

RL78/G10

用户手册 硬件篇

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 本手册以理解 RL78/G10 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。
对象产品如下所示：

- 10 引脚：R5F10Y1x (x=4、7、6)
- 16 引脚：R5F10Y4x (x=4、6、7)

目 的 本手册以帮助用户理解下述结构中所示的功能为目的。

构 成 RL78/G10 的用户手册分为用户手册硬件篇（本手册）和用户手册软件篇（RL78 族通用）共 2 本。



- 引脚功能
- 内部块功能
- 中断
- 其他的内部外围功能
- 电特性

- CPU 功能
- 指令集
- 指令的说明

阅读方法 阅读本手册的读者应具备电气、逻辑电路以及微控制器的基础知识。

- 要理解全部功能时
→ 请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 RL78/G10 微控制器的指令功能时
→ 请参照另一本手册《RL78 family User's Manual: Software》（R01US0015E）。

凡 例 数据表示法： 左侧为高位，右侧为低位。
有效低电平表示法： xxx （在引脚或者信号名称上标注上划线）
注： 正文中加“注”的说明
注意： 需要留心阅读的内容
备注： 正文的补充说明
数制表示法： 二进制 XXXX 或者 XXXXB
 十进制 XXXX
 十六进制 XXXXH

相关资料 相关资料中可能包括暂定版。但是，在以下资料中并未特别注明“暂定版”，请谅解。

产品的相关资料

资料名称	资料号
RL78/G10 用户手册 硬件篇	本手册
RL78 family User's Manual: Software	R01US0015E

闪存编程器的相关资料（用户手册）

资料名称	资料号
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

其他资料

资料名称	资料号
Renesas MPUs & MCUs RL78 Family	R01CP0003E
半导体封装安装手册	注
Quality Grades on NEC Semiconductor Devices	C11531E
Guide to Prevent Damage for Semiconductor Derices by Electrostatic Discharge (ESD)	C11892E
Semiconductor Reliability Handbook	R51ZZ0001E

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

注意 上述相关资料的内容如有变更，恕不另行通知。设计等时请使用最新版本的资料。

所有商标及注册商标分别归属于其所有者。

本用户手册仅为参考译文，对应的日文版和英文版具有正式效力。

EEPROM 是瑞萨电子株式会社的注册商标。

Windows、Windows NT 和 Window XP 是美国 Microsoft Corporation 在美国以及其他国家的注册商标或者商标。

PC/AT 是美国 IBM 公司的商标。

SuperFlash 是美国 Silicon Storage Technology, Inc. 在美国以及日本等国的注册商标。

注意：本产品使用已获得 Silicon Storage Technology, Inc. 授权的 SuperFlash®。

目 录

第 1 章 概述	1
1.1 特点	1
1.2 产品型号一览表	3
1.3 引脚连接图（俯视图）	4
1.3.1 10 引脚产品	4
1.3.2 16 引脚产品	4
1.4 引脚名称	5
1.5 框图	6
1.5.1 10 引脚产品	6
1.5.2 16 引脚产品	7
1.6 功能概要	8
第 2 章 引脚功能	9
2.1 端口功能	9
2.1.1 10 引脚产品	9
2.1.2 16 引脚产品	10
2.2 端口以外的功能	11
2.2.1 各产品内置的功能	11
2.2.2 功能说明	12
2.3 未使用引脚的处理	13
2.4 引脚框图	14
第 3 章 CPU 体系结构	20
3.1 存储空间	21
3.1.1 内部程序存储空间	24
3.1.2 镜像区	26
3.1.3 内部数据存储空间	27
3.1.4 特殊功能寄存器（SFR: Special Function Register）的区域	27
3.1.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）的区域	27
3.1.6 数据存储器的寻址	28
3.2 处理器的寄存器	29
3.2.1 控制寄存器	29
3.2.2 通用寄存器	31
3.2.3 ES 寄存器和 CS 寄存器	32
3.2.4 特殊功能寄存器（SFR: Special Function Register）	33
3.2.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）	36
3.3 指令地址的寻址	39
3.3.1 相对寻址	39
3.3.2 立即寻址	39
3.3.3 表间接寻址	40
3.3.4 寄存器直接寻址	40
3.4 处理数据地址的寻址	41
3.4.1 隐含寻址	41
3.4.2 寄存器寻址	41
3.4.3 直接寻址	42
3.4.4 短直接寻址	43
3.4.5 SFR 寻址	44
3.4.6 寄存器间接寻址	45
3.4.7 基址寻址	46

3.4.8	基址变址寻址	50
3.4.9	堆栈寻址	51
第 4 章	端口功能	55
4.1	端口功能	55
4.2	端口结构	55
4.2.1	端口 0	56
4.2.2	端口 4	56
4.2.3	端口 12	56
4.2.4	端口 13	56
4.3	控制端口功能的寄存器	57
4.3.1	端口模式寄存器 0、4 (PM0、PM4)	58
4.3.2	端口寄存器 0、4、12、13 (P0、P4、P12、P13)	59
4.3.3	上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12)	60
4.3.4	端口输出模式寄存器 0 (POM0)	61
4.3.5	端口模式控制寄存器 0 (PMC0)	62
4.3.6	外围 I/O 重定向寄存器 (PIOR)	63
4.4	端口功能的运行	64
4.4.1	输入 / 输出端口的写操作	64
4.4.2	输入 / 输出端口的读操作	64
4.4.3	输入 / 输出端口的运算	64
4.5	使用复用功能时的寄存器设定	65
4.5.1	使用复用功能时的基本观点	65
4.5.2	不使用输出功能的复用功能的寄存器设定	66
4.5.3	使用的端口功能和复用功能的寄存器设定例子	66
4.6	功能时的注意事项	70
4.6.1	有关对端口寄存器 n (Pn) 的位操作指令的注意事项	70
4.6.2	引脚设定的注意事项	70
第 5 章	时钟发生电路	71
5.1	时钟发生电路的功能	71
5.2	时钟发生电路的结构	72
5.3	时钟发生电路的控制寄存器	74
5.3.1	时钟运行模式控制寄存器 (CMC)	75
5.3.2	系统时钟控制寄存器 (CKC)	76
5.3.3	时钟运行状态控制寄存器 (CSC)	77
5.3.4	振荡稳定时间计数器的状态寄存器 (OSTC)	78
5.3.5	振荡稳定时间选择寄存器 (OSTS)	79
5.3.6	外围允许寄存器 0 (PER0)	80
5.3.7	运行速度模式控制寄存器 (OSMC)	81
5.3.8	高速内部振荡器的频率选择寄存器 (HOCODIV)	82
5.4	系统时钟振荡电路	83
5.4.1	X1 振荡电路 (只限于 16 引脚产品)	83
5.4.2	高速内部振荡器	86
5.4.3	低速内部振荡器	86
5.5	时钟发生电路的运行	86
5.6	时钟控制	88
5.6.1	高速内部振荡器的设定例子	88
5.6.2	X1 振荡电路的设定例子	89
5.6.3	CPU 时钟的状态转移图	90
5.6.4	CPU 时钟转移前的条件和转移后的处理	93
5.6.5	切换 CPU 时钟和主系统时钟所需要的时间	93

5.6.6	时钟振荡停止前的条件	94
5.7	谐振器和振荡电路常数	94
第 6 章	 定时器阵列单元	95
6.1	定时器阵列单元的功能	97
6.1.1	独立通道运行功能	97
6.1.2	多通道联动运行功能	99
6.1.3	8 位定时器运行功能（只限于通道 1 和通道 3）	100
6.2	定时器阵列单元的结构	101
6.2.1	定时器计数寄存器 0n（TCR0n）	105
6.2.2	定时器数据寄存器 0n（TDR0n）	107
6.3	定时器阵列单元的控制寄存器	109
6.3.1	外围允许寄存器 0（PER0）	110
6.3.2	定时器时钟选择寄存器 0（TPS0）	111
6.3.3	定时器模式寄存器 0n（TMR0n）	112
6.3.4	定时器状态寄存器 0n（TSR0n）	117
6.3.5	定时器通道允许状态寄存器 0（TE0、TEH0（8 位模式））	118
6.3.6	定时器通道开始寄存器 0（TS0、TSH0（8 位模式））	119
6.3.7	定时器通道停止寄存器 0（TT0、TTH0（8 位模式））	120
6.3.8	定时器输出允许寄存器 0（TOE0）	121
6.3.9	定时器输出寄存器 0（TO0）	122
6.3.10	定时器输出电平寄存器 0（TOL0）	123
6.3.11	定时器输出模式寄存器 0（TOM0）	124
6.3.12	噪声滤波器允许寄存器 1（NFEN1）	125
6.3.13	输入切换控制寄存器（ISC）	126
6.3.14	定时器输入 / 输出引脚的端口功能控制寄存器	127
6.4	定时器阵列单元的基本规则	128
6.4.1	多通道联动运行功能的基本规则	128
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）	130
6.5	计数器的运行	131
6.5.1	计数时钟（ f_{TCLK} ）	131
6.5.2	计数器的开始时序	133
6.5.3	计数器的运行	134
6.6	通道输出（TO0n 引脚）的控制	139
6.6.1	TO0n 引脚输出电路的结构	139
6.6.2	TO0n 引脚的输出设定	140
6.6.3	通道输出运行的注意事项	141
6.6.4	TO0n 位的一次性操作	145
6.6.5	有关开始计数时的定时器中断和 TO0n 引脚输出	146
6.7	定时器输入（TI0n）的控制	147
6.7.1	TI0n 的输入电路结构	147
6.7.2	噪声滤波器	147
6.7.3	进行通道输入操作时的注意事项	148
6.8	定时器阵列单元的独立通道运行功能	149
6.8.1	作为间隔定时器 / 方波输出的运行	149
6.8.2	作为外部事件计数器的运行	153
6.8.3	作为分频器的运行（只限于通道 0 和通道 3）	157
6.8.4	作为输入脉冲间隔测量的运行	161
6.8.5	作为输入信号高 / 低电平宽度测量的运行	165
6.8.6	作为延迟计数器的运行	169
6.9	定时器阵列单元的多通道联动运行功能	173
6.9.1	作为单触发脉冲输出的运行	173

6.9.2	作为双输入式单触发脉冲输出的运行	180
6.9.3	作为 PWM 输出功能的运行	187
6.9.4	作为多重 PWM 输出功能的运行	194
6.10	使用定时器阵列单元时的注意事项	200
6.10.1	使用定时器输出时的注意事项	200
第 7 章	12 位间隔定时器	201
7.1	12 位间隔定时器的功能	201
7.2	12 位间隔定时器的结构	201
7.3	12 位间隔定时器的控制寄存器	202
7.3.1	外围允许寄存器 0 (PER0)	202
7.3.2	运行速度模式控制寄存器 (OSMC)	203
7.3.3	间隔定时器的控制寄存器 (ITMCH、ITMCL)	204
7.4	12 位间隔定时器的运行	205
7.4.1	12 位间隔定时器的运行时序	205
7.4.2	从 HALT/STOP 模式返回后计数器开始运行并且再次转移到 HALT/STOP 模式的设定步骤	206
第 8 章	时钟输出 / 蜂鸣器输出控制电路	207
8.1	时钟输出 / 蜂鸣器输出控制电路的功能	207
8.2	时钟输出 / 蜂鸣器输出控制电路的结构	208
8.3	时钟输出 / 蜂鸣器输出控制电路的控制寄存器	208
8.3.1	时钟输出选择寄存器 0 (CKS0)	209
8.3.2	时钟输出 / 蜂鸣器输出引脚的端口功能的控制寄存器	210
8.4	时钟输出 / 蜂鸣器输出控制电路的运行	211
8.4.1	输出引脚的运行	211
第 9 章	看门狗定时器	212
9.1	看门狗定时器的功能	212
9.2	看门狗定时器的结构	213
9.3	看门狗定时器的控制寄存器	214
9.3.1	看门狗定时器的允许寄存器 (WDTE)	214
9.4	看门狗定时器的运行	215
9.4.1	看门狗定时器的运行控制	215
9.4.2	看门狗定时器上溢时间的设定	216
第 10 章	A/D 转换器	217
10.1	A/D 转换器的功能	217
10.2	A/D 转换器的结构	219
10.3	A/D 转换器的控制寄存器	221
10.3.1	外围允许寄存器 0 (PER0)	221
10.3.2	A/D 转换器的模式寄存器 0 (ADM0)	222
10.3.3	A/D 转换器的模式寄存器 2 (ADM2)	225
10.3.4	A/D 转换器结果高位保存寄存器 (ADCRH)	226
10.3.5	A/D 转换结果低位保存寄存器 (ADCRL)	226
10.3.6	模拟输入通道指定寄存器 (ADS)	227
10.3.7	A/D 测试寄存器 (ADTES)	228
10.3.8	模拟输入引脚的端口功能的控制寄存器	228
10.4	A/D 转换器的转换运行	229
10.5	输入电压和转换结果	230
10.6	A/D 转换器的运行模式	231
10.7	A/D 转换器的设定流程图	232
10.7.1	A/D 转换对象为 ANI0 ~ ANI6 的设定	232

10.7.2	A/D 转换对象为内部基准电压的设定（只限于 16 引脚产品）	233
10.8	A/D 转换器特性表的阅读方法	234
10.8.1	分辨率	234
10.8.2	综合误差	234
10.8.3	量化误差	234
10.8.4	零刻度误差	235
10.8.5	满刻度误差	235
10.8.6	积分线性误差	235
10.8.7	微分线性误差	235
10.8.8	转换时间	236
10.8.9	采样时间	236
10.9	A/D 转换器的注意事项	237
10.9.1	有关 STOP 模式中的工作电流	237
10.9.2	有关 ANI0 ~ ANI6 引脚的输入电压	237
10.9.3	有关竞争	237
10.9.4	有关噪声对策	237
10.9.5	模拟输入（ANIn）引脚	238
10.9.6	有关模拟输入（ANIn）引脚的输入阻抗	238
10.9.7	有关中断请求标志（ADIF）	238
10.9.8	有关 A/D 转换开始后的初次转换结果	239
10.9.9	有关 A/D 转换结果寄存器（ADCRH、ADCRL）的读操作	239
10.9.10	有关内部等效电路	239
10.9.11	有关 A/D 转换器的运行开始	239
第 11 章	比较器	240
11.1	比较器的功能	240
11.2	比较器的结构	240
11.3	比较器的控制寄存器	242
11.3.1	外围允许寄存器 0（PER0）	242
11.3.2	比较器模式设定寄存器（COMPMDR）	243
11.3.3	比较器的滤波器控制寄存器（COMPFIR）	244
11.3.4	比较器输出控制寄存器（COMPOCR）	245
11.3.5	比较器的输入 / 输出引脚的端口功能控制寄存器	245
11.4	比较器的运行说明	246
11.4.1	比较器 0 的数字滤波器运行	247
11.4.2	比较器 0 的中断运行	247
11.4.3	比较器 0 输出	247
11.5	比较器的设定流程	248
11.5.1	允许比较器的运行	248
11.5.2	禁止比较器的运行	249
第 12 章	串行阵列单元	250
12.1	串行阵列单元的功能	251
12.1.1	3 线串行 I/O（CSI00、CSI01）	251
12.1.2	UART（UART0）	252
12.1.3	简易 I ² C（IIC00）	252
12.2	串行阵列单元的结构	253
12.2.1	移位寄存器	254
12.2.2	串行数据寄存器 0nL（SDR0nL）	255
12.3	串行阵列单元的控制寄存器	256
12.3.1	外围允许寄存器 0（PER0）	257
12.3.2	串行时钟选择寄存器 0（SPS0）	258

12.3.3	串行模式寄存器 0n (SMR0nH、SMR0nL)	259
12.3.4	串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)	261
12.3.5	串行数据寄存器 0n (SDR0nH、SDR0nL)	263
12.3.6	串行标志清除触发寄存器 0n (SIR0n)	264
12.3.7	串行状态寄存器 0n (SSR0n)	265
12.3.8	串行通道开始寄存器 0 (SS0)	267
12.3.9	串行通道停止寄存器 0 (ST0)	268
12.3.10	串行通道允许状态寄存器 0 (SE0)	269
12.3.11	串行输出允许寄存器 0 (SOE0)	270
12.3.12	串行输出寄存器 0 (SO0)	271
12.3.13	串行时钟输出寄存器 0 (CKO0)	272
12.3.14	串行输出电平寄存器 0 (SOL0)	273
12.3.15	噪声滤波器允许寄存器 0 (NFEN0)	274
12.3.16	输入切换控制寄存器 (ISC)	275
12.3.17	串行输入 / 输出引脚的端口功能控制寄存器	276
12.4	运行停止模式	277
12.4.1	以单元为单位停止运行的情况	277
12.4.2	按通道停止运行的情况	278
12.5	3 线串行 I/O (CSI00、CSI01) 通信的运行	279
12.5.1	主控发送	280
12.5.2	主控接收	289
12.5.3	主控的发送和接收	298
12.5.4	从属发送	307
12.5.5	从属接收	316
12.5.6	从属发送和接收	323
12.5.7	传送时钟频率的计算	332
12.5.8	在 3 线串行 I/O (CSI00、CSI01) 通信过程中发生错误时的处理步骤	334
12.6	UART (UART0) 通信的运行	335
12.6.1	UART 发送	336
12.6.2	UART 接收	344
12.6.3	波特率的计算	351
12.6.4	UART (UART0) 通信过程中发生错误时的处理步骤	355
12.7	简易 I ² C (IIC00) 通信的运行	356
12.7.1	地址字段发送	357
12.7.2	数据发送	363
12.7.3	数据接收	367
12.7.4	停止条件的产生	372
12.7.5	传送速率的计算	373
12.7.6	在简易 I ² C (IIC00) 通信过程中发生错误时的处理步骤	375
第 13 章	串行接口 IICA	376
13.1	串行接口 IICA 的功能	376
13.2	串行接口 IICA 的结构	379
13.3	串行接口 IICA 的控制寄存器	382
13.3.1	外围允许寄存器 0 (PER0)	382
13.3.2	IICA 控制寄存器 00 (IICCTL00)	383
13.3.3	IICA 状态寄存器 0 (IICS0)	387
13.3.4	IICA 标志寄存器 0 (IICF0)	389
13.3.5	IICA 控制寄存器 01 (IICCTL01)	391
13.3.6	IICA 低电平宽度设定寄存器 0 (IICWL0)	393
13.3.7	IICA 高电平宽度设定寄存器 0 (IICWH0)	393
13.3.8	IICA 串行输入 / 输出引脚的端口功能控制寄存器	393

13.4	I ² C 总线模式的功能	394
13.4.1	引脚结构	394
13.4.2	通过 IICWL0 寄存器和 IICWH0 寄存器设定传送时钟的方法	395
13.5	I ² C 总线的定义和控制方法	396
13.5.1	开始条件	396
13.5.2	地址	397
13.5.3	传送方向的指定	397
13.5.4	应答 (ACK)	398
13.5.5	停止条件	399
13.5.6	等待	400
13.5.7	等待的解除方法	402
13.5.8	中断请求 (INTIICA0) 的产生时序和等待控制	403
13.5.9	地址匹配的检测方法	404
13.5.10	错误的检测	404
13.5.11	扩展码	405
13.5.12	仲裁	406
13.5.13	唤醒功能	408
13.5.14	通信预约	411
13.5.15	其他注意事项	414
13.5.16	通信运行	415
13.5.17	I ² C 中断请求 (INTIICA0) 的产生时序	423
13.6	时序图	443
第 14 章	中断功能	458
14.1	中断功能的种类	458
14.2	中断源和结构	458
14.3	中断功能的控制寄存器	463
14.3.1	中断请求标志寄存器 (IF0L、IF0H、IF1L)	465
14.3.2	中断屏蔽标志寄存器 (MK0L、MK0H、MK1L)	467
14.3.3	优先级指定标志寄存器 (PR00L、PR00H、PR10L、PR10H、PR01L、PR11L)	468
14.3.4	外部中断上升沿允许寄存器 0 (EGP0) 和外部中断下降沿允许寄存器 0 (EGN0)	470
14.3.5	程序状态字 (PSW)	471
14.4	中断处理的操作	472
14.4.1	可屏蔽中断请求的接受	472
14.4.2	软件中断请求的接受	474
14.4.3	多重中断处理	474
14.4.4	中断请求的保留	478
第 15 章	键中断功能	479
15.1	键中断的功能	479
15.2	键中断的结构	479
15.3	键中断的控制寄存器	481
15.3.1	键返回控制寄存器 (KRCTL)	481
15.3.2	键返回模式寄存器 (KRM0)	482
15.3.3	键返回标志寄存器 (KRF)	483
15.3.4	键中断输入引脚的端口功能控制寄存器	483
15.4	键中断功能的运行	484
15.4.1	不使用键中断标志的情况 (KRMD=0)	484
15.4.2	使用键中断标志的情况 (KRMD=1)	485

第 16 章 待机功能	487
16.1 待机功能	487
16.2 待机功能的控制寄存器	487
16.3 待机功能的运行	488
16.3.1 HALT 模式	488
16.3.2 STOP 模式	491
第 17 章 复位功能	495
17.1 复位运行的时序	496
17.2 复位期间的运行状态	498
17.3 复位源的确认寄存器	500
17.3.1 复位控制标志寄存器 (RESF)	500
第 18 章 可选择的上电复位电路	502
18.1 可选择的上电复位电路的功能	502
18.2 可选择的上电复位电路的结构	502
18.3 可选择的上电复位电路的运行	503
18.4 可选择的上电复位电路的注意事项	504
第 19 章 选项字节	505
19.1 选项字节的功能	505
19.1.1 用户选项字节 (000C0H ~ 000C2H)	505
19.1.2 片上调试选项字节 (000C3H)	505
19.2 用户选项字节的格式	506
19.3 片上调试选项字节的格式	508
19.4 选项字节的设定	508
第 20 章 闪存	509
20.1 通过闪存编程器进行的串行编程	510
20.1.1 编程环境	511
20.1.2 通信方式	511
20.2 通过外部器件 (内置 UART) 进行的编程	512
20.2.1 编程环境	512
20.2.2 通信方式	512
20.3 电路板上的引脚处理	513
20.3.1 P40/TOOL0 引脚	513
20.3.2 RESET 引脚	513
20.3.3 端口引脚	514
20.3.4 X1 引脚和 X2 引脚 (只限于 16 引脚产品)	514
20.3.5 电源	514
20.4 串行编程方法	515
20.4.1 串行编程步骤	515
20.4.2 闪存编程模式	515
20.4.3 通信方式	516
20.4.4 通信命令	516
20.5 使用 PG-FP5 时的各命令处理时间 (参考值)	517
第 21 章 片上调试功能	518
21.1 与 E1 片上调试仿真器的连接	518
21.2 片上调试安全 ID	519
21.3 用户资源的确保	519

第 22 章 十进制校正 (BCD) 电路	520
22.1 十进制校正电路的功能	520
22.2 十进制校正电路使用的寄存器	520
22.2.1 BCD 校正结果寄存器 (BCDADJ)	520
22.3 十进制校正电路的运行	521
第 23 章 指令集的概要	523
23.1 凡例	523
23.1.1 操作数的表现形式和记述方法	523
23.1.2 操作栏的说明	524
23.1.3 标志栏的说明	525
23.1.4 PREFIX 指令	525
23.2 操作一览表	526
第 24 章 电特性	543
24.1 绝对最大额定值	544
24.2 振荡电路特性	545
24.2.1 X1 振荡电路特性	545
24.2.2 内部振荡器特性	545
24.3 DC 特性	546
24.3.1 引脚特性	546
24.3.2 电源电流特性	548
24.4 AC 特性	551
24.5 外围功能特性	553
24.5.1 串行阵列单元	553
24.5.2 串行接口 IICA	557
24.6 模拟特性	558
24.6.1 A/D 转换器特性	558
24.6.2 比较器特性	558
24.6.3 内部基准电压特性	559
24.6.4 SPOR 电路特性	559
24.6.5 电源电压上升斜率特性	559
24.7 RAM 数据保持特性	560
24.8 闪存编程特性	560
24.9 专用闪存编程器通信 (UART)	560
24.10 闪存编程模式的转移时序	561
第 25 章 封装尺寸图	562
25.1 10 引脚产品	562
25.2 16 引脚产品	563
附录 A 修订记录	564

第 1 章 概述

1.1 特点

超低功耗科技

- $V_{DD}=2.0\sim 5.5V$ 的单电源
(因为包含可选择的上电复位 (SPOR) 电路的检测电压 (V_{SPOR}), 所以必须在 $2.25\sim 5.5V$ 的电压范围内使用。)
- HALT 模式
- STOP 模式

RL78 CPU 内核

- 3 段流水线的 CISC 哈佛体系结构
- 最短指令执行时间: 能在高速 (0.05 μ s: 以高速内部振荡器时钟 20MHz 运行时) 到超低速 (1.0 μ s: 以 1MHz 运行时) 之间变换。
- 地址空间: 1M 字节
- 通用寄存器: 8 位寄存器 $\times 8$
- 内部 RAM: 128B \sim 512B

代码闪存

- 代码闪存: 1KB \sim 4KB
- 内置片上调试功能

高速内部振荡器

- 可从 20MHz/10MHz/5MHz/2.5MHz/1.25MHz 中选择
- 高精度 $\pm 2.0\%$ ($V_{DD}=2.0\sim 5.5V$ 、 $T_A=-20\sim +85^\circ C$)

工作环境温度

- $T_A=-40\sim +85^\circ C$

电源管理和复位功能

- 内置可选择的上电复位 (SPOR) 电路

串行接口

- CSI: 1/2 注个通道
- UART: 1 个通道
- 简易 I²C: 1 个通道
- I²C: 1 个通道注 1

定时器

- 8/16 位定时器: 2/4 注个通道
- 12 位间隔定时器注: 1 个通道
- 看门狗定时器: 1 个通道 (能以专用低速内部振荡器时钟运行)

注 只限于 16 引脚产品。

A/D转换器

- 8/10位分辨率A/D转换器 ($V_{DD}=2.4\sim 5.5V$)
- 模拟输入: 4/7^注个通道
- 内部基准电压 (0.815V(TYP.))^注

比较器^注

- 1个通道
- 运行模式: 比较器高速模式、比较器低速模式
- 基准电压可选择外部基准电压或者内部基准电压

输入/输出端口

- I/O端口: 8/14^注个 (N沟道漏极开路输出 [V_{DD} 耐压]: 2/4^注个)
- 能进行N沟道漏极开路、内部上拉的切换
- 内置键中断功能
- 内置时钟输出/蜂鸣器输出控制电路

其他

- 内置十进制校正 (BCD) 电路

ROM、RAM容量

闪存 ROM	RAM	10 引脚	16 引脚
4KB	512B	R5F10Y17	R5F10Y47
2KB	256B	R5F10Y16	R5F10Y46
1KB	128B	R5F10Y14	R5F10Y44

注 只限于 16 引脚产品

备注 配置的功能因产品而不同。请参照“1.6 功能概要”。

1.2 产品型号一览表

图 1-1 RL78/G10 的产品型号、存储容量和封装

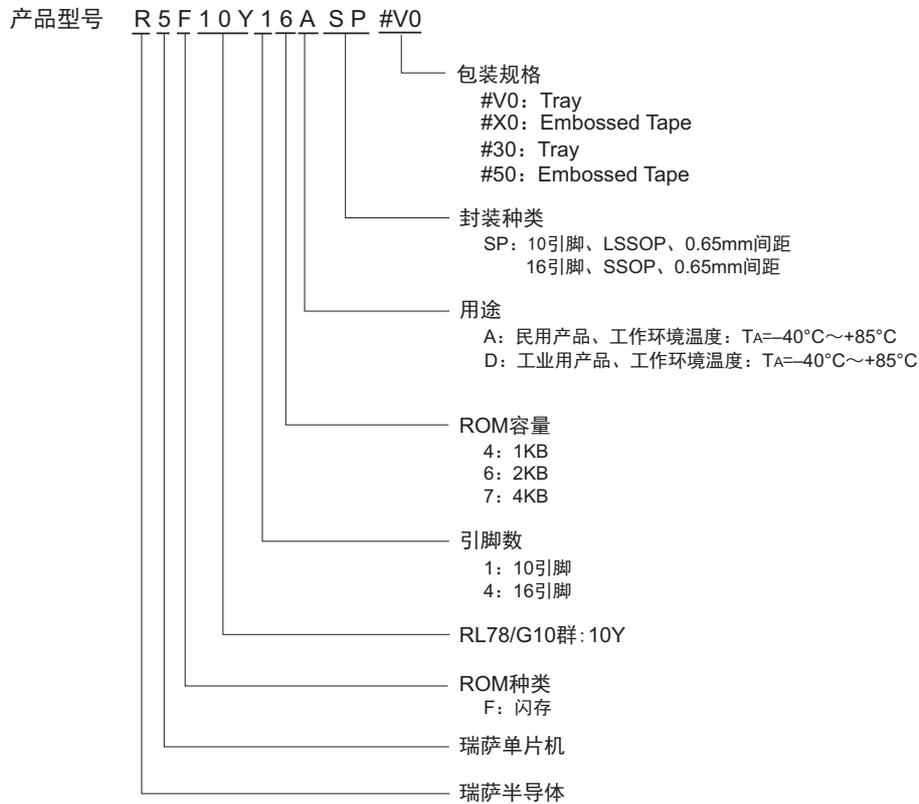


表 1-1 产品型号一览

引脚个数	封装	用途区分注 1	产品型号
10 引脚	10 引脚塑封 LSSOP (4.4×3.6mm、0.65mm 间距)	A	R5F10Y17ASP#30、R5F10Y17ASP#50 R5F10Y16ASP#V0、R5F10Y16ASP#X0 R5F10Y14ASP#V0、R5F10Y14ASP#X0
		D 注 2	R5F10Y17DSP#30、R5F10Y17DSP#50 R5F10Y16DSP#30、R5F10Y16DSP#50 R5F10Y14DSP#30、R5F10Y14DSP#50
16 引脚	16 引脚塑封 SSOP (4.4×5.0mm、0.65mm 间距)	A	R5F10Y47ASP#30、R5F10Y47ASP#50 R5F10Y46ASP#30、R5F10Y46ASP#50 R5F10Y44ASP#30、R5F10Y44ASP#50
		D 注 2	R5F10Y47DSP#30、R5F10Y47DSP#50 R5F10Y46DSP#30、R5F10Y46DSP#50 R5F10Y44DSP#30、R5F10Y44DSP#50

注 1. 有关用途区分请参照“图 1-1 RL78/G10 的产品型号、存储容量和封装”。

2. 开发中

注意 产品型号为本手册发行时的型号。最新的产品型号请参照瑞萨主页的对象产品网页。

1.3 引脚连接图（俯视图）

1.3.1 10 引脚产品

- 10 引脚塑封 LSSOP（4.4×3.6mm、0.65mm 间距）

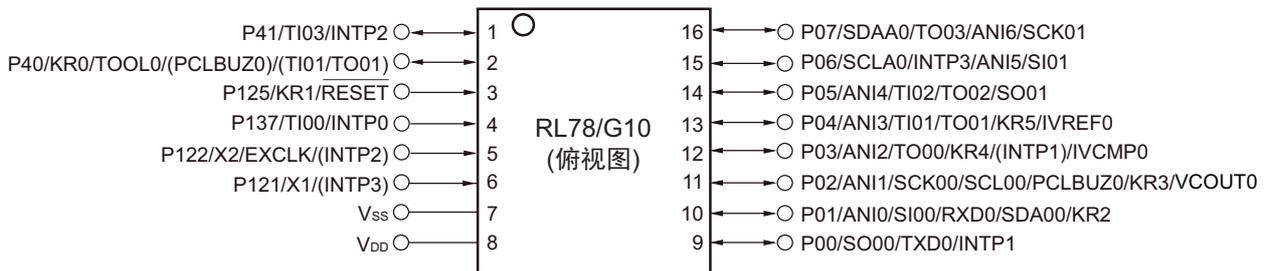


备注 1. 有关引脚名称，请参照“1.4 引脚名称”。

- 能通过设定外围 I/O 重定向寄存器（PIOR），分配上图（）内的功能。有关详细内容请参照“图 4-6 外围 I/O 重定向寄存器（PIOR）的格式”。

1.3.2 16 引脚产品

- 16 引脚塑封 SSOP（4.4×5.0mm、0.65mm 间距）



备注 1. 有关引脚名称，请参照“1.4 引脚名称”。

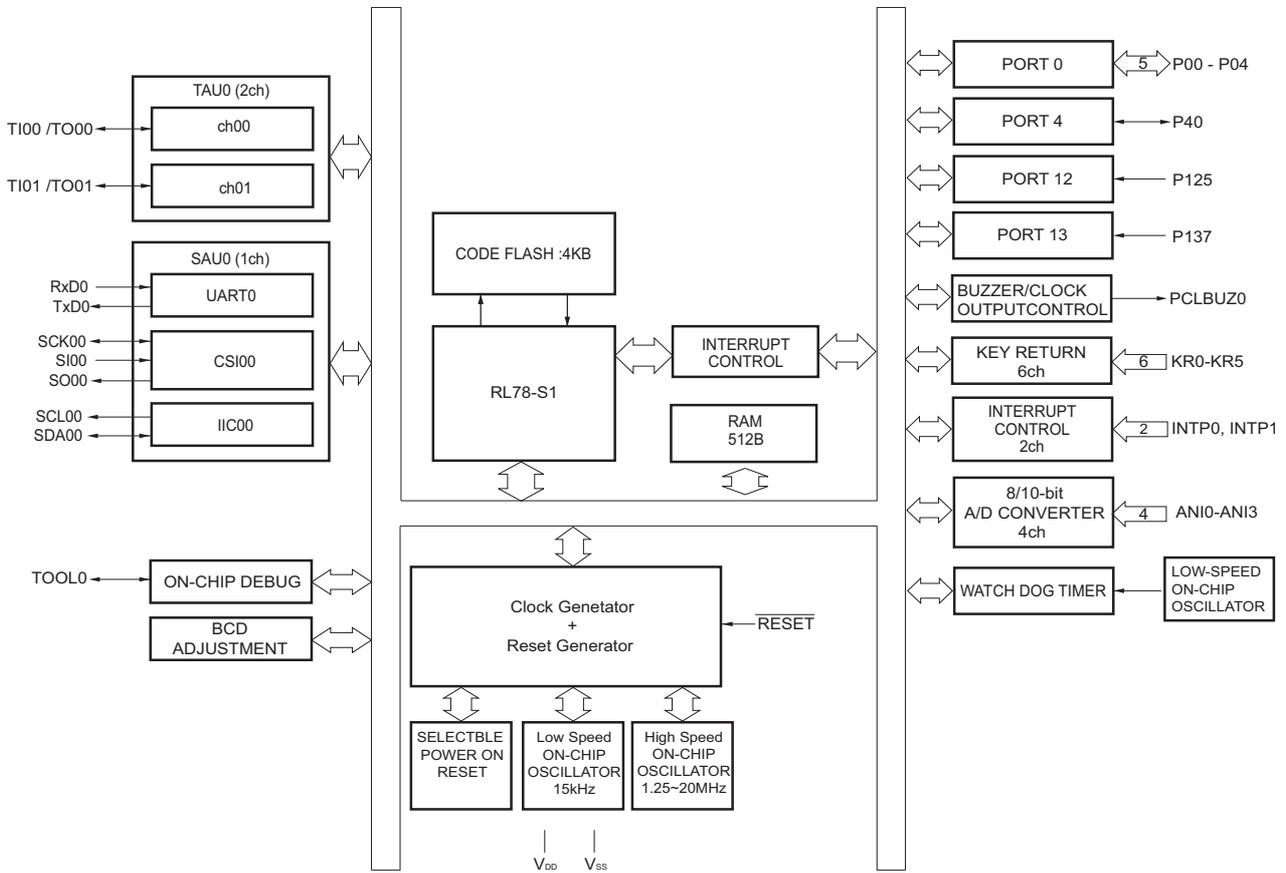
- 能通过设定外围 I/O 重定向寄存器（PIOR），分配上图（）内的功能。有关详细内容请参照“图 4-6 外围 I/O 重定向寄存器（PIOR）的格式”。

1.4 引脚名称

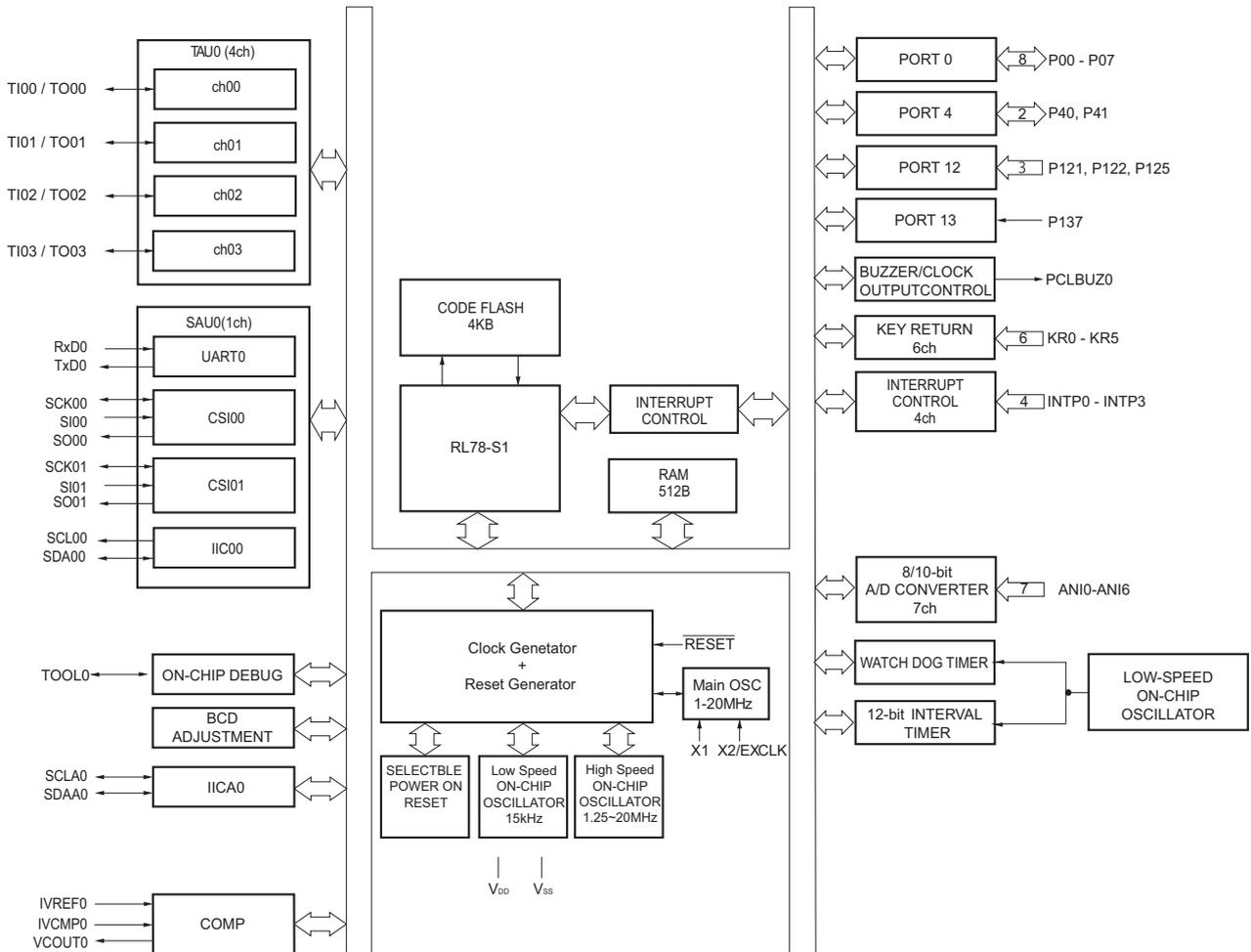
ANI0-ANI6,	Analog Input
INTP0-INTP3:	Interrupt Request From Peripheral
KR0-KR5:	Key Return
P00-P07:	Port 0
P40、P41:	Port 4
P121、P122、P125:	Port 12
P137:	Port 13
PCLBUZ0	Programmable Clock Output/Buzzer Output
EXCLK	External Clock Input
X1、X2	Crystal Oscillator (Main System Clock)
IVCMP0	Comparator Input
VCOUT0	Comparator Output
IVREF0	Comparator Reference Input
$\overline{\text{RESET}}$	Reset
RxD0	Receive Data
SCK00、SCK01	Serial Clock Input/Output
SCL00、SCLA0	Serial Clock Output
SDA00、SDAA0	Serial Data Input/Output
SI00、SI01	Serial Data Input
SO00、SO01	Serial Data Output
TI00-TI03	Timer Input
TO00-TO03	Timer Output
TOOL0	Data Input/Output for Tool
TxD0	Transmit Data
V _{DD}	Power Supply
V _{SS}	Ground

1.5 框图

1.5.1 10 引脚产品



1.5.2 16 引脚产品



1.6 功能概要

这是将外围 I/O 重定向寄存器（PIOR）置“00H”时的功能概要。

项目		10 引脚			16 引脚		
		R5F10Y14	R5F10Y16	R5F10Y17	R5F10Y44	R5F10Y46	R5F10Y47
代码闪存		1KB	2KB	4KB	1KB	2KB	4KB
RAM		128B	256B	512B	128B	256B	512B
主系统 时钟	高速系统时钟	—			X1、X2（晶体 / 陶瓷）振荡、外部主系统时钟输入（EXCLK）： 1 ~ 20MHz：V _{DD} =2.7 ~ 5.5V、 1 ~ 5MHz：V _{DD} =2.0 ~ 5.5V 注 3		
	高速内部 振荡器时钟	<ul style="list-style-type: none"> 1.25 ~ 20MHz（V_{DD}=2.7 ~ 5.5V） 1.25 ~ 5MHz（V_{DD}=2.0 ~ 5.5V 注 3） 					
低速内部振荡器时钟		15kHz(TYP.)					
通用寄存器		8 位寄存器 ×8					
最短指令执行时间		0.05μs（以 20MHz 运行时）					
指令集		<ul style="list-style-type: none"> • 数据传送（8 位） • 加减 / 逻辑运算（8 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	8			14		
	CMOS 输入 / 输出	6（N 沟道漏极开路输出 [V _{DD} 耐压]：2）			10（N 沟道漏极开路输出 [V _{DD} 耐压]：4）		
	CMOS 输入	2			4		
定时器	16 位定时器	2 个通道			4 个通道		
	看门狗定时器	1 个通道					
	12 位间隔定时器	—			1 个通道		
	定时器输出	2 个（PWM 输出：1 个）			4 个（PWM 输出：3 个注 1）		
时钟输出 / 蜂鸣器输出		1 个 2.44kHz ~ 10MHz（外围硬件时钟：f _{MAIN} =20MHz 运行时）					
比较器		—			1		
8/10 位分辨率 A/D 转换器		4 个通道			7 个通道		
串行接口		【10 引脚产品】 CSI：1 个通道 / 简易 I ² C：1 个通道 / UART：1 个通道 【16 引脚产品】 CSI：2 个通道 / 简易 I ² C：1 个通道 / UART：1 个通道					
	I ² C 总线	—			1 个通道		
向量中断 源个数	内部	8			14		
	外部	3			5		
键中断		6					
复位		<ul style="list-style-type: none"> • 通过 RESET 引脚进行的复位 • 通过看门狗定时器进行的内部复位 • 通过可选择的的上电复位进行的内部复位 • 因执行非法指令而产生的内部复位注 2 • 因数据保持下限电压而产生的内部复位 					
可选择的的上电复位电路		<ul style="list-style-type: none"> • 检测电压 上升（V_{SPOR}）：2.25V/2.68V/3.02V/4.45V(MAX.) 下降（V_{SPDR}）：2.20V/2.62V/2.96V/4.37V(MAX.) 					
片上调试功能		有					
电源电压		V _{DD} =2.0 ~ 5.5V 注 3					
工作环境温度		T _A =-40 ~ +85°C					

注 1. 输出个数因主控个数和所用通道的设定而变（参照“6.9.4 作为多重 PWM 输出功能的运行”）。

2. 在执行 FFH 的指令码时产生。

在通过片上调试仿真器进行仿真时不发生因执行非法指令而产生的复位。

3. 因为包括可选择的的上电复位（SPOR）电路的检测电压（V_{SPOR}），所以必须在 2.25 ~ 5.5V 的电压范围内使用。

第 2 章 引脚功能

2.1 端口功能

在各端口设定的输入 / 输出、缓冲以及上拉电阻对复用功能也有效。

2.1.1 10 引脚产品

功能名称	引脚类型	输入 / 输出	复位解除后	复用功能	功能
P00	7-1-2	输入 / 输出	输入端口	SO00/TXD0/INTP1	端口 0 5 位输入 / 输出端口 能以 1 位为单位指定输入或者输出。 输入端口时，能通过软件的设定使用内部上拉电阻。 P00、P01 的输出能设定为 N 沟道漏极开路输出（ V_{DD} 耐压）。 P01 ~ P04 能设定为模拟输入注。
P01	7-3-2		模拟输入	ANI0/SI00/RXD0/ SDA00/KR2	
P02	7-3-1		ANI1/SCK00/ SCL00/PCLBUZ0/ KR3		
P03			ANI2/TO00/KR4/ (INTP1)		
P04			ANI3/TI01/TO01/ KR5		
P40	7-1-1	输入 / 输出	输入端口	KR0/TOOL0/ (PCLBUZ0)/ (TI01/TO01)	端口 4 1 位输入 / 输出端口 能以 1 位为单位指定输入或者输出。 输入端口时，能通过软件的设定使用内部上拉电阻。
P125	3-1-1	输入	输入端口	KR1/RESET	端口 12 1 位输入专用端口 能通过软件的设定使用内部上拉电阻。 P125 复用外部复位的输入引脚 ($\overline{\text{RESET}}$)。 用作外部复位时，必须将选项字节 (000C1H) 的 PORTSELB 位置“1”。
P137	2-1-2	输入	输入端口	TI00/INTP0	端口 13 1 位输入专用端口

注 通过端口模式控制寄存器 0 (PMC0) 将各引脚设定为数字或者模拟。(能以 1 位单位进行设定。)

备注 能通过设定外围 I/O 重定向寄存器 (PIOR)，分配上表 () 内的功能。有关详细内容请参照“图 4-6 外围 I/O 重定向寄存器 (PIOR) 的格式”。

2.1.2 16 引脚产品

功能名称	引脚类型	输入 / 输出	复位解除后	复用功能	功能
P00	7-1-2	输入 / 输出	输入端口	SO00/TXD0/ INTP1	端口 0 8 位输入 / 输出端口 能以 1 位为单位指定输入或者输出。 输入端口时，能通过软件的设定使用内部上拉电阻。 P00、P01、P06、P07 的输出能设定为 N 沟道漏极开路输出（V _{DD} 耐压）。 P01 ~ P07 能设定为模拟输入注。
P01	7-3-2		模拟输入	ANI0/SI00/ RXD0/SDA00/ KR2	
P02	7-3-1		ANI1/SCK00/ SCL00/PCLBUZ0/ KR3/VCOUT0		
P03	7-9-1		ANI2/TO00/KR4/ (INTP1)/IVCMP0		
P04			ANI3/TI01/TO01/ KR5/IVREF0		
P05	7-3-1		ANI4/TI02/TO02/ SO01		
P06	7-3-2		ANI5/SCLA0/ INTP3/SI01		
P07			ANI6/SDAA0/ TO03/SCK01		
P40	7-1-1	输入 / 输出	输入端口	KR0/TOOL0/ (PCLBUZ0)/ (TI01/TO01)	端口 4 2 位输入 / 输出端口 能以 1 位为单位指定输入或者输出。 输入端口时，能通过软件的设定使用内部上拉电阻。
P41			TI03/INTP2	端口 12	
P121	2-2-1	输入	输入端口	X1/(INTP3)	3 位输入专用端口 P125 能通过软件的设定使用内部上拉电阻。 能以 1 位为单位指定输入或者输出。 P125 复用外部复位的输入引脚（ $\overline{\text{RESET}}$ ）。 用作外部复位时，必须将选项字节（000C1H）的 PORTSELB 位置“1”。
P122				X2/EXCLK/ (INTP2)	
P125	3-1-1	KR1/RESET			
P137	2-1-2	输入	输入端口	TI00/INTP0	端口 13 1 位输入专用端口

注 通过端口模式控制寄存器 0（PMC0）将各引脚设定为数字或者模拟。（能以 1 位单位进行设定。）

备注 能通过设定外围 I/O 重定向寄存器（PIOR），分配上表（）内的功能。有关详细内容请参照“图 4-6 外围 I/O 重定向寄存器（PIOR）的格式”。

2.2 端口以外的功能

2.2.1 各产品内置的功能

功能名称	16 引脚产品	10 引脚产品
ANI0	○	○
ANI1	○	○
ANI2	○	○
ANI3	○	○
ANI4	○	—
ANI5	○	—
ANI6	○	—
IVCMP0	○	—
IVREF0	○	—
VCOU0	○	—
INTP0	○	○
INTP1	○	○
INTP2	○	—
INTP3	○	—
KR0	○	○
KR1	○	○
KR2	○	○
KR3	○	○
KR4	○	○
KR5	○	○
PCLBUZ0	○	○
TOO0	○	○
RESET	○	○
X1	○	—
X2	○	—
EXCLK	○	—
V _{DD}	○	○
V _{SS}	○	○

功能名称	16 引脚产品	10 引脚产品
RxD0	○	○
TxD0	○	○
SCL00	○	○
SDA00	○	○
SCK00	○	○
SCK01	○	—
SI00	○	○
SI01	○	—
SO00	○	○
SO01	○	—
SCLA0	○	—
SDAA0	○	—
TI00	○	○
TO00	○	○
TI01	○	○
TO01	○	○
TI02	○	—
TO02	○	—
TI03	○	—
TO03	○	—

2.2.2 功能说明

功能名称	输入 / 输出	功能
ANI0 ~ ANI6	输入	A/D 转换器的模拟输入（参照“图 10-23 模拟输入引脚的处理”）
IVCOUT0	输出	比较器的输出
IVCMP0	输入	比较器的模拟输入
IVREF0	输入	比较器的基准电压输入
INTP0 ~ INTP3	输入	外部中断请求输入 有效边沿指定：上升沿、下降沿、上升沿和下降沿的双边沿
KR0 ~ KR5	输入	键中断输入 有效边沿指定：上升沿、下降沿
PCLBUZ0	输出	时钟输出 / 蜂鸣器输出
RESET	输入	低电平有效的系统复位输入 当不使用外部复位时，必须直接或者通过电阻连接 V_{DD} 。
RxD0	输入	串行接口 UART0 的串行数据输入
TxD0	输出	串行接口 UART0 的串行数据输出
SCK00、SCK01	输入 / 输出	串行接口 CSI00、CSI01 的串行时钟输入 / 输出
SI00、SI01	输入	串行接口 CSI00、CSI01 的串行数据输入
SO00、SO01	输出	串行接口 CSI00、CSI01 的串行数据输出
SCL00	输出	串行接口简易 I ² C（IIC00）的串行时钟输出
SDA00	输入 / 输出	串行接口简易 I ² C（IIC00）的串行数据输入 / 输出
SCLA0	输入 / 输出	串行接口 IICA0 的时钟输入 / 输出
SDAA0	输入 / 输出	串行接口 IICA0 的串行数据输入 / 输出
TI00 ~ TI03	输入	16 位定时器 00 ~ 03 的外部计数时钟 / 捕捉触发输入
TO00 ~ TO03	输出	16 位定时器 00 ~ 03 的定时器输出
X1、X2	—	连接主系统时钟的谐振器
EXCLK	输入	主系统时钟的外部时钟输入
V_{DD}	—	正电源
V_{SS}	—	接地电位
TOOL0	输入 / 输出	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 2-1 解除复位时的 P40/TOOL0 和运行模式的关系

P40/TOOL0	运行模式
V_{DD}	通常运行模式
0V	闪存编程模式

详细内容请参照“20.4.2 闪存编程模式”。

备注 作为噪声和锁定的对策，必须在 V_{DD} ~ V_{SS} 之间通过最短并且较粗的布线连接旁路电容器（0.1 μ F 左右）。

2.3 未使用引脚的处理

未使用引脚的处理如表 2-2 所示。

备注 配置的引脚因产品而不同。请参照“1.3 引脚连接图（俯视图）”和“2.1 端口功能”。

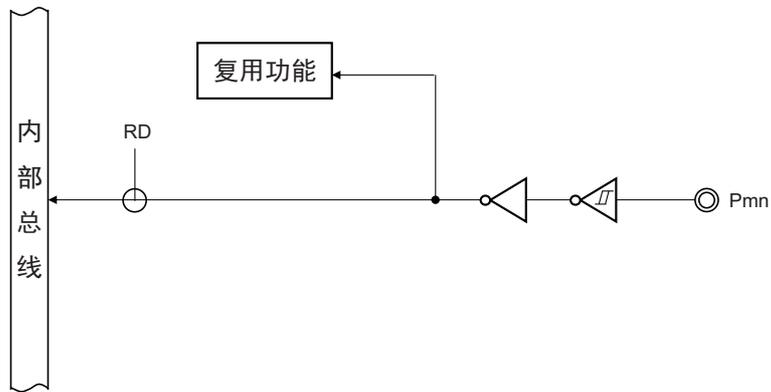
表 2-2 各未使用引脚的处理

引脚名	输入 / 输出	未使用时的推荐连接方法
P00 ~ P07	输入 / 输出	输入时：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。 输出时：必须置为开路。
P40/TOOL0		输入时：必须单独通过电阻连接 V_{DD} 。 输出时：必须置为开路。
P41		输入时：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。 输出时：必须置为开路。
P121、P122	输入	必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P125/RESET	输入	PORTSELB=1：必须置为开路或者连接 V_{DD} 。
P137	输入	必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。

2.4 引脚框图

有关在 2.1.1 10 引脚产品 ~ 2.1.2 16 引脚产品中记载的引脚类型，其引脚框图如图 2-1 ~ 图 2-8 所示。

图 2-1 引脚类型 2-1-2 的引脚框图



备注 复用功能请参照“2.1 端口功能”。

图 2-2 引脚类型 2-2-1 的引脚框图

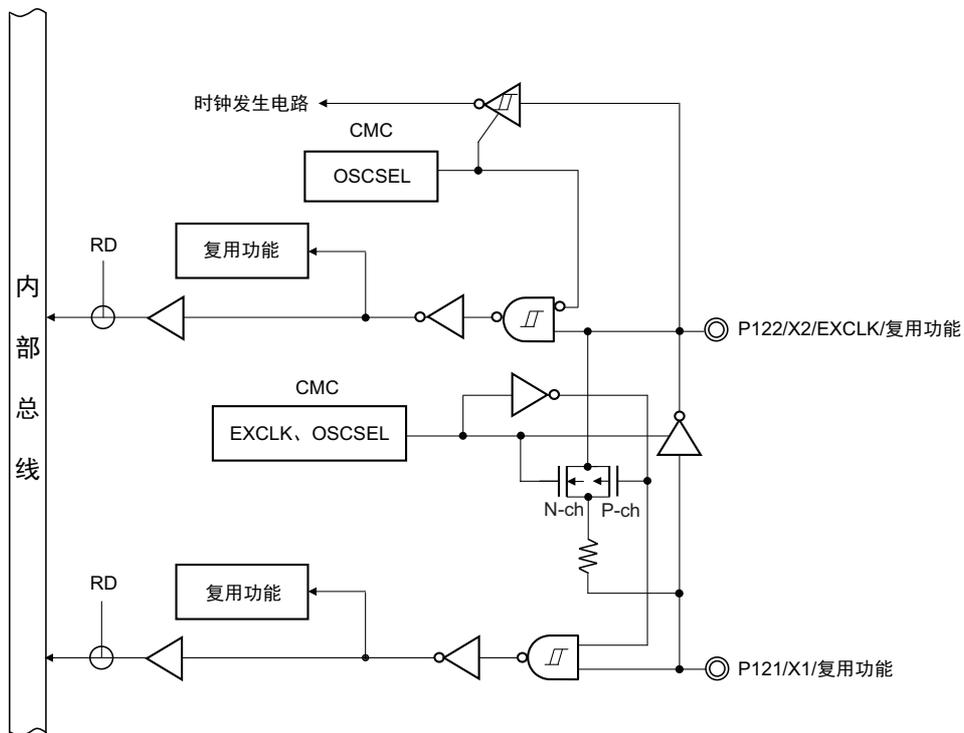
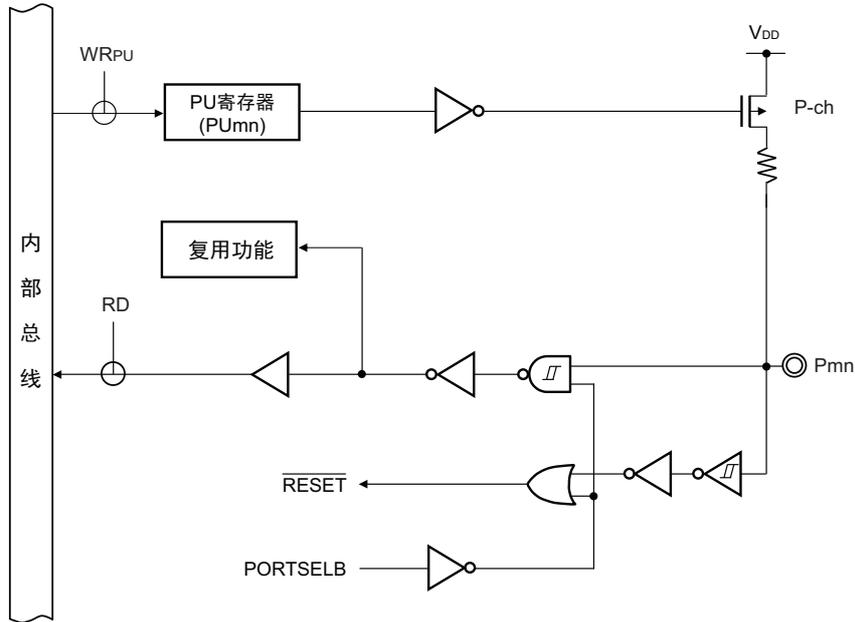
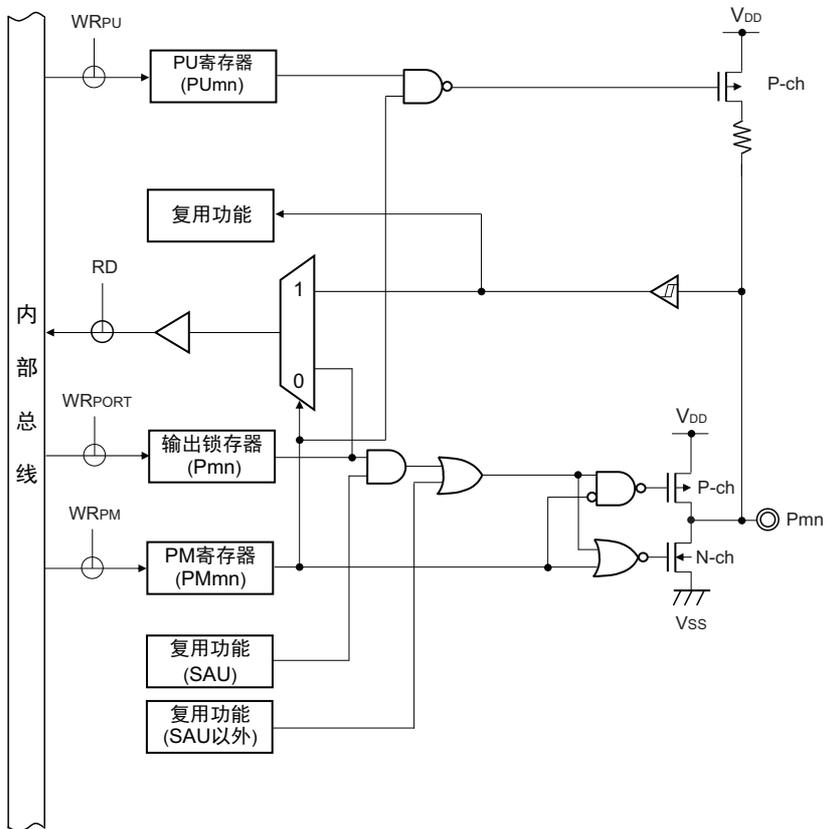


图 2-3 引脚类型 3-1-1 的引脚框图



备注 复用功能请参照“2.1 端口功能”。

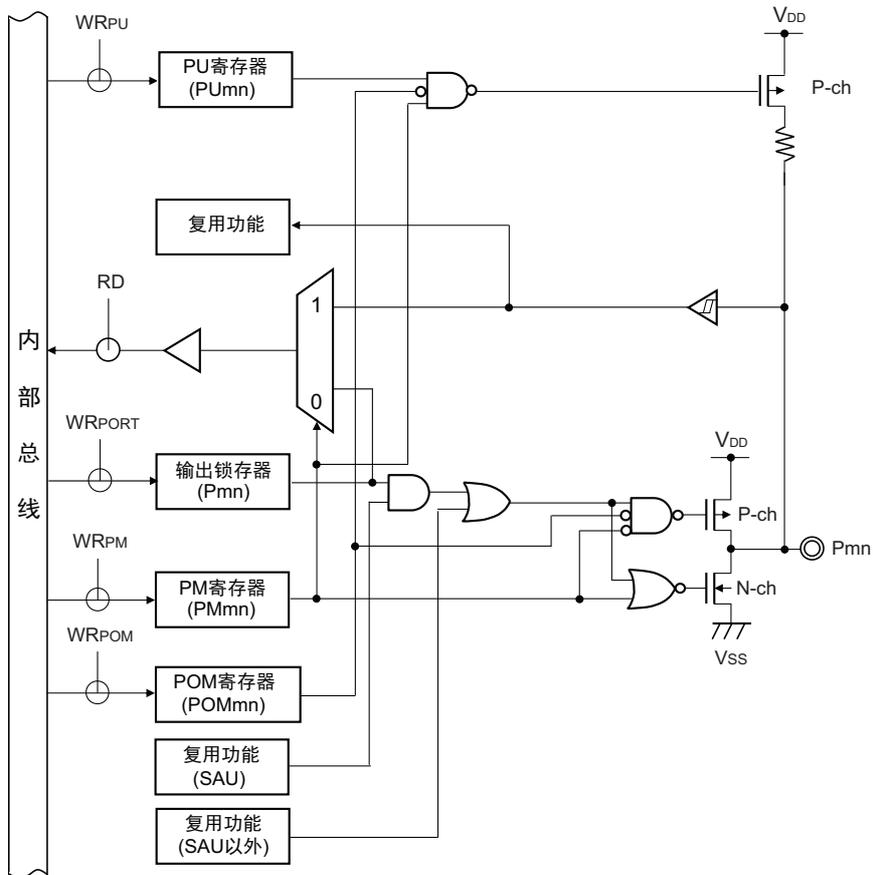
图 2-4 引脚类型 7-1-1 的引脚框图



备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

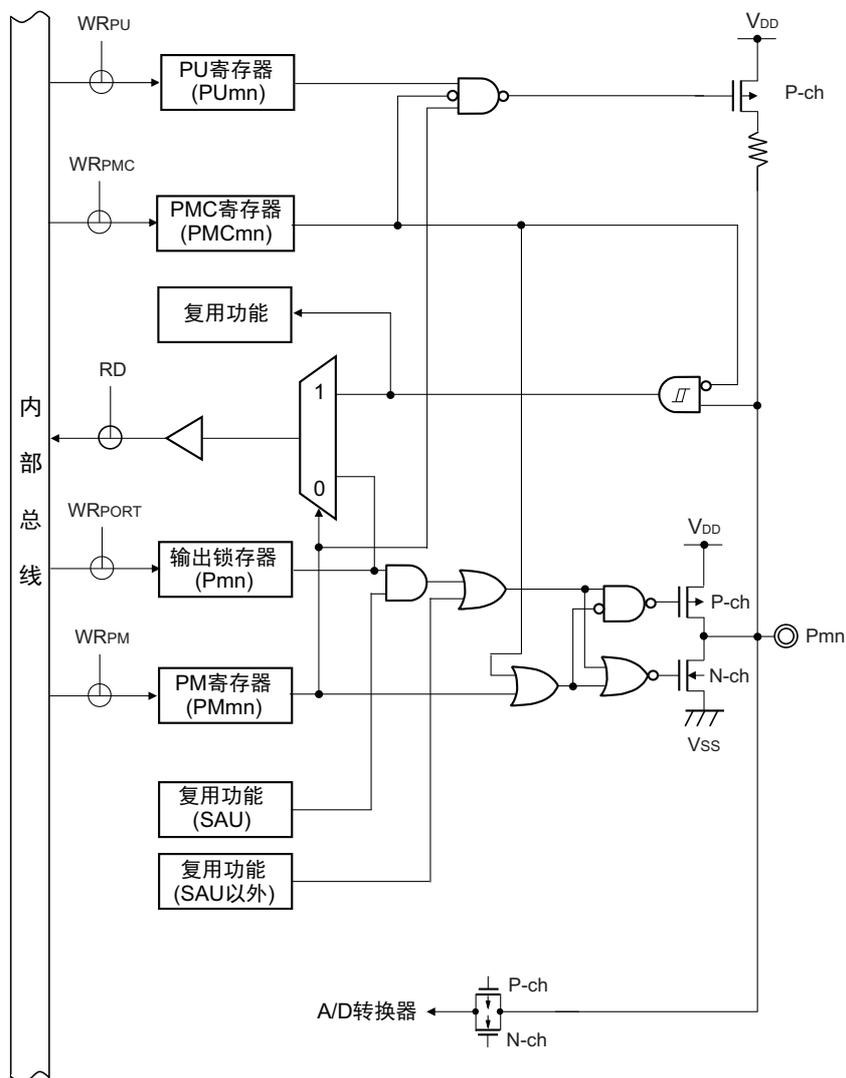
图 2-5 引脚类型 7-1-2 的引脚框图



备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

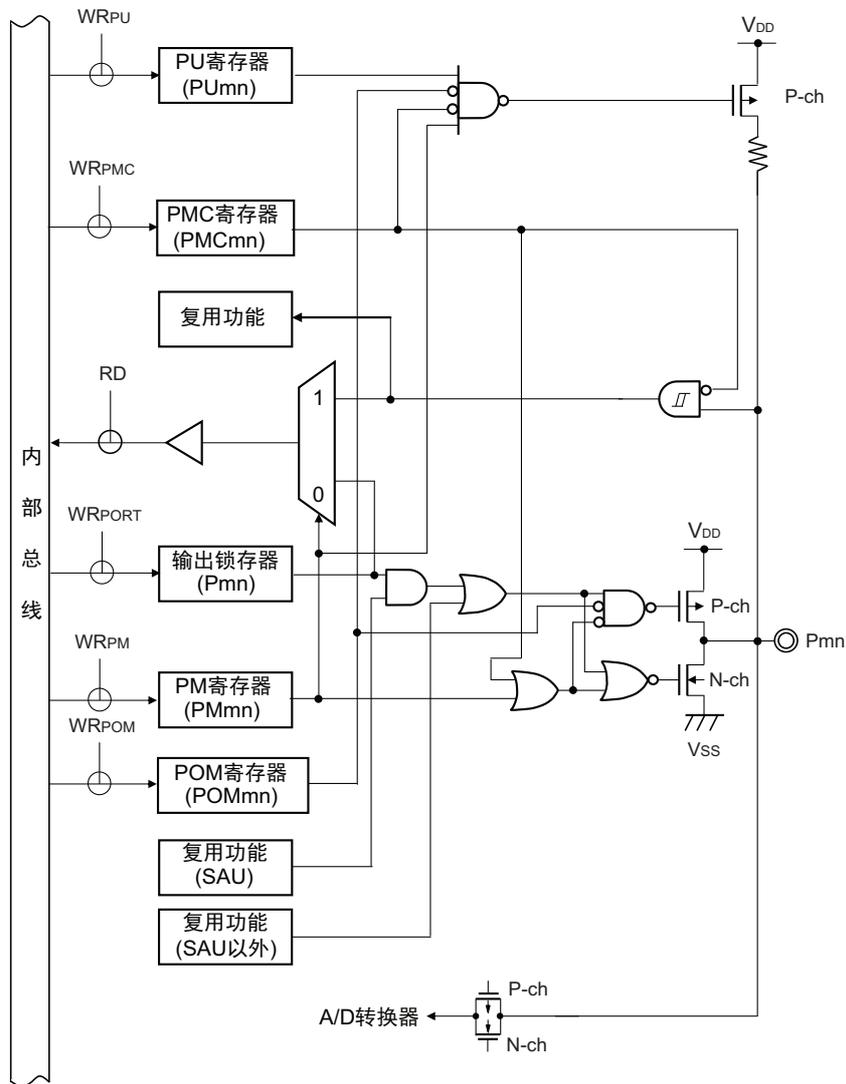
图 2-6 引脚类型 7-3-1 的引脚框图



备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

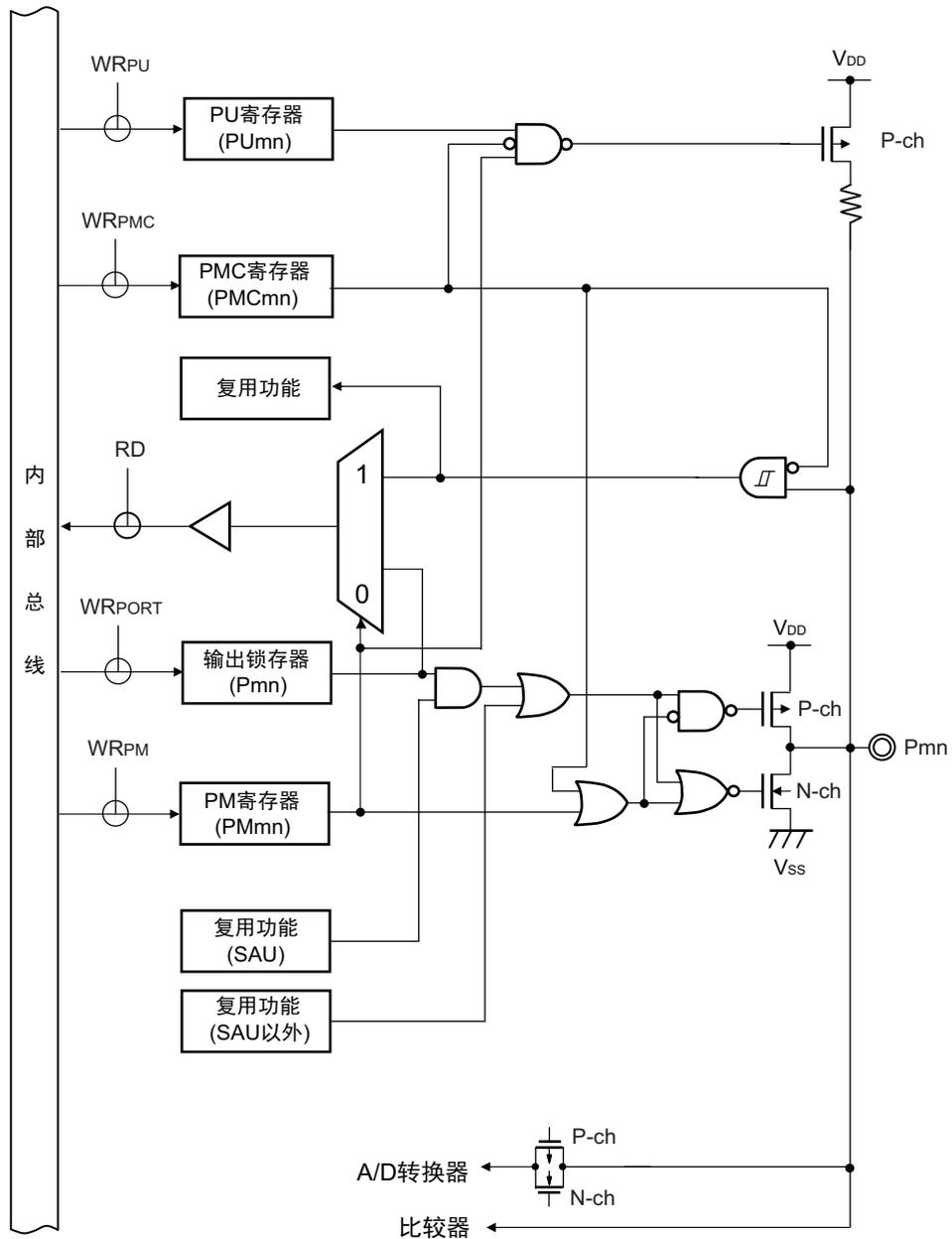
图 2-7 引脚类型 7-3-2 的引脚框图



备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

图 2-8 引脚类型 7-9-1 的引脚框图



备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

第3章 CPU 体系结构

RL78/G10 内置 RL78-S1 内核。

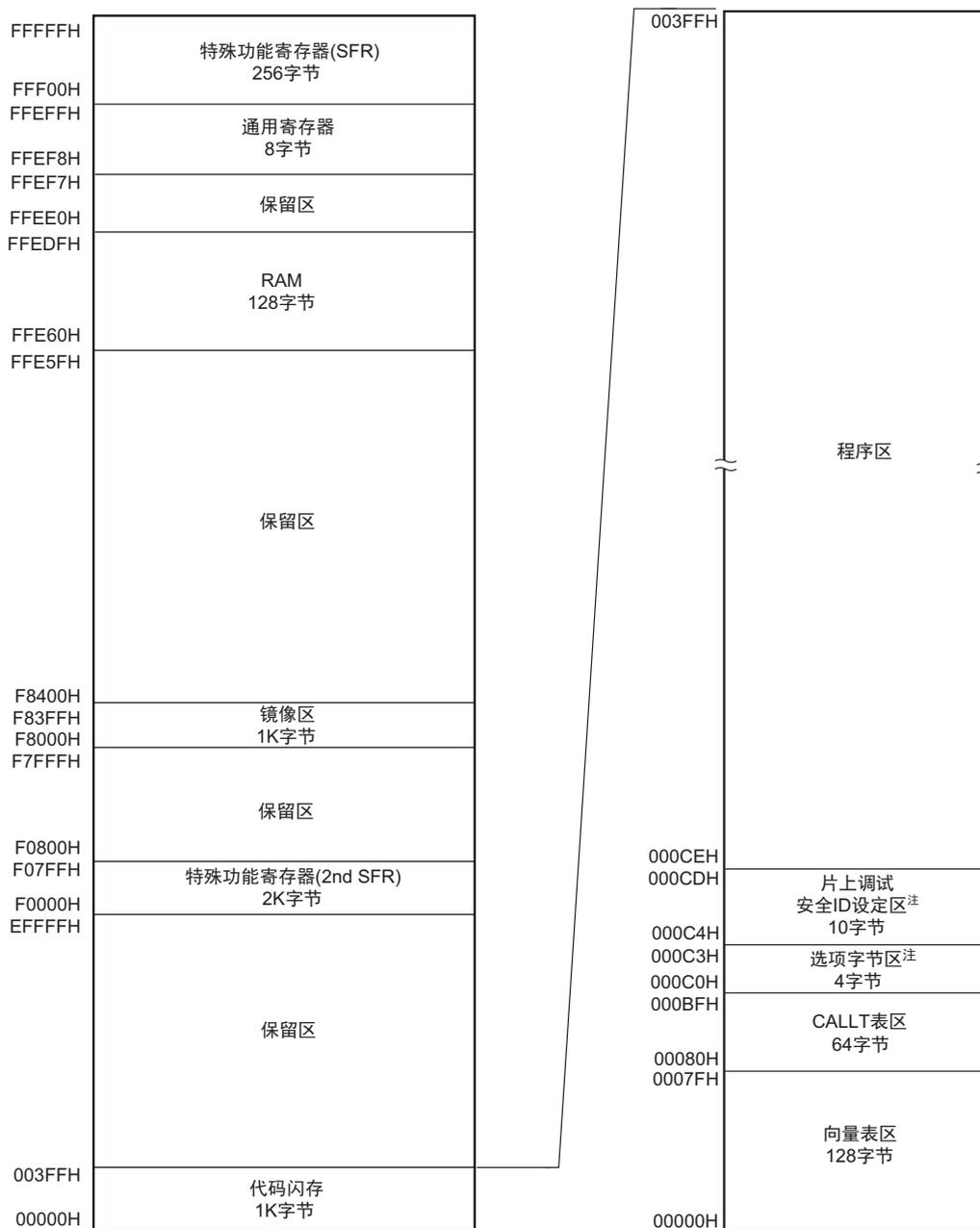
RL78-S1 内核的特征如下所示。

- 3段流水线的CISC哈佛体系结构
- 地址空间：1M字节
- 通用寄存器：8位寄存器×8
- 指令与RL78-S2内核通用。但是，以下指令的时钟数在RL78-S1内核和RL78-S2内核中不同。详细内容请参照“第23章 指令集的概要”。
 - 16位数据传送（MOVW、XCHW、ONEW、CLRW）
 - 16位运算（ADDW、SUBW、CMPW）
 - 乘法运算（MULU）
 - 16位递增/递减运算（INCW、DECW）
 - 16位移位（SHRW、SHLW、SARW）
 - 16位循环（ROLWC）
 - 调用/返回（CALL、CALLT、BRK、RET、RETI、RETB）
 - 堆栈操作（PUSH、POP、MOVW、ADDW、SUBW）

3.1 存储空间

RL78/G10 能存取 1M 字节的地址空间。存储器映像如图 3-1 ~ 图 3-3 所示。

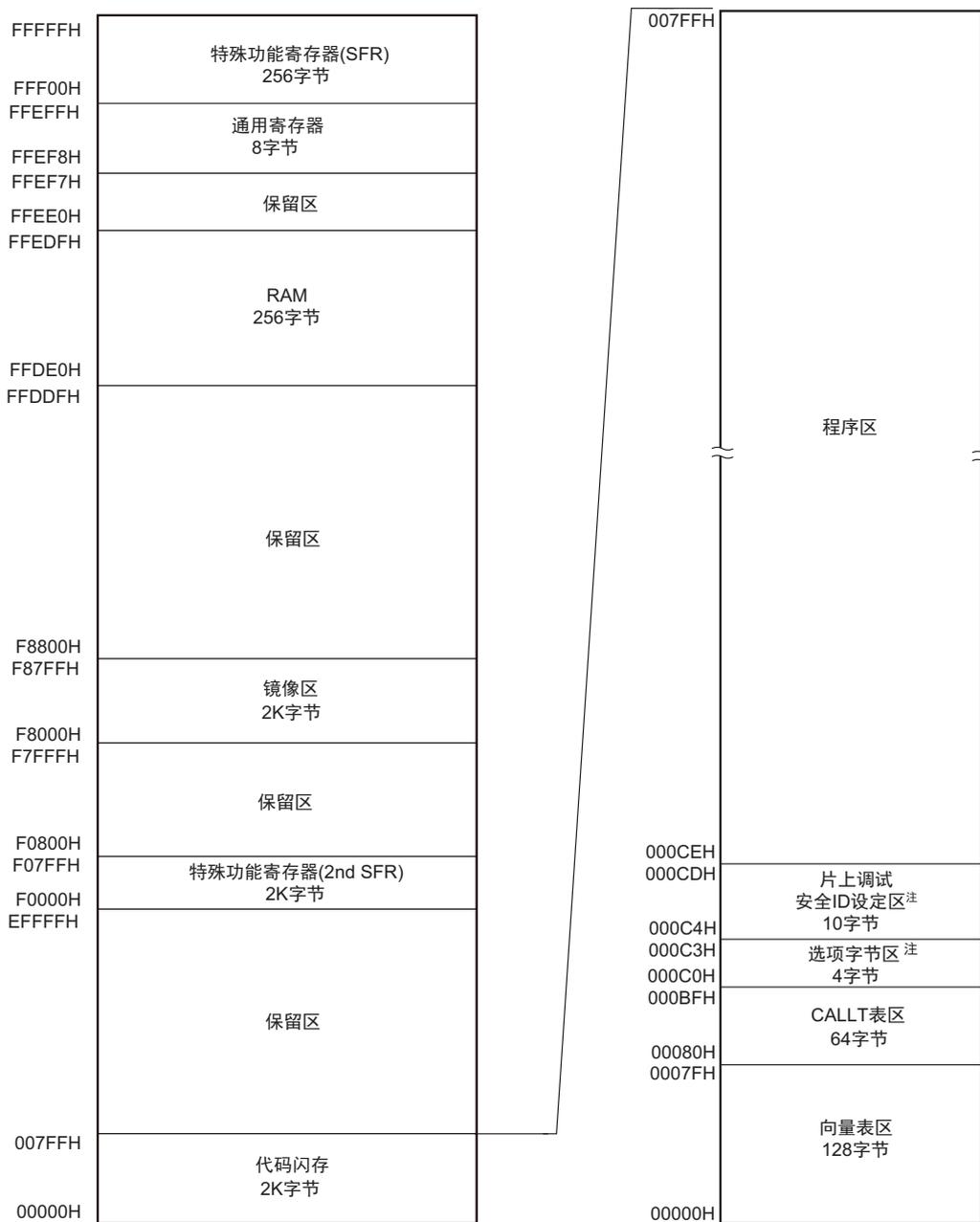
图 3-1 存储器映像 (R5F10Y14、R5F10Y44)



注 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 禁止存取保留区。

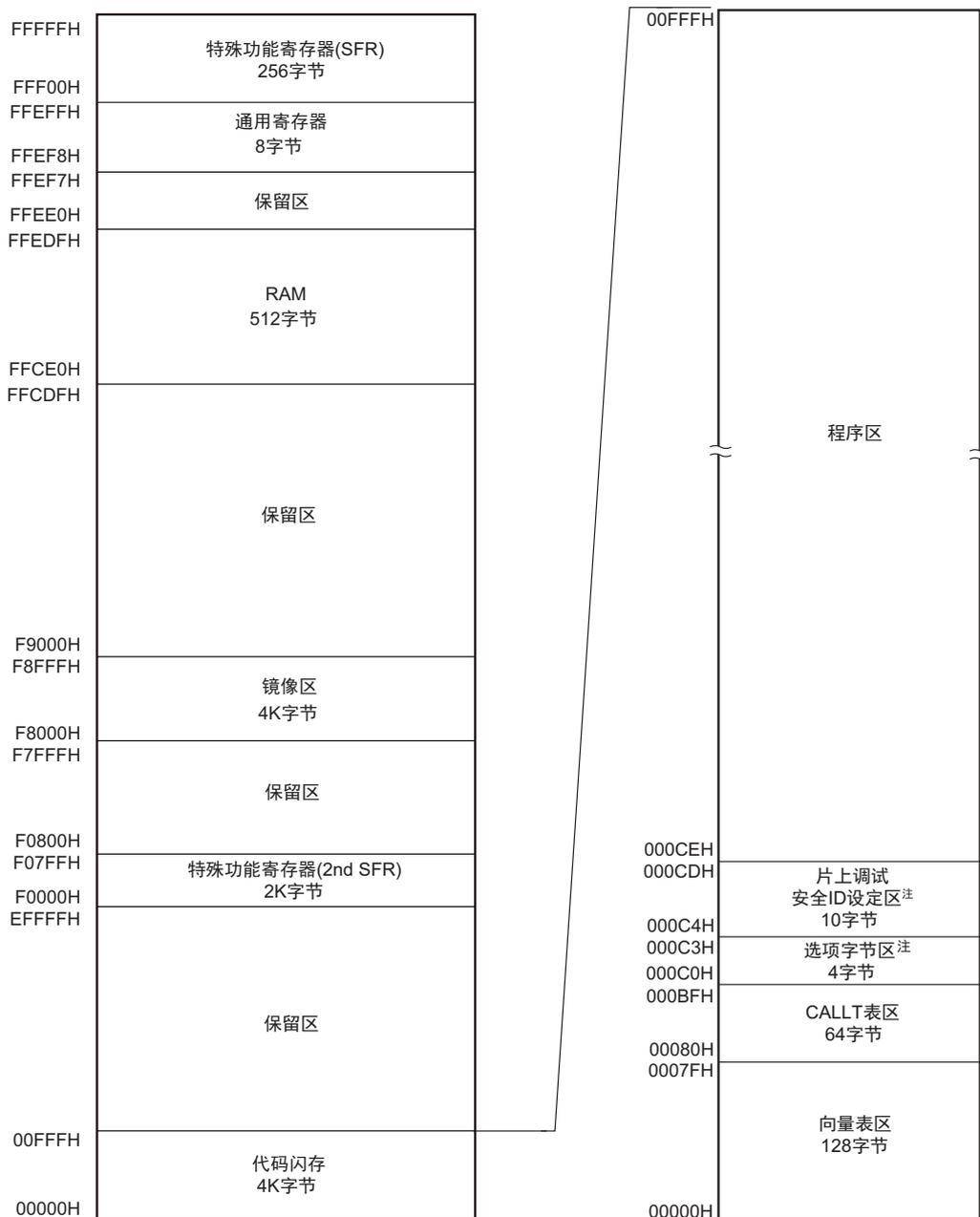
图 3-2 存储器映像 (R5F10Y16、R5F10Y46)



注 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 禁止存取保留区。

图 3-3 存储器映像 (R5F10Y17、R5F10Y47)



注 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 禁止存取保留区。

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，RL78/G10 内置的 ROM（闪存）如下所示。

表 3-1 内部 ROM 容量

产品	内部 ROM	
	构造	容量
R5F10Y14、R5F10Y44	闪存	1024×8 位 (00000H ~ 003FFH)
R5F10Y16、R5F10Y46		2048×8 位 (00000H ~ 007FFH)
R5F10Y17、R5F10Y47		4096×8 位 (00000H ~ 00FFFH)

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 3-2 向量表

向量表地址	中断源	16 引脚	10 引脚
00000H	RESET、SPOR、WDT、TRAP	○	○
00004H	INTWDTI	○	○
00006H	INTP0	○	○
00008H	INTP1	○	○
0000AH	INTST0、INTCSI00、INTIIC00	○	○
0000CH	INTSR0、INTCSI01	○	注
0000EH	INTSRE0	○	○
00010H	INTTM01H	○	○
00012H	INTTM00	○	○
00014H	INTTM01	○	○
00016H	INTAD	○	○
00018H	INTKR	○	○
0001AH	INTP2	○	—
0001CH	INTP3	○	—
0001EH	INTTM03H	○	—
00020H	INTIICA0	○	—
00022H	INTTM02	○	—
00024H	INTTM03	○	—
00026H	INTIT	○	—
00028H	INTCMP0	○	—
0007EH	BRK	○	○

注 仅 INTSR0

(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。必须给子程序入口地址设定 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 19 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 21 章 片上调试功能”。

3.1.2 镜像区

在闪存为 1/2/4K 字节的产品中，将 00000H ~ 03FFFH/007FFH/00FFFH 的代码闪存区镜像到 F8000H ~ F83FFH/F87FFH/F8FFFH。

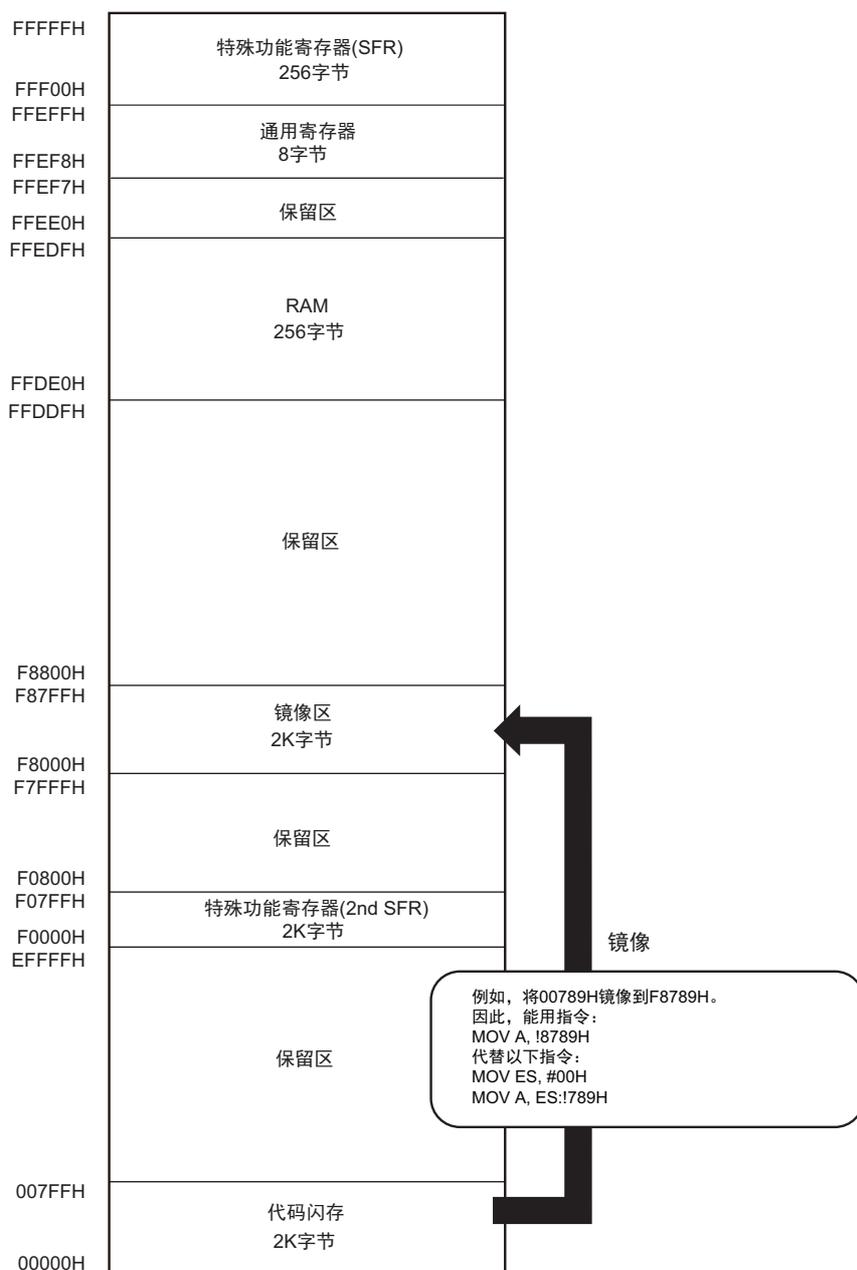
能通过从镜像目标的 F8000H ~ F83FFH/F87FFH/F8FFFH 读数据来使用操作数中不具有 ES 寄存器的指令，因此能用短代码读代码闪存的内容。

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 R5F10Y16（闪存为 2K 字节）的情况



3.1.3 内部数据存储空间

RL78/G10 内置以下 RAM。

表 3-3 内部 RAM 容量

产品	内部 RAM
R5F10Y14、R5F10Y44	128 字节 (FFE60H ~ FFEDFH)
R5F10Y16、R5F10Y46	256 字节 (FFDE0H ~ FFEDFH)
R5F10Y17、R5F10Y47	512 字节 (FFCE0H ~ FFEDFH)

内部 RAM 除了能用作数据区以外，还能作为程序区执行（不能在分配了通用寄存器的区域执行指令）。另外，堆栈存储器使用内部 RAM。

注意 不能将分配了通用寄存器的空间 (FFEF8H ~ FFEFFH) 用于取指令和堆栈区。

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 3-4）。

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 3-5）。

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

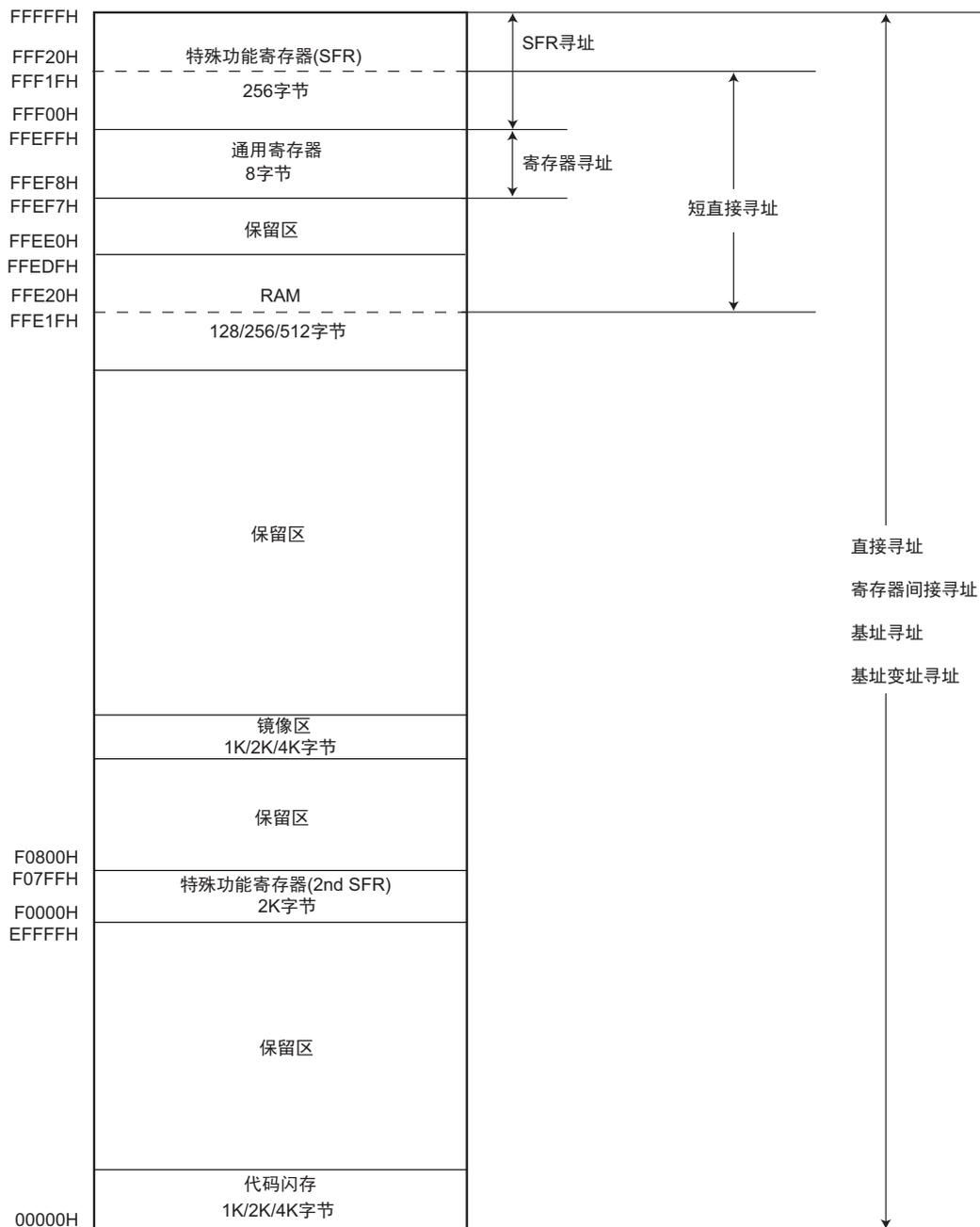
所谓寻址，是指定下一次要执行的指令地址以及指令执行操作对象的寄存器或者存储器等地址的方法。

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，RL78/G10 提供了丰富的寻址方式。尤其是在内置了数据存储器的区域中，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 3-4 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 3-4 数据存储器和寻址的对应



3.2 处理器的寄存器

RL78/G10 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 3-5 程序计数器的结构



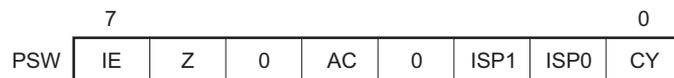
(2) 程序状态字（PSW）

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 3-6 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请求。

当 IE 位是“1”时，为中断允许（EI）状态，通过优先级控制标志（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”。

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（FFEF8H ~ FFEFFH），以 8 个 8 位寄存器（X、A、C、B、E、D、L、H）为 1 组的寄存器构成。

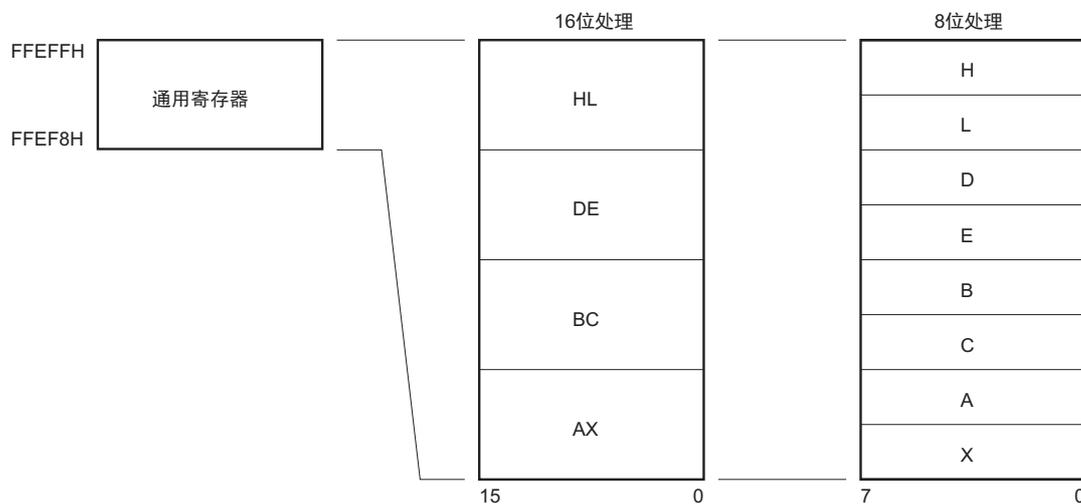
各寄存器除了能分别用作 8 位寄存器以外，还能将 2 个 8 位寄存器成对用作 1 个 16 位寄存器（AX、BC、DE、HL）。

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

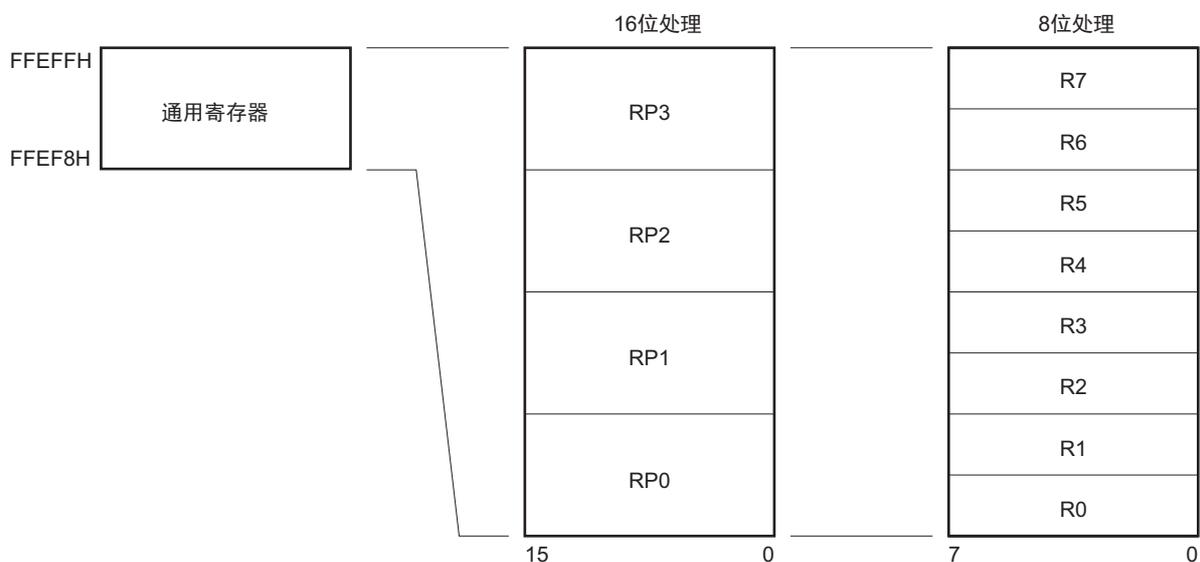
注意 禁止将通用寄存器的空间（FFEF8H ~ FFEFFH）用于取指令和堆栈区。

图 3-8 通用寄存器的结构

(a) 功能名称



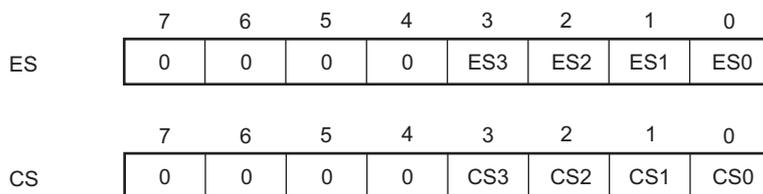
(b) 绝对名称



3.2.3 ES 寄存器和 CS 寄存器

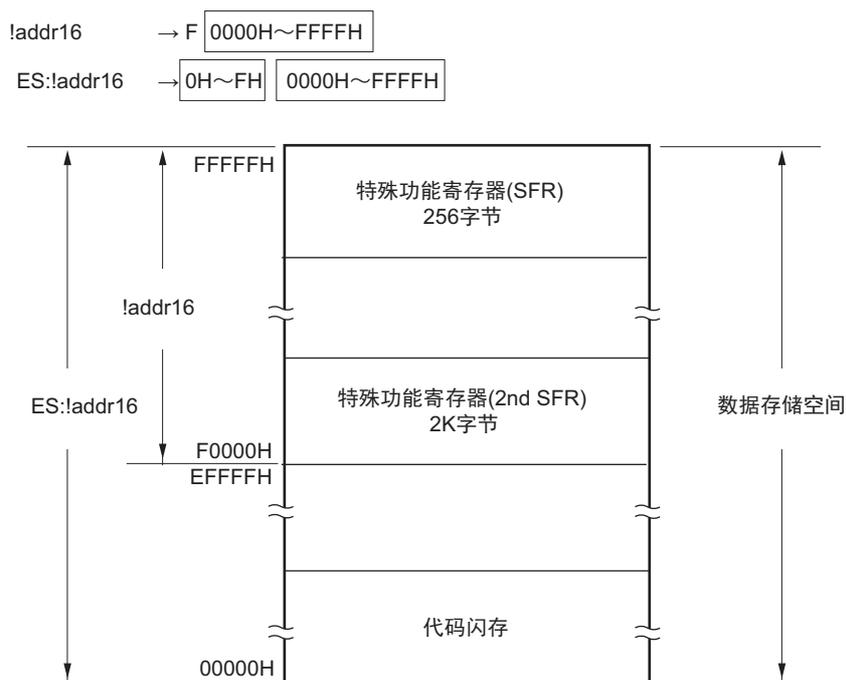
能通过 ES 寄存器指定数据存取并且通过 CS 寄存器（寄存器直接寻址）指定执行转移时的高位地址。ES 寄存器复位后的初始值为“0FH”，CS 寄存器复位后的初始值为“00H”。

图 3-9 ES/CS 寄存器的结构



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 3-10 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）因各 SFR 而不同。

各操作位单位的指定方法如下所示。

- 1位操作
给1位操作指令的操作数（sfr.bit）记述汇编程序定义的符号，也能指定地址。
- 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。

SFR 一览表如表 3-4 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）。“—”表示不能操作的位单位。
- 复位后
表示在产生复位信号后的各个寄存器的状态。

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 3-4 SFR 一览表 (1/2)

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	
FFF00H	端口寄存器 0	P0		R/W	○	○	00H
FFF04H	端口寄存器 4	P4		R/W	○	○	00H
FFF0CH	端口寄存器 12	P12		R	○	○	不定值
FFF0DH	端口寄存器 13	P13		R	○	○	不定值
FFF10H	串行数据寄存器 00L	TXD0/ SIO00	SDR00L	R/W	—	○	00H
FFF11H	串行数据寄存器 00H	—	SDR00H	R/W	—	○	00H
FFF12H	串行数据寄存器 01L	RXD0/ SIO01	SDR01L	R/W	—	○	00H
FFF13H	串行数据寄存器 01H	—	SDR01H	R/W	—	○	00H
FFF18H	定时器数据寄存器 00L	TDR00L		R/W	—	○	00H
FFF19H	定时器数据寄存器 00H	TDR00H		R/W	—	○	00H
FFF1AH	定时器数据寄存器 01L	TDR01L		R/W	—	○	00H
FFF1BH	定时器数据寄存器 01H	TDR01H		R/W	—	○	00H
FFF1EH	A/D 转换结果低位寄存器	ADCRL		R	—	○	00H
FFF1FH	A/D 转换结果高位寄存器	ADCRH		R	—	○	00H
FFF20H	端口模式寄存器 0	PM0		R/W	○	○	FFH
FFF24H	端口模式寄存器 4	PM4		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
FFF34H	键中断控制寄存器	KRCTL		R/W	○	○	00H
FFF35H	键中断标志寄存器	KRF		R/W	—	○	00H
FFF37H	键中断模式控制寄存器 0	KRM0		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 注	IICA0		R/W	—	○	00H
FFF51H	IICA 状态寄存器 0 注	IICS0		R/W	○	○	00H
FFF52H	IICA 标志寄存器 0 注	IICF0		R/W	○	○	00H
FFF60H	比较器模式设定寄存器注	COMPMDR		R/W	○	○	00H
FFF61H	比较器的滤波器控制寄存器注	COMPFIR		R/W	○	○	00H
FFF62H	比较器输出控制寄存器注	COMPOCR		R/W	○	○	00H
FFF64H	定时器数据寄存器 02L 注	TDR02L		R/W	—	○	00H
FFF65H	定时器数据寄存器 02H 注	TDR02H		R/W	—	○	00H
FFF66H	定时器数据寄存器 03L 注	TDR03L		R/W	—	○	00H
FFF67H	定时器数据寄存器 03H 注	TDR03H		R/W	—	○	00H
FFF90H	间隔定时器的控制寄存器 L 注	ITMCL		R/W	—	○	FFH
FFF91H	间隔定时器的控制寄存器 H 注	ITMCH		R/W	—	○	0FH
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	○	○	80H

注 只限于 16 引脚产品。

表 3-4 SFR 一览表 (2/2)

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	
FFFA2H	振荡稳定时间计数器的状态寄存器注 1	OSTC	R	○	○	00H
FFFA3H	振荡稳定时间选择寄存器注 1	OSTS	R/W	—	○	07H
FFFA4H	系统时钟控制寄存器注 1	CKC	R/W	○	○	00H
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	○	○	00H
FFFA8H	复位控制标志寄存器	RESF	R	—	○	不定值注 2
FFFABH	看门狗定时器允许寄存器	WDTE	R/W	—	○	1AH/9AH 注 3
FFFE0H	中断请求标志寄存器 0L	IF0L	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 注 1	IF1L	R/W	○	○	00H
FFFE4H	中断屏蔽标志寄存器 0L	MK0L	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 注 1	MK1L	R/W	○	○	FFH
FFFE8H	优先级指定标志寄存器 00L	PR00L	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 注 1	PR01L	R/W	○	○	FFH
FFFECH	优先级指定标志寄存器 10L	PR10L	R/W	○	○	FFH
FF FEDH	优先级指定标志寄存器 10H	PR10H	R/W	○	○	FFH
FF FEEH	优先级指定标志寄存器 11L 注 1	PR11L	R/W	○	○	FFH
FF FFEH	处理器模式控制寄存器	PMC	R/W	○	○	00H

注 1. 只限于 16 引脚产品。

2. 复位值因复位源不同, 如下所示。

复位源		RESET 输入	因执行非法指令发生的复位	WDT 复位	SPOR 复位	因数据保持下限电压发生的复位
RESF	TRAP	清“0”	置“1”	保持	保持	清“0”
	WDTRF		保持	置“1”	保持	
	SPORF		保持		置“1”	

3. WDTE 寄存器的复位值取决于选项字节的设定。

备注 有关扩展 SFR (2nd SFR), 请参照“表 3-5 扩展 SFR (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）因各扩展 SFR 而不同。

各操作位单位的指定方法如下所示。

- 1位操作
给1位操作指令的操作数（!addr16.bit）记述汇编程序定义的符号，也能指定地址。
- 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。

扩展 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）。“—”表示不能操作的位单位。
- 复位后
表示产生复位信号后的各个寄存器的状态。

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 3-5 扩展 SFR (2nd SFR) 一览表 (1/2)

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	
F0010H	A/D 转换器的模式寄存器 2	ADM2	R/W	○	○	00H
F0013H	A/D 测试寄存器	ADTES	R/W	—	○	00H
F0030H	上拉电阻选择寄存器 0	PU0	R/W	○	○	00H
F0034H	上拉电阻选择寄存器 4	PU4	R/W	○	○	01H
F003CH	上拉电阻选择寄存器 12	PU12	R/W	○	○	20H
F0050H	端口输出模式寄存器 0	POM0	R/W	○	○	00H
F0060H	端口模式控制寄存器 0	PMC0	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
F0077H	外围 I/O 重定向寄存器	PIOR	R/W	—	○	00H
F00A8H	高速内部振荡器的频率选择寄存器	HOCODIV	R/W	—	○	不定值注 2
F00F0H	外围允许寄存器 0	PER0	R/W	○	○	00H
F00F3H	运行速度模式控制寄存器注 1	OSMC	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	SSR00	R	—	○	00H
F0102H	串行状态寄存器 01	SSR01	R	—	○	00H
F0108H	串行标志清除触发寄存器 00	SIR00	R/W	—	○	00H
F010AH	串行标志清除触发寄存器 01	SIR01	R/W	—	○	00H
F0110H	串行模式寄存器 00L	SMR00L	R/W	—	○	20H
F0111H	串行模式寄存器 00H	SMR00H	R/W	—	○	00H
F0112H	串行模式寄存器 01L	SMR01L	R/W	—	○	20H
F0113H	串行模式寄存器 01H	SMR01H	R/W	—	○	00H
F0118H	串行通信运行设定寄存器 00L	SCR00L	R/W	—	○	87H
F0119H	串行通信运行设定寄存器 00H	SCR00H	R/W	—	○	00H
F011AH	串行通信运行设定寄存器 01L	SCR01L	R/W	—	○	87H
F011BH	串行通信运行设定寄存器 01H	SCR01H	R/W	—	○	00H
F0120H	串行通道允许状态寄存器 0	SE0	R	○	○	00H
F0122H	串行通道开始寄存器 0	SS0	R/W	○	○	00H
F0124H	串行通道停止寄存器 0	ST0	R/W	○	○	00H
F0126H	串行时钟选择寄存器 0	SPS0	R/W	—	○	00H
F0128H	串行输出寄存器 0	SO0	R/W	—	○	03H
F0129H	串行时钟输出寄存器 0	CKO0	R/W	—	○	03H
F012AH	串行输出允许寄存器 0	SOE0	R/W	○	○	00H

注 1. 只限于 16 引脚产品。

2. 通过选项字节 000C2H 的 FRQSEL2 ~ FRQSEL0 设定的值。

表 3-5 扩展 SFR (2nd SFR) 一览表 (2/2)

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	
F0134H	串行输出电平寄存器 0	SOL0	R/W	—	○	00H
F0180H	定时器计数寄存器 00L	TCR00L	R	—	○	FFH
F0181H	定时器计数寄存器 00H	TCR00H	R	—	○	FFH
F0182H	定时器计数寄存器 01L	TCR01L	R	—	○	FFH
F0183H	定时器计数寄存器 01H	TCR01H	R	—	○	FFH
F0184H	定时器计数寄存器 02L 注	TCR02L	R	—	○	FFH
F0185H	定时器计数寄存器 02H 注	TCR02H	R	—	○	FFH
F0186H	定时器计数寄存器 03L 注	TCR03L	R	—	○	FFH
F0187H	定时器计数寄存器 03H 注	TCR03H	R	—	○	FFH
F0190H	定时器模式寄存器 00L	TMR00L	R/W	—	○	00H
F0191H	定时器模式寄存器 00H	TMR00H	R/W	—	○	00H
F0192H	定时器模式寄存器 01L	TMR01L	R/W	—	○	00H
F0193H	定时器模式寄存器 01H	TMR01H	R/W	—	○	00H
F0194H	定时器模式寄存器 02L 注	TMR02L	R/W	—	○	00H
F0195H	定时器模式寄存器 02H 注	TMR02H	R/W	—	○	00H
F0196H	定时器模式寄存器 03L 注	TMR03L	R/W	—	○	00H
F0197H	定时器模式寄存器 03H 注	TMR03H	R/W	—	○	00H
F01A0H	定时器状态寄存器 00	TSR00	R	—	○	00H
F01A2H	定时器状态寄存器 01	TSR01	R	—	○	00H
F01A4H	定时器状态寄存器 02 注	TSR02	R	—	○	00H
F01A6H	定时器状态寄存器 03 注	TSR03	R	—	○	00H
F01B0H	定时器通道允许状态寄存器 0	TE0	R	○	○	00H
F01B1H	定时器通道允许状态寄存器 0 (8 位模式)	TEH0	R	○	○	00H
F01B2H	定时器通道开始寄存器 0	TS0	R/W	○	○	00H
F01B3H	定时器通道开始寄存器 0 (8 位模式)	TSH0	R/W	○	○	00H
F01B4H	定时器通道停止寄存器 0	TT0	R/W	○	○	00H
F01B5H	定时器通道停止寄存器 0 (8 位模式)	TTH0	R/W	○	○	00H
F01B6H	定时器时钟选择寄存器 0	TPS0	R/W	—	○	00H
F01B8H	定时器输出寄存器 0	TO0	R/W	—	○	00H
F01BAH	定时器输出允许寄存器 0	TOE0	R/W	○	○	00H
F01BCH	定时器输出电平寄存器 0	TOL0	R/W	—	○	00H
F01BEH	定时器输出模式寄存器 0	TOM0	R/W	—	○	00H
F0230H	IICA 控制寄存器 00 注	IICCTL00	R/W	○	○	00H
F0231H	IICA 控制寄存器 01 注	IICCTL01	R/W	○	○	00H
F0232H	IICA 低电平宽度设定寄存器 0 注	IICWL0	R/W	—	○	FFH
F0233H	IICA 高电平宽度设定寄存器 0 注	IICWH0	R/W	—	○	FFH
F0234H	从属地址寄存器 0 注	SVA0	R/W	—	○	00H

注 只限于 16 引脚产品。

备注 有关 SFR 区的 SFR，请参照“表 3-4 SFR 一览表”。

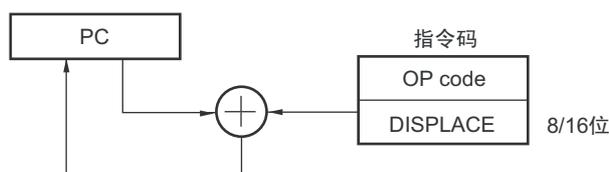
3.3 指令地址的寻址

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 3-11 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 3-12 CALL !!addr20/BR !!addr20 的例子

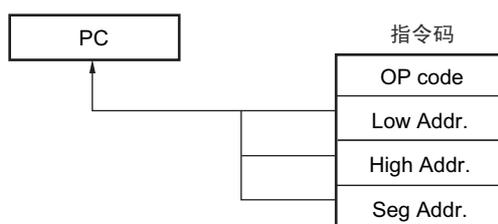
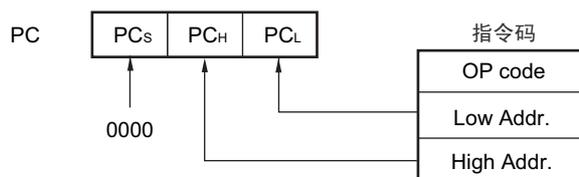


图 3-13 CALL !addr16/BR !addr16 的例子



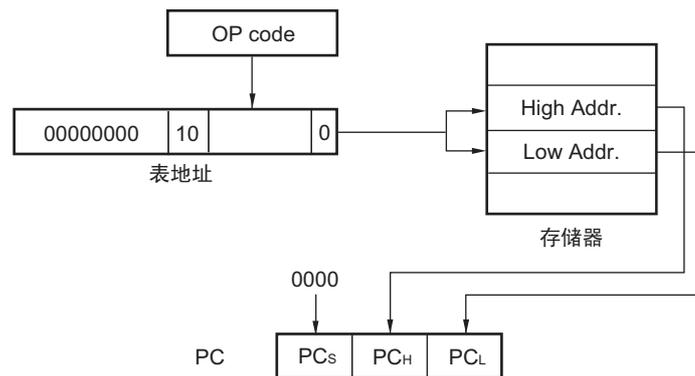
3.3.3 表间接寻址

【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 单片机只能在 00000H ~ 0FFFFH 的 64K 字节空间中进行转移。

图 3-14 表间接寻址的概要

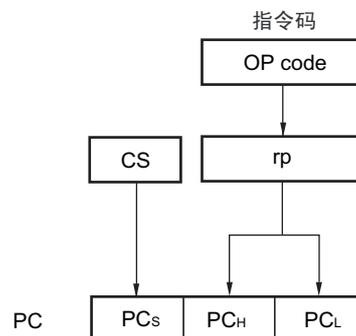


3.3.4 寄存器直接寻址

【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 3-15 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

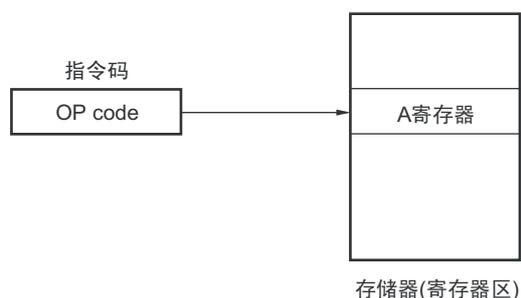
【功能】

对于具有累加器等特殊功能寄存器的存取指令，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 3-16 隐含寻址的概要



3.4.2 寄存器寻址

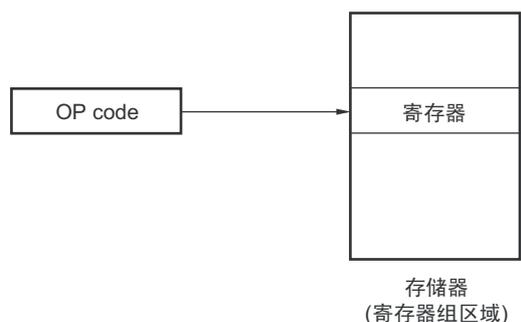
【功能】

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 3-17 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 3-18 !addr16 的例子

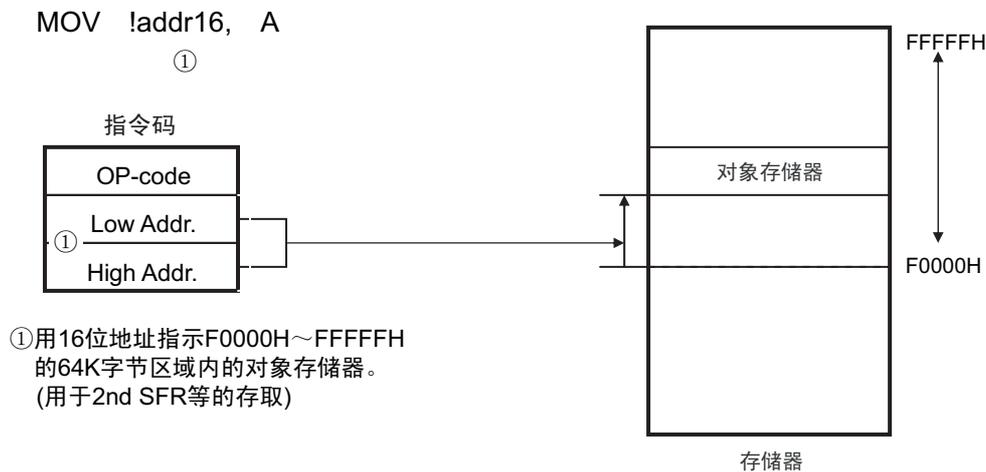
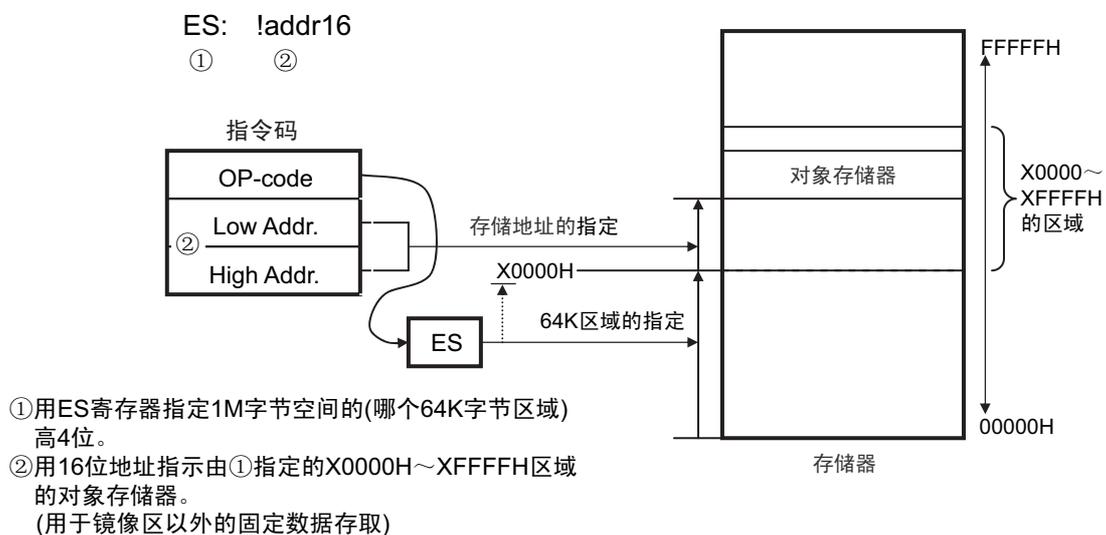


图 3-19 ES:!addr16 的例子



3.4.4 短直接寻址

【功能】

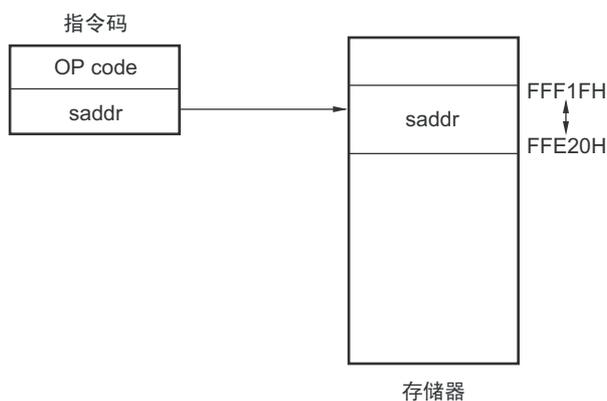
短直接寻址是通过指令码中的 8 位数据直接指定对象地址的寻址方式。此寻址方式只适用于 FFE20H ~ FFF1FH 的空间。

但是，禁止使用 FFEE0H ~ FFEF7H 的区域。RAM128B 产品中的 FFE20H ~ FFE5FH 的区域也禁止使用。

【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 3-20 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，并且能通过 20 位立即数记述 FFE20H ~ FFF1FH 的值。

但是，无论使用哪种形式，都指定存储器的 FFE20H ~ FFF1FH 空间地址。

3.4.5 SFR 寻址

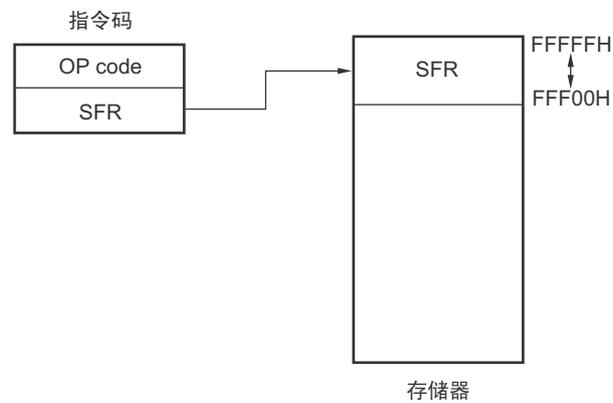
【功能】

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（只限于偶数地址）

图 3-21 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 3-22 [DE]、[HL] 的例子

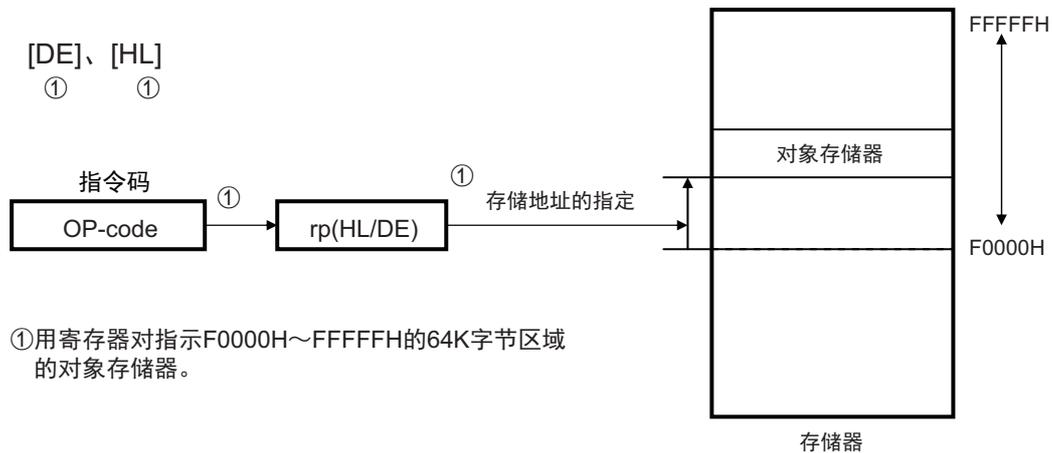
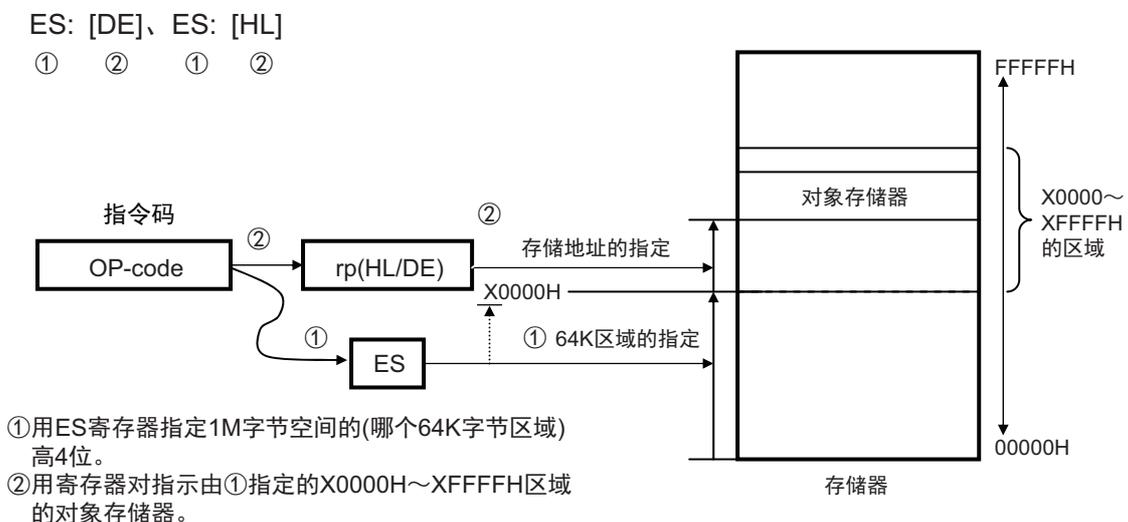


图 3-23 ES:[DE]、ES:[HL] 的例子



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 3-24 [SP+byte] 的例子

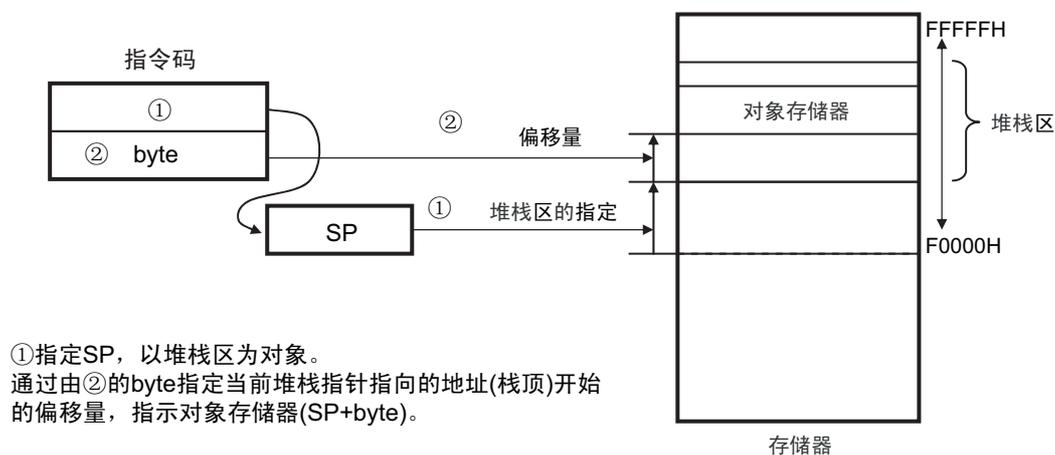


图 3-25 [HL+byte]、[DE+byte] 的例子

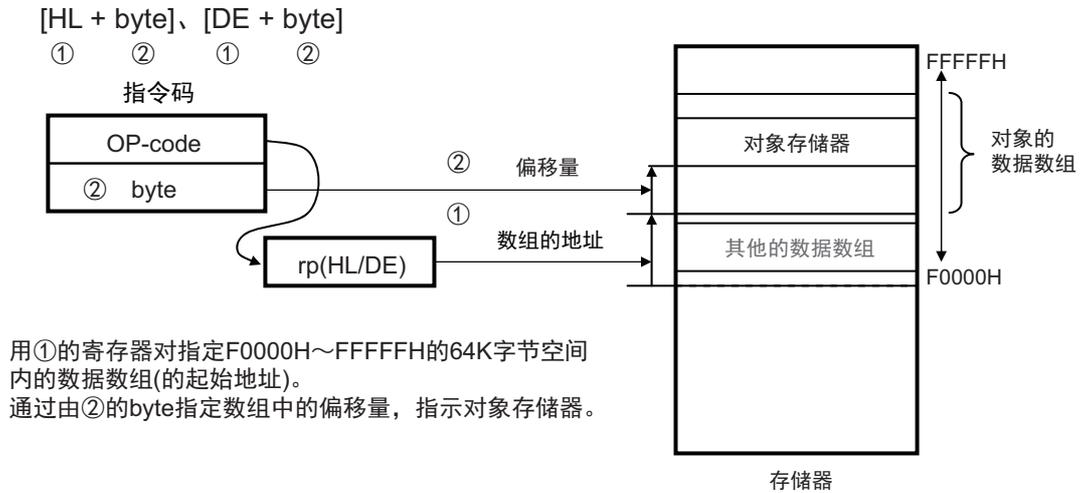


图 3-26 word[B]、word[C] 的例子

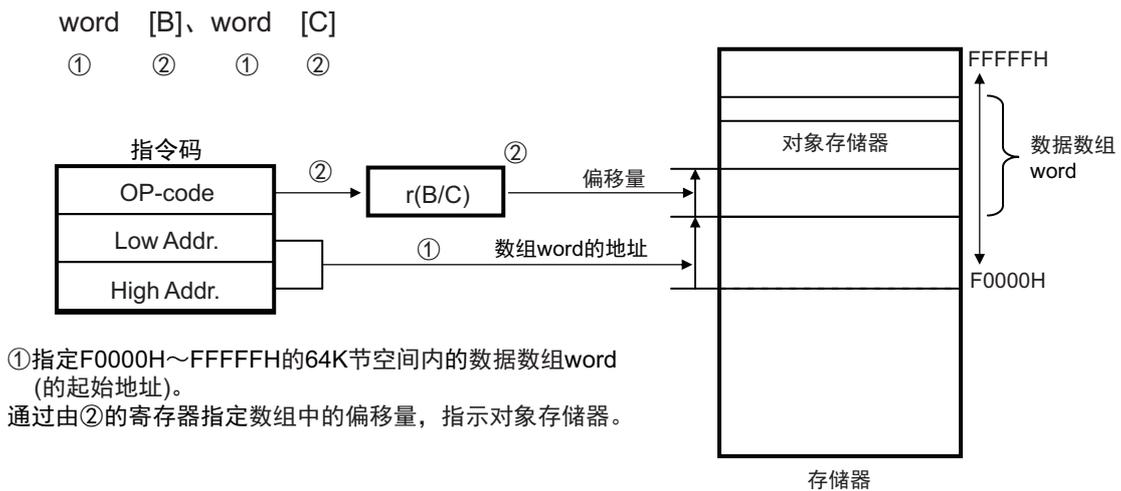


图 3-27 word[BC] 的例子

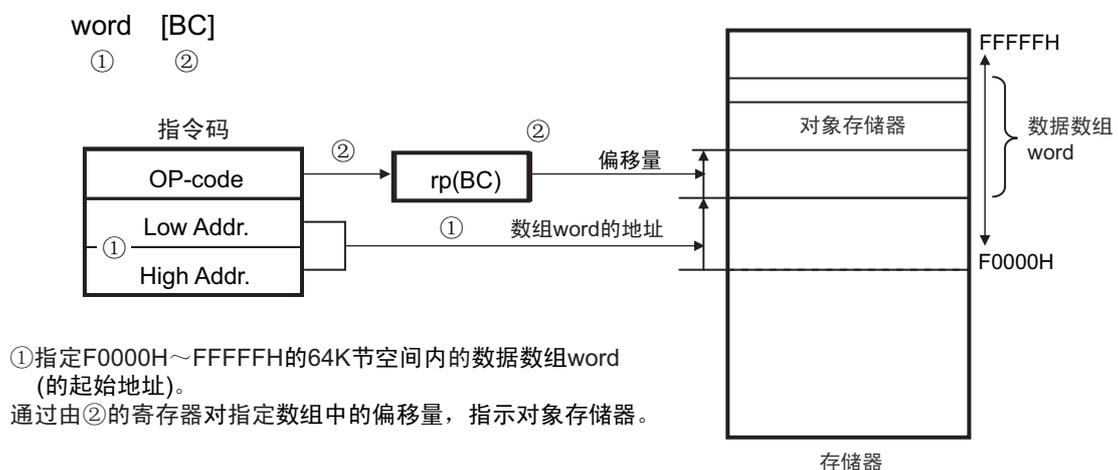


图 3-28 ES:[HL+byte]、ES:[DE+byte] 的例子

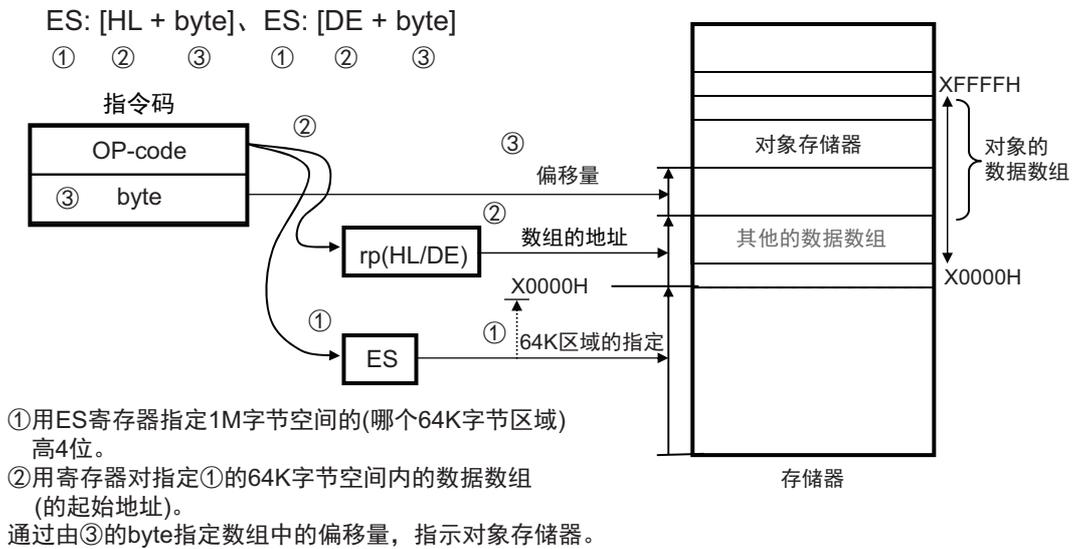


图 3-29 ES:word[B]、ES:word[C] 的例子

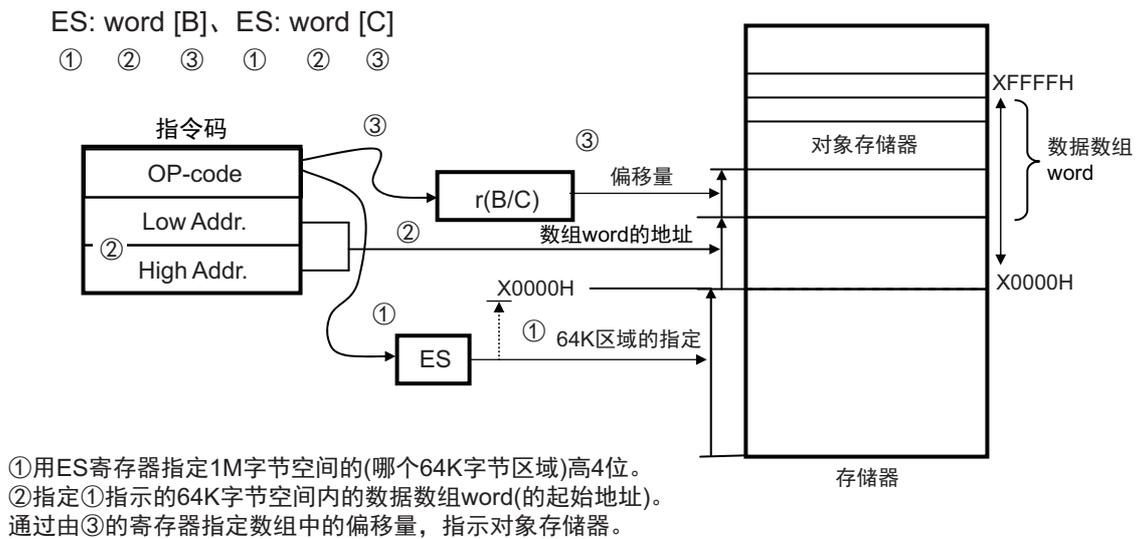
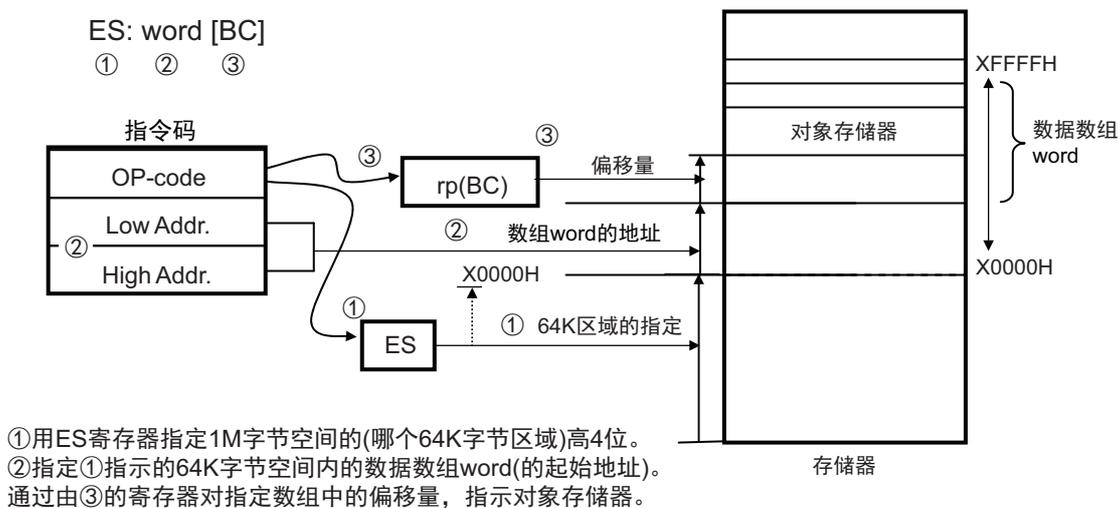


图 3-30 ES:word[BC] 的例子



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 3-31 [HL+B]、[HL+C] 的例子

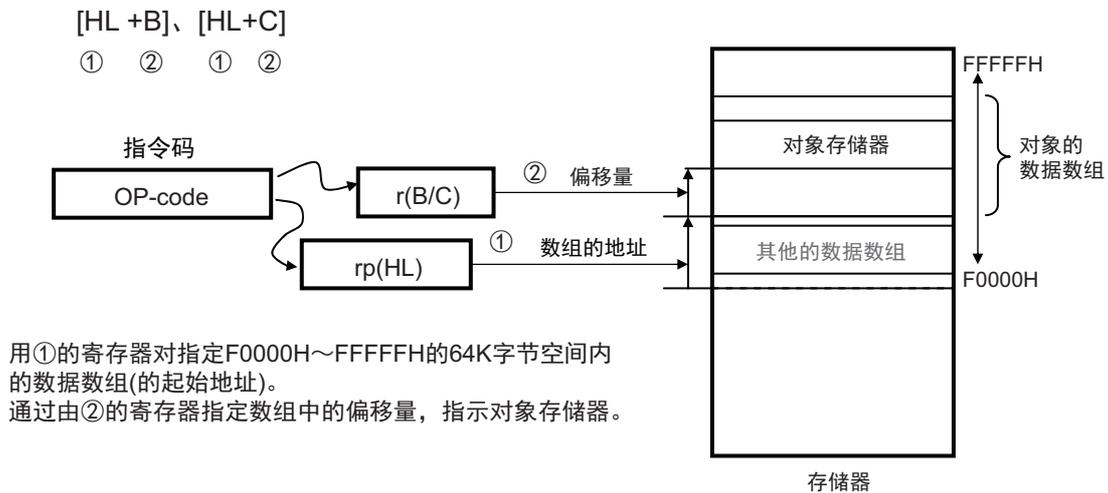
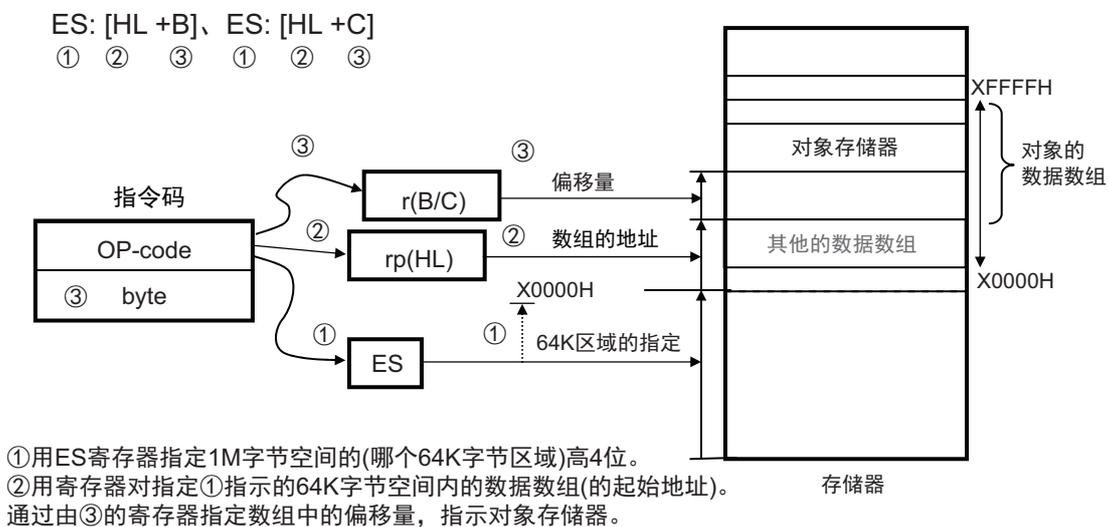


图 3-32 ES:[HL+B]、ES:[HL+C] 的例子



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 3-33 ~ 图 3-38 所示。

图 3-33 PUSH rp 的例子

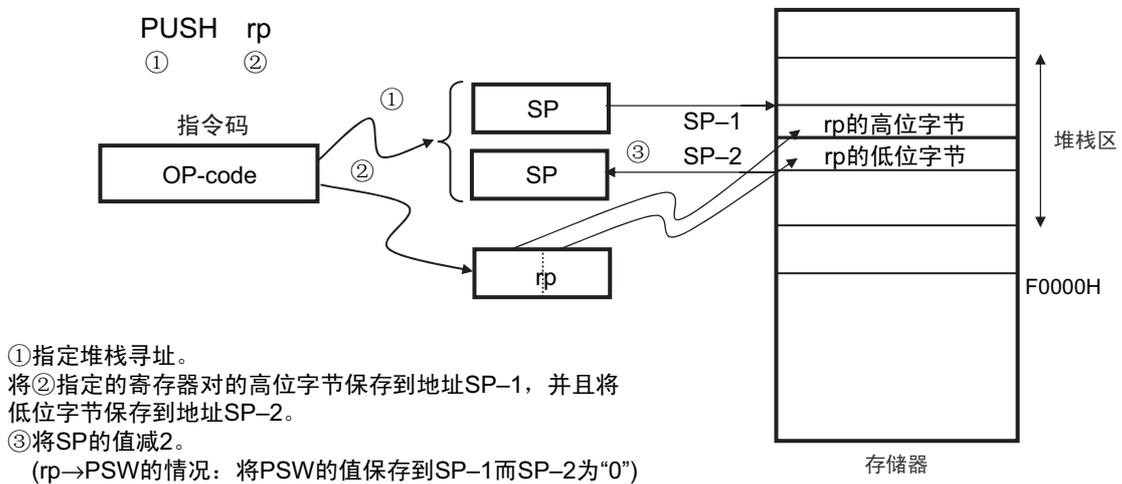


图 3-34 POP 的例子

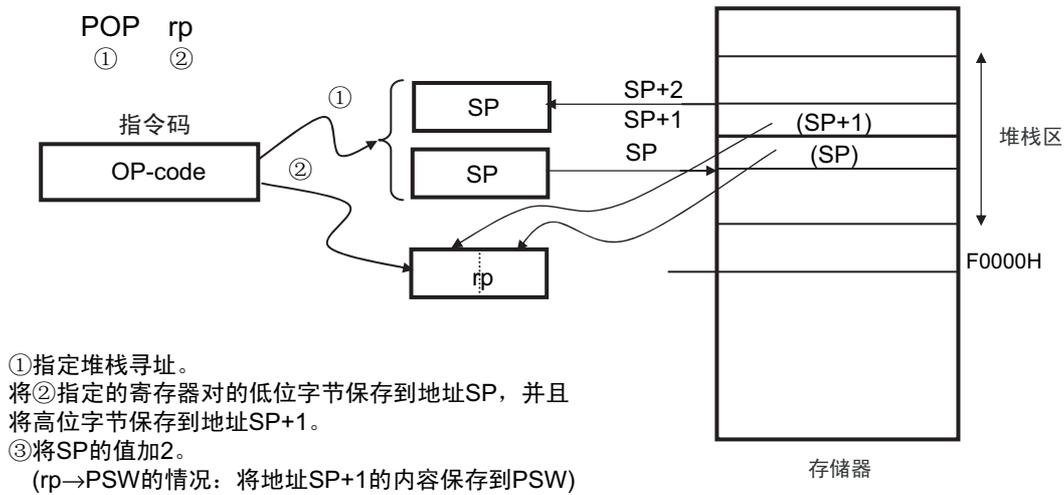


图 3-35 CALL、CALLT 的例子

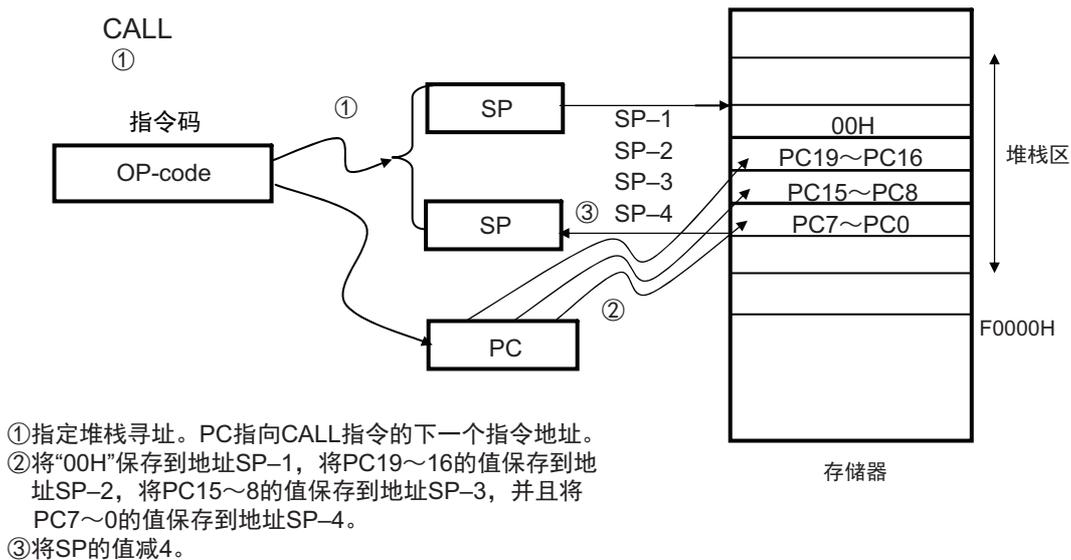


图 3-36 RET 的例子

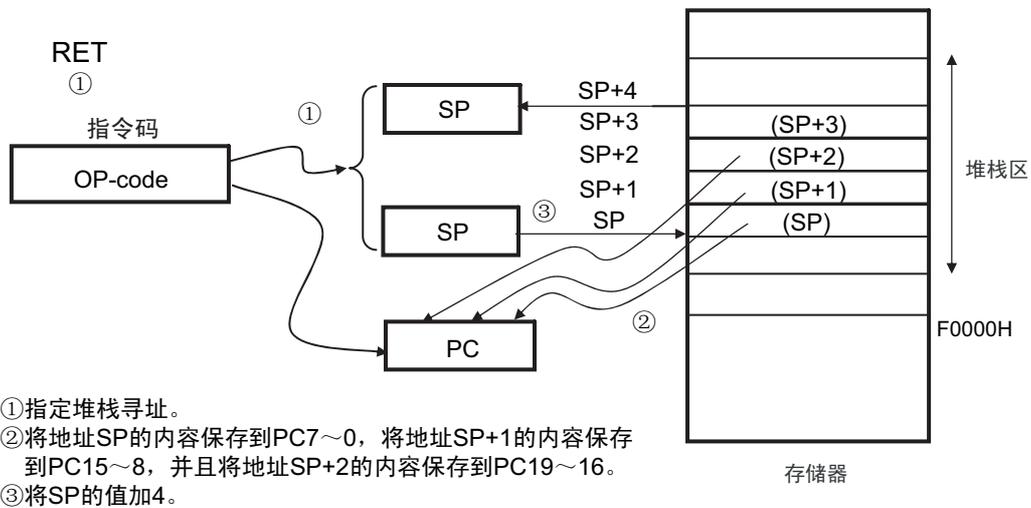


图 3-37 中断、BRK 的例子

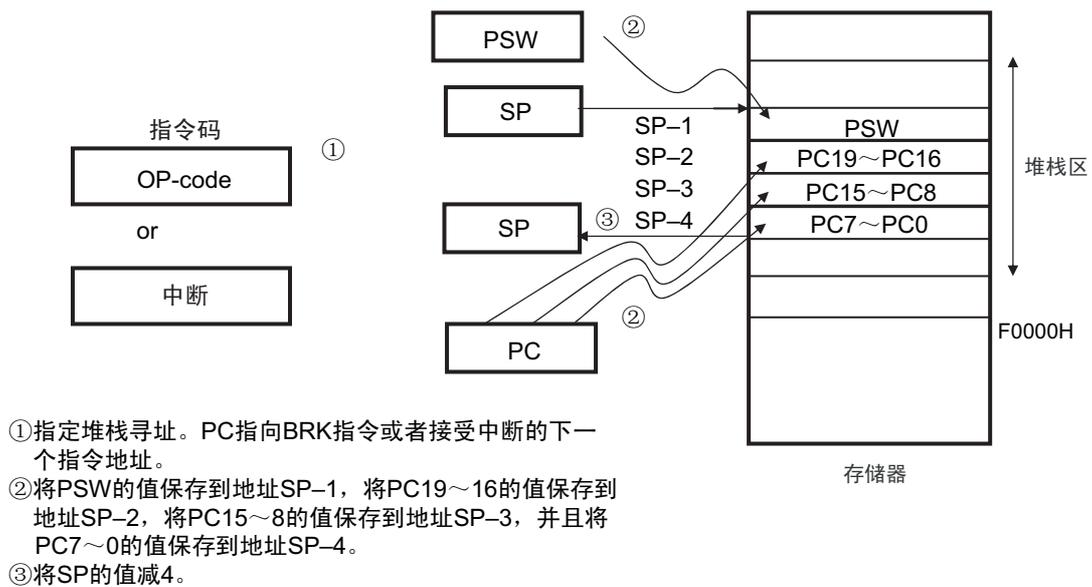
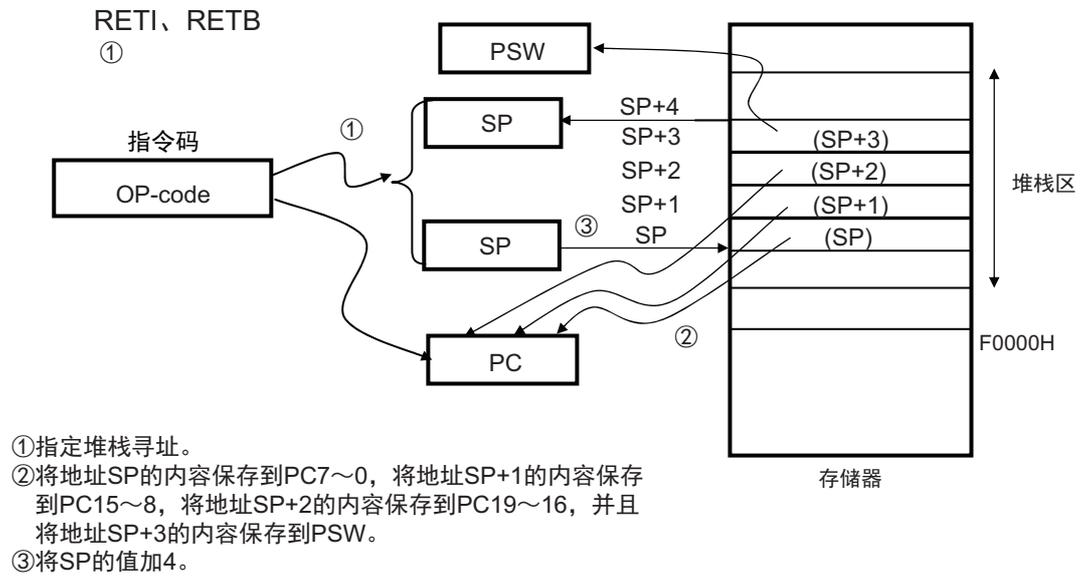


