

本产品仅在中国大陆和香港以及印度销售

R7F0C903-908

用户手册 硬件篇

16位单芯片微控制器

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
“High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled 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。用户如在产品或系统设计中应用本文档中的电路、软件和相关信息或将此等内容用于其他目的时，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和损害，瑞萨电子不承担任何责任。
2. 瑞萨电子在此明确声明，对于因使用瑞萨电子产品或本文档中所述技术信息（包括但不限于产品数据、图、表、程序、算法、应用实例）而造成的与第三方专利、版权或其他知识产权相关的侵权或任何其他索赔，瑞萨电子不作任何保证并概不承担责任。
3. 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. 用户不得对瑞萨电子的任何产品进行全部或部分更改、修改、复制或反向工程。对于用户或第三方因上述更改、修改、复制或反向工程的行为而遭受的任何损失或损害，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预期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、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（交通信号灯）、大型通讯设备、关键金融终端系统、安全控制设备等。
除非是瑞萨电子数据表或其他瑞萨电子文档中明确指定为高可靠性产品或用于恶劣环境的产品，否则瑞萨电子产品不能用于、亦未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（太空系统、海底增压机、核能控制系统、飞机控制系统、关键装置系统、军用设备等）中。对于用户或任何第三方因使用不符合瑞萨电子数据表、使用说明书或其他瑞萨电子文档的瑞萨电子产品而遭受的任何损害或损失，瑞萨电子不承担任何责任。
6. 使用瑞萨电子产品时，请参阅最新产品信息（数据表、使用说明书、应用指南、可靠性手册中的“半导体元件处理和使用一般注意事项”等），并确保使用条件在瑞萨电子指定的最大额定值、电源工作电压范围、散热特性、安装条件等范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的任何故障、失效或事故，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。除非是瑞萨电子数据表或其他瑞萨电子文档中指定为高可靠性产品或用于恶劣环境的产品，否则瑞萨电子产品未进行防辐射设计。用户负责执行安全保护措施，以避免因瑞萨电子产品失效或发生故障而造成身体伤害、火灾导致伤害或损害和/或其他对公众构成危险事故。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。由于对微机软件单独进行评估非常困难且不实际，所以请用户自行负责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。用户负责仔细并充分查阅对管制物质的使用或含量进行管理的所有适用法律法规（包括但不限于《欧盟 RoHS 指令》），并在使用瑞萨电子产品时遵守所有适用法律法规。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。也不可将瑞萨电子产品或技术用于(1)与大规模杀伤性武器（例如核武器、化学武器、生物武器或运送此等武器的导弹，包括无人机(UAV)）的开发、设计、制造、使用、存储等相关的任何目的；(2)与常规武器的开发、设计、制造或使用相关的任何目的；(3)扰乱国际和平与安全的任何其他目的，并且不可向任何第三方销售、出口、租赁、转让、或让与瑞萨电子产品或技术，无论直接或间接知悉或者有理由知悉该第三方或任何其他方将从事上述活动。用户必须遵守对各方或交易行使司法管辖权的任意国家/地区政府所公布和管理的任何适用出口管制法律法规。
10. 瑞萨电子产品的买方或分销商，或者分销、处置产品、或以其他方式向第三方出售或转让产品的任何其他方有责任事先向所述第三方通知本文档规定的内容和条件。
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再版、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，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。

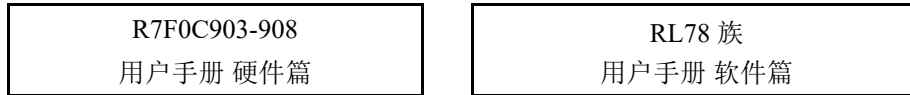
本手册的使用方法

对 象 本手册以理解 R7F0C903-908 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：

- 32 引脚： R7F0C903
R7F0C904
R7F0C905
R7F0C906
R7F0C907
R7F0C908

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 R7F0C903-908 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 R7F0C903-908 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）。

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
1.4 引脚名	5
1.5 框图	6
1.6 功能概要	7
第 2 章 引脚功能	9
2.1 端口功能	9
2.1.1 32 引脚产品	9
2.2 端口以外的功能	11
2.2.1 各产品配置的功能	11
2.2.2 功能说明	13
2.3 未使用引脚的处理	15
2.4 引脚框图	16
第 3 章 CPU 体系结构	27
3.1 存储空间	27
3.1.1 内部程序存储空间	32
3.1.2 镜像区	35
3.1.3 内部数据存储空间	36
3.1.4 特殊功能寄存器（SFR: Special Function Register）的区域	36
3.1.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）的区域	36
3.1.6 数据存储器的寻址	37
3.2 处理器的寄存器	38
3.2.1 控制寄存器	38
3.2.2 通用寄存器	40
3.2.3 ES 寄存器和 CS 寄存器	41
3.2.4 特殊功能寄存器（SFR: Special Function Register）	42
3.2.5 扩展特殊功能寄存器（2nd SFR: 2nd Special Function Register）	47
3.3 指令地址的寻址	53
3.3.1 相对寻址	53
3.3.2 立即寻址	53
3.3.3 表间接寻址	54
3.3.4 寄存器直接寻址	54
3.4 处理数据地址的寻址	55
3.4.1 隐含寻址	55
3.4.2 寄存器寻址	55
3.4.3 直接寻址	56
3.4.4 短直接寻址	57
3.4.5 SFR 寻址	58
3.4.6 寄存器间接寻址	59
3.4.7 基址寻址	60
3.4.8 基址变址寻址	64
3.4.9 堆栈寻址	65
第 4 章 端口功能	69
4.1 端口功能	69

4.2	端口结构	69
4.2.1	端口 0	69
4.2.2	端口 1	70
4.2.3	端口 2	70
4.2.4	端口 3	70
4.2.5	端口 4	71
4.2.6	端口 5	71
4.2.7	端口 6	71
4.2.8	端口 7	71
4.2.9	端口 12	71
4.2.10	端口 13	71
4.2.11	端口 14	72
4.3	控制端口功能的寄存器	73
4.3.1	端口模式寄存器 (PM _{xx})	74
4.3.2	端口寄存器 (P _{xx})	75
4.3.3	上拉电阻选择寄存器 (PU _{xx})	76
4.3.4	端口输入模式寄存器 (PIM _{xx})	77
4.3.5	端口输出模式寄存器 (POM _{xx})	77
4.3.6	端口模式控制寄存器 (PMC _{xx})	78
4.3.7	A/D 端口配置寄存器 (ADPC)	79
4.3.8	外围 I/O 重定向寄存器 (PIOR)	80
4.4	端口功能的运行	81
4.4.1	输入 / 输出端口的写操作	81
4.4.2	输入 / 输出端口的读操作	81
4.4.3	输入 / 输出端口的运算	81
4.4.4	通过输入 / 输出缓冲器进行的不同电位 (1.8V、2.5V、3V) 的对应	82
4.5	使用复用功能时的寄存器设定	84
4.5.1	使用复用功能时的基本思想	84
4.5.2	不使用输出功能的复用功能的寄存器设定	85
4.5.3	使用的端口功能和复用功能的寄存器设定例子	86
4.6	使用端口功能时的注意事项	91
4.6.1	有关对端口寄存器 n (P _n) 的位存储器操作指令的注意事项	91
4.6.2	有关引脚设定的注意事项	92
第 5 章	时钟发生电路	93
5.1	时钟发生电路的功能	93
5.2	时钟发生电路的结构	94
5.3	控制时钟发生电路的寄存器	96
5.3.1	时钟运行模式控制寄存器 (CMC)	96
5.3.2	系统时钟控制寄存器 (CKC)	97
5.3.3	时钟运行状态控制寄存器 (CSC)	98
5.3.4	振荡稳定时间计数器的状态寄存器 (OSTC)	99
5.3.5	振荡稳定时间选择寄存器 (OSTS)	100
5.3.6	外围允许寄存器 0 (PER0)	101
5.3.7	副系统时钟提供模式控制寄存器 (OSMC)	102
5.3.8	高速内部振荡器的频率选择寄存器 (HOCODIV)	103
5.3.9	高速内部振荡器的微调寄存器 (HIOTRM)	104
5.4	系统时钟振荡电路	105
5.4.1	X1 振荡电路	105
5.4.2	高速内部振荡器	108
5.4.3	低速内部振荡器	108
5.5	时钟发生电路的运行	108

5.6	时钟控制	110
5.6.1	高速内部振荡器的设定例子	110
5.6.2	X1 振荡电路的设定例子	112
5.6.3	CPU 时钟的状态转移图	113
5.6.4	CPU 时钟状态转移前的条件和转移后的处理	116
5.6.5	CPU 时钟和主系统时钟的切换所需时间	117
5.6.6	时钟振荡停止前的条件	117
5.7	谐振器和振荡电路常数	118
第 6 章	定时器阵列单元	121
6.1	定时器阵列单元的功能	122
6.1.1	独立通道运行功能	122
6.1.2	多通道联动运行功能	123
6.1.3	8 位定时器运行功能（只限于通道 1 和通道 3）	124
6.2	定时器阵列单元的结构	125
6.2.1	定时器计数寄存器 0n（TCR0n）	129
6.2.2	定时器数据寄存器 0n（TDR0n）	131
6.3	控制定时器阵列单元的寄存器	132
6.3.1	外围允许寄存器 0（PER0）	133
6.3.2	定时器时钟选择寄存器 0（TPS0）	134
6.3.3	定时器模式寄存器 0n（TMR0n）	136
6.3.4	定时器状态寄存器 0n（TSR0n）	141
6.3.5	定时器通道允许状态寄存器 0（TE0）	142
6.3.6	定时器通道开始寄存器 0（TS0）	143
6.3.7	定时器通道停止寄存器 0（TT0）	144
6.3.8	定时器输入选择寄存器 0（TIS0）	145
6.3.9	定时器输出允许寄存器 0（TOE0）	145
6.3.10	定时器输出寄存器 0（TO0）	146
6.3.11	定时器输出电平寄存器 0（TOL0）	147
6.3.12	定时器输出模式寄存器 0（TOM0）	148
6.3.13	噪声滤波器允许寄存器 1（NFEN1）	149
6.3.14	控制定时器输入 / 输出引脚端口功能的寄存器	150
6.4	定时器阵列单元的基本规则	151
6.4.1	多通道联动运行功能的基本规则	151
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）	153
6.5	计数器的运行	154
6.5.1	计数时钟（ f_{TCLK} ）	154
6.5.2	计数器的开始时序	156
6.5.3	计数器的运行	157
6.6	通道输出（TO0n 引脚）的控制	162
6.6.1	TO0n 引脚输出电路的结构	162
6.6.2	TO0n 引脚的输出设定	163
6.6.3	通道输出运行的注意事项	164
6.6.4	TO0n 位的一次性操作	168
6.6.5	有关开始计数时的定时器中断和 TO0n 引脚输出	169
6.7	定时器输入（TI0n）的控制	170
6.7.1	TI0n 引脚输入电路的结构	170
6.7.2	噪声滤波器	170
6.7.3	操作通道输入时的注意事项	171
6.8	定时器阵列单元的独立通道运行功能	172
6.8.1	作为间隔定时器 / 方波输出的运行	172
6.8.2	作为外部事件计数器的运行	176

6.8.3	作为分频器的运行	180
6.8.4	作为输入脉冲间隔测量的运行	184
6.8.5	作为输入信号高低电平宽度测量的运行	188
6.8.6	作为延迟计数器的运行	192
6.9	定时器阵列单元的多通道联动运行功能	196
6.9.1	作为单触发脉冲输出功能的运行	196
6.9.2	作为 PWM 功能的运行	203
6.9.3	作为多重 PWM 输出功能的运行	210
6.10	使用定时器阵列单元时的注意事项	217
6.10.1	使用定时器输出时的注意事项	217
第 7 章	12 位间隔定时器	218
7.1	12 位间隔定时器的功能	218
7.2	12 位间隔定时器的结构	218
7.3	控制 12 位间隔定时器的寄存器	219
7.3.1	外围允许寄存器 0 (PER0)	219
7.3.2	副系统时钟提供模式控制寄存器 (OSMC)	220
7.3.3	间隔定时器的控制寄存器 (ITMC)	221
7.4	12 位间隔定时器的运行	222
7.4.1	12 位间隔定时器的运行时序	222
7.4.2	从 HALT/STOP 模式返回后开始计数器的运行并且再次向 HALT/STOP 模式的转移	223
第 8 章	时钟输出 / 蜂鸣器输出控制电路	224
8.1	时钟输出 / 蜂鸣器输出控制电路的功能	224
8.2	时钟输出 / 蜂鸣器输出控制电路的结构	225
8.3	控制时钟输出 / 蜂鸣器输出控制电路的寄存器	226
8.3.1	时钟输出选择寄存器 n (CKSn)	226
8.3.2	控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器	227
8.4	时钟输出 / 蜂鸣器输出控制电路的运行	227
8.4.1	输出引脚的运行	227
8.5	时钟输出 / 蜂鸣器输出控制电路的注意事项	227
第 9 章	看门狗定时器	228
9.1	看门狗定时器的功能	228
9.2	看门狗定时器的结构	229
9.3	控制看门狗定时器的寄存器	230
9.3.1	看门狗定时器的允许寄存器 (WDTE)	230
9.4	看门狗定时器的运行	231
9.4.1	看门狗定时器的运行控制	231
9.4.2	看门狗定时器上溢时间的设定	232
9.4.3	看门狗定时器窗口打开期间的设定	233
9.4.4	看门狗定时器间隔中断的设定	234
第 10 章	A/D 转换器	235
10.1	A/D 转换器的功能	235
10.2	A/D 转换器的结构	237
10.3	控制 A/D 转换器的寄存器	238
10.3.1	外围允许寄存器 0 (PER0)	239
10.3.2	A/D 转换器的模式寄存器 0 (ADM0)	240
10.3.3	A/D 转换器的模式寄存器 1 (ADM1)	248
10.3.4	A/D 转换器的模式寄存器 2 (ADM2)	249
10.3.5	10 位 A/D 转换结果寄存器 (ADCR)	251

10.3.6	8 位 A/D 转换结果寄存器 (ADCRH)	251
10.3.7	模拟输入通道指定寄存器 (ADS)	252
10.3.8	转换结果比较上限值设定寄存器 (ADUL)	253
10.3.9	转换结果比较下限值设定寄存器 (ADLL)	253
10.3.10	A/D 测试寄存器 (ADTES)	254
10.3.11	控制模拟输入引脚端口功能的寄存器	254
10.4	A/D 转换器的转换运行	255
10.5	输入电压和转换结果	257
10.6	A/D 转换器的运行模式	258
10.6.1	软件触发模式 (选择模式、连续转换模式)	258
10.6.2	软件触发模式 (选择模式、单次转换模式)	259
10.6.3	软件触发模式 (扫描模式、连续转换模式)	260
10.6.4	软件触发模式 (扫描模式、单次转换模式)	261
10.6.5	硬件触发无等待模式 (选择模式、连续转换模式)	262
10.6.6	硬件触发无等待模式 (选择模式、单次转换模式)	263
10.6.7	硬件触发无等待模式 (扫描模式、连续转换模式)	264
10.6.8	硬件触发无等待模式 (扫描模式、单次转换模式)	265
10.6.9	硬件触发等待模式 (选择模式、连续转换模式)	266
10.6.10	硬件触发等待模式 (选择模式、单次转换模式)	267
10.6.11	硬件触发等待模式 (扫描模式、连续转换模式)	268
10.6.12	硬件触发等待模式 (扫描模式、单次转换模式)	269
10.7	A/D 转换器的设定流程图	270
10.7.1	软件触发模式的设定	270
10.7.2	硬件触发无等待模式的设定	271
10.7.3	硬件触发等待模式的设定	272
10.7.4	选择内部基准电压时的设定 (以软件触发模式、单次转换模式为例)	273
10.7.5	测试模式的设定	274
10.8	SNOOZE 模式功能	275
10.9	A/D 转换器特性表的阅读方法	279
10.10	A/D 转换器的注意事项	281
第 11 章	串行阵列单元	285
11.1	串行阵列单元的功能	286
11.1.1	3 线串行 I/O (CSI00、CSI11、CSI20)	286
11.1.2	UART (UART0 ~ UART2)	287
11.1.3	简易 I ² C (IIC00、IIC11、IIC20)	288
11.2	串行阵列单元的结构	289
11.2.1	移位寄存器	292
11.2.2	串行数据寄存器 mn (SDRmn) 的低 8 位或者低 9 位	292
11.3	控制串行阵列单元的寄存器	294
11.3.1	外围允许寄存器 0 (PER0)	295
11.3.2	串行时钟选择寄存器 m (SPSm)	296
11.3.3	串行模式寄存器 mn (SMRmn)	297
11.3.4	串行通信运行设定寄存器 mn (SCRmn)	298
11.3.5	串行数据寄存器 mn (SDRmn)	301
11.3.6	串行标志清除触发寄存器 mn (SIRmn)	302
11.3.7	串行状态寄存器 mn (SSRmn)	303
11.3.8	串行通道开始寄存器 m (SSm)	305
11.3.9	串行通道停止寄存器 m (STm)	306
11.3.10	串行通道允许状态寄存器 m (SEm)	307
11.3.11	串行输出允许寄存器 m (SOEm)	308
11.3.12	串行输出寄存器 m (SOM)	309

11.3.13	串行输出电平寄存器 m (SOLm)	310
11.3.14	串行待机控制寄存器 0 (SSC0)	312
11.3.15	噪声滤波器允许寄存器 0 (NFEN0)	313
11.3.16	控制串行输入 / 输出引脚端口功能的寄存器	314
11.4	运行停止模式	315
11.4.1	以单元为单位停止运行的情况	315
11.4.2	按通道停止运行的情况	316
11.5	3 线串行 I/O (CSI00、CSI11、CSI20) 通信的运行	317
11.5.1	主控发送	318
11.5.2	主控接收	326
11.5.3	主控的发送和接收	334
11.5.4	从属发送	342
11.5.5	从属接收	350
11.5.6	从属的发送和接收	356
11.5.7	SNOOZE 模式功能	364
11.5.8	传送时钟频率的计算	368
11.5.9	在 3 线串行 I/O (CSI00、CSI11、CSI20) 通信过程中发生错误时的处理步骤	370
11.6	UART (UART0 ~ UART2) 通信的运行	371
11.6.1	UART 发送	372
11.6.2	UART 接收	381
11.6.3	SNOOZE 模式功能	388
11.6.4	波特率的计算	395
11.6.5	在 UART (UART0 ~ UART2) 通信过程中发生错误时的处理步骤	399
11.7	简易 I ² C (IIC00、IIC11、IIC20) 通信的运行	400
11.7.1	地址段发送	401
11.7.2	数据发送	406
11.7.3	数据接收	409
11.7.4	停止条件的产生	413
11.7.5	传送速率的计算	414
11.7.6	在简易 I ² C (IIC00、IIC11、IIC20) 通信过程中发生错误时的处理步骤	416
第 12 章	串行接口 IICA	417
12.1	串行接口 IICA 的功能	417
12.2	串行接口 IICA 的结构	420
12.3	控制串行接口 IICA 的寄存器	422
12.3.1	外围允许寄存器 0 (PER0)	423
12.3.2	IICA 控制寄存器 n0 (IICCTLn0)	423
12.3.3	IICA 状态寄存器 n (IICSn)	428
12.3.4	IICA 标志寄存器 n (IICFn)	430
12.3.5	IICA 控制寄存器 n1 (IICCTLn1)	432
12.3.6	IICA 低电平宽度设定寄存器 n (IICWLn)	434
12.3.7	IICA 高电平宽度设定寄存器 n (IICWHn)	434
12.3.8	端口模式寄存器 6 (PM6)	435
12.4	I ² C 总线模式的功能	436
12.4.1	引脚结构	436
12.4.2	通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法	437
12.5	I ² C 总线的定义和控制方法	438
12.5.1	开始条件	438
12.5.2	地址	439
12.5.3	传送方向的指定	439
12.5.4	应答 (ACK)	440
12.5.5	停止条件	441

12.5.6	等待	442
12.5.7	等待的解除方法	444
12.5.8	中断请求 (INTIICAn) 的产生时序和等待控制	445
12.5.9	地址匹配的检测方法	446
12.5.10	错误的检测	446
12.5.11	扩展码	446
12.5.12	仲裁	447
12.5.13	唤醒功能	449
12.5.14	通信预约	452
12.5.15	其他注意事项	455
12.5.16	通信运行	456
12.5.17	I ² C 中断请求 (INTIICAn) 的产生时序	463
12.6	时序图	484
第 13 章 乘除器和乘加器		499
13.1	乘除器和乘加器的功能	499
13.2	乘除器和乘加器的结构	499
13.2.1	乘除数据寄存器 A (MDAH、MDAL)	501
13.2.2	乘除数据寄存器 B (MDBL、MDBH)	502
13.2.3	乘除数据寄存器 C (MDCL、MDCH)	503
13.3	控制乘除器和乘加器的寄存器	505
13.3.1	乘除控制寄存器 (MDUC)	505
13.4	乘除器和乘加器的运算	506
13.4.1	乘法运算 (无符号)	506
13.4.2	乘法运算 (带符号)	507
13.4.3	乘加运算 (无符号)	508
13.4.4	乘加运算 (带符号)	509
13.4.5	除法运算	511
第 14 章 DMA 控制器		513
14.1	DMA 控制器的功能	513
14.2	DMA 控制器的结构	514
14.2.1	DMA SFR 地址寄存器 n (DSAn)	514
14.2.2	DMA RAM 地址寄存器 n (DRAn)	515
14.2.3	DMA 字节计数寄存器 n (DBCn)	516
14.3	控制 DMA 控制器的寄存器	517
14.3.1	DMA 模式控制寄存器 n (DMCn)	517
14.3.2	DMA 运行控制寄存器 n (DRCn)	519
14.4	DMA 控制器的运行	520
14.4.1	运行步骤	520
14.4.2	传送模式	521
14.4.3	DMA 传送的结束	521
14.5	DMA 控制器的设定例子	521
14.5.1	CSI 连续发送	521
14.5.2	A/D 转换结果的连续读取	523
14.5.3	UART 连续接收 +ACK 发送	524
14.5.4	通过 DWAITn 位保留 DMA 传送	525
14.5.5	通过软件强制结束	526
14.6	DMA 控制器的注意事项	528
第 15 章 中断功能		531
15.1	中断功能的种类	531

15.2	中断源和结构	531
15.3	控制中断功能的寄存器	535
15.3.1	中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)	538
15.3.2	中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)	539
15.3.3	优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)	541
15.3.4	外部中断上升沿允许寄存器 (EGP0)、外部中断下降沿允许寄存器 (EGN0)	543
15.3.5	程序状态字 (PSW)	544
15.4	中断处理的操作	545
15.4.1	可屏蔽中断请求的接受	545
15.4.2	软件中断请求的接受	548
15.4.3	多重中断处理	548
15.4.4	中断请求的保留	552
第 16 章	待机功能	553
16.1	待机功能	553
16.2	控制待机功能的寄存器	554
16.3	待机功能的运行	554
16.3.1	HALT 模式	554
16.3.2	STOP 模式	557
16.3.3	SNOOZE 模式	562
第 17 章	复位功能	565
17.1	复位时序	567
17.2	复位期间的运行状态	569
17.3	确认复位源的寄存器	571
17.3.1	复位控制标志寄存器 (RESF)	571
第 18 章	上电复位电路	574
18.1	上电复位电路的功能	574
18.2	上电复位电路的结构	575
18.3	上电复位电路的运行	575
第 19 章	电压检测电路	579
19.1	电压检测电路的功能	579
19.2	电压检测电路的结构	580
19.3	控制电压检测电路的寄存器	581
19.3.1	电压检测寄存器 (LVIM)	581
19.3.2	电压检测电平寄存器 (LVIS)	582
19.4	电压检测电路的运行	585
19.4.1	用作复位模式时的设定	585
19.4.2	用作中断模式时的设定	587
19.4.3	用作中断 & 复位模式时的设定	589
19.5	电压检测电路的注意事项	595
第 20 章	安全功能	597
20.1	安全功能的概要	597
20.2	安全功能使用的寄存器	598
20.3	安全功能的运行	598
20.3.1	闪存 CRC 运算功能 (高速 CRC)	598
20.3.2	CRC 运算功能 (通用 CRC)	601
20.3.3	RAM 奇偶校验错误检测功能	603

20.3.4	RAM 保护功能	604
20.3.5	SFR 保护功能	605
20.3.6	非法存储器存取检测功能	606
20.3.7	频率检测功能	608
20.3.8	A/D 测试功能	609
第 21 章	稳压器	613
21.1	稳压器的概要	613
第 22 章	选项字节	614
22.1	选项字节的功能	614
22.1.1	用户选项字节 (000C0H ~ 000C2H/010C0H ~ 010C2H)	614
22.1.2	片上调试选项字节 (000C3H/010C3H)	615
22.2	用户选项字节的格式	616
22.3	片上调试选项字节的格式	621
22.4	选项字节的设定	622
第 23 章	闪存	623
23.1	使用闪存编程器的串行编程	624
23.1.1	编程环境	625
23.1.2	通信方式	625
23.2	使用外部器件 (内置 UART) 的串行编程	626
23.2.1	编程环境	626
23.2.2	通信方式	627
23.3	电路板上的引脚处理	628
23.3.1	P40/TOOL0 引脚	628
23.3.2	RESET 引脚	628
23.3.3	端口引脚	629
23.3.4	REGC 引脚	629
23.3.5	X1 引脚和 X2 引脚	629
23.3.6	电源	629
23.4	串行编程方法	629
23.4.1	串行编程的步骤	629
23.4.2	闪存编程模式	630
23.4.3	通信方式	631
23.4.4	通信命令	632
23.5	使用 PG-FP5 时的各命令处理时间 (参考值)	633
23.6	自编程	634
23.6.1	自编程的步骤	635
23.6.2	引导交换功能	636
23.6.3	闪存屏蔽窗口功能	638
23.7	安全设定	639
23.8	数据闪存	641
23.8.1	数据闪存的概要	641
23.8.2	控制数据闪存的寄存器	641
23.8.3	数据闪存的存取步骤	642
第 24 章	片上调试功能	643
24.1	和 E1 片上调试仿真器的连接	643
24.2	片上调试安全 ID	644
24.3	用户资源的确保	644

第 25 章 十进制校正 (BCD) 电路	646
25.1 十进制校正电路的功能	646
25.2 十进制校正电路使用的寄存器	646
25.2.1 BCD 校正结果寄存器 (BCDADJ)	646
25.3 十进制校正电路的运行	647
第 26 章 指令集的概要	649
26.1 凡例	649
26.1.1 操作数的表现形式和记述方法	649
26.1.2 操作栏的说明	650
26.1.3 标志栏的说明	651
26.1.4 PREFIX 指令	651
26.2 操作一览表	652
第 27 章 电特性 ($T_A=-40 \sim +85^{\circ}\text{C}$)	669
27.1 绝对最大额定值	669
27.2 振荡电路特性	671
27.2.1 X1 振荡电路特性	671
27.2.2 内部振荡器特性	671
27.3 DC 特性	672
27.3.1 引脚特性	672
27.3.2 电源电流特性	677
27.4 AC 特性	680
27.5 外围功能特性	684
27.5.1 串行阵列单元	684
27.5.2 串行接口 IICA	711
27.6 模拟特性	714
27.6.1 A/D 转换器特性	714
27.6.2 内部基准电压的特性	718
27.6.3 POR 电路特性	718
27.6.4 LVD 电路特性	719
27.6.5 电源电压的上升斜率特性	720
27.7 RAM 数据保持特性	721
27.8 闪存编程特性	721
27.9 专用闪存编程器通信 (UART)	721
27.10 闪存编程模式的转移时序	722
第 28 章 封装尺寸图	723
附录 A 修订记录	724

第 1 章 概述

1.1 特点

超低功耗技术

- 工作电源为 1.6V~5.5V 的单电源。
- STOP 模式：0.23 μ A（RAM 保持）、0.31 μ A（LVD 工作）
- HALT 模式（LVD 工作）：0.57 μ A
- SNOOZE 模式：0.70mA（UART）、1.20mA（A/D）
- 运行模式：66 μ A/MHz

16 位 RL78 CPU 内核

- 在最高工作频率 24MHz 时为 31 DMIPS。
- 指令执行：86% 的指令执行能在 1~2 个时钟周期内完成。
- 3 段流水线的 CISC 体系结构（哈佛）
- 在 1 个时钟周期内完成 16 位 \times 16 位=32 位的带符号或者无符号乘法运算。
- 在 2 个时钟周期内完成 16 位 \times 16 位+32 位=32 位的乘加运算。
- 在 1 个时钟周期内完成 16 位桶式移位器的移位和循环。
- 内置单线片上调试功能。

代码闪存

- 闪存：16KB~48KB
- 块大小：1KB
- 内置单电源闪存（带块擦除/编程保护功能）。
- 自编程：有引导交换功能和闪存屏蔽窗口功能。

数据闪存

- 带后台操作功能的数据闪存
- 数据闪存：2KB 或者不带数据闪存
- 擦除次数：1000000 次(TYP.)
- 擦除/编程电压： $V_{DD}=1.8\sim 5.5V$

RAM

- 2KB 或者 3KB
- 支持操作数和指令。
- 在所有模式中保持数据。

高速内部振荡器

- 24MHz $\pm 1.0\%$ 高精度（ $V_{DD}=1.8\sim 5.5V$ 、 $T_A=-20\sim +85^\circ C$ ）
- 可选择 24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 或者 1MHz。

电源管理和复位功能

- 上电复位（POR）监视/发生电路
- 有 14 种设定选项（中断和复位）的低电压检测（LVD）

数据存储器存取（DMA）控制器

- 2个通道
- 传送单位：8位或者16位

多种通信接口

- I²C多主控：1个通道
- CSI/UART/简易I²C：最多3个通道

扩展功能定时器

- 多功能16位定时器：最多8个通道
- 12位间隔定时器：1个通道
- 15kHz看门狗定时器：1个通道（窗口功能）

丰富的模拟模块

- A/D转换器：最多8个通道、10位分辨率、2.1μs转换时间
- 支持1.6V。
- 内部基准电压（1.45V）

安全功能（符合 IEC 或者 UL 60730）

- 闪存CRC运算
- RAM奇偶校验错误检测
- RAM写保护
- SFR写保护
- 非法存储器存取检测
- 时钟停止/频率检测
- A/D测试功能

输入/输出端口

- 5V耐压、大电流（各引脚最大20mA）
- 支持N沟道漏极开路 and 内部上拉。
- 支持不同电位的接口：能连接1.8V、2.5V或者3V的器件。

工作环境温度

- 标准：-40~+85°C

封装和引脚数

- 32引脚LQFP（7×7mm，0.8mm间距）

○ ROM、RAM容量

闪存 ROM	数据闪存	RAM	R7F0C903-908
			32 引脚
48KB	2KB	3KB	R7F0C908B2
	—		R7F0C905B2
32KB	2KB	2KB	R7F0C907B2
	—		R7F0C904B2
16KB	2KB	2KB	R7F0C906B2
	—		R7F0C903B2

1.2 产品型号一览表

图 1-1 R7F0C903-908 的产品型号、存储容量和封装

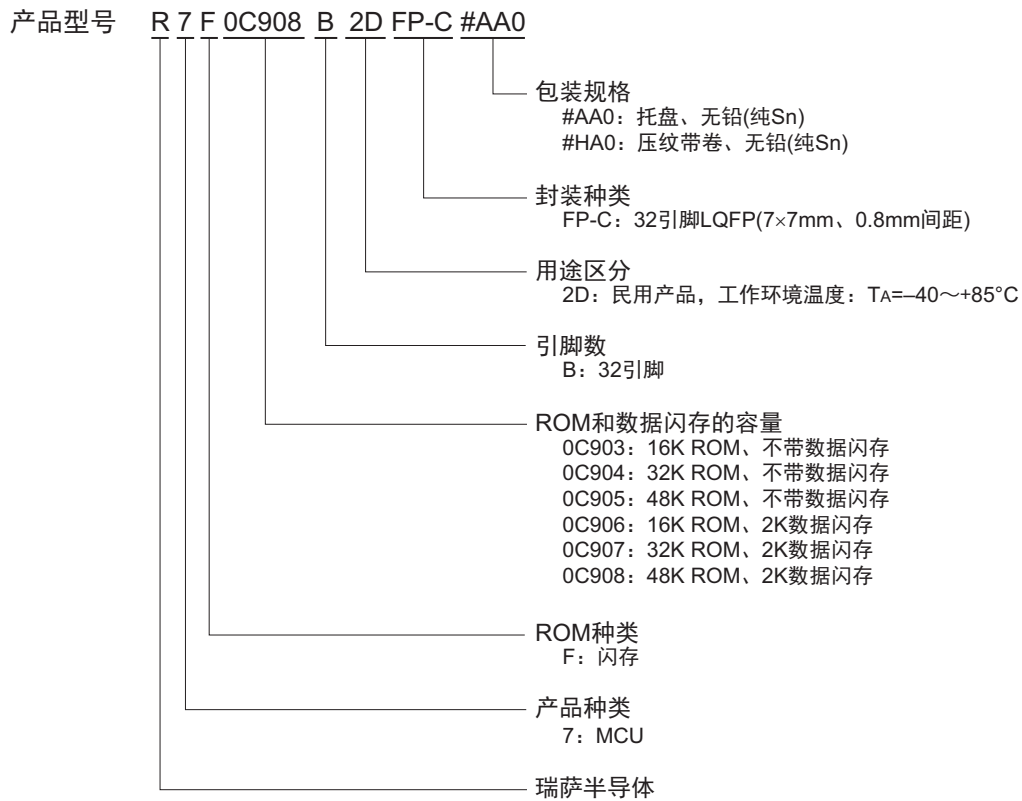


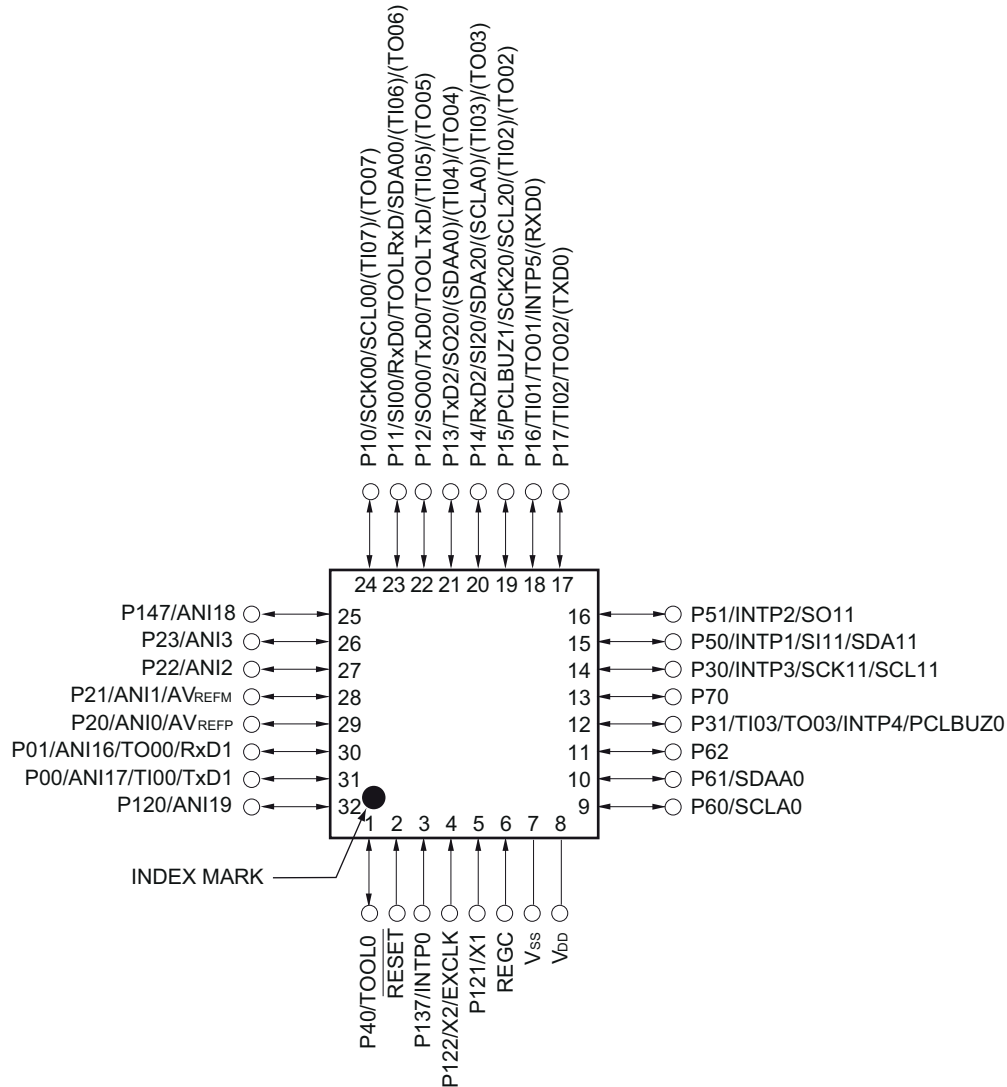
表 1-1 产品型号一览表

引脚数	封装	闪存 ROM	数据闪存	RAM	包装规格和环境对应	订购产品型号	
32 引脚	32 引脚 LQFP (7×7mm、0.8mm 间距)	48KB	2KB	3KB	托盘、无铅 (纯 Sn)	R7F0C908B2DFP-C#AA0	
					压纹带卷、无铅 (纯 Sn)	R7F0C908B2DFP-C#HA0	
		32KB	2KB	2KB	托盘、无铅 (纯 Sn)	R7F0C907B2DFP-C#AA0	
					压纹带卷、无铅 (纯 Sn)	R7F0C907B2DFP-C#HA0	
					托盘、无铅 (纯 Sn)	R7F0C906B2DFP-C#AA0	
					压纹带卷、无铅 (纯 Sn)	R7F0C906B2DFP-C#HA0	
		16KB	—	3KB	托盘、无铅 (纯 Sn)	R7F0C905B2DFP-C#AA0	
					压纹带卷、无铅 (纯 Sn)	R7F0C905B2DFP-C#HA0	
		48KB	—	—	3KB	托盘、无铅 (纯 Sn)	R7F0C905B2DFP-C#AA0
						压纹带卷、无铅 (纯 Sn)	R7F0C905B2DFP-C#HA0
						托盘、无铅 (纯 Sn)	R7F0C904B2DFP-C#AA0
						压纹带卷、无铅 (纯 Sn)	R7F0C904B2DFP-C#HA0
32KB	2KB	2KB	托盘、无铅 (纯 Sn)	R7F0C904B2DFP-C#AA0			
			压纹带卷、无铅 (纯 Sn)	R7F0C904B2DFP-C#HA0			
16KB	2KB	2KB	托盘、无铅 (纯 Sn)	R7F0C903B2DFP-C#AA0			
			压纹带卷、无铅 (纯 Sn)	R7F0C903B2DFP-C#HA0			

注意 订购产品型号是本手册发行时的型号。最新的订购产品型号请参照本公司网页的对象产品页。

1.3 引脚连接图（俯视图）

- 32 引脚 LQFP（7×7mm、0.8mm 间距）



注意 必须通过电容器（0.47 ~ 1μF）将 REGC 引脚连接 V_{SS} 引脚。

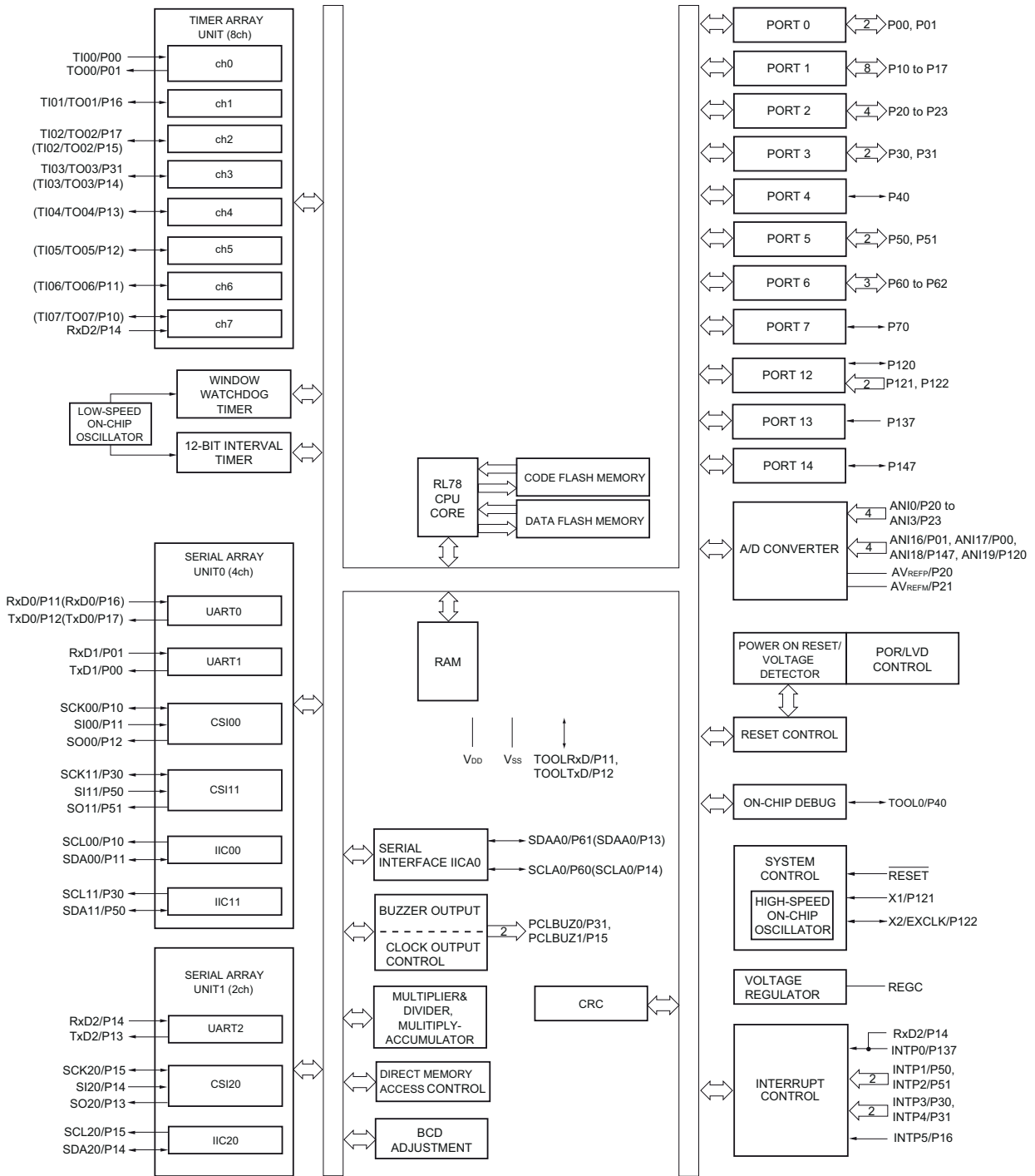
备注 1. 有关引脚名，请参照“1.4 引脚名”。

2. 能通过设定外围 I/O 重定向寄存器（PIOR），分配上图（）内的功能。详细内容请参照“图 4-8 外围 I/O 重定向寄存器（PIOR）的格式”。

1.4 引脚名

ANI0 ~ ANI3、		REGC:	Regulator capacitance
ANI16 ~ ANI19:	Analog input	RESET:	Reset
AVREFM:	A/D converter reference potential (–side) input	RxD0 ~ RxD2:	Receive data
AVREFP:	A/D converter reference potential (+side) input	SCK00、SCK11、SCK20、	
EXCLK:	External clock input (main system clock)	SCLA0:	Serial clock input/output
INTP0 ~ INTP5:	Interrupt request from peripheral	SCL00、SCL11、SCL20:	Serial clock output
P00、P01:	Port 0	SDA00、SDA11、SDA20、	
P10 ~ P17:	Port 1	SDAA0:	Serial data input/output
P20 ~ P23:	Port 2	SI00、SI11、SI20:	Serial data input
P30、P31:	Port 3	SO00、SO11、SO20:	Serial data output
P40:	Port 4	TI00 ~ TI07:	Timer input
P50、P51:	Port 5	TO00 ~ TO07:	Timer output
P60 ~ P62:	Port 6	TOOL0:	Data input/output for tool
P70:	Port 7	TOOLRxD、TOOLTxD:	Data input/output for external device
P120 ~ P122:	Port 12	TxD0 ~ TxD2:	Transmit data
P137:	Port 13	V _{DD} :	Power supply
P147:	Port 14	V _{SS} :	Ground
PCLBUZ0、PCLBUZ1:	Programmable clock output/buzzer output	X1、X2:	Crystal oscillator (main system clock)

1.5 框图



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。详细内容请参照“图 4-8 外围 I/O 重定向寄存器 (PIOR) 的格式”。

1.6 功能概要

注意 这是将外围 I/O 重定向寄存器 (PIOR) 置“00H”时的功能概要。

(1/2)

项目		32 引脚	
		R7F0C906/7/8	R7F0C903/4/5
代码闪存 (KB)		16 ~ 48	
数据闪存 (KB)		2	—
RAM (KB)		2 或者 3	
地址空间		1MB	
主系统时钟	高速系统时钟	X1 (晶体 / 陶瓷) 振荡、外部主系统时钟输入 (EXCLK) 1 ~ 20MHz: V _{DD} =2.7 ~ 5.5V, 1 ~ 8MHz: V _{DD} =1.8 ~ 2.7V, 1 ~ 4MHz: V _{DD} =1.6 ~ 1.8V	
	高速内部振荡器	HS (高速主) 模式: 1 ~ 24MHz (V _{DD} =2.7 ~ 5.5V) HS (高速主) 模式: 1 ~ 16MHz (V _{DD} =2.4 ~ 5.5V) LS (低速主) 模式: 1 ~ 8MHz (V _{DD} =1.8 ~ 5.5V) LV (低电压主) 模式: 1 ~ 4MHz (V _{DD} =1.6 ~ 5.5V)	
副系统时钟		—	
低速内部振荡器		15kHz(TYP.)	
通用寄存器		(8 位 × 8 个寄存器) × 4 组	
最短指令执行时间		0.04167μs (24MHz 运行时)	
		0.05μs (高速系统时钟: f _{MX} =20MHz 运行时)	
指令集		<ul style="list-style-type: none"> • 数据传送 (8/16 位) • 加减 / 逻辑运算 (8/16 位) • 乘法运算 (8 位 × 8 位) • 循环、桶式移位、位操作 (置位、复位、测试和布尔运算) 等 	
I/O 端口	合计	28	
	CMOS 输入 / 输出	22 (N 沟道漏极开路输入 / 输出 [V _{DD} 耐压]: 9)	
	CMOS 输入	3	
	CMOS 输出	—	
	N 沟道漏极开路输入 / 输出 (6V 耐压)	3	
定时器	16 位定时器	8 个通道	
	看门狗定时器	1 个通道	
	12 位间隔定时器 (IT)	1 个通道	
	定时器输出	4 个通道 (PWM 输出: 3 个注 1)、8 个通道 (PWM 输出: 7 个注 1) 注 2	

注 1. PWM 输出的通道数因所用通道的设定 (主控和从属的个数) 而变 (参照“6.9.3 作为多重 PWM 输出功能的运行”)。

2. 这是将 PIOR0 置“1”的情况。

(2/2)

项目	32 引脚	
	R7F0C906/7/8	R7F0C903/4/5
时钟输出 / 蜂鸣器输出	2 个	
	<ul style="list-style-type: none"> • 2.44kHz、4.88kHz、9.76kHz、1.25MHz、2.5MHz、5MHz、10MHz (主系统时钟: $f_{\text{MAIN}}=20\text{MHz}$ 运行时) 	
8/10 位分辨率 A/D 转换器	8 个通道	
串行接口	<ul style="list-style-type: none"> • CSI: 1 个通道 / 简易 I²C: 1 个通道 / UART: 1 个通道 • CSI: 1 个通道 / 简易 I²C: 1 个通道 / UART: 1 个通道 • CSI: 1 个通道 / 简易 I²C: 1 个通道 / UART: 1 个通道 	
I ² C 总线	1 个通道	
乘除器和乘加器	<ul style="list-style-type: none"> • 16 位 × 16 位 = 32 位 (无符号或者带符号) • 32 位 ÷ 32 位 = 32 位 (无符号) • 16 位 × 16 位 + 32 位 = 32 位 (无符号或者带符号) 	
DMA 控制器	2 个通道	
向量中断源	内部	26
	外部	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 • 通过上电复位进行的内部复位 • 通过电压检测电路进行的内部复位 • 因执行非法指令而产生的内部复位注 • 因 RAM 奇偶校验错误而产生的内部复位 • 因存取非法存储器而产生的内部复位 	
上电复位电路	<ul style="list-style-type: none"> • 上电复位: 1.51(TYP.) • 断电复位: 1.50(TYP.) 	
电压检测电路	<ul style="list-style-type: none"> • 上升沿: 1.67V ~ 4.06V (14 种) • 下降沿: 1.63V ~ 3.98V (14 种) 	
片上调试功能	有	
电源电压	$V_{\text{DD}}=1.6 \sim 5.5\text{V}$	
工作环境温度	$T_{\text{A}}=-40 \sim +85^{\circ}\text{C}$ (2D: 民用产品)	

注 当执行指令码 FFH 时, 发生非法指令错误。

在通过在线仿真器或者片上调试仿真器进行仿真时, 不会因执行非法指令而产生复位。

第 2 章 引脚功能

2.1 端口功能

各端口设定的输入 / 输出、缓冲器、上拉电阻也对复用功能有效。

2.1.1 32 引脚产品

(1/2)

功能名称	引脚类型	输入 / 输出	解除复位后	复用功能	功能
P00	7-3-2	输入 / 输出	模拟输入	ANI17/TI00/TxD1	端口 0 2 位输入 / 输出端口 能以位为单位指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。 P01 的输入能设定为 TTL 输入缓冲。 P00 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。 P00 和 P01 能设定为模拟输入注 1。
P01	8-3-1			ANI16/TO00/RxD1	
P10	8-1-2	输入 / 输出	输入端口	SCK00/SCL00/(TI07)/(TO07)	端口 1 8 位输入 / 输出端口 能以位为单位指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。 P10、P11 和 P13 ~ P17 的输入能设定为 TTL 输入缓冲。 P10 ~ P15 和 P17 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。
P11				SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-2			SO00/TxD0/TOOLTxD/(TI05)/(TO05)	
P13				TxD2/SO20/(SDAA0)/(TI04)/(TO04)	
P14	8-1-2			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)	
P16	8-1-1			TI01/TO01/INTP5/(RxD0)	
P17	8-1-2			TI02/TO02/(TxD0)	
P20	4-3-1	输入 / 输出	模拟输入	ANI0/AV _{REFP}	端口 2 4 位输入 / 输出端口 能以位为单位指定输入或者输出。 能设定为模拟输入注 2。
P21				ANI1/AV _{REFM}	
P22				ANI2	
P23				ANI3	
P30	7-1-1	输入 / 输出	输入端口	INTP3/SCK11/SCL11	端口 3 2 位输入 / 输出端口 能以位为单位指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。
P31				TI03/TO03/INTP4/PCLBUZ0	

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

2. 通过 A/D 端口配置寄存器 (ADPC) 将各引脚设定为数字或者模拟。

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。详细内容请参照“图 4-8 外围 I/O 重定向寄存器 (PIOR) 的格式”。

(2/2)

功能名称	引脚类型	输入 / 输出	解除复位后	复用功能	功能
P40	7-1-1	输入 / 输出	输入端口	TOOL0	端口 4 1 位输入 / 输出端口 能以位为单位指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。
P50	7-1-2	输入 / 输出	输入端口	INTP1/SI11/SDA11	端口 5 2 位输入 / 输出端口 能以位为单位指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。 P50 的输出能设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。
P51	7-1-1			INTP2/SO11	
P60	12-1-1	输入 / 输出	输入端口	SCLA0	端口 6 3 位输入 / 输出端口 能以位为单位指定输入或者输出。 P60 ~ P62 的输出能设定为 N 沟道漏极开路输出 (6V 耐压)。
P61				SDAA0	
P62				—	
P70	7-1-1	输入 / 输出	输入端口	—	端口 7 1 位输入 / 输出端口 能指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。
P120	7-3-1	输入 / 输出	模拟输入	ANI19	端口 12 1 位输入 / 输出端口和 2 位输入专用端口 只有 P120 才能指定输入或者输出。 只有 P120 的输入端口能通过软件的设定, 使用内部上拉电阻。 P120 能设定为模拟输入注。
P121	2-2-1	输入	输入端口	X1	
P122				X2/EXCLK	
P137	2-1-2	输入	输入端口	INTP0	端口 13 1 位输入专用端口
P147	7-3-1	输入 / 输出	模拟输入	ANI18	端口 14 1 位输入 / 输出端口 能指定输入或者输出。 输入端口能通过软件的设定, 使用内部上拉电阻。 P147 能设定为模拟输入注。
RESET	2-1-1	输入	—	—	外部复位的输入专用引脚 当不使用外部复位时, 必须直接或者通过电阻连接 V_{DD} 。

注 通过端口模式控制寄存器 (PMCxx) 将各引脚设定为数字或者模拟 (能以位为单位进行设定)。

2.2 端口以外的功能

2.2.1 各产品配置的功能

(1/2)

功能名称	32 引脚
ANI0	○
ANI1	○
ANI2	○
ANI3	○
ANI16	○
ANI17	○
ANI18	○
ANI19	○
INTP0	○
INTP1	○
INTP2	○
INTP3	○
INTP4	○
INTP5	○
PCLBUZ0	○
PCLBUZ1	○
REGC	○
$\overline{\text{RESET}}$	○
RxD0	○
RxD1	○
RxD2	○
TxD0	○
TxD1	○
TxD2	○
SCK00	○
SCK11	○
SCK20	○
SCL00	○
SCL11	○
SCL20	○
SDA00	○
SDA11	○
SDA20	○
SI00	○
SI11	○
SI20	○
SO00	○

(2/2)

功能名称	32 引脚
SO11	○
SO20	○
SCLA0	○
SDAA0	○
TI00	○
TI01	○
TI02	○
TI03	○
TI04	(○)
TI05	(○)
TI06	(○)
TI07	(○)
TO00	○
TO01	○
TO02	○
TO03	○
TO04	(○)
TO05	(○)
TO06	(○)
TO07	(○)
X1	○
X2	○
EXCLK	○
V _{DD}	○
AV _{REFP}	○
AV _{REFM}	○
V _{SS}	○
TOOLRxD	○
TOOLTxD	○
TOOL0	○

备注 上表的(○)只有在将外围 I/O 重定向寄存器 (PIOR) 的对应位置“1”后才能使用。

2.2.2 功能说明

(1/2)

功能名称	输入 / 输出	功能
ANI0 ~ ANI3、ANI16 ~ ANI19	输入	A/D 转换器的模拟输入（参照“图 10-44 模拟输入引脚的处理”）
INTP0 ~ INTP5	输入	外部中断请求输入 有效边沿的指定：上升沿、下降沿、上升和下降的双边沿
PCLBUZ0、PCLBUZ1	输出	时钟输出 / 蜂鸣器输出
REGC	—	连接用于内部工作的稳压器输出稳定电容器。 必须通过电容器（0.47 ~ 1 μ F）连接 V _{SS} 。 为了稳定内部电压，必须使用特性好的电容器。
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 ~ RxD2	输入	串行接口 UART0、UART1、UART2 的串行数据输入
TxD0 ~ TxD2	输出	串行接口 UART0、UART1、UART2 的串行数据输出
SCK00、SCK11、SCK20	输入 / 输出	串行接口 CSI00、CSI11、CSI20 的串行时钟输入 / 输出
SCL00、SCL11、SCL20	输出	串行接口 IIC00、IIC11、IIC20 的串行时钟输出
SDA00、SDA11、SDA20	输入 / 输出	串行接口 IIC00、IIC11、IIC20 的串行数据输入 / 输出
SI00、SI11、SI20	输入	串行接口 CSI00、CSI11、CSI20 的串行数据输入
SO00、SO11、SO20	输出	串行接口 CSI00、CSI11、CSI20 的串行数据输出
SCLA0	输入 / 输出	串行接口 IICA0 的串行时钟输入 / 输出
SDAA0	输入 / 输出	串行接口 IICA0 的串行数据输入 / 输出
TI00 ~ TI07	输入	16 位定时器 00 ~ 07 的外部计数时钟 / 捕捉触发输入
TO00 ~ TO07	输出	16 位定时器 00 ~ 07 的定时器输出
X1、X2	—	连接用于主系统时钟的谐振器。
EXCLK	输入	主系统时钟的外部时钟输入

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。

(2/2)

功能名称	输入 / 输出	功能
V _{DD}	—	全部引脚的正电源
AV _{REFP}	输入	A/D 转换器的正 (+) 基准电压输入
AV _{REFM}	输入	A/D 转换器的负 (-) 基准电压输入
V _{SS}	—	全部引脚的接地电位
TOOLRxD	输入	用于在闪存编程时连接外部器件的 UART 串行数据接收
TOOLTxD	输出	用于在闪存编程时连接外部器件的 UART 串行数据发送
TOOL0	输入 / 输出	用于闪存编程器 / 调试器的数据输入 / 输出

注意 解除复位时的 P40/TOOL0 和运行模式的关系如下：

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“23.4 串行编程方法”。

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

2.3 未使用引脚的处理

各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、P01	输入 / 输出	输入时：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P10 ~ P17		输出时：必须置为开路。
P20 ~ P23		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。
P30、P31		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: Port4		输入时：必须单独通过电阻连接 V_{DD} 或者置为开路。
		输出时：必须置为开路。
P50、P51		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。
P60 ~ P62	输入时：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。	输出时：必须将端口的输出锁存器置“0”并且置为开路，或者将端口的输出锁存器置“1”并且单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P70		输入时：必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P120		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} 。
P137	输入	必须单独通过电阻连接 V_{DD} 或者 V_{SS} 。
P147	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。
RESET	输入	必须直接或者通过电阻连接 V_{DD} 。
REGC	—	必须通过电容器（0.47 ~ 1 μ F）连接 V_{SS} 。

2.4 引脚框图

对于“2.1.1 32 引脚产品”中记载的引脚类型，引脚框图如图 2-1 ~ 图 2-12 所示。

图 2-1 引脚类型 2-1-1 的引脚框图

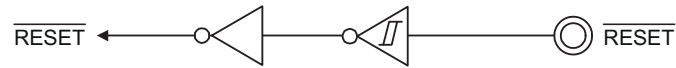
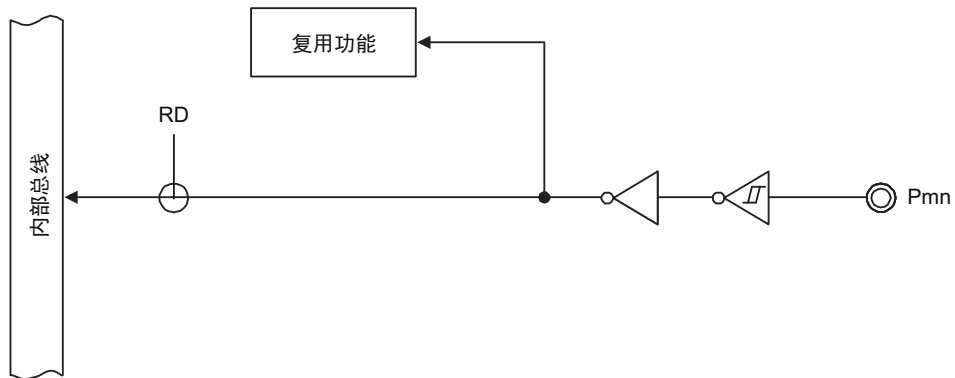
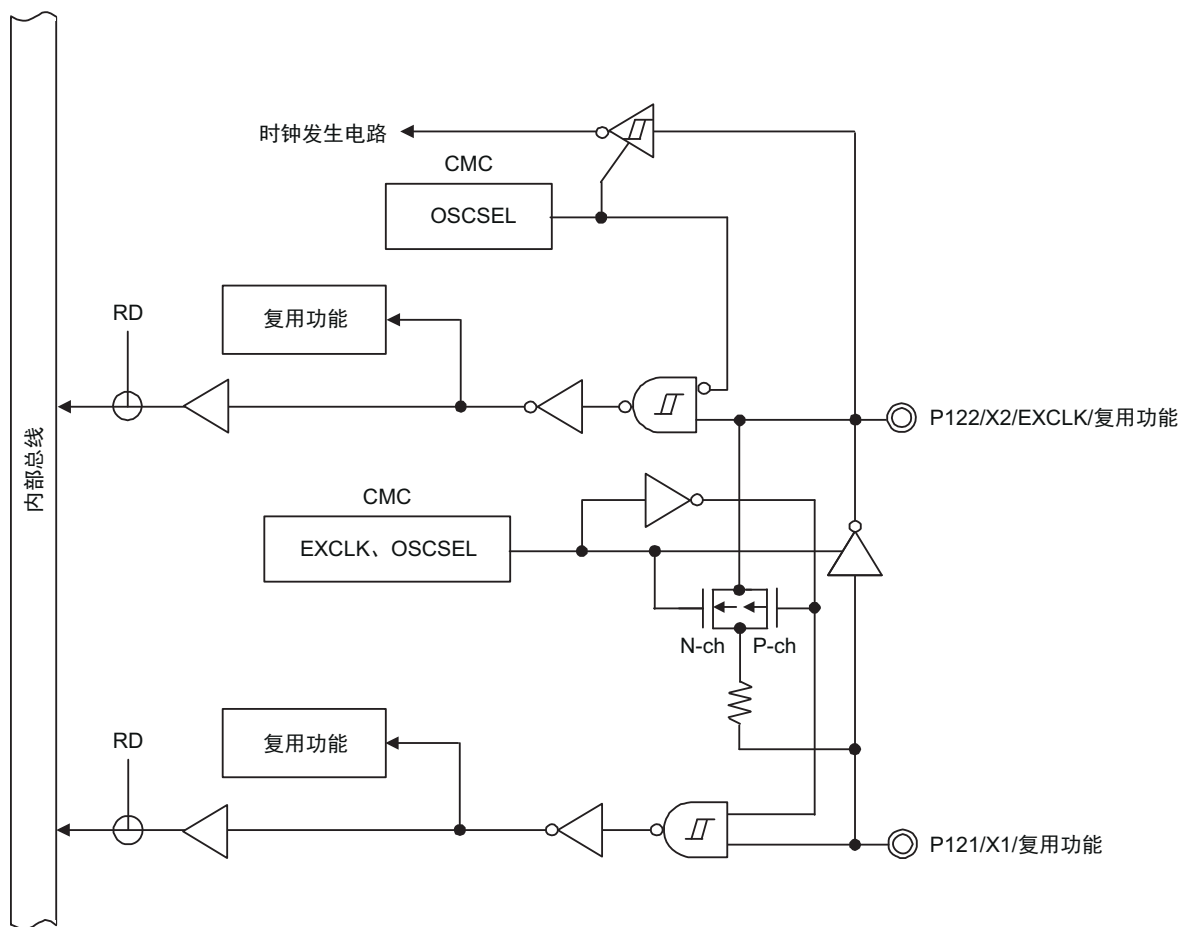


图 2-2 引脚类型 2-1-2 的引脚框图



备注 复用功能请参照“2.1 端口功能”。

图 2-3 引脚类型 2-2-1 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 4-3-1 的引脚框图

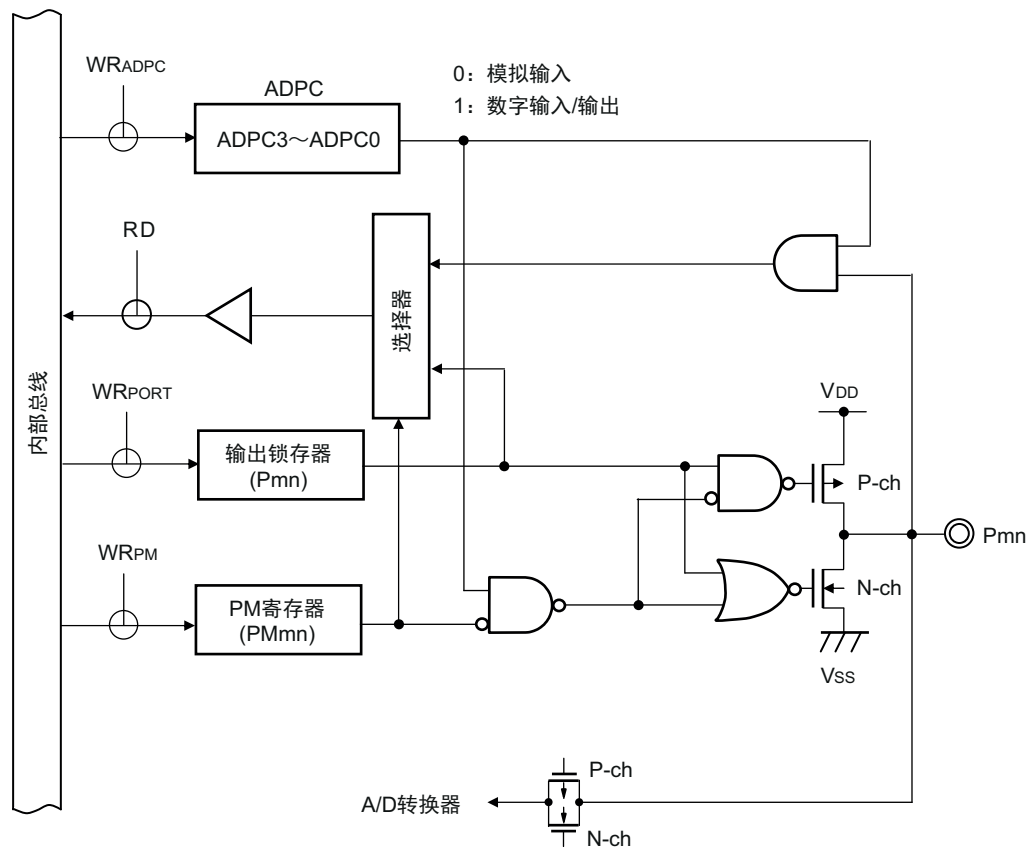
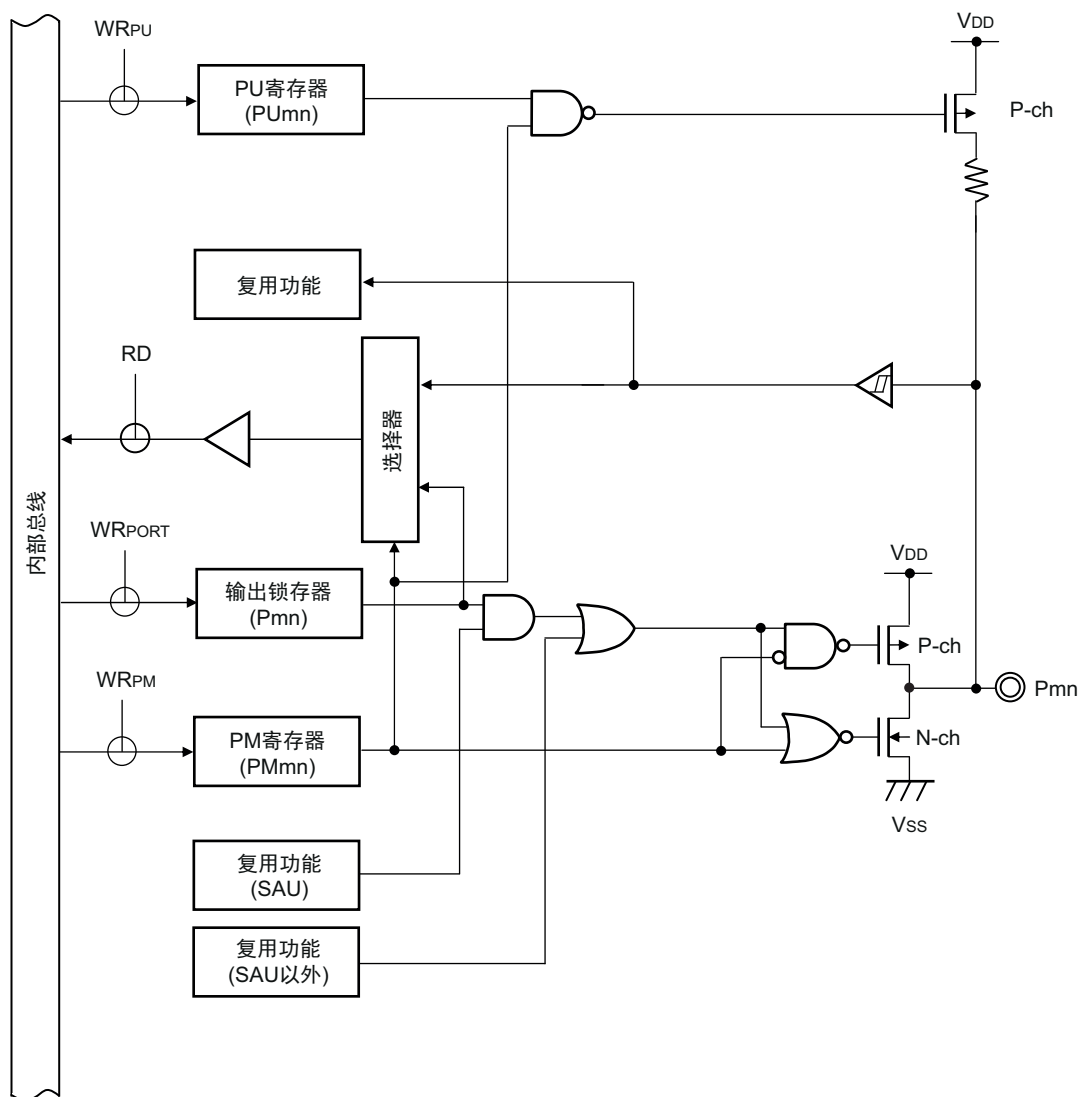


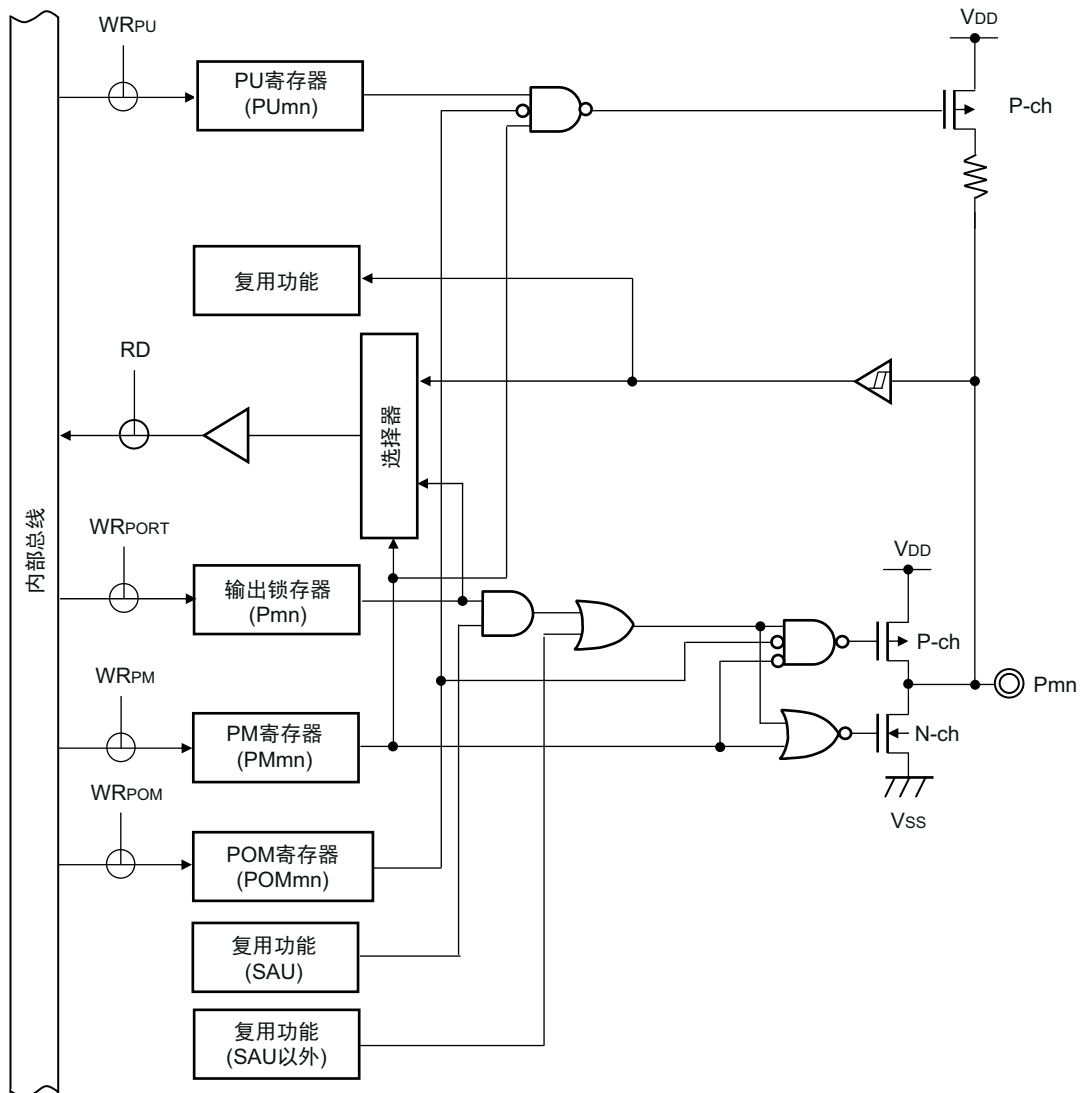
图 2-5 引脚类型 7-1-1 的引脚框图



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 2-6 引脚类型 7-1-2 的引脚框图

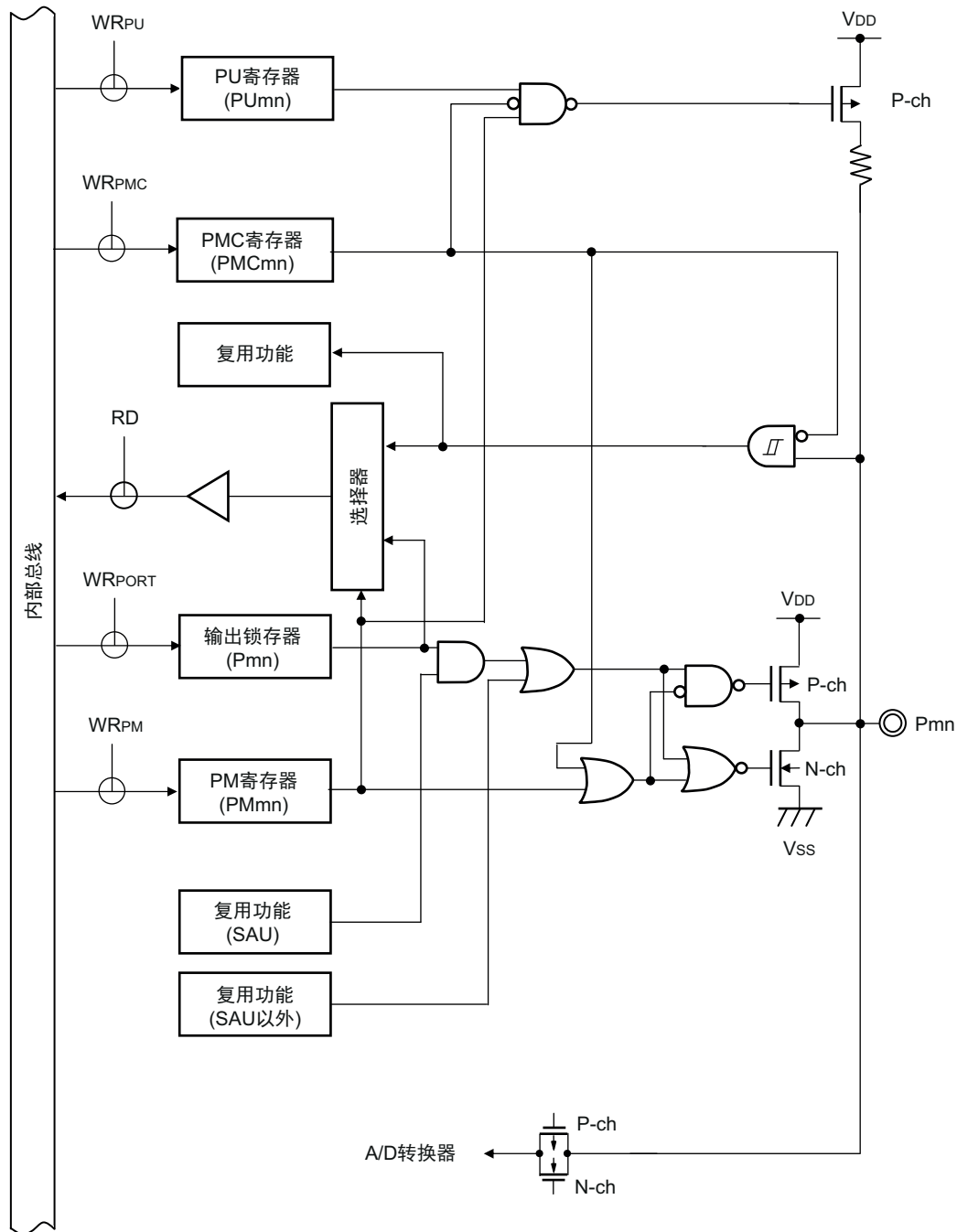


注意 当通过端口输出模式寄存器（POMxx）将引脚设定为 N 沟道漏极开路输出时，因为在输出模式中输入缓冲器处于 ON 的状态，所以如果引脚为中间电位，就可能有贯通电流流过。

备注 1. 复用功能请参照“2.1 端口功能”。

2. SAU: 串行阵列单元

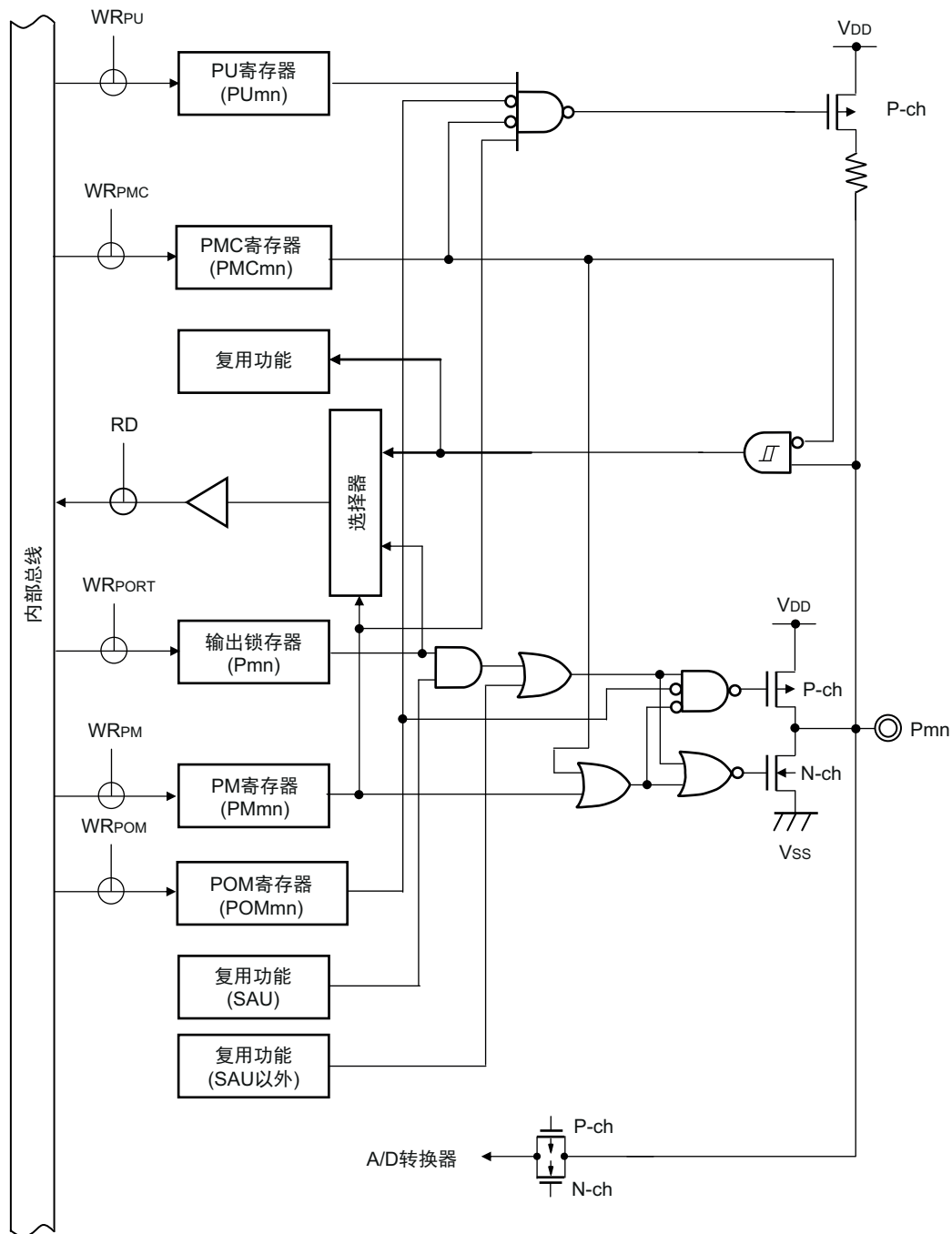
图 2-7 引脚类型 7-3-1 的引脚框图



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 2-8 引脚类型 7-3-2 的引脚框图

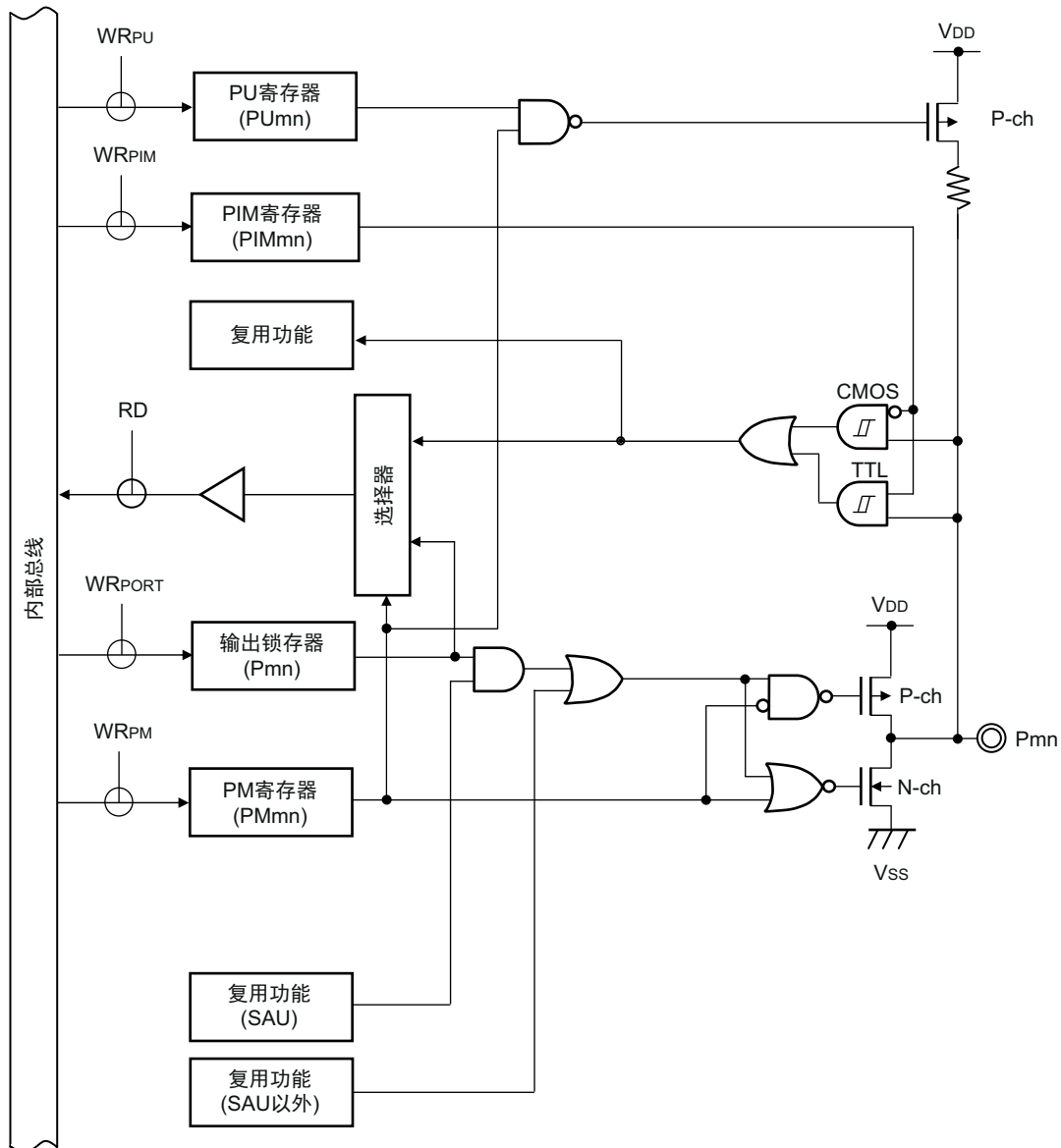


注意 当通过端口输出模式寄存器 (POMxx) 将引脚设定为 N 沟道漏极开路输出时, 因为在输出模式中输入缓冲器处于 ON 的状态, 所以如果引脚为中间电位, 就可能有贯通电流流过。

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 2-9 引脚类型 8-1-1 的引脚框图

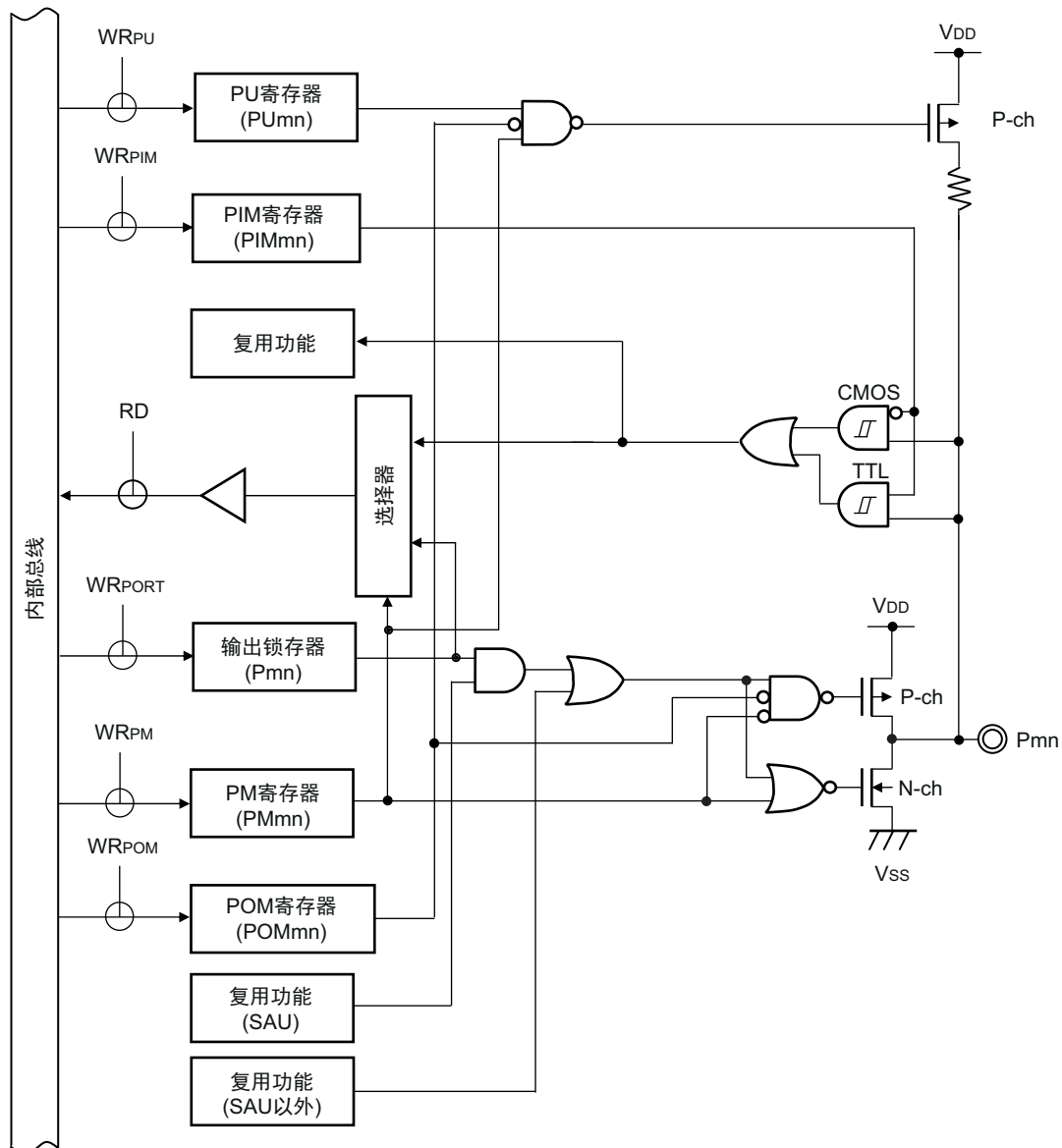


注意 如果通过端口输入模式寄存器（PIMxx）将引脚设定为 TTL 输入缓冲并且引脚有高电平输入，就可能因 TTL 输入缓冲器的结构而有贯通电流流过。为了防止待机模式时的贯通电流，必须给引脚输入低电平。

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 2-10 引脚类型 8-1-2 的引脚框图

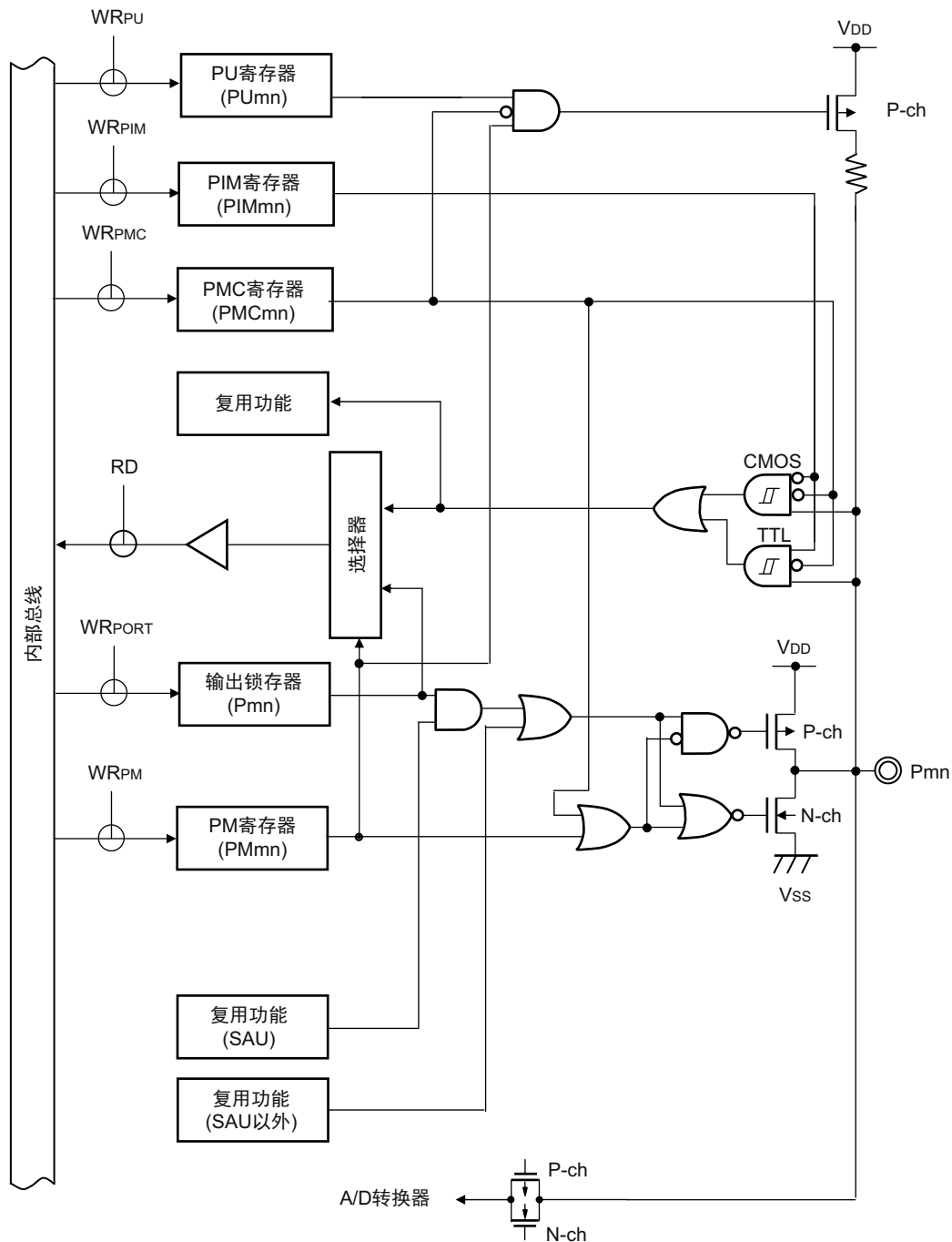


- 注意 1. 当通过端口输出模式寄存器（POMxx）将引脚设定为 N 沟道漏极开路输出时，因为在输出模式中输入缓冲器处于 ON 的状态，所以如果引脚电平为中间电平，就可能有贯通电流流过。
2. 如果通过端口输入模式寄存器（PIMxx）将引脚设定为 TTL 输入缓冲并且引脚有高电平输入，就可能因 TTL 输入缓冲器的结构而有贯通电流流过。为了防止待机模式时的贯通电流，必须给引脚输入低电平。

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 2-11 引脚类型 8-3-1 的引脚框图

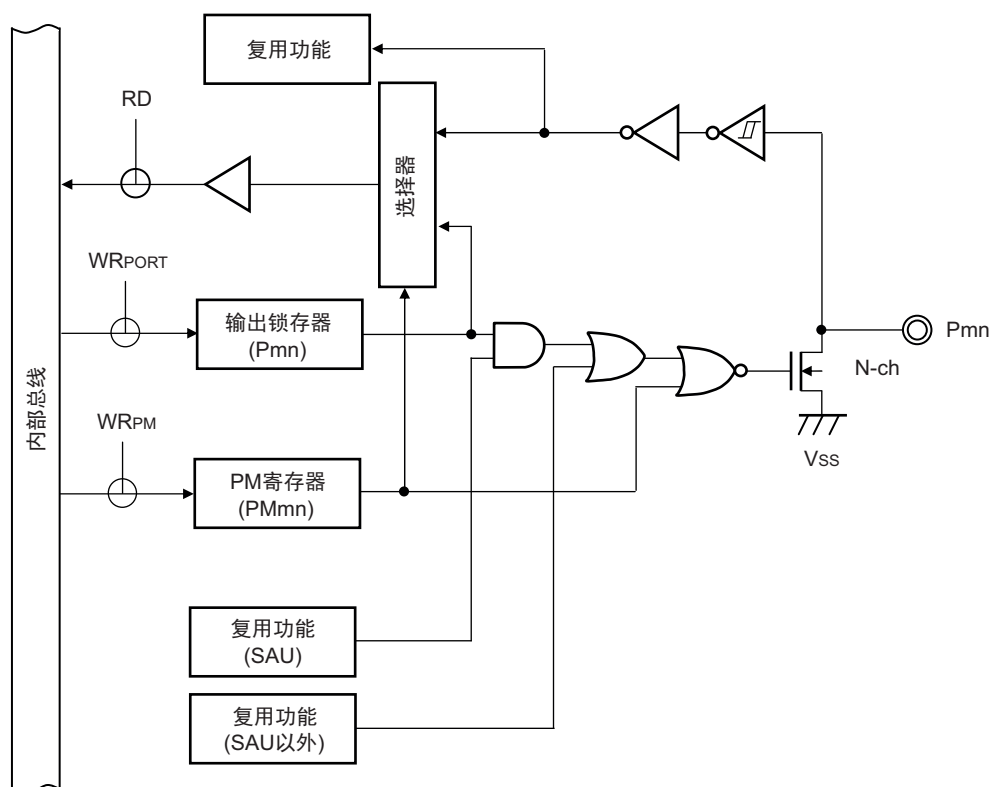


注意 如果通过端口输入模式寄存器（PIMxx）将引脚设定为 TTL 输入缓冲并且引脚有高电平输入，就可能因 TTL 输入缓冲器的结构而有贯通电流流过。为了防止待机模式时的贯通电流，必须给引脚输入低电平。

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 2-12 引脚类型 12-1-1 的引脚框图



注意 因为在输出模式中输入缓冲器处于 ON 的状态，所以如果引脚为中间电位，就可能有贯通电流流过。

备注 1. 复用功能请参照“2.1 端口功能”。

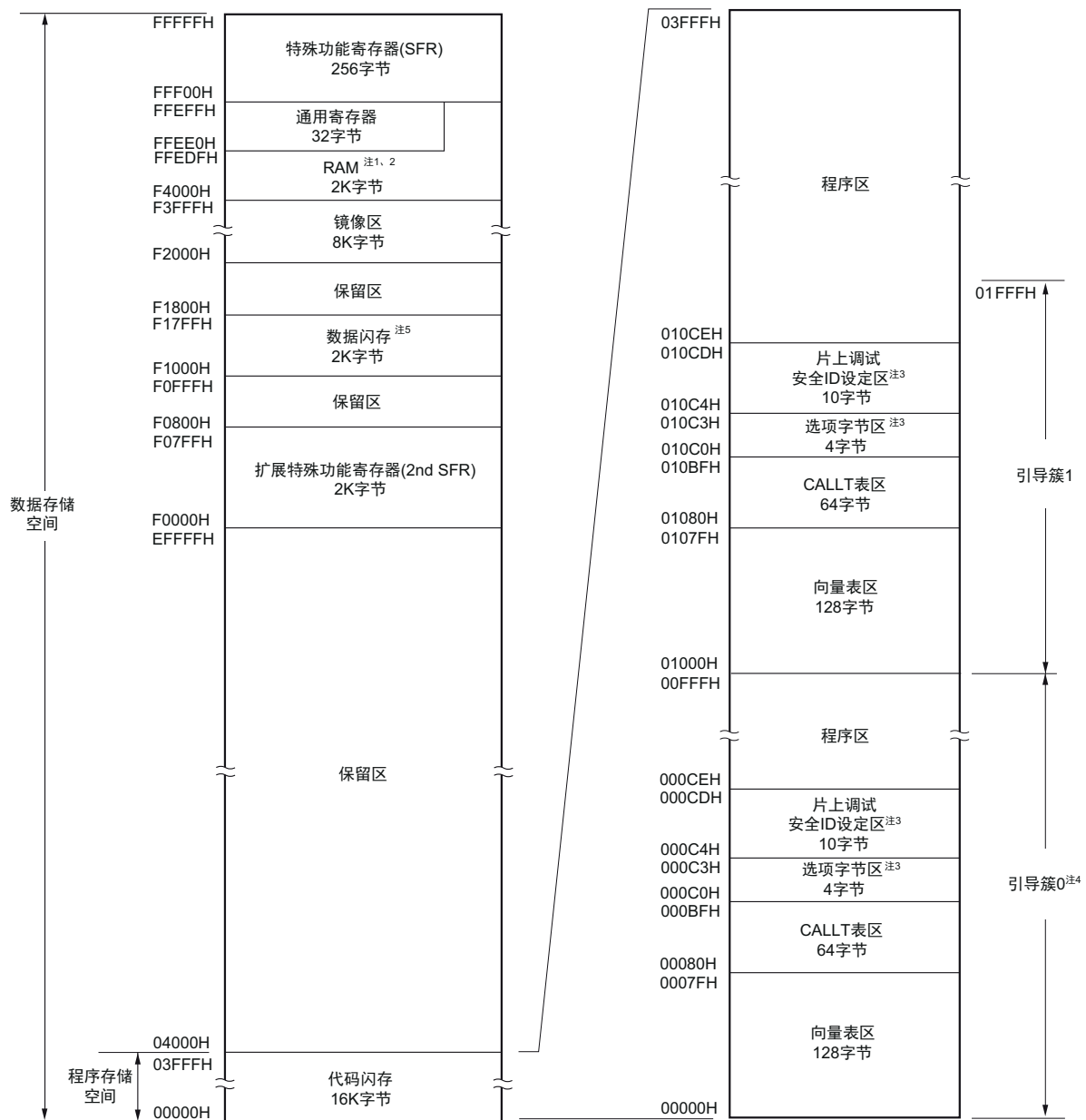
2. SAU: 串行阵列单元

第 3 章 CPU 体系结构

3.1 存储空间

R7F0C903-908 能存取 48K 字节的地址空间。存储器映像如图 3-1 ~图 3-3 所示。

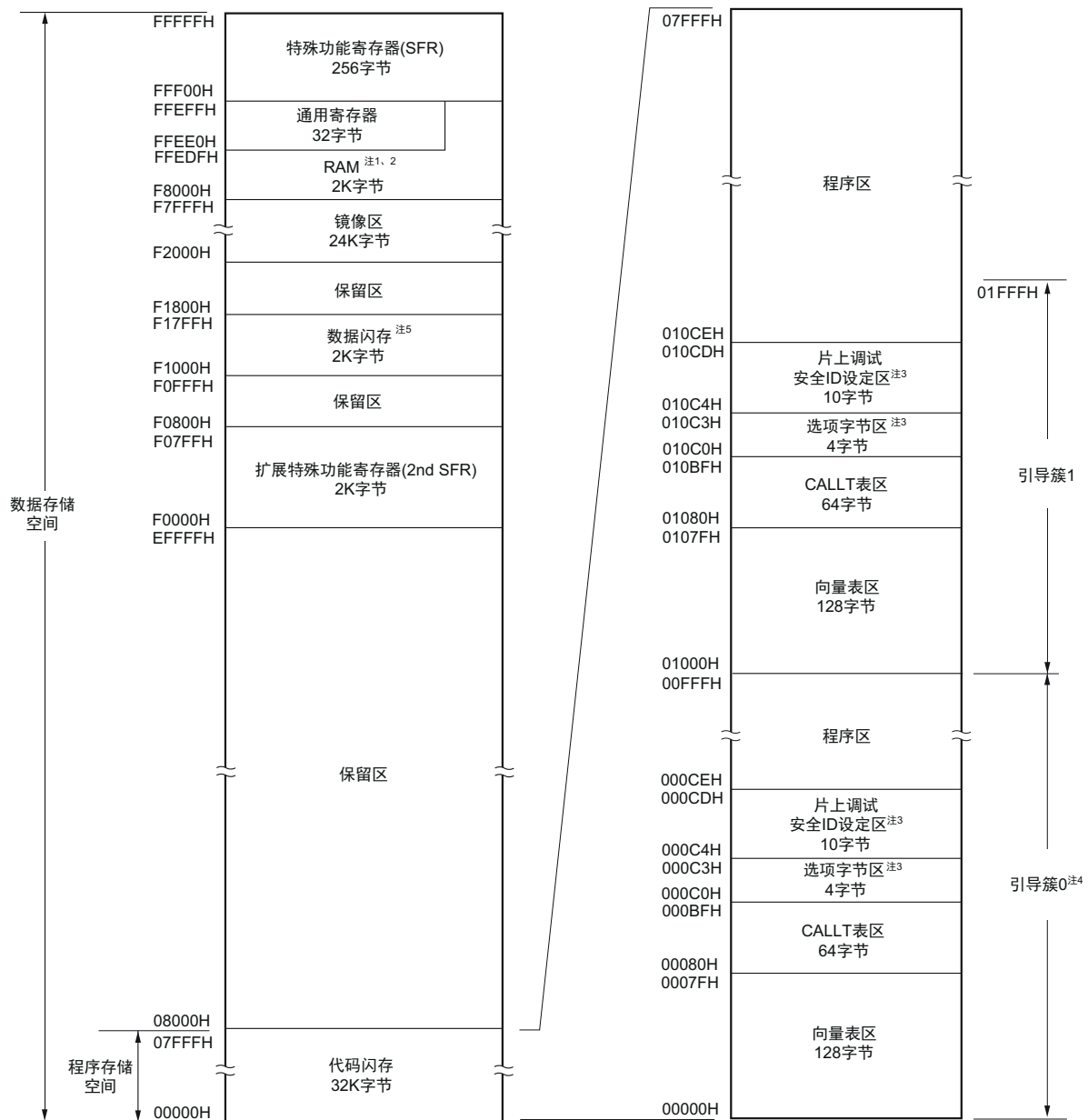
图 3-1 存储器映像 (R7F0C903、R7F0C906)



- 注
1. 在自编程以及改写数据闪存时，不能将堆栈、闪存库使用的数据缓冲器、库函数的参数、向量中断处理的转移目标和 DMA 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDFH 的区域。
 2. 能从除了通用寄存器以外的 RAM 区执行指令。
 3. 不使用引导交换功能时：给 000C0H ~ 000C3H 设定选项字节，并且给 000C4H ~ 000CDH 设定片上调试安全 ID。
使用引导交换功能时：给 000C0H ~ 000C3H 和 010C0H ~ 010C3H 设定选项字节，并且给 000C4H ~ 000CDH 和 010C4H ~ 010CDH 设定片上调试安全 ID。
 4. 能通过安全功能的设定来禁止改写引导簇 0（参照“23.7 安全设定”）。
 5. 在 R7F0C903 产品中为保留区。

注意 在允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区 +10 字节”的区域进行初始化。通过产生复位，进入允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 的状态。详细内容请参照“20.3.3 RAM 奇偶校验错误检测功能”。

图 3-2 存储器映像 (R7F0C904、R7F0C907)



