标志</div> <div>状态转移</div>	CSC 寄存器	振荡精度稳定的等待	CKC 寄存器
	HIOSTOP		MCM0
(C)→(B)	0	18μs ~ 65μs	0

  
在高速内部振荡器时钟运行中不需要。

- 备注 1. 表 5-3 的 (A) ~ (H) 对应图 5-14 的 (A) ~ (H)。
2. 高速内部振荡器时钟的振荡精度稳定等待因温度条件和 STOP 模式期间而变。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (3/3)

- (5) • CPU在高速内部振荡器时钟运行中(B)转移到HALT模式(D)。
- CPU在高速系统时钟运行中(C)转移到HALT模式(E)。

状态转移	设定内容
(B)→(D)	执行 HALT 指令。
(C)→(E)	

- (6) • CPU在高速内部振荡器时钟运行中(B)转移到STOP模式(F)。
- CPU在高速系统时钟运行中(C)转移到STOP模式(G)。

(设定顺序) →

状态转移		设定内容		
(B)→(F)		停止禁止在 STOP 模式中运行的外围功能。	—	执行 STOP 指令。
(C)→(G)	X1 振荡		设定 OSTS 寄存器。	
	外部时钟		—	

- (7) CPU从STOP模式(F)转移到SNOOZE模式(H)。
- 有关从STOP模式转移到SNOOZE模式的设定，请参照“9.8 SNOOZE 模式功能”、“11.5.7 SNOOZE 模式功能（CSI00）”和“11.7.3 SNOOZE 模式功能”。

备注 表 5-3 的 (A) ~ (H) 对应图 5-14 的 (A) ~ (H)。

#### 5.6.4 CPU 时钟转移前的条件和转移后的处理

CPU 时钟转移前的条件和转移后的处理如下所示。

表 5-4 有关 CPU 时钟的转移

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

5.6.5 CPU 时钟和系统时钟的切换所需时间

能通过设定系统时钟控制寄存器（CKC）的 bit4（MCM0）进行主系统时钟的切换（高速内部振荡器时钟 ↔ 高速系统时钟）。

在改写 CKC 寄存器后不立即进行实际的切换，而是在更改 CKC 寄存器后仍然以切换前的时钟继续运行数个时钟（参照表 5-5 和表 5-6）。

能通过 CKC 寄存器的 bit5（MCS）来判断主系统时钟是以高速系统时钟还是以高速内部振荡器时钟运行。如果切换 CPU 时钟，就同时切换外围硬件时钟。

表 5-5 切换系统时钟所需要的最长时间

时钟 A	切换方向	时钟 B	备注
f <sub>IH</sub>	↔	f <sub>MX</sub>	参照表 5-6。

表 5-6 f<sub>IH</sub>↔f<sub>MX</sub> 所需要的最大时钟数

切换前的设定值		切换后的设定值	
MCM0		MCM0	
		0 (f <sub>MAIN</sub> =f <sub>IH</sub> )	1 (f <sub>MAIN</sub> =f <sub>MX</sub> )
0 (f <sub>MAIN</sub> =f <sub>IH</sub> )	f <sub>MX</sub> ≥ f <sub>IH</sub>		2 个时钟
	f <sub>MX</sub> < f <sub>IH</sub>		2 个 f <sub>IH</sub> /f <sub>MX</sub> 时钟
1 (f <sub>MAIN</sub> =f <sub>MX</sub> )	f <sub>MX</sub> ≥ f <sub>IH</sub>	2 个 f <sub>MX</sub> /f <sub>IH</sub> 时钟	
	f <sub>MX</sub> < f <sub>IH</sub>	2 个时钟	

备注 1. 表 5-6 中的时钟数是切换前的 CPU 时钟数。

2. 表 5-6 中的时钟数是舍入小数部分的时钟数。

例 主系统时钟从高速系统时钟切换到高速内部振荡器时钟的情况（选择 f<sub>IH</sub>=8MHz、f<sub>MX</sub>=10MHz 振荡时）

2f<sub>MX</sub>/f<sub>IH</sub>=2(10/8)=2.5→3 个时钟

5.6.6 时钟振荡停止前的条件

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

表 5-7 时钟振荡停止前的条件和标志设定

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

## 5.7 谐振器和振荡电路常数

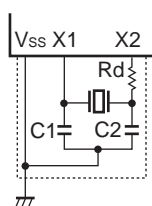
已验证的谐振器及其振荡电路常数（供参考）如下所示。

注意 1. 此振荡电路常数是基于谐振器厂商在特定环境下进行评估的参考值。在实际应用中，请委托谐振器厂商给予安装电路后的评估。

在更改其他产品的单片机和电路板时，请委托谐振器厂商再次给予安装电路后的评估。

2. 振荡电压和振荡频率原则上是表示振荡电路的特性。有关 RL78 微控制器的内部工作条件，请在 DC、AC 特性的规格内使用。

图 5-15 外接振荡电路的例子



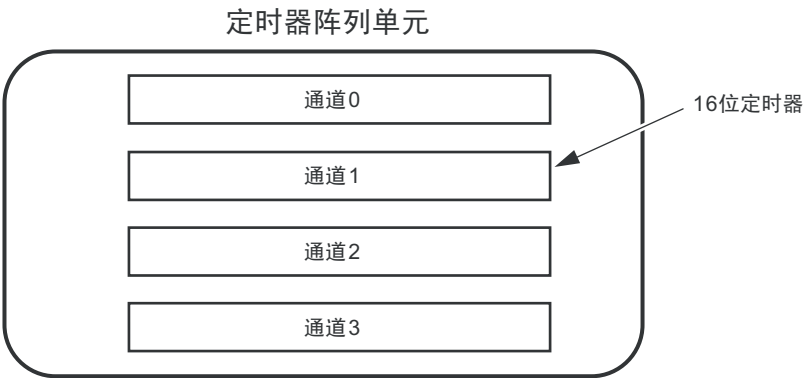
第 6 章 定时器阵列单元

定时器阵列单元的通道数因产品而不同。

单元	通道	24 引脚和 32 引脚
单元 0	通道 0	○
	通道 1	○
	通道 2	○
	通道 3	○

注意 本章的下述内容主要针对 32 引脚产品进行说明。

定时器阵列单元有 4 个 16 位定时器。  
各 16 位定时器称为“通道”，既能分别用作独立的定时器，也能组合多个通道用作高级的定时器功能。



有关各功能的详细内容，请参照下表。

独立通道运行功能	多通道联动运行功能
<ul style="list-style-type: none"><li>• 间隔定时器（→ 参照 6.8.1）</li><li>• 方波输出（→ 参照 6.8.1）</li><li>• 外部事件计数器（→ 参照 6.8.2）</li><li>• 输入脉冲间隔的测量（→ 参照 6.8.3）</li><li>• 输入信号的高低电平宽度的测量（→ 参照 6.8.4）</li><li>• 延迟计数器（→ 参照 6.8.5）</li></ul>	<ul style="list-style-type: none"><li>• 单触发脉冲输出（→ 参照 6.9.1）</li><li>• PWM 输出（→ 参照 6.9.2）</li><li>• 多重 PWM 输出（→ 参照 6.9.3）</li></ul>

能将单元 0 的通道 1 和通道 3 的 16 位定时器用作 2 个 8 位定时器（高位和低位）。通道 1 和通道 3 能用作 8 位定时器的功能如下：

- 间隔定时器（高 8 位和低 8 位定时器）/方波输出（只限于低 8 位定时器）
- 外部事件计数器（只限于低 8 位定时器）
- 延迟计数器（只限于低 8 位定时器）

## 6.1 定时器阵列单元的功能

定时器阵列单元有以下功能：

### 6.1.1 独立通道运行功能

独立通道运行功能是能不受其他通道运行模式的影响而独立使用任意通道的功能。

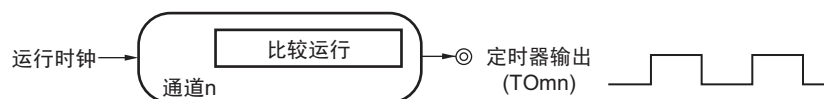
#### (1) 间隔定时器

能用作以固定间隔产生中断（INTTMmn）的基准定时器。



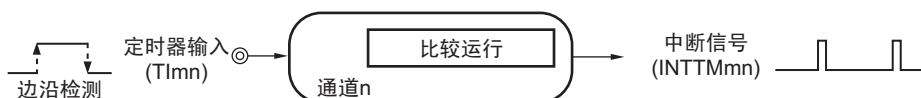
#### (2) 方波输出

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



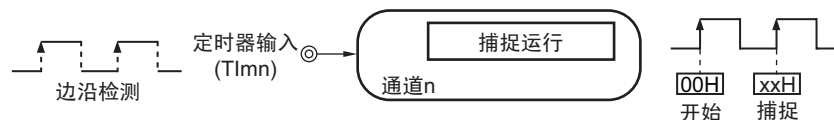
#### (3) 外部事件计数器

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



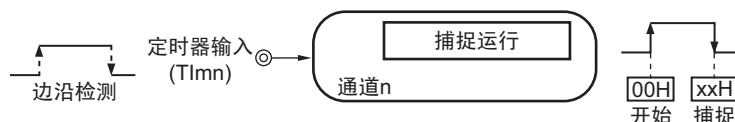
#### (4) 输入脉冲间隔的测量

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



#### (5) 输入信号的高低电平宽度的测量

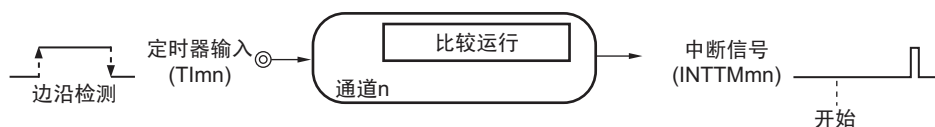
在定时器输入引脚（TIMn）的输入信号的一个边沿开始计数并且在另一个边沿捕捉计数值，从而测量输入信号的高低电平的宽度。





### (6) 延迟计数器

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



备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

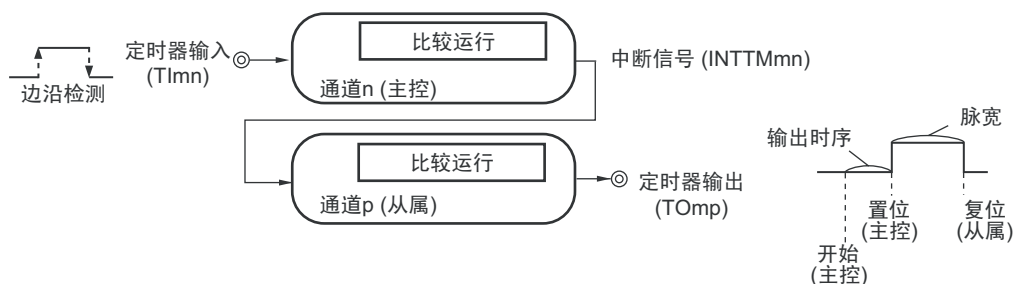
## 6.1.2 多通道联动运行功能

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

多通道联动运行功能能用作以下模式。

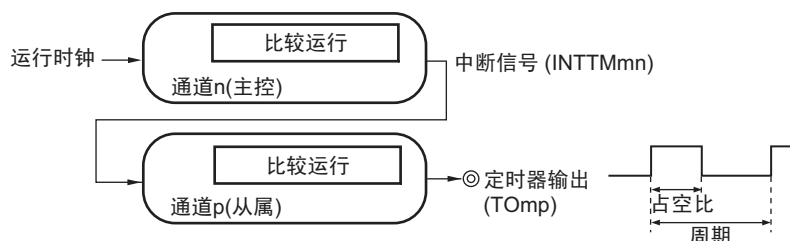
### (1) 单触发脉冲输出

将 2 个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



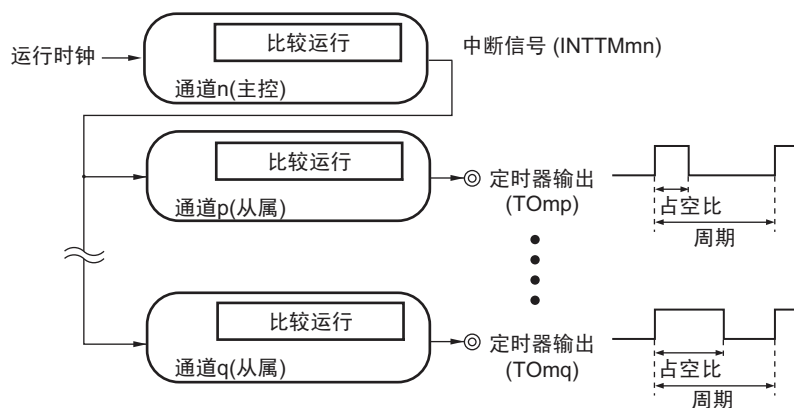
### (2) PWM (Pulse Width Modulation) 输出

将 2 个通道成对使用，生成能任意设定周期和占空比的脉冲。



### (3) 多重 PWM (Pulse Width Modulation) 输出

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



**注意** 有关多通道联动运行功能规则的详细内容，请参照“6.4.1 多通道联动运行功能的基本规则”。

**备注** m: 单元号 (m=0) n: 通道号 (n=0 ~ 3) p、q: 从属通道号 ( $n < p < q \leq 3$ )

### 6.1.3 8 位定时器运行功能 (只限于通道 1 和通道 3)

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

**注意** 在使用 8 位定时器运行功能时，有几个规则。

详细内容请参照“6.4.2 8 位定时器运行功能的基本规则”。

## 6.2 定时器阵列单元的结构

定时器阵列单元由以下硬件构成。

表 6-1 定时器阵列单元的结构

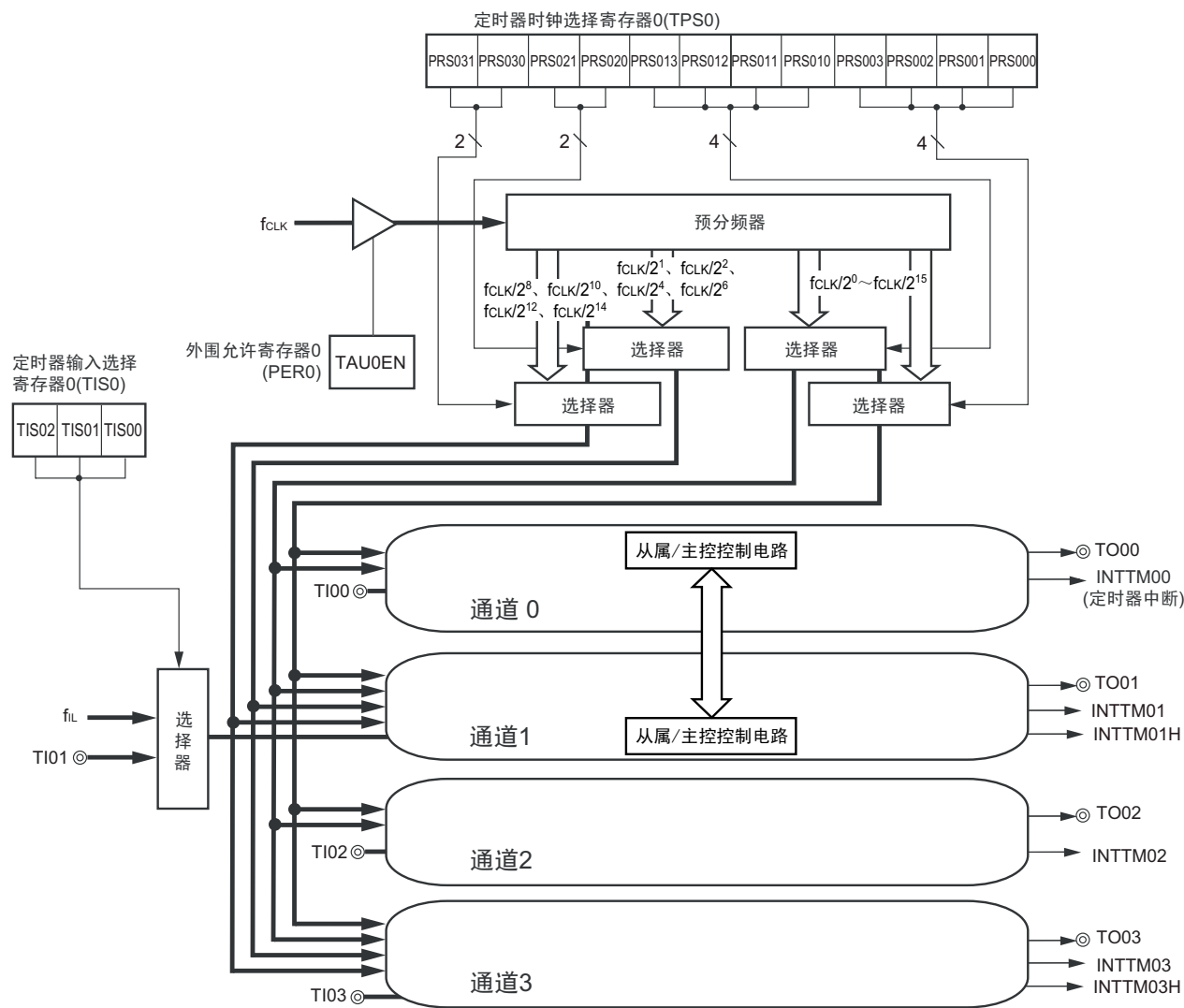
项目	结构
计数器	定时器计数寄存器 mn (TCRmn)
寄存器	定时器数据寄存器 mn (TDRmn)
定时器的输入	TI00 ~ TI03
定时器的输出	TO00 ~ TO03、输出控制电路
控制寄存器	<单元设定部的寄存器> <ul style="list-style-type: none"> <li>• 外围允许寄存器 0 (PER0)</li> <li>• 定时器时钟选择寄存器 m (TPSm)</li> <li>• 定时器通道允许状态寄存器 m (TEm)</li> <li>• 定时器通道开始寄存器 m (TSm)</li> <li>• 定时器通道停止寄存器 m (TTm)</li> <li>• 定时器输入选择寄存器 0 (TIS0)</li> <li>• 定时器输出允许寄存器 m (TOEm)</li> <li>• 定时器输出寄存器 m (TOm)</li> <li>• 定时器输出电平寄存器 m (TOLm)</li> <li>• 定时器输出模式寄存器 m (TOMm)</li> </ul>
	<每个通道的寄存器> <ul style="list-style-type: none"> <li>• 定时器模式寄存器 mn (TMRmn)</li> <li>• 定时器状态寄存器 mn (TSRmn)</li> <li>• 噪声滤波器允许寄存器 1 (NFEN1)</li> <li>• 端口模式寄存器 (PMxx) 注</li> <li>• 端口寄存器 (Pxx) 注</li> </ul>

注 设定的端口模式寄存器 (PMxx) 和端口寄存器 (Pxx) 因产品而不同。详细内容请参照“4.5 使用复用功能时的端口相关寄存器的设定”。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

定时器阵列单元的框图如图 6-1 ~ 图 6-5 所示。

图 6-1 定时器阵列单元 0 的整体框图



备注  $f_{IL}$  : 低速内部振荡器时钟频率

图 6-2 定时器阵列单元 0 的通道 0 的内部框图

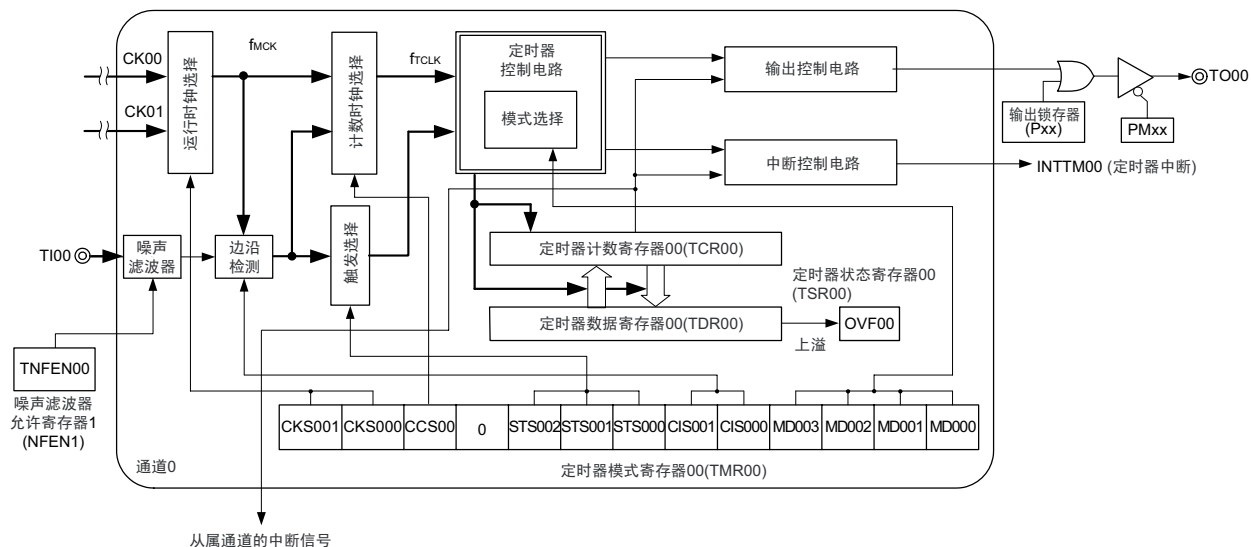


图 6-3 定时器阵列单元 0 的通道 1 的内部框图

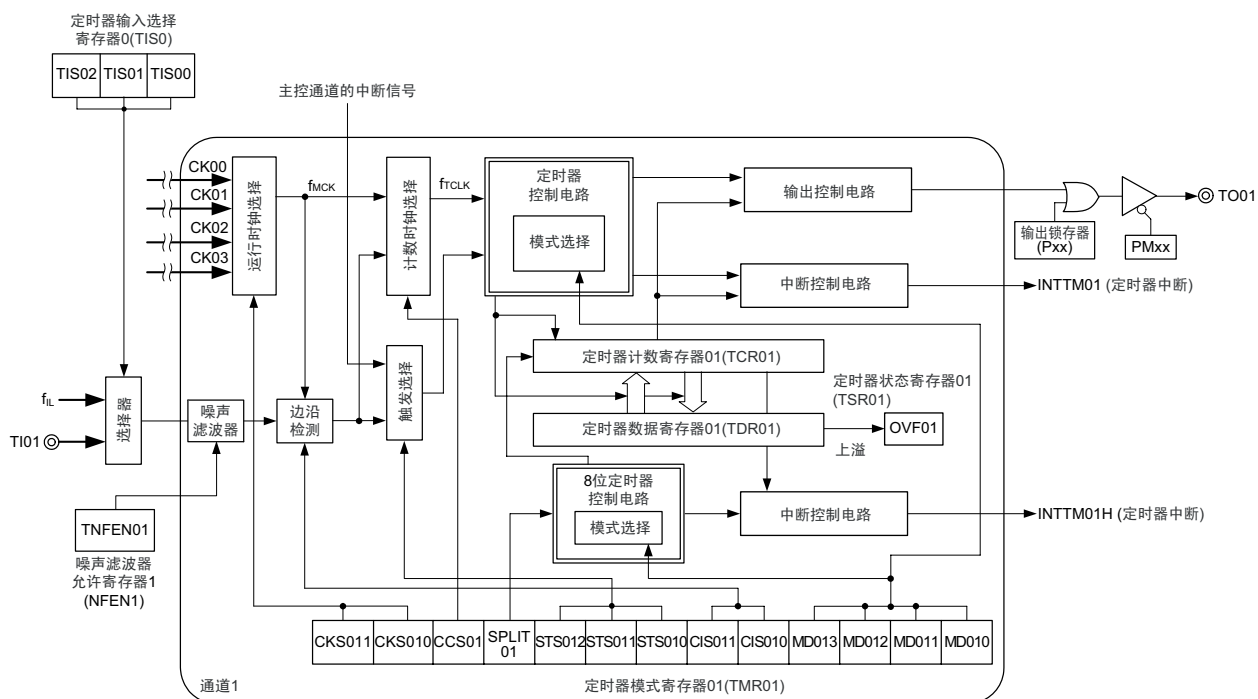


图 6-4 定时器阵列单元 0 的通道 2 的内部框图

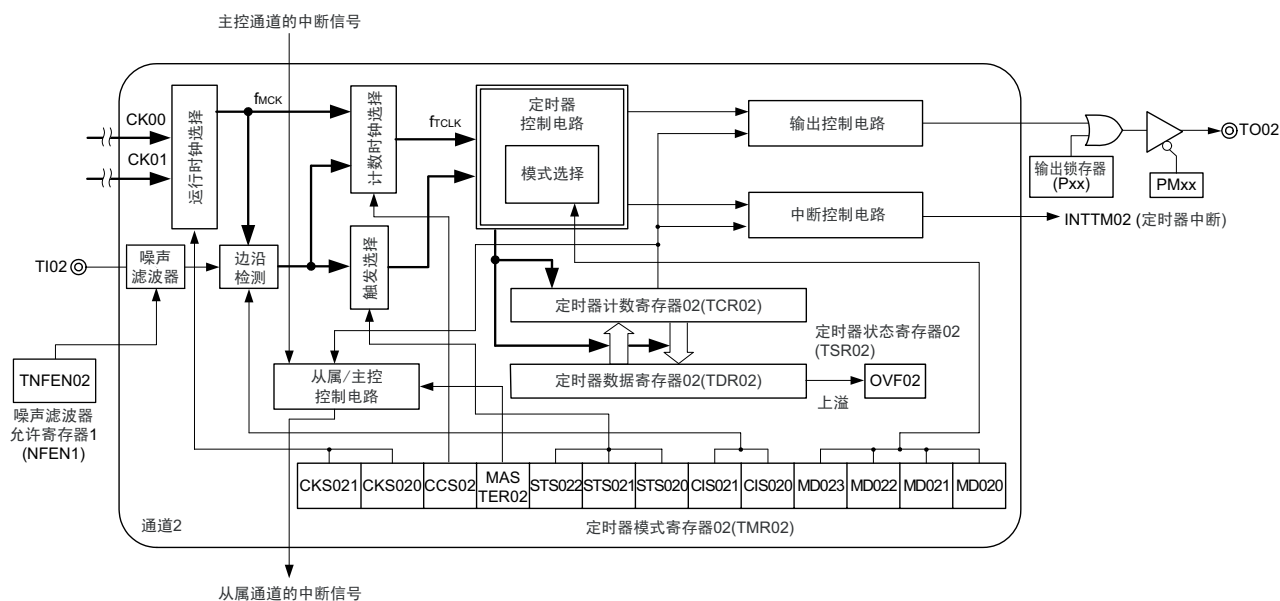
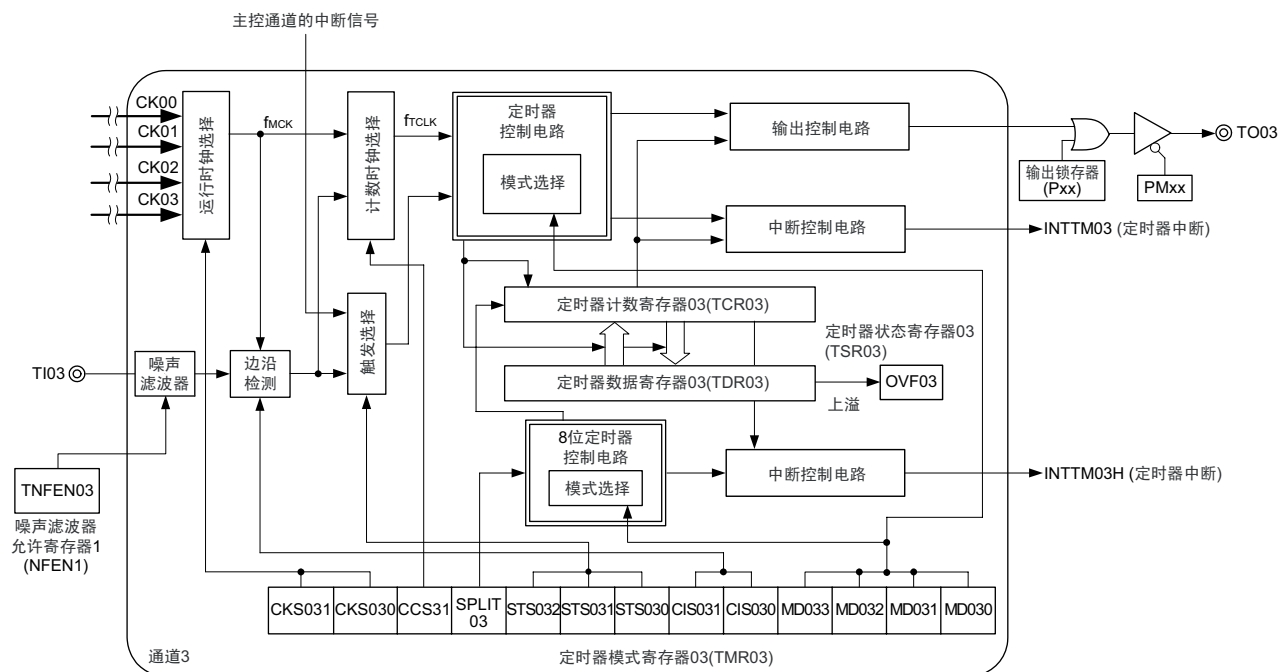


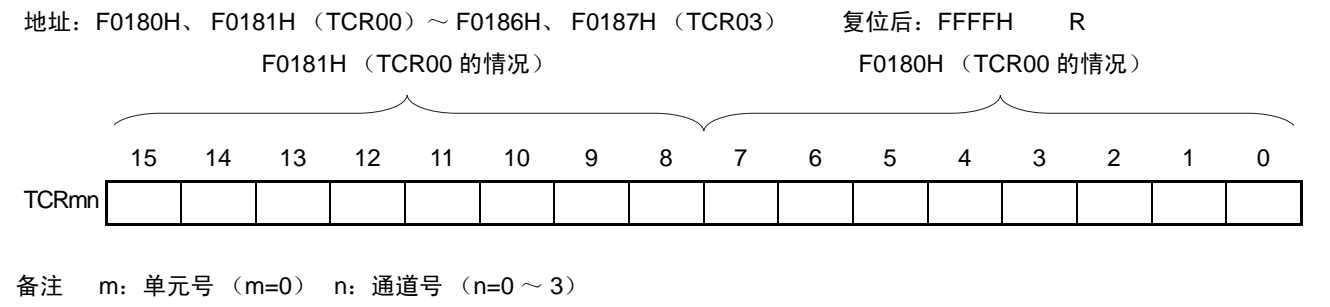
图 6-5 定时器阵列单元 0 的通道 3 的内部框图



6.2.1 定时器计数寄存器 mn（TCRmn）

TCRmn 寄存器是对计数时钟进行计数的 16 位只读寄存器。  
与计数时钟的上升沿同步进行递增或者递减计数。  
通过定时器模式寄存器 mn（TMRmn）的 MDmn3 ~ MDmn0 位来选择运行模式，进行递增和递减计数的切换（参照“6.3.3 定时器模式寄存器 mn（TMRmn）”）。

图 6-6 定时器计数寄存器 mn（TCRmn）的格式



能通过读定时器计数寄存器  $mn$  (TCR $mn$ ) 来读计数值。

在以下情况下，计数值变为“FFFFH”。

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

在以下情况下，计数值变为“0000H”。

- 在捕捉模式中输入开始触发时
- 在捕捉模式中捕捉结束时

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

如下所示，TCR $mn$  寄存器的读取值因运行模式和运行状态而不同。

表 6-2 各运行模式中的定时器计数寄存器  $mn$  (TCR $mn$ ) 的读取值

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

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

**备注**  $m$ : 单元号 ( $m=0$ )  $n$ : 通道号 ( $n=0 \sim 3$ )



## 6.2.2 定时器数据寄存器 mn (TDRmn)

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

能随时改写 TDRmn 寄存器的值。

能以 16 位为单位读写此寄存器。

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

在产生复位信号后，TDRmn 寄存器的值变为“0000H”。

图 6-7 定时器数据寄存器 mn (TDRmn) 的格式 (n=0、2)

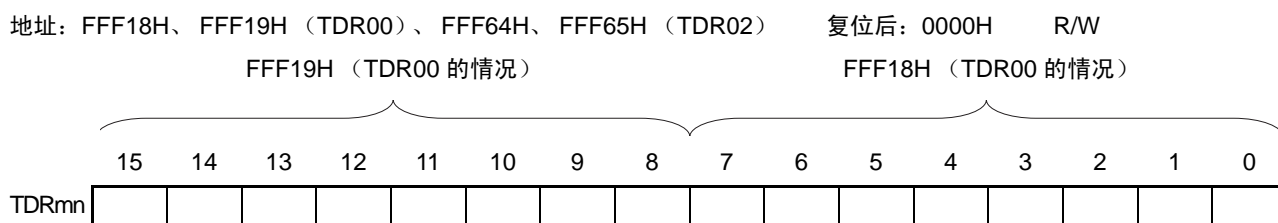
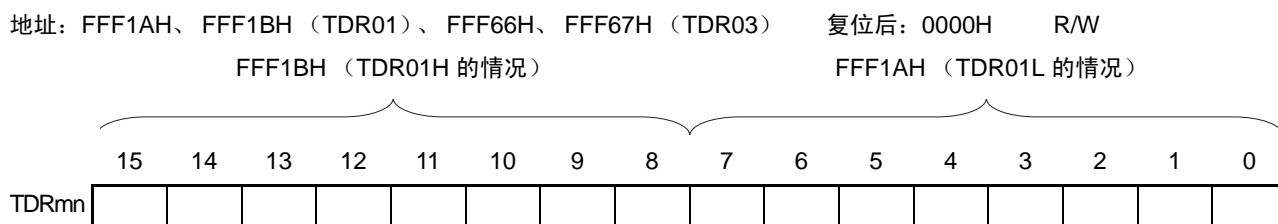


图 6-8 定时器数据寄存器 mn (TDRmn) 的格式 (n=1、3)



(i) 定时器数据寄存器 mn (TDRmn) 用作比较寄存器的情况

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

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

(ii) 定时器数据寄存器 mn (TDRmn) 用作捕捉寄存器的情况

通过输入捕捉触发，将定时器计数寄存器 mn (TCRmn) 的计数值捕捉到 TDRmn 寄存器。

能选择 TIMn 引脚的有效边沿作为捕捉触发信号。通过定时器模式寄存器 mn (TMRmn) 来设定捕捉触发的选择。

**备注** m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

### 6.3 控制定时器阵列单元的寄存器

控制定时器阵列单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 定时器时钟选择寄存器m (TPSm)
- 定时器模式寄存器mn (TMRmn)
- 定时器状态寄存器mn (TSRmn)
- 定时器通道允许状态寄存器m (TEm)
- 定时器通道开始寄存器m (TSm)
- 定时器通道停止寄存器m (TTm)
- 定时器输入选择寄存器0 (TIS0)
- 定时器输出允许寄存器m (TOEm)
- 定时器输出寄存器m (TOM)
- 定时器输出电平寄存器m (TOLm)
- 定时器输出模式寄存器m (TOMm)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器1 (NFEN1)
- 端口模式寄存器 (PMxx)
- 端口寄存器 (Pxx)

**注意** 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

**备注** m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.1 外围允许寄存器 0（PER0）

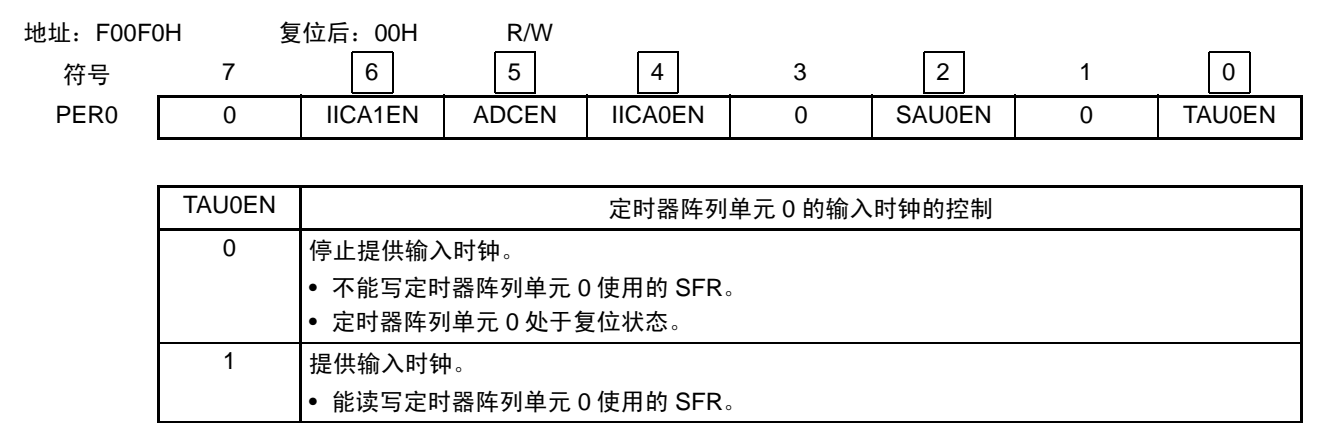
PER0 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用定时器阵列单元 0 时，必须将 bit0（TAU0EN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，PER0 寄存器的值变为“00H”。

图 6-9 外围允许寄存器 0（PER0）的格式



注意 1. 要设定定时器阵列单元时，必须先在 TAUmEN 位为“1”的状态下设定以下的寄存器。当 TAUmEN 位为“0”时，忽视定时器阵列单元的控制寄存器的写操作，而且读取值都为初始值（定时器输入选择寄存器 0（TISO）、输入切换控制寄存器（ISC）、噪声滤波器允许寄存器 1（NFEN1）、端口模式寄存器 1、3（PM1、PM3）和端口寄存器 1、3（P1、P3）除外）。

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

2. 必须将 bit1、3、7 置“0”。

### 6.3.2 定时器时钟选择寄存器 m (TPSm)

TPSm 寄存器是选择从外部预分频器共同提供给各通道的 2 种或者 4 种运行时钟 (CKm0、CKm1、CKm2、CKm3) 的 16 位寄存器。通过 TPSm 寄存器的 bit7 ~ 4 选择 CKm1, 通过 bit3 ~ 0 选择 CKm0。另外, 对于通道 1 和通道 3, 通过 TPSm 寄存器的 bit9 和 bit8 选择 CKm2, 通过 bit13 和 bit12 选择 CKm3。

只有在以下情况下才能改写定时器运行中的 TPSm 寄存器。

能改写 PRSm00 ~ PRSm03 位的情况 (n=0 ~ 3):

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

能改写 PRSm10 ~ PRSm13 位的情况 (n=0 ~ 3):

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

能改写 PRSm20 位和 PRSm21 位的情况 (n=1、3):

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

能改写 PRSm30 位和 PRSm31 位的情况 (n=1、3):

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

通过 16 位存储器操作指令设定 TPSm 寄存器。

在产生复位信号后, TPSm 寄存器的值变为“0000H”。

图 6-10 定时器时钟选择寄存器 m (TPSm) 的格式 (1/2)

地址: F01B6H、F01B7H (TPS0)

复位后: 0000H

R/W

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

PRS mk3	PRS mk2	PRS mk1	PRS mk0	运行时钟 (CKmk) 的选择注 (k=0、1)					
					f <sub>CLK</sub> = 2MHz	f <sub>CLK</sub> = 5MHz	f <sub>CLK</sub> = 10MHz	f <sub>CLK</sub> = 20MHz	f <sub>CLK</sub> = 32MHz
0	0	0	0	f <sub>CLK</sub>	2MHz	5MHz	10MHz	20MHz	32MHz
0	0	0	1	f <sub>CLK</sub> /2	1MHz	2.5MHz	5MHz	10MHz	16MHz
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500kHz	1.25MHz	2.5MHz	5MHz	8MHz
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250kHz	625kHz	1.25MHz	2.5MHz	4MHz
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125kHz	312.5kHz	625kHz	1.25MHz	2MHz
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5kHz	156.2kHz	312.5kHz	625kHz	1MHz
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.25kHz	78.1kHz	156.2kHz	312.5kHz	500kHz
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.62kHz	39.1kHz	78.1kHz	156.2kHz	250kHz
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81kHz	19.5kHz	39.1kHz	78.1kHz	125kHz
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91kHz	9.76kHz	19.5kHz	39.1kHz	62.5kHz
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95kHz	4.88kHz	9.76kHz	19.5kHz	31.25kHz
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	976Hz	2.44kHz	4.88kHz	9.76kHz	15.63kHz
1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	488Hz	1.22kHz	2.44kHz	4.88kHz	7.81kHz
1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	244Hz	610Hz	1.22kHz	2.44kHz	3.91kHz
1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	122Hz	305Hz	610Hz	1.22kHz	1.95kHz
1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	61Hz	153Hz	305Hz	610Hz	976Hz

注 在更改选择为 f<sub>CLK</sub> 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止定时器阵列单元 (TTm=00FFH)。即使在选择运行时钟 (f<sub>MCK</sub>) 或者 TImn 引脚输入信号的有效边沿时, 也需要停止定时器阵列单元。

注意 1. 必须将 bit15、14、11、10 置“0”。

2. 如果选择 f<sub>CLK</sub> (无分频) 作为运行时钟 (CKmk) 并且将 TDRnm 置“0000H” (n=0, m=0 ~ 3), 就不检测来自定时器阵列单元的中断请求。

备注 1. f<sub>CLK</sub>: CPU/ 外围硬件的时钟频率

2. TPSm 寄存器选择的时钟波形从上升沿开始只有 1 个 f<sub>CLK</sub> 周期为高电平 (m=1 ~ 15)。详细内容请参照“6.5.1 计数时钟 (f<sub>TCLK</sub>)”。

图 6-10 定时器时钟选择寄存器 m (TPSm) 的格式 (2/2)

地址: F01B6H、F01B7H (TPS0) 复位后: 0000H R/W

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

PRS m21	PRS m20	运行时钟 (CKm2) 的选择注					
			$f_{CLK}=2MHz$	$f_{CLK}=5MHz$	$f_{CLK}=10MHz$	$f_{CLK}=20MHz$	$f_{CLK}=32MHz$
0	0	$f_{CLK}/2$	1MHz	2.5MHz	5MHz	10MHz	16MHz
0	1	$f_{CLK}/2^2$	500kHz	1.25MHz	2.5MHz	5MHz	8MHz
1	0	$f_{CLK}/2^4$	125kHz	312.5kHz	625MHz	1.25MHz	2MHz
1	1	$f_{CLK}/2^6$	31.25kHz	78.1kHz	156.2kHz	312.5kHz	500kHz

PRS m31	PRS m30	运行时钟 (CKm3) 的选择注					
			$f_{CLK}=2MHz$	$f_{CLK}=5MHz$	$f_{CLK}=10MHz$	$f_{CLK}=20MHz$	$f_{CLK}=32MHz$
0	0	$f_{CLK}/2^8$	7.81kHz	19.5kHz	39.1kHz	78.1kHz	125kHz
0	1	$f_{CLK}/2^{10}$	1.95kHz	4.88kHz	9.76kHz	19.5kHz	31.25kHz
1	0	$f_{CLK}/2^{12}$	488Hz	1.22kHz	2.44kHz	4.88kHz	7.81kHz
1	1	$f_{CLK}/2^{14}$	122Hz	305Hz	610Hz	1.22kHz	1.95kHz

注 在更改选择为  $f_{CLK}$  的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止定时器阵列单元 (TTm=00FFH)。即使在选择运行时钟 ( $f_{MCK}$ ) 或者 TImn 引脚输入信号的有效边沿时, 也需要停止定时器阵列单元。

注意 必须将 bit15、14、11、10 置“0”。

如果在 8 位定时器模式中使用通道 1 和通道 3 并且将 CKm2 和 CKm3 作为运行时钟, 就能通过间隔定时器功能实现表 6-3 所示的间隔时间。

表 6-3 运行时钟 CKSm2 和 CKSm3 能设定的间隔时间

时钟		间隔时间注 ( $f_{CLK}=32MHz$ )			
		10 $\mu$ s	100 $\mu$ s	1ms	10ms
CKm2	$f_{CLK}/2$	○	—	—	—
	$f_{CLK}/2^2$	○	—	—	—
	$f_{CLK}/2^4$	○	○	—	—
	$f_{CLK}/2^6$	○	○	—	—
CKm3	$f_{CLK}/2^8$	—	○	○	—
	$f_{CLK}/2^{10}$	—	○	○	—
	$f_{CLK}/2^{12}$	—	—	○	○
	$f_{CLK}/2^{14}$	—	—	○	○

注 ○ 包含 5% 以内的误差。

备注 1.  $f_{CLK}$ : CPU/ 外围硬件的时钟频率

2. 有关 TPSm 寄存器所选  $f_{CLK}/2^i$  波形的详细内容, 请参照“6.5.1 计数时钟 ( $f_{TCLK}$ )”。

### 6.3.3 定时器模式寄存器 mn (TMRmn)

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

禁止在运行中 (TEmn=1) 改写 TMRmn 寄存器。但是，能在一部分的功能运行中 (TEmn=1) 改写 bit7 和 bit6 (CISmn1、CISmn0)（详细内容请参照“6.8 定时器阵列单元的独立通道运行功能”和“6.9 定时器阵列单元的多通道联动运行功能”）。

通过 16 位存储器操作指令设定 TMRmn 寄存器。

在产生复位信号后，TMRmn 寄存器的值变为“0000H”。

注意 TMRmn 寄存器的 bit11 因通道而不同。

TMRm2、TMRm4、TMRm6: MASTERmn 位 (n=2)

TMRm1、TMRm3: SPLITmn 位 (n=1、3)

TMRm0、TMRm5、TMRm7: 固定为“0”。

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (1/4)

地址: F0190H、F0191H (TMR00) ~ F0196H、F0197H (TMR03) 复位后: 0000H R/W

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

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

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

CKS mn1	CKS mn0	通道 n 运行时钟 ( $f_{MCK}$ ) 的选择
0	0	定时器时钟选择寄存器 m (TPSm) 设定的运行时钟 CKm0
0	1	定时器时钟选择寄存器 m (TPSm) 设定的运行时钟 CKm2
1	0	定时器时钟选择寄存器 m (TPSm) 设定的运行时钟 CKm1
1	1	定时器时钟选择寄存器 m (TPSm) 设定的运行时钟 CKm3
运行时钟 ( $f_{MCK}$ ) 用于边沿检测电路。通过设定 CCSmn 位来产生采样时钟和计数时钟 ( $f_{TCLK}$ )。 只有通道 1 和通道 3 才能选择运行时钟 CKm2 和 CKm3。		

CCS mn	通道 n 计数时钟 ( $f_{TCLK}$ ) 的选择
0	CKSmn0 位和 CKSmn1 位指定的运行时钟 ( $f_{MCK}$ )
1	TImn 引脚输入信号的有效边沿 在通道 5 的情况下, 为 TIS0 选择的输入信号的有效边沿。
计数时钟 ( $f_{TCLK}$ ) 用于计数器、输出控制电路和中断控制电路。	

注 bit11 是只读位, 固定为“0”, 忽视写操作。

注意 1. 必须将 bit13、5、4 置“0”。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)



图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (2/4)

地址: F0190H、F0191H (TMR00) ~ F0196H、F0197H (TMR03) 复位后: 0000H R/W

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

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

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

(TMRmn (n=2) 的 bit11)

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

(TMRmn (n=1、3) 的 bit11)

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

STS mn2	STS mn1	STS mn0	通道 n 的开始触发和捕捉触发的设定
0	0	0	只有软件触发开始有效 (不选择其他触发源)。
0	0	1	将 TImn 引脚输入的有效边沿用于开始触发和捕捉触发。
0	1	0	将 TImn 引脚输入的双边沿分别用于开始触发和捕捉触发。
1	0	0	使用主控通道的中断信号 (多通道联动运行功能的从属通道的情况)。
上述以外			禁止设定。

注 bit11 是只读位, 固定为“0”, 忽视写操作。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-11  定时器模式寄存器 mn（TMRmn）的格式 (3/4)

地址: F0190H、F0191H（TMR00）～F0196H、F0197H（TMR03）      复位后: 0000H      R/W

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

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

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

CIS mn1	CIS mn0	Tlmn 引脚有效边沿的选择
0	0	下降沿
0	1	上升沿
1	0	双边沿（测量低电平宽度时） 开始触发：下降沿，捕捉触发：上升沿
1	1	双边沿（测量高电平宽度时） 开始触发：上升沿，捕捉触发：下降沿
当 STSmn2～STSmn0 位不为“010B”并且使用双边沿指定时，必须将 CISmn1～CISmn0 位置“10B”。		

注     bit11 是只读位，固定为“0”，忽视写操作。

备注   m: 单元号（m=0）   n: 通道号（n=0～3）

图 6-11 定时器模式寄存器 mn (TMRmn) 的格式 (4/4)

地址: F0190H、F0191H (TMR00) ~ F0196H、F0197H (TMR03) 复位后: 0000H R/W

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

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

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

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

运行模式 (MDmn3 ~ MDmn1 位的设定 (参照上表))	MD mn0	开始计数和中断的设定
<ul style="list-style-type: none"> <li>间隔定时器模式 (0、0、0)</li> <li>捕捉模式 (0、1、0)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断 (定时器的输出也发生变化)。
<ul style="list-style-type: none"> <li>事件计数器模式 (0、1、1)</li> <li>单次计数模式<sup>注2</sup> (1、0、0)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	计数运行中的开始触发无效。此时不产生中断。
<ul style="list-style-type: none"> <li>捕捉 &amp; 单次计数模式 (1、1、0)</li> </ul>	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	计数运行中的开始触发有效 <sup>注3</sup> 。此时不产生中断。
上述以外		禁止设定。

注 1. bit11 是只读位, 固定为“0”, 忽视写操作。

2. 在单次计数模式中, 不控制开始计数时的中断输出 (INTTMRmn) 和 TMRmn 输出。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

### 6.3.4 定时器状态寄存器 mn (TSRmn)

TSRmn 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSRmn 寄存器只在捕捉模式 (MDmn3~MDmn1=010B) 和捕捉&单次计数模式 (MDmn3~MDmn1=110B) 中有效。有关各运行模式中的 OVF 位的变化和置位 / 清除条件, 请参照表 6-4。

通过 16 位存储器操作指令读 TSRmn 寄存器。

能用 TSRmnL 并且通过 8 位存储器操作指令读 TSRmn 寄存器的低 8 位。

在产生复位信号后, TSRmn 寄存器的值变为“0000H”。

图 6-12 定时器状态寄存器 mn (TSRmn) 的格式

地址: F01A0H、F01A1H (TSR00) ~ F01A6H、F01A7H (TSR03)

复位后: 0000H

R

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

OVF	通道 n 的计数器上溢状态
0	没有发生上溢。
1	发生上溢。

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

表 6-4 各运行模式中的 OVF 位的变化和置位 / 清除条件

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

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

6.3.5 定时器通道允许状态寄存器 m（TE<sub>m</sub>）

TE<sub>m</sub> 寄存器是表示各通道定时器运行的允许或者停止状态的寄存器。

TE<sub>m</sub> 寄存器的各位对应定时器通道开始寄存器 m（TS<sub>m</sub>）和定时器通道停止寄存器 m（TT<sub>m</sub>）的各位。如果将 TS<sub>m</sub> 寄存器的各位置“1”，TE<sub>m</sub> 寄存器的对应位就被置“1”。如果将 TT<sub>m</sub> 寄存器的各位置“1”，就将其对应位清“0”。

通过 16 位存储器操作指令读 TE<sub>m</sub> 寄存器。

能用 TE<sub>m</sub>L 并且通过 1 位或者 8 位存储器操作指令读 TE<sub>m</sub> 寄存器的低 8 位。

在产生复位信号后，TE<sub>m</sub> 寄存器的值变为“0000H”。

图 6-13 定时器通道允许状态寄存器 m（TE<sub>m</sub>）的格式

地址: F01B0H、F01B1H (TE0)                      复位后: 0000H                      R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE <sub>m</sub>	0	0	0	0	TEH <sub>m</sub> 3	0	TEH <sub>m</sub> 1	0	0	0	0	0	TE <sub>m</sub> 3	TE <sub>m</sub> 2	TE <sub>m</sub> 1	TE <sub>m</sub> 0

TEH <sub>m3</sub>	通道 3 为 8 位定时器模式时的高 8 位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

TEH <sub>m1</sub>	通道 1 为 8 位定时器模式时的高 8 位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

TE <sub>m</sub> <sub>n</sub>	通道 n 的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

在通道 1 和通道 3 为 8 位定时器模式时，TE<sub>m</sub>1 和 TE<sub>m</sub>3 表示低 8 位定时器的运行允许或者停止状态。

备注    m: 单元号 (m=0)    n: 通道号 (n=0 ~ 3)

### 6.3.6 定时器通道开始寄存器 m (TSm)

TSm 寄存器是对定时器计数寄存器 mn (TCRmn) 进行初始化并且设定各通道计数运行开始的触发寄存器。

如果将各位置“1”，定时器通道允许状态寄存器 m (TEm) 的对应位就被置“1”。因为 TSmn 位、TSHm1 位和 TSHm3 位是触发位，所以如果变为运行允许状态 (TEmn、TEHm1、TEHm3=1)，就立即清除 TSmn 位、TSHm1 位和 TSHm3 位。

通过 16 位存储器操作指令设定 TSm 寄存器。

能用 TSmL 并且通过 1 位或者 8 位存储器操作指令设定 TSm 寄存器的低 8 位。

在产生复位信号后，TSm 寄存器的值变为“0000H”。

图 6-14 定时器通道开始寄存器 m (TSm) 的格式

地址: F01B2H、F01B3H (TS0)

复位后: 0000H

R/W

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

TSHm3	通道 3 为 8 位定时器模式时的高 8 位定时器的运行允许（开始）触发
0	没有触发。
1	将 TEHm3 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm3 寄存器的计数，就进入间隔定时器模式（参照“6.5.2 计数器的开始时序”的表 6-5）。

TSHm1	通道 1 为 8 位定时器模式时的高 8 位定时器的运行允许（开始）触发
0	没有触发。
1	将 TEHm1 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCRm1 寄存器的计数，就进入间隔定时器模式（参照“6.5.2 计数器的开始时序”的表 6-5）。

TSmn	通道 n 的运行允许（开始）触发
0	没有触发。
1	将 TEmn 位置“1”，进入计数允许状态。 计数允许状态下的 TCRmn 寄存器的计数开始因各运行模式而不同（参照“6.5.2 计数器的开始时序”的表 6-5）。 在通道 1 和通道 3 为 8 位定时器模式时， TSm1 和 TSm3 为低 8 位定时器的运行允许（开始）触发。

注意 1. 必须将 bit15 ~ 12、10、8 ~ 4 置“0”。

2. 在从不使用 TImn 引脚输入的功能切换到使用 TImn 引脚输入的功能时，从设定定时器模式寄存器 mn (TMRmn) 到将 TSmn (TSHm1、TSHm3) 位置“1”为止，需要以下期间的等待：

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

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

备注 1. TSm 寄存器的读取值总是“0”。

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.7 定时器通道停止寄存器 m（TTm）

TTm 寄存器是设定各通道计数停止的触发寄存器。

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

通过 16 位存储器操作指令设定 TTm 寄存器。

能用 TTmL 并且通过 1 位或者 8 位存储器操作指令设定 TTm 寄存器的低 8 位。

在产生复位信号后，TTm 寄存器的值变为“0000H”。

图 6-15 定时器通道停止寄存器 m（TTm）的格式

地址: F01B4H、F01B5H (TT0)

复位后: 0000H

R/W

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

TTHm3	通道 3 为 8 位定时器模式时的高 8 位定时器的运行停止触发
0	没有触发。
1	将 TEHm3 位清“0”，进入计数停止状态。

TTHm1	通道 1 为 8 位定时器模式时的高 8 位定时器的运行停止触发
0	没有触发。
1	将 TEHm1 位清“0”，进入计数停止状态。

TTmn	通道 n 的运行停止触发
0	没有触发。
1	将 TEMn 位清“0”，进入可停止计数的状态。 在通道 1 和通道 3 为 8 位定时器模式时，TTm1 和 TTm3 为低 8 位定时器的运行停止触发。

注意 必须将 bit15 ~ 12、10、8 ~ 4 置“0”。

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

6.3.8 定时器输入选择寄存器 0（TIS0）

TIS0 寄存器是选择单元 0 的通道 1 定时器输入的寄存器。  
通过 8 位存储器操作指令设定 TIS0 寄存器。  
在产生复位信号后，TIS0 寄存器的值变为“00H”。

图 6-16 定时器输入选择寄存器 0（TIS0）的格式

地址：F0074H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	通道 1 使用的定时器输入的选择
0	0	0	定时器输入引脚（TI01）的输入信号
0	0	1	
0	1	0	
0	1	1	
1	0	0	低速内部振荡器时钟（f <sub>IL</sub> ）
上述以外			禁止设定。

注意 选择的定时器输入的高低电平宽度必须大于等于 1/f<sub>MCK</sub>+10ns。



6.3.9 定时器输出允许寄存器 m（TOEm）

TOEm 寄存器是设定允许或者禁止各通道定时器输出的寄存器。

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

通过 16 位存储器操作指令设定 TOEm 寄存器。

能用 TOEmL 并且通过 1 位或者 8 位存储器操作指令设定 TOEm 寄存器的低 8 位。

在产生复位信号后，TOEm 寄存器的值变为“0000H”。

图 6-17 定时器输出允许寄存器 m（TOEm）的格式

地址: F01BAH、F01BBH (TOE0)      复位后: 0000H      R/W

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

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

注意    必须将 bit15 ~ 4 置“0”。

备注    m: 单元号（m=0）    n: 通道号（n=0 ~ 3）

6.3.10 定时器输出寄存器 m（TOM）

TOM 寄存器是各通道定时器输出的缓冲寄存器。

此寄存器各位的值从各通道定时器的输出引脚（TOMn）输出。

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

要将 P13/TI00/TO00、P16/TI01/TO01/INTP5、P33/TI02/TO02/SSI00、P12/TI03/TO03/INTP4/PCLBUZ0 引脚用作端口功能时，必须将相应的 TOMn 位置“0”。

通过 16 位存储器操作指令设定 TOM 寄存器。

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

在产生复位信号后，TOM 寄存器的值变为“0000H”。

图 6-18 定时器输出寄存器 m（TOM）的格式

地址：F01B8H、F01B9H（TO0）

复位后：0000H

R/W

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

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

注意 必须将 bit15 ~ 4 置“0”。

备注 m: 单元号（m=0） n: 通道号（n=0 ~ 3）

6.3.11 定时器输出电平寄存器 m (TOLm)

TOLm 寄存器是控制各通道定时器输出电平的寄存器。

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

通过 16 位存储器操作指令设定 TOLm 寄存器。

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

在产生复位信号后, TOLm 寄存器的值变为“0000H”。

图 6-19 定时器输出电平寄存器 m (TOLm) 的格式

地址: F01BCH、F01BDH (TOL0)														复位后: 0000H				R/W	
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOL m3	TOL m2	TOL m1	0			
TOL mn	通道 n 的定时器输出电平的控制																		
0	正逻辑输出 (高电平有效)																		
1	负逻辑输出 (低电平有效)																		

注意 必须将 bit15 ~ 4 和 bit0 置“0”。

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

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.3.12 定时器输出模式寄存器 m（TOMm）

TOMm 寄存器是控制各通道定时器输出模式的寄存器。

当用作独立通道运行功能时，将所用通道的对应位置“0”。

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

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

通过 16 位存储器操作指令设定 TOMm 寄存器。

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

在产生复位信号后，TOMm 寄存器的值变为“0000H”。

图 6-20 定时器输出模式寄存器 m（TOMm）的格式

地址：F01BEH、F01BFH（TOM0）

复位后：0000H

R/W

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

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

注意 必须将 bit15 ~ 4 和 bit0 置“0”。

备注 m: 单元号（m=0）

n: 通道号

n=0~3（主控通道时：n=0、2）

p: 从属通道号

n < p ≤ 3

（有关主控通道和从属通道关系的详细内容，请参照“6.4.1 多通道联动运行功能的基本规则”）

6.3.13 噪声滤波器允许寄存器 1（NFEN1）

NFEN1 寄存器设定噪声滤波器是否用于各通道定时器输入引脚的输入信号。

对于需要消除噪声的引脚，必须将对应的位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，通过 CPU/ 外围硬件时钟（f<sub>MCK</sub>）与 2 个时钟的一致检测进行同步；当噪声滤波器无效时，只通过 CPU/ 外围硬件时钟（f<sub>MCK</sub>）进行同步注。

通过 1 位或者 8 位存储器操作指令设定 NFEN1 寄存器。

在产生复位信号后，NFEN1 寄存器的值变为“00H”。

注 详细内容请参照“6.5.1 (2) 选择 TImn 引脚输入信号的有效边沿的情况（CCSmn=1）”和“6.5.2 计数器的开始时序”。

图 6-21 噪声滤波器允许寄存器 1（NFEN1）的格式

地址: F0071H

复位后: 00H

R/W

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

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

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

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

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

### 6.3.14 端口模式寄存器 1、3（PM1、PM3）

这是以位为单位设定端口 1、3 的输入 / 输出的寄存器。

在将定时器输出引脚的复用端口（P13/TO00/TI00、P33/TO02/TI02/SSI00 等）用作定时器的输出时，必须将各端口对应的端口模式寄存器（PM<sub>xx</sub>）的位和端口寄存器（P<sub>xx</sub>）的位置“0”。

例）将 P33/TO02/TI02/SSI00 用作定时器输出的情况

将端口模式寄存器 3 的 PM33 位置“0”。

将端口寄存器 3 的 P33 位置“0”。

在将定时器输入引脚的复用端口（P13/TO00/TI00、P33/TO02/TI02/SSI00 等）用作定时器的输入时，必须将各端口对应的端口模式寄存器（PM<sub>xx</sub>）的位置“1”。此时，端口寄存器（P<sub>xx</sub>）的位可以是“0”或者“1”。

例）将 P33/TO02/TI02/SSI00 用作定时器输入的情况

将端口模式寄存器 3 的 PM33 位置“1”。

将端口寄存器 3 的 P33 位置“0”或者“1”。

通过 1 位或者 8 位存储器操作指令设定 PM1 寄存器和 PM3 寄存器。

在产生复位信号后，这些寄存器的值变为“FFH”。

图 6-22 端口模式寄存器 1、3（PM1、PM3）的格式（32 引脚产品）

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	PM35	PM34	PM33	PM32	PM31	PM30		

PM <sub>m</sub> <sub>n</sub>	P <sub>m</sub> <sub>n</sub> 引脚的输入 / 输出模式的选择（m=1、3，n=0～7）									
0	输出模式（输出缓冲器 ON）									
1	输入模式（输出缓冲器 OFF）									

## 6.4 定时器阵列单元的基本规则

### 6.4.1 多通道联动运行功能的基本规则

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

多通道联动运行功能的基本规则如下所示。

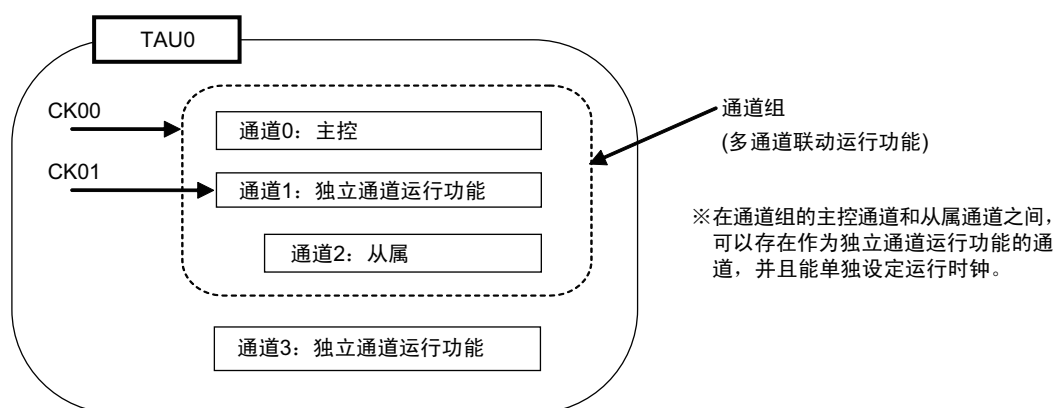
- (1) 只能将偶数通道（通道0、通道2）设定为主控通道。
- (2) 通道0以外的任何通道都能设定为从属通道。
- (3) 只能将主控通道的低位通道设定为从属通道。  
例 在将通道2设定为主控通道时，能将通道3设定为从属通道。
- (4) 能对1个主控通道设定多个从属通道。
- (5) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。  
例 在将通道0设定为主控通道时，能将通道1～3设定为主控通道0的从属通道。
- (6) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的CKSmn0位和CKSmn1位（定时器模式寄存器mn（TMRmn）的bit15和bit14）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

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

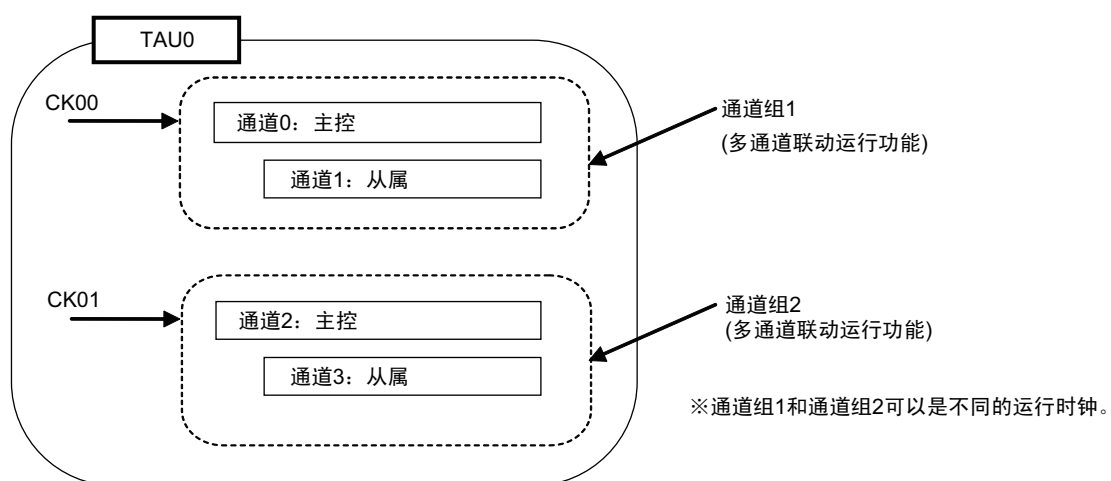
如果设定2个或者更多的相互不联动的通道群，就在通道群之间不适用上述的基本规则。

备注 m: 单元号（m=0） n: 通道号（n=0～3）

例 1



例 2





### 6.4.2 8位定时器运行功能的基本规则（只限于通道1和通道3）

8位定时器运行功能是将16位定时器的通道用作2个8位定时器的通道的功能。  
只有通道1和通道3才能使用8位定时器运行功能，使用时需要遵守几个规则。

8位定时器运行功能的基本规则如下所示。

- (1) 8位定时器运行功能只适用于通道1和通道3。
- (2) 当用作8位定时器时，将定时器模式寄存器mn（TMRmn）的SPLIT位置“1”。
- (3) 高8位定时器能用作间隔定时器功能。
- (4) 在开始运行时，高8位定时器输出INTTMm1H/INTTMm3H（中断）（和MDmn0位为“1”的运行相同）。
- (5) 高8位定时器的运行时钟的选择取决于低位TMRmn寄存器的CKSmn1位和CKSmn0位的设定。
- (6) 对于高8位定时器，通过操作TSHm1/TSHm3位来开始通道的运行，并且通过操作TTHm1/TTHm3位来停止通道的运行。能通过TEHm1/TEHm3位确认通道的状态。
- (7) 低8位定时器的运行取决于TMRmn寄存器的设定，有以下3种支持低8位定时器运行的功能：
  - 间隔定时器功能
  - 外部事件计数器功能
  - 延迟计数功能
- (8) 对于低8位定时器，通过操作TSm1/TSm3位来开始通道的运行，并且通过操作TTm1/TTm3位来停止通道的运行。能通过TEm1/TEm3位确认通道的状态。
- (9) 在16位定时器运行时，TSHm1/TSHm3/TTHm1/TTHm3位的操作无效。通过操作TSm1/TSm3位和TTm1/TTm3位使通道1和通道3运行。TEHm3位和TEHm1位不变。
- (10) 8位定时器功能不能使用联动运行功能（单触发脉冲、PWM和多重PWM）。

备注 m: 单元号 (m=0) n: 通道号 (n=1、3)

## 6.5 计数器的运行

### 6.5.1 计数时钟 ( $f_{TCLK}$ )

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

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

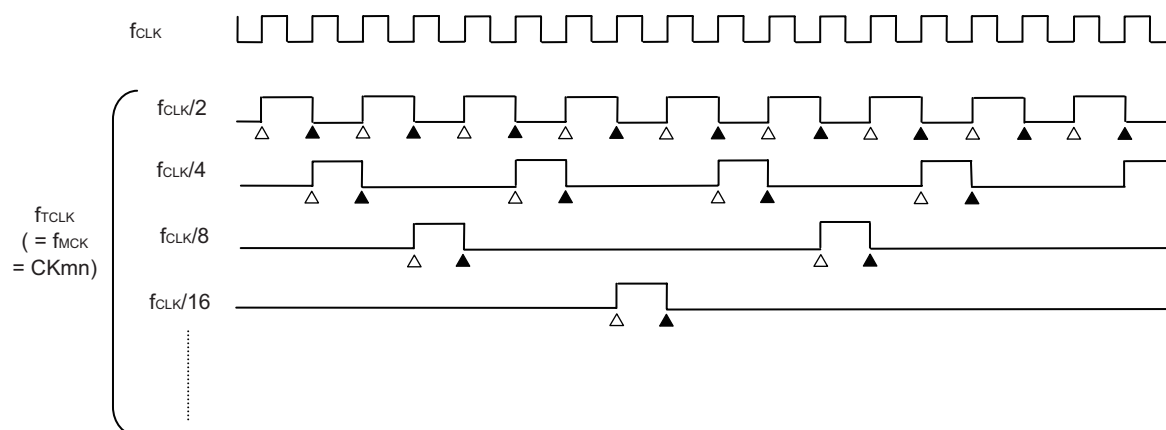
定时器阵列单元被设计为与  $f_{CLK}$  同步运行, 因此计数时钟 ( $f_{TCLK}$ ) 的时序如下。

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

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

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

图 6-23  $f_{CLK}$  和计数时钟 ( $f_{TCLK}$ ) 的时序 (CCSmn=0 的情况)



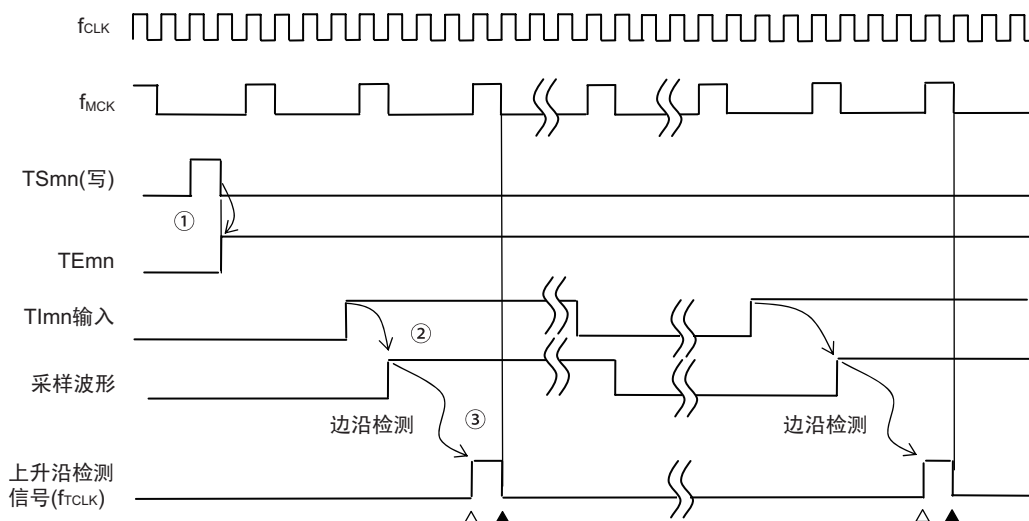
- 备注 1.  $\triangle$ : 计数时钟的上升沿  
 $\blacktriangle$ : 同步、计数器的递增 / 递减
2.  $f_{CLK}$ : CPU/ 外围硬件的时钟

## (2) 选择 TImn 引脚输入信号的有效边沿的情况 (CCSmn=1)

计数时钟 ( $f_{TCLK}$ ) 是检测 TImn 引脚输入信号的有效边沿并且与下一个  $f_{MCK}$  上升沿同步的信号。实际上，这是比 TImn 引脚的输入信号延迟了 1~2 个  $f_{MCK}$  时钟的信号 (在使用噪声滤波器时，延迟 3~4 个  $f_{MCK}$  时钟)。

为了取得与  $f_{CLK}$  的同步，定时器计数寄存器 mn (TCRmn) 从计数时钟的上升沿开始延迟 1 个  $f_{CLK}$  时钟后进行计数，出于方便而将其称为“在 TImn 引脚输入信号的有效边沿进行计数”。

图 6-24 计数时钟 ( $f_{TCLK}$ ) 的时序 (CCSmn=1, 未使用噪声滤波器的情况)



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

备注 1. △: 计数时钟的上升沿  
▲: 同步、计数器的递增 / 递减

2.  $f_{CLK}$ : CPU/ 外围硬件时钟  
 $f_{MCK}$ : 通道 n 的运行时钟
3. 输入脉冲间隔的测量、输入信号高低电平的测量、延迟计数器和单触发脉冲输出功能的 TImn 输入也是同样的波形。

## 6.5.2 计数器的开始时序

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

从计数允许状态到定时器计数寄存器  $mn$  (TCR $mn$ ) 开始计数为止的运行如表 6-5 所示。

表 6-5 从计数允许状态到定时器计数寄存器  $mn$  (TCR $mn$ ) 开始计数为止的运行

定时器的运行模式	将 TS $mn$ 位置“1”后的运行
• 间隔定时器模式	从检测到开始触发 (TS $mn$ =1) 到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将 TDR $mn$ 寄存器的值装入 TCR $mn$ 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“6.5.3 (1) 间隔定时器模式的运行”)。
• 事件计数器模式	通过给 TS $mn$ 位写“1”, 将 TDR $mn$ 寄存器的值装入 TCR $mn$ 寄存器。 如果检测到 TI $mn$ 的输入边沿, 就通过后续的计数时钟进行递减计数 (参照“6.5.3 (2) 事件计数器模式的运行”)。
• 捕捉模式	从检测到开始触发 (TS $mn$ =1) 到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将“0000H”装入 TCR $mn$ 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“6.5.3 (3) 捕捉模式的运行 (输入脉冲的间隔测量)”)。
• 单次计数模式	通过在定时器停止运行 (TE $mn$ =0) 的状态下给 TS $mn$ 位写“1”, 进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将 TDR $mn$ 寄存器的值装入 TCR $mn$ 寄存器, 并且通过后续的计数时钟进行递减计数 (参照“6.5.3 (4) 单次计数模式的运行”)。
• 捕捉 & 单次计数模式	通过在定时器停止运行 (TE $mn$ =0) 的状态下给 TS $mn$ 位写“1”, 进入开始触发的等待状态。 从检测到开始触发到产生计数时钟为止, 不执行任何操作。 通过第 1 个计数时钟将“0000H”装入 TCR $mn$ 寄存器, 并且通过后续的计数时钟进行递增计数 (参照“6.5.3 (5) 捕捉 & 单次计数模式中的运行 (测量高电平宽度)”)。

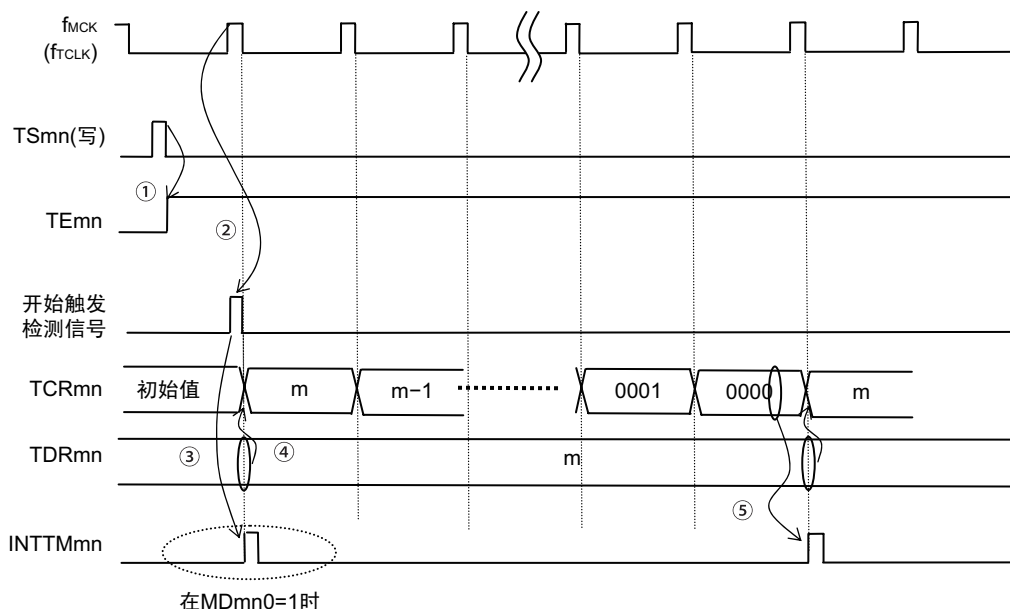
### 6.5.3 计数器的运行

以下说明各模式的计数器运行。

#### (1) 间隔定时器模式的运行

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

图 6-25 运行时序（间隔定时器模式）



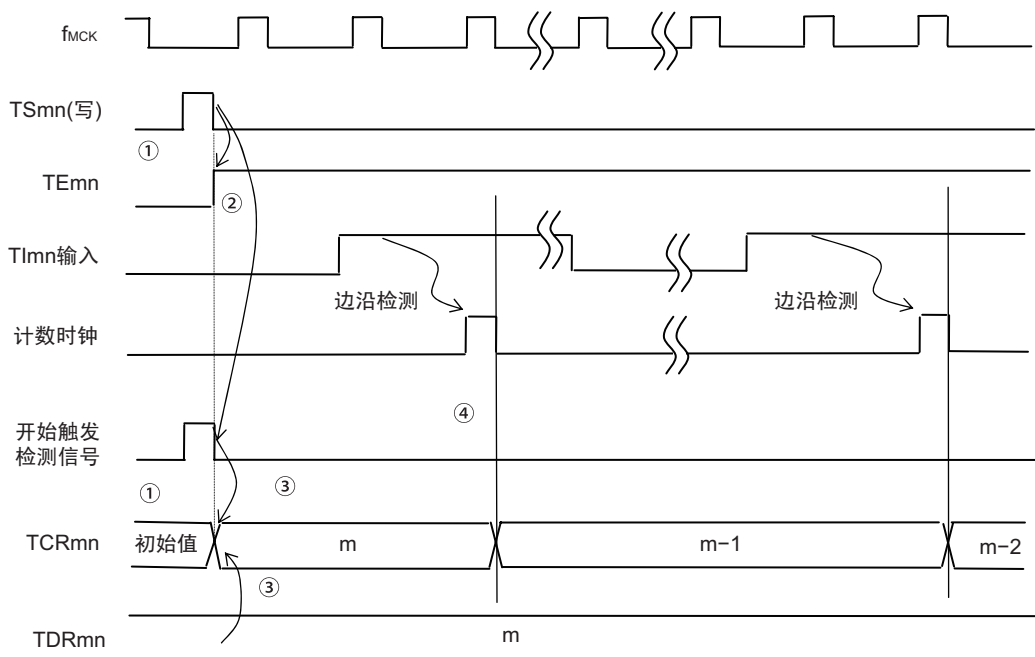
**注意** 因为第 1 个计数时钟周期的运行在写 TSmn 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MDmn0 位置“1”，以便能在开始计数时产生中断。

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

## (2) 事件计数器模式的运行

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

图 6-26 运行时序 (事件计数器模式)

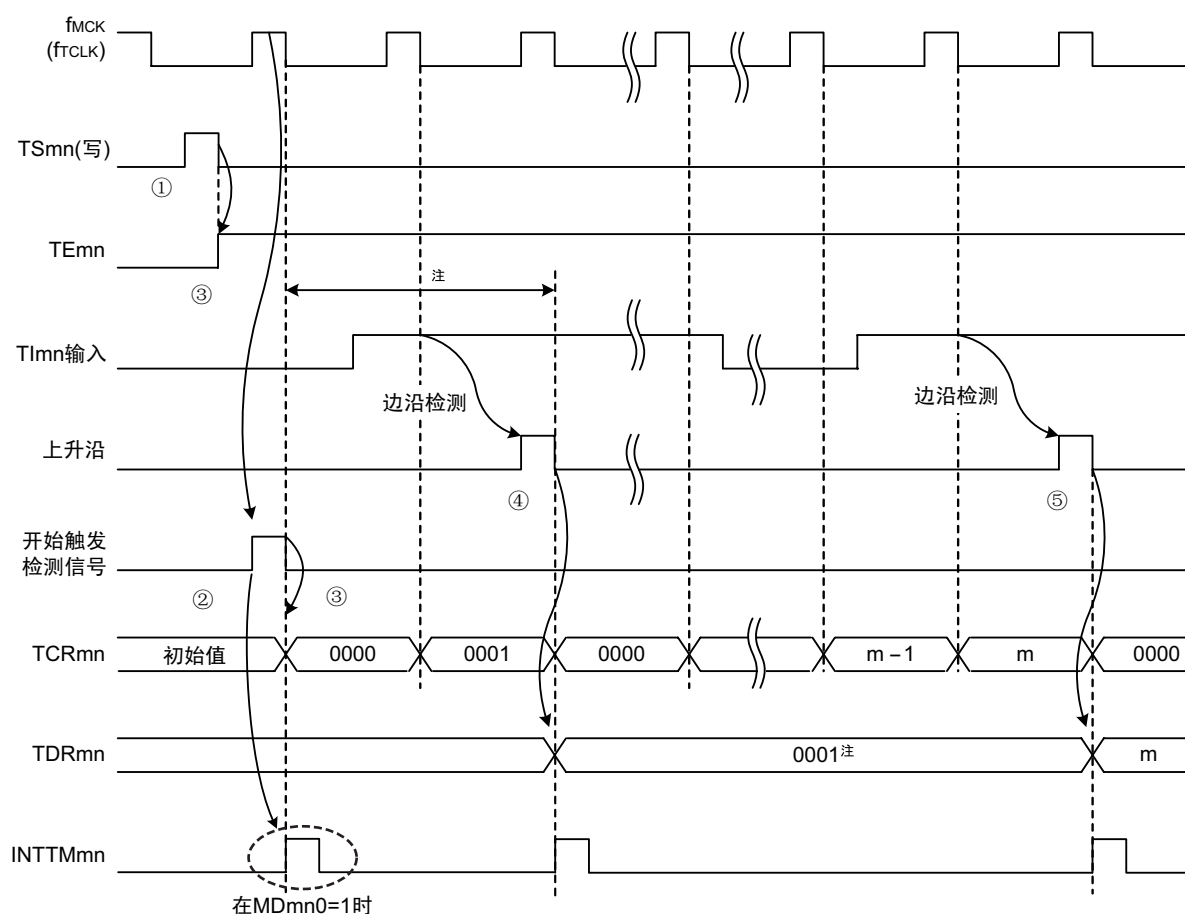


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从  $TI_{mn}$  输入开始再延迟 2 个  $f_{MCK}$  周期 (合计 3 ~ 4 个周期)。1 个周期的误差是因为  $TI_{mn}$  输入与计数时钟 ( $f_{MCK}$ ) 不同步。

## (3) 捕捉模式的运行（输入脉冲的间隔测量）

- ① 通过给  $TSmn$  位写“1”，进入运行允许状态（ $TEmn=1$ ）。
- ② 定时器计数寄存器  $mn$ （ $TCRmn$ ）保持初始值，直到产生计数时钟为止。
- ③ 通过允许运行后的第 1 个计数时钟（ $f_{MCK}$ ）产生开始触发信号。然后，将“0000H”装入  $TCRmn$  寄存器并且以捕捉模式开始计数（当  $MDmn0$  位为“1”时，通过开始触发信号产生  $INTTMmn$ ）。
- ④ 如果检测到  $TI_{mn}$  输入的有效边沿，就将  $TCRmn$  寄存器的值捕捉到  $TDRmn$  寄存器，并且产生  $INTTMmn$  中断。此时的捕捉值没有意义。 $TCRmn$  寄存器从“0000H”开始继续进行计数。
- ⑤ 如果检测到下一个  $TI_{mn}$  输入的有效边沿，就将  $TCRmn$  寄存器的值捕捉到  $TDRmn$  寄存器，并且产生  $INTTMmn$  中断。

图 6-27 运行时序（捕捉模式：输入脉冲的间隔测量）



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

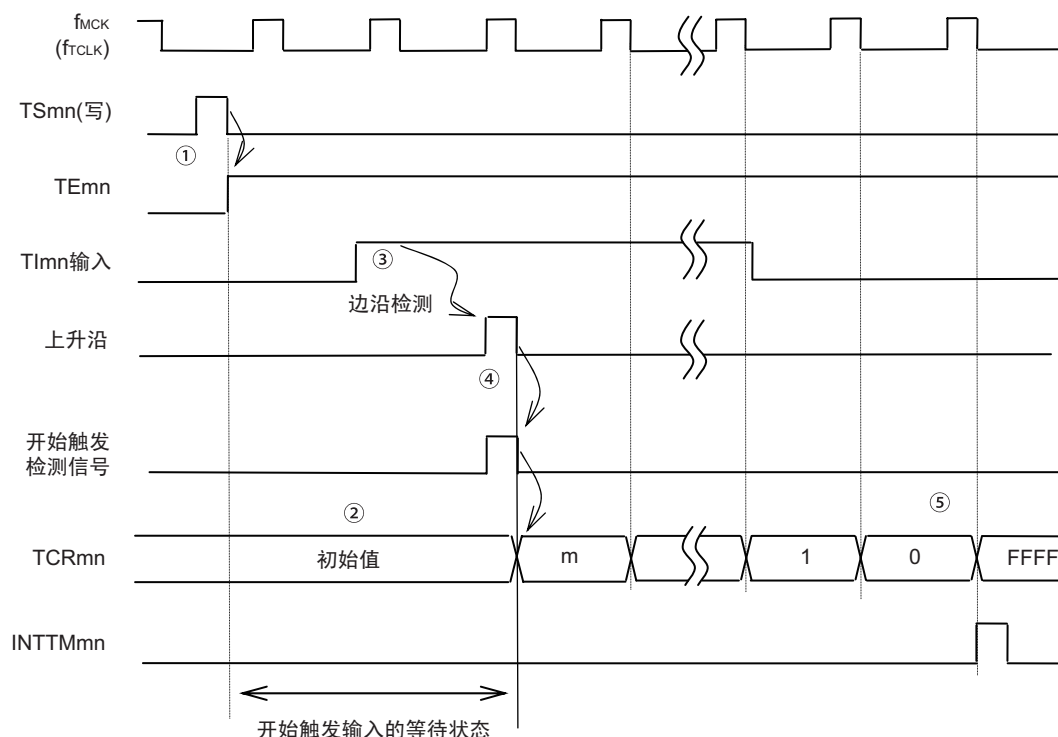
注意 因为第 1 个计数时钟周期的运行在写  $TSmn$  位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将  $MDmn0$  位置“1”，以便能在开始计数时产生中断。

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

## (4) 单次计数模式的运行

- ① 通过给  $TS_{mn}$  位写“1”，进入运行允许状态 ( $TE_{mn}=1$ )。
- ② 定时器计数寄存器  $mn$  ( $TCR_{mn}$ ) 保持初始值，直到产生开始触发信号为止。
- ③ 检测  $TI_{mn}$  输入的上升沿。
- ④ 在产生开始触发信号后将  $TDR_{mn}$  寄存器的值 ( $m$ ) 装入  $TCR_{mn}$  寄存器，并且开始计数。
- ⑤ 当  $TCR_{mn}$  寄存器递减计数到“0000H”时，产生  $INTT_{Mmn}$  中断，并且  $TCR_{mn}$  寄存器的值变为“FFFF”，停止计数。

图 6-28 运行时序（单次计数模式）



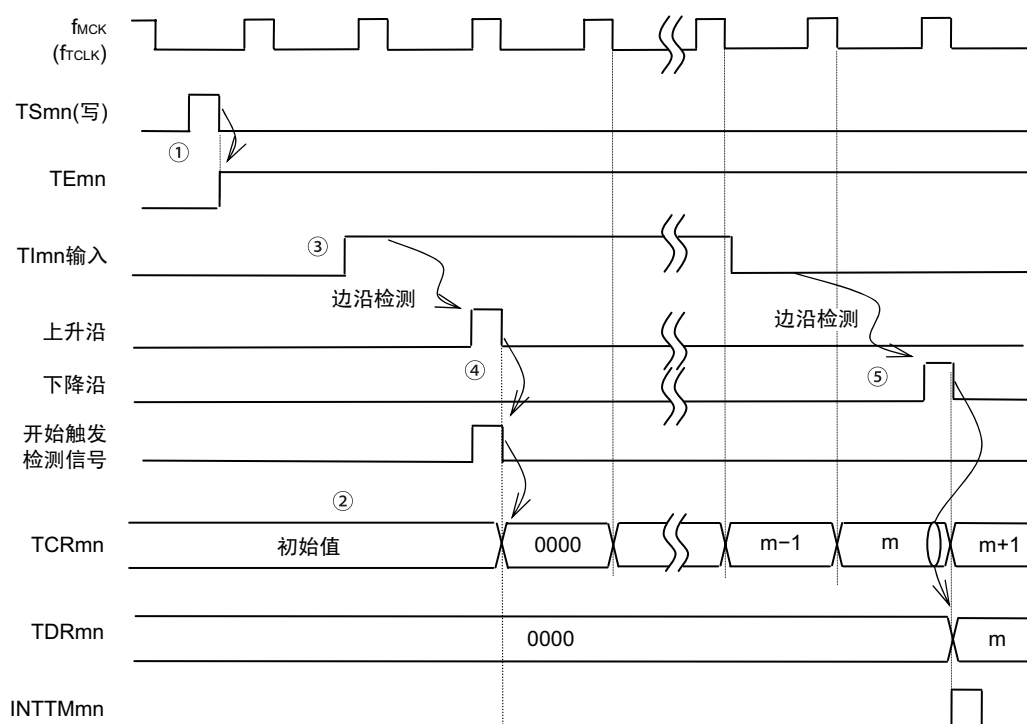
备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从  $TI_{mn}$  输入开始再延迟 2 个  $f_{MCK}$  周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为  $TI_{mn}$  输入与计数时钟 ( $f_{MCK}$ ) 不同步。



## (5) 捕捉 &amp; 单次计数模式中的运行（测量高电平宽度）

- ① 通过给定时器通道开始寄存器  $m$ （ $TSm$ ）的  $TSmn$  位写“1”，进入运行允许状态（ $TEmn=1$ ）。
- ② 定时器计数寄存器  $mn$ （ $TCRmn$ ）保持初始值，直到产生开始触发信号为止。
- ③ 检测  $TI_{mn}$  输入的上升沿。
- ④ 在产生开始触发信号后将“0000H”装入  $TCRmn$  寄存器，并且开始计数。
- ⑤ 如果检测到  $TI_{mn}$  输入的下降沿，就将  $TCRmn$  寄存器的值捕捉到  $TDRmn$  寄存器，并且产生  $INTTMmn$  中断。

图 6-29 运行时序（捕捉 &amp; 单次计数模式：测量高电平宽度）

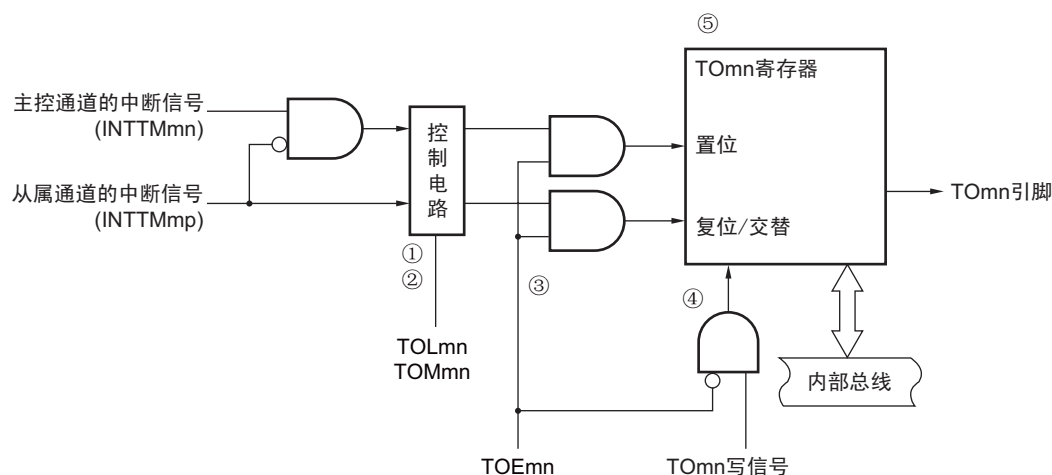


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从  $TI_{mn}$  输入开始再延迟 2 个  $f_{MCK}$  周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为  $TI_{mn}$  输入与计数时钟（ $f_{MCK}$ ）不同步。

## 6.6 通道输出（TOMn 引脚）的控制

### 6.6.1 TOMn 引脚输出电路的结构

图 6-30 输出电路的结构



以下说明 TOMn 引脚的输出电路。

- ① 当TOMmn位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器m（TOLm）的设定值，只将INTTMmp（从属通道定时器中断）传给定时器输出寄存器m（TOM）。
- ② 当TOMmn位为“1”（从属通道输出模式）时，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOM寄存器。  
此时，TOLm寄存器有效并且进行以下信号的控制：  
TOLmn=0时：正逻辑输出（INTTMmn→置位、INTTMmp→复位）  
TOLmn=1时：负逻辑输出（INTTMmn→复位、INTTMmp→置位）  
当同时产生INTTMmn和INTTMmp时（PWM输出的0%输出），优先INTTMmp（复位信号）而屏蔽INTTMmn（置位信号）。
- ③ 在允许定时器输出（TOEmn=1）的状态下，将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOM寄存器。TOM寄存器的写操作（TOMn写信号）无效。  
当TOEmn位为“1”时，除了中断信号以外，不改变TOMn引脚的输出。  
要对TOMn引脚的输出电平进行初始化时，需要在设定为禁止定时器输出（TOEmn=0）后给TOM寄存器写值。
- ④ 在禁止定时器输出（TOEmn=0）的状态下，对象通道的TOMn位的写操作（TOMn写信号）有效。当定时器输出为禁止状态（TOEmn=0）时，不将INTTMmn（主控通道定时器中断）和INTTMmp（从属通道定时器中断）传给TOM寄存器。
- ⑤ 能随时读TOM寄存器，并且能确认TOMn引脚的输出电平。

备注 m: 单元号（m=0）

n: 通道号

n=0~3（主控通道时：n=0、2）

p: 从属通道号

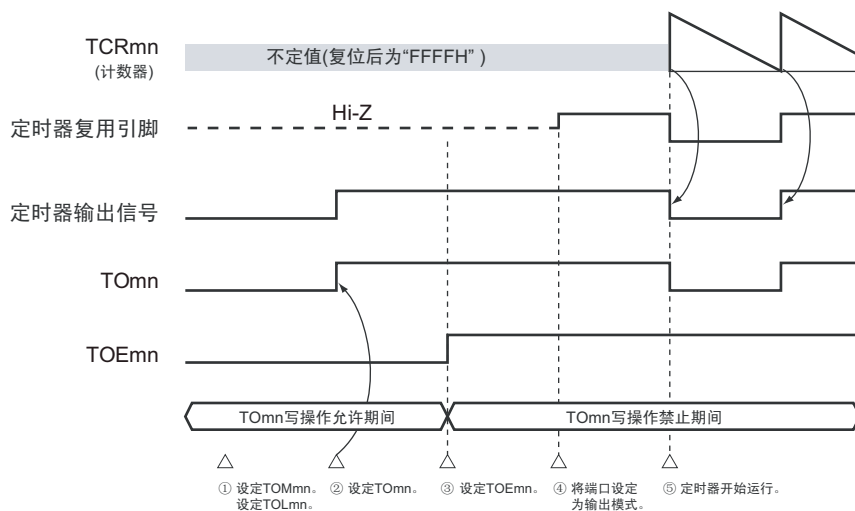
n=0: p=1、2、3

n=2: p=3

## 6.6.2 T0mn 引脚的输出设定

从 T0mn 输出引脚的初始设定到定时器开始运行的步骤和状态变化如下所示。

图 6-31 从设定定时器的输出到开始运行的状态变化



### ① 设定定时器输出的运行模式。

- T0Mmn位 (0: 主控通道输出模式、1: 从属通道输出模式)
- T0Lmn位 (0: 正逻辑输出、1: 负逻辑输出)

### ② 通过设定定时器输出寄存器m (T0m)，将定时器输出信号设定为初始状态。

### ③ 给T0Emn位写“1”，允许定时器输出 (禁止写T0m寄存器)。

### ④ 将端口的输入/输出设定为输出 (参照“6.3.14 端口模式寄存器1、3 (PM1、PM3)”)。

### ⑤ 允许定时器运行 (TSmn=1)。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

6.6.3 通道输出运行的注意事项

(1) 有关定时器运行中的 T<sub>Om</sub>、T<sub>OEm</sub>、T<sub>OLm</sub> 寄存器的设定值变更

定时器运行（定时器计数寄存器  $mn$ （TCR $mn$ ）和定时器数据寄存器  $mn$ （TDR $mn$ ）的运行）和 T<sub>Om $n$</sub>  输出电路相互独立。因此，定时器输出寄存器  $m$ （T<sub>Om</sub>）、定时器输出允许寄存器  $m$ （T<sub>OEm</sub>）和定时器输出电平寄存器  $m$ （T<sub>OLm</sub>）的设定值的变更不会影响定时器运行，能在定时器运行中更改设定值。但是，为了在各定时器的运行中从 T<sub>Om $n$</sub>  引脚输出期待的波形，必须设定为 6.8 和 6.9 所示的各运行的寄存器设定内容例子的值。

如果在产生各通道的定时器中断（INTT $Mmn$ ）信号前后更改除了 T<sub>Om</sub> 寄存器以外的 T<sub>OEm</sub> 寄存器和 T<sub>OLm</sub> 寄存器的设定值，就根据是在产生定时器中断（INTT $Mmn$ ）信号前更改还是在产生后更改，T<sub>Om $n$</sub>  引脚输出的波形可能不同。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

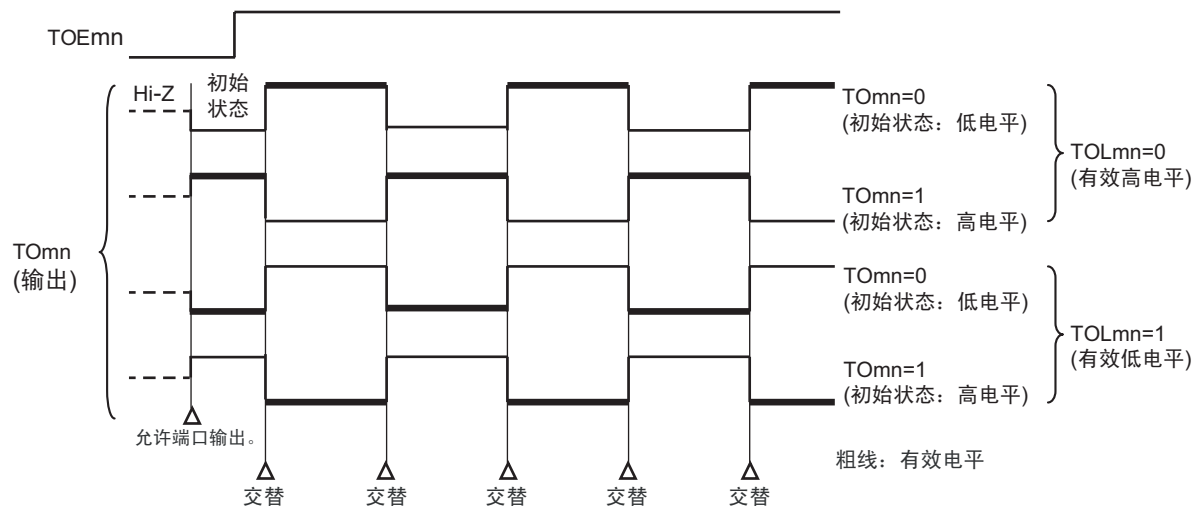
(2) 有关 T<sub>Om $n$</sub>  引脚的初始电平和定时器开始运行后的输出电平

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

(a) 在主导通道输出模式（T<sub>OMmn</sub>=0）中开始运行的情况

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

图 6-32 交替输出时（T<sub>OMmn</sub>=0）的 T<sub>Om $n$</sub>  引脚输出状态

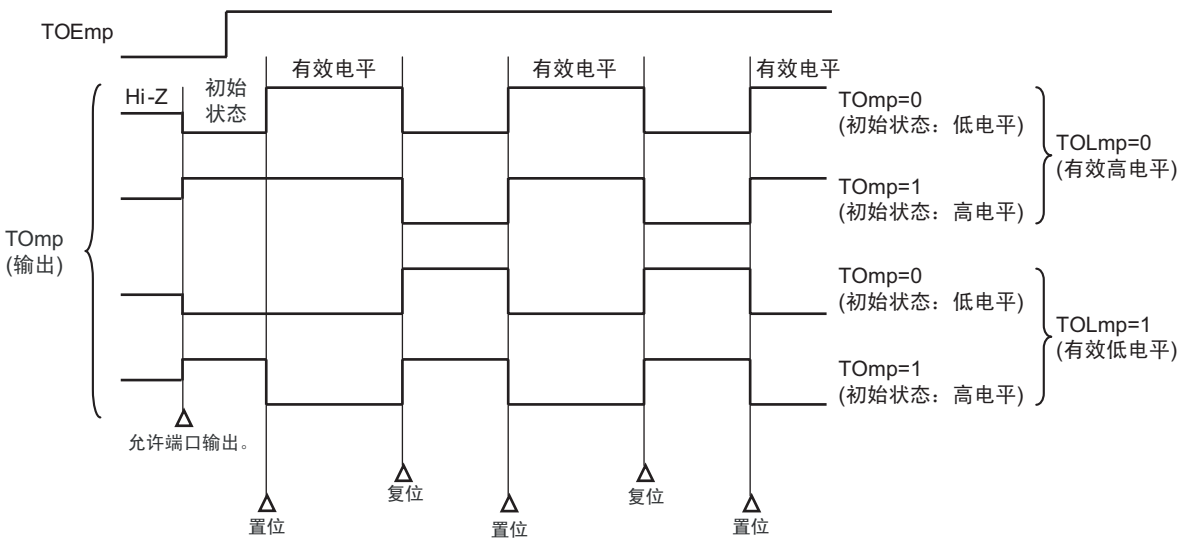


备注 1. 交替: 反相 T<sub>Om $n$</sub>  引脚的输出状态。  
2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

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

图 6-33 PWM 输出时（TOMmp=1）的 TOmp 引脚输出状态



备注 1. 置位：TOmp 引脚的输出信号从无效电平变为有效电平。

复位：TOmp 引脚的输出信号从有效电平变为无效电平。

2. m：单元号（m=0） p：通道号（p=1 ~ 3）

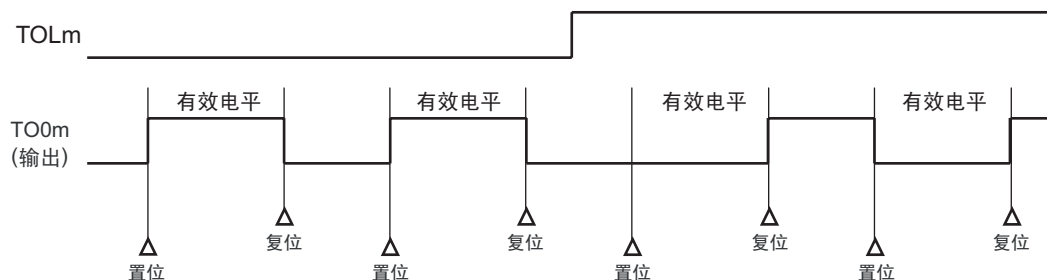
## (3) 有关从属通道输出模式 (TOMmn=1) 的 TOMn 引脚变化

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

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

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

图 6-34 在定时器运行中更改 TOLm 寄存器的内容时的运行



备注 1. 置位: TOMn 引脚的输出信号从无效电平变为有效电平。

复位: TOMn 引脚的输出信号从有效电平变为无效电平。

2. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

## (b) 置位 / 复位时序

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

当置位条件和复位条件同时产生时, 优先复位条件。

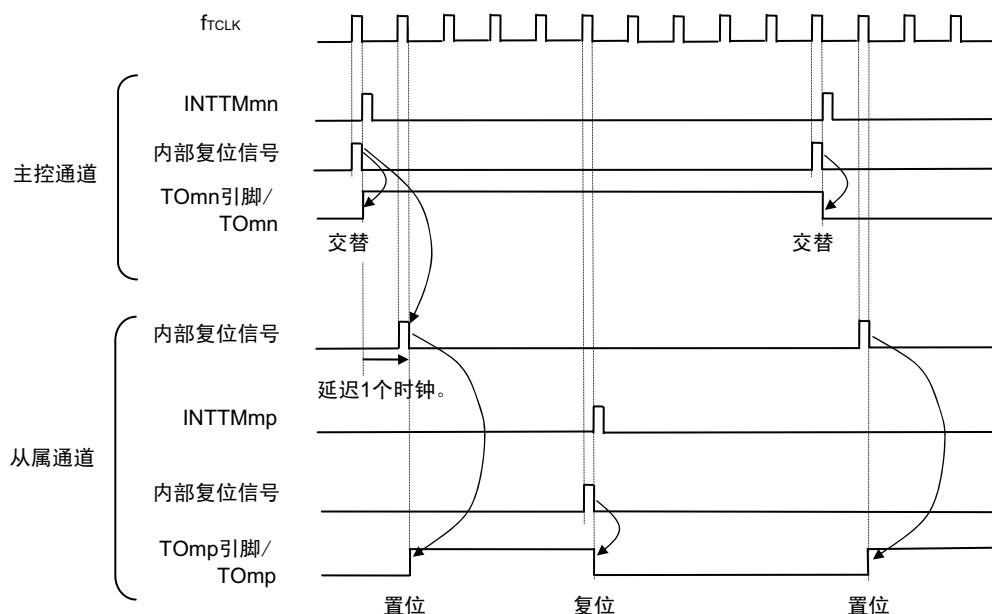
按照以下方法设定主控 / 从属通道时的置位 / 复位运行状态如图 6-35 所示。

主控通道: TOEmn=1、TOMmn=0、TOLmn=0

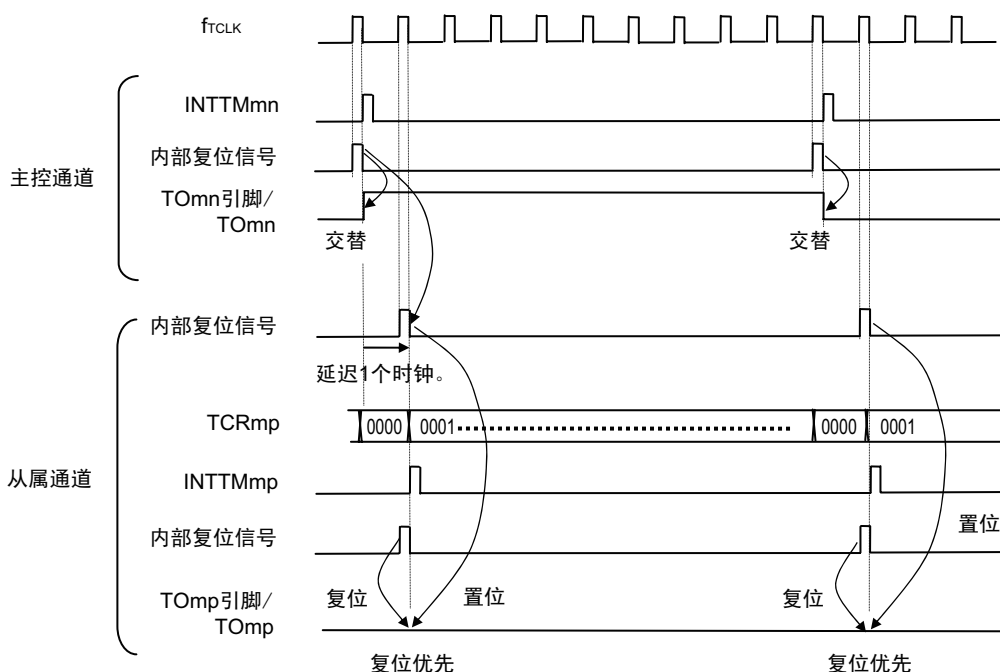
从属通道: TOEmp=1、TOMmp=1、TOLmp=0

图 6-35 置位 / 复位时序运行状态

## (1) 基本运行时序



## (2) 0% 占空比的运行时序



备注 1. 内部复位信号: TOn 引脚的复位 / 交替信号

内部置位信号: TOn 引脚的置位信号

2. m: 单元号 (m=0)

n: 通道号

n=0 ~ 3 (主控通道时: n=0、2)

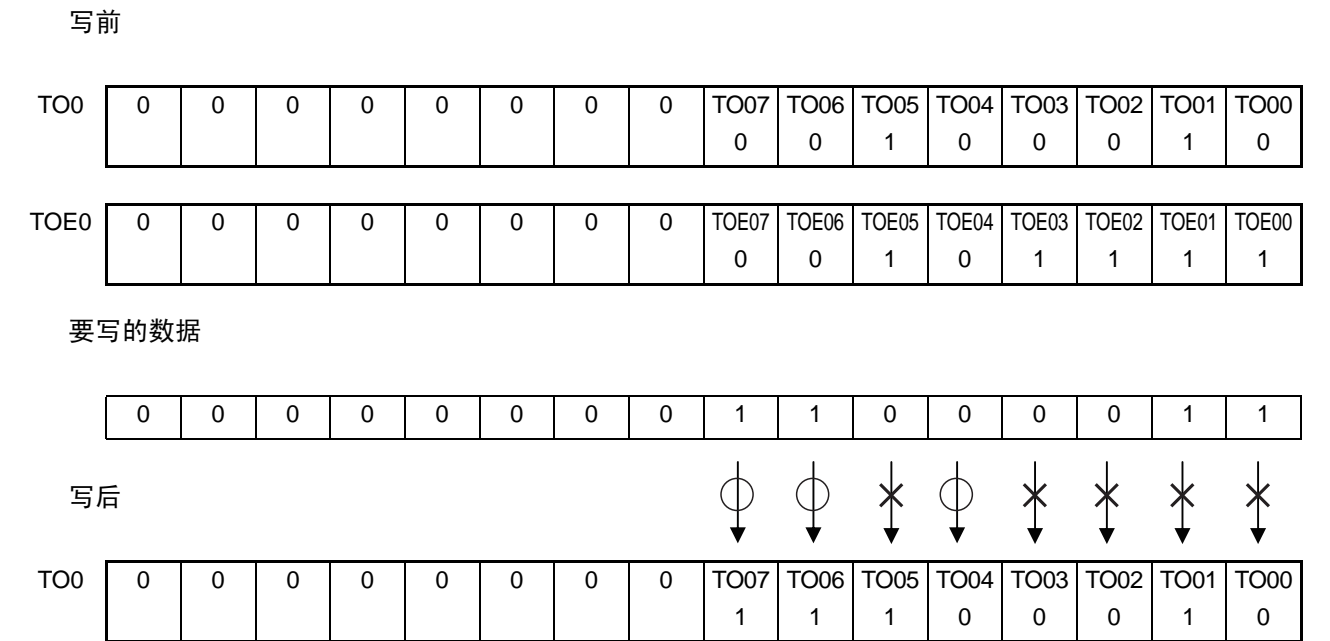
p: 从属通道号

n < p ≤ 3

6.6.4 TOmn 位的一次性操作

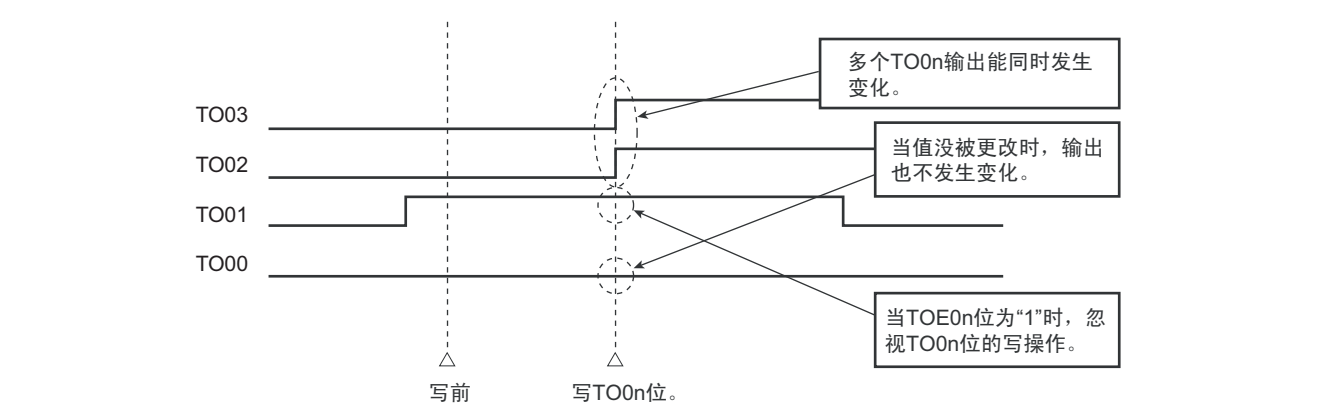
和定时器通道开始寄存器 m (TSm) 相同，定时器输出寄存器 m (TOm) 有全部通道的设定位 (TOmn)，因此能一次性地操作全部通道的 TOmn 位。另外，能给只想作为操作对象的通道输出 (TOmn) 的 TOmn 位写数据 (TOEmn=0)，因此也能操作任意的位。

图 6-36 TO0n 位的一次性操作例子



只能写 TOEmn 位为“0”的 TOmn 位，忽视 TOEmn 位为“1”的 TOmn 位的写操作。  
TOEmn 位为“1”的 TOmn（通道输出）不受写操作的影响，即使写 TOmn 位也被忽视，由定时器运行引起的输出变化正常进行。

图 6-37 一次性操作 TO0n 位时的 TO0n 引脚状态



注意 在允许定时器输出 (TOEmn=1) 的状态下，即使各通道的定时器中断 (INTTMmn) 引起的输出和 TOmn 位的写操作发生竞争，TOmn 引脚也能正常输出。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)



### 6.6.5 有关开始计数时的定时器中断和 TOn 引脚输出

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

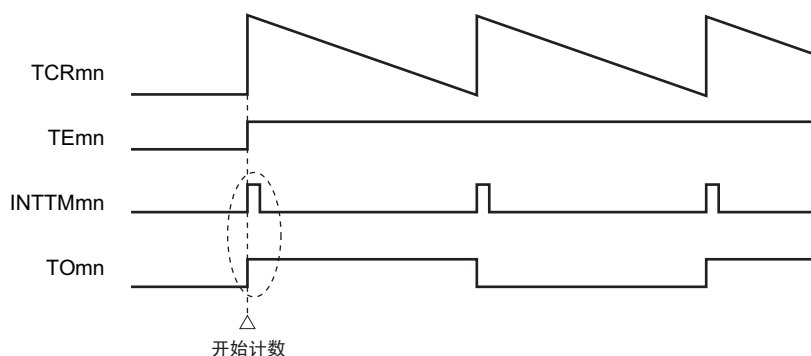
当 MDmn0 位为“1”时，能通过产生定时器中断（INTTMmn）得知计数的开始时序。

在其他模式中，不控制开始计数时的定时器中断和 TOn 输出。

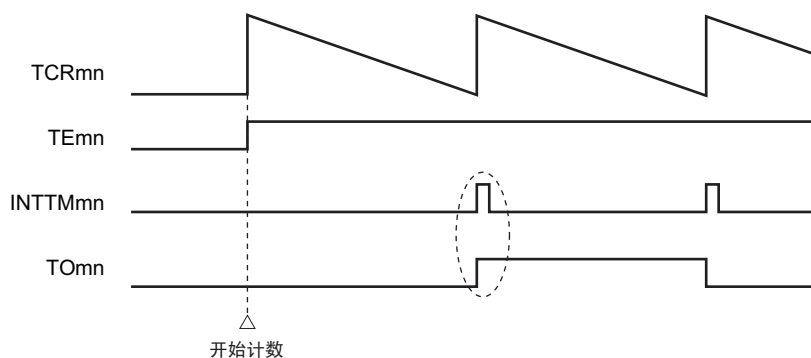
设定为间隔定时器模式（TOEmn=1、TOMmn=0）时的运行例子如下所示。

图 6-38 开始计数时的定时器中断和 TOn 输出的运行例子

#### (a) MDmn0 位为“1”的情况



#### (b) MDmn0 位为“0”的情况



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

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

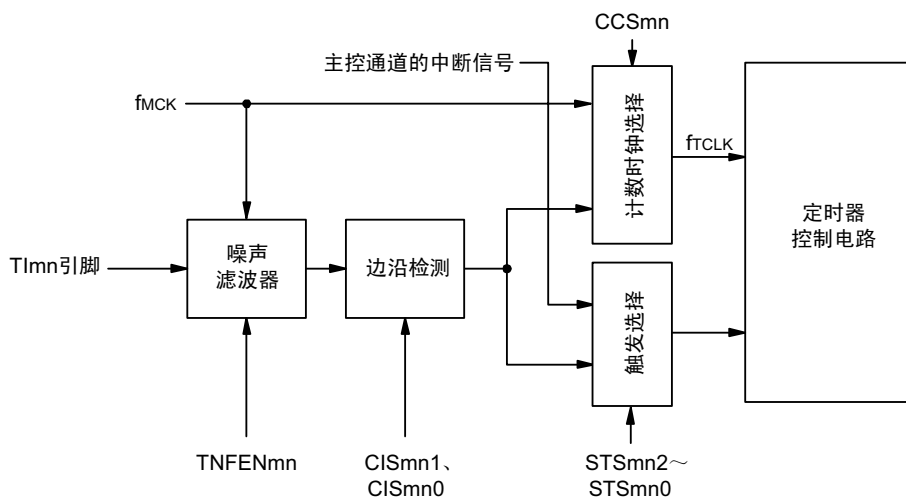
## 6.7 定时器输入 (Tlmn) 的控制

### 6.7.1 Tlmn 引脚输入电路的结构

定时器输入引脚的信号通过噪声滤波器和边沿检测电路输入到定时器控制电路。

对于需要消除噪声的引脚，必须将对应的引脚噪声滤波器置为有效。输入电路的结构如下所示。

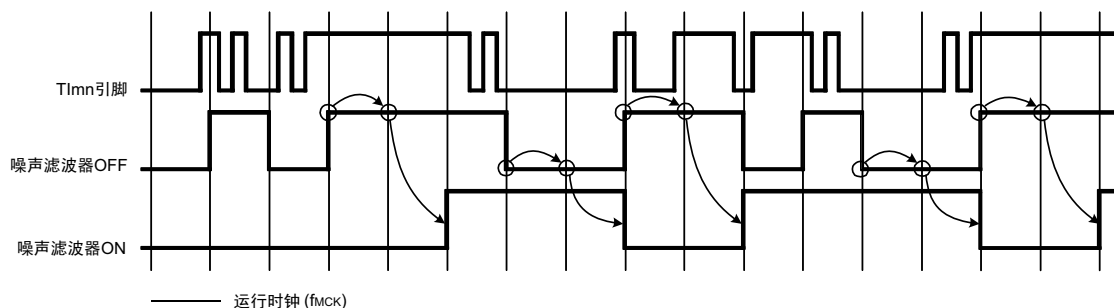
图 6-39 输入电路的结构



### 6.7.2 噪声滤波器

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

图 6-40 Tlmn 输入引脚在噪声滤波器 ON 或者 OFF 情况下的采样波形



**注意** Tlmn 引脚的输入波形用于说明噪声滤波器 ON 或者 OFF 的运行。实际使用时，必须按照“27.4 AC 特性”所示的 Tlmn 输入高低电平宽度进行输入。

### 6.7.3 操作通道输入时的注意事项

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

#### (1) 噪声滤波器为 OFF 的情况

如果在定时器模式寄存器  $mn$  (TMR $mn$ ) 的 bit12 (CCS $mn$ )、bit9 (STS $mn$ 1) 和 bit8 (STS $mn$ 0) 全都为“0”的状态下将任意一位置位，就必须至少在经过 2 个运行时钟 ( $f_{MCK}$ ) 周期后将定时器通道开始寄存器 (TS $m$ ) 的运行允许触发置位。

#### (2) 噪声滤波器为 ON 的情况

如果在定时器模式寄存器  $mn$  (TMR $mn$ ) 的 bit12 (CCS $mn$ )、bit9 (STS $mn$ 1) 和 bit8 (STS $mn$ 0) 全都为“0”的状态下将任意一位置位，就必须至少在经过 4 个运行时钟 ( $f_{MCK}$ ) 周期后将定时器通道开始寄存器 (TS $m$ ) 的运行允许触发置位。

## 6.8 定时器阵列单元的独立通道运行功能

### 6.8.1 作为间隔定时器 / 方波输出的运行

#### (1) 间隔定时器

能用作以固定间隔产生 INTTMmn（定时器中断）的基准定时器。

中断产生周期能用以下计算式进行计算：

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

#### (2) 作为方波输出的运行

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

TOmn 输出方波的周期和频率能用以下计算式进行计算：

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

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

在间隔定时器模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

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

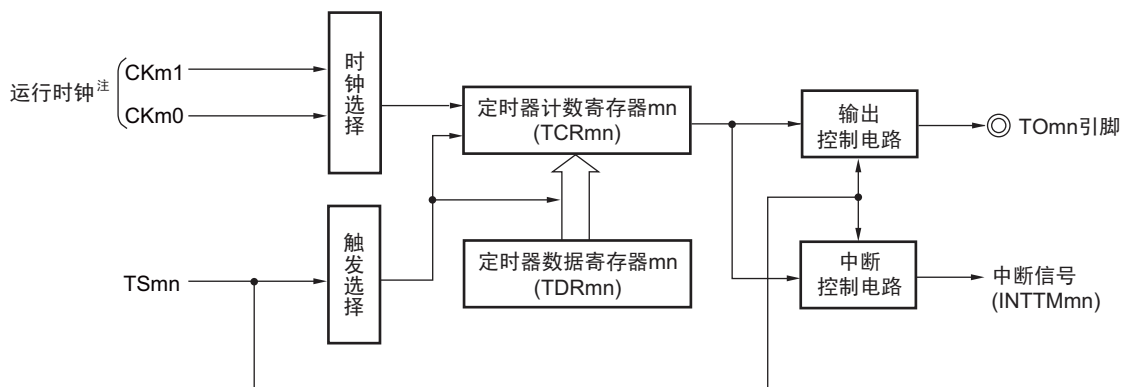
然后，TCRmn 寄存器通过计数时钟进行递减计数。

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

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

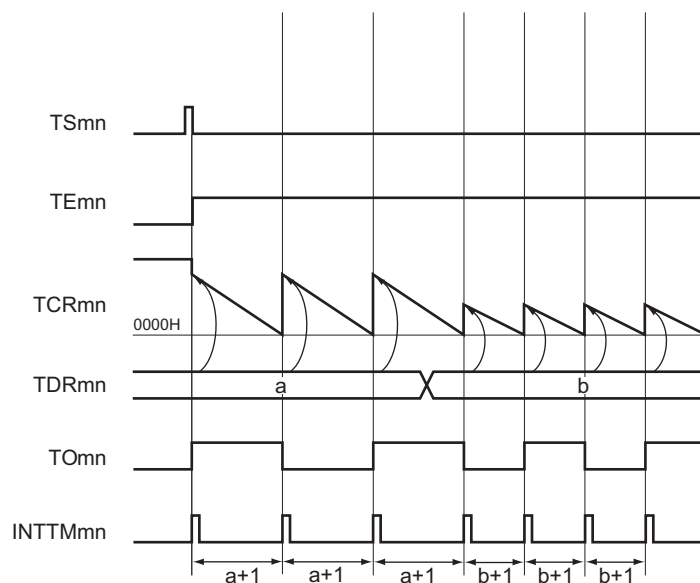
备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-41 作为间隔定时器 / 方波输出运行的框图



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

图 6-42 作为间隔定时器 / 方波输出运行的基本时序例子 (MDmn0=1)

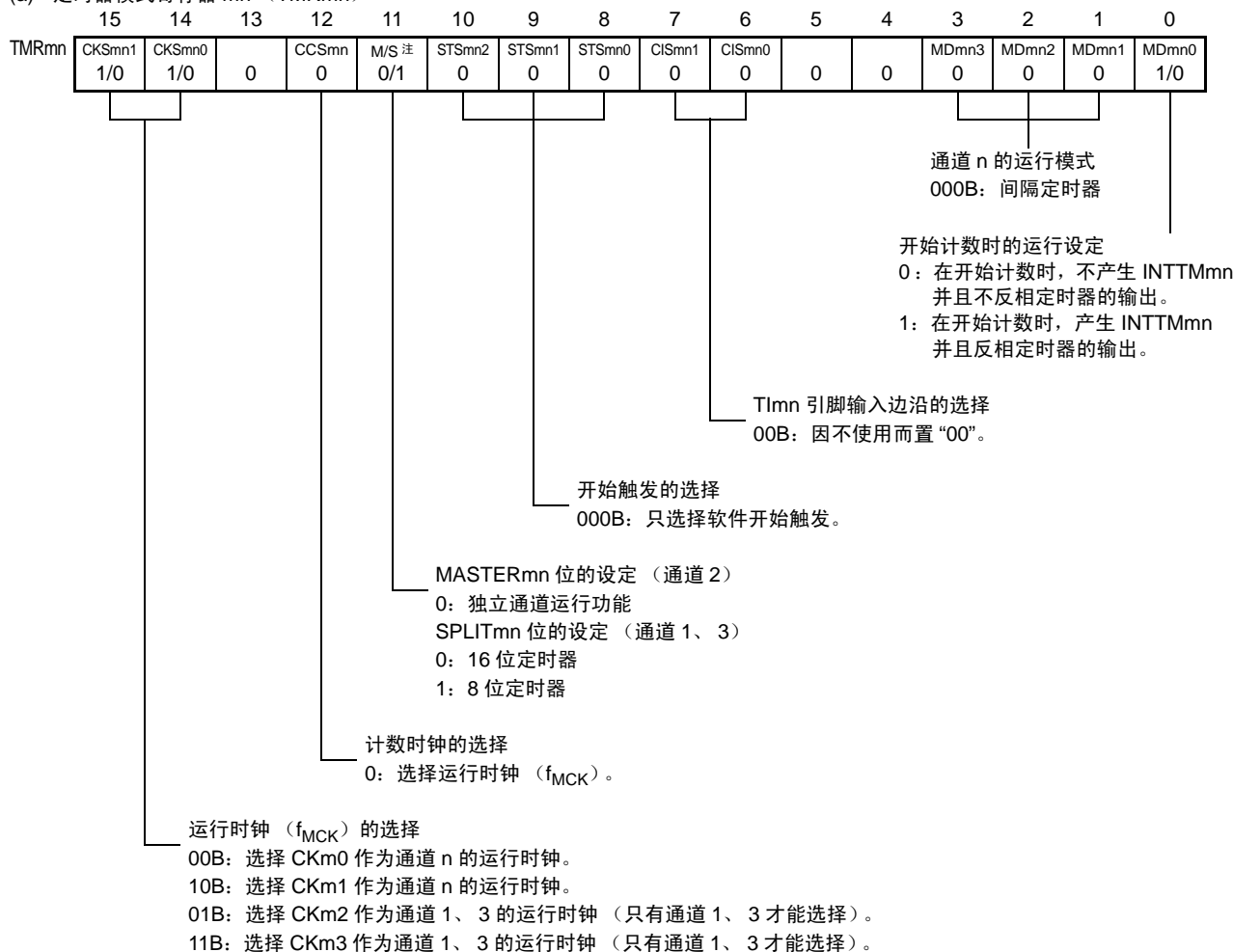


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

2. TSmn : 定时器通道开始寄存器 m (TSM) 的 bit n
- TEmn : 定时器通道允许状态寄存器 m (TEM) 的 bit n
- TCRmn : 定时器计数寄存器 mn (TCRmn)
- TDRmn : 定时器数据寄存器 mn (TDRmn)
- TOMn : TOMn 引脚输出信号

图 6-43 间隔定时器 / 方波输出时的寄存器设定内容例子

(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器输出寄存器 m (TOM)

bit n	
TOMn	0: 由 TOMn 输出“0”。
1/0	1: 由 TOMn 输出“1”。

(c) 定时器输出允许寄存器 m (TOEm)

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

(d) 定时器输出电平寄存器 m (TOLm)

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

(e) 定时器输出模式寄存器 m (TOMm)

bit n	
TOMmn	0: 设定主控通道输出模式。
0	

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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-44 间隔定时器 / 方波输出功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器 mn (TMRmn) (确定通道的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	使用 TOMn 输出的情况: 将定时器输出模式寄存器 m (TOMm) 的 TOMmn 位置“0” (主控通道输出模式)。 将 TOLmn 位置“0”。 设定 TOMn 位, 确定 TOMn 输出的初始电平。→ 将 TOEmn 位置“1”, 允许 TOMn 的运行。→ 将端口寄存器和端口模式寄存器置“0”。→	TOMn 引脚处于 Hi-Z 输出状态。  当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TOMn 初始设定的电平。 因为通道处于运行停止状态, 所以 TOMn 不变。 TOMn 引脚输出 TOMn 设定的电平。
重新 开始 运行	开始运行 (只在使用 TOMn 输出并且重新开始时, 将 TOEmn 位置“1”) 将 TSmn (TSHm1、TSHm3) 位置“1”。→ 因为 TSmn (TSHm1、TSHm3) 位是触发位, 所以自动返回到“0”。	TEmn (TEHm1、TEHm3) 位为“1”并且开始计数。 通过输入计数时钟, 将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn)。当 TMRmn 寄存器的 MDmn0 位为“1”时, 产生 INTTMmn 并且 TOMn 进行交替输出。
	运行中 禁止更改 TMRmn 寄存器、TOMmn 位和 TOLmn 位的设定值。 能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 能更改 TOM 寄存器和 TOEm 寄存器的设定值。	计数器 (TCRmn) 进行递减计数。如果计数到“0000H”, 就再次将 TDRmn 寄存器的值装入 TCRmn 寄存器, 继续进行计数。当检测到 TCRmn 为“0000H”时, 产生 INTTMmn 并且 TOMn 进行交替输出。此后, 重复此运行。
	停止运行 将 TTmn (TTHm1、TTHm3) 位置“1”。→ 因为 TTmn (TTHm1、TTHm3) 位是触发位, 所以自动返回到“0”。	TEmn (TEHm1、TEHmn) 位为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 TOMn 输出不被初始化而保持状态。
	将 TOEmn 位置“0”并且给 TOMn 位设定值。→	TOMn 引脚输出 TOMn 位设定的电平。

图 6-44 间隔定时器 / 方波输出功能时的操作步骤 (2/2)

	软件操作	硬件状态
TAU 停止	要保持 TOmn 引脚输出电平的情况： 在端口寄存器设定要保持的值后将 TOmn 位置“0”。 不需要保持 TOmn 引脚输出电平的情况： 不需要设定。	通过端口功能保持 TOmn 引脚的输出电平。
	将 PER0 寄存器的 TAUmEN 位置“0”。	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。 (TOmn 位变为“0”并且 TOmn 引脚变为端口功能)

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)



## 6.8.2 作为外部事件计数器的运行

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

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

在事件计数器模式中，定时器计数寄存器 mn（TCRmn）用作递减计数器。

通过将定时器通道开始寄存器 m（TSMn）的任意通道开始触发位（TSMn、TSHm1、TSHm3）置“1”，将定时器数据寄存器 mn（TDRmn）的值装入 TCRmn 寄存器。

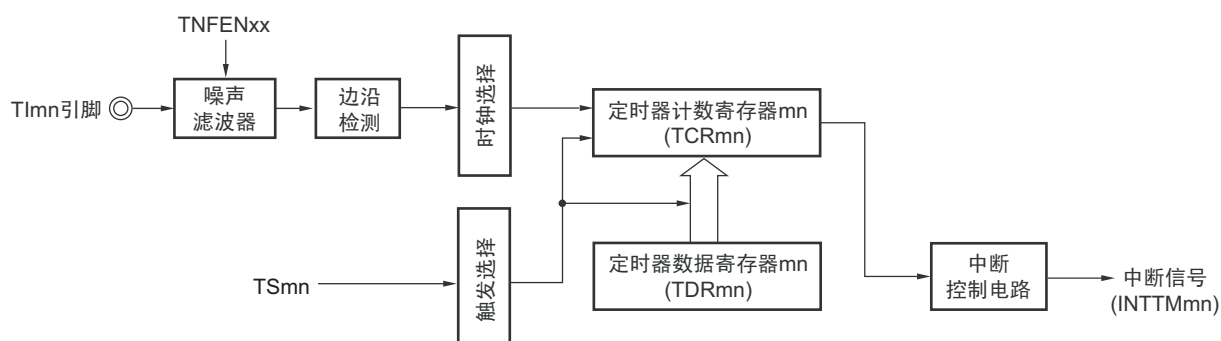
TCRmn 寄存器在检测到 TIMn 引脚输入的有效边沿的同时进行递减计数。如果 TCRmn 变为“0000H”，就再次装入 TDRmn 寄存器的值并且输出 INTTMmn。

此后，继续同样的运行。

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

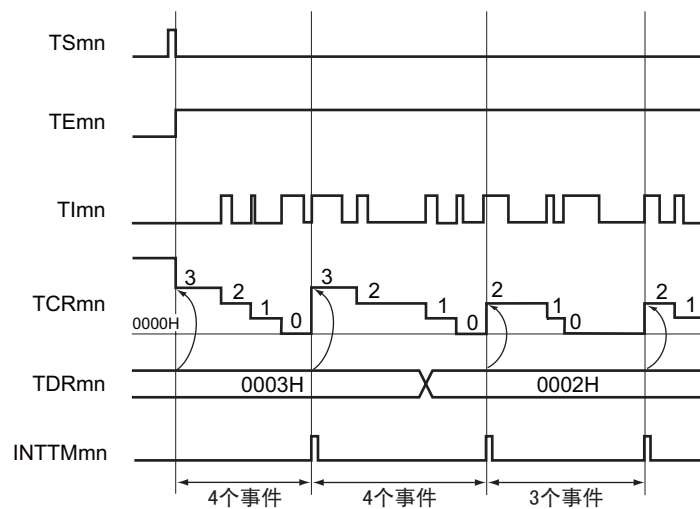
能随时改写 TDRmn 寄存器，改写的 TDRmn 寄存器的值在下一个计数期间有效。

图 6-45 作为外部事件计数器运行的框图



备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-46 作为外部事件计数器运行的基本时序例子

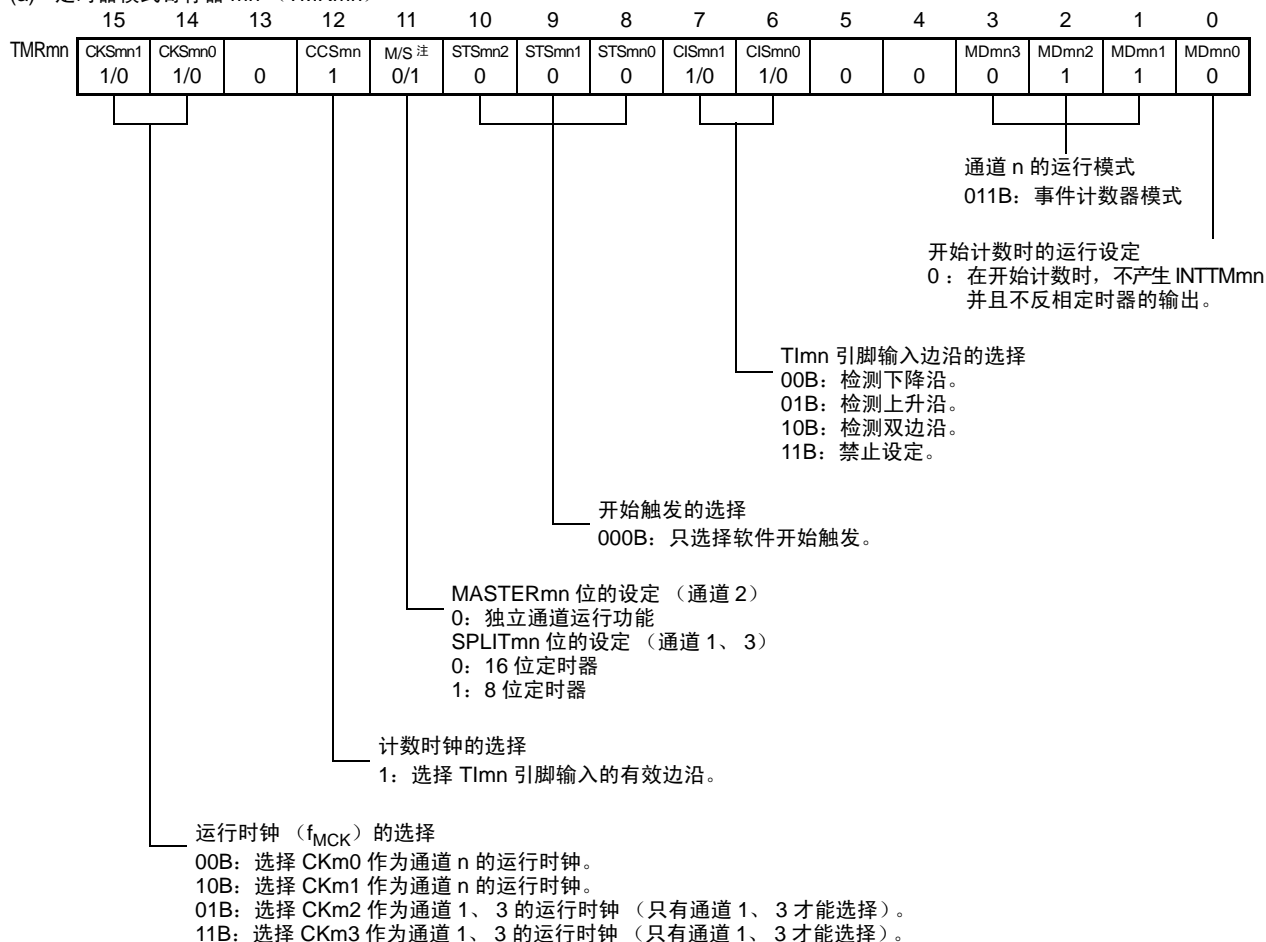


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

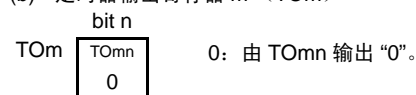
2. TSmn : 定时器通道开始寄存器 m (TSm) 的 bit n
- TE mn : 定时器通道允许状态寄存器 m (TEm) 的 bit n
- TImn : TImn 引脚输入信号
- TCRmn : 定时器计数寄存器 mn (TCRmn)
- TDRmn : 定时器数据寄存器 mn (TDRmn)

图 6-47 外部事件计数器模式时的寄存器设定内容例子

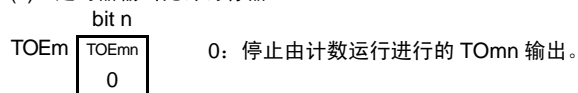
(a) 定时器模式寄存器 mn (TMRmn)



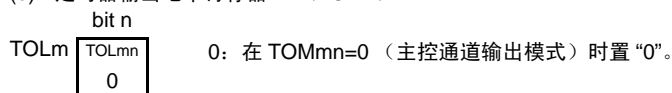
(b) 定时器输出寄存器 m (TOM)



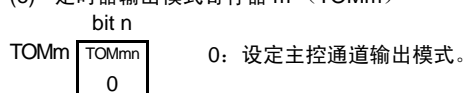
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-48 外部事件计数器功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 mn (TMRmn) (确定通道的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定计数值。 将定时器输出允许寄存器 m (TOEm) 的 TOEmn 位置“0”。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
重新 开始 运行	开始 运行	将 TSmn 位置“1”。————→ 因为 TSmn 位是触发位, 所以自动返回到“0”。 TEmn 位为“1”并且开始计数。 将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn), 进入 TImn 引脚输入边沿的检测等待状态。
	运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 禁止更改 TMRmn 寄存器、TOMmn 位、TOLmn 位、TOmn 位和 TOEmn 位的设定值。 每当检测到 TImn 引脚的输入边沿时, 计数器 (TCRmn) 就进行递减计数。如果计数到“0000H”, 就再次将 TDRmn 寄存器的值装入 TCRmn 寄存器, 继续进行计数。当检测到 TCRmn 为“0000H”时, 产生 INTTMmn。 此后, 重复此运行。
	停止 运行	将 TTmn 位置“1”。————→ 因为 TTmn 位是触发位, 所以自动返回到“0”。 TEmn 位为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
	TAU 停止	将 PER0 寄存器的 TAUmEN 位置“0”。————→ 定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

### 6.8.3 作为输入脉冲间隔测量的运行

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

脉冲间隔能用以下计算式进行计算：

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

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

在捕捉模式中，定时器计数寄存器 mn（TCRmn）用作递增计数器。

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

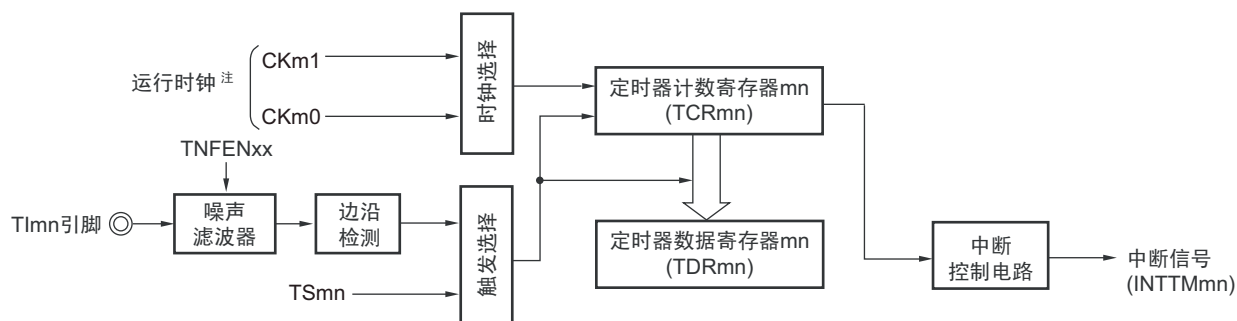
如果检测到 TImn 引脚输入的有效边沿，就将 TCRmn 寄存器的计数值传送（捕捉）到定时器数据寄存器 mn（TDRmn），同时将 TCRmn 寄存器清“0000H”，然后输出 INTTMmn。此时，如果计数器发生上溢，就将定时器状态寄存器 mn（TSRmn）的 OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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

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

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

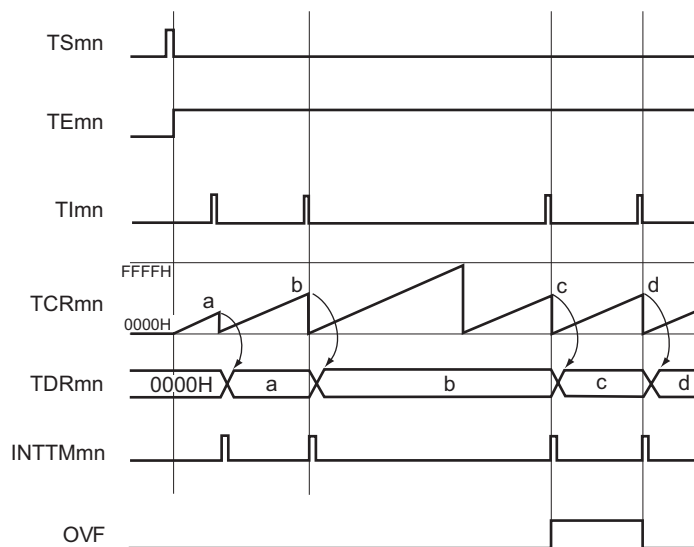
图 6-49 作为输入脉冲间隔测量运行的框图



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

**备注** m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-50 作为输入脉冲间隔测量的运行基本时序例子 (MDmn0=0)



备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

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

图 6-51 测量输入脉冲间隔时的寄存器设定内容例子

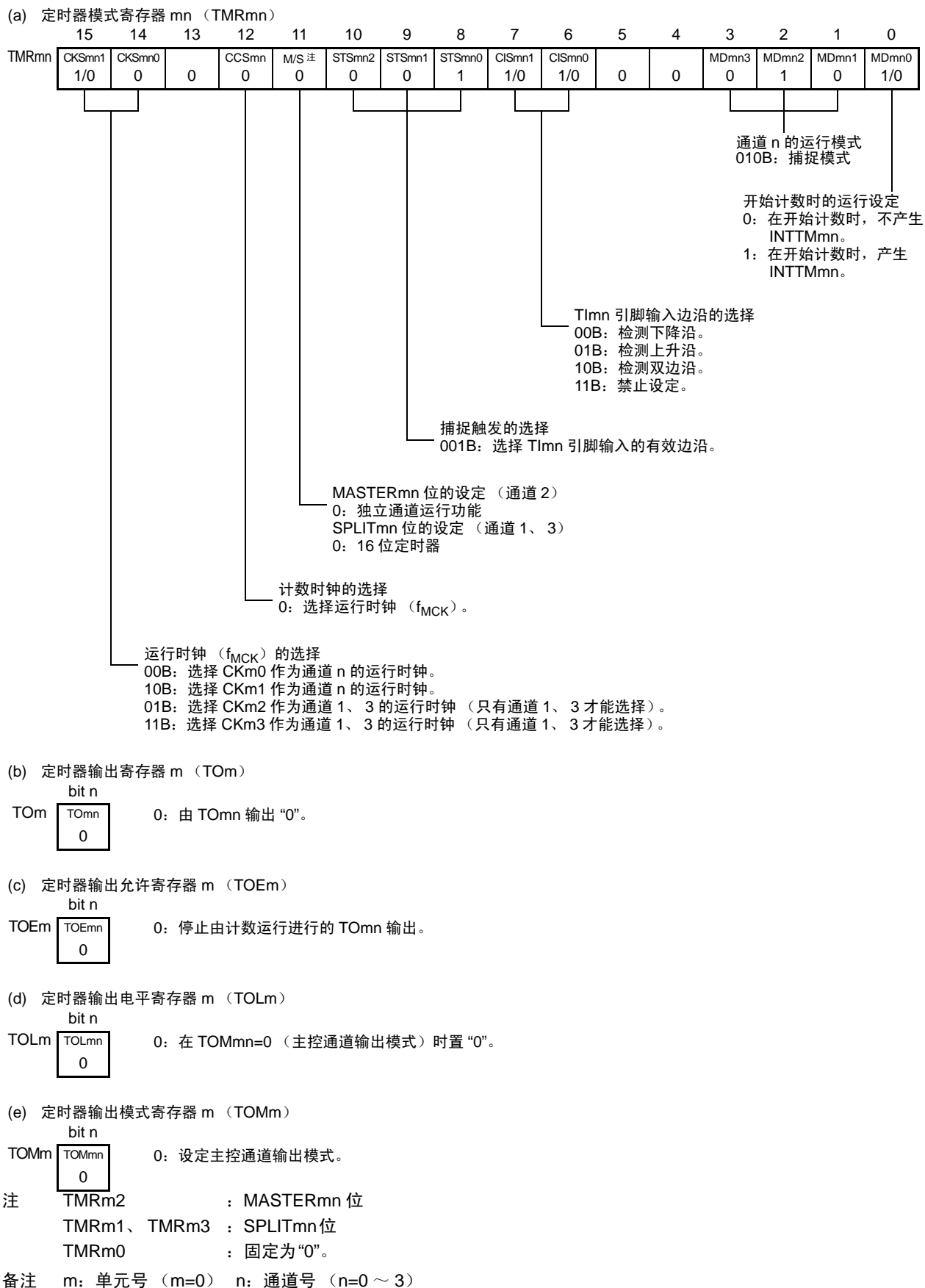


图 6-52 输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 m 的 TAUmEN 位置“1”。 →	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	设定定时器模式寄存器 mn (TMRmn) (确定通道的运行模式)。 将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置“1”。 → 因为 TSmn 位是触发位, 所以自动返回到“0”。	TEmn 位为“1”并且开始计数。 通过输入计数时钟将定时器计数寄存器 mn (TCRmn) 清“0000H”。当 TMRmn 寄存器的 MDmn0 位为“1”时, 产生 INTTMmn。
运行中	只能更改 TMRmn 寄存器的 CISmn1 位和 CISmn0 位的设定值。 能随时读 TDRmn 寄存器。 能随时读 TCRmn 寄存器。 能随时读 TSRmn 寄存器。 禁止更改 TOMmn 位、TOLmn 位、TOMn 位和 TOEmn 位的设定值。	计数器 (TCRmn) 从“0000H”开始递增计数, 如果检测到 TImn 引脚输入的有效边沿, 就将计数值传送 (捕捉) 到定时器数据寄存器 mn (TDRmn), 同时将 TCRmn 寄存器清“0000H”并且产生 INTTMmn。此时, 如果发生上溢, 就将定时器状态寄存器 mn (TSRmn) 的 OVF 位置位。如果不发生上溢, 就清除 OVF 位。 此后, 重复此运行。
停止 运行	将 TTmn 位置“1”。 → 因为 TTmn 位是触发位, 所以自动返回到“0”。	TEmn 位变为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 保持 TSRmn 寄存器的 OVF 位。
TAU 停止	将 PER0 寄存器的 TAUmEN 位置“0”。 →	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)



#### 6.8.4 作为输入信号高低电平宽度测量的运行

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

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

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

在捕捉 & 单次计数模式中，定时器计数寄存器 mn（TCRmn）用作递增计数器。

如果将定时器通道开始寄存器 m（TSM）的通道开始触发位（TSmn）置“1”，TEmn 位就变为“1”，并且进入 TIMn 引脚的开始边沿检测等待状态。

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

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

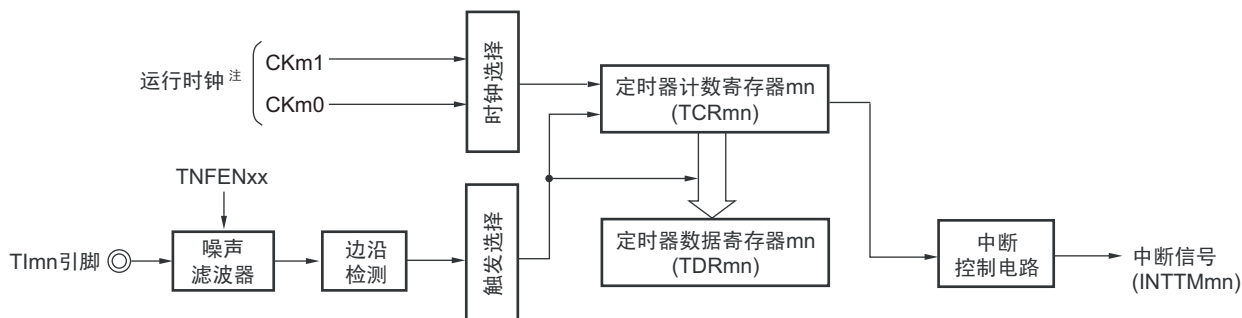
即使计数器进行了 2 个周期或者 2 个周期以上的完整计数，也认为发生上溢而将 TSRmn 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

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

TMRmn 寄存器的 CISmn1、CISmn0=10B：测量低电平宽度。

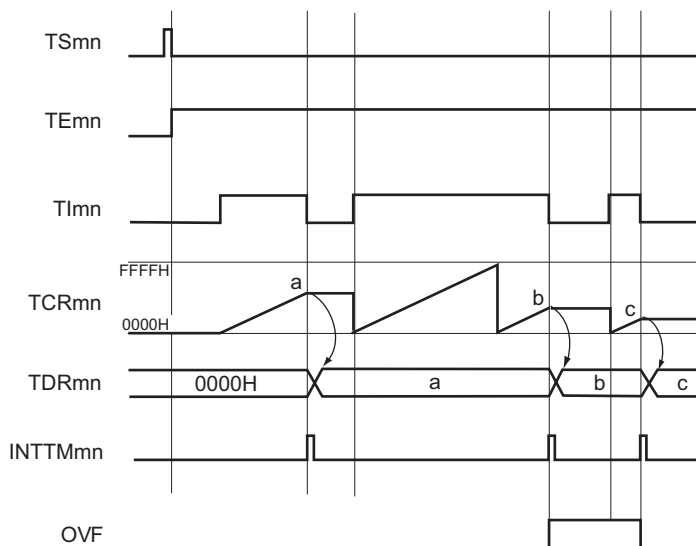
TMRmn 寄存器的 CISmn1、CISmn0=11B：测量高电平宽度。

图 6-53 作为输入信号高低电平宽度测量运行的框图



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

图 6-54 作为输入信号高低电平宽度测量的运行基本时序例子

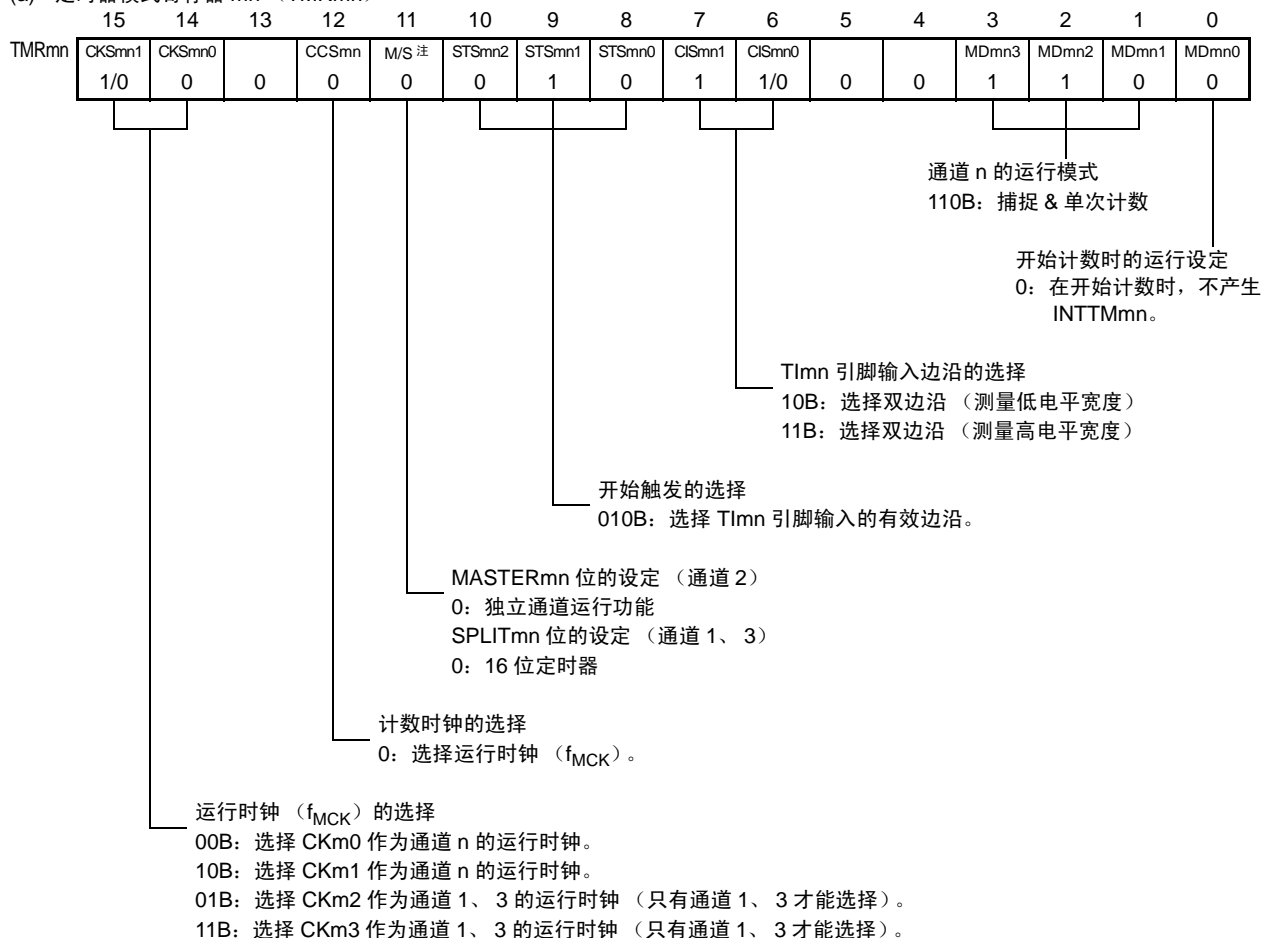


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

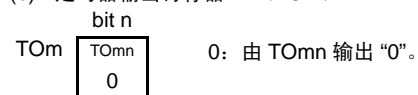
2. TSmn : 定时器通道开始寄存器 m (TSm) 的 bit n
- TEmn : 定时器通道允许状态寄存器 m (TEm) 的 bit n
- TImn : TImn 引脚输入信号
- TCRmn : 定时器计数寄存器 mn (TCRmn)
- TDRmn : 定时器数据寄存器 mn (TDRmn)
- OVF : 定时器状态寄存器 mn (TSRmn) 的 bit0

图 6-55 测量输入信号的高低电平宽度时的寄存器设定内容例子

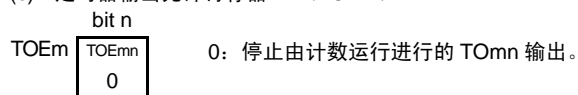
(a) 定时器模式寄存器 mn (TMRmn)



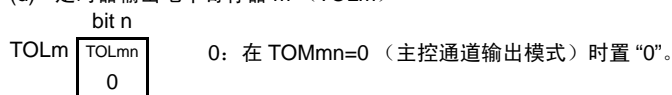
(b) 定时器输出寄存器 m (TOM)



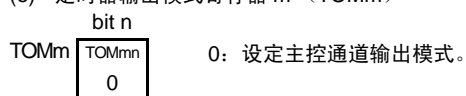
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-56 输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 mn (TMRmn) (确定通道的运行模式)。 将 TOEmn 位置“0”, 并且停止 TOmn 运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置“1”。 因为 TSmn 位是触发位, 所以自动返回到“0”。	TEmn 位为“1”并且进入 TImn 引脚的开始边沿检测等待状态。
	检测 TImn 引脚输入的计数开始边沿。	将定时器计数寄存器 mn (TCRmn) 清“0000H”并且开始递增计数。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。 禁止更改 TMRmn 寄存器、TOMmn 位、TOLmn 位、TOmn 位和 TOEmn 位的设定值。	在检测到 TImn 引脚的开始边沿后, 计数器 (TCRmn) 从“0000H”开始递增计数。如果检测到 TImn 引脚的捕捉边沿, 就将计数值传送到定时器数据寄存器 mn (TDRmn), 并且产生 INTTMmn。 此时, 如果发生上溢, 就将定时器状态寄存器 mn (TSRmn) 的 OVF 位置位。如果没有发生上溢, 就清除 OVF 位。TCRmn 寄存器在检测到下一个 TImn 引脚的开始边沿前停止计数。 此后, 重复此运行。
停止 运行	将 TTmn 位置“1”。 因为 TTmn 位是触发位, 所以自动返回到“0”。	TEmn 位为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。 保持 TSRmn 寄存器的 OVF 位。
TAU 停止	将 PER0 寄存器的 TAUmEN 位置“0”。	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

### 6.8.5 作为延迟计数器的运行

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

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

中断产生周期能用以下计算式进行计算：

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

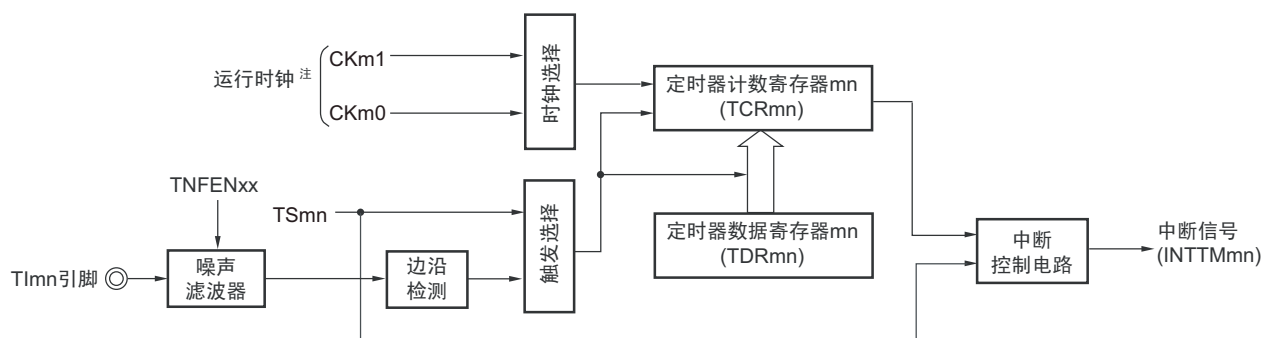
在单次计数模式中，定时器计数寄存器  $mn$ （ $TCR_{mn}$ ）用作递减计数器。

如果将定时器通道开始寄存器  $m$ （ $TS_m$ ）的通道开始触发位（ $TS_{mn}$ 、 $TSH_{m1}$ 、 $TSH_{m3}$ ）置“1”， $TE_{mn}$  位、 $TEH_{m1}$  位和  $TEH_{m3}$  位就变为“1”，并且进入  $TI_{mn}$  引脚的有效边沿检测等待状态。

通过  $TI_{mn}$  引脚输入的有效边沿检测，开始  $TCR_{mn}$  寄存器的运行，并且装入定时器数据寄存器  $mn$ （ $TDR_{mn}$ ）的值。 $TCR_{mn}$  寄存器通过计数时钟，从装入的  $TDR_{mn}$  寄存器的值开始递减计数。如果  $TCR_{mn}$  变为“0000H”，就输出  $INTTM_{mn}$ ，并且在检测到下一个  $TI_{mn}$  引脚输入的有效边沿前停止计数。

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

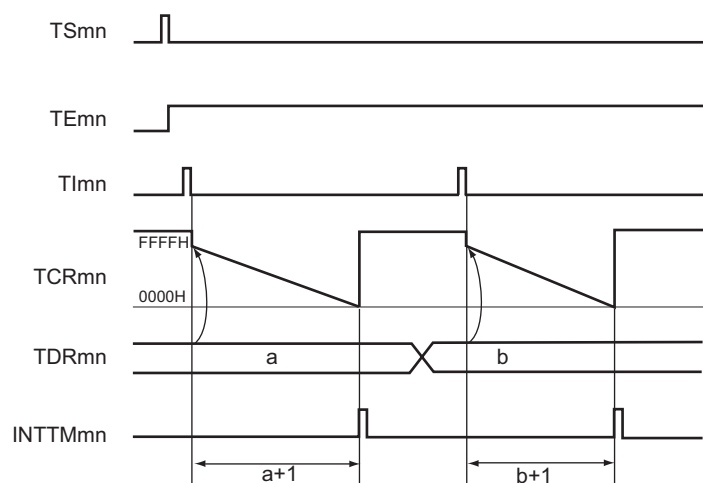
图 6-57 作为延迟计数器运行的框图



注 在通道 1 和通道 3 时，能从 CK<sub>m0</sub>、CK<sub>m1</sub>、CK<sub>m2</sub> 和 CK<sub>m3</sub> 中选择时钟。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-58 作为延迟计数器的运行基本时序例子

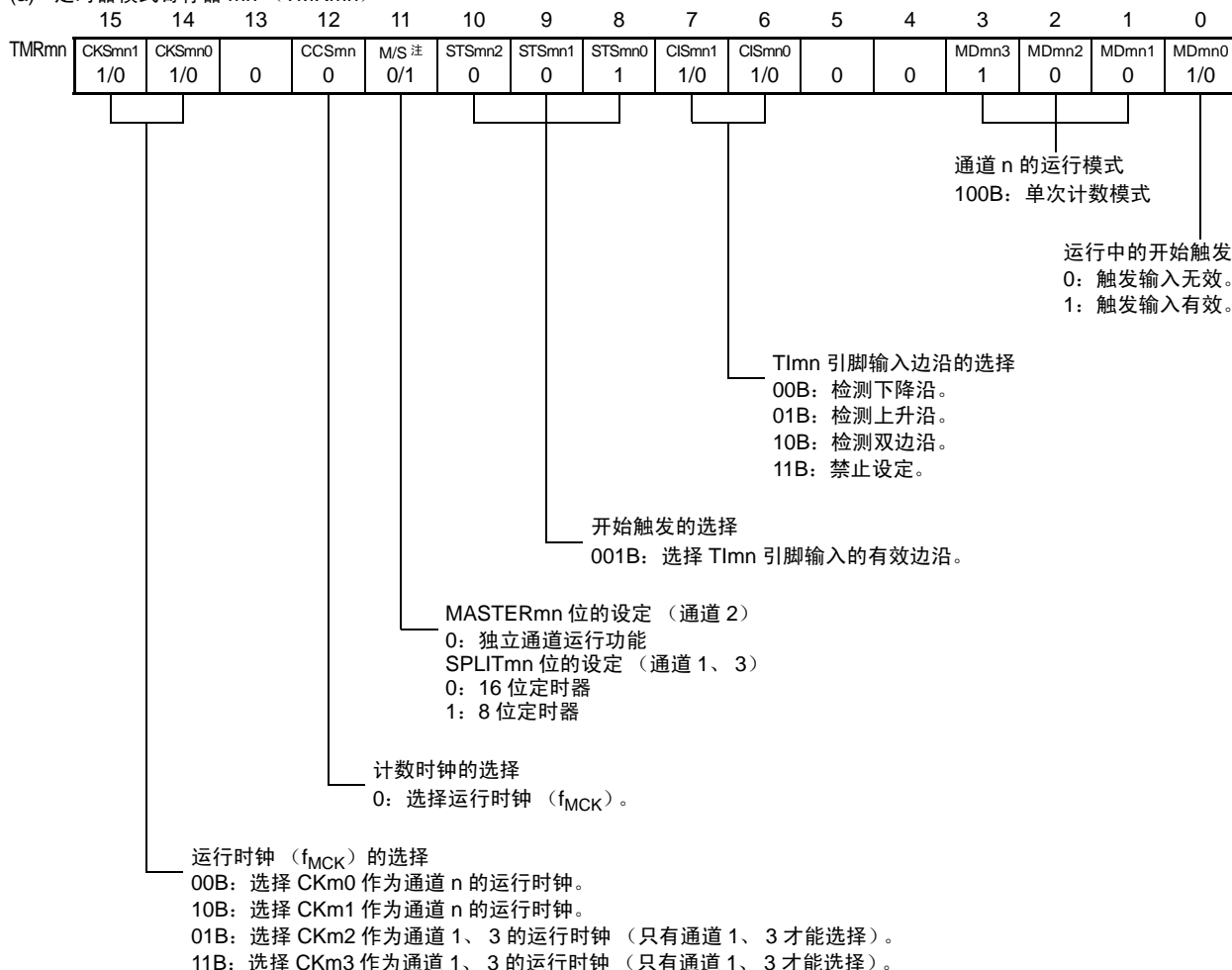


备注 1. m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

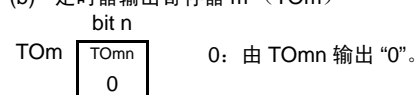
2. TSmn : 定时器通道开始寄存器 m (TSM) 的 bit n
- TE mn : 定时器通道允许状态寄存器 m (TEM) 的 bit n
- TImn : TImn 引脚输入信号
- TCRmn : 定时器计数寄存器 mn (TCRmn)
- TDRmn : 定时器数据寄存器 mn (TDRmn)