图 3-38 RETI、RETB 的例子



第 4 章 端口功能

4.1 端口功能

RL78 微控制器提供数字输入 / 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 0、4 (PM0、PM4) 端口寄存器 0、4、12、13 (P0、P4、P12、P13) 上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12) 端口输出模式寄存器 0 (POM0) 端口模式控制寄存器 0 (PMC0) 外围 I/O 重定向寄存器 (PIOR)
端口	<ul style="list-style-type: none"> • 10 引脚产品： 合计：8 个 (CMOS 输入 / 输出：6 个 (N 沟道漏极开路输出 [V_{DD} 耐压]：2 个)，CMOS 输入：2 个) • 16 引脚产品： 合计：14 个 (CMOS 输入 / 输出：10 个 (N 沟道漏极开路输出 [V_{DD} 耐压]：4 个)，CMOS 输入：4 个)
内部上拉电阻	<ul style="list-style-type: none"> • 10 引脚产品：合计 7 个 • 16 引脚产品：合计 11 个

4.2.1 端口 0

端口 0 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 0 (PM0) 以位为单位指定输入模式或者输出模式。当将 P00 ~ P07 引脚用作输入端口时，能通过上拉电阻选择寄存器 0 (PU0) 以位为单位使用内部上拉电阻。

能通过端口输出模式寄存器 0 (POM0)，以位为单位将 P01、P02、P06 和 P07 引脚的输出设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。

端口 0 作为复用功能，有串行接口的数据输入 / 输出、时钟输入 / 输出、模拟输入、键返回输入、时钟 / 蜂鸣器输出、定时器的输入 / 输出和外部中断请求输入。

在产生复位信号后，P00 变为输入模式，P01 ~ P07 变为模拟输入模式。

注 10 引脚产品时为 P00 ~ P04，16 引脚产品时为 P00 ~ P07。

4.2.2 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) 以位为单位指定输入模式或者输出模式。当将 P40、P41 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，有闪存编程器 / 调试器的数据输入 / 输出、键返回输入、时钟 / 蜂鸣器输出、定时器的输入 / 输出和外部中断请求输入。

注 10 引脚产品时为 P40、16 引脚产品时为 P40、P41。

4.2.3 端口 12

端口 12 是输入专用端口。P125 引脚能通过上拉电阻选择寄存器 12 (PU12) 使用内部上拉电阻 (选择 $\overline{\text{RESET}}$ 引脚 (PORTSELB=1) 时，内部上拉电阻总是有效。)

端口 12 作为复用功能，有键返回输入、外部中断请求输入、主系统时钟的谐振器连接、主系统时钟的外部时钟输入和复位输入。

注意 接通电源后，P125 用作 $\overline{\text{RESET}}$ 输入。通过选项字节 (000C1H) 的 PORTSELB 位，将该端口设定为 P125/KR1 或者 $\overline{\text{RESET}}$ 。如果设定为 P125/KR1，在可选择的上电复位 (SPOR) 电路引起的复位期间以及在解除 SPOR 电路引起的复位到开始正常运行的期间，不能向该引脚输入低电平。如果在此期间持续输入低电平，就保持外部复位状态。另外，在接通电源后，该引脚的内部上拉电阻变为有效。

4.2.4 端口 13

端口 13 是 1 位输入专用端口。

端口 13 作为复用功能，有定时器输入和外部中断请求输入。

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 0、4 (PM0、PM4)
- 端口寄存器 0、4、12、13 (P0、P4、P12、P13)
- 上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12)
- 端口输出模式寄存器 0 (POM0)
- 端口模式控制寄存器 0 (PMC0)
- 外围 I/O 重定向寄存器 (PIOR)

注意 配置的寄存器和位因产品而不同。有关各产品配置的寄存器和位，请参照表 4-2 和表 4-3。另外，必须给未配置的位设定初始值。

表 4-2 Pm、PMn、PUy、POM0、PMC0 寄存器及其位 (10 引脚产品)

端口		位名				
		Pm 寄存器	PMn 寄存器	PUy 寄存器	POM0 寄存器	PMC0 寄存器
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	—	PMC03
	4	P04	PM04	PU04	—	PMC04
PORT4	0	P40	PM40	PU40	—	—
PORT12	5	P125	—	PU125	—	—
PORT13	7	P137	—	—	—	—

表 4-3 Pm、PMn、PUy、POM0、PMC0 寄存器及其位 (16 引脚产品)

端口		位名				
		Pm 寄存器	PMn 寄存器	PUy 寄存器	POM0 寄存器	PMC0 寄存器
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	—	PMC03
	4	P04	PM04	PU04	—	PMC04
	5	P05	PM05	PU05	—	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT4	0	P40	PM40	PU40	—	—
	1	P41	PM41	PU41	—	—
PORT12	1	P121	—	—	—	—
	2	P122	—	—	—	—
	5	P125	—	PU125	—	—
PORT13	7	P137	—	—	—	—

备注 m=0、4、12、13, n=0、4, y=0、4、12

各寄存器的格式说明如下所示。

4.3.1 端口模式寄存器 0、4 (PM0、PM4)

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 使用复用功能时的寄存器设定”进行设定。

图 4-1 端口模式寄存器 0、4 (PM0、PM4) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W

PMmn	Pmn 引脚的输入 / 输出模式的选择
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

m=0、4, n=0 ~ 7

注意 必须给未配置的位设定初始值。

4.3.2 端口寄存器 0、4、12、13 (P0、P4、P12、P13)

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，P12、P13 的值不变，其他寄存器的值变为“00H”。

注 如果读已设定为模拟输入 (PMC0x=1、PM0x=1) 的端口，读取值就总是为“0”，与引脚输入电平无关。

在将 P125/KR1/RESET 引脚设定为 RESET 输入 (PORTSELB=1) 时，如果读 P125，读取值就总是为“1”。

图 4-2 端口寄存器 0、4、12、13 (P0、P4、P12、P13) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	P04	P03	P02	P01	P00	FFF00H	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
P12	0	0	P125	0	0	0	0	0	FFF0CH	不定值	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	不定值	R

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (输出锁存器)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (输出锁存器)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	不定值	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	不定值	R

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。

m=0、4、12、13, n=0~7

注意 必须给未配置的位设定初始值。

4.3.3 上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12)

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为使用内部上拉电阻的引脚并且满足以下使用条件的位，以位为单位使用内部上拉电阻。

内部上拉电阻的使用条件：

- PMmn=1 (输入模式)
- PMCmn=1 (数字输入/输出)
- POM0n=0 (通常输出模式)

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，PU4 的值变为“01H”、PU12 的值变为“20H”、PU0 的值变为“00H”。

图 4-3 上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU12	0	0	PU125注	0	0	0	0	0	F003CH	20H	R/W

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125注	0	0	0	0	0	F003CH	20H	R/W

PUmn	Pmn 的内部上拉电阻的选择
0	不连接内部上拉电阻。
1	连接内部上拉电阻。

m=0、4、12, n=0 ~ 7

注 只有在设定为 P125/KR1 引脚 (PORTSELB=0) 时才能操作。
(在设定为 RESET 输入 (PORTSELB=1) 时, 总是有效 (PU125=1)。)

注意 必须给未配置的位设定初始值。

4.3.4 端口输出模式寄存器 0 (POM0)

这是以位为单位设定 COMS 输出 /N 沟道漏极开路输出的寄存器。

在和外部器件进行简易 I²C 通信时，SDA00 引脚能选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

通过 1 位或者 8 位存储器操作指令设定端口输出模式寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注意 设定为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式 (POM0n=1) 的位，不能连接内部上拉电阻。

图 4-4 端口输出模式寄存器 0 (POM0) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	0	0	0	POM01	POM00	F0050H	00H	R/W

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	POM07	POM06	0	0	0	0	POM01	POM00	F0050H	00H	R/W

POM0n	P0n 引脚的输出模式的选择
0	通常的输出模式
1	N 沟道漏极开路输出 (V _{DD} 耐压) 模式

n=0、1、6、7

注意 必须给未配置的位设定初始值。

4.3.5 端口模式控制寄存器 0 (PMC0)

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 4-5 端口模式控制寄存器 0 (PMC0) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMC0n	P0n 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

n=1 ~ 7

注意 1. 必须通过端口模式寄存器 0 (PM0) 将通过 PMC0 寄存器设定为模拟输入的端口设定为输入模式。

2. 必须给未配置的位设定初始值。

4.3.6 外围 I/O 重定向寄存器 (PIOR)

这是设定允许或者禁止外围 I/O 重定向功能的寄存器。

外围 I/O 重定向功能切换被分配了复用功能的端口。

必须通过 PIOR 寄存器将端口分配给重定向功能，然后置为允许重定向功能的运行。能在允许重定向功能的运行前更改重定向的设定。

通过 8 位存储器操作指令设定 PIOR 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 4-6 外围 I/O 重定向寄存器 (PIOR) 的格式

10 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIOR	0	0	0	0	0	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

16 引脚产品

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIOR	0	0	0	PIOR4	PIOR3	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

位	复用功能	设定值	
		0	1
PIOR4 注	INTP3	P06	P121
PIOR3 注	INTP2	P41	P122
PIOR2	INTP1	P00	P03
PIOR1	TI01/TO01	P04	P40
PIOR0	PCLBUZ0	P02	P40

注 只限于 16 引脚产品。

注意 1. 禁止同时设定 PIOR0=1、PIOR1=1。

2. 必须给未配置的位设定初始值。

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，并且从引脚输出输出锁存器的内容。

保持被写在输出锁存器中的数据，直到下次写数据为止。

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操作。

保持被写在输出锁存器中的数据，直到下次写数据为止。

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，并且从引脚输出输出锁存器的内容。

保持被写在输出锁存器中的数据，直到下次写数据为止。

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操作。

在产生复位信号时清除输出锁存器的数据。

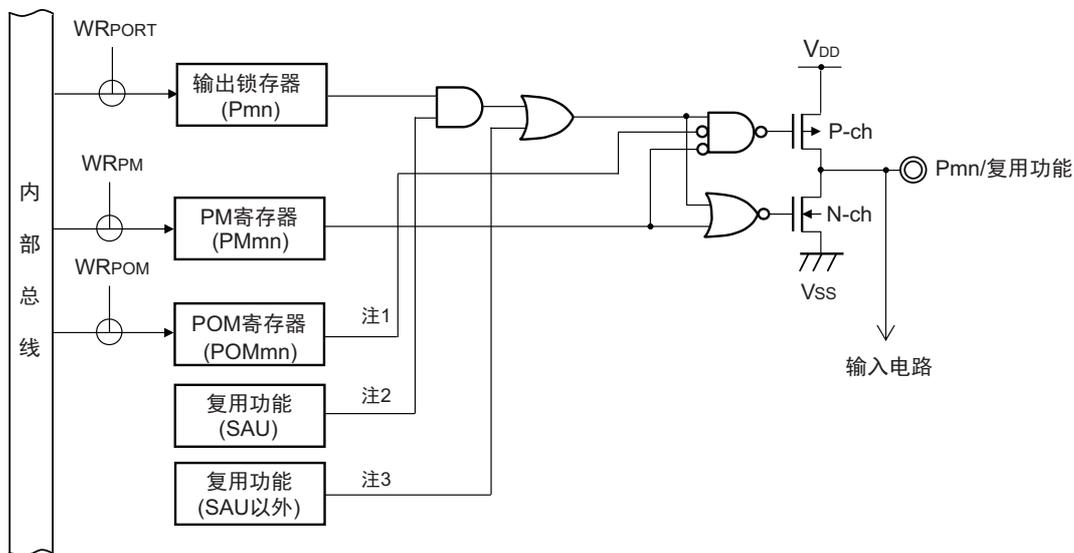
4.5 使用复用功能时的寄存器设定

4.5.1 使用复用功能时的基本观点

通过端口模式控制寄存器 0 (PMC0) 设定是将与模拟输入复用的引脚用作模拟输入还是用作数字输入 / 输出。

用作数字输入 / 输出的引脚的输出电路的基本结构如图 4-7 所示。与端口输出锁存器的输出复用的 SAU 功能的输出被输入到 AND 门。AND 门的输出被输入到 OR 门。OR 门的其他输入与复用的 SAU 以外的功能 (TAU、时钟 / 蜂鸣器的输出、IICA 等) 的输出连接。将这样的引脚用作端口功能或者复用功能时, 必须注意不使用的复用功能不能影响要使用的功能的输出。此时的基本设定的观点如表 4-4 所示。

图 4-7 引脚输出电路的基本结构



- 注 1. 在无 POM 寄存器时, 此信号为 Low 电平 (0)。
- 2. 在无复用功能时, 此信号为 High 电平 (1)。
- 3. 在无复用功能时, 此信号为 Low 电平 (0)。

备注 m: 端口号 (m=0、4、12、13), n: 位号 (n=0 ~ 7)

表 4-4 基本设定的观点

使用引脚的输出功能	不使用的复用功能的输出设定		
	端口功能	SAU 的输出功能	SAU 以外的输出功能
端口输出功能	—	High 电平输出 (1)	Low 电平输出 (0)
SAU 的输出功能	High (1)	—	Low 电平输出 (0)
SAU 以外的输出功能	Low (0)	High 电平输出 (1)	Low 电平输出 (0) 注

注 因为 1 个引脚有可能复用多个 SAU 以外的输出功能, 所以必须将不使用的复用功能的输出设定为 Low 电平 (0)。有关详细设定方法请参照“4.5.2 不使用输出功能的复用功能的寄存器设定”。

4.5.2 不使用输出功能的复用功能的寄存器设定

不使用引脚复用功能的输出时，必须执行以下设定。另外，在为外围 I/O 重定向功能的对象时，能通过设定外围 I/O 重定向寄存器（PIOR），将输出切换到其他引脚。这样就能使用分配到对象引脚的端口功能或者其他复用功能。

1. $SOp=1/TxDq=1$ （不使用 SAU 的串行输出（ $SOp/TxDq$ ）时的设定）
只使用 SAU 的串行输入，不使用串行输出（ $SOp/TxDq$ ）时，将与不使用的输出对应的串行输出允许寄存器 0（ $SOE0$ ）的位置“0”（禁止输出），并且将串行输出寄存器 0（ $SO0$ ）的 $SO0n$ 位置“1”（High 电平）。这是与初始状态相同的设定。
2. $SCKp=1/SDAr=1/SCLr=1$ （不使用 SAU 的通道 n 时的设定）
不使用 SAU 时，将串行通道允许状态寄存器 0（ $SE0$ ）的 bit n（ $SE0n$ ）置“0”（运行停止状态），并且将与不使用的输出对应的串行输出允许寄存器 0（ $SOE0$ ）的位置“0”（禁止输出），串行输出寄存器 0（ $SO0$ ）的 $SO0n$ 位和 $CKO0n$ 位置“1”（High 电平）。这是与初始状态相同的设定。
3. $TO0n=0$ （不使用 TAU 的通道 n 的输出时的设定）
不使用 TAU 的 $TO0n$ 输出时，将与不使用的输出对应的定时器输出允许寄存器 0（ $TOE0$ ）的位置“0”（禁止输出），并且将定时器输出寄存器 0（ $TO0$ ）的位置“0”（Low 电平）。这是与初始状态相同的设定。
4. $SDAA0=0/SCLA0=0$ （不使用 IICA 时的设定）
不使用 IICA 时，将 IICA 控制寄存器 00（ $IICCTL00$ ）的 $IICE0$ 位置“0”（运行停止）。这是与初始状态相同的设定。
5. $PCLBUZ0=0$ （不使用时钟输出/蜂鸣器输出时的设定）
不使用时钟输出/蜂鸣器输出时，将时钟输出选择寄存器 0（ $CKS0$ ）的 $PCLOE0$ 位置“0”（禁止输出）。这是与初始状态相同的设定。

4.5.3 使用的端口功能和复用功能的寄存器设定例子

使用的端口功能和复用功能的寄存器设定例子如表 4-5 所示。端口功能的控制寄存器的设定如表 4-5 所示。另外，有关表 4-5 的中的记载说明请参照以下备注。

备注	—:	对象外
	x:	忽略
	PIOR:	外围 I/O 重定向寄存器
	POM0:	端口输出模式寄存器 0
	PMC0:	端口模式控制寄存器 0
	PMn:	端口模式寄存器 n (n=0、4)
	Pm:	端口的输出锁存器 (m=0、4、12、13)

能通过设定外围 I/O 重定向寄存器（PIOR），分配 () 内的功能。

表 4-5 使用引脚功能时的寄存器、输出锁存器的设定例子 (1/4)

引脚名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	复用功能输出		16 引脚	10 引脚
	功能名称	输入 / 输出						SAU 的输出功能	SAU 以外		
P00	P00	输入	—	x	—	1	x	x	—	○	○
		输出	—	0	—	0	0/1	TxD0/SO00=1	—		
		N 沟道漏极开路输出	—	1	—	0	0/1				
	SO00	输出	—	0	—	0	1	x	—	○	○
	TXD0	输出	—	0/1	—	0	1	x	—	○	○
	INTP1	输入	PIOR2=0	x	—	1	x	x	—	○	○
P01	P01	输入	—	x	0	1	x	x	—	○	○
		输出	—	0	0	0	0/1	SDA00=1	—		
		N 沟道漏极开路输出	—	1	0	0	0/1				
	ANI0	模拟输入	—	x	1	1	x	x	—	○	○
	SI00	输入	—	x	0	1	x	x	—	○	○
	RxD0	输入	—	x	0	1	x	x	—	○	○
	SDA00	输入 / 输出	—	1	0	0	1	x	—	○	○
	KR2	输入	—	x	0	1	x	x	—	○	○
P02	P02	输入	—	—	0	1	x	x	x	○	○
		输出	—	—	0	0	0/1	SCK00/SCL00=1	PCLBUZ0=0 VCOUT0=0 注		
	ANI1	模拟输入	—	—	1	1	x	x	x	○	○
	SCK00	输入	—	—	0	1	x	x	x	○	○
		输出	—	—	0	0	1	x	PCLBUZ0=0 VCOUT0=0 注		
	SCL00	输出	—	—	0	0	1	x	PCLBUZ0=0 VCOUT0=0 注	○	○
	PCLBUZ0	输出	PIOR0=0	—	0	0	0	SCK00/SCL00=1	VCOUT0=0 注	○	○
	KR3	输入	—	—	0	1	x	x	x	○	○
VCOUT0	输出	—	—	0	0	0	SCK00/SCL00=1	PCLBUZ0=0	○	—	
P03	P03	输入	—	—	0	1	x	—	x	○	○
		输出	—	—	0	0	0/1	—	TO00=0		
	ANI2	模拟输入	—	—	1	1	x	—	x	○	○
	TO00	输出	—	—	0	0	0	—	x	○	○
	KR4	输入	—	—	0	1	x	—	x	○	○
	(INTP1)	输入	PIOR2=1	—	0	1	x	—	x	○	○
	IVCMP0	输入	—	—	1	1	x	—	x	○	—
P04	P04	输入	—	—	0	1	x	—	x	○	○
		输出	—	—	0	0	0/1	—	TO01=0		
	ANI3	模拟输入	—	—	1	1	x	—	x	○	○
	TI01	输入	PIOR1=0	—	0	1	x	—	x	○	○
	TO01	输出	PIOR1=0	—	0	0	0	—	x	○	○
	KR5	输入	—	—	0	1	x	—	x	○	○
	IVREF0	输入	—	—	1	1	x	—	x	○	—

注 只限于 16 引脚产品。

表 4-5 使用引脚功能时的寄存器、输出锁存器的设定例子 (2/4)

引脚名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	复用功能输出		16 引脚	10 引脚
	功能名称	输入 / 输出						SAU 的输出功能	SAU 以外		
P05	P05	输入	—	—	0	1	x	x	x	○	—
		输出	—	—	0	0	0/1	SO01=1	TO02=0	○	—
	ANI4	模拟输入	—	—	1	1	x	x	x	○	—
	TI02	输入	—	—	0	1	x	x	x	○	—
	TO02	输出	—	—	0	0	0	SO01=1	x	○	—
	SO01	输出	—	—	0	0	1	x	TO02=0	○	—
P06	P06	输入	—	x	0	1	x	—	x	○	—
		输出	—	0	0	0	0/1	—	SCLA0=0		
		N 沟道漏极开路输出	—	1	0	0	0/1				
	ANI5	模拟输入	—	x	1	1	x	—	x	○	—
	SI01	输入	—	x	0	1	x	—	x	○	—
	SCLA0	输入 / 输出	—	1	0	0	1	—	x	○	—
	INTP3	输入	PIOR4=0	x	0	1	x	—	x	○	—
P07	P07	输入	—	x	0	1	x	x	x	○	—
		输出	—	0	0	0	0/1	SCK01=1	TO03=0 SDAA0=0		
		N 沟道漏极开路输出	—	1	0	0	0/1				
	ANI6	模拟输入	—	x	1	1	x	x	x	○	—
	TO03	输出	—	0/1	0	0	0	SCK01=1	SDAA0=0	○	—
	SCK01	输入	—	x	0	1	x	x	x	○	—
输出		—	0	0	0	1	x	TO03=0 SDAA0=0			
	SDAA0	输入 / 输出	—	1	0	0	1	SCK01=1	TO03=0	○	—
P40	P40	输入	—	—	—	1	x	—	x	○	○
		输出	—	—	—	0	0/1	—	(PCLBUZ0)=0 (TO01)=0		
	KR0	输入	—	—	—	1	x	—	x	○	○
	(PCLBUZ0)	输出	PIOR0=1	—	—	0	0	—	(TO01)=0	○	○
	(TI01)	输入	PIOR1=1	—	—	1	x	—	x	○	○
	(TO01)	输出	PIOR1=1	—	—	0	0	—	(PCLBUZ0)=0	○	○
P41	P41	输入	—	—	—	1	x	—	—	○	—
		输出	—	—	—	0	0/1	—	—		
	TI03	输入	—	—	—	1	x	—	—	○	—
	INTP2	输入	PIOR3=0	—	—	1	x	—	—	○	—

表 4-5 使用引脚功能时的寄存器、输出锁存器的设定例子 (3/4)

引脚名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	CMC (EXCLK、OSCSEL)	16 引脚	10 引脚
	功能名称	输入 / 输出								
P121	P121	输入	—	—	—	—	x	00/10/11	○	—
	X1	输入	—	—	—	—	x	01	○	—
	(INTP3)	输入	PIOR4=1	—	—	—	x	00/10/11	○	—
P122	P122	输入	—	—	—	—	x	00/10	○	—
	X2	输入	—	—	—	—	x	01	○	—
	EXCLK	输入	—	—	—	—	x	11	○	—
	(INTP2)	输入	PIOR3=1	—	—	—	x	00/10	○	—

表 4-5 使用引脚功能时的寄存器、输出锁存器的设定例子 (4/4)

引脚名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	备注	16 引脚	10 引脚
	功能名称	输入 / 输出								
P125	P125	输入	—	—	—	—	x	选项字节 000C1H PORTSELB=0	○	○
	KR1	输入	—	—	—	—	x		○	○
	RESET	输入	—	—	—	—	x	选项字节 000C1H PORTSELB=1	○	○
P137	P137	输入	—	—	—	—	x	—	○	○
	TI00	输入	—	—	—	—	x	—	○	○
	INTP0	输入	—	—	—	—	x	—	○	○

4.6 功能时的注意事项

4.6.1 有关对端口寄存器 n (Pn) 的位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 P00 为输出端口并且 P01 ~ P07 为输入端口（引脚状态全部为高电平）而且端口 0 的输出锁存器值为“00H”时，如果通过位操作指令将输出端口 P00 的输出从低电平改为高电平，端口 0 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 RL78 微控制器内部，位操作指令按照以下顺序执行：

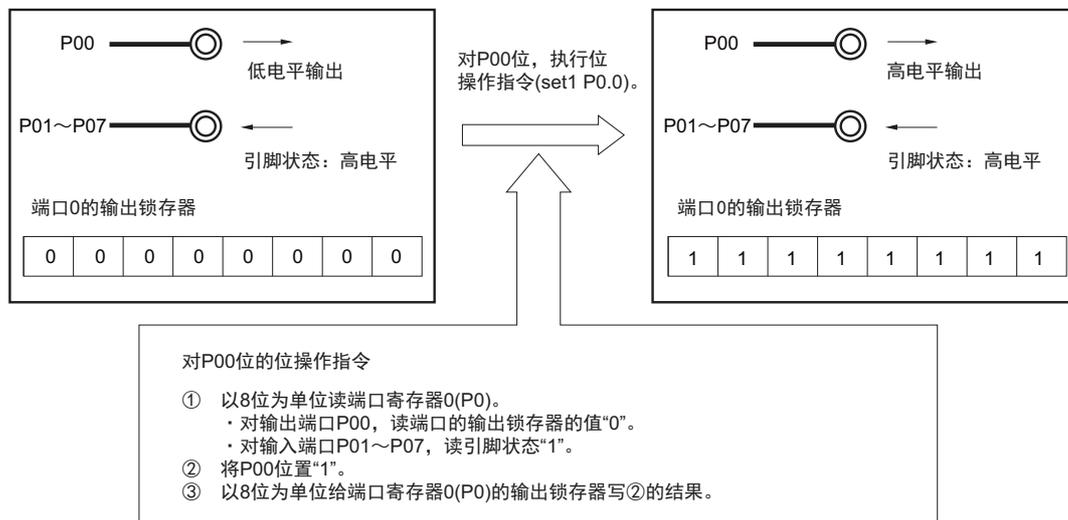
- <1> 以 8 位为单位读 Pn 寄存器。
- <2> 操作 1 位对象位。
- <3> 以 8 位为单位写 Pn 寄存器。

在 <1> 时，读输出端口 P00 的输出锁存器的值“0”，并且读输入端口 P01 ~ P07 的引脚状态。此时，如果 P01 ~ P07 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 4-8 位操作指令（P00 的情况）



4.6.2 引脚设定的注意事项

有关分配了多个复用功能的输出引脚，需要将不使用的复用功能的输出设置为初始状态（回避输出的冲突），在通过设定外围 I/O 重定向寄存器（PIOR）给引脚分配功能时也这样。有关复用输出，请参照“4.5 使用复用功能时的寄存器设定”。

对于用作输入的引脚，复用功能的输出无效（缓冲器输出为 Hi-Z），因此不需要进行处理。

另外，为了实现低功耗，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。

有以下几种系统时钟和时钟振荡电路。

(1) 主系统时钟

① X1 振荡电路（只限于 16 引脚产品）

能通过给 X1 引脚和 X2 引脚连接谐振器使 $f_X=1 \sim 20\text{MHz}$ 的时钟振荡。

能从 EXCLK/X2/P122 引脚提供外部主系统时钟（ $f_{EX}=1 \sim 20\text{MHz}$ ）。

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）使谐振器停止振荡，或者使外部主系统时钟的输入无效。

② 高速内部振荡器

能通过选项字节（000C2H）从 $f_{IH}=20\text{MHz}$ 、10MHz、5MHz、2.5MHz 和 1.25MHz(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。

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，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]				
	1.25	2.5	5	10	20
$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	○	○	○	○	○
$2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注	○	○	○	—	—

备注 ○：能选择、—：不能选择

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）和高速内部振荡器时钟的切换。

注 因为包括可选择的上电复位（SPOR）电路的检测电压（ V_{SPOR} ），所以必须在 2.25 ~ 5.5V 的电压范围内使用。

(2) 低速内部振荡器时钟

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。

只有以下外围硬件能通过低速内部振荡器时钟运行：

- 看门狗定时器
- 12 位间隔定时器注

当选项字节 (000C0H) 的 bit4 (WDTON) 和运行速度模式控制寄存器 (OSMC) 注的 bit4 (WUTMMCK0) 中的一个或者两个都为“1”时，低速内部振荡器时钟振荡。

但是，在 WDTON 位为“1”并且 WUTMMCK0 位为“0”而且选项字节 (000C0H) 的 bit0 (WDSTBYON) 为“0”时，如果执行 HALT 指令或者 STOP 指令，低速内部振荡器就停止振荡。

注 只限于 16 引脚产品。

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_{IL} : 低速内部振荡器的时钟频率

5.2 时钟发生电路的结构

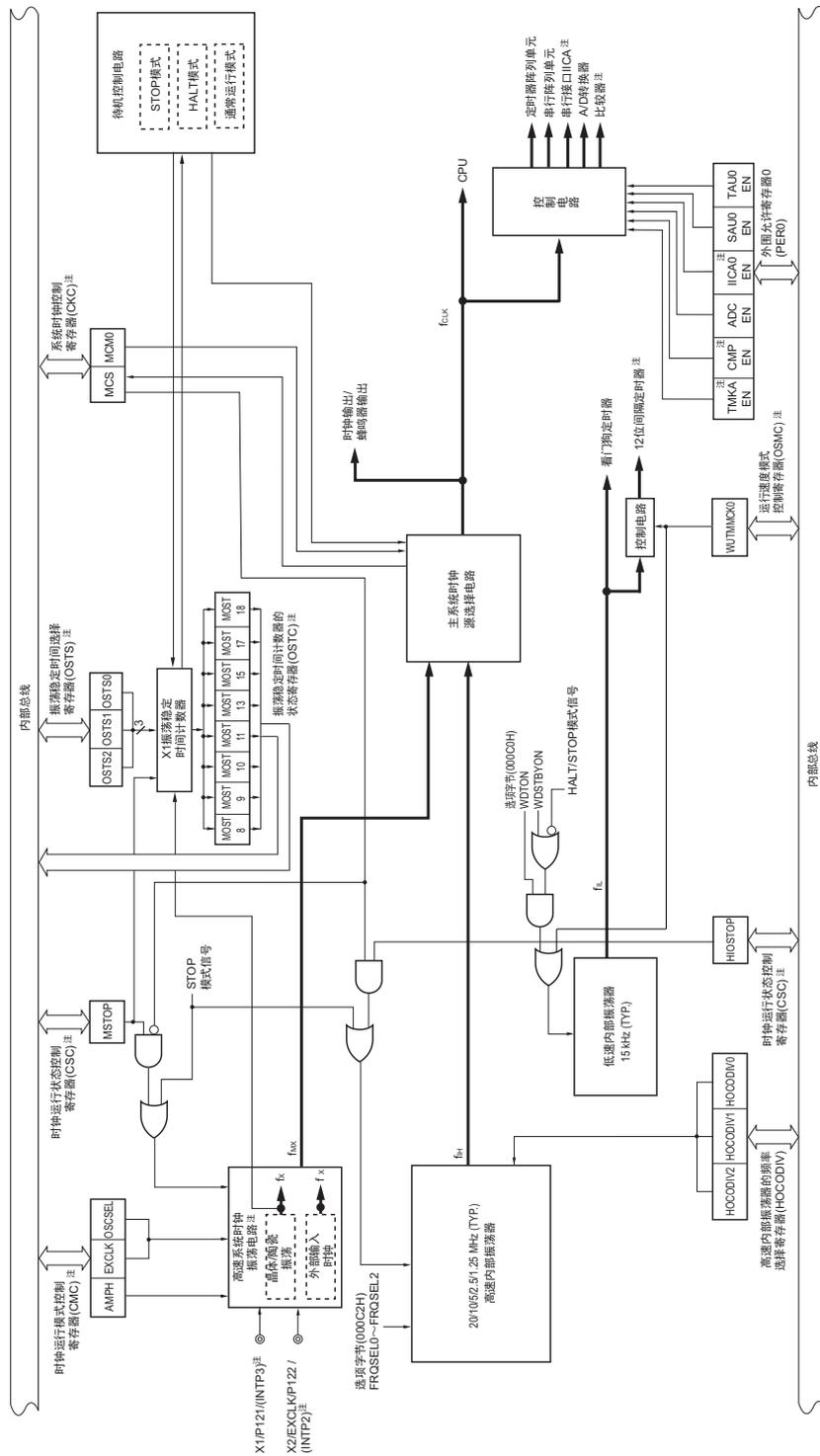
时钟发生电路由以下硬件构成。

表 5-1 时钟发生电路的结构

项目	结构	10 引脚产品	16 引脚产品
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 (PER0)	○	○
	运行速度模式控制寄存器 (OSMC)	—	○
	高速内部振荡器的频率选择寄存器 (HOCODIV)	○	○
振荡电路	X1 振荡电路	—	○
	高速内部振荡器	○	○
	低速内部振荡器	○	○

备注 ○: 能选择、—: 不能选择

图 5-1 时钟发生电路的框图



注 仅限于 16 引脚产品。

- 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 时钟发生电路的控制寄存器

根据产品通过以下的寄存器控制时钟发生电路。

1. 10 引脚产品
 - 外围允许寄存器0 (PER0)
 - 高速内部振荡器频率选择寄存器 (HOCODIV)

2. 16 引脚产品
 - 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 (PER0)
 - 运行速度模式控制寄存器 (OSMC)
 - 高速内部振荡器的频率选择寄存器 (HOCODIV)

注意 配置的寄存器和位因产品而不同。必须给未配置的寄存器和位设定初始值。

5.3.1 时钟运行模式控制寄存器 (CMC)

这是设定 X1/P121/(INTP3)、X2/EXCLK/P122/(INTP2) 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/(INTP3) 引脚	X2/EXCLK/P122/(INTP2) 引脚
0	0	输入端口模式	输入端口	
0	1	X1 振荡模式	连接晶体振荡器或者陶瓷谐振器。	
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	$1\text{MHz} \leq f_x \leq 10\text{MHz}$
1	$10\text{MHz} < f_x \leq 20\text{MHz}$

- 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. 在 X1 时钟振荡频率大于 10MHz 时，必须将 AMPH 位置“1”。在复位解除后选择 f_{IH} 作为 f_{CLK} 的状态（将 f_{CLK} 切换为 f_{MX} 前）下设定 AMPH 位。
4. 在 MSTOP=1 的状态下切换 X1/X2 引脚的运行模式。

备注 f_x : X1 时钟频率

5.3.2 系统时钟控制寄存器 (CKC)

这是选择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_{MAIN}) 的状态
0	高速内部振荡器时钟 (f_{IH})
1	高速系统时钟 (f_{MX})

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

注 bit5 是只读位。

注意 1. 必须将 bit0 ~ 3、bit6 和 bit7 置“0”。

2. 如果选择高速系统时钟 (f_{MX}) 作为主系统时钟 (f_{MAIN})，必须开始 f_{MX} 的振荡，并且在经过振荡稳定时间后才能进行切换。
3. 如果更改主系统时钟频率 (f_{MAIN})，外围硬件时钟也同时被更改。如果要更改 f_{MAIN} ，必须先停止各外围功能后进行 MCM0 位的设定。

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（低速内部振荡器时钟除外）运行的寄存器。

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“80H”。

图 5-4 时钟运行状态控制寄存器（CSC）的格式

地址：FFFA1H 复位后：80H 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	高速内部振荡器时钟运行
1	高速内部振荡器时钟停止

- 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. 必须在 MSTOP=1 的状态下切换 X1/X2 引脚的运行模式。
 3. 如果要设定 MSTOP=0，必须先将 X1/X2 引脚切换至 f_X 运行模式。在输入端口模式中，MSTOP 标志的设定无效。
 4. 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。
 5. 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。
 6. 如果要在 f_X 运行模式中设定 MSTOP=1，必须先确认 CKC 寄存器的 MCS 位是否为“0”。
 7. 在 f_X 运行时，MSTOP 标志的写操作有效，但是不进行停止控制。
 8. 不能通过 CSC 寄存器停止选择为 CPU/ 外围硬件时钟（ f_{CLK} ）的时钟。
 9. 有关用于停止时钟振荡的寄存器的标志设定和停止前的条件，请参照表 5-2。

表 5-2 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	CSC 寄存器的标志设定
X1 时钟	CPU/ 外围硬件时钟以高速内部振荡器时钟运行（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系统时钟运行（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状态寄存器。

能在以下情况下确认 X1 时钟的振荡稳定时间：

- 当 CPU 时钟为高速内部振荡器时钟并且开始 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、OSCSEL=0、1→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 $(2^8+16)/f_X$	小于 27.2 μs	小于 13.6 μs
1	0	0	0	0	0	0	0	至少 $(2^8+16)/f_X$	至少 27.2 μs	至少 13.6 μs
1	1	0	0	0	0	0	0	至少 $(2^9+16)/f_X$	至少 52.8 μs	至少 26.4 μs
1	1	1	0	0	0	0	0	至少 $(2^{10}+16)/f_X$	至少 104 μs	至少 52.0 μs
1	1	1	1	0	0	0	0	至少 $(2^{11}+16)/f_X$	至少 206 μs	至少 103 μs
1	1	1	1	1	0	0	0	至少 $(2^{13}+16)/f_X$	至少 820 μs	至少 410 μs
1	1	1	1	1	1	0	0	至少 $(2^{15}+16)/f_X$	至少 3.27ms	至少 1.63ms
1	1	1	1	1	1	1	0	至少 $(2^{17}+16)/f_X$	至少 13.1ms	至少 6.55ms
1	1	1	1	1	1	1	1	至少 $(2^{18}+16)/f_X$	至少 26.2ms	至少 13.1ms

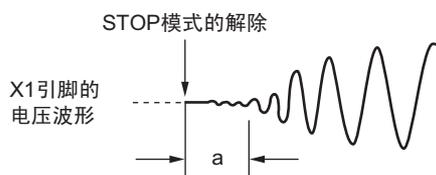
注意 1. 在经过上述时间后，各位从 MOST8 位开始依次变为“1”并且保持“1”的状态。

2. 振荡稳定时间计数器只在振荡稳定时间选择寄存器（OSTS）所设振荡稳定时间内进行计数。

在以下情况下，OSTS 寄存器的振荡稳定时间的设定值必须大于振荡开始后通过 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，在解除 STOP 模式后只给 OSTC 寄存器设定 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)

这是选择解除 STOP 模式时的 X1 时钟的振荡稳定时间的寄存器。

选择 X1 时钟作为 CPU 时钟时，必须在解除 STOP 模式后，自动等待 OSTS 寄存器设定的时间。

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。

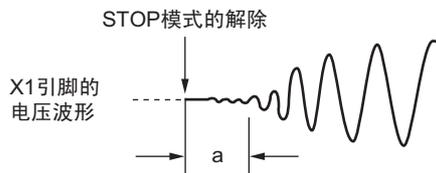
在产生复位信号后，此寄存器的值变为“07H”。

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	
				$f_X=10\text{MHz}$	$f_X=20\text{MHz}$
0	0	0	$(2^8+16)/f_X$	27.2 μs	13.6 μs
0	0	1	$(2^9+16)/f_X$	52.8 μs	26.4 μs
0	1	0	$(2^{10}+16)/f_X$	104 μs	52.0 μs
0	1	1	$(2^{11}+16)/f_X$	206 μs	103 μs
1	0	0	$(2^{13}+16)/f_X$	820 μs	410 μs
1	0	1	$(2^{15}+16)/f_X$	3.27ms	1.63ms
1	1	0	$(2^{17}+16)/f_X$	13.1ms	6.55ms
1	1	1	$(2^{18}+16)/f_X$	26.2ms	13.1ms

- 注意 1. 如果要在 CPU 时钟为 X1 时钟时转移到 STOP 模式，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，必须在将时钟运行状态控制寄存器 (CSC) 的 MSTOP 位置“0”前进行更改。
3. 在 X1 时钟的振荡稳定时间中，不能更改 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。
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，在解除 STOP 模式后只给 OSTC 寄存器设定 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_X : X1 时钟振荡频率

5.3.6 外围允许寄存器 0 (PER0)

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，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”。

- 12位间隔定时器
- A/D转换器
- 比较器
- 串行接口 IICA
- 串行阵列单元
- 定时器阵列单元

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	TMKAEN ^注	CMPEN ^注	ADCEN	IICA0EN ^注	0	SAU0EN	0	TAU0EN

TMKAEN	控制 12 位间隔定时器的输入时钟的提供
0	停止提供输入时钟。 • 不能写 12 位间隔定时器使用的 SFR。 • 12 位间隔定时器处于复位状态。
1	提供输入时钟。 • 能读写 12 位间隔定时器使用的 SFR。

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

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

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

注 只限于 16 引脚产品。

注意 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

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	TMKAEN注	CMPEN注	ADCEN注	IICA0EN注	0	SAU0EN	0	TAU0EN

SAU0EN	控制串行阵列单元的输入时钟的提供
0	停止提供输入时钟。 • 不能写串行阵列单元使用的 SFR。 • 串行阵列单元处于复位状态。
1	提供输入时钟。 • 能读写串行阵列单元 0 使用的 SFR。

TAU0EN	控制定时器阵列单元的输入时钟的提供
0	停止提供输入时钟。 • 不能写定时器阵列单元使用的 SFR。 • 定时器阵列单元处于复位状态。
1	提供输入时钟。 • 能读写定时器阵列单元使用的 SFR。

注 只限于 16 引脚产品。

注意 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

5.3.7 运行速度模式控制寄存器 (OSMC)

OSMC 寄存器控制 12 位间隔定时器的运行时钟的提供。

如果要运行 12 位间隔定时器, 必须先将 WUTMMCK0 置“1”, 并且到停止前不能将 WUTMMCK0 置“0”。

通过 8 位存储器操作指令设定 OSMC 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 5-8 运行速度模式控制寄存器 (OSMC) 的格式

地址: F00F3H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	控制 12 位间隔定时器的运行时钟的提供
0	停止提供时钟
1	提供低速内部振荡器时钟 (f_{IL})

5.3.8 高速内部振荡器的频率选择寄存器（HOCODIV）

这是更改选项字节（000C2H）设定的高速内部振荡器时钟频率的寄存器。

通过 8 位存储器操作指令设定 HOCODIV 寄存器。

在产生复位信号后，此寄存器的值变为通过选项字节（000C2H）FRQSEL2 ~ FRQSEL0 设定的值。

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）FRQSEL2 ~ FRQSEL0 的设定置 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
0	0	1	20MHz
0	1	0	10MHz
0	1	1	5MHz
1	0	0	2.5MHz
1	0	1	1.25MHz
上述以外			禁止设定

注意 1. 更改频率前后，都必须在可工作电压范围内进行 HOCODIV 寄存器的设定。

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，最多运行 3 个时钟。
- 以更改后的频率，最多等待 3 个 CPU/ 外围硬件时钟。

5.4 系统时钟振荡电路

5.4.1 X1 振荡电路（只限于 16 引脚产品）

X1 振荡电路通过连接 X1 引脚和 X2 引脚的晶体振荡器或者陶瓷谐振器（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引脚时，请参照“表 2-2 各未使用引脚的处理”。

X1 振荡电路的外接电路例子如图 5-10 所示。

图 5-10 X1 振荡电路的外接电路例子

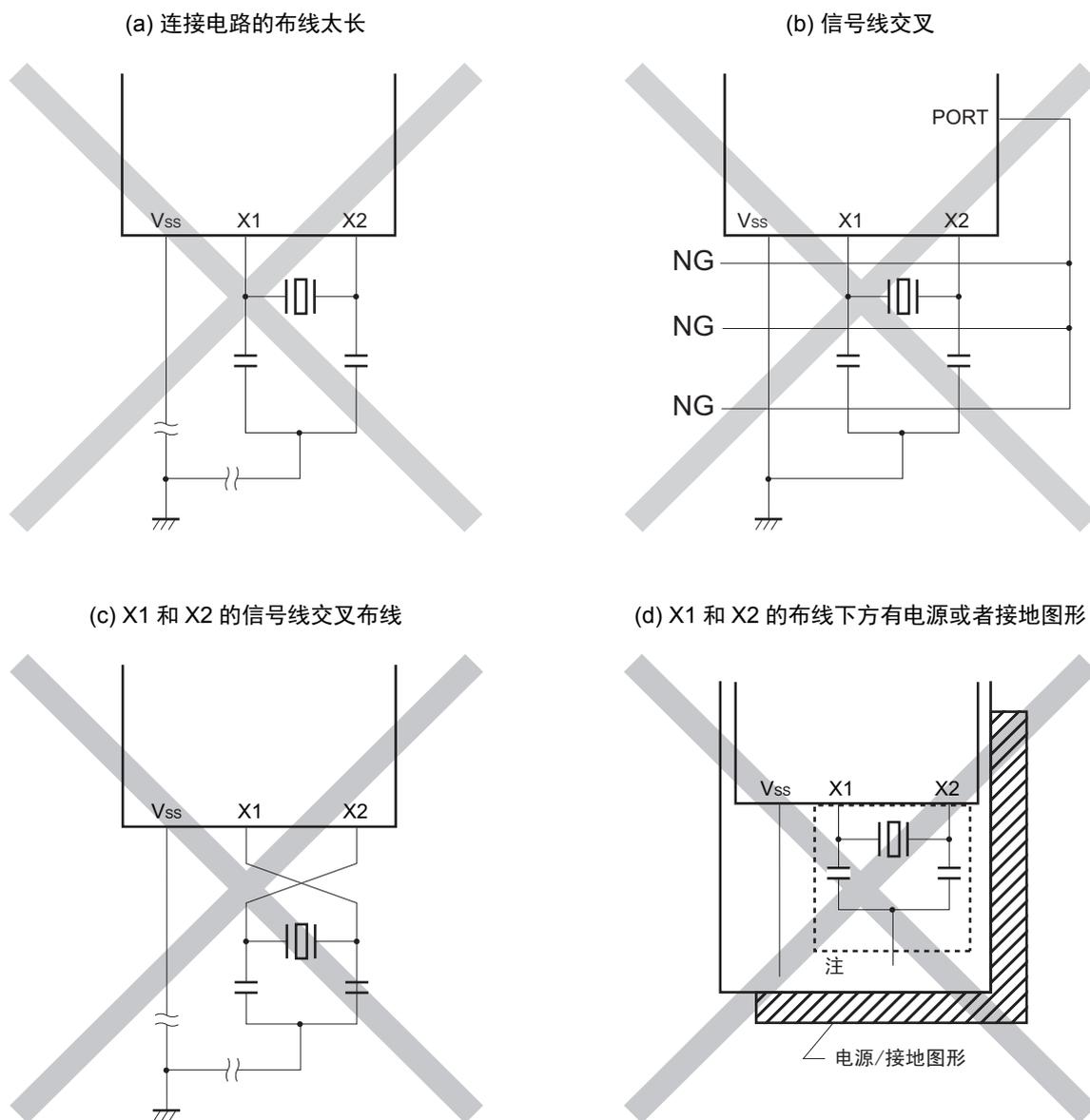


注意 当使用 X1 振荡电路时，为了避免布线电容等的影响，必须通过以下方法对图 5-10 中的虚线部分中进行布线：

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

不正确的谐振器连接例子如图 5-11 所示。

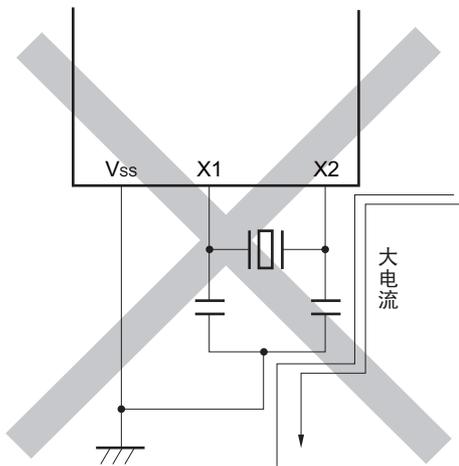
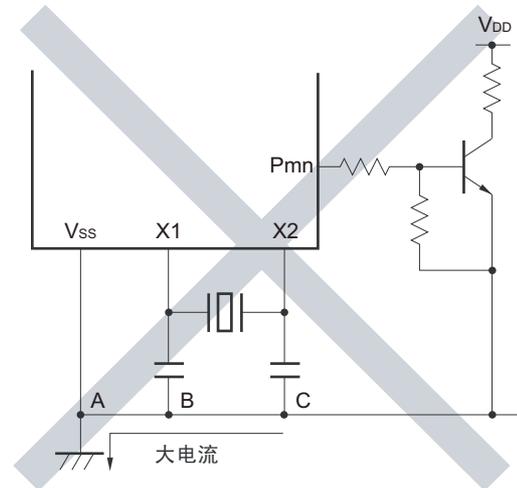
图 5-11 不正确的谐振器连接例子 (1/2)



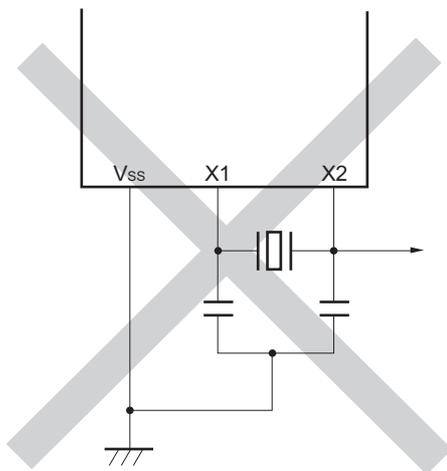
注 在多层板或者双面板中，不能在 X1 引脚、X2 引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

图 5-11 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) 振荡电路的接地线上有电流流过
(A 点、B 点、C 点的电位发生变动)

(g) 取出信号



5.4.2 高速内部振荡器

RL78/G10 内置高速内部振荡器。能通过选项字节（000C2H）从 20MHz、10MHz、5MHz、2.5MHz 和 1.25MHz 中选择频率。能通过时钟运行状态控制寄存器（CSC）注的 bit0（HIOSSTOP）控制振荡。

在解除复位后，高速内部振荡器自动开始振荡。

注 只限于 16 引脚产品。

5.4.3 低速内部振荡器

RL78/G10 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和 12 位间隔定时器的时钟注，但是不能用作 CPU 时钟。

当选项字节（000C0H）的 bit4（WDTON）和运行速度模式控制寄存器（OSMC）注的 bit4（WUTMMCK0）中的 1 个或者两个都为“1”时，低速内部振荡器运行。

当看门狗定时器运行并且在 WUTMMCK0 位不为“0”时，低速内部振荡器继续振荡。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。

注 只限于 16 引脚产品。

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）。

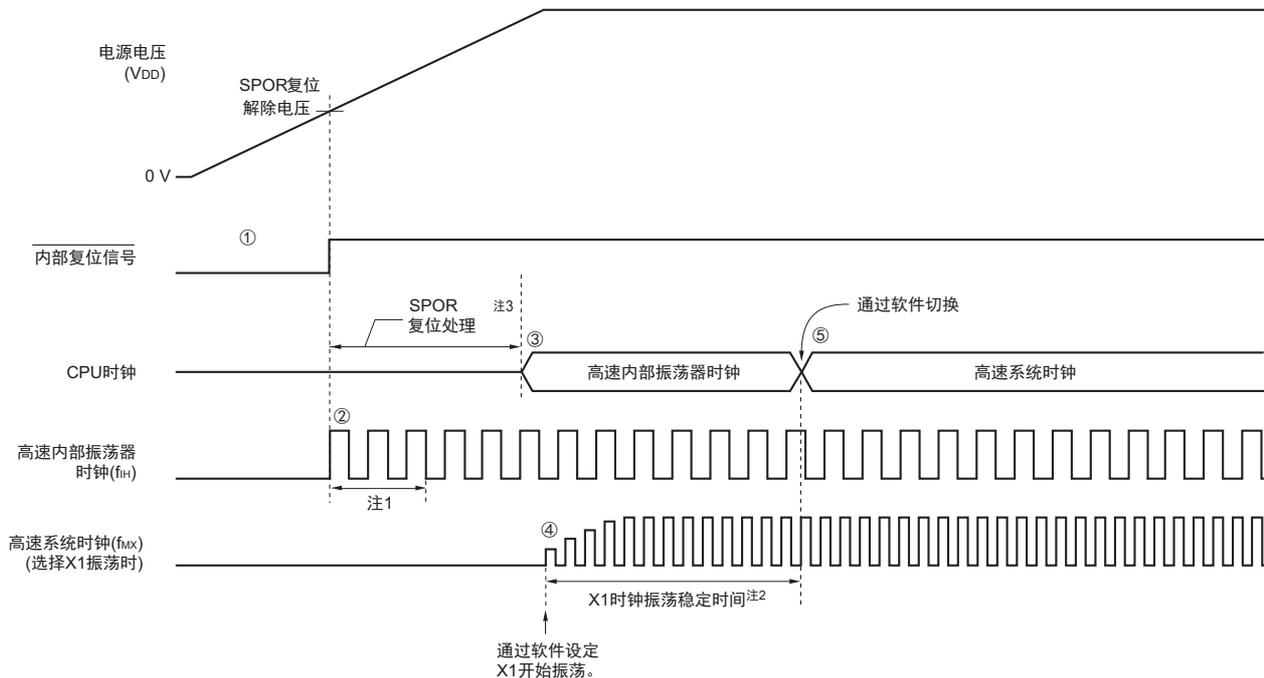
- 主系统时钟 f_{MAIN}
 - 高速系统时钟注 f_{MX}
 - X1 时钟注 f_{X}
 - 外部主系统时钟注 f_{EX}
 - 高速内部振荡器时钟 f_{IH}
- 低速内部振荡器时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

RL78/G10 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 5-12 所示。

注 只限于 16 引脚产品。

图 5-12 接通电源电压时的时钟发生电路的运行



- ① 在接通电源后，通过可选择的上电复位（SPOR）电路产生内部复位信号。
- ② 如果电源电压超出 SPOR 电路的检测电压，高速内部振荡器就在解除复位后自动开始振荡。
- ③ 在解除复位后，进行电压稳定等待和 SPOR 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。

- 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 SPOR 复位处理时间，请参照“第 18 章 可选择的上电复位电路”。

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，提供高速内部振荡器时钟作为 CPU/ 外围硬件时钟 (f_{CLK})。能通过选项字节 (000C2H) 的 FRQSEL0 ~ FRQSEL2 位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
	1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的频率
0	0	1	20MHz
0	1	0	10MHz
0	1	1	5MHz
1	0	0	2.5MHz
1	0	1	1.25MHz
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

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

HOCODIV2	HOCODIV1	HOCODIV0	选择频率
0	0	1	20MHz
0	1	0	10MHz
0	1	1	5MHz
1	0	0	2.5MHz
1	0	1	1.25MHz
上述以外			禁止设定

- 注意 1. 更改频率前后，都必须在可工作电压范围内进行 HOCODIV 寄存器的设定。
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，最多运行 3 个时钟。
 - 以更改后的频率，最多等待 3 个 CPU/ 外围硬件时钟。

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 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控制振荡的开始，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 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 OSCSEL 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	OSCSEL						AMPH
	0	1	0	0	0	0	0	0/1

AMPH 位：在 X1 时钟小于 10MHz 时，必须设定为“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 104 μ 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	0	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 104 μ 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

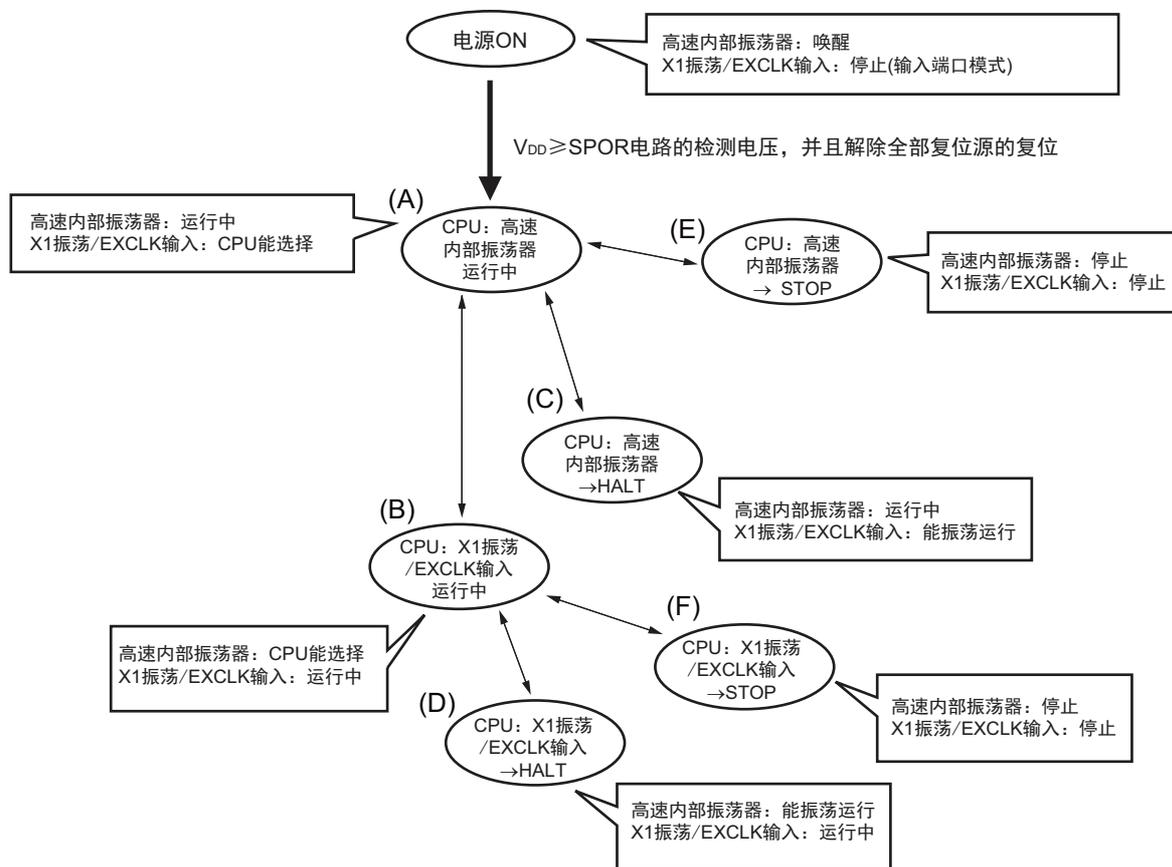
- ⑤ 通过 CKC 寄存器的 MCM0 位将 X1 时钟设定为 CPU/ 外围硬件时钟。

	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 5-13 所示。

图 5-13 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 (1/2)

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

(SFR 寄存器的设定顺序) →

SFR 寄存器的设定标志 状态转移	CMC 寄存器注 1			OSTS 寄存器	CSC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSSEL	AMPH		MSTOP		MCM0
(A)→(B) (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) (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) (外部主系统时钟)	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. 必须通过振荡稳定时间选择寄存器 (OSTS) 如下设定振荡稳定时间:
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 24 章 电特性”)后设定时钟。

备注 1. ×: 忽略

2. 表 5-3 的 (A) ~ (F) 对应图 5-13 的 (A) ~ (F)。

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

(SFR 寄存器的设定顺序) →

SFR 寄存器的设定标志 状态转移	CSC 寄存器	振荡精度稳定的等待	CKC 寄存器
	HIOSTOP		MCM0
(B)→(A)	0	27 μ s(TYP.)	0

在高速系统时钟运行过程中高速内部振荡器也运行时不需要设定。

表 5-3 CPU 时钟的转移和 SFR 寄存器的设定例子 (2/2)

- (3) • CPU 从高速内部振荡器时钟运行(A)转移到HALT模式(C)。
• CPU 从高速系统时钟运行(B)转移到HALT模式(D)。

状态转移	设定内容
(A)→(C) (B)→(D)	执行 HALT 指令。

- (4) • CPU 从高速内部振荡器时钟运行(A)转移到STOP模式(E)。
• CPU 从高速系统时钟运行(B)转移到STOP模式(F)。

(设定顺序) →

状态转移		设定内容		
(A)→(E)		停止在 STOP 模式中不能运行的外围功能	—	执行 STOP 指令
(B)→(F)	X1 振荡		设定 OSTS 寄存器	
	外部时钟		—	

备注 表 5-3 的 (A) ~ (F) 对应图 5-13 的 (A) ~ (F)。

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的振荡。 • HIOSTOP=0 • 经过振荡精度稳定时间后	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	允许高速内部振荡器的振荡。 • HIOSTOP=0 • 经过振荡精度稳定时间后	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 切换 CPU 时钟和主系统时钟所需要的时间

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)。

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 切换主系统时钟 ($f_{IH} \leftrightarrow f_{MX}$) 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 ($f_{MAIN}=f_{IH}$)	1 ($f_{MAIN}=f_{MX}$)
0 ($f_{MAIN}=f_{IH}$)	$f_{MX} \geq f_{IH}$		$1+f_{IH}/f_{MX}$
	$f_{MX} < f_{IH}$		2 个 f_{IH}/f_{MX} 时钟
1 ($f_{MAIN}=f_{MX}$)	$f_{MX} \geq f_{IH}$	2 个 f_{MX}/f_{IH} 时钟	
	$f_{MX} < f_{IH}$	$1+f_{MX}/f_{IH}$	

备注 1. 这是切换前的 CPU 时钟数。

2. 时钟数是舍入小数部分的时钟数。

例 主系统时钟从高速系统时钟切换到高速内部振荡器时钟的情况 (选择 $f_{IH}=5\text{MHz}$ 、 $f_{MX}=10\text{MHz}$ 振荡时)

$$2f_{MX}/f_{IH}=2(10/5)=4 \rightarrow 4 \text{ 个时钟}$$

5.6.6 时钟振荡停止前的条件

停止时钟振荡的寄存器的标志设定和停止前的条件如下所示。

表 5-6 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	SFR 寄存器的标志设定
高速内部振荡器时钟	MCS=1 (CPU 以高速系统时钟运行)	HIOSTOP=1
X1 时钟	MCS=0	MSTOP=1
外部主系统时钟	(CPU 以高速内部振荡器时钟运行)	

5.7 谐振器和振荡电路常数

经工作验证的谐振器及其振荡电路常数（供参考）如下所示。

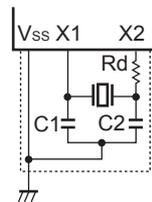
注意 1. 此振荡电路常数是基于谐振器厂商在特定环境下进行评估的参考值。在实际应用中，请谐振器厂商给予安装电路中的评估。

在更改其他产品的单片机和电路板时，请谐振器厂商再次给予安装电路中的评估。

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 5-14 外接振荡电路的例子

X1 振荡



第 6 章 定时器阵列单元

定时器阵列单元的通道个数因产品而不同。

通道	10 引脚产品	16 引脚产品
通道 0	○	○
通道 1	○	○
通道 2	—	○
通道 3	—	○

备注 ○：内置通道、—：未内置通道

定时器阵列单元最多有 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。



注 只限于 16 引脚产品。

有关各功能的详细内容，请参照下表。

独立通道运行功能	多通道联动运行功能
<ul style="list-style-type: none"> • 间隔定时器 (→ 参照 6.8.1) • 方波输出 (→ 参照 6.8.1) • 外部事件计数器 (→ 参照 6.8.2) • 分频器功能注² (→ 参照 6.8.3) • 输入脉冲间隔的测量 (→ 参照 6.8.4) • 输入信号的高/低电平宽度的测量 (→ 参照 6.8.5) • 延迟计数器 (→ 参照 6.8.6) 	<ul style="list-style-type: none"> • 单触发脉冲输出 (→ 参照 6.9.1) • 双输入式单触发脉冲输出注¹ (→ 参照 6.9.2) • PWM 输出功能 (→ 参照 6.9.3) • 多重 PWM 输出功能注¹ (→ 参照 6.9.4)

能将通道 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）。能使用通道 1 和通道 3 能的 8 位定时器的功能如下所示。

- 间隔定时器（高/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）
- PWM 输出功能（只限于低 8 位定时器）
- 多重 PWM 输出功能注¹（只限于低 8 位定时器）

能通过设定 ISC 寄存器，使通道 1 和串行阵列单元的 UART0 联合。通过使用输入脉冲间隔测量模式可以测量通信方的波特率宽度，从而实现 UART0 的波特率校正。

注 1. 只限于 16 引脚产品。

2. 只限于通道 0 和通道 3。

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。

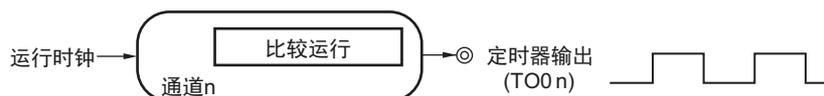
(1) 间隔定时器

能用作以固定间隔产生中断（INTTM0n）的基准定时器。



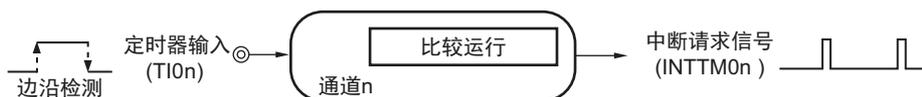
(2) 方波输出

每当产生 INTTM0n 中断时，就进行交替运行并且从定时器输出引脚（TO0n）输出 50% 占空比的方波。



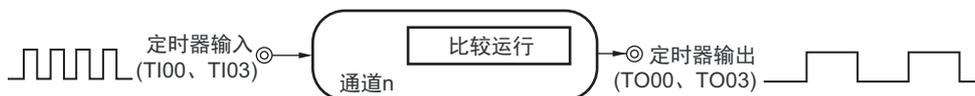
(3) 外部事件计数器

对定时器输入引脚（TI0n）的输入信号的有效边沿进行计数，如果达到规定次数，就能用作产生中断的事件计数器。



(4) 分频器功能（只限于通道 0 和通道 3）

对定时器输入引脚（TI00、TI03）的输入时钟进行分频，然后从输出引脚（TO00、TO03）输出。



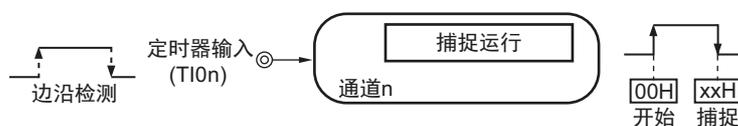
(5) 输入脉冲间隔的测量

在定时器输入引脚 (TI0n) 的输入脉冲信号的有效边沿开始计数并且在下一个脉冲的有效边沿捕捉计数值, 从而测量输入脉冲的间隔。



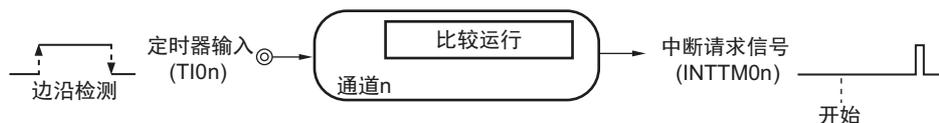
(6) 输入信号的高 / 低电平宽度的测量

在定时器输入引脚 (TI0n) 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电平或者低电平的宽度。



(7) 延迟计数器

在定时器输入引脚 (TI0n) 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 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

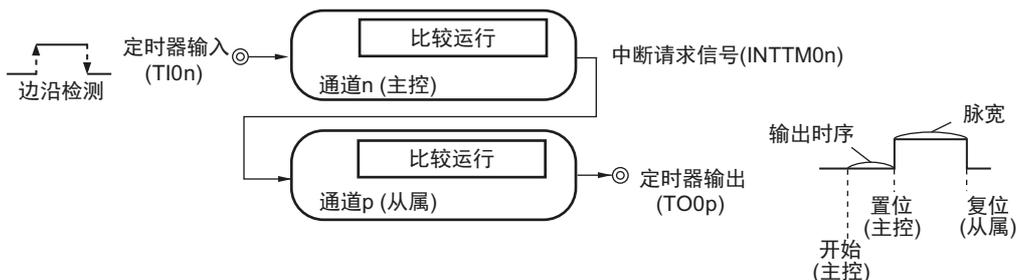
6.1.2 多通道联动运行功能

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。

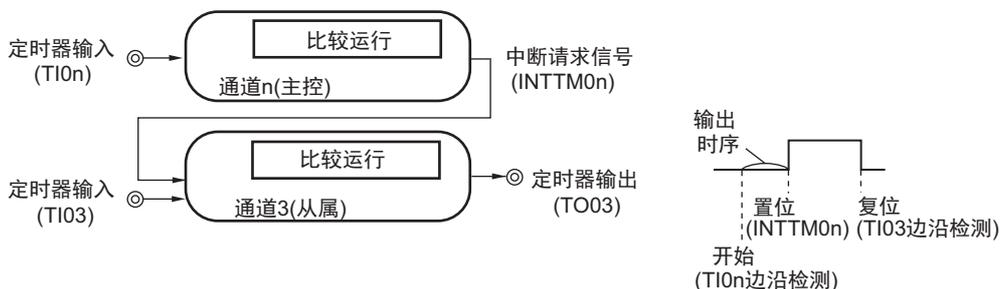
(1) 单触发脉冲输出

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) 双输入式单触发脉冲输出（只限于 16 引脚产品）

将 2 个通道成对使用，并且通过输入到定时器输入引脚（TI0n、TI03）的有效边沿，置位 / 复位定时器输出引脚（TO03），生成任意的单触发脉冲。



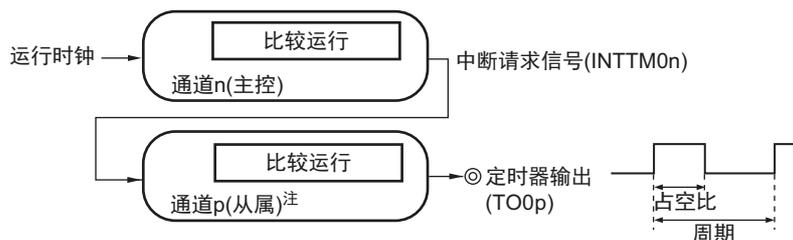
注意 有关多通道联动运行功能的使用规则的详细内容，请参照“6.4.1 多通道联动运行功能的基本规则”。

备注 n: 通道号。n=0（10 引脚产品）、n=0、2（16 引脚产品）

p: 从属通道号（ $0 < p \leq 3$ ）

(3) PWM（Pulse Width Modulation）输出功能

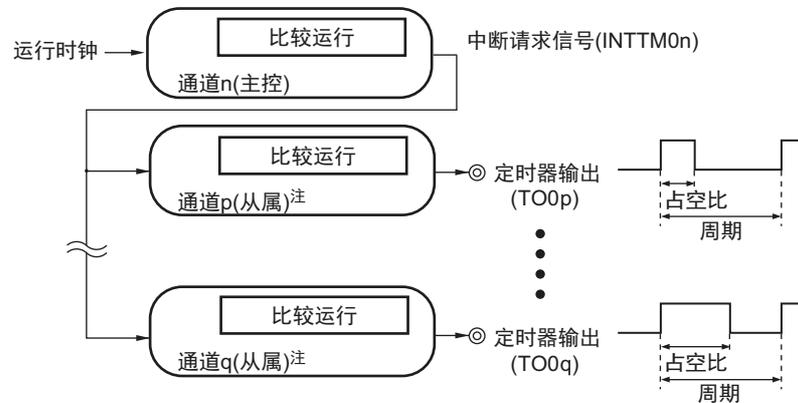
将 2 个通道成对使用，生成能任意设定周期和占空比的脉冲。



注 在通道 1 和通道 3 的低 8 位定时器模式中也能运行。

(4) 多重 PWM (Pulse Width Modulation) 输出功能 (只限于 16 引脚产品)

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 在通道 1 和通道 3 的低 8 位定时器中也能运行。

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

备注 n: 通道号。n=0 (10 引脚产品)、n=0、2 (16 引脚产品)

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通道 1 和通道 3 的 16 位定时器用作 2 个通道的 8 位定时器的功能。能使用通道 1 和通道 3 的 8 位定时器的功能如下所示。

- 间隔定时器 (高 8 位定时器、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)
- PWM 输出功能 (只限于低 8 位定时器)
- 多重 PWM 输出功能注 (只限于低 8 位定时器)

注 只限于 16 引脚产品。

注意 有关 8 位定时器运行功能的使用规则的详细内容，请参照“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)”。

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
定时器 / 计数器	定时器计数寄存器 0n (TCR0nH、TCR0nL)
寄存器	定时器数据寄存器 0n (TDR0nH、TDR0nL)
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	<p><单元设定部的寄存器></p> <ul style="list-style-type: none"> • 外围允许寄存器 0 (PER0) • 定时器时钟选择寄存器 0 (TPS0) • 定时器通道允许状态寄存器 0 (TE0、TEH0) • 定时器通道开始寄存器 0 (TS0、TSH0) • 定时器通道停止寄存器 0 (TT0、TTH0) • 定时器输出允许寄存器 0 (TOE0) • 定时器输出寄存器 0 (TO0) • 定时器输出电平寄存器 0 (TOL0) • 定时器输出模式寄存器 0 (TOM0) <p><各通道的寄存器></p> <ul style="list-style-type: none"> • 定时器模式寄存器 0n (TMR0nH、TMR0nL) • 定时器状态寄存器 0n (TSR0n) • 噪声滤波器允许寄存器 1 (NFEN1) • 输入切换控制寄存器 (ISC) • 端口模式控制寄存器 0 (PMC0) • 端口模式寄存器 0、4 (PM0、PM4) • 端口寄存器 0、4 (P0、P4)

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

定时器阵列单元的各通道的定时器输入 / 输出的复用端口因产品而不同。

表 6-2 各产品配置的定时器输入 / 输出引脚

定时器阵列单元的通道	10 引脚产品	16 引脚产品
通道 0	P03/TO00、P137/TI00	P03/TO00、P137/TI00
通道 1	P04/TI01/TO01、(P40/TI01/TO01)	P04/TI01/TO01、(P40/TI01/TO01)
通道 2	—	P05/TI02/TO02
通道 3	—	P41/TI03、P07/TO03

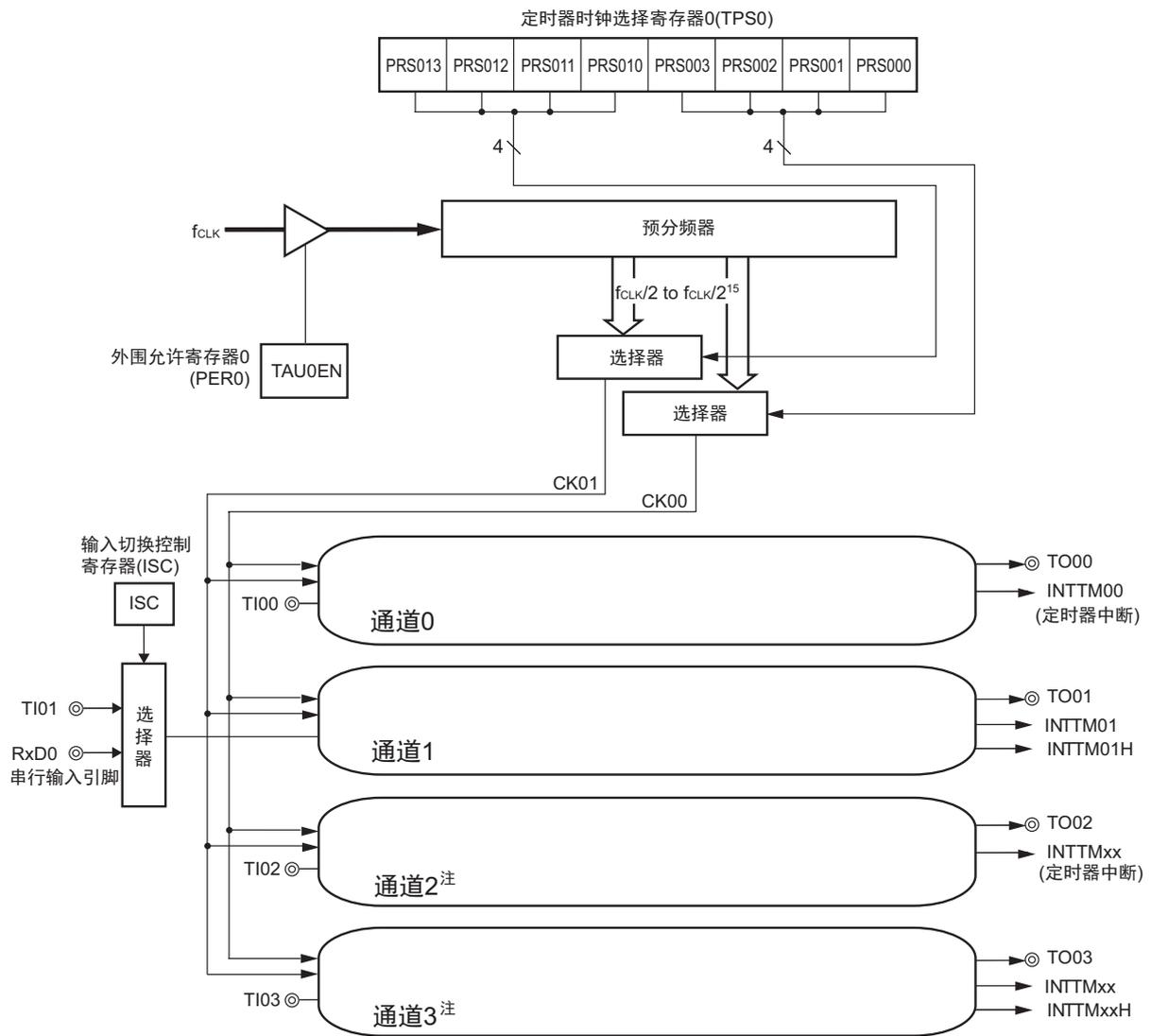
备注 1. 在定时器输入和定时器输出被同一个引脚复用时, 只能用作定时器输入或者定时器输出。

2. —: 没有内置通道。

3. () 是将外围 I/O 重定向寄存器 (PIOR) 的 PIOR0 位置“1”时的复用端口。

定时器阵列单元的框图如图 6-1、图 6-2 所示。

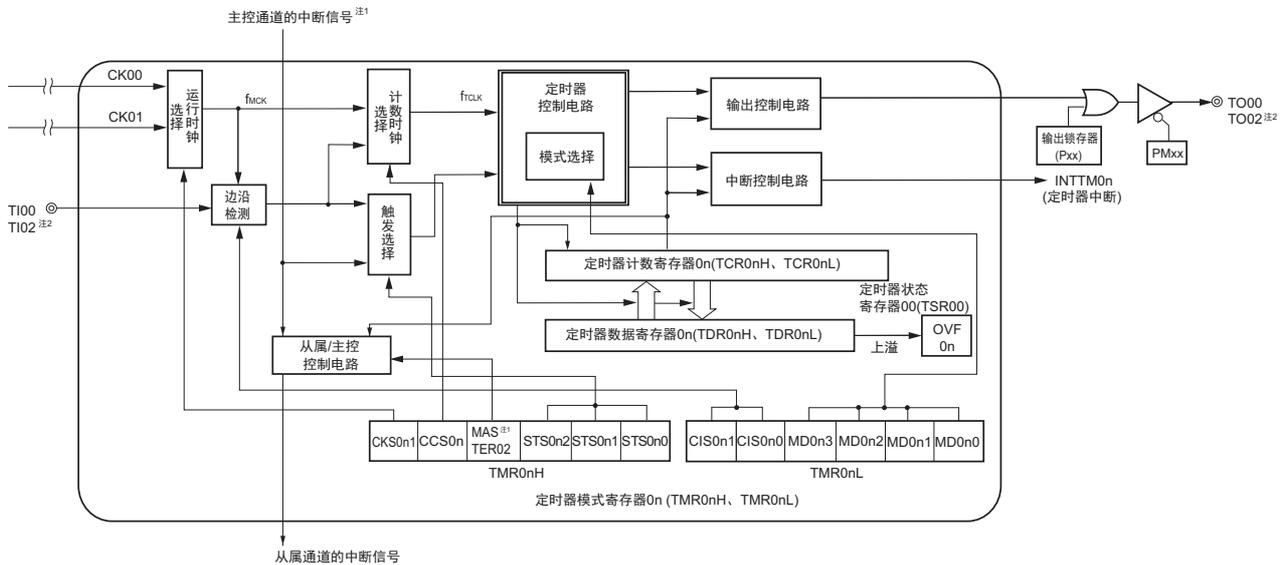
图 6-1 定时器阵列单元的整体框图



注 只限于 16 引脚产品。

图 6-2 定时器阵列单元的通道内部框图

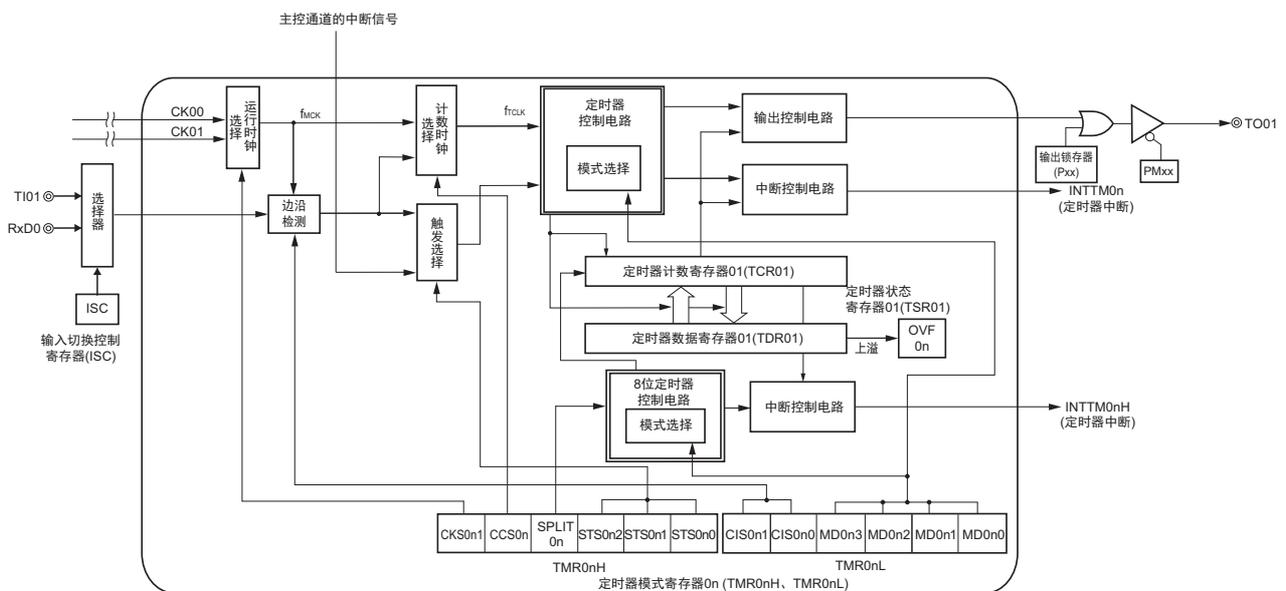
(a) 通道 0、2



- 注 1. 只限于通道 2。
- 2. 只限于 16 引脚产品。

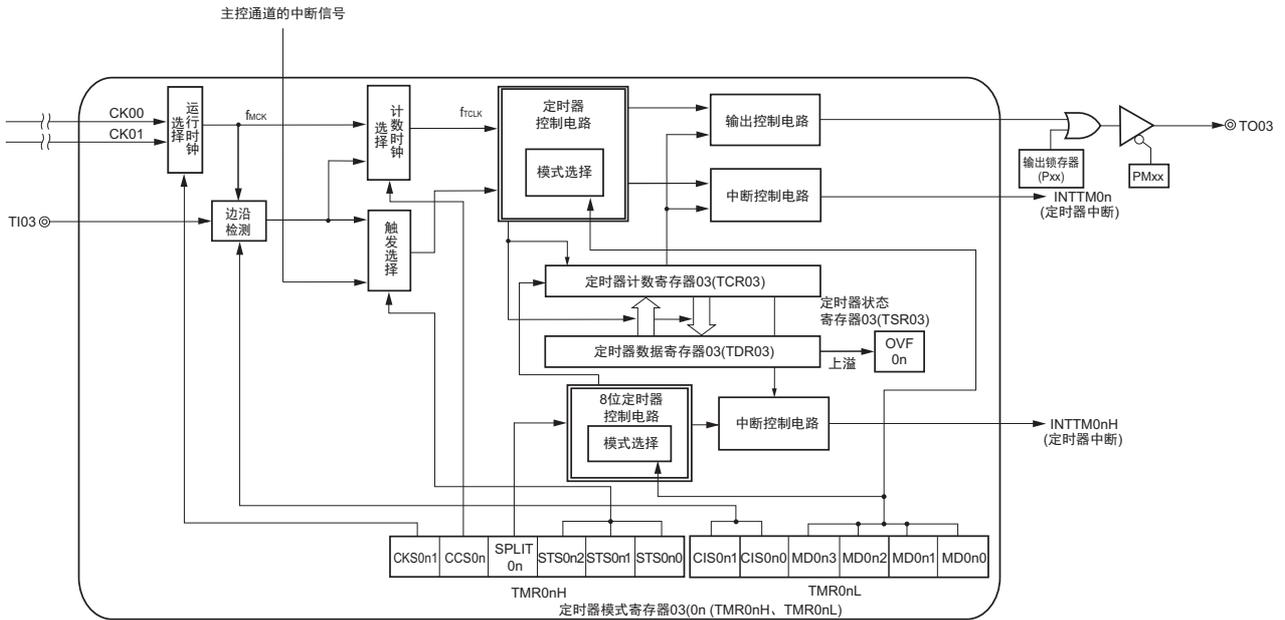
备注 n=0、2

(b) 通道 1



备注 n=1

(c) 通道 3 注



注 只限于 16 引脚产品。

备注 n=3

6.2.1 定时器计数寄存器 0n (TCR0n)

TCR0n 寄存器由 2 个对计数时钟 (f_{TCLK}) 进行计数的 8 位只读寄存器 (TCR0nH、TCR0nL) 构成。

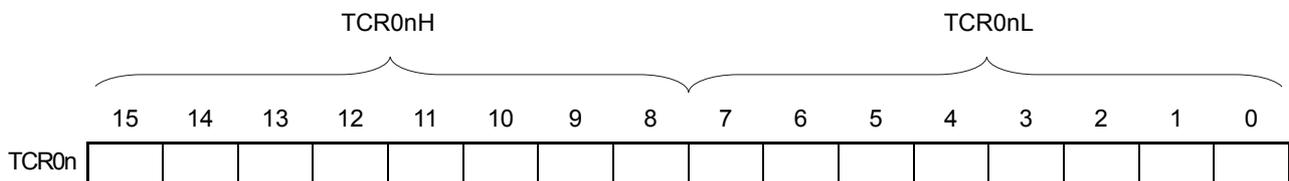
如果要读 TCR0n 寄存器, 必须连续存取 TCR0nH 和 TCR0nL 寄存器。

与计数时钟 (f_{TCLK}) 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 0n (TMR0n) 的 MD0n3 ~ MD0n0 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 0n (TMR0n)”)。

图 6-3 定时器计数寄存器 0n (TCR0n) 的格式 (n=0 ~ 3)

地址: F0180H (TCR00L)、F0181H (TCR00H) 复位后: FFH R
 F0182H (TCR01L)、F0183H (TCR01H)
 F0184H (TCR02L)、F0185H (TCR02H)
 F0186H (TCR03L)、F0187H (TCR03H)



备注 n: 通道号。n=0、1 (10 引脚产品)、0=0 ~ 3 (16 引脚产品)

必须按照 TCR0nL 寄存器 → TCR0nH 寄存器的顺序连续读 TCR0nH 和 TCR0nL 寄存器。在进行连续读操作的期间, 如果插入读 TCR0nL 寄存器等, 就无法进行正常的读操作。

注意 必须在执行 DI 指令后禁止中断的状态下, 进行 TCR0nH 寄存器和 TCR0nL 寄存器的连续读操作。

能通过读定时器计数寄存器 0n (TCR0n) 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 TAU0EN 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 只限于 16 引脚产品。

注意 1. 即使读 TCR0n 寄存器, 也不将计数值捕捉到定时器数据寄存器 0n (TDR0n)。

2. 在 8 位定时器模式 (SPLIT=1) 中使用通道 1 和通道 3 时, 禁止读 TCR01H 寄存器和 TDR01H 寄存器或者 TCR03H 寄存器和 TDR03H 寄存器。

如下所示, TCR0n 寄存器的读取值因运行模式和运行状态而不同。

表 6-3 各运行模式中的定时器计数寄存器 0n (TCR0n) 的读取值

运行模式	计数方式	定时器计数寄存器 0n (TCR0n) 的读取值 ^注			
		解除复位后更改运行模式时的值	计数暂停 (TT0n=1) 时的值	计数暂停 (TT0n=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	TDR0n 寄存器的捕捉值 +1

注 表示通道 n 从定时器运行停止状态 (TE0n=0) 变为计数允许状态 (TS0n=1) 时的 TCR0n 寄存器的读取值。将此值保持在 TCR0n 寄存器，直到开始计数为止。

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

注意 在 8 位定时器模式 (SPLIT=1) 中使用通道 1 和通道 3 时，禁止读 TCR01H 寄存器和 TDR01H 或者 TCR03H 寄存器和 TDR03H 寄存器。

6.2.2 定时器数据寄存器 0n (TDR0n)

TDR0n 寄存器由 2 个能进行捕捉功能和比较功能切换使用的 8 位寄存器 (TDR0nH、TDR0nL) 构成。通过定时器模式寄存器 0n (TMR0n) 的 MD0n3 ~ MD0n0 位来选择运行模式, 进行捕捉功能和比较功能的切换。

在用作比较寄存器时, 能随时改写 TDR0nH 寄存器和 TDR0nL 寄存器。

如果要存取 TDR0n 寄存器, 必须连续存取 TDR0nH 寄存器和 TDR0nL 寄存器。

另外, 在 8 位定时器模式中 (定时器模式寄存器 0n (TMR0n) 的 SPLIT0n 位为“1”), 将 TDR0n 寄存器的高 8 位作为 TDR0nH 寄存器, 低 8 位作为 TDR0nL 寄存器, 并且能以 8 位为单位进行改写。

在读写 TDR0nH 寄存器和 TDR0nL 寄存器时的注意事项如下所示。

- 16 位定时器模式的情况 (通道 0、2 或者通道 1、3 的 TMR0nH 寄存器的 bit3 (SPLIT0n) 为“0”)
 - 必须按照 TDR0nH 寄存器 → TDR0nL 寄存器的顺序连续写 TDR0nH 寄存器和 TDR0nL 寄存器。在 TDR0nL 寄存器的值被改写时, 更新 TDR0nH 寄存器和 TDR0nL 寄存器的值。
 - 必须按照 TDR0nL 寄存器 → TDR0nH 寄存器的顺序连续读 TDR0nH 寄存器和 TDR0nL 寄存器。在 TDR0nL 寄存器的值被读取时, 更新 TDR0nH 寄存器的值。
 - 在连续读操作或者连续写操作的期间, 如果插入写 TDR0nH 寄存器、读 TDR0nL 寄存器或者读 TCR0n 寄存器等, 就无法正常进行读写操作。
 - 必须在执行 DI 指令后禁止中断的状态下进行 TDR0nH 寄存器和 TDR0nL 寄存器的连续读写操作。
- 8 位定时器模式的情况 (通道 1、3 的 TMR0nH 寄存器的 bit3 (SPLIT0n) 为“1”)
 - 在 8 位定时器模式中, 能以 8 位为单位写 TDR0nH 寄存器和 TDR0nL 寄存器。
 - 必须按照 TDR0nL 寄存器 → TDR0nH 寄存器的顺序连续读 TDR0nH 寄存器。在 TDR0nL 寄存器的值被读取时, 更新 TDR0nH 寄存器的值。
 - 在连续读操作的期间, 如果插入读 TDR0nL 寄存器或者读 TCR0n 寄存器等, 就无法正常进行读操作。
 - 必须在执行 DI 指令后禁止中断的状态下进行 TDR0nH 寄存器和 TDR0nL 寄存器的连续读操作。

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

注意 在 8 位定时器模式 (SPLIT=1) 中使用通道 1 和通道 3 时, 禁止读 TCR01H 寄存器和 TDR01H 或者 TCR03H 寄存器和 TDR03H 寄存器。

图 6-4 定时器数据寄存器 0n (TDR0nH、TDR0nL) 的格式 (n=0、2)

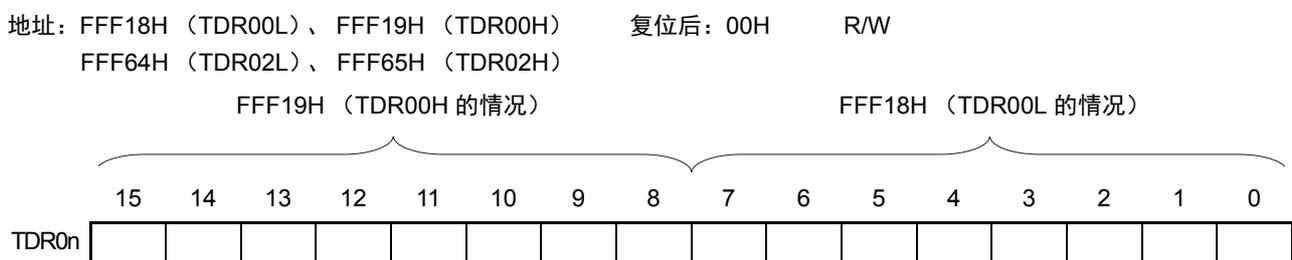
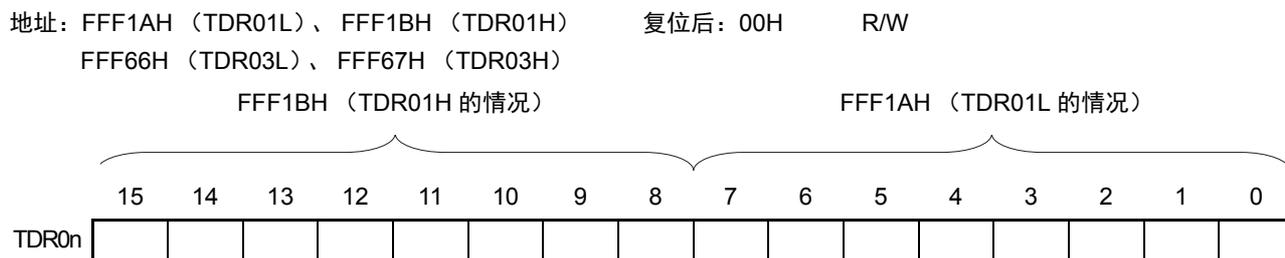


图 6-5 定时器数据寄存器 0n (TDR0n) 的格式 (n=1、3)



- (i) 定时器数据寄存器 0n (TDR0nH、TDR0nL) 用作比较寄存器的情况
 从 TDR0nH、TDR0nL 寄存器的设定值开始递减计数，当计数值变为“0000H”时，产生中断请求信号 (INTTM0n)。保持 TDR0n 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 TDR0n 寄存器也不进行捕捉运行。

- (ii) 定时器数据寄存器 0n (TDR0nH、TDR0nL) 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 0n (TCR0n) 的计数值捕捉到 TDR0nH 寄存器和 TDR0nL 寄存器。
 能选择 TI0n 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 0n (TMR0n) 来设定捕捉触发的选择。

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

6.3 定时器阵列单元的控制寄存器

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0 (TPS0)
- 定时器通道允许状态寄存器0 (TE0、TEH0)
- 定时器通道开始寄存器0 (TS0、TSH0)
- 定时器通道停止寄存器0 (TT0、TTH0)
- 定时器输出允许寄存器0 (TOE0)
- 定时器输出寄存器0 (TO0)
- 定时器输出电平寄存器0 (TOL0)
- 定时器输出模式寄存器0 (TOM0)
- 定时器模式寄存器0n (TMR0nH、TMR0nL)
- 定时器状态寄存器0n (TSR0n)
- 噪声滤波器允许寄存器1 (NFEN1)
- 输入切换控制寄存器 (ISC)
- 端口模式控制寄存器0 (PMC0)
- 端口模式寄存器0、4 (PM0、PM4)
- 端口寄存器0、4 (P0、P4)

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

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，降低功耗和噪声。

要使用定时器阵列单元时，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 6-6 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	TMKAEN 注	CMPEN 注	ADCEN	IICA0EN 注	0	SAU0EN	0	TAU0EN

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

注 只限于 16 引脚产品。

注意 1. 要设定定时器阵列单元时，必须在 TAU0EN 位为“1”的状态下，设定以下寄存器。当 TAU0EN 位为“0”时，定时器阵列单元的控制寄存器为初始值，并且忽视写操作（噪声滤波器允许寄存器 1 (NFEN1)、输入切换控制寄存器 (ISC)、端口模式寄存器 0、4 (PM0、PM4)、端口寄存器 0、4 (P0、P4、) 和端口模式控制寄存器 0 (PMC0) 除外）。

- 定时器计数寄存器 0n (TCR0nH、TCR0nL)
- 定时器数据寄存器 0n (TDR0nH、TDR0nL)
- 定时器时钟选择寄存器 0 (TPS0)
- 定时器通道允许状态寄存器 0 (TE0、TEH0)
- 定时器通道开始寄存器 0 (TS0、TSH0)
- 定时器通道停止寄存器 0 (TT0、TTH0)
- 定时器输出允许寄存器 0 (TOE0)
- 定时器输出寄存器 0 (TO0)
- 定时器输出电平寄存器 0 (TOL0)
- 定时器输出模式寄存器 0 (TOM0)
- 定时器模式寄存器 0n (TMR0nH、TMR0nL)
- 定时器状态寄存器 0n (TSR0n)

2. 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

6.3.2 定时器时钟选择寄存器 0 (TPS0)

TPS0 寄存器是选择从预分频器共同提供给各通道的 2 种运行时钟 (CK00、CK01) 的 8 位寄存器。只有在以下情况下才能改写定时器运行中的 TPS0 寄存器。

能改写 PRS000 ~ PRS003 位的情况 (n=0 ~ 3):

选择 CK00 作为运行时钟 (CKS0n1=0) 的通道全部处于停止状态 (TE0n=0)。

能改写 PRS010 ~ PRS013 位的情况 (n=0 ~ 3):

选择 CK01 作为运行时钟 (CKS0n1=1) 的通道全部处于停止状态 (TE0n=0)。

通过 8 位存储器操作指令设定 TPS0 寄存器。

在产生复位信号后, TPS0 寄存器的值变为“00H”。

图 6-7 定时器时钟选择寄存器 0 (TPS0) 的格式

地址: F01B6H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS0k3	PRS0k2	PRS0k1	PRS0k0	运行时钟 (CK0k) 的选择注 (k=0、1)	运行时钟 (CK0k) 的选择注 (k=0、1)				
					$f_{CLK}=1.25MHz$	$f_{CLK}=2.5MHz$	$f_{CLK}=5MHz$	$f_{CLK}=10MHz$	$f_{CLK}=20MHz$
0	0	0	0	f_{CLK}	1.25MHz	2.5MHz	5MHz	10MHz	20MHz
0	0	0	1	$f_{CLK}/2$	625kHz	1.25kHz	2.5MHz	5MHz	10MHz
0	0	1	0	$f_{CLK}/2^2$	313kHz	625kHz	1.25MHz	2.5MHz	5MHz
0	0	1	1	$f_{CLK}/2^3$	156kHz	313kHz	625kHz	1.25MHz	2.5MHz
0	1	0	0	$f_{CLK}/2^4$	78.1kHz	156kHz	313kHz	625kHz	1.25MHz
0	1	0	1	$f_{CLK}/2^5$	39.1kHz	78.1kHz	156kHz	313kHz	625kHz
0	1	1	0	$f_{CLK}/2^6$	19.5kHz	39.1kHz	78.1kHz	156kHz	313kHz
0	1	1	1	$f_{CLK}/2^7$	9.77kHz	19.5kHz	39.1kHz	78.1kHz	156kHz
1	0	0	0	$f_{CLK}/2^8$	4.88kHz	9.77kHz	19.5kHz	39.1kHz	78.1kHz
1	0	0	1	$f_{CLK}/2^9$	2.44kHz	4.88kHz	9.77kHz	19.5kHz	39.1kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22kHz	2.44kHz	4.88kHz	9.77kHz	19.5kHz
1	0	1	1	$f_{CLK}/2^{11}$	610Hz	1.22kHz	2.44kHz	4.88kHz	9.77kHz
1	1	0	0	$f_{CLK}/2^{12}$	305Hz	610Hz	1.22kHz	2.44kHz	4.88kHz
1	1	0	1	$f_{CLK}/2^{13}$	453Hz	305Hz	610Hz	1.22kHz	2.44kHz
1	1	1	0	$f_{CLK}/2^{14}$	76.3Hz	153Hz	305Hz	610Hz	1.22kHz
1	1	1	1	$f_{CLK}/2^{15}$	38.1Hz	76.3Hz	153Hz	305Hz	610Hz

注 在更改 f_{CLK} (更改系统时钟控制寄存器 (CKC) 的值) 时, 必须停止定时器阵列单元 (TT0=0FH、TTH0=0AH)。

注意 如果选择 f_{CLK} (无分频) 作为运行时钟 (CK0k) 并且将 TDR0nH 置“00H”、TDR0nL 置“00H” (n=0 ~ 3), 就不能使用来自定时器阵列单元的中断请求信号 (INTTM0n)。

备注 1. f_{CLK} : CPU/ 外围硬件的时钟频率

2. TPS0 寄存器选择的时钟波形从上升沿开始只有 1 个 f_{CLK} 周期为高电平。详细内容请参照“6.5.1 计数时钟 (f_{TCLK})”。

6.3.3 定时器模式寄存器 0n (TMR0n)

TMR0n 寄存器由 2 个设定通道 n 的运行模式的 8 位寄存器 (TMR0nH、TMR0nL) 构成。进行运行时钟 (f_{MCK}) 的选择、计数时钟 (f_{TCLK}) 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 (TE0n=1) 改写 TMR0nH 寄存器和 TMR0nL 寄存器。

通过 8 位存储器操作指令设定 TMR0nH 寄存器和 TMR0nL 寄存器。

在产生复位信号后, TMR0nH 寄存器和 TMR0nL 寄存器的值变为“00H”。

注意 TMR0nH 寄存器的 bit3 因通道而不同。

TMR02H:	MASTER02 位
TMR01H、TMR03H:	SPLIT0n 位 (n=1、3)
TMR00H:	固定为“0”。

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (1/3)

地址: F0190H (TMR00L)、F0191H (TMR00H) 复位后: 00H R/W
 F0192H (TMR01L)、F0193H (TMR01H)
 F0194H (TMR02L)、F0195H (TMR02H)
 F0196H (TMR03L)、F0197H (TMR03H)

符号	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000

符号	7	6	5	4	3	2	1	0
TMR02H	CKS021	0	0	CCS02	MASTER02	STS022	STS021	STS020

符号	7	6	5	4	3	2	1	0
TMR0nH (n=1, 3)	CKS0n1	0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0

符号	7	6	5	4	3	2	1	0
TMR0nL (n=0~3)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CKS0n1	通道 n 的运行时钟 (f_{MCK}) 选择
0	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK00
1	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK01
运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定 CCS0n 位来产生采样时钟和计数时钟 (f_{TCLK})。	

CCS0n	通道 n 的计数时钟 (f_{TCLK}) 选择
0	CKS0n1 位指定的运行时钟 (f_{MCK})
1	TI0n 引脚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。	

注意 1. 必须将以下位置“0”。

TMR00H 寄存器: bit3、5、6

TMR01H ~ TMR03H 寄存器: bit5 和 bit6

TMR00L ~ TMR03L 寄存器: bit4 和 bit5

- 要更改选择为 f_{CLK} 时钟 (更改系统时钟控制寄存器 (CKC) 的值) 时, 即使选择了 CKS0n1 位指定的运行时钟 (f_{MCK}) 或者 TI0n 引脚的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 (TT0=0FH、TTH0=0AH)。

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (2/3)

符号	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000
符号	7	6	5	4	3	2	1	0
TMR02H	CKS021	0	0	CCS02	MASTER02	STS022	STS021	STS020
符号	7	6	5	4	3	2	1	0
TMR0nH (n=1, 3)	CKS0n1	0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0

(TMR02H 的 bit3)

MASTER02	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。
只有通道 0、2 能用作主控通道。 将通道 2 用作主控通道时, 给 TMR02H 的 bit3 (MASTER02) 设定“1”。 因为通道 0 为最高位通道, 所以与 TMR00H 的 bit3 注的设定无关, 用作主控通道。 对于用作独立通道运行功能的通道, 将 MASTER02 位置“0”。	

(TMR01H、TMR03H 的 bit3)

SPLIT0n	通道 1 和通道 3 的 8 位定时器 / 16 位定时器的运行选择 (n=1, 3)
0	用作 16 位定时器。
1	用作 8 位定时器。

STS 0n2	STS 0n1	STS 0n0	通道 n 的开始触发和捕捉触发的设定 (n=0, 1 (10 引脚产品)、n=0, 3 (16 引脚产品))
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 TI0n 引脚输入的有效边沿用于开始触发和捕捉触发。
0	1	0	将 TI0n 引脚输入的双边沿分别用于开始触发和捕捉触发。
1	0	0	单触发脉冲输出、PWM 输出功能、多重 PWM 输出功能的从属通道的情况: 将主控通道的中断请求信号 (INTTM0n) 用作开始触发。
1	1	0	双输入式单触发脉冲输出的从属通道的情况: 将主控通道的中断请求信号 (INTTM0n) 用作开始触发。 将从属通道的 TI03 引脚输入的有效边沿用作结束触发。
上述以外			禁止设定。

注 TMR00H 寄存器的 bit3 是只读位, 固定为“0”, 忽视写操作。

注意 必须将以下位置“0”。

TMR00H 寄存器: bit3、5、6

TMR01H ~ TMR03H 寄存器: bit5 和 bit6

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (3/3)

符号	7	6	5	4	3	2	1	0
TMR0nL (n=0~3)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CIS0n1	CIS0n0	Ti0n 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 STS0n2 ~ STS0n0 位不为“010B”并且使用双边沿指定时, 必须将 CIS0n1 ~ CIS0n0 位置“10B”。

MD0n3	MD0n2	MD0n1	通道 n 的运行模式设定	对应功能	TCR 的计数运行
0	0	0	间隔定时器模式	间隔定时器 / 方波输出 / 分频器功能 / PWM 输出 (主控)	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量 / 双输入式单触发脉冲输出 (从属)	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 / 双输入式单触发脉冲输出 (主控) / 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 MD0n0 位而变 (参照下表)。

运行模式 (通过 MD0n3 ~ MD0n1 位设定)	MD0n0	开始计数和中断的设定
<ul style="list-style-type: none"> 间隔定时器模式^{注2} (0、0、0) 捕捉模式 (0、1、0) 	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断 (定时器的输出也发生变化)。
<ul style="list-style-type: none"> 事件计数器模式^{注2} (0、1、1) 	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
<ul style="list-style-type: none"> 单次计数模式^{注1} (1、0、0) 	0	计数运行中的开始触发无效。此时不产生定时器中断。
	1	计数运行中的开始触发有效 ^{注2} 。此时不产生定时器中断。
<ul style="list-style-type: none"> 捕捉 & 单次计数模式 (1、1、0) 	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。计数运行中的开始触发无效。此时不产生定时器中断。
上述以外		禁止设定。

- 注
1. 在单次计数模式中，不控制开始计数时的中断请求信号（INTTM0n）和 TO0n 输出。
 2. 如果在运行中产生开始触发（TS0n=1），就对计数器进行初始化并且重新开始计数（不产生中断请求信号（INTTM0n））。

注意 必须将 TMR00L ~ TMR03L 寄存器的 bit4 和 bit5 置“0”。

备注 n: 通道号。n=0、1（10 引脚产品）、n=0 ~ 3（16 引脚产品）

6.3.4 定时器状态寄存器 0n (TSR0n)

TSR0n 寄存器是表示通道 n 计数器的上溢状态的寄存器。

TSR0n 寄存器只在捕捉模式 (MD0n3 ~ MD0n1=010B) 和捕捉 & 单次计数模式 (MD0n3 ~ MD0n1=110B) 中有效。在其他模式中不能置位。有关各运行模式中的 OVF 位的变化和设定 / 清除条件, 请参照表 6-4。

通过 8 位存储器操作指令读 TSR0n 寄存器的低 8 位。

在产生复位信号后, TSR0n 寄存器的值变为“00H”。

图 6-9 定时器状态寄存器 0n (TSR0n) 的格式

地址: F01A0H (TSR00)、F01A2H (TSR01) 复位后: 00H R
F01A4H (TSR02)、F01A6H (TSR03)

符号	7	6	5	4	3	2	1	0
TSR0n	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 n: 通道号。n=0、1 (10 引脚产品)、n=0 ~ 3 (16 引脚产品)

表 6-4 各运行模式中的 OVF 位的变化和设定 / 清除条件

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

备注 在计数器发生上溢后, OVF 位不立即发生变化, 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 0 (TE0、TEH0 (8 位模式))

TE0、TEH0 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。

TE0、TEH0 寄存器的各位对应定时器通道开始寄存器 0 (TS0、TSH0) 和定时器通道停止寄存器 0 (TT0、TTH0) 的各位。如果将 TS0、TSH0 寄存器的各位置“1”，TE0、TEH0 寄存器的对应位就被置“1”。如果将 TT0、TTH0 寄存器的各位置“1”，就将其对应位清“0”。

通过 1 位或者 8 位存储器操作指令读 TE0 寄存器和 TEH0 寄存器。

在产生复位信号后，TE0、TEH0 寄存器的值变为“00H”。

图 6-10 定时器通道允许状态寄存器 0 (TE0) 的格式

地址: F01B0H	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TE03注	TE02注	TE01	TE00

TE0n	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

表示 16 位定时器运行的允许或者停止状态。
在通道 1 和通道 3 为 8 位定时器模式时，通过 TE01 和 TE03 寄存器表示低 8 位定时器的运行允许或者停止状态。

注 只限于 16 引脚产品。

备注 n: 通道号。n=0、1 (10 引脚产品)、n=0~3 (16 引脚产品)

图 6-11 定时器通道允许状态寄存器 0 (TEH0) 的格式

地址: F01B1H	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
TEH0	0	0	0	0	TEH03注	0	TEH01	0

TEH0n	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，表示高 8 位定时器的运行允许或者停止状态。

注 只限于 16 引脚产品。

6.3.6 定时器通道开始寄存器 0 (TS0、TSH0 (8 位模式))

TS0、TSH0 寄存器是对定时器计数寄存器 0n (TCR0n) 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 TS0 寄存器和 TSH0 寄存器的各位置“1”，定时器通道允许状态寄存器 0 (TE0、TEH0) 的对应位就被置“1”。因为 TS0 寄存器和 TSH0 寄存器的 TS0n 位和 TSH0n 位是触发位，所以如果变为运行允许状态 (TE0n=1)，就立即将 TS0n 位和 TSH0n 位清“0”。

通过 1 位或者 8 位存储器操作指令设定 TS0 寄存器和 TSH0 寄存器。

在产生复位信号后，TS0 寄存器和 TSH0 寄存器的值变为“00H”。

图 6-12 定时器通道开始寄存器 0 (TS0) 的格式

地址: F01B2H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TS03 注	TS02 注	TS01	TS00

TS0n	通道 n 的运行允许 (开始) 触发 (n=0 ~ 3)
0	没有触发运行。
1	将 TE0n 位置“1”，进入计数允许状态。 计数允许状态下的 TCR0n 寄存器的计数开始运行因运行模式而不同 (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，TS01 和 TS03 为低 8 位定时器的运行允许 (开始) 触发。

图 6-13 定时器通道开始寄存器 0 (TSH0) 的格式

地址: F01B3H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TSH0	0	0	0	0	TSH03 注	0	TSH01	0

TSH0n	通道 n 的运行允许 (开始) 触发 (n=1、3)
0	没有触发运行。
1	将 TEH0n 位置“1”，进入计数允许状态。 计数允许状态下的 TCR0n 寄存器的计数开始运行时间为间隔定时器模式 (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，为高 8 位定时器的运行允许 (开始) 触发。

注 只限于 16 引脚产品。

注意 1. 必须将以下位置“0”。

TS0: 10 位引脚产品的 bit2 ~ 7, 16 引脚产品的 bit4 ~ 7

TSH0: 10 位引脚产品的 bit0、bit2 ~ 7, 16 引脚产品的 bit0、bit2、bit4 ~ 7

2. 在从不使用 TI0n 引脚输入的功能切换到使用 TI0n 引脚输入的功能时，从设定定时器模式寄存器 0n (TMR0n) 到将 TS0n 位置“1”为止，需要以下期间的等待：

TI0n 引脚噪声滤波器有效时 (TNFEN=1)：4 个运行时钟 (f_{MCK})

TI0n 引脚噪声滤波器无效时 (TNFEN=0)：2 个运行时钟 (f_{MCK})

备注 TS0 寄存器和 TSH0 寄存器的读取值总是为“0”。

6.3.7 定时器通道停止寄存器 0 (TT0、TTH0 (8 位模式))

TT0、TTH0 寄存器是对各通道设定定时器计数寄存器 0n (TCR0n) 的计数停止的触发寄存器。

如果将 TT0 寄存器和 TTH0 寄存器的各位置“1”，定时器通道允许状态寄存器 0 (TE0、TEH0) 的对应位就被清“0”。因为 TT0n 位和 TTH0n 位是触发位，所以如果变为运行停止状态 (TE0n、TEH0n=0)，TT0n 位和 TTH0n 位就立即被清“0”。

通过 1 位或者 8 位存储器操作指令设定 TT0 寄存器和 TTH0 寄存器。

在产生复位信号后，TT0 寄存器和 TTH0 寄存器的值变为“00H”。

图 6-14 定时器通道停止寄存器 0 (TT0) 的格式

地址: F01B4H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TT03注	TT02注	TT01	TT00

TT0n	通道 n 的运行停止触发 (n=0 ~ 3)
0	没有触发运行。
1	将 TE0n 位清“0”，进入计数运行停止状态。 在通道 1 和通道 3 为 8 位定时器模式时，TT01 和 TT03 为低 8 位定时器的运行停止触发。

图 6-15 定时器通道停止寄存器 0 (TTH0) 的格式

地址: F01B5H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TTH0	0	0	0	0	TTH03注	0	TTH01	0

TTH0n	通道 n 的运行停止触发 (n=1、3)
0	没有触发运行。
1	将 TEH0n 位清“0”，进入计数运行停止状态。 在通道 1 和通道 3 为 8 位定时器模式时，为高 8 位定时器的运行停止触发。

注 只限于 16 引脚产品。

注意 必须将以下位置“0”。

TT0: 10 引脚产品的 bit2 ~ 7, 16 引脚产品的 bit4 ~ 7

TTH0: 10 位引脚产品的 bit0、bit2 ~ 7, 16 引脚产品的 bit0、bit2、bit4 ~ 7

备注 TT0 寄存器和 TTH0 寄存器的读取值总是“0”。

6.3.8 定时器输出允许寄存器 0 (TOE0)

TOE0 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 0 (TO0) 的 TO0n 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 (TO0n) 输出。

通过 1 位或者 8 位存储器操作指令设定 TOE0 寄存器的低 8 位。

在产生复位信号后，TOE0 寄存器的值变为“00H”。

图 6-16 定时器输出允许寄存器 0 (TOE0) 的格式

地址: F01BAH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	TOE03 注	TOE02 注	TOE01	TOE00

TOE0n	通道 n 的定时器输出允许 / 禁止
0	禁止定时器的输出。 定时器的运行不反映到 TO0n 位，固定输出。 能写 TO0n 位，并且从 TO0n 引脚输出给 TO0n 位设定的电平。
1	允许定时器的输出。 定时器的运行反映到 TO0n 位，产生输出波形。 忽视 TO0n 位的写操作。

注 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit2 ~ 7、16 引脚产品的 bit4 ~ 7 置“0”。

备注 n: 通道号。n=0、1 (10 引脚产品)，n=0 ~ 3 (16 引脚产品)

6.3.9 定时器输出寄存器 0 (TO0)

TO0 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的定时器的输出引脚 (TO0n) 输出。

只有在禁止定时器输出 (TOE0n=0) 时才能通过软件改写此寄存器的 TO0n 位。当允许定时器输出时 (TOE0n=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 TO0n 引脚用作端口功能时，必须将相应的 TO0n 位置“0”。

通过 8 位存储器操作指令设定 TO0 寄存器。

在产生复位信号后，TO0 寄存器的值变为“00H”。

图 6-17 定时器输出寄存器 0 (TO0) 的格式

地址: F01B8H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TO0	0	0	0	0	TO03注	TO02注	TO01	TO00

TO0n	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 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit2 ~ 7、16 引脚产品的 bit4 ~ 7 置“0”。

备注 n: 通道号。n=0、1 (10 引脚产品)，n=0 ~ 3 (16 引脚产品)

6.3.10 定时器输出电平寄存器 0 (TOL0)

TOL0 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 (TOE0n=1) 并且为多通道联动运行功能 (TOM0n=1) 时, 在定时器输出信号的置位和复位时序, 反映此寄存器进行的各通道 n 的反相设定。在主导通道输出模式 (TOM0n=0) 中, 此寄存器的设定无效。

通过 8 位存储器操作指令设定 TOL0 寄存器。

在产生复位信号后, TOL0 寄存器的值变为“00H”。

图 6-18 定时器输出电平寄存器 0 (TOL0) 的格式

地址: F01BCH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	TOL03注	TOL02注	TOL01	0

TOL0n	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 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit0、bit2 ~ 7、16 引脚产品的 bit0、bit4 ~ 7 置“0”。

备注 1. 是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 TOL0 寄存器后立即反相。

2. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

6.3.11 定时器输出模式寄存器 0 (TOM0)

TOM0 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（单触发脉冲输出、双输入式单触发脉冲输出注、多重 PWM 输出注）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（TOE0n=1）时，在定时器输出信号的置位和复位时序，反映此寄存器进行的各通道 n 的设定。

通过 8 位存储器操作指令设定 TOM0 寄存器。

在产生复位信号后，TOM0 寄存器的值变为“00H”。

图 6-19 定时器输出模式寄存器 0 (TOM0) 的格式

地址: F01BEH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	TOM03注	TOM02注	TOM01	0

TOM0n	通道 n 的定时器输出模式的控制
0	用作独立通道运行功能 (通过中断请求信号 (INTTM0n) 进行交替输出)
1	从属通道输出模式 (通过主控通道的中断请求信号 (INTTM00、INTTM02) 将输出置位，并且通过从属通道的中断请求信号 (INTTM0p) 对输出进行复位)

注 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit0、bit2 ~ 7、16 引脚产品的 bit0、bit4 ~ 7 置“0”。

备注 n: 主控通道号。n=0 (10 引脚产品)，n=0、2 (16 引脚产品)

p: 从属通道号。p=1 (10 引脚产品)， $n < p \leq 3$ (16 引脚产品)

(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”。)

6.3.12 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 (TI0n) 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, 通过对象通道的运行时钟 (f_{MCK}) 进行同步后, 执行 2 个时钟的一致检测。当噪声滤波器无效时, 只通过对象通道的运行时钟 (f_{MCK}) 进行同步。有关定时器输入 (TI0n) 运行, 请参照“6.5.1 (2) 选择 TI0n 引脚输入信号的有效边沿的情况 (CCS0n=1)”、“6.5.2 计数器的开始时序”和“6.7 定时器输入 (TI0n) 的控制”。

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”。

图 6-20 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注	TNFEN02注	TNFEN01	TNFEN00

TNFEN0n	TI0n 引脚输入信号噪声滤波器的使用与否 (n=0 ~ 3)
0	噪声滤波器 OFF
1	噪声滤波器 ON

注 只限于 16 引脚产品。

注意 TNFEN01 位的噪声滤波器根据输入切换控制寄存器 (ISC) 的 ISC1 位的设定如下切换适合的输入引脚。

- ISC1=0 时: 对 TI01 引脚的输入信号选择是否使用噪声滤波器。
- ISC1=1 时: 对 RxD0 引脚的输入信号选择是否使用噪声滤波器。

6.3.13 输入切换控制寄存器 (ISC)

ISC 寄存器将通道 1 与串行阵列单元联合以实现 UART0 的波特率校正。

如果将 ISC1 位置“1”，串行数据输入 (RxD0) 引脚的输入信号就被作为定时器输入 (TI01) 选择。

能使用定时器阵列单元的输入脉冲间隔测量模式，将开始位的输入边沿信号作为触发，测量通信方的波特率 (传送速率) 宽度。

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 6-21 输入切换控制寄存器 (ISC)

地址: F0073H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	定时器阵列单元的通道 1 的输入切换
0	将 TI01 引脚的输入信号用作定时器输入 (通常运行)
1	将 RxD0 引脚的输入信号用作定时器输入 (唤醒信号的检测和校正波特率的脉宽的测量)

ISC0	外部中断 (INTP0) 的输入切换
0	将 INTP0 引脚的输入信号用作外部中断输入 (通常运算)
1	将 RxD0 引脚的输入信号用作外部中断输入 (唤醒信号的检测)

注意 必须将 bit2 ~ 7 置“0”。

6.3.14 定时器输入 / 输出引脚的端口功能控制寄存器

在使用定时器阵列单元时，必须设定对象通道的复用端口功能的控制寄存器（端口模式寄存器（PMxx）、端口寄存器（Pxx）、端口模式控制寄存器（PMCxx））。详细内容请参照“4.3.1 端口模式寄存器 0、4（PM0、PM4）”、“4.3.2 端口寄存器 0、4、12、13（P0、P4、P12、P13）”和“4.3.5 端口模式控制寄存器 0（PMC0）”。

用作定时器输入 / 输出时设定例子，请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

在将定时器输出引脚的复用端口（P04/ANI3/TI01/TO01/KR5 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（PMxx）的位、端口寄存器（Pxx）的位和端口模式控制寄存器（PMCxx）的位置“0”。

例 将 P04/ANI3/TI01/TO01/KR5 用作定时器输出的情况

将端口模式控制寄存器 0 的 PMC04 位置“0”。

将端口模式寄存器 0 的 PM04 位置“0”。

将端口寄存器 0 的 P04 位置“0”。

在将定时器输入引脚的复用端口（P04/ANI3/TI01/TO01/KR5 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（PMxx）的位置“1”并且将端口模式控制寄存器（PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

例 将 P04/ANI3/TI01/TO01/KR5 用作定时器输入的情况

将端口模式控制寄存器 0 的 PMC04 位置“0”。

将端口模式寄存器 0 的 PM04 位置“1”。

将端口寄存器 0 的 P04 位置“0”或者“1”。

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。

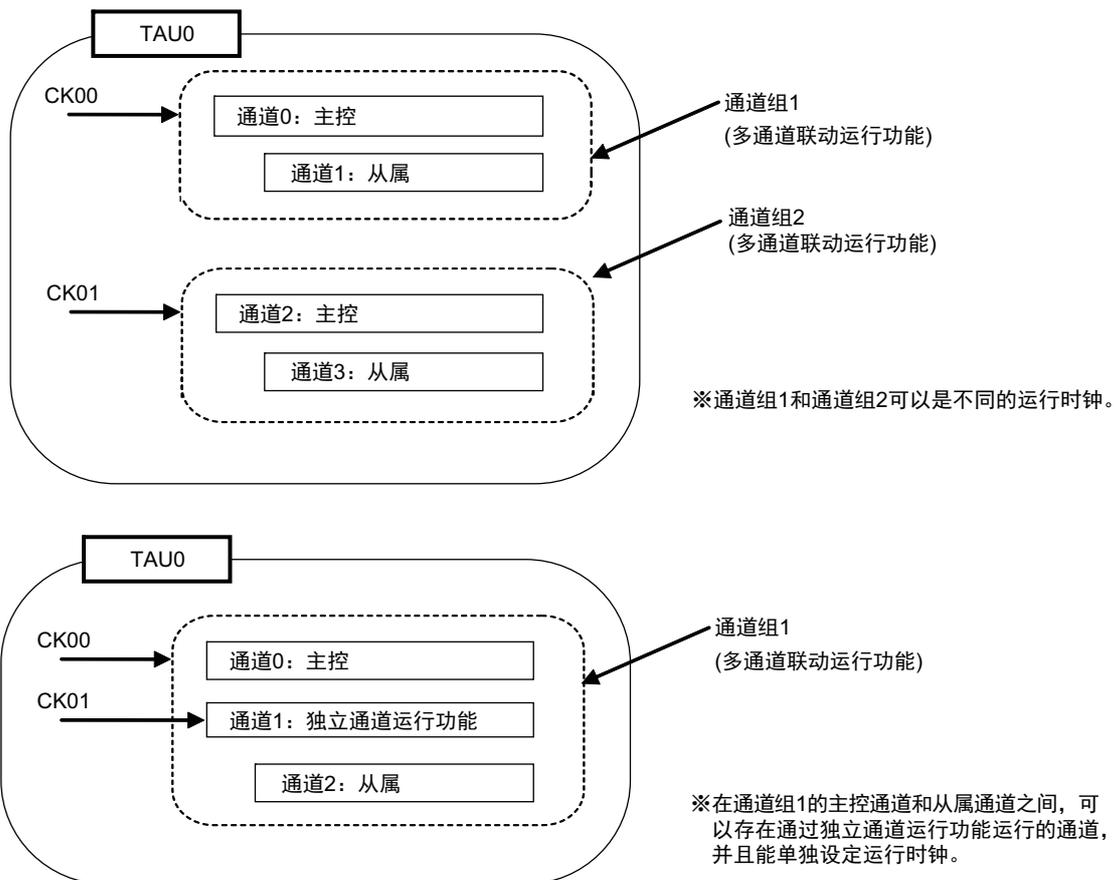
- (1) 只能将偶数通道（通道0、通道2）设定为主控通道。
- (2) 通道0以外的任何通道都能设定为从属通道注。
- (3) 只能将主控通道的低位通道设定为从属通道。
例 在将通道2设定为主控通道时，能将通道3设定为从属通道。
- (4) 能对1个主控通道设定多个从属通道。
- (5) 当使用多个主控通道时，不能设定跨越主控通道的从属通道。
例 在将通道0和通道2设定为主控通道时，只能将通道1设定为主控通道0的从属通道，而不能将通道3设定为主控通道0的从属通道。
- (6) 和主控通道联动的从属通道需要设定与主控通道相同的运行时钟。和主控通道联动的从属通道的CKS0n1位（定时器模式寄存器0nH（TMR0nH）的bit7）的值需要是与主控通道相同的设定值。
- (7) 主控通道能将中断请求信号（INTTM0n）、开始软件触发和计数时钟（ f_{TCLK} ）传给低位通道。
- (8) 从属通道能将主控通道的中断请求信号（INTTM0n）、开始软件触发和计数时钟（ f_{TCLK} ）用作源时钟，但是不能将自己的INTTM0n、开始软件触发和计数时钟（ f_{TCLK} ）传给低位通道。
- (9) 主控通道不能将其他高位主控通道的INTTM0n、开始软件触发和计数时钟（ f_{TCLK} ）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（TS0n）。
- (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TS0位的设定。不能只使用从属通道的TS0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（TT0n）。

注 通道1和通道3为8位定时器模式时，能选择低8位作为联动运行功能的从属通道。此时，通道1和通道3的高8位能用作间隔定时器。

备注 n：通道号。n=0、1（10引脚产品），n=0～3（16引脚产品）

多通道联动运行功能的基本规则是必须适用于通道组（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）。当设定了至少2个不联动的通道组时，在通道组之间则不适用上述基本规则。

例



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 0nH（TMR0nH）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 位定时器输出中断请求信号（INTTM01H、INTTM03H）（和 MD0n0 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 TMR0nH 寄存器的 CKS0n1 位的设定。
- (6) 对于高 8 位定时器，通过操作 TSH0n 位来开始通道的运行，并且通过操作 TTH0n 位来停止通道的运行。能通过 TEH0n 位确认通道的状态。
- (7) 低 8 位定时器的运行取决于 TMR0nH 寄存器和 TMR0nL 寄存器的设定，低 8 位定时器支持一下定时器功能：
 - 间隔定时器
 - 方波输出
 - 外部事件计数器
 - 延迟计数器
 - PWM 输出功能
 - 多重 PWM 输出功能（只限于 16 引脚产品）
- (8) 对于低 8 位定时器，通过操作 TS0n 位来开始通道的运行，并且通过操作 TT0n 位来停止通道的运行。能通过 TE0n 位确认通道的状态。
- (9) 在 16 位定时器运行时，TSH0n/TTH0n 位的操作无效。通过操作 TS0n 位和 TT0n 位来控制通道 n 的运行。TEH0n 位不变。

备注 n: 通道号 (n=1、3)

注意 在 8 位定时器模式（SPLIT=1）中使用通道 1 和通道 3 时，禁止读 TCR01H 寄存器和 TDR01H 寄存器或者 TCR03H 寄存器和 TDR03H 寄存器。

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 0n (TMR0n) 的 CCS0n 位选择以下任意一个时钟:

- CKS0n1 位指定的运行时钟 (f_{MCK})
- TI0n 引脚的输入信号的有效边沿

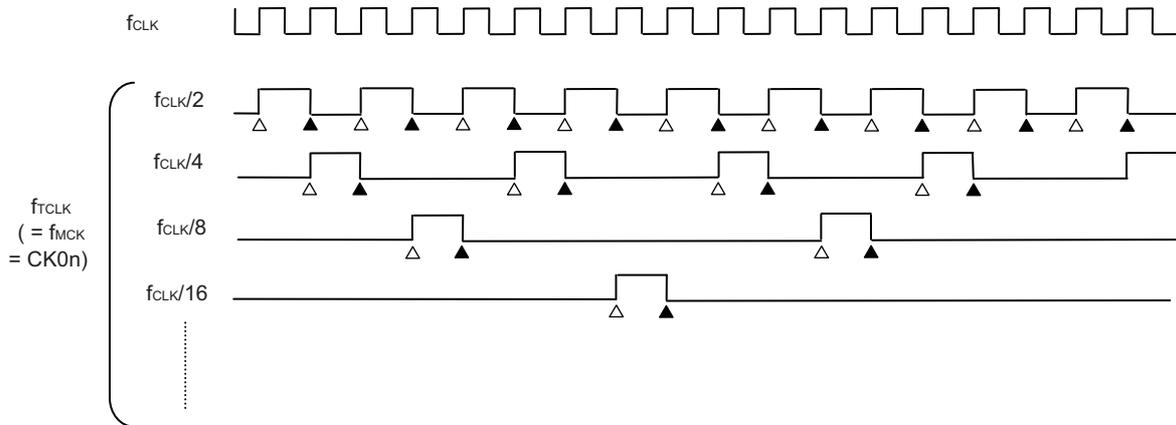
定时器阵列单元被设计为与 f_{CLK} 同步运行, 因此计数时钟 (f_{TCLK}) 的时序如下。

(1) 选择 CKS0n1 位指定的运行时钟 (f_{MCK}) 的情况 (CCS0n=0)

根据定时器时钟选择寄存器 0 (TPS0) 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, TPS0 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 0n (TCR0n) 从计数时钟 (f_{TCLK}) 的上升沿开始延迟 1 个 f_{TCLK} 时钟后进行计数, 出于方便, 将其称为“在计数时钟 (f_{TCLK}) 的上升沿进行计数”。

图 6-22 f_{CLK} 和计数时钟 (f_{TCLK}) 的时序 (CCS0n=0 的情况)



备注 1. Δ : 计数时钟 (f_{TCLK}) 的上升沿

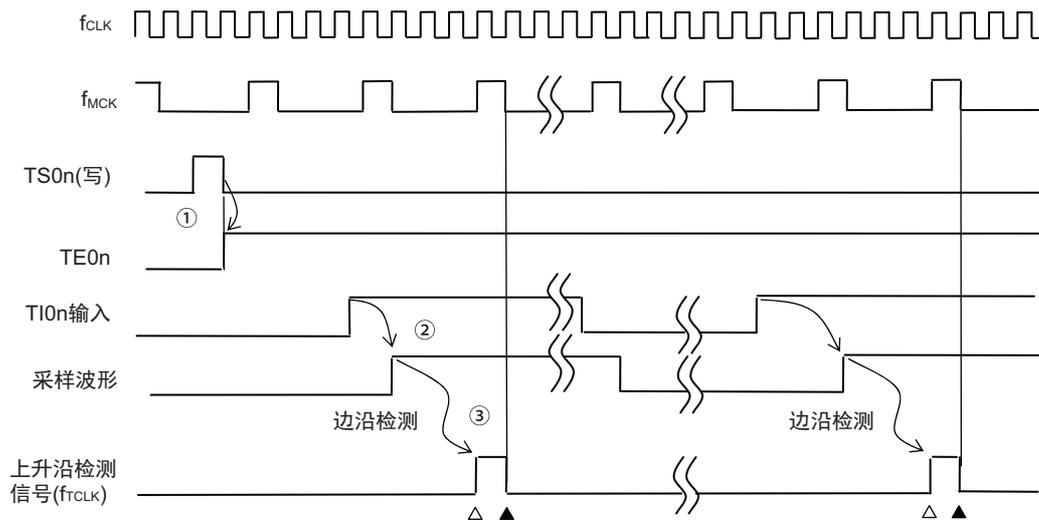
▲: 同步、计数器的递增 / 递减

2. f_{CLK} : CPU/ 外围硬件的时钟

(2) 选择 TI0n 引脚输入信号的有效边沿的情况 (CCS0n=1)

计数时钟 (f_{TCLK}) 是检测 TI0n 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 TI0n 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 0n (TCR0n) 从计数时钟 (f_{TCLK}) 的上升沿开始延迟 1 个 f_{CLK} 时钟后进行计数, 出于方便, 将其称为“在 TI0n 引脚输入信号的有效边沿进行计数”。

图 6-23 计数时钟 (f_{TCLK}) 的时序 (CCS0n=1, 未使用噪声滤波器的情况)

- ① 通过将 TS0n 位置位来开始定时器运行, 并且等待 TI0n 输入的有效边沿。
- ② 通过 f_{MCK} 对 TI0n 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 (f_{TCLK}))。

备注 1. Δ : 计数时钟 (f_{TCLK}) 的上升沿

\blacktriangle : 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 TI0n 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 0 (TS0) 的 TS0n 位置位, 定时器计数寄存器 0n (TCR0n) 进入运行允许状态。从计数运行允许状态到定时器计数寄存器 0n (TCR0n) 开始计数为止的运行如表 6-5 所示。

表 6-5 从计数运行允许状态到定时器计数寄存器 0n (TCR0n) 开始计数为止的运行

定时器的运行模式	将 TS0n 位置“1”后的运行
• 间隔定时器模式	从检测到开始触发 (TS0n=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 TDR0n 寄存器的值装入 TCR0n 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 TS0n 位写“1”, 将 TDR0n 寄存器的值装入 TCR0n 寄存器。 如果检测到 TI0n 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 (TS0n=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 TCR0n 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 (TE0n=0) 的状态下给 TS0n 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 TDR0n 寄存器的值装入 TCR0n 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 (TE0n=0) 的状态下给 TS0n 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 TCR0n 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)”)。

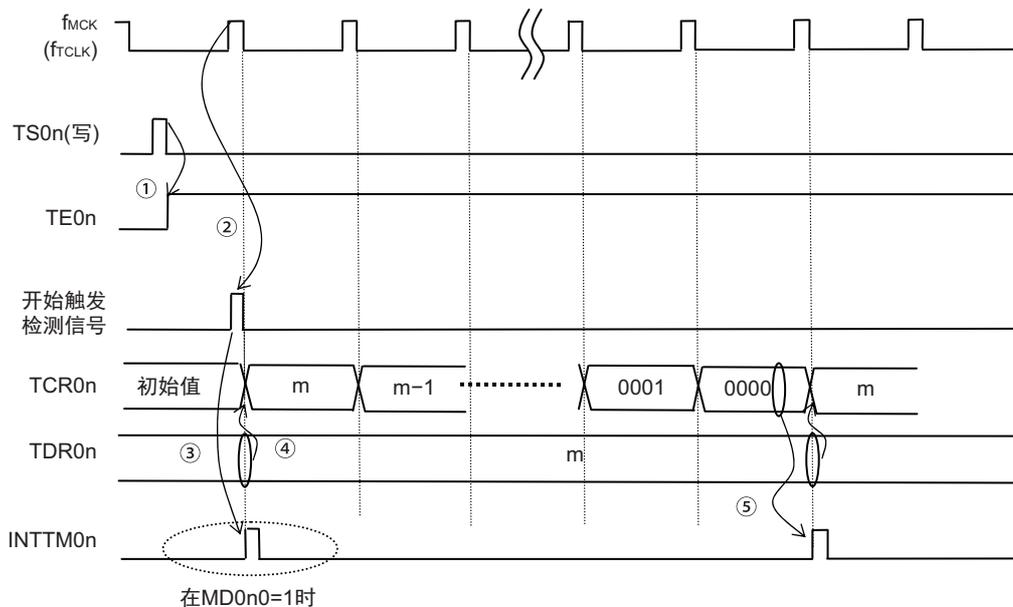
6.5.3 计数器的运行

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 TS0n 位写“1”，进入运行允许状态（TE0n=1）。定时器计数寄存器 0n（TCR0n）保持初始值，直到产生计数时钟（ f_{TCLK} ）为止。
- ② 通过允许运行后的第 1 个计数时钟产生开始触发信号。
- ③ 当 MD0n0 位为“1”时，通过开始触发信号产生 INTTM0n。
- 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 0n（TDR0n）的值装入 TCR0n 寄存器，并且以间隔定时器模式开始计数。
- ⑤ 如果 TCR0n 寄存器递减计数到“0000H”，就通过下一个计数时钟，产生 INTTM0n，并且在将定时器数据寄存器 0n（TDR0n）的值装入 TCR0n 寄存器后继续进行计数。

图 6-24 运行时序（间隔定时器模式）



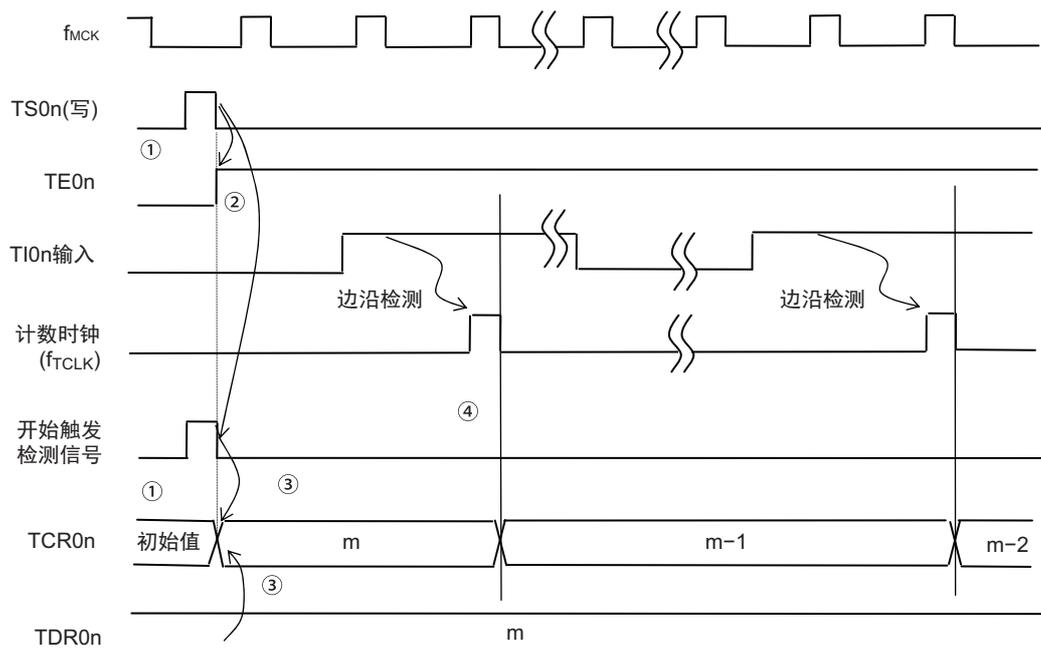
注意 因为第 1 个计数时钟（ f_{TCLK} ）周期的运行在写 TS0n 位后并且在产生计数时钟（ f_{TCLK} ）前会延迟开始计数，所以产生最大为 1 个计数时钟（ f_{TCLK} ）周期的误差。另外，如果需要开始计数时序的信息，就将 MD0n0 位置“1”，以便能在开始计数时产生中断请求信号（INTTM0n）。

备注 f_{MCK} 、开始触发检测信号和 INTTM0n 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 ($TE0n=0$) 的期间, 定时器计数寄存器 $0n$ ($TCR0n$) 保持初始值。
- ② 通过给 $TS0n$ 位写“1”, 进入运行允许状态 ($TE0n=1$)。
- ③ 在 $TS0n$ 位和 $TE0n$ 位都变为“1”的同时将定时器数据寄存器 $0n$ ($TDR0n$) 的值装入 $TCR0n$ 寄存器, 并且开始计数。
- ④ 此后, 在 $TI0n$ 输入的有效边沿, 通过计数时钟 (f_{TCLK}) 对 $TCR0n$ 寄存器的值进行递减计数。

图 6-25 运行时序 (事件计数器模式)

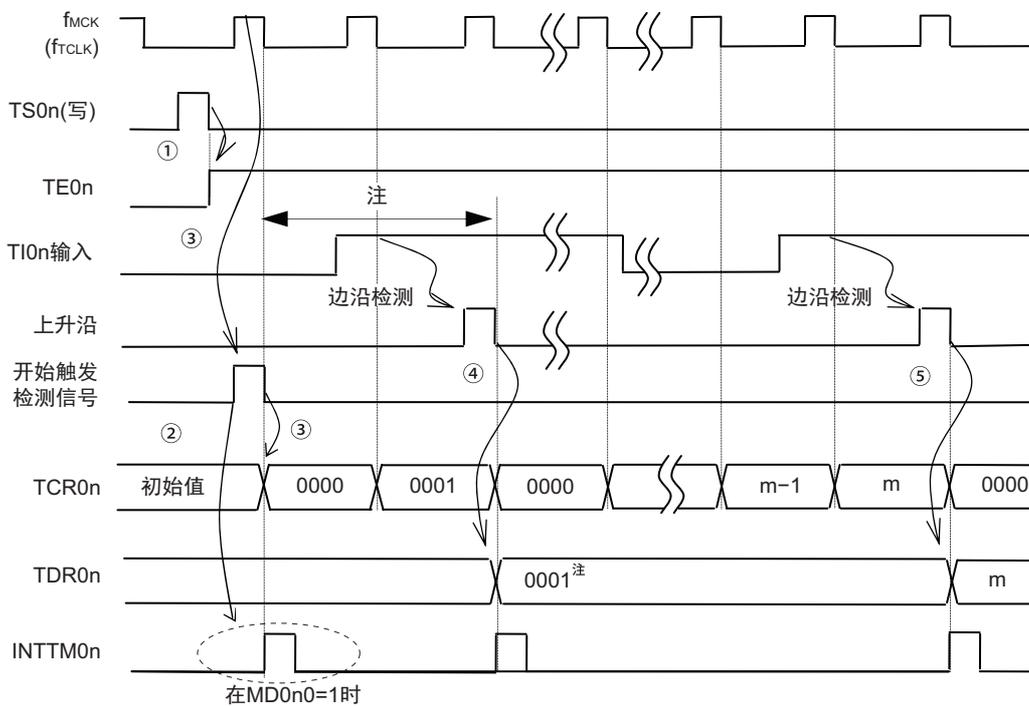


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从 $TI0n$ 输入开始再延迟 2 个运行时钟 (f_{MCK}) 周期 (合计 3 ~ 4 个周期)。因为 $TI0n$ 输入与运行时钟 (f_{MCK}) 异步, 所以产生 1 个周期的误差。

(3) 捕捉模式的运行（输入脉冲的间隔测量）

- ① 通过给 TS0n 位写“1”，进入运行允许状态（TE0n=1）。
- ② 定时器计数寄存器 0n（TCR0n）保持初始值，直到产生计数（ f_{TCLK} ）时钟为止。
- ③ 通过允许运行后的第 1 个计数时钟，产生开始触发信号。然后，将“0000H”装入 TCR0n 寄存器并且以捕捉模式开始计数（当 MD0n0 位为“1”时，通过开始触发信号产生 INTTM0n）。
- ④ 如果检测到 TI0n 输入的有效边沿，就将 TCR0n 寄存器的值捕捉到 TDR0n 寄存器，并且产生中断请求信号（INTTM0n）。此时的捕捉值没有意义。TCR0n 寄存器从“0000H”开始继续进行计数。
- ⑤ 如果检测到下一个 TI0n 输入的有效边沿，就将 TCR0n 寄存器的值捕捉到 TDR0n 寄存器，并且产生中断请求信号（INTTM0n）。

图 6-26 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 TI0n（有触发）时，即使没有检测边沿也通过写 TS0n（①）产生开始触发（③）从而开始计数，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。

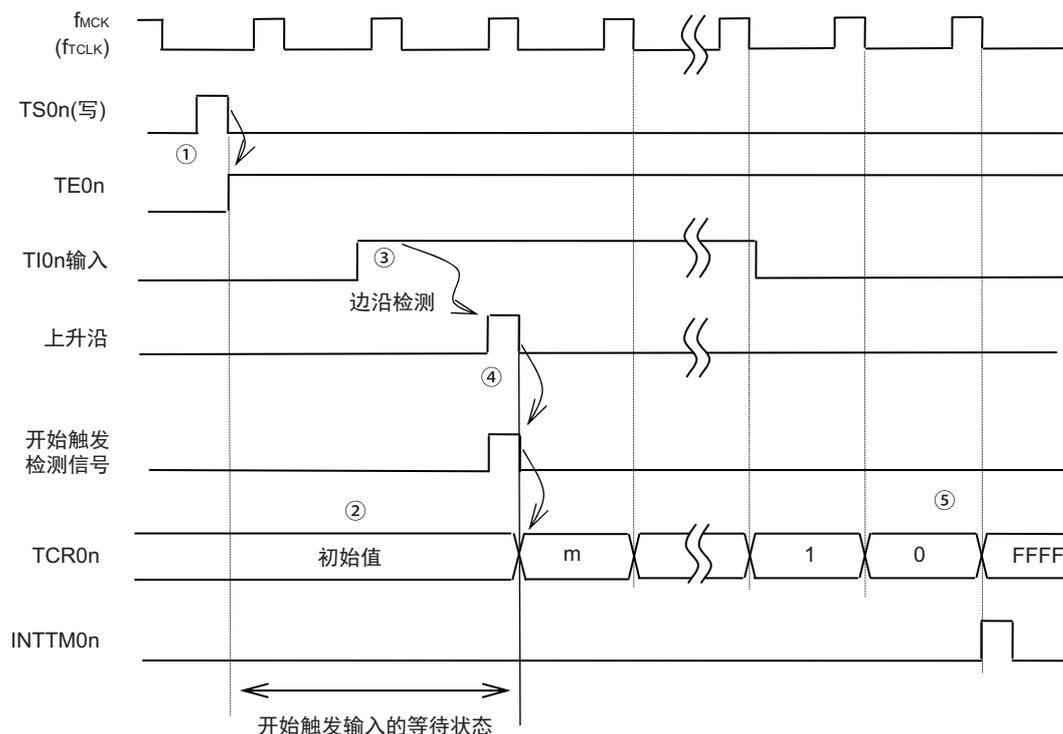
注意 因为第 1 个计数时钟（ f_{TCLK} ）周期的运行在写 TS0n 位后并且在产生计数（ f_{TCLK} ）时钟前会延迟开始计数，所以产生最大为 1 个计数时钟（ f_{TCLK} ）周期的误差。另外，如果需要开始计数时序的信息，就将 MD0n0 位置“1”，以便能在开始计数时产生中断请求信号（INTTM0n）。

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 TI0n 输入开始再延迟 2 个运行时钟（ f_{MCK} ）周期（合计 3~4 个周期）。因为 TI0n 输入与运行时钟（ f_{MCK} ）异步，所以产生 1 个周期的误差。

(4) 单次计数模式的运行

- ① 通过给 TS0n 位写“1”，进入运行允许状态 (TE0n=1)。
- ② 定时器计数寄存器 0n (TCR0n) 保持初始值，直到产生开始触发信号为止。
- ③ 检测 TI0n 输入的上升沿。
- ④ 在产生开始触发信号后将 TDR0n 寄存器的值 (m) 装入 TCR0n 寄存器，并且开始计数。
- ⑤ 当 TCR0n 寄存器递减计数到“0000H”时，产生中断请求信号 (INTTM0n)，并且 TCR0n 寄存器的值变为“FFFFH”，停止计数。

图 6-27 运行时序 (单次计数模式)

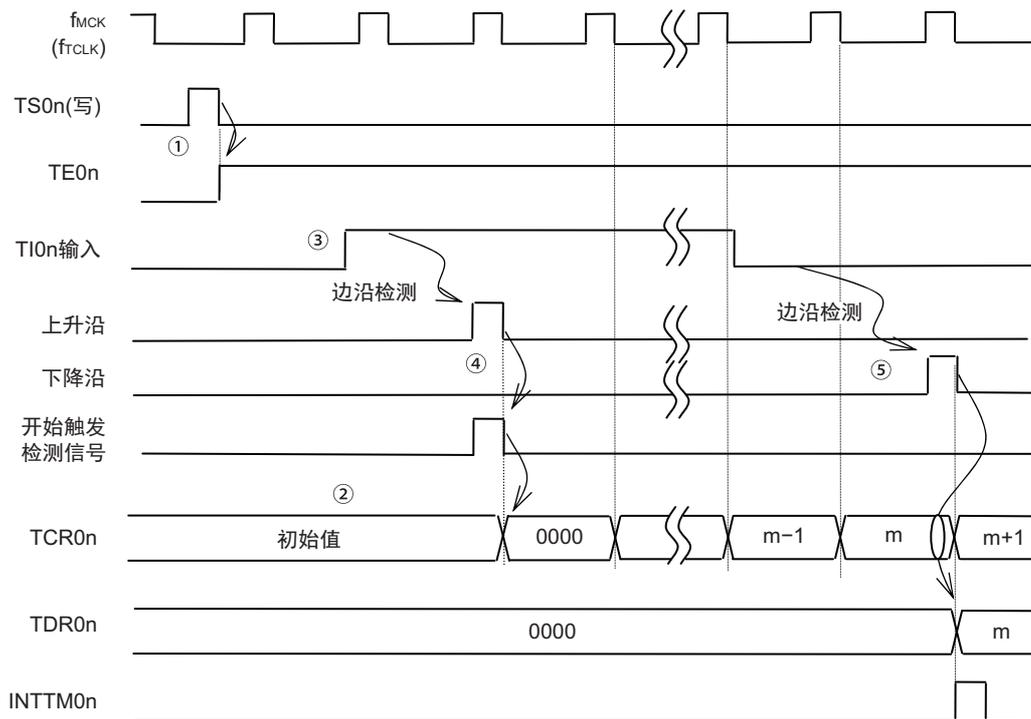


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI0n 输入开始再延迟 2 个运行时钟 (f_{MCK}) 周期 (合计 3~4 个周期)。因为 TI0n 输入与运行时钟 (f_{MCK}) 异步，所以产生 1 个周期的误差。

(5) 捕捉 & 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 0（TS0）的 TS0n 位写“1”，进入运行允许状态（TE0n=1）。
- ② 定时器计数寄存器 0n（TCR0n）保持初始值，直到产生开始触发信号为止。
- ③ 检测 TI0n 输入的上升沿。
- ④ 在产生开始触发信号后将“0000H”装入 TCR0n 寄存器，并且开始计数。
- ⑤ 如果检测到 TI0n 输入的下降沿，就将 TCR0n 寄存器的值捕捉到 TDR0n 寄存器，并且产生中断请求信号（INTTM0n）。

图 6-28 运行时序（捕捉 & 单次计数模式：测量高电平宽度）

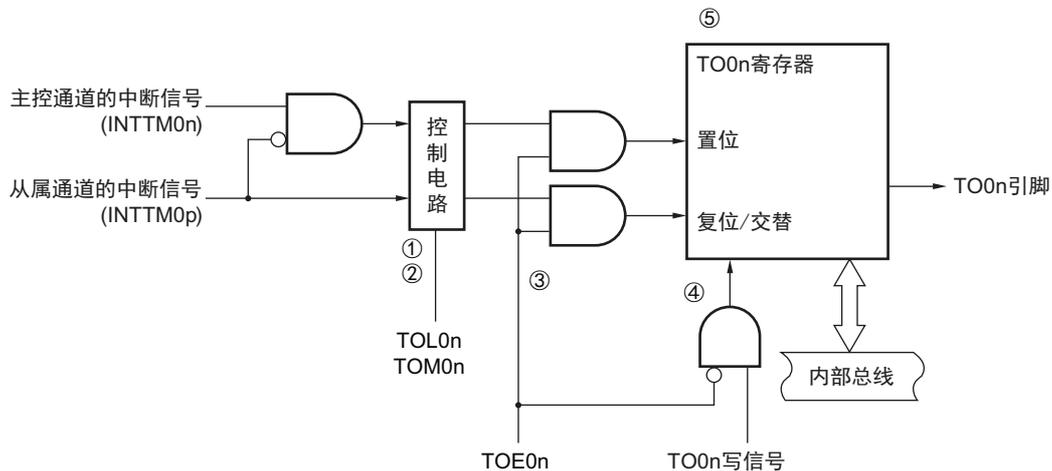


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI0n 输入开始再延迟 2 个运行时钟（ f_{MCK} ）周期（合计 3~4 个周期）。因为 TI0n 输入与运行时钟（ f_{MCK} ）异步，所以产生 1 个周期的误差。

6.6 通道输出（TO0n 引脚）的控制

6.6.1 TO0n 引脚输出电路的结构

图 6-29 输出电路的结构



以下说明 TO0n 引脚的输出电路。

① 当 TOM0n 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 0（TOL0）的设定值，只将 INTTM0p（从属通道定时器中断）传给定时器输出寄存器 0（TO0）。

② 当 TOM0n 位为“1”（从属通道输出模式）时，将 INTTM0n（主控通道定时器中断）和 INTTM0p（从属通道定时器中断）传给 TO0 寄存器。

此时，TOL0 寄存器有效并且如下控制信号：

TOL0n=0 时：正逻辑输出（INTTM0n→置位、INTTM0p→复位）

TOL0n=1 时：负逻辑输出（INTTM0n→复位、INTTM0p→置位）

当同时产生 INTTM0n 和 INTTM0p 时（PWM 输出的 0% 输出），优先 INTTM0p（复位信号）而屏蔽 INTTM0n（置位信号）。

③ 在定时器输出允许状态（TOE0n=1）下，将 INTTM0n（主控通道定时器中断）和 INTTM0p（从属通道定时器中断）传给 TO0 寄存器。TO0 寄存器的写操作（TO0n 写信号）无效。