- 注 1. 在自编程以及改写数据闪存时，不能将堆栈、闪存库使用的数据缓冲器、库函数的参数、向量中断处理的转移目标和 DMA 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDFH 的区域。
2. 能从除了通用寄存器以外的 RAM 区执行指令。
3. 不使用引导交换功能时：给 000C0H ~ 000C3H 设定选项字节，并且给 000C4H ~ 000CDH 设定片上调试安全 ID。
使用引导交换功能时：给 000C0H ~ 000C3H 和 010C0H ~ 010C3H 设定选项字节，并且给 000C4H ~ 000CDH 和 010C4H ~ 010CDH 设定片上调试安全 ID。
4. 能通过安全功能的设定来禁止改写引导簇 0（参照“23.7 安全设定”）。
5. 在 R7F0C904 产品中为保留区。

注意 在允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区 +10 字节”的区域进行初始化。通过产生复位，进入允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的状态。详细内容请参照“20.3.3 RAM 奇偶校验错误检测功能”。

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



（R7F0C906、R7F0C907、R7F0C908 的情况）

闪存的地址值和块号的对应如下所示。

表 3-1 闪存的地址值和块号的对应

地址值	块号	地址值	块号
00000H ~ 003FFH	00H	06000H ~ 063FFH	18H
00400H ~ 007FFH	01H	06400H ~ 067FFH	19H
00800H ~ 00BFFH	02H	06800H ~ 06BFFH	1AH
00C00H ~ 00FFFH	03H	06C00H ~ 06FFFH	1BH
01000H ~ 013FFH	04H	07000H ~ 073FFH	1CH
01400H ~ 017FFH	05H	07400H ~ 077FFH	1DH
01800H ~ 01BFFH	06H	07800H ~ 07BFFH	1EH
01C00H ~ 01FFFH	07H	07C00H ~ 07FFFH	1FH
02000H ~ 023FFH	08H	08000H ~ 083FFH	20H
02400H ~ 027FFH	09H	08400H ~ 087FFH	21H
02800H ~ 02BFFH	0AH	08800H ~ 08BFFH	22H
02C00H ~ 02FFFH	0BH	08C00H ~ 08FFFH	23H
03000H ~ 033FFH	0CH	09000H ~ 093FFH	24H
03400H ~ 037FFH	0DH	09400H ~ 097FFH	25H
03800H ~ 03BFFH	0EH	09800H ~ 09BFFH	26H
03C00H ~ 03FFFH	0FH	09C00H ~ 09FFFH	27H
04000H ~ 043FFH	10H	0A000H ~ 0A3FFH	28H
04400H ~ 047FFH	11H	0A400H ~ 0A7FFH	29H
04800H ~ 04BFFH	12H	0A800H ~ 0ABFFH	2AH
04C00H ~ 04FFFH	13H	0AC00H ~ 0AFFFH	2BH
05000H ~ 053FFH	14H	0B000H ~ 0B3FFH	2CH
05400H ~ 057FFH	15H	0B400H ~ 0B7FFH	2DH
05800H ~ 05BFFH	16H	0B800H ~ 0BBFFH	2EH
05C00H ~ 05FFFH	17H	0BC00H ~ 0BFFFH	2FH

备注 R7F0C903、R7F0C906：块号 00H ~ 0FH
 R7F0C904、R7F0C907：块号 00H ~ 1FH
 R7F0C905、R7F0C908：块号 00H ~ 2FH

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，本产品内置的 ROM（闪存）如下所示。

表 3-2 内部 ROM 容量

产品	内部 ROM	
	构造	容量
R7F0C903、R7F0C906	闪存	16384×8 位 (00000H ~ 03FFFFH)
R7F0C904、R7F0C907		32768×8 位 (00000H ~ 07FFFFH)
R7F0C905、R7F0C908		49152×8 位 (00000H ~ 0BFFFFH)

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。

在使用引导交换功能时，还必须给 01000H ~ 0107FH 设定向量表。

表 3-3 向量表

向量表地址	中断源	本产品
00000H	RESET、POR、LVD、WDT、TRAP、IAW、RPE	○
00004H	INTWDTI	○
00006H	INTLVI	○
00008H	INTP0	○
0000AH	INTP1	○
0000CH	INTP2	○
0000EH	INTP3	○
00010H	INTP4	○
00012H	INTP5	○
00014H	INTST2/INTCSI20/INTIIC20	○
00016H	INTSR2	○
00018H	INTSRE2	○
0001AH	INTDMA0	○
0001CH	INTDMA1	○
0001EH	INTST0/INTCSI00/INTIIC00	○
00020H	INTSR0	○
00022H	INTSRE0	○
	INTTM01H	○
00024H	INTST1	○
00026H	INTSR1/INTCSI11/INTIIC11	○
00028H	INTSRE1	○
	INTTM03H	○
0002AH	INTIICA0	○
0002CH	INTTM00	○
0002EH	INTTM01	○
00030H	INTTM02	○
00032H	INTTM03	○
00034H	INTAD	○
00038H	INTIT	○
00042H	INTTM04	○
00044H	INTTM05	○
00046H	INTTM06	○
00048H	INTTM07	○
0005EH	INTMD	○
00062H	INTFL	○
0007EH	BRK	○

(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)。

在使用引导交换功能时, 还必须给 01080H ~ 010BFH 设定 CALLT 指令表。

(3) 选项字节区

000C0H ~ 000C3H 的 4 字节区域用作选项字节区。在使用引导交换功能时, 还必须给 010C0H ~ 010C3H 设定选项字节。详细内容请参照“第 22 章 选项字节”。

(4) 片上调试安全 ID 设定区

000C4H ~ 000CDH 和 010C4H ~ 010CDH 的 10 字节区域用作片上调试安全 ID 设定区。当不使用引导交换功能时, 必须给 000C4H ~ 000CDH 设定 10 字节的片上调试安全 ID; 当使用引导交换功能时, 必须给 000C4H ~ 000CDH 和 010C4H ~ 010CDH 设定 10 字节的片上调试安全 ID。详细内容请参照“第 24 章 片上调试功能”。

3.1.2 镜像区

本产品将 00000H ~ 0FFFFH 的代码闪存区镜像到 F0000H ~ FFFFFH。

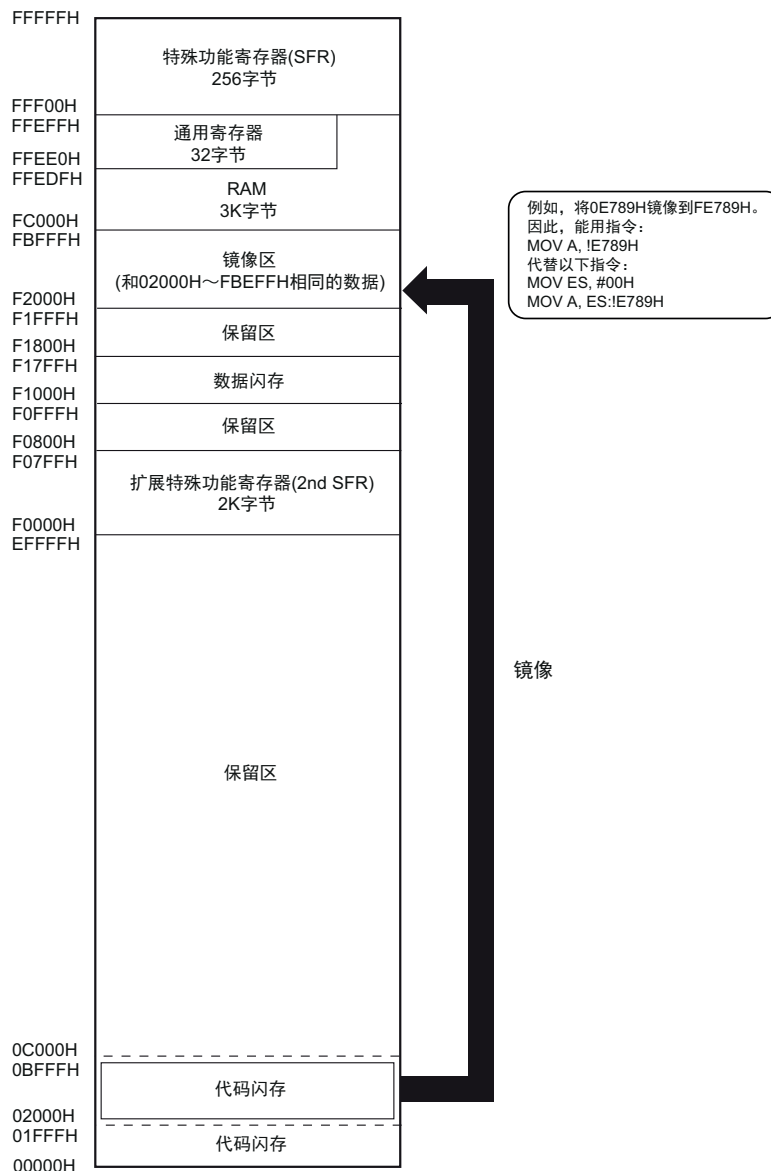
能通过从镜像目标的 F0000H ~ FFFFFH 读数据来使用操作数中不持有 ES 寄存器的指令，因此能用短代码读代码闪存的内容。但是，不能将代码闪存区镜像到 SFR、扩展 SFR、RAM 区以及保留区。

有关各产品的镜像区，请参照“3.1 存储空间”。

镜像区为只读区，不能从此区域取指令。

例子如下所示。

例 R7F0C908（闪存为 48K 字节，RAM 为 3K 字节）的情况



3.1.3 内部数据存储空间

本产品内置以下 RAM。

表 3-4 内部 RAM 容量

产品	内部 RAM
R7F0C903、R7F0C906	2048×8 位 (FF700H ~ FFEFFH)
R7F0C904、R7F0C907	
R7F0C905、R7F0C908	3072×8 位 (FF300H ~ FFEFFH)

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

另外，堆栈存储器使用内部 RAM。

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

2. 在自编程以及改写数据闪存时，不能将堆栈、闪存库使用的数据缓冲器、库函数的参数、向量中断处理的转移目标和 DMA 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEDFH 的区域。

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

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

注意 不能存取未分配 SFR 的地址。

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

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

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

注意 不能存取未分配扩展 SFR 的地址。

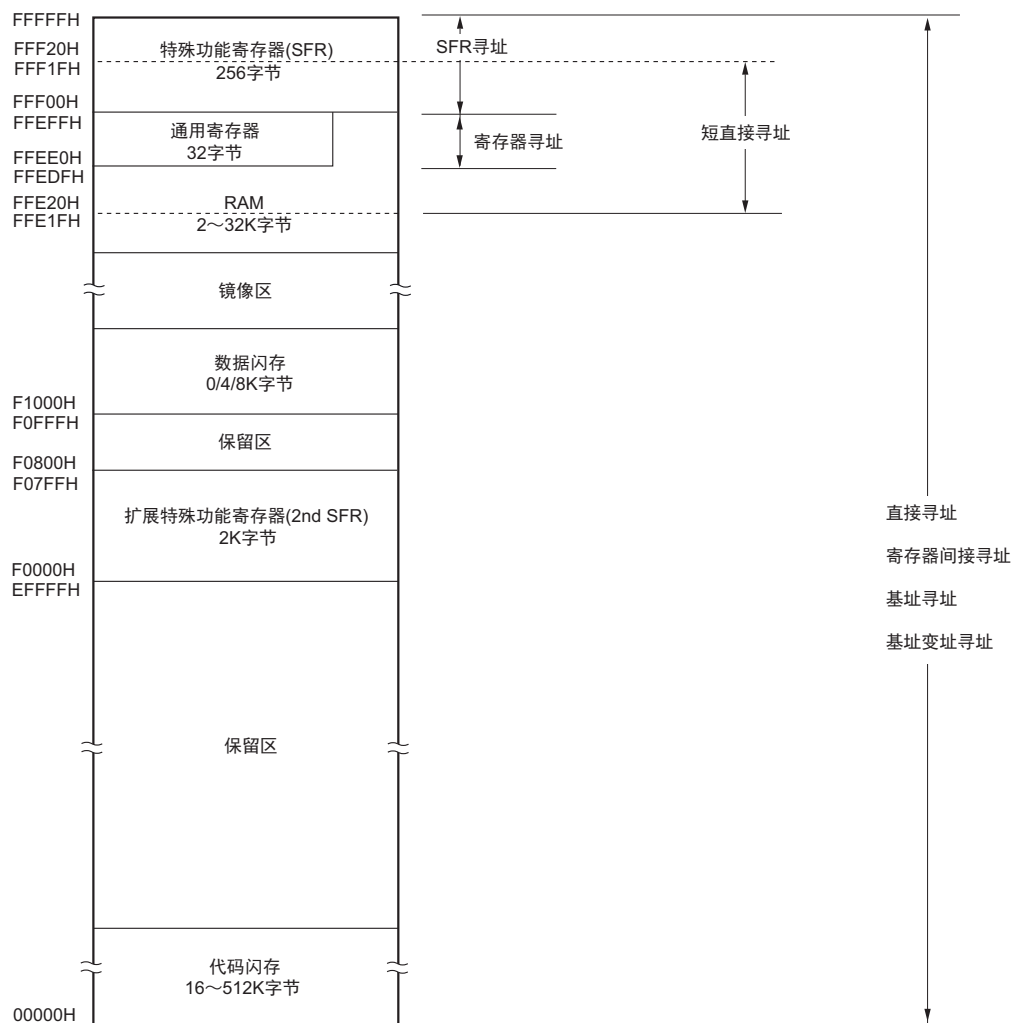
3.1.6 数据存储器的寻址

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

对于指令执行操作对象的存储器的寻址，考虑到可操作性等，本产品提供了丰富的寻址方式。尤其是能根据特殊功能寄存器（SFR）和通用寄存器等各种功能进行特殊的寻址。数据存储器和寻址的对应如图 3-4 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

本产品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）。

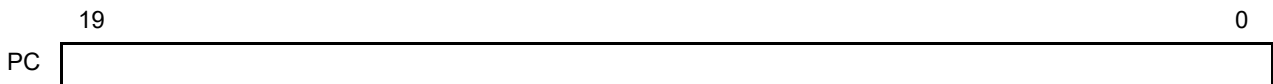
(1) 程序计数器（PC）

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 00000H 和 00001H 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



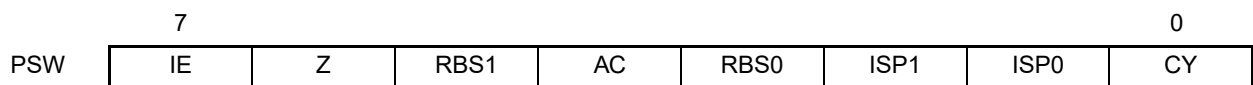
(2) 程序状态字（PSW）

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。

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

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

(b) 零标志（Z）

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

(c) 寄存器组选择标志 (RBS0、RBS1)

这是从 4 组寄存器中选择 1 组的 2 位标志。
此标志保存通过执行 SEL RBn 指令所选寄存器组的 2 位信息。

(d) 辅助进位标志 (AC)

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

(e) 优先级控制标志 (ISP1、ISP0)

这是管理能接受的可屏蔽向量中断优先级的标志。禁止接受优先级指定标志寄存器 (PRn0L、PRn0H、PRn1L、PRn1H、PRn2L、PRn2H、PRn3L) (参照 15.3.3) 指定的低于 ISP0 标志值和 ISP1 标志值的向量中断请求。另外，实际上根据中断允许标志 (IE) 的状态控制是否接受向量中断请求。

备注 n=0、1

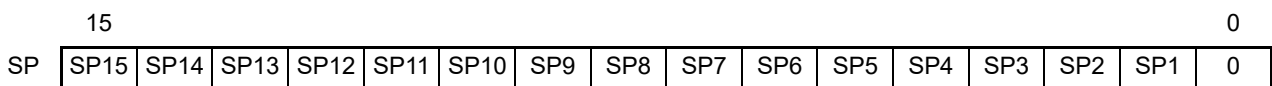
(f) 进位标志 (CY)

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

(3) 堆栈指针 (SP)

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

图 3-7 堆栈指针的结构



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

- 注意
1. 在产生复位信号后，SP 的内容变为不定值，因此必须在使用堆栈前对 SP 进行初始化。
 2. 不能将分配通用寄存器 (FFEE0H ~ FFEFFH) 的空间用于取指令和堆栈区。
 3. 在自编程以及改写数据闪存时，不能将堆栈、闪存库使用的数据缓冲器、库函数的参数、向量中断处理的转移目标和 DMA 传送目标 / 传送源使用的 RAM 地址分配到 FFE20H ~ FFEFDH 的区域。

3.2.2 通用寄存器

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

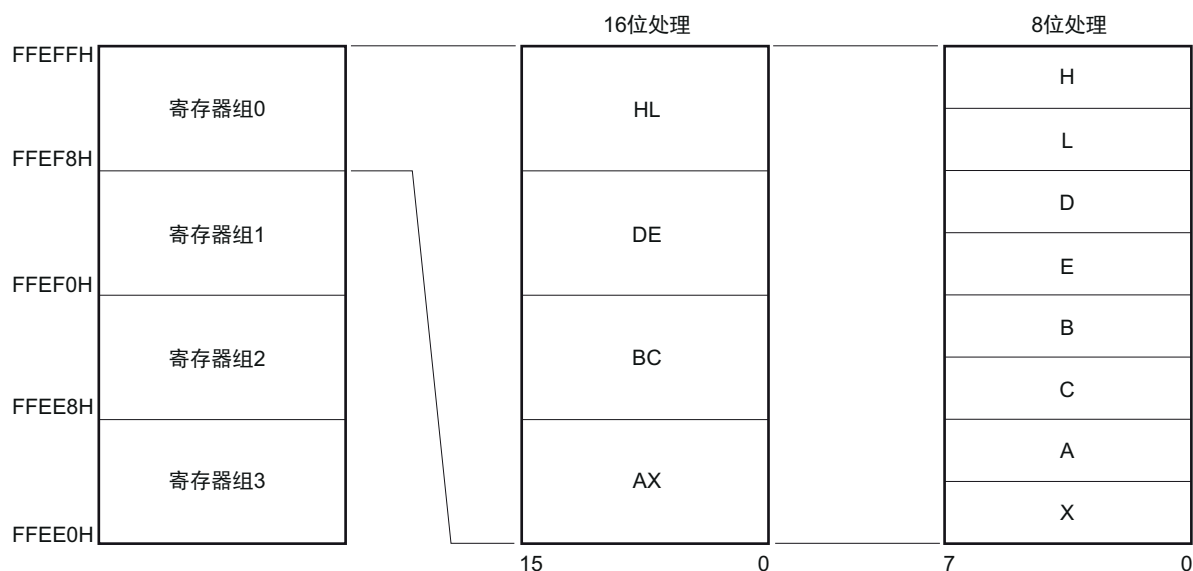
各寄存器除了能分别用作 8 位寄存器以外，还能将 2 个 8 位寄存器成对用作 1 个 16 位寄存器（AX、BC、DE、HL）。

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

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

图 3-8 通用寄存器的结构

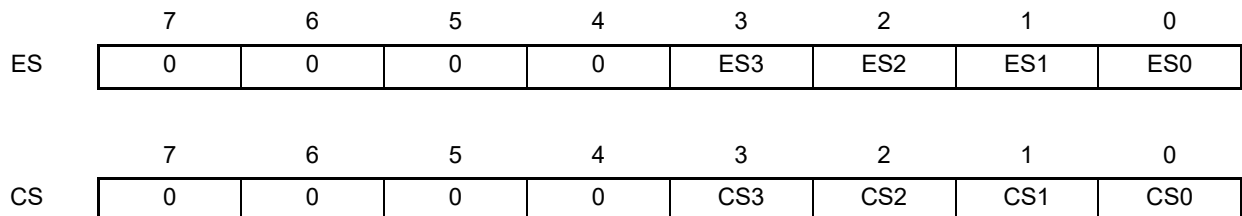
(a) 功能名称



3.2.3 ES 寄存器和 CS 寄存器

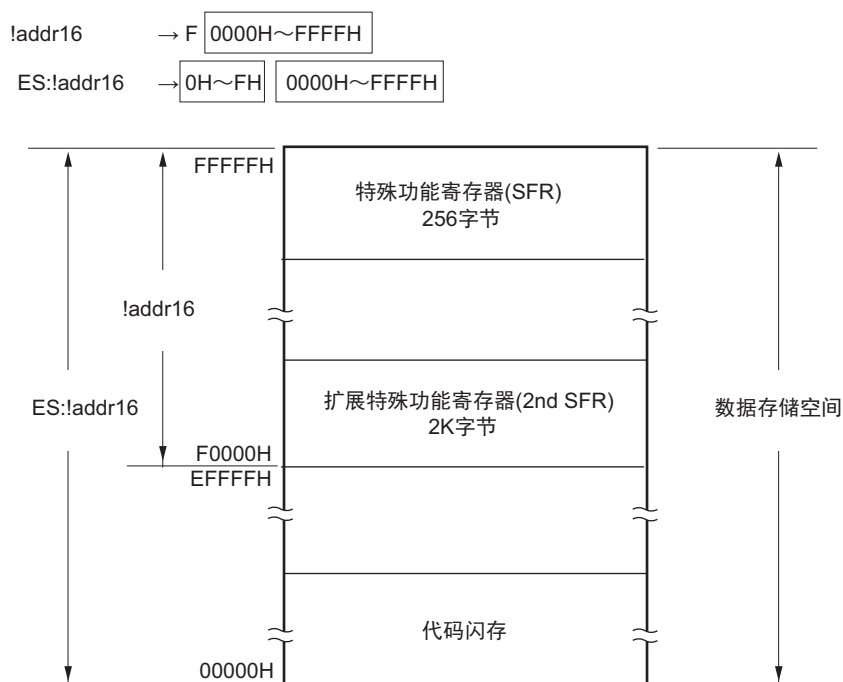
能通过 ES 寄存器和 CS 寄存器（寄存器直接寻址）分别指定存取数据和执行转移指令时的高位地址。ES 寄存器的复位后的初始值为“0FH”，CS 寄存器的复位后的初始值为“00H”。

图 3-9 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。

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

各操作位单位的指定方法如下所示。

- 1 位操作
给 1 位操作指令的操作数（sfr.bit）进行以下的记述：
已定义位名的情况：<位名>
没有定义位名的情况：<寄存器名>.<位号>或者<地址>.<位号>
- 8 位操作
给 8 位操作指令的操作数（sfr）记述汇编程序定义的符号，也能指定地址。
- 16 位操作
给 16 位操作指令的操作数（sfrp）记述汇编程序定义的符号。当指定地址时，必须记述偶数地址。

SFR 一览表如表 3-5 所示。表中的项目的含义如下所示。

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

注意 不能存取未分配 SFR 的地址。

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

表 3-5 SFR 一览表 (1/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
FFF00H	端口寄存器 0	P0		R/W	○	○	—	00H
FFF01H	端口寄存器 1	P1		R/W	○	○	—	00H
FFF02H	端口寄存器 2	P2		R/W	○	○	—	00H
FFF03H	端口寄存器 3	P3		R/W	○	○	—	00H
FFF04H	端口寄存器 4	P4		R/W	○	○	—	00H
FFF05H	端口寄存器 5	P5		R/W	○	○	—	00H
FFF06H	端口寄存器 6	P6		R/W	○	○	—	00H
FFF07H	端口寄存器 7	P7		R/W	○	○	—	00H
FFF0CH	端口寄存器 12	P12		R/W	○	○	—	不定值
FFF0DH	端口寄存器 13	P13		R/W	○	○	—	不定值
FFF0EH	端口寄存器 14	P14		R/W	○	○	—	00H
FFF10H	串行数据寄存器 00	TXD0/ SIO00	SDR00	R/W	—	○	○	0000H
FFF11H		—			—			
FFF12H	串行数据寄存器 01	RXD0/ SIO01	SDR01	R/W	—	○	○	0000H
FFF13H		—			—			
FFF18H	定时器数据寄存器 00	TDR00		R/W	—	—	○	0000H
FFF19H								
FFF1AH	定时器数据寄存器 01	TDR01L	TDR01	R/W	—	○	○	00H
FFF1BH		TDR01H			—	○		00H
FFF1EH	10 位 A/D 转换结果寄存器	ADCR		R	—	—	○	0000H
FFF1FH	8 位 A/D 转换结果寄存器	ADCRH		R	—	○	—	00H
FFF20H	端口模式寄存器 0	PM0		R/W	○	○	—	FFH
FFF21H	端口模式寄存器 1	PM1		R/W	○	○	—	FFH
FFF22H	端口模式寄存器 2	PM2		R/W	○	○	—	FFH
FFF23H	端口模式寄存器 3	PM3		R/W	○	○	—	FFH
FFF24H	端口模式寄存器 4	PM4		R/W	○	○	—	FFH
FFF25H	端口模式寄存器 5	PM5		R/W	○	○	—	FFH
FFF26H	端口模式寄存器 6	PM6		R/W	○	○	—	FFH
FFF27H	端口模式寄存器 7	PM7		R/W	○	○	—	FFH

表 3-5 SFR 一览表 (2/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
FFF2CH	端口模式寄存器 12	PM12		R/W	○	○	—	FFH
FFF2EH	端口模式寄存器 14	PM14		R/W	○	○	—	FFH
FFF30H	A/D 转换器的模式寄存器 0	ADM0		R/W	○	○	—	00H
FFF31H	模拟输入通道指定寄存器	ADS		R/W	○	○	—	00H
FFF32H	A/D 转换器的模式寄存器 1	ADM1		R/W	○	○	—	00H
FFF38H	外部中断上升沿允许寄存器 0	EGP0		R/W	○	○	—	00H
FFF39H	外部中断下降沿允许寄存器 0	EGN0		R/W	○	○	—	00H
FFF44H	串行数据寄存器 02	TXD1/ SIO10	SDR02	R/W	—	○	○	0000H
FFF45H		—			—	—		
FFF46H	串行数据寄存器 03	RXD1/ SIO11	SDR03	R/W	—	○	○	0000H
FFF47H		—			—	—		
FFF48H	串行数据寄存器 10	TXD2/ SIO20	SDR10	R/W	—	○	○	0000H
FFF49H		—			—	—		
FFF4AH	串行数据寄存器 11	RXD2/ SIO21	SDR11	R/W	—	○	○	0000H
FFF4BH		—			—	—		
FFF50H	IICA 移位寄存器 0	IICA0		R/W	—	○	—	00H
FFF51H	IICA 状态寄存器 0	IICS0		R	○	○	—	00H
FFF52H	IICA 标志寄存器 0	IICF0		R/W	○	○	—	00H
FFF64H	定时器数据寄存器 02	TDR02		R/W	—	—	○	0000H
FFF65H					—	—	—	
FFF66H	定时器数据寄存器 03	TDR03L	TDR03	R/W	—	○	○	00H
FFF67H		TDR03H			—	○	00H	
FFF68H	定时器数据寄存器 04	TDR04		R/W	—	—	○	0000H
FFF69H					—	—	—	
FFF6AH	定时器数据寄存器 05	TDR05		R/W	—	—	○	0000H
FFF6BH					—	—	—	
FFF6CH	定时器数据寄存器 06	TDR06		R/W	—	—	○	0000H
FFF6DH					—	—	—	
FFF6EH	定时器数据寄存器 07	TDR07		R/W	—	—	○	0000H
FFF6FH					—	—	—	
FFF90H	间隔定时器的控制寄存器	ITMC		R/W	—	—	○	0FFFH
FFF91H					—	—	—	
FFFA0H	时钟运行模式控制寄存器	CMC		R/W	—	○	—	00H
FFFA1H	时钟运行状态控制寄存器	CSC		R/W	○	○	—	C0H
FFFA2H	振荡稳定时间计数器的状态寄存器	OSTC		R	○	○	—	00H
FFFA3H	振荡稳定时间选择寄存器	OSTS		R/W	—	○	—	07H
FFFA4H	系统时钟控制寄存器	CKC		R/W	○	○	—	00H

表 3-5 SFR 一览表 (3/4)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位的范围			复位后	
				1 位	8 位	16 位		
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	○	○	—	00H	
FFFA6H	时钟输出选择寄存器 1	CKS1	R/W	○	○	—	00H	
FFFA8H	复位控制标志寄存器	RESF	R	—	○	—	不定值 ^{注1}	
FFFA9H	电压检测寄存器	LVIM	R/W	○	○	—	00H ^{注1}	
FFFAAH	电压检测电平寄存器	LVIS	R/W	○	○	—	00H/01H/ 81H ^{注1}	
FFFABH	看门狗定时器允许寄存器	WDTE	R/W	—	○	—	1AH/9AH ^{注2}	
FFFACH	CRC 输入寄存器	CRCIN	R/W	—	○	—	00H	
FFF0H	DMA SFR 地址寄存器 0	DSA0	R/W	—	○	—	00H	
FFF1H	DMA SFR 地址寄存器 1	DSA1	R/W	—	○	—	00H	
FFF2H	DMA RAM 地址寄存器 0	DRA0L	DRA0	R/W	—	○	○	00H
FFF3H		DRA0H		R/W	—	○	○	00H
FFF4H	DMA RAM 地址寄存器 1	DRA1L	DRA1	R/W	—	○	○	00H
FFF5H		DRA1H		R/W	—	○	○	00H
FFF6H	DMA 字节计数寄存器 0	DBC0L	DBC0	R/W	—	○	○	00H
FFF7H		DBC0H		R/W	—	○	○	00H
FFF8H	DMA 字节计数寄存器 1	DBC1L	DBC1	R/W	—	○	○	00H
FFF9H		DBC1H		R/W	—	○	○	00H
FFFB0H	DMA 模式控制寄存器 0	DMC0	R/W	○	○	—	00H	
FFFB1H	DMA 模式控制寄存器 1	DMC1	R/W	○	○	—	00H	
FFFB2H	DMA 运行控制寄存器 0	DRC0	R/W	○	○	—	00H	
FFFB3H	DMA 运行控制寄存器 1	DRC1	R/W	○	○	—	00H	
FFFD0H	中断请求标志寄存器 2H	IF2H	R/W	○	○	○	00H	
FFFD1H	中断屏蔽标志寄存器 2H	MK2H	R/W	○	○	○	FFH	

注 1. 寄存器的复位值因复位源而不同, 如下所示。

复位源		RESET 输入	POR 产生的复位	执行非法指令产生的复位	WDT 产生的复位	RAM 奇偶校验错误产生的复位	存取非法存储器产生的复位	LVD 产生的复位		
RESF	TRAP	清“0”		置“1”	保持			保持		
	WDTRF			保持	置“1”				保持	
	RPERF			保持					置“1”	保持
	IARWF			保持						置“1”
	LVIRF			保持						
LVIM	LVISEN	清“0”						保持		
	LVIOMSK								保持	
	LVIF									
LVIS		清除 (00H/01H/81H)								

2. WDTE 寄存器的复位值取决于选项字节的设定。

表 3-5 SFR 一览表 (4/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
FFFD9H	优先级指定标志寄存器 02H	PR02H		R/W	○	○	○	FFH
FFFDDH	优先级指定标志寄存器 12H	PR12H		R/W	○	○	○	FFH
FFFE0H	中断请求标志寄存器 0	IF0L	IF0	R/W	○	○	○	00H
FFFE1H		IF0H		R/W	○	○		00H
FFFE2H	中断请求标志寄存器 1	IF1L	IF1	R/W	○	○	○	00H
FFFE3H		IF1H		R/W	○	○		00H
FFFE4H	中断屏蔽标志寄存器 0	MK0L	MK0	R/W	○	○	○	FFH
FFFE5H		MK0H		R/W	○	○		FFH
FFFE6H	中断屏蔽标志寄存器 1	MK1L	MK1	R/W	○	○	○	FFH
FFFE7H		MK1H		R/W	○	○		FFH
FFFE8H	优先级指定标志寄存器 00	PR00L	PR00	R/W	○	○	○	FFH
FFFE9H		PR00H		R/W	○	○		FFH
FFFEAH	优先级指定标志寄存器 01	PR01L	PR01	R/W	○	○	○	FFH
FFFEBH		PR01H		R/W	○	○		FFH
FFFECH	优先级指定标志寄存器 10	PR10L	PR10	R/W	○	○	○	FFH
FF FEDH		PR10H		R/W	○	○		FFH
FFFE EH	优先级指定标志寄存器 11	PR11L	PR11	R/W	○	○	○	FFH
FFFE FH		PR11H		R/W	○	○		FFH
FFFF0H	乘除数据寄存器 A (L)	MDAL		R/W	—	—	○	0000H
FFFF1H								
FFFF2H	乘除数据寄存器 A (H)	MDAH		R/W	—	—	○	0000H
FFFF3H								
FFFF4H	乘除数据寄存器 B (H)	MDBH		R/W	—	—	○	0000H
FFFF5H								
FFFF6H	乘除数据寄存器 B (L)	MDBL		R/W	—	—	○	0000H
FFFF7H								
FFFFEH	处理器模式控制寄存器	PMC		R/W	○	○	—	00H

备注 有关扩展 SFR (2nd SFR), 请参照“表 3-6 扩展 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。

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

各操作位单位的指定方法如下所示。

- 1 位操作
给 1 位操作指令的操作数（!addr16.bit）进行以下的记述：
已定义位名的情况：<位名>
没有定义位名的情况：<寄存器名>.<位号>或者<地址>.<位号>
- 8 位操作
给 8 位操作指令的操作数（!addr16）记述汇编程序定义的符号，也能指定地址。
- 16 位操作
给 16 位操作指令的操作数（!addr16）记述汇编程序定义的符号。当指定地址时，必须记述偶数地址。

扩展 SFR 一览表如表 3-6 所示。表中的项目的含义如下所示。

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

注意 不能存取未分配扩展 SFR（2nd SFR）的地址。

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

表 3-6 扩展 SFR (2nd SFR) 一览表 (1/5)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号	R/W	可操作位的范围			复位后
				1 位	8 位	16 位	
F0010H	A/D 转换器的模式寄存器 2	ADM2	R/W	○	○	—	00H
F0011H	转换结果比较上限值设定寄存器	ADUL	R/W	—	○	—	FFH
F0012H	转换结果比较下限值设定寄存器	ADLL	R/W	—	○	—	00H
F0013H	A/D 测试寄存器	ADTES	R/W	—	○	—	00H
F0030H	上拉电阻选择寄存器 0	PU0	R/W	○	○	—	00H
F0031H	上拉电阻选择寄存器 1	PU1	R/W	○	○	—	00H
F0033H	上拉电阻选择寄存器 3	PU3	R/W	○	○	—	00H
F0034H	上拉电阻选择寄存器 4	PU4	R/W	○	○	—	01H
F0035H	上拉电阻选择寄存器 5	PU5	R/W	○	○	—	00H
F0036H	上拉电阻选择寄存器 6	PU6	R/W	○	○	—	00H
F0037H	上拉电阻选择寄存器 7	PU7	R/W	○	○	—	00H
F003CH	上拉电阻选择寄存器 12	PU12	R/W	○	○	—	00H
F003EH	上拉电阻选择寄存器 14	PU14	R/W	○	○	—	00H
F0040H	端口输入模式寄存器 0	PIM0	R/W	○	○	—	00H
F0041H	端口输入模式寄存器 1	PIM1	R/W	○	○	—	00H
F0050H	端口输出模式寄存器 0	POM0	R/W	○	○	—	00H
F0051H	端口输出模式寄存器 1	POM1	R/W	○	○	—	00H
F0055H	端口输出模式寄存器 5	POM5	R/W	○	○	—	00H
F0060H	端口模式控制寄存器 0	PMC0	R/W	○	○	—	FFH
F0061H	端口模式控制寄存器 1	PMC1	R/W	○	○	—	FFH
F006EH	端口模式控制寄存器 14	PMC14	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
F0074H	定时器输入选择寄存器 0	TIS0	R/W	—	○	—	00H
F0076H	A/D 端口配置寄存器	ADPC	R/W	—	○	—	00H
F0077H	外围 I/O 重定向寄存器	PIOR	R/W	—	○	—	00H
F0078H	非法存储器存取检测控制寄存器	IAWCTL	R/W	—	○	—	00H
F0090H	数据闪存控制寄存器	DFLCTL	R/W	○	○	—	00H
F00A0H	高速内部振荡器的微调寄存器	HIOTRM	R/W	—	○	—	不定值 ^{注1}
F00A8H	高速内部振荡器的频率选择寄存器	HOCODIV	R/W	—	○	—	不定值 ^{注2}
F00E0H	乘除数据寄存器 C (L)	MDCL	R/W	—	—	○	0000H
F00E2H	乘除数据寄存器 C (H)	MDCH	R/W	—	—	○	0000H
F00E8H	乘除控制寄存器	MDUC	R/W	○	○	—	00H
F00F0H	外围允许寄存器 0	PER0	R/W	○	○	—	00H
F00F3H	副系统时钟提供模式控制寄存器	OSMC	R/W	—	○	—	00H
F00F5H	RAM 奇偶校验错误控制寄存器	RPECTL	R/W	○	○	—	00H
F00FEH	BCD 校正结果寄存器	BCDADJ	R	—	○	—	不定值
F0100H	串行状态寄存器 00	SSR00L	R	—	○	○	0000H
F0101H		—		—			

注 1. 复位值是产品发货时的调整值。

2. 这是选项字节 000C2H 的 FRQSEL2 ~ FRQSEL0 位设定的值。

表 3-6 扩展 SFR (2nd SFR) 一览表 (2/5)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
F0102H	串行状态寄存器 01	SSR01L	SSR01	R	—	○	○	0000H
F0103H		—			—			
F0104H	串行状态寄存器 02	SSR02L	SSR02	R	—	○	○	0000H
F0105H		—			—			
F0106H	串行状态寄存器 03	SSR03L	SSR03	R	—	○	○	0000H
F0107H		—			—			
F0108H	串行标志清除触发寄存器 00	SIR00L	SIR00	R/W	—	○	○	0000H
F0109H		—			—			
F010AH	串行标志清除触发寄存器 01	SIR01L	SIR01	R/W	—	○	○	0000H
F010BH		—			—			
F010CH	串行标志清除触发寄存器 02	SIR02L	SIR02	R/W	—	○	○	0000H
F010DH		—			—			
F010EH	串行标志清除触发寄存器 03	SIR03L	SIR03	R/W	—	○	○	0000H
F010FH		—			—			
F0110H	串行模式寄存器 00	SMR00		R/W	—	—	○	0020H
F0111H								
F0112H	串行模式寄存器 01	SMR01		R/W	—	—	○	0020H
F0113H								
F0114H	串行模式寄存器 02	SMR02		R/W	—	—	○	0020H
F0115H								
F0116H	串行模式寄存器 03	SMR03		R/W	—	—	○	0020H
F0117H								
F0118H	串行通信运行设定寄存器 00	SCR00		R/W	—	—	○	0087H
F0119H								
F011AH	串行通信运行设定寄存器 01	SCR01		R/W	—	—	○	0087H
F011BH								
F011CH	串行通信运行设定寄存器 02	SCR02		R/W	—	—	○	0087H
F011DH								
F011EH	串行通信运行设定寄存器 03	SCR03		R/W	—	—	○	0087H
F011FH								
F0120H	串行通道允许状态寄存器 0	SE0L	SE0	R	○	○	○	0000H
F0121H		—			—			
F0122H	串行通道开始寄存器 0	SS0L	SS0	R/W	○	○	○	0000H
F0123H		—			—			
F0124H	串行通道停止寄存器 0	ST0L	ST0	R/W	○	○	○	0000H
F0125H		—			—			
F0126H	串行时钟选择寄存器 0	SPS0L	SPS0	R/W	—	○	○	0000H
F0127H		—			—			
F0128H	串行输出寄存器 0	SO0		R/W	—	—	○	0F0FH
F0129H								
F012AH	串行输出允许寄存器 0	SOE0L	SOE0	R/W	○	○	○	0000H
F012BH		—			—			

表 3-6 扩展 SFR (2nd SFR) 一览表 (3/5)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
F0134H	串行输出电平寄存器 0	SOL0L	SOL0	R/W	—	○	○	0000H
F0135H		—			—			
F0138H	串行待机控制寄存器 0	SSC0L	SSC0	R/W	—	○	○	0000H
F0139H		—			—			
F0140H	串行状态寄存器 10	SSR10L	SSR10	R	—	○	○	0000H
F0141H		—			—			
F0142H	串行状态寄存器 11	SSR11L	SSR11	R	—	○	○	0000H
F0143H		—			—			
F0148H	串行标志清除触发寄存器 10	SIR10L	SIR10	R/W	—	○	○	0000H
F0149H		—			—			
F014AH	串行标志清除触发寄存器 11	SIR11L	SIR11	R/W	—	○	○	0000H
F014BH		—			—			
F0150H	串行模式寄存器 10	SMR10		R/W	—	—	○	0020H
F0151H								
F0152H	串行模式寄存器 11	SMR11		R/W	—	—	○	0020H
F0153H								
F0158H	串行通信运行设定寄存器 10	SCR10		R/W	—	—	○	0087H
F0159H								
F015AH	串行通信运行设定寄存器 11	SCR11		R/W	—	—	○	0087H
F015BH								
F0160H	串行通道允许状态寄存器 1	SE1L	SE1	R	○	○	○	0000H
F0161H		—			—			
F0162H	串行通道开始寄存器 1	SS1L	SS1	R/W	○	○	○	0000H
F0163H		—			—			
F0164H	串行通道停止寄存器 1	ST1L	ST1	R/W	○	○	○	0000H
F0165H		—			—			
F0166H	串行时钟选择寄存器 1	SPS1L	SPS1	R/W	—	○	○	0000H
F0167H		—			—			
F0168H	串行输出寄存器 1	SO1		R/W	—	—	○	0303H
F0169H								
F016AH	串行输出允许寄存器 1	SOE1L	SOE1	R/W	○	○	○	0000H
F016BH		—			—			
F0174H	串行输出电平寄存器 1	SOL1L	SOL1	R/W	—	○	○	0000H
F0175H		—			—			
F0180H	定时器计数寄存器 00	TCR00		R	—	—	○	FFFFH
F0181H								
F0182H	定时器计数寄存器 01	TCR01		R	—	—	○	FFFFH
F0183H								
F0184H	定时器计数寄存器 02	TCR02		R	—	—	○	FFFFH
F0185H								

表 3-6 扩展 SFR (2nd SFR) 一览表 (4/5)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
F0186H	定时器计数寄存器 03	TCR03		R	—	—	○	FFFFH
F0187H								
F0188H	定时器计数寄存器 04	TCR04		R	—	—	○	FFFFH
F0189H								
F018AH	定时器计数寄存器 05	TCR05		R	—	—	○	FFFFH
F018BH								
F018CH	定时器计数寄存器 06	TCR06		R	—	—	○	FFFFH
F018DH								
F018EH	定时器计数寄存器 07	TCR07		R	—	—	○	FFFFH
F018FH								
F0190H	定时器模式寄存器 00	TMR00		R/W	—	—	○	0000H
F0191H								
F0192H	定时器模式寄存器 01	TMR01		R/W	—	—	○	0000H
F0193H								
F0194H	定时器模式寄存器 02	TMR02		R/W	—	—	○	0000H
F0195H								
F0196H	定时器模式寄存器 03	TMR03		R/W	—	—	○	0000H
F0197H								
F0198H	定时器模式寄存器 04	TMR04		R/W	—	—	○	0000H
F0199H								
F019AH	定时器模式寄存器 05	TMR05		R/W	—	—	○	0000H
F019BH								
F019CH	定时器模式寄存器 06	TMR06		R/W	—	—	○	0000H
F019DH								
F019EH	定时器模式寄存器 07	TMR07		R/W	—	—	○	0000H
F019FH								
F01A0H	定时器状态寄存器 00	TSR00L	TSR00	R	—	○	○	0000H
F01A1H		—			—	—		
F01A2H	定时器状态寄存器 01	TSR01L	TSR01	R	—	○	○	0000H
F01A3H		—			—	—		
F01A4H	定时器状态寄存器 02	TSR02L	TSR02	R	—	○	○	0000H
F01A5H		—			—	—		
F01A6H	定时器状态寄存器 03	TSR03L	TSR03	R	—	○	○	0000H
F01A7H		—			—	—		
F01A8H	定时器状态寄存器 04	TSR04L	TSR04	R	—	○	○	0000H
F01A9H		—			—	—		
F01AAH	定时器状态寄存器 05	TSR05L	TSR05	R	—	○	○	0000H
F01ABH		—			—	—		
F01ACH	定时器状态寄存器 06	TSR06L	TSR06	R	—	○	○	0000H
F01ADH		—			—	—		

表 3-6 扩展 SFR (2nd SFR) 一览表 (5/5)

地址	扩展特殊功能寄存器 (2nd SFR) 名称	符号		R/W	可操作位的范围			复位后
					1 位	8 位	16 位	
F01AEH	定时器状态寄存器 07	TSR07L	TSR07	R	—	○	○	0000H
F01AFH		—			—			
F01B0H	定时器通道允许状态寄存器 0	TE0L	TE0	R	○	○	○	0000H
F01B1H		—			—			
F01B2H	定时器通道开始寄存器 0	TS0L	TS0	R/W	○	○	○	0000H
F01B3H		—			—			
F01B4H	定时器通道停止寄存器 0	TT0L	TT0	R/W	○	○	○	0000H
F01B5H		—			—			
F01B6H	定时器时钟选择寄存器 0	TPS0		R/W	—	—	○	0000H
F01B7H								
F01B8H	定时器输出寄存器 0	TO0L	TO0	R/W	—	○	○	0000H
F01B9H		—			—			
F01BAH	定时器输出允许寄存器 0	TOE0L	TOE0	R/W	○	○	○	0000H
F01BBH		—			—			
F01BCH	定时器输出电平寄存器 0	TOL0L	TOL0	R/W	—	○	○	0000H
F01BDH		—			—			
F01BEH	定时器输出模式寄存器 0	TOM0L	TOM0	R/W	—	○	○	0000H
F01BFH		—			—			
F0230H	IICA 控制寄存器 00	IICCTL00		R/W	○	○	—	00H
F0231H	IICA 控制寄存器 01	IICCTL01		R/W	○	○	—	00H
F0232H	IICA 低电平宽度设定寄存器 0	IICWL0		R/W	—	○	—	FFH
F0233H	IICA 高电平宽度设定寄存器 0	IICWH0		R/W	—	○	—	FFH
F0234H	从属地址寄存器 0	SVA0		R/W	—	○	—	00H
F02F0H	闪存 CRC 控制寄存器	CRC0CTL		R/W	○	○	—	00H
F02F2H	闪存 CRC 运算结果寄存器	PGCRCL		R/W	—	—	○	0000H

备注 有关 SFR 区的 SFR, 请参照“表 3-5 SFR 一览表”。

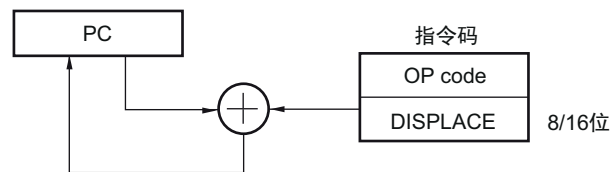
3.3 指令地址的寻址

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

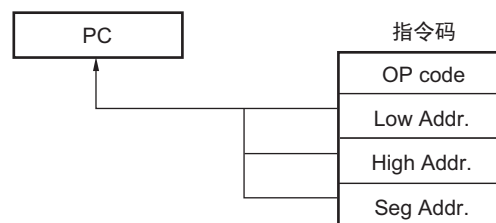
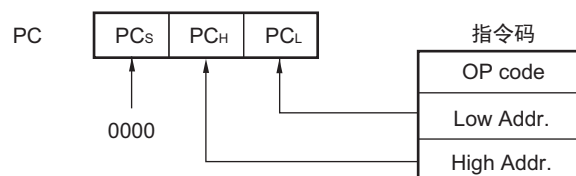


图 3-13 CALL !addr16/BR !addr16 的例子



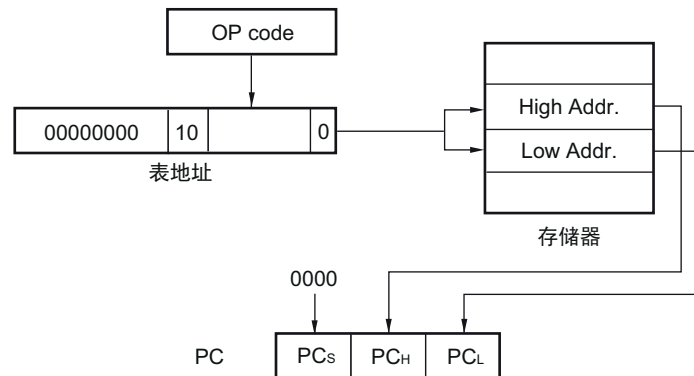
3.3.3 表间接寻址

【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 微控制器只能在 00000H ~ 0FFFFH 的 64K 字节空间内进行转移。

图 3-14 表间接寻址的概要

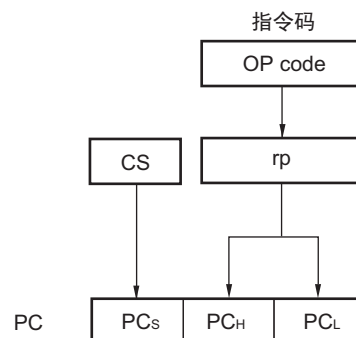


3.3.4 寄存器直接寻址

【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】

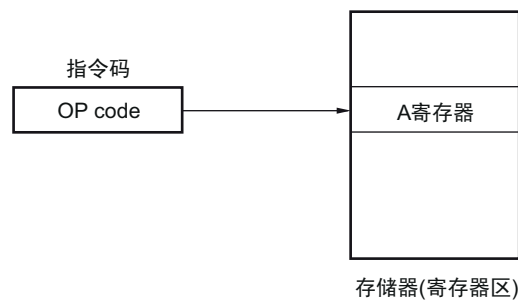
对于具有累加器等特殊功能寄存器的存取指令，在指令码中没有寄存器指定字段，而通过指令码直接指定。

【操作数形式】

指令能自动使用隐含寻址，因此没有特定的操作数形式。

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

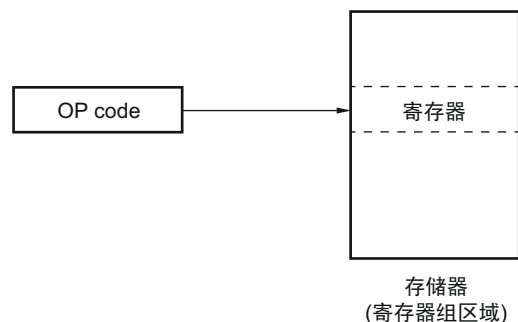
【功能】

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

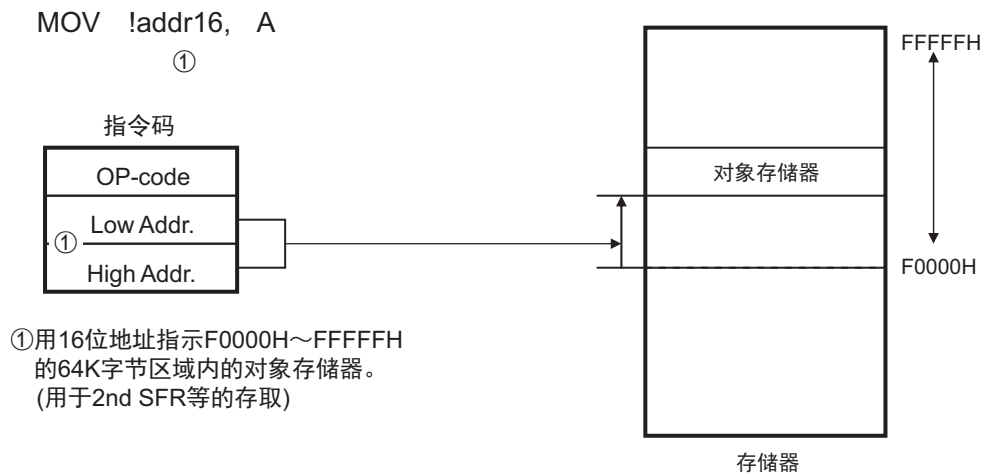
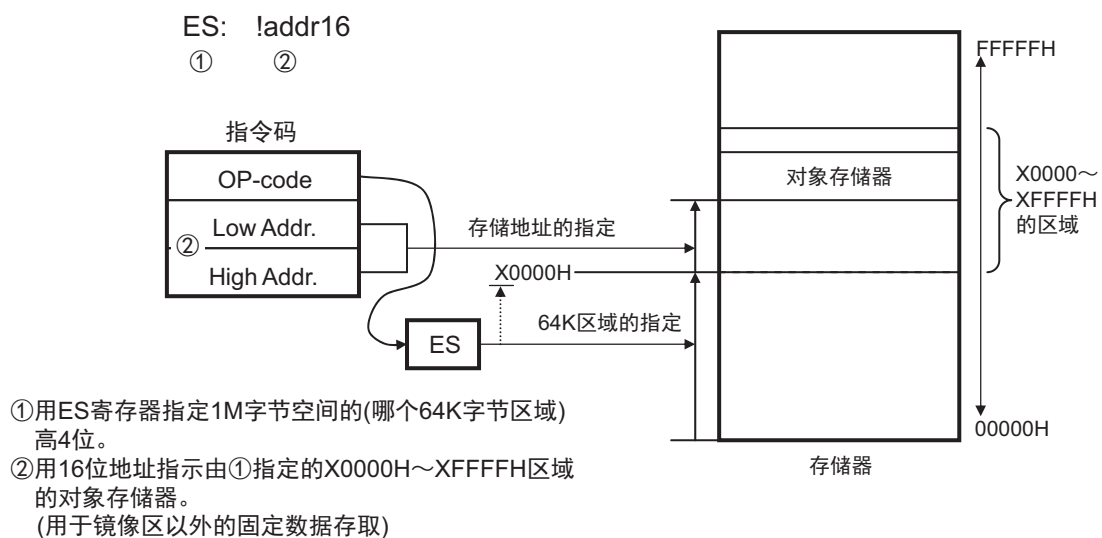


图 3-19 ES:!addr16 的例子



3.4.4 短直接寻址

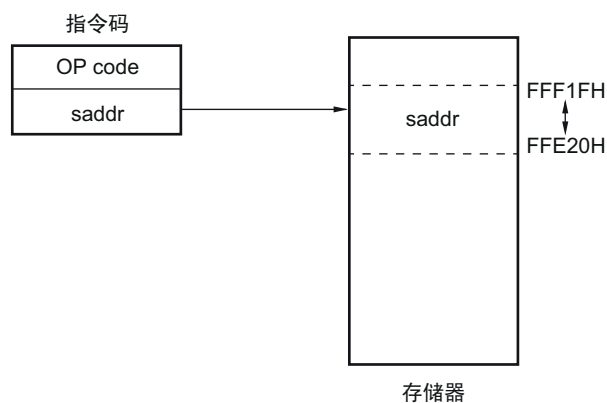
【功能】

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。

【操作数形式】

表现形式	记述方法
SADDR	标号、 FFE20H ~ FFF1FH 的立即数或者 0FE20H ~ 0FF1FH 的立即数 (只能指定 FFE20H ~ FFF1FH 的空间)
SADDRP	标号、 FFE20H ~ FFF1FH 的立即数或者 0FE20H ~ 0FF1FH 的立即数 (只限于偶数地址) (只能指定 FFE20H ~ FFF1FH 的空间)

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，并且还能通过 20 位立即数记述 FFE20H ~ FFF1FH 的值。

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

3.4.5 SFR 寻址

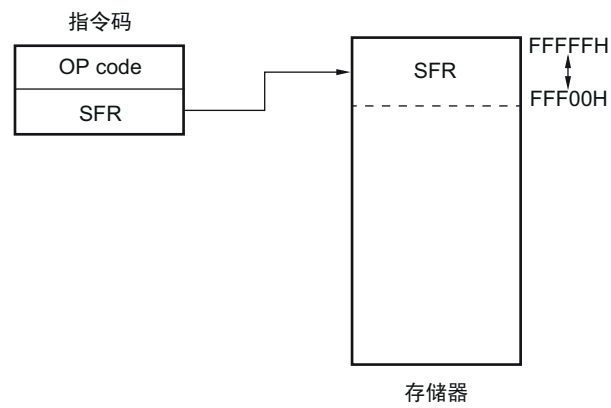
【功能】

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

【操作数形式】

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

图 3-21 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

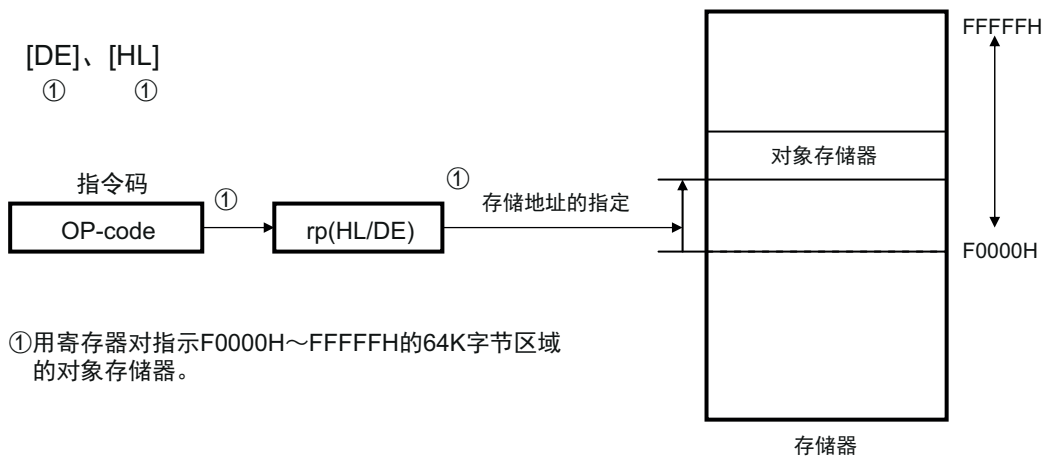
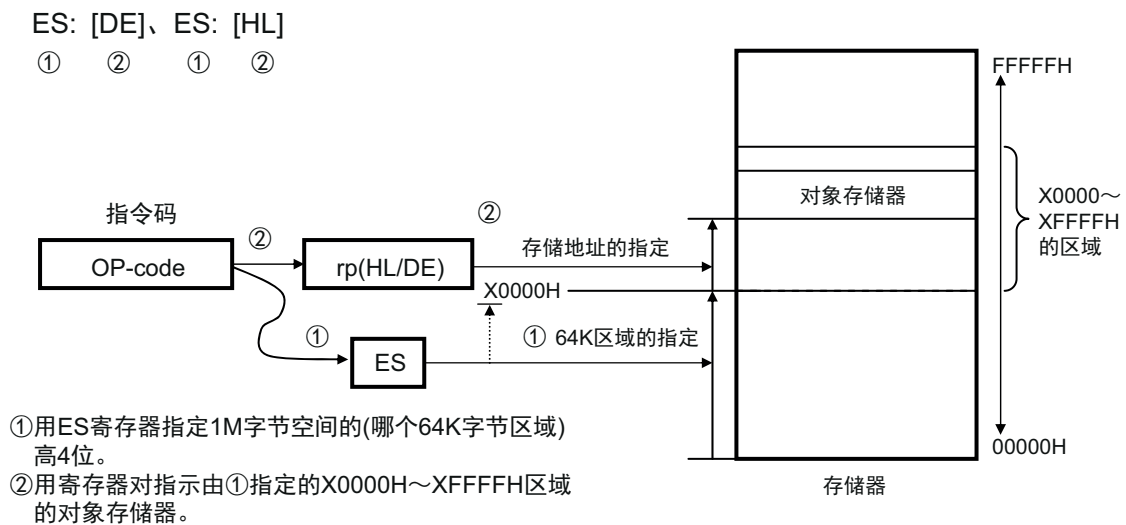


图 3-23 ES:[DE]、ES:[HL] 的例子



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

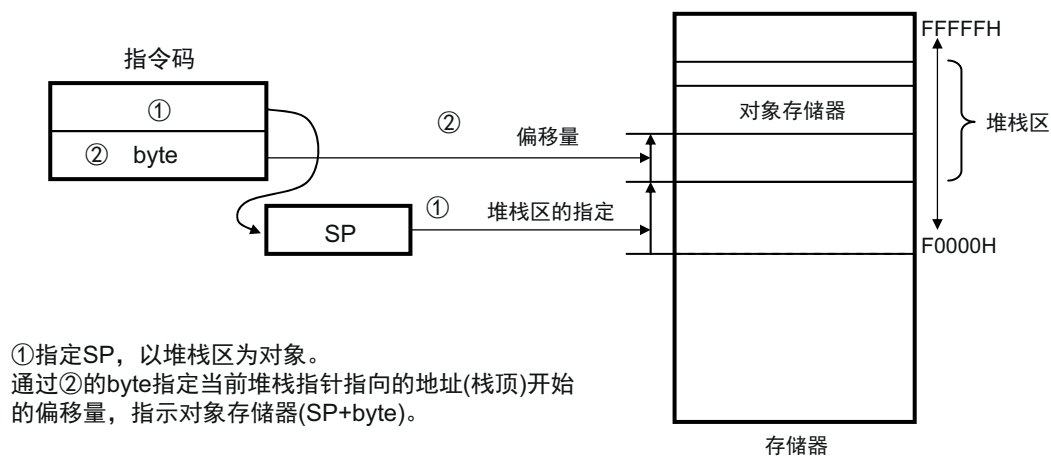


图 3-25 [HL+byte]、[DE+byte] 的例子

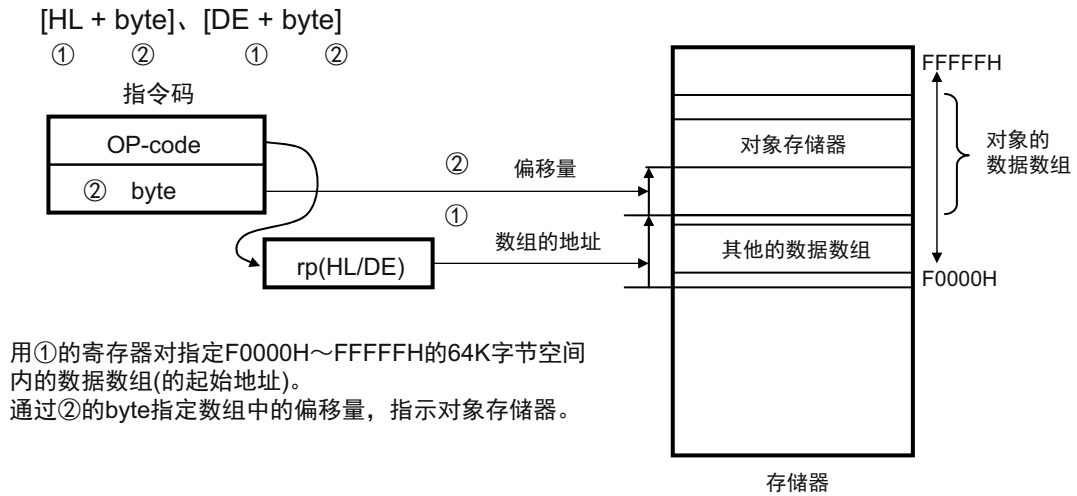


图 3-26 word[B]、word[C] 的例子

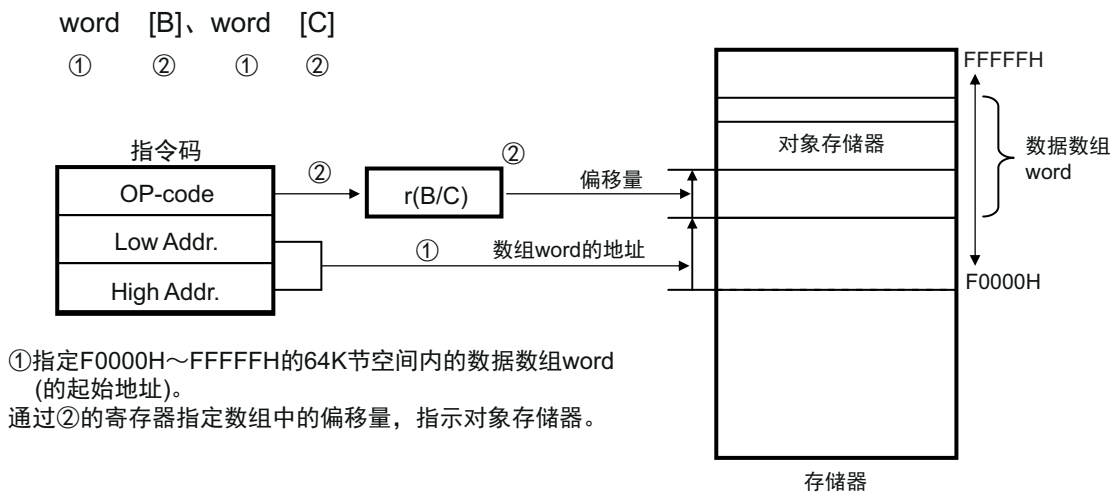


图 3-27 word[BC] 的例子

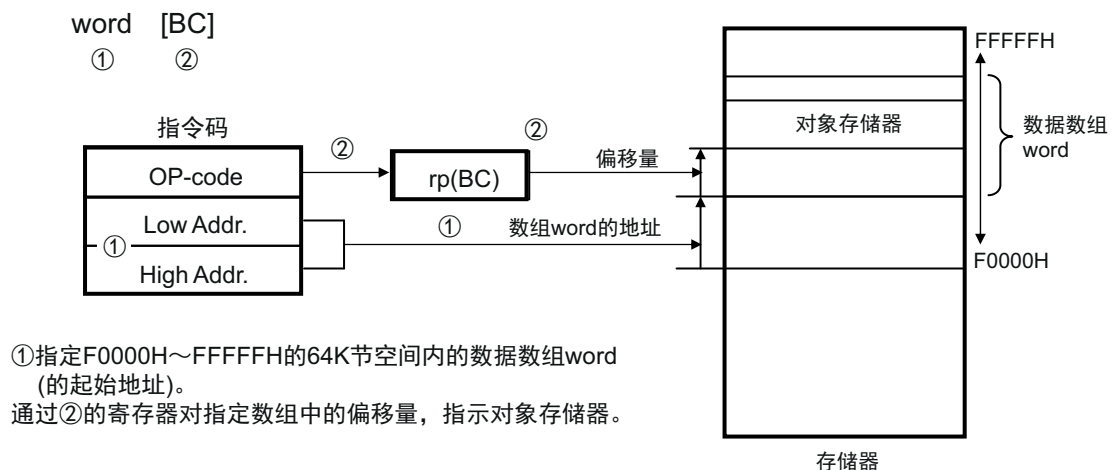


图 3-28 ES:[HL+byte]、ES:[DE+byte] 的例子

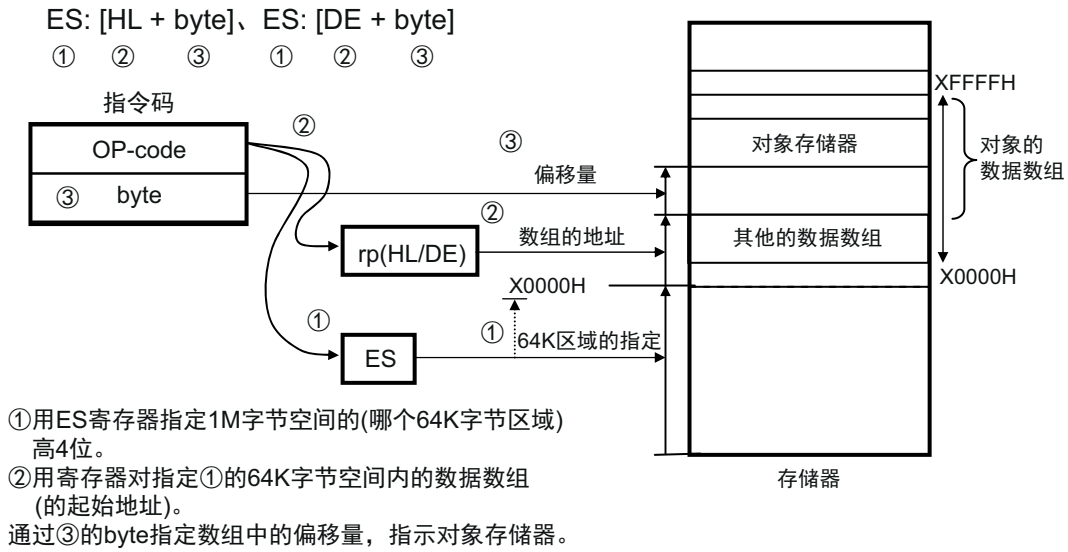


图 3-29 ES:word[B]、ES:word[C] 的例子

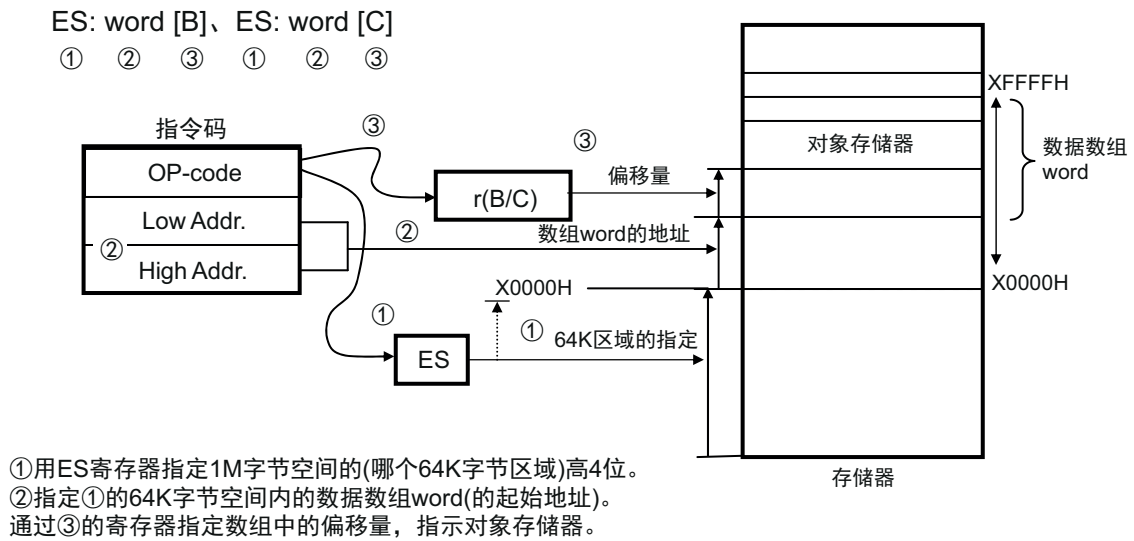
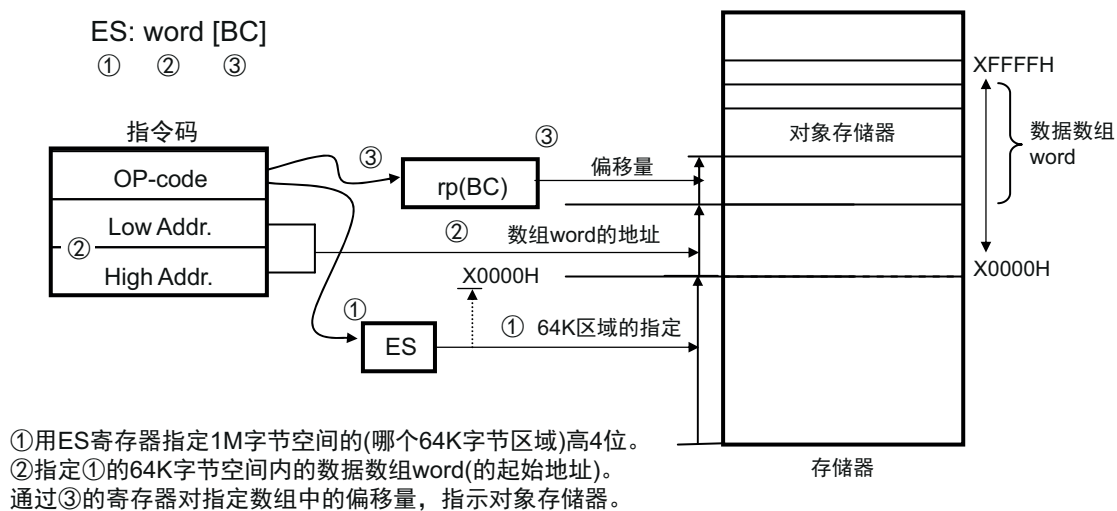


图 3-30 ES:word[BC] 的例子



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

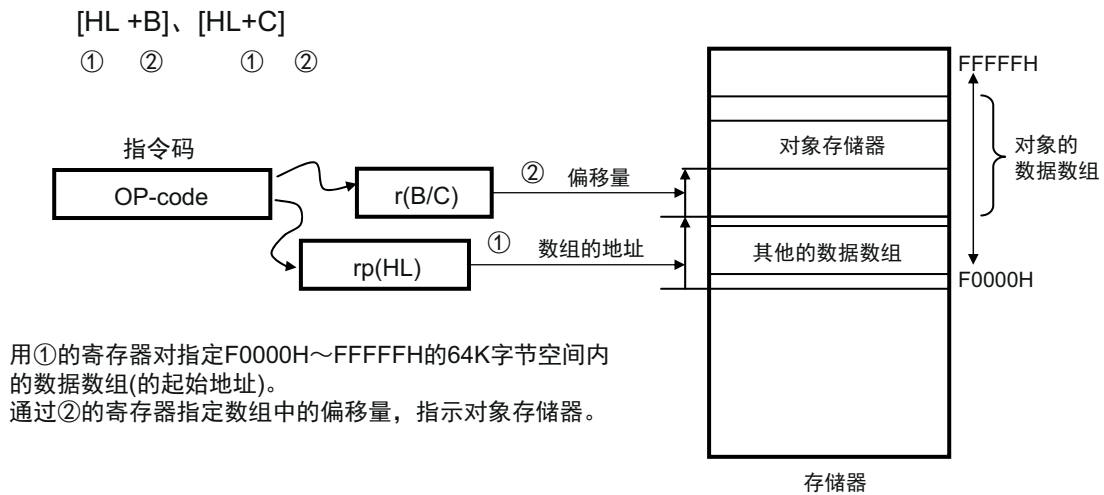
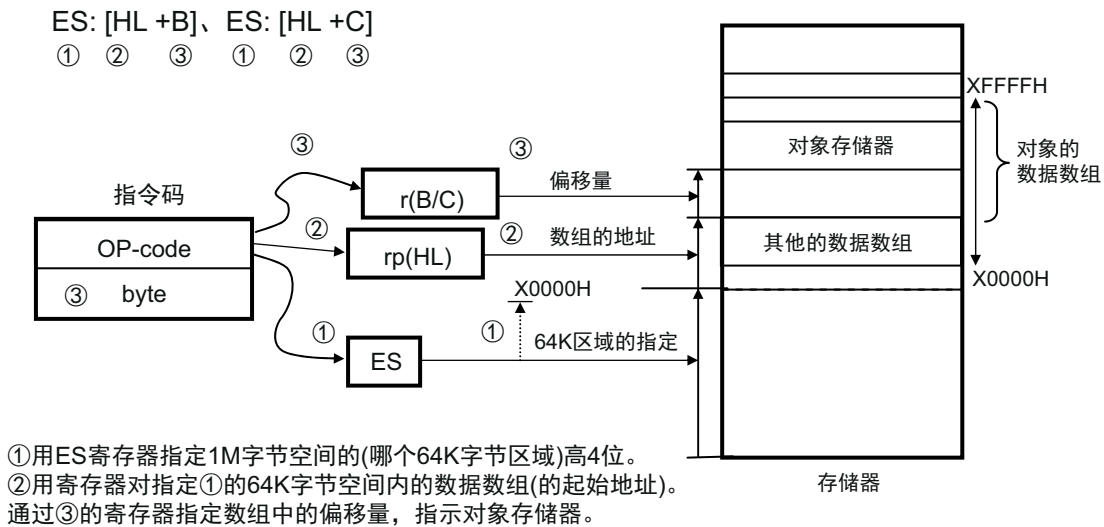


图 3-32 ES:[HL+B]、ES:[HL+C] 的例子



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

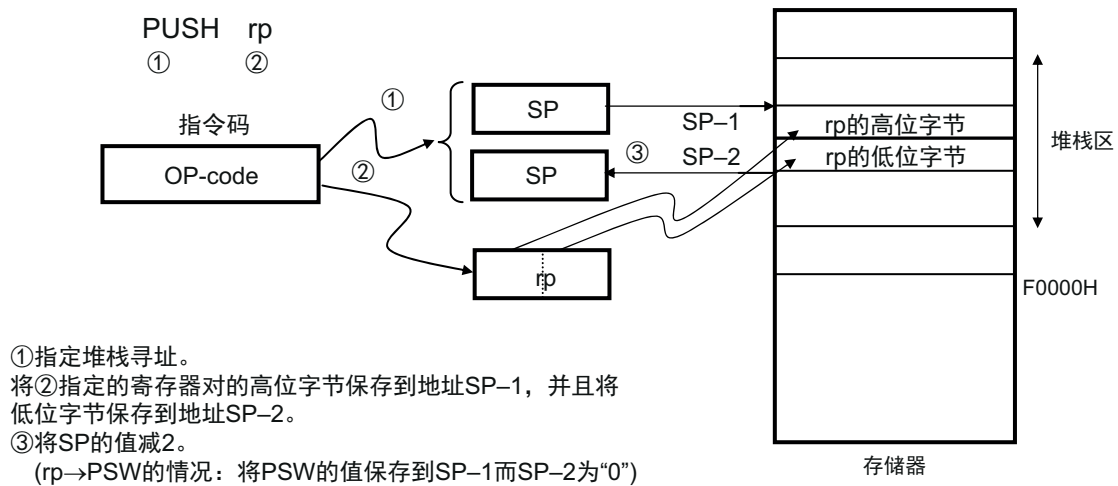


图 3-34 POP 的例子

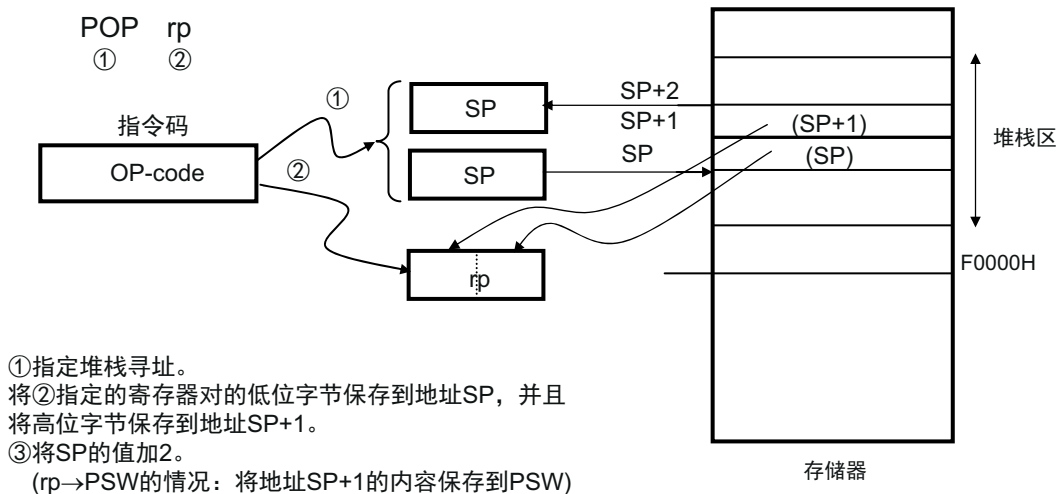


图 3-35 CALL、CALLT 的例子

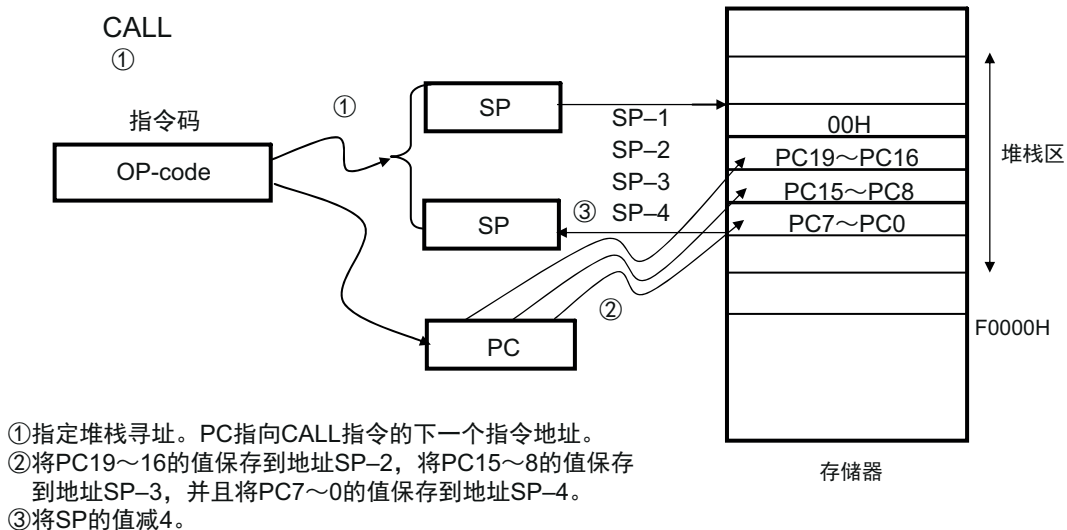


图 3-36 RET 的例子

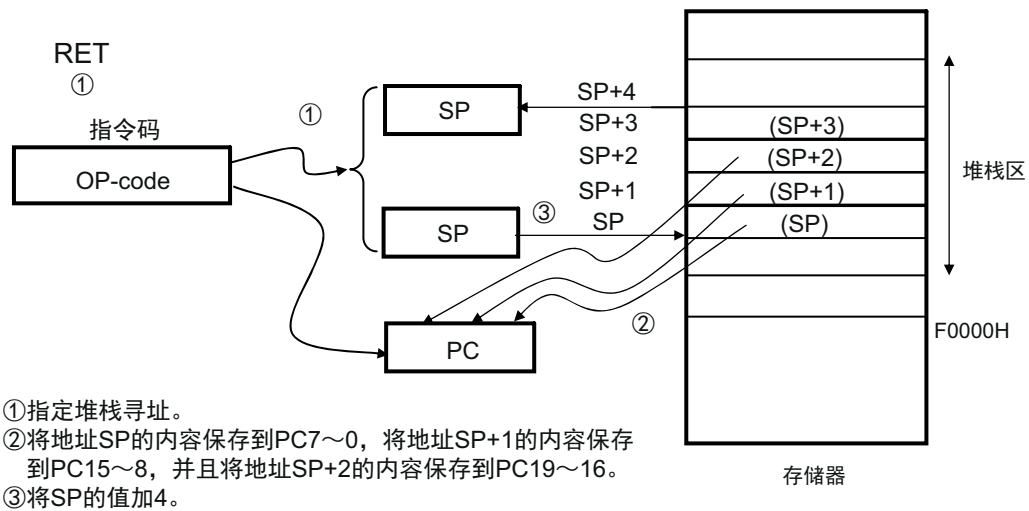


图 3-37 中断、BRK 的例子

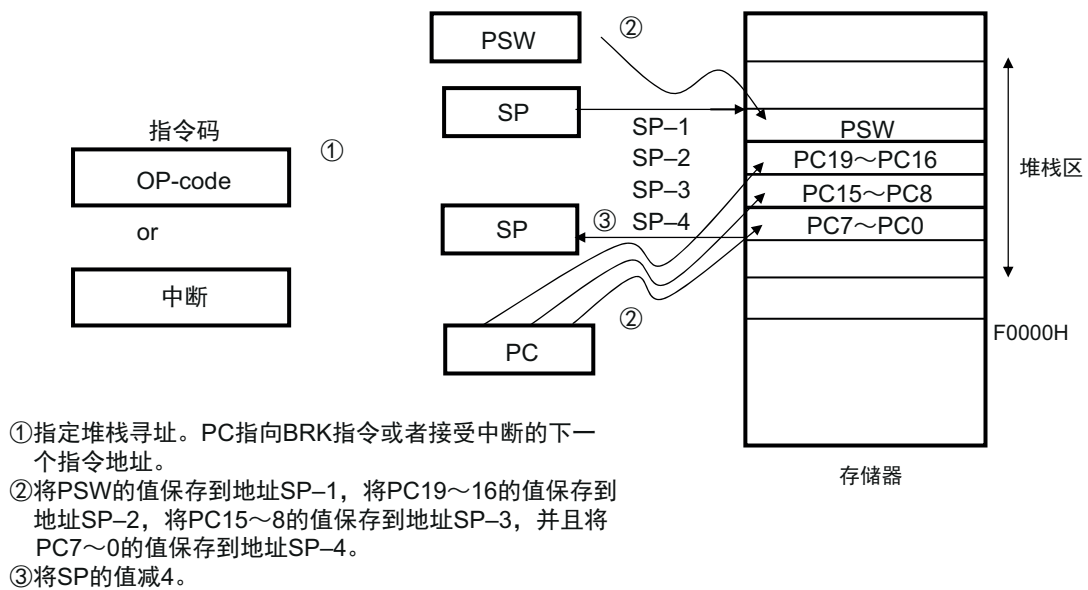
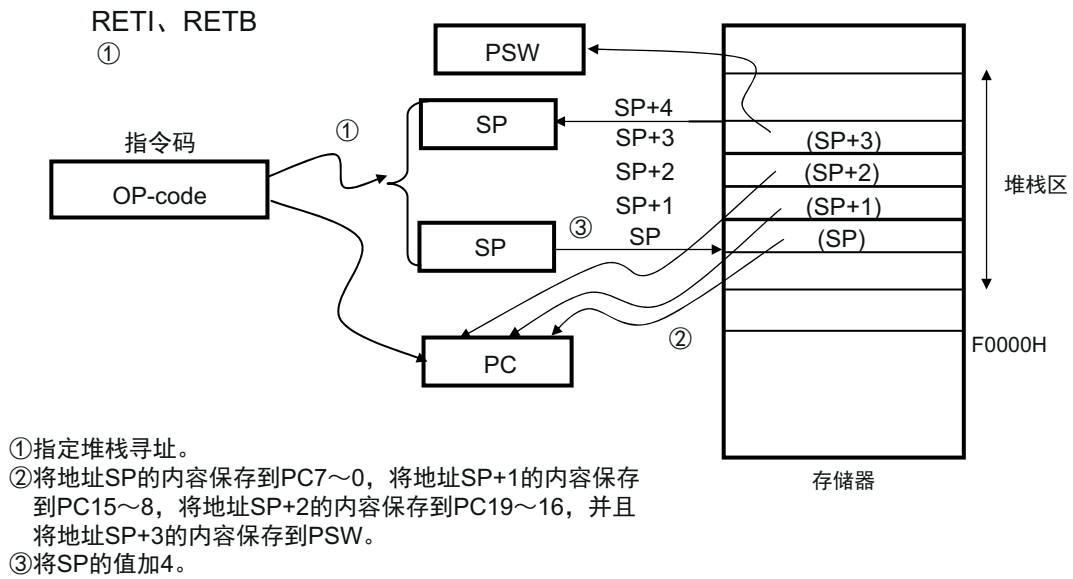


图 3-38 RETI、RETB 的例子



第 4 章 端口功能

4.1 端口功能

本产品提供数字输入 / 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 (PM0 ~ PM7、PM12、PM14) 端口寄存器 (P0 ~ P7、P12 ~ P14) 上拉电阻选择寄存器 (PU0、PU1、PU3 ~ PU5、PU7、PU12、PU14) 端口输入模式寄存器 (PIM0、PIM1) 端口输出模式寄存器 (POM0、POM1、POM5) 端口模式控制寄存器 (PMC0、PMC12、PMC14) A/D 端口配置寄存器 (ADPC) 外围 I/O 重定向寄存器 (PIOR)
端口	合计：28 个 (CMOS 输入 / 输出：22 个 (N 沟道漏极开路输入 / 输出 [V_{DD} 耐压]：9 个)， CMOS 输入：3 个，N 沟道漏极开路输入 / 输出 [6V 耐压]：3 个)
上拉电阻	合计：18 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。

能通过设定端口输入模式寄存器 0 (PIM0)，以位为单位给 P01、P03、P04 引脚的输入指定通常的输入缓冲或者 TTL 输入缓冲。

能通过端口输出模式寄存器 0 (POM0)，以位为单位将 P00 引脚和 P02 ~ P04 引脚的输出设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。

当将 P00 ~ P03 引脚用作数字输入 / 输出端口时，必须通过端口模式控制寄存器 0 (PMC0) 设定为数字输入 / 输出 (能以位为单位进行设定)。

端口 0 作为复用功能，有定时器的输入 / 输出、A/D 转换器的模拟输入、串行接口的数据输入 / 输出和时钟的输入 / 输出。

在产生复位信号后，端口 0 变为以下状态：

- 本产品的 P00 引脚和 P01 引脚：模拟输入

4.2.2 端口 1

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

能通过设定端口输入模式寄存器 1 (PIM1)，以位为单位给 P10、P11、P13 ~ P17 引脚的输入指定通常的输入缓冲或者 TTL 输入缓冲。

能通过端口输出模式寄存器 1 (POM1)，以位为单位将 P10 ~ P15 引脚和 P17 引脚的输出设定为 N 沟道漏极开路输出 (V_{DD} 耐压)。

端口 1 作为复用功能，有串行接口的数据输入 / 输出、时钟的输入 / 输出、编程 UART 的输入 / 输出、定时器的输入 / 输出和外部中断的请求输入。

在产生复位信号后，端口 1 变为输入模式。

4.2.3 端口 2

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

端口 2 作为复用功能，有 A/D 转换器的模拟输入、A/D 转换器的正 (+) 基准电压输入和 A/D 转换器的负 (-) 基准电压输入。

当将 P20/ANI0 ~ P23/ANI3 用作数字输入 / 输出时，必须在通过 A/D 端口配置寄存器 (ADPC) 设定为数字输入 / 输出后从高位开始使用。

当将 P20/ANI0 ~ P23/ANI3 用作模拟输入时，必须在通过 A/D 端口配置寄存器 (ADPC) 设定为模拟输入并且通过 PM2 寄存器设定为输入模式后从低位开始使用。

表 4-2 P20/ANI0 ~ P23/ANI3 引脚的功能设定

ADPC 寄存器	PM2 寄存器	ADS 寄存器	P20/ANI0 ~ P23/ANI3 引脚
数字输入 / 输出的选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟功能的选择	输入模式	选择 ANI。	模拟输入 (转换对象)
		不选择 ANI。	模拟输入 (非转换对象)
	输出模式	选择 ANI。	禁止设定。
		不选择 ANI。	

在产生复位信号后，P20/ANI0 ~ P23/ANI3 都变为模拟输入。

4.2.4 端口 3

端口 3 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 3 (PM3) 以位为单位指定输入模式或者输出模式。当将 P30 引脚和 P31 引脚用作输入端口时，能通过上拉电阻选择寄存器 3 (PU3) 以位为单位使用内部上拉电阻。

端口 3 作为复用功能，有外部中断的请求输入、时钟 / 蜂鸣器的输出、定时器的输入 / 输出和 A/D 转换器的模拟输入。

在产生复位信号后，P30 和 P31 变为输入模式。

4.2.5 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 引脚用作输入端口时，能通过上拉电阻选择寄存器 4（PU4）以位为单位使用内部上拉电阻。

端口 4 作为复用功能，有用于闪存编程器 / 调试器的数据输入 / 输出。

在产生复位信号后，端口 4 变为输入模式。

4.2.6 端口 5

端口 5 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 5（PM5）以位为单位指定输入模式或者输出模式。当将 P50 引脚和 P51 引脚用作输入端口时，能通过上拉电阻选择寄存器 5（PU5）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 5（POM5），以位为单位将 P50 引脚的输出设定为 N 沟道漏极开路输出（ V_{DD} 耐压）。

端口 5 作为复用功能，有串行接口的数据输入 / 输出。

在产生复位信号后，端口 5 变为输入模式。

4.2.7 端口 6

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

P60 ~ P62 引脚的输出为 N 沟道漏极开路输出（6V 耐压）。

端口 6 作为复用功能，有串行接口的数据输入 / 输出。

在产生复位信号后，端口 6 变为输入模式。

4.2.8 端口 7

端口 7 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 7（PM7）以位为单位指定输入模式或者输出模式。当将端口 7 用作输入端口时，能通过上拉电阻选择寄存器 7（PU7）以位为单位使用内部上拉电阻。

在产生复位信号后，端口 7 变为输入模式。

4.2.9 端口 12

P120 是带输出锁存器的 1 位输入 / 输出端口，能通过端口模式寄存器 12（PM12）以位为单位指定输入模式或者输出模式。当将端口 12 用作输入端口时，能通过上拉电阻选择寄存器 12（PU12）使用内部上拉电阻。

P121 和 P122 是 2 位输入专用端口。

能通过设定端口模式控制寄存器 12（PMC12）将 P120 引脚指定为数字输入 / 输出或者模拟输入。

端口 12 作为复用功能，有 A/D 转换器的模拟输入、主系统时钟的谐振器连接和主系统时钟的外部时钟输入。

在产生复位信号后，P120 变为模拟输入，P121 和 P122 变为输入模式。

4.2.10 端口 13

P137 是 1 位输入专用端口。

P137 固定为输入模式。

端口 13 作为复用功能，有外部中断的请求输入。

4.2.11 端口 14

端口 14 是带输出锁存器的输入 / 输出端口，能通过端口模式寄存器 14（PM14）以位为单位指定输入模式或者输出模式。

能通过设定端口模式控制寄存器 14（PMC14），以位为单位将 P147 引脚指定为数字输入 / 输出或者模拟输入。

端口 14 作为复用功能，有 A/D 转换器的模拟输入。

在产生复位信号后，P147 变为模拟输入。

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 (PMxx)
- 端口寄存器 (Pxx)
- 上拉电阻选择寄存器 (PUxx)
- 端口输入模式寄存器 (PIMx)
- 端口输出模式寄存器 (POMx)
- 端口模式控制寄存器 (PMCxx)
- A/D 端口配置寄存器 (ADPC)
- 外围 I/O 重定向寄存器 (PIOR)

表 4-3 各产品分配的 PMxx、Pxx、PUxx、PIMxx、POMxx、PMCxx 寄存器及其位

端口		位名						32 引脚
		PMxx 寄存器	Pxx 寄存器	PUxx 寄存器	PIMxx 寄存器	POMxx 寄存器	PMCxx 寄存器	
端口 0	0	PM00	P00	PU00	—	POM00	PMC00	○
	1	PM01	P01	PU01	PIM01	—	PMC01	○
端口 1	0	PM10	P10	PU10	PIM10	POM10	—	○
	1	PM11	P11	PU11	PIM11	POM11	—	○
	2	PM12	P12	PU12	—	POM12	—	○
	3	PM13	P13	PU13	PIM13	POM13	—	○
	4	PM14	P14	PU14	PIM14	POM14	—	○
	5	PM15	P15	PU15	PIM15	POM15	—	○
	6	PM16	P16	PU16	PIM16	—	—	○
端口 2	7	PM17	P17	PU17	PIM17	POM17	—	○
	0	PM20	P20	—	—	—	—	○
	1	PM21	P21	—	—	—	—	○
	2	PM22	P22	—	—	—	—	○
端口 3	3	PM23	P23	—	—	—	—	○
	0	PM30	P30	PU30	—	—	—	○
端口 4	1	PM31	P31	PU31	—	—	—	○
	0	PM40	P40	PU40	—	—	—	○
端口 5	0	PM50	P50	PU50	—	POM50	—	○
	1	PM51	P51	PU51	—	—	—	○
端口 6	0	PM60	P60	—	—	—	—	○
	1	PM61	P61	—	—	—	—	○
	2	PM62	P62	—	—	—	—	○
端口 7	0	PM70	P70	PU70	—	—	—	○
端口 12	0	PM120	P120	PU120	—	—	PMC120	○
	1	—	P121	—	—	—	—	○
	2	—	P122	—	—	—	—	○
端口 13	7	—	P137	—	—	—	○	
端口 14	7	PM147	P147	PU147	—	—	PMC147	○

4.3.1 端口模式寄存器 (PMxx)

这是以位为单位设定端口输入 / 输出的寄存器。

通过 1 位或者 8 位存储器操作指令设定端口模式寄存器。

在产生复位信号后，这些寄存器的值变为“FFH”。

当将端口引脚用作复用功能的引脚时，必须参照“4.5 使用复用功能时的寄存器设定”进行设定。

图 4-1 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	1	1	1	1	1	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	1	1	1	1	1	1	1	FFF2EH	FFH	R/W
PMmn	Pmn 引脚的输入 / 输出模式的选择 (m=0 ~ 7、12、14, n=0 ~ 7)										
0	输出模式 (输出缓冲器 ON)										
1	输入模式 (输出缓冲器 OFF)										

注意 必须给未分配的位设定初始值。

4.3.2 端口寄存器 (Pxx)

这是设定端口输出锁存器的值的寄存器。

读时，在输入模式中读引脚电平，而在输出模式中读端口的输出锁存器的值注。

通过 1 位或者 8 位存储器操作指令设定端口寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

注 在将 P20 ~ P23、P120 和 P147 设定为 A/D 转换器的模拟输入的情况下，如果在输入模式中读端口，读取值就不是引脚电平而总是“0”。

图 4-2 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (输出锁存器)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (输出锁存器)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (输出锁存器)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (输出锁存器)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (输出锁存器)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (输出锁存器)	R/W
P6	0	0	0	0	0	P62	P61	P60	FFF06H	00H (输出锁存器)	R/W
P7	0	0	0	0	0	0	0	P70	FFF07H	00H (输出锁存器)	R/W
P12	0	0	0	0	0	0	0	P120	FFF0CH	不定值	R/W
P13	P137	0	0	0	0	0	0	0	FFF0DH	不定值	R
P14	P147	0	0	0	0	0	0	0	FFF0EH	00H (输出锁存器)	R/W

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

注意 必须给未分配的位设定初始值。

备注 m=0 ~ 7、12 ~ 14, n=0 ~ 7

4.3.3 上拉电阻选择寄存器 (PUxx)

这是设定是否使用内部上拉电阻的寄存器。只能对通过上拉电阻选择寄存器指定使用内部上拉电阻的引脚并且设定为通常的输出模式 (POMmn=0) 和输入模式 (PMmn=1) 的位, 以位为单位使用内部上拉电阻。对于设定为输出模式的位, 与上拉电阻选择寄存器的设定无关, 不连接内部上拉电阻。当用作复用功能的输出引脚或者设定为模拟输入 (PMC=1、ADPC=1) 时也相同。

通过 1 位或者 8 位存储器操作指令设定上拉电阻选择寄存器。

在产生复位信号后, 这些寄存器的值变为“00H”(只有 PU4 为“01H”)。

注意 当是带 PIMn 寄存器的端口并且将信号从不同电位的设备输入到 TTL 缓冲器时, 必须将 PUm 位置“0”并且通过外部电阻将此引脚上拉到不同电位设备的电源。

图 4-3 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU7	0	0	0	0	0	0	0	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	0	0	0	0	0	0	0	F003EH	00H	R/W

PUmn	Pmn 引脚的内部上拉电阻的选择 (m=0、1、3~5、7、12、14, n=0~7)
0	不连接内部上拉电阻。
1	连接内部上拉电阻。

注意 必须给未分配的位设定初始值。

4.3.4 端口输入模式寄存器 (PIMxx)

这是以位为单位设定输入缓冲器的寄存器。

在和不同电位的外部设备进行串行通信时，能选择 TTL 输入缓冲器。

通过 1 位或者 8 位存储器操作指令设定端口输入模式寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 4-4 端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIM0	0	0	0	0	0	0	0	PIM00	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	PIM13	0	PIM11	PIM10	F0041H	00H	R/W

PIMmn	Pmn 引脚的输入缓冲器的选择 (m=0、1, n=0~7)
0	通常的输入缓冲器
1	TTL 输入缓冲器

注意 必须给未分配的位设定初始值。

4.3.5 端口输出模式寄存器 (POMxx)

这是以位为单位设定输出模式的寄存器。

在和不同电位的外部设备进行串行通信以及和同电位的外部设备进行简易 I²C 通信时，能给 SDA00、SDA11、SDA20 引脚选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

POMxx 寄存器和 PUxx 寄存器一起设定是否使用内部上拉电阻。

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，这些寄存器的值变为“00H”。

注意 对于设定 N 沟道漏极开路输出 (V_{DD} 耐压) 模式的位，不连接内部上拉电阻。

图 4-5 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W
POM5	0	0	0	0	0	0	0	POM50	F0055H	00H	R/W

POMmn	Pmn 引脚的输出模式的选择 (m=0、1、5, n=0~7)
0	通常的输出模式
1	N 沟道漏极开路输出 (V _{DD} 耐压) 模式

注意 必须给未分配的位设定初始值。

4.3.6 端口模式控制寄存器 (PMCxx)

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，这些寄存器的值变为“FFH”。

图 4-6 端口模式控制寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC0	1	1	1	1	1	1	PMC01	PMC00	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	F006EH	FFH	R/W
PMCmn	Pmn 引脚的数字输入 / 输出或者模拟输入的选择 (m=0、12、14, n=0 ~ 7)										
0	数字输入 / 输出 (模拟输入以外的复用功能)										
1	模拟输入										

- 注意 1. 对于由 PMCxx 寄存器设定为模拟输入的端口，必须通过端口模式寄存器 0、12、14 (PM0、PM12、PM14) 将其设定为输入模式。
2. 对于由 PMCxx 寄存器设定为数字输入 / 输出的引脚，不能通过模拟输入通道指定寄存器 (ADS) 进行设定。
3. 必须给未配置的位设定初始值。

4.3.7 A/D 端口配置寄存器 (ADPC)

这是将 P20/ANI0 ~ P23/ANI3 引脚切换为端口的数字输入 / 输出或者 A/D 转换器的模拟输入的寄存器。
通过 8 位存储器操作指令设定 ADPC 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 4-7 A/D 端口配置寄存器 (ADPC) 的格式

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

ADPC3	ADPC2	ADPC1	ADPC0	模拟输入 (A) / 数字输入 / 输出 (D) 的切换			
				ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A
0	0	0	1	D	D	D	D
0	0	1	0	D	D	D	A
0	0	1	1	D	D	A	A
0	1	0	0	D	A	A	A
0	1	0	1	A	A	A	A

- 注意 1. 对于由 ADPC 寄存器设定为模拟输入的端口，必须通过端口模式寄存器 2 (PM2) 将其设定为输入模式。
2. 对于由 ADPC 寄存器设定为数字输入 / 输出的引脚，不能通过模拟输入通道指定寄存器 (ADS) 进行设定。
3. 当使用 AV_{REFP} 和 AV_{REFM} 时，必须将 ANI0 和 ANI1 设定为模拟输入并且将端口模式寄存器设定为输入模式。

4.3.8 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。

在允许该功能的运行前，能更改重定向的设定。

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 4-8 外围 I/O 重定向寄存器 (PIOR) 的格式

地址: F0077H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	0	PIOR2	PIOR1	PIOR0

位	复用功能	32 引脚	
		设定值	
		0	1
PIOR2	SCLA0	P60	P14
	SDAA0	P61	P13
PIOR1	TxD2	P13	—
	RxD2	P14	—
	SCL20	P15	—
	SDA20	P14	—
	SI20	P14	—
	SO20	P13	—
	SCK20	P15	—
	TxD0	P12	P17
	RxD0	P11	P16
	SCL00	P10	—
	SDA00	P11	—
	SI00	P11	—
	SO00	P12	—
	SCK00	P10	—
PIOR0	TI02/TO02	P17	P15
	TI03/TO03	P31	P14
	TI04/TO04	—	P13
	TI05/TO05	—	P12
	TI06/TO06	—	P11
	TI07/TO07	—	P10

备注 一：不能用作复用功能。

4.4 端口功能的运行

如下所示，端口的运行因输入 / 输出模式的设定而不同。

4.4.1 输入 / 输出端口的写操作

(1) 输出模式的情况

能通过传送指令给输出锁存器写值，并且输出锁存器的内容从引脚输出。

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

(2) 输入模式的情况

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

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

4.4.2 输入 / 输出端口的读操作

(1) 输出模式的情况

能通过传送指令读输出锁存器的内容，但是输出锁存器的内容不发生变化。

(2) 输入模式的情况

能通过传送指令读引脚状态，但是输出锁存器的内容不发生变化。

4.4.3 输入 / 输出端口的运算

(1) 输出模式的情况

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

保持被写在输出锁存器中的数据，直到下一次写数据为止。

在产生复位信号时清除输出锁存器的数据。

(2) 输入模式的情况

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

在产生复位信号时清除输出锁存器的数据。

4.4.4 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应

能通过端口输入模式寄存器（PIM_{xx}）和端口输出模式寄存器（POM_{xx}）切换输入 / 输出缓冲器，连接不同电位（1.8V、2.5V、3V）的外部设备。

当从不同电位（1.8V、2.5V、3V）的外部设备输入时，通过按位设定端口输入模式寄存器 0、1（PIM0、PIM1），进行通常的输入（CMOS）和 TTL 输入缓冲的切换。

当输出到不同电位（1.8V、2.5V、3V）的外部设备时，通过按位设定端口输出模式寄存器 0、1（POM0、POM1），进行通常的输出（CMOS）和 N 沟道漏极开路（V_{DD} 耐压）的切换。

以下说明有关串行接口的连接。

(1) 将 UART0 ~ UART2、CSI00、CSI20 功能的输入端口用于 TTL 输入缓冲器时的设定步骤

UART0 的情况：P11（P16）
 UART1 的情况：P03
 UART2 的情况：P14
 CSI00 的情况：P10、P11（P16）
 CSI20 的情况：P14、P15

备注 能通过设定外围 I/O 重定向寄存器（PIOR），分配（）内的功能。

- ① 通过外部电阻将使用的输入引脚上拉到对象设备的电源（不能使用内部上拉电阻）。
- ② 将 PIM0 寄存器和 PIM1 寄存器的对应位置“1”，切换到 TTL 输入缓冲器。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。
- ③ 允许串行阵列单元的运行，设定为 UART/CSI 模式。

(2) 将 UART0 ~ UART2、CSI00、CSI20 功能的输出端口用于 N 沟道漏极开路输出模式时的设定步骤

UART0 的情况：P12（P17）
 UART1 的情况：P02
 UART2 的情况：P13
 CSI00 的情况：P10、P12（P17）
 CSI20 的情况：P13、P15