图 6-59 延迟计数器功能时的寄存器设定内容例子

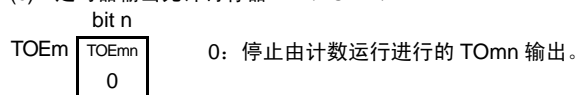
(a) 定时器模式寄存器 mn (TMRmn)



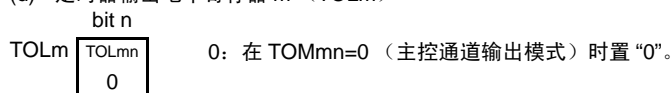
(b) 定时器输出寄存器 m (TOM)



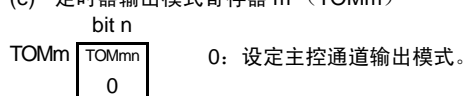
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)



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

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)

图 6-60 延迟计数器功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 ~ CKm3 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 mn (TMRmn) (确定通道 n 的运行模式)。 给定时器数据寄存器 mn (TDRmn) 设定输出延迟时间。 将 TOEmn 位置“0”并且停止 T0mn 运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TSmn 位置“1”。 因为 TSmn 位是触发位, 所以自动返回到“0”。	TEmn 位为“1”并且进入开始触发 (检测 TImn 引脚输入的有效边沿或者将 TSmn 位置“1”)的检测等待状态。
	通过检测到下一个开始触发, 开始递减计数。 • 检测 TImn 引脚输入的有效边沿。 • 通过软件将 TSmn 位置“1”。	将 TDRmn 寄存器的值装入定时器计数寄存器 mn (TCRmn)。
运行中	能任意更改 TDRmn 寄存器的设定值。 能随时读 TCRmn 寄存器。 不使用 TSRmn 寄存器。	计数器 (TCRmn) 进行递减计数。如果 TCRmn 计数到“0000H”, 就输出 INTTmn, 并且在检测到下一次开始触发 (检测 TImn 引脚输入的有效边沿或者将 TSmn 位置“1”)前 TCRmn 为“0000H”而停止计数。
停止 运行	将 TTmn 位置“1”。 因为 TTmn 位是触发位, 所以自动返回到“0”。	TEmn 位为“0”并且停止计数。 TCRmn 寄存器保持计数值而停止计数。
TAU 停止	将 PER0 寄存器的 TAUmEN 位置“0”。	定时器单元 0 的输入时钟处于停止提供状态。 对全部电路和各通道的 SFR 进行初始化。

备注 m: 单元号 (m=0) n: 通道号 (n=0 ~ 3)



## 6.9 定时器阵列单元的多通道联动运行功能

### 6.9.1 作为单触发脉冲输出功能的运行

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

$\begin{aligned} \text{延迟} &= \{\text{TDR}_{mn} \text{ (主控) 的设定值} + 2\} \times \text{计数时钟周期} \\ \text{脉宽} &= \{\text{TDR}_{mp} \text{ (从属) 的设定值}\} \times \text{计数时钟周期} \end{aligned}$
--

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

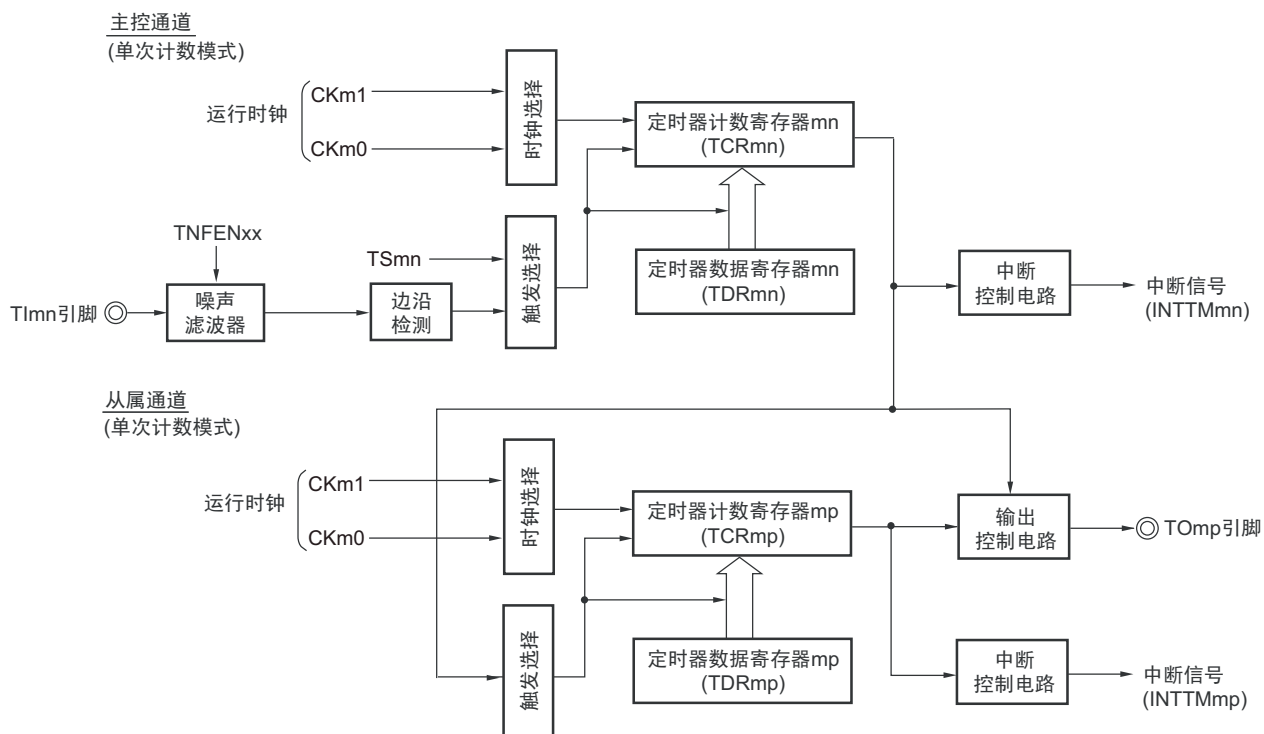
在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的 **INTTM<sub>mn</sub>** 作为开始触发，从属通道的 **TCR<sub>mp</sub>** 寄存器开始运行并且装入 **TDR<sub>mp</sub>** 寄存器的值。**TCR<sub>mp</sub>** 寄存器通过计数时钟，从装入的 **TDR<sub>mp</sub>** 寄存器值开始递减计数。如果计数值变为“0000H”，就输出 **INTTM<sub>mp</sub>**，并且在检测到下一个开始触发（主控通道的 **INTTM<sub>mn</sub>**）前停止计数。在从主控通道产生 **INTTM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

不使用 **TI<sub>mn</sub>** 引脚输入也能将软件操作（**TS<sub>mn</sub>**=1）作为开始触发来输出单触发脉冲。

**注意** 因为主控通道的定时器数据寄存器 **mn** (**TDR<sub>mn</sub>**) 和从属通道的 **TDR<sub>mp</sub>** 寄存器的装入时序不同，所以如果在运行中改写 **TDR<sub>mn</sub>** 寄存器和 **TDR<sub>mp</sub>** 寄存器，就输出不正常的波形。必须在产生 **INTTM<sub>mn</sub>** 后改写 **TDR<sub>mn</sub>** 寄存器，并且在产生 **INTTM<sub>mp</sub>** 后改写 **TDR<sub>mp</sub>** 寄存器。

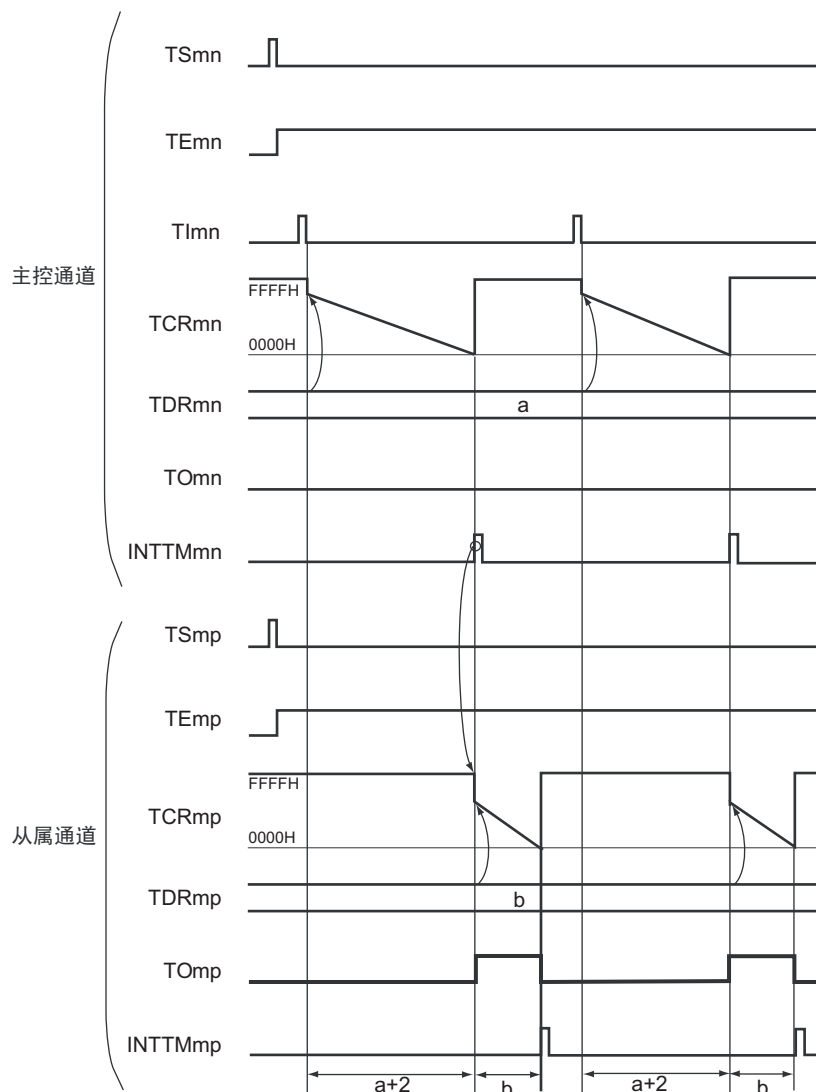
**备注** **m**: 单元号 (**m**=0) **n**: 主控通道号 (**n**=0、2) **p**: 从属通道号 (**n**=0: **p**=1、2、3, **n**=2: **p**=3)

图 6-61 作为单触发脉冲输出功能运行的框图



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

图 6-62 作为单触发脉冲输出功能的运行基本时序例子

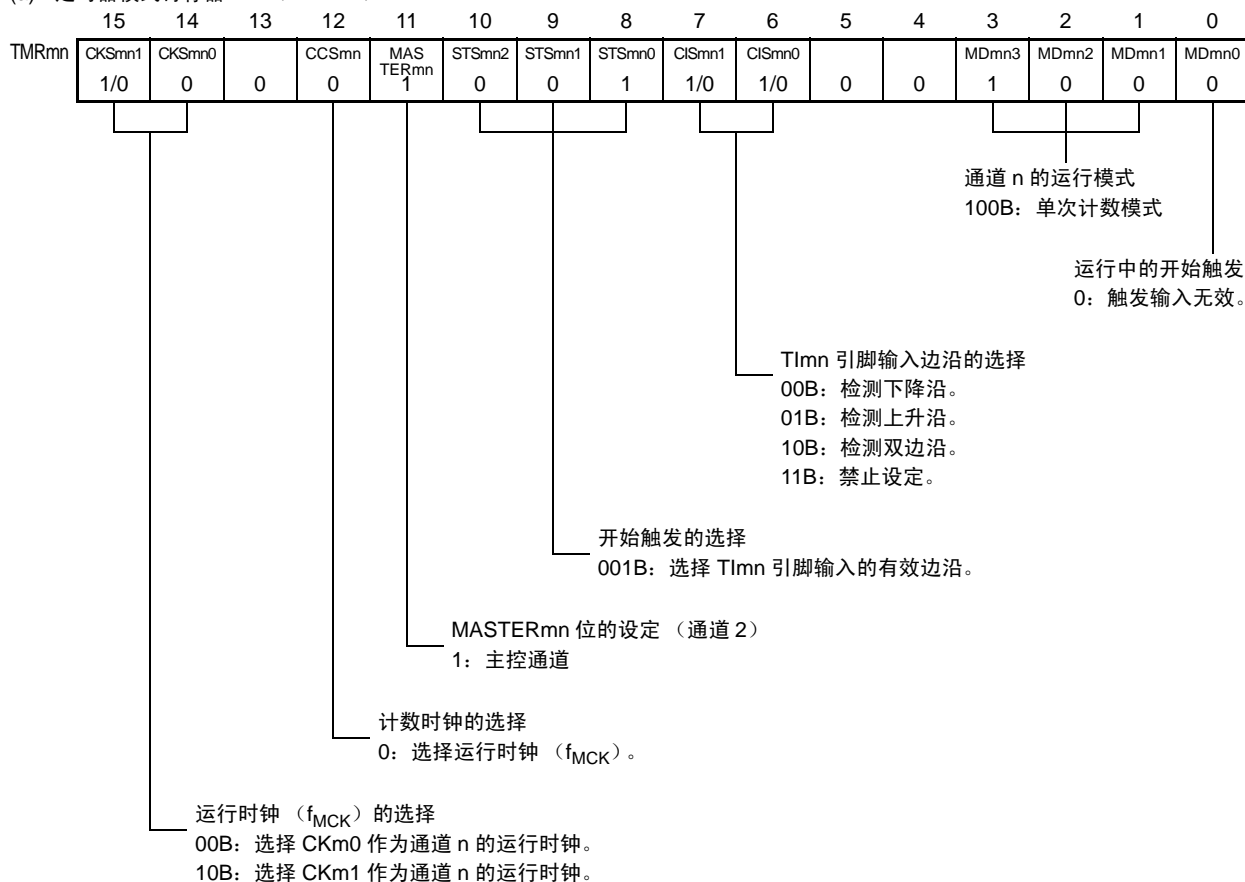


备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

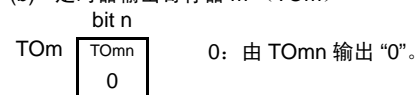
2. TSmn、TSmp : 定时器通道开始寄存器 m (TSM) 的 bit n、p
- TEmn、TEmp : 定时器通道允许状态寄存器 m (TEM) 的 bit n、p
- TImn、TImp : TImn 引脚和 TImp 引脚的输入信号
- TCRmn、TCRmp : 定时器计数寄存器 mn、mp (TCRmn、TCRmp)
- TDRmn、TDRmp : 定时器数据寄存器 mn、mp (TDRmn、TDRmp)
- TOmn、TOmp : TOmn 引脚和 TOmp 引脚的输出信号

图 6-63 单触发脉冲输出功能时（主控通道）的寄存器设定内容例子

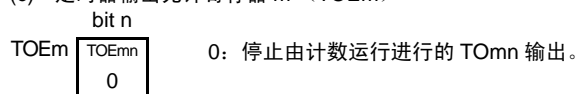
(a) 定时器模式寄存器 mn (TMRmn)



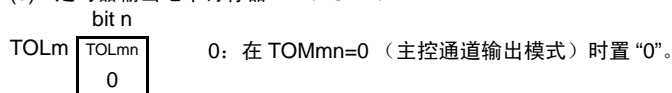
(b) 定时器输出寄存器 m (TOM)



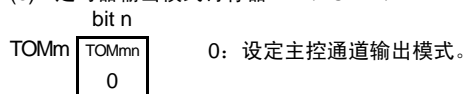
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



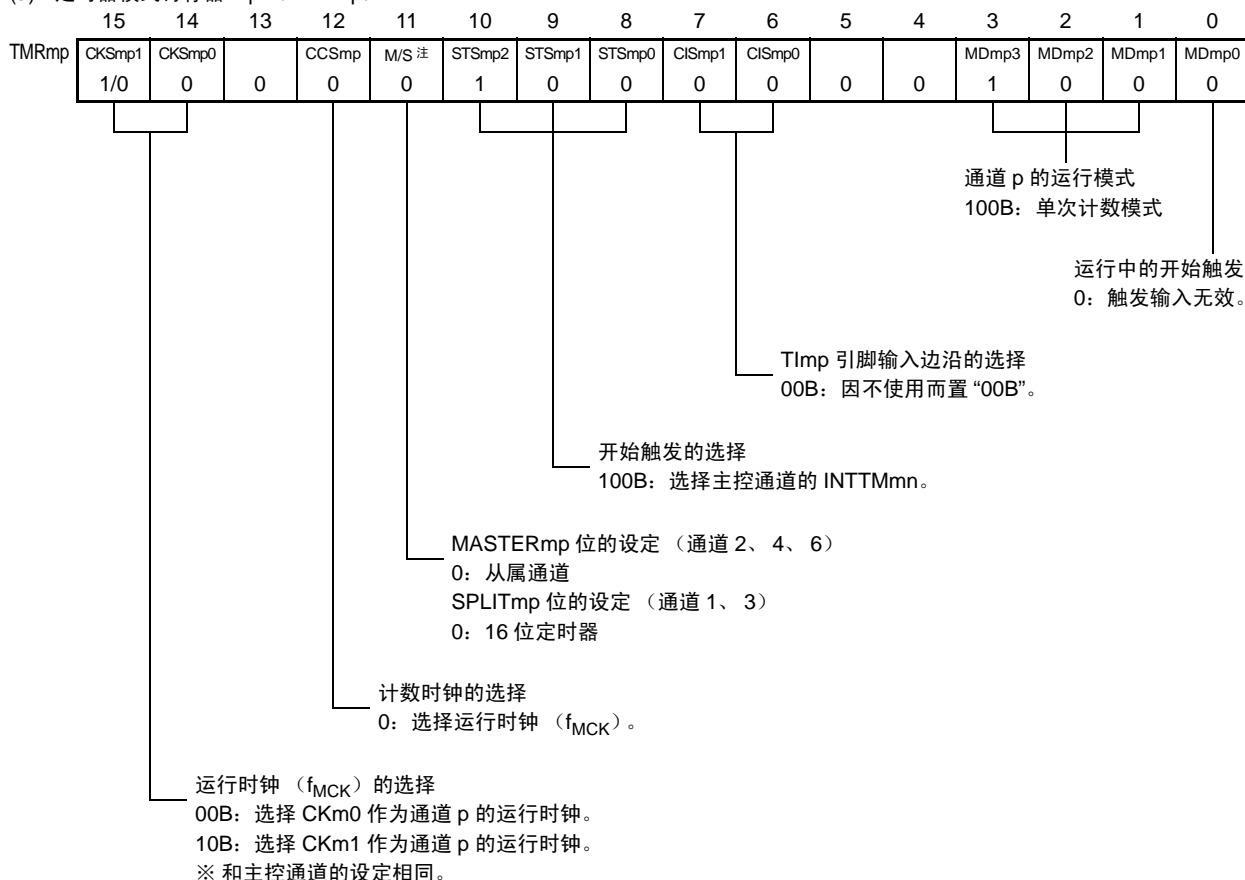
(e) 定时器输出模式寄存器 m (TOMm)



备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图 6-64 单触发脉冲输出功能时（从属通道）的寄存器设定内容例子

## (a) 定时器模式寄存器 mp (TMRmp)



## (b) 定时器输出寄存器 m (TOm)

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

## (c) 定时器输出允许寄存器 m (TOEm)

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

## (d) 定时器输出电平寄存器 m (TOLm)

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

## (e) 定时器输出模式寄存器 m (TOMm)

bit p	
TOMm	TOMmp
	1
	1: 设定从属通道输出模式。

注 TMRm2、TMRm4、TMRm6 : MASTERmp 位  
TMRm1、TMRm3 : SPLITmp 位  
TMRm5、TMRm7 : 固定为“0”。

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

图 6-65 单触发脉冲输出功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 和 CKm1 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“1”。 设定使用的 2 个通道的定时器模式寄存器 mn、mp (TMRmn、TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 mn (TDRmn) 设定输出延迟时间, 并且给从属通道的 TDRmp 寄存器设定脉宽。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp 位置“1” (从属通道输出模式)。 设定 TOLmp 位。 设定 TOmp 位并且确定 TOmp 输出的初始电平。 将 TOEmp 位置“1”, 允许 TOmp 运行。 将端口寄存器和端口模式寄存器置“0”。	TOmp 引脚处于 Hi-Z 输出状态。  当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TOmp 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 不变。 TOmp 引脚输出 TOmp 设定的电平。

图 6-65 单触发脉冲输出功能时的操作步骤 (2/2)

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

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

### 6.9.2 作为 PWM 功能的运行

将 2 个通道成对使用，能生成任意周期和占空比的脉冲。

输出脉冲的周期和占空比能用以下计算式进行计算：

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

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

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

当用作 PWM 功能时，主控通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出 (TOmp) 周期。

从属通道用作单次计数模式。以主控通道的 INTTMmn 中断为开始触发，将 TDRmp 寄存器的值装入 TCRmp 寄存器，并且进行递减计数，计数到“0000H”为止。当计数到“0000H”时，输出 INTTMmp 中断，并且等待下一个开始触发 (主控通道的 INTTMmn)。

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

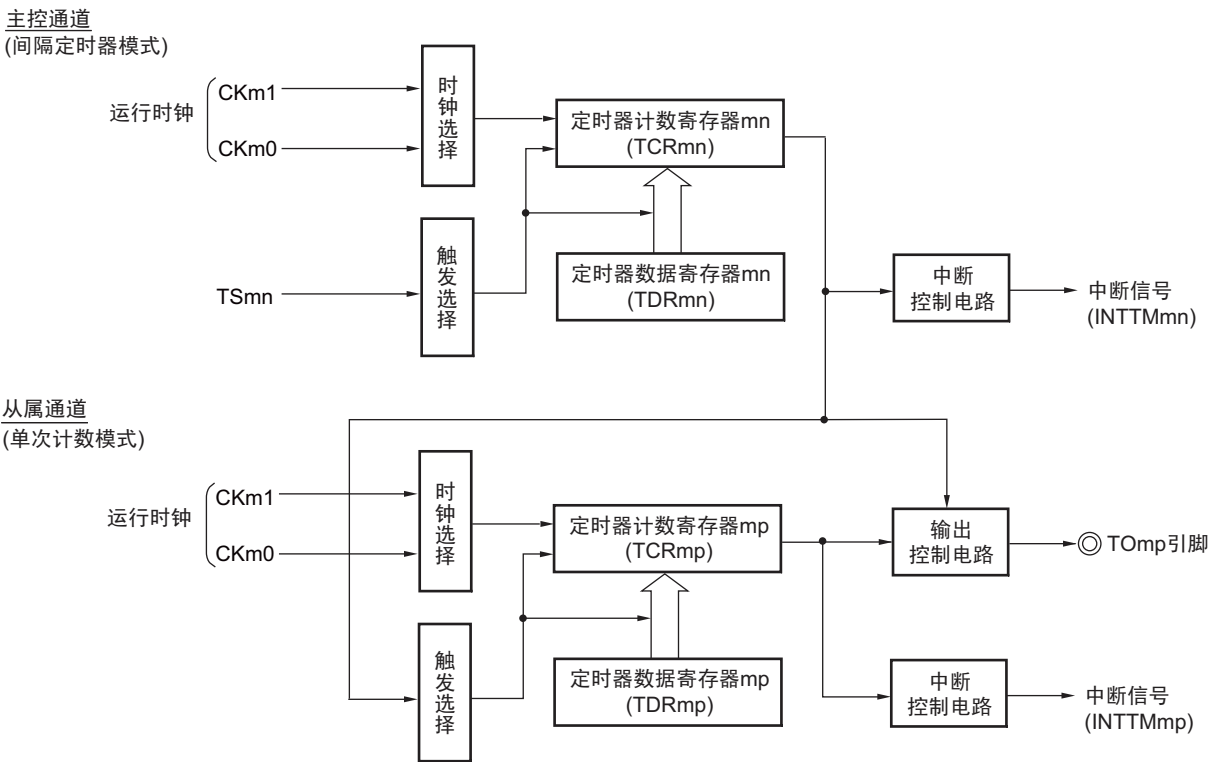
在从主控通道产生 INTTMmn 中断并且经过 1 个时钟后，PWM 输出 (TOmp) 变为有效电平，并且在从属通道的 TCRmp 寄存器的值为“0000H”时变为无效电平。

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

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

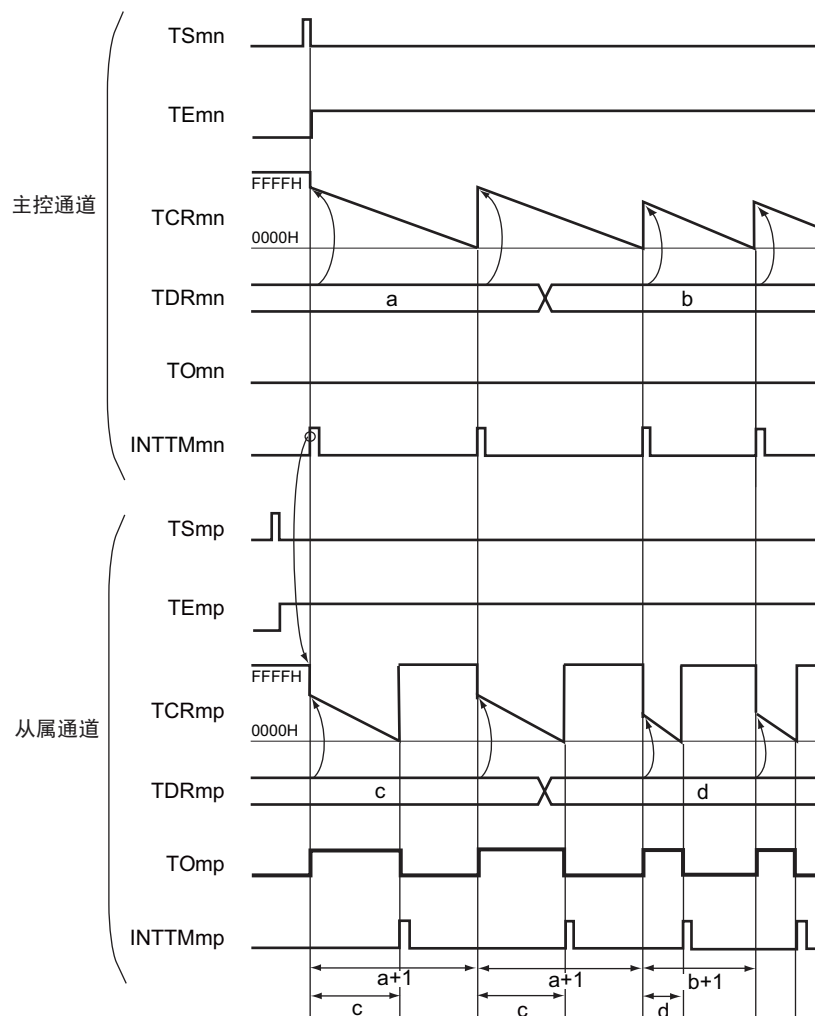


图 6-66 作为 PWM 功能运行的框图



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

图 6-67 作为 PWM 功能的运行基本时序例子

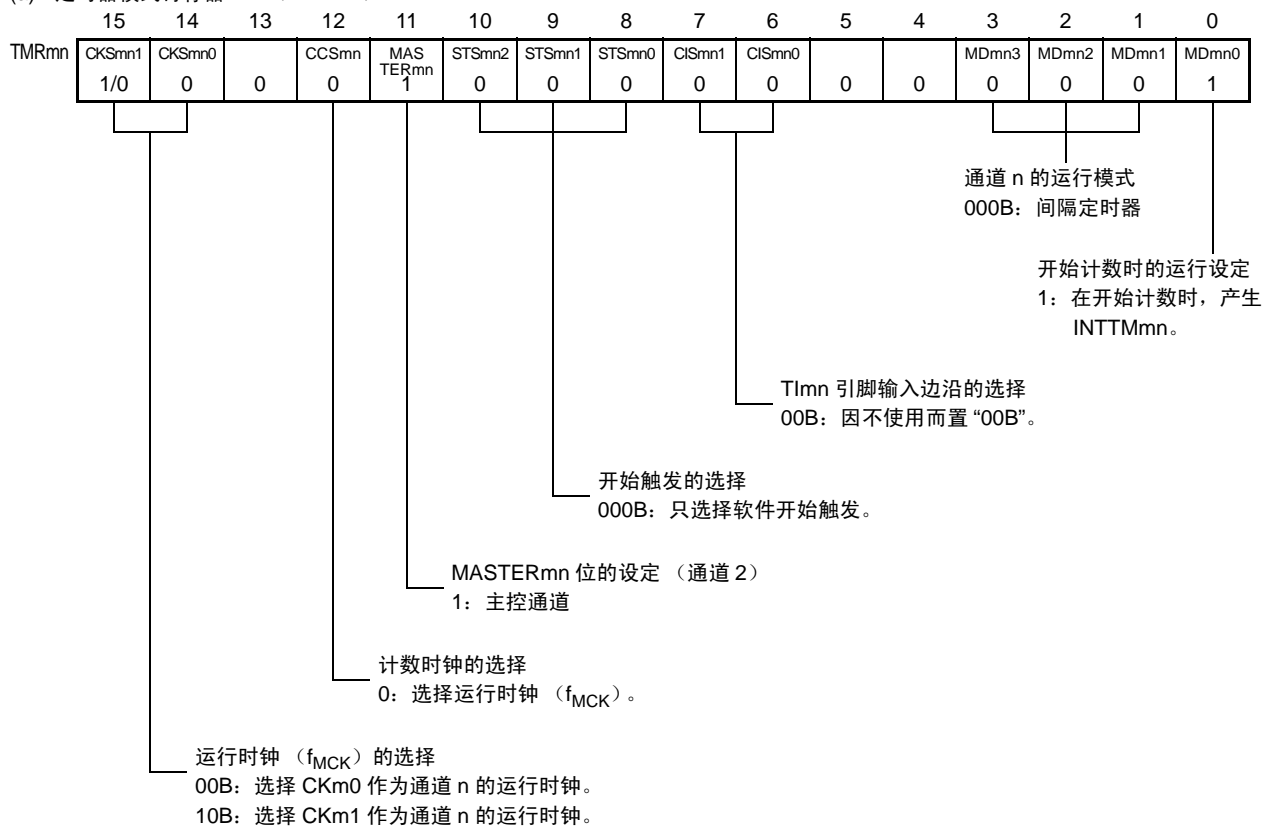


备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 (n=0: p=1、2、3, n=2: p=3)

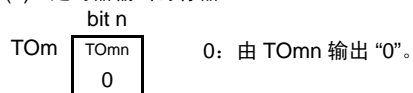
2. TSmn、TSmp : 定时器通道开始寄存器 m (TSm) 的 bit n、p  
 TEmn、TEmp : 定时器通道允许状态寄存器 m (TEm) 的 bit n、p  
 TCRmn、TCRmp : 定时器计数寄存器 mn、mp (TCRmn、TCRmp)  
 TDRmn、TDRmp : 定时器数据寄存器 mn、mp (TDRmn、TDRmp)  
 TOMn、TOMP : TOMn 引脚和 TOMP 引脚的输出信号

图 6-68 PWM 功能时（主控通道）的寄存器设定内容例子

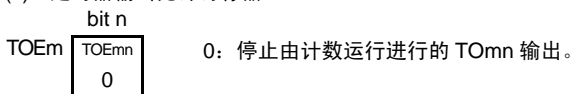
(a) 定时器模式寄存器 mn (TMRmn)



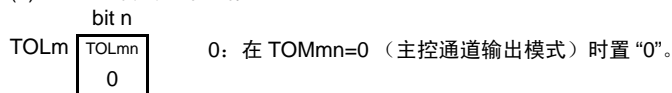
(b) 定时器输出寄存器 m (TOM)



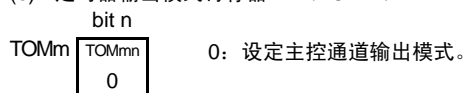
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



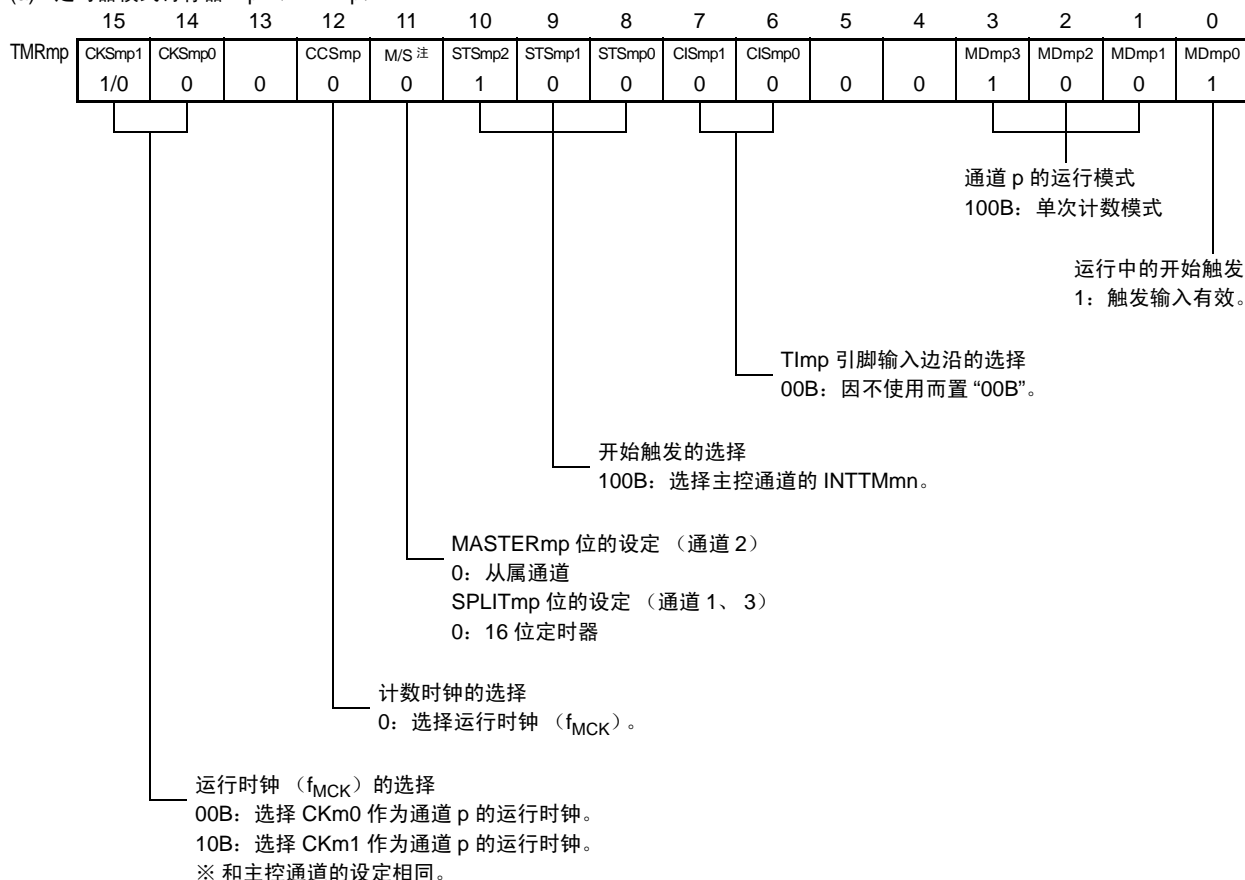
(e) 定时器输出模式寄存器 m (TOMm)



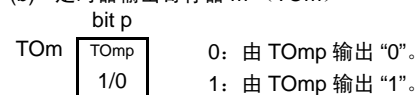
备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图 6-69 PWM 功能时（从属通道）的寄存器设定内容例子

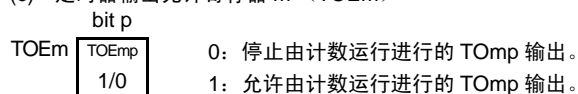
## (a) 定时器模式寄存器 mp (TMRmp)



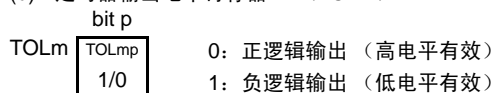
## (b) 定时器输出寄存器 m (TOM)



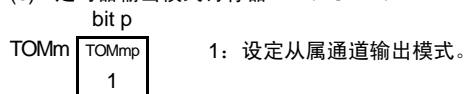
## (c) 定时器输出允许寄存器 m (TOEm)



## (d) 定时器输出电平寄存器 m (TOLm)



## (e) 定时器输出模式寄存器 m (TOMm)



注 TMRm5、TMRm7: 固定为“0”。  
TMRm1、TMRm3: SPLITmp 位

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

图 6-70 PWM 功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 和 CKm1 的时钟频率。	
通道初 始设定	设定使用的 2 个通道的定时器模式寄存器 mn、mp (TMRmn、TMRmp) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 mn (TDRmn) 设定间隔 (周期) 值, 并且给从属通道的 TDRmp 寄存器设定占空比的值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp 位置“1” (从属通道输出模式)。 设定 TOLmp 位。 设定 TOmp 位并且确定 TOmp 输出的初始电平。	TOmp 引脚处于 Hi-Z 输出状态。
	将 TOEmp 位置“1”, 允许 TOmp 运行。————→ 将端口寄存器和端口模式寄存器置“0”。————→	当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TOmp 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 不变。 TOmp 引脚输出 TOmp 设定的电平。

图 6-70 PWM 功能时的操作步骤 (2/2)

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

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

### 6.9.3 作为多重 PWM 输出功能的运行

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

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

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

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

在间隔定时器模式中，主控通道的定时器计数寄存器 mn (TCRmn) 运行并且对周期进行计数。

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

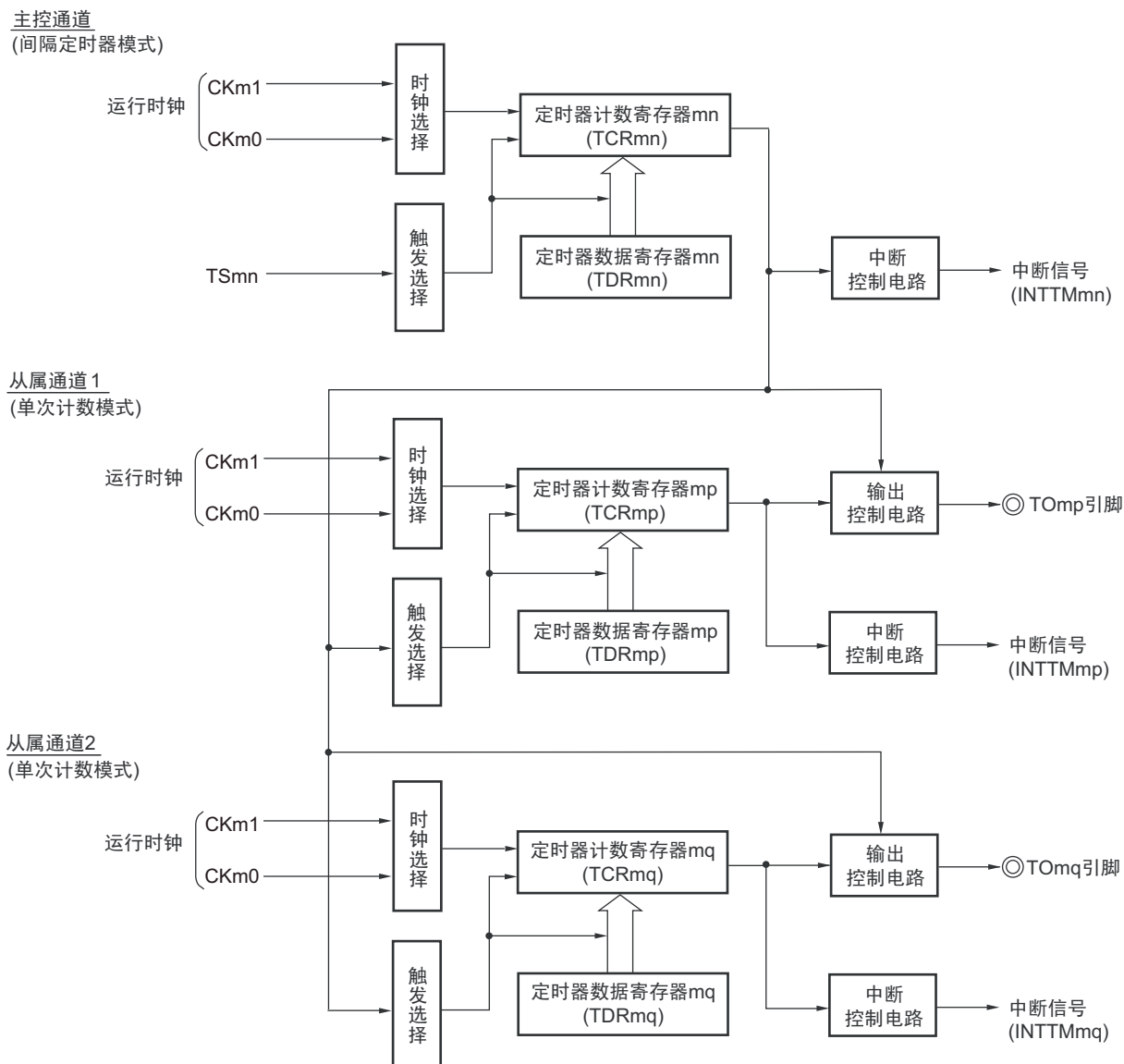
和从属通道 1 的 TCRmp 寄存器相同，在单次计数模式中，从属通道 2 的 TCRmq 寄存器运行并且对占空比进行计数以及从 TOmq 引脚输出 PWM 波形。以主控通道的 INTTMmn 为开始触发，将 TDRmq 寄存器的值装入 TCRmq 寄存器并且开始递减计数。如果 TCRmq 变为“0000H”，就输出 INTTMmq，并且在输入下一个开始触发（主控通道的 INTTMmn）前停止计数。在从主控通道产生 INTTMmn 并且经过 1 个计数时钟后，TOmq 的输出电平变为有效电平，如果 TCRmq 变为“0000H”，就变为无效电平。

当通过如此的运行将通道 0 用作主控通道时，最多能同时输出 7 种 PWM 信号。

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

**备注** m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 q: 从属通道号  
 $n < p < q \leq 3$  (p 和 q 是大于 n 的整数)

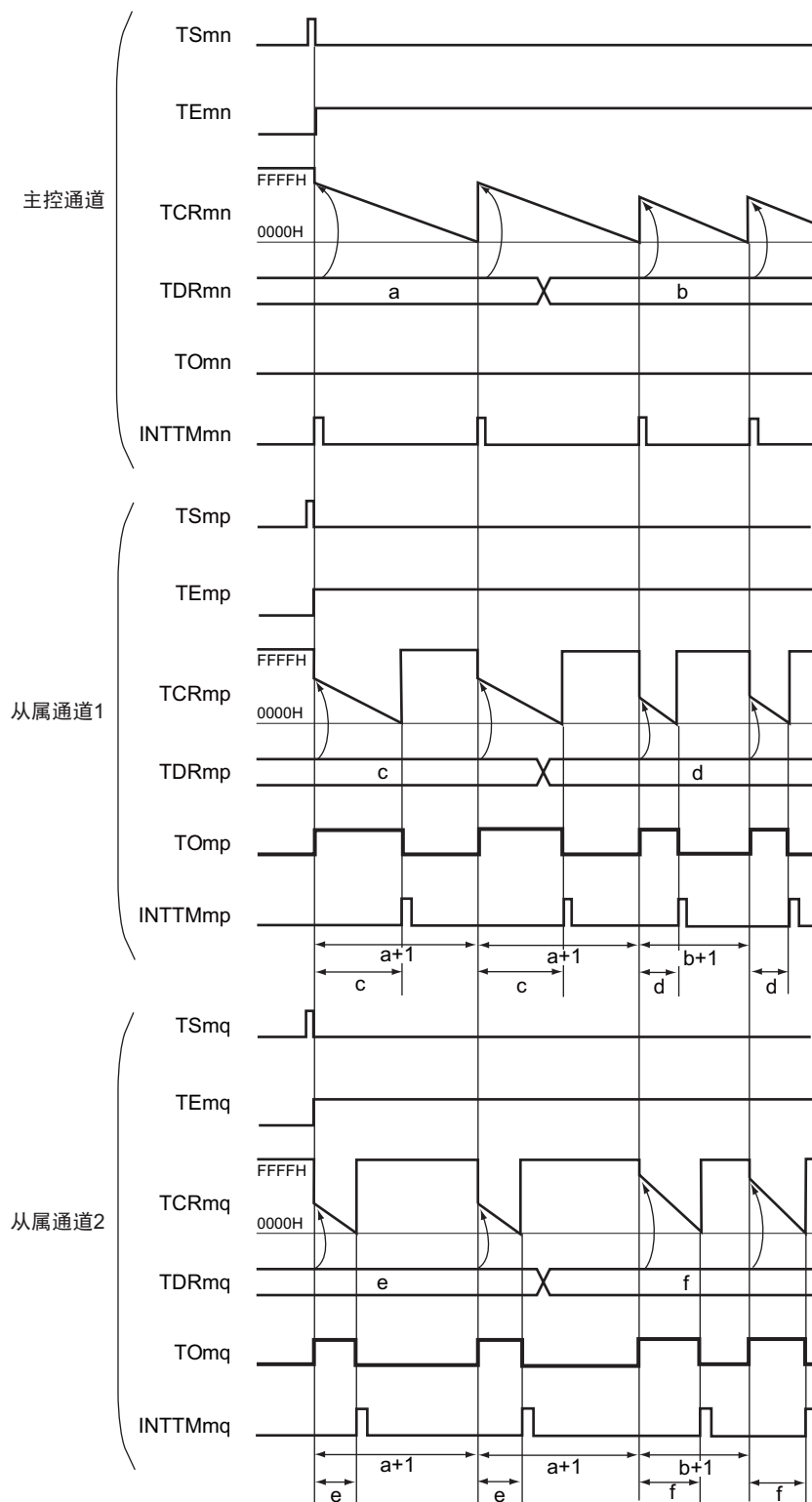
图 6-71 作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）



备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 q: 从属通道号  
 $n < p < q \leq 3$  (p和q是大于n的整数)



图 6-72 作为多重 PWM 输出功能的运行基本时序例子（输出 2 种 PWM 的情况）



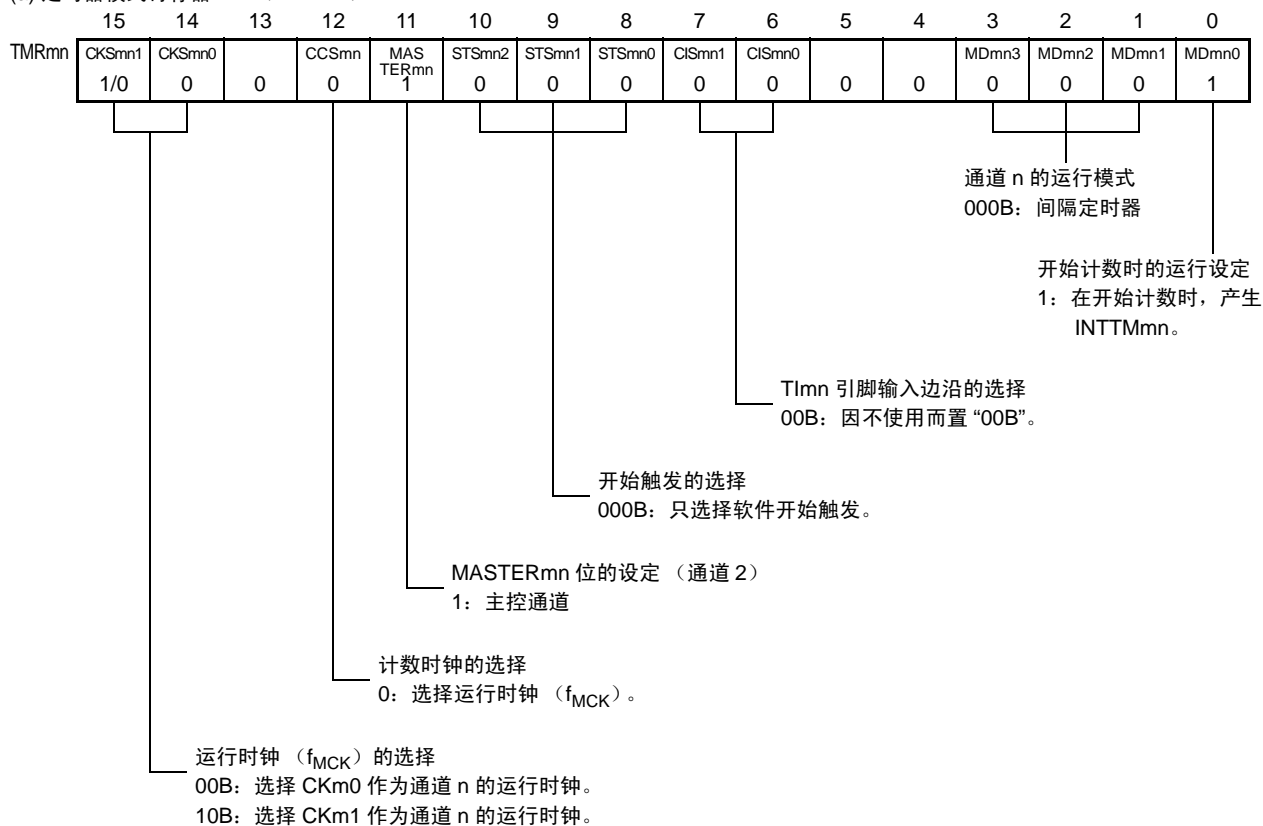
备注 1. m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 q: 从属通道号

$n < p < q \leq 3$  (p 和 q 是大于 n 的整数)

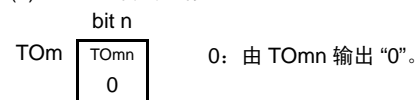
- 2. TSmn、TSmp、TSmq : 定时器通道开始寄存器 m (TSm) 的 bit n、p、q
- TEmn、TEmp、TEmq : 定时器通道允许状态寄存器 m (TEm) 的 bit n、p、q
- TCRmn、TCRmp、TCRmq : 定时器计数寄存器 mn、mp、mq (TCRmn、TCRmp、TCRmq)
- TDRmn、TDRmp、TDRmq : 定时器数据寄存器 mn、mp、mq (TDRmn、TDRmp、TDRmq)
- TOmn、TOmp、TOmq : TOmn、TOmp、TOmq 引脚的输出信号

图 6-73 多重 PWM 输出功能时（主控通道）的寄存器设定内容例子

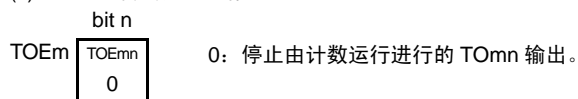
(a) 定时器模式寄存器 mn (TMRmn)



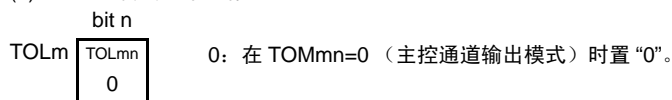
(b) 定时器输出寄存器 m (TOM)



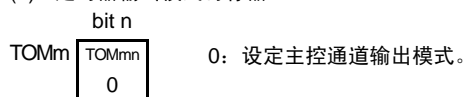
(c) 定时器输出允许寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)



(e) 定时器输出模式寄存器 m (TOMm)

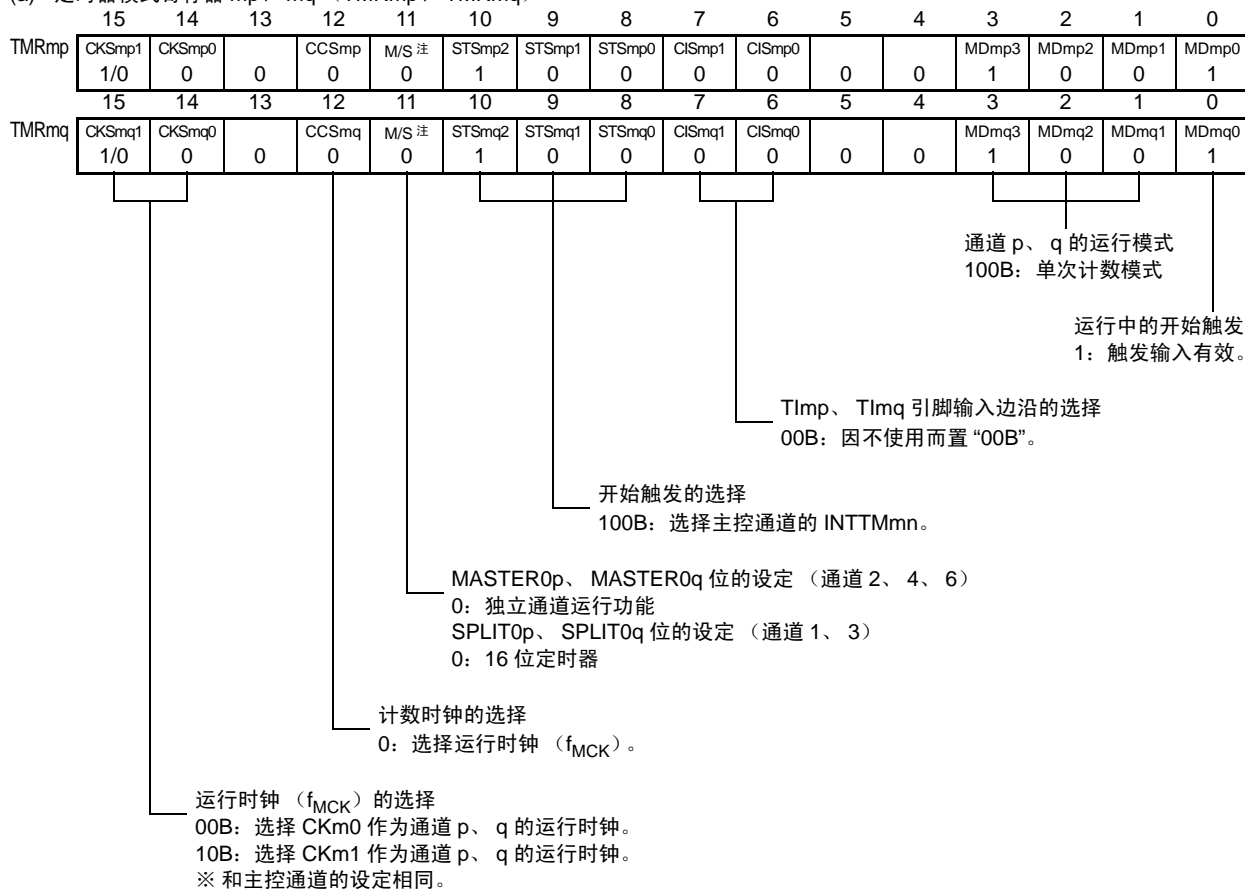


注 TMRm2 : MASTERmn=1  
TMRm0 : 固定为“0”。

备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2)

图 6-74 多重 PWM 输出功能时（从属通道）的寄存器设定内容例子（输出 2 种 PWM 的情况）

(a) 定时器模式寄存器 mp、mq（TMRmp、TMRmq）



(b) 定时器输出寄存器 m（TOm）

	bit q	bit p	
TOm	TOmq 1/0	TOmp 1/0	0: 由 TOmp 和 TOmq 输出“0”。 1: 由 TOmp 和 TOmq 输出“1”。

(c) 定时器输出允许寄存器 m（TOEm）

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

(d) 定时器输出电平寄存器 m（TOLm）

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

(e) 定时器输出模式寄存器 m（TOMm）

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

注 TMRm2、TMRm4、TMRm6 : MASTERmp 位、MASTERmq 位  
TMRm1、TMRm3 : SPLITmp 位、SPLITmq 位  
TMRm5、TMRm7 : 固定为“0”。

备注 m: 单元号（m=0） n: 主控通道号（n=0、2） p: 从属通道号 q: 从属通道号  
n < p < q ≤ 3（p 和 q 是大于 n 的整数）

图 6-75 多重 PWM 输出功能时的操作步骤（输出 2 种 PWM 的情况）(1/2)

	软件操作	硬件状态
TAU 初始 设定		定时器单元 0 的输入时钟处于停止提供状态。 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAUmEN 位置“1”。	定时器单元 0 的输入时钟处于提供状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 m (TPSm)。 确定 CKm0 和 CKm1 的时钟频率。	
通道初 始设定	设定使用的各通道的定时器模式寄存器 mn、mp、mq (TMRmn、TMRmp、TMRmq) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 mn (TDRmn) 设定间隔 (周期) 值, 并且给从属通道的 TDRmp 寄存器和 TDRmq 寄存器设定占空比的值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 m (TOMm) 的 TOMmp 位和 TOMmq 位置“1” (从属通道输出模式)。 设定 TOLmp 位和 TOLmq 位。 设定 TOmp 位和 TOmq 位, 并且确定 TOmp 和 TOmq 输出的初始电平。————→ 将 TOEmp 位和 TOEmq 位置“1”, 允许 TOmp 和 TOmq 的运行。————→ 将端口寄存器和端口模式寄存器置“0”。————→	TOmp 引脚和 TOmq 引脚处于 Hi-Z 输出状态。  当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TOmp 和 TOmq 初始设定的电平。 因为通道处于运行停止状态, 所以 TOmp 和 TOmq 不变。 TOmp 引脚和 TOmq 引脚输出 TOmp 和 TOmq 设定的电平。

图 6-75 多重 PWM 输出功能时的操作步骤 (输出 2 种 PWM 的情况) (2/2)

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

备注 m: 单元号 (m=0) n: 主控通道号 (n=0、2) p: 从属通道号 q: 从属通道号  
 $n < p < q \leq 3$  (p 和 q 是大于 n 的整数)

## 6.10 使用定时器阵列单元时的注意事项

### 6.10.1 使用定时器输出时的注意事项

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

- 使用被分配给 P12 的 TO03 输出的情况

为了将复用的 PCLBUZ0 输出置“0”，除了将端口模式寄存器（PM12 位）和端口寄存器（P12 位）置“0”以外，还必须将时钟输出选择寄存器 0（CKS0）的 bit7 置“0”（初始状态）。

## 第 7 章 时钟输出 / 蜂鸣器输出控制电路

### 7.1 时钟输出 / 蜂鸣器输出控制电路的功能

时钟输出是在遥控发送时输出载波并且输出提供给外围 IC 时钟的功能，蜂鸣器输出是输出蜂鸣器频率方波的功能。

能用 1 个引脚选择用作时钟输出或者蜂鸣器输出。

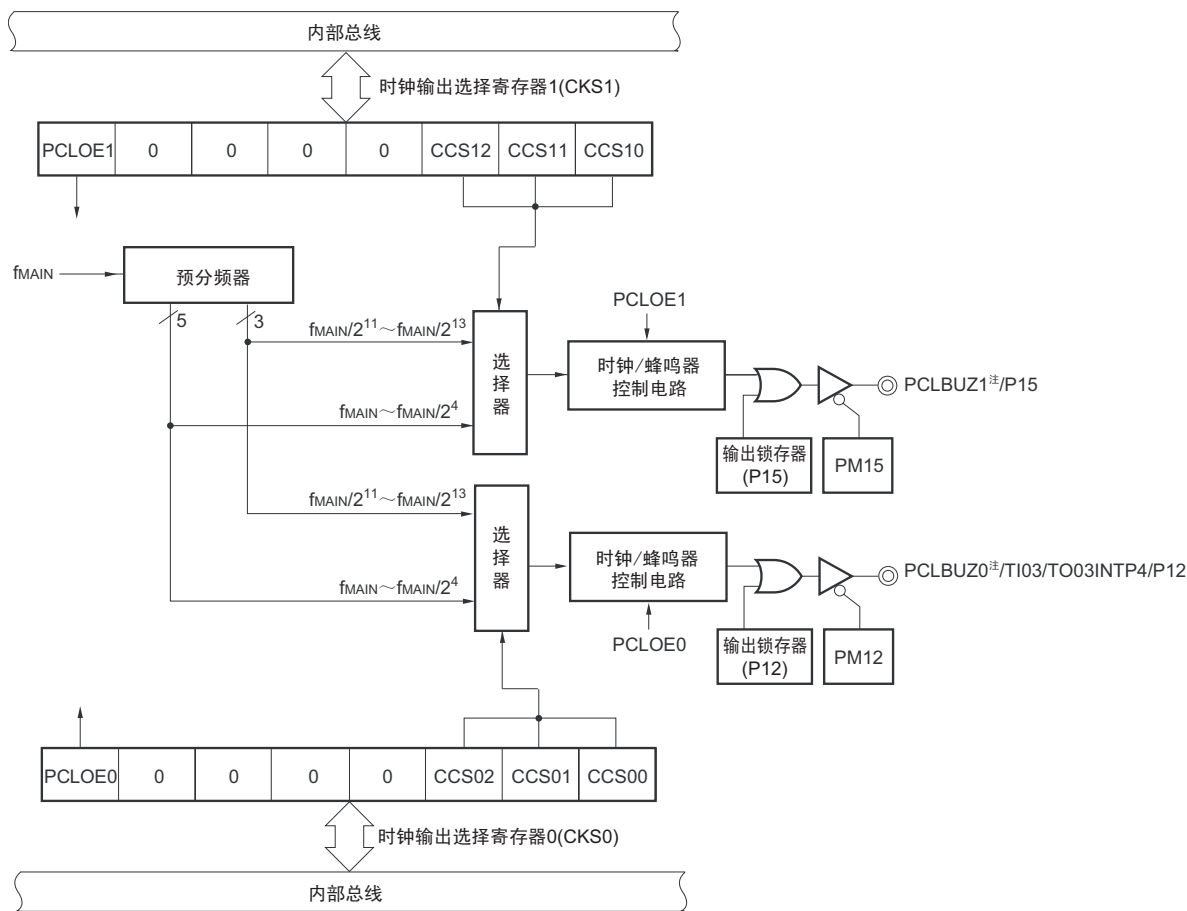
PCLBUZ $n$  引脚输出由时钟输出选择寄存器  $n$  (CKSn) 选择的时钟。

时钟输出 / 蜂鸣器输出控制电路的框图如图 7-1 所示。

备注  $n=0、1$



图 7-1 时钟输出 / 蜂鸣器输出控制电路的框图



注 有关能从 PCLBUZ0 引脚和 PCLBUZ1 引脚输出的频率，请参照“27.4 AC 特性”。

7.2 时钟输出 / 蜂鸣器输出控制电路的结构

时钟输出 / 蜂鸣器输出控制电路由以下硬件构成。

表 7-1 时钟输出 / 蜂鸣器输出控制电路的结构

项目	结构
控制寄存器	时钟输出选择寄存器 n（CKSn） 端口模式寄存器 1（PM1） 端口寄存器 1（P1）

## 7.3 控制时钟输出 / 蜂鸣器输出控制电路的寄存器

### 7.3.1 时钟输出选择寄存器 n (CKSn)

这是允许或者禁止时钟输出引脚或者蜂鸣器频率输出引脚 (PCLBUZn) 的输出以及设定输出时钟的寄存器。

通过 CKSn 寄存器选择 PCLBUZn 引脚输出的时钟。

通过 1 位或者 8 位存储器操作指令设定 CKSn 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 7-2 时钟输出选择寄存器 n (CKSn) 的格式

地址: FFFA5H (CKS0)、FFFA6H (CKS1)      复位后: 00H      R/W

符号	7	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	0	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn 引脚输出允许 / 禁止的指定
0	禁止输出 (默认值)。
1	允许输出。

CCSn2	CCSn1	CCSn0		PCLBUZn 引脚输出时钟的选择			
				$f_{\text{MAIN}}=5\text{MHz}$	$f_{\text{MAIN}}=10\text{MHz}$	$f_{\text{MAIN}}=20\text{MHz}$	$f_{\text{MAIN}}=32\text{MHz}$
0	0	0	$f_{\text{MAIN}}$	5MHz	禁止设定注	禁止设定注	禁止设定注
0	0	1	$f_{\text{MAIN}}/2$	2.5MHz	5MHz	禁止设定注	禁止设定注
0	1	0	$f_{\text{MAIN}}/2^2$	1.25MHz	2.5MHz	5MHz	8MHz 注
0	1	1	$f_{\text{MAIN}}/2^3$	625kHz	1.25MHz	2.5MHz	4MHz
1	0	0	$f_{\text{MAIN}}/2^4$	312.5kHz	625kHz	1.25MHz	2MHz
1	0	1	$f_{\text{MAIN}}/2^{11}$	2.44kHz	4.88kHz	9.77kHz	15.63kHz
1	1	0	$f_{\text{MAIN}}/2^{12}$	1.22kHz	2.44kHz	4.88kHz	7.81kHz
1	1	1	$f_{\text{MAIN}}/2^{13}$	610Hz	1.22kHz	2.44kHz	3.91kHz

注 必须在 8MHz 以内的范围内使用输出时钟。详细内容请参照“27.4 AC 特性”。

注意 1. 输出时钟的切换必须在设定为禁止输出 (PCLOEn=0) 后进行。

2. 在选择主系统时钟 (CSELn=0) 时, 如果要转移到 STOP 模式, 就必须在执行 STOP 指令前将 PCLOEn 置“0”。

备注 1. n=0、1

2.  $f_{\text{MAIN}}$  : 主系统时钟频率

### 7.3.2 控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器

在用作时钟输出 / 蜂鸣器输出功能时，必须设定与对象通道复用的端口功能的控制寄存器（端口模式寄存器（PM<sub>xx</sub>）和端口寄存器（P<sub>xx</sub>））。详细内容请参照“4.3.1 端口模式寄存器（PM<sub>xx</sub>）”和“4.3.2 端口寄存器（P<sub>xx</sub>）”。

在将时钟输出 / 蜂鸣器输出引脚的复用端口（P12/TI03/TO03/INTP4/PCLBUZ0、P15/PCLBUZ1 等）用作时钟输出 / 蜂鸣器输出时，必须将各端口对应的端口模式寄存器（PM<sub>xx</sub>）的位和端口寄存器（P<sub>xx</sub>）的位置“0”。

（例）将 P12/TO03/INTP4/PCLBUZ0 用作时钟输出 / 蜂鸣器输出的情况

将端口模式寄存器 1 的 PM12 位置“0”。

将端口寄存器 1 的 P12 位置“0”。

## 7.4 时钟输出 / 蜂鸣器输出控制电路的运行

能用 1 个引脚选择用作时钟输出或者蜂鸣器输出。

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

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

### 7.4.1 输出引脚的运行

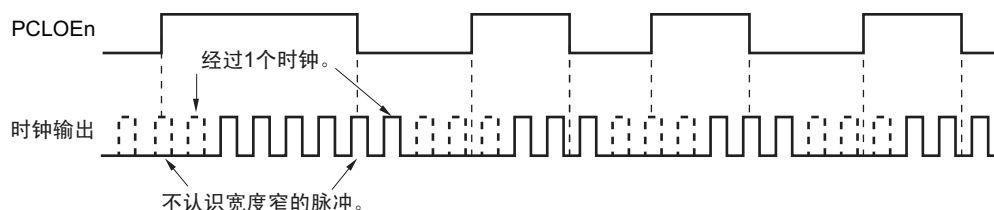
PCLBUZ<sub>n</sub> 引脚按照以下步骤进行输出：

- ① 将用作 PCLBUZ0 引脚的端口对应的端口模式寄存器 (PM<sub>xx</sub>) 和端口寄存器 (P<sub>xx</sub>) 的位置“0”。
- ② 通过 PCLBUZ<sub>n</sub> 引脚的时钟输出选择寄存器 (CKS<sub>n</sub>) 的 bit0 ~ 3 (CCS<sub>n0</sub> ~ CCS<sub>n2</sub>) 选择输出频率 (输出为禁止状态)。
- ③ 将 CKS<sub>n</sub> 寄存器的 bit7 (PCLOEn) 置“1”，允许时钟/蜂鸣器的输出。

备注 1. 用作时钟输出时的控制电路在允许或者禁止时钟输出 (PCLOEn 位) 后的 1 个时钟之后，开始或者停止时钟输出。此时不输出宽度窄的脉冲。通过 PCLOEn 位允许或者停止输出以及时钟输出的时序如图 7-3 所示。

2. n=0、1

图 7-3 PCLBUZ<sub>n</sub> 引脚的时钟输出时序



## 7.5 时钟输出 / 蜂鸣器输出控制电路的注意事项

当选择主系统时钟作为 PCLBUZ<sub>n</sub> 输出时，如果在设定停止输出 (PCLOEn=0) 后的 1.5 个 PCLBUZ<sub>n</sub> 引脚的输出时钟内转移到 STOP/HALT 模式，PCLBUZ<sub>n</sub> 的输出宽度就变窄。

## 第 8 章 看门狗定时器

### 8.1 看门狗定时器的功能

看门狗定时器通过选项字节（000C0H）设定计数运行。

看门狗定时器用于检测程序失控。在检测到程序失控时，产生内部复位信号。

下述情况判断为程序失控。

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

当因看门狗定时器而发生复位时，将复位控制标志寄存器（RESF）的 bit4（WDTRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

当达到上溢时间的  $75\% + 1/2f_{IL}$  时，能产生间隔中断。

## 8.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 8-1 看门狗定时器的结构

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

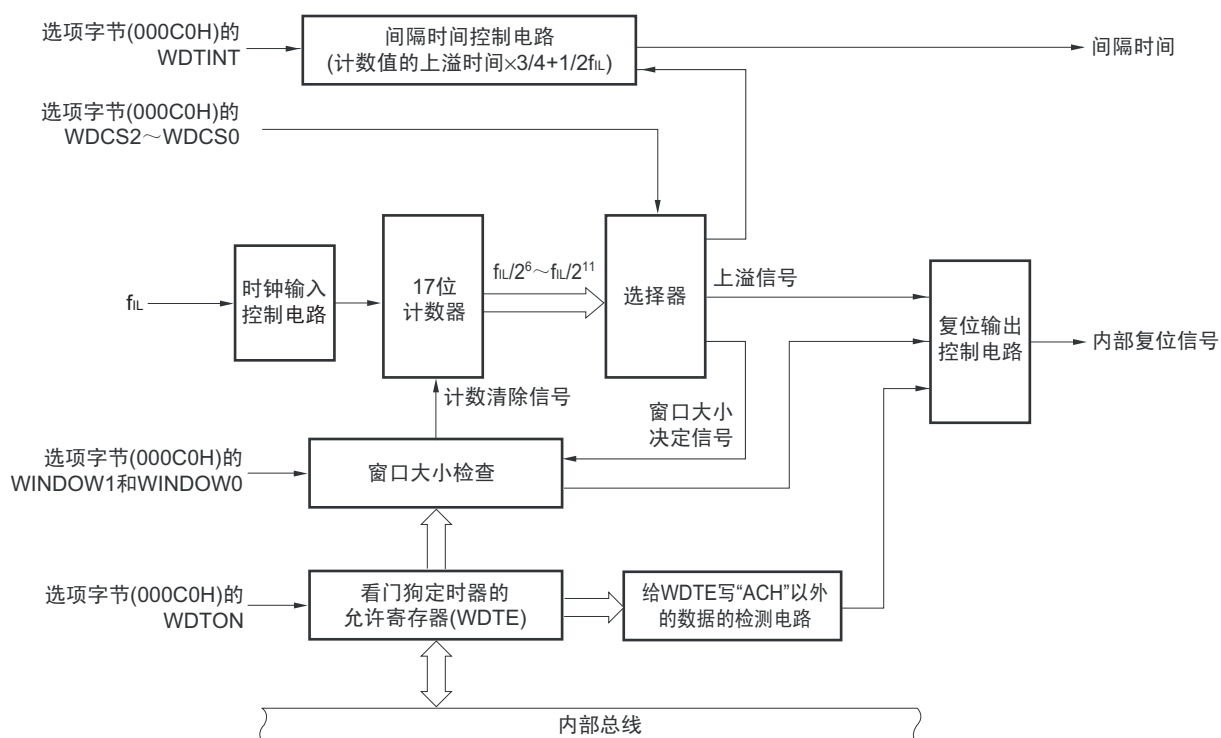
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表 8-2 选项字节和看门狗定时器的设定内容

看门狗定时器的设定内容	选项字节（000C0H）
看门狗定时器的间隔中断的设定	bit7（WDTINT）
窗口打开期间的设定	bit6 和 bit5（WINDOW1、WINDOW0）
看门狗定时器的计数器运行控制	bit4（WDTON）
看门狗定时器的上溢时间的设定	bit3 ~ 1（WDSC2 ~ WDSC0）
看门狗定时器的计数器运行控制（HALT/STOP 时）	bit0（WDSTBYON）

备注 有关选项字节，请参照“第 22 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

通过看门狗定时器的允许寄存器（WDTE）控制看门狗定时器。

#### 8.3.1 看门狗定时器的允许寄存器（WDTE）

通过给 WDTE 寄存器写“ACH”，清除看门狗定时器的计数器并且重新开始计数。

通过 8 位存储器操作指令设定 WDTE 寄存器。

在产生复位信号后，此寄存器的值变为“9AH”或者“1AH”注。

图 8-2 看门狗定时器的允许寄存器（WDTE）的格式

地址: FFFABH		复位后: 9AH/1AH 注			R/W			
符号	7	6	5	4	3	2	1	0
WDTE								

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

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

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

## 8.4 看门狗定时器的运行

### 8.4.1 看门狗定时器的运行控制

- 当使用看门狗定时器时，通过选项字节（000C0H）设定以下内容：
  - 必须将选项字节（000C0H）的 bit4（WDTON）置“1”，允许看门狗定时器的计数运行（在解除复位后，计数器开始运行）（详细内容请参照第 22 章）。

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

- 必须通过选项字节（000C0H）的 bit3～1（WDCS2～WDCS0）设定上溢时间（详细内容请参照 8.4.2 和第 22 章）。
  - 必须通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照 8.4.3 和第 22 章）。
- 在解除复位后，看门狗定时器开始计数。
  - 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
  - 此后，解除复位后第 2 次以后的 WDTE 寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写 WDTE 寄存器，就产生内部复位信号。
  - 如果不给 WDTE 寄存器写“ACH”而超过上溢时间，就产生内部复位信号。  
以下情况会产生内部复位信号：
    - 当对 WDTE 寄存器执行位操作指令时
    - 当给 WDTE 寄存器写“ACH”以外的数据时

- 注意 1. 只在解除复位后第 1 次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任何时刻写 WDTE，就清除看门狗定时器并且重新开始计数。
- 从给 WDTE 寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大 2 个  $f_{\text{IL}}$  时钟的误差。
  - 在计数值发生上溢前，都能清除看门狗定时器。
  - 如下所示，看门狗定时器在 HALT 模式、STOP 模式或者 SNOOZE 模式中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

	WDSTBYON=0	WDSTBYON=1
HALT 模式	停止看门狗定时器运行。	继续看门狗定时器运行。
STOP 模式		
SNOOZE 模式		

当 WDSTBYON 位为“0”时，在解除 HALT 模式或者 STOP 模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。

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

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



### 8.4.2 看门狗定时器上溢时间的设定

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

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

能设定的上溢时间如下所示。

表 8-3 看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ( $f_{IL}=17.25\text{kHz}(\text{MAX.})$ 的情况)
0	0	0	$2^6/f_{IL}$ (3.71ms)
0	0	1	$2^7/f_{IL}$ (7.42ms)
0	1	0	$2^8/f_{IL}$ (14.84ms)
0	1	1	$2^9/f_{IL}$ (29.68ms)
1	0	0	$2^{11}/f_{IL}$ (118.72ms)
上述以外			禁止设定。

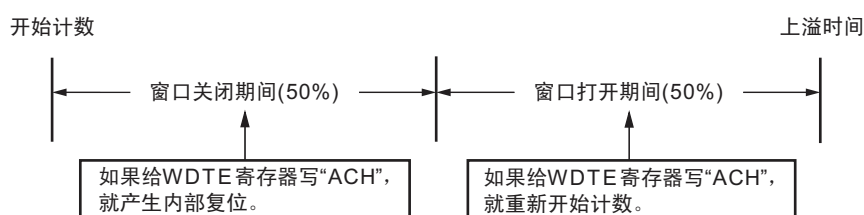
备注  $f_{IL}$ : 低速内部振荡器的时钟频率

### 8.4.3 看门狗定时器窗口打开期间的设定

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

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

例 窗口打开期间为 50% 的情况



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

能设定的窗口打开期间如下所示。

表 8-4 看门狗定时器窗口打开期间的设定

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	0	禁止设定。
0	1	50%
1	0	75%
1	1	100%

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

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

	窗口打开期间的设定		
	50%	75%	100%
窗口关闭时间	0 ~ 20.08ms	0 ~ 10.04ms	无
窗口打开时间	20.08 ~ 29.68ms	10.04 ~ 29.68ms	0 ~ 29.68ms

<当窗口打开期间为 50% 时>

- 上溢时间：  
 $2^9/f_{IL}(\text{MAX.}) = 2^9/17.25\text{kHz} = 29.68\text{ms}$
- 窗口关闭时间：  
 $0 \sim 2^9/f_{IL}(\text{MIN.}) \times (1-0.5) = 0 \sim 2^9/12.75\text{kHz} \times 0.5 = 0 \sim 20.08\text{ms}$
- 窗口打开时间：  
 $2^9/f_{IL}(\text{MIN.}) \times (1-0.5) \sim 2^9/f_{IL}(\text{MAX.}) = 2^9/12.75\text{kHz} \times 0.5 \sim 2^9/17.25\text{kHz} = 20.08 \sim 29.68\text{ms}$

#### 8.4.4 看门狗定时器间隔中断的设定

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

表 8-5 看门狗定时器间隔中断的设定

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

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

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

## 第 9 章 A/D 转换器

A/D 转换器的模拟输入通道数因产品而不同。

	24 引脚	32 引脚
模拟输入通道	6ch (ANI0 ~ ANI3、ANI7、ANI6)	8ch (ANI0 ~ ANI7)

备注 本章的下述内容主要针对 32 引脚产品进行说明。

### 9.1 A/D 转换器的功能

A/D 转换器是将模拟输入转换为数字值的转换器，最多能选择 8 个通道的 A/D 转换器模拟输入（ANI0 ~ ANI7）。能通过 A/D 转换器的模式寄存器 2（ADM2）的 ADTYP 位选择 12 位分辨率和 8 位分辨率。

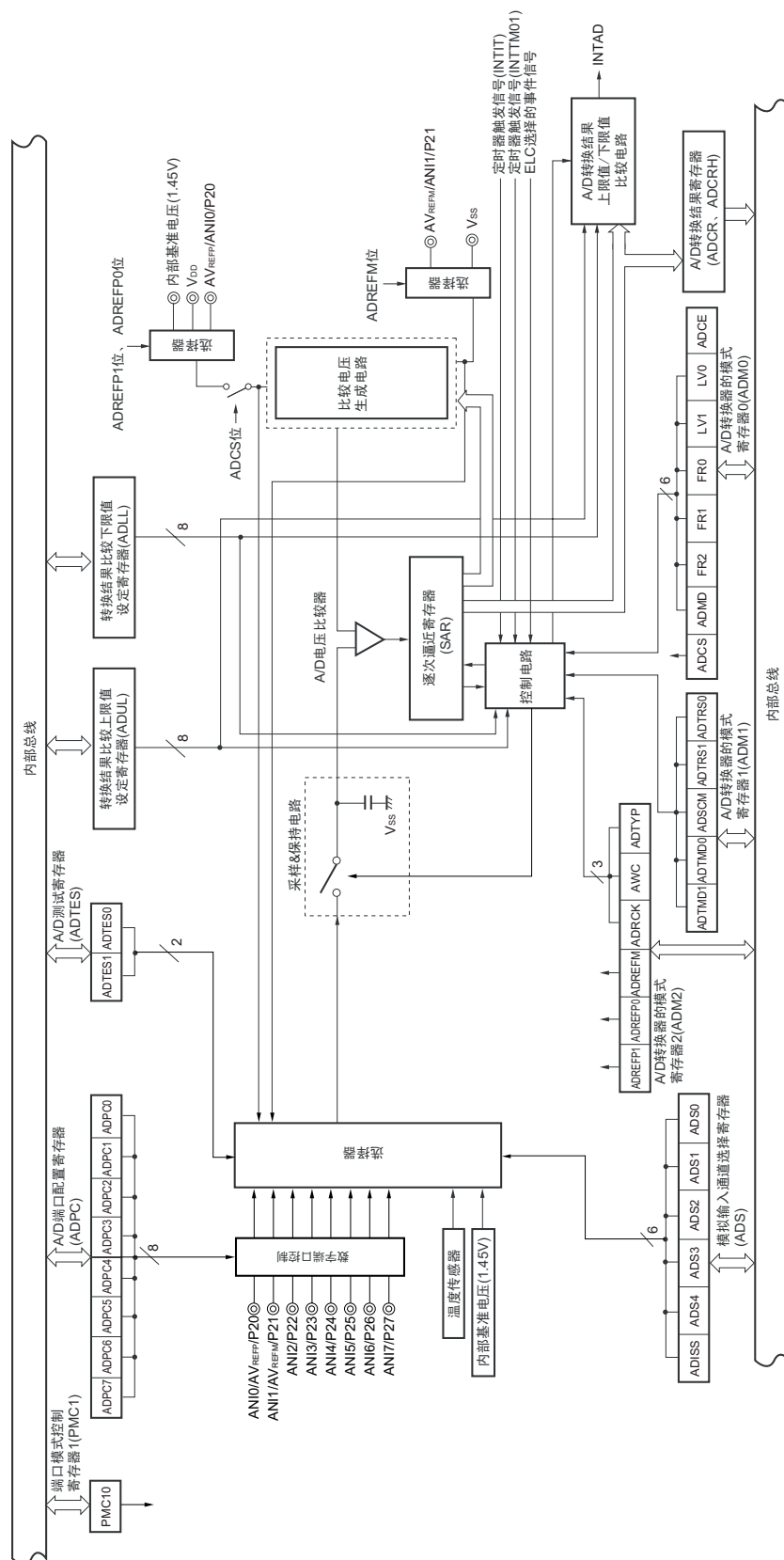
A/D 转换器有以下功能。

- 12 位或者 8 位分辨率的 A/D 转换  
从 ANI0 ~ ANI7 和 ANI16 中选择 1 个通道的模拟输入，重复进行 12 位或者 8 位分辨率的 A/D 转换。每结束 1 次 A/D 转换，就产生中断请求（INTAD）（选择模式的情况）。

能通过下述的模式组合设定各种 A/D 转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断电源的转换待机状态下，通过检测硬件触发来接通电源，在经过 A/D 电源稳定等待时间后自动开始转换。当使用 SNOOZE 模式功能时，必须选择硬件触发等待模式。
通道选择模式	选择模式	对所选 1 个通道的模拟输入进行 A/D 转换。
	扫描模式	按顺序对 4 个通道的模拟输入进行 A/D 转换。能选择 ANI0 ~ ANI7 中连续的 4 个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行 1 次 A/D 转换。
	连续转换模式	对所选通道进行连续的 A/D 转换，直到被软件停止为止。
工作电压模式	标准 1/ 标准 2 模式	用于在 $2.7V \leq V_{DD} \leq 3.6V$ 的工作电压范围内转换的情况。
采样时间的选择	采样时钟数：7 个 $f_{AD}$	标准 1 模式的采样时间为 7 个转换时钟（ $f_{AD}$ ）。用于因模拟输入源的输出阻抗高而想要延长采样时间的情况。
	采样时钟数：5 个 $f_{AD}$	标准 2 模式的采样时间为 5 个转换时钟（ $f_{AD}$ ）。用于因模拟输入源的输出阻抗低而能充分确保采样时间的情况。

图 9-1 A/D 转换器的框图



备注 此图的模拟输入引脚是 32 引脚产品的情况。

## 9.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

### (1) ANI0 ~ ANI7 引脚

这些是 A/D 转换器的 8 个通道的模拟输入引脚，输入要进行 A/D 转换的模拟信号。没有被选择为模拟输入的引脚能用作输入 / 输出端口。

### (2) 采样 & 保持电路

采样 & 保持电路依次对来自输入电路的模拟输入电压进行采样并且送给 A/D 电压比较器。在 A/D 转换过程中，保持采样到的模拟输入电压。

### (3) A/D 电压比较器

通过 A/D 电压比较器将采样的电压值和比较电压生成电路的分接头电压的输出值进行比较。如果比较结果是模拟输入电压大于基准电压 ( $1/2AV_{REF}$ )，就将逐次逼近寄存器 (SAR) 的最高位 (MSB) 置位。如果模拟输入电压小于基准电压 ( $1/2AV_{REF}$ )，就对 SAR 寄存器的 MSB 位进行复位。

然后，自动将 SAR 寄存器的 bit10 置位并且进行下一次比较。在此，根据 bit11 被设定的结果值，选择比较电压生成电路的分接头电压。

bit11=0: ( $1/4AV_{REF}$ )

bit11=1: ( $3/4AV_{REF}$ )

将比较电压生成电路的分接头电压和模拟输入电压进行比较，根据比较结果操作 SAR 寄存器的 bit10。

模拟输入电压  $\geq$  比较电压生成电路的分接头电压: bit10=1

模拟输入电压  $\leq$  比较电压生成电路的分接头电压: bit10=0

将此比较继续进行到 SAR 寄存器的 bit0 为止。

当以 8 位分辨率进行 A/D 转换时，比较继续进行到 SAR 寄存器的 bit4 为止。

备注  $AV_{REF}$  是 A/D 转换器的正 (+) 基准电压 (可选择  $AV_{REFP}$ 、内部基准电压 (1.45V) 或者  $AV_{DD}$ )。

### (4) 比较电压生成电路

比较电压生成电路生成模拟输入电压的比较电压。

#### (5) 逐次逼近寄存器 (SAR: Successive Approximation Register)

SAR 寄存器从最高位 (MSB) 按位设定比较电压生成电路的分接头电压值和模拟输入引脚的电压值相同的数据。

如果设定到 SAR 寄存器的最低位 (LSB) (A/D 转换结束), 就将 SAR 寄存器的内容 (转换结果) 保持到 A/D 转换结果寄存器 (ADCR)。而且, 如果指定的全部 A/D 转换都结束, 就产生 A/D 转换结束中断请求信号 (INTAD)。

#### (6) 12 位 A/D 转换结果寄存器 (ADCR)

每当 A/D 转换结束时, 就将逐次逼近寄存器的转换结果装入 ADCR 寄存器, A/D 转换结果保持在低 12 位 (高 4 位固定为“0”)。

#### (7) 8 位 A/D 转换结果寄存器 (ADCRH)

每当 A/D 转换结束时, 就将逐次逼近寄存器的转换结果装入 ADCRH 寄存器, 保存 A/D 转换结果的高 8 位。

#### (8) 控制电路

此电路控制要进行 A/D 转换的模拟输入的转换时间、转换的开始或者停止等。当 A/D 转换结束时, 通过 A/D 转换结果上限值 / 下限值比较电路产生 INTAD 信号。

#### (9) $AV_{REFP}$ 引脚

这是从外部输入基准电压 ( $AV_{REFP}$ ) 的引脚。

要将  $AV_{REFP}$  用作 A/D 转换器的正 (+) 基准电压时, 必须将 A/D 转换器的模式寄存器 2 (ADM2) 的 ADREFP1 位和 ADREFP0 位置“1”。

根据加载的  $AV_{REFP}$  和负 (-) 基准电压 ( $AV_{REFM}/V_{SS}$ ) 之间的电压, 将 ANI0 ~ ANI7 输入的模拟信号转换为数字信号。

除了  $AV_{REFP}$  以外, 可选择  $V_{DD}$  或者内部基准电压 (1.45V) 作为 A/D 转换器的正 (+) 基准电压。

#### (10) $AV_{REFM}$ 引脚

这是从外部输入基准电压 ( $AV_{REFM}$ ) 的引脚。要将  $AV_{REFM}$  用作 A/D 转换器的负 (-) 基准电压时, 必须将 ADM2 寄存器的 ADREFM 位置“1”。

除了  $AV_{REFM}$  以外, 可选择  $V_{SS}$  作为 A/D 转换器的负 (-) 基准电压。

### 9.3 控制 A/D 转换器的寄存器

控制 A/D 转换器的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- A/D转换器的模式寄存器0 (ADM0)
- A/D转换器的模式寄存器1 (ADM1)
- A/D转换器的模式寄存器2 (ADM2)
- 12位A/D转换结果寄存器 (ADCR)
- 8位A/D转换结果寄存器 (ADCRH)
- 模拟输入通道指定寄存器 (ADS)
- 转换结果比较上限值设定寄存器 (ADUL)
- 转换结果比较下限值设定寄存器 (ADLL)
- A/D测试寄存器 (ADTES)
- A/D端口配置寄存器 (ADPC)
- 端口模式控制寄存器1 (PMC1) 注
- 端口模式寄存器1、2 (PM1注、PM2)

注 只限于 24 引脚产品。



### 9.3.1 外围允许寄存器 0（PER0）

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

要使用 A/D 转换器时，必须将 bit5（ADCEN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-2 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	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

- 注意 1. 要设定 A/D 转换器时，必须先在 ADCEN 位为“1”的状态下设定以下的寄存器。当 ADCEN 位为“0”时，A/D 转换器的控制寄存器的值为初始值，忽视写操作（端口模式寄存器 1、2（PM1 注、PM2）、端口模式控制寄存器 1（PMC1）注以及 A/D 端口配置寄存器（ADPC）除外）。
- A/D 转换器的模式寄存器 0（ADM0）
  - A/D 转换器的模式寄存器 1（ADM1）
  - A/D 转换器的模式寄存器 2（ADM2）
  - 12 位 A/D 转换结果寄存器（ADCR）
  - 8 位 A/D 转换结果寄存器（ADCRH）
  - 模拟输入通道指定寄存器（ADS）
  - 转换结果比较上限值设定寄存器（ADUL）
  - 转换结果比较下限值设定寄存器（ADLL）
  - A/D 测试寄存器（ADTES）
2. 必须将 bit1、3、7 置“0”。

注 只限于 24 引脚产品。

### 9.3.2 A/D 转换器的模式寄存器 0 (ADM0)

这是设定要进行 A/D 转换的模拟输入的转换时间、转换开始或者停止的寄存器。

通过 1 位或者 8 位存储器操作指令设定 ADM0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-3 A/D 转换器的模式寄存器 0 (ADM0) 的格式

地址: FFF30H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADM0	ADCS	ADMD	FR2 注 1	FR1 注 1	FR0 注 1	0	LV0 注 1	ADCE

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

ADMD	A/D 转换通道选择模式的设定
0	选择模式
1	扫描模式

ADCE	A/D 电压比较器的运行控制注 2
0	停止 A/D 电压比较器的运行。
1	允许 A/D 电压比较器的运行。

注 1. 有关 FR2 ~ FR0 位、LV0 位和 A/D 转换的详细内容，请参照“表 9-3 A/D 转换时间的选择”。

2. 在软件触发模式或者硬件触发无等待模式中，ADCS 位和 ADCE 位控制 A/D 电压比较器的运行，从开始运行到稳定为止需要稳定等待时间。因此，在将 ADCE 位置“1”后经过稳定等待时间，然后将 ADCS 位置“1”，从而最初的转换数据开始有效。如果不经稳定等待时间而将 ADCS 位置“1”，就必须忽视最初的转换数据。

[ 稳定等待时间 ]

选择高精度通道作为模拟输入通道的情况 : 0.5μs

设定测试模式 (ADTES 寄存器的 ADTES1=1) 的情况 : 0.5μs

选择标准通道作为模拟输入通道的情况 : 2μs

选择温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 作为模拟输入通道的情况: 2μs

注意 1. 必须在转换停止状态 (ADCS=0、ADCE=0) 下更改 ADMD 位、FR2 ~ FR0 位和 LV0 位。

2. 禁止进行 ADCS=1、ADCE=0 的设定。

3. 禁止通过 8 位操作指令将 ADCE=0、ADCS=0 的状态设定为 ADCE=1、ADCS=1。必须按照“9.7 A/D 转换器的设定流程图”的步骤进行设定。

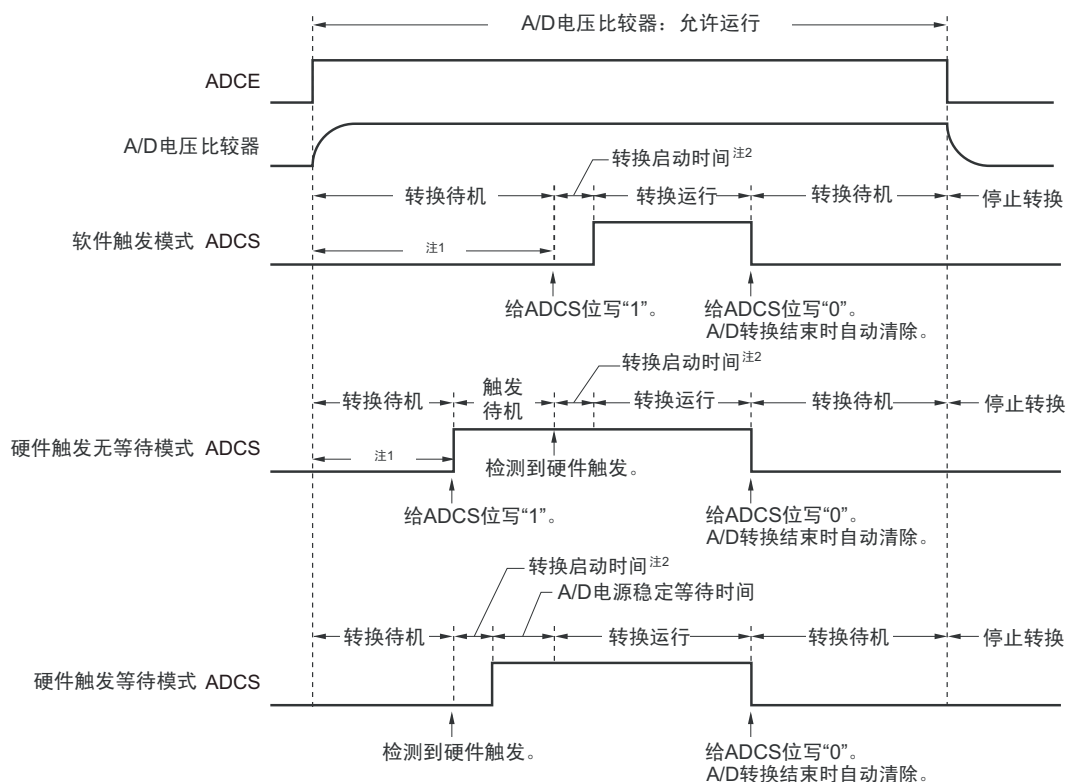
表 9-1 ADCS 位和 ADCE 位的设定

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

表 9-2 ADCS 位的置位和清除条件

A/D 转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给 ADCE 位和 ADCS 位写“1”时	当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> <li>当给 ADCS 位写“0”时</li> <li>在 A/D 转换结束时自动清“0”。</li> </ul>
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> <li>当给 ADCS 位写“0”时</li> <li>当设定的 4 个通道转换结束时，自动清“0”。</li> </ul>
硬件触发无等待模式	选择模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		当给 ADCS 位写“0”时
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		当给 ADCS 位写“0”时
硬件触发等待模式	选择模式	连续转换模式	当 ADCE 位为“1”并且输入硬件触发时	当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> <li>当给 ADCS 位写“0”时</li> <li>在 A/D 转换结束时自动清“0”。</li> </ul>
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> <li>当给 ADCS 位写“0”时</li> <li>当设定的 4 个通道转换结束时，自动清“0”。</li> </ul>

图 9-4 使用 A/D 电压比较器时的时序图



注 1. 在软件触发模式或者硬件触发无等待模式中，为了稳定内部电路，从 ADCE 位上升到 ADCS 位上升的时间至少需要以下的稳定等待时间。

[ 稳定等待时间 ]

选择高精度通道作为模拟输入通道的情况 : 0.5 $\mu$ s

设定测试模式 (ADTES 寄存器的 ADTES1=1) 的情况 : 0.5 $\mu$ s

选择标准通道作为模拟输入通道的情况 : 2 $\mu$ s

选择温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 作为模拟输入通道的情况: 2 $\mu$ s

2. 在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中，在检测到硬件触发后不发生 A/D 电源稳定等待时间。

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

2. 在硬件触发无等待模式的单次转换模式中，ADCS 标志在 A/D 转换结束时不自动清“0”，而保持“1”的状态。

3. 必须在 ADCS 位为“0”（停止转换 / 转换待机状态）时改写 ADCE 位。

4. 为了结束 A/D 转换，必须至少将硬件触发间隔设定为以下时间：

硬件触发无等待模式时：2 个  $f_{CLK}$  时钟 + 转换启动时间 + A/D 转换时间

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

备注  $f_{CLK}$ : CPU/ 外围硬件的时钟频率

表 9-3 A/D 转换时间的选择 (1/4)

(1) 12 位 A/D 转换 无稳定等待时间 (软件触发模式 / 硬件触发无等待模式)

A/D 转换器的模式 寄存器 0 (ADM0)				模式	转换时钟 ( $f_{AD}$ )	转换时钟 数 (采样 时钟数)	转换时间	转换时间的选择							
								$V_{DD}=2.7 \sim 3.6V$							
FR2	FR1	FR0	LV0					$f_{CLK}=1MHz$	$f_{CLK}=4MHz$	$f_{CLK}=8MHz$	$f_{CLK}=16MHz$	$f_{CLK}=32MHz$			
0	0	0	0	标准 1	$f_{CLK}/32$	54 个 $f_{AD}$ (采样时 钟数: 11 个 $f_{AD}$ )	$1728/f_{CLK}$	禁止设定	禁止设定	禁止设定	禁止设定	54 $\mu s$			
0	0	1			$f_{CLK}/16$					$864/f_{CLK}$		54 $\mu s$	27 $\mu s$		
0	1	0			$f_{CLK}/8$					$432/f_{CLK}$		54 $\mu s$	27 $\mu s$	13.5 $\mu s$	
0	1	1			$f_{CLK}/6$					$324/f_{CLK}$		40.5 $\mu s$	20.25 $\mu s$	10.125 $\mu s$	
1	0	0			$f_{CLK}/5$					$270/f_{CLK}$		33.75 $\mu s$	16.875 $\mu s$	8.4375 $\mu s$	
1	0	1			$f_{CLK}/4$					$216/f_{CLK}$		54 $\mu s$	27 $\mu s$	13.5 $\mu s$	6.75 $\mu s$
1	1	0			$f_{CLK}/2$					$108/f_{CLK}$		27 $\mu s$	13.5 $\mu s$	6.75 $\mu s$	3.375 $\mu s$
1	1	1			$f_{CLK}/1$					$54/f_{CLK}$	54 $\mu s$	13.5 $\mu s$	6.75 $\mu s$	3.375 $\mu s$	禁止设定
0	0	0	1	标准 2	$f_{CLK}/32$	66 个 $f_{AD}$ (采样时 钟数: 23 个 $f_{AD}$ )	$2112/f_{CLK}$	禁止设定	禁止设定	禁止设定	禁止设定	66 $\mu s$			
0	0	1			$f_{CLK}/16$					$1056/f_{CLK}$		66 $\mu s$	33 $\mu s$		
0	1	0			$f_{CLK}/8$					$528/f_{CLK}$		66 $\mu s$	33 $\mu s$	16.5 $\mu s$	
0	1	1			$f_{CLK}/6$					$396/f_{CLK}$		49.5 $\mu s$	24.75 $\mu s$	12.375 $\mu s$	
1	0	0			$f_{CLK}/5$					$330/f_{CLK}$		41.25 $\mu s$	20.625 $\mu s$	10.3125 $\mu s$	
1	0	1			$f_{CLK}/4$					$264/f_{CLK}$		66 $\mu s$	33 $\mu s$	16.5 $\mu s$	8.25 $\mu s$
1	1	0			$f_{CLK}/2$					$132/f_{CLK}$		33 $\mu s$	16.5 $\mu s$	8.25 $\mu s$	4.125 $\mu s$
1	1	1			$f_{CLK}/1$					$66/f_{CLK}$	66 $\mu s$	16.5 $\mu s$	8.25 $\mu s$	4.125 $\mu s$	禁止设定

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 ( $t_{CONV}$ ) 范围内。

- 要将 FR2 ~ FR0 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 上述转换时间不包含时钟频率的误差, 因此在选择转换时间时必须考虑时钟频率的误差。
- 必须根据以下条件设定软件触发模式或者硬件触发无等待模式中的转换时间。
  - $f_{AD}$  为 1 ~ 16MHz 的范围。
  - 在将温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 设定为模拟输入通道时, 为以下条件:  
当 LV0=0 时, 禁止设定。  
当 LV0=1 时, 能设定。

备注  $f_{CLK}$ : CPU/ 外围硬件的时钟频率

表 9-3 A/D 转换时间的选择 (2/4)

(2) 12 位 A/D 转换 有稳定等待时间 (硬件触发等待模式 (连续转换模式的第 2 次以后和扫描模式的扫描 1 以后除外注))

A/D 转换器的模式寄存器 0 (ADM0)				模式	转换时钟 (f <sub>AD</sub> )	稳定等待时钟数	转换时钟数 (采样时钟数)	稳定等待时间 + 转换时间	A/D 电源稳定等待时间 + 转换时间的选择							
FR2	FR1	FR0	LV0						V <sub>DD</sub> =2.7 ~ 3.6V							
									f <sub>CLK</sub> =1MHz	f <sub>CLK</sub> =4MHz	f <sub>CLK</sub> =8MHz	f <sub>CLK</sub> =16MHz	f <sub>CLK</sub> =32MHz			
0	0	0	0	标准 1	f <sub>CLK</sub> /32	4 个 f <sub>CLK</sub>	54 个 f <sub>AD</sub> (采样时钟数: 11 个 f <sub>AD</sub> )	1732/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	54.125μs			
0	0	1			f <sub>CLK</sub> /16			868/f <sub>CLK</sub>					54.25μs	27.125μs		
0	1	0			f <sub>CLK</sub> /8			436/f <sub>CLK</sub>					54.5μs	27.25μs	13.625μs	
0	1	1			f <sub>CLK</sub> /6			328/f <sub>CLK</sub>					41μs	20.5μs	10.25μs	
1	0	0			f <sub>CLK</sub> /5			274/f <sub>CLK</sub>					34.25μs	17.125μs	8.5625μs	
1	0	1			f <sub>CLK</sub> /4			220/f <sub>CLK</sub>					55μs	27.5μs	13.75μs	6.875μs
1	1	0			f <sub>CLK</sub> /2			112/f <sub>CLK</sub>					28μs	14μs	7μs	3.5μs
1	1	1			f <sub>CLK</sub> /1	2 个 f <sub>CLK</sub>		56/f <sub>CLK</sub>	56μs	14μs	7μs	3.5μs	禁止设定			
0	0	0	1	标准 2	f <sub>CLK</sub> /32	58 个 f <sub>CLK</sub>	66 个 f <sub>AD</sub> (采样时钟数: 23 个 f <sub>AD</sub> )	2170/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	67.8125μs			
0	0	1			f <sub>CLK</sub> /16			1114/f <sub>CLK</sub>					69.625μs	34.8125μs		
0	1	0			f <sub>CLK</sub> /8			586/f <sub>CLK</sub>					73.25μs	36.625μs	18.3125μs	
0	1	1			f <sub>CLK</sub> /6			454/f <sub>CLK</sub>					56.75μs	28.375μs	14.1875μs	
1	0	0			f <sub>CLK</sub> /5			388/f <sub>CLK</sub>					48.5μs	24.25μs	12.125μs	
1	0	1			f <sub>CLK</sub> /4			322/f <sub>CLK</sub>					80.5μs	40.25μs	20.125μs	10.0625μs
1	1	0			f <sub>CLK</sub> /2			190/f <sub>CLK</sub>					47.5μs	23.75μs	11.875μs	5.9375μs
1	1	1			f <sub>CLK</sub> /1	29 个 f <sub>CLK</sub>		95/f <sub>CLK</sub>	95μs	23.75μs	11.875μs	5.9375μs	禁止设定			

注 在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中, 在检测到硬件触发后不发生 A/D 电源稳定等待时间 (参照表 9-3(1/4))。

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 (t<sub>CONV</sub>) 范围内。转换时间 (t<sub>CONV</sub>) 不包含 A/D 电源稳定等待时间。

- 要将 FR2 ~ FR0 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 上述转换时间不包含时钟频率的误差, 因此在选择转换时间时必须考虑时钟频率的误差。
- 必须根据以下条件设定硬件触发等待模式中的转换时间。
  - f<sub>AD</sub> 为 1 ~ 16MHz 的范围。
  - 在将温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 设定为模拟输入通道时, 为以下条件:
    - 当 LV0=0 时, 禁止设定。
    - 当 LV0=1 时, 能设定。

备注 f<sub>CLK</sub>: CPU/ 外围硬件的时钟频率

表 9-3 A/D 转换时间的选择 (3/4)  
(3) 8 位 A/D 转换 无稳定等待时间 (软件触发模式 / 硬件触发无等待模式)

A/D 转换器的模式 寄存器 0（ADM0）				模式	转换时钟 （f <sub>AD</sub> ）	转换时钟 数（采样 时钟数）	转换时间	转换时间的选择					
FR2	FR1	FR0	LV0					V <sub>DD</sub> =2.7 ~ 3.6V					
								f <sub>CLK</sub> = 1MHz	f <sub>CLK</sub> = 4MHz	f <sub>CLK</sub> = 8MHz	f <sub>CLK</sub> = 16MHz	f <sub>CLK</sub> = 32MHz	
0	0	0	0	标准 1	f <sub>CLK</sub> /32	41 个 f <sub>AD</sub> （采样时钟 数： 11 个 f <sub>AD</sub> ）	1312/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	41μs	
0	0	1			f <sub>CLK</sub> /16		656/f <sub>CLK</sub>				41μs	20.5μs	
0	1	0			f <sub>CLK</sub> /8		328/f <sub>CLK</sub>				41μs	20.5μs	10.25μs
0	1	1			f <sub>CLK</sub> /6		246/f <sub>CLK</sub>				30.75μs	15.375μs	7.6875μs
1	0	0			f <sub>CLK</sub> /5		205/f <sub>CLK</sub>				25.625μs	12.8125μs	6.40625μs
1	0	1			f <sub>CLK</sub> /4		164/f <sub>CLK</sub>			41μs	20.5μs	10.25μs	5.125μs
1	1	0			f <sub>CLK</sub> /2		82/f <sub>CLK</sub>			20.5μs	10.25μs	5.125μs	2.5625μs
1	1	1			f <sub>CLK</sub> /1		41/f <sub>CLK</sub>			41μs	10.25μs	5.125μs	2.5625μs
0	0	0	1	标准 2	f <sub>CLK</sub> /32	53 个 f <sub>AD</sub> （采样时钟 数： 23 个 f <sub>AD</sub> ）	1696/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	53μs	
0	0	1			f <sub>CLK</sub> /16		848/f <sub>CLK</sub>				53μs	26.5μs	
0	1	0			f <sub>CLK</sub> /8		424/f <sub>CLK</sub>				53μs	26.5μs	13.25μs
0	1	1			f <sub>CLK</sub> /6		318/f <sub>CLK</sub>				39.75μs	19.875μs	9.9375μs
1	0	0			f <sub>CLK</sub> /5		265/f <sub>CLK</sub>				33.125μs	16.5625μs	8.28125μs
1	0	1			f <sub>CLK</sub> /4		212/f <sub>CLK</sub>			53μs	26.5μs	13.25μs	6.625μs
1	1	0			f <sub>CLK</sub> /2		106/f <sub>CLK</sub>			26.5μs	13.25μs	6.625μs	3.3125μs
1	1	1			f <sub>CLK</sub> /1		53/f <sub>CLK</sub>			53μs	13.25μs	6.625μs	3.3125μs

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 ( $t_{CONV}$ ) 范围内。

2. 要将 FR2 ~ FR0 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
3. 上述转换时间不包含时钟频率的误差, 因此在选择转换时间时必须考虑时钟频率的误差。
4. 必须根据以下条件设定软件触发模式 / 硬件触发无等待模式中的转换时间。
  - $f_{AD}$  为 1 ~ 16MHz 的范围。
  - 在将温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 设定为模拟输入通道时, 为以下条件:  
当 LV0=0 时, 禁止设定。  
当 LV0=1 时, 能设定。

备注  $f_{CLK}$ : CPU / 外围硬件的时钟频率

表 9-3 A/D 转换时间的选择 (4/4)

(4) 8 位 A/D 转换 有稳定等待时间（硬件触发等待模式（连续转换模式的第 2 次以后和扫描模式的扫描 1 以后除外注））

A/D 转换器的模式寄存器 0 (ADM0)				模式	转换时钟 (f <sub>AD</sub> )	稳定等待时钟数	转换时钟数 (采样时钟数)	稳定等待时间 + 转换时间	A/D 电源稳定等待时间 + 转换时间的选择							
FR2	FR1	FR0	LV0						AV <sub>DD</sub> =2.7 ~ 3.6V							
									f <sub>CLK</sub> =1MHz	f <sub>CLK</sub> =4MHz	f <sub>CLK</sub> =8MHz	f <sub>CLK</sub> =16MHz	f <sub>CLK</sub> =32MHz			
0	0	0	0	标准 1	f <sub>CLK</sub> /32	4 个 f <sub>CLK</sub>	41 个 f <sub>AD</sub> (采样时钟数: 11 个 f <sub>AD</sub> )	1316/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	41.125μs			
0	0	1			f <sub>CLK</sub> /16			660/f <sub>CLK</sub>					41.25μs	20.625μs		
0	1	0			f <sub>CLK</sub> /8			332/f <sub>CLK</sub>					41.5μs	20.75μs	10.375μs	
0	1	1			f <sub>CLK</sub> /6			250/f <sub>CLK</sub>					31.25μs	15.625μs	7.8125μs	
1	0	0			f <sub>CLK</sub> /5			209/f <sub>CLK</sub>					26.125μs	13.0625μs	6.53125μs	
1	0	1			f <sub>CLK</sub> /4			168/f <sub>CLK</sub>					42μs	21μs	10.5μs	5.25μs
1	1	0			f <sub>CLK</sub> /2			86/f <sub>CLK</sub>					21.5μs	10.75μs	5.375μs	2.6875μs
1	1	1			f <sub>CLK</sub> /1	2 个 f <sub>CLK</sub>		43/f <sub>CLK</sub>	43μs	10.75μs	5.375μs	2.6875μs	禁止设定			
0	0	0	1	标准 2	f <sub>CLK</sub> /32	58 个 f <sub>CLK</sub>	53 个 f <sub>AD</sub> (采样时钟数: 23 个 f <sub>AD</sub> )	1754/f <sub>CLK</sub>	禁止设定	禁止设定	禁止设定	禁止设定	54.8125μs			
0	0	1			f <sub>CLK</sub> /16			906/f <sub>CLK</sub>					56.625μs	28.3125μs		
0	1	0			f <sub>CLK</sub> /8			482/f <sub>CLK</sub>					60.25μs	30.125μs	15.0625μs	
0	1	1			f <sub>CLK</sub> /6			376/f <sub>CLK</sub>					47μs	23.5μs	11.75μs	
1	0	0			f <sub>CLK</sub> /5			323/f <sub>CLK</sub>					40.375μs	20.1875μs	10.09375μs	
1	0	1			f <sub>CLK</sub> /4			270/f <sub>CLK</sub>					67.5μs	33.75μs	16.875μs	8.4375μs
1	1	0			f <sub>CLK</sub> /2			164/f <sub>CLK</sub>					41μs	20.5μs	10.25μs	5.125μs
1	1	1			f <sub>CLK</sub> /1	29 个 f <sub>CLK</sub>		82/f <sub>CLK</sub>	82μs	20.5μs	10.25μs	5.125μs	禁止设定			

注 在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中，在检测到硬件触发后不发生 A/D 电源稳定等待时间（参照表 9-3(3/4)）。

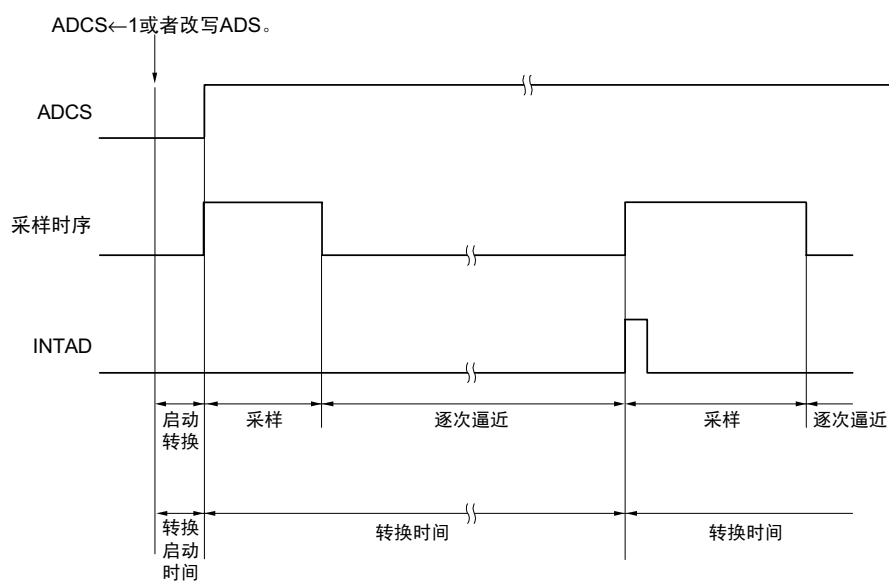
注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 ( $t_{\text{CONV}}$ ) 范围内。转换时间 ( $t_{\text{CONV}}$ ) 不包含 A/D 电源稳定等待时间。

- 要将 FR2 ~ FR0 位和 LV0 位改写为不同数据时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 上述转换时间不包含时钟频率的误差，因此在选择转换时间时必须考虑时钟频率的误差。
- 必须根据以下条件设定硬件触发等待模式中的转换时间。
  - $f_{\text{AD}}$  为 1 ~ 16MHz 的范围。
  - 在将温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 设定为模拟输入通道时，为以下条件：
    - 当 LV0=0 时，禁止设定。
    - 当 LV0=1 时，能设定。

备注  $f_{\text{CLK}}$ : CPU/ 外围硬件的时钟频率



图 9-5 A/D 转换器的采样和 A/D 转换时序（以软件触发模式为例的情况）



9.3.3      A/D 转换器的模式寄存器 1（ADM1）

这是设定 A/D 转换触发、转换模式和硬件触发信号的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 ADM1 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 9-6    A/D 转换器的模式寄存器 1（ADM1）的格式

地址：FFF32H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

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

ADSCM	A/D 转换模式的设定
0	连续转换模式
1	单次转换模式

ADTRS1	ADTRS0	硬件触发信号的选择
0	0	定时器通道 01 的计数结束或者捕捉结束中断信号（INTTM01）
0	1	ELC 选择的事件信号
上述以外		禁止设定。

- 注意 1. 要改写 ADM1 寄存器时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。
2. 为了结束 A/D 转换，必须至少将硬件触发间隔设定为以下时间：
- 硬件触发无等待模式时：2 个 f<sub>CLK</sub> 时钟 + 转换启动时间 + A/D 转换时间
- 硬件触发等待模式时：2 个 f<sub>CLK</sub> 时钟 + A/D 电源稳定等待时间 + A/D 转换时间

- 备注 1. ×：忽略
2. f<sub>CLK</sub>：CPU/ 外围硬件的时钟频率

### 9.3.4 A/D 转换器的模式寄存器 2 (ADM2)

这是选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压和分辨率并且检查 A/D 转换结果的上限值和下限值以及设定 SNOOZE 模式的寄存器。

通过 1 位或者 8 位存储器操作指令设定 ADM2 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-7 A/D 转换器的模式寄存器 2 (ADM2) 的格式 (1/2)

地址: F0010H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	A/D 转换器的正 (+) 基准电压源的选择
0	0	由 $V_{DD}$ 提供。
0	1	由 P20/AV <sub>REFP</sub> /ANI0 提供。
1	0	由内部基准电压 (1.45V) 提供注。
1	1	禁止设定。

• 要改写 ADREFP1 位和 ADREFP0 位时，必须按照以下步骤进行设定：

- ①将 ADCE 位置“0”。
- ②更改 ADREFP1 位和 ADREFP0 位的值。
- ③等待基准电压的稳定等待时间 (A)。
- ④将 ADCE 位置“1”。
- ⑤等待基准电压的稳定等待时间 (B)。

在更改 ADREFP1 位和 ADREFP0 位的值时，需要③的稳定等待时间。

当将 ADREFP1 位和 ADREFP0 位改为“1”和“0”时，A=10μs。

当将 ADREFP1 位和 ADREFP0 位改为“0”和“0”或者“0”和“1”时，A=1μs。

在将 ADCE 位置“1”时，需要⑤的稳定等待时间。

当选择高精度通道作为模拟输入通道时，B=0.5μs。

当设定测试模式 (ADTES 寄存器的 ADTES1=1) 时，B = 0.5μs。

当选择标准通道作为模拟输入通道时，B=2μs。

当选择温度传感器输出 / 内部基准电压输出 (ADS 寄存器的 ADISS=1) 作为模拟输入通道时，B=2μs。

必须在等待⑤的稳定时间后开始 A/D 转换。

• 当将 ADREFP1 位和 ADREFP0 位置“1”和“0”时，不能对温度传感器和内部基准电压的输出进行 A/D 转换。必须在 ADISS 位为“0”时进行 A/D 转换。

ADREFM	A/D 转换器的负 (-) 基准电压源的选择
0	由 $V_{SS}$ 提供。
1	由 P21/AV <sub>REFM</sub> /ANI1 提供。

注 只有在 HS (高速主) 模式中才能选择。详细内容请参照“图 22-3 用户选项字节 (000C2H) 的格式”。

注意 1. 要改写 ADM2 寄存器时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

2. 在 STOP 模式中，不能将 ADREFP1 位置“1”。在将 ADREFP1 位置“1”时，如果在 CPU 以主系统时钟运行中转移到 HALT 模式，就加上“27.3.2 电源电流特性”所示的 A/D 转换器基准电压电流 ( $I_{ADREF}$ ) 的电流值。

3. 当使用 AV<sub>REFP</sub> 和 AV<sub>REFM</sub> 时，必须将 ANI0 和 ANI1 设定为模拟输入，并且通过端口模式寄存器设定为输入模式。

图 9-7  A/D 转换器的模式寄存器 2（ADM2）的格式 (2/2)

地址：F0010H

复位后：00H

R/W

符号	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	转换结果上限值和下限值的检查
0	当 ADLL 寄存器 ≤ ADCR 寄存器 ≤ ADUL 寄存器（AREA1）时，产生中断信号（INTAD）。
1	当 ADCR 寄存器 < ADLL 寄存器（AREA2）或者 ADUL 寄存器 < ADCR 寄存器（AREA3）时，产生中断信号（INTAD）。
AREA1 ~ AREA3 的中断信号（INTAD）的产生范围如图 9-8 所示。	

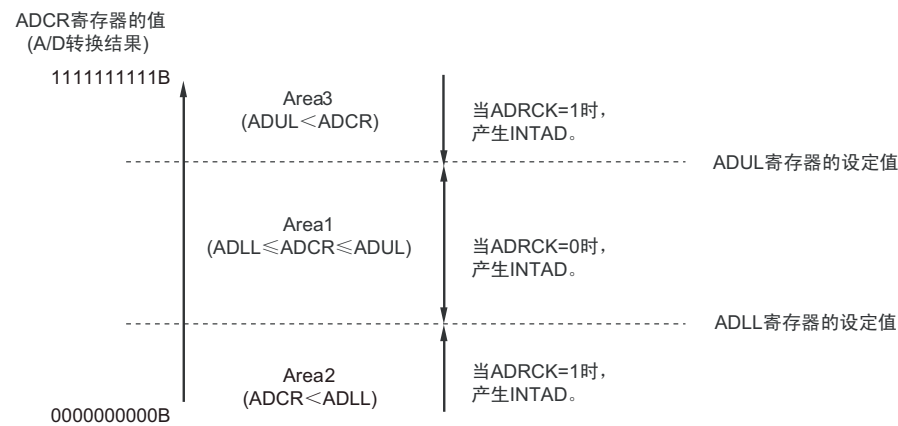
AWC	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
通过 STOP 模式中的硬件触发信号来解除 STOP 模式，并且在 CPU 不运行的情况下进行 A/D 转换（SNOOZE 模式）。 <ul style="list-style-type: none"><li>• 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件的时钟（f<sub>CLK</sub>）的情况下才能设定 SNOOZE 模式功能。在选择高速内部振荡器以外的时钟的情况下，禁止设定。</li><li>• 在软件触发模式或者硬件触发无等待模式中，禁止使用 SNOOZE 模式功能。</li><li>• 在连续转换模式中，禁止使用 SNOOZE 模式功能。</li><li>• 在使用 SNOOZE 模式功能时，必须至少将硬件触发间隔设定为“SNOOZE 模式的转移时间注 + 转换启动时间 + A/D 电源稳定等待时间 + A/D 转换时间 + 2 个 f<sub>CLK</sub> 时钟”。</li><li>• 即使使用 SNOOZE 模式，也必须在通常运行模式中将 AWC 位置“0”，并且在即将要转移到 STOP 模式前将 AWC 位改为“1”。</li></ul> 另外，必须在从 STOP 模式返回到通常运行模式后将 AWC 位改为“0”。 如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。	

ADTYP	A/D 转换分辨率的选择
0	12 位分辨率
1	8 位分辨率

注    请参照“16.3.3 SNOOZE 模式”的“STOP 模式 → SNOOZE 模式的转移时间”。

注意  要改写 ADM2 寄存器时，必须在转换停止状态（ADCS=0、ADCE=0）下进行。

图 9-8  ADRCK 位的中断信号产生范围



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

### 9.3.5 12 位 A/D 转换结果寄存器（ADCR）

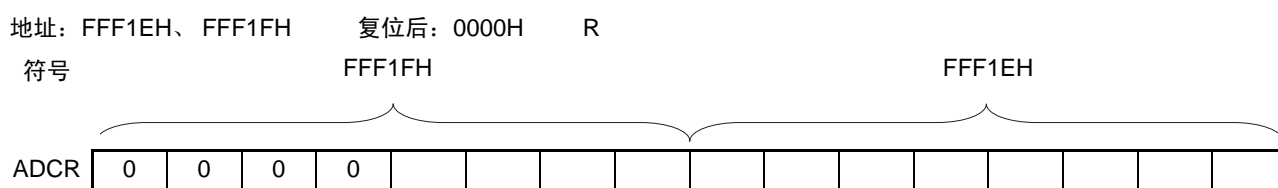
这是保存 A/D 转换结果的 16 位寄存器，高 4 位固定为“0”。每当 A/D 转换结束时，就将 ADSAR[11:0] 的值保存到 A/D 转换结果寄存器（但是，是否保存取决于 ADM2 寄存器的 ADRCK 位、ADUL 寄存器和 ADLL 寄存器的设定）。转换结果的高 4 位保存到 FFF1FH 的低 4 位，低 8 位保存到 FFF1EH 注。

通过 16 位存储器操作指令读 ADCR 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

注 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 9-8））的设定值范围内，就不保存 A/D 转换结果。

图 9-9 12 位 A/D 转换结果寄存器（ADCR）的格式



- 注意 1. 当写 A/D 转换器的模式寄存器 0（ADM0）、模拟输入通道指定寄存器（ADS）和 A/D 端口配置寄存器（ADPC）时，ADCR 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS、ADPC 寄存器前读转换结果。否则，就可能读不到正确的转换结果。
2. 当不发生 INTAD 时，A/D 转换结果不保存到 ADCR 寄存器。

### 9.3.6 8 位 A/D 转换结果寄存器 (ADCRH)

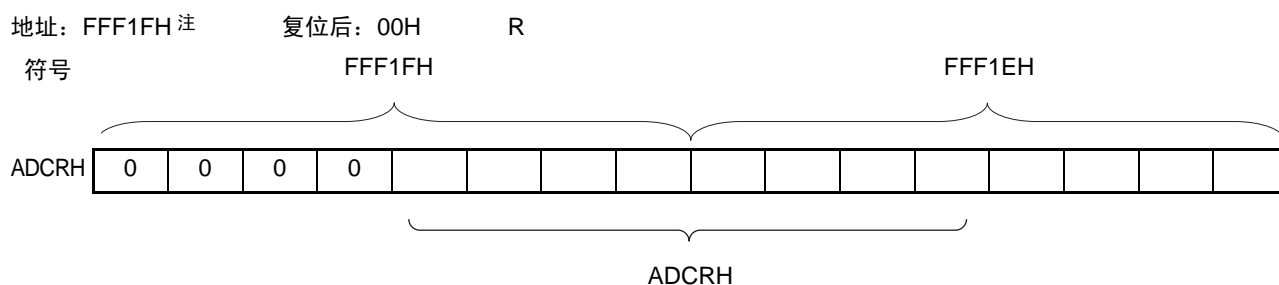
这是保存 ADCR[11:4] 所示的 8 位寄存器，保存 12 位分辨率的高 8 位注。

通过 8 位存储器操作指令读 ADCRH 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCR 位和 ADUL/ADLL 寄存器进行设定（参照图 9-8））的设定值范围内，就不保存 A/D 转换结果。

图 9-10 8 位 A/D 转换结果寄存器 (ADCRH) 的格式



注 如果读地址 FFF1FH，就读 ADCRH 的数据（FFF1FH 的低 4 位 + FFF1EH 的高 4 位）。

- 注意 1. 当写 A/D 转换器的模式寄存器 0 (ADM0)、模拟输入通道指定寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 时，ADCRH 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS、ADPC 寄存器前读转换结果。否则，就可能读不到正确的转换结果。
2. 当不发生 INTAD 时，A/D 转换结果不保存到 ADCRH 寄存器。

9.3.7 模拟输入通道指定寄存器（ADS）

这是指定要进行 A/D 转换的模拟电压输入通道的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 ADS 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 9-11 模拟输入通道指定寄存器（ADS）的格式 (1/2)

地址: FFF31H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	0	ADS2	ADS1	ADS0

○选择模式（ADMD=0）

ADISS	ADS4	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	0	ANI0	P20/ANI0/AV <sub>REFP</sub> 引脚
0	0	0	0	1	ANI1	P21/ANI1/AV <sub>REFM</sub> 引脚
0	0	0	1	0	ANI2	P22/ANI2 引脚
0	0	0	1	1	ANI3	P23/ANI3 引脚
0	0	1	0	0	ANI4	P24/ANI4 引脚注 1
0	0	1	0	1	ANI5	P25/ANI5 引脚注 1
0	0	1	1	0	ANI6	P26/ANI6 引脚注 1
0	0	1	1	1	ANI7	P27/ANI7 引脚
0	1	0	0	0	ANI16	P10/ANI16 引脚注 2
1	0	0	0	0	—	温度传感器的输出电压注 3
1	0	0	0	1	—	内部基准电压输出（1.45V）注 3
上述以外					禁止设定。	

注 1. ANI4 ~ ANI6 只限于 32 引脚产品。  
2. ANI16 只限于 24 引脚产品。  
3. 只有在 HS（高速主）模式中才能选择。详细内容请参照“图 22-3 用户选项字节（000C2H）的格式”。

图 9-11 模拟输入通道指定寄存器 (ADS) 的格式 (2/2)

地址: FFF31H      复位后: 00H      R/W

符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	0	ADS2	ADS1	ADS0

○扫描模式 (ADMD=1)

ADISS	ADS4	ADS2	ADS1	ADS0	模拟输入通道			
					扫描 0	扫描 1	扫描 2	扫描 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1 注	ANI2 注	ANI3 注	ANI4 注
0	0	0	1	0	ANI2 注	ANI3 注	ANI4 注	ANI5 注
0	0	0	1	1	ANI3 注	ANI4 注	ANI5 注	ANI6 注
0	0	1	0	0	ANI4 注	ANI5 注	ANI6 注	ANI7 注
上述以外					禁止设定。			

注      只限于 32 引脚产品。

注意 1. 必须将 bit3、5、6 置“0”。

- 对于由 ADPC 寄存器和 PMCx 寄存器设定为模拟输入的端口，必须通过端口模式寄存器 2 (PM2) 将其设定为输入模式。
- 对于由 A/D 端口配置寄存器 (ADPC) 设定为数字输入 / 输出的引脚，不能通过 ADS 寄存器进行设定。
- 要改写 ADISS 位时，必须在 A/D 电压比较器处于停止状态 (ADCS=0、ADCE=0) 下进行。
- 当将  $AV_{REFP}$  用作 A/D 转换器的正 (+) 基准电压源时，不能选择 ANI0 作为 A/D 转换通道。
- 当将  $AV_{REFM}$  用作 A/D 转换器的负 (-) 基准电压源时，不能选择 ANI1 作为 A/D 转换通道。
- 在将 ADISS 位置“1”的情况下，不能将内部基准电压 (1.45V) 用作正 (+) 基准电压源。另外，在将 ADISS 位置“1”后，不能使用第1次转换结果。详细设定流程请参照“9.7.4 选择温度传感器的输出电压/内部基准电压时的设定”。
- 要转移到 STOP 模式时，不能将 ADISS 位置“1”。当 ADISS 位为“1”时，如果在 CPU 以主系统时钟运行中转移到 HALT 模式，就加上“27.3.2 电源电流特性”所示的温度传感器工作电流 ( $I_{TMPS}$ ) 或者 A/D 转换器基准电压电流 ( $I_{ADREF}$ ) 的电流值。
- 当使用的产品不存在对应的 ANI 引脚时，必须忽视转换结果。



### 9.3.8 转换结果比较上限值设定寄存器（ADUL）

这是用于检查 A/D 转换结果上限值的设定寄存器。

将 A/D 转换结果和 ADUL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设定范围内（参照图 9-8）控制中断信号（INTAD）的产生。

通过 8 位存储器操作指令设定 ADUL 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

- 注意 1. 当选择 12 位分辨率 A/D 转换时，将 12 位 A/D 转换结果寄存器（ADCR）的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。
2. 要改写 ADUL 寄存器和 ADLL 寄存器时，必须在处于转换停止状态（ADCS=0、ADCE =0）时进行。
3. 在设定 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL > ADLL。

图 9-12 转换结果比较上限值设定寄存器（ADUL）的格式

地址: F0011H	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

### 9.3.9 转换结果比较下限值设定寄存器（ADLL）

这是用于检查 A/D 转换结果下限值的设定寄存器。

将 A/D 转换结果和 ADLL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2（ADM2）的 ADRCK 位的设定范围内（参照图 9-8）控制中断信号（INTAD）的产生。

通过 8 位存储器操作指令设定 ADLL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 9-13 转换结果比较下限值设定寄存器（ADLL）的格式

地址: F0012H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

- 注意 1. 当选择 12 位分辨率 A/D 转换时，将 12 位 A/D 转换结果寄存器（ADCR）的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。
2. 要改写 ADUL 寄存器和 ADLL 寄存器时，必须在处于转换停止状态（ADCS=0、ADCE =0）时进行。
3. 在设定 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL > ADLL。

### 9.3.10 A/D 测试寄存器 (ADTES)

此寄存器选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANLxx)、温度传感器的输出电压和内部基准电压 (1.45V) 作为 A/D 转换对象。

通过 8 位存储器操作指令设定 ADTES 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 9-14 A/D 测试寄存器 (ADTES) 的格式

地址: F0013H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D 转换对象
0	0	ANLxx/ 温度传感器的输出电压注 / 内部基准电压输出 (1.45V) 注 (由模拟输入通道指定寄存器 (ADS) 进行设定) 注
1	0	负 (-) 基准电压 (通过 ADM2 寄存器的 ADREFM 位进行设定)
1	1	正 (+) 基准电压 (通过 ADM2 寄存器的 ADREFP1 位和 ADREFP0 位进行设定)
上述以外		禁止设定。

注 只有在 HS (高速主) 模式中才能选择温度传感器输出和内部基准电压输出 (1.45V)。详细内容请参照“图 22-3 用户选项字节 (000C2H) 的格式”。

注意 有关 A/D 测试功能的详细内容, 请参照“第 20 章 安全功能”。

### 9.3.11 控制模拟输入引脚端口功能的寄存器

必须设定与 A/D 转换器模拟输入复用的端口功能的控制寄存器 (端口模式寄存器 (PMxx)、端口模式控制寄存器 1 (PMC1) 和 A/D 端口配置寄存器 (ADPC))。详细内容请参照“4.3.1 端口模式寄存器 (PMxx)”、“4.3.4 端口模式控制寄存器 1 (PMC1) (只限于 24 引脚产品)”和“4.3.5 A/D 端口配置寄存器 (ADPC)”。

在将 ANI0 ~ ANI7 引脚用作 A/D 转换器的模拟输入时, 必须将各端口对应的端口模式寄存器 (PMxx) 的位置“1”, 并且通过 A/D 端口配置寄存器 (ADPC) 设定为模拟输入。

在将 ANI16 引脚用作 A/D 转换器的模拟输入时, 必须将各端口对应的端口模式寄存器 (PMxx) 和端口模式控制寄存器 1 (PMC1) 的位置“1”。

## 9.4 A/D 转换器的转换运行

A/D 转换器的转换运行如下所示。

- ① 通过采样 & 保持电路对所选模拟输入通道的输入电压进行采样。
- ② 在进行规定时间的采样后，采样 & 保持电路为保持状态，并且将采样电压保持到 A/D 转换结束为止。
- ③ 将逐次逼近寄存器（SAR）的 bit11 置位，并且通过分接头选择器将串联电阻串的分接头电压设定为  $(1/2) AV_{REF}$ 。
- ④ 通过 A/D 电压比较器将串联电阻串的分接头电压和采样电压的电压差进行比较。如果模拟输入大于  $(1/2) AV_{REF}$ ，SAR 寄存器的 MSB 位就保持被置位的状态。如果模拟输入小于  $(1/2) AV_{REF}$ ，就对 MSB 位进行复位。
- ⑤ 然后，自动将 SAR 寄存器的 bit10 置位并且进行下一次比较。在此，根据 bit11 被设定的结果值，选择比较电压生成电路的分接头电压。
  - bit11=1:  $(3/4) AV_{REF}$
  - bit11=0:  $(1/4) AV_{REF}$
 将此分接头电压和采样电压进行比较，根据比较结果对 SAR 寄存器的 bit10 进行以下操作：
  - 采样电压  $\geq$  分接头电压: bit10=1
  - 采样电压  $<$  分接头电压: bit10=0
- ⑥ 将此比较继续进行到 SAR 寄存器的 bit0 为止。
- ⑦ 当 12 位比较结束时，SAR 寄存器保存了有效的数字结果，并且将其值传送到 A/D 转换结果寄存器（ADCR、ADCRH），而且被锁存注<sup>1</sup>。  
同时，能产生 A/D 转换结束中断请求（INTAD）注<sup>1</sup>。
- ⑧ 重复①～⑦的运行，直到 ADCS 位变为“0”为止注<sup>2</sup>。  
要停止 A/D 转换器时，必须将 ADCS 位置“0”。

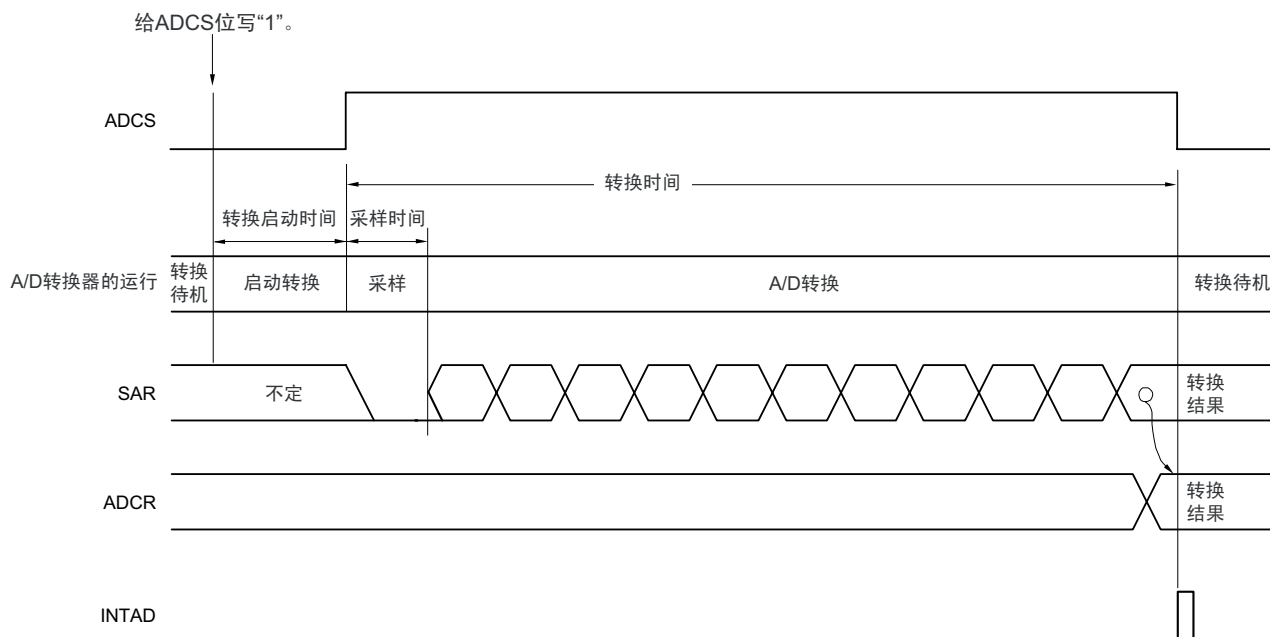
- 注 1. 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 9-8））的设定值范围内，就不产生 A/D 转换结束中断请求信号（INTAD），并且 ADCR 寄存器和 ADCRH 寄存器不保存 A/D 转换结果。
2. 在连续转换模式中，ADCS 标志不自动清“0”。即使在硬件触发无等待模式的单次转换模式中，ADCS 标志也不自动清“0”而保持“1”的状态。

备注 1. 有 2 种 A/D 转换结果寄存器。

- ADCR 寄存器（16 位）：保存 12 位的 A/D 转换值。
- ADCRH 寄存器（8 位）：保存 8 位的 A/D 转换值。

2.  $AV_{REF}$  是 A/D 转换器的正（+）基准电压，可选择  $AV_{REFP}$ 、内部基准电压（1.45V）或者  $V_{DD}$ 。

图 9-15 A/D 转换器的转换运行（软件触发模式的情况）



如果在单次转换模式中进行 A/D 转换，ADCS 位就在 A/D 转换结束后自动清“0”。

在连续转换模式中，A/D 转换连续进行到通过软件将 A/D 转换器的模式寄存器 0 (ADM0) 的 bit7 (ADCS) 清“0”为止。

在 A/D 转换过程中，如果改写或者重写模拟输入通道指定寄存器 (ADS)，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。

在产生复位信号后，A/D 转换结果寄存器 (ADCR、ADCRH) 的值变为“0000H”和“00H”。

## 9.5 输入电压和转换结果

模拟输入引脚（ANI0 ~ ANI7、ANI16）的模拟输入电压和理论上的 A/D 转换结果（12 位 A/D 转换结果寄存器（ADCR））有以下表达式的关系。

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

或者

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

INT() : 将括号中的数值的整数部分返回的函数

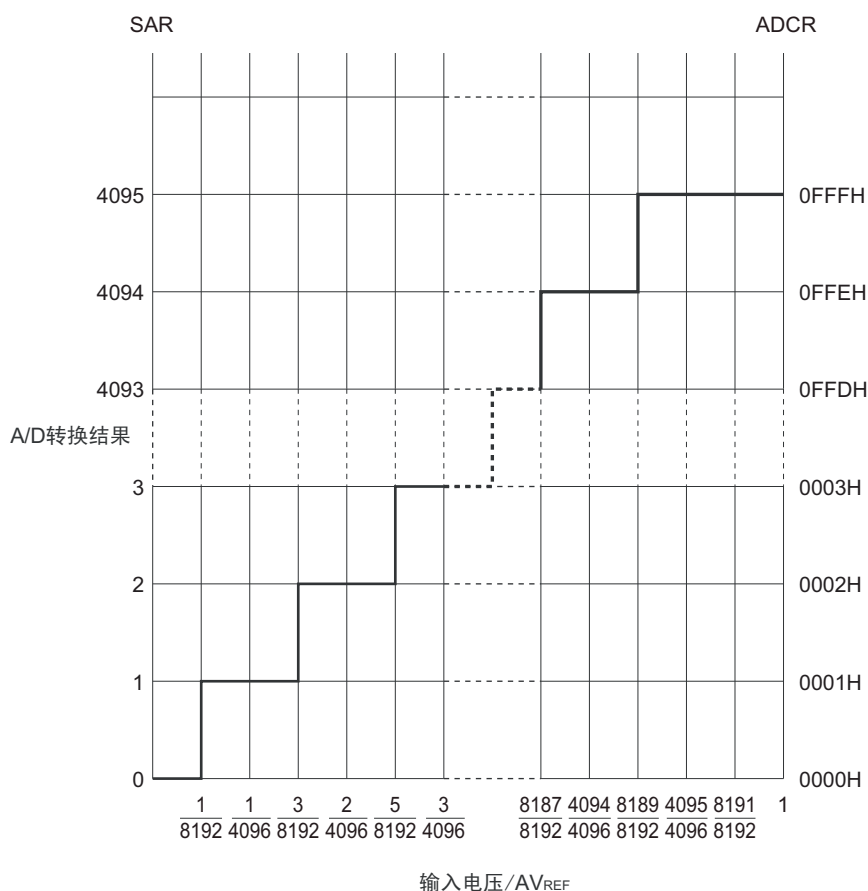
$V_{\text{AIN}}$  : 模拟输入电压

$V_{\text{REF}}$  :  $V_{\text{REF}}$  引脚电压

ADCR : A/D 转换结果寄存器（ADCR）的值

模拟输入电压和 A/D 转换结果的关系如图 9-16 所示。

图 9-16 模拟输入电压和 A/D 转换结果的关系



备注  $V_{\text{REF}}$  是 A/D 转换器的正 (+) 基准电压，可选择  $V_{\text{REFP}}$ 、内部基准电压 (1.45V) 或者  $V_{\text{DD}}$ 。

9.6 A/D 转换器的运行模式

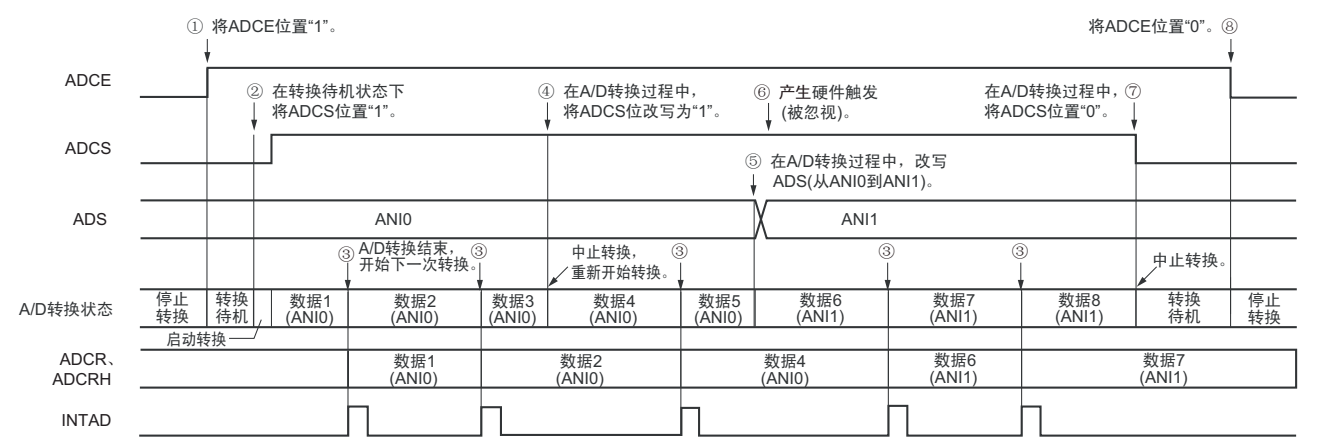
A/D 转换器的各模式的运行如下所示。有关各模式的设定步骤，请参照“9.7 A/D 转换器的设定流程图”。

9.6.1 软件触发模式（选择模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间注进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换。
- ④ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑥ 即使在转换过程中输入硬件触发也不开始 A/D 转换。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间 = 0.5μs。  
当选择标准通道作为模拟输入通道时，稳定等待时间 = 2μs。

图 9-17 软件触发模式（选择模式、连续转换模式）的运行时序例子

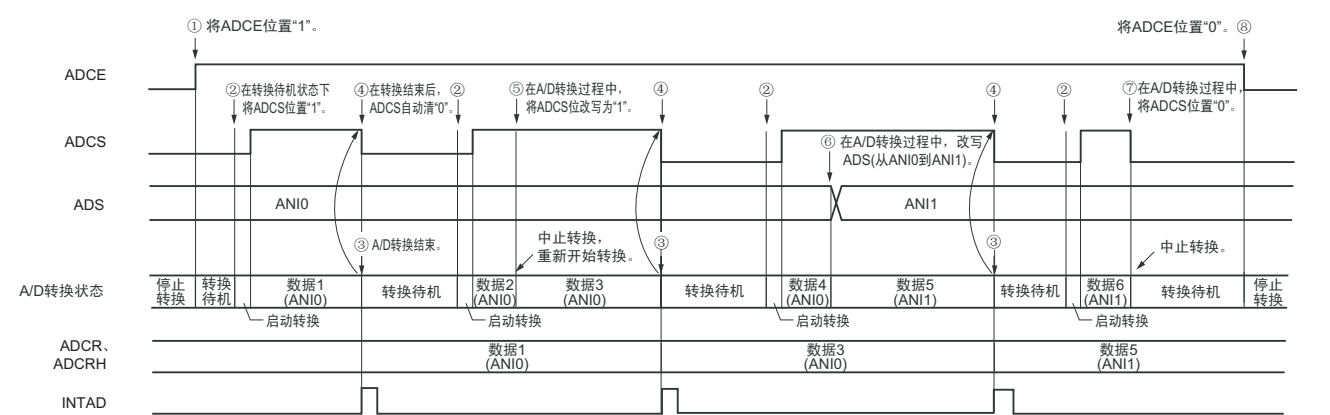


9.6.2 软件触发模式（选择模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ④ 在A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。
- ⑤ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而对由ADS寄存器重新指定的模拟输入进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间 =0.5μs。  
当选择标准通道作为模拟输入通道时，稳定等待时间 =2μs。

图 9-18 软件触发模式（选择模式、单次转换模式）的运行时序例子

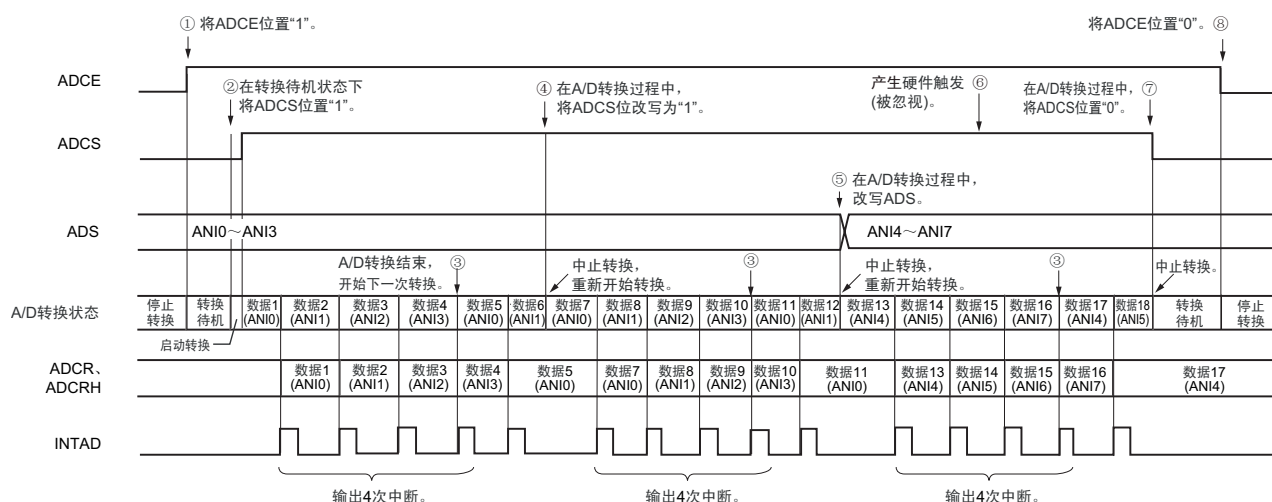


## 9.6.3 软件触发模式（扫描模式、连续转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换（4个通道）。
- ④ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑥ 即使在转换过程中输入硬件触发也不开始A/D转换。
- ⑦ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间=0.5μs。  
当选择标准通道作为模拟输入通道时，稳定等待时间=2μs。

图 9-19 软件触发模式（扫描模式、连续转换模式）的运行时序例子



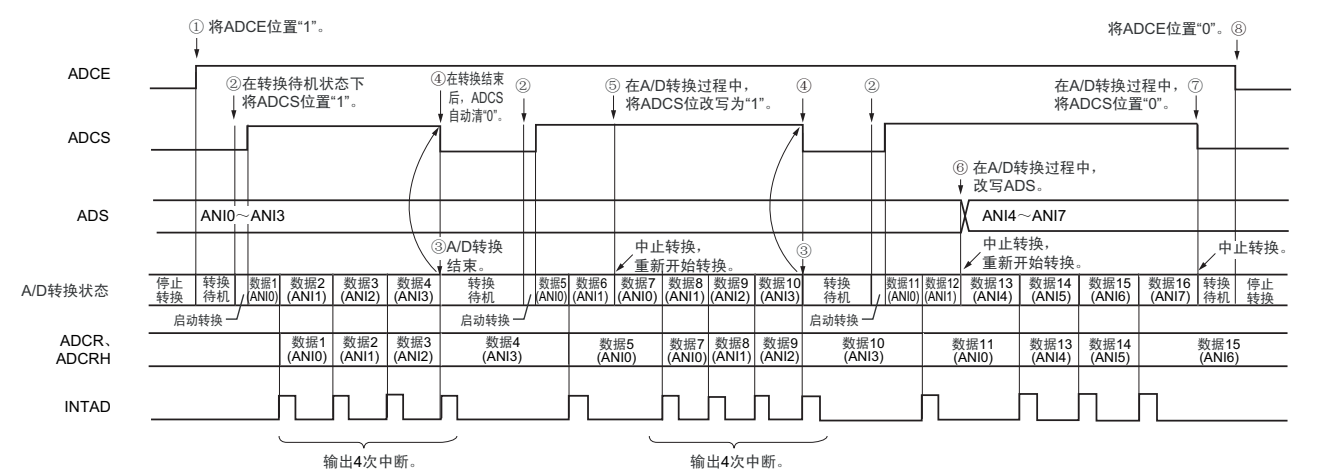


9.6.4 软件触发模式（扫描模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间注进行计数后，将ADM0寄存器的ADCS位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ④ 在4个通道的A/D转换结束后，ADCS位自动清“0”，进入A/D转换待机状态。
- ⑤ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。
- ⑧ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。即使在A/D转换待机的状态下输入硬件触发也不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间 =0.5μs。  
当选择标准通道作为模拟输入通道时，稳定等待时间 =2μs。

图 9-20 软件触发模式（扫描模式、单次转换模式）的运行时序例子



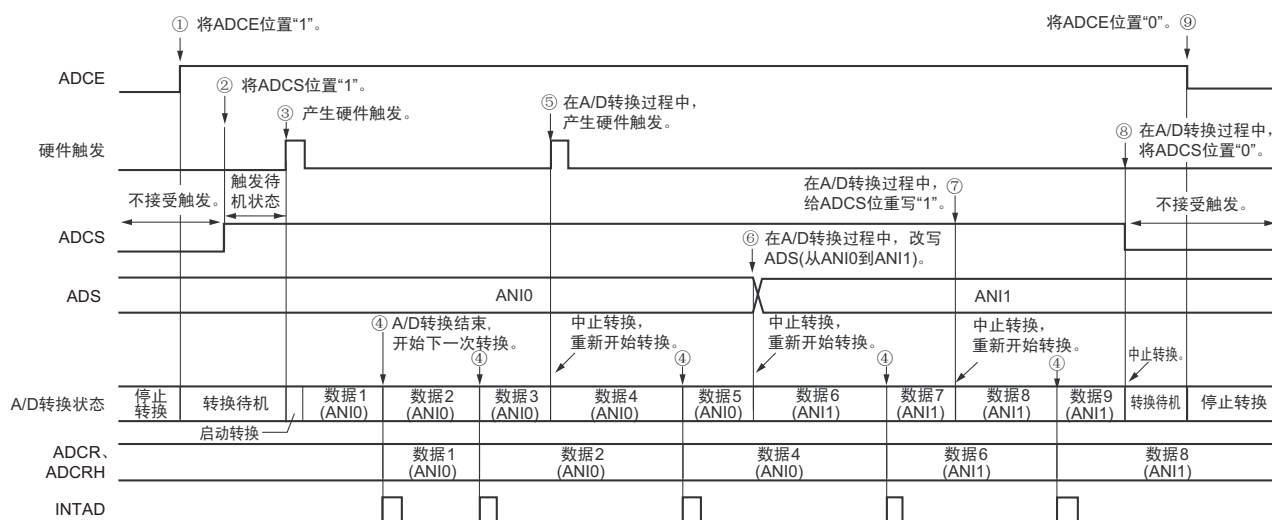
### 9.6.5 硬件触发无等待模式（选择模式、连续转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADMO）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADMO寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。
- ④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在A/D转换结束后立即开始下一次A/D转换。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而对由ADS寄存器重新指定的模拟输入进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时, 稳定等待时间 = 0.5μs。

当选择标准通道作为模拟输入通道时，稳定等待时间=2 $\mu$ s。

图 9-21 硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



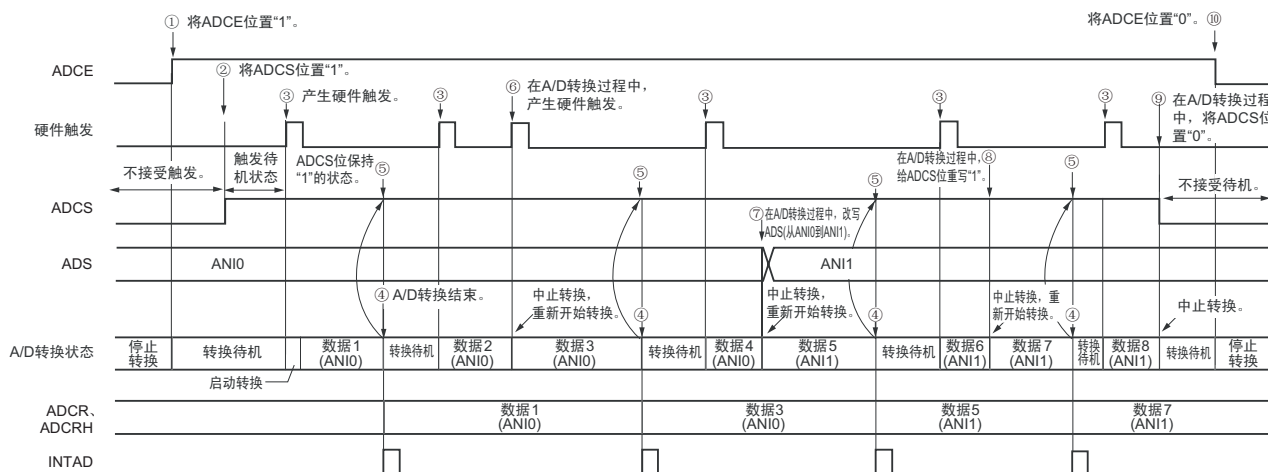
## 9.6.6 硬件触发无等待模式（选择模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。
- ④ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ⑤ 在A/D转换结束后，ADCS位保持“1”的状态，进入A/D转换待机状态。
- ⑥ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而对由ADS寄存器重新指定的模拟输入进行A/D转换。取消转换途中的数据。
- ⑧ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑨ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑩ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间=0.5 $\mu$ s。

当选择标准通道作为模拟输入通道时，稳定等待时间=2 $\mu$ s。

图 9-22 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



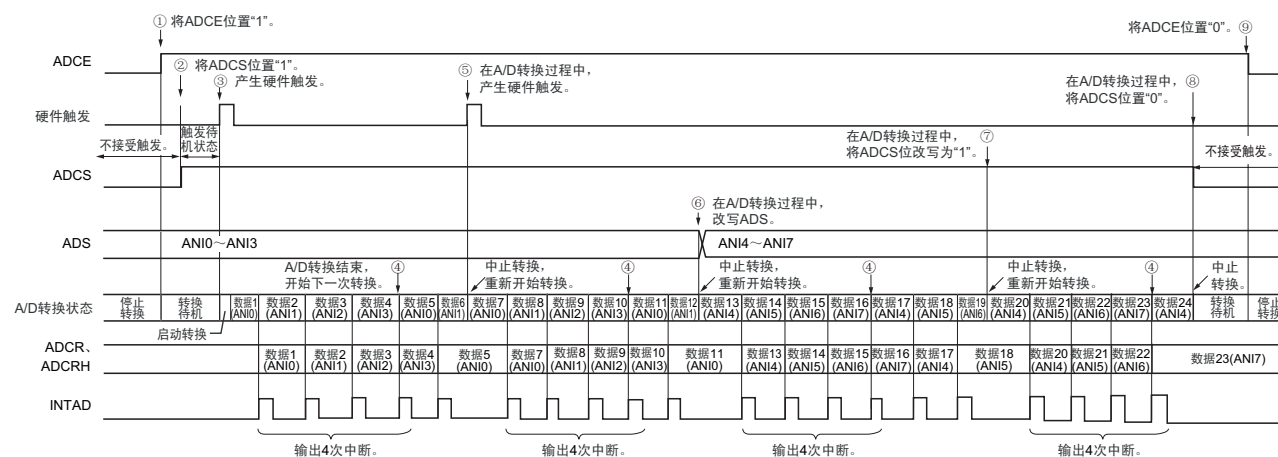
## 9.6.7 硬件触发无等待模式（扫描模式、连续转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ④ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑨ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCE位为“0”时，即使将ADCS位置“1”也被忽视，不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间=0.5μs。

当选择标准通道作为模拟输入通道时，稳定等待时间=2μs。

图 9-23 硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



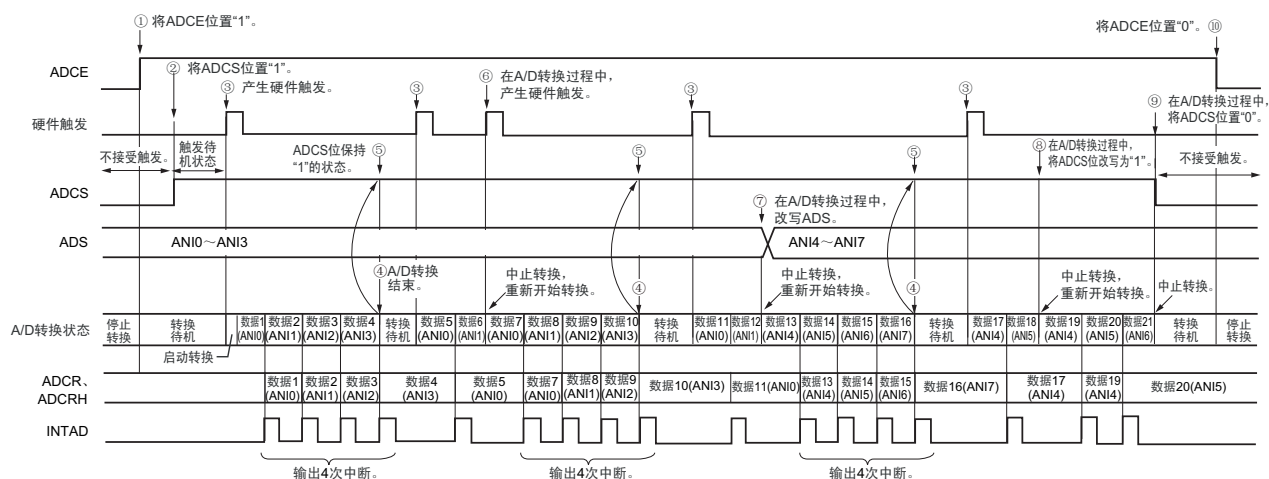
## 9.6.8 硬件触发无等待模式（扫描模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入A/D转换待机状态。
- ② 在通过软件对稳定等待时间<sup>注</sup>进行计数后，将ADM0寄存器的ADCS位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将ADCS位置“1”也不开始A/D转换。
- ③ 如果在ADCS位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。依次从扫描0指定的模拟输入通道进行A/D转换。
- ④ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ⑤ 在4个通道的A/D转换结束后，ADCS位保持“1”的状态，进入A/D转换待机状态。
- ⑥ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑧ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑨ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入A/D转换待机状态。但是，在此状态下A/D转换器不进入停止状态。
- ⑩ 如果在A/D转换待机状态下将ADCE位置“0”，A/D转换器就进入停止状态。当ADCS位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

注 当选择高精度通道作为模拟输入通道时，稳定等待时间=0.5μs。

当选择标准通道作为模拟输入通道时，稳定等待时间=2μs。

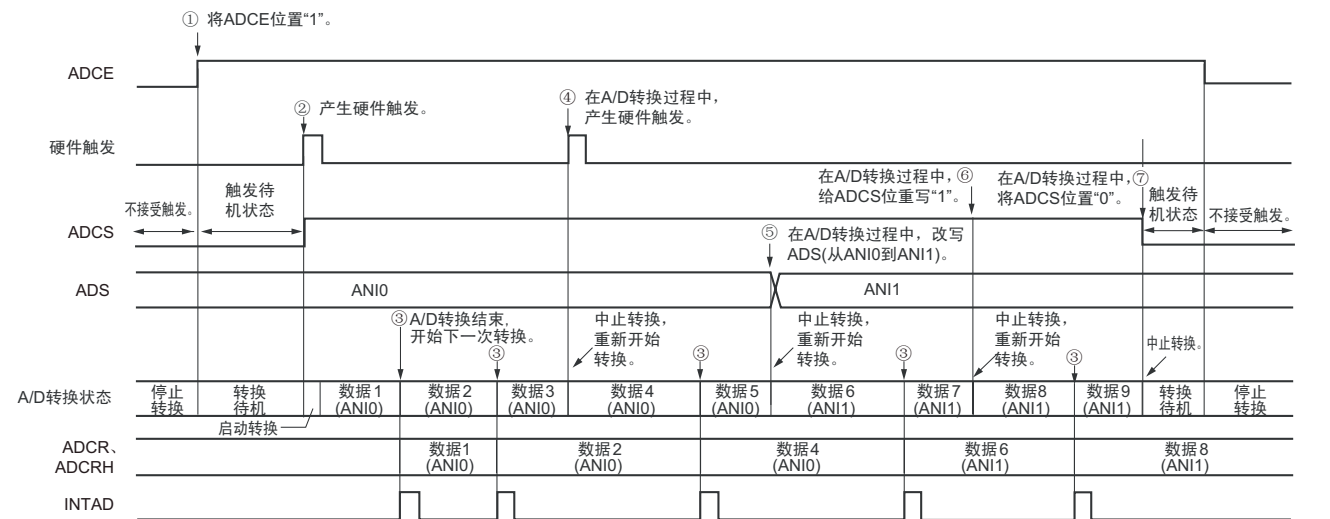
图 9-24 硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



9.6.9 硬件触发等待模式（选择模式、连续转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在A/D转换结束后立即开始下一次A/D转换（此时，不需要硬件触发）。
- ④ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而对由ADS寄存器重新指定的模拟输入进行A/D转换。取消转换途中的数据。
- ⑥ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 9-25 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子

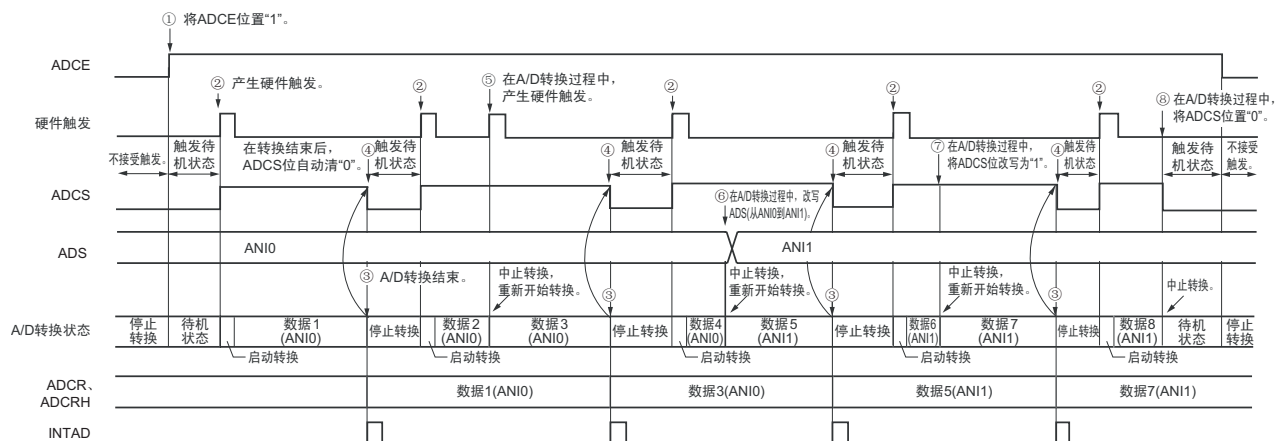




## 9.6.10 硬件触发等待模式（选择模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。
- ③ 如果A/D转换结束，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而对由ADS寄存器重新指定的模拟输入进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

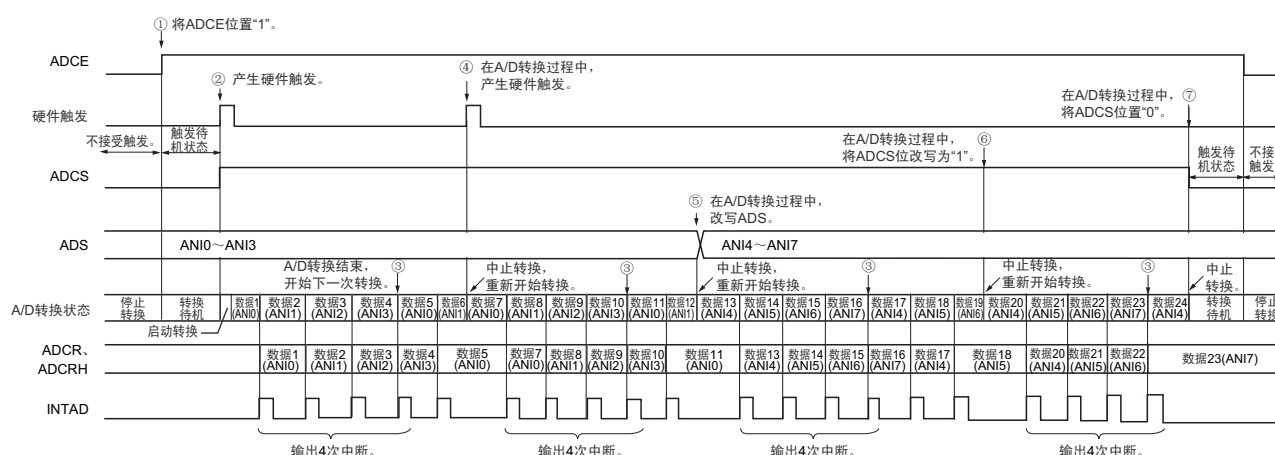
图 9-26 硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



## 9.6.11 硬件触发等待模式（扫描模式、连续转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。在4个通道的A/D转换结束后立即从所设通道自动开始下一次A/D转换。
- ④ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑥ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 9-27 硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子