当 TOE0n 位为“1”时，除了中断信号以外，不改变 TO0n 引脚的输出。

要对 TO0n 引脚的输出电平进行初始化时，需要在设定为定时器输出禁止状态（TOE0n=0）后给 TO0 寄存器写值。

④ 在定时器输出禁止状态（TOE0n=0）下，对象通道的 TO0n 位的写操作（TO0n 写信号）有效。当定时器输出为禁止状态（TOE0n=0）时，不将 INTTM0n（主控通道定时器中断）和 INTTM0p（从属通道定时器中断）传给 TO0 寄存器。

⑤ 能随时读 TO0 寄存器，并且能确认 TO0n 引脚的输出电平。

备注 n：通道号。n=0（10 引脚产品），n=0、2（16 引脚产品）

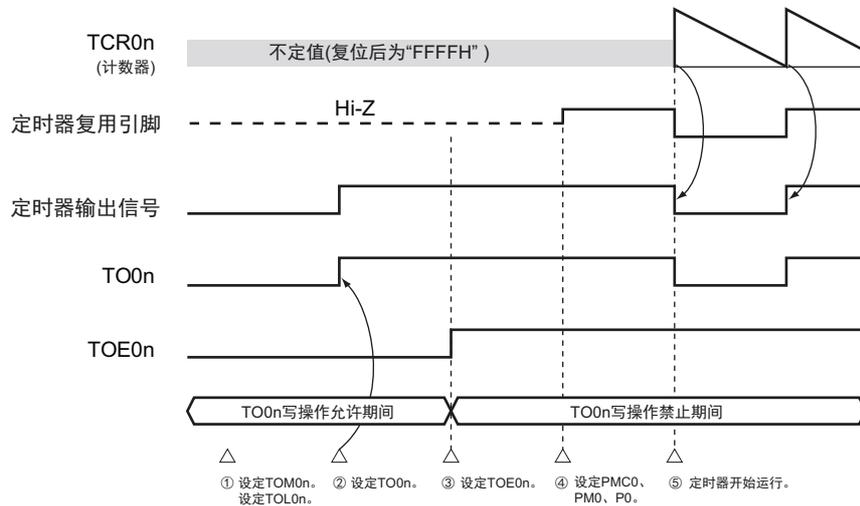
p：从属通道号

$n < p \leq 3$

6.6.2 TO0n 引脚的输出设定

从 TO0n 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 6-30 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。
 - TOM0n位 (0: 主控通道输出模式、1: 从属通道输出模式)
 - TOL0n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0 (TO0)，将定时器输出信号设定为初始状态。
- ③ 给TOE0n位写“1”，允许定时器输出 (禁止写TO0寄存器)。
- ④ 通过端口模式控制寄存器0 (PMC0) 将端口设定为数字输入/输出，通过端口模式寄存器0 (PM0) 将端口设定为输出模式，通过端口寄存器0 (P0) 将端口的输出锁存器设定为“0” (参照“6.3.14 定时器输入/输出引脚的端口功能控制寄存器”)。
- ⑤ 允许定时器运行 (TS0n=1)。

备注 n: 通道号。n=0、1 (10 引脚产品)，n=0 ~ 3 (16 引脚产品)

6.6.3 通道输出运行的注意事项

(1) 有关定时器运行中的 TO0、TOE0、TOL0、TOM0 寄存器的设定值变更

定时器运行（定时器计数寄存器 0n（TCR0n）和定时器数据寄存器 0n（TDR0n）的运行）和 TO0n 输出电路相互独立。因此，定时器输出寄存器 0（TO0）、定时器输出允许寄存器 0（TOE0）、定时器输出电平寄存器 0（TOL0）和定时器输出模式寄存器 0（TOM0）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 TO0n 引脚输出期待的波形，必须设定为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请求信号（INTTM0n）前后更改除了 TO0 寄存器以外的 TOE0 寄存器、TOL0 寄存器和 TOM0 寄存器的设定值，就根据是在产生 INTTM0n 前更改还是在产生 INTTM0n 后更改，TO0n 引脚输出的波形可能不同。

备注 n：通道号。n=0、1（10 引脚产品），n=0～3（16 引脚产品）

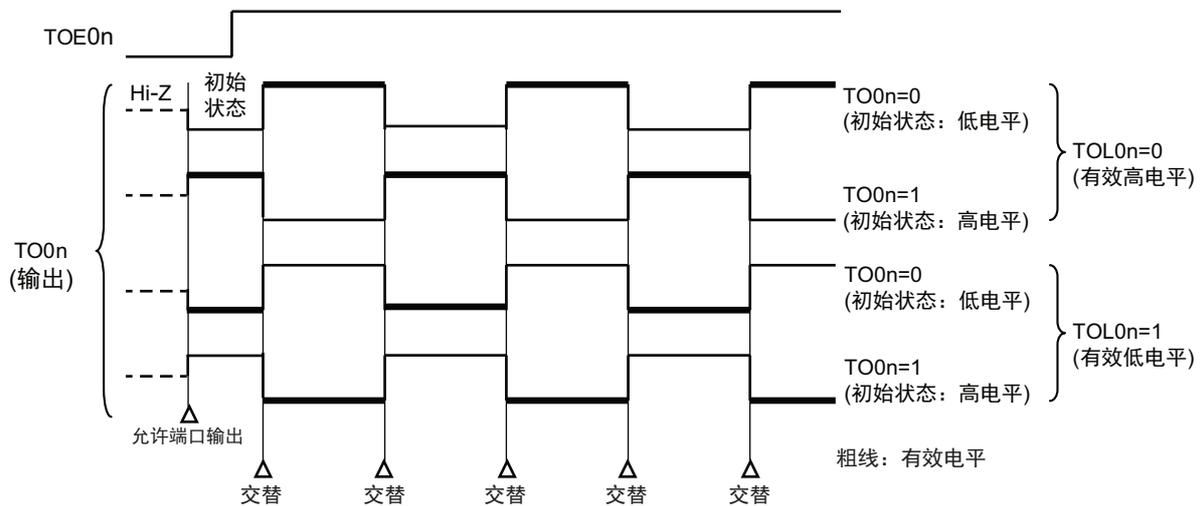
(2) 有关 TO0n 引脚的初始电平和定时器开始运行后的输出电平

在允许端口输出前并且在定时器输出禁止（TOE0n=0）的状态下写定时器输出寄存器 0（TO0），在更改初始电平后设定为定时器输出允许状态（TOE0n=1）时的 TO0n 引脚输出电平的变化如下所示。

(a) 在主导通道输出模式（TOM0n=0）中开始运行的情况

在主导通道输出模式（TOM0n=0）中，定时器输出电平寄存器 0（TOL0）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 TO0n 引脚的输出电平。

图 6-31 交替输出时（TOM0n=0）的 TO0n 引脚输出状态



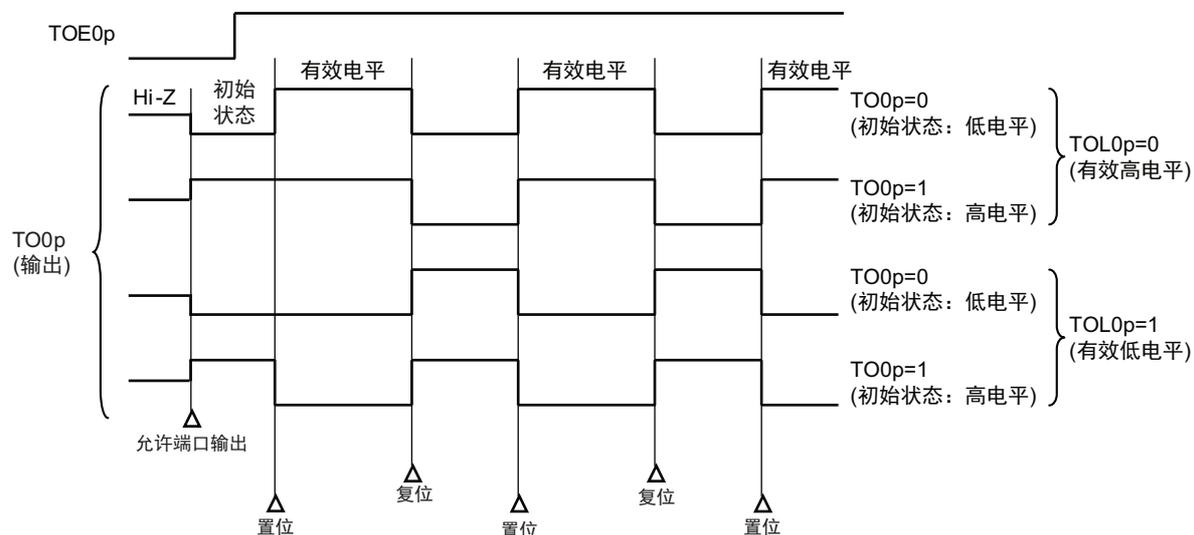
备注 1. 交替：反相 TO0n 引脚的输出状态。

2. n：通道号。n=0、1（10 引脚产品），n=0～3（16 引脚产品）

(b) 在从属通道输出模式 (TOM0p=1) 中开始运行的情况 (PWM 输出)

在从属通道输出模式 (TOM0p=1) 中, 有效电平取决于定时器输出电平寄存器 0 (TOL0) 的设定。

图 6-32 PWM 输出时 (TOM0p=1) 的 TO0p 引脚输出状态



备注 1. 置位: TO0p 引脚的输出信号从无效电平变为有效电平。

复位: TO0p 引脚的输出信号从有效电平变为无效电平。

2. p: 通道号 ($n < p \leq 3$)

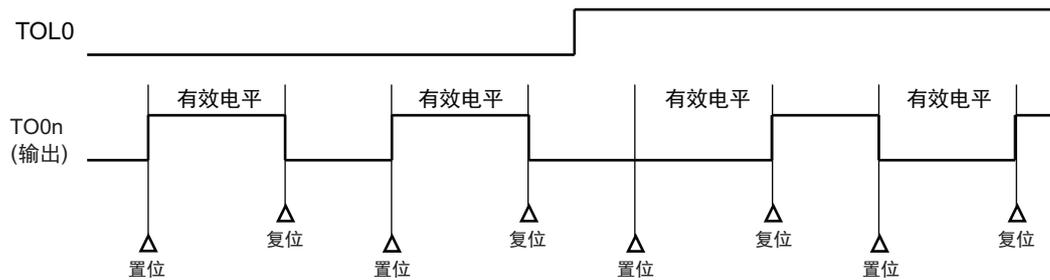
(3) 有关从属通道输出模式 (TOM0n=1) 的 TO0n 引脚变化

(a) 在定时器运行中更改定时器输出电平寄存器 0 (TOL0) 的设定的情况

如果在定时器运行中更改 TOL0 寄存器的设定, 就在满足 TO0n 引脚变化条件时设定有效。无法通过改写 TOL0 寄存器来改变 TO0n 引脚的输出电平。

当 TOM0n 位为“1”时, 在定时器运行中 (TE0n=1) 更改 TOL0 寄存器的值时的运行如下所示。

图 6-33 在定时器运行中更改 TOL0 寄存器的内容时的运行



备注 1. 置位: TO0n 引脚的输出信号从无效电平变为有效电平。

复位: TO0n 引脚的输出信号从有效电平变为无效电平。

2. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

(b) 置位 / 复位时序

为了在 PWM 输出时实现 0%/100% 输出, 通过从属通道将产生主控通道中断请求信号 (INTTM0n) 时的 TO0n 引脚 / TO0n 位的设定时序延迟 1 个计数时钟 (f_{TCLK})。

当置位条件和复位条件同时产生时, 优先复位条件。

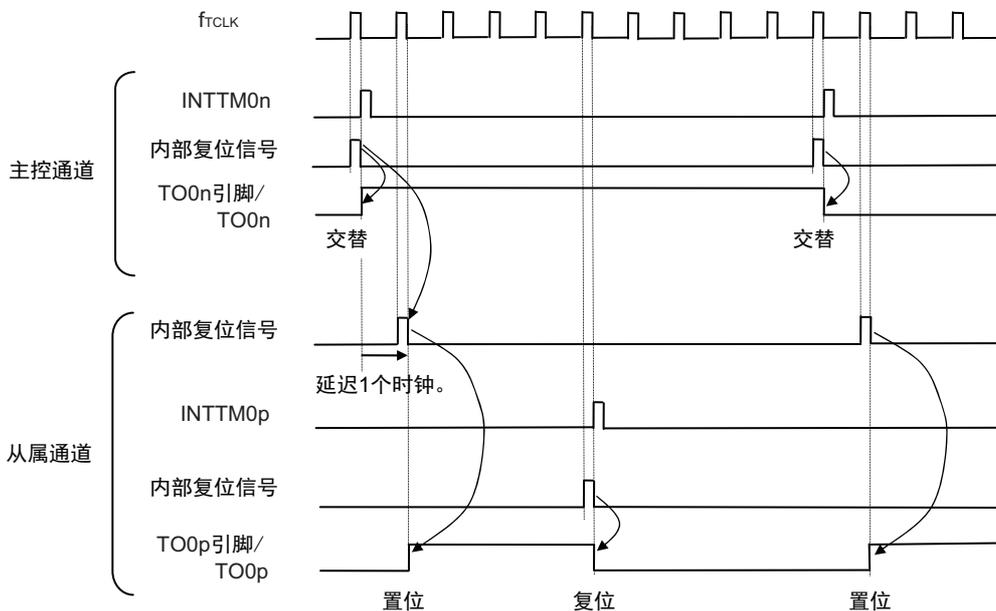
按照以下方法设定主控 / 从属通道时的置位 / 复位运行状态如图 6-34 所示。

主控通道: TOE0n=1、TOM0n=0、TOL0n=0

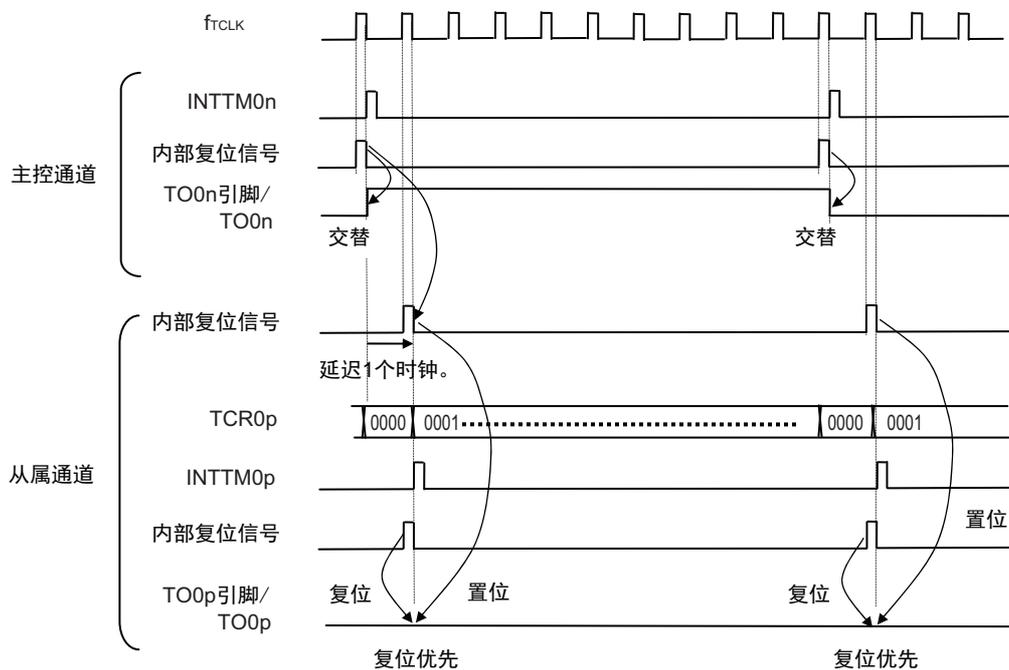
从属通道: TOE0p=1、TOM0p=1、TOL0p=0

图 6-34 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: $TO0n$ 引脚的复位 / 交替信号

内部置位信号: $TO0n$ 引脚的置位信号

2. n: 主控通道号。n=0 (10 引脚产品), n=0、2 (16 引脚产品)

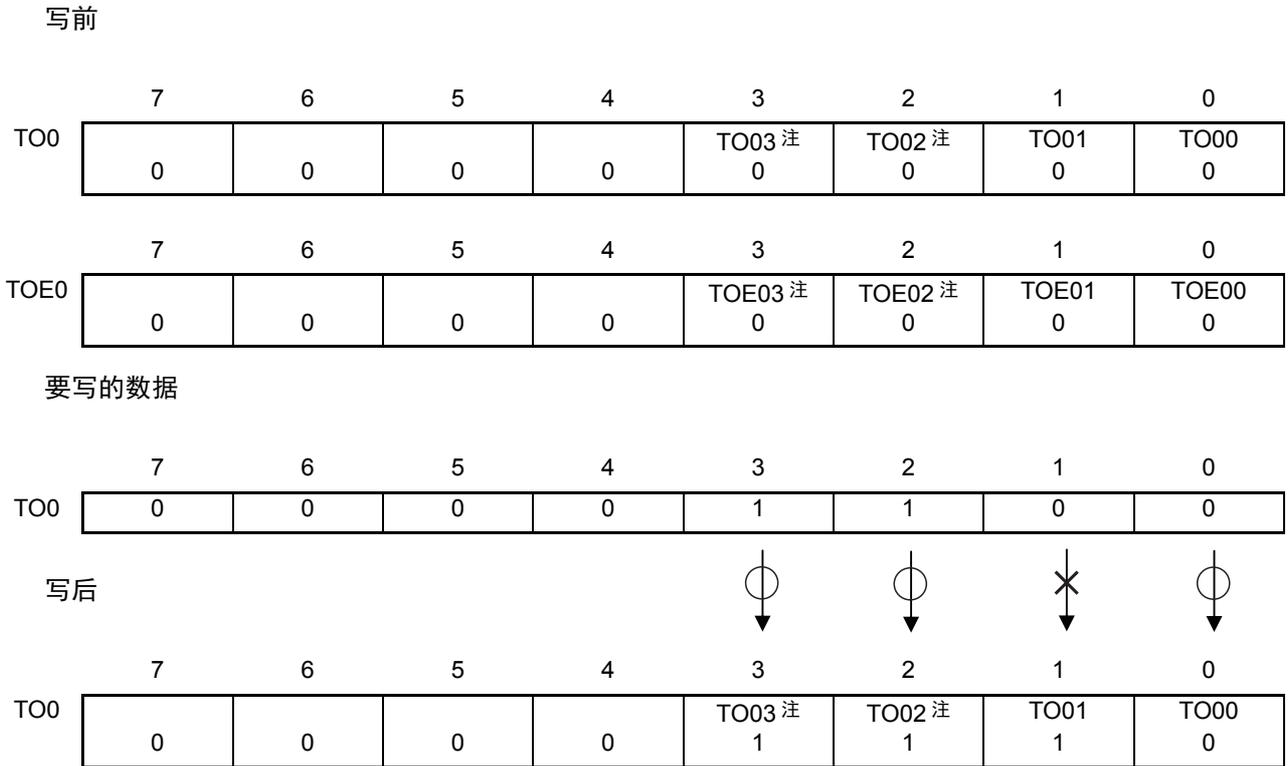
p: 从属通道号

$n < p \leq 3$

6.6.4 TO0n 位的一次性操作

和定时器通道开始寄存器 0 (TS0) 相同, 定时器输出寄存器 0 (TO0) 有全部通道的设定位 (TO0n), 因此能一次性地操作全部通道的 TO0n 位。另外, 能只给想作为操作对象的通道输出 (TO0n) 的 TO0n 位写数据 (TOE0n=0), 因此也能只操作任意的位。

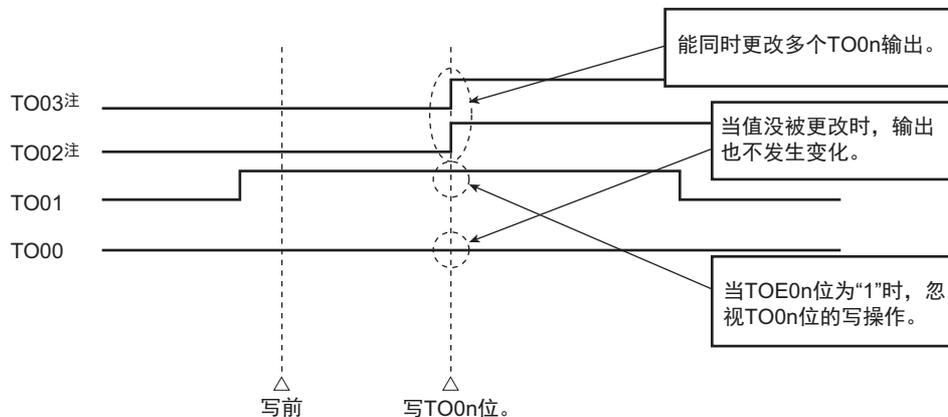
图 6-35 TO0n 位的一次性操作例子



只能写 TOE0n 位为“0”的 TO0n 位, 忽视 TOE0n 位为“1”的 TO0n 位的写操作。

TOE0n 位为“1”的 TO0n (通道输出) 不受写操作的影响, 即使写 TO0n 位也被忽视, 由定时器运行引起的输出变化正常进行。

图 6-36 一次性操作 TO0n 位时的 TO0n 引脚状态



注 只限于 16 引脚产品

备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

6.6.5 有关开始计数时的定时器中断和 TO0n 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 0n（TMR0n）的 MD0n0 位是设定是否在开始计数时产生定时器中断的位。

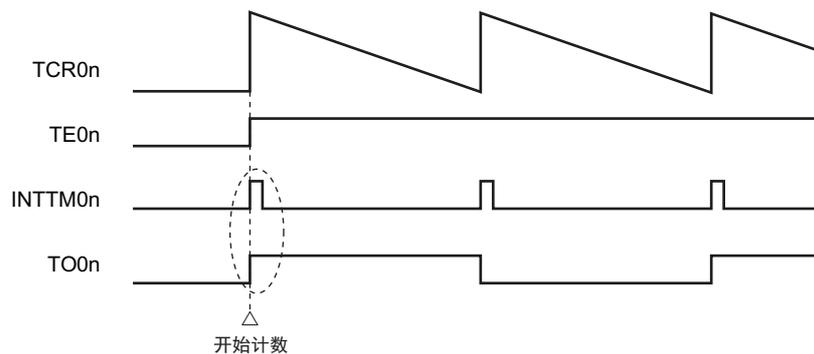
当 MD0n0 位为“1”时，能通过产生中断请求信号（INTTM0n）得知计数的开始时序。

在其他模式中，不控制开始计数时的 INTTM0n 和 TO0n 输出。

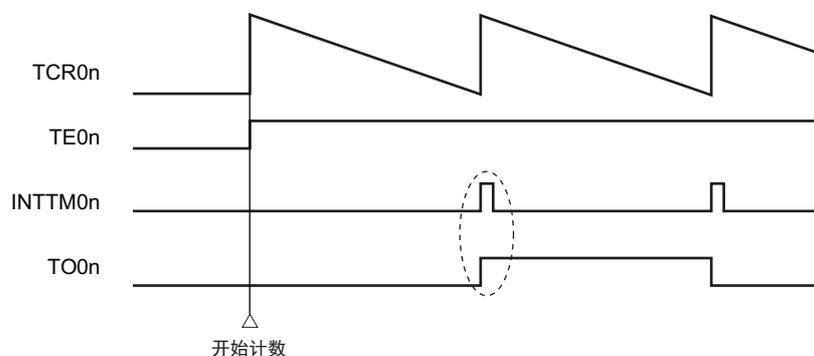
设定为间隔定时器模式（TOE0n=1、TOM0n=0）时的运行例子如下所示。

图 6-37 开始计数时的定时器中断和 TO0n 输出的运行例子

(a) MD0n0 位为“1”的情况



(b) MD0n0 位为“0”的情况



当 MD0n0 位为“1”时，在开始计数时输出定时器中断（INTTM0n）并且 TO0n 进行交替输出。

当 MD0n0 位为“0”时，在开始计数时不输出定时器中断（INTTM0n）并且 TO0n 也不发生变化，而在对 1 个周期进行计数后输出 INTTM0n 并且 TO0n 进行交替输出。

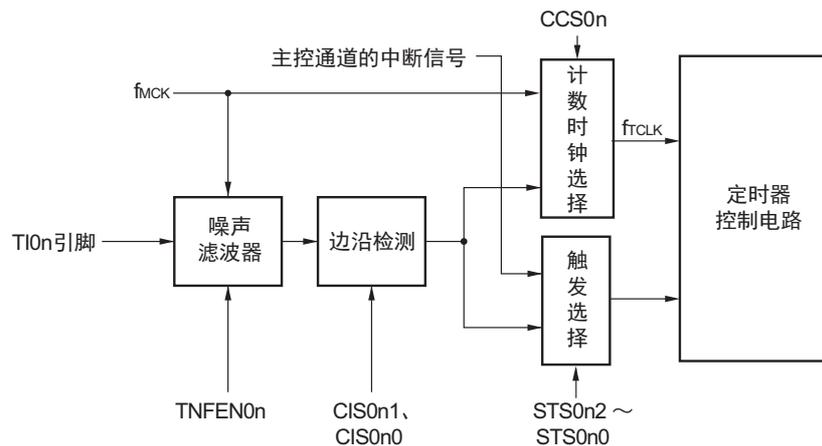
备注 n：通道号。n=0、1（10 引脚产品），n=0～3（16 引脚产品）

6.7 定时器输入 (TI0n) 的控制

6.7.1 TI0n 的输入电路结构

定时器输入引脚的信号经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，必须将对应引脚的噪声滤波器设定为有效。输入电路的结构图如图 6-38 所示。

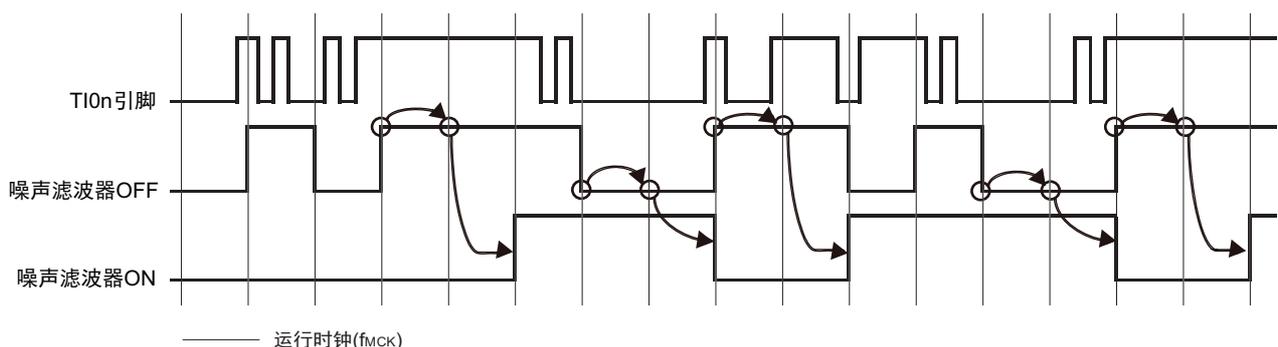
图 6-38 输入电路的结构图



6.7.2 噪声滤波器

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。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，执行 2 个时钟的一致检测。在 TI0n 输入引脚的噪声滤波器 ON/OFF 时经过噪声滤波器电路后的波形如图 6-39 所示。

图 6-39 TI0n 输入引脚的噪声滤波器 ON/OFF 时的采样波形



6.7.3 进行通道输入操作时的注意事项

在设定为不使用定时器输入引脚时，不向噪声滤波器电路提供运行时钟。因此，在设定为使用定时器输入引脚后，到设定定时器输入引脚的对应通道的运行允许触发前，需要以下的等待时间。

1. 噪声滤波器OFF的情况

在从定时器模式寄存器0n (TMR0n)的bit12 (CCS0n)、bit9 (STS0n1)、bit8 (STS0n0)全部为“0”的状态，到将其中1位置位后，必须在至少经过2个运行时钟 (f_{MCK})周期后，才能设定定时器通道开始寄存器 (TS0)的运行允许触发。

2. 噪声滤波器ON的情况

在从定时器模式寄存器0n (TMR0n)的bit12 (CCS0n)、bit9 (STS0n1)、bit8 (STS0n0)全部为“0”的状态，到将其中1位置位后，必须在至少经过4个运行时钟 (f_{MCK})周期后，才能设定定时器通道开始寄存器 (TS0)的运行允许触发。

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中断请求信号（INTTM0n）的基准定时器。
INTTM0n 的产生周期能用以下计算式进行计算：

$$\text{INTTM0n 的产生周期} = \text{计数时钟周期} \times (\text{TDR0n 的设定值} + 1)$$

当通道 1 和通道 3 用作 8 位定时器运行功能（SPLIT0n=1）时，高 8 位定时器和低 8 位定时器都能用作间隔定时器。

(2) 作为方波输出的运行

TO0n 引脚在产生 INTTM0n 的同时进行交替输出，输出占空比为 50% 的方波。
TO0n 输出方波的周期和频率能用以下计算式进行计算：

$$\bullet \text{ TO0n 引脚输出的方波周期} = \text{计数时钟周期} \times (\text{TDR0n 的设定值} + 1) \times 2$$

$$\bullet \text{ TO0n 引脚输出的方波频率} = \text{计数时钟频率} / \{(\text{TDR0n 的设定值} + 1) \times 2\}$$

当通道 1 和通道 3 用作 8 位定时器运行功能（SPLIT0n=1）时，只有低 8 位定时器都能用作方波输出。

在间隔定时器模式中，定时器计数寄存器 0n（TCR0n）用作递减计数器。

在将定时器通道开始寄存器 0（TS0、TSH0）的通道开始触发位（TS0n、TSH01、TSH03）置“1”后，通过第 1 个计数时钟将定时器数据寄存器 0n（TDR0n）的值装入 TCR0n 寄存器。此时，如果定时器模式寄存器 0n（TMR0n）的 MD0n0 位为“0”，就不输出 INTTM0n 并且 TO0n 也不进行交替输出。如果 TMR0n 寄存器的 MD0n0 位为“1”，就输出 INTTM0n 并且 TO0n 进行交替输出。

然后，TCR0n 寄存器通过计数时钟进行递减计数。

如果 TCR0n 变为“0000H”，就通过下一个计数时钟输出 INTTM0n 并且 TO0n 进行交替输出。同时，再次将 TDR0n 寄存器的值装入 TCR0n 寄存器。此后，继续同样的运行。

能随时改写 TDR0n 寄存器，改写的 TDR0n 寄存器的值从下一个周期开始有效。

备注 n：通道号。n=0、1（10 引脚产品），n=0～3（16 引脚产品）

图 6-40 作为间隔定时器 / 方波输出运行的框图

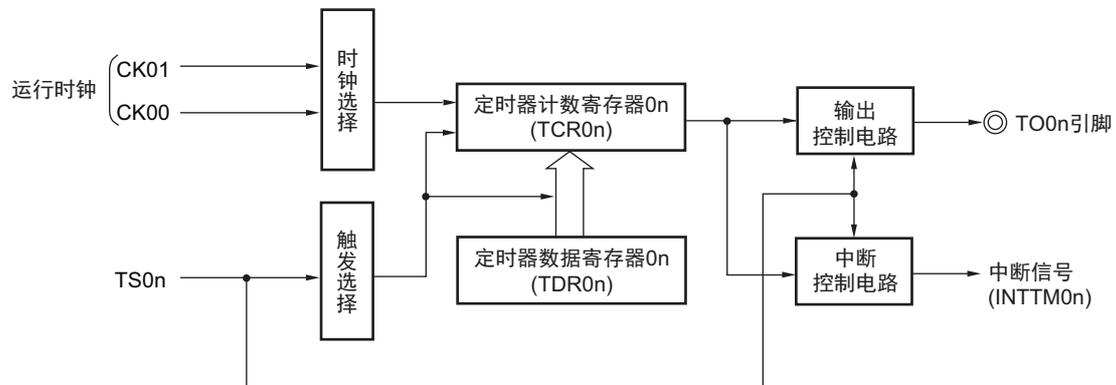
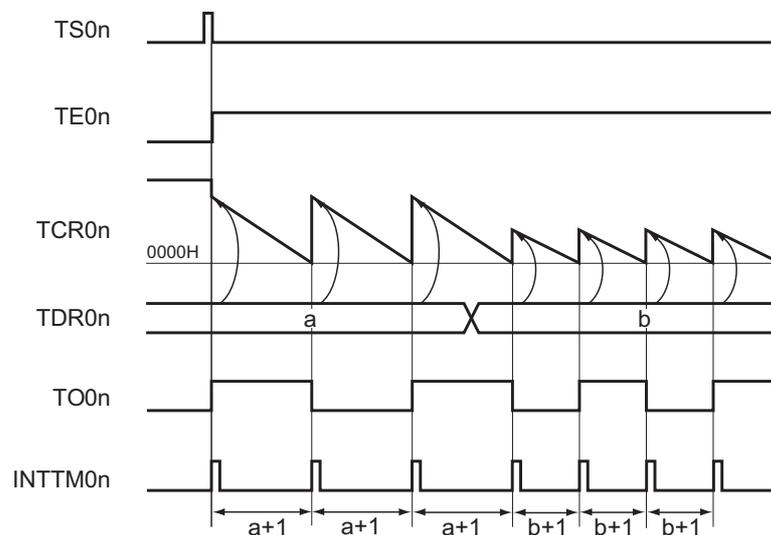


图 6-41 作为间隔定时器 / 方波输出运行的基本时序例子 (MD0n0=1)

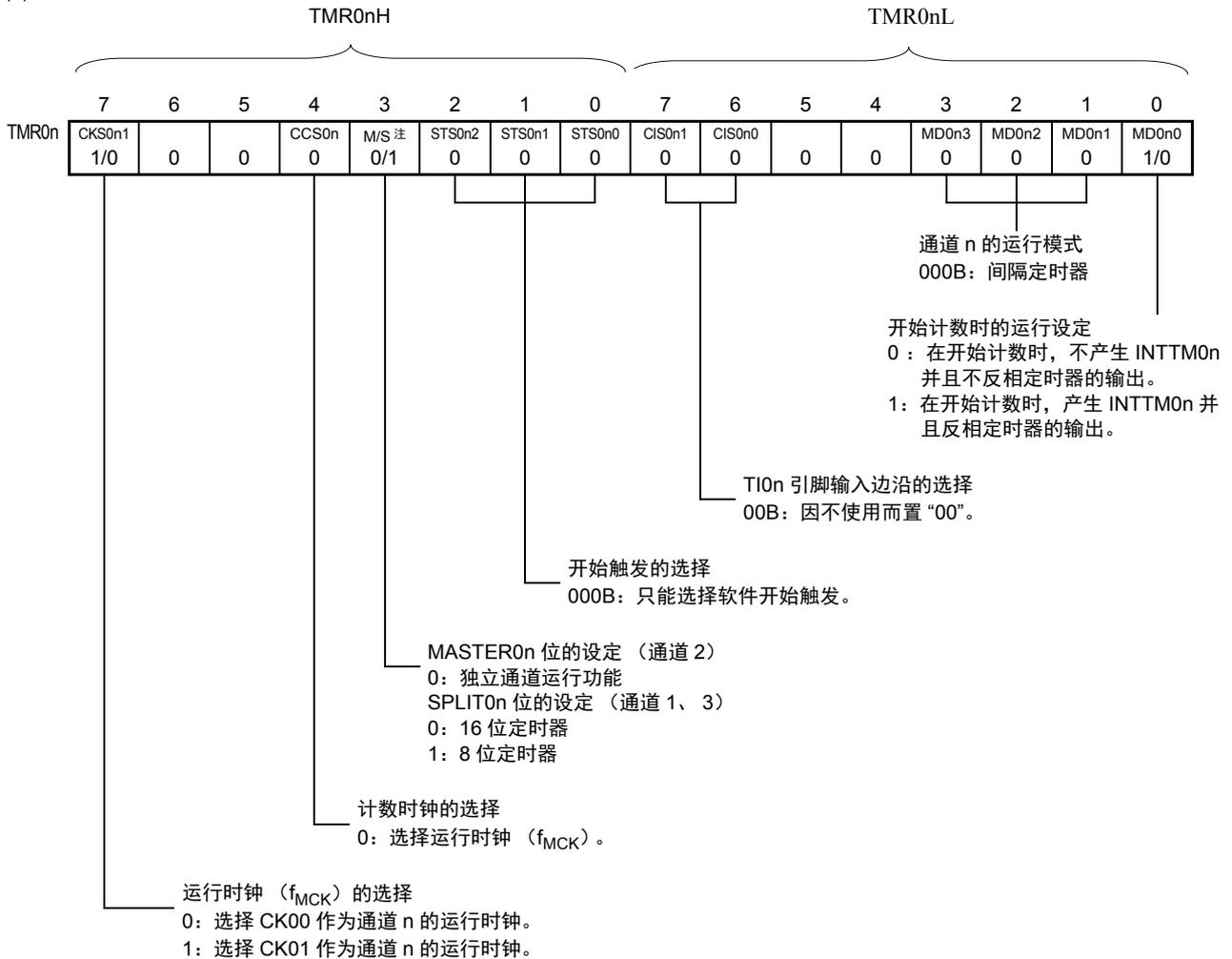


备注 1. n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

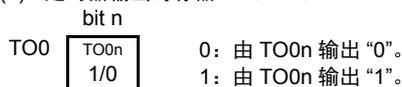
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- TO0n: TO0n 引脚输出信号

图 6-42 间隔定时器 / 方波输出的寄存器设定内容例子

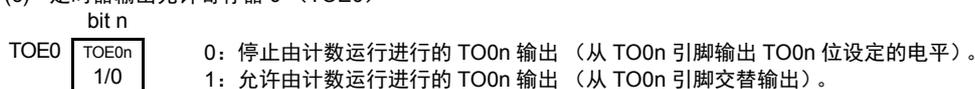
(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



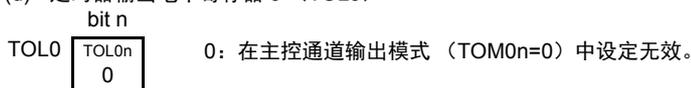
(b) 定时器输出寄存器 0 (TO0)



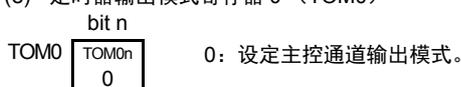
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00: 固定为“0”。

备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

图 6-43 间隔定时器 / 方波输出的操作步骤

	软件操作	硬件状态
TAU 初始 设定	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1” (在 TAU0EN 位为“0”时不能读写)。	断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
通道初 始设定	设定定时器模式寄存器 0n (TMR0n) (确定各通道的运行模式)。 给定时器数据寄存器 0n (TDR0n) 设定间隔 (周期) 值 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。	通道处于运行停止状态。 TO0n 引脚处于 Hi-Z 状态。 (端口模式寄存器为输入模式)
	用作方波输出的情况: 设定 TO0n 位, 确定 TO0n 输出的初始电平。 将 TOE0n 位置“1”, 允许 TO0n 的运行。	因为通道处于运行停止状态, 所以 TO0n 不变 (即使改写 TO0p 位也不反映到 TO0p 引脚)。 从 TO0n 引脚输出 TO0n 位设定的电平。
重新 开始 运行	开始运行 将 TOE0n 位置“1”, 允许 TO0n 的运行 (只在重新开始方波输出运行时)。 将 TS0/TSH0 寄存器的对象位置“1”。	因为 TS0/TSH0 寄存器的对象位是触发位, 所以自动返回到“0”。
	运行中 能任意更改 TDR0n 寄存器的设定值。 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 能更改 TO0、TOE0 寄存器的对象位的设定值。 禁止更改 TMR0n、TOM0 和 TOL0 寄存器的对象位的设定值。	TE0/TEH0 寄存器的对象位变为“1”并且开始计数。 通过下一个计数时钟将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n)。当 TMR0nL 寄存器的 MD0n0 位为“1”时, 产生 INTTM0n 并且 TO0n 进行交替输出。 计数器 (TCR0n) 进行递减计数。如果计数到“0000H”, 就再次将 TDR0n 寄存器的值装入 TCR0n 寄存器, 并且继续进行计数。当检测到 TCR0n 为“0000H”时, 产生 INTTM0n 并且 TO0n 进行交替输出。 此后, 重复此运行。
重新 开始 运行	停止运行 将 TT0/TTH0 寄存器的对象位置“1”。	TE0/TEH0 寄存器的对象位变为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。 TO0n 输出不被初始化而保持状态。
	因为 TT0/TTH0 寄存器的对象位是触发位, 所以自动返回到“0”。 将 TOE0n 位置“0”并且给 TO0n 位设定值。	从 TO0n 引脚输出 TO0n 位设定的电平。
TAU 停止	要保持 TO0n 引脚输出电平的情况: 在给端口寄存器设定要保持的值 (输出锁存器) 后将 TO0n 位置“0”。	通过端口功能保持 TO0n 引脚的输出电平。
	将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

注意 在 8 位定时器模式 (SPLIT=1) 中使用通道 1 和通道 3 时, 禁止读 TCR01H 寄存器和 TDR01H 寄存器或者 TCR03H 寄存器和 TDR03H 寄存器。

6.8.2 作为外部事件计数器的运行

对检测到的 TI0n 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，就能用作产生中断请求信号（INTTM0n）的事件计数器。规定的计数值能用以下计算式进行计算：

$$\text{规定的计数值} = \text{TDR0n 的设定值} + 1$$

在通道 1 和通道 3 用作 8 位定时器运行功能（SPLIT0n）时，只有低 8 位定时器能用作外部事件计数器。

在事件计数器模式中，定时器计数寄存器 0n（TCR0n）用作递减计数器。

通过将定时器通道开始寄存器 0（TS0）的任意通道开始触发位（TS0n）置“1”，将定时器数据寄存器 0n（TDR0n）的值装入 TCR0n 寄存器。

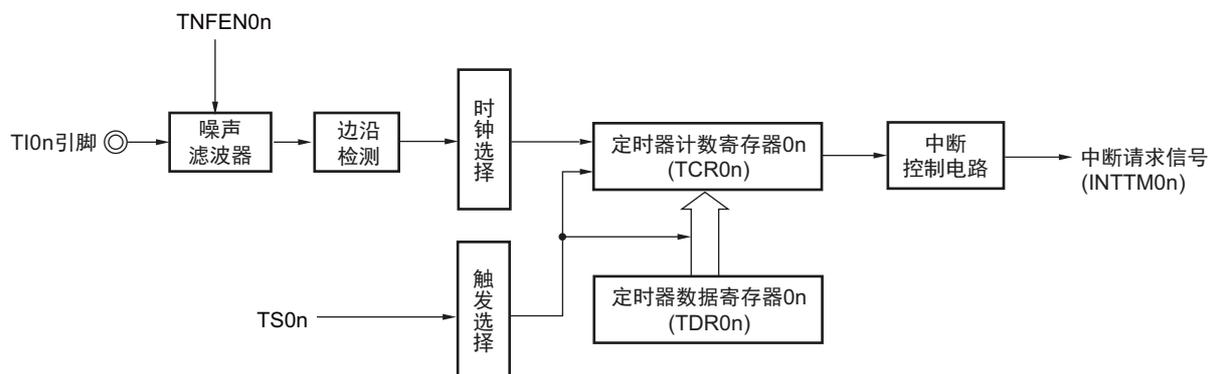
TCR0n 寄存器在检测到 TI0n 引脚输入的有效边沿的同时进行递减计数。如果 TCR0n 变为“0000H”，就再次装入 TDR0n 寄存器的值并且输出 INTTM0n。

此后，继续同样的运行。

因为 TO0n 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 0（TOE0）的 TOE0n 位置“0”，停止输出。

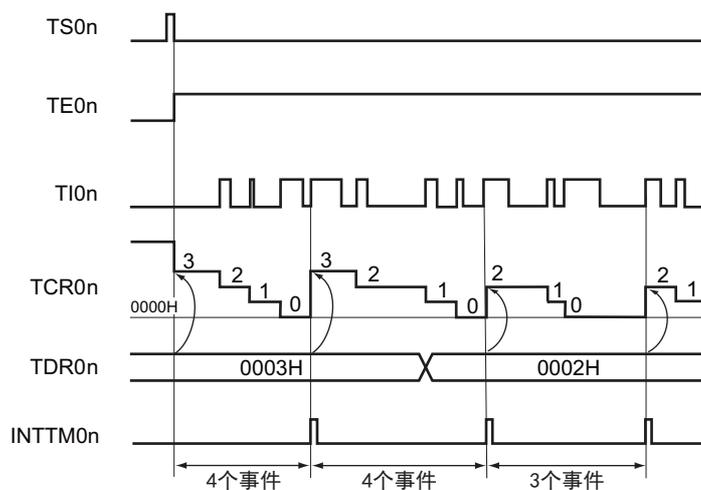
能随时改写 TDR0n 寄存器，改写的 TDR0n 寄存器的值在下一个计数期间有效。

图 6-44 作为外部事件计数器运行的框图



备注 n: 通道号。n=0、1（10 引脚产品），n=0～3（16 引脚产品）

图 6-45 作为外部事件计数器运行的基本时序例子

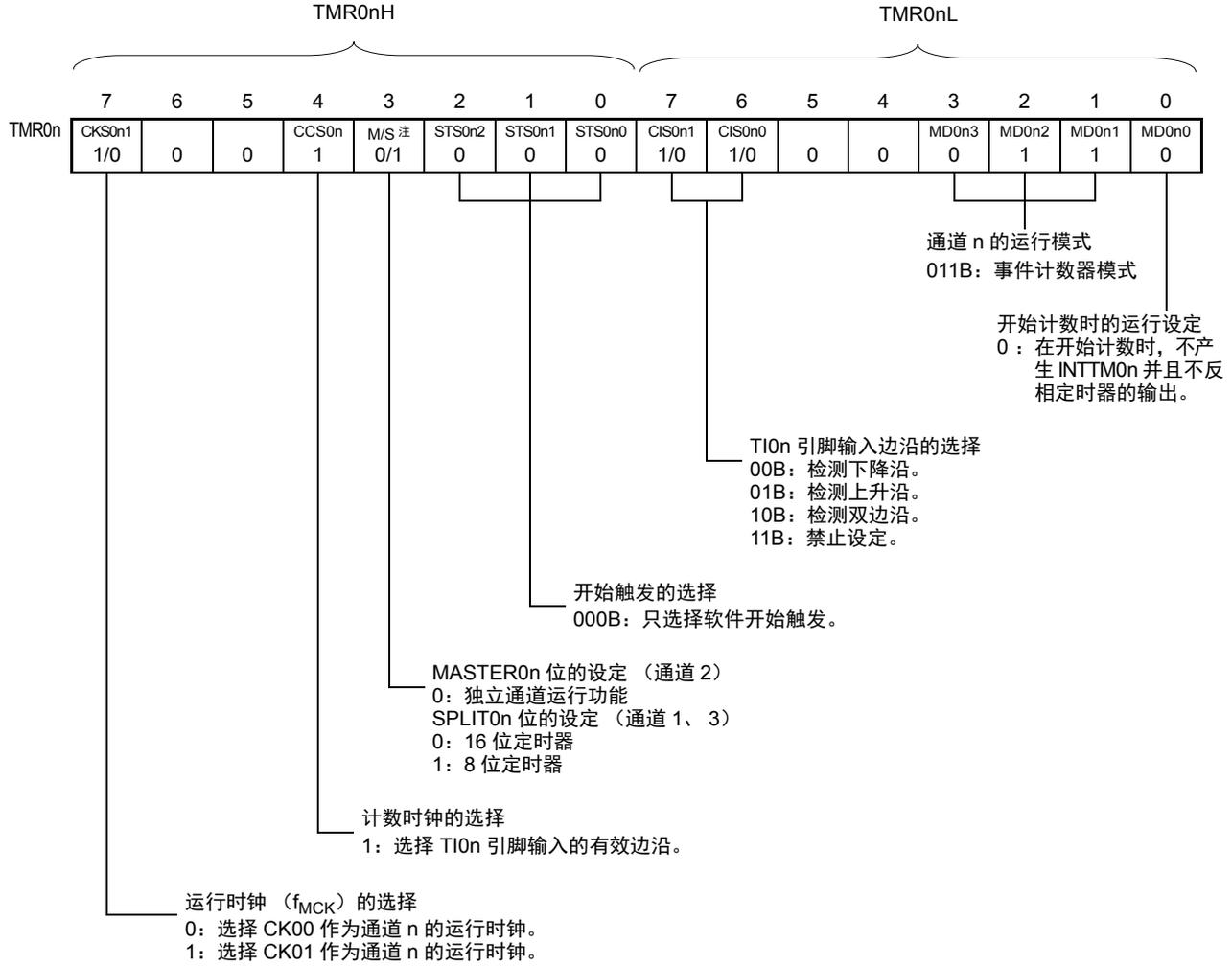


备注 1. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

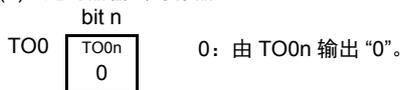
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)

图 6-46 外部事件计数器的寄存器设定内容例子

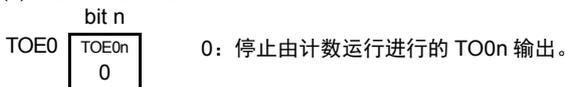
(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



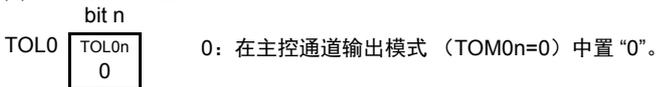
(b) 定时器输出寄存器 0 (TO0)



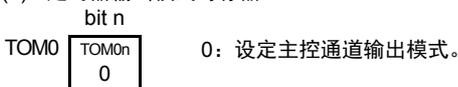
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00: 固定为“0”。

备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

图 6-47 外部事件计数器的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 “1” (在 TAU0EN 位为“0”时不能读写)。→	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n (TMR0n) (确定各通道 的运行模式, 选择检测边沿。) 给定时器数据寄存器 0n (TDR0n) 设定计数值 (TDR0nH、TDR0nL 寄存器的存取步骤请参照 “6.2.2 定时器数据寄存器 0n (TDR0n)”)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置 “0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置“0”。	通道处于运行停止状态。
开始 运行	将 TS0 寄存器的对象位置“1”。→ 因为 TS0 寄存器的对象位是触发位, 所以自动返 回到“0”。	TE0 寄存器的对象位变为“1”并且开始计数。 将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n), 进入 TI0n 引脚输入的边沿检测等待状 态。
运行中	能任意更改 TDR0n 寄存器的设定值。 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄 存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 禁止更改 TMR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。	每当检测到 TI0n 引脚的输入边沿时, 计数器 (TCR0n) 就进行递减计数。如果计数到“0000H”, 就再次将 TDR0n 寄存器的值装入 TCR0n 寄存器, 并且继续进行计数。当检测到 TCR0n 为“0000H” 时, 产生 INTTM0n。 此后, 重复此运行。
停止 运行	将 TT0 寄存器的对象位置“1”。→ 因为 TT0 寄存器的对象位是触发位, 所以自动返 回到“0”。	TE0 寄存器的对象位变为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。
TAU 停止	将 PER0 寄存器的 TAU0EN 位置“0”。→	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

重新开始运行

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

6.8.3 作为分频器的运行（只限于通道 0 和通道 3）

能对 TI0n 引脚输入的时钟进行分频并且用作 TO0n 引脚时钟输出的分频器。
TO0n 输出的分频时钟频率能用以下计算式进行计算：

- 选择上升沿或者下降沿的情况：
分频时钟频率 = 输入时钟频率 / {(TDR0n 的设定值 + 1) × 2}
- 选择双边沿的情况：
分频时钟频率 ≈ 输入时钟频率 / (TDR0n 的设定值 + 1)

在间隔定时器模式中，定时器计数寄存器 0n（TCR0n）用作递减计数器。

在将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS0n）置“1”后，通过检测到 TI0n 的有效边沿将定时器数据寄存器 0n（TDR0n）的值装入 TCR0n 寄存器。此时，如果定时器模式寄存器 0n（TMR0n）的 MD00n 位为“0”，就不输出 INTTM0n 并且 TO0n 不进行交替输出；如果 TMR0n 寄存器的 MD00n 位为“1”，就输出 INTTM0n 并且 TO0n 进行交替输出。

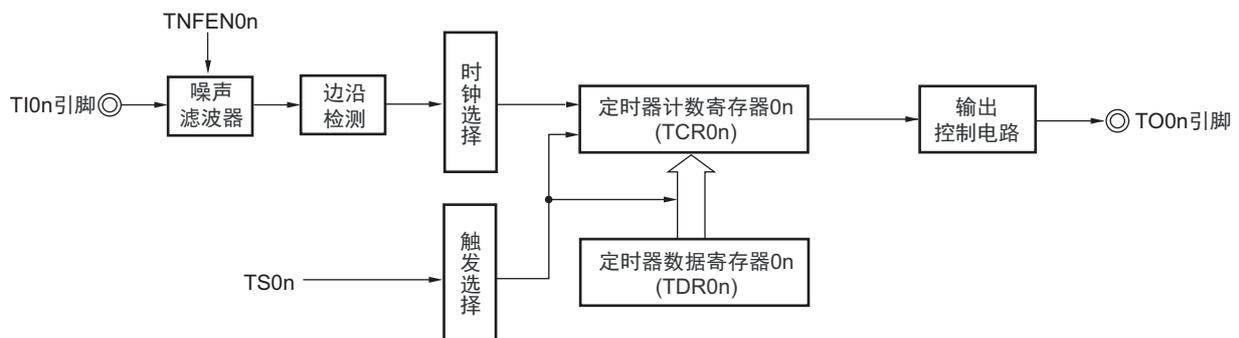
然后，TCR0n 寄存器通过 TI0n 引脚输入的有效边沿进行递减计数。如果 TCR0n 变为“0000H”，TO0n 就进行交替输出。同时，将 TDR0n 寄存器的值装入 TCR0n 寄存器，并且继续进行计数。

如果选择 TI0n 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO0n 输出的分频时钟周期。TO0n 的输出时钟周期包含最大 1 个运行时钟（ f_{MCK} ）周期的采样误差。

TO0n 输出的时钟周期 = 理想的 TO0n 输出时钟周期 ± 运行时钟周期（误差）

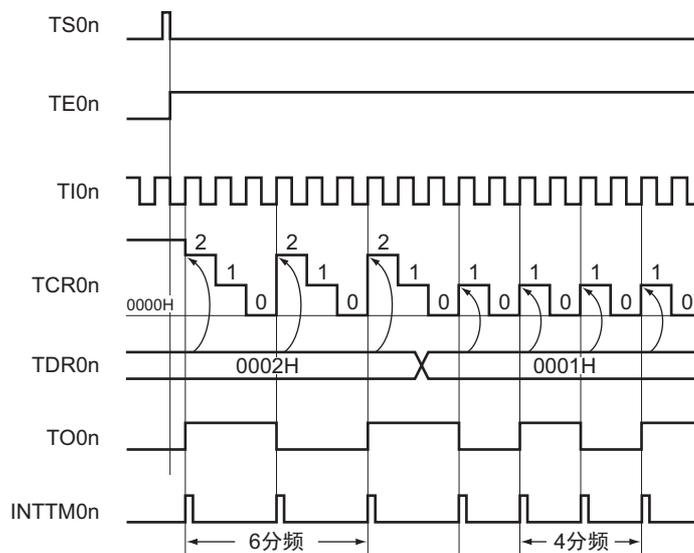
能随时改写 TDR0n 寄存器，改写的 TDR0n 寄存器的值在下一个计数期间有效。

图 6-48 作为分频器运行的框图



备注 n=0、3

图 6-49 作为分频器运行的基本时序例子 (MD00n=1)



备注 n: 通道号。n=0 (10 引脚产品), n=0、3 (16 引脚产品)

TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n

TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n

TI0n: TI0n 引脚输入信号

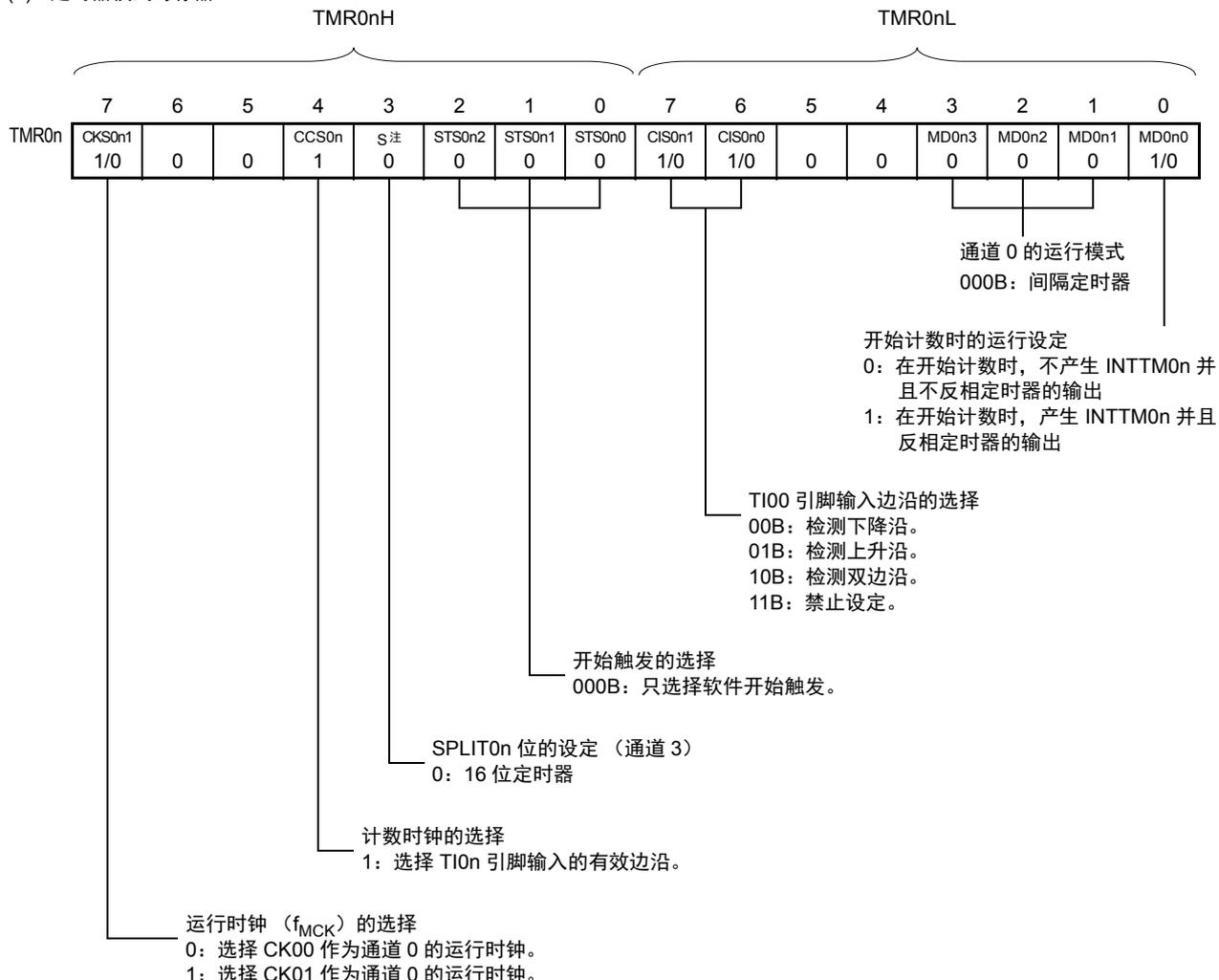
TCR0n: 定时器计数寄存器 0n (TCR0n)

TDR0n: 定时器数据寄存器 0n (TDR0n)

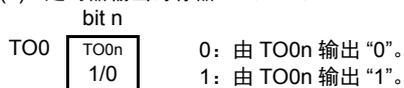
TO0n: TO0n 引脚输出信号

图 6-50 分频器的寄存器设定内容例子

(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



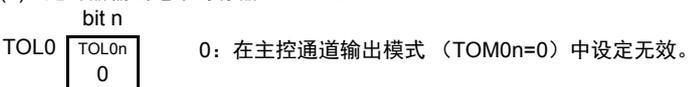
(b) 定时器输出寄存器 0 (TO0)



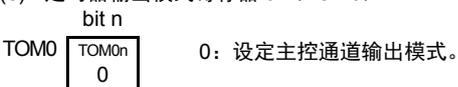
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR03: SPLIT03 位
TMR00: 固定为“0”

备注 n 通道号。n=0 (10 引脚产品), n=0、3 (16 引脚产品)

图 6-51 分频器的操作步骤

	软件操作	硬件状态
TAU 初始 设定	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1” (在 TAU0EN 位为“0”时不能读写)。	断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n (TMR0n) (确定各通道的运行模式, 选择检测边沿。) 给定时器数据寄存器 0n (TDR0n) 设定间隔 (周期) 值 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 设定 TO0n 位并且确定 TO0n 输出的初始电平。 将 TOE0n 位置“1”, 允许 TO0n 的运行。	通道处于运行停止状态。 TO0n 引脚处于 Hi-Z 输出状态。 (端口模式寄存器为输入模式)
	将端口寄存器和端口模式寄存器置“0” (设定输出模式)。	因为通道处于运行停止状态, 所以 TO0n 不变 (即使改写 TO0p 位也不反映到 TO0p 引脚)。 从 TO0n 引脚输出 TO0n 位设定的电平。
开始 运行	将 TOE0n 位置“1”, 允许 TO0n 的运行 (只限于重新开始运行时)。 将 TS0 寄存器的对象位置“1”。 因为 TS0 寄存器的对象位是触发位, 所以自动返回到“0”。	TE0 寄存器的对象位变为“1”并且开始计数。 将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n)。当 TMR0nL 寄存器的 MD0n0 位为“1”时, 产生 INTTM0n 并且 TO0n 进行交替输出。
	运行中	计数器 (TCR0n) 进行递减计数。如果计数到“0000H”, 就再次将 TDR0n 寄存器的值装入 TCR0n 寄存器, 并且继续进行计数。当检测到 TCR0n 为“0000H”时, 产生 INTTM0n 并且 TO0n 进行交替输出。 此后, 重复此运行。
停止 运行	将 TT0 寄存器的对象位置“1”。 因为 TT0 寄存器的对象位是触发位, 所以自动返回到“0”。	TE0 寄存器的对象位为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。 TO0n 输出不被初始化而保持状态。
	将 TOE0n 位置“0”并且给 TO0n 位设定值。	从 TO0n 引脚输出 TO0n 位设定的电平。
TAU 停止	要保持 TO0n 引脚的输出电平的情况: 在给端口寄存器设定要维持的值 (输出锁存器) 后将 TO0n 位置“0”。	通过端口功能保持 TO0n 引脚的输出电平。
	将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

重新开始运行

备注 n: 通道号。n=0 (10 引脚产品), n=0、3 (16 引脚产品)

6.8.4 作为输入脉冲间隔测量的运行

能通过 TI0n 引脚输入的有效边沿检测捕捉计数值，并且测量 TI0n 引脚输入脉冲的间隔。另外，在 TE0n 为“1”的期间，如果通过软件将 TS0n 位置“1”，就能捕捉计数值。

如果要进行 UART0 的波特率校正，必须将输入切换控制寄存器（ISC）的 bit1（ISC1）置“1”。

在之后的说明中，用“RXD0”代替“TI0n”。如果给 ISC1 位置“1”，串行数据输入（RXD0）引脚的输入信号就作为定时器输入（TI01）被选择。使用输入脉冲间隔测量模式，能将开始位的输入边沿信号作为触发，并且测量通信方的波特率（传送速率）宽度。

输入脉冲间隔能用以下计算式进行计算：

$$\text{TI0n 输入脉冲间隔} = \text{计数时钟的周期} \times ((10000\text{H} \times \text{TSR0n:OVF}) + (\text{TDR0n 的捕捉值} + 1))$$

注意 因为通过定时器模式寄存器 0n（TMR0n）的 CKS0n1 位选择的运行时钟（ f_{MCK} ）对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟（ f_{MCK} ）的误差。

在捕捉模式中，定时器计数寄存器 0n（TCR0n）用作递增计数器。

如果将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS0n）置“1”，TCR0n 寄存器就通过计数时钟从“0000H”开始递增计数。

如果检测到 TI0n 引脚输入的有效边沿，就将 TCR0n 寄存器的计数值传送（捕捉）到定时器数据寄存器 0n（TDR0n），同时将 TCR0n 寄存器清“0000H”，然后输出 INTTM0n。此时，如果计数器发生上溢，就将定时器状态寄存器 0n（TSR0n）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。

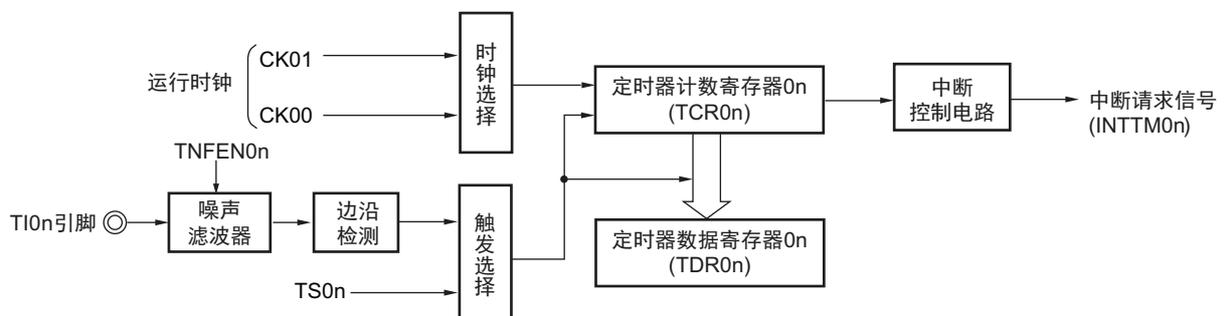
在将计数值捕捉到 TDR0n 寄存器的同时，根据在测量期间是否发生上溢，更新 TSR0n 寄存器的 OVF 位，并且能确认捕捉值的上溢状态。

即使计数器进行了 2 个或者 2 个周期以上的完整计数，也认为发生上溢而将 TSR0n 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 TMR0n 寄存器的 STS0n2 ~ STS0n0 位置“001B”，将 TI0n 的有效边沿用于开始触发和捕捉触发。

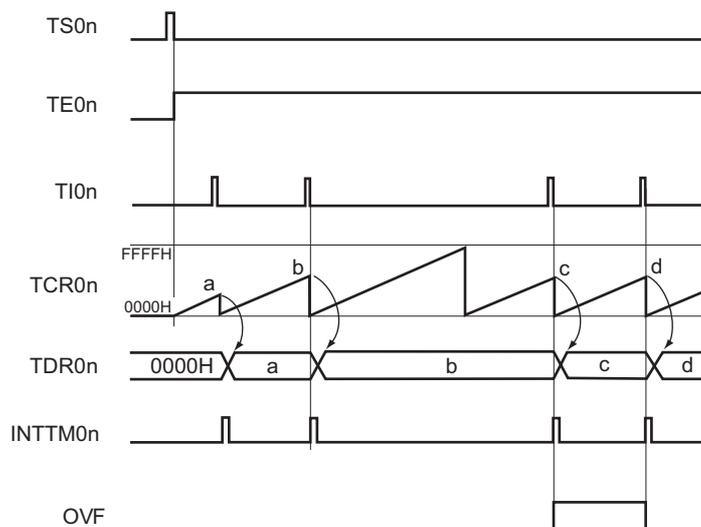
不使用 TI0n 引脚输入时，能通过软件操作（TS0n=1）置为捕捉触发。在 TE0n 为“1”的期间，将 TS0n 位置“1”，在与运行时钟（ f_{MCK} ）同步时捕捉计数值。

图 6-52 作为输入脉冲间隔测量运行的框图



备注 n: 通道号。n=0、1（10 引脚产品），n=0 ~ 3（16 引脚产品）

图 6-53 作为输入脉冲间隔测量运行的基本时序例子 (MD0n0=0)

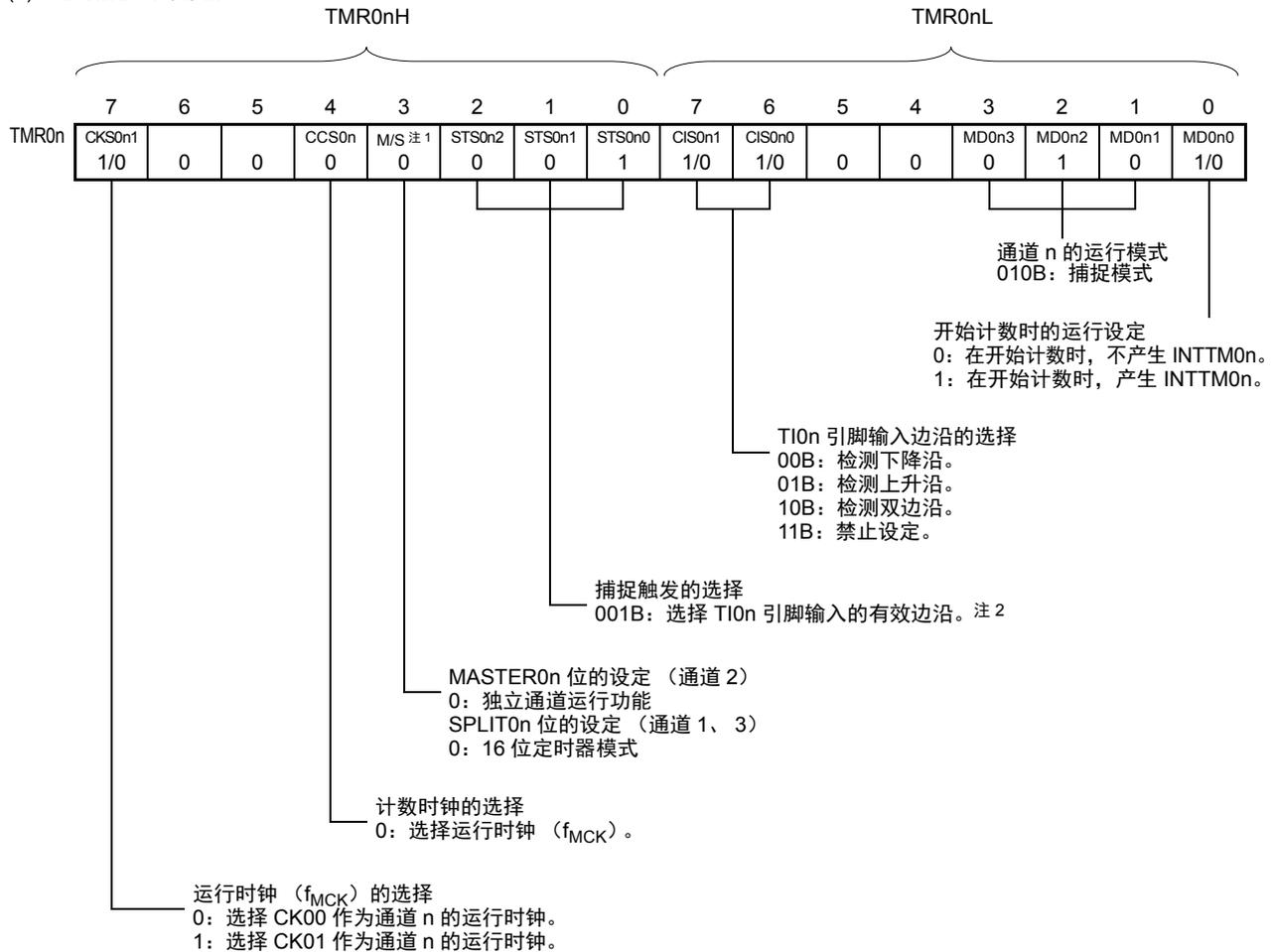


备注 1. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

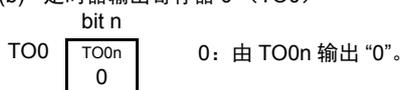
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- OVF: 定时器状态寄存器 0n (TSR0n) 的 bit0

图 6-54 输入脉冲间隔测量的寄存器设定内容例子

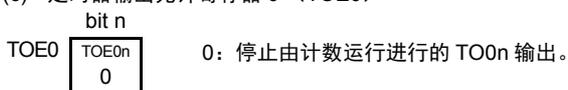
(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



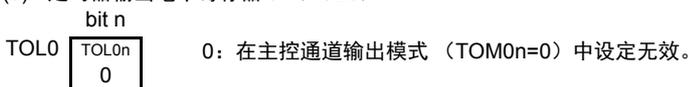
(b) 定时器输出寄存器 0 (TO0)



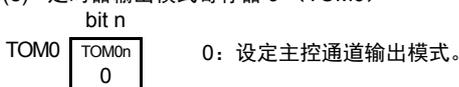
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 1. TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00: 固定为“0”。

2. 不使用 TI0n 引脚时, 能通过软件操作 (TS0n=1) 置为开始触发。

备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

图 6-55 输入脉冲间隔测量的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 的 TAU0EN 位置“1”(在 TAU0EN 位为“0”时不能读写)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n (TMR0n) (确定各通道的运行模式, 选择检测边沿)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0”(主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置“0”。	通道处于运行停止状态。
开始 运行	将 TS0 寄存器的对象位置“1”。 因为 TS0 寄存器的对象位是触发位, 所以自动返回到“0”。	TE0 寄存器的对象位变为“1”并且开始计数。 通过下一个计数时钟将定时器计数寄存器 0n (TCR0n) 清“0000H”。当 TMR0n 寄存器的 MD0n0 位为“1”时, 产生 INTTM0n。
运行中	只能更改 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位的设定值。 随时能读 TDR0n 寄存器 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 随时能读 TSR0n 寄存器。 禁止更改 TO0、TOE0、TOM0n 和 TOL0 寄存器的对象位的设定值。	计数器 (TCR0n) 从“0000H”开始递增计数, 如果检测到 TIO0n 引脚输入的有效边沿或者将 TS0n 位置“1”, 就将计数值传送 (捕捉) 到定时器数据寄存器 0n (TDR0n), 同时将 TCR0n 寄存器清“0000H”并且产生 INTTM0n。 此时, 如果发生上溢, 就将定时器状态寄存器 0n (TSR0n) 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 TT0 寄存器的对象位置“1”。 因为 TT0 寄存器的对象位是触发位, 所以自动返回到“0”。	TE0 寄存器的对象位变为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。 保持 TSR0n 寄存器的 OVF 位。
TAU 停止	将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

重新
开始
运行

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

6.8.5 作为输入信号高 / 低电平宽度测量的运行

能通过 TI0n 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 TI0n 的信号宽度（高电平宽度 / 低电平宽度）。TI0n 的信号宽度能用以下计算式进行计算。

$$\text{TI0n 输入的信号宽度} = \text{计数时钟的周期} \times ((10000\text{H} \times \text{TSR0n:OVF}) + (\text{TDR0n 的捕捉值} + 1))$$

注意 因为通过定时器模式寄存器 0n (TMR0n) 的 CKS0n1 位选择的运行时钟 (f_{MCK}) 对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟 (f_{MCK}) 的误差。

在捕捉 & 单次计数模式中，定时器计数寄存器 0n (TCR0n) 用作递增计数器。

如果将定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS0n) 置“1”，TE0n 位就变为“1”，并且进入 TI0n 引脚的开始边沿检测等待状态。

如果检测到 TI0n 引脚输入的开始边沿（在测量高电平宽度时为 TI0n 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 TI0n 引脚输入的下落沿），就在将计数值传送到定时器数据寄存器 0n (TDR0n) 的同时，输出 INTTM0n。此时，如果计数器发生上溢，就将定时器状态寄存器 0n (TSR0n) 的 OVF 位置位。如果计数器没有发生上溢，就清除 OVF 位。TCR0n 寄存器的值变为“传送到 TDR0n 寄存器的值 +1”而停止计数，并且进入 TI0n 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 TDR0n 寄存器的同时，根据在测量期间是否发生上溢，更新 TSR0n 寄存器的 OVF 位，并且能确认捕捉值的上溢状态。

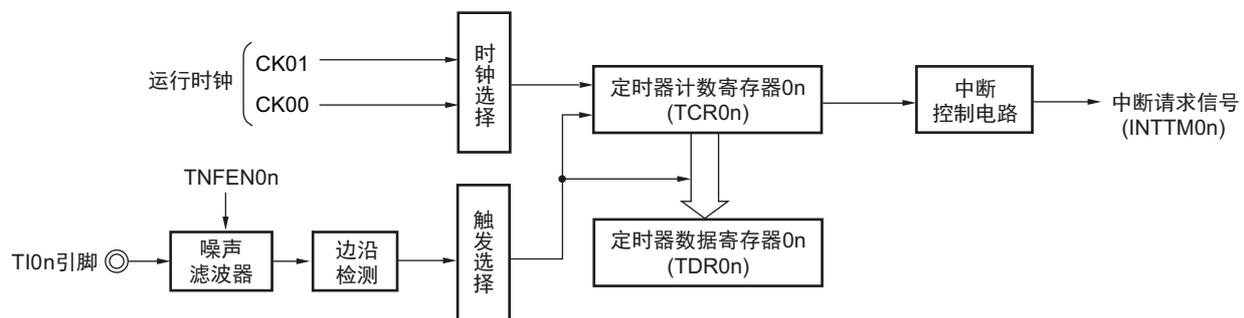
即使计数器进行了 2 个或者 2 个周期以上的完整计数，也认为发生上溢而将 TSR0n 寄存器的 OVF 位置“1”。但是，在发生 2 次或者 2 次以上的上溢时，无法通过 OVF 位正常测量间隔值。

能通过 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位来设定是测量 TI0n 引脚的高电平宽度还是低电平宽度。此功能是以测量 TI0n 引脚的输入信号宽度为目的，因此不能在 TE0n 位为“1”的期间将 TS0n 位置“1”。

TMR0n 寄存器的 CIS0n1、CIS0n0=10B：测量低电平宽度。

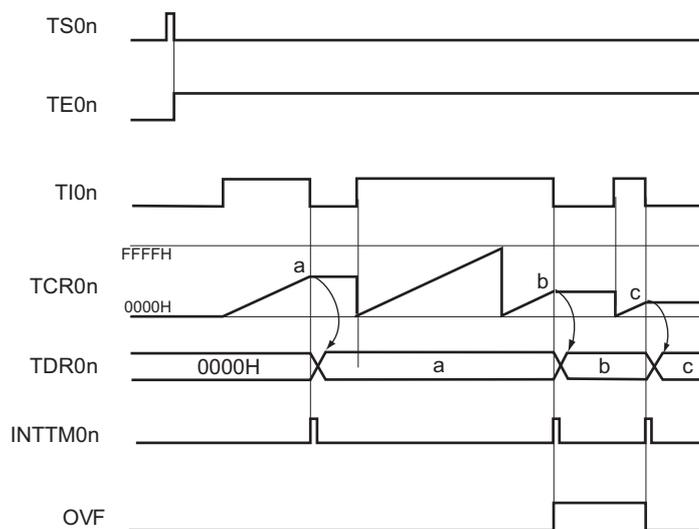
TMR0n 寄存器的 CIS0n1、CIS0n0=11B：测量高电平宽度。

图 6-56 作为输入信号高 / 低电平宽度测量运行的框图



备注 n = 通道号。n=0、1（10 引脚产品），n=0 ~ 3（16 引脚产品）

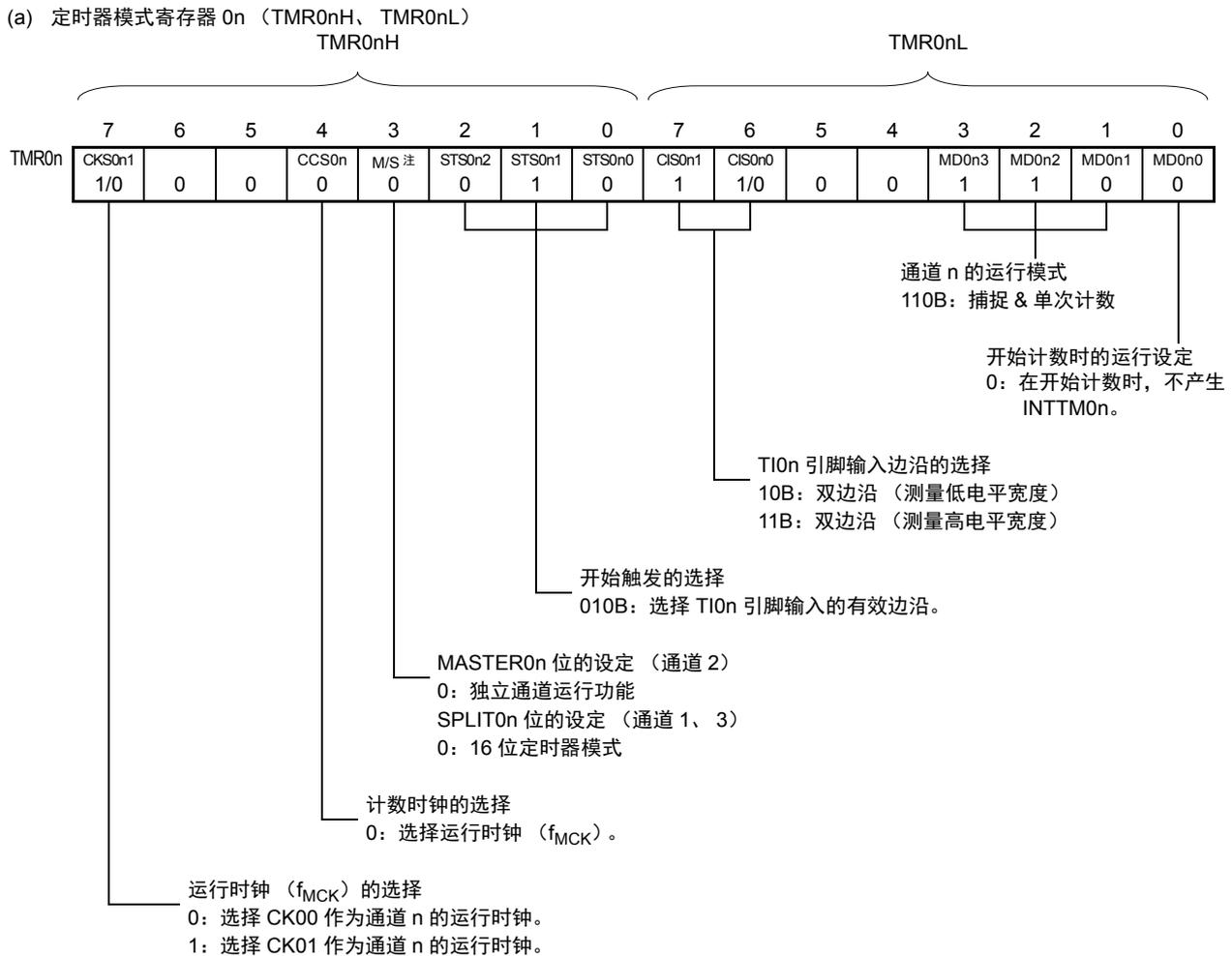
图 6-57 作为输入信号高 / 低电平宽度测量运行的基本时序例子



备注 1. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- OVF: 定时器状态寄存器 0n (TSR0n) 的 bit0

图 6-58 输入信号高 / 低电平宽度测量的寄存器设定内容例子



(b) 定时器输出寄存器 0 (TO0)

bit n
TO0

TO0n
0

 0: 由 TO0n 输出 "0"。

(c) 定时器输出允许寄存器 0 (TOE0)

bit n
TOE0

TOE0n
0

 0: 停止由计数运行进行的 TO0n 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

bit n
TOL0

TOL0n
0

 0: 在主导通道输出模式 (TOM0n=0) 中置 "0"。

(e) 定时器输出模式寄存器 0 (TOM0)

bit n
TOM0

TOM0n
0

 0: 设定主导通道输出模式。

注 TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00: 固定为 "0"。

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

图 6-59 输入信号高 / 低电平宽度测量的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 “1” (在 TAU0EN 位为“0”时不能读写)。—————▶	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n (TMR0n) (确定各通道 的运行模式, 选择检测边沿)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置 “0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置 “0”。	通道处于运行停止状态。
开始 运行	将 TS0 寄存器的对象位置“1”。—————▶ 因为 TS0 寄存器的对象位是触发位, 所以自动返 回到“0”。	TE0 寄存器的对象位变为“1”并且进入 TI0n 引脚的 开始边沿检测等待状态。
	检测 TI0n 引脚输入的计数开始边沿。—————▶	将定时器计数寄存器 0n (TCR0n) 清“0000H”并 且开始递增计数。
运行中	随时能读 TDR0n 寄存器 (TDR0nH、TDR0nL 寄 存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄 存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 随时能读 TSR0n 寄存器。 禁止更改 TMR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。	在检测到 TI0n 引脚的开始边沿后, 计数器 (TCR0n) 从“0000H”开始递增计数。如果检测到 TI0n 引脚的捕 捉边沿, 就将计数值传送到定时器数据寄存器 0n (TDR0n), 并且产生 INTTM0n。 此时, 如果发生上溢, 就将定时器状态寄存器 0n (TSR0n) 的 OVF 位置位。如果没有发生上溢, 就清 除 OVF 位。TCR0n 寄存器在检测到下一个 TI0n 引脚 的开始边沿前停止计数。 此后, 重复此运行。
停止 运行	将 TT0 寄存器的对象位置“1”。—————▶ 因为 TT0 寄存器的对象位是触发位, 所以自动返 回到“0”。	TE0 寄存器的对象位变为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。 保持 TSR0n 寄存器的 OVF 位。
TAU 停止	将 PER0 寄存器的 TAU0EN 位置“0”。—————▶	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

重新开始运行

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

6.8.6 作为延迟计数器的运行

能通过 TI0n 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中断请求信号（INTTM0n）。

在 TE0n 位为“1”的期间，能通过软件将 TS0n 位置“1”，开始递减计数，并且以任意的设定间隔产生 INTTM0n。

中断请求信号（INTTM0n）的产生周期能用以下计算式进行计算：

$$\text{中断请求信号 (INTTM0n) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR0n 的设定值} + 1)$$

注意 因为通过定时器模式寄存器 0n（TMR0n）的 CKS0n1 位选择的运行时钟（ f_{MCK} ）对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟（ f_{MCK} ）的误差。

在通道 1 和通道 3 用作 8 位定时器功能（SPLIT0n=1）时，只有低 8 位定时器能用作延迟计数器。

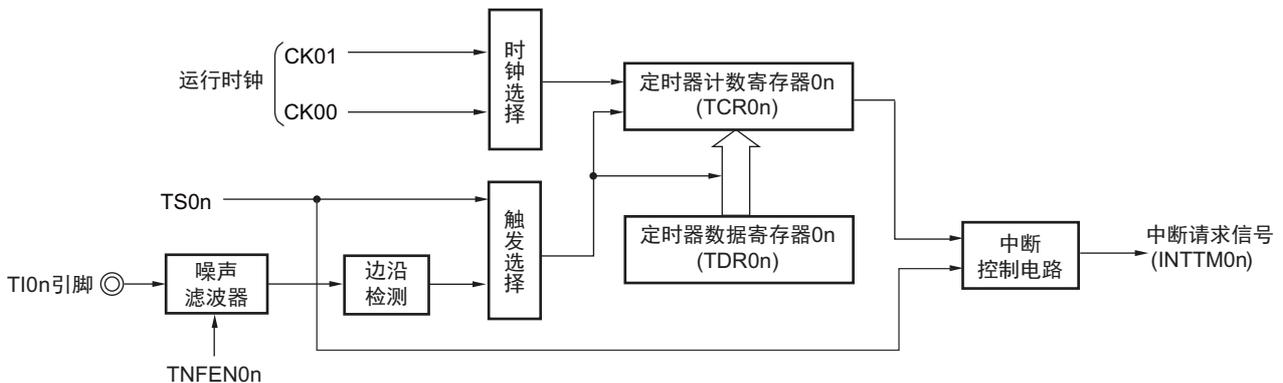
在单次计数模式中，定时器计数寄存器 0n（TCR0n）用作递减计数器。

如果将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS0n）置“1”，TE0n 位就变为“1”，并且进入 TI0n 引脚的有效边沿检测等待状态。

通过 TI0n 引脚输入的有效边沿检测，开始 TCR0n 寄存器的运行，并且装入定时器数据寄存器 0n（TDR0n）的值。TCR0n 寄存器通过计数时钟，从装入的 TDR0n 寄存器的值开始递减计数。如果 TCR0n 变为“0000H”，就输出 INTTM0n，并且在检测到下一个 TI0n 引脚输入的有效边沿前，TCR0p 为“FFFFH”而且停止计数。

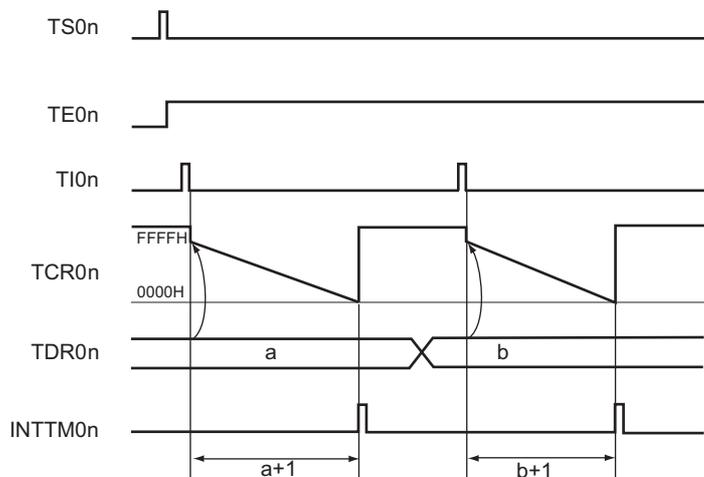
能随时改写 TDR0n 寄存器，改写的 TDR0n 寄存器的值从下一个周期开始有效。

图 6-60 作为延迟计数器运行的框图



备注 n: 通道号。n=0、1（10 引脚产品），n=0 ~ 3（16 引脚产品）

图 6-61 作为延迟计数器运行的基本时序例子

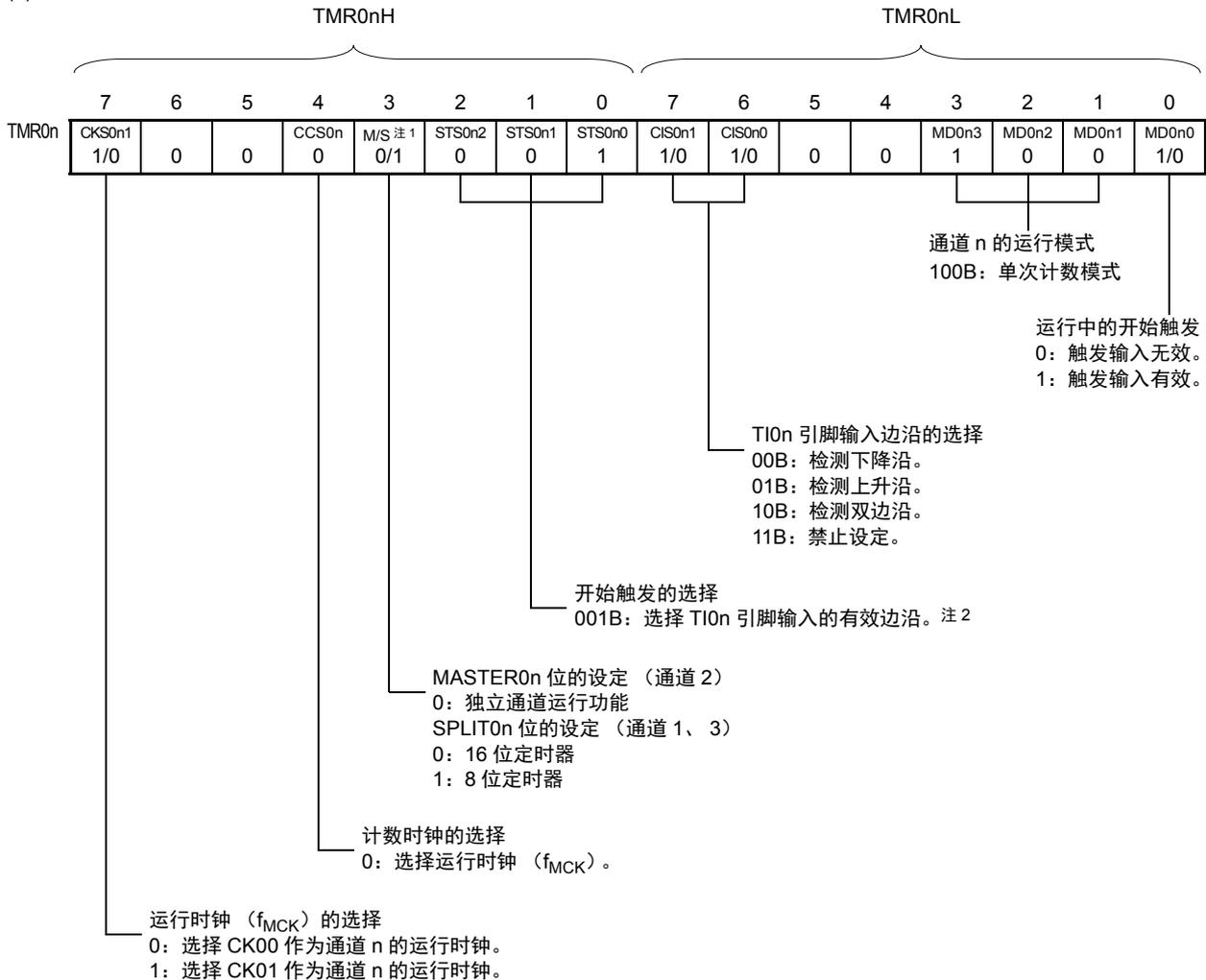


备注 1. n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

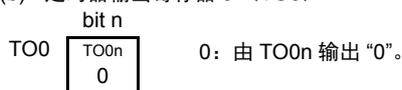
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)

图 6-62 延迟计数器的寄存器设定内容例子

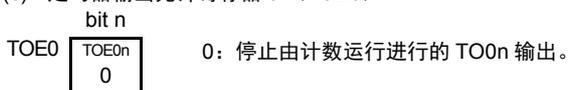
(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



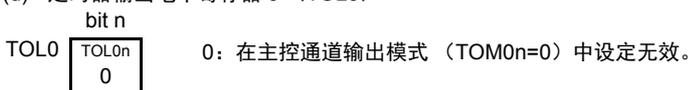
(b) 定时器输出寄存器 0 (TO0)



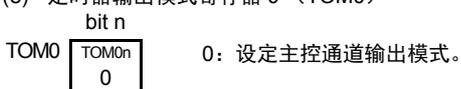
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



- 注 1. TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00: 固定为“0”。
2. 不使用 TI0n 引脚输入时, 能通过软件操作 (TS0n=1) 置为开始触发。
- 备注 n: 通道号。n=0、1 (10 引脚产品), n=0~3 (16 引脚产品)

图 6-63 延迟计数器的操作步骤

	软件操作	硬件状态
TAU 初始 设定	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1” (在 TAU0EN 位为“0”时不能读写)。	断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n (TMR0n) (确定各通道的运行模式, 选择检测边沿)。 给定时器数据寄存器 0n (TDR0n) 设定延迟时间 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置“0”。	通道处于运行停止状态。
开始 运行	将 TS0 寄存器的对象位置“1”。	TE0 寄存器的对象位变为“1”并且进入开始触发检测 (检测到 TI0n 引脚的有效边沿或者将 TS0n 位置“1”) 等待状态。
	通过检测到以下开始触发, 开始计数。 • 检测到 TI0n 引脚输入的有效边沿 • 通过软件将 TS0n 位置“1”	将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n), 并且开始递减计数。
运行中	能任意更改 TDR0n 寄存器的设定值。 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 禁止更改 TMR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。	计数器 (TCR0n) 进行递减计数。如果 TCR0n 计数到“0000H”, 就产生 INTTM0n, 直到下一个开始触发检测 (检测到 TI0n 引脚输入的有效边沿或者将 TS0n 位置“1”) 前 TCR0n 为“FFFFH”而且停止计数。 此后, 重复此运行。
停止 运行	将 TT0 寄存器的对象位置“1”。	TE0 寄存器的对象位变为“0”并且停止计数。 TCR0n 寄存器保持计数值而停止计数。
TAU 停止	将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 (停止提供时钟, TAU 的 SFR 被初始化。)

重新
开始
运行

备注 n: 通道号。n=0、1 (10 引脚产品), n=0 ~ 3 (16 引脚产品)

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的运行

将 2 个通道成对使用，能通过 TI0n 引脚输入生成具有任意延迟（输出延迟时间）的单触发脉冲。在 TE0n 位为“1”的期间，能通过软件将 TS0n 位置“1”，开始递减计数。

延迟和单触发脉宽能用以下计算式进行计算：

$\text{延迟} = \{\text{TDR0n (主控) 的设定值} + 2\} \times \text{计数时钟周期}$ $\text{单触发脉宽} = \{\text{TDR0p (从属) 的设定值}\} \times \text{计数时钟周期}$
--

注意 因为通过定时器模式寄存器 0n (TMR0n) 的 CKS0n1 位选择的运行时钟 (f_{MCK}) 对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟 (f_{MCK}) 的误差。

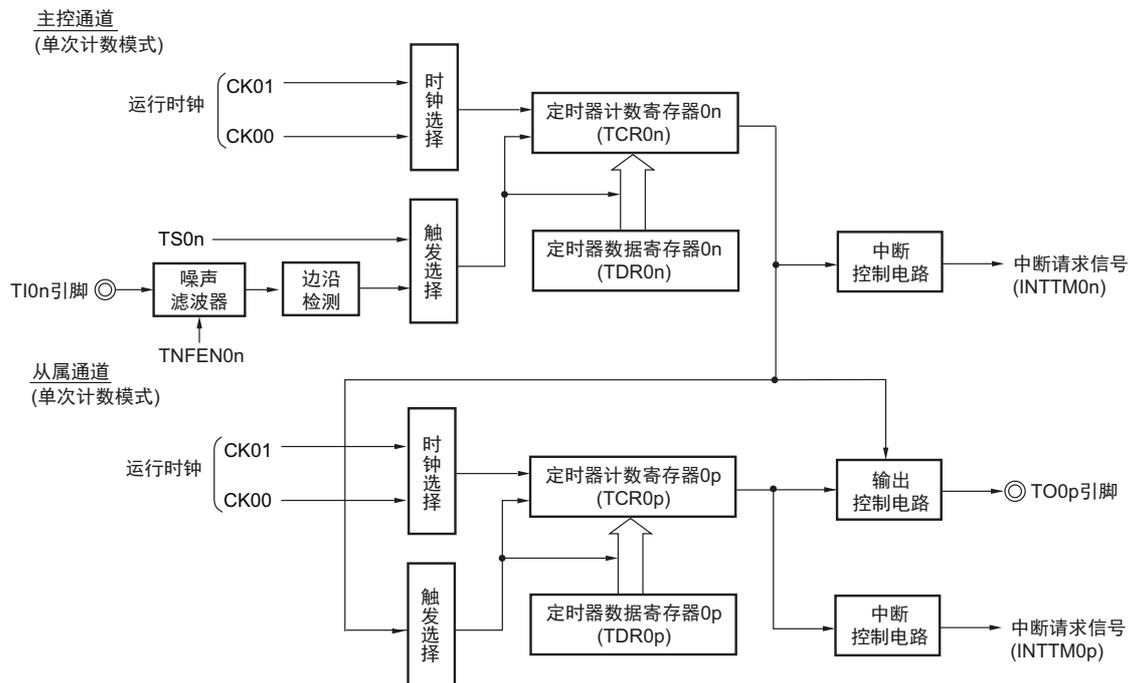
主控通道以单次计数模式运行并且对延迟进行计数。通过开始触发的检测，主控通道的定时器计数寄存器 0n (TCR0n) 开始运行并且装入定时器数据寄存器 0n (TDR0n) 的值。TCR0n 寄存器通过计数时钟 (f_{TCLK})，从装入的 TDR0n 寄存器的值开始递减计数。如果 TCR0n 变为“0000H”，就输出 INTTM0n，并且在检测到下一个开始触发前停止计数。

从属通道以单次计数模式运行并且对单触发脉宽进行计数。将主控通道的 INTTM0n 作为开始触发，从属通道的 TCR0p 寄存器开始运行并且装入 TDR0p 寄存器的值。TCR0p 寄存器通过计数时钟 (f_{TCLK})，从装入的 TDR0p 寄存器值开始递减计数。如果 TCR0p 变为“0000H”，就输出 INTTM0p，并且在检测到下一个开始触发（主控通道的 INTTM0n）前 TCR0p 为“FFFFH”而且停止计数。在从主控通道产生 INTTM0n 并且经过 1 个计数时钟 (f_{TCLK}) 后，TO0p 的输出电平变为有效电平，如果 TCR0p 变为“0000H”，就变为无效电平。

注意 因为主控通道的定时器数据寄存器 0n (TDR0n) 和从属通道的 TDR0p 寄存器的装入时序不同，所以如果在运行中改写 TDR0n 寄存器和 TDR0p 寄存器，就有可能输出不正常的波形。必须在产生 INTTM0n 后改写 TDR0n 寄存器，并且在产生 INTTM0p 后改写 TDR0p 寄存器。

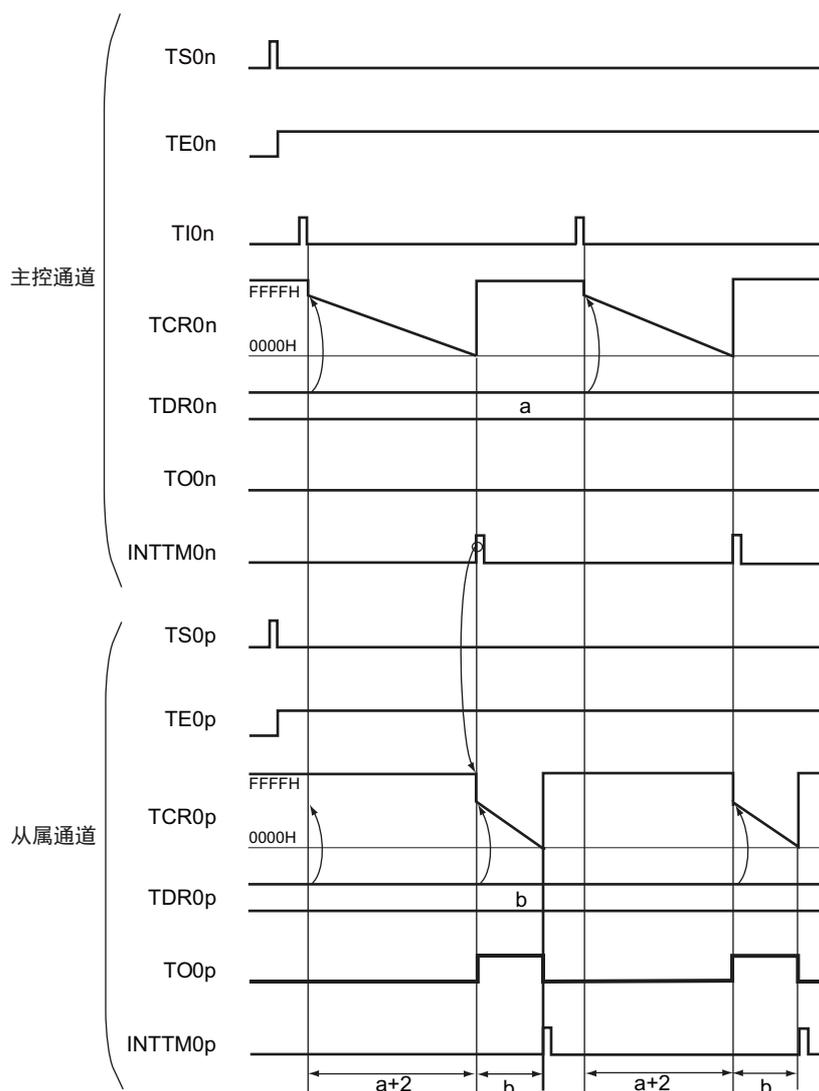
备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

图 6-64 作为单触发脉冲输出运行的框图



备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

图 6-65 作为单触发脉冲输出运行的基本时序例子



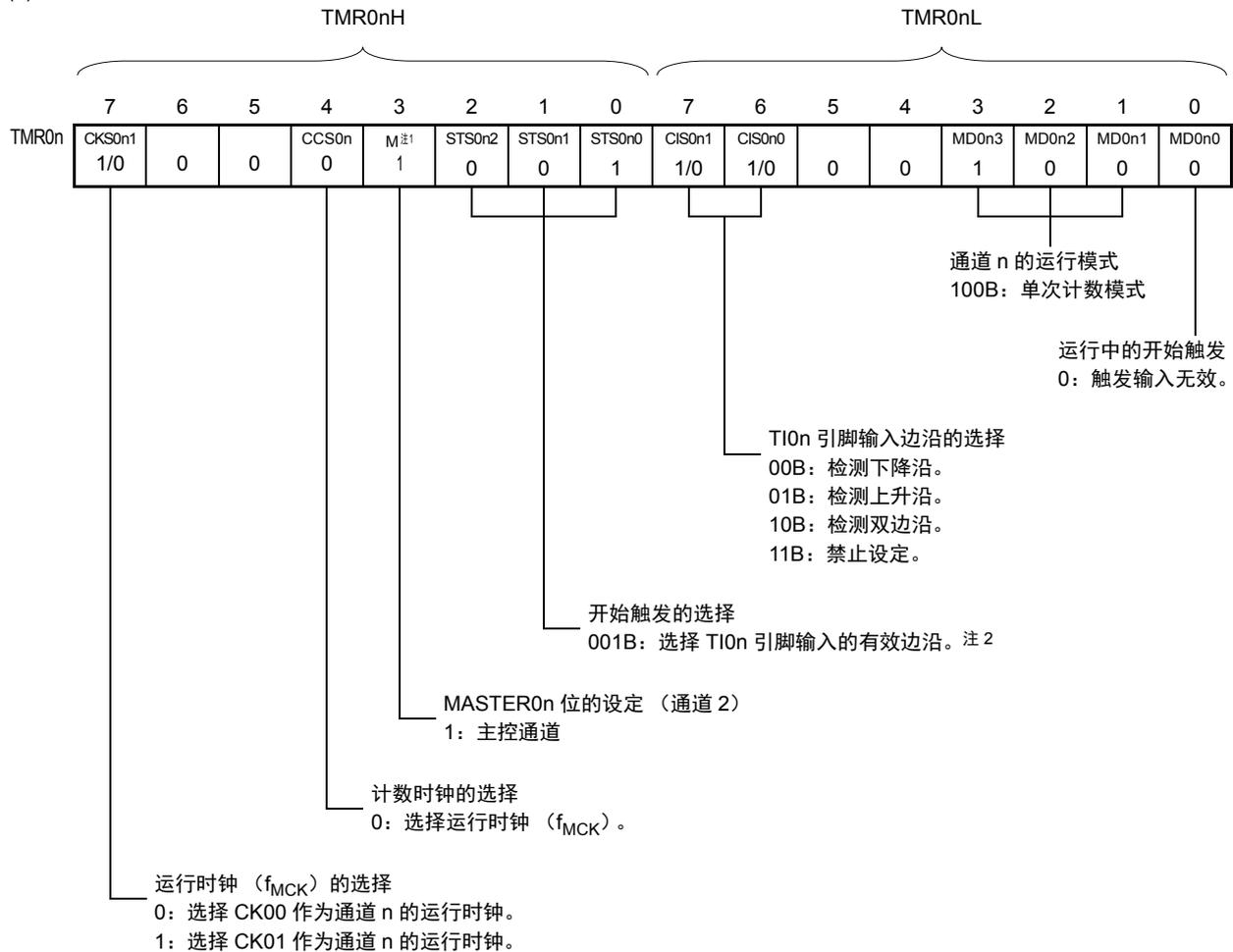
备注 1. n: 主控通道号 (n=0、2)

p: 从属通道号 (n < p ≤ 3)

2. TS0n、TS0p: 定时器通道开始寄存器 0 (TS0) 的 bit n、p
- TE0n、TE0p: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p
- TI0n、TI0p: TI0n 引脚和 TI0p 引脚的输入信号
- TCR0n、TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)
- TDR0n、TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)
- TO0n、TO0p: TO0n 引脚和 TO0p 引脚的输出信号

图 6-66 单触发脉冲输出（主控通道）的寄存器设定内容例子

(a) 定时器模式寄存器 0n (TMR0nH、TMR0nL)



(b) 定时器输出寄存器 0 (TO0)

bit n

TO0

TO0n
0

 0: 由 TO0n 输出“0”。

(c) 定时器输出允许寄存器 0 (TOE0)

bit n

TOE0

TOE0n
0

 0: 停止由计数运行进行的 TO0n 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

bit n

TOL0

TOL0n
0

 0: 在主控通道输出模式 (TOM0n=0) 中设定无效。

(e) 定时器输出模式寄存器 0 (TOM0)

bit n

TOM0

TOM0n
0

 0: 设定主控通道输出模式。

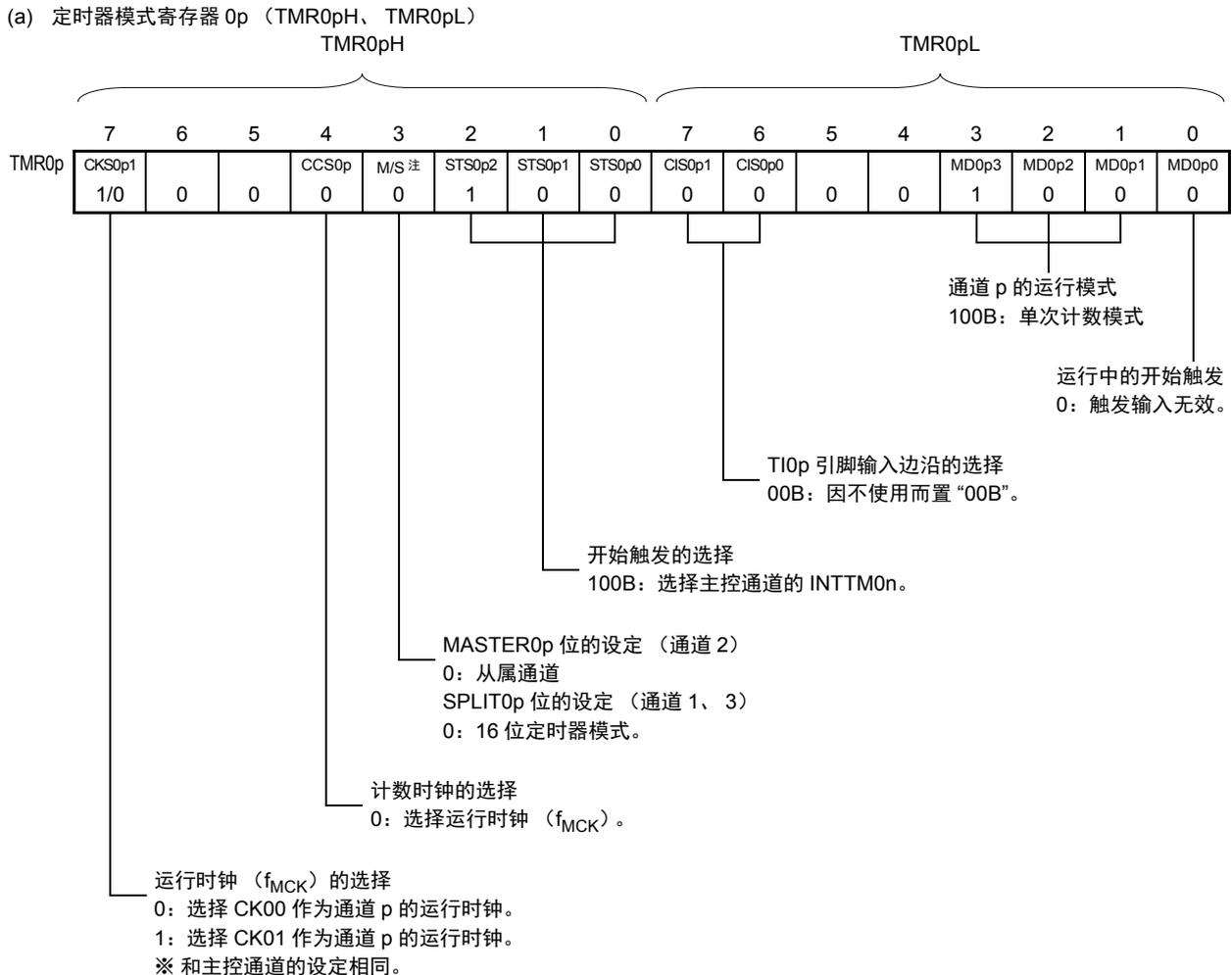
注 1. TMR02: MASTER02 位

TMR00: 固定为“0”

2. 不使用 TI0n 引脚时, 能通过软件操作 (TS0n=1) 置为开始触发。

备注 n: 主控通道号 (n=0、2)

图 6-67 单触发脉冲输出（从属通道）的寄存器设定内容例子



(b) 定时器输出寄存器 0 (TO0)

bit p	
TO0p	0: 由 TO0p 输出“0”。 1: 由 TO0p 输出“1”。
1/0	

(c) 定时器输出允许寄存器 0 (TOE0)

bit p	
TOE0p	0: 停止由计数运行进行的 TO0p 输出 (从 TO0p 引脚输出 TO0p 位设定的电平)。
1/0	1: 允许由计数运行进行的 TO0p 输出 (从 TO0p 引脚交替输出)。

(d) 定时器输出电平寄存器 0 (TOL0)

bit p	
TOL0p	0: 正逻辑输出 (高电平有效)
1/0	1: 负逻辑输出 (低电平有效)

(e) 定时器输出模式寄存器 0 (TOM0)

bit p	
TOM0p	1: 设定从属通道输出模式。
1	

注 TMR02: MASTER0n 位

TMR01、TMR03: SPLIT0p 位

备注 n: 主控通道号 (n=0、2)

p: 从属通道号 (n < p ≤ 3)

图 6-68 单触发脉冲输出的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 “1” (在 TAU0EN 位为“0”时不能读写)。————→	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	
通道初 始设定	设定定时器模式寄存器 0n、0p (TMR0n、TMR0p) (确定各通道的运行模式, 选择主控通道的检测边 沿。) 给定时器数据寄存器 0n、0p (TDR0n、TDR0p) 设 定主控通道的输出延迟时间, 并且设定从属通道的单 触发脉宽 (TDR0nH、TDR0nL 寄存器的存取步骤请 参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。	通道处于运行停止状态。
	主控通道的设定 设定噪声滤波器允许寄存器 1 (NFEN1)。 将定时器输出模式寄存器 0 (TOM0) 的对象位 置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置 “0”。 从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位置 “1” (从属通道输出模式)。 设定 TOL0 寄存器的对象位。 设定 TO0p 位并且确定 TO0p 输出的初始电平。 将 TOE0p 位置“1”, 允许 TO0p 的运行。————→ 将端口寄存器和端口模式寄存器置“0” (设定输出 模式)。————→	TO0p 引脚处于 Hi-Z 状态。 (端口模式寄存器为输入模式) 因为通道处于运行停止状态, 所以 TO0p 不变 (即 使改写 TO0p 位也不反映到 TO0p 引脚)。 从 TO0p 引脚输出 TO0p 位设定的电平。

备注 n: 主控通道号 (n=0、2)

p: 从属通道号 (n < p ≤ 3)

图 6-68 单触发脉冲输出的操作步骤 (2/2)

	软件操作	硬件状态	
重新开始运行	开始运行	<p>将从属通道的 TOE0p 位置“1”，允许 TO0p 的运行（只在重新开始运行时）。</p> <p>将 TS0 寄存器的对象位（主控、从属）同时置“1”→ 因为 TS0 寄存器的对象位是触发位，所以自动返回到“0”。</p> <p>通过检测到以下的开始触发，开始计数。→</p> <ul style="list-style-type: none"> 检测到 TI0n 引脚输入的有效边沿 通过软件将 TS0n 位置“1” 	<p>TE0 寄存器的对象位变为“1”并且主控通道进入 TI0n 引脚输入的有效边沿检测等待状态。</p> <p>将 TDR0n 寄存器的值装入主控通道的定时器计数寄存器 0n (TCR0n)，开始递减计数。</p>
	运行中	<p>更改主控通道的设定</p> <p>随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。</p> <p>只能更改 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位的设定值。</p> <p>禁止更改 TDR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。</p> <p>更改从属通道的设定</p> <p>随时能读 TCR0p 寄存器。</p> <p>能更改 TO0p、TOE0p、TOM0 和 TOL0 寄存器的对象位的设定值。</p> <p>禁止更改 TMR0p 和 TDR0p 寄存器的设定值。</p>	<p>主控通道的计数器 (TCR0n) 进行递减计数。如果 TCR0n 计数到“0000H”，就产生 INTTM0n，并且在下一个开始触发检测（检测到 TI0n 引脚输入的有效边沿或者将 TS0n 位置“1”）前 TCR0n 为“FFFFH”而且停止计数。</p> <p>从属通道将主控通道的 INTTM0n 作为开始触发，将 TDR0p 寄存器的值装入 TCR0p 寄存器并且开始递减计数。在产生主控通道的 INTTM0n 并且经过 1 个计数时钟 (f_{TCLK}) 后，将 TO0p 的输出电平置为有效电平。之后，如果 TCR0p 计数到“0000H”，就再将 TO0p 的输出电平置为无效电平，并且 TCR0p 为“FFFFH”而且停止计数。此后，重复此运行。</p>
	停止运行	<p>将 TT0 寄存器的对象位（主控、从属）同时置“1”。→</p> <p>因为 TT0 寄存器的对象位是触发位，所以自动返回到“0”。</p> <p>将从属通道的 TOE0p 位置“0”并且给 TO0p 位设定值。→</p>	<p>TE0 寄存器的对象位变为“0”并且停止计数。TCR0n 和 TCR0p 寄存器保持计数值而停止计数。TO0p 输出不被初始化而保持状态。</p> <p>从 TO0p 引脚输出 TO0p 位设定的电平。</p>
	TAU 停止	<p>要保持 TO0p 引脚的输出电平的情况</p> <p>在给端口寄存器设定要保持的值（输出锁存器）后将 TO0p 位置“0”。→</p> <p>将 PER0 寄存器的 TAU0EN 位置“0”。→</p>	<p>通过端口功能保持 TO0p 引脚的输出电平。</p> <p>断电状态 (停止提供时钟，TAU 的 SFR 被初始化。)</p>

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

6.9.2 作为双输入式单触发脉冲输出的运行

能通过双引脚输入（TI0n、TI0p 引脚输入）生成具有任意延迟（输出延迟时间）的单触发脉冲。只有 16 引脚产品能使用双输入式单触发脉冲。

延迟（输出延迟时间）和单触发脉宽能用以下计算式进行计算：

$$\begin{aligned} \text{延迟} &= \{\text{TDR0n (主控) 的设定值} + 2\} \times \text{计数时钟周期} \\ \text{单触发脉冲的有效电平宽度} &= \text{计数时钟周期} \times ((10000\text{H} \times \text{TSR0p : OVF}) + (\text{TDR0p (从属) 的捕捉值} + 1)) \end{aligned}$$

注意 因为通过定时器模式寄存器 0n（TMR0n）的 CKS0n1 位选择的运行时钟（ f_{MCK} ）对 TI0n、TI0p 引脚输入进行采样，所以产生 1 个运行时钟（ f_{MCK} ）的误差。

主控通道以单次计数模式运行。将主控通道的 TI0n 引脚输入的有效边沿检测置为开始触发后对延迟（输出延迟时间）进行计数。通过 TI0n 引脚输入检测到开始触发后，就将定时器数据寄存器 0n（TDR0n）的设定值装入定时器计数寄存器 0n（TCR0n），并且根据计数时钟（ f_{TCLK} ）进行递减计数。如果 TCR0n 变为“0000H”，就输出 INTTM0n，并且从 TO0p 引脚输出有效电平。直到检测下一个开始触发前，主控通道停止计数。

从属通道以捕捉模式运行。将从属通道的 TI0p 引脚输入的有效边沿检测作为结束触发并且将单触发脉冲置为无效电平。通过 TI0p 引脚输入检测到结束触发后，就将 TCR0p 寄存器的计数值传送（捕捉）到 TDR0p 寄存器，并且将 TCR0p 寄存器清“0000H”。与此同时，输出 INTTM0p 并且从 TO0p 引脚输出无效电平。此时，如果发生计数器的上溢，定时器状态寄存器 0p（TSR0p）的 OVF 位就被置“1”，如果没有发生上溢，就清除 OVF 位。此后，持续同样的运行。

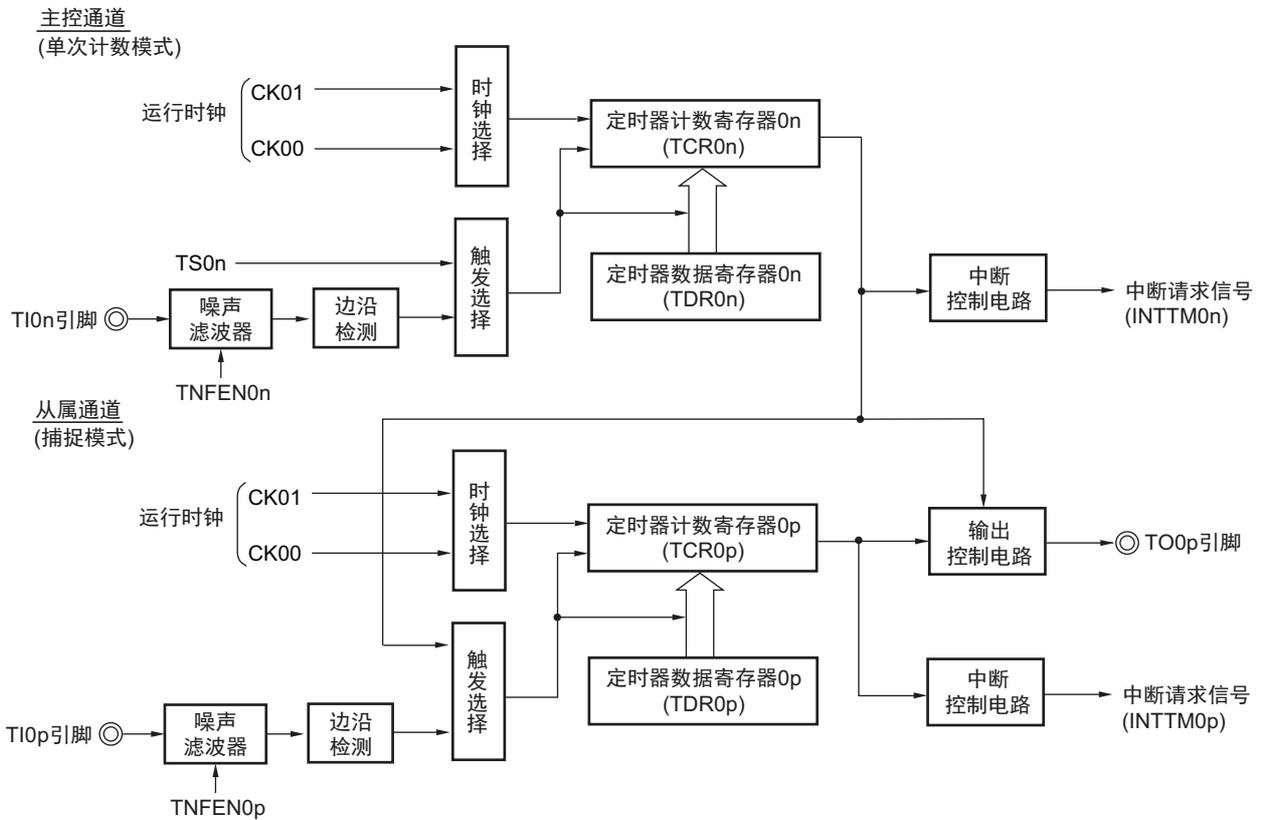
在计数值被捕捉到 TDR0p 寄存器的同时，能根据有无有效电平期间的上溢，更新 TSR0p 寄存器的 OVF 位，并且确认捕捉值的上溢状态。

即使计数器进行了 2 个或者 2 个周期以上的完整计数，也认为发生上溢而将 TSR0p 寄存器的 OVF 位置“1”。OVF 位在发生至少 2 次的上溢后，就无法测量正常的间隔值。

不使用 TI0n 引脚输入进行主控通道的开始触发检测时，通过软件操作（TS0n=1）置为开始触发。

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (p=3)

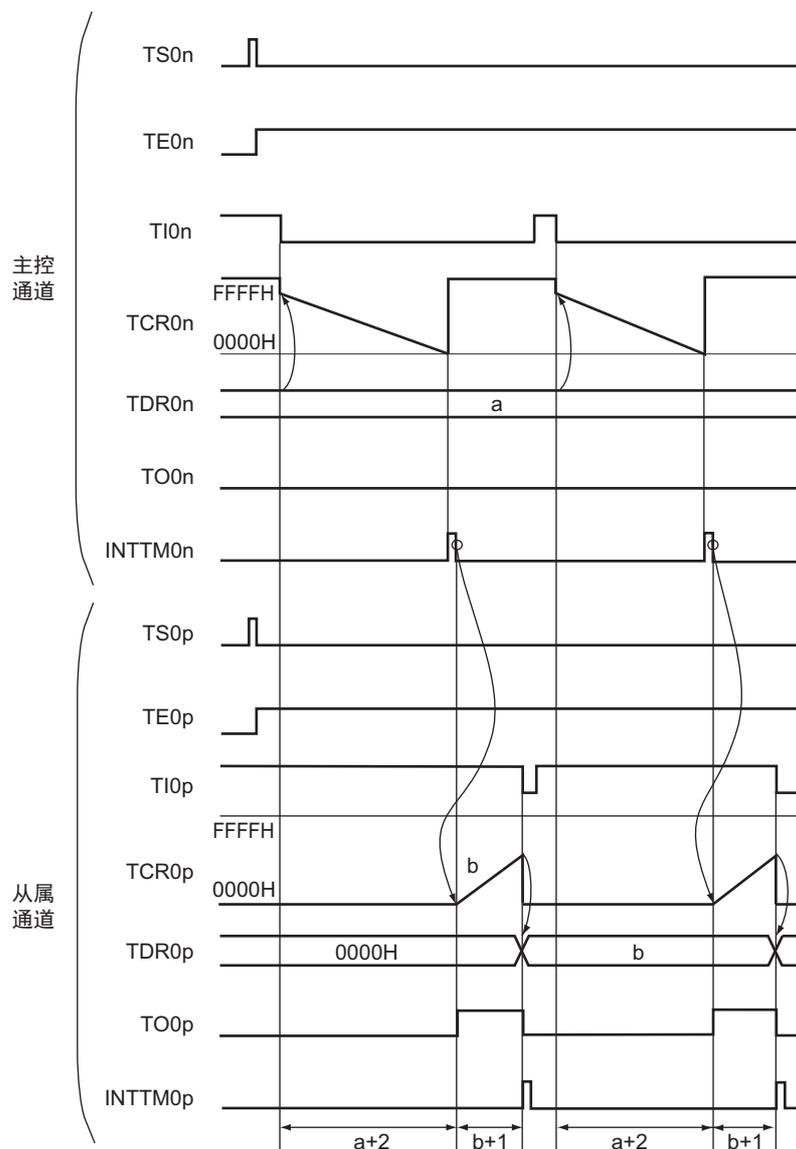
图 6-69 作为双输入式单触发脉冲输出运行的框图



备注 n: 主控通道号 (n=0、2)

p: 从属通道号 (p=3)

图 6-70 作为双输入式单触发脉冲输出运行的基本时序例子



备注 1. n: 主控通道号 (n=0、2)

p: 从属通道号 (p=3)

- TS0n、TS0p: 定时器通道开始寄存器 0 (TS0) 的 bit n、p

TE0n、TE0p: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p

TI0n、TI0p: TI0n 引脚和 TI0p 引脚的输入信号

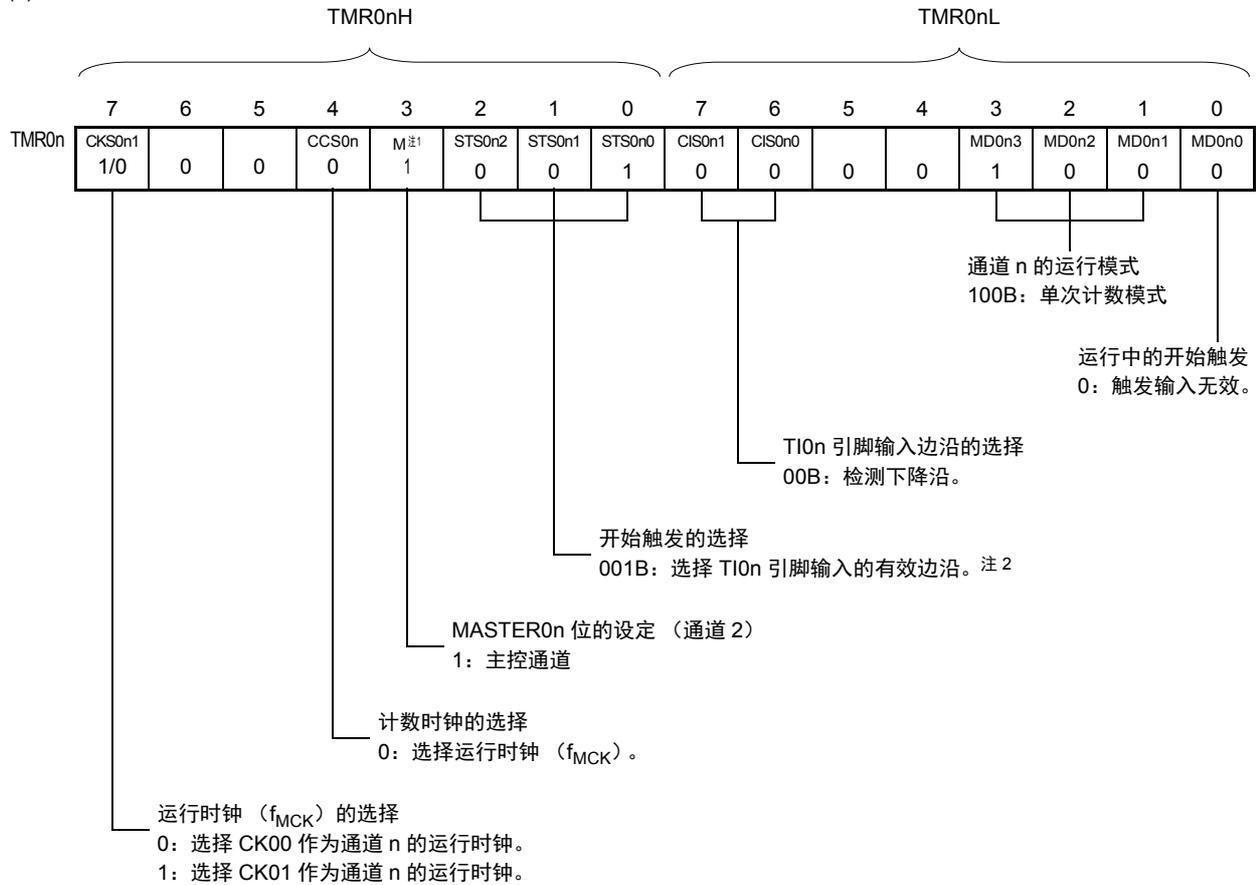
TCR0n、TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)

TDR0n、TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)

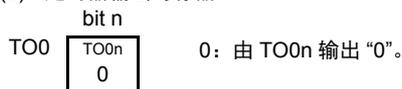
TO0n、TO0p: TO0n 引脚和 TO0p 引脚的输出信号

图 6-71 双输入式单触发脉冲输出（主控通道）的寄存器设定内容例子

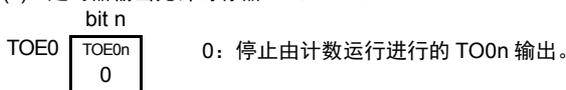
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



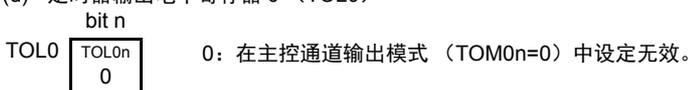
(b) 定时器输出寄存器 0 (TO0)



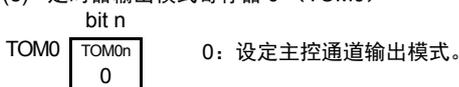
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)

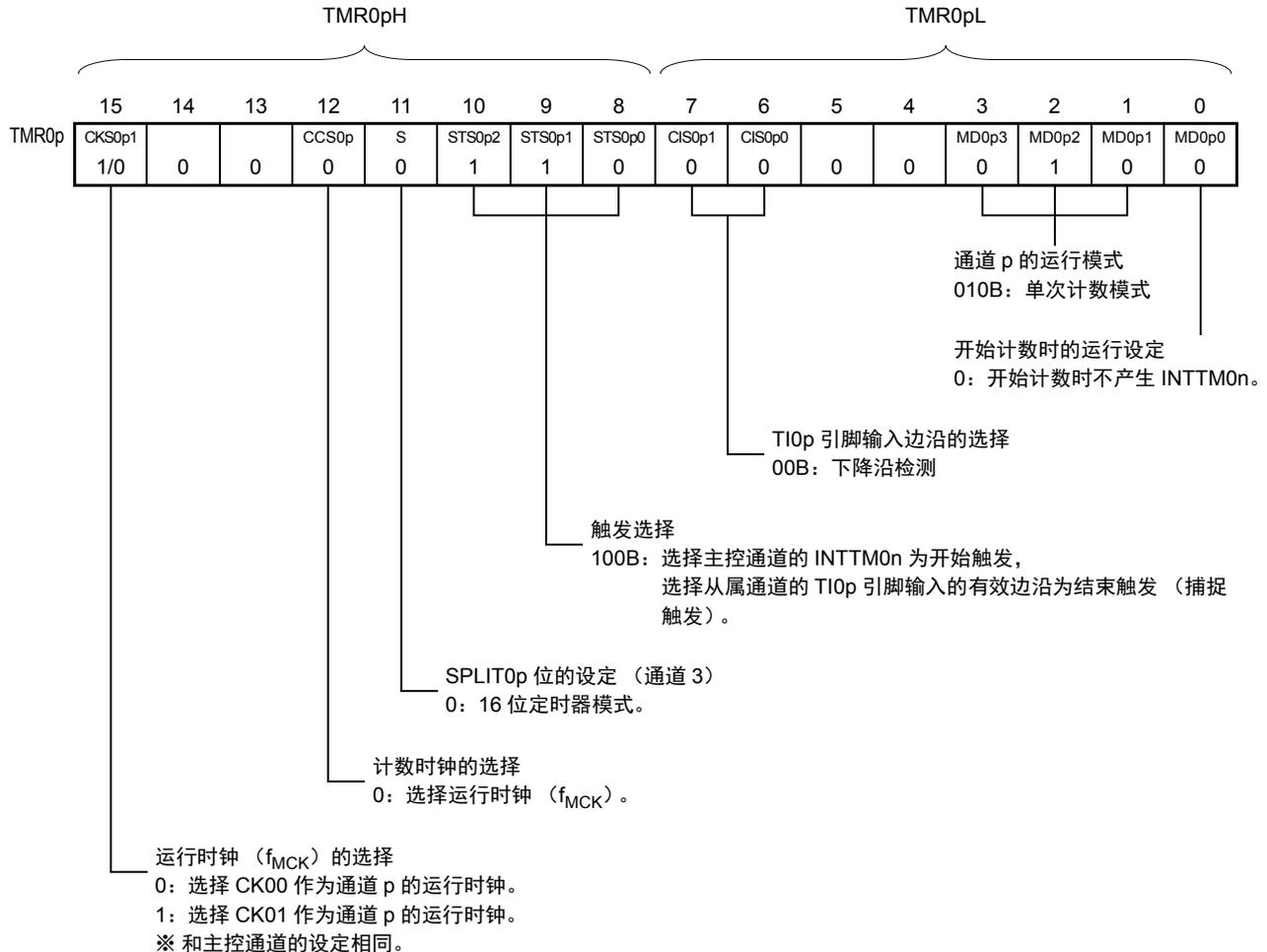


- 注 1. TMR02: MASTER02 位
TMR00: 固定为“0”
2. 不使用 TI0n 引脚时, 能通过软件操作 (TS0n=1) 置为开始触发。

备注 n: 主控通道号 (n=0、2)

图 6-72 双输入式单触发脉冲输出（从属通道）的寄存器设定内容例子

(a) 定时器模式寄存器 0p (TMR0pH、TMR0pL)



(b) 定时器输出寄存器 0 (TO0)

bit p		
TO0	TO0p 1/0	0: 由 TO0p 输出“0”。 1: 由 TO0p 输出“1”。

(c) 定时器输出允许寄存器 0 (TOE0)

bit p		
TOE0	TOE0p 1/0	0: 停止由计数运行进行的 TO0p 输出（从 TO0p 引脚输出 TO0p 位设定的电平）。 1: 允许由计数运行进行的 TO0p 输出（从 TO0p 引脚交替输出）。

(d) 定时器输出电平寄存器 0 (TOL0)

bit p		
TOL0	TOL0p 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 0 (TOM0)

bit p		
TOM0	TOM0p 1	1: 设定从属通道输出模式。

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (p=3)

图 6-73 双输入式单触发脉冲输出的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 “1” (在 TAU0EN 位为“0”时不能读写)。—————>	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	
通道初 始设定	设定噪声滤波器允许寄存器 1 (NFEN1)。 设定定时器模式寄存器 0n、0p (TMR0n、TMR0p) (确定各通道的运行模式, 选择检测边沿。)	通道处于运行停止状态。
	主控通道的设定 给定时器数据寄存器 0n (TDR0n) 设定延迟 (输出 延迟时间) (TDR0nH、TDR0nL 寄存器的存取步骤 请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 将定时器输出模式寄存器 0 (TOM0) 的对象位 置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位 置“0”。 从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位 置“1” (从属通道输出模式)。 设定 TOL0 寄存器的对象位。 设定 TO0p 位并且确定 TO0p 输出的初始电平。 将 TOE0p 位置“1”, 允许 TO0p 的运行。—————> 将端口寄存器和端口模式寄存器置“0” (设定输出 模式)。—————>	TO0p 引脚处于 Hi-Z 状态。 (端口模式寄存器为输入模式) 因为通道处于运行停止状态, 所以 TO0p 不变 (即 使改写 TO0p 位也不反映到 TO0p 引脚)。 从 TO0p 引脚输出 TO0p 位设定的电平。

备注 n: 主控通道号 (n=0、2)

p: 从属通道号 (p=3)

图 6-73 双输入式单触发脉冲输出的操作步骤 (2/2)

	软件操作	硬件状态
开始运行	将从属通道的 TOE0p 位置“1”，允许 TO0p 的运行（只在重新开始运行时）。 将 TS0 寄存器的对象位（主控、从属）同时置“1”→ 因为 TS0 寄存器的对象位是触发位，所以自动返回到“0”。	TE0 寄存器的对象位变为“1”并且主控通道进入 TI0n 引脚输入的有效边沿检测等待状态。
	通过检测到以下的开始触发，开始计数。→ • 检测到 TI0n 引脚输入的有效边沿 • 通过软件将 TS0n 位置“1”	将 TDR0n 寄存器的值装入主控通道的定时器计数寄存器 0n (TCR0n)，开始递减计数。
运行中	更改主控通道的设定 随时能读 TCR0n 寄存器 (TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n (TCR0n)”)。 只能更改 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位的设定值。 禁止更改 TDR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。 更改从属通道的设定 随时能读 TDR0p 寄存器。 随时能读 TCR0p 寄存器。 随时能读 TSR0p 寄存器。 能更改 TMR0p 寄存器的 CIS0p1 位和 CIS0p0 位的设定值。 能更改 TO0p、TOE0p、TOM0 和 TOL0 寄存器的对象位的设定值。	主控通道的计数器 (TCR0n) 进行递减计数。如果 TCR0n 计数到“0000H”，就产生 INTTM0n，并且在下一个开始触发检测（检测到 TI0n 引脚输入的有效边沿或者将 TS0n 位置“1”前 TCR0n 为“FFFFH”而且停止计数。 从属通道将主控通道的 INTTM0n 作为开始触发，将定时器计数寄存器 0n (TCR0n) 清“0000H”。计数器 (TCR0n) 从“0000H”开始进行递增计数，在检测到 TI0n 引脚输入的有效边沿后，将计数值传送（捕捉）到定时器数据寄存器 0n (TDR0n) 并且将 TCR0n 寄存器清“0000H”。与此同时，产生 INTTM0n，并且 TO0p 输出电平变为无效电平。此后，重复此运行。
停止运行	将 TT0 寄存器的对象位（主控、从属）同时置“1”。→ 因为 TT0 寄存器的对象位是触发位，所以自动返回到“0”。	TE0 寄存器的对象位变为“0”并且停止计数。 TCR0n 和 TCR0p 寄存器保持计数值而停止计数。 TO0p 输出不被初始化而保持状态。
	将从属通道的 TOE0p 位置“0”并且给 TO0p 位设定值。→	从 TO0p 引脚输出 TO0p 位设定的电平。
TAU 停止	要保持 TO0p 引脚的输出电平的情况 在给端口寄存器设定要保持的值（输出锁存器）后将 TO0p 位置“0”。→ 将 PER0 寄存器的 TAU0EN 位置“0”。→	通过端口功能保持 TO0p 引脚的输出电平。 断电状态 (停止提供时钟，TAU 的 SFR 被初始化。)

重新开始运行

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

6.9.3 作为 PWM 输出功能的运行

将 2 个通道成对使用，能生成任意周期和占空比的脉冲。

在将通道 1 和通道 3 用作 8 位定时器运行功能（SPLIT0n=1）时，只有低 8 位定时器能用作 PWM 输出功能的从属通道。

输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{\text{TDR0n (主控) 的设定值} + 1\} \times \text{计数时钟周期} \\ \text{占空比 [\%]} &= \{\text{TDR0p (从属) 的设定值}\} / \{\text{TDR0n (主控) 的设定值} + 1\} \times 100 \\ \text{0\% 输出: } &\text{TDR0p (从属) 的设定值} = 0000\text{H (8 位定时器运行时为 "00H")} \\ \text{100\% 输出: } &\text{TDR0p (从属) 的设定值} \geq \{\text{TDR0n (主控) 的设定值} + 1\} \end{aligned}$$

备注 当 TDR0p（从属）的设定值 > {TDR0n（主控）的设定值 + 1} 时，占空比的计算值超过 100%，但是在 PWM 输出运行的情况下为 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 0（TS0）的通道开始触发位（TS0n）置“1”，就输出中断请求信号（INTTM0n），然后将定时器数据寄存器 0n（TDR0n）的设定值装入定时器计数寄存器 0n（TCR0n），并且通过计数时钟（ f_{TCLK} ）进行递减计数。当 TCR0n 计数到“0000H”时，在输出 INTTM0n 后再次将 TDR0n 寄存器的值装入 TCR0n 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 0（TT0）的通道停止触发位（TT0n）置“1”前，重复此运行。

当用作 PWM 输出功能时，主控通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出（TO0p）周期。

从属通道用作单次计数模式。以主控通道的 INTTM0n 中断为开始触发，将 TDR0p 寄存器的值装入 TCR0p 寄存器，并且进行递减计数，计数到 TCR0p 变为“0000H”为止。当 TCR0p 计数到“0000H”时，输出 INTTM0p，并且在下一个开始触发检测（主控通道的 INTTM0n）前 TCR0p 为“FFFFH”而且停止计数。

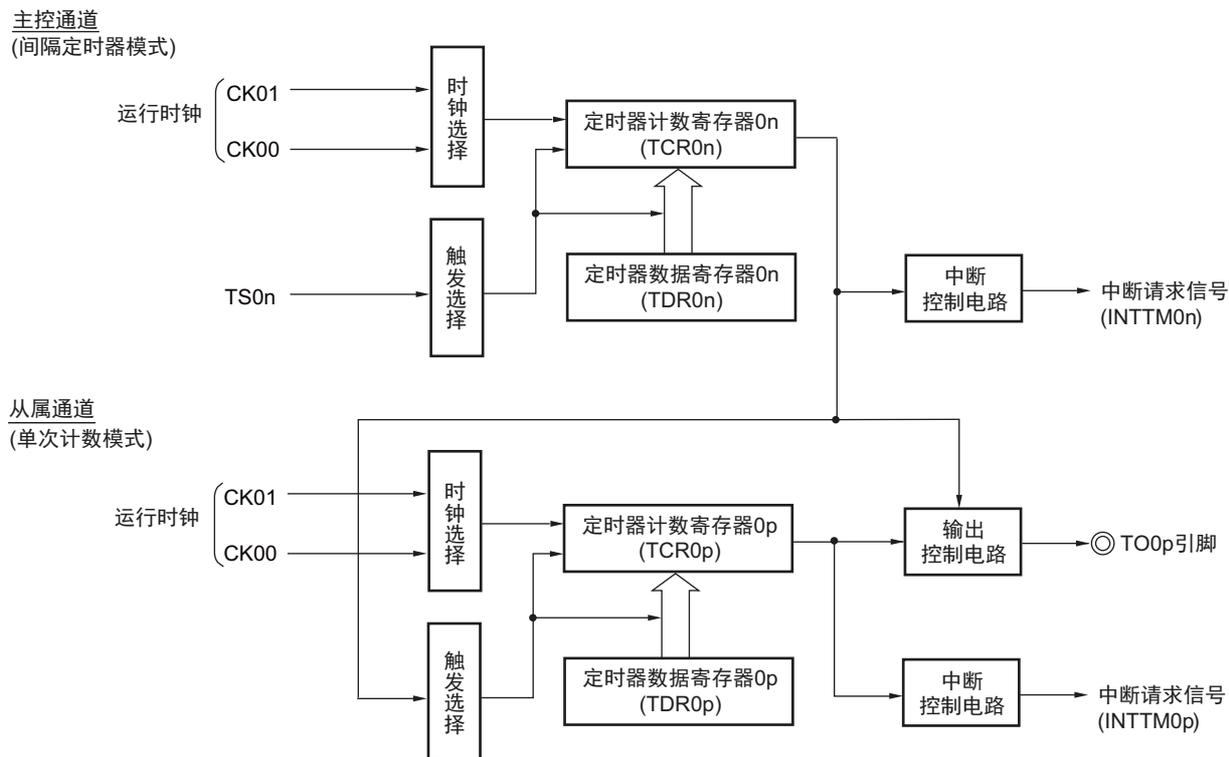
当用作 PWM 输出功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出（TO0p）的占空比。

在从主控通道产生 INTTM0n 并且经过 1 个计数时钟（ f_{TCLK} ）后，PWM 输出（TO0p）变为有效电平，并且在从属通道的 TCR0p 寄存器的值变为“0000H”时变为无效电平。

- 注意** 1. 要同时改写主控通道的定时器数据寄存器 0n（TDR0nH、TDR0nL）和从属通道的 TDR0pH、TDR0pL 寄存器时，至少需要 4 次写存取。因为在产生主控通道的 INTTM0n 后将 TDR0nH、TDR0nL、TDR0pH 和 TDR0pL 寄存器的值装入 TCR0n、TCR0nL、TCR0pH 和 TCR0pL 寄存器，所以如果分别在产生主控通道的 INTTM0n 前和产生后进行改写，TO0p 引脚就不能输出期待的波形。因此，要同时改写 TDR0nH、TDR0nL、TDR0pH 和 TDR0pL 寄存器时，必须在产生主控通道的 INTTM0n 后立即连续改写这 4 个寄存器。
2. 要将 PWM 输出功能用作 8 位定时器运行功能时，必须将主控通道的 TDR0nH 设定为“00H”，并且将脉冲周期设定为 8 位定时器。主控通道的 TDR0nL 寄存器的设定值范围为“00H ~ FEH”（0% 输出 ~ 10% 输出）。

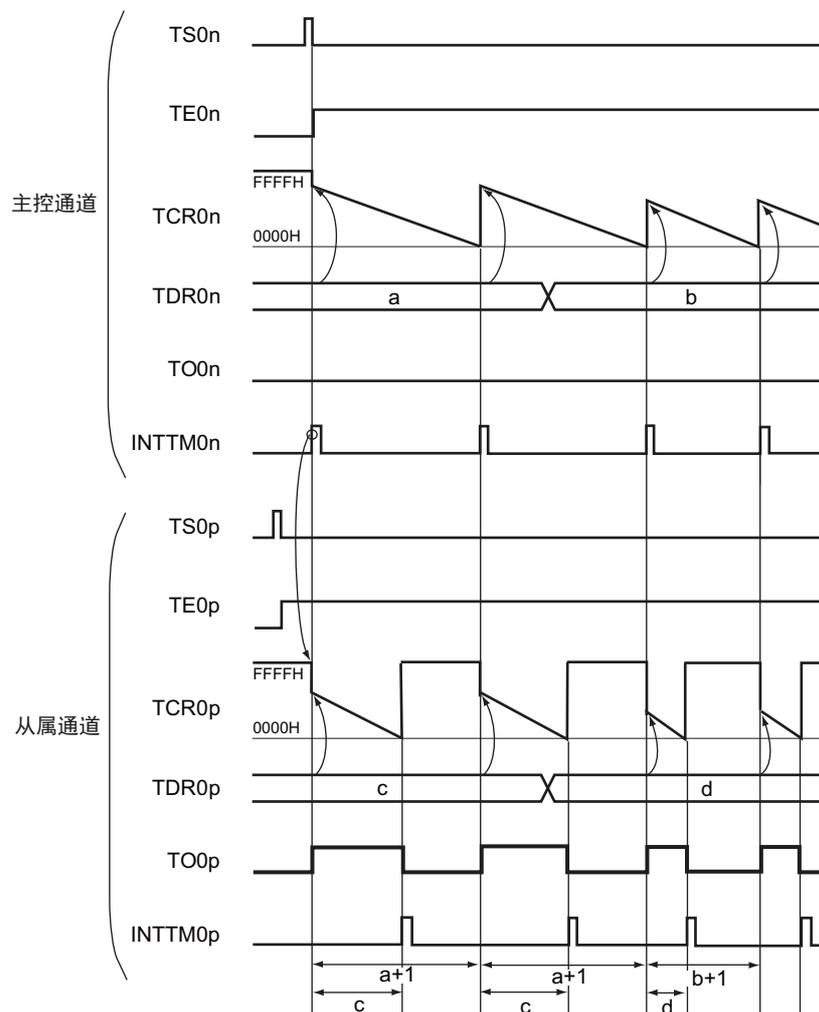
备注 n: 主控通道号（n=0、2）
p: 从属通道号（n < p ≤ 3）

图 6-74 作为 PWM 输出功能运行的框图



备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

图 6-75 作为 PWM 输出功能运行的基本时序例子



备注 1. n: 主控通道号 (n=0、2)

p: 从属通道号 ($n < p \leq 3$)

2. TS0n、TS0p: 定时器通道开始寄存器 0 (TS0) 的 bit n、p

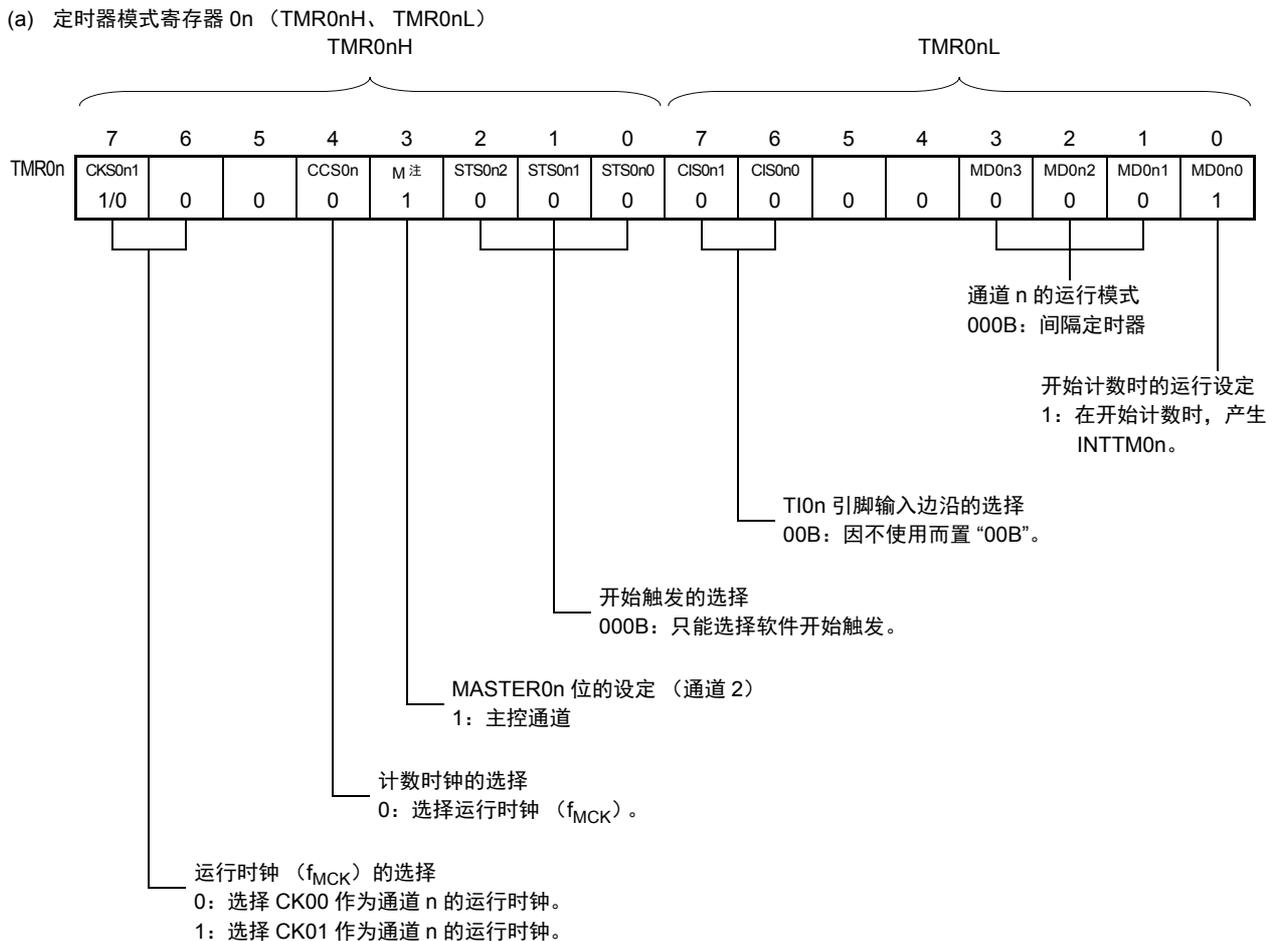
TE0n、TE0p: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p