备注 能通过设定外围 I/O 重定向寄存器（PIOR），分配（）内的功能。

- ① 通过外部电阻将使用的输入引脚上拉到对象设备的电源（不能使用内部上拉电阻）。
- ② 在解除复位后，端口模式为输入模式（Hi-Z）。
- ③ 将对应的端口输出锁存器置“1”。
- ④ 将 POM0 寄存器和 POM1 寄存器的对应位置“1”，设定为 N 沟道漏极开路输出（V_{DD} 耐压）模式。
- ⑤ 允许串行阵列单元的运行，设定为 UART/CSI 模式。
- ⑥ 将 PM0 寄存器和 PM1 寄存器的对应位设定为输出模式。此时，因为输出数据是高电平，所以引脚处于 Hi-Z 状态。

(3) 将 IIC00、IIC20 功能的输入 / 输出端口用于不同电位（1.8V、2.5V、3V）时的设定步骤

IIC00 的情况： P10、P11

IIC20 的情况： P14、P15

备注 能通过设定外围 I/O 重定向寄存器（PIOR），分配（）内的功能。

- ① 通过外部电阻将使用的输入引脚上拉到对象设备的电源（不能使用内部上拉电阻）。
- ② 在解除复位后，端口模式为输入模式（Hi-Z）。
- ③ 将对应的端口输出锁存器置“1”。
- ④ 将POM0寄存器和POM1寄存器的对应位置“1”，设定为N沟道漏极开路输出（ V_{DD} 耐压）模式。
- ⑤ 将PIM0寄存器和PIM1寄存器的对应位置“1”，切换到TTL输入缓冲器。 V_{IH} 和 V_{IL} 请参照选择为TTL输入缓冲器时的DC特性。
- ⑥ 允许串行阵列单元的运行，设定为简易I²C模式。
- ⑦ 将PM0寄存器和PM1寄存器的对应位设定为输出模式（能在输出模式中进行数据的输入/输出）。此时，因为输出数据是高电平，所以引脚处于Hi-Z状态。

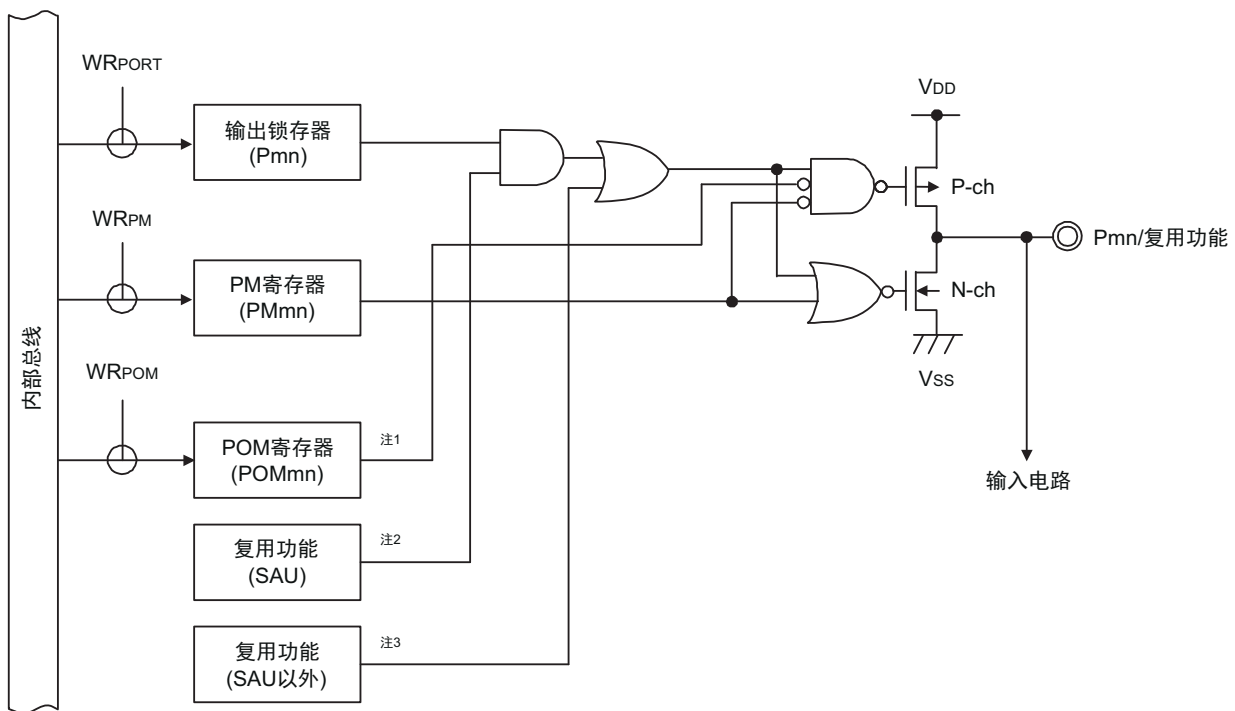
4.5 使用复用功能时的寄存器设定

4.5.1 使用复用功能时的基本思想

首先，必须对与模拟输入复用的引脚，通过 ADPC 寄存器或者端口模式控制寄存器（PMCxx）设定是用作模拟输入还是用作数字输入 / 输出。

用作数字输入 / 输出的引脚输出电路的基本结构如图 4-9 所示。与端口的输出锁存器输出复用的 SAU 功能的输出被输入到 AND 门，AND 门的输出被输入到 OR 门，OR 门的其他输入连接复用的非 SAU 功能（TAU、时钟 / 蜂鸣器的输出、IICA 等）的输出。当将这样的引脚用作端口功能或者复用功能时，不使用的复用功能不能影响要使用的功能的输出。此时的设定基本思想如表 4-4 所示。

图 4-9 引脚的输出电路的基本结构



- 注 1. 当没有 POM 寄存器时，此信号为低电平（0）。
 2. 当没有复用功能时，此信号为高电平（1）。
 3. 当没有复用功能时，此信号为低电平（0）。

备注 m: 端口号 (m=0 ~ 7、12 ~ 14)、n: 位号 (n=0 ~ 7)

表 4-4 设定的基本思想

使用的引脚输出功能	不使用的复用功能的输出设定		
	端口输出功能	SAU 的输出功能	SAU 以外的输出功能
端口输出功能	—	高电平输出 (1)	低电平输出 (0)
SAU 的输出功能	高电平 (1)	—	低电平输出 (0)
SAU 以外的输出功能	低电平 (0)	高电平输出 (1)	低电平输出 (0) 注

注 因为 1 个引脚有可能复用多个 SAU 以外的输出功能，所以需要将不使用的复用功能的输出置为低电平（0）。有关具体的设定方法，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

当不使用引脚复用功能的输出时，必须进行以下的设定。如果为外围 I/O 重定向功能的对象，就能通过设定外围 I/O 重定向寄存器（PIOR），将输出切换到其他引脚。从而能使用分配给对象引脚的端口功能和其他复用功能。

(1) SOp=1/TxDq=1（不使用 SAU 的串行输出（SOp/TxDq）时的设定）

当只将 SAU 用作串行输入等而不使用串行输出（SOp/TxDq）时，必须将与不使用的输出对应的串行输出允许寄存器 m（SOEm）的位置“0”（禁止输出），并且将串行输出寄存器 m（SOM）的 SOMn 位置“1”（高电平）。这是和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）

当不使用 SAU 时，必须将串行通道允许状态寄存器 m（SEm）的 bit n（SEmn）置“0”（运行停止状态），将与不使用的输出对应的串行输出允许寄存器 m（SOEm）的位置“0”（禁止输出），并且将串行输出寄存器 m（SOM）的 SOMn 位和 CKOmn 位置“1”（高电平）。这是和初始状态相同的设定。

(3) TOMn=0（不使用 TAU 通道 n 的输出时的设定）

当不使用 TAU 的 TOMn 输出时，必须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”（低电平）。这是和初始状态相同的设定。

(4) SDAAn=0/SCLAn=0（不使用 IICA 时的设定）

当不使用 IICA 时，必须将 IICA 控制寄存器 n0（IICCTLn0）的 IICEn 位置“0”（停止运行）。这是和初始状态相同的设定。

(5) PCLBUZn=0（不使用时钟输出 / 蜂鸣器输出时的设定）

当不使用时钟输出 / 蜂鸣器输出时，必须将时钟输出选择寄存器 n（CKSn）的 PCLOEn 位置“0”（禁止输出）。这是和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。必须按照表 4-5 所示的内容设定要控制端口功能的寄存器。另外，有关表 4-5 中的记载，请参照以下备注。

备注 一： 对象外

×： 忽略

PIORx： 外围 I/O 重定向寄存器

POMxx： 端口输出模式寄存器

PMCxx： 端口模式控制寄存器

PMxx： 端口模式寄存器

Pxx： 端口的输出锁存器

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和输出锁存器设定例子 (1/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚
	功能名称	输入 / 输出						SAU 的输出功能	SAU 以外	
P00	P00	输入	—	×	0	1	×	×	—	○
		输出	—	0	0	0	0/1	TxD1=1	—	
		N 沟道漏极开路输出	—	1	0	0	0/1			
	ANI17	模拟输入	—	×	1	1	×	×	—	○
	TI00	输入	—	×	0	1	×	×	—	○
	TxD1	输出	—	0/1	0	0	1	×	—	○
P01	P01	输入	—	—	0	1	×	×	×	○
		输出	—	—	0	0	0/1	—	TO00=0	
	ANI16	模拟输入	—	—	1	1	×	×	×	○
	TO00	输出	—	—	0	0	0	—	×	○
	RxD1	输入	—	—	0	1	×	—	×	○
P10	P10	输入	—	×	—	1	×	×	×	○
		输出	—	0	—	0	0/1	SCK00/ SCL00=1	(TO07)=0	
		N 沟道漏极开路输出	—	1	—	0	0/1			
	SCK00	输入	PIOR1=0	×	—	1	×	×	×	○
		输出	PIOR1=0	0/1	—	0	1	×	(TO07)=0	○
	SCL00	输出	PIOR1=0	0/1	—	0	1	×	(TO07)=0	○
	(TI07)	输入	PIOR0=1	×	—	1	×	×	×	○
(TO07)	输出	PIOR0=1	0	—	0	0	SCK00/ SCL00=1	×	○	

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (2/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚	
	功能名称	输入 / 输出						SAU 的 输出功能	SAU 以外		
P11	P11	输入	—	×	—	1	×	×	×	○	
		输出	—	0	—	0	0/1	SDA00=1	(TO06)=0		
		N 沟道漏极 开路输出	—	1	—	0	0/1	SDA00=1	(TO06)=0		
	SI00	输入	PIOR1=0	×	—	1	×	×	×	○	
	RxD0	输入	PIOR1=0	×	—	1	×	×	×	○	
	SDA00	输入 / 输出	PIOR1=0	1	—	0	1	×	(TO06)=0	○	
	(TI06) (TO06)	输入 输出	PIOR0=1 PIOR0=1	×	— 0	1 0	×	×	×	×	○ ○
P12	P12	输入	—	×	—	1	×	—	—	○	
		输出	—	0	—	0	0/1	SO00/TxD0=1	(TO05)=0		
		N 沟道漏极 开路输出	—	1	—	0	0/1				
	SO00	输出	PIOR1=0	0/1	—	0	1	×	(TO05)=0	○	
	TxD0	输出	PIOR1=0	0/1	—	0	1	×	(TO05)=0	○	
	(TI05) (TO05)	输入 输出	PIOR0=1 PIOR0=1	×	— 0	1 0	×	×	×	×	○ ○
	P13	P13	输入	—	×	—	1	×	×	×	○
输出			—	0	—	0	0/1	TxD2/SO20=1	(TO04)=0 (SDAA0)=0		
N 沟道漏极 开路输出			—	1	—	0	0/1				
TxD2		输出	PIOR1=0	0/1	—	0	1	×	(TO04)=0 (SDAA0)=0	○	
SO20		输出	PIOR1=0	0/1	—	0	1	×	(TO04)=0 (SDAA0)=0	○	
(SDAA0)		输入 / 输出	PIOR2=1	1	—	0	0	TxD2/SO20=1	(TO04)=0	○	
(TI04) (TO04)		输入 输出	PIOR0=1 PIOR0=1	×	— 0	1 0	×	×	×	×	○ ○
P14		P14	输入	—	×	—	1	×	×	×	○
	输出		—	0	—	0	0/1	SDA20=1	(TO03)=0 (SCLA0)=0		
	N 沟道漏极 开路输出		—	1	—	0	0/1				
	RxD2	输入	PIOR1=0	×	—	1	×	×	×	○	
	SI20	输入	PIOR1=0	×	—	1	×	×	×	○	
	SDA20	输入 / 输出	PIOR1=0	1	—	0	1	×	(TO03)=0 (SCLA0)=0	○	
	(SCLA0)	输入 / 输出	PIOR2=1	1	—	0	0	SDA20=1	(TO03)=0	○	
	(TI03) (TO03)	输入 输出	PIOR0=1 PIOR0=1	×	— 0	1 0	×	×	×	×	○ ○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (3/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚
	功能名称	输入 / 输出						SAU 的 输出功能	SAU 以外	
P15	P15	输入	—	×	—	1	×	×	×	○
		输出	—	0	—	0	0/1	SCK20/ SCL20=1	PCLBUZ1=0 (TO02)=0	
		N 沟道漏极 开路输出	—	1	—	0	0/1			
	PCLBUZ1	输出	—	0	—	0	0	SCK20/ SCL20=1	(TO02)=0	○
	SCK20	输入	PIOR1=0	×	—	1	×	×	×	○
		输出	PIOR1=0	0/1	—	0	1	×	PCLBUZ1=0 (TO02)=0	○
	SCL20	输出	PIOR1=0	0/1	—	0	1	×	PCLBUZ1=0 (TO02)=0	○
	(TI02)	输入	PIOR0=1	×	—	1	×	×	×	○
(TO02)	输出	PIOR0=1	0	—	0	0	SCK20/ SCL20=1	PCLBUZ1=0	○	
P16	P16	输入	—	—	—	1	×	×	×	○
		输出	—	—	—	0	0/1	×	TO01=0	
	TI01	输入	—	—	—	1	×	×	×	○
	TO01	输出	—	—	—	0	0	×	×	○
	INTP5 (RxD0)	输入	— PIOR1=1	— —	— —	1 1	×	×	×	○
P17	P17	输入	—	×	—	1	×	×	×	○
		输出	—	0	—	0	0/1	(TxD0)=1	TO02=0	
		N 沟道漏极 开路输出	—	1	—	0	0/1			
	TI02	输入	PIOR0=0	×	—	1	×	×	×	○
	TO02	输出	PIOR0=0	0	—	0	0	(TxD0)=1	×	○
	(TxD0)	输出	PIOR1=1	0/1	—	0	1	×	TO02=0	○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (4/8)

引脚名	使用的功能		ADPC	ADM2	PMxx	Pxx	32 引脚
	功能名称	输入 / 输出					
P20	P20	输入	ADPC=01H	×	1	×	○
		输出	ADPC=01H	×	0	0/1	
	ANI0	模拟输入	ADPC=00H/02H ~ 0FH	00x0xx0x、 10x0xx0x	1	×	○
	AV _{REFP}	基准电压输入	ADPC=00H/02H ~ 0FH	01x0xx0x	1	×	○
P21	P21	输入	ADPC=01H/02H	×	1	×	○
		输出	ADPC=01H/02H	×	0	0/1	
	ANI1	模拟输入	ADPC=00H/03H ~ 0FH	xx00xx0x	1	×	○
	AV _{REFM}	基准电压输入	ADPC=00H/03H ~ 0FH	xx10xx0x	1	×	○
P22	P22	输入	ADPC=01H ~ 03H	×	1	×	○
		输出	ADPC=01H ~ 03H	×	0	0/1	
	ANI2	模拟输入	ADPC=00H/04H ~ 0FH	×	1	×	○
P23	P23	输入	ADPC=01H ~ 04H	×	1	×	○
		输出	ADPC=01H ~ 04H	×	0	0/1	
	ANI3	模拟输入	ADPC=00H/05H ~ 0FH	×	1	×	○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (5/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚
	功能名称	输入 / 输出						SAU 的 输出功能	SAU 以外	
P30	P30	输入	—	—	—	1	×	×	×	○
		输出	—	—	—	0	0/1	SCK11/ SCL11=1	×	
	INTP3	输入	—	—	—	1	×	×	×	○
	SCK11	输入	—	—	—	1	×	×	×	○
		输出	—	—	—	0	1	×	×	○
SCL11	输出	—	—	—	0	1	×	×	○	
P31	P31	输入	—	—	—	1	×	—	×	○
		输出	—	—	—	0	0/1	—	TO03=0 PCLBUZ0=0	
	TI03	输入	PIOR0=0	—	—	1	×	—	×	○
	TO03	输出	PIOR0=0	—	—	0	0	—	PCLBUZ0=0	○
	INTP4	输入	—	—	—	1	×	—	×	○
	PCLBUZ0	输出	—	—	—	0	0	—	TO03=0	○
P40	P40	输入	—	—	—	1	×	—	—	○
		输出	—	—	—	0	0/1	—	—	
P50	P50	输入	—	—	—	1	×	×	—	○
		输出	—	0	—	0	0/1	SDA11=1	—	
		N 沟道漏极 开路输出	—	1	—	0	0/1			
	INTP1	输入	—	×	—	1	×	×	—	○
	SI11	输入	—	×	—	1	×	×	—	○
	SDA11	输入 / 输出	—	1	—	0	1	×	—	○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (6/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚
	功能名称	输入 / 输出						SAU 的输出功能	SAU 以外	
P51	P51	输入	—	—	—	1	×	×	—	○
		输出	—	—	—	0	0/1	SO01=1	—	
	INTP2	输入	—	—	—	1	×	×	—	○
	SO11	输出	—	—	—	0	1	×	—	○
P60	P60	输入	—	—	—	1	×	—	×	○
		N 沟道漏极开路输出 (6V 耐压)	—	—	—	0	0/1	—	SCLA0=0	
	SCLA0	输入 / 输出	PIOR2=0	—	—	0	0	—	×	○
P61	P61	输入	—	—	—	1	×	—	×	○
		N 沟道漏极开路输出 (6V 耐压)	—	—	—	0	0/1	—	SDAA0=0	
	SDAA0	输入 / 输出	PIOR2=0	—	—	0	0	—	×	○
P62	P62	输入	—	—	—	1	×	—	×	○
		N 沟道漏极开路输出 (6V 耐压)	—	—	—	0	0/1	—	×	
P70	P70	输入	—	—	—	1	×	×	—	○
		输出	—	—	—	0	0/1	×	—	
P120	P120	输入	—	—	0	1	×	—	—	○
		输出	—	—	0	0	0/1	—	—	
	ANI19	模拟输入	—	—	1	1	×	—	—	○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (7/8)

引脚名	使用的功能		CMC		Pxx	32 引脚
	功能名称	输入 / 输出	(EXCLK、OSCSEL)			
P121	P121	输入	00xx/10xx/11xx		×	○
	X1	—	01xx		—	○
P122	P122	输入	00xx/10xx		×	○
	X2	—	01xx		—	○
	EXCLK	输入	11xx		—	○

表 4-5 使用引脚功能时的寄存器和输出锁存器设定例子 (8/8)

引脚名	使用的功能		PIORx	POMxx	PMCxx	PMxx	Pxx	复用功能的输出		32 引脚
	功能名称	输入 / 输出						SAU 的输出功能	SAU 以外	
P137	P137	输入	—	—	—	—	×	—	—	○
	INTP0	输入	—	—	—	—	×	—	—	○
P147	P147	输入	—	—	0	1	×	—	—	○
		输出	—	—	0	0	0/1	—	—	
	ANI18	模拟输入	—	—	1	1	×	—	—	○

4.6 使用端口功能时的注意事项

4.6.1 有关对端口寄存器 n (Pn) 的位存储器操作指令的注意事项

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

因此，建议在将任意的端口从输入模式切换为输出模式前重新写输出锁存器的值。

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

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

在本产品内部，位存储器操作指令按照以下顺序执行：

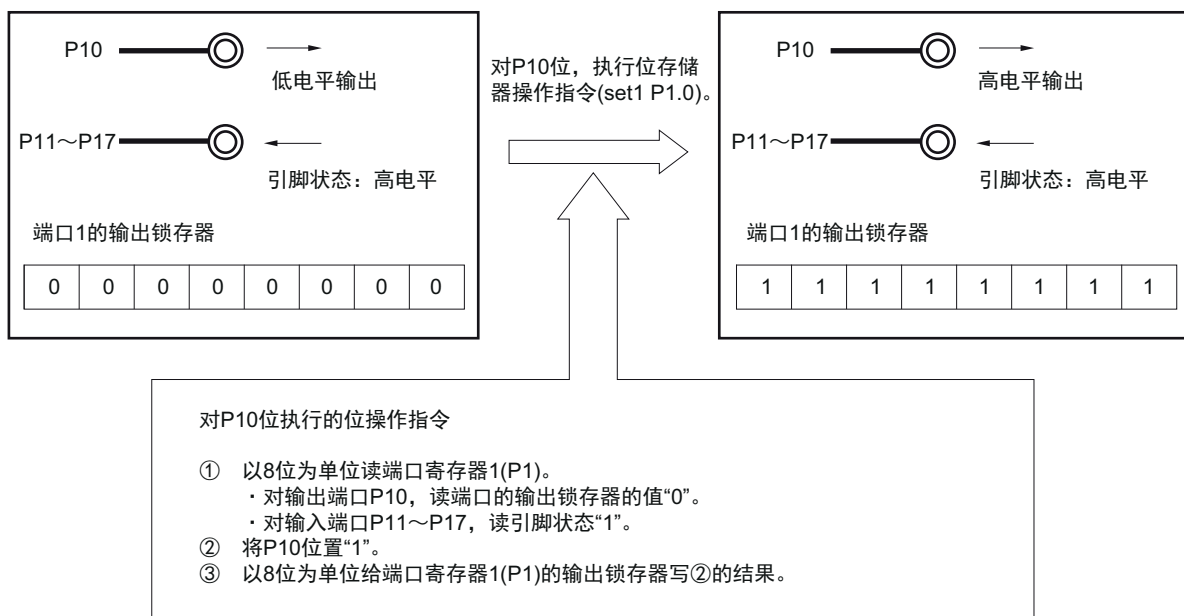
- <1> 以 8 位为单位读 Pn 寄存器。
- <2> 操作 1 位对象位。
- <3> 以 8 位为单位写 Pn 寄存器。

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

通过 <2> 的操作，值变为“FFH”。

通过 <3> 的操作，给输出锁存器写“FFH”。

图 4-10 位存储器操作指令（P10 的情况）



4.6.2 有关引脚设定的注意事项

对于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置初始值（避免输出的冲突）。对于根据外围 I/O 重定向寄存器（PIOR）的设定来分配的功能也同样。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。

为了降低功耗，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 3 种系统时钟和时钟振荡电路。

(1) 主系统时钟

① X1 振荡电路

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

② 高速内部振荡器

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

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

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

电源电压	振荡频率 (MHz)								
	1	2	3	4	6	8	12	16	24
$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	○	○	○	○	○	○	○	○	○
$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$	○	○	○	○	○	○	○	○	—
$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	○	○	○	○	○	○	—	—	—
$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$	○	○	○	○	—	—	—	—	—

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

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

主系统时钟能使用的频率范围因电源电压 V_{DD} 而不同，需要通过选项字节（000C2H）的 CMODE0 和 CMODE1 设定闪存的工作电压模式（参照“第 22 章 选项字节”）。

(2) 低速内部振荡器时钟（低速内部振荡器）

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

不能将低速内部振荡器时钟用作 CPU 时钟。

只有以下外围硬件才能通过低速内部振荡器时钟运行：

- 看门狗定时器
- 12 位间隔定时器

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

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

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

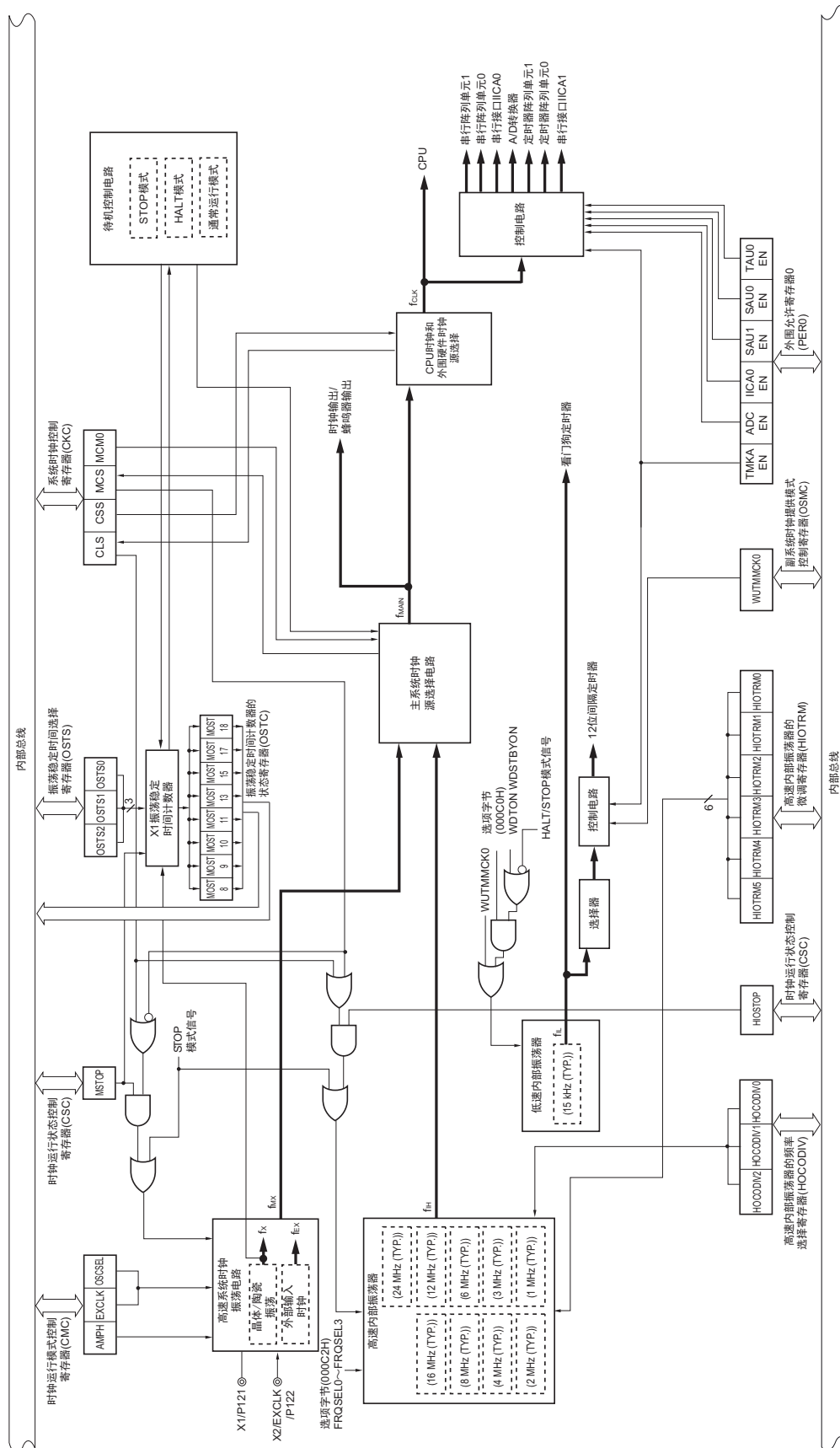
5.2 时钟发生电路的结构

时钟发生电路由以下硬件构成。

表 5-1 时钟发生电路的结构

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

图 5-1 时钟发生电路的框图



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

5.3 控制时钟发生电路的寄存器

通过以下 9 种寄存器控制时钟发生电路。

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

注意 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

5.3.1 时钟运行模式控制寄存器 (CMC)

这是设定 X1/P121 引脚和 X2/EXCLK/P122 引脚的运行模式以及选择振荡电路增益的寄存器。

在解除复位后，只能通过 8 位存储器操作指令写 1 次 CMC 寄存器。能通过 8 位存储器操作指令读此寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 5-2 时钟运行模式控制寄存器 (CMC) 的格式

地址: FFFA0H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	AMPH

EXCLK	OSCSEL	高速系统时钟 引脚的运行模式	X1/P121 引脚	X2/EXCLK/P122 引脚
0	0	输入端口模式	输入端口	
0	1	X1 振荡模式	连接晶体或者陶瓷谐振器。	
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. 当 X1 时钟振荡频率超过 10MHz 时，必须将 AMPH 位置 “1”。
4. 必须在解除复位后并且在选择 f_{IH} 作为 f_{CLK} 的状态（将 f_{CLK} 切换为 f_{MX} 前的状态）下设定 AMPH 位。
5. 系统时钟的频率上限为 24MHz，但是 X1 振荡电路的频率上限为 20MHz。

备注 f_X : X1 时钟振荡频率

5.3.2 系统时钟控制寄存器（CKC）

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

图 5-3 系统时钟控制寄存器（CKC）的格式

地址: FFFA4H	复位后: 00H	R/W 注						
符号	7	6	5	4	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

MCS	主系统时钟 (f_{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。

备注 f_{IH} : 高速内部振荡器的时钟频率
 f_{MX} : 高速系统时钟频率
 f_{MAIN} : 主系统时钟频率

注意 必须将 bit7、bit6 和 bit0 ~ 3 置 “0”。

5.3.3 时钟运行状态控制寄存器（CSC）

这是控制高速系统时钟和高速内部振荡器（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“C0H”。

图 5-4 时钟运行状态控制寄存器（CSC）的格式

地址：FFFA1H 复位后：C0H R/W

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

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

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

- 注意 1. 在解除复位后，必须在设定时钟运行模式控制寄存器（CMC）后设定 CSC 寄存器。
- 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。
 - 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。
 - 不能通过 CSC 寄存器停止被选择为 CPU/ 外围硬件时钟（ f_{CLK} ）的时钟。
 - 有关用于停止时钟振荡（外部时钟输入无效）的寄存器标志设定和停止前的条件，请参照表 5-2。必须在停止时钟振荡前确认条件，然后停止时钟振荡。

表 5-2 时钟停止方法

时钟	时钟停止前的条件（外部时钟输入无效）	CSC 寄存器的标志设定
X1 时钟	CPU/ 外围硬件时钟以高速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 $2^8/f_X$	小于 $25.6\mu\text{s}$	小于 $12.8\mu\text{s}$
1	0	0	0	0	0	0	0	至少 $2^8/f_X$	至少 $25.6\mu\text{s}$	至少 $12.8\mu\text{s}$
1	1	0	0	0	0	0	0	至少 $2^9/f_X$	至少 $51.2\mu\text{s}$	至少 $25.6\mu\text{s}$
1	1	1	0	0	0	0	0	至少 $2^{10}/f_X$	至少 $102\mu\text{s}$	至少 $51.2\mu\text{s}$
1	1	1	1	0	0	0	0	至少 $2^{11}/f_X$	至少 $204\mu\text{s}$	至少 $102\mu\text{s}$
1	1	1	1	1	0	0	0	至少 $2^{13}/f_X$	至少 $819\mu\text{s}$	至少 $409\mu\text{s}$
1	1	1	1	1	1	0	0	至少 $2^{15}/f_X$	至少 3.27ms	至少 1.63ms
1	1	1	1	1	1	1	0	至少 $2^{17}/f_X$	至少 13.1ms	至少 6.55ms
1	1	1	1	1	1	1	1	至少 $2^{18}/f_X$	至少 26.2ms	至少 13.1ms

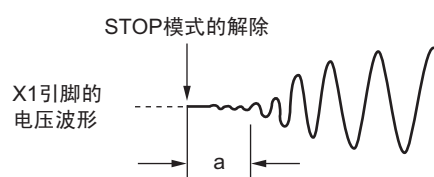
注意 1. 在经过上述时间后，各位从 MOST8 位开始依次变为“1”并且保持“1”的状态。

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

在以下情况下，OSTS 寄存器的振荡稳定时间的设定值必须大于通过 OSTC 寄存器确认的计数值。

- 当 CPU 时钟为高速内部振荡器时钟并且开始 X1 时钟的振荡时
- 当 CPU 时钟为高速内部振荡器时钟并且在 X1 时钟振荡的状态下转移到 STOP 模式后解除 STOP 模式时（因此必须注意，解除 STOP 模式后的 OSTC 寄存器只设定 OSTS 寄存器所设振荡稳定时间内的状态）

3. X1 时钟的振荡稳定时间不包含时钟开始振荡前的时间（下图 a）。



备注 f_X : X1 时钟振荡频率

5.3.5 振荡稳定时间选择寄存器（OSTS）

这是选择 X1 时钟的振荡稳定时间的寄存器。

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

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

通过 8 位存储器操作指令设定 OSTS 寄存器。

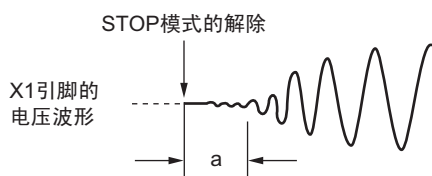
在产生复位信号后，此寄存器的值变为“07H”。

图 5-6 振荡稳定时间选择寄存器（OSTS）的格式

地址：FFFA3H	复位后：07H	R/W						
符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择	
				$f_X=10\text{MHz}$	$f_X=20\text{MHz}$
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204 μs	102 μs
1	0	0	$2^{13}/f_X$	819 μs	409 μs
1	0	1	$2^{15}/f_X$	3.27ms	1.63ms
1	1	0	$2^{17}/f_X$	13.1ms	6.55ms
1	1	1	$2^{18}/f_X$	26.2ms	13.1ms

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



备注 f_X : X1 时钟振荡频率

5.3.6 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，以降低功耗和噪声。

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

- 12位间隔定时器
- A/D转换器
- 串行接口 IICA0
- 串行阵列单元1
- 串行阵列单元0
- 定时器阵列单元0

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 5-7 外围允许寄存器 0 (PER0) 的格式 (1/2)

地址: F00F0H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TMKAEN	12 位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 • 不能写 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。

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

IICA0EN	串行接口 IICA0 的输入时钟的控制
0	停止提供输入时钟。 • 不能写串行接口 IICA0 使用的 SFR。 • 串行接口 IICA0 处于复位状态。
1	提供输入时钟。 • 能读写串行接口 IICA0 使用的 SFR。

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	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAU1EN	串行阵列单元 1 的输入时钟的控制	
0	停止提供输入时钟。 • 不能写串行阵列单元 1 使用的 SFR。 • 串行阵列单元 1 处于复位状态。	
1	提供输入时钟。 • 能读写串行阵列单元 1 使用的 SFR。	

SAU0EN	串行阵列单元 0 的输入时钟的控制	
0	停止提供输入时钟。 • 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	提供输入时钟。 • 能读写串行阵列单元 0 使用的 SFR。	

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

5.3.7 副系统时钟提供模式控制寄存器 (OSMC)

OSMC 寄存器控制 12 位间隔定时器运行时钟的提供。

要使用 12 位间隔定时器时, 必须事先将 WUTMMCK0 位置“1”, 并且不能在定时器停止前将 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 5-8 副系统时钟提供模式控制寄存器 (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	12 位间隔定时器运行时钟的提供	
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频率的寄存器。但是，能选择的频率因选项字节（000C2H）的 FRQSEL3 位的值而不同。

通过 8 位存储器操作指令设定 HOCODIV 寄存器。

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

图 5-9 高速内部振荡器的频率选择寄存器（HOCODIV）的格式

地址：F00A8H	复位后：选项字节（000C2H）的 FRQSEL2 ~ FRQSEL0 位的设定值							R/W
符号	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

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

注意 1. 在更改频率前后，必须将高速内部振荡器的频率选择寄存器（HOCODIV）设定在选项字节（000C2H）所设闪存运行模式的可行工作电压范围内。

选项字节（000C2H）的值		闪存运行模式	工作频率范围	工作电压范围
CMODE1	CMODE0			
0	0	LV（低电压主）模式	1MHz ~ 4MHz	1.6V ~ 5.5V
1	0	LS（低速主）模式	1MHz ~ 8MHz	1.8V ~ 5.5V
1	1	HS（高速主）模式	1MHz ~ 16MHz	2.4V ~ 5.5V
			1MHz ~ 24MHz	2.7V ~ 5.5V

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

5.3.9 高速内部振荡器的微调寄存器 (HIOTRM)

这是校正高速内部振荡器精度的寄存器。

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

通过 8 位存储器操作指令设定 HIOTRM 寄存器。

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

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

图 5-10 高速内部振荡器的微调寄存器 (HIOTRM) 的格式

地址: F00A0H 复位后: 不定值注 R/W

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

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

注 复位值是发货时的调整值。

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

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

5.4 系统时钟振荡电路

5.4.1 X1 振荡电路

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

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

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

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

而且，当也不用作输入端口时，请参照“表 2-2 各未使用引脚的处理”。

X1 振荡电路的外接电路例子如图 5-11 所示。

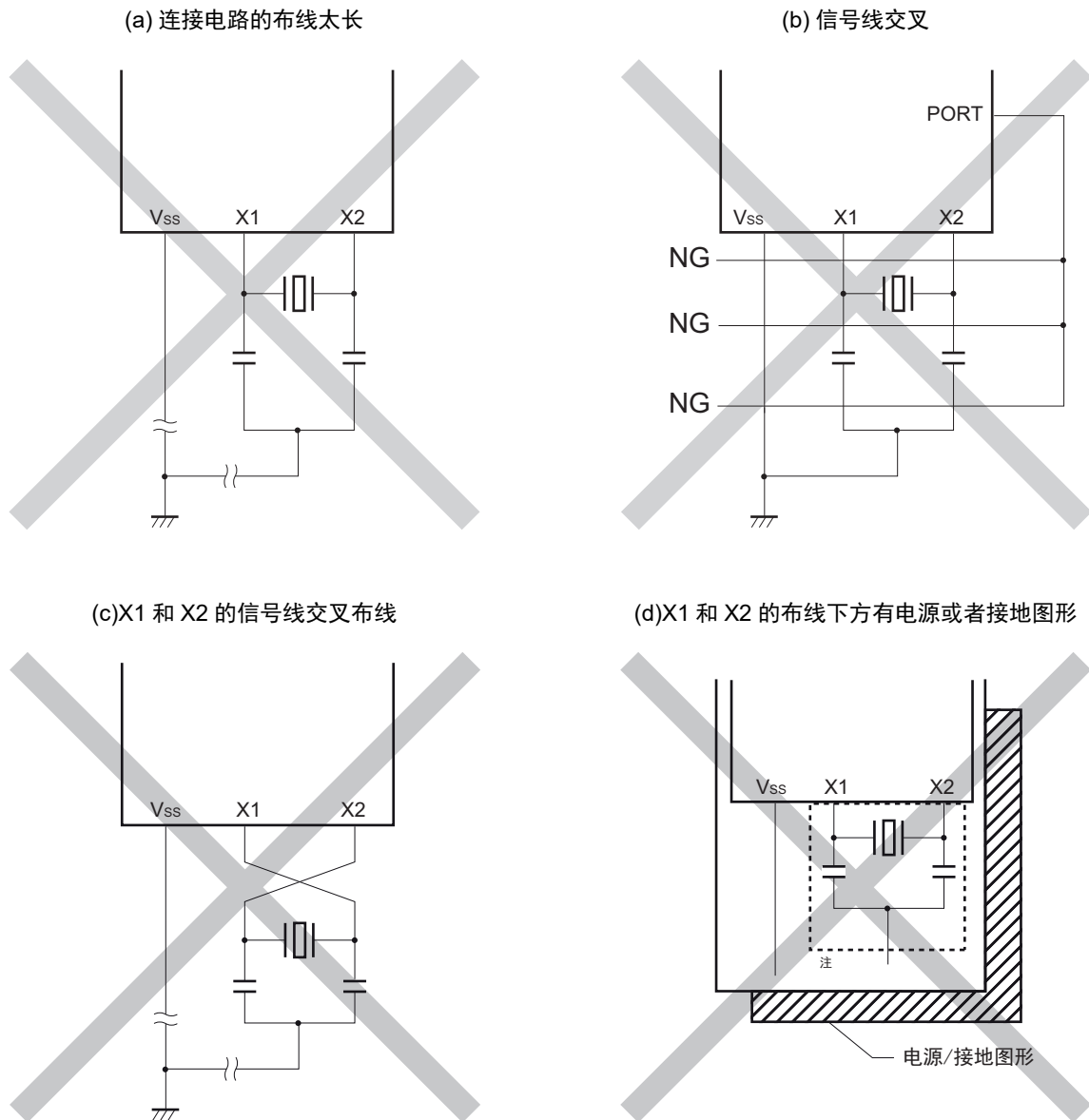
图 5-11 X1 振荡电路的外接电路例子



注意事项如下页所示。

不正确的谐振器连接例子如图 5-12 所示。

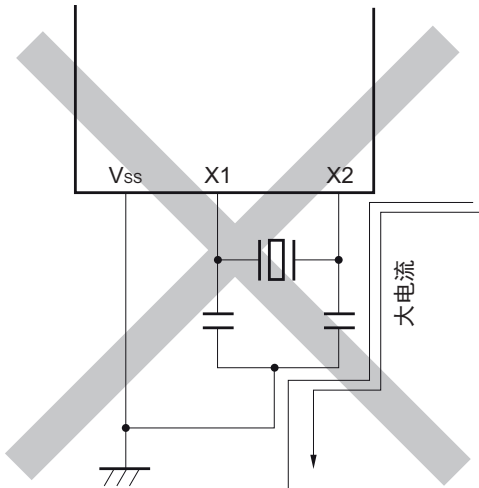
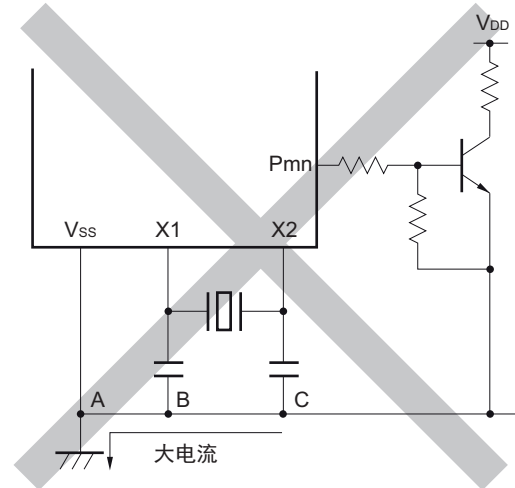
图 5-12 不正确的谐振器连接例子 (1/2)



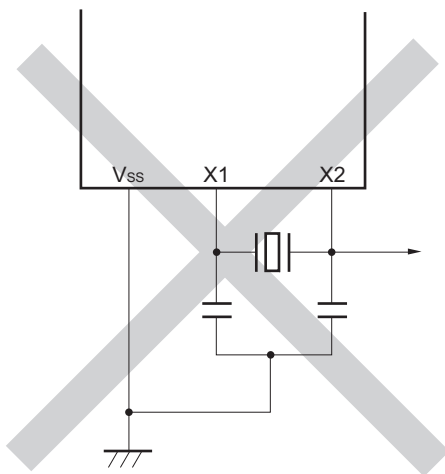
注 在多层板或者双面板中，不能在 X1 引脚、X2 引脚和谐振器的布线区（图中虚线部分）下方配置电源或者接地图形。布线不能产生电容成分而影响振荡特性。

图 5-12 不正确的谐振器连接例子 (2/2)

(e) 有变化的大电流接近信号线

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

(g) 取出信号



5.4.2 高速内部振荡器

本产品内置高速内部振荡器。能通过选项字节（000C2H）从 24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 和 1MHz 中选择频率。能通过时钟运行状态控制寄存器（CSC）的 bit0（HIOSTOP）控制振荡。

在解除复位后，高速内部振荡器自动开始振荡。

5.4.3 低速内部振荡器

本产品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，但是不能用作 CPU 时钟。

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

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

5.5 时钟发生电路的运行

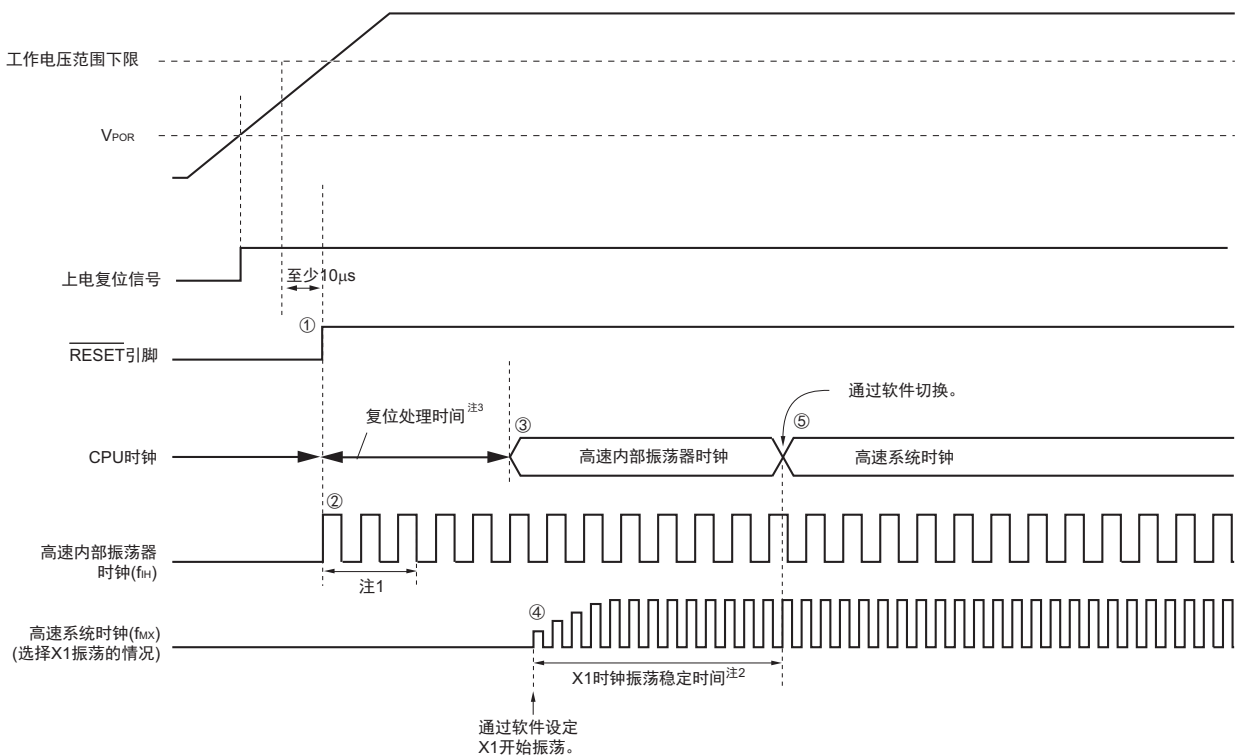
时钟发生电路产生以下所示各种时钟，并且控制待机模式等 CPU 的运行模式（参照图 5-1）。

- 主系统时钟 f_{MAIN}
 - 高速系统时钟 f_{MX}
 - X1 时钟 f_{X}
 - 外部主系统时钟 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}

本产品解除复位后，CPU 通过高速内部振荡器的输出开始运行。

接通电源时的时钟发生电路的运行如图 5-13 所示。

图 5-13 接通电源时的时钟发生电路的运行



- ① 在接通电源后，通过上电复位（POR）电路产生内部复位信号。但是，在达到“27.4 AC特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态（上图是使用外部复位时的例子）。
- ② 如果解除复位，高速内部振荡器就自动开始振荡。
- ③ 在解除复位后，进行电压稳定等待和复位处理，然后CPU以高速内部振荡器时钟开始运行。
- ④ 必须通过软件设定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. 有关复位处理时间，请参照“第18章 上电复位电路”。

注意 如果使用EXCLK引脚输入的外部时钟，就不需要振荡稳定等待时间。

5.6 时钟控制

5.6.1 高速内部振荡器的设定例子

在解除复位后，CPU/ 外围硬件时钟 (f_{CLK}) 一定以高速内部振荡器时钟运行。能通过选项字节 (000C2H) 的 FRQSEL0 ~ FRQSEL3 位，从 24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz 和 1MHz 中选择高速内部振荡器的频率。另外，也能通过高速内部振荡器的频率选择寄存器 (HOCODIV) 更改频率。

【选项字节的设定】

地址：000C2H

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

CMODE1	CMODE0	闪存运行模式设定	
0	0	LV (低电压主) 模式	$V_{DD}=1.6V \sim 5.5V@1MHz \sim 4MHz$
1	0	LS (低速主) 模式	$V_{DD}=1.8V \sim 5.5V@1MHz \sim 8MHz$
1	1	HS (高速主) 模式	$V_{DD}=2.4V \sim 5.5V@1MHz \sim 16MHz$ $V_{DD}=2.7V \sim 5.5V@1MHz \sim 24MHz$
上述以外		禁止设定。	

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的频率
0	0	0	0	24MHz
1	0	0	1	16MHz
0	0	0	1	12MHz
1	0	1	0	8MHz
0	0	1	0	6MHz
1	0	1	1	4MHz
0	0	1	1	3MHz
1	1	0	0	2MHz
1	1	0	1	1MHz
上述以外				禁止设定。

【高速内部振荡器的频率选择寄存器（HOCODIV）的设置】

地址：F00A8H

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

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

5.6.2 X1 振荡电路的设定例子

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

【寄存器的设定】 必须按照①~⑤的顺序设定寄存器。

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

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

- ② 通过 OSTS 寄存器选择解除 STOP 模式时的 X1 振荡电路的振荡稳定时间。

例) 要通过 10MHz 谐振器至少等待 102 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 102 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

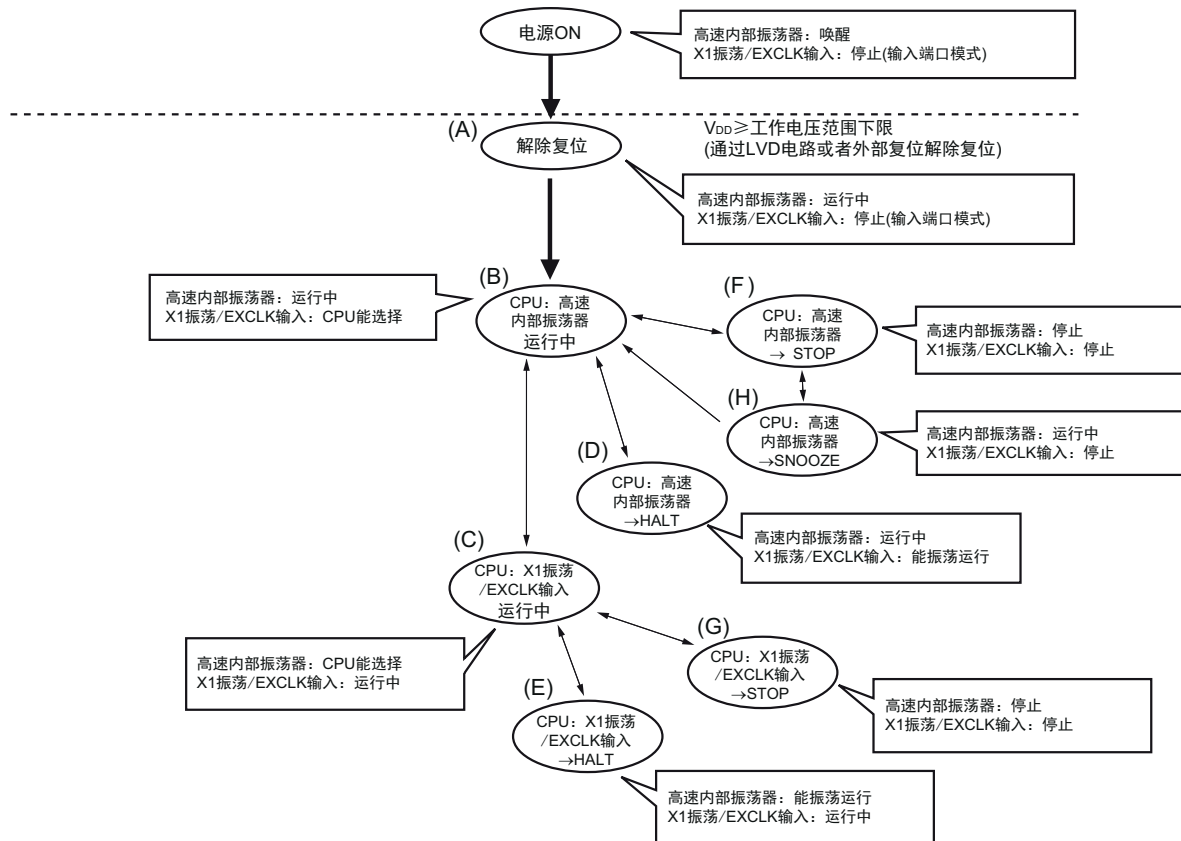
- ⑤ 通过 CKC 寄存器的 MCM0 位将 X1 振荡时钟设定为 CPU/ 外围硬件时钟。

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

5.6.3 CPU 时钟的状态转移图

本产品的 CPU 时钟状态转移图如图 5-14 所示。

图 5-14 CPU 时钟的状态转移图



CPU 时钟的状态转移和 SFR 寄存器的设定例子等如表 5-3 所示。

表 5-3 CPU 时钟的状态转移和 SFR 寄存器的设定例子 (1/3)

(1) 在解除复位(A)后, CPU 转移到高速内部振荡器时钟运行(B)。

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

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

(SFR 寄存器的设定顺序) →

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	OSCSEL	AMPH		MSTOP		MCM0
(A)→(B)→(C) (X1 时钟: $1\text{MHz} \leq f_X \leq 10\text{MHz}$)	0	1	0	注 2	0	需要确认	1
(A)→(B)→(C) (X1 时钟: $10\text{MHz} < f_X \leq 20\text{MHz}$)	0	1	1	注 2	0	需要确认	1
(A)→(B)→(C) (外部主时钟)	1	1	×	注 2	0	不需要 确认	1

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

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

- 期待的振荡稳定时间计数器的状态寄存器 (OSTC) 的振荡稳定时间 \leq OSTS 寄存器设定的振荡稳定时间

注意 必须在电源电压达到设定的时钟可运行电压 (参照“第 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”)后设定时钟。

备注 1. ×: 忽略

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

表 5-3 CPU 时钟的状态转移和 SFR 寄存器的设定例子 (2/3)

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

(SFR 寄存器的设定顺序) →

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

如果已设定就不需要。
在高速系统时钟运行中不需要。

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

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

- 期待的振荡稳定时间计数器的状态寄存器 (OSTC) 的振荡稳定时间 \leq OSTS 寄存器设定的振荡稳定时间

注意 必须在电源电压达到设定的时钟可运行电压 (参照“第 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”)后设定时钟。

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

(SFR 寄存器的设定顺序) →

状态转移	SFR 寄存器的设定标志		振荡精度稳定的等待	CKC 寄存器 MCM0
	CSC 寄存器 HIOSTOP	HIOSTOP		
(C)→(B)	0		$18\mu\text{s} \sim 65\mu\text{s}$	0

在高速内部振荡器时钟运行中不需要。

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

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

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

备注 1. ×: 忽略

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

表 5-3 CPU 时钟的状态转移和 SFR 寄存器的设定例子 (3/3)

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

(设定顺序) →

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

- (7) CPU从STOP模式(F)转移到SNOOZE模式(H)。

有关从STOP模式转移到SNOOZE模式的详细设定，请参照“10.8 SNOOZE模式功能”、“11.5.7 SNOOZE模式功能”和“11.6.3 SNOOZE模式功能”。

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

5.6.4 CPU 时钟状态转移前的条件和转移后的处理

CPU 时钟状态转移前的条件和转移后的处理如下所示。

表 5-4 有关 CPU 时钟的状态转移

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	在确认了 CPU 被切换到转移后的时钟后，如果停止高速内部振荡器的振荡 (HIOSTOP=1)，就能减小工作电流。
	外部主系统时钟	将 EXCLK 引脚输入的外部时钟置为有效。 • OSCSEL=1, EXCLK=1, MSTOP=0	
X1 时钟	高速内部振荡器 时钟	允许高速内部振荡器振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	在确认了 CPU 被切换到转移后的时钟后，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振荡。 • HIOSTOP=0 • 经过振荡稳定时间后	在确认了 CPU 被切换到转移后的时钟后，能将外部主系统时钟的输入置为无效 (MSTOP=1)。
	X1 时钟	不能转移。	—

5.6.5 CPU 时钟和主系统时钟的切换所需时间

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

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

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 $f_{IH} \leftrightarrow f_{MX}$ 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 ($f_{MAIN}=f_{IH}$)	$f_{MX} \geq f_{IH}$		2 个时钟
	$f_{MX} < f_{IH}$		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	
	$f_{MX} < f_{IH}$	2 个时钟	

备注 1. 时钟数是切换前的 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 $f_{IH}=8\text{MHz}$ 、 $f_{MX}=10\text{MHz}$ 振荡的情况）
 $2f_{MX}/f_{IH}=2(10/8)=2.5 \rightarrow 3$ 个时钟

5.6.6 时钟振荡停止前的条件

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

必须在停止时钟振荡前确认条件，然后停止时钟振荡。

表 5-6 时钟振荡停止前的条件和标志设定

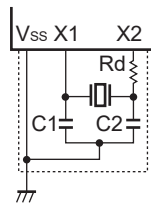
时钟	时钟停止前的条件（外部时钟输入无效）	SFR 寄存器的标志设定
高速内部振荡器时钟	MCS=1 (CPU 以高速系统时钟运行)	HIOSTOP=1
X1 时钟	MCS=0	MSTOP=1
外部主系统时钟	(CPU 以高速内部振荡器时钟运行)	

5.7 谐振器和振荡电路常数

已验证的谐振器及其振荡电路常数（供参考）如下所示。

- 注意 1. 此振荡电路常数是基于谐振器厂商在特定环境下进行评估的参考值。在实际应用中，请委托谐振器厂商给予安装电路后的评估。
在更改其他产品的单片机和电路板时，请委托谐振器厂商再次给予安装电路后的评估。
2. 振荡电压和振荡频率原则上是表示振荡电路的特性。有关 RL78 微控制器的内部工作条件，请在 DC、AC 特性的规格内使用。

图 5-15 外接振荡电路的例子（X1 振荡）



(1) X1 振荡

截至 2013 年 3 月 (1/2)

厂商	谐振器	产品名称注 3	SMD/ 引线	频率 (MHz)	闪存运行 模式注 1	振荡电路常数注 2 (参考)			电压范围 (V)	
						C1(pF)	C2(pF)	Rd(kΩ)	MIN.	MAX.
村田制作 所公司 注 4	陶瓷 谐振器	CSTCC2M00G56-R0	SMD	2.0	LV	(47)	(47)	0	1.6	5.5
		CSTCR4M00G55-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	引线			(15)	(15)	0		
		CSTCC2M00G56-R0	SMD	2.0	LS	(47)	(47)	0	1.8	5.5
		CSTCR4M00G55-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	引线			(15)	(15)	0		
		CSTCR4M19G55-R0	SMD	4.194		(39)	(39)	0		
		CSTLS4M19G53-B0	引线			(15)	(15)	0		
		CSTCR4M91G53-R0	SMD	4.915		(15)	(15)	0		
		CSTLS4M91G53-B0	引线			(15)	(15)	0		
		CSTCR5M00G53-R0	SMD	5.0		(15)	(15)	0		
		CSTLS5M00G53-B0	引线			(15)	(15)	0		
		CSTCR6M00G53-R0	SMD	6.0		(15)	(15)	0		
		CSTLS6M00G53-B0	引线		(15)	(15)	0			
		CSTCE8M00G52-R0	SMD	8.0	(10)	(10)	0			
		CSTLS8M00G53-B0	引线		(15)	(15)	0			
		CSTCE8M38G52-R0	SMD	8.388	HS	(10)	(10)	0	2.4	5.5
		CSTLS8M38G53-B0	引线			(15)	(15)	0		
		CSTCE10M0G52-R0	SMD	10.0		(10)	(10)	0		
		CSTLS10M0G53-B0	引线			(15)	(15)	0		
CSTCE12M0G52-R0	SMD	12.0	(10)	(10)		0				
CSTCE16M0V53-R0	SMD	16.0	(15)	(15)		0				
CSTLS16M0X51-B0	引线		(5)	(5)	0					
CSTCE20M0V51-R0	SMD	20.0	(5)	(5)	0	2.7	5.5			
CSTLS20M0X51-B0	引线		(5)	(5)	0					

注 1. 通过选项字节 (000C2H) 的 CMODE1 位和 CMODE0 位设定闪存运行模式。

2. C1 和 C2 栏的 () 内表示内部电容值。

3. 支持 105°C 的产品名称不同。详细内容请向村田制作所公司 (<http://www.murata.com>) 询问。

4. 在使用此振荡器时, 有关匹配的详细内容, 请向村田制作所公司 (<http://www.murata.com>) 询问。

备注 1. 工作电压范围、CPU 工作频率和运行模式的关系如下所示:

HS (高速主) 模式: $2.7V \leq V_{DD} \leq 5.5V@1MHz \sim 24MHz$

$2.4V \leq V_{DD} \leq 5.5V@1MHz \sim 16MHz$

LS (低速主) 模式: $1.8V \leq V_{DD} \leq 5.5V@1MHz \sim 8MHz$

LV (低电压主) 模式: $1.6V \leq V_{DD} \leq 5.5V@1MHz \sim 4MHz$

2. 有关最新的已得到验证的谐振器及其振荡电路常数 (参考), 请参阅本公司网站 (<http://www.renesas.com>) 对应产品的网页。

截至 2013 年 3 月 (2/2)

厂商	谐振器	产品名称注 2	SMD/ 引线	频率 (MHz)	闪存运行 模式注 1	振荡电路常数 (参考)			电压范围 (V)	
						C1(pF)	C2(pF)	Rd(kΩ)	MIN.	MAX.
日本电波 工业注 3	晶体 振荡器	NX8045GB	SMD	8.0	注 3					
		NX5032GA	SMD	16.0						
		NX3225HA	SMD	20.0						
京瓷晶体 元件公司 注 4	晶体 振荡器	CX8045GB04000D0PPTZ1	SMD	4.0	LV	12	12	0	1.6	5.5
					LS				1.8	5.5
		CX8045GB04915D0PPTZ1	SMD	4.915	LS	12	12	0	1.8	5.5
		CX8045GB08000D0PPTZ1	SMD	8.0		12	12	0		
		CX8045GB10000D0PPTZ1	SMD	10.0	HS	12	12	0	2.4	5.5
		CX3225GB12000B0PPTZ1	SMD	12.0		5	5	0		
		CX3225GB16000B0PPTZ1	SMD	16.0		5	5	0		
CX3225SB20000B0PPTZ1	SMD	20.0	5	5		0	2.7	5.5		
RIVER ELETEC 公司注 5	晶体 振荡器	FCX-03-8.000MHZ-J21140	SMD	8.0	HS	3	3	0	2.4	5.5
		FCX-04C-10.000MHZ-J21139	SMD	10.0		4	4	0		
		FCX-05-12.000MHZ-J21138	SMD	12.0		6	6	0		
		FCX-06-16.000MHZ-J21137	SMD	16.0		4	4	0		

- 注 1. 通过选项字节 (000C2H) 的 CMODE1 位和 CMODE0 位设定闪存运行模式。
2. 此振荡器支持 85°C 为止的产品。有关支持 105°C 的产品, 请向谐振器厂商询问。
3. 在使用此振荡器时, 有关匹配的详细内容, 请向日本电波工业公司 (<http://www.ndk.com/en>) 询问。
4. 在使用此振荡器时, 有关匹配的详细内容, 请向京瓷晶体元件公司公司 (<http://www.kyocera-crystal.jp/eng/index.html>、<http://global.kyocera.com>) 询问。
5. 在使用此振荡器时, 有关匹配的详细内容, 请向 RIVER ELETEC 公司 (<http://www.river-ele.co.jp/english/index.html>) 询问。

备注 1. 工作电压范围、CPU 工作频率和运行模式的关系如下所示:

HS (高速主) 模式: $2.7V \leq V_{DD} \leq 5.5V@1MHz \sim 24MHz$

$2.4V \leq V_{DD} \leq 5.5V@1MHz \sim 16MHz$

LS (低速主) 模式: $1.8V \leq V_{DD} \leq 5.5V@1MHz \sim 8MHz$

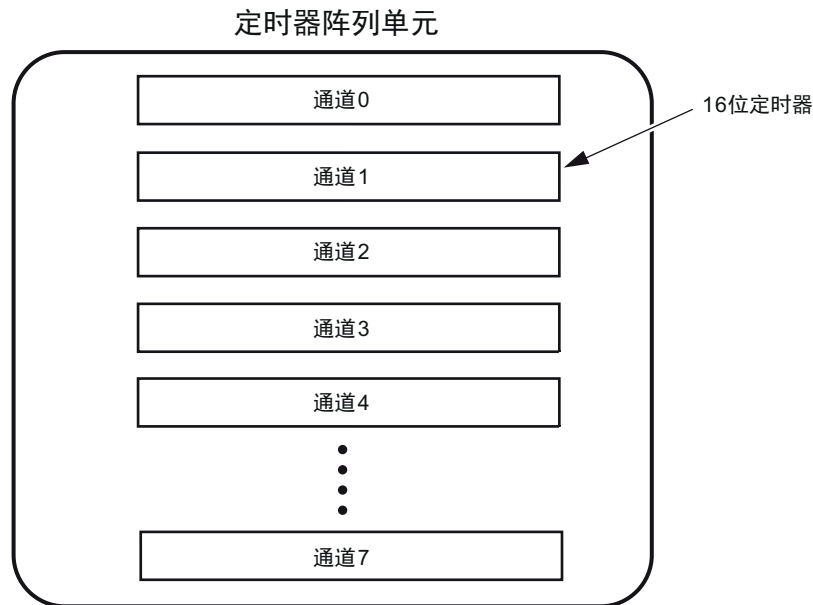
LV (低电压主) 模式: $1.6V \leq V_{DD} \leq 5.5V@1MHz \sim 4MHz$

2. 有关最新的已得到验证的谐振器及其振荡电路常数 (参考), 请参阅本公司网站 (<http://www.renesas.com>) 对应产品的页。

第 6 章 定时器阵列单元

定时器阵列单元有 8 个 16 位定时器。

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



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 (→ 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) • PWM 输出 (→ 参照 6.9.2) • 多重 PWM 输出 (→ 参照 6.9.3)

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）。通道 1 和通道 3 能用作 8 位定时器的功能如下：

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

6.1 定时器阵列单元的功能

定时器阵列单元有以下功能：

6.1.1 独立通道运行功能

独立通道运行功能是不受其他通道运行模式的影响而能独立使用任意通道的功能。

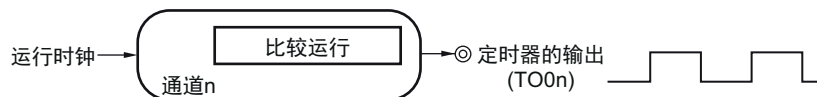
(1) 间隔定时器

能用作以固定间隔产生中断（INTTM0n）的基准定时器。



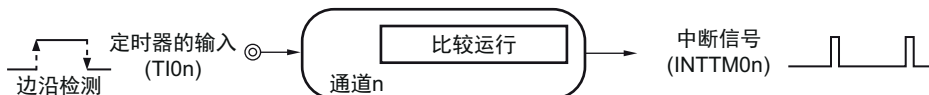
(2) 方波输出

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



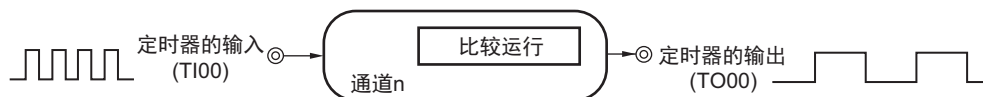
(3) 外部事件计数器

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。



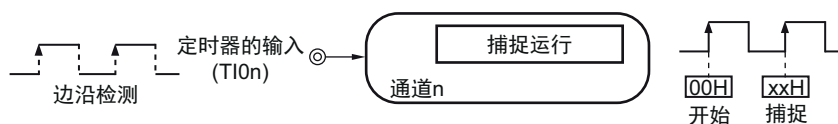
(4) 分频器

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



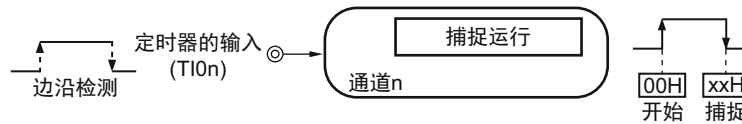
(5) 输入脉冲间隔的测量

在定时器输入引脚（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。



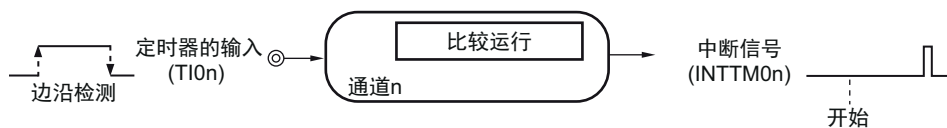
(6) 输入信号的高低电平宽度的测量

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低电平宽度。



(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 ~ 7)

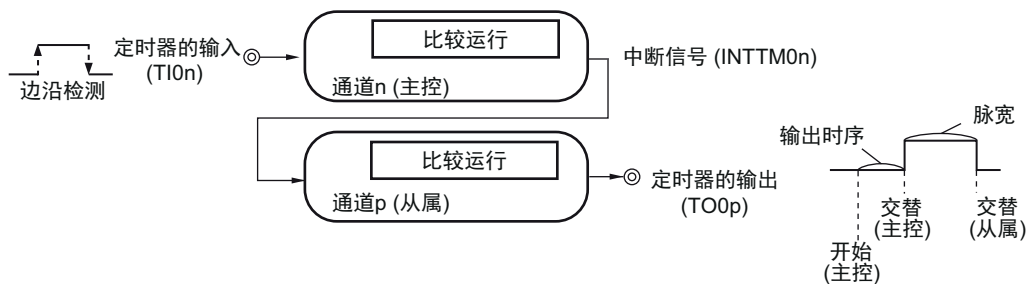
6.1.2 多通道联动运行功能

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

多通道联动运行功能可用作以下模式。

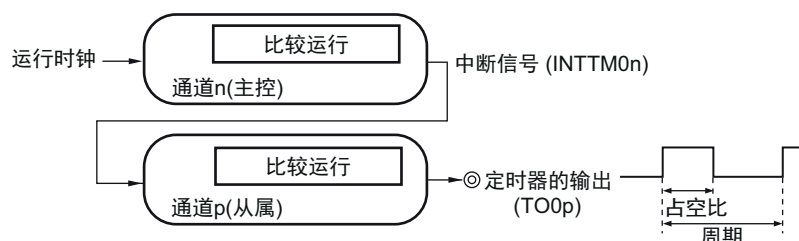
(1) 单触发脉冲输出

将 2 个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



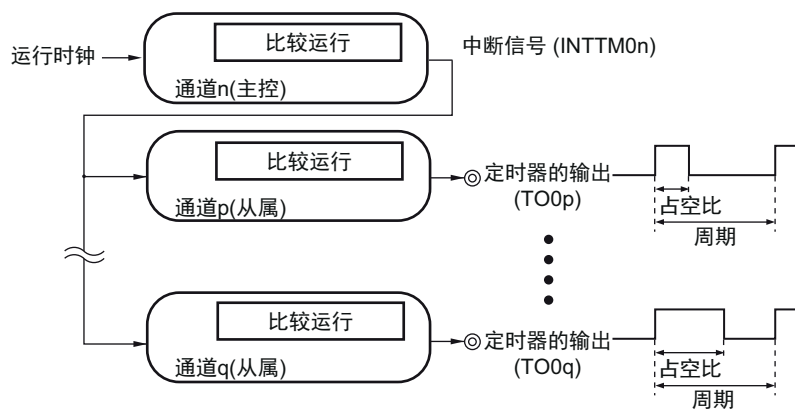
(2) PWM (Pulse Width Modulation) 输出

将 2 个通道成对使用，生成能任意设定周期和占空比的脉冲。



(3) 多重 PWM (Pulse Width Modulation) 输出

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



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

备注 n: 通道号 (n=0 ~ 7)

p、q: 从属通道号 (n < p < q ≤ 7)

6.1.3 8 位定时器运行功能 (只限于通道 1 和通道 3)

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

注意 在使用 8 位定时器运行功能时，有几个规则。

详细内容请参照“6.4.2 8 位定时器运行功能的基本规则 (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
计数器	定时器计数寄存器 0n (TCR0n)
寄存器	定时器数据寄存器 0n (TDR0n)
定时器的输入	TI00 ~ TI07
定时器的输出	TO00 ~ TO07 引脚、输出控制电路
控制寄存器	<单元设定部的寄存器> <ul style="list-style-type: none"> • 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) • 定时器通道开始寄存器 0 (TS0) • 定时器通道停止寄存器 0 (TT0) • 定时器输入选择寄存器 0 (TIS0) • 定时器输出允许寄存器 0 (TOE0) • 定时器输出寄存器 0 (TO0) • 定时器输出电平寄存器 0 (TOL0) • 定时器输出模式寄存器 0 (TOM0)
	<每个通道的寄存器> <ul style="list-style-type: none"> • 定时器模式寄存器 0n (TMR0n) • 定时器状态寄存器 0n (TSR0n) • 噪声滤波器允许寄存器 1 (NFEN1) • 端口模式控制寄存器 (PMCxx) 注 • 端口模式寄存器 (PMxx) 注 • 端口寄存器 (Pxx) 注

注 设定的端口模式控制寄存器 (PMCxx)、端口模式寄存器 (PMxx) 和端口寄存器 (Pxx) 因产品而不同。详细内容请参照“4.5.3 使用的端口功能和复用功能的寄存器设定例子”。

备注 n: 通道号 (n=0 ~ 7)

定时器阵列单元的各通道的定时器输入 / 输出复用端口因产品而不同。

表 6-2 本产品具有的定时器输入 / 输出引脚

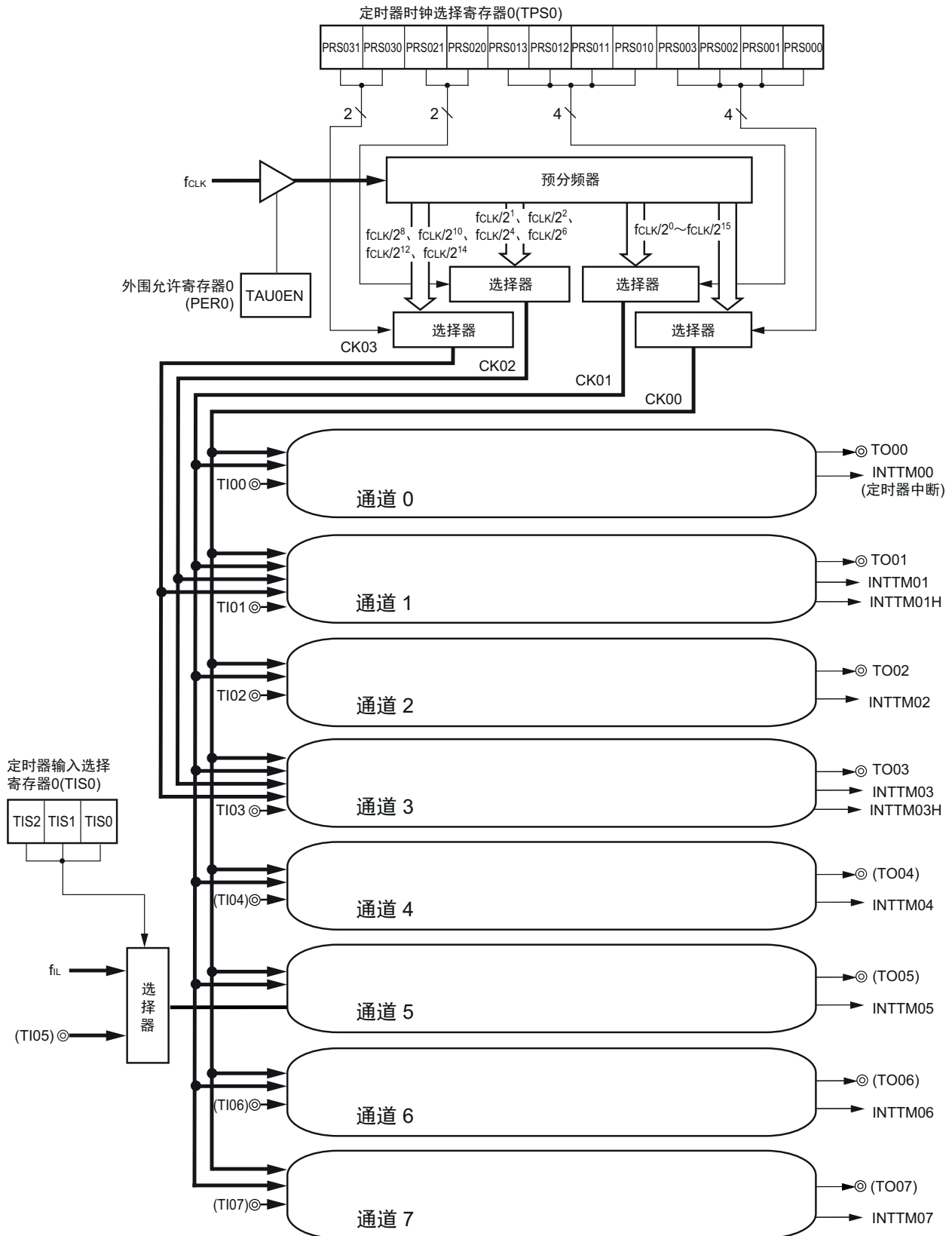
定时器阵列单元的通道	本产品
通道 0	TI00/TO00
通道 1	TI01/TO01
通道 2	TI02/TO02
通道 3	TI03/TO03
通道 4	(TI04/TO04)
通道 5	(TI05/TO05)
通道 6	(TI06/TO06)
通道 7	(TI07/TO07)

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

2. ×: 没有内置通道。

3. () 是将外围 I/O 重定向寄存器 (PIOR) 的 bit0 (PIOR0) 置“1”时的复用端口。

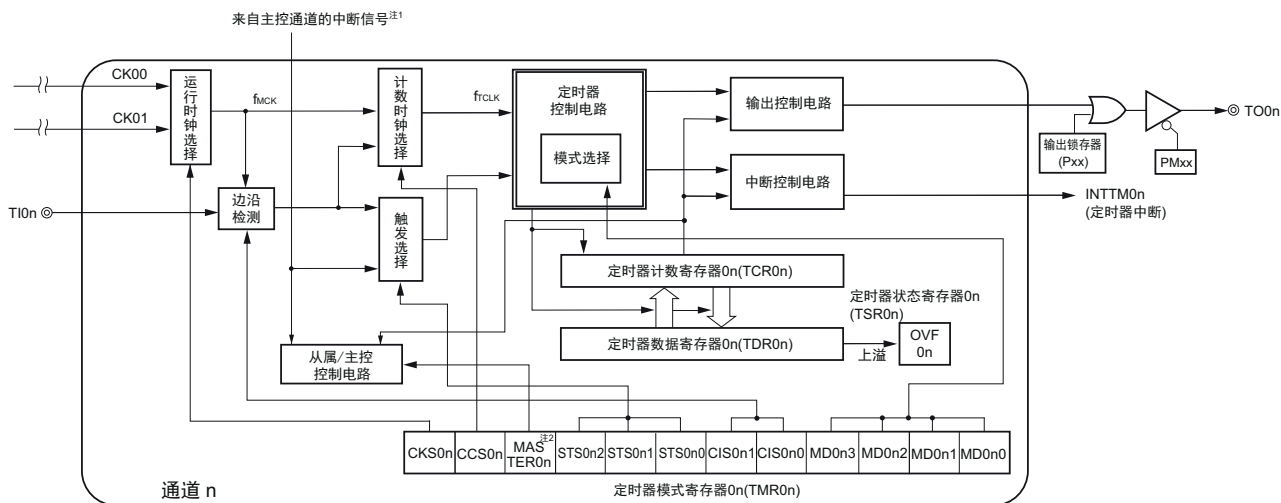
图 6-1 定时器阵列单元的整体框图



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。

图 6-2 定时器阵列单元 0 的通道内部框图

(a) 通道 0、2、4、6



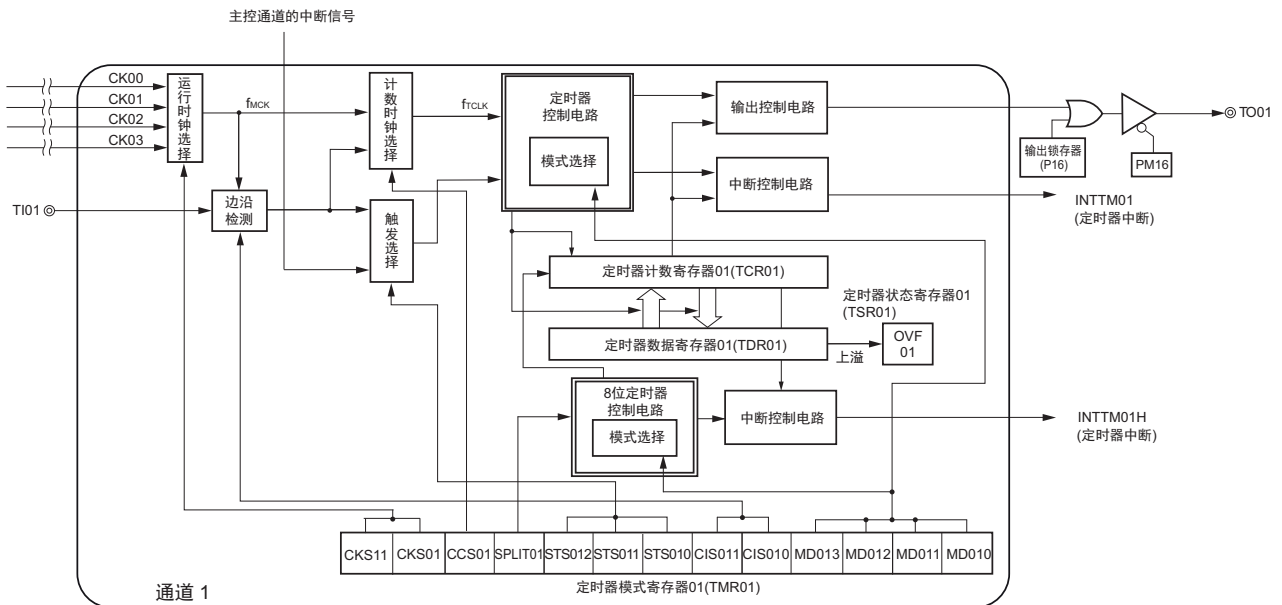
注 1. 只限于通道 2、4、6。

2. n=2、4、6

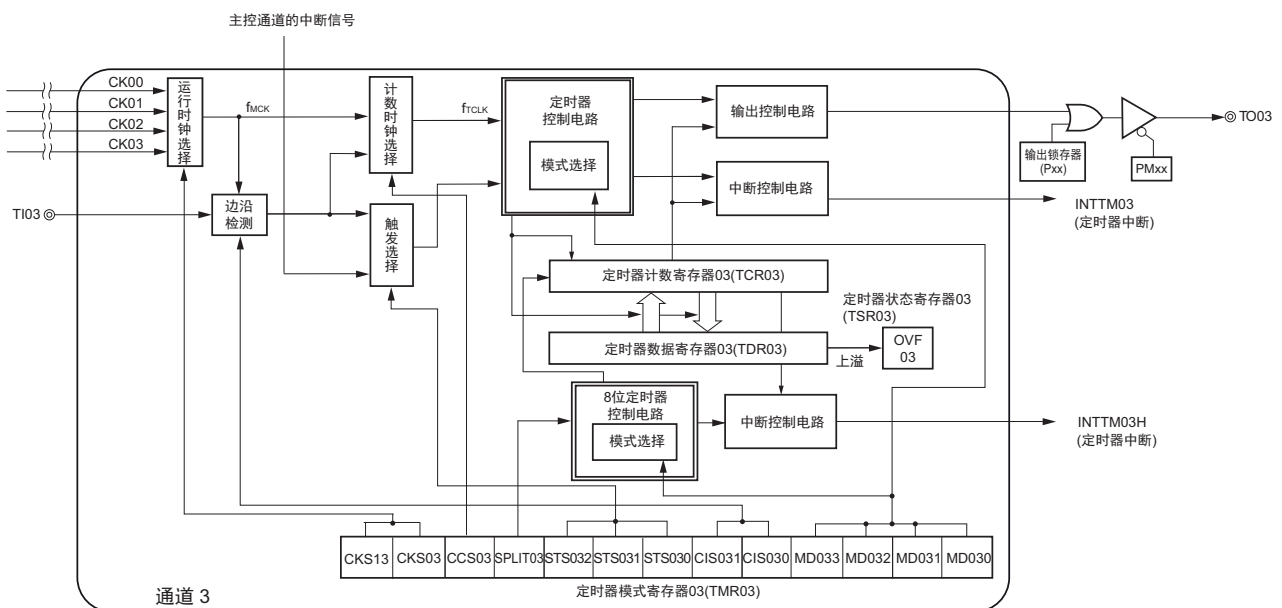
备注 1. n=0、2、4、6

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

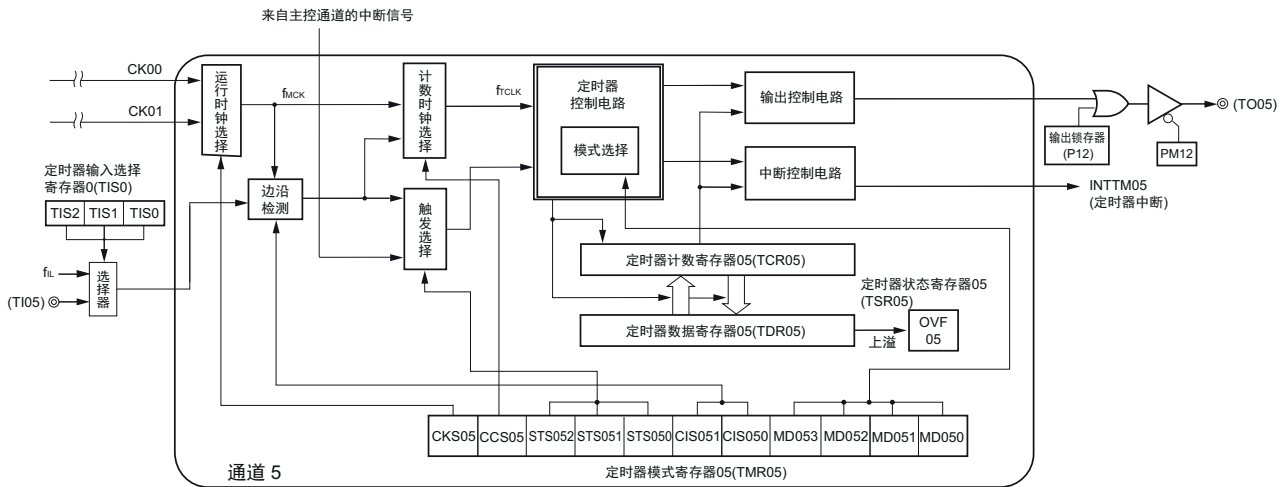
(b) 通道 1



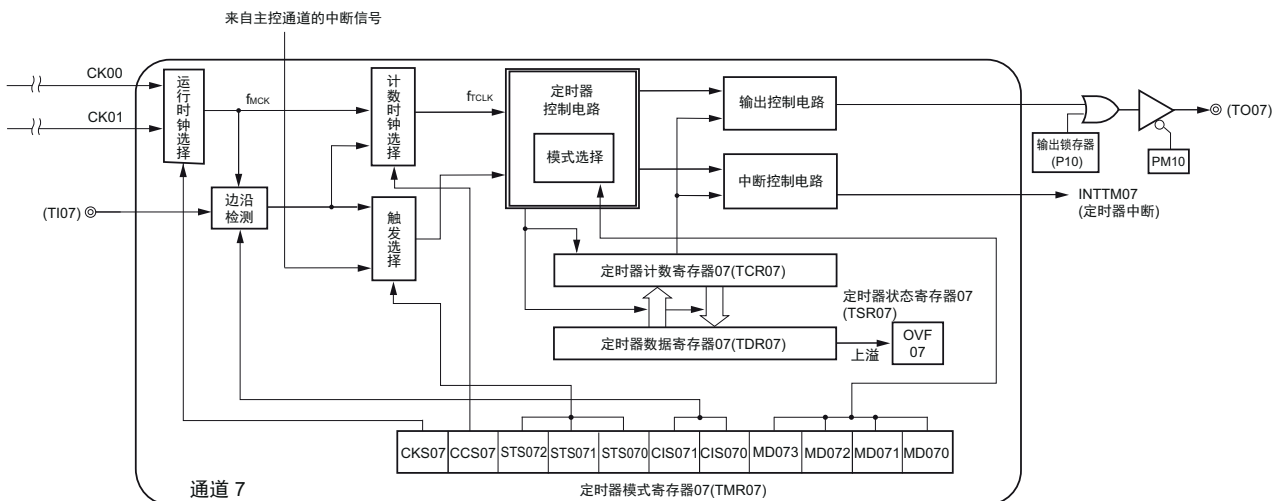
(c) 通道 3



(d) 通道 5



(e) 通道 7



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。

6.2.1 定时器计数寄存器 0n (TCR0n)

TCR0n 寄存器是对计数时钟进行计数的 16 位只读寄存器。

与计数时钟的上升沿同步进行递增或者递减计数。

通过定时器模式寄存器 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)”）。

6.3 控制定时器阵列单元的寄存器

控制定时器阵列单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 定时器时钟选择寄存器0 (TPS0)
- 定时器模式寄存器0n (TMR0n)
- 定时器状态寄存器0n (TSR0n)
- 定时器通道允许状态寄存器0 (TE0)
- 定时器通道开始寄存器0 (TS0)
- 定时器通道停止寄存器0 (TT0)
- 定时器输入选择寄存器0 (TIS0)
- 定时器输出允许寄存器0 (TOE0)
- 定时器输出寄存器0 (TO0)
- 定时器输出电平寄存器0 (TOL0)
- 定时器输出模式寄存器0 (TOM0)
- 噪声滤波器允许寄存器1 (NFEN1)
- 端口模式控制寄存器 (PMCxx)
- 端口模式寄存器 (PMxx)
- 端口寄存器 (Pxx)

备注 n: 通道号 (n=0 ~ 7)

注意 分配的寄存器和位因产品而不同。必须给未分配的位设定初始值。

6.3.1 外围允许寄存器 0 (PER0)

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

要使用定时器阵列单元时，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	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	定时器阵列单元的输入时钟的控制
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。

注意 1. 要设定定时器阵列单元时，必须先在 TAU0EN 位为“1”的状态下设定以下的寄存器。当 TAU0EN 位为“0”时，定时器阵列单元的控制寄存器的值为初始值，忽视写操作（定时器输入选择寄存器 0 (TISO)、噪声滤波器允许寄存器 1 (NFEN1)、端口模式寄存器 0、1、3、4 (PM0、PM1、PM3、PM4)、端口寄存器 0、1、3、4 (P0、P1、P3、P4) 和端口模式控制寄存器 0、1、4 (PMC0、PMC1、PMC4) 除外）。

- 定时器时钟选择寄存器 0 (TPS0)
- 定时器模式寄存器 0n (TMR0n)
- 定时器状态寄存器 0n (TSR0n)
- 定时器通道允许状态寄存器 0 (TE0)
- 定时器通道开始寄存器 0 (TS0)
- 定时器通道停止寄存器 0 (TT0)
- 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)

2. 必须将 bit6 和 bit1 置“0”。

6.3.2 定时器时钟选择寄存器 0 (TPS0)

TPS0 寄存器是 16 位寄存器，选择提供给各通道的 2 种或者 4 种公共运行时钟 (CK00 ~ CK03)。通过 TPS0 寄存器的 bit3 ~ 0 选择 CK00，通过 TPS0 寄存器的 bit7 ~ 4 选择 CK01。另外，只有通道 1 和通道 3 才能选择 CK02 和 CK03。能通过 TPS0 寄存器的 bit9 和 bit8 选择 CK02，通过 TPS0 寄存器的 bit13 和 bit12 选择 CK03。

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 ~ 7)：

选择 CK00 作为运行时钟 (CKS0n1、CKS0n0=0、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 ~ 7)：

选择 CK01 作为运行时钟 (CKS0n1、CKS0n0=0、1) 的通道全部处于停止状态 (TE0n=0)。

能改写 PRS020 位和 PRS021 位的情况 (n=1、3)：

选择 CK02 作为运行时钟 (CKS0n1、CKS0n0=1、0) 的通道全部处于停止状态 (TE0n=0)。

能改写 PRS030 位和 PRS031 位的情况 (n=1、3)：

选择 CK03 作为运行时钟 (CKS0n1、CKS0n0=1、1) 的通道全部处于停止状态 (TE0n=0)。

通过 16 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“0000H”。

图 6-7 定时器时钟选择寄存器 0 (TPS0) 的格式

地址: F01B6H、F01B7H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	运行时钟 (CK0k) 的选择注 (k=0、1)						
				f_{CLK}	$f_{CLK}=$ 2MHz	$f_{CLK}=$ 4MHz	$f_{CLK}=$ 8MHz	$f_{CLK}=$ 16MHz	$f_{CLK}=$ 20MHz	$f_{CLK}=$ 24MHz
0	0	0	0	f_{CLK}	2MHz	4MHz	8MHz	16MHz	20MHz	24MHz
0	0	0	1	$f_{CLK}/2$	1MHz	2MHz	4MHz	8MHz	10MHz	12MHz
0	0	1	0	$f_{CLK}/2^2$	500kHz	1MHz	2MHz	4MHz	5MHz	6MHz
0	0	1	1	$f_{CLK}/2^3$	250kHz	500kHz	1MHz	2MHz	2.5MHz	3MHz
0	1	0	0	$f_{CLK}/2^4$	125kHz	250kHz	500kHz	1MHz	1.25MHz	1.5MHz
0	1	0	1	$f_{CLK}/2^5$	62.5kHz	125kHz	250kHz	500kHz	625kHz	750kHz
0	1	1	0	$f_{CLK}/2^6$	31.3kHz	62.5kHz	125kHz	250kHz	313kHz	375kHz
0	1	1	1	$f_{CLK}/2^7$	15.6kHz	31.3kHz	62.5kHz	125kHz	156kHz	188kHz
1	0	0	0	$f_{CLK}/2^8$	7.81kHz	15.6kHz	31.3kHz	62.5kHz	78.1kHz	93.8kHz
1	0	0	1	$f_{CLK}/2^9$	3.91kHz	7.81kHz	15.6kHz	31.3kHz	39.1kHz	46.9kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95kHz	3.91kHz	7.81kHz	15.6kHz	19.5kHz	23.4kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	1.95kHz	3.91kHz	7.81kHz	9.77kHz	11.7kHz
1	1	0	0	$f_{CLK}/2^{12}$	488Hz	977Hz	1.95kHz	3.91kHz	4.88kHz	5.86kHz
1	1	0	1	$f_{CLK}/2^{13}$	244Hz	488Hz	977Hz	1.95kHz	2.44kHz	2.93kHz
1	1	1	0	$f_{CLK}/2^{14}$	122Hz	244Hz	488Hz	977Hz	1.22kHz	1.46kHz
1	1	1	1	$f_{CLK}/2^{15}$	61Hz	122Hz	244Hz	488Hz	610Hz	732Hz

PRS 021	PRS 020	运行时钟 (CK02) 的选择注						
		f_{CLK}	$f_{CLK}=$ 2MHz	$f_{CLK}=$ 4MHz	$f_{CLK}=$ 8MHz	$f_{CLK}=$ 16MHz	$f_{CLK}=$ 20MHz	$f_{CLK}=$ 24MHz
0	0	$f_{CLK}/2$	1MHz	2MHz	4MHz	8MHz	10MHz	12MHz
0	1	$f_{CLK}/2^2$	500kHz	1MHz	2MHz	4MHz	5MHz	6MHz
1	0	$f_{CLK}/2^4$	125kHz	250kHz	500kHz	1MHz	1.25MHz	1.5MHz
1	1	$f_{CLK}/2^6$	31.3kHz	62.5kHz	125kHz	250kHz	313kHz	375kHz

PRS 031	PRS 030	运行时钟 (CK03) 的选择注						
		f_{CLK}	$f_{CLK}=$ 2MHz	$f_{CLK}=$ 4MHz	$f_{CLK}=$ 8MHz	$f_{CLK}=$ 16MHz	$f_{CLK}=$ 20MHz	$f_{CLK}=$ 24MHz
0	0	$f_{CLK}/2^8$	7.81kHz	15.6kHz	31.3kHz	62.5kHz	78.1kHz	93.8kHz
0	1	$f_{CLK}/2^{10}$	1.95kHz	3.91kHz	7.81kHz	15.6kHz	19.5kHz	23.4kHz
1	0	$f_{CLK}/2^{12}$	488Hz	977Hz	1.95kHz	3.91kHz	4.88kHz	5.86kHz
1	1	$f_{CLK}/2^{14}$	122Hz	244Hz	488Hz	977Hz	1.22kHz	1.46kHz

注 在更改选择为 f_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 的情况下, 必须停止定时器阵列单元 (TT0=00FFH)。

注意 1. 必须将 bit15、14、11、10 置“0”。

2. 如果选择 f_{CLK} （无分频）作为运行时钟（CKmk）并且将 TDRnm 置“0000H”（ $n=0, 1, m=0 \sim 7$ ），就不能使用定时器阵列单元的中断请求。

备注 1. f_{CLK} : CPU/ 外围硬件的时钟频率

2. 上述的选择时钟从上升沿开始只有 1 个 f_{CLK} 周期为高电平（ $m=2 \sim 15$ ）。详细内容请参照“6.5.1 计数时钟（ f_{TCLK} ）”。

如果在 8 位定时器模式中使用通道 1 和通道 3 并且将 CK02 和 CK03 作为运行时钟，就能通过间隔定时器功能实现表 6-4 所示的间隔时间。

表 6-4 运行时钟 CKS02 和 CKS03 能设定的间隔时间

时钟		间隔时间 ($f_{CLK}=20\text{MHz}$) 注			
		10 μs	100 μs	1ms	10ms
CK02	$f_{CLK}/2$	○	—	—	—
	$f_{CLK}/2^2$	○	—	—	—
	$f_{CLK}/2^4$	○	○	—	—
	$f_{CLK}/2^6$	—	○	○	—
CK03	$f_{CLK}/2^8$	—	○	○	—
	$f_{CLK}/2^{10}$	—	—	○	—
	$f_{CLK}/2^{12}$	—	—	—	○
	$f_{CLK}/2^{14}$	—	—	—	—

注 ○包含 4% 以内的误差。

备注 有关 TPS0 寄存器所选 $f_{CLK}/2^n$ 波形的详细内容，请参照“6.5.1 计数时钟（ f_{TCLK} ）”。

6.3.3 定时器模式寄存器 0n（TMR0n）

TMR0n 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟（ f_{MCK} ）的选择、计数时钟的选择、主控 / 从属的选择、16 位 / 8 位定时器的选择（只限于通道 1 和通道 3）、开始触发和捕捉触发的设定、定时器输入有效边沿的选择以及运行模式（间隔、捕捉、事件计数器、单次计数、捕捉 & 单次计数）的设定。

禁止在运行中（TE0n=1）改写 TMR0n 寄存器。但是，能在一部分的功能运行中（TE0n=1）改写 bit7 和 bit6（CIS0n1、CIS0n0）（详细内容请参照“6.8 定时器阵列单元的独立通道运行功能”和“6.9 定时器阵列单元的多通道联动运行功能”）。

通过 16 位存储器操作指令设定 TMR0n 寄存器。

在产生复位信号后，TMR0n 寄存器的值变为“0000H”。

注意 TMR0n 寄存器的 bit11 因通道而不同。

TMR02、TMR04、TMR06: MASTER0n 位（ $n=2, 4, 6$ ）

TMR01、TMR03: SPLIT0n 位（ $n=1, 3$ ）

TMR00、TMR05、TMR07: 固定为“0”。

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (1/4)

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

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

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

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

CKS0n1	CKS0n0	通道 n 运行时钟 (f_{MCK}) 的选择
0	0	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK00
0	1	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK02
1	0	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK01
1	1	定时器时钟选择寄存器 0 (TPS0) 设定的运行时钟 CK03
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})。 只有通道 1 和通道 3 才能选择运行时钟 CK02 和 CK03。		

CCS0n	通道 n 计数时钟 (f_{TCLK}) 的选择
0	CKS0n0 位和 CKS0n1 位指定的运行时钟 (f_{MCK})
1	TIO _n 引脚输入信号的有效边沿 在通道 5 的情况下, 为 TIS0 选择的输入信号的有效边沿。
计数时钟 (f_{TCLK}) 用于计数器、输出控制电路和中断控制电路。	

注 bit11 是只读位, 固定为“0”, 忽视写操作。

注意 1. 必须将 bit13、5、4 置“0”。

- 要更改选择为 f_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 时, 即使选择了 CKS0n0 位和 CKS0n1 位指定的运行时钟 (f_{MCK}) 或者 TIO_n 引脚输入信号的有效边沿作为计数时钟 (f_{TCLK}), 也必须停止定时器阵列单元 (TT0=00FFH)。

备注 n: 通道号 (n=0 ~ 7)

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (2/4)

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

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

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

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

(TMR0n (n=2、4、6) 的 bit11)

MASTER0n	通道 n 的独立通道运行 / 多通道联动运行 (从属或者主控) 的选择
0	用作独立通道运行功能或者多通道联动运行功能的从属通道。
1	用作多通道联动运行功能的主控通道。
只能将通道 2、4、6 设为主控通道 (MASTER0n=1)。 通道 0、5、7 固定为“0” (因为通道 0 为最高位通道, 所以与此位的设定无关, 用作主控通道)。 对于用作独立通道运行功能的通道, 将 MASTER0n 位置“0”。	

(TMR0n (n=1、3) 的 bit11)

SPLIT0n	通道 1 和通道 3 的 8 位定时器 /16 位定时器的运行选择
0	用作 16 位定时器。 (用作独立通道运行功能或者多通道联动运行功能的从属通道)
1	用作 8 位定时器。

STS0n2	STS0n1	STS0n0	通道 n 的开始触发和捕捉触发的设定
0	0	0	只有软件触发开始有效 (不选择其他触发源)。
0	0	1	将 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	使用主控通道的中断信号 (多通道联动运行功能的从属通道的情况)。
上述以外			禁止设定。

注 bit11 是只读位, 固定为“0”, 忽视写操作。

备注 n: 通道号 (n=0 ~ 7)

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (3/4)

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

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

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

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

CIS0n1	CIS0n0	TIOn 引脚的有效边沿选择
0	0	下降沿
0	1	上升沿
1	0	双边沿 (测量低电平宽度时) 开始触发: 下降沿, 捕捉触发: 上升沿
1	1	双边沿 (测量高电平宽度时) 开始触发: 上升沿, 捕捉触发: 下降沿

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

注 bit11 是只读位, 固定为 "0", 忽视写操作。

备注 n: 通道号 (n=0 ~ 7)

图 6-8 定时器模式寄存器 0n (TMR0n) 的格式 (4/4)

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

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n=2, 4, 6)	CKS On1	CKS On0	0	CCS On	MAST ER0n	STS On2	STS On1	STS On0	CIS On1	CIS On0	0	0	MD On3	MD On2	MD On1	MD On0

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n=1, 3)	CKS On1	CKS On0	0	CCS On	SPLIT On	STS On2	STS On1	STS On0	CIS On1	CIS On0	0	0	MD On3	MD On2	MD On1	MD On0

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n=0, 5, 7)	CKS On1	CKS On0	0	CCS On	0注 1	STS On2	STS On1	STS On0	CIS On1	CIS On0	0	0	MD On3	MD On2	MD On1	MD On0

MD On3	MD On2	MD On1	通道 n 运行模式的设定	对应功能	TCR 的计数运行
0	0	0	间隔定时器模式	间隔定时器 / 方波输出 / 分频器功能 / PWM 输出 (主控)	递减计数
0	1	0	捕捉模式	输入脉冲间隔的测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单次计数模式	延迟计数器 / 单触发脉冲输出 / PWM 输出 (从属)	递减计数
1	1	0	捕捉 & 单次计数模式	输入信号的高低电平宽度的测量	递增计数
上述以外			禁止设定。		
各模式的运行因 MDOn0 位而变 (参照下表)。					

运行模式 (MDOn3 ~ MDOn1 位的设定 (参照上表))	MD On0	开始计数和中断的设定
• 间隔定时器模式 (0、0、0) • 捕捉模式 (0、1、0)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
	1	在开始计数时产生定时器中断 (定时器的输出也发生变化)。
• 事件计数器模式 (0、1、1)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。
• 单次计数模式注 2 (1、0、0)	0	计数运行中的开始触发无效。此时不产生中断。
	1	计数运行中的开始触发有效注 3。此时不产生中断。
• 捕捉 & 单次计数模式 (1、1、0)	0	在开始计数时不产生定时器中断 (定时器的输出也不发生变化)。计数运行中的开始触发无效。此时不产生中断。
上述以外		禁止设定。

注 1. bit11 是只读位, 固定为“0”, 忽视写操作。

2. 在单次计数模式中, 不控制开始计数时的中断输出 (INTTM0n) 和 TO0n 输出。

3. 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)。

备注 n: 通道号 (n=0 ~ 7)

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, 在其他模式中不被置位。有关各运行模式中的 OVF 位的变化和置位 / 清除条件, 请参照表 6-5。

通过 16 位存储器操作指令读 TSR0n 寄存器。

能用 TSR0nL 并且通过 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“0000H”。

图 6-9 定时器状态寄存器 0n (TSR0n) 的格式

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

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

OVF	通道 n 的计数器上溢状态
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 ~ 7)