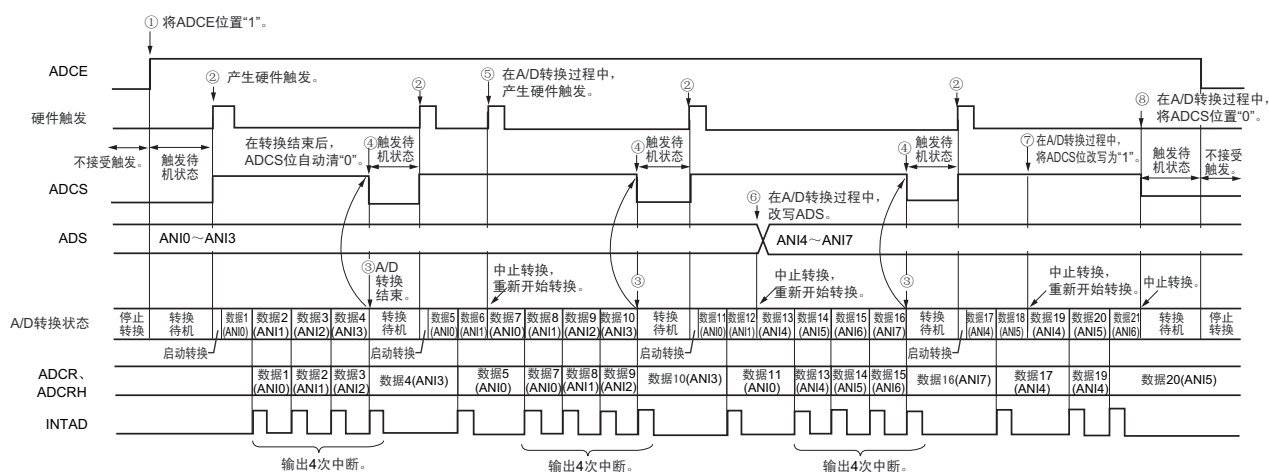




## 9.6.12 硬件触发等待模式（扫描模式、单次转换模式）

- ① 在停止状态下，将A/D转换器的模式寄存器0（ADM0）的ADCE位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描0～扫描3的4个模拟输入通道进行A/D转换。在输入硬件触发的同时自动将ADM0寄存器的ADCS位置“1”。依次从扫描0指定的模拟输入通道进行A/D转换。
- ③ 连续进行4个模拟输入通道的A/D转换。每当A/D转换结束时，就将转换结果保存到A/D转换结果寄存器（ADCR、ADCRH），并且产生A/D转换结束中断请求信号（INTAD）。
- ④ 在A/D转换结束后，ADCS位自动清“0”，A/D转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写ADS寄存器，就中止当前的A/D转换而从由ADS寄存器重新指定的最初通道进行A/D转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给ADCS位重写“1”，就中止当前的A/D转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将ADCS位置“0”，就中止当前的A/D转换而进入硬件触发待机状态，并且A/D转换器进入停止状态。当ADCE位为“0”时，即使输入硬件触发也被忽视，不开始A/D转换。

图 9-28 硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子

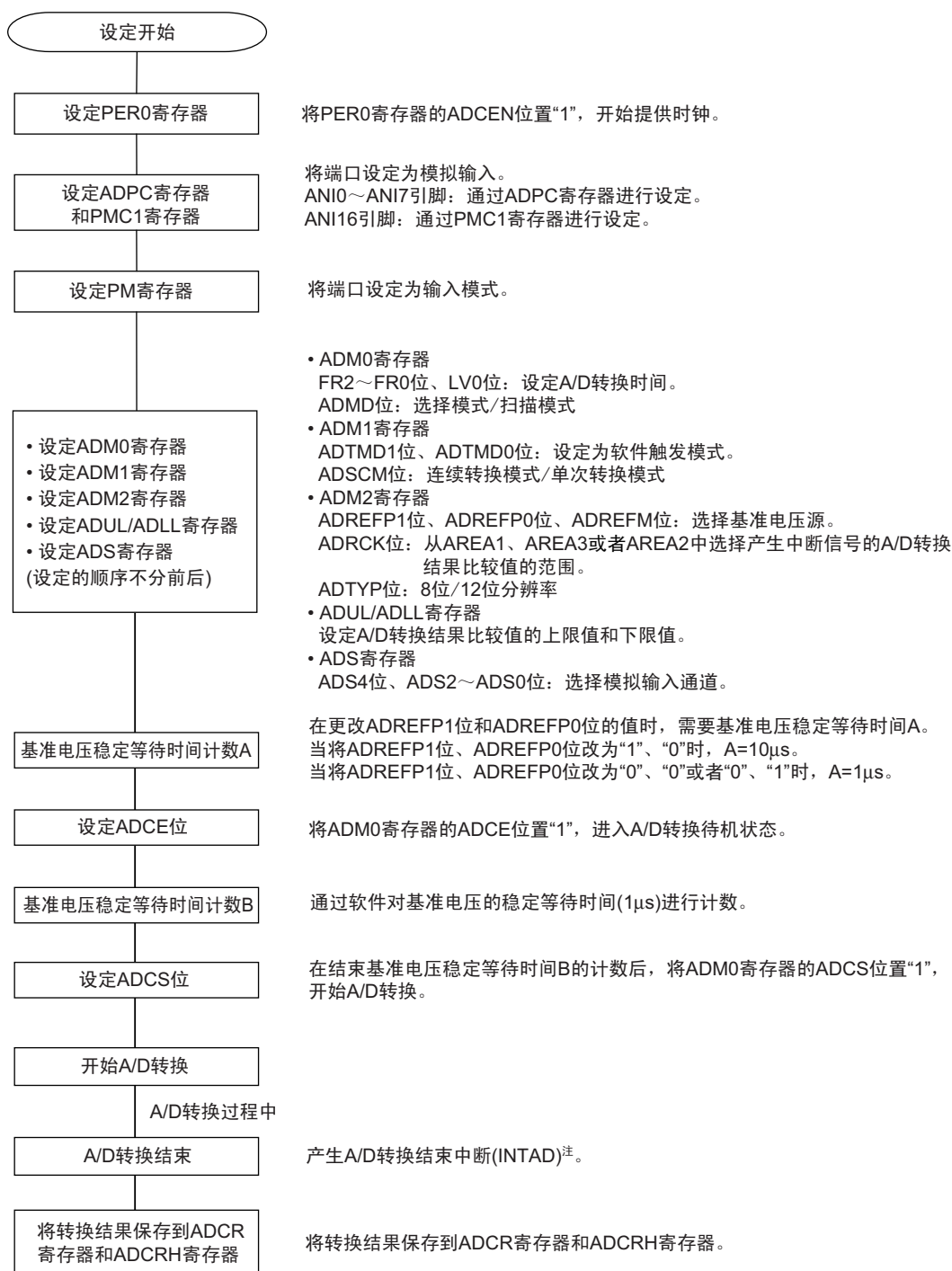


## 9.7 A/D 转换器的设定流程图

各运行模式的 A/D 转换器的设定流程图如下所示。

### 9.7.1 软件触发模式的设定

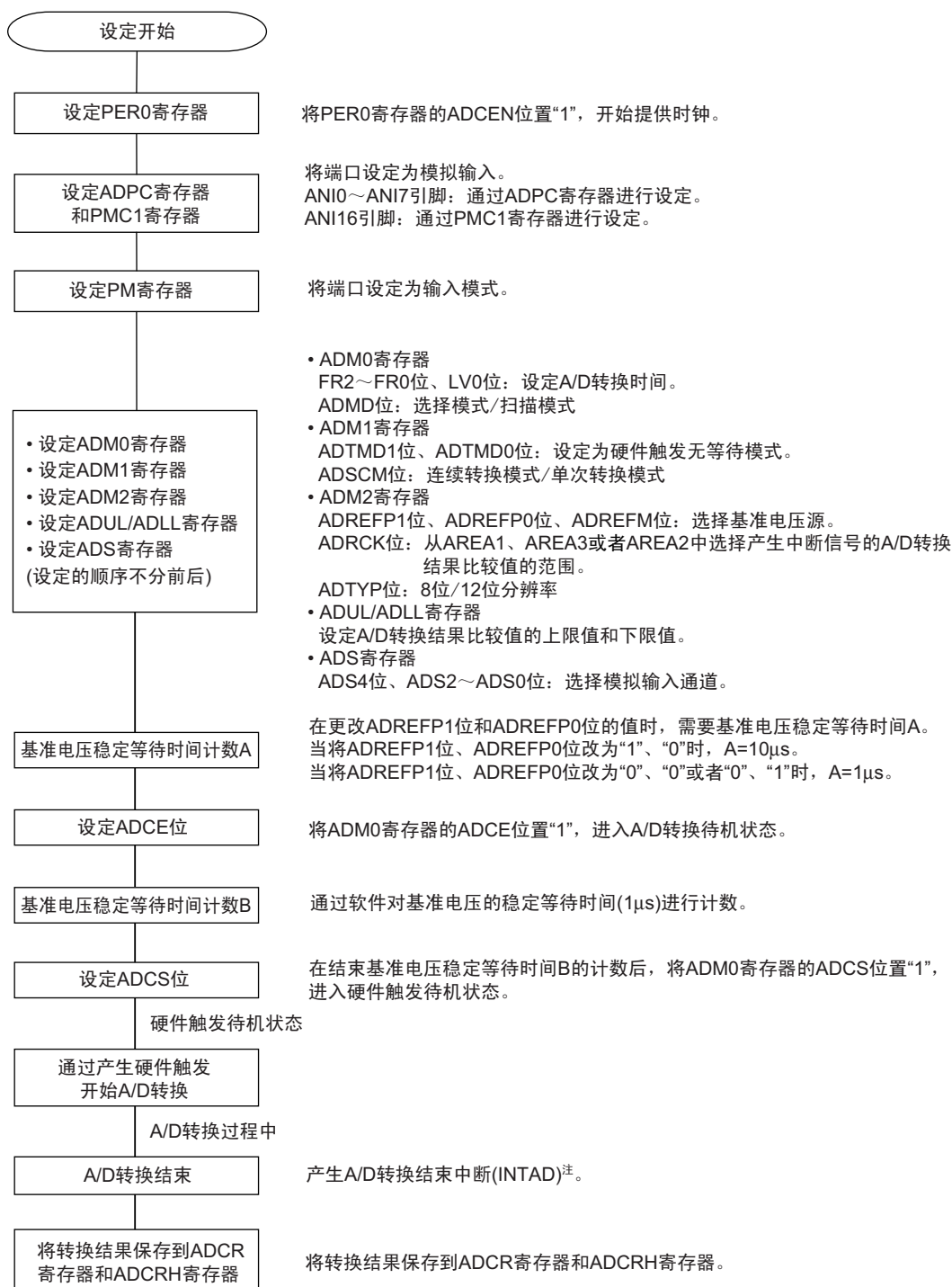
图 9-29 软件触发模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

## 9.7.2 硬件触发无等待模式的设定

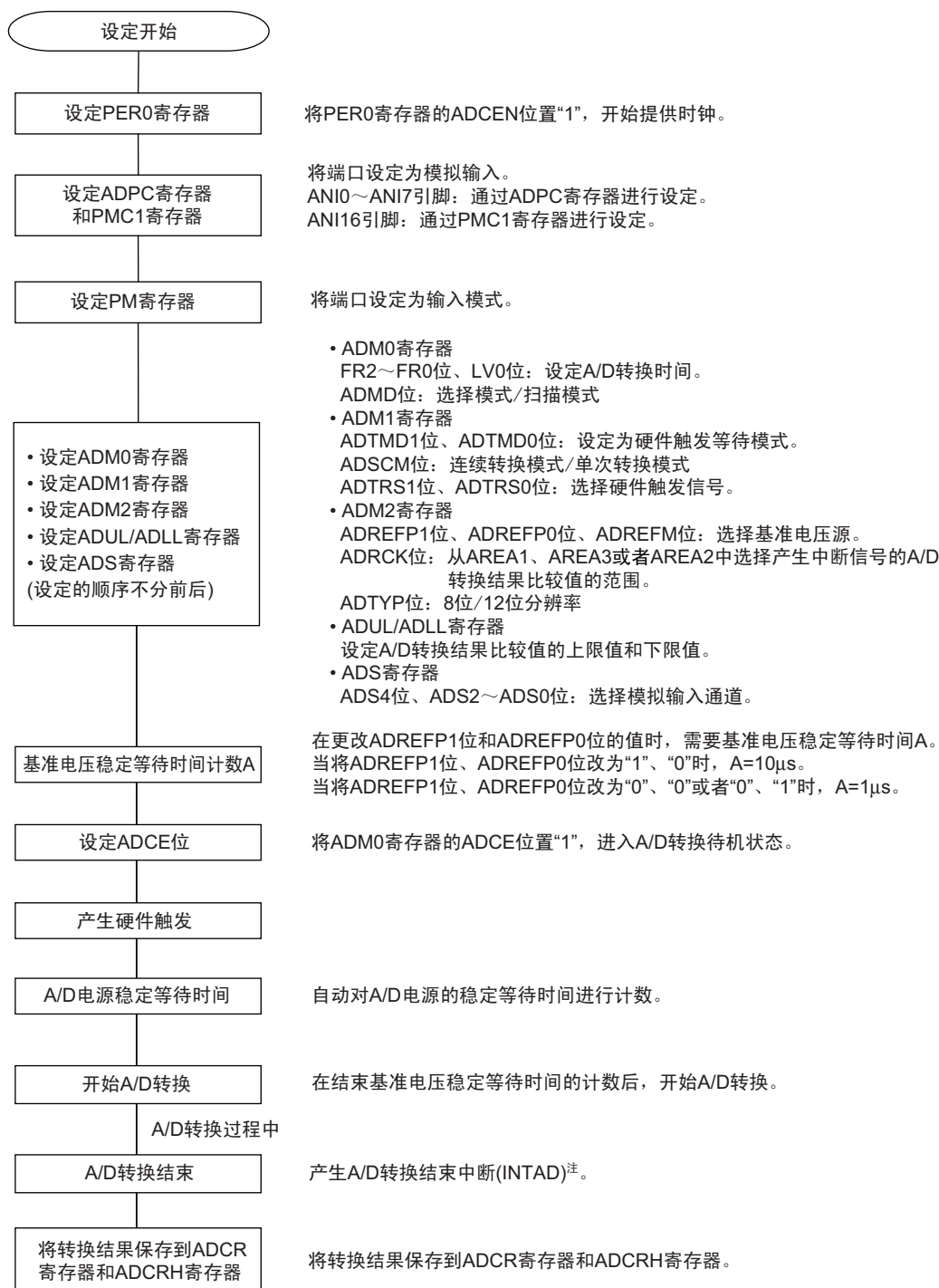
图 9-30 硬件触发无等待模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

## 9.7.3 硬件触发等待模式的设定

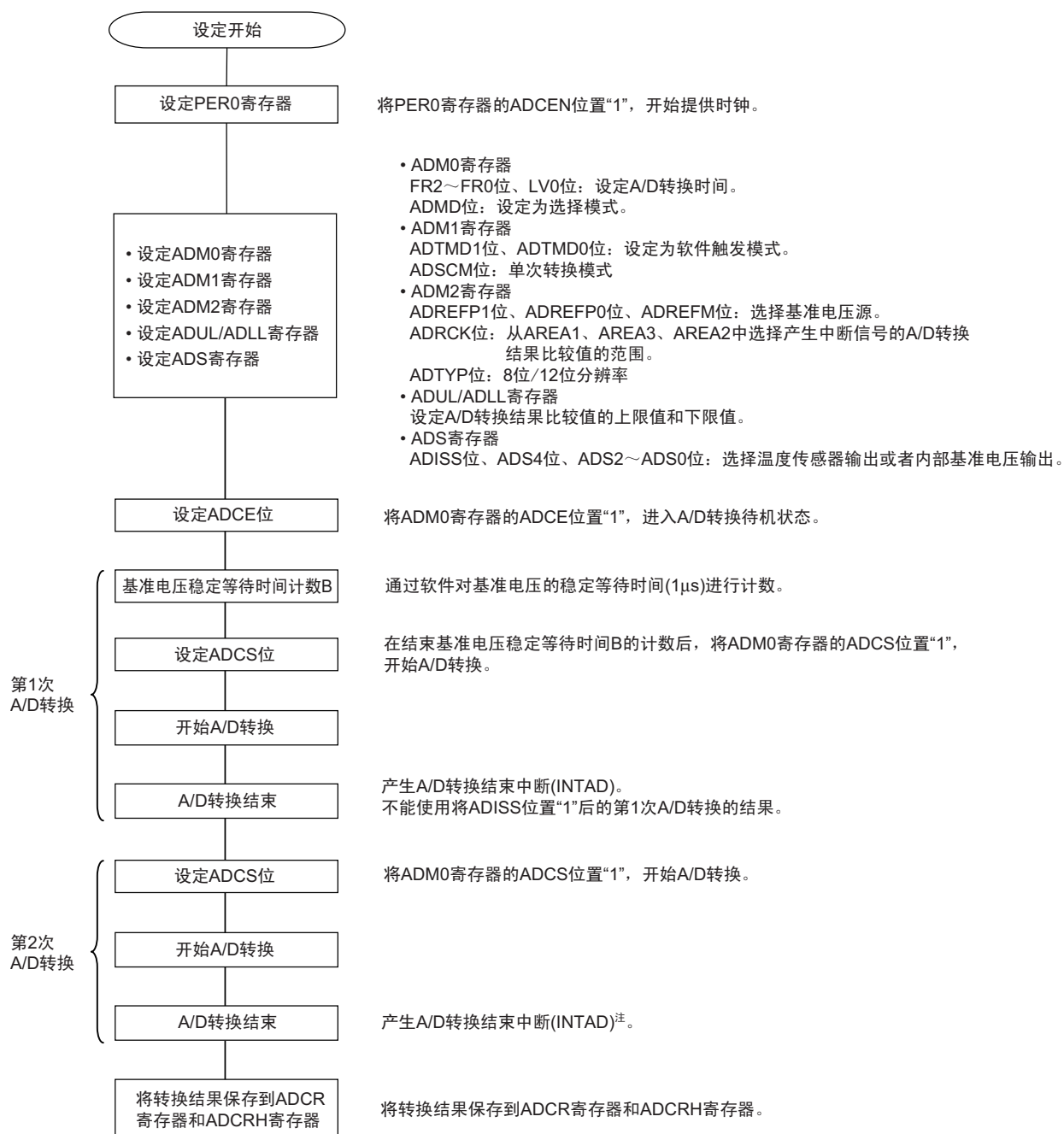
图 9-31 硬件触发等待模式的设定



注 根据 ADRCCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

### 9.7.4 选择温度传感器的输出电压 / 内部基准电压时的设定 (以软件触发模式、单次转换模式为例)

图 9-32 选择温度传感器的输出电压 / 内部基准电压时的设定

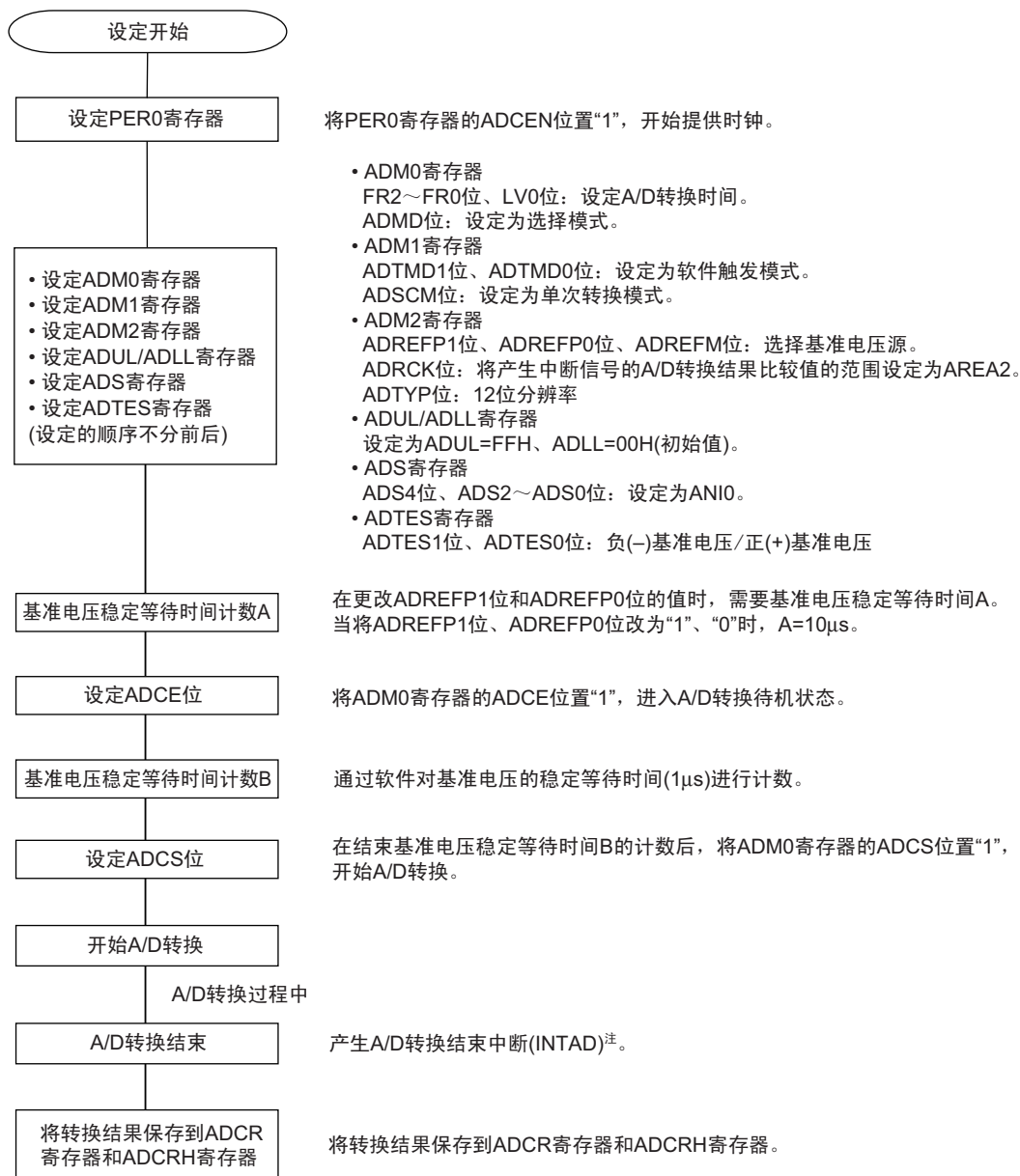


**注** 根据 ADRCCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

**注意** 只有在 HS（高速主）模式中才能选择。详细内容请参照“图 22-3 用户选项字节（000C2H）的格式”。

## 9.7.5 测试模式的设定

图 9-33 测试模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

## 9.8 SNOOZE 模式功能

这是在 STOP 模式中通过输入硬件触发进行 A/D 转换的模式。在通常的 STOP 模式中停止 A/D 转换，但是如果使用此模式，就能在 CPU 不运行的状态下通过输入硬件触发进行 A/D 转换，减小工作电流。

在 SNOOZE 模式中，如果通过 ADUL 寄存器和 ADLL 寄存器指定转换结果的范围，就能每隔一段时间判断 A/D 转换结果。从而能判断电源电压监视或者通过 A/D 输入进行的键输入。

在 SNOOZE 模式中，只能使用以下 2 种转换模式：

- 硬件触发等待模式（选择模式和单次转换模式）
- 硬件触发等待模式（扫描模式和单次转换模式）

**注意** 只有在选择高速内部振荡器时钟作为  $f_{CLK}$  时才能设定 SNOOZE 模式。

图 9-34 使用 SNOOZE 模式功能时的框图



当使用 SNOOZE 模式功能时，在转移到 STOP 模式前进行各寄存器的初始设定（参照“9.7.3 硬件触发等待模式的设定”注2）。在即将要转移到 STOP 模式前，将 A/D 转换器的模式寄存器 2（ADM2）的 bit2（AWC）置“1”。在初始设定结束后，将 A/D 转换器的模式寄存器 0（ADM0）的 bit0（ADCE）置“1”。

如果在转移到 STOP 模式后输入硬件触发，就给 A/D 转换器提供高速内部振荡器时钟。在提供高速内部振荡器时钟后，自动对 A/D 电源稳定等待时间进行计数，开始 A/D 转换。

A/D 转换结束后的 SNOOZE 模式的运行因是否产生中断信号而不同注1。

- 注**
1. 根据 A/D 转换结果比较功能的设定（ADRCK 位、ADUL/ADLL 寄存器），有可能不产生中断请求信号。
  2. 必须将 ADM1 寄存器置“E1H”。

**备注** 硬件触发为 ELC 选择的事件信号（INTP0 ~ INTP5）。  
必须通过 A/D 转换器的模式寄存器 1（ADM1）设定硬件触发。

## (1) 在 A/D 转换结束后发生中断的情况

如果 A/D 转换结果的值在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定）的设定值范围内，就产生 A/D 转换结束中断请求信号（INTAD）。

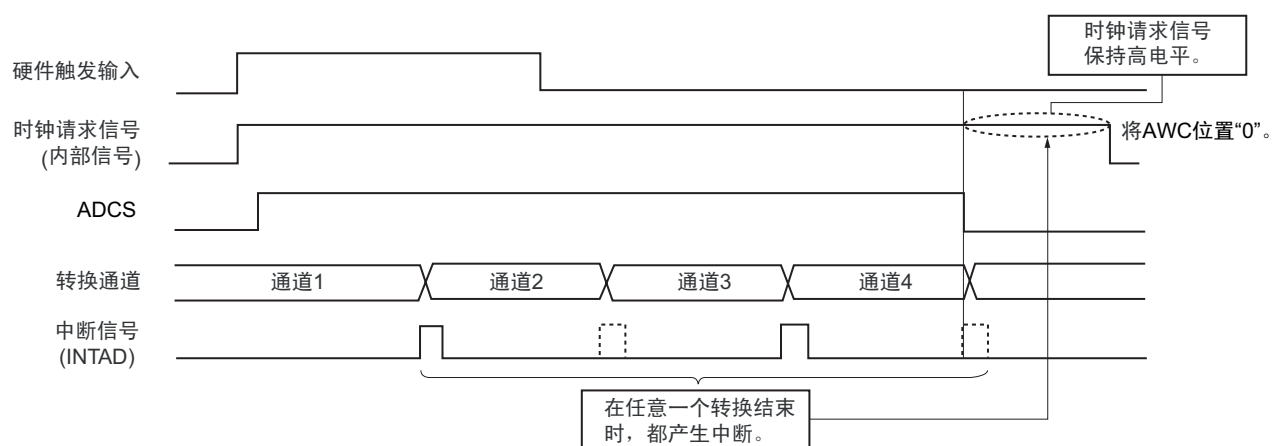
## ● 选择模式

如果在 A/D 转换结束后产生 A/D 转换结束中断请求信号（INTAD），A/D 转换器就从 SNOOZE 模式转移到通常运行模式。在此，必须清除 A/D 转换器的模式寄存器 2（ADM2）的 bit2（AWC=0：解除 SNOOZE 模式）。如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。

## ● 扫描模式

在 4 个通道的 A/D 转换过程中只要产生 1 次 A/D 转换结束中断请求信号（INTAD），A/D 转换器就从 SNOOZE 模式转移到通常运行模式。在此，必须清除 A/D 转换器模式寄存器 2（ADM2）的 bit2（AWC = 0：解除 SNOOZE 模式）。如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。

图 9-35 在 A/D 转换结束后发生中断时的运行例子（扫描模式）





## (2) 在 A/D 转换结束后不发生中断的情况

如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定）的设定值范围内，就不产生 A/D 转换结束中断请求信号（INTAD）。

## ● 选择模式

如果不产生 A/D 转换结束中断请求信号（INTAD），时钟请求信号（内部信号）就在 A/D 转换结束后自动变为低电平，并且停止提供高速内部振荡器时钟。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。

## ● 扫描模式

如果在 4 个通道的 A/D 转换过程中 1 次也不产生 A/D 转换结束中断请求信号（INTAD），时钟请求信号（内部信号）就在 4 个通道的 A/D 转换结束后自动变为低电平，并且停止提供高速内部振荡器时钟。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。

图 9-36 在 A/D 转换结束后不发生中断时的运行例子（扫描模式）

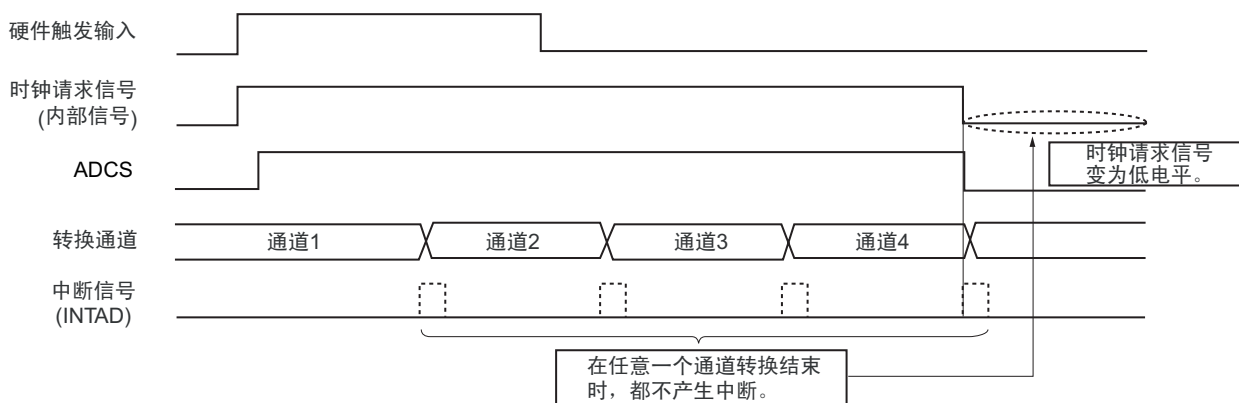
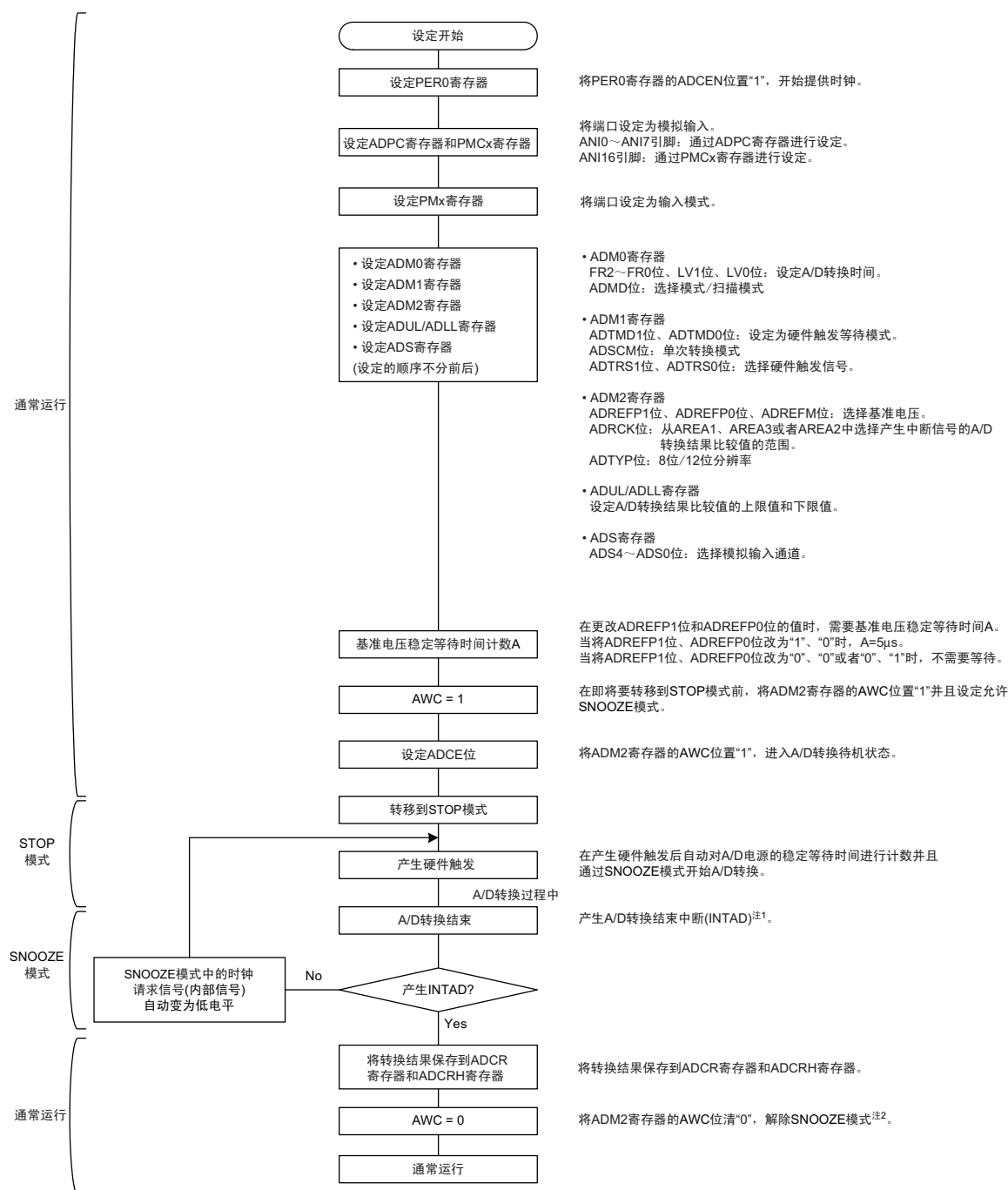


图 9-37 A/D 转换结束后不产生中断时的运行例子（扫描模式）



- 注 1. 如果根据 ADRCK 位和 ADUL/ADLL 寄存器的设定不产生 A/D 转换结束中断信号（INTAD），就不将结果保存到 ADCR 寄存器和 ADCRH 寄存器，再次转移到 STOP 模式。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。
2. 如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与之后的 SNOOZE 模式和通常运行模式无关。必须将 AWC 位置“0”。

## 9.9 A/D 转换器特性表的阅读方法

以下说明 A/D 转换器特有的专业术语。

### (1) 分辨率

分辨率是能分辨的最小模拟输入电压。也就是说，数字输出的每 1 位与模拟输入电压的比率称为 1 LSB (Least Significant Bit)。将对 1 LSB 满刻度的比率表示为 %FSR (Full Scale Range)。

当分辨率为 12 位时，

$$1 \text{ LSB} = 1/2^{12} = 1/4096 \\ \approx 0.024\% \text{FSR}$$

精度与分辨率无关而取决于综合误差。

### (2) 综合误差

综合误差是指实际测量值和理论值的最大差值，是将零刻度误差、满刻度误差、积分线性误差、微分线性误差以及这些组合所产生的误差综合起来的误差。

特性表中的综合误差不包含量化误差。

### (3) 量化误差

在将模拟值转换为数字值时，必然会出现  $\pm 1/2 \text{LSB}$  的误差。A/D 转换器将  $\pm 1/2 \text{LSB}$  范围内的模拟输入电压转换为相同的数字码，因此不能避免量化误差。

特性表中的综合误差、零刻度误差、满刻度误差、积分线性误差和微分线性误差不包含量化误差。

图 9-38 综合误差

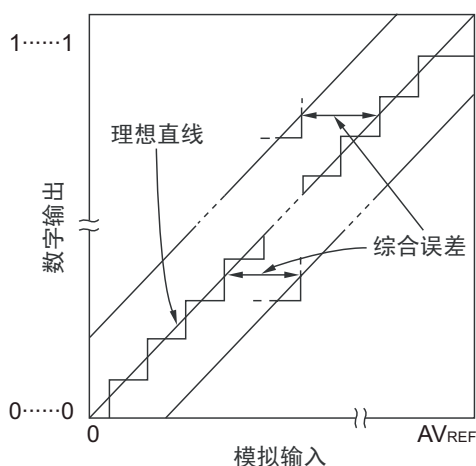
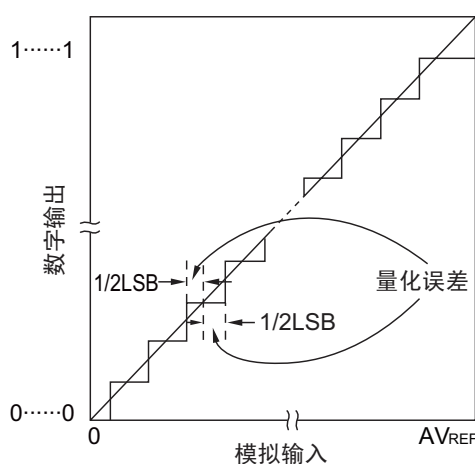


图 9-39 量化误差



## (4) 零刻度误差

零刻度误差是指数字输出从 0.....000 变为 0.....001 时的模拟输入电压的实际测量值和理论值 ( $1/2 \text{ LSB}$ ) 的差。如果实际测量值大于理论值, 零刻度误差就是指数字输出从 0.....001 变为 0.....010 时的模拟输入电压的实际测量值和理论值 ( $3/2 \text{ LSB}$ ) 的差。

## (5) 满刻度误差

满刻度误差是指数字输出从 1.....110 变为 1.....111 时的模拟输入电压的实际测量值和理论值 (满刻度  $-3/2 \text{ LSB}$ ) 的差。

## (6) 积分线性误差

积分线性误差是指转换特性从理想线性关系偏离的程度, 是零刻度误差和满刻度误差为 0 时的实际测量值和理想直线的差的最大值。

## (7) 微分线性误差

微分线性误差是指在输出代码的理想宽度为  $1 \text{ LSB}$  时某个代码输出宽度的实际测量值和理想值的差值。

图 9-40 零刻度误差

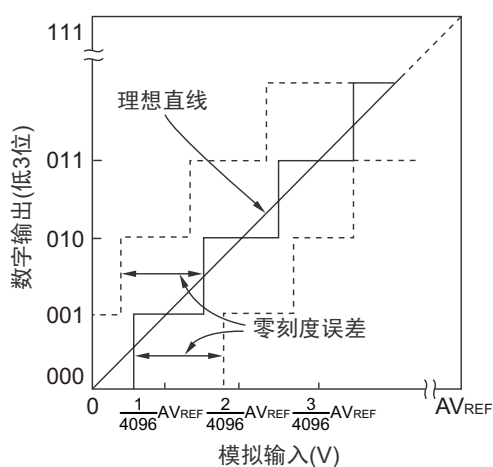


图 9-41 满刻度误差

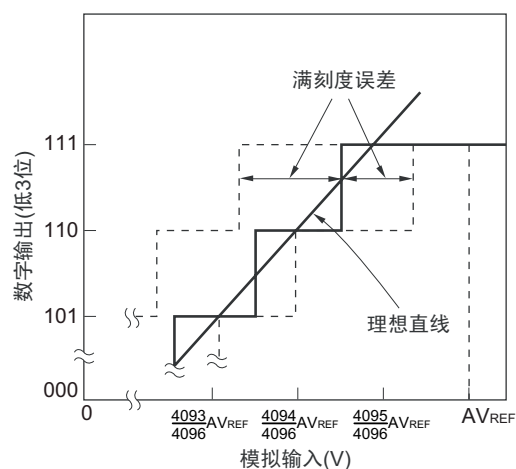


图 9-42 积分线性误差

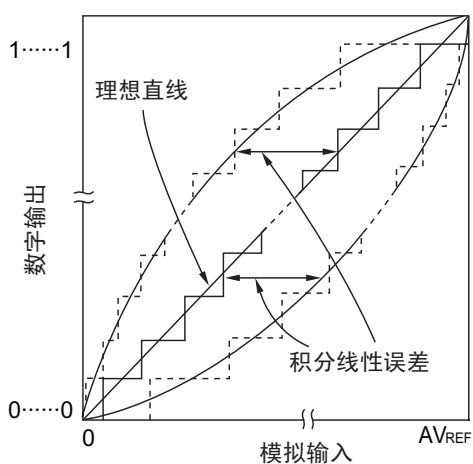
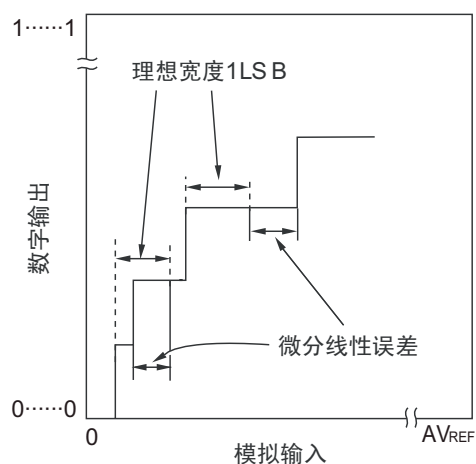


图 9-43 微分线性误差



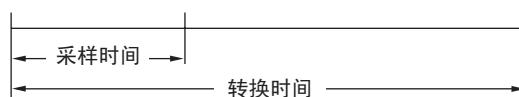
### (8) 转换时间

转换时间是指从开始采样到取得数字输出的时间。

特性表中的转换时间包含采样时间。

### (9) 采样时间

采样时间是指为了将模拟电压取入采样 & 保持电路而使模拟开关 ON 的时间。



## 9.10 A/D 转换器的注意事项

### (1) 有关 STOP 模式中的工作电流

要转移到 STOP 模式时，必须在停止 A/D 转换器（将 A/D 转换器的模式寄存器 0 (ADM0) 的 bit7 (ADCS) 置“0”) 后进行。此时，还能通过将 ADM0 寄存器的 bit0 (ADCE) 置“0”来减小工作电流。

要从待机状态重新运行时，必须在将中断请求标志寄存器 1H (IF1H) 的 bit0 (ADIF) 清“0”后开始运行。

### (2) 有关 ANI0 ~ ANI7 引脚和 ANI16 引脚的输入范围

用 ANI0 ~ ANI7 引脚和 ANI16 引脚的输入电压必须满足规格范围。如果输入大于  $V_{DD}$  和  $AV_{REFP}$  或者小于  $V_{SS}$  和  $AV_{REFM}$ （即使在绝对最大额定范围内）的电压，该通道的转换值就为不定值，并且还可能影响其他通道的转换值。

当选择内部基准电压（1.45V）作为 A/D 转换器的正（+）基准电压源时，不能给 ADS 寄存器选择的引脚输入大于内部基准电压的电压。但是，能给 ADS 寄存器没有选择的引脚输入大于内部基准电压的电压。

**注意** 只有在 HS（高速主）模式中才能选择内部基准电压（1.45V）。详细内容请参照“图 22-3 用户选项字节（000C2H）的格式”。

### (3) 有关竞争

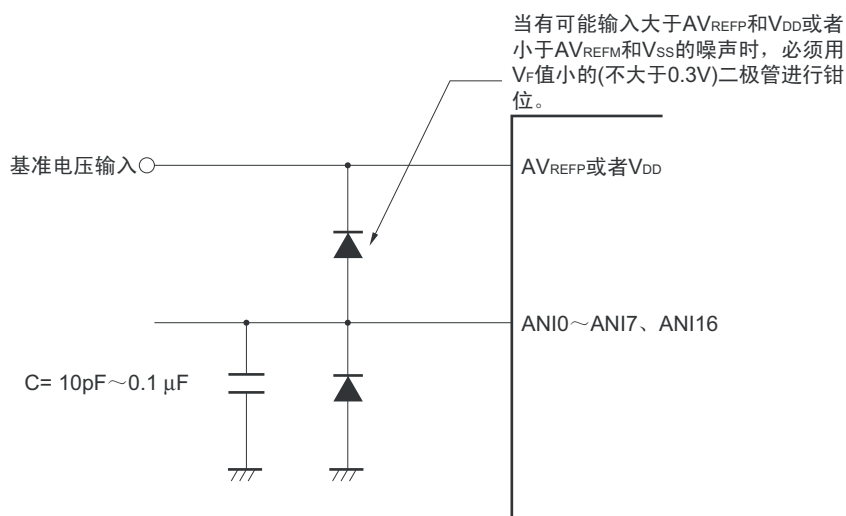
- ① 转换结束时的 A/D 转换结果寄存器（ADCR、ADCRH）的写操作与 ADCR 寄存器和 ADCRH 寄存器的指令读操作的竞争  
优先读 ADCR 寄存器和 ADCRH 寄存器。在读后，将新的转换结果写到 ADCR 寄存器和 ADCRH 寄存器。
- ② 转换结束时的 ADCR 寄存器和 ADCRH 寄存器的写操作与 A/D 转换器的模式寄存器 0（ADM0）、模拟输入通道指定寄存器（ADS）或者 A/D 端口配置寄存器（ADPC）的写操作的竞争  
优先写 ADM0、ADS、ADPC 寄存器。不写 ADCR 寄存器和 ADCRH 寄存器，也不产生转换结束中断信号（INTAD）。

### (4) 有关噪声对策

为了保持 12 位分辨率，必须注意输入到  $AV_{REFP}$ 、 $V_{DD}$ 、ANI0 ~ ANI7、ANI16 引脚的噪声。

- ①  $V_{DD}$  和  $V_{SS}$  必须和其他电源分开，在  $V_{DD}$  和  $V_{SS}$  之间连接的电容器必须使用等效电阻小并且频率响应好的电容器（0.01 $\mu$ F 左右）。
- ② 模拟输入源的输出阻抗越高影响就越大，因此为了降低噪声，建议按照图 9-44 的方法外接 C。
- ③ 不能在转换过程中切换其他引脚。
- ④ 如果在转换开始后立即设定为 HALT 模式，就会提高精度。
- ⑤ 数字信号不能和模拟信号交叉，而且不能靠近必须分离。

图 9-44 模拟输入引脚的处理



#### (5) 模拟输入 (ANIn) 引脚

##### ① $ANI0 \sim ANI7$ 引脚和 $P20 \sim P27$ 引脚复用。

要选择  $ANI0 \sim ANI7$  引脚中的任意引脚进行 A/D 转换时，不能在转换过程中更改  $P20 \sim P27$  的输出值。否则，可能降低转换精度。

##### ② 如果将正在进行 A/D 转换的引脚的相邻引脚用作数字输入/输出端口，就可能因耦合噪声而取得与期待值不同的 A/D 转换值。因此，在 A/D 转换过程中，不能给相邻的引脚输入或者输出类似数字信号的有急剧变化的脉冲。

#### (6) 有关模拟输入 (ANIn) 引脚的输入阻抗

此 A/D 转换器在采样时间内给内部的采样电容充电并且进行采样。

因此，在不采样时只有漏电流流过，而在采样时还有电容充电的电流流过。所以输入阻抗根据是否采样而发生变化。

但是，为了充分地进行采样，必须将模拟输入源的输出阻抗保持在  $1\text{k}\Omega$  以下。建议在无法将输出阻抗保持在  $1\text{k}\Omega$  以下时延长采样时间或者给  $ANI0 \sim ANI7$  引脚和  $ANI16$  引脚连接  $0.1\mu\text{F}$  左右的电容器 (参照图 9-44)。

另外，如果在转换过程中将 ADCS 位置 “0” 或者重新开始转换，采样电容器的充电电压就不稳定。因此，当将 ADCS 位置 “0” 时，下一次转换就从不稳定状态开始充电；当重新转换时，该转换就从不稳定状态开始充电。为了充分地进行充电，与模拟信号变化的大小无关，必须降低模拟输入源的输出阻抗或者确保充分的采样时间。

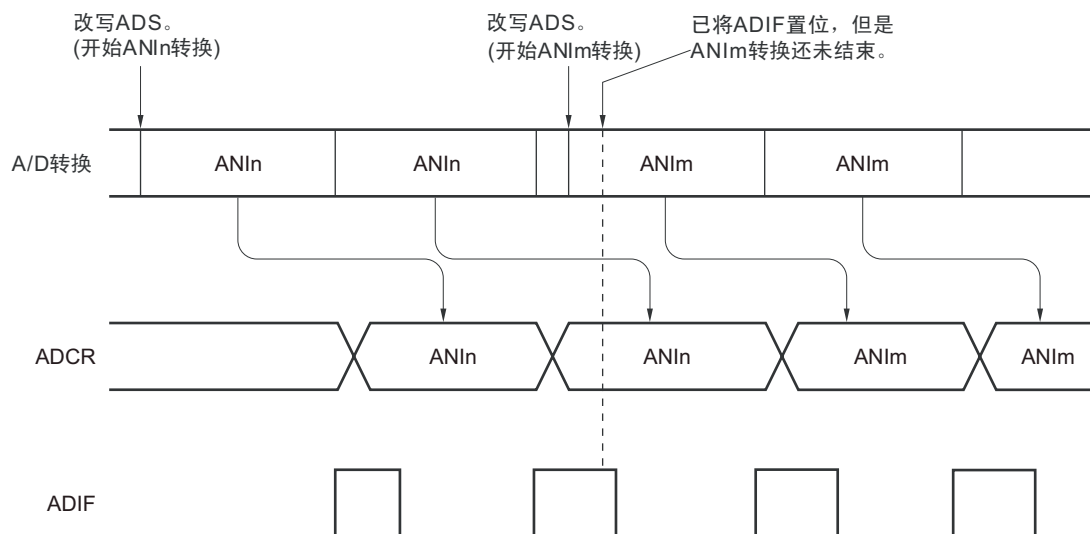
#### (7) 有关中断请求标志 (ADIF)

即使更改模拟输入通道指定寄存器 (ADS)，也不将中断请求标志 (ADIF) 清 “0”。

因此，如果在 A/D 转换过程中更改模拟输入引脚，就可能在即将改写 ADS 寄存器前，更改前的模拟输入的 A/D 转换结果和 ADIF 标志被设定。必须注意：如果在改写 ADS 寄存器后立即读 ADIF 标志，尽管转换后的模拟输入的 A/D 转换还未结束，也会将 ADIF 标志置位。

另外，在暂停后重新开始 A/D 转换时，必须在重新开始前将 ADIF 标志清 “0”。

图 9-45 A/D 转换结束中断请求的产生时序



#### (8) 有关 A/D 转换开始后的初次转换结果

在软件触发模式或者硬件触发无等待模式中，如果在将 ADCE 位置“1”后的  $1\mu\text{s}$  内将 ADCS 位置“1”，开始 A/D 转换后的初次 A/D 转换值就可能不满足额定值。必须采取轮询 A/D 转换结束中断请求 (INTAD) 并且取消初次的转换结果等对策。

#### (9) 有关 A/D 转换结果寄存器 (ADCR、ADCRH) 的读操作

当写 A/D 转换器的模式寄存器 0 (ADM0)、模拟输入通道指定寄存器 (ADS)、A/D 端口配置寄存器 (ADPC) 和端口模式控制寄存器 (PMC) 时，ADCR 寄存器和 ADCRH 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS、ADPC、PMC 寄存器前读转换结果。否则，就可能读不到正确的转换结果。



(10) 有关内部等效电路

模拟输入部的等效电路如下所示。

图 9-46 ANIn 引脚的内部等效电路

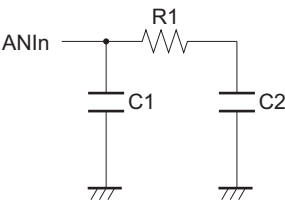


表 9-4 等效电路的各电阻和电容值（参考值）

$V_{DD}$	ANIn 引脚	R1[kΩ]	C1[pF]	C2[pF]
$2.7V \leq V_{DD} \leq 3.6V$	ANI0 ~ ANI7	7.4	8	6.3
	ANI16	12.3	8	7.4

备注 表 9-4 的各电阻和电容值不是保证值。

(11) 有关 A/D 转换器的运行开始

必须在  $AV_{REFP}$  和  $V_{DD}$  的电压稳定后开始 A/D 转换器的运行。

## 第 10 章 D/A 转换器

这是采用 R-2R 方式将数字输入转换为模拟信号的 10 位分辨率的 D/A 转换器，能控制 2 个独立通道的模拟输出。

### 10.1 D/A 转换器的功能

D/A 转换器有以下功能：

- 10 位分辨率×2ch
- R-2R 梯形方式
- 模拟输出电压
  - 10 位分辨率： $V_{DD} \times m10 / 1024$ （m10：DACSi 寄存器的设定值）
- 运行模式
  - 通常模式
  - 实时输出模式

备注 i=0、1

### 10.2 D/A 转换器的结构

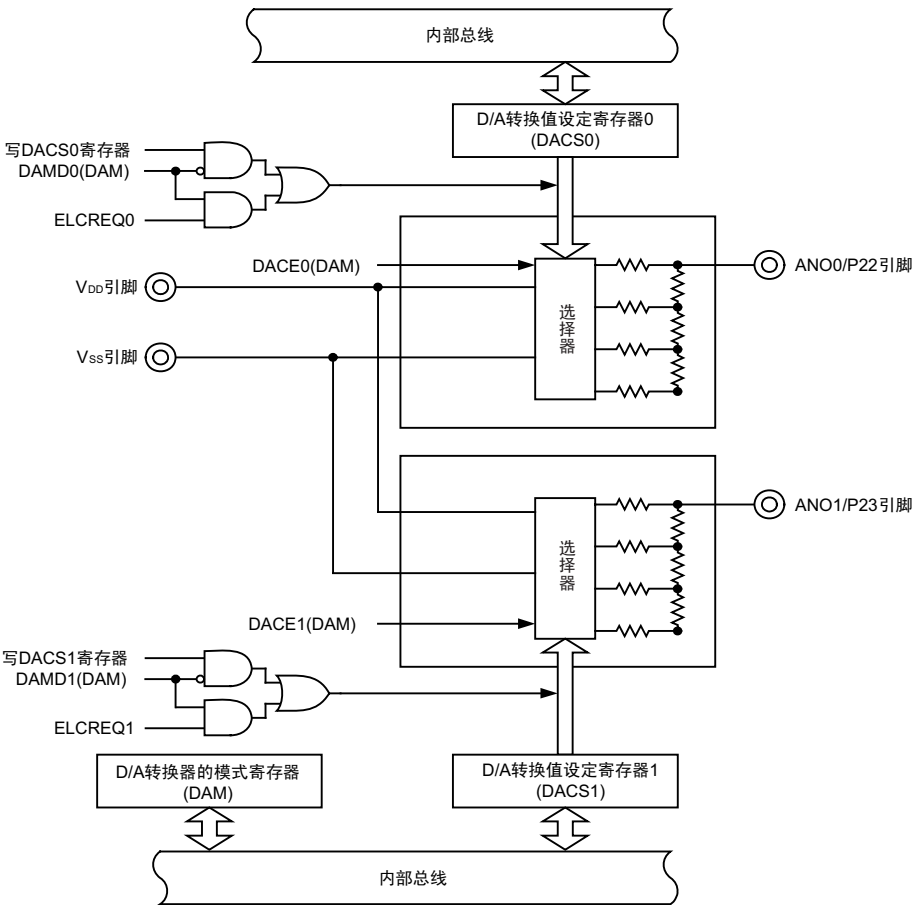
D/A 转换器由以下硬件构成。

表 10-1 D/A 转换器的结构

项目	结构
控制寄存器	A/D 端口配置寄存器（ADPC） 外围允许寄存器 1（PER1） D/A 转换器模式寄存器（DAM） D/A 转换值设定寄存器 0、1（DACs0、DACs1） 端口模式寄存器 2（PM2）

D/A 转换器的框图如图 10-1 所示。

图 10-1 D/A 转换器的框图



备注 ELCREQ0 和 ELCREQ1 是用于实时输出模式的触发信号（ELC 的请求信号）。

### 10.3 D/A 转换器使用的寄存器

D/A 转换器使用以下寄存器：

- A/D 端口配置寄存器（ADPC）
- 外围允许寄存器 1（PER1）
- D/A 转换器的模式寄存器（DAM）
- D/A 转换值设定寄存器 0、1（DACS0、DACS1）
- 端口模式寄存器 2（PM2）

#### 10.3.1 A/D 端口配置寄存器（ADPC）

这是将 ANI0/P20 ~ ANI7/P27 引脚切换为模拟输入或者端口的数字输入 / 输出的寄存器。

通过 8 位存储器操作指令设定 ADPC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-2 A/D 端口配置寄存器（ADPC）的格式

地址：F0076H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
ADPC	ADPC7 注	ADPC6 注	ADPC5 注	ADPC4 注	ADPC3	ADPC2	ADPC1	ADPC0

ADPCn	P2n/ANI2n 的模拟输入（A）和数字输入 / 输出（D）的选择
0	模拟输入（A）（默认值）
1	数字输入 / 输出（D）

注 只限于 32 引脚产品。

注意 1. 必须通过端口模式寄存器 2（PM2）将 D/A 转换使用的通道设定为输入模式。

2. 对于由 ADPC 寄存器设定为数字输入/输出的引脚，不能通过 D/A 转换器的模式寄存器（DAM）设定为允许 D/A 转换。

10.3.2 外围允许寄存器 1（PER1）

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

要使用 D/A 转换器时，必须将 bit7（DACEN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-3 外围允许寄存器 1（PER1）的格式

地址: F007AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER1	DACEN	0	0	0	0	0	0	0

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

- 注意 1. 在设定 D/A 转换器时，必须先将 DACEN 位置“1”。
- 如果 DACEN 位为“0”，就忽视 D/A 转换器的控制寄存器的写操作，而且读取值也都为初始值（端口模式寄存器 2（PM2）和端口寄存器 2（P2）除外）。
2. 必须将 bit0 ~ 6 置“0”。

10.3.3 D/A 转换器的模式寄存器（DAM）

这是控制 D/A 转换器运行的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 DAM 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 10-4 D/A 转换器的模式寄存器（DAM）的格式

地址：FFF5CH

复位后：00H

R/W

符号

7

6

5

4

3

2

1

0

DAM

—	—	DACE1	DACE0	—	—	DAMD1	DAMD0
---	---	-------	-------	---	---	-------	-------

DACEi	D/A 转换器的转换运行的控制
0	停止 D/A 转换。
1	允许 D/A 转换。

DAMDi	D/A 转换器的运行模式的选择
0	通常运行模式
1	实时输出模式

备注 i=0、1

10.3.4 D/A 转换值设定寄存器 i（DACSi）（i=0、1）

这是设定在使用 D/A 转换器时输出到 ANO0 引脚和 ANO1 引脚的模拟电压值的寄存器。  
通过 16 位存储器操作指令设定 DACSi 寄存器。  
在产生复位信号后，这些寄存器的值变为“00H”。

图 10-5 D/A 转换值设定寄存器 i（DACSi）（i=0、1）的格式

地址：FFF58H、FFF59H（DACS0）、FFF5AH、FFF5BH（DACS1）										复位后：0000H		R/W				
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACSi	0	0	0	0	0	0	DAC Si9	DAC Si8	DAC Si7	DAC Si6	DAC Si5	DAC Si4	DAC Si3	DAC Si2	DAC Si1	DAC Si0

备注 D/A 转换器的模拟输出电压（VANOi）如下：  
 $VANOi = \text{D/A 转换器的基准电压} \times (\text{DACSi}) / 1024$

当不使用 D/A 转换器时，为了减小不必要的消费电流，必须将 DACEi 位置“0”（禁止输出）并且将 DACSi 寄存器置“00H”，使 R-2R 的电阻没有电流流过。

10.3.5 端口模式寄存器 2（PM2）

当将 ANO0/ANI2/P22 引脚和 ANO1/ANI3/P23 引脚用作模拟输入端口时，必须将 PM22 位和 PM23 位分别置“1”。

如果将 PM22 位和 P23 位分别置“0”，就不能用作模拟输入端口。

通过 1 位或者 8 位存储器操作指令设定 PM2 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

注意 当读被设定为模拟输入端口的引脚时，读取值不是引脚电平而总是“0”。

图 10-6 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 注	PM25 注	PM24 注	PM23	PM22	PM21	PM20

PM2n	P2n 引脚的输入 / 输出模式的选择（n=0 ~ 7）
0	输出模式（输出缓冲器 ON）
1	输入模式（输出缓冲器 OFF）

注 24 引脚产品没有此位。

ANO0/ANI2/P22 引脚和 ANO1/ANI3/P23 引脚的功能取决于 A/D 端口配置寄存器（ADPC）、D/A 转换器的模式寄存器（DAM）、模拟输入通道指定寄存器（ADS）和 PM2 寄存器的设定。

表 10-2 ANO0/ANI2/P22 引脚和 ANO1/ANI3/P23 引脚的功能设定

ADPC	PM2	DAM	ADS	ANO0/ANI2/P22 引脚和 ANO1/ANI3/P23 引脚的功能
数字输入 / 输出的选择	输入模式	—	—	数字输入
	输出模式	—	—	数字输出
模拟输入的选择	输入模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	模拟输出
		停止 D/A 转换。	选择 ANI。	模拟输入 （A/D 转换对象）
			不选择 ANI。	模拟输入 （非 A/D 转换对象）
	输出模式	允许 D/A 转换。	选择 ANI。	禁止设定。
			不选择 ANI。	
		停止 D/A 转换。	选择 ANI。	
			不选择 ANI。	

## 10.4 D/A 转换器的运行

### 10.4.1 通常模式中的运行

将 DACSi 寄存器的写操作作为启动触发，进行 D/A 转换。

其设定方法如下所示：

- ① 将 PER1 寄存器（外围允许寄存器 1）的 DACEN 位置“1”，开始给 D/A 转换器提供输入时钟。
- ② 通过 ADPC 寄存器（端口配置寄存器）将端口设定为模拟引脚。
- ③ 将 DAM 寄存器（D/A 转换器的模式寄存器）的 DAMDi 位置“0”（通常模式）。
- ④ 给 DACSi 寄存器（D/A 转换值设定寄存器 i）设定 ANOi 引脚输出的模拟电压值。

以上①～④为初始设定。

- ⑤ 将 DAM 寄存器的 DACEi 位置“1”（允许 D/A 转换）。  
开始 D/A 转换，在经过稳定时间后将④设定的模拟电压输出到 ANOi 引脚。
- ⑥ 此后，要进行 D/A 转换时，写 DACSi 寄存器。

在进行下一次 D/A 转换前，保持前一次 D/A 转换的结果。

如果将 DAM 寄存器的 DACEi 位置“0”（停止 D/A 转换），就停止 D/A 转换。

在通过 ADPC 寄存器将端口设定为数字引脚的情况下，当端口的 PM2 寄存器的 PM2i 位为“1”（输入模式）时 ANOi 引脚为高阻抗；当 PM2i 位为“0”（输出模式）时 ANOi 引脚输出 P2 寄存器的设定值。

注意 1. 即使将 DACEi 位的设定值进行“1”→“0”→“1”的设定，也在最后置“1”后产生等待时间。

2. 如果在稳定时间内改写 DACSi 寄存器，就中止转换并且以改写的值重新开始转换。

备注 i=0、1



### 10.4.2 实时输出模式中的运行

D/A 转换器的各通道将 ELC 的独立请求信号作为启动触发，进行 D/A 转换。  
其设定方法如下所示：

- ① 将 PER1 寄存器（外围允许寄存器 1）的 DACEN 位置“1”，开始给 D/A 转换器提供输入时钟。
- ② 通过 ADPC 寄存器（端口配置寄存器）将端口设定为模拟引脚。
- ③ 将 DAM 寄存器（D/A 转换器的模式寄存器）的 DAMDi 位置“0”（通常模式）。
- ④ 给 DACSi 寄存器（D/A 转换值设定寄存器 i）设定 ANOi 引脚输出的模拟电压值。
- ⑤ 将 DAM 寄存器的 DACEi 位置“1”（允许 D/A 转换）。  
开始 D/A 转换，在经过稳定时间后将③设定的模拟电压输出到 ANOi 引脚。
- ⑥ 通过 ELSELR（ELC 控制寄存器）设定实时触发信号。
- ⑦ 将 DAM 寄存器的 DAMDi 位置“1”（实时输出模式）。
- ⑧ 开始 ELC 请求源的运行。

以上①～⑧为初始设定。

- ⑨ 此后，通过产生实时输出触发信号，开始 D/A 转换，在经过稳定时间后将④设定的模拟电压输出到 ANOi 引脚。  
在进行下一次 D/A 转换（产生实时输出触发信号）前，给 DACSi 寄存器设定 ANOi 引脚输出的模拟电压值。

必须在进行下一次 D/A 转换（产生 ELC 触发信号）前给 DACSi 寄存器设定 ANOi 引脚输出的模拟电压值。  
如果将 DAM 寄存器的 DACEi 位置“0”（停止 D/A 转换），就停止 D/A 转换。

在通过 ADPC 寄存器将端口设定为数字引脚的情况下，当端口的 PM2 寄存器的 PM2i 位为“1”（输入模式）时 ANOi 引脚为高阻抗；当 PM2i 位为“0”（输出模式）时 ANOi 引脚输出 P2 寄存器的设定值。

注意 1. 即使将 DACEi 位的设定值进行“1”→“0”→“1”的设定，也在最后置“1”后产生等待时间。

2. ELC 事件请求触发信号的产生间隔必须大于稳定时间。如果在稳定时间内产生 ELC 事件请求触发信号，就中止 D/A 转换并且重新开始转换。
3. 即使 ELC 事件请求触发信号的产生时序和 DACSi 寄存器的改写时序发生竞争，也正常输出 D/A 转换结果。

## 10.5 使用 D/A 转换器时的注意事项

使用 D/A 转换器时的注意事项如下所示。

- (1) 当通过 ADPC 寄存器（端口配置寄存器）将端口设定为模拟引脚时，和 ANO0 引脚、ANO1 引脚复用的数字端口的输入/输出功能不工作。如果在通过 ADPC 寄存器将端口设定为模拟引脚的过程中读 P2 寄存器，就在输入模式中读取值为“0”，而在输出模式中读取值为 P2 的设定值。另外，即使设定为数字输出模式，也不将输出数据输出到引脚。
- (2) 在 HALT 模式和 STOP 模式中，D/A 转换器继续运行。为了降低功耗，必须将 DACEi 位清“0”，在停止 D/A 转换后执行 HALT 指令或者 STOP 指令。

备注 i=0、1

- (3) 在停止实时输出模式时（包括变为通常模式的情况）需要按照以下步骤进行：
  - 在停止触发输出源后至少等待 3 个时钟，然后将 DACEi 位和 DAMDi 位置“0”。
  - 在将 DACEi 位和 DAMDi 位置“0”后将 PER1 寄存器的 DACEN 位置“0”（停止 DAC）。
  - 如果将 DACEN 位置“0”，就清除 DAC 内部的全部寄存器。因此，在重新开始运行时，需要设定各 SFR。
- (4) 当允许 D/A 转换时，不能对和 ANO0 引脚、ANO1 引脚复用的模拟输入引脚进行 A/D 转换。
- (5) 在实时输出模式中，必须在产生定时器触发前设定 DACSi 寄存器的值。  
在触发信号有效期间，不能更改 DACSi 寄存器的设定值。

## 第 11 章 串行阵列单元

串行阵列单元 0 有 1 个串行通道，能实现 3 线串行（CSI）和 UART 的通信功能。

R7F0C010 支持的各通道的功能分配如下：

单元	通道	用作 CSI	用作 UART
0	0	CSI00（支持从属选择输入功能）	UART0
	1	—	

在使用 UART0 时，不能使用 CSI00。

### 11.1 串行阵列单元的功能

R7F0C010 支持的各串行接口的特征如下所示。

#### 11.1.1 3 线串行 I/O（CSI00）

与主控设备输出的串行时钟（SCK）同步进行数据的发送和接收。

这是使用 1 条串行时钟（SCK）、1 条发送串行数据（SO）和 1 条接收串行数据（SI）共 3 条通信线进行通信的时钟同步通信功能。

有关具体的设定例子，请参照“11.5 3 线串行 I/O（CSI00）通信的运行”。

[ 数据的发送和接收 ]

- 7 位或者 8 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[ 时钟控制 ]

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

主控通信：Max.  $f_{MCK}/2$

从属通信：Max.  $f_{MCK}/6$

[ 中断功能 ]

- 传送结束中断、缓冲器空中断

[ 错误检测标志 ]

- 溢出错误

CSI00 支持 SNOOZE 模式。SNOOZE 模式功能是指：如果在 STOP 模式的状态下检测到 SCK 的输入，就不需要 CPU 运行而接收数据。

注 必须在满足 SCK 周期时间（ $t_{KCY}$ ）特性的范围内使用（参照“第 27 章 电特性”）。

### 11.1.2 UART (UART0)

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

[ 数据的发送和接收 ]

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

[ 中断功能 ]

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

[ 错误检测标志 ]

- 帧错误、奇偶校验错误、溢出错误

UART0 支持 SNOOZE 模式。SNOOZE 模式功能是指: 如果在 STOP 模式的状态下检测到 RxD 的输入, 就不需要 CPU 运行而接收数据。支持接收时的波特率调整功能。

## 11.2 串行阵列单元的结构

串行阵列单元由以下硬件构成。

表 11-1 串行阵列单元的结构

项目	结构
移位寄存器	9 位
缓冲寄存器	串行数据寄存器 mn (SDRmn) 的低 9 位 <sup>注</sup>
串行时钟输入 / 输出	SCK00 引脚 (用于 3 线串行 I/O)
串行数据输入	SI00 引脚 (用于 3 线串行 I/O)、RxD0 引脚 (用于 UART)
串行数据输出	SO00 引脚 (用于 3 线串行 I/O)、TxD0 引脚 (用于 UART)、输出控制电路
从属选择输入	SSI00 引脚 (用于从属选择输入功能)
控制寄存器	<div> <div>&lt;单元设定部的寄存器&gt;</div> <ul style="list-style-type: none"> <li>• 外围允许寄存器 0 (PER0)</li> <li>• 串行时钟选择寄存器 m (SPSm)</li> <li>• 串行通道允许状态寄存器 m (SEm)</li> <li>• 串行通道开始寄存器 m (SSm)</li> <li>• 串行通道停止寄存器 m (STm)</li> <li>• 串行输出允许寄存器 m (SOEm)</li> <li>• 串行输出寄存器 m (SOM)</li> <li>• 串行输出电平寄存器 m (SOLm)</li> <li>• 串行待机控制寄存器 m (SSCm)</li> <li>• 输入切换控制寄存器 (ISC)</li> <li>• 噪声滤波器允许寄存器 0 (NFEN0)</li> </ul> </div> <div> <div>&lt;各通道部的寄存器&gt;</div> <ul style="list-style-type: none"> <li>• 串行数据寄存器 mn (SDRmn)</li> <li>• 串行模式寄存器 mn (SMRmn)</li> <li>• 串行通信运行设定寄存器 mn (SCRmn)</li> <li>• 串行状态寄存器 mn (SSRmn)</li> <li>• 串行标志清除触发寄存器 mn (SIRmn)</li> </ul> </div> <div> <ul style="list-style-type: none"> <li>• 端口模式寄存器 3 (PM3)</li> <li>• 端口寄存器 3 (P3)</li> </ul> </div>

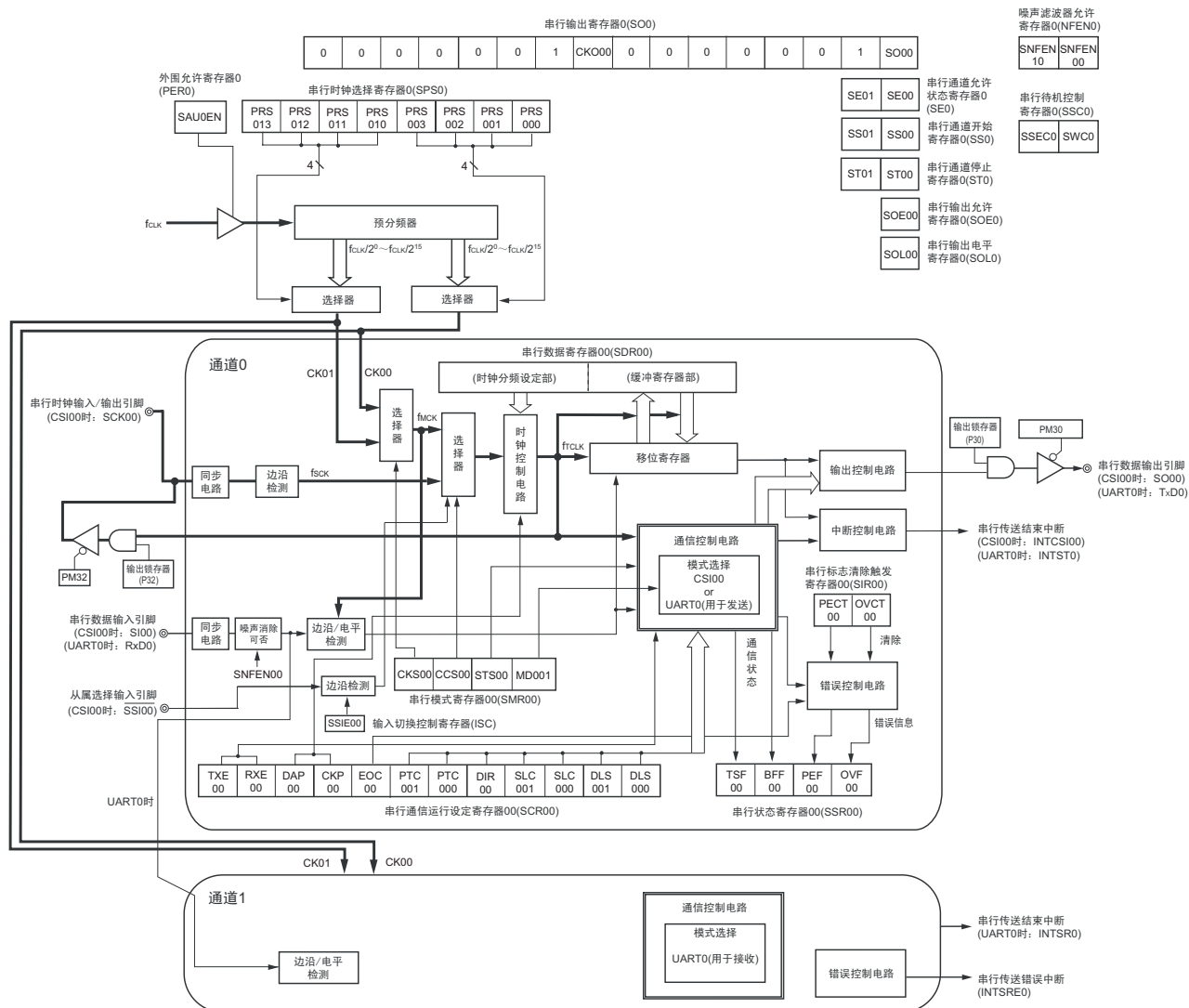
注 根据通信模式，能用以下 SFR 名称读写串行数据寄存器 mn (SDRmn) 的低 8 位。

- 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)

备注 m: 单元号 (m=0) n: 通道号 (n=0、1) p: CSI 号 (p=00) q: UART 号 (q=0)

串行阵列单元 0 的框图如图 11-1 所示。

图 11-1 串行阵列单元 0 的框图



### 11.2.1 移位寄存器

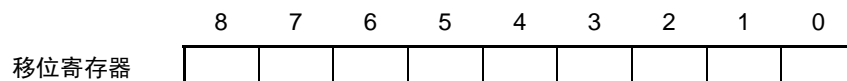
这是进行并行和串行相互转换的 9 位寄存器。

在以 9 位数据长度进行 UART 通信时，使用 9 位（bit0 ~ 8）注 1。

在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出。

不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，使用串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位。



### 11.2.2 串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位

SDRmn 寄存器是通道 n 的发送和接收数据寄存器（16 位）。

bit8 ~ 0（低 9 位）或者 bit7 ~ 0（低 8 位）用作发送和接收缓冲寄存器，bit15 ~ 9 用作运行时钟（f<sub>MCK</sub>）的分频设定寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到低 8 位或者低 9 位；在发送数据时，将被传送到移位寄存器的发送数据设定到低 8 位或者低 9 位。

与数据的输出顺序无关，根据串行通信运行设定寄存器 mn（SCRmn）的 bit0 和 bit1（DLSmn0、DLSmn1）的设定，保存到低 8 位或者低 9 位的数据如下所示：

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

能以 16 位为单位读写 SDRmn 寄存器。

根据通信模式，能用以下 SFR 名称，以 8 位为单位读写 SDRmn 寄存器的低 8 位或者低 9 位注。

- 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）

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

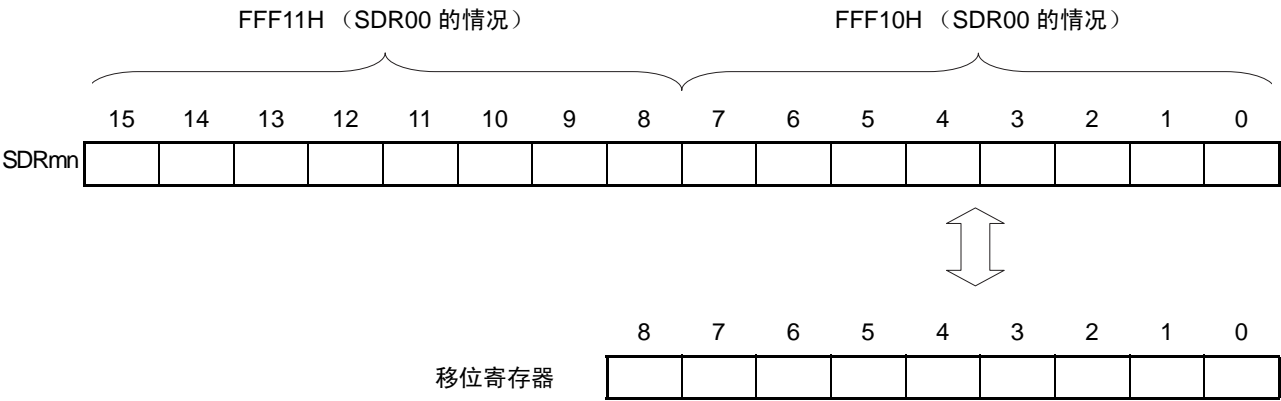
注 在停止运行（SEmn=0）时，禁止以 8 位为单位的写操作。

备注 1. 在接收结束后，bit0 ~ 8 中超过数据长度的部分的位为“0”。

2. m: 单元号（m=0） n: 通道号（n=0、1） p: CSI 号（p=00） q: UART 号（q=0）

图 11-2 串行数据寄存器 mn（SDRmn）（mn=00）的格式

地址：FFF10H、FFF11H（SDR00）、FFF12H、FFF13H（SDR01）



备注 有关 SDRmn 寄存器的高 7 位的功能，请参照“11.3 控制串行阵列单元的寄存器”。



### 11.3 控制串行阵列单元的寄存器

控制串行阵列单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 串行时钟选择寄存器m (SPSm)
- 串行模式寄存器mn (SMRmn)
- 串行通信运行设定寄存器mn (SCRmn)
- 串行数据寄存器mn (SDRmn)
- 串行标志清除触发寄存器mn (SIRmn)
- 串行状态寄存器mn (SSRmn)
- 串行通道开始寄存器m (SSm)
- 串行通道停止寄存器m (STm)
- 串行通道允许状态寄存器m (SEm)
- 串行输出允许寄存器m (SOEm)
- 串行输出电平寄存器m (SOLm)
- 串行输出寄存器m (SOM)
- 串行待机控制寄存器m (SSCm)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器允许寄存器0 (NFEN0)
- 端口模式寄存器3 (PM3)
- 端口寄存器3 (P3)

备注 m: 单元号 (m=0) n: 通道号 (n=0、1)

### 11.3.1 外围允许寄存器 0 (PER0)

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

要使用串行阵列单元 0 时，必须将 bit2 (SAU0EN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，PER0 寄存器的值变为“00H”。

图 11-3 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	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

SAUmEN	提供串行阵列单元 m 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> <li>不能写串行阵列单元 m 使用的 SFR。</li> <li>串行阵列单元 m 处于复位状态。</li> </ul>
1	允许提供输入时钟。 <ul style="list-style-type: none"> <li>能读写串行阵列单元 m 使用的 SFR。</li> </ul>

注意 1. 要设定串行阵列单元 m 时，必须先在 SAUmEN 位为“1”的状态下设定以下的寄存器。当 SAUmEN 位为“0”时，忽视串行阵列单元 m 的控制寄存器的写操作，而且读取值都为初始值（输入切换控制寄存器 (ISC)、噪声滤波器允许寄存器 0 (NFEN0)、端口模式寄存器 3 (PM3) 以及端口寄存器 3 (P3) 除外）。

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

2. 必须将 bit7、3、1 置“0”。

### 11.3.2 串行时钟选择寄存器 m (SPSm)

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

禁止在运行过程中 (SEmn=1) 改写 SPSm 寄存器。

通过 16 位存储器操作指令设定 SPSm 寄存器。

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

在产生复位信号后，SPSm 寄存器的值变为“0000H”。

图 11-4 串行时钟选择寄存器 m (SPSm) 的格式

地址: F0126H、F0127H      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 mk3	PRS mk2	PRS mk1	PRS mk0	运行时钟 (CKmk) 的选择 <sup>注</sup>					
					f <sub>CLK</sub> =2MHz	f <sub>CLK</sub> =5MHz	f <sub>CLK</sub> =10MHz	f <sub>CLK</sub> =20MHz	f <sub>CLK</sub> =32MHz
0	0	0	0	f <sub>CLK</sub>	2MHz	5MHz	10MHz	20MHz	32MHz
0	0	0	1	f <sub>CLK</sub> /2	1MHz	2.5MHz	5MHz	10MHz	16MHz
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500kHz	1.25MHz	2.5MHz	5MHz	8MHz
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250kHz	625kHz	1.25MHz	2.5MHz	4MHz
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125kHz	313kHz	625kHz	1.25MHz	2MHz
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5kHz	156kHz	313kHz	625kHz	1MHz
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.3kHz	78.1kHz	156kHz	313kHz	500kHz
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.6kHz	39.1kHz	78.1kHz	156kHz	250kHz
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81kHz	19.5kHz	39.1kHz	78.1kHz	125kHz
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91kHz	9.77kHz	19.5kHz	39.1kHz	62.5kHz
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95kHz	4.88kHz	9.77kHz	19.5kHz	31.3kHz
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	977Hz	2.44kHz	4.88kHz	9.77kHz	15.6kHz
1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	488Hz	1.22kHz	2.44kHz	4.88kHz	7.81kHz
1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	244Hz	610Hz	1.22kHz	2.44kHz	3.91kHz
1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	122Hz	305Hz	610Hz	1.22kHz	1.95kHz
1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	61Hz	153kHz	305Hz	610Hz	977Hz

注 要在串行阵列单元 (SAU) 运行过程中更改被选择为 f<sub>CLK</sub> 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 时，必须在停止 SAU 的运行 (串行通道停止寄存器 m (STm) =000FH) 后进行更改。

注意 必须将 bit15 ~ 8 置“0”。

备注 1. f<sub>CLK</sub>: CPU/ 外围硬件的时钟频率

2. m: 单元号 (m=0)

3. k=0、1

11.3.3 串行模式寄存器 mn（SMRmn）

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

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

通过 16 位存储器操作指令设定 SMRmn 寄存器。

在产生复位信号后，SMRmn 寄存器的值变为“0020H”。

图 11-5 串行模式寄存器 mn（SMRmn）的格式 (1/2)

地址: F0110H、F0111H（SMR00）、F0112H、F0113H（SMR01）

复位后: 0020H

R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn 注	0	SISmn0 注	1	0	0	0	MDmn1	MDmn0

CKSmn	通道 n 运行时钟（f <sub>MCK</sub> ）的选择
0	SPSm 寄存器设定的运行时钟 CKm0
1	SPSm 寄存器设定的运行时钟 CKm1
运行时钟（f <sub>MCK</sub> ）用于边沿检测电路。通过设定 CCSmn 位和 SDRmn 寄存器的高 7 位，生成传送时钟（f <sub>TCLK</sub> ）。	

CCSmn	通道 n 传送时钟（f <sub>TCLK</sub> ）的选择
0	CKSmn 位指定的运行时钟 f <sub>MCK</sub> 的分频时钟
1	来自 SCKp 引脚的输入时钟 f <sub>SCK</sub> （CSI 模式的从属传送）
传送时钟 f <sub>TCLK</sub> 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当 CCSmn 位为“0”时，通过 SDRmn 寄存器的高 7 位进行运行时钟（f <sub>MCK</sub> ）的分频设定。	

STSmn 注	开始触发源的选择
0	只有软件触发有效（在 CSI、UART 发送时选择）。
1	RxDq 引脚的有效边沿（在 UART 接收时选择）
在将 SSm 寄存器置“1”后满足上述条件时，开始传送。	

图 11-5 串行模式寄存器 mn (SMRmn) 的格式 (2/2)

地址: F0110H、F0111H (SMR00)、F0112H、F0113H (SMR01)      复位后: 0020H      R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn 注	0	SIS mn0 注	1	0	0	0	MD mn1	MD mn0

SIS mn0 注	UART 模式中的通道 n 接收数据的电平反相控制
0	将下降沿检测为起始位。 不将输入的通信数据进行反相。
1	将上升沿检测为起始位。 将输入的通信数据进行反相。

MD mn1	通道 n 运行模式的设定
0	CSI 模式
1	UART 模式

MDmn0	通道 n 中断源的选择
0	传送结束中断
1	缓冲器空中断 (在数据从 SDRmn 寄存器传送到移位寄存器时发生)

在连续发送时，如果 MDmn0 位为“1”并且 SDRmn 的数据为空，就写下下一个发送数据。

注 只限于 SMR01 寄存器。

**注意** 必须将 bit13 ~ 9、7、4 ~ 2（在 SMR00 寄存器时，为 bit13 ~ 6、4 ~ 2）置“0”，并且将 bit5 置“1”。

备注 m: 单元号 (m=0) n: 通道号 (n=0、1) p: CSI 号 (p=00) q: UART 号 (q=0)

#### 11.3.4 串行通信运行设定寄存器 mn (SCRmn)

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

禁止在运行过程中（SEmn=1）改写 SCRmn 寄存器。

通过 16 位存储器操作指令设定 SCR<sub>mn</sub> 寄存器。

在产生复位信号后，SCRmn 寄存器的值变为“0087H”。

图 11-6 串行通信运行设定寄存器 mn (SCRmn) 的格式 (1/2)

地址: F0118H、F0119H (SCR00)、F011AH、F011BH (SCR01) 复位后: 0087H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 注1	SLC mn0	0	1	DLSm n1	DLS mn0

TXE mn	RXE mn	通道 n 运行模式的设定
0	0	禁止通信。
0	1	只进行接收。
1	0	只进行发送。
1	1	进行发送和接收。

DAP mn	CKP mn	CSI 模式中的数据与时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4
在 UART 模式中, 必须将 DAPmn 位和 CKPmn 位都置“0”。			

EOC mn	错误中断信号 (INTSREx (x=0)) 的屏蔽控制
0	禁止产生错误中断 INTSREx (产生 INTSRx)。
1	允许产生错误中断 INTSREx (在发生错误时不产生 INTSRx)。
在 CSI 模式中或者在 UART 发送时, 必须将 EOCmn 位置“0”注 2。	

注 1. 只限于 SCR00 寄存器。

2. 在 EOCmn 位为“0”并且不使用 CSImn 时, 有可能产生错误中断 INTSREn。

注意 必须将 bit3、6、11 置“0” (也必须将 SCR01 寄存器的 bit5 置“0”), 并且将 bit2 置“1”。

备注 m: 单元号 (m=0) n: 通道号 (n=0、1) p: CSI 号 (p=00)

图 11-6 串行通信运行设定寄存器 mn (SCRmn) 的格式 (2/2)

地址: F0118H、F0119H (SCR00)、F011AH、F011BH (SCR01) 复位后: 0087H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 注1	SLC mn0	0	1	DLSm n1	DLS mn0

PTC mn1	PTC mn0	UART 模式中的奇偶校验位的设定	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出零校验注2。	不判断奇偶校验。
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。

在 CSI 模式中，必须将 PTCmn1 位和 PTCmn0 位都置“0”。

DIR mn	CSI 和 UART 模式中的数据传送顺序的选择
0	进行 MSB 优先的输入 / 输出。
1	进行 LSB 优先的输入 / 输出。

SLCm n1 注1	SLC mn0	UART 模式中的停止位的设定
0	0	无停止位
0	1	停止位长度 =1 位
1	0	停止位长度 =2 位 (只限于 mn=00)
1	1	禁止设定。

如果选择了传送结束中断，就在传送完所有停止位后产生中断。  
 在 UART 接收时，必须设定为 1 个停止位 (SLCmn1、SLCmn0=0、1)。  
 在 CSI 模式中，必须设定为无停止位 (SLCmn1、SLCmn0=0、0)。  
 在 UART 发送时，必须设定为 1 位 (SLCmn1、SLCmn0=0、1) 或者 2 位 (SLCmn1、SLCmn0=1、0)。

DLSm n1 注2	DLS mn0	CSI 和 UART 模式中的数据长度的设定
0	1	9 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 8) (只在 UART 模式中可选择)
1	0	7 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 6)
1	1	8 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 7)
其他		禁止设定。

注 1. 只限于 SCR00 寄存器。

2. 与数据的内容无关，总是附加“0”。

注意 必须将 bit3、6、11 置“0” (也必须将 SCR01 寄存器的 bit5 置“0”)，并且将 bit2 置“1”。

备注 m: 单元号 (m=0) n: 通道号 (n=0、1) p: CSI 号 (p=00)

### 11.3.5 串行数据寄存器 mn (SDRmn) 的高 7 位

SDRmn 寄存器是通道 n 发送和接收的数据寄存器 (16 位)。

SDR00、SDR01 的 bit8 ~ 0 (低 9 位) 用作发送和接收缓冲寄存器, bit15 ~ 9 用作运行时钟 ( $f_{MCK}$ 、 $f_{SCK}$ ) 的分频设定寄存器。

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

如果将 CCSmn 位置“1”, 就必须将 SDR00、SDR01 的 bit15 ~ 9 (高 7 位) 置“0000000B”。SCKp 引脚的输入时钟  $f_{SCK}$  (CSI 模式的从属传送) 为传送时钟。

SDRmn 寄存器的低 8 位或者低 9 位用作发送和接收缓冲寄存器。在接收数据时, 将移位寄存器转换的并行数据保存到低 9 位; 在发送数据时, 将被传送到移位寄存器的发送数据设定到低 9 位。

能以 16 位为单位读写 SDRmn 寄存器。但是, 只有在运行停止状态 (SEmn=0) 时才能读写高 7 位。在运行中 (SEmn=1) 只能写 SDRmn 寄存器的低 9 位, 而且 SDRmn 寄存器的读取值总是“0”。

在产生复位信号后, SDRmn 寄存器的值变为“0000H”。

图 11-7 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

FFF11H (SDR00 的情况)

FFF10H (SDR00 的情况)

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

SDRmn[15:9]							运行时钟分频的传送时钟设定	
0	0	0	0	0	0	0	$f_{MCK}/2$ 、 $f_{SCK}/2$ (CSI 从属的情况)	
0	0	0	0	0	0	1	$f_{MCK}/4$	
0	0	0	0	0	1	0	$f_{MCK}/6$	
0	0	0	0	0	1	1	$f_{MCK}/8$	
•	•	•	•	•	•	•	•	
•	•	•	•	•	•	•	•	
•	•	•	•	•	•	•	•	
1	1	1	1	1	1	0	$f_{MCK}/254$	
1	1	1	1	1	1	1	$f_{MCK}/256$	

注意 1. 在使用 UART 时, 禁止将 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

2. 在运行停止状态 (SEmn=0) 时, 不能以 8 位为单位写低 8 位 (否则, 高 7 位被清“0”)。

备注 1. 有关 SDRmn 寄存器的低 9 位的功能, 请参照“11.2 串行阵列单元的结构”。

2. m: 单元号 (m=0)    n: 通道号 (n=0、1)



### 11.3.6 串行标志清除触发寄存器 mn（SIRmn）

这是用于清除通道 n 各错误标志的触发寄存器。

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

通过 16 位存储器操作指令设定 SIRmn 寄存器。

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

在产生复位信号后，SIRmn 寄存器的值变为“0000H”。

图 11-8 串行标志清除触发寄存器 mn（SIRmn）的格式

地址: F0108H、F0109H (SIR00)、F010AH、F010BH (SIR01)

复位后: 0000H

R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn注	PEC Tmn	OVC Tmn

FEC Tmn	通道 n 帧错误标志的清除触发														
0	不清除。														
1	将 SSRmn 寄存器的 FEFmn 位清“0”。														

PEC Tmn	通道 n 奇偶校验错误标志的清除触发														
0	不清除。														
1	将 SSRmn 寄存器的 PEFmn 位清“0”。														

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

注 只限于 SIR01 寄存器。

注意 必须将 bit15～3（在 SIR00 寄存器时，为 bit15～2）置“0”。

备注 1. m：单元号（m=0） n：通道号（n=0、1）

2. SIRmn 寄存器的读取值总是“0000H”。

11.3.7 串行状态寄存器 mn（SSRmn）

SSRmn 寄存器表示通道 n 的通信状态和发生错误的情况。表示的错误为帧错误、奇偶校验错误和溢出错误。  
通过 16 位存储器操作指令读取 SSRmn 寄存器。  
能用 SSRmnL 并且通过 8 位存储器操作指令读取 SSRmn 寄存器的低 8 位。  
在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

图 11-9 串行状态寄存器 mn（SSRmn）的格式 (1/2)

地址: F0100H、 F0101H （SSR00）、 F0102H、 F0103H （SSR01）										复位后: 0000H			R			
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn 注	PEF mn	OVF mn

TSF mn	通道 n 通信状态的表示标志
0	通信处于停止或者待机状态。
1	通信处于运行状态。
[ 清除条件 ]	
• 当将 STm 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信待机状态）时	
• 当通信结束时	
[ 置位条件 ]	
• 当开始通信时	

BFF mn	通道 n 缓冲寄存器的状态表示标志
0	SDRmn 寄存器没有保存有效数据。
1	SDRmn 寄存器保存了有效数据。
[ 清除条件 ]	
• 在发送过程中传送完从 SDRmn 寄存器到移位寄存器的发送数据时	
• 在接收过程中从 SDRmn 寄存器读完接收数据时	
• 当将 STm 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信允许状态）时	
[ 置位条件 ]	
• 在 SCRmn 寄存器的 TXEmn 位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给 SDRmn 寄存器写发送数据时	
• 在 SCRmn 寄存器的 RXEmn 位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到 SDRmn 寄存器时	
• 当发生接收错误时	

注 只限于 SSR01 寄存器。

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

备注 m: 单元号（m=0） n: 通道号（n=0、1）

图 11-9 串行状态寄存器 mn (SSRmn) 的格式 (2/2)

地址: F0100H、F0101H (SSR00)、F0102H、F0103H (SSR01) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn 注	PEF mn	OVF mn

FEF mn 注	通道 n 帧错误的检测标志
0	没有发生错误。
1	发生错误 (UART 接收时)。
[清除条件] • 当给 SIRmn 寄存器的 FECTmn 位写“1”时	
[置位条件] • 在 UART 接收结束时没有检测到停止位时	

PEF mn	通道 n 奇偶校验 /ACK 错误的检测标志
0	没有发生错误。
1	发生奇偶校验错误 (UART 接收时) 或者未检测到 ACK (I <sup>2</sup> C 发送时)。
[清除条件] • 当给 SIRmn 寄存器的 PECTmn 位写“1”时	
[置位条件] • 在 UART 接收结束时发送数据的奇偶校验和奇偶校验位不同 (奇偶校验错误) 时	

OVF mn	通道 n 溢出错误的检测标志
0	没有发生错误。
1	发生错误。
[清除条件] • 当给 SIRmn 寄存器的 OVCTmn 位写“1”时	
[置位条件] • 在 SCRmn 寄存器的 RXEmn 位为“1” (各通信模式中的接收模式、发送和接收模式) 的状态下, 虽然接收数据被保存在 SDRmn 寄存器, 但是没有读接收数据而写发送数据或者写下一个接收数据时 • 在 CSI 模式的从属发送或者从属发送和接收过程中未准备好发送数据时	

注 只限于 SSR01 寄存器。

注意 在 SNOOZE 模式 (SWCm=1) 中进行 CSI 接收时, OVFmn 标志不变。

备注 m: 单元号 (m=0) n: 通道号 (n=0、1)

11.3.8 串行通道开始寄存器 m（SSm）

SSm 寄存器是设定允许各通道的通信 / 开始计数的触发寄存器。

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

通过 16 位存储器操作指令设定 SSm 寄存器。

能用 SSmL 并且通过 1 位或者 8 位存储器操作指令设定 SSm 寄存器的低 8 位。

在产生复位信号后，SSm 寄存器的值变为“0000H”。

图 11-10 串行通道开始寄存器 m（SSm）的格式

地址: F0122H、F0123H (SS0)

复位后: 0000H

R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS01	SS00

SSmn	通道 n 运行开始的触发
0	没有触发。
1	将 SEmn 位置“1”，转移到通信待机状态注。

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

注意 1. 必须将 bit15 ~ 2 置“0”。

2. 在 UART 接收时，必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 fMCK 时钟，然后将 SSmn 置“1”。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0、1)

2. SSm 寄存器的读取值总是“0000H”。

11.3.9 串行通道停止寄存器 m（STm）

STm 寄存器是设定允许各通道的通信 / 停止计数的触发寄存器。

如果给各位（STmn）写“1”，就将串行通道允许状态寄存器 m（SEm）的对应位（SEmn）清“0”（运行停止状态）。因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

通过 16 位存储器操作指令设定 STm 寄存器。

能用 STmL 并且通过 1 位或者 8 位存储器操作指令设定 STm 寄存器的低 8 位。

在产生复位信号后，STm 寄存器的值变为“0000H”。

图 11-11 串行通道停止寄存器 m（STm）的格式

地址: F0124H、F0125H (ST0)

复位后: 0000H

W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST01	ST00

STmn	通道 n 运行停止的触发
0	没有触发。
1	将 SEmn 位清“0”，停止通信运行注。

注 控制寄存器和移位寄存器的值、SCKmn引脚和SOMn引脚以及FEFmn标志、PEFmn标志和OVFmn标志保持状态。

注意 必须将 bit15 ~ 2 置“0”。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0、1)  
2. STm 寄存器的读取值总是“0000H”。

11.3.10 串行通道允许状态寄存器 m（SEm）

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

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

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

对于停止运行的通道 n，能通过软件设定 SOm 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令读取 SEm 寄存器。

能用 SEmL 并且通过 1 位或者 8 位存储器操作指令读取 SEm 寄存器的低 8 位。

在产生复位信号后，SEm 寄存器的值变为“0000H”。

图 11-12 串行通道允许状态寄存器 m（SEm）的格式

地址: F0120H、F0121H (SE0)      复位后: 0000H      R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

SEmn	通道 n 运行的允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

备注    m: 单元号 (m=0)    n: 通道号 (n=0、1)

11.3.11 串行输出允许寄存器 m（SOEm）

SOEm 寄存器设定允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道 n，无法通过软件改写后述的串行输出寄存器 m（SOm）的 SOmn 位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道 n，能通过软件设定 SOm 寄存器的 SOmn 位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令设定 SOEm 寄存器。

能用 SOEmL 并且通过 1 位或者 8 位存储器操作指令设定 SOEm 寄存器的低 8 位。

在产生复位信号后，SOEm 寄存器的值变为“0000H”。

图 11-13 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	0	0	SOE00

SOE mn	通道 n 串行输出的允许或者停止
0	停止串行通信的输出。
1	允许串行通信的输出。

注意 必须将 bit15 ~ 1 置“0”。

备注 m: 单元号（m=0） n: 通道号（n=0）

11.3.12 串行输出寄存器 m（S0m）

S0m 寄存器是各通道串行输出的缓冲寄存器。

从通道 n 的串行数据输出引脚输出此寄存器的 S0mn 位的值。

从通道 n 的串行时钟输出引脚输出此寄存器的 CKOmn 位的值。

只有在禁止串行输出时（SOEmn=0）才能通过软件改写此寄存器的 S0mn 位。当允许串行输出（SOEmn=1）时，忽视通过软件的改写而只能通过串行通信更改此寄存器的 S0mn 位的值。

只有在停止通道运行时（SEmn=0）才能通过软件改写此寄存器的 CKOmn 位。当允许通道运行（SEmn=1）时，忽视通过软件的改写而只能通过串行通信更改此寄存器的 CKOmn 位的值。

要将串行接口引脚用作端口功能等非串行接口功能时，必须将相应的 CKOmn 位和 S0mn 位置“1”。

通过 16 位存储器操作指令设定 S0m 寄存器。

在产生复位信号后，S0m 寄存器的值变为“0303H”。

图 11-14 串行输出寄存器 m（S0m）的格式

地址: F0128H、F0129H (SO0)      复位后: 0303H      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	0	0	1	CKO00	0	0	0	0	0	0	1	SO00

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

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

注意 必须将 S00 寄存器的 bit15 ~ 10 和 bit7 ~ 2 置“0”并且将 S00 寄存器的 bit9 和 bit1 置“1”。

备注 m：单元号（m=0） n：通道号（n=0）



11.3.13 串行输出电平寄存器 m（SOLm）

SOLm 寄存器是设定各通道的数据输出电平反相的寄存器。  
只有在 UART 模式中才能设定此寄存器。在 CSI 模式中，必须将对应的位置“0”。

只在允许串行输出时（SOEmn=1），将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时（SOEmn=0），将 SOMn 位的值直接输出。  
禁止在运行过程中（SEmn=1）改写 SOLm 寄存器。  
通过 16 位存储器操作指令设定 SOLm 寄存器。  
能用 SOLmL 并且通过 8 位存储器操作指令设定 SOLm 寄存器的低 8 位。  
在产生复位信号后，SOLm 寄存器的值变为“0000H”。

图 11-15 串行输出电平寄存器 m（SOLm）的格式

地址: F0134H、F0135H (SOL0)

复位后: 0000H

R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL00

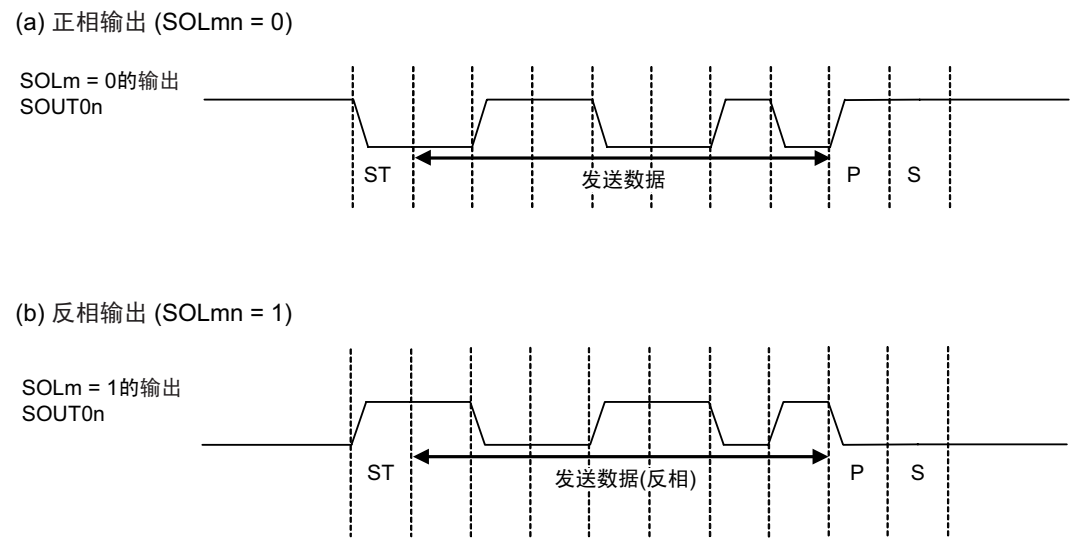
SOLmn	UART 模式中的通道 n 发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

注意 必须将 bit15 ~ 1 置“0”。

备注 m: 单元号（m=0） n: 通道号（n=0）

当进行 UART 发送时，发送数据的电平反相例子如图 11-16 所示。

图 11-16 发送数据的电平反相例子



备注 m: 单元号（m=0） n: 通道号（n=0、2）

### 11.3.14 串行待机控制寄存器 m (SSCm)

SSC0 寄存器是控制在 CSI00 或者 UART0 的串行数据接收时从 STOP 模式启动接收运行 (SNOOZE 模式) 的寄存器。

通过 16 位存储器操作指令设定 SSCm 寄存器。

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

在产生复位信号后, SSCm 寄存器的值变为“0000H”。

注意 SNOOZE 模式中的最大传送速率如下所示:

- CSI00: ~1Mbps
- UART0: 只限于 4800bps。

图 11-17 串行待机控制寄存器 m (SSCm) 的格式

地址: F0138H	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS ECm	SWC m

SS ECm	传送结束中断的允许或者停止的选择
0	允许产生错误中断 (INTSRE0)。
1	停止产生错误中断 (INTSRE0)。
<ul style="list-style-type: none"> <li>• 在 SNOOZE 模式中进行 UART 接收的情况下, 只有在 SWCm 位为“1”并且 EOCmn 位为“1”时才能将 SSECm 位置“1”或者“0”, 否则必须将 SSECm 位置“0”。</li> <li>• 禁止将 SSECm 位和 SWCm 位分别置“1”和“0”。</li> </ul>	

SWC m	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
<ul style="list-style-type: none"> <li>• 在 STOP 模式中, 通过硬件触发信号解除 STOP 模式, 并且在 CPU 不运行的状态下进行 CSI/UART 的接收 (SNOOZE 模式)。</li> <li>• 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟 (<math>f_{CLK}</math>) 时才能设定 SNOOZE 模式功能, 而在选择其他时钟的情况下禁止设定。</li> <li>• 即使使用 SNOOZE 模式, 也必须在通常运行模式中将 SWC 位置“0”并且在即将要转移到 STOP 模式前将 SWC 位改为“1”。</li> </ul> <p>另外, 必须在从 STOP 模式恢复到通常运行模式后将 SWC 位改为“0”。</p>	

注意 禁止将 SSECm 位和 SWCm 位分别置“1”和“0”。

图 11-18 在 SNOOZE 模式中进行 UART 接收时的中断

EOCmn 位	SSECm 位	正常接收	接收错误
0	0	产生 INTSRx。	产生 INTSRx。
0	1	产生 INTSRx。	产生 INTSRx。
1	0	产生 INTSRx。	产生 INTSREx。
1	1	产生 INTSRx。	不产生中断。

11.3.15 输入切换控制寄存器（ISC）

ISC 寄存器的 SSIE00 位在 CSI00 通信的从属模式中控制通道 0 的  $\overline{\text{SSI00}}$  引脚输入。在给  $\overline{\text{SSI00}}$  引脚输入高电平的期间，即使输入串行时钟，也不进行发送和接收而从 SO00 引脚输出高阻抗。

在给  $\overline{\text{SSI00}}$  引脚输入低电平的期间，如果输入串行时钟，就根据各模式的设定进行发送和接收。

通过 1 位或者 8 位存储器操作指令设定 ISC 寄存器。

在产生复位信号后，ISC 寄存器的值变为“00H”。

图 11-19 输入切换控制寄存器（ISC）的格式

地址: F0073H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	0	0

SSIE00	CSI00 通信的从属模式中通道 0 的 $\overline{\text{SSI00}}$ 输入的设置
0	$\overline{\text{SSI00}}$ 引脚输入无效。
1	$\overline{\text{SSI00}}$ 引脚输入有效。

注意 必须将 bit6 ~ 0 置“0”。

11.3.16 噪声滤波器允许寄存器 0（NFEN0）

NFEN0 寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 CSI 的引脚，必须将对应位置“0”，使噪声滤波器无效。

对于用于 UART 通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，通过 CPU/ 外围硬件时钟（f<sub>CLK</sub>）与 2 个时钟的一致检测进行同步；当噪声滤波器无效时，只通过 CPU/ 外围硬件时钟（f<sub>MCK</sub>）进行同步。

通过 1 位或者 8 位存储器操作指令设定 NFEN0 寄存器。

在产生复位信号后，NFEN0 寄存器的值变为“00H”。

图 11-20 噪声滤波器允许寄存器 0（NFEN0）的格式

地址：F0070H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	RxD0 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON
当用作 RxD0 引脚时，必须将 SNFEN00 位置“1”。	
当用作 RxD0 引脚以外的功能时，必须将 SNFEN00 位置“0”。	

注意 必须将 bit7 ~ 1 置“0”。

### 11.3.17 端口模式寄存器 3 (PM3)

这是以位为单位设定端口 3 的输入 / 输出的寄存器。

在将串行数据输出引脚或者串行时钟输出引脚的复用端口 (P30/INTP2/TxD0/TOOLTxD/SO0 等) 用作串行数据输出或者串行时钟输出时, 必须将各端口对应的端口模式寄存器 (PMxx) 的位置 “0”, 并且将端口寄存器 (Pxx) 的位置 “1”。

例) P30/INTP2/TxD0/TOOLTxD/SO0 用作串行数据输出的情况

将端口模式寄存器 3 的 PM30 位置 “0”。

将端口寄存器 3 的 P30 位置 “1”。

在将串行数据输入引脚或者串行时钟输入引脚的复用端口 (P30/INTP2/TxD0/TOOLTxD/SO0 等) 用作串行数据输入或者串行时钟输入时, 必须将各端口对应的端口模式寄存器 (PMxx) 的位置 “1”。此时, 端口寄存器 (Pxx) 的位可以是 “0” 或者 “1”。

例) P30/INTP2/TxD0/TOOLTxD/SO0 用作串行数据输入的情况

将端口模式寄存器 3 的 PM30 位置 “1”。

将端口寄存器 3 的 P30 位置 “0” 或者 “1”。

通过 1 位或者 8 位存储器操作指令设定 PM3 寄存器。

在产生复位信号后, PM3 寄存器的值变为 “FFH”。

有关各产品是否有 PMxx 寄存器, 请参照表 4-11。

图 11-21 端口模式寄存器 3 (PM3) 的格式

地址: FFF23H	复位后: FFH		R/W					
符号	7	6	5	4	3	2	1	0
PM3	1	1	PM35 注	PM34 注	PM33	PM32	PM31	PM30

PM3n	P3n 引脚的输入 / 输出模式的选择 (n=0 ~ 5)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

注 24 引脚产品没有此位。

注意 必须将 bit7 和 bit6 置 “1”。

### 11.4 运行停止模式

串行阵列单元的各串行接口有运行停止模式。  
在运行停止模式中不能进行串行通信，因此能降低功耗。  
另外，在运行停止模式中能将用于串行接口的引脚用作端口功能。

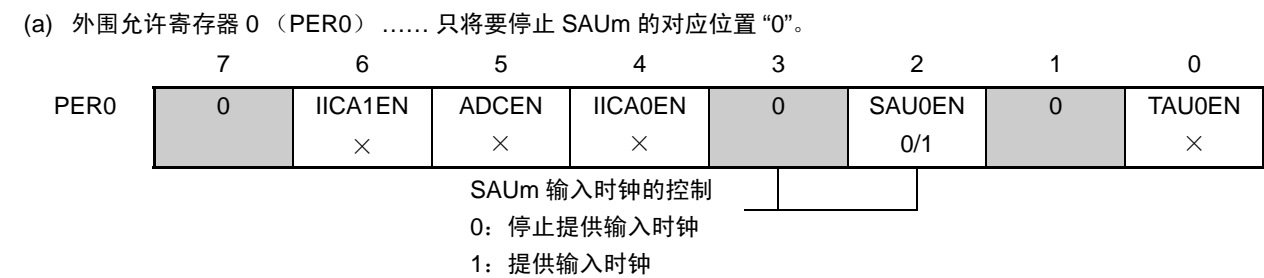
#### 11.4.1 以单元为单位停止运行的情况

通过外围允许寄存器 0（PER0）设定以单元为单位的停止运行。

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

要停止串行阵列单元 0 时，必须将 bit2（SAU0EN）置“0”。

图 11-22 以单元为单位停止运行时的外围允许寄存器 0（PER0）的设定



注意 1. 当 SAUmEN 位为“0”时，忽视串行阵列单元 m 的控制寄存器的写操作，而且读取值都为初始值。

但是，以下的寄存器除外：

- 输入切换控制寄存器（ISC）
- 噪声滤波器允许寄存器 0（NFEN0）
- 端口模式寄存器 3（PM3）
- 端口寄存器 3（P3）

2. 必须将 bit7、3、1 置“0”。

备注    ×   ：这是串行阵列单元未使用的位（取决于其他外围功能的设定）。

          ■   ：不能设定（设定初始值）。

          0/1   ：根据用户的用途置“0”或者“1”。

## 11.4.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

图 11-23 按通道停止运行时的各寄存器的设定

(a) 串行通道停止寄存器 m (STm) ..... 这是设定允许各通道的通信 / 停止计数的寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1 0/1	STm0 0/1

1: 将 SEmn 位清“0”并且停止通信运行

※ 因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

(b) 串行通道允许状态寄存器 m (SEm) ..... 此寄存器表示各通道的数据发送和接收的运行或者停止状态。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1 0/1	SEm0 0/1

0: 停止运行

※SEm 寄存器是只读状态寄存器，通过 STm 寄存器来停止运行。

对于已经停止运行的通道，能通过软件设定 SOm 寄存器的 CKOmn 位的值。

(c) 串行输出允许寄存器 m (SOEm) ..... 这是设定允许或者停止各通道串行通信输出的寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm0 0/1

0: 通过串行通信运行来停止输出

※ 对于已经停止串行输出的通道，能通过软件设定 SOm 寄存器的 SOmn 位的值。

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	0	CKOm0 0/1	0	0	0	0	0	0	0	SOm0 0/1

1: 串行时钟的输出值为“1”

1: 串行数据的输出值为“1”

※ 当将各通道对应的引脚用作端口功能时，必须将相应的 CKOmn 位和 SOmn 位置“1”。

(e) 输入切换控制寄存器 (ISC) ..... 这是 CSI00 从属通道 (单元 0 的通道 0) 的 SSI00 引脚的控制。

	7	6	5	4	3	2	1	0
ISC	SSI00 0/1	0	0	0	0	0	0	0

0: SSI00 引脚的输入值无效

备注 1. m: 单元号 (m=0) n: 通道号 (n=0、1)

2.   : 不能设定 (设定初始值)。 0/1: 根据用户的用途置“0”或者“1”。

### 11.5 3 线串行 I/O（CSI00）通信的运行

这是通过串行时钟（SCK）和串行数据（SI 和 SO）共 3 条线实现的时钟同步通信功能。

[ 数据的发送和接收 ]

- 7 位或者 8 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[ 时钟控制 ]

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

主控通信:  $\text{Max. } f_{\text{MCK}}/2$

从属通信:  $\text{Max. } f_{\text{MCK}}/6$

[ 中断功能 ]

- 传送结束中断、缓冲器空中断

[ 错误检测标志 ]

- 溢出错误

CSI00 支持 SNOOZE 模式。SNOOZE 模式功能是指：如果在 STOP 模式的状态下检测到 SCK 的输入，就不需要 CPU 运行而接收数据。

注 必须在满足 SCK 周期时间（ $t_{\text{KCY}}$ ）特性的范围内使用（参照“第 27 章 电特性”）。

SAU0 的通道 0 是支持 3 线串行 I/O（CSI00）的通道。

单元	通道	用作 CSI	用作 UART
0	0	CSI00（支持从属选择输入功能）	UART0
	1	—	

3 线串行 I/O（CSI00）有以下 7 种通信运行：

- 主控发送（参照 11.5.1）
- 主控接收（参照 11.5.2）
- 主控发送和接收（参照 11.5.3）
- 从属发送（参照 11.5.4）
- 从属接收（参照 11.5.5）
- 从属发送和接收（参照 11.5.6）
- SNOOZE 模式功能（参照 11.5.7）



## 11.5.1 主控发送

主控发送是指 R7F0C010 输出传送时钟并且将数据发送到其他设备的运行。

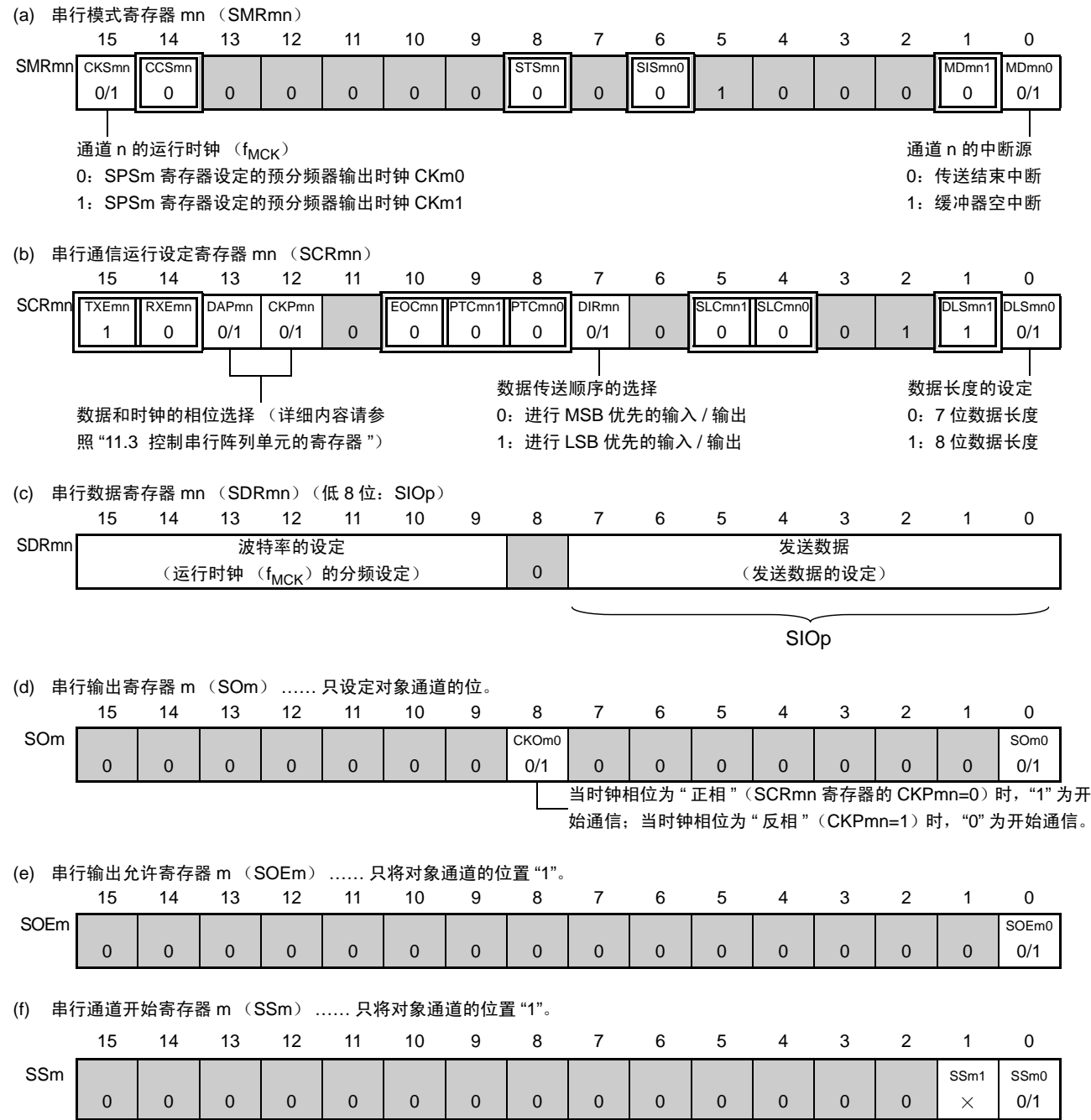
3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SO00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	无
传送数据长度	7 位或者 8 位
传送速率注	Max. $f_{MCK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] $f_{CLK}$ : 系统时钟频率
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>DAPmn=0: 在串行时钟开始运行时, 开始数据输出。</li> <li>DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>CKPmn=0: 正相</li> <li>CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 m: 单元号 (m=0) n: 通道号 (n=0) mn=00

(1) 寄存器的设定

图 11-24 3 线串行 I/O（CSI00）主控发送时的寄存器设定内容例子



备注 1. m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

2. □: 在 CSI 主控发送模式中为固定设定。■: 不能设定（设定初始值）。  
×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-25 主控发送的初始设定步骤

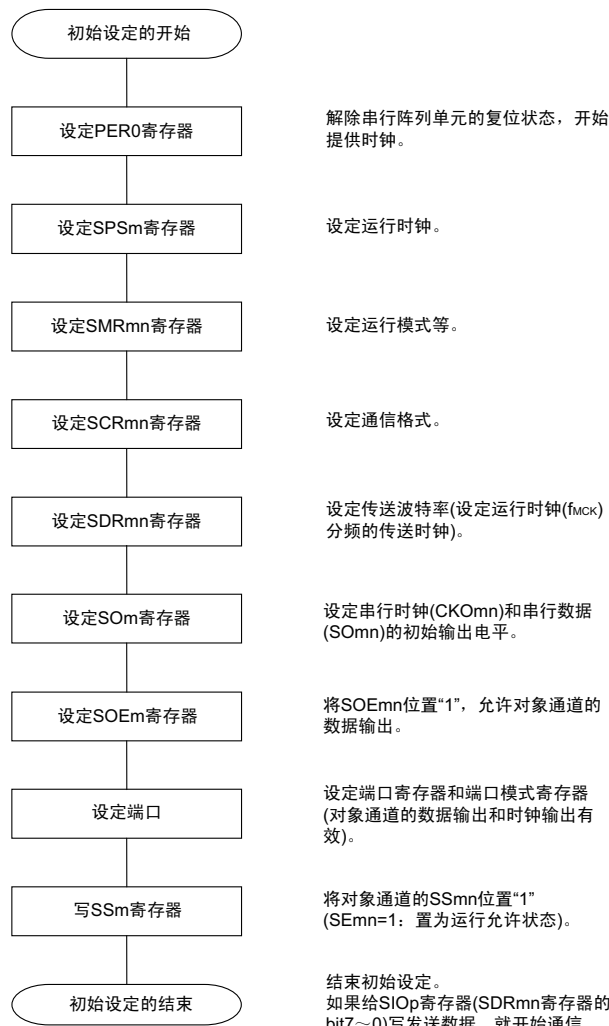


图 11-26 主控发送的中止步骤

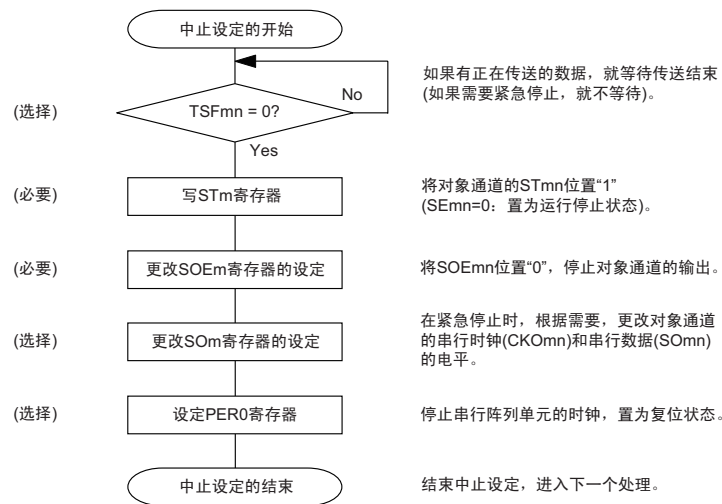


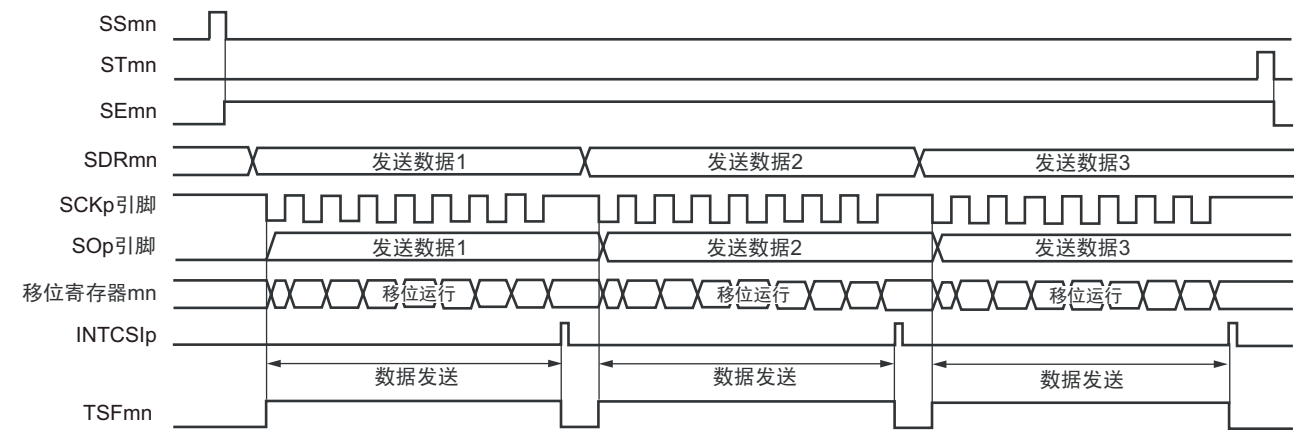
图 11-27 重新开始主控发送的设定步骤



**备注** 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

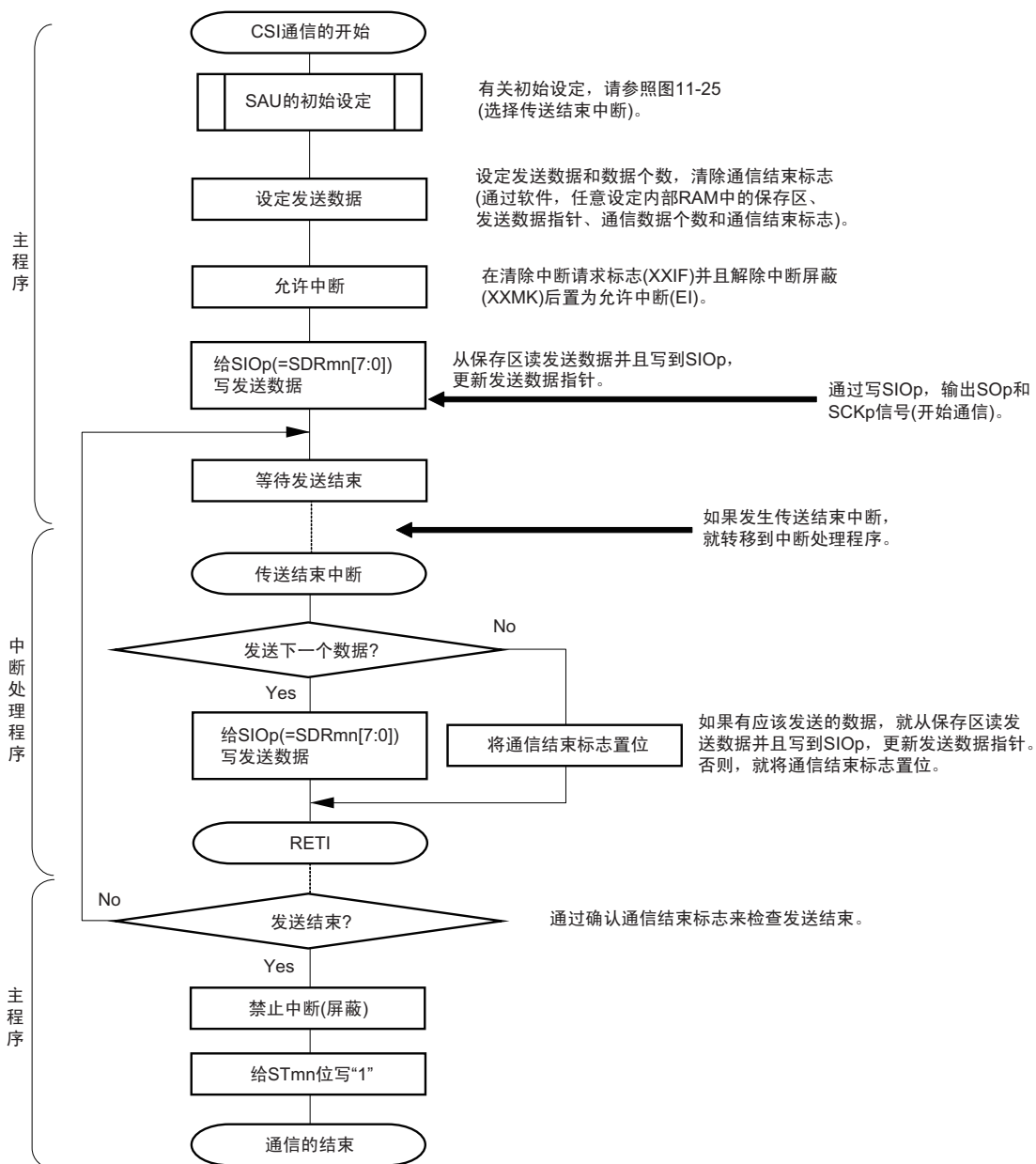
(3) 处理流程（单次发送模式）

图 11-28 主控发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



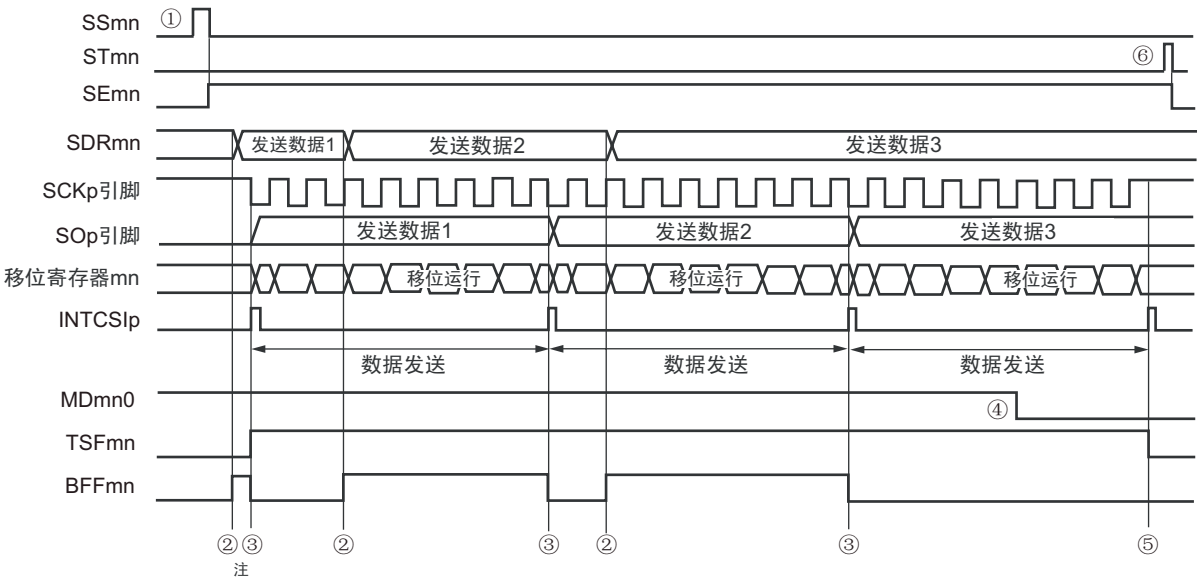
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-29 主控发送（单次发送模式）的流程图



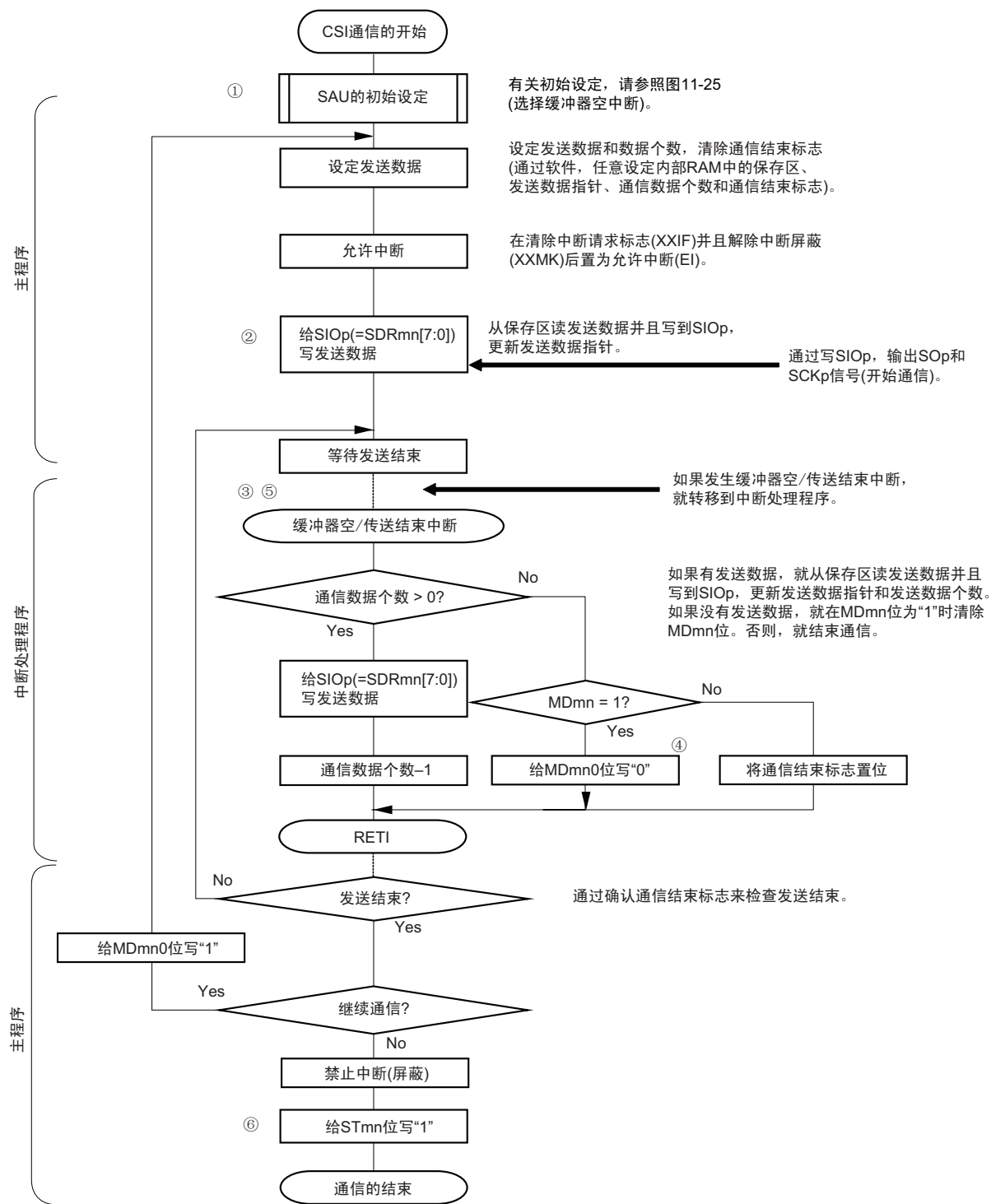
(4) 处理流程（连续发送模式）

图 11-30 主控发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



- 注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
- 注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。
- 备注 m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

图 11-31 主控发送（连续发送模式）的流程图



备注 图中的①～⑥对应“图 11-30 主控发送（连续发送模式）的时序图”中的①～⑥。



### 11.5.2 主控接收

主控接收是指 R7F0C010 输出传送时钟并且从其他设备接收数据的运行。

3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率注	Max. $f_{MCK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] $f_{CLK}$ : 系统时钟频率
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输入。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

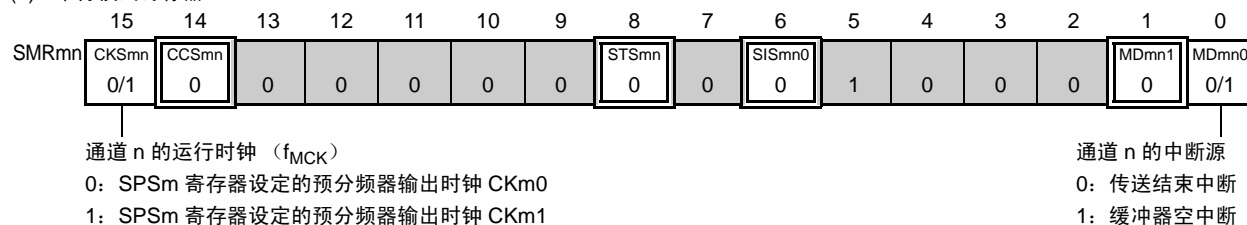
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 m: 单元号 (m=0) n: 通道号 (n=0) mn=00

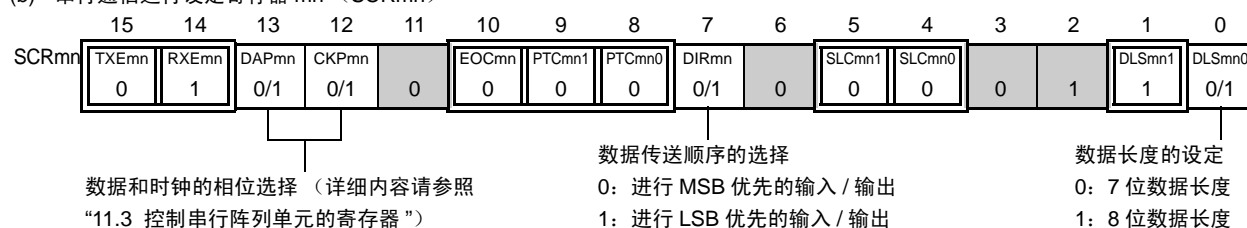
## (1) 寄存器的设定

图 11-32 3 线串行 I/O (CSI00) 主控接收时的寄存器设定内容例子

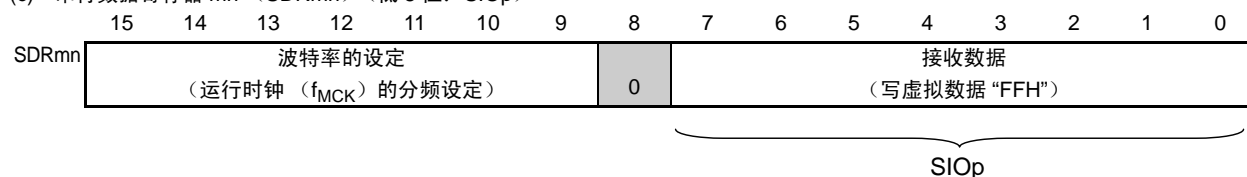
## (a) 串行模式寄存器 mn (SMRmn)



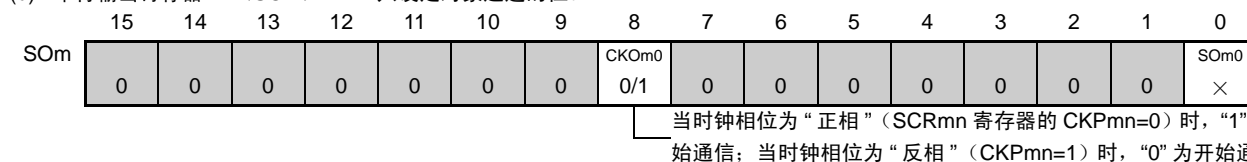
## (b) 串行通信运行设定寄存器 mn (SCRmn)



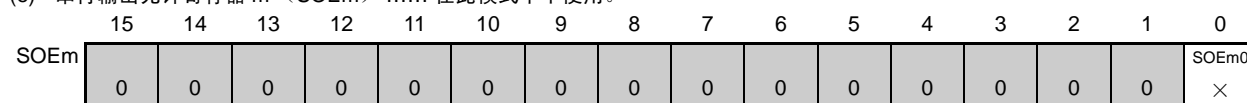
## (c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



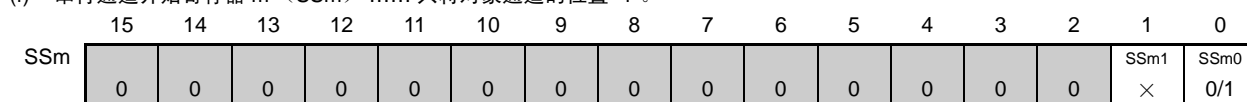
## (d) 串行输出寄存器 m (SOM) ..... 只设定对象通道的位。



## (e) 串行输出允许寄存器 m (SOEm) ..... 在此模式中不使用。



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



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

2.   : 在 CSI 主控接收模式中为固定设定。   : 不能设定 (设定初始值)。

×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-33 主控接收的初始设定步骤

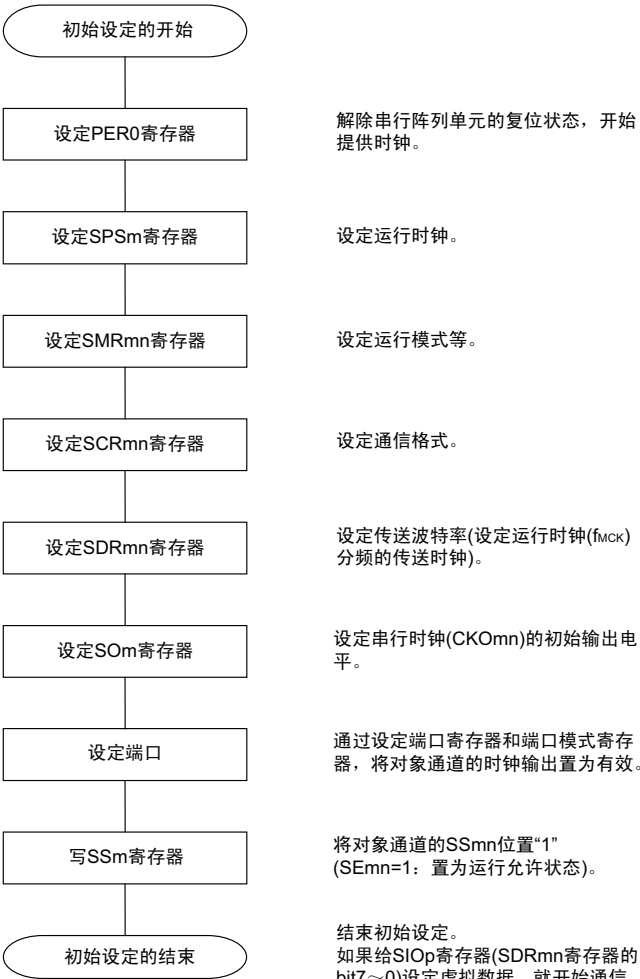


图 11-34 主控接收的中止步骤

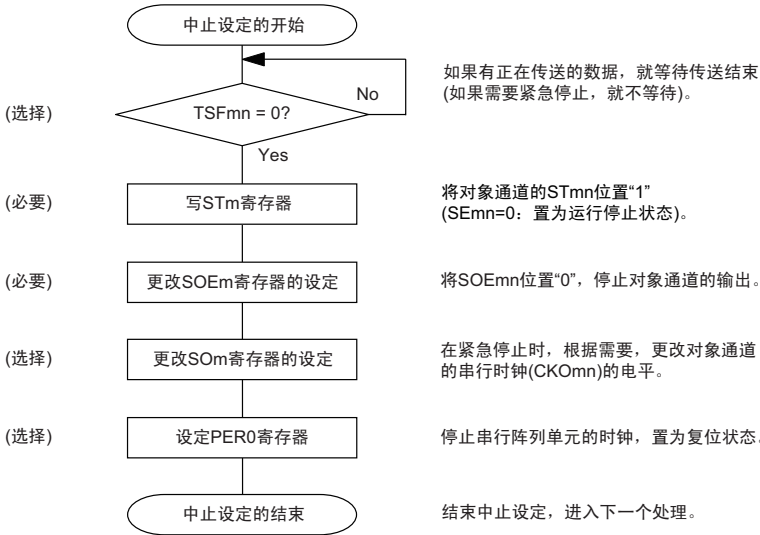
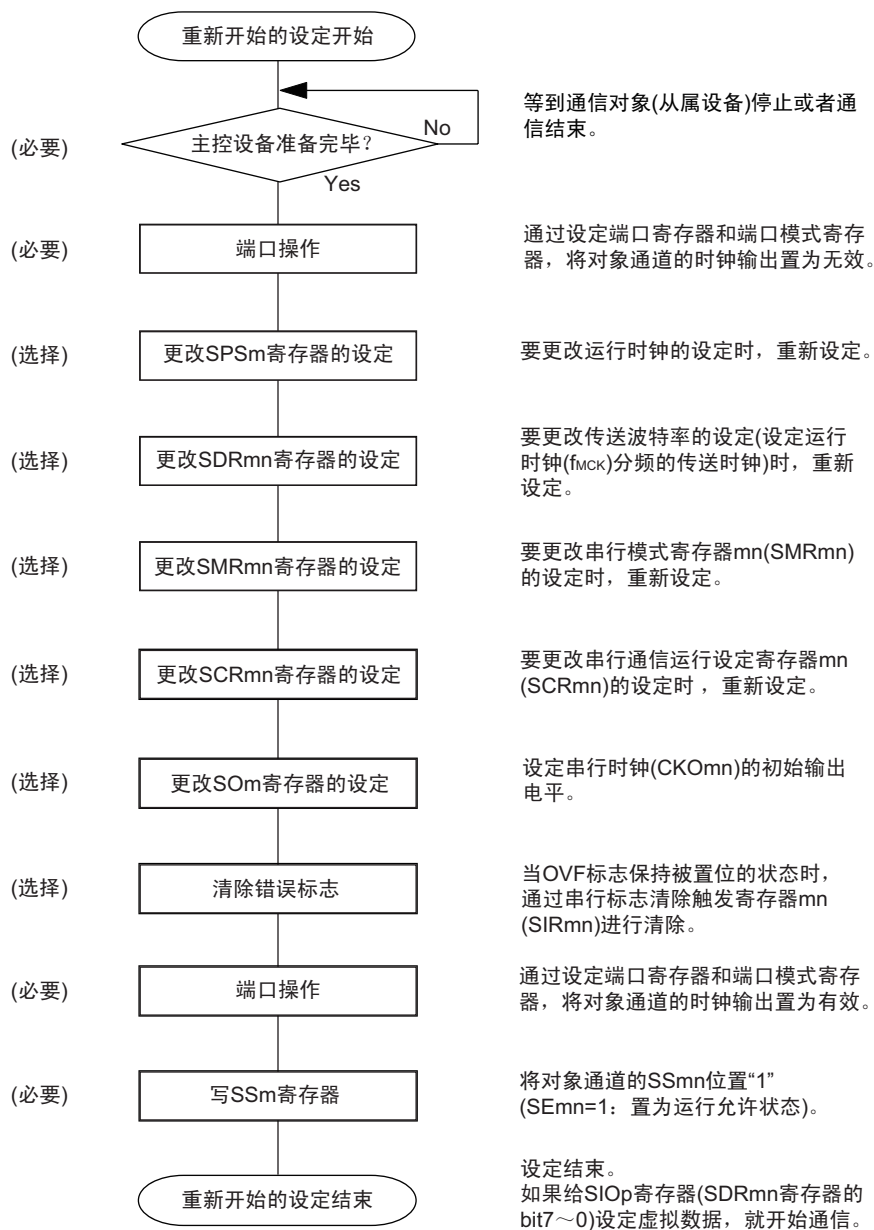


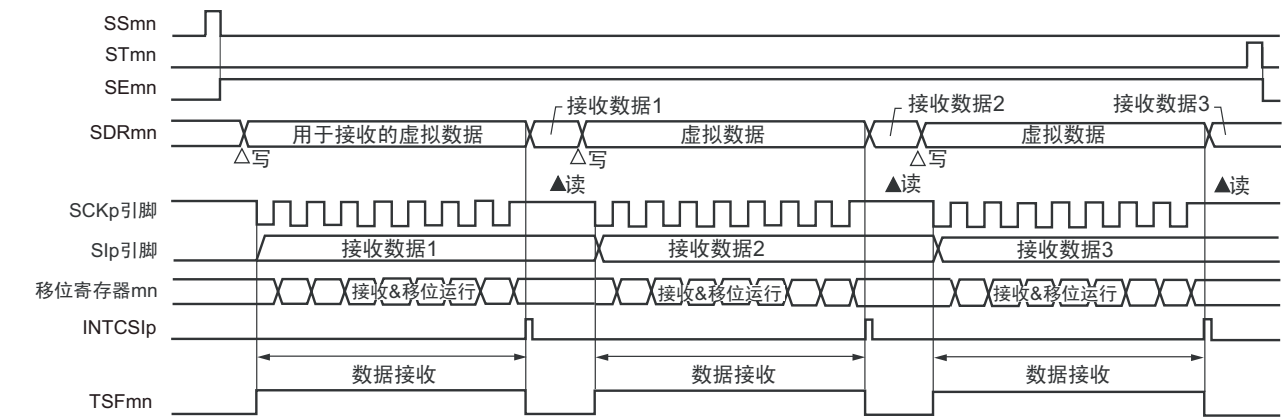
图 11-35 重新开始主控接收的设定步骤



**备注** 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（从属设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

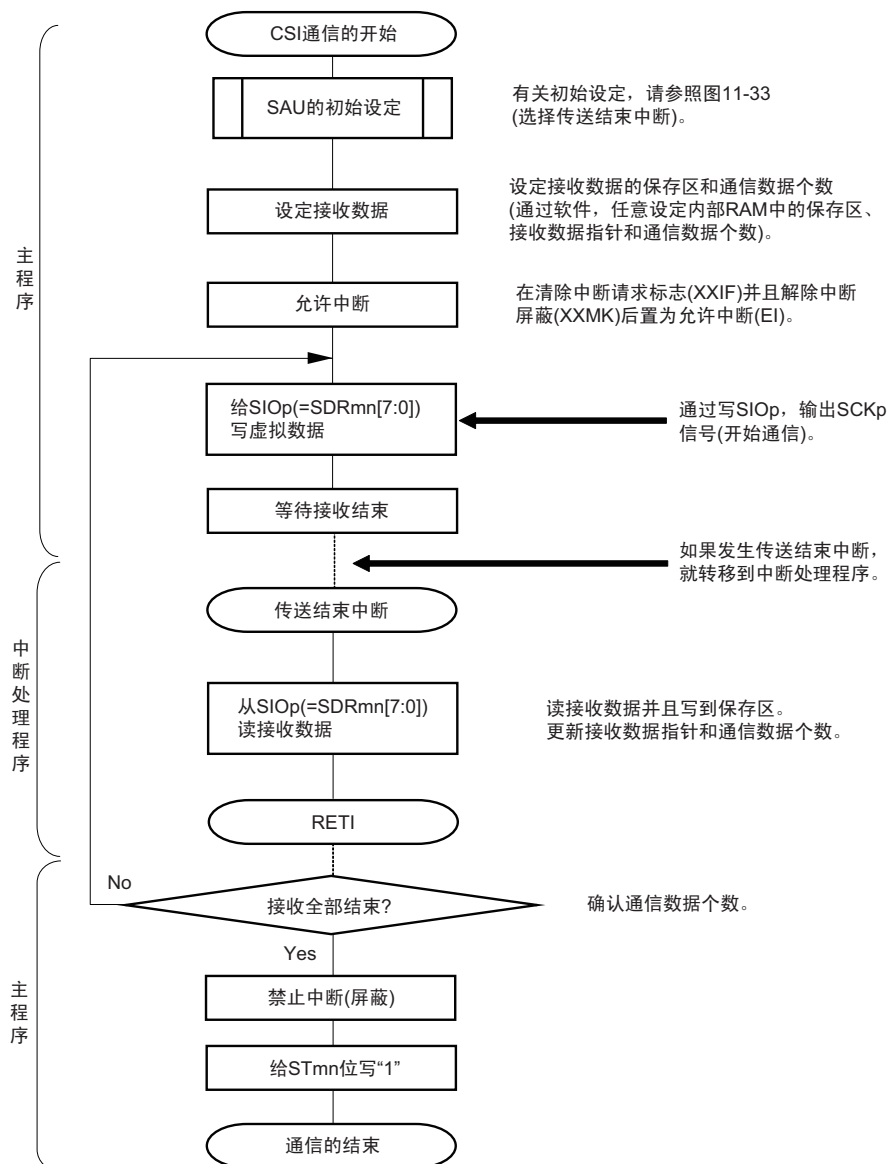
(3) 处理流程（单次接收模式）

图 11-36 主控接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



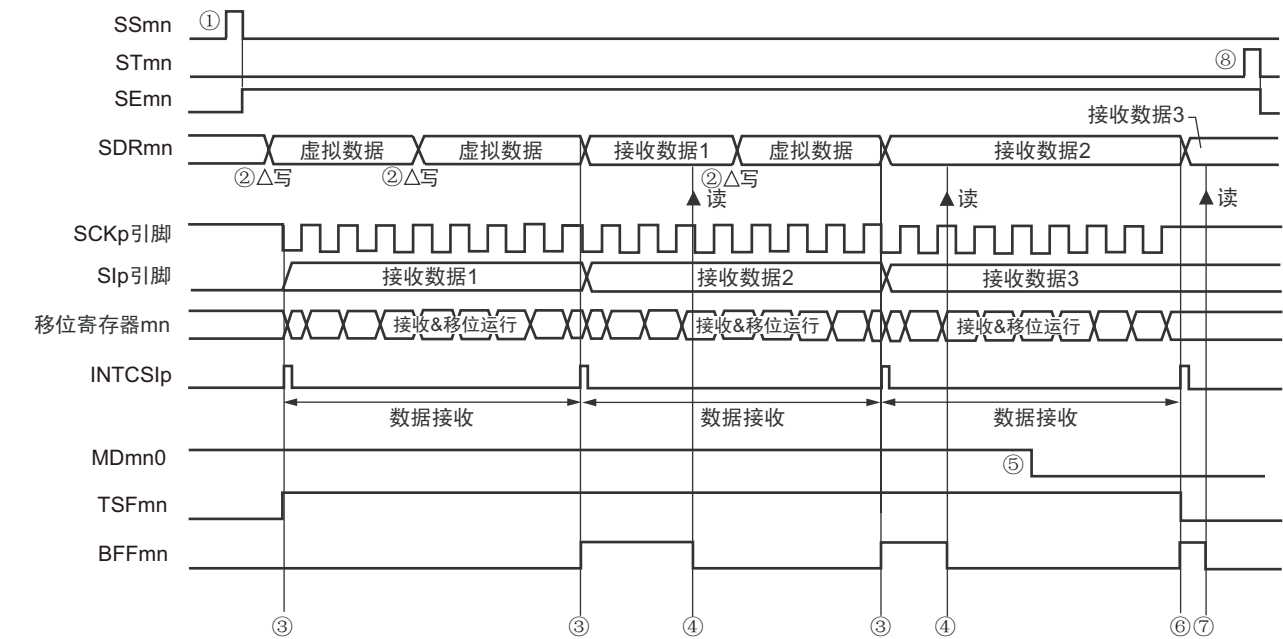
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-37 主控接收（单次接收模式）的流程图



(4) 处理流程（连续接收模式）

图 11-38 主控接收（连续接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

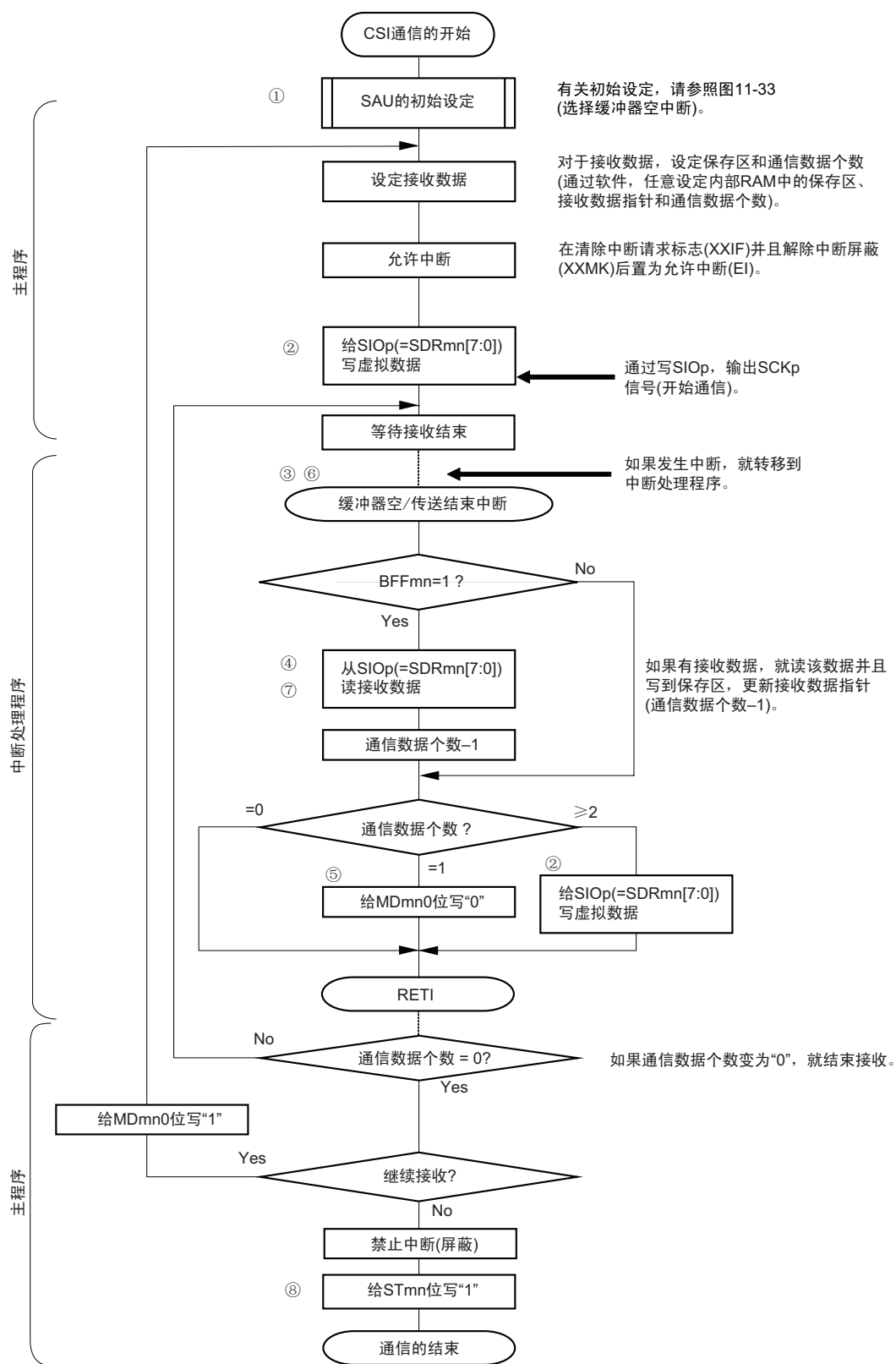


注意 即使在运行过程中也能改写 MDmn0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注 1. 图中的①～⑧对应“图 11-39 主控接收（连续接收模式）的流程图”中的①～⑧。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-39 主控接收（连续接收模式）的流程图



备注 图中的①~⑧对应“图 11-38 主控接收（连续接收模式）的时序图”中的①~⑧。



### 11.5.3 主控的发送和接收

主控的发送和接收是指 R7F0C010 输出传送时钟并且和其他设备进行数据发送和接收的运行。

3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、SO00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率注	Max. $f_{MCK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] $f_{CLK}$ : 系统时钟频率
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>DAPmn=0: 在串行时钟开始运行时，开始数据输入 / 输出。</li> <li>DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入 / 输出。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>CKPmn=0: 正相</li> <li>CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

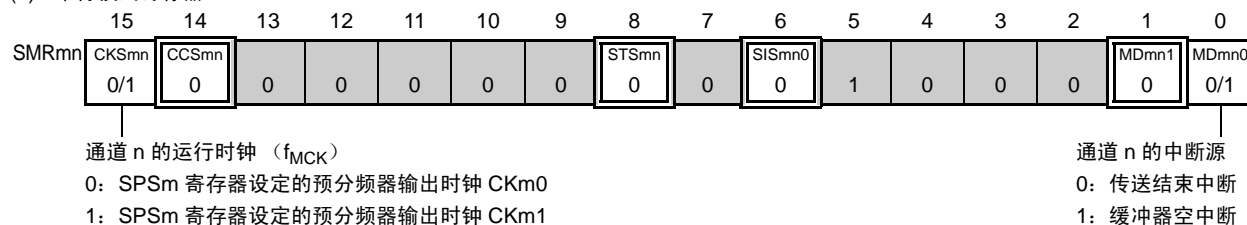
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

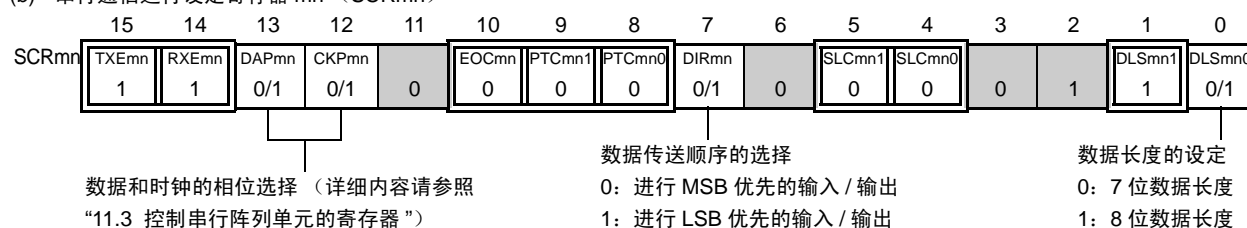
## (1) 寄存器的设定

图 11-40 3 线串行 I/O (CSI00) 主控发送和接收时的寄存器设定内容例子

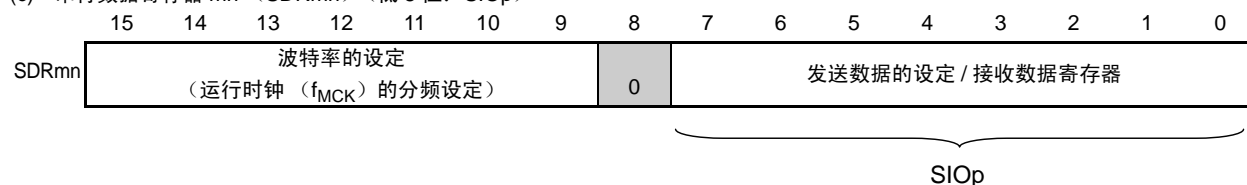
## (a) 串行模式寄存器 mn (SMRmn)



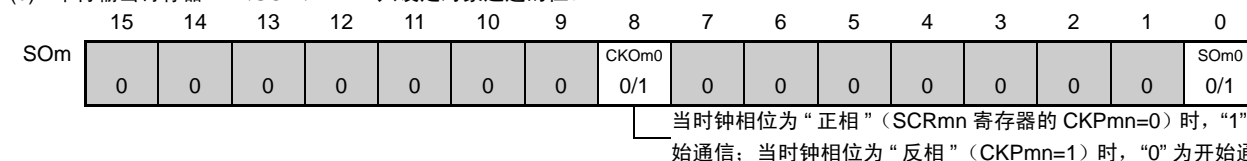
## (b) 串行通信运行设定寄存器 mn (SCRmn)



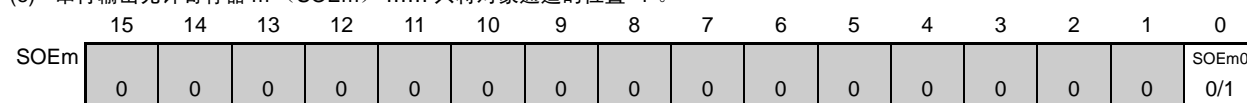
## (c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



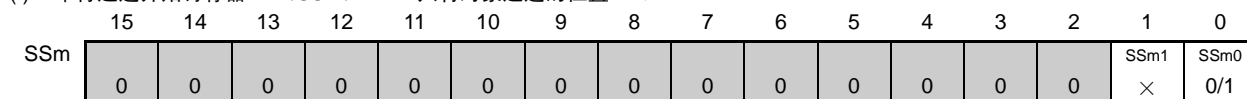
## (d) 串行输出寄存器 m (SOM) ..... 只设定对象通道的位。



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



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



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

2.   : 在 CSI 主控发送和接收模式中为固定设定。  : 不能设定 (设定初始值)。  
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-41 主控发送和接收的初始设定步骤