TCR0n、TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)

TDR0n、TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)

TO0n、TO0p: TO0n 引脚和 TO0p 引脚的输出信号

图 6-76 PWM 输出功能（主控通道）的寄存器设定内容例子



(b) 定时器输出寄存器 0 (TO0)

bit n
TO0

TO0n
0

 0: 由 TO0n 输出“0”。

(c) 定时器输出允许寄存器 0 (TOE0)

bit n
TOE0

TOE0n
0

 0: 停止由计数运行进行的 TO0n 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

bit n
TOL0

TOL0n
0

 0: 在 主控通道输出模式 (TOM0n=0) 中设定无效。

(e) 定时器输出模式寄存器 0 (TOM0)

bit n
TOM0

TOM0n
0

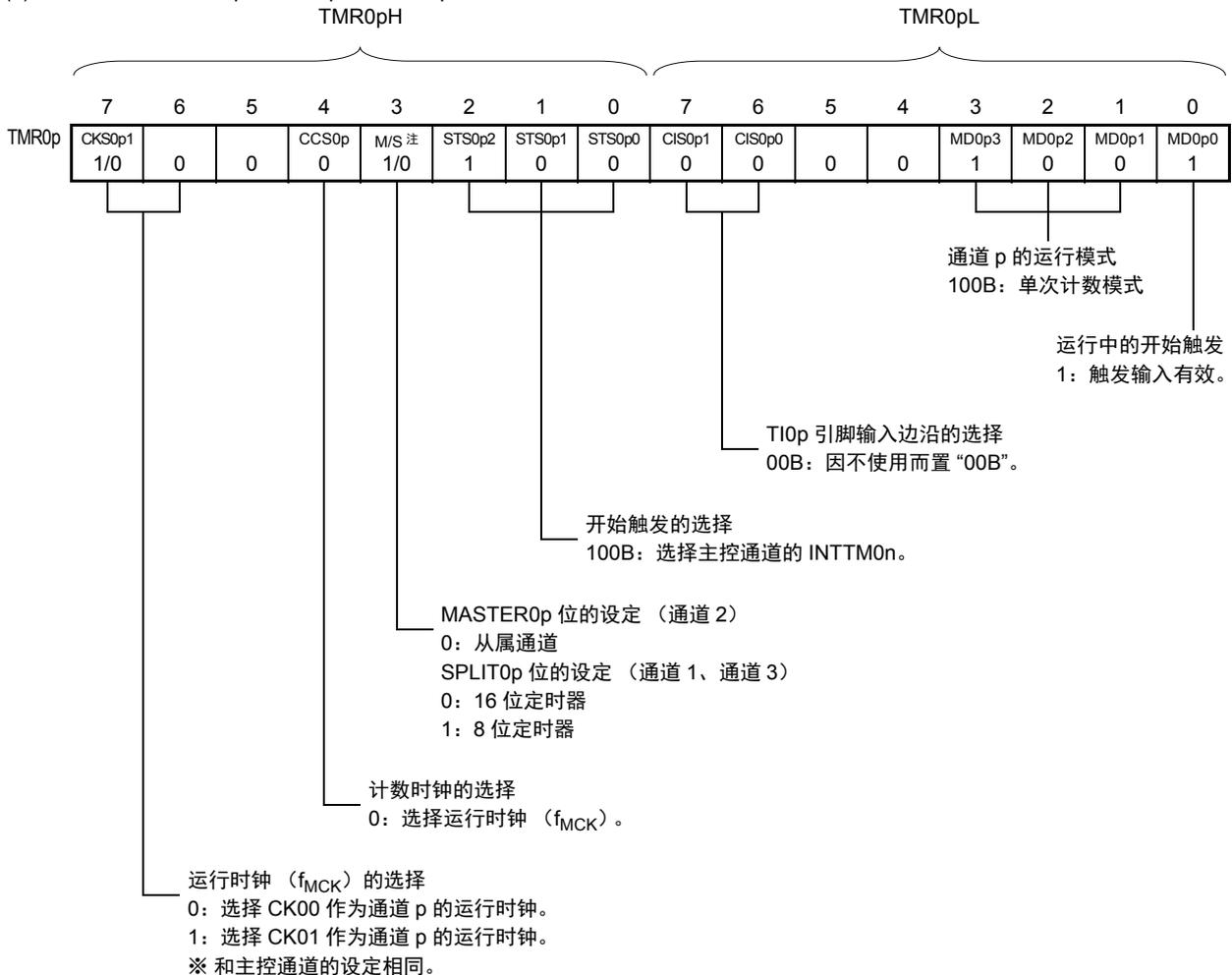
 0: 设定 主控通道输出模式。

注 TMR02: MASTER02 位
TMR00: 固定为“0”

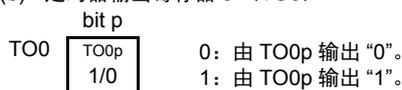
备注 n: 主控通道号 (n=0、2)

图 6-77 PWM 输出功能（从属通道）的寄存器设定内容例子

(a) 定时器模式寄存器 0p (TMR0pH、TMR0pL)



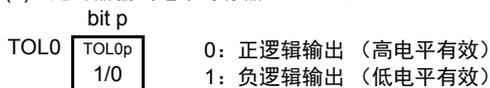
(b) 定时器输出寄存器 0 (TO0)



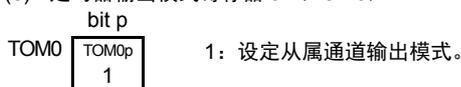
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR02: MASTER0n 位
TMR01、TMR03: SPLIT0p 位

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

图 6-78 PWM 输出功能的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1” (在 TAU0EN 位为“0”时不能读写)。	断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
通道初 始设定	设定定时器模式寄存器 0n、0p (TMR0n、TMR0p) (确定各通道的运行模式)。 给定时器数据寄存器 0n、0p (TDR0n、TDR0p) 设定主控通道的间隔 (周期) 值, 并且设定从属通道的占空比值 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。	通道处于运行停止状态。 (提供时钟, 消耗少许电力。)
	主控通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置“0”。 从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位置“1” (从属通道输出模式)。 设定 TOL0 寄存器的对象位。 设定 TO0p 位并且确定 TO0p 输出的初始电平。 将 TOE0p 位置“1”, 允许 TO0p 的运行。	TO0p 引脚处于 Hi-Z 状态。 (端口模式寄存器为输入模式) 因为通道处于运行停止状态, 所以 TO0p 不变 (即使改写 TO0p 位也不反映到 TO0p 引脚)。 从 TO0p 引脚输出 TO0p 位设定的电平。

备注 n: 主控通道号 (n=0、2)

p: 从属通道号 (n < p ≤ 3)

图 6-78 PWM 输出功能的操作步骤 (2/2)

	软件操作	硬件状态
重新开始运行	<p>开始运行</p> <p>将从属寄存器的 TOE0p 位置“1”，允许 TO0p 的运行（只限于重新开始运行时）。</p> <p>将 TS0 寄存器的对象位（主控、从属）同时置“1”→ 因为 TS0 寄存器的对象位是触发位，所以自动返回到“0”。</p>	<p>TE0 寄存器的对象位变为“1”并且将 TDR0n 寄存器的值装入主控通道的定时器计数寄存器 0n（TCR0n），开始递减计数。</p>
	<p>运行中</p> <p>更改主控通道的设定</p> <p>在产生主控通道的 INTTM0n 后能更改 TDR0n 寄存器的设定值。</p> <p>随时能读 TCR0n 寄存器（TCR0nH、TCR0nL 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n（TCR0n）”）。</p> <p>禁止更改 TMR0n、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。</p> <p>更改从属通道的设定</p> <p>在产生主控通道的 INTTM0n 后能更改 TDR0p 寄存器的设定值</p> <p>随时能读 TCR0p 寄存器。</p> <p>能更改 TO0、TOE0 和 TOL0 寄存器的对象位的设定值。</p> <p>禁止更改 TMR0p 和 TOM0 寄存器的对象位的设定值。</p>	<p>主控通道的定时器计数寄存器 0n（TCR0n）进行递减计数。如果 TCR0n 计数到“0000H”，就产生 INTTM0n，同时，在将 TDR0n 寄存器的值装入 TCR0n 寄存器后，又开始递减计数。</p> <p>从属通道将主控通道的 INTTM0n 作为开始触发，将 TDR0p 寄存器的值装入 TCR0p 寄存器并且开始递减计数。在产生主控通道的 INTTM0n 并且经过 1 个计数时钟（f_{TCLK}）后，将 TO0p 的输出电平置为有效电平。之后，如果 TCR0p 计数到“0000H”，就再将 TO0p 的输出电平置为无效电平，并且 TCR0p 为“FFFFH”而且停止计数。此后，重复此运行。</p>
	<p>停止运行</p> <p>将 TT0 寄存器的对象位（主控、从属）同时置“1”。→</p> <p>因为 TT0 寄存器的对象位是触发位，所以自动返回到“0”。</p> <p>将从属通道的 TOE0p 位置“0”并且给 TO0p 位设定值。→</p>	<p>TE0 寄存器的对象位变为“0”并且停止计数。</p> <p>TCR0n 和 TCR0p 寄存器保持计数值而停止计数。TO0p 输出不被初始化而保持状态。</p> <p>从 TO0p 引脚输出 TO0p 位设定的电平。</p>
	<p>TAU 停止</p> <p>要保持 TO0p 引脚的输出电平的情况</p> <p>在给端口寄存器设定要保留的值（输出锁存器）后将 TO0p 位置“0”。→</p> <p>将 PER0 寄存器的 TAU0EN 位置“0”。→</p>	<p>通过端口功能保持 TO0p 引脚的输出电平。</p> <p>断电状态 (停止提供时钟，TAU 的 SFR 被初始化。)</p>

备注 n: 主控通道号 (n=0、2)
p: 从属通道号 (n < p ≤ 3)

6.9.4 作为多重 PWM 输出功能的运行

这是通过扩展 PWM 输出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。只有 16 引脚产品能使用多重 PWM 输出功能。

通道 1 和通道 3 用作 8 位定时器运行功能（SPLIT0n）时，只有低 8 位定时器能用作 PWM 输出功能的从属通道。

例如，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：

<p>脉冲周期 = {TDR0n（主控）的设定值 + 1} × 计数时钟周期</p> <p>占空比 1 [%] = {TDR0p（从属 1）的设定值} / {TDR0n（主控）的设定值 + 1} × 100</p> <p>占空比 2 [%] = {TDR0q（从属 2）的设定值} / {TDR0n（主控）的设定值 + 1} × 100</p>
--

备注 当 TDR0p（从属 1）的设定值 > {TDR0n（主控）的设定值 + 1} 或者 {TDR0q（从属 2）的设定值} > {TDR0n（主控）的设定值 + 1} 时，占空比的计数值超过 100%，但是在 PWM 运行的情况下为 100% 输出。

主控通道对脉冲周期进行计数。在间隔定时器模式中运行主控通道，将 TDR0n 寄存器的值装入 TCR0n 寄存器，开始递减计数。

从属通道 1 对占空比进行计数，并且从 TO0p 引脚输出任意的 PWM 波形。在单次计数模式中运行从属通道 1，将主控通道的 INTTM0n 作为开始触发，并且将 TDR0p 寄存器的值装入 TCR0p 寄存器，直到 TCR0p 变为 "0000H" 前进行递减计数。如果 TCR0p 变为 "0000H"，就输出 INTTM0p，并且在下一个开始触发检测（主控通道的 INTTM0n）前 TCR0p 为 "FFFFH" 而且停止计数。

从属通道 2 和从属通道 1 相同，对占空比进行计数，并且从 TO0q 引脚输出任意的 PWM 波形。在单次计数模式中运行从属通道 2，将主控通道的 INTTM0n 作为开始触发，并且将 TDR0q 寄存器的值装入 TCR0q 寄存器，直到 TCR0q 变为 "0000H" 前进行递减计数。如果 TCR0q 变为 "0000H"，就输出 INTTM0q，并且在下一个开始触发检测（主控通道的 INTTM0n）前 TCR0q 为 "FFFFH" 而且停止计数。

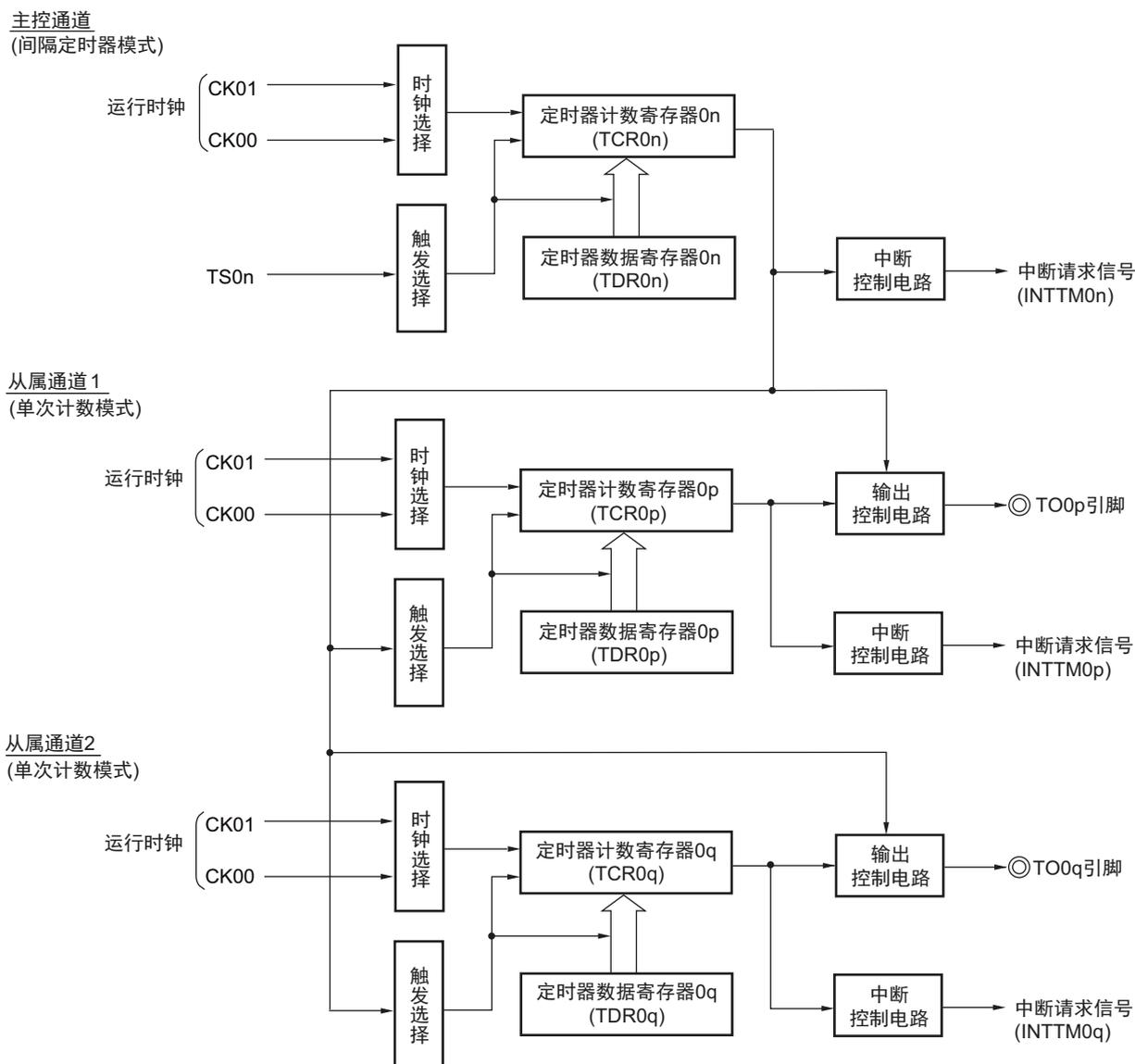
PWM 输出（TO0p、TO0q）在产生主控通道的 INTTM0n 并且经过 1 个计数时钟（ f_{TCLK} ）后变为有效电平，在 TCR0p 变为 "0000H" 或者 TCR0q 变为 "0000H" 时变为无效电平。

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

- 注意** 1. 要同时改写主控通道的定时器数据寄存器 0n（TDR0nH、TDR0nL）和从属通道的 TDR0pH、TDR0pL 寄存器时，至少需要 4 次写存取。因为在产生主控通道的 INTTM0n 后将 TDR0nH、TDR0nL、TDR0pH 和 TDR0pL 寄存器的值装入 TCR0nH、TCR0nL、TCR0pH 和 TCR0pL 寄存器，所以如果分别在产生主控通道的 INTTM0n 前和产生后进行改写，TO0p 引脚就不能输出期待的波形。因此，要同时改写 TDR0nH、TDR0nL、TDR0pH 和 TDR0pL 寄存器时，必须在产生主控通道的 INTTM0n 后立即连续改写这 4 个寄存器。
2. 要将 PWM 输出功能用作 8 位定时器运行功能时，必须将主控通道的 TDR0nH 设定为 "00H"，并且将脉冲周期设定为 8 位定时器。主控通道的 TDR0nL 寄存器的设定值范围为 "00H ~ FEH"（0% 输出 ~ 100% 输出）。

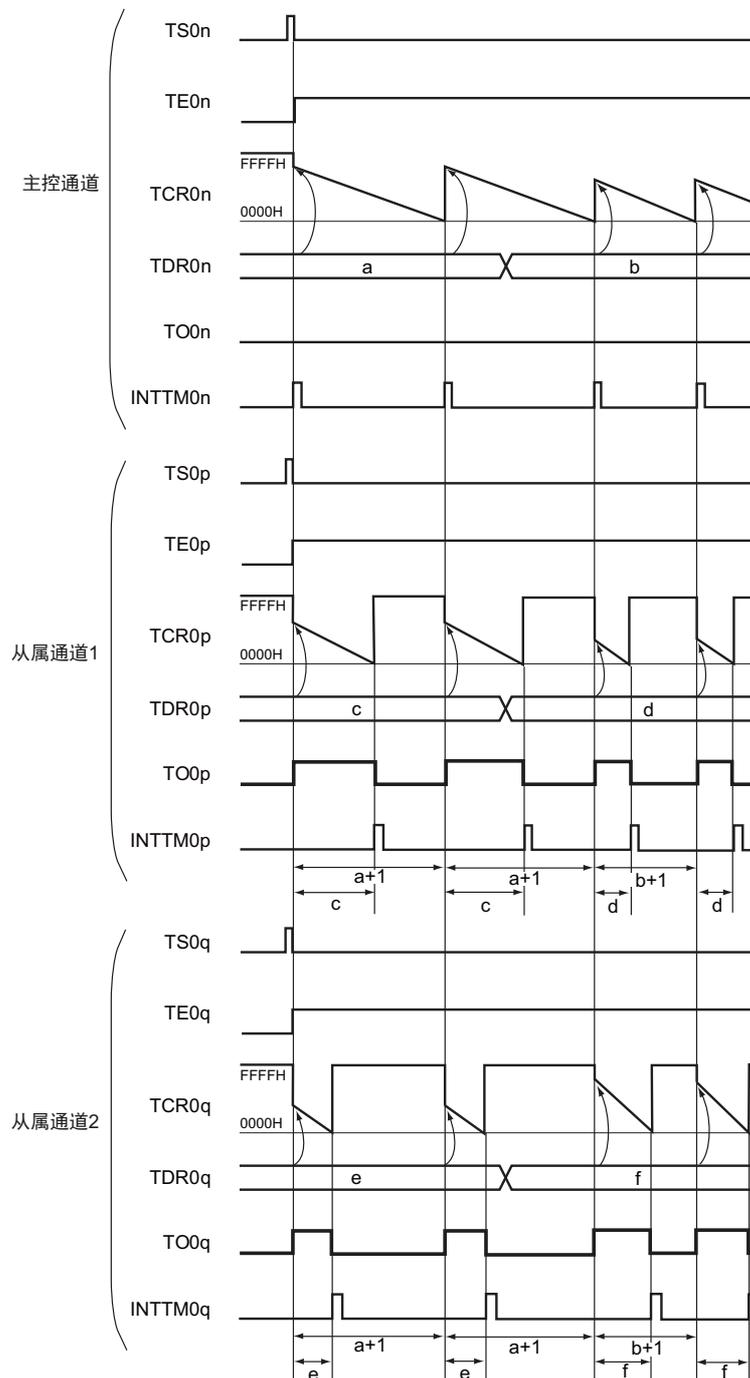
备注 n: 主控通道号（n=0）
 p: 从属通道号 q: 从属通道号
 $n < p < q \leq 3$ （p 和 q 是大于 n 的整数）

图 6-79 作为多重 PWM 输出功能运行的框图（输出 2 种 PWM 的情况）



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

图 6-80 作为多重 PWM 输出功能运行的基本时序例子（输出 2 种 PWM 的情况）



备注 1. n: 主控通道号 (n=0)

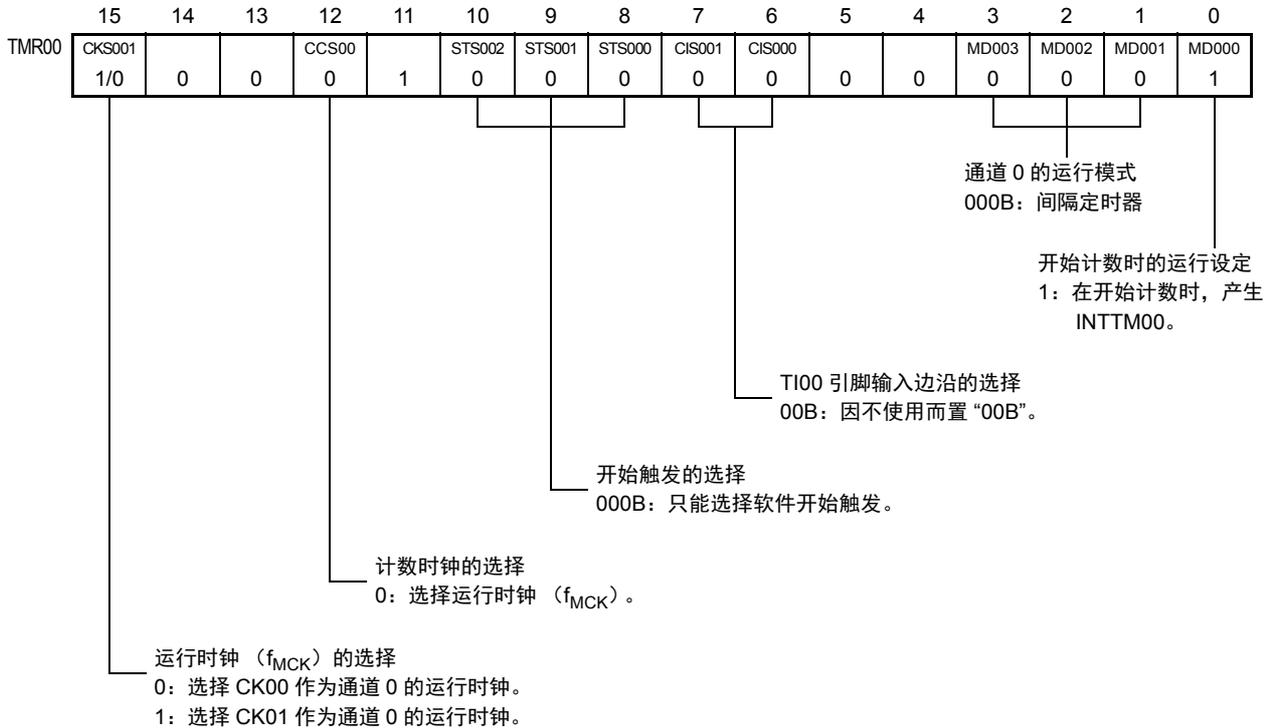
p: 从属通道号 q: 从属通道号

$n < p < q \leq 3$ (p 和 q 是大于 n 的整数)

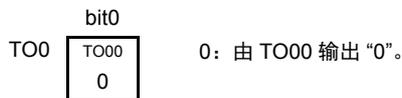
2. TS0n、TS0p、TS0q: 定时器通道开始寄存器 0 (TS0) 的 bit n、p、q
- TE0n、TE0p、TE0q: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p、q
- TCR0n、TCR0p、TCR0q: 定时器计数寄存器 0n、0p、0q (TCR0n、TCR0p、TCR0q)
- TDR0n、TDR0p、TDR0q: 定时器数据寄存器 0n、0p、0q (TDR0n、TDR0p、TDR0q)
- TO0n、TO0p、TO0q: TO0n、TO0p、TO0q 引脚的输出信号

图 6-81 多重 PWM 输出功能（主控通道）的寄存器设定内容例子

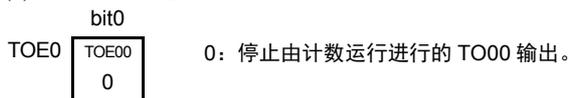
(a) 定时器模式寄存器 00 (TMR00)



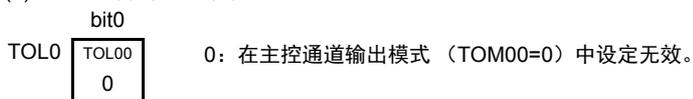
(b) 定时器输出寄存器 0 (TO0)



(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)

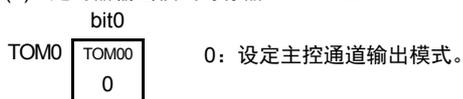
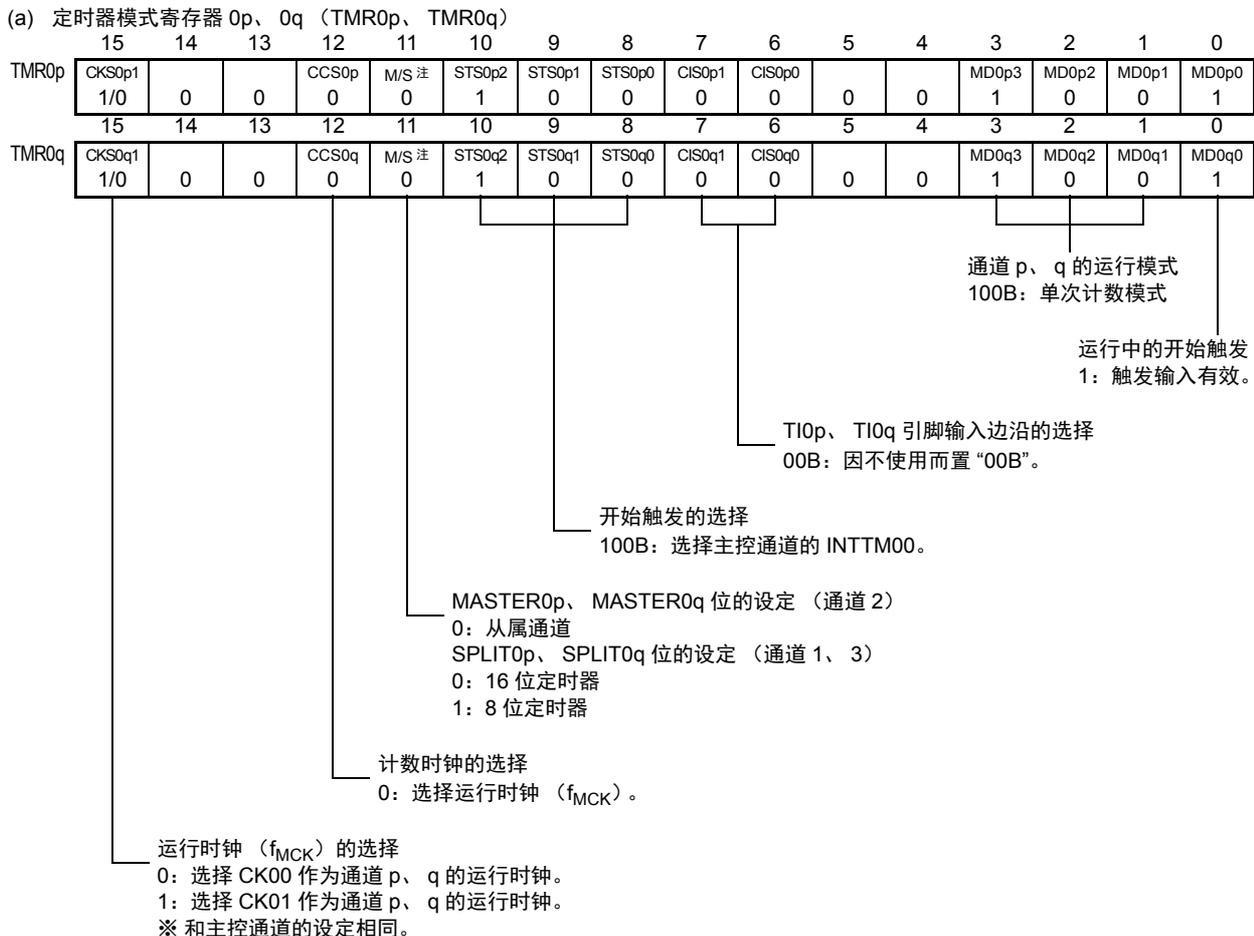


图 6-82 多重 PWM 输出功能（从属通道）的寄存器设定内容例子（输出 2 种 PWM 的情况）



(b) 定时器输出寄存器 0 (TO0)

	bit q	bit p	
TO0	TO0q	TO0p	0: 由 TO0p 和 TO0q 输出“0”。
	1/0	1/0	1: 由 TO0p 和 TO0q 输出“1”。

(c) 定时器输出允许寄存器 0 (TOE0)

	bit q	bit p	
TOE0	TOE0q	TOE0p	0: 停止由计数运行进行的 TO0p 和 TO0q 输出 (从 TO0p、TO0q 引脚输出 TO0p、TO0q 位设定的电平)。
	1/0	1/0	1: 允许由计数运行进行的 TO0p 和 TO0q 输出 (从 TO0p、TO0q 引脚交替输出)。

(d) 定时器输出电平寄存器 0 (TOL0)

	bit q	bit p	
TOL0	TOL0q	TOL0p	0: 正逻辑输出 (高电平有效)
	1/0	1/0	1: 负逻辑输出 (低电平有效)

(e) 定时器输出模式寄存器 0 (TOM0)

	bit q	bit p	
TOM0	TOM0q	TOM0p	1: 设定从属通道输出模式。
	1	1	

注 TMR02: MASTER0p 位、MASTER0q 位
TMR01、TMR03: SPLIT0p 位、SPLIT0q 位

备注 p: 从属通道号 q: 从属通道号
 $0 < p < q \leq 3$ (p 和 q 是大于 0 的整数)

图 6-83 多重 PWM 输出功能的操作步骤（输出 2 种 PWM 的情况）(1/2)

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写 TAU 的 SFR。)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1” (在 TAU0EN 位为“0”时不能读写)。 设定定时器时钟选择寄存器 0 (TPS0)。 确定各通道的运行时钟 (CK00、CK01)。	上电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写 TAU 的 SFR。)
通道初 始设定	设定定时器模式寄存器 00、0p、0q (TMR0n、TMR0p、TMR0q) (确定各通道的运行模式)。 给定时器数据寄存器 00、0p、0q (TDR00、TDR0p、TDR0q) 设定主控通道的间隔 (周期) 值, 并且设定从属通道的占空比值 (TDR0nH、TDR0nL 寄存器的存取步骤请参照“6.2.2 定时器数据寄存器 0n (TDR0n)”)。 主控通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位置“0” (主控通道输出模式)。 将 TOL0 寄存器的对象位置“0”。 将定时器输出允许寄存器 0 (TOE0) 的对象位置“0”。 从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的对象位置“1” (从属通道输出模式)。 设定 TOL0 寄存器的对象位。 设定 TO0p、TO0q 位并且确定 TO0p、TO0q 输出的初始电平。 将 TOE0p、TOE0q 位置“1”, 允许通道计数进行的 TO0p、TO0q 的运行。	通道处于运行停止状态。 (提供时钟, 消耗少许电力)。 TO0p、TO0q 引脚处于 Hi-Z 状态。 (端口模式寄存器为输入模式)
	将端口寄存器和端口模式寄存器置“0” (设定输出模式)。	因为通道处于运行停止状态, 所以 TO0p、TO0q 不变 (即使改写 TO0p、TO0q 位也不反映到 TO0p、TO0q 引脚)。 从 TO0p、TO0q 引脚输出 TO0p、TO0q 位设定的电平。

备注 p: 从属通道号 q: 从属通道号
 $0 < p < q \leq 3$ (p和q是大于0的整数)

图 6-83 多重 PWM 输出功能的操作步骤 (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 将从属寄存器的 TOE0p、TOE0q 位置“1”，允许通过计数进行的 TO0p、TO0q 输出（只限于重新开始运行时）。 将 TS0 寄存器的对象位（主控、从属）同时置“1”→ 因为 TS0 寄存器的对象位是触发位，所以自动返回到“0”。	TE0 寄存器的对象位变为“1”并且将 TDR00 寄存器的值装入主控通道的定时器计数寄存器 00（TCR00），开始递减计数。
	运行中 更改主控通道的设定 在产生主控通道的 INTTM00 后能更改 TDR00 寄存器的设定值。 随时能读 TCR00 寄存器（TCR00H、TCR00L 寄存器的存取步骤请参照“6.2.1 定时器计数寄存器 0n（TCR0n）”）。 禁止更改 TMR00、TO0、TOE0、TOM0 和 TOL0 寄存器的对象位的设定值。 更改从属通道的设定 在产生主控通道的 INTTM00 后能更改 TDR0p、TDR0q 寄存器的设定值 随时能读 TCR0p、TCR0q 寄存器。 能更改 TO0、TOE0 和 TOL0 寄存器的对象位的设定值。 禁止更改 TMR0p、TMR0q 和 TOM0 寄存器的对象位的设定值。	主控通道的定时器计数寄存器 00（TCR00）进行递减计数。如果 TCR00 计数到“0000H”，就产生 INTTM00，同时，在将 TDR00 寄存器的值装入 TCR00 寄存器后，又开始递减计数。 从属通道将主控通道的 INTTM00 作为开始触发，将 TDR0p、TDR0q 寄存器的值装入 TCR0p、TCR0q 寄存器并且开始递减计数。在产生主控通道的 INTTM00 并且经过 1 个计数时钟（ f_{TCLK} ）后，将 TO0p、TO0q 的输出电平置为有效电平。之后，如果 TCR0p 计数到“0000H”或者 TCR0q 计数到“0000H”，就再将 TO0p、TO0q 的输出电平置为无效电平，并且 TCR0p 为“FFFFH”、TCR0q 为“FFFFH”而且停止计数。 此后，重复此运行。
	停止运行 将 TT0 寄存器的对象位（主控、从属）同时置“1”。→ 因为 TT0 寄存器的对象位是触发位，所以自动返回到“0”。 将从属通道的 TOE0p、TOE0q 位置“0”并且给 TO0p、TO0q 位设定值。→	TE0 寄存器的对象位变为“0”并且停止计数。 TCR00、TCR0p 和 TCR0q 寄存器保持计数值而停止计数。 TO0p、TO0q 输出不被初始化而保持状态。 从 TO0p、TO0q 引脚输出 TO0p、TO0q 位设定的电平。
	TAU 停止 要保持 TO0p、TO0q 引脚的输出电平的情况 在给端口寄存器设定要保留的值（输出锁存器）后将 TO0p、TO0q 位置“0”。→ 将 PER0 寄存器的 TAU0EN 位置“0”。→	通过端口功能保持 TO0p、TO0q 引脚的输出电平。 断电状态 (停止提供时钟，TAU 的 SFR 被初始化。)

备注 p: 从属通道号 q: 从属通道号
 $0 < p < q \leq 3$ (p和q是大于0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输出设定为初始状态。

详细内容请参照“4.5 使用复用功能时的寄存器设定”。

第 7 章 12 位间隔定时器

注意 16 引脚产品中内置 1 个通道的 12 位间隔定时器。

7.1 12 位间隔定时器的功能

按事先设定的任意时间间隔产生中断请求信号 (INTIT)，能用于 STOP 模式、HALT 模式的唤醒触发。

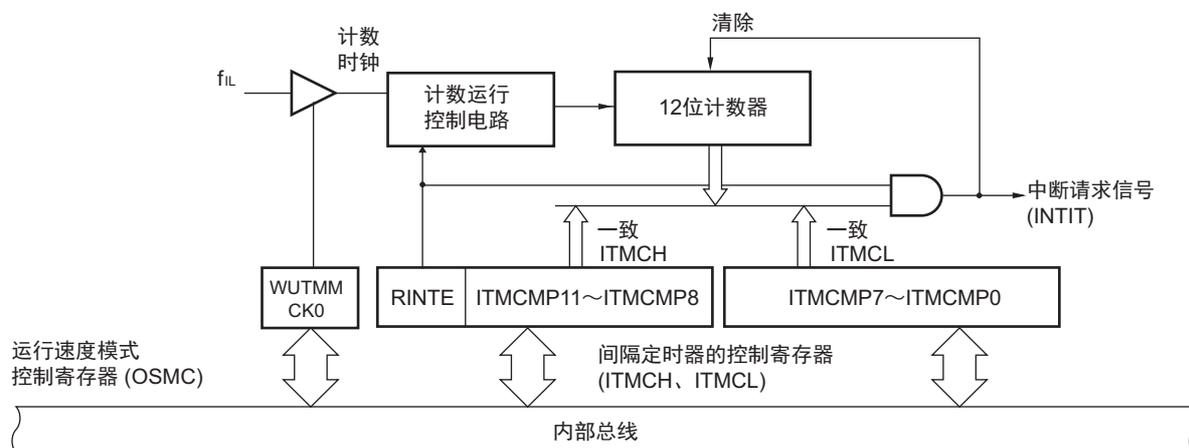
7.2 12 位间隔定时器的结构

12 位间隔定时器由以下硬件构成。

表 7-1 12 位间隔定时器的结构

项目	结构
计数器	12 位计数器
控制寄存器	外围允许寄存器 0 (PER0)
	运行速度模式控制寄存器 (OSMC)
	间隔定时器的控制寄存器 H (ITMCH) 间隔定时器的控制寄存器 L (ITMCL)

图 7-1 12 位间隔定时器的框图



7.3 12 位间隔定时器的控制寄存器

通过以下寄存器控制 12 位间隔定时器。

- 外围允许寄存器 0 (PER0)
- 运行速度模式控制寄存器 (OSMC)
- 间隔定时器的控制寄存器 H (ITMCH)
- 间隔定时器的控制寄存器 L (ITMCL)

7.3.1 外围允许寄存器 0 (PER0)

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

在使用 12 位间隔定时器时，必须首先将 bit7 (TMKAEN) 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 7-2 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	TMKAEN 注	CMPEN 注	ADCEN	IICA0EN 注	0	SAU0EN	0	TAU0EN

TMKAEN	控制 12 位间隔定时器的输入时钟的提供
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写 12 位间隔定时器使用的 SFR。 • 12 位间隔定时器处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> • 能读写 12 位间隔定时器使用的 SFR。

注 只限于 16 引脚产品。

注意 1. 在向 12 位间隔定时器提供输入时钟 (TMKAEN=1) 前，必须将 OSMC 寄存器的 WUTMMCK0 位置“1”，并且确定计数时钟。

2. 如果要设定 12 位间隔定时器，就必须在计数时钟振荡稳定的状态下将 TMKAEN 位置“1”，然后设定以下寄存器。当 TMKAEN 位为“0”时，忽视 12 位间隔定时器的写操作，而且读取值为初始值 (运行速度模式控制寄存器 (OSMC) 除外)。

- 间隔定时器的控制寄存器 H (ITMCH)
- 间隔定时器的控制寄存器 L (ITMCL)

3. 必须将以下位置“0”。

10 引脚产品: bit1、bit3、bit4、bit6、bit7

16 引脚产品: bit1、bit3

7.3.2 运行速度模式控制寄存器（OSMC）

能通过 WUTMMCK0 位控制 12 位间隔定时器的计数时钟的提供。

如果要运行 12 位间隔定时器，必须先将 WUTMMCK0 位置“1”，直到停止计数运行前，不能将 WUTMMCK0 位置“0”。

通过 8 位存储器操作指令设定 OSMC 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 7-3 运行速度模式控制寄存器（OSMC）的格式

地址: F00F3H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	提供 12 位间隔定时器的计数时钟
0	停止提供时钟
1	提供低速内部振荡器时钟 (f_{IL})

7.3.3 间隔定时器的控制寄存器 (ITMCH、ITMCL)

这是设定 12 位间隔定时器的运行开始和停止以及比较值的寄存器。

首先通过 ITMCL 寄存器设定比较值的低 8 位 (ITCMP7 ~ ITCMP0)，然后通过 ITMCH 寄存器设定比较值的高 4 位 (ITCMP11 ~ ITCMP8) 以及计数器运行的停止或者开始。

通过 8 位存储器操作指令设定 ITMCH 寄存器和 ITCML 寄存器。

在产生复位信号后，ITMCH 寄存器的值变为“0FH”、ITMCL 寄存器的值变为“FFH”。

图 7-4 间隔定时器控制寄存器 (ITMCH、ITMCL) 的格式

地址: FFF91H	复位后: 0FH	R/W						
符号	7	6	5	4	3	2	1	0
ITMCH	RINTE	0	0	0	ITCMP11 ~ ITCMP8			

地址: FFF90H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
ITMCL	ITCMP7 ~ ITCMP0							

RINTE	12 位间隔定时器的运行控制
0	停止计数运行 (清除计数)。
1	开始计数运行。

ITCMP11 ~ ITCMP0	12 位间隔定时器比较值的设定
001H	这些位产生“计数时钟周期 × (ITCMP 设定值 + 1)”的固定周期中断。
...	
FFFH	
000H	禁止设定
ITCMP11 ~ ITCMP0 为“001H”或者“FFFH”时的中断周期例子	
• ITCMP11 ~ ITCMP0 = 001H, 计数时钟: $f_{IL}=15\text{kHz}$ $1/15\text{kHz} \times (1+1) \approx 0.1333[\text{ms}] = 133.3[\mu\text{s}]$	
• ITCMP11 ~ ITCMP0 = FFFH, 计数时钟: $f_{IL}=15\text{kHz}$ $1/15[\text{kHz}] \times (4095+1) \approx 273[\text{ms}]$	

- 注意 1. 要停止计数器运行 (将 RINTE 位清“0”) 时, 必须先将 TMKAMK 标志置“1”, 禁止 INTIT 的中断处理后进行改写。要重新开始计数器运行 (将 RINTE 位置“1”) 时, 必须先将 TMKAIF 标志清“0”, 再将 TMKAMK 标志清“0”, 允许 INTIT 的中断处理。
2. RINTE 位的读取值在设定 RINTE 位为“1”并且经过 1 个计数时钟后被反映。
3. 要在从待机模式返回后设定 RINTE 位并且再次转移到待机模式时, 必须确认在 RINTE 位的写入值被反映后或者在从待机模式返回后至少经过 1 个计数时钟的时间之后再转移到待机模式。
4. 要更改 ITCMP11 ~ ITCMP0 位的值时, 必须在停止计数器运行 (RINTE=0) 的状况下进行。
但是, 能在将 RINTE 位从“0”改为“1”或者从“1”改为“0”的同时更改 ITCMP11 ~ ITCMP8 位的值。

7.4 12 位间隔定时器的运行

7.4.1 12 位间隔定时器的运行时序

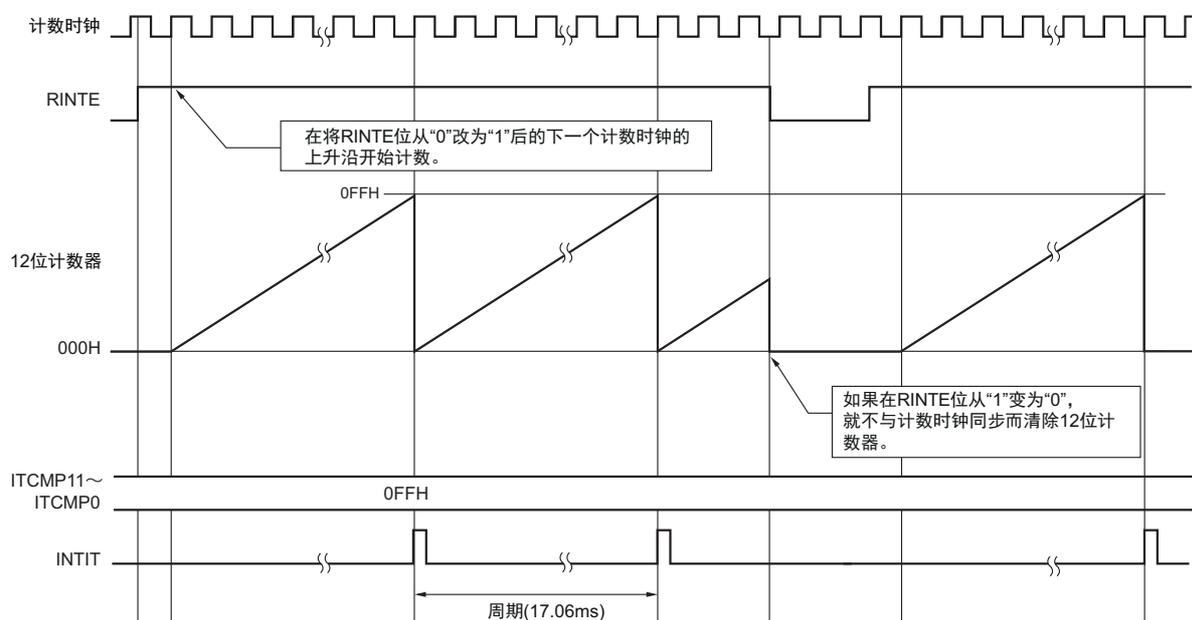
以 ITCMP11 ~ ITCMP0 位设定的计数值为间隔，作为重复产生中断请求（INTIT）的 12 位间隔定时器运行。

如果将 RINTE 位置“1”，12 位计数器就开始计数。

当 12 位计数器的值和 ITCMP11 ~ ITCMP0 位的设定值相同时，将 12 位计数器清“0”并且继续进行计数，同时产生中断请求信号（INTIT）。

12 位间隔定时器的基本运行如图 7-5 所示。

图 7-5 12 位间隔定时器的运行时序
(ITCMP11 ~ ITCMP0=0FFH, 计数时钟: $f_{IL}=15\text{kHz}$)

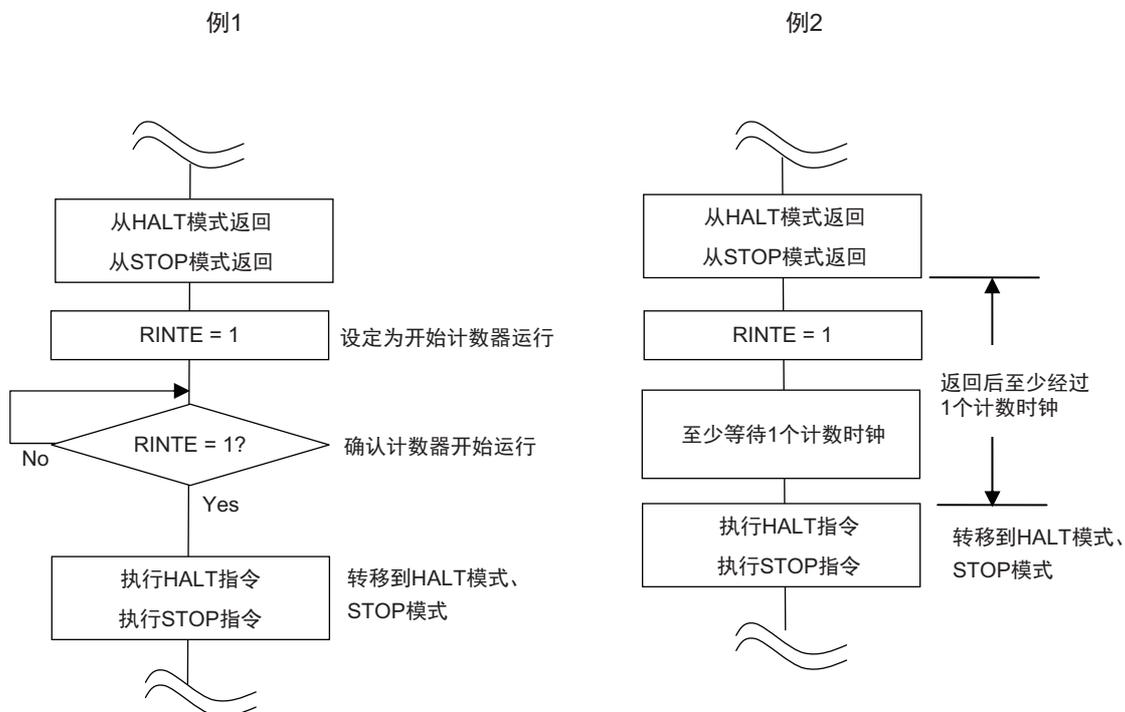


7.4.2 从 HALT/STOP 模式返回后计数器开始运行并且再次转移到 HALT/STOP 模式的设定步骤

如果从 HALT 模式或者 STOP 模式返回后将 RINTE 置“1”，并且要再次转移到 HALT 模式或者 STOP 模式，就必须在将 RINTE 置“1”后，确认是否已反映 RINTE 位的写入值，或者在返回后是否至少经过 1 个计数时钟，然后再进行转移。

- 将 RINTE 位置“1”后，通过轮询确认 RINTE 位已变为“1”，然后转移到 HALT 模式或者 STOP 模式（参照图 7-6 的例 1）。
- 将 RINTE 位置“1”后，至少经过 1 个计数时钟，然后转移到 HALT 模式或者 STOP 模式（参照图 7-6 的例 2）。

图 7-6 将 RINTE 位置“1”后向 HALT 模式或者 STOP 模式转移的步骤



第 8 章 时钟输出 / 蜂鸣器输出控制电路

8.1 时钟输出 / 蜂鸣器输出控制电路的功能

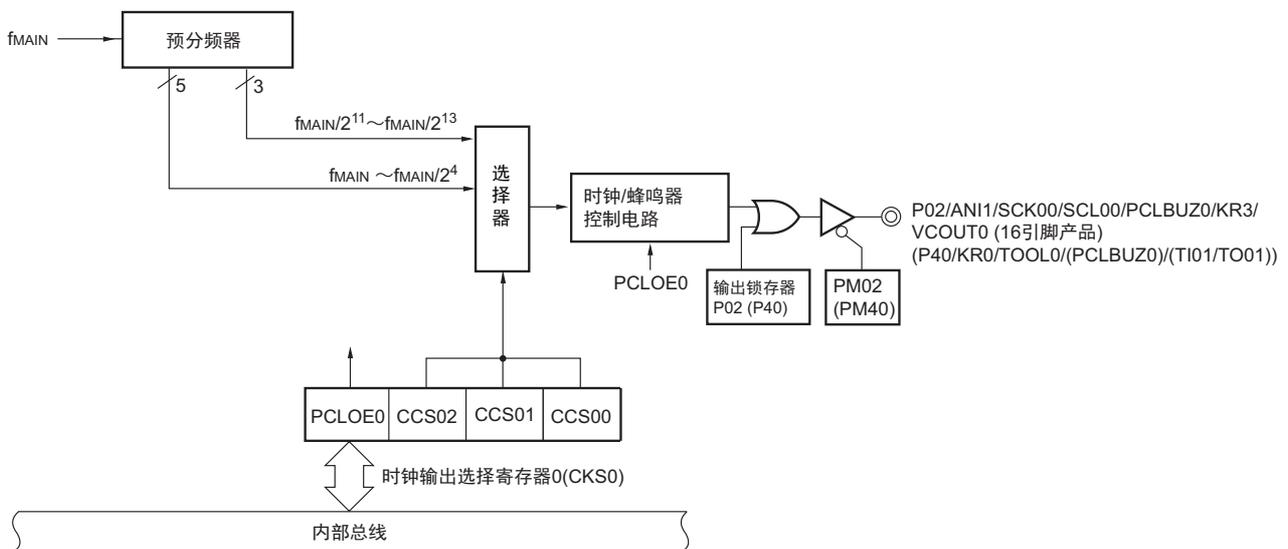
时钟输出是输出提供给外围 IC 时钟的功能，蜂鸣器输出是输出蜂鸣器频率方波的功能。

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。

时钟输出 / 蜂鸣器输出控制电路的框图如图 8-1 所示。

图 8-1 时钟输出 / 蜂鸣器输出控制电路的框图



注意 能从 PCLBUZ0 引脚输出的频率请参照“24.4 AC 特性”。

备注 能通过设定外围 I/O 重定向寄存器 (PIOR)，分配上图 () 内的功能。

8.2 时钟输出 / 蜂鸣器输出控制电路的结构

时钟输出 / 蜂鸣器输出控制电路由以下硬件构成。

表 8-1 时钟输出 / 蜂鸣器输出控制电路的结构

项目	结构
控制寄存器	时钟输出选择寄存器 0 (CKS0) 端口模式寄存器 0 (PM0) (端口模式寄存器 4 (PM4)) 端口寄存器 0 (P0) (端口寄存器 4 (P4)) 端口模式控制寄存器 0 (PMC0) 外围 I/O 重定向寄存器 (PIOR)

备注 能通过设定外围 I/O 重定向寄存器 (PIOR)，分配上表 () 内的功能。

8.3 时钟输出 / 蜂鸣器输出控制电路的控制寄存器

通过以下寄存器控制时钟输出 / 蜂鸣器输出控制电路。

- 时钟输出选择寄存器 0 (CKS0)
- 端口模式寄存 0 (PM0) (端口模式寄存器 4 (PM4))
- 端口模式控制寄存器 0 (PMC0)
- 外围 I/O 重定向寄存器 (PIOR)

备注 能通过设定外围 I/O 重定向寄存器 (PIOR)，分配上述 () 内的功能。

8.3.1 时钟输出选择寄存器 0 (CKS0)

这是允许或者禁止时钟输出引脚或者蜂鸣器频率输出引脚 (PCLBUZ0) 的输出以及设定输出时钟的寄存器。

通过 1 位或者 8 位存储器操作指令设定 CKS0 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 8-2 时钟输出选择寄存器 0 (CKS0) 的格式

地址: FFFA5H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 引脚输出允许 / 禁止的指定
0	禁止输出 (默认值)。
1	允许输出。

CCS02	CCS01	CCS00	PCLBUZ0 引脚输出时钟的选择					
			$f_{MAIN}(MHz)$					
			1.25	2.5	5	10	20	
0	0	0	f_{MAIN}	1.25MHz	2.5MHz	5MHz 注	10MHz 注	禁止设定注
0	0	1	$f_{MAIN}/2$	625kHz	1.25MHz	2.5MHz	5MHz 注	10MHz 注
0	1	0	$f_{MAIN}/2^2$	312.5kHz	625kHz	1.25MHz	2.5MHz	5MHz 注
0	1	1	$f_{MAIN}/2^3$	156.3kHz	312.5kHz	625kHz	1.25MHz	2.5MHz
1	0	0	$f_{MAIN}/2^4$	78.1kHz	156.3kHz	312.5kHz	625kHz	1.25MHz
1	0	1	$f_{MAIN}/2^{11}$	610Hz	1.22kHz	2.44kHz	4.88kHz	9.77kHz
1	1	0	$f_{MAIN}/2^{12}$	305Hz	610Hz	1.22kHz	2.44kHz	4.88kHz
1	1	1	$f_{MAIN}/2^{13}$	153Hz	305Hz	610Hz	1.22kHz	2.44kHz

注 能选择的输出时钟因工作电压范围而不同。详细内容请参照“24.4 AC 特性”。

注意 1. 输出时钟的切换必须在设定为禁止 PCLBUZ0 引脚的输出 (PCLOE0=0) 后进行。

2. 如果要转移到 STOP 模式, 就必须先将 PCLBUZ0 引脚设定为禁止输出 (PCLOE0=0), 再经过 1.5 个 PCLBUZ0 的输出时钟后, 执行 STOP 指令。

备注 f_{MAIN} : 主系统时钟频率

8.3.2 时钟输出 / 蜂鸣器输出引脚的端口功能的控制寄存器

在使用时钟输出 / 蜂鸣器输出控制电路时，必须设定与时钟输出 / 蜂鸣器输出引脚复用的端口功能的控制寄存器（端口模式寄存器（PM0/PM4）、端口寄存器（P0/P4）、端口模式控制寄存器 0（PMC0）和外围 I/O 重定向寄存器（PIOR））。

端口功能的控制寄存器的详细内容请参照“4.3.1 端口模式寄存器 0、4（PM0、PM4）”、“4.3.2 端口寄存器 0、4、12、13（P0、P4、P12、P13）”、“4.3.5 端口模式控制寄存器 0（PMC0）”、“4.3.6 外围 I/O 重定向寄存器（PIOR）”。

在用作 PCLBUZ0 引脚时，必须将对应的端口模式寄存器 0（PM0）和端口模式控制寄存器 0（PMC0）的位置“0”，将端口寄存器 0（P0）和端口输出模式寄存器 0（POM0）的位置“1”。详细内容请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

另外，在将外围 I/O 重定向寄存器（PIOR）的 PIOR0 位置“1”后，还能给 P40 引脚分配 PCLBUZ0 引脚输出功能。

8.4 时钟输出 / 蜂鸣器输出控制电路的运行

能用 1 个引脚选择用作时钟输出或者蜂鸣器输出。

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

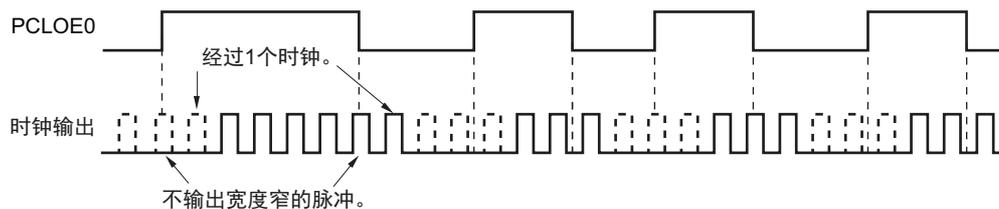
8.4.1 输出引脚的运行

PCLBUZ0 引脚按照以下步骤进行输出：

- ① 将 PCLBUZ0 引脚复用端口的端口模式寄存器 (PM0/PM4) 和端口寄存器 (P0/P4) 以及端口模式控制寄存器 0 (PMC0) 的对应位置“0”。
- ② 通过 PCLBUZ0 引脚的时钟输出选择寄存器 (CKS0) 的 bit0~2 (CCS00~CCS02) 选择输出频率 (输出为禁止状态)。
- ③ 将 CKS0 寄存器的 bit7 (PCLOE0) 置“1”，允许时钟输出 / 蜂鸣器输出。

备注 用作时钟输出时的控制电路在切换到允许或者禁止时钟输出 (PCLOE0 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 PCLOE0 位允许或者停止输出以及时钟输出的时序如图 8-3 所示。

图 8-3 PCLBUZ0 引脚的时钟输出时序



注意 如果在设定为禁止 PCLBUZ0 引脚输出 (PCLOE0=0) 后的 1.5 个 PCLBUZ0 引脚的输出时钟内转移到 STOP 模式，PCLBUZ0 引脚的时钟输出宽度就变窄。在设定为禁止 PCLBUZ0 引脚输出并且至少经过 1.5 个 PCLBUZ0 引脚的输出时钟后，才能执行 STOP 指令。

第 9 章 看门狗定时器

9.1 看门狗定时器的功能

看门狗定时器通过用户选项字节（000C0H）设定计数运行。

看门狗定时器以低速内部振荡器时钟运行。

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1位操作指令时
- 当给WDTE寄存器写“ACH”以外的数据时

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% 时，产生间隔中断。

9.2 看门狗定时器的结构

看门狗定时器由以下硬件构成。

表 9-1 看门狗定时器的结构

项目	结构
控制寄存器	看门狗定时器的允许寄存器 (WDTE)

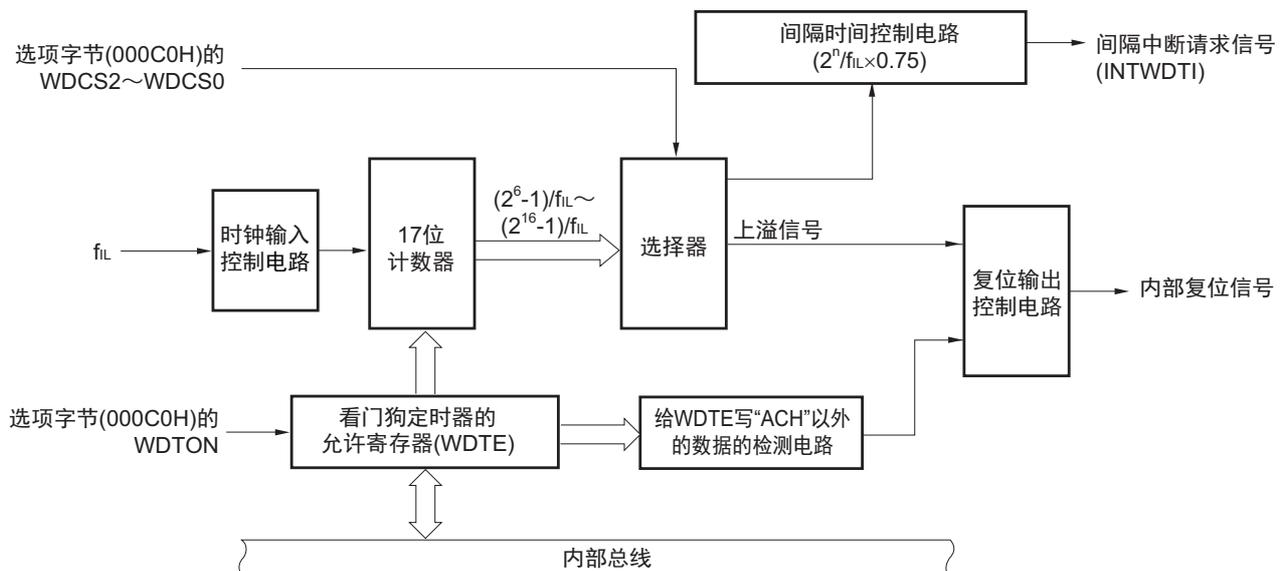
通过选项字节控制计数器的运行以及设定上溢时间。

表 9-2 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)
看门狗定时器的计数器运行控制	bit4 (WDTON)
看门狗定时器的上溢时间的设定	bit3 ~ 1 (WDCS2 ~ WDCS0)
看门狗定时器的计数器运行控制 (HALT/STOP 模式时)	bit0 (WDSTBYON)

备注 有关选项字节, 请参照“第 19 章 选项字节”。

图 9-1 看门狗定时器的框图



9.3 看门狗定时器的控制寄存器

通过看门狗定时器的允许寄存器（WDTE）控制看门狗定时器。

9.3.1 看门狗定时器的允许寄存器（WDTE）

通过给 WDTE 寄存器写“ACH”，清除看门狗定时器的计数器并且重新开始计数。

通过 8 位存储器操作指令设定 WDTE 寄存器。

在产生复位信号后，此寄存器的值变为“1AH”或者“9AH”注。

图 9-2 看门狗定时器的允许寄存器（WDTE）的格式

地址: FFFABH	复位后: 1AH/9AH 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 1 位操作指令时，产生内部复位信号。

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

9.4 看门狗定时器的运行

9.4.1 看门狗定时器的运行控制

- 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 19 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（WDSC2～WDSC0）设定上溢时间（详细内容请参照 9.4.2 和第 19 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。
 - 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 1 位操作指令时
 - 当给 WDTE 寄存器写“ACH”以外的数据时

注意 1. 在给 WDTE 寄存器写“ACH”并且清除了看门狗定时器的计数器时，实际的上溢时间可能比选项字节设定的上溢时间短最多 1 个 f_{IL} 时钟。

- 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 模式或中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

WDSTBYON=0：看门狗定时器停止运行

WDSTBYON=1：看门狗定时器继续运行

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 时钟^注运行时，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 时钟^注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。

- 禁止进行 WDTON=0、WDSTBYON=1 的设定。

注 只限于 16 引脚产品。

9.4.2 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和间隔中断时间。

在发生上溢时，产生内部复位信号。在上溢时间前通过给看门狗定时器的允许寄存器（WDTE）写“ACH”，清除计数并且重新开始计数。在达到上溢时间的大约 75% 时发生间隔中断。

能设定的上溢时间和间隔中断时间如下所示。

表 9-3 上溢时间和间隔中断时间的设定

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)	看门狗定时器的间隔中断时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$ 的情况)
0	0	0	$(2^6-1)/f_{IL}$ (3.65ms)	$2^6/f_{IL}\times 0.75$ (2.78ms)
0	0	1	$(2^7-1)/f_{IL}$ (7.36ms)	$2^7/f_{IL}\times 0.75$ (5.56ms)
0	1	0	$(2^8-1)/f_{IL}$ (14.7ms)	$2^8/f_{IL}\times 0.75$ (11.1ms)
0	1	1	$(2^9-1)/f_{IL}$ (29.6ms)	$2^9/f_{IL}\times 0.75$ (22.2ms)
1	0	0	$(2^{11}-1)/f_{IL}$ (118ms)	$2^{11}/f_{IL}\times 0.75$ (89.0ms)
1	0	1	$(2^{13}-1)/f_{IL}$ (474ms)	$2^{13}/f_{IL}\times 0.75$ (356ms)
1	1	0	$(2^{14}-1)/f_{IL}$ (949ms)	$2^{14}/f_{IL}\times 0.75$ (712ms)
1	1	1	$(2^{16}-1)/f_{IL}$ (3799ms)	$2^{16}/f_{IL}\times 0.75$ (2849ms)

注意 1. 当解除 STOP 模式后以 X1 时钟^注运行时，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，要在通过间隔中断解除 STOP 模式后以 X1 时钟^注运行并且清除看门狗定时器时，必须设定考虑振荡稳定时间的上溢时间。

2. 在产生 INTWDTI 后（给看门狗定时器的允许寄存器（WDTE）写“ACH”前）也继续计数。如果在上溢时间前没有写“ACH”，就产生内部复位信号。
3. 总是发生看门狗定时器的间隔中断。不使用看门狗定时器的间隔中断时，必须将 WDTIMK 位置“1”，禁止中断处理。

注 只限于 16 引脚产品。

备注 f_{IL} ：低速内部振荡器的时钟频率

第 10 章 A/D 转换器

A/D 转换器的模拟输入通道数因产品而不同。

- 10 引脚产品：4 个通道（ANI0～ANI3）
- 16 引脚产品：7 个通道（ANI0～ANI6）、内部基准电压注（0.815V(TYP.))

注 A/D 转换器和比较器不能同时使用内部基准电压。选择内部基准电压作为 A/D 转换器的转换对象时，比较器的基准电压不能设定为内部基准电压。

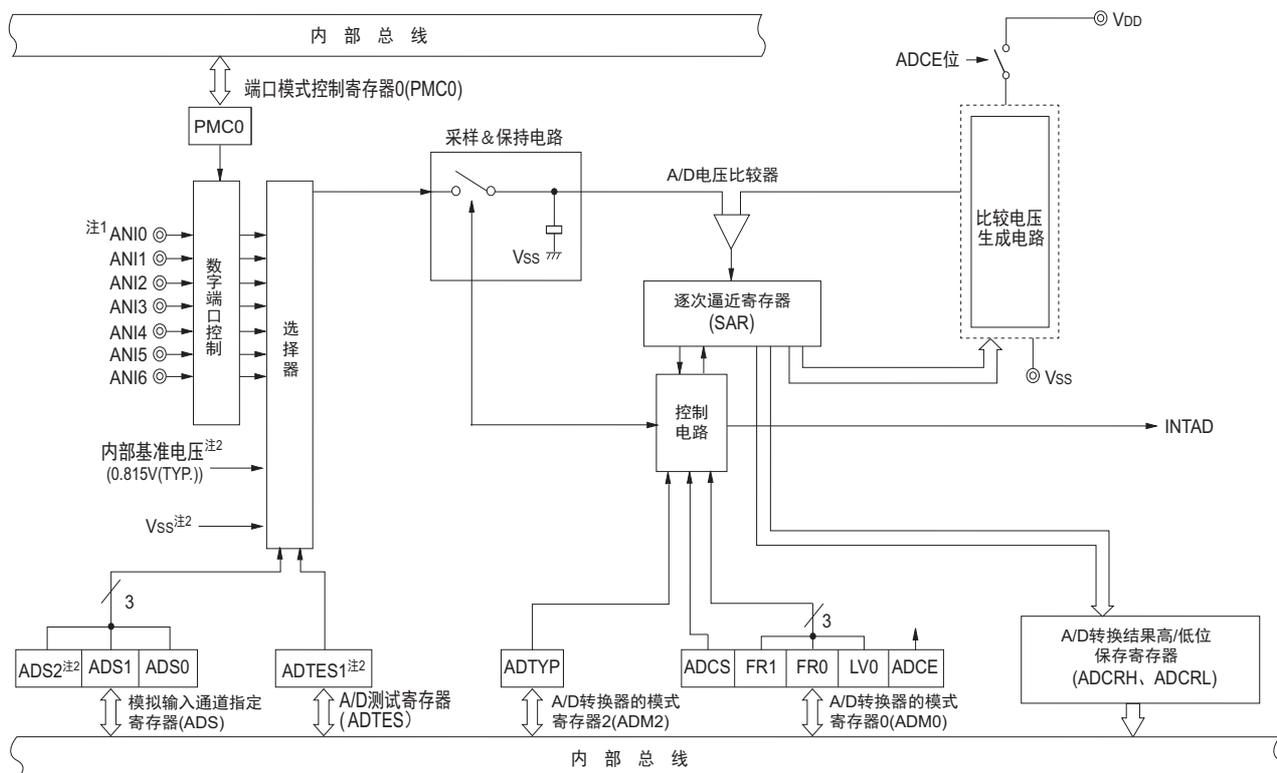
10.1 A/D 转换器的功能

A/D 转换器是将模拟输入转换为数字值的转换器，最多能控制 7 个通道的模拟输入。通过 A/D 转换器的模式寄存器 2（ADM2）的 ADTYP 位，选择 A/D 转换器的 10 位分辨率或者 8 位分辨率。

A/D 转换器有以下功能。

- 10 位/8 位分辨率的 A/D 转换
从 ANI0～ANI6 中选择 1 个通道的模拟输入，通过软件操作开始 10 位/8 位分辨率的 A/D 转换。在 A/D 转换结束时，产生 A/D 转换结束中断请求信号（INTAD）。A/D 转换器的工作电压为 2.4V～5.5V。

图 10-1 A/D 转换器的框图



- 注 1. 10 引脚产品时为 ANI0 ~ ANI3。
 注 2. 只限于 16 引脚产品。

10.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

(1) ANI0 ~ ANI6 注

这些是 A/D 转换器的 7 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。

A/D 转换器的转换对象除了 ANI0 ~ ANI6 的模拟输入，还能选择内部基准电压 (0.815V(TYP.))。

注 10 引脚产品时为 ANI0 ~ ANI3。

(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。如果比较结果是模拟输入电压大于基准电压 ($1/2V_{DD}$)，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 ($1/2V_{DD}$)，就对 SAR 寄存器的 MSB 位进行复位。

然后，自动将 SAR 寄存器的 bit8 置位并且进行下一次比较。在此，根据 bit9 被设定的结果值，选择比较电压生成电路的分接头电压。

bit9=0: ($1/4V_{DD}$)

bit9=1: ($3/4V_{DD}$)

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 bit8。

模拟输入电压 \geq 比较电压生成电路的分接头电压: bit8=1

模拟输入电压 \leq 比较电压生成电路的分接头电压: bit8=0

继续此比较直到 SAR 寄存器的 bit0 为止。

当以 8 位分辨率进行 A/D 转换时，继续此比较直到 SAR 寄存器的 bit2 为止。

(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) 开始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 转换结果高位保存寄存器 (ADCRH) 和 A/D 转换结果低位保存寄存器 (ADCRL)。而且, 如果 A/D 转换都结束, 就产生 A/D 转换结束中断请求信号 (INTAD)。

(6) A/D 转换结果高位保存寄存器 (ADCRH)

保存 A/D 转换结果的 8 位寄存器。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。10 位分辨率时, 将 A/D 转换结果的低 2 位保存到 ADCRL 寄存器。

(7) A/D 转换结果低位保存寄存器 (ADCRL)

保存 10 位分辨率时的低 2 位的转换结果 (ADCR1、ADCR0) 的 8 位寄存器。该寄存器的低 6 位固定为“0”。

(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, 产生 A/D 转换结束中断请求信号 (INTAD)。

10.3 A/D 转换器的控制寄存器

A/D 转换器的控制寄存器如下所示。

- 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 2 (ADM2)
- A/D 转换结果高位保存寄存器 (ADCRH)
- A/D 转换结果低位保存寄存器 (ADCRL)
- 模拟输入通道指定寄存器 (ADS)
- A/D 测试寄存器 (ADTES)
- 端口模式寄存器 0 (PM0)
- 端口模式控制寄存器 0 (PMC0)

10.3.1 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，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 10-2 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	TMKAEN 注	CMPEN 注	ADCEN	IICA0EN 注	0	SAU0EN	0	TAU0EN

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

注 只限于 16 引脚的产品。

注意 1. 在设定 A/D 转换器时，必须在 ADCEN 位为“1”的状态下设定以下寄存器。如果 ADCEN 位为“0”，A/D 转换器的控制寄存器都为初始值，并且忽视写操作（端口模式寄存器 0 (PM0) 和端口模式控制寄存器 0 (PMC0) 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 2 (ADM2)
- A/D 转换结果高位保存寄存器 (ADCRH)
- A/D 转换结果低位保存寄存器 (ADCRL)
- 模拟输入通道指定寄存器 (ADS)
- A/D 测试寄存器 (ADTES)

2. 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

10.3.2 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 10-3 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	0	0	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 (转换停止状态 / 转换待机状态)。
1	允许转换运行 (转换运行状态)。
< 清除条件 > • 给 ADCS 位写“0”时 • A/D 转换结束时自动清“0” < 置位条件 > • 在 ADCE=1 的情况下给 ADCS 位写“1”时	

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。

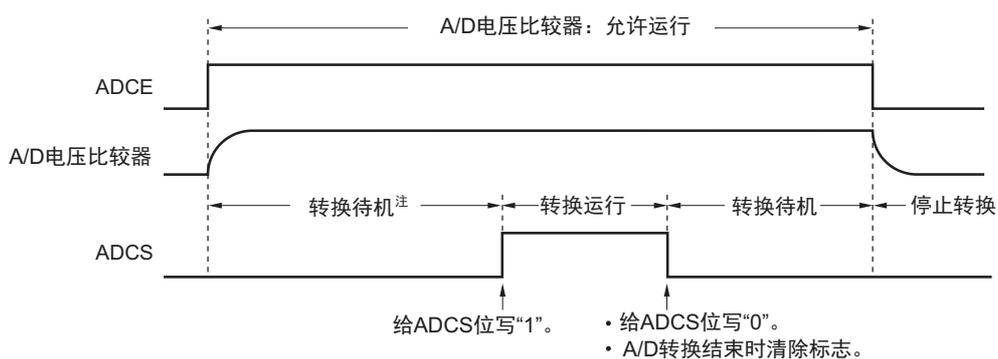
- 注 1. 有关 FR1、FR0、LV0 位和 A/D 转换的详细内容，请参照“表 10-2 10 位分辨率 A/D 转换时间的选择”和“表 10-3 8 位分辨率 A/D 转换时间的选择”。
2. 通过 ADCS 位和 ADCE 位控制 A/D 电压比较器的运行，从开始运行到稳定需要 0.1μs。因此，在将 ADCE 位置“1”并且至少经过 0.1μs 后，再将 ADCS 位置“1”，就从第 1 个转换数据开始有效。如果不等待 0.1μs 就将 ADCS 位置“1”开始 A/D 转换时，就忽视转换数据。

- 注意 1. 必须在转换待机状态 (ADCS=0、ADCE=1) 或者转换停止状态 (ADCS=0、ADCE=0) 下改写 FR1、FR0 和 LV0 位。另外，禁止通过 8 位操作指令同时改写 FR1、FR0、LV0 位和 ADCS 位。
2. 禁止进行 ADCS=1、ADCE=0 的设定。在转换停止状态 (ADCS=0、ADCE=0) 下，即使给 ADCS 位写“1”，也不被置“1”。
3. 禁止通过 8 位操作指令将 ADCS=0、ADCE=0 的状态设定为 ADCS=1、ADCE=1。必须按照“10.7 A/D 转换器的设定流程图”的步骤进行设定。
4. 必须将 bit2、5、6 置“0”。
5. 禁止在转换运行状态 (ADCS=1) 下再次将 ADCS 位置“1”。如果在转换运行状态 (ADCS=1) 下需要再次进行相同通道的转换时，必须先停止转换运行 (ADCS=0)，然后重新开始 A/D 转换运行 (ADCS=1)。

表 10-1 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

图 10-4 使用 A/D 电压比较器时的时序图



注 从允许 A/D 电压转换器的运行 (ADCE=1) 到开始 A/D 转换 (ADCS=1)，至少需要等待 0.1 μ s 的时间，以确保内部电路稳定。如果不等待 0.1 μ s 就将 ADCS 位置“1”，则必须忽视第 1 个转换数据。

表 10-2 10 位分辨率 A/D 转换时间的选择

A/D 转换器的模式 寄存器 0 (ADM0)			转换时钟	转换时钟数	转换时间	转换时间的选择例子 (μs)				
FR1	FR0	LV0 注 2				$f_{\text{CLK}}=$ 1.25MHz	$f_{\text{CLK}}=$ 2.5MHz	$f_{\text{CLK}}=$ 5MHz	$f_{\text{CLK}}=$ 10MHz	$f_{\text{CLK}}=$ 20MHz 注 1
0	0	0	$f_{\text{CLK}}/8$	23 个 f_{AD} (采样时钟数: 9 个 f_{AD})	$184/f_{\text{CLK}}$	禁止设定	禁止设定	禁止设定	18.4	9.2
0	1		$f_{\text{CLK}}/4$		$92/f_{\text{CLK}}$			18.4	9.2	4.6
1	0		$f_{\text{CLK}}/2$		$46/f_{\text{CLK}}$		18.4	9.2	4.6	禁止设定
1	1		f_{CLK}		$23/f_{\text{CLK}}$	18.4	9.2	4.6	禁止设定	
0	0	1 注 1	$f_{\text{CLK}}/8$	17 个 f_{AD} (采样时钟数: 3 个 f_{AD})	$136/f_{\text{CLK}}$	禁止设定	禁止设定	禁止设定	13.6	6.8
0	1		$f_{\text{CLK}}/4$		$68/f_{\text{CLK}}$			13.6	6.8	3.4
1	0		$f_{\text{CLK}}/2$		$34/f_{\text{CLK}}$		13.6	6.8	3.4	禁止设定
1	1		f_{CLK}		$17/f_{\text{CLK}}$	13.6	6.8	3.4	禁止设定	

注 1. 在 $2.4\text{V} \leq V_{\text{DD}} < 2.7\text{V}$ 时禁止设定。在 $2.7\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ 时能选择。

2. 选择内部基准电压为 A/D 转换器的转换对象时，必须将 LV0 位置“0”。

表 10-3 8 位分辨率 A/D 转换时间的选择

A/D 转换器的模式 寄存器 0 (ADM0)			转换时钟	转换时钟数	转换时间	转换时间的选择例子 (μs)				
FR1	FR0	LV0 注 2				$f_{\text{CLK}}=$ 1.25MHz	$f_{\text{CLK}}=$ 2.5MHz	$f_{\text{CLK}}=$ 5MHz	$f_{\text{CLK}}=$ 10MHz	$f_{\text{CLK}}=$ 20MHz 注 1
0	0	0	$f_{\text{CLK}}/8$	21 个 f_{AD} (采样时钟数: 9 个 f_{AD})	$168/f_{\text{CLK}}$	禁止设定	禁止设定	禁止设定	16.8	8.4
0	1		$f_{\text{CLK}}/4$		$84/f_{\text{CLK}}$			16.8	8.4	4.2
1	0		$f_{\text{CLK}}/2$		$42/f_{\text{CLK}}$		16.8	8.4	4.2	禁止设定
1	1		f_{CLK}		$21/f_{\text{CLK}}$	16.8	8.4	4.2	禁止设定	
0	0	1 注 1	$f_{\text{CLK}}/8$	15 个 f_{AD} (采样时钟数: 3 个 f_{AD})	$120/f_{\text{CLK}}$	禁止设定	禁止设定	禁止设定	12.0	6.0
0	1		$f_{\text{CLK}}/4$		$60/f_{\text{CLK}}$			12.0	6.0	3.0
1	0		$f_{\text{CLK}}/2$		$30/f_{\text{CLK}}$		12.0	6.0	3.0	禁止设定
1	1		f_{CLK}		$15/f_{\text{CLK}}$	12.0	6.0	3.0	禁止设定	

注 1. 在 $2.4\text{V} \leq V_{\text{DD}} < 2.7\text{V}$ 时禁止设定。在 $2.7\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ 时能选择。

2. 选择内部基准电压为 A/D 转换器的转换对象时，必须将 LV0 位置“0”。

注意 1. 必须在“24.6.1 A/D 转换器特性”中所示的转换时间 (t_{CONV}) 范围内使用 A/D 转换时间。

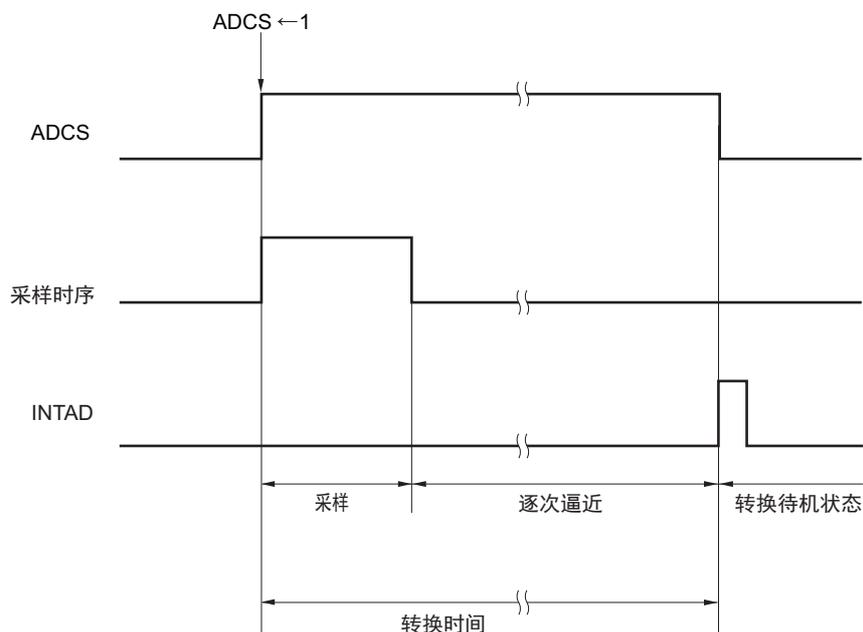
2. 选择内部基准电压为 A/D 转换器的转换对象时，比较器的基准电压就不能使用内部基准电压。

3. 要将 FR1、FR0 和 LV0 位改写为不同数据时，必须在转换待机状态 (ADCS=0、ADCE=1) 或者转换停止状态 (ADCS=0、ADCE=0) 下进行。另外，禁止通过 8 位操作指令同时改写 FR1、FR0、LV0 和 ADCS 位。

3. 另外，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。

备注 f_{CLK} : CPU/ 外围硬件的时钟频率

图 10-5 A/D 转换器的采样和 A/D 转换时序



10.3.3 A/D 转换器的模式寄存器 2 (ADM2)

这是设定 A/D 转换器的分辨率的寄存器。

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 10-6 A/D 转换器的模式寄存器 2 (ADM2) 的格式

地址: F0010H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADM2	0	0	0	0	0	0	0	ADTYP

ADTYP	A/D 转换分辨率的选择
0	10 位分辨率
1	8 位分辨率

注意 必须在转换停止状态 (ADCS=0、ADCE=0) 下改写 ADM2 寄存器。

10.3.4 A/D 转换器结果高位保存寄存器 (ADCRH)

这是保存 A/D 转换结果的 8 位寄存器。如果 A/D 转换结束，就将逐次逼近寄存器 (SAR) 的转换结果装入 ADCRH 寄存器。10 位分辨率时，保存高 8 位，低 2 位保存在 ADCRL 寄存器。

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”。

图 10-7 A/D 转换结果高位保存寄存器 (ADCRH) 的格式

地址: FFF1FH	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
ADCRH	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2

注意 当写 A/D 转换器的模式寄存器 0 (ADM0) 和模拟输入通道指定寄存器 (ADS) 时，ADCRH/ADCRL 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS 寄存器前读转换结果。否则，就可能读不到正确的转换结果。

10.3.5 A/D 转换结果低位保存寄存器 (ADCRL)

这是保存 10 位分辨率时的低 2 位的转换结果的 8 位寄存器。该寄存器的低 6 位固定为“0”。

通过 8 位存储器操作指令读 ADCRL 寄存器。

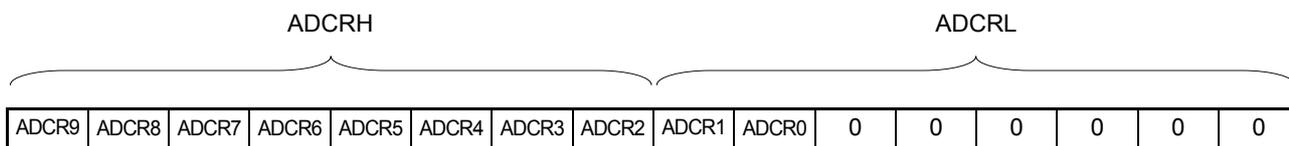
在产生复位信号后，此寄存器的值变为“00H”。

图 10-8 A/D 转换结果低位保存寄存器 (ADCRL) 的格式

地址: FFF1EH	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
ADCRL	ADCR1	ADCR0	0	0	0	0	0	0

10 位分辨率时的 A/D 转换结果的保存如图 10-9 所示。每当 A/D 转换结束时，就将逐次逼近寄存器 (SAR) 的转换结果装入 ADCRH 寄存器和 ADCRL 寄存器。转换结果的高 8 位保存在 ADCRH 寄存器，低 2 位保存在 ADCRL 寄存器的高 2 位。

图 10-9 10 位分辨率时的 A/D 转换结果的保存



注意 1. 当写 A/D 转换器的模式寄存器 0 (ADM0) 和模拟输入通道指定寄存器 (ADS) 时，ADCRH/ADCRL 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS 寄存器前读转换结果。否则，就可能读不到正确的转换结果。

2. 如果在选择 8 位分辨率 A/D 转换 (A/D 转换器的模式寄存器 2 (ADM2) 的 ADTYP=1) 时读 ADCRL 寄存器，高 2 位 (ADCR1、ADCR0) 的读取值就为“0”。但是，当选择 8 位分辨率 A/D 转换后立即在 A/D 转换没有结束的情况下读 ADCRL 寄存器时，高 2 位 (ADCR1、ADCR0) 的读取值有可能不为“0”。

10.3.6 模拟输入通道指定寄存器 (ADS)

这是指定要进行 A/D 转换的模拟电压输入通道的寄存器。

通过 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 10-10 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	0	0	0	0	0	ADS2 注 1	ADS1	ADS0

10 引脚产品的情况

ADS1	ADS0	A/D 转换对象	模拟输入引脚
0	0	ANI0	P01/ANI0 引脚
0	1	ANI1	P02/ANI1 引脚
1	0	ANI2	P03/ANI2 引脚
1	1	ANI3	P04/ANI3 引脚

16 引脚产品的情况

ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	ANI0	P01/ANI0 引脚
0	0	1	ANI1	P02/ANI1 引脚
0	1	0	ANI2	P03/ANI2 引脚
0	1	1	ANI3	P04/ANI3 引脚
1	0	0	ANI4	P05/ANI4 引脚
1	0	1	ANI5	P06/ANI5 引脚
1	1	0	ANI6	P07/ANI6 引脚
1	1	1	内部基准电压 (0.815(TYP.)) 注 2	—

注 1. 只限于 16 引脚产品。

2. 选择内部基准电压为 A/D 转换器的转换对象时，必须将 A/D 转换器的模式寄存器 0 (ADM0) 的 LV0 位置“0”。

注意 1. 必须在转换待机状态 (ADCS=0、ADCE=1) 或者转换停止状态 (ADCS=0、ADCE=0) 下改写 ADS 寄存器。

2. 通过端口模式寄存器 0 (PM0) 将用作模拟输入端口的端口设定为输入模式，通过端口模式控制寄存器 0 (PMC0) 设定为模拟输入。对于通过端口模式控制寄存器 0 (PMC0) 设定为数字输入 / 输出的引脚，不能通过 ADS 寄存器进行设定。

3. A/D 转换器和比较器不能同时使用内部基准电压。选择内部基准电压为 A/D 转换器的转换对象 (ADS2 ~ ADS0 为“111B”) 时，比较器的基准电压就不能设定为内部基准电压。

4. 必须将以下位置“0”。

10 引脚产品: bit2 ~ 7

16 引脚产品: bit3 ~ 7

10.3.7 A/D 测试寄存器 (ADTES)

这是选择 V_{SS} 作为 A/D 转换的模拟输入的寄存器。如果选择内部基准电压 (0.815(TYP.)) 作为 A/D 转换对象, 在内部基准电压的 A/D 转换前需要对采样电容器进行放电转换。将 ADTES 寄存器的 ADTES1 位置“1”, 进行 1 次 A/D 转换。

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 10-11 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	0

ADTES1	A/D 转换对象的选择
0	ANIX/ 内部基准电压 (0.815V(TYP.)) (通过模拟输入通道指定寄存器 (ADS) 进行设定)
1	V_{SS} (采样电容器的放电转换)

注意 对内部基准电压 (0.815(TYP.)) 进行 A/D 转换时, 必须按照“10.7.2 A/D 转换对象为内部基准电压的设定 (只限于 16 引脚产品)”执行 1 次采样电容器的放电转换。

备注 必须将 bit0、bit2 ~ 7 置“0”。

10.3.8 模拟输入引脚的端口功能的控制寄存器

必须设定与 A/D 转换器的模拟输入复用的端口功能的控制寄存器 (端口模式寄存器 0 (PM0) 和端口模式控制寄存器 0 (PMC0))。详细内容请参照“4.3.1 端口模式寄存器 0、4 (PM0、PM4)”、“4.3.5 端口模式控制寄存器 0 (PMC0)”。

用作 A/D 转换器的模拟输入引脚时的设定例子, 请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

将 ANI0 ~ ANI6 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 0 (PM0) 和端口模式控制寄存器 0 (PMC0) 的位置“1”。

10.4 A/D 转换器的转换运行

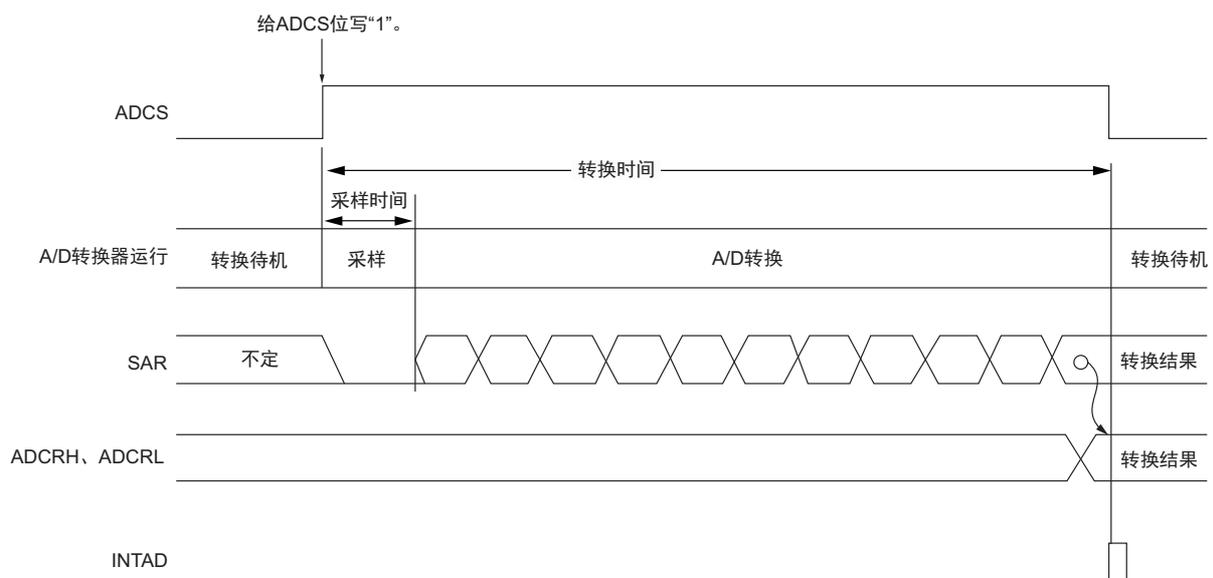
A/D 转换器的转换运行如下所示。

- 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 如果进行了一定时间的采样，采样 & 保持电路就为保持状态，并且保持采样电压，直到 A/D 转换结束为止。
- ③ 将逐次逼近寄存器（SAR）的 bit9 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 $(1/2)V_{DD}$ 。
- 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 $(1/2)V_{DD}$ ，SAR 寄存器的 MSB 位就保持被置位的状态。如果模拟输入小于 $(1/2)V_{DD}$ ，就对 MSB 位进行复位。
- ⑤ 然后，SAR 寄存器的 bit8 自动置位并且进行下一次比较。在此，根据已被设定的 bit9 的结果值选择串联电阻串的分接头电压。
 - bit9=1: $(3/4)V_{DD}$
 - bit9=0: $(1/4)V_{DD}$
 将此分接头电压和采样电压进行比较，根据比较结果如下操作 SAR 寄存器的 bit8。
 - 采样电压 \geq 分接头电压: bit8=1
 - 采样电压 $<$ 分接头电压: bit8=0
- ⑥ 继续此比较直到 SAR 寄存器的 bit0 为止。
- ⑦ 当 10 位比较结束时，SAR 寄存器保存了有效的数字结果，并且将该值传送到 A/D 转换结果寄存器（ADCRH、ADCRL），而且被锁存。同时，能产生 A/D 转换结束中断请求信号（INTAD）。A/D 转换结束后，ADCS 位自动清“0”，进入 A/D 转换待机状态。

备注 有 2 种 A/D 转换结果寄存器。

- ADCRH 寄存器（8 位）：保存 10 位分辨率 A/D 转换结果的高 8 位，或者保存 8 位分辨率 A/D 转换值。
- ADCRL 寄存器（2 位）：保存 10 位分辨率 A/D 转换结果的低 2 位。

图 10-12 A/D 转换器的转换运行（软件触发模式的情况）



如果通过软件操作将 A/D 转换器的模式寄存器 0（ADM0）的 bit7（ADCS）置“1”，就能执行 1 次 A/D 转换运行。A/D 转换结束后，ADCS 位自动清“0”。

产生复位信号后，A/D 转换结果寄存器（ADCRH、ADCRL）的值变为“00H”。

10.5 输入电压和转换结果

输入模拟输入（ANI0 ~ ANI6^注、内部基准电压）的模拟输入电压和理论上的 A/D 转换结果（A/D 转换结果寄存器（ADCR=ADCRH+ADCRL））有以下表达式的关系。

$$\text{SAR} = \text{INT} \left(\frac{V_{\text{AIN}}}{V_{\text{DD}}} \times 1024 + 0.5 \right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

或者

$$\left(\frac{\text{ADCR}}{64} - 0.5 \right) \times \frac{V_{\text{DD}}}{1024} \leq V_{\text{AIN}} < \left(\frac{\text{ADCR}}{64} + 0.5 \right) \times \frac{V_{\text{DD}}}{1024}$$

INT(): 将 () 中的数值的整数部分返回的函数

V_{AIN} : 模拟输入电压

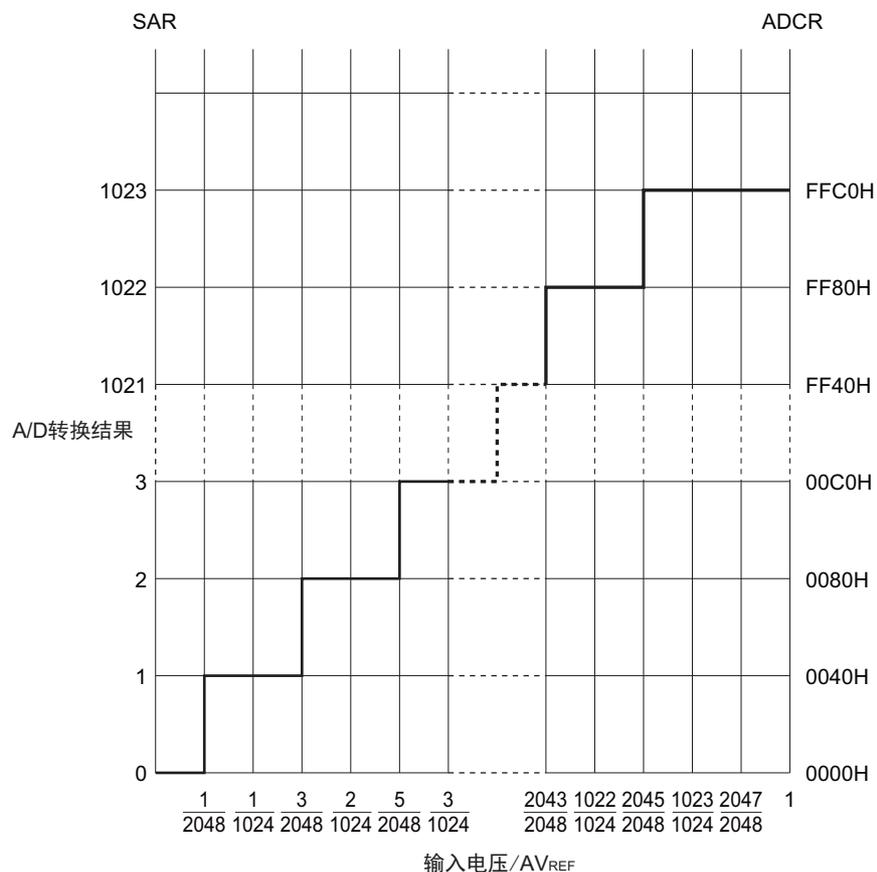
ADCR: A/D 转换结果寄存器（ADCRH+ADCRL）的值

SAR: 逐次逼近寄存器

注 10 引脚产品时为 ANI0 ~ ANI3。

模拟输入电压和 A/D 转换结果的关系如图 10-13 所示。

图 10-13 模拟输入电压和 A/D 转换结果的关系



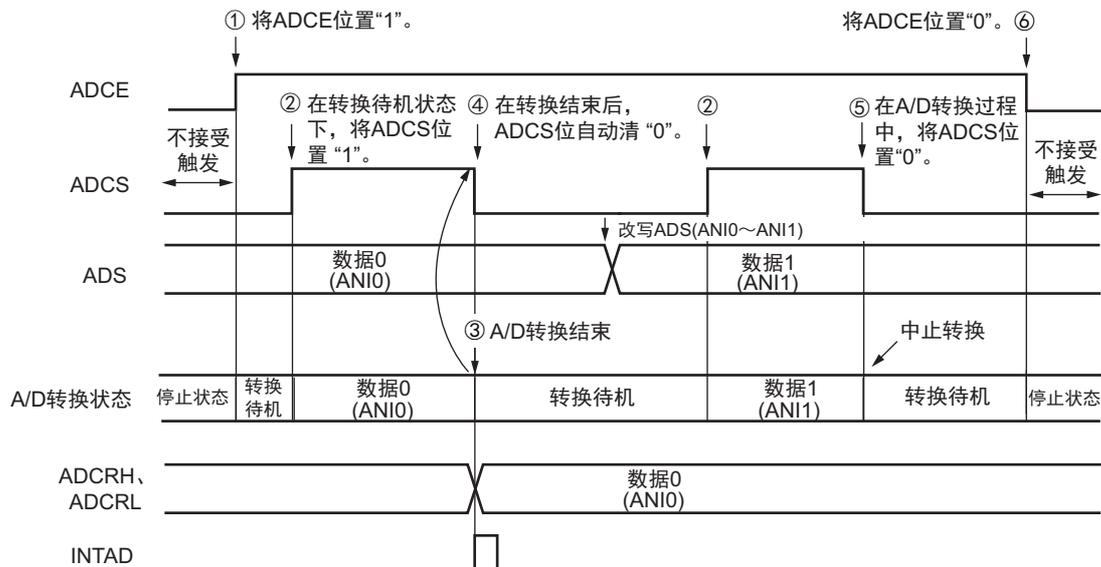
10.6 A/D 转换器的运行模式

A/D 转换器的运行如下所示。有关设定步骤，请参照“10.7 A/D 转换器的设定流程图”。

- ① 在转换停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入转换待机状态。
- ② 在通过软件对稳定等待时间（0.1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对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（ADCRH、ADCRL），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ④ A/D 转换结束后，ADCS 位自动清“0”，进入转换待机状态。
- ⑤ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入转换待机状态。
- ⑥ 如果在转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入转换停止状态。

禁止进行 ADCS=1、ADCE=0 的设定。在转换停止状态（ADCS=0、ADCE=0）下，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

图 10-14 运行时序例子

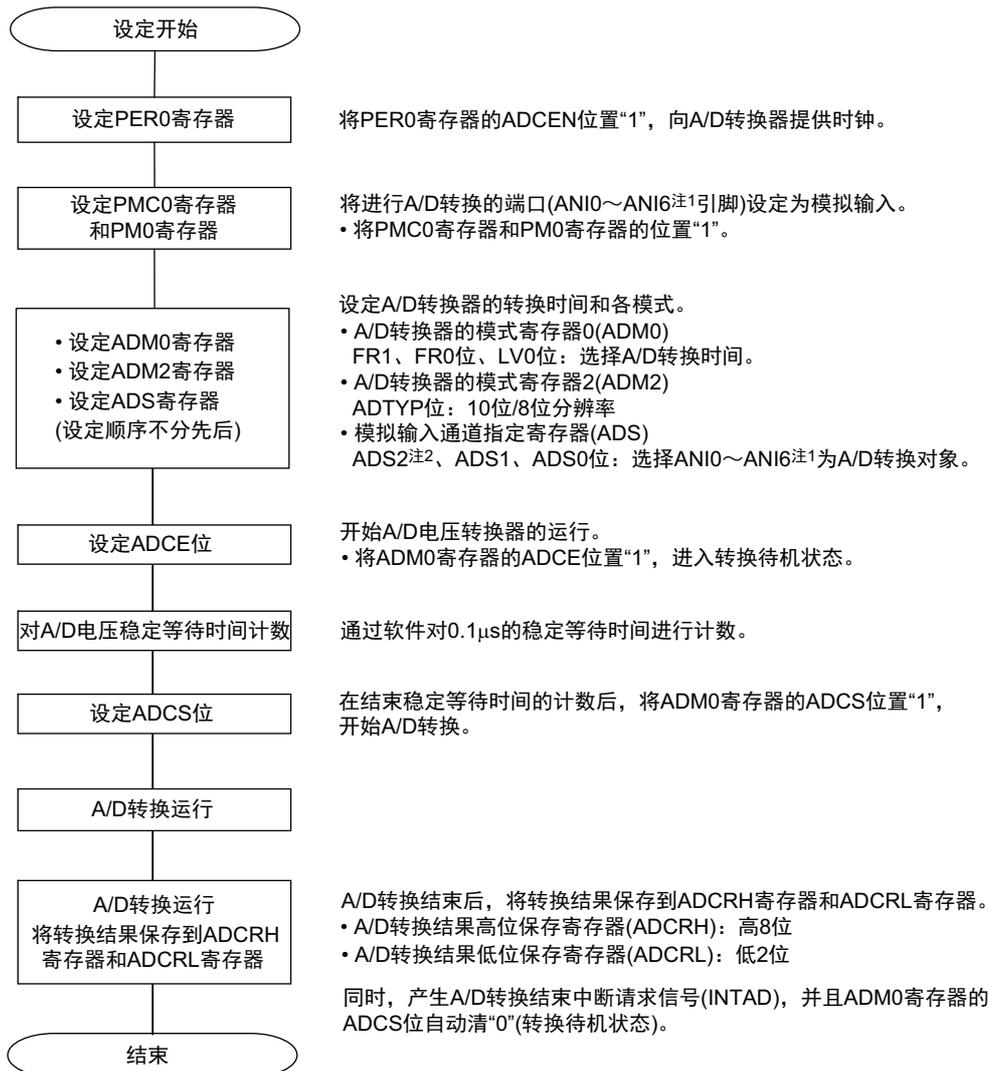


10.7 A/D 转换器的设定流程图

A/D 转换器的设定流程图如下所示。

10.7.1 A/D 转换对象为 ANI0 ~ ANI6 的设定

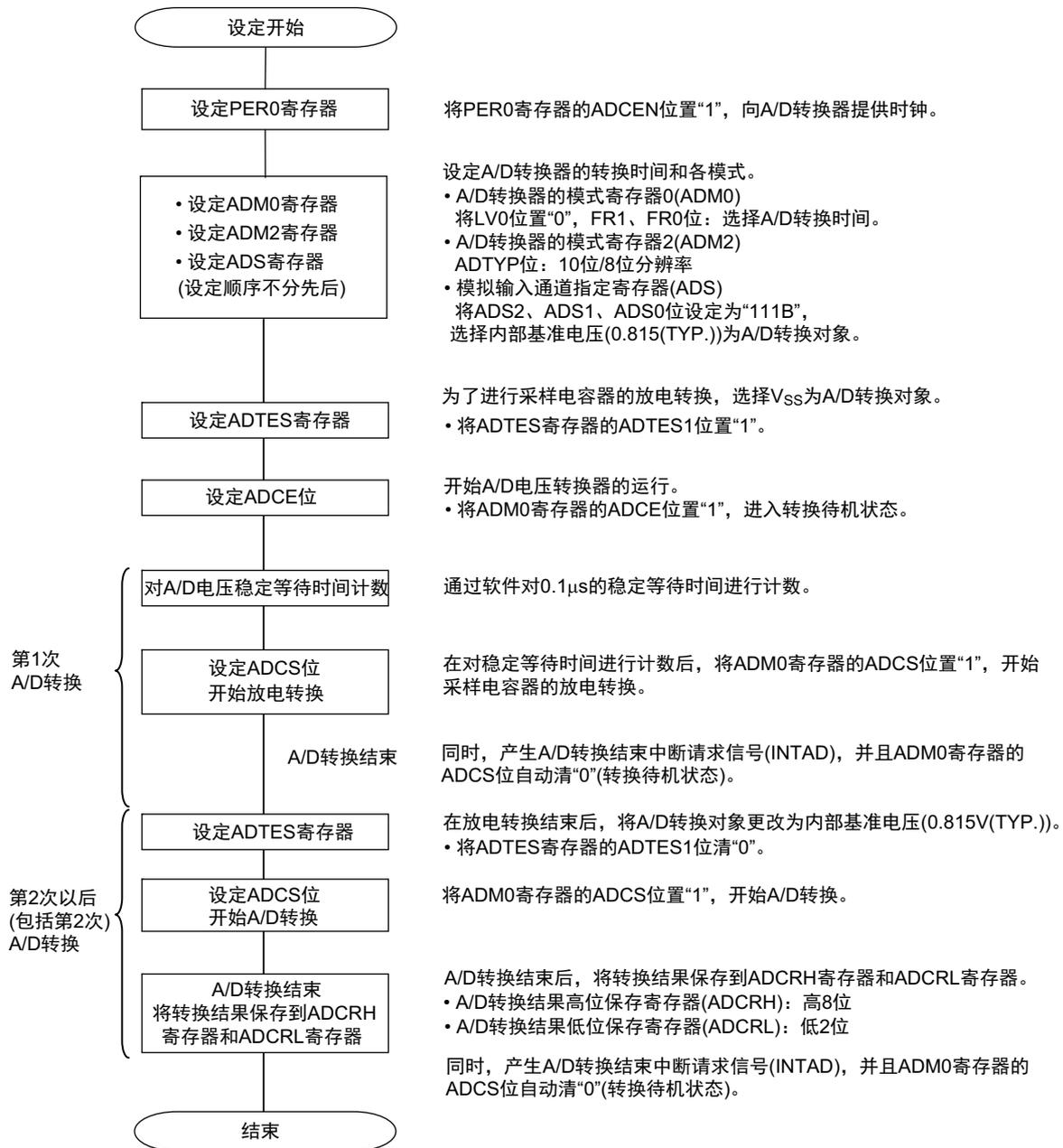
图 10-15 A/D 转换对象为 ANI0 ~ ANI6 的设定



- 注 1. 10 引脚产品时为 ANI0 ~ ANI3。
2. 只限于 16 引脚产品。

10.7.2 A/D 转换对象为内部基准电压的设定（只限于 16 引脚产品）

图 10-16 A/D 转换对象为内部基准电压的设定



10.8 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。

10.8.1 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 10 位时，

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\% \text{FSR} \end{aligned}$$

精度与分辨率无关而取决于综合误差。

10.8.2 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。

10.8.3 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 10-17 综合误差

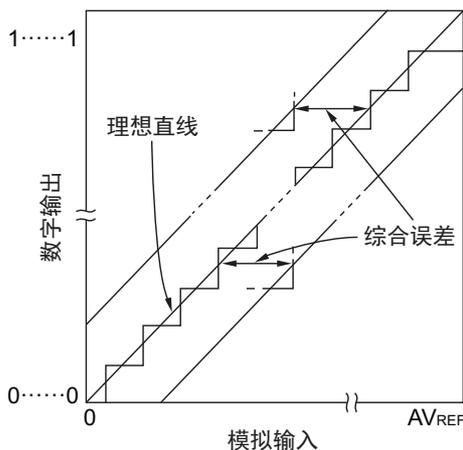
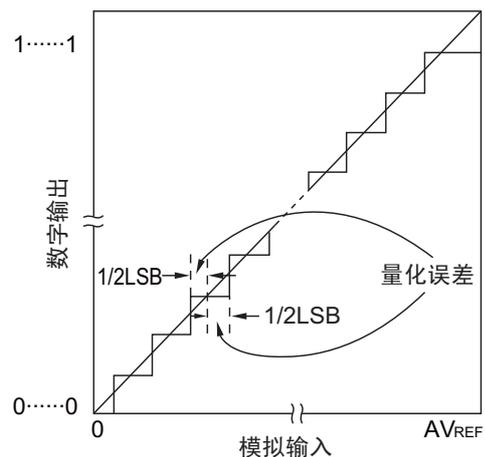


图 10-18 量化误差



10.8.4 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 LSB）的差。如果实际测量值大于理论值，零刻度误差就是指数码输出从 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 LSB）的差。

10.8.5 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 LSB）的差。

10.8.6 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最大差值。

10.8.7 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 1LSB 时某个代码输出宽度的实际测量值和理想值的差值。

图 10-19 零刻度误差

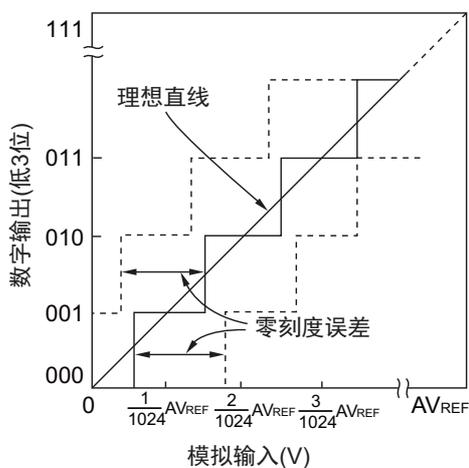


图 10-20 满刻度误差

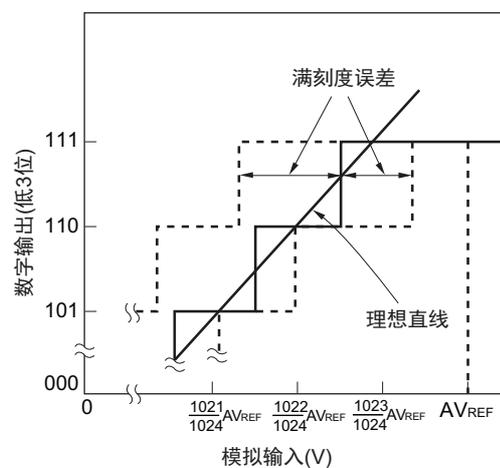


图 10-21 积分线性误差

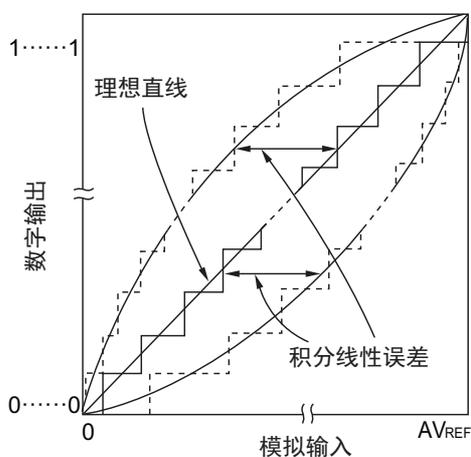
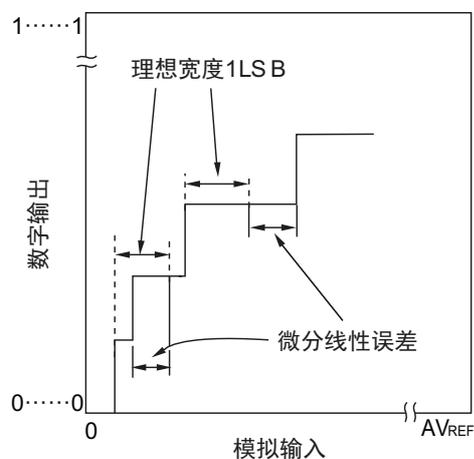


图 10-22 微分线性误差

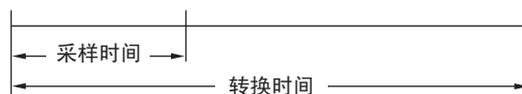


10.8.8 转换时间

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。

10.8.9 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开启的时间。



10.9 A/D 转换器的注意事项

10.9.1 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”来减少工作电流。

10.9.2 有关 ANI0 ~ ANI6 引脚的输入电压

必须在规格范围内使用 ANI0 ~ ANI6 注引脚的输入电压。如果输入大于 V_{DD} 或者小于 V_{SS} （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，并且还影响其他通道的转换值。

注 10 引脚产品时为 ANI0 ~ ANI3。

10.9.3 有关竞争

- ① 当转换结束时的 A/D 转换结果寄存器 (ADCRH、ADCRL) 的写操作与通过软件操作进行的 ADCRH、ADCRL 寄存器的读操作发生竞争时，优先读 ADCRH、ADCRL 寄存器。
在读后，将新的转换结果写到 ADCRH 寄存器和 ADCRL 寄存器。
- ② 当转换结束时的 ADCRH、ADCRL 寄存器的写操作与 A/D 转换器的模式寄存器 0 (ADM0) 的写操作发生竞争时，优先写 ADM0 寄存器。不写 ADCRH 寄存器和 ADCRL 寄存器，也不产生 A/D 转换结束中断请求信号 (INTAD)。

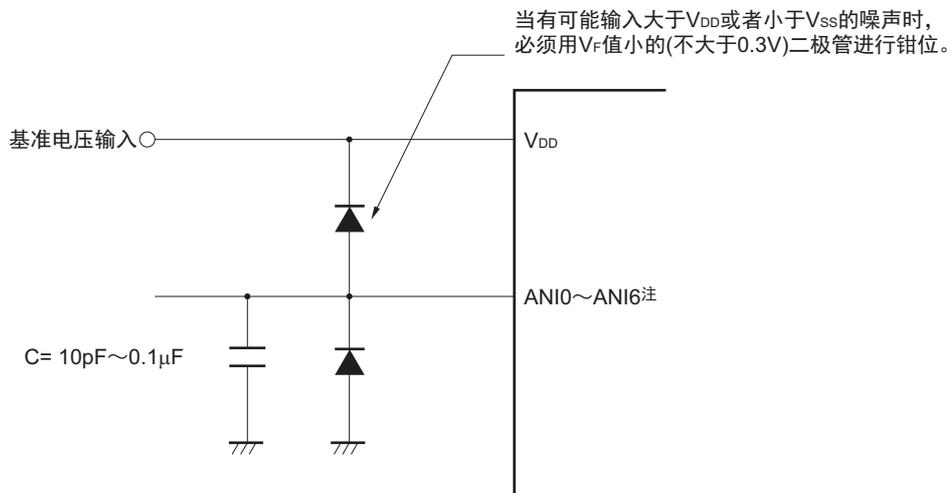
10.9.4 有关噪声对策

为了保持 10 位分辨率，必须注意输入到 V_{DD} 、ANI0 ~ ANI6 注引脚的噪声。

- ① 连接电源的电容器必须使用等效电阻小并且频率响应好的电容器。
- 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 10-23 的方法外接电容器。
- 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。

注 10 引脚产品时为 ANI0 ~ ANI3

图 10-23 模拟输入引脚的处理



注 10 引脚产品时为 ANI0 ~ ANI3。

10.9.5 模拟输入（ANIn）引脚

- ① 模拟输入（ANI0 ~ ANI6注1）引脚和输入端口（P01 ~ P07注2）引脚复用。
要选择 ANI0 ~ ANI6注1 中的任意引脚进行 A/D 转换时，不能在转换过程中更改 P01 ~ P07注2 的复用端口的输出值。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，不能输入或者输出会发生急剧变化（如数字信号）的脉冲。

注 1. 10 引脚产品时为 ANI0 ~ ANI3。

2. 只有 16 引脚产品有 P05 ~ P07。

10.9.6 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动。

但是，为了充分进行采样，必须将模拟输入源的输出阻抗保持在 $1\text{k}\Omega$ 以下。如果输出阻抗高，建议在 ANI0 ~ ANI6注引脚和接地之间连接 $0.1\mu\text{F}$ 左右的电容器（参照图 10-23）。

注 10 引脚产品时为 ANI0 ~ ANI3。

10.9.7 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”。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”。

10.9.8 有关 A/D 转换开始后的初次转换结果

如果在将 ADCE 位置“1”后的 0.1 μ s 内将 ADCS 位置“1”，开始 A/D 转换后的初次 A/D 转换值就可能不足额定值。必须采取轮询 A/D 转换结束中断请求信号（INTAD）并且取消初次的转换结果等对策。

10.9.9 有关 A/D 转换结果寄存器（ADCRH、ADCRL）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）和端口模式控制寄存器 0（PMC0）时，ADCRH 寄存器和 ADCRL 寄存器的内容可能不定，从而有可能无法读取正确的转换结果。必须在转换结束后并且在写 ADM0、ADS 和 PMC0 寄存器前读转换结果。

10.9.10 有关内部等效电路

模拟输入部的等效电路如下所示。

图 10-24 ANIn 引脚的内部等效电路

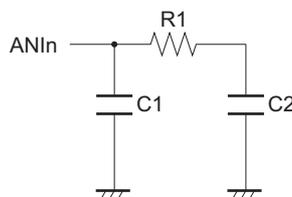


表 10-4 等效电路的各电阻和电容值

V_{DD}	引脚	R1[k Ω]	C1[pF]	C2[pF]
$2.7V \leq V_{DD} \leq 5.5V$	ANI0 ~ ANI6 注	40	8	1.7
$2.4V \leq V_{DD} < 2.7V$	ANI0 ~ ANI6 注	200		

注 10 引脚产品时为 ANI0 ~ ANI3。

备注 各电阻和电容值不是保证值。

10.9.11 有关 A/D 转换器的运行开始

A/D 转换器的工作电压为 2.4V ~ 5.5V。必须在 V_{DD} 的电压稳定后开始 A/D 转换器的运行。

第 11 章 比较器

注意 16 引脚产品内置 1 个通道的比较器。

11.1 比较器的功能

比较器有以下功能。

- 能选择比较器的响应速度
比较器高速模式：响应延迟时间变短但功耗变大。
比较器低速模式：响应延迟时间变长但功耗变小。
- 能选则外部基准电压/内部基准电压注（0.815V(TVP)）。
- 配置用于消除噪声的数字滤波器，并且能选择噪声的消除宽度。
- 反相/不反相比较器的输出，并且能从 VCOUT0 引脚输出。
- 能检测比较器输出的有效边沿，并且产生中断（INTCMP0）

注 内部基准电压不能同时用于 A/D 转换器和比较器。在选择内部基准电压作为比较器的基准电压时，A/D 转换器的转换对象就不能设定为内部基准电压。

11.2 比较器的结构

比较器由以下硬件构成。

(1) IVCMP0 引脚

这是比较器的模拟输入引脚。输入通过比较器比较的模拟信号。

(2) IVREF0 引脚

这是从外部提供基准电压的输入引脚。将比较器的基准电压和输入到 IVCMP0 引脚的模拟输入电压进行比较。

除了能选择从外部向 IVREF0 引脚提供的电压，还能选择内部基准电压（0.815V(TYP)）作为比较器的基准电压。

比较器的基准电压设定请参照“11.3.2 比较器模式设定寄存器（COMPMDR）”。

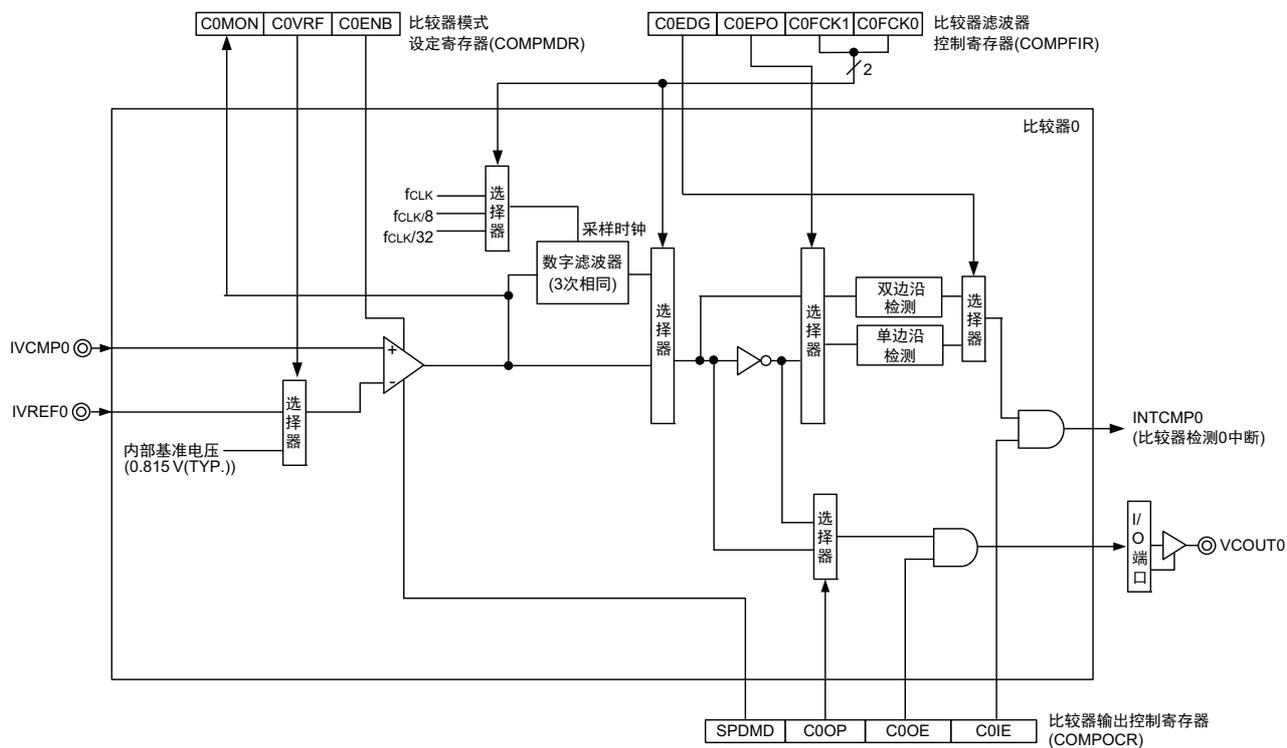
(3) VCOUT0 引脚

这是输出比较器的比较结果的引脚。能反相 / 不反相比较器输出，并且从 VCOUT0 引脚输出。

从 VCOUT0 引脚进行比较器输出时，请参照“11.4.3 比较器 0 输出”。

比较器的框图如图 11-1 所示。

图 11-1 比较器的框图



11.3 比较器的控制寄存器

比较器的控制寄存器如下所示。

- 外围允许寄存器0 (PER0)
- 比较器模式设定寄存器 (COMPMDR)
- 比较器的滤波器控制寄存器 (COMPFIR)
- 比较器输出控制寄存器 (COMPOCR)
- 端口模式控制寄存器0 (PMCO)
- 端口模式寄存器0 (PM0)
- 端口寄存器0 (P0)

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，降低功耗和噪声。

当使用比较器时，必须将 bit6 (CMPEN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

通过产生复位信号，此寄存器的值变为“00H”。

图 11-2 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	TMKAEN 注	CMPEN 注	ADCEN	IICA0EN 注	0	SAU0EN	0	TAU0EN

CMPEN	比较器输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写比较器使用的 SFR。 • 比较器处于复位状态。
1	提供输入时钟。 <ul style="list-style-type: none"> • 能读写比较器使用的 SFR。

注 只限于 16 引脚产品。

注意 1. 在设定比较器时，必须在 CMPEN 位为“1”的状态下，设定以下的寄存器。在 CMPEN 位为“0”时，比较器的控制寄存器为初始值，并且忽略写操作（端口模式寄存器 0 (PM0)、端口寄存器 0 (P0) 和端口模式控制寄存器 0 (PMCO) 除外）。

- 比较器模式设定寄存器 (COMPMDR)
- 比较器的滤波器控制寄存器 (COMPFIR)
- 比较器输出控制寄存器 (COMPOCR)

2. 必须将以下位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

11.3.2 比较器模式设定寄存器 (COMPMDR)

COMPMDR 寄存器选择比较器的基准电压、设定开始或者停止比较运行以及显示比较结果的状态。通过 1 位或者 8 位存储器操作指令设定 COMPMDR 寄存器。但是，COMON 位为只读位。通过产生复位信号，此寄存器的值变为“00H”。

图 11-3 比较器模式设定寄存器 (COMPMDR) 的格式

地址: FFF60H	复位后: 00H	R/W 注 1						
符号	7	6	5	4	3	2	1	0
COMPMDR	0	0	0	0	COMON	COVRF	0	COENB

COMON 注 2	比较器 0 监视标志
0	IVCMP0 < 比较器 0 的基准电压
1	IVCMP0 > 比较器 0 的基准电压

COVRF	比较器 0 的基准电压的选择
0	从 IVREF0 引脚提供
1	内部基准电压 (0.815V(TYP.)) 注 3。

COENB	比较器 0 运行的控制
0	禁止比较器 0 的运行。
1	允许比较器 0 的运行。

注 1. bit3 是只读位。

- 在设定为允许比较器 0 的运行 (COENB=1) 后，能通过 COMON 位确认 IVREF0 引脚的状态。此后，如果设定为禁止比较器 0 的运行 (COENB=0)，COMON 位的值就变为不定值。
- 选择内部基准电压 (0.815V(TYP.)) 作为比较器 0 的基准电压时，A/D 转换器就不能选择内部基准电压。

11.3.3 比较器的滤波器控制寄存器（COMPFIR）

COMPFIR 寄存器选择比较器中断信号的有效边沿，并且设定是否使用数字滤波器。

在需要消除噪声时，通过设定 C0FCK1 位和 C0FCK0 位，将数字滤波器置为有效。在数字滤波器有效时，通过数字滤波器的采样时钟进行 3 个时钟的一致检测。

通过 8 位存储器操作指令设定 COMPFIR 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 11-4 比较器的滤波器控制寄存器（COMPFIR）的格式

地址: FFF61H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
COMPFIR	0	0	0	0	C0EDG	C0EPO	C0FCK1	C0FCK0

C0EDG	C0EPO	比较器 0 中断信号的有效边沿的选择注 1
0	0	上升沿
0	1	下降沿
1	x	上升沿、下降沿的双边沿

C0FCK1	C0FCK0	是否使用比较器 0 的数字滤波器注 1、2、3
0	0	数字滤波器 OFF
0	1	数字滤波器 ON、采样时钟: f_{CLK}
1	0	数字滤波器 ON、采样时钟: $f_{CLK}/8$
1	1	数字滤波器 ON、采样时钟: $f_{CLK}/32$

- 注 1. 如果在允许比较器 0 的运行过程中更改 C0EDG 位、C0EPO 位和 C0FCK1 位、C0FCK0 位，就可能产生比较器 0 中断（INTCMP0）。如果要更改 C0EDG 位、C0EPO 位和 C0FCK1 位、C0FCK0 位，必须事先将 COMPOCR 寄存器的 C0IE 位清“0”，禁止中断请求。
2. 如果更改了 C0FCK1 位和 C0FCK0 位，在更新数字滤波器时就需要等待 4 个采样时钟。如果使用比较器 0 中断（INTCMP0），就在经过此等待后将 COMPOCR 寄存器的 C0IE 位置“1”。
3. 在 STOP 模式中使用比较器时，必须将数字滤波器设定为 OFF（C0FCK1、C0FCK0=00B）。

备注 x: 忽略

11.3.4 比较器输出控制寄存器 (COMPOCR)

COMPOCR 寄存器选择比较器的响应速度、控制 VCOUT0 输出、以及设定禁止或者允许中断请求信号。通过 1 位或者 8 位存储器操作指令设定 COMPOCR 寄存器。在产生复位信号后，此寄存器变为“00H”。

图 11-5 比较器输出控制寄存器 (COMPOCR) 的格式

地址: FFF62H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
COMPOCR	SPDMD	0	0	0	0	COOP	COOE	COIE

SPDMD 注 1	比较器速度的选择
0	比较器低速模式
1	比较器高速模式

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

COOE	允许 / 禁止 VCOUT0 引脚输出
0	禁止比较器 0 的 VCOUT0 引脚输出。
1	允许比较器 0 的 VCOUT0 引脚输出。

COIE	允许 / 禁止比较器 0 中断请求
0	禁止比较器 0 的中断请求。
1	允许比较器 0 的中断请求。

注 必须在将 COMPMDR 寄存器的 COENB 位置“0”后改写 SPDMD 位。

11.3.5 比较器的输入 / 输出引脚的端口功能控制寄存器

必须设定与比较器的输入 / 输出引脚复用的端口功能的控制寄存器 (端口模式寄存器 0 (PM0)、端口寄存器 0 (P0)、端口模式控制寄存器 0 (PMC0))。详细内容请参照“4.3.1 端口模式寄存器 0、4 (PM0、PM4)”、“4.3.2 端口寄存器 0、4、12、13 (P0、P4、P12、P13)”和“4.3.5 端口模式控制寄存器 0 (PMC0)”。另外，用作比较器的输入 / 输出引脚时的设定例子请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

将 IVCMP0 引脚和 IVREF0 引脚用作比较器的模拟输入时，必须将各端口对应的端口模式寄存器 (PM0) 和端口模式控制寄存器 (PMC0) 的位置“1”。

将 VCOUT0 引脚用作比较器的输出时，必须将端口模式寄存器 0 (PM0)、端口寄存器 0 (P0)、端口模式控制寄存器 0 (PMC0) 的位置“0”。另外，必须按照“11.4.3 比较器 0 输出”进行 VCOUT0 引脚的输出设定。

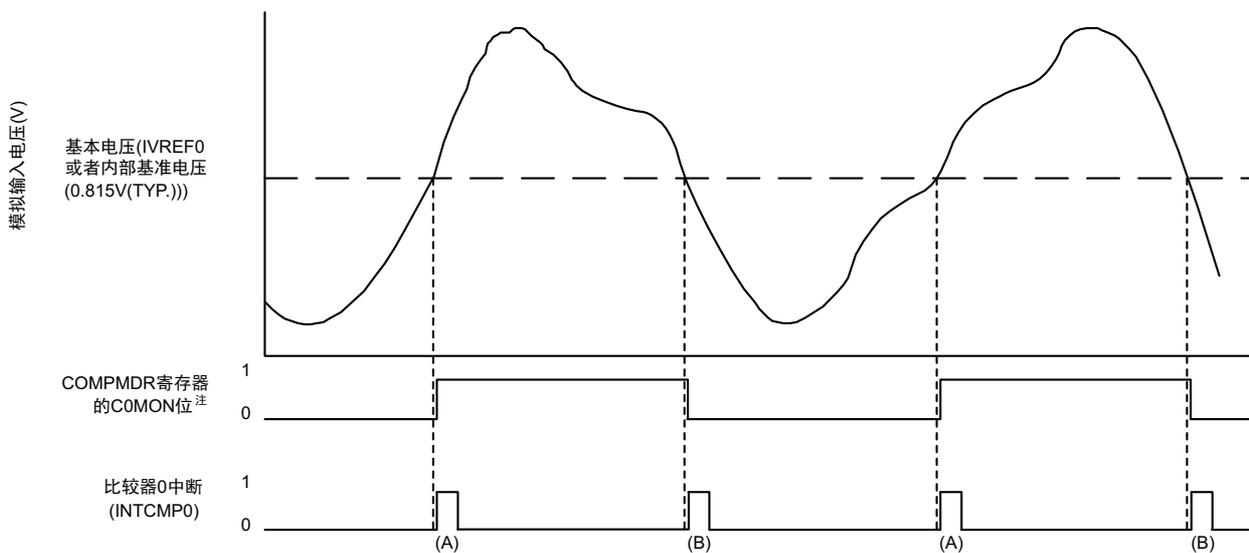
11.4 比较器的运行说明

当 IVCMP0 引脚的模拟输入电压大于基准电压时，COMPMDR 寄存器的 C0MON 位变为“1”。当 IVCMP0 引脚的模拟输入电压小于基准电压时，C0MON 位变为“0”。

使用比较器 0 中断 (INTCMP0) 时，必须将 COMPOCR 寄存器的 C0IE 位置“1”（允许中断请求）。此时，如果比较结果发生变化，就产生比较器 0 中断请求信号。比较器 0 中断的详细内容请参照“11.4.2 比较器 0 的中断运行”。

比较器 0 的运行例子（无数字滤波器（COMPFIR 寄存器的 C0FCK1、C0FCK0=00B）、中断双边沿检测（C0EDG=1））如图 11-6 所示。

图 11-6 比较器 0 的运行例子（无数字滤波器、中断双边沿检测）



注 输出延迟的时间取决于比较器运行模式。详细内容请参照“24.6.2 比较器特性”。

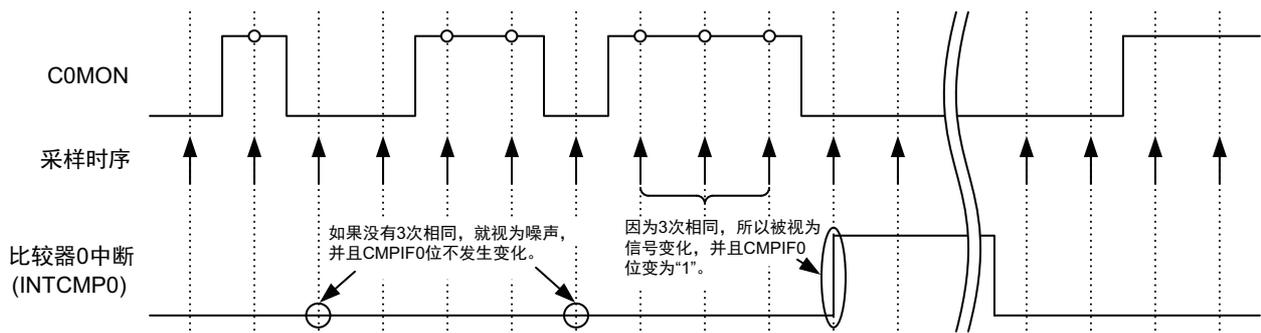
注意 将中断有效边沿设定为上升沿（C0EDG=0、C0EPO=0）时的 INTCMP0 只发生 (A) 的变化。
将中断有效边沿设定为下降沿（C0EDG=0、C0EPO=1）时的 INTCMP0 只发生 (B) 的变化。

11.4.1 比较器 0 的数字滤波器运行

比较器 0 内置数字滤波器。通过 COMPFIR 寄存器的 C0FCK1 位和 C0FCK0 位选择采样时钟。各采样时钟对比较器 0 的输出信号进行采样，在输出信号的电平 3 次相同后的下一个采样时钟时，数字滤波器输出发生变化。

比较器 0 的数字滤波器和中断运行例子如图 11-7 所示。

图 11-7 比较器 0 的数字滤波器和中断运行例子



备注 图 11-7 是有数字滤波器（COMPFIR 寄存器的 C0FCK1 位和 C0FCK0 位为“01B”、“10B”或者“11B”）时的运行例子。

11.4.2 比较器 0 的中断运行

使用比较器 0 中断时，必须将 COMPOCR 寄存器的 C0IE 位置“1”（允许中断请求）。通过 COMPFIR 寄存器设定产生中断请求的条件。另外，在比较器输出时可附加数字滤波器。

寄存器设定的详细内容请参照“11.3.3 比较器的滤波器控制寄存器（COMPFIR）”和“11.3.4 比较器输出控制寄存器（COMPOCR）”。

11.4.3 比较器 0 输出

能从 VCOUT0 引脚输出比较器的比较结果。通过 COMPOCR 寄存器的 C0OP 位设定 VCOUT0 引脚的输出极性（反相 / 不反相），通过 C0OE 位设定禁止或者允许 VCOUT0 引脚输出。寄存器设定的详细内容请参照“11.3.4 比较器输出控制寄存器（COMPOCR）”。

从 VCOUT0 引脚输出比较器的比较结果时，必须按照“图 11-8 比较器的运行允许步骤”进行设定。

11.5 比较器的设定流程

比较器的设定流程如下所示。

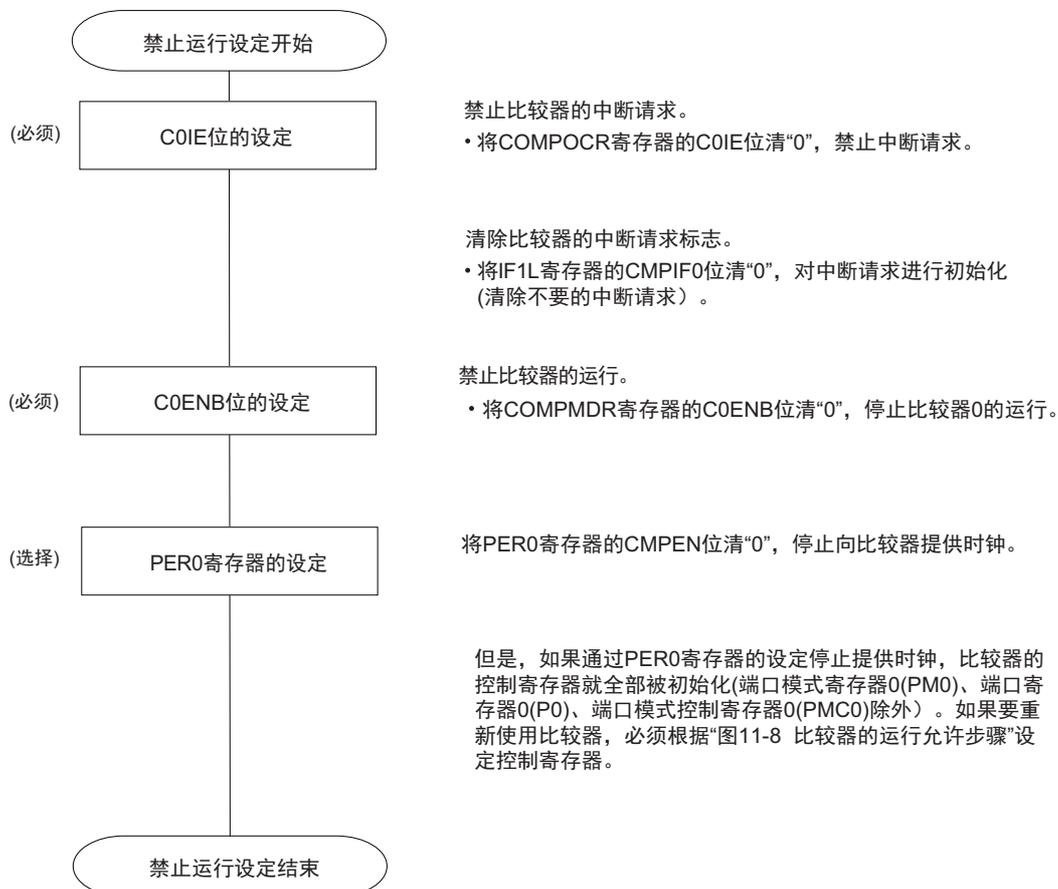
11.5.1 允许比较器的运行

图 11-8 比较器的运行允许步骤



11.5.2 禁止比较器的运行

图 11-9 比较器的运行禁止步骤



第 12 章 串行阵列单元

串行阵列单元 0 最多有 2 个串行通道。各通道能实现 3 线串行（CSI）、UART 和简易 I²C 的通信功能。RL78/G10 支持的各通道的功能分配如下：

○ 10 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	—		—

○ 16 引脚产品

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		—

不能通过多个通信方式同时使用 1 个通道。设定不同通信方式时，必须使用其他通道。

12.1 串行阵列单元的功能

RL78/G10 支持各串行接口的特征如下所示。

12.1.1 3 线串行 I/O (CSI00、CSI01)

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“12.5 3 线串行 I/O (CSI00、CSI01) 通信的运行”。

[数据的发送和接收]

- 7 位或者 8 位的数据长度
- 发送和接收数据的相位控制
- MSB/LSB 优先的选择

[时钟控制]

- 主控或者从属的选择
- 输入/输出时钟的相位控制
- 设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: $\text{Max. } f_{\text{CLK}}/4$

从属通信: $\text{Max. } f_{\text{MCK}}/6$

[中断功能]

- 传送结束中断/缓冲器空中断

[错误检测标志]

- 溢出错误

注 必须在满足 SCK 周期时间 (t_{KCY}) 特性的范围内使用。详细内容请参照“第 24 章 电特性”。

12.1.2 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。

有关具体的设定例子, 请参照“12.6 UART (UART0) 通信的运行”。

[数据的发送和接收]

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

[中断功能]

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

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

能通过设定 ISC 寄存器, 将 UART0 的 RxD0 引脚的输入信号设定为外部中断输入或者定时器阵列单元的定时器输入。另外, 能通过使用定时器阵列单元的输入脉冲间隔测量模式测量通信方的波特率宽度, 并且能进行 UART0 的波特率校正。

12.1.3 简易 I²C (IIC00)

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

开始条件和停止条件与操作控制寄存器一样, 必须遵守 AC 特性, 通过软件进行处理。

有关具体的设定例子, 请参照“12.7 简易 I²C (IIC00) 通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收 (只限于单主控的主控功能)
- ACK 输出功能^{注1}、ACK 检测功能
- 8 位数据长度 (在发送地址时, 用高 7 位指定地址, 用最低位进行 R/W 控制)
- 手动产生开始条件和停止条件

[中断功能]

- 传送结束中断

[错误检测标志]

- 溢出错误
- ACK 错误

※ [简易 I²C 不支持的功能^{注2}]

- 从属发送、从属接收
- 多主功能 (仲裁失败检测功能)
- 等待检测功能

注 1. 在接收最后的数据时, 如果给 SOE00 位 (串行输出允许寄存器 0 (SOE0)) 写“0”来停止串行通信数据的输出, 就不输出 ACK。详细内容请参照“12.7.3(2) 处理流程”。

2. 简易 I²C 请参照“第 13 章 串行接口 IICA” (只限于 16 引脚产品)。

12.2 串行阵列单元的结构

串行阵列单元由以下硬件构成。

表 12-1 串行阵列单元的结构

项目	结构
移位寄存器	8 位
缓冲寄存器	串行数据寄存器 0nL (SDR0nL 注2)
串行时钟输入 / 输出	SCK00、SCK01 注1 引脚 (用于 3 线串行 I/O)、SCL00 引脚 (用于简易 I ² C)
串行数据输入	SI00、SI01 注1 引脚 (用于 3 线串行 I/O)、RxD0 引脚 (用于 UART)
串行数据输出	SO00、SO01 注1 引脚 (用于 3 线串行 I/O)、TxD0 引脚 (用于 UART)
串行数据输入 / 输出	SDA00 引脚 (用于简易 I ² C)
控制寄存器	<单元设定部的寄存器> <ul style="list-style-type: none"> • 外围允许寄存器 0 (PER0) • 串行时钟选择寄存器 0 (SPS0) • 串行通道允许状态寄存器 0 (SE0) • 串行通道开始寄存器 0 (SS0) • 串行通道停止寄存器 0 (ST0) • 串行输出允许寄存器 0 (SOE0) • 串行输出寄存器 0 (SO0) • 串行时钟输出寄存器 0 (CKO0) • 串行输出电平寄存器 0 (SOL0) • 噪声滤波器允许寄存器 0 (NFEN0) • 输入切换控制寄存器 (ISC)
	<各通道部的寄存器> <ul style="list-style-type: none"> • 串行数据寄存器 0n (SDR0nH、SDR0nL 注2) • 串行模式寄存器 0n (SMR0nH、SMR0nL) • 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL) • 串行状态寄存器 0n (SSR0n) • 串行标志清除触发寄存器 0n (SIR0n)
	<端口功能部的寄存器> <ul style="list-style-type: none"> • 端口输出模式寄存器 0 (POM0) • 端口模式控制寄存器 0 (PMC0) • 端口模式寄存器 0 (PM0) • 端口寄存器 0 (P0)

注 1. CSI01 通信只限于 16 引脚产品。

2. 根据通信模式，能用以下 SFR 名称读写串行数据寄存器 0nL (SDR0nL)。

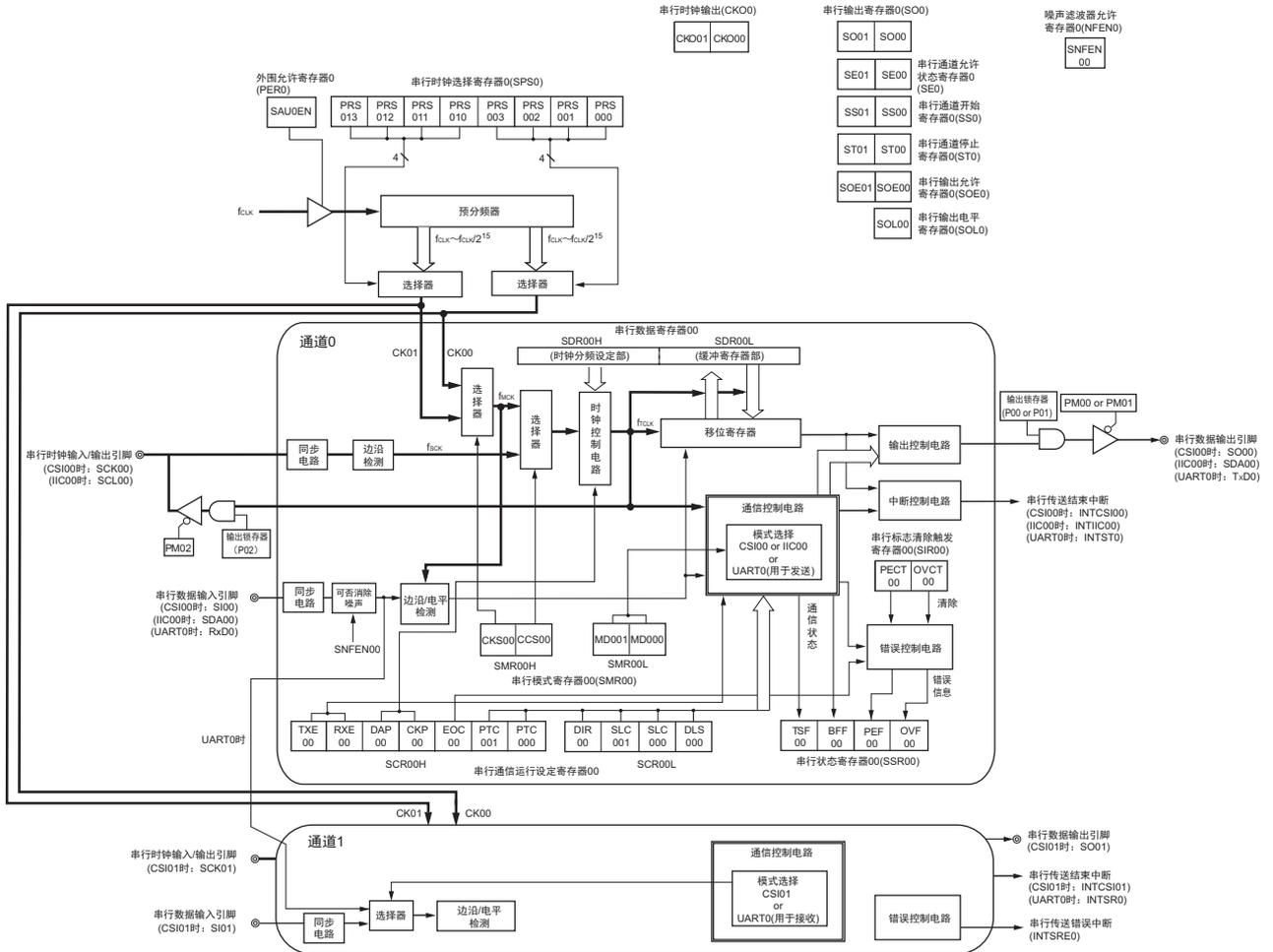
- CSIp 通信SIOp (CSIp 数据寄存器)
- UART0 接收RxD0 (UART0 接收数据寄存器)
- UART0 发送TxD0 (UART0 发送数据寄存器)
- IIC0 通信SIO0 (IIC0 数据寄存器)

备注 n: 通道号 (n=0、1)

p: CSI 号 (p=00、01)

串行阵列单元 0 的框图如图 12-1 所示。

图 12-1 串行阵列单元 0 的框图



注 只有 16 引脚产品有 CSI01 通信。

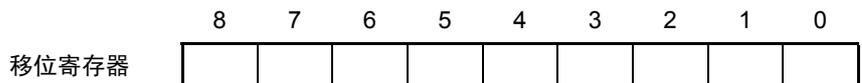
12.2.1 移位寄存器

这是进行并行和串行相互转换的 8 位寄存器。

在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出。

不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，使用串行数据寄存器 0nL (SDR0nL)。



12.2.2 串行数据寄存器 0nL (SDR0nL)

SDR0nL 寄存器用作通道 n 的发送和接收缓冲寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到 SDR0nL 寄存器。

在发送数据时，将传送到移位寄存器的发送数据设定到 SDR0nL 寄存器。

与数据的输出顺序无关，根据串行通信运行设定寄存器 0n (SCR0nL) 的 bit0 (DLS0n0) 的设定，保存到 SDR0nL 寄存器的数据如下所示：

- 7 位数据长度 (保存在 SDR0nL 寄存器的 bit0~6)
- 8 位数据长度 (保存在 SDR0nL 寄存器的 bit0~7)

在允许运行 (SE0n=1) 时，通过 8 位存储器操作指令设定 SDR0nL 寄存器。在停止运行 (SE0n=0) 时，禁止写 SDR0nL 寄存器。

在产生复位信号后，SDR0nL 寄存器的值变为“00H”。

根据通信模式，能用以下 SFR 名称，通过 8 位存储器操作指令设定 SDR0nL 寄存器。

- CSIp 通信SIOp (CSIp 数据寄存器)
- UART0 接收RXD0 (UART0 接收数据寄存器)
- UART0 发送TXD0 (UART0 发送数据寄存器)
- IIC00 通信SIO00 (IIC00 数据寄存器)

备注 n: 通道号 (n=0、1)

p: CSI 号 (p=00、01)

图 12-2 串行数据寄存器 0nL (SDR0nL) 的格式 (n=0、1)



备注 有关 SDR0nH 寄存器的功能，请参照“12.3 串行阵列单元的控制寄存器”。

12.3 串行阵列单元的控制寄存器

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0 (SPS0)
- 串行模式寄存器0n (SMR0nH、SMR0nL)
- 串行通信运行设定寄存器0n (SCR0nH、SCR0nL)
- 串行数据寄存器0n (SDR0nH、SDR0nL)
- 串行标志清除触发寄存器0n (SIR0n)
- 串行状态寄存器0n (SSR0n)
- 串行通道开始寄存器0 (SS0)
- 串行通道停止寄存器0 (ST0)
- 串行通道允许状态寄存器0 (SE0)
- 串行输出允许寄存器0 (SOE0)
- 串行输出电平寄存器0 (SOL0)
- 串行输出寄存器0 (SO0)
- 串行时钟输出寄存器0 (CKO0)
- 噪声滤波器允许寄存器0 (NFEN0)
- 输入切换控制寄存器 (ISC)
- 端口输出模式寄存器0 (POM0)
- 端口模式控制寄存器0 (PMC0)
- 端口模式寄存器0 (PM0)
- 端口寄存器0 (P0)

备注 n: 通道号 (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，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 12-3 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	TMKAEN ^注	CMPEN ^注	ADCEN	IICA0EN ^注	0	SAU0EN	0	TAU0EN

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

注 只限于 16 引脚产品。

注意 1. 在设定串行阵列单元 0 时，必须在 SAU0EN 位为“1”的状态下设定以下寄存器。如果 SAU0EN 位为“0”，串行阵列单元 0 的控制寄存器为初始值，并且忽视写操作（噪声滤波器允许寄存器 0 (NFEN0)、数据切换控制寄存器 (ISC)、端口输出模式寄存器 0 (POM0)、端口模式寄存器 0 (PM0)、端口模式控制寄存器 0 (PMC0) 端口寄存器 0 (P0) 除外）。