表 6-5 各运行模式中的 OVF 位的变化和置位 / 清除条件

定时器运行模式	OVF 位	置位 / 清除条件
<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 即使计数器发生上溢, OVF 位也不立即发生变化, 而在此后的捕捉时发生变化。

6.3.5 定时器通道允许状态寄存器 0 (TE0)

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

通过 16 位存储器操作指令读 TE0 寄存器。

能用 TE0L 并且通过 1 位或者 8 位存储器操作指令读 TE0 寄存器的低 8 位。

在产生复位信号后，TE0 寄存器的值变为“0000H”。

图 6-10 定时器通道允许状态寄存器 0 (TE0) 的格式

地址: F01B0H、F01B1H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TEH03	0	TEH01	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TEH03	通道 3 为 8 位定时器模式时的高 8 位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

TEH01	通道 1 为 8 位定时器模式时的高 8 位定时器的运行允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

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

在通道 1 和通道 3 为 8 位定时器模式时，TE01 和 TE03 表示低 8 位定时器的运行允许或者停止状态。

备注 n: 通道号 (n=0 ~ 7)

6.3.6 定时器通道开始寄存器 0 (TS0)

TS0 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。

如果将各位置“1”，定时器通道允许状态寄存器 0 (TE0) 的对应位就被置“1”。因为 TS0n 位、TSH01 位和 TSH03 位是触发位，所以如果变为运行允许状态 (TE0n、TEH01、TEH03=1)，就立即清除 TS0n 位、TSH01 位和 TSH03 位。

通过 16 位存储器操作指令设定 TS0 寄存器。

能用 TS0L 并且通过 1 位或者 8 位存储器操作指令设定 TS0 寄存器的低 8 位。

在产生复位信号后，TS0 寄存器的值变为“0000H”。

图 6-11 定时器通道开始寄存器 0 (TS0) 的格式

地址：F01B2H、F01B3H 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TSH03	0	TSH01	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TSH03	通道 3 为 8 位定时器模式时的高 8 位定时器的运行允许 (开始) 触发
0	没有触发。
1	将 TEH03 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCR03 寄存器的计数，就进入间隔定时器模式 (参照表 6-6)。

TSH01	通道 1 为 8 位定时器模式时的高 8 位定时器的运行允许 (开始) 触发
0	没有触发。
1	将 TEH01 位置“1”，进入计数允许状态。 如果在计数允许状态下开始 TCR01 寄存器的计数，就进入间隔定时器模式 (参照表 6-6)。

TS0n	通道 n 的运行允许 (开始) 触发
0	没有触发。
1	将 TE0n 位置“1”，进入计数允许状态。 计数允许状态下的 TCR0n 寄存器的计数开始因各运行模式而不同 (参照表 6-6)。 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。

注意 1. 必须将未定义的位置“0”。

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 (TSH01、TSH03) 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 (TNFEN0n=1)：4 个运行时钟 (f_{MCK})

TI0n 引脚噪声滤波器无效时 (TNFEN0n=0)：2 个运行时钟 (f_{MCK})

备注 1. TS0 寄存器的读取值总是“0”。

2. n: 通道号 (n=0 ~ 7)

6.3.7 定时器通道停止寄存器 0 (TT0)

TT0 寄存器是设定各通道计数停止的触发寄存器。

如果将各位置“1”，定时器通道允许状态寄存器 0 (TE0) 的对应位就被清“0”。因为 TT0n 位、TTH01 位和 TTH03 位是触发位，所以如果变为运行停止状态 (TE0n、TEH01、TEH03=0)，就立即清除 TT0n 位、TTH01 位和 TTH03 位。

通过 16 位存储器操作指令设定 TT0 寄存器。

能用 TT0L 并且通过 1 位或者 8 位存储器操作指令设定 TT0 寄存器的低 8 位。

在产生复位信号后，TT0 寄存器的值变为“0000H”。

图 6-12 定时器通道停止寄存器 0 (TT0) 的格式

地址：F01B4H、F01B5H 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TTH03	0	TTH01	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

TTH03	通道 3 为 8 位定时器模式时的高 8 位定时器的运行停止触发
0	没有触发。
1	将 TEH03 位清“0”，进入计数停止状态。

TTH01	通道 1 为 8 位定时器模式时的高 8 位定时器的运行停止触发
0	没有触发。
1	将 TEH01 位清“0”，进入计数停止状态。

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

注意 必须将未定义的位置“0”。

备注 1. TT0 寄存器的读取值总是“0”。

2. n: 通道号 (n=0 ~ 7)

6.3.8 定时器输入选择寄存器 0 (TIS0)

通过 8 位存储器操作指令设定 TIS0 寄存器。

在产生复位信号后，TIS0 寄存器的值变为“00H”。

图 6-13 定时器输入选择寄存器 0 (TIS0) 的格式

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

TIS02	TIS01	TIS00	通道 5 使用的定时器输入的选择
0	×	×	定时器输入引脚 (TI05) 的输入信号
1	0	0	低速内部振荡器时钟 (f_{IL})
上述以外			禁止设定。

×: 忽略

6.3.9 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 16 位存储器操作指令设定 TOE0 寄存器。

能用 TOE0L 并且通过 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“0000H”。

图 6-14 定时器输出允许寄存器 0 (TOE0) 的格式

地址: F01BAH、F01BBH	复位后: 0000H																R/W
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00	

TOE0n	通道 n 的定时器输出的允许 / 禁止
0	禁止定时器输出。 定时器的运行不反映到 TO0n 位，固定输出。 能写 TO0n 位，并且从 TO0n 引脚输出 TO0n 位设定的电平。
1	允许定时器输出。 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。

注意 必须将未定义的位置“0”。

备注 n: 通道号 (n=0 ~ 7)

6.3.10 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 TO00、TO01、TO02、TO03、(TO04)、(TO05)、(TO06)、(TO07) 引脚用作端口功能时，必须将相应的 TO0n 位置“0”。

通过 16 位存储器操作指令设定 TO0 寄存器。

能用 TO0L 并且通过 8 位存储器操作指令设定 TO0 寄存器的低 8 位。

在产生复位信号后，TO0 寄存器的值变为“0000H”。

图 6-15 定时器输出寄存器 0 (TO0) 的格式

地址: F01B8H、F01B9H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO07	TO06	TO05	TO04	TO03	TO02	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”。

注意 必须将未定义的位置“0”。

备注 n: 通道号 (n=0 ~ 7)

6.3.11 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) 并且使用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。在主控通道输出模式 (TOM0n=0) 中, 此寄存器的设定无效。

通过 16 位存储器操作指令设定 TOL0 寄存器。

能用 TOL0L 并且通过 8 位存储器操作指令设定 TOL0 寄存器的低 8 位。

在产生复位信号后, TOL0 寄存器的值变为“0000H”。

图 6-16 定时器输出电平寄存器 0 (TOL0) 的格式

地址: F01BCH、F01BDH 复位后: 0000H R/W

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

注意 必须将未定义的位置“0”。

备注 1. 如果在定时器运行中改写此寄存器的值, 就在下一次定时器输出信号发生变化时定时器反相输出, 而不是在改写后立即反相。

2. n: 通道号 (n=1 ~ 7)

6.3.12 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（PWM 输出、单触发脉冲输出和多重 PWM 输出）时，将主控通道的对应位置“0”并且将从属通道的对应位置“1”。

当允许定时器输出（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 16 位存储器操作指令设定 TOM0 寄存器。

能用 TOM0L 并且通过 8 位存储器操作指令设定 TOM0 寄存器的低 8 位。

在产生复位信号后，TOM0 寄存器的值变为“0000H”。

图 6-17 定时器输出模式寄存器 0 (TOM0) 的格式

地址：F01BEH、F01BFH 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	0

TOM0n	通道 n 的定时器输出模式的控制
0	主控通道输出模式（通过定时器中断请求信号（INTTM0n）进行交替输出）
1	从属通道输出模式（通过主控通道的定时器中断请求信号（INTTM0n）将输出置位，并且通过从属通道的定时器中断请求信号（INTTM0p）对输出进行复位）

注意 必须将未定义的位置“0”。

备注 n: 通道号

n=1~7（主控通道：n=0、2、4、6）

p: 从属通道号

n < p ≤ 7

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

6.3.13 噪声滤波器允许寄存器 1 (NFEN1)

NFEN1 寄存器设定噪声滤波器是否用于各通道定时器输入引脚的输入信号。

对于需要消除噪声的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟 (f_{MCK}) 进行同步后检测 2 个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟 (f_{MCK}) 进行同步注。

通过 1 位或者 8 位存储器操作指令设定 NFEN1 寄存器。

在产生复位信号后，NFEN1 寄存器的值变为“00H”。

注 详细内容请参照“6.5.1 (2) 选择 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”。

图 6-18 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	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00
	TNFEN0n	TI0n 引脚的输入信号噪声滤波器的使用与否						
	0	噪声滤波器 OFF						
	1	噪声滤波器 ON						

备注 通道 0 ~ 7 的定时器输入 / 输出引脚的有无因产品而不同。详细内容请参照“表 6-2 本产品具有的定时器输入 / 输出引脚”。

6.3.14 控制定时器输入 / 输出引脚端口功能的寄存器

在使用定时器阵列单元时，必须设定与对象通道复用的端口功能的控制寄存器（端口模式寄存器（PMxx）、端口寄存器（Pxx）和端口模式控制寄存器（PMCxx））。详细内容请参照“4.3.1 端口模式寄存器（PMxx）”、“4.3.2 端口寄存器（Pxx）”和“4.3.6 端口模式控制寄存器（PMCxx）”。

设定的端口模式寄存器（PMxx）、端口寄存器（Pxx）和端口模式控制寄存器（PMCxx）因产品而不同。详细内容请参照“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（P01/TO00 等）用作定时器的输出时，必须将各端口对应的端口模式控制寄存器（PMCxx）的位、端口模式寄存器（PMxx）的位和端口寄存器（Pxx）的位置“0”。

例) 将 P01/TO00 用作定时器输出的情况

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“0”。

将端口寄存器 0 的 P01 位置“0”。

在将定时器输入引脚的复用端口（P00/TI00 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”，并且将各端口对应的端口控制模式寄存器（PMCxx）的位置“0”。此时，端口寄存器（Pxx）的位可以是“0”或者“1”。

例) 将 P00/TI00 用作定时器输入的情况

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“1”。

将端口寄存器 0 的 P00 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、通道4、通道6）设定为主控通道。

(2) 能将通道0以外的全部通道设定为从属通道。

(3) 只能将主控通道的低位通道设定为从属通道。

例 在将通道2设定为主控通道时，能将通道3开始的通道（通道3～7通道）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4设定为主控通道时，能将通道1～通道3设定为主控通道0的从属通道，而不能将通道5～通道7设定为主控通道0的从属通道。

(6) 和主控通道联动的从属通道需要设定相同的运行时钟。和主控通道联动的从属通道的CKS0n0位和CKS0n1位（定时器模式寄存器0n（TMR0n）的bit15和bit14）的值需要是相同的设定值。

(7) 主控通道能将INTTM0n（中断）、开始软件触发和计数时钟传给低位通道。

(8) 从属通道能将主控通道的INTTM0n（中断）、开始软件触发和计数时钟用作源时钟，但是不能将自己的INTTM0n（中断）、开始软件触发和计数时钟传给低位通道。

(9) 主控通道不能将其他高位主控通道的INTTM0n（中断）、开始软件触发和计数时钟用作源时钟。

(10) 为了同时启动要联动的通道，需要同时设定联动通道的通道开始触发位（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TS0n位的设定。不能只使用从属通道的TS0n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）。

(13) 在联动运行时，因为主控通道和从属通道需要相同的运行时钟，所以不能选择CK02/CK03。

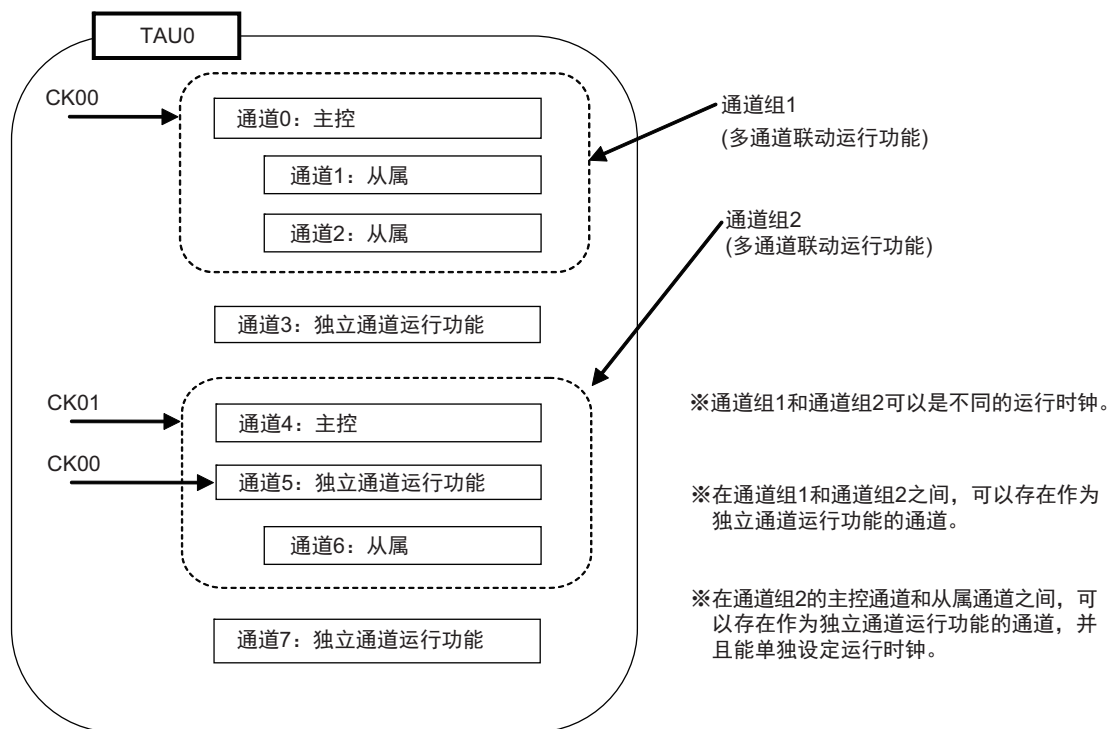
(14) 定时器模式寄存器0n（TMR0n）没有主控位而固定为“0”。但是，因为通道0是最高位的通道，所以在联动运行时能将通道0用作主控通道。

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

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

备注 n: 通道号 (n=0～7)

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 0n（TMR0n）的 SPLIT0n 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 INTTM01H/INTTM03H（中断）（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 TMR0n 寄存器的 CKS0n1 位和 CKS0n0 位的设定。
- (6) 对于高 8 位定时器，通过操作 TSH01/TSH03 位来开始通道的运行，并且通过操作 TTH01/TTH03 位来停止通道的运行。能通过 TEH01/TEH03 位确认通道的状态。
- (7) 低 8 位定时器的运行取决于 TMR0n 寄存器的设定，有以下 3 种支持低 8 位定时器运行的功能：
 - 间隔定时器功能/方波输出功能
 - 外部事件计数器功能
 - 延迟计数功能
- (8) 对于低 8 位定时器，通过操作 TS01/TS03 位来开始通道的运行，并且通过操作 TT01/TT03 位来停止通道的运行。能通过 TE01/TE03 位确认通道的状态。
- (9) 在 16 位定时器运行时，TSH01/TSH03/TTH01/TTH03 位的操作无效。通过操作 TS01/TS03 位和 TT01/TT03 位使通道 1 和通道 3 运行。TEH03 位和 TEH01 位不变。
- (10) 8 位定时器功能不能使用联动运行功能（单触发脉冲、PWM 和多重 PWM）。

备注 n: 通道号 (n=1、3)

6.5 计数器的运行

6.5.1 计数时钟 (f_{TCLK})

定时器阵列单元的计数时钟 (f_{TCLK}) 能通过定时器模式寄存器 0n (TMR0n) 的 CCS0n 位选择以下任意一个时钟:

- CKS0n0 位和 CKS0n1 位指定的运行时钟 (f_{MCK})
- TI0n 引脚输入信号的有效边沿

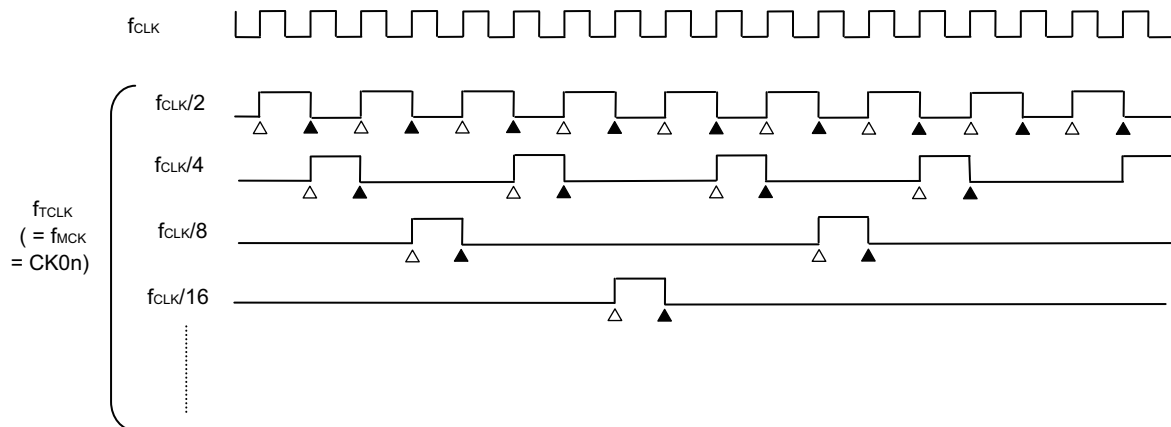
定时器阵列单元被设计为与 f_{CLK} 同步运行, 因此计数时钟 (f_{TCLK}) 的时序如下。

(1) 选择 CKS0n0 位和 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) 从计数时钟的上升沿开始延迟 1 个 f_{CLK} 时钟后进行计数, 出于方便而将其称为“在计数时钟的上升沿进行计数”。

图 6-19 f_{CLK} 和计数时钟 (f_{TCLK}) 的时序 (CCS0n=0 的情况)

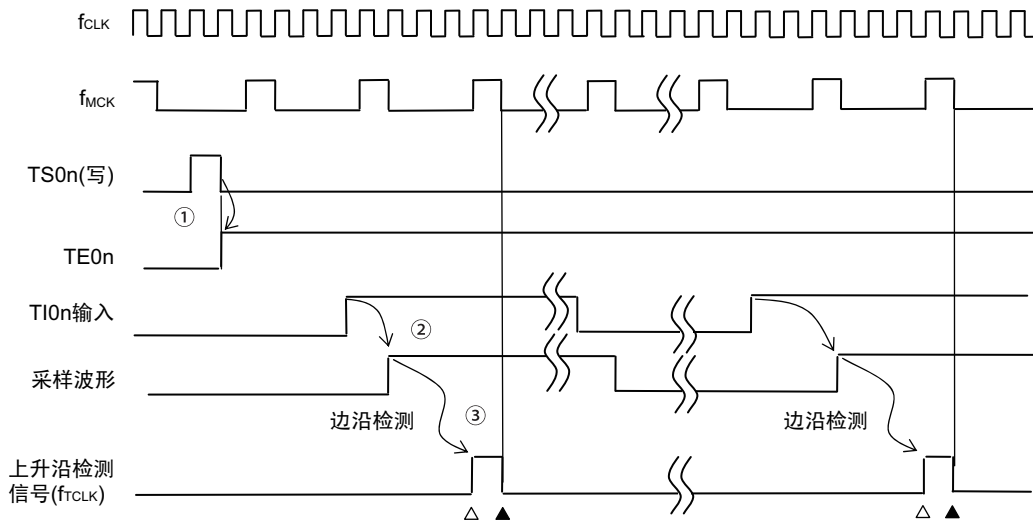


- 备注 1. Δ : 计数时钟的上升沿
 \blacktriangle : 同步、计数器的递增 / 递减
2. f_{CLK} : CPU/ 外围硬件时钟

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

图 6-20 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的运行，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)。

备注 1. Δ : 计数时钟的上升沿

\blacktriangle : 同步、计数器的递增 / 递减

2. f_{CLK} : CPU/ 外围硬件时钟
- 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 6-6 所示。

表 6-6 从计数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, 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 (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 (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 (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 (5) 捕捉 & 单次计数模式的运行 (高电平宽度的测量)”)。

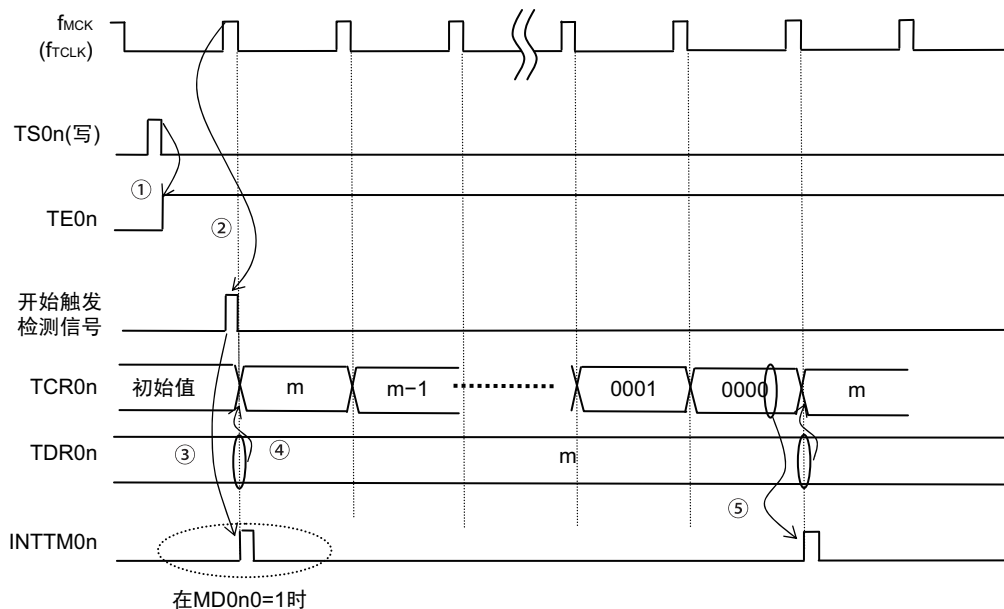
6.5.3 计数器的运行

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，直到产生计数时钟为止。
- ② 通过允许运行后的第 1 个计数时钟（ f_{MCK} ）产生开始触发信号。
- ③ 当 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（ f_{MCK} ）产生 INTTM0n，并且在将定时器数据寄存器 0n（TDR0n）的值装入 TCR0n 寄存器后继续计数。

图 6-21 运行时序（间隔定时器模式）



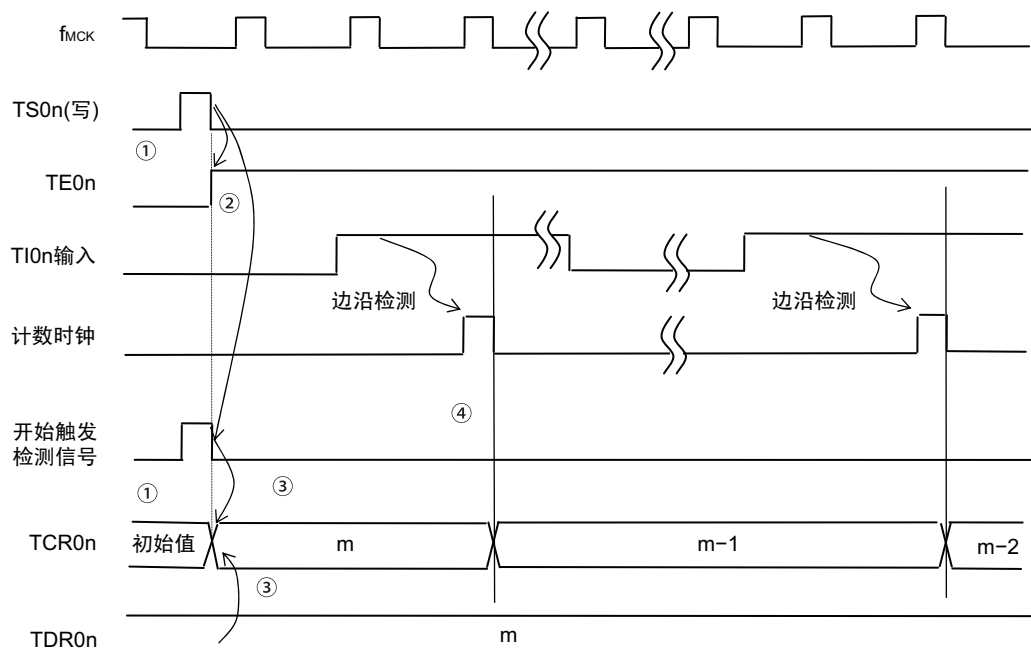
注意 因为第 1 个计数时钟周期的运行在写 TS0n 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MD0n0 位置“1”，以便能在开始计数时产生中断。

备注 f_{MCK} 、开始触发检测信号和 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, 通过计数时钟对 $TCR0n$ 寄存器的值进行递减计数。

图 6-22 运行时序 (事件计数器模式)

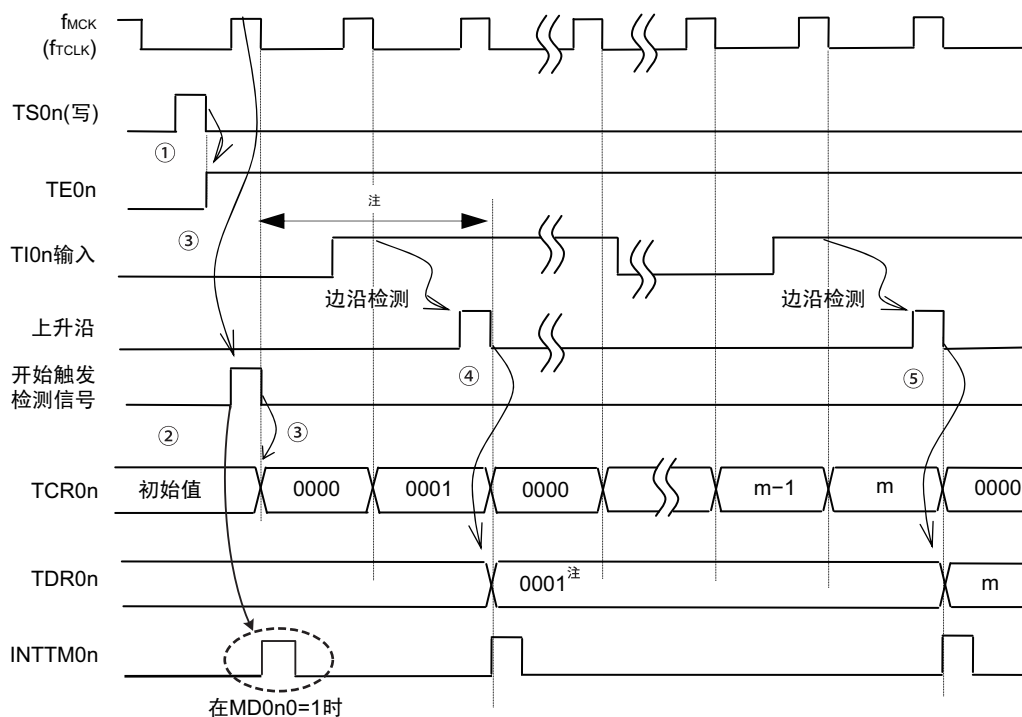


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器, 边沿检测就从 $TI0n$ 输入开始再延迟 2 个 f_{MCK} 周期 (合计 3 ~ 4 个周期)。1 个周期的误差是因为 $TI0n$ 输入和计数时钟 (f_{MCK}) 不同步。

(3) 捕捉模式的运行（输入脉冲的间隔测量）

- ① 通过给 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，直到产生计数时钟为止。
- ③ 通过允许运行后的第 1 个计数时钟（ f_{MCK} ）产生开始触发信号。然后，将“0000H”装入 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 INTTM0n 中断。此时的捕捉值没有意义。TCR0n 寄存器从“0000H”开始继续计数。
- 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 INTTM0n 中断。

图 6-23 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，即使没有检测到边沿也通过触发检测来开始计数，因此第 1 次捕捉时（④）的捕捉值不是脉冲间隔（在此例子中，0001：2 个时钟间隔），必须忽视。

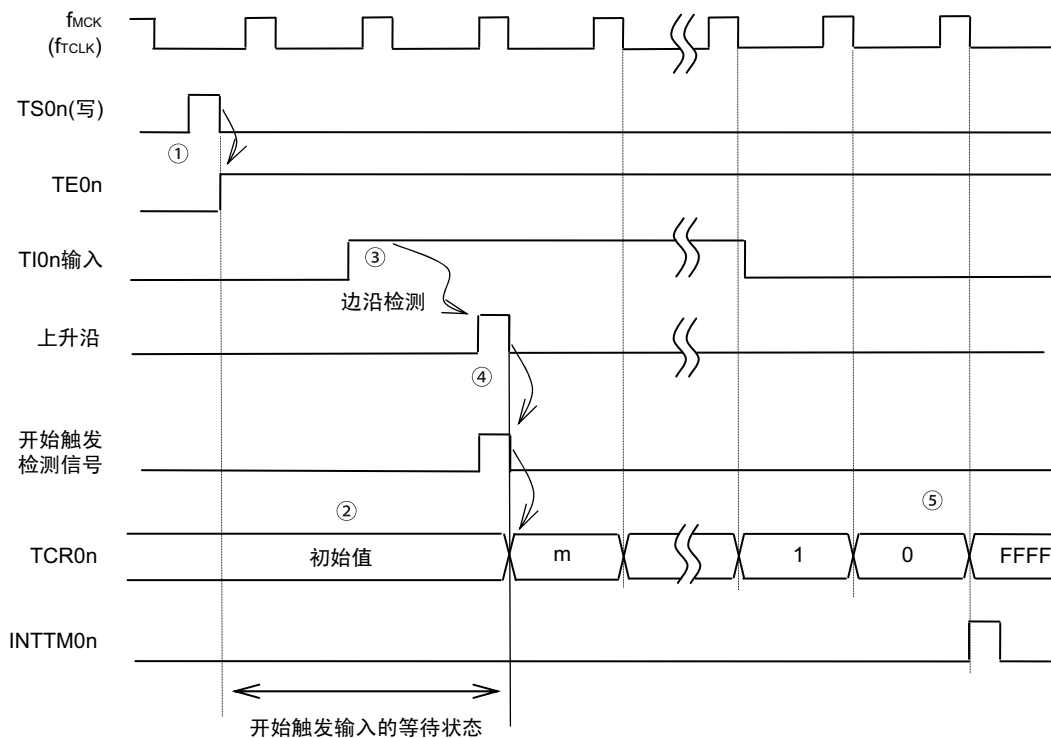
注意 因为第 1 个计数时钟周期的运行在写 TS0n 位后并且在产生计数时钟前延迟计数的开始，所以产生最大为 1 个时钟周期的误差。另外，如果需要开始计数时序的信息，就将 MD0n0 位置“1”，以便能在开始计数时产生中断。

备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI0n 输入开始再延迟 2 个 f_{MCK} 周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为 TI0n 输入和计数时钟（ f_{MCK} ）不同步。

(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 INTTM0n 中断，并且 TCR0n 寄存器的值变为“FFFF”，停止计数。

图 6-24 运行时序 (单次计数模式)

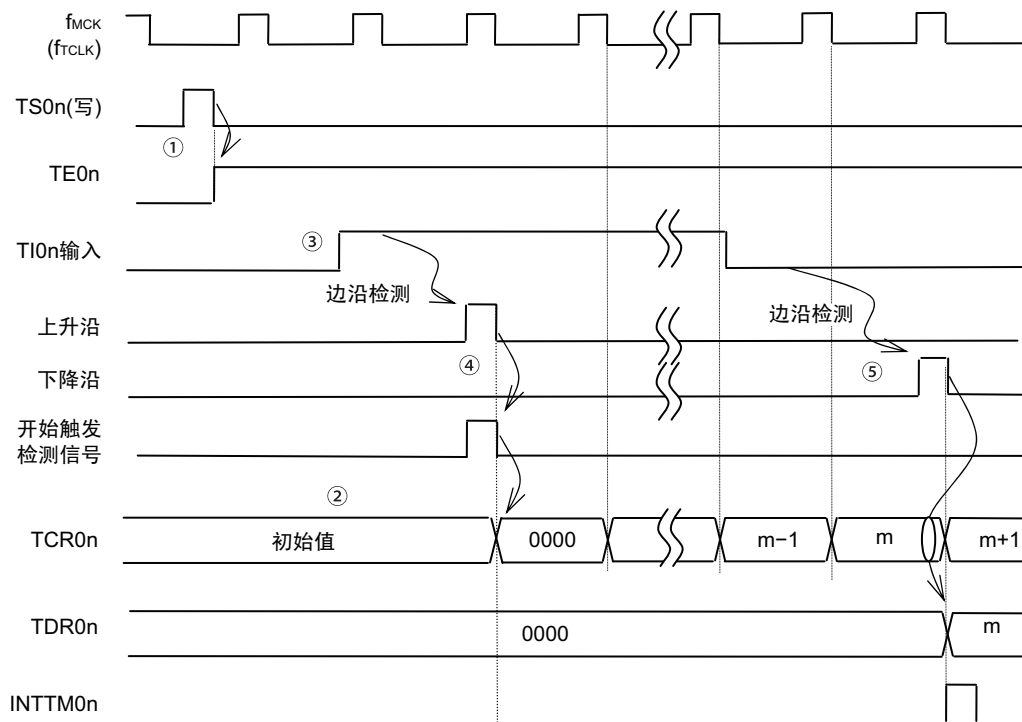


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI0n 输入开始再延迟 2 个 f_{MCK} 周期 (合计 3~4 个周期)。1 个周期的误差是因为 TI0n 输入与计数时钟 (f_{MCK}) 不同步。

(5) 捕捉 & 单次计数模式的运行（高电平宽度的测量）

- ① 通过给定时器通道开始寄存器 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 INTTM0n 中断。

图 6-25 运行时序（捕捉 & 单次计数模式：高电平宽度的测量）

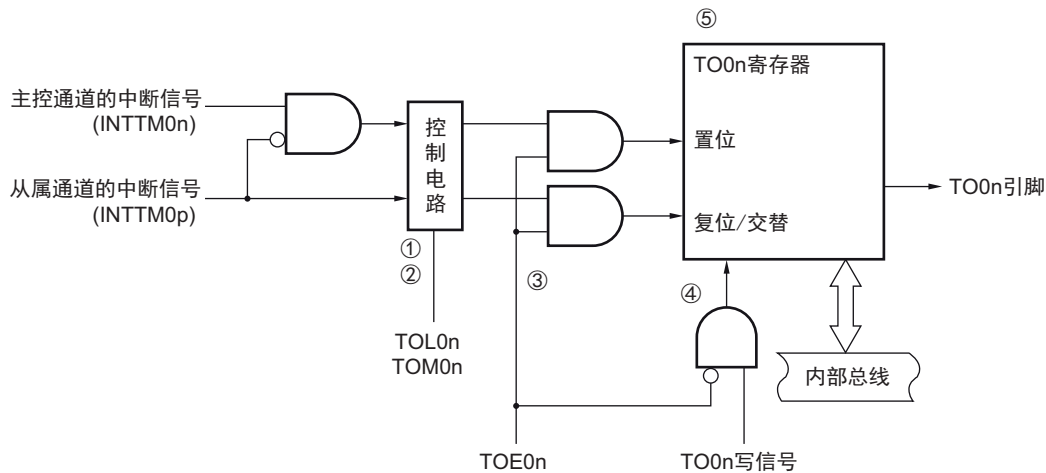


备注 这是不使用噪声滤波器时的时序。如果使用噪声滤波器，边沿检测就从 TI0n 输入开始再延迟 2 个 f_{MCK} 周期（合计 3 ~ 4 个周期）。1 个周期的误差是因为 TI0n 输入与计数时钟（ f_{MCK} ）不同步。

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 6-26 输出电路的结构



以下说明 TO0n 引脚的输出电路。

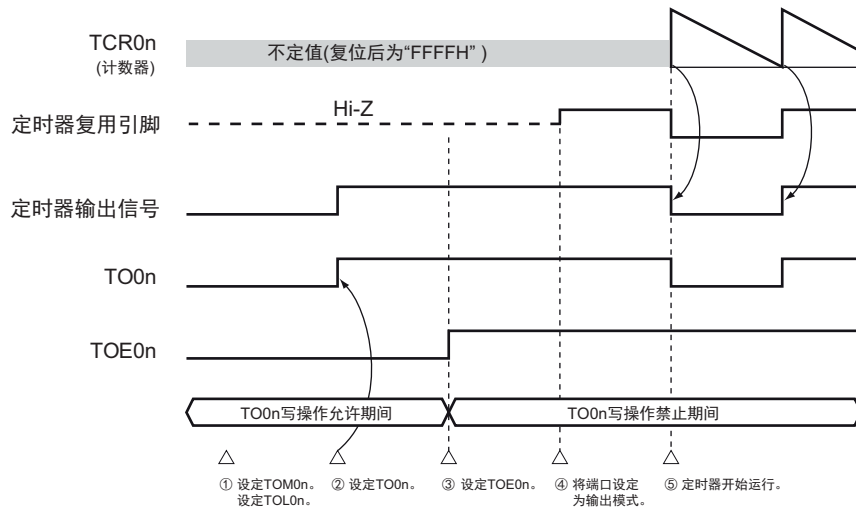
- ① 当 TOM0n 位为“0”（主控通道输出模式）时，忽视定时器输出电平寄存器 0（TOL0）的设定值，只将 INTTM0p（从属通道定时器中断）传给定时器输出寄存器 0（TO0）。
- ② 当 TOM0n 位为“1”（从属通道输出模式）时，将 INTTM0n（主控通道定时器中断）和 INTTM0p（从属通道定时器中断）传给 TO0 寄存器。
此时，TOL0 寄存器有效并且进行以下信号的控制：
TOL0n=0 时：正逻辑输出（INTTM0n→置位、INTTM0p→复位）
TOL0n=1 时：负逻辑输出（INTTM0n→复位、INTTM0p→置位）
当同时产生 INTTM0n 和 INTTM0p 时（PWM 输出的 0% 输出），优先 INTTM0p（复位信号）而屏蔽 INTTM0n（置位信号）。
- ③ 在允许定时器输出（TOE0n=1）的状态下，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，需要在设定为禁止定时器输出（TOE0n=0）后给 TO0 寄存器写值。
- ④ 在禁止定时器输出（TOE0n=0）的状态下，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=0~7（主控通道：n=0、2、4、6）
p: 从属通道号
n < p ≤ 7

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 6-27 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 (PMCxx) 将端口设定为数字输入/输出 (参照“6.3.14 控制定时器输入/输出引脚端口功能的寄存器”)。

⑤ 将端口的输入/输出设定为输出 (参照“6.3.14 控制定时器输入/输出引脚端口功能的寄存器”)。

⑥ 允许定时器运行 (TS0n=1)。

备注 n: 通道号 (n=0 ~ 7)

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 寄存器的设定值变更

定时器的运行（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）和定时器输出电平寄存器 0（TOL0）的设定值的变更不会影响定时器的运行，能在定时器运行中更改设定值。但是，为了在各定时器的运行中从 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（INTTM0n）信号前后更改除了 TO0 寄存器以外的 TOE0 寄存器和 TOL0 寄存器的设定值，就根据是在产生定时器中断（INTTM0n）信号前更改还是在产生后更改，TO0n 引脚输出的波形可能不同。

备注 n：通道号（n=0～7）

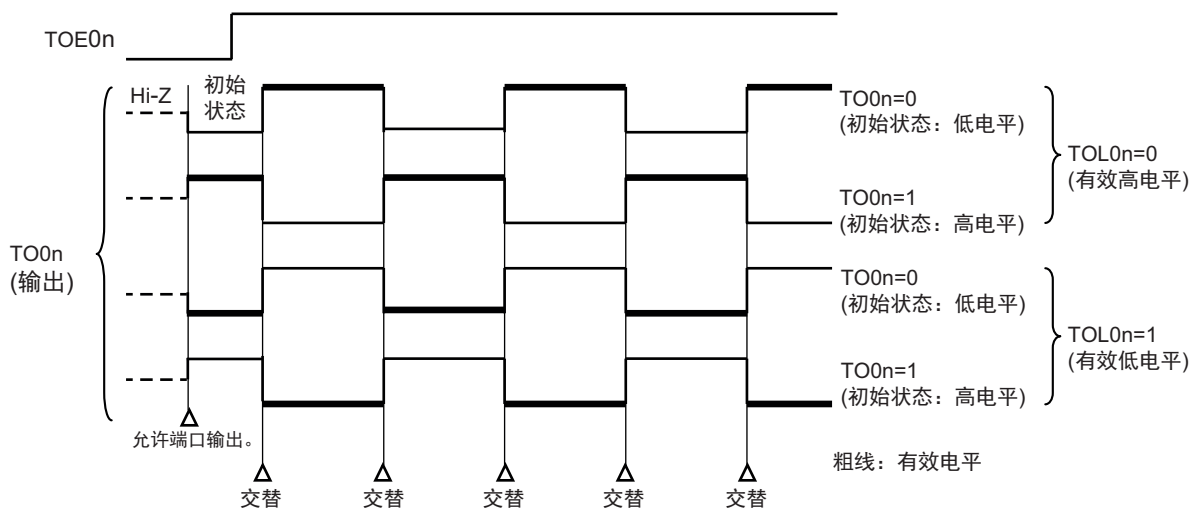
(2) 有关 TO0n 引脚的初始电平和定时器开始运行后的输出电平

在允许端口输出前并且在禁止定时器输出（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 6-28 交替输出时（TOM0n=0）的 TO0n 引脚输出状态



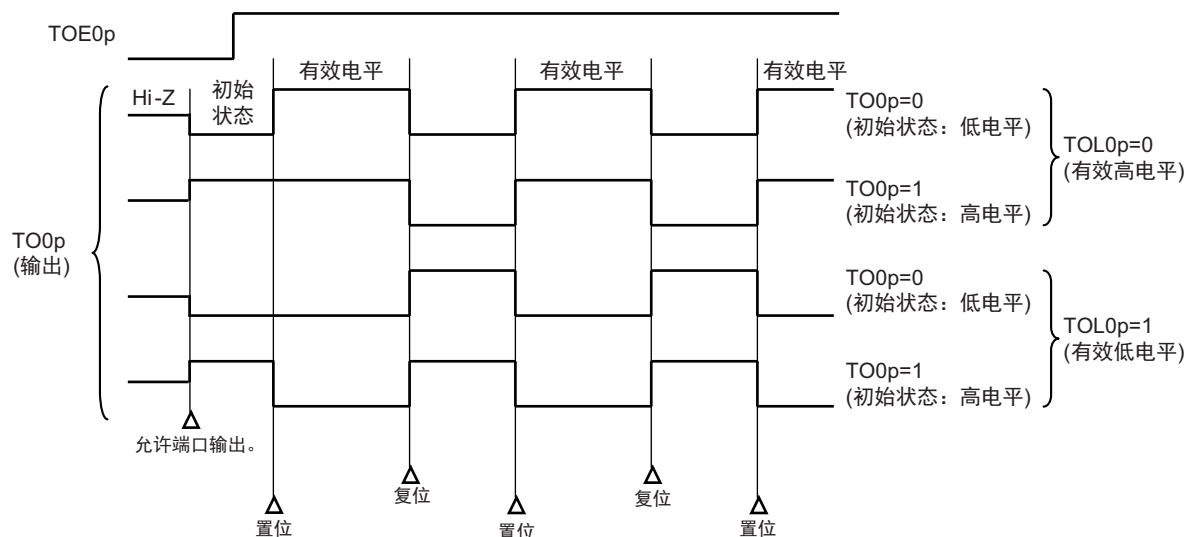
备注 1. 交替：反相 TO0n 引脚的输出状态。

2. n：通道号（n=0～7）

(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 6-29 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 ($n < p \leq 7$)

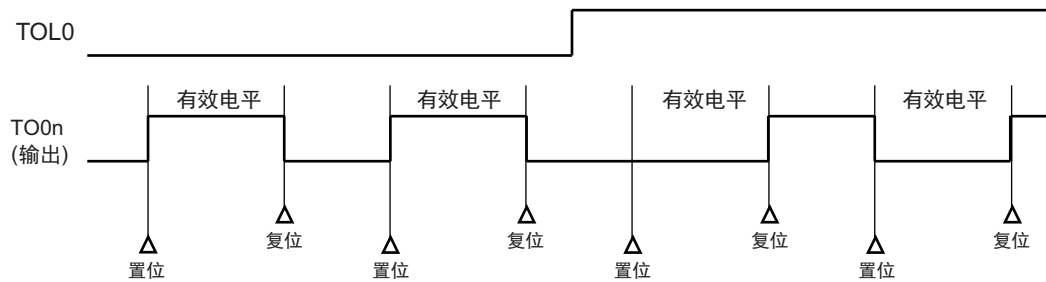
(3) 有关从属通道输出模式 (TOM0n=1) 的 TO0n 引脚变化

(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, 就在产生 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 6-30 在定时器运行中更改 TOL0 寄存器内容时的运行



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 ~ 7)

(b) 置位 / 复位时序

为了在 PWM 输出时实现 0% 和 100% 的输出, 通过从属通道将产生主控通道定时器中断 (INTTM0n) 时的 TO0n 引脚 / TO0n 位的置位时序延迟 1 个计数时钟。

当置位条件和复位条件同时产生时, 优先复位条件。

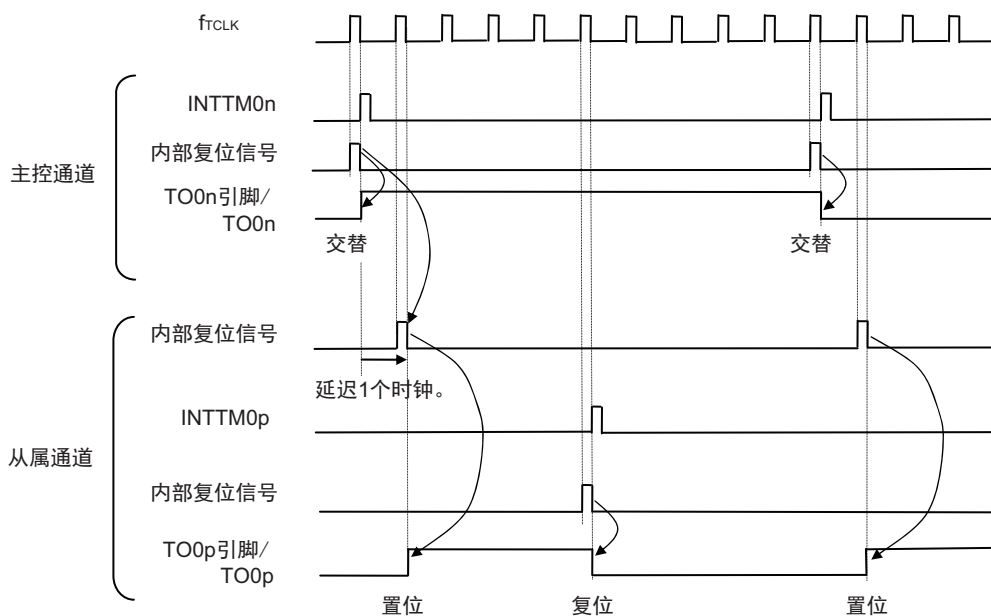
按照以下方法设定主控 / 从属通道时的置位 / 复位运行状态如图 6-31 所示。

主控通道: TOE0n=1、TOM0n=0、TOL0n=0

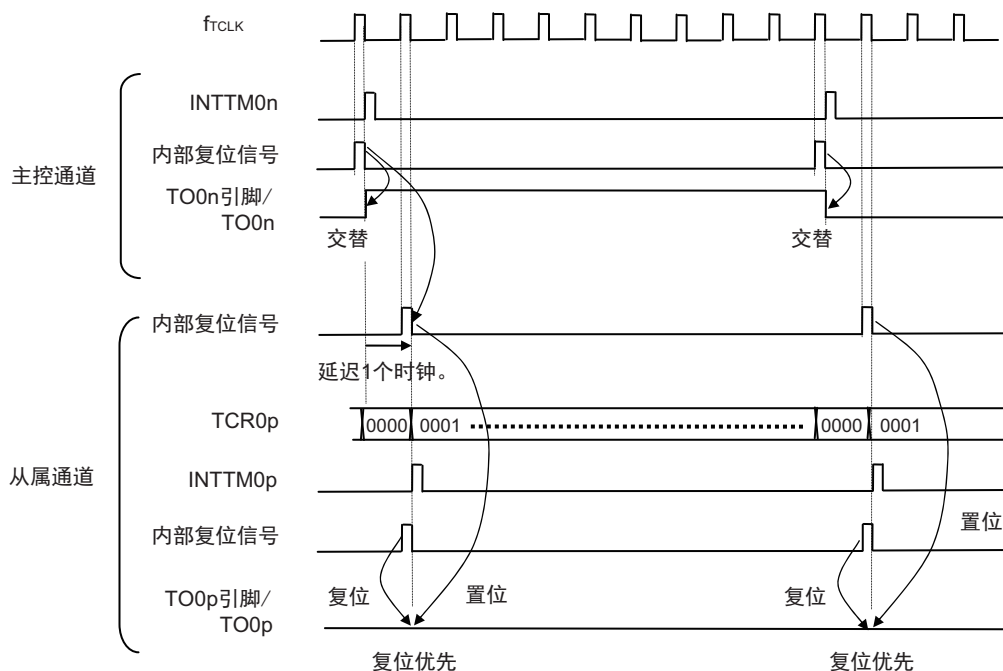
从属通道: TOE0p=1、TOM0p=1、TOL0p=0

图 6-31 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 : 通道号

$n=0\sim7$ (主控通道: $n=0, 2, 4, 6$)

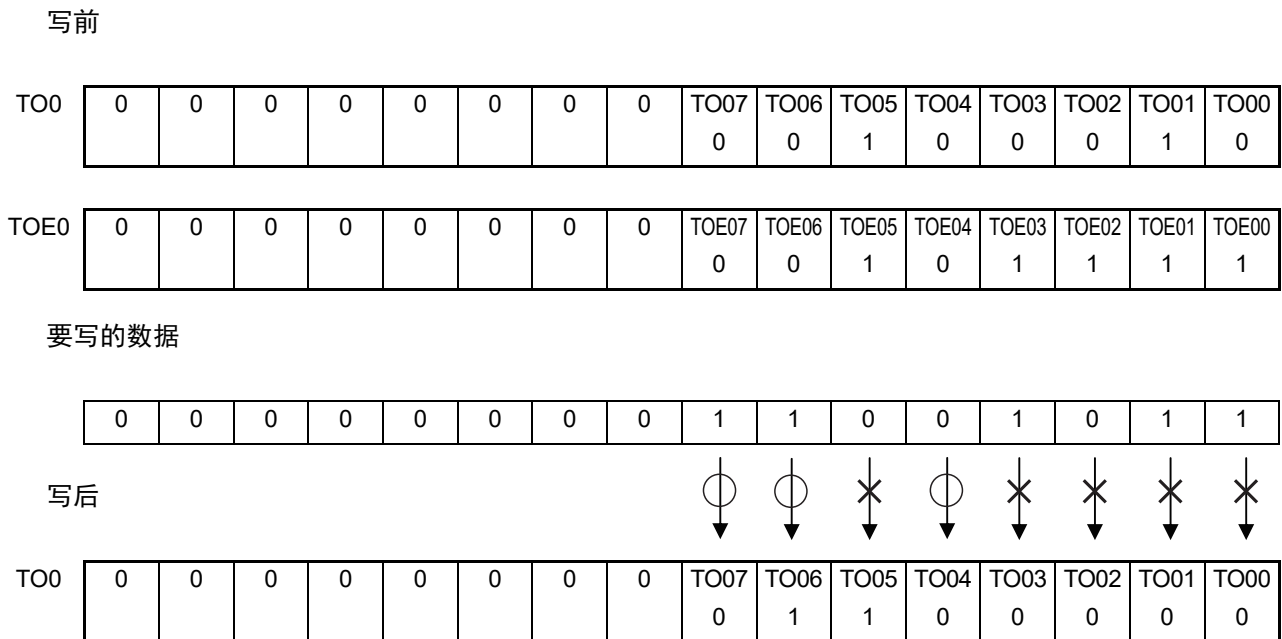
p : 从属通道号

$n < p \leq 7$

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, 能给只想作为操作对象的通道输出 (TO0n) 的 TO0n 位写数据 (TOE0n=0), 因此能操作任意的位。

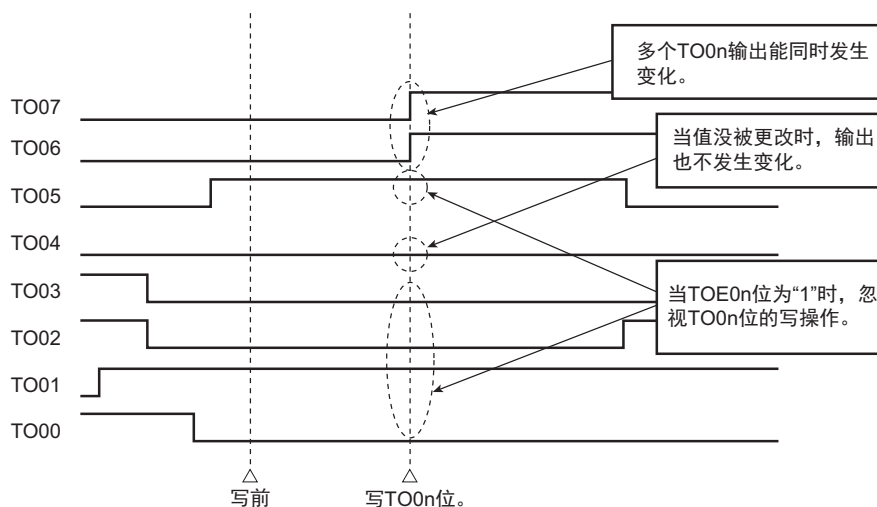
图 6-32 TO0n 位的一次性操作例子



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 6-33 一次性操作 TO0n 位时的 TO0n 引脚状态



注意 在允许定时器输出 (TOE0n=1) 的状态下, 即使各通道的定时器中断 (INTTM0n) 引起的输出和 TO0n 位的写操作发生竞争, TO0n 引脚也能正常输出。

备注 n: 通道号 (n=0 ~ 7)

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。

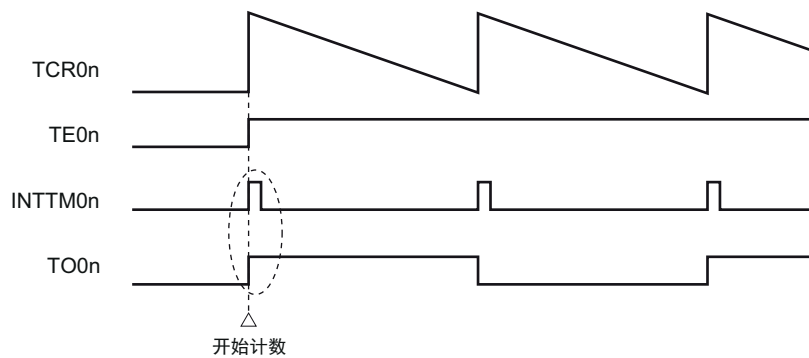
当 MD0n0 位为“1”时，能通过产生定时器中断（INTTM0n）得知计数的开始时序。

在其他模式中，不控制开始计数时的定时器中断和 TO0n 输出。

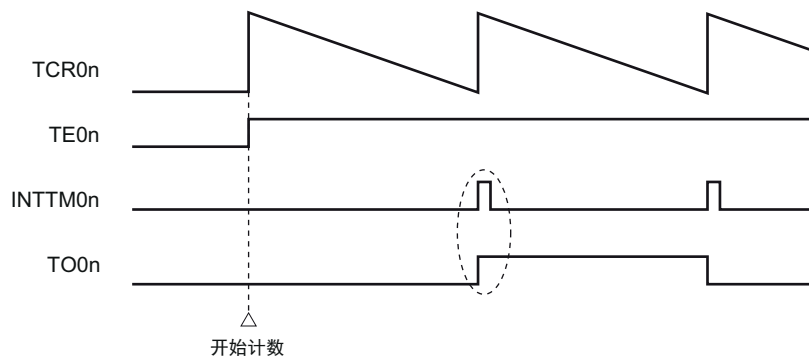
设定为间隔定时器模式（TOE0n=1、TOM0n=0）时的运行例子如下所示。

图 6-34 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。

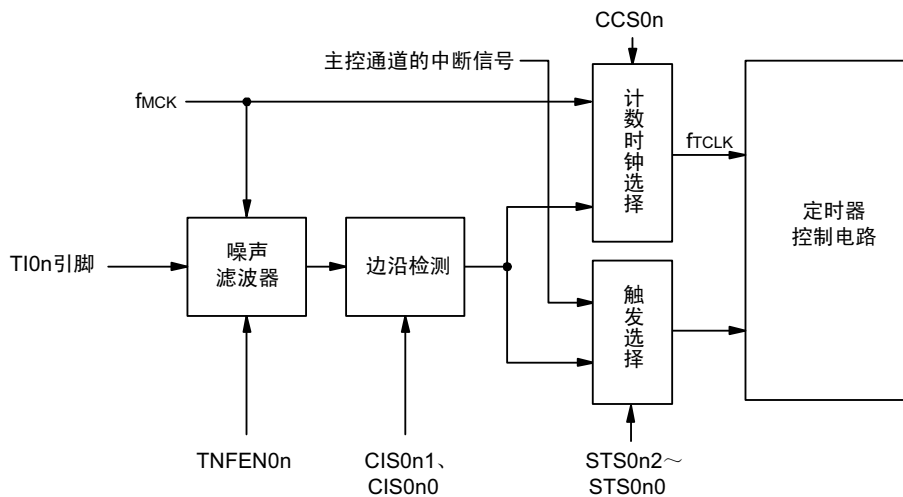
备注 n: 通道号 (n=0 ~ 7)

6.7 定时器输入 (TI0n) 的控制

6.7.1 TI0n 引脚输入电路的结构

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

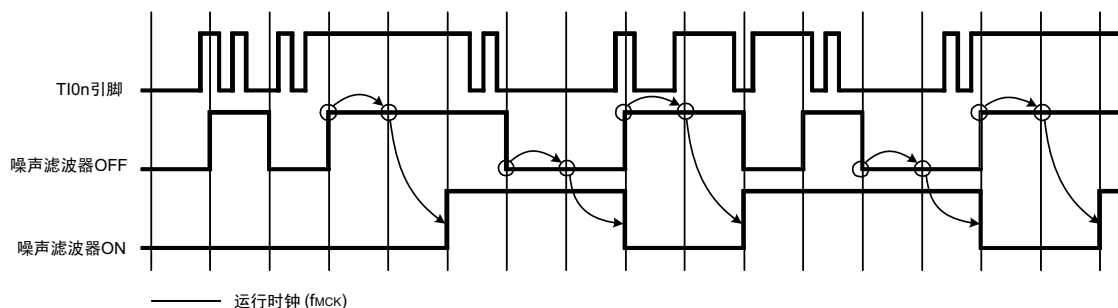
图 6-35 输入电路的结构



6.7.2 噪声滤波器

当噪声滤波器无效时，只通过通道 n 的运行时钟 (f_{MCK}) 进行同步；当噪声滤波器有效时，在通过通道 n 的运行时钟 (f_{MCK}) 进行同步后检测 2 个时钟是否一致。TI0n 输入引脚在噪声滤波器 ON 或者 OFF 的情况下，经过噪声滤波器电路后的波形如下所示。

图 6-36 TI0n 输入引脚在噪声滤波器 ON 或者 OFF 情况下的采样波形



6.7.3 操作通道输入时的注意事项

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

(1) 噪声滤波器为 OFF 的情况

如果在定时器模式寄存器 0n (TMR0n) 的 bit12 (CCS0n)、bit9 (STS0n1) 和 bit8 (STS0n0) 都为“0”的状态下将任意一位置位，就必须至少在经过 2 个运行时钟 (f_{MCK}) 周期后将定时器通道开始寄存器 (TS0) 的运行允许触发置位。

(2) 噪声滤波器为 ON 的情况

如果在定时器模式寄存器 0n (TMR0n) 的 bit12 (CCS0n)、bit9 (STS0n1) 和 bit8 (STS0n0) 都为“0”的状态下将任意一位置位，就必须至少在经过 4 个运行时钟 (f_{MCK}) 周期后将定时器通道开始寄存器 (TS0) 的运行允许触发置位。

6.8 定时器阵列单元的独立通道运行功能

6.8.1 作为间隔定时器 / 方波输出的运行

(1) 间隔定时器

能用作以固定间隔产生 INTTM0n（定时器中断）的基准定时器。
中断产生周期能用以下计算式进行计算：

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

(2) 作为方波输出的运行

TO0n 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：

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

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

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、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。

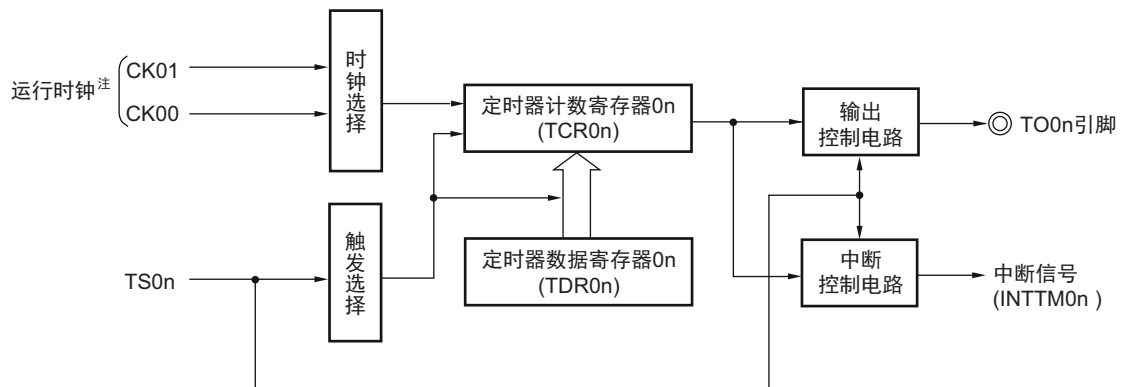
然后，TCR0n 寄存器通过计数时钟进行递减计数。

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。

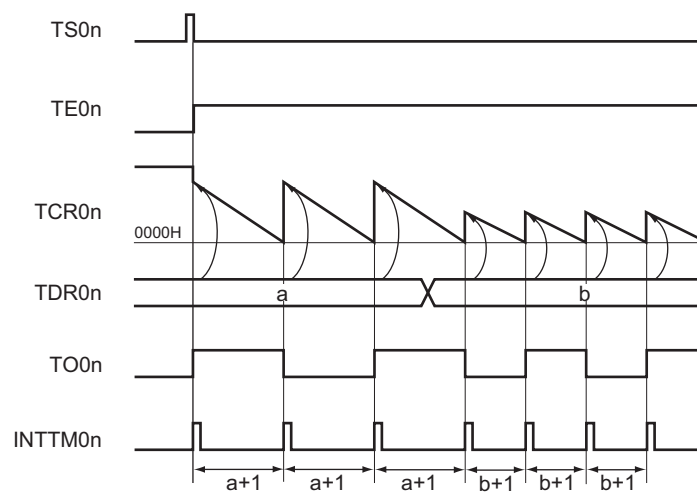
备注 n: 通道号 (n=0 ~ 7)

图 6-37 作为间隔定时器 / 方波输出运行的框图



注 在通道 1 和通道 3 时，能从 CK00 ~ CK03 中选择时钟。

图 6-38 作为间隔定时器 / 方波输出运行的基本时序例子 (MD0n0=1)

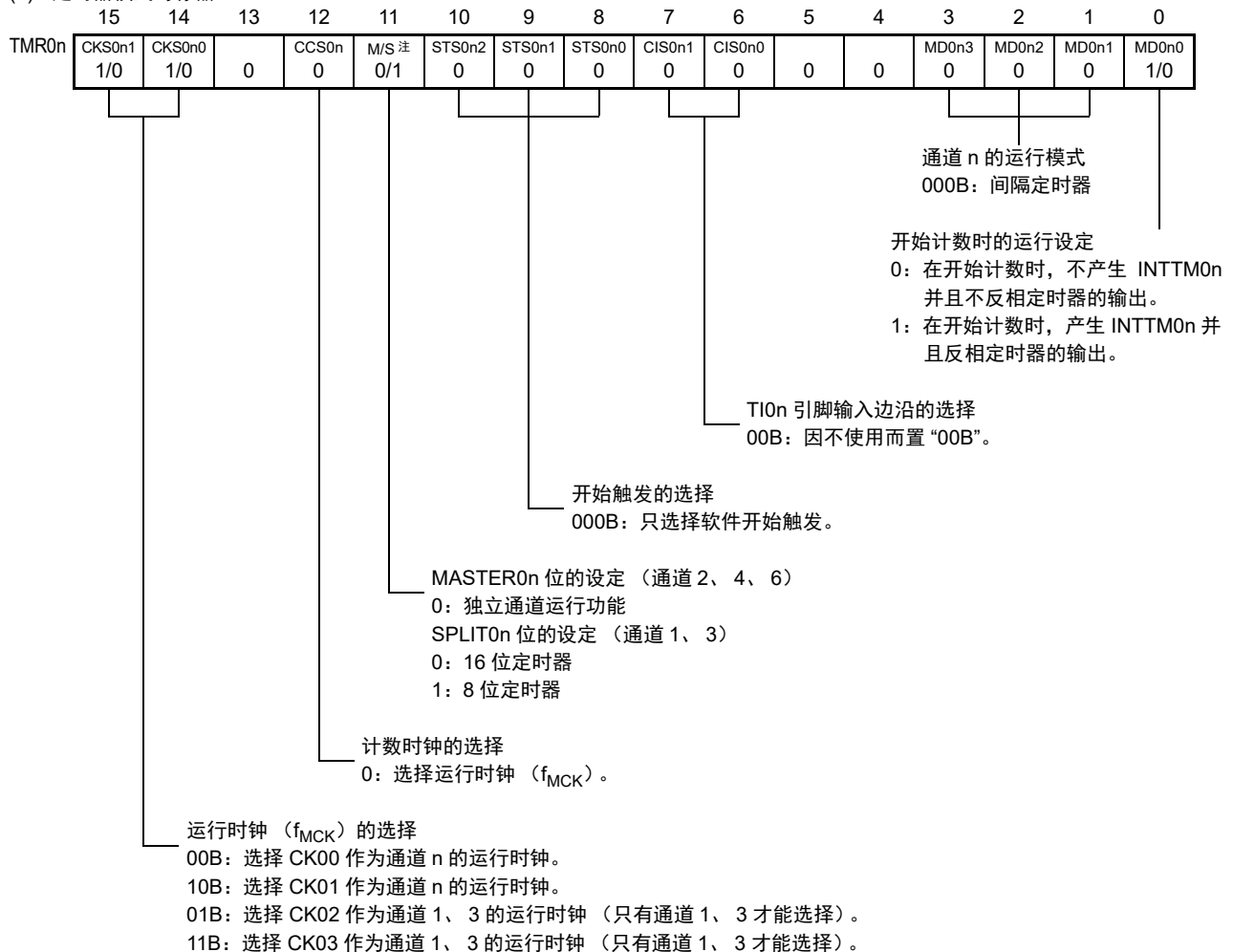


备注 1. n: 通道号 (n=0 ~ 7)

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 6-39 间隔定时器 / 方波输出时的寄存器设定内容例子

(a) 定时器模式寄存器 0n (TMR0n)



(b) 定时器输出寄存器 0 (TO0)

	bit n	
TO0	TO0n 1/0	0: 由 TO0n 输出“0”。 1: 由 TO0n 输出“1”。

(c) 定时器输出允许寄存器 0 (TOE0)

	bit n	
TOE0	TOE0n 1/0	0: 停止由计数运行进行的 TO0n 输出。 1: 允许由计数运行进行的 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) 时置“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、TMR04、TMR06: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00、TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0 ~ 7)

图 6-40 间隔定时器 / 方波输出功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 (在 8 位定时器模式中为 CK02 和 CK03) 的时钟频率。	
通道初 始设定	设定定时器模式寄存器 0n (TMR0n) (确定通道的运行模式)。 给定时器数据寄存器 0n (TDR0n) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	使用 TO0n 输出的情况: 将定时器输出模式寄存器 0 (TOM0) 的 TOM0n 位置“0” (主控通道输出模式)。 将 TOL0n 位置“0”。 设定 TO0n 位, 确定 TO0n 输出的初始电平。 将 TOE0n 位置“1”, 允许 TO0n 输出。 将端口寄存器和端口模式寄存器置“0”。	TO0n 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TO0n 初始设定的电平。 因为通道处于运行停止状态, 所以 TO0n 不变。 TO0n 引脚输出 TO0n 设定的电平。
重新 开始 运行	开始运行 (只在使用 TO0n 输出并且重新开始时, 将 TOE0n 位置“1”)。 将 TS0n (TSH01、TSH03) 位置“1”。 因为 TS0n (TSH01、TSH03) 位是触发位, 所以自动返回到“0”。	TE0n (TEH01、TEH03) 位变为“1”并且开始计数。 通过输入计数时钟, 将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n)。当 TMR0n 寄存器的 MD0n0 位为“1”时, 产生 INTTM0n 并且 TO0n 进行交替输出。
	运行中 能任意更改 TDR0n 寄存器的设定值。 能随时读 TCR0n 寄存器。 不使用 TSR0n 寄存器。 禁止更改 TMR0n 寄存器、TOM0n 位和 TOL0n 位的设定值。	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 TT0n (TTH01、TTH03) 位置“1”。 因为 TT0n (TTH01、TTH03) 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。	TE0n (TEH01、TEH03) 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。 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持有的值后将 TO0n 位置“0”。 不需要保持 TO0n 引脚输出电平的情况: 不需要设定。	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 对全部电路和各通道的 SFR 进行初始化。 (TO0n 位变为“0”并且 TO0n 引脚变为端口功能)

备注 n: 通道号 (n=0~7)

6.8.2 作为外部事件计数器的运行

能用作事件计数器，对检测到的 $TI0n$ 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，就产生中断。规定的计数值能用以下计算式进行计算：

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

在事件计数器模式中，定时器计数寄存器 $0n$ （TCR $0n$ ）用作递减计数器。

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

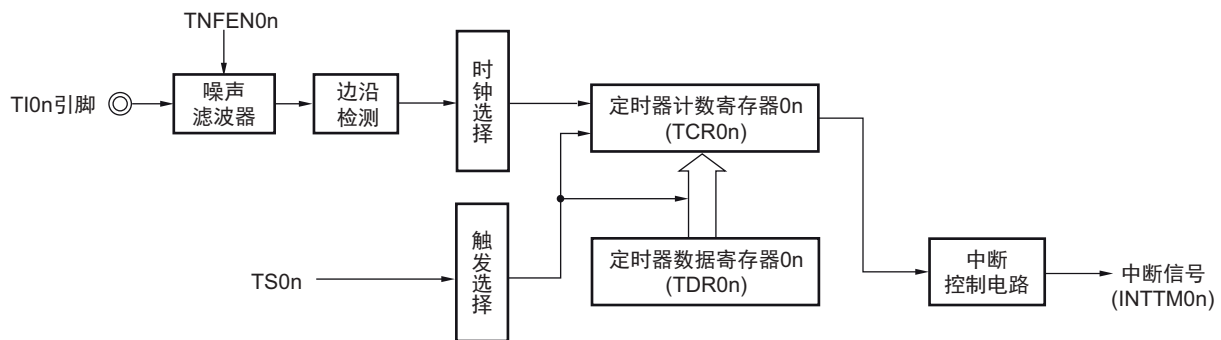
TCR $0n$ 寄存器在检测到 $TI0n$ 引脚输入的有效边沿的同时进行递减计数。如果 TCR $0n$ 变为“0000H”，就再次装入 TDR $0n$ 寄存器的值并且输出 INTTM $0n$ 。

此后，继续同样的运行。

因为 TO $0n$ 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 （TOE 0 ）的 TOE $0n$ 位置“0”，停止输出。

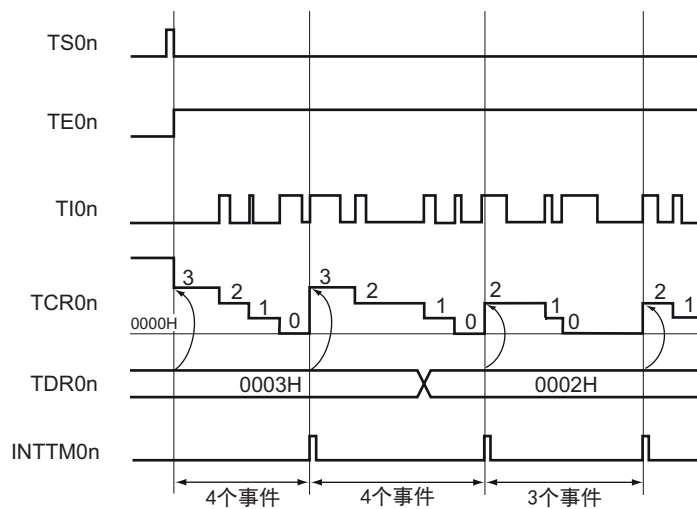
能随时改写 TDR $0n$ 寄存器，改写的 TDR $0n$ 寄存器的值在下一个计数期间有效。

图 6-41 作为外部事件计数器运行的框图



备注 n: 通道号 (n=0 ~ 7)

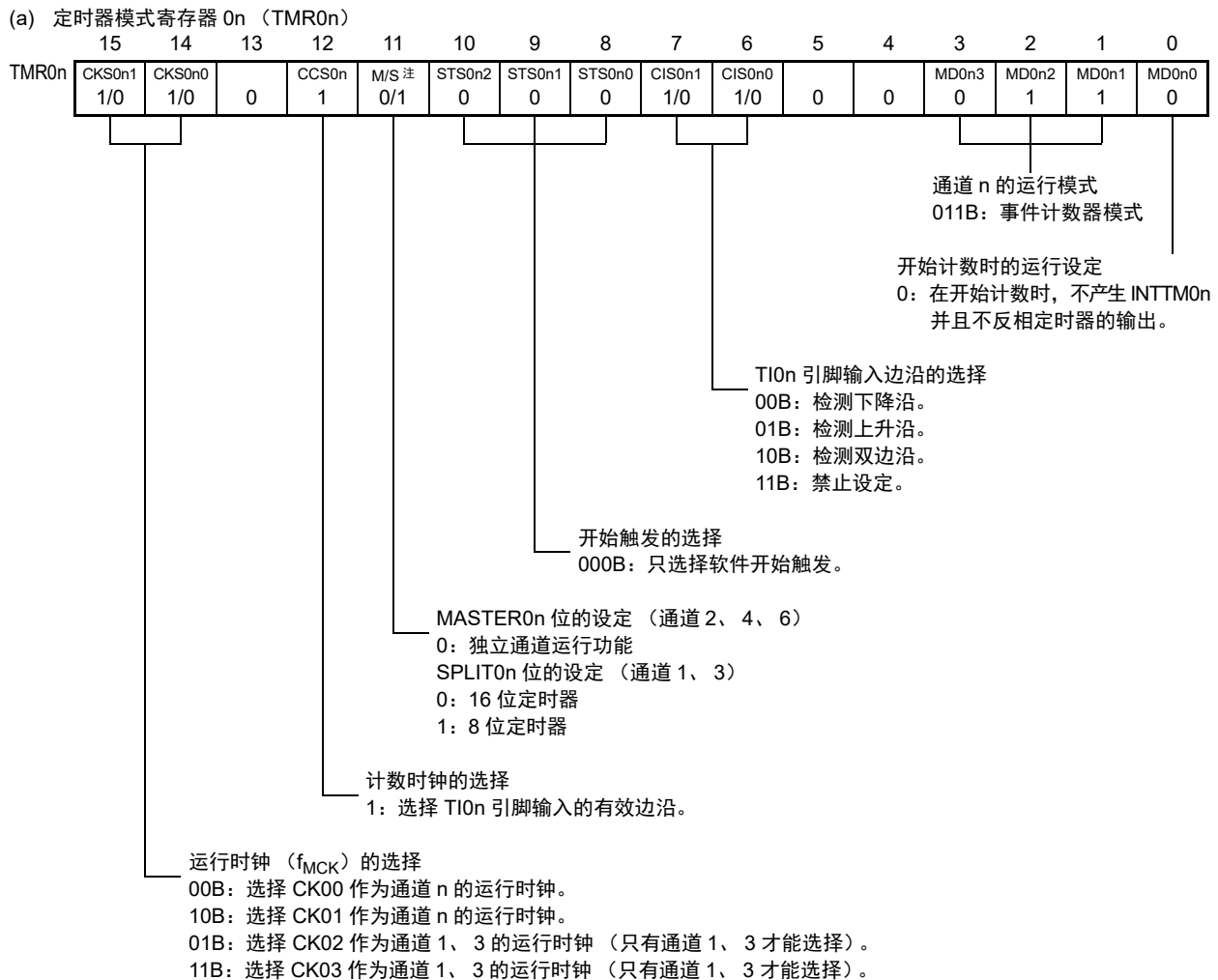
图 6-42 作为外部事件计数器运行的基本时序例子



备注 1. n: 通道号 (n=0 ~ 7)

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 6-43 外部事件计数器模式时的寄存器设定内容例子



(b) 定时器输出寄存器 0 (TO0)

bit n
 TO0n

TO0n
0

 0: 由 TO0n 输出“0”。

(c) 定时器输出允许寄存器 0 (TOE0)

bit n
 TOE0n

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
 TOL0n

TOL0n
0

 0: 在主导通道输出模式 (TOM0n=0) 时置“0”。

(e) 定时器输出模式寄存器 0 (TOM0)

bit n
 TOM0n

TOM0n
0

 0: 设定主导通道输出模式。

注 TMR02、TMR04、TMR06: MASTER0n 位
 TMR01、TMR03: SPLIT0n 位
 TMR00、TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0~7)

图 6-44 外部事件计数器功能时的操作步骤

	软件操作	硬件状态	
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)	
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)	
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 (在 8 位定时器模式中为 CK02 和 CK03) 的时钟频率。		
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置 “0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 0n (TMR0n) (确定通道的 运行模式)。 给定时器数据寄存器 0n (TDR0n) 设定计数值。 设定噪声滤波器允许寄存器 1 (NFEN1)。 将定时器输出允许寄存器 0 (TOE0) 的 TOE0n 位 置“0”。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)	
重新 开始 运行	开始 运行	将 TS0n 位置“1”。 因为 TS0n 位是触发位, 所以自动返回到“0”。	TE0n 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 引脚输入边沿的检测等待状 态。
	运行中	能任意更改 TDR0n 寄存器的设定值。 能随时读 TCR0n 寄存器。 不使用 TSR0n 寄存器。 禁止更改 TMR0n 寄存器、TOM0n 位、TOL0n 位、 TO0n 位和 TOE0n 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 TT0n 位置“1”。 因为 TT0n 位是触发位, 所以自动返回到“0”。	TE0n 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 对全部电路和各通道的 SFR 进行初始化。	

备注 n: 通道号 (n=0 ~ 7)

6.8.3 作为分频器的运行

能对 TI00 引脚输入的时钟进行分频并且用作 TO00 引脚输出的分频器。

TO00 输出的分频时钟频率能用以下计算式进行计算：

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

在间隔定时器模式中，定时器计数寄存器 00（TCR00）用作递增计数器。

在将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS00）置“1”后，通过检测到 TI00 的有效边沿将定时器数据寄存器 00（TDR00）的值装入 TCR00 寄存器。此时，如果定时器模式寄存器 00（TMR00）的 MD000 位为“0”，就不输出 INTTM00 并且 TO00 不进行交替输出；如果 TMR00 寄存器的 MD000 位为“1”，就输出 INTTM00 并且 TO00 进行交替输出。

然后，TCR00 寄存器通过 TI00 引脚输入的有效边沿进行递减计数。如果 TCR00 变为“0000H”，TO00 就进行交替输出。同时，将 TDR00 寄存器的值装入 TCR00 寄存器并且继续计数。

如果选择 TI00 引脚输入的双边沿检测，输入时钟的占空比误差就会影响 TO00 输出的分频时钟周期。

TO00 输出的时钟周期包含 1 个运行时钟周期的采样误差。

$$\text{TO00 输出的时钟周期} = \text{理想的 TO00 输出时钟周期} \pm \text{运行时钟周期（误差）}$$

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

图 6-45 作为分频器运行的框图

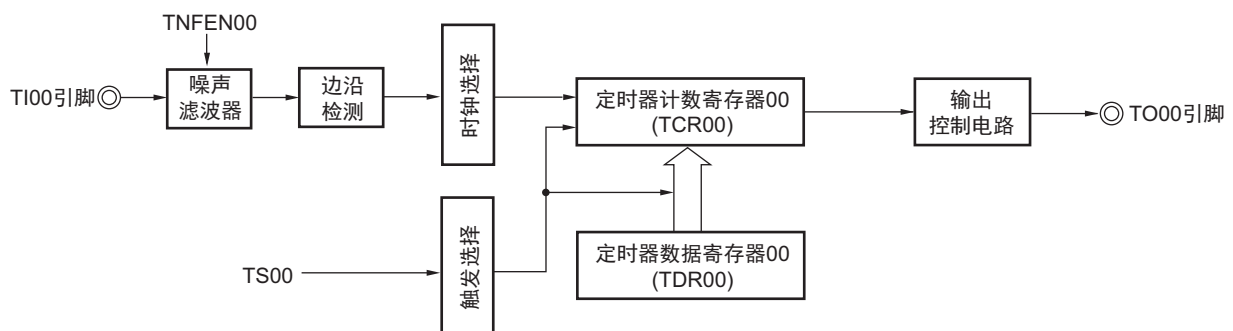
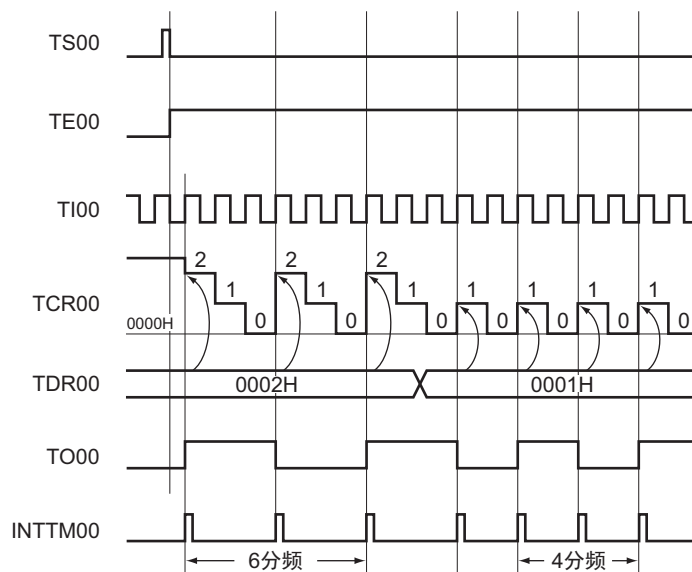


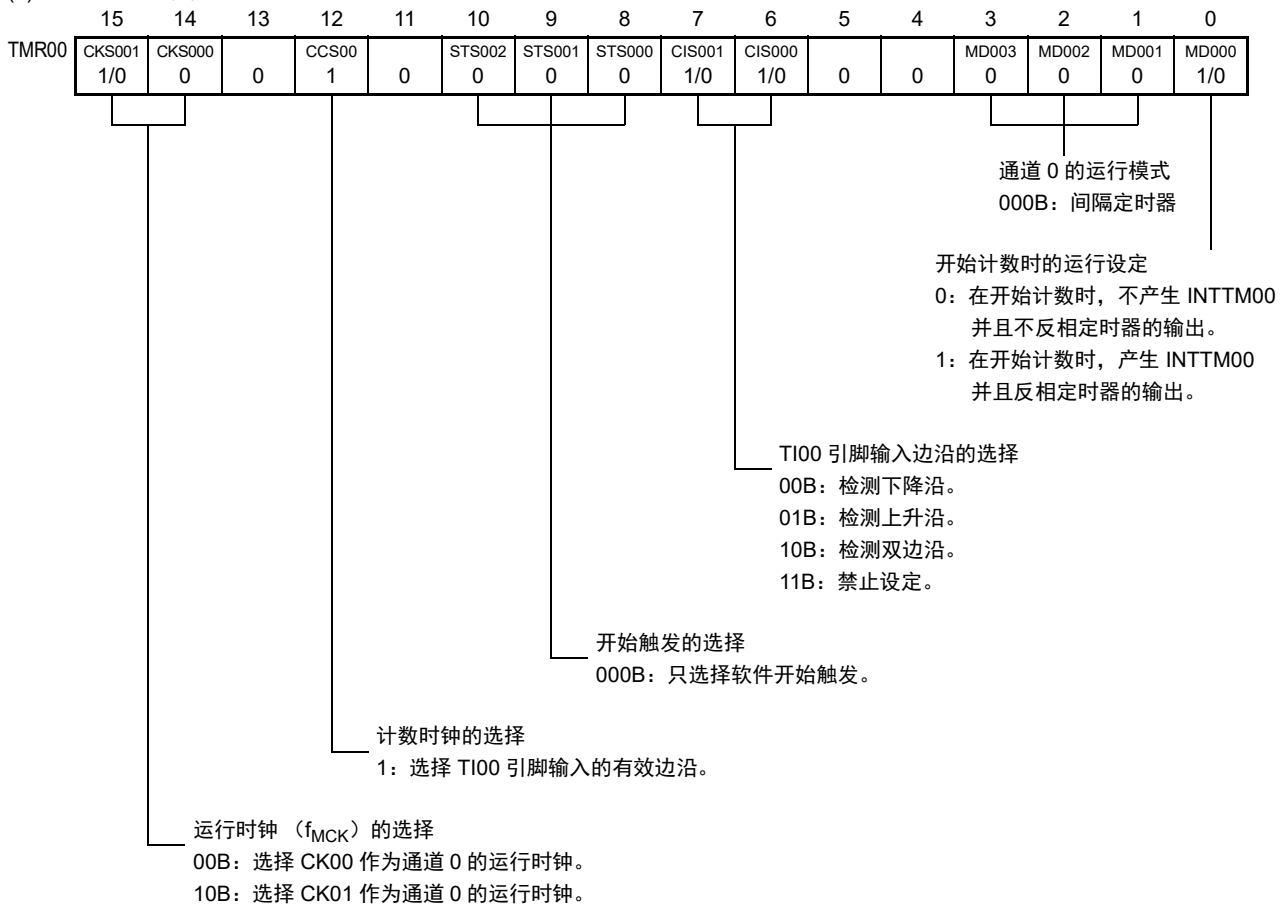
图 6-46 作为分频器运行的基本时序例子 (MD000=1)



- 备注
- TS00: 定时器通道开始寄存器 0 (TS0) 的 bit0
 - TE00: 定时器通道允许状态寄存器 0 (TE0) 的 bit0
 - TI00: TI00 引脚输入信号
 - TCR00: 定时器计数寄存器 00 (TCR00)
 - TDR00: 定时器数据寄存器 00 (TDR00)
 - TO00: TO00 引脚输出信号

图 6-47 作为分频器运行时的寄存器设定内容例子

(a) 定时器模式寄存器 00 (TMR00)



(b) 定时器输出寄存器 0 (TO0)

	bit0	
TO0	TO00 1/0	0: 由 TO00 输出“0”。 1: 由 TO00 输出“1”。

(c) 定时器输出允许寄存器 0 (TOE0)

	bit0	
TOE0	TOE00 1/0	0: 停止由计数运行进行的 TO00 输出。 1: 允许由计数运行进行的 TO00 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

	bit0	
TOL0	TOL00 0	0: 在主导通道输出模式 (TOM00=0) 中置“0”。

(e) 定时器输出模式寄存器 0 (TOM0)

	bit0	
TOM0	TOM00 0	0: 设定主导通道输出模式。

图 6-48 分频器功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 00 (TMR00) (确定通道的运行模式, 选择检测边沿)。 给定时器数据寄存器 00 (TDR00) 设定间隔 (周期) 值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	将定时器输出模式寄存器 0 (TOM0) 的 TOM00 位置“0” (主控通道输出模式)。 将 TOL00 位置“0”。 设定 TO00 位并且确定 TO00 输出的初始电平。	TO00 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为“0”时, 输出 TO00 初始设定的电平。
	将 TOE00 位置“1”, 允许 TO00 输出。	因为通道处于运行停止状态, 所以 TO00 不变。
	将端口寄存器和端口模式寄存器置“0”。	TO00 引脚输出 TO00 设定的电平。
开始 运行	将 TOE00 位置“1” (只限于重新开始运行)。 将 TS00 位置“1”。 因为 TS00 位是触发位, 所以自动返回到“0”。	TE00 位变为“1”并且开始计数。 将 TDR00 寄存器的值装入定时器计数寄存器 00 (TCR00)。当 TMR00 寄存器的 MD000 位为“1”时, 产生 INTTM00 并且 TO00 进行交替输出。
	运行中	能任意更改 TDR00 寄存器的设定值。 能随时读 TCR00 寄存器。 不使用 TSR00 寄存器。 能更改 TO0 寄存器和 TOE0 寄存器的设定值。 禁止更改 TMR00 寄存器、TOM00 位和 TOL00 位的设定值。
停止 运行	将 TT00 位置“1”。 因为 TT00 位是触发位, 所以自动返回到“0”。	TE00 位变为“0”并且停止计数。 TCR00 寄存器保持计数值而停止计数。 TO00 输出不被初始化而保持状态。
	将 TOE00 位置“0”并且给 TO00 位设定值。	TO00 引脚输出 TO00 设定的电平。
TAU 停止	要保持 TO00 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 TO00 位置“0”。	通过端口功能保持 TO00 引脚的输出电平。
	不需要保持 TO00 引脚输出电平的情况: 不需要设定。 将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 对全部电路和各通道的 SFR 进行初始化。 (TO00 位变为“0”并且 TO00 引脚变为端口功能)

重新开始运行

6.8.4 作为输入脉冲间隔测量的运行

能在 TI0n 有效边沿捕捉计数值，测量 TI0n 输入脉冲的间隔。在 TE0n 位为“1”的期间，也能将软件操作 (TS0n=1) 设定为捕捉触发，捕捉计数值。

脉冲间隔能用以下计算式进行计算：

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

注意 因为通过定时器模式寄存器 0n (TMR0n) 的 CKS0n 位选择的运行时钟对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟的误差。

在捕捉模式中，定时器计数寄存器 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。

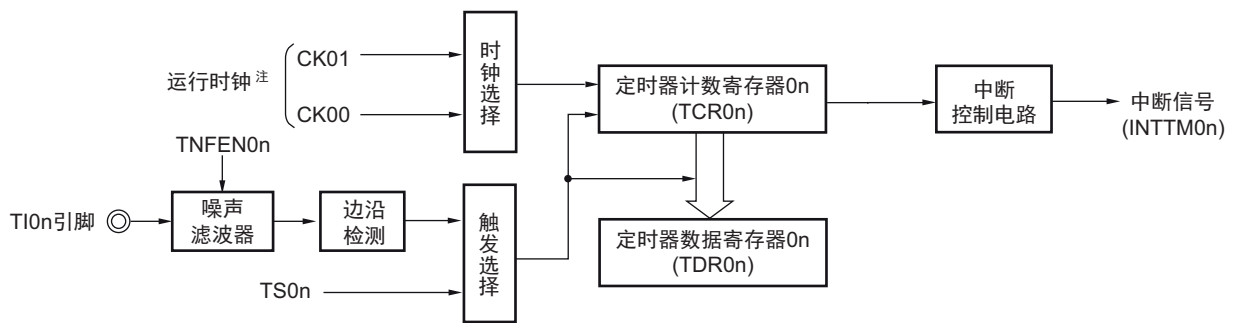
如果检测到 TI0n 引脚输入的有效边沿，就将 TCR0n 寄存器的计数值传送 (捕捉) 到定时器数据寄存器 0n (TDR0n)，同时将 TCR0n 寄存器清为“0000H”，然后输出 INTTM0n。此时，如果计数器发生上溢，就将定时器状态寄存器 0n (TSR0n) 的 OVF 位置“1”。如果计数器没有发生上溢，就清除 OVF 位。此后，继续同样的运行。

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 个周期或者 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”，并且将 TI0n 的有效边沿用于开始触发和捕捉触发。

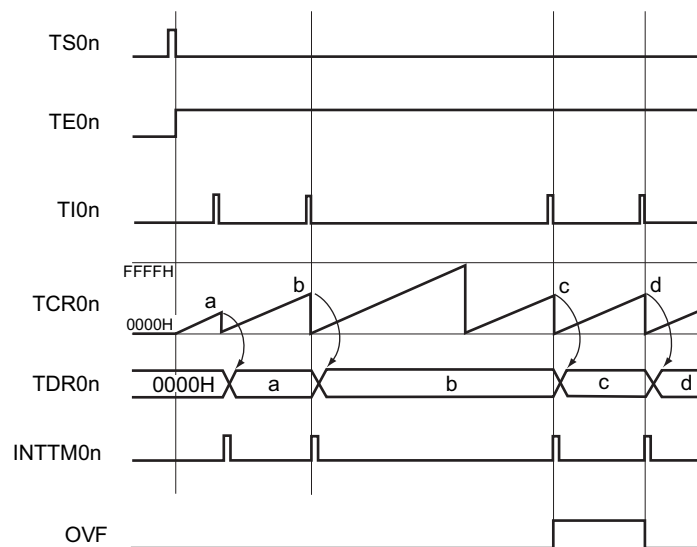
图 6-49 作为输入脉冲间隔测量运行的框图



注 在通道 1 和通道 3 时，能从 CK00、CK01、CK02 和 CK03 中选择时钟。

备注 n: 通道号 (n=0 ~ 7)

图 6-50 作为输入脉冲间隔测量的运行基本时序例子 (MD0n0=0)

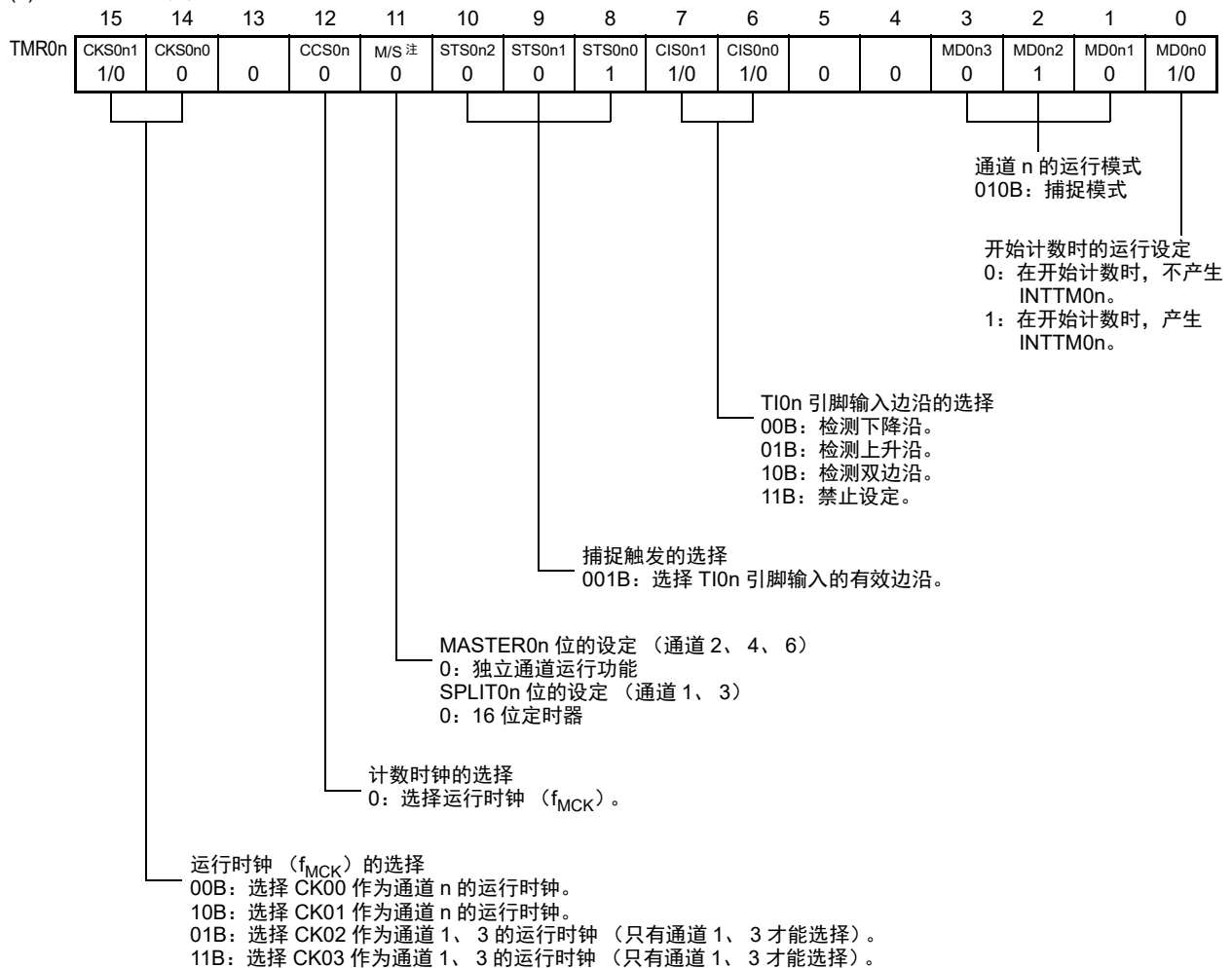


备注 1. n: 通道号 (n=0 ~ 7)

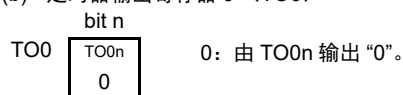
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- OVF: 定时器状态寄存器 0n (TSR0n) 的 bit0

图 6-51 测量输入脉冲间隔时的寄存器设定内容例子

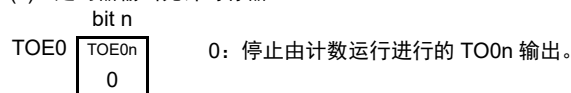
(a) 定时器模式寄存器 0n (TMR0n)



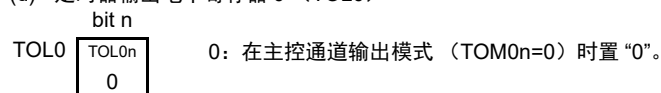
(b) 定时器输出寄存器 0 (TO0)



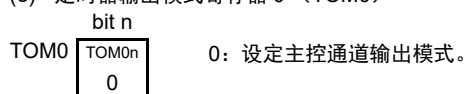
(c) 定时器输出允许寄存器 0 (TOE0)



(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、TMR04、TMR06: MASTER0n 位
 TMR01、TMR03: SPLIT0n 位
 TMR00、TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0~7)

图 6-52 输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 ~ CK03 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置 “0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 0n (TMR0n) (确定通道的 运行模式)。 设定噪声滤波器允许寄存器 1 (NFEN1)。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TS0n 位置“1”。 因为 TS0n 位是触发位, 所以自动返回到“0”。	TE0n 位变为“1”并且开始计数。 通过输入计数时钟将定时器计数寄存器 0n (TCR0n) 清为“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。 能随时读 TDR0n 寄存器。 能随时读 TCR0n 寄存器。 能随时读 TSR0n 寄存器。 禁止更改 TOM0n 位、TOL0n 位、TO0n 位和 TOE0n 位的设定值。	计数器 (TCR0n) 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 寄存器清为“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。如果没有发生上溢, 就 清除 OVF 位。 此后, 重复此运行。
停止 运行	将 TT0n 位置“1”。 因为 TT0n 位是触发位, 所以自动返回到“0”。	TE0n 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 对全部电路和各通道的 SFR 进行初始化。

重新开始运行

备注 n: 通道号 (n=0 ~ 7)

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 CKS0n 位选择的运行时钟对 TI0n 引脚输入进行采样，所以产生 1 个运行时钟的误差。

在捕捉 & 单次计数模式中，定时器计数寄存器 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 引脚输入的下降沿），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 个周期或者 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。

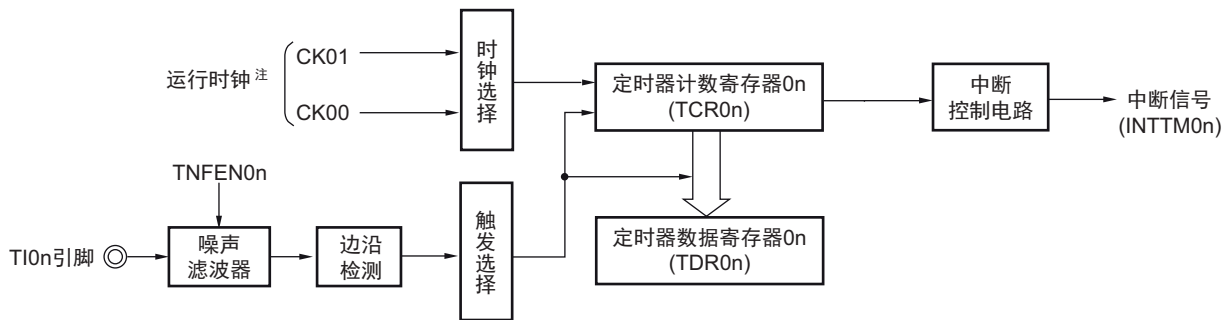
能通过 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位来设定是测量 TI0n 引脚的高电平宽度还是低电平宽度。

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。

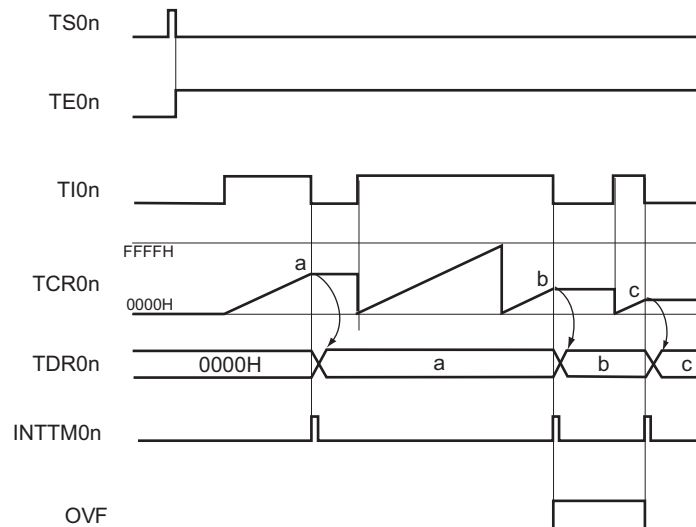
TMR0n 寄存器的 CIS0n1、CIS0n0=11B：测量高电平宽度。

图 6-53 作为输入信号高低电平宽度测量运行的框图



注 在通道 1 和通道 3 时，能从 CK00、CK01、CK02 和 CK03 中选择时钟。

图 6-54 作为输入信号高低电平宽度测量的运行基本时序例子

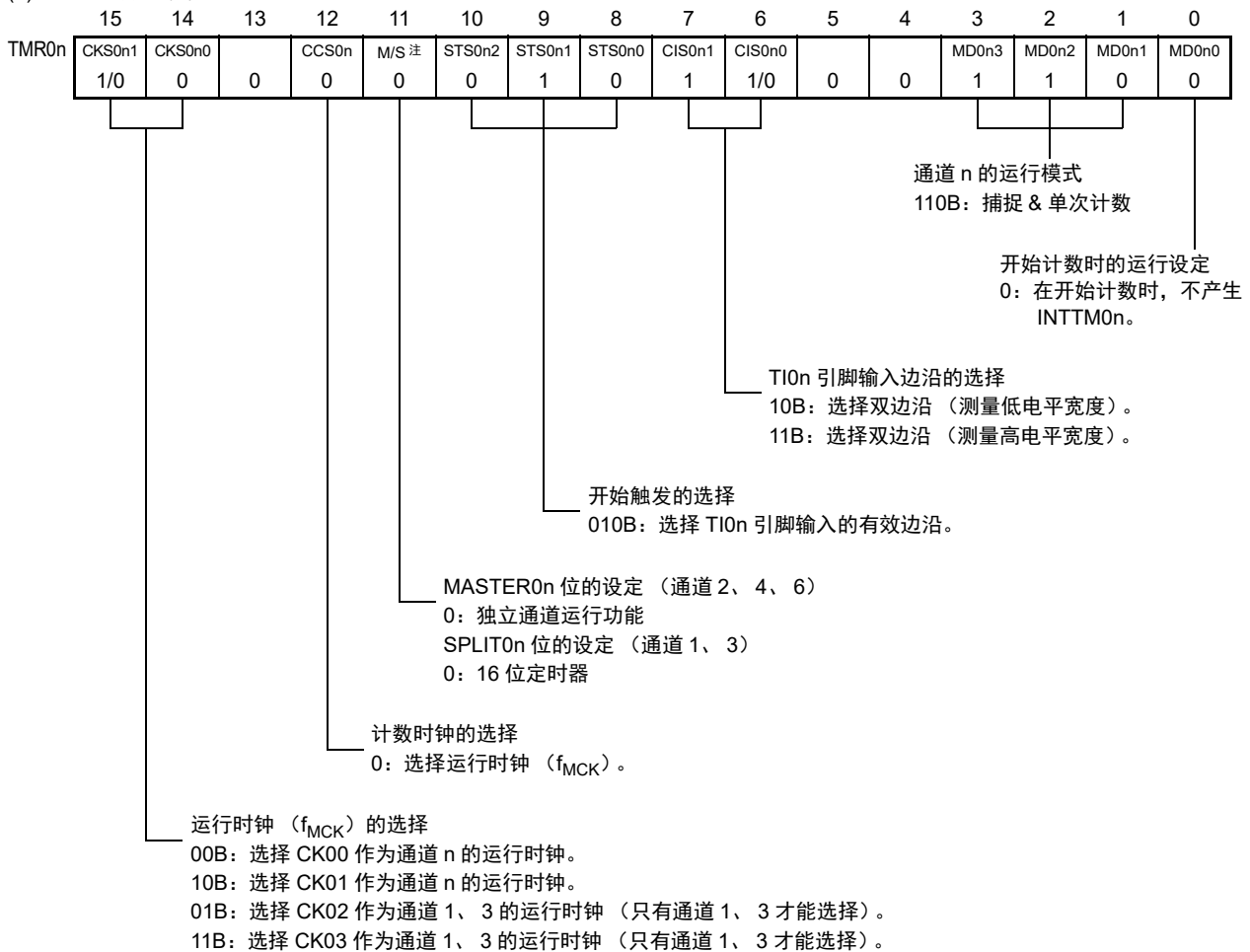


备注 1. n: 通道号 (n=0 ~ 7)