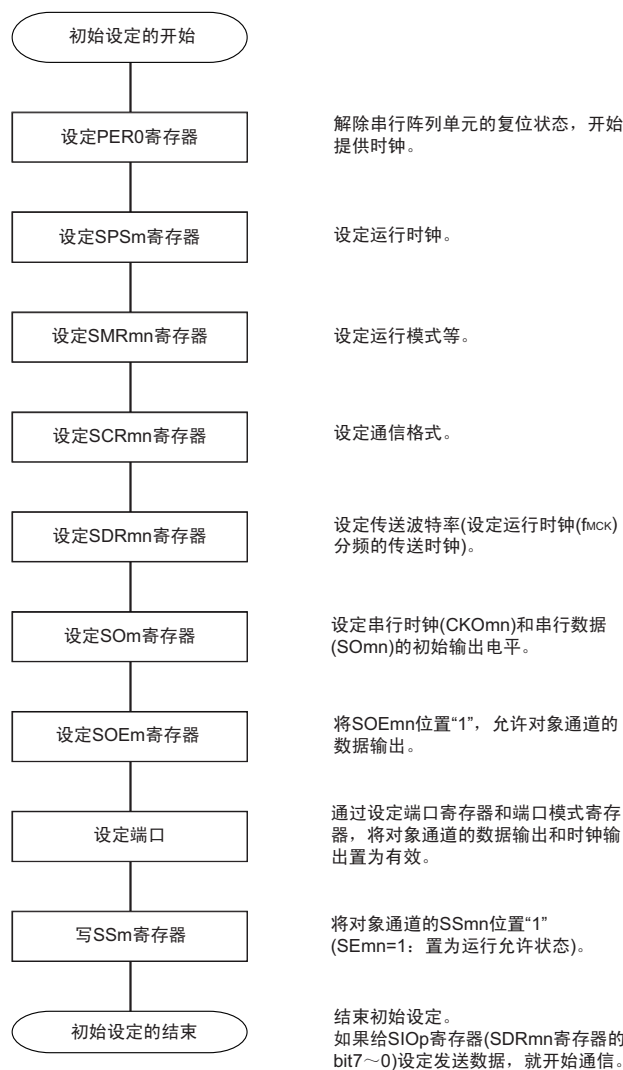


图 11-42 主控发送和接收的中止步骤

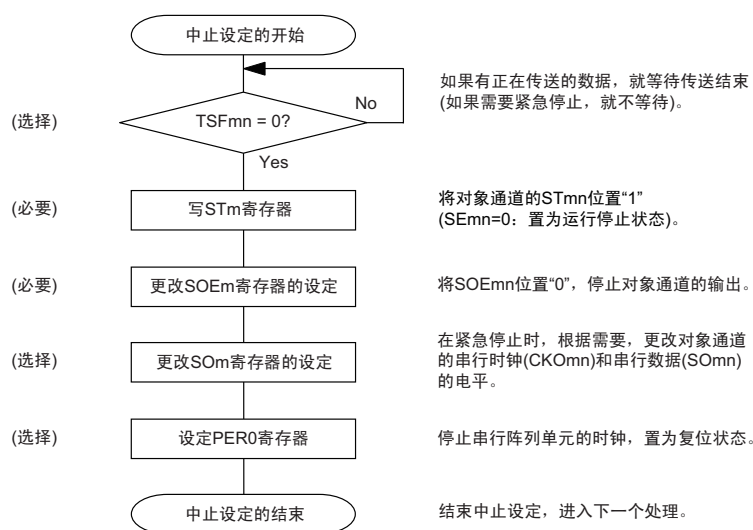
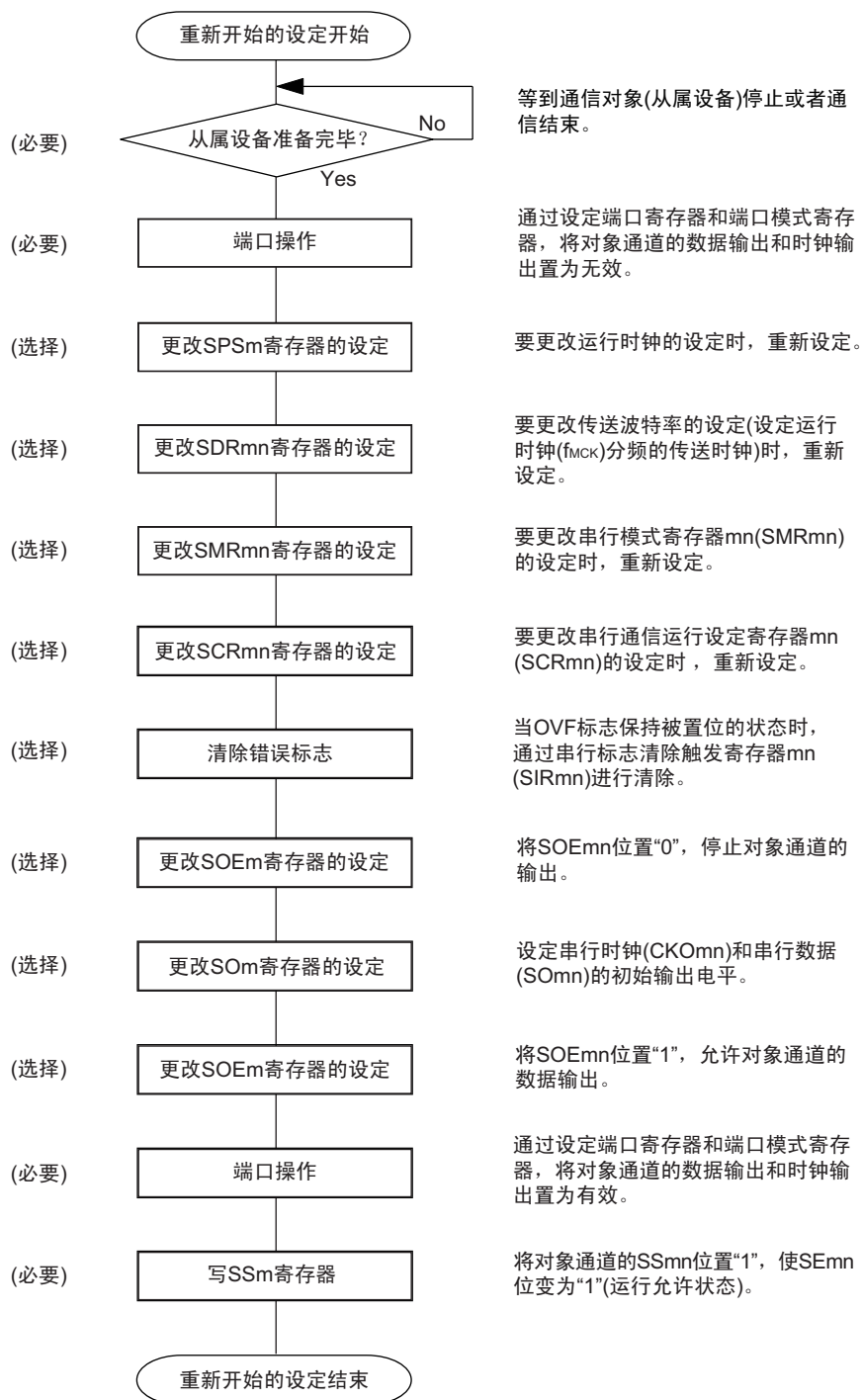
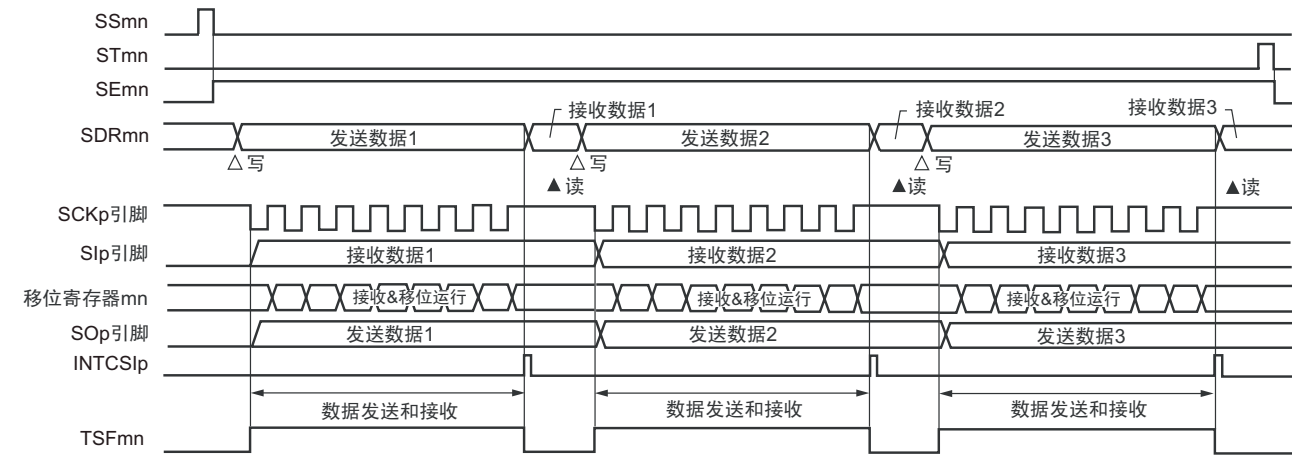


图 11-43 重新开始主控发送和接收的设定步骤



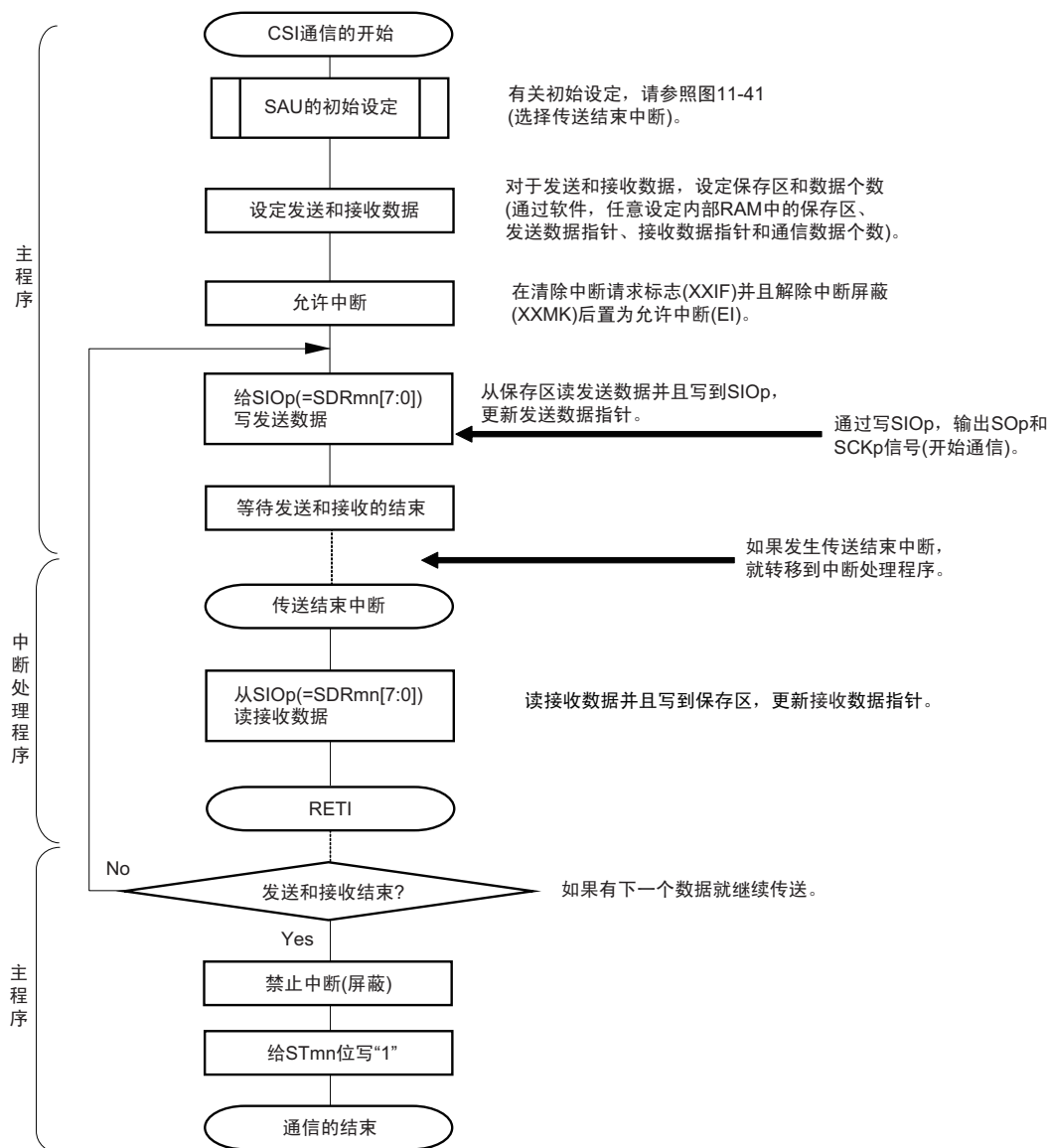
(3) 处理流程（单次发送和接收模式）

图 11-44 主控发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



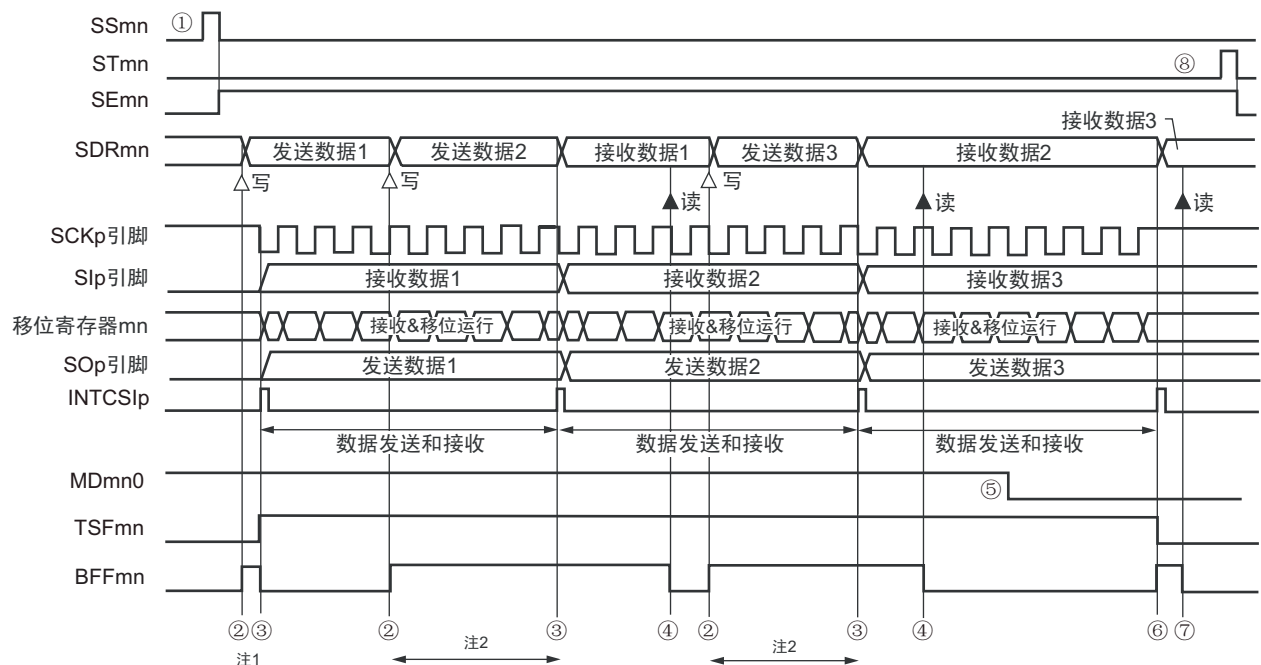
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-45 主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图 11-46 主控发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



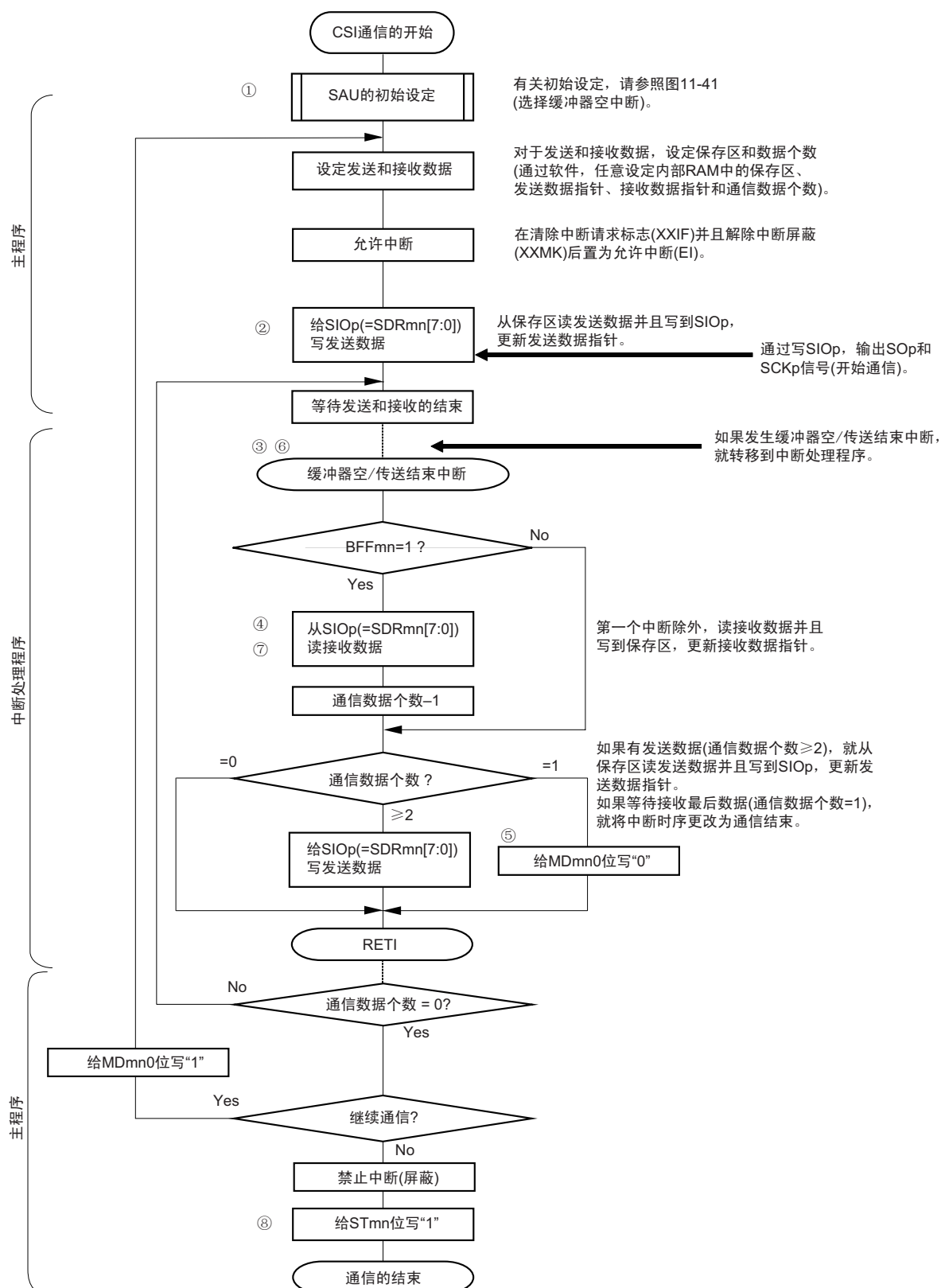
- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注 1. 图中的①～⑧对应“图 11-47 主控发送和接收（连续发送和接收模式）的流程图”中的①～⑧。

2. m：单元号（m=0） n：通道号（n=0） p：CSI 号（p=00） mn=00

图 11-47 主控发送和接收（连续发送和接收模式）的流程图



备注 图中的①~⑧对应“图 11-46 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。



## 11.5.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下 R7F0C010 将数据发送到其他设备的运行。

3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SO00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输出。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

- 注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{MCK}/6[\text{Hz}]$ 。  
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

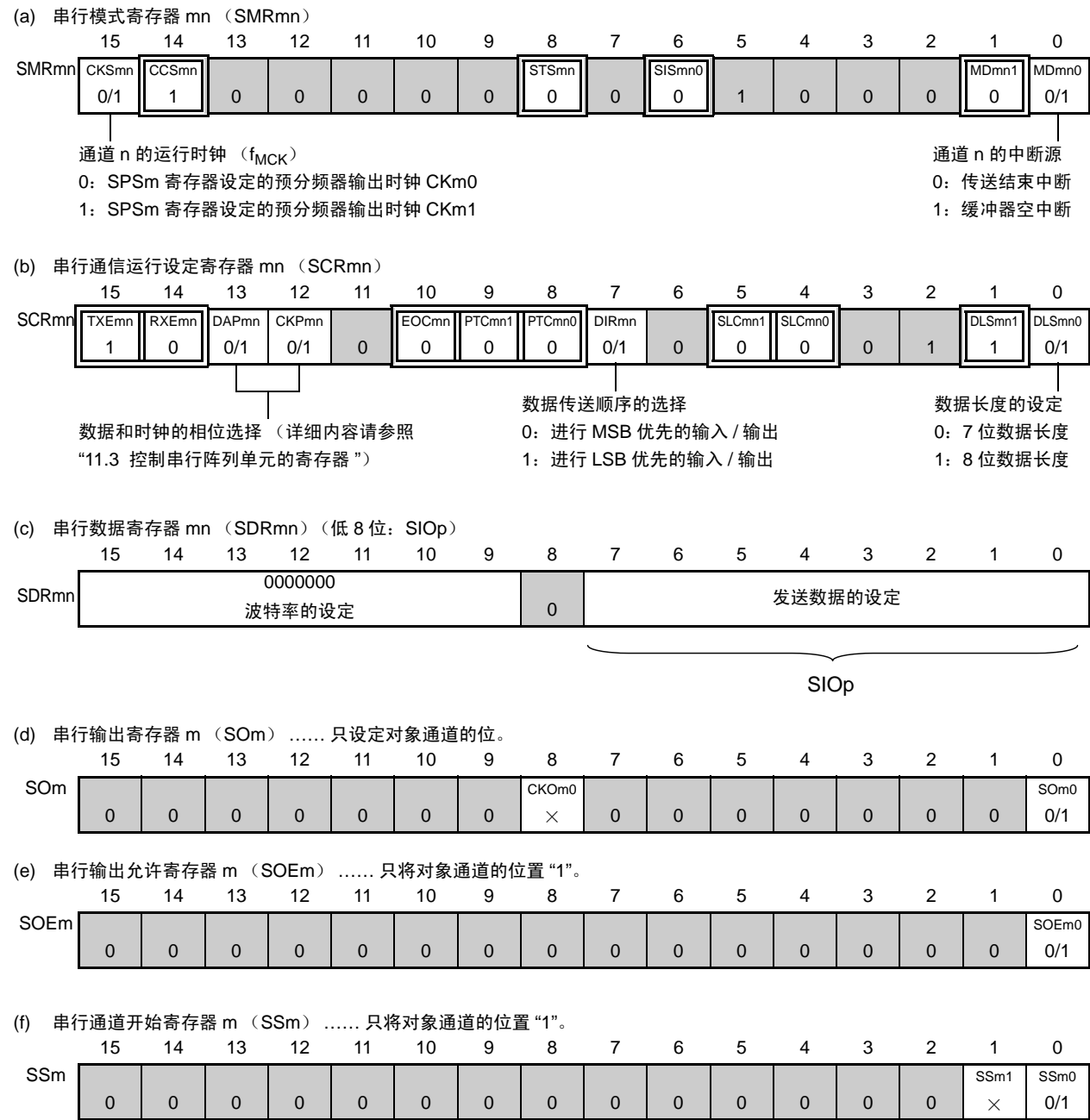
备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

$f_{SCK}$ : 串行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0) mn=00

(1) 寄存器的设定

图 11-48 3 线串行 I/O（CSI00）从属发送时的寄存器设定内容例子



备注 1. m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

2. □: 在 CSI 从属发送模式中为固定设定。■: 不能设定（设定初始值）。  
×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-49 从属发送的初始设定步骤

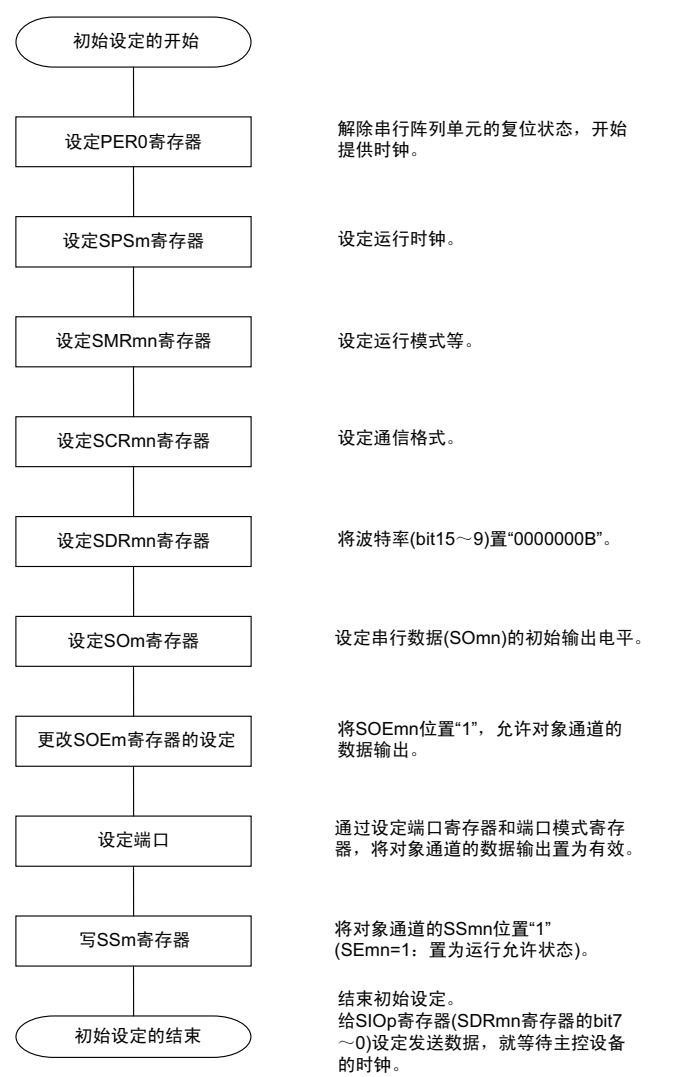


图 11-50 从属发送的中止步骤

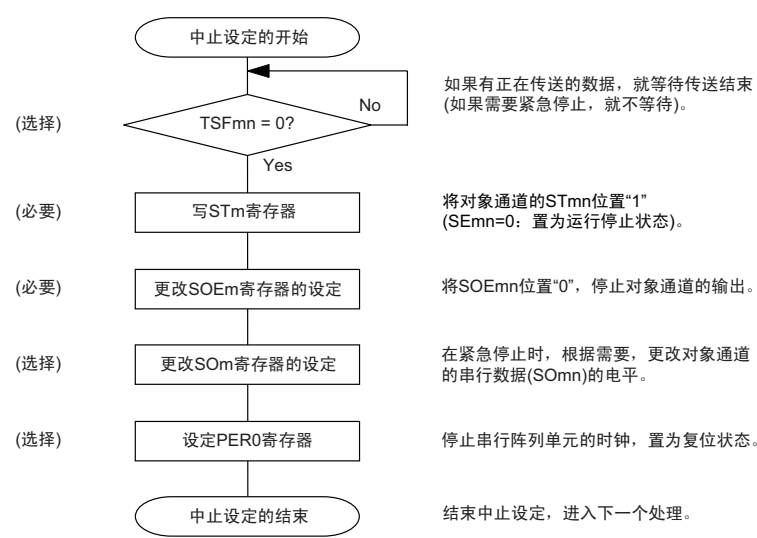
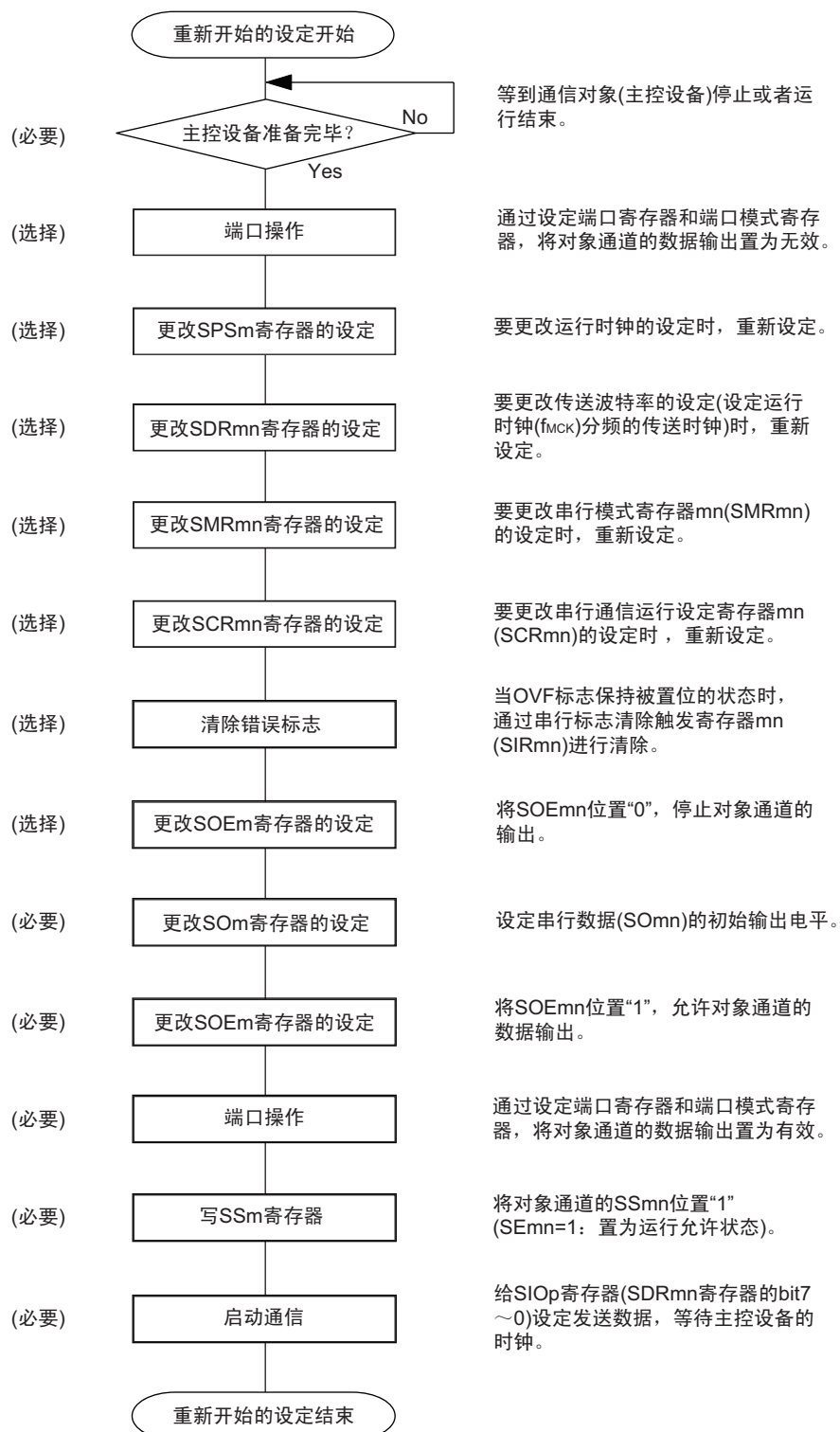


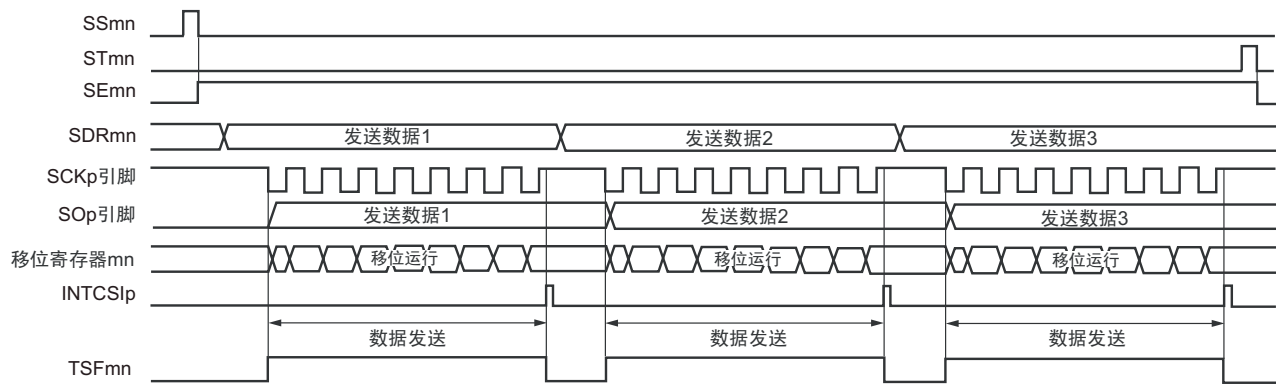
图 11-51 重新开始从属发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

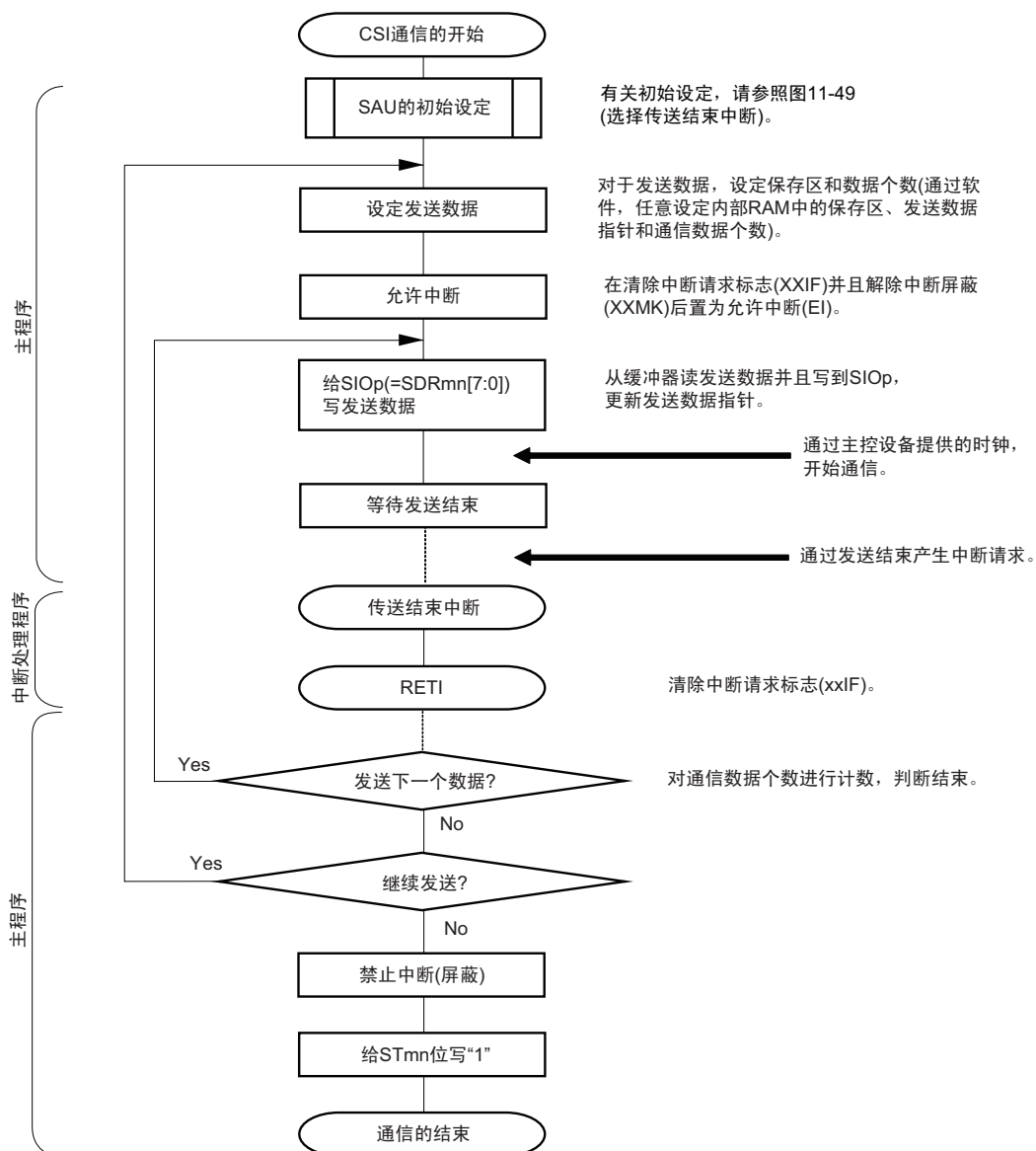
(3) 处理流程（单次发送模式）

图 11-52 从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



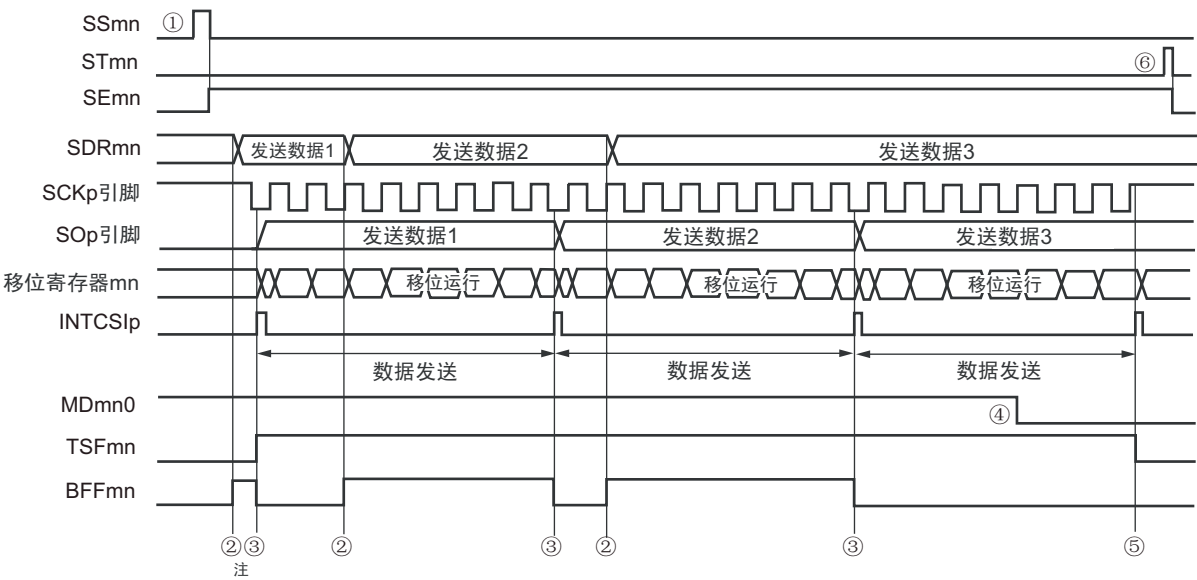
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-53 从属发送（单次发送模式）的流程图



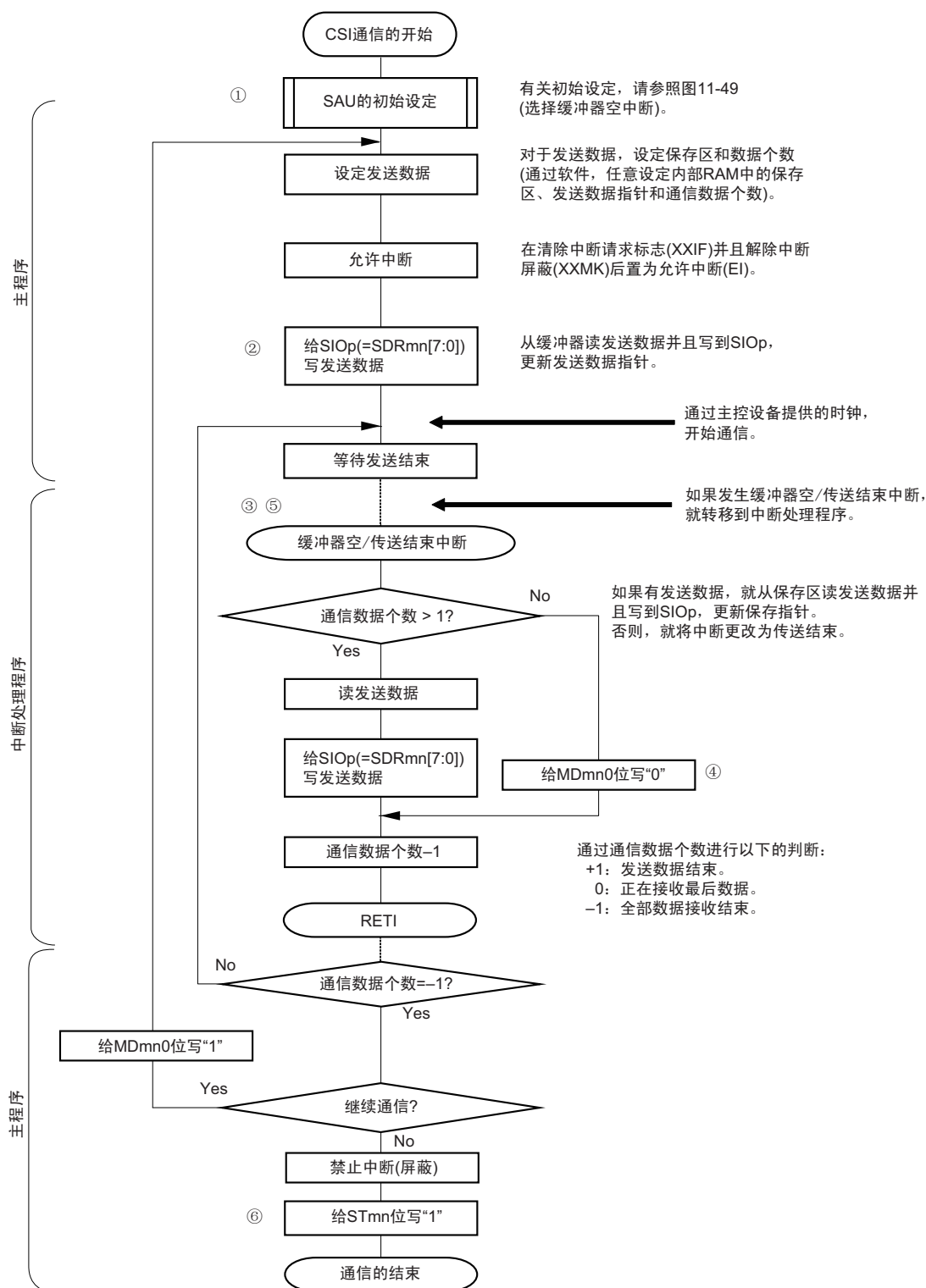
(4) 处理流程（连续发送模式）

图 11-54 从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



- 注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
- 注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。
- 备注 m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

图 11-55 从属发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 11-54 从属发送（连续发送模式）的时序图”中的①~⑥。



## 11.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下 R7F0C010 从其他设备接收数据的运行。

3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00
中断	INTCSI00 只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输入。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入。</li> </ul>
时钟相位	能通过 SCRMn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

- 注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{MCK}/6[\text{Hz}]$ 。  
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

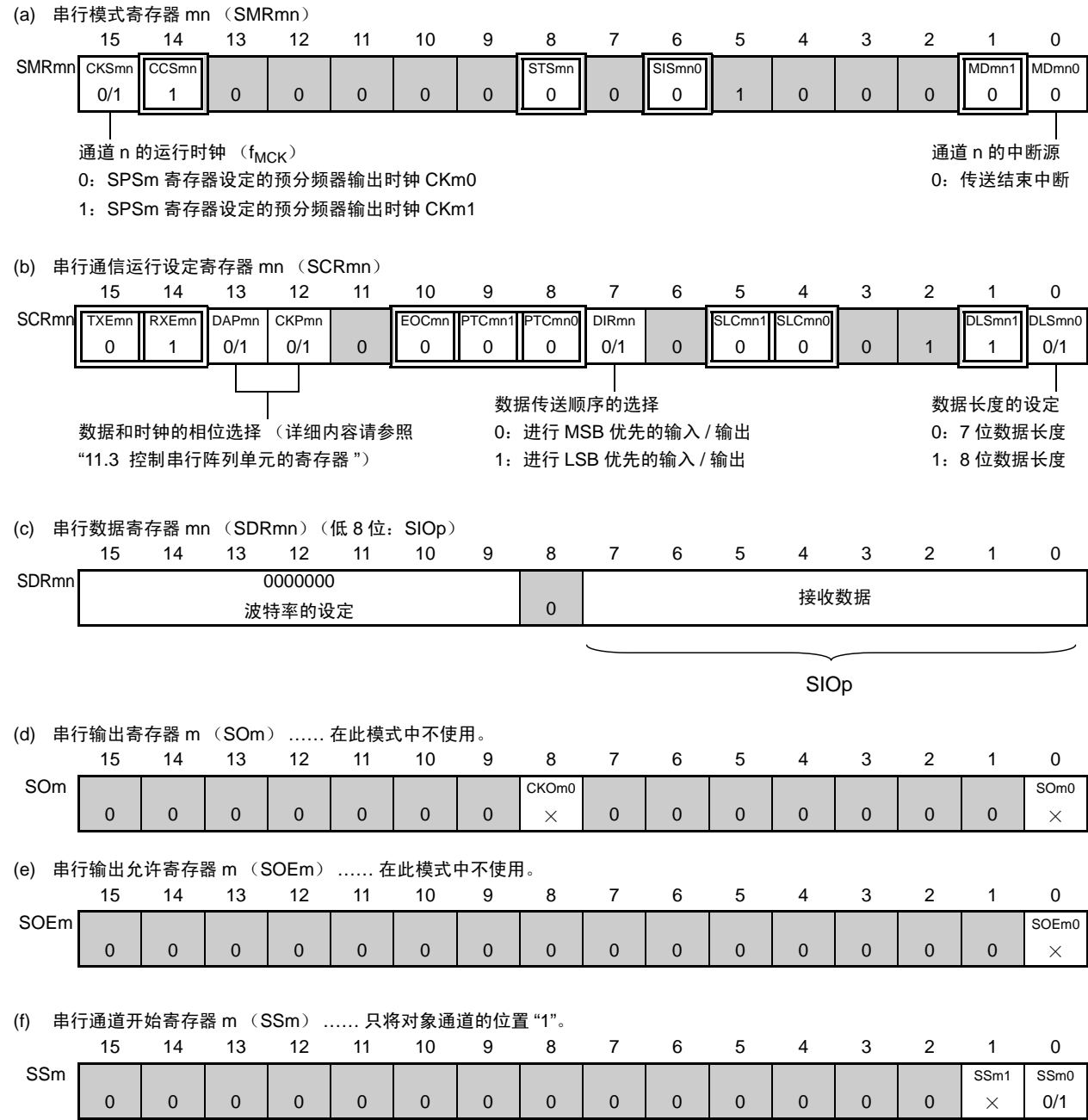
备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

$f_{SCK}$ : 串行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0) mn=00

(1) 寄存器的设定

图 11-56 3 线串行 I/O（CSI00）从属接收时的寄存器设定内容例子



备注 1. m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

2. □: 在从属接收模式中为固定设定。■: 不能设定（设定初始值）。  
×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-57 从属接收的初始设定步骤

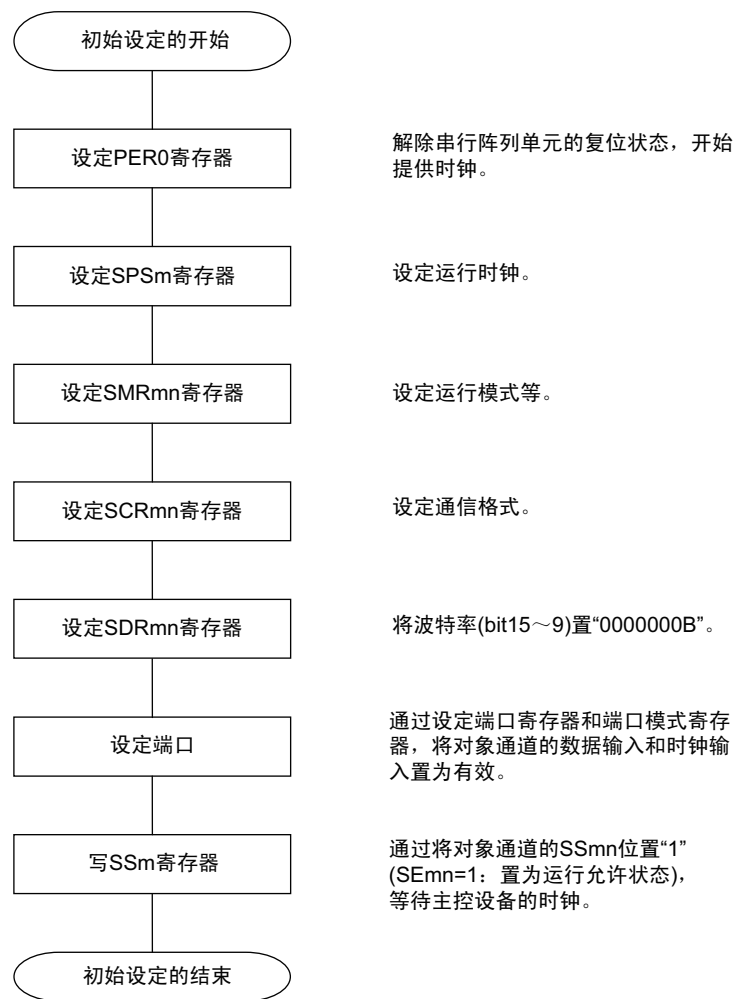


图 11-58 从属接收的中止步骤

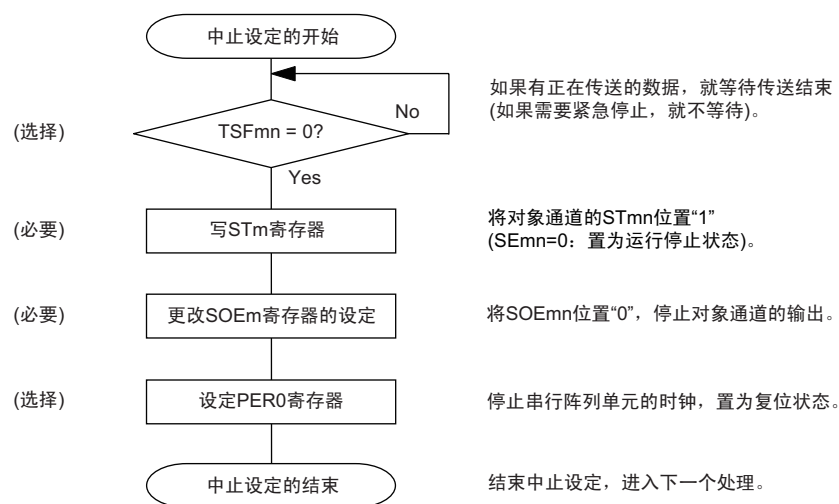
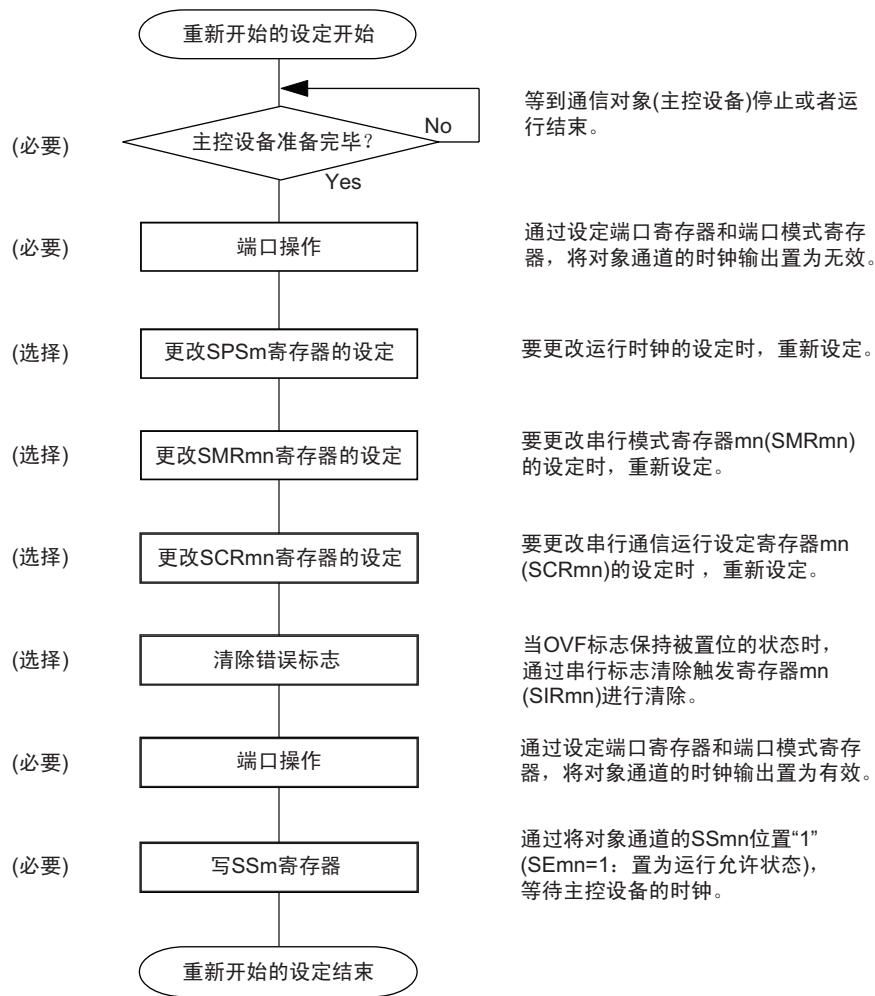


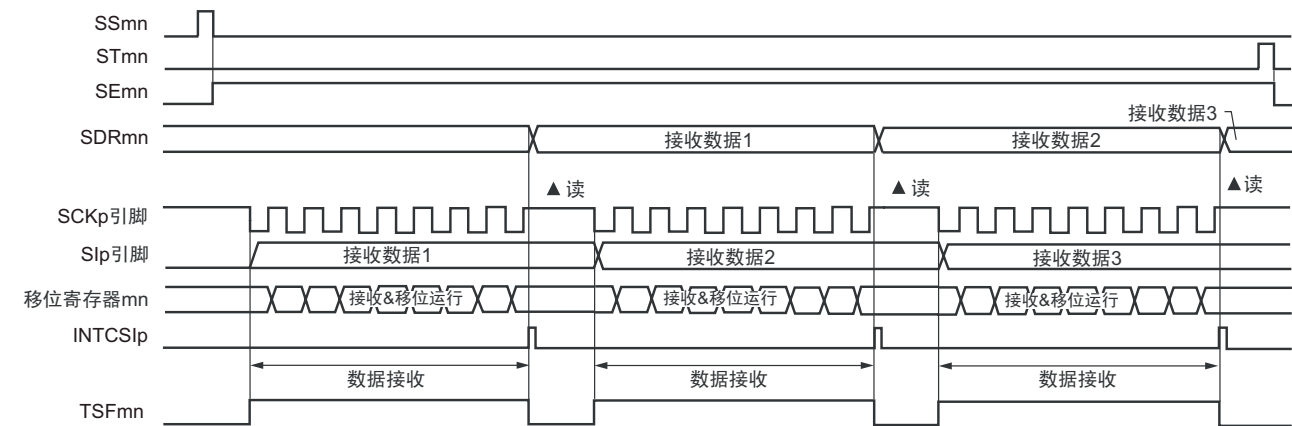
图 11-59 重新开始从属接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

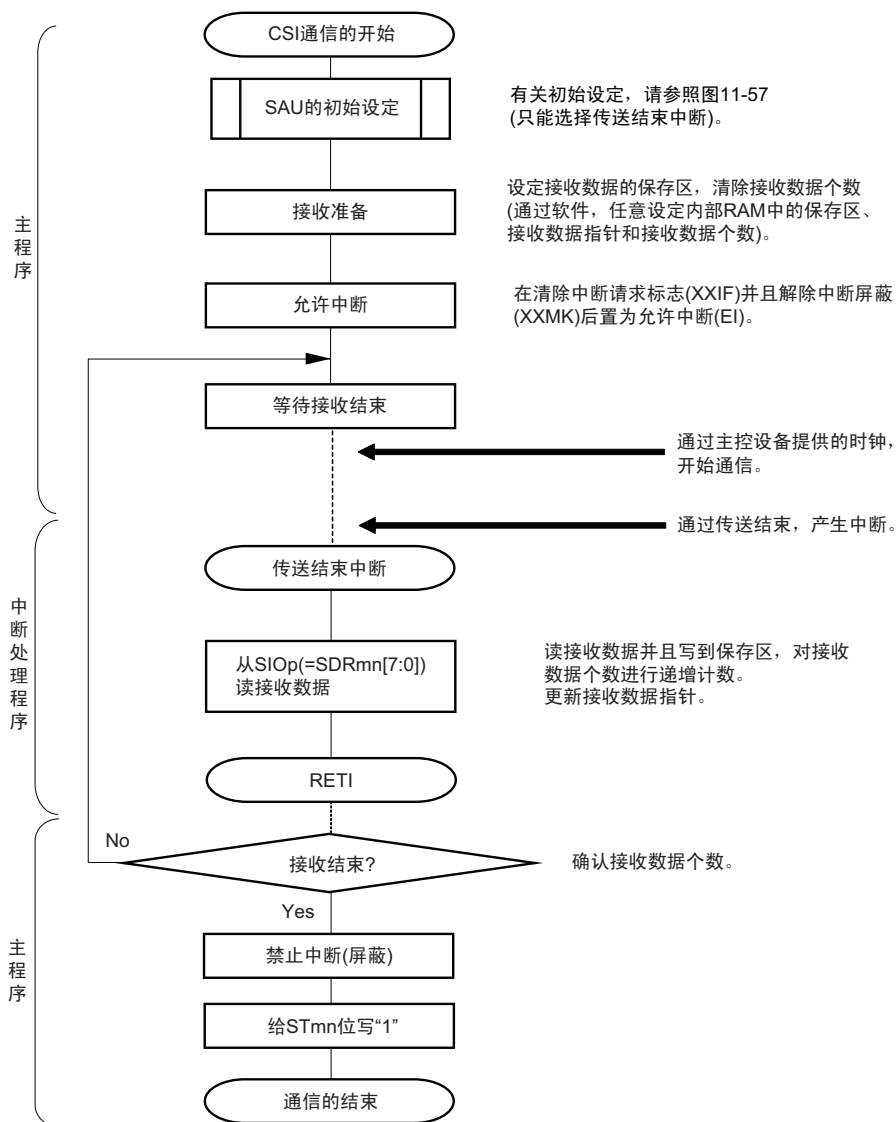
(3) 处理流程（单次接收模式）

图 11-60 从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

图 11-61 从属接收（单次接收模式）的流程图



### 11.5.6 从属发送和接收

从属发送和接收是指在从其他设备输入传送时钟的状态下 R7F0C010 和其他设备进行数据发送和接收的运行。

3 线串行 I/O	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、S0m0
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输入 / 输出。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入 / 输出。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先

- 注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{MCK}/6[\text{Hz}]$ 。  
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

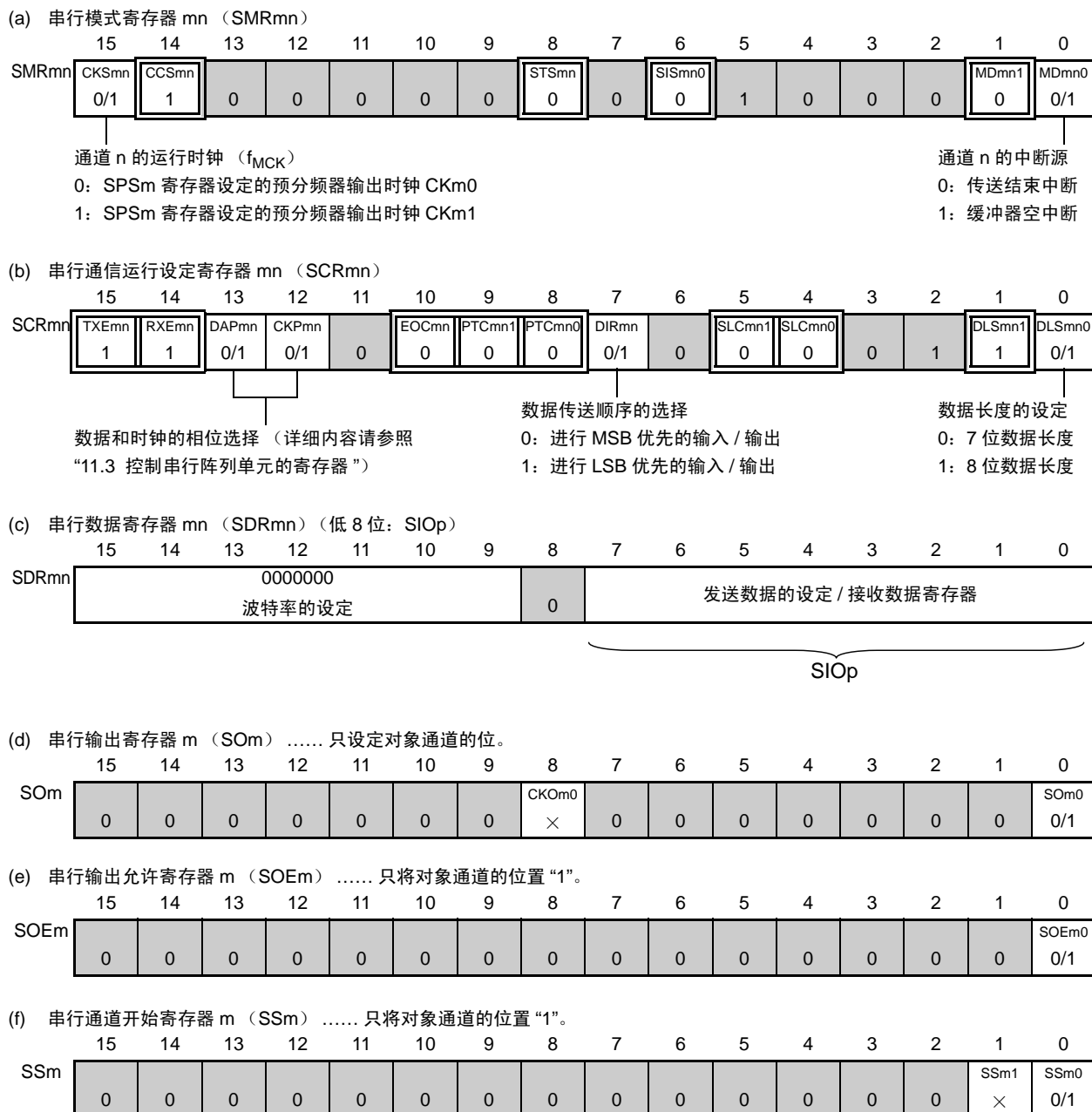
备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

$f_{SCK}$ : 串行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0) mn=00

## (1) 寄存器的设定

图 11-62 3 线串行 I/O (CSI00) 从属发送和接收时的寄存器设定内容例子



注意 在主控设备开始输出时钟前，必须给 SIOP 寄存器设定发送数据。

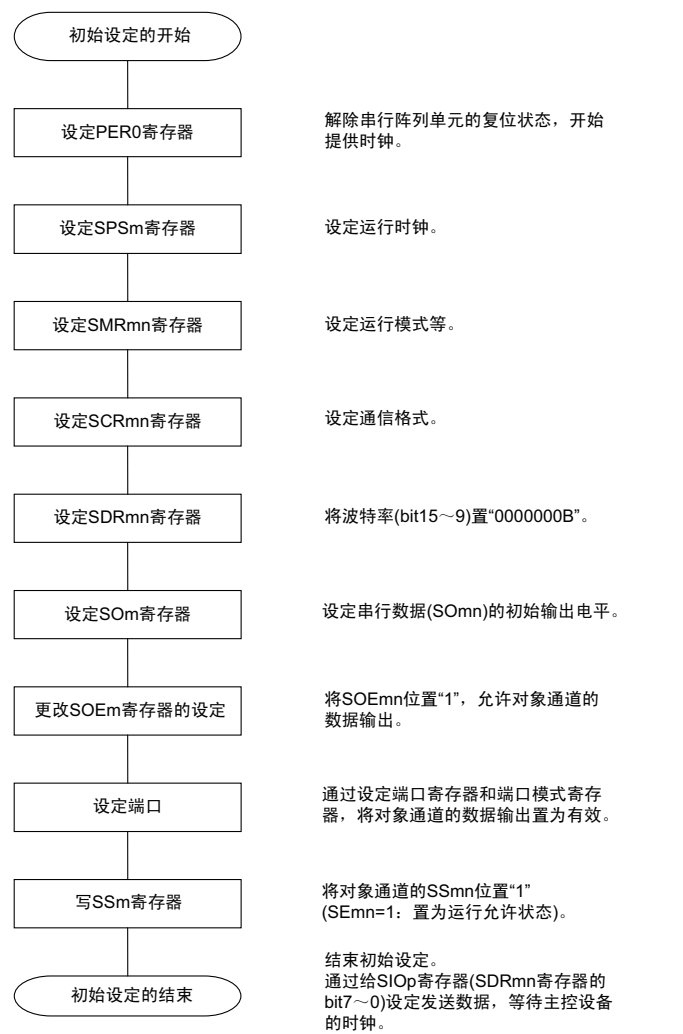
备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

2.   : 在 CSI 从属发送和接收模式中为固定设定。  : 不能设定 (设定初始值)。  
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。



(2) 操作步骤

图 11-63 从属发送和接收的初始设定步骤



注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

图 11-64 从属发送和接收的中止步骤

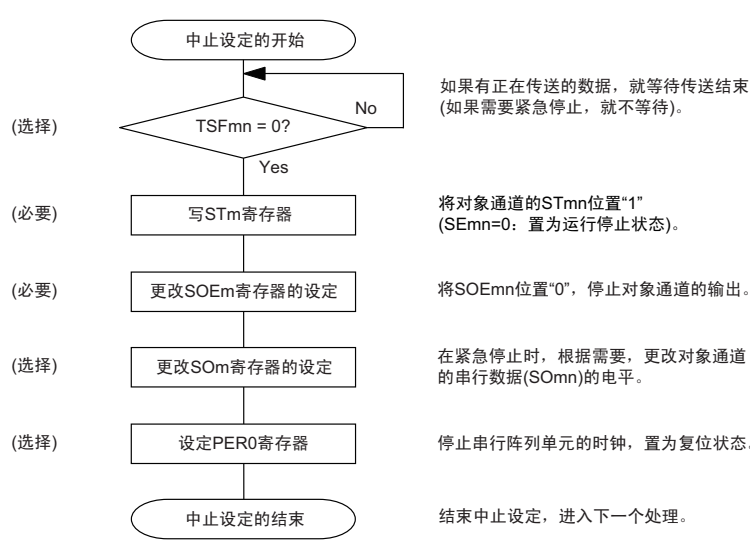
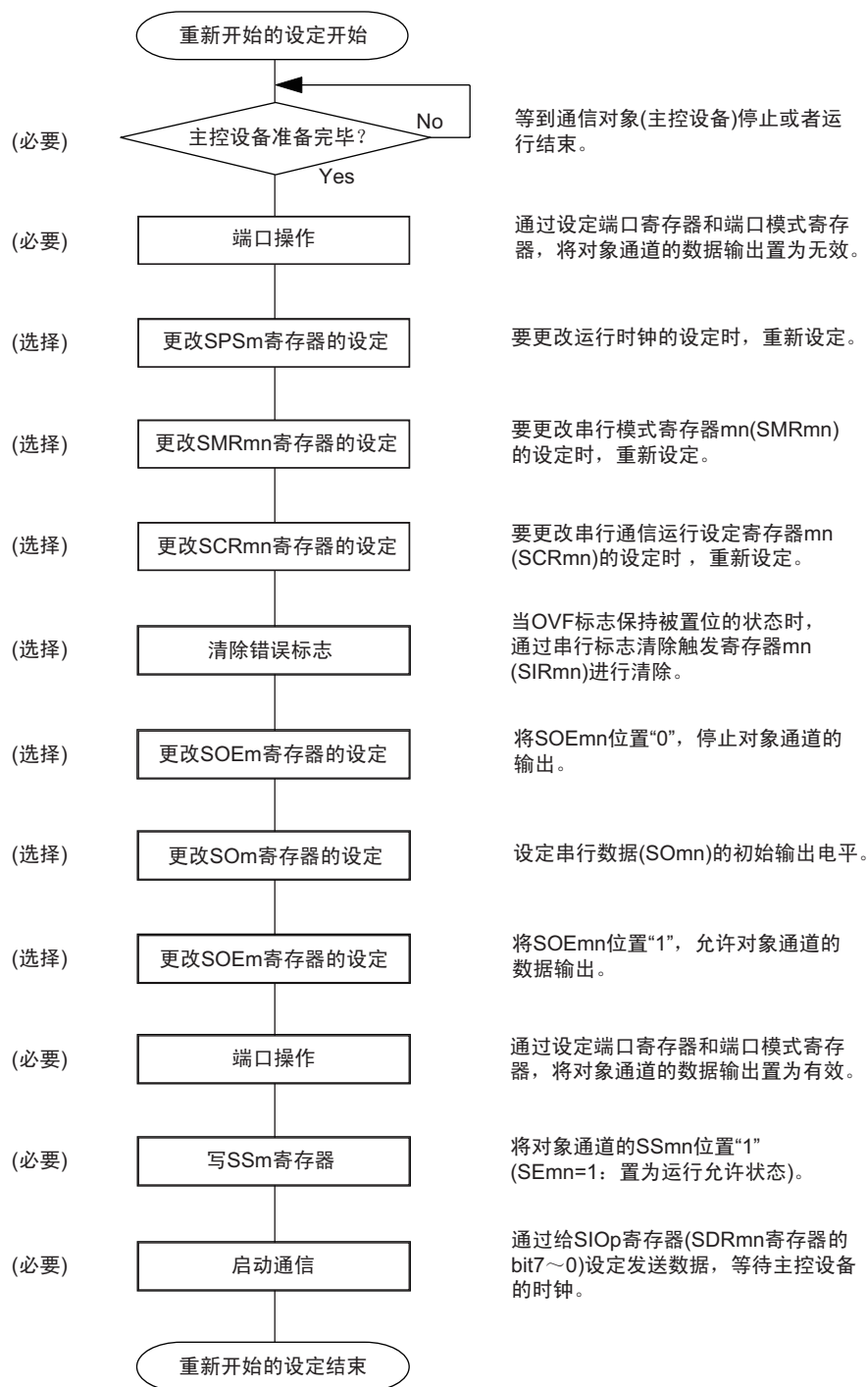


图 11-65 重新开始从属发送和接收的设定步骤

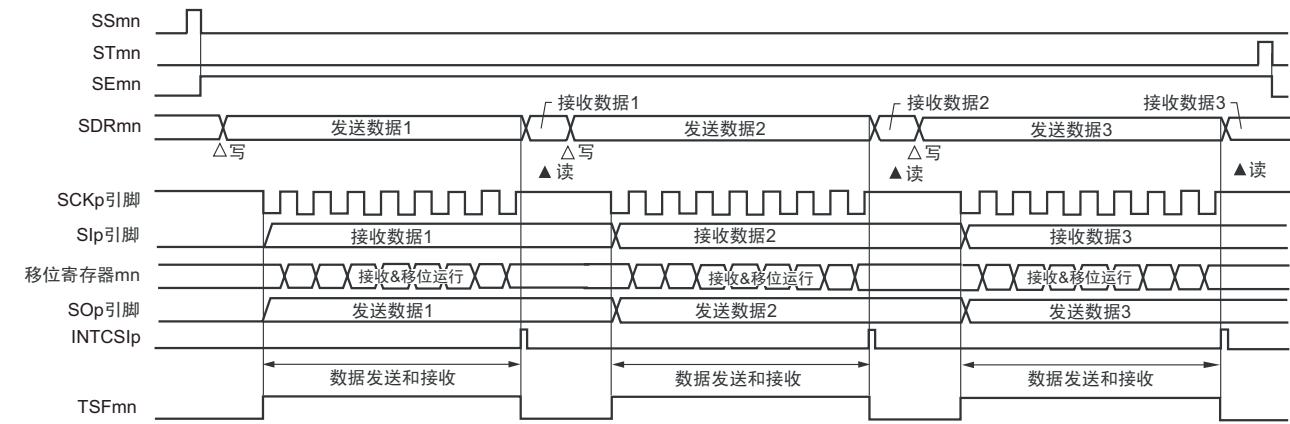


注意 1. 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (主控设备) 停止或者通信结束后进行初始设定而不是进行重新开始的设定。

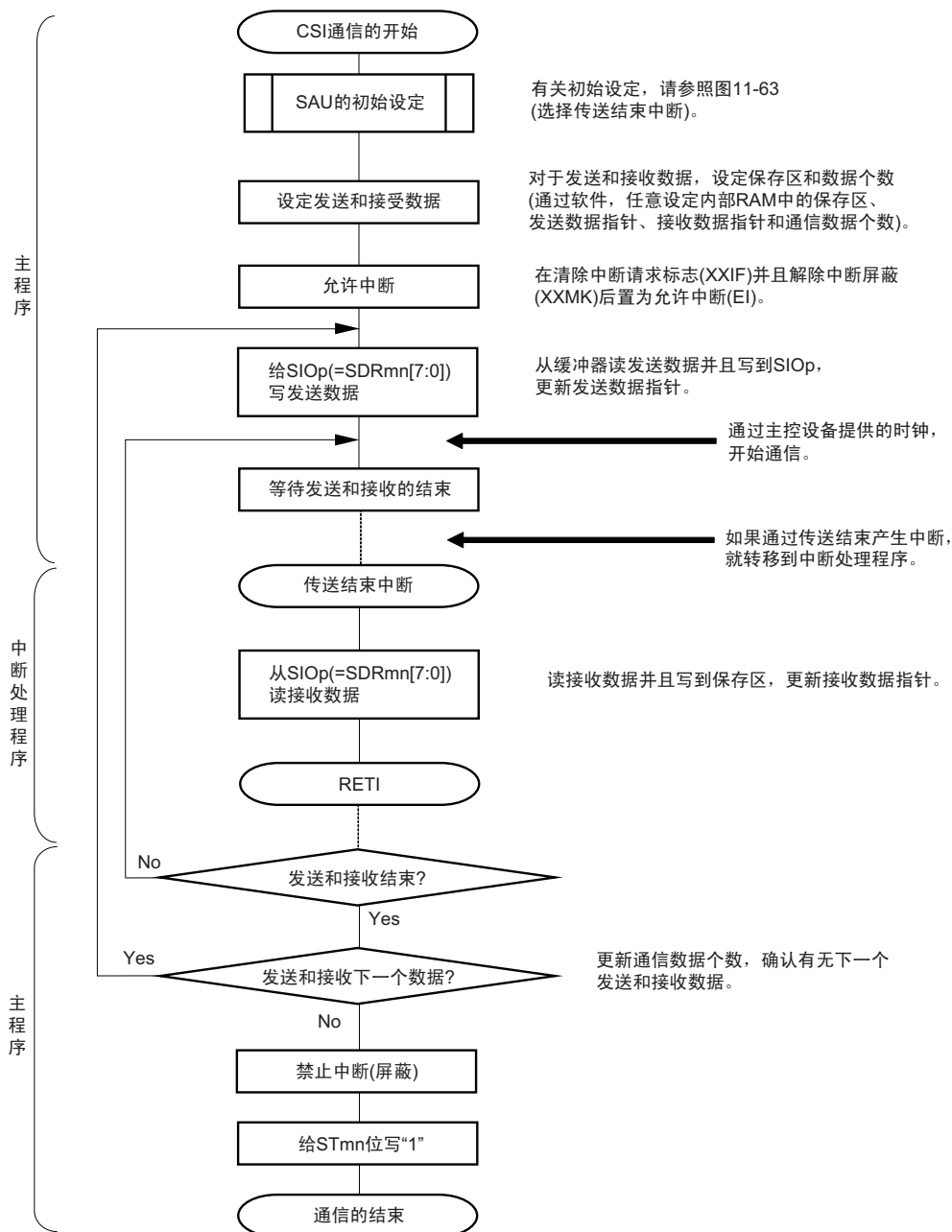
(3) 处理流程（单次发送和接收模式）

图 11-66 从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00) mn=00

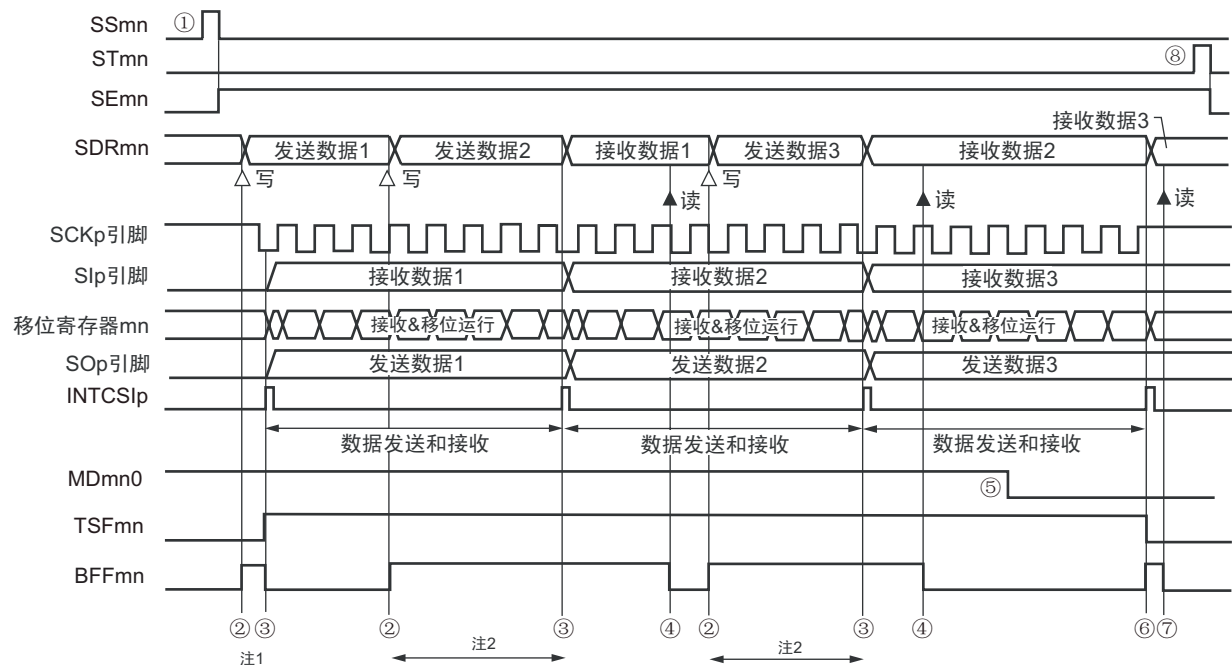
图 11-67 从属发送和接收（单次发送和接收模式）的流程图



**注意** 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

(4) 处理流程（连续发送和接收模式）

图 11-68 从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



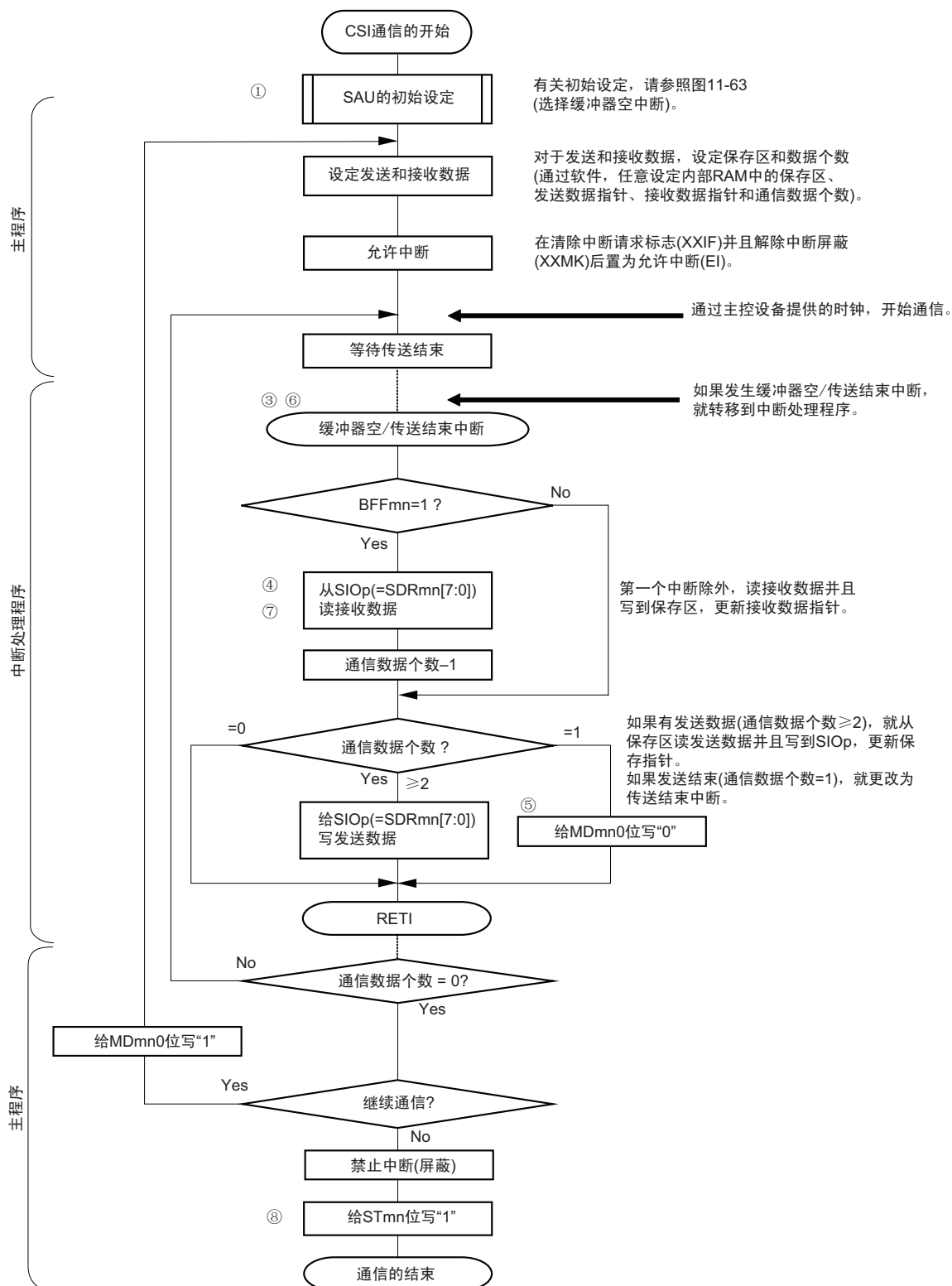
- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注 1. 图中的①～⑧对应“图 11-69 从属发送和接收（连续发送和接收模式）的流程图”中的①～⑧。

2. m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00） mn=00

图 11-69 从属发送和接收（连续发送和接收模式）的流程图



**注意** 在主机设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 图中的①~⑧对应“图 11-68 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

11.5.7 SNOOZE 模式功能（CSI00）

这是在 STOP 模式中通过检测 SCKp 引脚的输入使 CSI 进行接收的模式。通常，在 STOP 模式中 CSI 停止通信，但是使用此模式并且通过检测 SCKp 引脚的输入，能在 CPU 不运行的状态下进行 CSI 的接收。

要在 SNOOZE 模式中使用 CSI 时，必须在转移到 STOP 模式前进行以下的设定（参照“图 11-71 和图 11-73 SNOOZE 模式运行的流程图”）。

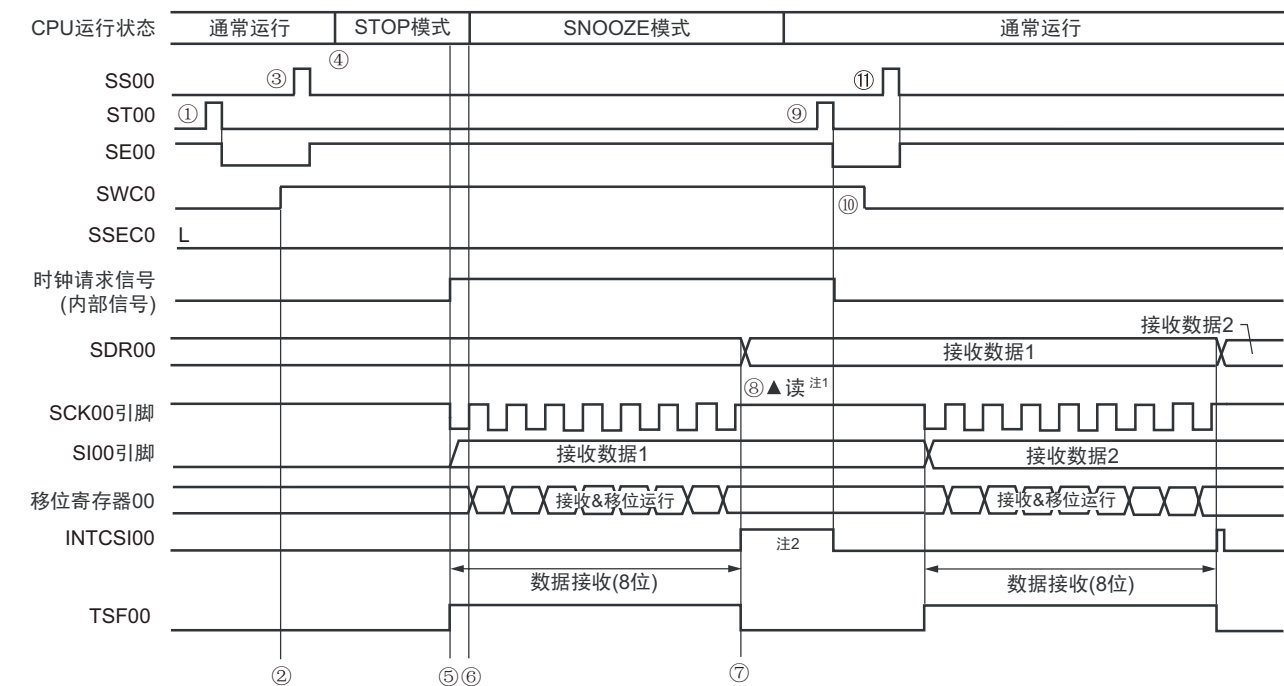
- 必须在即将转移到 STOP 模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。在初始设定结束后将串行通道开始寄存器 m（SSm）的 SSm1 位置“1”。

在转移到 STOP 模式后，如果检测到 SCKp 引脚的边沿，就开始 CSI 接收。

- 注意 1. 只有在选择高速内部振荡器时钟作为 f<sub>CLK</sub> 时才能设定 SNOOZE 模式。
2. 在 SNOOZE 模式中使用时，最大传送速率为 1Mbps。

(1) SNOOZE 模式运行（启动一次）

图 11-70 SNOOZE 模式运行（启动一次）的时序图（类型 1：DAPmn=0、CKPmn=0）

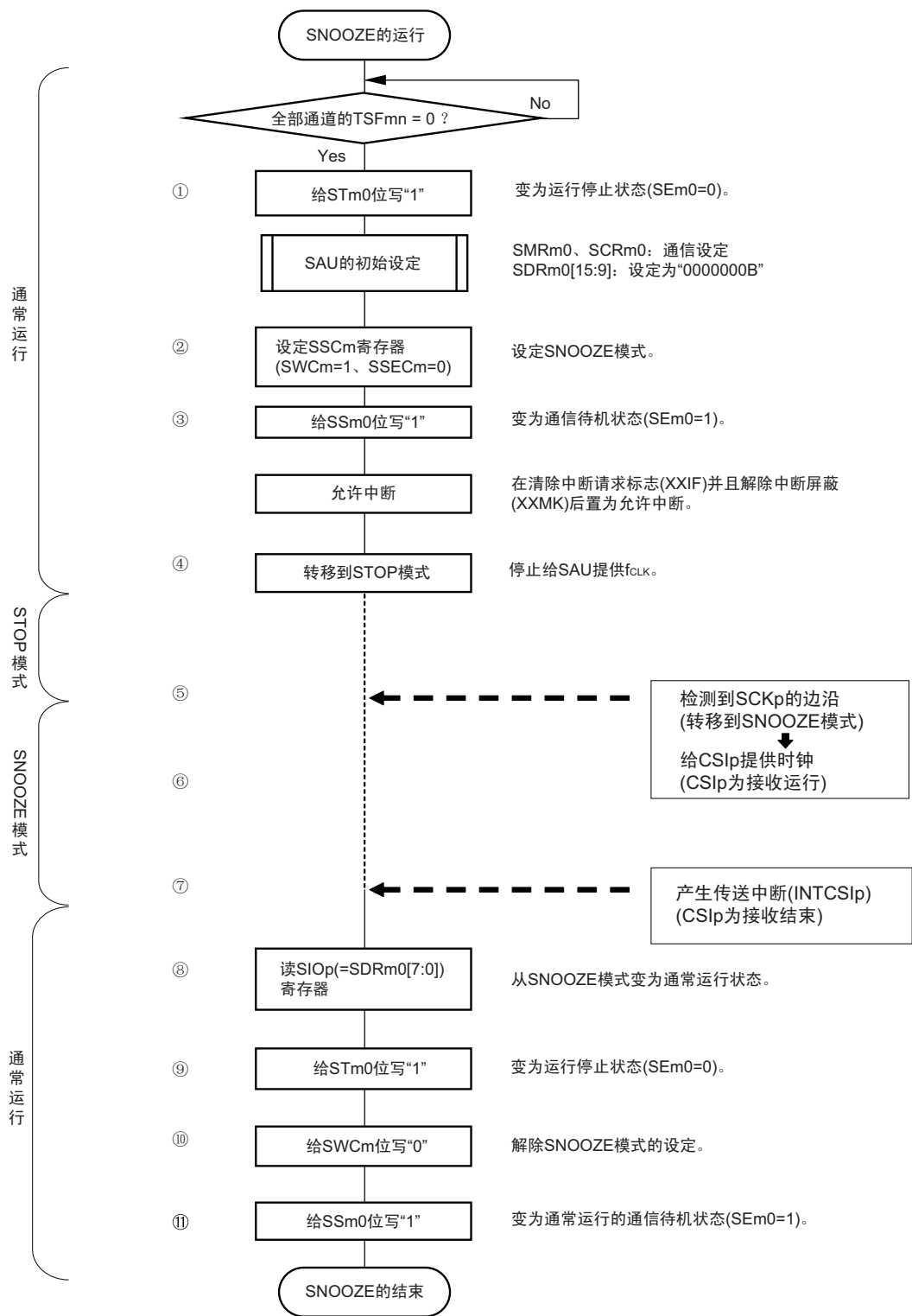


- 注 1. 必须在 SWCm 位为“1”的状态下并且在检测到 SCKp 引脚输入的下一个边沿前读接收数据。
2. 在将 SWCm 位置“0”或者检测到 SCKp 引脚输入的下一个边沿时，清除传送结束中断（INTCSIp）。

注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm0 位置“1”（清除 SEm0 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

- 备注 1. 图中的①～⑪对应“图 11-71 SNOOZE 模式运行（启动一次）的流程图”中的①～⑪。
2. m=0, p=00

图 11-71 SNOOZE 模式运行（启动一次）的流程图

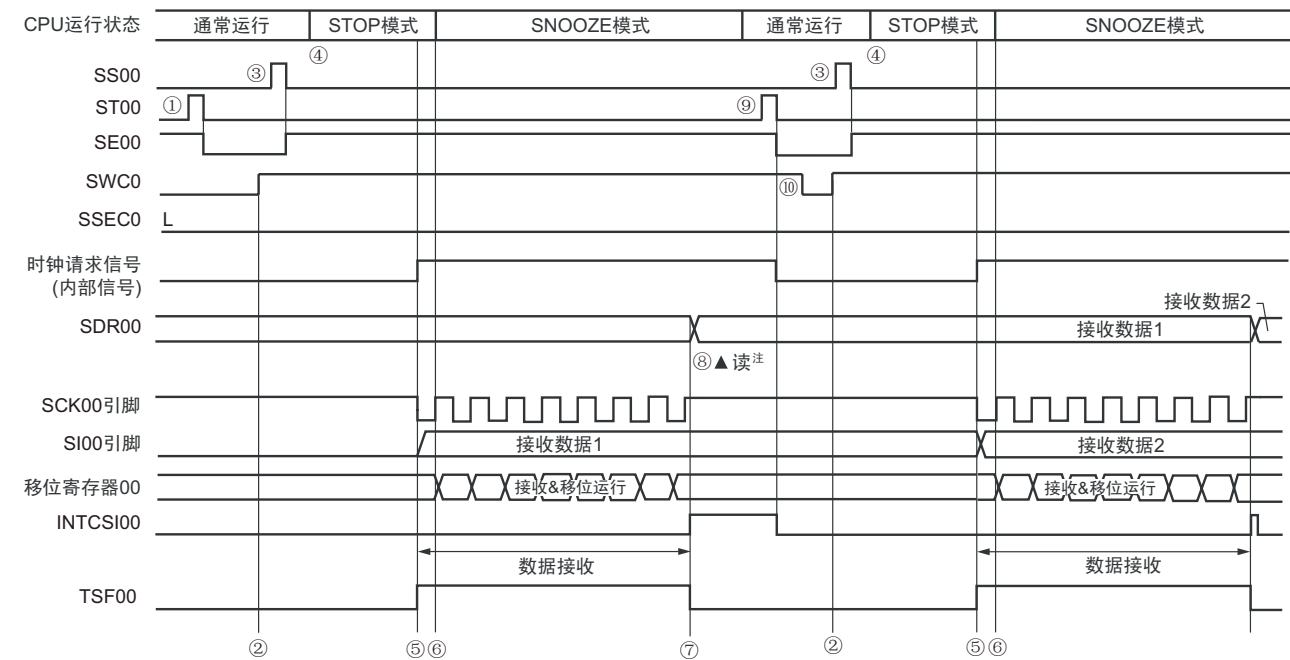


备注 1. 图中的①～⑪对应“图 11-70 SNOOZE 模式运行（启动一次）的时序图”中的①～⑪。  
2. m=0, p=00



(2) SNOOZE 模式运行（连续启动）

图 11-72 SNOOZE 模式运行（连续启动）的时序图（类型 1：DAPmn=0、CKPmn=0）



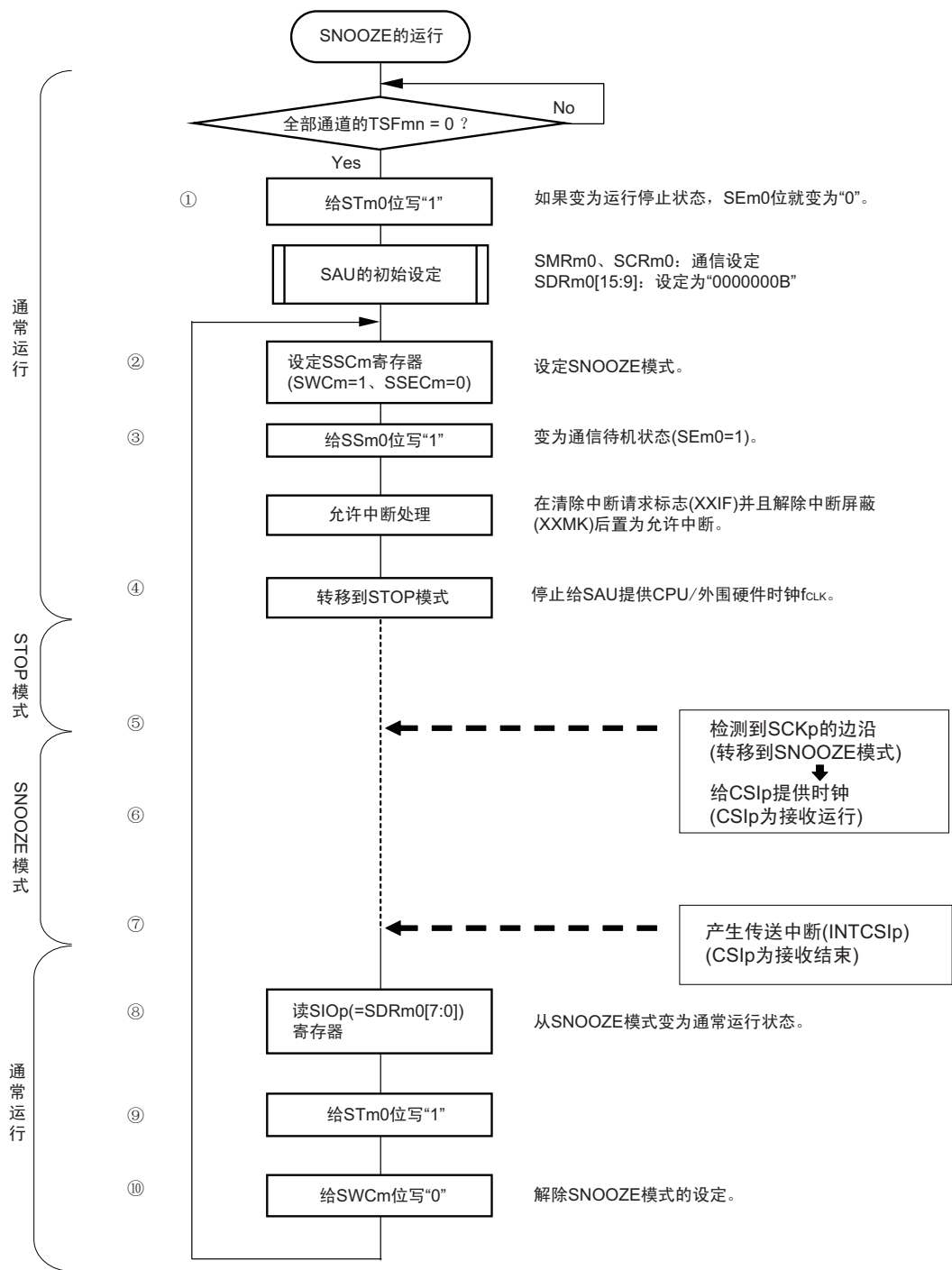
注 必须在 SWCm 位为“1”的状态下并且在检测到 SCKp 引脚输入的下一个边沿前读接收数据。

- 注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm0 位置“1”（清除 SEm0 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。
2. 当 SWCm 位为“1”时，BFFm1 标志和 OVFm1 标志不变。

备注 1. 图中的①～⑩对应“图 11-73 SNOOZE 模式运行（连续启动）的流程图”中的①～⑩。

2. m=0, p=00

图 11-73 SNOOZE 模式运行（连续启动）的流程图



备注 1. 图中的①～⑩对应“图 11-72 SNOOZE 模式运行（连续启动）的时序图”中的①～⑩。  
2. m=0, p=00

### 11.5.8 传送时钟频率的计算

3 线串行 I/O（CSI00）通信的传送时钟频率能用以下计算式进行计算。

#### (1) 主控设备

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

#### (2) 从属设备

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

注 容许的最大传送时钟频率为  $f_{\text{MCK}}/6$ 。

备注 因为SDRmn[15:9]的值为串行数据寄存器mn（SDRmn）的bit15～9的值（0000000B～1111111B），所以为0～127。

运行时钟（ $f_{\text{MCK}}$ ）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn）。

表 11-2 3 线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 ( $f_{MCK}$ ) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$f_{CLK}=32\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	$f_{CLK}$	32MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	7.81kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	3.91kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.95kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	977Hz
1	0	0	0	0	X	X	X	X	$f_{CLK}$	32MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	7.81kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	3.91kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.95kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	977Hz
上述以外									禁止设定。	

注 要更改被选择为  $f_{CLK}$  的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（SAU）的运行（串行通道停止寄存器 m（STM）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0) n: 通道号 (n=0) mn=00

11.5.9 在 3 线串行 I/O（CSI00）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（CSI00）通信过程中发生错误时的处理步骤如图 11-74 所示。

图 11-74 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

备注 m: 单元号（m=0） n: 通道号（n=0） mn=00

## 11.6 从属选择输入功能的时钟同步串行通信的运行

SAU0 的通道 0 是支持从属选择输入功能的时钟同步串行通信的通道。

[ 数据的发送和接收 ]

- 7 位或者 8 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定

[ 时钟控制 ]

- 输入/输出时钟的相位控制
- 设定由预分频器和通道内部计数器产生的传送周期。
- 最大传送速率注

从属通信:  $\text{Max. } f_{\text{MCK}}/6$

[ 中断功能 ]

- 传送结束中断、缓冲器空中断

[ 错误检测标志 ]

- 溢出错误

[ 扩展功能 ]

- 从属选择功能

注 必须在满足 SCK 周期时间 ( $t_{\text{KCY}}$ ) 特性的范围内使用。详细内容请参照“第 27 章 电特性”。

单元	通道	用作 CSI	用作 UART
0	0	CSI00 (支持从属选择输入功能)	UART0
	1	—	

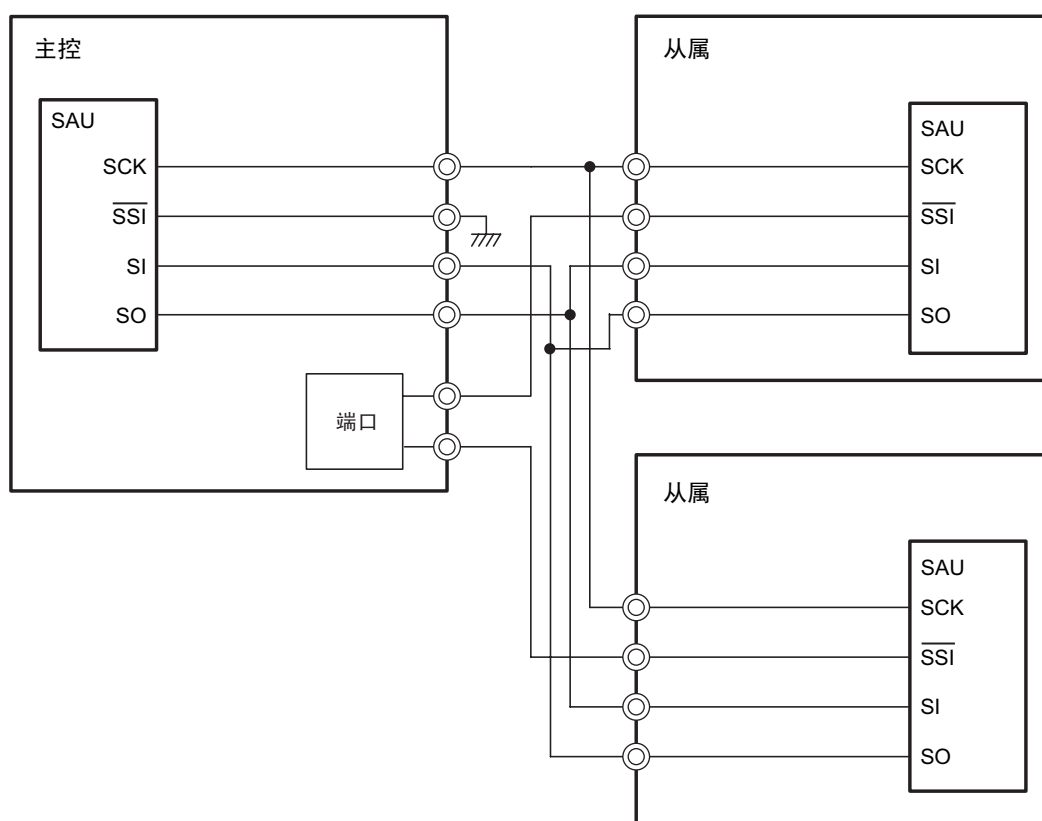
从属选择输入功能有以下 3 种通信运行:

- 从属发送 (参照 11.6.1)
- 从属接收 (参照 11.6.2)
- 从属的发送和接收 (参照 11.6.3)

能通过使用从属选择输入功能，使 1 个主控设备连接多个从属设备进行通信。主控设备对通信对象的从属设备（1 个）进行从属选择信号的输出，各从属设备判断自己是否被选择为通信对象并且控制 SO 引脚的输出。当被选择为通信对象的从属设备时，SO 引脚能对主控设备进行发送数据的通信；当不被选择为通信对象的从属设备时，SO 引脚变为高阻抗输出，因此能避免和其他从属设备 SO 输出的短路，而且即使输入主控设备的串行时钟也不进行发送和接收。

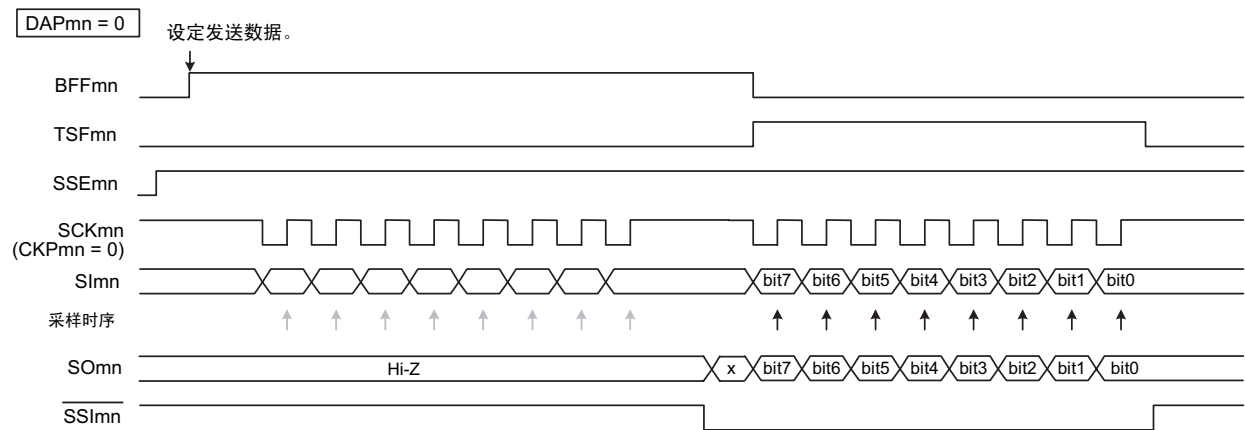
注意 必须通过端口的操作输出从属选择信号。

图 11-75 从属选择输入功能的结构例子

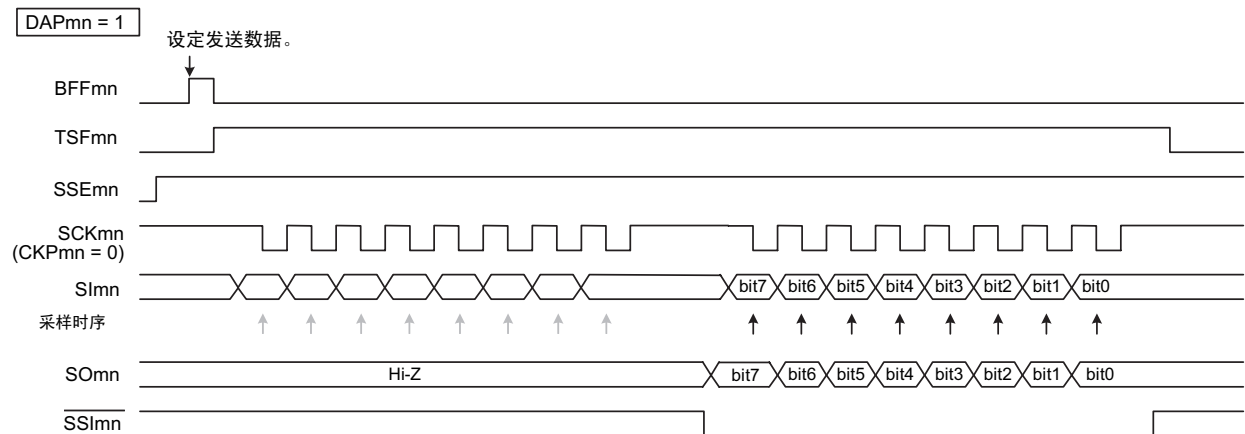


注意 必须在  $V_{DD} \geq V_b$  的情况下使用。

图 11-76 从属选择输入功能的时序图



在SSImn为高电平期间，即使在SCKmn(串行时钟)的下降沿也不进行发送，而且也不进行与上升沿同步的接收数据的采样。  
在SSImn为低电平期间，与串行时钟的下降沿同步输出数据(移位)并且与上升沿同步接收数据。



当DAPmn位为“1”时，如果在SSImn为高电平期间设定发送数据，就将最初的数据(bit7)提供给数据输出。但是，即使在SCKmn(串行时钟)的上升沿也不移位，而且也不进行与下降沿同步的接收数据的采样。如果SSImn变为低电平，就与下一个上升沿同步输出数据(移位)并且与下降沿同步接收数据。

备注    m: 单元号 (m=0)    n: 通道号 (n=0)



### 11.6.1 从属发送

从属发送是指在从其他设备输入传送时钟的状态下 R7F0C010 将数据发送到其他设备的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SO00、SSI00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[Hz]$ 注 1、2
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输出。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。</li> </ul>
时钟相位	能通过 SCRMn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{MCK}/6[Hz]$ 。

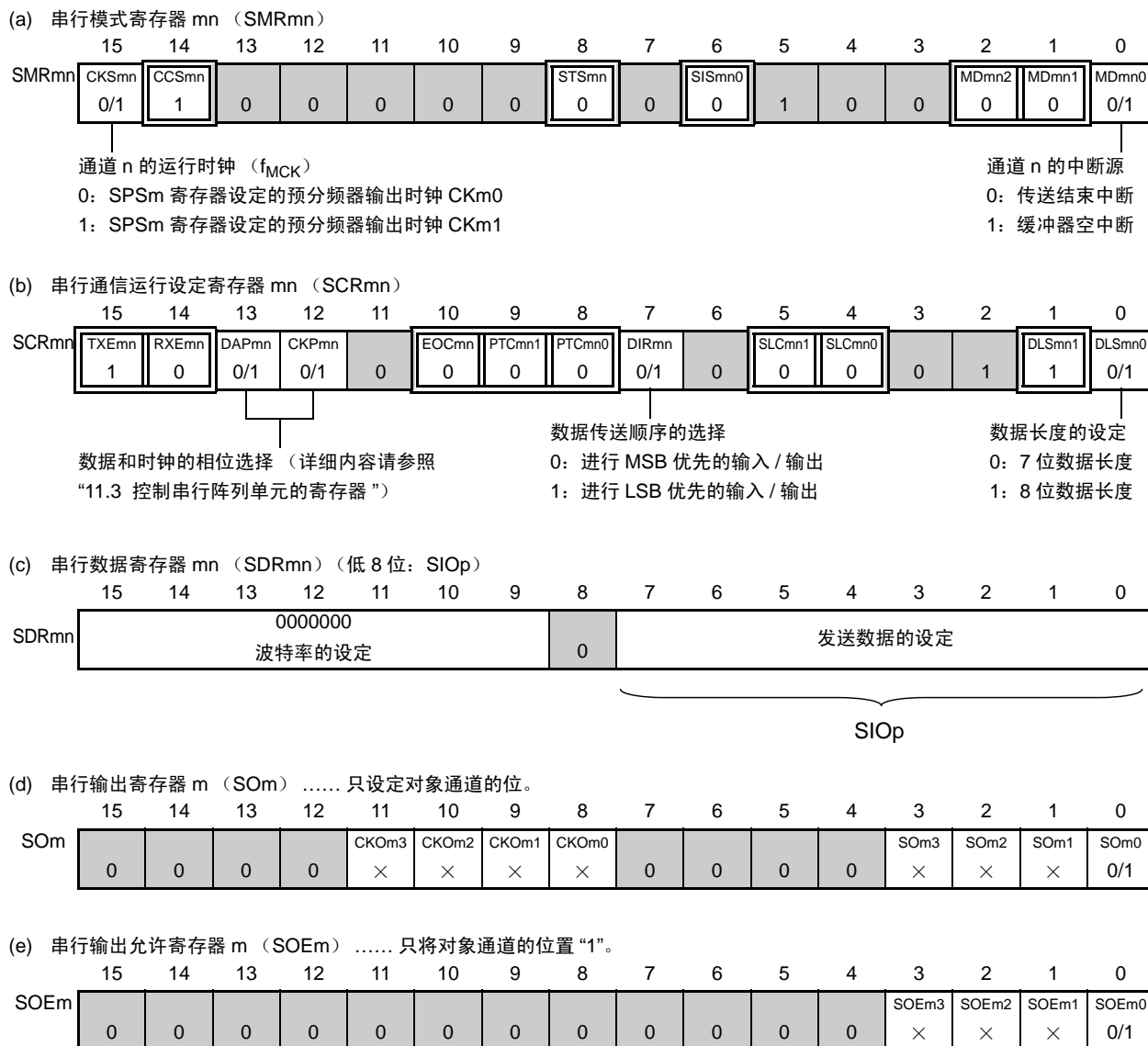
2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

## (1) 寄存器的设定

图 11-77 从属选择输入功能 (CSI00) 从属发送时的寄存器设定内容例子 (1/2)



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2.   : 在 CSI 从属发送模式中为固定设定。  : 不能设定 (设定初始值)。  
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。

图 11-77 从属选择输入功能（CSI00）从属发送时的寄存器设定内容例子 (2/2)

(f) 串行通道开始寄存器 m（SSm）…… 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC）…… 这是 CSI00 从属通道（单元 0 的通道 0）的  $\overline{\text{SSI00}}$  引脚的控制。

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

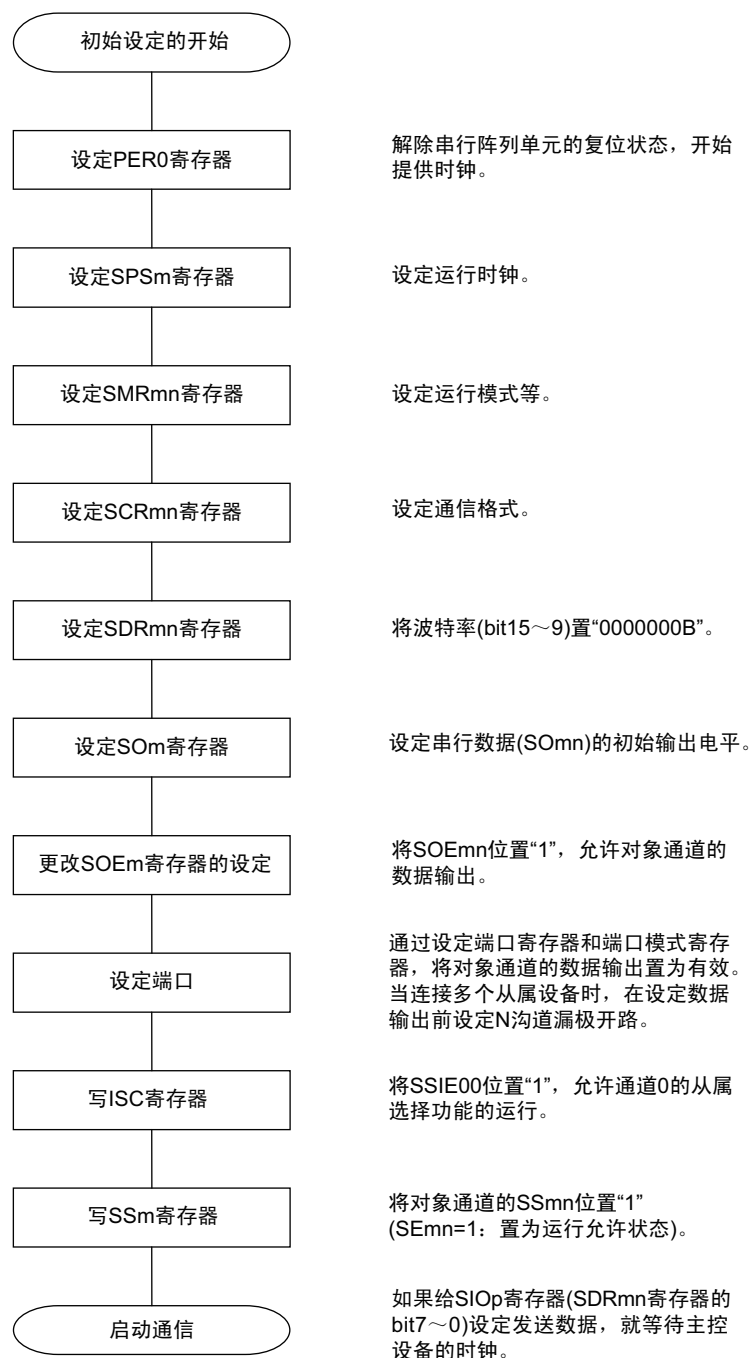
0:  $\overline{\text{SSI00}}$  引脚的输入值无效  
1:  $\overline{\text{SSI00}}$  引脚的输入值有效

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在 CSI 从属发送模式中为固定设定。☐ : 不能设定（设定初始值）。  
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
 0/1 : 根据用户的用途置“0”或者“1”。

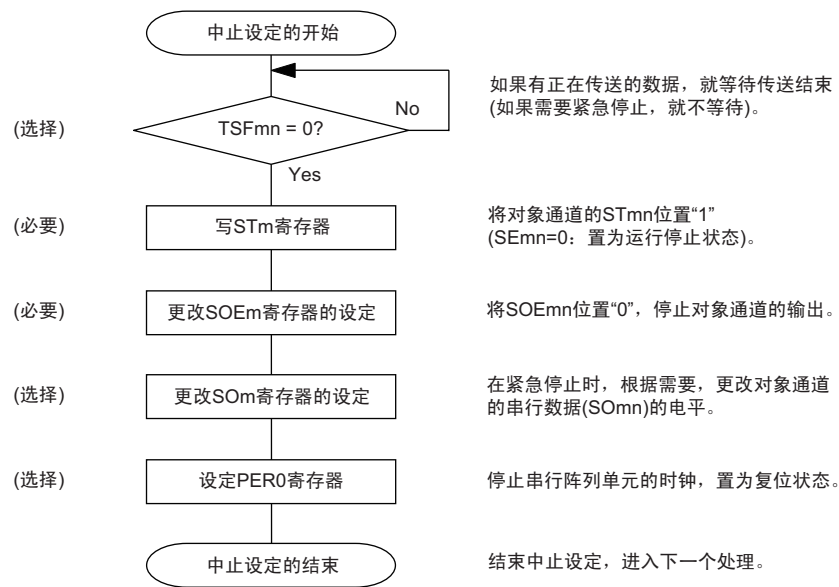
(2) 操作步骤

图 11-78 从属发送的初始设定步骤



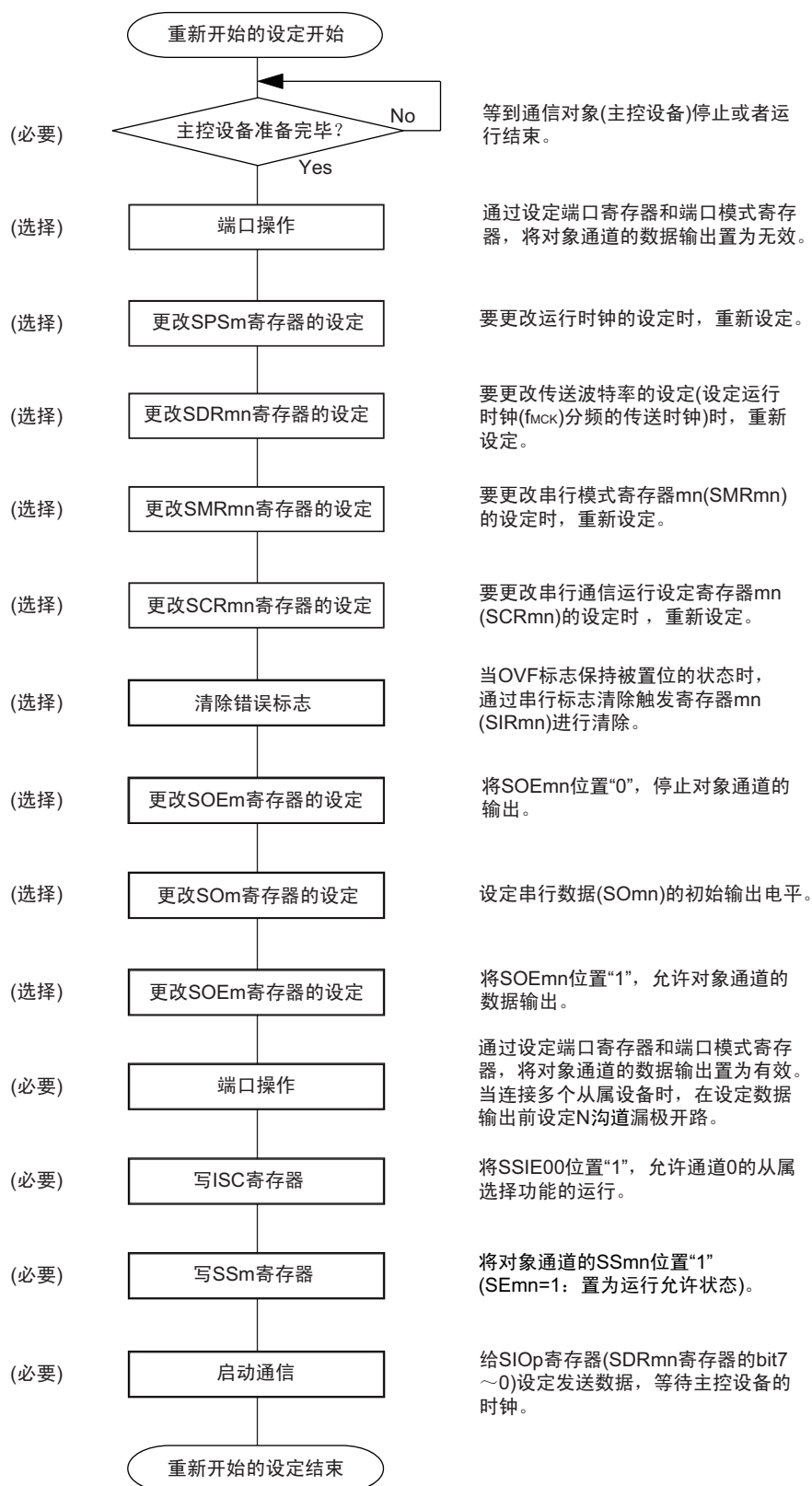
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 11-79 从属发送的中止步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

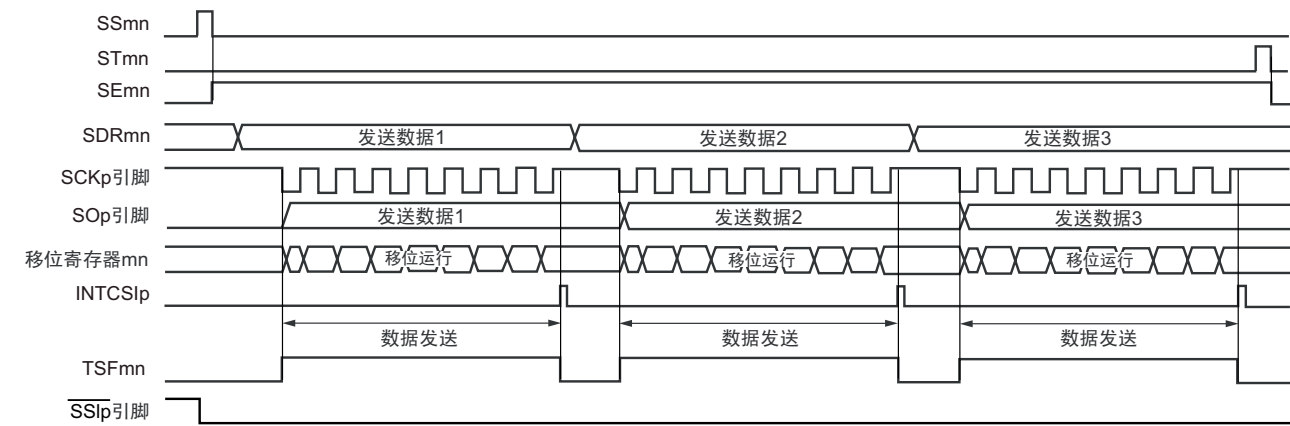
图 11-80 重新开始从属发送的设定步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

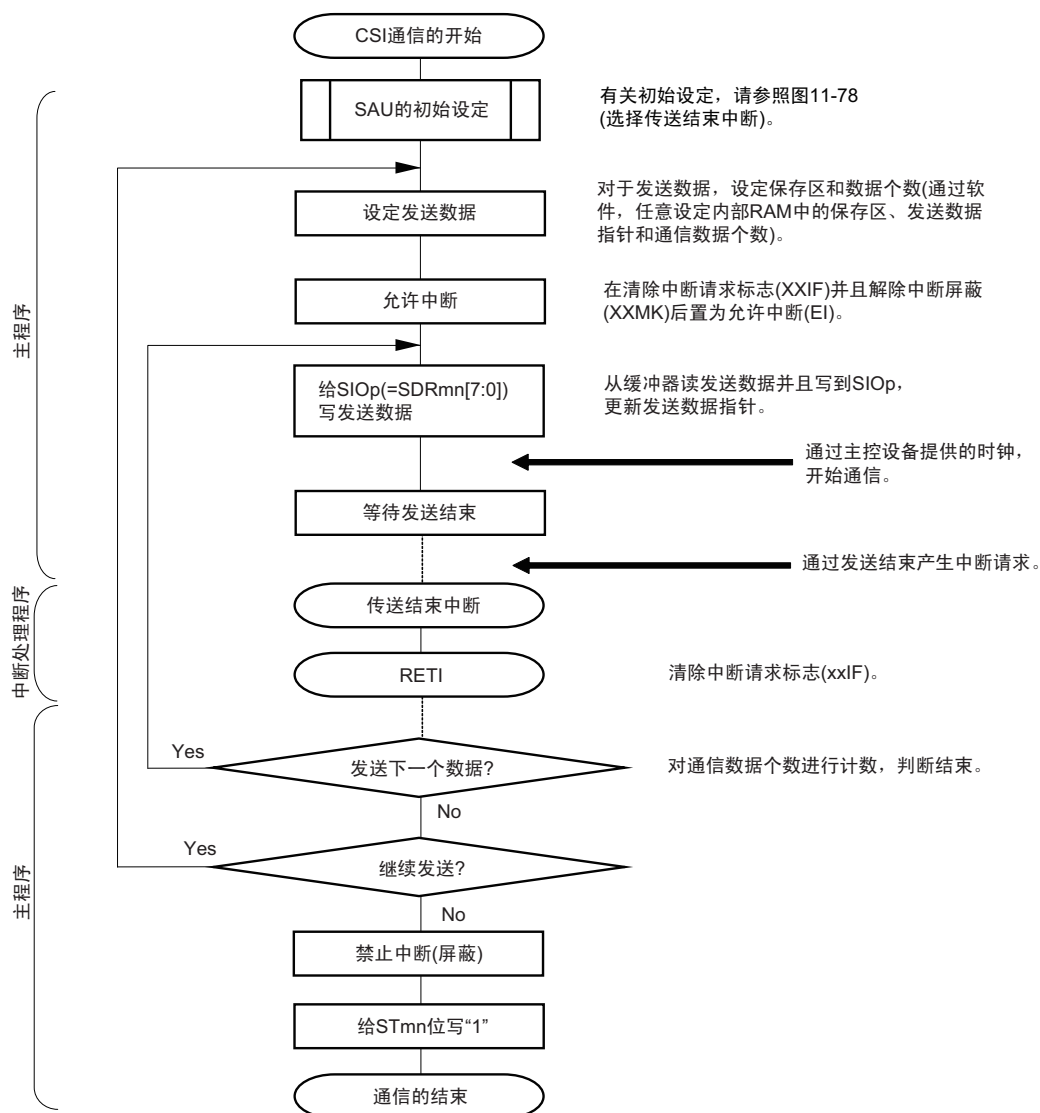
(3) 处理流程（单次发送模式）

图 11-81 从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 11-82 从属发送（单次发送模式）的流程图

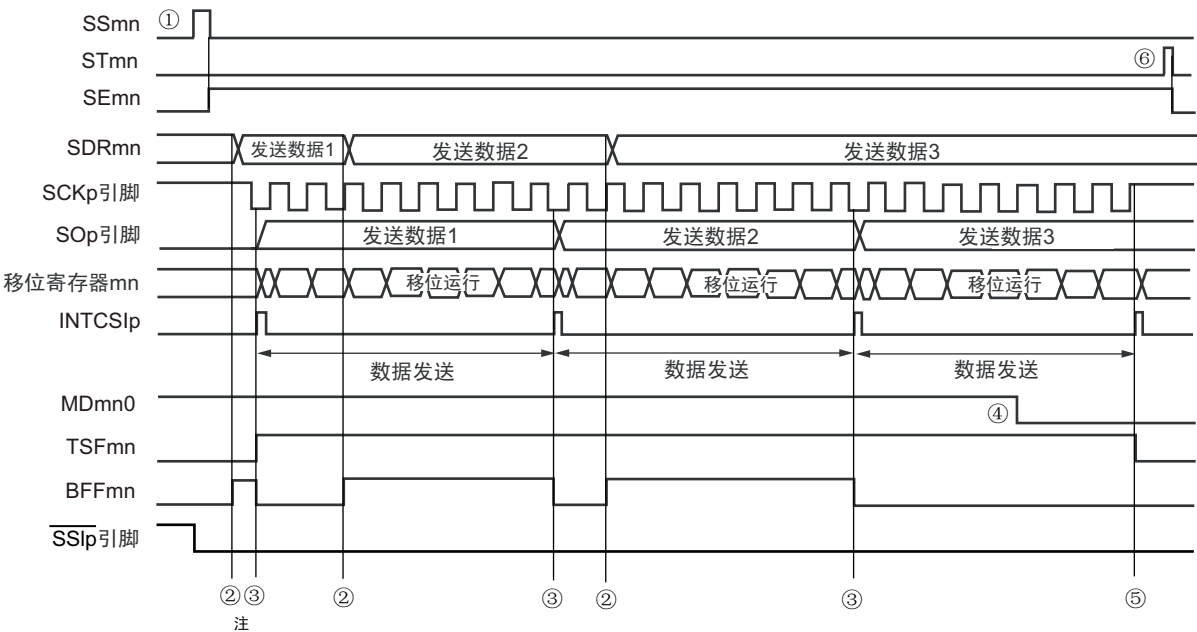


备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)



(4) 处理流程（连续发送模式）

图 11-83 从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

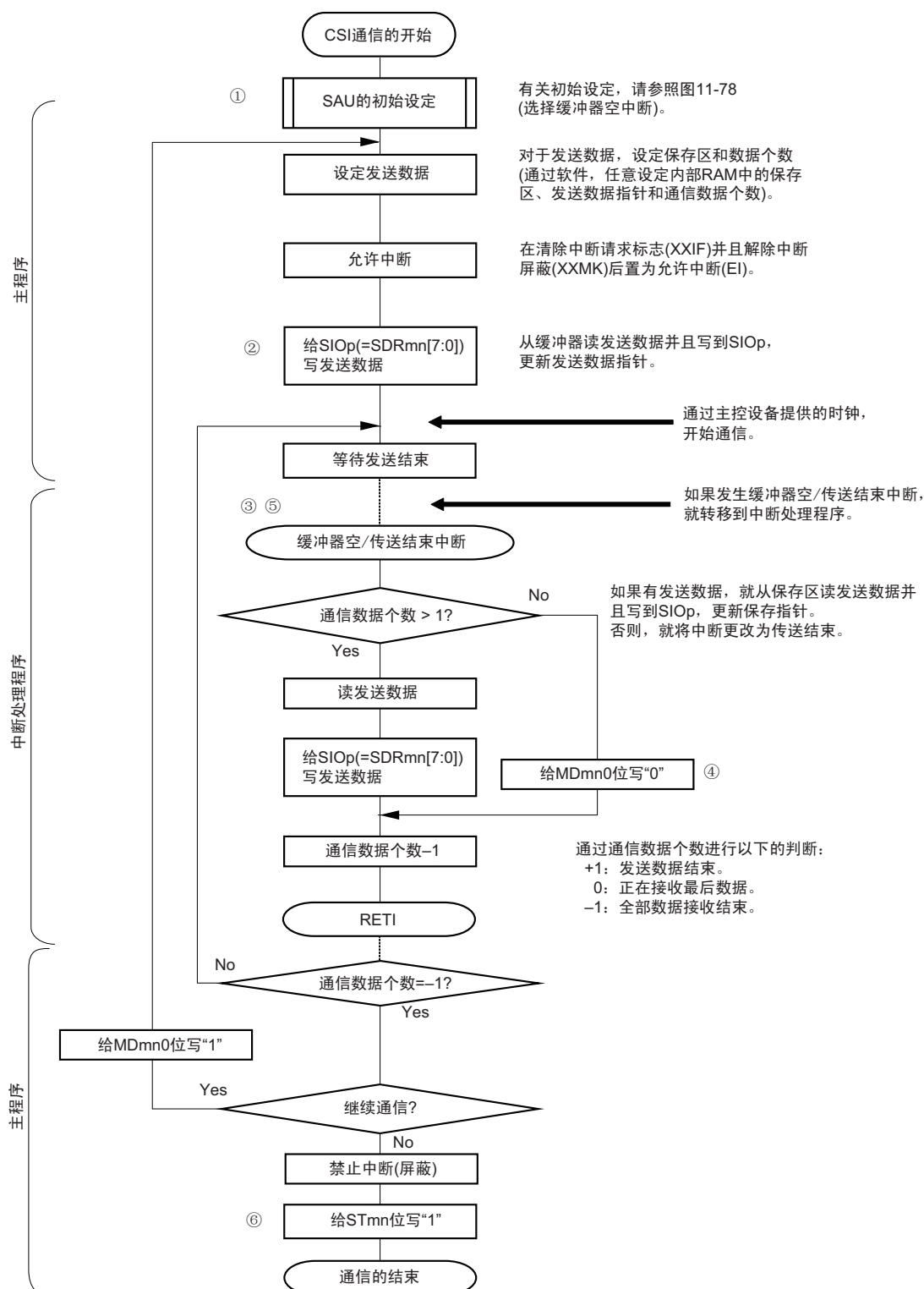


注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

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

备注 m: 单元号（m=0） n: 通道号（n=0） p: CSI 号（p=00）

图 11-84 从属发送（连续发送模式）的流程图



备注 1. 图中的①~⑥对应“图 11-83 从属发送（连续发送模式）的时序图”中的①~⑥。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

## 11.6.2 从属接收

从属接收是指在从其他设备输入传送时钟的状态下 R7F0C010 从其他设备接收数据的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、 $\overline{\text{SSI00}}$
中断	INTCSI00 只限于传送结束中断（禁止设定缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{\text{MCK}}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRMn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>DAPmn=0: 在串行时钟开始运行时，开始数据输入。</li> <li>DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入。</li> </ul>
时钟相位	能通过 SCRMn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>CKPmn=0: 正相</li> <li>CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{\text{MCK}}/6[\text{Hz}]$ 。

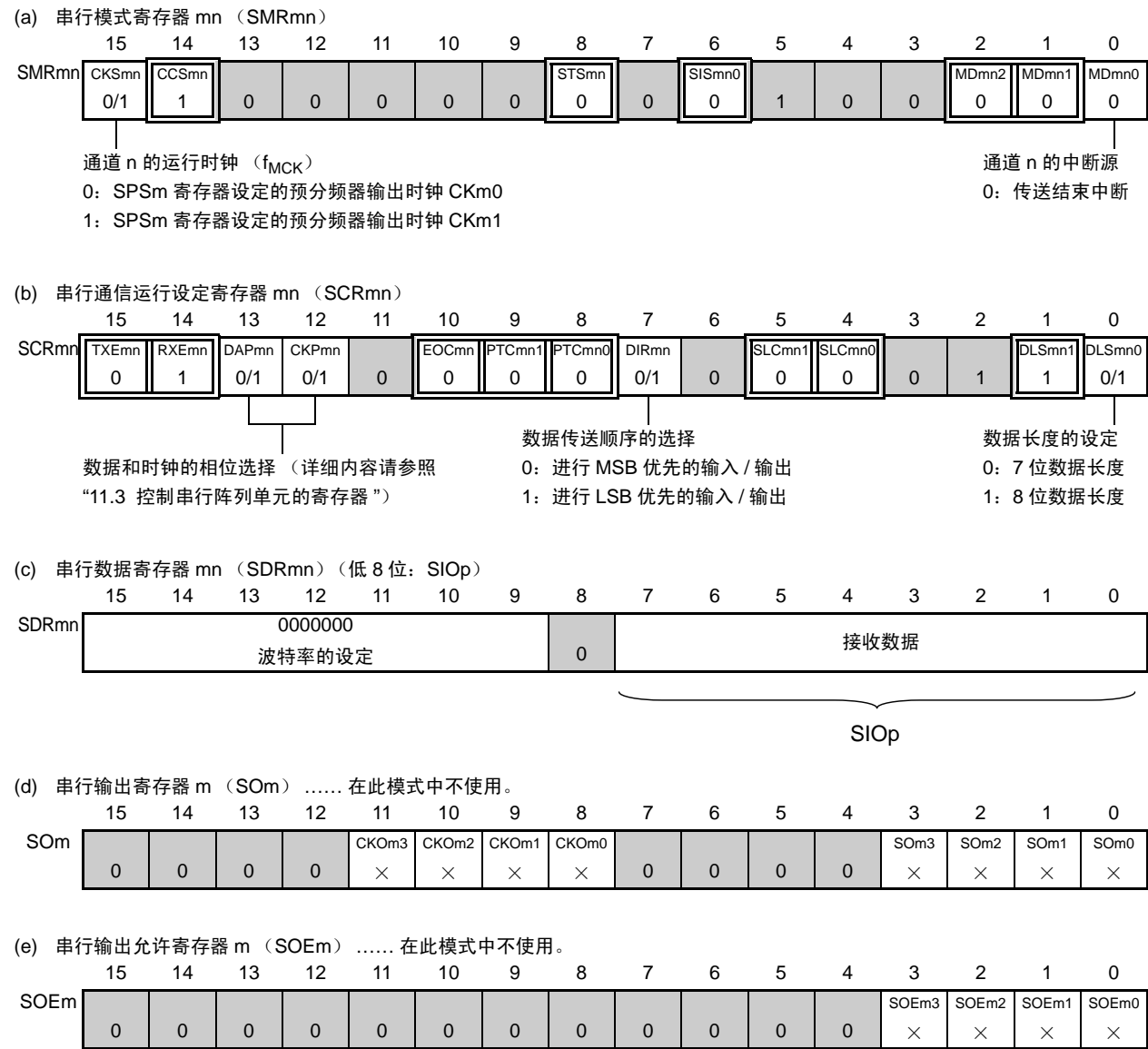
2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 1.  $f_{\text{MCK}}$ : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

(1) 寄存器的设定

图 11-85 从属选择输入功能（CSI00）从属接收时的寄存器设定内容例子 (1/2)



备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. □: 在从属接收模式中为固定设定。■: 不能设定（设定初始值）。  
×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
0/1: 根据用户的用途置“0”或者“1”。

图 11-85 从属选择输入功能（CSI00）从属接收时的寄存器设定内容例子 (2/2)

(f) 串行通道开始寄存器 m（SSm）…… 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC）…… 这是 CSI00 从属通道（单元 0 的通道 0）的  $\overline{\text{SSI00}}$  引脚的控制。

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

0:  $\overline{\text{SSI00}}$  引脚的输入值无效  
1:  $\overline{\text{SSI00}}$  引脚的输入值有效

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在从属发送模式中为固定设定。☐ : 不能设定（设定初始值）。  
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-86 从属接收的初始设定步骤

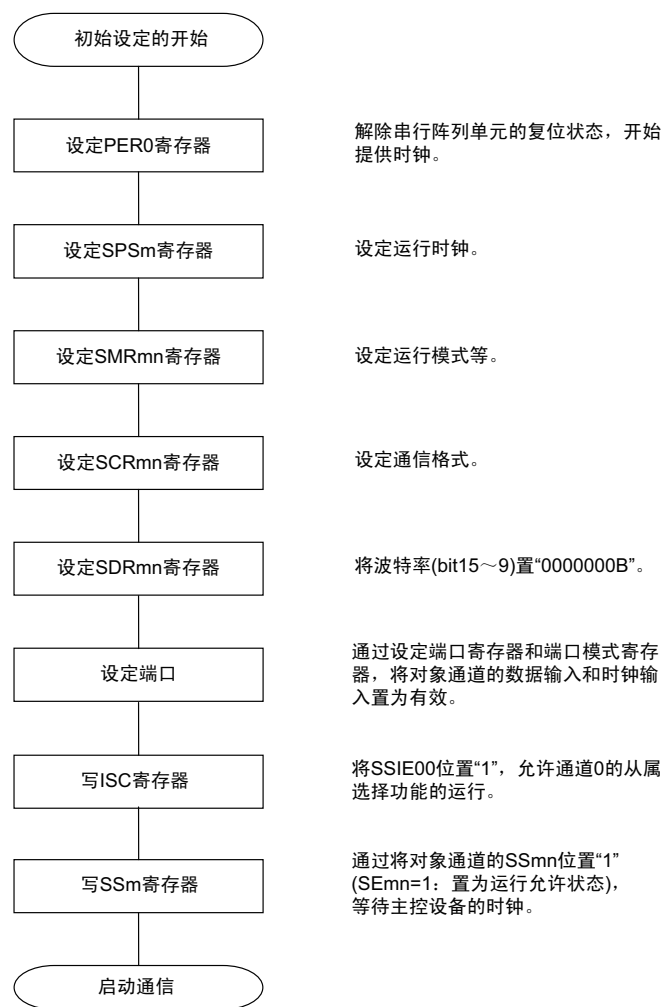
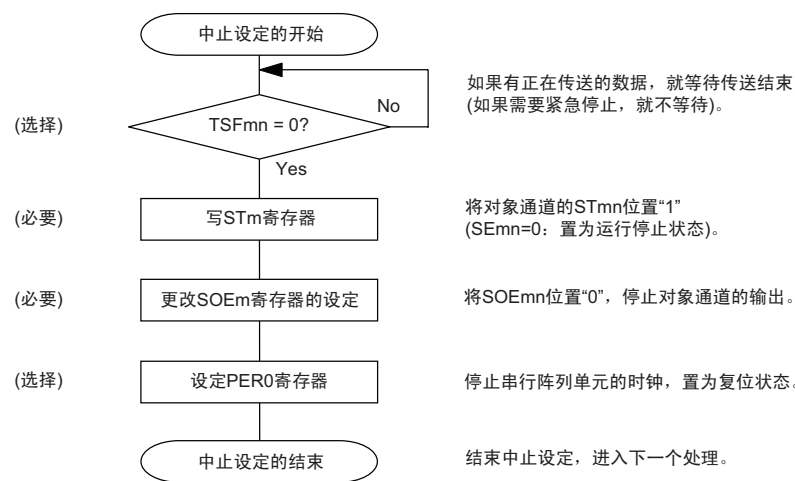
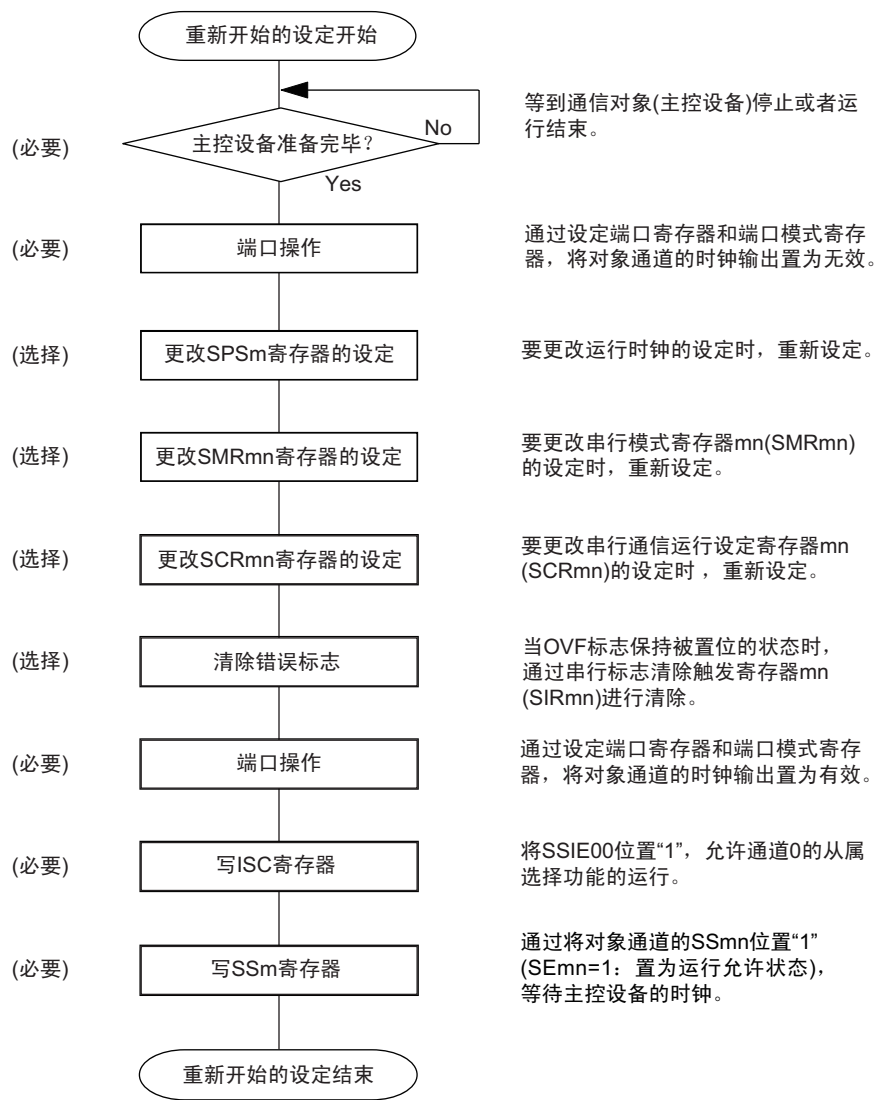


图 11-87 从属接收的中止步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

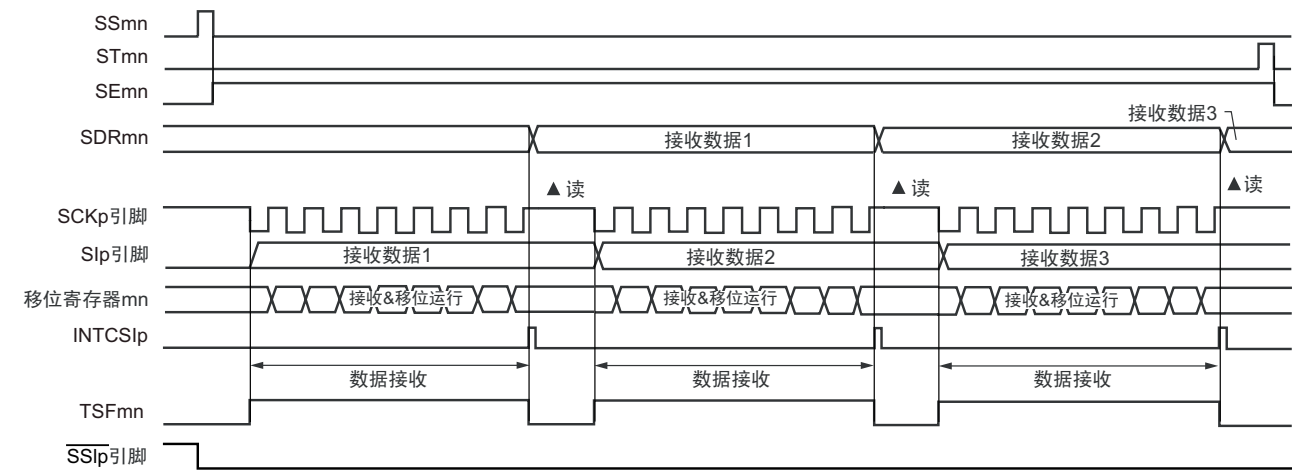
图 11-88 重新开始从属接收的设定步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

(3) 处理流程（单次接收模式）

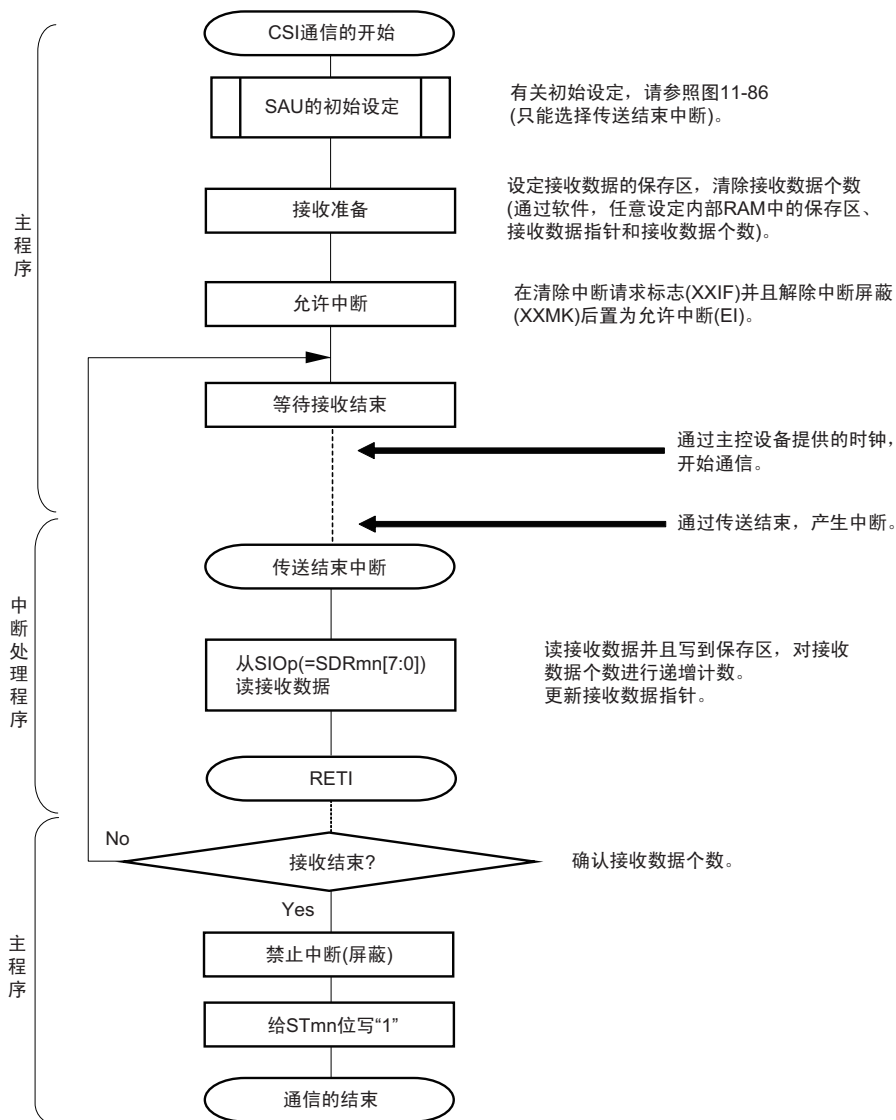
图 11-89 从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)



图 11-90 从属接收（单次接收模式）的流程图



### 11.6.3 从属的发送和接收

从属发送和接收是指在从其他设备输入传送时钟的状态下 R7F0C010 和其他设备进行数据发送和接收的运行。

从属选择输入功能	CSI00
对象通道	SAU0 的通道 0
使用的引脚	SCK00、SI00、SO00、SSI00
中断	INTCSI00 可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。
错误检测标志	只有溢出错误检测标志（OVFmn）。
传送数据长度	7 位或者 8 位
传送速率	Max. $f_{MCK}/6[\text{Hz}]$ 注 1、2
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> <li>• DAPmn=0: 在串行时钟开始运行时，开始数据输入 / 输出。</li> <li>• DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输入 / 输出。</li> </ul>
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> <li>• CKPmn=0: 正相</li> <li>• CKPmn=1: 反相</li> </ul>
数据方向	MSB 优先或者 LSB 优先
从属选择输入功能	可选择从属选择输入功能的运行。

注 1. 因为在内部对 SCK00 引脚输入的外部串行时钟进行采样后使用，所以最大传送速率为  $f_{MCK}/6[\text{Hz}]$ 。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

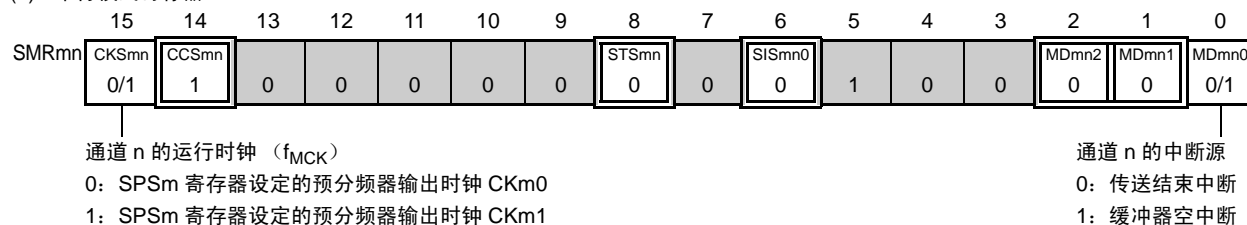
备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0)

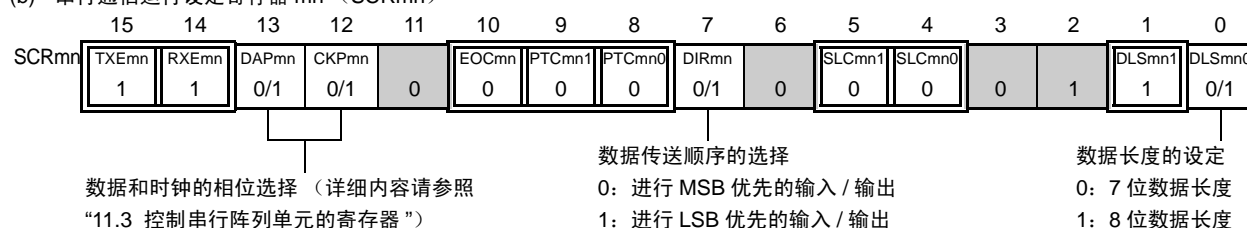
## (1) 寄存器的设定

图 11-91 从属选择输入功能 (CSI00) 从属发送和接收时的寄存器设定内容例子 (1/2)

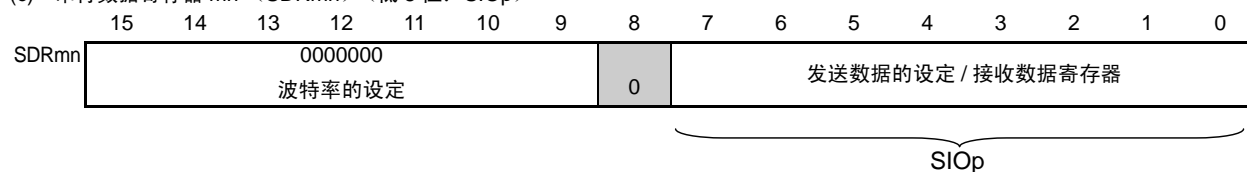
## (a) 串行模式寄存器 mn (SMRmn)



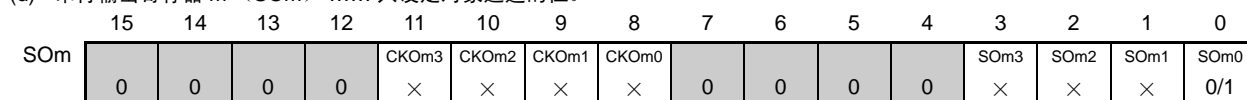
## (b) 串行通信运行设定寄存器 mn (SCRmn)



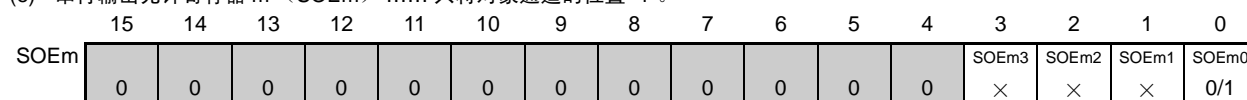
## (c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



## (d) 串行输出寄存器 m (SOm) ..... 只设定对象通道的位。



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



注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2.   : 在 CSI 从属发送和接收模式中为固定设定。  : 不能设定 (设定初始值)。  
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。  
 0/1: 根据用户的用途置“0”或者“1”。

图 11-91 从属选择输入功能（CSI00）从属发送和接收时的寄存器设定内容例子 (2/2)

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) 输入切换控制寄存器（ISC）..... 这是 CSI00 从属通道（单元 0 的通道 0）的  $\overline{\text{SSI00}}$  引脚的控制。

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

0:  $\overline{\text{SSI00}}$  引脚的输入值无效  
1:  $\overline{\text{SSI00}}$  引脚的输入值有效

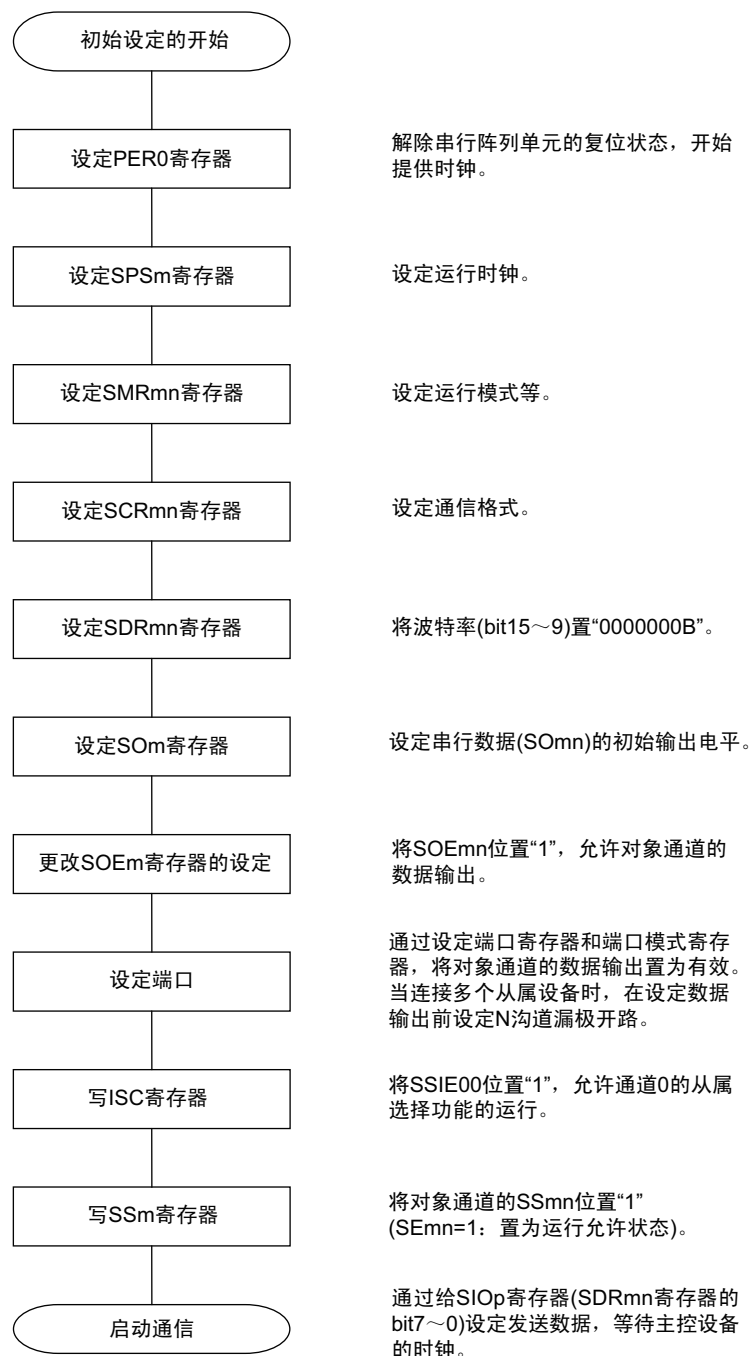
注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

2. ☐ : 在 CSI 从属发送和接收模式中为固定设定。 ☐ : 不能设定（设定初始值）。  
 × : 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

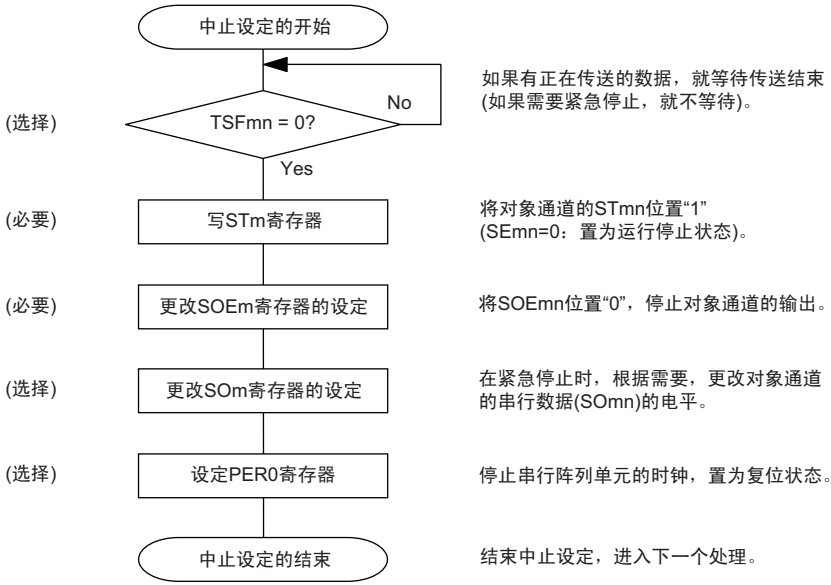
图 11-92 从属发送和接收的初始设定步骤



注意 在主导设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

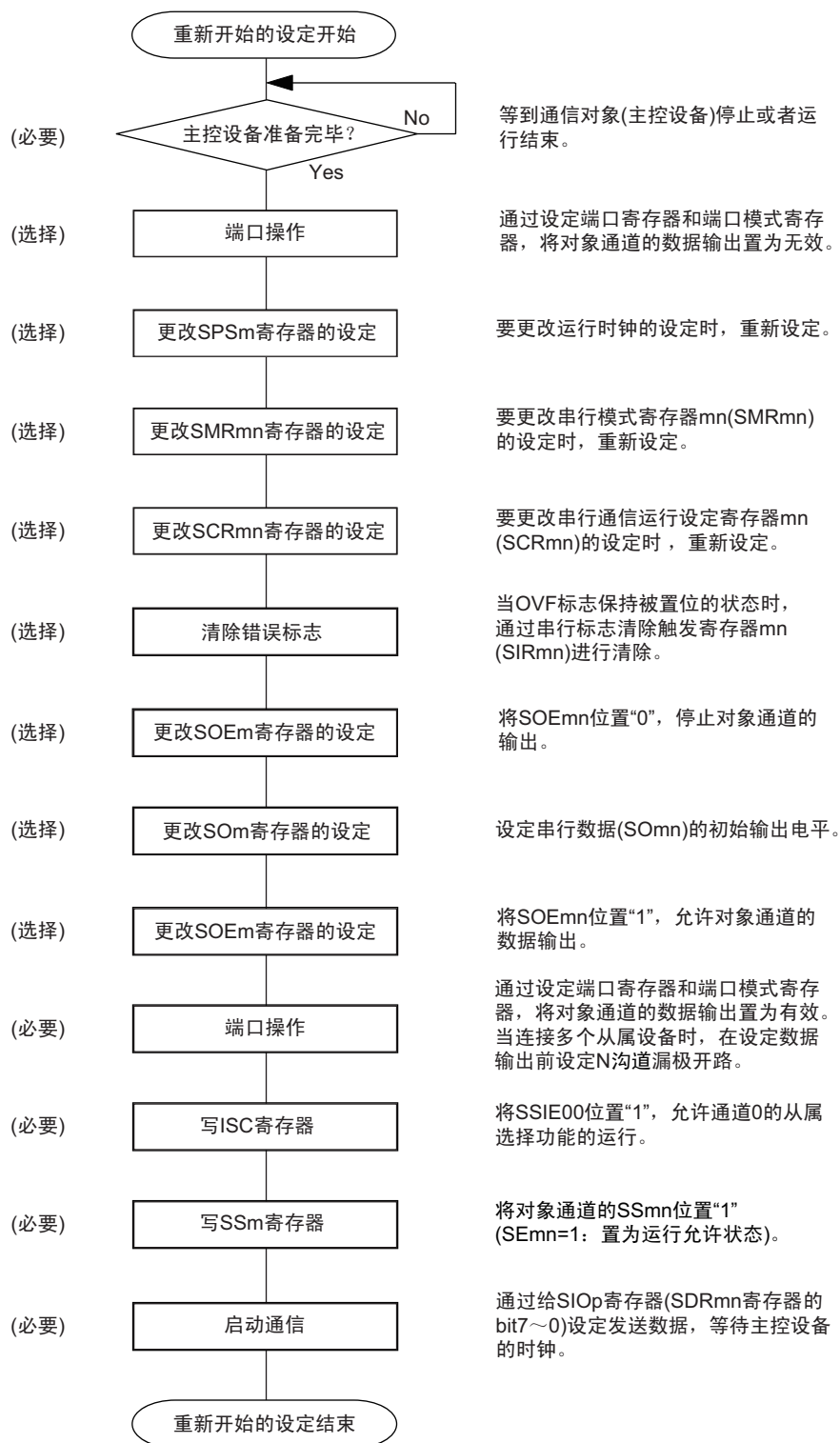
备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 11-93 从属发送和接收的中止步骤



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 11-94 重新开始从属发送和接收的设定步骤

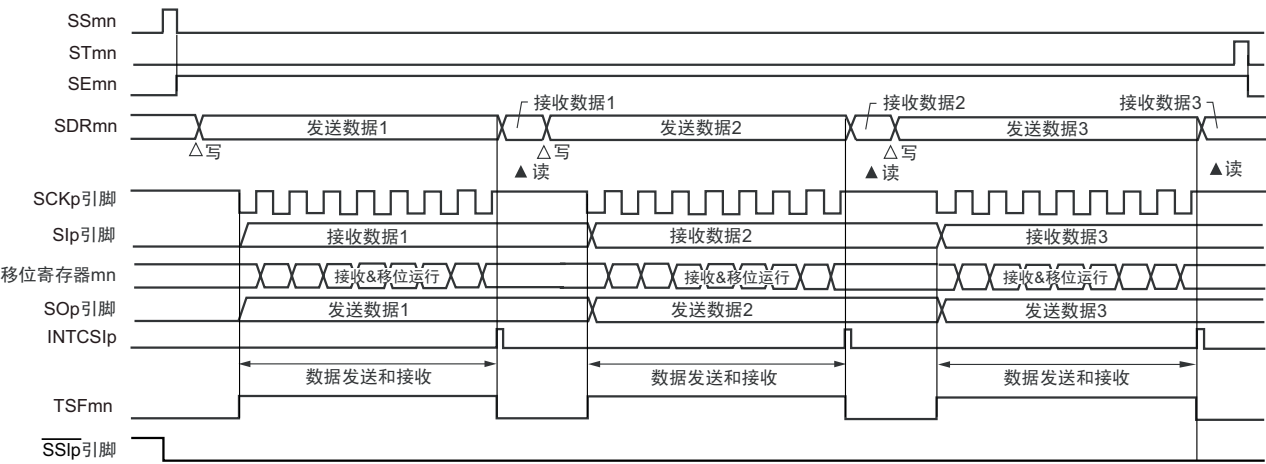


注意 1. 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始设定。

(3) 处理流程（单次发送和接收模式）

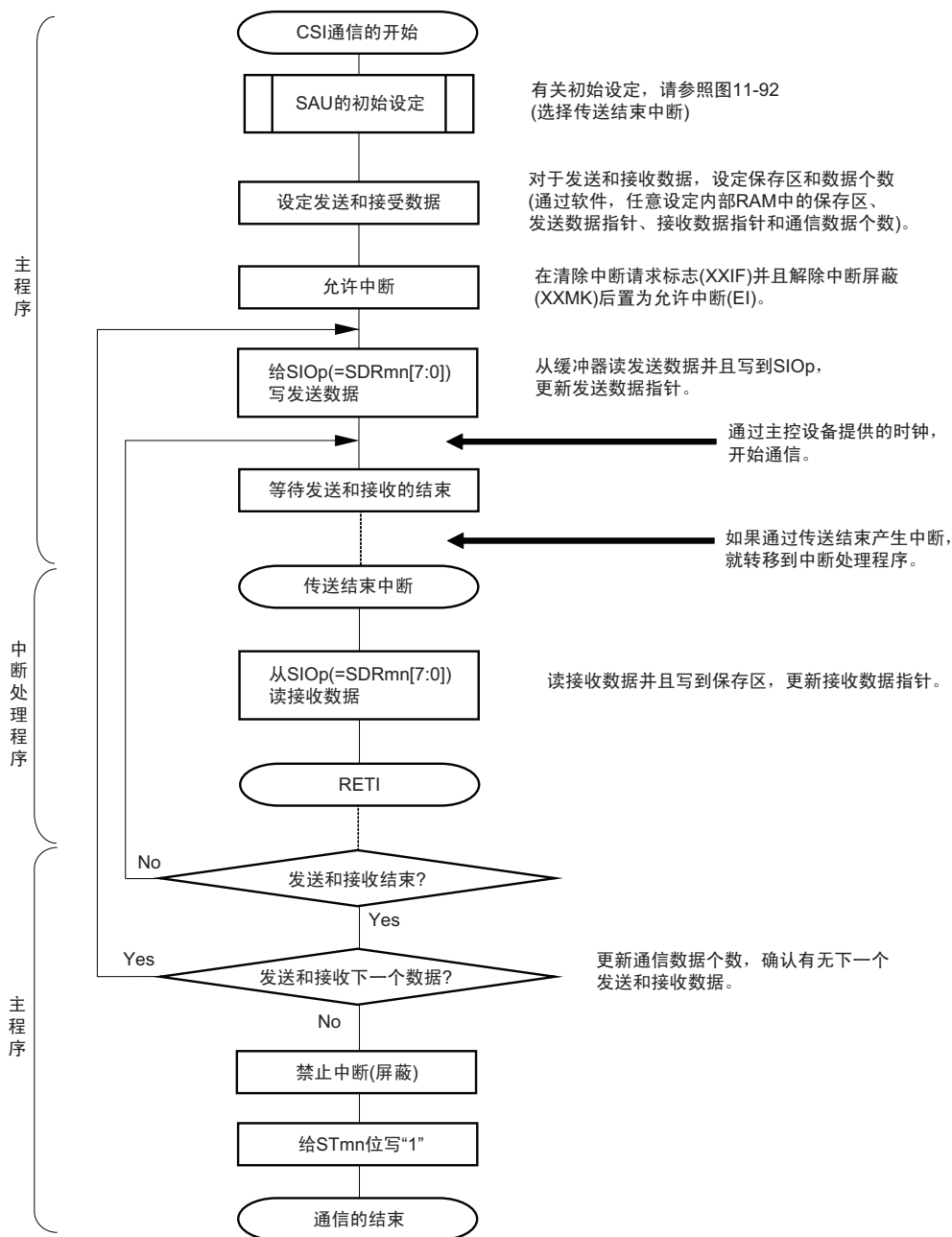
图 11-95 从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)



图 11-96 从属发送和接收（单次发送和接收模式）的流程图

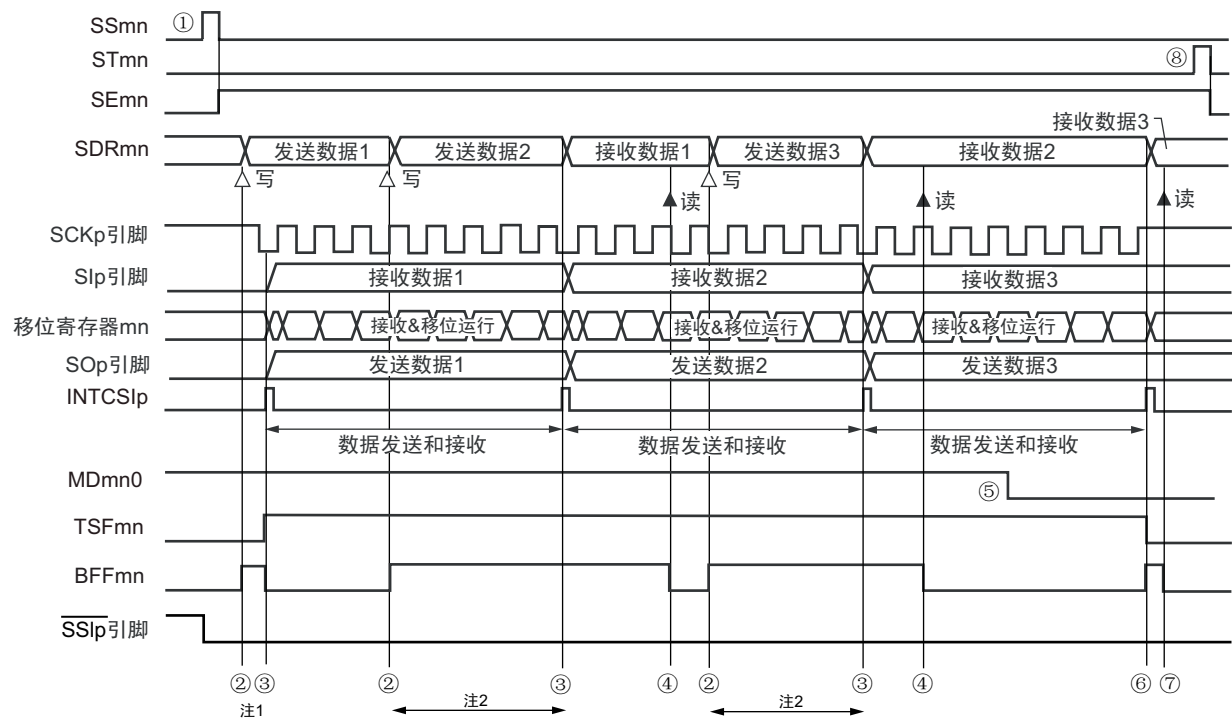


**注意** 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

**备注** m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

(4) 处理流程（连续发送和接收模式）

图 11-97 从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



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

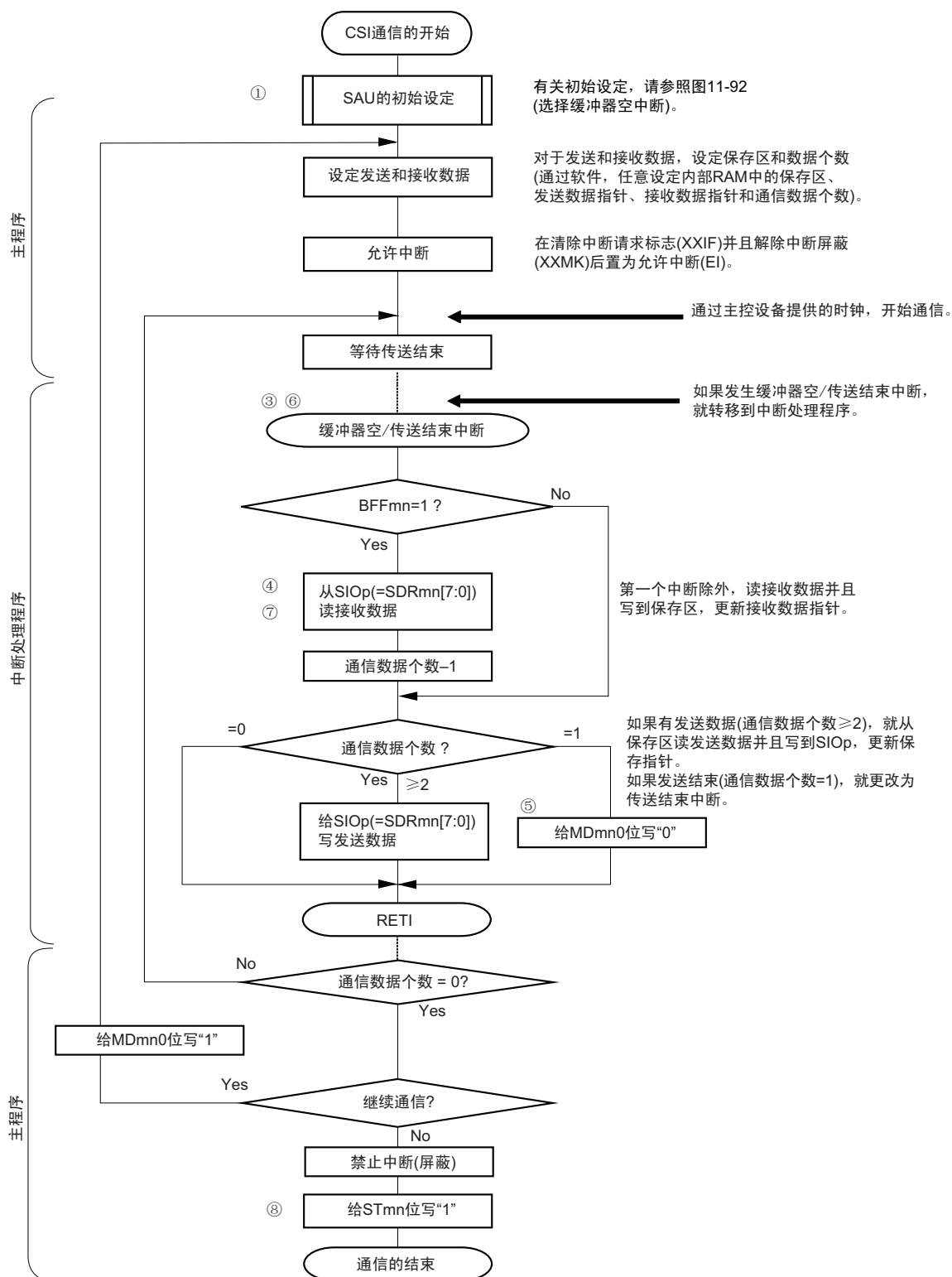
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

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

备注 1. 图中的①～⑧对应“图 11-98 从属发送和接收（连续发送和接收模式）的流程图”中的①～⑧。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

图 11-98 从属发送和接收（连续发送和接收模式）的流程图



注意 在主控设备开始输出时钟前, 必须给 SIop 寄存器设定发送数据。

备注 1. 图中的①~⑧对应“图 11-97 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

2. m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

## 11.6.4 传送时钟频率的计算

从属选择输入功能（CSI00）通信的传送时钟频率能用以下计算式进行计算。

## (1) 从属设备

(传送时钟频率) = { 主控设备提供的串行时钟 (SCK) 频率 } 注	[Hz]
---------------------------------------	------

注 容许的最大传送时钟频率为  $f_{MCK}/6$ 。

备注 m: 单元号 (m=0) n: 通道号 (n=0) p: CSI 号 (p=00)

表 11-3 从属选择输入功能运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟（f <sub>MCK</sub> ）注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f <sub>CLK</sub> =32MHz 运行时
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	32MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	16MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	8MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	4MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	2MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	1MHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	500kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	250kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	125kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	62.5kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	31.25kHz
	X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	15.63kHz
	X	X	X	X	1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	7.81kHz
	X	X	X	X	1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	3.9kHz
	X	X	X	X	1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	1.95kHz
	X	X	X	X	1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	977Hz

注 要更改被选择为  $f_{CLK}$  的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（SAU）的运行（串行通道停止寄存器 m（STm）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0) n: 通道号 (n=0)

11.6.5 在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤

在从属选择输入功能的时钟同步串行通信过程中发生错误时的处理步骤如图 11-99 所示。

图 11-99 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误的种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

备注 m: 单元号（m=0） n: 通道号（n=0）

## 11.7 UART (UART0) 通信的运行

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

[ 数据的发送和接收 ]

- 7 位、8 位或者 9 位的数据长度
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定 (选择是否将电平反相)
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位检查功能

[ 中断功能 ]

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

[ 错误检测标志 ]

- 帧错误、奇偶校验错误、溢出错误

UART0 接收支持 SNOOZE 模式。SNOOZE 模式功能是指: 如果在 STOP 模式的状态下检测到 RxD 的输入, 就不需要 CPU 运行而接收数据。

UART0 使用 SAU0 的通道 0 和通道 1。

单元	通道	用作 CSI	用作 UART
0	0	CSI00 (支持从属选择输入功能)	UART0
	1	—	

在单元 0 的通道 0 和通道 1 使用 UART0 时, 不能使用 CSI00。

**注意** 当用作 UART 时, 发送方 (偶数通道) 或者接收方 (奇数通道) 只能用作 UART。

UART 有以下 2 种通信运行:

- UART 发送 (参照 11.7.1)
- UART 接收 (参照 11.7.2)

## 11.7.1 UART 发送

UART 发送是 R7F0C010 将数据异步发送到其他设备的运行。

UART 使用的 2 个通道中的偶数通道用于 UART 发送。

UART	UART0
对象通道	SAU0 的通道 0
使用的引脚	TxD0
中断	INTST0 可选择发送结束中断（单次发送模式）或者缓冲器空中断（连续发送模式）。
错误检测标志	无
传送数据长度	7 位、8 位或者 9 位
传送速率	Max. $f_{MCK}/6[\text{bps}]$ ( $\text{SDRmn}[15:9] \geq 2$ )、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$ 注
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> <li>• 无奇偶校验位。</li> <li>• 附加零校验。</li> <li>• 附加偶校验。</li> <li>• 附加奇校验。</li> </ul>
停止位	可选择以下内容： <ul style="list-style-type: none"> <li>• 附加 1 位。</li> <li>• 附加 2 位。</li> </ul>
数据方向	MSB 优先或者 LSB 优先

注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

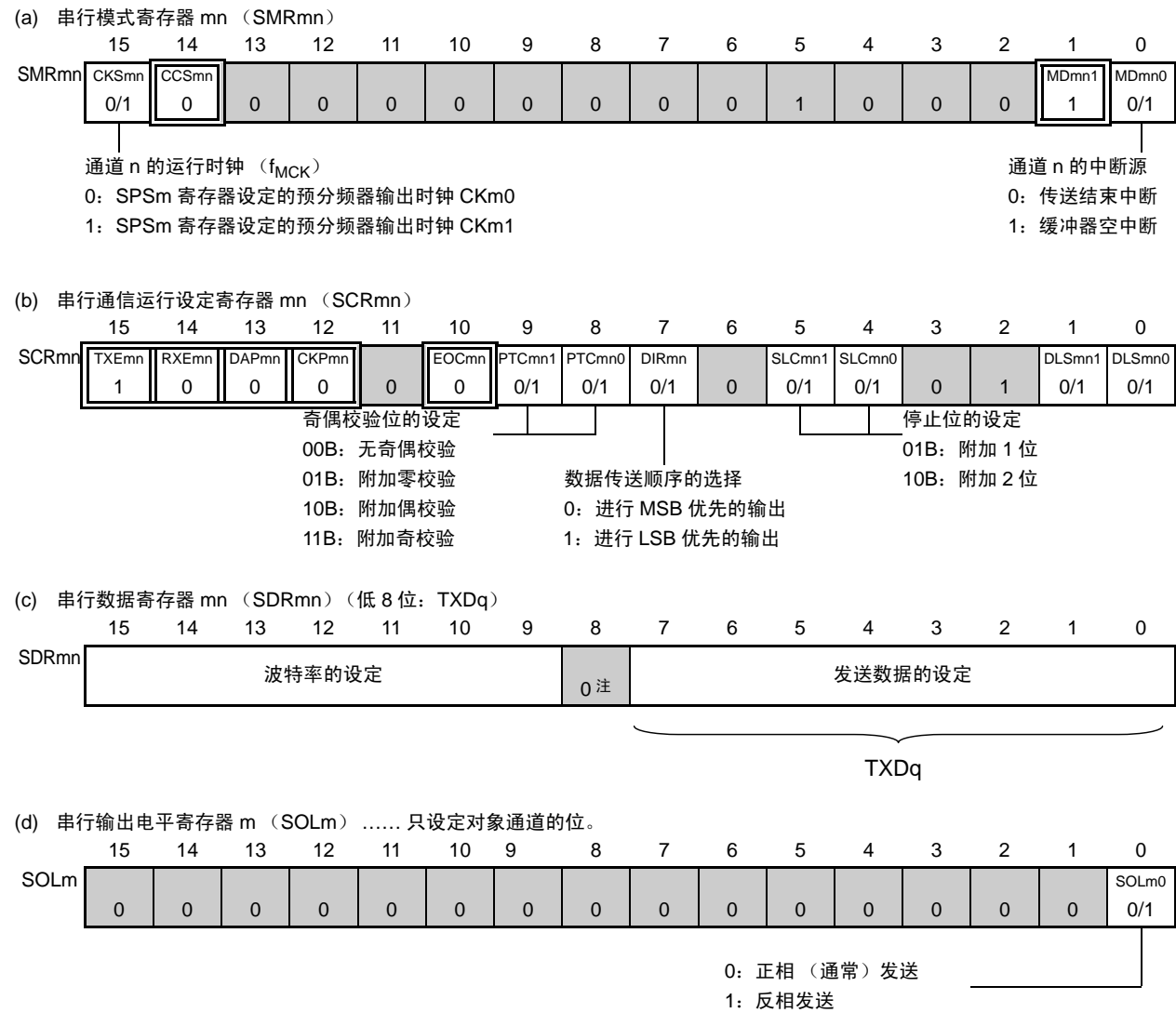
备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

$f_{CLK}$ : 系统时钟频率

2. m: 单元号 (m=0) n: 通道号 (n=0) mn=00

(1) 寄存器的设定

图 11-100 UART（UART0）的 UART 发送时的寄存器设定内容例子 (1/2)



注 当进行 9 位数据长度的通信时，SDRm0 寄存器的 bit0 ~ 8 为发送数据的设定区。

备注 1. m: 单元号 (m=0) n: 通道号 (n=0) q: UART 号 (q=0) mn=00

2. □: 在 UART 发送模式中为固定设定。■: 不能设定 (设定初始值)。  
×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。  
0/1: 根据用户的用途置“0”或者“1”。



图 11-100 UART（UART0）的 UART 发送时的寄存器设定内容例子 (2/2)

(e) 串行输出寄存器 m（SOm） ..... 只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm								CKOm0								SOm0
	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	0/1 注

0: 串行数据输出值为“0”  
1: 串行数据输出值为“1”

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm																SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1

注 在开始发送前，当对应通道的 SOLmn 位为“0”时，必须置“1”；当对应通道的 SOLmn 位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注 1. m: 单元号（m=0） n: 通道号（n=0） q: UART 号（q=0） mn=00

2. 