- 串行时钟选择寄存器 0 (SPS0)
- 串行模式寄存器 0n (SMR0nH、SMR0nL)
- 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)
- 串行数据寄存器 0n (SDR0nH、SDR0nL)
- 串行标志清除触发寄存器 0n (SIR0n)
- 串行状态寄存器 0n (SSR0n)
- 串行通道开始寄存器 0 (SS0)
- 串行通道停止寄存器 0 (ST0)
- 串行通道允许状态寄存器 0 (SE0)
- 串行输出允许寄存器 0 (SOE0)
- 串行输出电平寄存器 0 (SOL0)
- 串行输出寄存器 0 (SO0)
- 串行时钟输出寄存器 0 (CKO0)

2. 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

12.3.2 串行时钟选择寄存器 0 (SPS0)

SPS0 寄存器是 8 位寄存器，选择提供给各通道的 2 种公共运行时钟（CK00、CK01）。通过 SPS0 寄存器的 bit7 ~ 4 选择 CK01，通过 bit3 ~ 0 选择 CK00。

禁止在运行允许状态（SE0n=1）改写 SPS0 寄存器。

通过 8 位存储器操作指令设定 SPS0 寄存器。

在产生复位信号后，SPS0 寄存器的值变为“00H”。

图 12-4 串行时钟选择寄存器 0 (SPS0) 的格式

地址: F0126H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0		运行时钟 (CKn) 的选择 ^注				
					$f_{CLK}=1.25MHz$	$f_{CLK}=2.5MHz$	$f_{CLK}=5MHz$	$f_{CLK}=10MHz$	$f_{CLK}=20MHz$
0	0	0	0	f_{CLK}	1.25MHz	2.5MHz	5MHz	10MHz	20MHz
0	0	0	1	$f_{CLK}/2$	625kHz	1.25MHz	2.5MHz	5MHz	10MHz
0	0	1	0	$f_{CLK}/2^2$	313kHz	625kHz	1.25MHz	2.5MHz	5MHz
0	0	1	1	$f_{CLK}/2^3$	156kHz	313kHz	625kHz	1.25MHz	2.5MHz
0	1	0	0	$f_{CLK}/2^4$	78kHz	156kHz	313kHz	625kHz	1.25MHz
0	1	0	1	$f_{CLK}/2^5$	39kHz	78kHz	156kHz	313kHz	625kHz
0	1	1	0	$f_{CLK}/2^6$	19.5kHz	39kHz	78kHz	156kHz	313kHz
0	1	1	1	$f_{CLK}/2^7$	9.8kHz	19.5kHz	39kHz	78kHz	156kHz
1	0	0	0	$f_{CLK}/2^8$	4.9kHz	9.8kHz	19.5kHz	39kHz	78kHz
1	0	0	1	$f_{CLK}/2^9$	2.5kHz	4.9kHz	9.8kHz	19.5kHz	39kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22kHz	2.5kHz	4.9kHz	9.8kHz	19.5kHz
1	0	1	1	$f_{CLK}/2^{11}$	625Hz	1.22kHz	2.5kHz	4.9kHz	9.8kHz
1	1	0	0	$f_{CLK}/2^{12}$	313Hz	625Hz	1.22kHz	2.5kHz	4.9kHz
1	1	0	1	$f_{CLK}/2^{13}$	152Hz	313Hz	625Hz	1.22kHz	2.5kHz
1	1	1	0	$f_{CLK}/2^{14}$	78Hz	152Hz	313Hz	625Hz	1.22kHz
1	1	1	1	$f_{CLK}/2^{15}$	39Hz	78Hz	152Hz	313Hz	625Hz

注 要更改 f_{CLK} 选择的时钟（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 0（ST0）=03H）后进行更改。

备注 1. f_{CLK} : CPU/ 外围硬件的时钟频率

2. n: 通道号 (n=0、1)

12.3.3 串行模式寄存器 0n (SMR0nH、SMR0nL)

SMR0nH、SMR0nL 寄存器是设定通道 n 的运行模式的寄存器，进行运行时钟 (f_{MCK}) 的选择、能否使用串行时钟 (f_{SCK}) 输入的指定、开始触发的设定、运行模式 (CSI、UART、简易 I²C) 的设定以及中断源的选择。另外，只在 UART 模式中设定接收数据的反相电平。

禁止在运行允许状态 ($SE0n=1$) 下改写 SMR0nH、SMR0nL 寄存器。但是，能在运行运行允许状态 ($SE0n=1$) 下改写 MD0n0 位。

通过 8 位存储器操作指令设定 SMR0nH、SMR0nL 寄存器。

在产生复位信号后，SMR0nH、SMR0nL 寄存器的值分别变为“00H”和“20H”。

图 12-5 串行模式寄存器 0n (SMR0nH、SMR0nL) 的格式 (1/2)

地址: F0111H (SMR00H)、F0113H (SMR01H) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SMR0nH	CKS0n	CCS0n	0	0	0	0	0	STS0n 注 1

地址: F0110H (SMR00L)、F0112H (SMR01L) 复位后: 20H R/W

符号	7	6	5	4	3	2	1	0
SMR0nL	0	SIS0n0 注 2	1	0	0	MD0n2	MD0n1	MD0n0

CKS0n	通道 n 的运行时钟 (f_{MCK}) 选择
0	SPS0 寄存器设定的运行时钟 CK00
1	SPS0 寄存器设定的运行时钟 CK01
运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定 CCS0n 位和 SDR0nH 寄存器，生成传送时钟 (f_{TCLK})。	

CCS0n	通道 n 的传送时钟 (f_{TCLK}) 选择
0	CKS0n 位指定的运行时钟 f_{MCK} 的分频时钟
1	来自 SCKp 引脚的输入时钟 f_{SCK} (CSI 模式的从属传送)
传送时钟 f_{TCLK} 用于移位寄存器、通信控制电路、输出控制器、中断控制电路和错误控制电路。当 CCS0n 位为“0”时，通过 SDR0nH 寄存器的高 7 位进行运行时钟 (f_{MCK}) 的分频设定。	

STS0n 注 1	开始触发源的选择
0	只有软件触发有效 (在 CSI、UART 发送、简易 I ² C 时选择)。
1	RxD0 引脚的有效边沿 (在 UART 接收时选择)
在将 SS0 寄存器置“1”后满足上述条件时，开始传送。	

- 注 1. 只限于 SMR01H 寄存器。
2. 只限于 SMR01L 寄存器。

注意 必须将以下位固定为初始值。

SMR00H: 必须将 bit0 ~ 5 置“0”。

SMR01H: 必须将 bit1 ~ 5 置“0”。

SMR00L: 必须将 bit3、4、6、7 置“0”、将 bit5 置“1”。

SMR01L: 必须将 bit3、4、7 置“0”、将 bit5 置“1”。

备注 n: 通道号 (n=0、1)

图 12-5 串行模式寄存器 0n (SMR0nH、SMR0nL) 的格式 (2/2)

地址: F0111H (SMR00H)、F0113H (SMR01H) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SMR0nH	CKS0n	CCS0n	0	0	0	0	0	STS0n 注 1

地址: F0110H (SMR00L)、F0112H (SMR01L) 复位后: 20H R/W

符号	7	6	5	4	3	2	1	0
SMR0nL	0	SIS0n0 注 2	1	0	0	MD0n2	MD0n1	MD0n0

SIS0n0 注 2	UART0 的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。

MD0n2	MD0n1	通道 n 的运行模式设定
0	0	CSI 模式
0	1	UART 模式
1	0	简易 I ² C 模式
1	1	禁止设定。

MD0n0	通道 n 的中断源选择
0	传送结束中断
1	缓冲器空中断 (在数据从 SDR0nL 寄存器传送到移位寄存器时发生。)
在连续发送时, 如果 MD0n0 位为“1”并且 SDR0nL 的数据为空, 就写下一个发送数据。	

- 注 1. 只限于 SMR01H 寄存器。
2. 只限于 SMR01L 寄存器。

注意 必须将以下位固定为初始值。

SMR00H: 必须将 bit0 ~ 5 置“0”。

SMR01H: 必须将 bit1 ~ 5 置“0”。

SMR00L: 必须将 bit3、4、6、7 置“0”、将 bit5 置“1”。

SMR01L: 必须将 bit3、4、7 置“0”、将 bit5 置“1”。

备注 n: 通道号 (n=0、1)

12.3.4 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)

这是通道 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。

禁止在运行允许状态 (SE0n=1) 下改写 SCR0nH、SCR0nL 寄存器。

通过 8 位存储器操作指令设定 SCR0nH、SCR0nL 寄存器。

在产生复位信号后，SCR0nH、SCR0nL 寄存器的值分别变为“00H”和“87H”。

图 12-6 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL) 的格式 (1/2)

地址: F0119H (SCR00H)、F011BH (SCR01H) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SCR0nH	TXE0n	RXE0n	DAP0n	CKP0n	0	EOC0n	PTC0n1	PTC0n0

地址: F0118H (SCR00L)、F011AH (SCR01L) 复位后: 87H R/W

符号	7	6	5	4	3	2	1	0
SCR0nL	DIR0n	0	SLC0n1 注 1	SLC0n0	0	1	1	DLS0n0

TXE0n	RXE0n	通道 n 的运行模式设定
0	0	禁止通信
0	1	只接收
1	0	只发送
1	1	发送和接收

DAP0n	CKP0n	CSI 模式中的数据 and 时钟的相位选择	类型
0	0		1
0	1		2
1	0		3
1	1		4

在 UART 模式和简易 I²C 模式中，必须将 DAP0n 位和 CKP0n 位都置“0”。

EOC0n	错误中断信号 (INTSREx (x=0、1)) 是否屏蔽的选择
0	禁止发生错误中断 INTSREx (发生 INTSRx)。
1	允许发生错误中断 INTSREx (在发生错误时不发生 INTSRx)。

在 CSI 模式和简易 I²C 模式中或者在 UART 发送时，必须将 EOC0n 位置“0”注 2。

注 1. 只限 SCR00L 寄存器。

注 2. 在 EOC0n 位为“0”并且不使用 CSI0n 时，有可能产生错误中断 INTSREn。

图 12-6 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL) 的格式 (2/2)

地址: F0119H (SCR00H)、F011BH (SCR01H) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SCR0nH	TXE0n	RXE0n	DAP0n	CKP0n	0	EOC0n	PTC0n1	PTC0n0

地址: F0118H (SCR00L)、F011AH (SCR01L) 复位后: 87H R/W

符号	7	6	5	4	3	2	1	0
SCR0nL	DIR0n	0	SLC0n1 注1	SLC0n0	0	1	1	DLS0n0

PTC0n1	PTC0n0	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 模式和简易 I²C 模式中, 必须将 PTC0n1 位和 PTC0n0 位都置“0”。

DIR0n	CSI 和 UART 模式中的数据传送顺序的选择
0	MSB 优先的输入 / 输出
1	LSB 优先的输入 / 输出

在简易 I²C 模式中, 必须将 DIR0n 位置“0”。

SLC0n1 注1	SLC0n0	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 n=0)
1	1	禁止设定。

如果选择了传送结束中断, 就在传送完所有停止位后产生中断。在 UART 接收时或者在简易 I²C 模式中, 必须设定为 1 个停止位 (SLC0n1、SLC0n0=0、1)。在 CSI 模式中, 必须设定为无停止位 (SLC0n1、SLC0n0=0、0)。在 UART 发送时, 设定为 1 个 (SLC0n1、SLC0n0=0、1) 或者 2 个 (SLC0n1、SLC0n0=1、0) 停止位。

DLS0n0	CSI 和 UART 模式中的数据长度的设定
0	7 位数据长度 (保存到 SDR0nL 寄存器的 bit0 ~ 6)
1	8 位数据长度 (保存到 SDR0nL 寄存器的 bit0 ~ 7)

在简易 I²C 模式中, 必须将 DLS0n0 位置“1”。

- 注 1. 只限于 SCR00L 寄存器。
2. 与数据的内容无关, 总是附加“0”。

注意 必须将以下位固定为初始值。
SCR0nH: 必须将 bit3 置“0”。
SCR00L: 必须将 bit3 和 bit6 置“0”、将 bit1 和 bit2 置“1”。
SCR01L: 必须将 bit3、5、6 置“0”、将 bit1 和 bit2 置“1”。

备注 n: 通道号 (n=0、1)

12.3.5 串行数据寄存器 0n (SDR0nH、SDR0nL)

SDR0nH、SDR0nL 寄存器是通道 n 的发送和接收数据寄存器。

通过 8 位存储器操作指令设定 SDR0nH、SDR0nL 寄存器。

在产生复位信号后，SDR0nH、SDR0nL 寄存器的值都变为“00H”。

SDR0nH 寄存器用作运行时钟 (f_{MCK}) 的分频设定寄存器。

如果将 SMR0nH 寄存器的 CCS0n 位置“0”，由 SDR0nH 寄存器设定的运行时钟的分频时钟就用作传送时钟。

如果将 CCS0n 位置“1”，就必须将 SDR0nH 寄存器置“0000000B”。SCKp 引脚的输入时钟 f_{SCK} (CSI 模式的从属传送) 为传送时钟。

在运行停止状态 (SE0n=0) 下，通过 8 位存储器操作指令设定 SDR0nH 寄存器。在运行允许状态 (SE0n=1) 下，SDR0nH 寄存器的写操作无效，并且 SDR0nH 寄存器的读取值总是为“0”。

SDR0nL 寄存器用作发送或者接收缓冲寄存器。在接收时，将通过移位寄存器转换的并行数据保存到 SDR0nL 寄存器。在发送时，将要传送到移位寄存器的发送数据设定到 SDR0nL 寄存器。

在运行允许状态 (SE0n=1) 下，通过 8 位存储器操作指令设定 SDR0nL 寄存器。在运行停止状态 (SE0n=0) 下，禁止写 SDR0nL 寄存器。

图 12-7 串行数据寄存器 0n (SDR0n) 的格式



注意 1. 在使用 UART 时，禁止设定为 SDR0nH[7:1]= (0000000B、0000001B)。

2. 在使用简易 I²C 时，禁止设定为 SDR0nH[7:1]=0000000B。

备注 1. 有关 SDR0nL 寄存器的功能，请参照“12.2 串行阵列单元的结构”。

2. n: 通道号 (n=0、1)

12.3.6 串行标志清除触发寄存器 0n (SIR0n)

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 (FECT0n、PECT0n、OVCT0n) 置“1”，就将串行状态寄存器 0n (SSR0n) 的对应位 (FEF0n、PEF0n、OVF0n) 清“0”。因为 SIR0n 寄存器是触发寄存器，所以如果清除 SSR0n 寄存器的对应位，也会立即清除 SIR0n 寄存器。

通过 8 位存储器操作指令设定 SIR0n 寄存器。

在产生复位信号后，SIR0n 寄存器的值变为“00H”。

图 12-8 串行标志清除触发寄存器 0n (SIR0n) 的格式

地址: F0108H (SIR00)、F010AH (SIR01)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	FECT0n注	PECT0n	OVCT0n

FECT0n 注	通道 n 帧错误标志的清除触发
0	不清除。
1	将 SSR0n 寄存器的 FEF0n 位清“0”。

PECT0n	通道 n 奇偶校验错误标志的清除触发
0	不清除。
1	将 SSR0n 寄存器的 PEF0n 位清“0”。

OVCT0n	通道 n 溢出错误标志的清除触发
0	不清除。
1	将 SSR0n 寄存器的 OVF0n 位清“0”。

注 只限于 SIR01 寄存器。

注意 必须将以下位置“0”。

SIR00 寄存器: bit2 ~ 7

SIR01 寄存器: bit3 ~ 7

备注 1. n: 通道号 (n=0、1)

2. SIR0n 寄存器的读取值总是为“00H”。

12.3.7 串行状态寄存器 0n (SSR0n)

SSR0n 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 8 位存储器操作指令读取 SSR0n 寄存器。

在产生复位信号后，SSR0n 寄存器的值变为“00H”。

图 12-9 串行状态寄存器 0n (SSR0n) 的格式 (1/2)

地址: F0100H (SSR00)、F0102H (SSR01)	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n 注	PEF0n	OVF0n

TSF0n	通道 n 的通信状态表示标志
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]	
<ul style="list-style-type: none"> 当将 ST0 寄存器的 ST0n 位置“1”（通信停止状态）或者将 SS0 寄存器的 SS0n 位置“1”（通信待机状态）时 当通信结束时 	
[置位条件]	
<ul style="list-style-type: none"> 当开始通信时 	

BFF0n	通道 n 的缓冲寄存器状态表示标志
0	SDR0nL 寄存器没有保存有效数据。
1	SDR0nL 寄存器保存了有效数据。
[清除条件]	
<ul style="list-style-type: none"> 在发送过程中将发送数据从 SDR0nL 寄存器传送到移位寄存器时 在接收过程中从 SDR0nL 寄存器读到接收数据时 将 ST0 寄存器的 ST0n 位置“1”（通信停止状态）或者将 SS0 寄存器的 SS0n 位置“1”（通信允许状态）。 	
[置位条件]	
<ul style="list-style-type: none"> 在 SCR0nH 寄存器的 TXE0n 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 SDR0nL 寄存器写发送数据时 在 SCR0nH 寄存器的 RXE0n 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 SDR0nL 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 BFF0n 位为“0”时写 SDR0nL 寄存器，已保存的发送 / 接收数据就被破坏，并且检测为溢出错误 (OVE0n=1)。

备注 n: 通道号 (n=0、1)

图 12-9 串行状态寄存器 0n (SSR0n) 的格式 (2/2)

地址: F0100H (SSR00)、F0102H (SSR01) 复位后: 00H R

符号	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n注	PEF0n	OVF0n

FEF0n注	通道 n 的帧错误检测标志
0	没有发生错误。
1	发生错误 (UART 接收时)。
[清除条件]	
• 当给 SIR0n 寄存器的 FECT0n 位写“1”时	
[置位条件]	
• 在 UART 接收结束时没有检测到停止位时	

PEF0n	通道 n 的奇偶校验 /ACK 错误检测标志
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 (简易 I ² C 发送时)。
[清除条件]	
• 当给 SIR0n 寄存器的 PECT0n 位写“1”时	
[置位条件]	
• 在 UART 接收结束时发送数据的奇偶校验和奇偶校验位不同 (奇偶校验错误) 时	
• 在简易 I ² C 发送时并且在 ACK 接收时序从属方没有返回 ACK 信号 (未检测到 ACK) 时	

OVF0n	通道 n 的溢出错误检测标志
0	没有发生错误。
1	发生错误。
[清除条件]	
• 当给 SIR0n 寄存器的 OVCT0n 位写“1”时	
[置位条件]	
• 在 SCR0nH 寄存器的 RXE0n 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 SDR0nL 寄存器中, 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。

备注 n: 通道号 (n=0、1)

12.3.8 串行通道开始寄存器 0 (SS0)

SS0 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 (SS0n) 写“1”，就将串行通道允许状态寄存器 0 (SE0) 的对应位 (SE0n) 置“1” (运行允许状态)。因为 SS0n 位是触发位，所以如果 SE0n 位为“1”就立即清除 SS0n 位。

通过 1 位或者 8 位存储器操作指令设定 SS0 寄存器。

在产生复位信号后，SS0 寄存器的值变为“00H”。

图 12-10 串行通道开始寄存器 0 (SS0) 的格式

地址: F0122H (SS0)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01	SS00

SS0n	通道 n 的运行开始触发
0	没有触发。
1	将 SE0n 位置“1”，转移到通信待机状态注。

注 如果在通信中将 SS0n 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、SCK0n 引脚、SO0n 引脚和 FEF0n 标志、PEF0n 标志、OVF0n 标志保持状态。

注意 1. 必须将 bit2 ~ 7 置“0”。

2. 在 UART 接收时，必须在将 SCR0nH 寄存器的 RXE0n 位置“1”后至少间隔 4 个 f_{MCK} 时钟，然后将 SS0n 置“1”。

备注 1. n: 通道号 (n=0、1)

2. SS0 寄存器的读取值总是为“00H”。

12.3.9 串行通道停止寄存器 0 (ST0)

ST0 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 (ST0n) 写“1”，就将串行通道允许状态寄存器 0 (SE0) 的对应位 (SE0n) 清“0” (运行停止状态)。因为 ST0n 位是触发位，所以如果 SE0n 位为“0”就立即清除 ST0n 位。

通过 1 位或者 8 位存储器操作指令设定 ST0 寄存器。

在产生复位信号后，ST0 寄存器的值变为“00H”。

图 12-11 串行通道停止寄存器 0 (ST0) 的格式

地址: F0124H (ST0)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	ST01	ST00

ST0n	通道 n 的运行停止触发
0	没有触发。
1	将 SE0n 位清“0”，停止通信运行注。

注 控制寄存器和移位寄存器的值、SCK0n 引脚和 SO0n 引脚以及 FEF0n 标志、PEF0n 标志和 OVF0n 标志保持状态。

注意 必须将 bit2 ~ 7 置“0”。

备注 1. n: 通道号 (n=0、1)

2. ST0 寄存器的读取值总是为“00H”。

12.3.10 串行通道允许状态寄存器 0 (SE0)

SE0 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 0 (SS0) 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 0 (ST0) 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 0 (SO0) 的 CKO0n 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 SO0 寄存器的 CKO0n 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 1 位或者 8 位存储器操作指令读取 SE0 寄存器。

在产生复位信号后，SE0 寄存器的值变为“00H”。

图 12-12 串行通道允许状态寄存器 0 (SE0) 的格式

地址: F0120H (SE0) 复位后: 00H R

符号	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	SE01	SE00

SE0n	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。

注意 必须将 bit2 ~ 7 置“0”。

备注 n: 通道号 (n=0、1)

12.3.11 串行输出允许寄存器 0 (SOE0)

SOE0 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 0 (SO0) 的 SO0n 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 SO0 寄存器的 SO0n 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 1 位或者 8 位存储器操作指令设定 SOE0 寄存器。

在产生复位信号后，SOE0 寄存器的值变为“00H”。

图 12-13 串行输出允许寄存器 0 (SOE0) 的格式

地址: F012AH (SOE0)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01注	SOE00

SOE0n	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 只限于 16 引脚产品。

注意 必须将以下位置“0”。

10 引脚产品: bit1 ~ 7

16 引脚产品: bit2 ~ 7

备注 n: 通道号 (n=0、1)

12.3.12 串行输出寄存器 0 (SO0)

SO0 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 SO0n 位的值。

只有在禁止串行输出 (SOE0n=0) 时才能通过软件改写此寄存器的 SO0n 位。当允许串行输出 (SOE0n=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 SO0n 位的值。

要将串行接口功能的复用引脚用作端口功能时, 必须将对应的 SO0n 位置“1”。

通过 8 位存储器操作指令设定 SO0 寄存器。

在产生复位信号后, SO0 寄存器的值变为“03H”。

图 12-14 串行输出寄存器 0 (SO0) 的格式

地址: F0128H (SO0)	复位后: 03H	R/W						
符号	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 注	SO00

SO0n	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 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit1 置“1”, bit2 ~ 7 置“0”; 将 16 引脚产品的 bit2 ~ 7 置“0”。

备注 n: 通道号 (n=0、1)

12.3.13 串行时钟输出寄存器 0 (CKO0)

CKO0 寄存器是各通道串行时钟输出的缓冲寄存器。

从通道 n 的串行时钟输出引脚输出此寄存器的 CKO0n 位的值。

只有在停止通道运行 (SE0n=0) 时才能通过软件改写此寄存器的 CKO0n 位。当允许通道运行 (SE0n=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 CKO0n 位的值。

要将串行接口功能的复用引脚用作端口功能时, 必须将对应的 CKO0n 位置“1”。

通过 8 位存储器操作指令设定 CKO0 寄存器。

在产生复位信号后, CKO0 寄存器的值变为“03H”。

图 12-15 串行输出寄存器 0 (CKO0) 的格式

地址: F0129H (CKO0)	复位后: 03H	R/W						
符号	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 注	CKO00

CKO0n	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”。

注 只限于 16 引脚产品。

注意 必须将 10 引脚产品的 bit1 置“1”, bit2 ~ 7 置“0”; 将 16 引脚产品的 bit2 ~ 7 置“0”。

备注 n: 通道号 (n=0、1)

12.3.14 串行输出电平寄存器 0 (SOL0)

SOL0 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 模式和简易 I²C 模式中，必须将对应的位置“0”。

只在允许串行输出 (SOE0n=1) 时，将此寄存器的通道 n 反相设定反映到引脚输出。在禁止串行输出 (SOE0n=0) 时，将 SO0n 位的值直接输出而不反相输出。

禁止在运行允许状态 (SE0n=1) 下改写 SOL0 寄存器。

通过 8 位存储器操作指令设定 SOL0 寄存器。

在产生复位信号后，SOL0 寄存器的值变为“00H”。

图 12-16 串行输出电平寄存器 0 (SOL0) 的格式

地址: F0134H (SOL0)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	SOL00

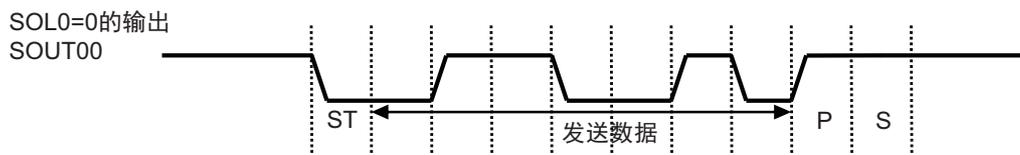
SOL00	UART 模式中的通道 0 发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

注意 必须将 bit1 ~ 7 置“0”。

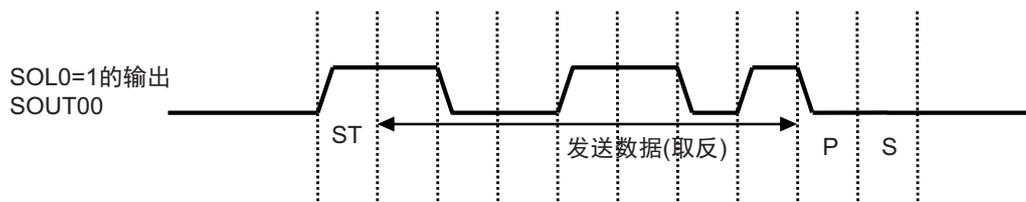
在进行 UART 发送时，发送数据的电平取反例子如图 12-17 所示。

图 12-17 发送数据的电平取反例子

(a)非取反输出(SOL00=0)



(b)取反输出(SOL00=1)



12.3.15 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可用于 UART 串行数据输入引脚的输入信号。

对于用于 CSI 或者简易 I²C 通信的引脚，必须将对应位置“0”，使噪声滤波器无效。

对于用于 UART 通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

在噪声滤波器有效时，通过对象通道的运行时钟 (f_{MCK}) 进行同步后，执行 2 个时钟的一致检测。在噪声滤波器无效时，只通过对象通道的运行时钟 (f_{MCK}) 进行同步。

通过 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 12-18 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 bit1 ~ 7 置“0”。

12.3.16 输入切换控制寄存器 (ISC)

在进行 UART0 的波特率校正时，ISC 寄存器的 ISC1 位和 ISC0 位用于与外部中断或者定时器阵列单元联合。

如果将 bit0 置“1”，串行数据输入 (RxD0) 引脚的输入信号就作为外部中断输入 (INTP0)。从而能通过 INTP0 中断检测到开始位的输入边沿信号。

如果将 bit1 置“1”，串行数据输入 (RxD0) 引脚的输入信号就作为定时器输入 (TI01)。通过使用定时器阵列单元的输入脉冲间隔测量模式，能将开始位的输入边沿信号作为触发，测量通信方的波特率 (传送速率) 宽度。

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 12-19 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	0	0	0	0	0	0	ISC1	ISC0

ISC1	定时器阵列单元的通道 1 的输入切换
0	将 TI01 引脚的输入信号作为定时器输入 (通常运行)
1	将 RxD0 引脚的输入信号作为定时器输入 (唤醒信号检测和校正波特率的脉宽测量)

ISC0	外部中断 (INTP0) 的输入切换
0	将 INTP0 引脚的输入信号作为外部中断输入 (通常运行)
1	将 RxD0 引脚的输入信号作为外部中断输入 (唤醒信号检测)

注意 必须将 bit2 ~ 7 置“0”。

12.3.17 串行输入 / 输出引脚的端口功能控制寄存器

在使用串行阵列单元时，必须设定与对象通道复用的端口相关寄存器（端口模式寄存器 0（PM0）、端口寄存器 0（P0）、0 端口输出模式寄存器 0（POM0）和端口模式控制寄存器 0（PMC0））。

详细内容请参照“4.3.1 端口模式寄存器 0、4（PM0、PM4）”、“4.3.2 端口寄存器 0、4、12、13（P0、P4、P12、P13）”、“4.3.4 端口输出模式寄存器 0（POM0）”、“4.3.5 端口模式控制寄存器 0（PMC0）”。

用作串行输入 / 输出引脚时的设定例子请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

在将串行数据输出或者串行时钟输出引脚的复用端口（P00/SO00/TXD0/INTP1 等）用作串行数据输出或者串行时钟输出时，必须将与各端口对应的端口模式控制寄存器 0（PMC0）的位和端口模式寄存器 0（PM0）的位置“0”，将端口寄存器 0（P0）的位置“1”。

另外，在用作 N 沟道漏极开路输出（ V_{DD} 耐压）模式时，必须将与各端口对应的端口输出模式寄存器 0（POM0）的位置“1”。

例） P00/SO00/TXD0/INTP1 用作串行时钟输出的情况

将端口模式控制寄存器 0 的 PMC00 位置“0”。

将端口模式寄存器 0 的 PM00 位置“0”。

将端口寄存器 0 的 P00 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（P01/ANI0/SI00/RXD0/SDA00/KR2 等）用作串行数据输入或者串行时钟输入时，必须将与各端口对应的端口模式寄存器 0（PM0）的位置“1”，并且将端口模式控制寄存器 0（PMC0）的位置“0”。此时，端口寄存器 0（P0）的位可以是“0”或者“1”。

例） P01/ANI0/SI00/RXD0/SDA00/KR2 用作串行时钟输入的情况

将端口模式控制寄存器 0 的 PMC01 位置“0”。

将端口模式寄存器 0 的 PM01 位置“1”。

12.4 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。

12.4.1 以单元为单位停止运行的情况

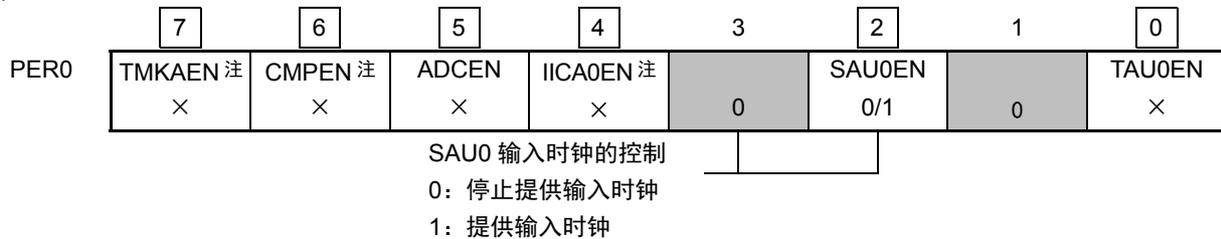
通过外围允许寄存器 0（PER0）设定以单元为单位的停止运行。

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，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 12-20 以单元为单位停止运行时的外围允许寄存器 0（PER0）的设定

(a) 外围允许寄存器 0（PER0）..... 将要停止 SAU0 的位置“0”。



注 只限于 16 引脚产品。

注意 1. 要设定串行阵列单元 0 时，必须在 SAU0EN 位为“1”的状态下设定串行阵列单元 0 的控制寄存器。在 SAU0EN 位为“0”时，串行阵列单元 0 的控制寄存器都为初始值，并且忽视写操作（噪声滤波器允许寄存器 0（NFEN0）、输入切换控制寄存器（ISC）、端口输出模式寄存器 0（POM0）、端口模式寄存器 0（PM0）、端口模式控制寄存器 0（PMC0）和端口寄存器 0（P0）除外）。

2. 必须将以下的位置“0”。

10 引脚产品：bit1、3、4、6、7

16 引脚产品：bit1 和 bit3

备注 ：不能设定（设定初始值）。

×：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）。

0/1：根据用户的用途置“0”或者“1”。

12.4.2 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。

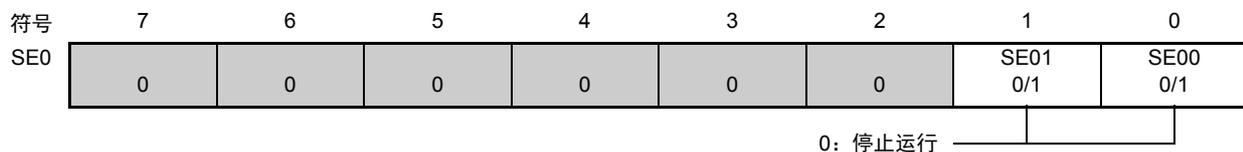
图 12-21 按通道停止运行时的各寄存器的设定

(a) 串行通道停止寄存器 0 (ST0) 这是设定允许各通道的通信 / 停止计数的寄存器。



※ 因为 ST0n 位是触发位，所以如果 SE0n 位为“0”就立即清除 ST0n 位。

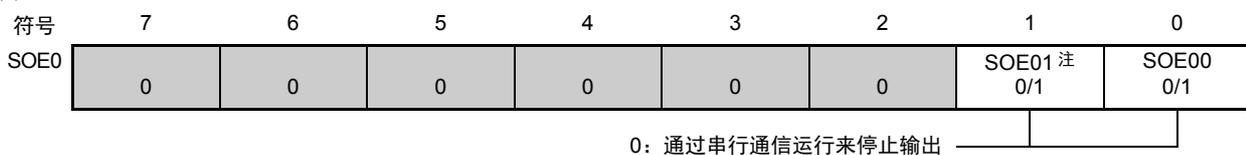
(b) 串行通道允许状态寄存器 0 (SE0) 此寄存器表示各通道的串行发送和接收的运行或者停止状态。



※ SE0 寄存器是只读状态寄存器，通过 ST0 寄存器来停止运行。

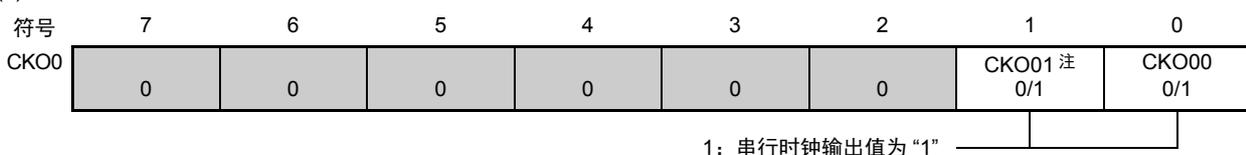
对于已经停止运行的通道，能通过软件设定 SO0 寄存器的 CKO0n 位的值。

(c) 串行输出允许寄存器 0 (SOE0) 这是设定允许或者停止各通道的串行通信输出的寄存器。



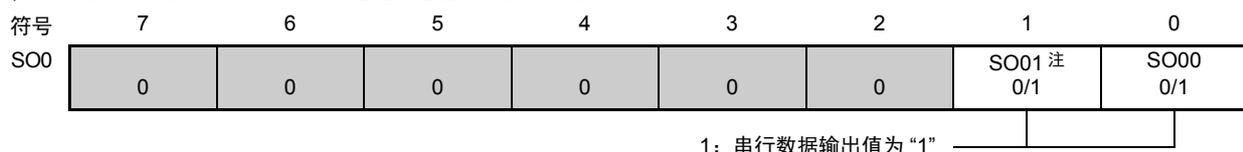
※ 对于已经停止串行输出的通道，能通过软件设定 SO0 寄存器的 SO0n 位的值。

(d) 串行时钟输出寄存器 0 (CKO0) 这是各通道的串行输出的缓冲寄存器。



※ 将各通道对应的引脚用作端口功能时，必须将对应的 CKO0n 位置“1”。

(e) 串行输出寄存器 0 (SO0) 这是各通道串行输出值的寄存器。



※ 将各通道对应的引脚用作端口功能时，必须将对应的 SO0n 位置“1”。

注 只限于 16 引脚产品。

备注 1. 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”。

12.5 3 线串行 I/O (CSI00、CSI01) 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

[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

主控通信: Max. $f_{CLK}/4$

从属通信: Max. $f_{MCK}/6$

[中断功能]

- 传送结束中断、缓冲器空中断

[错误检测标志]

- 溢出错误

注 必须在满足 SCK 周期时间 (t_{KCY}) 特性的范围内使用。详细内容请参照“第 24 章 电特性”。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01注		—

3 线串行 I/O (CSI00、CSI01注) 有以下 6 种通信运行:

- 主控发送 (参照 12.5.1)
- 主控接收 (参照 12.5.2)
- 主控发送和接收 (参照 12.5.3)
- 从属发送 (参照 12.5.4)
- 从属接收 (参照 12.5.5)
- 从属发送和接收 (参照 12.5.6)

注 只限于 16 引脚产品。

12.5.1 主控发送

主控发送是指此 RL78/G10 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	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SO00	SCK01、SO01
中断	INTCSI00	INTCSI01
	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 2	Max. $f_{CLK}/4$ [Hz] ($SDR0nH[7:1] \geq 1$) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
数据相位	能通过 SCR0nH 寄存器的 DAP0n 位进行选择。 <ul style="list-style-type: none"> • DAP0n=0: 在串行时钟开始运行时, 开始数据输出。 • DAP0n=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 <ul style="list-style-type: none"> • CKP0n=0: 不反相（在 SCK 的下降沿输出数据, 上升沿输入数据。） • CKP0n=1: 反相（在 SCK 的上升沿输出数据, 下降沿输入数据。） 	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

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 24 章 电特性”）的范围内使用。

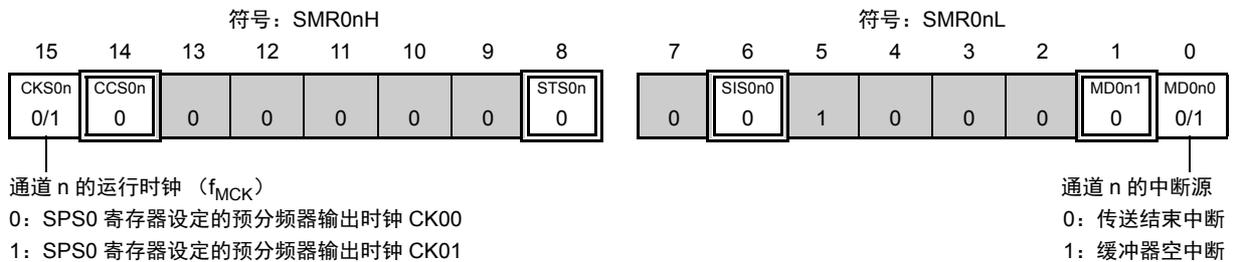
备注 1. f_{CLK} : 系统时钟频率

2. n=0、1

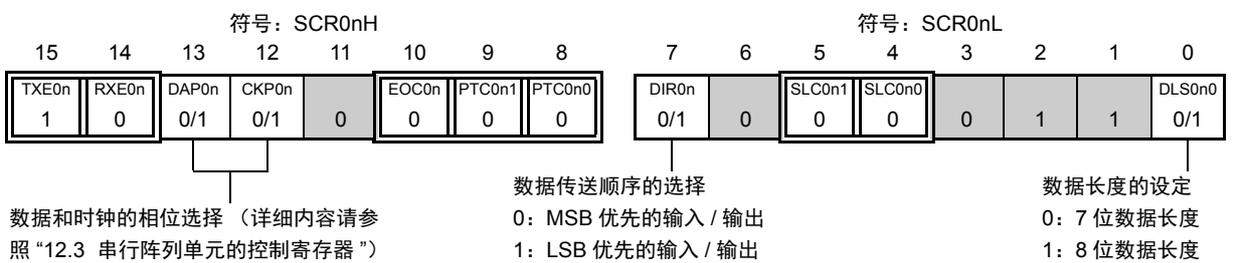
(1) 寄存器的设定

图 12-22 3 线串行 I/O (CSI00、CSI01 注) 主控发送时的寄存器设定内容例子 (1/2)

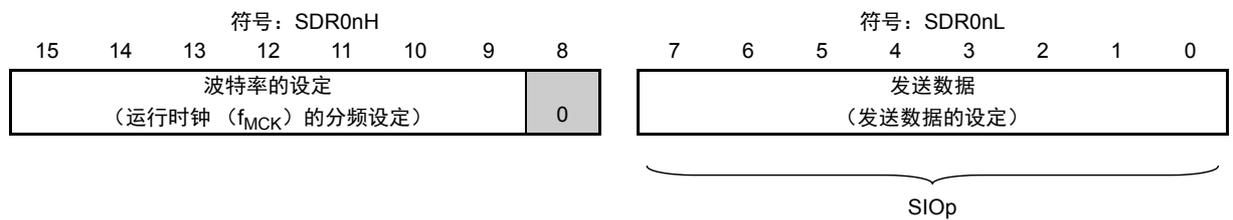
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



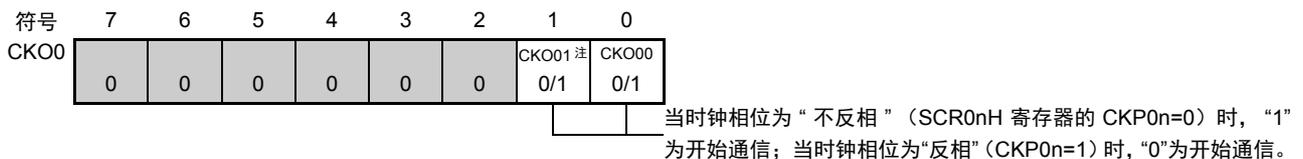
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



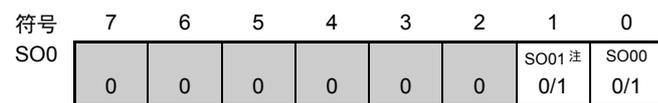
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 只设定对象通道的位。



(e) 串行输出寄存器 0 (SO0) 只设定对象通道的位。



注 只限于 16 引脚产品。

图 12-22 3 线串行 I/O (CSI00、CSI01 注) 主控发送时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 0/1	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- | |
|--|
| |
|--|

 : 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 12-23 主控发送的初始设定步骤

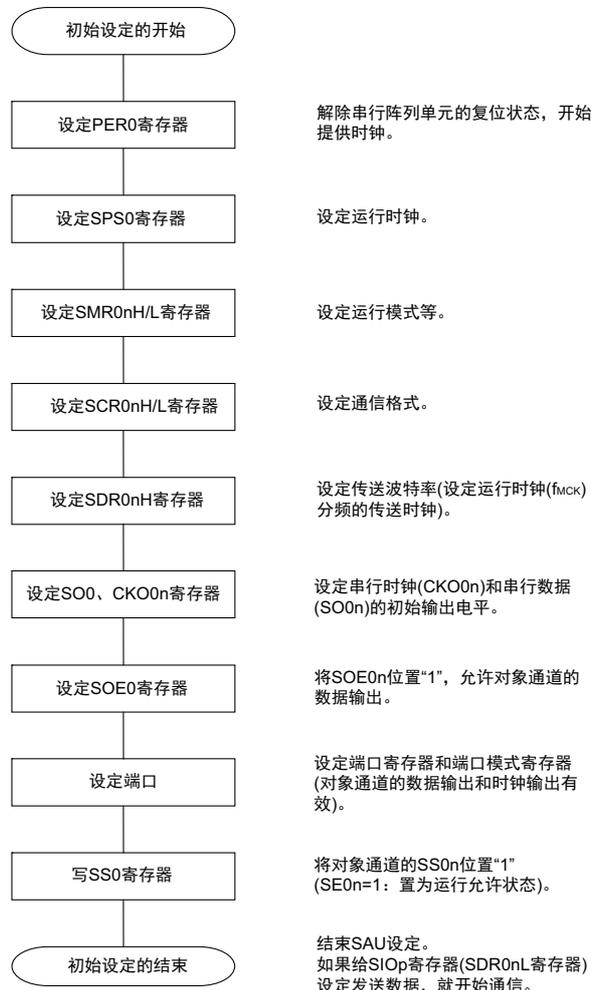


图 12-24 主控发送的中止步骤

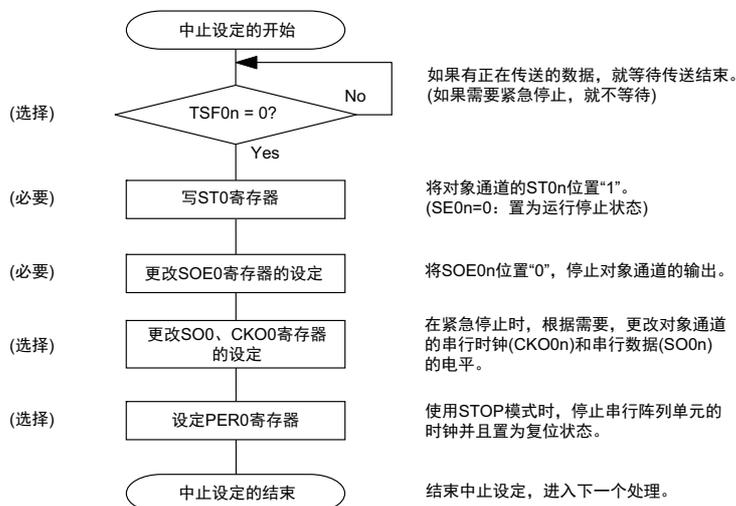


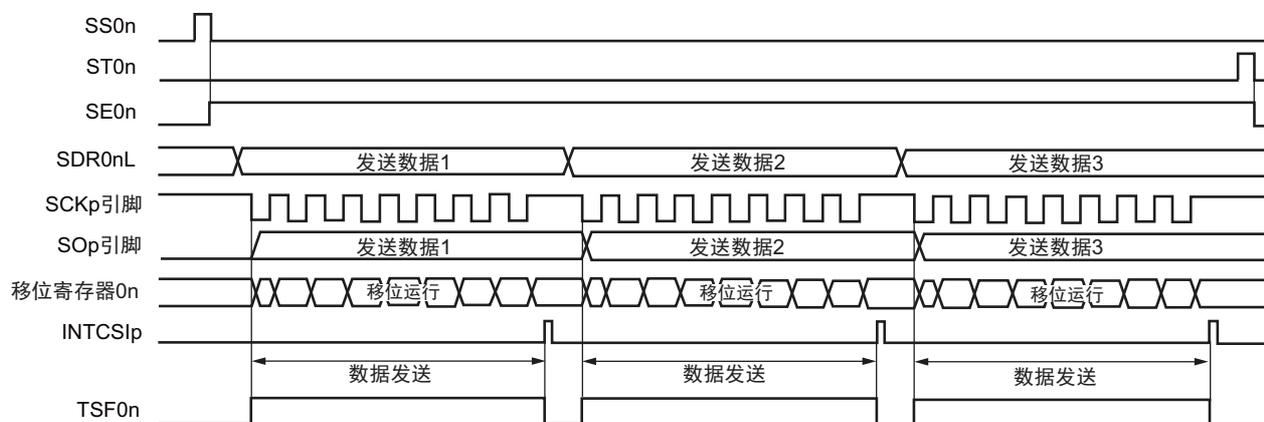
图 12-25 重新开始主控发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等待通信对象(从属设备)的停止或者通信结束后进行初始设定而不是进行重新开始设定。

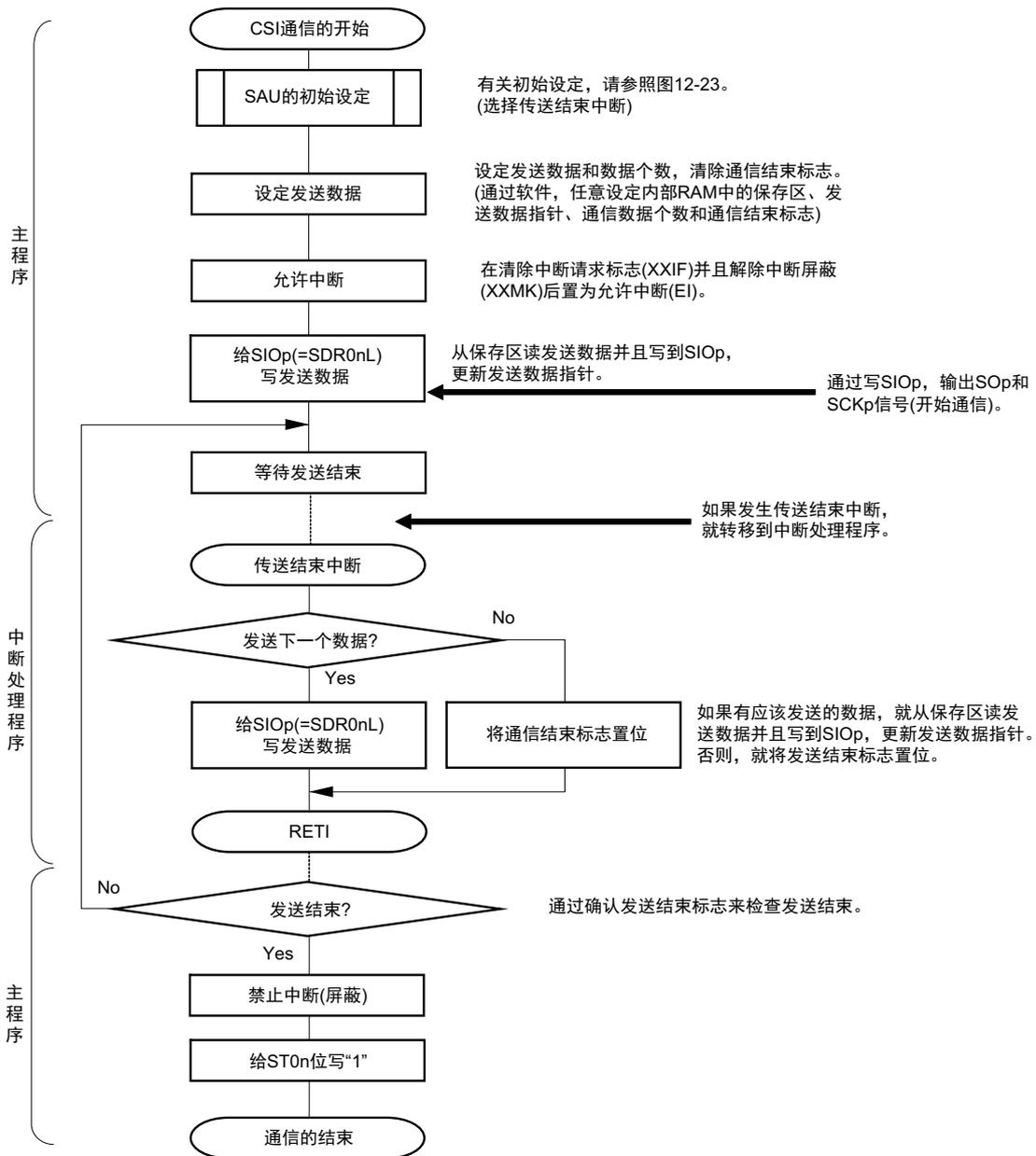
(3) 处理流程（单次发送模式）

图 12-26 主控发送（单次发送模式）的时序图（类型 1: DAP0n=0、CKP0n=0）



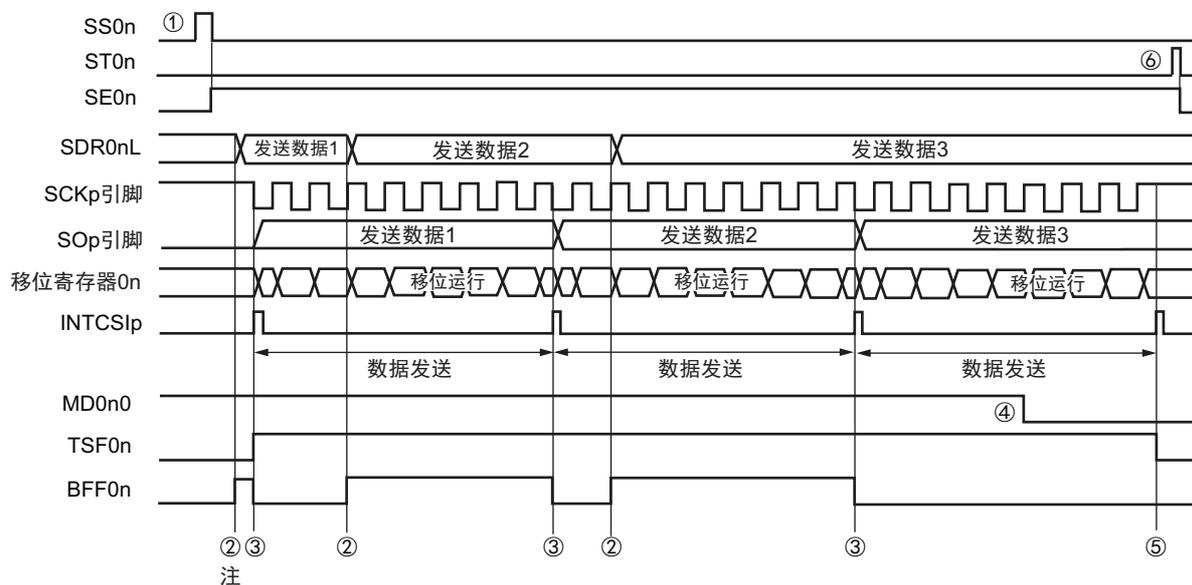
备注 n=0、1, p: CSI号 (p=00、01)

图 12-27 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 12-28 主控发送（连续发送模式）的时序图（类型 1：DAP0n=0、CKP0n=0）

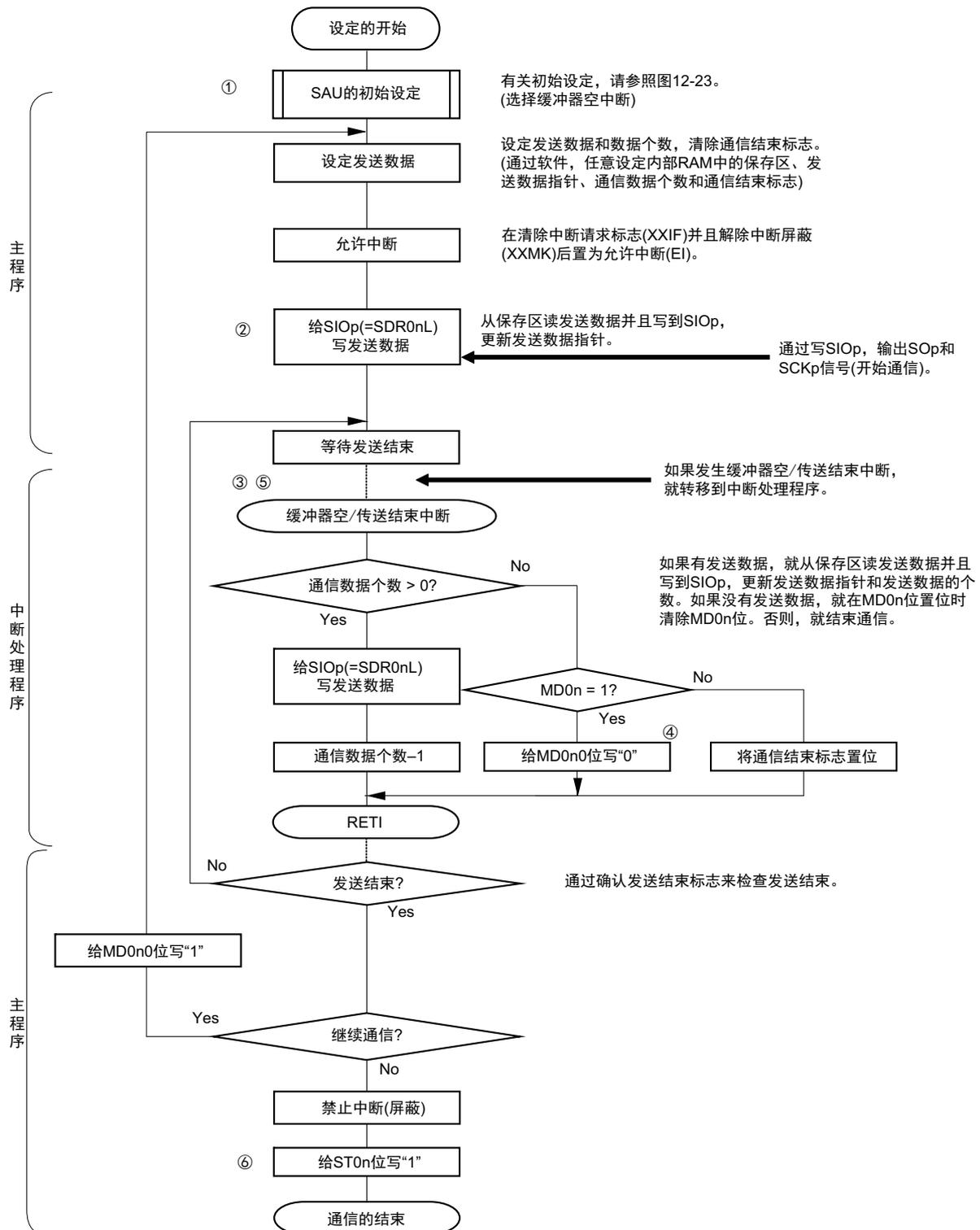


注 如果在串行状态寄存器 0n（SSR0n）的 BFF0n 位为“1”期间（有效数据保存在串行数据寄存器 0n（SDR0nL）时）给 SDR0nL 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 0n（SMR0nH/L）的 MD0n0 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 n=0、1，p：CSI 号（p=00、01）

图 12-29 主控发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 12-28 主控发送（连续发送模式）的时序图”中的①~⑥。

12.5.2 主控接收

主控接收是 RL78/G10 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	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SI00	SCK01、SI01
中断	INTCSI00	INTCSI01
	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（OVF0n）。	
传送数据长度	7 位或者 8 位	
传送速率注 2	Max. $f_{CLK}/4$ [Hz] ($SDR0nH[7:1] \geq 1$) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
数据相位	能通过 SCR0n 寄存器的 DAP0n 位进行选择。 <ul style="list-style-type: none"> DAP0n=0: 在串行时钟开始运行时, 开始数据输出。 DAP0n=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 <ul style="list-style-type: none"> CKP0n=0: 不反相 CKP0n=1: 反相 	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

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 24 章 电特性”）的范围内使用。

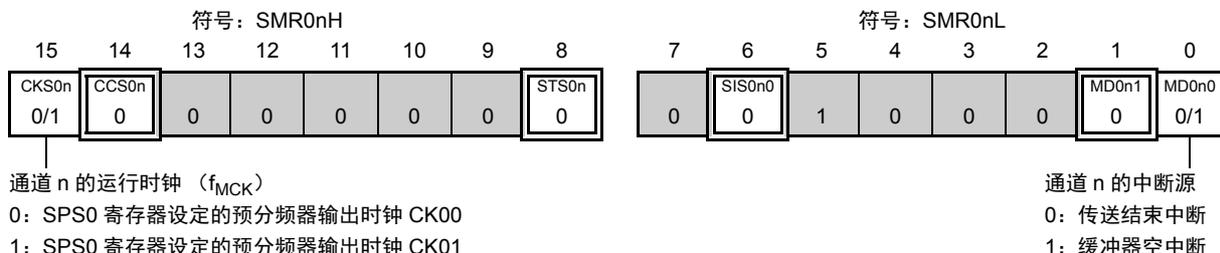
备注 1. f_{CLK} : 系统时钟频率

2. n=0、1

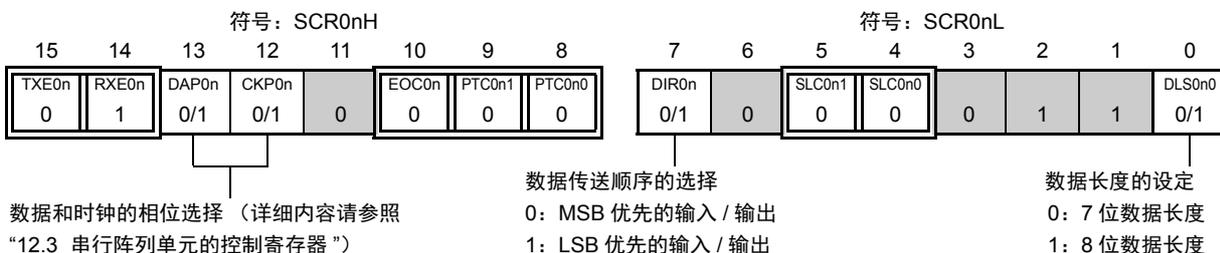
(1) 寄存器的设定

图 12-30 3 线串行 I/O (CSI00、CSI01 注) 主控接收时的寄存器设定内容例子 (1/2)

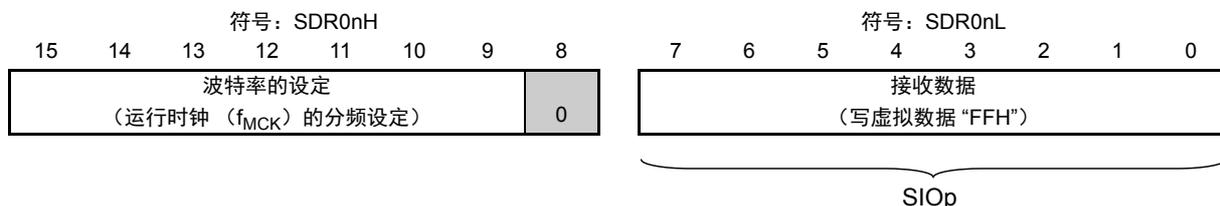
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



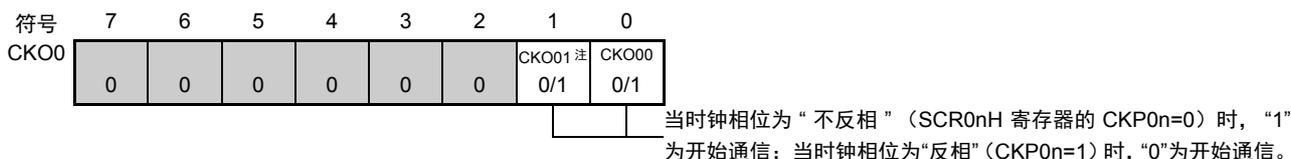
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



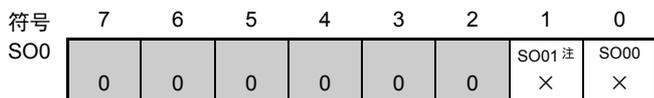
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 只设定对象通道的位。



(e) 串行输出寄存器 0 (SO0) 在此模式中不使用。



注 只限于 16 引脚产品。

图 12-30 3 线串行 I/O (CSI00、CSI01 注) 主控接收时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 在此模式中不使用。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 ×	SOE00 ×

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	1	1	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- | |
|--|
| |
|--|

 : 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 12-31 主控接收的初始设定步骤

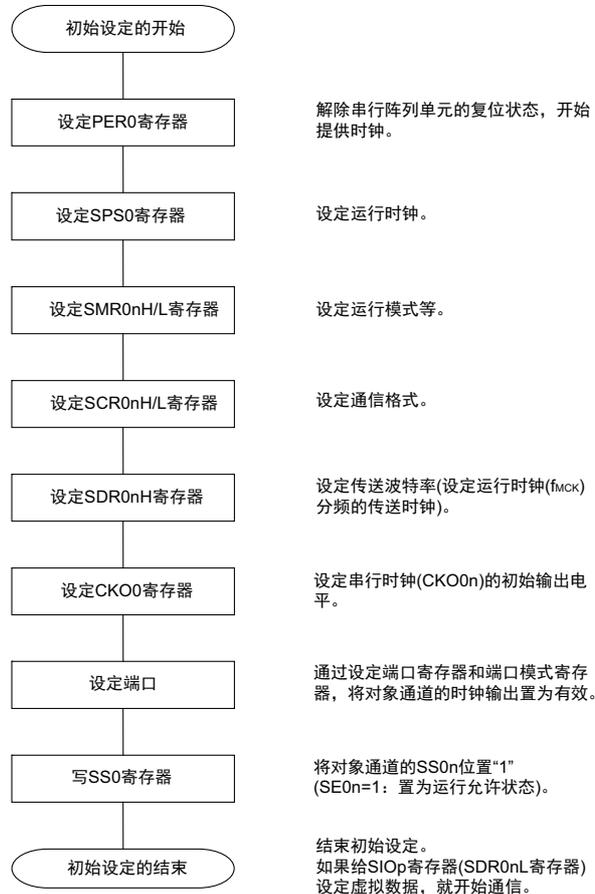


图 12-32 主控接收的中止步骤

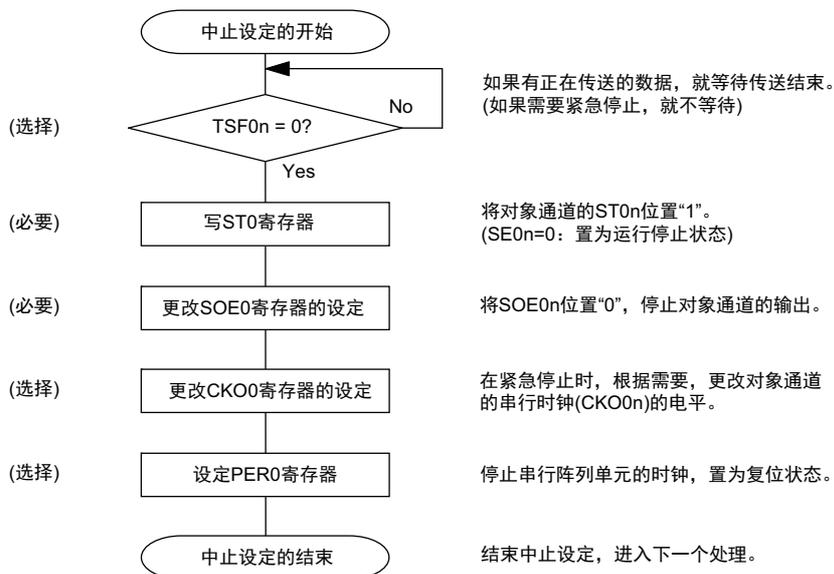
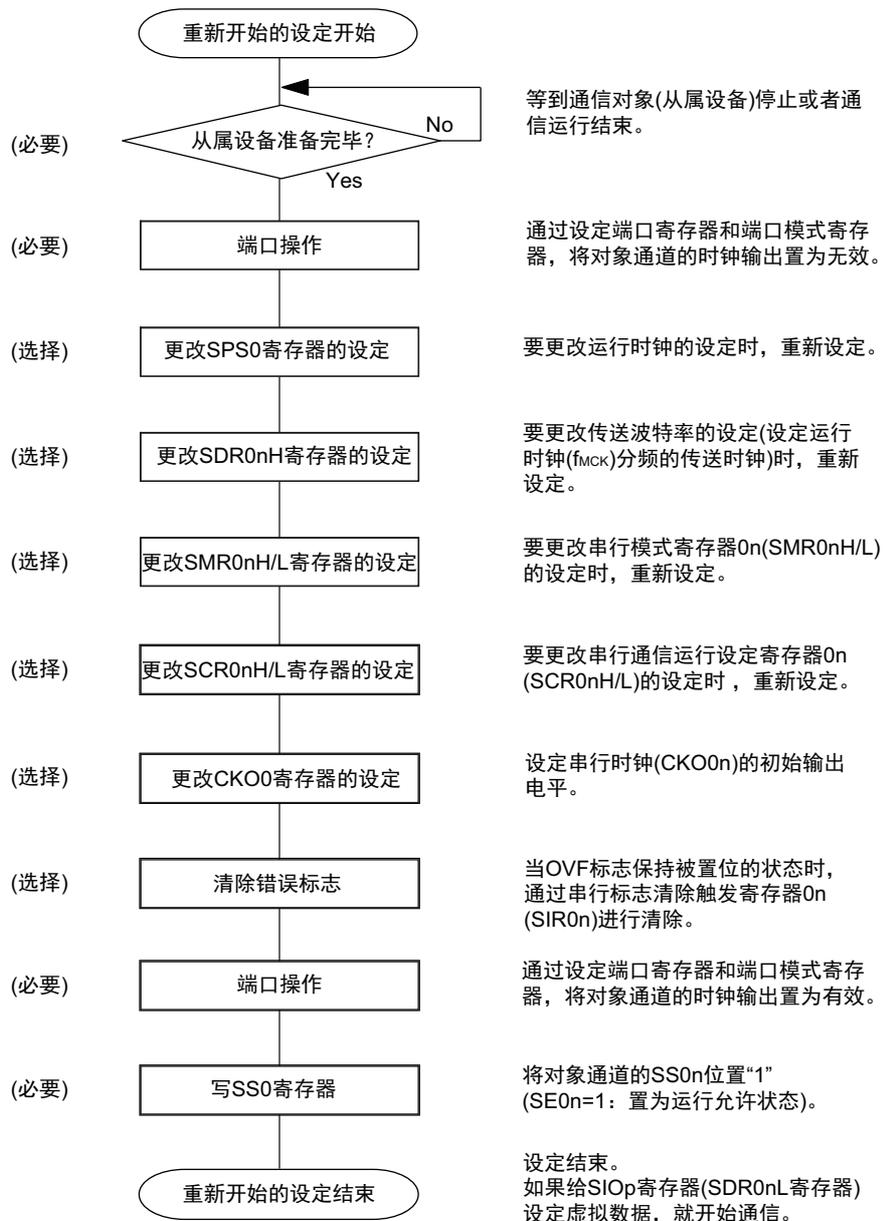


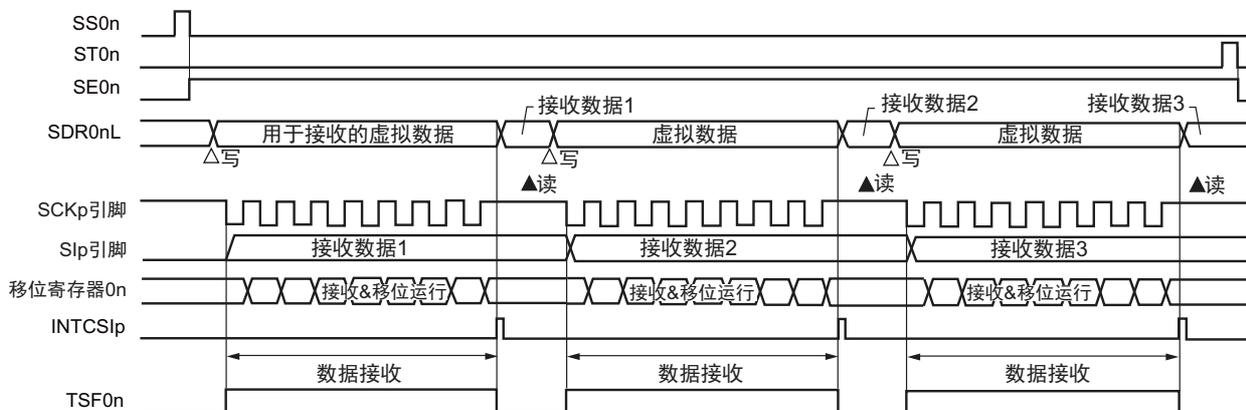
图 12-33 重新开始主控接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等待通信对象（从属设备）的停止或者通信结束后进行初始设定而不是进行重新开始设定。

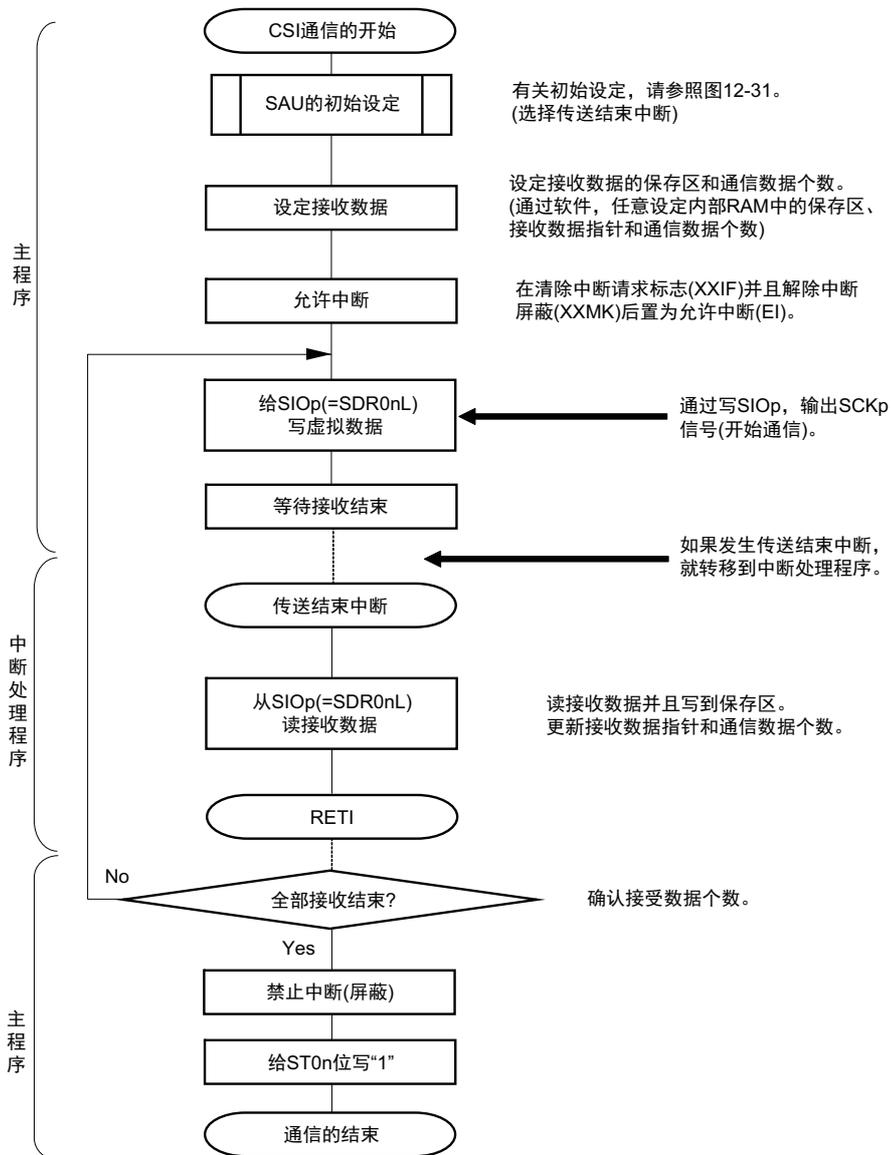
(3) 处理流程（单次接收模式）

图 12-34 主控接收（单次接收模式）的时序图（类型 1: DAP0n=0、CKP0n=0）



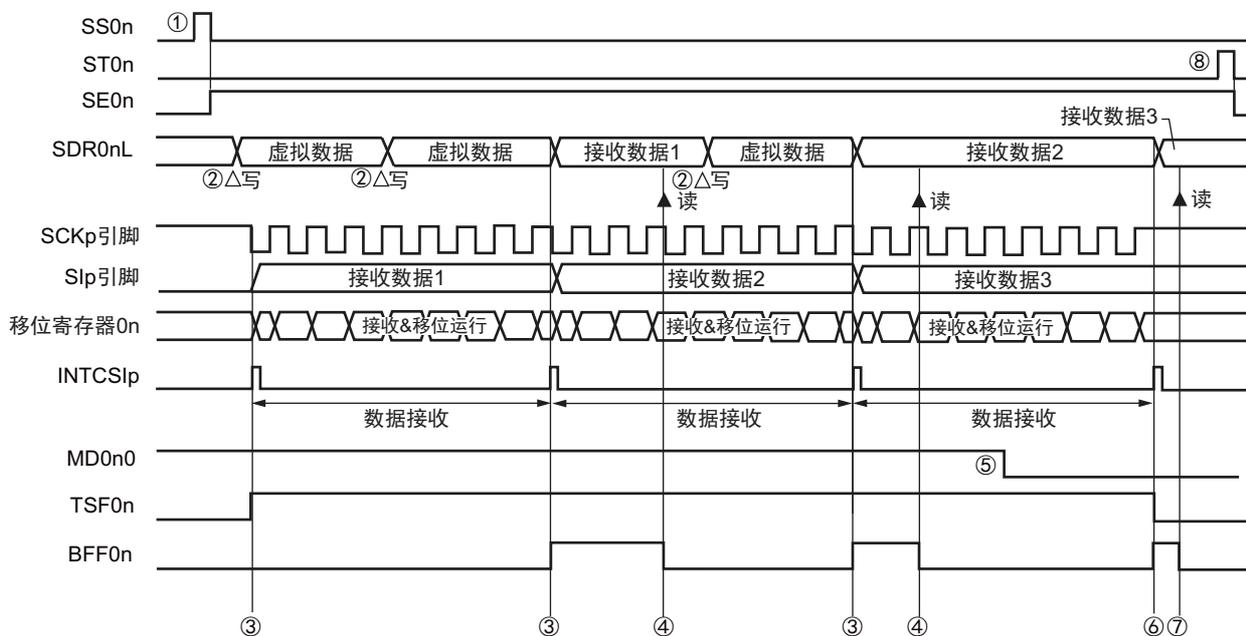
备注 n=0、1, p: CSI号 (p=00、01)

图 12-35 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 12-36 主控接收（连续接收模式）的时序图（类型 1：DAP0n=0、CKP0n=0）

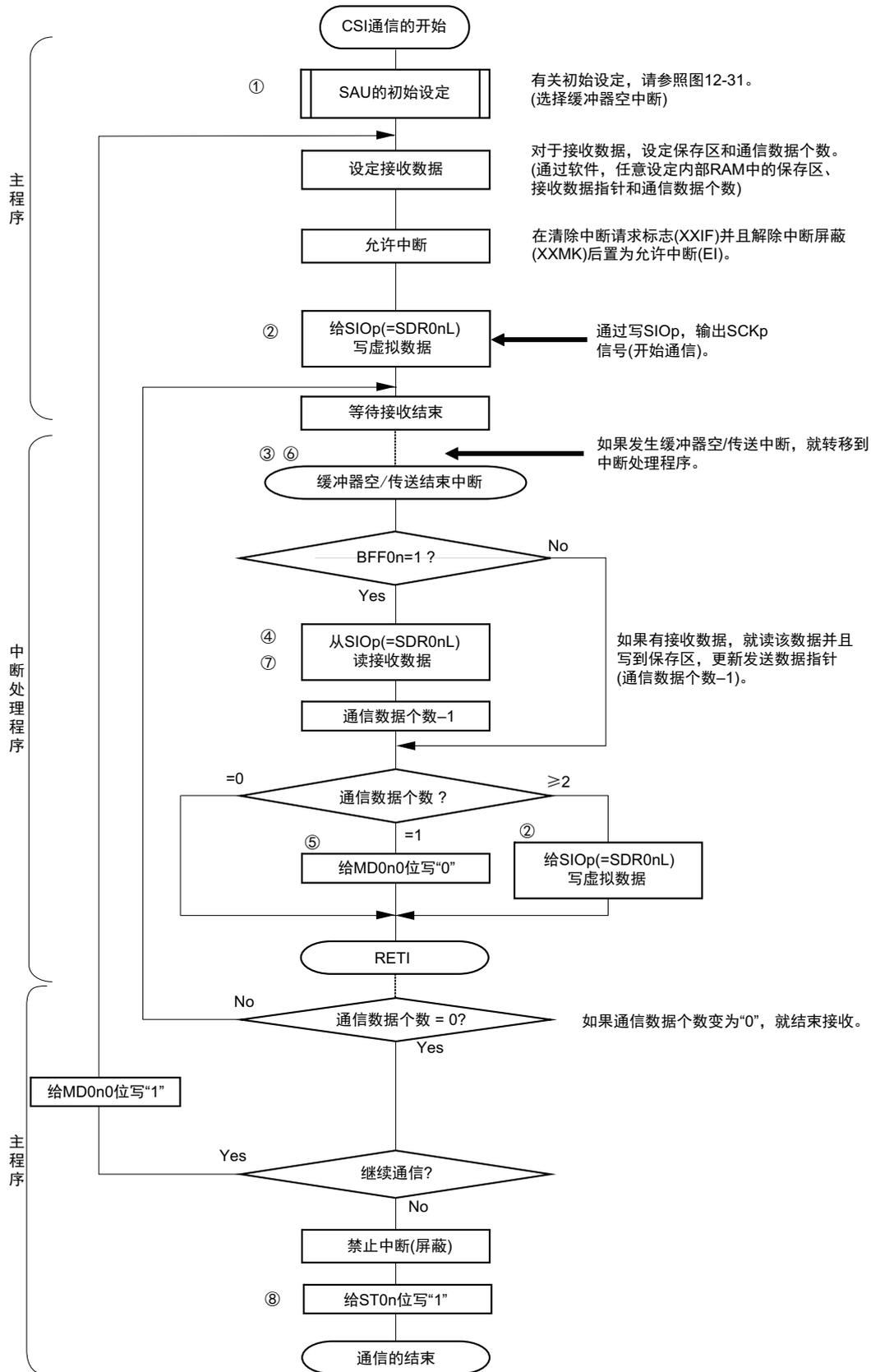


注意 即使在运行过程中也能改写 MD0n0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注 1. 图中的①~⑧对应“图 12-37 主控接收（连续接收模式）的流程图”中的①~⑧。

2. n=0、1，p: CSI 号 (p=00、01)

图 12-37 主控接收（连续接收模式）的流程图



备注 图中的①~⑧对应“图 12-36 主控接收（连续接收模式）的时序图”中的①~⑧。

12.5.3 主控的发送和接收

主控的发送和接收是 RL78/G10 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	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SI00、SO00	SCK01、SI01、SO01
中断	INTCSI00	INTCSI01
	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（OVF0n）。	
传送数据长度	7 位或者 8 位	
传送速率注 2	Max. $f_{CLK}/4$ [Hz] ($SDR0nH[7:1] \geq 1$) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
数据相位	能通过 SCR0nH 寄存器的 DAP0n 位进行选择。 • DAP0n=0: 在串行时钟开始运行时, 开始数据输出。 • DAP0n=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 • CKP0n=0: 不反相 • CKP0n=1: 反相	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

2. 必须在满足此条件并且满足电特性的 AC 特性（参照“第 24 章 电特性”）的范围内使用。

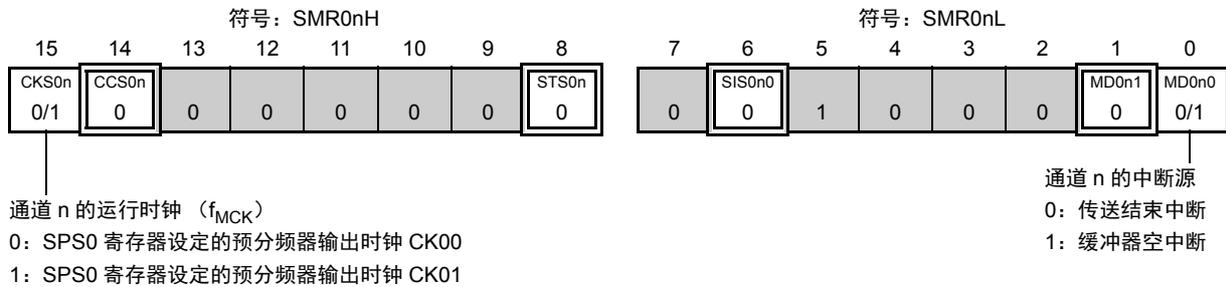
备注 1. f_{CLK} : 系统时钟频率

2. n=0、1

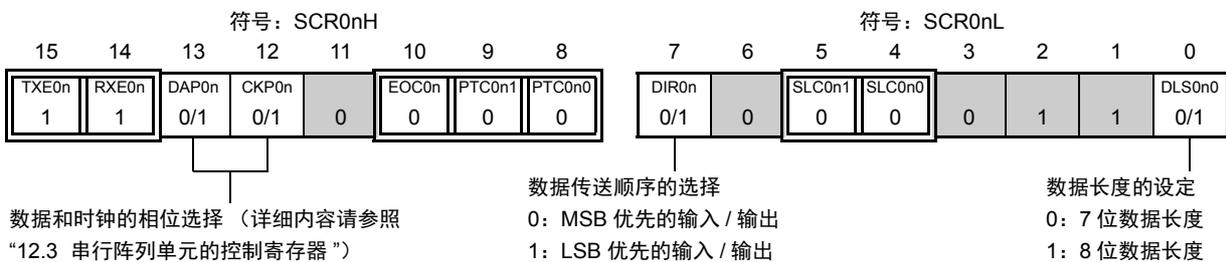
(1) 寄存器的设定

图 12-38 3 线串行 I/O (CSI00、CSI01 注) 主控发送和接收时的寄存器设定内容例子 (1/2)

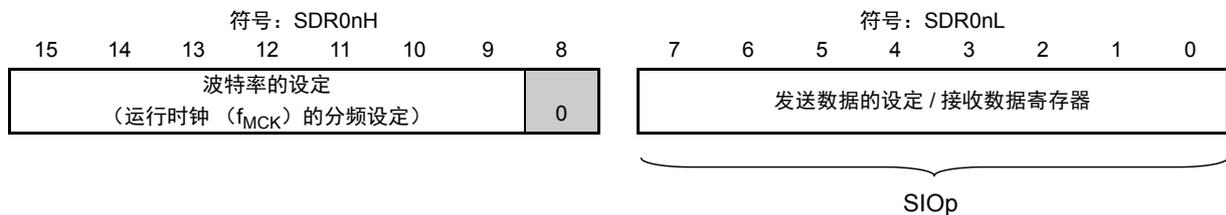
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



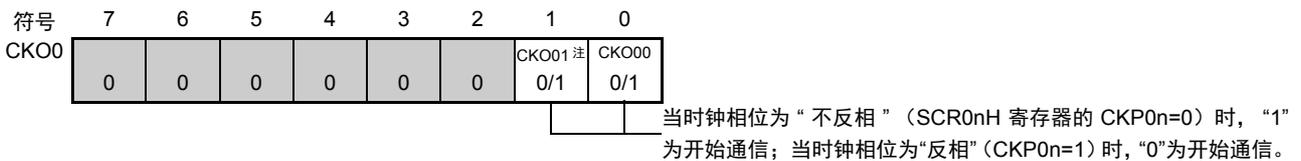
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



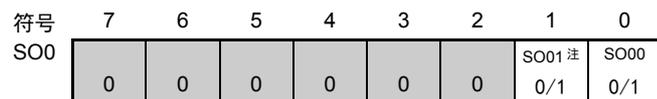
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 只设定对象通道的位。



(e) 串行输出寄存器 0 (SO0) 只设定对象通道的位。



注 只限于 16 引脚产品。

图 12-38 3 线串行 I/O (CSI00、CSI01 注) 主控发送和接收时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 0/1	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- | |
|--|
| |
|--|

 : 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 12-39 主控发送和接收的初始设定步骤

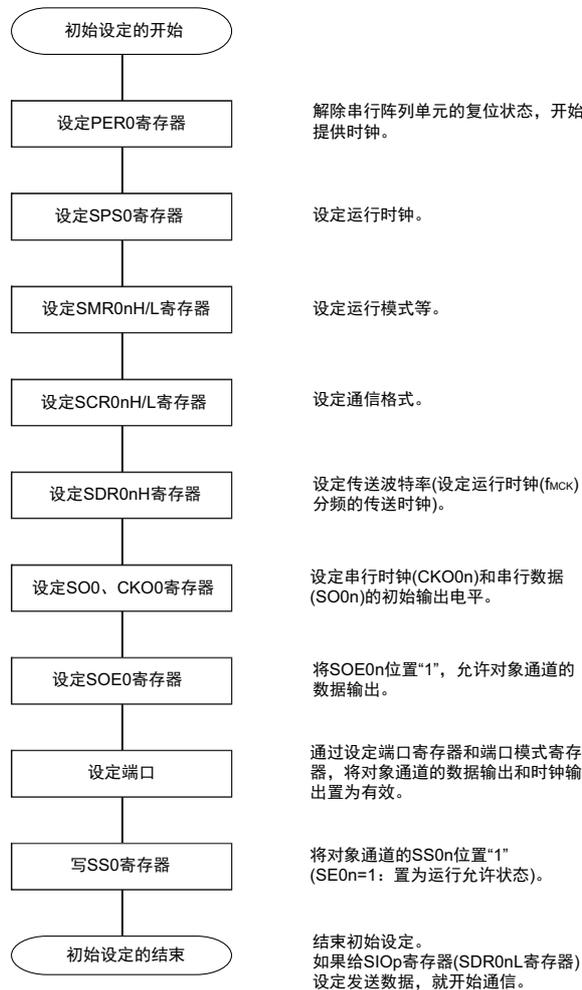


图 12-40 主控发送和接收的中止步骤

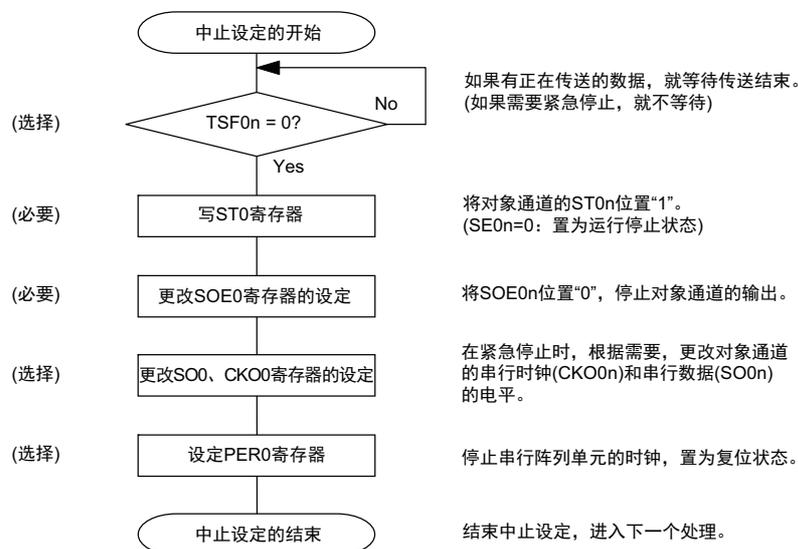
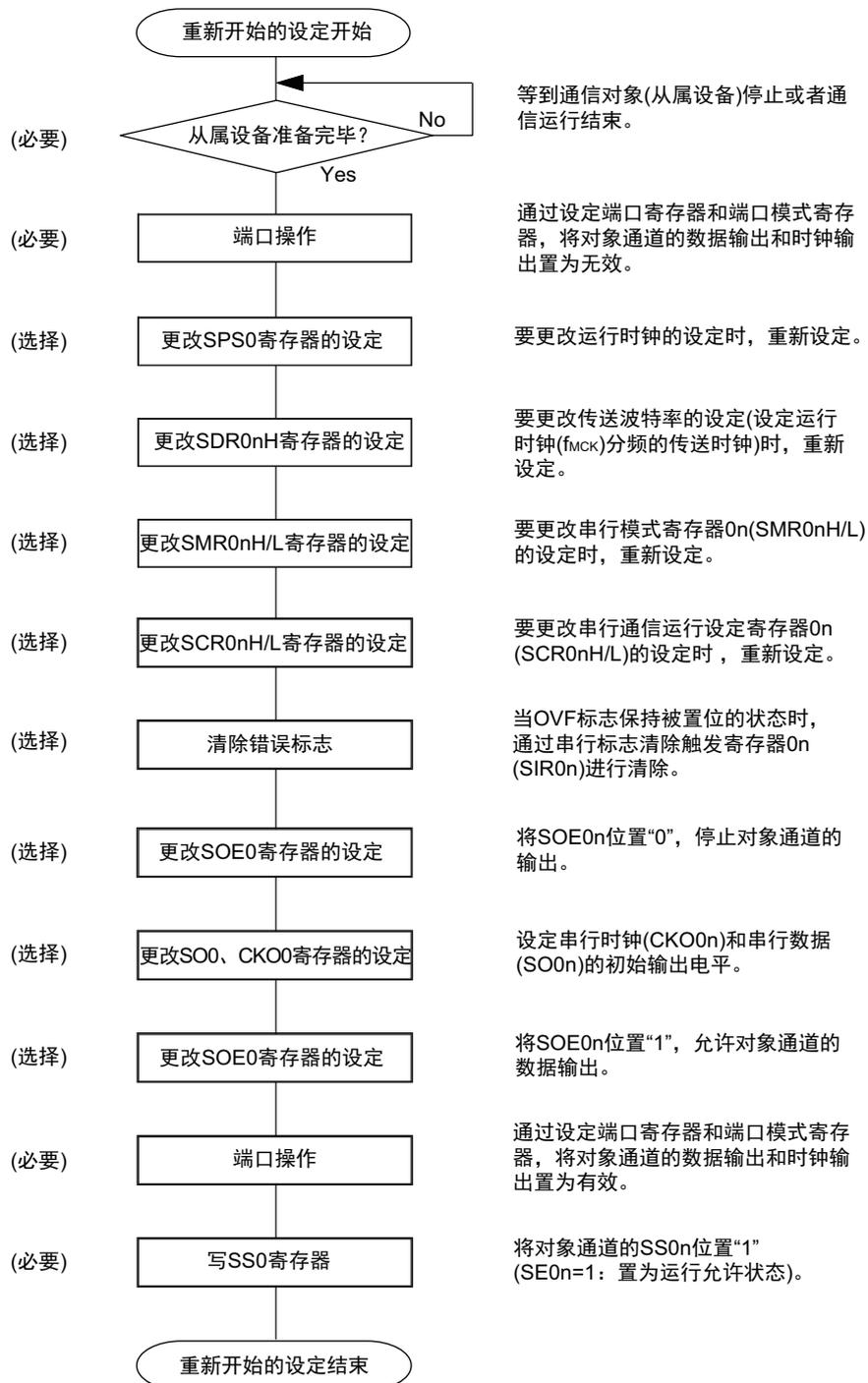


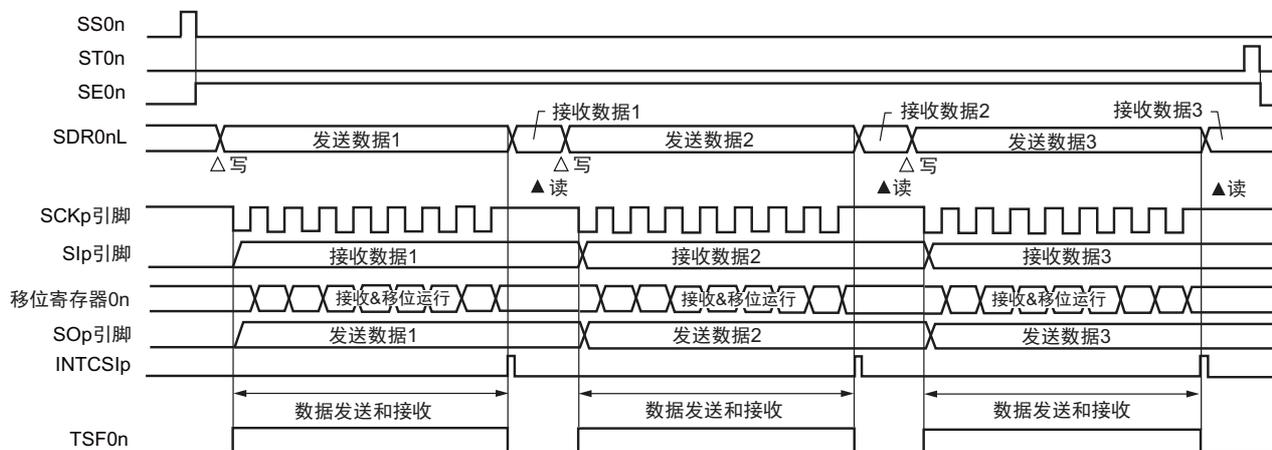
图 12-41 重新开始主控发送和接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等待通信对象 (从属设备) 的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。

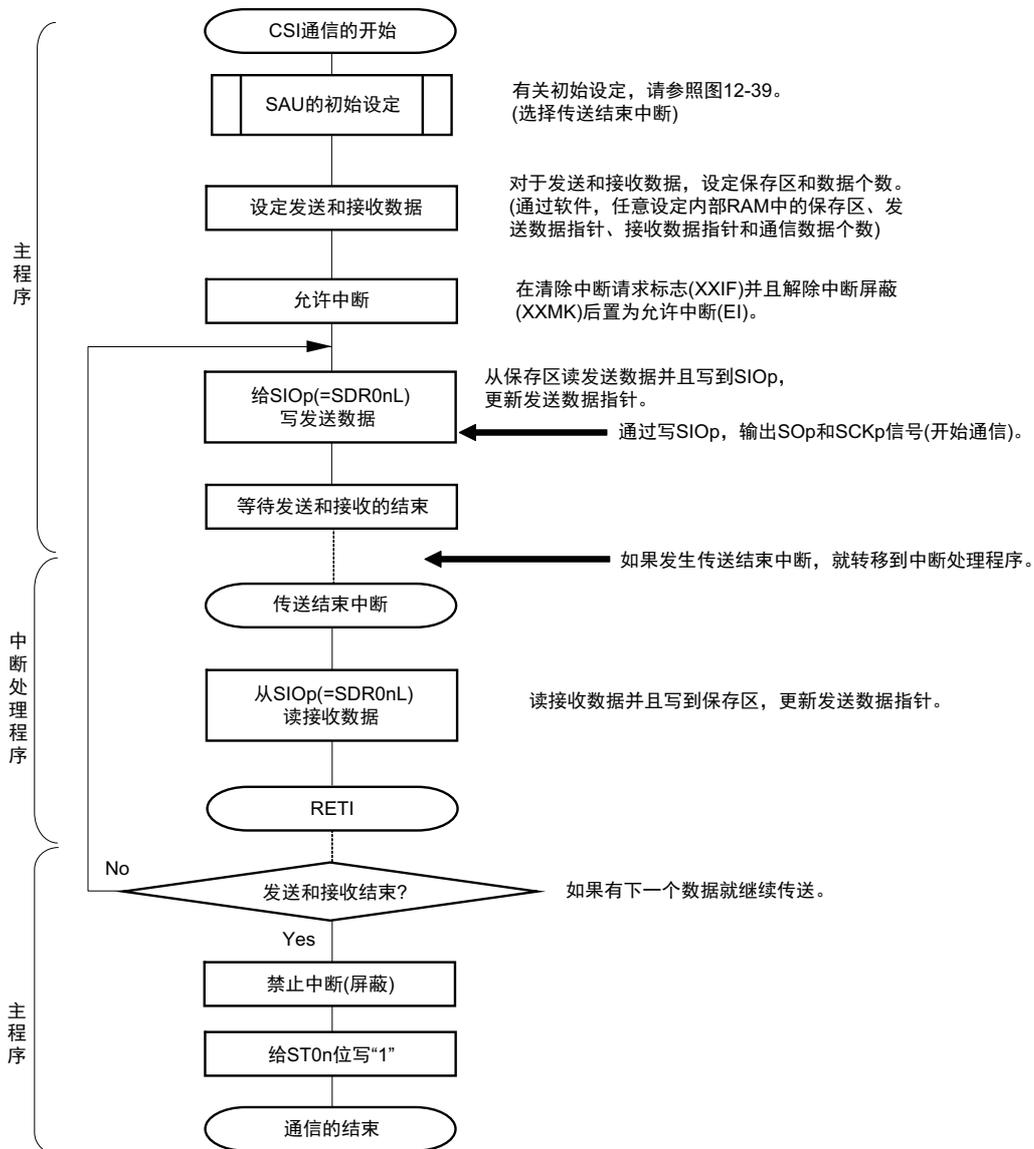
(3) 处理流程（单次发送和接收模式）

图 12-42 主控发送和接收（单次发送和接收模式）的时序图（类型 1：DAP0n=0、CKP0n=0）



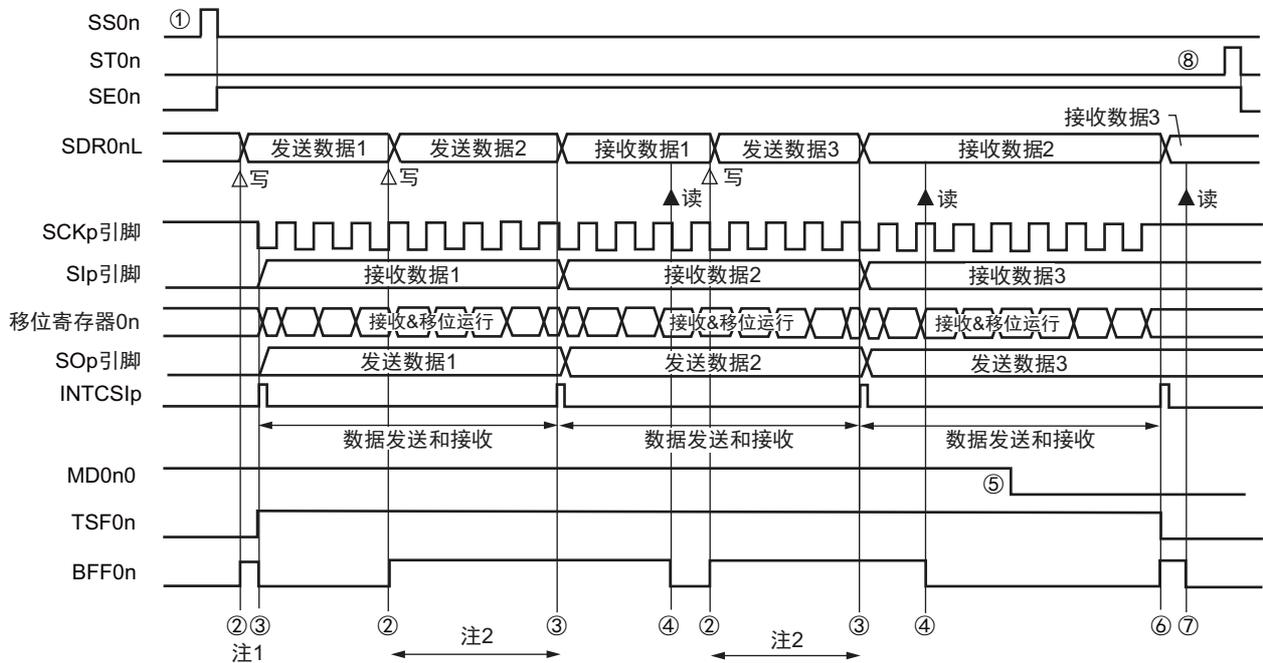
备注 n=0、1, p: CSI号 (p=00、01)

图 12-43 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 12-44 主控发送和接收（连续发送和接收模式）的时序图（类型 1：DAP0n=0、CKP0n=0）



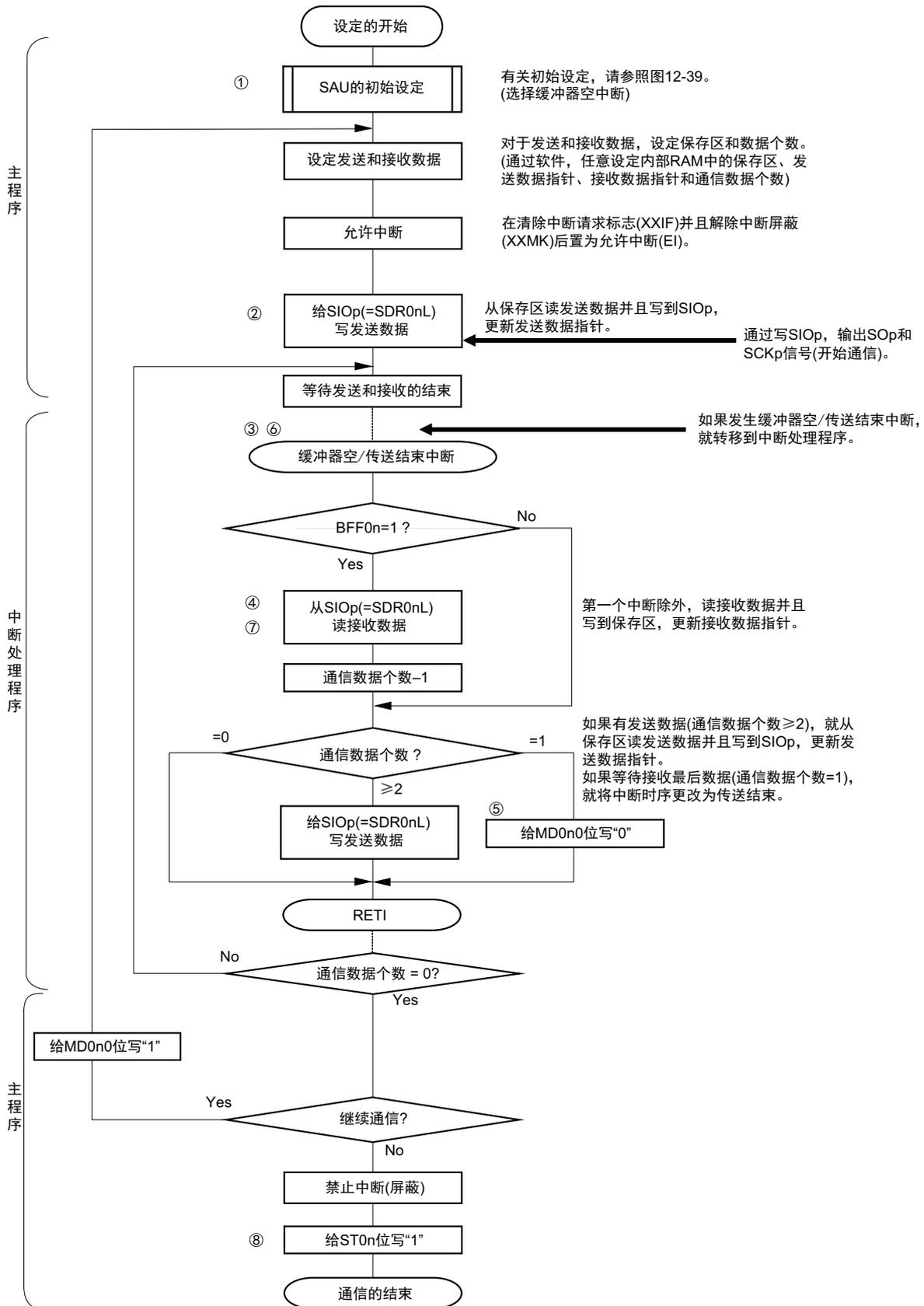
- 注 1. 如果在串行状态寄存器 0n（SSR0n）的 BFF0n 位为“1”期间（有效数据保存在串行数据寄存器 0n（SDR0nL）时）给 SDR0nL 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDR0nL 寄存器，就能读发送数据。此时，不影响传送运行。

注意 即使在运行中也能改写串行模式寄存器 0n（SMR0nL）的 MD0n0 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 12-45 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. n=0、1，p: CSI 号（p=00、01）

图 12-45 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 12-44 主控发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

12.5.4 从属发送

从属发送是在从其他设备输入传送时钟的状态下 RL78/G10 将数据发送给其他设备的运行。

3 线串行 I/O	CSI00	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SO00	SCK01、SO01
中断	INTCSI00	INTCSI01
	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（OVF0n）。	
传送数据长度	7 位或者 8 位	
传送速率	Max. $f_{MCK}/6$ [Hz] 注 2、3	
数据相位	能通过 SCR0nH 寄存器的 DAP0n 位进行选择。 <ul style="list-style-type: none"> • DAP0n=0: 在串行时钟开始运行时, 开始数据输出。 • DAP0n=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 <ul style="list-style-type: none"> • CKP0n=0: 不反相 • CKP0n=1: 反相 	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

2. 因为在内部对 SCK00、SCK01 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]。

3. 必须在满足此条件并且满足电特性的 AC 特性（参照“第 24 章 电特性”）的范围内使用。

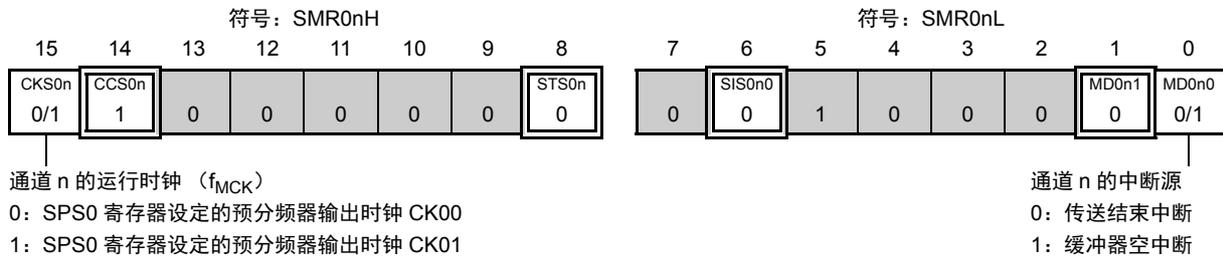
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. n=0、1

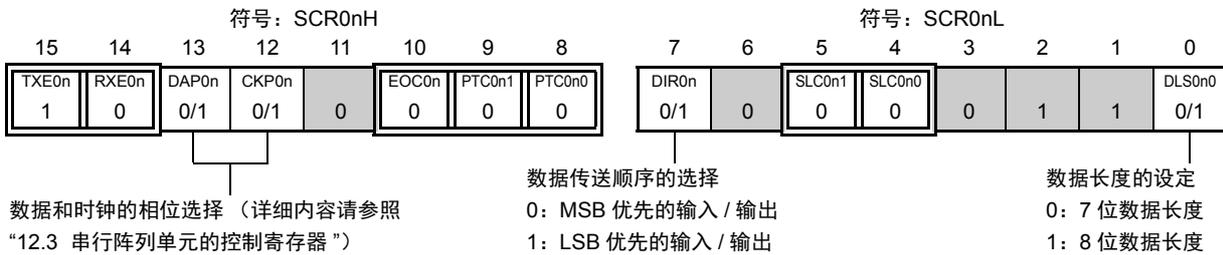
(1) 寄存器的设定

图 12-46 3 线串行 I/O (CSI00、CSI01 注) 从属发送时的寄存器设定内容例子 (1/2)

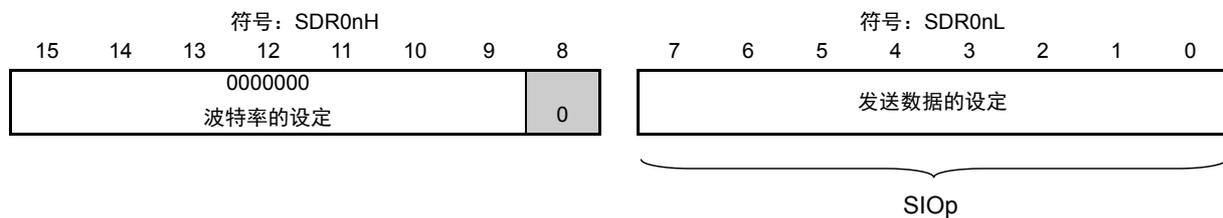
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



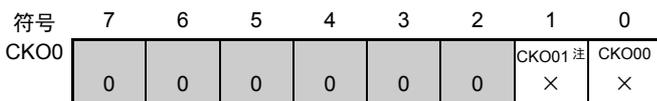
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



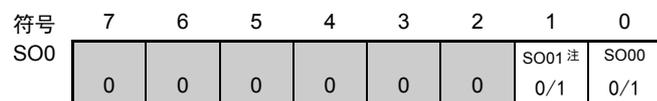
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 在此模式中不使用。



(e) 串行输出寄存器 0 (SO0) 只设定对象通道的位。



注 只限于 16 引脚产品。

图 12-46 3 线串行 I/O (CSI00、CSI01 注) 从属发送时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 0/1	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- | |
|--|
| |
|--|

 : 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 12-47 从属发送的初始设定步骤

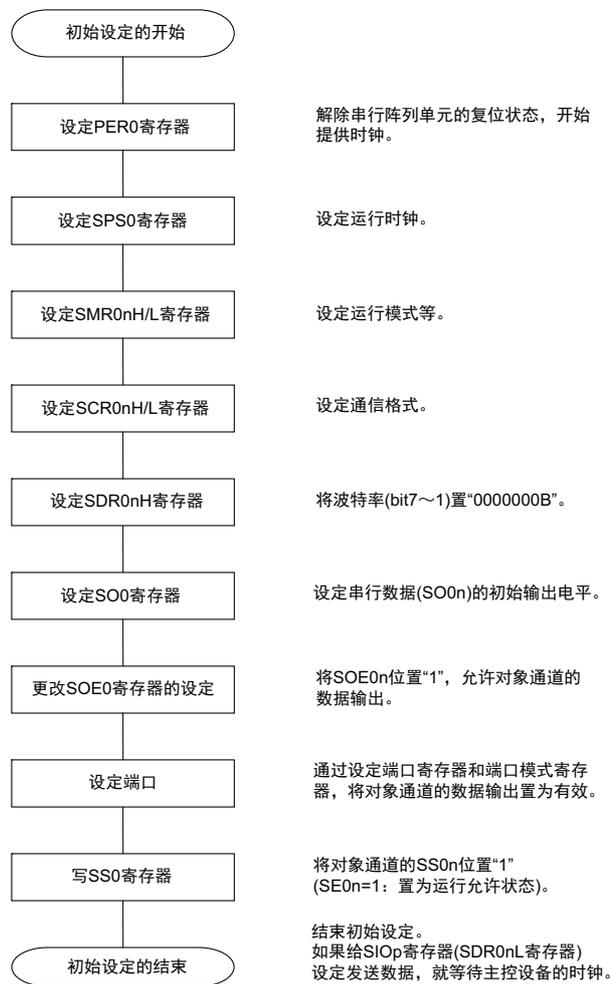


图 12-48 从属发送的中止步骤

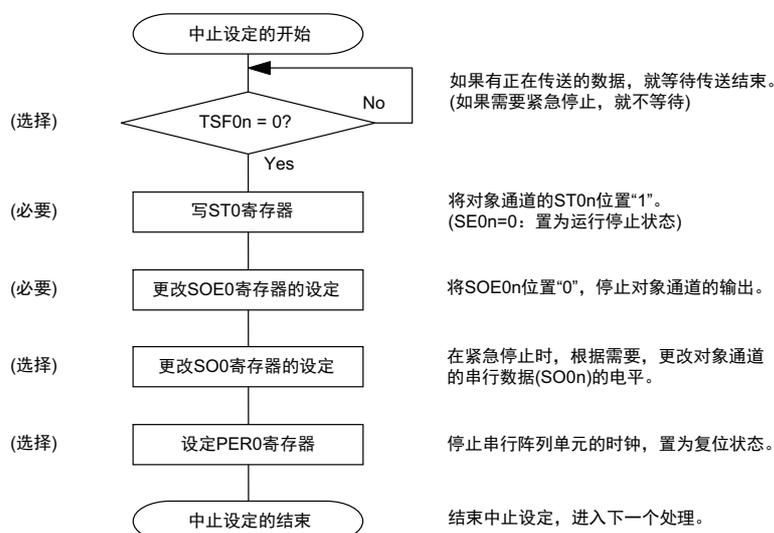
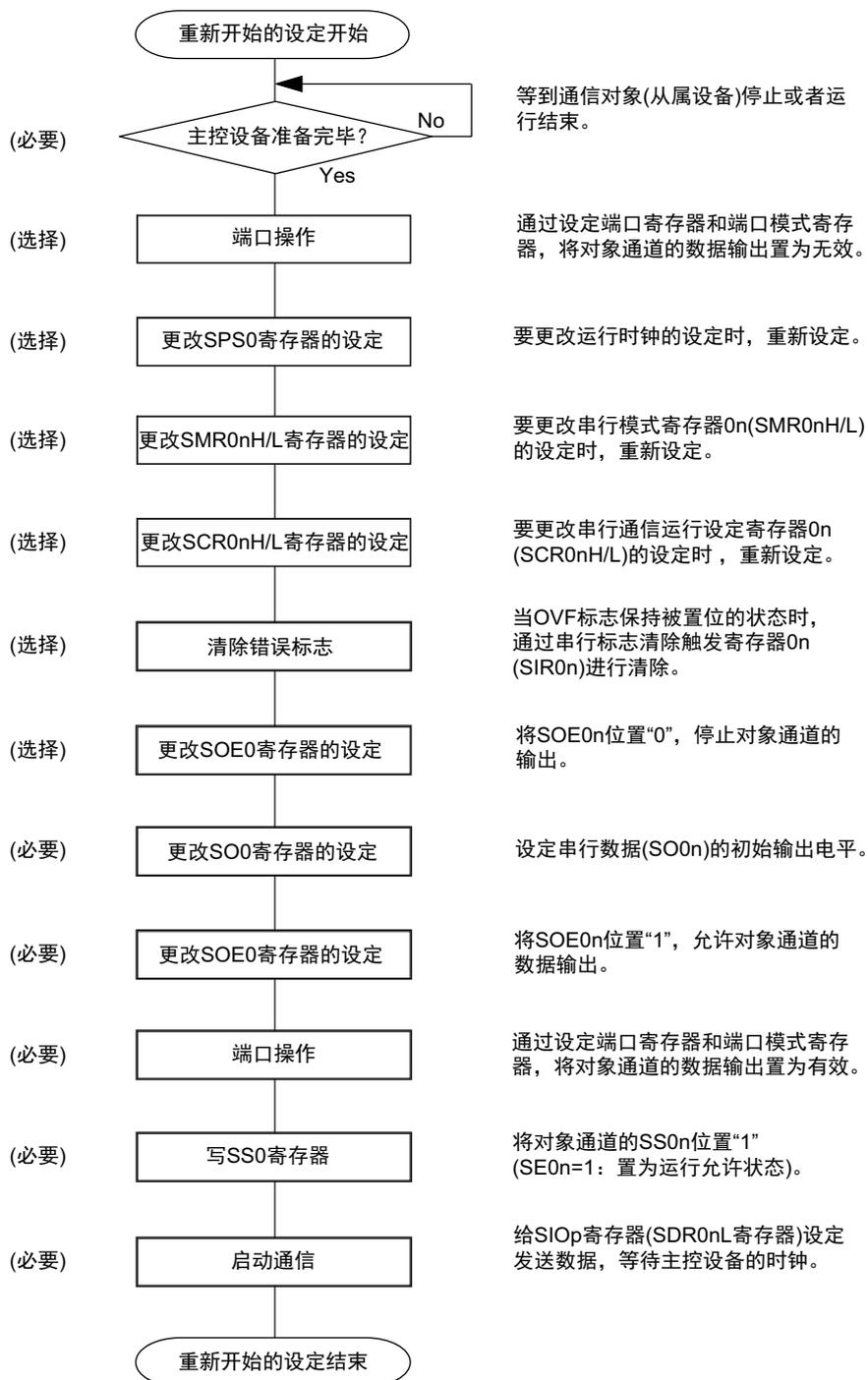


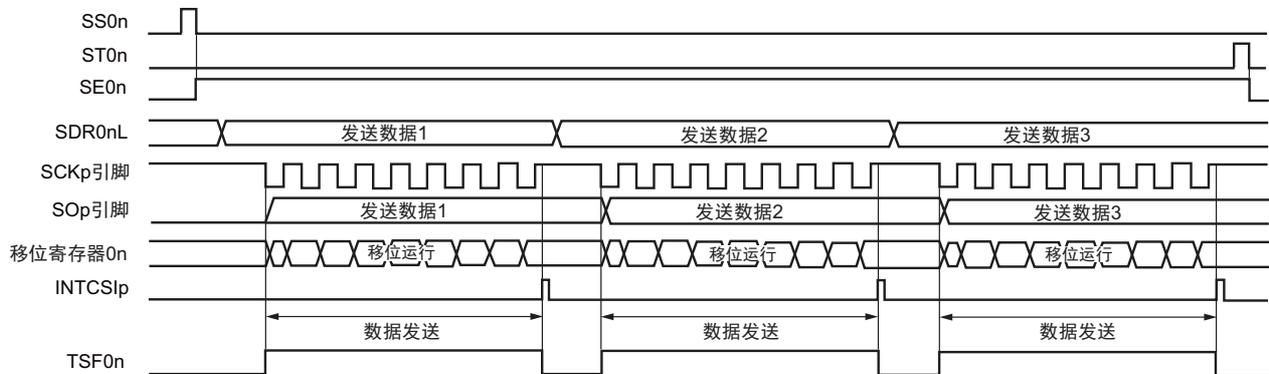
图 12-49 重新开始从属发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等待通信对象（主控设备）的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。

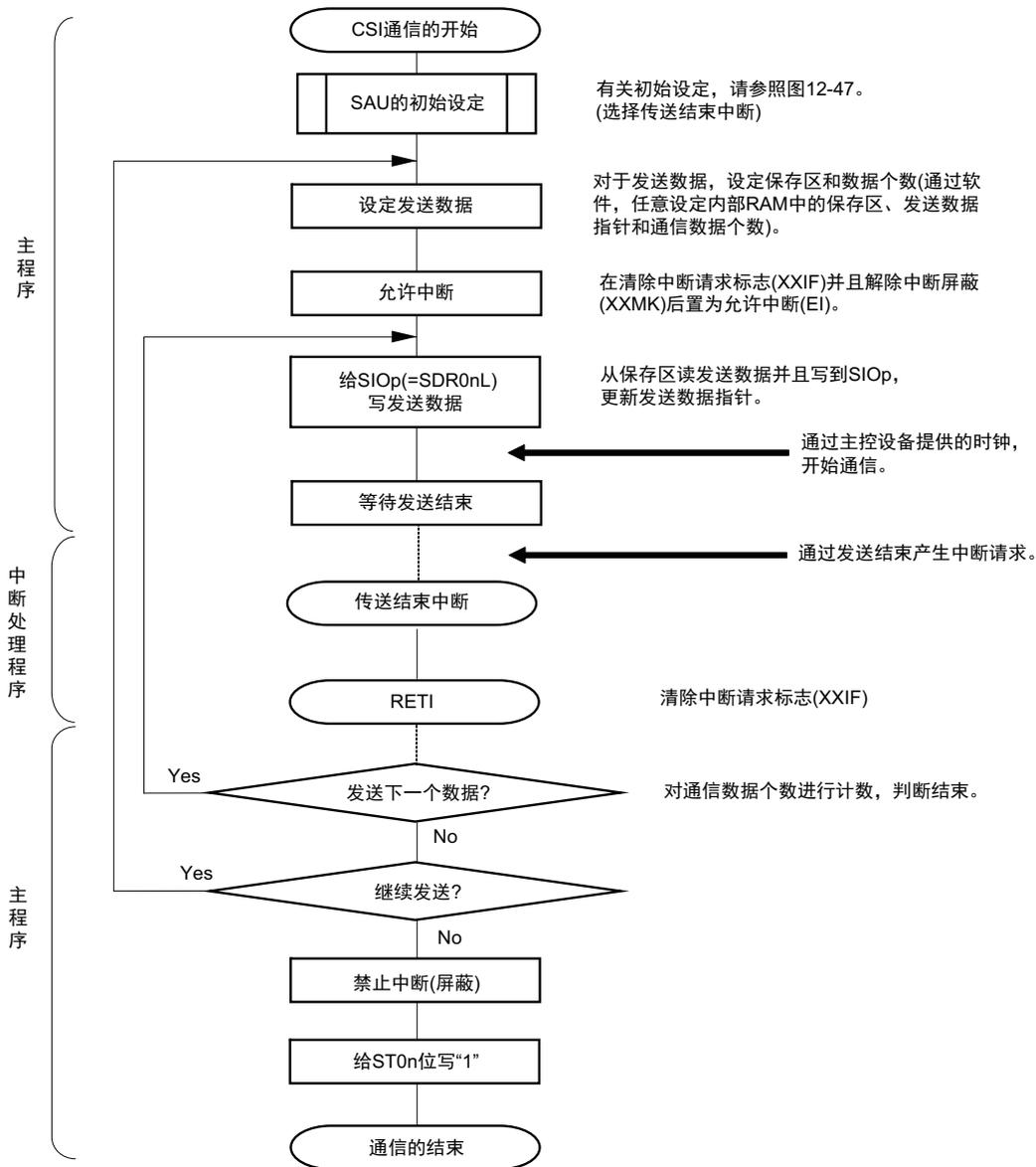
(3) 处理流程（单次发送模式）

图 12-50 从属发送（单次发送模式）的时序图（类型 1: DAP0n=0、CKP0n=0）



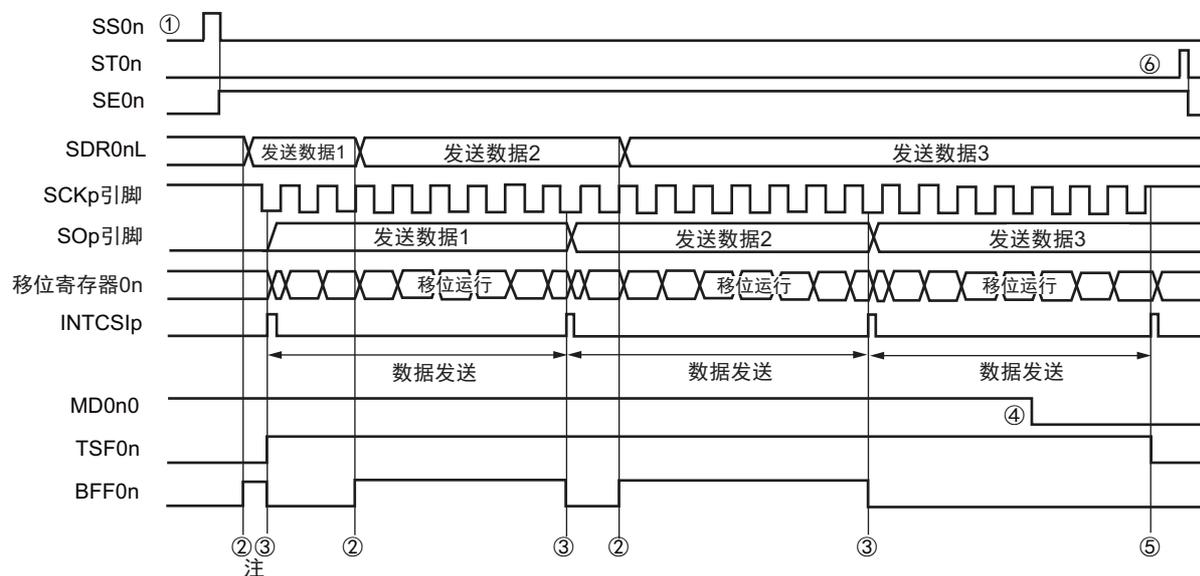
备注 n=0、1, p: CSI号 (p=00、01)

图 12-51 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 12-52 从属发送（连续发送模式）的时序图（类型 1：DAP0n=0、CKP0n=0）

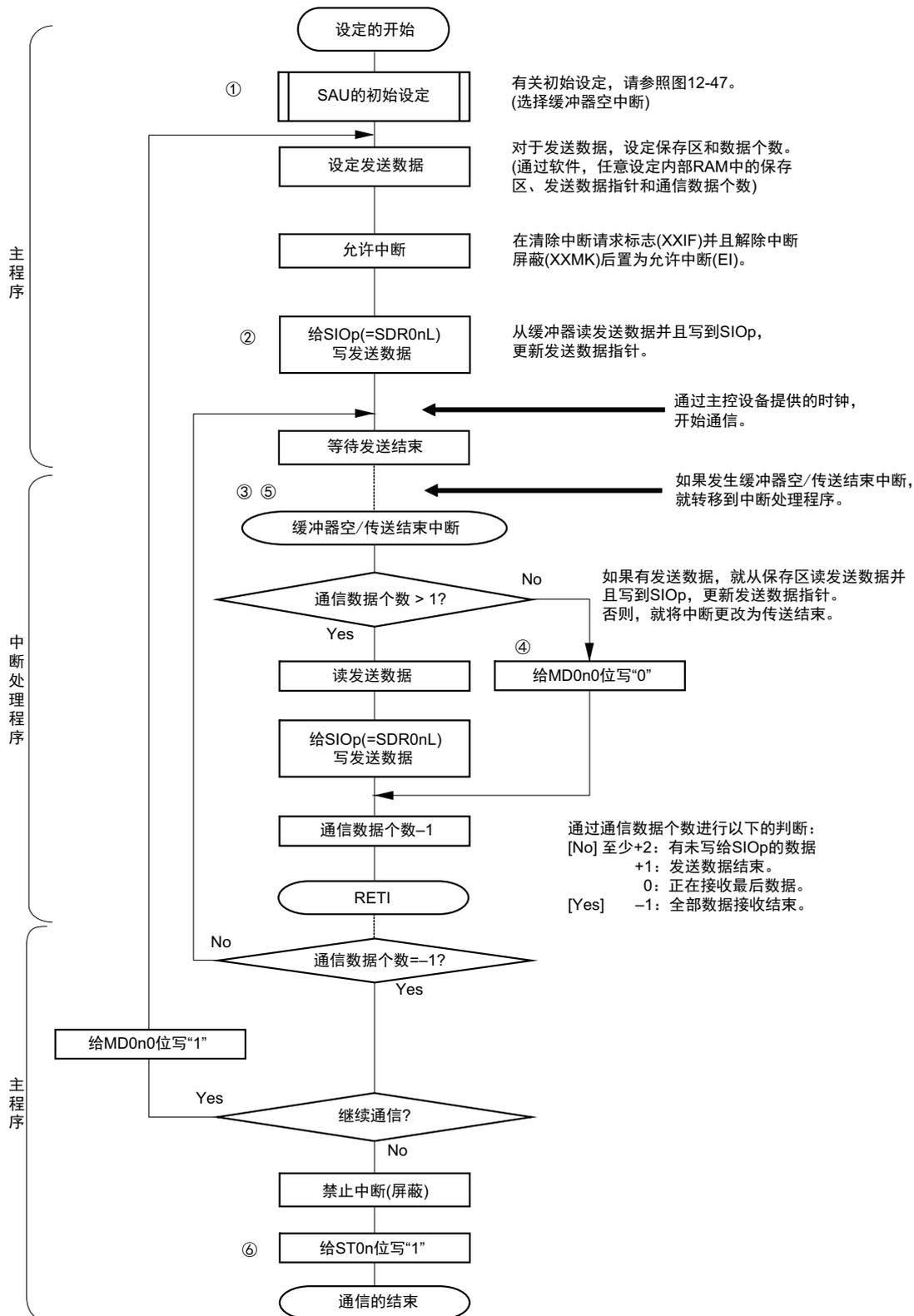


注 如果在串行状态寄存器 0n（SSR0n）的 BFF0n 位为“1”期间（有效数据保存在串行数据寄存器 0n（SDR0nL）时）给 SDR0nL 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 0n（SMR0nL）的 MD0n0 位。但是，必须在开始传送最后一位之前进行改写。

备注 n=0、1，p：CSI 号（p=00、01）

图 12-53 从属发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 12-52 从属发送（连续发送模式）的时序图”中的①~⑥。

12.5.5 从属接收

从属接收是在其他设备输入传送时钟的状态下 RL78/G10 从其他设备接收数据的运行。

3 线串行 I/O	CSI00	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SI00	SCK01、SI01
中断	INTCSI00	INTCSI01
	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（OVF0n）。	
传送数据长度	7 位或者 8 位	
传送速率	Max. $f_{MCK}/6$ [Hz] 注 2、3	
数据相位	能通过 SCR0nH 寄存器的 DAP0n 位进行选择。 <ul style="list-style-type: none"> • DAP0n=0: 在串行时钟开始运行时, 开始数据输出。 • DAP0n=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 <ul style="list-style-type: none"> • CKP0n=0: 不反相 • CKP0n=1: 反相 	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

2. 因为在内部对 SCK00、SCK01 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]。

3. 必须在满足此条件并且满足电特性的 AC 特性（参照“第 24 章 电特性”）的范围内使用。

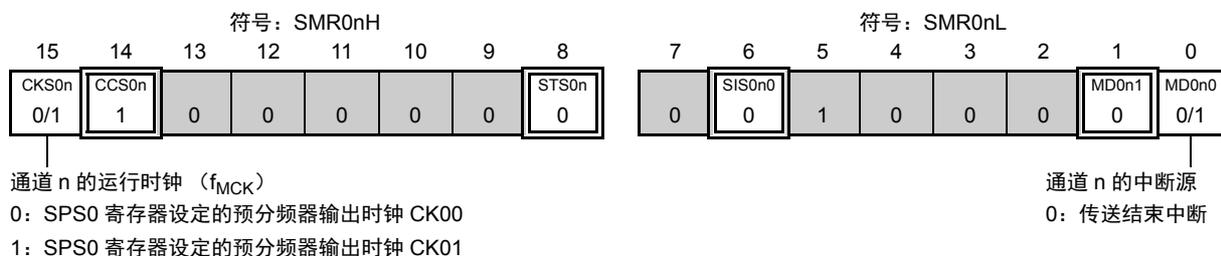
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. n=0、1

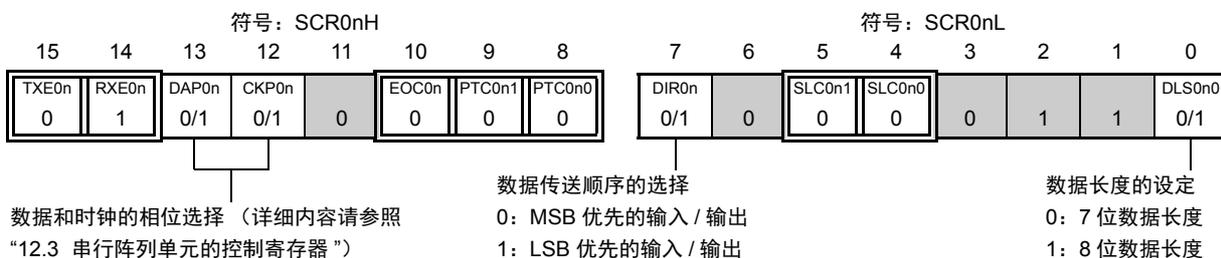
(1) 寄存器的设定

图 12-54 3 线串行 I/O (CSI00、CSI01 注) 从属接收时的寄存器设定内容例子 (1/2)

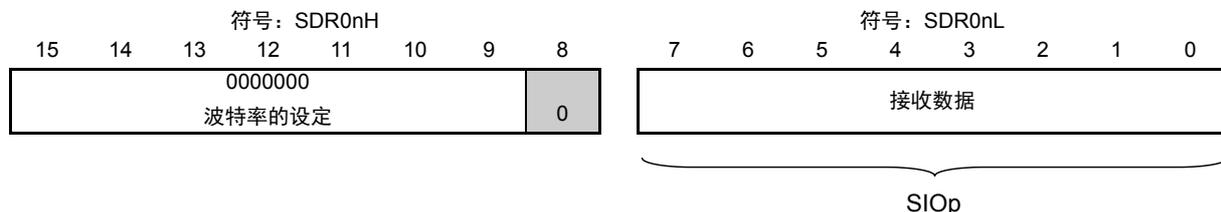
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



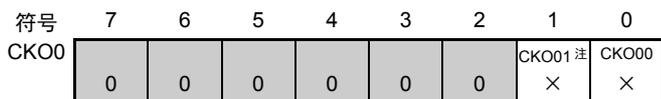
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



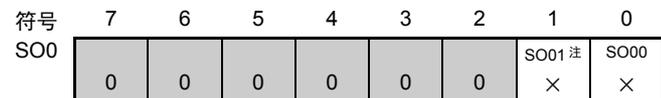
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 在此模式中不使用。



(e) 串行输出寄存器 0 (SO0) 在此模式中不使用。



注 只限于 16 引脚产品。

图 12-54 3 线串行 I/O (CSI00、CSI01 注) 从属接收时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 在此模式中不使用。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 ×	SOE00 ×

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- | |
|--|
| |
|--|

 : 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 12-55 从属接收的初始设定步骤

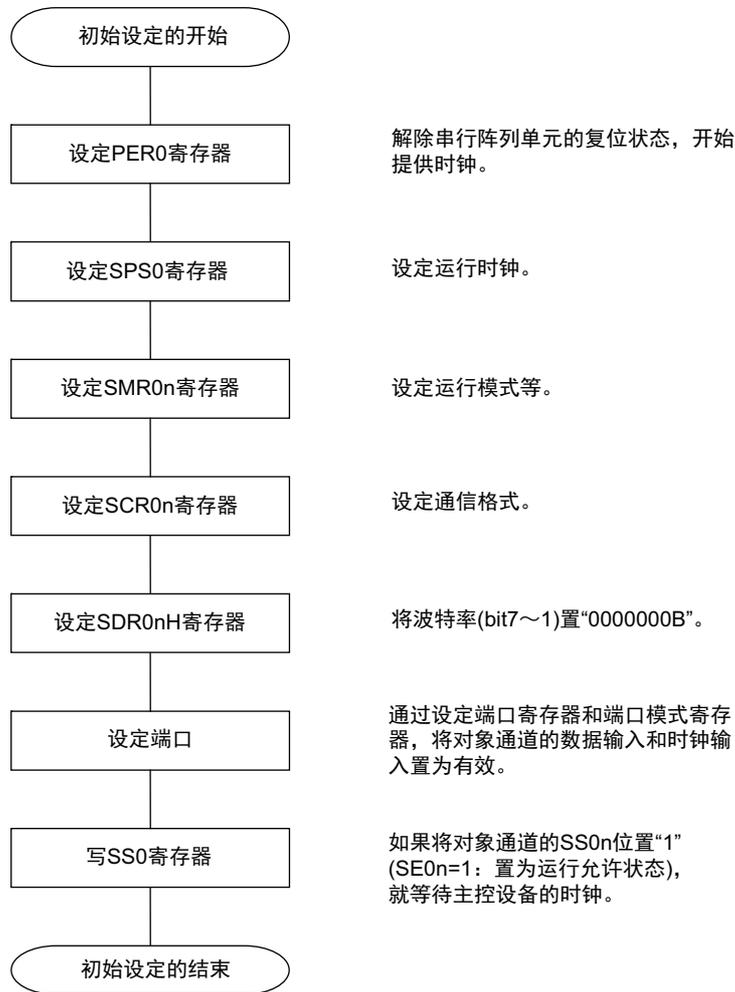


图 12-56 从属接收的中止步骤

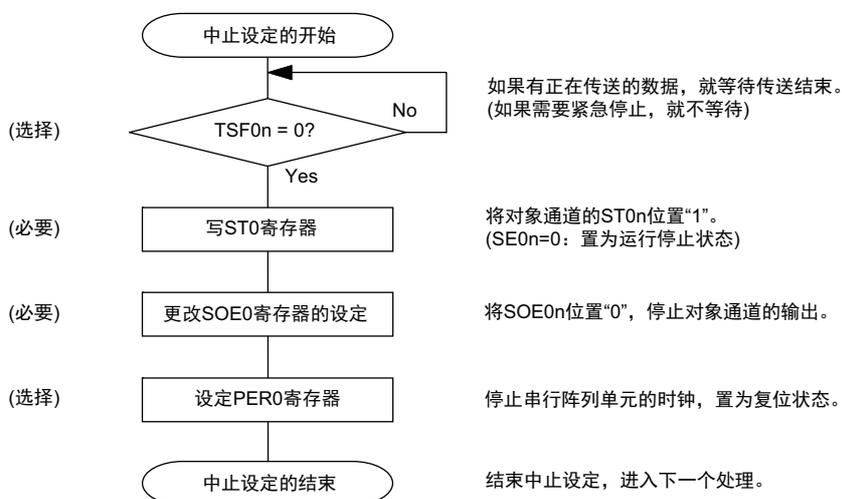
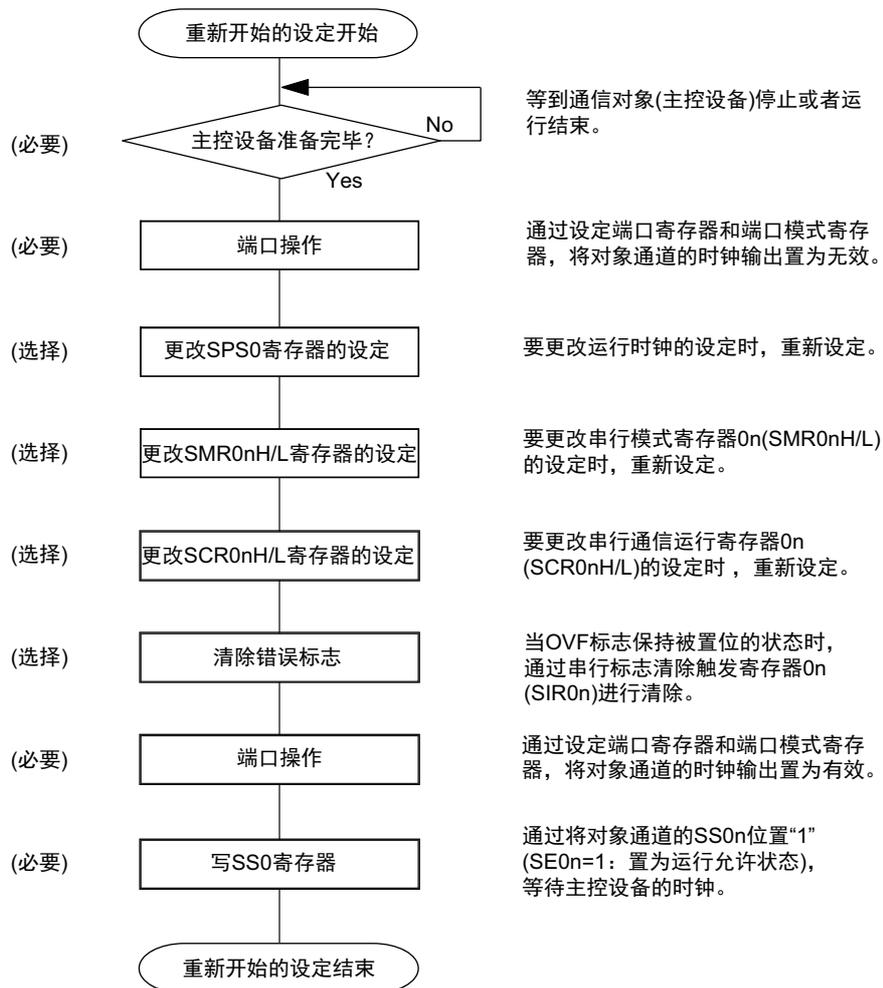


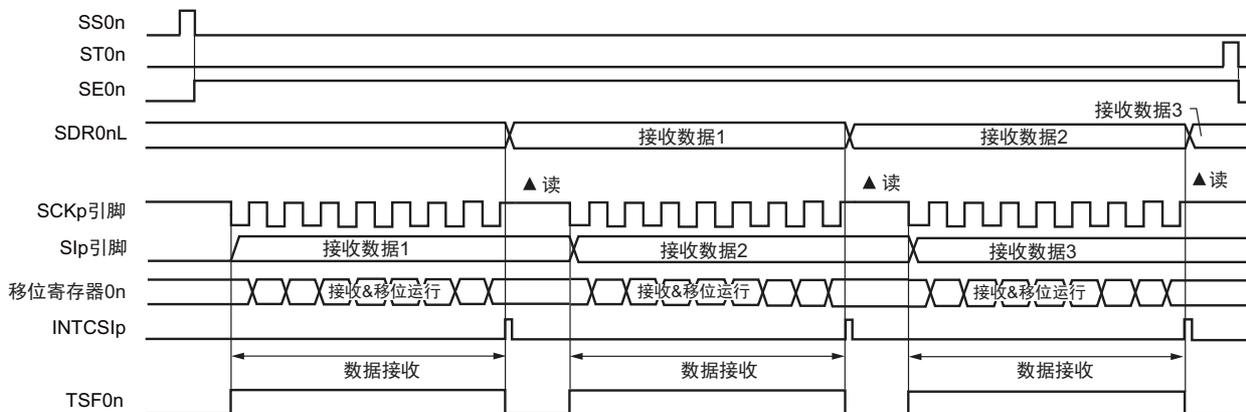
图 12-57 重新开始从属接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等待通信对象 (主控设备) 的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。

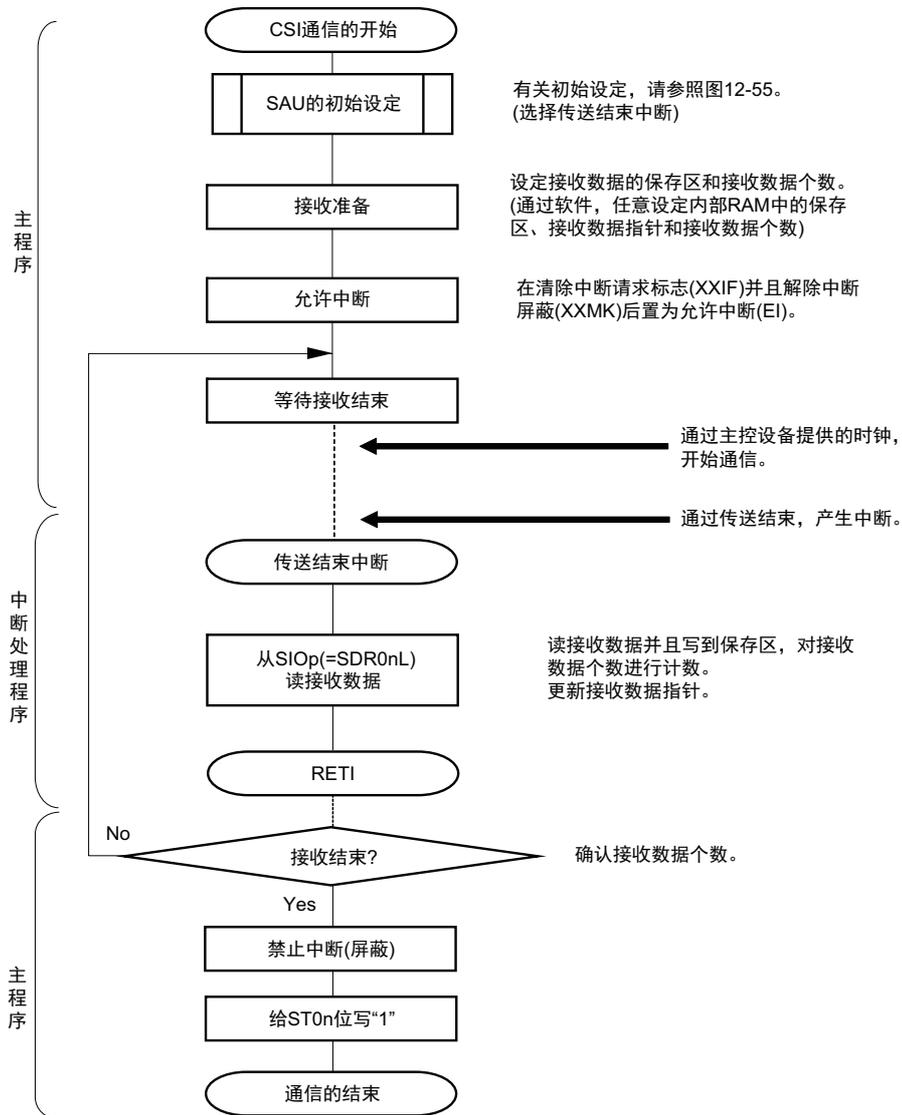
(3) 处理流程（单次接收模式）

图 12-58 从属接收（单次接收模式）的时序图（类型 1: DAP0n=0、CKP0n=0）



备注 n=0、1, p: CSI号 (p=00、01)

图 12-59 从属接收（单次接收模式）的流程图



12.5.6 从属发送和接收

从属发送和接收是在其他设备输入传送时钟的状态下 RL78/G10 和其他设备进行数据发送和接收的运行。

3 线串行 I/O	CSI00	CSI01 注 1
对象通道	SAU0 的通道 0	SAU0 的通道 1
使用的引脚	SCK00、SI00、SO00	SCK01、SO01
中断	INTCSI00	INTCSI01
	可选择传送结束中断（单次传送模式时）或者缓冲器空中断（连续传送模式时）。	
错误检测标志	只有溢出错误检测标志（OVF0n）。	
传送数据长度	7 位或者 8 位	
传送速率	Max. $f_{MCK}/6$ [Hz] 注 2、3	
数据相位	能通过 SCR0nH 寄存器的 DAP0n 位进行选择。 <ul style="list-style-type: none"> • DAP0n=0: 在串行时钟开始运行时，开始数据输出。 • DAP0n=1: 在串行时钟开始运行的半个时钟前，开始数据输出。 	
时钟相位	能通过 SCR0nH 寄存器的 CKP0n 位进行选择。 <ul style="list-style-type: none"> • CKP0n=0: 不反相 • CKP0n=1: 反相 	
数据方向	MSB 优先或者 LSB 优先	

注 1. 只限于 16 引脚产品。

2. 因为在内部对 SCK00、SCK01 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]。

3. 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 24 章 电特性”）的范围内使用。

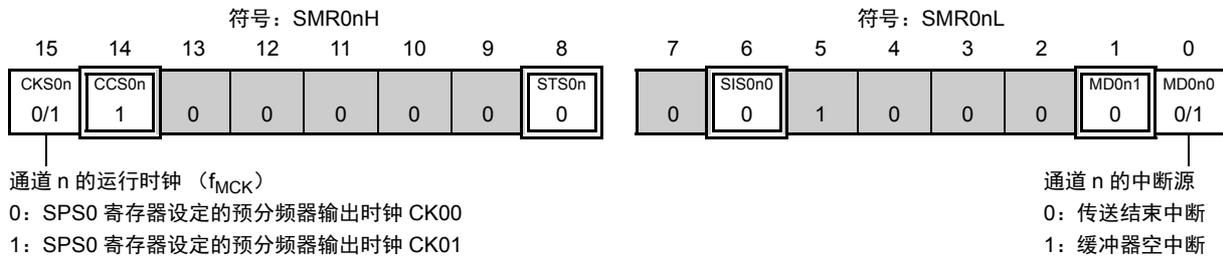
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. n=0、1

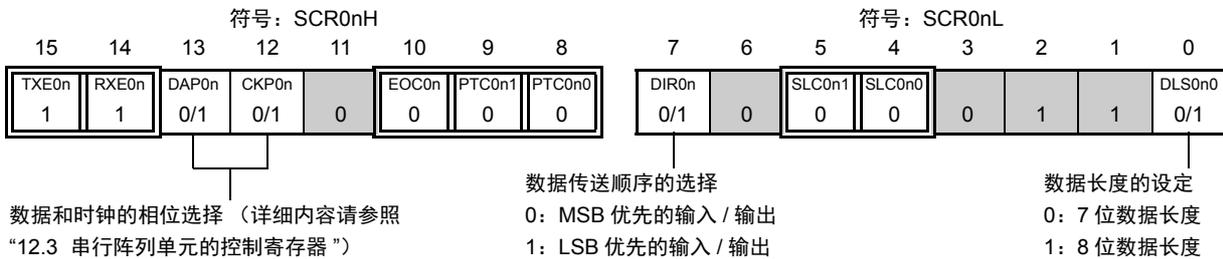
(1) 寄存器的设定

图 12-60 3 线串行 I/O (CSI00、CSI01 注) 从属发送和接收时的寄存器设定内容例子 (1/2)

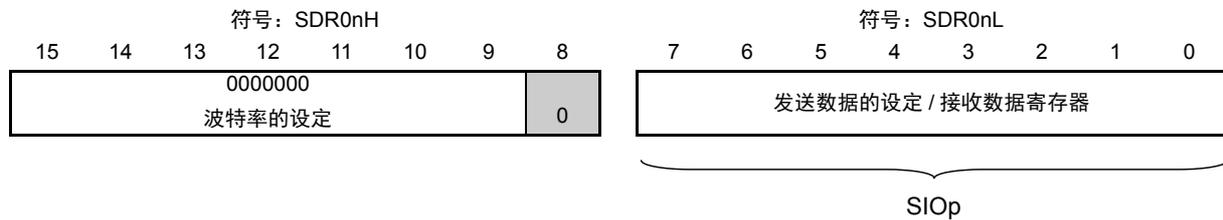
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



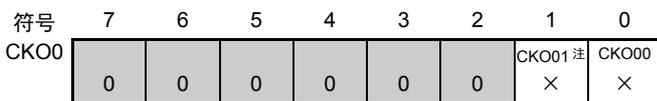
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



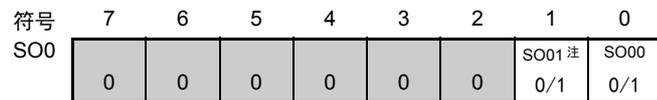
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 在此模式中不使用。



(e) 串行输出寄存器 0 (SO0) 只设定对象通道的位。



注 只限于 16 引脚产品。

图 12-60 3 线串行 I/O (CSI00、CSI01 注) 从属发送和接收时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 注 0/1	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 0/1

注 只限于 16 引脚产品。

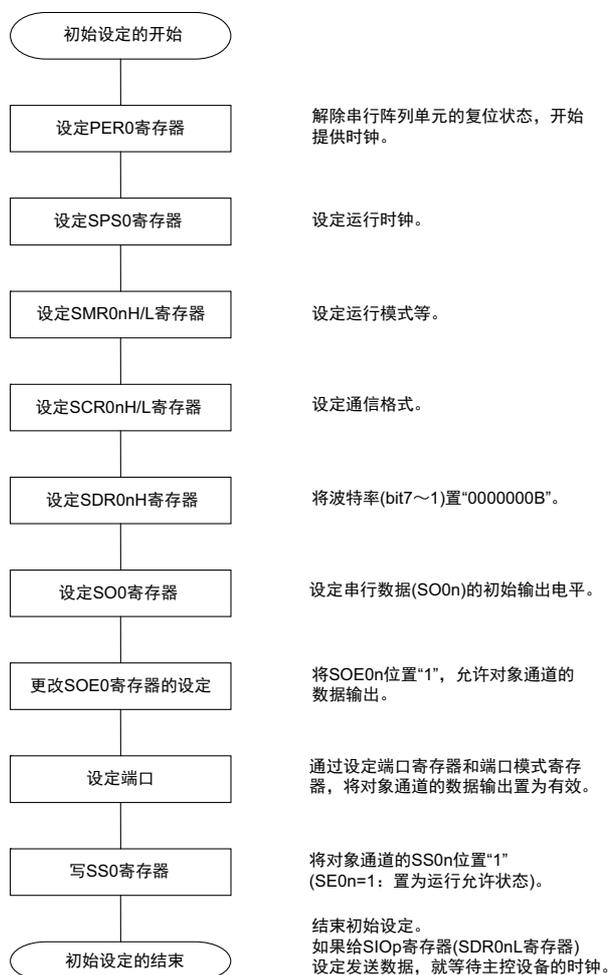
注意 在主控设备开始输出时钟前，必须给 SIOp 寄存器设定发送数据。

备注 1. n=0、1, p: CSI 号 (p=00、01)