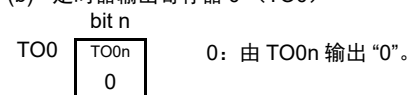
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的 bit n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的 bit n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- OVF: 定时器状态寄存器 0n (TSR0n) 的 bit0

图 6-55 测量输入信号的高低电平宽度时的寄存器设定内容例子

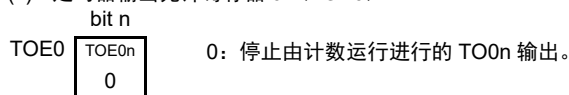
(a) 定时器模式寄存器 0n (TMR0n)



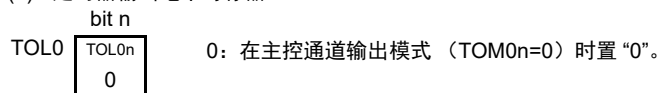
(b) 定时器输出寄存器 0 (TO0)



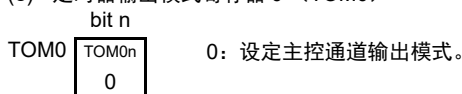
(c) 定时器输出允许寄存器 0 (TOE0)



(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、TMR04、TMR06: MASTER0n 位
 TMR01、TMR03: SPLIT0n 位
 TMR00、TMR05、TMR07: 固定为 "0"。

备注 n: 通道号 (n=0~7)

图 6-56 输入信号高低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 "1"。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 ~ CK03 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置 "0" (OFF) 或者 "1" (ON)。 设定定时器模式寄存器 0n (TMR0n) (确定通道的 运行模式)。 设定噪声滤波器允许寄存器 1 (NFEN1)。 将 TOE0n 位置 "0", 并且停止 TO0n 的运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TS0n 位置 "1"。 →	TE0n 位变为 "1" 并且进入 TI0n 引脚的开始边沿检测等待状态。
	因为 TS0n 位是触发位, 所以自动返回到 "0"。 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) 清为 "0000H" 并且开始递增计数。
运行中	能任意更改 TDR0n 寄存器的设定值。 能随时读 TCR0n 寄存器。 不使用 TSR0n 寄存器。 禁止更改 TMR0n 寄存器、TOM0n 位、TOL0n 位、 TO0n 位和 TOE0n 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 TT0n 位置 "1"。 → 因为 TT0n 位是触发位, 所以自动返回到 "0"。	TE0n 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 对全部电路和各通道的 SFR 进行初始化。

重新
开始
运行

备注 n: 通道号 (n=0 ~ 7)

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 INTTM0n（定时器中断）。

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（定时器中断）。

中断产生周期能用以下计算式进行计算：

$$\text{INTTM0n (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR0n 的设定值} + 1)$$

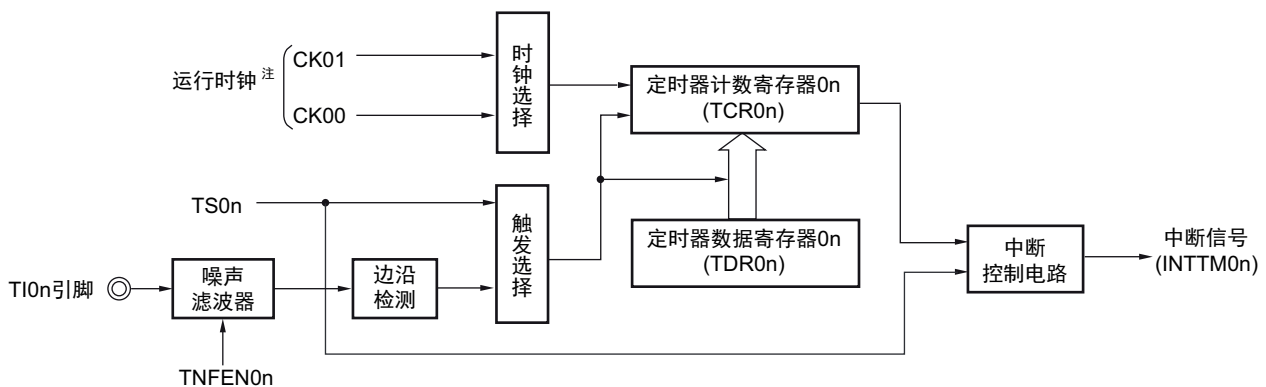
在单次计数模式中，定时器计数寄存器 0n（TCR0n）用作递减计数器。

如果将定时器通道开始寄存器 0（TS0）的通道开始触发位（TS0n、TSH01、TSH03）置“1”，TE0n 位、TEH01 位和 TEH03 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。

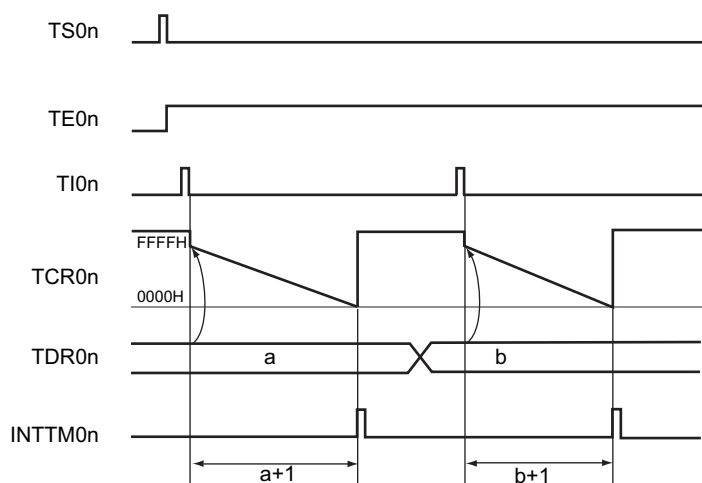
图 6-57 作为延迟计数器运行的框图



注 在通道 1 和通道 3 时，能从 CK00、CK01、CK02 和 CK03 中选择时钟。

备注 n：通道号（n=0～7）

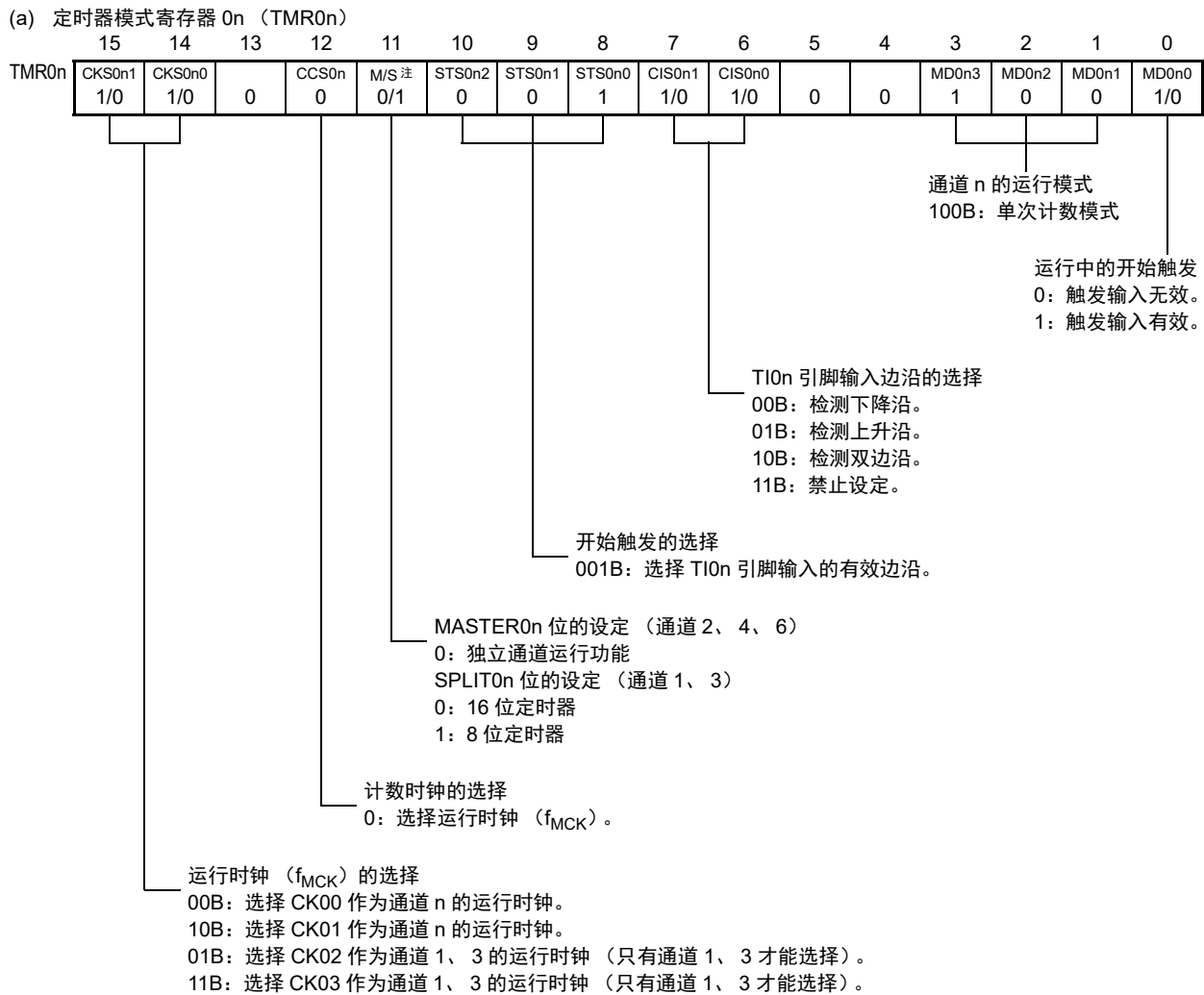
图 6-58 作为延迟计数器的运行基本时序例子



备注 1. n: 通道号 (n=0 ~ 7)

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 6-59 延迟计数器功能时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) 时置“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、TMR04、TMR06: MASTER0n 位
TMR01、TMR03: SPLIT0n 位
TMR00、TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0~7)

图 6-60 延迟计数器功能时的操作步骤

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置“1”。	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 (在 8 位定时器模式中为 CK02 和 CK03) 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置“0” (OFF) 或者“1” (ON)。 设定定时器模式寄存器 0n (TMR0n) (确定通道 n 的运行模式)。 给定时器数据寄存器 0n (TDR0n) 设定输出延迟时间。 设定噪声滤波器允许寄存器 1 (NFEN1)。 将 TOE0n 位置“0”并且停止 TO0n 的运行。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
开始 运行	将 TS0n 位置“1”。 因为 TS0n 位是触发位, 所以自动返回到“0”。	TE0n 位变为“1”并且进入开始触发 (检测 TI0n 引脚输入的有效边沿或者将 TS0n 位置“1”) 的检测等待状态。
	通过检测到下一个开始触发, 开始递减计数。 • 检测 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)。
运行中	能任意更改 TDR0n 寄存器的设定值。 能随时读 TCR0n 寄存器。 不使用 TSR0n 寄存器。	计数器 (TCR0n) 进行递减计数。如果 TCR0n 计数到“0000H”, 就产生 INTTM0n, 并且在检测到下一次开始触发 (检测 TI0n 引脚输入的有效边沿或者将 TS0n 位置“1”) 前 TCR0n 为“0000H”而停止计数。
停止 运行	将 TT0n 位置“1”。 因为 TT0n 位是触发位, 所以自动返回到“0”。	TE0n 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 对全部电路和各通道的 SFR 进行初始化。

重新
开始
运行

备注 n: 通道号 (n=0~7)

6.9 定时器阵列单元的多通道联动运行功能

6.9.1 作为单触发脉冲输出功能的运行

将 2 个通道成对使用，能通过 TI0n 引脚的输入生成任意延迟脉宽的单触发脉冲。
延迟和脉宽能用以下计算式进行计算：

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

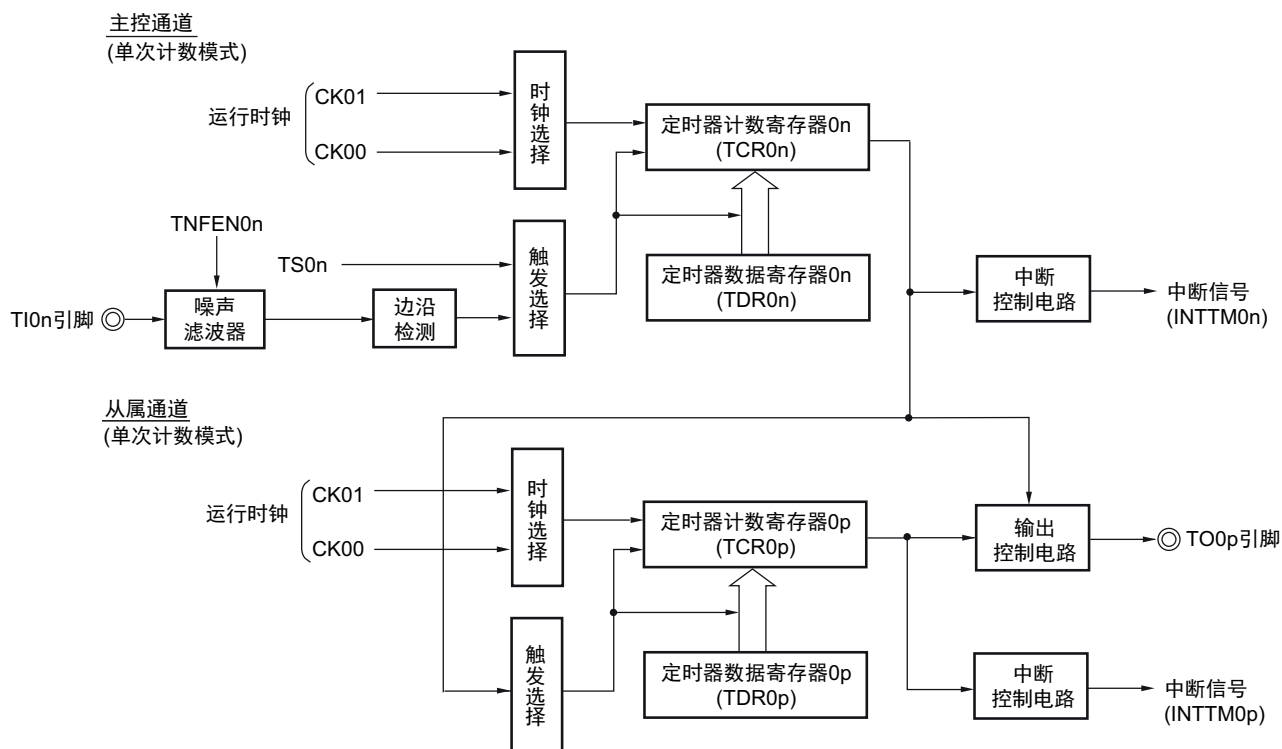
在单次计数模式中，主控通道运行并且对延迟进行计数。通过检测开始触发，主控通道的定时器计数寄存器 0n (TCR0n) 开始运行并且装入定时器数据寄存器 0n (TDR0n) 的值。TCR0n 寄存器通过计数时钟，从装入的 TDR0n 寄存器的值开始递减计数。如果 TCR0n 变为“0000H”，就输出 INTTM0n，并且在检测到下一个开始触发前停止计数。

在单次计数模式中，从属通道运行并且对脉宽进行计数。将主控通道的 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，从装入的 TDR0p 寄存器值开始递减计数。如果计数值变为“0000H”，就输出 INTTM0p，并且在检测到下一个开始触发（主控通道的 INTTM0n）前停止计数。在从主控通道产生 INTTM0n 并且经过 1 个计数时钟后，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: 通道号 (n=0、2、4、6)
p: 从属通道号 (n < p ≤ 7)

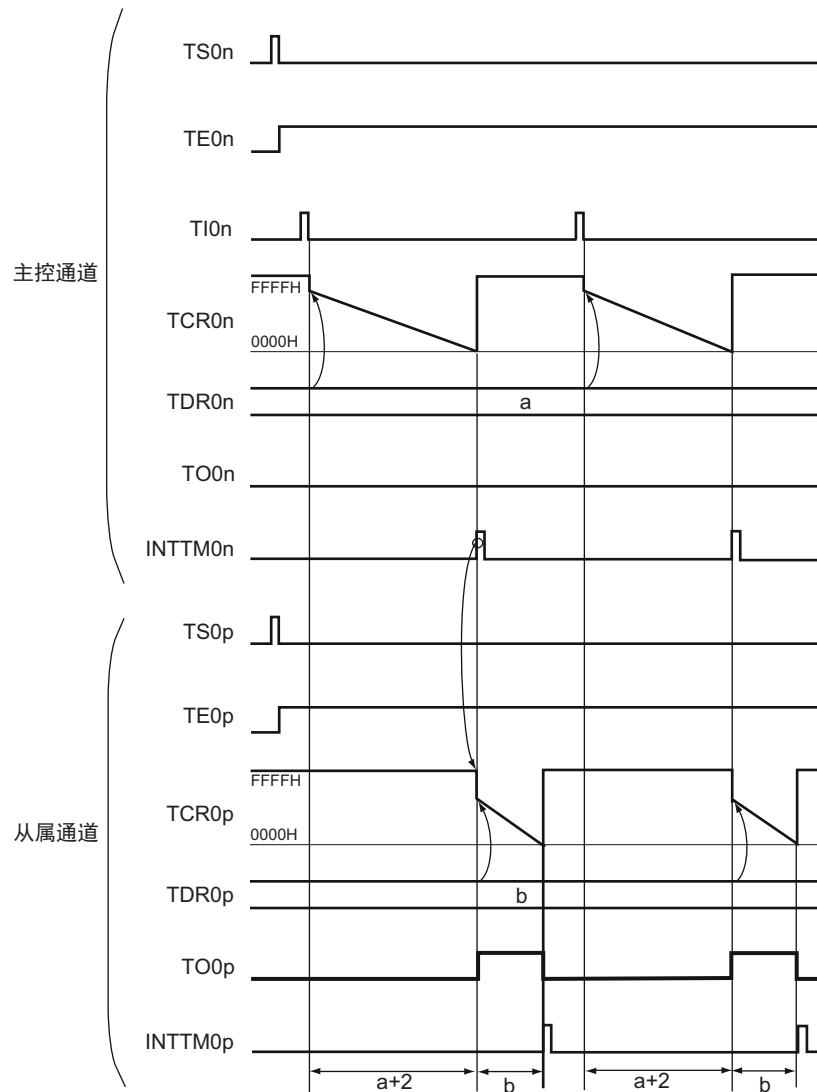
图 6-61 作为单触发脉冲输出功能运行的框图



备注 n: 通道号 (n=0、2、4、6)

p: 从属通道号 (n < p ≤ 7)

图 6-62 作为单触发脉冲输出功能的运行基本时序例子



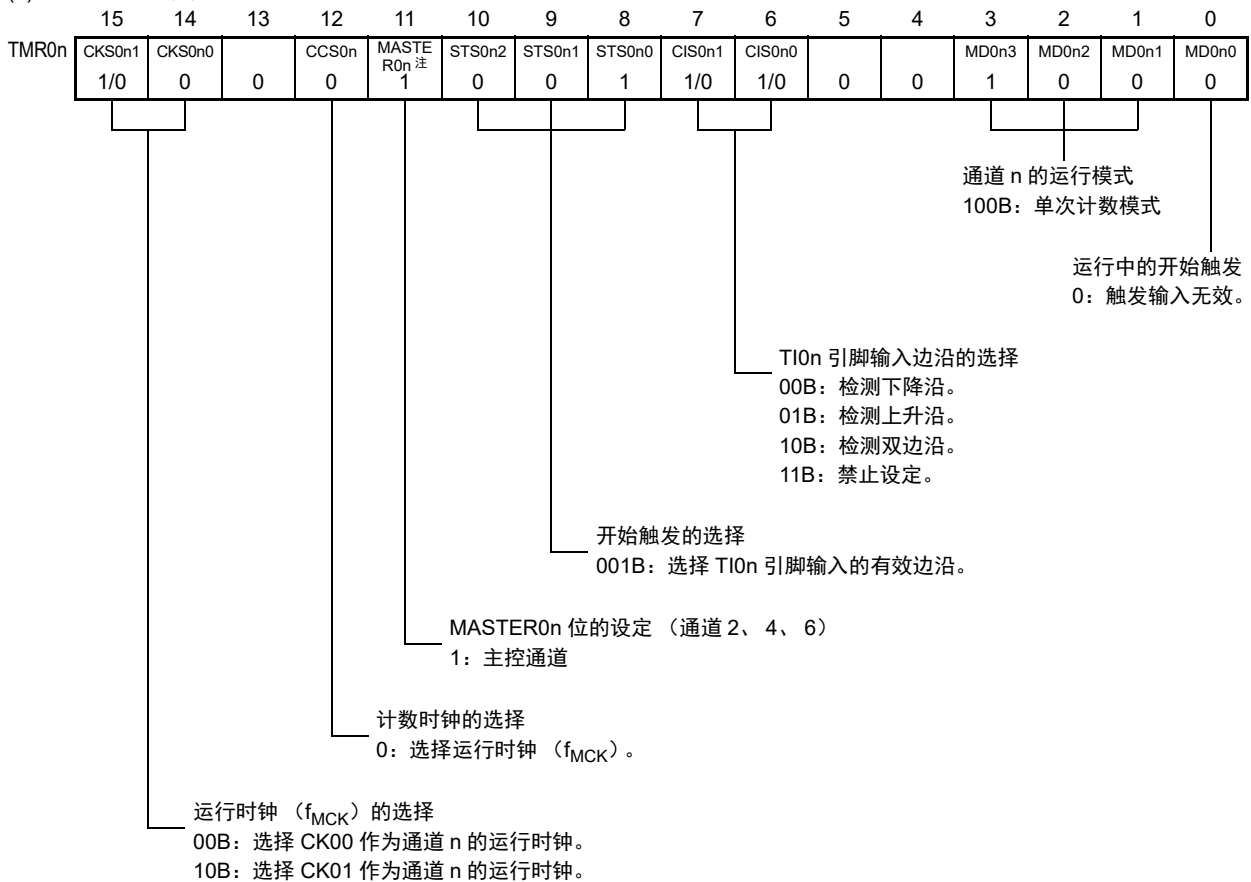
备注 1. n: 通道号 (n=0、2、4、6)

p: 从属通道号 ($n < p \leq 7$)

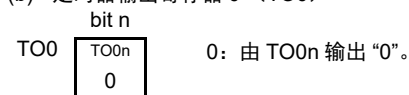
2. TS0n、TS0p: 定时器通道开始寄存器 0 (TS0) 的 bit n、p
- TE0n、TE0p: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p
- TI0n: TI0n 引脚的输入信号
- TCR0n、TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)
- TDR0n、TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)
- TO0n、TO0p: TO0n 引脚和 TO0p 引脚的输出信号

图 6-63 单触发脉冲输出功能时（主控通道）的寄存器设定内容例子

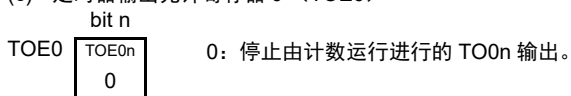
(a) 定时器模式寄存器 0n (TMR0n)



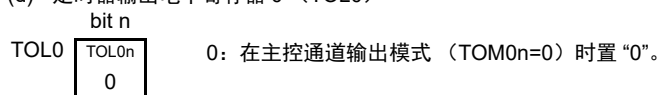
(b) 定时器输出寄存器 0 (TO0)



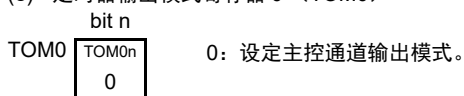
(c) 定时器输出允许寄存器 0 (TOE0)



(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)

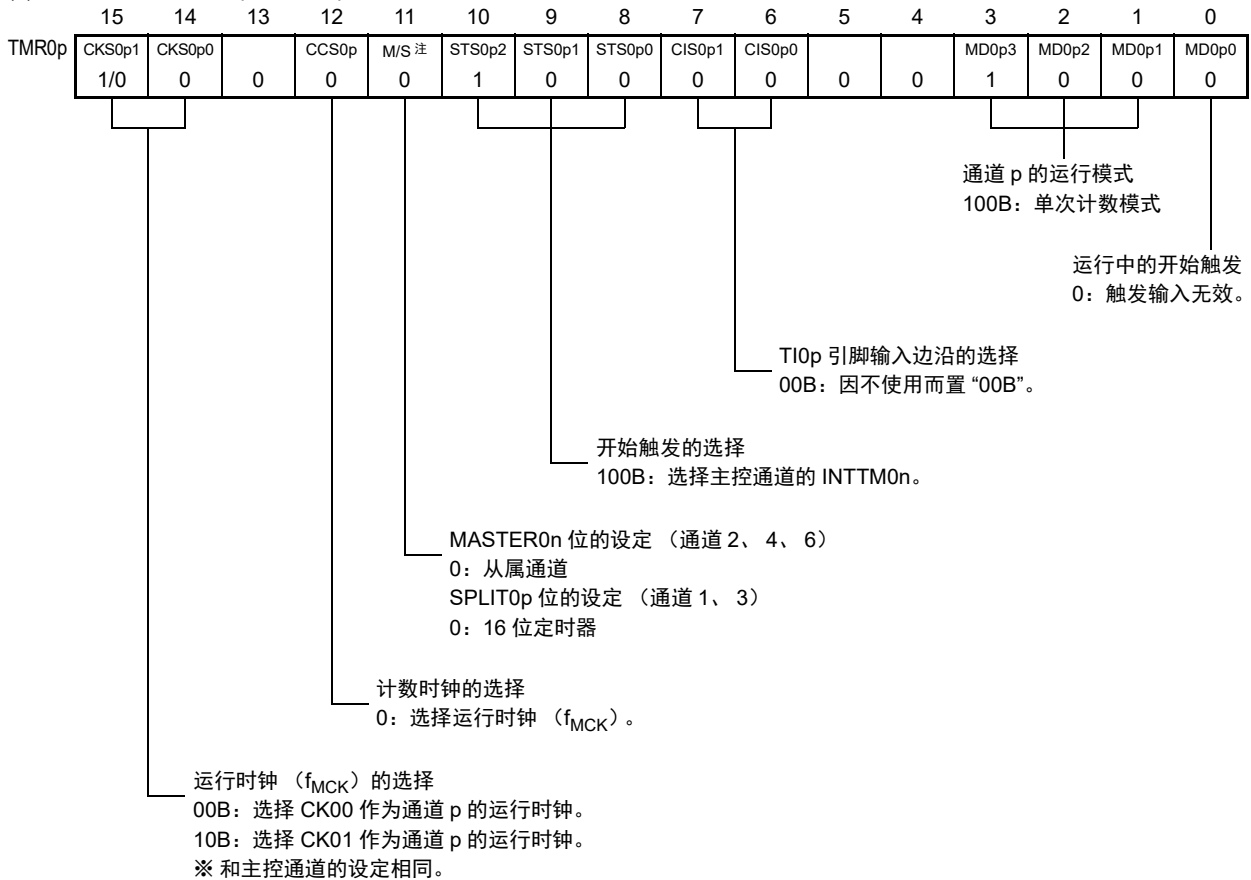


注 TMR02、TMR04、TMR06: MASTER0n=1
TMR00: 固定为“0”。

备注 n: 通道号 (n=0、2、4、6)

图 6-64 单触发脉冲输出功能时（从属通道）的寄存器设定内容例子

(a) 定时器模式寄存器 0p (TMR0p)



(b) 定时器输出寄存器 0 (TO0)

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

(c) 定时器输出允许寄存器 0 (TOE0)

	bit p	
TOE0	TOE0p	0: 停止由计数运行进行的 TO0p 输出。
	1/0	1: 允许由计数运行进行的 TO0p 输出。

(d) 定时器输出电平寄存器 0 (TOL0)

	bit p	
TOL0	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	
TOM0	TOM0p	1: 设定从属通道输出模式。
	1	

注 TMR02、TMR04、TMR06: MASTER0n 位
TMR01、TMR03: SPLIT0p 位
TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0、2、4、6)
p: 从属通道号 (n < p ≤ 7)

图 6-65 单触发脉冲输出功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 "1"。→	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 的时钟频率。	
通道初 始设定	将噪声滤波器允许寄存器 1 (NFEN1) 的对应位置 "0" (OFF) 或者 "1" (ON)。 设定使用的 2 个通道的定时器模式寄存器 0n、0p (TMR0n、TMR0p) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 0n (TDR0n) 设定 输出延迟时间, 并且给从属通道的 TDR0p 寄存器设 定脉宽。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的 TOM0p 位 置 "1" (从属通道输出模式)。 设定 TOL0p 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 输出。————→ 将端口寄存器和端口模式寄存器置 "0"。————→	TO0p 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 "0" 时, 输出 TO0p 初始设定的电平。 因为通道处于运行停止状态, 所以 TO0p 不变。 TO0p 引脚输出 TO0p 设定的电平。

备注 n: 通道号 (n=0、2、4、6)

p: 从属通道号 (n < p ≤ 7)

图 6-65 单触发脉冲输出功能时的操作步骤 (2/2)

	软件操作	硬件状态
重新 开始 运行	开始运行 将 TOE0p 位 (从属) 置“1” (只限于重新开始运行)。 将定时器通道开始寄存器 0 (TS0) 的 TS0n (主控) 和 TS0p (从属) 位同时置“1”。 因为 TS0n 位和 TS0p 位是触发位, 所以自动返回到“0”。	TE0n 位和 TE0p 位都变为“1”, 主控通道进入开始触发 (检测 TI0n 引脚输入的有效边沿或者将主控通道的 TS0n 位置“1”) 的检测等待状态。 计数器还处于停止状态。
	通过检测主控通道的开始触发, 开始主控通道的计数。 • 检测 TI0n 引脚输入的有效边沿。 • 通过软件将主控通道的 TS0n 位置“1”注。 注 不能将从属通道的 TS0n 位置“1”。	主控通道开始计数。
	运行中 只能更改 TMR0n 寄存器的 CIS0n1 位和 CIS0n0 位的设定值。 禁止更改 TMR0p、TDR0n、TDR0p 寄存器以及 TOM0n 位、TOM0p 位、TOL0n 位和 TOL0p 位的设定值。 能随时读 TCR0n 寄存器和 TCR0p 寄存器。 不使用 TSR0n 寄存器和 TSR0p 寄存器。 能更改从属通道的 TO0 寄存器和 TOE0 寄存器的设定值。	主控通道通过检测开始触发 (检测 TI0n 引脚输入的有效边沿或者将主控通道的 TS0n 位置“1”), 将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n), 并且进行递减计数。 如果 TCR0n 计数到“0000H”, 就产生 INTTM0n, 并且在下一次开始触发前停止计数。 从属通道以主控通道的 INTTM0n 为触发, 将 TDR0p 寄存器的值装入 TCR0p 寄存器并且计数器开始递减计数。在从主控通道输出 INTTM0n 并且经过 1 个计数时钟后, 将 TO0p 的输出电平置为有效电平。然后, 如果 TCR0p 计数到“0000H”, 就在将 TO0p 的输出电平置为无效电平后停止计数。 此后, 重复此运行。
	停止运行 将 TT0n 位 (主控) 和 TT0p 位 (从属) 同时置“1”。 因为 TT0n 位和 TT0p 位是触发位, 所以自动返回到“0”。 将从属通道的 TOE0p 位置“0”并且给 TO0p 位设定值。	TE0n 位和 TE0p 位都变为“0”并且停止计数。 TCR0n 寄存器和 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。 TO0p 引脚输出 TO0p 设定的电平。
TAU 停止	要保持 TO0p 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 TO0p 位置“0”。 不需要保持 TO0p 引脚输出电平的情况: 不需要设定。 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 对全部电路和各通道的 SFR 进行初始化。 (TO0p 位变为“0”并且 TO0p 引脚变为端口功能)

备注 n: 通道号 (n=0、2、4、6)

p: 从属通道号 (n < p ≤ 7)

6.9.2 作为 PWM 功能的运行

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

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDR0n (主控) 的设定值} + 1 \} \times \text{计数时钟周期} \\ \text{占空比 [\%]} &= \{ \text{TDR0p (从属) 的设定值} \} / \{ \text{TDR0n (主控) 的设定值} + 1 \} \times 100 \\ \text{0\% 输出:} & \quad \text{TDR0p (从属) 的设定值} = 0000\text{H} \\ \text{100\% 输出:} & \quad \text{TDR0p (从属) 的设定值} \geq \{ \text{TDR0n (主控) 的设定值} + 1 \} \end{aligned}$$

备注 当 TDR0p (从属) 的设定值 > {TDR0n (主控) 的设定值 + 1} 时，占空比超过 100%，但是为 100% 输出。

主控通道用作间隔定时器模式。如果将定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS0n) 置“1”，就输出中断 (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)，并且通过计数时钟进行递减计数。当计数到“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 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 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“0000H”为止。当计数到“0000H”时，输出 INTTM0p，并且等待下一个开始触发 (主控通道的 INTTM0n)。

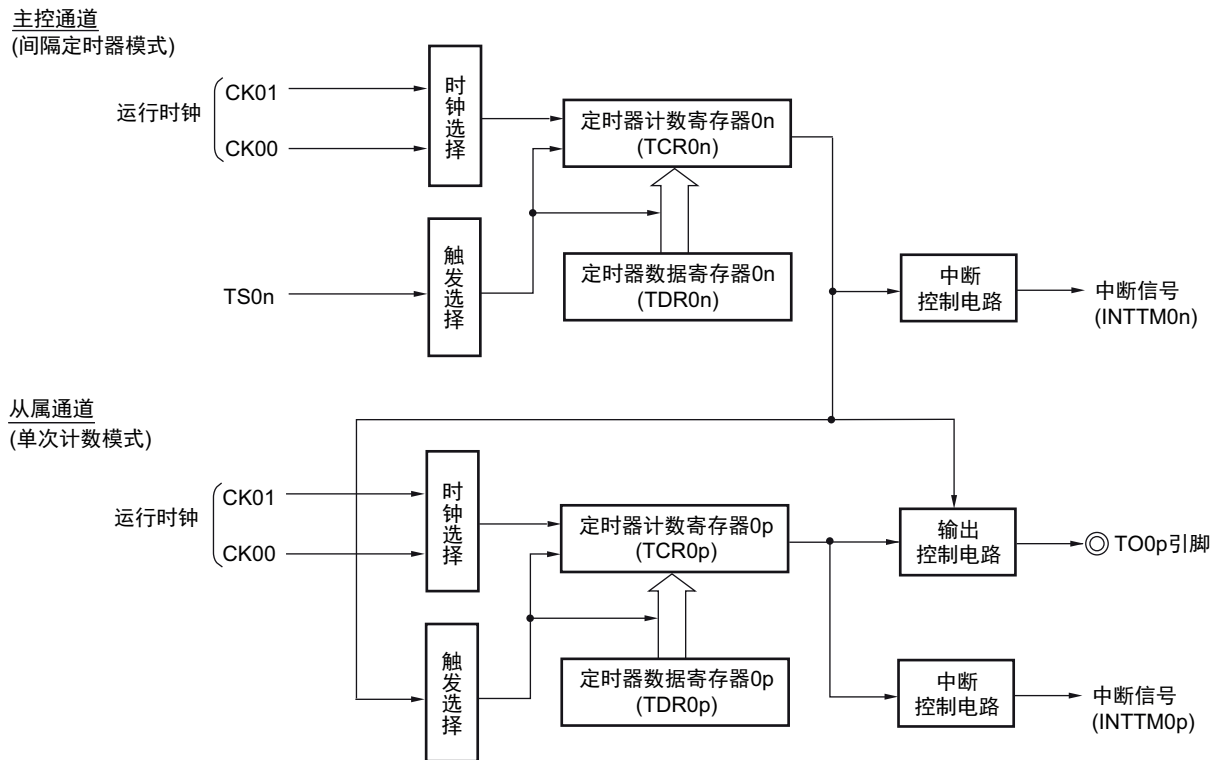
当用作 PWM 功能时，从属通道进行递减计数，在计数到“0000H”为止的期间为 PWM 输出 (TO0p) 的占空比。

在从主控通道产生 INTTM0n 并且经过 1 个时钟后，PWM 输出 (TO0p) 变为有效电平，并且在从属通道的 TCR0p 寄存器的值为“0000H”时变为无效电平。

注意 要同时改写主控通道的定时器数据寄存器 0n (TDR0n) 和从属通道的 TDR0p 寄存器时，需要 2 次写存取。因为在主控通道产生 INTTM0n 时将 TDR0n 寄存器和 TDR0p 寄存器的值装入 TCR0n 寄存器和 TCR0p 寄存器，所以如果分别在主控通道产生 INTTM0n 前后进行改写，TO0p 引脚就不能输出期待的波形。因此，要同时改写主控的 TDR0n 寄存器和从属的 TDR0p 寄存器时，必须在主控通道产生 INTTM0n 后立即改写这 2 个寄存器。

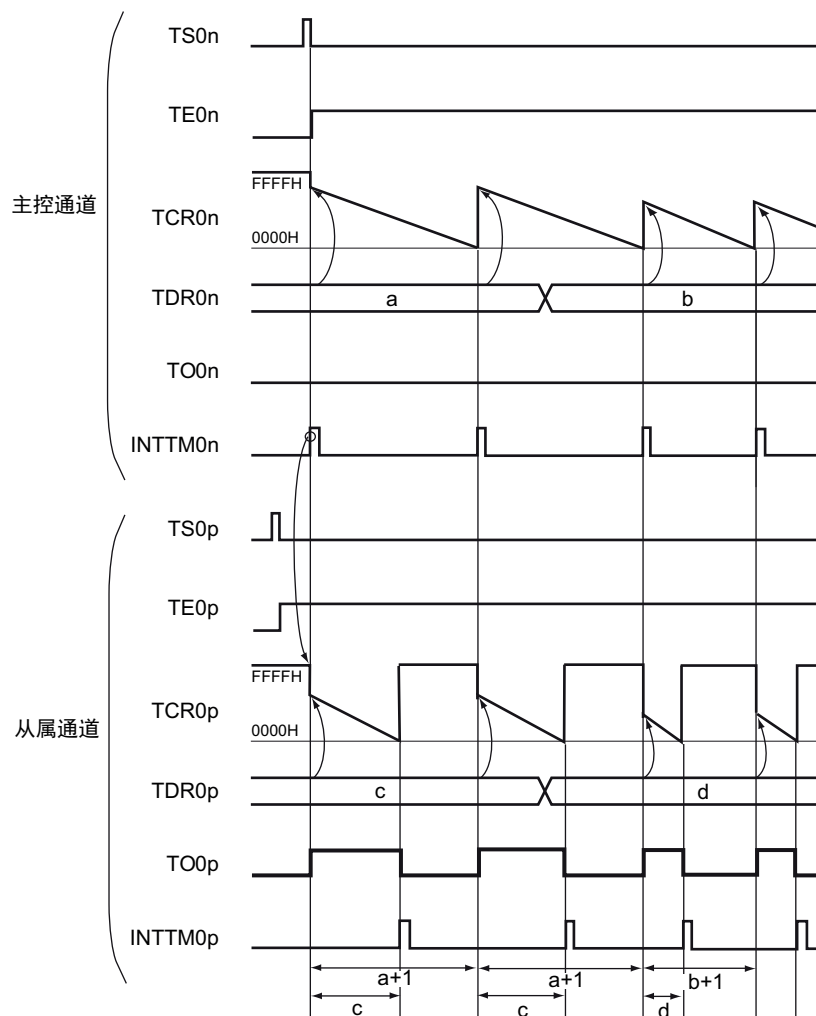
备注 n: 通道号 (n=0、2、4、6)
p: 从属通道号 (n < p ≤ 7)

图 6-66 作为 PWM 功能运行的框图



备注 n: 通道号 (n=0、2、4、6)
p: 从属通道号 (n < p ≤ 7)

图 6-67 作为 PWM 功能的运行基本时序例子



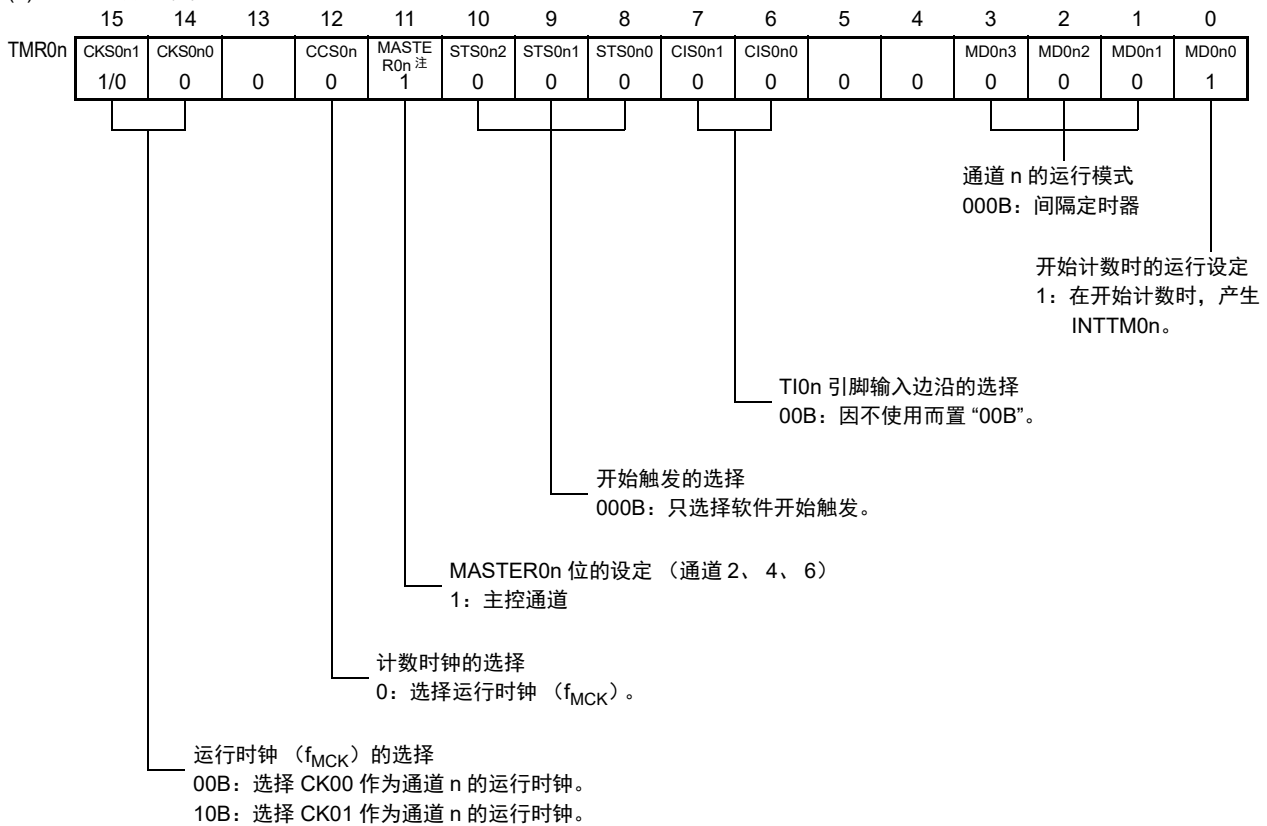
备注 1. n: 通道号 (n=0、2、4、6)

p: 从属通道号 ($n < p \leq 7$)

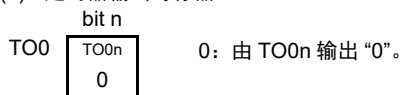
2. TS0n、TS0p: 定时器通道开始寄存器 0 (TS0) 的 bit n、p
- TE0n、TE0p: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p
- TCR0n、TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)
- TDR0n、TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)
- TO0n、TO0p: TO0n 引脚和 TO0p 引脚的输出信号

图 6-68 PWM 功能时（主控通道）的寄存器设定内容例子

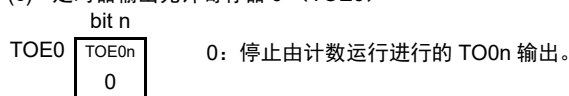
(a) 定时器模式寄存器 0n (TMR0n)



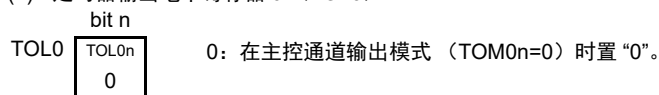
(b) 定时器输出寄存器 0 (TO0)



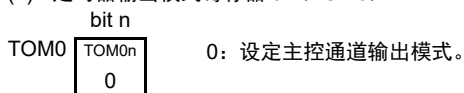
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



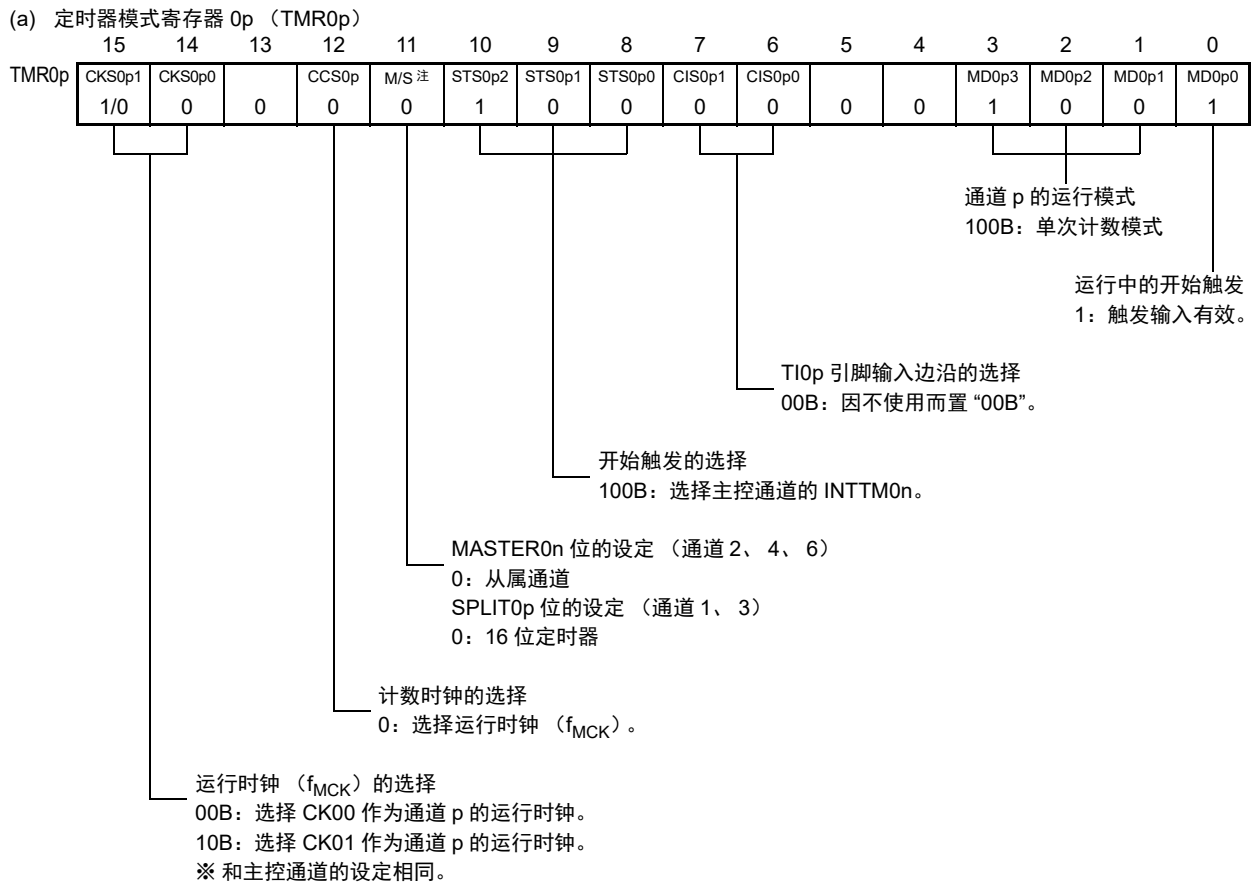
(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR02、TMR04、TMR06: MASTER0n=1
TMR00: 固定为“0”。

备注 n: 通道号 (n=0、2、4、6)

图 6-69 PWM 功能时（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	
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。
1: 允许由计数运行进行的 TO0p 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。

注 TMR02、TMR04、TMR06: MASTER0n 位
TMR01、TMR03: SPLIT0p 位
TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0、2、4、6)
p: 从属通道号 (n < p ≤ 7)

图 6-70 PWM 功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 "1"。→	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 的时钟频率。	
通道初 始设定	设定使用的 2 个通道的定时器模式寄存器 0n、0p (TMR0n、TMR0p) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 0n (TDR0n) 设定间隔 (周期) 值, 并且给从属通道的 TDR0p 寄存器设定占空比的值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的 TOM0p 位置 "1" (从属通道输出模式)。 设定 TOL0p 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 输出。——→ 将端口寄存器和端口模式寄存器置 "0"。——→	TO0p 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 "0" 时, 输出 TO0p 初始设定的电平。 因为通道处于运行停止状态, 所以 TO0p 不变。 TO0p 引脚输出 TO0p 设定的电平。

备注 n: 通道号 (n=0、2、4、6)

p: 从属通道号 (n < p ≤ 7)

图 6-70 PWM 功能时的操作步骤 (2/2)

	软件操作	硬件状态
重新开始运行	开始运行 将 TOE0p 位 (从属) 置“1” (只限于重新开始运行)。 将定时器通道开始寄存器 0 (TS0) 的 TS0n 位 (主控) 和 TS0p 位 (从属) 同时置“1”。 因为 TS0n 位和 TS0p 位是触发位, 所以自动返回到“0”。	TE0n 位和 TE0p 位都变为“1”。 主控通道开始计数并且产生 INTTM0n。以此为触发, 从属通道也开始计数。
	运行中 禁止更改 TMR0n 寄存器和 TMR0p 寄存器以及 TOM0n 位、TOM0p 位、TOL0n 位和 TOL0p 位的设定值。 能在主控通道产生 INTTM0n 后更改 TDR0n 寄存器和 TDR0p 寄存器的设定值。 能随时读 TCR0n 寄存器和 TCR0p 寄存器。 不使用 TSR0n 寄存器和 TSR0p 寄存器。	主控通道将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n), 并且进行递减计数。如果 TCR0n 计数到“0000H”, 就产生 INTTM0n。同时, 将 TDR0n 寄存器的值装入 TCR0n 寄存器, 并且重新开始递减计数。 从属通道以主控通道的 INTTM0n 为触发, 将 TDR0p 寄存器的值装入 TCR0p 寄存器, 并且计数器进行递减计数。在从主控通道输出 INTTM0n 并且经过 1 个计数时钟后, 将 TO0p 的输出电平置为有效电平。然后, 如果 TCR0p 计数到“0000H”, 就在将 TO0p 的输出电平置为无效电平后停止计数。 此后, 重复此运行。
	停止运行 将 TT0n 位 (主控) 和 TT0p 位 (从属) 同时置“1”。 因为 TT0n 位和 TT0p 位是触发位, 所以自动返回到“0”。	TE0n 位和 TE0p 位都变为“0”并且停止计数。 TCR0n 寄存器和 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。	TO0p 引脚输出 TO0p 设定的电平。
	TAU 停止 要保持 TO0p 引脚输出电平的情况: 在给端口寄存器设定要保持的值后将 TO0p 位置“0”。 不需要保持 TO0p 引脚输出电平的情况: 不需要设定。	通过端口功能保持 TO0p 引脚的输出电平。
	将 PER0 寄存器的 TAU0EN 位置“0”。	断电状态 对全部电路和各通道的 SFR 进行初始化。 (TO0p 位变为“0”并且 TO0p 引脚变为端口功能)

备注 n: 通道号 (n=0、2、4、6)

p: 从属通道号 (n < p ≤ 7)

6.9.3 作为多重 PWM 输出功能的运行

这是通过扩展 PWM 功能并且使用多个从属通道进行不同占空比的多个 PWM 输出的功能。
例如，当 2 个从属通道成对使用时，输出脉冲的周期和占空比能用以下计算式进行计算：

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDR0n (主控) 的设定值} + 1 \} \times \text{计数时钟周期} \\ \text{占空比 1 [\%]} &= \{ \text{TDR0p (从属 1) 的设定值} \} / \{ \text{TDR0n (主控) 的设定值} + 1 \} \times 100 \\ \text{占空比 2 [\%]} &= \{ \text{TDR0q (从属 2) 的设定值} \} / \{ \text{TDR0n (主控) 的设定值} + 1 \} \times 100 \end{aligned}$$

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，占空比超过 100%，但是为 100% 输出。

在间隔定时器模式中，主控通道的定时器计数寄存器 0n (TCR0n) 运行并且对周期进行计数。

在单次计数模式中，从属通道 1 的 TCR0p 寄存器运行并且对占空比进行计数以及从 TO0p 引脚输出 PWM 波形。以主控通道的 INTTM0n 为开始触发，将定时器数据寄存器 0p (TDR0p) 的值装入 TCR0p 寄存器并且进行递减计数。如果 TCR0p 变为“0000H”，就输出 INTTM0p，并且在输入下一个开始触发 (主控通道的 INTTM0n) 前停止计数。在从主控通道产生 INTTM0n 并且经过 1 个计数时钟后，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。

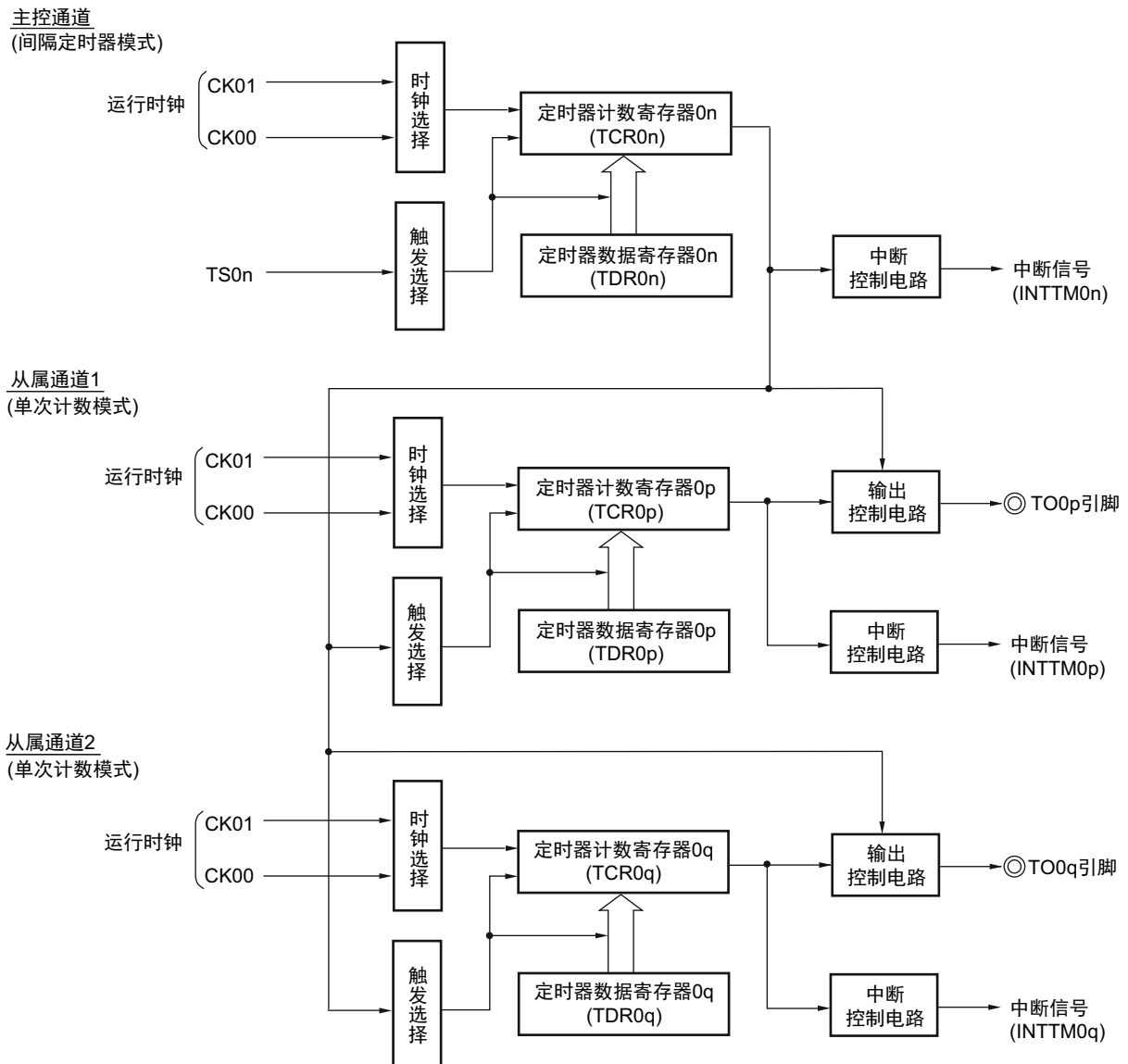
和从属通道 1 的 TCR0p 寄存器相同，在单次计数模式中，从属通道 2 的 TCR0q 寄存器运行并且对占空比进行计数以及从 TO0q 引脚输出 PWM 波形。以主控通道的 INTTM0n 为开始触发，将 TDR0q 寄存器的值装入 TCR0q 寄存器并且进行递减计数。如果 TCR0q 变为“0000H”，就输出 INTTM0q，并且在输入下一个开始触发 (主控通道的 INTTM0n) 前停止计数。在从主控通道产生 INTTM0n 并且经过 1 个计数时钟后，TO0q 的输出电平变为有效电平，如果 TCR0q 变为“0000H”，就变为无效电平。

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

注意 要同时改写主控通道的定时器数据寄存器 0n (TDR0n) 和从属通道 1 的 TDR0p 寄存器时，至少需要 2 次写存取。因为在主控通道产生 INTTM0n 时将 TDR0n 寄存器和 TDR0p 寄存器的值装入 TCR0n 寄存器和 TCR0p 寄存器，所以如果分别在主控通道产生 INTTM0n 前后进行改写，TO0p 引脚就不能输出期待的波形。因此，要同时改写主控的 TDR0n 寄存器和从属的 TDR0p 寄存器时，必须在主控通道产生 INTTM0n 后立即改写这 2 个寄存器 (同样也适用于从属通道 2 的 TDR0q 寄存器)。

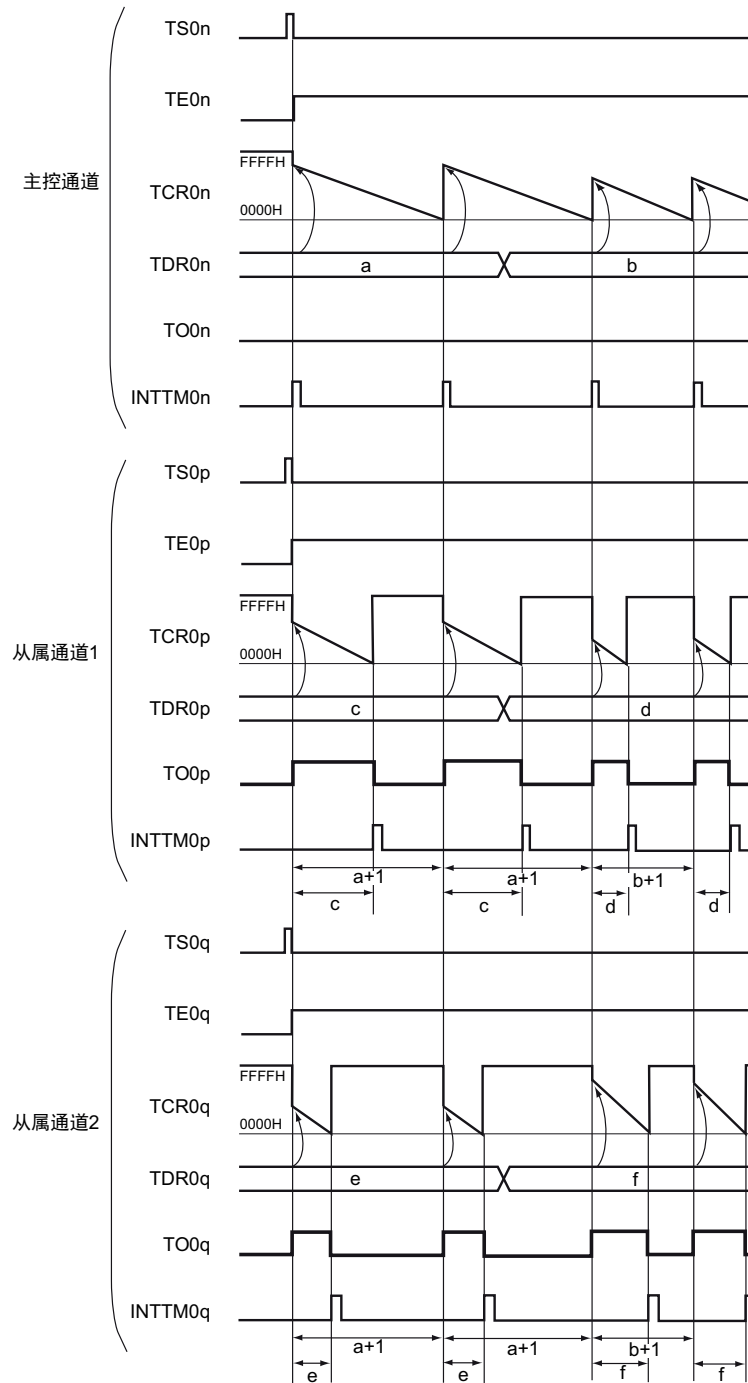
备注 n: 主控通道号 (n=0、2、4)
p: 从属通道号 1 q: 从属通道号 2
n < p < q ≤ 7 (p、q 是大于 n 的整数)

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



备注 n: 通道号 (n=0、2、4)
 p: 从属通道号1 q: 从属通道号2
 $n < p < q \leq 7$ (p、q是大于n的整数)

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



备注 1. n: 主控通道号 (n=0、2、4)

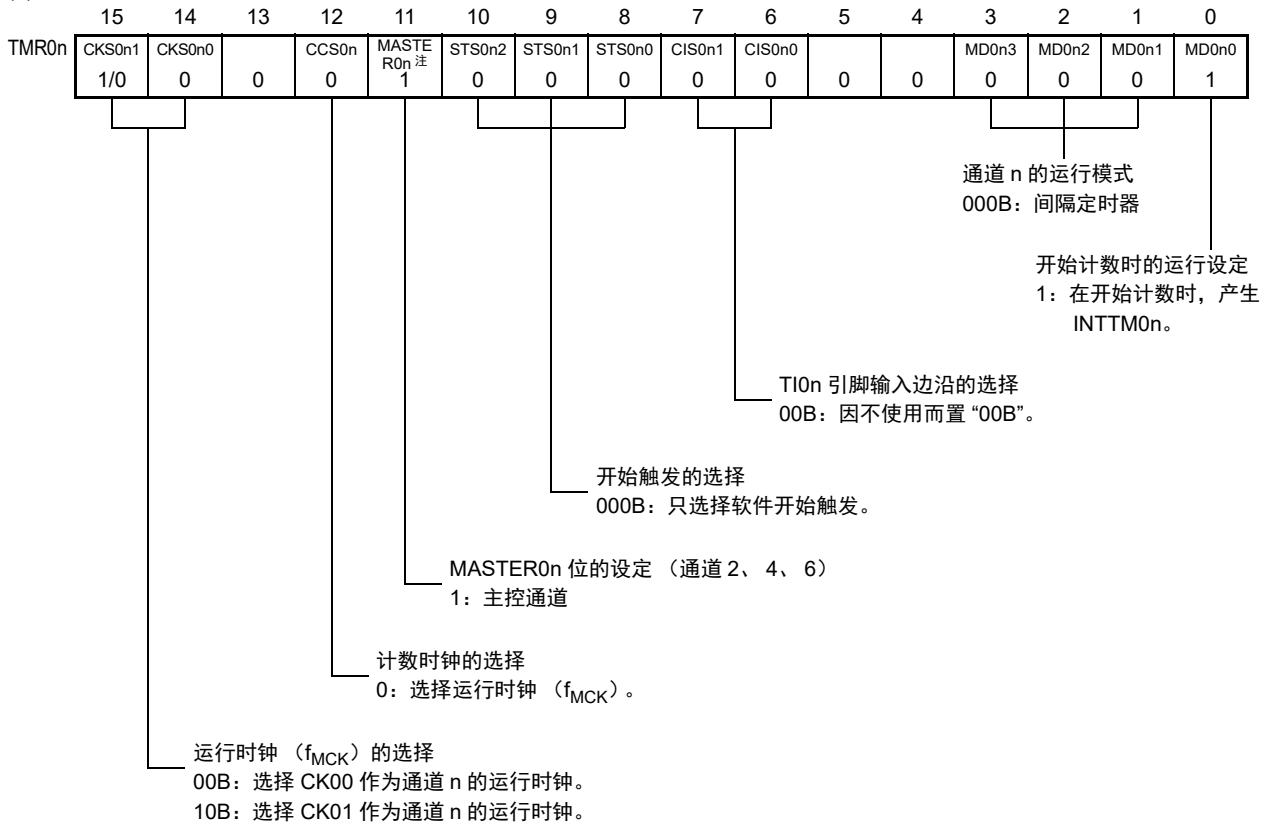
p: 从属通道号 1 q: 从属通道号 2

$n < p < q \leq 7$ (p、q 是大于 n 的整数)

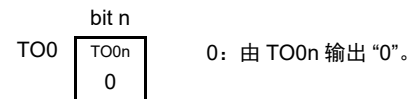
2. TS0n、TS0p、TS0q: 定时器通道开始寄存器 0 (TS0) 的 bit n、p、q
- TE0n、TE0p、TE0q: 定时器通道允许状态寄存器 0 (TE0) 的 bit n、p、q
- TCR0n、TCR0p、TCR0q: 定时器计数寄存器 0n、0p、0q (TCR0n、TCR0p、TCR0q)
- TDR0n、TDR0p、TDR0q: 定时器数据寄存器 0n、0p、0q (TDR0n、TDR0p、TDR0q)
- TO0n、TO0p、TO0q: TO0n、TO0p、TO0q 引脚的输出信号

图 6-73 多重 PWM 输出功能时（主控通道）的寄存器设定内容例子

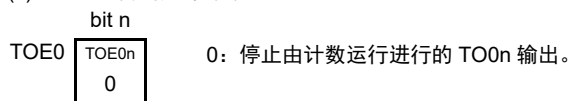
(a) 定时器模式寄存器 0n (TMR0n)



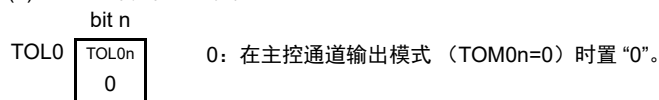
(b) 定时器输出寄存器 0 (TO0)



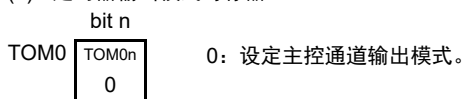
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



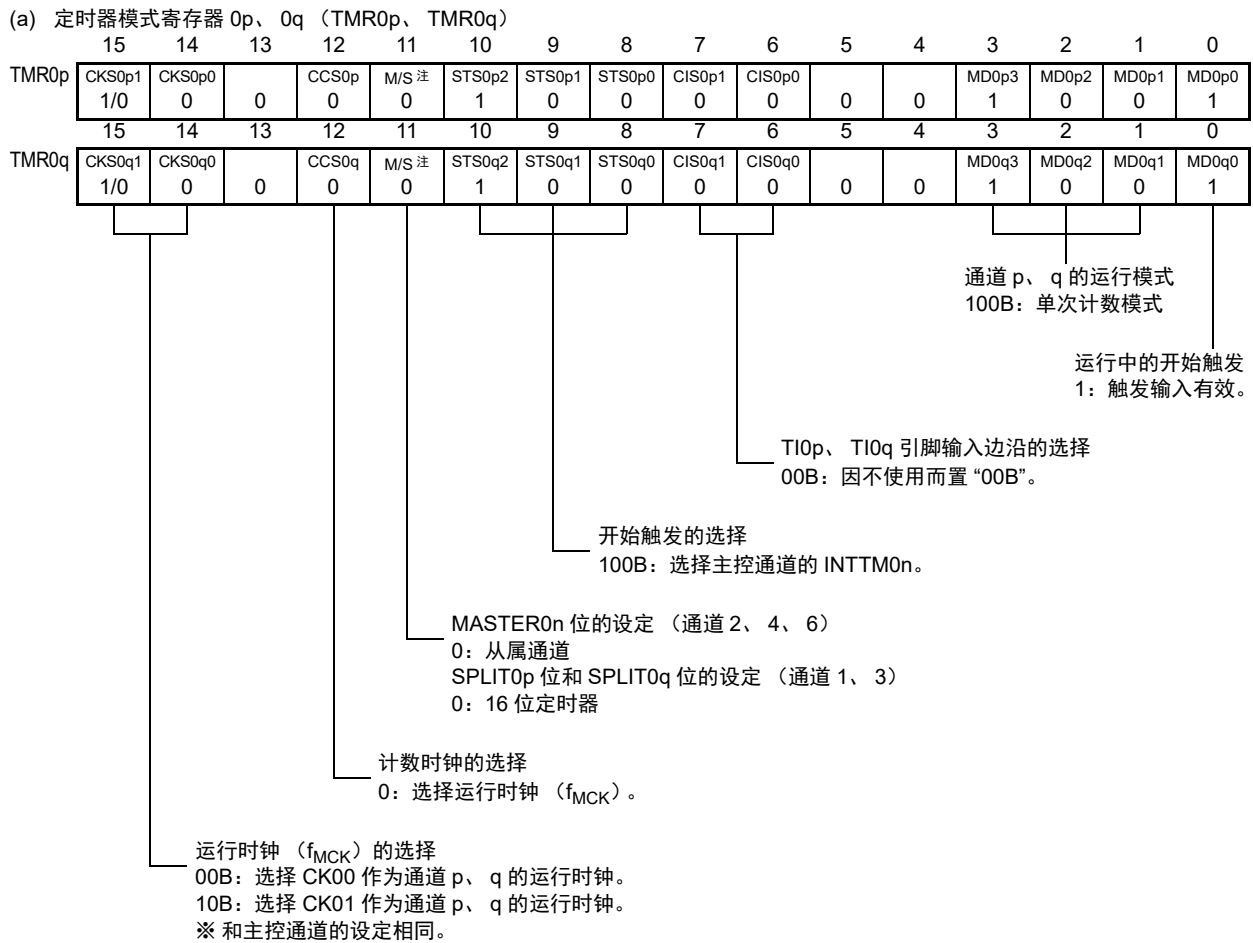
(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR02、TMR04、TMR06: MASTER0n=1
TMR00: 固定为“0”。

备注 n: 通道号 (n=0、2、4)

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



(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。
	1/0	1/0	1: 允许由计数运行进行的 TO0p 和 TO0q 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、TMR04、TMR06: MASTER0n 位
TMR01、TMR03: SPLIT0p 位、SPLIT0q 位
TMR05、TMR07: 固定为“0”。

备注 n: 通道号 (n=0、2、4)
p: 从属通道号 1 q: 从属通道号 2
n < p < q ≤ 7 (p、q 是大于 n 的整数)

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

	软件操作	硬件状态
TAU 初始 设定		断电状态 (停止提供时钟, 不能写各寄存器)
	将外围允许寄存器 0 (PER0) 的 TAU0EN 位置 "1"。→	通电状态, 各通道处于运行停止状态。 (开始提供时钟, 能写各寄存器)
	设定定时器时钟选择寄存器 0 (TPS0)。 确定 CK00 和 CK01 的时钟频率。	
通道初 始设定	设定使用的各通道的定时器模式寄存器 0n、0p、0q (TMR0n、TMR0p、TMR0q) (确定通道的运行模式)。 给主控通道的定时器数据寄存器 0n (TDR0n) 设定间隔 (周期) 值, 并且给从属通道的 TDR0p 寄存器和 TDR0q 寄存器设定占空比的值。	通道处于运行停止状态。 (提供时钟, 消耗一部分电力)
	从属通道的设定 将定时器输出模式寄存器 0 (TOM0) 的 TOM0p 位和 TOM0q 位置 "1" (从属通道输出模式)。 设定 TOL0p 位和 TOL0q 位。 设定 TO0p 位和 TO0q 位, 并且确定 TO0p 和 TO0q 输出的初始电平。 将 TOE0p 位和 TOE0q 位置 "1", 允许 TO0p 和 TO0q 的输出。 将端口寄存器和端口模式寄存器置 "0"。	TO0p 引脚和 TO0q 引脚处于 Hi-Z 输出状态。 当端口模式寄存器为输出模式并且端口寄存器为 "0" 时, 输出 TO0p 和 TO0q 初始设定的电平。 因为通道处于运行停止状态, 所以 TO0p 和 TO0q 不变。 TO0p 引脚和 TO0q 引脚输出 TO0p 和 TO0q 设定的电平。

备注 n: 主控通道号 (n=0、2、4)
p: 从属通道号1 q: 从属通道号2
 $n < p < q \leq 7$ (p、q 是大于 n 的整数)

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

	软件操作	硬件状态
重新开始运行	开始运行 (只在重新开始运行时将 TOE0p 位和 TOE0q 位 (从属) 置“1”) 将定时器通道开始寄存器 0 (TS0) 的 TS0n 位 (主控)、TS0p 位和 TS0q 位 (从属) 同时置“1”。因为 TS0n 位、TS0p 位和 TS0q 位是触发位, 所以自动返回到“0”。	TE0n 位、TE0p 位和 TE0q 位都变为“1”。 主控通道开始计数并且产生 INTTM0n。以此为触发, 从属通道也开始计数。
	运行中 禁止更改 TMR0n、TMR0p、TMR0q 寄存器以及 TOM0n 位、TOM0p 位、TOM0q 位、TOL0n 位、TOL0p、TOL0q 位的设定值。 能在主控通道产生 INTTM0n 后更改 TDR0n、TDR0p、TDR0q 寄存器的设定值。 能随时读 TCR0n、TCR0p、TCR0q 寄存器。 不使用 TSR0n、TSR0p、TSR0q 寄存器。	主控通道将 TDR0n 寄存器的值装入定时器计数寄存器 0n (TCR0n), 并且进行递减计数。如果 TCR0n 计数到“0000H”, 就产生 INTTM0n。同时, 将 TDR0n 寄存器的值装入 TCR0n 寄存器, 并且重新开始递减计数。 从属通道 1 以主控通道的 INTTM0n 信号为触发, 将 TDR0p 寄存器的值传送到 TCR0p 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTM0n 并且经过 1 个计数时钟后, 将 TO0p 的输出电平置为有效电平。然后, 如果计数到“0000H”, 就在将 TO0p 的输出电平置为无效电平后停止计数。 从属通道 2 以主控通道的 INTTM0n 信号为触发, 将 TDR0q 寄存器的值传送到 TCR0q 寄存器, 并且计数器开始递减计数。在从主控通道输出 INTTM0n 并且经过 1 个计数时钟后, 将 TO0q 的输出电平置为有效电平。然后, 如果计数到“0000H”, 就在将 TO0q 的输出电平置为无效电平后停止计数。 此后, 重复此运行。
	停止运行 将 TT0n 位 (主控)、TT0p 位和 TT0q 位 (从属) 位同时置“1”。因为 TT0n 位、TT0p 位和 TT0q 位是触发位, 所以自动返回到“0”。	TE0n 位、TE0p 位和 TE0q 位都变为“0”并且停止计数。 TCR0n、TCR0p、TCR0q 寄存器保持计数值而停止计数。 TO0p 和 TO0q 输出不被初始化而保持状态。
	将从属通道的 TOE0p 位和 TOE0q 位置“0”并且给 TO0p 位和 TO0q 位设定值。	TO0p 引脚和 TO0q 引脚输出 TO0p 和 TO0q 设定的电平。
	TAU 停止 要保持 TO0p 引脚和 TO0q 引脚的输出电平的情况: 在给端口寄存器设定要保持的值后将 TO0p 位和 TO0q 位置“0”。 不需要保持 TO0p 引脚和 TO0q 引脚的输出电平的情况: 不需要设定。 将 PER0 寄存器的 TAU0EN 位置“0”。	通过端口功能保持 TO0p 引脚和 TO0q 引脚的输出电平。 断电状态 对全部电路和各通道的 SFR 进行初始化。 (TO0p 位和 TO0q 位变为“0”并且 TO0p 引脚和 TO0q 引脚变为端口功能)

备注 n: 通道号 (n=0、2、4)
p: 从属通道号1 q: 从属通道号2
 $n < p < q \leq 7$ (p、q 是大于 n 的整数)

6.10 使用定时器阵列单元时的注意事项

6.10.1 使用定时器输出时的注意事项

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

7.1 12 位间隔定时器的功能

以事先设定的任意时间间隔产生中断（INTIT），能用于从 STOP 模式的唤醒以及 A/D 转换器的 SNOOZE 模式的触发。

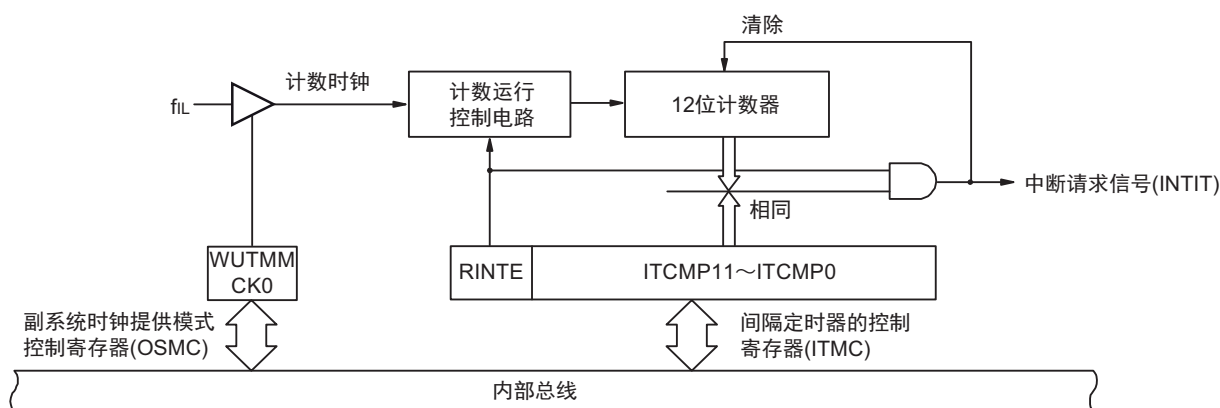
7.2 12 位间隔定时器的结构

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）
	副系统时钟提供模式控制寄存器（OSMC）
	12 位间隔定时器的控制寄存器（ITMC）

图 7-1 12 位间隔定时器的框图



7.3 控制 12 位间隔定时器的寄存器

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)
- 副系统时钟提供模式控制寄存器 (OSMC)
- 间隔定时器的控制寄存器 (ITMC)

7.3.1 外围允许寄存器 0 (PER0)

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

要使用 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	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TMKAEN	12 位间隔定时器的输入时钟的控制
0	停止提供输入时钟。 <ul style="list-style-type: none"> • 不能写 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。

注意 1. 如果要使用 12 位间隔定时器，就必须先在提供低速内部振荡器时钟 (WUTMMCK0=1) 后将 TMKAEN 位置“1”，然后设定间隔定时器的控制寄存器 (ITMC)。当 TMKAEN 位为“0”时，忽视 12 位间隔定时器的控制寄存器的写操作，而且读取值为初始值 (副系统时钟提供模式控制寄存器 (OSMC) 除外)。

2. 必须将 bit6 和 bit1 置“0”。

7.3.2 副系统时钟提供模式控制寄存器 (OSMC)

OSMC 寄存器控制 12 位间隔定时器运行时钟的提供。

要使用 12 位间隔定时器时，必须事先将 WUTMMCK0 位置“1”，并且不能在定时器停止前将 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 副系统时钟提供模式控制寄存器 (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	12 位间隔定时器运行时钟的提供
0	停止提供时钟。
1	提供低速内部振荡器时钟 (f_{IL})。

7.3.3 间隔定时器的控制寄存器 (ITMC)

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。

通过 16 位存储器操作指令设定 ITMC 寄存器。

在产生复位信号后，此寄存器的值变为“0FFFH”。

图 7-4 间隔定时器控制寄存器 (ITMC) 的格式

地址: FFF90H 复位后: 0FFFH R/W

符号 15 14 13 12 11 ~ 0

ITMC	RINTE	0	0	0	ITCMP11 ~ ITCMP0
------	-------	---	---	---	------------------

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

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	
<ul style="list-style-type: none"> 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. 要将 RINTE 位从“1”改为“0”时，必须在通过中断屏蔽标志寄存器将 INTIT 设定为禁止中断处理后进行改写。要重新开始运行 (从“0”改为“1”) 时，必须在清除 ITIF 标志后设定为允许中断处理。

2. RINTE 位的读取值在设定 RINTE 位后的 1 个计数时钟之后被反映。

3. 在从待机模式返回后，如果要设定 RINTE 位并且再次转移到待机模式，就必须在确认 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 位的设定时，必须在 RINTE 位为“0”的状态下进行。

但是，能在将 RINTE 位从“0”改为“1”或者从“1”改为“0”的同时更改 ITCMP11 ~ ITCMP0 位的设定。

7.4 12 位间隔定时器的运行

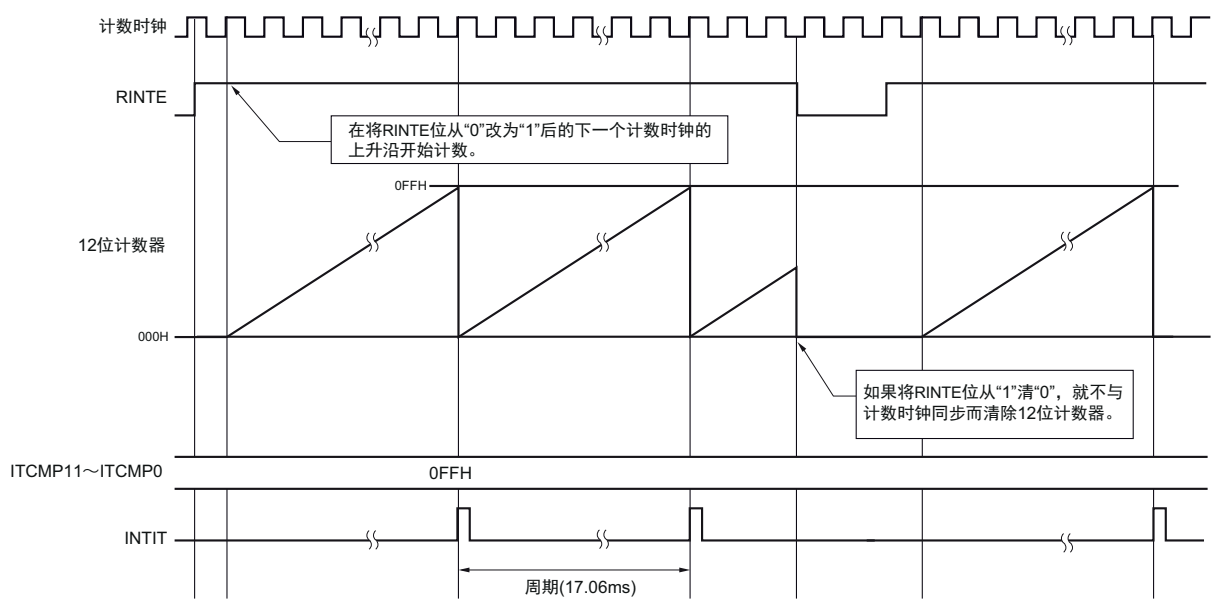
7.4.1 12 位间隔定时器的运行时序

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 位计数值和 ITCMP11 ~ ITCMP0 位的设定值相同时，将 12 位计数值清“0”并且继续计数，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: $f_{\text{IL}}=15\text{kHz}$)

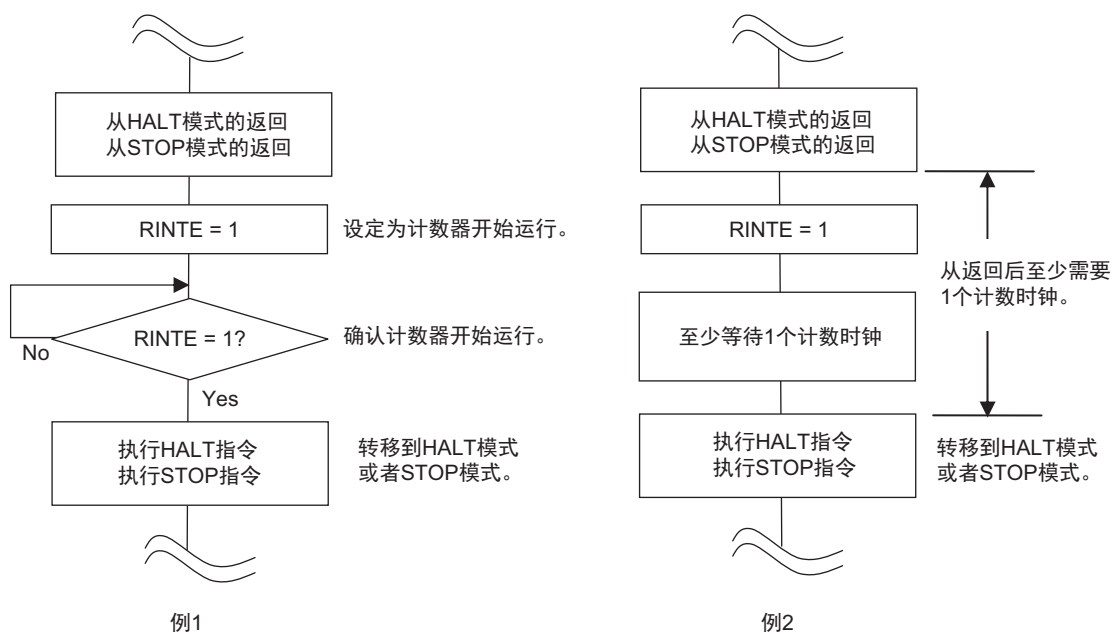


7.4.2 从HALT/STOP模式返回后开始计数器的运行并且再次向HALT/STOP模式的转移

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

- 在将 RINTE 位置“1”后，通过轮询确认 RINTE 位变为“1”，然后转移到 HALT 模式或者 STOP 模式（参照图 7-6 的例 1）。
- 在将 RINTE 位置“1”后至少经过 1 个计数时钟的时间之后再转移到 HALT 模式或者 STOP 模式（参照图 7-6 的例 2）。

图 7-6 将 RINTE 位置“1”后的 HALT 模式或者 STOP 模式的转移步骤



第 8 章 时钟输出 / 蜂鸣器输出控制电路

时钟输出 / 蜂鸣器输出控制电路的输出引脚数因产品而不同。

输出引脚	本产品
PCLBUZ0	○
PCLBUZ1	○

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。

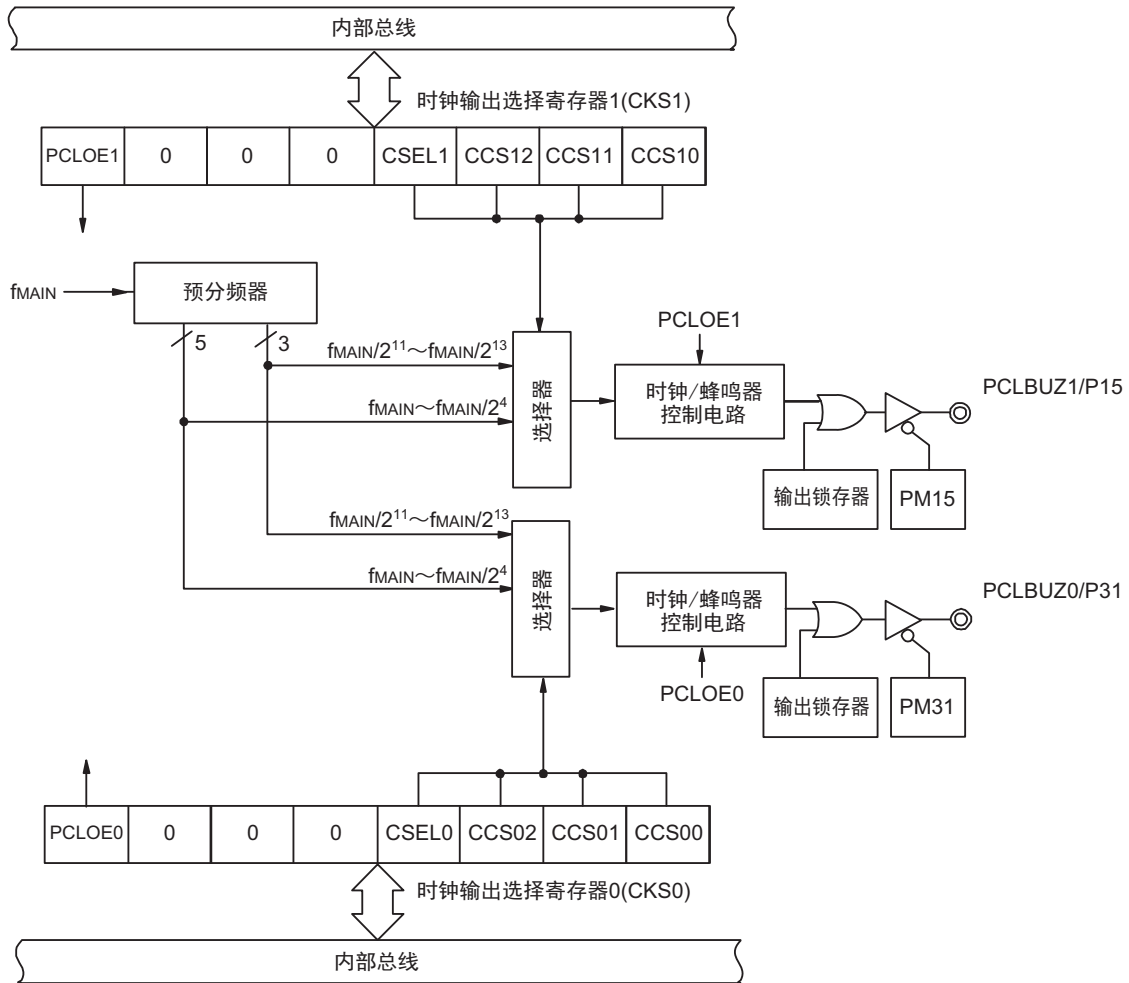
能用 1 个引脚选择用作时钟输出或者蜂鸣器输出。

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

时钟输出 / 蜂鸣器输出控制电路的框图如图 8-1 所示。

备注 n=0、1

图 8-1 时钟输出 / 蜂鸣器输出控制电路的框图



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

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 n (CKSn) 端口模式寄存器 1、3 (PM1、PM3) 端口寄存器 1、3 (P1、P3)

8.3 控制时钟输出 / 蜂鸣器输出控制电路的寄存器

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 n (CKSn)
- 端口模式寄存器 1、3 (PM1、PM3)
- 端口寄存器 1、3 (P1、P3)

8.3.1 时钟输出选择寄存器 n (CKSn)

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

通过 CKSn 寄存器选择 PCLBUZn 引脚输出的时钟。

通过 1 位或者 8 位存储器操作指令设定 CKSn 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 8-2 时钟输出选择寄存器 n (CKSn) 的格式

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

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

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

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn 引脚输出时钟的选择				
				$f_{MAIN}=5MHz$	$f_{MAIN}=10MHz$	$f_{MAIN}=20MHz$	$f_{MAIN}=24MHz$	
0	0	0	0	f_{MAIN}	5MHz	10MHz 注	禁止设定注	禁止设定注
0	0	0	1	$f_{MAIN}/2$	2.5MHz	5MHz	10MHz 注	12MHz 注
0	0	1	0	$f_{MAIN}/2^2$	1.25MHz	2.5MHz	5MHz	6MHz 注
0	0	1	1	$f_{MAIN}/2^3$	625kHz	1.25MHz	2.5MHz	3MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5kHz	625kHz	1.25MHz	1.5MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44kHz	4.88kHz	9.76kHz	11.7kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22kHz	2.44kHz	4.88kHz	5.86kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610Hz	1.22kHz	2.44kHz	2.93kHz

注 必须在 16MHz 以内的范围内使用输出时钟。详细内容请参照“27.4 AC 特性”。

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

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

备注 1. n=0、1

2. f_{MAIN} : 主系统时钟频率

8.3.2 控制时钟输出 / 蜂鸣器输出引脚端口功能的寄存器

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

在将时钟输出 / 蜂鸣器输出引脚的复用端口用作时钟输出 / 蜂鸣器输出时，必须将各端口对应的端口模式寄存器（PMxx）的位和端口寄存器（Pxx）的位置“0”。

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。

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

8.4.1 输出引脚的运行

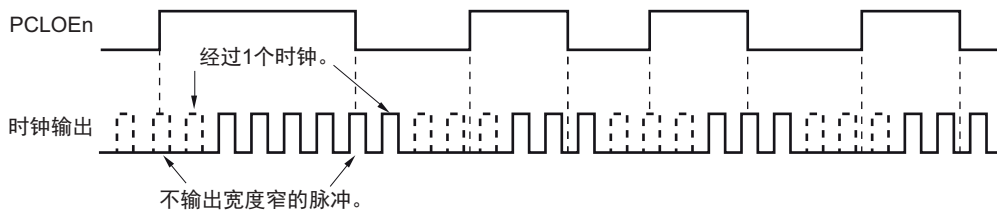
PCLBUZn 引脚按照以下步骤进行输出：

- ① 将用作 PCLBUZ0 引脚的端口对应的端口模式寄存器（PMxx）和端口寄存器（Pxx）的位置“0”。
- ② 通过 PCLBUZn 引脚的时钟输出选择寄存器（CKSn）的 bit0～3（CCSn0～CCSn2、CSELn）选择输出频率（输出为禁止状态）。
- ③ 将 CKSn 寄存器的 bit7（PCLOEn）置“1”，允许时钟 / 蜂鸣器的输出。

备注 1. 用作时钟输出时的控制电路在允许或者禁止时钟输出（PCLOEn 位）后的 1 个时钟之后，开始或者停止时钟输出。此时不输出宽度窄的脉冲。通过 PCLOEn 位允许或者停止输出以及时钟输出的时序如图 8-3 所示。

2. n=0、1

图 8-3 PCLBUZn 引脚的时钟输出时序



8.5 时钟输出 / 蜂鸣器输出控制电路的注意事项

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

第 9 章 看门狗定时器

9.1 看门狗定时器的功能

看门狗定时器通过选项字节 (000C0H) 设定计数运行。

看门狗定时器以低速内部振荡器时钟 (f_{IL}) 运行。

看门狗定时器用于检测程序失控。在检测到程序失控时, 产生内部复位信号。

下述情况判断为程序失控。

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

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

当达到上溢时间的 $75\%+1/2f_{IL}$ 时, 能产生间隔中断。

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

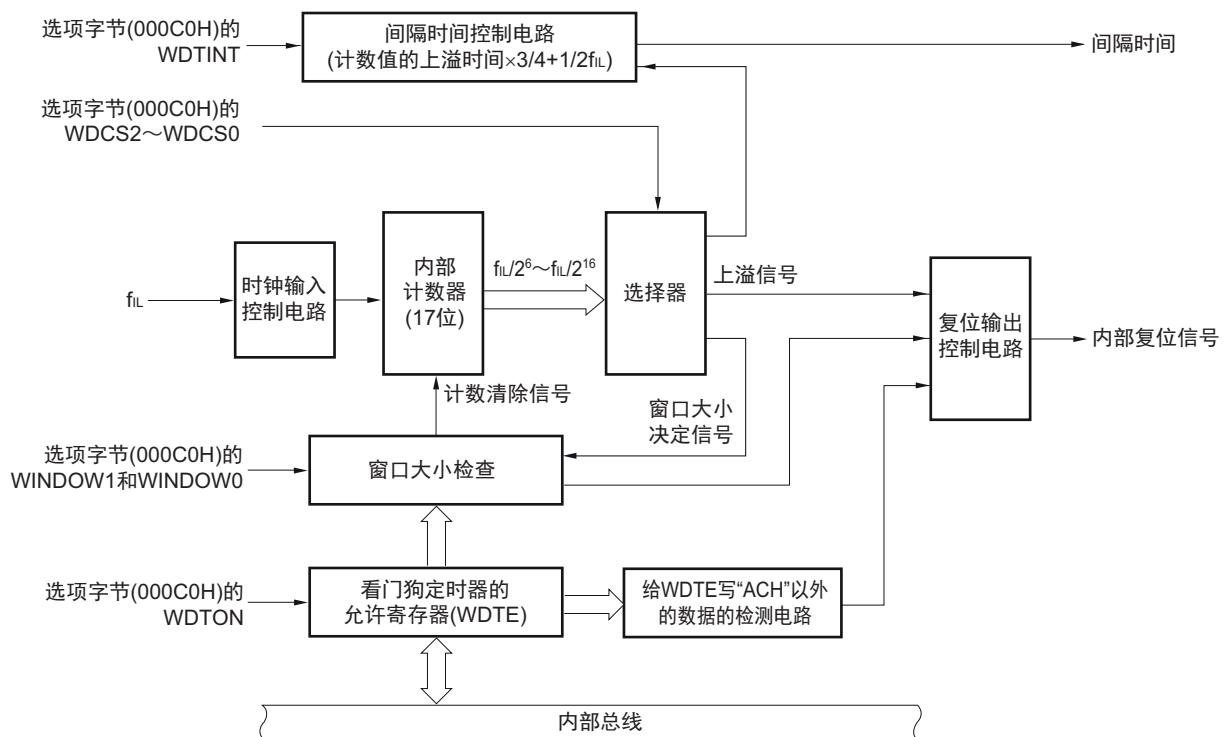
通过选项字节控制计数器的运行以及设定上溢时间、窗口打开期间和间隔中断。

表 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)
看门狗定时器的间隔中断的设定	bit7 (WDTINT)
窗口打开期间的设定	bit6 和 bit5 (WINDOW1、WINDOW0)
看门狗定时器的计数器运行控制	bit4 (WDTON)
看门狗定时器的上溢时间的设定	bit3 ~ 1 (WDCS2 ~ WDCS0)
看门狗定时器的计数器运行控制 (HALT/STOP 模式时)	bit0 (WDSTBYON)

备注 有关选项字节, 请参照“第 22 章 选项字节”。

图 9-1 看门狗定时器的框图



备注 f_{IL} : 低速内部振荡器的时钟频率

9.3 控制看门狗定时器的寄存器

通过看门狗定时器的允许寄存器（WDTE）控制看门狗定时器。

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“9AH”或者“1AH”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: 9AH/1AH 注	R/W						
符号	7	6	5	4	3	2	1	0
WDTE								

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

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

注意 1. 当给 WDTE 寄存器写“ACH”以外的值时，产生内部复位信号。

2. 当对 WDTE 寄存器执行 1 位存储器操作指令时，产生内部复位信号。

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

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22章）。

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

- 必须通过选项字节（000C0H）的 bit3 ~ 1（WDCS2 ~ WDSC0）设定上溢时间（详细内容请参照 9.4.2 和第 22 章）。
 - 必须通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定窗口打开期间（详细内容请参照 9.4.3 和第 22 章）。
- 在解除复位后，看门狗定时器开始计数。
 - 在开始计数后并且在选项字节所设上溢时间前，如果给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
 - 此后，解除复位后第 2 次以后的 WDTE 寄存器的写操作必须在窗口打开期间进行。如果在窗口关闭期间写 WDTE 寄存器，就产生内部复位信号。
 - 如果不给 WDTE 寄存器写“ACH”而超过上溢时间，就产生内部复位信号。
以下情况会产生内部复位信号：
 - 当对 WDTE 寄存器执行 1 位操作指令时
 - 当给 WDTE 寄存器写“ACH”以外的数据时

- 注意 1. 只在解除复位后第 1 次写看门狗定时器的允许寄存器（WDTE）时，与窗口打开期间无关，只要在上溢时间前的任意时候写 WDTE，就清除看门狗定时器并且重新开始计数。
- 从给 WDTE 寄存器写“ACH”到清除看门狗定时器的计数器为止，有可能产生最大 2 个 f_{IL} 时钟的误差。
 - 在计数值发生上溢前，都能清除看门狗定时器。
 - 如下所示，看门狗定时器在 HALT 模式、STOP 模式或者 SNOOZE 模式中的运行因选项字节（000C0H）的 bit0（WDSTBYON）的设定值而不同。

	WDSTBYON=0	WDSTBYON=1
HALT 模式	停止看门狗定时器运行。	继续看门狗定时器运行。
STOP 模式		
SNOOZE 模式		

当 WDSTBYON 位为“0”时，在解除 HALT 模式或者 STOP 模式后重新开始看门狗定时器的计数。此时，将计数器清“0”，开始计数。

当解除 STOP 模式后以 X1 振荡时钟运行时，CPU 在经过振荡稳定时间后开始运行。

如果从解除 STOP 模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除 STOP 模式后，如果要以 X1 振荡时钟运行并且清除看门狗定时器，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。

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。

在发生上溢时，产生内部复位信号。如果在上溢时间前的窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除计数并且重新开始计数。

能设定的上溢时间如下所示。

表 9-3 看门狗定时器上溢时间的设定

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$ 的情况)
0	0	0	$2^6/f_{IL}$ (3.71ms)
0	0	1	$2^7/f_{IL}$ (7.42ms)
0	1	0	$2^8/f_{IL}$ (14.84ms)
0	1	1	$2^9/f_{IL}$ (29.68ms)
1	0	0	$2^{11}/f_{IL}$ (118.72ms)
1	0	1	$2^{13}/f_{IL}$ (474.89ms) 注
1	1	0	$2^{14}/f_{IL}$ (949.79ms) 注
1	1	1	$2^{16}/f_{IL}$ (3799.18ms) 注

注 如果满足以下的各使用条件，就有可能在清除看门狗定时器的计数后，经过 1 个看门狗定时器时钟，发生看门狗定时器的间隔中断（INTWDTI）。通过执行步骤①~⑤清除看门狗定时器的计数，能屏蔽此间隔中断。

〈使用条件〉

- 将看门狗定时器的上溢时间设定为 $2^{13}/f_{IL}$ 、 $2^{14}/f_{IL}$ 或者 $2^{16}/f_{IL}$
- 使用看门狗定时器的间隔中断
- 当看门狗定时器的计数值不小于上溢时间的 75% 时，给 WDTE 寄存器（FFFABH）写“ACH”。

①在清除看门狗定时器的计数前，将中断屏蔽标志寄存器 0（MK0L）的 WDTIMK 位置“1”。

②清除看门狗定时器的计数器

③至少等待 80 μ s

④将中断请求标志寄存器 0（IF0L）的 WDTIF 位清“0”

⑤将中断屏蔽标志寄存器 0（MK0L）的 WDTIMK 位清“0”

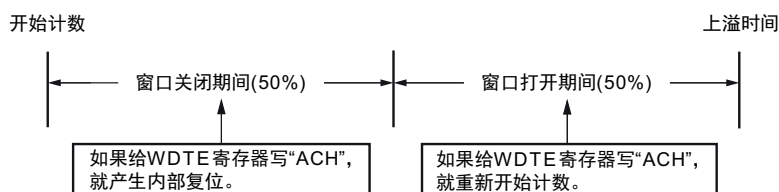
备注 f_{IL} ：低速内部振荡器的时钟频率

9.4.3 看门狗定时器窗口打开期间的设定

通过选项字节（000C0H）的 bit6 和 bit5（WINDOW1、WINDOW0）设定看门狗定时器的窗口打开期间。窗口概要如下：

- 如果在窗口打开期间给看门狗定时器的允许寄存器（WDTE）写“ACH”，就清除看门狗定时器并且重新开始计数。
- 在窗口关闭期间，即使给WDTE寄存器写“ACH”，也会检测到异常并且产生内部复位信号。

例 窗口打开期间为 50% 的情况



注意 只在解除复位后第 1 次写 WDTE 寄存器时，与窗口打开期间无关，只要在上溢时间前的任意时候写 WDTE，就清除看门狗定时器并且重新开始计数。

能设定的窗口打开期间如下所示。

看门狗定时器窗口打开期间的设定

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间
0	0	禁止设定。
0	1	50%
1	0	75% 注
1	1	100%

注 在将窗口打开期间设定为 75% 时，如果进行看门狗定时器的计数器清除（给 WDTE 写“ACH”），就必须在下表所示的计数器清除禁止期间外的时序进行看门狗定时器的间隔中断请求标志（WDTIIF）确认等。

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)	将窗口打开期间设定为 75% 时的计数器清除禁止期间
0	0	0	$2^6/f_{IL}$ (3.71ms)	1.85 ~ 2.51ms
0	0	1	$2^7/f_{IL}$ (7.42ms)	3.71 ~ 5.02ms
0	1	0	$2^8/f_{IL}$ (14.84ms)	7.42 ~ 10.04ms
0	1	1	$2^9/f_{IL}$ (29.68ms)	14.84 ~ 20.08ms
1	0	0	$2^{11}/f_{IL}$ (118.72ms)	56.36 ~ 80.32ms
1	0	1	$2^{13}/f_{IL}$ (474.89ms)	237.44 ~ 321.26ms
1	1	0	$2^{14}/f_{IL}$ (949.79ms)	474.89 ~ 642.51ms
1	1	1	$2^{16}/f_{IL}$ (3799.18ms)	1899.59 ~ 2570.04ms