<div></div>	: 在 UART 发送模式中为固定设定。	<div></div>	: 不能设定（设定初始值）。
×	: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。		
0/1	: 根据用户的用途置“0”或者“1”。		

(2) 操作步骤

图 11-101 UART 发送的初始设定步骤

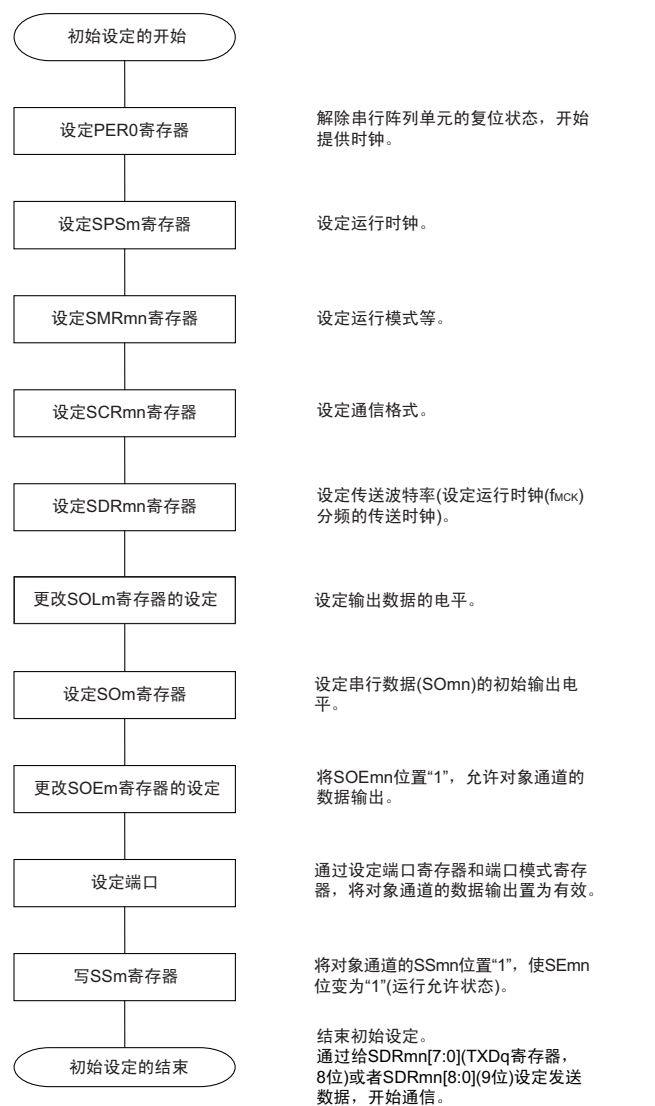


图 11-102 UART 发送的中止步骤

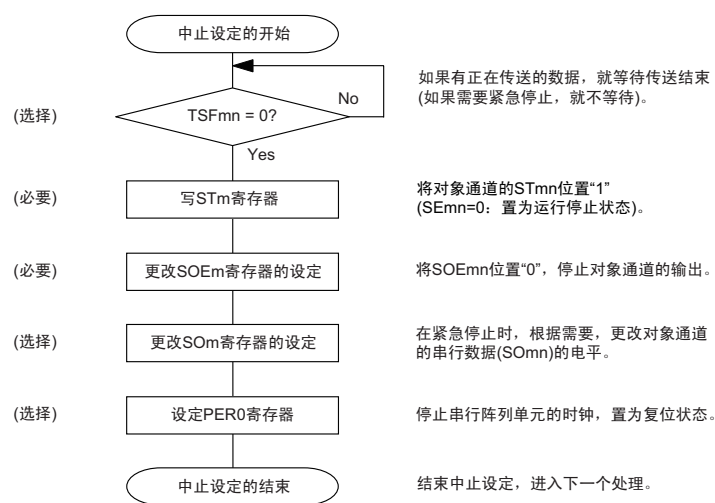
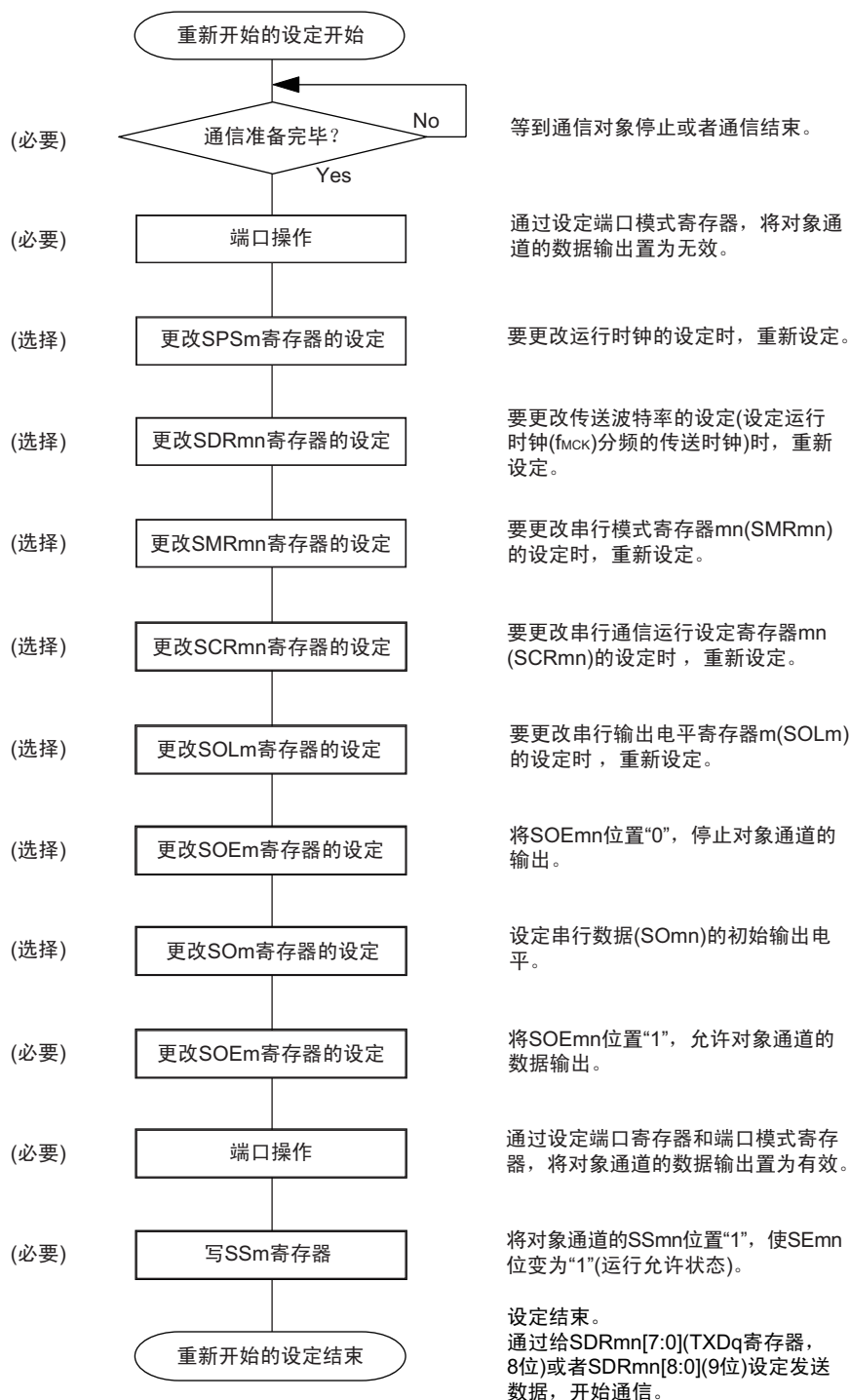


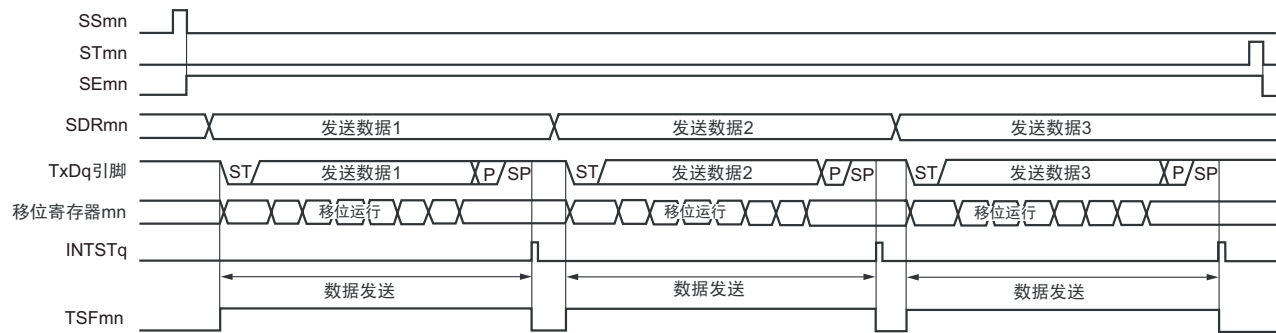
图 11-103 重新开始 UART 发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始的设定。

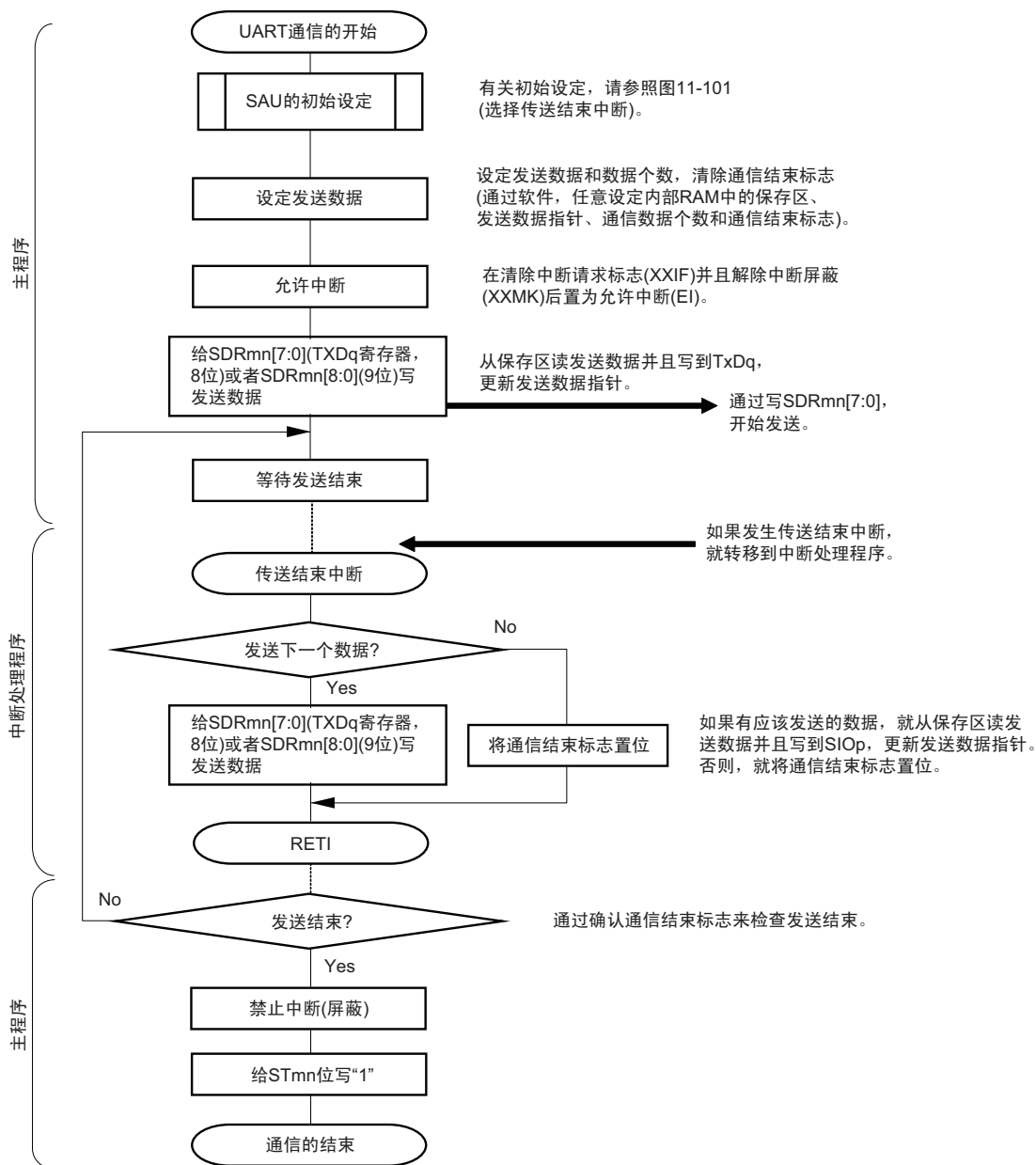
(3) 处理流程（单次发送模式）

图 11-104 UART 发送（单次发送模式）的时序图



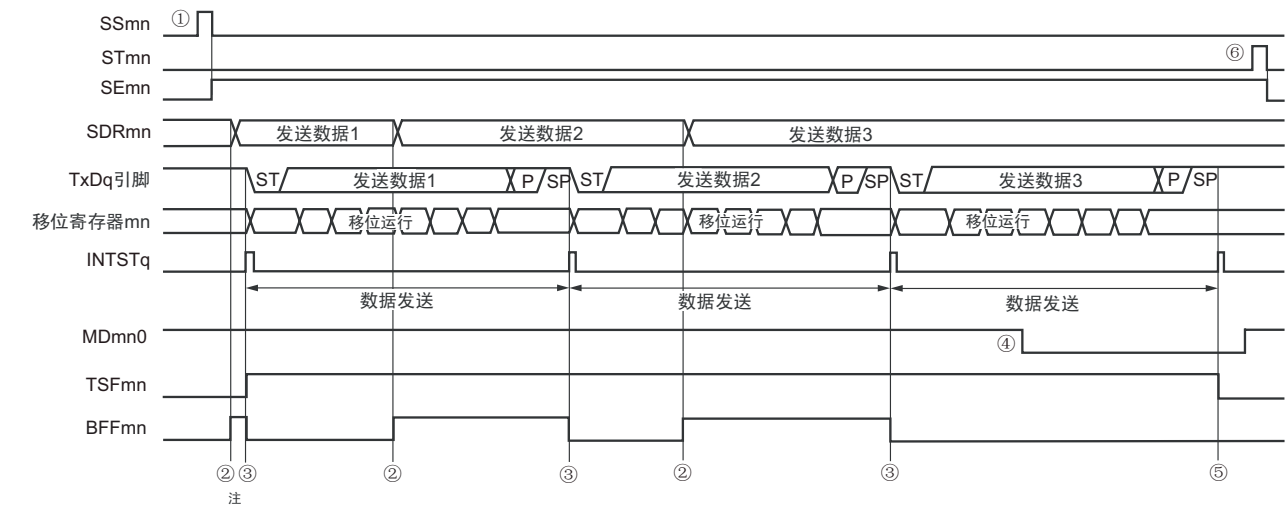
备注 m: 单元号 (m=0) n: 通道号 (n=0) q: UART 号 (q=0) mn=00

图 11-105 UART 发送（单次发送模式）的流程图



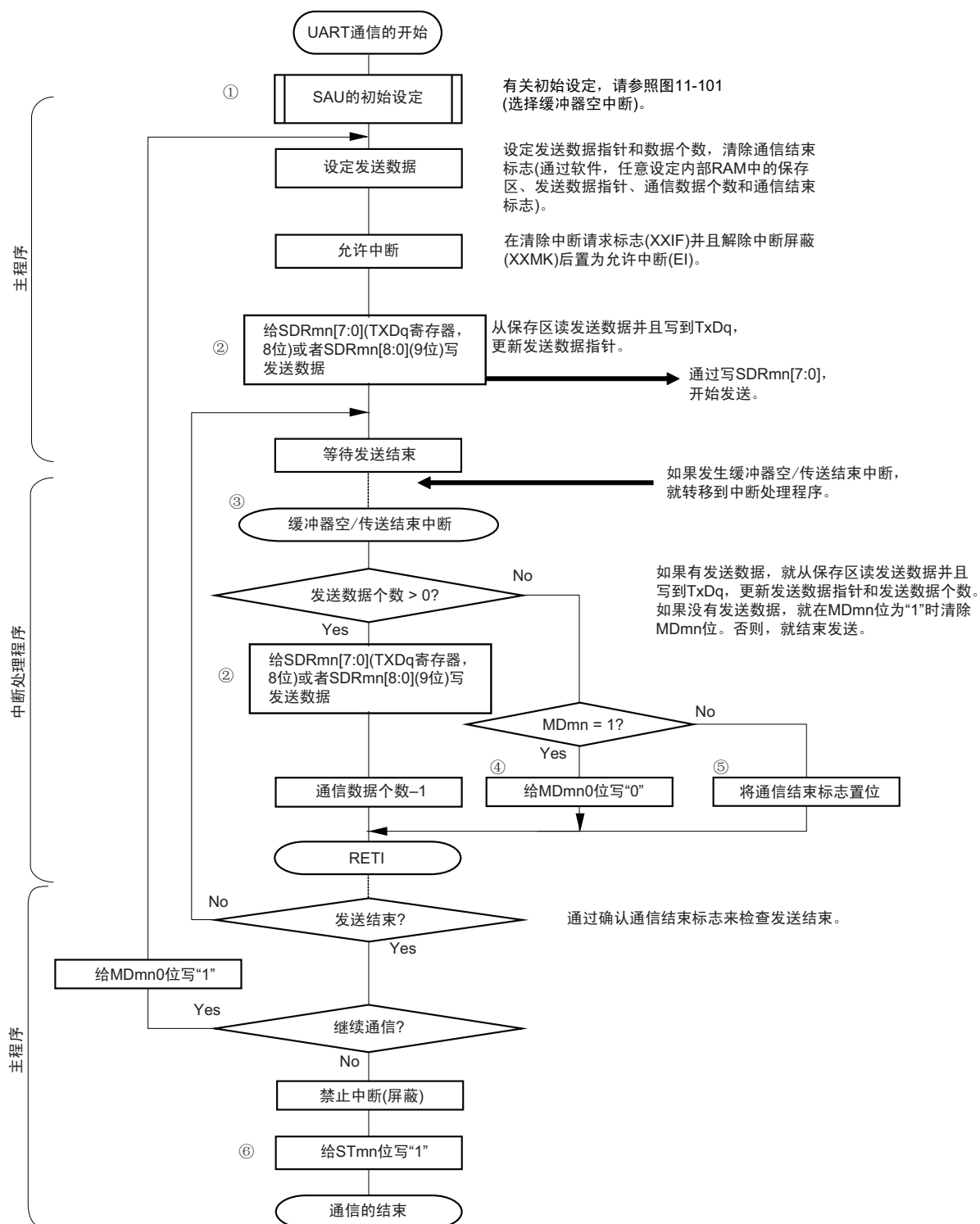
(4) 处理流程（连续发送模式）

图 11-106 UART 发送（连续发送模式）的时序图



- 注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
- 注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。
- 备注 m: 单元号（m=0） n: 通道号（n=0） q: UART 号（q=0） mn=00

图 11-107 UART 发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 11-106 UART 发送（连续发送模式）的时序图”中的①~⑥。

### 11.7.2 UART 接收

UART 接收是 R7F0C010 从其他设备异步接收数据的运行。

UART 使用的 2 个通道中的奇数通道用于 UART 接收。但是，需要设定奇数通道和偶数通道的 SMR 寄存器。

UART	UART0
对象通道	SAU0 的通道 1
使用的引脚	RxD0
中断	INTSR0 只限于传送结束中断（禁止设定缓冲器空中断）。
错误中断	INTSRE0
错误检测标志	<ul style="list-style-type: none"> <li>帧错误检测标志（FEFmn）</li> <li>奇偶校验错误检测标志（PEFmn）</li> <li>溢出错检测标志（OVFmn）</li> </ul>
传送数据长度	7 位、8 位或者 9 位
传送速率	Max. $f_{MCK}/6[\text{bps}]$ ( $\text{SDRmn}[15:9] \geq 2$ )、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$ 注
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> <li>无奇偶校验位（无奇偶校验）。</li> <li>不判断奇偶校验（零校验）。</li> <li>偶校验</li> <li>奇校验</li> </ul>
停止位	检查 1 位。
数据方向	MSB 优先或者 LSB 优先

注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性”）的范围内使用。

备注 1.  $f_{MCK}$ : 对象通道的运行时钟频率

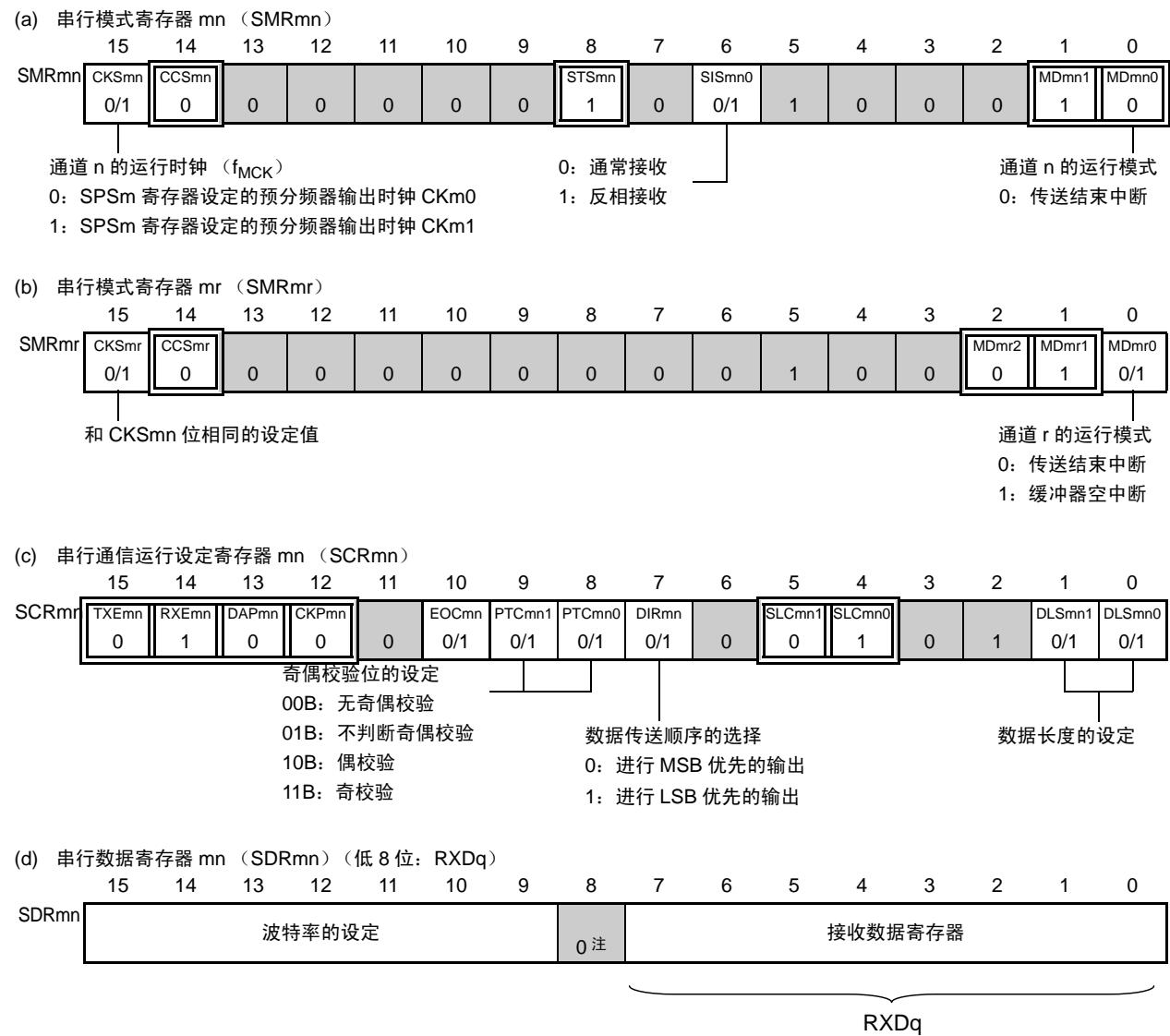
$f_{CLK}$ : 系统时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=1) mn=01



(1) 寄存器的设定

图 11-108 UART（UART0）的 UART 接收时的寄存器设定内容例子 (1/2)



- 注 当进行 9 位数据长度的通信时，SDRm1 寄存器的 bit0 ~ 8 为发送数据的设定区。
- 注意 在 UART 接收时，还必须将与通道 n 成对的通道 r 的 SMRmr 寄存器设定为 UART 发送模式。
- 备注 1. m: 单元号（m=0） n: 通道号（n=1） mn=01 r: 通道号（r=0） q: UART 号（q=0）
2.   : 在 UART 接收模式中为固定设定。  : 不能设定（设定初始值）。  
×: 这是在此模式中不能使用的位（在其他模式中也不使用的情况下，设定初始值）。  
0/1: 根据用户的用途置“0”或者“1”。

图 11-108 UART (UART0) 的 UART 接收时的寄存器设定内容例子 (2/2)

(e) 串行输出寄存器 m (SOm) ..... 在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm								CKOm0								SOm0
	0	0	0	0	0	0	0	×	0	0	0	0	0	0	0	×

(f) 串行输出允许寄存器 m (SOEm) ..... 在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm																SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	×

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×

备注 1. m: 单元号 (m=0) n: 通道号 (n=1) mn=01 r: 通道号 (r=0) q: UART 号 (q=0)

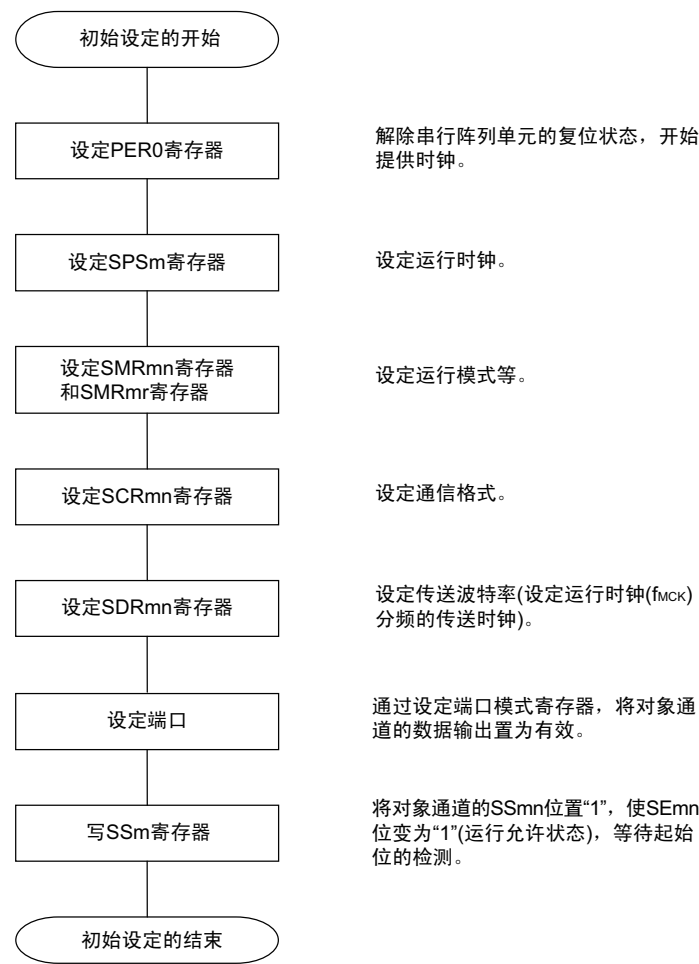
2. ☐ : 在 UART 接收模式中为固定设定。 ☐ : 不能设定 (设定初始值)。

× : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。

0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-109 UART 接收的初始设定步骤



注意 必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个  $f_{MCK}$  时钟，然后将 SSmn 位置“1”。

图 11-110 UART 接收的中止步骤

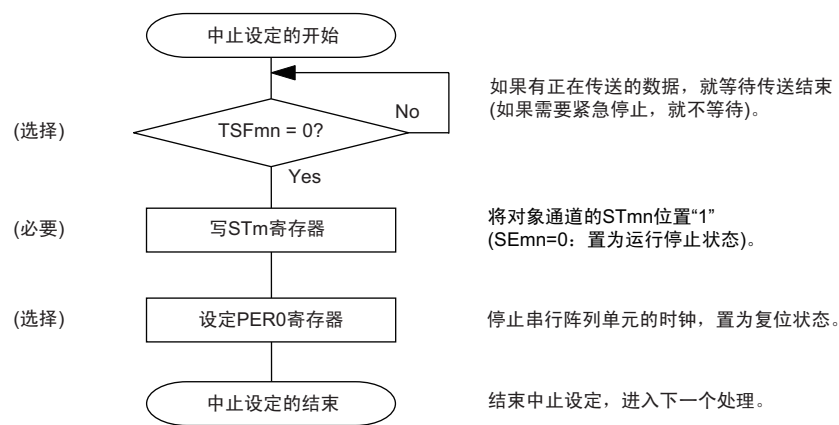
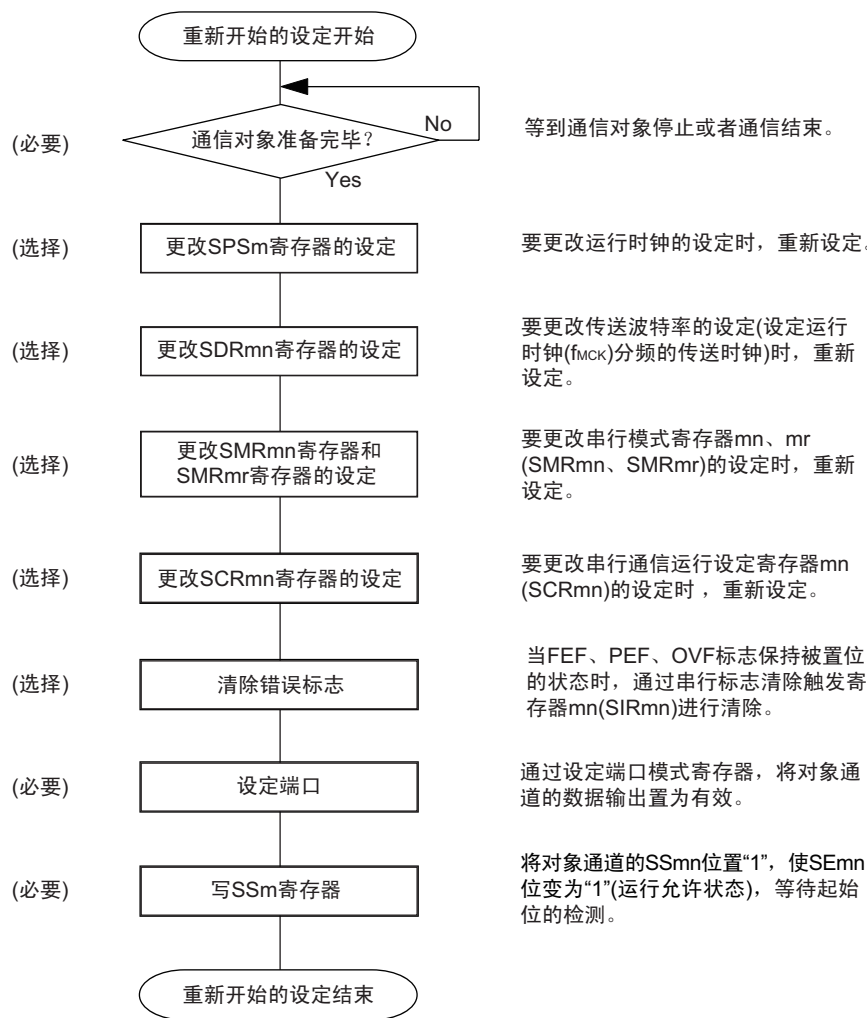


图 11-111 重新开始 UART 接收的设定步骤

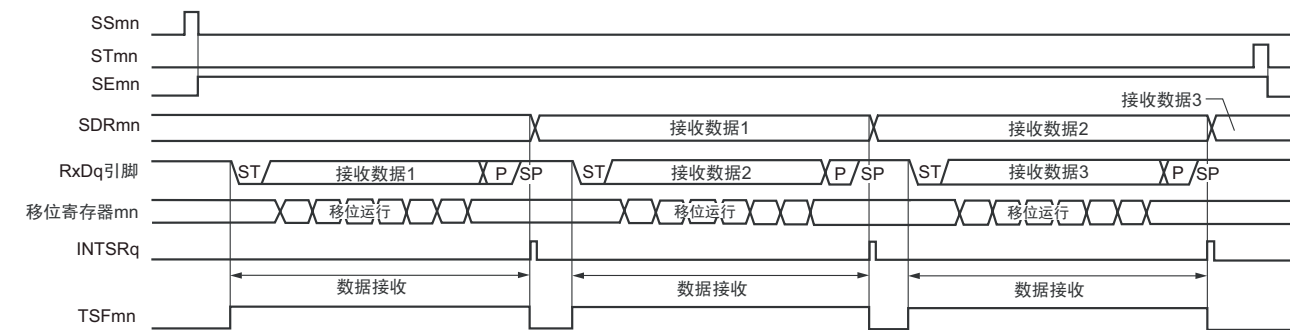


注意 必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个  $f_{MCK}$  时钟，然后将 SSmn 位置“1”。

备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始的设定。

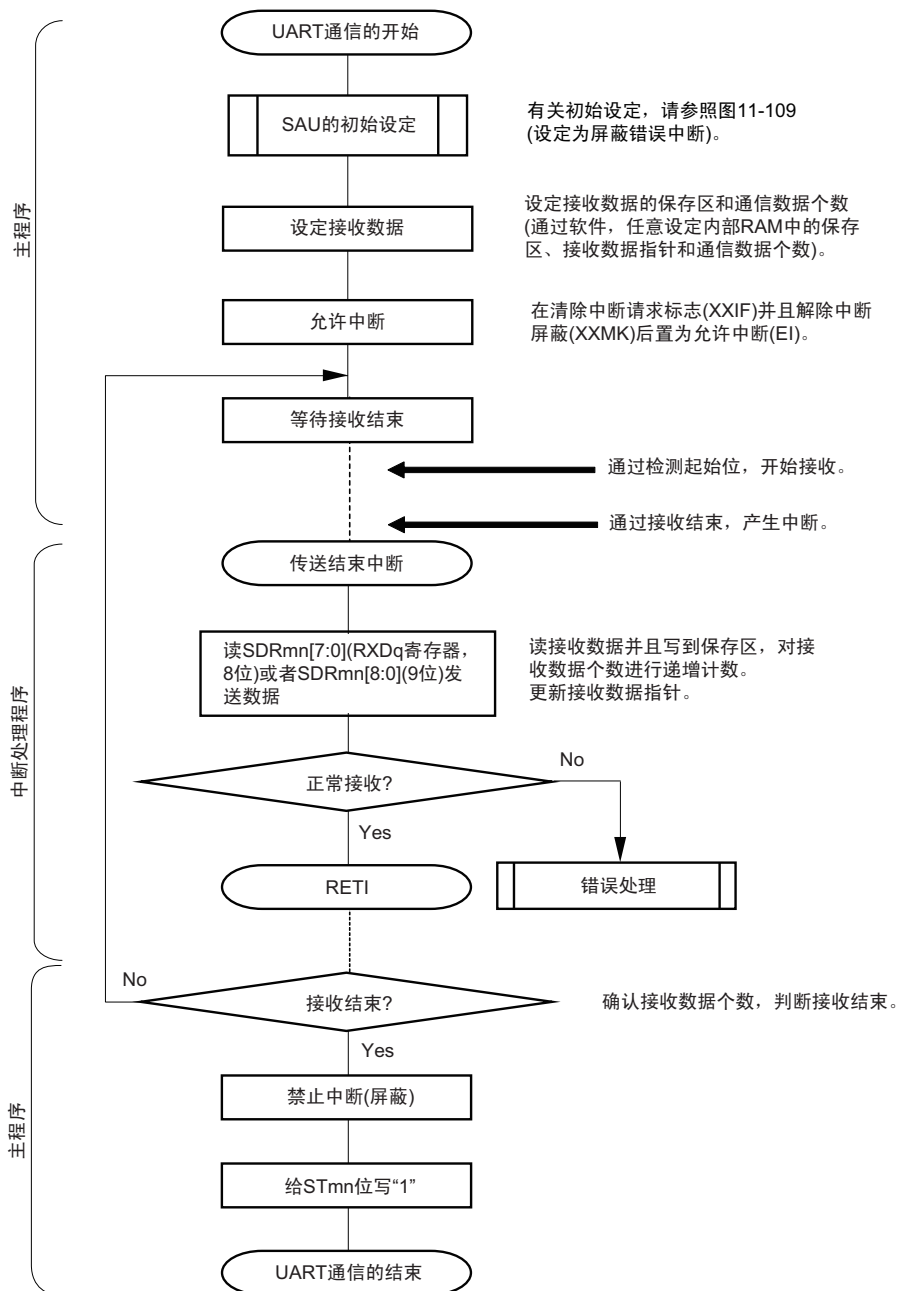
(3) 处理流程

图 11-112 UART 接收的时序图



备注 m: 单元号 (m=0) n: 通道号 (n=1) mn=01 r: 通道号 (r=0) q: UART 号 (q=0)

图 11-113 UART 接收的流程图



11.7.3 SNOOZE 模式功能

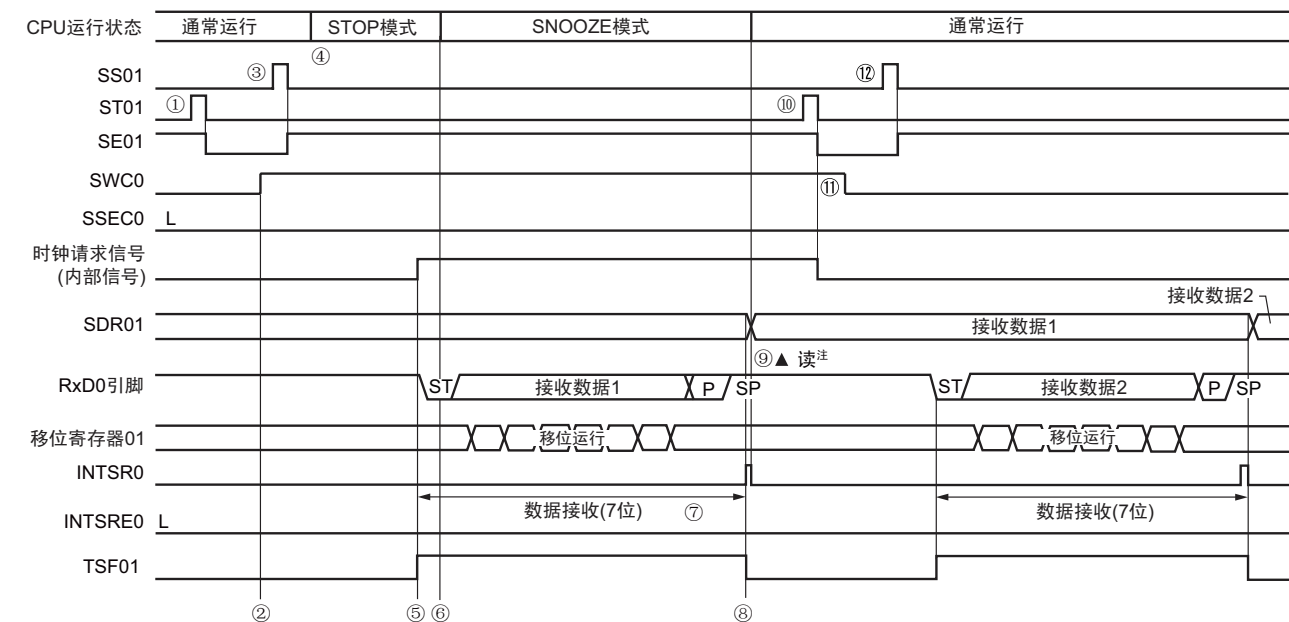
这是在 STOP 模式中通过检测 RxDq 引脚的输入使 UART 进行接收的模式。通常，在 STOP 模式中 UART 停止通信，但是使用此模式并且通过检测 RxDq 引脚的输入，能在 CPU 不运行的状态下进行 UART 的接收。

要使用 SNOOZE 模式功能时，必须在即将要转移到 STOP 模式前将串行待机控制寄存器 m（SSCm）的 SWCm 位置“1”。

- 注意 1. 只有在选择高速内部振荡器时钟作为 f<sub>CLK</sub> 时才能设定 SNOOZE 模式。  
2. 在 SNOOZE 模式中使用 UARTq 时，最大传送速率为 9600bps。

(1) SNOOZE 模式运行（通常运行）

图 11-114 SNOOZE 模式运行（通常运行）的时序图



注 必须在 SWCm 位为“1”的状态下读接收数据。

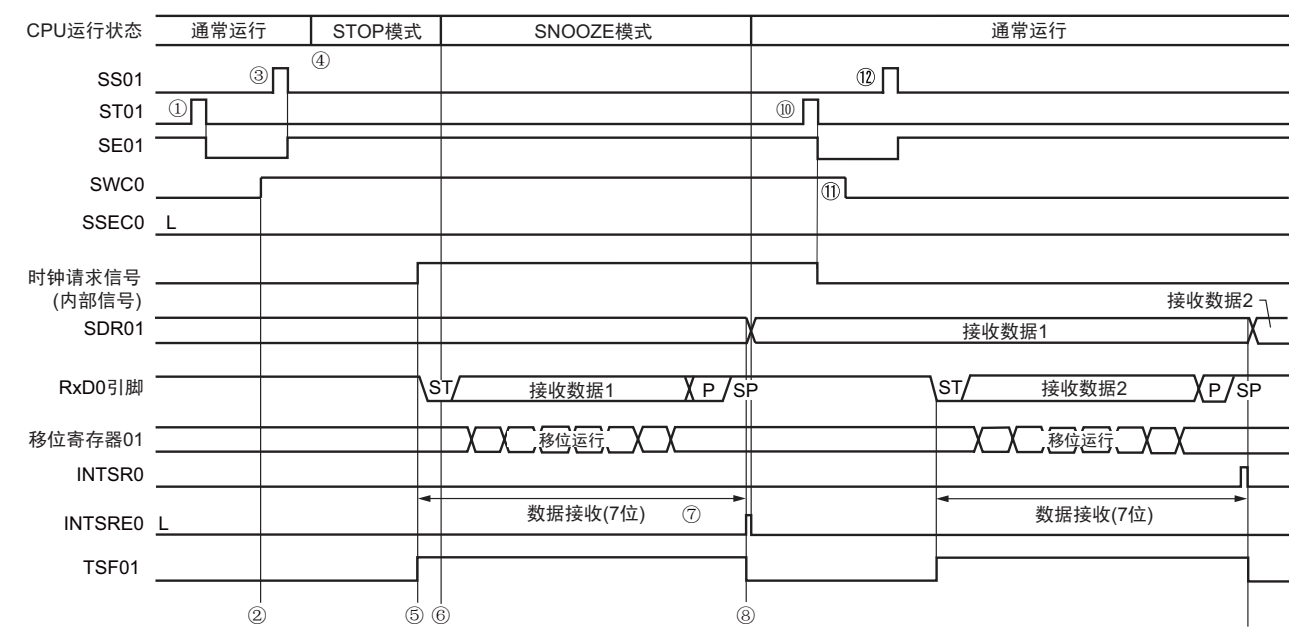
注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

- 备注 1. 图中的①～⑪对应“图 11-116 SNOOZE 模式运行（通常运行 / 异常运行①）的流程图”中的①～⑪。  
2. m=0, q=0

(2) SNOOZE 模式运行（异常运行①）

异常运行①是在 SSECm 位为“0”的状态下发生通信错误时的运行。  
因为 SSECm 位为“0”，所以在发生通信错误时产生错误中断（INTSRE0）。

图 11-115 SNOOZE 模式运行（异常运行①）的时序图



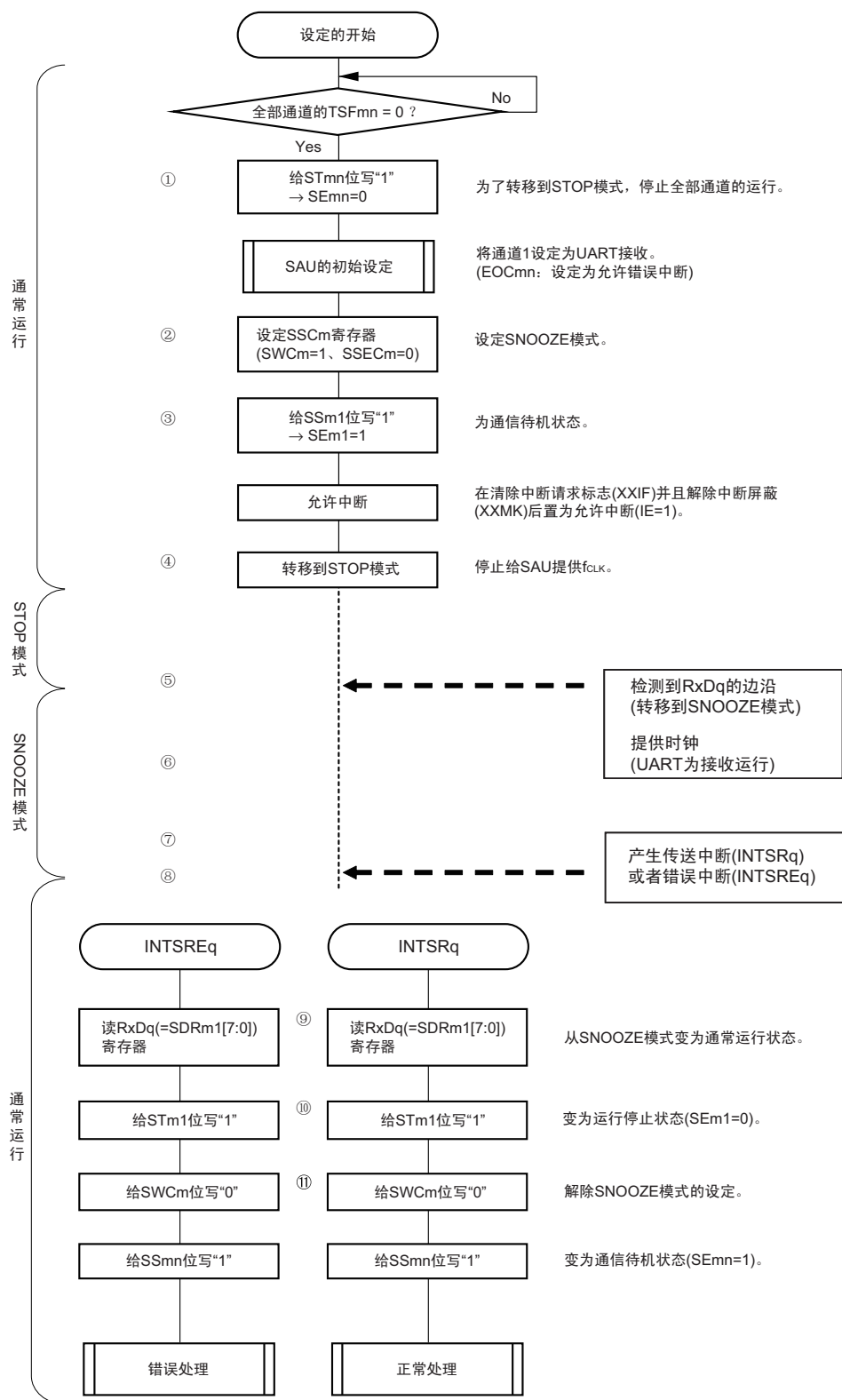
注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。

备注 1. 图中的①～⑪对应“图 11-116 SNOOZE 模式运行（通常运行 / 异常运行①）的流程图”中的①～⑪。

2. m=0, q=0



图 11-116 SNOOZE 模式运行（通常运行 / 异常运行①）的流程图



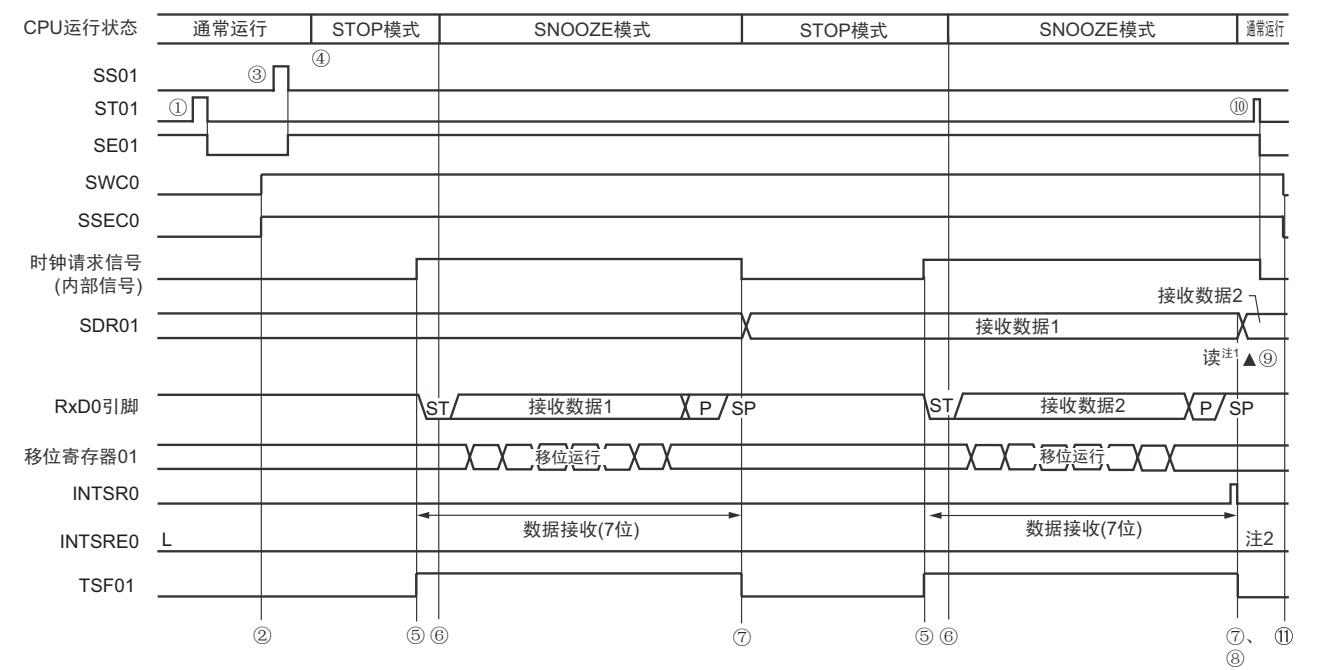
备注 1. 图中的①~⑪对应“图 11-114 SNOOZE 模式运行（通常运行）的时序图”和“图 11-115 SNOOZE 模式运行（异常运行①）的时序图”中的①~⑪。

2. m=0, q=0

(3) SNOOZE 模式运行（异常运行②）

异常运行②是在 SSECm 位为“1”的状态下发生通信错误时的运行。  
因为 SSECm 位为“1”，所以在发生通信错误时不产生错误中断（INTSREq）。

图 11-117 SNOOZE 模式运行（异常运行②）的时序图

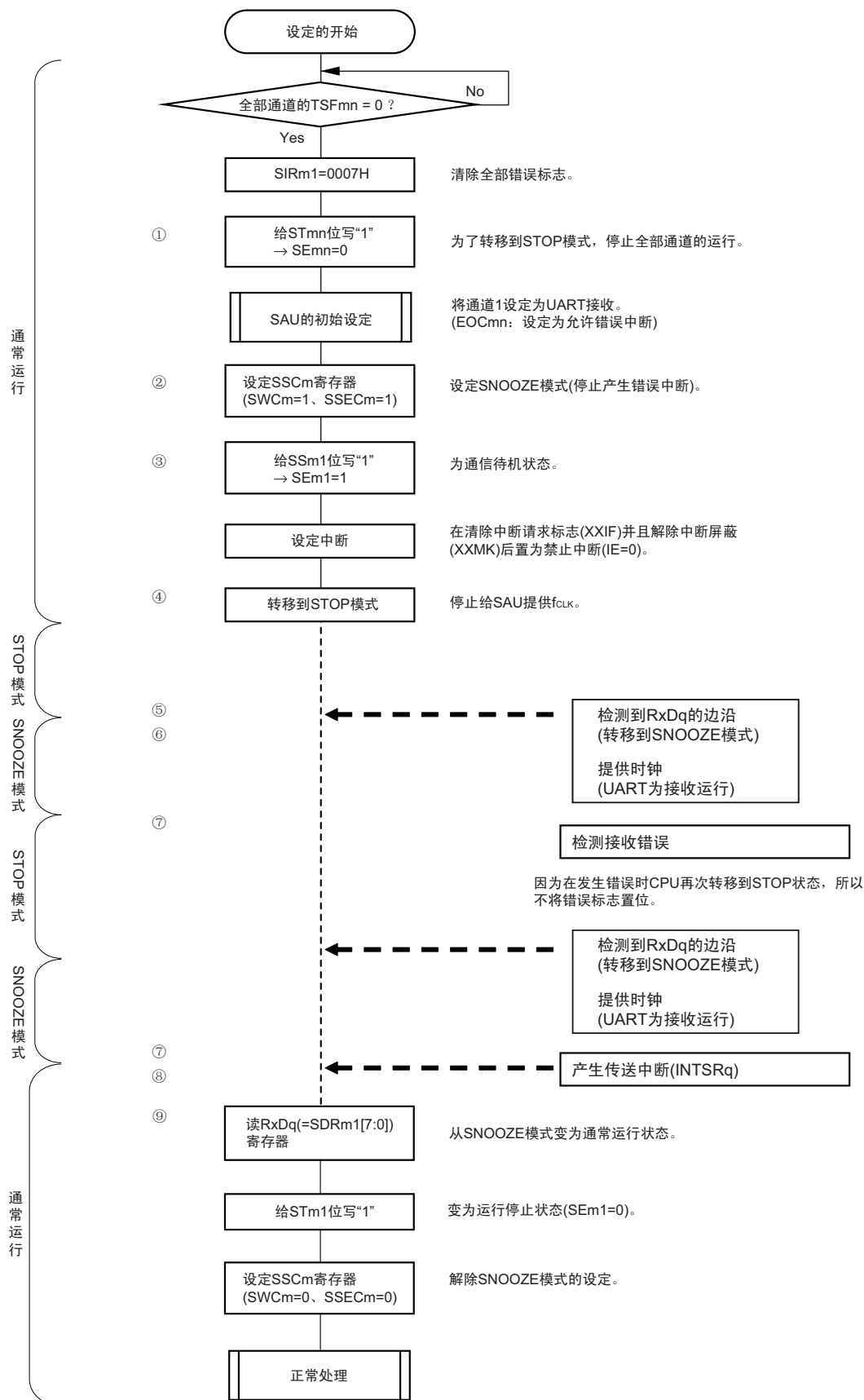


注 1. 必须在 SWCm 位为“1”的状态下读接收数据。  
2. 在 SNOOZE 模式中 UARTq 接收正常结束后，能不更改设定而继续进行通常的接收运行。但是因为 SSECm 位为“1”，所以即使发生帧错误或者奇偶校验错误，也不将 PEFm1 位和 FEFm1 位置位，并且也不产生错误中断（INTSREq）。

注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEM1 位并且停止运行），而且还必须在接收结束后清除 SWCm 位（解除 SNOOZE 模式）。  
2. 在 SSECm 位为“1”的状态下使用 SNOOZE 模式时，不发生溢出错误。因此，要使用 SNOOZE 模式时，必须在转移到 STOP 模式前读 SDRm1 寄存器的 bit7 ~ 0（RxDq）。

备注 1. 图中的①~⑨对应“图 11-118 SNOOZE 模式运行（异常运行②）的流程图”中的①~⑨。  
2. m=0, q=0

图 11-118 SNOOZE 模式运行（异常运行②）的流程图



注意 在 SSECm 位为“1”的状态下使用 SNOOZE 模式时，不发生溢出错误。因此，要使用 SNOOZE 模式时，必须在转移到 STOP 模式前读 SDRm1 寄存器的 bit7 ~ 0 (RxDq)。

备注 1. 图中的①~⑨对应“图 11-117 SNOOZE 模式运行（异常运行②）的时序图”中的①~⑨。

2. m=0, q=0

#### 11.7.4 波特率的计算

##### (1) 波特率的计算式

UART (UART0) 通信的波特率能用以下计算式进行计算：

$$(\text{波特率}) = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}}) \text{ 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

注意 禁止将串行数据寄存器 mn (SDRmn) 的 SDRmn[15:9] 设定为 “0000000B” 和 “0000001B”。

备注 1. 因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15～9 的值 (0000010B～1111111B)，所以为 2～127。

2. m: 单元号 (m=0) n: 通道号 (n=0、1) mn=00、01

运行时钟 (f<sub>MCK</sub>) 取决于串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的 bit15 (CKSmn 位)。

表 11-4 UART 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 ( $f_{MCK}$ ) 注	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		$f_{CLK}=32\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	$f_{CLK}$	32MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	7.81kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	3.91kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.95kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	977Hz
1	0	0	0	0	X	X	X	X	$f_{CLK}$	32MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	7.81kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	3.91kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.95kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	977Hz
上述以外									禁止设定。	

注 要更改被选择为  $f_{CLK}$  的时钟（更改系统时钟控制寄存器（CKC）的值）时，必须在停止串行阵列单元（SAU）的运行（串行通道停止寄存器 m（STM）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0) n: 通道号 (n=0、1) mn=00、01

## (2) 发送时的波特率误差

UART（UART0）通信发送时的波特率误差能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

$$(\text{波特率误差}) = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100 [\%]$$

$f_{\text{CLK}}=32\text{MHz}$  时的 UART 波特率的设定例子如下所示。

UART 波特率 (目标波特率)	$f_{\text{CLK}}=32\text{MHz}$			
	运行时钟 ( $f_{\text{MCK}}$ )	SDRmn[15:9]	波特率的计算值	与目标波特率的误差
300bps	$f_{\text{CLK}}/2^9$	103	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	63	31250.0bps	$\pm 0.0\%$
38400bps	$f_{\text{CLK}}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	103	76923.1bps	+0.16%
153600bps	$f_{\text{CLK}}$	103	153846bps	+0.16%
312500bps	$f_{\text{CLK}}$	50	313725.5bps	$\pm 0.39\%$

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

(3) 接收时的波特率容许范围

UART（UART0）通信接收时的波特率容许范围能用以下计算式进行计算，必须将发送方的波特率设定在接收方的波特率容许范围内。

$$\text{（可接收的最大波特率）} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{（可接收的最小波特率）} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

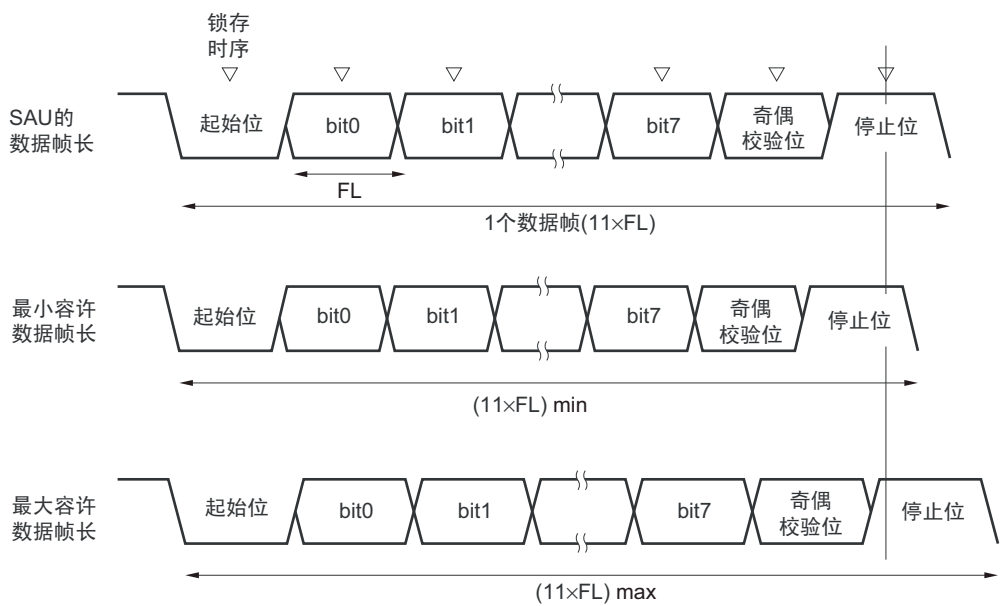
Brate   ：接收方的波特率的计算值（参照“11.7.4(1) 波特率的计算式”）

k       ：SDRmn[15:9]+1

Nfr     ：1 个数据帧的帧长 [ 位 ]  
          =（起始位）+（数据长度）+（奇偶校验位）+（停止位）

备注   m: 单元号（m=0）   n: 通道号（n=1）   mn=01

图 11-119 接收时的波特率容许范围（1 个数据帧的帧长 =11 位的情况）



如图 11-119 所示，在检测到起始位后，接收数据的锁存时序取决于串行数据寄存器 mn（SDRmn）的 bit15 ～ 9 设定的分频比。如果最后的数据（停止位）能赶上此锁存时序，就能正常接收。



### 11.7.5 在 UART（UART0）通信过程中发生错误时的处理步骤

在 UART（UART0）通信过程中发生错误时的处理步骤如图 11-120 和图 11-121 所示。

图 11-120 发生奇偶校验错误或者溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn（SIRmn）写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

图 11-121 发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn（SDRmn）。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn（SSRmn）。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器 mn（SIRmn）。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。
将串行通道停止寄存器 m（STm）的 STmn 位置“1”。	串行通道允许状态寄存器 m（SEm）的 SEmn 位为“0”并且通道 n 为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存器 m（SSm）的 SSmn 位置“1”。	串行通道允许状态寄存器 m（SEm）的 SEmn 位为“1”并且通道 n 为可运行状态。	

备注 m: 单元号（m=0） n: 通道号（n=0、1） mn=00、01

## 第 12 章 串行接口 IICA

R7F0C010 内置 2 个串行接口 IICA 的单元，支持 2 个从属地址。

当将 IICA0 和 IICA1 用作从属设备时，如果其中 1 个地址和从 I<sup>2</sup>C 总线接受的地址相同，对应的从属设备就能和主控设备进行通信。

### 12.1 串行接口 IICA 的功能

串行接口 IICA 有以下 3 种模式。

#### (1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

#### (2) I<sup>2</sup>C 总线模式（支持多主控）

此模式通过串行时钟（SCLAn）和串行数据总线（SDAAn）的 2 条线，与多个设备进行 8 位数据传送。

符合 I<sup>2</sup>C 总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的 I<sup>2</sup>C 总线控制部分。

因为串行接口 IICA 的 SCLAn 引脚和 SDAAn 引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

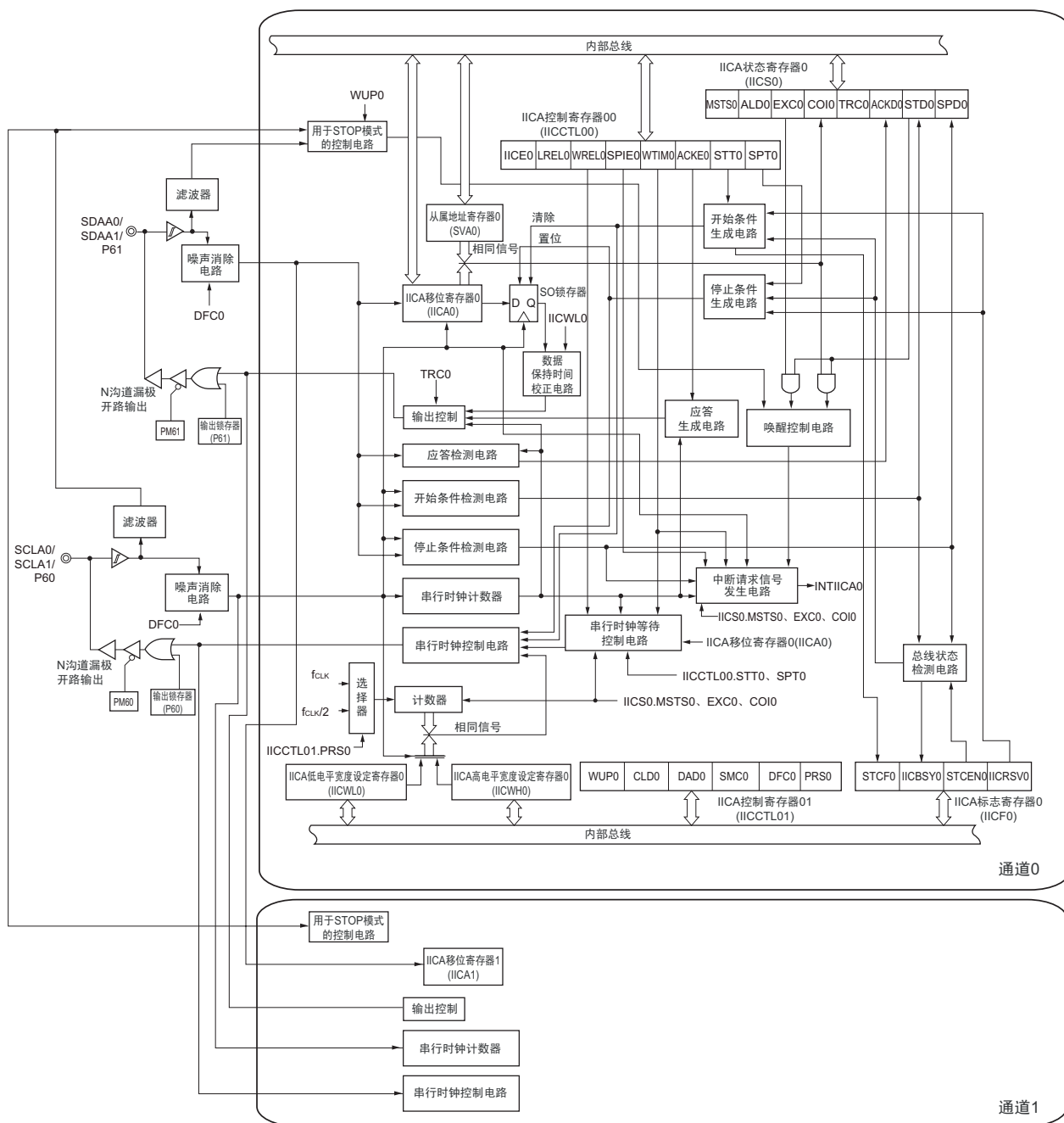
#### (3) 唤醒模式

在 STOP 模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除 STOP 模式。通过 IICA 控制寄存器 n1（IICCTLn1）的 WUPn 位进行设定。

串行接口 IICA 的框图如图 12-1 所示。

备注 n=0、1

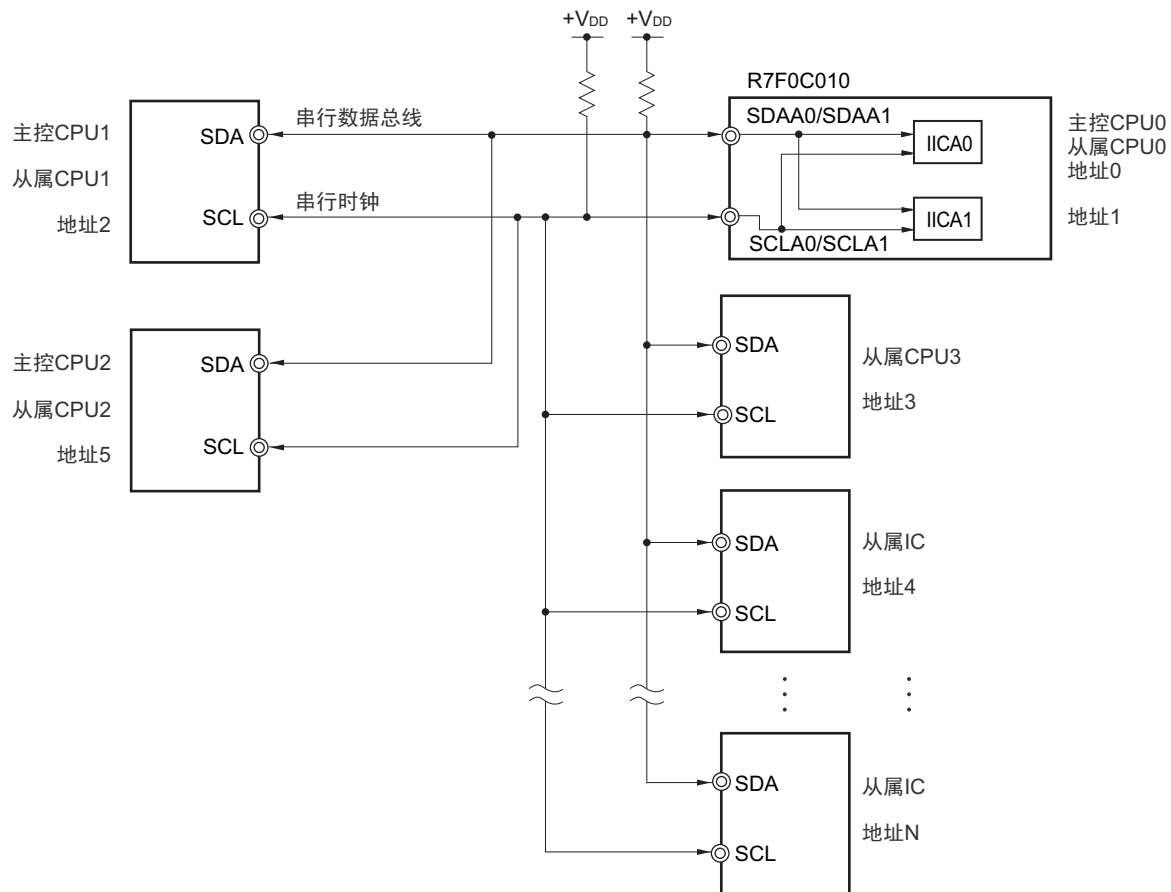
图 12-1 串行接口 IICA 的框图



**注意** 因为 R7F0C010 有通道 0 和通道 1，所以能同时等待接受。但是，因共享 P60/SCLA0/SCLA1 引脚和 P61/SDAA0/SDAA1 引脚而不能同时通信。

串行总线的结构例子如图 12-2 所示。

图 12-2 I<sup>2</sup>C 总线的串行总线结构例子



备注 因为 R7F0C010 内置 2 个串行接口 IICA 的单元，所以能支持 2 个地址进行通信。

## 12.2 串行接口 IICA 的结构

串行接口 IICA 由以下硬件构成。

表 12-1 串行接口 IICA 的结构

项目	结构
寄存器	IICA 移位寄存器 n (IICAn) 从属地址寄存器 n (SVAn)
控制寄存器	外围允许寄存器 0 (PER0) IICA 控制寄存器 n0 (IICCTLn0) IICA 状态寄存器 n (IICSn) IICA 标志寄存器 n (IICFn) IICA 控制寄存器 n1 (IICCTLn1) IICA 低电平宽度设定寄存器 n (IICWLn) IICA 高电平宽度设定寄存器 n (IICWHn) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

备注 n=0、1

### (1) IICA 移位寄存器 n (IICAn)

IICAn 寄存器是与串行时钟同步进行 8 位串行数据和 8 位并行数据相互转换的寄存器，用于发送和接收。能通过读写 IICAn 寄存器来控制实际的发送和接收。  
在等待期间，通过写 IICAn 寄存器来解除等待，开始传送数据。  
通过 8 位存储器操作指令设定 IICAn 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 12-3 IICAn 移位寄存器 n (IICAn) 的格式

地址: FFF50H (IICA0)、FFF54H (IICA1)				复位后: 00H		R/W		
符号	7	6	5	4	3	2	1	0

注意 1. 在数据传送过程中，不能给 IICAn 寄存器写数据。

- 只能在等待期间读写 IICAn 寄存器。除了等待期间以外，禁止在通信状态下存取 IICAn 寄存器。但是，在主控设备的情况下，能在将通信触发位 (STTn) 置“1”后写一次 IICAn 寄存器。
- 当预约通信时，必须在检测到由停止条件产生的中断后给 IICAn 寄存器写数据。

备注 n=0、1

## (2) 从属地址寄存器 n (SVAn)

这是在用作从属设备时保存 7 位本地站地址 {A6, A5, A4, A3, A2, A1, A0} 的寄存器。

通过 8 位存储器操作指令设定 SVAn 寄存器。但是，在 STDn 位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-4 从属地址寄存器 n (SVAn) 的格式

地址: F0234H (SVA0)、F023DH (SVA1)				复位后: 00H		R/W		
符号	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 注

注 bit0 固定为“0”。

## (3) SO 锁存器

SO 锁存器保持 SDAAn 引脚的输出电平。

## (4) 唤醒控制电路

当设定在从属地址寄存器 n (SVAn) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIICAn)。

## (5) 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了 8 位数据的发送和接收。

## (6) 中断请求信号发生电路

此电路控制产生中断请求信号 (INTIICAn)。

由以下 2 种触发产生 I<sup>2</sup>C 中断请求。

- 第 8 个或者第 9 个串行时钟的下降 (通过 WTIMn 位进行设定)
- 因检测到停止条件而产生中断请求 (通过 SPIEn 位进行设定)。

备注 WTIMn 位 : IICA 控制寄存器 n0 (IICCTLn0) 的 bit3

SPIEn 位 : IICA 控制寄存器 n0 (IICCTLn0) 的 bit4

## (7) 串行时钟控制电路

在主机模式中，此电路从采样时钟生成输出到 SCLAn 引脚的时钟。

## (8) 串行时钟等待控制电路

此电路控制等待时序。

备注 n=0、1

## (9) 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

## (10) 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

## (11) 开始条件生成电路

如果将  $STTn$  位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下（ $IICRSVn$  位 =1）并且没有释放总线（ $IICBSYn$  位 =1）时，忽视开始条件请求并且将  $STCFn$  位置“1”。

## (12) 停止条件生成电路

如果将  $SPTn$  位置“1”，此电路就生成停止条件。

## (13) 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。

但是，在刚运行时不能立即检测总线状态，因此必须通过  $STCENn$  位设定总线状态检测电路的初始状态。

备注	$STTn$ 位	: IICA 控制寄存器 n0（ $IICCTLn0$ ）的 bit1
	$SPTn$ 位	: IICA 控制寄存器 n0（ $IICCTLn0$ ）的 bit0
	$IICRSVn$ 位	: IICA 标志寄存器 n（ $IICFn$ ）的 bit0
	$IICBSYn$ 位	: IICA 标志寄存器 n（ $IICFn$ ）的 bit6
	$STCFn$ 位	: IICA 标志寄存器 n（ $IICFn$ ）的 bit7
	$STCENn$ 位	: IICA 标志寄存器 n（ $IICFn$ ）的 bit1

备注 n=0、1

### 12.3 控制串行接口 IICA 的寄存器

通过以下 8 种寄存器控制串行接口 IICA。

- 外围允许寄存器 0（PER0）
- IICA 控制寄存器 n0（IICCTLn0）
- IICA 标志寄存器 n（IICFn）
- IICA 状态寄存器 n（IICSn）
- IICA 控制寄存器 n1（IICCTLn1）
- IICA 低电平宽度设定寄存器 n（IICWLn）
- IICA 高电平宽度设定寄存器 n（IICWHn）
- 端口模式寄存器 6（PM6）
- 端口寄存器 6（P6）

备注 n=0、1

#### 12.3.1 外围允许寄存器 0（PER0）

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

要使用串行接口 IICAn 时，必须将 bit6 和 bit4（IICA1EN、IICA0EN）置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-5 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	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICAnEN	提供串行接口 IICAn 的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"><li>• 不能写串行接口 IICAn 使用的 SFR。</li><li>• 串行接口 IICAn 处于复位状态。</li></ul>
1	允许提供输入时钟。 <ul style="list-style-type: none"><li>• 能读写串行接口 IICAn 使用的 SFR。</li></ul>

- 注意 1. 要设定串行接口 IICAn 时，必须先将 IICAnEN 位置“1”。当 IICAnEN 位为“0”时，忽视串行接口 IICAn 的控制寄存器的写操作，而且读取值也都为初始值（端口模式寄存器 6（PM6）和端口寄存器 6（P6）除外）。
2. 当将本产品的 IICA 用作主控设备时，只需要允许 1 个通道的 IICA 运行，因此禁止将 IICA0EN 位和 IICA1EN 位都置“1”。
3. 当本产品的 IICA 支持 2 个从属地址时，需要将 IICA0EN 位和 IICA1EN 位都置“1”。
4. 必须将 bit1、3、7 置“0”。

备注 n=0、1



### 12.3.2 IICA 控制寄存器 n0 (IICCTLn0)

这是允许或者停止 I<sup>2</sup>C 运行、设定等待时序以及设定其他 I<sup>2</sup>C 运行的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICCTLn0 寄存器。但是，必须在 IICEn 位为“0”时或者在等待期间设定 SPIEn 位、WTIMn 位和 ACKEn 位，而且在将 IICEn 位从“0”置为“1”时能同时设定这些位。

在产生复位信号后，此寄存器的值变为“00H”。

备注 n=0、1

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (1/4)

地址: F0230H (IICCTL00)、F0238H (IICCTL10) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I <sup>2</sup> C 运行的允许
0	停止运行。对 IICA 状态寄存器 n (IICSn) 进行复位 <sup>注 1</sup> , 并且停止内部运行。
1	允许运行。
必须在 SCLAn 线和 SDAAn 线为高电平的状态下将此位置“1”。	
清除条件 (IICEn=0)	置位条件 (IICEn=1)
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

LRELn <sup>注 2、3</sup>	通信的退出
0	通常运行
1	退出当前的通信, 进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况时使用。 SCLAn 线和 SDAAn 线变为高阻状态。 IICA 控制寄存器 n0 (IICCTLn0) 和 IICA 状态寄存器 n (IICSn) 中的以下标志被清“0”: •STTn •SPTn •MSTSn •EXCn •COIn •TRCn •ACKDn •STDn
变为退出通信的待机状态, 保持到满足以下的通信参加条件为止。	
<ul style="list-style-type: none"> <li>在检测到停止条件后作为主控设备启动。</li> <li>在检测到开始条件后地址匹配或者接收到扩展码。</li> </ul>	
清除条件 (LRELn=0)	置位条件 (LRELn=1)
<ul style="list-style-type: none"> <li>在执行后自动清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

WRELn <sup>注 2、3</sup>	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRCn=1) 的第 9 个时钟等待期间将 WRELn 位 (解除等待) 置位, SDAAn 线就变为高阻抗状态 (TRCn=0)。	
清除条件 (WRELn=0)	置位条件 (WRELn=1)
<ul style="list-style-type: none"> <li>在执行后自动清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

- 注 1. 对 IICA 状态寄存器 n (IICA0)、IICA 标志寄存器 n (IICF0) 的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1 (IICCTLn1) 的 CLDn 位和 DADn 位进行复位。
2. 在 IICEn 位为“0”的状态下, 此位的信号无效。
3. LRELn 位和 WRELn 位的读取值总是“0”。

注意 如果在 SCLAn 线为高电平、SDAAn 线为低电平并且数字滤波器为 ON (IICCTLn1 寄存器的 DFCn=1) 时允许 I<sup>2</sup>C 运行 (IICEn=1), 就立即检测开始条件。此时, 必须在允许 I<sup>2</sup>C 运行 (IICEn=1) 后连续通过位存储器操作指令将 LRELn 位置“1”。

备注 n=0、1

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (2/4)

SPIEn 注 1	允许或者禁止停止条件检测产生的中断请求
0	禁止
1	允许
当 IICA 控制寄存器 n1 (IICCTLn1) 的 WUPn 位为“1”时, 即使将 SPIEn 位置“1”也不产生停止条件中断。	
清除条件 (SPIEn=0)	置位条件 (SPIEn=1)
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

WTIMn 注 1	等待和中断请求的控制
0	在第 8 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 8 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 8 个时钟后, 将时钟置为低电平, 然后等待主控设备。
1	在第 9 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 9 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 9 个时钟后, 将时钟置为低电平, 然后等待主控设备。
在地址传送期间, 与此位的设定无关, 在第 9 个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。主控设备在地址传送期间的第 9 个时钟下降沿进入等待。接收到本地站地址的从属设备在产生应答 (ACK) 后的第 9 个时钟下降沿进入等待, 但是接收到扩展码的从属设备在第 8 个时钟下降沿进入等待。	
清除条件 (WTIMn=0)	置位条件 (WTIMn=1)
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

ACKEn 注1、2	应答控制
0	禁止应答。
1	允许应答。在第 9 个时钟期间将 SDAAn 线置为低电平。
清除条件 (ACKEn=0)	置位条件 (ACKEn=1)
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>

注 1. 在 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。

2. 在地址传送过程中不是扩展码时, 设定值无效。

当为从属设备并且地址匹配时, 与设定值无关而生成应答。

备注 n=0、1

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (3/4)

STTn 注	开始条件的触发
0	不生成开始条件。
1	<p>当总线被释放时（待机状态，IICBSYn 位为“0”）： 如果将此位置“1”，就生成开始条件（作为主控设备的启动）。 当第三方正在通信时：</p> <ul style="list-style-type: none"> <li>• 允许通信预约功能的情况（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。</li> <li>• 禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除 STTn 位并且将 STTn 清除标志（STCFn）置“1”，不生成开始条件。</li> </ul> <p>等待状态（主控设备）： 在解除等待后生成重新开始条件。</p>
<p>有关置位时序的注意事项：</p> <ul style="list-style-type: none"> <li>• 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn 位置“0”并且通知从属设备接收已经完成后等待期间才能将此位置“1”。</li> <li>• 主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。</li> <li>• 禁止与停止条件的触发（SPTn）同时置“1”。</li> <li>• 在将 STTn 位置“1”后，禁止在满足清除条件前再次将此位“1”。</li> </ul>	
清除条件（STTn=0）	置位条件（STTn=1）
<ul style="list-style-type: none"> <li>• 在禁止通信预约的状态下将 STTn 位置“1”。</li> <li>• 在仲裁失败时</li> <li>• 主控设备生成开始条件。</li> <li>• 因 LRELn 位为“1”（退出通信）而进行的清除</li> <li>• 当 IICEn 位为“0”（停止运行）时</li> <li>• 当复位时</li> </ul>	<ul style="list-style-type: none"> <li>• 通过指令置位。</li> </ul>

注 在 IICEn 位为“0”的状态下，此位的信号无效。

备注 1. 如果在设定数据后读 bit1（STTn），此位就变为“0”。

2. IICRSVn : IICA 标志寄存器 n（IICFn）的 bit0

STCFn : IICA 标志寄存器 n（IICFn）的 bit7

3. n=0、1

图 12-6 IICA 控制寄存器 n0（IICCTLn0）的格式 (4/4)

SPTn	停止条件的触发
0	不生成停止条件。
1	生成停止条件 （作为主控设备的传送结束）。
有关置位时序的注意事项：	
<ul style="list-style-type: none"><li>• 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKEn 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。</li><li>• 主控发送：在应答期间，可能无法正常生成停止条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。</li><li>• 禁止与开始条件的触发（STTn）同时置“1”。</li><li>• 只有在主控设备的情况下才能将 SPTn 位置“1”。</li><li>• 在 WTIMn 位为“0”时，必须注意：如果在输出 8 个时钟后的等待期间将 SPTn 位置“1”，就在解除等待后的第 9 个时钟的高电平期间生成停止条件。必须在输出 8 个时钟后的等待期间将 WTIMn 位从“0”置为“1”并且在输出第 9 个时钟后的等待期间将 SPTn 位置“1”。</li><li>• 在将 SPTn 位置“1”后，禁止在满足清除条件前再次将此位置“1”。</li></ul>	
清除条件（SPTn=0）	置位条件（SPTn=1）
<ul style="list-style-type: none"><li>• 当仲裁失败时</li><li>• 在检测到停止条件后自动清除。</li><li>• 因 LRELn 位为“1”（退出通信）而进行的清除</li><li>• 当 IICEn 位为“0”（停止运行）时</li><li>• 当复位时</li></ul>	<ul style="list-style-type: none"><li>• 通过指令置位。</li></ul>

注意 在 IICA 状态寄存器 n（IICSn）的 bit3（TRCn）为“1”（发送状态）时，如果在第 9 个时钟将 IICCTLn0 寄存器的 bit5（WRELn）置“1”来解除等待，就在清除 TRCn 位（接收状态）后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1”（发送状态）时的等待解除。

备注 1. 如果在设定数据后读 bit0（SPTn），此位就变为“0”。

2. n=0、1

12.3.3 IICA 状态寄存器 n（IICS<sub>n</sub>）

这是表示 I<sup>2</sup>C 状态的寄存器。

只有在 STT<sub>n</sub> 位为“1”并且等待期间，才能通过 1 位或者 8 位存储器操作指令读 IICS<sub>n</sub> 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

**注意** 在 STOP 模式中允许地址匹配唤醒功能（WUP<sub>n</sub>=1）状态下，禁止读 IICS<sub>n</sub> 寄存器。在 WUP<sub>n</sub> 位为“1”的状态下，与 INTIICAn 中断请求无关，如果将 WUP<sub>n</sub> 位从“1”改为“0”（停止唤醒运行），就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许（SPIEn=1）因检测到停止条件而产生的中断，并且在检测到中断后读 IICS<sub>n</sub> 寄存器。

**备注** STT<sub>n</sub> : IICA 控制寄存器 n0（IICCTLn0）的 bit1  
WUP<sub>n</sub> : IICA 控制寄存器 n1（IICCTLn1）的 bit7

图 12-7 IICA 状态寄存器 n（IICS<sub>n</sub>）的格式 (1/3)



**注** 即使对 IICS<sub>n</sub> 寄存器以外的位执行位存储器操作指令，也清除此位。因此，在使用 ALD<sub>n</sub> 位时，必须在读其他位前先读 ALD<sub>n</sub> 位的数据。

**备注 1.** LREL<sub>n</sub> : IICA 控制寄存器 n0（IICCTLn0）的 bit6  
IICEn : IICA 控制寄存器 n0（IICCTLn0）的 bit7

**2.** n=0、1

图 12-7 IICA 状态寄存器 n (IICSn) 的格式 (2/3)

EXCn	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXCn=0)		置位条件 (EXCn=1)
<ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当检测到停止条件时</li> <li>因 LRELn 位为“1” (退出通信) 而进行的清除</li> <li>当 IICEn 位从“1”变为“0” (停止运行) 时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>当接收的地址数据的高 4 位为“0000”或者“1111”时 (在第 8 个时钟的上升沿置位)</li> </ul>

COIn	匹配地址的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COIn=0)		置位条件 (COIn=1)
<ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当检测到停止条件时</li> <li>因 LRELn 位为“1” (退出通信) 而进行的清除</li> <li>当 IICEn 位从“1”变为“0” (停止运行) 时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>当接收地址和本地站地址 (从属地址寄存器 n (SVAn)) 相同时 (在第 8 个时钟的上升沿置位)</li> </ul>

TRCn	发送 / 接收的状态检测	
0	接收状态 (发送状态除外)。将 SDAAn 线置为高阻抗。	
1	发送状态。设定为能将 SOn 锁存器的值输出到 SDAAn 线 (在第 1 字节的第 9 个时钟的下降沿以后有效)。	
清除条件 (TRCn=0)		置位条件 (TRCn=1)
<主控设备和从属设备> <ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>因 LRELn 位为“1” (退出通信) 而进行的清除</li> <li>当 IICEn 位从“1”变为“0” (停止运行) 时</li> <li>因 WRELn 位为“1” (解除等待) 而进行的清除注</li> <li>当 ALDn 位从“0”变为“1” (仲裁失败) 时</li> <li>当复位时</li> <li>不参加通信的情况 (MSTSn、EXCn、COIn=0)</li> </ul> <主控设备> <ul style="list-style-type: none"> <li>当第 1 字节的 LSB (传送方向指示位) 输出“1”时</li> </ul> <从属设备> <ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>当第 1 字节的 LSB (传送方向指示位) 输入“0”时</li> </ul>		<主控设备> <ul style="list-style-type: none"> <li>当生成开始条件时</li> <li>当第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输出“0” (主控发送) 时</li> </ul> <从属设备> <ul style="list-style-type: none"> <li>当主控设备的第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输入“1” (从属发送) 时</li> </ul>

注 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

备注 1. LRELn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit6

IICEn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0、1

图 12-7 IICA 状态寄存器 n (IICS<sub>n</sub>) 的格式 (3/3)

ACKD <sub>n</sub>	应答 (ACK) 的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件 (ACKD <sub>n</sub> =0)		置位条件 (ACKD <sub>n</sub> =1)
<ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>当下一个字节的第 1 个时钟上升时</li> <li>因 LREL<sub>n</sub> 位为 “1” (退出通信) 而进行的清除</li> <li>当 IICEn 位从 “1” 变为 “0” (停止运行) 时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>在 SCLAn 线的第 9 个时钟上升沿将 SDAAn 线置为低电平时</li> </ul>

STD <sub>n</sub>	开始条件的检测	
0	未检测到开始条件。	
1	检测到开始条件，表示处于地址传送期间。	
清除条件 (STD <sub>n</sub> =0)		置位条件 (STD <sub>n</sub> =1)
<ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>在地址传送后的下一个字节的第 1 个时钟上升时</li> <li>因 LREL<sub>n</sub> 位为 “1” (退出通信) 而进行的清除</li> <li>当 IICEn 位从 “1” 变为 “0” (停止运行) 时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>当检测到开始条件时</li> </ul>

SPD <sub>n</sub>	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件，主控设备结束通信并且已释放总线。	
清除条件 (SPD <sub>n</sub> =0)		置位条件 (SPD <sub>n</sub> =1)
<ul style="list-style-type: none"> <li>在将此位置位后，在检测到开始条件后的地址传送字节的第 1 个时钟上升时</li> <li>当 WUP<sub>n</sub> 位从 “1” 变为 “0” 时</li> <li>当 IICEn 位从 “1” 变为 “0” (停止运行) 时</li> <li>当复位时</li> </ul>		<ul style="list-style-type: none"> <li>当检测到停止条件时</li> </ul>

备注 1. LREL<sub>n</sub> : IICA 控制寄存器 n0 (IICCTL<sub>n</sub>0) 的 bit6

IICEn : IICA 控制寄存器 n0 (IICCTL<sub>n</sub>0) 的 bit7

2. n=0、1

#### 12.3.4 IICA 标志寄存器 n (IICFn)

这是设定 I<sup>2</sup>C 运行模式以及表示 I<sup>2</sup>C 总线状态的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICFn 寄存器。但是，只能读 STT<sub>n</sub> 清除标志 (STCF<sub>n</sub>) 和 I<sup>2</sup>C 总线状态标志 (IICBSY<sub>n</sub>)。

通过 IICRSV<sub>n</sub> 位设定允许或者禁止通信预约功能，并且通过 STCEN<sub>n</sub> 位设定 IICBSY<sub>n</sub> 位的初始值。

只有在禁止 I<sup>2</sup>C 运行 (IICA 控制寄存器 n0 (IICCTL<sub>n</sub>0) 的 bit7 (IICEn) =0) 时才能写 IICRSV<sub>n</sub> 位和 STCEN<sub>n</sub> 位。在允许运行后，只能读 IICFn 寄存器。

在产生复位信号后，此寄存器的值变为 “00H”。



图 12-8 IICA 标志寄存器 n (IICFn) 的格式

地址: FFF52H (IICF0)、FFF56H (IICF1)      复位后: 00H      R/W 注

符号	7	6	5	4	3	2	1	0
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn 清除标志
0	发行开始条件。
1	无法发行开始条件而清除 STTn 标志。
清除条件 (STCFn=0)	
<ul style="list-style-type: none"> <li>因 STTn 位为“1”而进行的清除</li> <li>当 IICEn 位为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	
置位条件 (STCFn=1)	
<ul style="list-style-type: none"> <li>在设定为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将 STTn 位清“0”时</li> </ul>	

IICBSYn	I <sup>2</sup> C 总线状态标志
0	总线释放状态 (STCENn=1 时的通信初始状态)
1	总线通信状态 (STCENn=0 时的通信初始状态)
清除条件 (IICBSYn=0)	
<ul style="list-style-type: none"> <li>当检测到停止条件时</li> <li>当 IICEn 位为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	
置位条件 (IICBSYn=1)	
<ul style="list-style-type: none"> <li>当检测到开始条件时</li> <li>STCENn 位为“0”时的 IICEn 位的置位</li> </ul>	

STCENn	初始开始允许触发
0	在允许运行 (IICEn=1) 后, 通过检测停止条件来允许生成开始条件。
1	在允许运行 (IICEn=1) 后, 不检测停止条件而允许生成开始条件。
清除条件 (STCENn=0)	
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当检测到开始条件时</li> <li>当复位时</li> </ul>	
置位条件 (STCENn=1)	
<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>	

IICRSVn	通信预约功能禁止位
0	允许通信预约。
1	禁止通信预约。
清除条件 (IICRSVn=0)	
<ul style="list-style-type: none"> <li>通过指令清除。</li> <li>当复位时</li> </ul>	
置位条件 (IICRSVn=1)	
<ul style="list-style-type: none"> <li>通过指令置位。</li> </ul>	

注 bit6 和 bit7 是只读位。

注意 1. 只有在停止运行 (IICEn=0) 时才能写 STCENn 位。

2. 如果 STCENn 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0), 因此为了避免在发行第 1 个开始条件 (STTn=1) 时破坏其他通信, 需要确认没有正在通信的第三方。

3. 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

备注 1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

2. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

### 12.3.5 IICA 控制寄存器 n1 (IICCTLn1)

这是用于设定 I<sup>2</sup>C 运行模式以及检测 SCLAn 引脚和 SDAAn 引脚状态的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICCTLn1 寄存器。但是，只能读 CLDn 位和 DADn 位。

除了 WUPn 位以外，必须在禁止 I<sup>2</sup>C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICCTLn1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (1/2)

地址: F0231H (IICCTL01)、F0239H (IICCTL11)

复位后: 00H

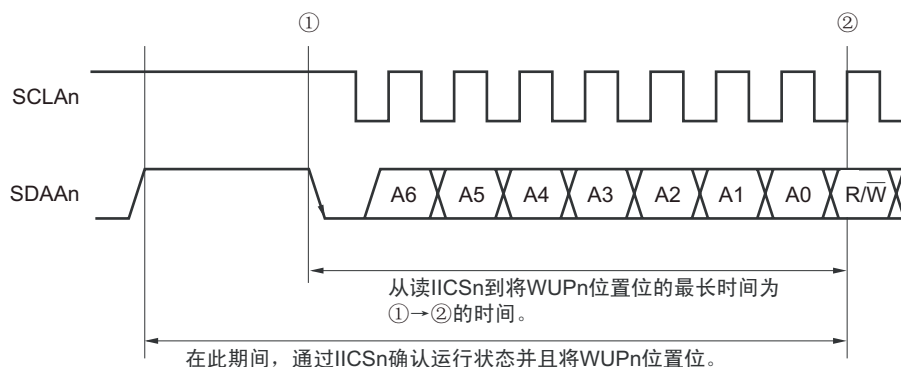
R/W 注 1

符号	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	地址匹配唤醒的控制
0	在 STOP 模式中，停止地址匹配唤醒功能的运行。
1	在 STOP 模式中，允许地址匹配唤醒功能的运行。
<p>要通过将 WUPn 位置“1”来转移到 STOP 模式时，必须在将 WUPn 位置“1”后至少经过 3 个时钟，然后执行 STOP 指令（参照“图 12-22 将 WUPn 位置“1”时的流程”）。</p> <p>在地址匹配或者接收到扩展码后，必须将 WUPn 位清“0”。能通过将 WUPn 位清“0”来参加后续的通信（需要在将 WUPn 位清“0”后解除等待以及写发送数据）。</p> <p>在 WUPn 位为“1”的状态下，地址匹配或者接收到扩展码时的中断时序与 WUPn 位为“0”时的中断时序相同（根据时钟产生采样误差的延迟差）。另外，当 WUPn 位为“1”时，即使将 SPIEn 位置“1”也不产生停止条件中断。</p>	
清除条件 (WUPn=0)	
<ul style="list-style-type: none"> <li>通过指令清除（在地址匹配或者接收到扩展码后）。</li> </ul>	
置位条件 (WUPn=1)	
<ul style="list-style-type: none"> <li>通过指令置位 (MSTSn=0、EXCn=0、COIn=0 并且 STDn=0（不参加通信）) 注 2。</li> </ul>	

注 1. bit4 和 bit5 是只读位。

2. 在以下所示的期间，需要确认 IICA 状态寄存器 n (IICSn) 的状态并且将其置位。



备注 n=0、1

图 12-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (2/2)

CLDn	SCLAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)
0	检测到 SCLAn 引脚为低电平。
1	检测到 SCLAn 引脚为高电平。
清除条件 (CLDn=0)	
置位条件 (CLDn=1)	
<ul style="list-style-type: none"> <li>当 SCLAn 引脚为低电平时</li> <li>当 IICEn 位为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>当 SCLAn 引脚为高电平时</li> </ul>

DADn	SDAAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)
0	检测到 SDAAn 引脚为低电平。
1	检测到 SDAAn 引脚为高电平。
清除条件 (DADn=0)	
置位条件 (DADn=1)	
<ul style="list-style-type: none"> <li>当 SDAAn 引脚为低电平时</li> <li>当 IICEn 位为“0” (停止运行) 时</li> <li>当复位时</li> </ul>	<ul style="list-style-type: none"> <li>当 SDAAn 引脚为高电平时</li> </ul>

SMCn	运行模式的切换
0	在标准模式中运行 (最大传送速率: 100kbps)。
1	在快速模式 (最大传送速率: 400kbps) 或者增强型快速模式 (最大传送速率: 1Mbps) 中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器 OFF
1	数字滤波器 ON
<p>只有在快速模式或者增强型快速模式中才能使用数字滤波器。</p> <p>在快速模式或者增强型快速模式中, 无论是将 DFCn 位置“1”还是清“0”, 传送时钟都不变。</p> <p>在快速模式或者增强型快速模式中, 数字滤波器用于消除噪声。</p>	

PRSn	运行时钟的分频控制
0	选择 $f_{CLK}$ 作为运行时钟。
1	选择 $f_{CLK}/2$ 作为运行时钟。

注意 串行接口 IICA 运行时钟的最大工作频率为 20MHz(Max.)。当  $f_{CLK}$  超过 20MHz 时, 必须通过将 PRSn 位置“1”, 选择  $f_{CLK}/2$  作为运行时钟。

备注 1. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0、1

### 12.3.6 IICA 低电平宽度设定寄存器 n (IICWL<sub>n</sub>)

此寄存器设定串行接口 IICA 输出的 SCLAn 引脚信号低电平宽度 ( $t_{LOW}$ )。

通过 8 位存储器操作指令设定 IICWL<sub>n</sub> 寄存器。

必须在禁止 I<sup>2</sup>C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICWL<sub>n</sub> 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

有关 IICWL<sub>n</sub> 寄存器的设定方法, 请参照“12.4.2 通过 IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器设定传送时钟的方法”。

图 12-10 IICA 低电平宽度设定寄存器 n (IICWL<sub>n</sub>) 的格式

地址: F0232H (IICWL0)、F023AH (IICWL1)	复位后: FFH	R/W								
符号	7	6	5	4	3	2	1	0		
IICWL <sub>n</sub>										

### 12.3.7 IICA 高电平宽度设定寄存器 n (IICWH<sub>n</sub>)

此寄存器设定串行接口 IICA 输出的 SCLAn 引脚信号高电平宽度。

通过 8 位存储器操作指令设定 IICWH<sub>n</sub> 寄存器。

必须在禁止 I<sup>2</sup>C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICWH<sub>n</sub> 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

图 12-11 IICA 高电平宽度设定寄存器 n (IICWH<sub>n</sub>) 的格式

地址: F0233H (IICWH0)、F023BH (IICWH1)	复位后: FFH	R/W								
符号	7	6	5	4	3	2	1	0		
IICWH <sub>n</sub>										

备注 1. 有关通过 IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器设定传送时钟的方法, 请参照“12.4.2 通过 IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器设定传送时钟的方法”。

2. n=0、1

12.3.8 端口模式寄存器 6（PM6）

此寄存器以位为单位设定端口 6 的输入 / 输出。

在将 P60/SCLA0/SCLA1 引脚用作时钟输入 / 输出并且将 P61/SDAA0/SDAA1 引脚用作串行数据输入 / 输出时，必须将 PM60 和 PM61 置“0”并且将 P60 和 P61 的输出锁存器置“1”。

当 IICEn 位（IICA 控制寄存器 n0（IICCTLn0）的 bit7）为“0”时，P60/SCLA0/SCLA1 引脚和 P61/SDAA0/SDAA1 引脚为低电平输出（固定），因此必须在将 IICEn 位置“1”后切换到输出模式。

通过 1 位或者 8 位存储器操作指令设定 PM6 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

图 12-12 端口模式寄存器 6（PM6）的格式

地址：FFF26H	复位后：FFH	R/W							
符号	7	6	5	4	3	2	1	0	
PM6	1	1	1	1	1	1	PM61	PM60	

PM6n	P6n 引脚输入 / 输出模式的选择（n=0、1）
0	输出模式（输出缓冲器 ON）
1	输入模式（输出缓冲器 OFF）

## 12.4 I<sup>2</sup>C 总线模式的功能

### 12.4.1 引脚结构

串行时钟引脚（SCLAn）和串行数据总线引脚（SDAAn）的结构如下。

(1) SCLAn..... 串行时钟的输入/输出引脚

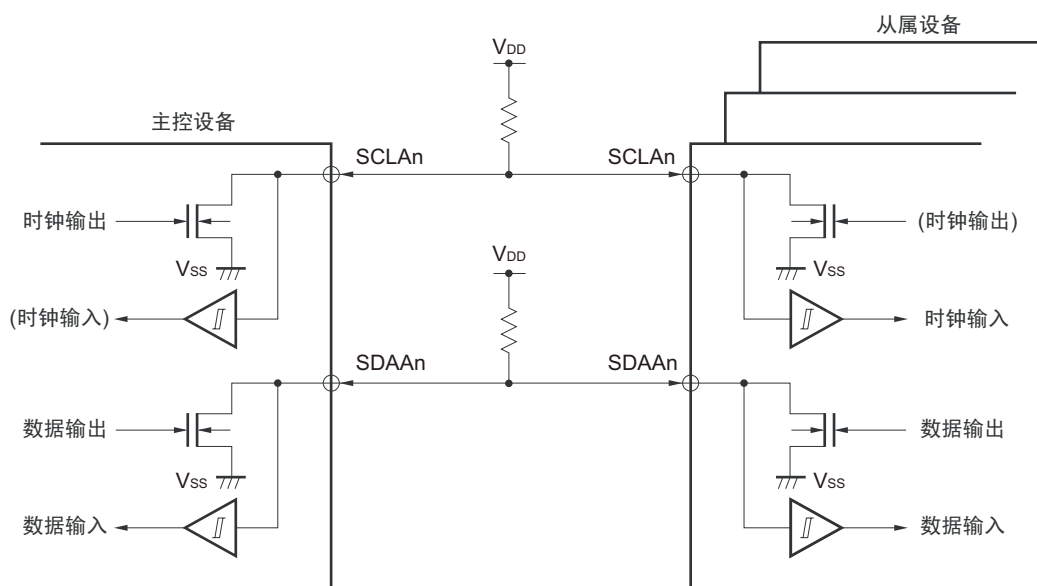
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

(2) SDAAn..... 串行数据的输入/输出复用引脚

主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

因为串行时钟线和串行数据总线的输出为 N 沟道漏极开路输出，所以需要外接上拉电阻。

图 12-13 引脚结构图



备注 n=0、1

### 12.4.2 通过 IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器设定传送时钟的方法

#### (1) 主控方传送时钟的设定方法

$$\text{传送时钟} = \frac{f_{\text{CLK}}}{\text{IICWL} + \text{IICWH} + f_{\text{CLK}}(t_{\text{R}} + t_{\text{F}})}$$

此时，IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器的最佳设定值如下：  
(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWL}_n = \frac{0.52}{\text{传送时钟}} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = \left( \frac{0.48}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}$$

- 标准模式

$$\text{IICWL}_n = \frac{0.47}{\text{传送时钟}} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = \left( \frac{0.53}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}$$

- 增强型快速模式

$$\text{IICWL}_n = \frac{0.50}{\text{传送时钟}} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = \left( \frac{0.50}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}$$

#### (2) 从属方 IICWL<sub>n</sub> 寄存器和 IICWH<sub>n</sub> 寄存器的设定方法

(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWL}_n = 1.3\mu\text{s} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = (1.2\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}$$

- 标准模式

$$\text{IICWL}_n = 4.7\mu\text{s} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = (5.3\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}$$

- 增强型快速模式

$$\text{IICWL}_n = 0.50\mu\text{s} \times f_{\text{CLK}}$$

$$\text{IICWH}_n = (0.50\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}$$

**注意** 在设定传送时钟的情况下，必须注意  $f_{\text{CLK}}$  的最小工作频率。串行接口 IICA 的  $f_{\text{CLK}}$  最小工作频率取决于运行模式。

快速模式 :  $f_{\text{CLK}}=3.5\text{MHz}(\text{Min.})$

增强型快速模式 :  $f_{\text{CLK}}=10\text{MHz}(\text{Min.})$

标准模式 :  $f_{\text{CLK}}=1\text{MHz}(\text{Min.})$

另外，串行接口 IICA 运行时钟的最大工作频率为 20MHz(Max.)。当  $f_{\text{CLK}}$  超过 20MHz 时，必须通过将 IICCTL<sub>n1</sub> 寄存器的 PRS<sub>n</sub> 位置“1”，选择  $f_{\text{CLK}}/2$  作为运行时钟。

备注 1. 因为 SDAAn 信号和 SCLAn 信号的上升时间 ( $t_R$ ) 和下降时间 ( $t_F$ ) 因上拉电阻和布线电容而不同, 所以必须各自计算。

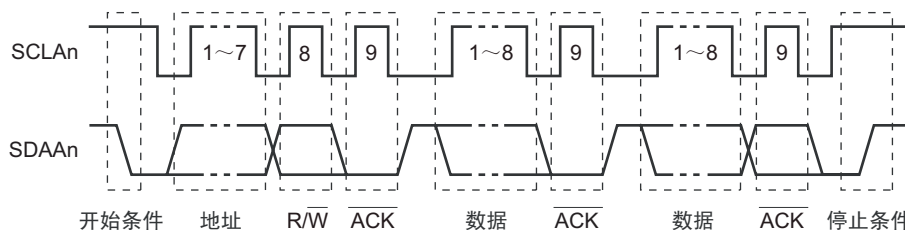
2. IICWLn : IICA 低电平宽度设定寄存器 n  
IICWHn : IICA 高电平宽度设定寄存器 n  
 $t_F$  : SDAAn 信号和 SCLAn 信号的下降时间  
 $t_R$  : SDAAn 信号和 SCLAn 信号的上升时间  
 $f_{CLK}$  : CPU/ 外围硬件的时钟频率
3. n=0、1

## 12.5 I<sup>2</sup>C 总线的定义和控制方法

以下说明 I<sup>2</sup>C 总线的串行数据通信格式和使用的信号。

I<sup>2</sup>C 总线的串行数据总线上生成的“开始条件”、“地址”、“数据”和“停止条件”的各传送时序如图 12-14 所示。

图 12-14 I<sup>2</sup>C 总线的串行数据传送时序



主控设备生成开始条件、从属地址和停止条件。

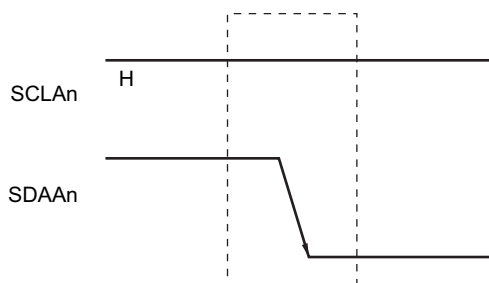
主控设备和从属设备都能生成应答 (ACK) (在一般情况下, 接收方输出 8 位数据)。

主控设备连续输出串行时钟 (SCLAn)。但是, 从属设备能延长 SCLAn 引脚的低电平期间并且插入等待。

### 12.5.1 开始条件

在 SCLAn 引脚为高电平时, 如果 SDAAn 引脚从高电平变为低电平, 就生成开始条件。SCLAn 引脚和 SDAAn 引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。在用作从属设备时, 能检测到开始条件。

图 12-15 开始条件



在检测到停止条件 (SPDn: IICA 状态寄存器 n (IICSn) 的 bit0=1) 的状态下, 如果将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”, 就输出开始条件。如果检测到开始条件, 就将 IICSn 寄存器的 bit1 (STDn) 置“1”。

备注 n=0、1



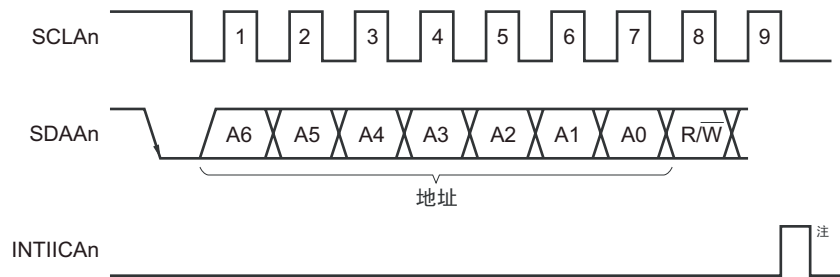
12.5.2 地址

开始条件的后续 7 位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的 7 位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查 7 位数据是否和从属地址寄存器 n（SVAn）的内容相同。此时，如果 7 位数据和 SVAn 寄存器的值相同，该从属设备就被选中，在主控设备生成开始条件或者停止条件前，与主控设备进行通信。

图 12-16 地址



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

如果将从属地址和“12.5.3 传送方向的指定”中说明的传送方向构成的 8 位数据写到 IICA 移位寄存器 n（IICAn），就输出地址。接收到的地址被写到 IICAn 寄存器。

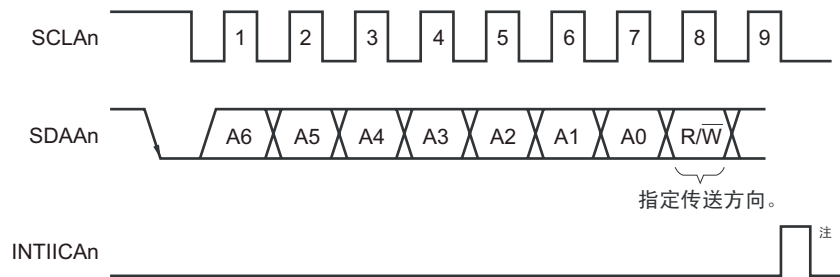
从属地址分配在 IICAn 寄存器的高 7 位。

12.5.3 传送方向的指定

主控设备在 7 位地址之后发送 1 位指定传送方向的数据。

当此传送方向指定位为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定位为“1”时，表示主控设备从从属设备接收数据。

图 12-17 传送方向的指定



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

备注 n=0、1

### 12.5.4 应答 ( $\overline{\text{ACK}}$ )

能通过应答 ( $\overline{\text{ACK}}$ ) 确认发送方和接收方的串行数据状态。

接收方在每次接收到 8 位数据时返回应答。

通常，发送方在发送 8 位数据后接收应答。当接收方返回应答时，认为已正常接收，继续处理。能通过 IICA 状态寄存器 n (IICSn) 的 bit2 (ACKDn) 确认应答的检测。

在主机设备为接收状态下接收到最后的数据时，不返回应答而生成停止条件。在从属设备接收数据后不返回应答时，主机设备输出停止条件或者重新开始条件，中止发送。不返回应答的原因如下：

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

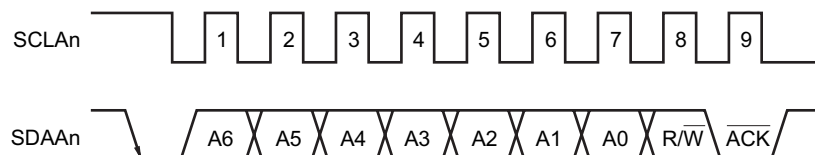
接收方在第 9 个时钟将 SDAAn 线置为低电平，生成应答（正常接收）。

通过将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit2 (ACKEn) 置“1”，变为能自动生成应答的状态。能通过 7 位地址信息后续的第 8 位数据设定 IICSn 寄存器的 bit3 (TRCn)。在接收 (TRCn=0) 的情况下，通常必须将 ACKEn 位置“1”。

在从属接收运行过程中 (TRCn=0) 不能接收数据或者不需要下一个数据时，必须将 ACKEn 位清“0”，通知主机方不能接收数据。

在主机接收运行过程中 (TRCn=0) 不需要下一个数据时，为了不生成应答，必须将 ACKEn 位清“0”，通知从属发送方数据的结束（停止发送）。

图 12-18 应答



当接收到本地站的地址时，与 ACKEn 位的值无关，自动生成应答；当接收到非本地站的地址时，不生成应答 (NACK)。

在接收到扩展码时，通过事先将 ACKEn 位置“1”，生成应答。

接收数据时的应答生成方法因等待时序的设定而不同，如下所示。

- 当选择 8 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) = 0)：  
通过在解除等待前将 ACKEn 位置“1”，与 SCLAn 引脚的第 8 个时钟下降同步生成应答。
- 当选择 9 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) = 1)：  
通过事先将 ACKEn 位置“1”，生成应答。

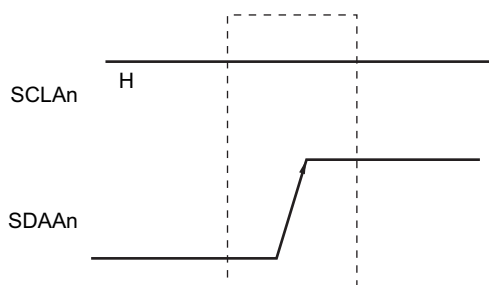
备注 n=0、1

### 12.5.5 停止条件

在 SCLAn 引脚为高电平时，如果 SDAAn 引脚从低电平变为高电平，就生成停止条件。

停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图 12-19 停止条件



如果将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit0 (SPTn) 置“1”，就生成停止条件。如果检测到停止条件，就将 IICA 状态寄存器 n (IICSn) 的 bit0 (SPDn) 置“1”，并且在 IICCTLn0 寄存器的 bit4 (SPIEn) 为“1”时产生 INTIICAn。

备注 n=0、1

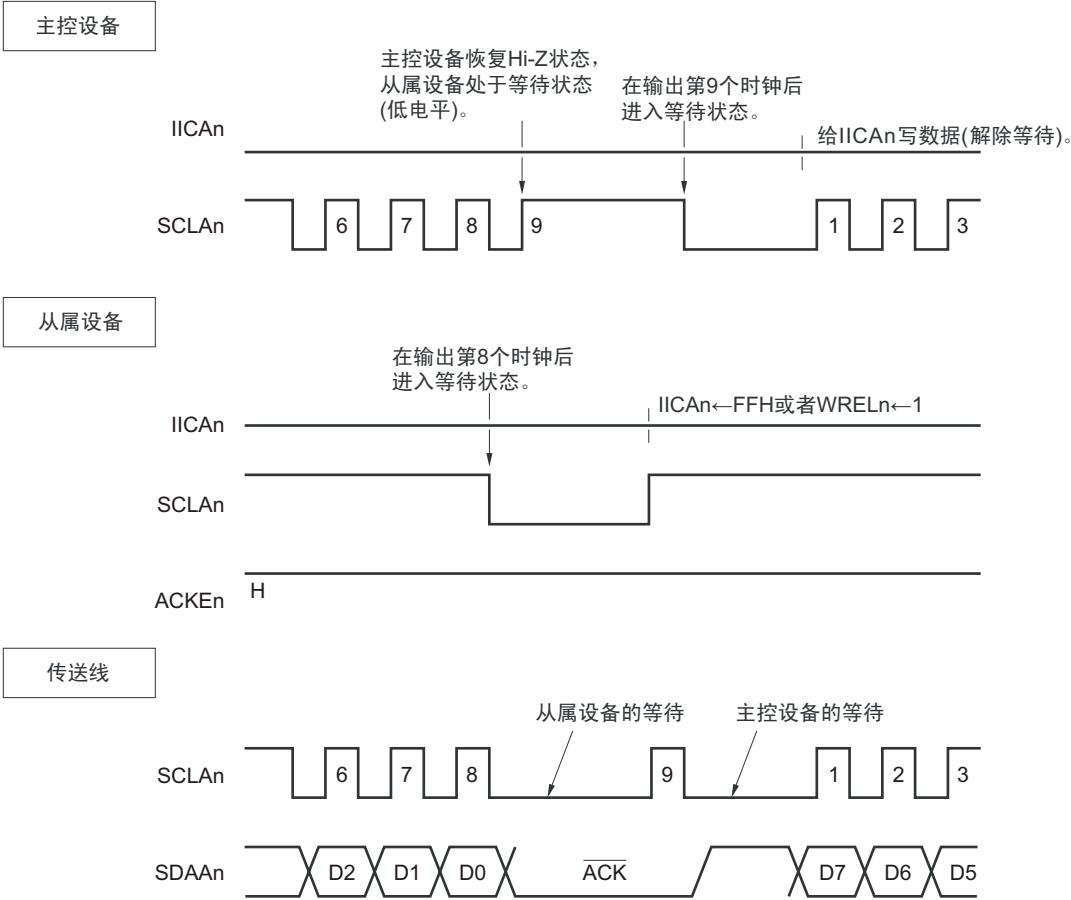
12.5.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送 / 接收（等待状态）。

通过将 SCLAn 引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图 12-20 等待 (1/2)

- (1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况  
(主控设备：发送，从属设备：接收，ACKEn=1)

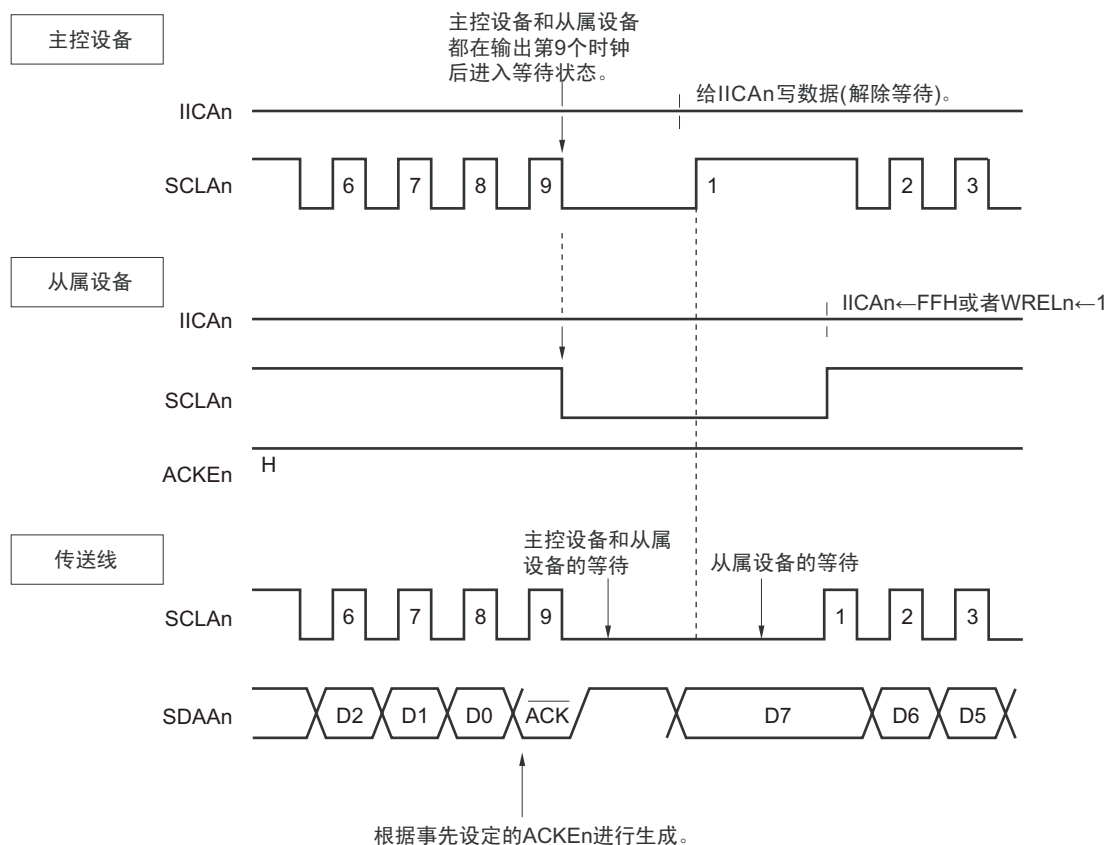


备注 n=0、1

图 12-20 等待 (2/2)

## (2) 主控设备和从属设备都为 9 个时钟等待的情况

(主控设备: 发送, 从属设备: 接收, ACKEn=1)



备注 ACKEn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit2

WRELn : IICA 控制寄存器 n0 (IICCTLn0) 的 bit5

通过设定 IICA 控制寄存器 n0 (IICCTLn0) 的 bit3 (WTIMn) 自动产生等待。

通常, 在接收方, 如果 IICCTLn0 寄存器的 bit5 (WRELn) 为“1”或者给 IICA 移位寄存器 n (IICAn) 写“FFH”, 就解除等待; 在发送方, 如果给 IICAn 寄存器写数据, 就解除等待。

主控设备还能通过以下方法解除等待:

- 将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”。
- 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

备注 n=0、1

### 12.5.7 等待的解除方法

在一般情况下，I<sup>2</sup>C 通过以下的处理来解除等待。

- 给 IICA 移位寄存器 n（IICAn）写数据。
- 将 IICA 控制寄存器 n0（IICCTLn0）的 bit5（WRELn）置位（解除等待）。
- 将 IICCTLn0 寄存器的 bit1（STTn）置位（生成开始条件）注。
- 将 IICCTLn0 寄存器的 bit0（SPTn）置位（生成停止条件）注。

注 只限于主控设备。

如果执行了这些等待的解除处理，I<sup>2</sup>C 就解除等待，重新开始通信。

要在解除等待后发送数据（包括地址）时，必须给 IICAn 寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将 IICCTLn0 寄存器的 bit5（WRELn）置“1”。

要在解除等待后生成重新开始条件时，必须将 IICCTLn0 寄存器的 bit1（STTn）置“1”。

要在解除等待后生成停止条件时，必须将 IICCTLn0 寄存器的 bit0（SPTn）置“1”。

对于一次等待只能执行一次解除处理。

例如，如果在通过将 WRELn 位置“1”来解除等待后给 IICAn 寄存器写数据，SDAAn 线的变化时序与 IICAn 寄存器的写时序就可能发生冲突，导致将错误的值输出到 SDAAn 线。

除了这些处理以外，在中途中止通信的情况下，如果将 IICEn 位清“0”，就停止通信，因此能解除等待。

在 I<sup>2</sup>C 总线状态因噪声而被死锁的情况下，如果将 IICCTLn0 寄存器的 bit6（LRELn）置“1”，就退出通信，因此能解除等待。

注意 如果在 WUPn 位为“1”时执行等待的解除处理，就不解除等待。

备注 n=0、1

### 12.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设定 IICA 控制寄存器 n0（IICCTLn0）的 bit3（WTIMn），在表 12-2 所示的时序产生 INTIICAn 并且进行等待控制。

表 12-2 INTIICAn 的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 注 1、2	8 注 2	8 注 2	9	8	8
1	9 注 1、2	9 注 2	9 注 2	9	9	9

注 1. 只有在接收的地址和从属地址寄存器 n（SVAn）的设定地址相同时，从属设备才在第 9 个时钟的下降沿产生 INTIICAn 信号并且进入等待状态。

此时，与 IICCTLn0 寄存器 bit2（ACKEn）的设定无关，生成应答。接收到扩展码的从属设备在第 8 个时钟的下降沿产生 INTIICAn。如果在重新开始后地址不同，就在第 9 个时钟的下降沿产生 INTIICAn，但是不进入等待状态。

2. 如果接收的地址和从属地址寄存器 n（SVAn）的内容不同并且未接收到扩展码，就不产生 INTIICAn 并且也不进入等待状态。

备注 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

#### (1) 地址的发送和接收

- 从属运行：与 WTIMn 位无关，根据上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控运行：与 WTIMn 位无关，在第 9 个时钟的下降沿产生中断和等待的时序。

#### (2) 数据的接收

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

#### (3) 数据的发送

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

备注 n=0、1

#### (4) 等待的解除方法

等待的解除方法有以下 4 种：

- 给 IICA 移位寄存器 n (IICAn) 写数据。
- 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件) 注。
- 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件) 注。

注 只限于主控设备。

当选择 8 个时钟的等待 (WTIMn=0) 时，需要在解除等待前决定是否生成应答。

#### (5) 停止条件的检测

如果检测到停止条件，就产生 INTIICAn (只限于 SPIEn=1 的情况)。

### 12.5.9 地址匹配的检测方法

在 I<sup>2</sup>C 总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。

能通过硬件自动检测地址匹配。当主控设备送来的从属地址和从属地址寄存器 n (SVAn) 的设定地址相同或者接收到扩展码时，产生 INTIICAn 中断请求。

### 12.5.10 错误的检测

在 I<sup>2</sup>C 总线模式中，因为发送过程中的串行数据总线 (SDAAn) 的状态被取到发送器件的 IICA 移位寄存器 n (IICAn)，所以能通过将开始发送前和发送结束后的 IICA 数据进行比较来检测发送错误。此时，如果 2 个数据不同，就判断为发生了发送错误。

备注 n=0、1



12.5.11 扩展码

- (1) 当接收地址的高4位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志（EXCn）置“1”，并且在第8个时钟的下降沿产生中断请求（INTIICAn）。  
不影响保存在从属地址寄存器n（SVAn）的本地站地址。
- (2) 当SVAn寄存器的设定值为“11110xx0”时，如果通过10位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第8个时钟的下降沿产生中断请求（INTIICAn）。
  - 高4位数据相同 : EXCn=1
  - 7位数据相同 : COIn=1

备注 EXCn : IICA 状态寄存器 n（IICSn）的 bit5  
COIn : IICA 状态寄存器 n（IICSn）的 bit4

- (3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。  
如果在从属运行时接收到扩展码，即使地址不同也在参加通信。  
例如，在接收到扩展码后不想作为从属设备运行时，必须将 IICA 控制寄存器 n0（IICCTLn0）的 bit6（LRELn）置“1”，进入下一次通信的待机状态。

表 12-3 主要扩展码的位定义

从属地址	R/W 位	说明
0 0 0 0 0 0 0	0	全呼地址
1 1 1 1 0 x x	0	10 位从属地址的指定（地址认证时）
1 1 1 1 0 x x	1	10 位从属地址的指定（在地址相同后发行读命令时）

备注 1. 有关上述以外的扩展码，请参照 NXP 公司发行的 I<sup>2</sup>C 总线规格书。  
2. n=0、1

12.5.12 仲裁

当多个主控设备同时生成开始条件时（在 **STDn** 位变为“1”前将 **STTn** 位置“1”的情况），边调整时钟边进行主控设备的通信，直到数据不同为止。此运行称为仲裁。

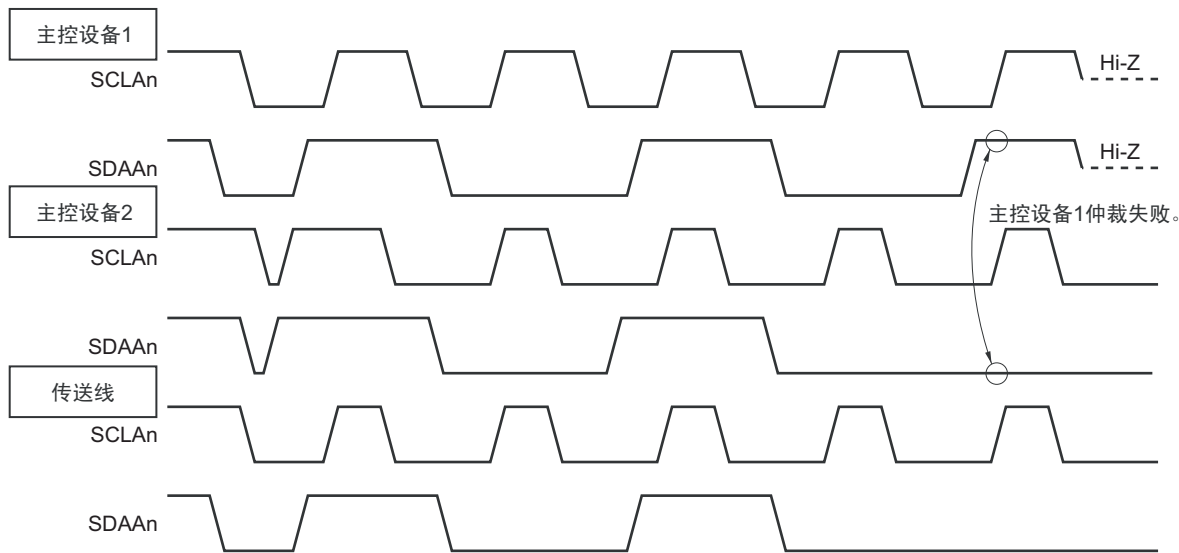
在仲裁失败时，仲裁失败的主控设备将 **IICA** 状态寄存器 **n**（**IICSn**）的仲裁失败标志（**ALDn**）置“1”，并且将 **SCLAn** 线和 **SDAAn** 线都置为高阻抗状态，释放总线。

在发生下一次中断请求时（例如：在第 8 或者第 9 个时钟检测到停止条件），用软件通过 **ALDn** 位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“12.5.8 中断请求（**INTIICAn**）的产生时序和等待控制”。

备注 **STDn**: **IICA** 状态寄存器 **n**（**IICSn**）的 bit1  
**STTn**: **IICA** 控制寄存器 **n0**（**IICCTLn0**）的 bit1

图 12-21 仲裁时序例子



备注 n=0、1

表 12-4 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第 8 或者第 9 个时钟的下降沿 <sup>注 1</sup>
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时（SPIEn=1） <sup>注 2</sup>
想要生成重新开始条件，但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 <sup>注 1</sup>
想要生成重新开始条件，但是检测到停止条件。	在生成停止条件时（SPIEn=1） <sup>注 2</sup>
想要生成停止条件，但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 <sup>注 1</sup>
想要生成重新开始条件，但是 SCLAn 为低电平。	

注 1. 当 WTIMn 位（IICA 控制寄存器 n0（IICCTLn0）的 bit3）为“1”时，在第 9 个时钟的下降沿产生中断请求；当 WTIMn 位为“0”并且接收到扩展码的从属地址时，在第 8 个时钟的下降沿产生中断请求。

2. 当有可能发生仲裁时，必须在主控运行时将 SPIEn 位置“1”。

备注 1. SPIEn: IICA 控制寄存器 n0（IICCTLn0）的 bit4

2. n=0、1

### 12.5.13 唤醒功能

这是 I<sup>2</sup>C 的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）的功能。

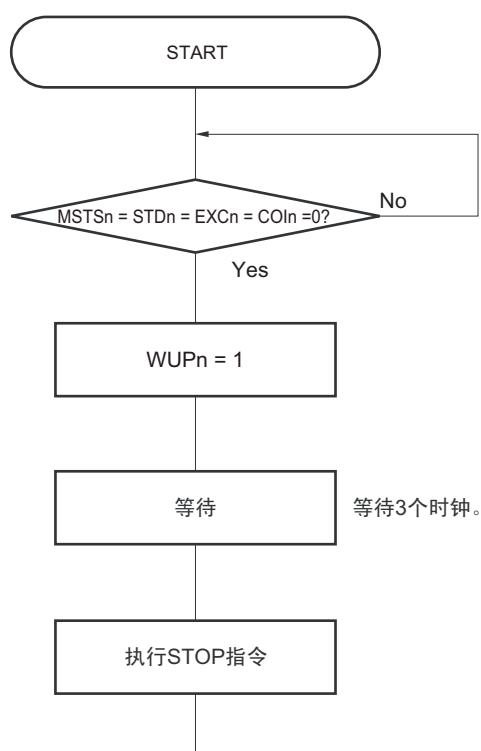
在地址不同的情况下不产生不需要的 INTIICAn 信号，能提高处理效率。

如果检测到开始条件，就进入唤醒待机状态。因为主控设备（已经生成开始条件的情况）也有可能因仲裁失败而变为从属设备，所以在发送地址的同时进入唤醒待机状态。

要在 STOP 模式中使用唤醒功能时，必须将 WUPn 位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）。在产生此中断后，通过指令将 WUPn 位清“0”，返回到通常运行。

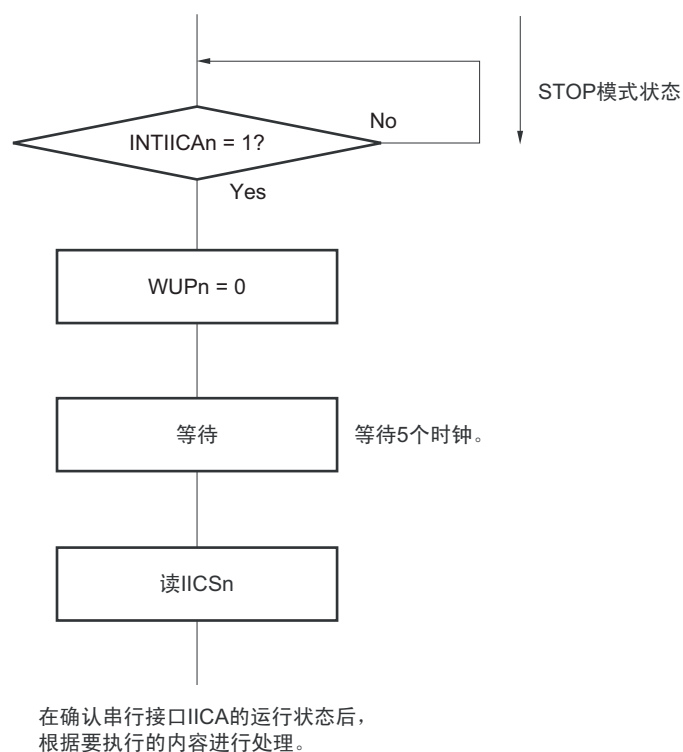
将 WUPn 位置“1”时的流程如图 12-22 所示，通过地址匹配将 WUPn 位置“0”时的流程如图 12-23 所示。

图 12-22 将 WUPn 位置“1”时的流程



备注 n=0、1

图 12-23 通过地址匹配将 WUPn 位置“0”时的流程（包括接收扩展码）

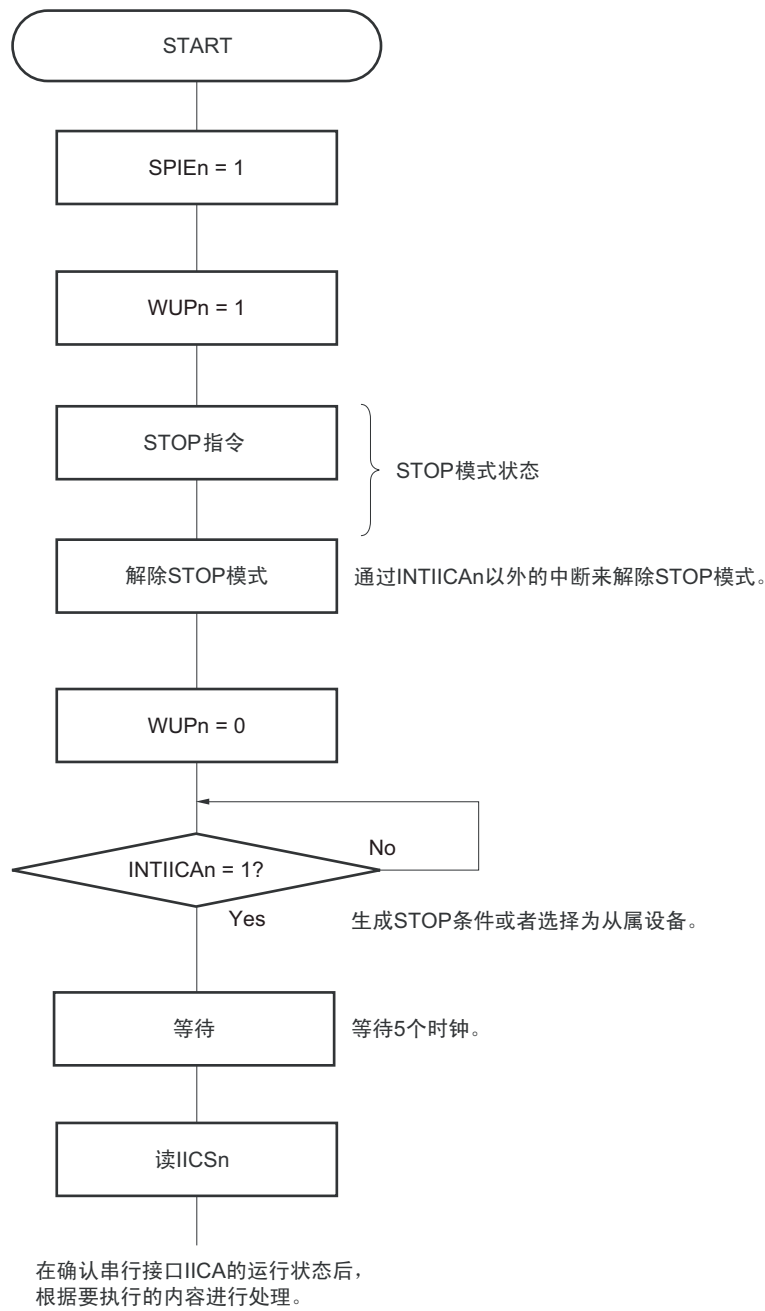


除了串行接口 IICA 产生的中断请求（INTIICAn）以外，必须通过以下的流程解除 STOP 模式。

- 作为主控设备运行的情况：图 12-24 的流程
- 作为从属设备运行的情况：和图 12-23 的流程相同。

备注 n=0、1

图 12-24 在通过 INTIICAn 以外的方法解除 STOP 模式后作为主控设备运行的情况



备注 n=0、1

### 12.5.14 通信预约

#### (1) 允许通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =0)

要在不加入总线的状态下进行下一次主控通信时，能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下 2 种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit6 (LRELn) 置“1”，退出通信后释放了总线)

如果在不加入总线的状态下将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将 IICCTLn0 寄存器的 bit4 (SPIEn) 置“1”，在通过产生的中断请求信号 (INTIICAn) 检测到总线的释放（检测到停止条件）后，如果给 IICA 移位寄存器 n (IICAn) 写地址，就自动作为主控设备开始通信。在检测到停止条件前，给 IICAn 寄存器写的的数据无效。

当将 STTn 位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时 ..... 生成开始条件
- 总线未处于释放状态（待机状态）时 ..... 通信预约

在将 STTn 位置“1”并且经过等待时间后，通过 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 确认是否作为通信预约运行。

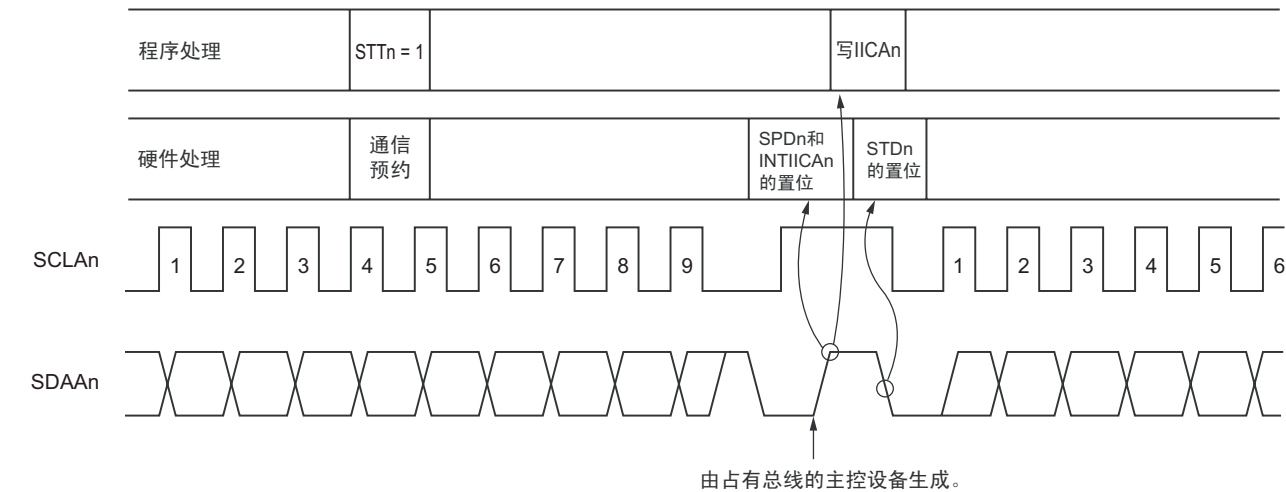
必须通过软件确保以下计算式计算的等待时间：

从将 STTn 位置“1”到确认 MSTSn 标志为止的等待时间：  
 $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) + t_F \times 2 \times f_{CLK} \text{ [ 时钟 ]}$

- 备注 1. IICWLn : IICA 低电平宽度设定寄存器 n  
 IICWHn : IICA 高电平宽度设定寄存器 n  
 $t_F$  : SDAAn 信号和 SCLAn 信号的下降时间  
 $f_{CLK}$  : CPU/ 外围硬件的时钟频率  
 2. n=0、1

通信预约的时序如图 12-25 所示。

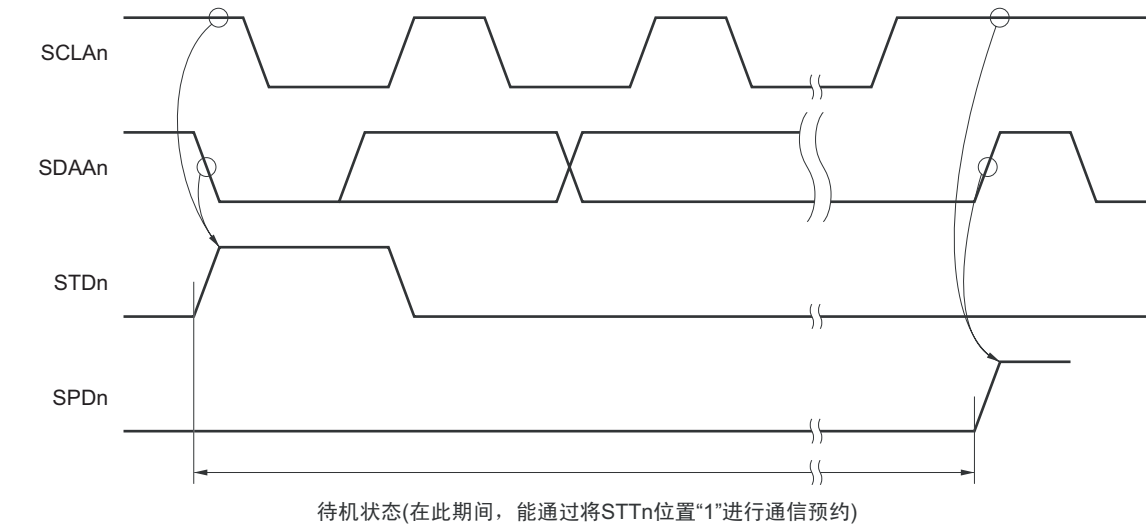
图 12-25 通信预约的时序



备注 IICAn : IICA 移位寄存器 n  
STTn : IICA控制寄存器 n0 (IICCTLn0) 的 bit1  
STDn : IICA 状态寄存器 n (IICSn) 的 bit1  
SPDn : IICA 状态寄存器 n (IICSn) 的 bit0

通过图 12-26 所示的时序接受通信预约。在 IICA 状态寄存器 n (IICSn) 的 bit1 (STDn) 变为“1”后并且在检测到停止条件前，将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”进行通信预约。

图 12-26 通信预约的接受时序

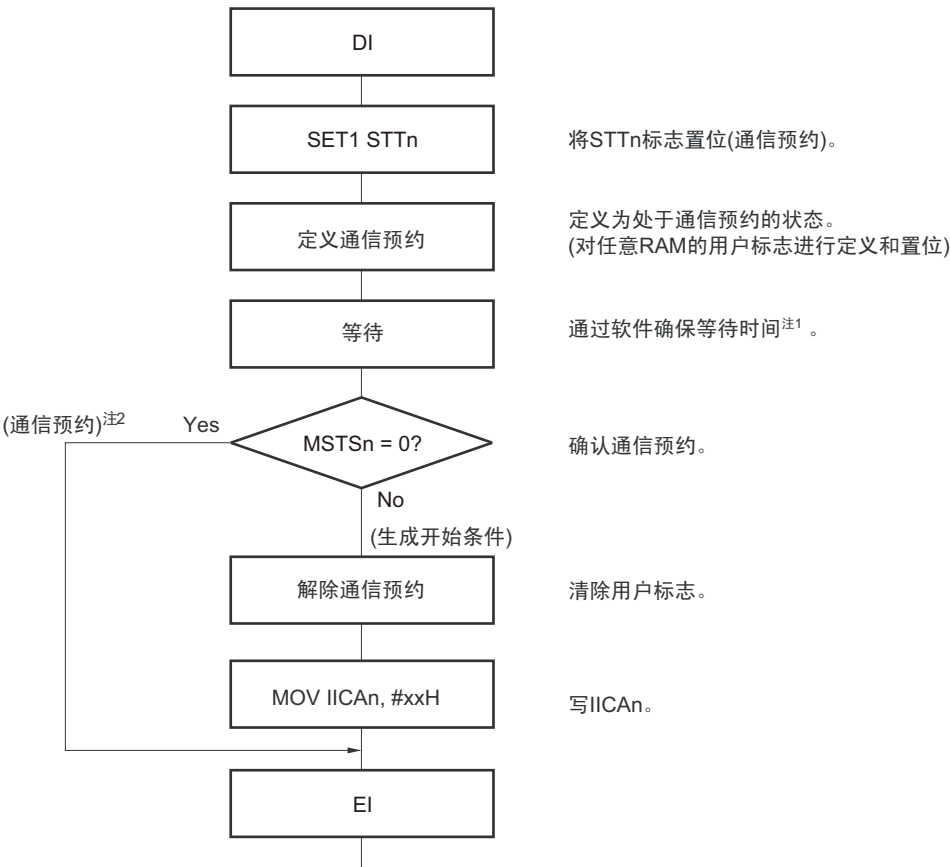


通信预约的步骤如图 12-27 所示。

备注 n=0、1



图 12-27 通信预约的步骤



- 注 1. 等待时间如下：  
(IICWL<sub>n</sub> 的设定值 + IICWH<sub>n</sub> 的设定值 + 4) + t<sub>F</sub> × 2 × f<sub>CLK</sub> [ 时钟 ]
2. 在通信预约运行时，通过停止条件中断请求写 IICA 移位寄存器 n（IICAn）。

- 备注 1. STT<sub>n</sub> : IICA 控制寄存器 n0（IICCTLn0）的 bit1  
MSTSn : IICA 状态寄存器 n（IICSn）的 bit7  
IICAn : IICA 移位寄存器 n  
IICWL<sub>n</sub> : IICA 低电平宽度设定寄存器 n  
IICWH<sub>n</sub> : IICA 高电平宽度设定寄存器 n  
t<sub>F</sub> : SDAAn 信号和 SCLAn 信号的下降时间  
f<sub>CLK</sub> : CPU/ 外围硬件的时钟频率
2. n=0、1

## (2) 禁止通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =1)

在总线通信过程中, 如果在不参加此通信的状态下将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”, 就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下 2 种状态:

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时 (不返回应答而将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 退出通信后释放了总线)

能通过 STCFn (IICFn 寄存器的 bit7) 来确认是生成了开始条件还是拒绝了请求。因为从 STTn 位为“1”到将 STCFn 位置“1”为止需要 5 个时钟的时间, 所以必须通过软件确保此时间。

备注 n=0、1

### 12.5.15 其他注意事项

#### (1) STCEN<sub>n</sub> 位为“0”的情况

在刚允许 I<sup>2</sup>C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSY<sub>n</sub>=1)。要在没有检测到停止条件的状态下进行主控通信时, 必须先生成停止条件, 在释放总线后进行主控通信。

对于多主控, 在总线未释放 (未检测到停止条件) 的状态下不能进行主控通信。

按照以下顺序生成停止条件:

- ① 设定 IICA 控制寄存器 n1 (IICCTLn1)。
- ② 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) 置“1”。
- ③ 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

#### (2) STCEN<sub>n</sub> 位为“1”的情况

在刚允许 I<sup>2</sup>C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSY<sub>n</sub>=0)。因此在生成第 1 个开始条件 (STTn=1) 时, 为了不破坏其他的通信, 需要确认总线已被释放。

#### (3) 正在和其他设备进行 I<sup>2</sup>C 通信的情况

在 SDAAn 引脚为低电平并且 SCLAn 引脚为高电平时, 如果允许 I<sup>2</sup>C 运行而且中途参加通信, I<sup>2</sup>C 的宏就视为 SDAAn 引脚从高电平变为低电平 (检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的 I<sup>2</sup>C 通信。为了避免这种情况, 必须按照以下顺序启动 I<sup>2</sup>C:

- ① 将 IICCTLn0 寄存器的 bit4 (SPIEn) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIICAn)。
- ② 将 IICCTLn0 寄存器的 bit7 (IICEn) 置“1”, 允许 I<sup>2</sup>C 运行。
- ③ 等待检测到开始条件。
- ④ 在返回应答前 (将 IICEn 位置“1”后的 4 ~ 80 个时钟内) 将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 强制将检测置为无效。

#### (4) 在将 STTn 位和 SPTn 位 (IICCTLn0 寄存器的 bit1 和 bit0) 置位后, 禁止清“0”前的再置位。

#### (5) 如果进行了通信预约, 就必须将 SPIEn 位 (IICCTLn0 寄存器的 bit4) 置“1”, 在检测到停止条件时产生中断请求。在产生中断请求后, 通过给 IICA 移位寄存器 n (IICAn) 写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因在开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 时, 不需要将 SPIEn 位置“1”。

备注 n=0、1

### 12.5.16 通信运行

在此通过流程图表示以下 3 个运行步骤。

#### (1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

#### (2) 多主控系统的主控运行

在 I<sup>2</sup>C 总线的多主控系统中，只根据 I<sup>2</sup>C 总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1 帧）为高电平，就将总线作为释放状态而参加通信。

此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

#### (3) 从属运行

用作 I<sup>2</sup>C 总线从属设备的例子如下所示。

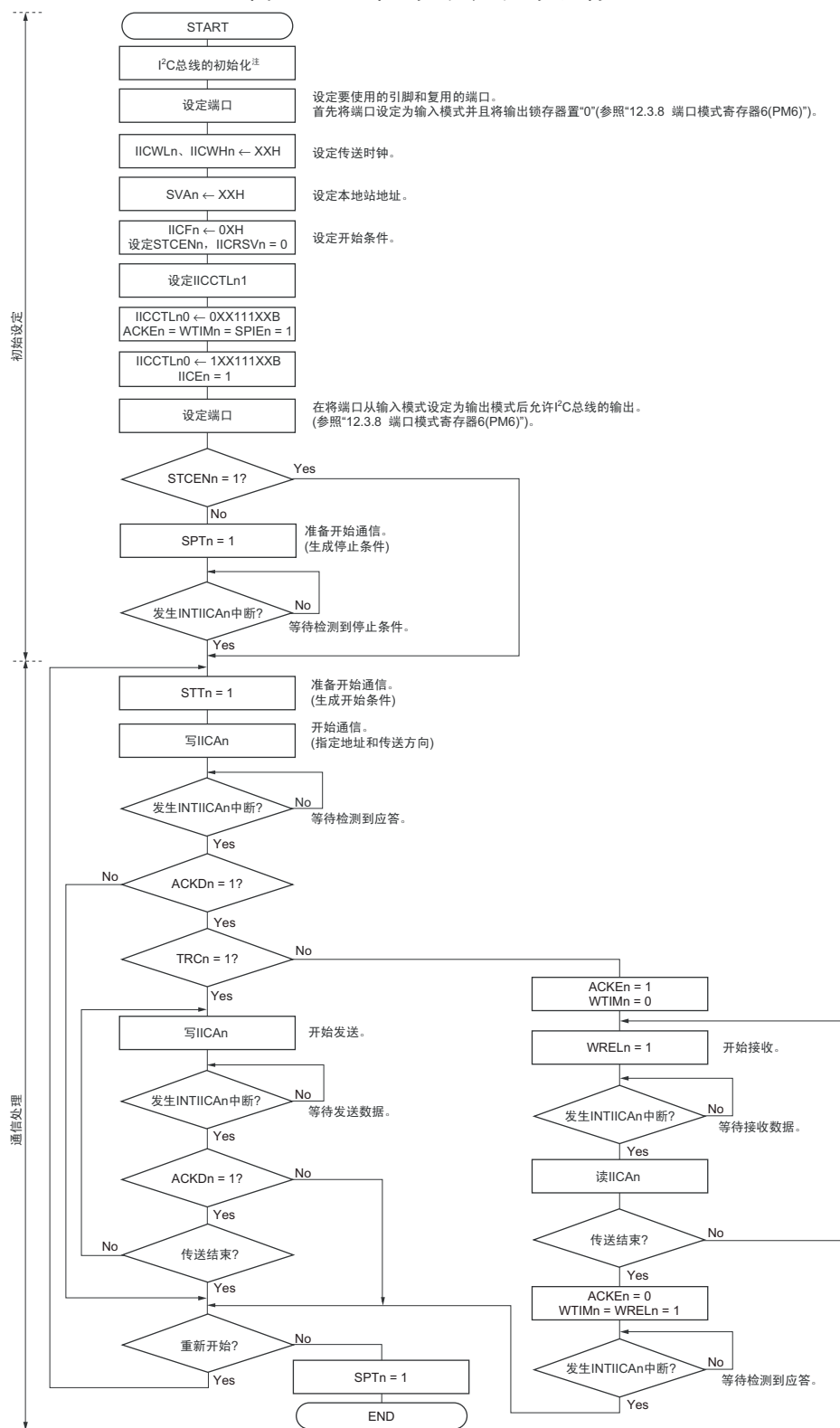
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注 n=0、1