- : 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 12-61 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 12-62 从属发送和接收的中止步骤

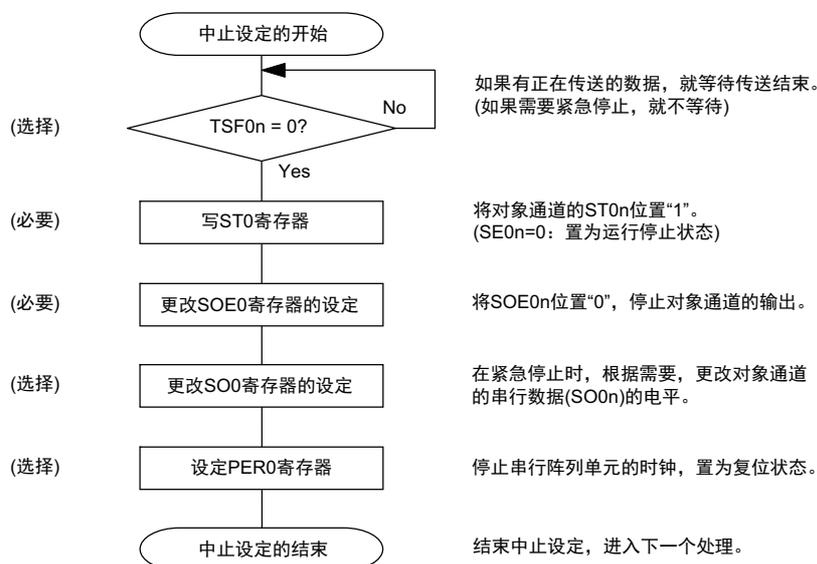
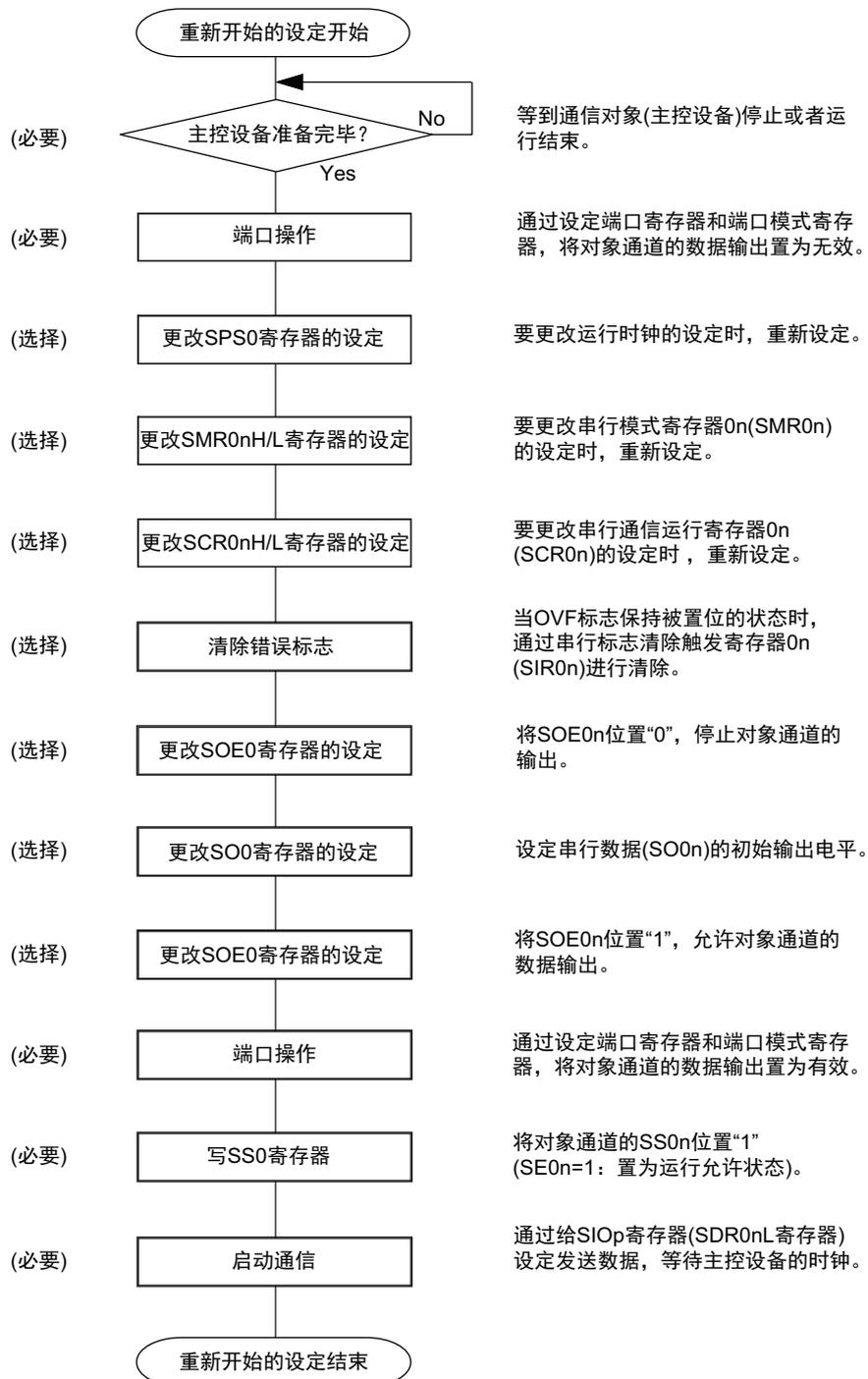


图 12-63 重新开始从属发送和接收的设定步骤

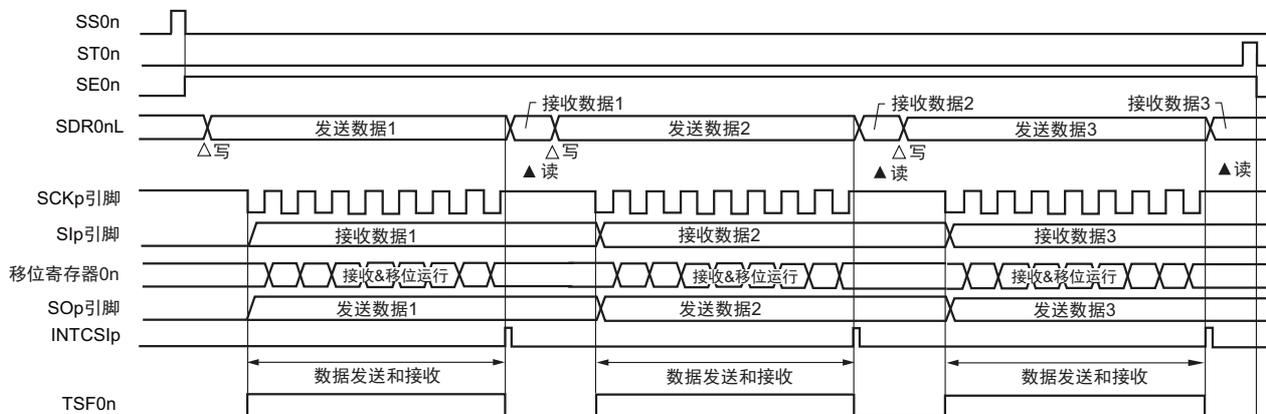


注意 1. 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等待通信对象 (主控设备) 的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。

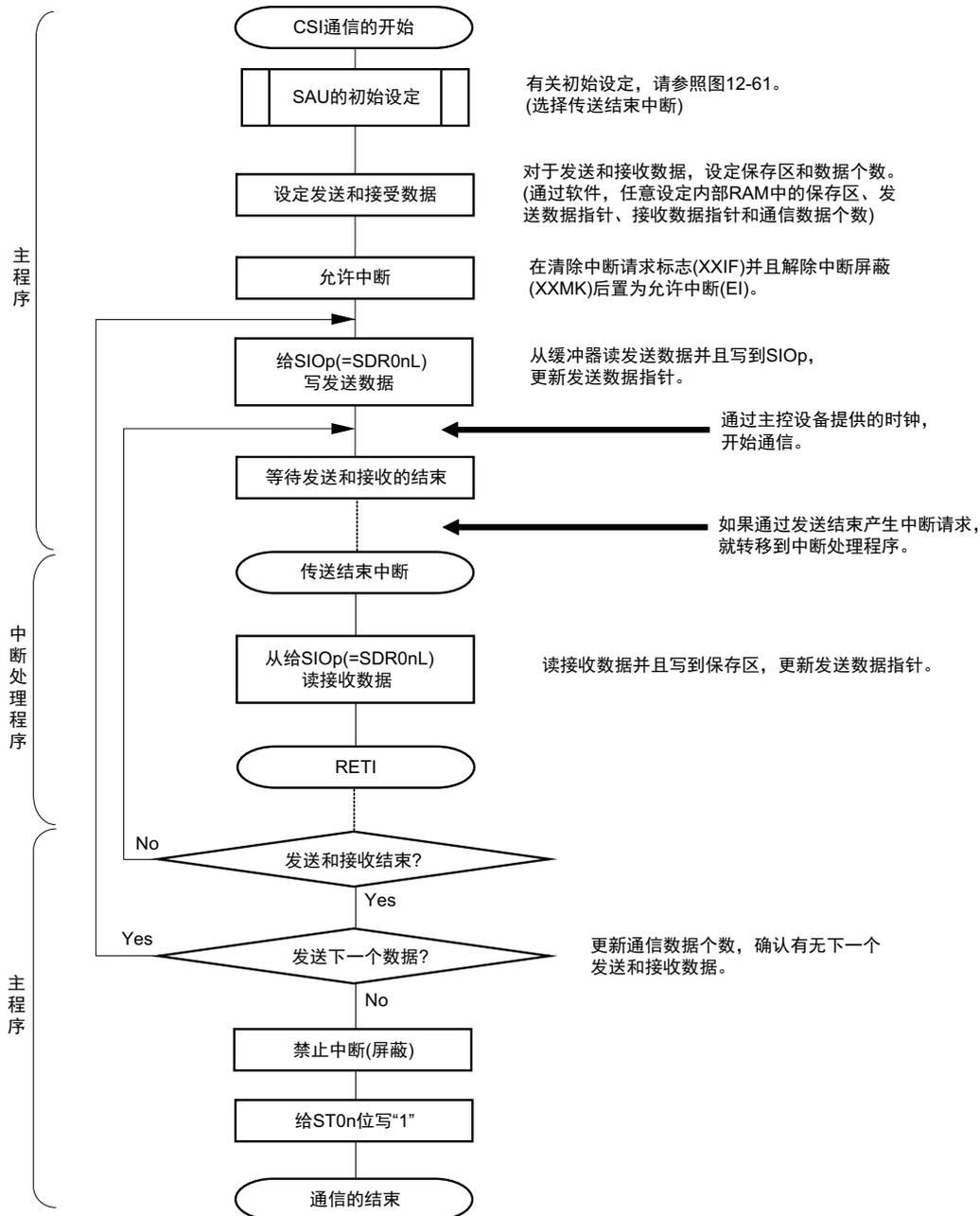
(3) 处理流程（单次发送和接收模式）

图 12-64 从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAP0n=0、CKP0n=0）



备注 n=0、1, p: CSI号 (p=00、01)

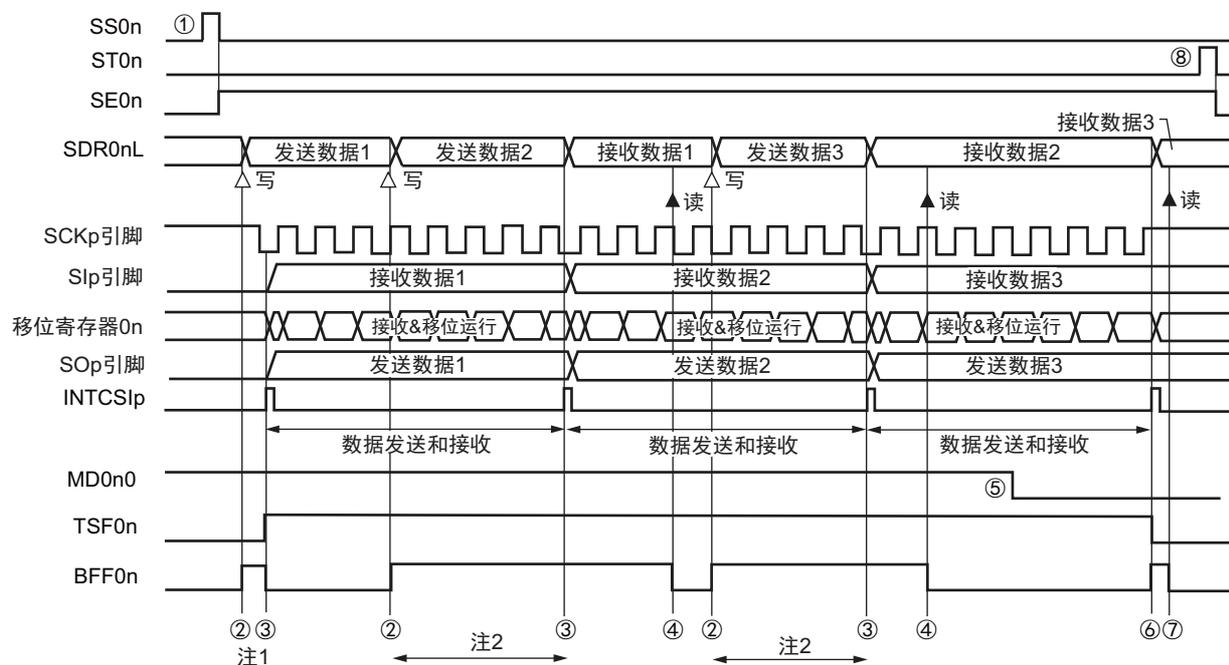
图 12-65 从属发送和接收（单次发送和接收模式）的流程图



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 12-66 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）



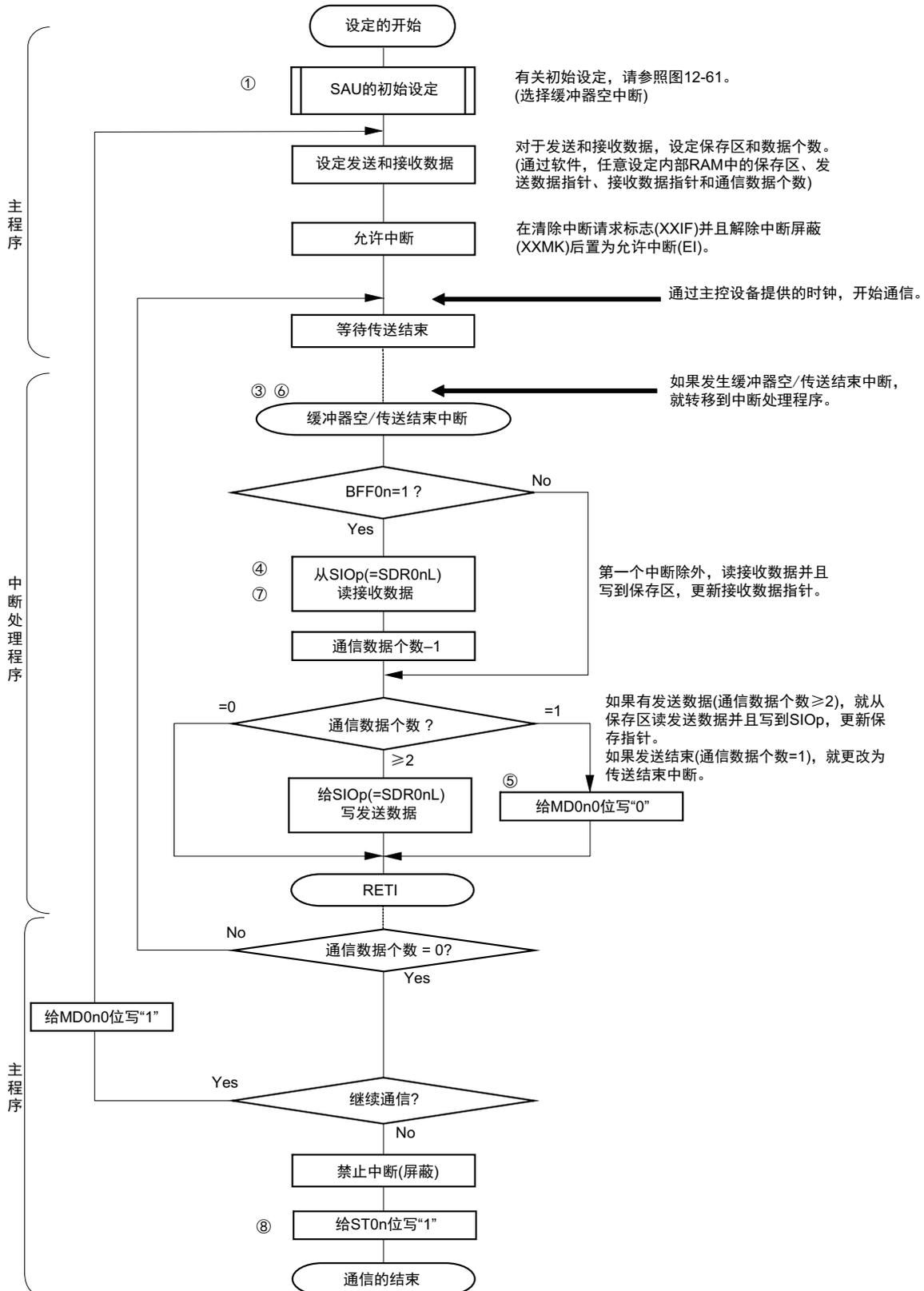
- 注 1. 如果在串行状态寄存器 0n（SSR0n）的 BFF0n 位为“1”期间（有效数据保存在串行数据寄存器 0n（SDR0nL）时）给 SDR0nL 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDR0nL 寄存器，就能读发送数据。此时，不影响传送运行。

注意 即使在运行中也能改写串行模式寄存器 0n（SMR0n）的 MD0n0 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 12-67 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. n=0、1，p: CSI 号（p=00、01）

图 12-67 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 图中的①~⑧对应“图 12-66 从属发送和接收（连续发送和接收模式）的时序图”中的①~⑧。

12.5.7 传送时钟频率的计算

3 线串行 I/O (CSI00、CSI01) 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

$$(\text{传送时钟频率}) = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}} \text{ 频率)} \div (\text{SDR0nH}[7:1] + 1) \div 2[\text{Hz}]$$

(2) 从属设备

$$(\text{传送时钟频率}) = \{ \text{主控设备提供的串行时钟 (SCK) 频率} \} \text{注} [\text{Hz}]$$

注 容许的最大传送时钟频率为 $f_{\text{MCK}}/6$ 。

备注 因为 SDR0nH[7:1] 的值为串行数据寄存器 0nH (SDR0nH) 的 bit7 ~ 1 的值 (0000000B ~ 1111111B)，所以为 0 ~ 127。

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0n (SMR0nH) 的 bit7 (CKS0n)。

表 12-2 3 线串行 I/O 运行时钟的选择

SMR0n 寄存器	SPS0 寄存器								运行时钟 (f_{MCK}) 注	
	CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00	$f_{CLK}=20\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	f_{CLK}	20MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22kHz
X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610Hz	
1	0	0	0	0	X	X	X	X	f_{CLK}	20MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22kHz
1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610Hz	

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 0（ST0）=03H）后进行更改。

备注 1. X: 忽略

2. n=0、1

12.5.8 在 3 线串行 I/O（CSI00、CSI01^注）通信过程中发生错误时的处理步骤

在 3 线串行 I/O（CSI00、CSI01^注）通信过程中发生的错误时的处理步骤如图 12-68 所示。

图 12-68 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 0n（SDR0nL）。	SSR0n 寄存器的 BFF0n 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 0n（SSR0n）。		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 0n（SIR0n）写“1”。	清除错误标志。	通过将 SSR0n 寄存器的读取值直接写到 SIR0n 寄存器，只能清除读操作时的错误。

注 只限于 16 引脚产品。

备注 n=0、1

12.6 UART (UART0) 通信的运行

这是通过串行发送数据 (TxD) 和串行接收数据 (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位和8位的数据长度注
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定 (选择是否取反电平)
- 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

能通过设定 ISC 寄存器, 将 UART0 的 RxD0 引脚的输入信号用作外部中断输入或者定时器阵列单元的定时器输入。能通过使用定时器阵列单元的输入脉冲间隔测量模式测量通信方的波特率宽度, 实现 UART0 的波特率校正。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		—

注意 选择 UART 后, 偶数通道和奇数通道只能分别用于 UART 的发送方和 UART 的接收方。

UART 有以下 2 种通信运行:

- UART 发送 (参照 12.6.1)
- UART 接收 (参照 12.6.2)

12.6.1 UART 发送

UART 发送是 RL78/G10 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 位或者 8 位
传送速率注	Max. $f_{MCK}/6[\text{bps}]$ ($\text{SDR0nH}[7:1] \geq 2$)、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$
数据相位	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> • 无奇偶校验位 • 附加零校验 • 附加偶校验 • 附加奇校验
停止位	可选择以下内容： <ul style="list-style-type: none"> • 附加 1 位 • 附加 2 位
数据方向	MSB 优先或者 LSB 优先

注 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 24 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

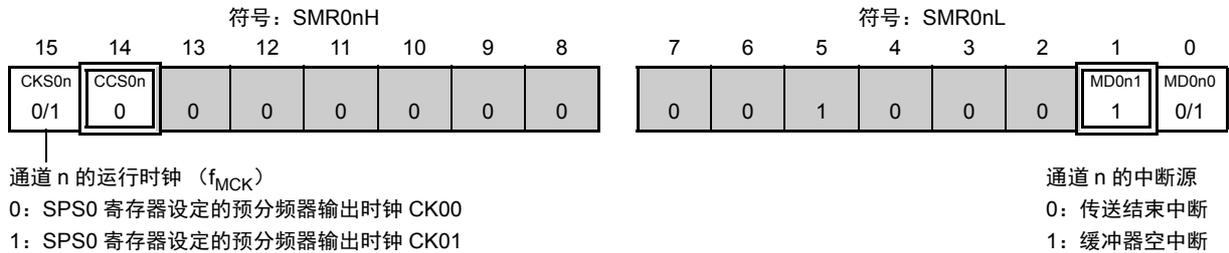
f_{CLK} : 系统时钟频率

2. n: 通道号 (n=0)

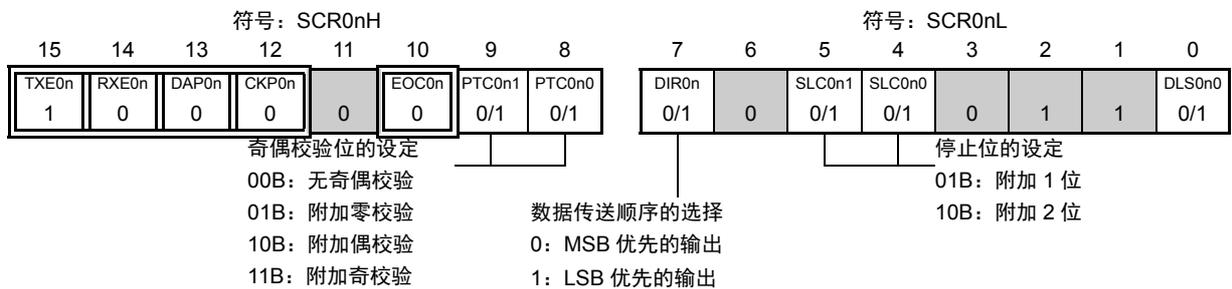
(1) 寄存器的设定

图 12-69 UART (UART0) 发送时的寄存器设定内容例子 (1/2)

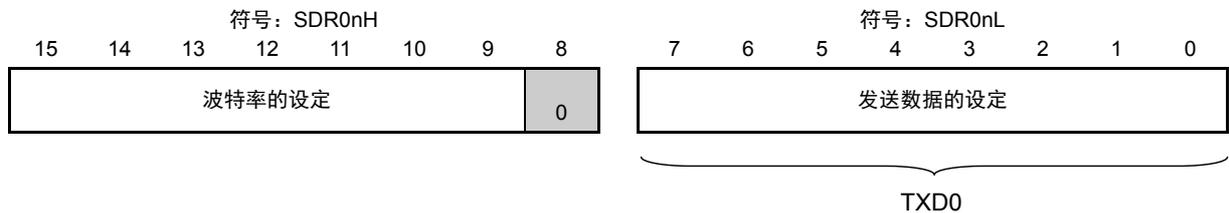
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行输出电平寄存器 0 (SOL0) 只设定对象通道的位。



备注 1. n=0

- 2. □ : 在 UART 发送模式中为固定设定。■ : 不能设定 (设定初始值)。
 0/1 : 根据用户的用途置“0”或者“1”。

图 12-69 UART (UART0) 发送时的寄存器设定内容例子 (2/2)

(e) 串行时钟输出寄存器 0 (CKO0) 只设定对象通道的位。

符号	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 ×	CKO00 ×

(f) 串行输出寄存器 0 (SO0) 只设定对象通道的位。

符号	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 ×	SO00 0/1

0: 串行数据输出值为“0”

1: 串行数据输出值为“1”

(g) 串行输出允许寄存器 0 (SOE0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 ×	SOE00 0/1 注

(h) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

注 当对应通道的 SOL00 位为“0”时，必须置“1”；当对应通道的 SOL00 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. n=0

2. : 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 12-70 UART 发送的初始设定步骤

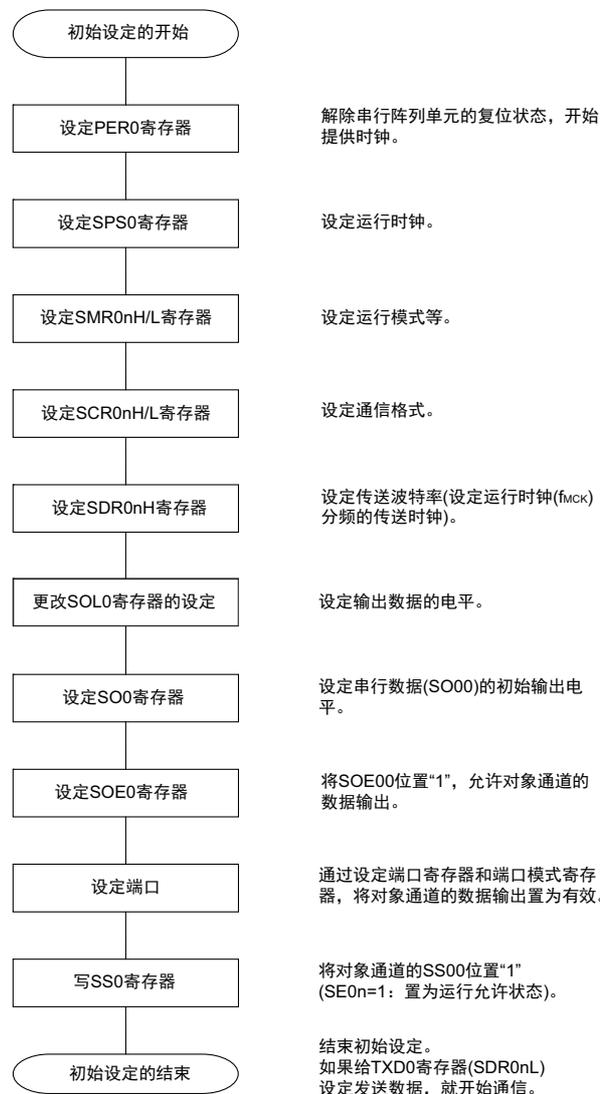


图 12-71 UART 发送的中止步骤

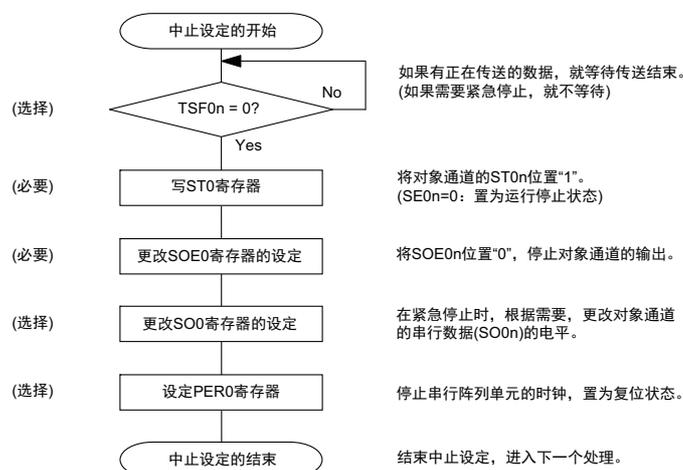
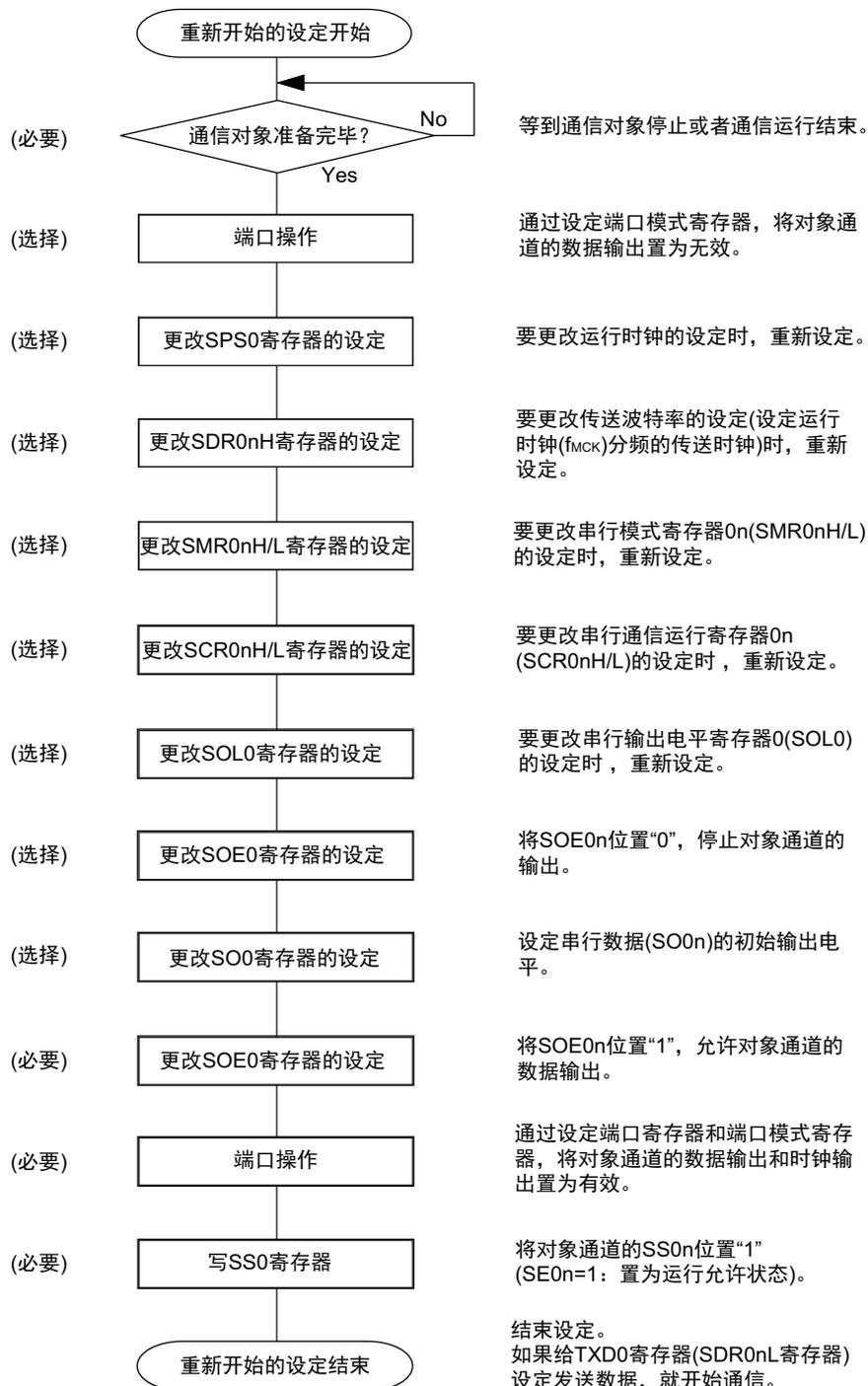


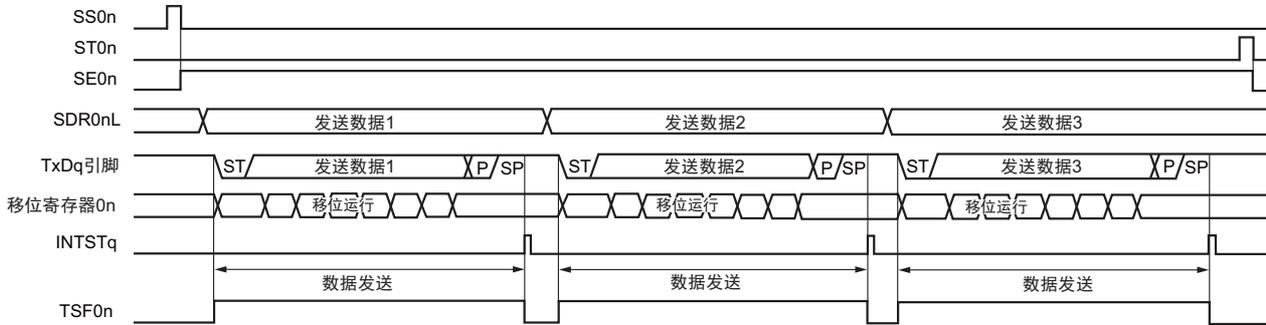
图 12-72 重新开始 UART 发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等待通信对象的停止或者通信结束后进行初始设定而不是进行重新开始设定。

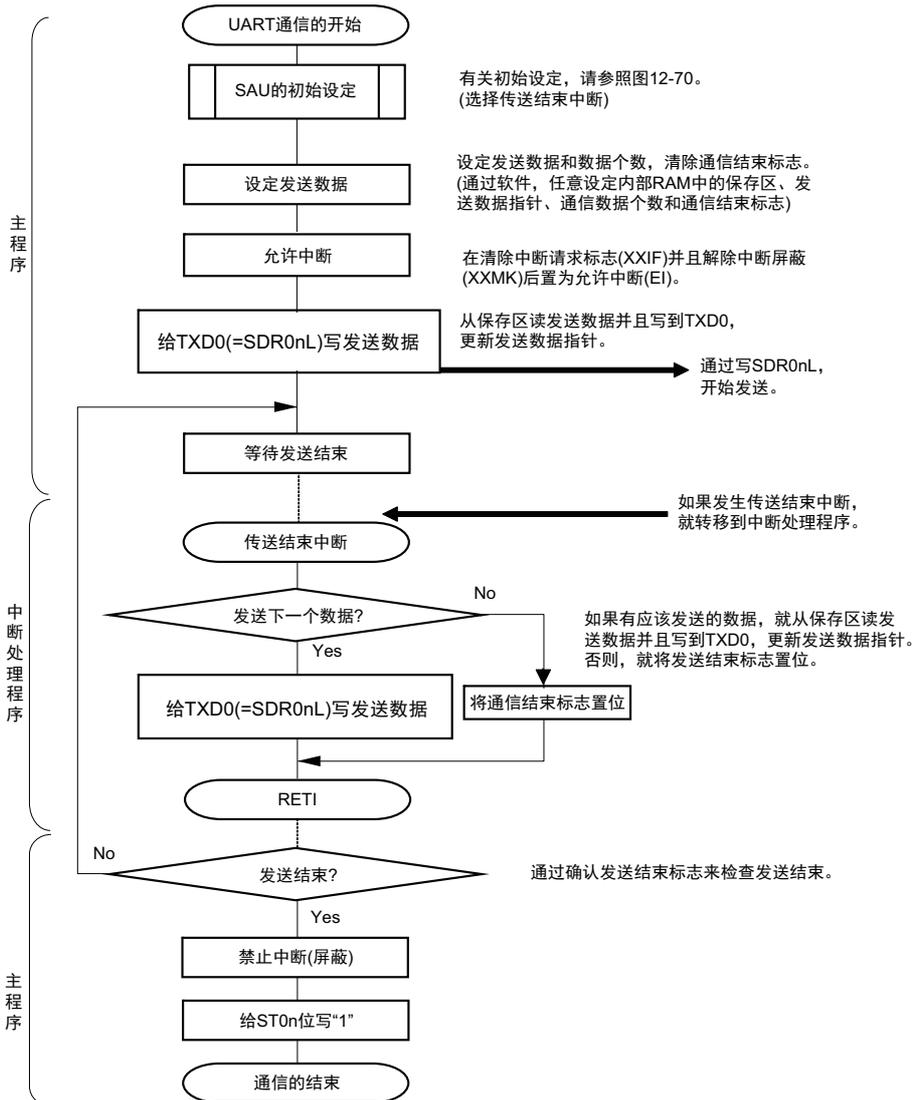
(3) 处理流程（单次发送模式）

图 12-73 UART 发送（单次发送模式）的时序图



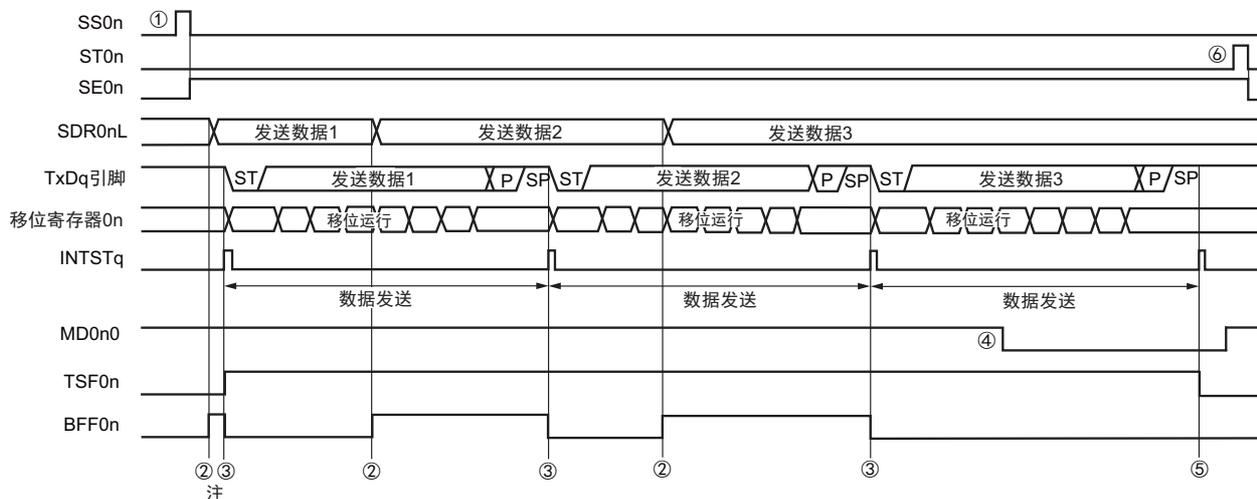
备注 q: UART 号 (q=0), n=0

图 12-74 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 12-75 UART 发送（连续发送模式）的时序图

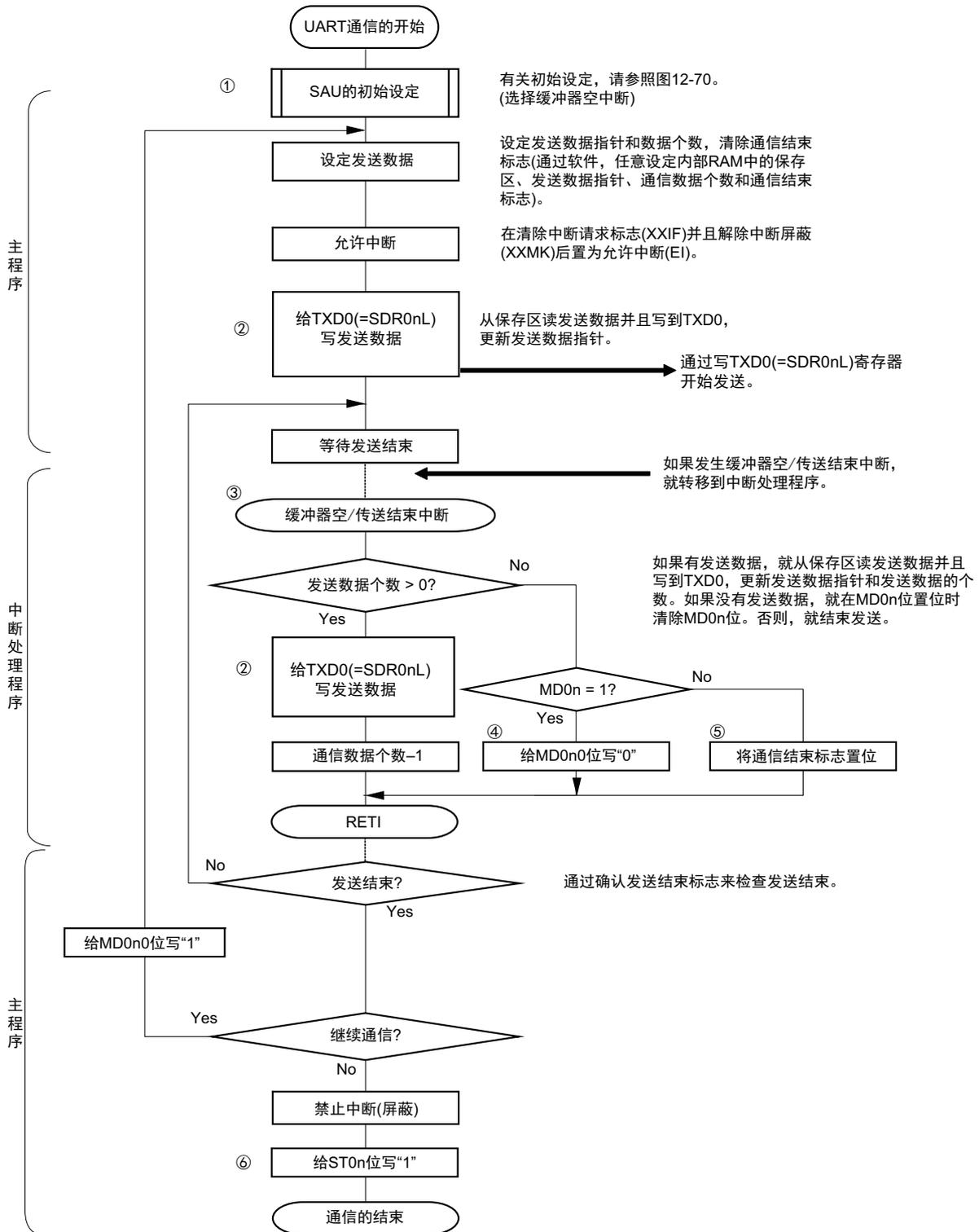


注 如果在串行状态寄存器 0n（SSR0n）的 BFF0n 位为“1”期间（有效数据保存在串行数据寄存器 0n（SDR0nL）时）给 SDR0nL 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 0n（SMR0nL）的 MD0n0 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 q: UART 号 (q=0), n=0

图 12-76 UART 发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 12-75 UART 发送（连续发送模式）的时序图”中的①~⑥。

12.6.2 UART 接收

UART 接收是 RL78/G10 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"> • 帧错误检测标志（FEF0n） • 奇偶校验错误检测标志（PEF0n） • 溢出错误检测标志（OVF0n）
传送数据长度	7 位或者 8 位
传送速率注	Max. $f_{MCK}/6[\text{bps}]$ ($\text{SDR0nH}[7:1] \geq 2$)、Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]
数据相位	不反相输出（默认值：高电平）。 反相输出（默认值：低电平）。
奇偶校验位	可选择以下内容： <ul style="list-style-type: none"> • 无奇偶校验位（无奇偶校验） • 不判断奇偶校验（零校验） • 偶校验 • 奇校验
停止位	附加 1 位。
数据方向	MSB 优先或者 LSB 优先

注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 24 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

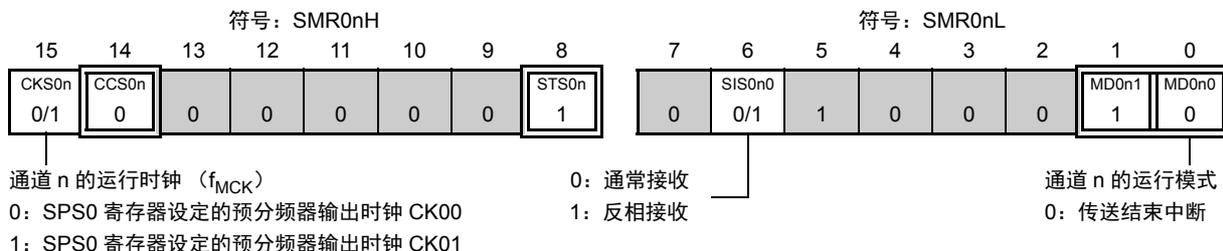
f_{CLK} : 系统时钟频率

2. n: 通道号 (n=1)

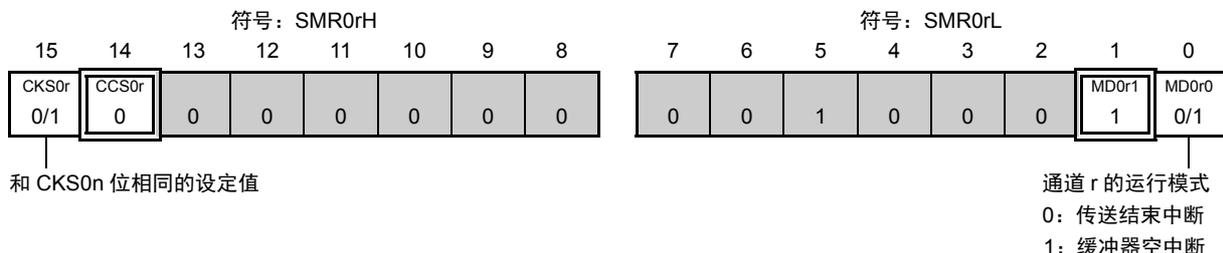
(1) 寄存器的设定

图 12-77 UART (UART0) 接收时的寄存器设定内容例子 (1/2)

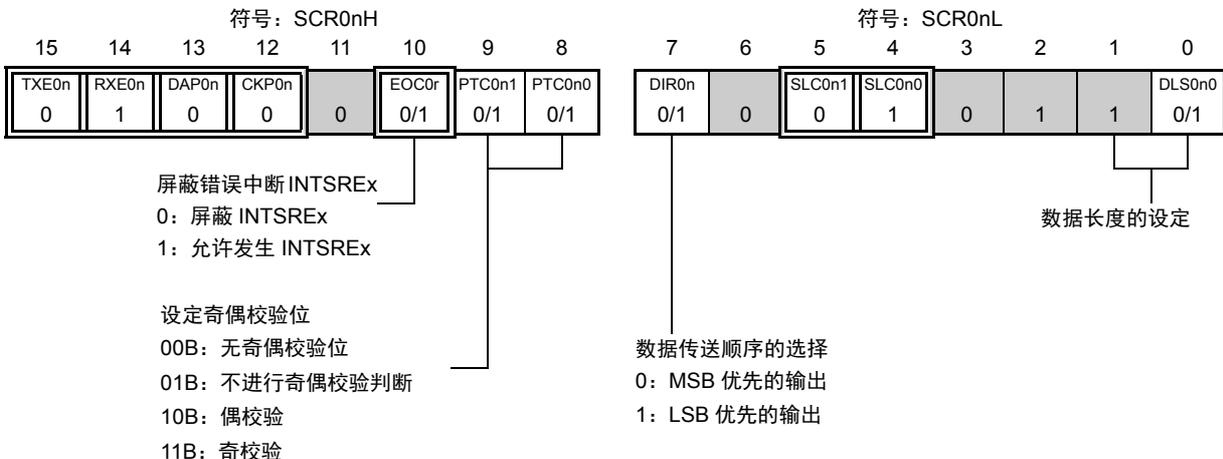
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



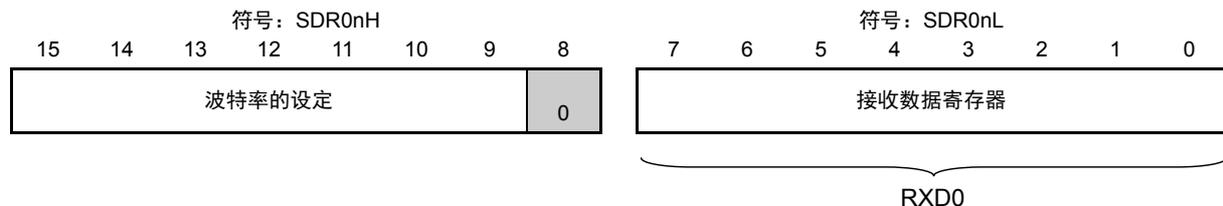
(b) 串行模式寄存器 0r (SMR0rH、SMR0rL)



(c) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



(d) 串行数据寄存器 0n (SDR0nH、SDR0nL)



注意 在 UART 接收时, 还必须设定与通道 n 成对的通道 r 的 SMR0r 寄存器。

备注 1. n: 通道号 (n=1), r: 通道号 (r=n-1), 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)。
0/1 : 根据用户的用途置“0”或者“1”。

图 12-77 UART (UART0) 接收时的寄存器设定内容例子 (2/2)

(e) 串行时钟输出寄存器 0 (CKO0) 在此模式中不使用。

符号	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 ×	CKO00 ×

(f) 串行输出寄存器 0 (SO0) 在此模式中不使用。

符号	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 ×	SO00 ×

(g) 串行输出允许寄存器 0 (SOE0) 在此模式中不使用。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 ×	SOE00 ×

(h) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 ×

注意 在 UART 接收时, 还必须设定与通道 n 成对的通道 r 的 SMR0r 寄存器。

备注 1. n: 通道号 (n=1), r: 通道号 (r=n-1), q: UART 号 (q=0)

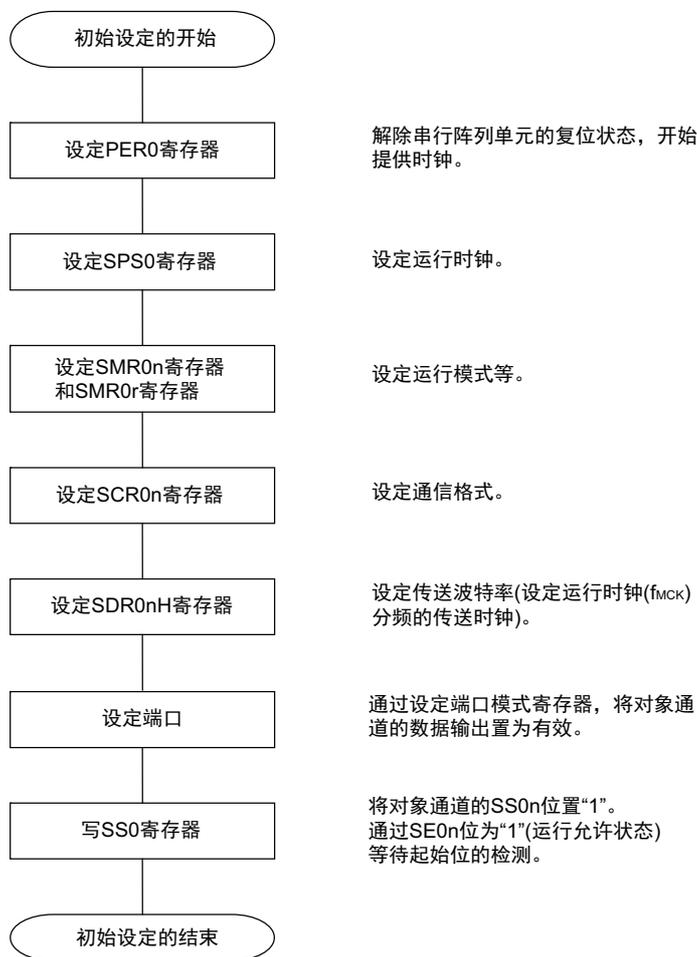
2. : 不能设定 (设定初始值)。

× : 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 12-78 UART 接收的初始设定步骤



注意 必须在将 SCR0n 寄存器的 RXE0n 位置 “1” 后至少经过 4 个 f_{MCK} 时钟，然后将 SS0n 位置 “1”。

图 12-79 UART 接收的中止步骤

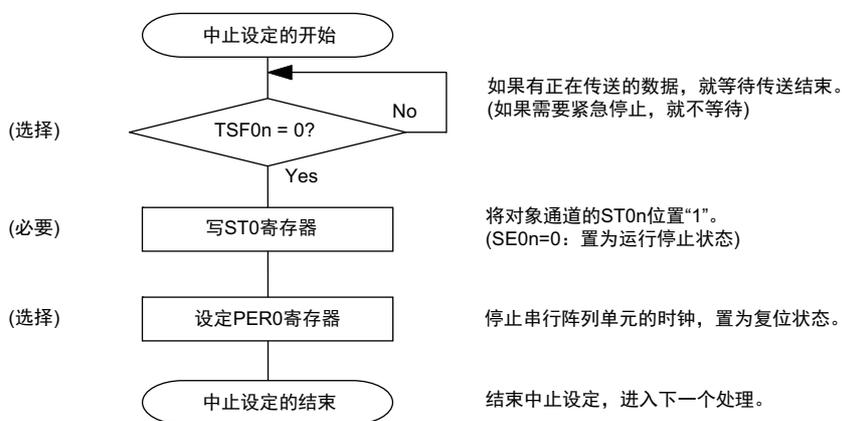
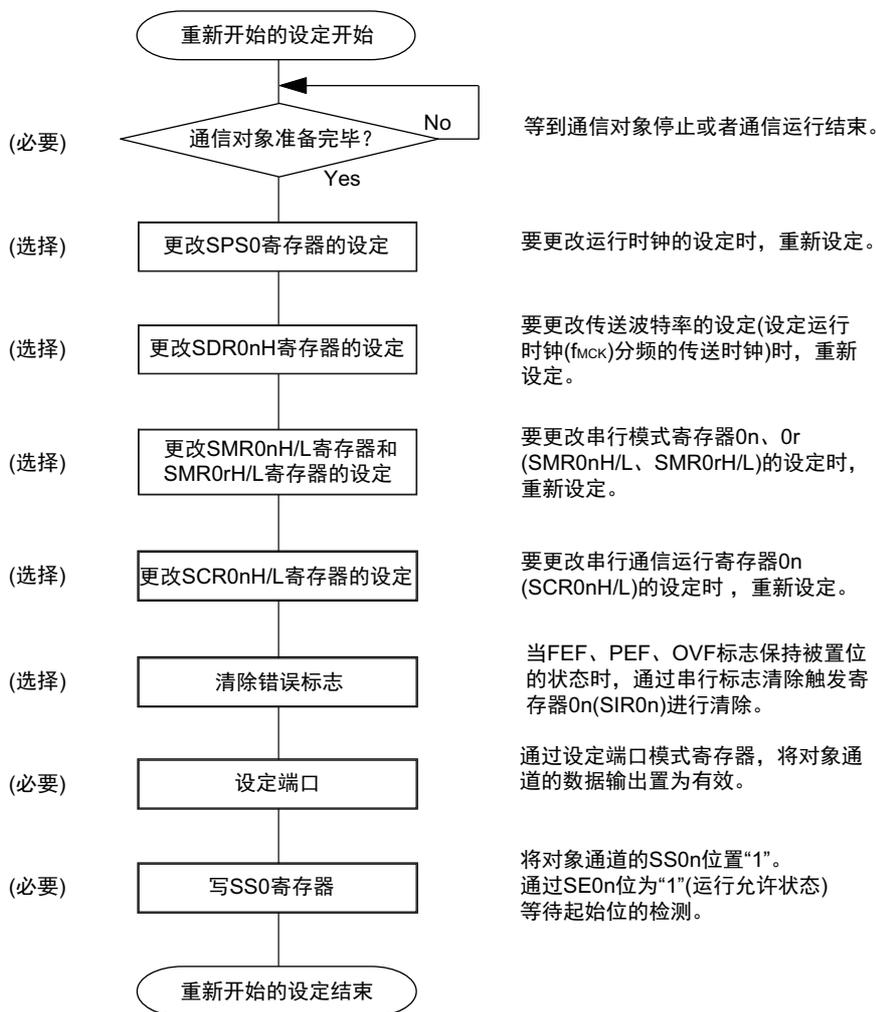


图 12-80 重新开始 UART 接收的设定步骤

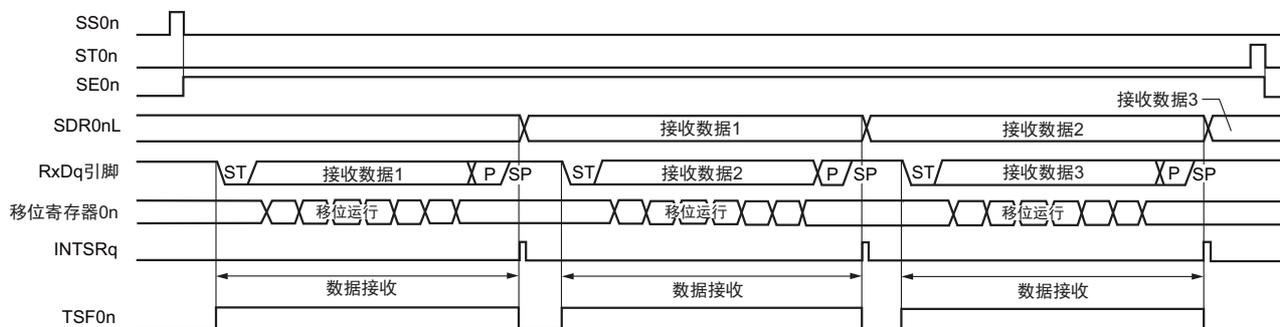


注意 必须在将 SCR0n 寄存器的 RXE0n 位置“1”后至少经过 4 个 f_{MCK} 时钟，然后将 SS0n 位置“1”。

备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等待通信对象的停止或者通信结束后进行初始设定而不是进行重新开始设定。

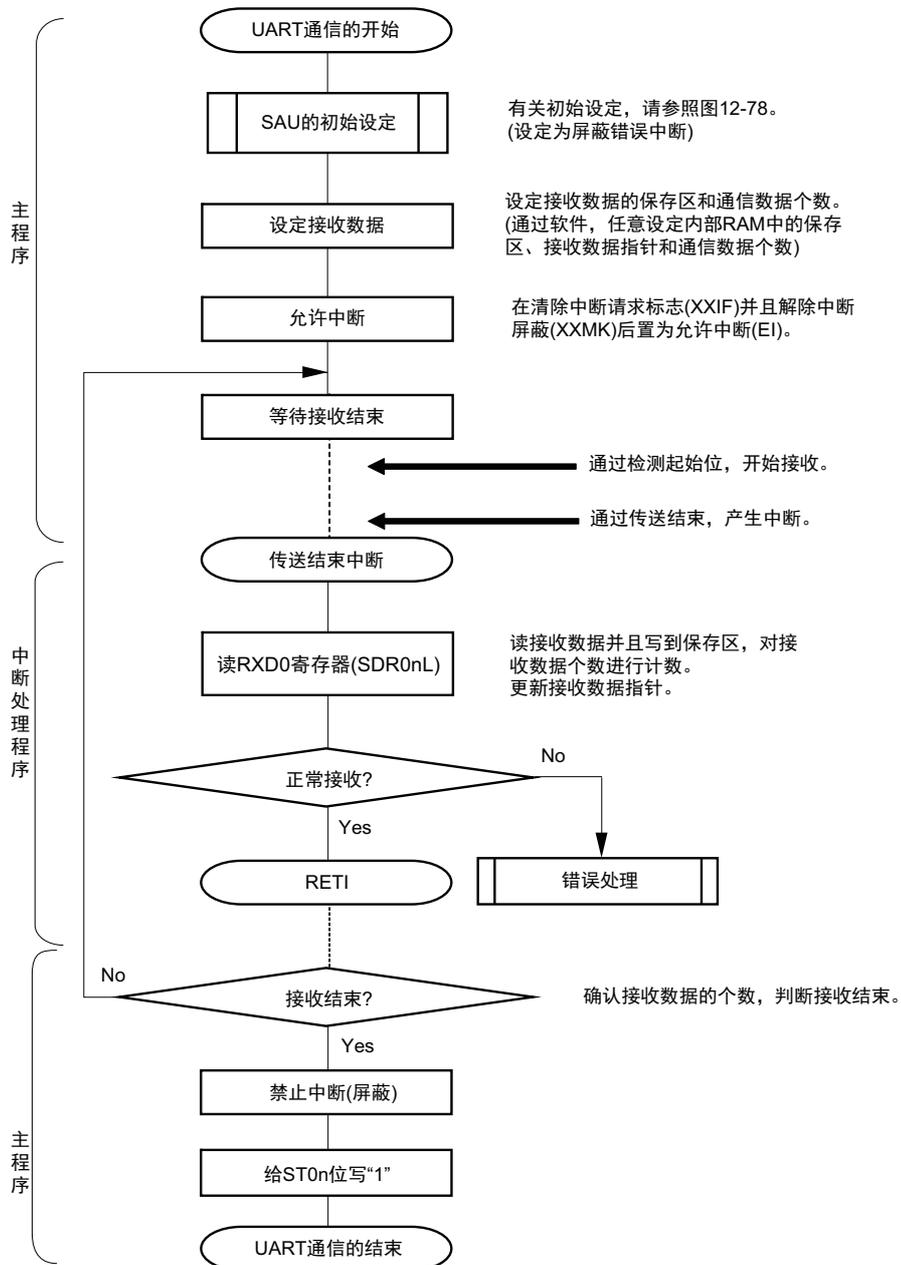
(3) 处理流程

图 12-81 UART 接收的时序图



备注 n: 通道号 (n=1), r: 通道号 (r=n-1), q: UART 号 (q=0)

图 12-82 UART 接收的流程图



12.6.3 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:

$$(\text{波特率}) = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}}) \text{ 频率} \} \div (\text{SDR0nH}[7:1] + 1) \div 2 [\text{bps}]$$

注意 禁止设定串行数据寄存器 0n (SDR0nH) 的 SDR0nH[7:1]=(0000000B、0000001B)。

备注 1. 因为在使用 UART 时 SDR0nH[7:1] 的值为 SDR0nH 寄存器的 bit7 ~ 1 的值 (0000010B ~ 1111111B), 所以为 2 ~ 127。

2. n=0、1

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0n (SMR0n) 的 bit15 (CKS0n 位)。

表 12-3 UART 运行时钟的选择

SMR0n 寄存器	SPS0 寄存器								运行时钟 (f_{MCK}) 注	
	CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00	$f_{CLK}=20\text{MHz}$ 运行时
0	X	X	X	X	0	0	0	0	f_{CLK}	20MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22kHz
X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610Hz	
1	0	0	0	0	X	X	X	X	f_{CLK}	20MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22kHz
1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610Hz	

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 0（ST0）=03H）后进行更改。

备注 1. X: 忽略

2. n=0、1

(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。

$$(\text{波特率误差}) = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100 [\%]$$

$f_{\text{CLK}}=20\text{MHz}$ 时的 UART 波特率的设定例子如下所示。

UART 波特率 (目标波特率)	$f_{\text{CLK}}=20\text{MHz}$			
	运行时钟 (f_{MCK})	SDR0nH[7:1]	波特率的计算值	与目标波特率的误差
300bps	$f_{\text{CLK}}/2^9$	64	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	64	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	64	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	64	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	64	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	64	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	64	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	39	31250.0bps	$\pm 0.0\%$
38400bps	$f_{\text{CLK}}/2^2$	64	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	64	76923.1bps	+0.16%
153600bps	f_{CLK}	64	153846bps	+0.16%
312500bps	f_{CLK}	31	312500bps	$\pm 0.0\%$

(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“12.6.3(1) 波特率的计算式”)

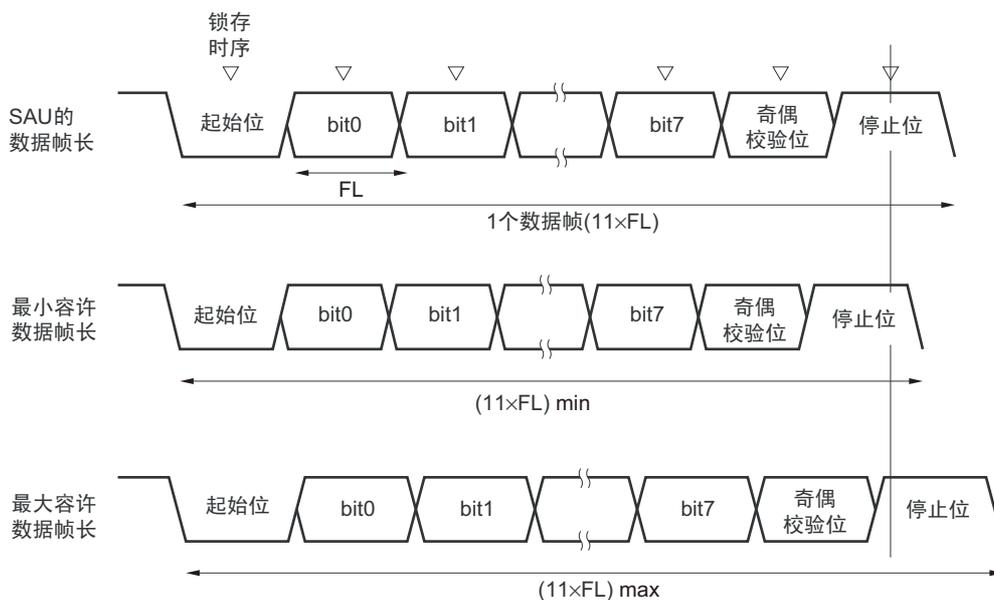
K : SDR0nH[7:1]+1

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 n=1

图 12-83 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 12-83 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 0nH (SDR0nH) 的 bit7 ~ 1 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。

12.6.4 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 12-84 和图 12-85 所示。

图 12-84 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 0n (SDR0nL)。	SSR0n 寄存器的 BFF0n 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出现溢出错误。
读串行状态寄存器 0n (SSR0n)。		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 0n (SIR0n) 写“1”。	清除错误标志。	通过将 SSR0n 寄存器的读取值直接写到 SIR0n 寄存器，只能清除读操作时的错误。

图 12-85 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 0n (SDR0nL)。	SSR0n 寄存器的 BFF0n 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出现溢出错误。
读串行状态寄存器 0n (SSR0n)。		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 0n (SIR0n)。	清除错误标志。	通过将 SSR0n 寄存器的读取值直接写到 SIR0n 寄存器，只能清除读操作时的错误。
将串行通道停止寄存器 0 (ST0) 的 ST0n 位置“1”。	串行通道允许状态寄存器 0 (SE0) 的 SE0n 位为“0”并且通道 n 为停止运行状态。	
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 0 (SS0) 的 SS0n 位置“1”。	串行通道允许状态寄存器 0 (SE0) 的 SE0n 位为“1”并且通道 n 为可运行状态。	

备注 n=0、1

12.7 简易 I²C (IIC00) 通信的运行

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

对于开始条件和停止条件，必须遵守 I²C 总线的总线特性，通过软件操作控制寄存器。

[数据的发送和接收]

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

[中断功能]

- 传送结束中断

[错误检测标志]

- 溢出错误
- ACK 错误

※ [简易 I²C 不支持的功能]

- 从属发送、从属接收
- 多主控功能 (仲裁失败检测功能)
- 等待检测功能

注 在接收最后的数据时，如果给 SOE0 寄存器的 SOE00 位写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“12.7.3(2) 处理流程”。

备注 全功能 I²C 请参照“第 13 章 串行接口 IICA” (只限于 16 引脚产品)。

支持简易 I²C (IIC00) 的通道为 SAU0 的通道 0。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		—

简易 I²C (IIC00) 有如下 4 种通信运行：

- 地址字段发送 (参照 12.7.1)
- 数据发送 (参照 12.7.2)
- 数据接收 (参照 12.7.3)
- 停止条件的产生 (参照 12.7.4)

12.7.1 地址字段发送

地址字段发送是为了特别指定传送对象（从属设备）而在 I²C 通信时最先进行的发送运行。在产生开始条件后，将地址（7 位）和传送方向（1 位）作为 1 帧进行发送。

简易 I ² C	IIC00
对象通道	SAU0 的通道 0
使用的引脚	SCL00、SDA00 注
中断	INTIIC00
	只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	ACK 错误检测标志（PEF0n）
传送数据长度	8 位（将高 7 位作为地址并且将低 1 位作为 R/W 控制进行发送）
传送速率注 2	Max. $f_{MCK}/4$ [Hz]（ $SDR0nH[7:1] \geq 1$ ） f_{MCK} : 对象通道的运行时钟频率 但是，必须在 I ² C 的各模式中满足以下条件： <ul style="list-style-type: none"> • Max. 400kHz（快速模式） • Max. 100kHz（标准模式）
数据电平	不反相输出（默认值：高电平）。
奇偶校验位	无奇偶校验位。
停止位	附加 1 位（用于 ACK 接收）
数据方向	MSB 优先

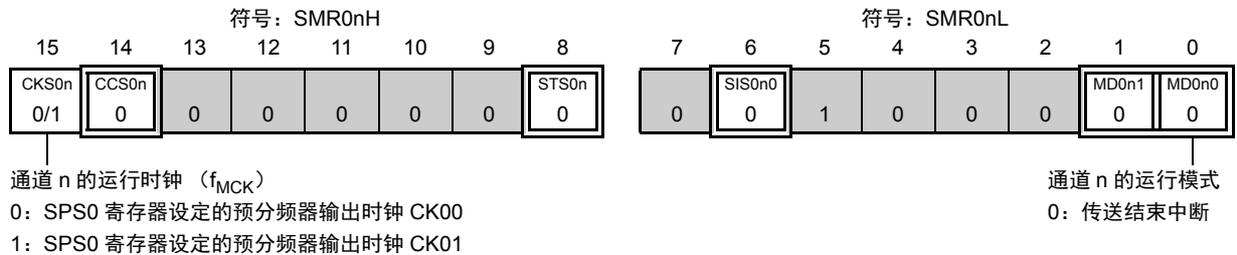
注 要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器 0（POM0）设定 N 沟道漏极开路输出（V_{DD} 耐压）模式（POM01=1）。详细内容请参照“4.3 控制端口功能的寄存器”、“4.5 使用复用功能时的寄存器设定”。

备注 n=0

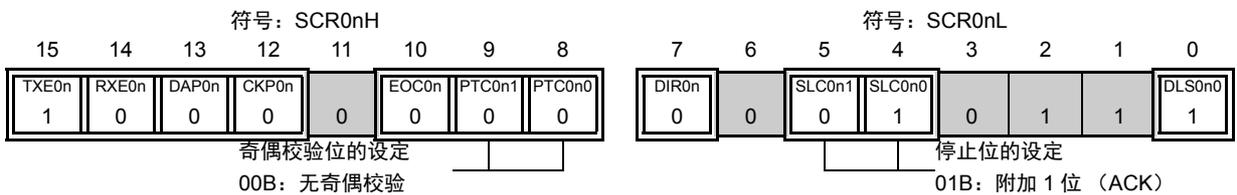
(1) 寄存器的设定

图 12-86 简易 I²C (IIC00) 地址字段发送时的寄存器设定内容例子 (1/2)

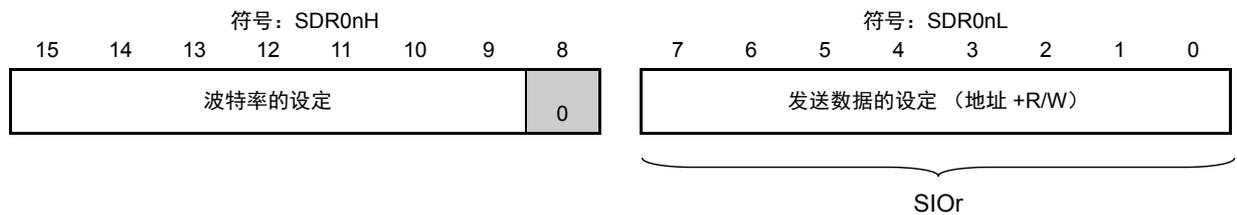
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL)



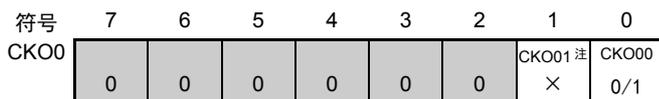
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL)



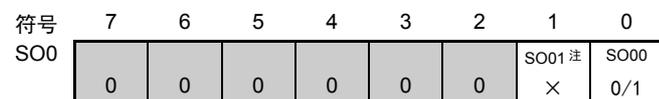
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0)



(e) 串行输出寄存器 0 (SO0)



设定 SO0n 位, 产生开始条件。

注 只限于 16 引脚产品。

图 12-86 简易 I²C (IIC00) 地址字段发送时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0)

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01注 ×	SOE00 0/1

在产生开始条件前 SOE0n 位为“0”，在产生开始条件后 SOE0n 位为“1”。

(g) 串行通道开始寄存器 0 (SS0) 只将对象通道的位置“1”。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

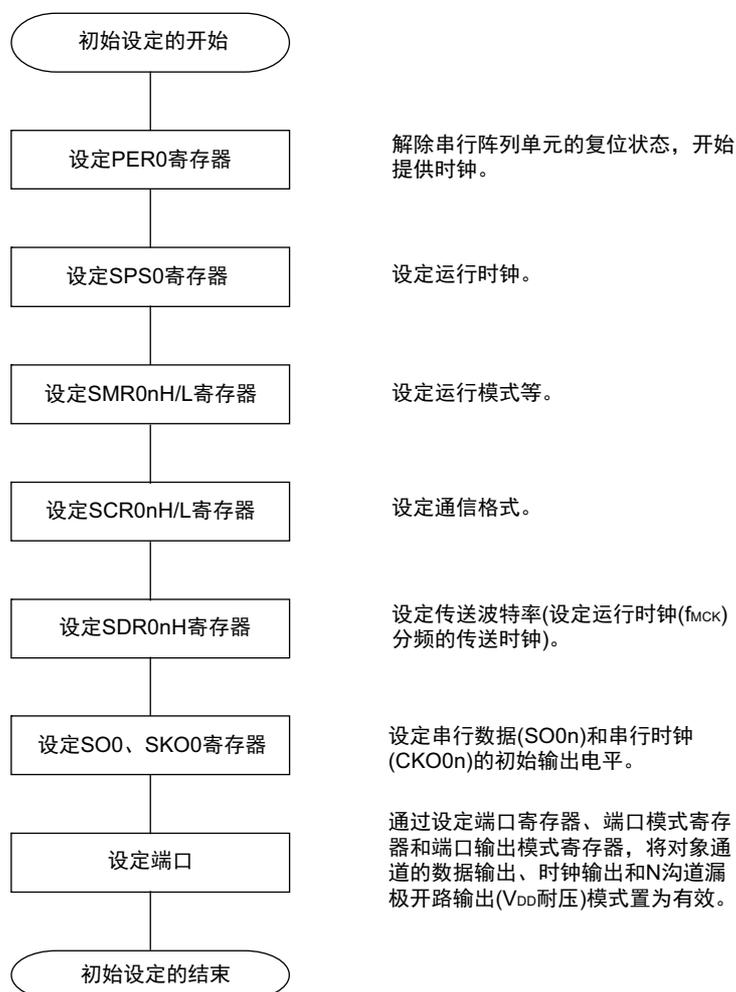
在产生开始条件前 SS0n 位不为“1”，在产生开始条件后 SS0n 位为“1”。

注 只限于 16 引脚产品。

备注 1. n=0, r: IIC 号 (r=00)

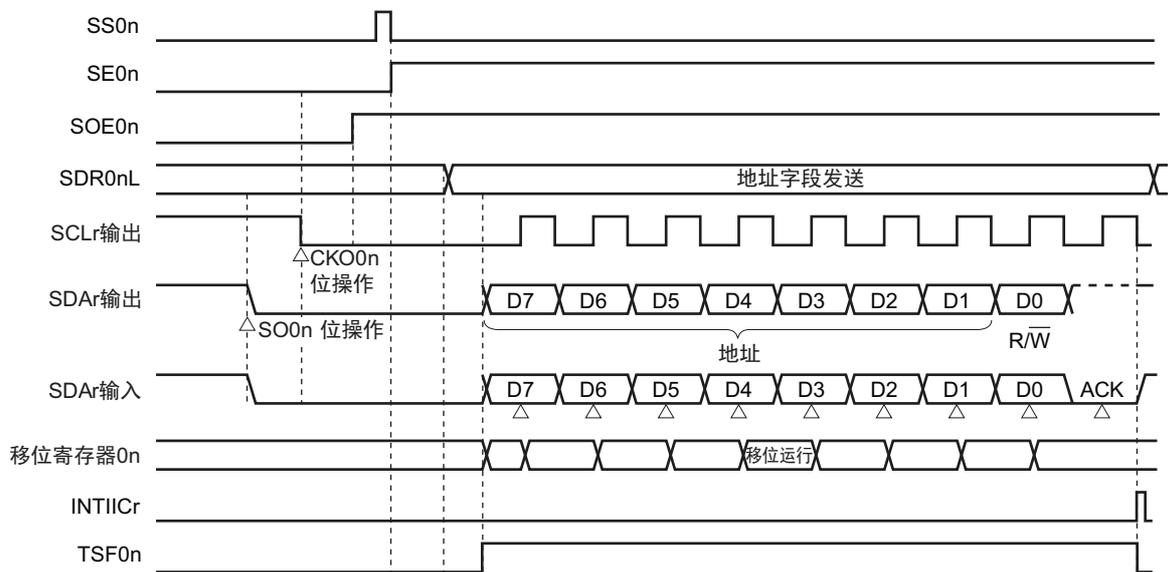
- : 在 IIC 模式中为固定设定。■ : 不能设定 (设定初始值)。
× : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 12-87 简易 I²C 地址字段发送的初始设定步骤

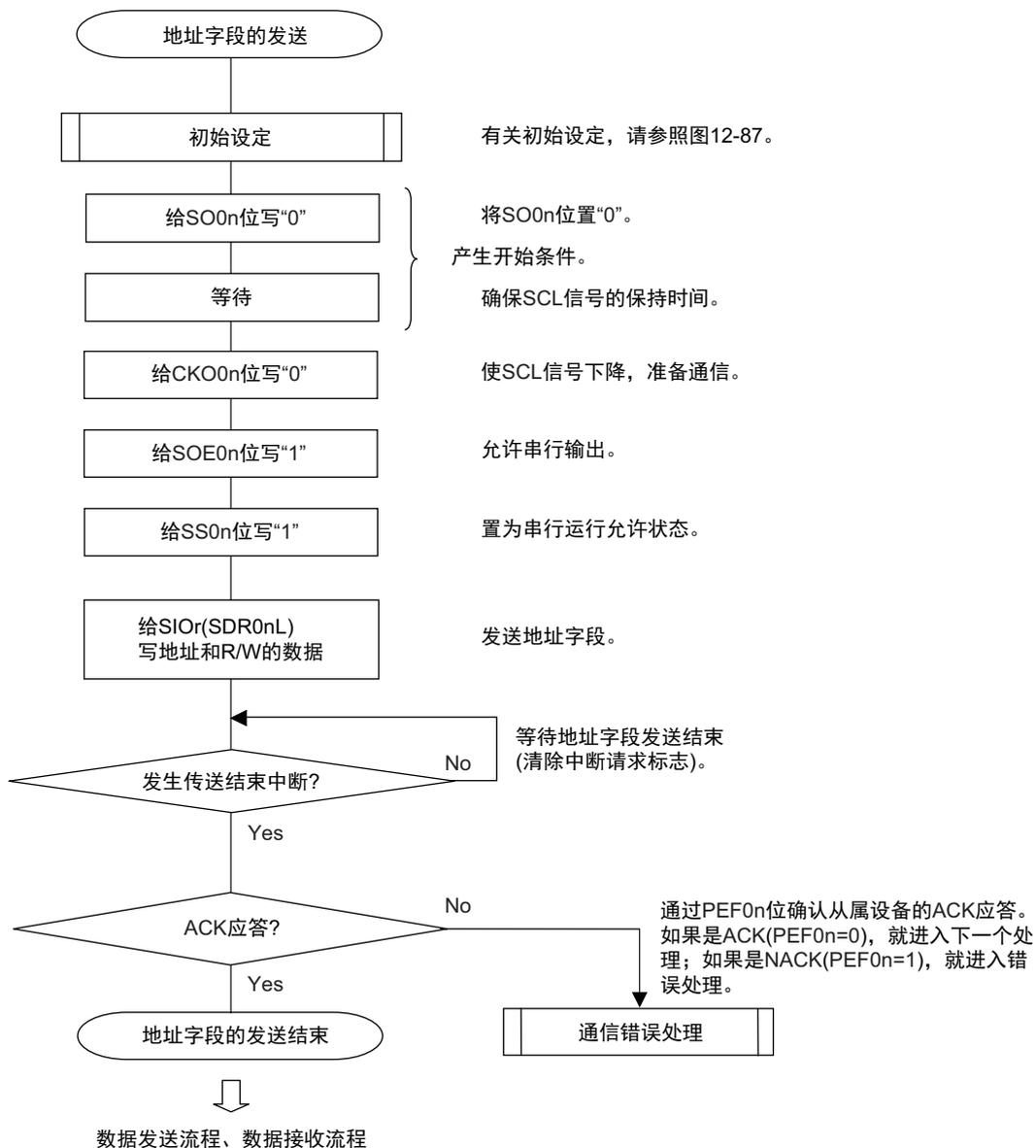
(3) 处理流程

图 12-88 地址字段发送的时序图



备注 0n=00, r: IIC 号 (r=00)

图 12-89 简易 I²C 地址字段发送的流程图



12.7.2 数据发送

数据发送是在发送地址字段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00
对象通道	SAU0 的通道 0
使用的引脚	SCL00、SDA00 注 1
中断	INTIIC00
	只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	ACK 错误检测标志（PEF0n）
传送数据长度	8 位
传送速率注 2	Max. $f_{MCK}/4$ [Hz] ($SDR0nH[7:1] \geq 1$) f_{MCK} : 对象通道的运行时钟频率 但是, 必须在 I ² C 的各模式中满足以下条件: <ul style="list-style-type: none"> • Max. 400kHz (快速模式) • Max. 100kHz (标准模式)
数据电平	不反相输出 (默认值: 高电平)。
奇偶校验位	无奇偶校验位。
停止位	附加 1 位 (用于 ACK 接收时序)
数据方向	MSB 优先

注 1. 要通过简易 I²C 进行通信时, 必须通过端口输出模式寄存器 0 (POM0) 设定 N 沟道漏极开路输出 (V_{DD} 耐压) 模式 (POM01=1)。详细内容请参照“4.3 控制端口功能的寄存器”。

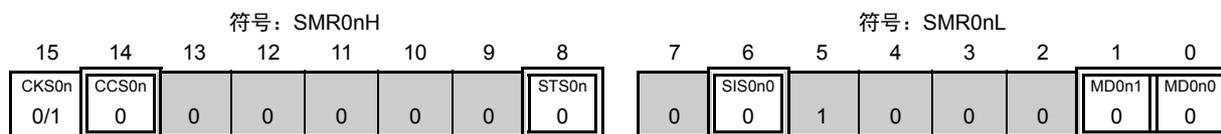
2. 必须在满足此条件并且满足电特性的外围功能特性 (参照“第 24 章 电特性”) 的范围内使用。

备注 n=0

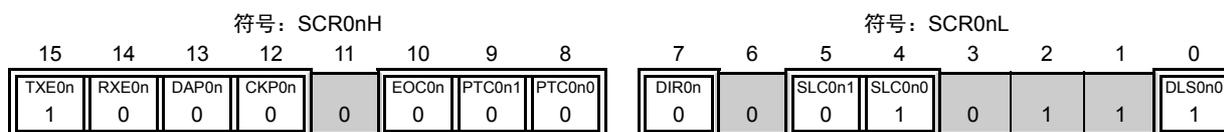
(1) 寄存器的设定

图 12-90 简易 I²C (IIC00) 数据发送时的寄存器设定内容例子 (1/2)

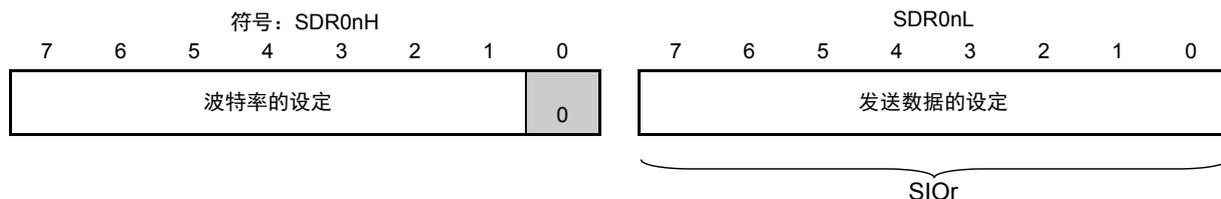
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL) 在数据发送和接收的过程中不操作此寄存器。



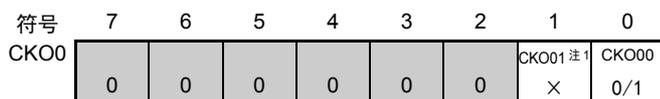
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL) 在数据发送和接收的过程中不操作 TXE0n 位和 RXE0n 位以外的位)。



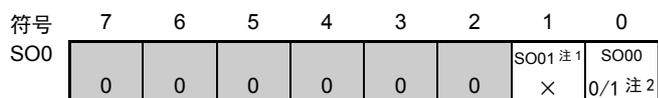
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL) 在数据发送和接收的过程中只有低 8 位有效 (SIOr)。



(d) 串行时钟输出寄存器 0 (CKO0) 在数据发送和接收的过程中不操作此寄存器。



(e) 串行输出寄存器 0 (SO0) 在数据发送和接收的过程中不操作此寄存器。



注 1. 只限于 16 引脚产品。

注 2. 在通信过程中此位的值根据通信数据而变。

图 12-90 简易 I²C (IIC00) 数据发送时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 在数据发送和接收的过程中不操作此寄存器。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01注 ×	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 在数据发送和接收的过程中不操作此寄存器。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

注 只限于 16 引脚产品。

备注 1. n=0, r: IIC 号 (r=00)

- | |
|--|
| |
|--|

 : 在 IIC 模式中为固定设定。

--

 : 不能设定 (设定初始值)。
 × : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 处理流程

图 12-91 数据发送的时序图

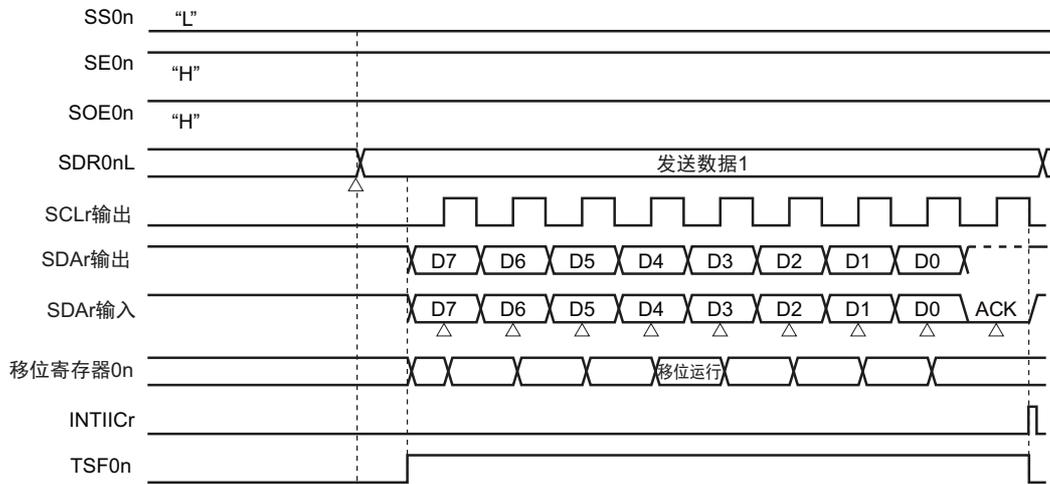
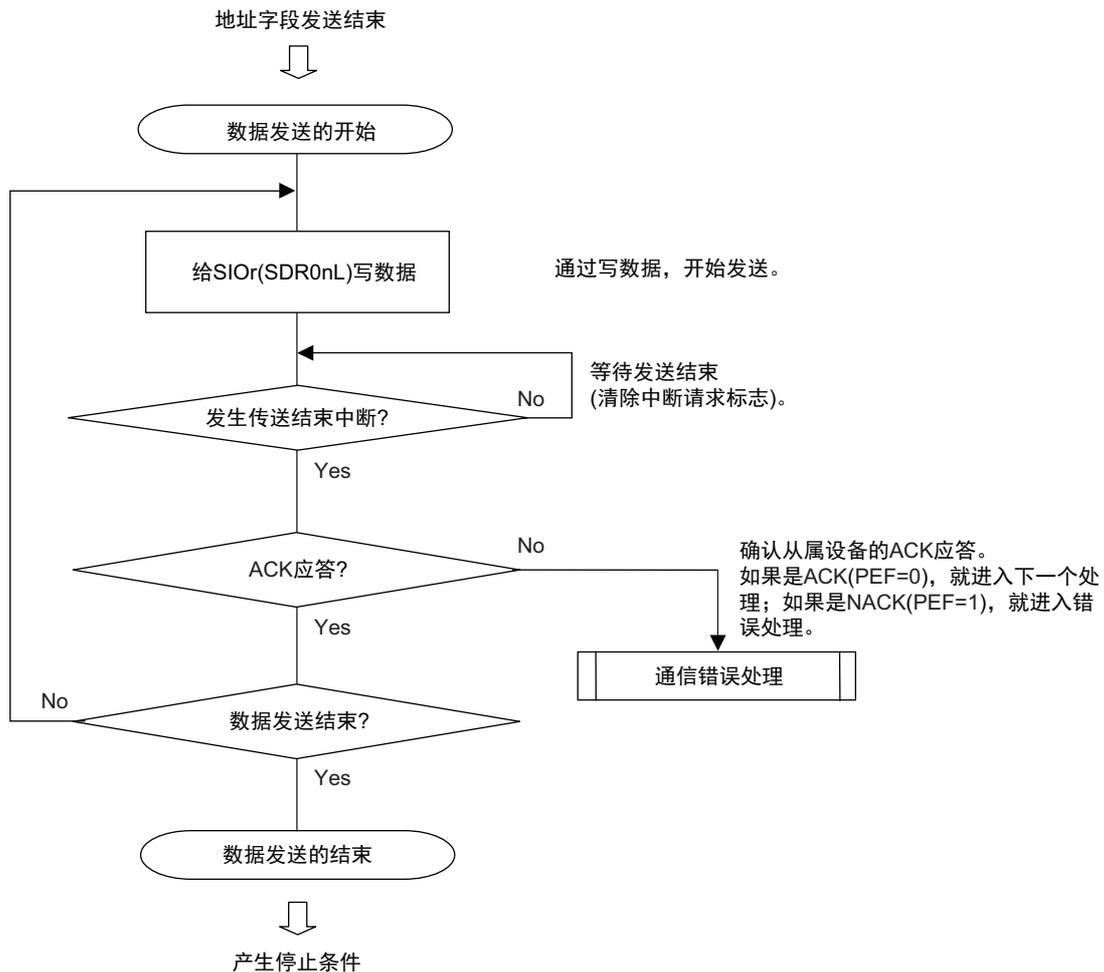


图 12-92 简易 I²C 数据发送的流程图



12.7.3 数据接收

数据接收是在发送地址字段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00
对象通道	SAU0 的通道 0
使用的引脚	SCL00、SDA00 注 1
中断	INTIIC00
	只限于传送结束中断（不能选择缓冲器空中断）。
错误检测标志	只有溢出错误检测标志（OVF0n）。
传送数据长度	8 位
传送速率注 2	Max. $f_{MCK}/4$ [Hz] ($SDR0nH[7:1] \geq 1$) f_{MCK} : 对象通道的运行时钟频率 但是, 必须在 I ² C 的各模式中满足以下条件: <ul style="list-style-type: none"> • Max. 400kHz (快速模式) • Max. 100kHz (标准模式)
数据电平	不反相输出 (默认值: 高电平)。
奇偶校验位	无奇偶校验位。
停止位	附加 1 位 (发送 ACK)。
数据方向	MSB 优先

注 1. 要通过简易 I²C 进行通信时, 必须通过端口输出模式寄存器 0 (POM0) 设定 N 沟道漏极开路输出 (V_{DD} 耐压) 模式 (POM01=1)。详细内容请参照“4.3 控制端口功能的寄存器”。

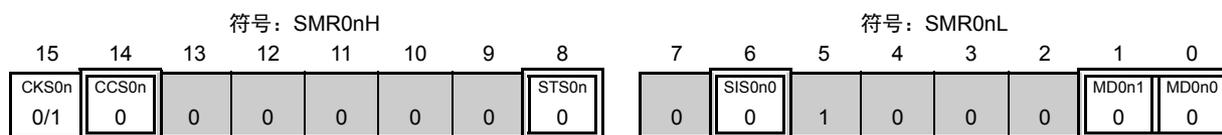
2. 必须在满足此条件并且满足电特性的外围功能特性 (参照“第 24 章 电特性”) 的范围内使用。

备注 n=0

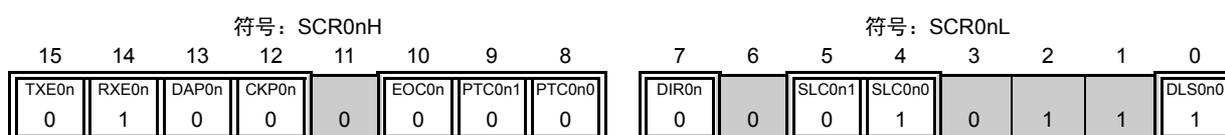
(1) 寄存器的设定

图 12-93 简易 I²C (IIC00) 数据接收时的寄存器设定内容例子 (1/2)

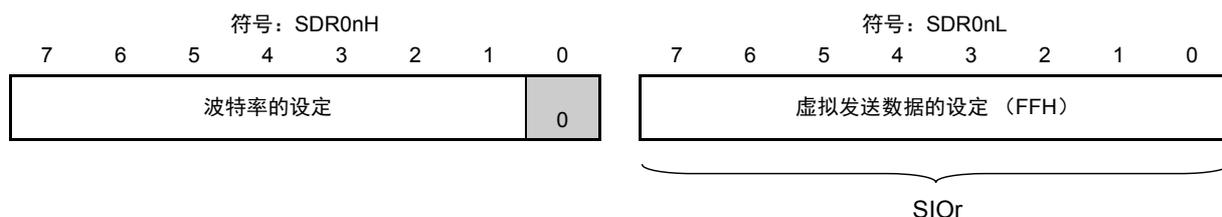
(a) 串行模式寄存器 0n (SMR0nH、SMR0nL) 在数据发送和接收的过程中不操作此寄存器。



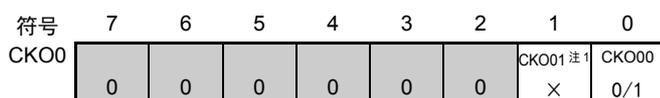
(b) 串行通信运行设定寄存器 0n (SCR0nH、SCR0nL) 在数据发送和接收的过程中不操作 TXE0n 位和 RXE0n 位以外的位。



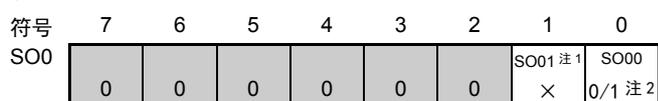
(c) 串行数据寄存器 0n (SDR0nH、SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) 在数据发送和接收的过程中不操作此寄存器。



(e) 串行输出寄存器 0 (SO0) 在数据发送和接收的过程中不操作此寄存器。



注 1. 只限于 16 引脚产品。

注 2. 在通信过程中此位的值根据通信数据而变。

图 12-93 简易 I²C (IIC00) 数据接收时的寄存器设定内容例子 (2/2)

(f) 串行输出允许寄存器 0 (SOE0) 在数据发送和接收的过程中不操作此寄存器。

符号	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01注 ×	SOE00 0/1

(g) 串行通道开始寄存器 0 (SS0) 在数据发送和接收的过程中不操作此寄存器。

符号	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

注 只限于 16 引脚产品。

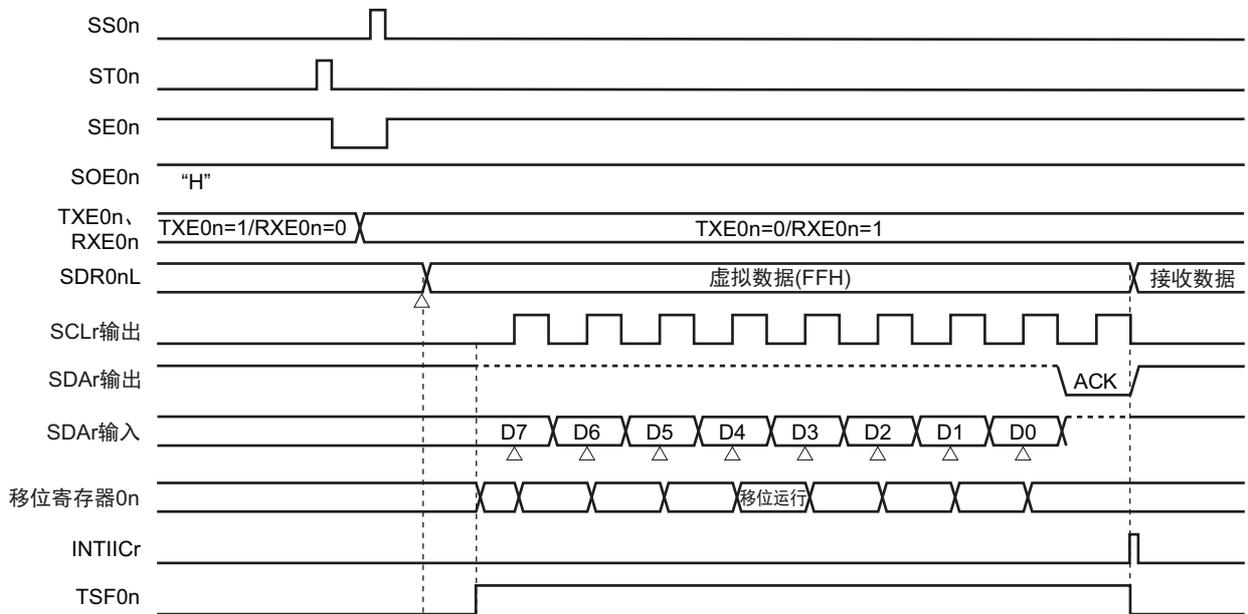
备注 1. n=0, r: IIC 号 (r=00)

- : 在 IIC 模式中为固定设定。 : 不能设定 (设定初始值)。
 × : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1 : 根据用户的用途置“0”或者“1”。

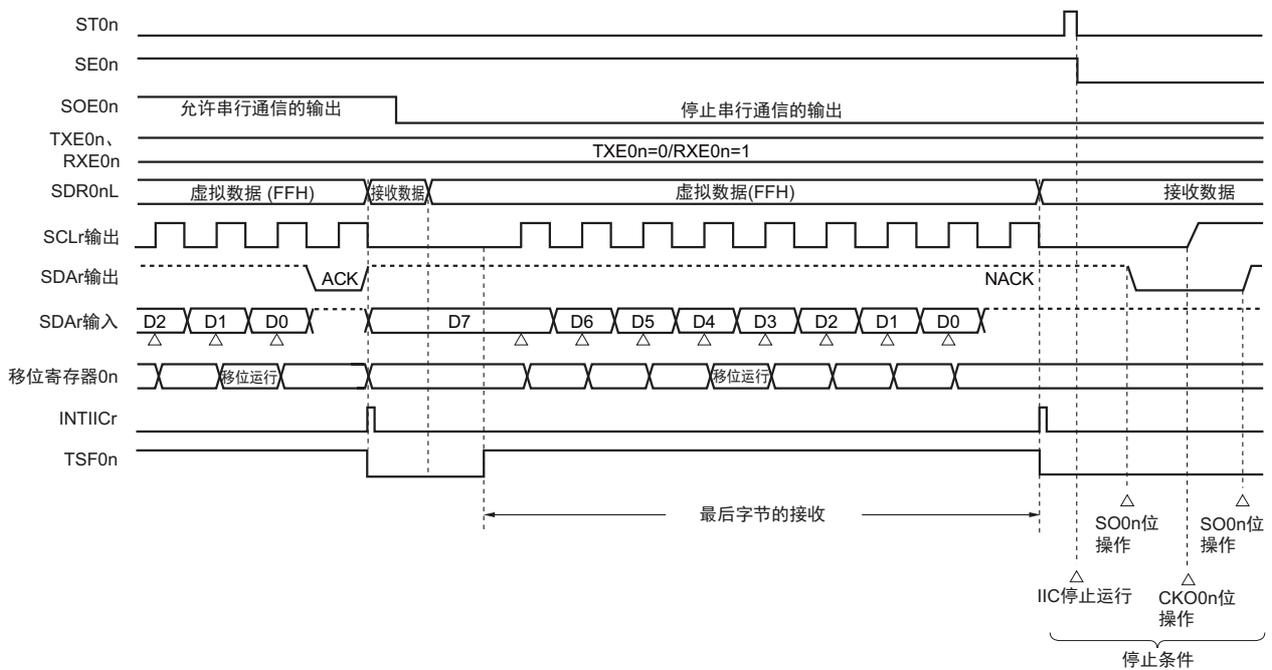
(2) 处理流程

图 12-94 数据接收的时序图

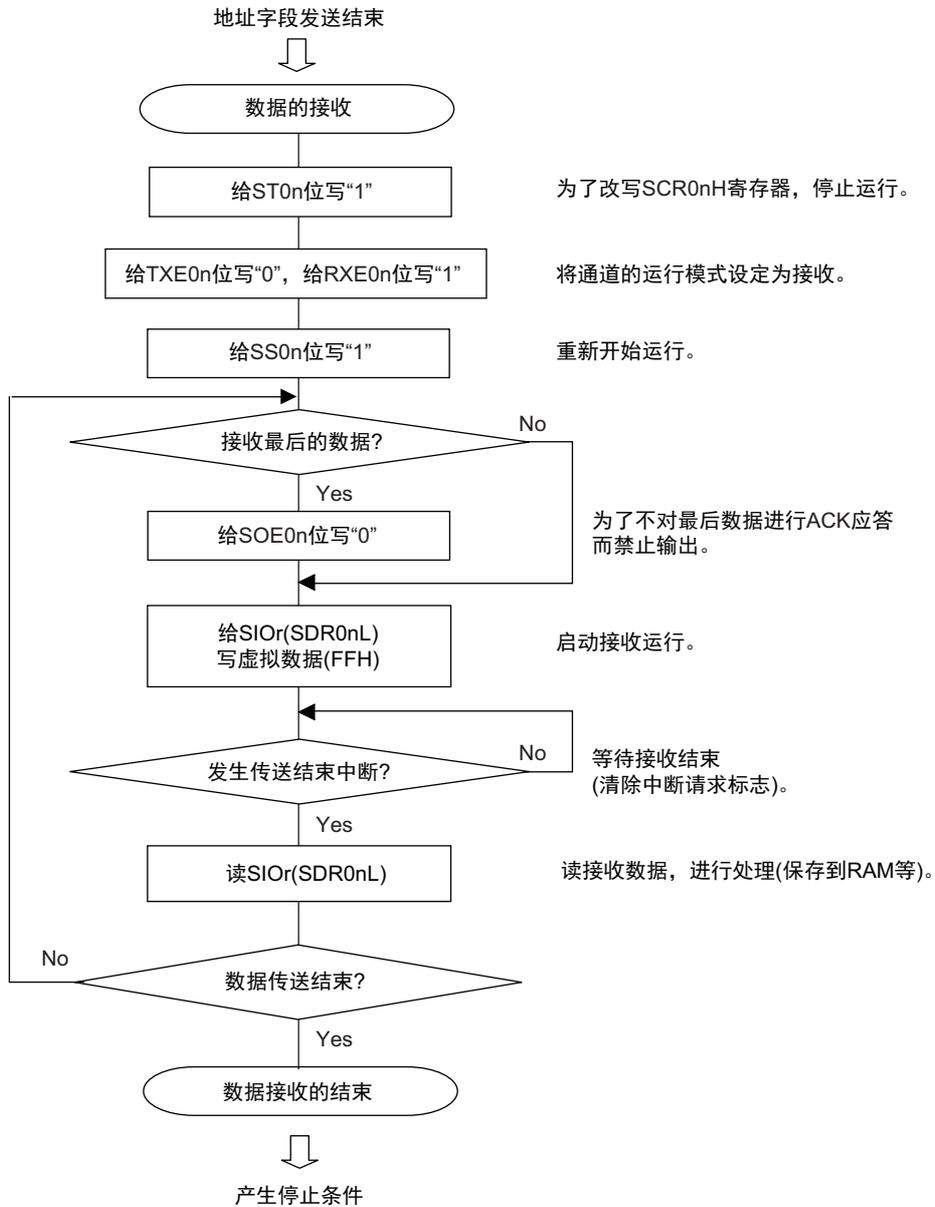
(a) 开始接收数据的情况



(b) 接收最后数据的情况



备注 n=0, r: IIC 号 (r=00)

图 12-95 简易 I²C 数据接收的流程图

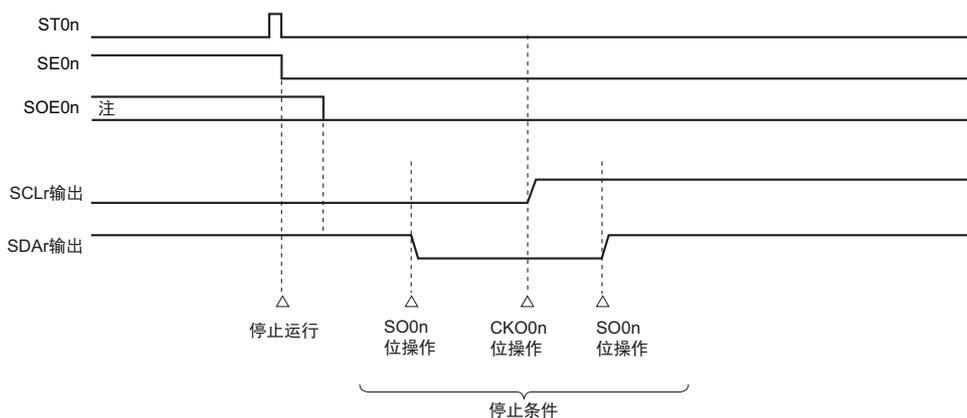
注意 在接收最后数据时不输出 ACK (NACK)。此后，先通过将串行通道停止寄存器 0 (ST0) 的 ST0n 位置“1”来停止运行，然后产生停止条件来结束通信。

12.7.4 停止条件的产生

在与对象从属设备进行了全部数据的发送和接收后，产生停止条件并且释放总线。

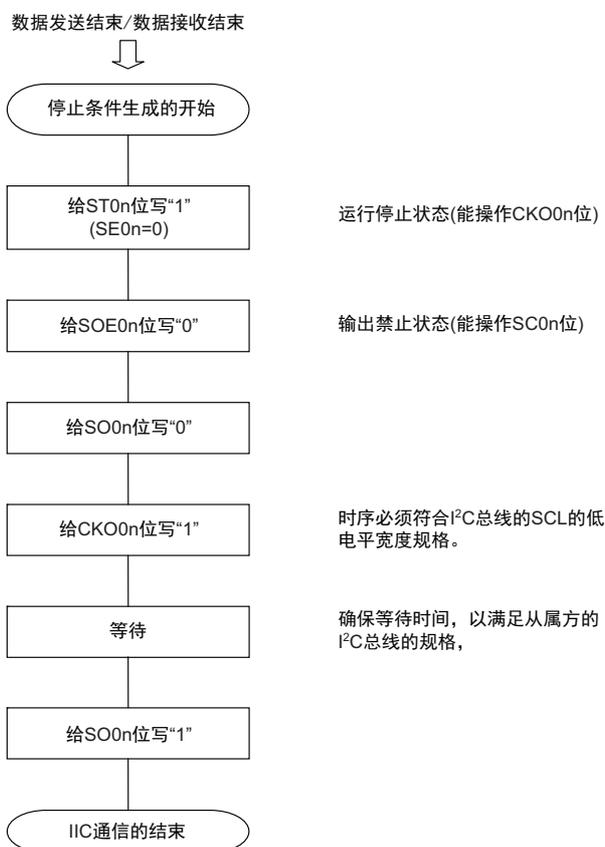
(1) 处理流程

图 12-96 产生停止条件的时序图



注 在接收时，在接收最后数据前已经将串行输出允许寄存器 0（SOE0）的 SOE0n 位置“0”。

图 12-97 产生简易 I²C 停止条件的流程图



12.7.5 传送速率的计算

简易 I²C (IIC00) 通信的传送速率能用以下计算式进行计算。

$$\text{(传送速率)} = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}} \text{) 频率} \} \div (\text{SDR0nH}[7:1] + 1) \div 2$$

注意 禁止将 SDR0nH[7:1] 设定为“0000000B”，SDR0nH[7:1] 的设定值必须大于等于“0000001B”。简易 I²C 输出的 SCL 信号的占空比为 50%。在 I²C 总线规格中，SCL 信号的低电平宽度大于高电平宽度。因此，如果设定为快速模式的 400kbps，SCL 信号输出的低电平宽度就小于 I²C 总线的规格值。必须给 SDR0nH[7:1] 设定能满足 I²C 总线规格的值。

备注 1. 因为 SDR0nH[7:1] 的值为串行数据寄存器 0nH (SDR0nH) 的 bit7 ~ 1 的值 (0000001B ~ 1111111B)，所以为 1 ~ 127。

2. n=0

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0nH (SMR0nH) 的 bit7 (CKS0n 位)。

表 12-4 简易 I²C 运行时钟的选择

SMR0n 寄存器	SPS0 寄存器								运行时钟 (f_{MCK}) 注	
	CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00	f_{CLK}
0	X	X	X	X	0	0	0	0	f_{CLK}	20MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22kHz
X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610Hz	
1	0	0	0	0	X	X	X	X	f_{CLK}	20MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22kHz
1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610Hz	

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 0（ST0）=000FH）后进行更改。

备注 1. X: 忽略

2. n=0

$f_{MCK}=f_{CLK}=20\text{MHz}$ 时的 I²C 传送速率的设定例子如下所示。

I ² C 传送模式 (期待的传送速率)	$f_{CLK}=20\text{MHz}$			
	运行时钟 (f_{MCK})	SDR0nH[7:1]	计算的传送速率	与期待的传送速率的误差
100kHz	$f_{CLK}/2$	49	100kHz	0.0%
400kHz	f_{CLK}	25	384.6kHz	3.8% 注

注 因为 SCL 信号的占空比不为 50%，所以不能将误差设定为“0”% 左右。

12.7.6 在简易 I²C (IIC00) 通信过程中发生错误时的处理步骤

在简易 I²C (IIC00) 通信过程中发生错误时的处理步骤分别如图 12-98、图 12-99 所示。

图 12-98 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 0n (SDR0n)。	SSR0n 寄存器的 BFF0n 位变为“0”并且通道 n 进入可接收状态。	在错误处理过程中结束下一个接收时执行此操作，可防止发生溢出错误。
读串行状态寄存器 0n (SSR0n)。		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 0n (SIR0n) 写“1”。	清除错误标志。	通过将 SSR0n 寄存器的读取值直接写到 SIR0n 寄存器，只能清除读操作时的错误。

图 12-99 发生简易 I²C 模式中的 ACK 错误时的处理步骤

软件操作	硬件状态	备注
读串行状态寄存器 0n (SSR0n)。		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 0n (SIR0n)。	清除错误标志。	通过将 SSR0n 寄存器的读取值直接写到 SIR0n 寄存器，只能清除读操作时的错误。
将串行通道停止寄存器 0 (ST0) 的 ST0n 位置“1”。	串行通道允许状态寄存器 0 (SE0) 的 SE0n 位为“0”并且通道 n 为运行停止状态。	因为没有返回 ACK，所以从属设备没有做好接收的准备。因此，生成停止条件并且释放总线，再次从开始条件开始通信，或者也能生成重新开始条件并且从地址发送开始重新进行。
生成停止条件。		
生成开始条件。		
将串行通道开始寄存器 0 (SS0) 的 SS0n 位置“1”。	串行通道允许状态寄存器 0 (SE0) 的 SE0n 位为“1”并且通道 n 为可运行状态。	

备注 n=0, r: IIC 号 (r=00)

第 13 章 串行接口 IICA

注意 16 引脚产品内置 1 个通道的串行接口 IICA。

13.1 串行接口 IICA 的功能

串行接口 IICA 有以下 3 种模式。

(1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

(2) I²C 总线模式（支持多主控）

此模式通过串行时钟（SCLA0）和串行数据总线（SDAA0）的 2 条线，与多个设备进行 8 位数据传送。

符合 I²C 总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的 I²C 总线控制部分。

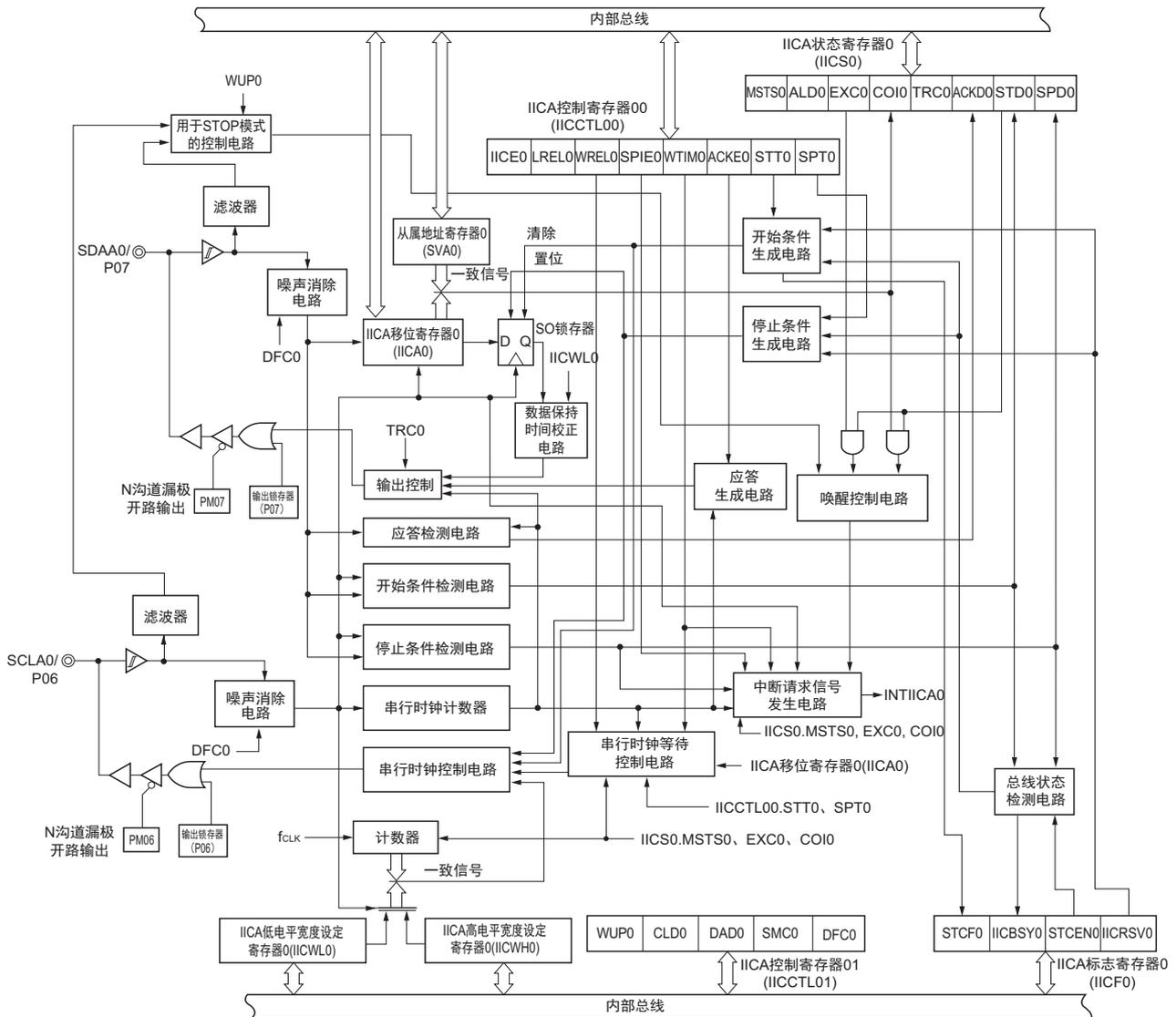
因为串行接口 IICA 的 SCLA0 引脚和 SDAA0 引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

(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（INTIICA0）解除 STOP 模式。通过 IICA 控制寄存器 01（IICCTL01）的 WUP0 位进行设定。

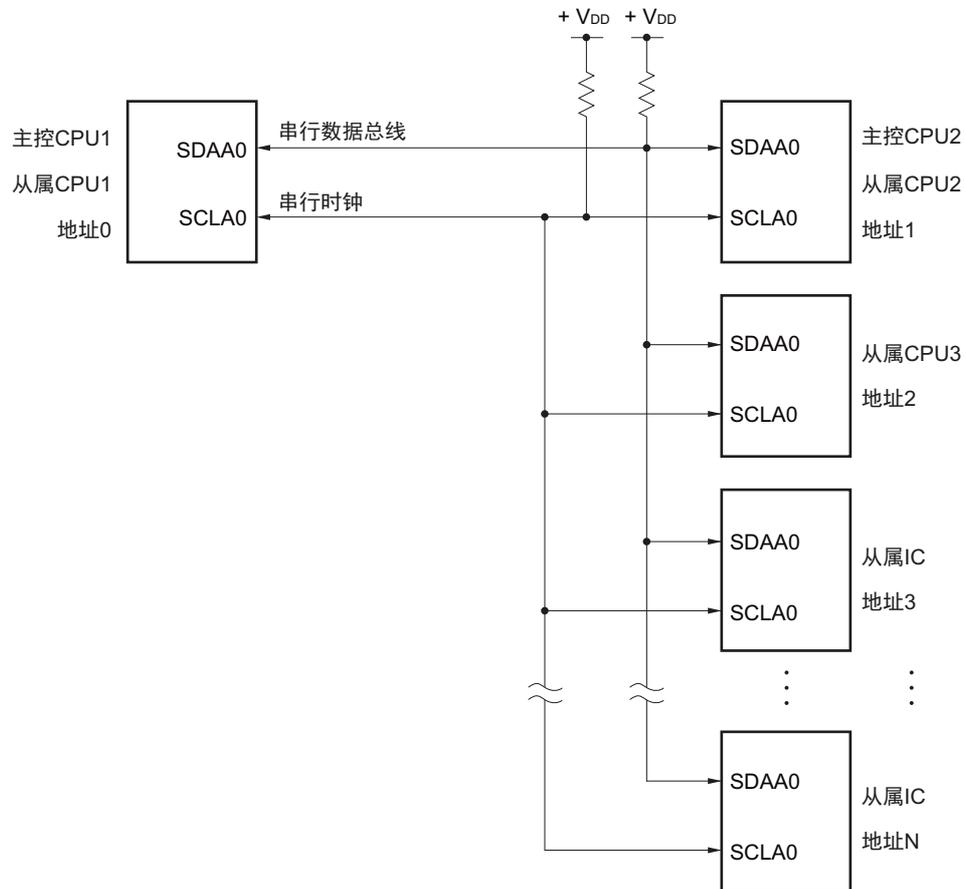
串行接口 IICA 的框图如图 13-1 所示。

图 13-1 串行接口 IICA 的框图



串行总线的结构例子如图 13-2 所示。

图 13-2 I²C 总线的串行总线结构例子



13.2 串行接口 IICA 的结构

串行接口 IICA 由以下硬件构成。

表 13-1 串行接口 IICA 的结构

项目	结构
寄存器	IICA 移位寄存器 0 (IICA0) 从属地址寄存器 0 (SVA0)
控制寄存器	外围允许寄存器 0 (PER0) IICA 控制寄存器 00 (IICCTL00) IICA 状态寄存器 0 (IICS0) IICA 标志寄存器 0 (IICF0) IICA 控制寄存器 01 (IICCTL01) IICA 低电平宽度设定寄存器 0 (IICWL0) IICA 高电平宽度设定寄存器 0 (IICWH0) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0) 端口输出模式寄存器 0 (POM0) 端口模式控制寄存器 0 (PMC0)

(1) IICA 移位寄存器 0 (IICA0)

IICA0 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 IICA0 寄存器来控制实际的接收和发送。
在等待期间，通过写 IICA0 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 IICA0 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 13-3 IICA 移位寄存器 0 (IICA0) 的格式



注意 1. 在数据传送过程中，不能给 IICA0 寄存器写数据。

- 只能在等待期间读写 IICA0 寄存器。除了等待期间以外，禁止在通信状态下存取 IICA0 寄存器。但是，在主控设备的情况下，能在将通信触发位 (STT0) 置“1”后写一次 IICA0 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 IICA0 寄存器写数据。

(2) 从属地址寄存器 0 (SVA0)

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 SVA0 寄存器。但是，在 STD0 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 13-4 从属地址寄存器 0 (SVA0) 的格式

地址: F0234H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0注

注 必须将 bit0 置“0”。

(3) SO 锁存器

SO 锁存器保持 SDAA0 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 0 (SVA0) 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 (INTIICA0)。

(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 (INTIICA0)。

由以下 2 种触发产生 I²C 中断请求。

- 第 8 个或者第 9 个串行时钟的下降沿（通过 WTIM0 位进行设定）
- 因检测到停止条件而产生中断请求（通过 SPIE0 位进行设定）。

备注 WTIM0 位： IICA 控制寄存器 00 (IICCTL00) 的 bit3
SPIE0 位： IICA 控制寄存器 00 (IICCTL00) 的 bit4

(7) 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到 SCLA0 引脚的时钟。

(8) 串行时钟等待控制电路

此电路控制等待时序。

(9) 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

(10) 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

(11) 开始条件生成电路

如果将 STT0 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（IICRSV0 位 =1）并且没释放总线（IICBSY0 位 =1）时，忽视开始条件请求并且将 STCF0 位置“1”。

(12) 停止条件生成电路

如果将 SPT0 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 STCEN0 位设定总线状态检测电路的初始状态。

备注 STT0 位： IICA 控制寄存器 00（IICCTL00）的 bit1
SPT0 位： IICA 控制寄存器 00（IICCTL00）的 bit0
IICRSV0 位： IICA 标志寄存器 0（IICF0）的 bit0
IICBSY0 位： IICA 标志寄存器 0（IICF0）的 bit6
STCF0 位： IICA 标志寄存器 0（IICF0）的 bit7
STCEN0 位： IICA 标志寄存器 0（IICF0）的 bit1

13.3 串行接口 IICA 的控制寄存器

通过以下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 00 (IICCTL00)
- IICA 标志寄存器 0 (IICF0)
- IICA 状态寄存器 0 (IICS0)
- IICA 控制寄存器 01 (IICCTL01)
- IICA 低电平宽度设定寄存器 0 (IICWL0)
- IICA 高电平宽度设定寄存器 0 (IICWH0)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)
- 端口输出模式寄存器 0 (POM0)
- 端口模式控制寄存器 0 (PMC0)

13.3.1 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，降低功耗和噪声。

要使用串行接口 IICA 时，必须将 bit4 (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 13-5 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	TMKAEN 注	CMPEN 注	ADCEN	IICA0EN 注	0	SAU0EN	0	TAU0EN

IICA0EN	控制串行接口 IICA 的输入时钟提供
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写串行接口 IICA 使用的 SFR。 • 串行接口 IICA 处于复位状态。
1	允许提供输入时钟。 <ul style="list-style-type: none"> • 能读写串行接口 IICA 使用的 SFR。

注 只限于 16 引脚产品。

注意 1. 在设定串行接口 IICA 时，必须在 IICA0EN 位为“1”的状态下设定以下寄存器。如果 IICA0EN 位为“0”，串行接口 IICA 的控制寄存器为初始值，并且忽视写操作（端口模式寄存器 0 (PM0)、端口寄存器 0 (P0)、端口输出模式寄存器 0 (POM0) 和端口模式控制寄存器 0 (PMC0) 除外）。

- IICA 控制寄存器 00 (IICCTL00)
- IICA 标志寄存器 0 (IICF0)
- IICA 状态寄存器 0 (IICS0)
- IICA 控制寄存器 01 (IICCTL01)
- IICA 低电平宽度设定寄存器 0 (IICWL0)
- IICA 高电平宽度设定寄存器 0 (IICWH0)

2. 必须将以下的位置“0”。

10 引脚产品: bit1、3、4、6、7

16 引脚产品: bit1 和 bit3

13.3.2 IICA 控制寄存器 00 (IICCTL00)

这是允许或者停止 I²C 运行、设定等待时序以及设定其他 I²C 运行的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICCTL00 寄存器。但是，必须在 IICE0 位为“0”时或者等待期间设定 SPIE0 位、WTIM0 位和 ACKE0 位，而且在将 IICE0 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”。

图 13-6 IICA 控制寄存器 00 (IICCTL00) 的格式 (1/4)

地址: F0230H	复位: 00H	R/W						
符号	7	6	5	4	3	2	1	0
IICCTL00	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I ² C 运行的允许
0	停止运行。对 IICA 状态寄存器 0 (IICS0) 进行复位 ^{注 1} ，并且停止内部运行。
1	允许运行。
必须在 SCLA0 线和 SDAA0 线为高电平的状态下将此位置“1”。	
清除条件 (IICE0=0)	置位条件 (IICE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

LRELO ^{注 2、3}	通信的退出
0	通常运行
1	退出当前的通信，进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况时使用。 SCLA0 线和 SDAA0 线变为高阻抗状态。 IICA 控制寄存器 00 (IICCTL00) 和 IICA 状态寄存器 0 (IICS0) 中的以下标志被清“0”： •STT0 •SPT0 •MSTS0 •EXC0 •COI0 •TRC0 •ACKD0 •STD0
变为退出通信的待机状态，直到满足以下的通信参加条件为止。	
<ul style="list-style-type: none"> 在检测到停止条件后作为主控设备启动。 在检测到开始条件后地址匹配或者接收到扩展码。 	
清除条件 (LRELO=0)	置位条件 (LRELO=1)
<ul style="list-style-type: none"> 在执行后自动清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

WRELO ^{注 2、3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRC0=1) 的第 9 个时钟等待期间将 WRELO 位 (解除等待) 置位，SDAA0 线就变为高阻抗状态 (TRC0=0)。	
清除条件 (WRELO=0)	置位条件 (WRELO=1)
<ul style="list-style-type: none"> 在执行后自动清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

- 注 1. 对 IICA 状态寄存器 0 (IICA0)、IICA 标志寄存器 0 (IICF0) 的 STCF0 位和 IICBSY0 位以及 IICA 控制寄存器 01 (IICCTL01) 的 CLD0 位和 DAD0 位进行复位。
2. 在 IICE0 位为“0”的状态下，此位的信号无效。
3. LRELO 位和 WRELO 位的读取值总是“0”。

注意 如果在 SCLA0 线为高电平、SDAA0 线为低电平并且数字滤波器为 ON (IICCTL01 寄存器的 DFC0=1) 时允许 I²C 运行 (IICE0=1)，就立即检测开始条件。此时，必须在允许 I²C 运行 (IICE0=1) 后连续通过 1 位存储器操作指令将 LRELO 位置“1”。

图 13-6 IICA 控制寄存器 00 (IICCTL00) 的格式 (2/4)

SPIE0 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 01 (IICCTL01) 的 WUP0 位为“1”时, 即使将 SPIE0 位置“1”也不产生停止条件中断。	
清除条件 (SPIE0=0)	置位条件 (SPIE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

WTIM0 注 1	等待和中断请求的控制
0	<p>在第 8 个时钟的下降沿产生中断请求信号。</p> <p>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。</p> <p>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。</p>
1	<p>在第 9 个时钟的下降沿产生中断请求信号。</p> <p>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。</p> <p>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。</p>
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 (WTIM0=0)	置位条件 (WTIM0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

ACKE0 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 SDA A0 线置为低电平。
清除条件 (ACKE0=0)	置位条件 (ACKE0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

注 1. 在 IICE0 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. 在地址传送过程中并且不是扩展码时, 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。

图 13-6 IICA 控制寄存器 00 (IICCTL00) 的格式 (3/4)

STT0 注 1、2	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，IICBSY0 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）。</p> <p>当第三方正在通信时：</p> <ul style="list-style-type: none"> • 允许通信预约功能的情况（IICRSV0=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 • 禁止通信预约功能的情况（IICRSV0=1） 即使将此位置“1”，也清除 STT0 位并且将 STT0 清除标志（STCF0）置“1”，不生成开始条件。 <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"> • 主控接收：禁止在传送过程中将此位置“1”。只有在将 ACKE0 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送：在应答期间，可能不能正常生成开始条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与停止条件的触发（SPT0）同时置“1”。 • 在将 STT0 位置“1”后，禁止在满足清除条件前再次将此位“1”。 	
清除条件（STT0=0）	置位条件（STT0=1）
<ul style="list-style-type: none"> • 在禁止通信预约的状态下将 STT0 位置“1”。 • 在仲裁失败时 • 主控设备生成开始条件。 • 因 LREL0 位为“1”（退出通信）而进行的清除 • 当 IICE0 位为“0”（停止运行）时 • 当复位时 	<ul style="list-style-type: none"> • 通过指令置位。

- 注 1. 在 IICE0 位为“0”的状态下，此位的信号无效。
2. STT0 位的读取值总是为“0”。

备注 IICRSV0: IIC 标志寄存器 0 (IICF0) 的 bit0
STCF0: IIC 标志寄存器 0 (IICF0) 的 bit7

图 13-6 IICA 控制寄存器 00 (IICCTL00) 的格式 (4/4)

SPT0注	停止条件的触发	
0	不生成停止条件。	
1	生成停止条件 (作为主控设备的传送结束)。	
有关置位时序的注意事项:		
<ul style="list-style-type: none"> • 主控接收: 禁止在传送过程中将此位置“1”。只有在将 ACKE0 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送: 在应答期间, 可能不能正常生成停止条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与开始条件的触发 (STT0) 同时置“1”。 • 只有在主控设备的情况下才能将 SPT0 位置“1”。 • 在 WTIM0 位为“0”时, 必须注意: 如果在输出 8 个时钟后的等待期间将 SPT0 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 WTIM0 位从“0”置为“1”并且在输出第 9 个时钟后的等待期间将 SPT0 位置“1”。 • 在将 SPT0 位置“1”后, 禁止在满足清除条件前再次将此位置“1”。 		
清除条件 (SPT0=0)		置位条件 (SPT0=1)
<ul style="list-style-type: none"> • 当仲裁失败时 • 在检测到停止条件后自动清除。 • 因 LRELO 位为“1” (退出通信) 而进行的清除 • 当 IICE0 位为“0” (停止运行) 时 • 当复位时 		<ul style="list-style-type: none"> • 通过指令置位。

注 SPT0 的读取值总是为“0”。

注意 在 IICA 状态寄存器 0 (IICS0) 的 bit3 (TRC0) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICCTL00 寄存器的 bit5 (WRELO) 置“1”来解除等待, 就在清除 TRC0 位 (接收状态) 后将 SDAA0 线置为高阻抗。必须通过写 IICA 移位寄存器 0 进行 TRC0 位为“1” (发送状态) 时的等待解除。

13.3.3 IICA 状态寄存器 0 (IICS0)

这是表示 I²C 状态的寄存器。

只有在 STT0 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 IICS0 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 (WUP0=1) 状态下，禁止读 IICS0 寄存器。在 WUP0 位为“1”的状态下，与 INTIICA0 中断请求无关，如果将 WUP0 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 (SPIE0=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 IICS0 寄存器。

备注 STT0: IICA 控制寄存器 00 (IICCTL00) 的 bit1
WUP0: IICA 控制寄存器 01 (IICCTL01) 的 bit7

图 13-7 IICA 状态寄存器 0 (IICS0) 的格式 (1/3)

地址: FFF51H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	主控状态的确认标志
0	从属状态或者通信待机状态
1	主控通信状态
清除条件 (MSTS0=0)	
<ul style="list-style-type: none"> 当检测到停止条件时 当 ALD0 位为“1”（仲裁失败）时 因 LREL0 位为“1”（退出通信）而进行的清除 当 IICE0 位从“1”变为“0”（停止运行）时 当复位时 	
置位条件 (MSTS0=1)	
<ul style="list-style-type: none"> 当生成开始条件时 	

ALD0	仲裁失败的检测
0	表示未发生仲裁或者赢得仲裁。
1	表示仲裁失败。将 MSTS0 位清除。
清除条件 (ALD0=0)	
<ul style="list-style-type: none"> 在读 IICS0 寄存器后自动清除注。 当 IICE0 位从“1”变为“0”（停止运行）时 当复位时 	
置位条件 (ALD0=1)	
<ul style="list-style-type: none"> 当仲裁失败时 	

注 即使对 IICS0 寄存器以外的位执行 1 位存储器操作指令，也清除此位。因此，在使用 ALD0 位时，必须在读其他位前先读 ALD0 位的数据。

备注 LREL0: IICA 控制寄存器 00 (IICCTL00) 的 bit6
IICE0: IICA 控制寄存器 00 (IICCTL00) 的 bit7

图 13-7 IICA 状态寄存器 0 (IICS0) 的格式 (2/3)

EXC0	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 (EXC0=0)		置位条件 (EXC0=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因 LRELO 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当接收的地址数据的高 4 位为“0000”或者“1111”时 (在第 8 个时钟的上升沿进行置位)。

COI0	匹配地址的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COI0=0)		置位条件 (COI0=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因 LRELO 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当接收地址和本地站地址 (从属地址寄存器 0 (SVA0)) 相同时 (在第 8 个时钟的上升沿置位)。

TRC0	发送 / 接收的状态检测	
0	接收状态 (发送状态除外)。将 SDAA0 线置为高阻抗。	
1	发送状态。设定为能将 SO0 锁存器的值输出到 SDAA0 线 (在第 1 字节的第 9 个时钟的下降沿以后有效)。	
清除条件 (TRC0=0)		置位条件 (TRC0=1)
<p><主控设备和从属设备></p> <ul style="list-style-type: none"> 当检测到停止条件时 因 LRELO 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 因 WRELO 位为“1” (解除等待) 而进行的清除注 当 ALD0 位从“0”变为“1” (仲裁失败) 时 当复位时 不参加通信的情况 (MSTS0、EXC0、COI0=0) <p><主控设备></p> <ul style="list-style-type: none"> 当第 1 字节的 LSB (传送方向指示位) 输出“1”时。 <p><从属设备></p> <ul style="list-style-type: none"> 当检测到开始条件时 当第 1 字节的 LSB (传送方向指示位) 输入“0”时 		<p><主控设备></p> <ul style="list-style-type: none"> 当生成开始条件时 当将第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输出“0” (主控发送) 时 <p><从属设备></p> <ul style="list-style-type: none"> 当来自主控设备的第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输入“1” (从属发送) 时

注 在 IICA 状态寄存器 0 (IICS0) 的 bit3 (TRC0) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICA 控制寄存器 00 (IICCTL00) 的 bit5 (WRELO) 置“1”来解除等待, 就在清除 TRC0 位 (接收状态) 后将 SDAA0 线置为高阻抗。必须通过写 IICA 移位寄存器 0 进行 TRC0 位为“1” (发送状态) 时的等待解除。

备注 LRELO: IICA 控制寄存器 00 (IICCTL00) 的 bit6
IICE0: IICA 控制寄存器 00 (IICCTL00) 的 bit7

图 13-7 IICA 状态寄存器 0 (IICS0) 的格式 (3/3)

ACKD0	应答 (ACK) 的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件 (ACKD0=0)		置位条件 (ACKD0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当下一字节的第 1 个时钟上升时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 在 SCLA0 线的第 9 个时钟上升沿将 SDAA0 线为低电平时

STD0	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 (STD0=0)		置位条件 (STD0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 在地址传送后的下一个字节的第 1 个时钟上升时 因 LREL0 位为“1” (退出通信) 而进行的清除 当 IICE0 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到开始条件时

SPD0	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 (SPD0=0)		置位条件 (SPD0=1)
<ul style="list-style-type: none"> 在将此位置位后, 在检测到开始条件后的地址传送位的第 1 个时钟上升时 当 WUP0 位从“1”变为“0”时 当 IICE0 位从“1”变为“0” (停止运行) 当复位时 		<ul style="list-style-type: none"> 当检测到停止条件时

备注 LREL0: IICA 控制寄存器 00 (IICCTL00) 的 bit6

IICE0: IICA 控制寄存器 00 (IICCTL00) 的 bit7

13.3.4 IICA 标志寄存器 0 (IICF0)

这是设定 I²C 运行模式以及表示 I²C 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 IICF0 寄存器。但是, STT0 清除标志 (STCF0) 和 I²C 总线状态标志 (IICBSY0) 位为只读位。

通过 IICRSV0 位设定允许或者禁止通信预约功能, 并且通过 STCEN0 位设定 IICBSY0 位的初始值。

只有在禁止 I²C 运行 (IICA 控制寄存器 00 (IICCTL00) 的 bit7 (IICE0) =0) 时才能写 IICRSV0 位和 STCEN0 位。在允许运行后, IICF0 寄存器为只读寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 13-8 IICA 标志寄存器 0 (IICF0) 的格式

地址: FFF52H 复位后: 00H R/W 注

符号	7	6	5	4	3	2	1	0
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 清除标志
0	发行开始条件。
1	无法发行开始条件而清除 STT0 标志。
清除条件 (STCF0=0)	置位条件 (STCF0=1)
<ul style="list-style-type: none"> 因 STT0 位为“1”而进行的清除 当 IICE0 位为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 在设定为禁止通信预约 (IICRSV0=1) 的状态下无法发行开始条件而将 STT0 位清“0”时

IICBSY0	I ² C 总线状态标志
0	总线释放状态 (STCEN0=1 时的通信初始状态)
1	总线通信状态 (STCEN0=0 时的通信初始状态)
清除条件 (IICBSY0=0)	置位条件 (IICBSY0=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当 IICE0 位为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 当检测到开始条件时 STCEN0 位为“0” 时的 IICE0 位的置位

STCEN0	初始开始允许触发
0	在允许运行 (IICE0=1) 后, 通过检测停止条件而允许生成开始条件。
1	在允许运行 (IICE0=1) 后, 不通过检测停止条件而允许而生成开始条件。
清除条件 (STCEN0=0)	置位条件 (STCEN0=1)
<ul style="list-style-type: none"> 通过指令清除。 当检测到开始条件时 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

IICRSV0	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 (IICRSV0=0)	置位条件 (IICRSV0=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 	<ul style="list-style-type: none"> 通过指令置位。

注 bit6 和 bit7 是只读位。

注意 1. 只有在停止运行 (IICE0=0) 时才能写 STCEN0 位。

2. 如果 STCEN0 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSY0=0), 因此为了避免在发行第 1 个开始条件 (STT0=1) 时破坏其他通信, 必须确认没有正在通信的第三方。

3. 只有在停止运行 (IICE0=0) 时才能写 IICRSV0。

备注 1. STT0: IICA 控制寄存器 00 (IICCTL00) 的 bit1

2. IICE0: IICA 控制寄存器 00 (IICCTL00) 的 bit7

13.3.5 IICA 控制寄存器 01 (IICCTL01)

这是用于设定 I²C 运行模式以及检测 SCLA0 引脚和 SDAA0 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 IICCTL01 寄存器。但是，CLD0 位和 DAD0 位是只读位。

除了 WUP0 位以外，必须在禁止 I²C 运行 (IICA 控制寄存器 00 (IICCTL00) 的 bit7 (IICE0) =0) 时设定 IICCTL01 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 13-9 IICA 控制寄存器 01 (IICCTL01) 的格式 (1/2)

地址: F0231H 复位后: 00H R/W 注 1

符号	7	6	5	4	3	2	1	0
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	0

WUP0	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 WUP0 位置“1”来转移到 STOP 模式时，必须在将 WUP0 位置“1”后至少经过 3 个 f_{CLK} 时钟，然后执行 STOP 指令（参照“图 13-21 将 WUP0 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 WUP0 位清“0”。能通过将 WUP0 位清“0”参加后续的通信（需要在将 WUP0 位清“0”后解除等待以及写发送数据）。</p> <p>在 WUP0 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 WUP0 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 WUP0 位为“1”时，即使将 SPIE0 位置“1”也不产生停止条件中断。</p>	
清除条件 (WUP0=0)	置位条件 (WUP0=1)
<ul style="list-style-type: none"> 通过指令清除（在地址匹配或者接收到扩展码后）。 	<ul style="list-style-type: none"> 通过指令置位 (MSTS0=0、EXC0=0、COI0=0 并且 STD0=0（不参加通信）) 注 2。

- 注 1. bit4 和 bit5 是只读位。
 2. 在以下所示的期间，需要确认 IICA 状态寄存器 0 (IICS0) 的状态。

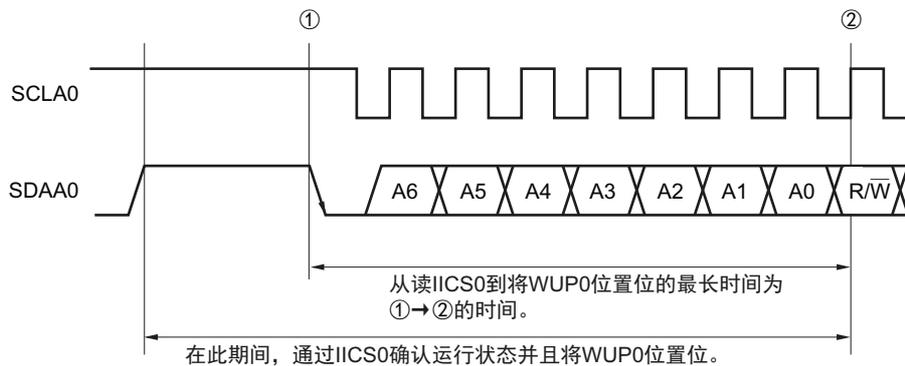


图 13-9 IICA 控制寄存器 01 (IICCTL01) 的格式 (2/2)

CLD0	SCLA0 引脚的电平检测 (只在 IICE0 位为“1”时有效)	
0	检测到 SCLA0 引脚为低电平。	
1	检测到 SCLA0 引脚为高电平。	
清除条件 (CLD0=0)		置位条件 (CLD0=1)
<ul style="list-style-type: none"> 当 SCLA0 引脚为低电平时 当 IICE0 位为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当 SCLA0 引脚为高电平时

DAD0	SDAA0 引脚的电平检测 (只在 IICE0 位为“1”时有效)	
0	检测到 SDAA0 引脚为低电平。	
1	检测到 SDAA0 引脚为高电平。	
清除条件 (DAD0=0)		置位条件 (DAD0=1)
<ul style="list-style-type: none"> 当 SDAA0 引脚为低电平时 当 IICE0 位为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当 SDAA0 引脚为高电平时

SMC0	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)。	

DFC0	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>数字滤波器用于消除噪声。</p> <p>无论是将 DFC0 位置“1”还是清“0”，传送时钟都不变。</p>		

注意 在设定传送时钟时，必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 的最小工作频率取决于模式。

标准模式中: $f_{CLK}=1\text{MHz}(\text{Min.})$

快速模式中: $f_{CLK}=3.5\text{MHz}(\text{Min.})$

备注 IICE0: IICA 控制寄存器 00 (IICCTL00) 的 bit7

13.3.6 IICA 低电平宽度设定寄存器 0 (IICWL0)

此寄存器控制串行接口 IICA 输出的 SCLA0 引脚信号低电平宽度 (t_{LOW}) 和 SDAA0 引脚信号。

通过 8 位存储器操作指令设定 IICWL0 寄存器。

必须在禁止 I²C 运行 (IICA 控制寄存器 00 (IICCTL00) 的 bit7 (IICE0) =0) 时设定 IICWL0 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

有关 IICWL0 寄存器的设定方法, 请参照“13.4.2 通过 IICWL0 寄存器和 IICWH0 寄存器设定传送时钟的方法”。

图 13-10 IICA 低电平宽度设定寄存器 0 (IICWL0) 的格式



13.3.7 IICA 高电平宽度设定寄存器 0 (IICWH0)

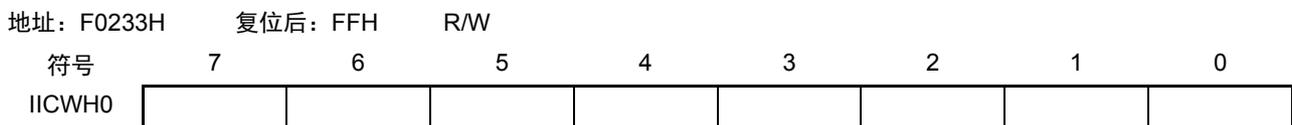
此寄存器控制串行接口 IICA 输出的 SCLA0 引脚信号高电平宽度和 SDAA0 引脚信号。

通过 8 位存储器操作指令设定 IICWH0 寄存器。

必须在禁止 I²C 运行 (IICA 控制寄存器 00 (IICCTL00) 的 bit7 (IICE0) =0) 时设定 IICWH0 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

图 13-11 IICA 高电平宽度设定寄存器 0 (IICWH0) 的格式



备注 有关主控方的传送时钟的设定方法, 请参照“13.4.2(1) 主控方传送时钟的设定方法”, 有关从属方的 IICWL_n、IICWH₀ 寄存器的设定方法, 请参照“13.4.2(2) 从属方 IICWL₀ 寄存器和 IICWH₀ 寄存器的设定方法”。

13.3.8 IICA 串行输入 / 输出引脚的端口功能控制寄存器

使用 IICA 时, 必须设定 IICA 串行输入 / 输出引脚 (SCLA0 引脚和 SDAA0 引脚) 的复用端口功能的控制寄存器 (端口模式寄存器 0 (PM0)、端口寄存器 0 (P0)、端口输出模式寄存器 0 (POM0) 和端口模式控制寄存器 0 (PMC0))。

端口功能的控制寄存器的详细内容请参照“4.3.1 端口模式寄存器 0、4 (PM0、PM4)”、“4.3.2 端口寄存器 0、4、12、13 (P0、P4、P12、P13)”、“4.3.4 端口输出模式寄存器 0 (POM0)”和“4.3.5 端口模式控制寄存器 0 (PMC0)”。

在用作 IICA0 的时钟输入 / 输出 (SCLA0) 引脚、串行数据输入 / 输出 (SDAA0) 引脚时, 必须将对应的端口模式寄存器 0 (PM0) 和端口模式控制寄存器 0 (PMC0) 的位置“0”, 将对应的端口寄存器 0 (PM0) 和端口输出模式寄存器 0 (POM0) 的位置“1”。详细内容请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

另外, 因为用作 N 沟道漏极开路输出 (V_{DD} 耐压) 模式, 必须通过电阻上拉至外部设备的电源。

13.4 I²C 总线模式的功能

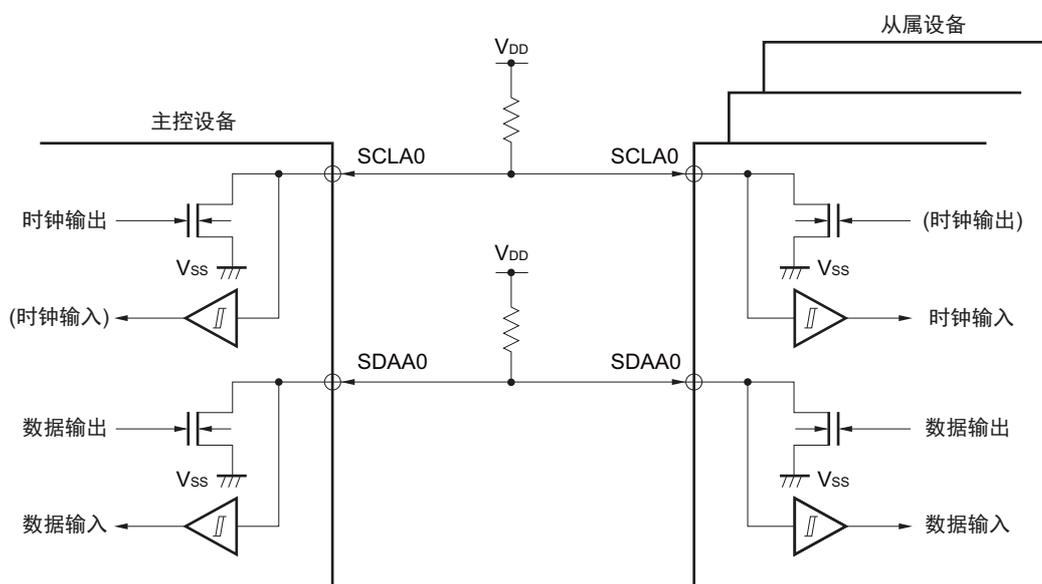
13.4.1 引脚结构

串行时钟引脚（SCLA0）和串行数据总线引脚（SDAA0）的结构如下。

- (1) SCLA0..... 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) SDAA0..... 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 13-12 引脚结构图



13.4.2 通过 IICWL0 寄存器和 IICWH0 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，IICWL0 寄存器和 IICWH0 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

$$\text{IICWL0} = \frac{0.52}{\text{传送时钟}} \times f_{\text{CLK}}$$

$$\text{IICWH0} = \left(\frac{0.48}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}$$

- 标准模式

$$\text{IICWL0} = \frac{0.47}{\text{传送时钟}} \times f_{\text{CLK}}$$

$$\text{IICWH0} = \left(\frac{0.53}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}$$

(2) 从属方 IICWL0 寄存器和 IICWH0 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

$$\text{IICWL0} = 1.3\mu\text{s} \times f_{\text{CLK}}$$

$$\text{IICWH0} = (1.2\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}$$

- 标准模式

$$\text{IICWL0} = 4.7\mu\text{s} \times f_{\text{CLK}}$$

$$\text{IICWH0} = (5.3\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}$$

注意 在设定传送时钟时，必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 的最小工作频率取决于模式。

快速模式中： $f_{\text{CLK}}=3.5\text{MHz}(\text{Min.})$

标准模式中： $f_{\text{CLK}}=1\text{MHz}(\text{Min.})$

备注 1. 因为 SDAA0 信号和 SCLA0 信号的上升时间 (t_{R}) 和下降时间 (t_{F}) 因上拉电阻和布线负载而不同，所以必须各自计算。

2. IICWL0: IICA 低电平宽度设定寄存器 0

IICWH0: IICA 高电平宽度设定寄存器 0

t_{F} : SDAA0 信号和 SCLA0 信号的下降时间

t_{R} : SDAA0 信号和 SCLA0 信号的上升时间

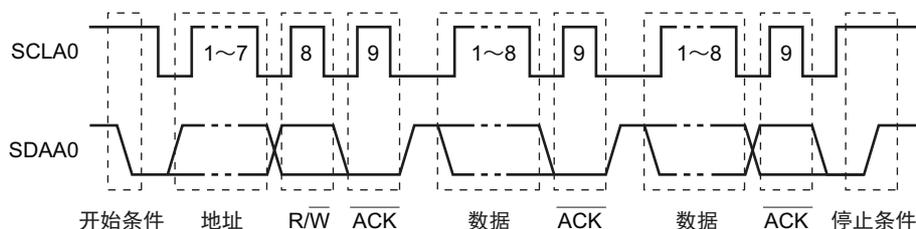
f_{CLK} : CPU/ 外围硬件的时钟频率

13.5 I²C 总线的定义和控制方法

以下说明 I²C 总线的串行数据通信格式和使用的信号。

I²C 总线的串行数据总线上生成的“开始条件”、“地址”、“数据”和“停止条件”的各传送时序如图 13-13 所示。

图 13-13 I²C 总线的串行数据传送时序



主控设备生成开始条件、从属地址和停止条件。

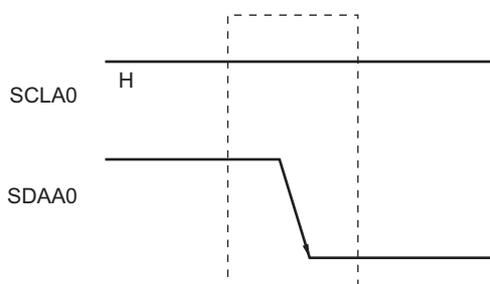
主控设备和从属设备都能生成应答（ACK）（在一般情况下，接收方输出 8 位数据）。

主控设备连续输出串行时钟（SCL A0）。但是，从属设备能延长 SCL A0 引脚的低电平期间并且插入等待。

13.5.1 开始条件

在 SCL A0 引脚为高电平时，如果 SDA A0 引脚从高电平变为低电平，就生成开始条件。SCL A0 引脚和 SDA A0 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。从属设备能检测到开始条件。

图 13-14 开始条件



在检测到停止条件（SPD0: IICA 状态寄存器 0（IICS0）的 bit0=1）的状态下，如果将 IICA 控制寄存器 00（ICCTL00）的 bit1（STT0）置“1”，就输出开始条件。如果检测到开始条件，就将 IICS0 寄存器的 bit1（STD0）置“1”。

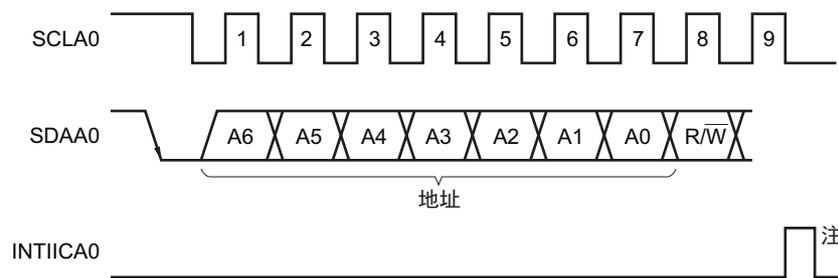
13.5.2 地址

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 0（SVA0）的内容相同。此时，如果 7 位数据和 SVA0 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 13-15 地址