注意 当选项字节（000C0H）的 bit0（WDSTBYON）为“0”时，与 WINDOW1 位和 WINDOW0 位的值无关，窗口打开期间为 100%。

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

	窗口打开期间的设定		
	50%	75%	100%
窗口关闭时间	0 ~ 20.08ms	0 ~ 10.04ms	无
窗口打开时间	20.08 ~ 29.68ms	10.04 ~ 29.68ms	0 ~ 29.68ms

<当窗口打开期间为50%时>

- 上溢时间：
 $2^9/f_{IL}(\text{MAX.})=2^9/17.25\text{kHz}(\text{MAX.})=29.68\text{ms}$
- 窗口关闭时间：
 $0\sim 2^9/f_{IL}(\text{MIN.})\times(1-0.5)=0\sim 2^9/12.75\text{kHz}\times 0.5=0\sim 20.08\text{ms}$
- 窗口打开时间：
 $2^9/f_{IL}(\text{MIN.})\times(1-0.5)\sim 2^9/f_{IL}(\text{MAX.})=2^9/12.75\text{kHz}\times 0.5\sim 2^9/17.25\text{kHz}=20.08\sim 29.68\text{ms}$

9.4.4 看门狗定时器间隔中断的设定

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

表 9-4 看门狗定时器间隔中断的设定

WDTINT	看门狗定时器间隔中断的使用 / 不使用
0	不使用间隔中断。
1	在达到上溢时间的75%+1/2 f_{IL} 时，产生间隔中断。

注意 当解除 STOP 模式后以 X1 振荡时钟运行时，CPU 在经过振荡稳定时间后开始运行。如果从解除 STOP 模式到看门狗定时器发生上溢的时间较短，就会在振荡稳定时间内发生看门狗上溢而产生复位。因此，在通过间隔中断解除 STOP 模式后，如果要以 X1 振荡时钟运行并且清除看门狗定时器，因为在经过振荡稳定时间后才清除看门狗定时器，所以必须考虑这种情况进行上溢时间的设定。

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

第 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。

本产品的 A/D 转换器有 8 个模拟输入通道（ANI0 ~ ANI3、ANI16 ~ ANI19）。

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 8 个通道的 A/D 转换器模拟输入（ANI0 ~ ANI3、ANI16 ~ ANI19）。能通过 A/D 转换器的模式寄存器 2（ADM2）的 ADTYP 位选择 10 位分辨率和 8 位分辨率。

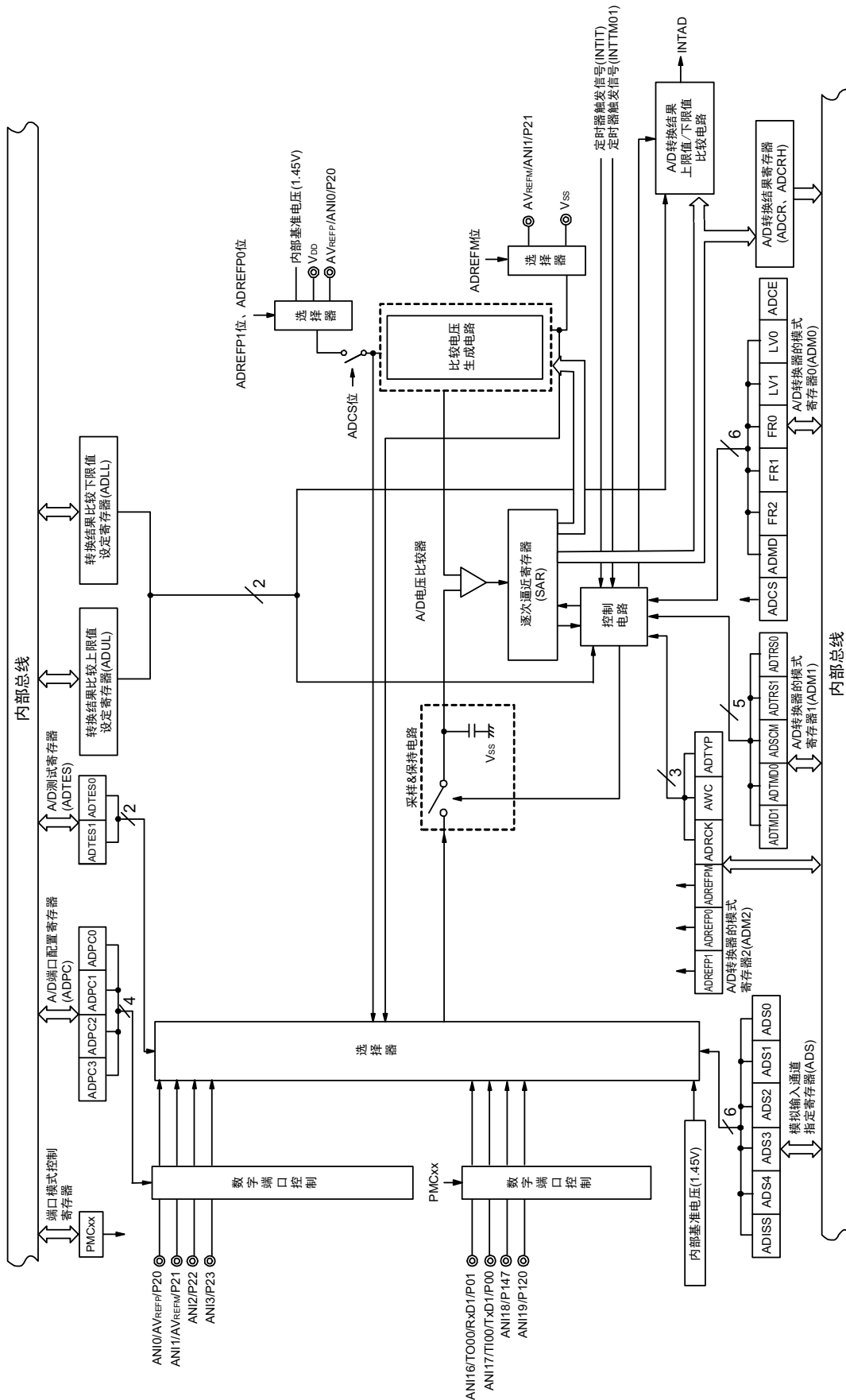
A/D 转换器有以下功能。

- 10 位或者 8 位分辨率的 A/D 转换
从 ANI0 ~ ANI3、ANI16 ~ ANI19 中选择 1 个通道的模拟输入，重复进行 10 位或者 8 位分辨率的 A/D 转换。每结束 1 次 A/D 转换，就产生中断请求（INTAD）（选择模式的情况）。

能通过下述的模式组合设定各种 A/D 转换模式。

触发模式	软件触发	通过软件操作来开始转换。
	硬件触发无等待模式	通过检测硬件触发来开始转换。
	硬件触发等待模式	在切断电源的转换待机状态下，通过检测硬件触发来接通电源，在经过 A/D 电源稳定等待时间后自动开始转换。当使用 SNOOZE 模式功能时，必须选择硬件触发等待模式。
通道选择模式	选择模式	选择 1 个通道的模拟输入，进行 A/D 转换。
	扫描模式	按顺序对 4 个通道的模拟输入进行 A/D 转换。能选择 ANI0 ~ ANI3 中连续的 4 个通道作为模拟输入。
转换模式	单次转换模式	对所选通道进行 1 次 A/D 转换。
	连续转换模式	对所选通道连续进行 A/D 转换，直到被软件停止为止。
工作电压模式	标准 1/ 标准 2 模式	用于在 $2.7V \leq V_{DD} \leq 5.5V$ 的工作电压范围内转换的情况。
	低电压 1/ 低电压 2 模式	能在 $1.6V \leq V_{DD} \leq 5.5V$ 的工作电压范围内进行转换。 用于在低电压时转换的情况。因为在低电压下运行，所以在转换时内部栅极升压。
采样时间的选择	采样时钟数：7 个 f_{AD}	标准 1/ 低电压 1 模式的采样时间为 7 个转换时钟（ f_{AD} ）。用于因模拟输入源的输出阻抗高而想要延长采样时间的情况。
	采样时钟数：5 个 f_{AD}	标准 2/ 低电压 2 模式的采样时间为 5 个转换时钟（ f_{AD} ）。用于因模拟输入源的输出阻抗低而能充分确保采样时间的情况。

图 10-1 A/D 转换器的框图



10.2 A/D 转换器的结构

A/D 转换器由以下硬件构成。

(1) ANI0 ~ ANI3 引脚和 ANI16 ~ ANI19 引脚

这些是 A/D 转换器的 8 个通道的模拟输入引脚，输入要进行 A/D 转换的模拟信号。没有被选择为模拟输入的引脚能用作输入 / 输出端口。

(2) 采样 & 保持电路

采样 & 保持电路依次对来自输入电路的模拟输入电压进行采样并且送给 A/D 电压比较器。在 A/D 转换过程中，保持采样到的模拟输入电压。

(3) A/D 电压比较器

通过 A/D 电压比较器将比较电压生成电路产生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 ($1/2AV_{REF}$)，就将逐次逼近寄存器 (SAR) 的最高位 (MSB) 置位。如果模拟输入电压小于基准电压 ($1/2AV_{REF}$)，就对 SAR 寄存器的 MSB 位进行复位。

然后，自动将 SAR 寄存器的 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: ($1/4AV_{REF}$)

bit9=1: ($3/4AV_{REF}$)

将比较电压生成电路的分接头电压和模拟输入电压进行比较，根据比较结果操作 SAR 寄存器的 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

将此比较继续进行到 SAR 寄存器的 bit0 为止。

当以 8 位分辨率进行 A/D 转换时，比较继续进行到 SAR 寄存器的 bit2 为止。

备注 AV_{REF} 是 A/D 转换器的正 (+) 基准电压。可选择 AV_{REFP} 、内部基准电压 (1.45V) 或者 V_{DD} 。

(4) 比较电压生成电路

比较电压生成电路生成模拟输入电压的比较电压。

(5) 逐次逼近寄存器 (SAR: Successive Approximation Register)

SAR 寄存器从最高位 (MSB) 按位设定比较电压生成电路的分接头电压值和模拟输入引脚的电压值相同的数据。

如果设定到 SAR 寄存器的最低位 (LSB) (A/D 转换结束)，就将 SAR 寄存器的内容 (转换结果) 保持到 A/D 转换结果寄存器 (ADCR)。而且，如果指定的全部 A/D 转换都结束，就产生 A/D 转换结束中断请求信号 (INTAD)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每当 A/D 转换结束时, 就将逐次逼近寄存器的转换结果装入 ADCR 寄存器, A/D 转换结果保持在高 10 位 (低 6 位固定为“0”)。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每当 A/D 转换结束时, 就将逐次逼近寄存器的转换结果装入 ADCRH 寄存器, 保存 A/D 转换结果的高 8 位。

(8) 控制电路

此电路控制要进行 A/D 转换的模拟输入的转换时间、转换的开始或者停止等。当 A/D 转换结束时, 通过 A/D 转换结果上限值 / 下限值比较电路产生 INTAD 信号。

(9) AV_{REFP} 引脚

这是从外部输入基准电压 (AV_{REFP}) 的引脚。

要将 AV_{REFP} 用作 A/D 转换器的正 (+) 基准电压时, 必须将 A/D 转换器的模式寄存器 2 (ADM2) 的 ADREFP1 位和 ADREFP0 位分别置“0”和“1”。

根据加载的 AV_{REFP} 和负 (-) 基准电压 (AV_{REFM}/V_{SS}) 之间的电压, 将 ANI2 ~ ANI3、ANI16 ~ ANI19 输入的模拟信号转换为数字信号。

除了 AV_{REFP} 以外, 可选择 V_{DD} 或者内部基准电压 (1.45V) 作为 A/D 转换器的正 (+) 基准电压。

(10) AV_{REFM} 引脚

这是从外部输入基准电压 (AV_{REFM}) 的引脚。要将 AV_{REFM} 用作 A/D 转换器的负 (-) 基准电压时, 必须将 ADM2 寄存器的 ADREFM 位置“1”。

除了 AV_{REFM} 以外, 可选择 V_{SS} 作为 A/D 转换器的负 (-) 基准电压。

10.3 控制 A/D 转换器的寄存器

控制 A/D 转换器的寄存器如下所示:

- 外围允许寄存器 0 (PER0)
- A/D 转换器的模式寄存器 0 (ADM0)
- A/D 转换器的模式寄存器 1 (ADM1)
- A/D 转换器的模式寄存器 2 (ADM2)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)
- 模拟输入通道指定寄存器 (ADS)
- 转换结果比较上限值设定寄存器 (ADUL)
- 转换结果比较下限值设定寄存器 (ADLL)
- A/D 测试寄存器 (ADTES)
- A/D 端口配置寄存器 (ADPC)
- 端口模式控制寄存器 0、12、14 (PMC0、PMC12、PMC14)
- 端口模式寄存器 0、2、12、14 (PM0、PM2、PM12、PM14)

10.3.1 外围允许寄存器 0 (PER0)

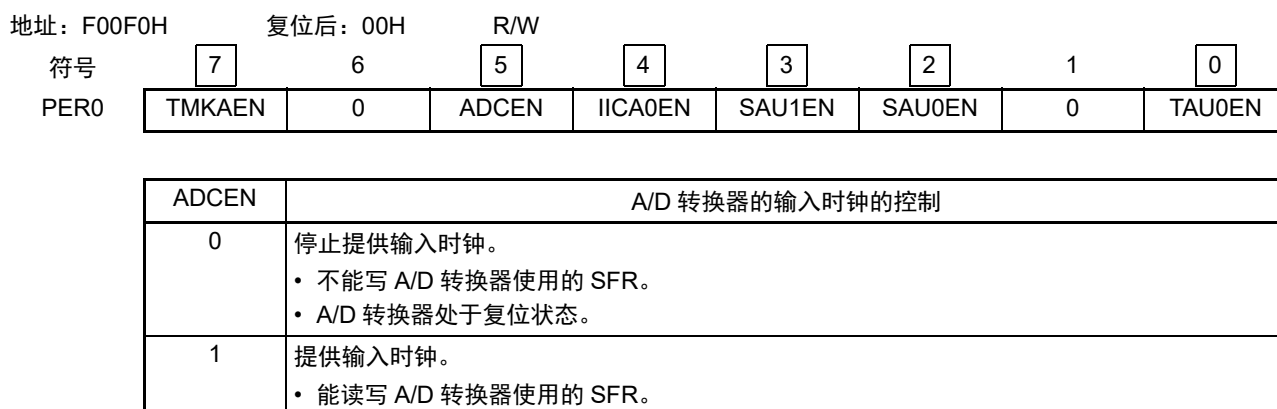
PER0 寄存器是设定允许或者禁止给各外围硬件提供时钟的寄存器。通过停止给不使用的硬件提供时钟，以降低功耗和噪声。

要使用 A/D 转换器时，必须将 bit5 (ADCEN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-2 外围允许寄存器 0 (PER0) 的格式



注意 1. 要设定 A/D 转换器时，必须先在 ADCEN 位为“1”的状态下设定以下的寄存器。当 ADCEN 位为“0”时，A/D 转换器的控制寄存器的值为初始值，忽视写操作（端口模式寄存器 0、2、12、14 (PM0、PM2、PM12、PM14)、端口模式控制寄存器 0、12、14 (PMC0、PMC12、PMC14) 以及 A/D 端口配置寄存器 (ADPC) 除外)。

- A/D 转换器的模式寄存器 0 (ADM0)
- A/D 转换器的模式寄存器 1 (ADM1)
- A/D 转换器的模式寄存器 2 (ADM2)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)
- 模拟输入通道指定寄存器 (ADS)
- 转换结果比较上限值设定寄存器 (ADUL)
- 转换结果比较下限值设定寄存器 (ADLL)
- A/D 测试寄存器 (ADTES)

2. 必须将 bit6 和 bit1 置“0”。

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	ADMD	FR2 注 1	FR1 注 1	FR0 注 1	LV1 注 1	LV0 注 1	ADCE
ADCS	A/D 转换运行的控制							
0	停止转换运行。 [读时] 停止转换运行 / 待机状态							
1	允许转换运行。 [读时] 软件触发模式时: 转换运行状态 硬件触发等待模式时: A/D 电源等待稳定状态 + 转换运行状态							
ADMD	A/D 转换通道选择模式的设定							
0	选择模式							
1	扫描模式							
ADCE	A/D 电压比较器的运行控制注 2							
0	停止 A/D 电压比较器的运行。							
1	允许 A/D 电压比较器的运行。							

- 注 1. 有关 FR2 ~ FR0 位、LV1 位、LV0 位和 A/D 转换的详细内容，请参照“表 10-3 A/D 转换时间的选择”。
2. 在软件触发模式或者硬件触发无等待模式中，ADCS 位和 ADCE 位控制 A/D 电压比较器的运行，从开始运行到稳定为止需要 1μs 时间。因此，在将 ADCE 位置“1”后至少经过 1μs 时间，然后将 ADCS 位置“1”，从而最初的转换数据开始有效。如果不至少等待 1μs 而将 ADCS 位置“1”，就必须忽视最初的转换数据。

- 注意 1. 必须在转换停止状态 (ADCS=0、ADCE=0) 下更改 ADMD 位、FR2 ~ FR0 位、LV1 位和 LV0 位。
2. 禁止进行 ADCS=1、ADCE=0 的设定。
3. 禁止通过 8 位操作指令将 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。

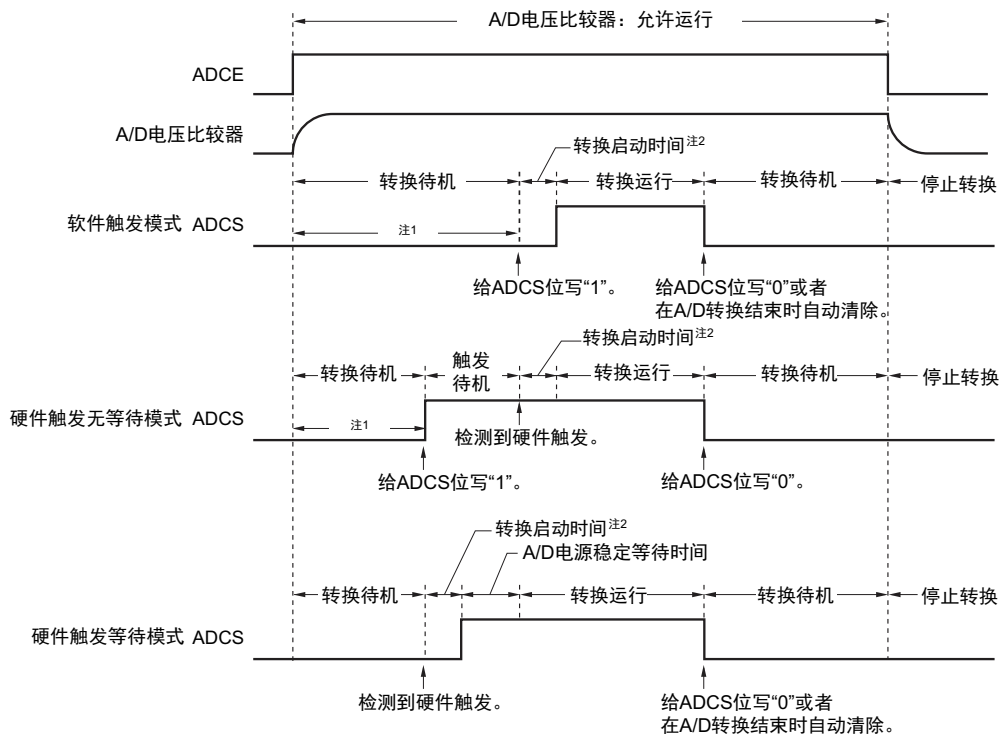
表 10-1 ADCS 位和 ADCE 位的设定

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

表 10-2 ADCS 位的置位和清除条件

A/D 转换模式			置位条件	清除条件
软件触发	选择模式	连续转换模式	当给 ADCS 位写“1”时	当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> 当给 ADCS 位写“0”时 在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> 当给 ADCS 位写“0”时 当设定的 4 个通道转换结束时，自动清“0”。
硬件触发无等待模式	选择模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		当给 ADCS 位写“0”时
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		当给 ADCS 位写“0”时
硬件触发等待模式	选择模式	连续转换模式	当输入硬件触发时	当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> 当给 ADCS 位写“0”时 在 A/D 转换结束时自动清“0”。
	扫描模式	连续转换模式		当给 ADCS 位写“0”时
		单次转换模式		<ul style="list-style-type: none"> 当给 ADCS 位写“0”时 当设定的 4 个通道转换结束时，自动清“0”。

图 10-4 使用 A/D 电压比较器时的时序图



- 注 1. 在软件触发模式或者硬件触发无等待模式中，为了稳定内部电路，从 ADCE 位上升到 ADCS 位上升的时间至少需要 1 μ s。
 2. 转换的启动最长需要以下的时间：

ADM0			转换时钟 (f_{AD})	转换启动时间 (f_{CLK} 时钟数)	
FR2	FR1	FR0		软件触发模式 / 硬件触发无等待模式	硬件触发等待模式
0	0	0	$f_{CLK}/64$	63	1
0	0	1	$f_{CLK}/32$	31	
0	1	0	$f_{CLK}/16$	15	
0	1	1	$f_{CLK}/8$	7	
1	0	0	$f_{CLK}/6$	5	
1	0	1	$f_{CLK}/5$	4	
1	1	0	$f_{CLK}/4$	3	
1	1	1	$f_{CLK}/2$	1	

但是，在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中检测到硬件触发后，不发生转换启动时间和 A/D 电源稳定等待时间。

- 注意 1. 要使用硬件触发等待模式时，禁止将 ADCS 位置“1”（当检测到硬件触发信号时，自动切换为“1”）。但是，为了设定为 A/D 转换待机状态，能将 ADCS 位置“0”。
 2. 在硬件触发无等待模式的单次转换模式中，ADCS 标志在 A/D 转换结束时不自动清“0”，而保持“1”的状态。
 3. 必须在 ADCS 位为“0”（停止转换 / 转换待机状态）时改写 ADCE 位。
 4. 为了结束 A/D 转换，必须至少将硬件触发间隔设定为以下时间：
 硬件触发无等待模式时：2 个 f_{CLK} 时钟 + 转换启动时间 + A/D 转换时间
 硬件触发等待模式时：2 个 f_{CLK} 时钟 + 转换启动时间 + A/D 电源稳定等待时间 + A/D 转换时间

备注 f_{CLK} : CPU/ 外围硬件的时钟频率

表 10-3 A/D 转换时间的选择 (1/4)

(1) 无 A/D 电源稳定等待时间 标准模式 1、2 (软件触发模式 / 硬件触发无等待模式)

A/D 转换器的模式寄存器 0 (ADM0)					模式	转换时钟 (f_{AD})	转换时钟数注	转换时间	10 位分辨率时的转换时间					
FR2	FR1	FR0	LV1	LV0					$2.7V \leq V_{DD} \leq 5.5V$					
									$f_{CLK}=1MHz$	$f_{CLK}=4MHz$	$f_{CLK}=8MHz$	$f_{CLK}=16MHz$	$f_{CLK}=24MHz$	
0	0	0	0	0	标准 1	$f_{CLK}/64$	19 个 f_{AD} (采样时钟数: 7 个 f_{AD})	$1216/f_{CLK}$	禁止设定	禁止设定	禁止设定	$76\mu s$	$50.67\mu s$	
0	0	1				$f_{CLK}/32$		$608/f_{CLK}$			$76\mu s$	$38\mu s$	$25.33\mu s$	
0	1	0				$f_{CLK}/16$		$304/f_{CLK}$			$76\mu s$	$38\mu s$	$19\mu s$	$12.67\mu s$
0	1	1				$f_{CLK}/8$		$152/f_{CLK}$			$38\mu s$	$19\mu s$	$9.5\mu s$	$6.33\mu s$
1	0	0				$f_{CLK}/6$		$114/f_{CLK}$			$28.5\mu s$	$14.25\mu s$	$7.125\mu s$	$4.75\mu s$
1	0	1				$f_{CLK}/5$		$95/f_{CLK}$	$95\mu s$	$23.75\mu s$	$11.875\mu s$	$5.938\mu s$	$3.96\mu s$	
1	1	0				$f_{CLK}/4$		$76/f_{CLK}$	$76\mu s$	$19\mu s$	$9.5\mu s$	$4.75\mu s$	$3.17\mu s$	
1	1	1				$f_{CLK}/2$		$38/f_{CLK}$	$38\mu s$	$9.5\mu s$	$4.75\mu s$	$2.375\mu s$	禁止设定	
0	0	0	0	1	标准 2	$f_{CLK}/64$	17 个 f_{AD} (采样时钟数: 5 个 f_{AD})	$1088/f_{CLK}$	禁止设定	禁止设定	禁止设定	$68\mu s$	$45.33\mu s$	
0	0	1				$f_{CLK}/32$		$544/f_{CLK}$			$68\mu s$	$34\mu s$	$22.67\mu s$	
0	1	0				$f_{CLK}/16$		$272/f_{CLK}$			$68\mu s$	$34\mu s$	$17\mu s$	$11.33\mu s$
0	1	1				$f_{CLK}/8$		$136/f_{CLK}$			$34\mu s$	$17\mu s$	$8.5\mu s$	$5.67\mu s$
1	0	0				$f_{CLK}/6$		$102/f_{CLK}$			$25.5\mu s$	$12.75\mu s$	$6.375\mu s$	$4.25\mu s$
1	0	1				$f_{CLK}/5$		$85/f_{CLK}$	$85\mu s$	$21.25\mu s$	$10.625\mu s$	$5.3125\mu s$	$3.54\mu s$	
1	1	0				$f_{CLK}/4$		$68/f_{CLK}$	$68\mu s$	$17\mu s$	$8.5\mu s$	$4.25\mu s$	$2.83\mu s$	
1	1	1				$f_{CLK}/2$		$34/f_{CLK}$	$34\mu s$	$8.5\mu s$	$4.25\mu s$	$2.125\mu s$	禁止设定	

注 这是 10 位分辨率时的转换时钟数。如果选择 8 位分辨率，就会减少 2 个转换时钟 (f_{AD})。

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 (t_{CONV}) 范围内。

2. 要将 FR2 ~ FR0 位、LV1 位和 LV0 位改写为不同数据时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
3. 转换时间不包含转换启动时间，因此在第 1 次转换时必须加上转换启动时间。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

表 10-3 A/D 转换时间的选择 (2/4)

(2) 无 A/D 电源稳定等待时间 低电压模式 1、2 (软件触发模式 / 硬件触发无等待模式)

A/D 转换器的模式寄存器 0 (ADM0)					模式	转换时钟 (f _{AD})	转换时钟数注 4	转换时间	10 位分辨率时的转换时间						
FR2	FR1	FR0	LV1	LV0					1.6V ≤ V _{DD} ≤ 5.5V		注 1	注 2	注 3		
									f _{CLK} = 1MHz	f _{CLK} = 4MHz	f _{CLK} = 8MHz	f _{CLK} = 16MHz	f _{CLK} = 24MHz		
0	0	0	1	0	低电压1	f _{CLK} /64	19 个 f _{AD} (采样时钟数: 7 个 f _{AD})	1216/f _{CLK}	禁止设定	禁止设定	禁止设定	76μs	50.67μs		
0	0	1		f _{CLK} /32		608/f _{CLK}		76μs				38μs	25.33μs		
0	1	0		f _{CLK} /16		304/f _{CLK}		76μs				38μs	19μs	12.67μs	
0	1	1		f _{CLK} /8		152/f _{CLK}		38μs				19μs	9.5μs	6.33μs	
1	0	0		f _{CLK} /6		114/f _{CLK}		28.5μs				14.25μs	7.125μs	4.75μs	
1	0	1		f _{CLK} /5		95/f _{CLK}		95μs				23.75μs	11.875μs	5.938μs	3.96μs
1	1	0		f _{CLK} /4		76/f _{CLK}		76μs				19μs	9.5μs	4.75μs	3.17μs
1	1	1		f _{CLK} /2		38/f _{CLK}		38μs				9.5μs	4.75μs	2.375μs	禁止设定
0	0	0	1	1	低电压2	f _{CLK} /64	17 个 f _{AD} (采样时钟数: 5 个 f _{AD})	1088/f _{CLK}	禁止设定	禁止设定	禁止设定	68μs	45.33μs		
0	0	1		f _{CLK} /32		544/f _{CLK}		68μs				34μs	22.667μs		
0	1	0		f _{CLK} /16		272/f _{CLK}		68μs				34μs	17μs	11.333μs	
0	1	1		f _{CLK} /8		136/f _{CLK}		34μs				17μs	8.5μs	5.667μs	
1	0	0		f _{CLK} /6		102/f _{CLK}		25.5μs				12.75μs	6.375μs	4.25μs	
1	0	1		f _{CLK} /5		85/f _{CLK}		85μs				21.25μs	10.625μs	5.3125μs	3.542μs
1	1	0		f _{CLK} /4		68/f _{CLK}		68μs				17μs	8.5μs	4.25μs	2.833μs
1	1	1		f _{CLK} /2		34/f _{CLK}		34μs				8.5μs	4.25μs	2.125μs	禁止设定

注 1. 1.8V ≤ V_{DD} ≤ 5.5V2. 2.4V ≤ V_{DD} ≤ 5.5V3. 2.7V ≤ V_{DD} ≤ 5.5V4. 这是 10 位分辨率时的转换时钟数。如果选择 8 位分辨率，就会减少 2 个转换时钟 (f_{AD})。注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 (t_{CONV}) 范围内。

2. 要将 FR2 ~ FR0 位、LV1 位和 LV0 位改写为不同数据时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
3. 转换时间不包含转换启动时间，因此在第 1 次转换时必须加上转换启动时间。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

表 10-3 A/D 转换时间的选择 (3/4)

(3) 有 A/D 电源稳定等待时间 标准模式 1、2 (硬件触发等待模式注 1)

A/D 转换器的模式寄存器 0 (ADM0)					模式	转换时钟 (f_{AD})	A/D 电源稳定等待时钟数	转换时钟数注 2	A/D 电源稳定等待时间 + 转换时间	A/D 电源稳定等待时间 + 10 位分辨率时的转换时间					
FR2	FR1	FR0	LV1	LV0						2.7V \leq V _{DD} \leq 5.5V					
										$f_{CLK}=1\text{MHz}$	$f_{CLK}=4\text{MHz}$	$f_{CLK}=8\text{MHz}$	$f_{CLK}=16\text{MHz}$	$f_{CLK}=24\text{MHz}$	
0	0	0	0	0	标准 1	$f_{CLK}/64$	8 个 f_{AD}	19 个 f_{AD} (采样时钟数: 7 个 f_{AD})	$1728/f_{CLK}$	禁止设定	禁止设定	禁止设定	108 μs	72 μs	
0	0	1	$f_{CLK}/32$	$864/f_{CLK}$		108 μs			54 μs				36 μs		
0	1	0	$f_{CLK}/16$	$432/f_{CLK}$		108 μs			54 μs				27 μs	18 μs	
0	1	1	$f_{CLK}/8$	$216/f_{CLK}$		54 μs			27 μs				13.5 μs	9 μs	
1	0	0	$f_{CLK}/6$	$162/f_{CLK}$		40.5 μs			20.25 μs				10.125 μs	6.75 μs	
1	0	1	$f_{CLK}/5$	$135/f_{CLK}$		135 μs			33.75 μs				16.875 μs	8.4375 μs	5.625 μs
1	1	0	$f_{CLK}/4$	$108/f_{CLK}$		108 μs			27 μs				13.5 μs	6.75 μs	4.5 μs
1	1	1	$f_{CLK}/2$	$54/f_{CLK}$		54 μs			13.5 μs				6.75 μs	3.375 μs	禁止设定
0	0	0	0	1	标准 2	$f_{CLK}/64$	8 个 f_{AD}	17 个 f_{AD} (采样时钟数: 5 个 f_{AD})	$1600/f_{CLK}$	禁止设定	禁止设定	禁止设定	100 μs	66.67 μs	
0	0	1	$f_{CLK}/32$	$800/f_{CLK}$		100 μs			50 μs				33.33 μs		
0	1	0	$f_{CLK}/16$	$400/f_{CLK}$		100 μs			50 μs				25 μs	16.67 μs	
0	1	1	$f_{CLK}/8$	$200/f_{CLK}$		50 μs			25 μs				12.5 μs	8.33 μs	
1	0	0	$f_{CLK}/6$	$150/f_{CLK}$		37.5 μs			18.75 μs				9.375 μs	6.25 μs	
1	0	1	$f_{CLK}/5$	$125/f_{CLK}$		125 μs			31.25 μs				15.625 μs	7.8125 μs	5.21 μs
1	1	0	$f_{CLK}/4$	$100/f_{CLK}$		100 μs			25 μs				12.5 μs	6.25 μs	4.17 μs
1	1	1	$f_{CLK}/2$	$50/f_{CLK}$		50 μs			12.5 μs				6.25 μs	3.125 μs	禁止设定

注 1. 在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中检测到硬件触发后, 不发生转换启动时间和 A/D 电源稳定等待时间 (参照表 10-3(1/4))。

2. 这是 10 位分辨率时的转换时钟数。如果选择 8 位分辨率, 就会减少 2 个转换时钟 (f_{AD})。

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 (t_{CONV}) 范围内。转换时间 (t_{CONV}) 不包含 A/D 电源稳定等待时间。

- 要将 FR2 ~ FR0 位、LV1 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 转换时间不包含转换启动时间, 因此在第 1 次转换时必须加上转换启动时间。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。
- 硬件触发等待模式中的转换时间包含检测到硬件触发后的 A/D 电源稳定等待时间。

备注 f_{CLK} : CPU/ 外围硬件的时钟频率

表 10-3 A/D 转换时间的选择 (4/4)

(4) 有 A/D 电源稳定等待时间 低电压模式 1、2 (硬件触发等待模式注 1)

A/D 转换器的模式寄存器 0 (ADM0)					模式	转换时钟 (f _{AD})	A/D 电源稳定等待时钟数	转换时钟数注 5	A/D 电源稳定等待时间 + 转换时间	A/D 电源稳定等待时间 + 10 位分辨率时的转换时间					
FR2	FR1	FR0	LV1	LV0						1.6V ≤ V _{DD} ≤ 5.5V		注 2	注 3	注 4	
										f _{CLK} =1MHz	f _{CLK} =4MHz	f _{CLK} =8MHz	f _{CLK} =16MHz	f _{CLK} =24MHz	
0	0	0	1	0	低电压 1	f _{CLK} /64	2 个 f _{AD}	19 个 f _{AD} (采样时钟数: 7 个 f _{AD})	1344/f _{CLK}	禁止设定	禁止设定	禁止设定	84μs	56μs	
0	0	1	f _{CLK} /32	672/f _{CLK}		84μs			42μs				28μs		
0	1	0	f _{CLK} /16	336/f _{CLK}		84μs			42μs				21μs	14μs	
0	1	1	f _{CLK} /8	168/f _{CLK}		42μs			21μs				10.5μs	7μs	
1	0	0	f _{CLK} /6	126/f _{CLK}		31.25μs			15.75μs				7.875μs	5.25μs	
1	0	1	f _{CLK} /5	105/f _{CLK}		105μs			26.25μs				13.125μs	6.5625μs	4.38μs
1	1	0	f _{CLK} /4	84/f _{CLK}		84μs			21μs				10.5μs	5.25μs	3.5μs
1	1	1	f _{CLK} /2	42/f _{CLK}		42μs			10.5μs				5.25μs	2.625μs	禁止设定
0	0	0	1	1	低电压 2	f _{CLK} /64	2 个 f _{AD}	17 个 f _{AD} (采样时钟数: 5 个 f _{AD})	1216/f _{CLK}	禁止设定	禁止设定	禁止设定	76μs	50.67μs	
0	0	1	f _{CLK} /32	608/f _{CLK}		76μs			38μs				25.33μs		
0	1	0	f _{CLK} /16	304/f _{CLK}		76μs			38μs				19μs	12.67μs	
0	1	1	f _{CLK} /8	152/f _{CLK}		38μs			19μs				9.5μs	6.33μs	
1	0	0	f _{CLK} /6	114/f _{CLK}		28.5μs			14.25μs				7.125μs	4.75μs	
1	0	1	f _{CLK} /5	95/f _{CLK}		96μs			23.75μs				11.88μs	5.938μs	3.96μs
1	1	0	f _{CLK} /4	76/f _{CLK}		76μs			19μs				9.5μs	4.75μs	3.17μs
1	1	1	f _{CLK} /2	38/f _{CLK}		38μs			9.5μs				4.75μs	2.375μs	禁止设定

注 1. 在连续转换模式的第 2 次以后的转换和扫描模式的扫描 1 以后的转换中检测到硬件触发后, 不发生转换启动时间和 A/D 电源稳定等待时间 (参照表 10-3(2/4))。

2. $1.8V \leq V_{DD} \leq 5.5V$

3. $2.4V \leq V_{DD} \leq 5.5V$

4. $2.7V \leq V_{DD} \leq 5.5V$

5. 这是 10 位分辨率时的转换时钟数。如果选择 8 位分辨率, 就会减少 2 个转换时钟 (f_{AD})。

注意 1. A/D 转换时间必须在“27.6.1 A/D 转换器特性”所示的转换时间 (t_{CONV}) 范围内。转换时间 (t_{CONV}) 不包含 A/D 电源稳定等待时间。

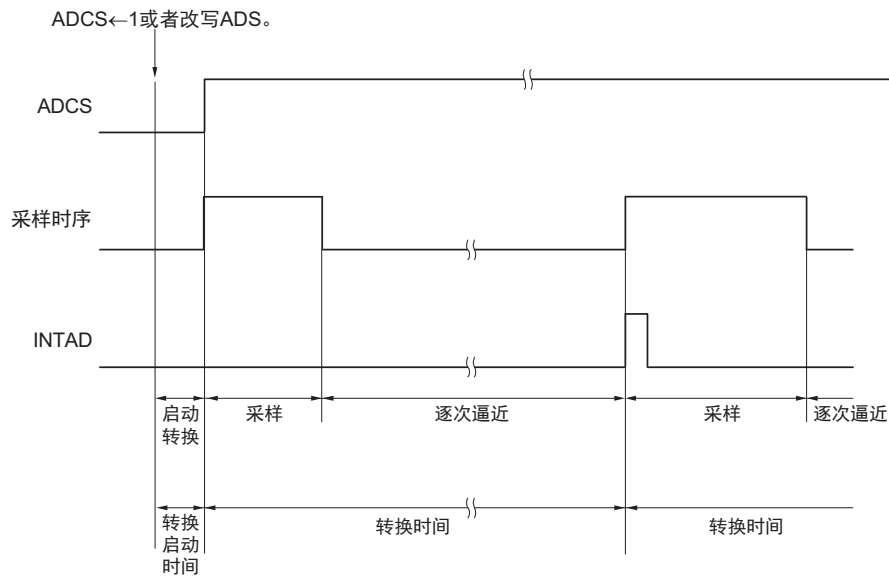
2. 要将 FR2 ~ FR0 位、LV1 位和 LV0 位改写为不同数据时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

3. 转换时间不包含转换启动时间, 因此在第 1 次转换时必须加上转换启动时间。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。

4. 硬件触发等待模式中的转换时间包含检测到硬件触发后的 A/D 电源稳定等待时间。

备注 f_{CLK}: CPU/ 外围硬件的时钟频率

图 10-5 A/D 转换器的采样和 A/D 转换时序（以软件触发模式为例的情况）



10.3.3 A/D 转换器的模式寄存器 1 (ADM1)

这是设定 A/D 转换触发、转换模式和硬件触发信号的寄存器。

通过 1 位或者 8 位存储器操作指令设定 ADM1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-6 A/D 转换器的模式寄存器 1 (ADM1) 的格式

地址: FFF32H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	A/D 转换触发模式的选择
0	×	软件触发模式
1	0	硬件触发无等待模式
1	1	硬件触发等待模式

ADSCM	A/D 转换模式的设定
0	连续转换模式
1	单次转换模式

ADTRS1	ADTRS0	硬件触发信号的选择
0	0	定时器通道 1 的计数结束或者捕捉结束中断信号 (INTTM01)
0	1	禁止设定。
1	0	禁止设定。
1	1	12 位间隔定时器中断信号 (INTIT)

注意 1. 要改写 ADM1 寄存器时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

2. 为了结束 A/D 转换，必须至少将硬件触发间隔设定为以下时间：

硬件触发无等待模式时：2 个 f_{CLK} 时钟 + 转换启动时间 + A/D 转换时间

硬件触发等待模式时：2 个 f_{CLK} 时钟 + 转换启动时间 + A/D 电源稳定等待时间 + A/D 转换时间

3. 在 SNOOZE 功能以外的模式中输入 INTIT 后的最多 4 个 f_{CLK} 时钟周期内，下一个 INTIT 的输入触发无效。

备注 1. ×：忽略

2. f_{CLK} ：CPU/ 外围硬件的时钟频率

10.3.4 A/D 转换器的模式寄存器 2 (ADM2)

这是选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压和分辨率并且检查 A/D 转换结果的上限值和下限值以及设定 SNOOZE 模式的寄存器。

通过 1 位或者 8 位存储器操作指令设定 ADM2 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-7 A/D 转换器的模式寄存器 2 (ADM2) 的格式 (1/2)

地址: F0010H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	A/D 转换器的正 (+) 基准电压源的选择
0	0	由 V_{DD} 提供。
0	1	由 P20/ AV_{REFP} /ANI0 提供。
1	0	由内部基准电压 (1.45V) 提供注。
1	1	禁止设定。

• 要改写 ADREFP1 位和 ADREFP0 位时，必须按照以下步骤进行设定：

- ①将 ADCE 位置“0”。
- ②更改 ADREFP1 位和 ADREFP0 位的值。
- ③等待基准电压的稳定等待时间 A。
- ④将 ADCE 位置“1”。
- ⑤等待基准电压的稳定等待时间 B。

当将 ADREFP1 位和 ADREFP0 位改为“1”和“0”时，A=5 μ s，B=1 μ s。
 当将 ADREFP1 位和 ADREFP0 位改为“0”和“0”或者“0”和“1”时，A 不需等待，B=1 μ s。

必须在等待⑤的稳定时间后开始 A/D 转换。

ADREFM	A/D 转换器的负 (-) 基准电压源的选择
0	由 V_{SS} 提供。
1	由 P21/ AV_{REFM} /ANI1 提供。

注 只有在 HS (高速主) 模式中才能设定。

注意 1. 要改写 ADM2 寄存器时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

2. 要转移到 STOP 模式时，不能将 ADREFP1 位置“1”。如果选择内部基准电压 (ADREFP1、ADREFP0=1、0)，就加上“27.3.2 电源电流特性”所示的 A/D 转换器基准电压电流 (I_{ADREF}) 的电流值。

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

图 10-7 A/D 转换器的模式寄存器 2 (ADM2) 的格式 (2/2)

地址: F0010H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	转换结果上限值和下限值的检查
0	当 ADLL 寄存器 \leq ADCR 寄存器 \leq ADUL 寄存器 (AREA1) 时, 产生中断信号 (INTAD)。
1	当 ADCR 寄存器 $<$ ADLL 寄存器 (AREA2) 或者 ADUL 寄存器 $<$ ADCR 寄存器 (AREA3) 时, 产生中断信号 (INTAD)。
AREA1 ~ AREA3 的中断信号 (INTAD) 的产生范围如图 10-8 所示。	

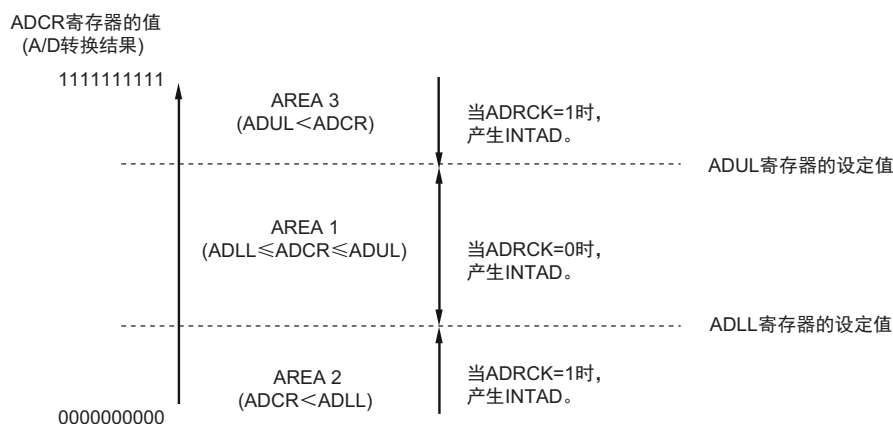
AWC	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
<p>通过 STOP 模式中的硬件触发信号来解除 STOP 模式, 并且在 CPU 不运行的状态下进行 A/D 转换 (SNOOZE 模式)。</p> <ul style="list-style-type: none"> • 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟 (f_{CLK}) 的情况下才能设定 SNOOZE 模式功能, 而在选择其他时钟的情况下禁止设定。 • 在软件触发模式或者硬件触发无等待模式中, 禁止使用 SNOOZE 模式功能。 • 在连续转换模式中, 禁止使用 SNOOZE 模式功能。 • 在使用 SNOOZE 模式功能时, 必须至少将硬件触发间隔设定为 “SNOOZE 模式的转移时间注 + 转换启动时间 + A/D 电源稳定等待时间 + A/D 转换时间 + 2 个 f_{CLK} 时钟”。 • 即使使用 SNOOZE 模式功能, 也必须在通常运行时将 AWC 位置 “0”, 并且在即将要转移到 STOP 模式前将 AWC 位改为 “1”。 <p>另外, 必须在从 STOP 模式返回到通常运行后将 AWC 位改为 “0”。</p> <p>如果 AWC 位保持 “1” 的状态, 就无法正常开始 A/D 转换, 而与其后的 SNOOZE 模式和通常运行模式无关。</p>	

ADTYP	A/D 转换分辨率的选择
0	10 位分辨率
1	8 位分辨率

注 请参照 “16.3.3 SNOOZE 模式” 的 “STOP 模式 \rightarrow SNOOZE 模式的转移时间”。

注意 要改写 ADM2 寄存器时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

图 10-8 ADRCK 位的中断信号产生范围



备注 当不发生 INTAD 时, A/D 转换结果不保存到 ADCR 寄存器和 ADCRH 寄存器。

10.3.5 10 位 A/D 转换结果寄存器 (ADCR)

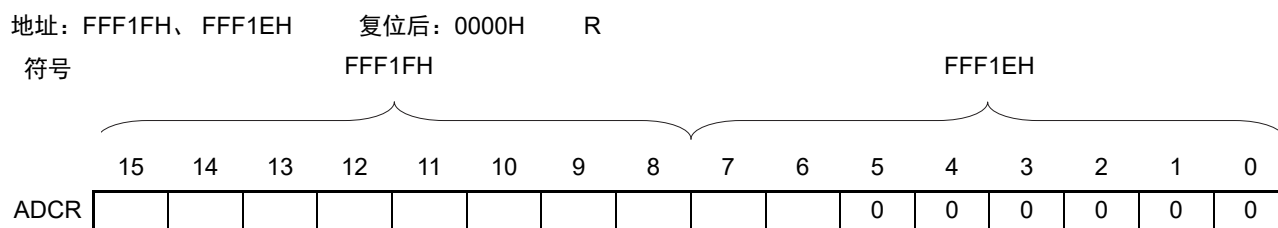
这是保存 A/D 转换结果的 16 位寄存器，低 6 位固定为“0”。每当 A/D 转换结束时，就从逐次逼近寄存器 (SAR) 装入转换结果。转换结果的高 8 位保存到 FFF1FH，低 2 位保存到 FFF1EH 的高 2 位注。

通过 16 位存储器操作指令读 ADCR 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

注 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 10-8））的设定值范围内，就不保存 A/D 转换结果。

图 10-9 10 位 A/D 转换结果寄存器 (ADCR) 的格式



- 注意 1. 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 ADCR 寄存器，低 2 位 (ADCR 寄存器的 bit7 和 bit6) 的读取值就为“0”。
2. 当对 ADCR 寄存器进行 16 位存取时，能从 ADCR 寄存器的 bit15 依次读转换结果的高 10 位。

10.3.6 8 位 A/D 转换结果寄存器 (ADCRH)

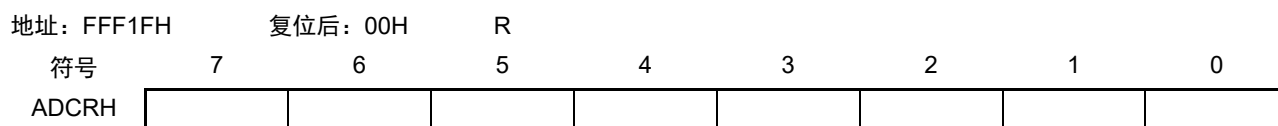
这是保存 A/D 转换结果的 8 位寄存器，保存 10 位分辨率的高 8 位注。

通过 8 位存储器操作指令读 ADCRH 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 10-8））的设定值范围内，就不保存 A/D 转换结果。

图 10-10 8 位 A/D 转换结果寄存器 (ADCRH) 的格式



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

10.3.7 模拟输入通道指定寄存器 (ADS)

这是指定要进行 A/D 转换的模拟电压输入通道的寄存器。

通过 1 位或者 8 位存储器操作指令设定 ADS 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-11 模拟输入通道指定寄存器 (ADS) 的格式

地址: FFF31H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○选择模式 (ADMD=0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	0	0	ANI0	P20/ANI0/AV _{REFP} 引脚
0	0	0	0	0	1	ANI1	P21/ANI1/AV _{REFM} 引脚
0	0	0	0	1	0	ANI2	P22/ANI2 引脚
0	0	0	0	1	1	ANI3	P23/ANI3 引脚
0	1	0	0	0	0	ANI16	P01/ANI16 引脚
0	1	0	0	0	1	ANI17	P00/ANI17 引脚
0	1	0	0	1	0	ANI18	P147/ANI18 引脚
0	1	0	0	1	1	ANI19	P120/ANI19 引脚
1	0	0	0	0	1	—	内部基准电压 (1.45V) 注
上述以外						禁止设定。	

注 只有在 HS (高速主) 模式中才能选择。

○扫描模式 (ADMD=1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	模拟输入通道			
						扫描 0	扫描 1	扫描 2	扫描 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	—
0	0	0	0	1	0	ANI2	ANI3	—	—
0	0	0	0	1	1	ANI3	—	—	—
上述以外						禁止设定。			

注意 1. 必须将 bit5 和 bit6 置“0”。

- 对于由 ADPC 寄存器和 PMCxx 寄存器设定为模拟输入的端口，必须通过端口模式寄存器 0、2、12、14 (PM0、PM2、PM12、PM14) 将其设定为输入模式。
- 对于由 A/D 端口配置寄存器 (ADPC) 设定为数字输入 / 输出的引脚，不能通过 ADS 寄存器进行设定。
- 对于由端口模式控制寄存器 0、12、14 (PMC0、PMC12、PMC14) 设定为数字输入 / 输出的引脚，不能通过 ADS 寄存器进行设定。
- 要改写 ADISS 位时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 当将 AV_{REFP} 用作 A/D 转换器的正 (+) 基准电压时，不能选择 ANI0 作为 A/D 转换通道。
- 当将 AV_{REFM} 用作 A/D 转换器的负 (-) 基准电压时，不能选择 ANI1 作为 A/D 转换通道。
- 在将 ADISS 位置“1”的情况下，不能将内部基准电压 (1.45V) 用作正 (+) 基准电压。另外，在将 ADISS 位置“1”后，不能使用第 1 次的转换结果。有关详细设定流程，请参照“10.7.4 选择内部基准电压时的设定”。
- 要转移到 STOP 模式时，不能将 ADISS 位置“1”。如果将 ADISS 位置“1”，就加上“27.3.2 电源电流特性”所示的 A/D 转换器基准电压电流 (I_{ADREF}) 的电流值。

10.3.8 转换结果比较上限值设定寄存器 (ADUL)

这是用于检查 A/D 转换结果上限值的设定寄存器。

将 A/D 转换结果和 ADUL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2 (ADM2) 的 ADRCK 位的设定范围内 (参照图 10-8) 控制中断信号 (INTAD) 的产生。

通过 8 位存储器操作指令设定 ADUL 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

图 10-12 转换结果比较上限值设定寄存器 (ADUL) 的格式

地址: F0011H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

10.3.9 转换结果比较下限值设定寄存器 (ADLL)

这是用于检查 A/D 转换结果下限值的设定寄存器。

将 A/D 转换结果和 ADLL 寄存器的值进行比较，并且在 A/D 转换器的模式寄存器 2 (ADM2) 的 ADRCK 位的设定范围内 (参照图 10-8) 控制中断信号 (INTAD) 的产生。

通过 8 位存储器操作指令设定 ADLL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 10-13 转换结果比较下限值设定寄存器 (ADLL) 的格式

地址: F0012H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

注意 1. 当选择 10 位分辨率 A/D 转换时，将 10 位 A/D 转换结果寄存器 (ADCR) 的高 8 位和 ADUL 寄存器以及 ADLL 寄存器进行比较。

2. 要改写 ADUL 寄存器和 ADLL 寄存器时，必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。

3. 在设定 ADUL 寄存器和 ADLL 寄存器时，必须使 ADUL > ADLL。

10.3.10 A/D 测试寄存器 (ADTES)

此寄存器选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (AN_{Ixx}) 和内部基准电压 (1.45V) 作为 A/D 转换对象。

当用作 A/D 测试功能时, 进行以下的设定:

- 在测量零刻度时, 选择负 (-) 基准电压作为 A/D 转换对象。
- 在测量满刻度时, 选择正 (+) 基准电压作为 A/D 转换对象。

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 10-14 A/D 测试寄存器 (ADTES) 的格式

地址: F0013H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
ADTES	0	0	0	0	0	0	ADTES1	ADTES0	

ADTES1	ADTES0	A/D 转换对象
0	0	内部基准电压 (1.45V) 注 (由模拟输入通道指定寄存器 (ADS) 进行设定)
1	0	负 (-) 基准电压 (通过 ADM2 寄存器的 ADREFM 位进行选择)
1	1	正 (+) 基准电压 (通过 ADM2 寄存器的 ADREFP1 位和 ADREFP0 位进行选择)
上述以外		禁止设定。

注 只有在 HS (高速主) 模式中才能选择内部基准电压 (1.45V)。

10.3.11 控制模拟输入引脚端口功能的寄存器

必须设定与 A/D 转换器模拟输入复用的端口功能的控制寄存器 (端口模式寄存器 (PM_{xx})、端口模式控制寄存器 (PMC_{xx}) 和 A/D 端口配置寄存器 (ADPC))。详细内容请参照“4.3.1 端口模式寄存器 (PM_{xx})”、“4.3.6 端口模式控制寄存器 (PMC_{xx})”和“4.3.7 A/D 端口配置寄存器 (ADPC)”。

在将 ANI0 ~ ANI3 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 (PM_{xx}) 的位置“1”, 并且通过 A/D 端口配置寄存器 (ADPC) 设定为模拟输入。

在将 ANI16 ~ ANI19 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 (PM_{xx}) 和端口模式控制寄存器 (PMC_{xx}) 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。

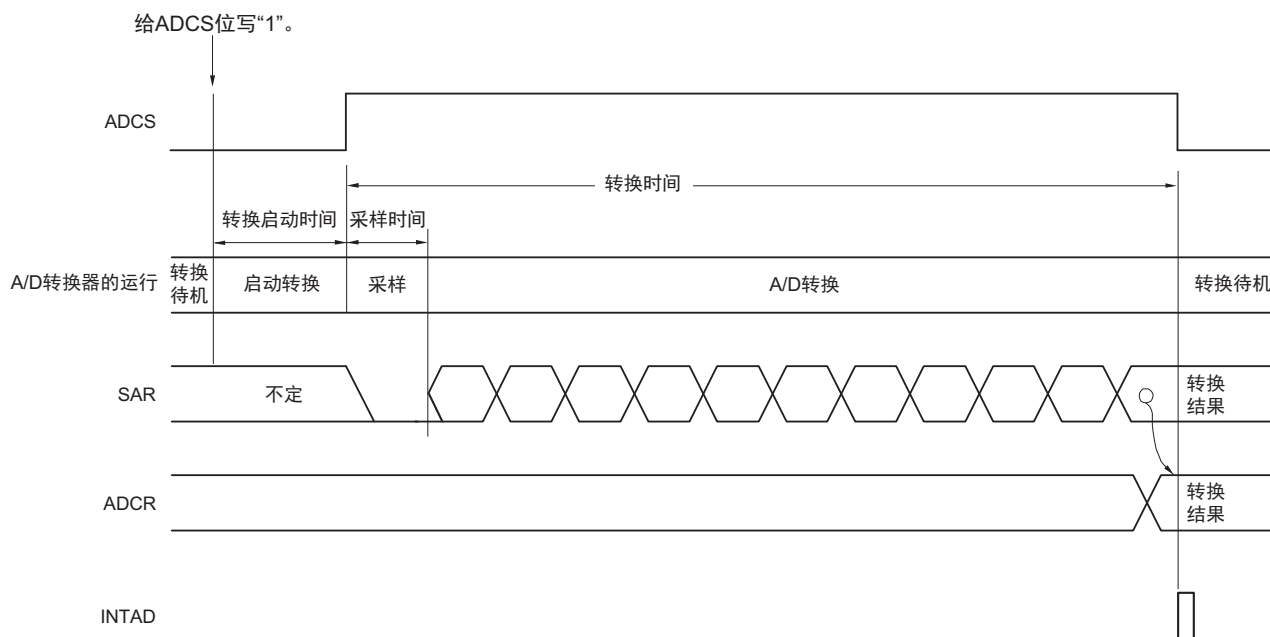
- ① 通过采样 & 保持电路对所选模拟输入通道的输入电压进行采样。
- ② 在进行规定时间的采样后，采样 & 保持电路为保持状态，并且将采样电压保持到 A/D 转换结束为止。
- ③ 将逐次逼近寄存器（SAR）的 bit9 置位，并且通过分接头选择器将串联电阻串的分接头电压设定为 $(1/2)AV_{REF}$ 。
- ④ 通过 A/D 电压比较器将串联电阻串的分接头电压和采样电压的电压差进行比较。如果模拟输入大于 $(1/2)AV_{REF}$ ，SAR 寄存器的 MSB 位就保持被置位的状态。如果模拟输入小于 $(1/2)AV_{REF}$ ，就对 MSB 位进行复位。
- ⑤ 然后，自动将 SAR 寄存器的 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=1: $(3/4)AV_{REF}$
 - bit9=0: $(1/4)AV_{REF}$
 将此分接头电压和采样电压进行比较，根据比较结果对 SAR 寄存器的 bit8 进行以下操作：
 - 采样电压 \geq 分接头电压: bit8=1
 - 采样电压 $<$ 分接头电压: bit8=0
- ⑥ 将此比较继续进行到 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，并且将其值传送到 A/D 转换结果寄存器（ADCR、ADCRH），而且被锁存注 1。
同时，能产生 A/D 转换结束中断请求（INTAD）注 1。
- ⑧ 重复①~⑦的运行，直到 ADCS 位变为“0”为止注 2。
要停止 A/D 转换器时，必须将 ADCS 位置“0”。

- 注 1. 如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定（参照图 10-8））的设定值范围内，就不产生 A/D 转换结束中断请求信号（INTAD），并且 ADCR 寄存器和 ADCRH 寄存器不保存 A/D 转换结果。
2. 在连续转换模式中，ADCS 标志不自动清“0”。即使在硬件触发无等待模式的单次转换模式中，ADCS 标志也不自动清“0”而保持“1”的状态。

备注 1. 有 2 种 A/D 转换结果寄存器。

- ADCR 寄存器（16 位）：保存 10 位的 A/D 转换值。
 - ADCRH 寄存器（8 位）：保存 8 位的 A/D 转换值。
2. AV_{REF} 是 A/D 转换器的正（+）基准电压，可选择 AV_{REFP} 、内部基准电压（1.45V）或者 V_{DD} 。

图 10-15 A/D 转换器的转换运行（软件触发模式的情况）



如果在单次转换模式中进行 A/D 转换，ADCS 位就在 A/D 转换结束后自动清“0”。

在连续转换模式中，A/D 转换连续进行到通过软件将 A/D 转换器的模式寄存器 0 (ADM0) 的 bit7 (ADCS) 清“0”为止。

如果在 A/D 转换过程中改写或者重写模拟输入通道指定寄存器 (ADS)，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。

在产生复位信号后，A/D 转换结果寄存器 (ADCR、ADCRH) 的值变为“0000H”和“00H”。

10.5 输入电压和转换结果

模拟输入引脚（ANI0 ~ ANI3、ANI16 ~ ANI19）的模拟输入电压和理论上的 A/D 转换结果（10 位 A/D 转换结果寄存器（ADCR））有以下表达式的关系。

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

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

或者

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

INT(): 将括号中的数值的整数部分返回的函数

V_{AIN} : 模拟输入电压

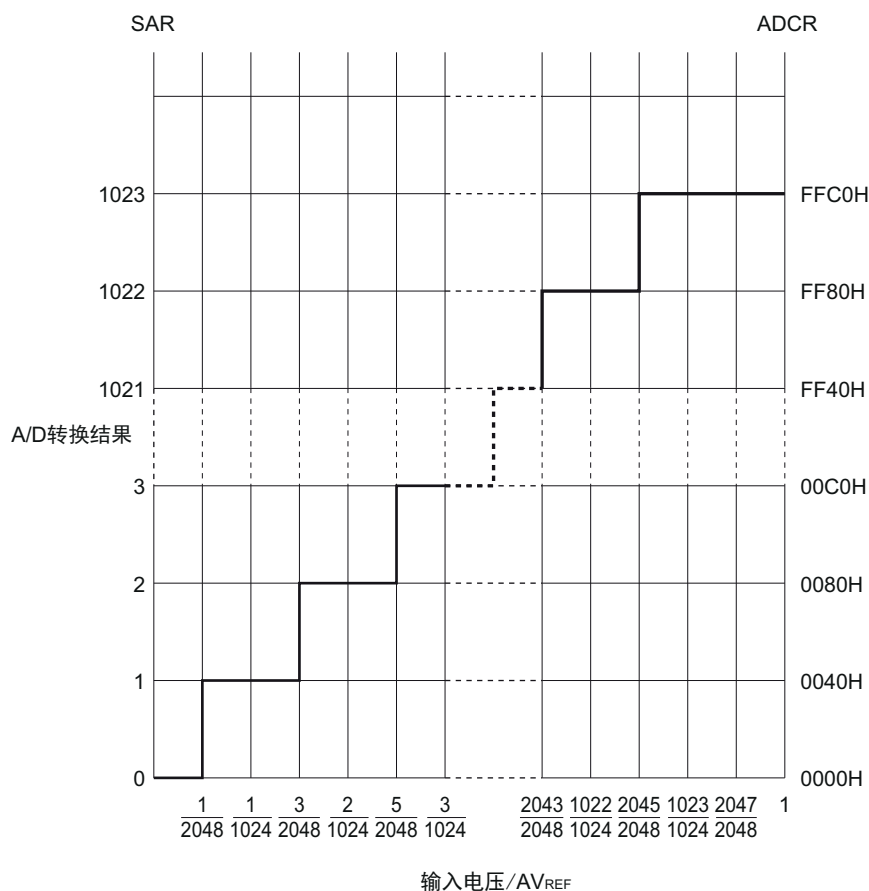
V_{REF} : V_{REF} 引脚电压

ADCR: A/D 转换结果寄存器（ADCR）的值

SAR: 逐次逼近寄存器

模拟输入电压和 A/D 转换结果的关系如图 10-16 所示。

图 10-16 模拟输入电压和 A/D 转换结果的关系



备注 V_{REF} 是 A/D 转换器的正 (+) 基准电压，可选择 V_{REFP} 、内部基准电压（1.45V）或者 V_{DD} 。

10.6 A/D 转换器的运行模式

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

10.6.1 软件触发模式（选择模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换。
- ④ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑥ 即使在转换过程中输入硬件触发也不开始 A/D 转换。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

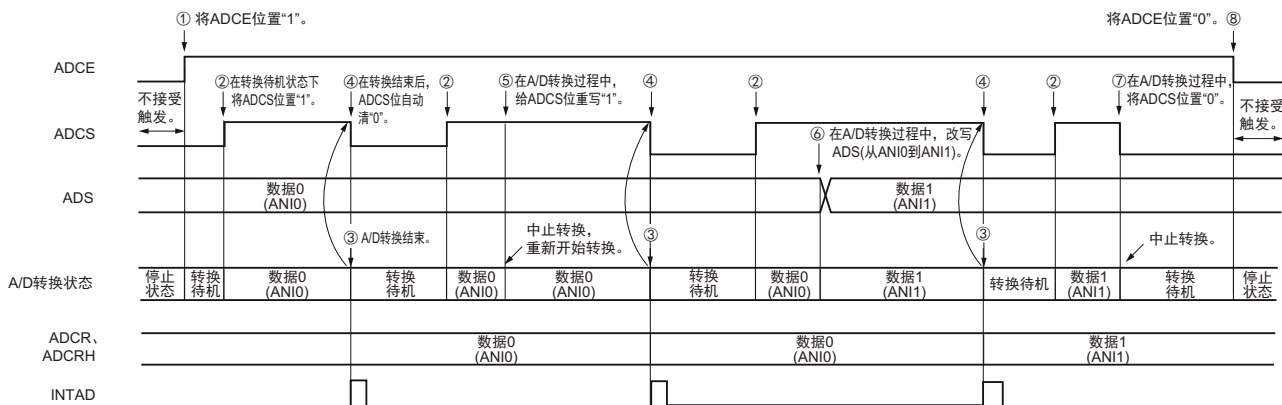
图 10-17 软件触发模式（选择模式、连续转换模式）的运行时序例子



10.6.2 软件触发模式（选择模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ④ 在 A/D 转换结束后，ADCS 位自动清“0”，进入 A/D 转换待机状态。
- ⑤ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。即使在 A/D 转换待机的状态下输入硬件触发也不开始 A/D 转换。

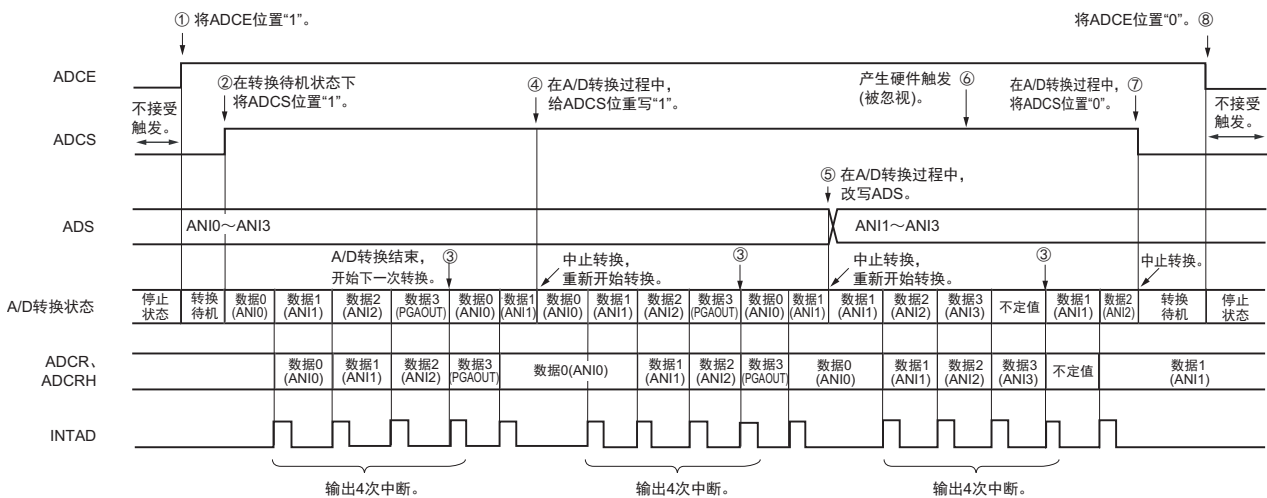
图 10-18 软件触发模式（选择模式、单次转换模式）的运行时序例子



10.6.3 软件触发模式（扫描模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1μs）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描 0~扫描 3 的 4 个模拟输入通道进行 A/D 转换。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ③ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 4 个通道的 A/D 转换结束后立即从所设通道自动开始下一次 A/D 转换（4 个通道）。
- ④ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑥ 即使在转换过程中输入硬件触发也不开始 A/D 转换。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

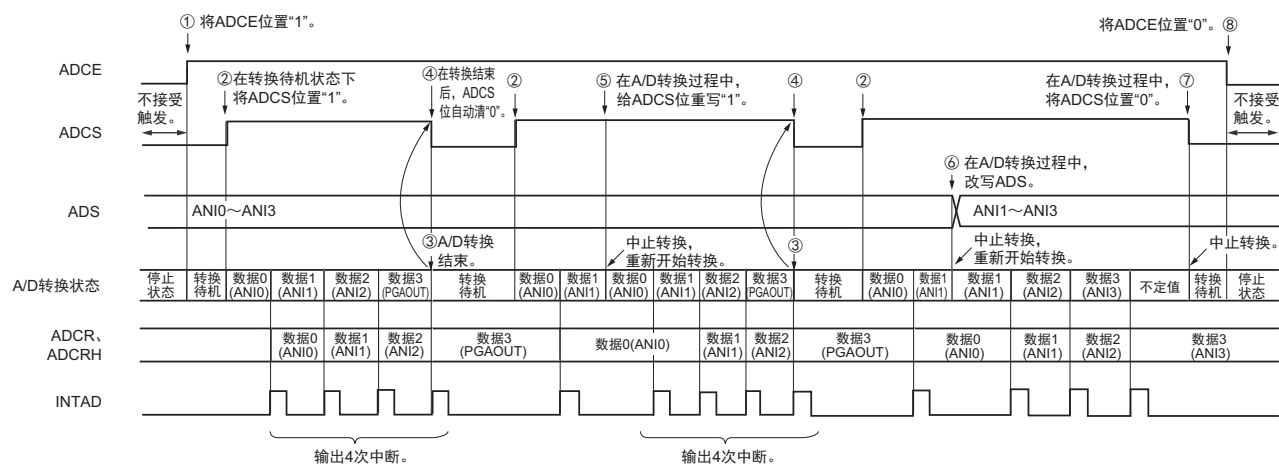
图 10-19 软件触发模式（扫描模式、连续转换模式）的运行时序例子



10.6.4 软件触发模式（扫描模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，对由模拟输入通道指定寄存器（ADS）指定的扫描 0~扫描 3 的 4 个模拟输入通道进行 A/D 转换。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ③ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ④ 在 4 个通道的 A/D 转换结束后，ADCS 位自动清“0”，进入 A/D 转换待机状态。
- ⑤ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。
- ⑧ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。即使在 A/D 转换待机的状态下输入硬件触发也不开始 A/D 转换。

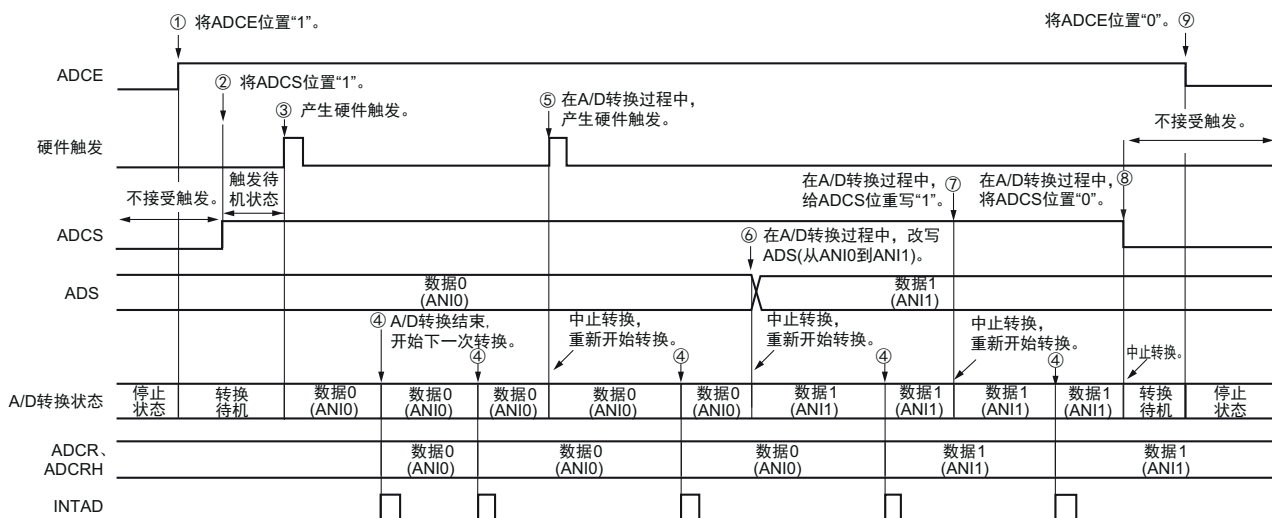
图 10-20 软件触发模式（扫描模式、单次转换模式）的运行时序例子



10.6.5 硬件触发无等待模式（选择模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。
- ③ 如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ④ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。
- ⑨ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCS 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

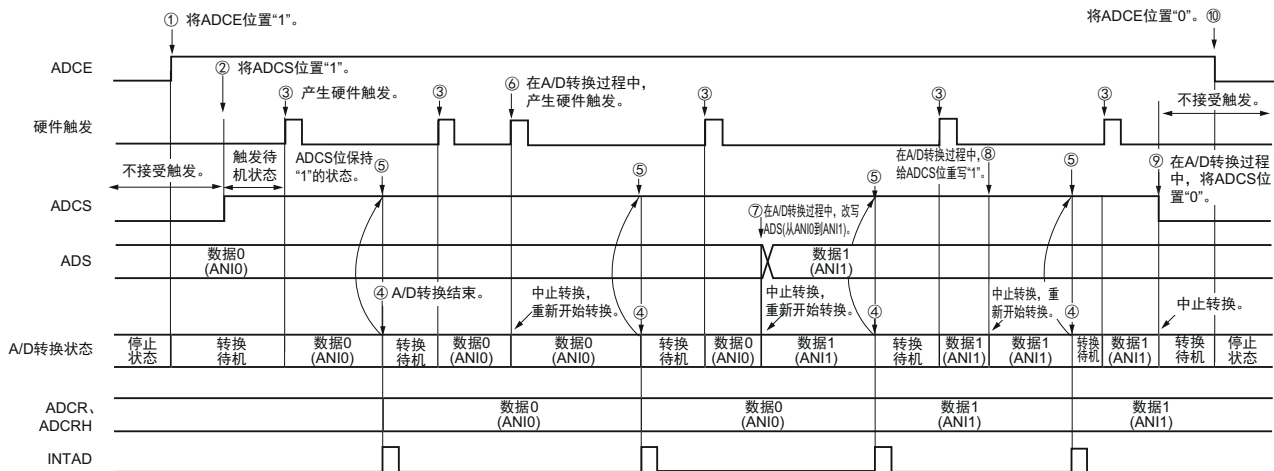
图 10-21 硬件触发无等待模式（选择模式、连续转换模式）的运行时序例子



10.6.6 硬件触发无等待模式（选择模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1 μ s）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。
- ③ 如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。
- ④ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ⑤ 在 A/D 转换结束后，ADCS 位保持“1”的状态，进入 A/D 转换待机状态。
- ⑥ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑧ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑨ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。
- ⑩ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCS 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

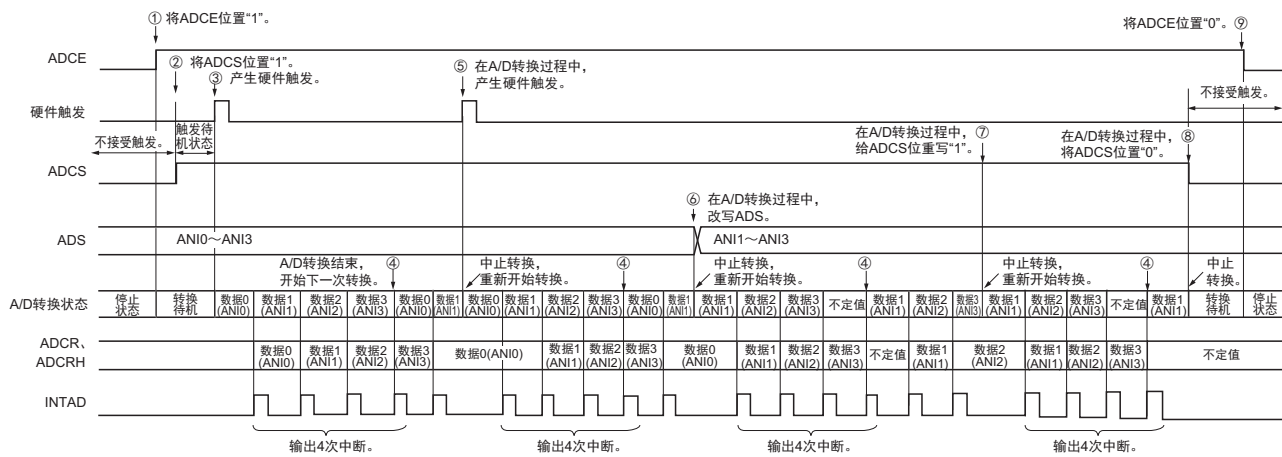
图 10-22 硬件触发无等待模式（选择模式、单次转换模式）的运行时序例子



10.6.7 硬件触发无等待模式（扫描模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1μs）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。
- ③ 如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0～扫描 3 的 4 个模拟输入通道进行 A/D 转换。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ④ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 4 个通道的 A/D 转换结束后立即从所设通道自动开始下一次 A/D 转换。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。
- ⑨ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCE 位为“0”时，即使将 ADCS 位置“1”也被忽视，不开始 A/D 转换。

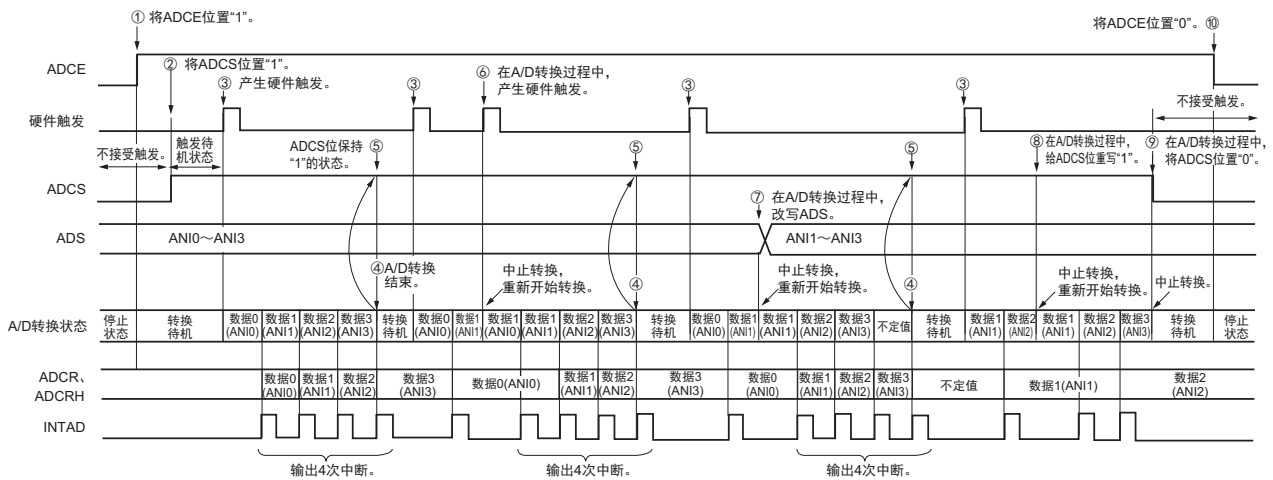
图 10-23 硬件触发无等待模式（扫描模式、连续转换模式）的运行时序例子



10.6.8 硬件触发无等待模式（扫描模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入 A/D 转换待机状态。
- ② 在通过软件对稳定等待时间（1μs）进行计数后，将 ADM0 寄存器的 ADCS 位置“1”，进入硬件触发待机状态（此阶段不开始转换）。在硬件触发待机状态时，即使将 ADCS 位置“1”也不开始 A/D 转换。
- ③ 如果在 ADCS 位为“1”的状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0～扫描 3 的 4 个模拟输入通道进行 A/D 转换。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ④ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ⑤ 在 4 个通道的 A/D 转换结束后，ADCS 位保持“1”的状态，进入 A/D 转换待机状态。
- ⑥ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑧ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑨ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入 A/D 转换待机状态。但是，在此状态下 A/D 转换器不进入停止状态。
- ⑩ 如果在 A/D 转换待机状态下将 ADCE 位置“0”，A/D 转换器就进入停止状态。当 ADCS 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

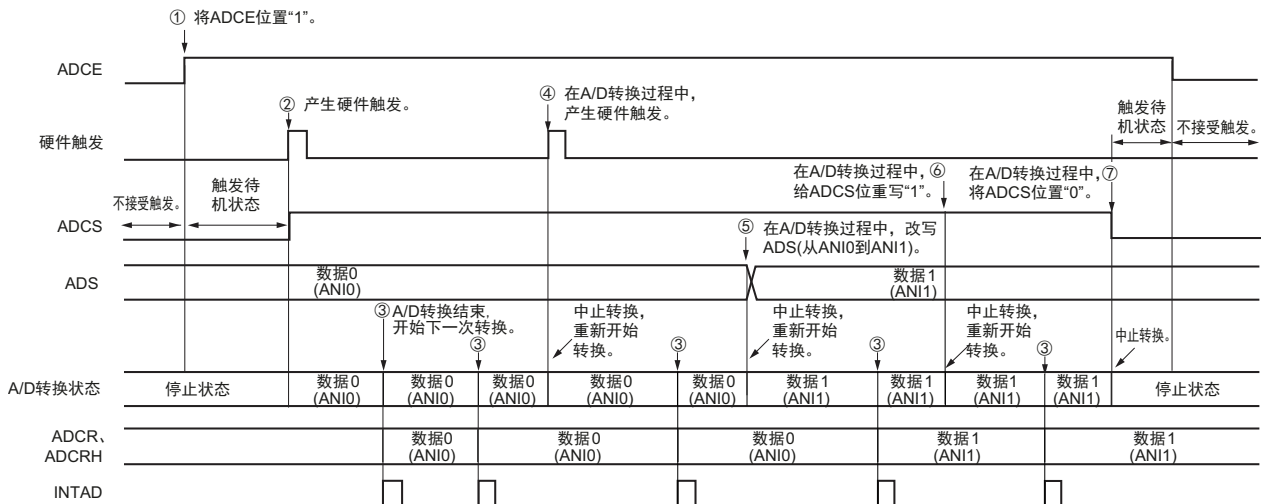
图 10-24 硬件触发无等待模式（扫描模式、单次转换模式）的运行时序例子



10.6.9 硬件触发等待模式（选择模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 A/D 转换结束后立即开始下一次 A/D 转换（此时，不需要硬件触发）。
- ④ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑥ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

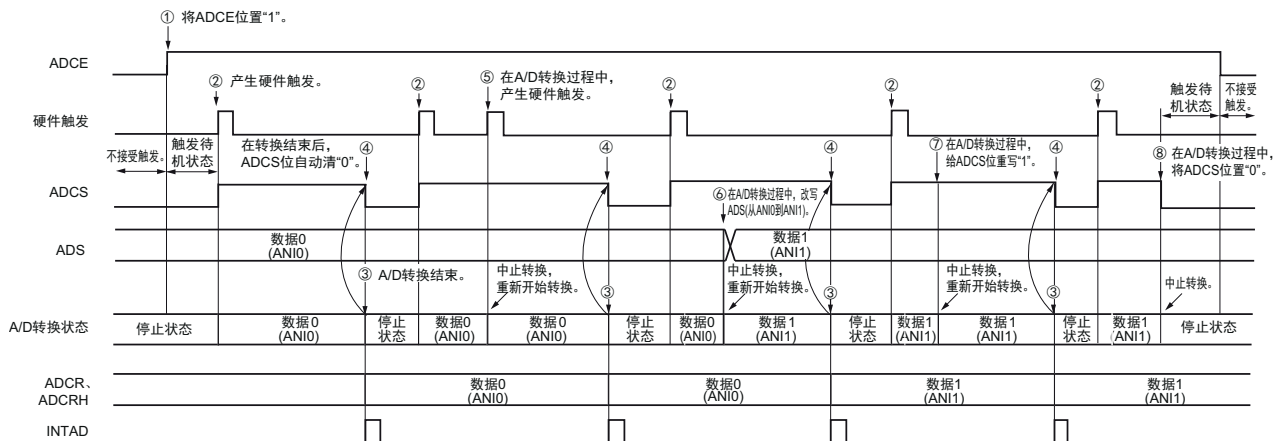
图 10-25 硬件触发等待模式（选择模式、连续转换模式）的运行时序例子



10.6.10 硬件触发等待模式（选择模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的模拟输入进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。
- ③ 如果 A/D 转换结束，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ④ 在 A/D 转换结束后，ADCS 位自动清“0”，A/D 转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而对由 ADS 寄存器重新指定的模拟输入进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

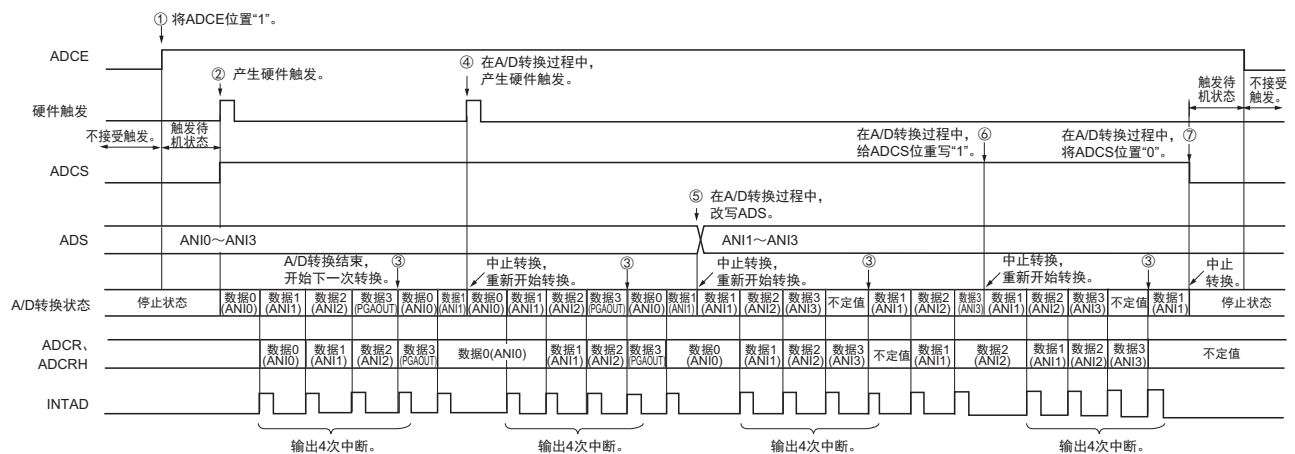
图 10-26 硬件触发等待模式（选择模式、单次转换模式）的运行时序例子



10.6.11 硬件触发等待模式（扫描模式、连续转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0～扫描 3 的 4 个模拟输入通道进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ③ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。在 4 个通道的 A/D 转换结束后立即从所设通道自动开始下一次 A/D 转换。
- ④ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑤ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑥ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑦ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

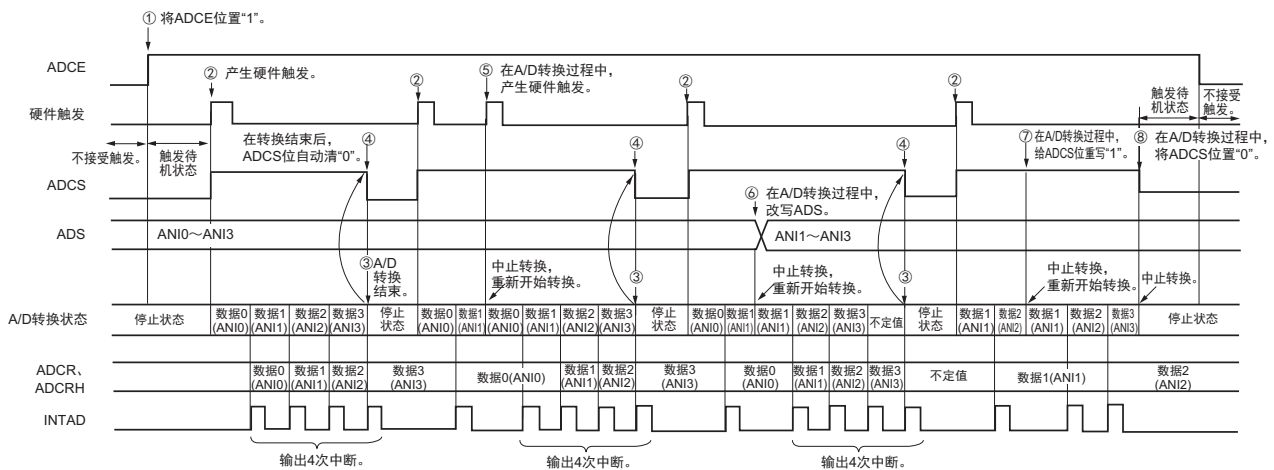
图 10-27 硬件触发等待模式（扫描模式、连续转换模式）的运行时序例子



10.6.12 硬件触发等待模式（扫描模式、单次转换模式）

- ① 在停止状态下，将 A/D 转换器的模式寄存器 0（ADM0）的 ADCE 位置“1”，进入硬件触发待机状态。
- ② 如果在硬件触发待机状态下输入硬件触发，就对由模拟输入通道指定寄存器（ADS）指定的扫描 0～扫描 3 的 4 个模拟输入通道进行 A/D 转换。在输入硬件触发的同时自动将 ADM0 寄存器的 ADCS 位置“1”。依次从扫描 0 指定的模拟输入通道进行 A/D 转换。
- ③ 连续进行 4 个模拟输入通道的 A/D 转换。每当 A/D 转换结束时，就将转换结果保存到 A/D 转换结果寄存器（ADCR、ADCRH），并且产生 A/D 转换结束中断请求信号（INTAD）。
- ④ 在 A/D 转换结束后，ADCS 位自动清“0”，A/D 转换器进入停止状态。
- ⑤ 如果在转换过程中输入硬件触发，就中止当前的 A/D 转换而从最初的通道重新开始转换。取消转换途中的数据。
- ⑥ 如果在转换过程中改写或者重写 ADS 寄存器，就中止当前的 A/D 转换而从由 ADS 寄存器重新指定的最初通道进行 A/D 转换。取消转换途中的数据。
- ⑦ 如果在转换过程中给 ADCS 位重写“1”，就中止当前的 A/D 转换而重新开始转换。取消转换途中的数据。
- ⑧ 如果在转换过程中将 ADCS 位置“0”，就中止当前的 A/D 转换而进入硬件触发待机状态，并且 A/D 转换器进入停止状态。当 ADCE 位为“0”时，即使输入硬件触发也被忽视，不开始 A/D 转换。

图 10-28 硬件触发等待模式（扫描模式、单次转换模式）的运行时序例子

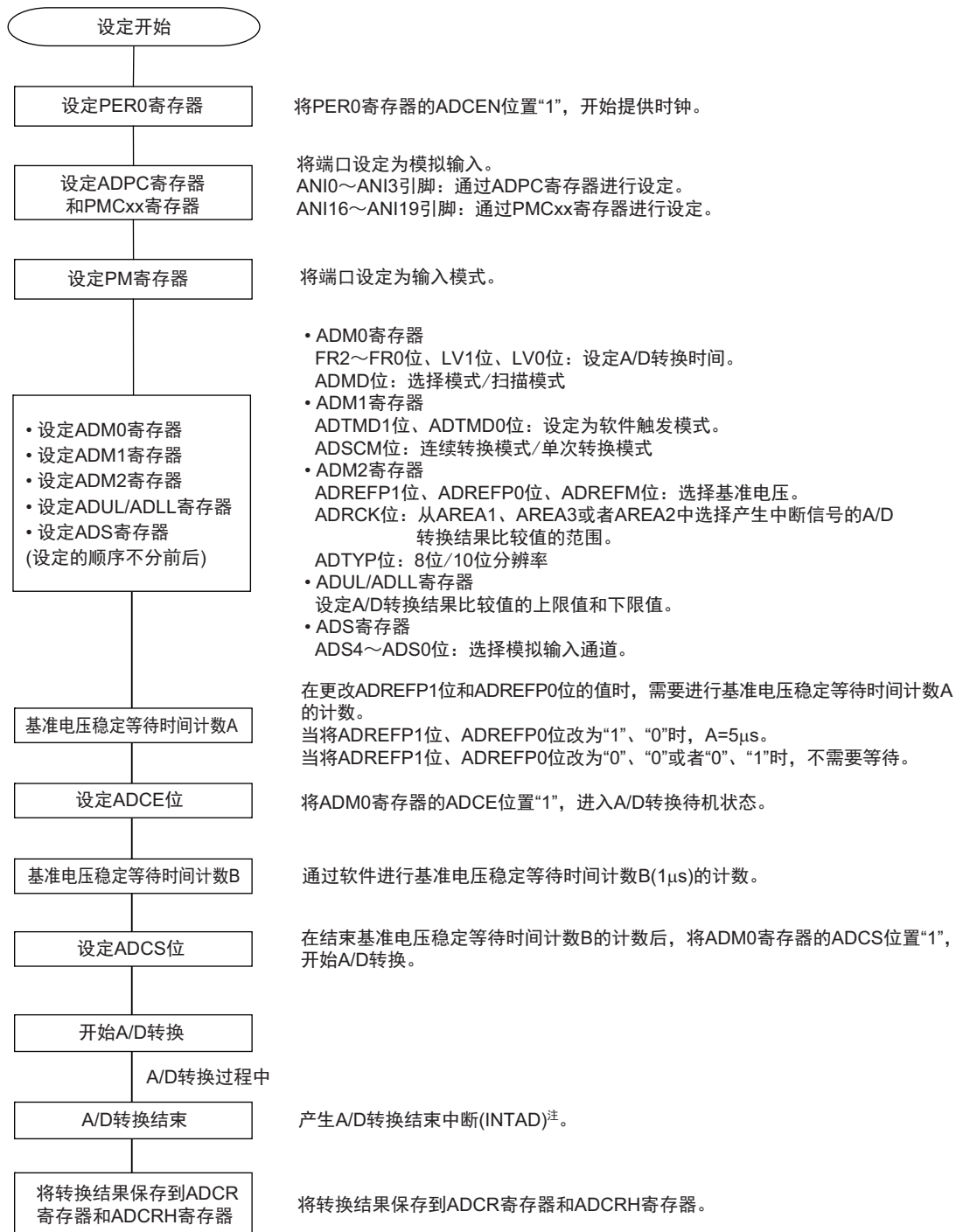


10.7 A/D 转换器的设定流程图

各运行模式的 A/D 转换器的设定流程图如下所示。

10.7.1 软件触发模式的设定

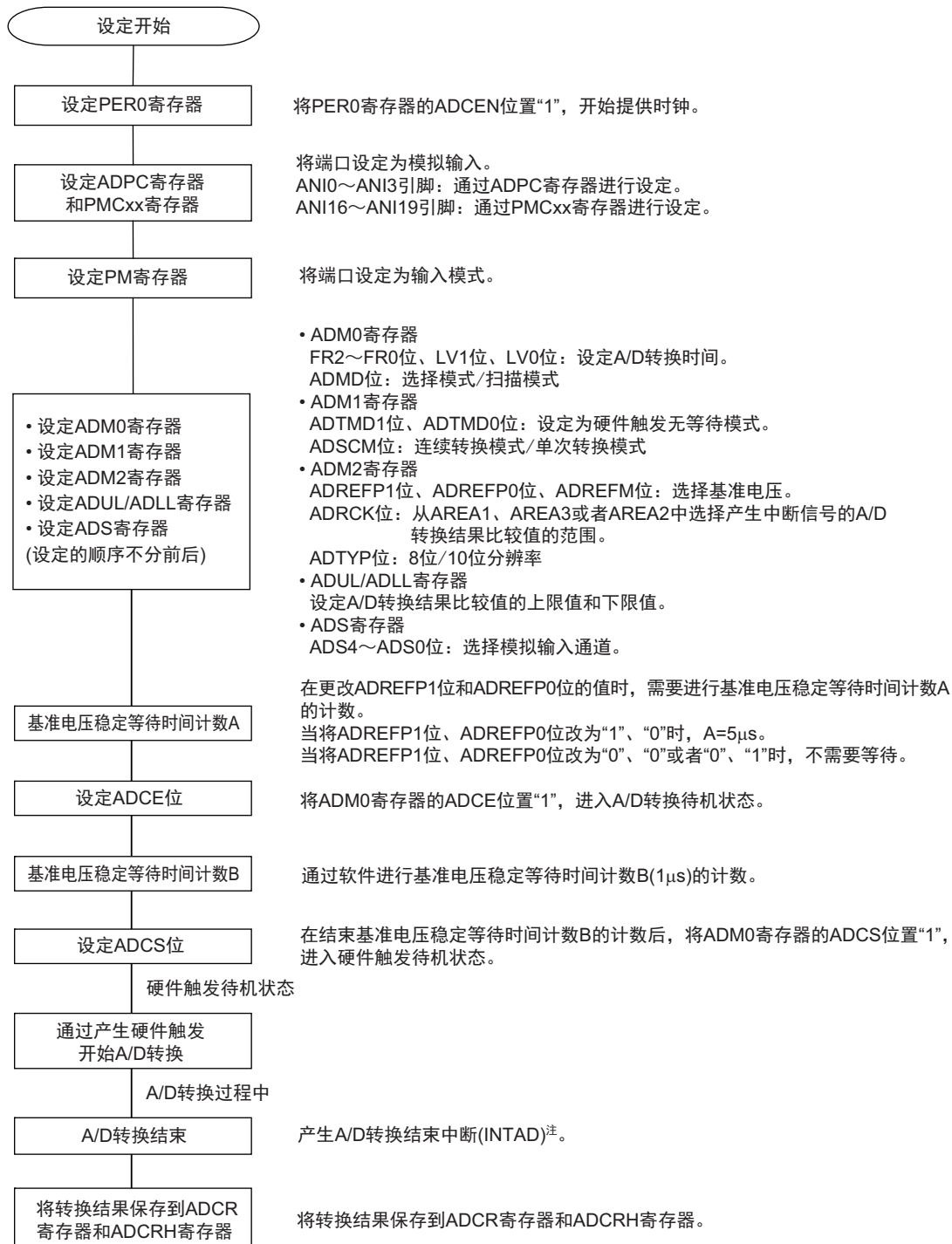
图 10-29 软件触发模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

10.7.2 硬件触发无等待模式的设定

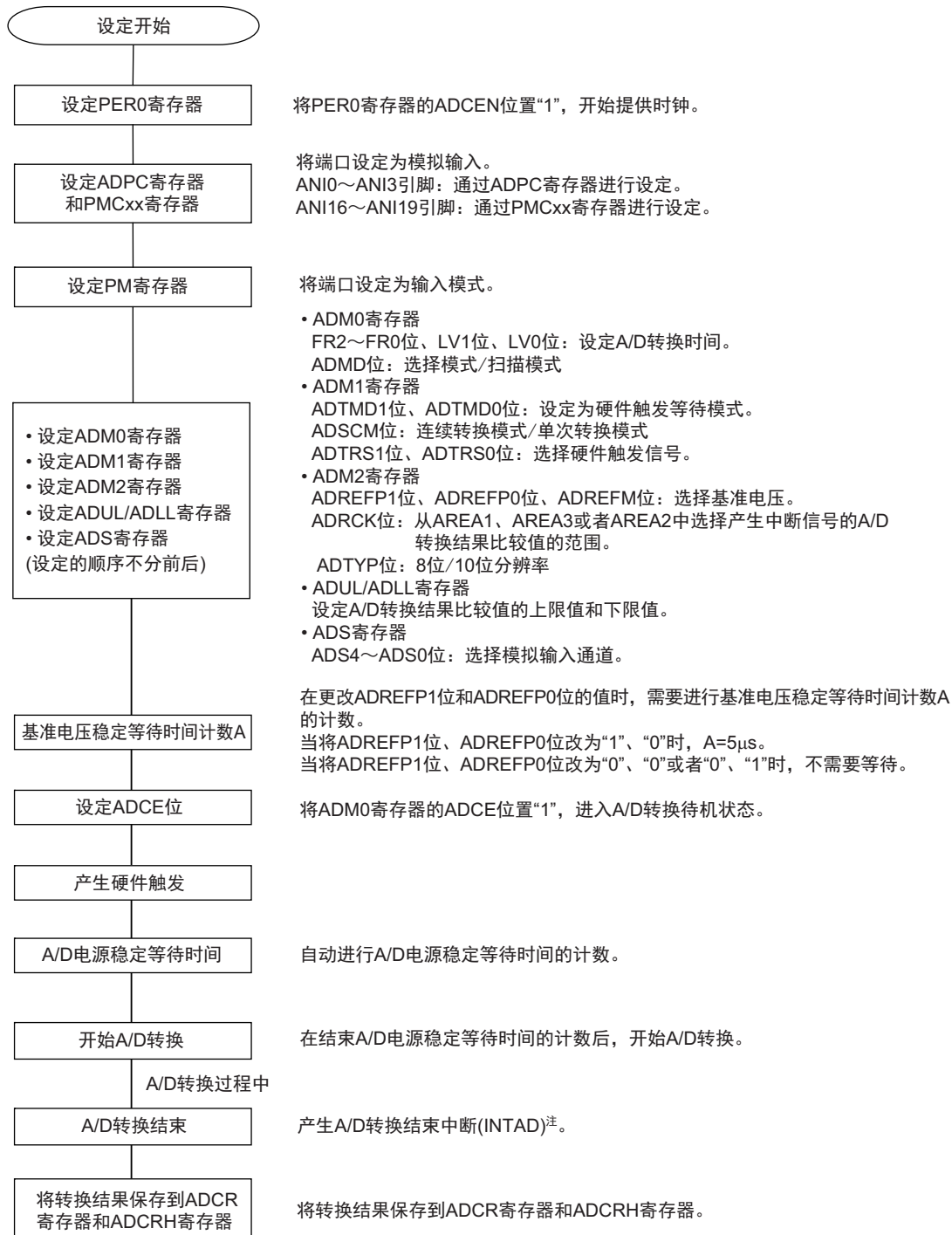
图 10-30 硬件触发无等待模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

10.7.3 硬件触发等待模式的设定

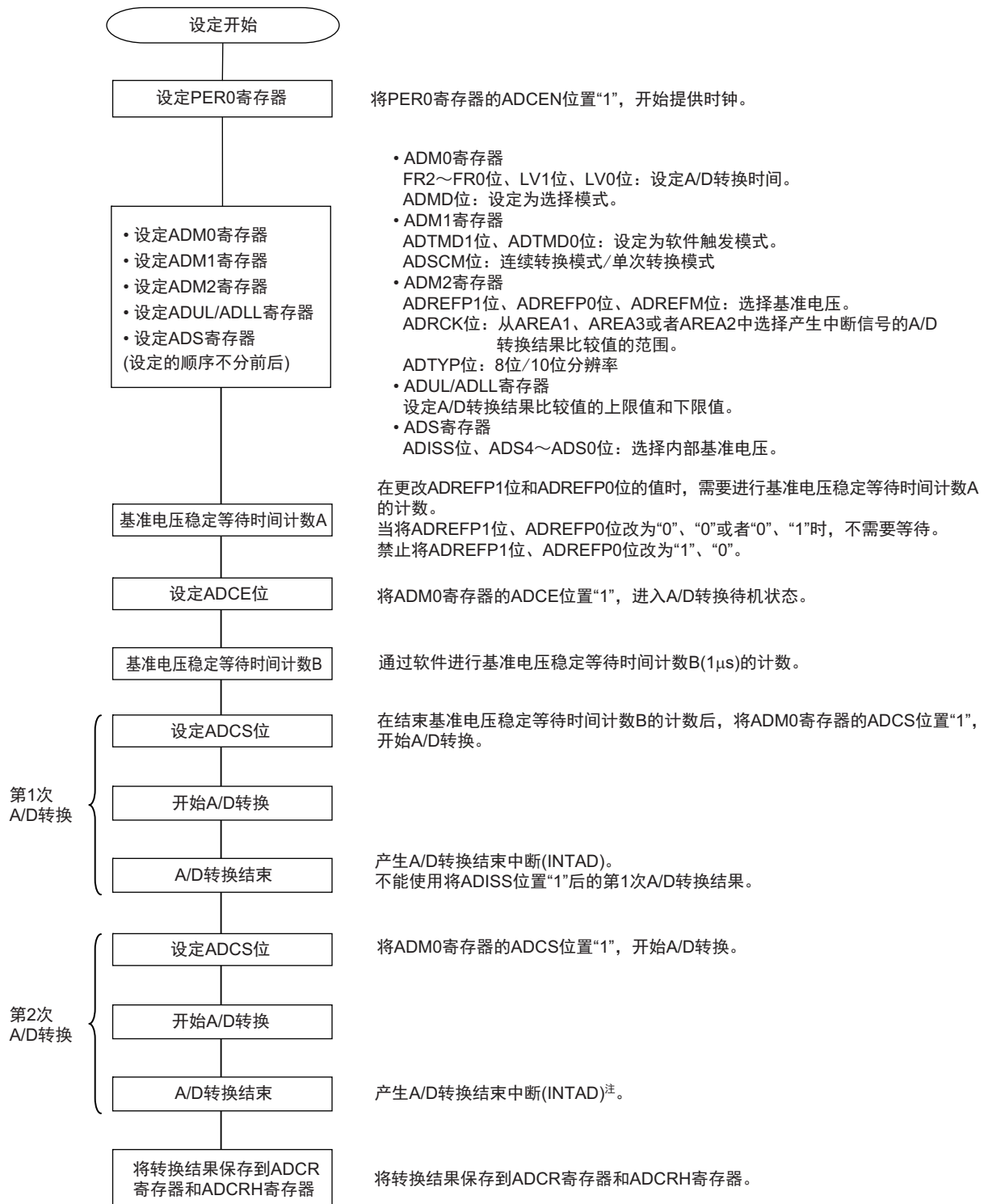
图 10-31 硬件触发等待模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