## (1) 单主控系统的主控运行

图 12-28 单主控系统的主控运行



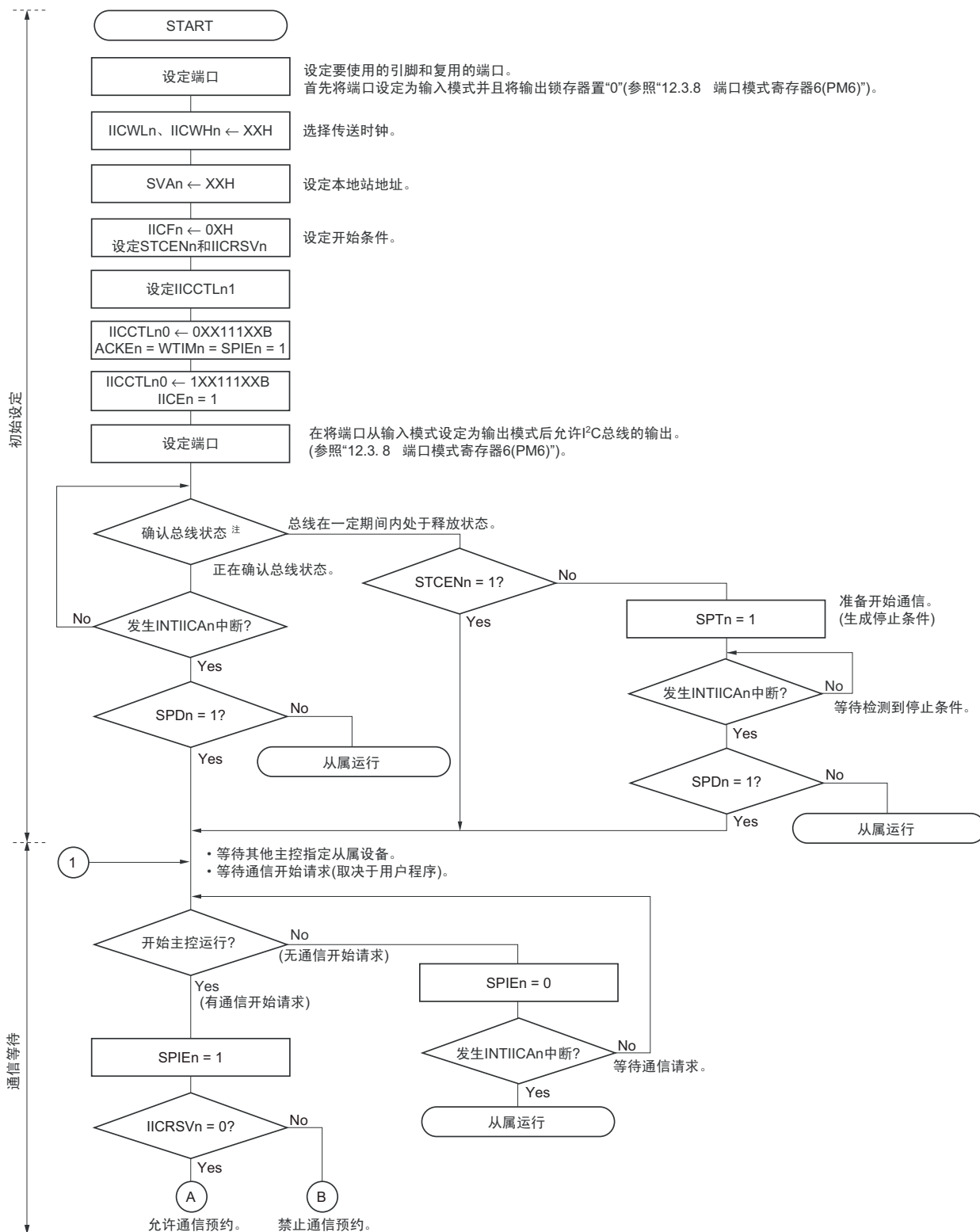
注 必须根据通信中的产品的规格，释放 I²C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。例如，如果 EEPROM 处于给 SDAAn 引脚输出低电平的状态，就必须将 SCLAn 引脚设定为输出端口，并且在 SDAAn 引脚固定为高电平前从输出端口输出时钟脉冲。

备注 1. 发送和接收的格式必须符合通信中的产品的规格。

2. n=0、1

## (2) 多主控系统的主控运行

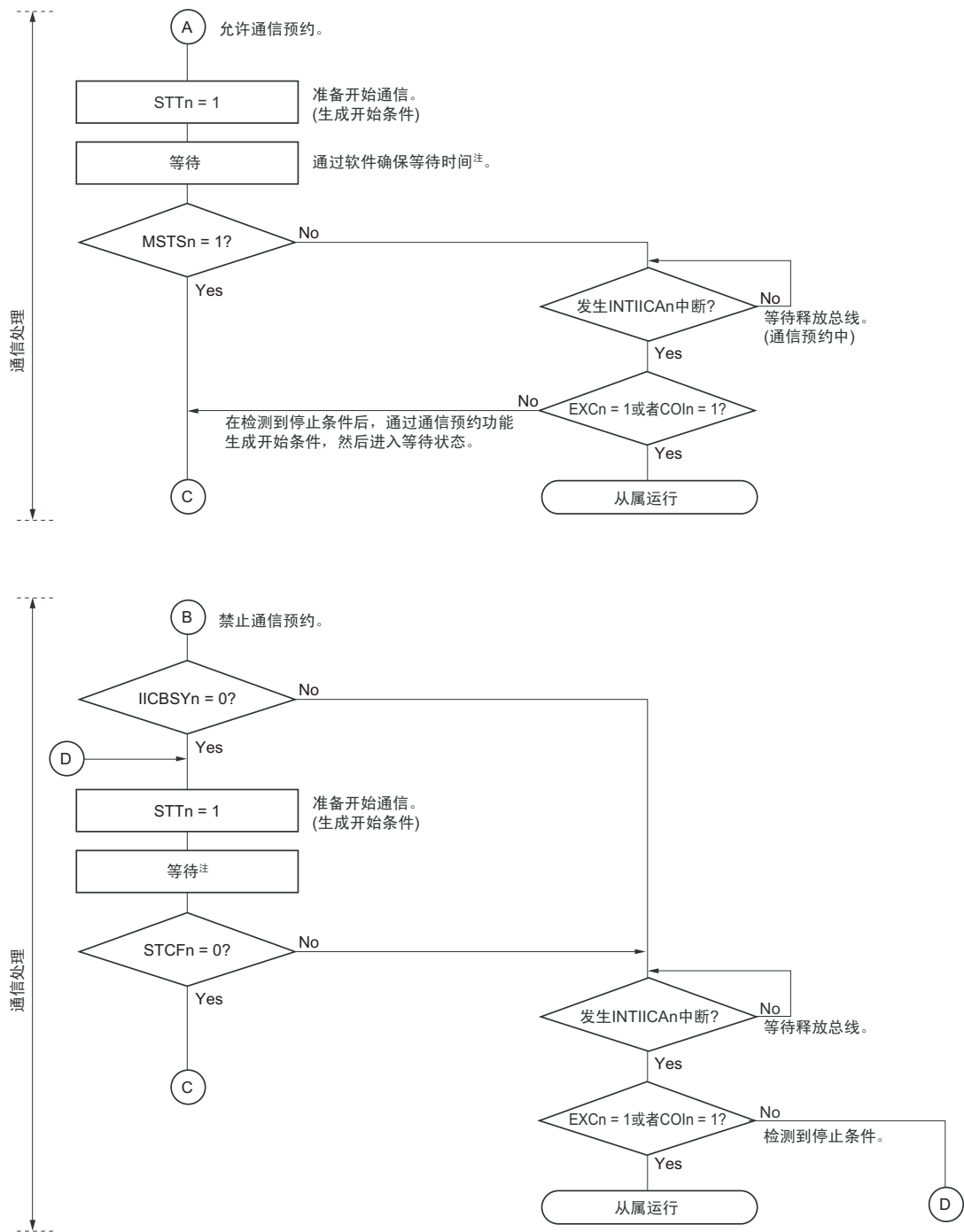
图 12-29 多主控系统的主控运行 (1/3)



注 必须确认总线在一定时间内（例如，1 帧）处于释放状态（CLDn 位 =1、DADn 位 =1）。当 SDAAn 引脚固定为低电平时，必须根据通信中的产品的规格，判断是否释放 I<sup>2</sup>C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。

备注 n=0、1

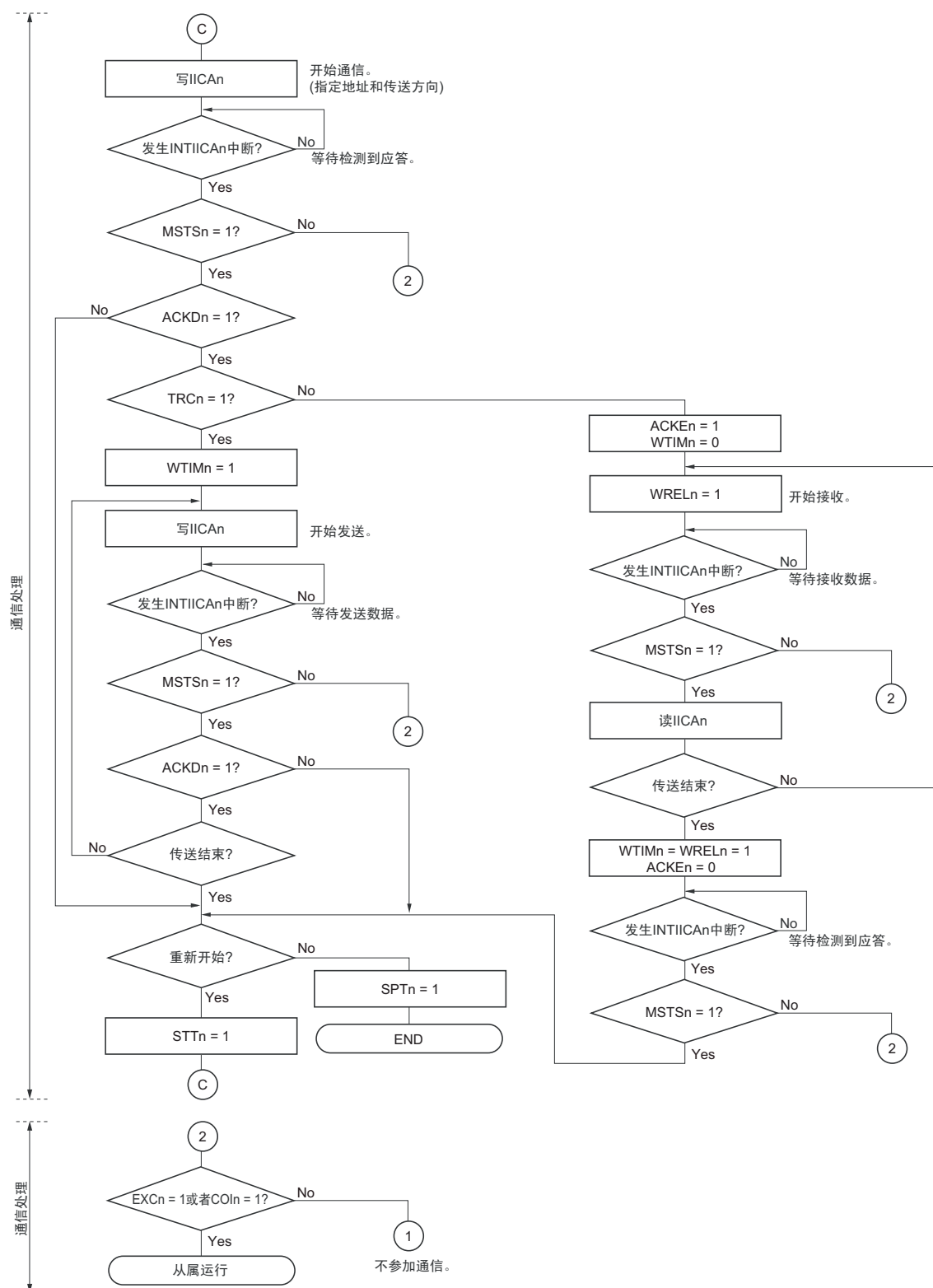
图 12-29 多主控系统的主控运行 (2/3)



注 等待时间如下：  
(IICWLn的设定值 + IICWHn的设定值 + 4个时钟) / f<sub>CLK</sub> + t<sub>F</sub> × 2

- 备注 1. IICWLn : IICA 低电平宽度设定寄存器 n  
IICWHn : IICA 高电平宽度设定寄存器 n  
t<sub>F</sub> : SDAAn 信号和 SCLAn 信号的下降时间  
f<sub>CLK</sub> : CPU/ 外围硬件的时钟频率  
2. n=0、1

图 12-29 多主控系统的主控运行 (3/3)



备注 1. 传送和接收的格式必须符合通信中的产品的规格。

2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIICAn 中断时读 MSTSn 位，确认仲裁结果。

3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIICAn 中断时通过 IICA 状态寄存器 n (IICS<sub>n</sub>) 和 IICA 标志寄存器 n (IICF<sub>n</sub>) 确认状态，决定以后的处理。

4. n=0、1

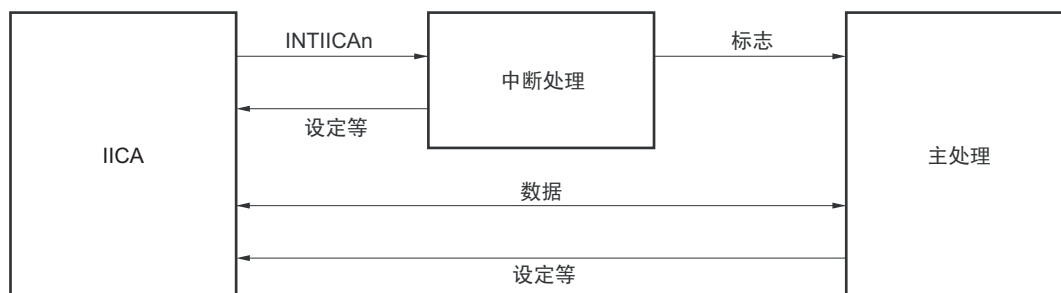


### (3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过 INTIICAn 中断进行处理（需要对通信中的停止条件检测等的运行状态进行很大的变更处理）。

在此说明中，假设数据通信不支持扩展码，INTIICAn 中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下 3 个标志并且代替 INTIICAn 将标志传递给主处理部，进行数据通信处理。

#### ① 通信模式标志

此标志表示以下 2 种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态（有效地址的检测～停止条件的检测，未检测到主控设备的应答，地址不同）

#### ② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和 INTIICAn 中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第 1 个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第 1 个数据（地址匹配被解释为下一次数据请求）。

#### ③ 通信方向标志

此标志表示通信方向，和 TRCn 位的值相同。

备注 n=0、1

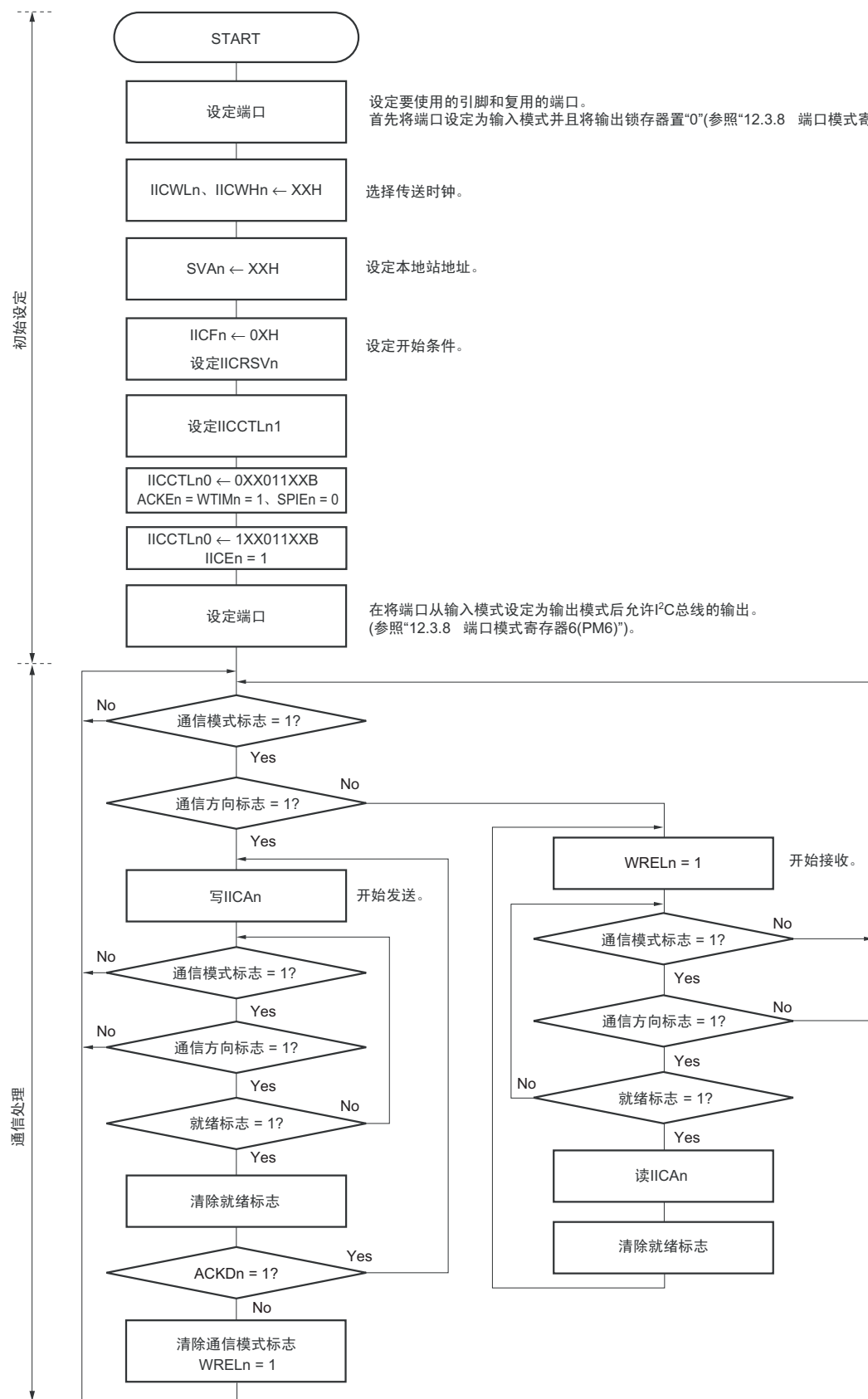
从属运行的主处理部的运行如下所示。

启动串行接口 IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。

在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图 12-30 从属运行步骤 (1)



备注 1. 传送和接收的格式必须符合通信中的产品的规格。

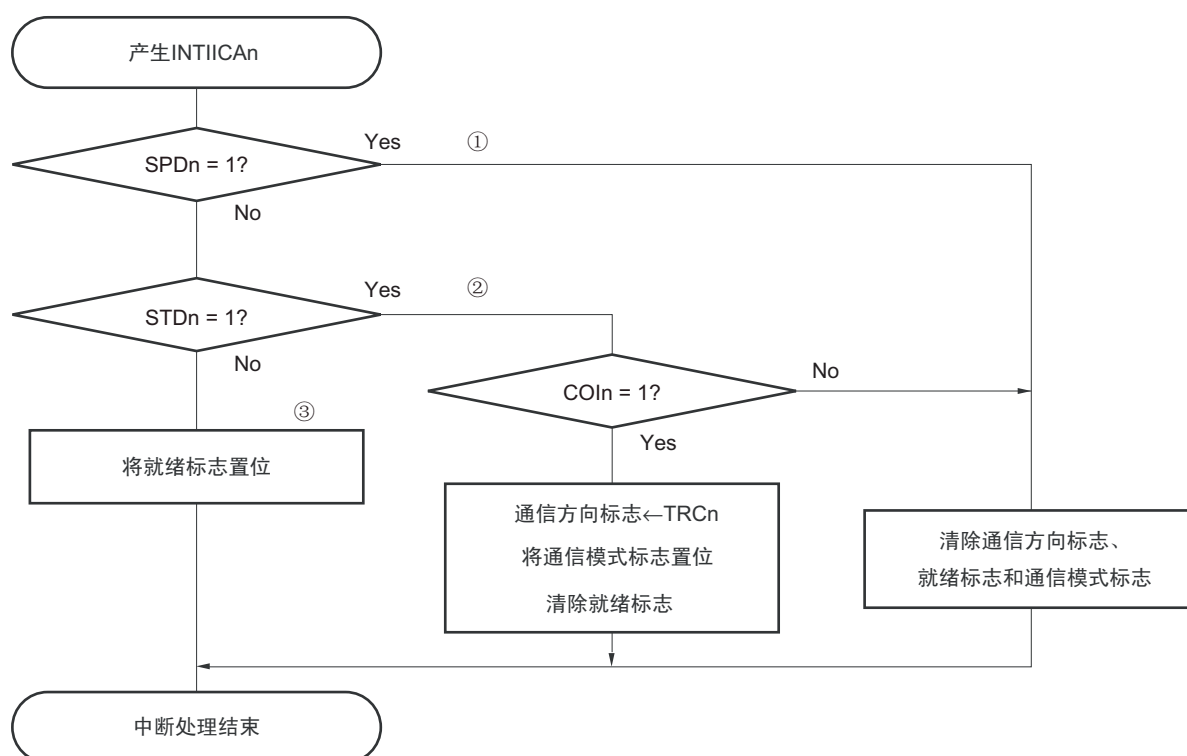
2. n=0、1

从属设备通过 INTIICAn 中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过 INTIICAn 中断确认状态并且进行以下处理。

- ① 如果生成停止条件，就结束通信。
- ② 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设定为通信模式并且解除等待，然后从中断返回（清除就绪标志）。
- ③ 当发送和接收数据时，只要将就绪标志置位，I<sup>2</sup>C 总线就保持等待状态并且从中断返回。

备注 上述的①～③对应“图 12-31 从属运行步骤 (2)”的①～③。

图 12-31 从属运行步骤 (2)



备注 n=0、1

### 12.5.17 I<sup>2</sup>C 中断请求（INTIICAn）的产生时序

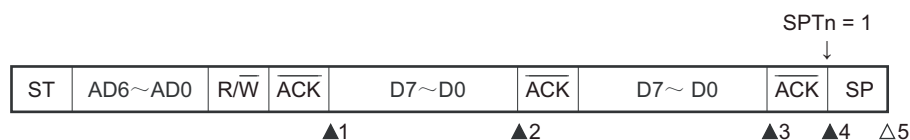
数据的发送和接收时序、INTIICAn 中断请求信号的产生时序以及产生 INTIICAn 信号时的 IICA 状态寄存器 n（IICSn）的值如下所示。

- 备注 1. ST : 开始条件  
AD6 ~ AD0 : 地址  
 $\overline{R/W}$  : 传送方向的指定  
 $\overline{ACK}$  : 应答  
D7 ~ D0 : 数据  
SP : 停止条件
2. n=0、1

## (1) 主控运行

## (a) Start ~ Address ~ Data ~ Data ~ Stop (发送和接收)

## (i) WTIMn=0 的情况



▲ 1: IICSn=1000×110B

▲ 2: IICSn=1000×000B

▲ 3: IICSn=1000×000B (将 WTIMn 位置“1”) 注

▲ 4: IICSn=1000××00B (将 SPTn 位置“1”)

△ 5: IICSn=00000001B

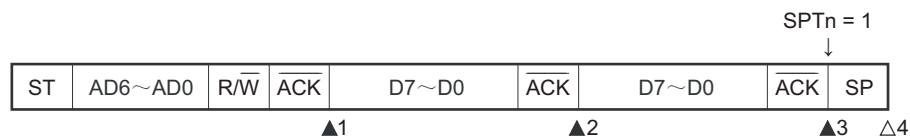
注 为了生成停止条件，必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

## (ii) WTIMn=1 的情况



▲ 1: IICSn=1000×110B

▲ 2: IICSn=1000×100B

▲ 3: IICSn=1000××00B (将 SPTn 位置“1”)

△ 4: IICSn=00000001B

备注 ▲ 一定产生。

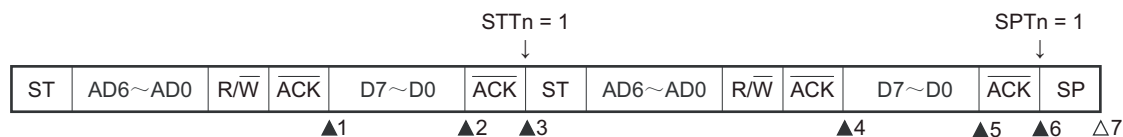
△ 只在 SPIEn 位为“1”时产生。

× 任意

备注 n=0、1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (重新开始)

(i) WTIMn=0 的情况

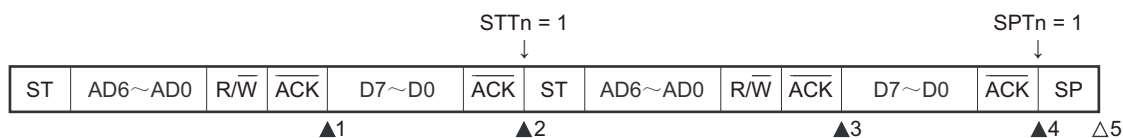


- ▲ 1: IICSn=1000×110B  
 ▲ 2: IICSn=1000×000B (将 WTIMn 位置“1”) 注 1  
 ▲ 3: IICSn=1000××00B (将 WTIMn 位清“0” 注 2 并且将 STTn 位置“1”)  
 ▲ 4: IICSn=1000×110B  
 ▲ 5: IICSn=1000×000B (将 WTIMn 位置“1”) 注 3  
 ▲ 6: IICSn=1000××00B (将 SPTn 位置“1”)  
 △ 7: IICSn=00000001B

- 注 1. 为了生成开始条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。  
 2. 为了恢复原来的设定, 必须将 WTIMn 位清“0”。  
 3. 为了生成停止条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

(ii) WTIMn=1 的情况



- ▲ 1: IICSn=1000×110B  
 ▲ 2: IICSn=1000××00B (将 STTn 位置“1”)  
 ▲ 3: IICSn=1000×110B  
 ▲ 4: IICSn=1000××00B (将 SPTn 位置“1”)  
 △ 5: IICSn=00000001B

备注 ▲ 一定产生。  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

备注 n=0、1

## (c) Start ~ Code ~ Data ~ Data ~ Stop (发送扩展码)

## (i) WTIMn=0 的情况



▲ 1: IICSn=1010×110B

▲ 2: IICSn=1010×000B

▲ 3: IICSn=1010×000B (将 WTIMn 位置“1”) 注

▲ 4: IICSn=1010××00B (将 SPTn 位置“1”)

△ 5: IICSn=00000001B

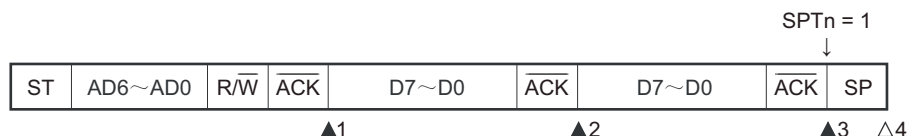
注 为了生成停止条件，必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

## (ii) WTIMn=1 的情况



▲ 1: IICSn=1010×110B

▲ 2: IICSn=1010×100B

▲ 3: IICSn=1010××00B (将 SPTn 位置“1”)

△ 4: IICSn=00001001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

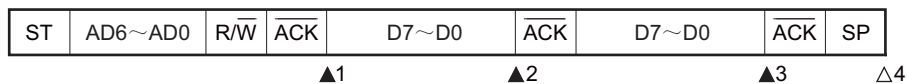
× 任意

备注 n=0、1

## (2) 从属运行（接收从属地址的情况）

## (a) Start ~ Address ~ Data ~ Data ~ Stop

## (i) WTIMn=0 的情况



▲ 1: IICSn=0001×110B

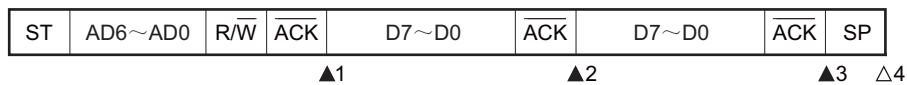
▲ 2: IICSn=0001×000B

▲ 3: IICSn=0001×000B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

## (ii) WTIMn=1 的情况



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×100B

▲ 3: IICSn=0001××00B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

备注 n=0、1



(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0 的情况（在重新开始后 SVAn 相同）



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×000B

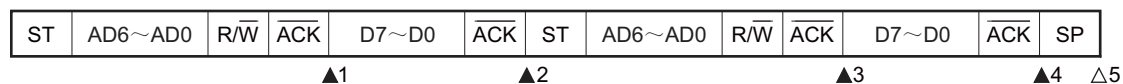
▲ 3: IICSn=0001×110B

▲ 4: IICSn=0001×000B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意

(ii) WTIMn=1 的情况（在重新开始后 SVAn 相同）



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001××00B

▲ 3: IICSn=0001×110B

▲ 4: IICSn=0001××00B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意

备注 n=0、1

## (c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

## (i) WTIMn=0的情况（在重新开始后地址不同（扩展码））



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001×000B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×000B

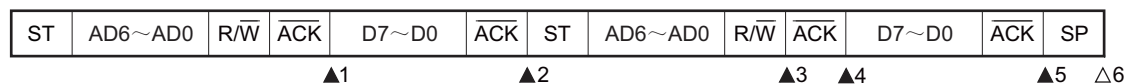
△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

× 任意

## (ii) WTIMn=1的情况（在重新开始后地址不同（扩展码））



▲ 1: IICSn=0001×110B

▲ 2: IICSn=0001××00B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×110B

▲ 5: IICSn=0010××00B

△ 6: IICSn=00000001B

备注 ▲ 一定产生。

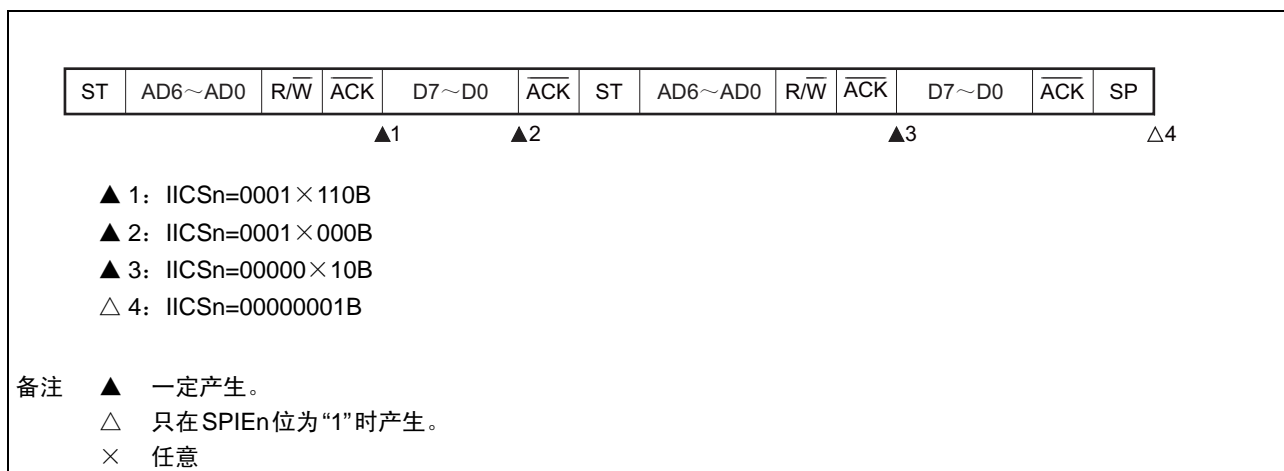
△ 只在SPIEn位为“1”时产生。

× 任意

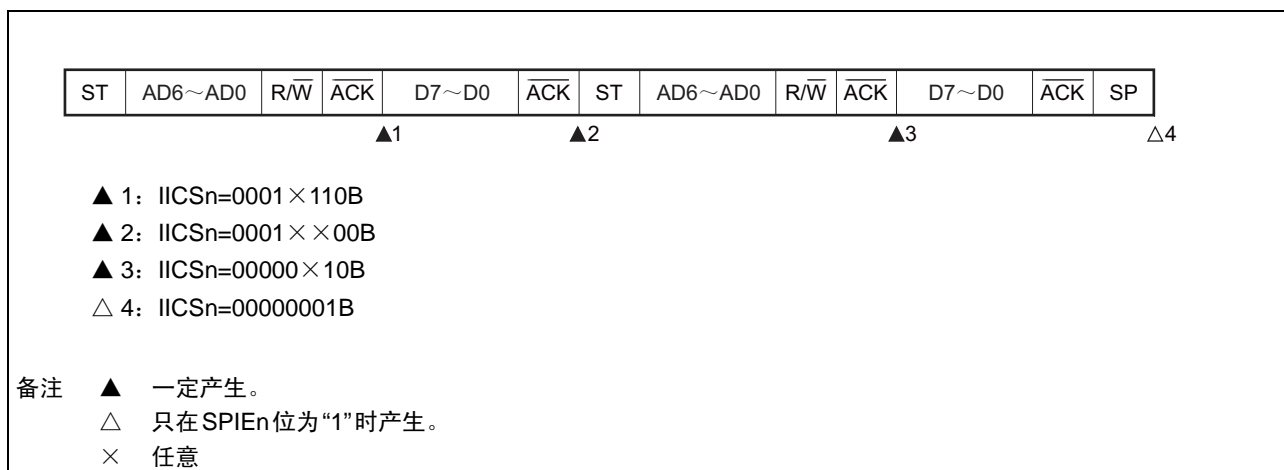
备注 n=0、1

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0 的情况（在重新开始后地址不同（非扩展码））



(ii) WTIMn=1 的情况（在重新开始后地址不同（非扩展码））



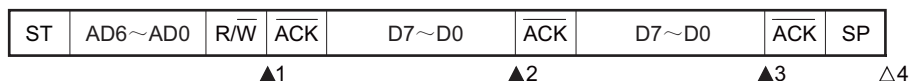
备注 n=0、1

## (3) 从属运行（接收扩展码的情况）

在接收扩展码时，始终参加通信。

## (a) Start ~ Code ~ Data ~ Data ~ Stop

## (i) WTIMn=0 的情况



▲ 1: IICSn=0010×010B

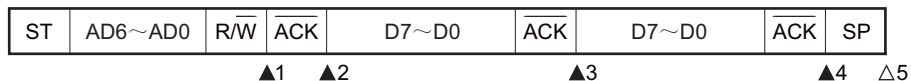
▲ 2: IICSn=0010×000B

▲ 3: IICSn=0010×000B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意

## (ii) WTIMn=1 的情况



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010×100B

▲ 4: IICSn=0010××00B

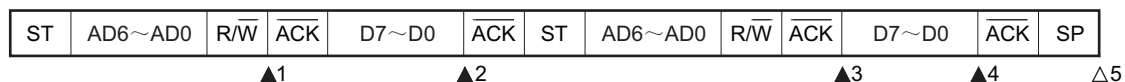
△ 5: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意

备注 n=0、1

## (b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

## (i) WTIMn=0 的情况 (在重新开始后 SVAn 相同)



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×000B

▲ 3: IICSn=0001×110B

▲ 4: IICSn=0001×000B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

## (ii) WTIMn=1 的情况 (在重新开始后 SVAn 相同)



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010××00B

▲ 4: IICSn=0001×110B

▲ 5: IICSn=0001××00B

△ 6: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

备注 n=0、1

## (c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

## (i) WTIMn=0 的情况（在重新开始后接收扩展码）



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×000B

▲ 3: IICSn=0010×010B

▲ 4: IICSn=0010×000B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

## (ii) WTIMn=1 的情况（在重新开始后接收扩展码）



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010××00B

▲ 4: IICSn=0010×010B

▲ 5: IICSn=0010×110B

▲ 6: IICSn=0010××00B

△ 7: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

备注 n=0、1

## (d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

## (i) WTIMn=0的情况（在重新开始后地址不同（非扩展码））



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×000B

▲ 3: IICSn=00000×10B

△ 4: IICSn=00000001B

备注 ▲ 一定产生。

△ 只在SPIEn位为“1”时产生。

× 任意

## (ii) WTIMn=1的情况（在重新开始后地址不同（非扩展码））



▲ 1: IICSn=0010×010B

▲ 2: IICSn=0010×110B

▲ 3: IICSn=0010××00B

▲ 4: IICSn=00000×10B

△ 5: IICSn=00000001B

备注 ▲ 一定产生。

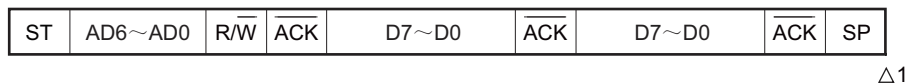
△ 只在SPIEn位为“1”时产生。

× 任意

备注 n=0、1

(4) 不参加通信的运行

(a) Start ~ Code ~ Data ~ Data ~ Stop



△ 1: IICSn=00000001B

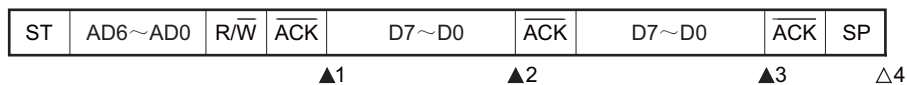
备注 △: 只在 SPIEn 位为“1”时产生。

(5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况

(i) WTIMn=0 的情况



▲ 1: IICSn=0101×110B

▲ 2: IICSn=0001×000B

▲ 3: IICSn=0001×000B

△ 4: IICSn=00000001B

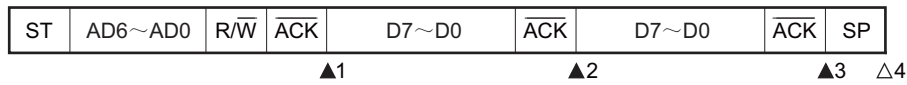
备注 ▲ 一定产生。

△ 只在 SPIEn 位为“1”时产生。

× 任意

备注 n=0、1



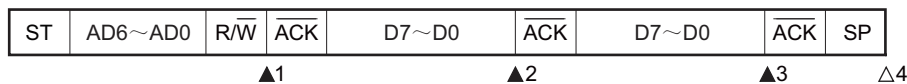
(ii) WTIM<sub>n</sub>=1 的情况

- ▲ 1: IICSn=0101×110B
- ▲ 2: IICSn=0001×100B
- ▲ 3: IICSn=0001××00B
- △ 4: IICSn=00000001B

备注	▲ 一定产生。
	△ 只在SPIEn位为“1”时产生。
	× 任意

(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIM<sub>n</sub>=0 的情况



- ▲ 1: IICSn=0110×010B  
▲ 2: IICSn=0010×000B  
▲ 3: IICSn=0010×000B  
△ 4: IICSn=00000001B

备注	▲ 一定产生。
	△ 只在SPIEn位为“1”时产生。
	× 任意

备注  $n=0, 1$

(ii) WTIMn=1 的情况

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1 ▲2		▲3		▲4	△5

▲ 1: IICSn=0110×010B  
▲ 2: IICSn=0010×110B  
▲ 3: IICSn=0010×100B  
▲ 4: IICSn=0010××00B  
△ 5: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意

(6) 仲裁失败的运行（在仲裁失败后不参加通信）

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
			▲1					△2

▲ 1: IICSn=01000110B  
△ 2: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。

备注 n=0、1

(b) 在发送扩展码的过程中仲裁失败的情况

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
----	---------	-----	-----	-------	-----	-------	-----	----

▲1

△2

▲ 1: IICSn=0110×010B  
通过软件将 LRELn 位置“1”。

△ 2: IICSn=00000001B

备注   ▲   一定产生。

         △   只在 SPIEn 位为“1”时产生。

         ×   任意

(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0 的情况

ST	AD6~AD0	R/W	ACK	D7~D0	ACK	D7~D0	ACK	SP
----	---------	-----	-----	-------	-----	-------	-----	----

▲1

▲2

△3

▲ 1: IICSn=10001110B

▲ 2: IICSn=01000000B

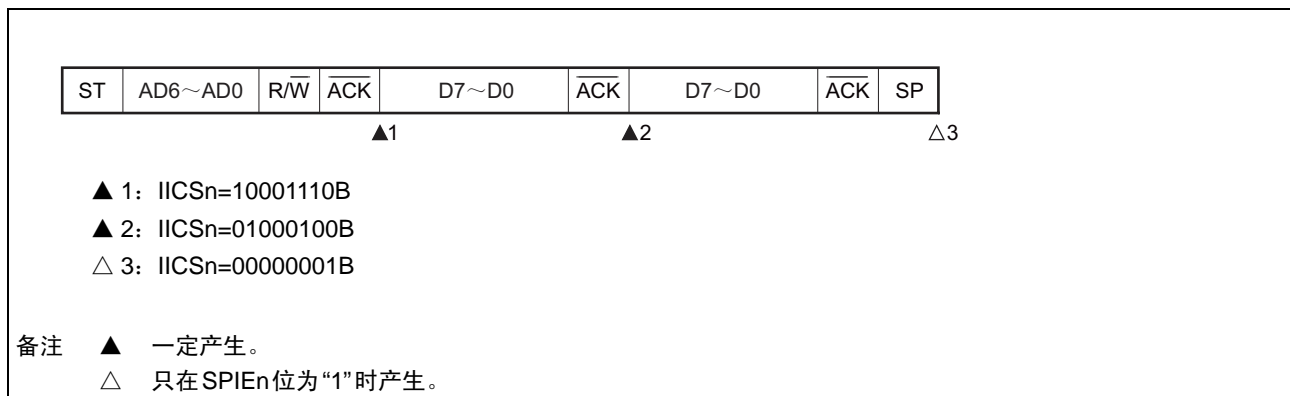
△ 3: IICSn=00000001B

备注   ▲   一定产生。

         △   只在 SPIEn 位为“1”时产生。

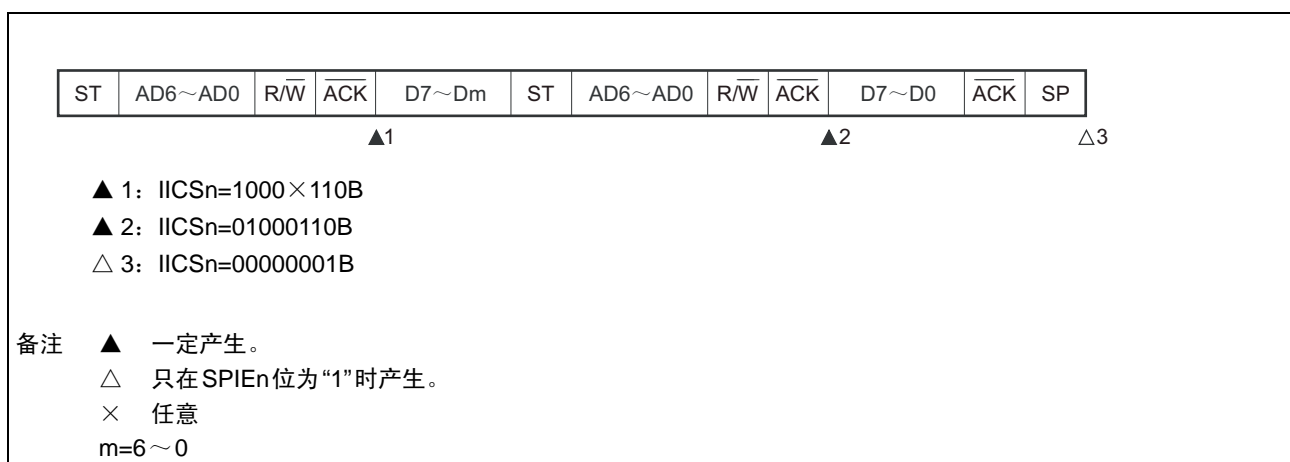
备注   n=0、1

(ii) WTIM<sub>n=1</sub>



(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码 (例如,  $SVAn$  不同)



备注  $n=0、1$

(ii) 扩展码

ST	AD6~AD0	R/W	ACK	D7~Dm	ST	AD6~AD0	R/W	ACK	D7~D0	ACK	SP
			▲1				▲2			△3	

▲ 1: IICSn=1000×110B  
▲ 2: IICSn=01100010B  
通过软件将 LRELn 位置“1”。  
△ 3: IICSn=00000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意  
m=6~0

(e) 在传送数据时因停止条件而仲裁失败的情况

ST	AD6~AD0	R/W	ACK	D7~Dm	SP
			▲1		△2

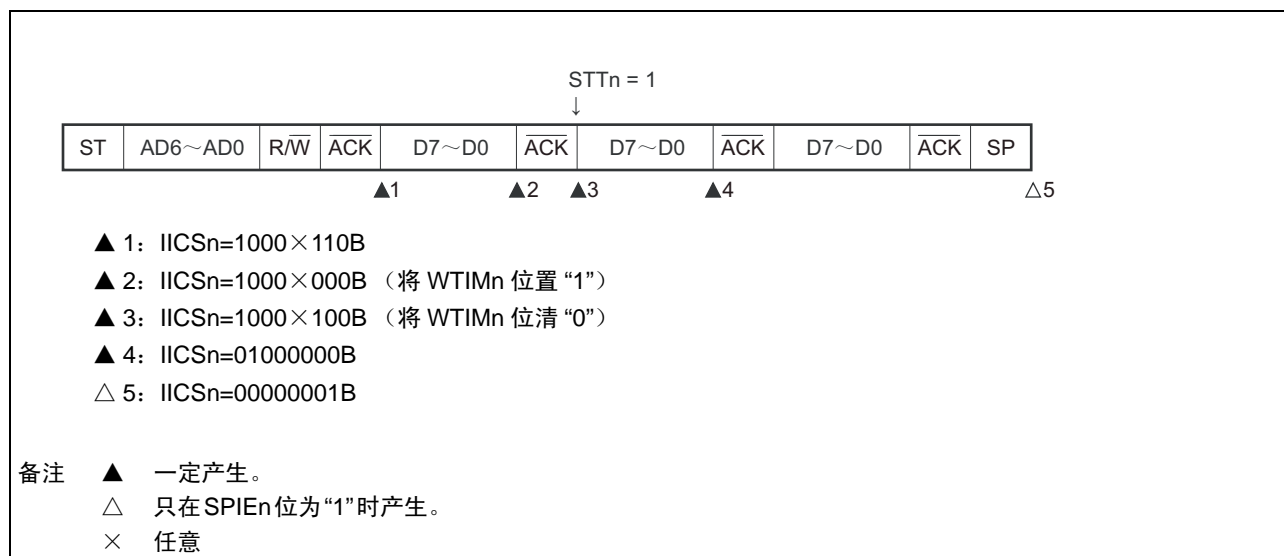
▲ 1: IICSn=10000110B  
△ 2: IICSn=01000001B

备注 ▲ 一定产生。  
△ 只在 SPIEn 位为“1”时产生。  
× 任意  
m=6~0

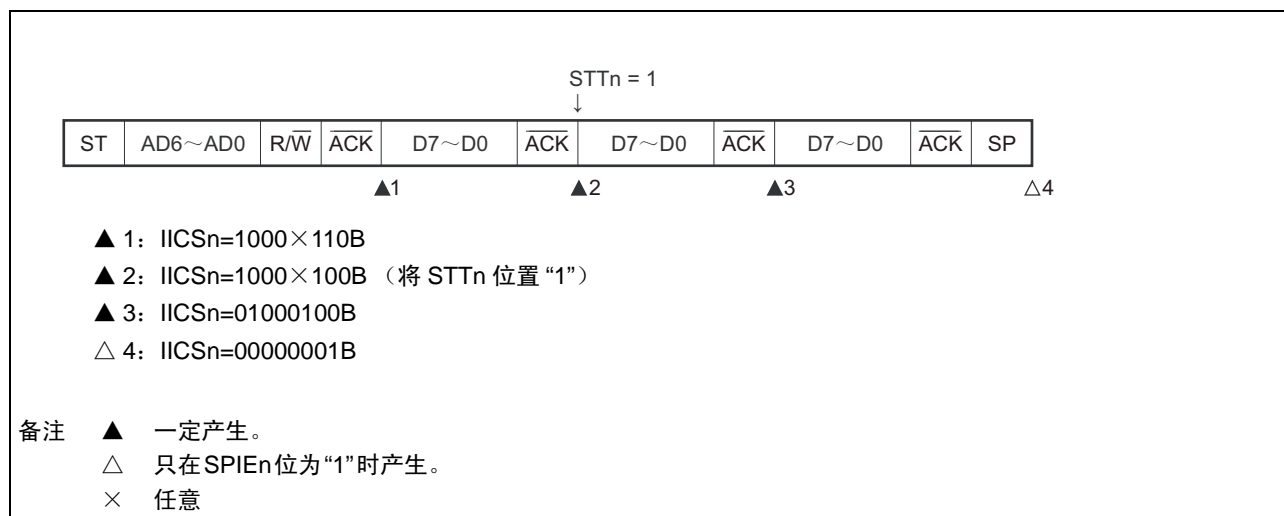
备注 n=0、1

(f) 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



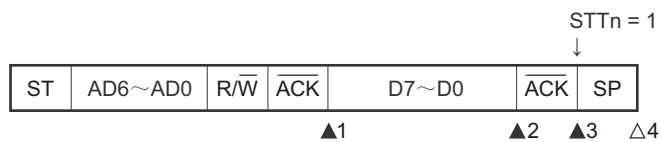
(ii) WTIMn=1的情况



备注   n=0、1

(g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

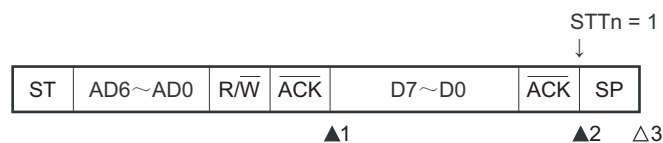
(i) WTIMn=0的情况



- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000×000B (将 WTIMn 位置“1”)
- ▲ 3: IICSn=1000××00B (将 STTn 位置“1”)
- △ 4: IICSn=01000001B

备注 ▲ 一定产生。  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

(ii) WTIMn=1的情况



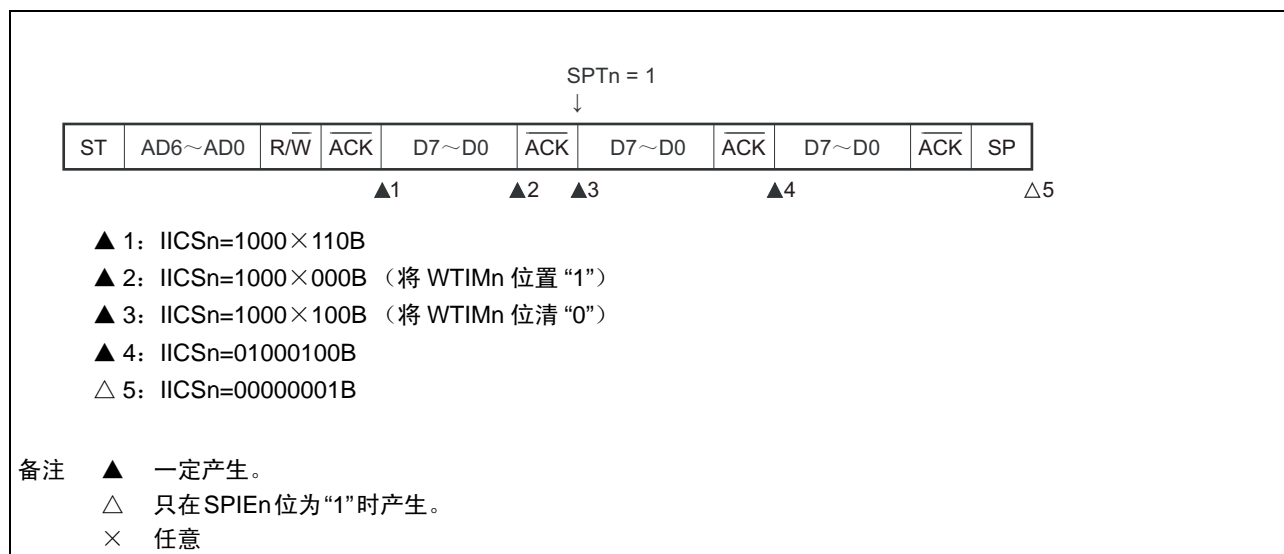
- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000××00B (将 STTn 位置“1”)
- △ 3: IICSn=01000001B

备注 ▲ 一定产生。  
 △ 只在 SPIEn 位为“1”时产生。  
 × 任意

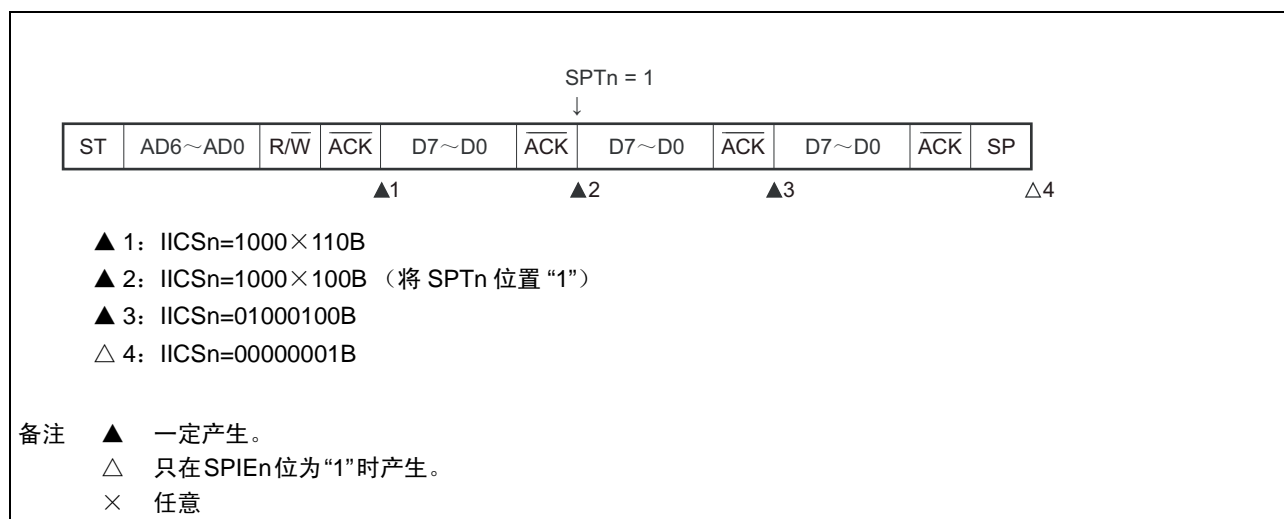
备注 n=0、1

(h) 在想要生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



(ii) WTIMn=1的情况



备注   n=0、1



## 12.6 时序图

在 I<sup>2</sup>C 总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。

主控设备在从属设备地址之后发送表示数据传送方向的 TRCn 位（IICA 状态寄存器 n（IICSn）的 bit3），开始与从属设备进行串行通信。

数据通信的时序图如图 12-32 和图 12-33 所示。

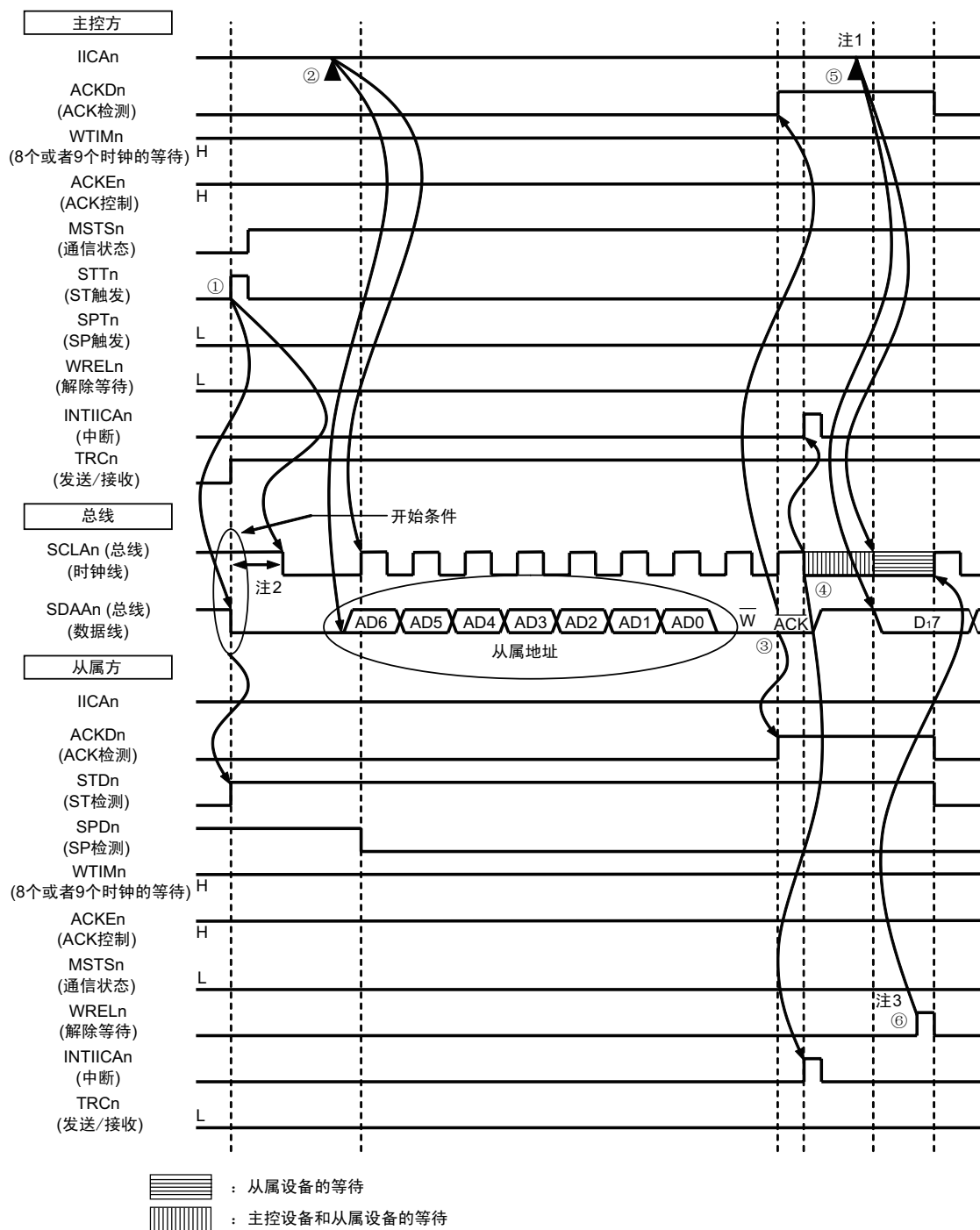
与串行时钟（SCLAn）的下降沿同步进行 IICA 移位寄存器 n（IICAn）的移位，并且将发送数据传送到 SO 锁存器，以 MSB 优先从 SDAAn 引脚输出数据。

在 SCLAn 的上升沿将 SDAAn 引脚输入的数据取到 IICAn。

备注 n=0、1

图 12-32 主控设备 → 从属设备的通信例子  
 (主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (1/4)

(1) 开始条件～地址～数据



- 注 1. 要在主控方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间, 在设定为标准模式时至少为 4.0μs, 在设定为快速模式时至少为 0.6μs。
3. 要在从属方的接收期间解除等待时, 必须将 IICAn 置 “FFH” 或者将 WRELn 位置位。

备注 n=0、1

图 12-32 的“(1) 开始条件～地址～数据”的①～⑥的说明如下：

- ① 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS<sub>n</sub>=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 n（IICAn）写地址+W（发送），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤ 主控方给 IICAn 寄存器写发送数据，解除主控方的等待。
- ⑥ 如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。  
但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

备注 1. 图 12-32 的①～⑮是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件～地址～数据”说明步骤①～⑥。

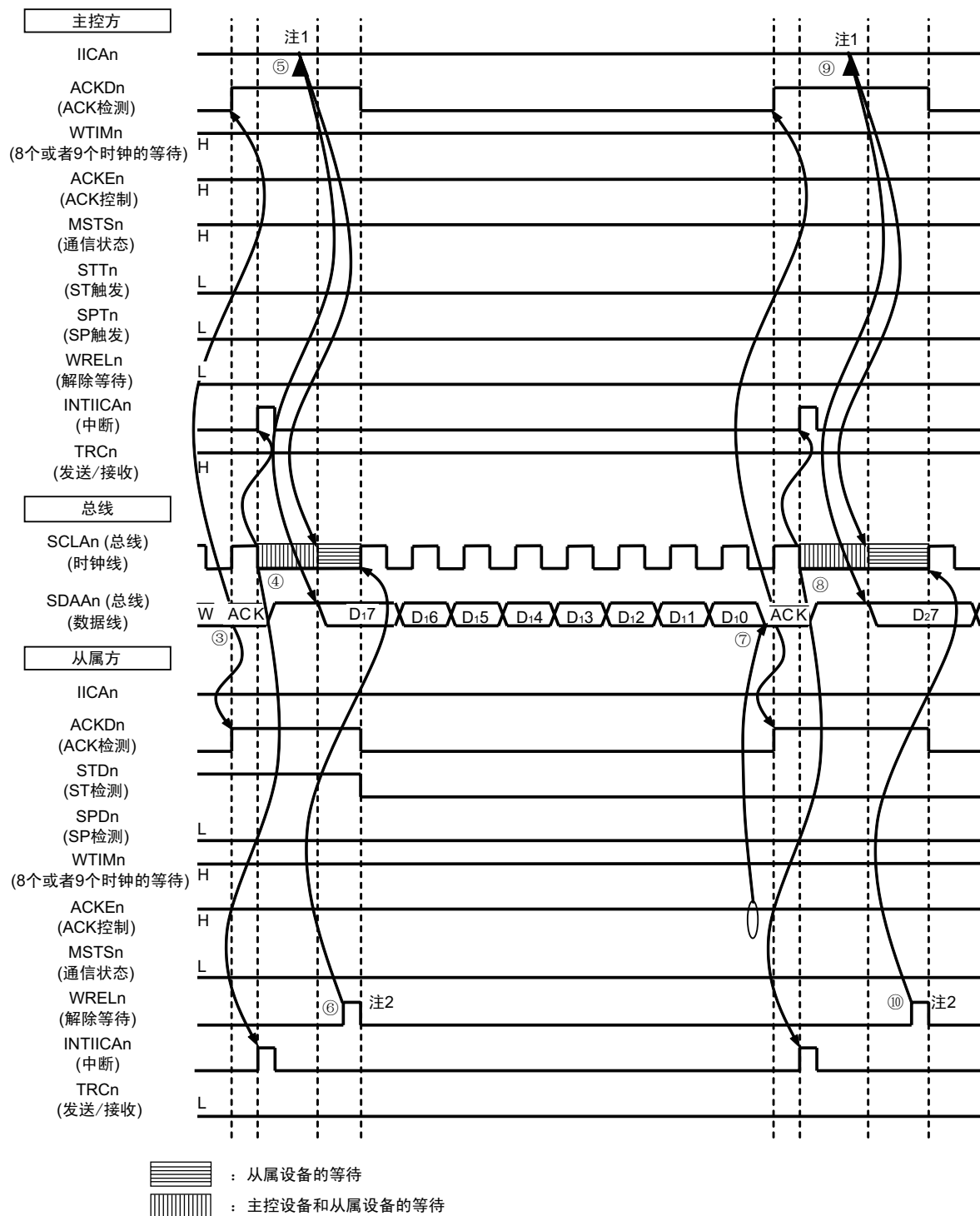
图 12-32 的“(2) 地址～数据～数据”说明步骤③～⑩。

图 12-32 的“(3) 数据～数据～停止条件”说明步骤⑦～⑮。

2. n=0、1

图 12-32 主控设备 → 从属设备的通信例子  
 (主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (2/4)

(2) 地址~数据~数据



注 1. 要在主控方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

2. 要在从属方的接收期间解除等待时, 必须将 IICAn 置 "FFH" 或者将 WRELn 位置位。

备注 n=0、1

图 12-32 的“(2) 地址~数据~数据”的③~⑩的说明如下：

- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同<sup>注</sup>，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）<sup>注</sup>。
- ⑤ 主控方给 IICA 移位寄存器 n（IICAn）写发送数据，解除主控方的等待。
- ⑥ 如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。
- ⑦ 在数据传送结束后，因为从属方的 ACKEn 位为“1”，所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且都产生中断（INTIICAn：传送结束中断）。
- ⑨ 主控方给 IICAn 寄存器写发送数据，解除主控方的等待。
- ⑩ 如果从属方读接收数据并且解除等待（WRELn=1），主控方就开始给从属方传送数据。

**注** 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。  
但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

**备注 1.** 图 12-32 的①~⑮是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件~地址~数据”说明步骤①~⑥。

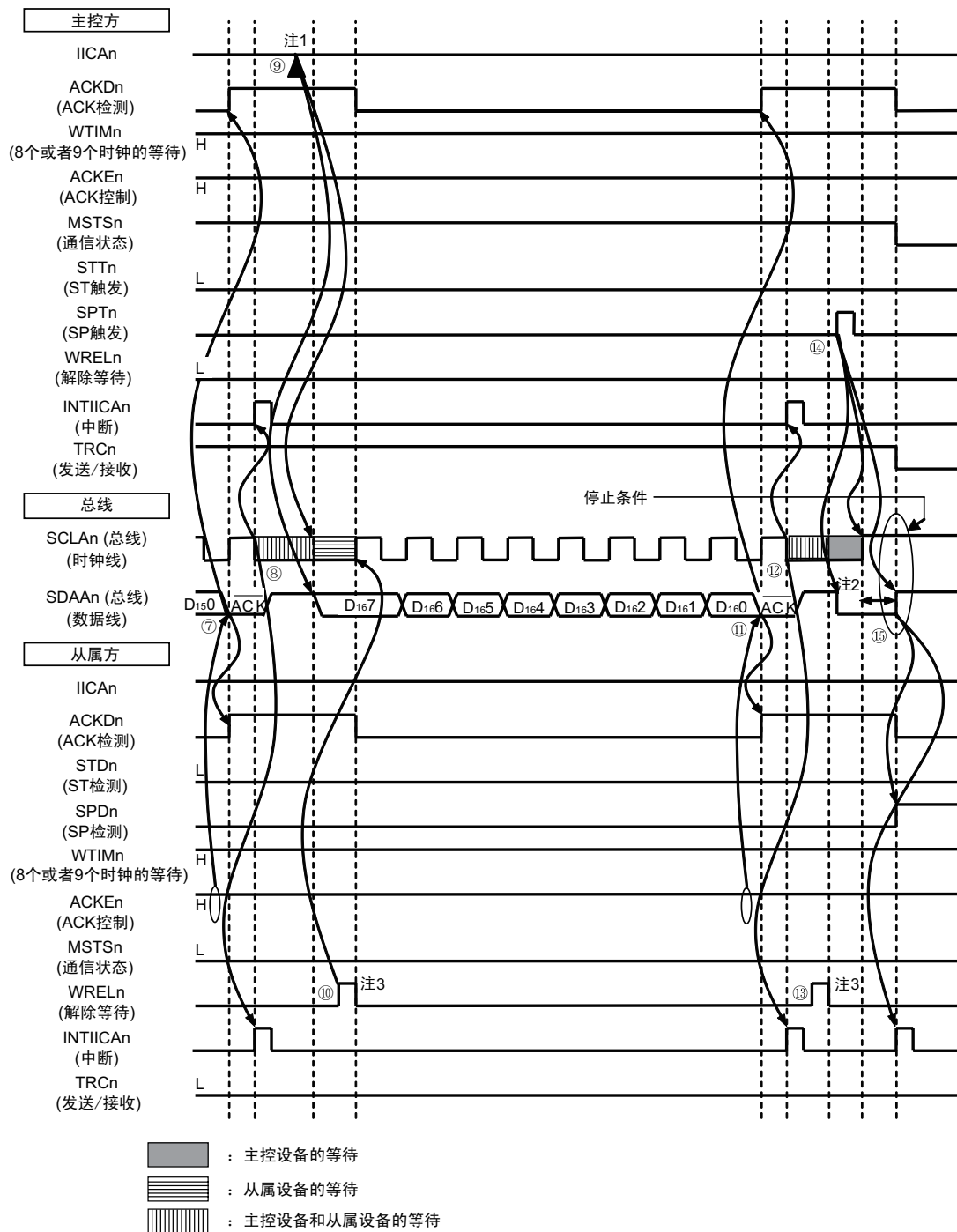
图 12-32 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 12-32 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

2. n=0、1

图 12-32 主控设备 → 从属设备的通信例子  
(主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (3/4)

(3) 数据～数据～停止条件



- 注 1. 要在主控方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。
2. 在发行停止条件后, 从 SCLAn 引脚信号上升到生成停止条件的的时间, 在设定为标准模式时至少为 4.0μs, 在设定为快速模式时至少为 0.6μs。
3. 要在从属方的接收期间解除等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0、1

图 12-32 的“(3) 数据～数据～停止条件”的⑦～⑮的说明如下：

- ⑦ 在数据传送结束后，因为从属方的 ACKEn 位为“1”，所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑨ 主控方给 IICA 移位寄存器 n (IICAn) 写发送数据，解除主控方的等待。
- ⑩ 如果从属方读接收数据并且解除等待 (WRELn=1)，主控方就开始给从属方传送数据。
- ⑪ 在数据传送结束后，从属方 (ACKEn=1) 通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑫ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑬ 从属方读接收数据，解除等待 (WRELn=1)。
- ⑭ 如果在主控方将停止条件触发置位 (SPTn=1)，就清除总线数据线 (SDAAn=0) 并且将总线时钟线置位 (SCLAn=1)，在经过停止条件的准备时间后将总线数据线置位 (SDAAn=1)，生成停止条件 (通过 SCLAn=1 使 SDAAn 从“0”变为“1”)。
- ⑮ 如果生成停止条件，从属方就检测到停止条件并且产生中断 (INTIICAn: 停止条件中断)。

备注 1. 图 12-32 的①～⑮是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件～地址～数据”说明步骤①～⑥。

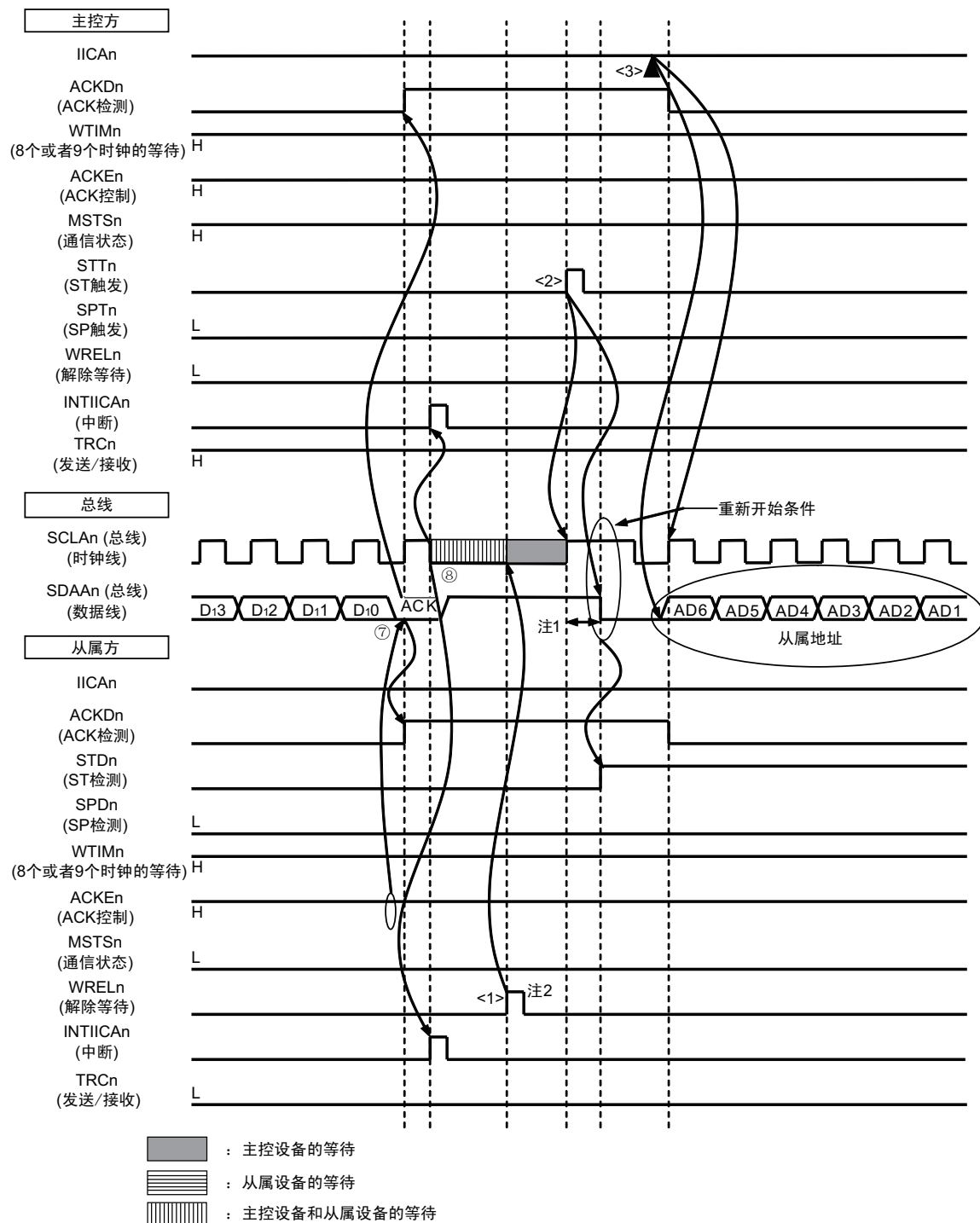
图 12-32 的“(2) 地址～数据～数据”说明步骤③～⑩。

图 12-32 的“(3) 数据～数据～停止条件”说明步骤⑦～⑮。

2. n=0、1

图 12-32 主控设备 → 从属设备的通信例子  
 (主控设备: 选择 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (4/4)

(4) 数据～重新开始条件～地址



- 注 1. 在发行重新开始条件后, 从 SCLAn 引脚信号上升到生成开始条件的的时间, 在设定为标准模式时至少为 4.7μs, 在设定为快速模式时至少为 0.6μs。
2. 要在从属方的接收期间解除等待时, 必须将 IICAn 置 "FFH" 或者将 WRELn 位置位。

备注 n=0、1



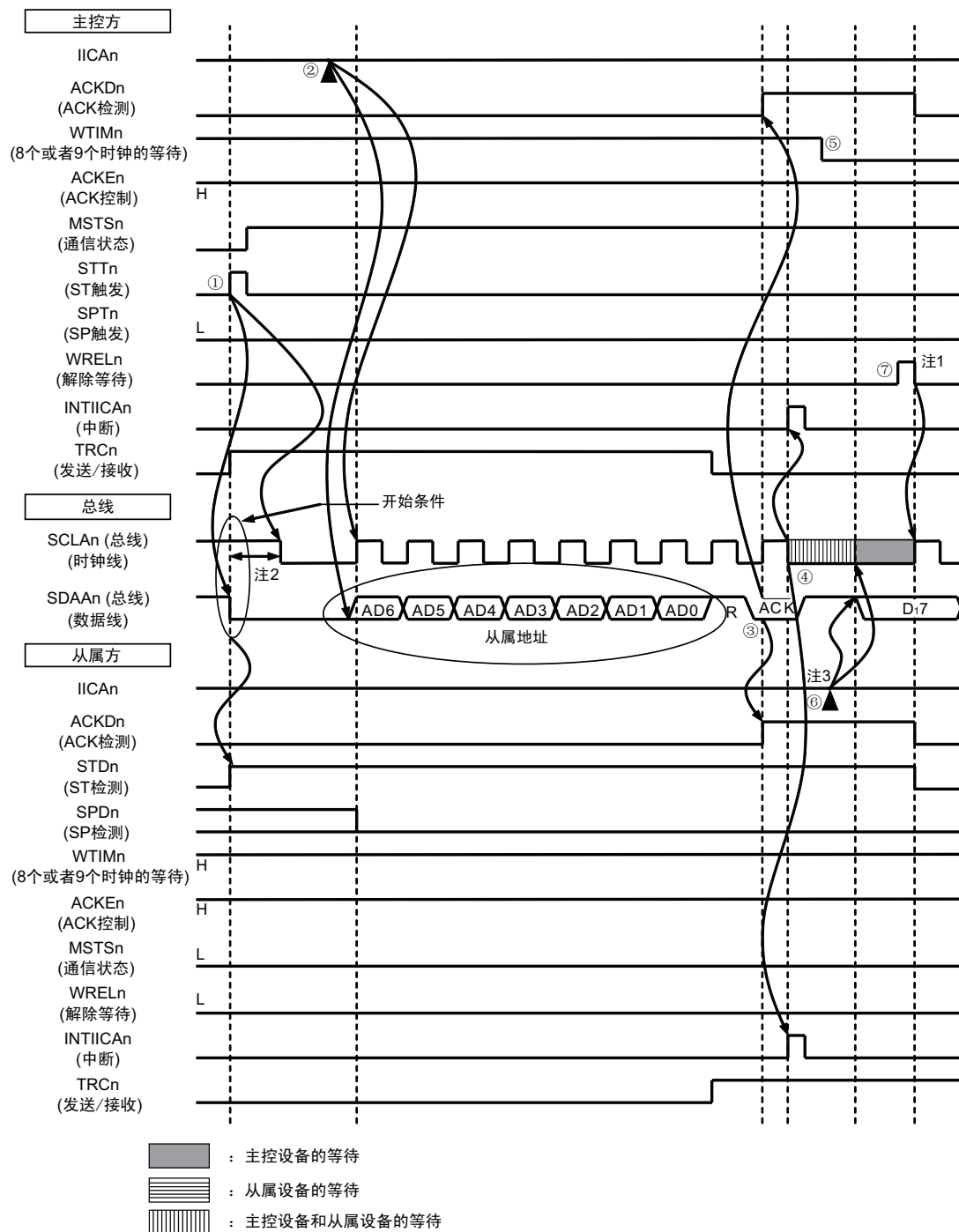
图 12-32 的“(4) 数据～重新开始条件～地址”的运行说明如下。在执行步骤⑦和⑧后执行 <1>～<3>，从而返回到步骤③的数据发送步骤。

- ⑦ 在数据传送结束后，因为从属方的  $ACKEn$  位为“1”，所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK ( $ACKDn=1$ )。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 ( $SCLAn=0$ )，并且都产生中断 ( $INTIICAn$ : 传送结束中断)。
  - <1> 从属方读接收数据，解除等待 ( $WRELn=1$ )。
  - <2> 如果在主控方再次将开始条件触发置位 ( $STTn=1$ )，总线时钟线就上升 ( $SCLAn=1$ )，而且在经过重新开始条件的准备时间后总线数据线下降 ( $SDAAn=0$ )，生成开始条件 (通过  $SCLAn=1$  使  $SDAAn$  从“1”变为“0”)。然后，如果检测到开始条件，就在经过保持时间后总线时钟线下降 ( $SCLAn=0$ )，结束通信准备。
  - <3> 如果主控方给 IICA 移位寄存器 n ( $IICAn$ ) 写地址 +R/W (发送)，就发送从属地址。

备注 n=0、1

图 12-33 从属设备 → 主控设备的通信例子  
(主控设备: 选择 8 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (1/3)

(1) 开始条件～地址～数据



- 注 1. 要在主控方的接收期间解除等待时, 必须将 IICAn 置“FFH”或者将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间, 在设定为标准模式时至少为 4.0μs, 在设定为快速模式时至少为 0.6μs。
3. 要在从属方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0、1

图 12-33 的“(1) 开始条件～地址～数据”的①～⑦的说明如下：

- ① 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS<sub>n</sub>=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 n（IICAn）写地址+R（接收），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤ 主控方将等待时序改为第 8 个时钟（WTIMn=0）。
- ⑥ 从属方给 IICAn 寄存器写发送数据，解除从属方的等待。
- ⑦ 主控方解除等待（WRELn=1），开始来自从属设备的数据传送。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。

但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

备注 1. 图 12-33 的①～⑱是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件～地址～数据”说明步骤①～⑦。

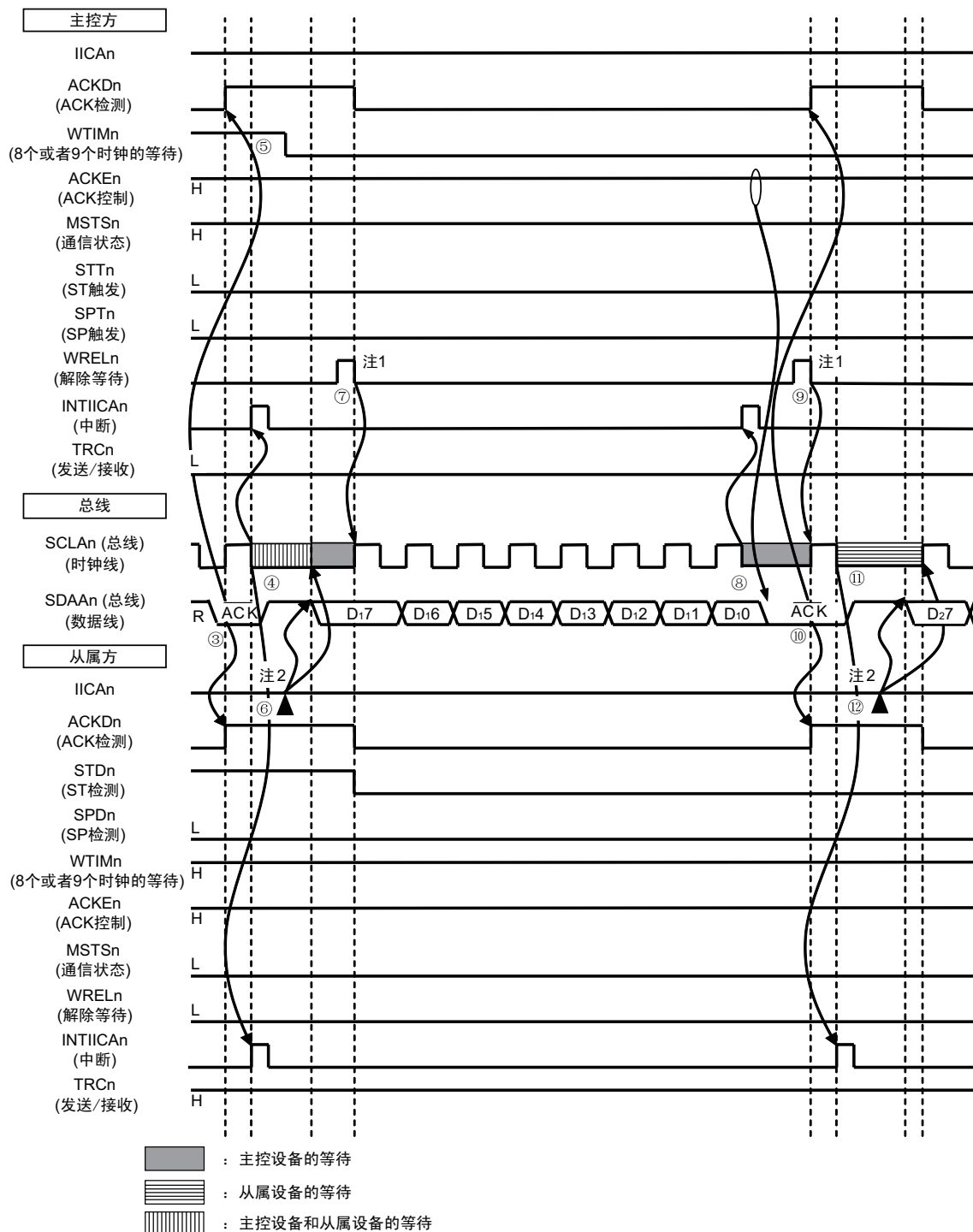
图 12-33 的“(2) 地址～数据～数据”说明步骤③～⑫。

图 12-33 的“(3) 数据～数据～停止条件”说明步骤⑧～⑱。

2. n=0、1

图 12-33 从属设备 → 主控设备的通信例子  
 (主控设备: 选择 8 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (2/3)

(2) 地址~数据~数据



- 注 1. 要在主控方的接收期间解除等待时, 必须将 IICAn 置 "FFH" 或者将 WRELn 位置位。  
 2. 要在从属方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0、1

图 12-33 的“(2) 地址~数据~数据”的③~⑫的说明如下：

- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤ 主控方将等待时序改为第 8 个时钟（WTIMn=0）。
- ⑥ 从属方给 IICA 移位寄存器 n（IICAn）写发送数据，解除从属方的等待。
- ⑦ 主控方解除等待（WRELn=1），开始来自从属设备的数据传送。
- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）。因为主控方的 ACKEn 位为“1”，所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据，解除等待（WRELn=1）。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态（SCLAn=0），并且产生中断（INTIICAn：传送结束中断）。
- ⑫ 如果从属方给 IICAn 寄存器写发送数据，就解除从属方的等待，开始从属设备到主控设备的数据传送。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。  
但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

备注 1. 图 12-33 的①~⑭是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

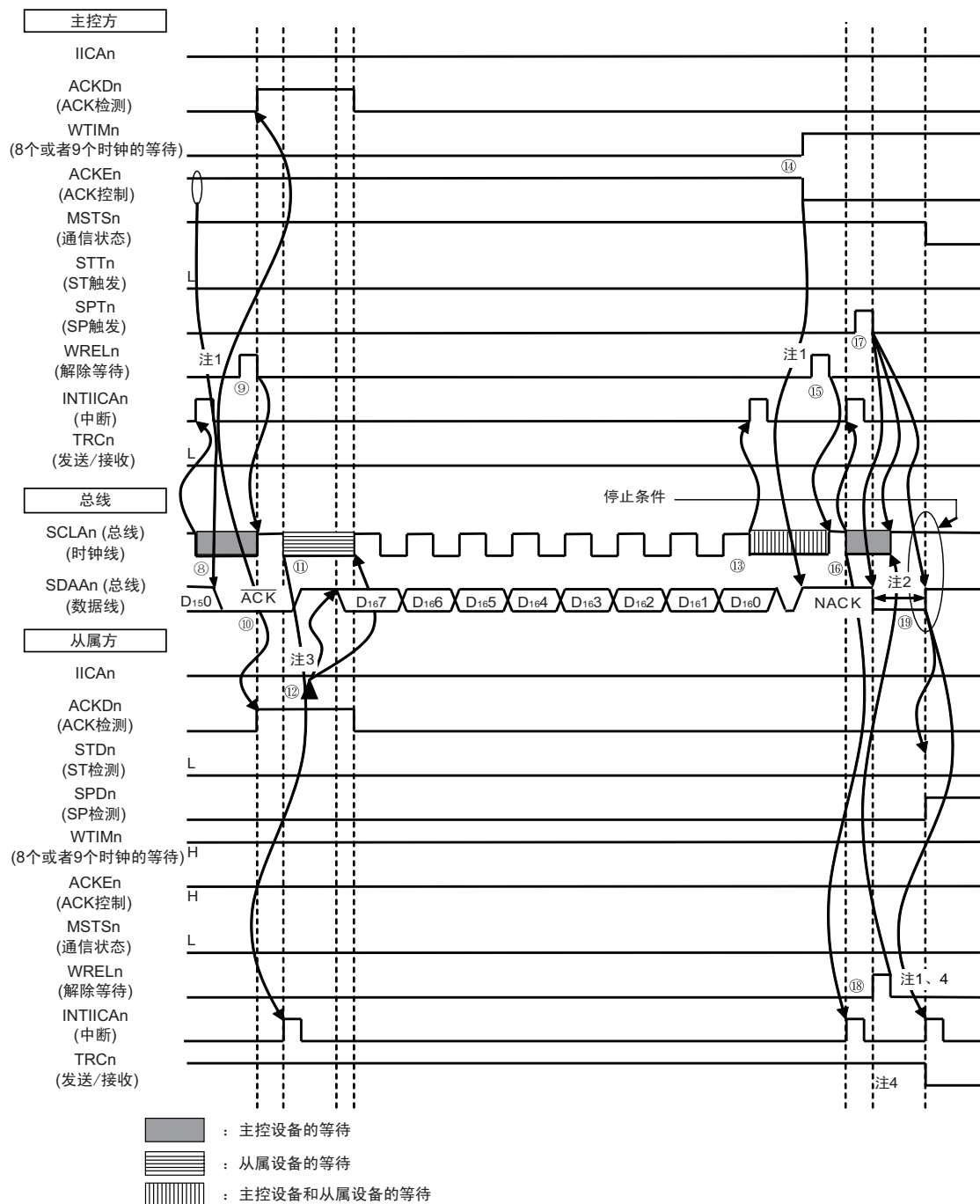
图 12-33 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 12-33 的“(3) 数据~数据~停止条件”说明步骤⑧~⑭。

2. n=0、1

图 12-33 从属设备 → 主控设备的通信例子  
 (主控设备: 选择 8 个 → 9 个时钟的等待, 从属设备: 选择 9 个时钟的等待) (3/3)

(3) 数据～数据～停止条件



- 注 1. 要解除等待时, 必须将 IICAn 置 "FFH" 或者将 WRELn 位置位。
2. 在发行停止条件后, 从 SCLAn 引脚信号上升到生成停止条件的的时间, 在设定为标准模式时至少为 4.0μs, 在设定为快速模式时至少为 0.6μs。
3. 要在从属方的发送期间解除等待时, 必须给 IICAn 写数据而不是将 WRELn 位置位。
4. 在从属方的发送期间, 如果通过 WRELn 位的置位来解除等待, 就清除 TRCn 位。

备注 n=0、1

图 12-33 的“(3) 数据～数据～停止条件”的⑧～⑲的说明如下：

- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态 ( $SCLAn=0$ )，并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的  $ACKEn$  位为“0”，所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据，解除等待 ( $WRELn=1$ )。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK ( $ACKDn=1$ )。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态 ( $SCLAn=0$ )，并且产生中断 (INTIICAn: 传送结束中断)。
- ⑫ 如果从属方给 IICA 移位寄存器 n (IICAn) 写发送数据，就解除从属方的等待，开始从属设备到主控设备的数据传送。
- ⑬ 主控方在第 8 个时钟的下降沿产生中断 (INTIICAn: 传送结束中断)，并且进入等待状态 ( $SCLAn=0$ )。因为进行 ACK 控制 ( $ACKEn=1$ )，所以此阶段的总线数据线变为低电平 ( $SDAAn=0$ )。
- ⑭ 主控方设定为 NACK 应答 ( $ACKEn=0$ )，并且将等待时序改为第 9 个时钟 ( $WTIMn=1$ )。
- ⑮ 如果主控方解除等待 ( $WRELn=1$ )，从属方就在第 9 个时钟的上升沿检测到 NACK ( $ACKDn=0$ )。
- ⑯ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 ( $SCLAn=0$ )，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑰ 如果主控方发行停止条件 ( $SPTn=1$ )，就清除总线数据线 ( $SDAAn=0$ )，并且解除主控方的等待。此后，主控设备处于待机状态，直到将总线时钟线置位 ( $SCLAn=1$ ) 为止。
- ⑱ 从属方在确认 NACK 后停止发送，为了结束通信，解除等待 ( $WRELn=1$ )。如果解除从属方的等待，就将总线时钟线置位 ( $SCLAn=1$ )。
- ⑲ 如果主控方确认到总线时钟线被置位 ( $SCLAn=1$ )，就在经过停止条件准备时间后将总线数据线置位 ( $SDAAn=1$ )，然后发行停止条件 (通过  $SCLAn=1$  使  $SDAAn$  从“0”变为“1”)。如果生成停止条件，从属方就检测到停止条件，并且产生中断 (INTIICAn: 停止条件中断)。

备注 1. 图 12-33 的①～⑲是通过 I<sup>2</sup>C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件～地址～数据”说明步骤①～⑦。

图 12-33 的“(2) 地址～数据～数据”说明步骤③～⑫。

图 12-33 的“(3) 数据～数据～停止条件”说明步骤⑧～⑲。

2.  $n=0, 1$

## 第 13 章 DMA 控制器

R7F0C010 内置一个 DMA（Direct Memory Access）控制器。

能在支持 DMA 外围硬件的 SFR 和内部 RAM 之间不经过 CPU 而自动传送数据。

因为能在进行通常的 CPU 内部运算和数据传送的同时进行 SFR 和内部 RAM 之间的传送，所以能进行大容量数据的处理，并且还能实现通过通信、定时器和 A/D 进行的实时控制。

### 13.1 DMA 控制器的功能

- DMA 通道个数：2 个通道
- 传送单位：8 位或者 16 位
- 最大传送单位：1024 次
- 传送类型：2 个周期传送（通过 2 个时钟进行一次传送并且在传送期间 CPU 停止运行）
- 传送模式：单次传送模式
- 传送请求：选择以下外围硬件中断。
  - A/D 转换器
  - 串行接口（CSI00、UART0）
  - 定时器（通道 0、1、2、3）
- 传送对象：SFR 和内部 RAM 之间的传送

使用 DMA 功能的例子如下所示：

- 串行接口的连续传送
- 模拟数据的批量传送
- 每隔一段时间的 A/D 转换结果的取得
- 每隔一段时间的端口值的取得



## 13.2 DMA 控制器的结构

DMA 控制器由以下硬件构成。

表 13-1 DMA 控制器的结构

项目	结构
地址寄存器	<ul style="list-style-type: none"> <li>DMA SFR 地址寄存器 0、1 (DSA0、DSA1)</li> <li>DMA RAM 地址寄存器 0、1 (DRA0、DRA1)</li> </ul>
计数寄存器	<ul style="list-style-type: none"> <li>DMA 字节计数寄存器 0、1 (DBC0、DBC1)</li> </ul>
控制寄存器	<ul style="list-style-type: none"> <li>DMA 模式控制寄存器 0、1 (DMC0、DMC1)</li> <li>DMA 运行控制寄存器 0、1 (DRC0、DRC1)</li> </ul>

### 13.2.1 DMA SFR 地址寄存器 n (DSAn)

这是设定 SFR 地址的 8 位寄存器，SFR 地址是 DMA 通道 n 的传送源或者传送目标。

必须设定 SFR 地址 FFF00H ~ FFFFFH 的低 8 位。

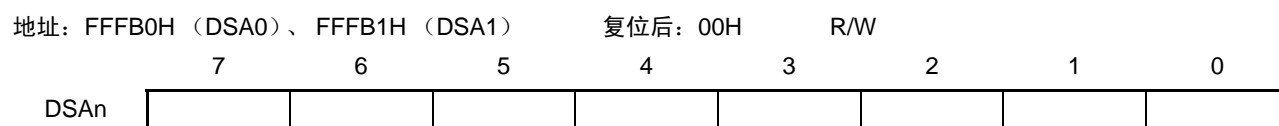
此寄存器为固定值而不会自动递增。

在 16 位传送模式中，忽略最低位而作为偶数地址处理。

能以 8 位为单位读写 DSAn 寄存器，但是不能在 DMA 传送过程中写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 13-1 DMA SFR 地址寄存器 n (DSAn) 的格式



备注 n: DMA 通道号 (n=0、1)

13.2.2 DMA RAM 地址寄存器 n（DRAn）

这是设定 RAM 地址的 16 位寄存器，RAM 地址是 DMA 通道 n 的传送源或者传送目标。  
能设定通用寄存器以外的内部 RAM 区（参照表 13-2）的地址。  
必须设定 RAM 地址的低 16 位。

如果开始 DMA 传送，此寄存器就自动递增。在 8 位传送模式中增 1；在 16 位传送模式中增 2。如果从此 DRAn 寄存器的设定地址开始传送到最后地址，DMA 传送就停止运行。此时，在 8 位传送模式中，DRAn 寄存器为最后地址 +1；在 16 位传送模式中，DRAn 寄存器为最后地址 +2。  
在 16 位传送模式中，忽略最低位而作为偶数地址处理。  
能以 8 位或者 16 位为单位读写 DRAn 寄存器，但是不能在 DMA 传送过程中写此寄存器。  
在产生复位信号后，此寄存器的值变为“0000H”。

图 13-2 DMA RAM 地址寄存器 n（DRAn）的格式

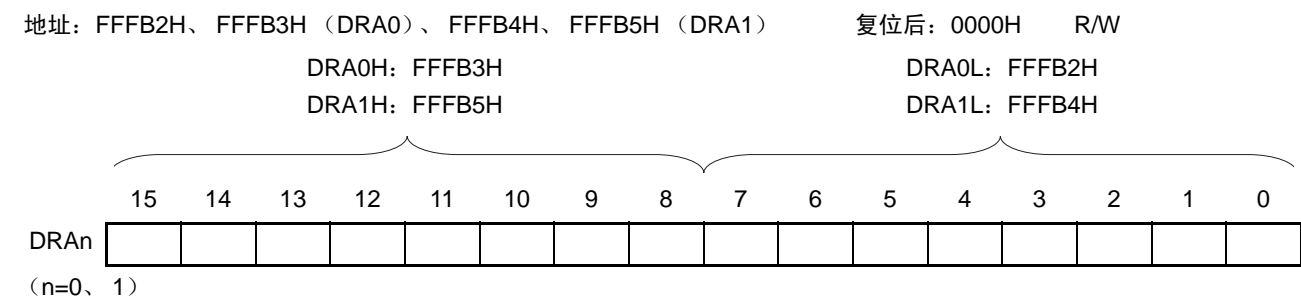


表 13-2 通用寄存器以外的内部 RAM 区

产品	通用寄存器以外的内部 RAM 区
R7F0C010	FF900H ~ FFEDFH

备注    n: DMA 通道号（n=0、1）

13.2.3 DMA 字节计数寄存器 n（DBCn）

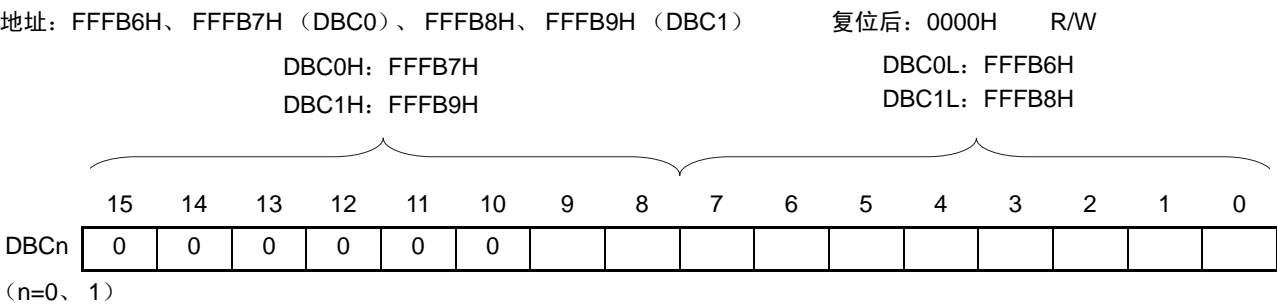
这是设定 DMA 通道 n 传送次数的 10 位寄存器。必须在 DMA 传送前给此 DBCn 寄存器设定连续的传送次数（最多 1024 次）。

每进行一次 DMA 传送，此寄存器就自动递减。在 DMA 传送过程中，能通过读此 DBCn 寄存器获得剩余的连续传送次数。

能以 8 位或者 16 位为单位读写 DBCn 寄存器，但是不能在 DMA 传送过程中写此寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 13-3 DMA 字节计数寄存器 n（DBCn）的格式



13.3 控制 DMA 控制器的寄存器

控制 DMA 控制器的寄存器如下所示：

- DMA 模式控制寄存器 n（DMCn）
- DMA 运行控制寄存器 n（DRCn）

备注 n：DMA 通道号（n=0、1）

13.3.1 DMA 模式控制寄存器 n（DMCn）

DMCn 寄存器设定 DMA 通道 n 的传送模式，选择传送方向、数据长度、保留设定和启动源。bit7（STGn）为启动 DMA 的软件触发。

禁止在 DMA 运行过程中（DSTn=1）改写 DMCn 寄存器的 bit6、5、3～0。

通过 1 位或者 8 位存储器操作指令设定 DMCn 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 13-4 DMA 模式控制寄存器 n（DMCn）的格式 (1/2)

地址：FFFBAH（DMC0）、FFFBH（DMC1）

复位后：00H

R/W

符号	7	6	5	4	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	0	IFCn2	IFCn1	IFCn0

STGn 注 1	DMA 传送开始的软件触发
0	没有软件触发。
1	在允许 DMA 运行（DENn=1）时，开始 DMA 传送。

在允许 DMA 运行（DENn=1）时，通过给 STGn 位写“1”进行一次 DMA 传送。  
此位的读取值总是“0”。

DRSn	DMA 传送方向的选择
0	SFR→ 内部 RAM
1	内部 RAM→SFR

DSn	DMA 传送数据长度的指定
0	8 位
1	16 位

DWAITn 注 2	DMA 传送的保留
0	通过 DMA 启动请求进行 DMA 传送（不保留）。
1	即使接收到 DMA 启动请求也保留 DMA 传送。

能通过将 DWAITn 位的值从“1”变为“0”来开始被保留的 DMA 传送。  
在将 DWAITn 位的值从“0”变为“1”后，实际上需要经过 2 个时钟，然后保留 DMA 传送。

- 注 1. 与 IFCn2～IFCn0 位的值无关，能使用软件触发（STGn）。  
2. 要在使用 2 个或者更多通道的 DMA 过程中保留 DMA 传送时，必须保留全部通道的 DMA 传送（DWAIT0=DWAIT1=1）。

备注 n：DMA 通道号（n=0、1）

图 13-4 DMA 模式控制寄存器 n（DMCn）的格式 (2/2)

地址：FFFB AH（DMC0）、FFFB BH（DMC1）		复位后：00H		R/W			
符号	7	6	5	4	3	2	1 0
DMCn	STGn	DRSn	DSn	DWAITn	0	IFCn2	IFCn1 IFCn0

IFCn2	IFCn1	IFCn0	DMA 启动源的选择注	
			触发信号	触发内容
0	0	0	—	禁止通过中断触发 DMA 传送。 （只能允许软件触发）
0	0	1	INTAD	A/D 转换结束中断
0	1	0	INTTM00	定时器通道00的计数结束或者捕捉中断
0	1	1	INTTM01	定时器通道01的计数结束或者捕捉中断
1	0	0	INTTM02	定时器通道02的计数结束或者捕捉中断
1	0	1	INTTM03	定时器通道03的计数结束或者捕捉中断
1	1	0	INTST0/INTCSI00	UART0 发送的传送结束或者缓冲器空中断/CSI00的传送结束或者缓冲器空中断
1	1	1	INTSR0	UART0 接收的传送结束中断
上述以外			禁止设定。	

注 与 IFCn2 ～ IFCn0 位的值无关，能使用软件触发（STGn）。

备注 n: DMA 通道号（n=0、1）

13.3.2 DMA 运行控制寄存器 n（DRCn）

DRCn 寄存器设定允许或者禁止 DMA 通道 n 的传送。  
禁止在 DMA 运行过程中（DSTn=1）改写 DRCn 寄存器的 bit7（DENn）。  
通过 1 位或者 8 位存储器操作指令设定 DRCn 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 13-5 DMA 运行控制寄存器 n（DRCn）的格式

地址：FFFBCH（DRC0）、FFBDH（DRC1）

复位后：00H

R/W

符号	<div>7</div>	6	5	4	3	2	1	<div>0</div>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA 运行允许标志
0	禁止 DMA 通道 n 的运行（停止 DMA 的运行时钟）。
1	允许 DMA 通道 n 的运行。

在设定为允许 DMA 运行（DENn=1）后，通过将 DSTn 标志置“1”进入等待 DMA 触发的状态。

DSTn	DMA 传送模式标志
0	DMA 通道 n 的 DMA 传送结束。
1	DMA 通道 n 的 DMA 传送没有结束（正在传送）。

在设定为允许 DMA 运行（DENn=1）后，通过将 DSTn 标志置“1”进入等待 DMA 触发的状态。

如果输入软件触发（STGn）或者输入由 IFCn2 ~ IFCn0 位设定的启动源触发，就开始 DMA 传送。

此后，如果 DMA 传送结束，就自动将此标志清“0”。

要在 DMA 传送过程中强制结束传送时，给此标志写“0”。

注意 如果 DMA 传送结束，就自动将 DSTn 标志清“0”。  
因为只有在 DSTn 标志为“0”时才允许写 DENn 标志，所以当不等待 DMA 中断（INTDMA<sub>n</sub>）的发生而结束 DMA 传送时，必须在将 DSTn 标志置“0”后将 DENn 标志置“0”（有关详细内容，请参照“13.5.5 通过软件强制结束”）。

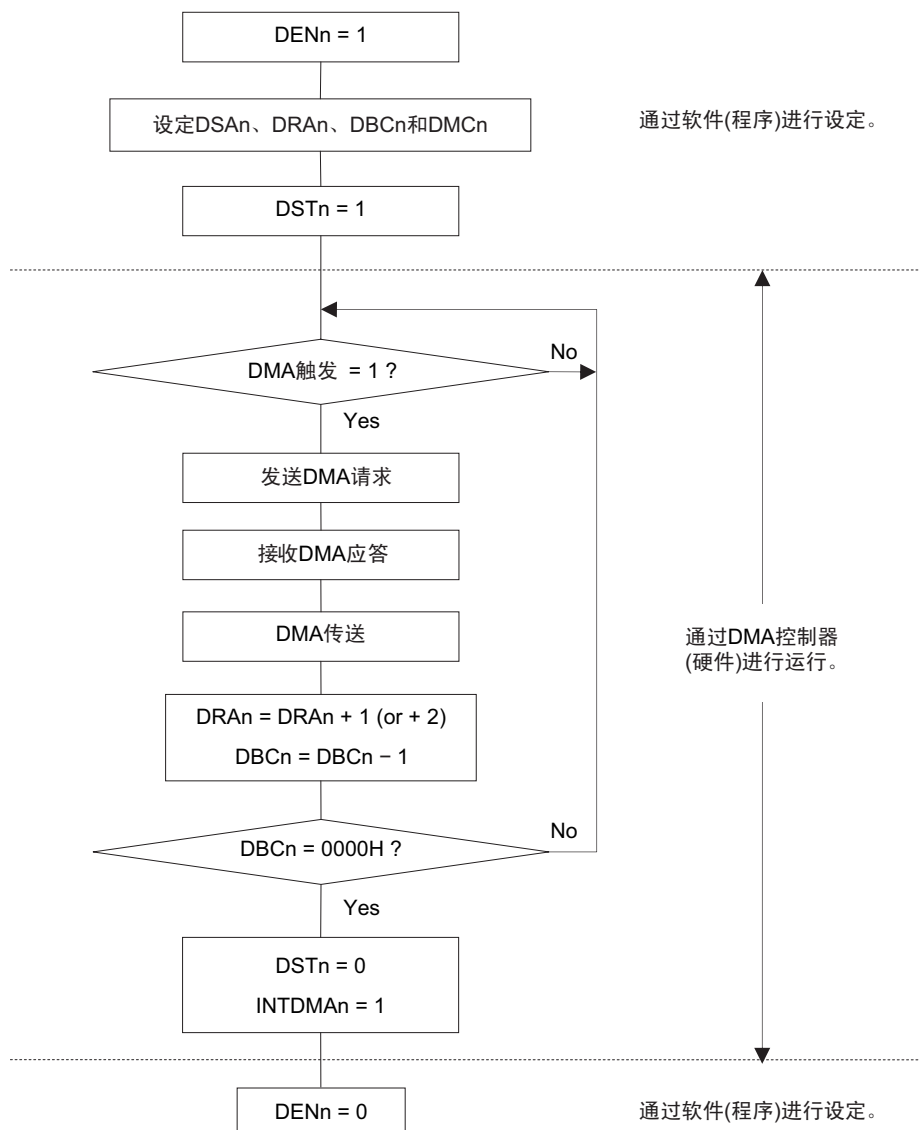
备注 n：DMA 通道号（n=0、1）

## 13.4 DMA 控制器的运行

### 13.4.1 运行步骤

- ① 通过将DEN<sub>n</sub>位置“1”，使DMA控制器变为允许运行状态。必须在将DEN<sub>n</sub>位置“1”后写其他寄存器。当通过8位操作指令进行写操作时，写“80H”。
- ② 给DMA SFR 地址寄存器n（DSAn）、DMA RAM 地址寄存器n（DRAn）、DMA 字节计数寄存器n（DBCn）和DMA 模式控制寄存器n（DMCn）设定DMA 传送的SFR 地址、RAM 地址、传送次数和传送模式。
- ③ 通过将DST<sub>n</sub>位置“1”，使DMA 变为等待触发的状态。当通过8位操作指令进行写操作时，写“81H”。
- ④ 如果输入软件触发（STG<sub>n</sub>）或者输入由IFC<sub>n2</sub>～IFC<sub>n0</sub>位设定的启动源触发，就开始DMA 传送。
- ⑤ 如果DBCn 寄存器设定的传送次数变为“0”，就完成传送并且产生中断请求（INTDMA<sub>n</sub>）而自动结束传送。
- ⑥ 此后，如果不使用DMA 控制器，就必须将DEN<sub>n</sub>位置“0”，进入运行停止状态。

图 13-6 运行步骤



备注 n: DMA 通道号 (n=0、1)

### 13.4.2 传送模式

能通过设定 DMA 模式控制寄存器 n (DMCn) 的 bit6 和 bit5 (DRSn、DSn)，给 DMA 传送选择以下 4 种传送模式。

DRSn	DSn	DMA 传送模式
0	0	从 1 字节数据 (固定地址) 的 SFR 到 RAM (地址递增 +1) 的传送
0	1	从 2 字节数据 (固定地址) 的 SFR 到 RAM (地址递增 +2) 的传送
1	0	从 1 字节数据 (地址递增 +1) 的 RAM 到 SFR (固定地址) 的传送
1	1	从 2 字节数据 (地址递增 +2) 的 RAM 到 SFR (固定地址) 的传送

通过这些传送模式，能利用串行接口最多连续传送 1024 字节的数据，并且能连续传送 A/D 转换结果的数据以及利用定时器每隔一段时间扫描端口的数据。

### 13.4.3 DMA 传送的结束

如果 DBCn 变为“00H”而完成 DMA 传送，就自动将 DSTn 位清“0”，然后产生中断请求 (INTDMAn) 而结束传送。

如果为了强制结束 DMA 传送而将 DSTn 位清“0”，DMA 字节计数寄存器 n (DBCn) 和 DMA RAM 地址寄存器 n (DRAn) 就保持停止时的值。

当强制结束传送时，不产生中断请求 (INTDMAn)。

备注 n: DMA 通道号 (n=0、1)



## 13.5 DMA 控制器的设定例子

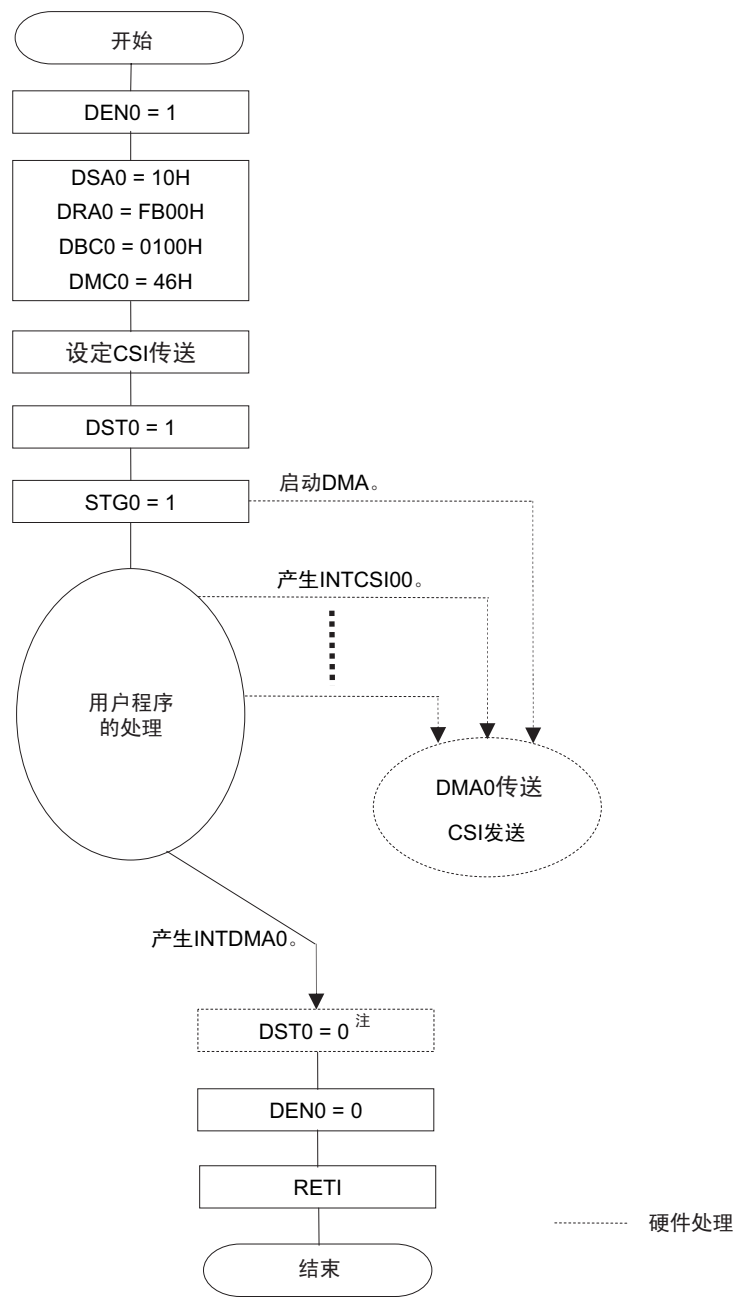
### 13.5.1 CSI 连续发送

CSI 连续发送的设定例子的流程图如下所示：

- CSI00 的连续发送（256 字节）
- DMA 传送使用 DMA 的通道 0。
- DMA 启动源：INTCSI00（软件触发（STG0）只限于最初的启动源）
- 通过 IFC02 ~ IFC00=0110B 指定 CSI00 的中断。
- 将 RAM 的 FFB00H ~ FFBFFH（256 字节）传送到 CSI 的数据寄存器（SIO00）的 FFF10H。

备注 IFC02 ~ IFC00：DMA 模式控制寄存器 0（DMC0）的 bit2 ~ 0

图 13-7 CSI 连续发送的设定例子



注 如果 DMA 传送结束，就自动将 DST0 标志清“0”。  
因为只有在 DST0 标志为“0”时才允许写 DEN0 标志，所以当不等待 DMA0 中断（INTDMA0）的发生而结束 DMA 传  
送时，必须在将 DST0 标志置“0”后将 DEN0 标志置“0”（有关详细内容，请参照“13.5.5 通过软件强制结束”）。

在连续发送时，CSI 中断不能启动第一次触发。在此例子中，通过软件触发进行启动。  
自动进行第二次及以后的 CSI 发送。  
在给数据寄存器写完最后的发送数据时，产生 DMA 中断（INTDMA0）。

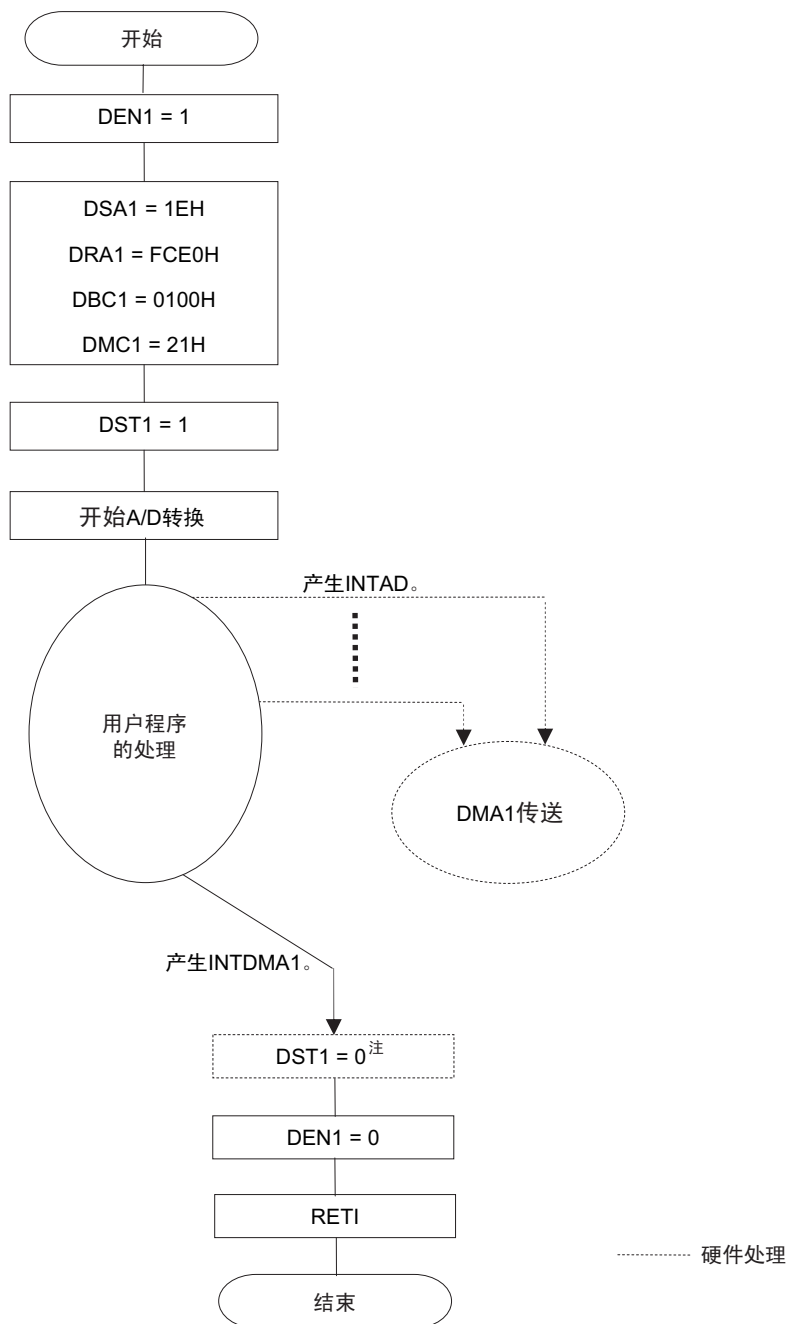
### 13.5.2 A/D 转换结果的连续读取

A/D 转换结果连续读取的设定例子的流程图如下所示：

- A/D 转换结果的连续读取
- DMA 传送使用 DMA 的通道 1。
- DMA 启动源：INTAD
- 通过 IFC12 ~ IFC10=0001B 指定 A/D 的中断。
- 将 12 位 A/D 转换结果寄存器（ADCR）的 FFF1EH 和 FFF1FH（2 字节）传送到 RAM 的 FFCE0H ~ FFEDFH（512 字节）。

备注 IFC12 ~ IFC10：DMA 模式控制寄存器 1（DMC1）的 bit2 ~ 0

图 13-8 A/D 转换结果连续读取的设定例子

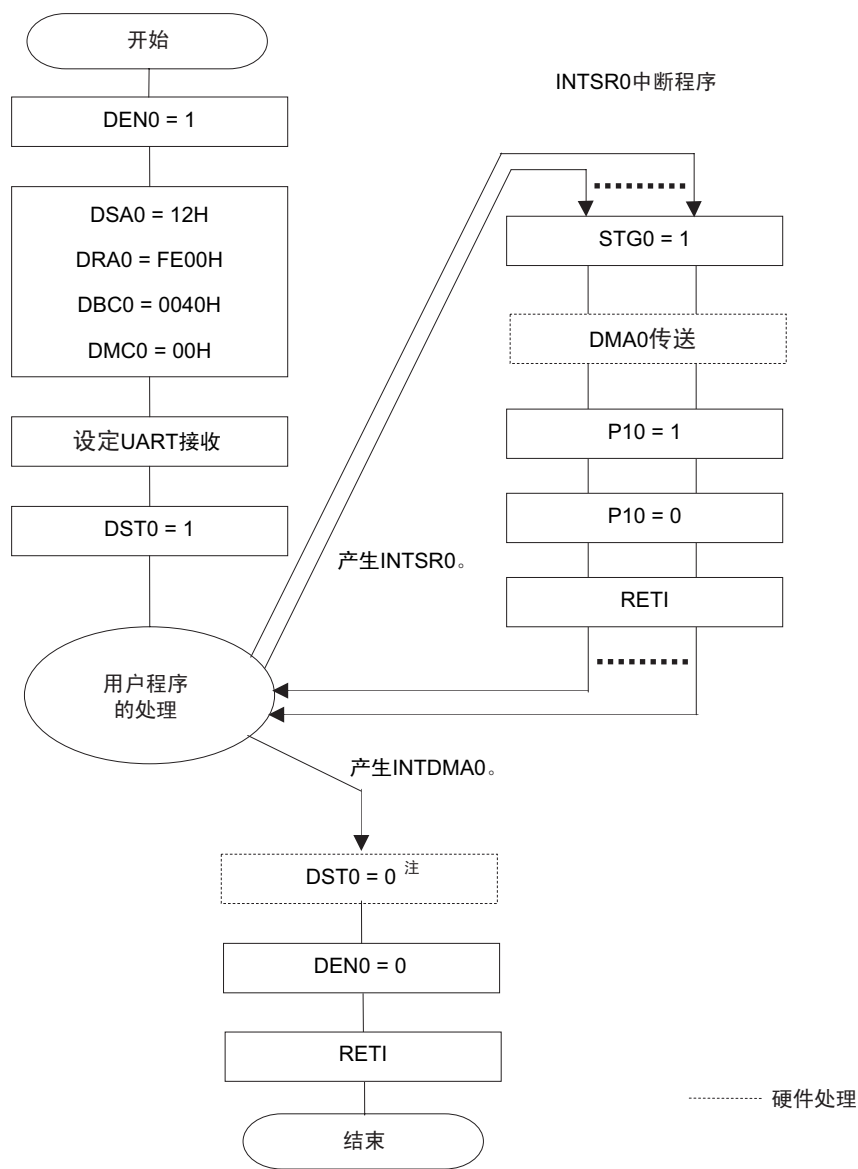


注 如果 DMA 传送结束，就自动将 DST1 标志清“0”。  
 因为只有在 DST1 标志为“0”时才允许写 DEN1 标志，所以当不等待 DMA1 中断（INTDMA1）的发生而结束 DMA 传送时，必须在将 DST1 标志置“0”后将 DEN1 标志置“0”（有关详细内容，请参照“13.5.5 通过软件强制结束”）。

13.5.3 UART 连续接收 +ACK 发送

- UART 连续接收 +ACK 发送的设定例子的流程图如下所示：
- UART0进行连续接收并且将接收结束的ACK输出到P10。
  - DMA 传送使用DMA 的通道0。
  - DMA 启动源：软件触发（禁止通过中断触发DMA 传送）
  - 将UART接收数据寄存器0（RXD0）的FFF12H传送到RAM的FFE00H～FFE3FH（64字节）。

图 13-9 UART 连续接收 +ACK 发送的设定例子



注 如果 DMA 传送结束，就自动将 DST0 标志清“0”。  
因为只有在 DST0 标志为“0”时才允许写 DEN0 标志，所以当不等待 DMA0 中断（INTDMA0）的发生而结束 DMA 传送时，必须在将 DST0 标志置“0”后将 DEN0 标志置“0”（有关详细内容，请参照“13.5.5 通过软件强制结束”）。

备注 这是将软件触发用于 DMA 启动源的例子。  
如果不发送 ACK 而只进行 UART 的连续接收，就能将 UART 接收结束中断（INTSR0）设定为 DMA 启动源进行数据接收。

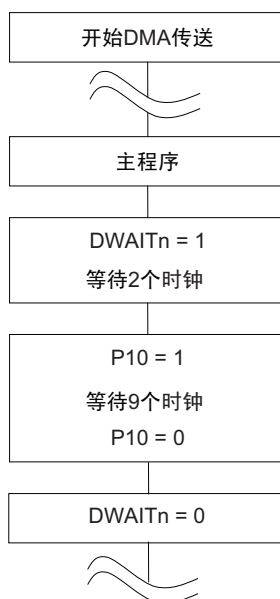
### 13.5.4 通过 DWAITn 位保留 DMA 传送

如果开始 DMA 传送，就在指令执行过程中进行传送。因此，此时 CPU 停止运行并且延迟 2 个时钟。如果这种情况会影响设备系统的正常运行，就能通过将 DWAITn 位置“1”，保留 DMA 传送。对于在保留期间发生的传送触发所对应的 DMA 传送，在解除保留后执行该 DMA 传送。但是，各通道只能保留一个传送触发，因此同一个通道即使在保留期间发生了 2 次或者更多次的传送触发，在解除保留后也只执行一次 DMA 传送。

以 P10 引脚输出 10 个时钟（工作频率）宽度的脉冲为例，如果在中途开始 DMA 传送，就增加到 12 个时钟。此时，能通过将 DWAITn 位置“1”来保留 DMA 传送。

在将 DWAITn 位置“1”后需要经过 2 个时钟，然后保留 DMA 传送。

图 13-10 通过 DWAITn 位保留 DMA 传送的设定例子



**注意** 要在使用 2 个通道 DMA 的过程中保留 DMA 传送时，必须保留全部通道的 DMA 传送（DWAIT0=DWAIT1=1）。如果在保留其他通道的 DMA 传送期间执行一个通道的 DMA 传送，就可能无法保留其他通道的 DMA。

备注 1. n: DMA 通道号（n=0、1）

2. 1 个时钟：1/f<sub>CLK</sub>（f<sub>CLK</sub>: CPU 时钟）

### 13.5.5 通过软件强制结束

在通过软件将 DSTn 位置“0”后最多需要经过 2 个时钟，然后停止 DMA 传送并且 DSTn 位变为“0”。因此，当不等待 DMA 中断（INTDMA<sub>n</sub>）的发生而通过软件强制结束 DMA 传送时，必须进行以下某种处理。

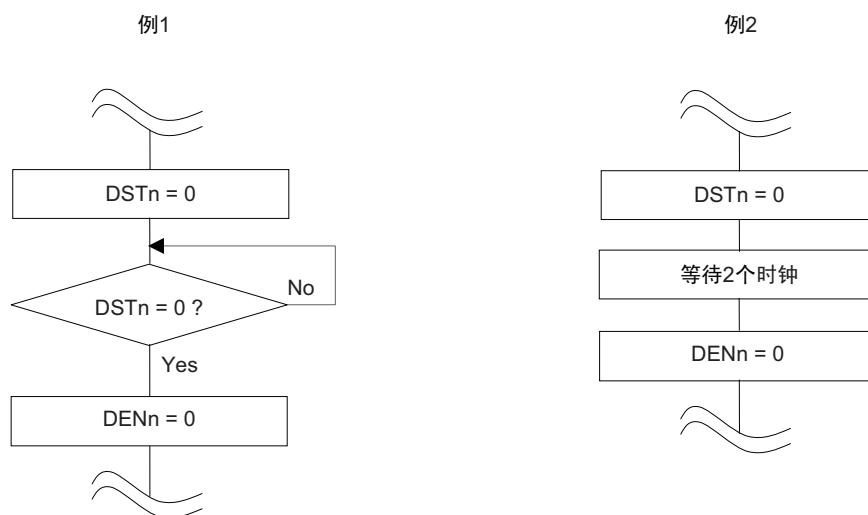
< 当使用 1 个通道的 DMA 时 >

- 在通过软件将 DSTn 位置“0”（当通过字节操作指令进行写操作时，DRCn=80H）后，通过轮询确认 DSTn 位已变为“0”，然后将 DENn 位置“0”（当通过字节操作指令进行写操作时，DRCn=00H）。
- 在通过软件将 DSTn 位置“0”（当通过字节操作指令进行写操作时，DRCn=80H）后经过 2 个时钟，然后将 DENn 位置“0”（当通过字节操作指令进行写操作时，DRCn=00H）。

< 当使用 2 个通道的 DMA 时 >

- 要在使用 2 个或者更多通道 DMA 的过程中通过软件强制结束 DMA 传送（DSTn=0）时，通过将所用全部通道的 DWAITn 位置“1”来保留 DMA 传送，然后将 DSTn 位清“0”。此后，通过将所用全部通道的 DWAITn 位清“0”来解除保留，然后将 DENn 位清“0”。

图 13-11 DMA 传送的强制结束 (1/2)



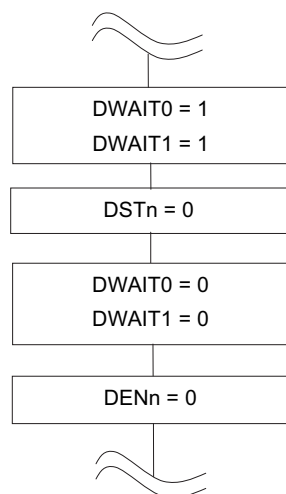
备注 1. n: DMA 通道号 (n=0、1)

2. 1 个时钟:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU 时钟)

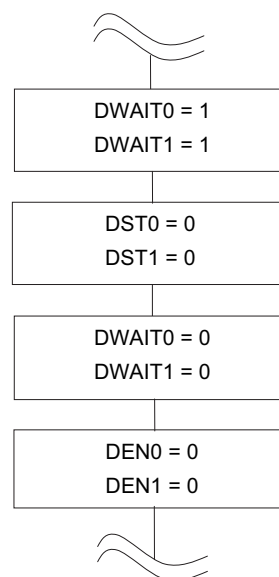
图 13-11 DMA 传送的强制结束 (2/2)

例3

- 在使用2个通道时，其中1个通道强制结束的步骤



- 在使用2个通道时，2个通道都强制结束的步骤



**注意** 在例 3 中，在将 `DWAITn` 位置“1”后不需要等待 2 个时钟。因为在从 `DSTn` 位清“0”到 `DENn` 位清“0”的期间已经至少经过了 2 个时钟，所以在将 `DSTn` 位清“0”后不需要等待 2 个时钟。

备注 1. n: DMA 通道号 (n=0、1)

2. 1 个时钟:  $1/f_{\text{CLK}}$  ( $f_{\text{CLK}}$ : CPU 时钟)



## 13.6 DMA 控制器的注意事项

### (1) DMA 的优先级

在 DMA 传送过程中，即使发生其他 DMA 通道的请求，也保留该通道的请求。在 DMA 传送结束后开始被保留的 DMA 传送。但是，当多个 DMA 请求同时发生时，优先级为 DMA 通道 0 > DMA 通道 1。

当 DMA 请求和中断请求同时发生时，优先 DMA 传送，然后执行中断处理。

### (2) DMA 应答时间

DMA 传送的应答时间如下所示。

表 13-3 DMA 传送的应答时间

	最短时间	最长时间
应答时间	3 个时钟	10 个时钟 <sup>注</sup>

注 当从内部 RAM 执行指令时，最长时间为 16 个时钟。

注意 1. 以上的应答时间不包括 DMA 传送的 2 个时钟。

2. 在执行 DMA 保留指令（参照 13.6(4)）时，各条件的最长应答时间加上在其条件下保留指令的执行时间。

3. 最长应答时间 +1 个时钟内的同一个通道的连续传送触发可能被忽视，因此不能设定。

备注 1 个时钟：1/f<sub>CLK</sub>（f<sub>CLK</sub>：CPU 时钟）

### (3) 待机模式中的运行

待机模式中的 DMA 控制器的运行如下所示。

表 13-4 待机模式中的 DMA 运行

状态	DMA 运行
HALT 模式	通常运行
STOP 模式	停止运行 当 DMA 传送和 STOP 指令发生竞争时，DMA 传送可能会遭到破坏，因此必须在执行 STOP 指令前停止 DMA。

## (4) DMA 传送的保留指令

即使发生 DMA 请求，也会在紧随着以下指令之后保留 DMA 传送。

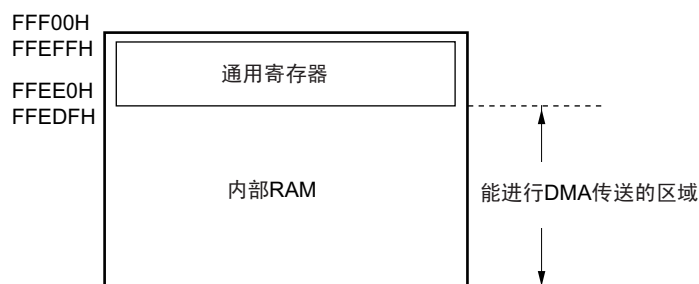
- CALL !addr16
- CALL \$!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- IF0L、IF0H、IF1L、MK0L、MK0H、MK1L、PR00L、PR00H、PR01L、PR10L、PR10H、PR11L 和 PSW 的各寄存器的位操作指令
- 存取数据闪存的指令

## (5) 指定通用寄存器区内或者内部 RAM 区外的地址时的运行

在 DMA 传送过程中，DMA RAM 地址寄存器 n (DRAn) 所示的地址进行递增。当该地址进入通用寄存器区内或者超过内部 RAM 区时，会出现以下情况。

- 从 SFR 传送到 RAM 的模式  
破坏该地址的数据。
- 从 RAM 传送到 SFR 的模式  
将不定的数据传送到 SFR。

以上 2 种情况都可能导致误动作或者系统的损坏，因此必须确保地址在通用寄存器区以外的内部 RAM 区内。



## (6) 存取数据闪存空间时的运行

如果存取数据闪存空间，就保留 DMA 传送，因此必须追加 DMA 保留指令。如果要在 1 条 DMA 传送指令之后存取数据闪存空间，就需要在此指令之间插入 3 个时钟的等待。

指令 1

DMA 传送

指令 2 ← 等待 3 个时钟。

MOV A, !DataFlash 空间

## 第 14 章 事件链接控制器（ELC）

### 14.1 ELC 的功能

事件链接控制器（ELC）将各外围功能输出的事件进行外围功能之间的相互链接（链接）。能通过事件链接不经过 CPU 而直接进行外围功能之间的协作运行。

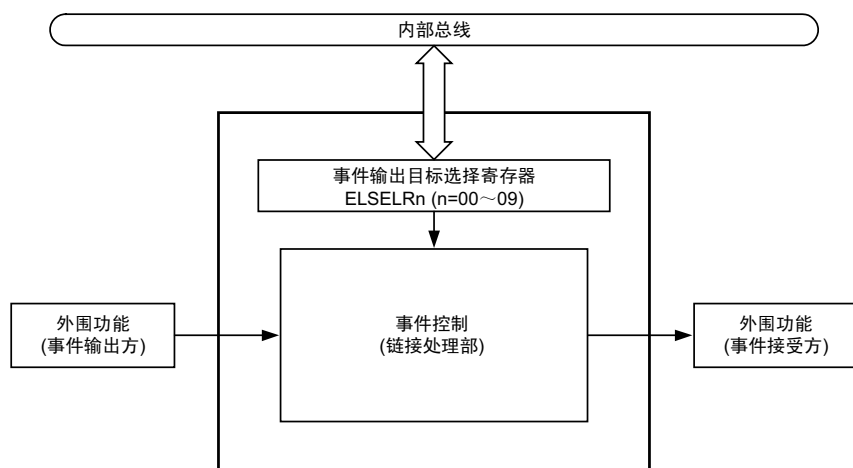
ELC 有以下功能：

- 能将 10 种外围功能的事件信号直接链接到指定的外围功能。
- 能将事件信号用作 3 种外围功能中的 1 种外围功能运行的启动源。

### 14.2 ELC 的结构

ELC 的框图如图 14-1 所示。

图 14-1 ELC 的框图



### 14.3 控制 ELC 的寄存器

控制 ELC 的寄存器如表 14-1 所示。

表 14-1 控制 ELC 的寄存器

项目	结构
控制寄存器	事件输出目标选择寄存器 n (ELSELRn)

备注 n=00 ~ 09

#### 14.3.1 事件输出目标选择寄存器 n (ELSELRn) (n=00 ~ 09)

ELSELRn 寄存器将各事件信号链接到事件接受方外围功能（链接目标外围功能）接受事件时的运行。

不能将多个事件输入链接到相同的事件输出目标（事件接受方）。否则，事件接受方外围功能的运行可能不定而无法正常地接受事件信号。另外，不能将事件链接发生源和事件输出目标设定为相同的功能。

必须在全部事件输出方的外围功能不产生事件信号的期间设定 ELSELRn 寄存器。

ELSELRn 寄存器（n=00 ~ 09）和外围功能的对应如表 14-2 所示，ELSELRn 寄存器（n=00 ~ 09）的设定值和链接目标外围功能接受事件时的运行的对应如表 14-3 所示。

图 14-2 事件输出目标选择寄存器 n (ELSELRn) 的格式

地址: F0300H (ELSELR00) ~ F0309H (ELSELR09)					复位后: 00H		R/W	
符号	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	0	ELSELRn1	ELSELRn0

ELSELRn1	ELSELRn0	事件链接的选择
0	0	禁止事件链接。
0	1	选择所链接的外围功能的运行注。
1	0	选择所链接的外围功能的运行注。
1	1	选择所链接的外围功能的运行注。

注 请参照“表 14-3 ELSELRn 寄存器（n=00 ~ 09）的设定值和链接目标外围功能接受事件时的运行的对应”。

表 14-2 ELSELRn 寄存器 (n=00 ~ 09) 和外围功能的对应

寄存器名	事件发生源 (事件输入 n 的输出源)	事件内容
ELSELR00	外部中断边沿检测 0	INTP0
ELSELR01	外部中断边沿检测 1	INTP1
ELSELR02	外部中断边沿检测 2	INTP2
ELSELR03	外部中断边沿检测 3	INTP3
ELSELR04	外部中断边沿检测 4	INTP4
ELSELR05	外部中断边沿检测 5	INTP5
ELSELR06	TAU 通道 00 的计数结束 / 捕捉结束	INTTM00
ELSELR07	TAU 通道 01 的计数结束 / 捕捉结束	INTTM01
ELSELR08	TAU 通道 02 的计数结束 / 捕捉结束	INTTM02
ELSELR09	TAU 通道 03 的计数结束 / 捕捉结束	INTTM03

表 14-3 ELSELRn 寄存器 (n=00 ~ 09) 的设定值和链接目标外围功能接受事件时的运行的对应

ELSELRn 寄存器的 ELSELR1 位和 ELSELR0 位	链接目标外围功能	接受事件时的运行
0000B	无	停止事件链接。
0001B	A/D 转换器	开始 A/D 转换。
0010B	DA0	实时输出
0011B	DA1	实时输出

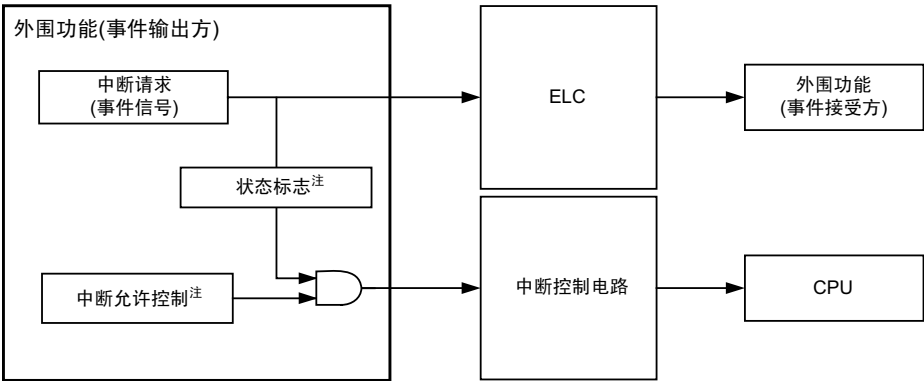
14.4 运行说明

将各外围功能产生的事件信号用作中断控制电路的中断请求所使用的路径和用作 ELC 事件所使用的路径相互独立。因此，各事件信号与中断控制无关，能用作事件接受方外围功能运行的事件信号。

中断处理和 ELC 的关系如图 14-3 所示。此图是以有中断请求状态标志和中断允许位（控制允许或者禁止）的外围功能为例的关系。

通过 ELC 接受事件的外围功能的运行是根据接受方外围功能在接收事件后的运行（参照“表14-3 ELSELn 寄存器（n=00 ~ 09）的设定值和链接目标外围功能接受事件时的运行的对应”）。

图 14-3 中断处理和 ELC 的关系



注 有些外围功能没有此功能。

表 14-4 接受事件的外围功能的响应

事件接受 目标 No.	事件链接目标的功能	事件接受后的运行	响应
1	A/D 转换器	A/D 转换	ELC 事件直接变为 A/D 转换的硬件触发。
2	D/A 转换器的通道 0	实时输出（通道 0）	在从发生 ELC 事件经过 2 个或者 3 个 $f_{CLK}$ 周期后开始通道 0 的 D/A 转换。
3	D/A 转换器的通道 1	实时输出（通道 1）	在从发生 ELC 事件经过 2 个或者 3 个 $f_{CLK}$ 周期后开始通道 1 的 D/A 转换。

## 第 15 章 中断功能

中断功能是指在程序执行过程中将处理程序切换为其他所需处理程序的一种功能。当转移目标处理结束时，返回到原来中止的执行程序。

### 15.1 中断功能的种类

中断功能有以下 2 种。

#### (1) 可屏蔽中断

这是接受屏蔽控制的中断。通过设定优先级指定标志寄存器（PR00L、PR00H、PR01L、PR10L、PR10H、PR11L），能将可屏蔽中断的优先级分成二组。高优先级的中断能对低优先级中断进行多重中断处理。如果同时发生相同优先级的多个中断请求，就根据向量中断处理的默认优先级进行处理。有关默认优先级，请参照表 15-1。

产生待机解除信号，解除 STOP 模式、HALT 模式和 SNOOZE 模式。

可屏蔽中断分为外部中断请求和内部中断请求。

外部：6，内部：12

#### (2) 软件中断

这是通过执行 BRK 指令而产生的向量中断。即使在禁止中断的状态下也接受软件中断，而且软件中断不受中断优先级控制。

### 15.2 中断源和结构

中断源除了有可屏蔽中断和软件中断以外还有最多 7 种复位源（参照表 15-1）。当发生复位或者各种中断请求而进行转移时，因为保存程序起始地址的向量码各为 2 字节，所以中断的转移目标地址为 00000H ~ 0FFFFH 的 64K 地址。

表 15-1 中断源一览表 (1/2)

中断处理	默认 优先级注 1	中断源		内部 / 外部	向量表地址	基本结构 类型注 2
		名称	触发			
可屏蔽	0	INTWDTI	看门狗定时器的间隔注 3 (上溢时间的 75%+1/2f <sub>IL</sub> )	内部	0004H	(A)
	1	INTLVI	电压检测注 4		0006H	
	2	INTP0	引脚输入边沿的检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	5	INTP3			000EH	
	6	INTP4			0010H	
	7	INTP5			0012H	
	8	INTAD	A/D 转换结束	内部	0014H	(A)
	9	INTIICA0	IICA0 通信结束		0016H	
	10	INTFL	保留注 5		0018H	
	11	INTDMA0	DMA0 的传送结束		001AH	
	12	INTDMA1	DMA1 的传送结束		001CH	
	13	INTST0 /INTCSI00	UART0 发送的传送结束或者缓冲器空中断/CSI00 的传送结束或者缓冲器空中断		001EH	
	14	INTSR0	UART0 接收的传送结束		0020H	
	15	INTSRE0	发生 UART0 接收的通信错误		0022H	
		INTTM01H	定时器通道 01 的计数结束或者捕捉结束 (高 8 位定时器工作时)			
	16	INTTM03H	定时器通道 03 的计数结束或者捕捉结束 (高 8 位定时器工作时)		0028H	
	17	INTIICA1	IICA1 通信结束		002AH	
	18	INTTM00	定时器通道 00 的计数结束或者捕捉结束		002CH	
	19	INTTM01	定时器通道 01 的计数结束或者捕捉结束 (16 位 / 低 8 位定时器工作时)		002EH	
	20	INTTM02	定时器通道 02 的计数结束或者捕捉结束		0030H	
	21	INTTM03	定时器通道 03 的计数结束或者捕捉结束 (16 位 / 低 8 位定时器工作时)		0032H	

- 注 1. 在发生多个可屏蔽中断时，默认优先级决定优先顺序。0 表示最高优先级，21 表示最低优先级。
2. 基本构成类型 (A) ~ (C) 分别对应图 15-1 的 (A) ~ (C)。
3. 这是将选项字节 (000C0H) 的 bit7 (WDTINT) 置“1”的情况。
4. 这是将电压检测电平寄存器 (LVIS) 的 bit7 (LVIMD) 置“0”的情况。
5. 用于闪存自编程库。



表 15-1 中断源一览表 (2/2)

中断处理	默认 优先级注 1	中断源		内部 / 外部	向量表地址	基本结构 类型注 2
		名称	触发			
软件	—	BRK	BRK 指令的执行	—	007EH	(C)
复位	—	RESET	RESET 引脚的输入	—	0000H	—
		POR	上电复位			
		LVD	电压检测注 3			
		WDT	看门狗定时器的上溢			
		TRAP	非法指令的执行注 4			
		IAW	非法存储器的存取			
		RPE	RAM 奇偶校验错误			

注 1. 在发生多个可屏蔽中断时，默认优先级决定优先顺序。0 表示最高优先级，21 表示最低优先级。

2. 基本构成类型 (A) ~ (C) 分别对应图 15-1 的 (A) ~ (C)。

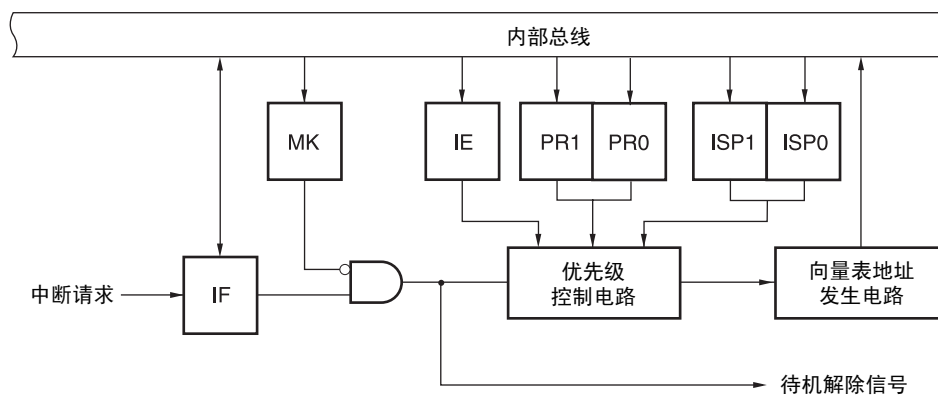
3. 这是将电压检测电平寄存器 (LVIS) 的 bit7 (LVIMD) 置“1”的情况。

4. 在执行指令码 FFH 时发生。

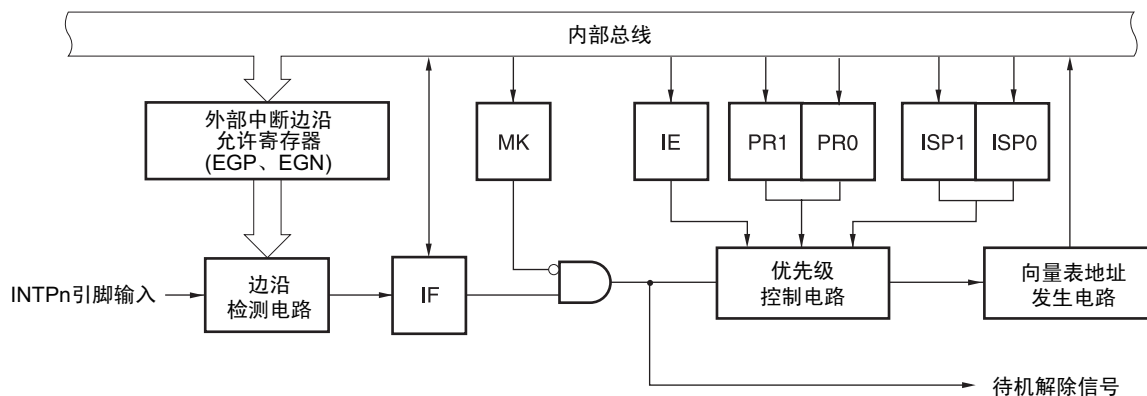
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

图 15-1 中断功能的基本结构

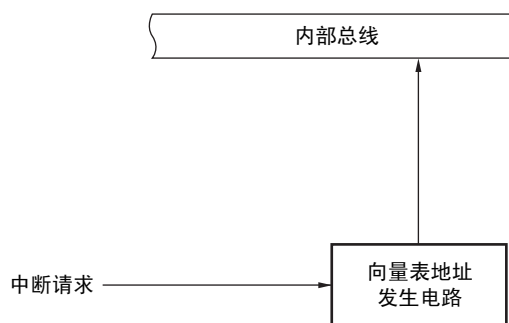
## (A) 内部可屏蔽中断



## (B) 外部可屏蔽中断 (INTPn)



## (C) 软件中断



IF : 中断请求标志  
 IE : 中断允许标志  
 ISP0 : 服务优先级标志 0  
 ISP1 : 服务优先级标志 1  
 MK : 中断屏蔽标志  
 PR0 : 优先级指定标志 0  
 PR1 : 优先级指定标志 1

备注 n=0 ~ 5

### 15.3 控制中断功能的寄存器

通过以下 6 种寄存器控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L)
- 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR10L、PR10H、PR11L)
- 外部中断上升沿允许寄存器 (EGP0)
- 外部中断下降沿允许寄存器 (EGN0)
- 程序状态字 (PSW)

各中断请求源对应的中断请求标志、中断屏蔽标志和优先级指定标志的名称如表 15-2 所示。

表 15-2 中断请求源对应的各种标志

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		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	
INTAD	ADIF	IF0H	ADMK	MK0H	ADPR0、ADPR1	PR00H、 PR10H
INTIICA0	IICAIF0		IICAMK0		IICAPR00、IICAPR10	
INTFL	FLIF		FLMK		FLPR0、FLPR1	
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	CSIF00 注 1		CSIMK00 注 1		CSIPR000、CSIPR100 注 1	
INTSR0	SRIF0		SRMK0		SRPR00、SRPR10	
INTSRE0 注 2	SREIF0 注 2		SREMK0 注 2		SREPR00、SREPR10 注 2	
INTTM01H 注 2	TMIF01H 注 2		TMMK01H 注 2		TMPR001H、TMPR101H 注 2	
INTTM03H	TMIF03H	IF1L	TMMK03H	MK1L	TMPR003H、TMPR103H	PR01L、 PR11L
INTIICA1	IICAIF1		IICAMK1		IICAPR01、IICAPR11	
INTTM00	TMIF00		TMMK00		TMPR000、TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001、TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002、TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003、TMPR103	

注 1. 如果发生 INTST0 或者 INTCSI00 的中断源，就将 IF0H 寄存器的 bit5 置“1”。MK0H、PR00H、PR10H 寄存器的 bit5 对应这 2 个中断源。

2. 因为 UART0 接收的错误中断和 TAU0 的通道 1（高 8 位定时器工作时）的中断共用中断请求源对应的各种标志，所以不能同时使用。当不使用 UART0 接收的错误中断（EOC01=0）时，能同时使用 UART0 和 TAU0 的通道 1（高 8 位定时器工作时）。如果发生 INTSRE0 或者 INTTM01H 的中断源，就将 IF0H 寄存器的 bit7 置“1”。MK0H、PR00H、PR10H 寄存器的 bit7 对应这 2 个中断源。

15.3.1 中断请求标志寄存器（IF0L、IF0H、IF1L）

通过发生对应的中断请求或者执行指令，将中断请求标志置“1”。通过接受中断请求或者产生复位信号或者执行指令，将中断请求标志清“0”。

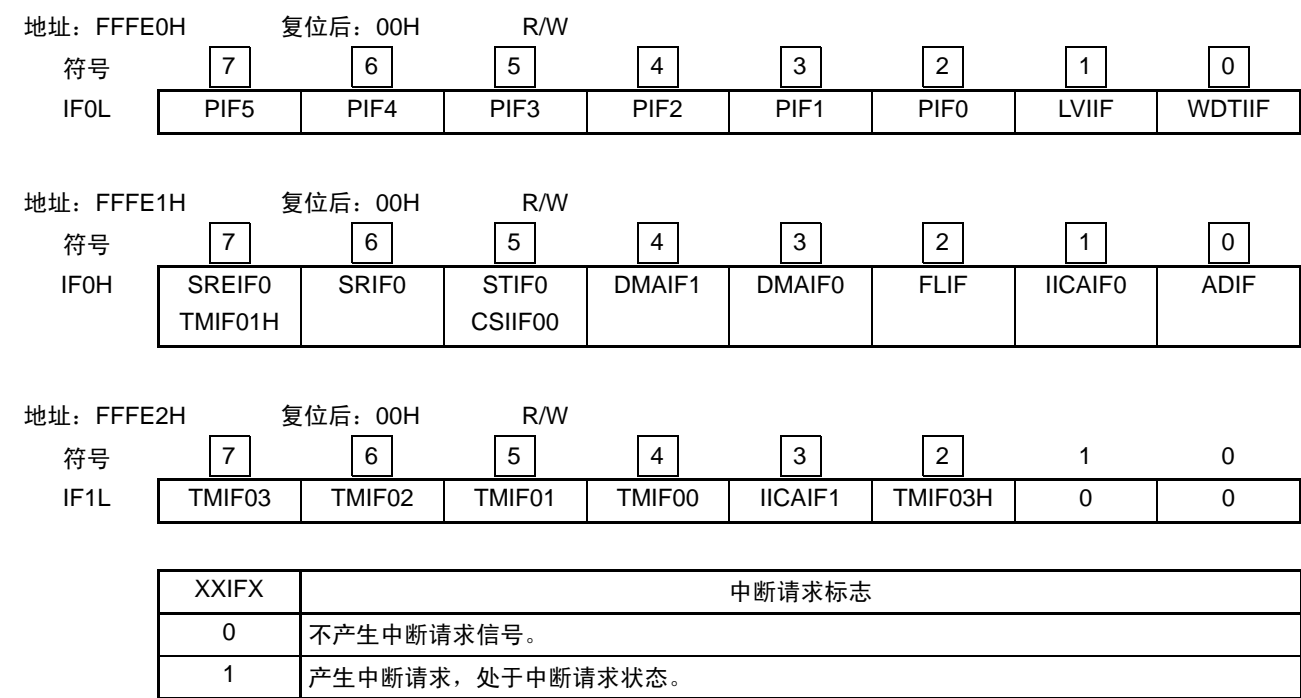
在接受中断时，首先自动清除中断请求标志，然后进入中断程序。

通过 1 位或者 8 位存储器操作指令设定 IF0L、IF0H、IF1L 寄存器。当将 IF0L 寄存器和 IF0H 寄存器一起用作 IF0 的 16 位寄存器时，通过 16 位存储器操作指令进行设定。

在产生复位信号后，这些寄存器的值变为“00H”。

备注 当执行写此寄存器的指令时，指令执行时钟数增加 2 个时钟。

图 15-2 中断请求标志寄存器（IF0L、IF0H、IF1L）的格式



注意 1. 必须给未分配的位设定初始值。

2. 在操作中中断请求标志寄存器的标志时，必须使用位存储器操作指令（CLR1）。当使用 C 语言描述时，编译后的汇编程序需要变为位存储器操作指令（CLR1），因此必须使用类似“IF0L.0=0;”或者“\_asm("clr1 IF0L, 0");”的位操作指令。

如果在 C 语言中使用类似于“IF0L &= 0xfe;”的 8 位存储器操作指令描述的情况下进行编译，就变为 3 条指令的汇编程序。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

此时，即使在“mov a, IF0L”与“mov IF0L, a”之间将同一中断请求标志寄存器（IF0L）的其他位的请求标志置“1”，该标志也会被“mov IF0L, a”清“0”。因此，在 C 语言中使用 8 位存储器操作指令时，必须注意。

15.3.2 中断屏蔽标志寄存器（MK0L、MK0H、MK1L）

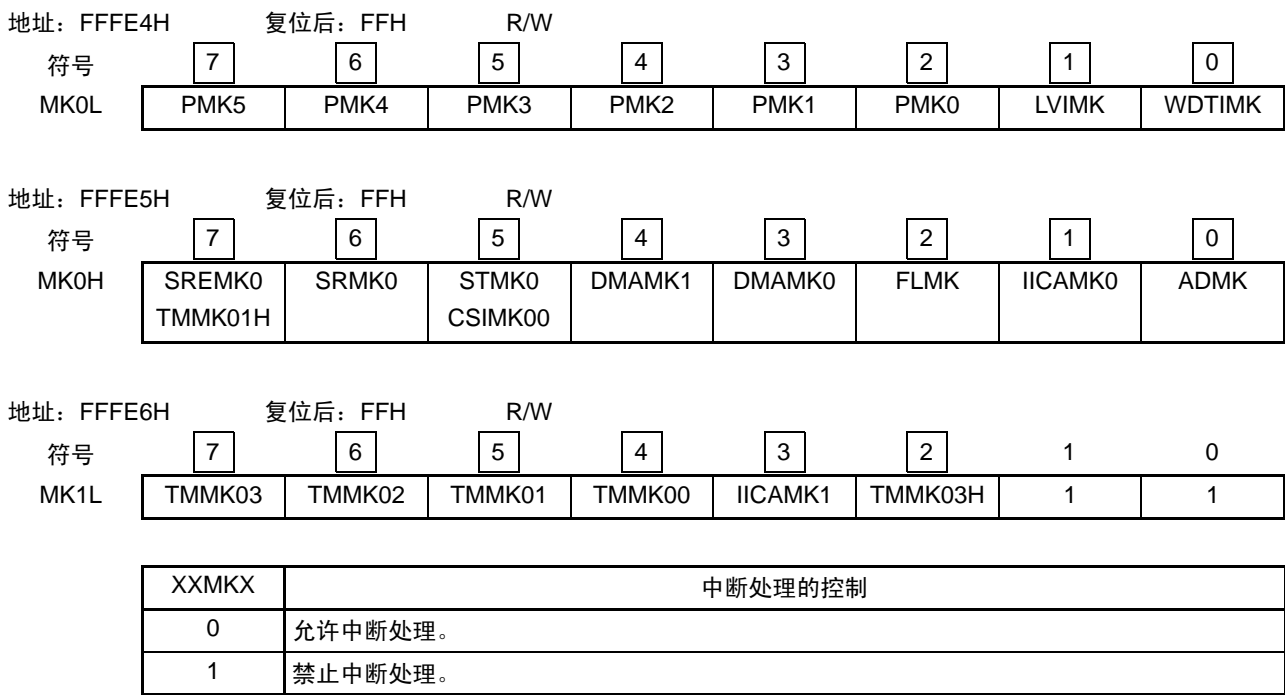
中断屏蔽标志设定允许或者禁止对应的可屏蔽中断处理。

通过 1 位或者 8 位存储器操作指令设定 MK0L、MK0H、MK1L 寄存器。当将 MK0L 寄存器和 MK0H 寄存器一起用作 MK0 的 16 位寄存器时，通过 16 位存储器操作指令进行设定。

在产生复位信号后，这些寄存器的值变为“FFH”。

备注 当执行写此寄存器的指令时，指令执行时钟数增加 2 个时钟。

图 15-3 中断屏蔽标志寄存器（MK0L、MK0H、MK1L）的格式



注意 必须给未分配的位设定初始值。

15.3.3 优先级指定标志寄存器（PR00L、PR00H、PR01L、PR10L、PR10H、PR11L）

优先级指定标志寄存器设定对应的可屏蔽中断优先级。

通过组合 PR0xy 寄存器和 PR1xy 寄存器来设定优先级（xy=0L、0H、1L、1H、2L、2H）。

通过 1 位或者 8 位存储器操作指令设定 PR00L、PR00H、PR01L、PR10L、PR10H、PR11L 寄存器。当将 PR00L 寄存器和 PR00H 寄存器、PR10L 寄存器和 PR10H 寄存器一起用作 PR00、PR10 的 16 位寄存器时，通过 16 位存储器操作指令进行设定。

在产生复位信号后，这些寄存器的值变为“FFH”。

备注 当执行写此寄存器的指令时，指令执行时钟数增加 2 个时钟。

图 15-4 优先级指定标志寄存器（PR00L、PR00H、PR01L、PR10L、PR10H、PR11L）的格式

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: FFFECH      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: FFFE9H      复位后: FFH      R/W

符号	7	6	5	4	3	2	1	0
PR00H	SREPR00 TMPR001H	SRPR00	STPR00 CSIPR000	DMAPR01	DMAPR00	FLPR0	IICAPR00	ADPR0

地址: FFFEDH      复位后: FFH      R/W

符号	7	6	5	4	3	2	1	0
PR10H	SREPR10 TMPR101H	SRPR10	STPR10 CSIPR100	DMAPR11	DMAPR10	FLPR1	IICAPR10	ADPR1

地址: FFFEAH      复位后: FFH      R/W

符号	7	6	5	4	3	2	1	0
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR01	TMPR003H	1	0

地址: FFEEH      复位后: FFH      R/W

符号	7	6	5	4	3	2	1	0
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR11	TMPR103H	1	1

XXPR1X	XXPR0X	优先级的选择
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 必须给未分配的位设定初始值。

### 15.3.4 外部中断上升沿允许寄存器（EGP0）和外部中断下降沿允许寄存器（EGN0）

这些寄存器设定 INTP0 ~ INTP5 的有效边沿。

通过 1 位或者 8 位存储器操作指令设定 EGP0 寄存器和 EGN0 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 15-5 外部中断上升沿允许寄存器（EGP0）和外部中断下降沿允许寄存器（EGN0）的格式

地址: FFF38H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FFF39H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效边沿的选择 (n=0 ~ 5)
0	0	禁止检测边沿。
0	1	下降沿
1	0	上升沿
1	1	上升和下降的双边沿

对应 EGPn 位和 EGNn 位的端口如表 15-3 所示。

表 15-3 对应 EGPn 位和 EGNn 位的中断请求信号

检测允许位		中断请求信号
EGP0	EGN0	INTP0
EGP1	EGN1	INTP1
EGP2	EGN2	INTP2
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP5	EGN5	INTP5

**注意** 如果将外部中断功能使用的输入端口切换到输出模式，就可能检测有效边沿，产生 INTPn 中断。

当切换到输出模式时，必须在设定为禁止边沿检测后（EGPn、EGNn=0、0）将端口模式寄存器（PMxx）置“0”。

备注 1. 有关边沿检测端口，请参照“2.1 引脚功能一览表”。

2. n=0 ~ 5



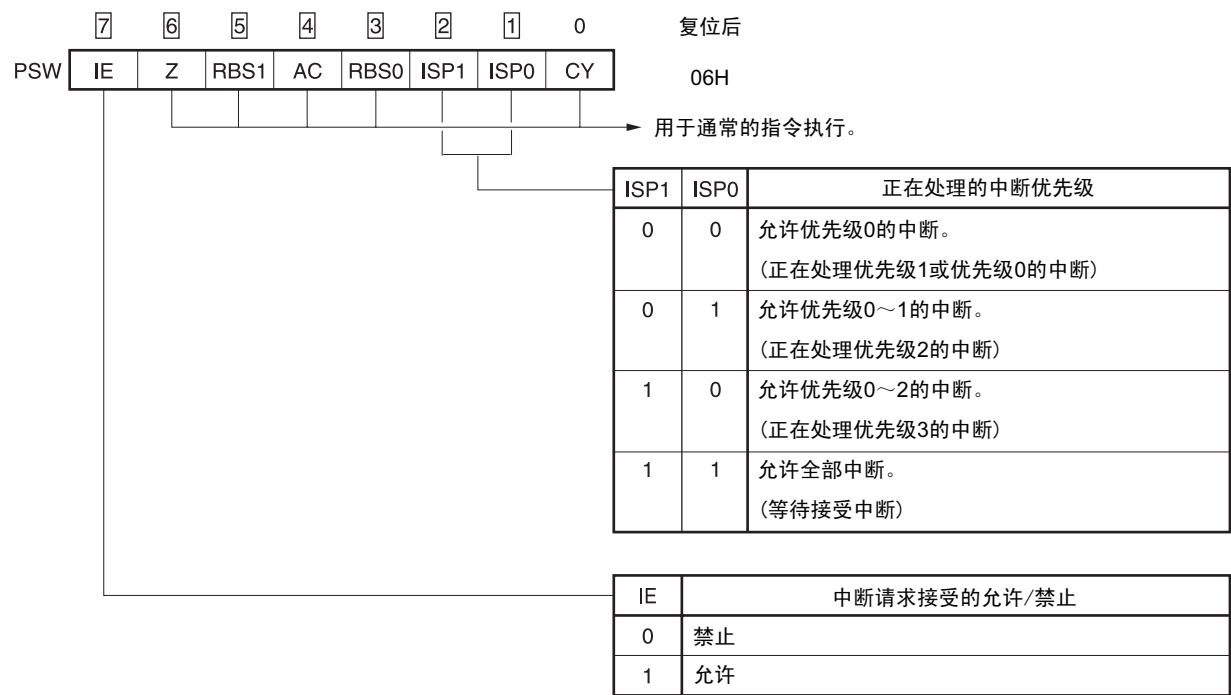
15.3.5 程序状态字（PSW）

程序状态字是保存指令执行结果和中断请求当前状态的寄存器，映射设定允许或者禁止可屏蔽中断的 IE 标志以及控制多重中断处理的 ISP0 标志和 ISP1 标志。

除了能以 8 位为单位读写 PSW 以外，还能利用位操作指令和专用指令（EI 和 DI）操作 PSW。在接受向量中断请求及执行 BRK 指令时，自动将 PSW 的内容压栈，并且将 IE 标志恢复为“0”。在接受可屏蔽中断请求时，如果接受的中断优先级指定标志的内容不为“00”，就将其减 1 的值传送到 ISP0 标志和 ISP1 标志。PSW 的内容也能通过 PUSH PSW 指令进行压栈，通过 RETI、RETB、POP PSW 指令进行退栈。

在产生复位信号后，PSW 的值变为“06H”。

图 15-6 程序状态字的结构



15.4 中断处理的操作

15.4.1 可屏蔽中断请求的接受

如果中断请求标志被置“1”并且该中断请求的屏蔽（MK）标志已被清“0”，就进入能接受可屏蔽中断请求的状态，在中断允许状态（IE 标志已被置“1”）下接受向量中断请求。但是，在处理优先级高的中断请求的过程中，不接受被指定的低优先级的中断请求。

从产生可屏蔽中断请求到进行向量中断处理的时间如表 15-4 所示。

有关中断请求的接受时序，请参照图 15-8 和图 15-9。

表 15-4 从产生可屏蔽中断请求到处理的时间

	最短时间	最长时间注
处理时间	9 个时钟	16 个时钟

注 从内部 RAM 区执行指令时除外。

备注 1 个时钟：1/f<sub>CLK</sub>（f<sub>CLK</sub>：CPU 时钟）

如果同时发生多个可屏蔽中断请求，就从优先级指定标志所指定的高优先级的请求开始接受。如果优先级指定标志所指定的优先级相同，就从默认优先级高的请求开始接受。

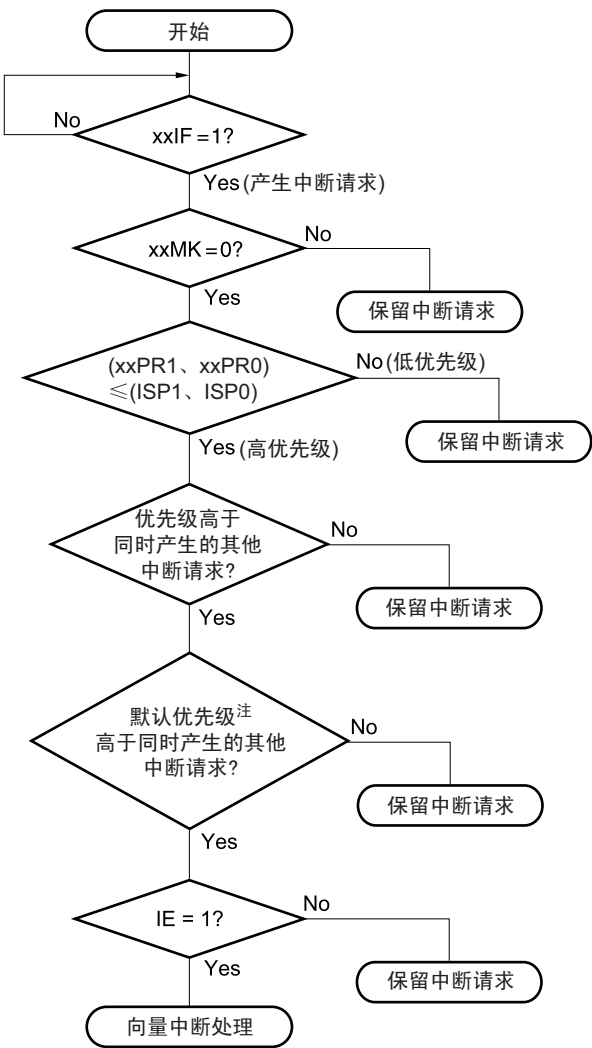
被保留的中断请求在可接受的状态时被接受。

中断请求的接受算法如图 15-7 所示。

如果接受可屏蔽中断请求，就按照程序状态字（PSW）、程序计数器（PC）的顺序将内容压栈，将 IE 标志复位为“0”，并且将接受的中断优先级指定标志的内容传送到 ISP1 标志和 ISP0 标志。将各中断请求决定的向量表中的数据装入 PC 并且进行转移。