注 如果在从属设备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 INTIICA0。

如果将从属地址和“13.5.3 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 0（IICA0），就输出地址。接收到的地址被写到 IICA0 寄存器。

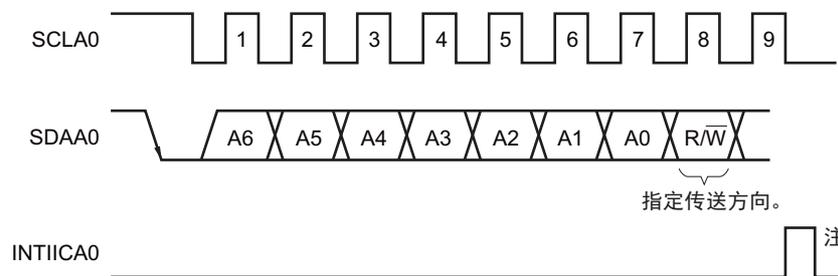
从属地址分配在 IICA0 寄存器的高 7 位。

13.5.3 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 13-16 传送方向的指定



注 如果在从属设备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 INTIICA0。

13.5.4 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 0 (IICS0) 的 bit2 (ACKD0) 确认应答的检测。

在主控设备为接收状态下接收到最后的数据时, 不返回应答而生成停止条件。在从属设备接收数据后不返回应答时, 主控设备输出停止条件或者重新开始条件, 中止传送。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。

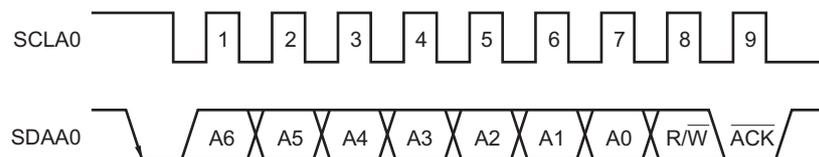
接收方在第 9 个时钟将 SDAA0 线置为低电平, 生成应答 (正常接收)。

通过将 IICA 控制寄存器 00 (IICCTL00) 的 bit2 (ACKE0) 置“1”, 变为能够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 IICS0 寄存器的 bit3 (TRC0)。在接收 (TRC0=0) 的情况下, 通常必须将 ACKE0 位置“1”。

在从属接收运行过程中 (TRC0=0) 不能接收数据或者不需要下一个数据时, 必须将 ACKE0 位清“0”, 通知主控方不能接收数据。

在从属接收运行过程中 (TRC0=0) 不需要下一个数据时, 为了不生成应答, 必须将 ACKE0 位清“0”, 通知从属发送方数据的结束 (停止发送)。

图 13-17 应答



当接收到本地站的地址时, 与 ACKE0 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)。

通过事先将 ACKE0 位置“1”, 在接收到扩展码时生成应答。

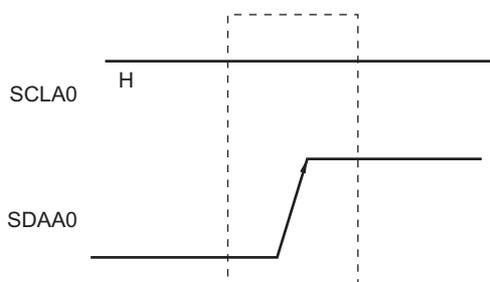
接收数据时的应答生成方法因等待时序的设定而不同, 如下所示。

- 当选择 8 个时钟等待时 (IICCTL00 寄存器的 bit3 (WTIM0) =0):
通过在解除等待前将 ACKE0 位置“1”, 与 SCLA0 引脚的第 8 个时钟下降同步生成应答。
- 当选择 9 个时钟等待时 (IICCTL00 寄存器的 bit3 (WTIM0) =1):
通过事先将 ACKE0 位置“1”, 生成应答。

13.5.5 停止条件

在 SCLA0 引脚为高电平时，如果 SDAA0 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。从属设备能检测到停止条件。

图 13-18 停止条件



如果将 IICA 控制寄存器 00 (IICCTL00) 的 bit0 (SPT0) 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 0 (IICS0) 的 bit0 (SPD0) 置“1”，并且在 IICCTL00 寄存器的 bit4 (SPIE0) 为“1”时产生 INTIICA0。

13.5.6 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接收 / 发送（等待状态）。

通过将 SCLA0 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解除，就能开始下一次传送。

图 13-19 等待 (1/2)

- (1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况
(主控设备：发送，从属设备：接收，ACKE0=1)

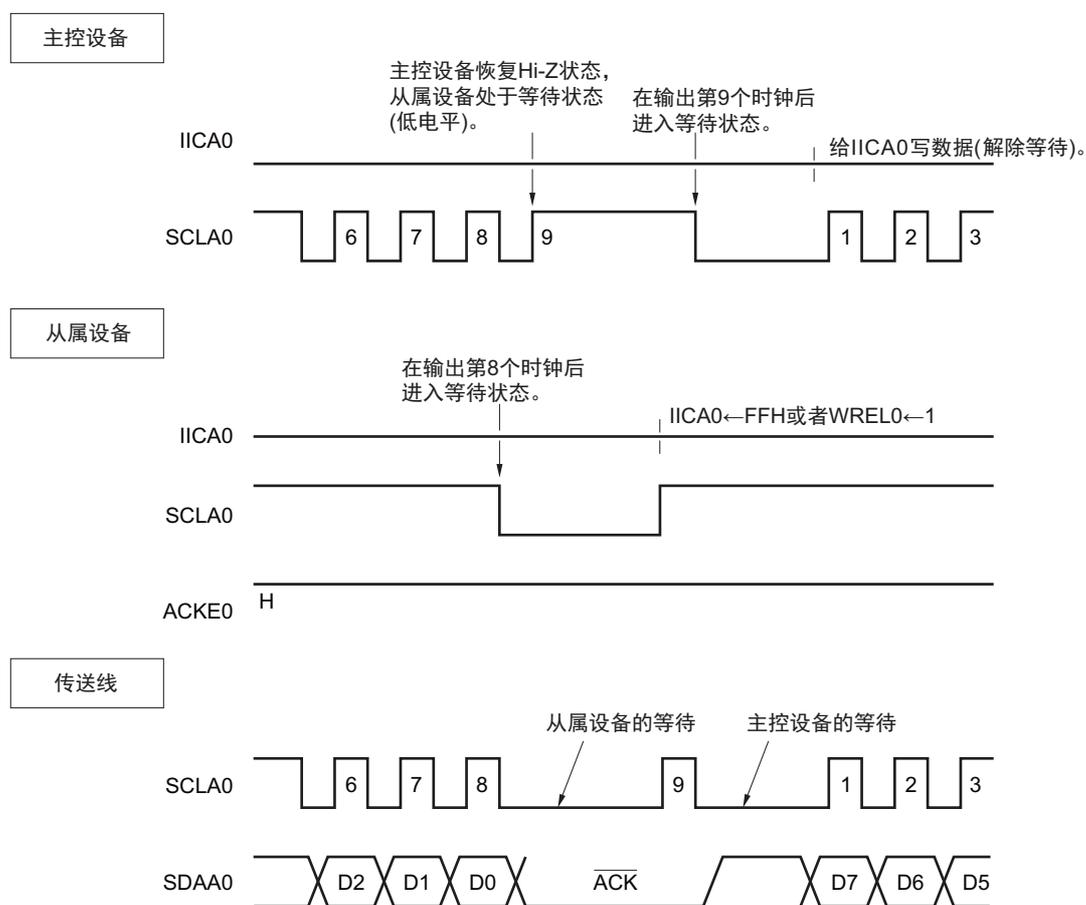
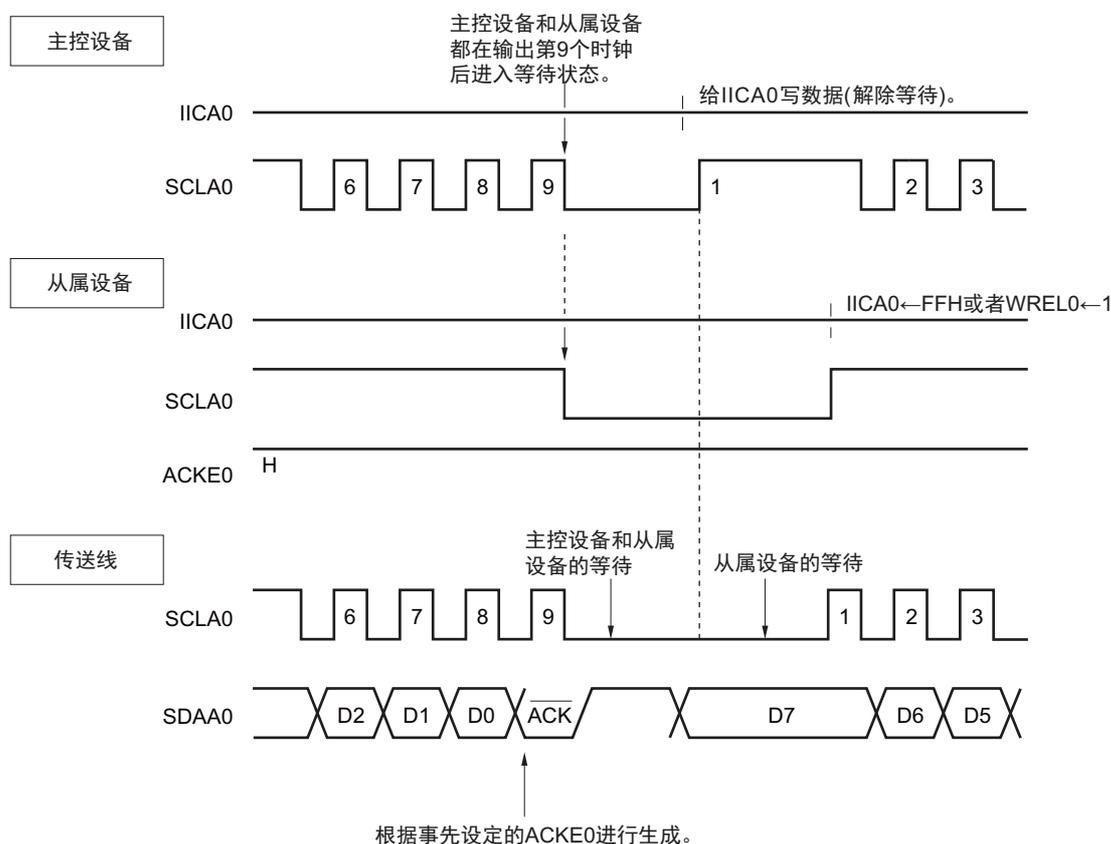


图 13-19 等待 (2/2)

- (2) 主控设备和从属设备都为 9 个时钟等待的情况
(主控设备: 发送, 从属设备: 接收, ACKE0=1)



备注 ACKE0: IICA 控制寄存器 00 (IICCTL00) 的 bit2
WRELO: IICA 控制寄存器 00 (IICCTL00) 的 bit5

通过设定 IICA 控制寄存器 00 (IICCTL00) 的 bit3 (WTIM0) 自动产生等待。

通常, 在接收方, 如果 IICCTL00 寄存器的 bit5 (WRELO) 为“1”或者给 IICA 移位寄存器 0 (IICA0) 写“FFH”, 就解除等待; 在发送方, 如果给 IICA0 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 IICCTL00 寄存器的 bit1 (STT0) 置“1”。
- 将 IICCTL00 寄存器的 bit0 (SPT0) 置“1”。

13.5.7 等待的解除方法

在一般情况下，I²C 通过以下的处理来解除等待。

- 给 IICA 移位寄存器 0 (IICA0) 写数据。
- 将 IICA 控制寄存器 00 (IICCTL00) 的 bit5 (WREL0) 置位 (解除等待)。
- 将 IICCTL00 寄存器的 bit1 (STT0) 置位 (生成开始条件) 注。
- 将 IICCTL00 寄存器的 bit0 (SPT0) 置位 (生成停止条件) 注。

注 只限于 I²C 通信的主控设备。

如果执行了这些等待的解除处理，I²C 就解除等待，重新开始通信。

要在解除等待后发送数据 (包括地址) 时，必须给 IICA0 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，就必须将 IICCTL00 寄存器的 bit5 (WREL0) 置“1”。

要在解除等待后生成重新开始条件，就必须将 IICCTL00 寄存器的 bit1 (STT0) 置“1”。

要在解除等待后生成停止条件，就必须将 IICCTL00 寄存器的 bit0 (SPT0) 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 WREL0 位置“1”来解除等待后给 IICA0 寄存器写数据，SDAA0 线的变化时序与 IICA0 寄存器的写时序就可能发生冲突，导致将错误的值输出到 SDAA0 线。

除了这些处理以外，在中途中止通信的情况下，如果将 IICE0 位清“0”，就停止通信，因此能解除等待。

在 I²C 总线状态因噪声而被死锁的情况下，如果将 IICCTL00 寄存器的 bit6 (LREL0) 置“1”，就退出通信，因此能解除等待。

注意 如果在 WUP0 位为“1”时执行等待的解除处理，就不解除等待。

13.5.8 中断请求（INTIICA0）的产生时序和等待控制

通过设定 IICA 控制寄存器 00（IICCTL00）的 bit3（WTIM0），在表 13-2 所示的时序产生 INTIICA0 并且进行等待控制。

表 13-2 INTIICA0 的产生时序和等待控制

WTIM0	从属设备			主控设备		
	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 0（SVA0）的设定值相同时，从属设备才在第 9 个时钟的下降沿产生 INTIICA0 信号并且进入等待状态。

此时，与 IICCTL00 寄存器 bit2（ACKE0）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 INTIICA0。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 INTIICA0，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 0（SVA0）的内容不同并且未接收到扩展码，就不产生 INTIICA0 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

- 从属设备：与 WTIM0 位无关，通过上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控设备：与 WTIM0 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

- 主控设备/从属设备：通过 WTIM0 位决定中断和等待的时序。

(3) 数据的发送

- 主控设备/从属设备：通过 WTIM0 位决定中断和等待的时序。

(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 0 (IICA0) 写数据。
- 将 IICA 控制寄存器 00 (IICCTL00) 的 bit5 (WREL0) 置位 (解除等待)。
- 将 IICCTL00 寄存器的 bit1 (STT0) 置位 (生成开始条件) 注。
- 将 IICCTL00 寄存器的 bit0 (SPT0) 置位 (生成停止条件) 注。

注 只限于主控设备。

当选择 8 个时钟等待 (WTIM0=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 INTIICA0 (只限于 SPIE0=1)。

13.5.9 地址匹配的检测方法

在 I²C 总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。

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 0 (SVA0) 的设定地址相同或者接收到扩展码时，产生 INTIICA0 中断请求。

13.5.10 错误的检测

在 I²C 总线模式中，因为发送过程中的串行数据总线 (SDAA0) 的状态被取到发送器件的 IICA 移位寄存器 0 (IICA0)，所以能通过将开始发送前和发送结束的 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。

13.5.11 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（EXC0）置“1”，并且在第8个时钟的下降沿产生中断请求（INTIICA0）。
不影响保存在从属地址寄存器0（SVA0）的本地站地址。
- (2) 当SVA0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INTIICA0。
 - 高4位数据相同：EXC0=1
 - 7位数据相同：COI0=1

备注 EXC0: IICA 状态寄存器 0 (IICS0) 的 bit5
COI0: IICA 状态寄存器 0 (IICS0) 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。
如果在从属设备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 00（IICCTL00）的 bit6（LREL0）置“1”，进入下次通信的待机状态。

表 13-3 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 有关上述以外的扩展码，请参照 NXP 公司发行的 I²C 总线规格书。

13.5.12 仲裁

当多个主控设备同时生成开始条件时（在 STD0 位变为“1”前将 STT0 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 0（IICS0）的仲裁失败标志（ALD0）置“1”，并且将 SCLA0 线和 SDAA0 线都置为高阻抗状态，以释放总线。

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 ALD0 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“13.5.8 中断请求（INTIICA0）的产生时序和等待控制”。

备注 STD0: IICA 状态寄存器 0（IICS0）的 bit1
STT0: IICA 控制寄存器 00（IICCTL00）的 bit1

图 13-20 仲裁时序例子

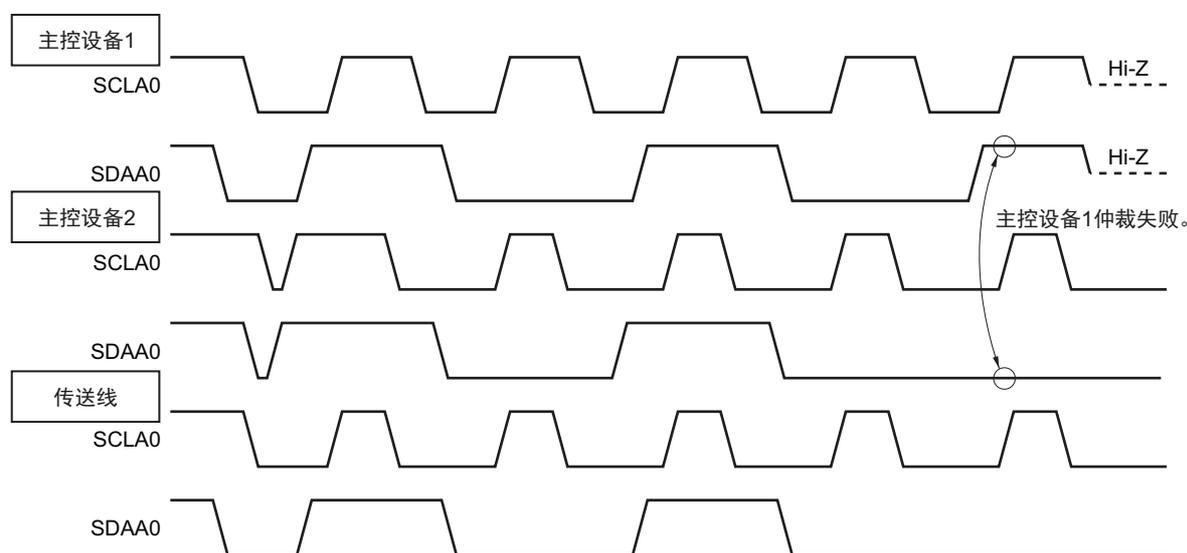


表 13-4 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时 (SPIE0=1) ^{注 2}
要生成重新开始条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
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 (SPIE0=1) ^{注 2}
要生成停止条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
要生成重新开始条件, 但是 SCLA0 为低电平。	

注 1. 当 WTIM0 位 (IICA 控制寄存器 00 (IICCTL00) 的 bit3) 为“1”时, 在第 9 个时钟的下降沿产生中断请求。当 WTIM0 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, 必须在主控设备时将 SPIE0 位置“1”。

备注 SPIE0: IICA 控制寄存器 00 (IICCTL00) 的 bit4

13.5.13 唤醒功能

这是 I²C 的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号（INTIICA0）的功能。

在地址不同的情况下不产生不需要的 INTIICA0 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 WUP0 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（INTIICA0）。在产生此中断后，通过指令将 WUP0 位清“0”，返回通常运行。

将 WUP0 位置“1”时的流程如图 13-21 所示，通过地址匹配将 WUP0 位置“0”时的流程如图 13-22 所示。

图 13-21 将 WUP0 位置“1”时的流程

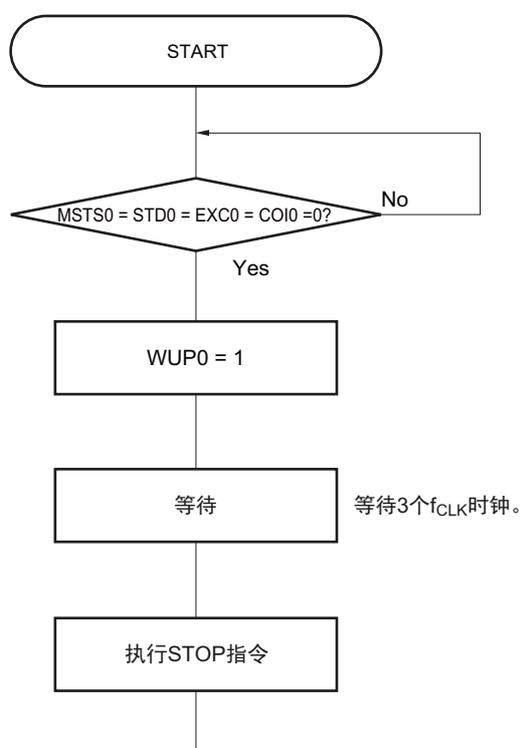
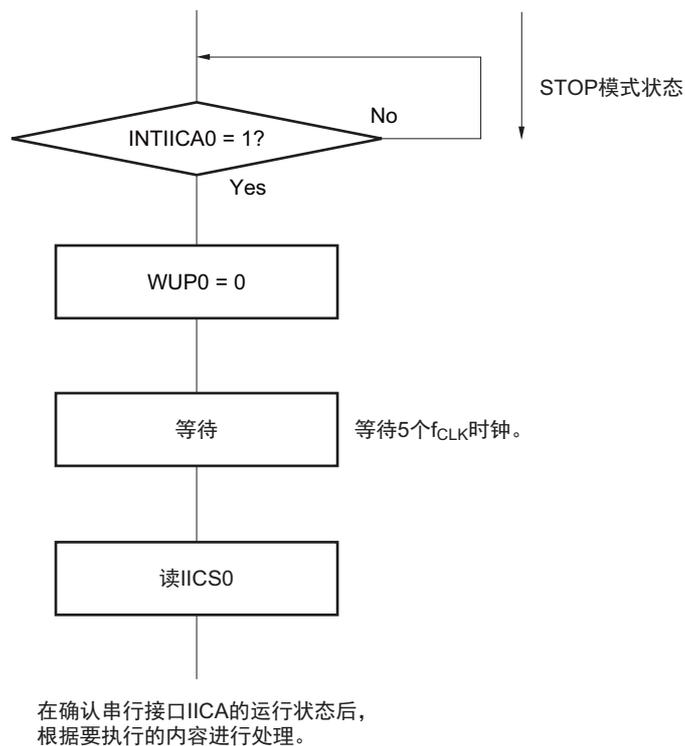


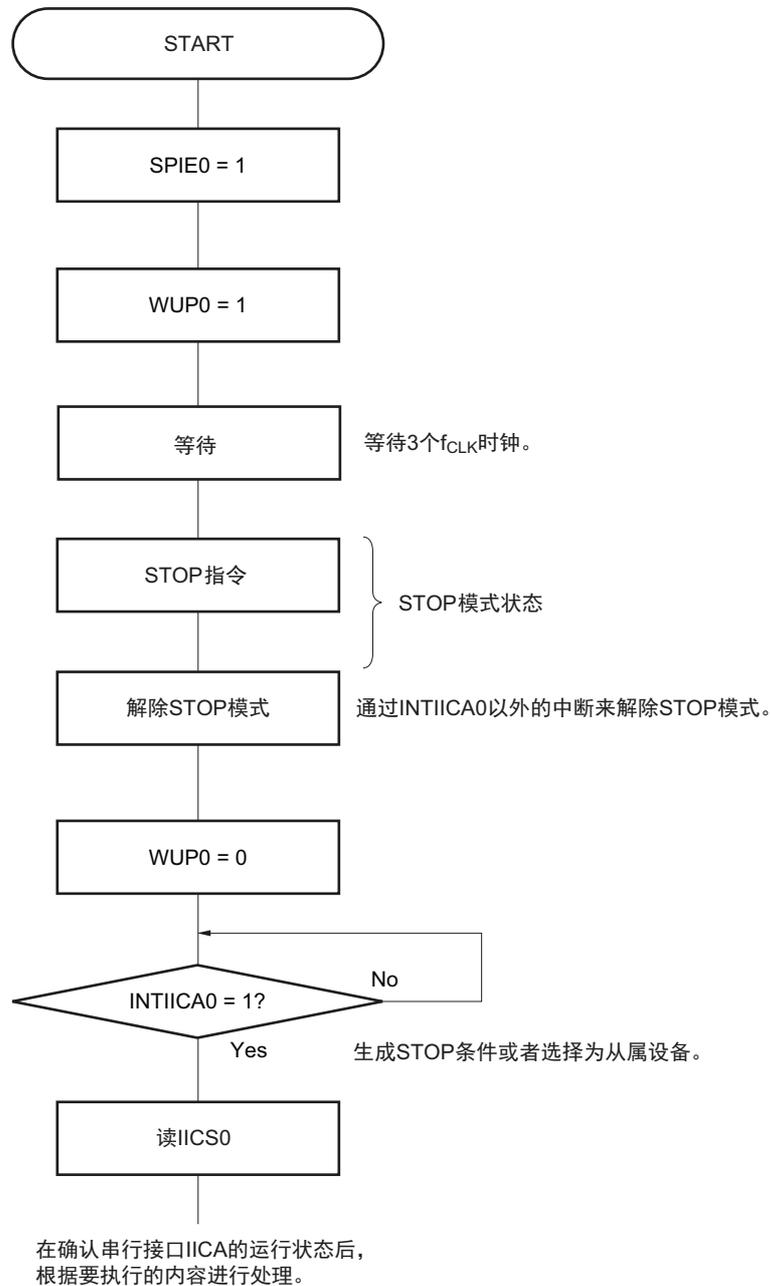
图 13-22 通过地址匹配将 WUP0 位置“0”时的流程（包括接收扩展码）



除了串行接口 IICA 产生的中断请求（INTIICA0）以外，必须通过以下的流程解除 STOP 模式。

- 下一个 IIC 通信作为主控设备运行的情况：图 13-23 的流程
- 下一个 IIC 通信作为从属设备运行的情况：
通过 INTIICA0 中断返回时：和图 13-22 的流程相同。
通过 INTIICA0 中断以外的中断返回时：继续 WUP0 为“1”时的运行直到发生 INTIICA0 中断。

图 13-23 在通过 INTIICA0 以外的方法解除 STOP 模式后作为主控设备运行的情况



13.5.14 通信预约

(1) 允许通信预约功能的情况 (IICA 标志寄存器 0 (IICF0) 的 bit0 (IICRSV0) =0)

要在不加入总线的状态下进行下次的主控通信时，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 00 (IICCTL00) 的 bit6 (LREL0) 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 IICCTL00 寄存器的 bit1 (STT0) 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 IICCTL00 寄存器的 bit4 (SPIE0) 置“1”，在通过产生的中断请求信号 (INTIICA0) 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 0 (IICA0) 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 IICA0 寄存器写的的数据无效。

当将 STT0 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 STT0 位置“1”并且经过等待时间后，通过 MSTS0 位 (IICA 状态寄存器 0 (IICS0) 的 bit7) 确认是否作为通信预约运行。

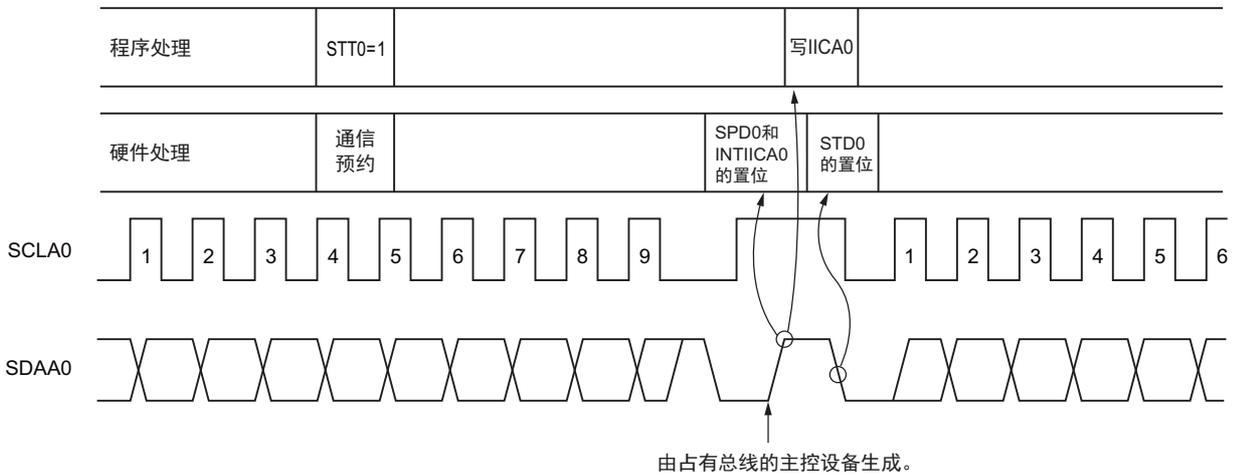
必须通过软件确保以下计算式计算的等待时间：

从将 STT0 位置“1”到确认 MSTS0 标志为止的等待时间 (f_{CLK} 的时钟数)：
 $(IICWL0 \text{ 的设定值} + IICWH0 \text{ 的设定值} + 4) + t_F \times 2 \times f_{CLK}[\text{时钟}]$

备注	IICWL0:	IICA 低电平宽度设定寄存器 0
	IICWH0:	IICA 高电平宽度设定寄存器 0
	t_F :	SDAA0 信号和 SCLA0 信号的下降时间
	f_{CLK} :	CPU/ 外围硬件的时钟频率

通信预约时序如图 13-24 所示。

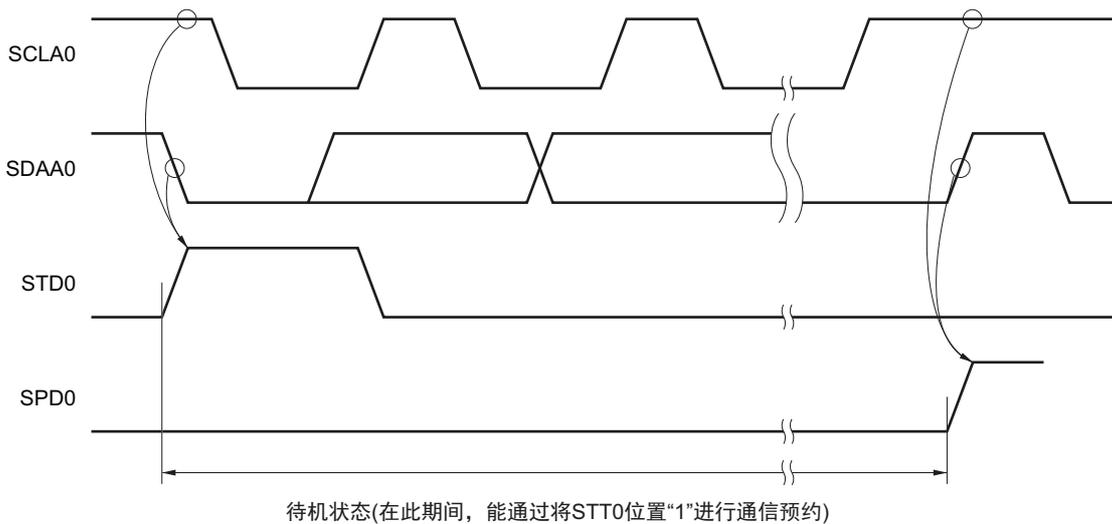
图 13-24 通信预约时序



- 备注 IICA0: IICA 移位寄存器 0
 STT0: IICA 控制寄存器 00 (IICCTL00) 的 bit1
 STD0: IICA 状态寄存器 0 (IICS0) 的 bit1
 SPD0: IICA 状态寄存器 0 (IICS0) 的 bit0

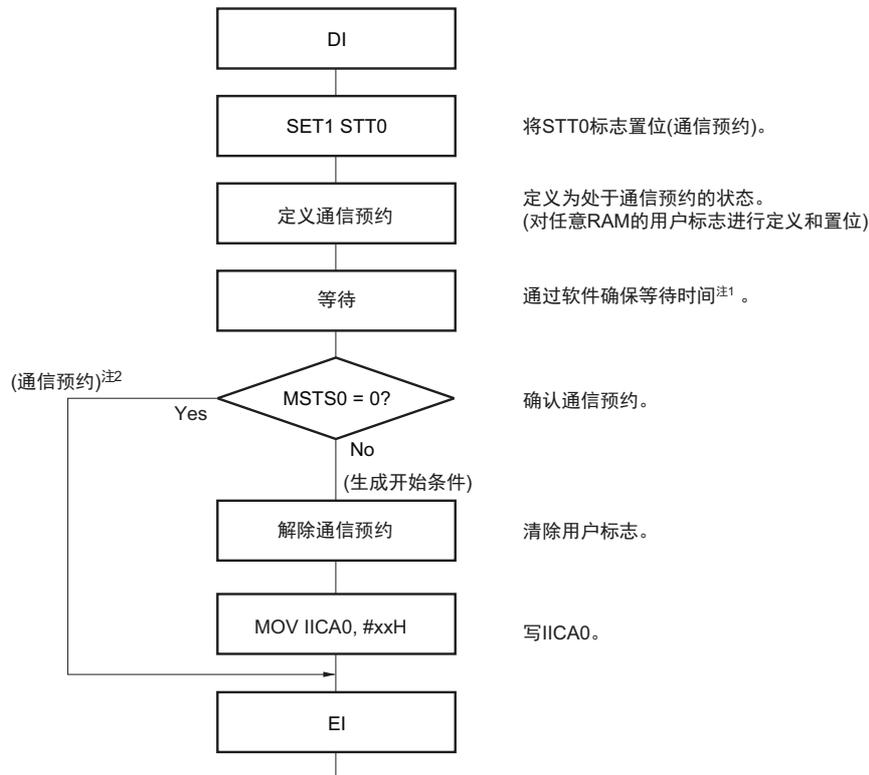
通过图 13-25 所示的时序接受通信预约。在 IICA 状态寄存器 0 (IICS0) 的 bit1 (STD0) 变为“1”后并且在检测到停止条件前，将 IICA 控制寄存器 00 (IICCTL00) 的 bit1 (STT0) 置“1”进行通信预约。

图 13-25 通信预约的接受时序



通信预约步骤如图 13-26 所示。

图 13-26 通信预约步骤



- 注 1. 等待时间（ f_{CLK} 的时钟数）如下：
 $(IICWLO \text{ 的设定值} + IICWH0 \text{ 的设定值} + 4) + t_F \times 2 \times f_{CLK}[\text{时钟}]$
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 0（IICA0）。

备注

STT0:	IICA 控制寄存器 00（IICCTL00）的 bit1
MSTS0:	IICA 状态寄存器 0（IICS0）的 bit7
IICA0:	IICA 移位寄存器 0
IICWLO:	IICA 低电平宽度设定寄存器 0
IICWH0:	IICA 高电平宽度设定寄存器 0
t_F :	SDAA0 信号和 SCLA0 信号的下降时间
f_{CLK} :	CPU/ 外围硬件的时钟频率

(2) 禁止通信预约功能的情况（IICA 标志寄存器 0（IICF0）的 bit0（IICRSV0）=1）

在总线通信过程中，如果不参加此通信的状态下将 IICA 控制寄存器 00（IICCTL00）的 bit1（STT0）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 IICCTL00 寄存器的 bit6（LREL0）置“1”，退出通信后释放了总线）。

通过 STCF0（IICF0 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 STT0 位为“1”到将 STCF0 位置“1”需要 5 个时钟的时间，所以必须通过软件确保此时间。

13.5.15 其他注意事项

(1) STCEN0 位为“0”的情况

在刚允许 I²C 运行 (IICE0=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSY0=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 01 (IICCTL01)。
- ② 将 IICA 控制寄存器 00 (IICCTL00) 的 bit7 (IICE0) 置“1”。
- ③ 将 IICCTL00 寄存器的 bit0 (SPT0) 置“1”。

(2) STCEN0 位为“1”的情况

在刚允许 I²C 运行 (IICE0=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSY0=0)。因此在生成第 1 个开始条件 (STT0=1) 时, 为了不破坏其他的通信, 需要确认总线已被释放。

(3) 正在和其他设备进行 I²C 通信的情况

在 SDAA0 引脚为低电平并且 SCLA0 引脚为高电平时, 如果允许 I²C 运行而且中途参加通信, I²C 的宏就视为 SDAA0 引脚从高电平变为低电平 (检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的 I²C 通信。为了避免这种情况, 必须按照以下顺序启动 I²C:

- ① 将 IICCTL00 寄存器的 bit4 (SPIE0) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIICA0)。
- ② 将 IICCTL00 寄存器的 bit7 (IICE0) 置“1”, 允许 I²C 运行。
- ③ 等待检测到开始条件。
- ④ 在返回应答前 (将 IICE0 位置“1”后的 4 ~ 72 个时钟内) 将 IICCTL00 寄存器的 bit6 (LREL0) 置“1”, 强制将检测置为无效。

(4) 在将 STT0 位和 SPT0 位 (IICCTL00 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 SPIE0 位 (IICCTL00 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 0 (IICA0) 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, 就因为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 MSTS0 位 (IICA 状态寄存器 0 (IICS0) 的 bit7) 时, 不需要将 SPIE0 位置“1”。

13.5.16 通信运行

在此通过流程图表示以下 3 个运行步骤。

(1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在 I²C 总线的多主控系统中，只根据 I²C 总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1 帧）为高电平，就将总线作为释放状态而参加通信。

此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

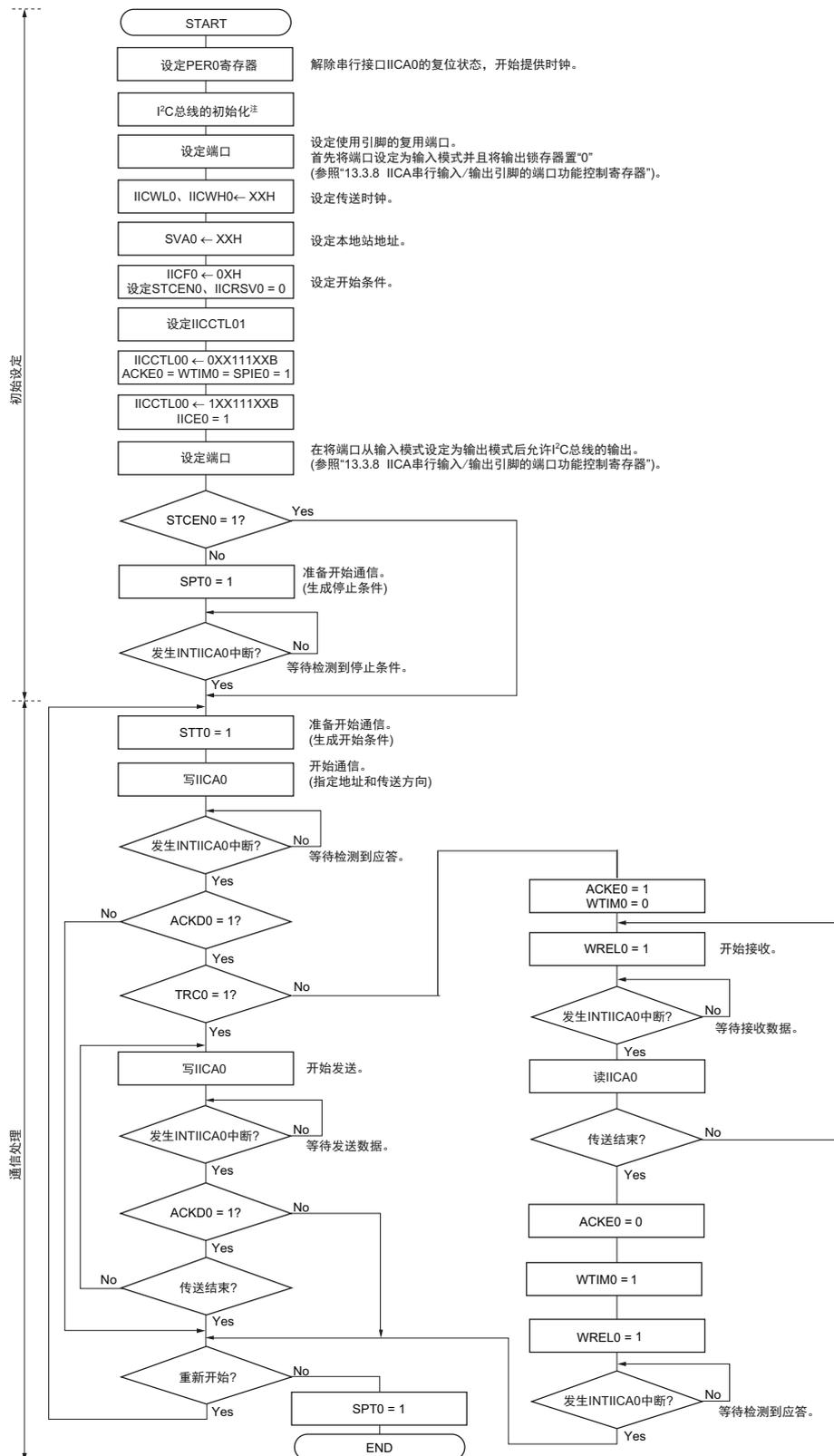
用作 I²C 总线从属设备的例子如下所示。

当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICA0 中断的发生。如果发生 INTIICA0 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，进行所要的“通信处理”。

(1) 单主控系统的主控运行

图 13-27 单主控系统的主控运行

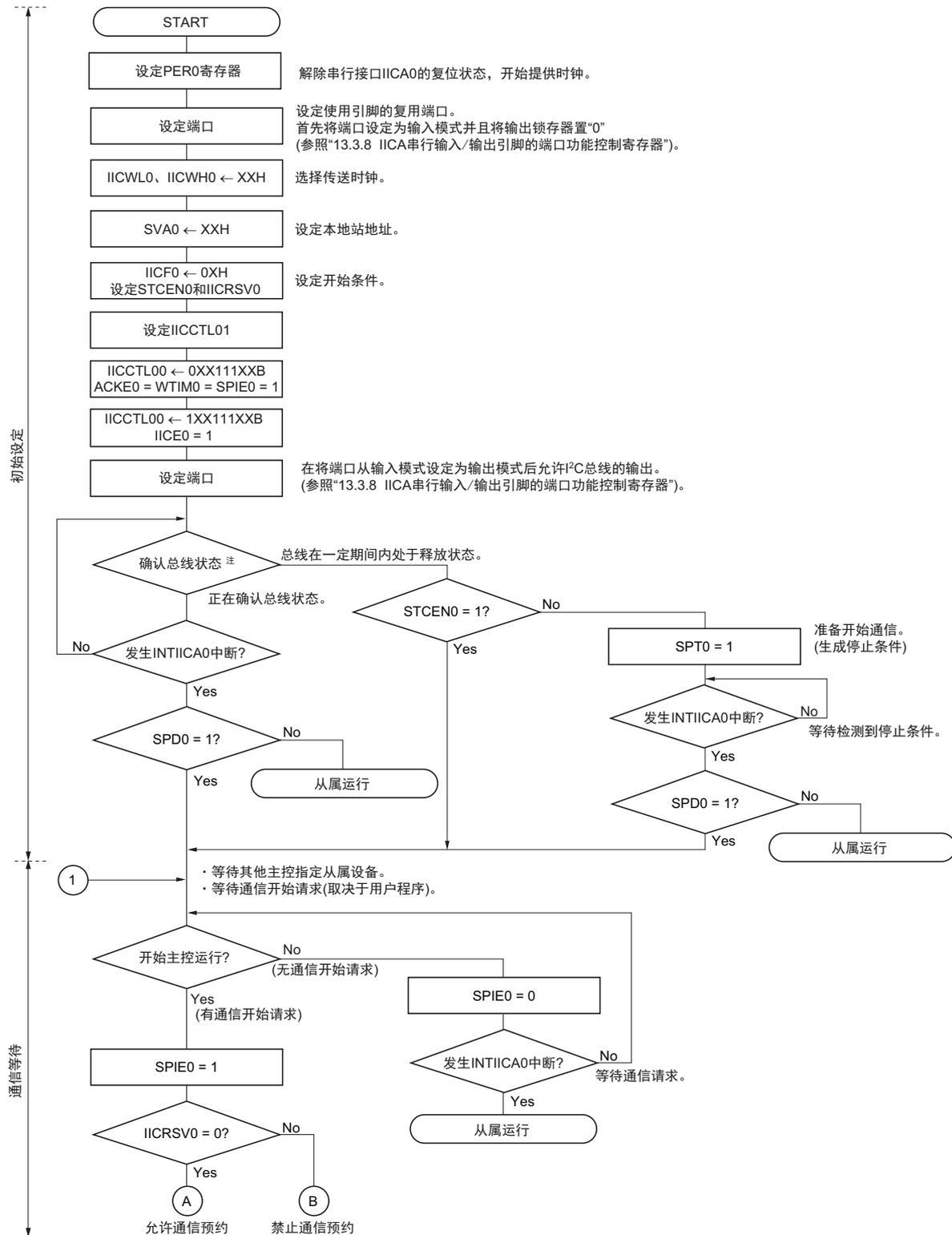


注 必须根据通信中的产品的规格，释放 I²C 总线（SCLA0 引脚和 SDA0 引脚为高电平）。例如，如果 EEPROM 处于给 SDA0 引脚输出低电平的状态，就必须将 SCLA0 引脚设定为输出端口，并且在 SDA0 引脚固定为高电平前从输出端口输出时钟脉冲。

备注 发送和接收格式必须符合通信中的产品的规格。

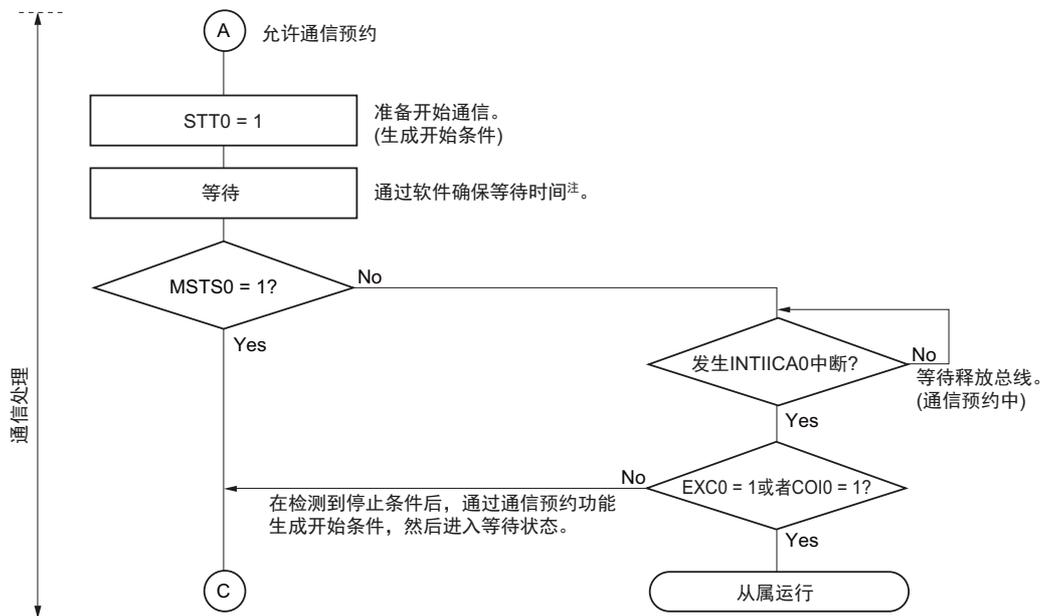
(2) 多主控系统的主控运行

图 13-28 多主控系统的主控运行 (1/3)

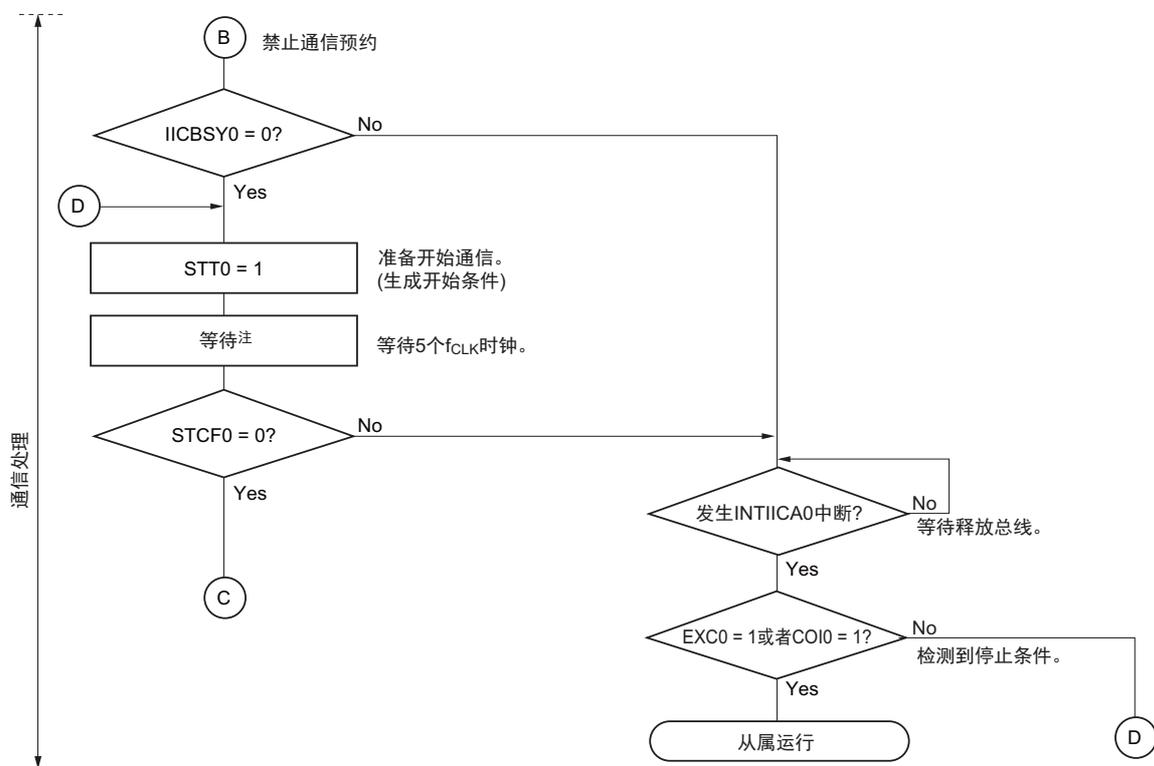


注 必须确认在一定时间内 (例如, 1 帧) 总线为释放状态 (CLD0 位 =1、DAD0 位 =1)。当 SDA0 引脚固定为低电平时, 必须根据通信中的产品的规格, 判断是否释放 I²C 总线 (SCLA0 引脚和 SDA0 引脚为高电平)。

图 13-28 多主控系统的主控运行 (2/3)

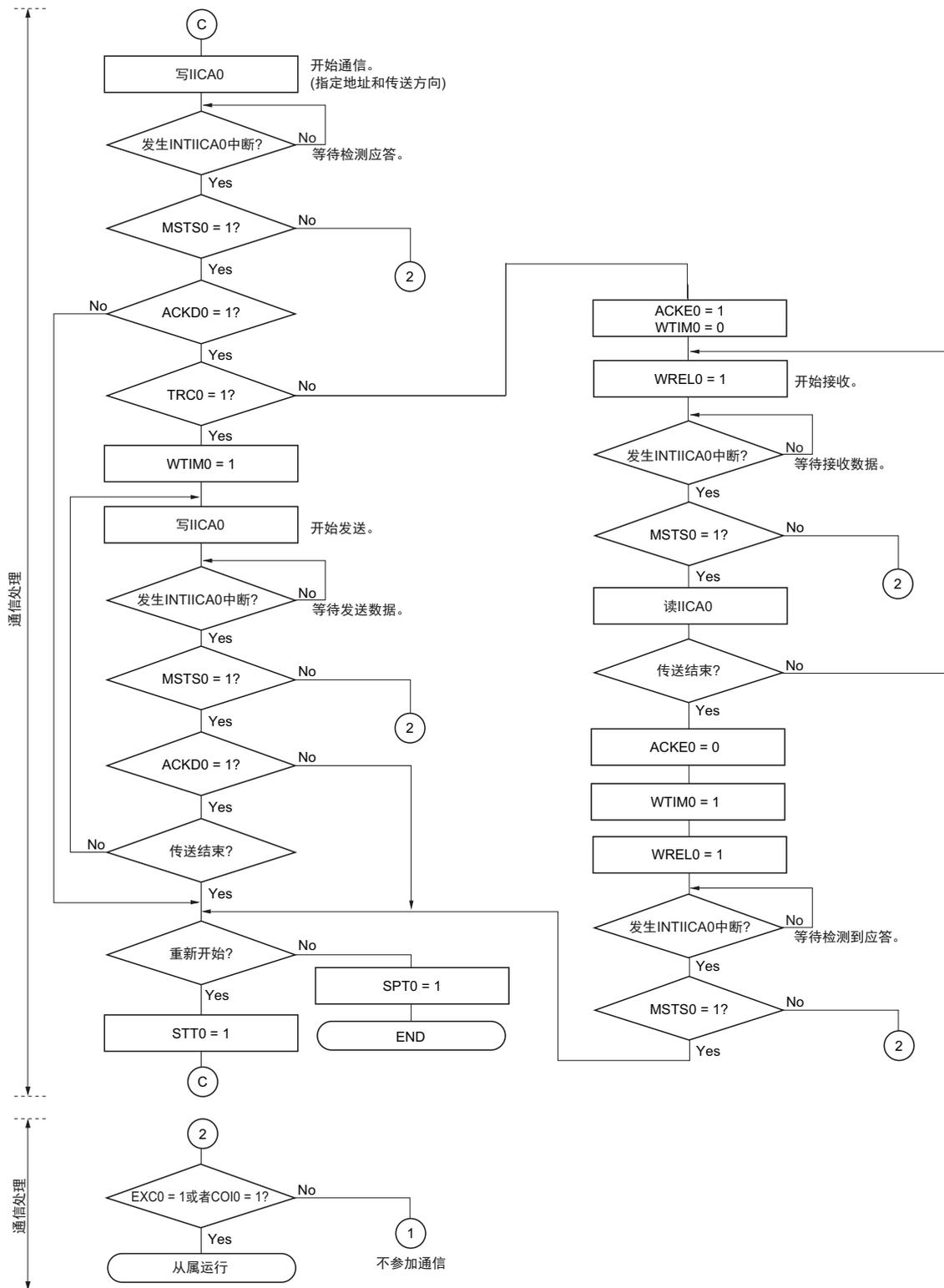


注 等待时间 (f_{CLK} 的时钟数) 如下:
 $(IICWL0$ 的设定值 + $IICWH0$ 的设定值 + 4 个时钟) f_{CLK} + $t_F \times 2$



备注 $IICWL0$: IICA 低电平宽度设定寄存器 n
 $IICWH0$: IICA 高电平宽度设定寄存器 n
 t_F : SDA0 信号和 SCLA0 信号的下降时间
 f_{CLK} : CPU/ 外围硬件的时钟频率

图 13-28 多主控系统的主控运行 (3/3)



备注 1. 传送和接收格式必须符合通信中的产品的规格。

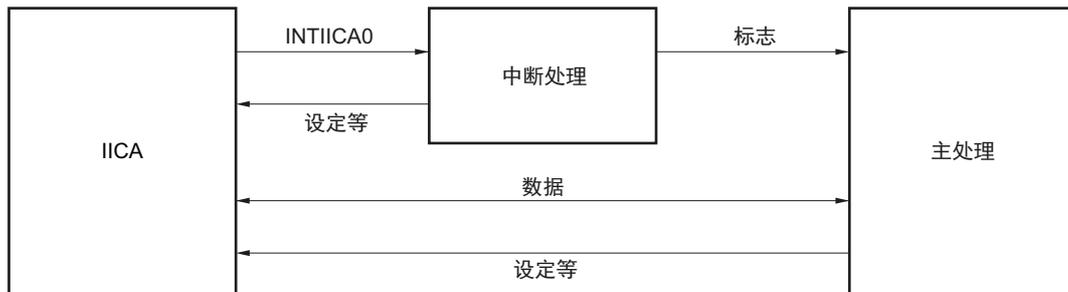
2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIICA0 中断时读 MSTS0 位，确认仲裁结果。
3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIICA0 中断时通过 IICA 状态寄存器 0 (IICS0) 和 IICA 标志寄存器 0 (IICF0) 确认状态，决定以后的处理。

(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 INTIICA0 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，INTIICA0 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 INTIICA0 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

此标志表示可进行数据通信。在通常的数据通信中，和 INTIICA0 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次的通信请求）。

③ 通信方向标志

此标志表示通信方向，和 TRC0 位的值相同。

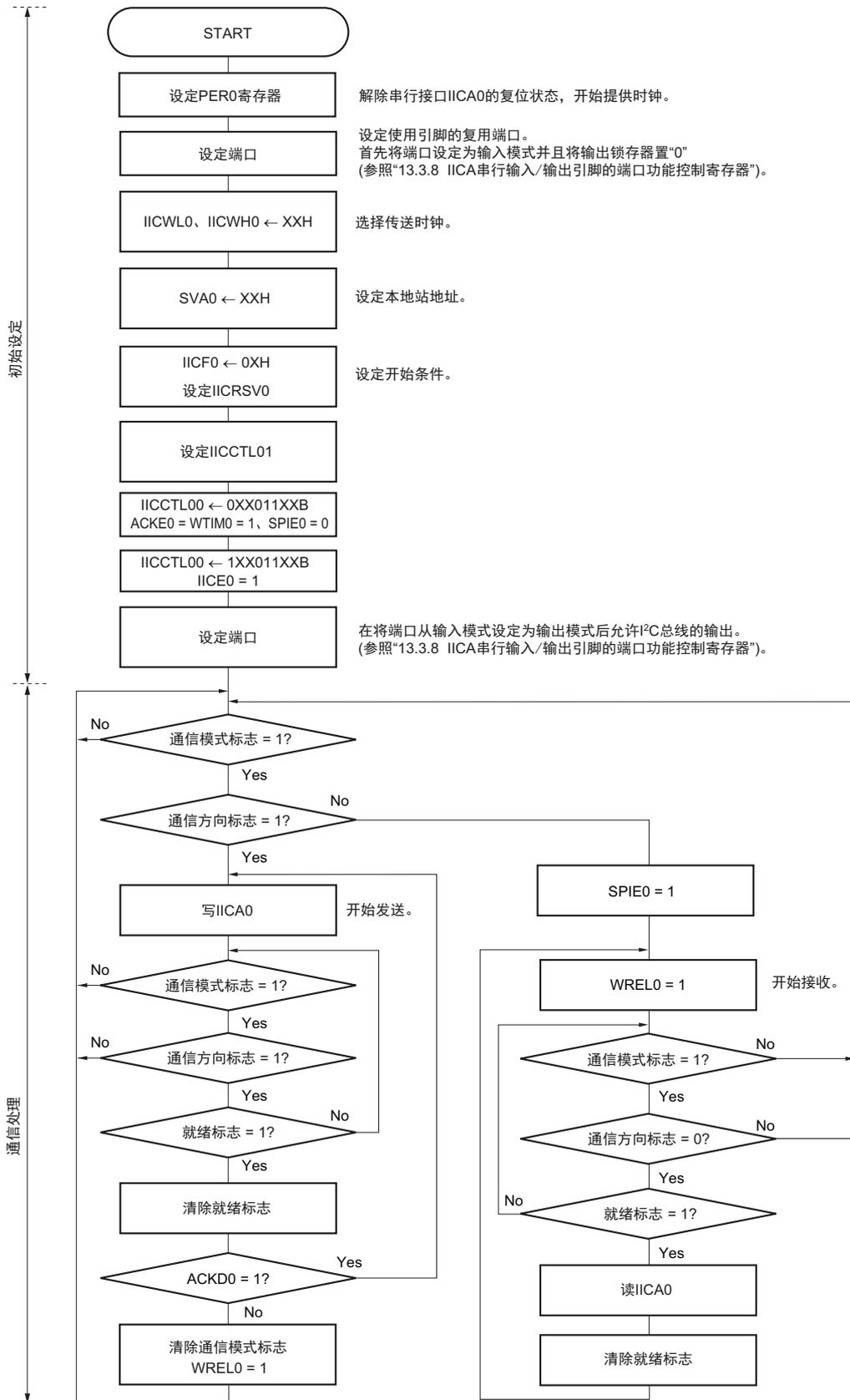
从属运行的主处理部的运行如下所示。

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（通过中断进行停止条件和开始条件的处理，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 13-29 从属运行步骤 (1)



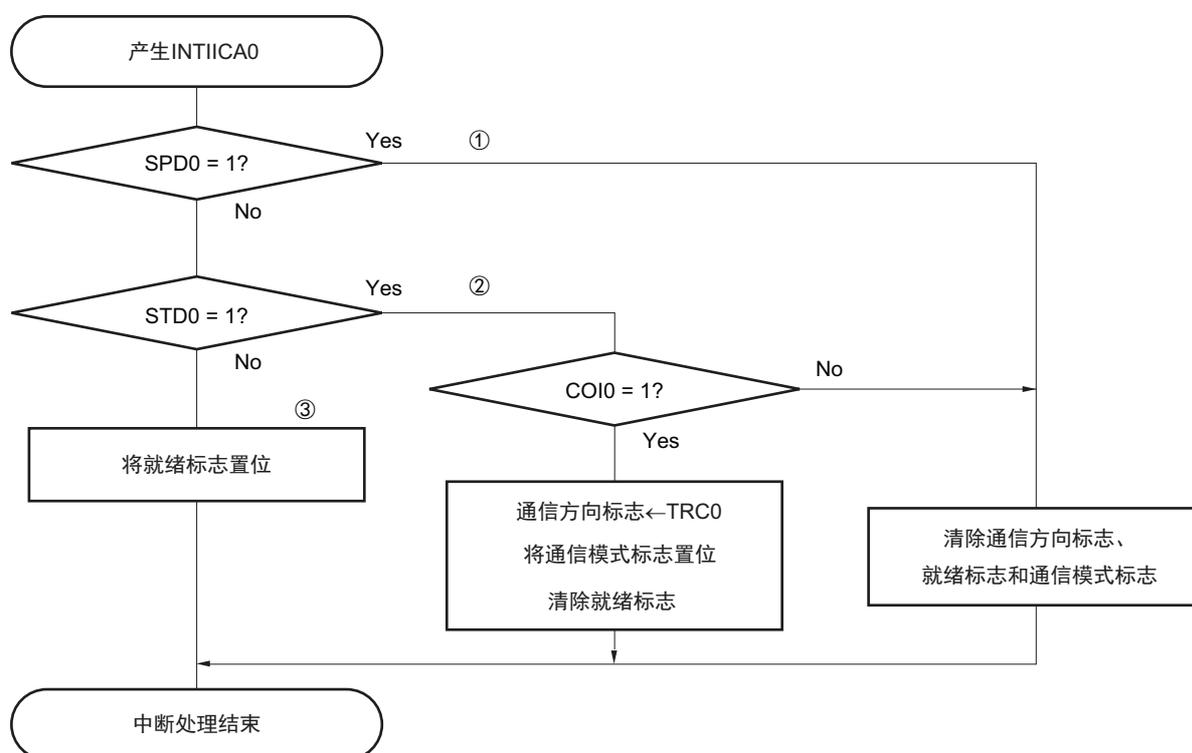
备注 传送和接收格式必须符合通信中的产品的规格。

从属设备通过 INTIICA0 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 INTIICA0 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，I²C 总线就保持等待状态并且从中断返回。

备注 上述的①~③对应“图 13-30 从属运行步骤 (2)”的①~③。

图 13-30 从属运行步骤 (2)



13.5.17 I²C 中断请求 (INTIICA0) 的产生时序

数据的发送和接收时序、INTIICA0 中断请求信号的产生时序以及产生 INTIICA0 信号时的 IICA 状态寄存器 0 (IICS0) 的值如下所示。

备注 ST: 开始条件
 AD6 ~ AD0: 地址
 R/W: 传送方向的指定
 ACK: 应答
 D7 ~ D0: 数据
 SP: 停止条件

(1) 主控运行

(a) Start ~ Address ~ Data ~ Data ~ Stop (发送和接收)

(i) WTIM0=0 的情况

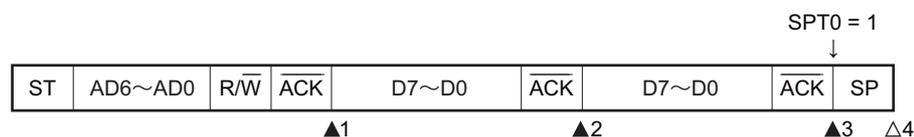


- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000×000B
- ▲ 3: IICS0=1000×000B (将 WTIM0 位置“1”)注
- ▲ 4: IICS0=1000××00B (将 SPT0 位置“1”)
- △ 5: IICS0=00000001B

注 为了生成停止条件，必须将 WTIM0 位置“1”并且更改 INTIICA0 中断请求信号的产生时序。

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1 的情况

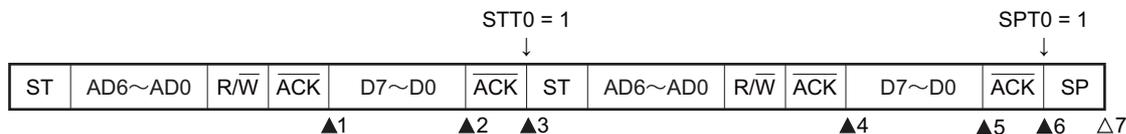


- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000×100B
- ▲ 3: IICS0=1000××00B (将 SPT0 位置“1”)
- △ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (重新开始)

(i) WTIM0=0的情况

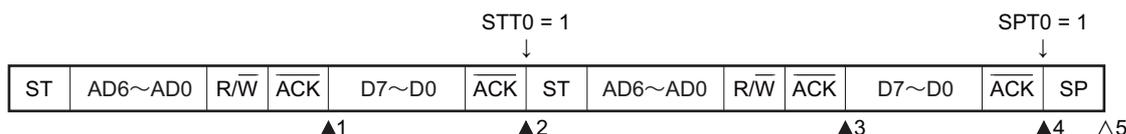


- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000×000B (将 WTIM0 位置“1”) 注 1
- ▲ 3: IICS0=1000××00B (将 WTIM0 位清“0”注 2 并且将 STT0 位置“1”)
- ▲ 4: IICS0=1000×110B
- ▲ 5: IICS0=1000×000B (将 WTIM0 位置“1”) 注 3
- ▲ 6: IICS0=1000××00B (将 SPT0 位置“1”)
- △ 7: IICS0=00000001B

- 注 1. 为了生成开始条件, 必须将 WTIM0 位置“1”并且更改 INTIICA0 中断请求信号的产生时序。
 注 2. 为了恢复原来的设定, 必须将 WTIM0 位清“0”。
 注 3. 为了生成停止条件, 必须将 WTIM0 位置“1”并且更改 INTIICA0 中断请求信号的产生时序。

备注 ▲ 一定产生
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1的情况



- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000××00B (将 STT0 位置“1”)
- ▲ 3: IICS0=1000×110B
- ▲ 4: IICS0=1000××00B (将 SPT0 位置“1”)
- △ 5: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(c) Start ~ Code ~ Data ~ Data ~ Stop (发送扩展码)

(i) WTIM0=0的情况

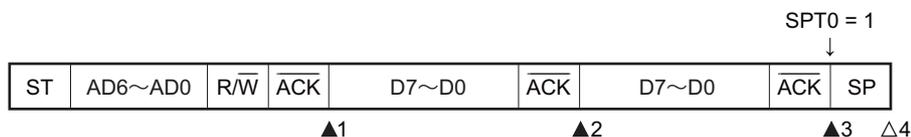


- ▲ 1: IICS0=1010×110B
- ▲ 2: IICS0=1010×000B
- ▲ 3: IICS0=1010×000B (将 WTIM0 位置“1”)注
- ▲ 4: IICS0=1010××00B (将 SPT0 位置“1”)
- △ 5: IICS0=00000001B

注 为了生成停止条件，必须将 WTIM0 位置“1”并且更改 INTIICA0 中断请求信号的产生时序。

- 备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1的情况



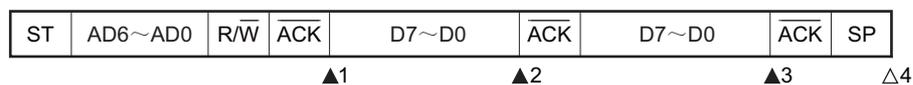
- ▲ 1: IICS0=1010×110B
- ▲ 2: IICS0=1010×100B
- ▲ 3: IICS0=1010××00B (将 SPT0 位置“1”)
- △ 4: IICS0=00001001B

- 备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(2) 从属运行（接收从属地址的情况）

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) WTIM0=0 的情况



▲ 1: IICS0=0001×110B

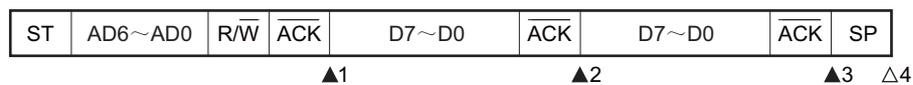
▲ 2: IICS0=0001×000B

▲ 3: IICS0=0001×000B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1 的情况



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001×100B

▲ 3: IICS0=0001××00B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后SVA0相同)



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001×000B

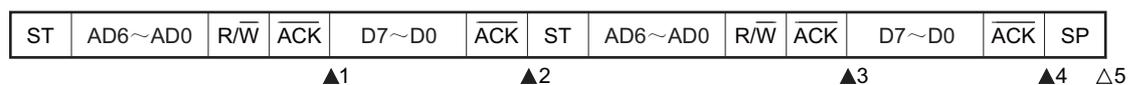
▲ 3: IICS0=0001×110B

▲ 4: IICS0=0001×000B

△ 5: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在SPIE0位为“1”时产生。
 × 任意

(ii) WTIM0=1的情况 (在重新开始后SVA0相同)



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001××00B

▲ 3: IICS0=0001×110B

▲ 4: IICS0=0001××00B

△ 5: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在SPIE0位为“1”时产生。
 × 任意

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIM0=0的情况（在重新开始后地址不同（扩展码））



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001×000B

▲ 3: IICS0=0010×010B

▲ 4: IICS0=0010×000B

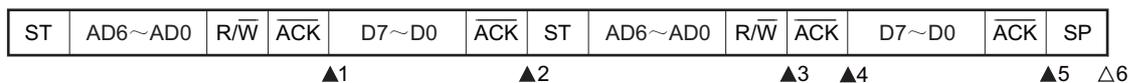
△ 5: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(ii) WTIM0=1的情况（在重新开始后地址不同（扩展码））



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001××00B

▲ 3: IICS0=0010×010B

▲ 4: IICS0=0010×110B

▲ 5: IICS0=0010××00B

△ 6: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后地址不同 (非扩展码))



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001×000B

▲ 3: IICS0=00000×10B

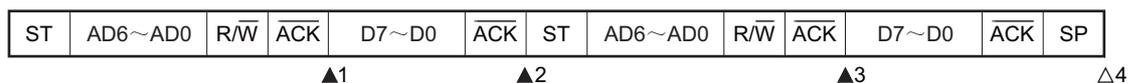
△ 4: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(ii) WTIM0=1的情况 (在重新开始后地址不同 (非扩展码))



▲ 1: IICS0=0001×110B

▲ 2: IICS0=0001××00B

▲ 3: IICS0=00000×10B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

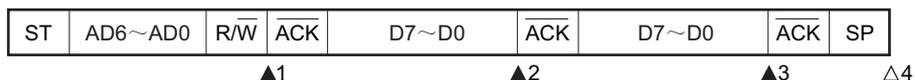
× 任意

(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) WTIM0=0的情况



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×000B

▲ 3: IICS0=0010×000B

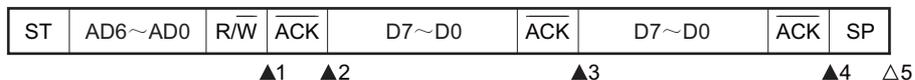
△ 4: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(ii) WTIM0=1的情况



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×110B

▲ 3: IICS0=0010×100B

▲ 4: IICS0=0010××00B

△ 5: IICS0=00000001B

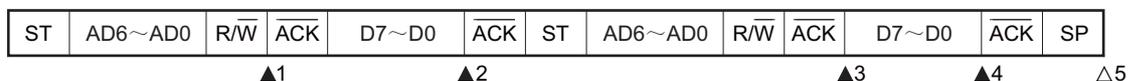
备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后SVA0相同)



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×000B

▲ 3: IICS0=0001×110B

▲ 4: IICS0=0001×000B

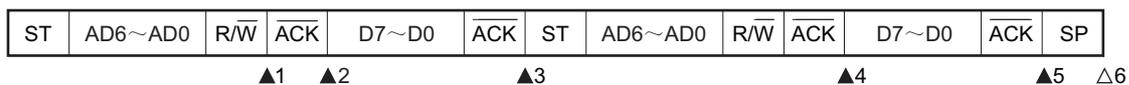
△ 5: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(ii) WTIM0=1的情况 (在重新开始后SVA0相同)



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×110B

▲ 3: IICS0=0010××00B

▲ 4: IICS0=0001×110B

▲ 5: IICS0=0001××00B

△ 6: IICS0=00000001B

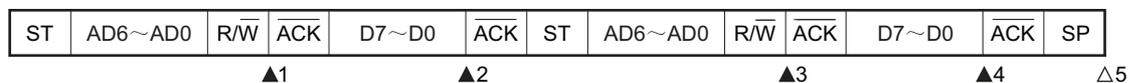
备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIM0=0的情况（在重新开始后接收扩展码）



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×000B

▲ 3: IICS0=0010×010B

▲ 4: IICS0=0010×000B

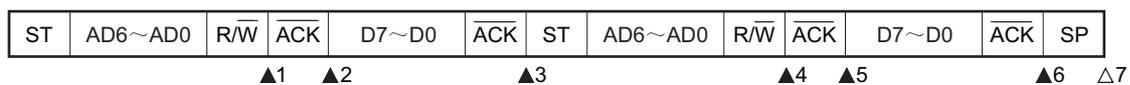
△ 5: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(ii) WTIM0=1的情况（在重新开始后接收扩展码）



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×110B

▲ 3: IICS0=0010××00B

▲ 4: IICS0=0010×010B

▲ 5: IICS0=0010×110B

▲ 6: IICS0=0010××00B

△ 7: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在SPIE0位为“1”时产生。

× 任意

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIM0=0的情况 (在重新开始后地址不同 (非扩展码))



▲ 1: IICS0=0010×010B

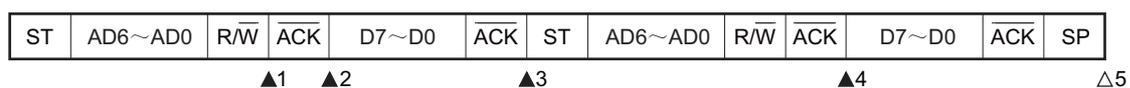
▲ 2: IICS0=0010×000B

▲ 3: IICS0=00000×10B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在SPIE0位为“1”时产生。
 × 任意

(ii) WTIM0=1的情况 (在重新开始后地址不同 (非扩展码))



▲ 1: IICS0=0010×010B

▲ 2: IICS0=0010×110B

▲ 3: IICS0=0010××00B

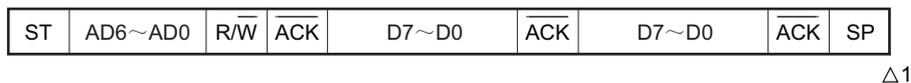
▲ 4: IICS0=00000×10B

△ 5: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在SPIE0位为“1”时产生。
 × 任意

(4) 不参加通信的运行

(a) Start ~ Code ~ Data ~ Data ~ Stop



△ 1: IICS0=00000001B

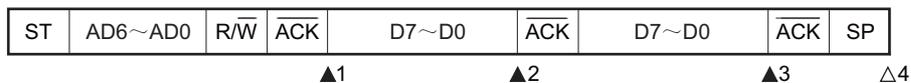
备注 △: 只在 SPIE0 位为“1”时产生。

(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 INTIICA0 中断请求信号时读 MSTS0 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) WTIM0=0 的情况



▲ 1: IICS0=0101×110B

▲ 2: IICS0=0001×000B

▲ 3: IICS0=0001×000B

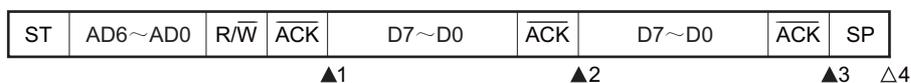
△ 4: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在 SPIE0 位为“1”时产生。

× 任意

(ii) WTIM0=1 的情况



▲ 1: IICS0=0101×110B

▲ 2: IICS0=0001×100B

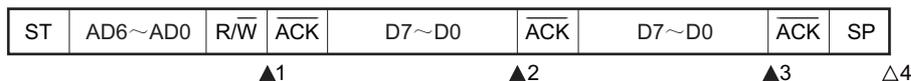
▲ 3: IICS0=0001××00B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 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) WTIM0=0 的情况



▲ 1: IICS0=0110×010B

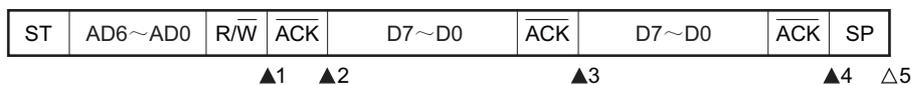
▲ 2: IICS0=0010×000B

▲ 3: IICS0=0010×000B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1 的情况



▲ 1: IICS0=0110×010B

▲ 2: IICS0=0010×110B

▲ 3: IICS0=0010×100B

▲ 4: IICS0=0010××00B

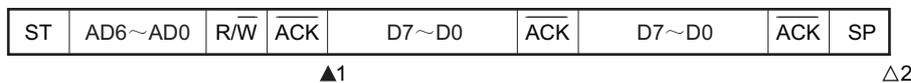
△ 5: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 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 INTIICA0 中断请求信号时读 MSTS0 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（WTIM0=1）

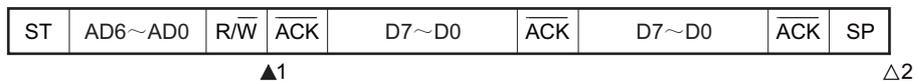


▲ 1: IICS0=01000110B

△ 2: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。

(b) 在发送扩展码的过程中仲裁失败的情况

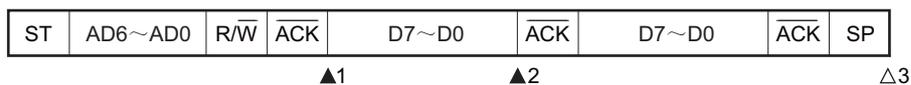


▲ 1: IICS0=0110×010B
 通过软件将 LREL0 位置“1”。
 △ 2: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(c) 在发送数据的过程中仲裁失败的情况

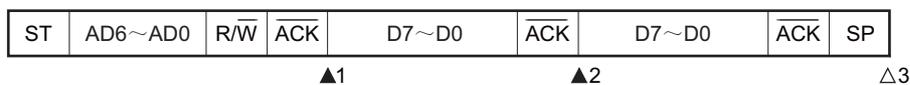
(i) WTIM0=0 的情况



▲ 1: IICS0=10001110B
 ▲ 2: IICS0=01000000B
 △ 3: IICS0=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。

(ii) WTIM0=1



▲ 1: IICS0=10001110B

▲ 2: IICS0=01000100B

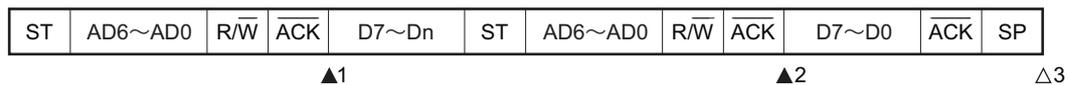
△ 3: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在 SPIE0 位为“1”时产生。

(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, SVA0 不同)



▲ 1: IICS0=1000×110B

▲ 2: IICS0=01000110B

△ 3: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在 SPIE0 位为“1”时产生。

× 任意

n=6~0

(ii) 扩展码



▲ 1: IICS0=1000×110B

▲ 2: IICS0=01100010B

通过软件将 LREL0 位置“1”。

△ 3: IICS0=00000001B

- 备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意
 n=6~0

(e) 在传送数据时因停止条件而仲裁失败的情况



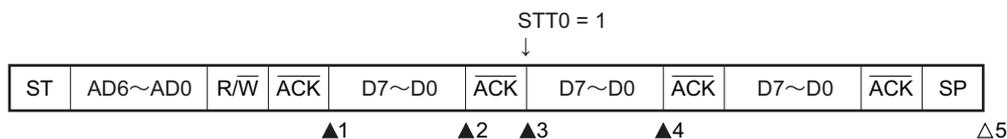
▲ 1: IICS0=10000110B

△ 2: IICS0=01000001B

- 备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意
 n=6~0

(f) 在想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) WTIM0=0的情况



▲ 1: IICS0=1000×110B

▲ 2: IICS0=1000×000B (将 WTIM0 位置“1”)

▲ 3: IICS0=1000×100B (将 WTIM0 位清“0”)

▲ 4: IICS0=01000000B

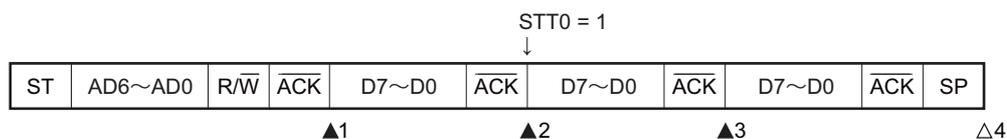
△ 5: IICS0=00000001B

备注 ▲ 一定产生。

△ 只在 SPIE0 位为“1”时产生。

× 任意

(ii) WTIM0=1的情况



▲ 1: IICS0=1000×110B

▲ 2: IICS0=1000×100B (将 STT0 位置“1”)

▲ 3: IICS0=01000100B

△ 4: IICS0=00000001B

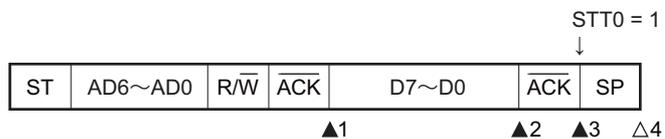
备注 ▲ 一定产生。

△ 只在 SPIE0 位为“1”时产生。

× 任意

(g) 在想生成重新开始条件时因停止条件而仲裁失败的情况

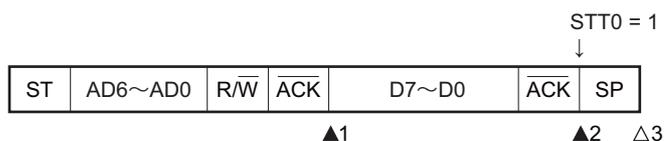
(i) WTIM0=0的情况



- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000×000B (将 WTIM0 位置“1”)
- ▲ 3: IICS0=1000××00B (将 STT0 位置“1”)
- △ 4: IICS0=01000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(ii) WTIM0=1的情况

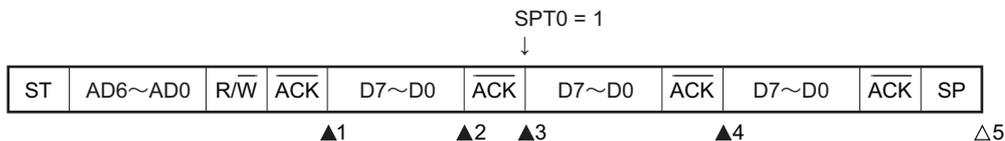


- ▲ 1: IICS0=1000×110B
- ▲ 2: IICS0=1000××00B (将 STT0 位置“1”)
- △ 3: IICS0=01000001B

备注 ▲ 一定产生。
 △ 只在 SPIE0 位为“1”时产生。
 × 任意

(h) 在想生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIM0=0的情况



▲ 1: IICS0=1000×110B

▲ 2: IICS0=1000×000B (将 WTIM0 位置“1”)

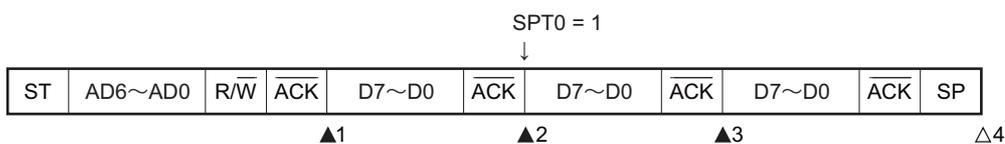
▲ 3: IICS0=1000×100B (将 WTIM0 位清“0”)

▲ 4: IICS0=01000100B

△ 5: IICS0=00000001B

备注 ▲ 一定产生。
△ 只在 SPIE0 位为“1”时产生。
× 任意

(ii) WTIM0=1的情况



▲ 1: IICS0=1000×110B

▲ 2: IICS0=1000×100B (将 SPT0 位置“1”)

▲ 3: IICS0=01000100B

△ 4: IICS0=00000001B

备注 ▲ 一定产生。
△ 只在 SPIE0 位为“1”时产生。
× 任意

13.6 时序图

在 I²C 总线模式中，主控设备通过给串行总线输出地址，从多个从属设备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 TRC0 位（IICA 状态寄存器 0（IICS0）的 bit3），开始与从属设备进行串行通信。

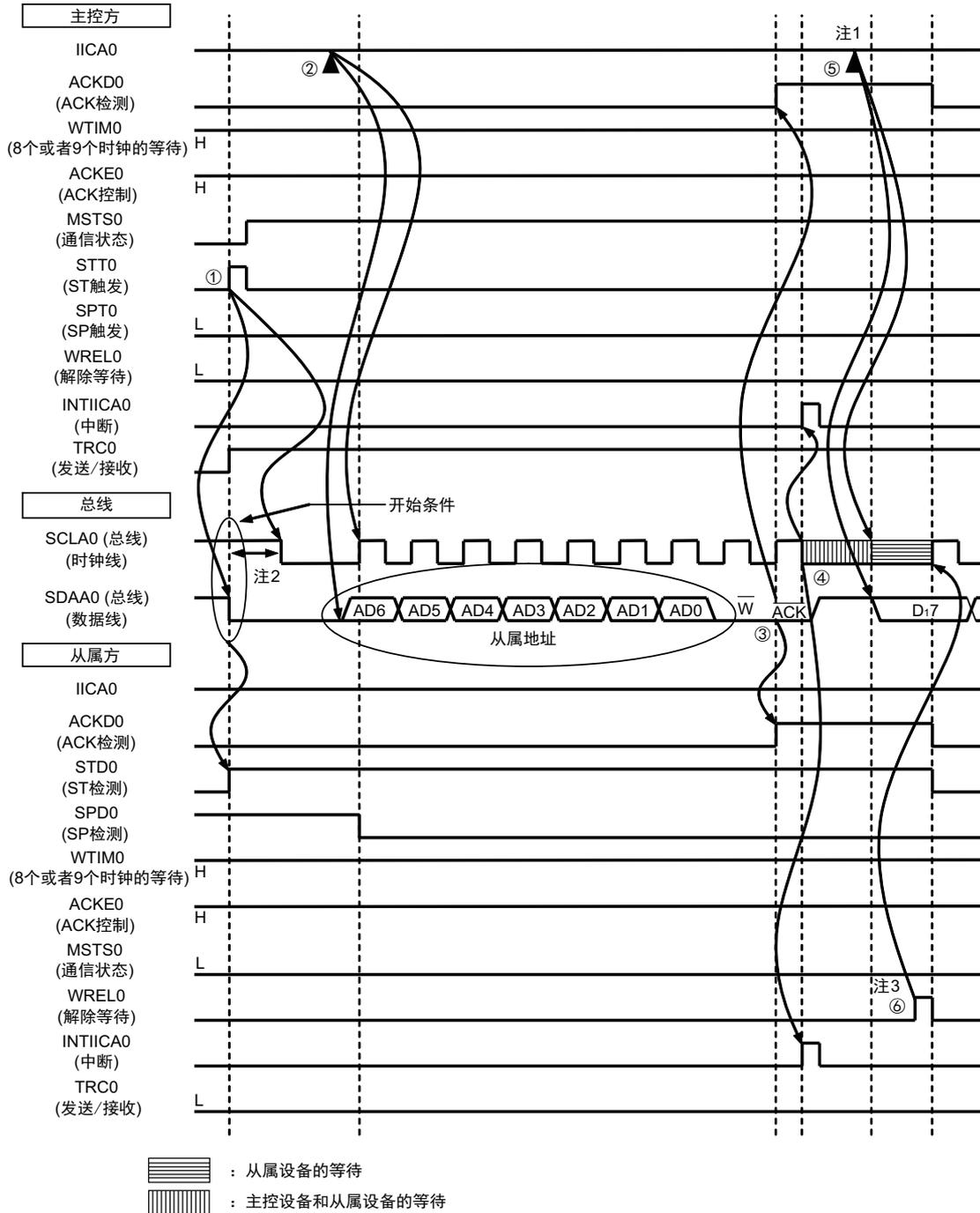
数据通信的时序图如图 13-31 和图 13-32 所示。

与串行时钟（SCLA0）的下降沿同步进行 IICA 移位寄存器 0（IICA0）的移位，并且将发送数据传送到 SO 锁存器，以 MSB 优先从 SDAA0 引脚输出数据。

在 SCLA0 的上升沿将 SDAA0 引脚输入的数据取到 IICA0。

图 13-31 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (1/4)

(1) 开始条件~地址~数据



- 注 1. 要在主控方的发送期间解除等待时，必须给 IICA0 写数据而不是将 WREL0 位置位。
 注 2. 从 SDA0 引脚信号下降到 SCLA0 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。
 注 3. 要在从属方的接收期间解除等待时，必须将 IICA0 置“FFH”或者将 WREL0 位置位。

图 13-31 的“(1) 开始条件~地址~数据”的①~⑥的说明如下：

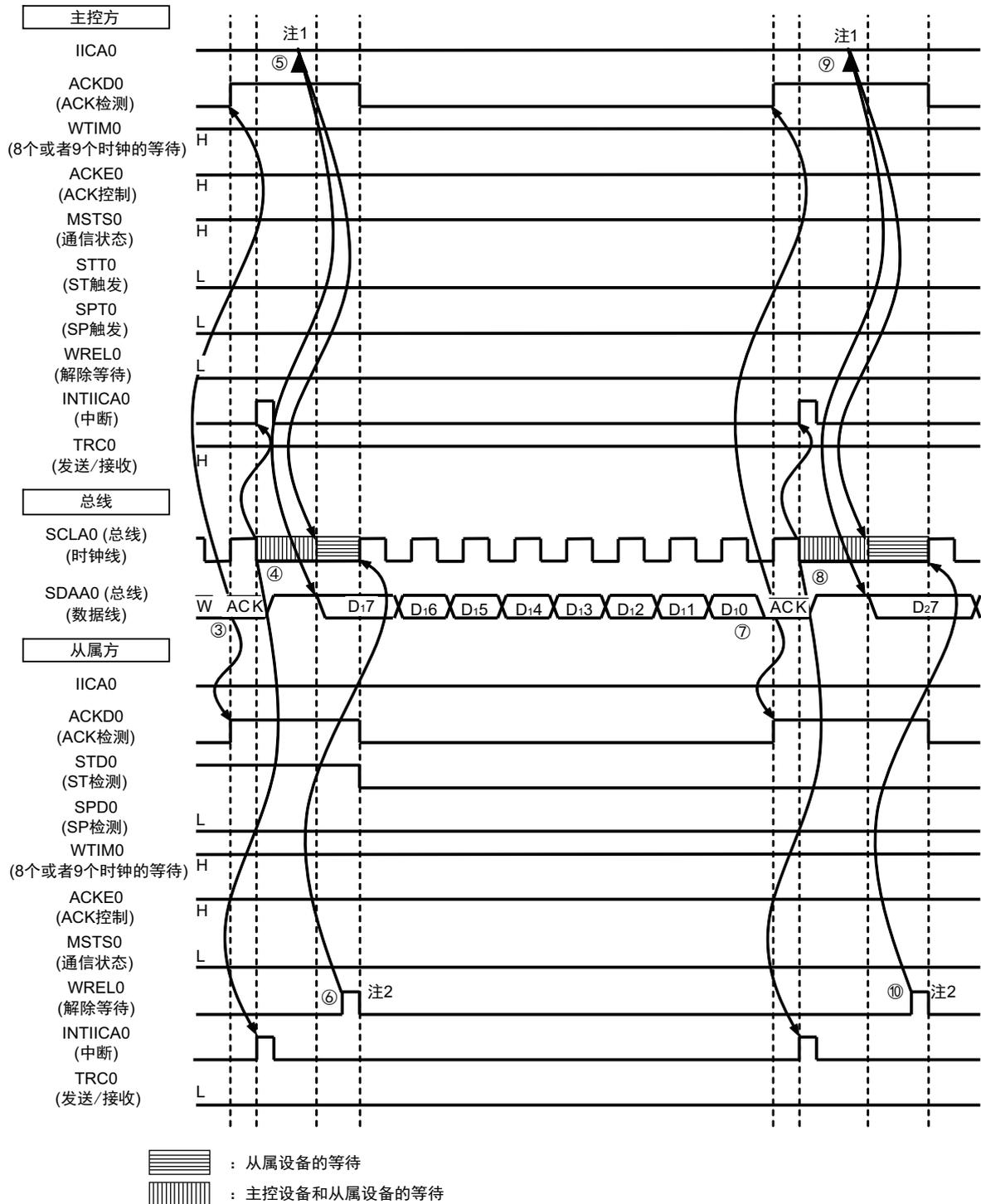
- ① 如果在主控方将开始条件触发置位（STT0=1），总线数据线（SDAA0）就下降，生成开始条件（SDAA0=0、SCLA0=1）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS0=1），在经过保持时间后总线时钟线下降（SCLA0=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 0（IICA0）写地址+W（发送），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVA0 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKD0=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICA0：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLA0=0），并且产生中断（INTIICA0：地址匹配中断）注。
- ⑤ 主控方给 IICA0 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（WREL0=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：SDAA0=1），并且不产生 INTIICA0 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 INTIICA0 中断（地址发送结束中断）。

备注 图 13-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。
图 13-31 的“(1) 开始条件~地址~数据”说明步骤①~⑥。
图 13-31 的“(2) 地址~数据~数据”说明步骤③~⑩。
图 13-31 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

图 13-31 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/4)

(2) 地址~数据~数据



注 1. 要在主控方的发送期间解除等待时，必须给 IICA0 写数据而不是将 WRELO 位置位。
 2. 要在从属方的接收期间解除等待时，必须将 IICA0 置“FFH”或者将 WRELO 位置位。

图 13-31 的“(2) 地址~数据~数据”的③~⑩的说明如下:

- ③ 在从属方, 如果接收地址和本地站地址 (SVA0 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 (ACKD0=1)。
- ④ 主控方在第 9 个时钟的下降沿产生中断 (INTIICA0: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLA0=0), 并且产生中断 (INTIICA0: 地址匹配中断)注。
- ⑤ 主控方给 IICA 移位寄存器 0 (IICA0) 写发送数据, 解除主控方的等待。
- ⑥ 如果从属方解除等待 (WREL0=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 ACKE0 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 (ACKD0=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 (SCLA0=0), 并且都产生中断 (INTIICA0: 传送结束中断)。
- ⑨ 主控方给 IICA0 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 (WREL0=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: SDAA0=1), 并且不产生 INTIICA0 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 INTIICA0 中断 (地址发送结束中断)。

备注 图 13-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

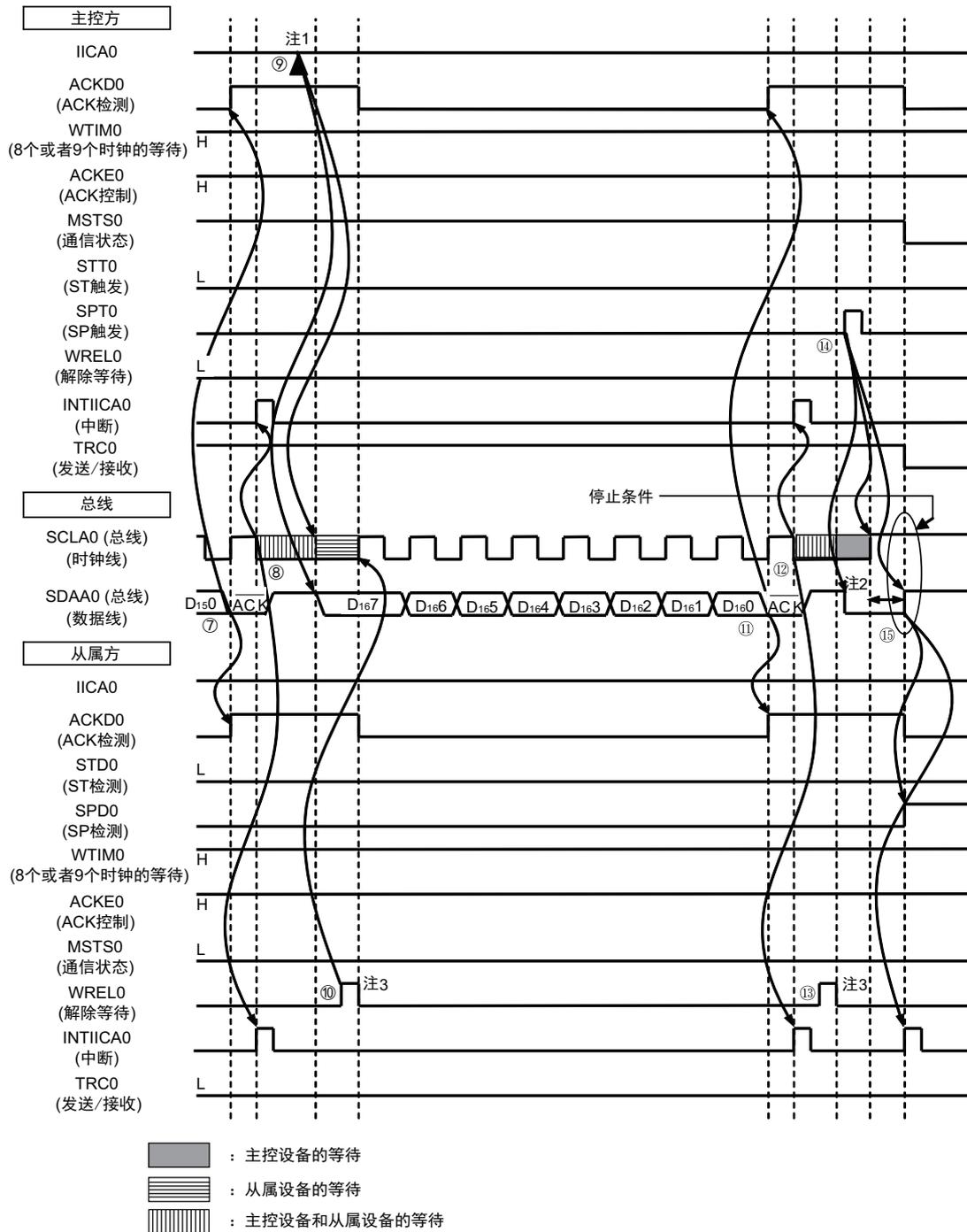
图 13-31 的“(1) 开始条件~地址~数据”说明步骤①~⑥。

图 13-31 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 13-31 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

图 13-31 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (3/4)

(3) 数据~数据~停止条件



- 注 1. 要在主控方的发送期间解除等待时，必须给 IICA0 写数据而不是将 WREL0 位置位。
2. 在发行停止条件后，从 SCLA0 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 IICA0 置“FFH”或者将 WREL0 位置位。

图 13-31 的“(3) 数据~数据~停止条件”的⑦~⑮的说明如下：

- ⑦ 在数据传送结束后，因为从属方的 ACKE0 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 (ACKD0=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 (SCLA0=0)，并且都产生中断 (INTIICA0：传送结束中断)。
- ⑨ 主控方给 IICA 移位寄存器 0 (IICA0) 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 (WREL0=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 (ACKE0=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 (ACKD0=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 (SCLA0=0)，并且都产生中断 (INTIICA0：传送结束中断)。
- ⑬ 从属方读接收数据，解除等待 (WREL0=1)。
- ⑭ 如果在主控方将停止条件触发置位 (SPT0=1)，就清除总线数据线 (SDAA0=0) 并且将总线时钟线置位 (SCLA0=1)，在经过停止条件的准备时间后将总线数据线置位 (SDAA0=1)，生成停止条件 (通过 SCLA0=1 使 SDAA0 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 (INTIICA0：停止条件中断)。

备注 图 13-31 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

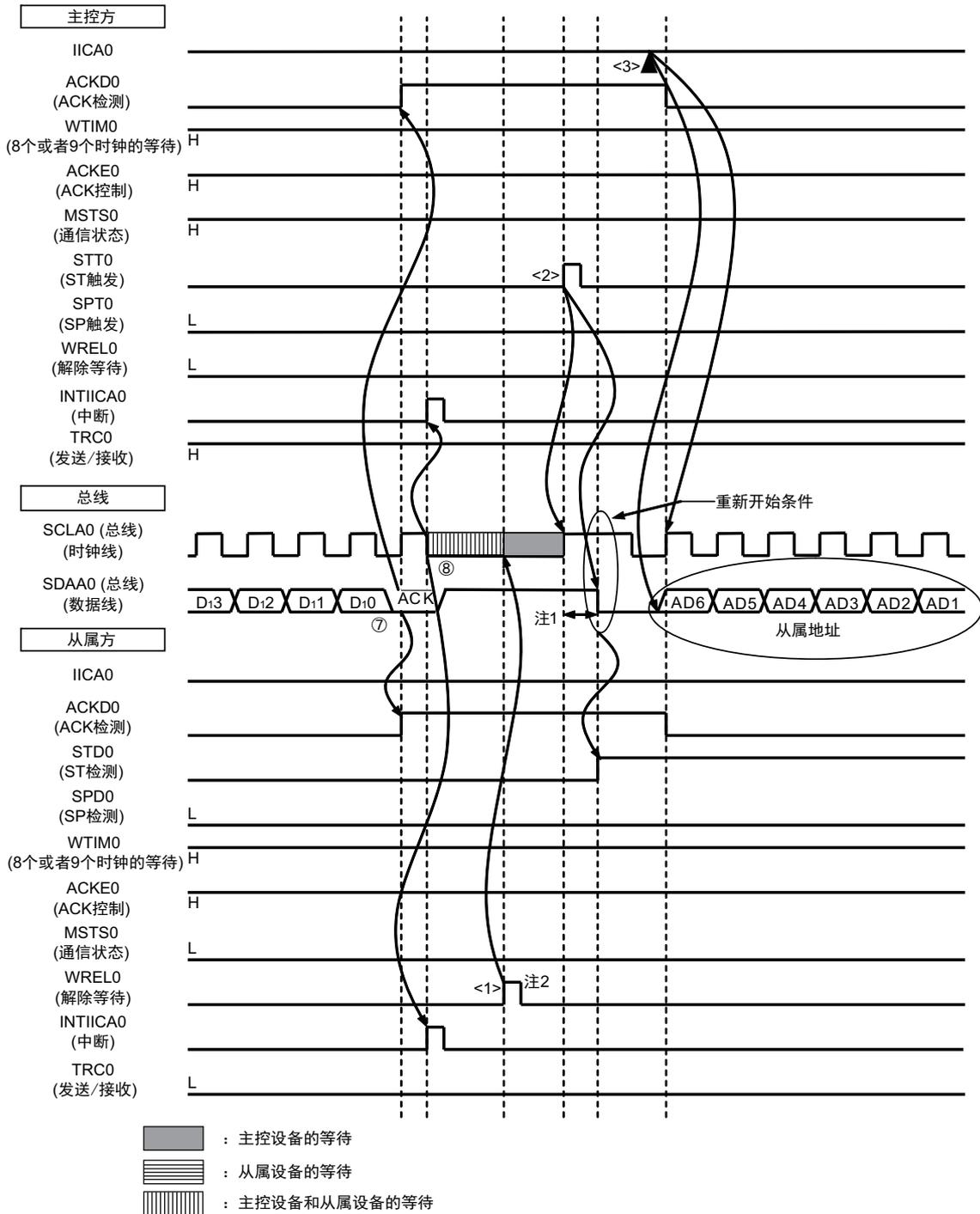
图 13-31 的“(1) 开始条件~地址~数据”说明步骤①~⑥。

图 13-31 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 13-31 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

图 13-31 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (4/4)

(4) 数据~重新开始条件~地址



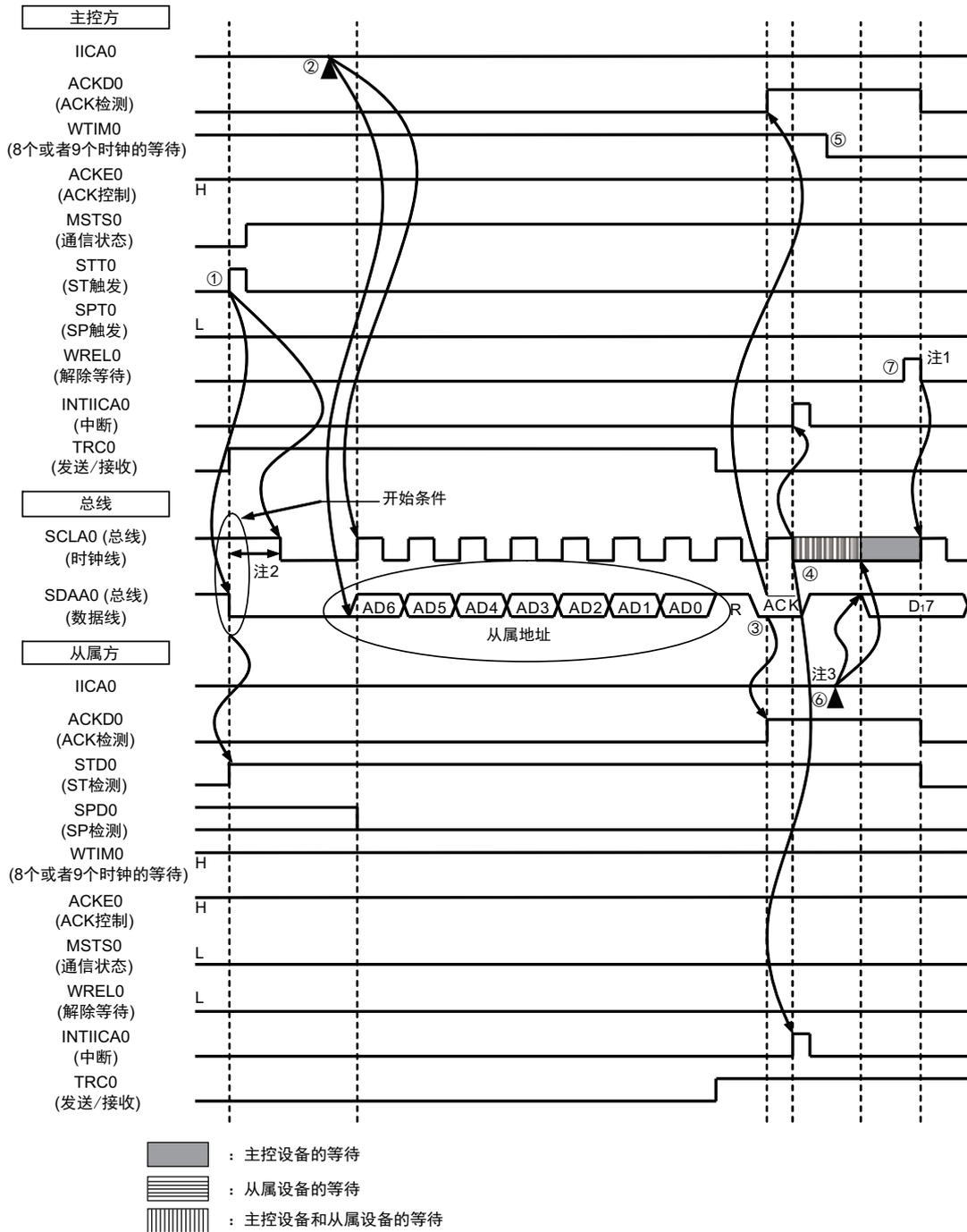
- 注 1. 在发行重新开始条件后，从 SCLA0 引脚信号上升到生成开始条件的的时间，在设定为标准模式时至少为 4.7μs，在设定为快速模式时至少为 0.6μs。
- 注 2. 要在从属方的接收期间解除等待时，必须将 IICA0 置“FFH”或者将 WREL0 位置位。

图 13-31 的“(4) 数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行 <1> ~ <3>, 从而返回步骤③的数据发送步骤。

- ⑦ 在数据传送结束后, 因为从属方的 ACKE0 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 (ACKD0=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 (SCLA0=0), 并且都产生中断 (INTIICA0: 传送结束中断)。
 - <1> 从属方读接收数据, 解除等待 (WREL0=1)。
 - <2> 如果在主控方再次将开始条件触发置位 (STT0=1), 总线时钟线就上升 (SCLA0=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 (SDAA0=0), 生成开始条件 (通过 SCLA0=1 使 SDAA0 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 (SCLA0=0), 结束通信准备。
 - <3> 如果主控方给 IICA 移位寄存器 0 (IICA0) 写地址 +R/W (发送), 就发送从属地址。

图 13-32 从属设备 → 主控设备的通信例子
 (主控设备: 选择 8 个时钟等待, 从属设备: 选择 9 个时钟等待) (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 IICA0 置 "FFH" 或者将 WREL0 位置位。
 2. 从 SDAA0 引脚信号下降到 SCLA0 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 IICA0 写数据而不是将 WREL0 位置位。

图 13-32 的“(1) 开始条件~地址~数据”的①~⑦的说明如下：

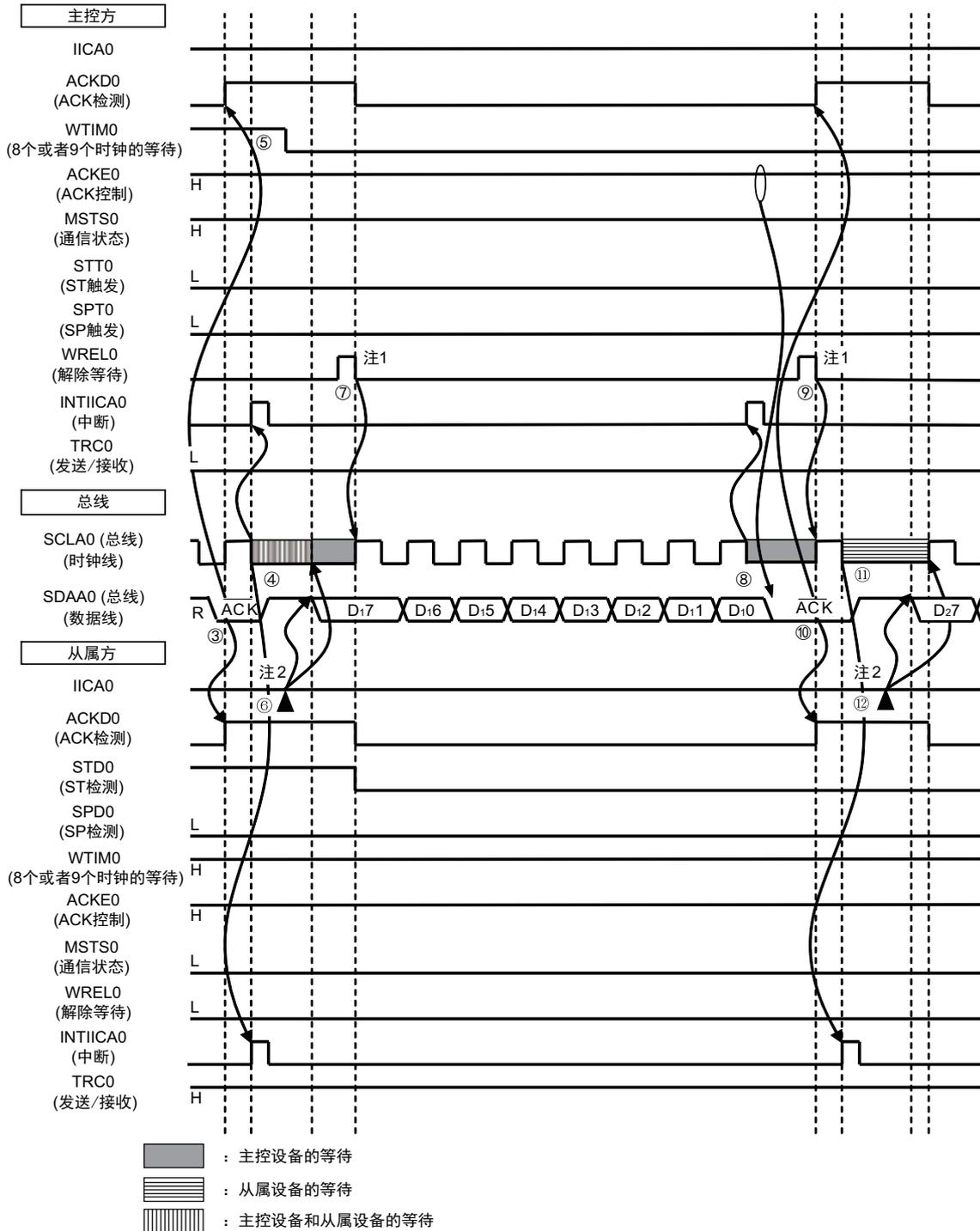
- ① 如果在主控方将开始条件触发置位（STT0=1），总线数据线（SDAA0）就下降，生成开始条件（通过 SCLA0=1 使 SDAA0 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS0=1），在经过保持时间后总线时钟线下降（SCLA0=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 0（IICA0）写地址+R（接收），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVA0 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKD0=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICA0：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLA0=0），并且产生中断（INTIICA0：地址匹配中断）注。
- ⑤ 主控方将等待时序更改为第 8 个时钟（WTIM0=0）。
- ⑥ 从属方给 IICA0 寄存器写发送数据，解除从属方的等待。
- ⑦ 主控方解除等待（WREL0=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：SDAA0=1），并且不产生 INTIICA0 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 INTIICA0 中断（地址发送结束中断）。

备注 图 13-32 的①~⑱是通过 I²C 总线进行数据通信的一系列运行步骤。
图 13-32 的“(1) 开始条件~地址~数据”说明步骤①~⑦。
图 13-32 的“(2) 地址~数据~数据”说明步骤③~⑫。
图 13-32 的“(3) 数据~数据~停止条件”说明步骤⑧~⑱。

图 13-32 从属设备 → 主控设备的通信例子
 (主控设备: 选择 8 个时钟等待, 从属设备: 选择 9 个时钟等待) (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 IICA0 置“FFH”或者将 WRELO 位置位。
 2. 要在从属方的发送期间解除等待时，必须给 IICA0 写数据而不是将 WRELO 位置位。

图 13-32 的“(2) 地址~数据~数据”的③~⑫的说明如下：

- ③ 在从属方，如果接收地址和本地站地址（SVA0 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（ACKD0=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICA0：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLA0=0），并且产生中断（INTIICA0：地址匹配中断）注。
- ⑤ 主控方将等待时序改为第 8 个时钟（WTIM0=0）。
- ⑥ 从属方给 IICA 移位寄存器 0（IICA0）写发送数据，解除从属方的等待。
- ⑦ 主控方解除等待（WREL0=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（SCLA0=0），并且产生中断（INTIICA0：传送结束中断）。因为主控方的 ACKE0 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（WREL0=1）。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK（ACKD0=1）。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态（SCLA0=0），并且产生中断（INTIICA0：传送结束中断）。
- ⑫ 如果从属方给 IICA0 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：SDAA0=1），并且不产生 INTIICA0 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 INTIICA0 中断（地址发送结束中断）。

备注 图 13-32 的①~⑱是通过 I²C 总线进行数据通信的一系列运行步骤。

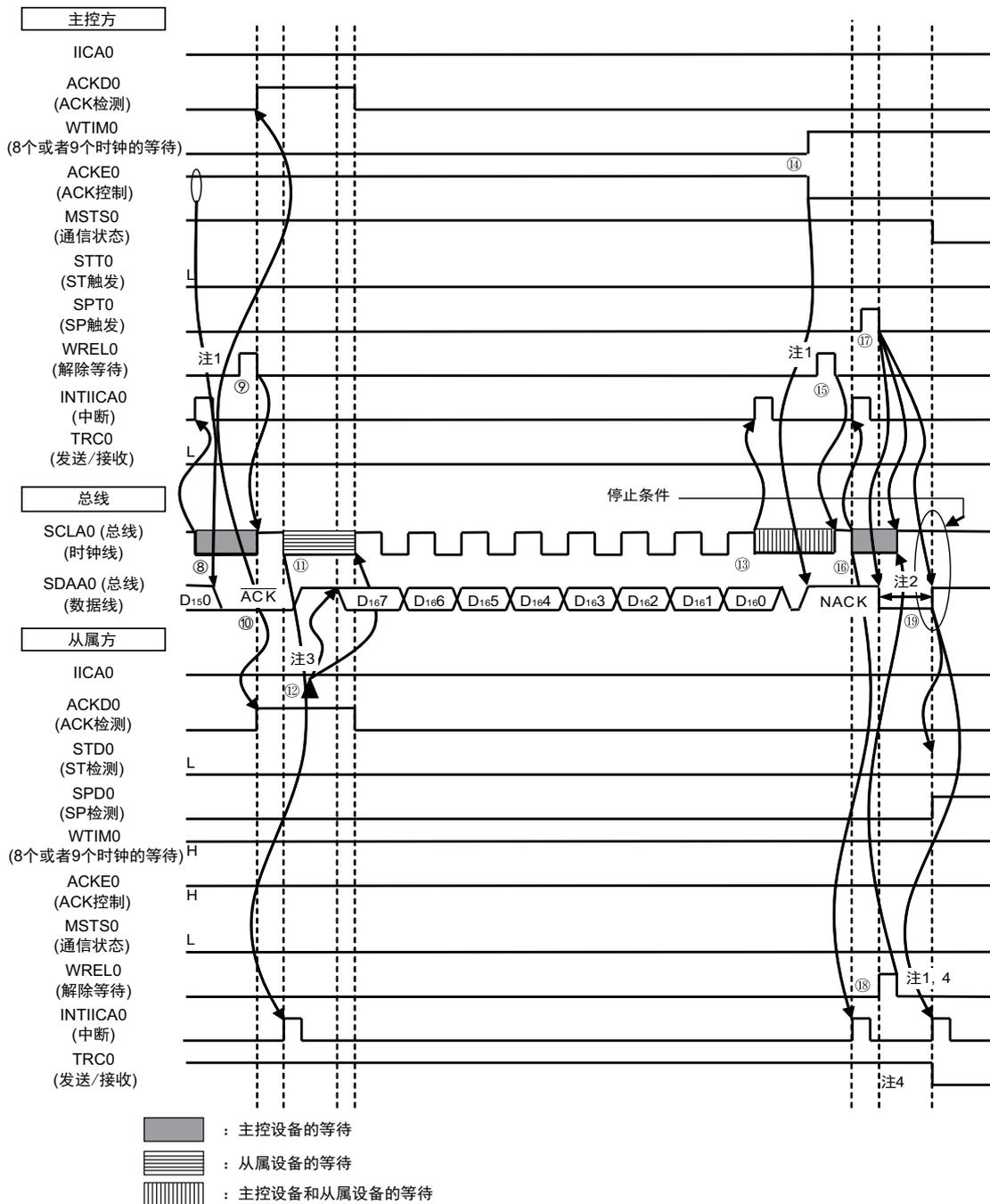
图 13-32 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

图 13-32 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 13-32 的“(3) 数据~数据~停止条件”说明步骤⑧~⑱。

图 13-32 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个 → 9 个时钟等待，从属设备：选择 9 个时钟等待) (3/3)

(3) 数据~数据~停止条件



- 注 1. 要解除等待时，必须将 IICA0 置“FFH”或者将 WREL0 位置位。
 2. 在发行停止条件后，从 SCLA0 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. 要解除从属发送期间的等待时，必须给 IICA0 写数据而不是将 WREL0 位置位。
 4. 在从属方的发送期间，如果通过 WREL0 位的置位来解除等待，就清除 TRC0 位。

图 13-32 的“(3) 数据~数据~停止条件”的⑧~⑱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 (SCLA0=0)，并且产生中断 (INTIICA0: 传送结束中断)。因为主控方的 ACKE0 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 (WREL0=1)。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK (ACKD0=1)。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态 (SCLA0=0)，并且产生中断 (INTIICA0: 传送结束中断)。
- ⑫ 如果从属方给 IICA 移位寄存器 0 (IICA0) 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 (INTIICA0: 传送结束中断)，并且进入等待状态 (SCLA0=0)。因为进行 ACK 控制 (ACKE0=1)，所以此阶段的总线数据线变为低电平 (SDAA0=0)。
- ⑭ 主控方设定为 NACK 应答 (ACKE0=0)，并且将等待时序更改为第 9 个时钟 (WTIM0=1)。
- ⑮ 如果主控方解除等待 (WREL0=1)，从属方就在第 9 个时钟的上升沿检测到 NACK (ACKD0=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 (SCLA0=0)，并且都产生中断 (INTIICA0: 传送结束中断)。
- ⑰ 如果主控方发行停止条件 (SPT0=1)，就清除总线数据线 (SDAA0=0)，并且解除主控方的等待。此后，主设备处于待机状态，直到将总线时钟线置位 (SCLA0=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 (WREL0=1)。如果解除从属方的等待，就将总线时钟线置位 (SCLA0=1)。
- ⑲ 如果主控方确认到总线时钟线被置位 (SCLA0=1)，就在经过停止条件准备时间后将总线数据线置位 (SDAA0=1)，然后发行停止条件 (通过 SCLA0=1 使 SDAA0 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 (INTIICA0: 停止条件中断)。

备注 图 13-32 的①~⑱是通过 I²C 总线进行数据通信的一系列运行步骤。

图 13-32 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

图 13-32 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 13-32 的“(3) 数据~数据~停止条件”说明步骤⑧~⑱。

第 14 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。在转移处理结束后返回到原来中断的执行程序。

中断源的个数因产品而不同。

		10 引脚产品	16 引脚产品
可屏蔽中断	外部	3	5
	内部	8	14

14.1 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、PR10L、PR10H、PR01L、PR11L），能将可屏蔽中断的优先级分成四组。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优先级（默认）进行处理。有关优先级（默认），请参照表 14-1、表 14-2。

产生待机解除信号，解除 STOP 模式和 HALT 模式。

可屏蔽中断分为外部中断请求和内部中断请求。

(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。

14.2 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 4 种复位源（参照表 14-1）。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代码各为两个字节，所以中断的转移目标地址为 00000H ~ 0FFFFH 的 64K 地址。

表 14-1 中断源一览表 (10 引脚产品)

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 (上溢时间的 75%+3/(4×f _L))	内部	0004H	(a)	
	1	INTP0	引脚输入边沿的检测	外部	0006H	(b)	
	2	INTP1			0008H		
	3	INTST0/ INTCSI00/ INTIIC00	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 / IIC00 的传送结束		内部	000AH	(a)
	4	INTSR0	UART0 接收的传送结束			000CH	
	5	INTSRE0	发生 UART0 接收的通信错误			000EH	
	6	INTTM01H	定时器通道 1 的计数结束或者运行开始时 (高 8 位定时器运行时)			0010H	
	7	INTTM00	定时器通道 0 的计数结束、捕捉结束或者运行开始时			0012H	
	8	INTTM01	定时器通道 1 的计数结束、捕捉结束或者运行开始时 (16 位定时器运行时或者低 8 位定时器运行时)			0014H	
	9	INTAD	A/D 转换结束			0016H	
10	INTKR	键返回信号的检测		外部	0018H	(c)	
软件	—	BRK	执行 BRK 指令		—	007EH	(d)
复位	—	RESET	RESET 引脚输入		—	0000H	—
		SPOR	可选择的上电复位				
		WDT	看门狗定时器的上溢				
		TRAP	非法指令的执行 ^{注3}				

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，10 表示最低优先级。

2. 基本构成类型 (a) ~ (d) 分别对应图 14-1 的 (a) ~ (d)。

3. 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。

表 14-2 中断源一览表 (16 引脚产品)

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 (上溢时间的 75%+3/(4×f _L))	内部	0004H	(a)	
	1	INTP0	引脚输入边沿的检测	外部	0006H	(b)	
	2	INTP1			0008H		
	3	INTST0/ INTCSI00/ INTIIC00	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 /IIC00 的传送结束	内部	000AH	(a)	
	4	INTSR0 INTCSI01	UART0 接收的传送结束 CSI01 通信的结束		000CH		
	5	INTSRE0	发生 UART0 接收的通信错误		000EH		
	6	INTTM01H	定时器通道 1 的计数结束或者运行开始时 (高 8 位定时器运行时)		0010H		
	7	INTTM00	定时器通道 0 的计数结束、捕捉结束或者运行开始时		0012H		
	8	INTTM01	定时器通道 1 的计数结束、捕捉结束或者运行开始时 (16 位定时器运行时或者低 8 位定时器运行时)		0014H		
	9	INTAD	A/D 转换结束		0016H		
	10	INTKR	键返回信号的检测		0018H		(c)
	11	INTP2	引脚输入边沿的检测		外部		001AH
	12	INTP3		001CH			
	13	INTTM03H	定时器通道 3 的计数结束或者运行开始时 (高 8 位定时器运行时)	内部	001EH	(a)	
	14	INTIICA0	IICA 通信的结束		0020H		
	15	INTTM02	定时器通道 2 的计数结束、捕捉结束或者运行开始时		0022H		
	16	INTTM03	定时器通道 3 的计数结束、捕捉结束或者运行开始时 (16 位定时器运行时或者低 8 位定时器运行时)		0024H		
	17	INTIT	间隔定时器的信号检测		0026H		
18	INTCMP0	比较器的有效边沿检测	0028H				
软件	—	BRK	执行 BRK 指令	—	007EH	(d)	
复位	—	RESET	RESET 引脚输入	—	0000H	—	
		SPOR	可选择的上电复位				
		WDT	看门狗定时器的上溢				
		TRAP	非法指令的执行注 3				

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，18 表示最低优先级。

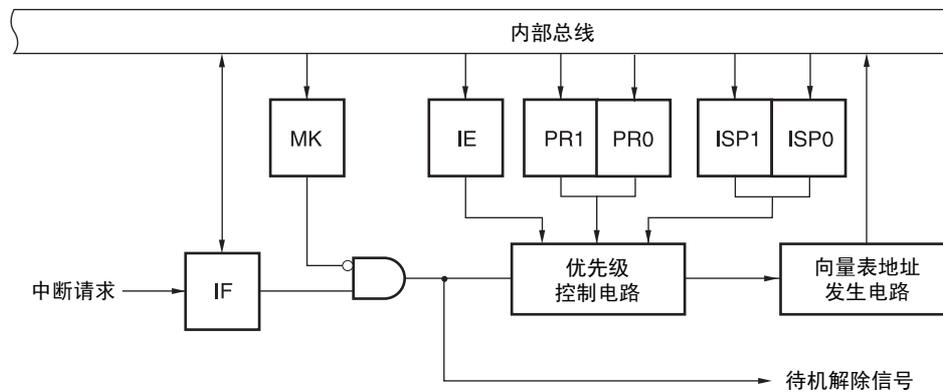
2. 基本构成类型 (a) ~ (d) 分别对应图 14-1 的 (a) ~ (d)。

3. 在执行指令码 FFH 时发生。

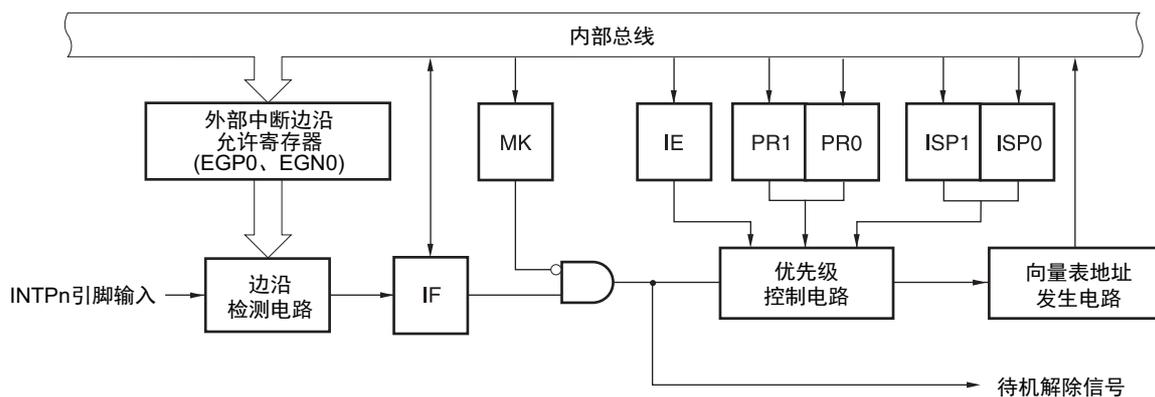
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

图 14-1 中断功能的基本结构 (1/2)

(A) 内部可屏蔽中断



(B) 外部可屏蔽中断 (INTPn)



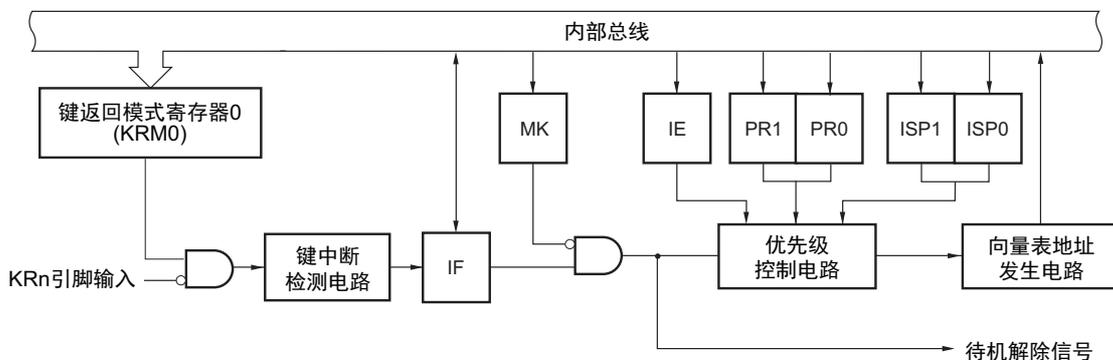
IF: 中断请求标志
 IE: 中断允许标志
 ISP0: 优先级控制标志 0
 ISP1: 优先级控制标志 1
 MK: 中断屏蔽标志
 PR0: 优先级指定标志 0
 PR1: 优先级指定标志 1

备注 10 引脚产品: n=0、1

16 引脚产品: n=0 ~ 3

图 14-1 中断功能的基本构成 (2/2)

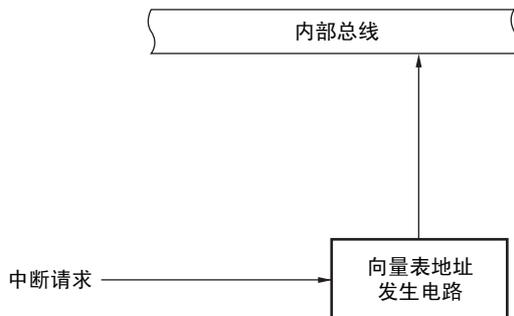
(C) 外部可屏蔽中断 (INTKR)



- IF: 中断请求标志
- IE: 中断允许标志
- ISP0: 优先级控制标志 0
- ISP1: 优先级控制标志 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

(D) 软件中断



14.3 中断功能的控制寄存器

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、PR10L、PR10H、PR01L、PR11L)
- 外部中断上升沿允许寄存器 0 (EGP0)
- 外部中断下降沿允许寄存器 0 (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 14-3、表 14-4 所示。

表 14-3 中断请求源对应的各种标志 (10 引脚产品)

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
INTP0	PIF0		PMK0		PPR00、PPR10	
INTP1	PIF1		PMK1		PPR01、PPR11	
INTST0 ^注	STIF0 ^注		STMK0 ^注		STPR00、STPR10 ^注	
INTCSI00 ^注	CSIIIF00 ^注		CSIMK00 ^注		CSIPR000、CSIPR100 ^注	
INTIIC00 ^注	IICIF00 ^注		IICMK00 ^注		IICPR000、IICPR100 ^注	
INTSR0	SRIF0		SRMK0		SRPR00、SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00、SREPR10	
INTTM01H	TMIF01H		TMMK01H		TMPR001H、TMPR101H	
INTTM00	TMIF00	TMMK00	TMPR000、TMPR100			
INTTM01	TMIF01	IF0H	TMMK01	MK0H	TMPR001、TMPR101	PR00H、 PR10H
INTAD	ADIF		ADMK		ADPR0、ADPR1	
INTKR	KRIF		KRMK		KRPR0、KRPR1	

注 如果发生 INTST0、INTCSI00 或者 INTIIC00 的中断源，IF0L 寄存器的 bit3 就被置“1”。MK0L、PR00L、PR10L 寄存器的 bit3 对应这 3 个中断源。

表 14-4 中断请求源对应的各种标志（16 引脚产品）

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
INTP0	PIF0		PMK0		PPR00、PPR10	
INTP1	PIF1		PMK1		PPR01、PPR11	
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	
INTIIC00 注 1	IICIF00 注 1		IICMK00 注 1		IICPR000、IICPR100 注 1	
INTSR0 注 2	SRIF0 注 2		SRMK0 注 2		SRPR00、SRPR10 注 2	
INTCSI01 注 2	CSIF01 注 2		CSIMK01 注 2		CSIPR001、CSIPR101 注 2	
INTSRE0	SREIF0	SREMK0	SREPR00、SREPR10			
INTTM01H	TMIF01H	TMMK01H	TMPR001H、TMPR101H			
INTTM00	TMIF00	TMMK00	TMPR000、TMPR100			
INTTM01	TMIF01	IF0H	TMMK01	MK0H	TMPR001、TMPR101	PR00H、 PR10H
INTAD	ADIF		ADMK		ADPR0、ADPR1	
INTKR	KRIF		KRMK		KRPR0、KRPR1	
INTP2	PIF2		PMK2		PPR02、PPR12	
INTP3	PIF3		PMK3		PPR03、PPR13	
INTTM03H	TMIF03H		TMMK03H		TMPR003H、TMPR103H	
INTIICA0	IICAIF0		IICAMK0		IICAPR00、IICAPR10	
INTTM02	TMIF02		TMMK02		TMPR002、TMPR102	
INTTM03	TMIF03	IF1L	TMMK03	MK1L	TMPR003、TMPR103	PR01L、 PR11L
INTIT	ITIF		ITMK		ITPR0、ITPR1	
INTCMP0	CMPIF0		CMPMK0		CMPPR00、CMPPR10	

- 注 1. 如果发生 INTST0、INTCSI00 或者 INTIIC00 的中断源，IF0L 寄存器的 bit3 就被置“1”。MK0L、PR00L、PR10L 寄存器的 bit3 对应这 3 个中断源。
2. 如果发生 INTSR0 或者 INTCSI01 的中断源，IF0L 寄存器的 bit4 就被置“1”。MK0L、PR00L、PR10L 寄存器的 bit4 对应这 2 个中断源。

14.3.1 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。

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 14-2 中断请求标志寄存器 (IF0L、IF0H) 的格式 (10 引脚产品)

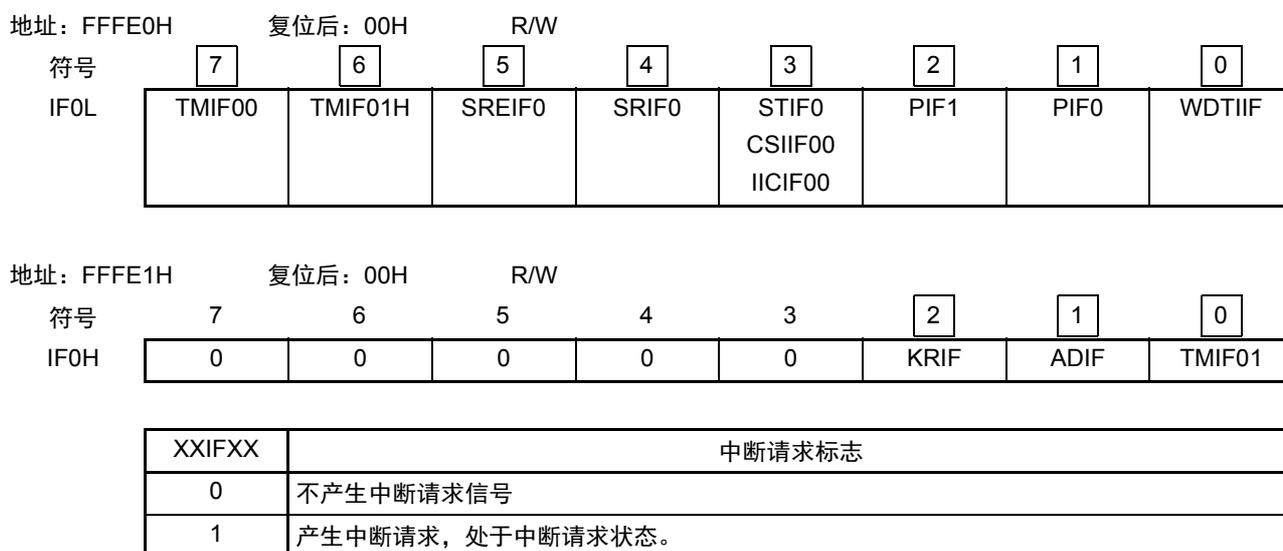


图 14-3 中断请求标志寄存器 (IF0L、IF0H、IF1L) 的格式 (16 引脚产品)

地址: FFFE0H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IF0L	TMIF00	TMIF01H	SREIF0	SRIF0 CSIIF01	STIF0 CSIIF00 IICIF00	PIF1	PIF0	WDTIIF

地址: FFFE1H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IF0H	TMIF02	IICAIF0	TMIF03H	PIF3	PIF2	KRIF	ADIF	TMIF01

地址: FFFE2H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IF1L	0	0	0	0	0	CMPIF0	ITIF	TMIF03

XXIFXX	中断请求标志
0	不产生中断请求信号
1	产生中断请求, 处于中断请求状态。

注意 1. 不能更能未定义位的内容。

- 在操作中中断请求标志寄存器的标志时, 必须使用 1 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 1 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;" 的描述时, 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需要注意。

14.3.2 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。

通过 1 位或者 8 位存储器操作指令设定 MK0L、MK0H、MK1L 寄存器。

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 14-4 中断屏蔽标志寄存器 (MK0L、MK0H) 的格式 (10 引脚产品)

地址: FFFE4H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00 IICMK00	PMK1	PMK0	WDTIMK

地址: FFFE5H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK0H	1	1	1	1	1	KRMK	ADMK	TMMK01

XXMKXX	中断处理的控制
0	允许中断处理。
1	禁止中断处理。

图 14-5 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L) 的格式 (16 引脚产品)

地址: FFFE4H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00 IICMK00	PMK1	PMK0	WDTIMK

地址: FFFE5H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK0H	TMMK02	IICAMK0	TMMK03H	PMK3	PMK2	KRMK	ADMK	TMMK01

地址: FFFE6H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
MK1L	1	1	1	1	1	CMPMK0	ITMK	TMMK03

XXMKXX	中断处理的控制
0	允许中断处理。
1	禁止中断处理。

注意 不能更改未定义位的内容。

14.3.3 优先级指定标志寄存器 (PR00L、PR00H、PR10L、PR10H、PR01L、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)。

通过 1 位或者 8 位存储器操作指令设定 PR00L、PR00H、PR10L、PR10H、PR01L、PR11L 寄存器。

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 14-6 优先级指定标志寄存器 (PR00L、PR00H、PR10L、PR10H) 的格式 (10 引脚产品)

地址: FFFE8H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR00L	TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000 IICPR000	PPR01	PPR00	WDTIPR0

地址: FFFECH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR10L	TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100 IICPR100	PPR11	PPR10	WDTIPR1

地址: FFFE9H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR00H	1	1	1	1	1	KRPR0	ADPR0	TMPR001

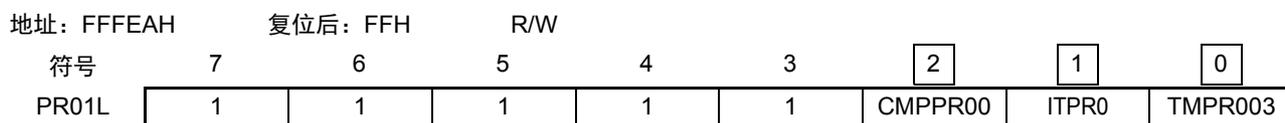
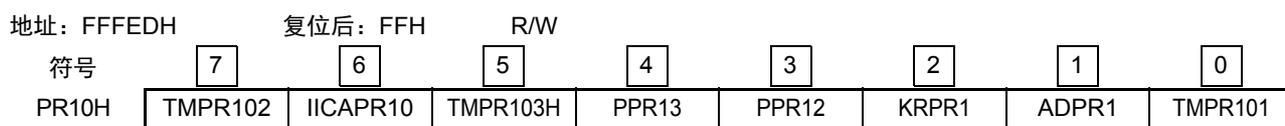
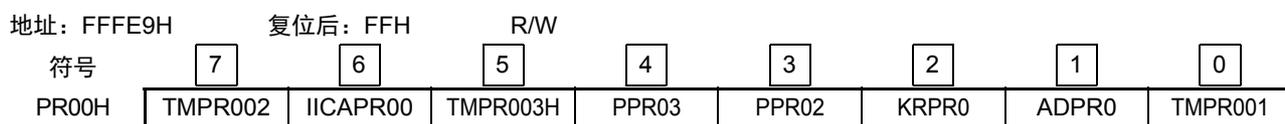
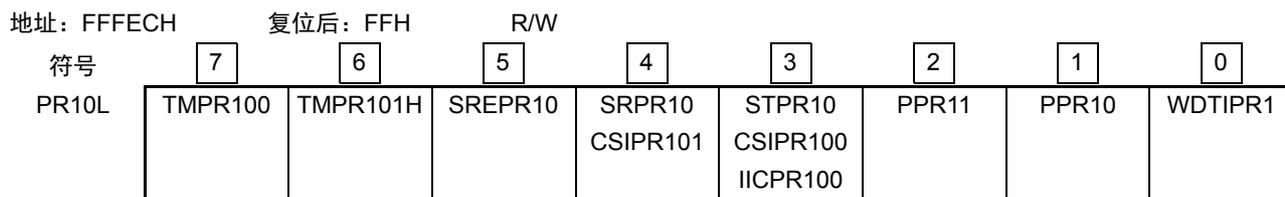
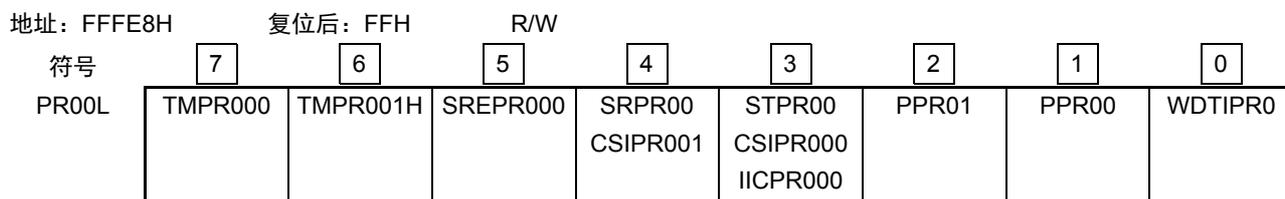
地址: FFFEDH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR10H	1	1	1	1	1	KRPR1	ADPR1	TMPR101

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 不能更改未定义位的内容。

图 14-7 优先级指定标志寄存器 (PR00L、PR00H、PR10L、PR10H、PR01L、PR11L) 的格式 (16 引脚产品)



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 不能更改未定义位的内容。

14.3.4 外部中断上升沿允许寄存器 0 (EGP0) 和外部中断下降沿允许寄存器 0 (EGN0)

这些寄存器设定 INTP0、INTP1、INTP2、INTP3 的有效边沿。

通过 1 位或者 8 位存储器操作指令设定 EGP0、EGN0 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 14-8 外部中断上升沿允许寄存器 0 (EGP0) 和外部中断下降沿允许寄存器 0 (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	0	0	EGP3注	EGP2注	EGP1	EGP0

地址: FFF39H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	EGN3注	EGN2注	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效边沿的选择 (n=0 ~ 3)
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

注 只限于 16 引脚产品。

注意 如果将用于外部中断功能的输入端口切换至输出模式，就有可能检测有效边沿并且发生 INTPn 中断。

如果要切换至输出模式，必须先禁止边沿检测 (EGPn、EGNn=0、0)，然后将端口模式寄存器 (PMxx) 置“0”。

备注 1. 有关边沿检测端口，请参照“2.1 端口功能”。

2. n=0 ~ 3

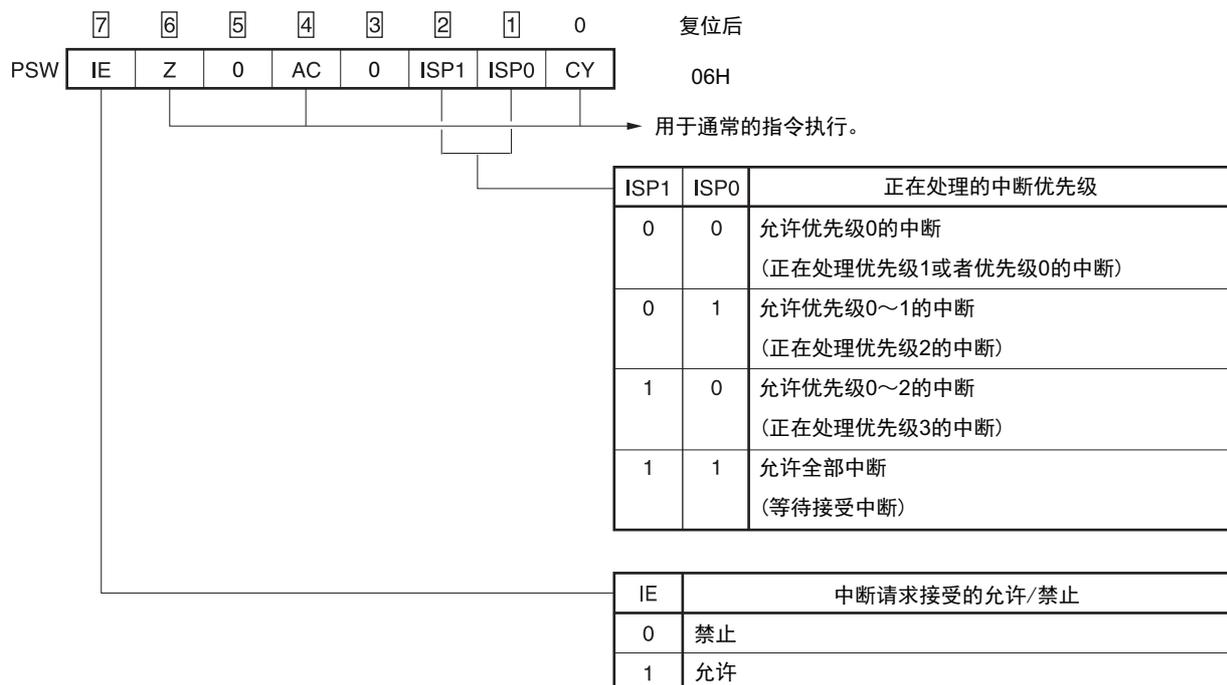
14.3.5 程序状态字 (PSW)

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，将接受的中断优先级指定标志寄存器的内容 (“00”以外) “-1”后的值传送到 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 14-9 程序状态字的结构



14.4 中断处理的操作

14.4.1 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”）下接受向量中断处理。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向量中断请求。

从产生可屏蔽中断请求到进行向量中断处理的时间如表 14-5 所示。

有关中断请求的接受时序，请参照图 14-11 和图 14-12。

表 14-5 从产生可屏蔽中断到处理的时间

	最短时间	最长时间注
处理时间	11 个时钟	18 个时钟

注 执行内部 RAM 区的指令时除外。

备注 1 个时钟： $1/f_{\text{CLK}}$ (f_{CLK} : 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。

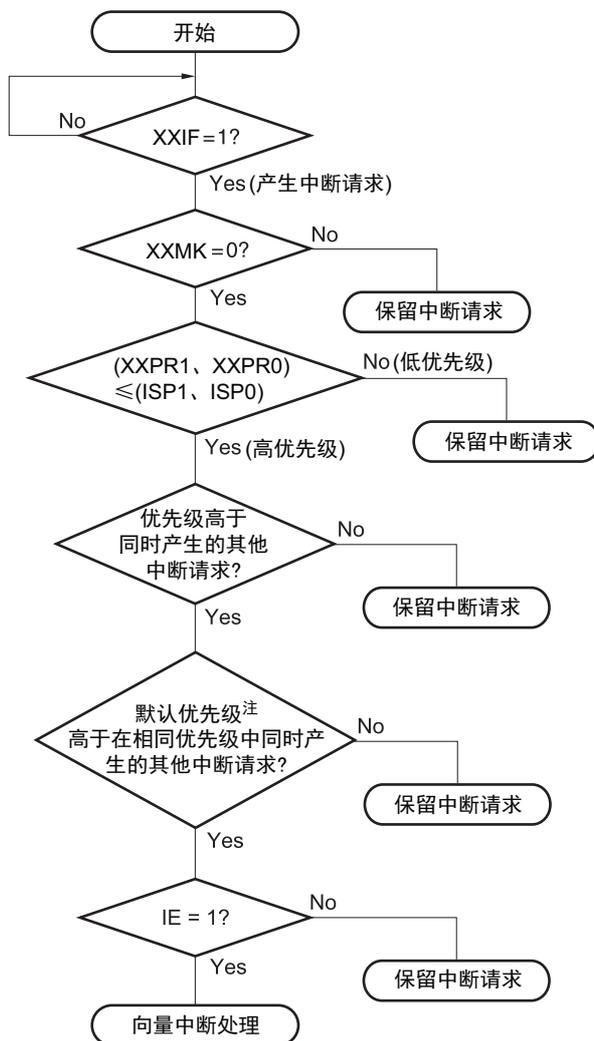
被保留的中断请求在可接受的状态时被接受。

接受中断请求的处理算法如图 14-10 所示。

如果接受可屏蔽中断请求，就按程序状态字（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 ISPO 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 14-10 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 14-9)

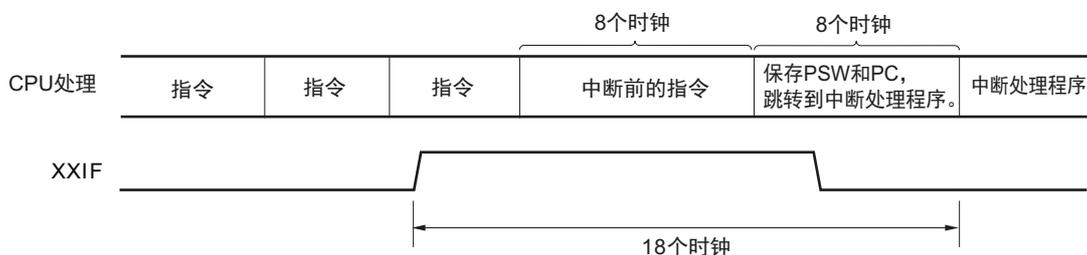
注 有关默认优先级, 请参照“表 14-1、表 14-2 中断源一览表”。

图 14-11 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 14-12 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)

14.4.2 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。

14.4.3 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。

能多重中断的中断请求的关系和多重中断的例子如表 14-6 和图 14-13 所示。

表 14-6 在中断处理过程中能多重中断的中断请求的关系

多重中断请求		可屏蔽中断请求								软件中 断请求
		优先级 0 (PR=00)		优先级 1 (PR=01)		优先级 2 (PR=10)		优先级 3 (PR=11)		
		IE=1	IE=0	IE=1	IE=0	IE=1	IE=0	IE=1	IE=0	
正在处理的中断										
可屏蔽中断	ISP1=0 ISP0=0	○	×	×	×	×	×	×	×	○
	ISP1=0 ISP0=1	○	×	○	×	×	×	×	×	○
	ISP1=1 ISP0=0	○	×	○	×	○	×	×	×	○
	ISP1=1 ISP0=1	○	×	○	×	○	×	○	×	○
软件中断		○	×	○	×	○	×	○	×	○

备注 1. ○：能多重中断

2. ×：不能多重中断

3. ISP0、ISP1 和 IE 是 PSW 中的标志。

ISP1=0、ISP0=0：正在处理优先级 0 或者优先级 1 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、PR10L、PR10H、PR01L、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）。

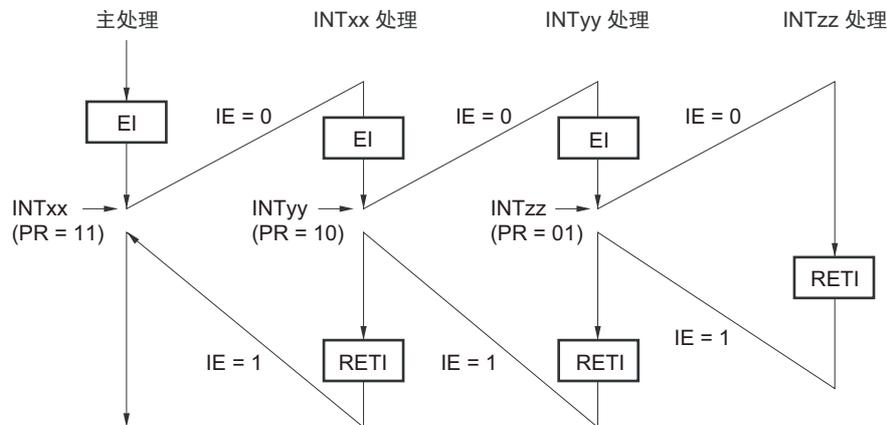
PR=01：通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