10.7.4 选择内部基准电压时的设定（以软件触发模式、单次转换模式为例）

图 10-32 选择内部基准电压时的设定

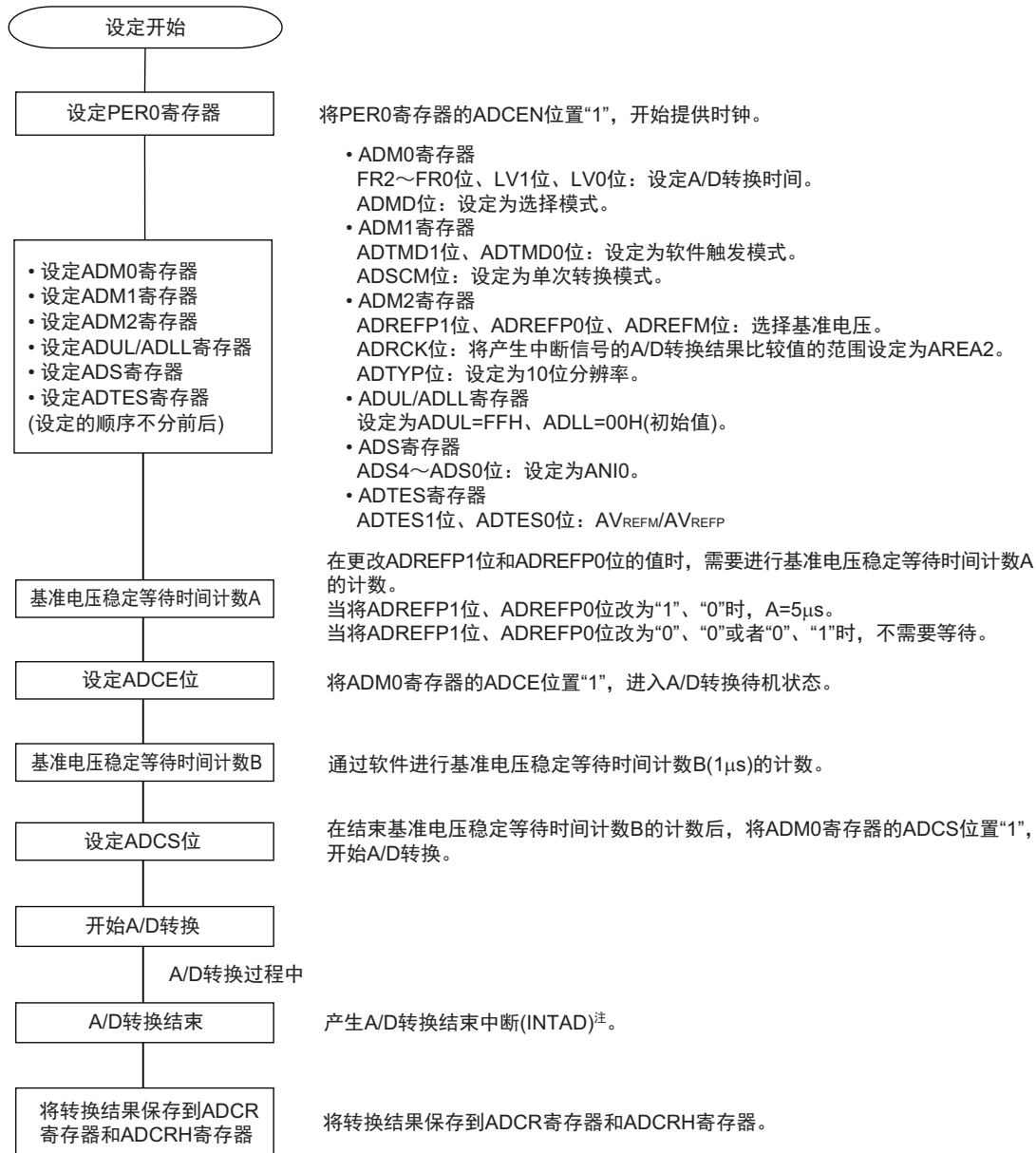


注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

注意 只有在 HS（高速主）模式中才能选择。

10.7.5 测试模式的设定

图 10-33 测试模式的设定



注 根据 ADRCK 位和 ADUL/ADLL 寄存器的设定，有可能不产生中断信号。此时，不将结果保存到 ADCR 寄存器和 ADCRH 寄存器。

注意 有关 A/D 转换器的测试方法，请参照“20.3.8 A/D 测试功能”。

10.8 SNOOZE 模式功能

这是在 STOP 模式中通过输入硬件触发进行 A/D 转换的模式。通常，在 STOP 模式中停止 A/D 转换，但是如果使用 SNOOZE 模式功能，就能在 CPU 不运行的状态下进行 A/D 转换，减小工作电流。

在 SNOOZE 模式中，如果通过 ADUL 寄存器和 ADLL 寄存器指定转换结果的范围，就能每隔一段时间判断 A/D 转换结果。从而能判断电源电压监视或者通过 A/D 输入进行的键输入。

在 SNOOZE 模式中，只能使用以下 2 种转换模式：

- 硬件触发等待模式（选择模式和单次转换模式）
- 硬件触发等待模式（扫描模式和单次转换模式）

注意 只有在选择高速内部振荡器时钟作为 f_{CLK} 时才能设定 SNOOZE 模式。

图 10-34 使用 SNOOZE 模式功能时的框图



当使用 SNOOZE 模式功能时，在转移到 STOP 模式前进行各寄存器的初始设定（参照“10.7.3 硬件触发等待模式的设定”注2）。在要转移到 STOP 模式前，将 A/D 转换器的模式寄存器 2（ADM2）的 bit2（AWC）置“1”。在初始设定结束后，将 A/D 转换器的模式寄存器 0（ADM0）的 bit0（ADCE）置“1”。

如果在转移到 STOP 模式后输入硬件触发，就给 A/D 转换器提供高速内部振荡器时钟。在提供高速内部振荡器时钟后，自动对 A/D 电源的稳定等待时间进行计数，开始 A/D 转换。

A/D 转换结束后的 SNOOZE 模式的运行因是否产生中断信号而不同注1。

- 注**
1. 根据 A/D 转换结果比较功能的设定（ADRCK 位、ADUL/ADLL 寄存器），有可能不产生中断请求信号。
 2. 必须将 ADM1 寄存器置“E2H”或者“E3H”。

备注 硬件触发为 INTIT。

必须通过 A/D 转换器的模式寄存器 1（ADM1）设定硬件触发。

(1) 在 A/D 转换结束后发生中断的情况

如果 A/D 转换结果的值在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定）的设定值范围内，就产生 A/D 转换结束中断请求信号（INTAD）。

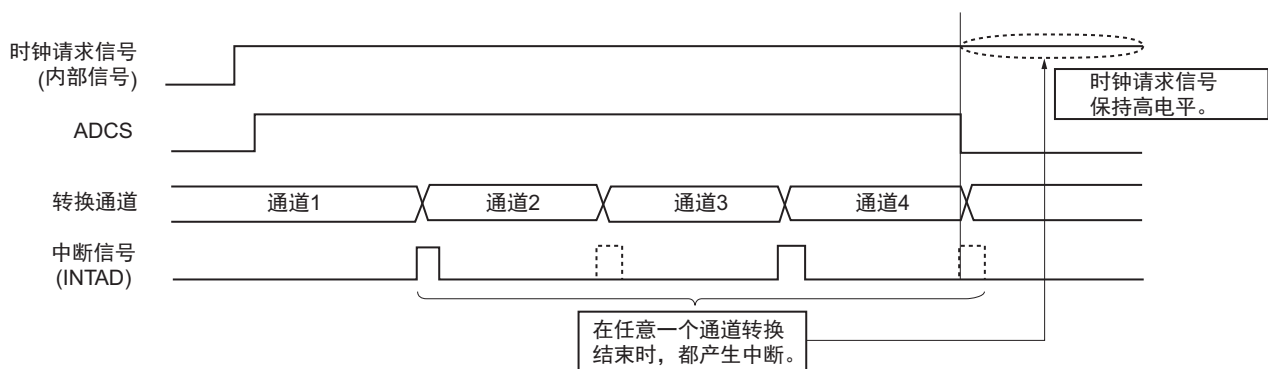
- 选择模式

如果在 A/D 转换结束后产生 A/D 转换结束中断请求信号（INTAD），A/D 转换器就从 SNOOZE 模式转移到通常运行模式。在此，必须清除 A/D 转换器的模式寄存器 2（ADM2）的 bit2（AWC=0：解除 SNOOZE 模式）。如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。

- 扫描模式

在 4 个通道的 A/D 转换过程中只要产生 1 次 A/D 转换结束中断请求信号（INTAD），A/D 转换器就从 SNOOZE 模式转移到通常运行模式。在此，必须清除 A/D 转换器模式寄存器 2（ADM2）的 bit2（AWC=0：解除 SNOOZE 模式）。如果 AWC 位保持“1”的状态，就无法正常开始 A/D 转换，而与其后的 SNOOZE 模式和通常运行模式无关。

图 10-35 在 A/D 转换结束后发生中断时的运行例子（扫描模式）



(2) 在 A/D 转换结束后不发生中断的情况

如果 A/D 转换结果的值不在 A/D 转换结果比较功能（通过 ADRCK 位和 ADUL/ADLL 寄存器进行设定）的设定值范围内，就不产生 A/D 转换结束中断请求信号（INTAD）。

- 选择模式

如果不产生 A/D 转换结束中断请求信号（INTAD），时钟请求信号（内部信号）就在 A/D 转换结束后自动变为低电平，并且停止提供高速内部振荡器时钟。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。

- 扫描模式

如果在 4 个通道的 A/D 转换过程中 1 次也不产生 A/D 转换结束中断请求信号（INTAD），时钟请求信号（内部信号）就在 4 个通道的 A/D 转换结束后自动变为低电平，并且停止提供高速内部振荡器时钟。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。

图 10-36 在 A/D 转换结束后不发生中断时的运行例子（扫描模式）

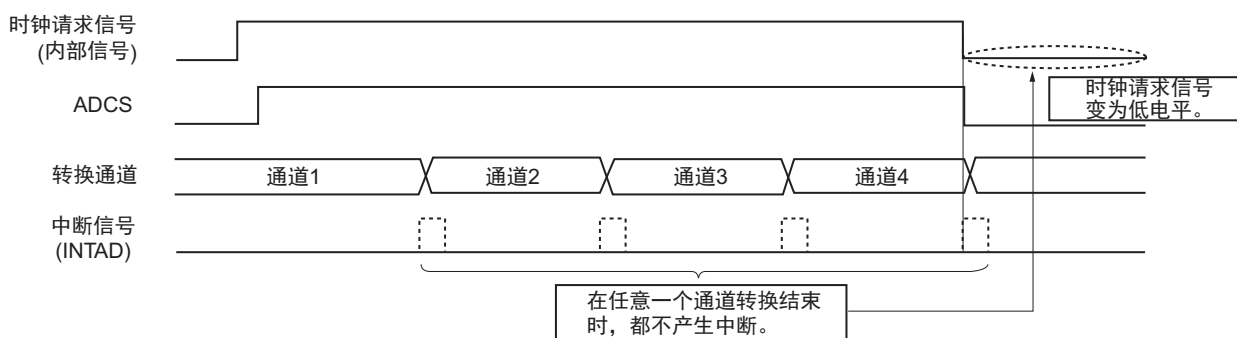
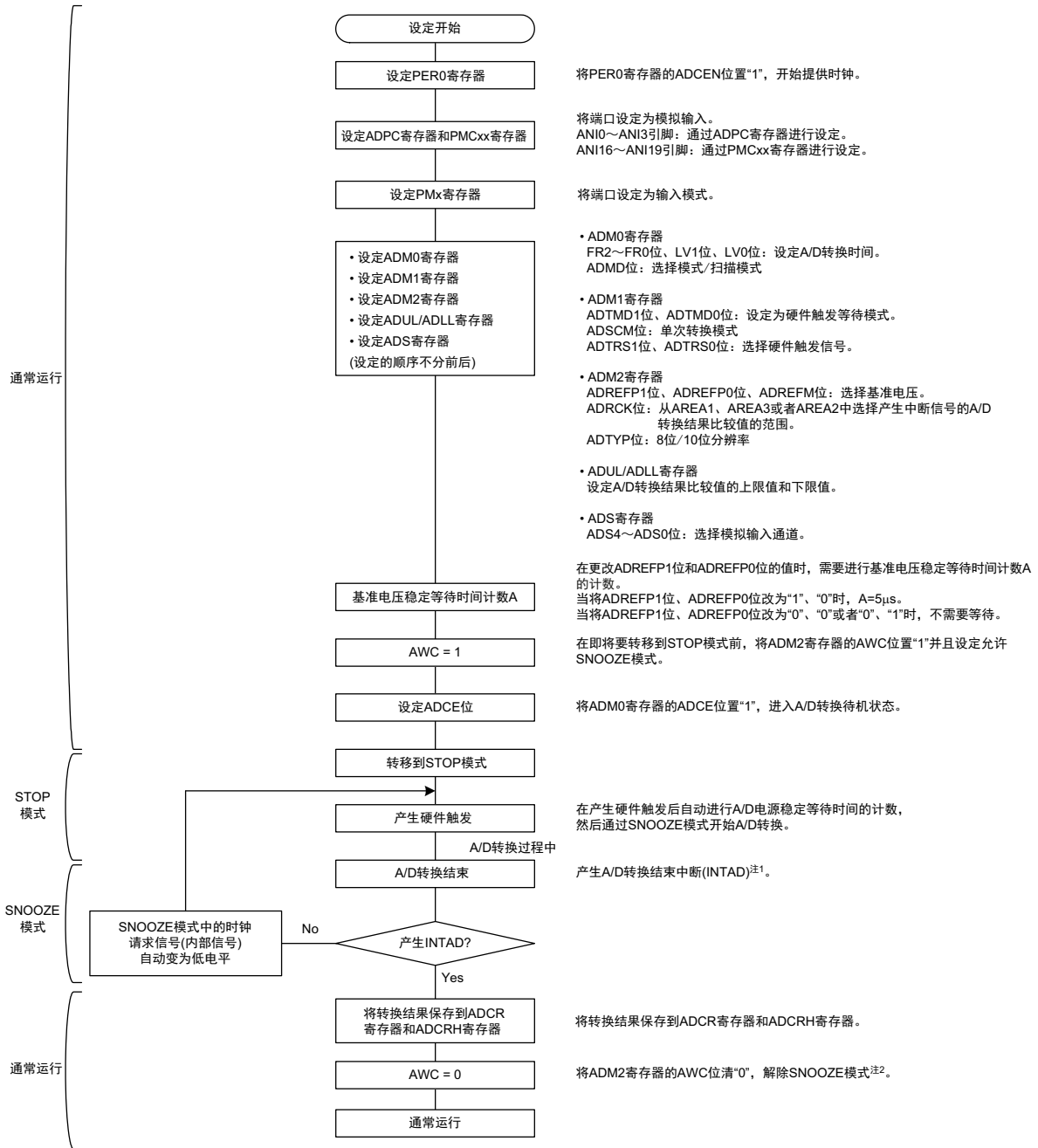


图 10-37 SNOOZE 模式的设定流程



- 注 1. 如果根据 ADRCK 位和 ADUL/ADLL 寄存器的设定不产生 A/D 转换结束中断请求信号 (INTAD)，就不将结果保存到 ADCR 寄存器和 ADCRH 寄存器，再次转移到 STOP 模式。此后，如果输入硬件触发，就再次在 SNOOZE 模式中进行 A/D 转换。
2. 如果 AWC 位保持 “1” 的状态，就无法正常开始 A/D 转换，而与之后的 SNOOZE 模式和通常运行模式无关。必须将 AWC 位置 “0”。

10.9 A/D 转换器特性表的阅读方法

以下说明 A/D 转换器特有的专业术语。

(1) 分辨率

分辨率是能分辨的最小模拟输入电压。也就是说，数字输出的每 1 位与模拟输入电压的比率称为 1 LSB (Least Significant Bit)。将对 1 LSB 满刻度的比率表示为 %FSR (Full Scale Range)。

当分辨率为 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。

(2) 综合误差

综合误差是指实际测量值和理论值的最大差值，是将零刻度误差、满刻度误差、积分线性误差、微分线性误差以及这些组合所产生的误差综合起来的误差。

特性表中的综合误差不包含量化误差。

(3) 量化误差

在将模拟值转换为数字值时，必然会出现 $\pm 1/2 \text{LSB}$ 的误差。A/D 转换器将 $\pm 1/2 \text{LSB}$ 范围内的模拟输入电压转换为相同的数字码，因此不能避免量化误差。

特性表中的综合误差、零刻度误差、满刻度误差、积分线性误差和微分线性误差不包含量化误差。

图 10-38 综合误差

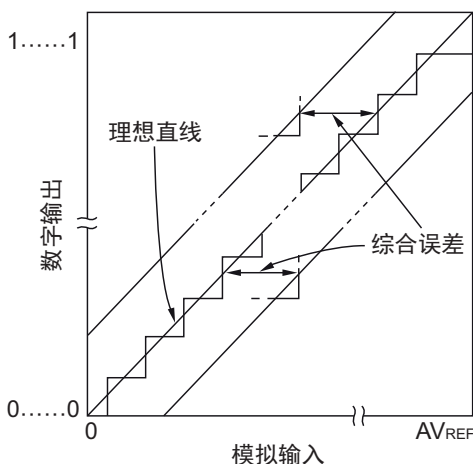
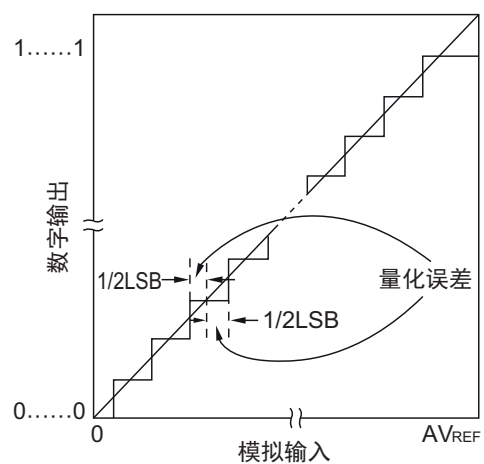


图 10-39 量化误差



(4) 零刻度误差

零刻度误差是指数字输出从 0.....000 变为 0.....001 时的模拟输入电压的实际测量值和理论值 ($1/2 \text{LSB}$) 的差。如果实际测量值大于理论值，零刻度误差就是指数字输出从 0.....001 变为 0.....010 时的模拟输入电压的实际测量值和理论值 ($3/2 \text{LSB}$) 的差。

(5) 满刻度误差

满刻度误差是指数字输出从 1.....110 变为 1.....111 时的模拟输入电压的实际测量值和理论值（满刻度 $-3/2$ LSB）的差。

(6) 积分线性误差

积分线性误差是指转换特性从理想线性关系偏离的程度，是零刻度误差和满刻度误差为 0 时的实际测量值和理想直线的差的最大值。

(7) 微分线性误差

微分线性误差是指在输出代码的理想宽度为 1LSB 时某个代码输出宽度的实际测量值和理想值的差值。

图 10-40 零刻度误差

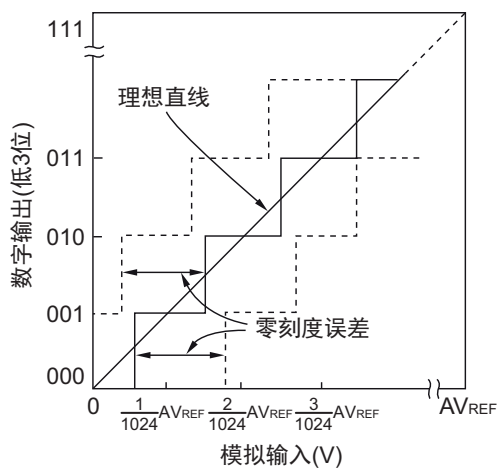


图 10-41 满刻度误差

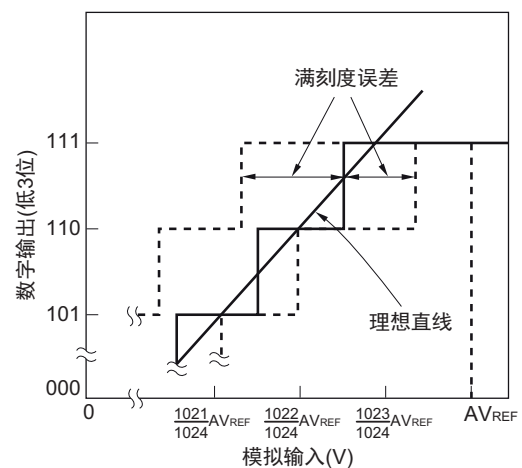


图 10-42 积分线性误差

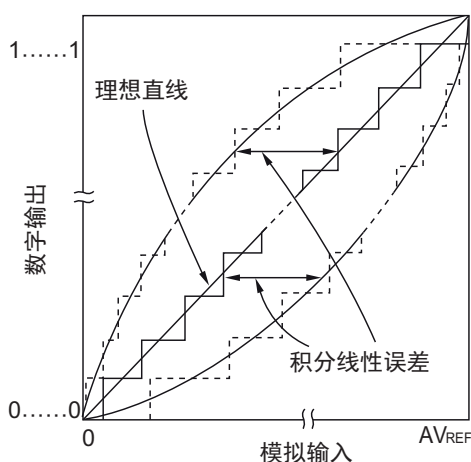
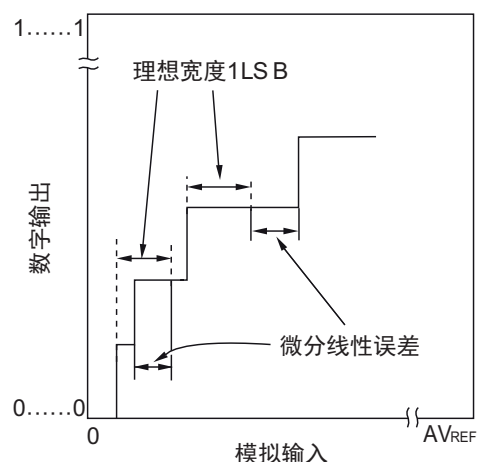


图 10-43 微分线性误差

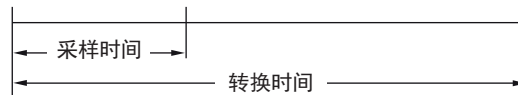


(8) 转换时间

转换时间是指从开始采样到取得数字输出的时间。
特性表中的转换时间包含采样时间。

(9) 采样时间

采样时间是指为了将模拟电压取入采样 & 保持电路而使模拟开关 ON 的时间。



10.10 A/D 转换器的注意事项

(1) 有关 STOP 模式中的工作电流

要转移到 STOP 模式时，必须在停止 A/D 转换器（将 A/D 转换器的模式寄存器 0 (ADM0) 的 bit7 (ADCS) 置“0”) 后进行。此时，还能通过将 ADM0 寄存器的 bit0 (ADCE) 置“0”来减小工作电流。

要从待机状态重新运行时，必须在将中断请求标志寄存器 1H (IF1H) 的 bit0 (ADIF) 清“0”后开始运行。

(2) 有关 ANI0 ~ ANI3 引脚和 ANI16 ~ ANI19 引脚的输入范围

ANI0 ~ ANI3 引脚和 ANI16 ~ ANI19 引脚的输入电压必须满足规格范围。如果输入大于 V_{DD} 和 AV_{REFP} 或者小于 V_{SS} 和 AV_{REFM} （即使在绝对最大额定范围内）的电压，该通道的转换值就为不定值，并且还可能影响其他通道的转换值。

当选择内部基准电压（1.45V）作为 A/D 转换器的正（+）基准电压源时，不能给 ADS 寄存器选择的引脚输入大于内部基准电压（1.45V）的电压。但是，能给 ADS 寄存器没有选择的引脚输入大于内部基准电压的电压（1.45V）。

注意 只有在 HS（高速主）模式中才能选择内部基准电压（1.45V）。

(3) 有关竞争

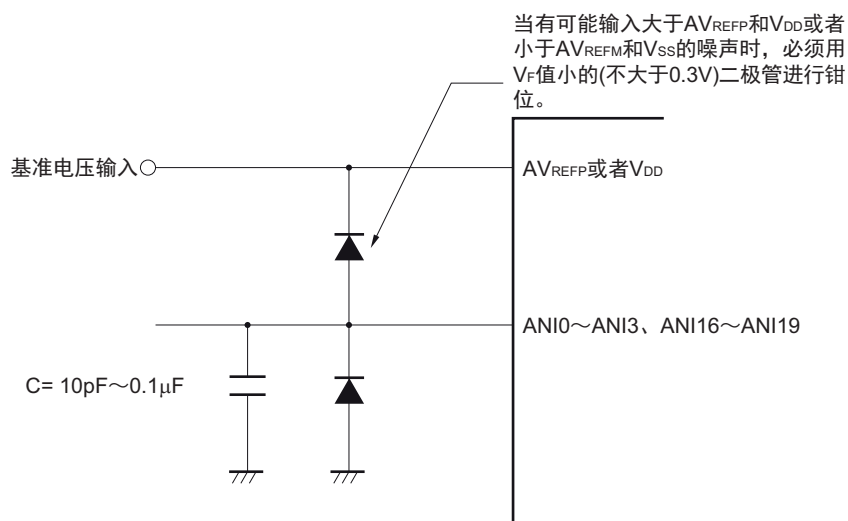
- ① 转换结束时的 A/D 转换结果寄存器（ADCR、ADCRH）的写操作与 ADCR 寄存器和 ADCRH 寄存器的指令读操作的竞争
优先读 ADCR 寄存器和 ADCRH 寄存器。在读后，将新的转换结果写到 ADCR 寄存器和 ADCRH 寄存器。
- ② 转换结束时的 ADCR 寄存器和 ADCRH 寄存器的写操作与 A/D 转换器的模式寄存器 0 (ADM0)、模拟输入通道指定寄存器 (ADS) 或者 A/D 端口配置寄存器 (ADPC) 的写操作的竞争
优先写 ADM0、ADS、ADPC 寄存器。不写 ADCR 寄存器和 ADCRH 寄存器，也不产生转换结束中断信号 (INTAD)。

(4) 有关噪声对策

为了保持 10 位分辨率，需要注意输入到 AV_{REFP} 、 V_{DD} 、 $ANI0 \sim ANI3$ 、 $ANI16 \sim ANI19$ 引脚的噪声。

- ① 必须以最短的距离并且用较粗的布线连接电容器，连接电源的电容器（ $0.01\mu\text{F}$ 左右）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 10-44 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。

图 10-44 模拟输入引脚的处理



(5) 模拟输入（ANIn）引脚

- ① 模拟输入（ $ANI0 \sim ANI3$ ）引脚和输入端口（ $P20 \sim P23$ ）引脚复用。
要选择 $ANI0 \sim ANI3$ 引脚中的任意引脚进行 A/D 转换时，不能在转换过程中更改 $P20 \sim P23$ 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，不能输入或者输出类似数字信号的有急剧变化的脉冲。

(6) 有关模拟输入（ANIn）引脚的输入阻抗

此 A/D 转换器在采样时间内给内部的采样电容充电并且进行采样。

因此，在不采样时只有漏电流流过，而在采样时还有电容充电的电流流过。所以输入阻抗根据是否采样而发生变化。

但是，为了充分地进行采样，必须将模拟输入源的输出阻抗保持在 $1\text{k}\Omega$ 以下。建议在无法将输出阻抗保持在 $1\text{k}\Omega$ 以下时延长采样时间或者给 $ANI0 \sim ANI3$ 引脚和 $ANI16 \sim ANI19$ 引脚连接 $0.1\mu\text{F}$ 左右的电容器（参照图 10-44）。另外，如果在转换过程中将 ADCS 位置“0”或者重新开始转换，采样电容器的充电电压就不稳定。因此，当将 ADCS 位置“0”时，下一次转换就从不稳定状态开始充电；当重新转换时，该转换就从不稳定状态开始充电。为了充分地进行充电，与模拟信号变化的大小无关，必须降低模拟输入源的输出阻抗或者确保充分的采样时间。

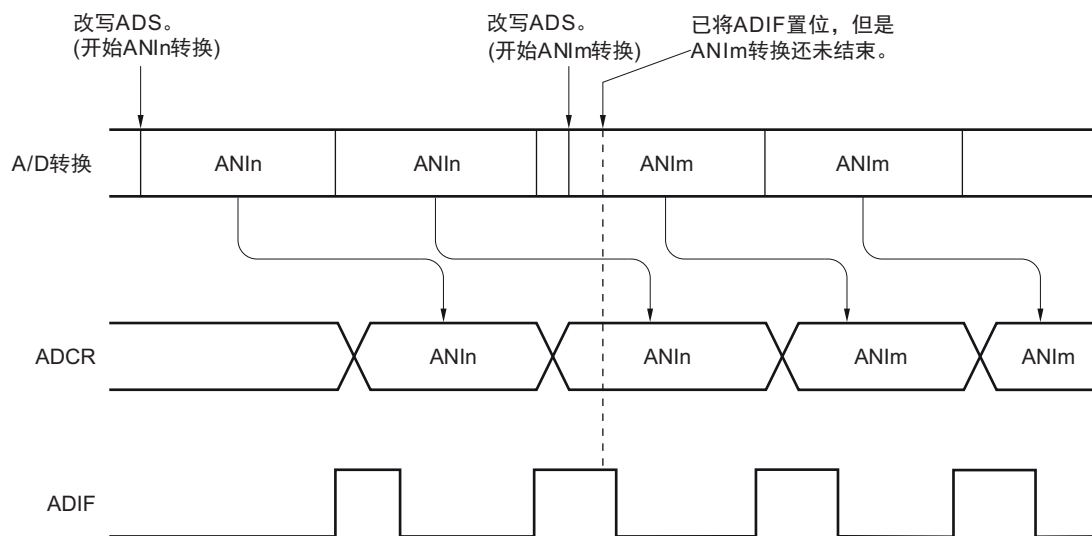
(7) 有关中断请求标志 (ADIF)

即使更改模拟输入通道指定寄存器 (ADS)，也不将中断请求标志 (ADIF) 清“0”。

因此，如果在 A/D 转换过程中更改模拟输入引脚，就可能在即将改写 ADS 寄存器前，更改前的模拟输入的 A/D 转换结果和 ADIF 标志被设定。必须注意：如果在改写 ADS 寄存器后立即读 ADIF 标志，尽管转换后的模拟输入的 A/D 转换还未结束，也会将 ADIF 标志置位。

另外，在暂停后重新开始 A/D 转换时，必须在重新开始前将 ADIF 标志清“0”。

图 10-45 A/D 转换结束中断请求的产生时序



(8) 有关 A/D 转换开始后的初次转换结果

在软件触发模式或者硬件触发无等待模式中，如果在将 ADCE 位置“1”后的 $1\mu\text{s}$ 内将 ADCS 位置“1”，开始 A/D 转换后的初次 A/D 转换值就可能不满足额定值。必须采取轮询 A/D 转换结束中断请求 (INTAD) 并且取消初次的转换结果等对策。

(9) 有关 A/D 转换结果寄存器 (ADCR、ADCRH) 的读操作

当写 A/D 转换器的模式寄存器 0 (ADM0)、模拟输入通道指定寄存器 (ADS)、A/D 端口配置寄存器 (ADPC) 和端口模式控制寄存器 (PMCxx) 时，ADCR 寄存器和 ADCRH 寄存器的内容可能不定。必须在转换结束后并且在写 ADM0、ADS、ADPC、PMCxx 寄存器前读转换结果。否则，就可能读不到正确的转换结果。

(10) 有关内部等效电路

模拟输入部的等效电路如下所示。

图 10-46 ANIn 引脚的内部等效电路

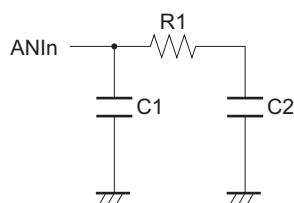


表 10-4 等效电路的各电阻和电容值（参考值）

AV_{REFP} 、 V_{DD}	ANIn 引脚	R1[k Ω]	C1[pF]	C2[pF]
$3.6V \leq V_{DD} \leq 5.5V$	ANI0 ~ ANI3	14	8	2.5
	ANI16 ~ ANI19	18	8	7.0
$2.7V \leq V_{DD} < 3.6V$	ANI0 ~ ANI3	39	8	2.5
	ANI16 ~ ANI19	53	8	7.0
$1.8V \leq V_{DD} < 2.7V$	ANI0 ~ ANI3	231	8	2.5
	ANI16 ~ ANI19	321	8	7.0
$1.6V \leq V_{DD} < 1.8V$	ANI0 ~ ANI3	632	8	2.5
	ANI16 ~ ANI19	902	8	7.0

备注 表 10-4 的各电阻和电容值不是保证值。

(11) 有关 A/D 转换器的运行开始

必须在 AV_{REFP} 和 V_{DD} 的电压稳定后开始 A/D 转换器的运行。

第 11 章 串行阵列单元

串行阵列单元的1个单元最多有4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。本产品支持的各通道的功能分配如下：

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

在单元 0 的通道 0 和通道 1 使用 UART0 时，不能使用 CSI00，但是能使用 UART1。

11.1 串行阵列单元的功能

本产品支持的各串行接口的特征如下所示。

11.1.1 3 线串行 I/O (CSI00、CSI11、CSI20)

与主控设备输出的串行时钟 (SCK) 同步进行数据的发送和接收。

这是使用 1 条串行时钟 (SCK)、1 条发送串行数据 (SO) 和 1 条接收串行数据 (SI) 共 3 条通信线进行通信的时钟同步通信功能。

有关具体的设定例子, 请参照“11.5 3 线串行 I/O (CSI00、CSI11、CSI20) 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: $\text{Max. } f_{\text{CLK}}/2$ (只限于 CSI00)

$\text{Max. } f_{\text{CLK}}/4$

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

以下通道的 CSI 支持 SNOOZE 模式。SNOOZE 模式功能是指: 如果在 STOP 模式的状态下检测到 SCK 的输入, 就不需要 CPU 运行而接收数据。只有支持异步接收的以下 CSI 才能设定 SNOOZE 模式。

- 本产品: CSI00

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 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”。

11.1.2 UART (UART0 ~ UART2)

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

有关具体的设定例子,请参照“11.6 UART (UART0 ~ UART2) 通信的运行”。

[数据的发送和接收]

- 7位、8位或者9位的数据长度注
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定、反相的选择
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加

[中断功能]

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

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

以下通道的 UART 接收支持 SNOOZE 模式。SNOOZE 模式功能是指:如果在 STOP 模式的状态下检测到 RxD 的输入,就不需要 CPU 运行而接收数据。只有在接收时支持波特率调整功能的以下 UART 才能设定 SNOOZE 模式。

- 本产品: UART0

注 只有以下的 UART 支持 9 位数据长度。

- 本产品: UART0

11.1.3 简易 I²C (IIC00、IIC11、IIC20)

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

对于开始条件和停止条件, 必须遵守 AC 规格, 在操作控制寄存器的同时通过软件进行处理。

有关具体的设定例子, 请参照“11.7 简易 I²C (IIC00、IIC11、IIC20) 通信的运行”。

[数据的发送和接收]

- 主控发送、主控接收 (只限于单主控的主控功能)
- ACK 输出功能注、ACK 检测功能
- 8 位数据长度 (在发送地址时, 用高 7 位指定地址, 用最低位进行 R/W 控制)
- 开始条件和停止条件的手动产生

[中断功能]

- 传送结束中断

[错误检测标志]

- ACK 错误、溢出错误

※ [简易 I²C 不支持的功能]

- 从属发送、从属接收
- 仲裁失败检测功能
- 等待检测功能

注 在接收最后的数据时, 如果给 SOEmn 位 (串行输出允许寄存器 m (SOEm)) 写“0”来停止串行通信数据的输出, 就不输出 ACK。详细内容请参照“11.7.3(2) 处理流程”。

备注 1. 在使用全功能的 I²C 总线时, 请参照“第 12 章 串行接口 IICA”。

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.2 串行阵列单元的结构

串行阵列单元由以下硬件构成。

表 11-1 串行阵列单元的结构

项目	结构
移位寄存器	8 位或者 9 位注 1
缓冲寄存器	串行数据寄存器 mn (SDRmn) 的低 8 位或者 9 位注 1、2
串行时钟输入 / 输出	SCK00、SCK11、SCK20 引脚 (用于 3 线串行 I/O)、SCL00、SCL11、SCL20 引脚 (用于简易 I ² C)
串行数据输入	SI00、SI11、SI20 引脚 (用于 3 线串行 I/O)、RxD0、RxD1 引脚 (用于 UART)、RxD2 引脚
串行数据输出	SO00、SO11、SO20 引脚 (用于 3 线串行 I/O)、TxD0、TxD1 引脚 (用于 UART)、TxD2 引脚
串行数据输入 / 输出	SDA00、SDA11、SDA20 引脚 (用于简易 I ² C)
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) • 串行时钟选择寄存器 m (SPSm) • 串行通道允许状态寄存器 m (SEm) • 串行通道开始寄存器 m (SSm) • 串行通道停止寄存器 m (STm) • 串行输出允许寄存器 m (SOEm) • 串行输出寄存器 m (SOM) • 串行输出电平寄存器 m (SOLm) • 串行待机控制寄存器 0 (SSC0) • 噪声滤波器允许寄存器 0 (NFEN0) <p><各通道部的寄存器></p> <ul style="list-style-type: none"> • 串行数据寄存器 mn (SDRmn) • 串行模式寄存器 mn (SMRmn) • 串行通信运行设定寄存器 mn (SCRmn) • 串行状态寄存器 mn (SSRmn) • 串行标志清除触发寄存器 mn (SIRmn) <ul style="list-style-type: none"> • 端口输入模式寄存器 0、1 (PIM0、PIM1) • 端口输出模式寄存器 0、1、5 (POM0、POM1、POM5) • 端口模式控制寄存器 0 (PMC0) • 端口模式寄存器 0、1、3、5 (PM0、PM1、PM3、PM5) • 端口寄存器 0、1、3、5 (P0、P1、P3、P5)

注 1. 用作移位寄存器和缓冲寄存器的位数因单元和通道而不同。

- 本产品并且 mn=00、01: 低 9 位
- 上述以外: 低 8 位

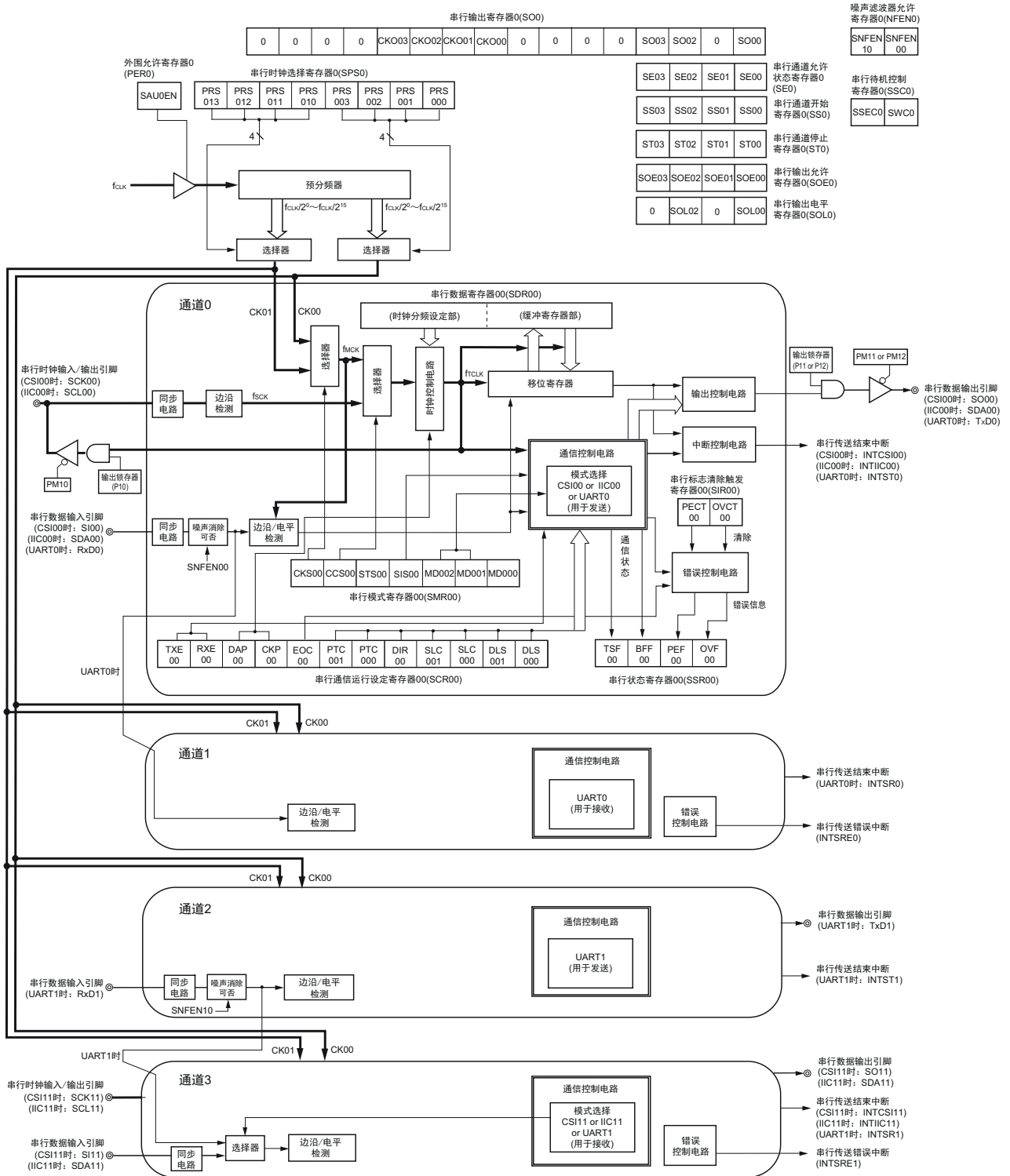
2. 根据通信模式, 能用以下 SFR 名称读写串行数据寄存器 mn (SDRmn) 的低 8 位。

- CSIp 通信SIOp (CSIp 数据寄存器)
- UARTq 接收RXDq (UARTq 接收数据寄存器)
- UARTq 发送TXDq (UARTq 发送数据寄存器)
- IICr 通信SIOr (IICr 数据寄存器)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20)
q: UART 号 (q=0~2) r: IIC 号 (r=00、11、20)

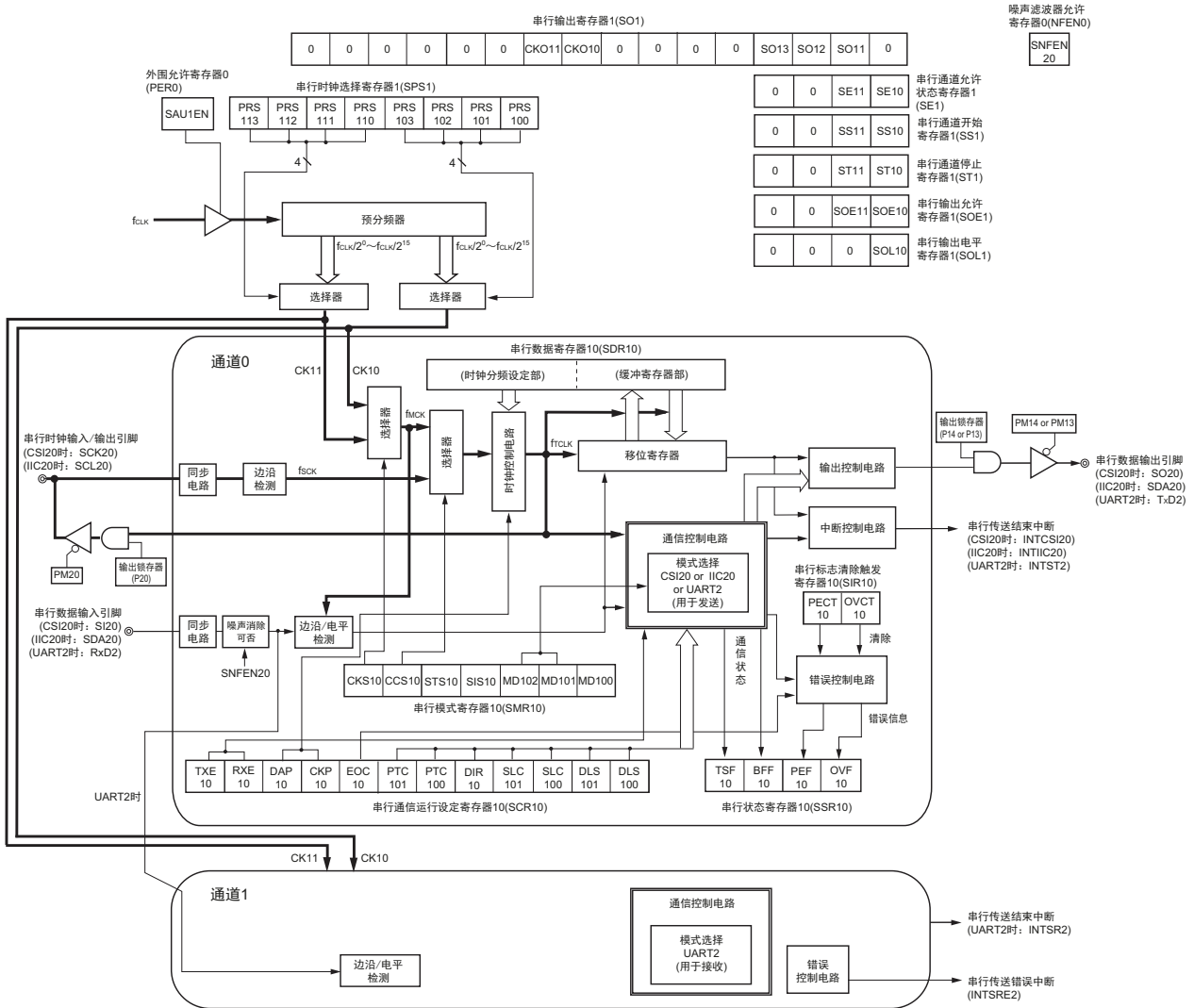
串行阵列单元 0 的框图如图 11-1 所示。

图 11-1 串行阵列单元 0 的框图



串行阵列单元 1 的框图如图 11-2 所示。

图 11-2 串行阵列单元 1 的框图



11.2.1 移位寄存器

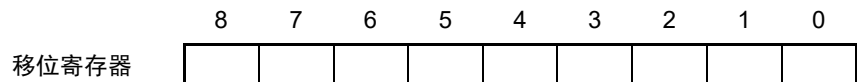
这是进行并行和串行相互转换的 9 位寄存器。

在以 9 位数据长度进行 UART 通信时，使用 9 位（bit0 ~ 8）注 1。

在接收数据时，将串行输入引脚的输入数据转换为并行数据；在发送数据时，将被传送到此寄存器的值作为串行数据从串行输出引脚输出。

不能通过程序直接操作移位寄存器。

要读写移位寄存器的数据时，使用串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位。



11.2.2 串行数据寄存器 mn（SDRmn）的低 8 位或者低 9 位

SDRmn 寄存器是通道 n 的发送和接收数据寄存器（16 位）。

bit8 ~ 0（低 9 位）注 1 或者 bit7 ~ 0（低 8 位）用作发送和接收缓冲寄存器，bit15 ~ 9 用作运行时钟（f_{MCK}）的分频设定寄存器。

在接收数据时，将由移位寄存器转换的并行数据保存到低 8 位或者低 9 位；在发送数据时，将被传送到移位寄存器的发送数据设定到低 8 位或者低 9 位。

与数据的输出顺序无关，根据串行通信运行设定寄存器 mn（SCRmn）的 bit0 和 bit1（DLSmn0、DLSmn1）的设定，保存到低 8 位或者低 9 位的数据如下所示：

- 7 位数据长度（保存在 SDRmn 寄存器的 bit0 ~ 6）
- 8 位数据长度（保存在 SDRmn 寄存器的 bit0 ~ 7）
- 9 位数据长度（保存在 SDRmn 寄存器的 bit0 ~ 8）注 1

能以 16 位为单位读写 SDRmn 寄存器。

根据通信模式，能用以下 SFR 名称，以 8 位为单位读写 SDRmn 寄存器的低 8 位或者低 9 位注 2。

- CSIp 通信SIOp（CSIp 数据寄存器）
- UARTq 接收RXDq（UARTq 接收数据寄存器）
- UARTq 发送TXDq（UARTq 发送数据寄存器）
- IICr 通信SIOr（IICr 数据寄存器）

在产生复位信号后，SDRmn 寄存器的值变为“0000H”。

注 1. 只有以下的 UART 支持 9 位数据长度。

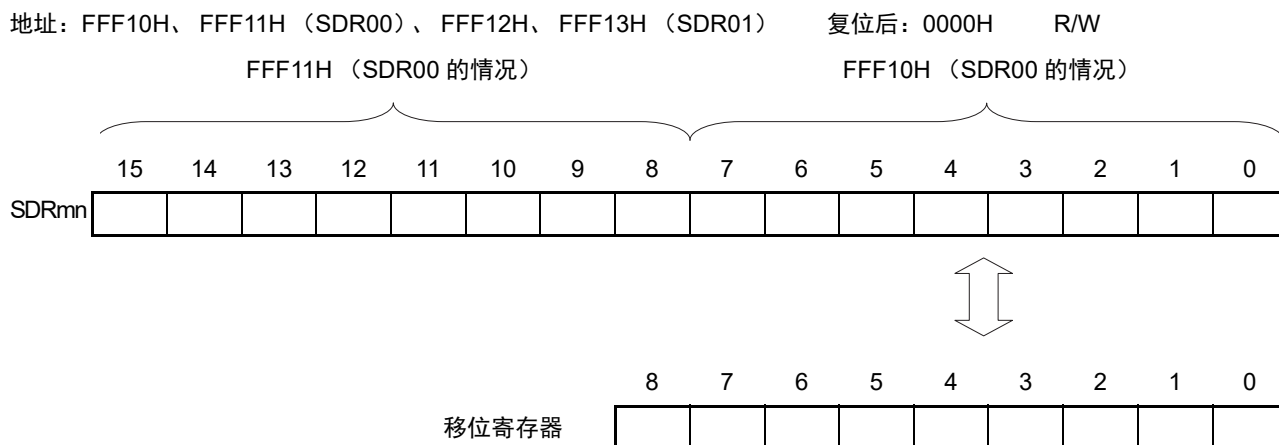
- 本产品：UART0

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

备注 1. 在接收结束后，bit0 ~ 8 中超过数据长度的部分的位为“0”。

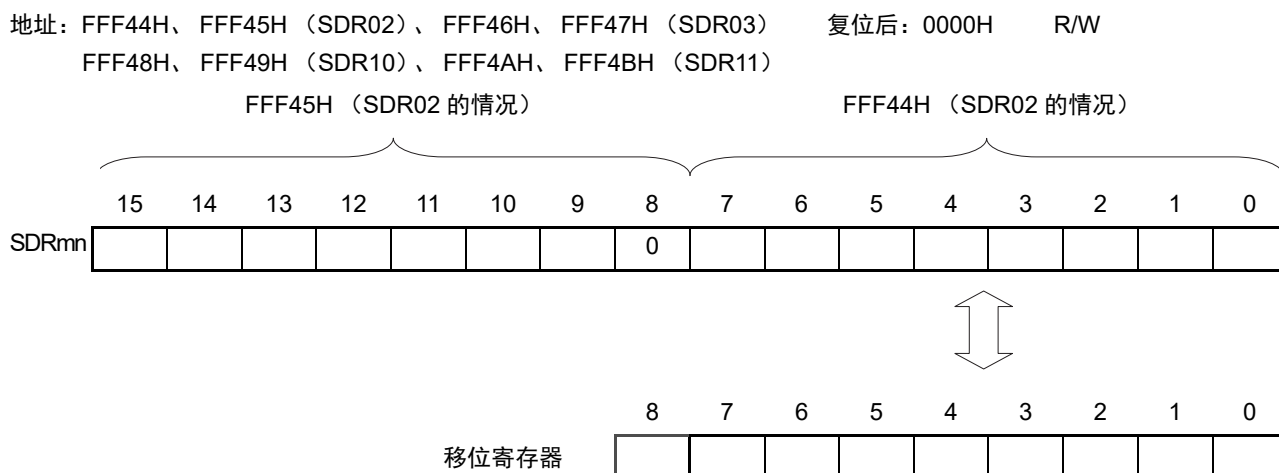
2. m: 单元号（m=0、1） n: 通道号（n=0 ~ 3） p: CSI 号（p=00、11、20）
q: UART 号（q=0 ~ 2） r: IIC 号（r=00、11、20）

图 11-3 串行数据寄存器 mn (SDRmn) (mn=00、01) 的格式



备注 有关 SDRmn 寄存器的高 7 位的功能, 请参照“11.3 控制串行阵列单元的寄存器”。

图 11-4 串行数据寄存器 mn (SDRmn) (mn=02、03、10、11) 的格式



注意 必须将 bit8 置“0”。

备注 有关 SDRmn 寄存器的高 7 位的功能, 请参照“11.3 控制串行阵列单元的寄存器”。

11.3 控制串行阵列单元的寄存器

控制串行阵列单元的寄存器如下所示：

- 外围允许寄存器0 (PER0)
- 串行时钟选择寄存器m (SPSm)
- 串行模式寄存器mn (SMRmn)
- 串行通信运行设定寄存器mn (SCRmn)
- 串行数据寄存器mn (SDRmn)
- 串行标志清除触发寄存器mn (SIRmn)
- 串行状态寄存器mn (SSRmn)
- 串行通道开始寄存器m (SSm)
- 串行通道停止寄存器m (STm)
- 串行通道允许状态寄存器m (SEm)
- 串行输出允许寄存器m (SOEm)
- 串行输出电平寄存器m (SOLm)
- 串行输出寄存器m (SOM)
- 串行待机控制寄存器0 (SSC0)
- 噪声滤波器允许寄存器0 (NFEN0)
- 端口输入模式寄存器0、1 (PIM0、PIM1)
- 端口输出模式寄存器0、1、5 (POM0、POM1、POM5)
- 端口模式控制寄存器0 (PMC0)
- 端口模式寄存器0、1、3、5 (PM0、PM1、PM3、PM5)
- 端口寄存器0、1、3、5 (P0、P1、P3、P5)

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.3.1 外围允许寄存器 0 (PER0)

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

要使用串行阵列单元 0 时，必须将 bit2 (SAU0EN) 置“1”。

要使用串行阵列单元 1 时，必须将 bit3 (SAU1EN) 置“1”。

通过 1 位或者 8 位存储器操作指令设定 PER0 寄存器。

在产生复位信号后，PER0 寄存器的值变为“00H”。

图 11-5 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	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

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

注意 1. 要设定串行阵列单元 m 时，必须先在 SAUmEN 位为“1”的状态下设定以下的寄存器。当 SAUmEN 位为“0”时，串行阵列单元 m 的控制寄存器的值为初始值，忽视写操作（噪声滤波器允许寄存器 0 (NFEN0)、端口输入模式寄存器 0、1 (PIM0、PIM1)、端口输出模式寄存器 0、1、5 (POM0、POM1、POM5)、端口模式控制寄存器 0 (PMC0)、端口模式寄存器 0、1、3、5 (PM0、PM1、PM3、PM5) 以及端口寄存器 0、1、3、5 (P0、P1、P3、P5) 除外）。

- 串行时钟选择寄存器 m (SPSm)
- 串行模式寄存器 mn (SMRmn)
- 串行通信运行设定寄存器 mn (SCRmn)
- 串行数据寄存器 mn (SDRmn)
- 串行标志清除触发寄存器 mn (SIRmn)
- 串行状态寄存器 mn (SSRmn)
- 串行通道开始寄存器 m (SSm)
- 串行通道停止寄存器 m (STm)
- 串行通道允许状态寄存器 m (SEm)
- 串行输出允许寄存器 m (SOEm)
- 串行输出电平寄存器 m (SOLm)
- 串行输出寄存器 m (SOM)
- 串行待机控制寄存器 0 (SSC0)

2. 必须将 bit6 和 bit1 置“0”。

11.3.2 串行时钟选择寄存器 m (SPSm)

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

禁止在运行过程中 (SEmn=1) 改写 SPSm 寄存器。

通过 16 位存储器操作指令设定 SPSm 寄存器。

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

在产生复位信号后，SPSm 寄存器的值变为“0000H”。

图 11-6 串行时钟选择寄存器 m (SPSm) 的格式

地址: F0126H、F0127H (SPS0)、F0166H、F0167H (SPS1) 复位后: 0000H R/W

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

PRS mk3	PRS mk2	PRS mk1	PRS mk0		运行时钟 (CKmk) 的选择注				
					f _{CLK} =2MHz	f _{CLK} =5MHz	f _{CLK} =10MHz	f _{CLK} =20MHz	f _{CLK} =24MHz
0	0	0	0	f _{CLK}	2MHz	5MHz	10MHz	20MHz	24MHz
0	0	0	1	f _{CLK} /2	1MHz	2.5MHz	5MHz	10MHz	12MHz
0	0	1	0	f _{CLK} /2 ²	500kHz	1.25MHz	2.5MHz	5MHz	6MHz
0	0	1	1	f _{CLK} /2 ³	250kHz	625kHz	1.25MHz	2.5MHz	3MHz
0	1	0	0	f _{CLK} /2 ⁴	125kHz	313kHz	625kHz	1.25MHz	1.5MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5kHz	156kHz	313kHz	625kHz	750kHz
0	1	1	0	f _{CLK} /2 ⁶	31.3kHz	78.1kHz	156kHz	313kHz	375kHz
0	1	1	1	f _{CLK} /2 ⁷	15.6kHz	39.1kHz	78.1kHz	156kHz	188kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81kHz	19.5kHz	39.1kHz	78.1kHz	93.8kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91kHz	9.77kHz	19.5kHz	39.1kHz	46.9kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95kHz	4.88kHz	9.77kHz	19.5kHz	23.4kHz
1	0	1	1	f _{CLK} /2 ¹¹	977Hz	2.44kHz	4.88kHz	9.77kHz	11.7kHz
1	1	0	0	f _{CLK} /2 ¹²	488Hz	1.22kHz	2.44kHz	4.88kHz	5.86kHz
1	1	0	1	f _{CLK} /2 ¹³	244Hz	610Hz	1.22kHz	2.44kHz	2.93kHz
1	1	1	0	f _{CLK} /2 ¹⁴	122Hz	305Hz	610Hz	1.22kHz	1.46kHz
1	1	1	1	f _{CLK} /2 ¹⁵	61Hz	153Hz	305Hz	610Hz	732Hz

注 要在串行阵列单元 (SAU) 运行过程中更改被选择为 f_{CLK} 的时钟 (更改系统时钟控制寄存器 (CKC) 的值) 时，必须在停止 SAU 的运行 (串行通道停止寄存器 m (STm) =000FH) 后进行更改。

注意 必须将 bit15 ~ 8 置“0”。

备注 1. f_{CLK}: CPU/ 外围硬件的时钟频率

2. m: 单元号 (m=0、1)

3. k=0、1

11.3.3 串行模式寄存器 mn (SMRmn)

SMRmn 寄存器是设定通道 n 运行模式的寄存器，进行运行时钟 (f_{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。

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

通过 16 位存储器操作指令设定 SMRmn 寄存器。

在产生复位信号后，SMRmn 寄存器的值变为“0020H”。

图 11-7 串行模式寄存器 mn (SMRmn) 的格式 (1/2)

地址: F0110H、F0111H (SMR00) ~ F0116H、F0117H (SMR03) 复位后: 0020H R/W
F0150H、F0151H (SMR10)、F0152H、F0153H (SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn 注	0	SISmn n0 注	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	通道 n 运行时钟 (f_{MCK}) 的选择
0	SPSm 寄存器设定的运行时钟 CKm0
1	SPSm 寄存器设定的运行时钟 CKm1
运行时钟 (f_{MCK}) 用于边沿检测电路。通过设定 CCSmn 位和 SDRmn 寄存器的高 7 位，生成传送时钟 (f_{TCLK})。	

CCSmn	通道 n 传送时钟 (f_{TCLK}) 的选择
0	CKSmn 位指定的运行时钟 f_{MCK} 的分频时钟
1	来自 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 CCSmn 位为“0”时，通过 SDRmn 寄存器的高 7 位进行运行时钟 (f_{MCK}) 的分频设定。	

STSmn 注	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	RxDq 引脚的有效边沿 (在 UART 接收时选择)
在将 SSm 寄存器置“1”后满足上述条件时，开始传送。	

注 只限于 SMR01、SMR03、SMR11 寄存器。

注意 必须将 bit13 ~ 9、7、4、3 (SMR00、SMR02、SMR10 寄存器为 bit13 ~ 6、4、3) 置“0”，并且将 bit5 置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、11、20)
q: UART 号 (q=0 ~ 2) r: IIC 号 (r=00、11、20)

图 11-7 串行模式寄存器 mn (SMRmn) 的格式 (2/2)

地址: F0110H、F0111H (SMR00) ~ F0116H、F0117H (SMR03) 复位后: 0020H R/W
 F0150H、F0151H (SMR10)、F0152H、F0153H (SMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn 注	0	SISm n0 注	1	0	0	MD mn2	MD mn1	MD mn0

SISmn0 注	UART 模式中的通道 n 接收数据的电平反相控制
0	将下降沿检测为起始位。 不将输入的通信数据进行反相。
1	将上升沿检测为起始位。 将输入的通信数据进行反相。

MDmn2	MDmn1	通道 n 运行模式的设定
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。

MDmn0	通道 n 中断源的选择
0	传送结束中断
1	缓冲器空中断 (在数据从 SDRmn 寄存器传送到移位寄存器时发生)
在连续发送时, 如果 MDmn0 位为“1”并且 SDRmn 的数据为空, 就写下一个发送数据。	

注 只限于 SMR01、SMR03、SMR11 寄存器。

注意 必须将 bit13 ~ 9、7、4、3 (SMR00、SMR02、SMR10 寄存器为 bit13 ~ 6、4、3) 置“0”, 并且将 bit5 置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、11、20)
 q: UART 号 (q=0 ~ 2) r: IIC 号 (r=00、11、20)

11.3.4 串行通信运行设定寄存器 mn (SCRmn)

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

禁止在运行过程中 (SEmn=1) 改写 SCRmn 寄存器。

通过 16 位存储器操作指令设定 SCRmn 寄存器。

在产生复位信号后, SCRmn 寄存器的值变为“0087H”。

图 11-8 串行通信运行设定寄存器 mn (SCRmn) 的格式 (1/2)

地址: F0118H、F0119H (SCR00) ~ F011EH、F011FH (SCR03) 复位后: 0087H R/W

F0158H、F0159H (SCR10)、F015AH、F015BH (SCR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1注1	SLC mn0	0	1	DLSm n1注2	DLS mn0

TXEmn	RXEmn	通道 n 运行模式的设定
0	0	禁止通信。
0	1	只进行接收。
1	0	只进行发送。
1	1	进行发送和接收。

DAPmn	CKPmn	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 DAPmn 位和 CKPmn 位都置“0”。

EOCmn	错误中断信号 (INTSREx (x=0 ~ 3)) 的屏蔽控制
0	禁止产生错误中断 INTSREx (产生 INTSRx)。
1	允许产生错误中断 INTSREx (在发生错误时不产生 INTSRx)。

在 CSI 模式和简易 I²C 模式中或者在 UART 发送时, 必须将 EOCmn 位置“0”注3。

- 注 1. 只限于 SCR00、SCR02、SCR10 寄存器。
 2. 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。
 3. 在 EOCmn 位为“0”并且不使用 CSImn 时, 有可能产生错误中断 INTSREn。

注意 必须将 bit3、6、11 置“0” (也必须将 SCR01、SCR03、SCR11 寄存器的 bit5 置“0”), 并且将 bit2 置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、11、20)

图 11-8 串行通信运行设定寄存器 mn (SCRmn) 的格式 (2/2)

地址: F0118H、F0119H (SCR00) ~ F011EH、F011FH (SCR03) 复位后: 0087H R/W
 F0158H、F0159H (SCR10)、F015AH、F015BH (SCR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1注1	SLC mn0	0	1	DLSm n1注2	DLS mn0

PTCmn1	PTCmn0	UART 模式中的奇偶校验位的设定	
		发送	接收
0	0	不输出奇偶校验位。	接收时没有奇偶校验。
0	1	输出奇偶校验注3。	不判断奇偶校验。
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。

在 CSI 模式和简易 I²C 模式中, 必须将 PTCmn1 位和 PTCmn0 位都置“0”。

DIRmn	CSI 和 UART 模式中的数据传送顺序的选择
0	进行 MSB 优先的输入 / 输出。
1	进行 LSB 优先的输入 / 输出。

在简易 I²C 模式中, 必须将 DIRmn 位置“0”。

SLCmn1注1	SLCmn0	UART 模式中的停止位的设定
0	0	无停止位
0	1	停止位长度 =1 位
1	0	停止位长度 =2 位 (只限于 mn=00、02、10)
1	1	禁止设定。

如果选择了传送结束中断, 就在传送完所有停止位后产生中断。
 在 UART 接收时或者在简易 I²C 模式中, 必须设定为 1 个停止位 (SLCmn1、SLCmn0=0、1)。
 在 CSI 模式中, 必须设定为无停止位 (SLCmn1、SLCmn0=0、0)。
 在 UART 发送时, 必须设定为 1 位 (SLCmn1、SLCmn0=0、1) 或者 2 位 (SLCmn1、SLCmn0=1、0)。

DLSmn1注2	DLSmn0	CSI 和 UART 模式中的数据长度的设定
0	1	9 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 8) (只在 UART 模式中可选择)
1	0	7 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 6)
1	1	8 位数据长度 (保存在 SDRmn 寄存器的 bit0 ~ 7)
其他		禁止设定。

在简易 I²C 模式中, 必须将 DLSmn1 位和 DLSmn0 位都置“1”。

- 注 1. 只限于 SCR00、SCR02、SCR10 寄存器。
 2. 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。
 3. 与数据的内容无关, 总是附加“0”。

注意 必须将 bit3、6、11 置“0” (也必须将 SCR01、SCR03、SCR11 寄存器的 bit5 置“0”), 并且将 bit2 置“1”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) p: CSI 号 (p=00、11、20)

11.3.5 串行数据寄存器 mn (SDRmn)

SDRmn 寄存器是通道 n 发送和接收的数据寄存器 (16 位)。

SDR00、SDR01 的 bit8 ~ 0 (低 9 位) 或者 SDR02、SDR03、SDR10、SDR11 的 bit7 ~ 0 (低 8 位) 用作发送和接收缓冲寄存器, bit15 ~ 9 (高 7 位) 用作运行时钟 (f_{MCK}) 的分频设定寄存器。

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

如果将 CCSmn 位置“1”, 就必须将 SDR00、SDR01 的 bit15 ~ 9 (高 7 位) 置“0000000B”。SCKp 引脚的输入时钟 f_{SCK} (CSI 模式的从属传送) 为传送时钟。

SDRmn 寄存器的低 8 位或者低 9 位用作发送和接收缓冲寄存器。在接收数据时, 将移位寄存器转换的并行数据保存到低 8 位或者低 9 位; 在发送数据时, 将被传送到移位寄存器的发送数据设定到低 8 位或者低 9 位。

能以 16 位为单位读写 SDRmn 寄存器。但是, 只有在运行停止 (SEmn=0) 时才能读写高 7 位。在运行中 (SEmn=1) 只能写 SDRmn 寄存器的低 8 位或者低 9 位, 而且 SDRmn 寄存器的高 7 位的读取值总是“0”。

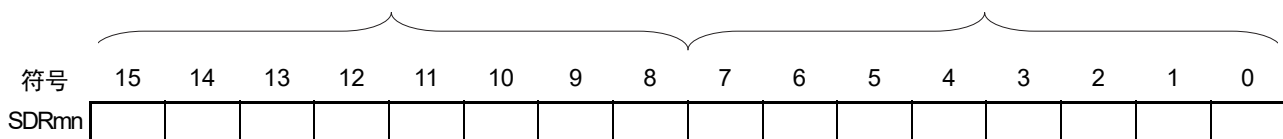
在产生复位信号后, SDRmn 寄存器的值变为“0000H”。

图 11-9 串行数据寄存器 mn (SDRmn) 的格式

地址: FFF10H、FFF11H (SDR00)、FFF12H、FFF13H (SDR01) 复位后: 0000H R/W

FFF11H (SDR00 的情况)

FFF10H (SDR00 的情况)

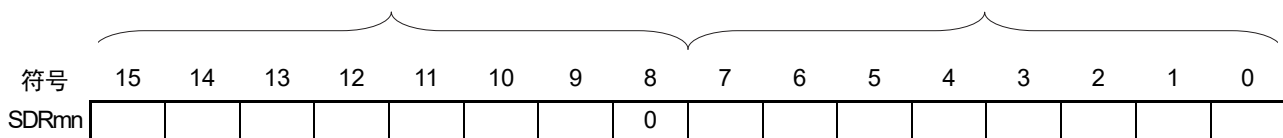


地址: FFF44H、FFF45H (SDR02)、FFF46H、FFF47H (SDR03) 复位后: 0000H R/W

FFF48H、FFF49H (SDR10)、FFF4AH、FFF4BH (SDR11)

FFF45H (SDR02 的情况)

FFF44H (SDR02 的情况)



SDRmn[15:9]							运行时钟分频的传送时钟设定
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

注意 1. 必须将 SDR02、SDR03、SDR10、SDR11 寄存器的 bit8 置“0”。

2. 在使用 UART 时, 禁止将 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

3. 在使用简易 I²C 时, 禁止将 SDRmn[15:9] 设定为“0000000B”, SDRmn[15:9] 的设定值必须大于等于“0000001B”。

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

备注 1. 有关 SDRmn 寄存器的低 8 位或者低 9 位的功能, 请参照“11.2 串行阵列单元的结构”。

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.3.6 串行标志清除触发寄存器 mn (SIRmn)

这是用于清除通道 n 各错误标志的触发寄存器。

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

通过 16 位存储器操作指令设定 SIRmn 寄存器。

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

在产生复位信号后，SIRmn 寄存器的值变为“0000H”。

图 11-10 串行标志清除触发寄存器 mn (SIRmn) 的格式

地址：F0108H、F0109H (SIR00) ~ F010EH、F010FH (SIR03) 复位后：0000H R/W
F0148H、F0149H (SIR10)、F014AH、F014BH (SIR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn注	PEC Tmn	OVC Tmn

FECTmn注	通道 n 帧错误标志的清除触发
0	不清除。
1	将 SSRmn 寄存器的 FEFmn 位清“0”。

PECTmn	通道 n 奇偶校验错误标志的清除触发
0	不清除。
1	将 SSRmn 寄存器的 PEFmn 位清“0”。

OVCTmn	通道 n 溢出错误标志的清除触发
0	不清除。
1	将 SSRmn 寄存器的 OVFmn 位清“0”。

注 只限于 SIR01、SIR03、SIR11 寄存器。

注意 必须将 bit15 ~ 3 (SIR00、SIR02、SIR10 寄存器为 bit15 ~ 2) 置“0”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

2. SIRmn 寄存器的读取值总是“0000H”。

11.3.7 串行状态寄存器 mn (SSRmn)

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

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

在产生复位信号后，SSRmn 寄存器的值变为“0000H”。

图 11-11 串行状态寄存器 mn (SSRmn) 的格式 (1/2)

地址: F0100H、F0101H (SSR00) ~ F0106H、F0107H (SSR03) 复位后: 0000H R
F0140H、F0141H (SSR10)、F0142H、F0143H (SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn注	PEF mn	OVF mn

TSFmn	通道 n 通信状态的表示标志
0	通信停止状态或者通信待机状态
1	通信运行状态
[清除条件]	
<ul style="list-style-type: none"> 当将 STm 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信待机状态）时 当通信结束时 	
[置位条件]	
<ul style="list-style-type: none"> 当开始通信时 	

BFFmn	通道 n 缓冲寄存器的状态表示标志
0	SDRmn 寄存器没有保存有效数据。
1	SDRmn 寄存器保存了有效数据。
[清除条件]	
<ul style="list-style-type: none"> 在发送过程中传送完从 SDRmn 寄存器到移位寄存器的发送数据时 在接收过程中从 SDRmn 寄存器读完接收数据时 当将 STm 寄存器的 STmn 位置“1”（通信停止状态）或者将 SSm 寄存器的 SSmn 位置“1”（通信允许状态）时 	
[置位条件]	
<ul style="list-style-type: none"> 在 SCRmn 寄存器的 TXEmn 位为“1”（各通信模式中的发送模式、发送和接收模式）的状态下给 SDRmn 寄存器写发送数据时 在 SCRmn 寄存器的 RXEmn 位为“1”（各通信模式中的接收模式、发送和接收模式）的状态下将接收数据保存到 SDRmn 寄存器时 当发生接收错误时 	

注 只限于 SSR01、SSR03、SSR11 寄存器。

注意 在 SNOOZE 模式（SWC0=1）中进行 CSI 接收时，BFFmn 标志不变。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

图 11-11 串行状态寄存器 mn (SSRmn) 的格式 (2/2)

地址: F0100H、F0101H (SSR00) ~ F0106H、F0107H (SSR03) 复位后: 0000H R
 F0140H、F0141H (SSR10)、F0142H、F0143H (SSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn注	PEF mn	OVF mn

FEFmn注	通道 n 帧错误的检测标志
0	没有发生错误。
1	发生错误 (UART 接收时)。
[清除条件]	
• 当给 SIRmn 寄存器的 FECTmn 位写“1”时	
[置位条件]	
• 在 UART 接收结束时没有检测到停止位时	

PEFmn	通道 n 奇偶校验 /ACK 错误的检测标志
0	没有发生错误。
1	发生错误 (UART 接收时) 或者未检测到 ACK (I ² C 发送时)。
[清除条件]	
• 当给 SIRmn 寄存器的 PECTmn 位写“1”时	
[置位条件]	
• 在 UART 接收结束时发送数据的奇偶校验和奇偶校验位不同 (奇偶校验错误) 时	
• 在 I ² C 发送时并且在 ACK 接收时序从属方没有返回 ACK 信号 (未检测到 ACK) 时	

OVFmn	通道 n 溢出错误的检测标志
0	没有发生错误。
1	发生错误。
[清除条件]	
• 当给 SIRmn 寄存器的 OVCTmn 位写“1”时	
[置位条件]	
• 在 SCRmn 寄存器的 RXEmn 位为“1” (各通信模式中的接收模式、发送和接收模式) 的状态下, 虽然接收数据被保存在 SDRmn 寄存器, 但是没有读接收数据而写发送数据或者写下一个接收数据时	
• 在 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、SSR03、SSR11 寄存器。

注意 1. 如果在 BFFmn 位为“1”时写 SDRmn 寄存器, 就会破坏被保存的发送或者接收数据, 并且检测到溢出错误 (OVEmn=1)。

2. 在 SNOOZE 模式 (SWC0=1) 中进行 CSI 接收时, OVFmn 标志不变。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.3.8 串行通道开始寄存器 m (SSm)

SSm 寄存器是设定允许各通道的通信 / 开始计数的触发寄存器。

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

通过 16 位存储器操作指令设定 SSm 寄存器。

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

在产生复位信号后，SSm 寄存器的值变为“0000H”。

图 11-12 串行通道开始寄存器 m (SSm) 的格式

地址:	F0122H、F0123H (SS0)				复位后: 0000H				R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

地址:	F0162H、F0163H (SS1)				复位后: 0000H				R/W							
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	通道 n 运行开始的触发
0	没有触发。
1	将 SEmn 位置“1”，转移到通信待机状态注。

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

注意 1. 必须将 SS0 寄存器的 bit15 ~ 4 和 SS1 寄存器的 bit15 ~ 2 置“0”。

2. 在 UART 接收时，必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 f_{MCK} 时钟，然后将 SSmn 置“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

2. SSm 寄存器的读取值总是“0000H”。

11.3.9 串行通道停止寄存器 m (STm)

STm 寄存器是设定允许各通道的通信 / 停止计数的触发寄存器。

如果给各位 (STmn) 写“1”，就将串行通道允许状态寄存器 m (SEm) 的对应位 (SEmn) 清“0” (运行停止状态)。因为 STmn 位是触发位，所以如果 SEmn 位为“0”就立即清除 STmn 位。

通过 16 位存储器操作指令设定 STm 寄存器。

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

在产生复位信号后，STm 寄存器的值变为“0000H”。

图 11-13 串行通道停止寄存器 m (STm) 的格式

地址: F0124H、F0125H (ST0) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

地址: F0164H、F0165H (ST1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10

STmn	通道 n 运行的停止触发
0	没有触发。
1	将 SEmn 位清“0”，停止通信运行注。

注 控制寄存器和移位寄存器的值、SCKmn 引脚和 SOmn 引脚以及 FEFmn 标志、PEFmn 标志和 OVFmn 标志保持状态。

注意 必须将 ST0 寄存器的 bit15 ~ 4 和 ST1 寄存器的 bit15 ~ 2 置“0”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

2. STm 寄存器的读取值总是“0000H”。

11.3.10 串行通道允许状态寄存器 m (SEm)

SEm 寄存器用于确认各通道的串行发送和接收的允许或者停止状态。

如果给串行开始允许寄存器 m (SSm) 的各位写“1”，就将其对应位置“1”。如果给串行通道停止寄存器 m (STm) 的各位写“1”，就将其对应位清“0”。

对于允许运行的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOM) 的 CKOmn 位（通道 n 的串行时钟输出）的值，而从串行时钟引脚输出由通信运行反映的值。

对于停止运行的通道 n，能通过软件设定 SOM 寄存器的 CKOmn 位的值，并且从串行时钟引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令读取 SEm 寄存器。

能用 SEmL 并且通过 1 位或者 8 位存储器操作指令读取 SEm 寄存器的低 8 位。

在产生复位信号后，SEm 寄存器的值变为“0000H”。

图 11-14 串行通道允许状态寄存器 m (SEm) 的格式

地址: F0120H、F0121H (SE0) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

地址: F0160H、F0161H (SE1) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE10

SEmn	通道 n 运行的允许或者停止状态的表示
0	运行停止状态
1	运行允许状态

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3)

11.3.11 串行输出允许寄存器 m (SOEm)

SOEm 寄存器设定允许或者停止各通道的串行通信的输出。

对于允许串行输出的通道 n，无法通过软件改写后述的串行输出寄存器 m (SOm) 的 SOmn 位的值，而从串行数据输出引脚输出由通信运行反映的值。

对于停止串行输出的通道 n，能通过软件设定 SOm 寄存器的 SOmn 位的值，并且从串行数据输出引脚输出该值。从而，能通过软件生成开始条件或者停止条件等的任意波形。

通过 16 位存储器操作指令设定 SOEm 寄存器。

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

在产生复位信号后，SOEm 寄存器的值变为“0000H”。

图 11-15 串行输出允许寄存器 m (SOEm) 的格式

地址: F012AH、F012BH (SOE0) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00

地址: F016AH、F016BH (SOE1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE11	SOE10

SOEmn	通道 n 串行输出的允许或者停止														
0	停止串行通信的输出。														
1	允许串行通信的输出。														

注意 必须将 SOE0 寄存器的 bit15 ~ 4 和 SOE1 寄存器的 bit15 ~ 2 置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.3.12 串行输出寄存器 m (SOm)

SOm 寄存器是各通道串行输出的缓冲寄存器。

从通道 n 的串行数据输出引脚输出此寄存器的 SOmn 位的值。

从通道 n 的串行时钟输出引脚输出此寄存器的 CKOmn 位的值。

只有在禁止串行输出时 (SOEmn=0) 才能通过软件改写此寄存器的 SOmn 位。当允许串行输出 (SOEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的 SOmn 位的值。

只有在通道停止运行时 (SEmn=0) 才能通过软件改写此寄存器的 CKOmn 位。当允许通道运行 (SEmn=1) 时, 忽视通过软件的改写而只能通过串行通信更改此寄存器的 CKOmn 位的值。

要将串行接口的引脚用作端口功能时, 必须将相应的 CKOmn 位和 SOmn 位置“1”。

通过 16 位存储器操作指令设定 SOm 寄存器。

在产生复位信号后, SOm 寄存器的值变为“0F0FH”。

图 11-16 串行输出寄存器 m (SOm) 的格式

地址: F0128H、F0129H (SO0) 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

地址: F0168H、F0169H (SO1) 复位后: 0303H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	0	0	CKO 11	CKO 10	0	0	0	0	0	0	SO 11	SO 10

CKOmn	通道 n 的串行时钟输出
0	串行时钟的输出值为“0”。
1	串行时钟的输出值为“1”。

SOmn	通道 n 的串行数据输出
0	串行数据的输出值为“0”。
1	串行数据的输出值为“1”。

注意 必须将 SO0 寄存器的 bit15 ~ 12 和 bit7 ~ 4 置“0”。
必须将 SO1 寄存器的 bit15 ~ 10 和 bit7 ~ 2 置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)

11.3.13 串行输出电平寄存器 m (SOLm)

SOLm 寄存器是设定各通道的数据输出电平反相的寄存器。

只有在 UART 模式中才能设定此寄存器。在 CSI 模式和简易 I²C 模式中，必须将对应位置“0”。

只在允许串行输出时 (SOEmn=1)，将此寄存器的各通道 n 反相设定反映到引脚输出。在禁止串行输出时 (SOEmn=0)，将 SOMn 位的值直接输出。

禁止在运行过程中 (SEmn=1) 改写 SOLm 寄存器。

通过 16 位存储器操作指令设定 SOLm 寄存器。

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

在产生复位信号后，SOLm 寄存器的值变为“0000H”。

图 11-17 串行输出电平寄存器 m (SOLm) 的格式

地址: F0134H、F0135H (SOL0) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

地址: F0174H、F0175H (SOL1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 10

SOLmn	UART 模式中的通道 n 发送数据电平反相的选择
0	将通信数据直接输出。
1	将通信数据反相输出。

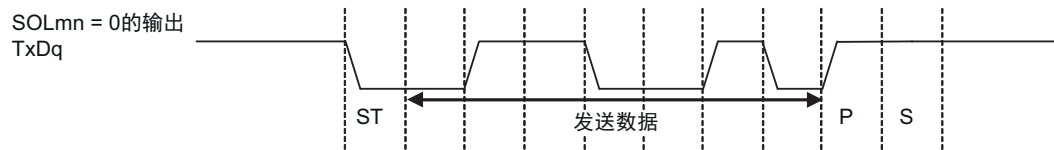
注意 必须将 SOL0 寄存器的 bit15 ~ 3、bit1 和 SOL1 寄存器的 bit15 ~ 1 置“0”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2)

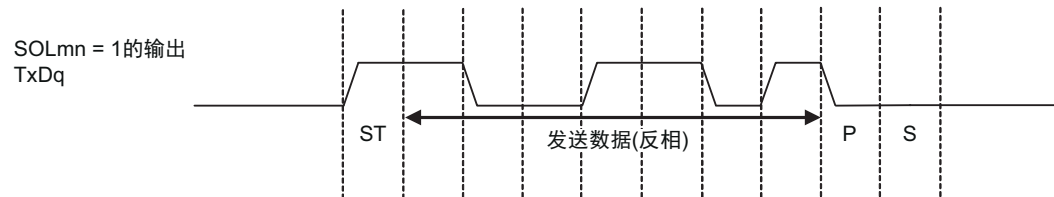
当进行 UART 发送时，发送数据的电平反相例子如图 11-18 所示。

图 11-18 发送数据的电平反相例子

(a) 正相输出 (SOLmn = 0)



(b) 反相输出 (SOLmn = 1)



备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2)

11.3.14 串行待机控制寄存器 0 (SSC0)

SSC0 寄存器是控制在 CSI00 或者 UART0 的串行数据接收时从 STOP 模式启动接收运行 (SNOOZE 模式) 的寄存器。

通过 16 位存储器操作指令设定 SSC0 寄存器。

能用 SSC0L 并且通过 8 位存储器操作指令设定 SSC0 寄存器的低 8 位。

在产生复位信号后, SSC0 寄存器的值变为“0000H”。

注意 SNOOZE 模式中的最大传送速率如下所示:

- CSI00、CSI20: ~1Mbps
- UART0、UART2: 只限于4800bps。

图 11-19 串行待机控制寄存器 0 (SSC0) 的格式

地址: F0138H、F0139H	复位后: 0000H	R/W														
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSEC0	SWC0
															C0	0

SSEC0	SNOOZE 模式中的通信错误中断产生的允许或者停止的选择
0	允许产生错误中断 (INTSRE0)。
1	停止产生错误中断 (INTSRE0)。
<ul style="list-style-type: none"> • 在 SNOOZE 模式中进行 UART 接收的情况下, 只有在 SWC0 位和 EOCmn 位为“1”时才能将 SSEC0 位置“1”或者“0”, 否则必须将 SSEC0 位置“0”。 • 禁止将 SSEC0 位和 SWC0 位分别置“1”和“0”。 	

SWC0	SNOOZE 模式的设定
0	不使用 SNOOZE 模式功能。
1	使用 SNOOZE 模式功能。
<ul style="list-style-type: none"> • 通过 STOP 模式中的硬件触发信号来解除 STOP 模式, 并且在 CPU 不运行的状态下进行 CSI/UART 的接收 (SNOOZE 模式)。 • 只有在选择高速内部振荡器时钟作为 CPU/ 外围硬件时钟 (f_{CLK}) 的情况下才能设定 SNOOZE 模式功能, 而在选择其他时钟的情况下禁止设定。 • 即使使用 SNOOZE 模式功能, 也必须在通常运行模式中将 SWC0 位置“0”, 并且在即将要转移到 STOP 模式前将 SWC0 位改为“1”。 <p>另外, 必须在从 STOP 模式返回到通常运行模式后将 SWC0 位改为“0”。</p>	

图 11-20 在 SNOOZE 模式中进行 UART 接收时的中断

EOCmn 位	SSEC0 位	正常接收	接收错误
0	0	产生 INTSRx。	产生 INTSRx。
0	1	产生 INTSRx。	产生 INTSRx。
1	0	产生 INTSRx。	产生 INTSREx。
1	1	产生 INTSRx。	不产生中断。

11.3.15 噪声滤波器允许寄存器 0 (NFEN0)

NFEN0 寄存器设定噪声滤波器是否用于各通道串行数据输入引脚的输入信号。

对于用于 CSI 或者简易 I²C 通信的引脚，必须将对应位置“0”，使噪声滤波器无效。

对于用于 UART 通信的引脚，必须将对应位置“1”，使噪声滤波器有效。

当噪声滤波器有效时，在通过对象通道的运行时钟 (f_{MCK}) 进行同步后检测 2 个时钟是否一致；当噪声滤波器无效时，只通过对象通道的运行时钟 (f_{MCK}) 进行同步。

通过 1 位或者 8 位存储器操作指令设定 NFEN0 寄存器。

在产生复位信号后，NFEN0 寄存器的值变为“00H”。

图 11-21 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	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	RxD2 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON
当用作 RxD2 引脚时，必须将 SNFEN20 位置“1”。	
当用作 RxD2 引脚以外的功能时，必须将 SNFEN20 位置“0”。	

SNFEN10	RxD1 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON
当用作 RxD1 引脚时，必须将 SNFEN10 位置“1”。	
当用作 RxD1 引脚以外的功能时，必须将 SNFEN10 位置“0”。	

SNFEN00	RxD0 引脚的噪声滤波器的使用与否
0	噪声滤波器 OFF
1	噪声滤波器 ON
当用作 RxD0 引脚时，必须将 SNFEN00 位置“1”。	
当用作 RxD0 引脚以外的功能时，必须将 SNFEN00 位置“0”。	

注意 必须将 bit7 ~ 5、3、1 置“0”。

11.3.16 控制串行输入 / 输出引脚端口功能的寄存器

在使用串行阵列单元时，必须设定与对象通道复用的端口功能的控制寄存器（端口模式寄存器（PM_{xx}）、端口寄存器（P_{xx}）、端口输入模式寄存器（PIM_{xx}）、端口输出模式寄存器（POM_{xx}）和端口模式控制寄存器（PMC_{xx}））。

详细内容请参照“4.3.1 端口模式寄存器（PM_{xx}）”、“4.3.2 端口寄存器（P_{xx}）”、“4.3.4 端口输入模式寄存器（PIM_{xx}）”、“4.3.5 端口输出模式寄存器（POM_{xx}）”和“4.3.6 端口模式控制寄存器（PMC_{xx}）”。

在将串行数据输出引脚或者串行时钟输出引脚的复用端口（P00/ANI17/TI00/TxD1 等）用作串行数据输出或者串行时钟输出时，必须将各端口对应的端口模式控制寄存器（PMC_{xx}）的位和端口模式寄存器（PM_{xx}）的位置“0”，并且将端口寄存器（P_{xx}）的位置“1”。

另外，当用于N沟道漏极开路输出（V_{DD}耐压）模式时，必须将各端口对应的端口输出模式寄存器（POM_{xx}）的位置“1”。在连接不同电位（1.8V、2.5V、3V）工作的外部设备时，请参照“4.4.4 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应”。

（例）P00/ANI17/TI00/TxD1 用作串行数据输出的情况

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/ANI16/TO00/RxD1 等）用作串行数据输入或者串行时钟输入时，必须将各端口对应的端口模式寄存器（PM_{xx}）的位置“1”，并且将端口模式控制寄存器（PMC_{xx}）的位置“0”。此时，端口寄存器（P_{xx}）的位可以是“0”或者“1”。

另外，当用作 TTL 输入缓冲器时，必须将各端口对应的端口输入模式寄存器（PIM_{xx}）的位置“1”。在连接不同电位（1.8V、2.5V、3V）工作的外部设备时，请参照“4.4.4 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应”。

（例）P01/ANI16/TO00/RxD1 用作串行数据输入的情况

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”。

将端口寄存器 0 的 P01 位置“0”或者“1”。

11.4 运行停止模式

串行阵列单元的各串行接口有运行停止模式。

在运行停止模式中不能进行串行通信，因此能降低功耗。

另外，在运行停止模式中能将用于串行接口的引脚用作端口功能。

11.4.1 以单元为单位停止运行的情况

通过外围允许寄存器 0（PER0）设定以单元为单位的停止运行。

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

要停止串行阵列单元 0 时，必须将 bit2（SAU0EN）置“0”；要停止串行阵列单元 1 时，必须将 bit3（SAU1EN）置“0”。

图 11-22 以单元为单位停止运行时的外围允许寄存器 0（PER0）的设定

(a) 外围允许寄存器 0（PER0）..... 只将要停止 SAUm 的对应位置“0”。

	7	6	5	4	3	2	1	0
PER0	TMKAEN		ADCEN	IICA0EN	SAU1EN	SAU0EN		TAU0EN
	×	0	×	×	0/1	0/1	0	×

SAUm 输入时钟的控制

0: 停止提供输入时钟

1: 提供输入时钟

注意 1. 当 SAUmEN 位为“0”时，忽视串行阵列单元 m 的控制寄存器的写操作，而且读取值都为初始值。

但是，以下的寄存器除外：

- 噪声滤波器允许寄存器 0（NFEN0）
- 端口输入模式寄存器 0、1（PIM0、PIM1）
- 端口输出模式寄存器 0、1、5（POM0、POM1、POM5）
- 端口模式控制寄存器 0（PMC0）
- 端口模式寄存器 0、1、3、5（PM0、PM1、PM3、PM5）
- 端口寄存器 0、1、3、5（P0、P1、P3、P5）

2. 必须将 bit6 和 bit1 置“0”。

备注 ×：这是串行阵列单元未使用的位（取决于其他外围功能的设定）。

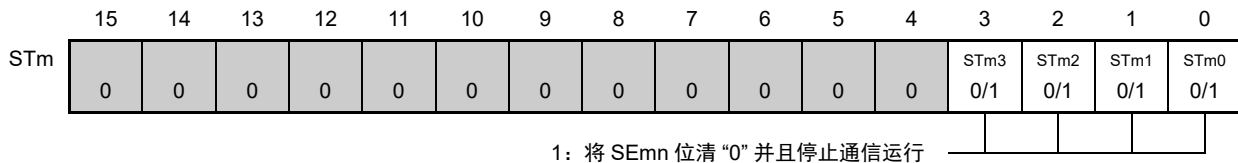
0/1：根据用户的用途置“0”或者“1”。

11.4.2 按通道停止运行的情况

通过以下各寄存器设定按通道停止运行。

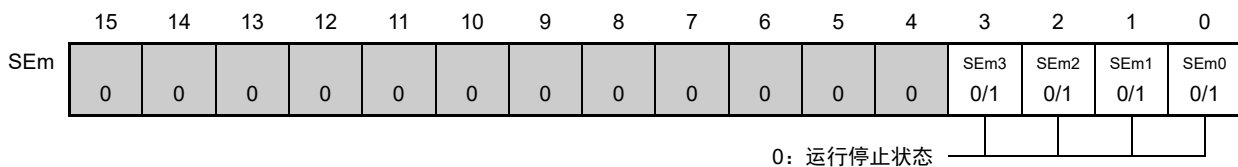
图 11-23 按通道停止运行时的各寄存器的设定

(a) 串行通道停止寄存器 m (STm) 这是设定允许各通道的通信 / 停止计数的寄存器。



※ 因为 ST_mn 位是触发位，所以如果 SE_mn 位为“0”就立即清除 ST_mn 位。

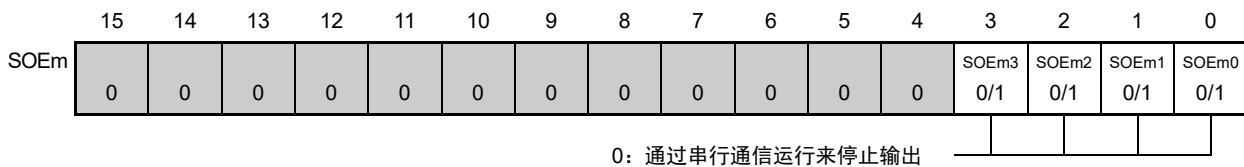
(b) 串行通道允许状态寄存器 m (SEm) 此寄存器表示各通道的数据发送和接收的运行或者停止状态。



※SE_m 寄存器是只读状态寄存器，通过 ST_m 寄存器来停止运行。

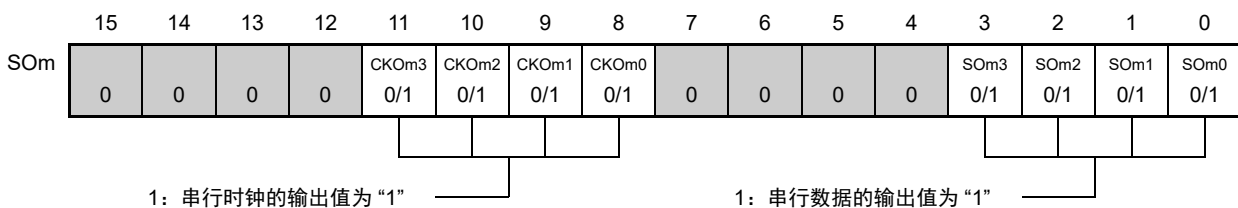
对于已经停止运行的通道，能通过软件设定 SO_m 寄存器的 CKO_mn 位的值。

(c) 串行输出允许寄存器 m (SOEm) 这是设定允许或者停止各通道串行通信输出的寄存器。



※ 对于已经停止串行输出的通道，能通过软件设定 SO_m 寄存器的 SO_mn 位的值。

(d) 串行输出寄存器 m (SOm) 这是各通道串行输出的缓冲寄存器。



※ 当将各通道对应的引脚用作端口功能时，必须将相应的 CKO_mn 位和 SO_mn 位置“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3)

2. : 不能设定 (设定初始值)。 0/1: 根据用户的用途置“0”或者“1”。

11.5 3 线串行 I/O (CSI00、CSI11、CSI20) 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]

- 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: $\text{Max. } f_{\text{CLK}}/2$ (只限于 CSI00)
 $\text{Max. } f_{\text{CLK}}/4$

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

以下通道的 CSI 支持 SNOOZE 模式。SNOOZE 模式功能是指：如果在 STOP 模式的状态下检测到 SCK 的输入，就不需要 CPU 运行而接收数据。只有以下的 CSI 才能设定 SNOOZE 模式。

- 本产品：CSI00

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 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”。

SAU0 的通道 0 ~ 3 以及 SAU1 的通道 0 ~ 3 是支持 3 线串行 I/O (CSI00、CSI11、CSI20) 的通道。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

3 线串行 I/O (CSI00、CSI11、CSI20) 有以下 7 种通信运行：

- 主控发送 (参照 11.5.1)
- 主控接收 (参照 11.5.2)
- 主控的发送和接收 (参照 11.5.3)
- 从属发送 (参照 11.5.4)
- 从属接收 (参照 11.5.5)
- 从属的发送和接收 (参照 11.5.6)
- SNOOZE 模式功能 (参照 11.5.7)

11.5.1 主控发送

主控发送是指 RL78 微控制器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	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SO00	SCK11、SO11	SCK20、SO20
中断	INTCSI00	INTCSI11	INTCSI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	无		
传送数据长度	7 位或者 8 位		
传送速率注	Max. $f_{CLK}/2$ [Hz]（只限于 CSI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : 系统时钟频率		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时，开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。 		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 		
数据方向	MSB 优先或者 LSB 优先		

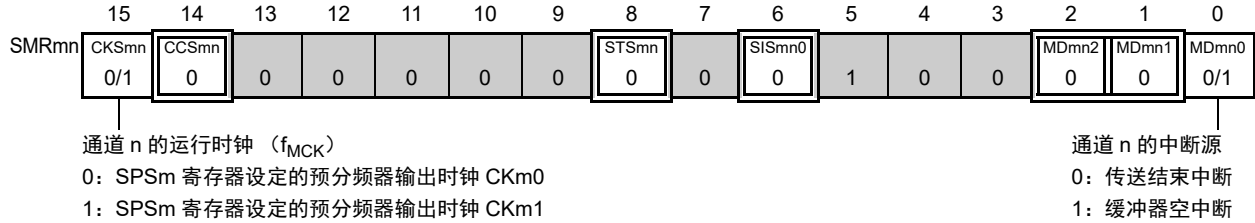
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性（ $T_A = -40 \sim +85^\circ\text{C}$ ）”）的范围内使用。

备注 m: 单元号（m=0、1） n: 通道号（n=0～3） mn=00、03、10

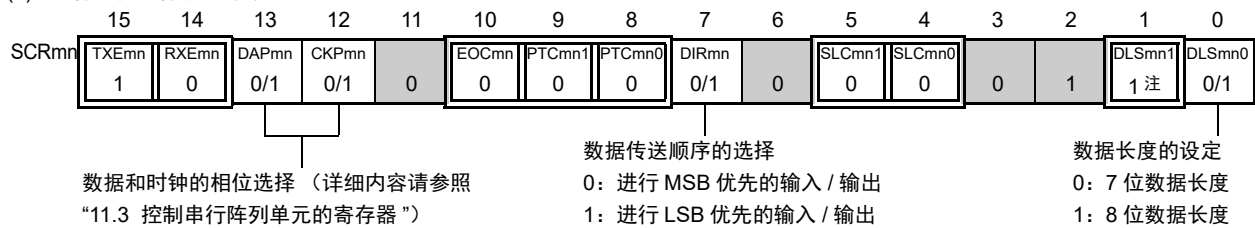
(1) 寄存器的设定

图 11-24 3 线串行 I/O (CSI00、CSI11、CSI20) 主控发送时的寄存器设定内容例子

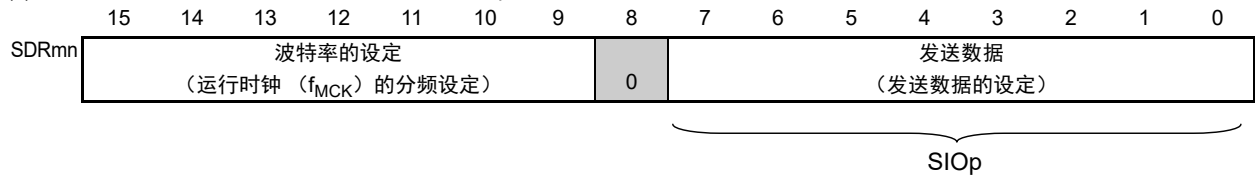
(a) 串行模式寄存器 mn (SMRmn)



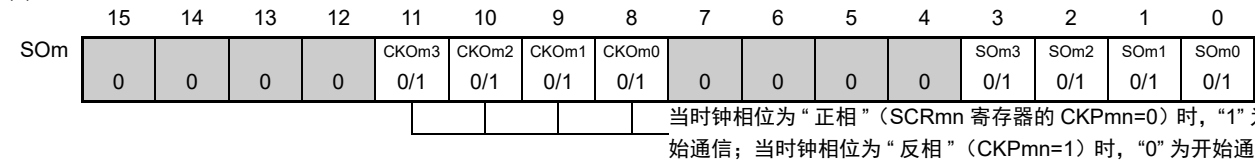
(b) 串行通信运行设定寄存器 mn (SCRmn)



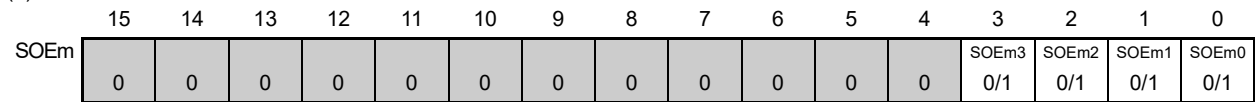
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



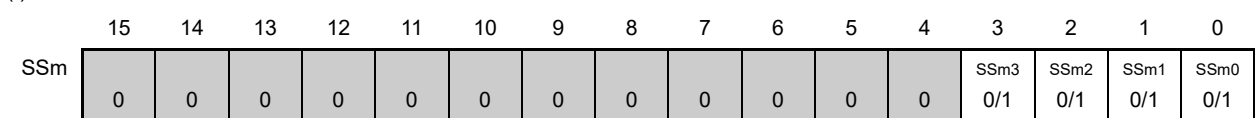
(d) 串行输出寄存器 m (SOM) 只设定对象通道的位。



(e) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。



(f) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。



注 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

2. □: 在 CSI 主控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”。

(2) 操作步骤

图 11-25 主控发送的初始设定步骤

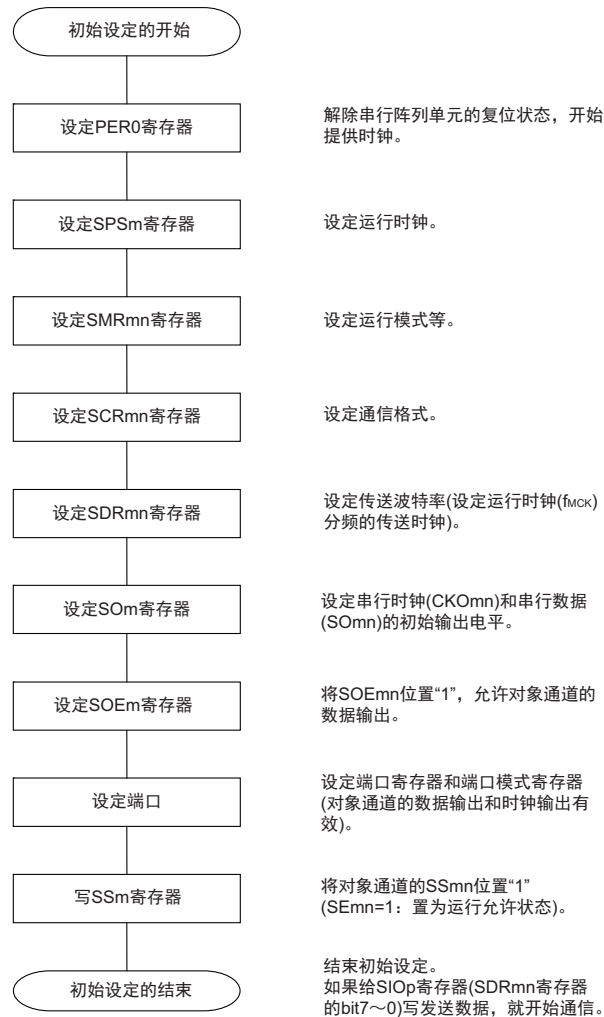


图 11-26 主控发送的中止步骤

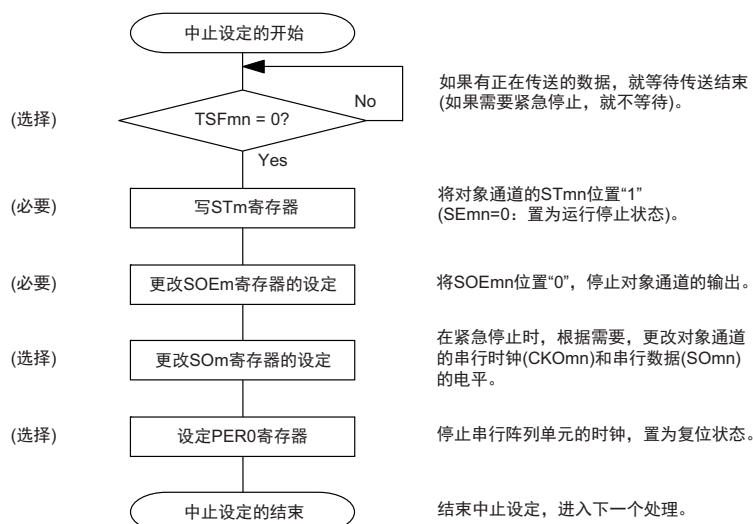


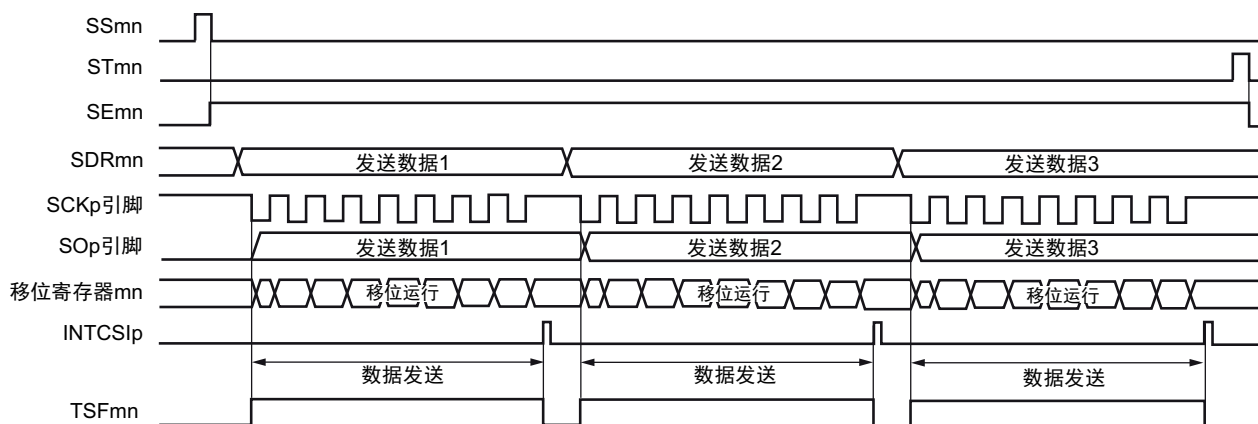
图 11-27 重新开始主控发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象(从属设备)停止或者通信结束后进行初始设定而不是进行重新开始的设定。