能使用 RETI 指令从中断返回。

图 15-7 接受中断请求的处理算法



- xxIF : 中断请求标志
- xxMK : 中断屏蔽标志
- xxPR0 : 优先级指定标志 0
- xxPR1 : 优先级指定标志 1
- IE : 可屏蔽中断请求的接受控制标志 (1= 允许、0= 禁止)
- ISP0、ISP1 : 表示正在处理的中断优先级的标志 (参照图 15-6)

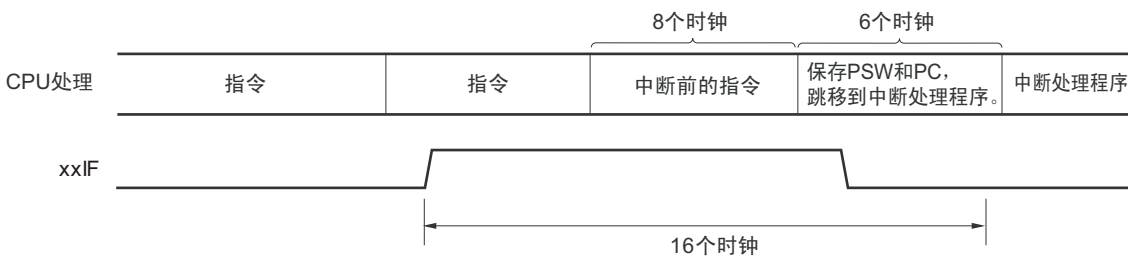
注 有关默认优先级，请参照“表 15-1 中断源一览表”。

图 15-8 中断请求的接受时序（最短时间）



备注 1 个时钟:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU 时钟)

图 15-9 中断请求的接受时序（最长时间）



备注 1 个时钟:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU 时钟)

### 15.4.2 软件中断请求的接受

通过执行 BRK 指令接受软件中断请求，不能禁止软件中断。

如果接受软件中断请求，就按照程序状态字（PSW）、程序计数器（PC）的顺序将内容压栈，将 IE 标志复位为“0”，并且将向量表（0007EH、0007FH）的内容装入 PC 进行转移。

能使用 RETB 指令从软件中断返回。

**注意** 不能使用 RETI 指令从软件中断返回。

### 15.4.3 多重中断处理

在中断处理过程中接受了另一个中断请求的情况称为多重中断。

除非中断请求为接受允许状态（IE=1），否则不会发生多重中断。在接受中断请求时，中断请求为接受被禁止状态（IE=0）。因此，如果要允许多重中断，就需要在中断处理过程中用 EI 指令将 IE 标志置“1”，进入中断允许状态。

另外，即使在中断允许状态下也可能不允许多重中断，由中断优先级进行控制。中断优先级有默认优先级和可编程优先级，通过可编程优先级控制多重中断。

在中断允许状态下，如果产生的中断请求的优先级高于正在处理的中断优先级，就作为多重中断进行接受。如果产生的中断请求的优先级等于或者低于正在处理的中断优先级，就不作为多重中断进行接受。但是，如果在处理优先级 0 的中断过程中将 IE 标志置“1”，就允许优先级 0 的其他中断。

保留因中断被禁止或者优先级低而不允许多重中断的中断请求，在当前的中断处理结束后至少执行一条主处理的指令，然后接受被保留的中断请求。

能发生多重中断的中断请求的关系和多重中断的例子如表 15-5 和图 15-10 所示。

表 15-5 在中断处理过程中能发生多重中断的中断请求的关系

多重中断请求  正在处理的中断		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	
可屏蔽中断	ISP1=0 ISP0=0	○	×	×	×	×	×	×	×	○
	ISP1=0 ISP0=1	○	×	○	×	×	×	×	×	○
	ISP1=1 ISP0=0	○	×	○	×	○	×	×	×	○
软件中断		○	×	○	×	○	×	○	×	○

备注 1. ○：能发生多重中断。

2. ×：不能发生多重中断。

3. ISP0、ISP1 和 IE 是 PSW 中的标志。

ISP1=0、ISP0=0：正在处理优先级 1 或者优先级 0 的中断。

ISP1=0、ISP0=1：正在处理优先级 2 的中断。

ISP1=1、ISP0=0：正在处理优先级 3 的中断。

ISP1=1、ISP0=1：等待接受中断（允许全部中断）。

IE=0：禁止接受中断请求。

IE=1：允许接受中断请求。

4. PR 是 PR00L、PR00H、PR01L、PR10L、PR10H、PR11L 寄存器中的标志。

PR=00：通过 xxPR1x=0、xxPR0x=0 指定优先级 0（高优先级）。

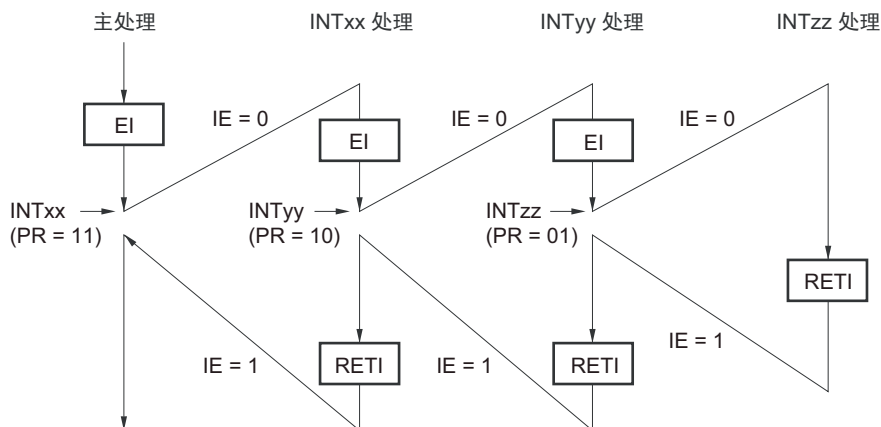
PR=01：通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

PR=10：通过 xxPR1x=1、xxPR0x=0 指定优先级 2。

PR=11：通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

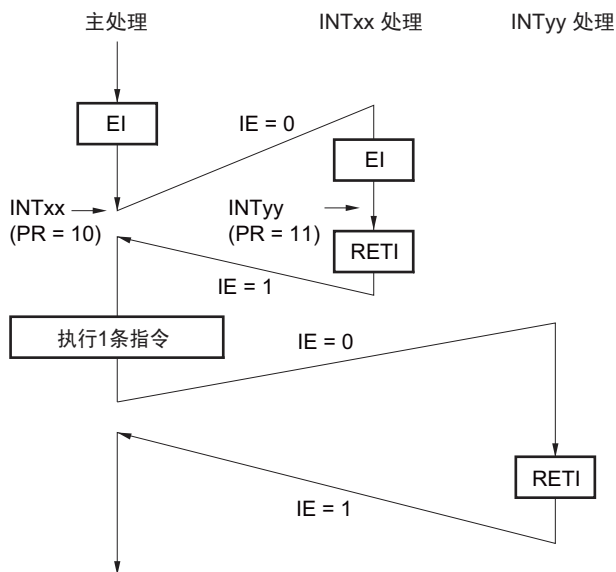
图 15-10 多重中断的例子 (1/2)

例 1. 发生 2 次多重中断的例子



在处理中断 INTxx 的过程中接受 INTyy 和 INTzz 的 2 个中断请求，发生多重中断。必须在接受各中断请求之前发行 EI 指令，使中断请求处于允许接受的状态。

例 2. 因控制优先级而不发生多重中断的例子



在处理中断 INTxx 的过程中，发生的 interrupt 请求 INTyy 因中断优先级低于 INTxx 而不被接受，所以不发生多重中断。保留 INTyy 请求，在执行一条主处理的指令后接受被保留的 INTyy 请求。

PR=00 : 通过 xxPR1x=0、xxPR0x=0 指定优先级 0 (高优先级)。

PR=01 : 通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

PR=10 : 通过 xxPR1x=1、xxPR0x=0 指定优先级 2。

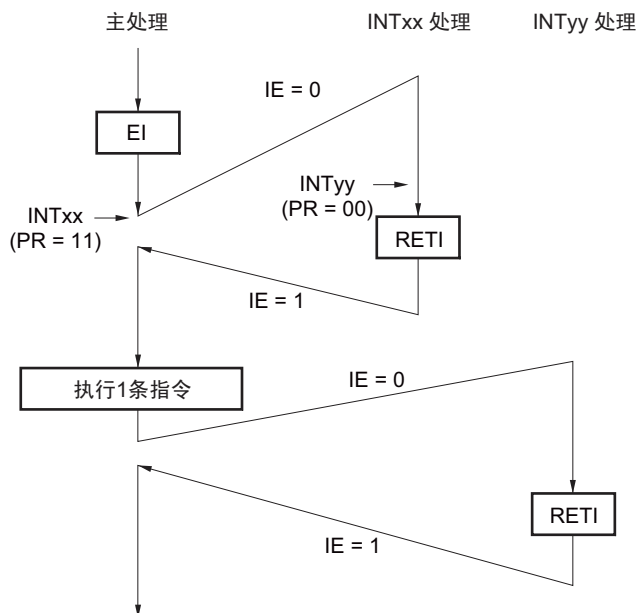
PR=11 : 通过 xxPR1x=1、xxPR0x=1 指定优先级 3 (低优先级)。

IE=0 : 禁止接受中断请求。

IE=1 : 允许接受中断请求。

图 15-10 多重中断的例子 (2/2)

例 3. 因不允许中断而不发生多重中断的例子



在处理中断 INTxx 的过程中，因不允许中断（未发行 EI 指令）而不接受中断请求 INTyy，所以不发生多重中断。保留 INTyy 请求，在执行一条主处理的指令后接受被保留的 INTyy 请求。

PR=00 : 通过 xxPR1x=0、xxPR0x=0 指定优先级 0（高优先级）。

PR=01 : 通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

PR=10 : 通过 xxPR1x=1、xxPR0x=0 指定优先级 2。

PR=11 : 通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

IE=0 : 禁止接受中断请求。

IE=1 : 允许接受中断请求。



15.4.4 中断请求的保留

有些指令，即使在执行指令的过程中发生了中断请求，也可能将中断请求保留到下一条指令执行结束为止。这些指令（中断请求保留指令）如下所示。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- IF0L、IF0H、IF1L、MK0L、MK0H、MK1L、PR00L、PR00H、PR01L、PR10L、PR10H和PR11L的各寄存器的写指令

中断请求的保留时序如图 15-11 所示。

图 15-11 中断请求的保留



- 备注 1. 指令 N：中断请求的保留指令
2. 指令 M：中断请求保留指令以外的指令

## 第 16 章 待机功能

### 16.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下 3 种模式。

#### (1) HALT 模式

通过执行 HALT 指令进入 HALT 模式。HALT 模式是停止 CPU 运行时钟的模式。在设定 HALT 模式前，如果高速系统时钟振荡电路或者高速内部振荡器正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到 STOP 模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

#### (2) STOP 模式

通过执行 STOP 指令进入 STOP 模式。STOP 模式是停止高速系统时钟振荡电路和高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小 CPU 的工作电流。

因为 STOP 模式能通过中断请求来解除，所以也能进行间歇运行。但是，在 X1 时钟的情况下，因为在解除 STOP 模式时需要确保振荡稳定的等待时间，所以如果一定要通过中断请求立即开始处理，就必须选择 HALT 模式。

#### (3) SNOOZE 模式

通过 CSIp 或者 UARTq 的数据接收产生的 A/D 转换请求，解除 STOP 模式，不需要 CPU 运行而进行 CSIp 或者 UARTq 的数据接收以及 A/D 转换。只有在选择高速内部振荡器作为 CPU/ 外围硬件时钟 ( $f_{CLK}$ ) 时才能设定 SNOOZE 模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设定为待机模式前的内容，并且还保持输入 / 输出端口的输出锁存器和输出缓冲器的状态。

- 注意 1. 在转移到 STOP 模式时，必须在停止以主系统时钟运行的外围硬件后执行 STOP 指令（SNOOZE 模式设定单元除外）。
2. 在 SNOOZE 模式中使用 CSIp、UARTq 或者 A/D 转换器时，必须在转移到 STOP 模式前设定串行待机控制寄存器 m (SSCm) 和 A/D 转换器模式寄存器 2 (ADM2)。详细内容请参照“9.3 控制 A/D 转换器的寄存器”和“11.3 控制串行阵列单元的寄存器”。
3. 为了降低 A/D 转换器的功耗，必须将 A/D 转换器模式寄存器 0 (ADM0) 的 bit7 (ADCS) 和 bit0 (ADCE) 清“0”，在停止 A/D 转换运行后执行 STOP 指令。
4. 能通过选项字节选择在 HALT 模式或者 STOP 模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“第 22 章 选项字节”。

备注 p=00, q=0, m=0

## 16.2 控制待机功能的寄存器

这是控制解除 STOP 模式时的振荡稳定时间的寄存器，有以下 2 种。

- 振荡稳定时间计数器的状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

**备注** 有关控制运行、停止、切换时钟的寄存器，请参照“第 5 章 时钟发生电路”。有关控制 SNOOZE 模式功能的寄存器，请参照“第 9 章 A/D 转换器”和“第 11 章 串行阵列单元”。

16.2.1 振荡稳定时间计数器的状态寄存器（OSTC）

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

- 当CPU时钟为高速内部振荡器时钟并且开始X1时钟的振荡时
- 在CPU时钟为高速内部振荡器时钟并且在X1时钟振荡的状态下转移到STOP模式后解除STOP模式时

能通过 1 位或者 8 位存储器操作指令读 OSTC 寄存器。

通过复位信号的产生（通过 RESET 的输入、POR、LVD、WDT 以及执行非法指令引起的复位）、STOP 指令或者 MSTOP 位（时钟运行状态控制寄存器（CSC）的 bit7）为“1”，此寄存器的值变为“00H”。

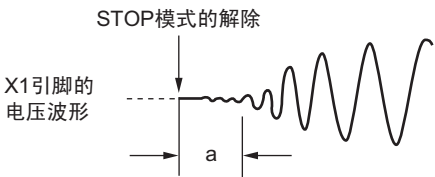
图 16-1 振荡稳定时间计数器的状态寄存器（OSTC）的格式

地址：FFFA2H      复位后：00H      R

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

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

- 注意 1. 在经过上述时间后，各位从 MOST8 位开始依次变为“1”并且保持“1”的状态。
2. 振荡稳定时间计数器只在振荡稳定时间选择寄存器（OSTS）所设振荡稳定时间内进行计数。如果在 CPU 时钟为高速内部振荡器时钟的情况下进入 STOP 模式，就必须在解除 STOP 模式时设定满足以下条件的 OSTS 寄存器的振荡稳定时间。
- 期待的 OSTC 寄存器的振荡稳定时间 ≤ OSTS 寄存器设定的振荡稳定时间
- 因此必须注意，解除 STOP 模式后的 OSTC 寄存器只设定 OSTS 寄存器所设振荡稳定时间内的状态。
3. X1 时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图 a）。



备注  $f_X$ : X1 时钟振荡频率

16.2.2 振荡稳定时间选择寄存器（OSTS）

这是选择在解除 STOP 模式时 X1 时钟的振荡稳定时间的寄存器。

如果选择 X1 时钟作为 CPU 时钟，就在解除 STOP 模式后等待 OSTS 寄存器设定的时间。

如果选择高速内部振荡器时钟作为 CPU 时钟，就必须在解除 STOP 模式后通过振荡稳定时间计数器的状态寄存器（OSTC）确认是否经过振荡稳定时间。能通过 OSTC 寄存器确认 OSTS 寄存器事先设定的时间。

通过 8 位存储器操作指令设定 OSTS 寄存器。

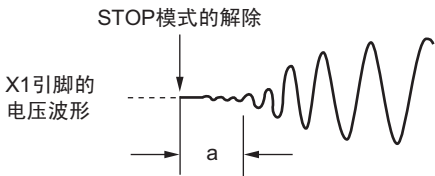
在产生复位信号后，此寄存器的值变为“07H”。

图 16-2 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 <sub>X</sub> =10MHz	f <sub>X</sub> =20MHz
0	0	0	2 <sup>8</sup> /f <sub>X</sub>	25.6μs	12.8μs
0	0	1	2 <sup>9</sup> /f <sub>X</sub>	51.2μs	25.6μs
0	1	0	2 <sup>10</sup> /f <sub>X</sub>	102μs	51.2μs
0	1	1	2 <sup>11</sup> /f <sub>X</sub>	204μs	102μs
1	0	0	2 <sup>13</sup> /f <sub>X</sub>	819μs	409μs
1	0	1	2 <sup>15</sup> /f <sub>X</sub>	3.27ms	1.63ms
1	1	0	2 <sup>17</sup> /f <sub>X</sub>	13.1ms	6.55ms
1	1	1	2 <sup>18</sup> /f <sub>X</sub>	26.2ms	13.1ms

- 注意 1. 如果要在 CPU 时钟为 X1 时钟的情况下转移到 STOP 模式，就必须在执行 STOP 指令前设定 OSTS 寄存器。
2. 要更改 OSTS 寄存器的设定时，必须在更改前确认 OSTC 寄存器的计数已结束。
3. 不能在 X1 时钟振荡稳定时间内更改 OSTS 寄存器。
4. 振荡稳定时间计数器只在 OSTS 寄存器所设振荡稳定时间内进行计数。如果在 CPU 时钟为高速内部振荡器时钟的情况下进入 STOP 模式，就必须在解除 STOP 模式时设定满足以下条件的 OSTS 寄存器的振荡稳定时间。
- 期待的 OSTC 寄存器的振荡稳定时间 ≤ OSTS 寄存器设定的振荡稳定时间
- 因此必须注意，解除 STOP 模式后的 OSTC 寄存器只设定 OSTS 寄存器所设振荡稳定时间内的状态。
5. X1 时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图 a）。



备注 f<sub>X</sub>: X1 时钟振荡频率

## 16.3 待机功能的运行

### 16.3.1 HALT 模式

#### (1) HALT 模式

通过执行 HALT 指令设定为 HALT 模式。无论设定前的 CPU 时钟是高速系统时钟还是高速内部振荡器时钟，都能进行设定。

HALT 模式中的运行状态如下所示。

**注意** 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除 HALT 模式。因此，即使在此情况下执行 HALT 指令，也不转移到 HALT 模式。

表 16-1 HALT 模式中的运行状态

HALT 模式的设定 项目			在 CPU 以主系统时钟运行的过程中执行 HALT 指令的情况					
			CPU 以高速内部振荡器时钟（f <sub>IH</sub> ）运行	CPU 以 X1 时钟（f <sub>X</sub> ）运行	CPU 以外部主系统时钟（f <sub>EX</sub> ）运行			
系统时钟			停止给 CPU 提供时钟。					
主系统时钟	f <sub>IH</sub>	继续运行（不能停止）。		禁止运行。				
	f <sub>X</sub>	禁止运行。		继续运行（不能停止）。	不能运行。			
	f <sub>EX</sub>			不能运行。	继续运行（不能停止）。			
	f <sub>IL</sub>	通过选项字节（000C0H）的 bit0（WDSTBYON）和 bit4（WDTON）进行设定。						
CPU			停止运行。					
代码闪存			停止运行。					
数据闪存								
RAM								
端口（锁存器）								
定时器阵列单元			能运行。					
看门狗定时器			参照“第 8 章 看门狗定时器”。					
时钟输出 / 蜂鸣器输出			能运行。					
A/D 转换器								
D/A 转换器								
串行阵列单元（SAU）								
串行接口（IICA）								
DMA 控制器								
事件链接控制器（ELC）						能在可运行的功能块之间进行链接。		
上电复位功能						能运行。		
电压检测功能								
外部中断								
CRC 运算功能	高速 CRC							
	通用 CRC	在 RAM 区的运算中执行 DMA 时，能运行。						
RAM 奇偶校验错误检测功能			在执行 DMA 时，能运行。					
RAM 保护功能								
SFR 保护功能								
非法存储器存取检测功能								

备注 停止运行：在转移到 HALT 模式时自动停止运行。

禁止运行：在转移到 HALT 模式前停止运行。

$f_{IH}$  : 高速内部振荡器时钟

$f_X$  : X1 时钟

$f_{EX}$  : 外部主系统时钟

$f_{IL}$  : 低速内部振荡器时钟

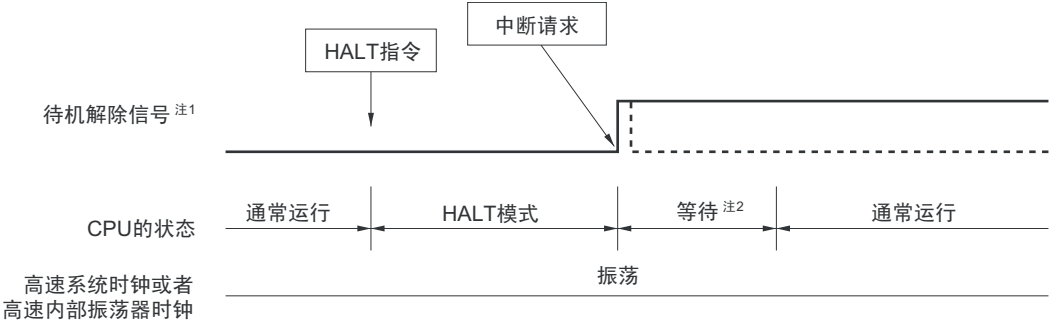
(2) HALT 模式的解除

能通过以下 2 种方法解除 HALT 模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除 HALT 模式。然后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图 16-3 通过中断请求解除 HALT 模式



注 1. 有关待机解除信号的详细内容，请参照图 15-1。

2. HALT 模式解除等待时间

- 进行向量中断处理的情况  
主系统时钟 : 15 ~ 16 个时钟
- 不进行向量中断处理的情况  
主系统时钟 : 9 ~ 10 个时钟

备注 虚线表示接受解除了待机模式的中断请求的情况。

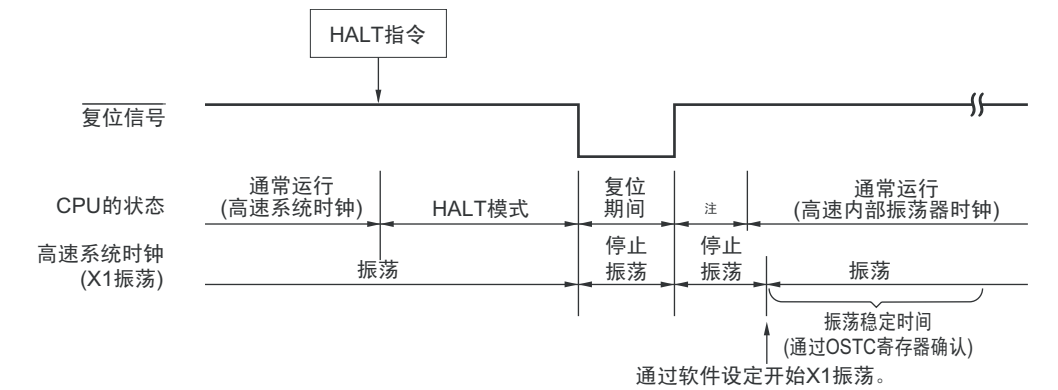


(b) 通过产生复位信号进行的解除

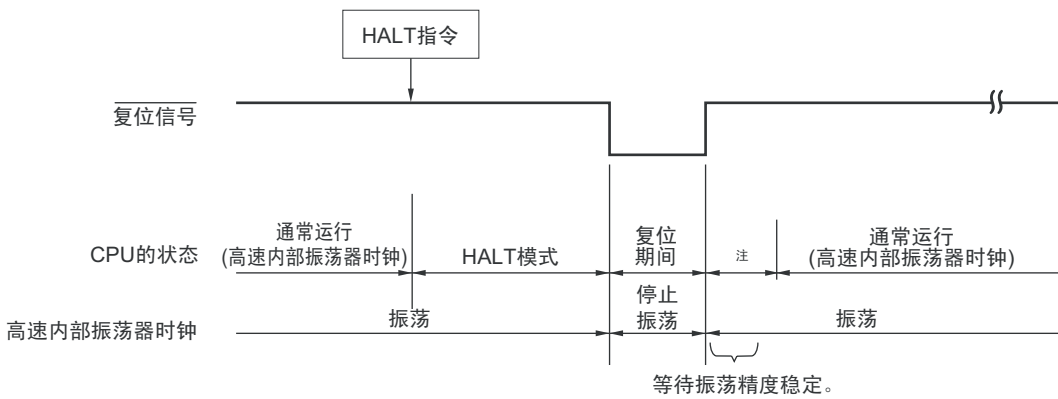
通过产生复位信号来解除 HALT 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

图 16-4 通过复位解除 HALT 模式

(1) CPU时钟为高速系统时钟的情况



(2) CPU时钟为高速内部振荡器时钟的情况



注 有关复位处理时间，请参照“第 17 章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第 18 章 上电复位电路”。

### 16.3.2 STOP 模式

#### (1) STOP 模式的设定和运行状态

通过执行 STOP 指令设定为 STOP 模式。只有在设定前的 CPU 时钟为主系统时钟的情况下才能进行设定。

注意 1. 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除 STOP 模式。因此，如果在此情况下执行 STOP 指令，就在一旦进入 STOP 模式后立即被解除。

在执行 STOP 指令并且经过 STOP 模式解除时间后返回到运行模式。但是，因为时钟不停止运行，所以此期间的工作电流和 HALT 模式中的工作电流相同。

2. 在 SNOOZE 模式中使用 CSIp、UARTq 或者 A/D 转换器时，必须在转移到 STOP 模式前设定串行待机控制寄存器 m（SSCm）和 A/D 转换器模式寄存器 2（ADM2）。详细内容请参照“9.3 控制 A/D 转换器的寄存器”和“11.3 控制串行阵列单元的寄存器”。

备注 p=00, q=0, m=0

STOP 模式中的运行状态如下所示。

表 16-2 STOP 模式中的运行状态

STOP 模式的设定 项目			在 CPU 以主系统时钟运行的过程中执行 STOP 指令的情况		
			CPU 以高速内部振荡器时钟 （f <sub>IH</sub> ）运行		CPU 以 X1 时钟 （f <sub>X</sub> ）运行
系统时钟			停止给 CPU 提供时钟。		
	主系统时钟	f <sub>IH</sub>	停止		
		f <sub>X</sub>			
		f <sub>EX</sub>			
	f <sub>IL</sub>		通过选项字节（000C0H）的 bit0（WDSTBYON）和 bit4（WDTON）进行设定。		
CPU			停止运行。		
代码闪存					
数据闪存			停止运行。		
RAM			停止运行。		
端口（锁存器）			保持设定为 STOP 模式前的状态。		
定时器阵列单元			禁止运行。		
看门狗定时器			参照“第 8 章 看门狗定时器”。		
时钟输出 / 蜂鸣器输出			停止运行。		
A/D 转换器			能进行唤醒（转移到 SNOOZE 模式）。		
D/A 转换器			能运行（保持设定为 STOP 模式前的状态）。		
串行阵列单元（SAU）			只有 CSIp 和 UARTq 才能进行唤醒（转移到 SNOOZE 模式）。 除了 CSIp 和 UARTq 以外，禁止运行。		
串行阵列单元（IICA）			能通过地址匹配进行唤醒。		
DMA 控制器			禁止运行。		
事件链接控制器（ELC）			能在可运行的功能块之间进行链接。		
上电复位功能			能运行。		
电压检测功能					
外部中断					
CRC 运算 功能	高速 CRC	通用 CRC	停止运行。		
RAM 奇偶校验错误检测功能					
RAM 保护功能					
SFR 保护功能					
非法存储器存取检测功能					

注意 1. 要在 STOP 模式中使用停止运行的外围硬件以及在解除 STOP 模式后使用时钟停止振荡的外围硬件时, 必须重新启动外围硬件。

2. 如果要在 STOP 模式中停止低速内部振荡器时钟, 就必须事先通过选项字节设定为“在 HALT/STOP 模式中看门狗定时器停止运行”(000C0H 的 bit0 (WDSTBYON) =0)。
3. 在 CPU 以高速系统时钟 (X1 振荡) 运行并且要缩短 STOP 模式解除后的振荡稳定时间时, 必须在执行 STOP 指令前暂时将 CPU 时钟切换为高速内部振荡器时钟。要在解除 STOP 模式后将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟 (X1 振荡) 时, 必须在通过振荡稳定时间计数器的状态寄存器 (OSTC) 确认振荡稳定时间后进行切换。

备注 1. 停止运行: 在转移到 STOP 模式时自动停止运行。

$f_X$  : X1 时钟

禁止运行: 在转移到 STOP 模式前停止运行。

$f_{EX}$  : 外部主系统时钟

$f_{IH}$  : 高速内部振荡器时钟

$f_{IL}$  : 低速内部振荡器时钟

2. p=00, q=0

## (2) STOP 模式的解除

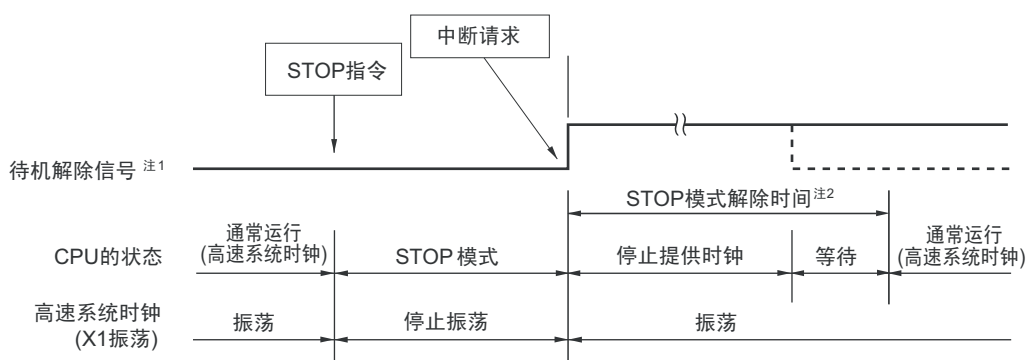
能通过以下 2 种方法解除 STOP 模式。

## (a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除 STOP 模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图 16-5 通过中断请求解除 STOP 模式 (1/2)

## (1) CPU 时钟为高速系统时钟（X1 振荡）的情况



注 1. 有关待机解除信号的详细内容，请参照图 15-1。

## 2. STOP 模式解除时间

停止提供时钟：

- 18 $\mu$ s ~ “65 $\mu$ s 和振荡稳定时间（通过 OSTS 进行设定）中较长的时间”

等待

- 进行向量中断处理的情况 : 10 ~ 11 个时钟
- 不进行向量中断处理的情况 : 4 ~ 5 个时钟

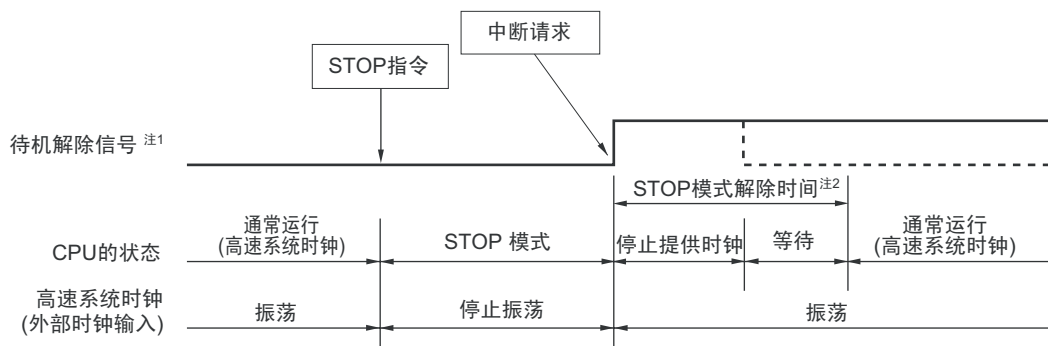
注意 在 CPU 以高速系统时钟（X1 振荡）运行并且要缩短 STOP 模式解除后的振荡稳定时间时，必须在执行 STOP 指令前暂时将 CPU 时钟切换为高速内部振荡器时钟。

备注 1. 时钟停止提供的时间因温度条件和 STOP 模式期间而变。

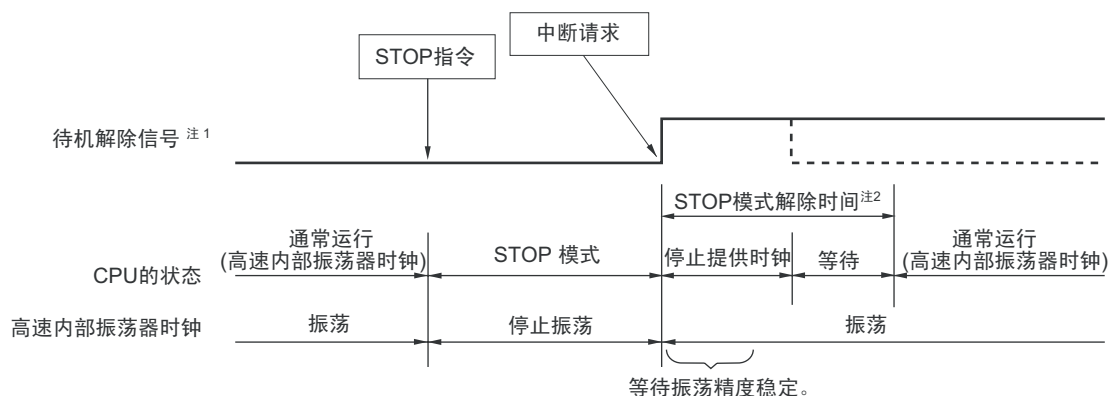
2. 虚线表示接受解除了待机模式的中断请求的情况。

图 16-5 通过中断请求解除 STOP 模式 (2/2)

## (2) CPU 时钟为高速系统时钟（外部时钟输入）的情况



## (3) CPU 时钟为高速内部振荡器时钟的情况



注 1. 有关待机解除信号的详细内容，请参照图 15-1。

## 2. STOP 模式解除时间

停止提供时钟：

- 18 $\mu$ s ~ 65 $\mu$ s

等待

- 进行向量中断处理的情况 : 7 个时钟
- 不进行向量中断处理的情况 : 1 个时钟

备注 1. 时钟停止提供的时间因温度条件和 STOP 模式期间而变。

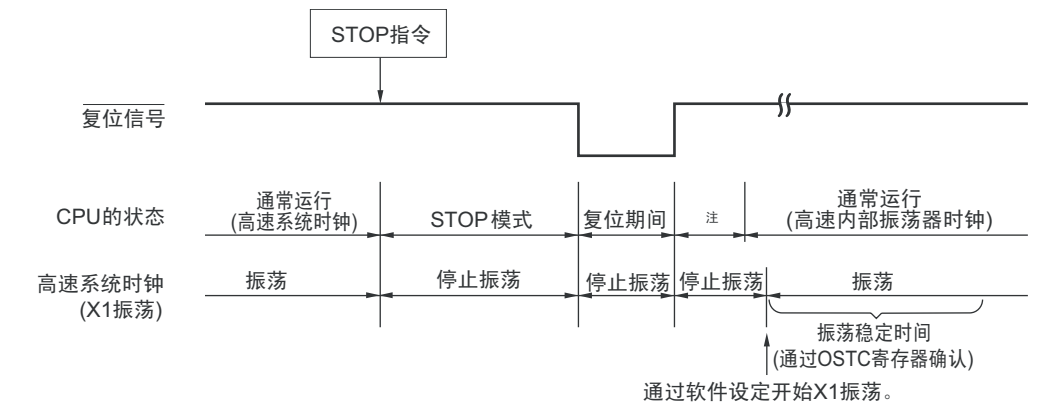
2. 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号进行的解除

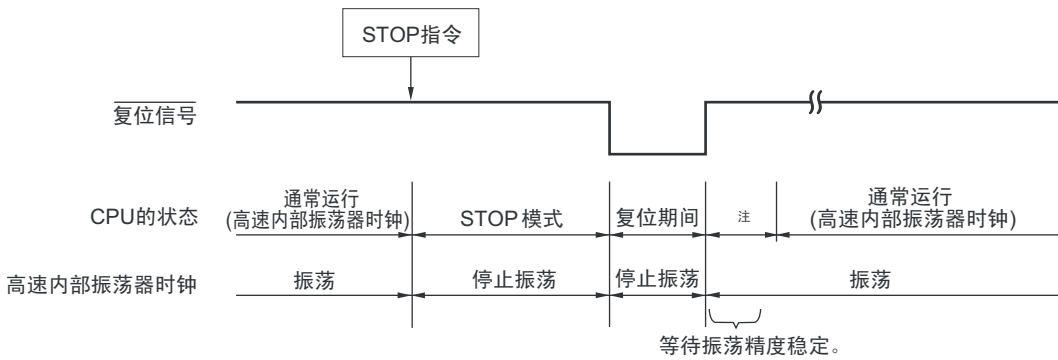
通过产生复位信号来解除 STOP 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

图 16-6 通过复位解除 STOP 模式

(1) CPU时钟为高速系统时钟的情况



(2) CPU时钟为高速内部振荡器时钟的情况



注 有关复位处理时间，请参照“第 17 章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第 18 章 上电复位电路”。

### 16.3.3 SNOOZE 模式

#### (1) SNOOZE 模式的设定和运行状态

只能给 CSIp、UARTq 或者 A/D 转换器设定 SNOOZE 模式，并且只有在设定前的 CPU 时钟为高速内部振荡器时钟的情况下才能进行设定。

要在 SNOOZE 模式中使用 CSIp 或者 UARTq 时，必须在即将要转移到 STOP 模式前将串行待机控制寄存器 m (SSCm) 的 SWCm 位置“1”。详细内容请参照“11.3 控制串行阵列单元的寄存器”。

要在 SNOOZE 模式中使用 A/D 转换器时，必须在即将要转移到 STOP 模式前将 A/D 转换器模式寄存器 2 (ADM2) 的 AWC 位置“1”。详细内容请参照“9.3 控制 A/D 转换器的寄存器”。

备注 p=00, q=0, m=0

在进行 SNOOZE 模式的转移时，只在以下的时间内变为等待状态。

#### STOP→SNOOZE

- 18.96μs ~ 28.95μs

#### SNOOZE→通常运行

- 进行向量中断处理的情况：
  - HS (高速主) 模式 : 6.79 ~ 12.4μs + 7 个时钟
  - LS (低速主) 模式 : 2.58 ~ 7.8μs + 7 个时钟
- 不进行向量中断处理的情况：
  - HS (高速主) 模式 : 6.79 ~ 12.4μs + 1 个时钟
  - LS (低速主) 模式 : 2.58 ~ 7.8μs + 1 个时钟

SNOOZE 模式中的运行状态如下所示。

表 16-3 SNOOZE 模式中的运行状态

SNOOZE 模式的设定			在 STOP 模式中输入 CSIp、UARTq 的数据接收信号和 A/D 转换器的定时器触发信号的情况
项目			CPU 以高速内部振荡器时钟 ( $f_{IH}$ ) 运行
系统时钟	主系统 时钟	$f_{IH}$	停止给 CPU 提供时钟。
		$f_X$	开始运行。
		$f_{EX}$	停止
	$f_{IL}$		通过选项字节 (000C0H) 的 bit0 (WDSTBYON) 和 bit4 (WDTON) 进行设定。
CPU			停止运行。
代码闪存			
数据闪存			
RAM			
端口 (锁存器)			保持 STOP 模式中的状态。
定时器阵列单元			禁止运行。
看门狗定时器			参照“第 8 章 看门狗定时器”。
时钟输出 / 蜂鸣器输出			停止运行。
A/D 转换器			能运行。
D/A 转换器			能运行 (保持转移到 SNOOZE 模式前的状态)。
串行阵列单元 (SAU)			只有 CSIp 和 UARTq 才能运行。 除了 CSIp 和 UARTq 以外, 禁止运行。
串行阵列单元 (IICA)			禁止运行。
DMA 控制器			
事件链接控制器 (ELC)			能在可运行的功能块之间进行链接。
上电复位功能			能运行。
电压检测功能			
外部中断			
CRC 运算功能			停止运行。
RAM 奇偶校验错误检测功能			
RAM 保护功能			
SFR 保护功能			
非法存储器存取检测功能			

备注 1. 停止运行: 在转移到 SNOOZE 模式时自动停止运行。  
禁止运行: 在转移到 SNOOZE 模式前停止运行。

$f_X$  : X1 时钟  
 $f_{EX}$  : 外部主系统时钟  
 $f_{IH}$  : 高速内部振荡器时钟  
 $f_{IL}$  : 低速内部振荡器时钟

2. p=00, q=0



## 第 17 章 复位功能

以下 7 种方法产生复位信号。

- (1) 通过  $\overline{\text{RESET}}$  引脚输入外部复位。
- (2) 通过看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位 (POR) 电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路 (LVD) 的电源电压和检测电压的比较产生内部复位。
- (5) 因执行非法指令而产生内部复位注。
- (6) 因 RAM 奇偶校验错误而产生内部复位。
- (7) 因存取非法存储器而产生内部复位。

内部复位和外部复位相同，在产生复位信号后，从写在地址 0000H 和 0001H 中的地址开始执行程序。

当给  $\overline{\text{RESET}}$  引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者执行非法指令注，或者发生 RAM 奇偶校验错误，或者存取非法存储器时，产生复位并且各硬件变为如表 17-1 所示的状态。

当给  $\overline{\text{RESET}}$  引脚输入低电平时，产生复位。然后，如果给  $\overline{\text{RESET}}$  引脚输入高电平，就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。对于因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序（参照图 17-2 和图 17-3）。对于由检测 POR 电路和 LVD 电路的电压产生的复位，如果在复位后满足  $V_{DD} \geq V_{POR}$  或者  $V_{DD} \geq V_{LVD}$ ，就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序（参照“第 18 章 上电复位电路”和“第 19 章 电压检测电路”）。

注 当执行指令码 FFH 时，发生非法指令错误。

在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

注意 1. 在进行外部复位时，必须至少给  $\overline{\text{RESET}}$  引脚输入 10 $\mu$ s 的低电平。

如果在电源电压上升时进行外部复位，就必须在工作电压范围内至少保持 10 $\mu$ s 的低电平。工作电压范围取决于用户选项字节 (000C2H) 的设定，如下所示：

HS (高速主) 模式 :  $V_{DD}=2.7 \sim 3.6V@1MHz \sim 24MHz$

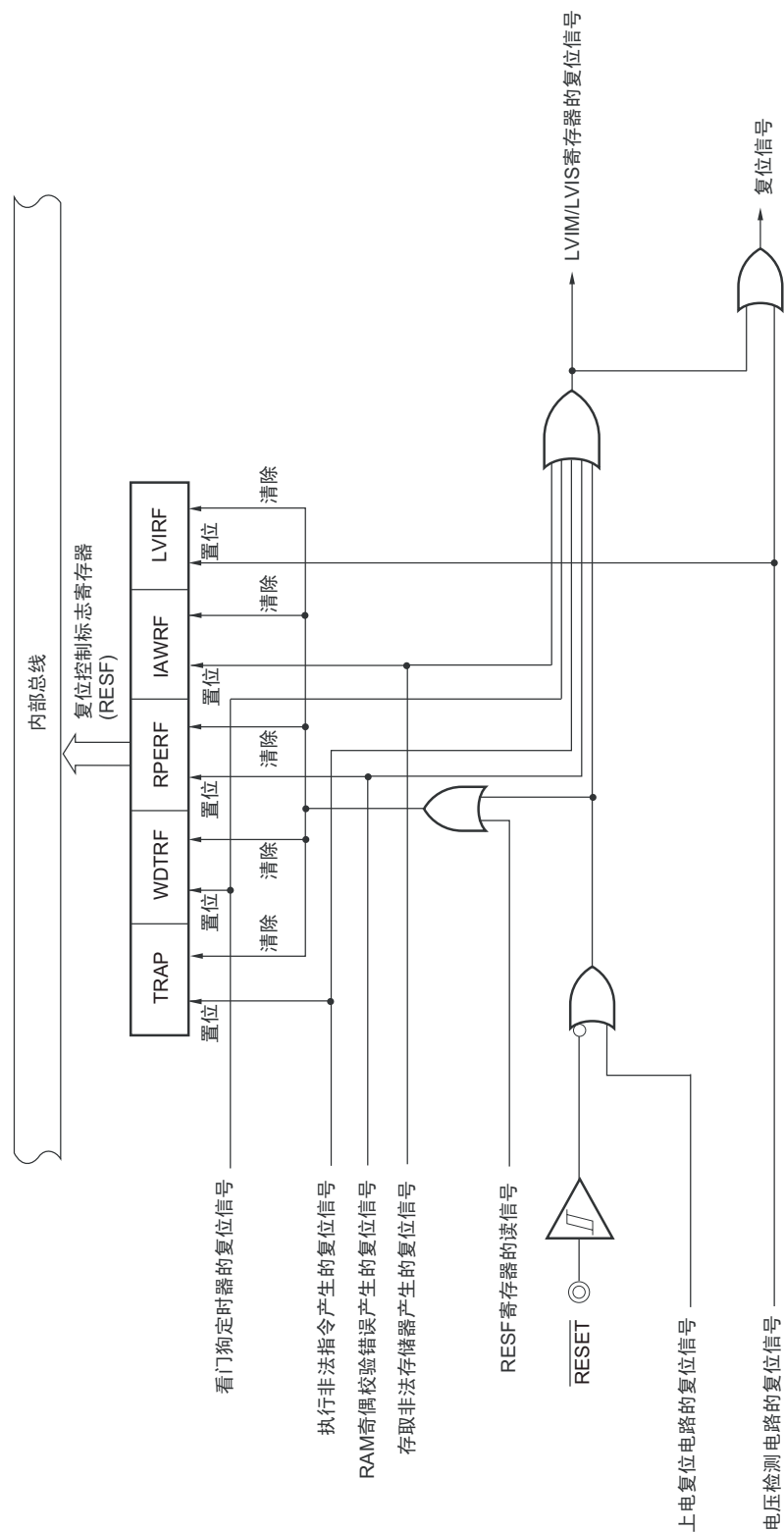
LS (低速主) 模式 :  $V_{DD}=2.7 \sim 3.6V@1MHz \sim 8MHz$

2. 在复位信号发生期间，停止 X1 时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟的输入无效。
3. 如果发生复位，就对各 SFR 和 2nd SFR 进行初始化，因此端口引脚变为以下状态：
  - P40: 在外部复位或者 POR 复位的期间为高阻抗。在其他复位期间以及接受复位后为高电平（连接内部上拉电阻）。
  - P40 以外的端口：在复位期间以及接受复位后为高阻抗。

备注  $V_{POR}$ : POR 电源电压上升检测电压

$V_{LVD}$ : LVD 检测电压

图 17-1 复位功能的框图



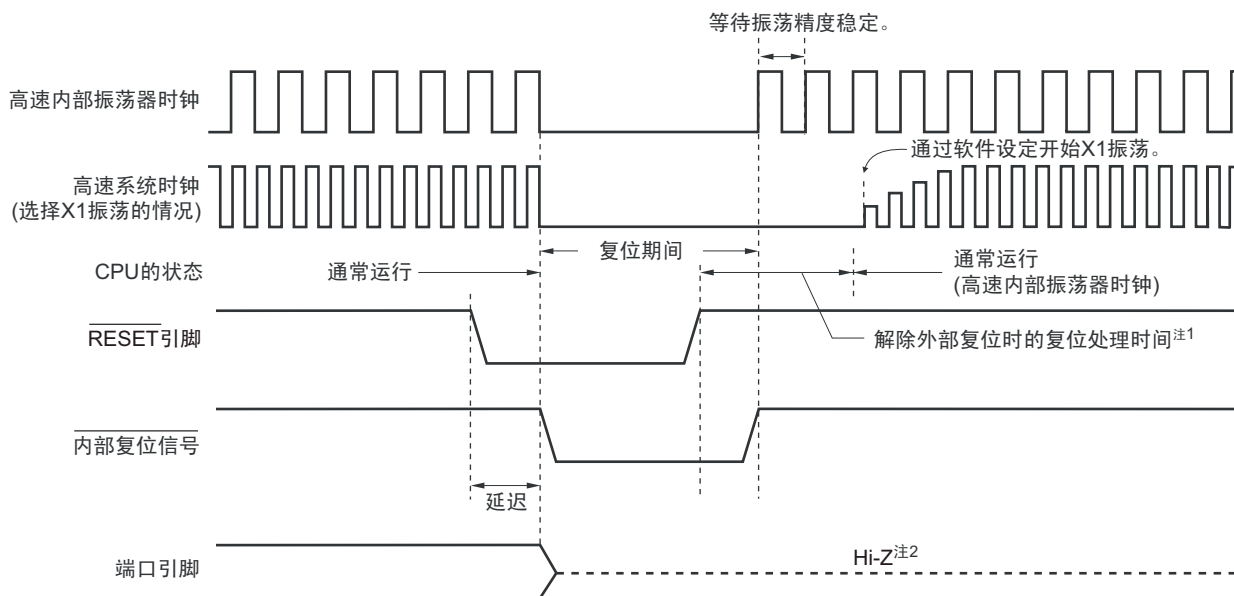
注意 LVD 电路的内部复位不会对 LVD 电路进行复位。

- 备注 1. LVIM: 电压检测寄存器  
2. LVIS: 电压检测电平寄存器

## 17.1 复位时序

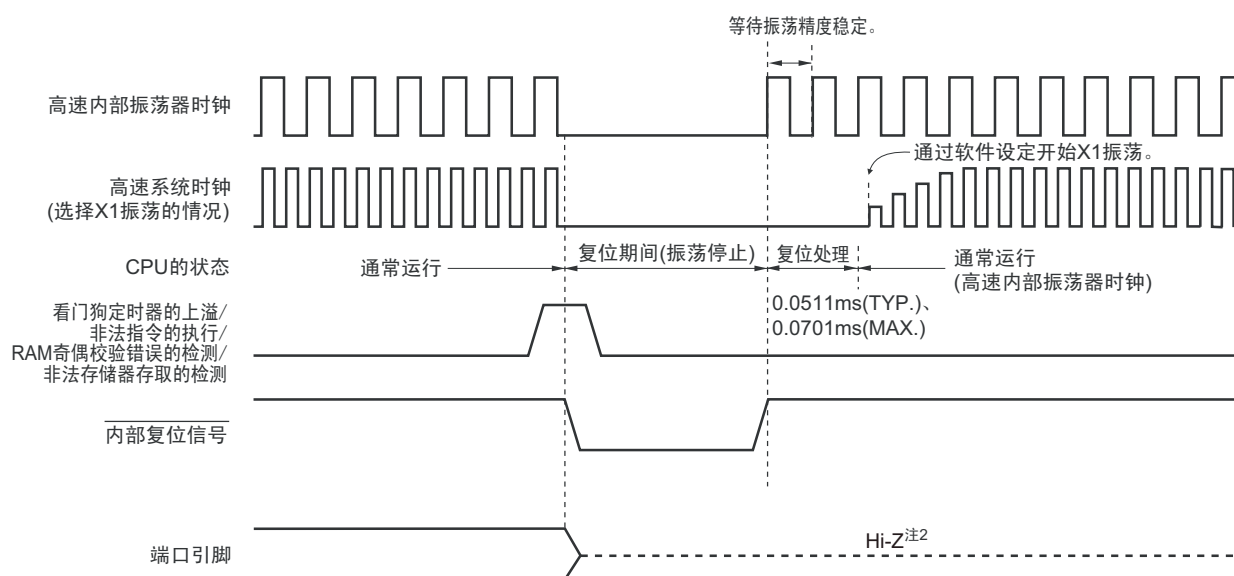
当给  $\overline{\text{RESET}}$  引脚输入低电平时，产生复位。然后，如果给  $\overline{\text{RESET}}$  引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 17-2  $\overline{\text{RESET}}$  输入的复位时序



对于因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 17-3 因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位时序



注 1. 解除外部复位时的复位处理时间:

- 解除 POR 后的第 1 次 : 0.672ms(TYP.)、0.832ms(MAX.) (使用 LVD 的情况)  
0.399ms(TYP.)、0.519ms(MAX.) (不使用 LVD 的情况)  
解除 POR 后的第 2 次以后 : 0.531ms(TYP.)、0.675ms(MAX.) (使用 LVD 的情况)  
0.259ms(TYP.)、0.362ms(MAX.) (不使用 LVD 的情况)

当电源电压上升时, 在解除外部复位时的复位处理时间之前需要电压稳定等待时间 (0.99ms(TYP.)、2.30ms(MAX.))。

2. 端口引脚 P40 变为以下状态:

- 在外部复位或者 POR 复位的期间为高阻抗。
- 在其他复位期间以及接受复位后为高电平 (连接内部上拉电阻)。

注意 看门狗定时器的内部复位也会对看门狗定时器进行复位。

备注 有关上电复位电路和电压检测电路的复位时序, 请参照“第 18 章 上电复位电路”和“第 19 章 电压检测电路”。

表 17-1 复位期间的运行状态

项目			复位期间
系统时钟			停止给 CPU 提供时钟。
	主系统时钟	f <sub>IH</sub>	停止运行。
		f <sub>X</sub>	停止运行（X1 引脚和 X2 引脚处于输入端口模式）。
		f <sub>EX</sub>	时钟输入无效（引脚处于输入端口模式）。
	f <sub>IL</sub>		停止运行。
CPU			
代码闪存			停止运行。
数据闪存			停止运行。
RAM			停止运行。
端口（锁存器）			P40 为高阻抗（外部复位和 POR 复位）或者被上拉（外部复位和 POR 复位以外的复位）。除 P40 以外，为高阻抗。
定时器阵列单元			停止运行。
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）			
串行接口（IICA）			
DMA 控制器			
事件链接控制器（ELC）			
PWM 选项单元			
上电复位功能			能进行检测运行。
电压检测功能			在 LVD 复位时，能运行。在其他复位时，停止运行。
外部中断			停止运行。
CRC 运算功能	高速 CRC		
	通用 CRC		
RAM 奇偶校验错误检测功能			
RAM 保护功能			
SFR 保护功能			
非法存储器存取检测功能			

备注  $f_{IH}$  : 高速内部振荡器时钟  
 $f_X$  : X1 振荡时钟  
 $f_{EX}$  : 外部主系统时钟  
 $f_{IL}$  : 低速内部振荡器时钟

表 17-2 接受复位后的各硬件状态 (1/3)

硬件		接受复位后的状态注 1
程序计数器 (PC)		设定复位向量表 (0000H、0001H) 的内容。
堆栈指针 (SP)		不定
程序状态字 (PSW)		06H
RAM	数据存储器	不定
	通用寄存器	不定
处理器模式控制寄存器 (PMC)		00H
端口寄存器 (P1 ~ P4、P6、P12、P13) (输出锁存器)		00H
端口模式寄存器 (PM1 ~ PM4、PM6)		FFH
端口模式控制寄存器 1 (PMC1)		FFH
上拉电阻选择寄存器 (PU1、PU3、PU4)		00H (PU4 为 "01H")
时钟运行模式控制寄存器 (CMC)		00H
时钟运行状态控制寄存器 (CSC)		C0H
系统时钟控制寄存器 (CKC)		00H
振荡稳定时间计数器的状态寄存器 (OSTC)		00H
振荡稳定时间选择寄存器 (OSTS)		07H
噪声滤波器允许寄存器 0、1 (NFEN0、NFEN1)		00H
外围允许寄存器 0 (PER0)		00H
外围允许寄存器 1 (PER1)		00H
高速内部振荡器的频率选择寄存器 (HOCODIV)		不定
高速内部振荡器的微调寄存器 (HIOTRM)		不定注 2
定时器阵列单元	定时器数据寄存器 00 ~ 03 (TDR00 ~ TDR03)	0000H
	定时器模式寄存器 00 ~ 03 (TMR00 ~ TMR03)	0000H
	定时器状态寄存器 00 ~ 03 (TSR00 ~ TSR03)	0000H
	定时器输入选择寄存器 0 (TIS0)	00H
	定时器计数寄存器 00 ~ 03 (TCR00 ~ TCR03)	FFFFH
	定时器通道允许状态寄存器 0 (TE0)	0000H
	定时器通道开始寄存器 0 (TS0)	0000H
	定时器通道停止寄存器 0 (TT0)	0000H
	定时器时钟选择寄存器 0 (TPS0)	0000H
	定时器输出寄存器 0 (TO0)	0000H
	定时器输出允许寄存器 0 (TOE0)	0000H
	定时器输出电平寄存器 0 (TOL0)	0000H
	定时器输出模式寄存器 0 (TOM0)	0000H

注 1. 在复位信号发生期间和振荡稳定等待期间的各硬件状态中，只有 PC 的内容不定，而其他的硬件状态保持复位后的状态。

2. 复位值是产品发货时的调整值。

表 17-2 接受复位后的各硬件状态 (2/3)

硬件		接受复位后的状态 <sup>注 1</sup>
时钟输出 / 蜂鸣器输出控制电路	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	12 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 0 ～ 2（ADM0 ～ ADM2）	00H
	转换结果比较上限值设定寄存器（ADUL）	FFH
	转换结果比较下限值设定寄存器（ADLL）	00H
	A/D 测试寄存器（ADTES）	00H
	模拟输入通道指定寄存器（ADS）	00H
	A/D 端口配置寄存器（ADPC）	00H
D/A 转换器	D/A 转换值设定寄存器 0、1（DACS0、DACS1）	00H
	D/A 转换器的模式寄存器（DAM）	00H
串行阵列单元（SAU）	串行数据寄存器 00、01（SDR00、SDR01）	0000H
	串行状态寄存器 00、01（SSR00、SSR01）	0000H
	串行标志清除触发寄存器 00、01（SIR00、SIR01）	0000H
	串行模式寄存器 00、01（SMR00、SMR01）	0020H
	串行通信运行设定寄存器 00、01（SCR00、SCR01）	0087H
	串行通道允许状态寄存器 0（SE0）	0000H
	串行通道开始寄存器 0（SS0）	0000H
	串行通道停止寄存器 0（ST0）	0000H
	串行时钟选择寄存器 0（SPS0）	0000H
	串行输出寄存器 0（SO0）	0303H
	串行输出允许寄存器 0（SOE0）	0000H
	串行输出电平寄存器 0（SOL0）	0000H
	串行待机控制寄存器 0（SSC0）	0000H
	输入切换控制寄存器（ISC）	00H
串行接口 IICA	IICA 移位寄存器 0、1（IICA0、IICA1）	00H
	IICA 状态寄存器 0、1（IICS0、IICS1）	00H
	IICA 标志寄存器 0、1（IICF0、IICF1）	00H
	IICA 控制寄存器 00、10（IICCTL00、IICCTL10）	00H
	IICA 控制寄存器 01、11（IICCTL01、IICCTL11）	00H
	IICA 低电平宽度设定寄存器 0、1（IICWL0、IICWL1）	FFH
	IICA 高电平宽度设定寄存器 0、1（IICWH0、IICWH1）	FFH
	从属地址寄存器 0、1（SVA0、SVA1）	00H

注 1. 在复位信号发生期间和振荡稳定等待期间的各硬件状态中，只有 PC 的内容不定，而其他的硬件状态保持复位后的状态。

2. WDTE 的复位值取决于选项字节的设定。

表 17-2 接受复位后的各硬件状态 (3/3)

硬件		接受复位后的状态 <sup>注 1</sup>
复位功能	复位控制标志寄存器 (RESF)	不定 <sup>注 2</sup>
电压检测电路 (LVD)	电压检测寄存器 (LVIM)	00H <sup>注 2</sup>
	电压检测电平寄存器 (LVIS)	00H/01H/81H <sup>注 2、3</sup>
DMA 控制器	SFR 地址寄存器 0、1 (DSA0、DSA1)	00H
	RAM 地址寄存器 0、1 (DRA0、DRA1)	0000H
	字节计数寄存器 0、1 (DBC0、DBC1)	0000H
	模式控制寄存器 0、1 (DMC0、DMC1)	00H
	运行控制寄存器 0、1 (DRC0、DRC1)	00H
ELC	事件输出选择寄存器 00 ~ 25 (ELSELR00 ~ ELSELR09)	00H
中断	请求标志寄存器 0L、0H、1L (IF0L、IF0H、IF1L)	00H
	屏蔽标志寄存器 0L、0H、1L (MK0L、MK0H、MK1L)	FFH
	优先级指定标志寄存器 00L、00H、01L、10L、10H、11L (PR00L、PR00H、PR01L、PR10L、PR10H、PR11L)	FFH
	外部中断上升沿允许寄存器 0 (EGP0)	00H
	外部中断下降沿允许寄存器 0 (EGN0)	00H
安全功能	闪存 CRC 控制寄存器 (CRC0CTL)	00H
	闪存 CRC 运算结果寄存器 (PGCRCL)	0000H
	CRC 输入寄存器 (CRCIN)	00H
	CRC 数据寄存器 (CRCD)	0000H
	非法存储器存取检测控制寄存器 (IAWCTL)	00H
	RAM 奇偶校验错误控制寄存器 (RPECTL)	00H
闪存	数据闪存控制寄存器 (DFLCTL)	00H
十进制校正 (BCD) 电路	BCD 校正结果寄存器 (BCDADJ)	不定

注 1. 在复位信号发生期间和振荡稳定等待期间的各硬件状态中，只有 PC 的内容不定，而其他的硬件状态保持复位后的状态。

2. 因复位源而发生的变化如下：

<div>复位源</div> <div>寄存器</div>		RESET 输入	POR 产生 的复位	执行非法指 令产生的 复位	WDT 产生的 复位	RAM 奇偶 校验错误产 生的复位	存取非法存 储器产生的 复位	LVD 产生的 复位
RESF	TRAP	清“0”		置“1”	保持			保持
	WDTRF			保持				
	RPERF			保持		置“1”	保持	
	IAWRF			保持			置“1”	
	LVIRF			保持				
LVIM	LVISEN	清“0”						保持
	LVIOMSK	保持						
	LVIF							
LVIS		清除（00H/01H/81H）						

3. 在发生 LVD 以外的复位时，变化如下：

- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
- 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
- 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H



17.2 确认复位源的寄存器

17.2.1 复位控制标志寄存器（RESF）

R7F0C010 存在多种内部复位发生源。复位控制标志寄存器（RESF）保存发生复位请求的复位源。  
能通过 8 位存储器操作指令读 RESF 寄存器。

通过 RESET 的输入、上电复位（POR）电路的复位和 RESF 寄存器的读取，清除 TRAP、WDTRF、RPERF、IAWRF、LVIRF 标志。

图 17-4 复位控制标志寄存器（RESF）的格式

地址: FFFA8H

复位后: 00 <sup>注 1</sup>

R

符号	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	执行非法指令产生的内部复位请求 <sup>注 2</sup>
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

WDTRF	看门狗定时器（WDT）产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

RPERF	RAM 奇偶校验错误产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

IAWRF	存取非法存储器产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

LVIRF	电压检测电路（LVD）产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

注 1. 因复位源而不同。请参照表 17-3。

2. 当执行指令码 FFH 时，发生非法指令错误。  
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

注意 1. 不能通过位存储器操作指令读数据。

2. 在允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区+10 字节”的区域进行初始化。  
通过产生复位，进入允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的状态。详细内容请参照“20.5 RAM 奇偶校验错误检测功能”。

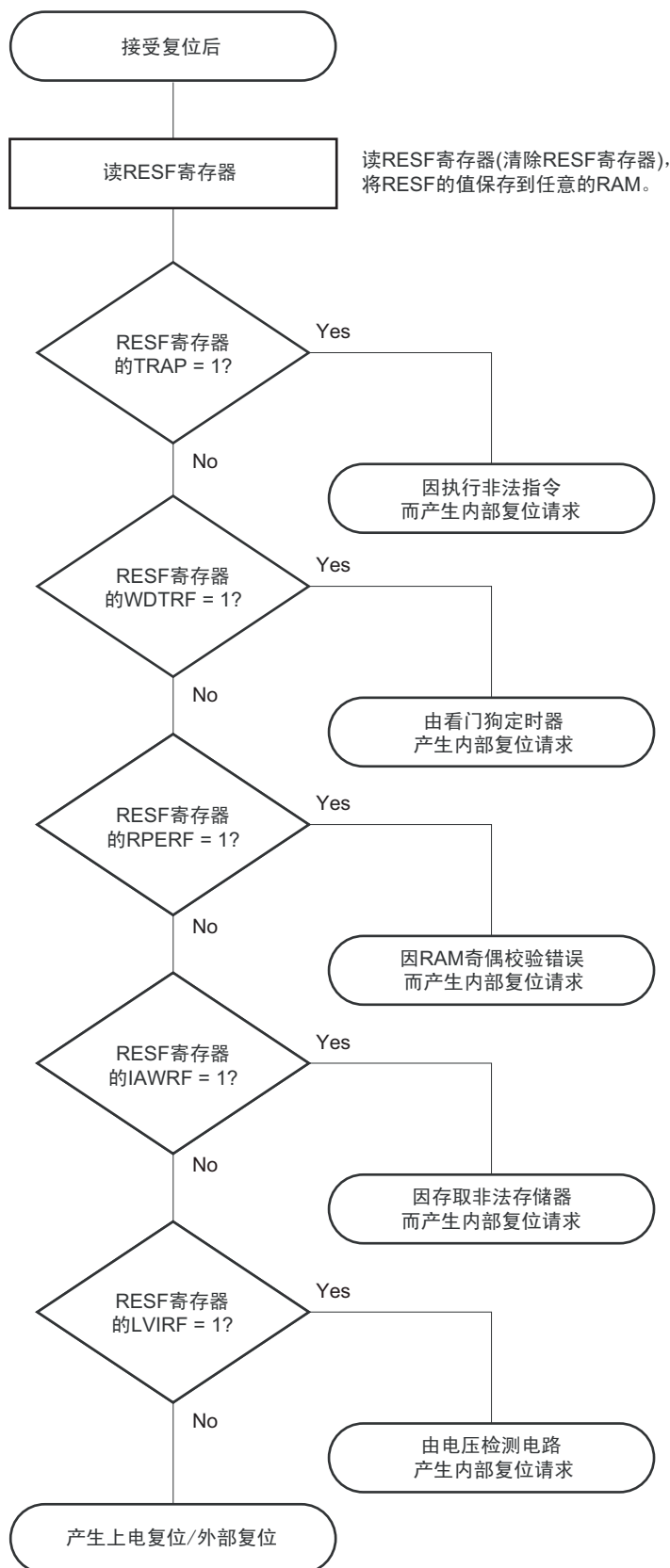
发生复位请求时的 RESF 寄存器状态如表 17-3 所示。

表 17-3 发生复位请求时的 RESF 寄存器状态

复位源 标志	$\overline{\text{RESET}}$ 输入	POR 产生 的复位	执行非法指 令产生的 复位	WDT 产生的 复位	RAM 奇偶 校验错误产 生的复位	存取非法存 储器产生的 复位	LVD 产生的 复位
TRAP	清“0”	清“0”	置“1”	保持	保持	保持	保持
WDTRF			保持	置“1”			
RPERF				保持	置“1”		
IAWRF					保持	置“1”	
LVIRF						保持	置“1”

复位源的确认步骤如图 17-5 所示。

图 17-5 复位源的确认步骤



## 第 18 章 上电复位电路

### 18.1 上电复位电路的功能

上电复位电路（POR）有以下功能。

- 在接通电源时产生内部复位信号。  
如果电源电压（ $V_{DD}$ ）超过检测电压（ $V_{POR}$ ），就解除复位。但是，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测功能或者外部复位引脚保持复位状态。
- 将电源电压（ $V_{DD}$ ）和检测电压（ $V_{PDR}$ ）进行比较。当  $V_{DD} < V_{PDR}$  时，产生内部复位信号。但是，当工作电压下降时，必须在工作电压低于“27.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移、电压检测功能或者外部复位引脚，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

**注意** 当 POR 电路产生内部复位信号时，将复位控制标志寄存器（RESF）的 TRAP、WDTRF、RPERF、IAWRF、LVIRF 标志清“00H”。

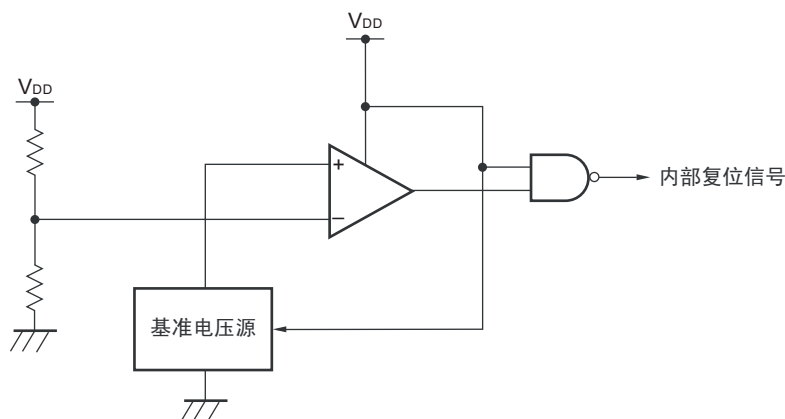
**备注 1.** 本产品内置多个产生内部复位信号的硬件。当由看门狗定时器（WDT）、电压检测（LVD）电路、非法指令的执行、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在 RESF 寄存器；当由 WDT、LVD、非法指令的执行、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清“00H”而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

- $V_{POR}$ : POR 电源电压上升检测电压  
 $V_{PDR}$ : POR 电源电压下降检测电压  
详细内容请参照“27.6.4 POR 电路特性”。

### 18.2 上电复位电路的结构

上电复位电路的框图如图 18-1 所示。

图 18-1 上电复位电路的框图

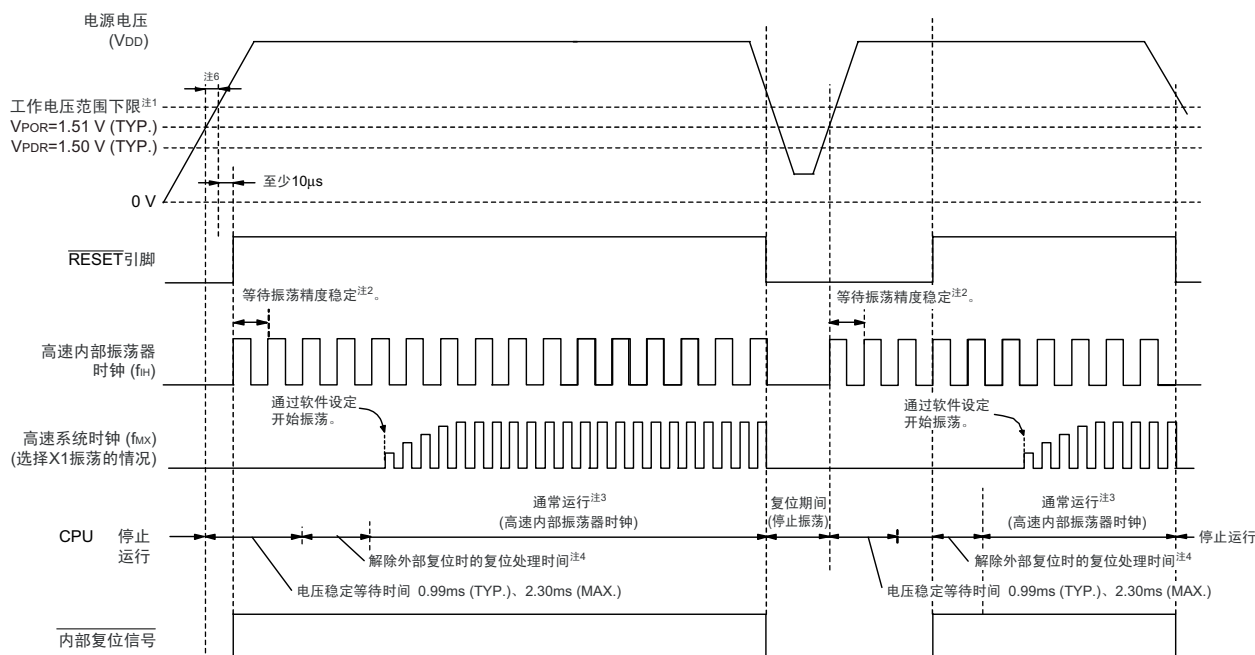


### 18.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (1/3)

#### (1) 上电复位电路 (LVD 为 OFF 时) (选项字节 000C1H 的 VPOC2=1)



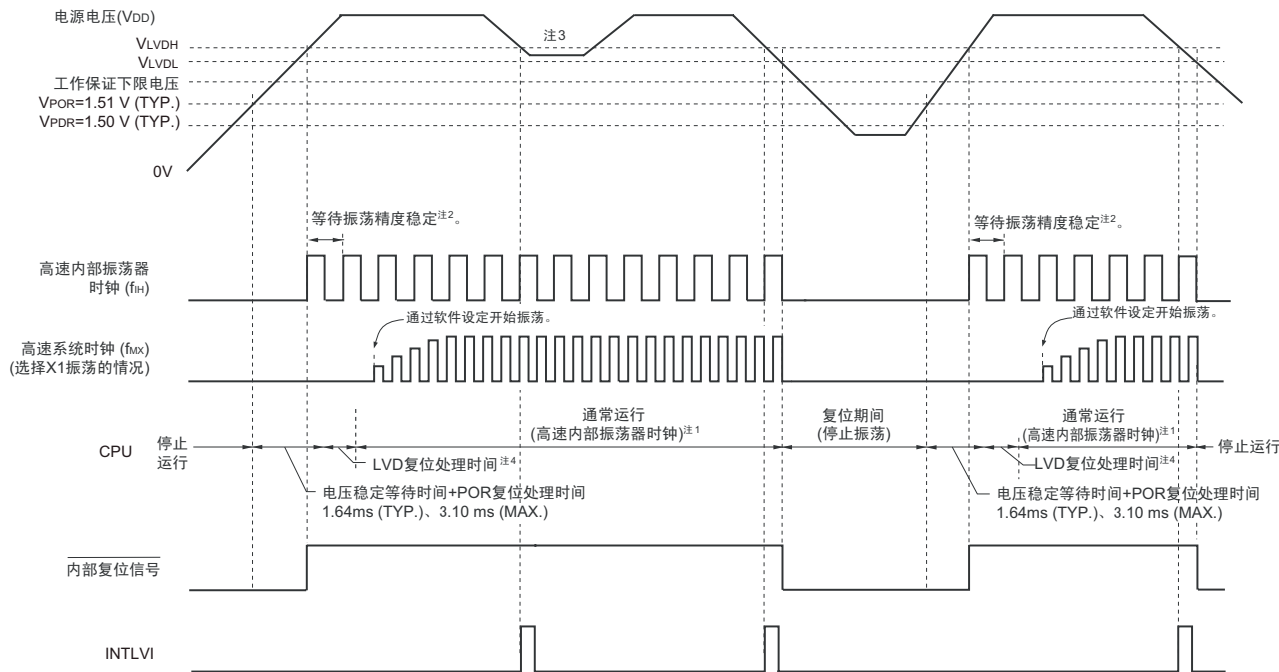
- 注 1. 内部电压稳定等待等的复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器 (OSTC) 确认振荡稳定时间后进行切换。
3. 到开始通常运行为止的时间除了达到  $V_{POR}$  (1.51V(TYP.)) 后的“电压稳定等待时间”以外，在将  $\overline{RESET}$  信号置为高电平 (“1”) 后还需要以下的“解除外部复位时的复位处理时间 (解除 POR 后的第 1 次)”。解除外部复位时的复位处理时间如下所示：
- 解除 POR 后的第 1 次： 0.672ms(TYP.)、0.832ms(MAX.) (使用 LVD 的情况)  
0.399ms(TYP.)、0.519ms(MAX.) (不使用 LVD 的情况)
4. 解除 POR 后第 2 次以后的解除外部复位时的复位处理时间如下所示：
- 解除 POR 后的第 2 次以后： 0.531ms(TYP.)、0.675ms(MAX.) (使用 LVD 的情况)  
0.259ms(TYP.)、0.362ms(MAX.) (不使用 LVD 的情况)
5. 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过 STOP 模式的转移、电压检测电路或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注意 在 LVD 为 OFF 时，必须使用  $\overline{RESET}$  引脚的外部复位。详细内容请参照“第 19 章 电压检测电路”。

备注  $V_{POR}$ : POR 电源电压上升检测电压  
 $V_{PDR}$ : POR 电源电压下降检测电压

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (2/3)

(2) LVD 为中断 & 复位模式的情况（选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、0）

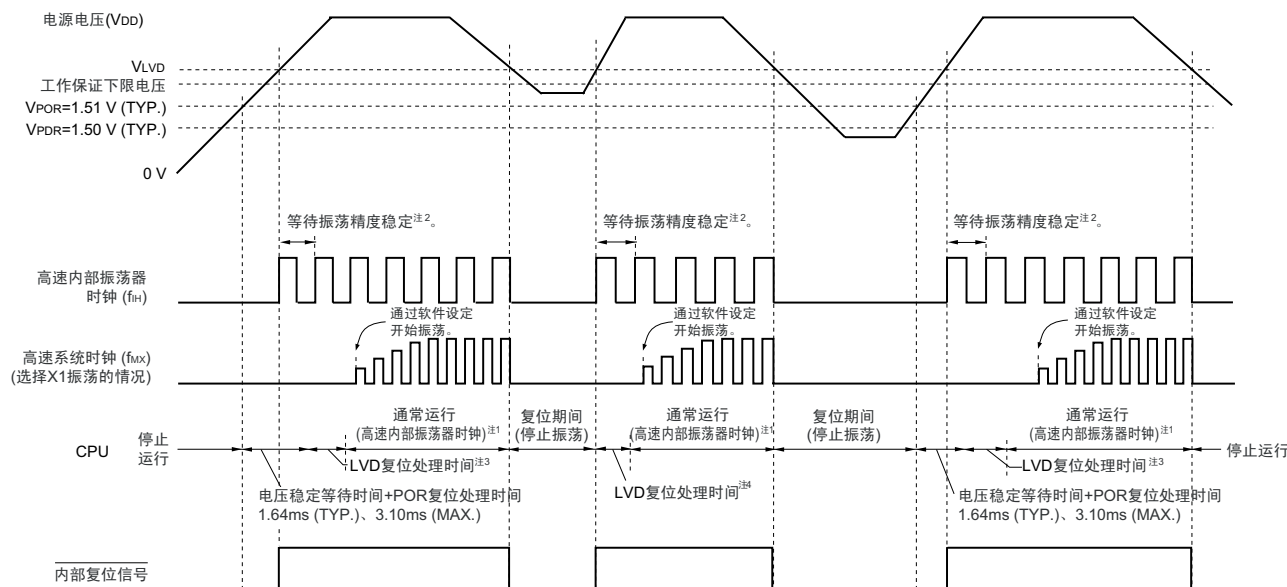


- 注 1. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换。
2. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
3. 在产生第 1 个中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的 LVIL 位和 LVIMD 位置“1”。因此，当可能出现工作电压在不低于电压检测电压（ $V_{LVDL}$ ）的情况下恢复到 2.7V 或者更高时，必须在产生 INTLVI 后进行需要的压栈处理并且通过软件进行初始设定（参照“图 19-8 中断 & 复位模式的初始设定步骤”）。
4. 到开始通常运行为止的时间除了达到  $V_{POR}$ （1.51V(TYP.)) 后的“电压稳定等待时间 + POR 复位处理时间”以外，在达到 LVD 检测电平（ $V_{LVDH}$ ）后还需要以下的“LVD 复位处理时间”。
- LVD 复位处理时间：0ms ~ 0.0701ms(MAX.)

备注  $V_{LVDH}$ 、 $V_{LVDL}$  : LVD 检测电压  
 $V_{POR}$  : POR 电源电压上升检测电压  
 $V_{PDR}$  : POR 电源电压下降检测电压

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (3/3)

## (3) LVD 复位模式的情况 (选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、1)



注 1. 能将 CPU 时钟从高速内部振荡器时钟切换为高速系统时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器 (OSTC) 确认振荡稳定时间后进行切换。

2. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。

3. 到开始通常运行为止的时间除了达到  $V_{POR}$  (1.51V(TYP.)) 后的“电压稳定等待时间+POR 复位处理时间”以外，在达到 LVD 检测电平 ( $V_{LVD}$ ) 后还需要以下的“LVD 复位处理时间”。

LVD 复位处理时间: 0ms ~ 0.0701ms(MAX.)

4. 在电源电压下降时，如果只在发生电压检测电路 (LVD) 的内部复位后恢复电源电压，就在达到 LVD 检测电平 ( $V_{LVD}$ ) 后需要以下的“LVD 复位处理时间”。

LVD 复位处理时间: 0.0511ms(TYP.)、0.0701ms(MAX.)

备注 1.  $V_{LVD}$  : LVD 检测电压

$V_{POR}$  : POR 电源电压上升检测电压

$V_{PDR}$  : POR 电源电压下降检测电压

2. 当选择 LVD 中断模式 (选项字节 000C1H 的 LVIMD1、LVIMD0=0、1) 时，从接通电源到开始通常运行的时间和“图 18-2 (3) LVD 复位模式的情况”的“注 3”的时间相同。

18.4 上电复位电路的注意事项

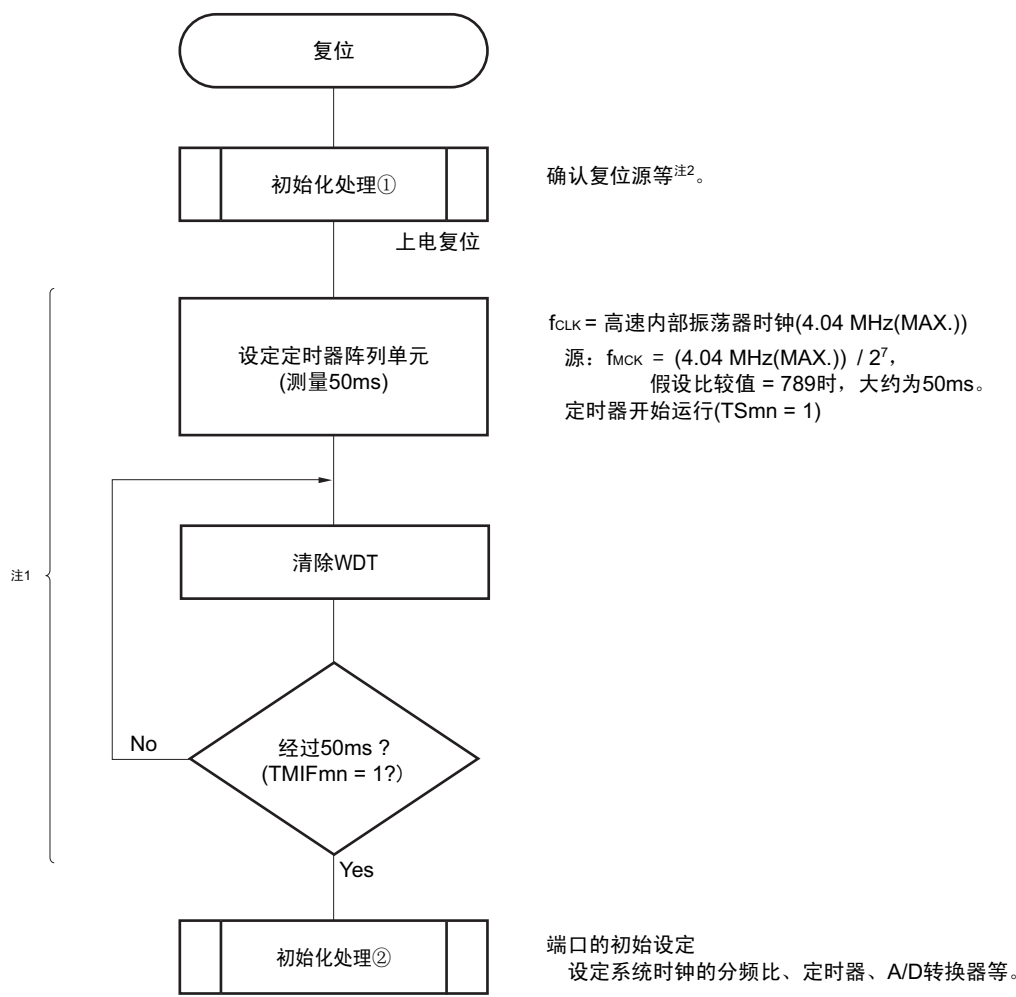
对于电源电压（V<sub>DD</sub>）在 POR 检测电压（V<sub>POR</sub>、V<sub>PDR</sub>）附近发生一定时间波动的系统，有可能重复进入复位状态和复位解除状态。能通过以下的处理，任意设定解除复位到单片机开始运行的时间。

<处理>

在解除复位后，必须通过使用定时器的软件计数器，在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 18-3 解除复位后的软件处理例子 (1/2)

- POR 检测电压附近的电源电压波动不超过50ms的情况



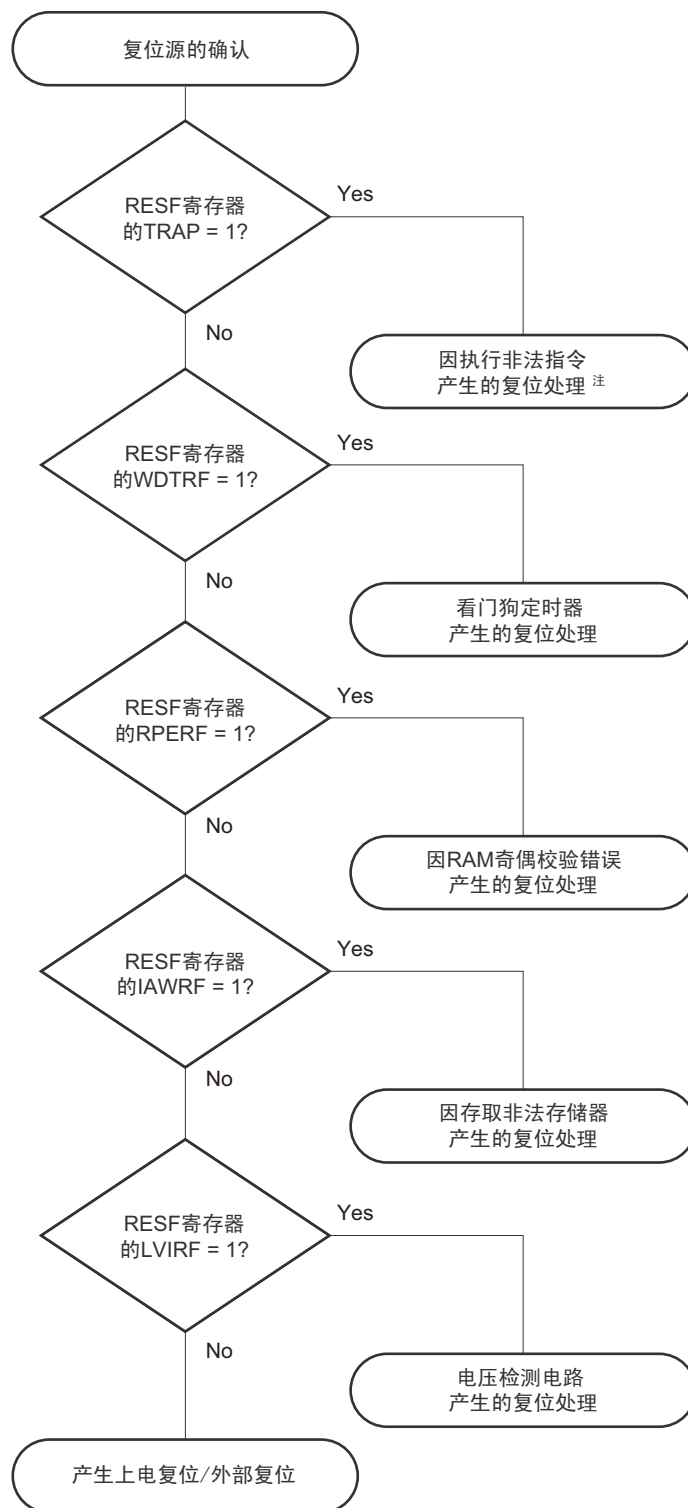
注 1. 如果在此期间再次发生复位，就不转移到初始化处理②。  
2. 流程图如下页所示。

备注 m=0 n=0 ~ 3



图 18-3 解除复位后的软件处理例子 (2/2)

- 复位源的确认



注 当执行指令码 FFH 时，发生非法指令错误。  
 在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

## 第 19 章 电压检测电路

### 19.1 电压检测电路的功能

电压检测（LVD）电路有以下功能。

- 将电源电压（ $V_{DD}$ ）和检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ ）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ）能通过选项字节选择4种检测电平（参照“第22章 选项字节”）。
- 也能在STOP模式中运行。
- 当电源电压上升时，必须在电源电压达到“27.4 AC特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，通过STOP模式的转移、电压检测电路或者外部复位，置为复位状态。工作电压范围取决于用户选项字节（000C2H）的设定。
- 能通过选项字节选择以下3种运行模式。

#### (a) 中断 & 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）

对于选项字节 000C1H 选择的 2 个检测电压，高电压检测电平（ $V_{LVDH}$ ）用于产生中断或者解除复位，低电压检测电平（ $V_{LVDL}$ ）用于产生复位。

#### (b) 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）

将选项字节 000C1H 选择的 1 个检测电压（ $V_{LVD}$ ）用于产生或者解除复位。

#### (c) 中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）

将选项字节 000C1H 选择的 1 个检测电压（ $V_{LVD}$ ）用于产生中断或者解除复位。

在中断 & 复位模式中，能设定 2 个检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ）；在复位模式和中断模式中，能设定 1 个检测电压（ $V_{LVD}$ ）。

通过选择选项字节（LVIMDS0、LVIMDS1）产生以下的复位和中断信号。

中断 & 复位模式 (LVIMDS1、LVIMDS0=1、0)	复位模式 (LVIMDS1、LVIMDS0=1、1)	中断模式 (LVIMDS1、LVIMDS0=0、1)
当 $V_{DD} < V_{LVDH}$ 时，产生内部中断信号； 当 $V_{DD} < V_{LVDL}$ 时，产生内部复位； 当 $V_{DD} \geq V_{LVDH}$ 时，解除内部复位。	当 $V_{DD} < V_{LVD}$ 时，产生内部复位； 当 $V_{DD} \geq V_{LVD}$ 时，解除内部复位。	在解除POR后的第1次工作电压上升时， 当检测到 $V_{DD} \geq V_{LVD}$ 时，解除内部复位； 在解除POR后第2次以后的工作电压下降时， 当检测到 $V_{DD} < V_{LVD}$ 或者 $V_{DD} \geq V_{LVD}$ 时，产生中断请求信号。

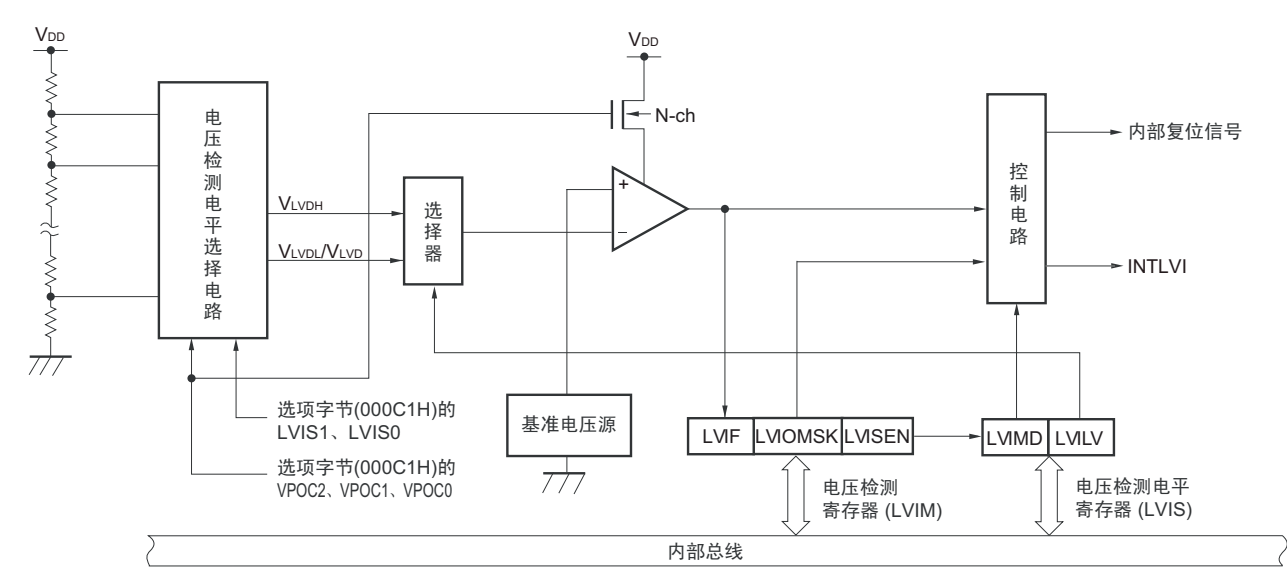
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位，就将复位控制标志寄存器（RESF）的 bit0（LVIRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

19.2 电压检测电路的结构

电压检测电路的框图如图 19-1 所示。

图 19-1 电压检测电路的框图



19.3 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器（LVIM）
- 电压检测电平寄存器（LVIS）

19.3.1 电压检测寄存器（LVIM）

此寄存器设定允许或者禁止改写电压检测电平寄存器（LVIS），并且确认 LVD 输出的屏蔽状态。  
通过 1 位或者 8 位存储器操作指令设定 LVIM 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 19-2 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

LVISEN

0

0

0

0

0

LVIOMSK

LVIF

LVISEN

电压检测电平寄存器（LVIS）的允许 / 禁止改写的设定

0

禁止改写 LVIS 寄存器（LVIOMSK=0（LVD 输出屏蔽无效））。

1

允许改写 LVIS 寄存器（LVIOMSK=1（LVD 输出屏蔽有效））注 3。

LVIOMSK

LVD 输出的屏蔽状态标志

0

LVD 输出屏蔽无效。

1

LVD 输出屏蔽有效注 3、4。

LVIF

电压检测标志

0

电源电压（V<sub>DD</sub>）≥检测电压（V<sub>LVD</sub>）或者禁止 LVD 运行。

1

电源电压（V<sub>DD</sub>）<检测电压（V<sub>LVD</sub>）

- 注 1. 复位值因复位源而变。  
在 LVD 发生复位时，不对 LVIM 寄存器的值进行复位而保持原来的值；在其他复位时，将 LVISEN 位清“0”。
2. bit0 和 bit1 是只读位。
3. 只有在选择中断 & 复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能设定，在其他模式中不能更改初始值。
4. 只有在选择中断 & 复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时，LVIOMSK 位才在以下期间自动变为“1”，屏蔽 LVD 产生的复位或者中断。
- LVISEN=1 的期间
  - 从发生 LVD 中断开始到 LVD 检测电压稳定为止的等待时间
  - 从更改 LVILV 位的值到 LVD 检测电压稳定为止的等待时间

19.3.2    电压检测电平寄存器（LVIS）

这是设定电压检测电平的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 LVIS 寄存器。  
在产生复位信号后，此寄存器的值变为“00H/01H/81H”注 1。

图 19-3    电压检测电平寄存器（LVIS）的格式

地址：FFFAAH

复位后：00H/01H/81H <sup>注 1</sup>

R/W

符号

7

6

5

4

3

2

1

0

LVIS

LVIMD

0

0

0

0

0

0

0

LVILV

LVIMD <sup>注 2</sup>

电压检测的运行模式

0

中断模式

1

复位模式

LVILV <sup>注 2</sup>

LVD 检测电平

0

高电压检测电平（V<sub>LVDH</sub>）

1

低电压检测电平（V<sub>LVDL</sub> 或者 V<sub>LVD</sub>）

- 注  1. 复位值因复位源和选项字节的设定而变。  
在发生 LVD 复位时，不将此寄存器清“00H”。  
在发生 LVD 以外的复位时，此寄存器的值如下：
- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
  - 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
  - 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H
2. 只有在选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”（中断 & 复位模式）时才能写“0”。在其他情况下，禁止写数据并且通过产生复位或者中断自动进行值的替换。
- 注意 1. 要改写 LVIS 寄存器时，必须在将 LVISEN 位（LVIM 寄存器的 bit7）置“1”后进行。
2. 通过选项字节（000C1H）设定 LVD 的运行模式和检测电压（V<sub>LVDH</sub>、V<sub>LVDL</sub>、V<sub>LVD</sub>）。选项字节（000C1H）的设定如表 19-1 所示。有关选项字节的详细内容，请参照“第 22 章 选项字节”。

表 19-1 通过用户选项字节（000C1H）进行 LVD 运行模式和检测电压的设定

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断&复位模式）

检测电压			选项字节的设定值						
V <sub>LVDH</sub>		V <sub>LVDL</sub>	模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	下降	LVIMDS1	LVIMDS0					
2.92V	2.86V	2.75V	1	0	0	1	1	1	0
3.02V	2.96V							0	1
上述以外			禁止设定。						

- LVD 的设定（复位模式）

检测电压		选项字节的设定值						
V <sub>LVD</sub>		模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	LVIMDS1	LVIMDS0					
2.81V	2.75V	1	1	0	1	1	1	1
2.92V	2.86V				1	1	1	0
3.02V	2.96V				1	1	0	1
3.13V	3.06V				0	1	0	0
上述以外		禁止设定。						

- LVD 的设定（中断模式）

检测电压		选项字节的设定值						
V <sub>LVD</sub>		模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	LVIMDS1	LVIMDS0					
2.81V	2.75V	0	1	0	1	1	1	1
2.92V	2.86V				1	1	1	0
3.02V	2.96V				1	1	0	1
3.13V	3.06V				0	1	0	0
上述以外		禁止设定。						

- LVD 的设定（LVD 为 OFF）

检测电压			选项字节的设定值						
$V_{LVD}$			模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降		LVIMDS1	LVIMDS0					
—	—		0/1	1	1	×	×	×	×
上述以外			禁止设定。						

备注 1. ×：忽略

- 检测电压是 TYP. 值。详细内容请参照“27.6.5 LVD 电路特性”。

## 19.4 电压检测电路的运行

### 19.4.1 用作复位模式时的设定

- 开始运行的情况

在以下初始设定的状态下开始运行。

事先通过选项字节000C1H设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1））和检测电压（ $V_{LVD}$ ）。

- 将电压检测寄存器（LVIM）的bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 如果将选项字节的LVIMDS1位和LVIMDS0位都置“1”，就将LVIS寄存器的初始值置“81H”。  
bit7（LVIMD）为“1”（复位模式）。  
bit0（LVILV）为“1”（低电压检测电平： $V_{LVDL}$ 或者 $V_{LVD}$ ）。

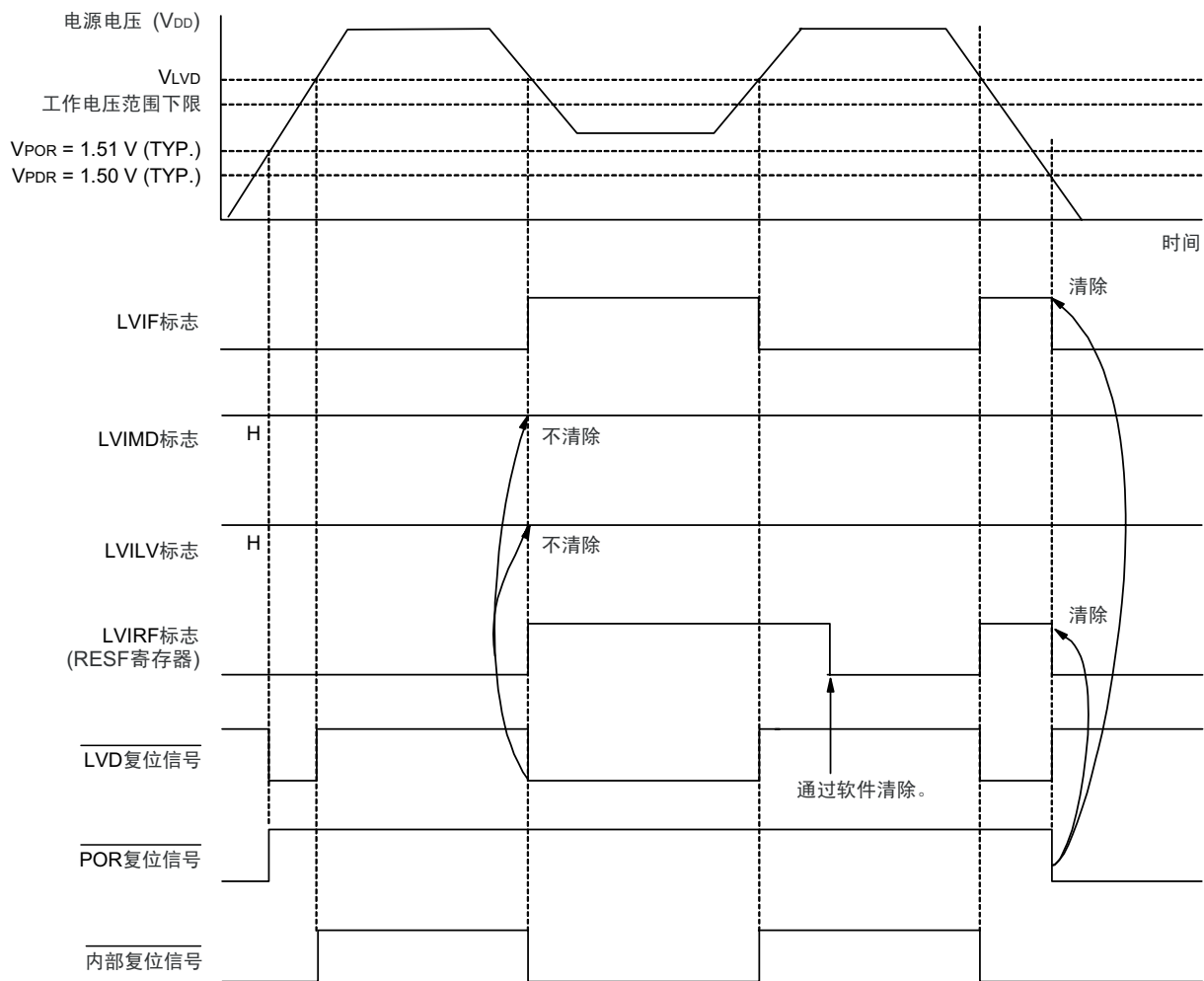
- LVD复位模式的运行

当接通电源时，复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持LVD的内部复位状态。如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就解除内部复位。

当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于电压检测电平（ $V_{LVD}$ ），就产生LVD的内部复位。

电压检测电路的内部复位信号的发生时序如图 19-4 所示。

图 19-4 内部复位信号的发生时序（选项字节的 LVIMDS1、LVIMDS0=1、1）



备注  $V_{POR}$ : POR 电源电压上升检测电压  
 $V_{PDR}$ : POR 电源电压下降检测电压



### 19.4.2 用作中断模式时的设定

- 开始运行的情况

事先通过选项字节 000C1H 设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1））和检测电压（ $V_{LVD}$ ）。

在以下初始设定的状态下开始运行。

- 电压检测寄存器（LVIM）的 bit7（LVISEN）为“0”（禁止改写电压检测电平寄存器（LVIS））。
- 如果将选项字节的 LVIMDS1 位和 LVIMDS0 位分别置“0”和“1”，就将 LVIS 寄存器的初始值置“01H”。  
bit7（LVIMD）为“0”（中断模式）。  
bit0（LVILV）为“1”（低电压检测电平： $V_{LVDL}$  或者  $V_{LVD}$ ）。

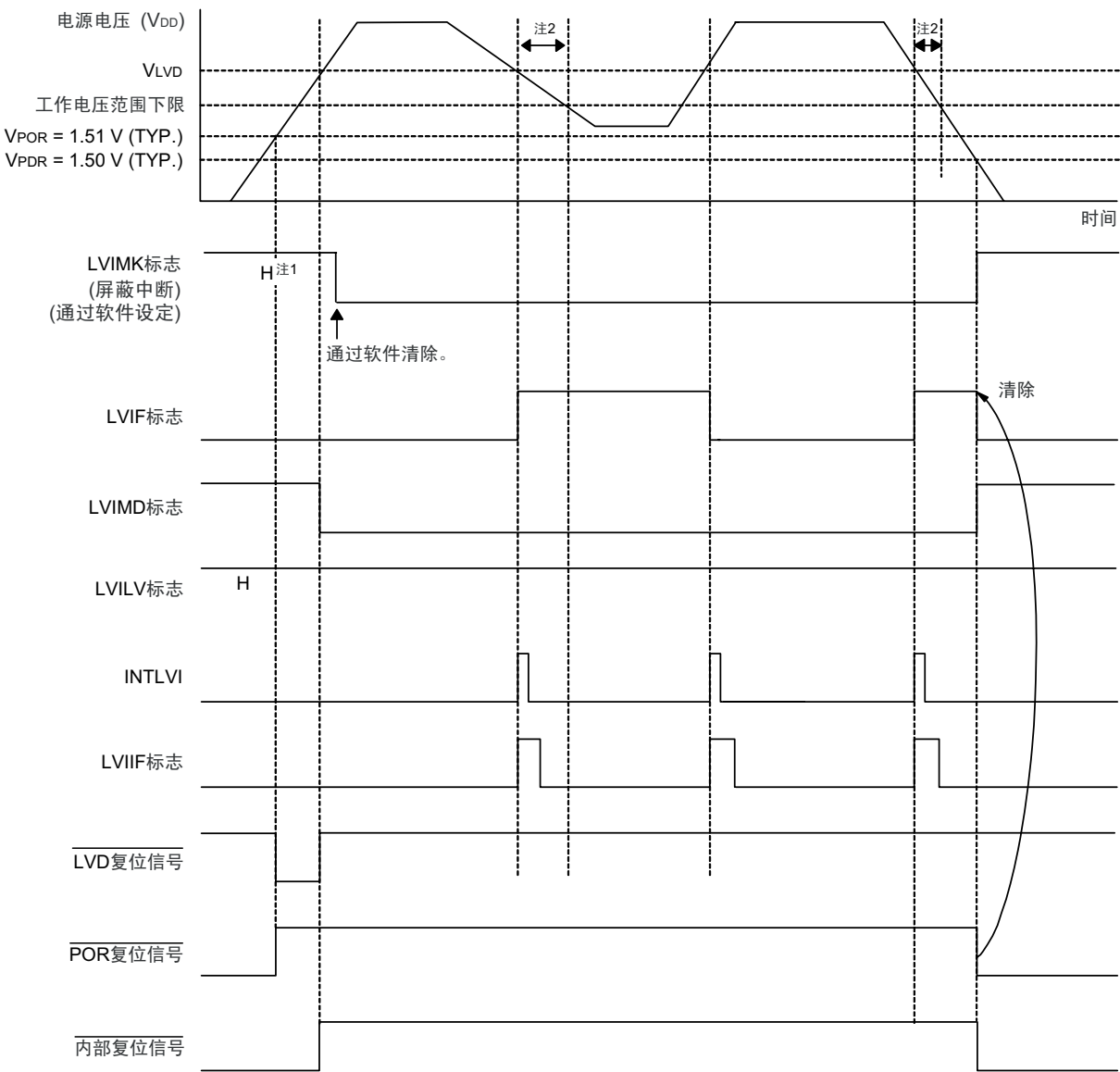
- LVD 中断模式的运行

当接通电源时（解除 POR 后的第 1 次），中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ）前保持 LVD 的内部复位状态。如果电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就解除内部复位。

当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于电压检测电平（ $V_{LVD}$ ）或者在解除 POR 后第 2 次以后的电源电压（ $V_{DD}$ ）超过电压检测电平（ $V_{LVD}$ ），就产生 LVD 的中断请求信号（INTLVI），而且必须在工作电压低于“27.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

电压检测电路的内部中断信号的发生时序如图 19-5 所示。

图 19-5 中断信号的发生时序（选项字节的 LVIMDS1、LVIMDS0=0、1）



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 当工作电压下降时，必须在工作电压低于“27.4 AC 特性”所示的工作电压范围前，通过 STOP 模式的转移或者外部复位，置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注  $V_{POR}$ : POR 电源电压上升检测电压  
 $V_{PDR}$ : POR 电源电压下降检测电压

### 19.4.3 用作中断 & 复位模式时的设定

- 开始运行的情况

事先通过选项字节 000C1H 设定运行模式（中断 & 复位模式（LVIMDS1、LVIMDS0=1、0））和检测电压（ $V_{LVDH}$ 、 $V_{LVDL}$ ）。

在以下初始设定的状态下开始运行。

- 电压检测寄存器（LVIM）的 bit7（LVISEN）为“0”（禁止改写电压检测电平寄存器（LVIS））。
- 如果将选项字节的 LVIMDS1 位和 LVIMDS0 位分别置“1”和“0”，就将 LVIS 寄存器的初始值置“00H”。bit7（LVIMD）为“0”（中断模式）。bit0（LVILV）为“0”（高电压检测电平： $V_{LVDH}$ ）。

- LVD 中断 & 复位模式的运行

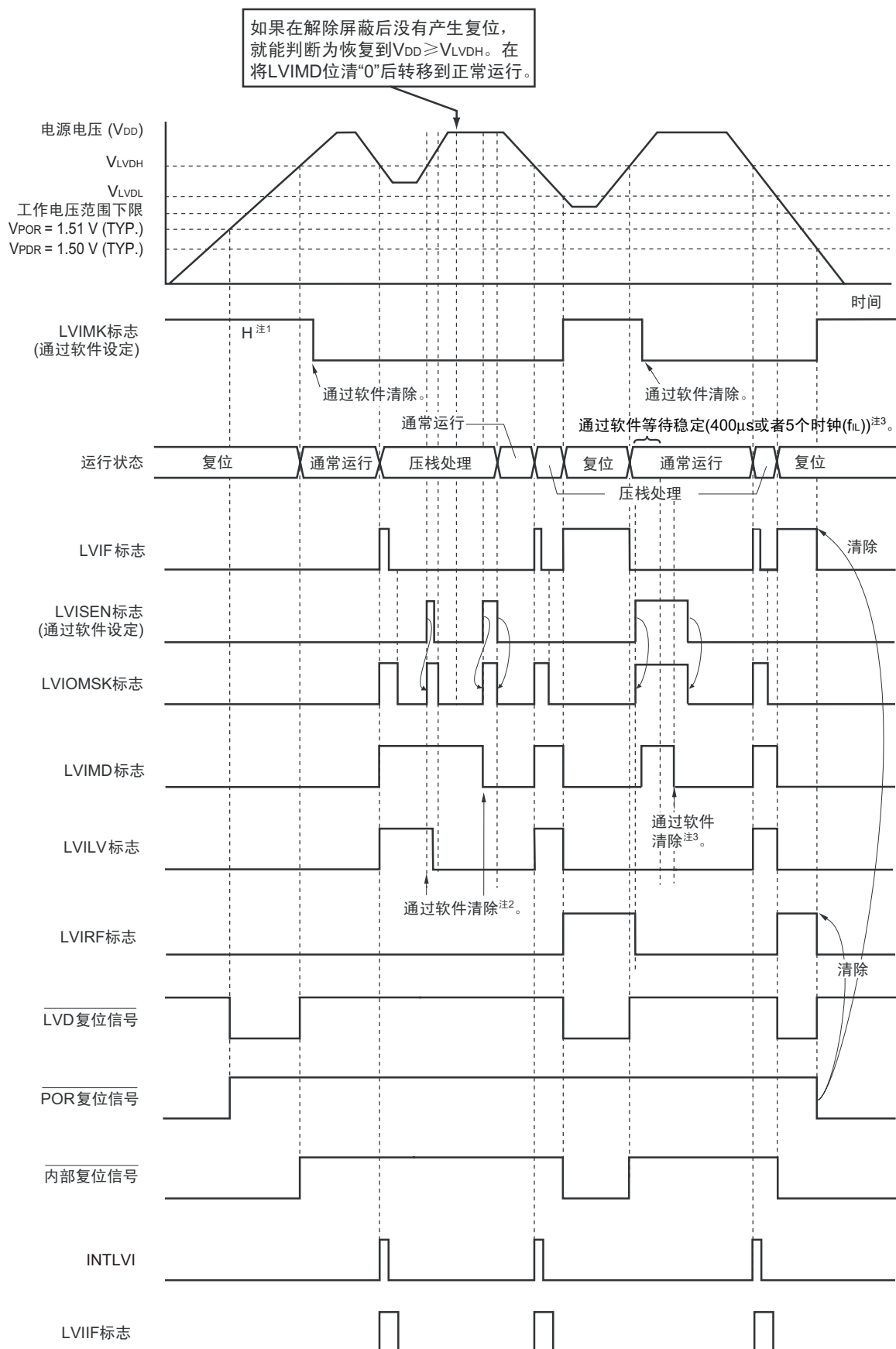
当接通电源时，中断 & 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（ $V_{DD}$ ）超过高电压检测电平（ $V_{LVDH}$ ）前保持 LVD 的内部复位状态。如果电源电压（ $V_{DD}$ ）超过高电压检测电平（ $V_{LVDH}$ ），就解除内部复位。

当工作电压下降时，如果电源电压（ $V_{DD}$ ）低于高电压检测电平（ $V_{LVDH}$ ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（ $V_{DD}$ ）低于低电压检测电平（ $V_{LVDL}$ ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压在不低于低电压检测电压（ $V_{LVDL}$ ）的状态下恢复到高电压检测电压（ $V_{LVDH}$ ）或者更高，也不产生中断请求信号。

当使用 LVD 中断 & 复位模式时，必须按照“图 19-7 工作电压的确认/复位的设定步骤”和“图 19-8 中断 & 复位模式的初始设定步骤”所示的流程图的步骤进行设定。

电压检测电路的内部复位信号和中断信号的发生时序如图 19-6 所示。

图 19-6 复位 &amp; 中断信号的发生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (1/2)

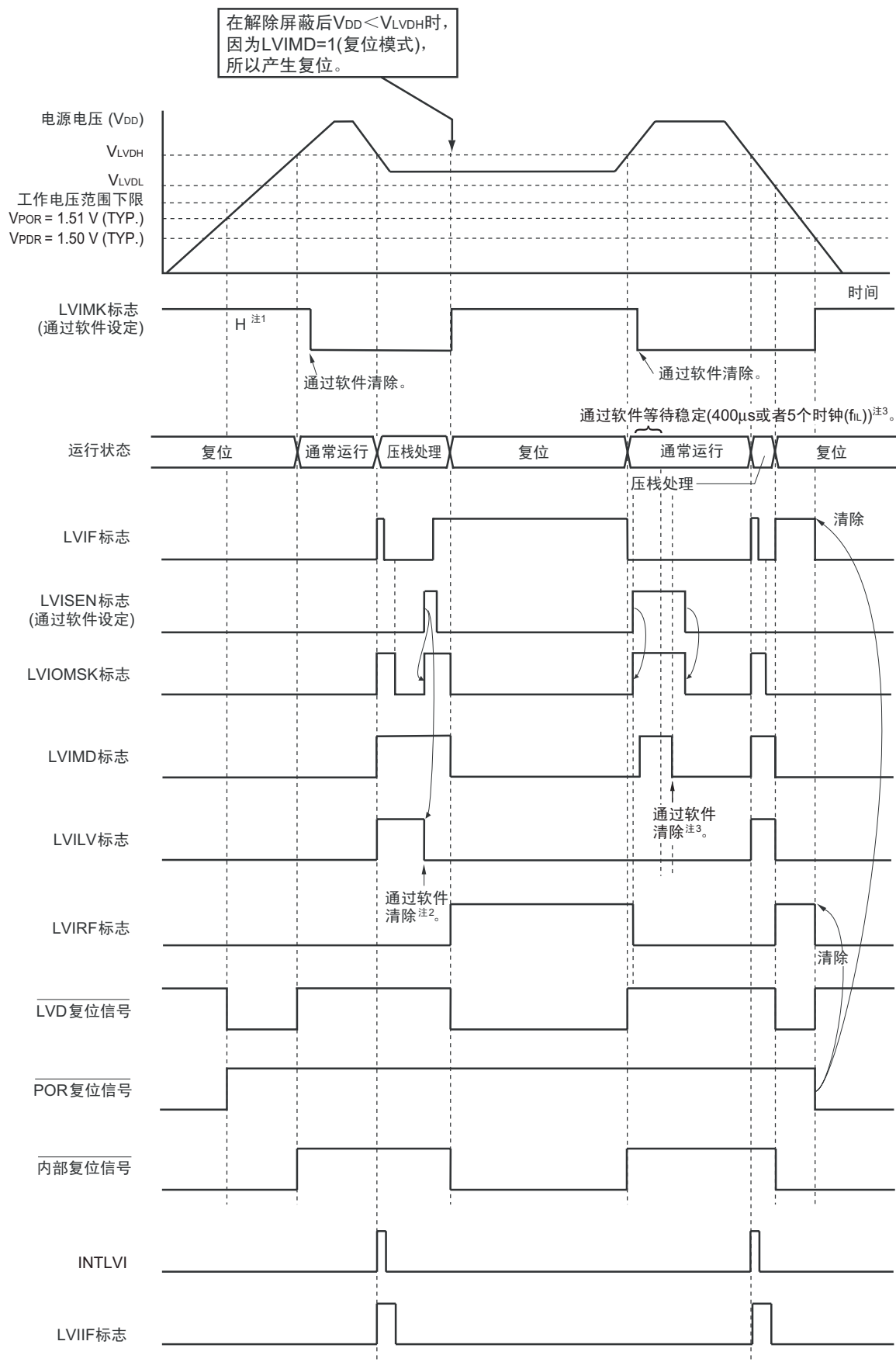


- 注
1. 在产生复位信号后，LVIMK 标志变为“1”。
  2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 19-7 工作电压的确认 / 复位的设定步骤”进行设定。
  3. 当使用中断 & 复位模式时，必须在解除复位后按照“图 19-8 中断 & 复位模式的初始设定步骤”进行设定。

备注  $V_{POR}$ : POR 电源电压上升检测电压

$V_{PDR}$ : POR 电源电压下降检测电压

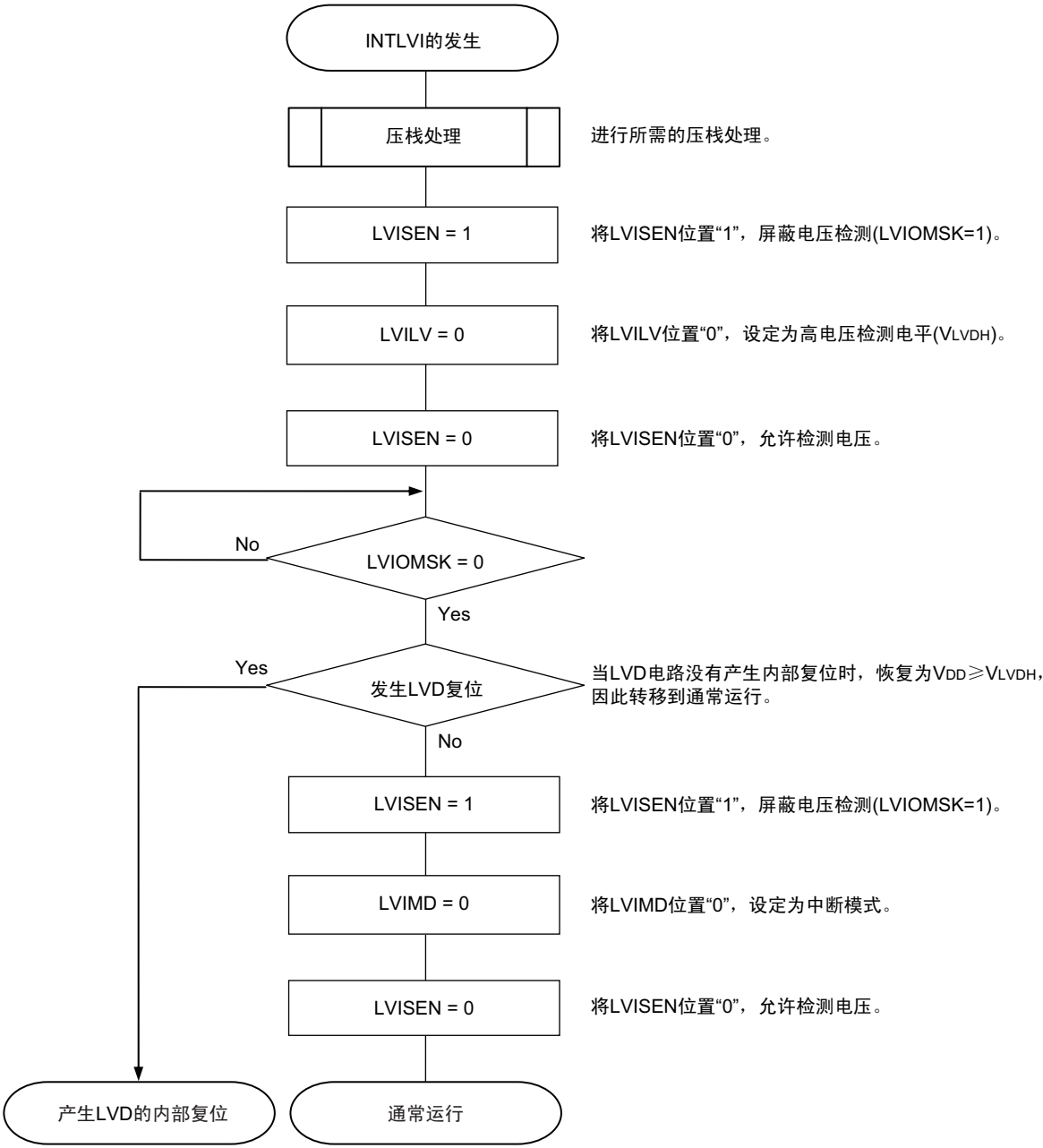
图 19-6 中断 &amp; 复位信号的发生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (2/2)



- 注
- 1. 在产生复位信号后，LVIMK 标志变为“1”。
  - 2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 19-7 工作电压的确认 / 复位的设定步骤”进行设定。
  - 3. 当使用中断 & 复位模式时，必须在解除复位后按照“图 19-8 中断 & 复位模式的初始设定步骤”进行设定。

备注     $V_{POR}$ : POR 电源电压上升检测电压  
          $V_{PDR}$ : POR 电源电压下降检测电压

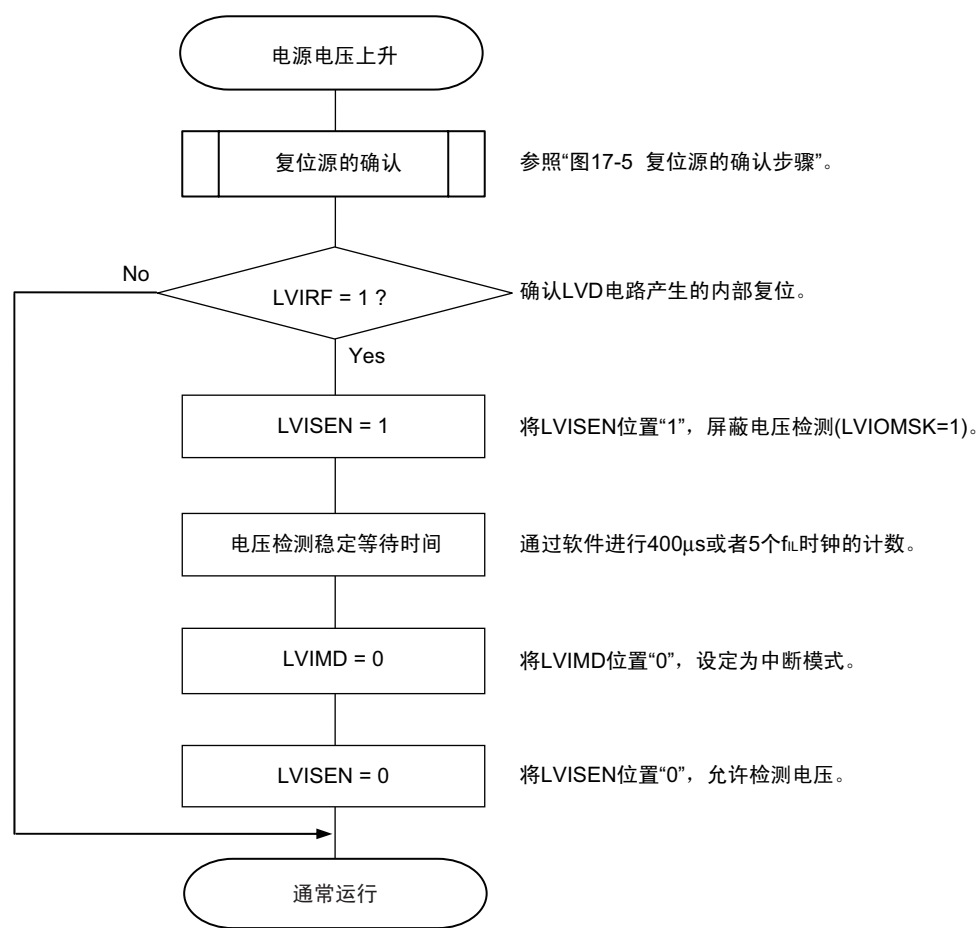
图 19-7 工作电压的确认 / 复位的设定步骤



如果设定中断 & 复位模式（LVIMDS1、LVIMDS0=1、0），就在解除 LVD 复位后（LVIRF=1）需要 400μs 或者 5 个  $f_{\text{IL}}$  时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时，必须将 LVISEN 位置“1”，屏蔽 LVD 产生的复位或者中断的产生。

中断 & 复位模式的初始设定步骤如图 19-8 所示。

图 19-8 中断 & 复位模式的初始设定步骤



备注  $f_{\text{IL}}$ ：低速内部振荡器的时钟频率



19.5 电压检测电路的注意事项

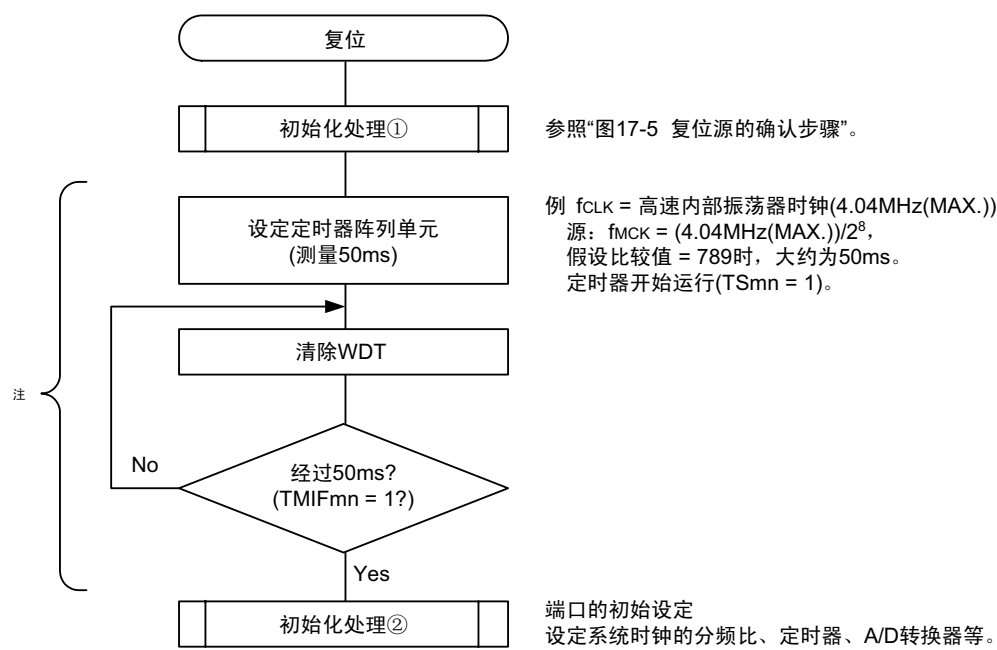
(1) 有关接通电源时的电压波动

对于电源电压（ $V_{DD}$ ）在 LVD 检测电压附近发生一定时间波动的系统，有可能重复进入复位状态和复位解除状态。能通过以下的处理，任意设定解除复位到单片机开始运行的时间。

<处理>

在解除复位后，必须通过使用定时器的软件计数器，在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 19-9 LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子



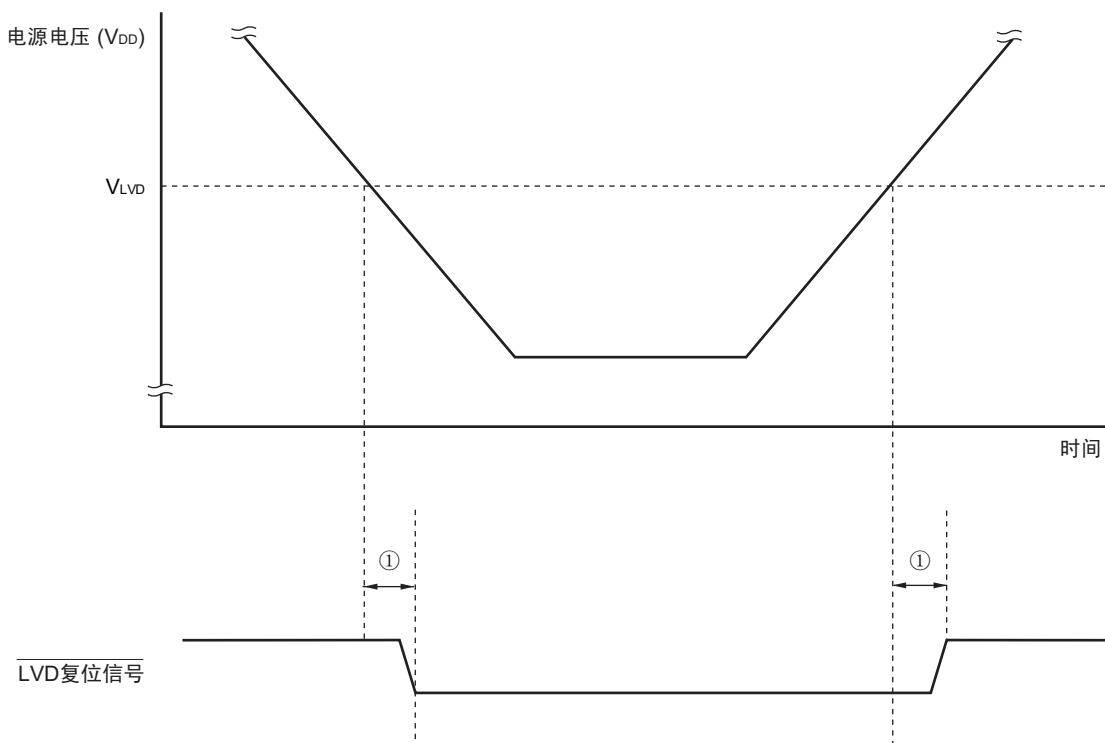
注 如果在此期间再次发生复位，就不转移到初始化处理②。

备注 m=0, n=0 ~ 3

## (2) 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压 ( $V_{DD}$ ) < LVD 检测电压 ( $V_{LVD}$ ) 到产生 LVD 复位为止会发生延迟。同样, 从 LVD 检测电压 ( $V_{LVD}$ ) ≤ 电源电压 ( $V_{DD}$ ) 到解除 LVD 复位为止也会发生延迟 (参照图 19-10)。

图 19-10 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟



①: 检测延迟 (300 $\mu$ s (MAX.))

- (3) 在使用中断模式并且工作电压下降时, 必须在工作电压低于“27.4 AC 特性”所示的工作电压范围前, 通过 STOP 模式的转移或者外部复位引脚, 置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。
- (4) 当将 LVD 置为 OFF 时, 需要进行外部复位。在进行外部复位时, 必须至少给  $\overline{\text{RESET}}$  引脚输入 10 $\mu$ s 的低电平。如果在电源电压上升时进行外部复位, 就必须在给  $\overline{\text{RESET}}$  引脚输入低电平后接通电源, 而且在“27.4 AC 特性”所示的工作电压范围内至少保持 10 $\mu$ s 的低电平, 然后输入高电平。在电源电压上升后, 不能在“27.4 AC 特性”所示的工作电压范围外给  $\overline{\text{RESET}}$  引脚输入高电平。

## 第 20 章 安全功能

### 20.1 安全功能的概要

为了对应 IEC60730 和 IEC61508 安全标准，R7F0C010 内置以下安全功能。  
此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

#### (1) 闪存 CRC 运算功能（高速 CRC、通用 CRC）

通过 CRC 运算检测闪存的数据错误。

能根据不同的用途和使用条件，分别使用以下 2 个 CRC。

- “高速 CRC”...在初始化程序中，能停止 CPU 的运行并且高速检查整个代码闪存区。
- “通用 CRC”...在 CPU 运行中，不限于代码闪存区而能用于多用途的检查。

#### (2) RAM 奇偶校验错误检测功能

在读 RAM 数据时，检测奇偶校验错误。

#### (3) RAM 保护功能

防止因 CPU 失控而改写 RAM 数据。

#### (4) SFR 保护功能

防止因 CPU 失控而改写 SFR。

#### (5) 非法存储器存取检测功能

检测对非法存储区（没有存储器的区域或者限制存取的区域）的非法存取。

#### (6) 频率检测功能

能使用定时器阵列单元进行 CPU/ 外围硬件时钟频率的自检。

#### (7) A/D 测试功能

能通过 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANI）、温度传感器输出和内部基准电压输出的 A/D 转换进行 A/D 转换的自检。

备注 有关符合 IEC60730 标准的使用方法，请参照应用说明（R01AN0749）。

## 20.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
<ul style="list-style-type: none"> <li>闪存 CRC 控制寄存器 (CRC0CTL)</li> <li>闪存 CRC 运算结果寄存器 (PGCRCL)</li> </ul>	闪存 CRC 运算功能 (高速 CRC)
<ul style="list-style-type: none"> <li>CRC 输入寄存器 (CRCIN)</li> <li>CRC 数据寄存器 (CRCD)</li> </ul>	CRC 运算功能 (通用 CRC)
<ul style="list-style-type: none"> <li>RAM 奇偶校验错误控制寄存器 (RPECTL)</li> </ul>	RAM 奇偶校验错误检测功能
<ul style="list-style-type: none"> <li>非法存储器存取检测控制寄存器 (IAWCTL)</li> </ul>	RAM 保护功能
	SFR 保护功能
	非法存储器存取检测功能
<ul style="list-style-type: none"> <li>定时器输入选择寄存器 0 (TIS0)</li> </ul>	频率检测功能
<ul style="list-style-type: none"> <li>A/D 测试寄存器 (ADTES)</li> </ul>	A/D 测试功能

有关各寄存器的内容，在“20.3 闪存 CRC 运算功能（高速 CRC）的运行”中进行说明。

## 20.3 闪存 CRC 运算功能（高速 CRC）的运行

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设定（初始化）程序中检查整个代码闪存区。只能通过 RAM 内的程序以主系统时钟的 HALT 模式执行高速 CRC。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短（例如，16KB 闪存：171μs@24MHz）。

CRC 生成多项式对应 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。

以 bit31→bit0 的 MSB 优先进行运算。

**注意** 在片上调试时，因为配置监视程序，所以 CRC 的运算结果不同。

**备注** 因为通用 CRC 为 LSB 优先，所以运算结果不同。

20.3.1 闪存 CRC 控制寄存器（CRC0CTL）

这是设定高速 CRC 运算器的运行控制和运算范围的寄存器。  
通过 1 位或者 8 位存储器操作指令设定 CRC0CTL 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 20-1 闪存 CRC 控制寄存器（CRC0CTL）的格式

地址: F02F0H		复位后: 00H		R/W				
符号	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	高速 CRC 运算器的运行控制
0	停止运行。
1	通过执行 HALT 指令开始运算。

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	高速 CRC 的运算范围
0	0	0	0	0	0	0000H ~ 3FFBH（16K-4 字节）
上述以外						禁止设定。

备注 必须事先将用于比较的 CRC 运算结果期待值存入闪存的最后 4 字节，因此运算范围为减去 4 字节的范围。

20.3.2 闪存 CRC 运算结果寄存器（PGCRCL）

这是保存高速 CRC 运算结果的寄存器。  
通过 16 位存储器操作指令设定 PGCRCL 寄存器。  
在产生复位信号后，此寄存器的值变为“0000H”。

图 20-2 闪存 CRC 运算结果寄存器（PGCRCL）的格式

地址: F02F2H	复位后: 0000H	R/W						
符号	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0
PGCRC15 ~ 0			高速 CRC 的运算结果					
0000H ~ FFFFH			保存高速 CRC 的运算结果。					

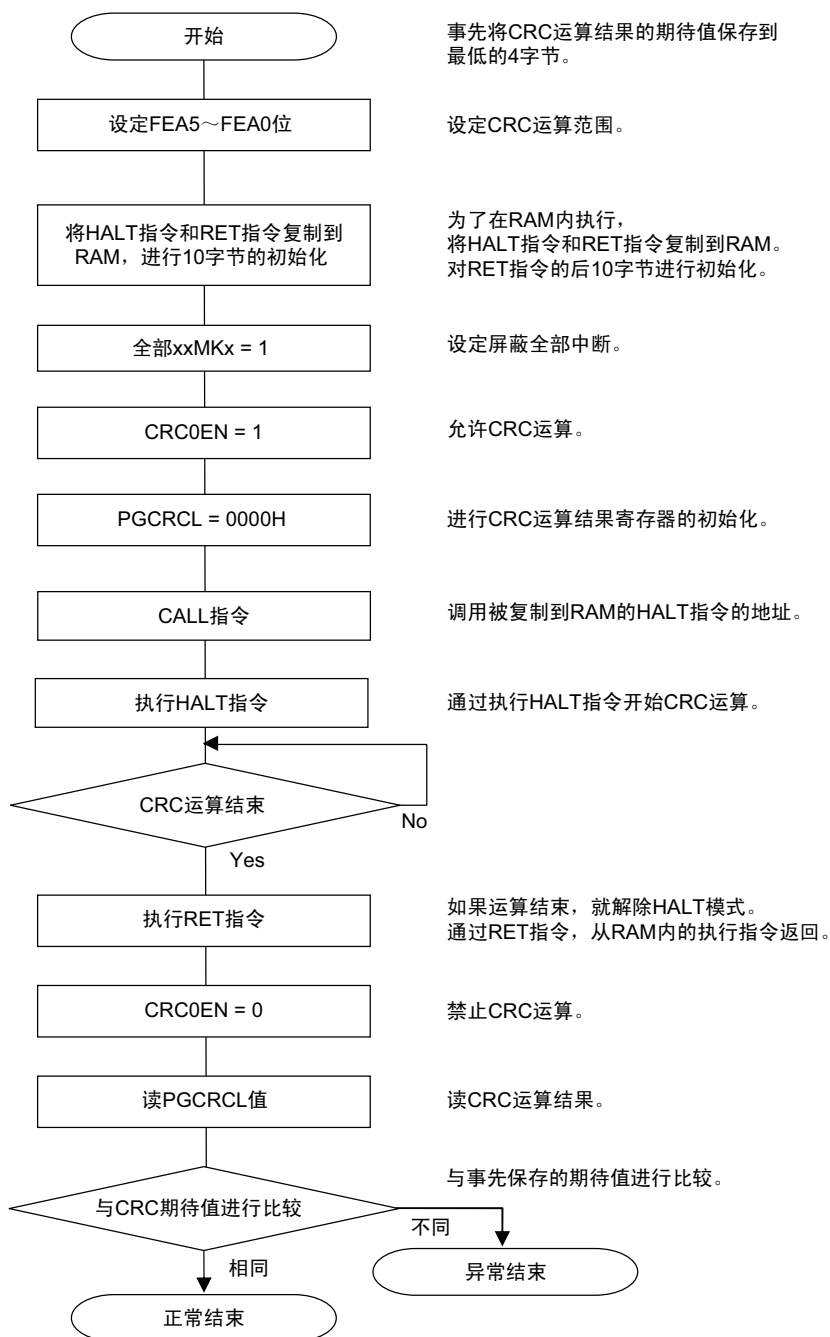
注意 只有在 CRC0EN（CRC0CTL 寄存器的 bit7）位为“1”时才能写 PGCRCL 寄存器。

闪存 CRC 运算功能（高速 CRC）的流程图如图 20-3 所示。

### 20.3.3 操作流程

闪存 CRC 运算功能（高速 CRC）的流程图如图 20-3 所示。

图 20-3 闪存 CRC 运算功能（高速 CRC）的流程图



注意 1. 只以代码闪存为 CRC 运算的对象。

2. 必须将 CRC 运算的期待值保存在代码闪存中的运算范围后的区域。

3. 通过在 RAM 区执行 HALT 指令，CRC 运算变为有效。

必须在 RAM 区执行 HALT 指令。

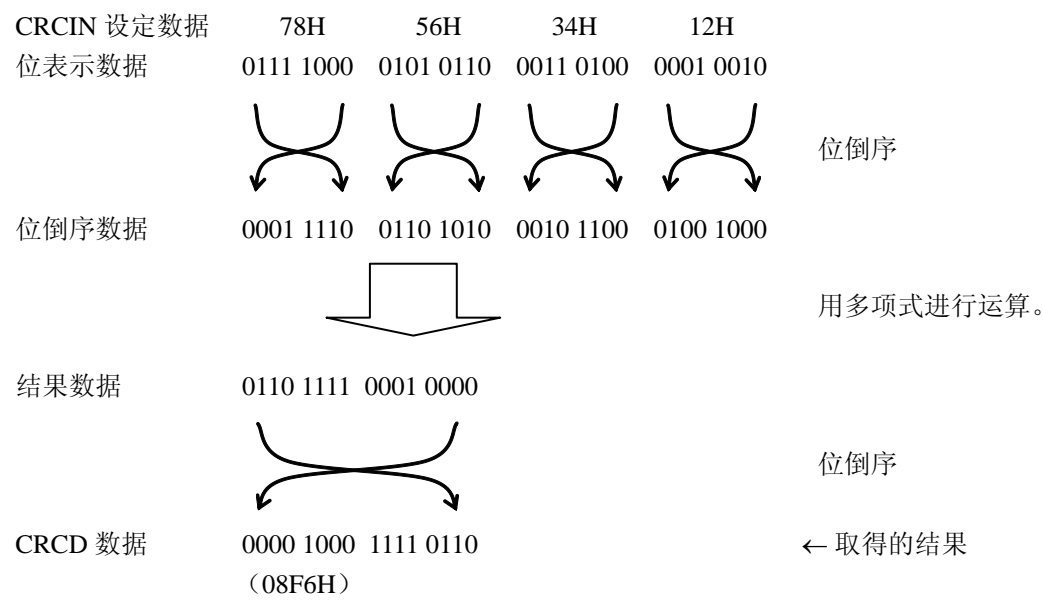
CRC 运算的期待值能使用综合开发环境“CubeSuite+”进行计算。有关详细内容，请参照《CubeSuite+ Development Environment User's Manual》。

20.4  CRC 运算功能（通用 CRC）

为了必须保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。

此通用 CRC 能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定要确认的数据。HALT 模式中的 CRC 运算功能只能在 DMA 传送过程中使用。

CRC 生成多项式使用 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以 LSB 优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从 LSB 发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给 CRCIN 寄存器写值，从 CRCD 寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行 CRC 运算的结果。



注意  在执行程序的过程中，因为调试程序将软件断点的设定行改写为断点指令，所以如果在 CRC 运算的对象区设定软件断点，CRC 的运算结果就不同。

20.4.1  CRC 输入寄存器（CRCIN）

这是设定通用 CRC 的 CRC 计算数据的 8 位寄存器。

能设定的范围为“00H ~ FFH”。

通过 8 位存储器操作指令设定 CRCIN 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

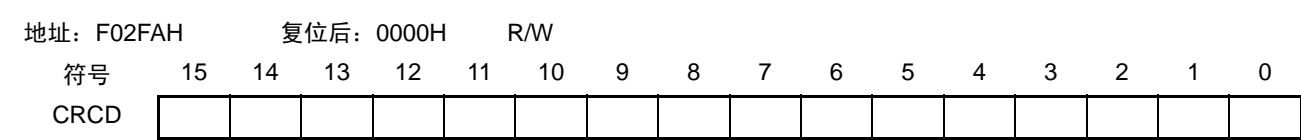
图 20-4  CRC 输入寄存器（CRCIN）的格式

地址: FFFACH		复位后: 00H		R/W				
符号	7	6	5	4	3	2	1	0
CRCIN								
bit7 ~ 0			功能					
00H ~ FFH			数据输入					

20.4.2 CRC 数据寄存器（CRCD）

这是保存通用 CRC 运算结果的寄存器。  
能设定的范围为“0000H ~ FFFFH”。  
在写 CRCIN 寄存器开始经过 1 个 CPU/ 外围硬件时钟 ( $f_{CLK}$ ) 后，将 CRC 运算结果保存到 CRCD 寄存器。  
通过 16 位存储器操作指令设定 CRCD 寄存器。  
在产生复位信号后，此寄存器的值变为“0000H”。

图 20-5 CRC 数据寄存器（CRCD）的格式

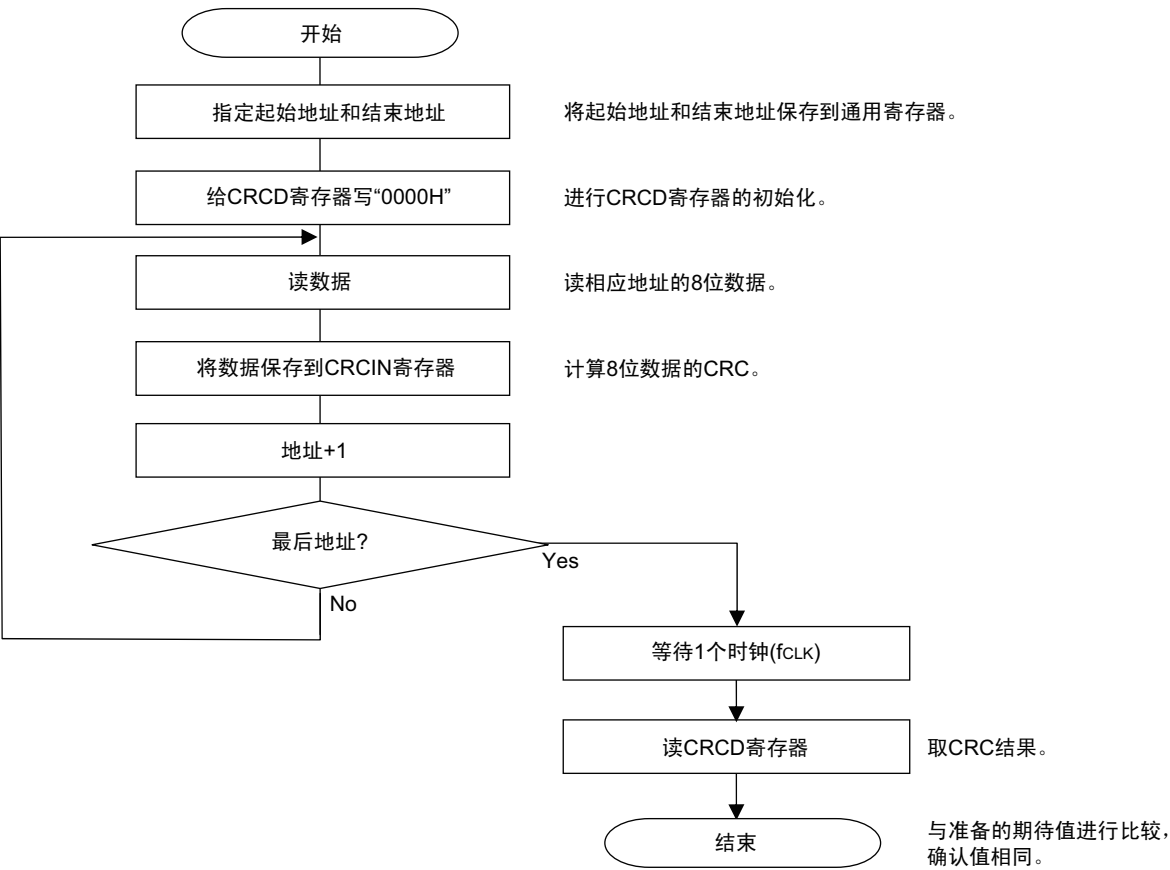


- 注意 1. 要读 CRCD 寄存器的写入值时，必须在写 CRCIN 寄存器前读 CRCD 寄存器。  
2. 如果 CRCD 寄存器的写操作与运算结果的保存发生竞争，就忽视写操作。

20.4.3 操作流程

CRC 运算功能（通用 CRC）的流程图如图 20-6 所示。

图 20-6 CRC 运算功能（通用 CRC）的流程图





20.5 RAM 奇偶校验错误检测功能

IEC60730 标准要求确认 RAM 数据。因此，R7F0C010 的 RAM 每 8 位附加 1 位奇偶校验位。RAM 奇偶校验错误检测功能在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位，并且能在发生奇偶校验错误时产生复位。

20.5.1 RAM 奇偶校验错误控制寄存器（RPECTL）

此寄存器控制奇偶校验的错误确认位和因奇偶校验错误而产生复位。

通过 1 位或者 8 位存储器操作指令设定 RPECTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-7 RAM 奇偶校验错误控制寄存器（RPECTL）的格式

地址: F00F5H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF	
	RPERDIS	奇偶校验错误复位的屏蔽标志							
	0	允许产生奇偶校验错误复位。							
	1	禁止产生奇偶校验错误复位。							
	RPEF	奇偶校验错误状态标志							
	0	没有发生奇偶校验错误。							
	1	发生奇偶校验错误。							

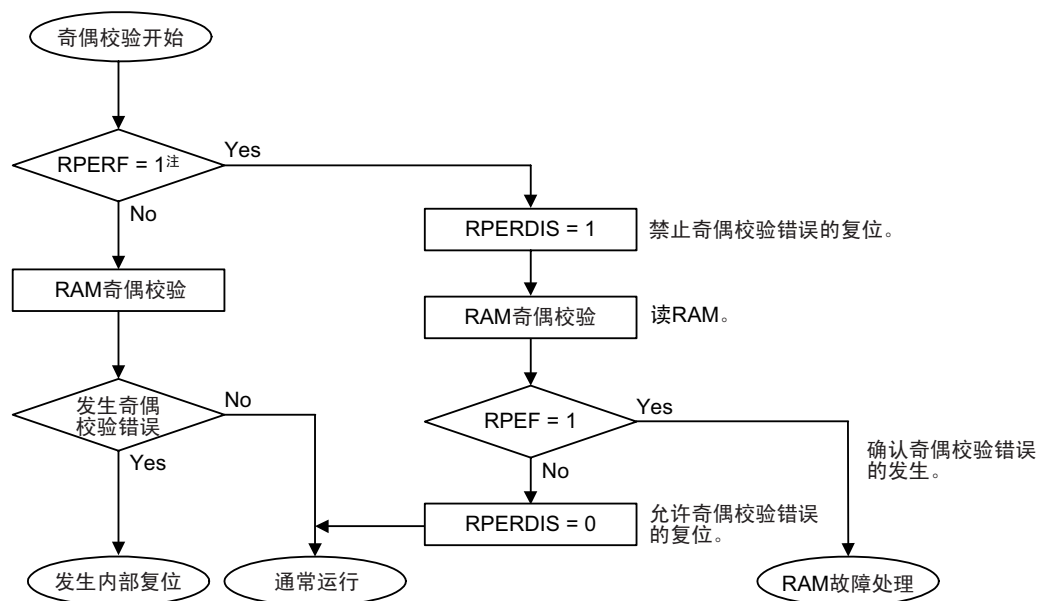
注意 在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位。

因此，当允许产生 RAM 奇偶校验错误复位（RPERDIS=0）时，必须在存取数据时并且在读数据前对“所用 RAM 区”进行初始化。

因为 RL78 为流水线运行，所以 CPU 进行预读，有可能因读所用 RAM 区前的未初始化 RAM 区而发生 RAM 奇偶校验错误。因此，当允许产生 RAM 奇偶校验错误复位（RPERDIS=0）时，必须在从 RAM 区执行指令时对“所用 RAM 区+10 字节”的区域进行初始化；当允许产生 RAM 奇偶校验错误复位（RPERDIS=0）并且使用自编程功能时，必须在改写前对“要改写的 RAM 区+10 字节”的区域进行初始化。

- 备注 1. RAM 奇偶校验错误检查总是处于允许状态，能通过 RPEF 标志确认其结果。
2. 初始状态为允许产生奇偶校验错误复位（RPERDIS=0）。
- 即使设定为禁止产生奇偶校验错误复位（RPERDIS=1），也在发生奇偶校验错误时将 RPEF 标志置“1”。
3. 因 RAM 奇偶校验错误而将 RPEF 标志置“1”，通过写“0”或者全部复位源将 RPEF 标志清“0”。当 RPEF 标志为“1”时，即使读未发生奇偶校验错误的 RAM，RPEF 标志也保持“1”的状态。

图 20-8 RAM 奇偶校验的流程



注 有关 RAM 奇偶错误的内部复位的确认，请参照“第 17 章 复位功能”。

20.6 RAM 保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护存储在 RAM 的重要数据。  
此 RAM 保护功能用于保护被指定的 RAM 空间的数据。  
如果设定为 RAM 保护功能，指定空间的 RAM 写操作就无效，但是能正常读取。

20.6.1 非法存储器存取检测控制寄存器（IAWCTL）

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。  
RAM 保护功能使用 GRAM1 位和 GRAM0 位。  
通过 8 位存储器操作指令设定 IAWCTL 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 20-9 非法存储器存取检测控制寄存器（IAWCTL）的格式

地址：F0078H	复位后：00H	R/W						
符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM 保护空间
0	0	无效。能写 RAM。
0	1	RAM 低位地址开始的 128 字节
1	0	RAM 低位地址开始的 256 字节
1	1	RAM 低位地址开始的 512 字节

20.7 SFR 保护功能

为了必须保证运行过程中的安全，IEC61508标准要求即使CPU失控也需要保护重要的SFR，使其免遭改写。  
SFR 保护功能用于保护端口功能、中断功能、时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的数据。  
如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

20.7.1 非法存储器存取检测控制寄存器（IAWCTL）

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。  
SFR 保护功能使用 GPORT 位、GINT 位和 GCSC 位。  
通过 8 位存储器操作指令设定 IAWCTL 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 20-10 非法存储器存取检测控制寄存器（IAWCTL）的格式

地址: F0078H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	端口功能的控制寄存器的保护
0	无效。能读写端口功能的控制寄存器。
1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] PMxx、PUxx、PMC1 注 1、ADPC 注 2

GINT	中断功能的寄存器的保护
0	无效。能读写中断功能的控制寄存器。
1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] IFxx、MKxx、PRxx、EGP0、EGN0

GCSC 注 3	时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的保护
0	无效。能读写时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器。
1	有效。时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] CMC、CSC、OSTS、CKC、PERx、LVIM、LVIS、RPECTL

注 1. 只限于 24 引脚产品。  
2. 不保护 Pxx（端口寄存器）。  
3. 在自编程 / 串行编程时，必须将 GCSC 位置“0”。

20.8 非法存储器存取检测功能

IEC60730 标准要求需要确认 CPU 和中断是否正常运行。  
非法存储器存取检测功能在存取被规定的非法存取检测空间时产生复位。  
非法存取检测空间为图 20-11 中记载为“NG”的范围。

图 20-11 非法存取检测空间

		能否存取		
		读	写	取指令 (执行)
FFFFFH	特殊功能寄存器 (SFR) 256字节	OK	OK	NG
FFF00H FFEFFH				OK
FFEE0H FFEDFH	RAM注		OK	
yyyyyH	保留区		NG	
F4000H F3FFFH	镜像区			
F2000H F1FFFFH	保留区			
F1800H F17FFH	数据闪存			
F1000H F0FFFFH	保留区		OK	
F0800H F07FFH	扩展特殊功能寄存器 (2nd SFR) 2K字节		OK	NG
F0000H EFFFFH			OK	
EF000H EEFFFFH	保留区	NG	NG	
10000H 0FFFFH	代码闪存注	OK	OK	
xxxxxxH				
00000H				

注 代码闪存和 RAM 地址如下所示：

产品	代码闪存	RAM
R7F0C010	16384×8 位（00000H ~ 03FFFH）	1536×8 位（FF900H ~ FFEFFH）

20.8.1 非法存储器存取检测控制寄存器（IAWCTL）

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。  
非法存储器存取检测功能使用 IAWEN 位。  
通过 8 位存储器操作指令设定 IAWCTL 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 20-12 非法存储器存取检测控制寄存器（IAWCTL）的格式

地址: F0078H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN 注	非法存储器存取的检测控制
0	非法存储器存取的检测无效。
1	非法存储器存取的检测有效。

注 IAWEN 位只有写“1”的操作有效，IAWEN 位为“1”后的写“0”的操作无效。

备注 在选项字节的 WDTON 位为“1”（允许看门狗定时器运行）时，即使 IAWEN 位为“0”，非法存储器存取检测功能也有效。

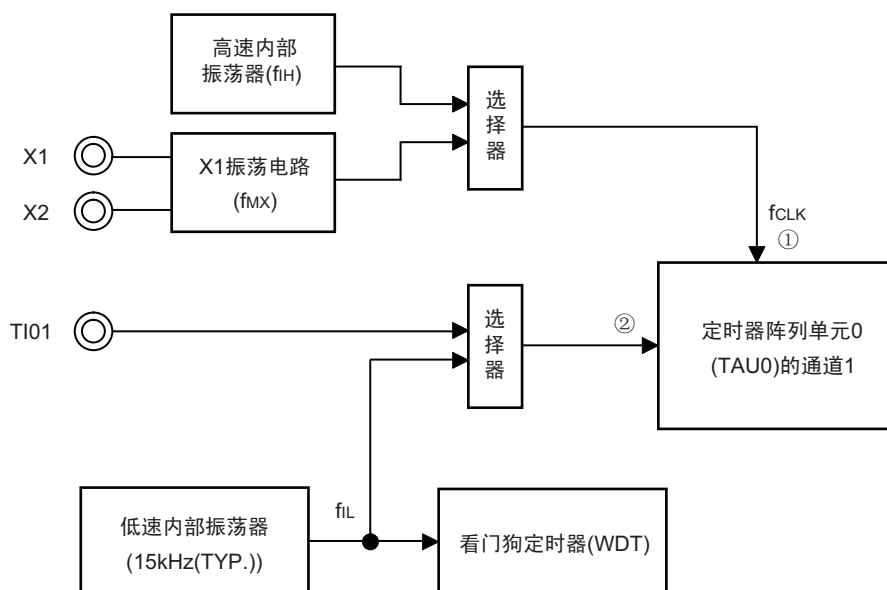
## 20.9 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

频率检测功能可使用 CPU/ 外围硬件的时钟频率 ( $f_{CLK}$ )，并且能通过测量定时器阵列单元 0 (TAU0) 的通道 1 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果某 1 个时钟或者 2 个时钟完全停止振荡，就不能判断 2 个时钟的比率关系。

图 20-13 频率检测功能的结构



### <运行概要>

根据以下条件的脉冲间隔的测量结果，判断时钟频率是否正常。

- 选择高速内部振荡器时钟 ( $f_{IH}$ ) 或者外接 X1 振荡时钟 ( $f_{MX}$ ) 作为 CPU/ 外围硬件时钟 ( $f_{CLK}$ )。
- 选择低速内部振荡器时钟 ( $f_{IL}$ : 15kHz) 作为定时器阵列单元 0 (TAU0) 的通道 1 的定时器输入。

当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。

有关输入脉冲间隔的测量方法，请参照“6.8.3 作为输入脉冲间隔测量的运行”。

20.9.1        定时器输入选择寄存器 0（TIS0）

此寄存器选择定时器阵列单元 0（TAU0）的通道 0 和通道 1 的定时器输入。  
能通过选择低速内部振荡器时钟作为定时器输入并且测量其脉冲，判断低速内部振荡器时钟和定时器运行时钟的比例关系是否正确。  
通过 8 位存储器操作指令设定 TIS0 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 20-14  定时器输入选择寄存器 0（TIS0）的格式

地址：F0074H

复位后：00H

R/W

符号	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	通道 1 使用的定时器输入的选择
0	0	0	定时器输入引脚（TI01）的输入信号
0	0	1	定时器输入引脚（TI01）的输入信号
0	1	0	定时器输入引脚（TI01）的输入信号
0	1	1	定时器输入引脚（TI01）的输入信号
1	0	0	低速内部振荡器时钟（f <sub>IL</sub> ）
上述以外			禁止设定。



## 20.10 A/D 测试功能

IEC60730 标准要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (AN1)、温度传感器的输出电压和内部基准电压进行 A/D 转换，确认 A/D 转换器是否正常运行。有关详细的确认方法，请参照《安全功能 (A/D 测试) 的应用说明》(R01AN0955)。

能通过以下步骤确认模拟多路转换器：

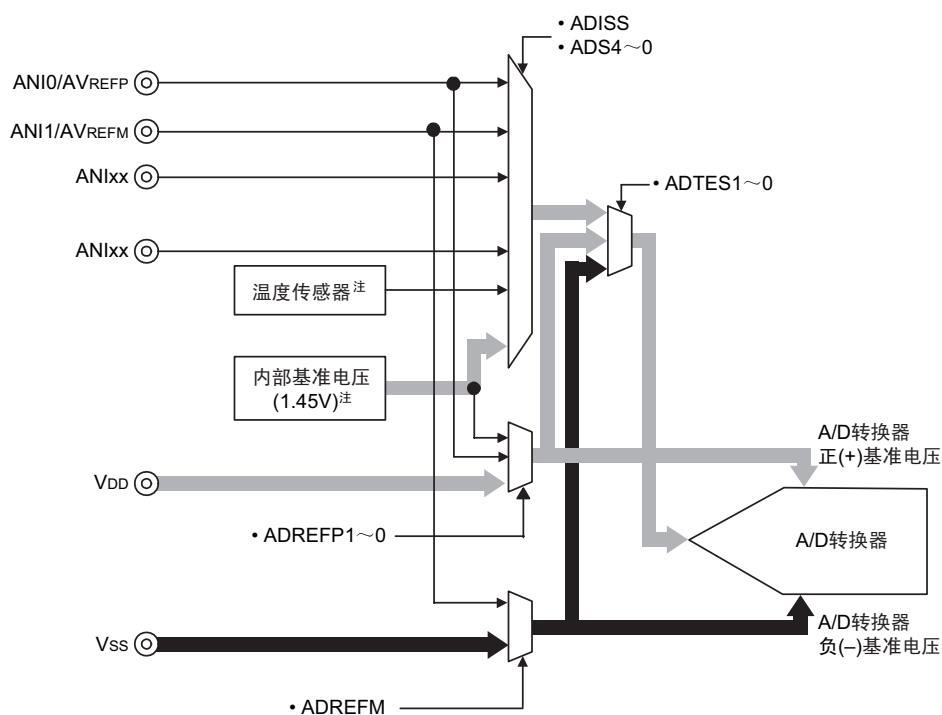
- ① 通过 ADTES 寄存器选择 AN1x 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ② 对 AN1x 引脚进行 A/D 转换 (转换结果 1-1)。
- ③ 通过 ADTES 寄存器选择 A/D 转换器的负 (-) 基准电压作为 A/D 转换对象 (ADTES1、ADTES0=1、0)。
- ④ 对 A/D 转换器的负 (-) 基准电压进行 A/D 转换 (转换结果 2-1)。
- ⑤ 通过 ADTES 寄存器选择 AN1x 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ⑥ 对 AN1x 引脚进行 A/D 转换 (转换结果 1-2)。
- ⑦ 通过 ADTES 寄存器选择 A/D 转换器的正 (+) 基准电压作为 A/D 转换对象 (ADTES1、ADTES0=1、1)。
- ⑧ 对 A/D 转换器的正 (+) 基准电压进行 A/D 转换 (转换结果 2-2)。
- ⑨ 通过 ADTES 寄存器选择 AN1x 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ⑩ 对 AN1x 引脚进行 A/D 转换 (转换结果 1-3)。
- ⑪ 确认“转换结果 1-1”、“转换结果 1-2”和“转换结果 1-3”相同。
- ⑫ 确认“转换结果 2-1”的 A/D 转换结果全部为“0”并且“转换结果 2-2”的 A/D 转换结果全部为“1”。

通过以上步骤，能选择模拟多路转换器以及确认布线没有断线。

备注 1. 在①~⑩的转换过程中，如果模拟输入电压可变，就必须采用其他方法来确认模拟多路转换器。

2. 转换结果含有误差，因此必须在比较转换结果时要适当考虑误差。

图 20-15 A/D 测试功能的结构



注 只有在 HS（高速主）模式中才能选择。

### 20.10.1  A/D 测试寄存器（ADTES）

此寄存器选择 A/D 转换器的正（+）基准电压、负（-）基准电压、模拟输入通道（ANLxx）、温度传感器的输出电压和内部基准电压（1.45V）作为 A/D 转换对象。

当用作 A/D 测试功能时，进行以下的设定：

- 在测量内部0V 电压时，选择AV<sub>REFM</sub>作为 A/D 转换对象。
- 在测量AV<sub>REF</sub>时，选择AV<sub>REFP</sub>作为 A/D 转换对象。

通过 8 位存储器操作指令设定 ADTES 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-16  A/D 测试寄存器（ADTES）的格式

地址：F0013H	复位后：00H		R/W					
符号	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D 转换对象
0	0	ANLxx/ 温度传感器输出注 / 内部基准电压输出（1.45V）注
1	0	AV <sub>REFM</sub>
1	1	AV <sub>REFP</sub>
上述以外		禁止设定。

注        只有在 HS（高速主）模式中才能选择温度传感器输出和内部基准电压输出（1.45V）。

## 20.10.2 模拟输入通道指定寄存器（ADS）

此寄存器指定 A/D 转换的模拟电压的输入通道。

要通过 A/D 测试功能测量 ANI<sub>xx</sub>、温度传感器输出或者内部基准电压（1.45V）时，必须将 A/D 测试寄存器（ADTES）置“00H”。

通过 1 位或者 8 位存储器操作指令设定 ADS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-17 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	ADS4 注 1	0	ADS2	ADS1	ADS0

○选项模式（ADMD=0）

ADISS	ADS4 注 1	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	0	ANI0	P20/ANI0/AV <sub>REFP</sub> 引脚
0	0	0	0	1	ANI1	P21/ANI1/AV <sub>REFM</sub> 引脚
0	0	0	1	0	ANI2	P22/ANI2 引脚
0	0	0	1	1	ANI3	P23/ANI3 引脚
0	0	1	0	0	ANI4 注 2	P24/ANI4 引脚
0	0	1	0	1	ANI5 注 2	P25/ANI5 引脚
0	0	1	1	0	ANI6 注 2	P26/ANI6 引脚
0	0	1	1	1	ANI7	P27/ANI7 引脚
0	1	0	0	0	ANI16 注 1	P03/ANI16 引脚
1	0	0	0	0	—	温度传感器输出注 3
1	0	0	0	1	—	内部基准电压输出（1.45V）注 3
上述以外					禁止设定。	