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）。

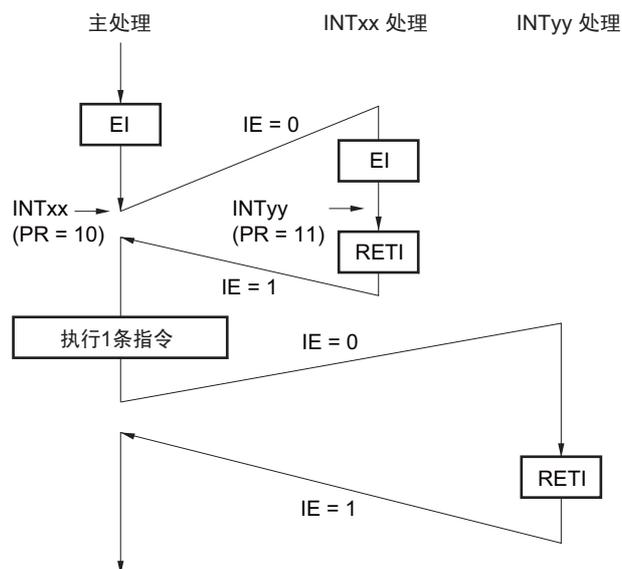
图 14-13 多重中断的例子 (1/2)

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 因其中断优先级低于 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。

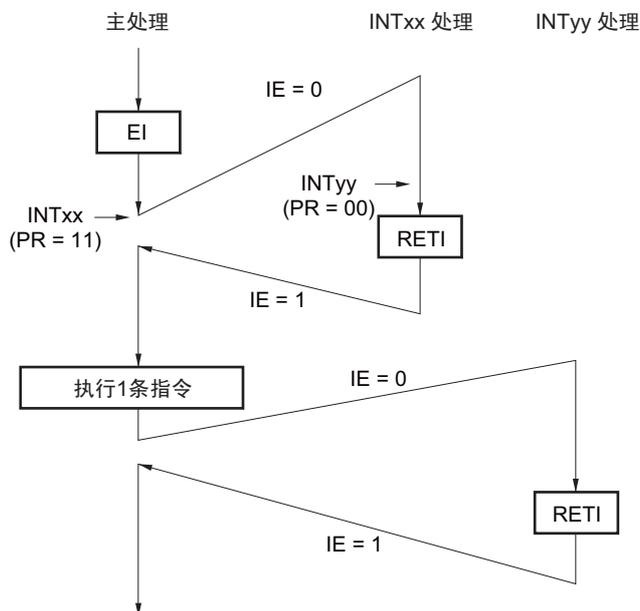
PR=11: 通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

IE=0: 禁止接受中断请求。

IE=1: 允许接受中断请求。

图 14-13 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。

14.4.4 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、PR10L、PR10H、PR01L和PR11L的各寄存器的写指令

中断请求的保留时序如图 14-14 所示。

图 14-14 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 15 章 键中断功能

15.1 键中断的功能

能通过给键中断输入引脚（KR0 ~ KR5）输入上升沿 / 下降沿，产生键中断（INTKR）。

表 15-1 键中断检测引脚的分配

键中断引脚	键返回模式寄存器（KRM0）	键返回标志寄存器（KRF）
KR0	KRM00	KRF0
KR1	KRM01	KRF1
KR2	KRM02	KRF2
KR3	KRM03	KRF3
KR4	KRM04	KRF4
KR5	KRM05	KRF5

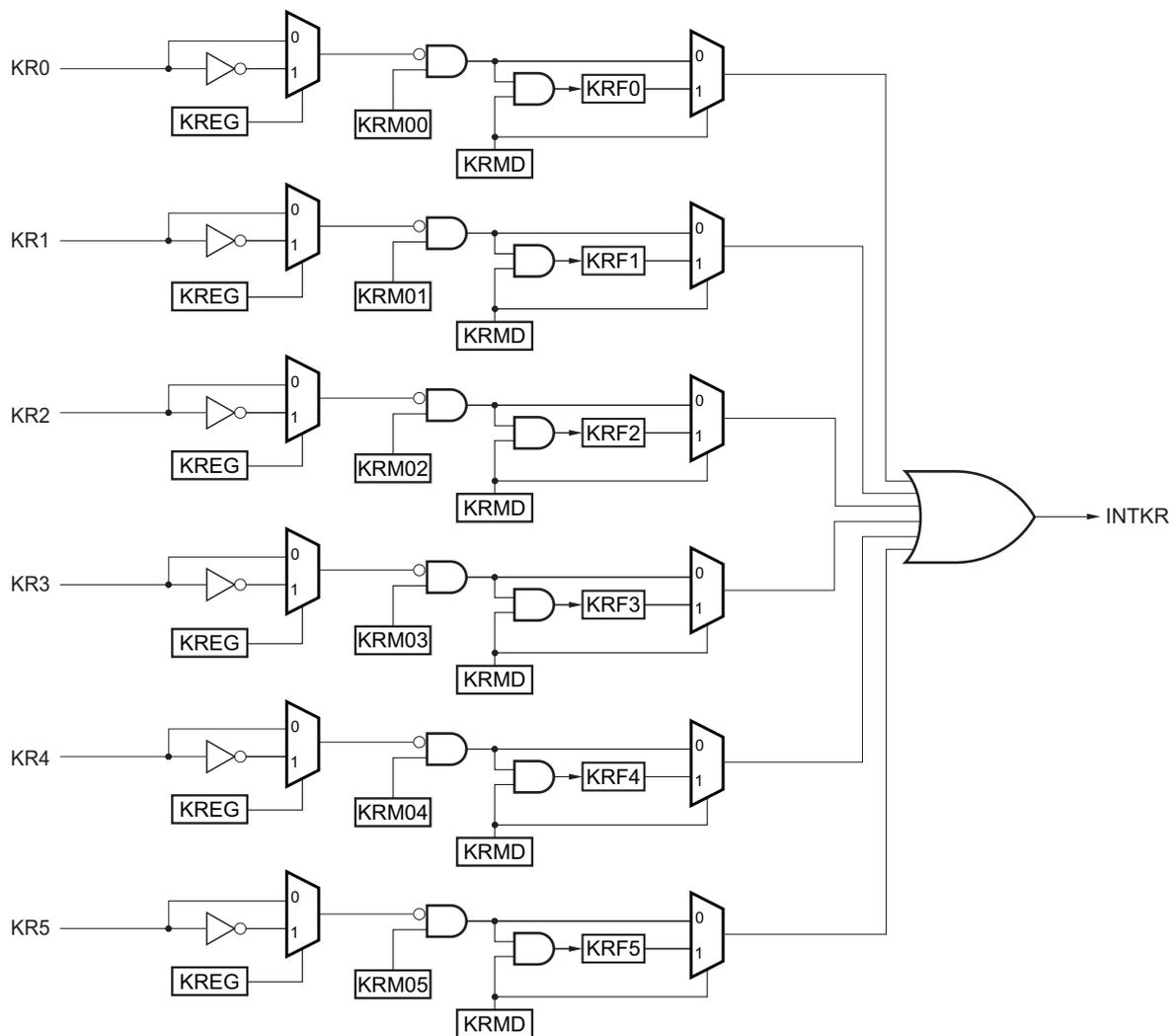
15.2 键中断的结构

键中断由以下硬件构成。

表 15-2 键中断的结构

项目	控制寄存器
控制寄存器	键返回控制寄存器（KRCTL） 键返回模式寄存器（KRM0） 键返回标志寄存器（KRF） 端口模式控制寄存器 0（PMC0） 端口模式寄存器 0、4（PM0、PM4）

图 15-1 键中断的框图



15.3 键中断的控制寄存器

通过以下 5 种寄存器控制键中断功能。

- 键返回控制寄存器 (KRCTL)
- 键返回模式寄存器 (KRM0)
- 键返回标志寄存器 (KRF)
- 端口模式控制寄存器0 (PMC0)
- 端口模式寄存器0、4 (PM0、PM4)

15.3.1 键返回控制寄存器 (KRCTL)

KRCTL 寄存器是设定键中断标志 (KRF0 ~ KRF5) 的使用和检测边沿的寄存器。

通过 1 位或者 8 位存储器操作指令设定 KRCTL 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 15-2 键返回控制寄存器 (KRCTL) 的格式

地址: FFF34H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	键中断标志 (KRF0 ~ KRF5) 的使用
0	不使用键中断标志
1	使用键中断标志

KREG	检测边沿的选择 (KR0 ~ KR5)
0	下降沿
1	上升沿

KRMD	KREG	中断功能
0	0	键中断、外部中断 (通过端口电平指定) 注
0	1	外部中断 (通过端口电平指定)
1	0	外部中断 (通道标志指定)
1	1	

注 在检测到下降沿时, 外部中断功能与键中断功能的功能和运行相同。

15.3.2 键返回模式寄存器 (KRM0)

KRM0 寄存器是设定键中断模式的寄存器。

通过 1 位或者 8 位存储器操作指令设定 KRM0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 15-3 键返回模式寄存器 (KRM0) 的格式

地址: FFF37H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	键中断模式的控制 (n=0 ~ 5)
0	不检测键中断信号。
1	检测键中断信号。

- 注意 1. 通过选择下降沿 (KRMD=0) 检测键中断信号 (KRM0n=1) 时，必须通过外部电阻将对应的输入引脚上拉到 V_{DD} 。另外，能通过将键中断输入引脚的 PU01 ~ PU04、PU40、PU125 (上拉电阻选择寄存器 0、4、12 (PU0、PU4、PU12) 中的对象位置“1”，使用内部上拉电阻。
2. 如果在给键中断输入引脚输入低电平 (将 KREG 置“0”时) / 高电平 (将 KREG 置“1”时) 的状态下，将 KRM0、寄存器的对应位置位，就产生中断。
如果要忽视此中断，必须在通过中断屏蔽标志禁止中断处理后设定 KRM0 寄存器。然后，清除中断请求标志，允许中断处理。
3. 在键中断模式中未使用的引脚能用作通常的端口。

15.3.3 键返回标志寄存器 (KRF)

KRF 寄存器是控制键中断标志 (KRF0 ~ KRF5) 的寄存器。

通过 8 位存储器操作指令设定 KRF 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 15-4 键返回标志寄存器 (KRF) 的格式

地址: FFF35H	复位后: 00H	R/W 注						
符号	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	键中断标志 (n=0 ~ 5)
0	未检测到键中断信号
1	检测到键中断信号

注 写“1”的操作无效。如果要清除 KRFn, 必须通过 8 位存储器操作指令将对象位置“0”, 其他位置“1”。

注意 不使用键中断标志 (KRMD=0) 时, 禁止存取 KRF 寄存器。

15.3.4 键中断输入引脚的端口功能控制寄存器

在用作键中断输入引脚时, 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 0、4 (PM0、PM4)、端口模式控制寄存器 0 (PMC0))。详细内容请参照“4.3.1 端口模式寄存器 0、4 (PM0、PM4)”和“4.3.5 端口模式控制寄存器 0 (PMC0)”。

用作键中断输入引脚时的设定例子请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

将与键中断输入引脚复用的端口 (P01/ANI0/SI00/RXD0/SDA00/KR2 等) 用作键中断输入时, 必须将与各端口对应的端口模式寄存器 0 (PM0) 的位置“1”, 将端口模式控制寄存器 0 (PMC0) 的位置“0”。此时, 端口寄存器 0 (P0) 的位可以是“0”也可以是“1”。

例 将 P01/ANI0/SI00/RXD0/SDA00/KR2 用作键中断输入的情况

将端口模式控制寄存器 0 的 PMC01 位置“0”。

将端口模式寄存器 0 的 PM01 位置“1”。

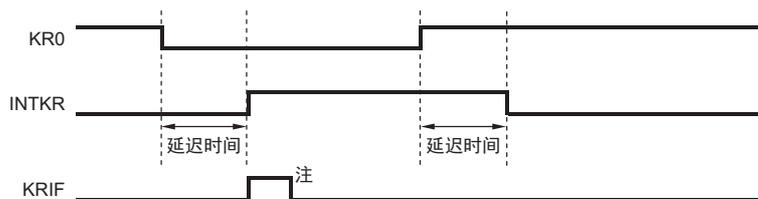
15.4 键中断功能的运行

15.4.1 不使用键中断标志的情况 (KRMD=0)

通过给键中断输入引脚 (KR0 ~ KR5) 输入 KREG 位设定的有效边沿, 产生键中断 (INTKR)。另外, 在产生键中断 (INTKR) 后, 能通过读端口寄存器并且确认输入电平, 确定输入有效边沿的通道。

INTKR 信号与键中断输入引脚 (KR0 ~ KR5) 的输入电平同步发生变化。

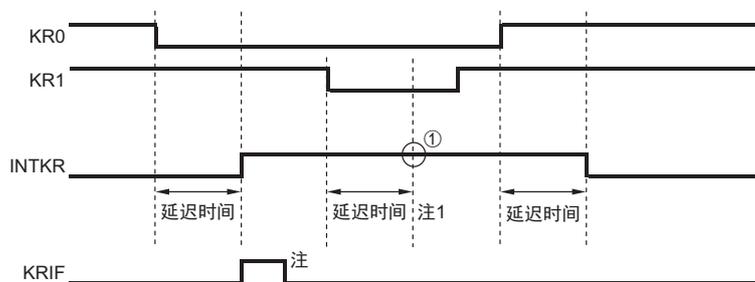
图 15-5 给 1 个通道输入键中断时的 INTKR 信号变化 (KRMD=0、KREG=0 的情况)



注 通过接受向量中断请求或者软件进行清除。

给多个键中断输入引脚输入有效边沿时的运行如图 15-6 所示。因为在给 1 个引脚输入低电平 (KREG=0 时) 的期间 INTKR 信号被置位, 所以在此期间即使给其他引脚输入下降沿也不再产生键中断 (INTKR) (图中的①)。

图 15-6 给多个通道输入键中断时的 INTKR 信号变化 (KRMD=0、KREG=0 的情况)



注 通过接受向量中断请求或者软件进行清除。

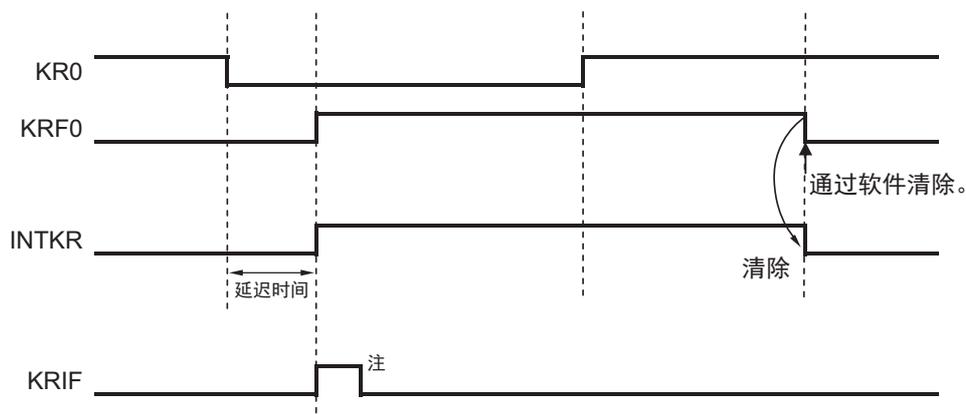
15.4.2 使用键中断标志的情况 (KRMD=1)

通过给键中断输入引脚 (KR0 ~ KR5) 输入 KREG 位设定的有效边沿, 产生键中断 (INTKR)。另外, 在产生键中断 (INTKR) 后, 能通过读键返回标志寄存器 (KRF), 确定输入有效边沿的通道。

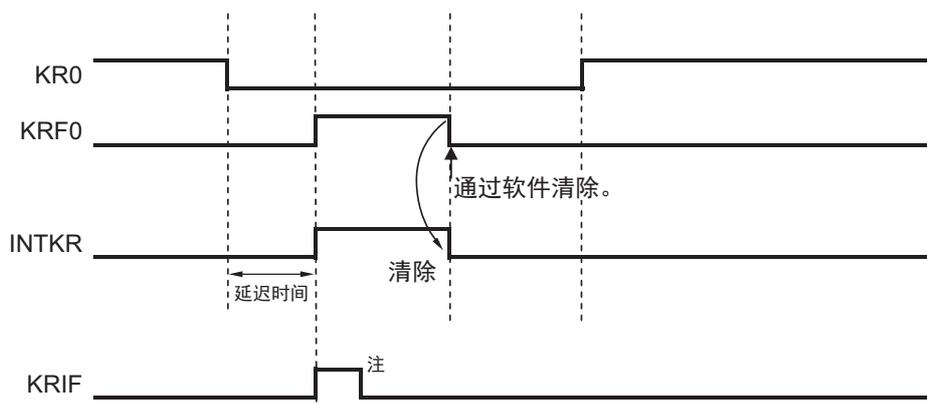
当 KRMD 位为“1”时, 通过清除 KRF 寄存器的对应位, 清除 INTKR 信号。如图 15-7 所示, 对于 1 个通道的 1 个下降沿 (KREG=0 时), 只产生 1 次中断, 与 KRFn 位的清除时序是否在上升沿的前后无关。

图 15-7 使用键中断标志时的 INTKR 信号基本变化 (KRMD=1、KREG=0 的情况)

(a) 给 KR0 引脚输入上升沿后的 KRF0 的清除



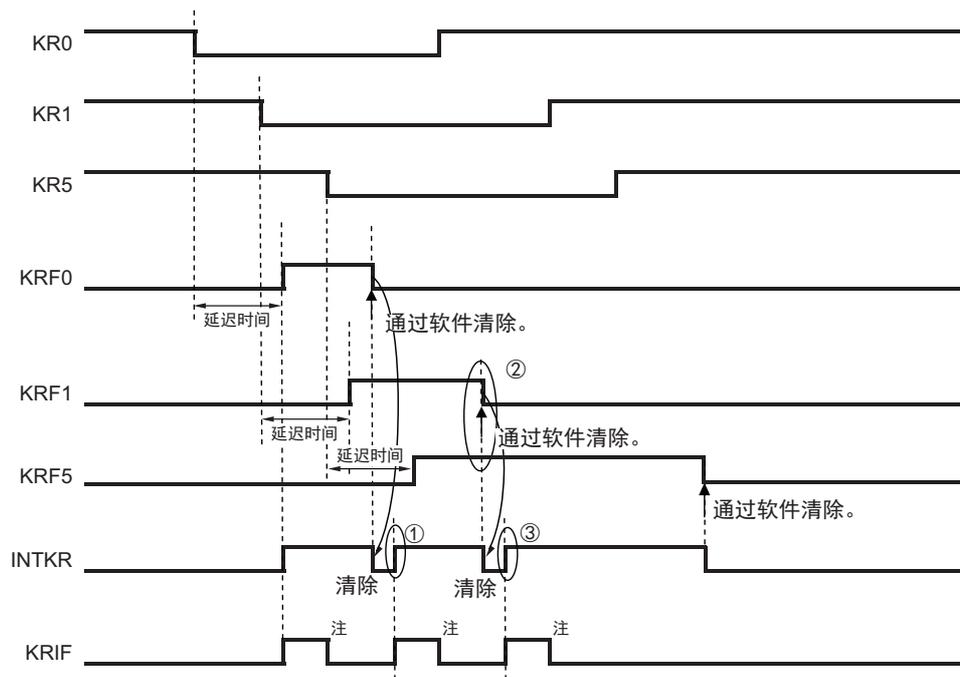
(b) 给 KR0 引脚输入上升沿前的 KRF0 的清除



注 通过接受向量中断请求或者软件进行清除。

给多个键中断输入引脚输入有效边沿时的运行如图 15-8 所示。在给 KR0 引脚输入下降沿后也给 KR1 引脚和 KR5 引脚输入下降沿（KREG=0 时）。因为在清除 KRF0 位时 KRF1 位已被置位，所以在清除 KRF0 位的 1 个时钟（ f_{CLK} ）后产生键中断（INTKR）（图中的①）。另外，因为给 KR5 引脚输入下降沿并且在清除 KRF5 位后（图中的②）给 KR5 引脚输入低电平，所以在清除 KRF1 位的 1 个时钟（ f_{CLK} ）后产生键中断（INTKR）（图中的③）。由此，对于多个通道输入的有效边沿，有可能产生键中断（INTKR）。

图 15-8 给多个通道输入键中断时的 INTKR 信号变化（KRMD=1、KREG=0 的情况）



注 通过接受向量中断请求或者软件进行清除。

备注 f_{CLK} : CPU/ 外围硬件的时钟频率

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 2 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。能大幅度地降低 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。

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 X1 振荡或者 EXCLK 输入运行的外围硬件后执行 STOP 指令注。

2. 为了降低 A/D 转换器的工作电流，必须将 A/D 转换器的模式寄存器 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。
3. 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 19 章 选项字节”。

注 只限于 16 引脚产品。

16.2 待机功能的控制寄存器

待机功能的控制寄存器如下所示。

有关各寄存器的详细内容，请参照“第 5 章 时钟发生电路”。

HALT/STOP 模式中控制允许或者停止低速内部振荡器的运行的寄存器。

- 运行速度模式控制寄存器 (OSMC)

在解除 STOP 模式时，控制 X1 时钟的振荡稳定时间的寄存器。

- 振荡稳定时间计数器的状态寄存器 (OSTC) 注
- 振荡稳定时间选择寄存器 (OSTS) 注

注 只限于 16 引脚产品。

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（只限于 16 引脚产品）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 在中断屏蔽标志为“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，在此状态下，即使执行 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		在 CPU 以主系统时钟运行的过程中执行 HALT 指令的情况			
项目			CPU 以高速内部振荡器时钟 (f_{IH}) 运行	CPU 以 X1 时钟 (f_X) 运行	CPU 以外部主系统时钟 (f_{EX}) 运行
	系统时钟		停止给 CPU 提供时钟。		
主系统时钟	f_{IH}		继续运行（不能停止）。	禁止运行。	
		f_X	禁止运行。	继续运行（不能停止）。	不能运行。
	f_{EX}			不能运行。	继续运行（不能停止）。
f_{IL}	通过选项字节（000C0H）的 bit0（WDSTBYON）和 bit4（WDTON）以及运行速度模式控制寄存器（OSMC）的 bit4（WUTMMCK0）进行设定。 <ul style="list-style-type: none"> • WUTMMCK0=1：振荡 • WUTMMCK0=0 并且 WDTON=0：停止 • WUTMMCK0=0、WDTON=1 并且 WDSTBYON=1：振荡 • WUTMMCK0=0、WDTON=1 并且 WDSTBYON=0：停止 				
CPU		停止运行。			
代码闪存					
RAM					
端口（锁存器）					
定时器阵列单元		能运行。			
12 位间隔定时器注					
看门狗定时器					
时钟输出 / 蜂鸣器输出		能运行。			
A/D 转换器					
比较器注					
串行阵列单元（SAU）					
串行接口（IICA）注					
可选择的上电复位功能					
外部中断					
键中断功能					

注 只限于 16 引脚产品。

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_{IL} ：低速内部振荡器时钟

f_X ：X1 时钟注

f_{EX} ：外部主系统时钟注

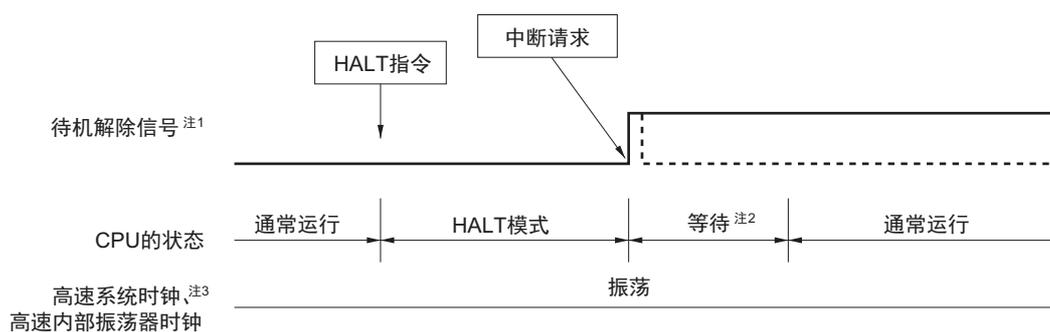
(2) HALT 模式的解除

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解除 HALT 模式

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 16-1 通过产生中断请求解除 HALT 模式



注 1. 有关待机解除信号的详细内容，请参照“图 14-1 中断功能的基本结构”。

2. HALT 模式解除的等待时间

- 进行向量中断处理的情况：28 ~ 29 个时钟
- 不进行向量中断处理的情况：20 ~ 21 个时钟

3. 只限于 16 引脚产品。

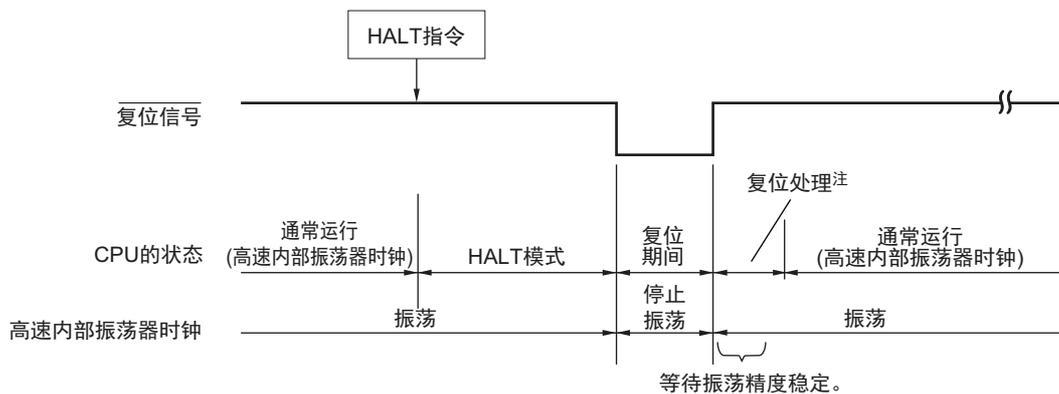
备注 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号解除 HALT 模式

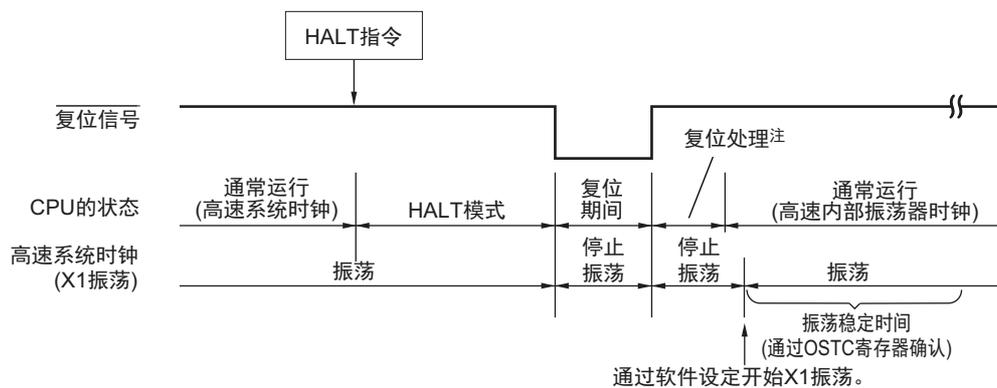
通过产生复位信号来解除 HALT 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

图 16-2 通过产生复位信号解除 HALT 模式

(1) CPU时钟为高速内部振荡器时钟的情况



(2) CPU时钟为高速系统时钟的情况（只限于 16 引脚产品）



注 有关复位处理时间，请参照“第 17 章 复位功能”。有关 SPOR 电路的复位处理时间，请参照“第 18 章 可选择的上电复位电路”。

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。

注意 因为在中断屏蔽标志为“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，所以在此状态下执行 STOP 指令时，一旦转移到 STOP 模式，然后立即被解除。因此，在执行 STOP 指令后，经过 STOP 模式解除时间，就返回通常的运行模式。

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		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_{IH}) 运行	CPU 以 X1 时钟 (f_X) 运行	CPU 以外部主系统时钟 (f_{EX}) 运行
系统时钟		停止给 CPU 提供时钟。		
主系统时钟	f_{IH}	停止		
	f_X			
	f_{EX}			
f_{IL}		通过选项字节 (000C0H) 的 bit0 (WDSTBYON) 和 bit4 (WDTON) 以及运行速度模式控制寄存器 (OSMC) 的 bit4 (WUTMMCK0) 进行设定。 <ul style="list-style-type: none"> • WUTMMCK0=1: 振荡 • WUTMMCK0=0 并且 WDTON=0: 停止 • WUTMMCK0=0、WDTON=1 并且 WDSTBYON=1: 振荡 • WUTMMCK0=0、WDTON=1 并且 WDSTBYON=0: 停止 		
CPU		停止运行。		
代码闪存		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。		
12 位间隔定时器 ^注		能运行。		
看门狗定时器		通过选项字节 (000C0H) 的 bit0 (WDSTBYON) 进行设定。 WDSTBYON=0: 停止运行 WDSTBYON=1: 继续运行 (不能停止)		
时钟输出 / 蜂鸣器输出		禁止运行。		
A/D 转换器		禁止运行。		
比较器 ^注		能运行 (只限于未使用数字滤波器时)。		
串行阵列单元 (SAU)		禁止运行。		
串行阵列单元 (IICA) ^注		能通过地址匹配进行唤醒。		
可选择的上电复位功能		能运行。		
外部中断		禁止运行。		
键中断功能		禁止运行。		

注 只限于 16 引脚产品。

备注 停止运行: 在转移到 STOP 模式时自动停止运行。

禁止运行: 在转移到 STOP 模式前停止运行。

f_{IH} : 高速内部振荡器时钟

f_{IL} : 低速内部振荡器时钟

f_X : X1 时钟^注

f_{EX} : 外部主系统时钟^注

(2) STOP 模式的解除

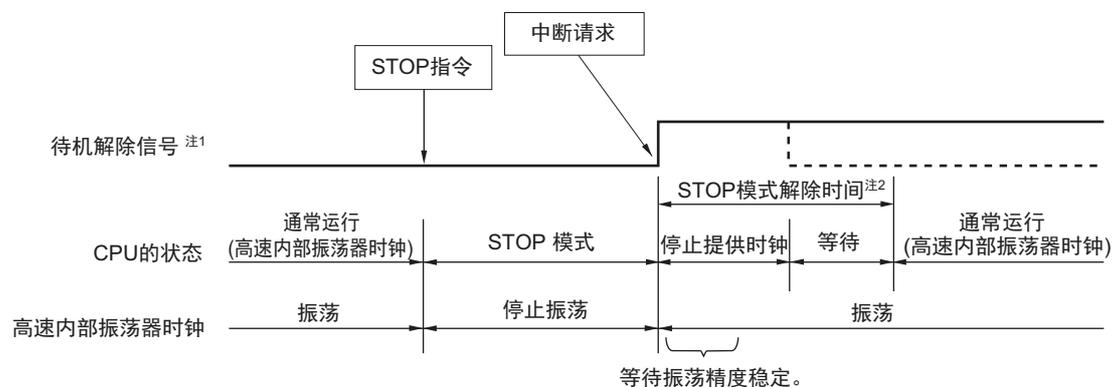
能通过以上 2 种方法解除 STOP 模式。

(a) 通过未屏蔽的中断请求解除 STOP 模式

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 16-3 通过产生中断请求解除 STOP 模式 (1/2)

(1) 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 14-1 中断功能的基本结构”。

2. STOP 模式的解除时间

停止提供时钟：27 μ s(TYP.)

等待

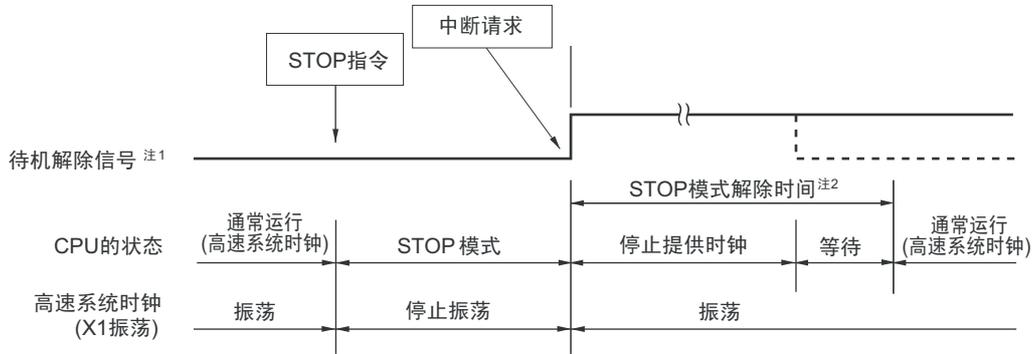
- 进行向量中断处理的情况： 11 个时钟
- 不进行向量中断处理的情况： 3 个时钟

备注 1. 时钟提供的停止时间因温度条件和 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 16-3 通过中断请求解除 STOP 模式 (2/2)

(2) CPU时钟为高速系统时钟 (X1 振荡) 的情况 (只限于 16 引脚产品)



注 1. 有关待机解除信号的详细内容, 请参照“图 14-1 中断功能的基本结构”。

2. STOP 模式的解除时间

停止提供时钟: 27μs(TYP.) 和振荡稳定时间 (通过 OSTS 进行设定) 中较长的时间”
等待

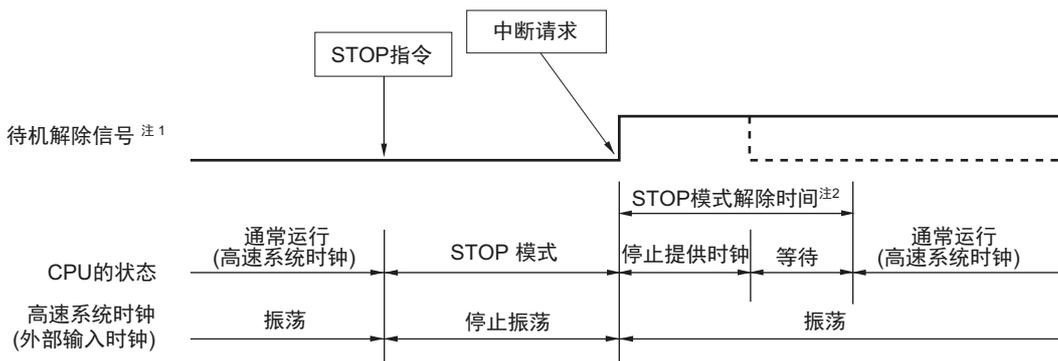
- 进行向量中断处理的情况: 14 ~ 15 个时钟
- 不进行向量中断处理的情况: 6 ~ 7 个时钟

注意 在 CPU 以高速系统时钟 (X1 振荡) 运行, 并且想要缩短解除 STOP 模式后的振荡稳定时间时, 必须在执行 STOP 指令前, 将 CPU 时钟暂时切换到高速内部振荡器时钟。

备注 1. 时钟提供的停止时间因温度条件和 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。

(3) CPU时钟为高速系统时钟 (外部时钟输入) 的情况 (只限于 16 引脚产品)



注 1. 有关待机解除信号的详细内容, 请参照“图 14-1 中断功能的基本结构”。

2. STOP 模式的解除时间: $2^4/f_{EX}$

等待

- 进行向量中断处理的情况: 11 个时钟
- 不进行向量中断处理的情况: 3 个时钟

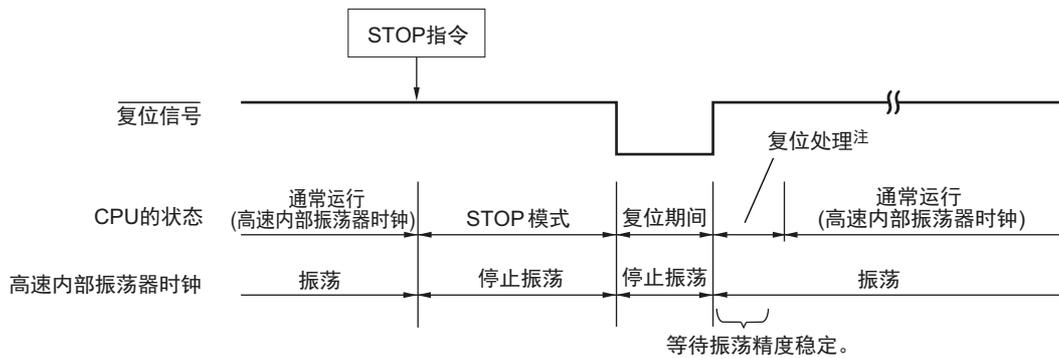
备注 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号解除 STOP 模式

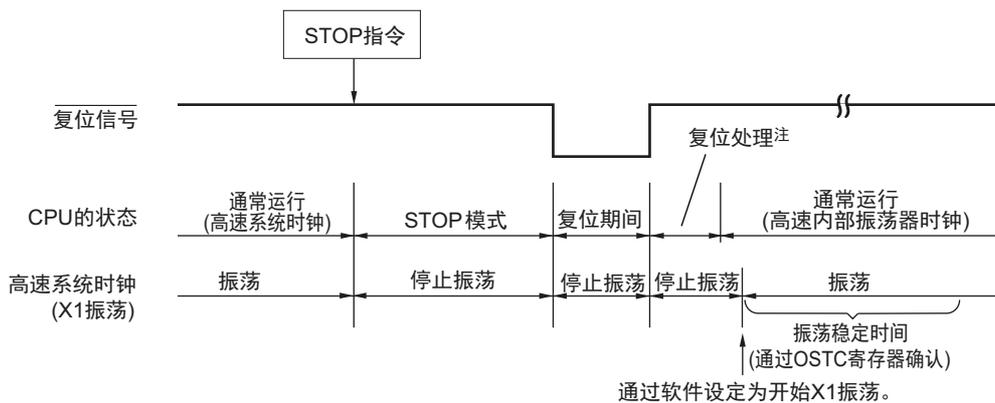
通过产生复位信号来解除 STOP 模式。然后，和通常的复位一样，在转移到复位向量地址后执行程序。

图 16-4 通过产生复位信号解除 STOP 模式

(1) CPU时钟为高速内部振荡器时钟的情况



(2) CPU时钟为高速系统时钟的情况（只限于 16 引脚产品）



注 有关复位处理时间，请参照“第 17 章 复位功能”。有关 SPOR 电路的复位处理时间，请参照“第 18 章 可选择的上电复位电路”。

第 17 章 复位功能

能产生复位信号的复位源有以下 5 种。

- (1) 因 $\overline{\text{RESET}}$ 引脚输入引起的外部复位。
- (2) 因检测到看门狗定时器的程序失控引起的内部复位。
- (3) 因可选择的上电复位 (SPOR) 电路的电源电压和检测电压的比较引起的内部复位。
- (4) 因执行非法指令引起的内部复位注。
- (5) 因数据保持电源电压引起的内部复位。

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。

注 当执行指令码 FFH 时，发生非法指令错误。

在通过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。

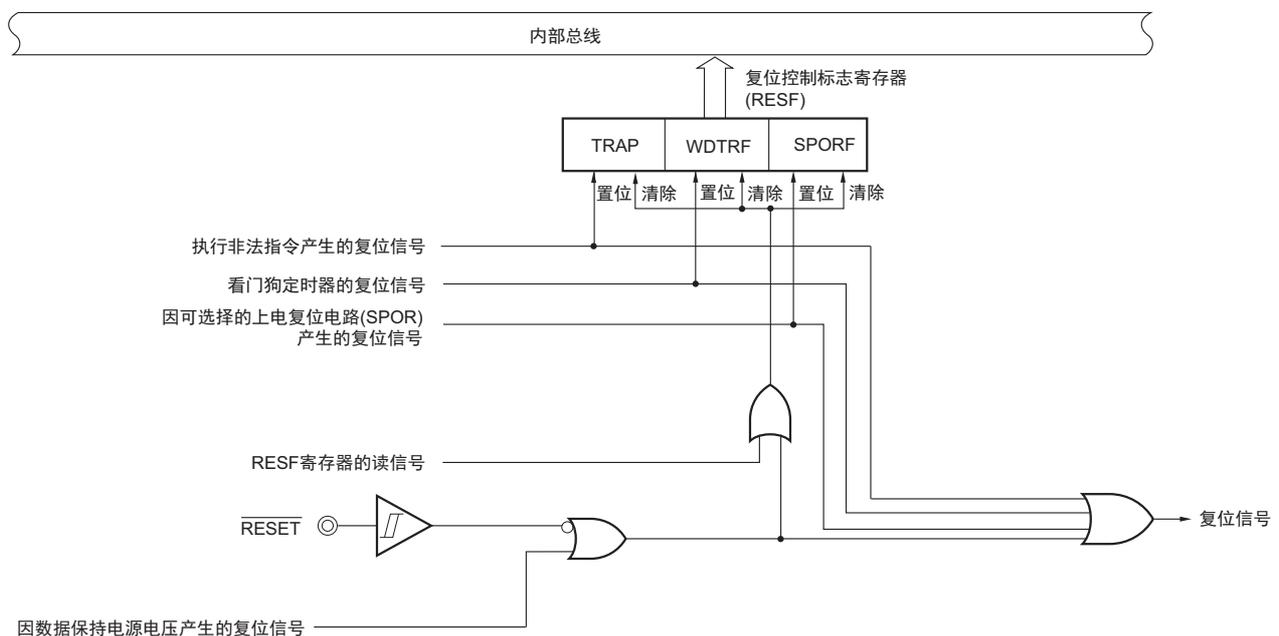
注意 1. 要进行外部复位时，必须通过选项字节 (000C1H) 设定 PORTSELB=1，然后选择 P125 作为 $\overline{\text{RESET}}$ 引脚并且至少输入 10 μs 的低电平。

(如果要在上电时进行外部复位，就必须在给 $\overline{\text{RESET}}$ 引脚输入低电平后接通电源，并且在“24.4 AC 特性”所示的工作电压范围内的期间至少维持 10 μs 的低电平，然后输入高电平。)

2. 在复位信号有效期间，停止 X1 时钟注、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟注的输入无效。
3. 如果发生复位，就对各 SFR 和 2nd SFR 进行初始化，端口引脚的状态如下变化。
 - P40：在外部复位期间或者因数据保持电源电压引起的复位期间为高阻抗。在其他复位期间和接受复位后为高电平（连接内部上拉电阻）。
 - P125：在外部复位期间为低电平（给 $\overline{\text{RESET}}$ 引脚输入低电平）。在其他复位期间和接受复位后为高电平（连接内部上拉电阻）。
 - P40、P125 以外的端口：在复位期间和接受复位后为高阻抗。

注 只限于 16 引脚产品。

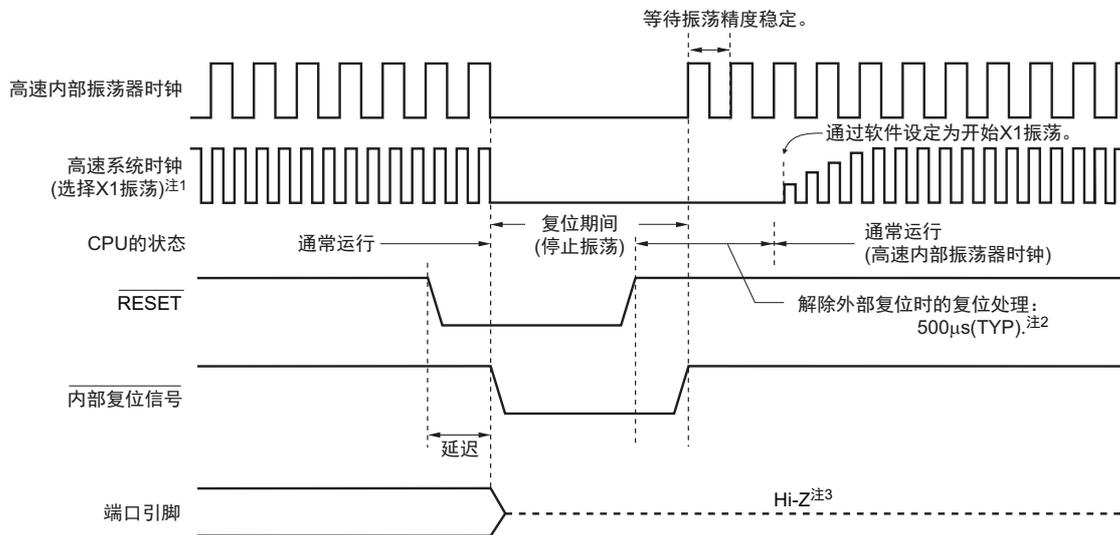
图 17-1 复位功能的框图



17.1 复位运行的时序

如果给 $\overline{\text{RESET}}$ 引脚输入低电平，就发生复位，如果给 $\overline{\text{RESET}}$ 引脚输入高电平，就解除复位。复位处理后，通过高速内部振荡器时钟开始执行程序。

图 17-2 因 $\overline{\text{RESET}}$ 引脚输入引起的复位时序



注 1. 只限于 16 引脚产品。

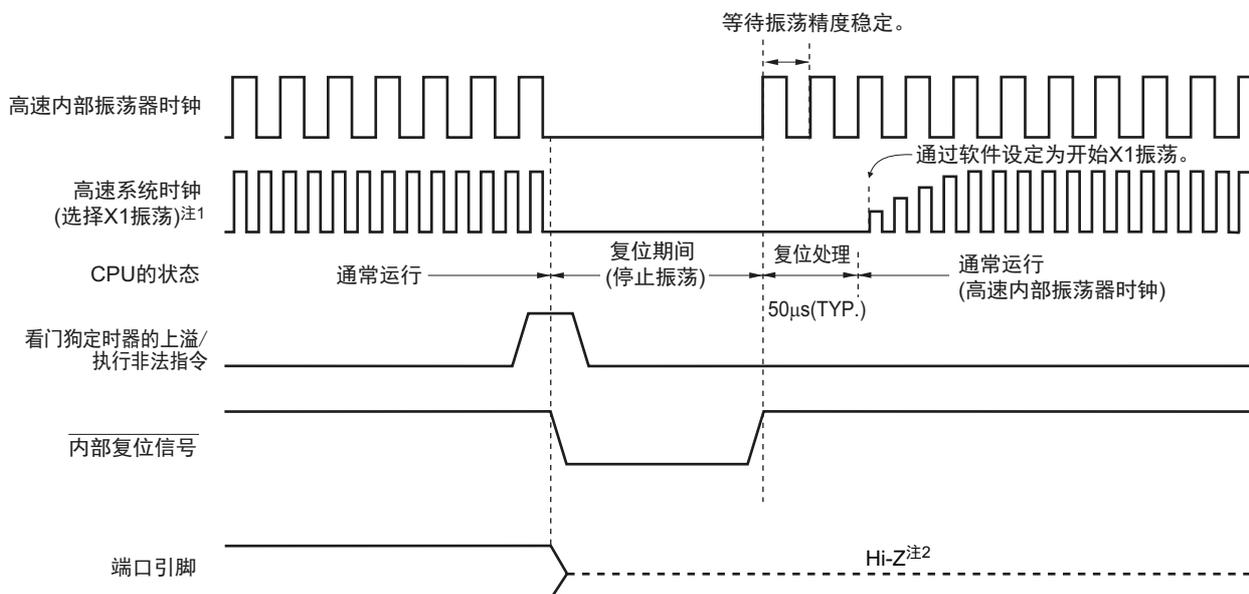
2. 上电时，在解除外部复位时的复位处理时间前需要 SPOR 复位处理时间 (MAX.3.01ms)

3. P40 引脚的状态如下所示。

- 在外部复位期间或者因数据保持电源电压引起的复位期间为高阻抗。
- 在接受复位后为高电平 (连接内部上拉电阻)

由看门狗定时器的上溢 / 执行非法指令而引起的复位自动被解除，并且在复位处理后，通过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 / 执行非法指令引起的复位时序



注 1. 只限于 16 引脚产品。

2. P40 引脚和 P125 引脚的状态如下所示。

- 在复位期间和接受复位后为高电平（连接内部上拉电阻）

备注 有关因可选择的上电复位（SPOR）电路的电压检测引起的复位时序，请参照“第 18 章 可选择的上电复位电路”。

17.2 复位期间的运行状态

复位期间的运行状态如表 17-1 所示，接受复位后的各硬件的状态如表 17-2 所示。

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_{IH}	停止运行	
	f_X	停止运行（X1 引脚和 X2 引脚处于输入端口模式）	
	f_{EX}	时钟输入无效（引脚处于输入端口模式）	
f_{IL}		停止运行	
CPU		停止运行	
代码闪存		停止运行	
RAM		停止运行	
端口（锁存器）		高阻抗注 2	
定时器阵列单元		停止运行	
12 位间隔定时器			
看门狗定时器			
时钟输出 / 蜂鸣器输出			
A/D 转换器			
比较器注 1			
串行阵列单元（SAU）			
串行接口（IICA）			
可选择的上电复位功能			能进行检测运行
外部中断			停止运行
键中断功能			

注 1. 只限于 16 引脚产品。

2. 端口引脚 P40、P125 的状态如下所示。

- P40：在外部复位期间或者因数据保持电源电压引起的复位期间为高阻抗，在其他复位期间和接受复位后变为高电平（连接内部上拉电阻）。
- P125：在外部复位期间为低电平（给 $\overline{\text{RESET}}$ 引脚输入低电平），在其他复位期间和接受复位后为高电平（连接内部上拉电阻）。

备注 f_{IH} : 高速内部振荡器时钟
 f_X : X1 振荡时钟
 f_{EX} : 外部主系统时钟
 f_{IL} : 低速内部振荡器时钟

表 17-2 接受复位后的各硬件状态

硬件		接受复位后的状态注
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

注 在产生复位信号期间和等待振荡稳定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。

备注 有关特殊功能寄存器 (SFR: Special Function Register) 在接受复位后的状态，请参照“3.1.4 特殊功能寄存器 (SFR: Special Function Register) 的区域”和“3.1.5 扩展特殊功能寄存器 (2nd SFR: 2nd Special Function Register) 的区域”。

17.3 复位源的确认寄存器

17.3.1 复位控制标志寄存器 (RESF)

RL78 微控制器存在多种内部复位发生源。复位控制标志寄存器 (RESF) 保存发生复位请求的复位源。能通过 8 位存储器操作指令读 RESF 寄存器。

能通过外部复位、因数据保持下限电压引起的复位以及读取 RESF 寄存器，清除 TRAP、WDTRF 和 SPORF 标志。

图 17-4 复位控制标志寄存器 (RESF) 的格式

地址: FFFA8H	复位后: 不定 ^{注1}							R
符号	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	0	0	SPORF

TRAP	因执行非法指令产生的内部复位请求 ^{注2}
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	因看门狗定时器 (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。

SPORF	因可选择的上电复位 (SPOR) 电路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。

2. 当执行指令码 FFH 时，发生非法指令错误。

在通过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 不能通过 1 位存储器操作指令读数据。

产生复位请求时的 RESF 寄存器状态如表 17-3 所示。

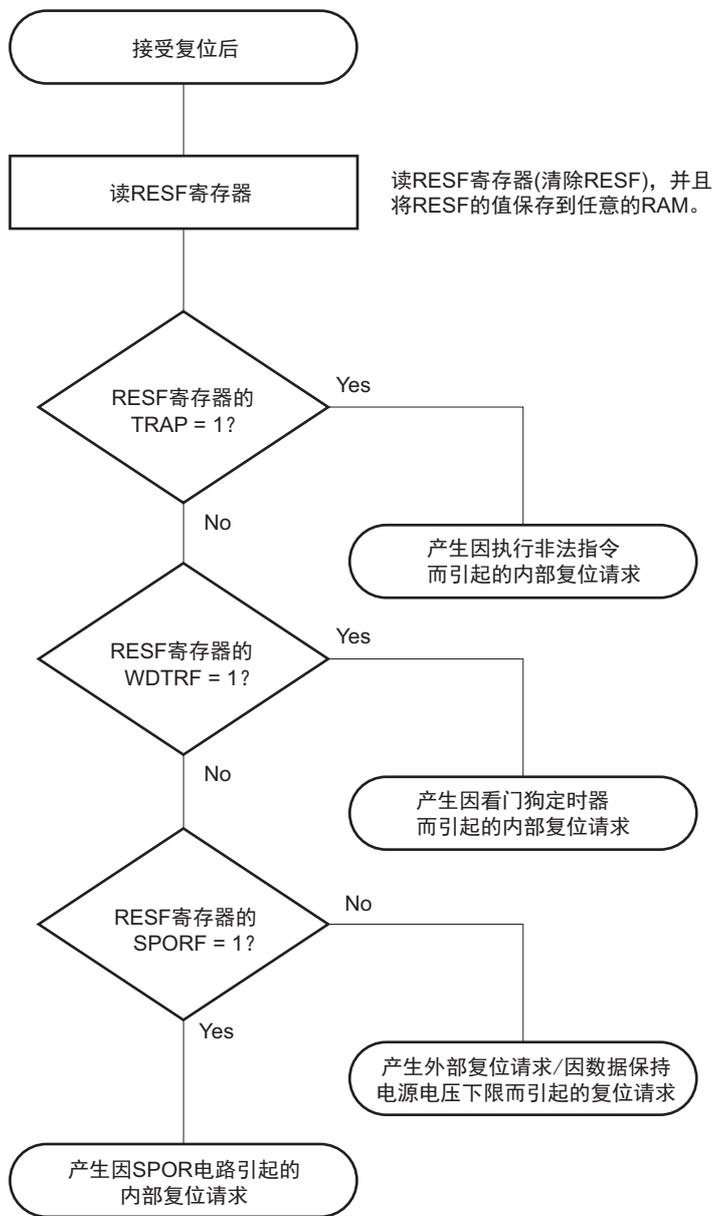
表 17-3 产生复位请求时的 RESF 寄存器的状态

复位源 标志	RESET 输入	因执行非法指令 引起的复位	因 WDT 引起的复位	因 SPOR 引起的复位	因数据保持下限 电压引起的复位
TRAP	清“0”	置“1”	保持	保持	清“0”
WDTRF		保持	置“1”	保持	
SPORF		保持	保持	置“1”	

如果通过 8 位存储器操作指令读 RESF 寄存器，就自动清除 RESF 寄存器。

复位源的确认步骤如图 17-5 所示。

图 17-5 复位源的确认步骤例子



第 18 章 可选择的上电复位电路

18.1 可选择的上电复位电路的功能

可选择的上电复位（SPOR）电路有以下功能。

- 在接通电源时产生内部复位信号。
如果电源电压（ V_{DD} ）大于检测电压（ V_{SPOR} ）（ $V_{DD} \geq V_{SPOR}$ ），就解除复位。
- 将电源电压（ V_{DD} ）和检测电压（ V_{SPDR} ）进行比较，当 $V_{DD} < V_{SPDR}$ 时，产生内部复位信号。
- 能通过选项字节（000C1H）从 4 种检测电平中选择电源电压的检测电压（ V_{SPOR} 、 V_{SPDR} ）。（参照“19.2 用户选项字节的格式”。）

如果发生复位，复位控制标志寄存器（RESF）的 bit0（SPORF）就被置“1”。有过 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

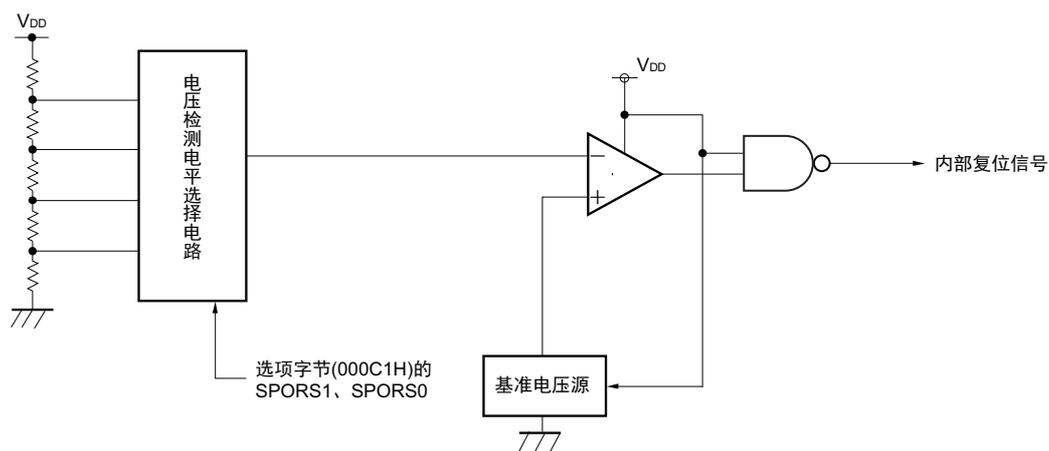
注意 保持复位控制标志寄存器（RESF）的各标志直到数据保持下限电压为止。

备注 V_{SPOR} : SPOR 电源上升检测电压
 V_{SPDR} : SPOR 电源下降检测电压
 详细内容请参照“24.6.4 SPOR 电路特性”。

18.2 可选择的上电复位电路的结构

可选择的上电复位电路的框图如图 18-1 所示。

图 18-1 可选择的上电复位电路的框图



18.3 可选择的上电复位电路的运行

通过选项字节 000C1H 设定电压检测电平。

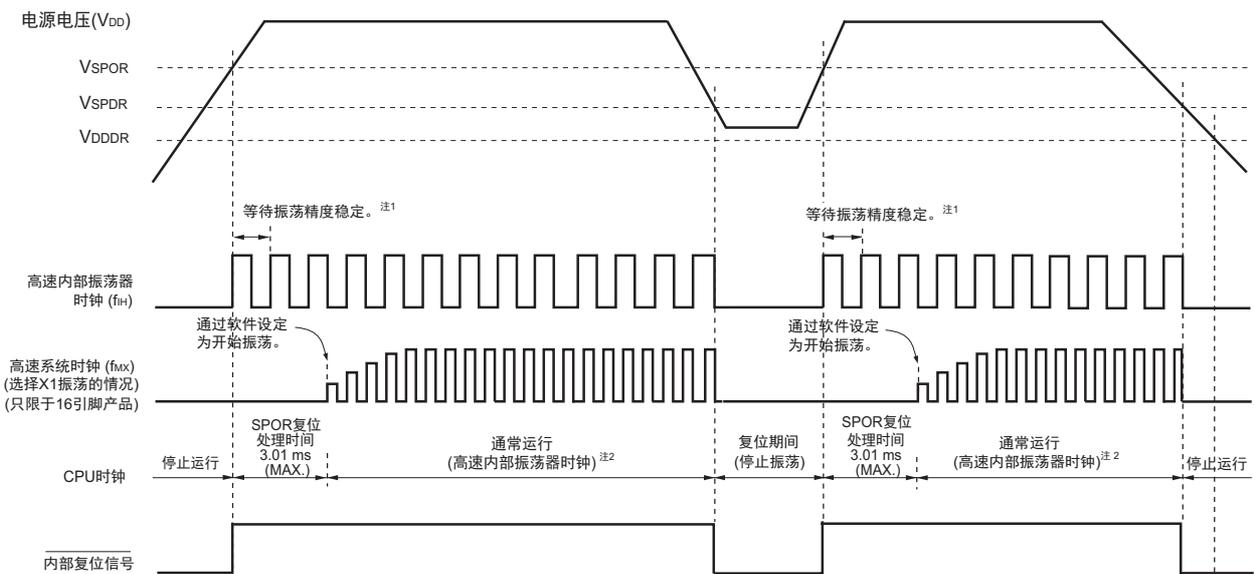
接通电源是产生内部复位信号。

电源电压 (V_{DD}) 保持内部复位状态直到大于检测电压 (V_{SPOR})。如果电源电压 (V_{DD}) 大于检测电压 (V_{SPOR})，就解除内部复位。

在工作电压下降时，如果电源电压 (V_{DD}) 小于检测电压 (V_{SPDR})，就发生内部复位。

可选择的上电复位电路的内部复位信号的产生时序如图 18-2 所示。

图 18-2 内部复位信号的产生时序



- 注 1. 内部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（只限于 16 引脚产品）。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。

备注 V_{SPOR} : SPOR 电源上升检测电压

V_{SPDR} : SPOR 电源下降检测电压

V_{DDDR} : 数据保持电源电压

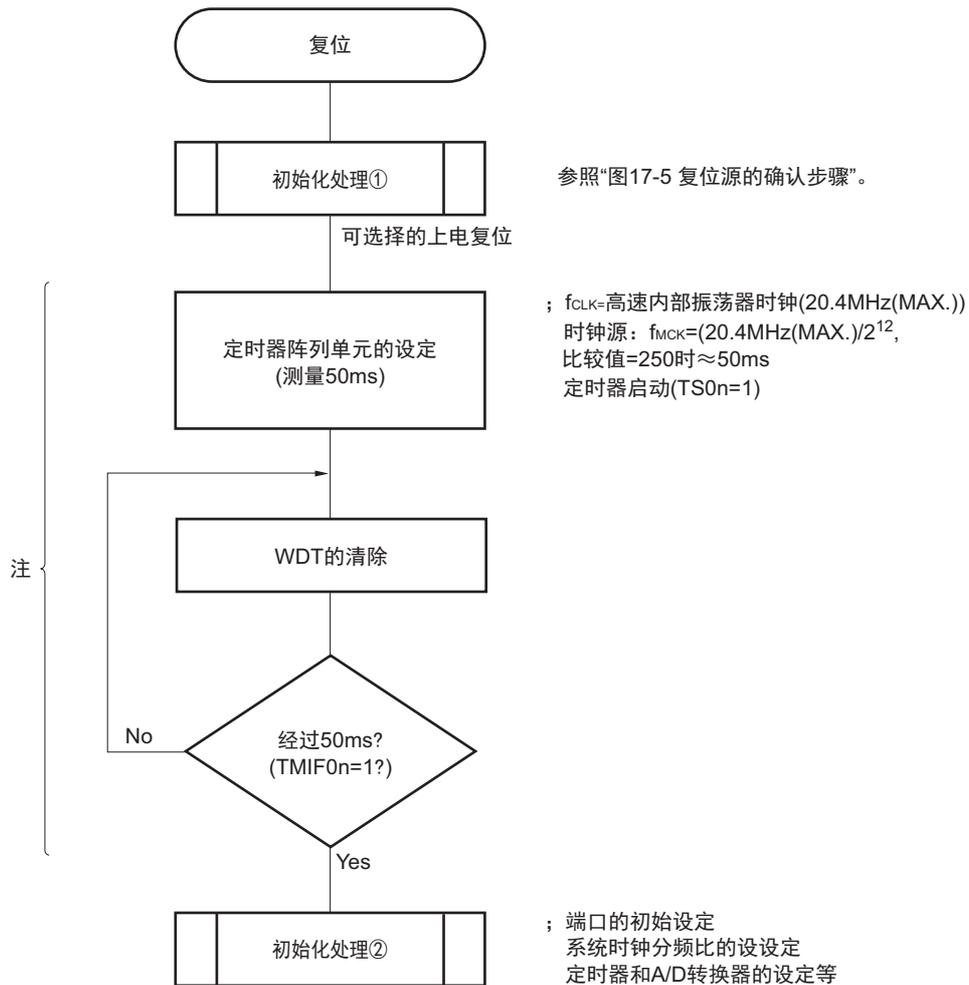
18.4 可选择的上电复位电路的注意事项

在因电源电压 (V_{DD}) 接近 SPOR 检测电压 (V_{SPOR} 、 V_{SPDR}) 导致某期间出现波动的系统中, 可能会重复出现复位状态 / 复位解除状态。可通过以下措施任意设定从复位解除到开始单片机运行的时间。

< 措施 >

复位解除后, 通过使用定时器等, 等待因系统而不同的电源电压波动期间后, 对端口等进行初始设置。

图 18-3 接近检测电压的电源电压波动小于 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 n: 通道号。

10 引脚产品时: n=0、1

16 引脚产品时: n=0 ~ 3

第 19 章 选项字节

19.1 选项字节的功能

RL78/G10 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。

注意 与是否使用各功能无关，必须设定选项字节。

19.1.1 用户选项字节（000C0H ~ 000C2H）

(1) 000C0H

- 看门狗定时器的运行
 - 允许或者停止计数器的运行。
 - 在 HALT/STOP 模式中停止或者允许计数器的运行。
- 看门狗定时器的时间设定
 - 看门狗定时器的上溢时间的设定。
 - 看门狗定时器的间隔中断的设定

(2) 000C1H

- SPOR 检测电平（ V_{SPOR} ）的设定
- P125/KR1/ \overline{RESET} 引脚的控制
 - 选择 P125/KR1/或者 \overline{RESET} 。

(3) 000C2H

- 高速内部振荡器的频率设定
 - 从 1.25 ~ 20MHz 的高速内部振荡器时钟中选择。

19.1.2 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

19.2 用户选项字节的格式

图 19-1 用户选项字节（000C0H）的格式

地址：000C0H

7	6	5	4	3	2	1	0
1	1	1	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$)	看门狗定时器的间隔中断时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$)
0	0	0	$(2^6-1)/f_{IL}$ (3.65ms)	$2^6/f_{IL}\times 0.75$ (2.78ms)
0	0	1	$(2^7-1)/f_{IL}$ (7.36ms)	$2^7/f_{IL}\times 0.75$ (5.56ms)
0	1	0	$(2^8-1)/f_{IL}$ (14.7ms)	$2^8/f_{IL}\times 0.75$ (11.1ms)
0	1	1	$(2^9-1)/f_{IL}$ (29.6ms)	$2^9/f_{IL}\times 0.75$ (22.2ms)
1	0	0	$(2^{11}-1)/f_{IL}$ (118ms)	$2^{11}/f_{IL}\times 0.75$ (89.0ms)
1	0	1	$(2^{13}-1)/f_{IL}$ (474ms)	$2^{13}/f_{IL}\times 0.75$ (356ms)
1	1	0	$(2^{14}-1)/f_{IL}$ (949ms)	$2^{14}/f_{IL}\times 0.75$ (712ms)
1	1	1	$(2^{16}-1)/f_{IL}$ (3799ms)	$2^{16}/f_{IL}\times 0.75$ (2849ms)

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，停止计数器运行。
1	在 HALT/STOP 模式中，允许计数器运行。

注意 1. 必须给 bit7 ~ 5 写“1”。

2. 禁止设定为 WDTON=0、WDSTBYON=1。

3. 总是发生看门狗定时器的间隔中断。不使用看门狗定时器的间隔中断时，必须将 WDTIMK 位置“1”，禁止中断处理。

备注 f_{IL} ：低速内部振荡器时钟频率

图 19-2 用户选项字节（000C1H）的格式

地址：000C1H

7	6	5	4	3	2	1	0
1	1	1	PORTSELB	SPORS1	SPORS0	1	1

- SPOR 检测电压的设定（中断&复位模式）

检测电压 (V_{SPOR})		选项字节的设定值	
上升	下降	SPORS1	SPORS0
4.28V	4.20V	0	0
2.90V	2.84V	0	1
2.57V	2.52V	1	0
2.16V	2.11V	1	1

- P125/KR1/ $\overline{\text{RESET}}$ 引脚的设定

PORTSELB	P125/KR1/ $\overline{\text{RESET}}$ 引脚的控制
0	端口功能 (P125/KR1)
1	$\overline{\text{RESET}}$ 输入 (内部上拉电阻总是有效)

注意 1. 必须给 bit7 ~ 5、1、0 写“1”。

- 在工作电压范围内设定检测电压 (V_{SPOR})。工作电压范围取决于用户选项字节 (000C2H) 的设定。

工作电压范围如下所示。

CPU 工作频率为 1MHz ~ 20MHz: $V_{DD}=2.7 \sim 5.5V$

CPU 工作频率为 1MHz ~ 5MHz: $V_{DD}=2.0 \sim 5.5V$

备注 1. 有关 SPOR 电路的详细内容，请参照“第 18 章 可选择的上电复位电路”。

- 检测电压为 TYP. 值。详细内容请参照“24.6.4 SPOR 电路特性”。

图 19-3 选项字节（000C2H）的格式

地址：000C2H

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	选择高速内部振荡器时钟的频率	
			工作频率 (f_{MAIN})	工作电压范围 (V_{DD})
0	0	1	20MHz	2.7V ~ 5.5V
0	1	0	10MHz	
0	1	1	5MHz	2.0V ~ 5.5V 注
1	0	0	2.5MHz	
1	0	1	1.25MHz	
上述以外			禁止设定。	

注 因为包含可选择上电复位 (SPOR) 电路的检测电压 (V_{SPOR})，所以必须在 2.25 ~ 5.5V 的电压范围下使用。

注意 必须给 bit7 ~ 3 写“1”。

19.3 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 19-4 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	0	1	0	1

OCDENSET	片上调试运行的控制
0	禁止片上调试运行。
1	允许片上调试运行注

注 在片上调试安全 ID 认证失败时，不消除闪存的数据。

注意 只能指定 bit7（OCDENSET）的值。
必须给 bit6 ~ 0 写“0000101B”。

备注 bit1 和 bit3 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）。

19.4 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	F7H	; 允许看门狗定时器运行。 ; 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	E7H	; V _{SPDR} 选择 2.7V。 ; 使用端口功能（P125/KR1）
	DB	FDH	; 选择高速内部振荡器的时钟频率 1.25MHz。
	DB	85H	; 允许片上调试运行。

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 20 章 闪存

RL78 微控制器内置了可进行编程、擦除和重写的闪存。闪存有可执行程序的“代码闪存”和数据保存区的“数据闪存”。

注意 闪存编程时的工作电压为 4.5 ~ 5.5V。



闪存的编程方法有以下几种。

能通过闪存编程器或者外部器件（UART 通信）进行的串行编程改写代码闪存。

- 使用闪存编程器进行的串行编程（参照 20.1）
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）进行的串行编程（参照 20.2）
能使用外部器件（单片机和 ASIC）的 UART 通信进行板上编程。

20.1 通过闪存编程器进行的串行编程

能使用以下专用的闪存编程器对 RL78 微控制器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 RL78 微控制器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

能在将 RL78 微控制器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 20-1 RL78/G10 和专用闪存编程器的连线表

专用闪存编程器的连接引脚		输入 / 输出	引脚功能	R78/G10 的 引脚名	10 引脚产品 的引脚号	16 引脚产 品的引脚号
信号名						
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	1	2
SI/RxD	—	输入 / 输出	发送 / 接收信号			
—	$\overline{\text{RESET}}$	输出	复位信号	$\overline{\text{RESET}}$	2	3
/RESET	—	输出				
V_{DD}		输入 / 输出	V_{DD} 电压生成 / 电源监视	V_{DD}	5	8
GND		—	接地	V_{SS}	4	7
EMV_{DD}		—	TOOL0 引脚驱动电源	V_{DD}	5	8

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置为开路。

有关 RL78 微控制器和连接器的连接例子，请参照各编程器的用户手册。与 E1 的连接请参照“21.1 与 E1 片上调试仿真器的连接”。

20.1.1 编程环境

RL78 微控制器的闪存的编程环境如下所示。

图 20-1 闪存的编程环境



需要控制专用闪存编程器的主机。

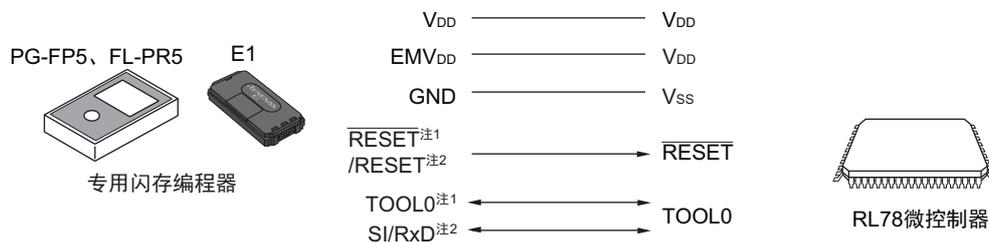
专用闪存编程器和 RL78 微控制器的接口使用 TOOL0 引脚，通过专用的单线 UART 进行编程和擦除。

20.1.2 通信方式

专用闪存编程器和 RL78 微控制器的通信使用 RL78 微控制器的 TOOL0 引脚，通过专用的单线 UART 进行串行通信。

传送速率：固定为 115200bps

图 20-2 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。

20.2 通过外部器件（内置 UART）进行的编程

能使用 RL78 微控制器和与 UART 连接的外部器件（单片机或者 ASIC）对内部闪存进行数据的板上编程。

20.2.1 编程环境

RL78 微控制器的闪存的编程环境如下所示。

图 20-3 闪存的编程环境



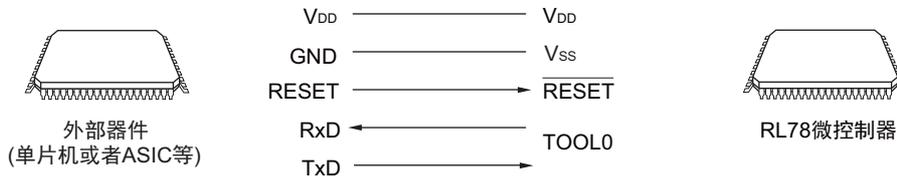
外部器件对 RL78 微控制器进行板上编程和擦除，但是不能进行板外编程。

20.2.2 通信方式

外部器件和 RL78 微控制器的通信使用 TOOL0 引脚，通过专用的 UART 进行串行通信。

传送速率：固定为 115200bps

图 20-4 和外部器件的通信



外部器件对 RL78 微控制器生成以下信号。

表 20-2 引脚连接一览表

外部器件		RL78 微控制器
信号名	输入 / 输出	引脚名
V _{DD}	输入 / 输出	V _{DD}
GND	—	V _{SS}
RESETOUT	输出	RESET
RxD	输入	TOOL0
TxD	输出	

20.3 电路板上的引脚处理

要通过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局引脚就保持刚复位后的相同状态。因此，如果外部器件不认可刚复位后的状态，就需要进行引脚处理。

备注 有关闪存编程模式，请参照“20.4.2 闪存编程模式”。

20.3.1 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的闪存编程器。

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_{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_{HD} ：在转移到闪存编程模式时，从解除外部 / 内部复位后 TOOL0 引脚保持低电平的时间。参照“24.10 闪存编程模式的转移时序”。

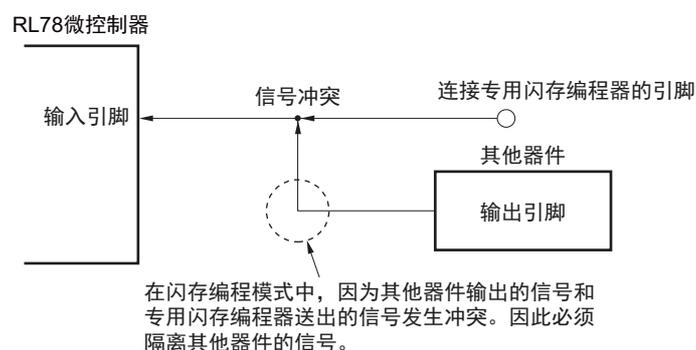
2. 因为 RL78 微控制器和专用闪存编程器的通信使用单线 UART（TOOL0 引脚），所以不使用 SAU 引脚和 IICA 引脚。

20.3.2 $\overline{\text{RESET}}$ 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 20-5 信号冲突（ $\overline{\text{RESET}}$ 引脚）



20.3.3 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。

20.3.4 X1 引脚和 X2 引脚（只限于 16 引脚产品）

X1 引脚和 X2 引脚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})。

20.3.5 电源

当使用闪存编程器进行串行编程时，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。

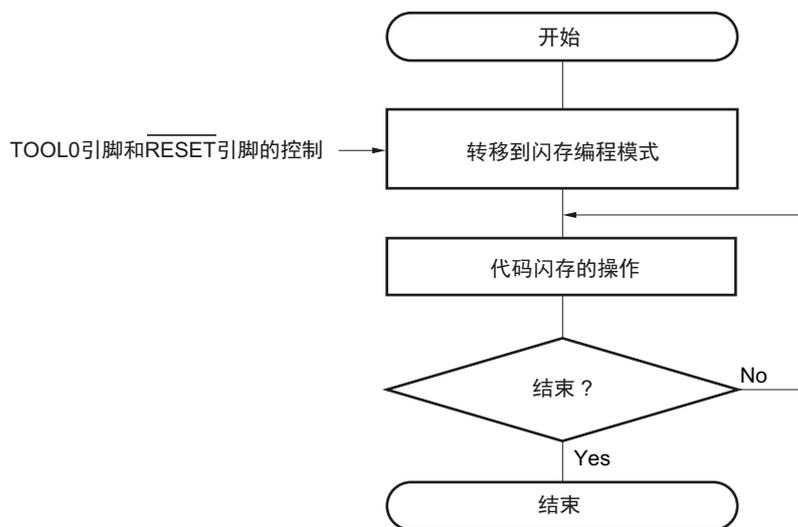
闪存编程时的工作电压为 4.5 ~ 5.5V。当板上的电源小于 4.5V 时，必须通过切换至提供专用闪存编程器的电源等隔离板上的电源后，提供 4.5 ~ 5.5V 的电压。

20.4 串行编程方法

20.4.1 串行编程步骤

通过串行编程执行代码闪存编程的步骤如图 20-6 所示。

图 20-6 代码闪存的操作步骤



有关闪存编程模式请参照“20.4.2 闪存编程模式”。

20.4.2 闪存编程模式

通过串行编程改写代码闪存的内容时，必须转移到闪存编程模式。

<使用专用闪存编程器进行串行编程的情况>

将 RL78 微控制器与专用闪存编程器连接。通过与专用闪存编程器的通信，自动转移到闪存编程模式。另外，闪存编程模式的工作电压为 4.5 ~ 5.5V。

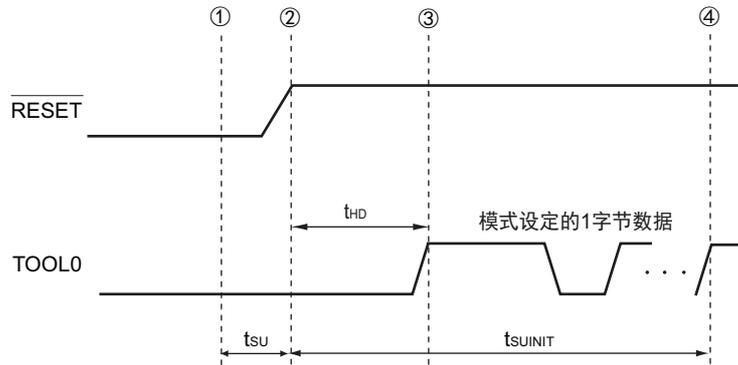
<使用外部器件（UART 通信）进行串行编程的情况>

在将 TOOL0 引脚置为低电平后解除复位（参照表 20-3）。此后，按照图 20-7 所示的步骤①~④转移到闪存编程模式。另外，闪存编程模式的工作电压为 4.5 ~ 5.5V。

表 20-3 复位解除时的 TOOL0 引脚与运行模式的关系

TOOL0	运行模式
V _{DD}	通常运行模式
0V	闪存编程模式

图 20-7 向闪存编程模式的转移



- ① 给 TOOL0 引脚输入低电位。
- ② 解除外部复位（在此之前需要解除 SPOR 复位）。
- ③ 解除 TOOL0 引脚的低电平。
- ④ 通过 UART 接收进行模式转移的设定。

备注 t_{SUNIT} : 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 引脚置低电平到解除外部复位为止的时间。
 t_{HD} : 这是从解除外部复位后 TOOL0 引脚保持低电平的时间。

详细内容请参照“24.10 闪存编程模式的转移时序”。

20.4.3 通信方式

RL78 微控制器的通信方式如下所示。

表 20-4 通信方式

通信方式	标准设定注 1				使用的引脚
	端口	速率注 2	频率	乘率	
单线 UART	UART	115200bps	—	—	TOOL0

注 1. 这是与闪存编程器 GUI 标准设定相关的设定项目。
 2. 除了波特率误差以外，信号波形的失真等也会影响 UART 通信，因此必须在充分评估后使用。

20.4.4 通信命令

RL78 微控制器通过表 20-5 所示的命令执行串行编程。

从专用闪存编程器或者外部器件发送到 RL78 微控制器的信号称为“命令”，并且执行与该命令对应的各功能处理。

表 20-5 闪存控制命令

分类	命令名称	功能
CRC 检查	CRC 检查	计算校验和。
擦除后 / 编程	擦除后编程	在擦除闪存后写数据。

20.5 使用 PG-FP5 时的各命令处理时间（参考值）

使用专用闪存编程器 PG-FP5 时的各命令处理时间（参考值）如表 20-6 所示。

表 20-6 使用 PG-FP5 时的各命令处理时间（参考值）

PG-FP5 的命令	代码闪存		
	1K 字节	2K 字节	4K 字节
	R5F10Y14 R5F10Y44	R5F10Y16 R5F10Y46	R5F10Y17 R5F10Y47
擦除后编程	1.0s	1.0s	1.3s
CRC 检查	0.5s	0.5s	0.5s

备注 命令处理时间（参考值）为 TYP. 值。条件如下所示。

端口： TOOL0（单线 UART）

速率： 115200bp

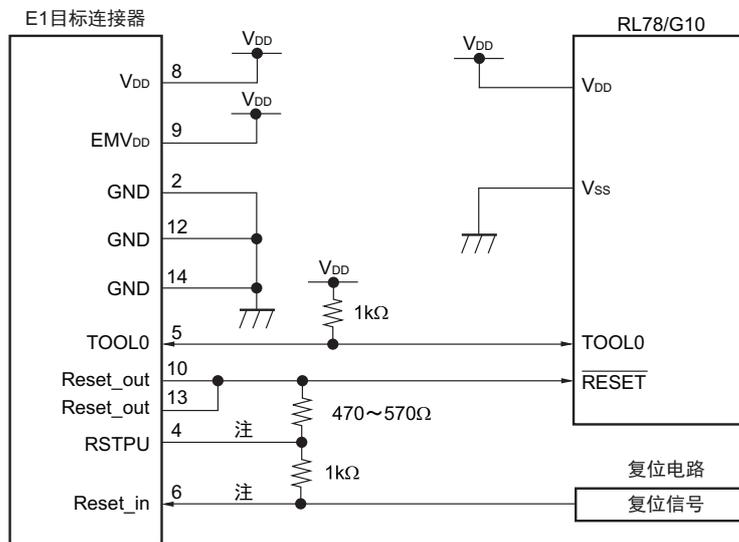
第 21 章 片上调试功能

21.1 与 E1 片上调试仿真器的连接

当 RL78 微控制器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{RESET} 、 $TOOL0$ 、 V_{SS} 引脚。通过使用 $TOOL0$ 引脚的单线 UART 进行串行通信。

注意 RL78 微控制器配置了用于开发和评估的片上调试功能。如果使用片上调试功能，就可能超过闪存保证的改写次数而无法保证产品的可靠性，因此不能对量产的产品使用此功能。对于使用片上调试功能的产品，瑞萨电子不承担任何责任。

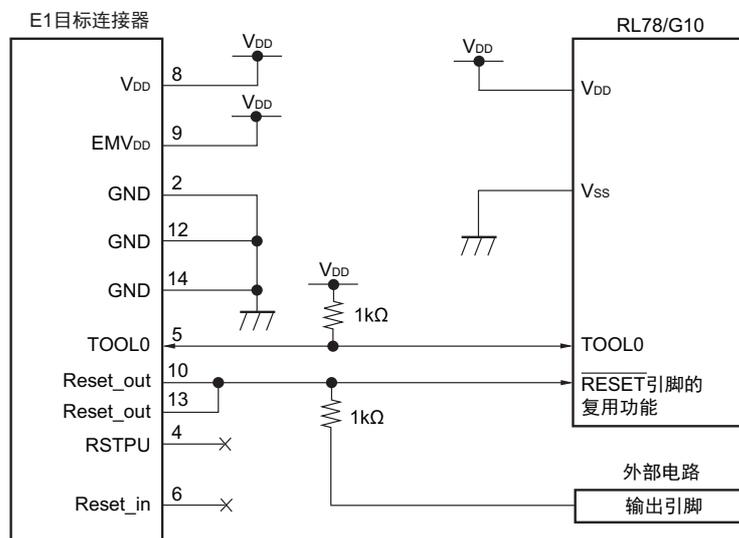
图 21-1 与 E1 片上调试仿真器的连接例子



注 在编程时，不需要连接。

另外，在使用 \overline{RESET} 引脚的复用输入功能的目标系统中，必须分离与外部电路的连接。

图 21-2 与 E1 片上调试仿真器的连接例子（使用 \overline{RESET} 引脚的复用输入功能的情况）



21.2 片上调试安全 ID

为了防止他人读取存储器的内容，RL78 微控制器在闪存的 000C3H 中提供了片上调试运行控制位（参照“第 19 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 21-1 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

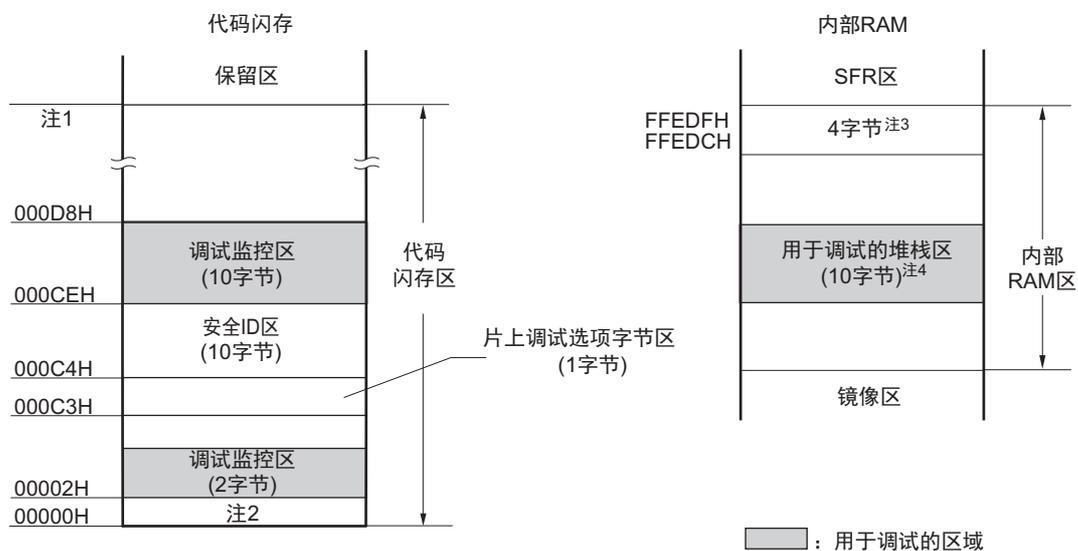
21.3 用户资源的确保

为了实现 RL78 微控制器与 E1 片上调试仿真器的通信和各种调试功能，需提前确保存储空间。当使用本公司的汇编程序和编译程序时，能通过链接程序的选项进行设定。

(1) 存储空间的确保

图 21-3 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 21-3 内置调试监控程序的存储空间



注 1. 如下所示，地址因产品而不同。

产品名	地址
R5F10Y14、R5F10Y44	003FFH
R5F10Y16、R5F10Y46	007FFH
R5F10Y17、R5F10Y47	00FFFH

- 在调试时，复位向量被改写为监控程序的分配地址。
- 在使用实时 RAM 监控（RRM）功能和 Dynamic Memory Modification（DMM）功能时，占用 FFEDCH ~ FFEDFH 的 4 字节。不使用 RRM 功能和 DMM 功能时，能用作内部 RAM。
- 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 10 字节。

第 22 章 十进制校正 (BCD) 电路

22.1 十进制校正电路的功能

能以 BCD 码 (二进制编码的十进制) 计算 BCD 码和 BCD 码相加减的结果。

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。

22.2 十进制校正电路使用的寄存器

十进制校正电路使用以下的寄存器。

- BCD 校正结果寄存器 (BCDADJ)

22.2.1 BCD 校正结果寄存器 (BCDADJ)

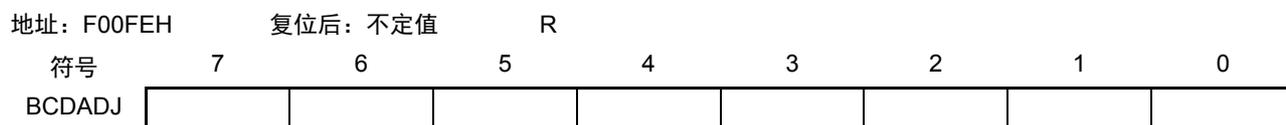
通过以 A 寄存器为操作数的加减运算指令, 将用于计算 BCD 码加减运算结果的校正值保存到 BCDADJ 寄存器。

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 22-1 BCD 校正结果寄存器 (BCDADJ) 的格式



22.3 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（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（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（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 23 章 指令集的概要

RL78 单片机的 RL78-S1 内核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》（R01US0015）。

备注 RL78-S1 内核的指令与 RL78-S2 完全通用。但是，部分指令的时钟数不同。RL78-S1 内核和 RL78-S2 内核的时钟数不同的指令，在“表 23.2”中用灰色表格表示。

23.1 凡例

23.1.1 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）。当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 23-1 中括号内的名称，如 R0、R1、R2 等）进行记述。

表 23-1 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（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

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 3-4 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 3-5 扩展 SFR（2nd SFR）一览表”。

23.1.2 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 23-2 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
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)
^	逻辑与 (AND)
∨	逻辑或者 (OR)
⊕	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)

23.1.3 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值变化。

表 23-3 标志栏的符号

符号	标志的变化
(Blank)	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。

23.1.4 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。通过将 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 ES 寄存器值的地址被执行。

在 PREFIX 指令码和紧随其后的 1 条指令之间，不接受中断或者 DMA 传送。

表 23-4 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。

23.2 操作一览表

表 23-5 操作一览表 (1/17)

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	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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (2/17)

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←sfr			
		sfr, A	2	1	—	sfr←A			
		A, [DE]	1	1	4	A←(DE)			
		[DE], A	1	1	—	(DE)←A			
		A, ES:[DE]	2	2	5	A←(ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE)←A			
		A, [HL]	1	1	4	A←(HL)			
		[HL], A	1	1	—	(HL)←A			
		A, ES:[HL]	2	2	5	A←(ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL)←A			
		A, [DE+byte]	2	1	4	A←(DE+byte)			
		[DE+byte], A	2	1	—	(DE+byte)←A			
		A, ES:[DE+byte]	3	2	5	A←((ES, DE)+byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE)+byte)←A			
		A, [HL+byte]	2	1	4	A←(HL+byte)			
		[HL+byte], A	2	1	—	(HL+byte)←A			
		A, ES:[HL+byte]	3	2	5	A←((ES, HL)+byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL)+byte)←A			
		A, [SP+byte]	2	1	—	A←(SP+byte)			
		[SP+byte], A	2	1	—	(SP+byte)←A			
		A, word[B]	3	1	4	A←(B+word)			
		word[B], A	3	1	—	(B+word)←A			
		A, ES:word[B]	4	2	5	A←((ES, B)+word)			
		ES:word[B], A	4	2	—	((ES, B)+word)←A			
		A, word[C]	3	1	4	A←(C+word)			
		word[C], A	3	1	—	(C+word)←A			
		A, ES:word[C]	4	2	5	A←((ES, C)+word)			
		ES:word[C], A	4	2	—	((ES, C)+word)←A			
		A, word[BC]	3	1	4	A←(BC+word)			
		word[BC], A	3	1	—	(BC+word)←A			
A, ES:word[BC]	4	2	5	A←((ES, BC)+word)					
ES:word[BC], A	4	2	—	((ES, BC)+word)←A					

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (3/17)

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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。
 2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (4/17)

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				
		laddr16	3	1	—	(addr16) \leftarrow 01H				
		ES:laddr16	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				
		laddr16	3	1	—	(addr16) \leftarrow 00H				
		ES:laddr16	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	2	—	rp \leftarrow word			
			saddrp, #word	4	2	—	(saddrp) \leftarrow word			
			sfrp, #word	4	2	—	sfrp \leftarrow word			
			AX, rp 注 3	1	2	—	AX \leftarrow rp			
			rp, AX 注 3	1	2	—	rp \leftarrow AX			
			AX, laddr16	3	2	5	AX \leftarrow (addr16)			
			laddr16, AX	3	2	—	(addr16) \leftarrow AX			
			AX, ES:laddr16	4	3	6	AX \leftarrow (ES, addr16)			
ES:laddr16, AX			4	3	—	(ES, addr16) \leftarrow AX				
AX, saddrp			2	2	—	AX \leftarrow (saddrp)				
saddrp, AX			2	2	—	(saddrp) \leftarrow AX				
AX, sfrp			2	2	—	AX \leftarrow sfrp				
sfrp, AX	2	2	—	sfrp \leftarrow AX						

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (5/17)

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	2	5	AX←(DE)			
		[DE], AX	1	2	—	(DE)←AX			
		AX, ES:[DE]	2	3	6	AX←(ES, DE)			
		ES:[DE], AX	2	3	—	(ES, DE)←AX			
		AX, [HL]	1	2	5	AX←(HL)			
		[HL], AX	1	2	—	(HL)←AX			
		AX, ES:[HL]	2	3	6	AX←(ES, HL)			
		ES:[HL], AX	2	3	—	(ES, HL)←AX			
		AX, [DE+byte]	2	2	5	AX←(DE+byte)			
		[DE+byte], AX	2	2	—	(DE+byte)←AX			
		AX, ES:[DE+byte]	3	3	6	AX←((ES, DE)+byte)			
		ES:[DE+byte], AX	3	3	—	((ES, DE)+byte)←AX			
		AX, [HL+byte]	2	2	5	AX←(HL+byte)			
		[HL+byte], AX	2	2	—	(HL+byte)←AX			
		AX, ES:[HL+byte]	3	3	6	AX←((ES, HL)+byte)			
		ES:[HL+byte], AX	3	3	—	((ES, HL)+byte)←AX			
		AX, [SP+byte]	2	2	—	AX←(SP+byte)			
		[SP+byte], AX	2	2	—	(SP+byte)←AX			
		AX, word[B]	3	2	5	AX←(B+word)			
		word[B], AX	3	2	—	(B+word)←AX			
		AX, ES:word[B]	4	3	6	AX←((ES, B)+word)			
		ES:word[B], AX	4	3	—	((ES, B)+word)←AX			
		AX, word[C]	3	2	5	AX←(C+word)			
		word[C], AX	3	2	—	(C+word)←AX			
		AX, ES:word[C]	4	3	6	AX←((ES, C)+word)			
		ES:word[C], AX	4	3	—	((ES, C)+word)←AX			
		AX, word[BC]	3	2	5	AX←(BC+word)			
		word[BC], AX	3	2	—	(BC+word)←AX			
AX, ES:word[BC]	4	3	6	AX←((ES, BC)+word)					
ES:word[BC], AX	4	3	—	((ES, BC)+word)←AX					

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (6/17)

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	2	5	BC←(addr16)			
		BC, ES:!addr16	4	3	6	BC←(ES, addr16)			
		DE, !addr16	3	2	5	DE←(addr16)			
		DE, ES:!addr16	4	3	6	DE←(ES, addr16)			
		HL, !addr16	3	2	5	HL←(addr16)			
		HL, ES:!addr16	4	3	6	HL←(ES, addr16)			
		BC, saddrp	2	2	—	BC←(saddrp)			
		DE, saddrp	2	2	—	DE←(saddrp)			
		HL, saddrp	2	2	—	HL←(saddrp)			
	XCHW	AX, rp 注 3	1	2	—	AX↔rp			
	ONEW	AX	1	2	—	AX←0001H			
		BC	1	2	—	BC←0001H			
	CLRW	AX	1	2	—	AX←0000H			
		BC	1	2	—	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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (7/17)

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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (8/17)

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←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	×	×	×
	AND	A, #byte	2	1	—	A←A∧byte	×		
		saddr, #byte	3	2	—	(saddr)←(saddr)∧byte	×		
		A, r 注 3	2	1	—	A←A∧r	×		
		r, A	2	1	—	r←r∧A	×		
		A, !addr16	3	1	4	A←A∧(addr16)	×		
		A, ES:!addr16	4	2	5	A←A∧(ES:addr16)	×		
		A, saddr	2	1	—	A←A∧(saddr)	×		
		A, [HL]	1	1	4	A←A∧(HL)	×		
		A, ES:[HL]	2	2	5	A←A∧(ES:HL)	×		
		A, [HL+byte]	2	1	4	A←A∧(HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A←A∧((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A←A∧(HL+B)	×		
A, ES:[HL+B]	3	2	5	A←A∧((ES:HL)+B)	×				
A, [HL+C]	2	1	4	A←A∧(HL+C)	×				
A, ES:[HL+C]	3	2	5	A←A∧((ES:HL)+C)	×				

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (9/17)

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	—	$A \leftarrow A \vee \text{byte}$		X	
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		X	
		A, r 注 3	2	1	—	$A \leftarrow A \vee r$		X	
		r, A	2	1	—	$r \leftarrow r \vee A$		X	
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr}16)$		X	
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr}16)$		X	
		A, saddr	2	1	—	$A \leftarrow A \vee (\text{saddr})$		X	
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$		X	
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$		X	
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL}+\text{byte})$		X	
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL})+\text{byte})$		X	
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL}+B)$		X	
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL})+B)$		X	
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL}+C)$		X	
	A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL})+C)$		X		
	XOR	A, #byte	2	1	—	$A \leftarrow A \oplus \text{byte}$		X	
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$		X	
		A, r 注 3	2	1	—	$A \leftarrow A \oplus r$		X	
		r, A	2	1	—	$r \leftarrow r \oplus A$		X	
		A, !addr16	3	1	4	$A \leftarrow A \oplus (\text{addr}16)$		X	
		A, ES:!addr16	4	2	5	$A \leftarrow A \oplus (\text{ES:addr}16)$		X	
		A, saddr	2	1	—	$A \leftarrow A \oplus (\text{saddr})$		X	
		A, [HL]	1	1	4	$A \leftarrow A \oplus (\text{HL})$		X	
		A, ES:[HL]	2	2	5	$A \leftarrow A \oplus (\text{ES:HL})$		X	
		A, [HL+byte]	2	1	4	$A \leftarrow A \oplus (\text{HL}+\text{byte})$		X	
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL})+\text{byte})$		X	
A, [HL+B]		2	1	4	$A \leftarrow A \oplus (\text{HL}+B)$		X		
A, ES:[HL+B]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL})+B)$		X			
A, [HL+C]	2	1	4	$A \leftarrow A \oplus (\text{HL}+C)$		X			
A, ES:[HL+C]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL})+C)$		X			

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (10/17)

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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (11/17)

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	2	—	AX, CY←AX+word	×	×	×
		AX, AX	1	2	—	AX, CY←AX+AX	×	×	×
		AX, BC	1	2	—	AX, CY←AX+BC	×	×	×
		AX, DE	1	2	—	AX, CY←AX+DE	×	×	×
		AX, HL	1	2	—	AX, CY←AX+HL	×	×	×
		AX, !addr16	3	2	5	AX, CY←AX+(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY←AX+(ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX, CY←AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY←AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY←AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	2	—	AX, CY←AX-word	×	×	×
		AX, BC	1	2	—	AX, CY←AX-BC	×	×	×
		AX, DE	1	2	—	AX, CY←AX-DE	×	×	×
		AX, HL	1	2	—	AX, CY←AX-HL	×	×	×
		AX, !addr16	3	2	5	AX, CY←AX-(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY←AX-(ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX, CY←AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY←AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY←AX-((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	2	—	AX-word	×	×	×
		AX, BC	1	2	—	AX-BC	×	×	×
		AX, DE	1	2	—	AX-DE	×	×	×
		AX, HL	1	2	—	AX-HL	×	×	×
		AX, !addr16	3	2	5	AX-(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX-(ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX-(HL+byte)	×	×	×
AX, ES: [HL+byte]		4	3	6	AX-((ES:HL)+byte)	×	×	×	
乘	MULU	X	1	2	—	AX←A×X			

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (12/17)

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	—	$r \leftarrow r+1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16)+1$	×	×	
		ES:laddr16	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	—	$r \leftarrow r-1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16)-1$	×	×	
		ES:laddr16	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	2	—	$rp \leftarrow rp+1$			
		laddr16	3	4	—	$(addr16) \leftarrow (addr16)+1$			
		ES:laddr16	4	5	—	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	4	—	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	4	—	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	5	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	2	—	$rp \leftarrow rp-1$			
		laddr16	3	4	—	$(addr16) \leftarrow (addr16)-1$			
		ES:laddr16	4	5	—	$(ES, addr16) \leftarrow (ES, addr16)-1$			
		saddrp	2	4	—	$(saddrp) \leftarrow (saddrp)-1$			
		[HL+byte]	3	4	—	$(HL+byte) \leftarrow (HL+byte)-1$			
		ES: [HL+byte]	4	5	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$			
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	2	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	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	2	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	2	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	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	2	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×	

注 1. 这是存取内部 RAM 区、SFR 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

2. cnt 是位的移位个数。

表 23-5 操作一览表 (13/17)

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	2	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
BC, 1		2	2	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×	
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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (14/17)

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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (15/17)

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	4	—	(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	4	—	(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	4	—	(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	4	—	(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	6	—	(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	7	—	(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	7	—	PC _L \leftarrow (SP), PC _H \leftarrow (SP+1), PC _S \leftarrow (SP+2), SP \leftarrow SP+4			
	RETI	—	2	8	—	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	8	—	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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (16/17)

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	2	—	(SP-1)←PSW, (SP-2)←00H, SP←SP-2			
		rp	1	2	—	(SP-1)←rp _H , (SP-2)←rp _L , SP←SP-2			
	POP	PSW	2	4	—	PSW←(SP+1), SP←SP+2	R	R	R
		rp	1	2	—	rp _L ←(SP), rp _H ←(SP+1), SP←SP+2			
	MOVW	SP, #word	4	2	—	SP←word			
		SP, AX	2	2	—	SP←AX			
		AX, SP	2	2	—	AX←SP			
		HL, SP	3	2	—	HL←SP			
		BC, SP	3	2	—	BC←SP			
		DE, SP	3	2	—	DE←SP			
ADDW	SP, #byte	2	2	—	SP←SP+byte				
SUBW	SP, #byte	2	2	—	SP←SP-byte				
无条件转移	BR	AX	2	3	—	PC←CS, AX			
		\$addr20	2	3	—	PC←PC+2+jdisp8			
		\$!addr20	3	3	—	PC←PC+3+jdisp16			
		!addr16	3	3	—	PC←0000, addr16			
		!!addr20	4	3	—	PC←addr20			
条件转移	BC	\$addr20	2	2/4注3	—	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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。
 2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

表 23-5 操作一览表 (17/17)

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	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 区或者扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。
 2. 这是存取代码闪存区时的 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 4 倍 +6 个时钟。

第 24 章 电特性

- 注意 1. RL78 微控制器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 端口功能”～“2.2.1 各产品内置的功能”。
 3. 因为 RL78/G10 包含可选择的上电复位（SPOR）的检测电压（ V_{SPOR} ），所以必须在 2.25 ~ 5.5V 的电压范围下使用。

24.1 绝对最大额定值

(T_A=25°C)

项目	符号	条件	额定值	单位	
电源电压	V _{DD}		-0.5 ~ +6.5	V	
输入电压	V _{I1}		-0.3 ~ V _{DD} +0.3 注	V	
输出电压	V _{O1}		-0.3 ~ V _{DD} +0.3	V	
高电平输出电流	I _{OH1}	1 个引脚	-40	mA	
		引脚合计	P40、P41	-70	mA
			P00 ~ P07	-100	mA
低电平输出电流	I _{OL1}	1 个引脚	40	mA	
		引脚合计	P40、P41	70	mA
			P00 ~ P07	100	mA
工作环境温度	T _A		-40 ~ +85	°C	
保存温度	T _{stg}		-65 ~ +150	°C	

注 不超过 6.5V。

注意 即使是各项目中的 1 个项目一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. 将 V_{SS} 作为基准电压。

24.2 振荡电路特性

24.2.1 X1 振荡电路特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	谐振器	条件	MIN.	TYP.	MAX.	单位
X1 时钟振荡 频率 (f_X) 注	陶瓷谐振器 / 晶体振荡器	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	1		20	MHz
		$2.0\text{V} \leq V_{DD} < 2.7\text{V}$	1		5	

注 只表示振荡电路的频率容许范围，指令执行时间请参照 AC 特性。

另外，委托谐振器厂商进行在安装电路中的评估，并且在确认振荡特性后再使用。

注意 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。

备注 在使用 X1 振荡电路时，请参照“5.4 系统时钟振荡电路”。

24.2.2 内部振荡器特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高速内部振荡器时钟频率注 1、注 2	f_{IH}		1.25		20	MHz
高速内部振荡器时钟频率精度		$T_A = -20 \sim +85^\circ\text{C}$	-2.0		+2.0	%
		$T_A = -40 \sim -20^\circ\text{C}$	-3.0		+3.0	%
低速内部振荡器时钟频率	f_{IL}			15		kHz
低速内部振荡器时钟频率精度			-15		+15	%

注 1. 通过选项字节（000C2H）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 AC 特性。

24.3 DC 特性

24.3.1 引脚特性

(T_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

(1/2)

项目	符号	条件	MIN.	TYP.	MAX.	单位
高电平输出电流 ^{注1}	I _{OH1}	10 引脚产品: P00 ~ P04、P40 16 引脚产品: P00 ~ P07、P40、P41 1 个引脚			-10.0 ^{注2}	mA
		10 引脚产品: P40 16 引脚产品: P40、P41 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ V _{DD} ≤ 5.5V		-20.0	mA
			2.7V ≤ V _{DD} < 4.0V		-4.0	mA
			2.0V ≤ V _{DD} < 2.7V		-3.0	mA
		10 引脚产品: P00 ~ P04 16 引脚产品: P00 ~ P07 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ V _{DD} ≤ 5.5V		-60.0	mA
			2.7V ≤ V _{DD} < 4.0V		-12.0	mA
			2.0V ≤ V _{DD} < 2.7V		-9.0	mA
		引脚合计 (占空比 ≤ 70% 时注3)				-80.0
低电平输出电流 ^{注4}	I _{OL1}	10 引脚产品: P00 ~ P04、P40 16 引脚产品: P00 ~ P07、P40、P41 1 个引脚			20.0 ^{注2}	mA
		10 引脚产品: P40 16 引脚产品: P40、P41 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ V _{DD} ≤ 5.5V		40.0	mA
			2.7V ≤ V _{DD} < 4.0V		6.0	mA
			2.0V ≤ V _{DD} < 2.7V		1.2	mA
		10 引脚产品: P00 ~ P04 16 引脚产品: P00 ~ P07 合计 (占空比 ≤ 70% 时注3)	4.0V ≤ V _{DD} ≤ 5.5V		80.0	mA
			2.7V ≤ V _{DD} < 4.0V		12.0	mA
			2.0V ≤ V _{DD} < 2.7V		2.4	mA
		引脚合计 (占空比 ≤ 70% 时注3)				120.0

注 1. 这是即使电流从 V_{DD} 引脚流到输出引脚也保证器件运行的电流值。

2. 不能超过合计的电流值。

3. 这是在占空比 ≤ 70% 条件下的输出电流值。

更改为占空比 > 70% 时的输出电流值能用以下的计算式进行计算 (将占空比更改为 n% 的情况)。

• 引脚合计的输出电流 = (I_{OH} × 0.7) / (n × 0.01)

<计算例子> I_{OH} = -10.0mA、n = 80%

引脚合计的输出电流 = (-10.0 × 0.7) / (80 × 0.01) ≈ -8.7mA

• 引脚合计的输出电流 = (I_{OL} × 0.7) / (n × 0.01)

<计算例子> I_{OL} = 10.0mA、n = 80%

引脚合计的输出电流 = (10.0 × 0.7) / (80 × 0.01) ≈ 8.7mA

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。

4. 这是电流即使从输出引脚流到 V_{SS} 引脚也保证器件运行的电流值。

注意 在 N 沟道漏极开路模式中, P00、P01、P06 和 P07 不输出高电平。

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_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

(2/2)

项目	符号	条件		MIN.	TYP.	MAX.	单位
高电平输入电压	V _{IH1}			0.8V _{DD}		V _{DD}	V
低电平输入电压	V _{IL1}			0		0.2V _{DD}	V
高电平输出电压 注 1	V _{OH1}	4.0V ≤ V _{DD} ≤ 5.5V	I _{OH1} =-10mA	V _{DD} -1.5			V
			I _{OH1} =-3.0mA	V _{DD} -0.7			V
		2.7V ≤ V _{DD} ≤ 5.5V	I _{OH1} =-2.0mA	V _{DD} -0.6			V
		2.0V ≤ V _{DD} ≤ 5.5V	I _{OH1} =-1.5mA	V _{DD} -0.5			V
低电平输出电压 注 2	V _{OL1}	4.0V ≤ V _{DD} ≤ 5.5V	I _{OL1} =20mA			1.3	V
			I _{OL1} =8.5mA			0.7	V
		2.7V ≤ V _{DD} ≤ 5.5V	I _{OL1} =3.0mA			0.6	V
		2.0V ≤ V _{DD} ≤ 5.5V	I _{OL1} =1.5mA			0.4	V
高电平输入 漏电流	I _{LIH1}	P00 ~ P07、P40、P41、P125、P137 V _I =V _{DD}				1	μA
			I _{LIH2}	P121、P122 (X1、X2、EXCLK) V _I =V _{DD}	输入端口或者外部时 钟输入时		
				连接谐振器时			
低电平输入 漏电流	I _{LIL1}	P00 ~ P07、P40、P41、P125、P137 V _I =V _{SS}				-1	μA
			I _{LIL2}	P121、P122 (X1、X2、EXCLK) V _I =V _{SS}	输入端口或者外部时 钟输入时		
				连接谐振器时			
内部上拉电阻	R _U	V _I =V _{SS}				100	kΩ

注 1. 这是在满足高电平输出电流 (I_{OH1}) 条件下的值。

2. 这是在满足低电平输出电流 (I_{OL1}) 条件下的值。

注意 即使在 N 沟道漏极开路模式中, P00、P01、P06 和 P07 的 V_{IH} 最大值 (MAX.) 也为 V_{DD}。
并且在 N 沟道漏极开路模式中, 不输出高电平。

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。

24.3.2 电源电流特性

(1) 10 引脚产品中闪存 ROM 为 1KB、2KB 的产品

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件		MIN.	TYP.	MAX.	单位		
电源电流 注1	I_{DD1}	运行模式	基本运行	$f_{IH} = 20\text{MHz}$	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		0.91	mA	
			通常运行	$f_{IH} = 20\text{MHz}$	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		1.57		2.04
				$f_{IH} = 5\text{MHz}$	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		0.85		1.15
	I_{DD2} 注2	HALT 模式		$f_{IH} = 20\text{MHz}$	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		350	820	μA
				$f_{IH} = 5\text{MHz}$	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		290	600	
	I_{DD3} 注3	STOP 模式			$V_{DD} = 3.0\text{V}$		0.56	2.00	μA

注 1. 这是流过 V_{DD} 的总电流，包含输入引脚固定为 V_{DD} 或者 V_{SS} 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、I/O 端口、内部上拉或者下拉电阻的电流。

2. 这是在闪存中执行 HALT 指令时的情况。
3. 不包含流到看门狗定时器的电流。

备注 1. f_{IH} : 高速内部振荡器的时钟频率

2. TYP. 值的温度条件是 $T_A = 25^\circ\text{C}$ 。

(2) 10 引脚产品中闪存 ROM 为 4KB 的产品和 16 引脚产品

 $(T_A = -40 \sim +85^\circ\text{C}, 2.0\text{V} \leq V_{DD} \leq 5.5\text{V}, V_{SS} = 0\text{V})$

项目	符号	条件		MIN.	TYP.	MAX.	单位		
电源电流 注 1	I_{DD1}	运行模式	基本运行	$f_{IH} = 20\text{MHz}$ 注 4	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		0.92		mA
			通常运行	$f_{IH} = 20\text{MHz}$ 注 4	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		1.59	2.14	
				$f_{IH} = 5\text{MHz}$ 注 4	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		0.87	1.20	
				$f_{MX} = 20\text{MHz}$ 注 5、6 $V_{DD} = 3.0\text{V}, 5.0\text{V}$	方波输入		1.43	1.93	
					谐振器连接		1.54	2.13	
			$f_{MX} = 5\text{MHz}$ 注 5、6 $V_{DD} = 3.0\text{V}, 5.0\text{V}$	方波输入		0.67	1.02		
				谐振器连接		0.72	1.12		
			I_{DD2} 注 2	HALT 模式	$f_{IH} = 20\text{MHz}$ 注 4	$V_{DD} = 3.0\text{V}, 5.0\text{V}$		360	
	$f_{IH} = 5\text{MHz}$ 注 4	$V_{DD} = 3.0\text{V}, 5.0\text{V}$				310	660		
	$f_{MX} = 20\text{MHz}$ 注 5、6 $V_{DD} = 3.0\text{V}, 5.0\text{V}$	方波输入				200	700		
		谐振器连接				300	900		
	$f_{MX} = 5\text{MHz}$ 注 5、6 $V_{DD} = 3.0\text{V}, 5.0\text{V}$	方波输入				100	440		
	I_{DD3} 注 3	STOP 模式	$V_{DD} = 3.0\text{V}$		0.61	2.25	μA		

注 1. 这是流过 V_{DD} 的总电流，包含输入引脚固定为 V_{DD} 或者 V_{SS} 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、比较器（只限于 16 引脚产品）、I/O 端口、内部上拉或者下拉电阻的电流。

2. 这是在闪存中执行 HALT 指令时的情况。
3. 不包含流到 12 位间隔定时器和看门狗定时器的电流。
4. 高速系统时钟停止振荡的情况。
5. 这是高速内部振荡器停止振荡的情况。
6. 只限于 16 引脚产品。

备注 1. f_{IH} : 高速内部振荡器的时钟频率

2. f_{MX} : 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）

3. TYP. 值的温度条件是 $T_A = 25^\circ\text{C}$ 。

(3) 外围功能 (适用全部产品)

(T_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件		MIN.	TYP.	MAX.	单位
低速内部振荡器工作电流	I _{FIL} 注 1				0.30		μA
12 位间隔定时器工作电流	I _{TMKA} 注 1、2、3				0.01		μA
看门狗定时器工作电流	I _{WDT} 注 1、4				0.01		μA
A/D 转换器工作电流	I _{ADC} 注 1、5	最高速转换时	V _{DD} =5.0V		1.30	1.90	mA
			V _{DD} =3.0V		0.50		
比较器工作电流	I _{CMP} 注 1、6	高速模式中	V _{DD} =5.0V		6.50		μA
		低速模式中	V _{DD} =5.0V		1.70		
内部基准电压工作电流	I _{VREG} 注 1				10		μA

- 注 1. 这是流过 V_{DD} 的电流。
2. 这是高速内部振荡器和高速系统时钟停止振荡的情况。
3. 这是只流到 12 位间隔定时器的电流 (不包含低速内部振荡器的工作电流)。12 位间隔定时器运行的情况下, RL78 微控制器的电流值为 I_{DD1}、I_{DD2} 或者 I_{DD3} 加上 I_{FIL} 和 I_{TMKA} 的值。
4. 这是只流到看门狗定时器的电流 (不包含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, RL78 微控制器的电流值为 I_{DD1}、I_{DD2} 或者 I_{DD3} 加上 I_{FIL} 和 I_{WDT} 的值。
5. 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, RL78 微控制器的电流值为 I_{DD1} 或者 I_{DD2} 加上 I_{ADC} 的值。
6. 这是只流到比较器的电流。在比较器运行的情况下, RL78 微控制器的电流值为 I_{DD1}、I_{DD2} 或者 I_{DD3} 加上 I_{CMP} 的值。

- 备注 1. f_{IL}: 低速内部振荡器时钟频率
2. TYP. 值的温度条件是 T_A=25°C。

24.4 AC 特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

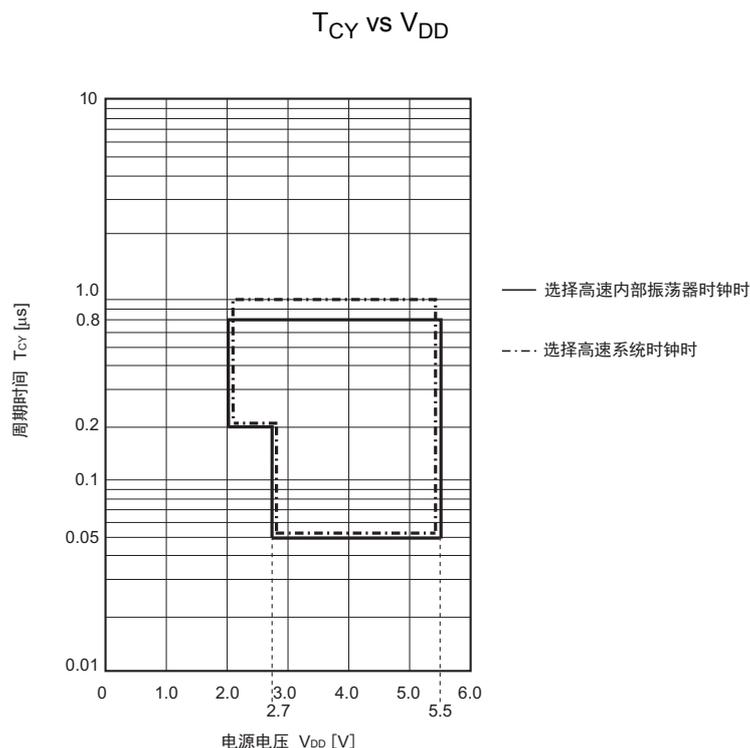
项目	符号	条件	MIN.	TYP.	MAX.	单位	
指令周期 (最小指令执行时间)	T_{CY}	选择高速内部振荡器时钟 (f_{IH}) 时	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	0.05		0.8	μs
			$2.0\text{V} \leq V_{DD} < 2.7\text{V}$	0.2		0.8	
		选择高速系统时钟 (f_{MX}) 时	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	0.05		1.0	μs
			$2.0\text{V} \leq V_{DD} < 2.7\text{V}$	0.2		1.0	
外部系统时钟频率	f_{EX}	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	1.0		20	MHz	
		$2.0\text{V} \leq V_{DD} < 2.7\text{V}$	1.0		5		
外部系统时钟输入的 高 / 低电平宽度	t_{EXH} 、 t_{EXL}	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	24			ns	
		$2.0\text{V} \leq V_{DD} < 2.7\text{V}$	95				
TI00 ~ TI03 输入的高 / 低 电平宽度	t_{TIH} 、 t_{TIL}	未使用噪声滤波器时	$1/f_{MCK}$ +10			ns	
TO00 ~ TO03 的输出 频率	f_{TO}	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$			10	MHz	
		$2.7\text{V} \leq V_{DD} < 4.0\text{V}$			5	MHz	
		$2.0\text{V} \leq V_{DD} < 2.7\text{V}$			2.5	MHz	
PCLBUZ0 的输出频率	f_{PCL}	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$			10	MHz	
		$2.7\text{V} \leq V_{DD} < 4.0\text{V}$			5	MHz	
		$2.0\text{V} \leq V_{DD} < 2.7\text{V}$			2.5	MHz	
RESET 的低电平宽度	t_{RSL}		10			μs	

备注 f_{MCK} : 定时器阵列单元的运行时钟频率

(这是定时器时钟选择寄存器 0 (TPS0) 和定时器模式寄存器 0nH (TMR0nH) 的 CKS0n1 位设定的运行时钟。

n: 通道号 (n=0~3))

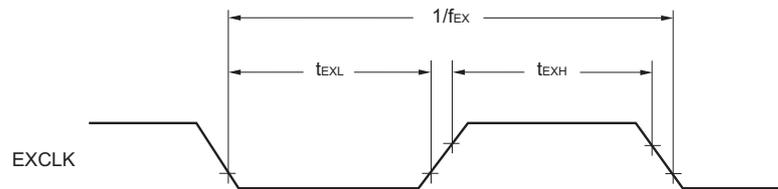
主系统时钟运行时的最短指令执行时间



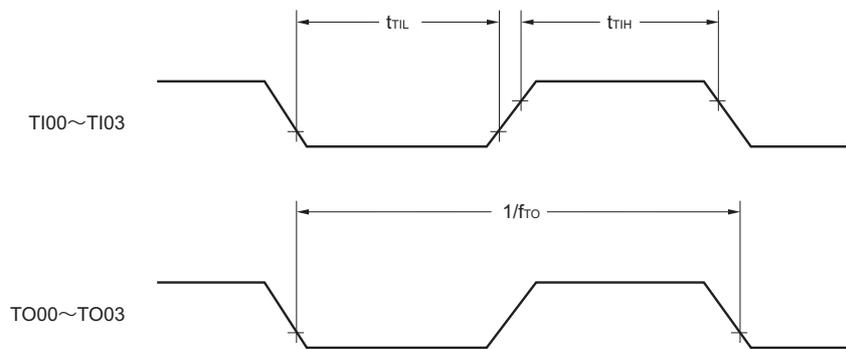
AC 时序测量点



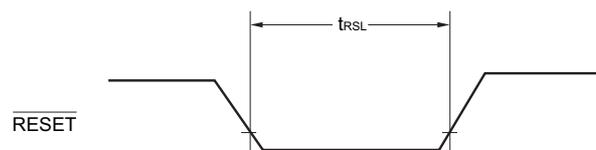
外部系统时钟的时序



TI/TO 时序



$\overline{\text{RESET}}$ 的输入时序



24.5 外围功能特性

AC 时序测量点



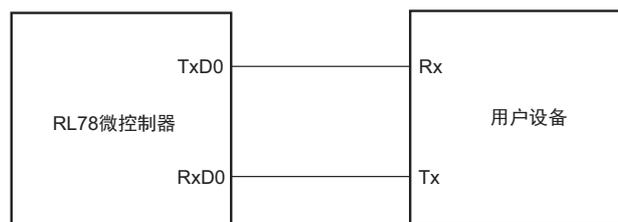
24.5.1 串行阵列单元

(1) UART 模式

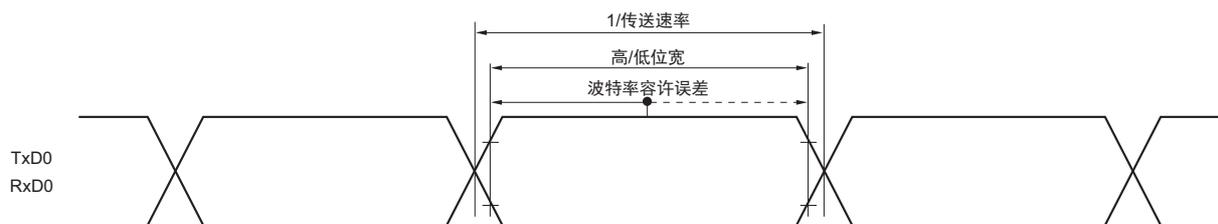
($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					$f_{MCK}/6$	bps
		最大传送速率的理论值 $f_{CLK} = f_{MCK} = 20\text{MHz}$			3.3	Mbps

UART 模式的连接图



UART 模式的位宽 (参考)



备注 f_{MCK} : 串行阵列单元的运行时钟频率

(这是串行时钟选择寄存 0 (SPS0) 和串行模式寄存器 0nH (SMR0nH) 的 CKS0n 位设定的运行时钟。n: 通道号 (n=0、1))

(2) CSI 模式（主控模式，SCKp..... 内部时钟输出）

(T_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
SCKp 周期时间	t _{KCY1}	t _{KCY1} ≥ 4/f _{CLK}	2.7V ≤ V _{DD} ≤ 5.5V	200		ns
			2.0V ≤ V _{DD} ≤ 5.5V	800		ns
SCKp 高 / 低电平宽度	t _{KH1} 、 t _{KL1}	2.7V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2-18			ns
		2.0V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2-50			ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK1}	2.7V ≤ V _{DD} ≤ 5.5V	47			ns
		2.0V ≤ V _{DD} ≤ 5.5V	110			ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KSI1}		19			ns
SCKp↓→SOp 输出延迟时间注 2	t _{KSO1}	C=30pF 注 3			25	ns

- 注 1. 这是 DAP0n=0、CKP0n=0 或者 DAP0n=1、CKP0n=1 的情况。在 DAP0n=0、CKP0n=1 或者 DAP0n=1、CKP0n=0 时，为“对 SCK0p↓”。
2. 这是 DAP0n=0、CKP0n=0 或者 DAP0n=1、CKP0n=1 的情况。在 DAP0n=0、CKP0n=1 或者 DAP0n=1、CKP0n=0 时，为“对 SCK0p↑”。
3. C 是 SCKp、SOp 输出线的负载电容。

(3) CSI 模式（从属模式，SCKp..... 外部时钟输出）

(T_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
SCKp 周期时间	t _{KCY2}	2.7V ≤ V _{DD} ≤ 5.5V	f _{MCK} > 16MHz	8/f _{MCK}		ns
			f _{MCK} ≤ 16MHz	6/f _{MCK}		ns
		2.0V ≤ V _{DD} ≤ 5.5V		6/f _{MCK}		ns
SCKp 高 / 低电平宽度	t _{KH2} 、 t _{KL2}	2.0V ≤ V _{DD} ≤ 5.5V	t _{KCY2} /2-18			ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK2}	2.7V ≤ V _{DD} ≤ 5.5V	1/f _{MCK} +20			ns
		2.0V ≤ V _{DD} ≤ 5.5V	1/f _{MCK} +30			ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KSI2}	2.0V ≤ V _{DD} ≤ 5.5V	1/f _{MCK} +31			ns
SCKp↓→SOp 输出延迟时间注 2	t _{KSO2}	C=30pF 注 3	2.7V ≤ V _{DD} ≤ 5.5V		2/f _{MCK} +50	ns
			2.0V ≤ V _{DD} ≤ 5.5V		2/f _{MCK} +110	ns

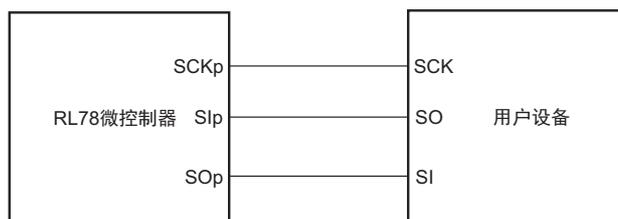
- 注 1. 这是 DAP0n=0、CKP0n=0 或者 DAP0n=1、CKP0n=1 的情况。在 DAP0n=0、CKP0n=1 或者 DAP0n=1、CKP0n=0 时，为“对 SCK0p↓”。
2. 这是 DAP0n=0、CKP0n=0 或者 DAP0n=1、CKP0n=1 的情况。在 DAP0n=0、CKP0n=1 或者 DAP0n=1、CKP0n=0 时，为“对 SCK0p↑”。
3. C 是 SOp 输出线的负载电容。

备注 1. P: CSI 号 (p=00、01)，n= 通道号 (n=0、1)

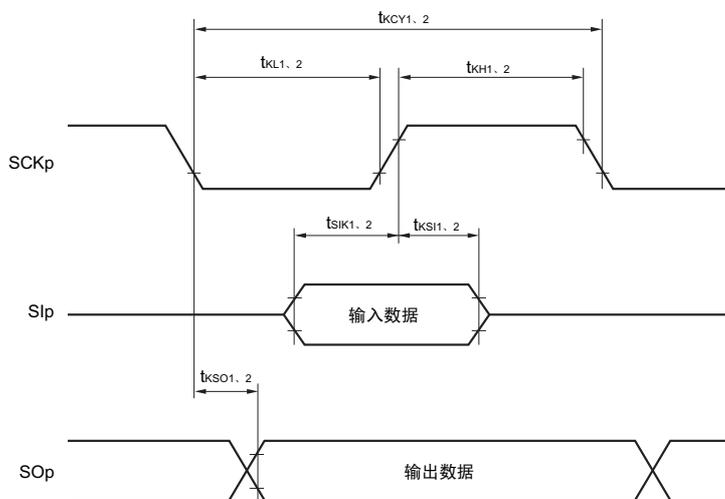
2. f_{MCK}: 串行阵列单元的运行时钟频率

(这是串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0nH (SMR0nH) 的 CKS0n 位设定的运行时钟。)

CSI 模式的连接图



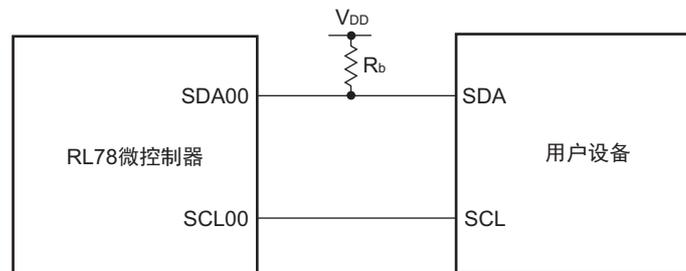
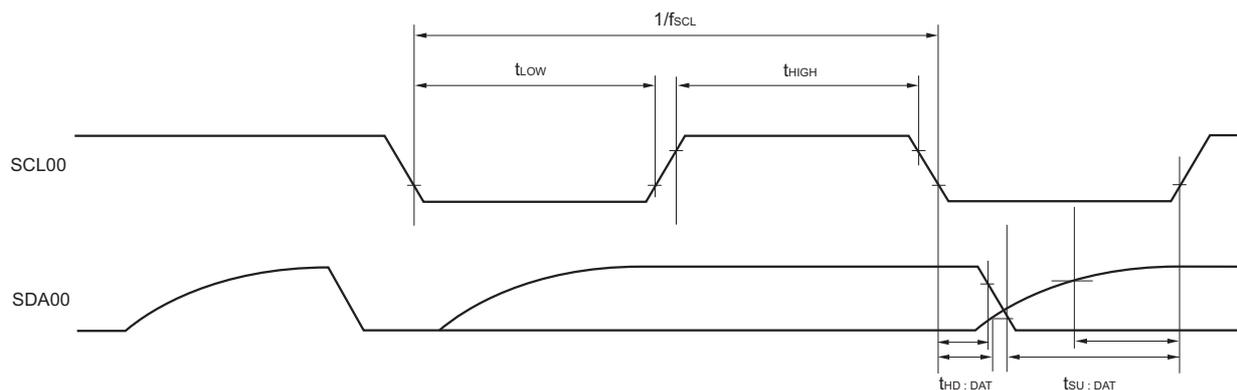
CSI 模式的串行传送时序
(DAP0n=0、CKP0n=0 或者 DAP0n=1、CKP0n=1 的情况)



备注 p: CSI 号 (p=00、01), n: 通道号 (n=0、1)

(4) 简易 I²C 模式(T_A=-40 ~ +85°C、2.0V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	HS (高速主) 模式 LS (低速主) 模式		单位
			MIN.	MAX.	
SCLr 时钟频率	f _{SCL}	C _b =100pF、R _b =3kΩ		400 注 1	kHz
SCLr="L" 的保持时间	t _{LOW}	C _b =100pF、R _b =3kΩ	1150		ns
SCLr="H" 的保持时间	t _{HIGH}	C _b =100pF、R _b =3kΩ	1150		ns
数据准备时间 (接收时)	t _{SU: DAT}	C _b =100pF、R _b =3kΩ	1/f _{MCK} +145 注 2		ns
数据保持时间 (发送时)	t _{HD: DAT}	C _b =100pF、R _b =3kΩ	0	355	ns

注 1. 并且必须设定为“≤ f_{MCK}/4”。2. f_{MCK} 的设定值不能超过 SCLr="L" 和 SCLr="H" 的保持时间的值。注意 通过端口输出模式寄存器 0 (POM0)，将 SDAr 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。简易 I²C 模式的连接图 (同电位的通信)简易 I²C 模式的串行传送时序备注 1. R_b[Ω]: 通信线 (SDAr) 的上拉电阻值 C_b[F]: 通信线 (SCLr、SDAr) 的负载电容值

2. r: IIC 号 (r=00)

3. f_{MCK}: 串行阵列单元的运行时钟频率

(这是串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0nH (SMR0nH) 的 CKS0n 位设定的运行时钟。

n: 通道号 (n=0))

24.5.2 串行接口 IICA

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	标准模式		快速模式		单位
			MIN.	MAX.	MIN.	MAX.	
SCLA0 时钟频率	f_{SCL}	快速模式: $f_{\text{CLK}} \geq 3.5\text{MHz}$			0	400	kHz
		标准模式: $f_{\text{CLK}} \geq 1\text{MHz}$	0	100			
重新开始条件的准备时间	$t_{\text{SU: STA}}$		4.7		0.6		ns
保持时间注 1	$t_{\text{HD: STA}}$		4.0		0.6		ns
SCLA0="L" 的保持时间	t_{LOW}		4.7		1.3		ns
SCLA0="H" 的保持时间	t_{HIGH}		4.0		0.6		ns
数据准备时间 (接收时)	$t_{\text{SU: DAT}}$		250		100		ns
数据保持时间 (发送时) 注 2	$t_{\text{HD: DAT}}$		0	3.45	0	0.9	μs
停止条件的准备时间	$t_{\text{SU: STO}}$		4.0		0.6		μs
总线空闲时间	t_{BUF}		4.7		1.3		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。

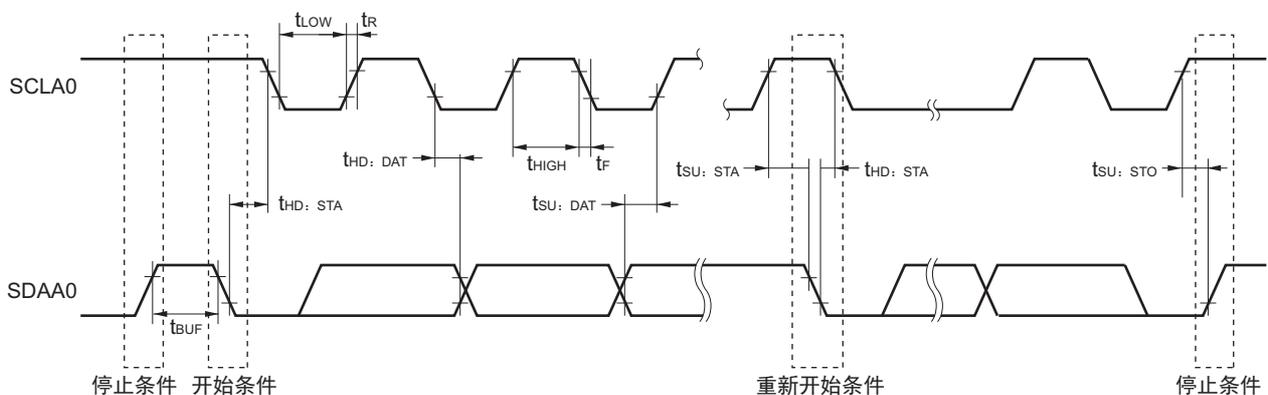
2. $t_{\text{HD: DAT}}$ 的最大值 (MAX.) 是通常传送时的数值, 在进行应答 ACK (应答) 时需要等待。

备注 各模式的 C_b (通信线电容) 的 MAX. 值和此时的 R_b (通信线的上拉电阻值) 的值如下所示。

标准模式: $C_b = 400\text{pF}$ 、 $R_b = 2.7\text{k}\Omega$

快速模式: $C_b = 200\text{pF}$ 、 $R_b = 1.7\text{k}\Omega$

IICA 串行传送时序



24.6 模拟特性

24.6.1 A/D 转换器特性

转换对象为：ANI0 ~ ANI6、内部基准电压 ($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
分辨率	RES		8		10	bit	
综合误差 注 1、2、3	AINL	10 位分辨率 $V_{DD} = 5\text{V}$		± 1.7	± 3.1	LSB	
			$V_{DD} = 3\text{V}$		± 2.3	± 4.5	LSB
转换时间	t_{CONV}	10 位分辨率 转换对象：ANI0 ~ ANI6	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	3.4		18.4	μs
			$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 注 5	4.6		18.4	
		10 位分辨率 转换对象：内部基准电压注 6	$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$	4.6		18.4	
零刻度误差 注 1、2、3、4	E_{ZS}	10 位分辨率 $V_{DD} = 5\text{V}$			± 0.19	%FSR	
			$V_{DD} = 3\text{V}$			± 0.39	%FSR
满刻度误差 注 1、2、3、4	E_{FS}	10 位分辨率 $V_{DD} = 5\text{V}$			± 0.29	%FSR	
			$V_{DD} = 3\text{V}$			± 0.42	%FSR
积分线性误差 注 1、2、3	ILE	10 位分辨率 $V_{DD} = 5\text{V}$			± 1.8	LSB	
			$V_{DD} = 3\text{V}$			± 1.7	LSB
微分线性误差 注 1、2、3	DLE	10 位分辨率 $V_{DD} = 5\text{V}$			± 1.4	LSB	
			$V_{DD} = 3\text{V}$			± 1.5	LSB
模拟输入电压	V_{AIN}	转换对象：ANI0 ~ ANI6	0		V_{DD}	V	
		内部基准电压注 6			V_{BGR} 注 7		

注 1. TYP. 值是 $T_A = 25^\circ\text{C}$ 时的平均值。MAX. 值是正态分布平均值 $\pm 3\sigma$ 的值。

2. 这是通过特性评估结果得出的值，未进行出厂检查。

3. 不包含量化误差 ($\pm 1/2$ LSB)。

4. 用对满刻度值的比率 (%FSR) 表示。

5. 在 $2.4\text{V} \leq V_{DD} \leq 2.7\text{V}$ 的工作电压范围内进行转换运行时，必须将 A/D 转换器模式寄存器 0 (ADM0) 的 LV0 位置“0”。

6. 选择内部基准电压作为 A/D 转换器的转换对象时，必须将 A/D 转换器模式寄存器 0 (ADM0) 的 LV0 位置“0”。

7. 请参照“24.6.3 内部基准电压特性”。

注意 1. 布线时为了避免电源 / 接地线进入噪声，实施插入电容器等措施。

2. 在 A/D 转换过程中，必须注意避免向转换引脚的邻接引脚输入或者输出如数字信号般发生急剧变化的脉冲。

3. 选择内部基准电压作为 A/D 转换器的转换对象时，比较器的基准电压就不能使用内部基准电压。

24.6.2 比较器特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
输入电压范围	V_{REF}	IVREF0 引脚输入 ($\text{C0VRF}=0$)	0		$V_{DD}-1.4$	V
		内部基准电压 ($\text{C0VRF}=1$) 注 1			V_{REG} 注 2	V
	V_{CMP}	IVCMP0 引脚输入	-0.3		$V_{DD}+0.3$	V
输出延迟	t_{d}	$V_{DD} = 3.0\text{V}$ 输入转换速率 $> 50\text{mV}/\mu\text{s}$	高速模式		0.5	μs
			低速模式		2.0	μs
运行稳定等待时间	t_{CMP}		100			μs

注 1. 选择内部基准电压作为比较器的基准电压时，A/D 转换器的转换对象就不能使用内部基准电压。

2. 请参照“24.6.3 内部基准电压特性”。

24.6.3 内部基准电压特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $2.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
内部基准电压	V_{REG}		0.74	0.815	0.89	V
运行稳定等待时间	t_{AMP}	使用 A/D 转换器时 (ADS 寄存器 =07H)	5			μs

注意 内部基准电压不能同时用于 A/D 转换器和比较器。只能其中一个选择内部基准电压。

24.6.4 SPOR 电路特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{SPOR0}	电源电压上升时	4.08	4.28	4.45	V
		电源电压下降时	4.00	4.20	4.37	V
	V_{SPOR1}	电源电压上升时	2.76	2.90	3.02	V
		电源电压下降时	2.70	2.84	2.96	V
	V_{SPOR2}	电源电压上升时	2.44	2.57	2.68	V
		电源电压下降时	2.40	2.52	2.62	V
	V_{SPOR3}	电源电压上升时	2.05	2.16	2.25	V
		电源电压下降时	2.00	2.11	2.20	V
最小脉宽注	T_{SPW}		300			μs

注 这是在 V_{DD} 低于 V_{SPOR} 时 SPOR 复位运行所需的时间。

注意 检测电压 (V_{SPOR}) 必须设定在工作电压范围内。工作电压范围取决于用户选项字节 (000C2H) 的设定。工作电压范围如下所示。

CPU 工作频率 1MHz ~ 20MHz: $V_{DD} = 2.7 \sim 5.5\text{V}$

CPU 工作频率 1MHz ~ 5MHz: $V_{DD} = 2.0 \sim 5.5\text{V}$

24.6.5 电源电压上升斜率特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $V_{SS} = 0\text{V}$)

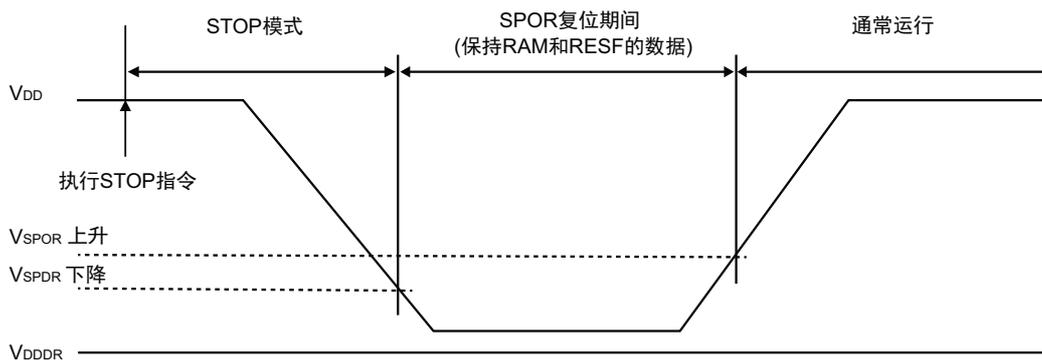
项目	符号	条件	MIN.	TYP.	MAX.	单位
电源电压上升斜率	S_{VDD}				54	V/ms

24.7 RAM 数据保持特性

($T_A = -40 \sim +85^\circ\text{C}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电源电压	V_{DDDR}		1.9		5.5	V

注意 保持 RAM 的数据直到电源电压小于数据保持电源电压 (V_{DDDR}) 的 MIN. 值。即使电源电压小于数据保持电源电压 (V_{DDDR}) 的 MIN. 值, RESF 寄存器的数据也有可能不被清除。



24.8 闪存编程特性

($T_A = 0 \sim +40^\circ\text{C}$ 、 $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
代码闪存的改写次数注 1、2、3	C_{erwr}	保持年数: 20 年 $T_A = 85^\circ\text{C}$	1000			次

- 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 次改写到下次改写为止的期间。
 2. 这是使用闪存编程器的情况。
 3. 此特性表示闪存的特性, 是本公司的可靠性试验的结果。

24.9 专用闪存编程器通信 (UART)

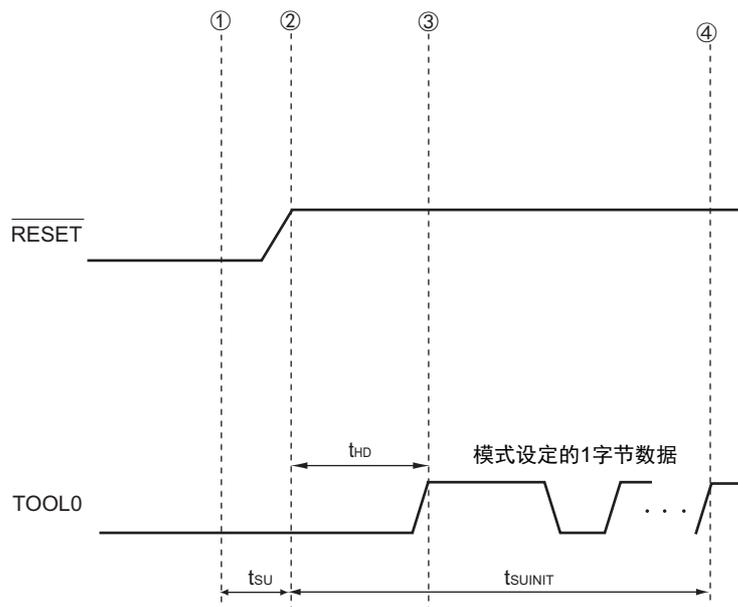
($T_A = 0 \sim +40^\circ\text{C}$ 、 $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS} = 0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率				115200		bps

备注 闪存编程时的传送速率固定为 115200bps。

24.10 闪存编程模式的转移时序

项目	符号	条件	MIN.	TYP.	MAX.	单位
从解除外部复位到结束初始设定的通信为止的时间	$t_{S\text{UNIT}}$	在解除外部复位前，解除 SPOR 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_{SU}	在解除外部复位前，解除 SPOR 复位。	10			μs
在解除外部复位后保持 TOOL0 引脚为低电平的时间	t_{HD}	在解除外部复位前，解除 SPOR 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 SPOR 复位）。
- ③ 解除 TOOL0 引脚的低电平。
- ④ 通过 UART 接收，完成模式转移的设定。

备注 $t_{S\text{UNIT}}$: 在此区间，必须在解除复位后的 100ms 之内完成通信的初始设定。

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后保持 TOOL0 引脚为低电平的时间（闪存固件处理时间除外）。

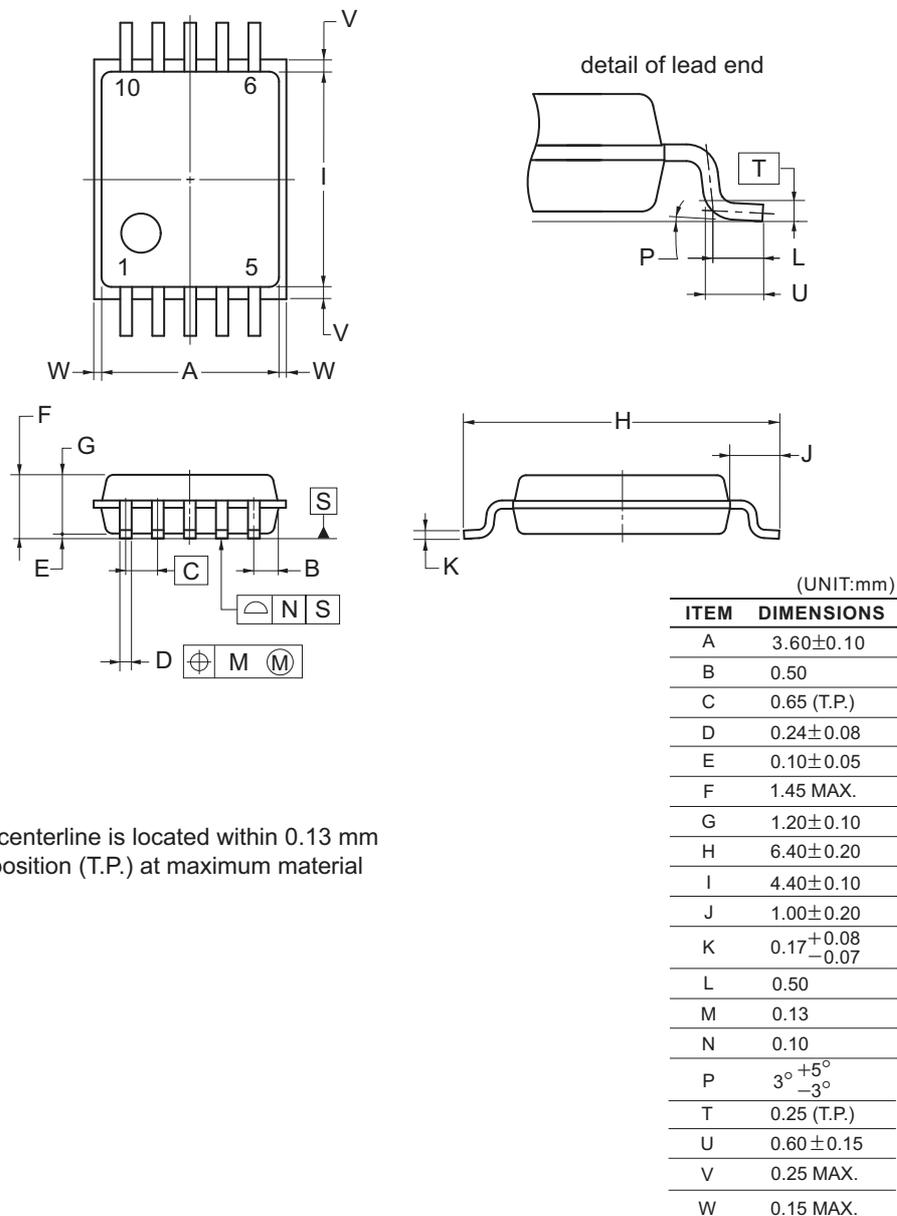
第 25 章 封装尺寸图

25.1 10 引脚产品

R5F10Y17ASP、R5F10Y16ASP、R5F10Y14ASP

R5F10Y17DSP 注、R5F10Y16DSP 注、R5F10Y14DSP 注

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP10-4.4x3.6-0.65	PLSP0010JA-A	P10MA-65-CAC-2	0.05

**NOTE**

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

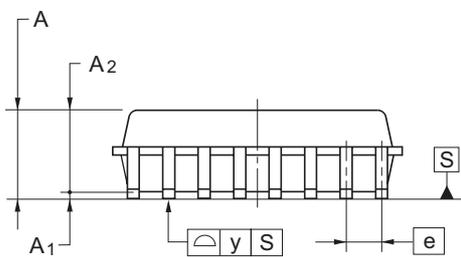
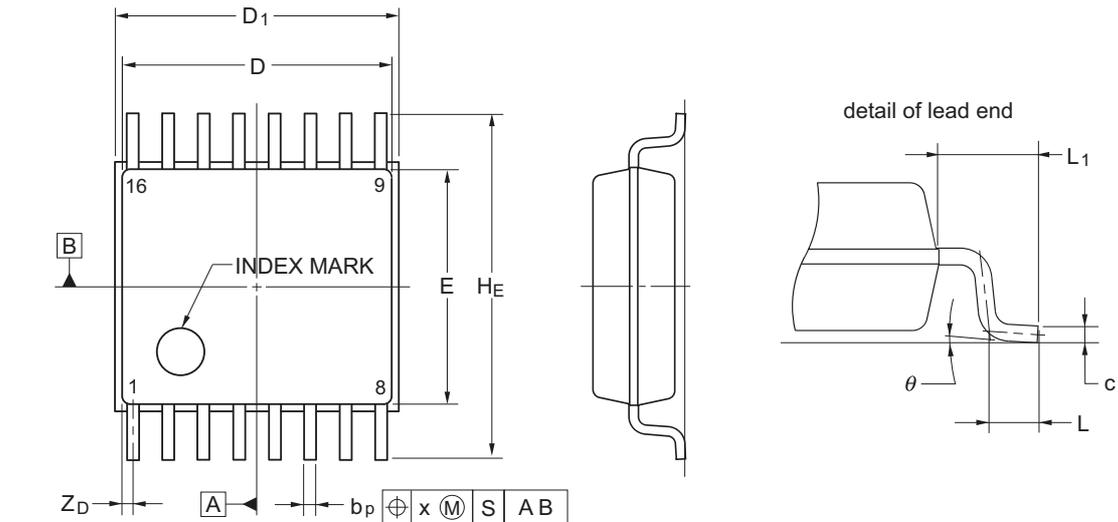
©2012 Renesas Electronics Corporation. All rights reserved.

注 开发中

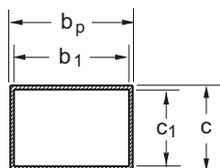
25.2 16 引脚产品

R5F10Y47ASP、R5F10Y46ASP、R5F10Y44ASP
 R5F10Y47DSP注、R5F10Y46DSP注、R5F10Y44DSP注

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-SSOP16-4.4x5-0.65	PRSP0016JC-B	P16MA-65-FAB-1	0.08



Terminal cross section



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	4.85	5.00	5.15
D1	5.05	5.20	5.35
E	4.20	4.40	4.60
A ₂	—	1.50	—
A ₁	0.075	0.125	0.175
A	—	—	1.725
b _p	0.17	0.24	0.32
b ₁	—	0.22	—
c	0.14	0.17	0.20
c ₁	—	0.15	—
θ	0°	—	8°
H _E	6.20	6.40	6.60
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z _D	—	0.225	—
L	0.35	0.50	0.65
L ₁	—	1.00	—

注 开发中

附录 A 修订记录

Rev.	发行日	修订内容	
		页	修订处
2.00	2014.03.18	—	初版发行
3.00	2015.03.02	2	更改 1.1 中 ROM、RAM 容量表。
		3	更改图 1-1 和表 1-1。
		4	在 1.3.1 和 1.3.2 中图中追加“RL78/G10（俯视图）”。
		5	将 1.4 中 INTPO ~ INTP3 的“External Interrupt Input”更改为“Interrupt Request From Peripheral”，X1、X2 的“Crystal Oscillator”更改为“Crystal Oscillator (Main System Clock)”。
		8	更改 1.6 表中的引脚名称。
		9、10	将 2.1.1 和 2.1.2 表中的“复位后”更改为“复位解除后”。
		21 ~ 23	更改图 3-1 ~ 图 3-3 的标题。
		24	更改表 3-1 中的产品名称。
		26	更改 3.1.2 中例标题，将 3.1.2 图中的“MOV A, !789H”更改为“MOV A, !8789H”。
		27	更改表 3-3 中的产品名称。
		37	将表 3-5(1/2) 中的“ADTEST”更改为“ADTES”。
		86	更改 5.4.3 第四段说明。
		94	追加 5.7。
		254	在 12.2.1 中追加图。
		271	更改图 12-14 的注意。
		279	在 12.5 表下说明中追加“CSIO1 注”。
		408	将图 13-21 中的“等待 3 个时钟”更改为“等待 3 个 f_{CLK} 时钟”。
		409	将图 13-22 中的“等待 5 个时钟”更改为“等待 5 个 f_{CLK} 时钟”。
		410	将图 13-23 中的“等待 3 个时钟”更改为“等待 3 个 f_{CLK} 时钟”。
		416	在图 13-27 中追加“解除串行接口 IICA0 的复位状态，开始提供时钟。”。
		417	在图 13-28(1/3) 中追加“解除串行接口 IICA0 的复位状态，开始提供时钟。”。
		418	更改图 13-28(2/3) 的注。
		421	在图 13-29(1) 中追加“解除串行接口 IICA0 的复位状态，开始提供时钟。”。
		445	将图 13-31(1/4) 下①中的“SDAA0=0”更改为“SDAA0”。
		466	在图 14-3 的注意 2 中追加“因此，在 C 语言中使用 8 位存储器操作指令时需要注意。”。
		501	更改图 17-5 的标题。
		507	在图 19-3 中追加“ f_{MAIN} ”和“ V_{DD} ”。
		513	更改 20.3.1。
		554	将 24.5.1(3) 表中的“ $t_{KCY2}/2$ ”更改为“ $t_{KCY2}/2-18$ ”。
		560	更改 24.7。
		562、563	更改 25.1 和 25.2。

RL78/G10
用户手册 硬件篇

Publication Date: Rev.3.00 Mar 02, 2015

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**9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004**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 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022**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 1207, 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 India Pvt. Ltd.**No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777**Renesas Electronics Korea Co., Ltd.**12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RL78/G10



瑞萨电子株式会社

R01UH0384CJ0300