- 注 1. 只限于 24 引脚产品。  
 2. 只限于 32 引脚产品。  
 3. 只有在 HS（高速主）模式中才能选择。

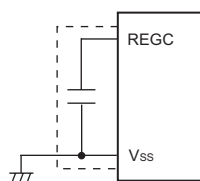
注意 1. 必须将 bit3、5、6 位置“0”。

- 要改写 ADISS 位时，必须在 A/D 转换比较器处于停止状态（A/D 转换器模式寄存器 0（ADM0）的 ADCS=0、ADCE=0）下进行。
- 当将 AV<sub>REFP</sub> 用作 A/D 转换器的正（+）基准电压时，不能选择 ANI0 作为 A/D 转换通道。
- 当将 AV<sub>REFM</sub> 用作 A/D 转换器的负（-）基准电压时，不能选择 ANI1 作为 A/D 转换通道。
- 在将 ADISS 位置“1”的情况下，不能将内部基准电压（1.45V）用作正（+）基准电压。

## 第 21 章 稳压器

### 21.1 稳压器的概要

R7F0C010 内置使器件内部恒压工作的电路。为了使稳压器的输出电压稳定，必须通过电容器（0.47 ~ 1μF）将 REGC 引脚连接 V<sub>SS</sub>。另外，为了稳定内部电压，必须使用特性好的电容器。



注意 必须尽量缩短上图虚线部分的布线。

稳压器的输出电压如表 21-1 所示。

表 21-1 稳压器的输出电压条件

模式	输出电压	条件
LS（低速主）模式	1.8V	—
HS（高速主）模式	1.8V	STOP 模式
	2.1V	上述以外（包含片上调试期间）注

注 在片上调试期间转移到 STOP 模式时，稳压器的输出电压保持 2.1V（不变为 1.8V）。

## 第 22 章 选项字节

### 22.1 选项字节的功能

R7F0C010 的闪存地址 000C0H ~ 000C3H 为选项字节区。

选项字节由用户选项字节（000C0H ~ 000C2H）和片上调试选项字节（000C3H）构成。

在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。

对于没有配置功能的位，不能更改初始值。

#### 22.1.1 用户选项字节（000C0H ~ 000C2H）

##### (1) 000C0H

- 看门狗定时器的运行
  - 允许或者禁止计数器的运行。
  - 在 HALT/STOP 模式中停止或者允许运行。
- 看门狗定时器的上溢时间的设定
- 看门狗定时器的窗口打开期间的设定
- 看门狗定时器的间隔中断的设定
  - 使用或者不使用。

##### (2) 000C1H

- LVD 运行模式的设定
  - 中断 & 复位模式
  - 复位模式
  - 中断模式
  - LVD 为 OFF（使用 RESET 引脚的外部复位输入）。
- LVD 检测电平（ $V_{LVDH}$ 、 $V_{LVDL}$ 、 $V_{LVD}$ ）的设定

## (3) 000C2H

- 闪存运行模式的设定
  - LS（低速主）模式
  - HS（高速主）模式
- 高速内部振荡器的频率设定
  - 从1MHz、4MHz、8MHz、12MHz、16MHz、24MHz、32MHz中选择。

## 22.1.2 片上调试选项字节（000C3H）

- 片上调试运行的控制
  - 禁止或者允许片上调试运行。
- 安全ID验证失败时闪存数据的处理
  - 是否在片上调试安全ID验证失败时擦除闪存的数据。

## 22.2 用户选项字节的格式

图 22-1 用户选项字节（000C0H）的格式

地址：000C0H

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	看门狗定时器的间隔中断的使用 / 不使用
0	不使用间隔中断。
1	当达到上溢时间的 $75\%+1/2f_{IL}$ 时，产生间隔中断。

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间注 2
0	0	禁止设定。
0	1	50%
1	0	75%
1	1	100%

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

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ( $f_{IL}=17.25\text{kHz}(\text{MAX.})$ )
0	0	0	$2^6/f_{IL}$ (3.71ms)
0	0	1	$2^7/f_{IL}$ (7.42ms)
0	1	0	$2^8/f_{IL}$ (14.84ms)
0	1	1	$2^9/f_{IL}$ (29.68ms)
1	0	0	$2^{11}/f_{IL}$ (118.72ms)
上述以外			禁止设定。

WDSTBYON	看门狗定时器的计数器运行控制（HALT/STOP 模式）
0	在 HALT/STOP 模式中，停止计数器的运行注。
1	在 HALT/STOP 模式中，允许计数器的运行。

注 当 WDSTBYON 位为“0”时，与 WINDOW1 位和 WINDOW0 位的值无关，窗口打开期间为 100%。

注意 即使在自编程或者改写数据闪存时看门狗定时器也继续运行，但是在这些处理过程中延迟中断的接受时间。因此，必须考虑此延迟，设定上溢时间和窗口大小。

备注  $f_{IL}$ ：低速内部振荡器的时钟频率

图 22-2 用户选项字节（000C1H）的格式

地址：000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断&复位模式）

检测电压			选项字节的设定值						
V <sub>LVDH</sub>		V <sub>LVDL</sub>	模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	下降	LVIMDS1	LVIMDS0					
2.92V	2.86V	2.75V	1	0	0	1	1	1	0
3.02V	2.96V							0	1
上述以外			禁止设定。						

- LVD 的设定（复位模式）

检测电压		选项字节的设定值						
V <sub>LVD</sub>		模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	LVIMDS1	LVIMDS0					
2.81V	2.75V	1	1	0	1	1	1	1
2.92V	2.86V				1	1	1	0
3.02V	2.96V				1	1	0	1
3.13V	3.06V				0	1	0	0
上述以外		禁止设定。						

- LVD 的设定（中断模式）

检测电压		选项字节的设定值						
V <sub>LVD</sub>		模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	LVIMDS1	LVIMDS0					
2.81V	2.75V	0	1	0	1	1	1	1
2.92V	2.86V				1	1	1	0
3.02V	2.96V				1	1	0	1
3.13V	3.06V				0	1	0	0
上述以外		禁止设定。						

- LVD 的设定（LVD 为 OFF）

检测电压		选项字节的设定值						
V <sub>LVD</sub>		模式设定		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
上升	下降	LVIMDS1	LVIMDS0					
—	—	0/1	1	1	×	×	×	×
上述以外		禁止设定。						

注意 必须给 bit4 写“1”。

备注 1. 有关 LVD 电路的详细内容，请参照“第 19 章 电压检测电路”。

2. 检测电压是 TYP. 值。详细内容请参照“27.6.5 LVD 电路特性”。



图 22-3  用户选项字节（000C2H）的格式

地址：000C2H

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	闪存运行模式的设定		
			工作频率范围	工作电压范围
1	0	LS（低速主）模式	1MHz～8MHz	2.7V～3.6V
1	1	HS（高速主）模式	1MHz～32MHz	2.7V～3.6V
上述以外		禁止设定。		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率
1	0	0	0	32MHz
0	0	0	0	24MHz
1	0	0	1	16MHz
0	0	0	1	12MHz
1	0	1	0	8MHz
1	0	1	1	4MHz
1	1	0	1	1MHz
上述以外				禁止设定。

注意  必须给 bit5～4 写“10B”。

22.3 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 22-4 片上调试选项字节（000C3H）的格式

地址：000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	片上调试运行的控制
0	0	禁止片上调试运行。
0	1	禁止设定。
1	0	允许片上调试运行。 在片上调试安全 ID 验证失败时，擦除闪存的数据。
1	1	允许片上调试运行。 在片上调试安全 ID 验证失败时，不擦除闪存的数据。

- 注意 只有 bit7 和 bit0（OCDENSET 和 OCDERSD）才能指定值。  
必须给 bit6 ~ 1 写“000010B”。
- 备注 bit3 ~ 1 的值在使用片上调试功能时被改写，因此在设定后变为不定值。  
但是，在设定时必须给 bit3 ~ 1 设定初始值（0、1、0）。

22.4 选项字节的设定

用户选项字节和片上调试选项字节除了在源程序中记述以外，还能通过编译程序的链接程序选项进行设定。此时，即使在如下的源程序中有记述也优先链接程序选项的设定。

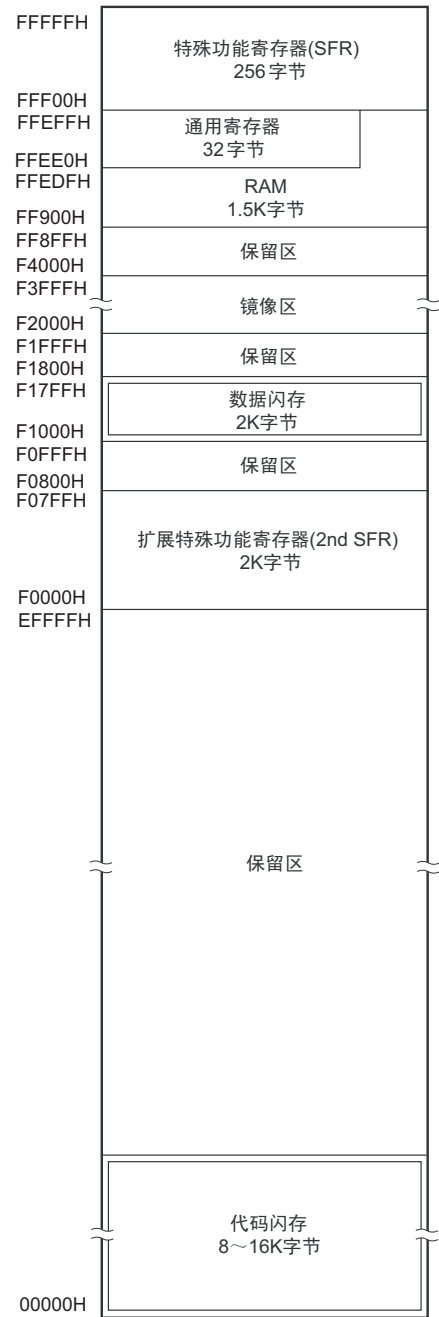
选项字节设定的软件记述例子如下所示。

OPT	CSEG	OPT_BYTE	
	DB	36H	； 不使用看门狗定时器的间隔中断。 ； 允许看门狗定时器运行。 ； 看门狗定时器的窗口打开期间为 50%。 ； 看门狗定时器的上溢时间为 $2^9/f_{IL}$ 。 ； 在 HALT/STOP 模式中，停止看门狗定时器的运行。
	DB	7AH	； $V_{LVDL}$ 选择 2.75V。 ； $V_{LVDH}$ 选择上升 1.77V、下降 2.86V。 ； 选择中断 & 复位模式作为 LVD 运行模式。
	DB	ADH	； 选择 LS（低速主）模式作为闪存运行模式。 ； 选择高速内部振荡器的时钟频率 1MHz。
	DB	85H	； 允许片上调试运行，在安全 ID 验证失败时不擦除闪存的数据。

注意 当通过汇编语言指定选项字节时，CSEG 伪指令的重定位属性名必须使用 OPT\_BYTE。

第 23 章 闪存

R7F0C010 内置在安装于电路板的状态下可进行编程、擦除和重新编程的闪存。闪存有可执行程序的“代码闪存”和数据保存区的“数据闪存”。



闪存的编程方法有以下 3 种：

- 使用闪存编程器进行的编程（参照 23.1）  
能使用专用闪存编程器进行板上或者板外编程。
- 使用外部器件（内置 UART）进行的编程（参照 23.2）  
能通过和外部器件（单片机或者 ASIC）的 UART 通信进行板上编程。
- 自编程（参照 23.7）  
能让用户应用程序利用闪存自编程库进行代码闪存的自改写。

### 23.1 使用闪存编程器的串行编程

能使用以下的专用闪存编程器对 R7F0C010 的内部闪存进行数据编程。

- PG-FP5、FL-PR5
- E1 片上调试仿真器

能使用专用闪存编程器进行板上或者板外编程。

#### (1) 板上编程

在将 R7F0C010 安装到目标系统后改写闪存的内容。必须在目标系统上安装连接专用闪存编程器的连接器。

#### (2) 板外编程

在将 R7F0C010 安装到目标系统前使用专用编程适配器（FA 系列）等进行闪存编程。

备注 FL-PR5 和 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

表 23-1 R7F0C010 和专用闪存编程器的连线表

专用闪存编程器的连接引脚				引脚名	引脚号	
					24 引脚	32 引脚
信号名		输入 / 输出	引脚功能		WQFN （4×4）	LQFP （7×7）
PG-FP5、 FL-PR5	E1 片上 调试仿真器					
—	TOOL0	输入 / 输出	发送 / 接收信号	TOOL0/P40	23	1
SI/RxD	—	输入 / 输出	发送 / 接收信号			
SCK	—	输出	—	—	—	—
CLK	—	输出	—	—	—	—
—	$\overline{\text{RESET}}$	输出	复位信号	$\overline{\text{RESET}}$	24	2
/RESET	—	输出				
FLMD0	—	输出	模式信号	—	—	—
V <sub>DD</sub>		输入 / 输出	V <sub>DD</sub> 电压生成 / 电源监视	V <sub>DD</sub>	6	8
GND		—	接地	V <sub>SS</sub>	5	7
				REGC 注	4	6
EMV <sub>DD</sub>		—	TOOL0 引脚驱动电源	V <sub>DD</sub>	6	8

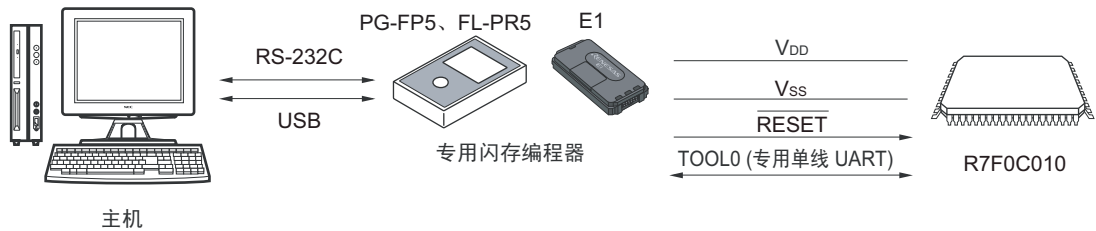
注 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。

备注 在使用闪存编程器进行编程时，此表中未记载的引脚可以开路。

23.1.1 编程环境

R7F0C010 闪存的编程环境如下所示。

图 23-1 闪存的编程环境



需要控制专用闪存编程器的主机。

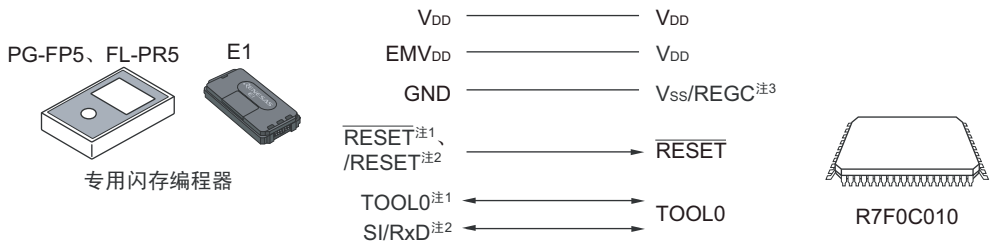
专用闪存编程器和 R7F0C010 的接口使用 TOOL0 引脚，通过专用的单线 UART 进行编程和擦除。

23.1.2 通信方式

专用闪存编程器和 R7F0C010 的通信使用 R7F0C010 的 TOOL0 引脚，通过专用的单线 UART 进行串行通信。

传送速率：1M、500k、250k、115.2kbps

图 23-2 和专用闪存编程器的通信



- 注 1. 这是使用 E1 片上调试仿真器的情况。
- 注 2. 这是使用 PG-FP5 或者 FL-PR5 的情况。
- 注 3. 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。

专用闪存编程器对 R7F0C010 生成以下信号。详细内容请参照 PG-FP5、FL-PR5 或者 E1 片上调试仿真器的手册。

表 23-2 引脚连接一览表

专用闪存编程器				R7F0C010	连接时的处理
信号名		输入 / 输出	引脚功能	引脚名	
PG-FP5、 FL-PR5	E1 片上调试 仿真器				
FLMD0	—	输出	模式信号	—	×
V <sub>DD</sub>		输入 / 输出	V <sub>DD</sub> 电压生成 / 电压监视	V <sub>DD</sub>	◎
GND		—	接地	V <sub>SS</sub> 、REGC 注	◎
EMV <sub>DD</sub>		—	TOOL0 引脚驱动电源	V <sub>DD</sub>	◎
CLK	—	输出	时钟输出	—	×
/RESET	—	输出	复位信号	$\overline{\text{RESET}}$	◎
—	$\overline{\text{RESET}}$	输出			
—	TOOL0	输入 / 输出	发送 / 接收信号	TOOL0	◎
SI/RxD	—	输入 / 输出	发送 / 接收信号		
SCK	—	输出	传送时钟	—	×

注 必须通过电容器（0.47 ~ 1 $\mu$ F）将 REGC 引脚接地。

备注 ◎：必须连接此引脚。

×：不需要连接此引脚。

## 23.2 使用外部器件（内置 UART）的串行编程

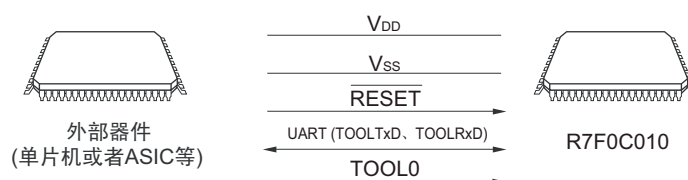
能使用 R7F0C010 和连接 UART 的外部器件（单片机或者 ASIC）对内部闪存进行数据的板上编程。

有关用户闪存编程器的开发，请参照 RL78 微控制器（RL78 协议 A）编程器的应用说明（R01AN0815）。

### 23.2.1 编程环境

R7F0C010 闪存的编程环境如下所示。

图 23-3 闪存的编程环境



外部器件对 R7F0C010 进行板上编程和擦除，但是不能进行板外编程。

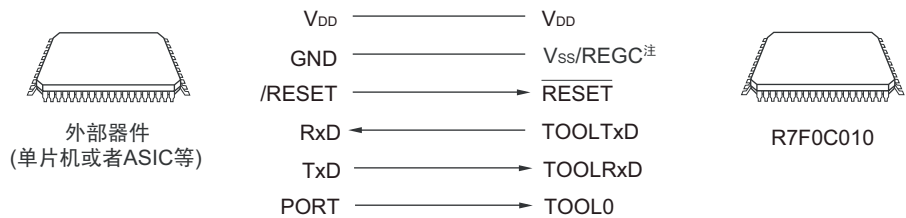


23.2.2 通信方式

外部器件和 R7F0C010 的通信使用 R7F0C010 的 TOOLTxD 引脚和 TOOLRxD 引脚，通过专用的 UART 进行串行通信。

传送速率：1M、500k、250k、115.2kbps

图 23-4 和外部器件的通信



注 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。

外部器件对 R7F0C010 生成以下信号。

表 23-3 引脚连接一览表

外部器件			R7F0C010	连接时的处理
信号名	输入 / 输出	引脚功能	引脚名	
V <sub>DD</sub>	输入 / 输出	V <sub>DD</sub> 电压生成 / 电压监视	V <sub>DD</sub>	◎
GND	—	接地	V <sub>SS</sub> 、REGC <sup>注</sup>	◎
CLK	输出	时钟输出	—	×
RESETOUT	输出	复位信号输出	RESET	◎
RxD	输入	接收信号	TOOLTxD	◎
TxD	输出	发送信号	TOOLRxD	◎
PORT	输出	模式信号	TOOL0	◎
SCK	输出	传送时钟	—	×

注 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚接地。

备注 ◎：必须连接此引脚。

×：不需要连接此引脚。

### 23.3 电路板上的引脚处理

要使用闪存编程器进行板上编程时，必须在目标系统上设置连接专用闪存编程器的连接器，并且在电路板上设置从通常运行模式到闪存编程模式的切换功能。

如果转移到闪存编程模式，闪存编程时未使用的全部引脚就保持刚复位后的相同状态。因此，如果外部器件不允许刚复位后的状态，就需要进行引脚处理。

**备注** 有关闪存编程模式，请参照“23.5.2 闪存编程模式”。

#### 23.3.1 P40/TOOL0 引脚

在闪存编程模式中，必须在外部通过  $1\text{k}\Omega$  电阻将此引脚上拉并且连接到专用闪存编程器。

当此引脚用作端口引脚时，必须按照以下方法使用此引脚。

用作输入引脚：在解除外部复位时的  $t_{\text{HD}}$  期间，不能输入低电平。另外，在通过下拉使用此引脚时，使用的电阻必须至少为  $500\text{k}\Omega$ 。

用作输出引脚：在通过下拉使用此引脚时，使用的电阻必须至少为  $500\text{k}\Omega$ 。

**备注 1.**  $t_{\text{HD}}$ ：这是在进入闪存编程模式时解除外部复位或者内部复位后保持 TOOL0 引脚低电平的时间。请参照“27.9 闪存编程模式的进入时序”。

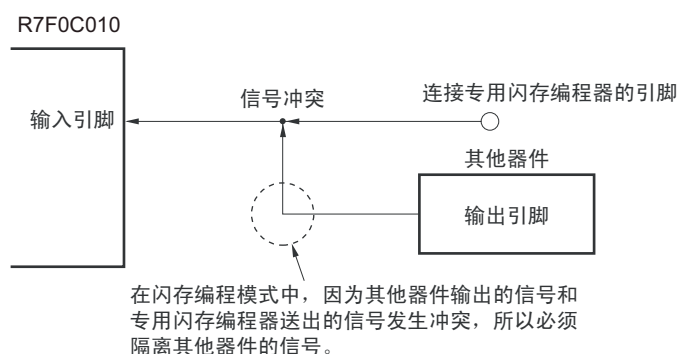
**2.** 因为 R7F0C010 和专用闪存编程器的通信使用单线 UART（TOOL0 引脚），所以不使用 SAU 引脚和 IICA 引脚。

#### 23.3.2 $\overline{\text{RESET}}$ 引脚

如果在电路板上将专用闪存编程器和外部器件的复位信号连接到与复位信号生成电路相连的  $\overline{\text{RESET}}$  引脚，就会发生信号冲突。为了避免此信号冲突，必须隔离与复位信号生成电路的连接。

在闪存编程模式的期间，如果从用户系统输入复位信号，就不能进行正常的编程，因此除了专用闪存编程器和外部器件的复位信号以外，不能输入其他复位信号。

图 23-5 信号冲突（ $\overline{\text{RESET}}$  引脚）



### 23.3.3 端口引脚

如果转移到闪存编程模式，闪存编程时未使用的全部引脚就保持刚复位后的相同状态。因此，如果连接各端口的外部器件不允许刚复位后的端口状态，就需要通过电阻将引脚连接  $V_{DD}$ ，或者通过电阻将引脚连接  $V_{SS}$  进行引脚处理。

### 23.3.4 REGC 引脚

和通常运行模式相同，必须通过特性好的电容器（ $0.47 \sim 1\mu\text{F}$ ）将 REGC 引脚连接 GND。另外，为了稳定内部电压，必须使用特性好的电容器。

### 23.3.5 X1 引脚和 X2 引脚

X1、X2 的连接方法必须和通常运行模式相同。

**备注** 在闪存编程模式中，使用高速内部振荡器时钟（ $f_{IH}$ ）。

### 23.3.6 电源

当使用闪存编程器输出的电源时，必须将  $V_{DD}$  引脚连接闪存编程器的  $V_{DD}$ ，并且将  $V_{SS}$  引脚连接闪存编程器的 GND。

当使用电路板上的电源时，必须按照通常运行模式进行连接。

但是，在使用闪存编程器进行编程时，即使使用电路板上的电源，也必须将  $V_{DD}$  引脚和  $V_{SS}$  引脚分别连接闪存编程器的  $V_{DD}$  和 GND，以便通过闪存编程器监视电压。

23.4 数据闪存

23.4.1 数据闪存的概要

在装载数据闪存的 R7F0C010 中，除了 16K 字节的代码闪存以外，还内置保存数据的 2K 字节的数据闪存。



数据闪存的概要如下。有关数据闪存的改写方法的详细内容，请参照《RL78 Family Data Flash Library User's Manual》。

- 支持使用闪存编程器或者外部器件进行的编程。
- 编程单位为8位。
- 能以1K字节为1块进行块擦除。
- CPU指令的存取只限于字节读（1个时钟+3个等待时钟）。
- 数据闪存是数据专用区域，因此不能用于执行指令（取代码）。
- 在改写数据闪存的过程中，能从代码闪存执行指令（支持后台操作（BGO））。
- 在改写代码闪存的过程中（自编程的情况），禁止存取数据闪存。
- 在解除复位后，数据闪存处于停止状态，因此在使用时设定数据闪存控制寄存器（DFLCTL）。
- 在改写数据闪存的过程中，禁止操作DFLCTL寄存器。
- 在改写数据闪存的过程中，禁止转移到HALT/STOP状态。

注意 1. 在改写数据闪存的过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30μs 后执行数据闪存库。

备注 有关闪存编程模式，请参照“23.7 通过自编程进行的闪存编程”。

## 23.4.2 控制数据闪存的寄存器

### 23.4.2.1 数据闪存控制寄存器（DFLCTL）

此寄存器设定允许或者禁止存取数据闪存。  
通过 1 位或者 8 位存储器操作指令设定 DFLCTL 寄存器。  
在产生复位信号后，此寄存器的值变为“00H”。

图 23-6 数据闪存控制寄存器的格式（DFLCTL）

地址: F0090H	复位后: 00H		R/W					
符号	7	6	5	4	3	2	1	0
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	数据闪存的存取控制
0	禁止存取数据闪存。
1	允许存取数据闪存。

注意 在改写数据闪存的过程中，禁止操作 DFLCTL 寄存器。

### 23.4.3 数据闪存的存取步骤

在解除复位后数据闪存为停止状态。要存取时，必须按照以下步骤初始设定。

- ① 将数据闪存控制寄存器（DFCTL）的bit0（DFLEN）置“1”。
- ② 通过软件定时器等来等待准备时间。  
准备时间因主时钟的各闪存运行模式而不同。  
<各闪存运行模式的准备时间>
  - HS（高速主）模式     : 5 $\mu$ s
  - LS（低速主）模式     : 720ns
- ③ 在等待准备时间后，能存取数据闪存。

注意 1. 在准备时间内，禁止存取数据闪存。

2. 禁止在准备时间内转移到 STOP 模式。要在准备时间内转移到 STOP 模式时，必须在将 DFLEN 位置“0”后执行 STOP 指令。
3. 在改写数据闪存的过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30 $\mu$ s 后执行数据闪存库。

在初始设定后，能通过 CPU 指令读数据闪存，或者通过数据闪存库读或改写数据闪存。

但是，如果在存取数据闪存时 DMA 控制器运行，就必须按照以下的任意一个步骤进行处理：

(A) 保留或者强制结束DMA传送。

在读数据闪存前，必须保留正在使用的全部通道的DMA传送。但是，必须在将 DWAITn 位置“1”后至少间隔 3 个时钟（ $f_{CLK}$ ），然后读数据闪存。在读数据闪存后，必须将 DWAITn 位置“0”，解除传送的保留。

或者，在读数据闪存前，必须按照“13.5.5 通过软件强制结束”的步骤强制结束DMA传送，在读数据闪存后重新开始DMA传送。

(B) 使用库进行数据闪存的存取。

必须使用最新的数据闪存库进行数据闪存的存取。

(C) 插入NOP。

必须在数据闪存的读指令之前插入NOP指令。

<例>

```
MOVW    HL, !addr16    ; 读RAM。
NOP                      ; 在数据闪存的读指令之前插入NOP指令。
MOV      A, [DE]        ; 读数据闪存。
```

但是，当使用C语言等高级语言时，编译程序可能对1个代码生成2条指令。此时，因为不在数据闪存的读指令之前插入NOP指令，所以必须通过(A)或者(B)读数据闪存。

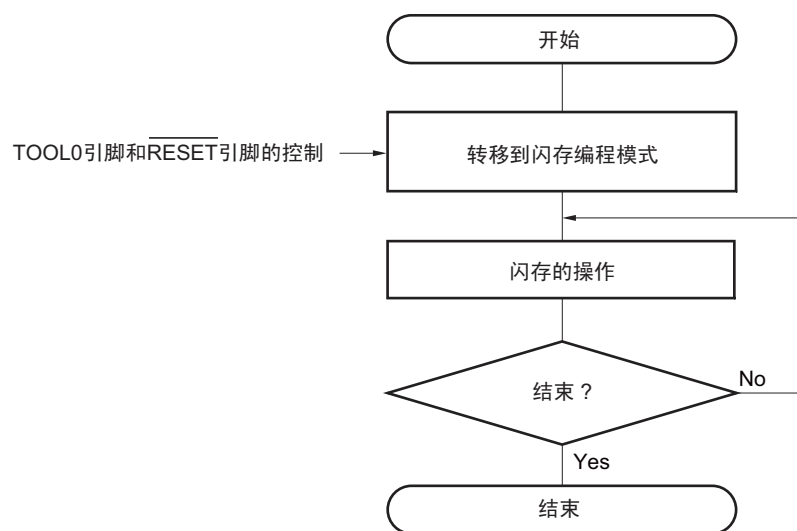
备注  $f_{CLK}$ : CPU/ 外围硬件的时钟频率

## 23.5 编程方法

### 23.5.1 闪存的控制

通过串行编程进行代码闪存改写的流程如下所示。

图 23-7 闪存的操作步骤



### 23.5.2 闪存编程模式

在改写闪存的内容时，必须将 R7F0C010 设定为闪存编程模式。要转移到闪存编程模式时，必须按照以下方法进行。

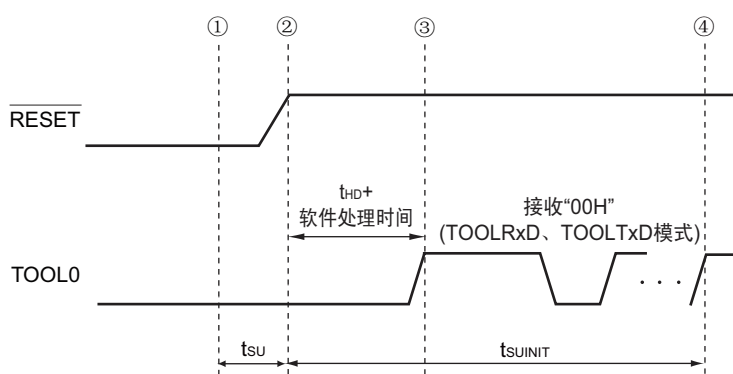
<使用专用闪存编程器进行编程的情况>

通过和专用闪存编程器的通信，自动转移到闪存编程模式。

<使用外部器件进行编程的情况>

在将 TOOL0 引脚置为低电平后解除复位。在解除复位到 1ms+软件处理结束为止的时间内，必须保持 TOOL0 引脚的低电平，然后通过 UART 通信从外部器件发送数据“00H”。必须在解除复位后的 100ms 之内完成 UART 初始设定的通信。

图 23-8 闪存编程模式的进入



- ① 给 TOOL0 引脚输入低电平。
- ② 解除外部复位（在此之前需要解除 POR 和 LVD 的复位）。
- ③ 解除 TOOL0 引脚的低电平。
- ④ 通过 UART 接收进入闪存编程模式，完成波特率的设定。

备注  $t_{SUINIT}$ ：在此区间，必须在解除复位后的 100ms 之内完成初始设定的通信。

$t_{SU}$ ：这是从将 TOOL0 引脚置为低电平到解除外部复位为止的时间。

$t_{HD}$ ：这是在解除外部复位或者内部复位后保持 TOOL0 引脚低电平的时间（软件处理时间除外）。

详细内容请参照“27.9 闪存编程模式的进入时序”。



表 23-4 解除复位时的 TOOL0 引脚和运行模式的关系

TOOL0	运行模式
V <sub>DD</sub>	通常运行模式
0V	闪存编程模式

闪存编程模式有宽电压模式和全速模式共 2 种编程模式。根据编程时提供给单片机的电源电压值以及进入闪存编程模式时的用户选项字节的设定信息，决定模式的选择。

在使用专用闪存编程器进行串行编程时，通过 GUI 进行的电压设定自动选择模式。

表 23-5 编程模式和可进行编程、擦除或者校验的电压

模式	可进行编程、擦除或者校验的电压	编程频率
宽电压模式	2.7V ~ 3.6V	8MHz(MAX.)
	2.7V ~ 3.6V	16MHz(MAX.)
	2.7V ~ 3.6V	32MHz(MAX.)
全速模式 <sup>注</sup>	2.7V ~ 3.6V	16MHz(MAX.)
	2.7V ~ 3.6V	32MHz(MAX.)

注 只有在选项字节 000C2H 的 CMODE1 位和 CMODE0 位都为“1”时才能设定。

必须根据进行编程的电压范围来设定相应的模式。在使用专用闪存编程器时，通过 GUI 进行的电压设定自动选择模式。

备注 1. 在同时使用宽电压模式和全速模式的情况下进行编程、擦除或者校验时，没有任何限制事项。

2. 有关通信命令的详细内容，请参照“23.5.4 通信命令”。

### 23.5.3 通信方式

RL78 微控制器的通信方式如下所示。

表 23-6 通信方式

通信方式	Standard 设定 <sup>注 1</sup>				使用的引脚
	Port	Speed <sup>注 2</sup>	Frequency	Multiply Rate	
单线 UART (使用闪存编程器 或者外部器件)	UART	115200bps、 250000bps、 500000bps、 1Mbps	—	—	TOOL0
专用 UART (使用外部器件)	UART	115200bps、 250000bps、 500000bps、 1Mbps	—	—	TOOLTxD、 TOOLRx

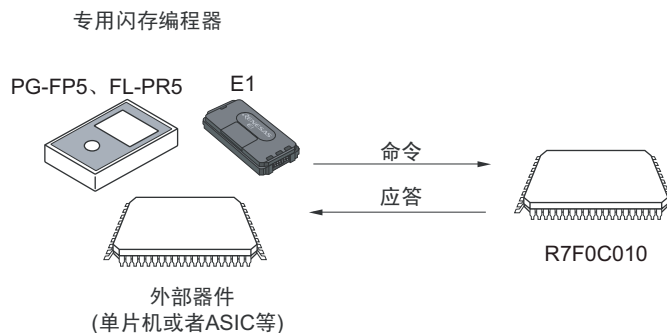
注 1. 这是与闪存编程器 GUI Standard 设定相关的设定项目。

2. 除了波特率误差以外，信号波形的失真等还会影响 UART 通信，因此必须在评估后使用。

### 23.5.4 通信命令

R7F0C010 和专用闪存编程器或者外部器件通过命令进行通信。从专用闪存编程器或者外部器件发送到 R7F0C010 的信号称为“命令”，从 R7F0C010 发送到专用闪存编程器或者外部器件的信号称为“应答”。

图 23-9 通信命令



R7F0C010 的闪存控制命令如下所示。这些命令都来自编程器或者外部器件，R7F0C010 进行对应命令的处理。

表 23-7 闪存控制命令

分类	命令名称	功能
校验	Verify	将指定闪存区的内容和从编程器送来的数据进行比较。
擦除	Block Erase	擦除指定的闪存区。
空白检查	Block Blank Check	检查指定块的闪存擦除状态。
编程	Programming	将数据写到指定的闪存区注。
信息取得	Silicon Signature	取得 R7F0C010 的信息（例如：产品名、闪存结构、用于编程的固件版本等）。
	Checksum	取得指定区域的校验和。
安全	Security Set	设定安全信息。
	Security Get	取得安全信息。
	Security Release	解除禁止编程的设定。
其他	Reset	用于通信的同步检测。
	Baud Rate Set	设定选择 UART 时的波特率。

注 必须确认编程区中还没有进行数据编程。因为在设定禁止块擦除后无法进行擦除，所以在数据没有被擦除时，不能进行数据编程。

对于专用闪存编程器或者外部器件发行的命令，R7F0C010 返回应答。R7F0C010 发送的应答名称如下所示。

表 23-8 应答名称

应答名称	功能
ACK	命令 / 数据等的应答
NAK	非法的命令 / 数据等的应答

### 23.5.5 特征数据的说明

如果执行“Silicon Signature”命令，就能获取 R7F0C010 的信息（例如：产品名、闪存结构、用于编程的固件版本等）。

特征数据一览表和特征数据的例子分别如表 23-9 和表 23-10 所示。

表 23-9 特征数据一览表

字段名	内容	发送字节数
器件代码	分配给器件的序列号	3 字节
器件名	器件名（ASCII 码）	10 字节
代码闪存区的结束地址	代码闪存区的结束地址 （从地址的低位开始发送。 例：00000H ~ 0FFFFH（64KB）→FFH、FFH、00H）	3 字节
数据闪存区的结束地址	数据闪存区的结束地址 （从地址的低位开始发送。 例：F1000H ~ F1FFFH（4KB）→FFH、1FH、0FH）	3 字节
固件版本	用于编程的固件版本信息 （从版本的高位开始发送。例：Ver.1.23→01H、02H、03H）	3 字节

表 23-10 特征数据的例子

字段名	内容	发送字节数	数据（十六进制）
器件代码	RL78 协议 A	3 字节	10 00 06
器件名	R7F0C0107	10 字节	52 = “R” 37 = “7” 46 = “F” 30 = “0” 43 = “C” 30 = “0” 31 = “1” 30 = “0” 37 = “7” 20 = “ ”
代码闪存区的结束地址	代码闪存区 00000H ~ 03FFFH（16KB）	3 字节	FF 3F 00
数据闪存区的结束地址	数据闪存区 F1000H ~ F17FFFH（2KB）	3 字节	FF 17 0F
固件版本	Ver.1.23	3 字节	01 02 03

## 23.6 安全设定

R7F0C010 支持安全功能，此安全功能禁止改写闪存中的用户程序，能防止他人更改程序。能通过使用 Security Set 命令进行以下的操作。

- 禁止块擦除  
在进行板上或者板外编程时，禁止执行闪存中的块擦除命令。但是，在自编程时能进行块擦除。
- 禁止编程  
在进行板上或者板外编程时，禁止对闪存中的全部块执行编程命令。但是，在自编程时能进行编程。  
在进行安全功能的设定后，Security Release 命令的解除在复位后有效。

出厂时的初始状态为允许块擦除和编程。能在板上或者板外编程以及自编程时进行安全功能的设定，并且能同时组合使用各种安全功能的设定。

R7F0C010 的安全功能有效和擦除、编程命令的关系如表 23-11 所示。

**注意** 闪存编程器的安全功能不支持自编程。

**备注** 要在自编程过程中禁止编程和擦除时，使用闪存屏蔽窗口功能（详细内容请参照 23.7.1）。

表 23-11 安全功能有效和命令的关系

## (1) 板上或者板外编程

有效的安全	执行的命令	
	块擦除	编程
禁止块擦除。	不能进行块擦除。	能进行编程注。
禁止编程。	能进行块擦除。	不能进行编程。

注 必须确认编程区中还没有进行数据编程。因为在设定禁止块擦除后无法进行擦除，所以在数据没有被擦除时，不能进行数据编程。

## (2) 自编程

有效的安全	执行的命令	
	块擦除	编程
禁止块擦除。	能进行块擦除。	能进行编程。
禁止编程。		

备注 要在自编程过程中禁止编程和擦除时，使用闪存屏蔽窗口功能（详细内容请参照 23.7.1）。

表 23-12 各编程模式中的安全设定方法

## (1) 板上或者板外编程

安全	安全设定方法	如何使安全设定无效
禁止块擦除。	通过专用闪存编程器的GUI等进行设定。	在设定后无法置为无效。
禁止编程。		执行安全解除命令。

注意 在没有设定为“禁止块擦除”并且代码闪存区和数据闪存区为空白时，安全解除命令有效。

## 23.7 通过自编程进行的闪存编程

R7F0C010 支持自编程功能，能通过用户程序改写闪存。因为此功能可让用户应用程序利用 R7F0C010 的自编程库来改写闪存，所以能在现场进行程序升级等。

- 注意 1. 为了在自编程过程中禁止中断，必须和通常运行模式中一样，在通过 DI 指令将 IE 标志清“0”的状态下执行自编程库。在允许中断的情况下，必须在通过 EI 指令将 IE 标志置“1”的状态下将接受中断的中断屏蔽标志清“0”，然后执行自编程库。
2. 当允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 时，必须在改写前对“所用 RAM 区+10 字节”的区域进行初始化。
3. 在自编程过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行 (HIOSTOP=0)，并且在经过 30μs 后执行自编程库。

- 备注 1. 有关自编程功能和 R7F0C010 自编程库的详细内容，请参照《RL78 Microcontroller Flash Self Programming Library Type01 User's Manual》(R01AN0350E)。
2. 有关自编程的执行处理时间，请参照闪存自编程库工具附属的使用时的注意点。

和使用闪存编程器进行的编程相同，根据可进行编程、擦除或者校验的不同电压范围，自编程功能有以下 2 种编程模式。

表 23-13 编程模式和可进行编程、擦除或者校验的电压

模式	可进行编程、擦除或者校验的电压	编程频率
宽电压模式	2.7V ~ 3.6V	32MHz(MAX.)
全速模式 <sup>注</sup>	2.7V ~ 3.6V	32MHz(MAX.)

注 只有在选项字节 000C2H 的 CMODE1 位和 CMODE0 位都为“1”时才能设定。

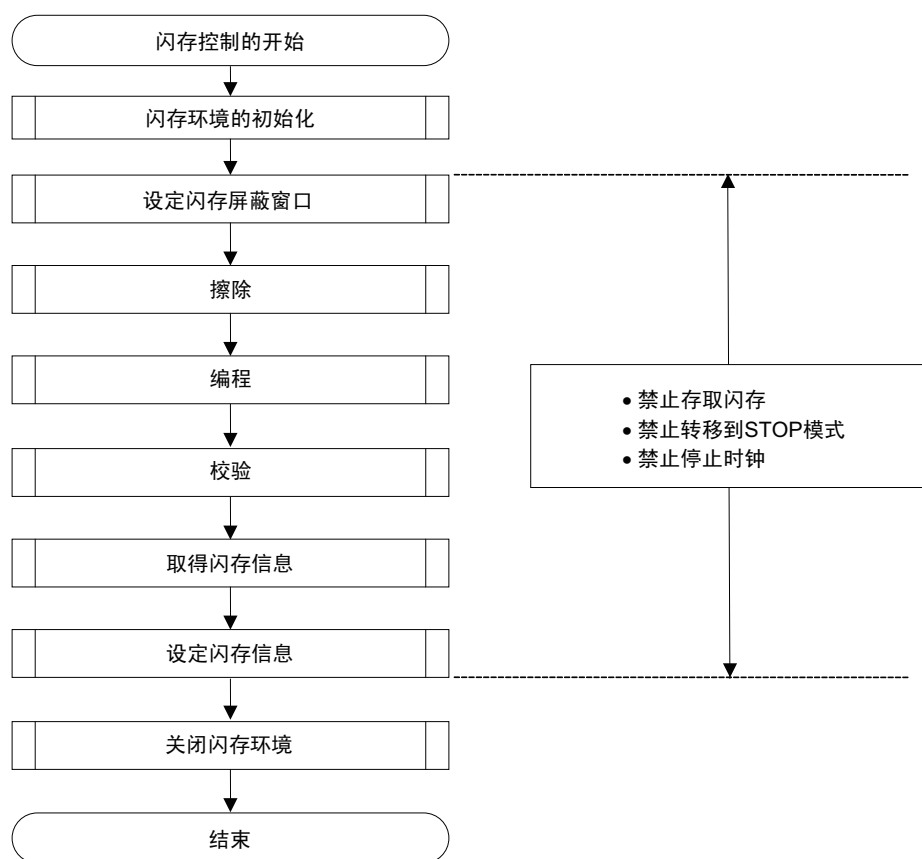
必须根据进行编程的电压范围来设定相应的模式。在执行本公司提供的自编程库的“FSL\_Init”函数时，如果参数“fsl\_flash\_voltage\_u08”不为“00H”，就设定为宽电压模式；如果参数“fsl\_flash\_voltage\_u08”为“00H”，就设定为全速模式。

备注 1. 在同时使用宽电压模式和全速模式的情况下进行编程、擦除或者校验时，没有任何限制事项。

2. 有关自编程功能和 R7F0C010 自编程库的详细内容，请参照《RL78 Microcontroller Flash Self Programming Library Type01 User's Manual》(R01AN0350E)。

利用自编程库改写闪存的流程如下所示。

图 23-10 自编程流程（改写闪存）





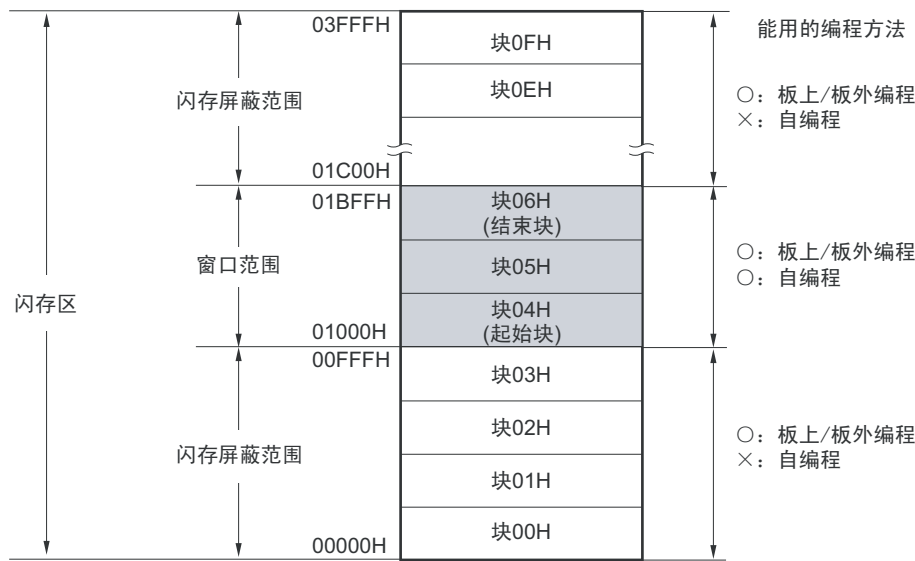
23.7.1 闪存屏蔽窗口功能

闪存屏蔽窗口功能作为自编程的一个安全功能，只在自编程时禁止对指定的窗口范围以外的区域进行编程和擦除。

能通过指定起始块和结束块来设定窗口范围。只能在板上或者板外编程时设定或者更改窗口范围的指定。

在自编程时，禁止对窗口范围以外的区域进行编程和擦除。但是，在进行板上或者板外编程时，也能对指定窗口范围以外的区域进行编程和擦除。

图 23-11 闪存屏蔽窗口的设定例子  
(对象器件: R7F0C010, 起始块: 04H, 结束块: 06H)



注意 只能对代码闪存设定闪存屏蔽窗口（不支持数据闪存）。

表 23-14 闪存屏蔽窗口功能的设定 / 更改方法和命令的关系

编程条件	窗口范围的设定 / 更改方法	执行的命令	
		块擦除	编程
板上 / 板外编程	通过专用闪存编程器的 GUI 等指定窗口的起始块和结束块。	也能擦除窗口范围以外的块。	也能对窗口范围以外的区域进行编程。

备注 有关要在板上或者板外编程时禁止编程或者擦除的情况，请参照“23.6 安全设定”。

## 23.8 使用 PG-FP5 时的各命令处理时间（参考值）

在将 PG-FP5 用作专用闪存编程器时，各命令处理时间（参考值）如下所示：

表 23-15 使用 PG-FP5 时的各命令处理时间（参考值）

PG-FP5 的命令	Port: TOOL0 (UART)
	Speed: 1Mbps
	16K 字节
擦除	1s
编程	1.5s
校验	1.5s
擦除后的编程	1.5s

备注 命令处理时间（参考值）是 TYP. 值，条件如下：

Port : TOOL0（单线 UART）

Speed: 1000000bps

Mode : 全速模式（闪存运行模式：HS（高速主）模式）

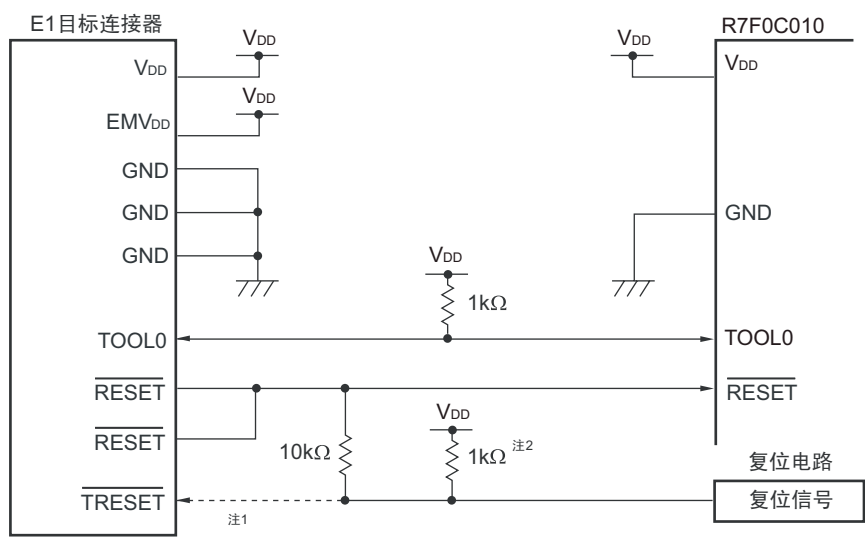
第 24 章 片上调试功能

24.1 E1 片上调试仿真器和 R7F0C010 的连接

当 R7F0C010 通过对应片上调试的 E1 片上调试仿真器与主机进行通信时，使用  $V_{DD}$ 、 $\overline{\text{RESET}}$ 、 $\text{TOOL0}$ 、 $V_{SS}$  引脚。通过使用  $\text{TOOL0}$  引脚的单线 UART 进行串行通信。

注意 R7F0C010 内置用于开发和评估的片上调试功能。如果使用片上调试功能，就可能会超过闪存的保证改写次数而无法保证产品的可靠性，因此批量生产的产品不能使用片上调试功能。对于使用片上调试功能的产品，不作为投诉受理对象。

图 24-1 E1 片上调试仿真器和 R7F0C010 的连接例子



- 注 1. 在闪存编程时，虚线部分不需要连接。
2. 如果目标系统的复位电路没有缓冲器而只通过电阻和电容器生成复位信号，就不需要此上拉电阻。

注意 这是假设复位信号的输出为 N 沟道漏极开路缓冲器（输出电阻不大于  $100\Omega$ ）的电路例子。

## 24.2 片上调试安全 ID

为了防止他人读取存储器的内容，R7F0C010 在闪存的 000C3H 中提供了片上调试运行控制位（参照“第 22 章 选项字节”），并且在 000C4H ~ 000CDH 中提供了片上调试安全 ID 设定区。

表 24-1 片上调试安全 ID

地址	片上调试安全 ID 码
000C4H ~ 000CDH	任意 10 字节的 ID 码
010C4H ~ 010CDH	

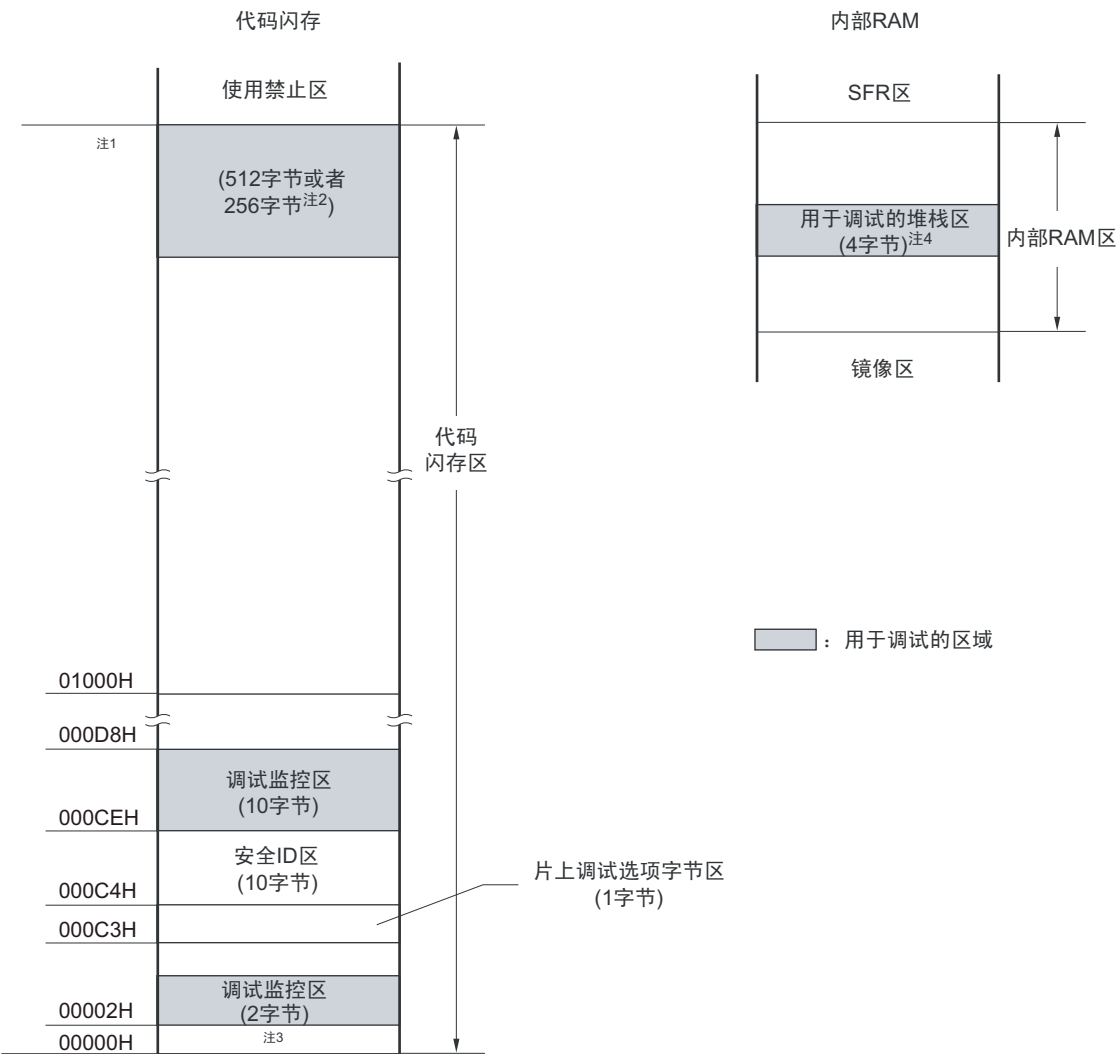
## 24.3 用户资源的确保

为了实现 R7F0C010 与 E1 片上调试仿真器的通信或者实现各种调试功能，需要事先确保存储空间。当使用本公司的汇编程序和编译程序时，也能通过链接程序的选项进行设定。

### (1) 存储空间的确保

图 24-2 的灰色部分表示的区域保存用于调试的监控程序，是不能保存用户程序或者数据的空间。要使用片上调试功能时，需要确保不使用此空间的区域，而且不能在用户程序中改写此空间。

图 24-2 保存调试监控程序的存储空间



注 1. 代码闪存的地址如下所示。

产品名	注 1 的地址
R7F0C010	03FFFH

2. 在不使用实时 RAM 监控（RRM）功能和动态存储器修改（DMM）功能时，为 256 字节。
3. 在调试时，复位向量被改写为监控程序的分配地址。
4. 此区域分配在堆栈区之后，因此用于调试的堆栈区地址随着堆栈的增减而变。即，对于使用的堆栈区，额外占用 4 字节。在自编程时，额外占用 12 字节。

第 25 章 十进制校正（BCD）电路

25.1 十进制校正电路的功能

能以 BCD 码（二 - 十进制）计算 BCD 码（二 - 十进制）和 BCD 码（二 - 十进制）相加减的结果。  
在执行以 A 寄存器为操作数的加减运算指令后，通过加减 BCD 校正结果寄存器（BCDADJ）的值，计算十进制的校正运算结果。

25.2 十进制校正电路使用的寄存器

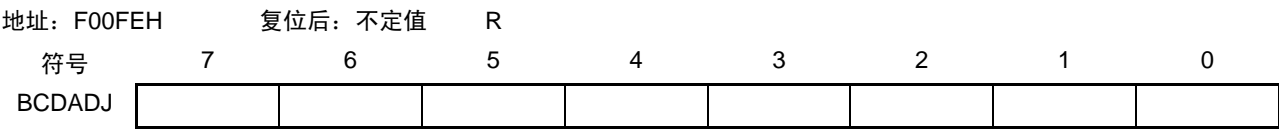
十进制校正电路使用以下寄存器：

- BCD 校正结果寄存器（BCDADJ）

25.2.1 BCD 校正结果寄存器（BCDADJ）

通过以 A 寄存器为操作数的加减运算指令，将用于计算 BCD 码加减运算结果的校正值保存到 BCDADJ 寄存器。  
BCDADJ 寄存器的读取值因读时的 A 寄存器、CY 标志和 AC 标志的值而变。  
通过 8 位存储器操作指令读 BCDADJ 寄存器。  
在产生复位信号后，此寄存器的值变为不定值。

图 25-1 BCD 校正结果寄存器（BCDADJ）的格式



### 25.3 十进制校正电路的运行

十进制校正电路的基本运行如下所示。

#### (1) 加法 以 BCD 码值计算 BCD 码值加上 BCD 码值的结果

- ① 将要相加的BCD码值（被加数）保存到A寄存器。
- ② 以二进制将A寄存器的值和第2个操作数的值（另一个要相加的BCD码值，加数）相加，二进制的运算结果保存在A寄存器，校正值保存在BCD校正结果寄存器（BCDADJ）。
- ③ 以二进制将A寄存器的值（二进制的加法运算结果）和BCDADJ寄存器的值（校正值）相加进行十进制校正运算，校正结果保存在A寄存器和CY标志。

**注意** BCDADJ寄存器的读取值因读时的A寄存器、CY标志和AC标志的值而变。因此，必须在②的指令后不执行其他指令而执行③的指令。要在允许中断的状态下进行BCD校正时，需要在中断函数中进行A寄存器的压栈和退栈。通过RETI指令恢复PSW（CY标志、AC标志）。

例子如下所示。

例 1 99+89=188

指令	A 寄存器	CY 标志	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	—

例 2 85+15=100

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #85H ; ①	85H	—	—	—
ADD A, #15H ; ②	9AH	0	0	66H
ADD A, !BCDADJ ; ③	00H	1	1	—

例 3 80+80=160

指令	A 寄存器	CY 标志	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) 减法 以 BCD 码值计算 BCD 码值减去 BCD 码值的结果

- ① 将要减去的BCD码值（被减数）保存到A寄存器。
- ② 以二进制将A寄存器的值减去第2个操作数的值（要减去的BCD码值，减数），二进制的运算结果保存在A寄存器，校正值保存在BCD校正结果寄存器（BCDADJ）。
- ③ 以二进制将A寄存器的值（二进制的减法运算结果）减去BCDADJ寄存器的值（校正值）进行十进制校正运算，校正结果保存在A寄存器和CY标志。

注意 BCDADJ寄存器的读取值因读时的A寄存器、CY标志和AC标志的值而变。因此，必须在②的指令后不执行其他指令而执行③的指令。要在允许中断的状态下进行BCD校正时，需要在中断函数中进行A寄存器的压栈和退栈。通过RETI指令恢复PSW（CY标志、AC标志）。

例子如下所示。

例 91-52=39

指令				A 寄存器	CY 标志	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 26 章 指令集的概要

RL78 微控制器的指令集表示在一览表中。有关各指令的详细动作和机器码（指令码），请参照《RL78 Family User's Manual Software》。

26.1 凡例

26.1.1 操作数的表现形式和记述方法

根据指令操作数的表现形式所对应的记述方法，在各指令的操作数栏中记述了操作数（详细内容请参照汇编程序规格）。在记述方法中存在多个内容时，选择其中一个。大写字母以及 #、!、!!、\$、\$!、[] 和 ES: 的符号为关键字，应按照原样记述。符号的说明如下：

- # : 指定立即数。
- ! : 指定 16 位绝对地址。
- !! : 指定 20 位绝对地址。
- \$ : 指定 8 位相对地址。
- \$! : 指定 16 位相对地址。
- [] : 指定间接地址。
- ES: : 指定扩展地址。

对于立即数，记述适当的数值或者标号。在使用标号时，必须记述 #、!、!!、\$、\$!、[] 和 ES: 的符号。  
对于操作数的寄存器记述形式 r 和 rp，能使用功能名（X、A、C 等）或者绝对名（表 26-1 中括号内的名称，如 R0、R1、R2 等）进行记述。

表 26-1 操作数的表现形式和记述方法

表现形式	记述方法	
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	特殊功能寄存器的符号（SFR 符号）FFF00H ~ FFFFFH	
sfrp	特殊功能寄存器的符号（可进行 16 位操作的 SFR 符号，只限于偶数地址注）FFF00H ~ FFFFFH	
saddr	FFE20H ~ FFF1FH	立即数或者标号
saddrp	FFE20H ~ FFF1FH	立即数或者标号（只限于偶数地址注）
addr20	00000H ~ FFFFFH	立即数或者标号
addr16	0000H ~ FFFFH	立即数或者标号（在 16 位数据时，只限于偶数地址注）
addr5	0080H ~ 00BFH	立即数或者标号（只限于偶数地址）
word	16 位立即数或者标号	
byte	8 位立即数或者标号	
bit	3 位立即数或者标号	
RBn	RB0 ~ RB3	

注 当指定奇数地址时，bit0 为“0”。

备注 能用符号将特殊功能寄存器记述为操作数 sfr。有关特殊功能寄存器的符号，请参照“表 3-5 特殊功能寄存器（SFR）一览表”。

能用符号将扩展特殊功能寄存器记述为操作数!addr16。有关扩展特殊功能寄存器的符号，请参照“表 3-6 扩展特殊功能寄存器（2nd SFR）一览表”。

## 26.1.2 操作栏的说明

在各指令的操作栏中，用以下符号表示指令执行时的动作。

表 26-2 操作栏的符号

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
()	() 内的地址或者寄存器的内容所示的存储器内容
$X_H$ 、 $X_L$	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： $X_S$ (bit19 ~ 16)、 $X_H$ (bit15 ~ 8)、 $X_L$ (bit7 ~ 0)
$\wedge$	逻辑与 (AND)
$\vee$	逻辑或者 (OR)
$\nabla$	异或 (exclusive OR)
—	数据取反
addr5	16 位立即数 (只限于偶数地址 0080H ~ 00BFH)
addr16	16 位立即数
addr20	20 位立即数
jdisp8	带符号的 8 位数据 (位移量)
jdisp16	带符号的 16 位数据 (位移量)

### 26.1.3 标志栏的说明

在各指令的标志栏中，用以下符号表示指令执行时的标志变化。

表 26-3 标志栏的符号

符号	标志的变化
(空白)	无变化。
0	清“0”。
1	置“1”。
×	根据结果置位或者复位。
R	恢复以前保存的值。

### 26.1.4 PREFIX 指令

ES: 所示的指令以 PREFIX 指令码为前缀，将能存取的数据区从 F0000H ~ FFFFFH 的 64K 字节空间扩展为附加 ES 寄存器值的 00000H ~ FFFFFH 的 1M 字节空间。将 PREFIX 指令码置于对象指令的前头，只有紧接在 PREFIX 指令码后的 1 条指令作为附加了 ES 寄存器值的地址进行执行。

在 PREFIX 指令码和紧随其后的 1 条指令之间，不接受中断和 DTC 传送。

表 26-4 PREFIX 指令码的使用例子

指令	指令码				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES:[HL]	11H	8BH	—	—	—

**注意** 在执行 PREFIX 指令前，必须事先通过 MOV ES, A 等设定 ES 寄存器的值。

## 26.2 操作一览表

表 26-5 操作一览表 (1/18)

指令群	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 传 送	MOV	r, #byte	2	1	—	r←byte			
		PSW, #byte	3	3	—	PSW←byte	×	×	×
		CS, #byte	3	1	—	CS←byte			
		ES, #byte	2	1	—	ES←byte			
		!addr16, #byte	4	1	—	(addr16)←byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16)←byte			
		saddr, #byte	3	1	—	(saddr)←byte			
		sfr, #byte	3	1	—	sfr←byte			
		[DE+byte], #byte	3	1	—	(DE+byte)←byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE)+byte)←byte			
		[HL+byte], #byte	3	1	—	(HL+byte)←byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL)+byte)←byte			
		[SP+byte], #byte	3	1	—	(SP+byte)←byte			
		word[B], #byte	4	1	—	(B+word)←byte			
		ES:word[B], #byte	5	2	—	((ES, B)+word)←byte			
		word[C], #byte	4	1	—	(C+word)←byte			
		ES:word[C], #byte	5	2	—	((ES, C)+word)←byte			
		word[BC], #byte	4	1	—	(BC+word)←byte			
		ES:word[BC], #byte	5	2	—	((ES, BC)+word)←byte			
		A, r 注 3	1	1	—	A←r			
		r, A 注 3	1	1	—	r←A			
		A, PSW	2	1	—	A←PSW			
		PSW, A	2	3	—	PSW←A	×	×	×
		A, CS	2	1	—	A←CS			
		CS, A	2	1	—	CS←A			
		A, ES	2	1	—	A←ES			
		ES, A	2	1	—	ES←A			
		A, !addr16	3	1	4	A←(addr16)			
		A, ES:!addr16	4	2	5	A←(ES, addr16)			
		!addr16, A	3	1	—	(addr16)←A			
		ES:!addr16, A	4	2	—	(ES, addr16)←A			
		A, saddr	2	1	—	A←(saddr)			
		saddr, A	2	1	—	(saddr)←A			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. r=A 除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (2/18)

指令集	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 传 送	MOV	A, sfr	2	1	—	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	—	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow ([DE])$			
		[DE], A	1	1	—	$([DE]) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	—	$(ES, DE) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow ([HL])$			
		[HL], A	1	1	—	$([HL]) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	—	$(ES, HL) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow ([DE] + \text{byte})$			
		[DE+byte], A	2	1	—	$([DE] + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((ES, DE) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow ([HL] + \text{byte})$			
		[HL+byte], A	2	1	—	$([HL] + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow ([SP] + \text{byte})$			
		[SP+byte], A	2	1	—	$([SP] + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	—	$(B + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	—	$((ES, B) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	—	$(C + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	—	$((ES, C) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	—	$(BC + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	—	$((ES, BC) + \text{word}) \leftarrow A$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (3/18)

指令集	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 传 送	MOV	A, [HL+B]	2	1	4	$A \leftarrow (HL+B)$			
		[HL+B], A	2	1	—	$(HL+B) \leftarrow A$			
		A, ES:[HL+B]	3	2	5	$A \leftarrow ((ES, HL)+B)$			
		ES:[HL+B], A	3	2	—	$((ES, HL)+B) \leftarrow A$			
		A, [HL+C]	2	1	4	$A \leftarrow (HL+C)$			
		[HL+C], A	2	1	—	$(HL+C) \leftarrow A$			
		A, ES:[HL+C]	3	2	5	$A \leftarrow ((ES, HL)+C)$			
		ES:[HL+C], A	3	2	—	$((ES, HL) + C) \leftarrow A$			
		X, !addr16	3	1	4	$X \leftarrow (addr16)$			
		X, ES:!addr16	4	2	5	$X \leftarrow (ES, addr16)$			
		X, saddr	2	1	—	$X \leftarrow (saddr)$			
		B, !addr16	3	1	4	$B \leftarrow (addr16)$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, addr16)$			
		B, saddr	2	1	—	$B \leftarrow (saddr)$			
		C, !addr16	3	1	4	$C \leftarrow (addr16)$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, addr16)$			
		C, saddr	2	1	—	$C \leftarrow (saddr)$			
		ES, saddr	3	1	—	$ES \leftarrow (saddr)$			
	XCH	A, r 注 3	1 (r=X) 2 (r=X 以外)	1	—	$A \leftrightarrow r$			
		A, !addr16	4	2	—	$A \leftrightarrow (addr16)$			
		A, ES:!addr16	5	3	—	$A \leftrightarrow (ES, addr16)$			
		A, saddr	3	2	—	$A \leftrightarrow (saddr)$			
		A, sfr	3	2	—	$A \leftrightarrow sfr$			
		A, [DE]	2	2	—	$A \leftrightarrow (DE)$			
		A, ES:[DE]	3	3	—	$A \leftrightarrow (ES, DE)$			
		A, [HL]	2	2	—	$A \leftrightarrow (HL)$			
		A, ES:[HL]	3	3	—	$A \leftrightarrow (ES, HL)$			
		A, [DE+byte]	3	2	—	$A \leftrightarrow (DE+byte)$			
		A, ES:[DE+byte]	4	3	—	$A \leftrightarrow ((ES, DE)+byte)$			
		A, [HL+byte]	3	2	—	$A \leftrightarrow (HL+byte)$			
		A, ES:[HL+byte]	4	3	—	$A \leftrightarrow ((ES, HL)+byte)$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. r=A 除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (4/18)

指令集	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 传 送	XCH	A, [HL+B]	2	2	—	$A \leftrightarrow (HL+B)$			
		A, ES:[HL+B]	3	3	—	$A \leftrightarrow ((ES, HL)+B)$			
		A, [HL+C]	2	2	—	$A \leftrightarrow (HL+C)$			
		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	—	$C \leftarrow 01H$			
		!addr16	3	1	—	$(addr16) \leftarrow 01H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 01H$			
		saddr	2	1	—	$(saddr) \leftarrow 01H$			
	CLRB	A	1	1	—	$A \leftarrow 00H$			
		X	1	1	—	$X \leftarrow 00H$			
		B	1	1	—	$B \leftarrow 00H$			
		C	1	1	—	$C \leftarrow 00H$			
		!addr16	3	1	—	$(addr16) \leftarrow 00H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 00H$			
		saddr	2	1	—	$(saddr) \leftarrow 00H$			
	MOVS	[HL+byte], X	3	1	—	$(HL+byte) \leftarrow X$	×		×
		ES:[HL+byte], X	4	2	—	$(ES, HL+byte) \leftarrow X$	×		×
16 位 数 据 传 送	MOVW	rp, #word	3	1	—	$rp \leftarrow word$			
		saddrp, #word	4	1	—	$(saddrp) \leftarrow word$			
		sfrp, #word	4	1	—	$sfrp \leftarrow word$			
		AX, rp 注 3	1	1	—	$AX \leftarrow rp$			
		rp, AX 注 3	1	1	—	$rp \leftarrow AX$			
		AX, !addr16	3	1	4	$AX \leftarrow (addr16)$			
		!addr16, AX	3	1	—	$(addr16) \leftarrow AX$			
		AX, ES:!addr16	4	2	5	$AX \leftarrow (ES, addr16)$			
		ES:!addr16, AX	4	2	—	$(ES, addr16) \leftarrow AX$			
		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	—	$sfrp \leftarrow AX$			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据的指令时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3.  $rp=AX$  除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (5/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位 数 据 传 送	MOVW	AX, [DE]	1	1	4	AX←(DE)			
		[DE], AX	1	1	—	(DE)←AX			
		AX, ES:[DE]	2	2	5	AX←(ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE)←AX			
		AX, [HL]	1	1	4	AX←(HL)			
		[HL], AX	1	1	—	(HL)←AX			
		AX, ES:[HL]	2	2	5	AX←(ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL)←AX			
		AX, [DE+byte]	2	1	4	AX←(DE+byte)			
		[DE+byte], AX	2	1	—	(DE+byte)←AX			
		AX, ES:[DE+byte]	3	2	5	AX←((ES, DE)+byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE)+byte)←AX			
		AX, [HL+byte]	2	1	4	AX←(HL+byte)			
		[HL+byte], AX	2	1	—	(HL+byte)←AX			
		AX, ES:[HL+byte]	3	2	5	AX←((ES, HL)+byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL)+byte)←AX			
		AX, [SP+byte]	2	1	—	AX←(SP+byte)			
		[SP+byte], AX	2	1	—	(SP+byte)←AX			
		AX, word[B]	3	1	4	AX←(B+word)			
		word[B], AX	3	1	—	(B+word)←AX			
		AX, ES:word[B]	4	2	5	AX←((ES, B)+word)			
		ES:word[B], AX	4	2	—	((ES, B)+word)←AX			
		AX, word[C]	3	1	4	AX←(C+word)			
		word[C], AX	3	1	—	(C+word)←AX			
		AX, ES:word[C]	4	2	5	AX←((ES, C)+word)			
		ES:word[C], AX	4	2	—	((ES, C)+word)←AX			
		AX, word[BC]	3	1	4	AX←(BC+word)			
		word[BC], AX	3	1	—	(BC+word)←AX			
		AX, ES:word[BC]	4	2	5	AX←((ES, BC)+word)			
		ES:word[BC], AX	4	2	—	((ES, BC)+word)←AX			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。



表 26-5 操作一览表 (6/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位 数 据 传 送	MOVW	BC, !addr16	3	1	4	BC←(addr16)			
		BC, ES:!addr16	4	2	5	BC←(ES, addr16)			
		DE, !addr16	3	1	4	DE←(addr16)			
		DE, ES:!addr16	4	2	5	DE←(ES, addr16)			
		HL, !addr16	3	1	4	HL←(addr16)			
		HL, ES:!addr16	4	2	5	HL←(ES, addr16)			
		BC, saddrp	2	1	—	BC←(saddrp)			
		DE, saddrp	2	1	—	DE←(saddrp)			
		HL, saddrp	2	1	—	HL←(saddrp)			
	XCHW	AX, rp 注 3	1	1	—	AX↔rp			
	ONEW	AX	1	1	—	AX←0001H			
		BC	1	1	—	BC←0001H			
	CLRW	AX	1	1	—	AX←0000H			
		BC	1	1	—	BC←0000H			
8 位 运 算	ADD	A, #byte	2	1	—	A, CY←A+byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)+byte	×	×	×
		A, r 注 4	2	1	—	A, CY←A+r	×	×	×
		r, A	2	1	—	r, CY←r+A	×	×	×
		A, !addr16	3	1	4	A, CY←A+(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A+(ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY←A+(saddr)	×	×	×
		A, [HL]	1	1	4	A, CY←A+(HL)	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A+(ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A+(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A+((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY←A+(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY←A+(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A+((ES, HL)+C)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. rp=AX 除外。

4. r=A 除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (7/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	ADDC	A, #byte	2	1	—	A, CY←A+byte+CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)+byte+CY	×	×	×
		A, r 注 3	2	1	—	A, CY←A+r+CY	×	×	×
		r, A	2	1	—	r, CY←r+A+CY	×	×	×
		A, !addr16	3	1	4	A, CY←A+(addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A+(ES, addr16)+CY	×	×	×
		A, saddr	2	1	—	A, CY←A+(saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY←A+(HL)+CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A+(ES, HL)+CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A+(HL+byte)+CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A+((ES, HL)+byte)+CY	×	×	×
		A, [HL+B]	2	1	4	A, CY←A+(HL+B)+CY	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY←A+(HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A+((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY←A-byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)-byte	×	×	×
		A, r 注 3	2	1	—	A, CY←A-r	×	×	×
		r, A	2	1	—	r, CY←r-A	×	×	×
		A, !addr16	3	1	4	A, CY←A-(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A-(ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY←A-(saddr)	×	×	×
		A, [HL]	1	1	4	A, CY←A-(HL)	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A-(ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A-(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A-((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY←A-(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A-((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY←A-(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY←A-((ES, HL)+C)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. r=A 除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (8/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	SUBC	A, #byte	2	1	—	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
		A, r 注 3	2	1	—	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	1	—	$r, CY \leftarrow r - A - CY$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (\text{ES}, \text{addr16}) - CY$	×	×	×
		A, saddr	2	1	—	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (\text{ES}, \text{HL}) - CY$	×	×	×
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + \text{byte}) - CY$	×	×	×
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (\text{HL} + B) - CY$	×	×	×
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + B) - CY$	×	×	×
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (\text{HL} + C) - CY$	×	×	×
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + C) - CY$	×	×	×
	AND	A, #byte	2	1	—	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	2	—	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \wedge r$	×		
		r, A	2	1	—	$r \leftarrow r \wedge A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (\text{ES}, \text{addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \wedge (saddr)$	×		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (\text{ES}, \text{HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \wedge (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \wedge (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + C)$	×		

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

3.  $r=A$  除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (9/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	OR	A, #byte	2	1	—	$A \leftarrow A \vee \text{byte}$	×		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \vee r$	×		
		r, A	2	1	—	$r \leftarrow r \vee A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \vee (\text{saddr})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	×		
	XOR	A, #byte	2	1	—	$A \leftarrow A \oplus \text{byte}$	×		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$	×		
		A, r 注 3	2	1	—	$A \leftarrow A \oplus r$	×		
		r, A	2	1	—	$r \leftarrow r \oplus A$	×		
		A, !addr16	3	1	4	$A \leftarrow A \oplus (\text{addr16})$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \oplus (\text{ES:addr16})$	×		
		A, saddr	2	1	—	$A \leftarrow A \oplus (\text{saddr})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \oplus (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \oplus (\text{ES:HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \oplus (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \oplus (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + C)$	×		

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{\text{CLK}}$ ) 数。

3.  $r=A$  除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (10/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位运算	CMP	A, #byte	2	1	—	A-byte	×	×	×
		!addr16, #byte	4	1	4	(addr16)-byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16)-byte	×	×	×
		saddr, #byte	3	1	—	(saddr)-byte	×	×	×
		A, r 注 3	2	1	—	A-r	×	×	×
		r, A	2	1	—	r-A	×	×	×
		A, !addr16	3	1	4	A-(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A-(ES:addr16)	×	×	×
		A, saddr	2	1	—	A-(saddr)	×	×	×
		A, [HL]	1	1	4	A-(HL)	×	×	×
		A, ES:[HL]	2	2	5	A-(ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A-(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A-((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A-(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A-((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A-(HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A-((ES:HL)+C)	×	×	×
	CMP0	A	1	1	—	A-00H	×	0	0
		X	1	1	—	X-00H	×	0	0
		B	1	1	—	B-00H	×	0	0
		C	1	1	—	C-00H	×	0	0
		!addr16	3	1	4	(addr16)-00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16)-00H	×	0	0
		saddr	2	1	—	(saddr)-00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X-(HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X-((ES:HL)+byte)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. r=A 除外。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (11/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位运算	ADDW	AX, #word	3	1	—	AX, CY←AX+word	×	×	×
		AX, AX	1	1	—	AX, CY←AX+AX	×	×	×
		AX, BC	1	1	—	AX, CY←AX+BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX+DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY←AX-word	×	×	×
		AX, BC	1	1	—	AX, CY←AX-BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX-DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX-HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX-((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX-word	×	×	×
		AX, BC	1	1	—	AX-BC	×	×	×
		AX, DE	1	1	—	AX-DE	×	×	×
		AX, HL	1	1	—	AX-HL	×	×	×
		AX, !addr16	3	1	4	AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX-((ES:HL)+byte)	×	×	×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5  操作一览表 (12/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
乘除和乘加运算	MULU	X	1	1	—	$AX \leftarrow A \times X$			

注  1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟（f<sub>CLK</sub>）数。  
     2. 这是存取程序存储区时的 CPU 时钟（f<sub>CLK</sub>）数。

备注  时钟数是在内部 ROM（闪存）区有程序的情况。当从内部 RAM 区取指令时，最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (13/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
递增 / 递减	INC	r	1	1	—	$r \leftarrow r+1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)+1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r-1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)-1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$	×	×	
	INCW	rp	1	1	—	$rp \leftarrow rp+1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	1	—	$rp \leftarrow rp-1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)-1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$			
		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. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 1. 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

2. cnt 是位的移位个数。



表 26-5 操作一览表 (14/18)

指令集	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
循环	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, A.bit	2	1	—	$CY \leftarrow A.bit$			×
		A.bit, CY	2	1	—	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	—	$CY \leftarrow PSW.bit$			×
		PSW.bit, CY	3	4	—	$PSW.bit \leftarrow CY$	×	×	
		CY, saddr.bit	3	1	—	$CY \leftarrow (saddr).bit$			×
		saddr.bit, CY	3	2	—	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	—	$CY \leftarrow sfr.bit$			×
		sfr.bit, CY	3	2	—	$sfr.bit \leftarrow CY$			
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		[HL].bit, CY	2	2	—	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			×
		ES:[HL].bit, CY	3	3	—	$(ES, HL).bit \leftarrow CY$			
	AND1	CY, A.bit	2	1	—	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \wedge sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			×
	OR1	CY, A.bit	2	1	—	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \vee PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \vee sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (15/18)

指令集	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, A.bit	2	1	—	$CY \leftarrow CY \oplus A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \oplus PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \oplus (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \oplus sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \oplus (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \oplus (ES, HL).bit$			×
	SET1	A.bit	2	1	—	$A.bit \leftarrow 1$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 1$	×	×	×
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 1$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 1$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 1$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 1$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 1$			
	CLR1	A.bit	2	1	—	$A.bit \leftarrow 0$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 0$	×	×	×
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 0$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 0$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 0$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 0$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 0$			
	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. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (16/18)

指令集	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. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (17/18)

指令集	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 <sub>H</sub> , (SP-2)←rp <sub>L</sub> , SP←SP-2			
	POP	PSW	2	3	—	PSW←(SP+1), SP←SP+2	R	R	R
		rp	1	1	—	rp <sub>L</sub> ←(SP), rp <sub>H</sub> ←(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			
		\$laddr20	3	3	—	PC←PC+3+jdisp16			
		laddr16	3	3	—	PC←0000, addr16			
		!!addr20	4	3	—	PC←addr20			
条件转移	BC	\$addr20	2	2/4 注3	—	PC←PC+2+jdisp8 if CY=1			
	BNC	\$addr20	2	2/4 注3	—	PC←PC+2+jdisp8 if CY=0			
	BZ	\$addr20	2	2/4 注3	—	PC←PC+2+jdisp8 if Z=1			
	BNZ	\$addr20	2	2/4 注3	—	PC←PC+2+jdisp8 if Z=0			
	BH	\$addr20	3	2/4 注3	—	PC←PC+3+jdisp8 if (Z∨CY)=0			
	BNH	\$addr20	3	2/4 注3	—	PC←PC+3+jdisp8 if (Z∨CY)=1			
	BT	saddr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if (saddr).bit=1			
		sfr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if sfr.bit=1			
		A.bit, \$addr20	3	3/5 注3	—	PC←PC+3+disp8 if A.bit=1			
		PSW.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if PSW.bit=1			
		[HL].bit, \$addr20	3	3/5 注3	6/7	PC←PC+3+jdisp8 if (HL).bit=1			
		ES:[HL].bit, \$addr20	4	4/6 注3	7/8	PC←PC+4+jdisp8 if (ES, HL).bit=1			

- 注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。
2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。
3. 表示“条件不成立时 / 条件成立时”的时钟数。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

表 26-5 操作一览表 (18/18)

指令集	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	—	PC←PC+4+jdisp8 if (saddr).bit=0			
		sfr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if sfr.bit=0			
		A.bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if A.bit=0			
		PSW.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if PSW.bit=0			
		[HL].bit, \$addr20	3	3/5 注3	6/7	PC←PC+3+jdisp8 if (HL).bit=0			
		ES:[HL].bit, \$addr20	4	4/6 注3	7/8	PC←PC+4+jdisp8 if (ES, HL).bit=0			
	BTCLR	saddr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if (saddr).bit=1 then reset (saddr).bit			
		sfr.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if sfr.bit=1 then reset sfr.bit			
		A.bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if A.bit=1 then reset A.bit			
		PSW.bit, \$addr20	4	3/5 注3	—	PC←PC+4+jdisp8 if PSW.bit=1 then reset PSW.bit	×	×	×
		[HL].bit, \$addr20	3	3/5 注3	—	PC←PC+3+jdisp8 if (HL).bit=1 then reset (HL).bit			
		ES:[HL].bit, \$addr20	4	4/6 注3	—	PC←PC+4+jdisp8 if (ES, HL).bit=1 then reset (ES, HL).bit			
条件跳转	SKC	—	2	1	—	Next instruction skip if CY=1			
	SKNC	—	2	1	—	Next instruction skip if CY=0			
	SKZ	—	2	1	—	Next instruction skip if Z=1			
	SKNZ	—	2	1	—	Next instruction skip if Z=0			
	SKH	—	2	1	—	Next instruction skip if (Z∨CY)=0			
	SKNH	—	2	1	—	Next instruction skip if (Z∨CY)=1			
CPU 控制	SEL 注4	RBn	2	1	—	RBS[1:0]←n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE←1 (Enable Interrupt)			
	DI	—	3	4	—	IE←0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 ( $f_{CLK}$ ) 数。

2. 这是存取程序存储区时的 CPU 时钟 ( $f_{CLK}$ ) 数。

3. 表示“条件不成立时 / 条件成立时”的时钟数。

4. n 为寄存器组号 (n=0 ~ 3)。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

## 第 27 章 电特性

- 注意 1. R7F0C010 内置用于开发和评估的片上调试功能。如果使用片上调试功能，就可能会超过闪存的保证改写次数而无法保证产品的可靠性，因此批量生产的产品不能使用片上调试功能。对于使用片上调试功能的产品，不作为投诉受理对象。
2. 配置的引脚因产品而不同。请参照“2.1 引脚功能一览表”～“2.2 端口以外的引脚”。

## 27.1 绝对最大额定值

绝对最大额定值 ( $T_A=25^{\circ}\text{C}$ ) (1/2)

项目	符号	条件	额定值	单位
电源电压	$V_{DD}$		$-0.5 \sim +4.6$	V
	$V_{SS}$		$-0.5 \sim +0.3$	V
REGC 引脚的 输入电压	$V_{IREGC}$	REGC	$-0.3 \sim +2.8$ 并且 $-0.3 \sim V_{DD}+0.3$ 注 1	V
输入电压	$V_{I1}$	P10 ~ P17、P20 ~ P27、P30 ~ P35、 P40、P121、P122、P137	$-0.3 \sim V_{DD}+0.3$ 注 2	V
	$V_{I2}$	P60、P61 (N 沟道漏极开路)	$-0.3 \sim +6.5$	V
输出电压	$V_{O1}$	P10 ~ P17、P20 ~ P27、P30 ~ P35、 P40、P60、P61、P121、P122、P137	$-0.3 \sim V_{DD}+0.3$ 注 2	V
模拟输入电压	$V_{AI1}$	AN10 ~ AN17、ANI16	$-0.3 \sim V_{DD}+0.3$ 并且 $-0.3 \sim AV_{REF(+)}+0.3$ 注 2、3	V

- 注 1. 必须通过电容器 ( $0.47 \sim 1\mu\text{F}$ ) 将 REGC 引脚连接  $V_{SS}$ 。此值是规定 REGC 引脚的绝对最大额定值, 使用时不能外加电压。
2. 不超过 4.6V。
3. A/D 转换对象的引脚不能超过  $AV_{REF(+)}+0.3$ 。

注意 即使是各项目中的 1 个项目瞬间超过绝对最大额定值, 也可能降低产品的质量。绝对最大额定值是可能给产品带来物理性损伤的额定值, 必须在不超过额定值的状态下使用产品。

备注 1. 在没有特别指定的情况下, 复用引脚的特性和端口引脚的特性相同。

2.  $AV_{REF(+)}$ : A/D 转换器的正 (+) 基准电压

绝对最大额定值 ( $T_A=25^{\circ}\text{C}$ ) (2/2)

项目	符号	条件		额定值	单位
高电平输出电流	I <sub>OH1</sub>	1 个引脚	P10 ~ P17、P30 ~ P35、P40	-40	mA
		引脚合计 -170mA	P40	-40	mA
			P10 ~ P17、 P30 ~ P35	-100	mA
	I <sub>OH2</sub>	1 个引脚	P20 ~ P27	-0.5	mA
		引脚合计		-2	mA
	低电平输出电流	I <sub>OL1</sub>	1 个引脚	P10 ~ P17、P30 ~ P35、P40、 P60、 P61	40
引脚合计 170mA			P40	40	mA
			P10 ~ P17、P30 ~ P35、P60、 P61	100	mA
I <sub>OL2</sub>		1 个引脚	P20 ~ P27	1	mA
		引脚合计		5	mA
工作环境温度		T <sub>A</sub>	通常运行时		-40 ~ +85
	闪存编程时		0 ~ +40		
保存温度	T <sub>stg</sub>			-65 ~ +150	°C

**注意** 即使是各项目中的 1 个项目瞬间超过绝对最大额定值，也可能降低产品的质量。绝对最大额定值是可能给产品带来物理性损伤的额定值，必须在不超过额定值的状态下使用产品。

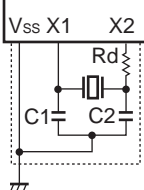
**备注** 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。



## 27.2 振荡电路特性

### 27.2.1 X1 振荡电路特性

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS}=0\text{V}$ )

项目	谐振器	推荐电路	条件	MIN.	TYP.	MAX.	单位
X1 时钟振荡频率 ( $f_X$ ) 注	陶瓷谐振器 / 晶体谐振器		$2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$	1.0		20.0	MHz

注 只表示振荡电路的特性，指令执行时间请参照“27.4 AC 特性”。

注意 1. 当使用 X1 振荡电路时，为了避免布线电容等的影响，必须对图中的虚线部分进行以下的布线：

- 尽量缩短布线。
  - 不和其他信号线交叉。
  - 不接近有变化的大电流流过的布线。
  - 振荡电路的电容器接地点总是和  $V_{SS}$  同电位。
  - 不将有大电流流过的接地图形接地。
  - 不从振荡电路取出信号。
2. 在解除复位后，通过高速内部振荡器时钟启动 CPU，因此用户必须通过振荡稳定时间计数器的状态寄存器（OSTC）确认 X1 时钟的振荡稳定时间，并且必须充分对所用谐振器进行振荡稳定时间的评估，然后决定 OSTC 寄存器和振荡稳定时间选择寄存器（OSTS）的振荡稳定时间。

### 27.2.2 内部振荡器特性

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS}=0\text{V}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
高速内部振荡器的振荡频率注 1、2	$f_{IH}$	选择 32MHz 时	31.36	32.00	32.64	MHz
		选择 24MHz 时	23.52	24.00	24.48	MHz
低速内部振荡器的振荡频率注 2	$f_{IL}$		12.75	15.00	17.25	kHz

注 1. 通过选项字节（000C2H）的 bit0 ~ 3 和 HOCODIV 寄存器的 bit0 ~ 2 选择高速内部振荡器的频率。

2. 只表示振荡电路的特性，指令执行时间请参照“27.4 AC 特性”。

## 27.3 DC 特性

## 27.3.1 引脚特性

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS} = 0\text{V}$ ) (1/3)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输出电流 <sup>注1</sup>	$I_{OH1}$	P10 ~ P17、P30 ~ P35、P40 1 个引脚			-2.0 <sup>注2</sup>	mA
		P40 合计 (占空比 = 70% 时 <sup>注3</sup> )			-2.0	mA
		P10 ~ P17、P30 ~ P35 合计 (占空比 = 70% 时 <sup>注3</sup> )			-19.0	mA
		全部引脚合计 (占空比 = 70% 时 <sup>注3</sup> )			-21.0	mA
	$I_{OH2}$	P20 ~ P27 1 个引脚			-0.1	mA
		全部引脚合计 (占空比 = 70% 时 <sup>注3</sup> )			-0.8	mA

注 1. 这是即使电流从  $V_{DD}$  引脚流到输出引脚也保证器件工作的电流值。

2. 不能超过合计的电流值。

3. 这是“占空比 = 70% 条件”的规格。

更改占空比的输出电流值能用以下的计算式进行计算（将 70% 占空比改为  $n\%$  的情况）。

• 引脚合计的输出电流 =  $(I_{OH} \times 0.7) / (n \times 0.01)$

<计算例子>  $I_{OH} = -10.0\text{mA}$ 、 $n = 50\%$

引脚合计的输出电流 =  $(-10.0 \times 0.7) / (50 \times 0.01) = -14.0\text{mA}$

各引脚的电流不会因占空比而变，而且不会流过绝对最大额定值以上的电流。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V) (2/3)

项目	符号	条件	MIN.	TYP.	MAX.	单位
低电平输出电流 <sup>注1</sup>	I <sub>OL1</sub>	P10 ~ P17、P30 ~ P35、P40 1 个引脚			3.0 <sup>注2</sup>	mA
		P60、P61 1 个引脚			3.0 <sup>注2</sup>	mA
		P40 合计 (占空比 =70% 时 <sup>注3</sup> )			3.0	mA
		P10 ~ P17、P30 ~ P35、P60、P61 合计 (占空比 =70% 时 <sup>注3</sup> )			35.0	mA
		全部引脚合计 (占空比 =70% 时 <sup>注3</sup> )			38.0	mA
	I <sub>OL2</sub>	P20 ~ P27 1 个引脚			0.4	mA
		全部引脚合计 (占空比 =70% 时 <sup>注3</sup> )			3.2	mA

- 注 1. 这是即使电流从输出引脚流到 V<sub>SS</sub> 引脚也保证器件工作的电流值。
2. 不能超过合计的电流值。
3. 这是“占空比 =70% 条件”的规格。  
更改占空比的输出电流值能用以下的计算式进行计算（将 70% 占空比改为 n% 的情况）。
- 引脚合计的输出电流 = (I<sub>OL</sub> × 0.7) / (n × 0.01)
- <计算例子> I<sub>OL</sub> = 10.0mA、n = 50%
- $$\text{引脚合计的输出电流} = (10.0 \times 0.7) / (50 \times 0.01) = 14.0\text{mA}$$

各引脚的电流不会因占空比而变，而且不会流过绝对最大额定值以上的电流。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V) (3/3)

项目	符号	条件		MIN.	TYP.	MAX.	单位
高电平输入电压	V <sub>IH1</sub>	P10 ~ P17、P30 ~ P35、P40、P121、P122、P137、EXCLK、 $\overline{\text{RESET}}$		0.8V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH2</sub>	P20 ~ P27		0.7V <sub>DD</sub>		V <sub>DD</sub>	V
	V <sub>IH3</sub>	P60、P61		0.7V <sub>DD</sub>		6.0	V
低电平输入电压	V <sub>IL1</sub>	P10 ~ P17、P30 ~ P35、P40、P121、P122、P137、EXCLK、 $\overline{\text{RESET}}$		0		0.2V <sub>DD</sub>	V
	V <sub>IL2</sub>	P20 ~ P27、P60、P61		0		0.3V <sub>DD</sub>	V
高电平输出电压	V <sub>OH1</sub>	P10 ~ P17、P30 ~ P35、P40	I <sub>OH1</sub> =−2.0mA	V <sub>DD</sub> −0.6			V
	V <sub>OH2</sub>	P20 ~ P27	I <sub>OH2</sub> =−100μA	V <sub>DD</sub> −0.5			V
低电平输出电压	V <sub>OL1</sub>	P10 ~ P17、P30 ~ P35、P40	I <sub>OL1</sub> =3.0mA			0.6	V
			I <sub>OL1</sub> =1.5mA			0.4	V
	V <sub>OL2</sub>	P20 ~ P27	I <sub>OL2</sub> =400μA			0.4	V
	V <sub>OL3</sub>	P60、P61	I <sub>OL3</sub> =3.0mA			0.4	V
高电平输入漏电流	I <sub>LIH1</sub>	P10 ~ P17、P20 ~ P27、P30 ~ P35、P40、P137、 $\overline{\text{RESET}}$	V <sub>I</sub> =V <sub>DD</sub>			1	μA
	I <sub>LIH2</sub>	P121、P122 (X1、X2、EXCLK)	V <sub>I</sub> =V <sub>DD</sub>	输入端口和外部时钟输入时		1	μA
				连接谐振器时		10	μA
低电平输入漏电流	I <sub>LIL1</sub>	P10 ~ P17、P20 ~ P27、P30 ~ P35、P40、P137、 $\overline{\text{RESET}}$	V <sub>I</sub> =V <sub>SS</sub>			−1	μA
	I <sub>LIL2</sub>	P121、P122 (X1、X2、EXCLK)	V <sub>I</sub> =V <sub>SS</sub>	输入端口和外部时钟输入时		−1	μA
				连接谐振器时		−10	μA
内部上拉电阻	R <sub>U</sub>	P10 ~ P17、P30 ~ P35、P40	V <sub>I</sub> =V <sub>SS</sub> 、输入端口时	10	20	100	kΩ

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

## 27.3.2 电源电流特性

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

(1/3)

项目	符号	条件					MIN.	TYP.	MAX.	单位
电源电流	I <sub>DD1</sub> 注 1	运行模式	HS（高速主）模式注 4	f <sub>IH</sub> =32MHz 注 3	基本运行	V <sub>DD</sub> =3.0V		2.1		mA
					通常运行	V <sub>DD</sub> =3.0V		4.8	7.0	mA
				f <sub>IH</sub> =24MHz 注 3	通常运行	V <sub>DD</sub> =3.0V		3.8	5.5	mA
					f <sub>IH</sub> =16MHz 注 3	通常运行	V <sub>DD</sub> =3.0V		2.8	4.0
			LS（低速主）运行注 4	f <sub>IH</sub> =8MHz 注 3	通常运行	V <sub>DD</sub> =3.0V		1.3	2.0	mA
			HS（高速主）模式注 4	f <sub>MX</sub> =20MHz 注 2、 V <sub>DD</sub> =3.0V	通常运行	输入方波		3.3	4.6	mA
						连接谐振器		3.5	4.8	
				f <sub>MX</sub> =10MHz 注 2、 V <sub>DD</sub> =3.0V	通常运行	输入方波		2.0	2.7	mA
						连接谐振器		2.1	2.7	
			LS（低速主）模式注 4	f <sub>MX</sub> =8MHz 注 2、 V <sub>DD</sub> =3.0V	通常运行	输入方波		1.2	2.0	mA
						连接谐振器		1.2	2.0	

注 1. 这是流过 V<sub>DD</sub> 的总电流，包含输入引脚固定为 V<sub>DD</sub> 或者 V<sub>SS</sub> 状态的输入漏电流。MAX. 值包含后台操作 (BGO) 以外的外围工作电流。但是，不包含流到 A/D 转换器、D/A 转换器、LVD 电路、I/O 端口以及内部上拉或者下拉电阻的电流。

2. 这是高速内部振荡器停止振荡的情况。

3. 这是高速系统时钟停止振荡的情况。

4. 工作电压范围、CPU 工作频率和运行模式的关系如下所示：

HS (高速主) 模式 : @1MHz ~ 32MHz

@1MHz ~ 16MHz

LS (低速主) 模式 : @1MHz ~ 8MHz

备注 1. f<sub>MX</sub> : 高速系统时钟频率 (X1 时钟振荡频率或者外部主系统时钟频率)

2. f<sub>IH</sub> : 高速内部振荡器的时钟频率

3. TYP. 值的温度条件是 T<sub>A</sub>=25°C。

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

(2/3)

项目	符号	条件				MIN.	TYP.	MAX.	单位	
电源电流 <sup>注1</sup>	I <sub>DD2</sub> <sup>注2</sup>	HALT 模式	HS（高速 主）模式 <sup>注6</sup>	f <sub>IH</sub> =32MHz <sup>注4</sup>	V <sub>DD</sub> =3.0V		0.60	1.63	mA	
				f <sub>IH</sub> =24MHz <sup>注4</sup>	V <sub>DD</sub> =3.0V		0.49	1.28	mA	
				f <sub>IH</sub> =16MHz <sup>注4</sup>	V <sub>DD</sub> =3.0V		0.45	1.00	mA	
			LS（低速 主）模式 <sup>注6</sup>	f <sub>IH</sub> =8MHz <sup>注4</sup>	V <sub>DD</sub> =3.0V		320	530	μA	
			HS（高速 主）模式 <sup>注6</sup>	f <sub>MX</sub> =20MHz <sup>注3</sup> 、 V <sub>DD</sub> =3.0V	输入方波		0.28	1.00	mA	
					连接谐振器		0.49	1.17		
				f <sub>MX</sub> =10MHz <sup>注3</sup> 、 V <sub>DD</sub> =3.0V	输入方波		0.19	0.60	mA	
					连接谐振器		0.30	0.67		
			LS（低速 主）模式 <sup>注6</sup>	f <sub>MX</sub> =8MHz <sup>注3</sup> 、 V <sub>DD</sub> =3.0V	输入方波		95	330	μA	
					连接谐振器		145	380		
	I <sub>DD3</sub> <sup>注5</sup>	STOP 模式	T <sub>A</sub> =−40℃					0.18		μA
			T <sub>A</sub> =+25℃					0.23	0.50	
			T <sub>A</sub> =+50℃					0.26	1.10	
T <sub>A</sub> =+70℃					0.29	1.90				
T <sub>A</sub> =+85℃					0.90	3.30				

注 1. 这是流过 V<sub>DD</sub> 的总电流，包含输入引脚固定为 V<sub>DD</sub> 或者 V<sub>SS</sub> 状态的输入漏电流。MAX. 值包含外围工作电流。但是，不包含流到 A/D 转换器、D/A 转换器、LVD 电路、I/O 端口以及内部上拉或者下拉电阻的电流。

2. 这是执行闪存中的 HALT 指令的情况。

3. 这是高速内部振荡器停止振荡的情况。

4. 这是高速系统时钟停止振荡的情况。

5. 这是高速内部振荡器和高速系统时钟停止振荡的情况，或者看门狗定时器停止运行的情况。MAX. 值包含漏电流。

6. 工作电压范围、CPU 工作频率和运行模式的关系如下所示：

HS (高速主) 模式 : @1MHz ~ 32MHz

@1MHz ~ 16MHz

LS (低速主) 模式 : @1MHz ~ 8MHz

备注 1. f<sub>MX</sub> : 高速系统时钟频率 (X1 时钟振荡频率或者外部主系统时钟频率)

2. f<sub>IH</sub> : 高速内部振荡器的时钟频率

3. “STOP 模式”以外的 TYP. 值的温度条件是 T<sub>A</sub>=25°C。

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

(3/3)

项目	符号	条件	MIN.	TYP.	MAX.	单位
看门狗定时器 工作电流	I <sub>WDT</sub> 注 1、2	f <sub>IL</sub> =15kHz		0.22		μA
A/D 转换器 工作电流	I <sub>ADC</sub> 注 3	V <sub>DD</sub> =3.0V、最高速转换时		0.58	0.82	mA
A/D 转换器 基准电压电流	I <sub>ADREF</sub>			75.0		μA
温度传感器 工作电流	I <sub>TMPS</sub>			75.0		μA
D/A 转换器 工作电流	I <sub>DAC</sub> 注 4				2	mA
LVD 工作电流	I <sub>LVD</sub> 注 5			0.08		μA
BGO 电流	I <sub>BGO</sub> 注 6			2.50	12.20	mA

注 1. 这是高速内部振荡器和高速系统时钟停止振荡的情况。

2. 这是只流到看门狗定时器的电流（包含 15kHz 低速内部振荡器的工作电流）。在看门狗定时器运行的情况下，R7F0C010 的电源电流值为 I<sub>DD1</sub> 或者 I<sub>DD2</sub> 或者 I<sub>DD3</sub> 加上 I<sub>WDT</sub> 的值。

3. 这是只流到 A/D 转换器的电流。在运行模式或者 HALT 模式中 A/D 转换器运行的情况下，R7F0C010 的电源电流值为 I<sub>DD1</sub> 或者 I<sub>DD2</sub> 加上 I<sub>ADC</sub> 的值。

4. 这是只流到 D/A 转换器的电流。在运行模式或者 HALT 模式中 D/A 转换器运行的情况下，R7F0C010 的电源电流值为 I<sub>DD1</sub> 或者 I<sub>DD2</sub> 加上 I<sub>DAC</sub> 的值。

5. 这是只流到 LVD 电路的电流。在 LVD 电路运行的情况下，R7F0C010 的电源电流值为 I<sub>DD1</sub> 或者 I<sub>DD2</sub> 或者 I<sub>DD3</sub> 加上 I<sub>LVD</sub> 的值。

6. 这是只流到 BGO 的电流。在运行模式中 BGO 运行的情况下，R7F0C010 的电流值为 I<sub>DD1</sub> 或者 I<sub>DD2</sub> 加上 I<sub>BGO</sub> 的值。

备注 1. f<sub>IL</sub> : 低速内部振荡器的时钟频率

2. f<sub>CLK</sub> : CPU/ 外围硬件的时钟频率

3. TYP. 值的温度条件是 T<sub>A</sub>=25°C。

## 27.4 AC 特性

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
指令周期 (最短指令执行时间)	T <sub>CY</sub>	主系统时钟 (f <sub>MAIN</sub> ) 运行	HS (高速主) 模式	0.03125	1	μs
			LS (低速主) 模式	0.125	1	μs
		自编程时	HS (高速主) 模式	0.03125	1	μs
			LS (低速主) 模式	0.125	1	μs
外部系统时钟频率	f <sub>EX</sub>		1.0		20.0	MHz
外部系统时钟输入的 高低电平宽度	t <sub>EXH</sub> 、 t <sub>EXL</sub>		24			ns
TI00 ~ TI03 输入的 高低电平宽度	t <sub>TIH</sub> 、 t <sub>TIL</sub>		1/f <sub>MCK</sub> +10			ns
TO00 ~ TO03 的 输出频率	f <sub>TO</sub>	HS (高速主) 模式			8	MHz
		LS (低速主) 模式			4	MHz
PCLBUZ0、PCLBUZ1 的输出频率	f <sub>PCL</sub>	HS (高速主) 模式			8	MHz
		LS (低速主) 模式			4	MHz
中断输入的高低电平 宽度	t <sub>INTH</sub> 、 t <sub>INTL</sub>	INTP0 ~ INTP5	1			μs
RESET 的低电平宽度	t <sub>RSL</sub>		10			μs

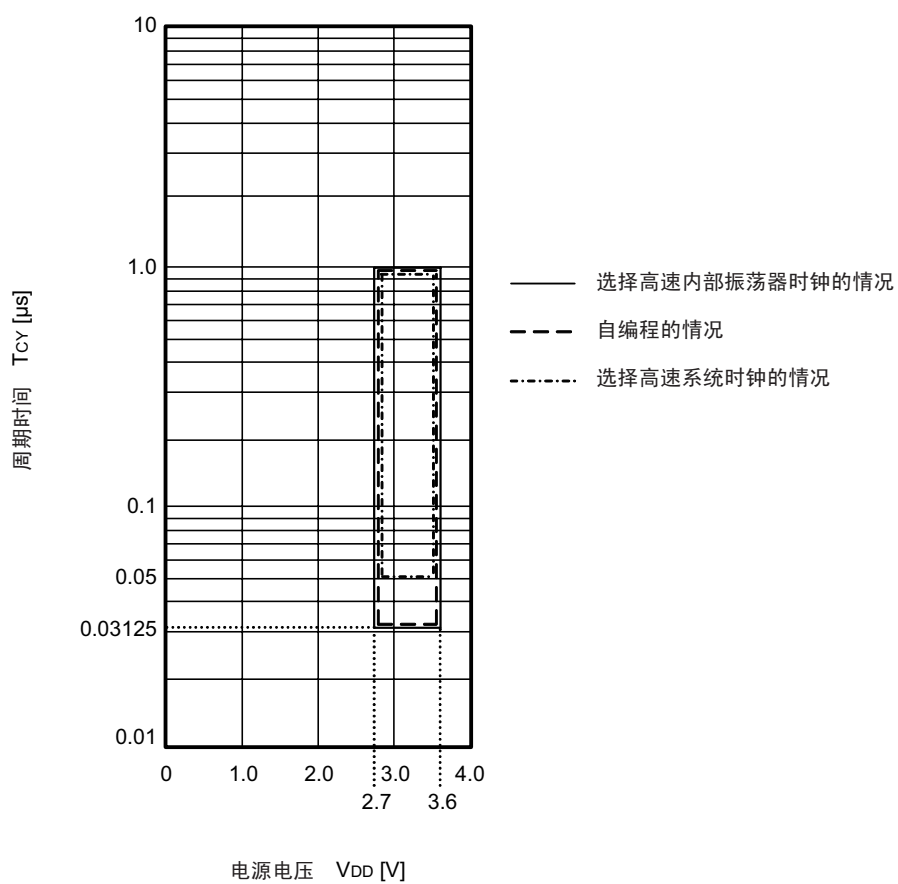
备注 f<sub>MCK</sub>: 定时器阵列单元的运行时钟频率

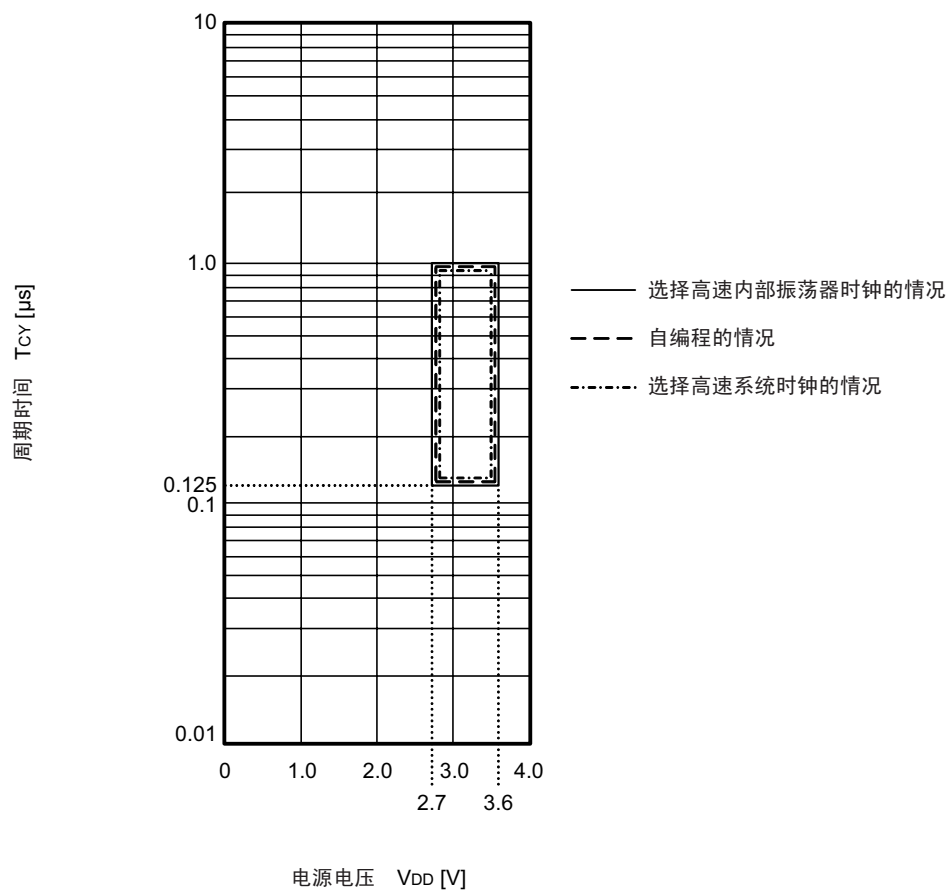
(这是定时器模式寄存器 mn (TMRmn) 的 CKSmn0 位和 CKSmn1 位设定的运行时钟。m: 单元号 (m=0)

n: 通道号 (n=0~3))



## 主系统时钟运行时的最短指令执行时间

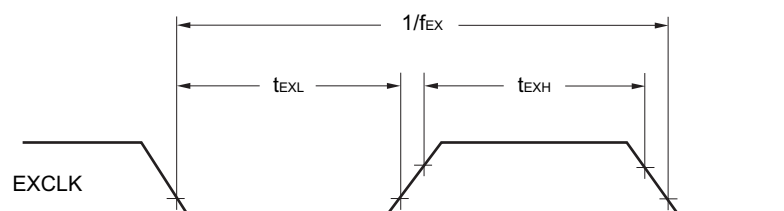
 $T_{CY}$  vs  $V_{DD}$  (HS (高速主) 模式)

$T_{CY}$  vs  $V_{DD}$  (LS (低速主) 模式)

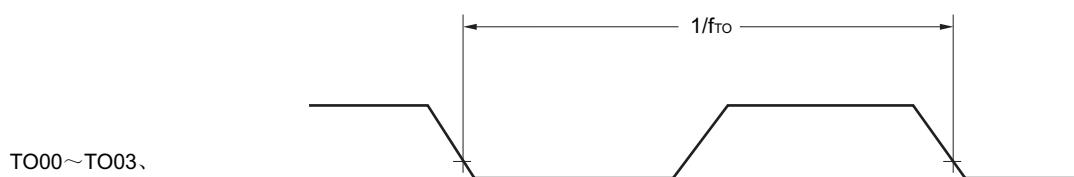
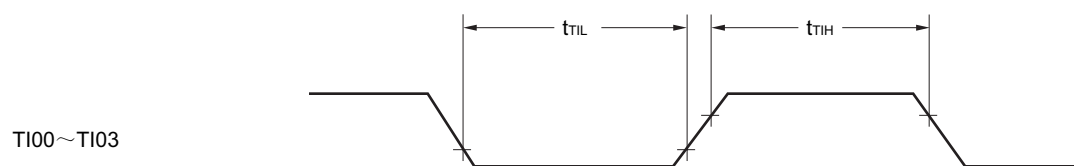
## AC 时序测量点



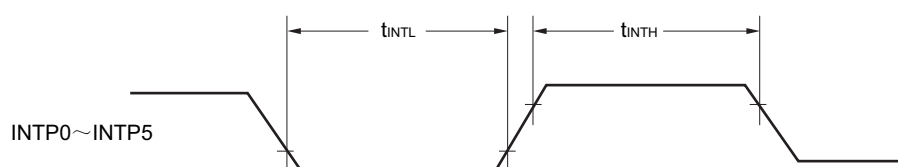
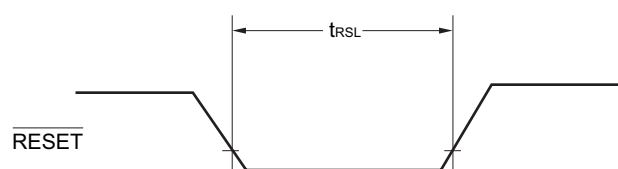
## 外部系统时钟的时序



## TI/TO 时序

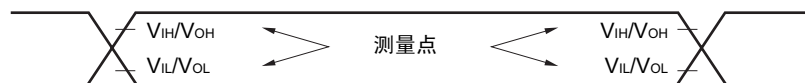


## 中断请求的输入时序

 $\overline{\text{RESET}}$  的输入时序

## 27.5 外围功能特性

### AC 时序测量点



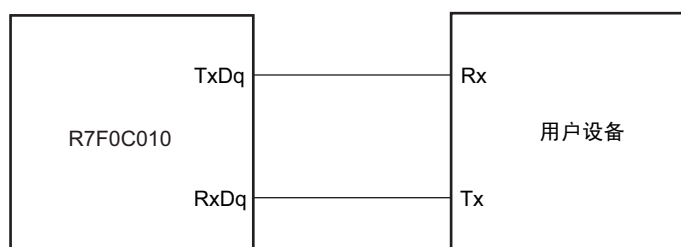
### 27.5.1 串行阵列单元

#### (1) 同电位的通信 (UART 模式)

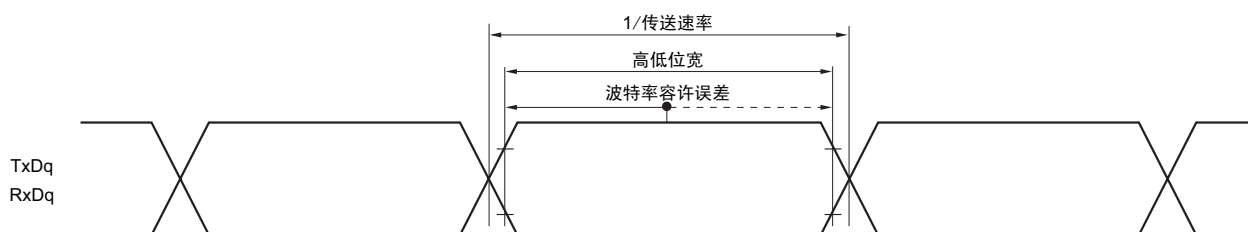
( $T_A = -40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	
传送速率注 1		$2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$		$f_{MCK}/6$		$f_{MCK}/6$	bps
		最大传送速率的理论值 $f_{MCK} = f_{CLK}$ 注 2		5.3		1.3	Mbps

UART 模式的连接图 (同电位的通信)



UART 模式的位宽 (同电位的通信) (参考)



- 注 1. SNOOZE 模式中的传送速率只为 4800bps。
2. CPU/ 外围硬件时钟 ( $f_{CLK}$ ) 的最大工作频率如下所示:
- HS (高速主) 模式 : 32MHz ( $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ )
- LS (低速主) 模式 : 8MHz ( $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ )

备注 1. q: UART 号 (q=0)

2.  $f_{MCK}$ : 串行阵列单元的运行时钟频率

(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00、01))

(2) 同电位的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出, 只对应 CSI00)

( $T_A = -40 \sim +85^\circ\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS} = 0\text{V}$ )

项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间	$t_{KCY1}$		83.3		250		ns
SCKp 高低 电平宽度	$t_{KH1}$ 、 $t_{KL1}$		$t_{KCY1}/2-10$		$t_{KCY1}/2-50$		ns
Slp 准备时间 (对 SCKp↑) 注 1	$t_{SIK1}$		33 注 3		110		ns
Slp 保持时间 (对 SCKp↑) 注 1	$t_{KSI1}$		10		10		ns
SCKp↓→SOp 输出延迟时间 注 2	$t_{KSO1}$	C=20pF 注 4		10		10	ns

- 注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。
3.  $f_{MCK}$  必须在不超过 24MHz 的情况下使用。
4. C 是 SCKp、SOp 输出线的负载电容。

备注 1. p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0)

2.  $f_{MCK}$ : 串行阵列单元的运行时钟频率

(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))

## (3) 同电位的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出)

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间	t <sub>KCY1</sub>		125		500		ns
SCKp 高低 电平宽度	t <sub>KH1</sub> 、 t <sub>KL1</sub>		t <sub>KCY1</sub> /2–18		t <sub>KCY1</sub> /2–50		ns
Slp 准备时间 (对 SCKp↑) 注 1	t <sub>SIK1</sub>		44		110		ns
Slp 保持时间 (对 SCKp↑) 注 1	t <sub>KSI1</sub>		19		19		ns
SCKp↓→SOp 输出延迟时间 注 2	t <sub>KSO1</sub>	C=30pF 注 3		25		25	ns

注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。

2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。

3. C 是 SCKp、SOp 输出线的负载电容。

备注 1. p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0)

2. f<sub>MCK</sub>: 串行阵列单元的运行时钟频率

(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))

## (4) 同电位的通信 (CSI 模式) (从属模式, SCKp..... 外部时钟输入)

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件		HS (高速主) 模式		LS (低速主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间注 4	t <sub>KCY2</sub>		16MHz < f <sub>MCK</sub>	8/f <sub>MCK</sub>		—		ns
			f <sub>MCK</sub> ≤ 16MHz	6/f <sub>MCK</sub>		6/f <sub>MCK</sub>		ns
SCKp 高低 电平宽度	t <sub>KH2</sub> 、 t <sub>KL2</sub>			t <sub>KCY2</sub> /2–8		t <sub>KCY2</sub> /2–8		ns
Slp 准备时间 (对 SCKp↑) 注 1	t <sub>SIK2</sub>			1/f <sub>MCK</sub> +20		1/f <sub>MCK</sub> +30		ns
Slp 保持时间 (对 SCKp↑) 注 1	t <sub>KSI2</sub>			1/f <sub>MCK</sub> +31		1/f <sub>MCK</sub> +31		ns
SCKp↓→SOp 输出延迟时间 注 2	t <sub>KSO2</sub>	C=30pF 注 3			2/f <sub>MCK</sub> +44		2/f <sub>MCK</sub> +110	ns
SSI00 准备 时间	t <sub>SSIK</sub>	DAPmn=0		120		120		ns
		DAPmn=1		1/f <sub>MCK</sub> +120		1/f <sub>MCK</sub> +120		ns
SSI00 保持 时间	t <sub>KSSI</sub>	DAPmn=0		1/f <sub>MCK</sub> +120		1/f <sub>MCK</sub> +120		ns
		DAPmn=1		120		120		ns

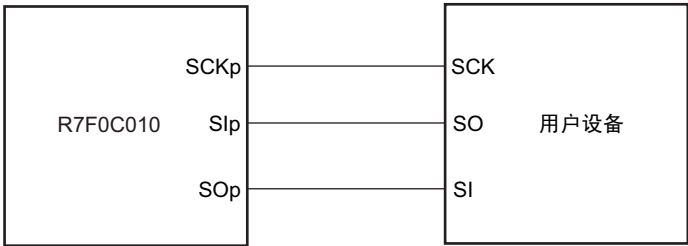
- 注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。
3. C 是 SOp 输出线的负载电容。
4. SNOOZE 模式中的传送速率为 MAX. 1Mbps。

备注 1. p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0)

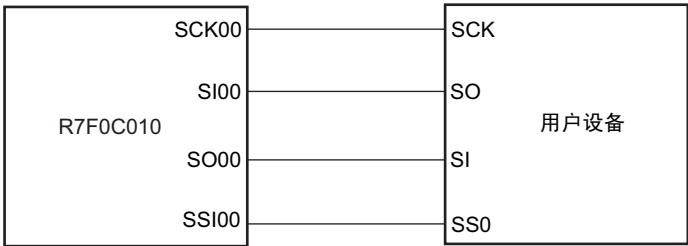
2. f<sub>MCK</sub>: 串行阵列单元的运行时钟频率

(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))

CSI模式的连接图（同电位的通信）



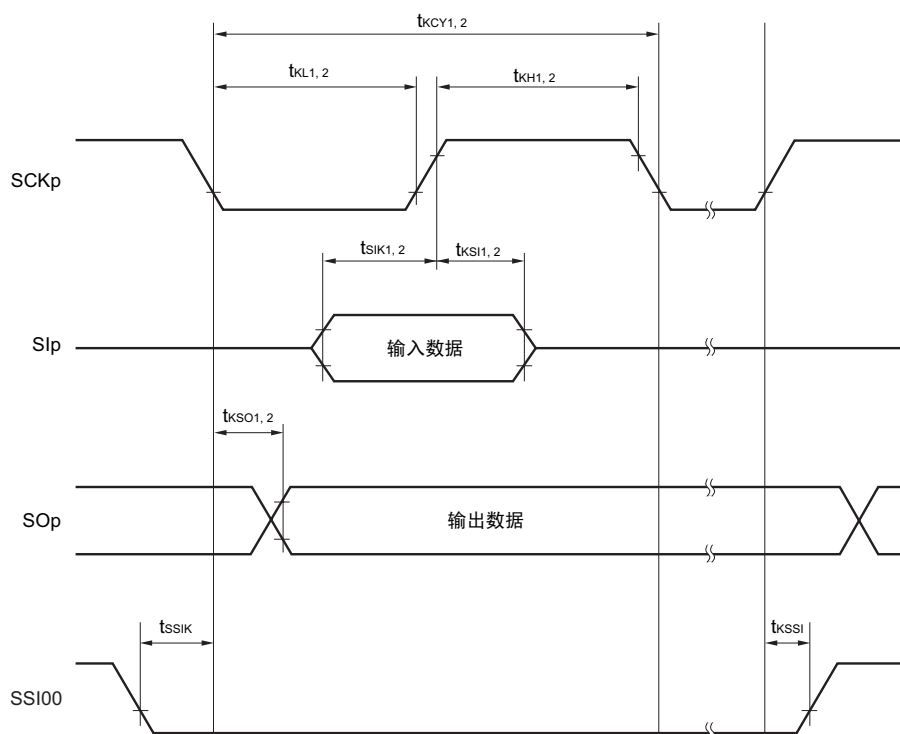
CSI模式的连接图（同电位的通信）  
（从属选择输入功能的从属发送的情况）



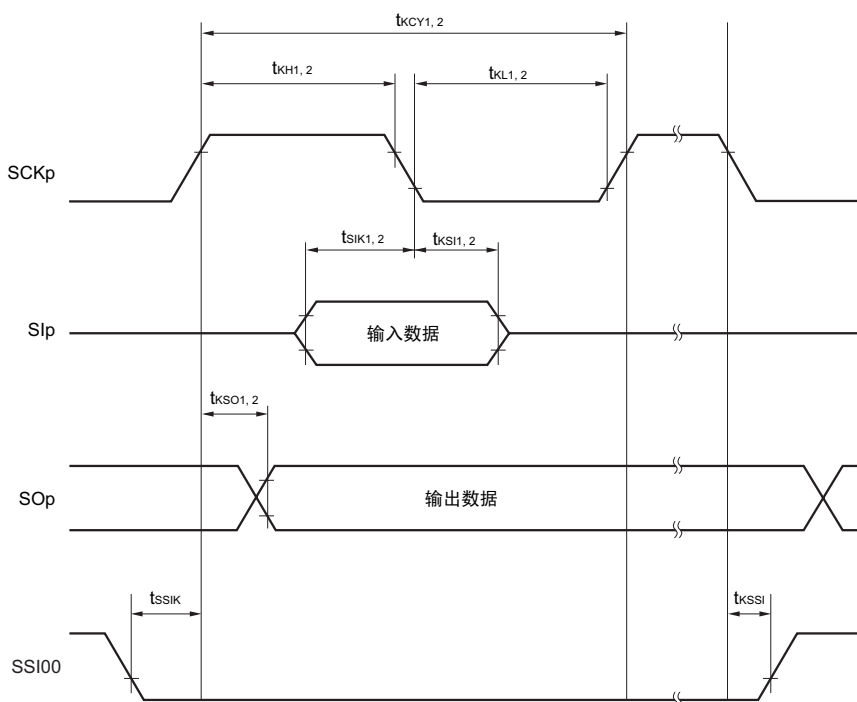
- 备注 1. p: CSI 号 (p=00)  
2. m: 单元号、n: 通道号 (mn=00)



CSI模式的串行传送时序（同电位的通信）  
 （DAPmn=0、CKPmn=0或者DAPmn=1、CKPmn=1的情况）



CSI模式的串行传送时序（同电位的通信）  
 （DAPmn=0、CKPmn=1或者DAPmn=1、CKPmn=0的情况）



备注 1. p: CSI 号 (p=00)

2. m: 单元号、n: 通道号 (mn=00)

## 27.5.2 串行接口 IICA

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

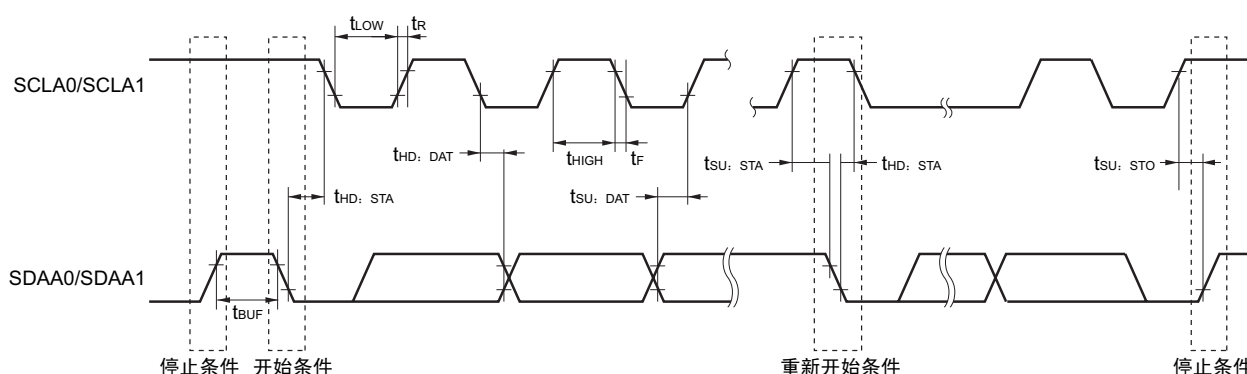
项目	符号	条件	标准模式		快速模式		增强型快速模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLAn 时钟频率	f <sub>SCL</sub>	增强型快速模式: f <sub>CLK</sub> ≥ 10MHz					0	1000	kHz
		快速模式: f <sub>CLK</sub> ≥ 3.5MHz			0	400			kHz
		标准模式: f <sub>CLK</sub> ≥ 1MHz	0	100					kHz
重新开始条件的准备时间	t <sub>SU: STA</sub>		4.7		0.6		0.26		μs
保持时间注 1	t <sub>HD: STA</sub>		4.0		0.6		0.26		μs
SCLAn="L" 的保持时间	t <sub>LOW</sub>		4.7		1.3		0.5		μs
SCLAn="H" 的保持时间	t <sub>HIGH</sub>		4.0		0.6		0.26		μs
数据准备时间 (接收时)	t <sub>SU: DAT</sub>		250		100		50		ns
数据保持时间 (发送时) 注 2	t <sub>HD: DAT</sub>		0	3.45	0	0.9	0	0.45	μs
停止条件的准备时间	t <sub>SU: STO</sub>		4.0		0.6		0.26		μs
总线空闲时间	t <sub>BUF</sub>		4.7		1.3		0.5		μs

注 1. 在开始条件和重新开始条件的情况下, 在此期间之后生成第一个时钟脉冲。

2. t<sub>HD: DAT</sub> 的最大值 (MAX.) 是通常传送时的数值, 在进行应答 (ACK) 时需要等待。备注 1. 各模式的 C<sub>b</sub> (通信线电容) 的 MAX. 值和此时的 R<sub>b</sub> (通信线的上拉电阻值) 的值如下所示:标准模式 : C<sub>b</sub>=400pF、R<sub>b</sub>=2.7kΩ快速模式 : C<sub>b</sub>=320pF、R<sub>b</sub>=1.1kΩ增强型快速模式 : C<sub>b</sub>=120pF、R<sub>b</sub>=1.1kΩ

2. n=0、1

IICA 串行传送时序



## 27.5.3 片上调试 (UART)

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率			115.2k		1M	bps

## 27.6 模拟特性

## 27.6.1 A/D 转换器特性

A/D 转换器特性的区分

基准电压 输入通道	基准电压 (+)=AV <sub>REFP</sub> 基准电压 (−)=AV <sub>REFM</sub>	基准电压 (+)=V <sub>DD</sub> 基准电压 (−)=V <sub>SS</sub>	基准电压 (+)=V <sub>BGR</sub> 基准电压 (−)=AV <sub>REFM</sub>
ANI0 ~ ANI7	参照 27.6.1(1)。	参照 27.6.1(2)。	参照 27.6.1(5)。
ANI16	参照 27.6.1(3)。	参照 27.6.1(4)。	
内部基准电压 温度传感器的输出电压	参照 27.6.1(3)。	参照 27.6.1(4)。	—

- (1) 选择 AV<sub>REF(+)</sub>=AV<sub>REFP</sub>/ANI0 (ADREFP1=0、ADREFP0=1)、AV<sub>REF(−)</sub>=AV<sub>REFM</sub>/ANI1 (ADREFM=1) 的情况, 对象 ANI 引脚: ANI0 ~ ANI7

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V、基准电压 (+)=AV<sub>REFP</sub>、基准电压 (−)=AV<sub>REFM</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		12	bit
综合误差注 1	AINL	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>			±6.0	LSB
转换时间	t <sub>CONV</sub>	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>	3.375		108	μs
零刻度误差注 1、2	E <sub>ZS</sub>	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>			±0.10	%FSR
满刻度误差注 1、2	E <sub>FS</sub>	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>			±0.10	%FSR
积分线性误差注 1	ILE	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>			±2.5	LSB
微分线性误差注 1	DLE	12 位分辨率 AV <sub>REFP</sub> =V <sub>DD</sub>			±1.5	LSB
基准电压 (+)	AV <sub>REFP</sub>		27		V <sub>DD</sub>	V
基准电压 (−)	AV <sub>REFM</sub>		−0.5		0.3	V
模拟输入电压	V <sub>AIN</sub>		0		AV <sub>REFP</sub>	V
	V <sub>BGR</sub>	选择内部基准电压输出。 2.7V ≤ V <sub>DD</sub> ≤ 3.6V、HS (高速主) 模式	1.38	1.45	1.5	V

注 1. 不包含量化误差 (±1/2 LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

- (2) 选择  $AV_{REF(+)} = V_{DD}$  ( $ADREFP1=0$ 、 $ADREFP0=0$ )、 $AV_{REF(-)} = V_{SS}$  ( $ADREFM=0$ ) 的情况，对象 ANI 引脚：ANI0 ~ ANI7

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS}=0\text{V}$ 、基准电压 (+) =  $V_{DD}$ 、基准电压 (-) =  $V_{SS}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		12	bit
综合误差注 1	AINL	10 位分辨率			$\pm 7.0$	LSB
转换时间	$t_{\text{CONV}}$	10 位分辨率	3.375		108	$\mu\text{s}$
零刻度误差注 1、2	$E_{\text{ZS}}$	10 位分辨率			$\pm 0.60$	%FSR
满刻度误差注 1、2	$E_{\text{FS}}$	10 位分辨率			$\pm 0.60$	%FSR
积分线性误差注 1	ILE	10 位分辨率			$\pm 4.0$	LSB
微分线性误差注 1	DLE	10 位分辨率			$\pm 2.0$	LSB
模拟输入电压	$V_{\text{AIN}}$		0		$AV_{\text{REFP}}$	V
	$V_{\text{BGR}}$	选择内部基准电压输出。 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、HS (高速主) 模式	1.38	1.45	1.5	V

注 1. 不包含量化误差 ( $\pm 1/2$  LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

- (3) 选择  $AV_{REF(+)} = AV_{\text{REFP}}/\text{ANI0}$  ( $ADREFP1=0$ 、 $ADREFP0=1$ )、 $AV_{REF(-)} = AV_{\text{REFM}}/\text{ANI1}$  ( $ADREFM=1$ ) 的情况，对象 ANI 引脚：ANI16、内部基准电压、温度传感器的输出电压

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS}=0\text{V}$ 、基准电压 (+) =  $AV_{\text{REFP}}$ 、基准电压 (-) =  $AV_{\text{REFM}}=0\text{V}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		12	bit
综合误差注 1	AINL	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$			$\pm 7.0$	LSB
转换时间	$t_{\text{CONV}}$	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$	4.125		132	$\mu\text{s}$
零刻度误差注 1、2	$E_{\text{ZS}}$	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$			$\pm 0.10$	%FSR
满刻度误差注 1、2	$E_{\text{FS}}$	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$			$\pm 0.10$	%FSR
积分线性误差注 1	ILE	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$			$\pm 3.0$	LSB
微分线性误差注 1	DLE	12 位分辨率 $AV_{\text{REFP}}=V_{DD}$			$\pm 2.0$	LSB
基准电压 (+)	$AV_{\text{REFP}}$		27		$V_{DD}$	V
基准电压 (-)	$AV_{\text{REFM}}$		-0.5		0.3	V
模拟输入电压	$V_{\text{AIN}}$		0		$AV_{\text{REFP}}$	V
	$V_{\text{BGR}}$	选择内部基准电压输出。 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、HS (高速主) 模式	1.38	1.45	1.5	V

注 1. 不包含量化误差 ( $\pm 1/2$  LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

- (4) 选择  $AV_{REF(+)} = V_{DD}$  (ADREFP1=0、ADREFP0=0)、 $AV_{REF(-)} = V_{SS}$  (ADREFM=0) 的情况, 对象 ANI 引脚: ANI16、内部基准电压、温度传感器的输出电压

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS} = 0\text{V}$ 、基准电压 (+) =  $V_{DD}$ 、基准电压 (-) =  $V_{SS}$ )

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8		12	bit
综合误差注 1	AINL	10 位分辨率			$\pm 7.0$	LSB
转换时间	$t_{\text{CONV}}$	10 位分辨率	4.125		132	$\mu\text{s}$
零刻度误差注 1、2	$E_{\text{ZS}}$	10 位分辨率			$\pm 0.60$	%FSR
满刻度误差注 1、2	$E_{\text{FS}}$	10 位分辨率			$\pm 0.60$	%FSR
积分线性误差注 1	ILE	10 位分辨率			$\pm 4.0$	LSB
微分线性误差注 1	DLE	10 位分辨率			$\pm 2.5$	LSB
模拟输入电压	$V_{\text{AIN}}$		0		$V_{\text{DD}}$	V
	$V_{\text{BGR}}$	选择内部基准电压输出。 $2.7\text{V} \leq V_{\text{DD}} \leq 3.6\text{V}$ 、HS (高速主) 模式	1.38	1.45	1.5	V

注 1. 不包含量化误差 ( $\pm 1/2$  LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

- (5) 选择  $AV_{REF(+)} =$  内部基准电压 (ADREFP1=1、ADREFP0=0)、 $AV_{REF(-)} = AV_{\text{REFM}}/\text{ANI1}$  (ADREFM=1) 的情况, 对象 ANI 引脚: ANI0 ~ ANI7、ANI16

( $T_A = -40 \sim +85^{\circ}\text{C}$ 、 $2.7\text{V} \leq V_{DD} \leq 3.6\text{V}$ 、 $V_{SS} = 0\text{V}$ 、基准电压 (+) =  $V_{\text{BGR}}$ 、基准电压 (-) =  $AV_{\text{REFM}} = 0\text{V}$ 、HS (高速主) 模式)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	RES		8			bit
转换时间	$t_{\text{CONV}}$	8 位分辨率	16		108	$\mu\text{s}$
零刻度误差注 1、2	$E_{\text{ZS}}$	8 位分辨率			$\pm 1.60$	%FSR
积分线性误差注 1	ILE	8 位分辨率			$\pm 2.5$	LSB
微分线性误差注 1	DLE	8 位分辨率			$\pm 2.5$	LSB
基准电压 (+)	$V_{\text{BGR}}$		1.38	1.45	1.5	V
模拟输入电压	$V_{\text{AIN}}$		0		$V_{\text{BGR}}$	V

注 1. 不包含量化误差 ( $\pm 1/2$  LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

## 27.6.2 温度传感器 / 内部基准电压特性

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V、HS (高速主) 模式)

项目	符号	条件	MIN.	TYP.	MAX.	单位
温度系数	SL	取决于温度传感器电压的温度。		−3.6		mV/°C
温度系数误差注	E <sub>SL</sub>			43		μV/°C
偏移	OFST25	T <sub>A</sub> =+25°C		1050		mV
偏移误差注	E <sub>OFST25</sub>	T <sub>A</sub> =+25°C		9.0		mV
运行稳定等待时间	t <sub>AMP</sub>				5	μs

注 这是标准偏差。

注意 不进行批量生产的测量。

## 27.6.3 D/A 转换器

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	R <sub>ES</sub>				10	bit
综合误差注 1	AINL	负载电流 =0mA、 0.2 V ≤ 输出电压 ≤ V <sub>DD</sub> −0.2V		±2.0	±4.0	LSB
稳定时间注 1	t <sub>SET</sub>	R <sub>load</sub> =47kΩ、C <sub>load</sub> =20pF			10	μs
D/A 输出电阻注 1	R <sub>O</sub>	每个通道 输出电压 =0V ~ 0.2V 或者 输出电压 =V <sub>DD</sub> −0.2V ~ V <sub>DD</sub>		40	60	Ω
		每个通道 输出电压 =0.2 V ~ V <sub>DD</sub> −0.2V		8	12	Ω
待机电流注 1	I <sub>sb</sub>	D/A 转换器的每个通道			0.1	μA
DAC 工作电流注 1、2	I <sub>DAC</sub>	R <sub>load</sub> =47 kΩ、C <sub>load</sub> =20pF、 V <sub>I</sub> =0.5V <sub>DD</sub> 、 D/A 转换器的每个通道			2	mA

注 1. 不进行批量生产的测量。

2. 这是只流到 D/A 转换器的电流。

## 27.6.4 POR 电路特性

(T<sub>A</sub>=−40 ~ +85°C、V<sub>SS</sub>=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V <sub>POR</sub>	电源电压上升时	1.47	1.51	1.55	V
	V <sub>PDR</sub>	电源电压下降时	1.46	1.50	1.54	V
最小脉宽	T <sub>PW</sub>		300			μs

## 27.6.5 LVD 电路特性

复位模式和中断模式的 LVD 检测电压

(T<sub>A</sub>=−40 ~ +85°C、V<sub>PDR</sub> ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目		符号	条件	MIN.	TYP.	MAX.	单位
检测电压	电源电压电平	V <sub>LVD2</sub>	电源电压上升时	3.07	3.13	3.19	V
			电源电压下降时	3.00	3.06	3.12	V
		V <sub>LVD3</sub>	电源电压上升时	2.96	3.02	3.08	V
			电源电压下降时	2.90	2.96	3.02	V
		V <sub>LVD4</sub>	电源电压上升时	2.86	2.92	2.97	V
			电源电压下降时	2.80	2.86	2.91	V
		V <sub>LVD5</sub>	电源电压上升时	2.76	2.81	2.87	V
			电源电压下降时	2.70	2.75	2.81	V
最小脉宽		t <sub>LW</sub>	300			μs	
检测延迟					300	μs	

中断&amp;复位模式的 LVD 检测电压

(T<sub>A</sub>=−40 ~ +85°C、V<sub>PDR</sub> ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

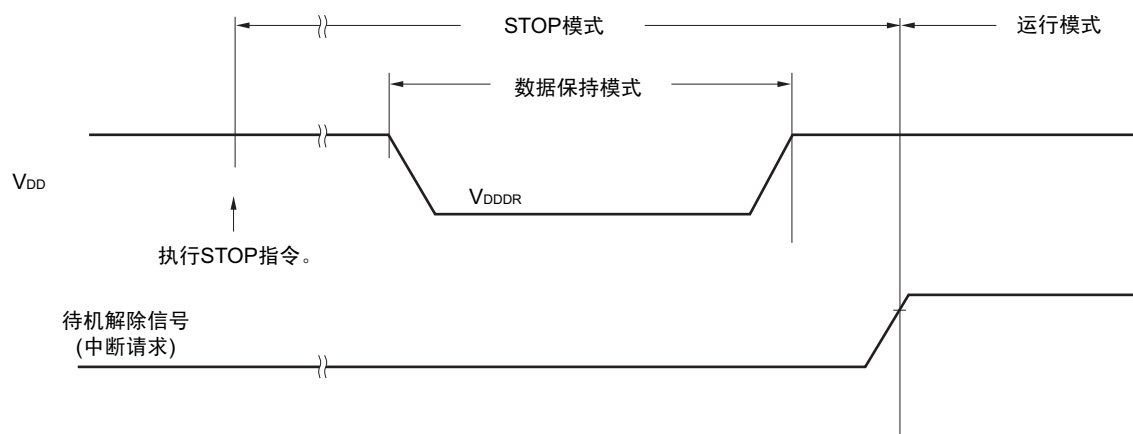
项目	符号	条件		MIN.	TYP.	MAX.	单位
中断 & 复位 模式	V <sub>LVD5</sub>	V <sub>POC2</sub> 、V <sub>POC1</sub> 、V <sub>POC0</sub> =0、1、1，下降复位电压		2.70	2.75	2.81	V
	V <sub>LVD4</sub>	LVIS1、LVIS0=1、0 (+0.1V)	上升复位解除电压	2.86	2.92	2.97	V
			下降中断电压	2.80	2.86	2.91	V
	V <sub>LVD3</sub>	LVIS1、LVIS0=0、1 (+0.2V)	上升复位解除电压	2.96	3.02	3.08	V
			下降中断电压	2.90	2.96	3.02	V

## 27.7 STOP 模式中的数据存储器的低电源电压数据保持特性

(T<sub>A</sub>=−40 ~ +85°C)

项目	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电源电压	V <sub>DDDR</sub>		1.46 注		3.6	V

注 取决于POR检测电压。当电压下降时，保持数据，直到发生POR复位为止。但是，当发生POR复位时，不保持数据。



## 27.8 闪存编程特性

(T<sub>A</sub>=−40 ~ +85°C、2.7V ≤ V<sub>DD</sub> ≤ 3.6V、V<sub>SS</sub>=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位
系统时钟频率	f <sub>CLK</sub>	2.7V ≤ V <sub>DD</sub> ≤ 3.6V		1		32	MHz
代码闪存的改写次数注 1、2、3	C <sub>erwr</sub>	1 次改写是指 1 次擦除 + 擦除后的 1 次编程。	保持年数: 20 年	1000			次
数据闪存的改写次数注 1、2、3			保持年数: 1 年		1000000		
			保持年数: 5 年	100000			
工作温度		编程时		0 ~ 40			°C
		自编程时		-40 ~ +85			°C

注 1. 1 次改写是指 1 次擦除 + 擦除后的 1 次编程，保持年数是指从 1 次改写到下一次改写为止的期间。

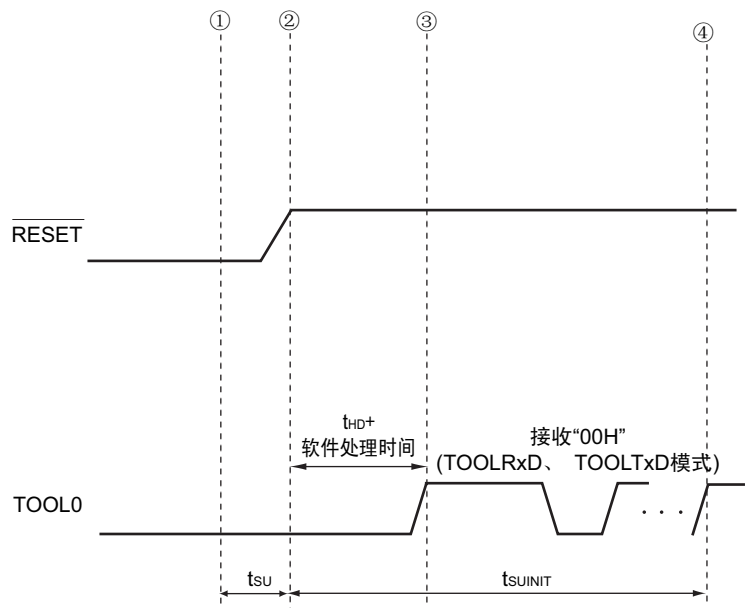
2. 这是使用闪存编程器和本公司提供的库程序的情况。

3. 此特性表示闪存的特性，是本公司的可靠性试验的结果。



27.9 闪存编程模式的进入时序

项目	符号	条件	MIN.	TYP.	MAX.	单位
从解除外部复位到完成初始设定通信为止的时间	$t_{\text{SUINIT}}$	在解除外部复位前，解除 POR 和 LVD 的复位。			100	ms
从将TOOL0引脚置为低电平到解除外部复位为止的时间	$t_{\text{SU}}$	在解除外部复位前，解除 POR 和 LVD 的复位。	10			$\mu\text{s}$
在解除复位后保持TOOL0引脚低电平的时间 (软件处理时间除外)	$t_{\text{HD}}$	在解除外部复位前，解除 POR 和 LVD 的复位。	1			ms



- ① 给TOOL0引脚输入低电平。
- ② 解除外部复位（在此之前需要解除POR和LVD的复位）。
- ③ 解除TOOL0引脚的低电平。
- ④ 通过UART接收进入闪存编程模式，完成波特率的设定。

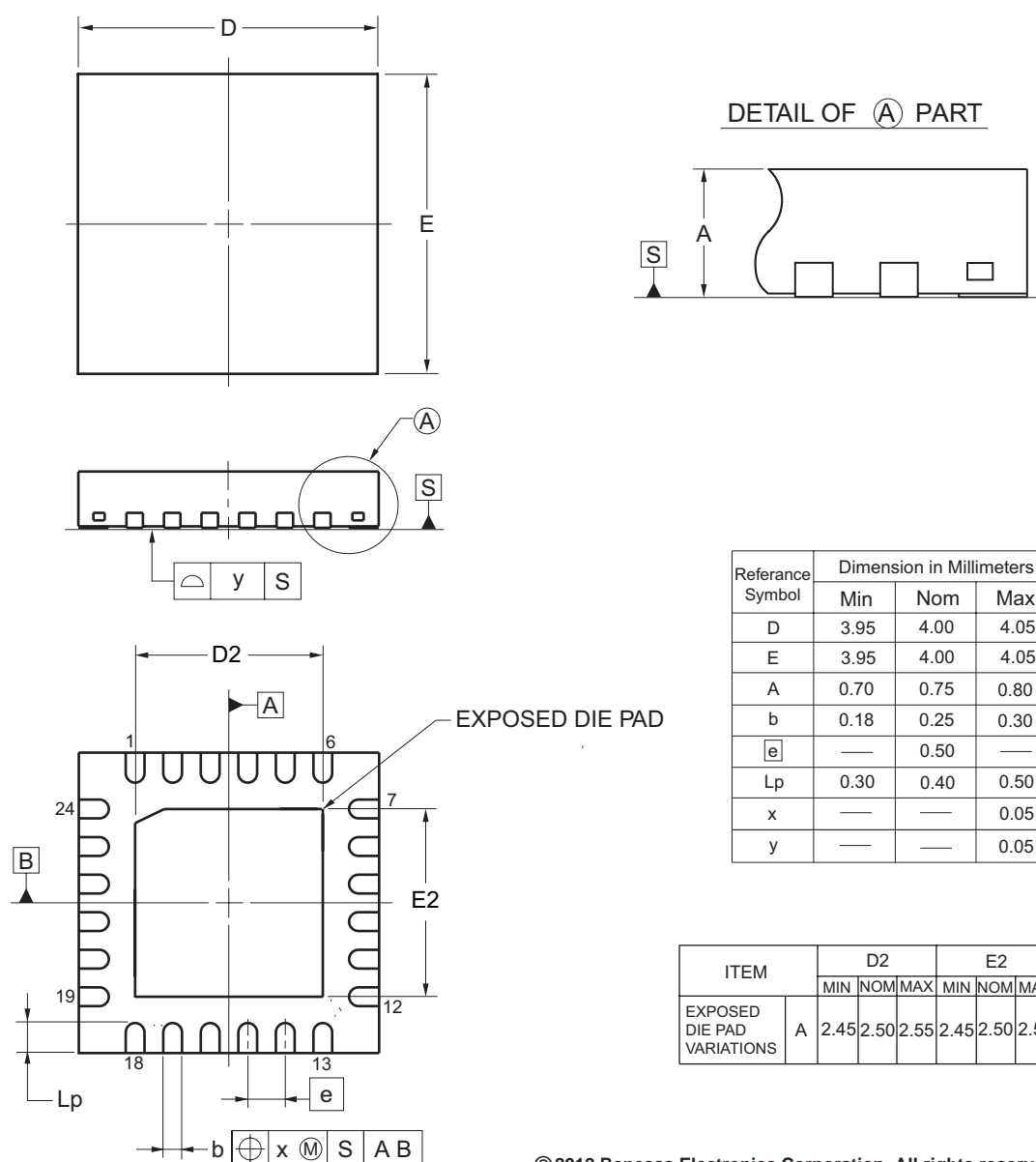
备注  $t_{\text{SUINIT}}$ : 在此区间，必须在解除复位后的 100ms 之内完成初始设定的通信。  
 $t_{\text{SU}}$  : 这是从将TOOL0引脚置为低电平到解除外部复位为止的时间。  
 $t_{\text{HD}}$  : 这是在解除外部复位或者内部复位后保持TOOL0引脚低电平的时间（软件处理时间除外）。

## 第 28 章 封装尺寸图

## 28.1 24 引脚产品

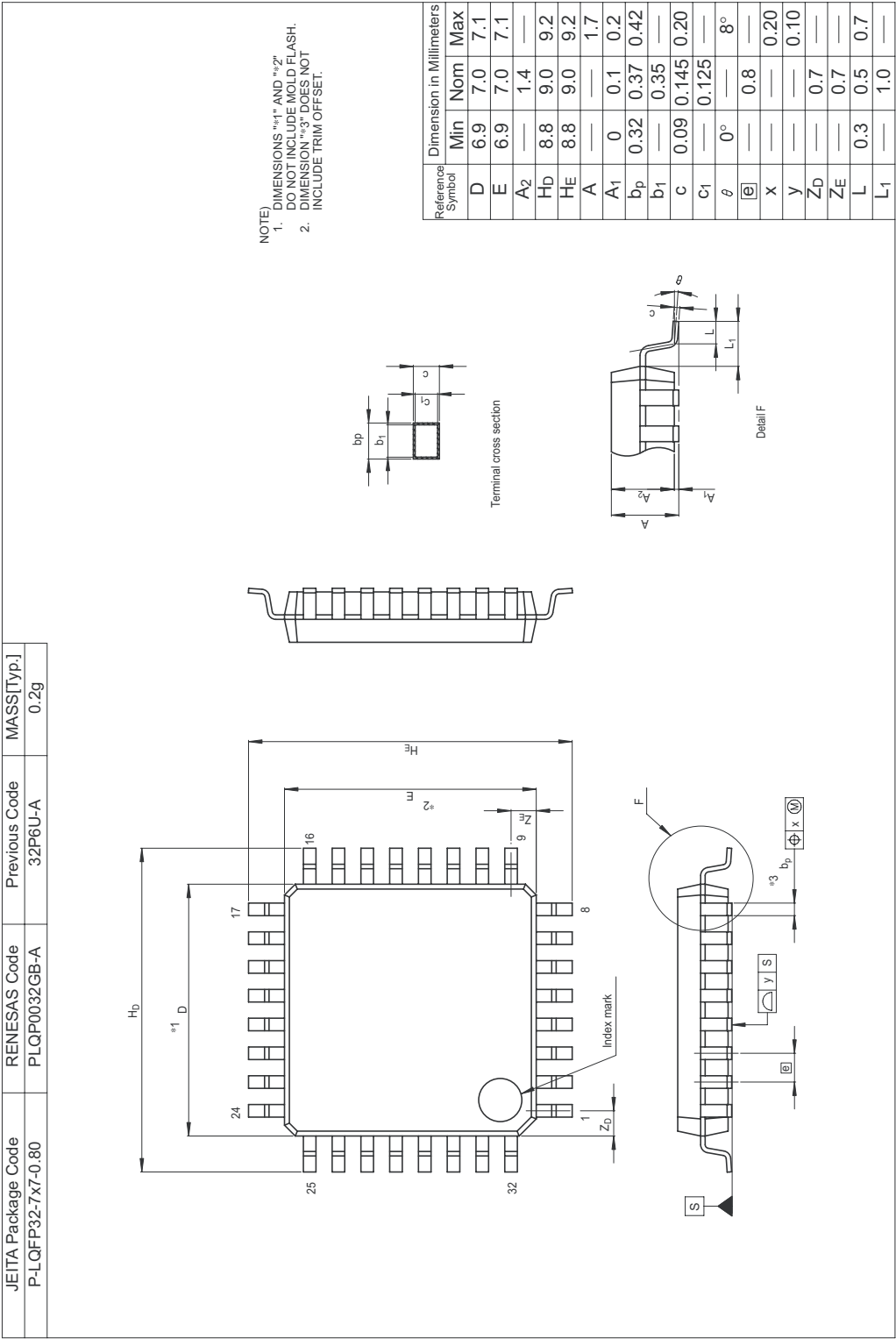
## R7F0C01072DNP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-HWQFN24-4x4-0.50	PWQN0024KE-A	P24K8-50-CAB-2	0.04



©2012 Renesas Electronics Corporation. All rights reserved.

28.2 32 引脚产品  
R7F0C010B2DFP-C



## 附录 A 修订记录

Rev.	发行日	修订内容	
		页	修订处
1.00	2013.09.20	—	初版发行
2.00	2014.10.24	1、 2	更改“1.1 特点”和“1.2 产品型号一览表”的内容。
		6、 7	更改“1.5 框图”的内容。
		10	更改“2.1 引脚功能一览表”的内容。
		12	更改“2.2.1 各产品配置的功能”的内容。
		13	追加“2.2.2 功能说明”的内容。
		25	更改图 3-1。
		32	更改“3.1.3 内部数据存储空间”的内容。
		33	更改图 3-3。
		35	更改“3.2.1 控制寄存器”(3)的说明和图 3-6。
		36	更改“3.2.2 通用寄存器”的内容
		37	追加图 3-9。
		38	更改“3.2.4 特殊功能寄存器（SFR: Special Function Register）”的内容。
		42	更改“3.2.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）”的内容。
		43	更改“表 3-6 扩展 SFR（2nd SFR）一览表 (1/4)”的内容。
		52	更改图 3-20。
		63	更改“4.1 端口功能”的内容。
		85	更改“4.3 控制端口功能的寄存器”的内容。
		90	更改图 4-19 和图 4-20。
		97	更改“5.1 时钟发生电路的功能”的“(1) 主系统时钟”的内容。
		100	更改“5.3 控制时钟发生电路的寄存器”的内容。
		105	更改图 5-5。
		106	更改“5.3.5 振荡稳定时间选择寄存器（OSTS）”的内容和图 5-6。
		109	更改“5.3.7 高速内部振荡器的频率选择寄存器（HOCODIV）”的内容。
		110	更改“5.3.8 高速内部振荡器的微调寄存器（HIOTRM）”的内容。
		114	更改“5.4.3 低速内部振荡器”的内容。
		116	更改图 5-13。
		119	更改图 5-14。
		120、 121	更改表 5-3 的 (1/3) 和 (2/3) 的内容。
		123	更改表 5-4 的内容。
		125	更改“5.7 谐振器和振荡电路常数”的内容。
		126	追加第 6 章开始的表的内容。
		130	更改表 6-1 的内容。
		132、 133	更改图 6-2 ～图 6-5。
		137	更改“6.3 控制定时器阵列单元的寄存器”的内容。
		138	更改图 6-9。
		139	更改“6.3.2 定时器时钟选择寄存器 m（TPSm）”的内容。
		140、 141	更改图 6-10。
		144	更改图 6-11 的 (2/4)。
		150	更改图 6-15。
		152	更改图 6-17。
		156	更改图 6-21。

Rev.	发行日	修订内容	
		页	修订处
2.00	2014.10.24	165	更改图 6-26。
		166	更改图 6-27。
		177、178	追加“6.7 定时器输入 (TImn) 的控制”的内容。
		181	更改图 6-43。
		182	更改图 6-44 的 (1/2)。
		184	更改图 6-45。
		186	更改图 6-47。
		187	更改图 6-48。
		188	更改“6.8.3 作为输入脉冲间隔测量的运行”的内容和图 6-49。
		191	更改图 6-52。
		193	更改图 6-53。
		195	更改图 6-56。
		196	更改图 6-57。
		198	更改图 6-59。
		199	更改图 6-60。
		200 ~ 213	将“p: 从属通道号 ( $n < p \leq 3$ )”更改为“p: 从属通道号 ( $n=0: p=1、2、3, n=2: p=3$ )”。
		201	更改图 6-61。
		203	更改图 6-63。
		205、206	更改图 6-65。
		210	更改图 6-68。
		211	更改图 6-69。
		218	更改图 6-73。
		225	更改“7.3 控制时钟输出 / 蜂鸣器输出控制电路的寄存器”的内容。
		226	更改“7.3.2 控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器”的内容。
		227	更改“7.4.1 输出引脚的运行”和“7.5 时钟输出 / 蜂鸣器输出控制电路的注意事项”的内容。
		228	更改“8.1 看门狗定时器的功能”的内容。
		229	更改表 8-1 的内容和图 8-1。
		231	更改“8.4.1 看门狗定时器的运行控制”的内容。
		232	更改表 8-3 的内容。
		233	更改“8.4.3 看门狗定时器窗口打开期间的设定”的表的内容。
		235	更改“9.1 A/D 转换器的功能”的内容。
		236	更改图 9-1。
		240	更改图 9-2。
		242	更改表 9-1 的内容。
		243	更改图 9-4。
		244 ~ 247	更改表 9-3 的内容。
		249	更改图 9-6。
		250	更改“9.3.4 A/D 转换器的模式寄存器 2 (ADM2)”的内容。
		252	更改“9.3.5 12 位 A/D 转换结果寄存器 (ADCR)”的内容。
		255	更改图 9-11 的 (2/2)。
		256	更改“9.3.8 转换结果比较上限值设定寄存器 (ADUL)”的内容和图 9-13。
		257	更改“9.3.10 A/D 测试寄存器 (ADTES)”和“9.3.11 控制模拟输入引脚端口功能的寄存器”的内容。
		259	更改图 9-15。

Rev.	发行日	修订内容	
		页	修订处
2.00	2014.10.24	261 ~ 272	更改图 9-17 ~ 图 9-28。
		273 ~ 275	更改图 9-29 ~ 图 9-31。
		276	更改“9.7.4 选择温度传感器的输出电压 / 内部基准电压时的设定（以软件触发模式、单次转换模式为例）”的内容。
		277	更改图 9-33。
		281	追加图 9-37。
		283	更改图 9-40 和图 9-41。
		286	更改“(6) 有关模拟输入（ANIn）引脚的输入阻抗”的内容。
		291	更改图 10-2。
		298	更改第 11 章的说明内容和“11.1.1 3 线串行 I/O（CSI00）”的内容。
		299	更改“11.1.2 UART（UART0）”的内容。
		302	更改“11.2.2 串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位”的内容。
		305	更改图 11-3。
		306	更改图 11-4。
		309、310	更改图 11-6 的 (1/2) 和 (2/2)。
		311	更改“11.3.5 串行数据寄存器 mn（SDRmn）的高 7 位”的内容。
		314	更改图 11-9 的 (2/2)。
		319	更改“11.3.12 串行输出寄存器 m（SOM）”的内容和图 11-14。
		320	更改图 11-15，追加图 11-16。
		321	更改“11.3.14 串行待机控制寄存器 m（SSCm）”的内容和图 11-17，追加图 11-18。
		327	更改“11.5 3 线串行 I/O（CSI00）通信的运行”的内容。
		328	更改“11.5.1 主控发送”的内容。
		332 ~ 335	更改图 11-28 ~ 图 14-31。
		336	更改“11.5.2 主控接收”的内容。
		338 ~ 343	更改图 11-33 ~ 图 11-39。
		344	更改“11.5.3 主控发送和接收”的内容。
		346 ~ 351	更改图 11-41 ~ 图 11-47。
		352	更改“11.5.4 从属发送”的内容。
		354 ~ 359	更改图 11-50 ~ 图 11-55。
		360	更改“11.5.5 从属接收”的内容。
		362 ~ 365	更改图 11-57 ~ 图 11-61。
		366	更改“11.5.6 从属发送和接收”的内容。
		368 ~ 373	更改图 11-64 ~ 图 11-69。
		374	更改“11.5.7 SNOOZE 模式功能”的内容。
		376、377	更改图 11-72 和图 11-73。
		381	更改“11.6 从属选择输入功能的时钟同步串行通信的运行”的内容。
		387 ~ 393	更改图 11-78 ~ 图 11-84。
		394	更改“11.6.2 从属接收”的内容。
		397 ~ 400	更改图 11-86 ~ 图 11-90。
		401	更改“11.6.3 从属发送和接收”的内容。
		404 ~ 410	更改图 11-92 ~ 图 11-98。
		411	更改“11.6.4 传送时钟频率的计算”的内容。
		413	更改“11.7 UART（UART0）通信的运行”的内容。
		414	更改“11.7.1 UART 发送”的内容。
		417 ~ 422	更改图 11-101 ~ 图 11-107。

Rev.	发行日	修订内容	
		页	修订处
2.00	2014.10.24	423	更改“11.7.2 UART 接收”的内容。
		425 ~ 429	更改图 11-108 的 (2/2) ~ 图 11-113。
		430	更改“11.7.3 SNOOZE 模式功能”的内容和图 11-114。
		431	更改图 11-115。
		433	更改图 11-117。
		546	更改“14.1 ELC 的功能”和“14.2 ELC 的结构”的内容。
		547	更改“14.3 控制 ELC 的寄存器”的内容。
		549	更改“14.4 运行说明”的内容。
		550	更改第 15 章的说明内容。
		555	更改表 15-2 的内容。
		556	更改“15.3.1 中断请求标志寄存器 (IF0L、IF0H、IF1L)”的内容。
		557	更改图 15-3。
		558	更改图 15-4。
		559	更改表 15-3 的内容。
		560	更改“15.3.5 程序状态字 (PSW)”的内容。
		563	更改图 15-8 和图 15-9。
		565	更改表 15-5 的内容。
		568	更改“15.4.4 中断请求的保留”的内容。
		573	更改“16.3.1 HALT 模式”的内容。
		577	更改“16.3.2 STOP 模式”的内容。
		579、580	更改图 16-5 的 (1/2) 和 (2/2)。
		582	更改“16.3.3 SNOOZE 模式”的内容。
		584	更改第 17 章的说明内容。
		586	更改“17.1 复位时序”的内容。
		588	更改表 17-1 的内容。
		592	更改图 17-4。
		594	追加图 17-5。
		595	更改“18.1 上电复位电路的功能”的内容。
		596	更改图 18-2 的 (1/3)。
		598	更改图 18-2 的 (3/3)。
		601	更改“19.1 电压检测电路的功能”的内容。
		602	更改图 19-1。
		603	更改图 19-2。
		604	更改图 19-3。
		605	更改表 19-1 的内容。
		606	更改“19.4.1 用作复位模式时的设定”的内容。
		607	更改图 19-4。
		608	更改“19.4.2 用作中断模式时的设定”的内容。
		609	更改图 19-5。
		610	更改“19.4.3 用作中断 & 复位模式时的设定”的内容。
		611 ~ 614	更改图 19-6 的 (1/2) 和 (2/2) 及图 19-7。
		616	更改“19.5 电压检测电路的注意事项”的内容。
		618	更改“20.1 安全功能的概要”的内容。
		619	更改“20.3 闪存 CRC 运算功能 (高速 CRC) 的运行”的内容。
		621	更改图 20-3。
		625	追加图 20-8。
		628	更改图 20-11。

Rev.	发行日	修订内容	
		页	修订处
2.00	2014.10.24	631	更改“20.9.1 定时器输入选择寄存器 0 (TIS0)”的内容。
		632	更改“20.10 A/D 测试功能”的内容和图 20-15。
		634	更改图 20-17。
		636	更改“22.1 选项字节的功能”和“22.1.1 用户选项字节 (000C0H ~ 000C2H)”的内容。
		638	更改图 22-1。
		639	更改图 22-2。
		640	更改图 22-3。
		644	更改“23.1 使用闪存编程器的串行编程”。
		646	更改图 23-2。
		647	更改“23.2 使用外部器件 (内置 UART) 的串行编程”的内容。
		649	更改“23.3 电路板上的引脚处理”和“23.3.1 P40/TOOL0 引脚”的内容。
		650	更改“23.3.4 REGC 引脚”的内容。
		654	更改“23.5.1 闪存的控制”的内容。
		655	更改“23.5.2 串行编程模式”的内容
		656	更改“23.5.3 通信方式”的内容。
		657	更改表 23-7 的内容。
		659	更改“23.6 安全设定”的内容。
		661	更改“23.7 通过自编程进行的闪存编程”的内容。
		665	更改“23.8 使用 PG-FP5 时的各命令处理时间 (参考值)”的内容。
		672	更改表 26-1 的内容。
		673	更改表 26-2 的内容。
		684	更改表 26-5(10/18) 的内容。
		697	更改“27.3.1 引脚特性”的内容。
		703	更改“27.4 AC 特性”的内容。
		704 ~ 706	更改“主系统时钟运行时的最短指令执行时间”~“TI/TO 时序”。
		707 ~ 710	更改“27.5 外围功能特性”的内容。
		714 ~ 716	更改“27.6.1 A/D 转换器特性”的内容。
		717	更改“27.6.4 POR 电路特性”的内容。
		719	更改“27.7 STOP 模式中的数据存储器器的低电源电压数据保持特性”的内容。



---

R7F0C01072DNP、R7F0C010B2DFP-C  
用户手册 硬件篇

Publication Date: Rev.2.00 Oct 24, 2014

Published by: Renesas Electronics Corporation

---



## SALES OFFICES

## Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

### **Renesas Electronics America Inc.**

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

### **Renesas Electronics Canada Limited**

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

### **Renesas Electronics Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

### **Renesas Electronics Europe GmbH**

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

### **Renesas Electronics (China) Co., Ltd.**

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

### **Renesas Electronics (Shanghai) Co., Ltd.**

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

### **Renesas Electronics Hong Kong Limited**

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

### **Renesas Electronics Taiwan Co., Ltd.**

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

### **Renesas Electronics Singapore Pte. Ltd.**

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

### **Renesas Electronics Malaysia Sdn.Bhd.**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

### **Renesas Electronics Korea Co., Ltd.**

12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R7F0C01072DNP、 R7F0C010B2DFP-C



瑞萨电子株式会社

R01UH0422CJ0200