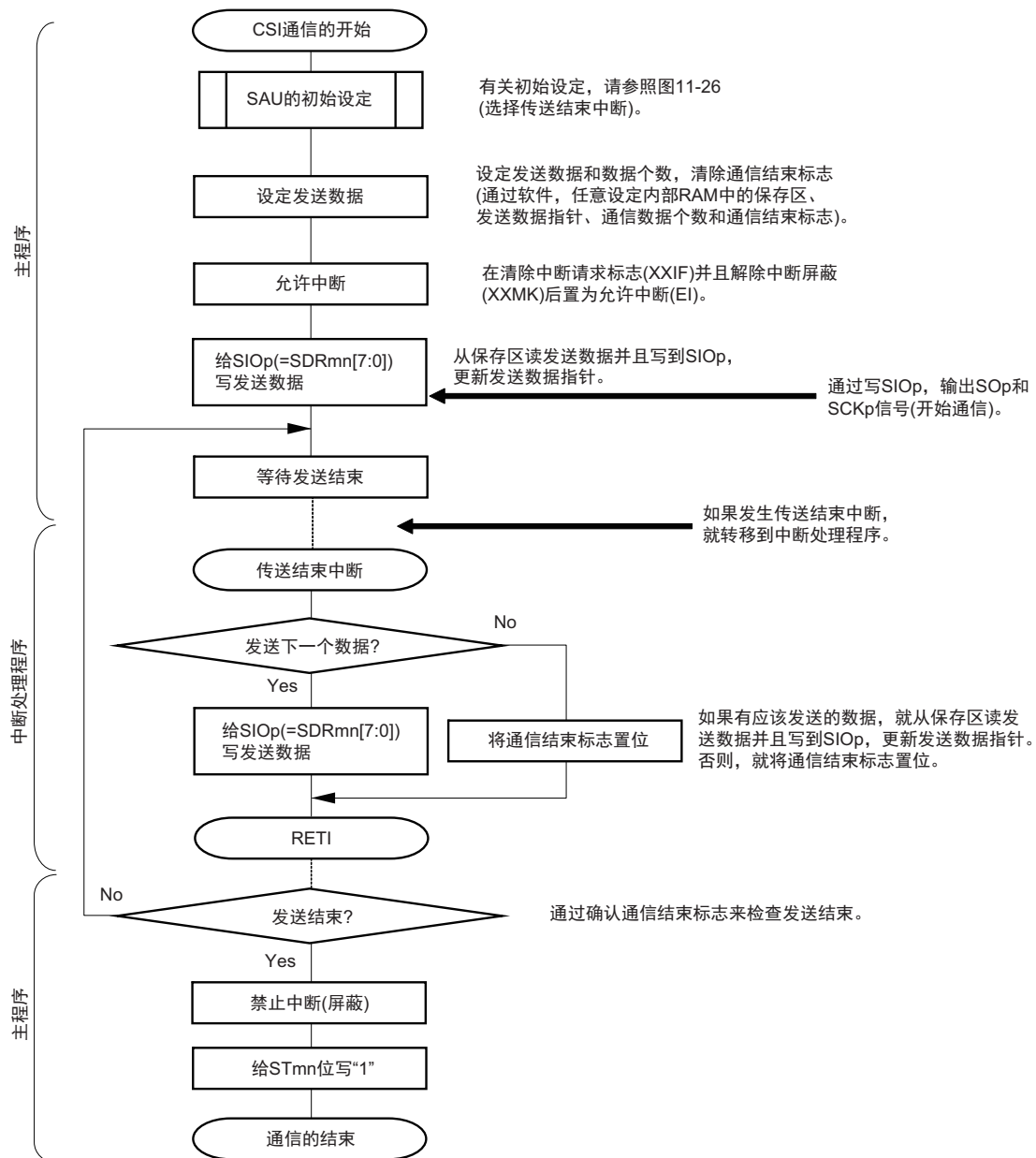
(3) 处理流程（单次发送模式）

图 11-28 主控发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



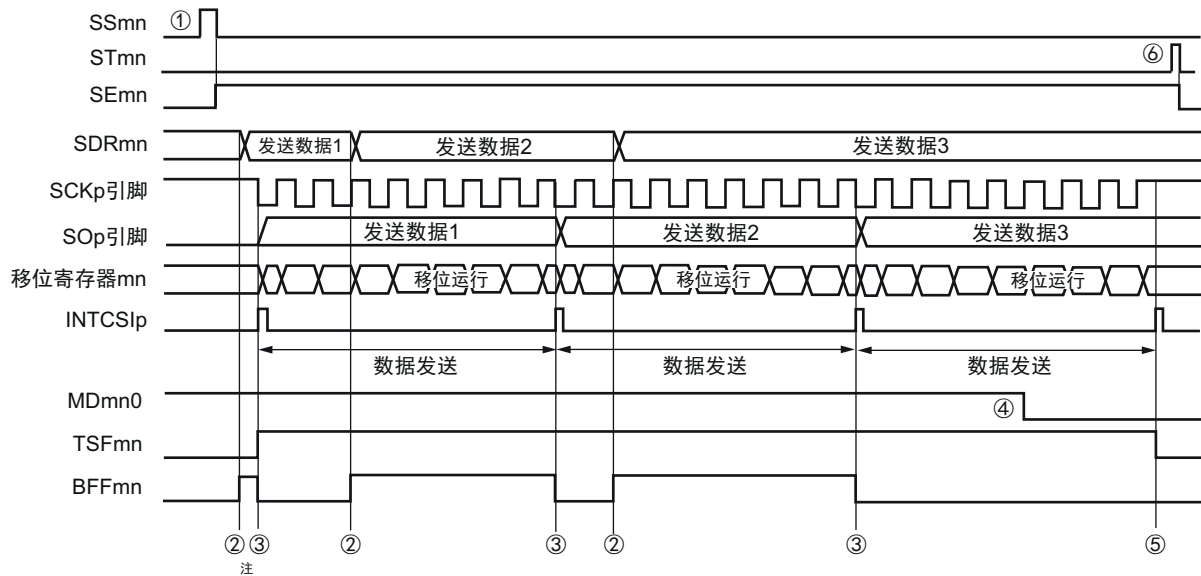
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI号 (p=00、11、20) mn=00、03、10

图 11-29 主控发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 11-30 主控发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

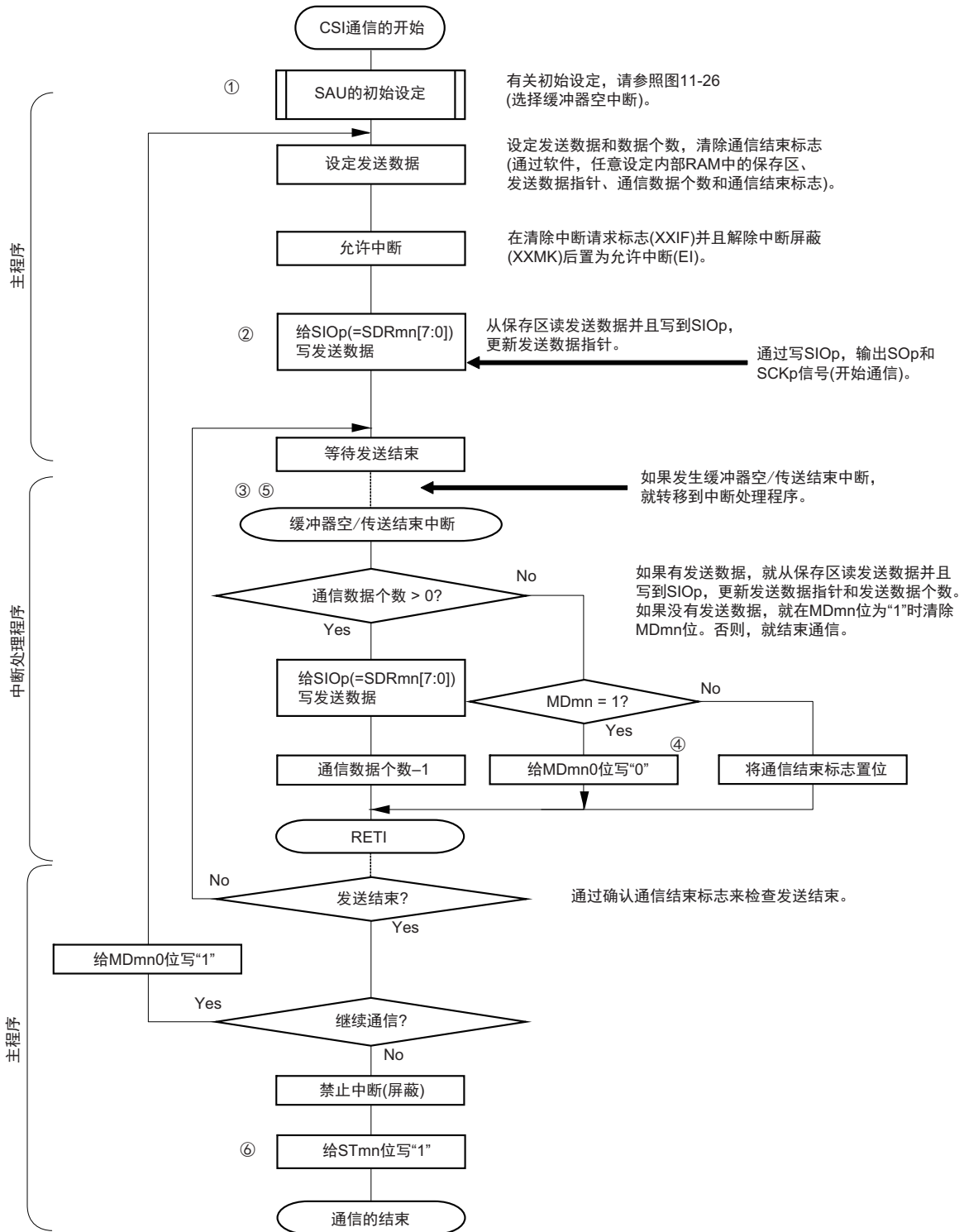


注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

图 11-31 主控发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 11-30 主控发送（连续发送模式）的时序图（类型 1: DAPmn=0、CKPmn=0）”中的①~⑥。

11.5.2 主控接收

主控接收是指 RL78 微控制器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	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SI00	SCK11、SI11	SCK20、SI20
中断	INTCSI00	INTCSI11	INTCSI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	7 位或者 8 位		
传送速率注	Max. $f_{CLK}/2$ [Hz]（只限于 CSI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : 系统时钟频率		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时，开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。 		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 		
数据方向	MSB 优先或者 LSB 优先		

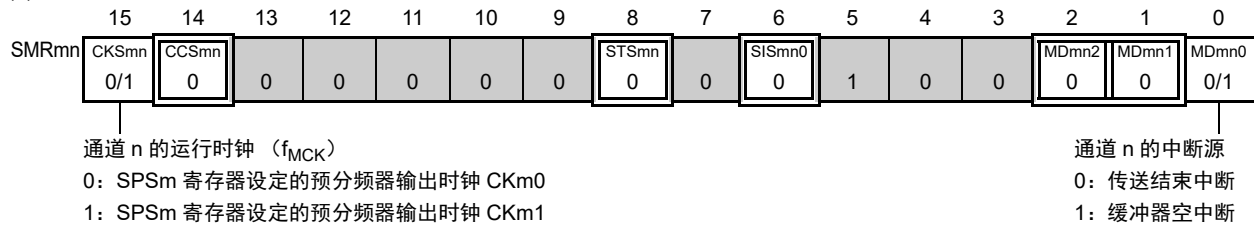
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性（ $T_A = -40 \sim +85^\circ\text{C}$ ）”）的范围内使用。

备注 m: 单元号（m=0、1） n: 通道号（n=0～3） mn=00、03、10

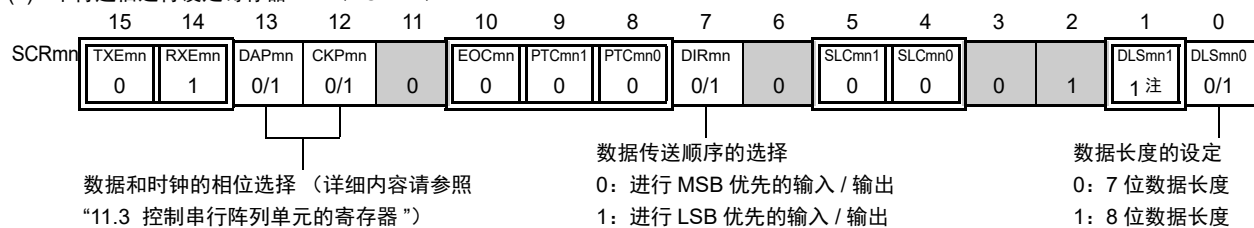
(1) 寄存器的设定

图 11-32 3 线串行 I/O (CSI00、CSI11、CSI20) 主控接收时的寄存器设定内容例子

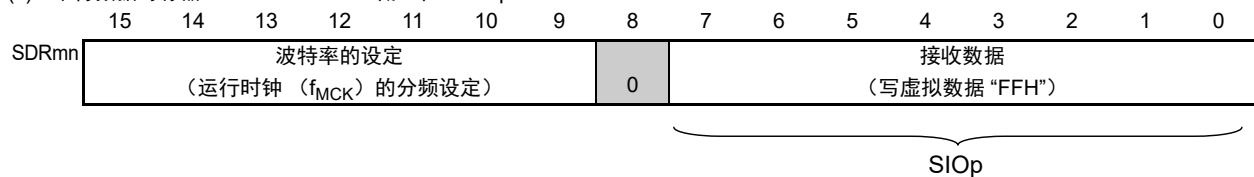
(a) 串行模式寄存器 mn (SMRmn)



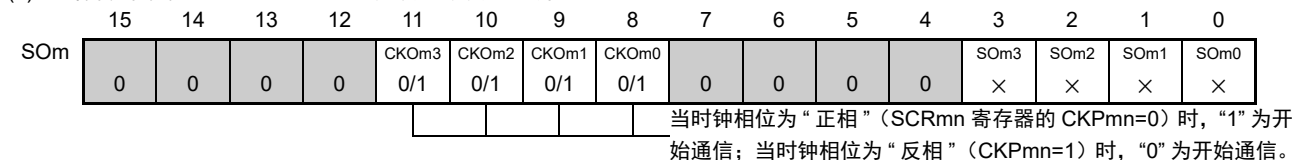
(b) 串行通信运行设定寄存器 mn (SCRmn)



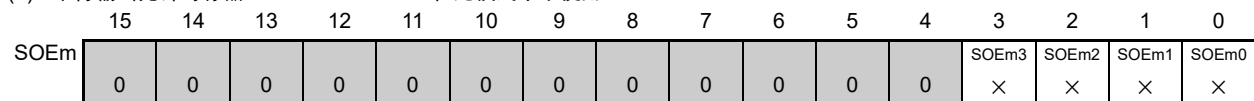
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



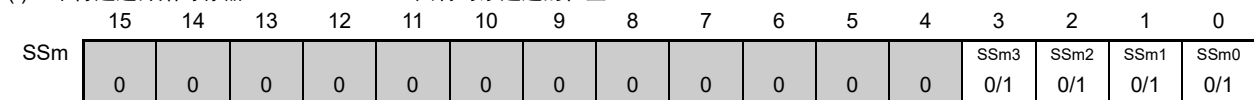
(d) 串行输出寄存器 m (SOm) 只设定对象通道的位。



(e) 串行输出允许寄存器 m (SOEm) 在此模式中不使用。



(f) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。



注 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

- : 在 CSI 主控接收模式中为固定设定。■: 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-33 主控接收的初始设定步骤

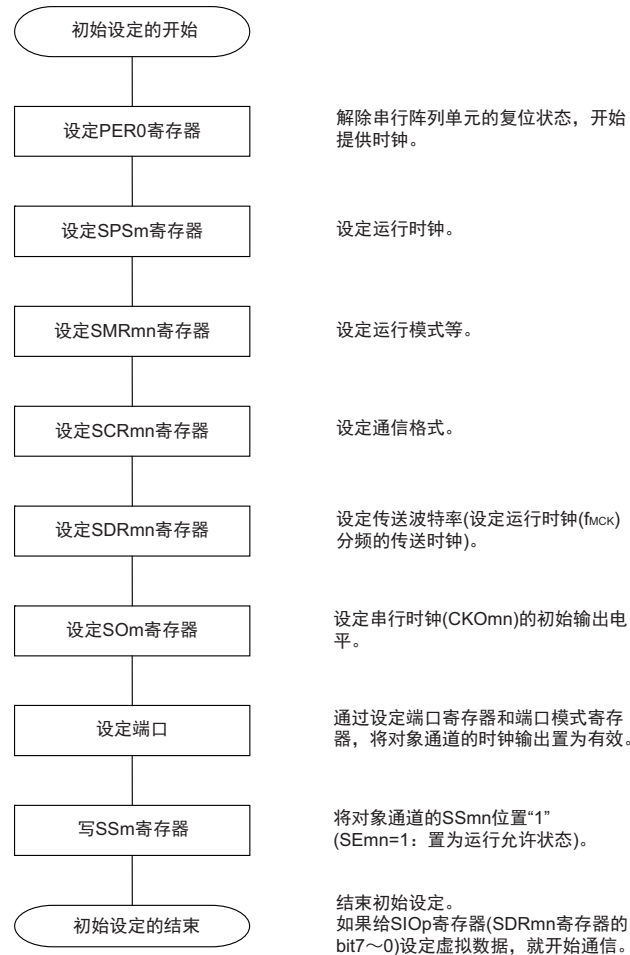


图 11-34 主控接收的中止步骤

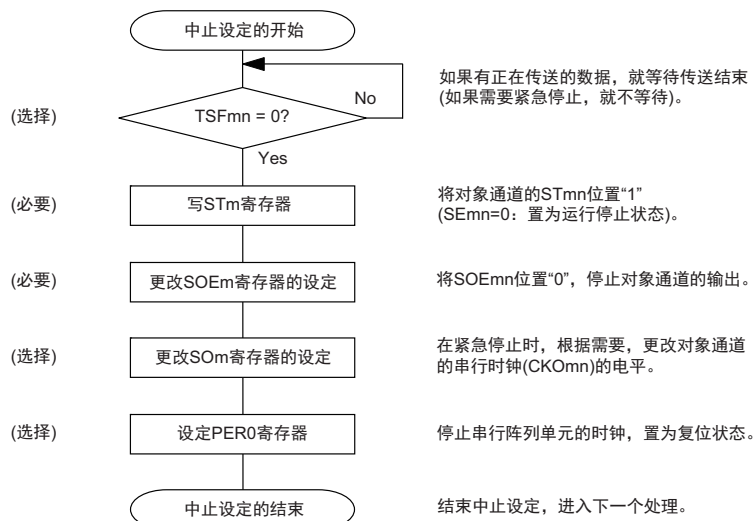
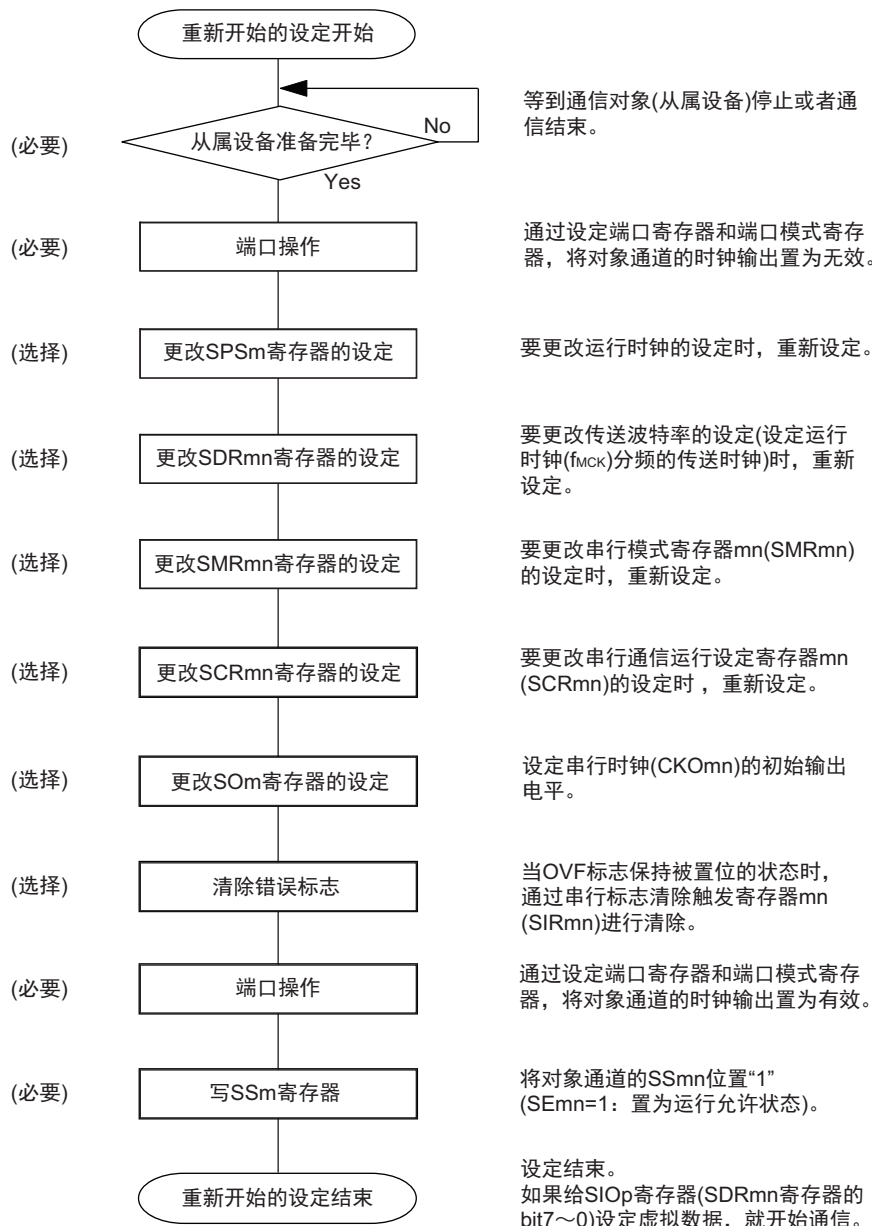


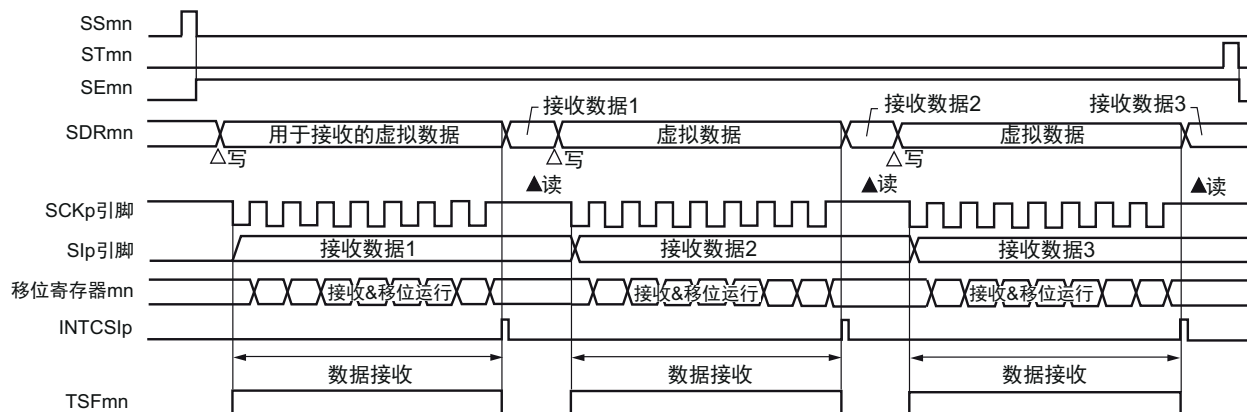
图 11-35 重新开始主控接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (从属设备) 停止或者通信结束后进行初始设定而不是进行重新开始的设定。

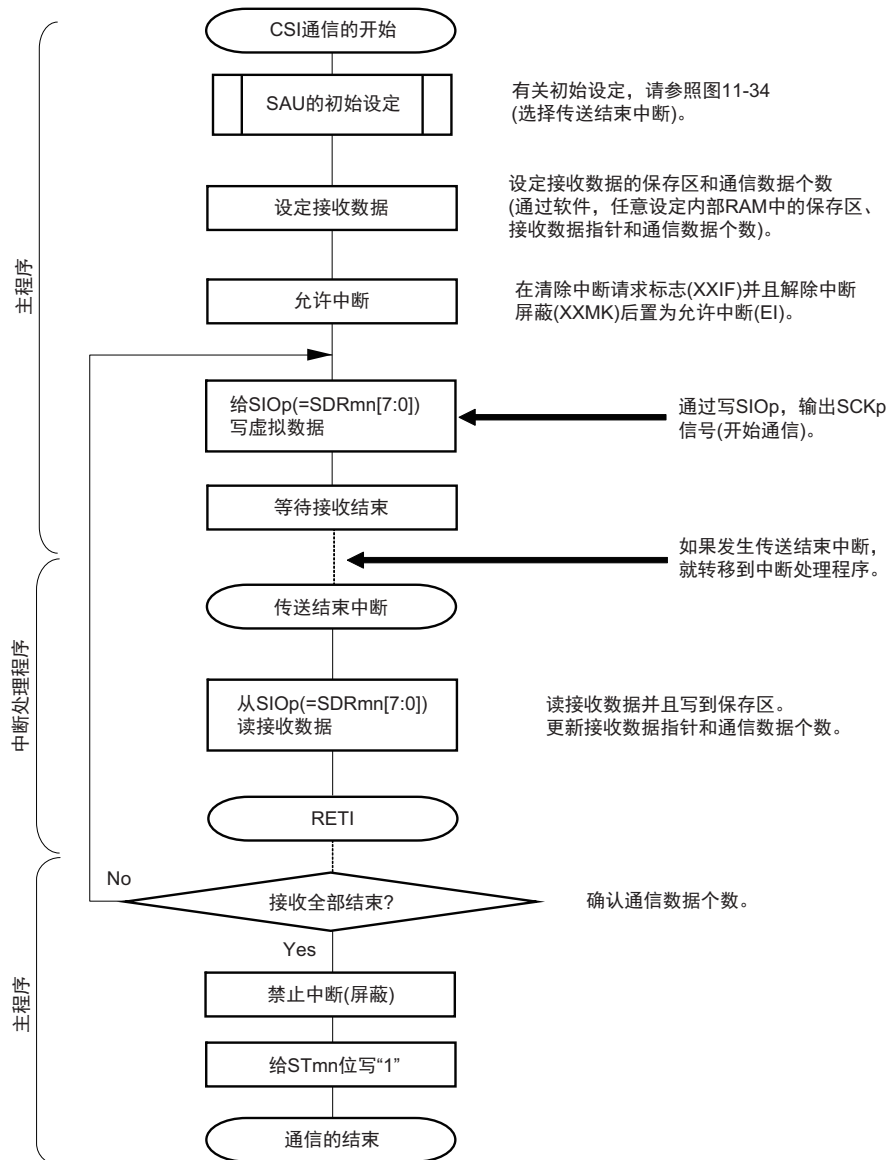
(3) 处理流程（单次接收模式）

图 11-36 主控接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



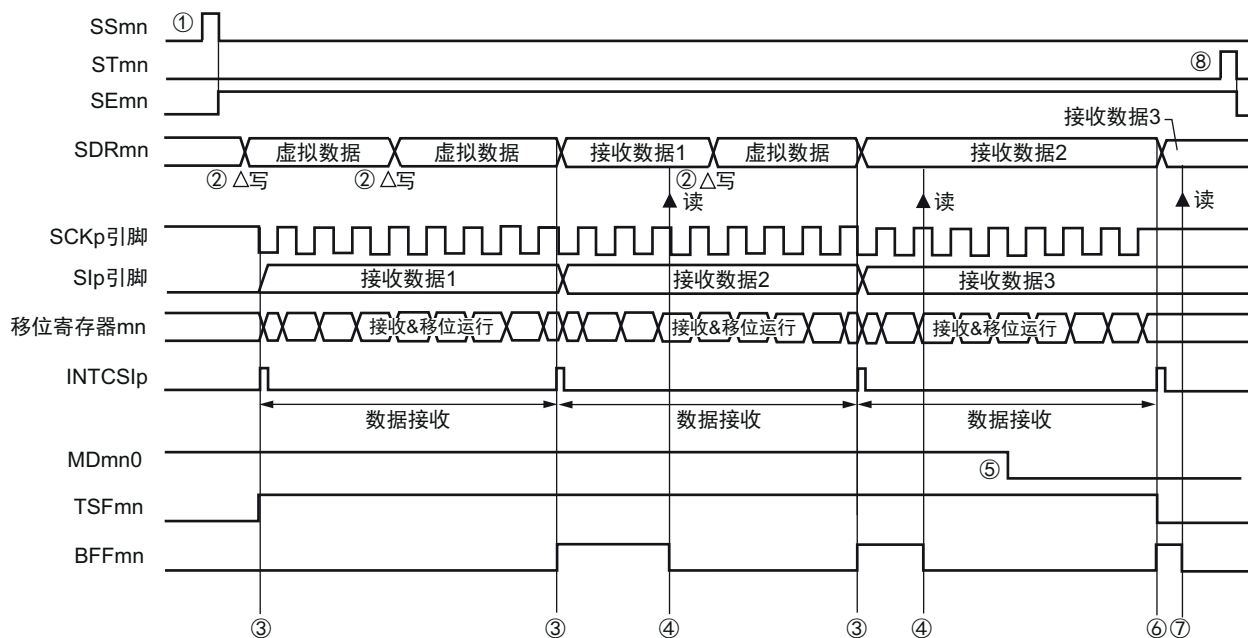
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI号 (p=00、11、20) mn=00、03、10

图 11-37 主控接收（单次接收模式）的流程图



(4) 处理流程（连续接收模式）

图 11-38 主控接收（连续接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

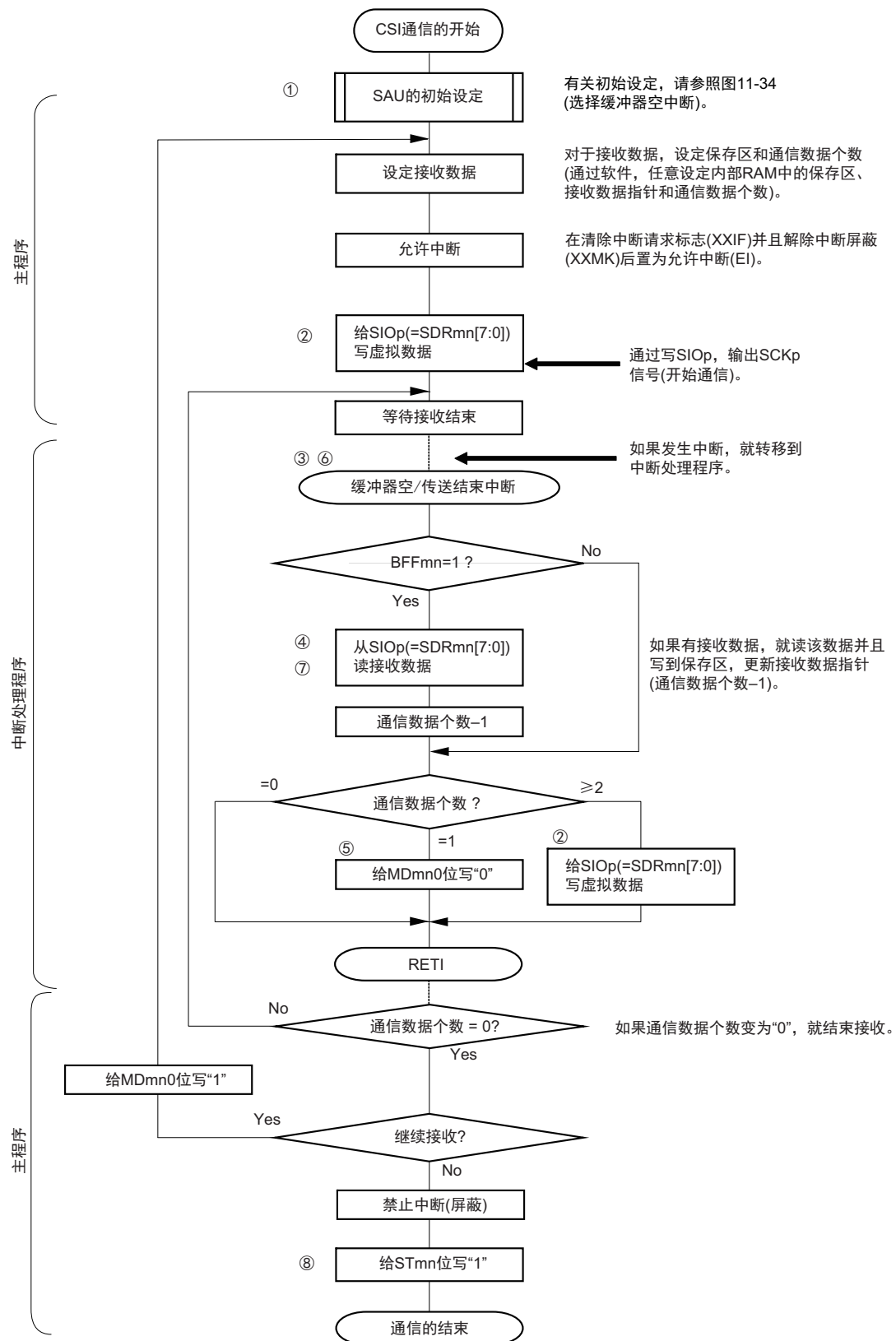


注意 即使在运行过程中也能改写 MDmn0 位。但是，为了能赶上最后接收数据的传送结束中断，必须在开始接收最后一位之前进行改写。

备注 1. 图中的①~⑧对应“图 11-39 主控接收（连续接收模式）的流程图”中的①~⑧。

备注 2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

图 11-39 主控接收（连续接收模式）的流程图



备注 图中的①~⑧对应“图 11-38 主控接收（连续接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）”中的①~⑧。

11.5.3 主控的发送和接收

主控的发送和接收是指 RL78 微控制器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	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SI00、SO00	SCK11、SI11、SO11	SCK20、SI20、SO20
中断	INTCSI00	INTCSI11	INTCSI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	7 位或者 8 位		
传送速率注	Max. $f_{CLK}/2$ [Hz]（只限于 CSI00）， $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : 系统时钟频率		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 • DAPmn=0: 在串行时钟开始运行时，开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前，开始数据输出。		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 • CKPmn=0: 正相 • CKPmn=1: 反相		
数据方向	MSB 优先或者 LSB 优先		

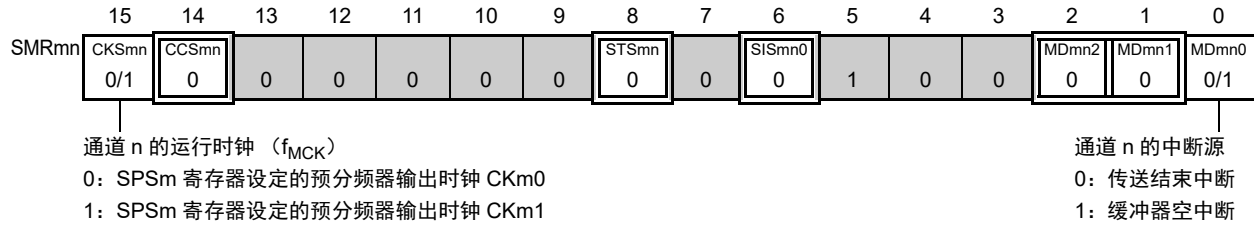
注 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性（ $T_A=-40 \sim +85^\circ\text{C}$ ）”）的范围内使用。

备注 m: 单元号（m=0、1） n: 通道号（n=0～3） mn=00、03、10

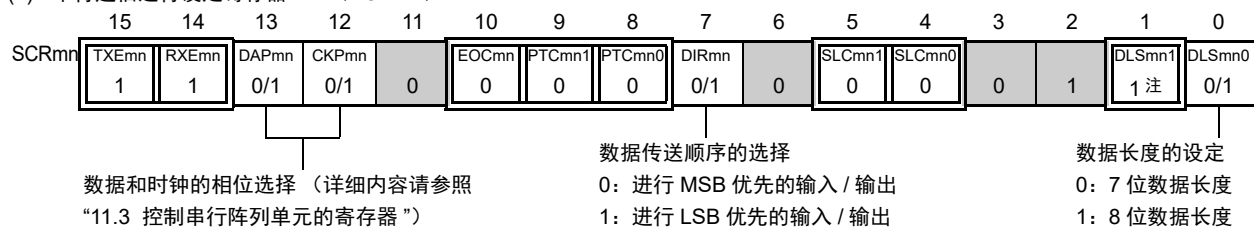
(1) 寄存器的设定

图 11-40 3 线串行 I/O (CSI00、CSI11、CSI20) 主控发送和接收时的寄存器设定内容例子

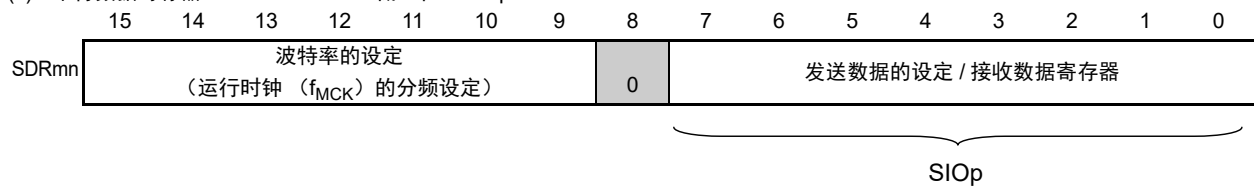
(a) 串行模式寄存器 mn (SMRmn)



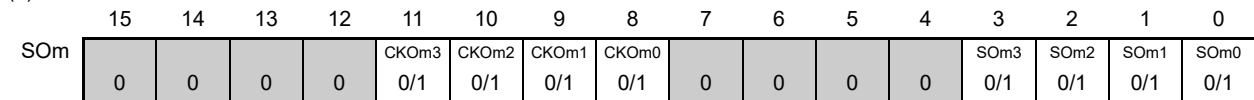
(b) 串行通信运行设定寄存器 mn (SCRmn)



(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)

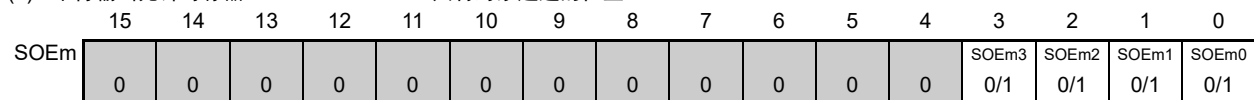


(d) 串行输出寄存器 m (SOM) 只设定对象通道的位。

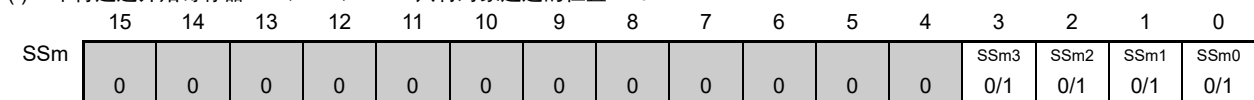


当时钟相位为“正相”(SCRmn 寄存器的 CKPmn=0) 时,“1”为开始通信;当时钟相位为“反相”(CKPmn=1) 时,“0”为开始通信。

(e) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。



(f) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。



注 只限于 SCR00 寄存器和 SCR01 寄存器,其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

2. : 在 CSI 主控发送和接收模式中为固定设定。 : 不能设定 (设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-41 主控发送和接收的初始设定步骤

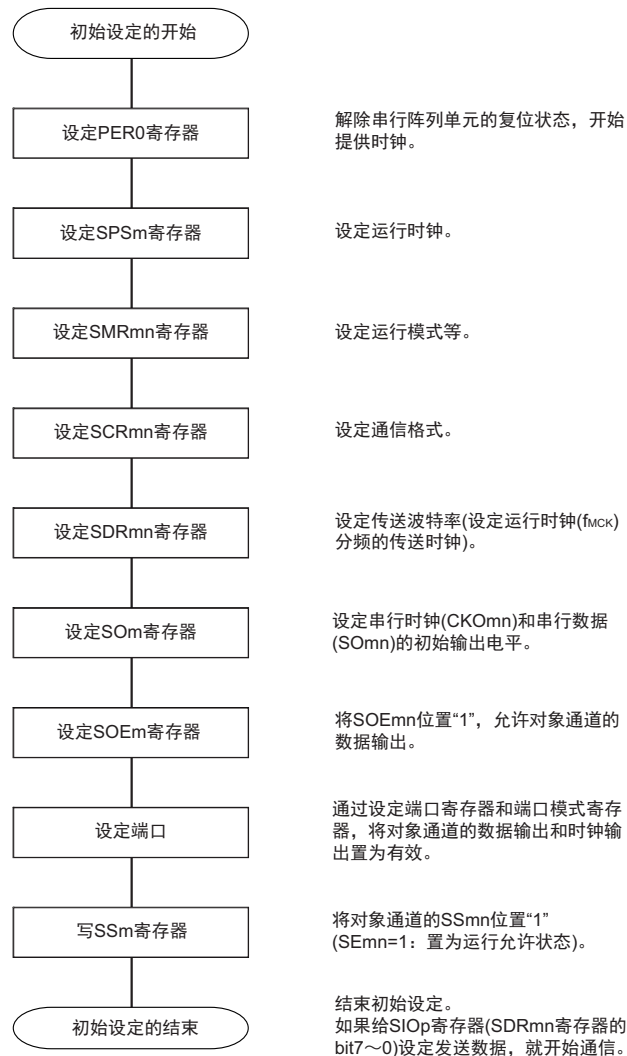


图 11-42 主控发送和接收的中止步骤

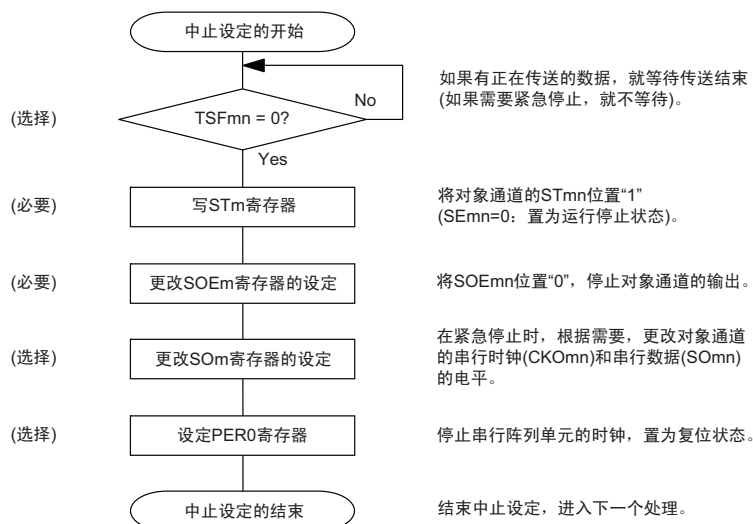
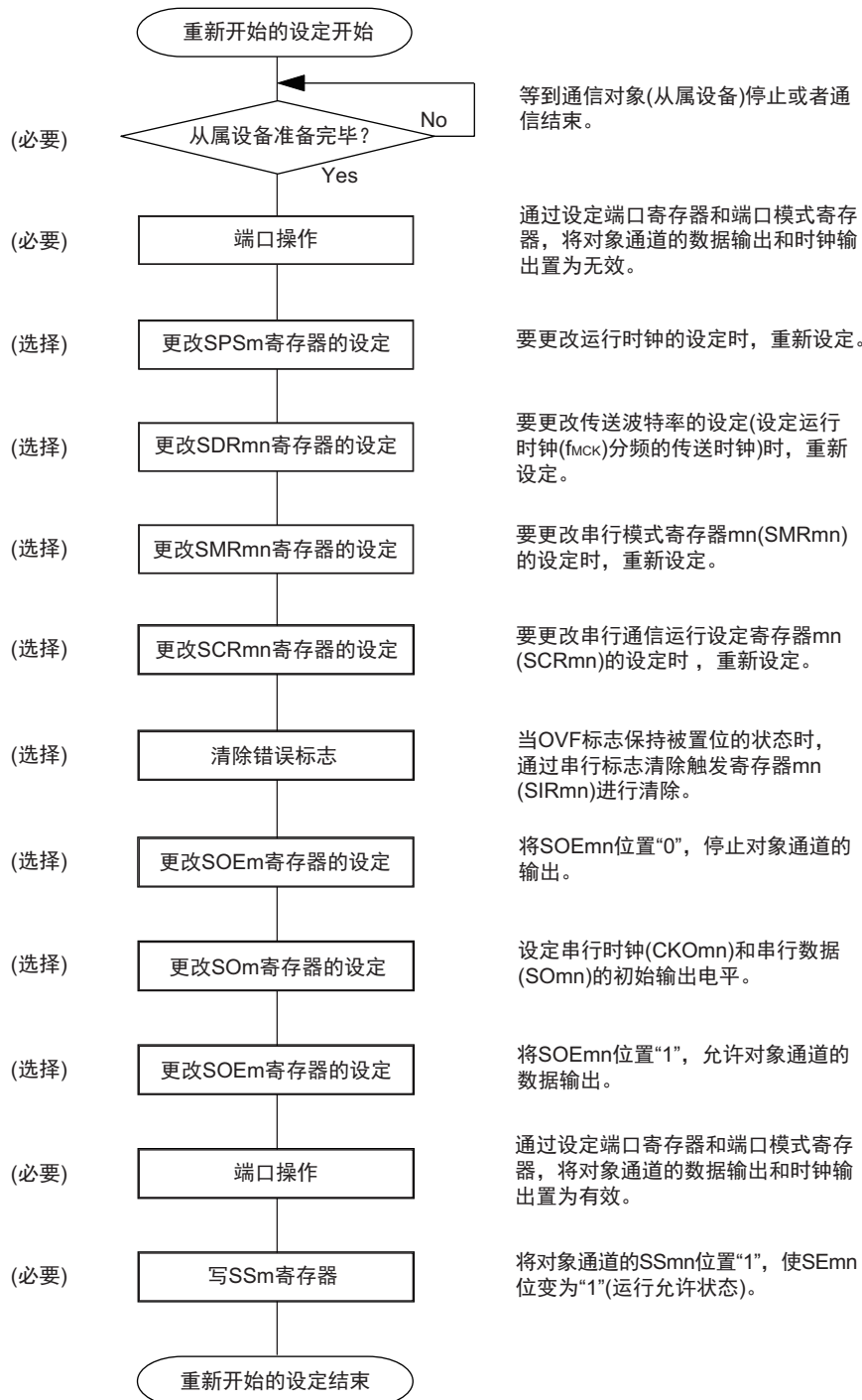
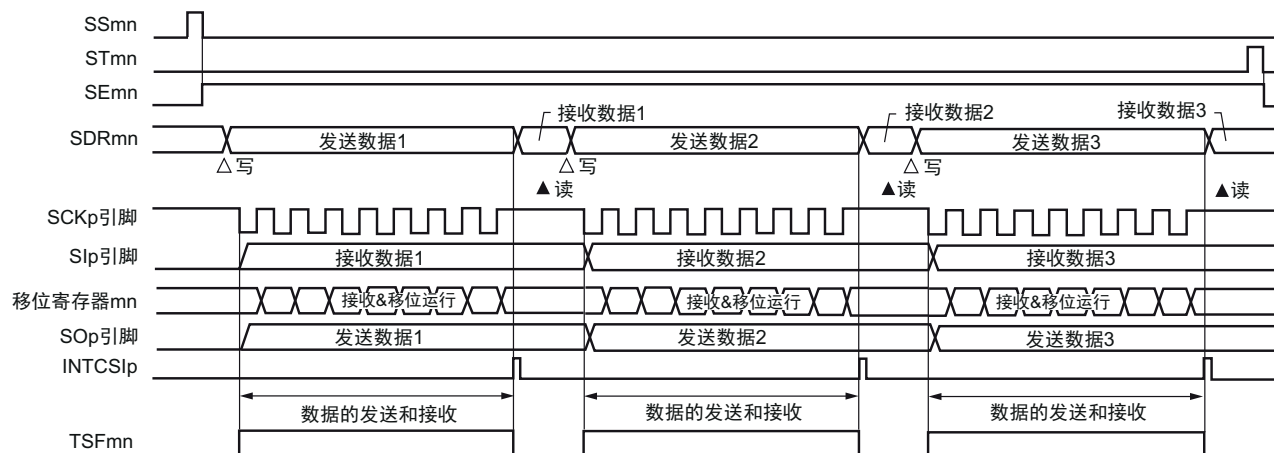


图 11-43 重新开始主控发送和接收的设定步骤



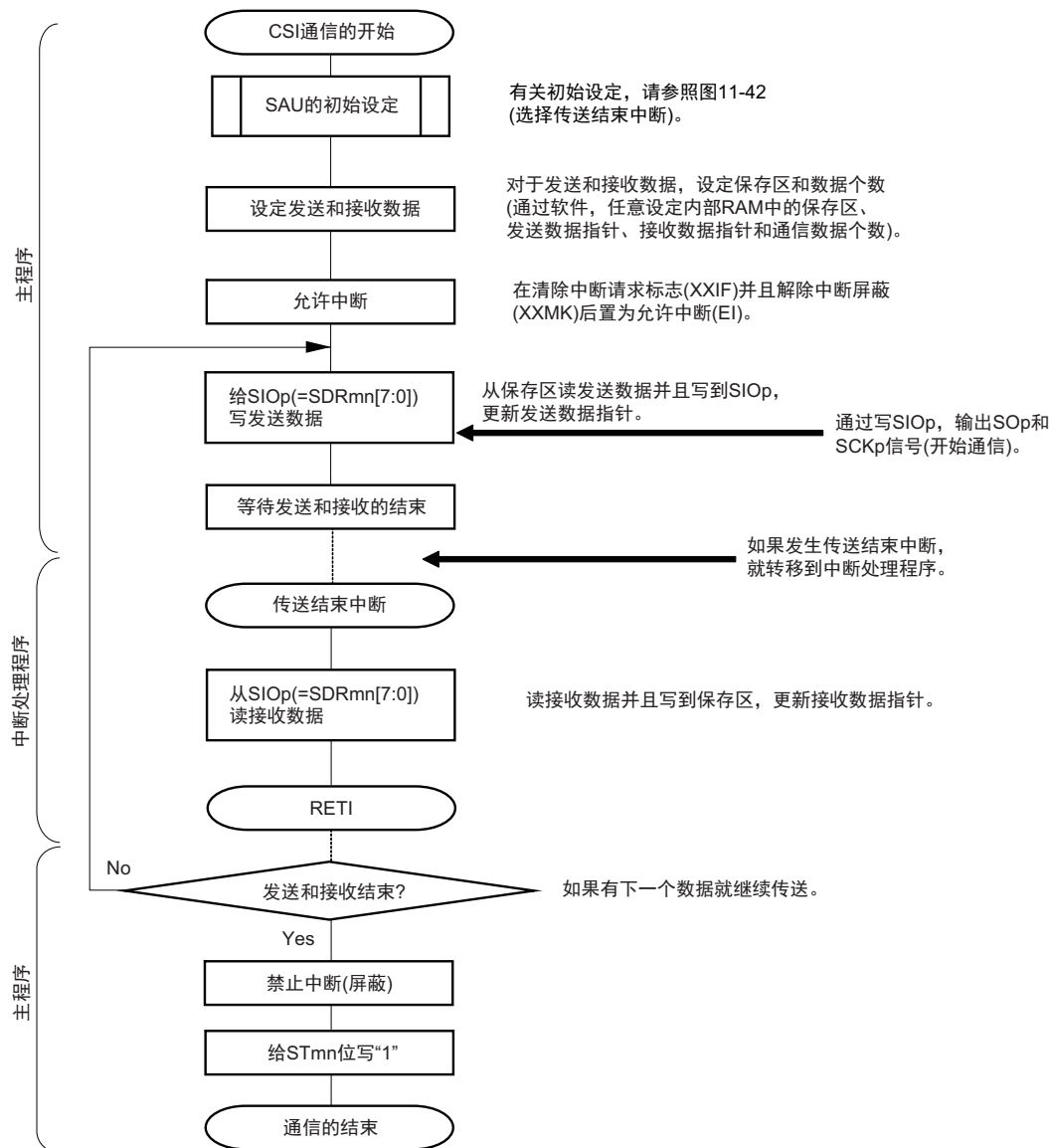
(3) 处理流程（单次发送和接收模式）

图 11-44 主控发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



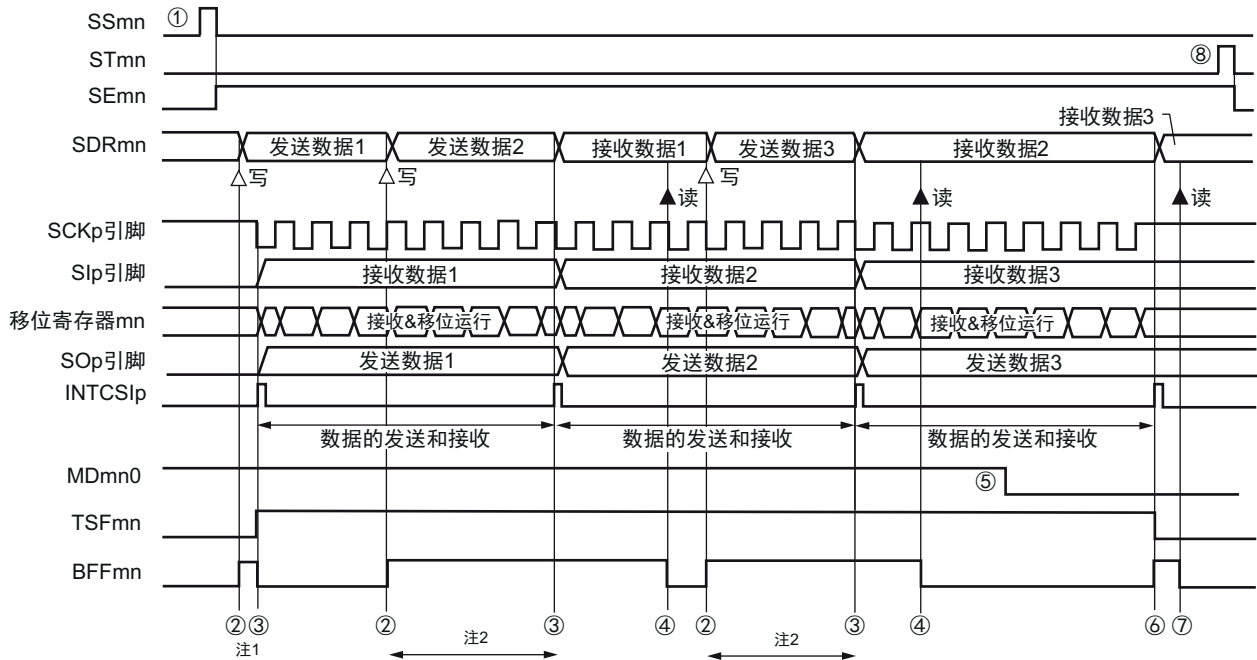
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI号 (p=00、11、20) mn=00、03、10

图 11-45 主控发送和接收（单次发送和接收模式）的流程图



(4) 处理流程（连续发送和接收模式）

图 11-46 主控发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



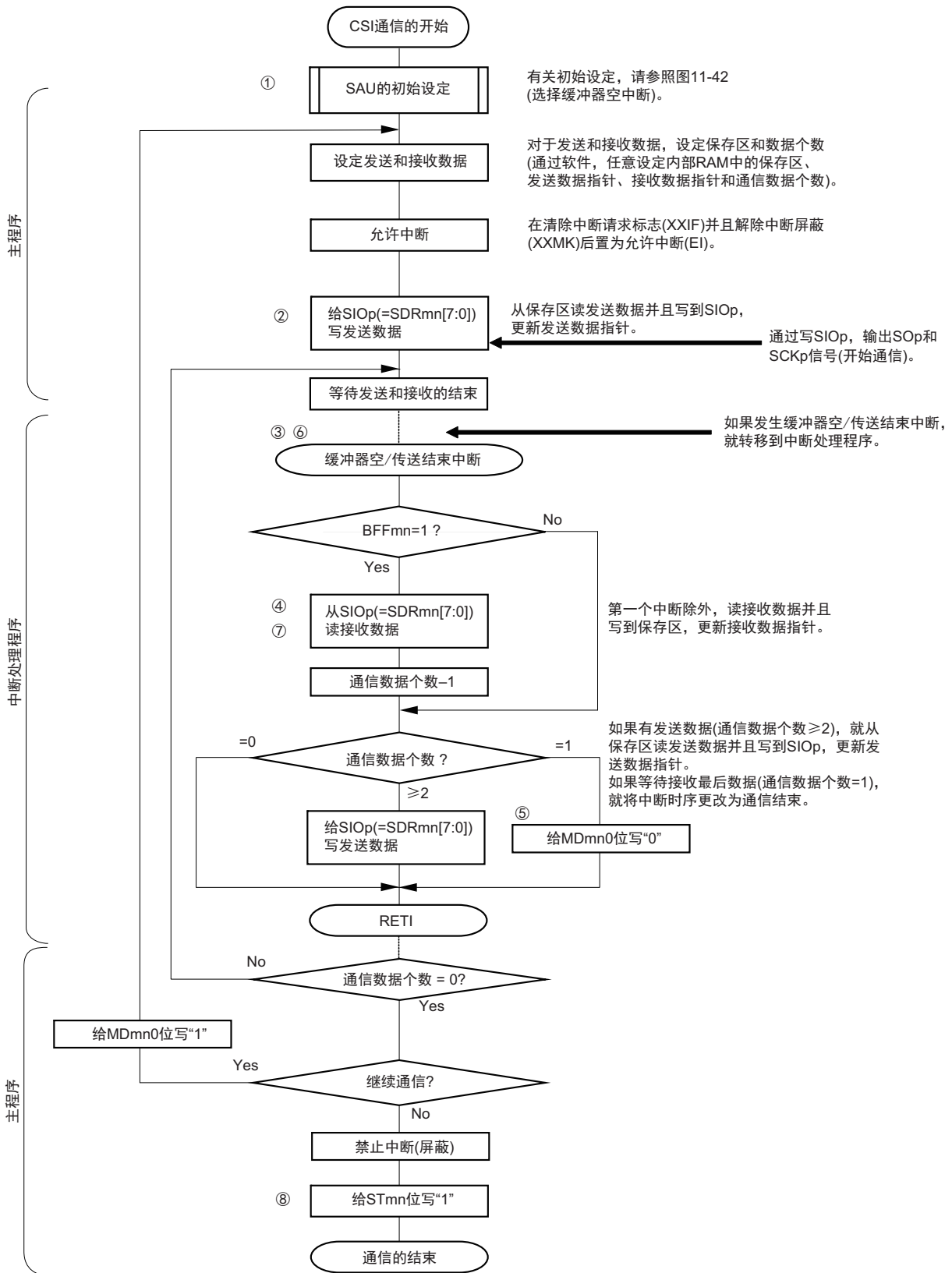
- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 1. 图中的①~⑧对应“图 11-47 主控发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

图 11-47 主控发送和接收（连续发送和接收模式）的流程图



备注 图中的①~⑧对应“图 11-46 主控发送和接收（连续发送和接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）”中的①~⑧。

11.5.4 从属发送

从属发送是指在从其他设备输入传送时钟的状态下 RL78 微控制器将数据发送到其他设备的运行。

3 线串行 I/O	CSI00	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SO00	SCK11、SO11	SCK20、SO20
中断	INTCSI00	INTCSI11	INTCSI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	7 位或者 8 位		
传送速率	Max. $f_{MCK}/6$ [Hz] 注 1、2		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 		
数据方向	MSB 优先或者 LSB 优先		

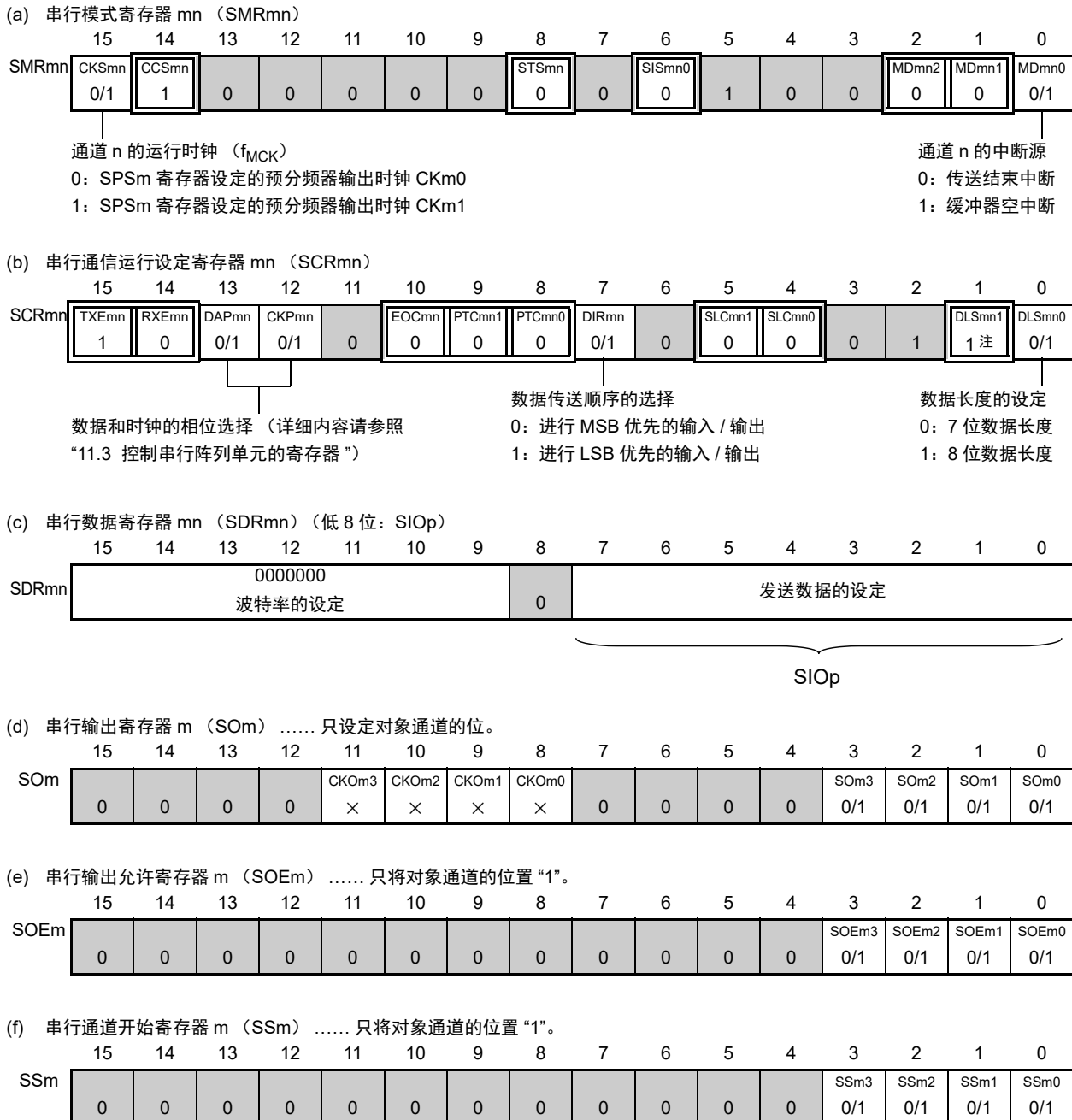
- 注 1. 因为在内部对 SCK00、SCK11、SCK20 引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为 $f_{MCK}/6$ [Hz]。
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性 ($T_A=-40 \sim +85^\circ\text{C}$)”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00、03、10

(1) 寄存器的设定

图 11-48 3 线串行 I/O (CSI00、CSI11、CSI20) 从属发送时的寄存器设定内容例子



注 只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

2. □: 在 CSI 从属发送模式中为固定设定。■: 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-49 从属发送的初始设定步骤

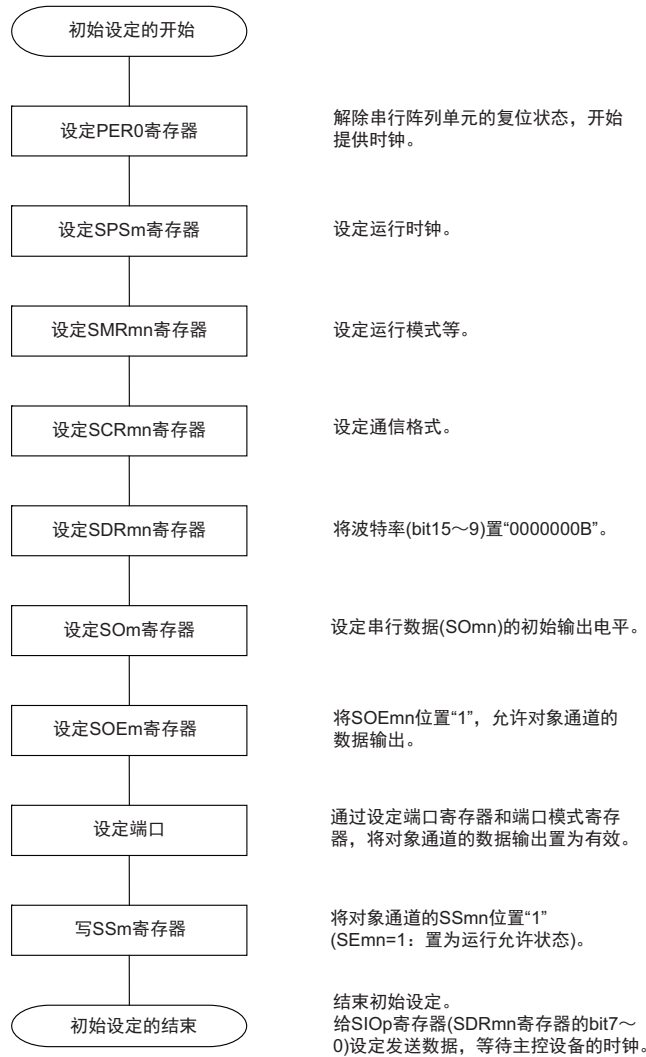


图 11-50 从属发送的中止步骤

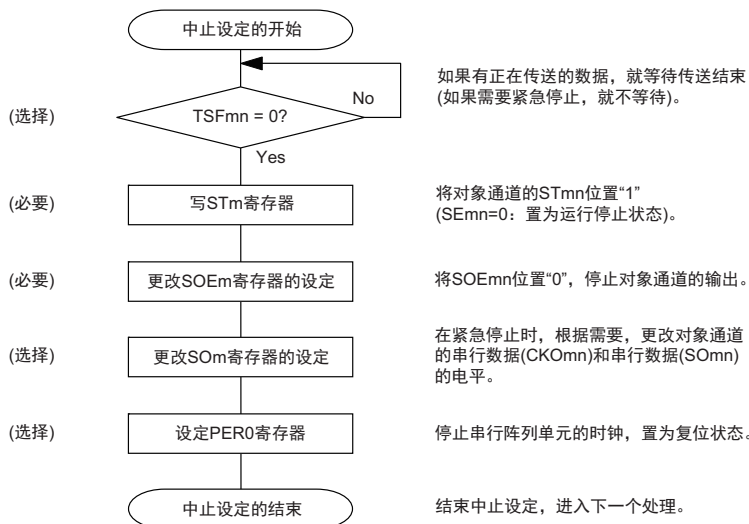
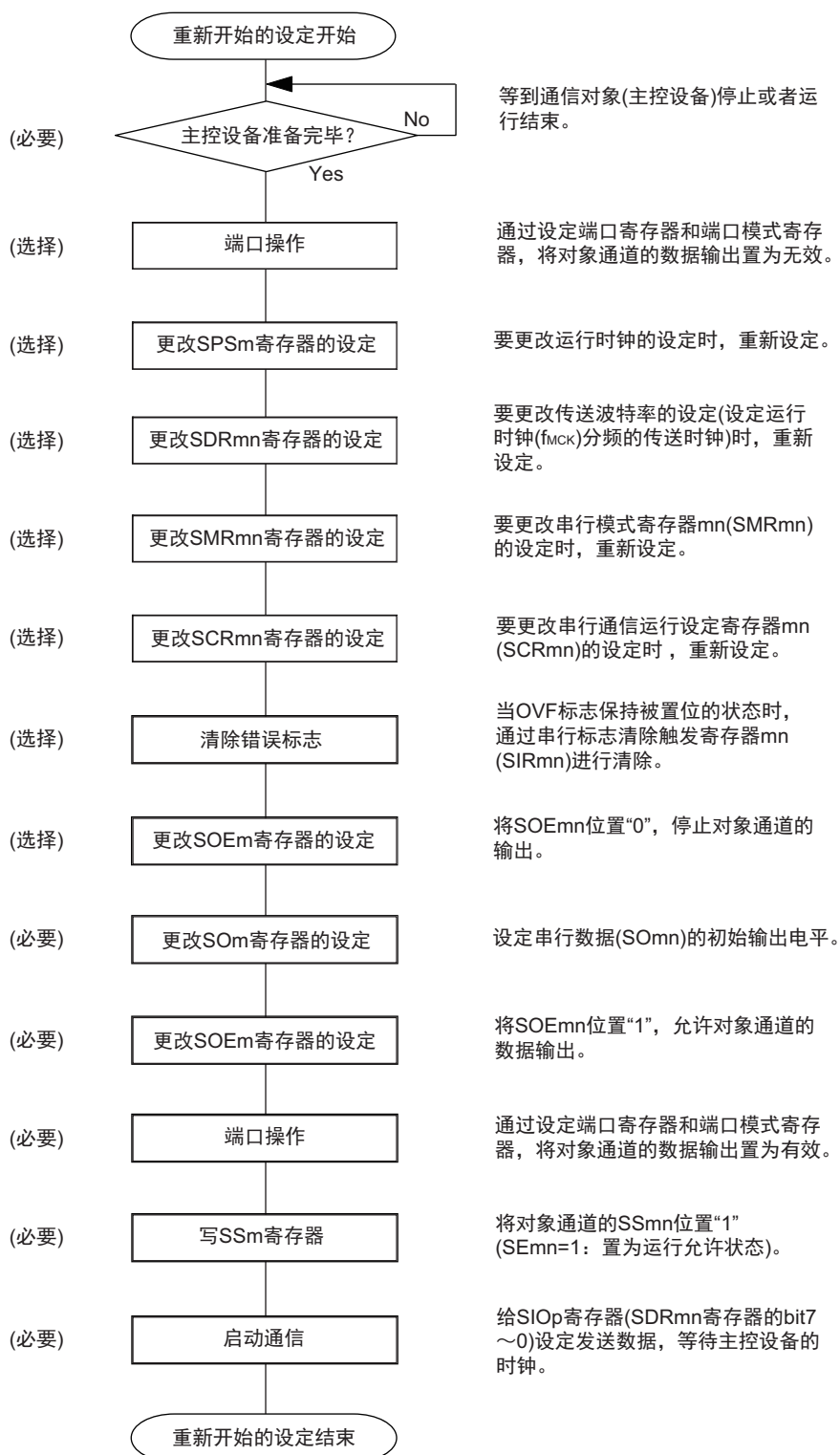


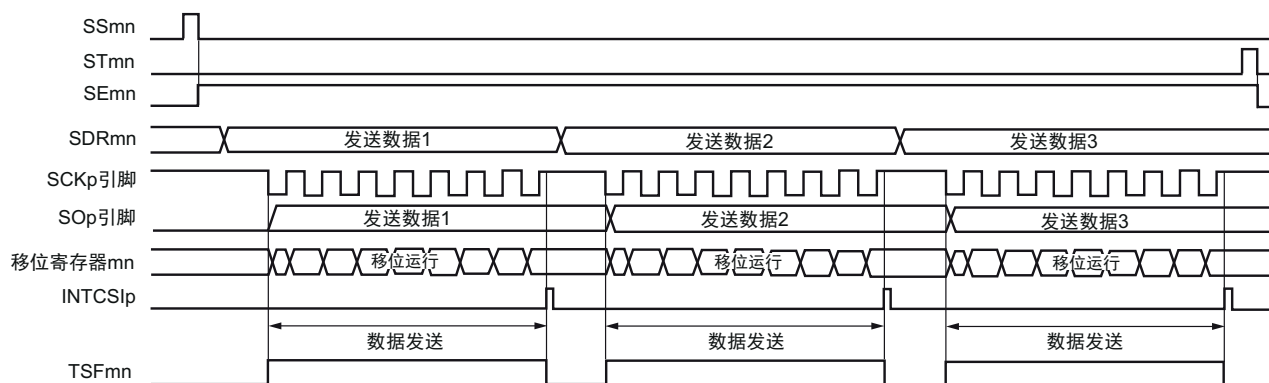
图 11-51 重新开始从属发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象（主控设备）停止或者通信结束后进行初始设定而不是进行重新开始的设定。

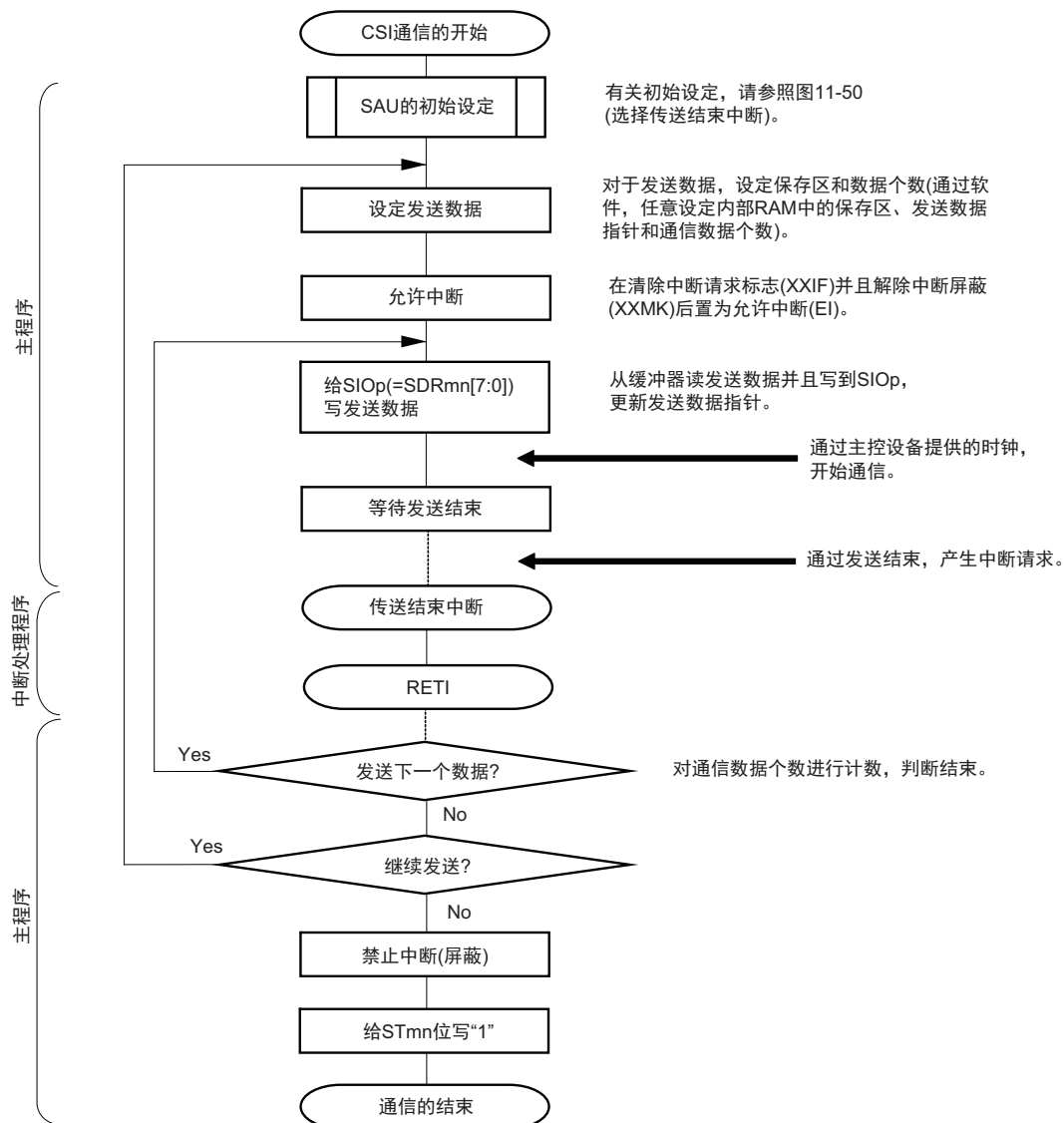
(3) 处理流程（单次发送模式）

图 11-52 从属发送（单次发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



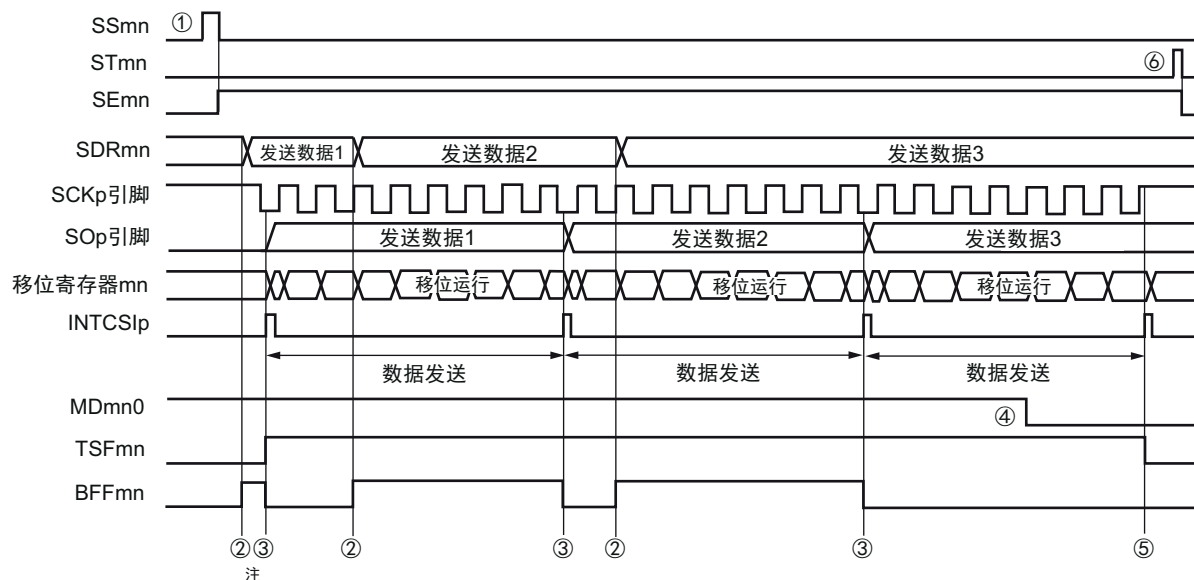
备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI号 (p=00、11、20) mn=00、03、10

图 11-53 从属发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 11-54 从属发送（连续发送模式）的时序图（类型 1：DAPmn=0、CKPmn=0）

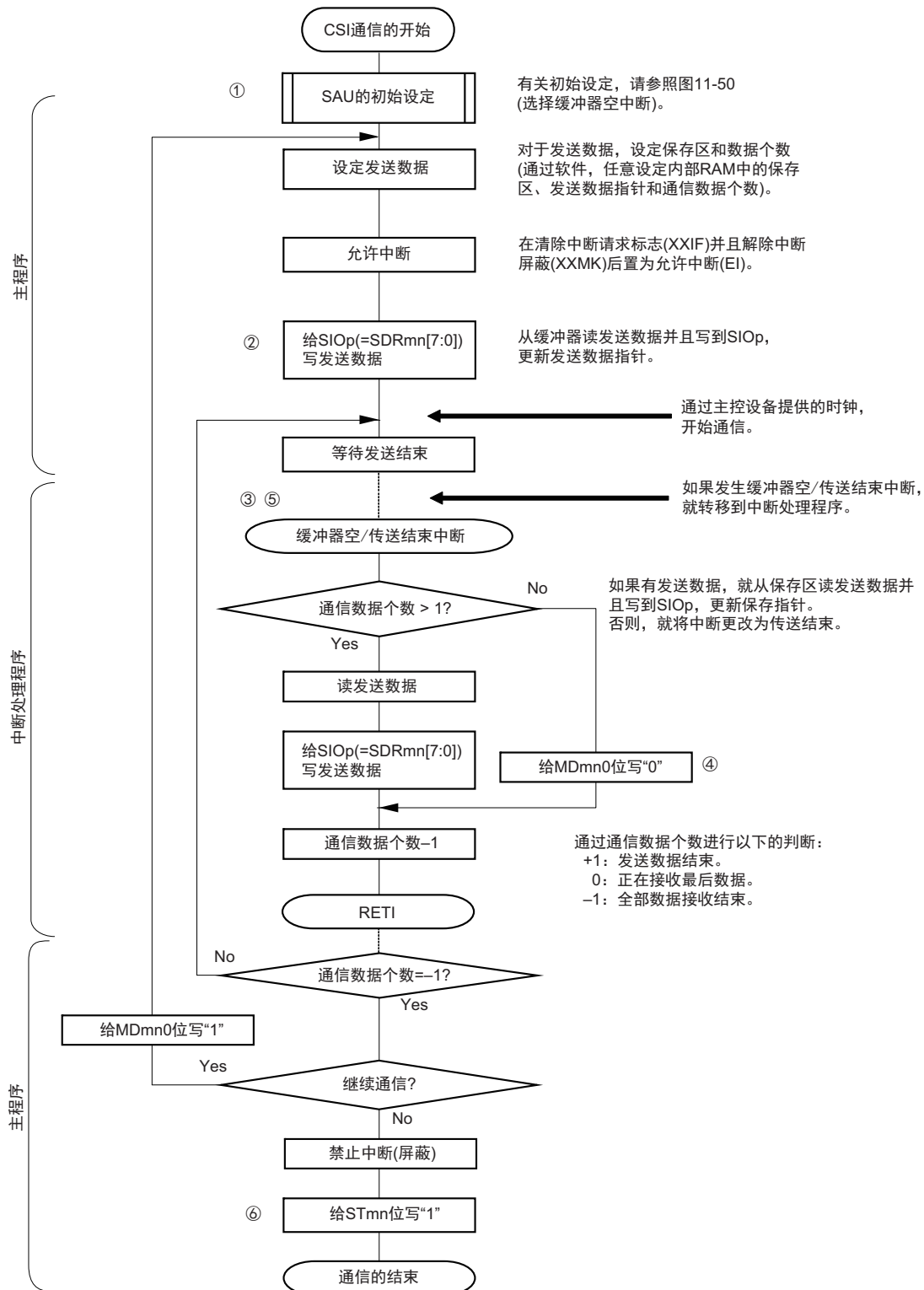


注 如果在串行状态寄存器 mn (SSRmn) 的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn (SDRmn) 时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 mn (SMRmn) 的 MDmn0 位。但是，必须在开始传送最后一位之前进行改写。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

图 11-55 从属发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 11-54 从属发送（连续发送模式）的时序图（类型 1: DAPmn=0、CKPmn=0）”中的①~⑥。

11.5.5 从属接收

从属接收是指在从其他设备输入传送时钟的状态下 RL78 微控制器从其他设备接收数据的运行。

3 线串行 I/O	CSI00	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SI00	SCK11、SI11	SCK20、SI20
中断	INTCSI00	INTCSI11	INTCSI20
	只限于传送结束中断（禁止设定缓冲器空中断）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	7 位或者 8 位		
传送速率	Max. $f_{MCK}/6$ [Hz] 注 1、2		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输出。 		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 		
数据方向	MSB 优先或者 LSB 优先		

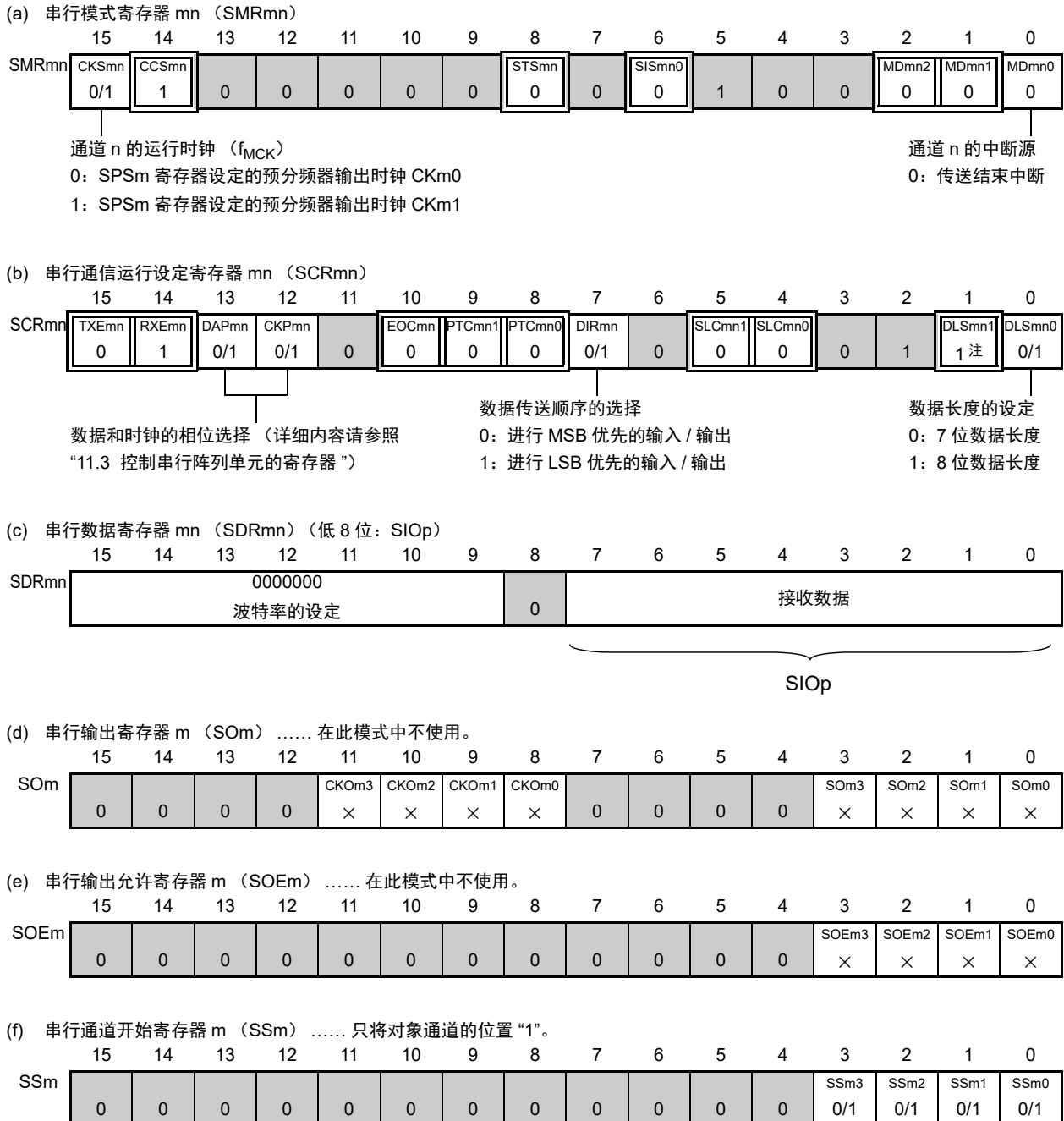
- 注 1. 因为在内部对 SCK00、SCK11、SCK20 引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为 $f_{MCK}/6$ [Hz]。
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性 ($T_A=-40 \sim +85^\circ\text{C}$)”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00、03、10

(1) 寄存器的设定

图 11-56 3 线串行 I/O (CSI00、CSI11、CSI20) 从属接收时的寄存器设定内容例子



注 只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

2. : 在从属接收模式中为固定设定。 : 不能设定 (设定初始值)。
 X : 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1 : 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-57 从属接收的初始设定步骤

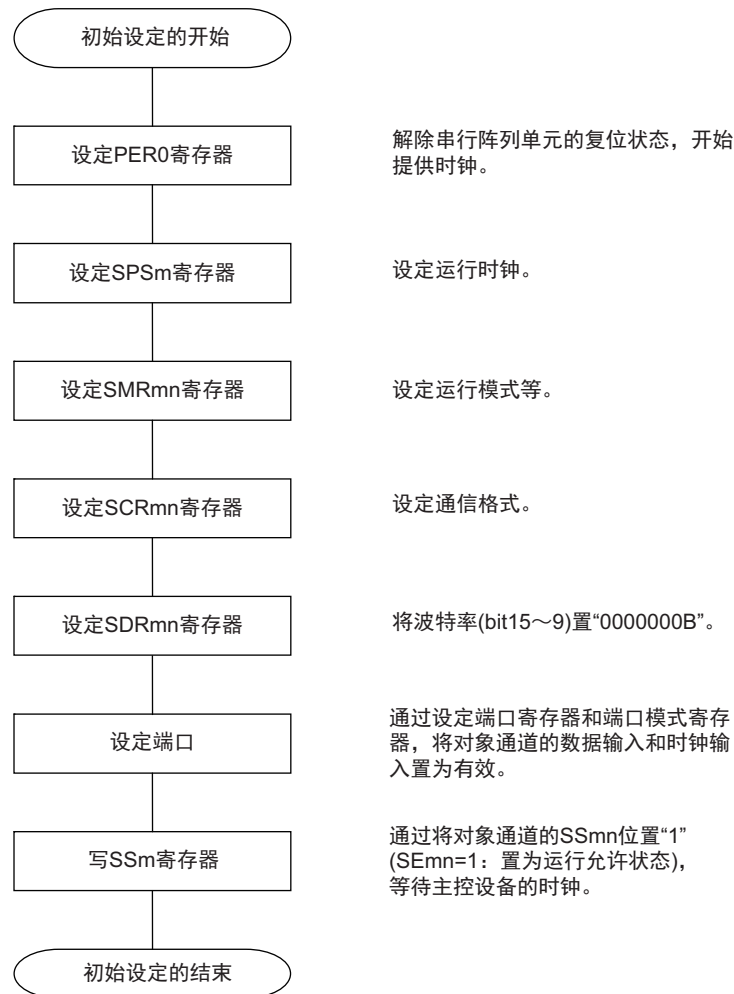


图 11-58 从属接收的中止步骤

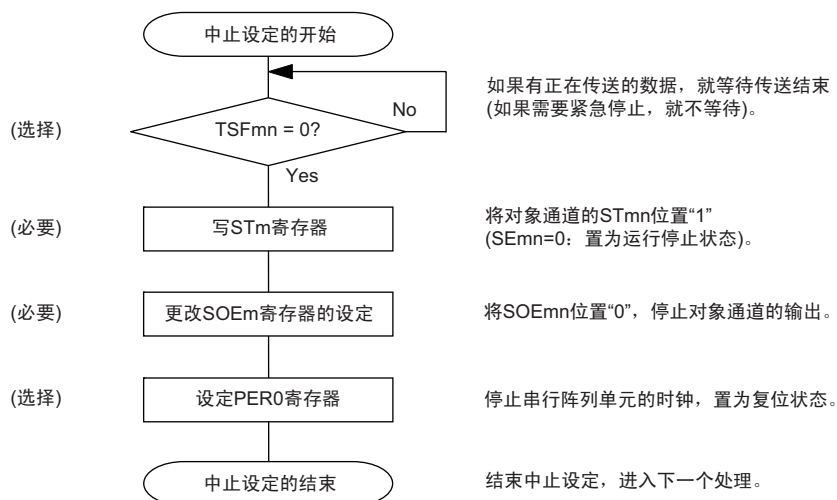
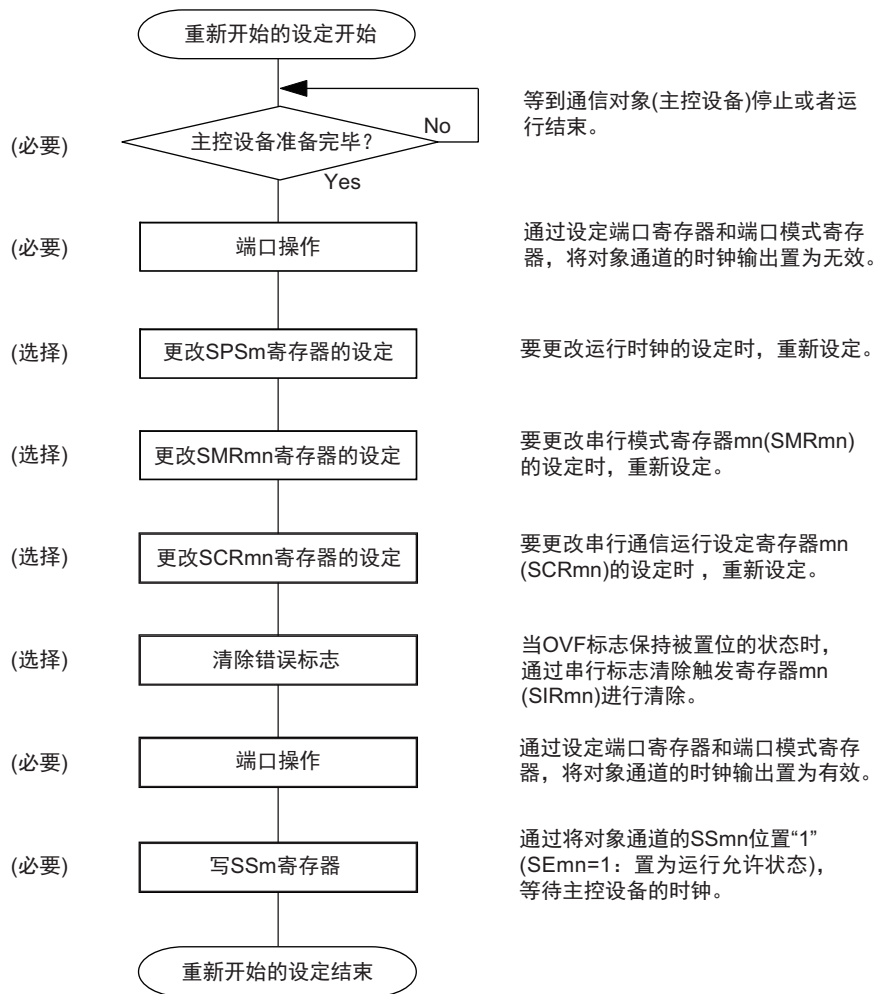


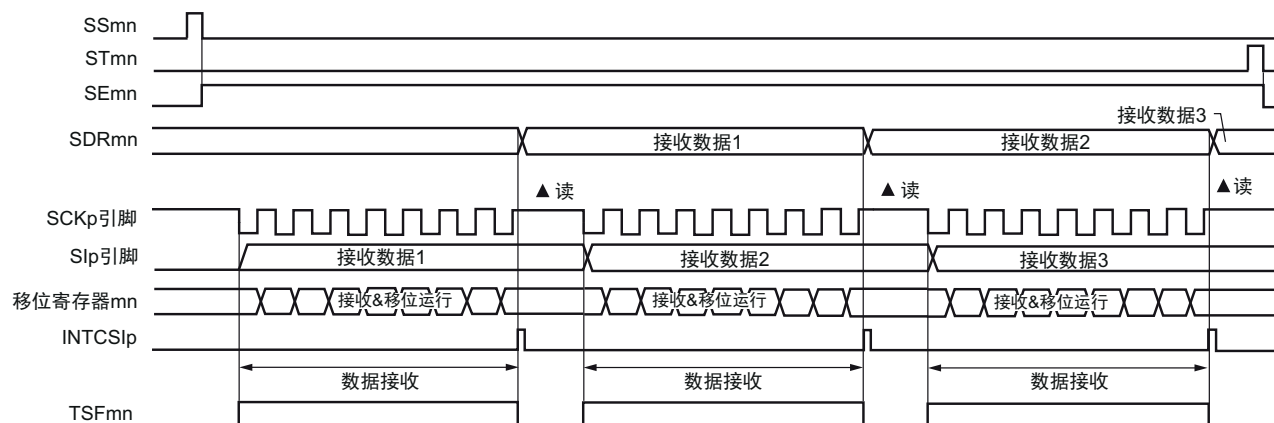
图 11-59 重新开始从属接收的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (主控设备) 停止或者通信结束后进行初始设定而不是进行重新开始的设定。

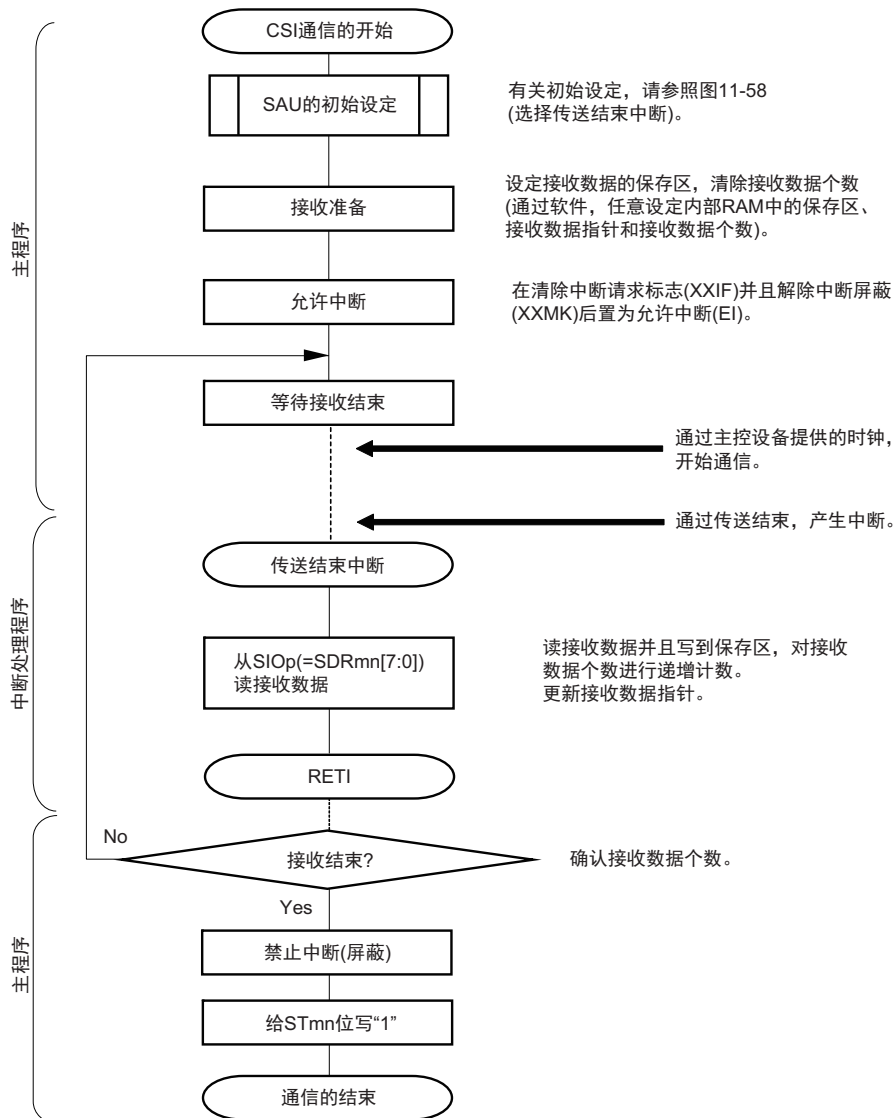
(3) 处理流程（单次接收模式）

图 11-60 从属接收（单次接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI号 (p=00、11、20) mn=00、03、10

图 11-61 从属接收（单次接收模式）的流程图



11.5.6 从属的发送和接收

从属的发送和接收是指在从其他设备输入传送时钟的状态下 RL78 微控制器和其他设备进行数据发送和接收的运行。

3 线串行 I/O	CSI00	CSI11	CSI20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCK00、SI00、SO00	SCK11、SI11、SO11	SCK20、SI20、SO20
中断	INTCSI00	INTCSI11	INTCSI20
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	7 位或者 8 位		
传送速率	Max. $f_{MCK}/6$ [Hz] 注 1、2		
数据相位	能通过 SCRmn 寄存器的 DAPmn 位进行选择。 <ul style="list-style-type: none"> • DAPmn=0: 在串行时钟开始运行时, 开始数据输入 / 输出。 • DAPmn=1: 在串行时钟开始运行的半个时钟前, 开始数据输入 / 输出。 		
时钟相位	能通过 SCRmn 寄存器的 CKPmn 位进行选择。 <ul style="list-style-type: none"> • CKPmn=0: 正相 • CKPmn=1: 反相 		
数据方向	MSB 优先或者 LSB 优先		

- 注 1. 因为在内部对 SCK00、SCK11、SCK20 引脚输入的外部串行时钟进行采样后使用, 所以最大传送速率为 $f_{MCK}/6$ [Hz]。
 2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性 ($T_A=-40 \sim +85^\circ\text{C}$)”）的范围内使用。

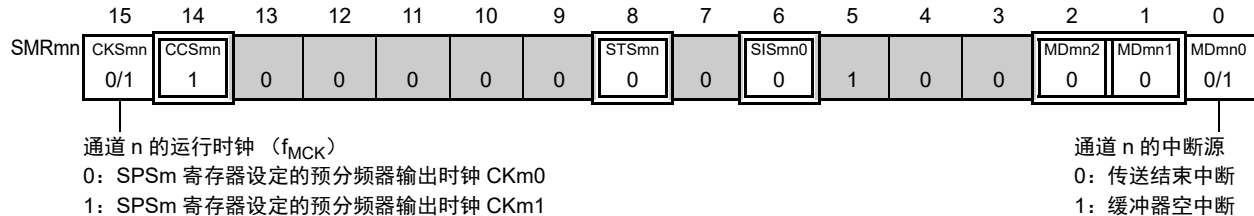
备注 1. f_{MCK} : 对象通道的运行时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00、03、10

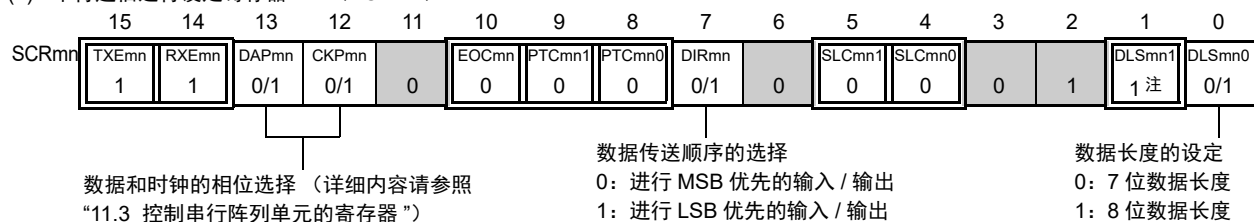
(1) 寄存器的设定

图 11-62 3 线串行 I/O (CSI00、CSI11、CSI20) 从属发送和接收时的寄存器设定内容例子

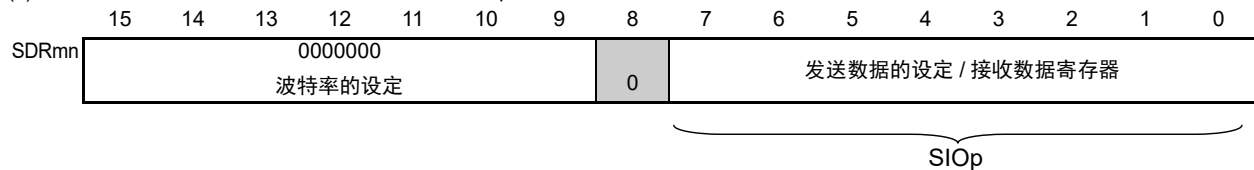
(a) 串行模式寄存器 mn (SMRmn)



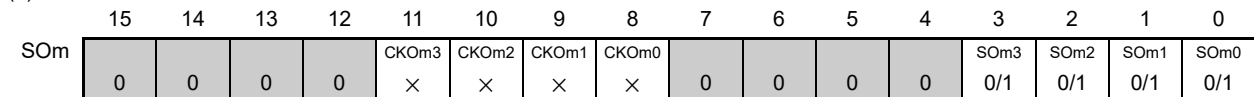
(b) 串行通信运行设定寄存器 mn (SCRmn)



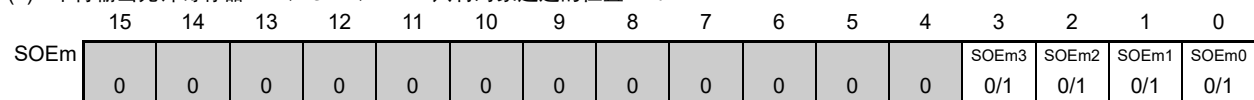
(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)



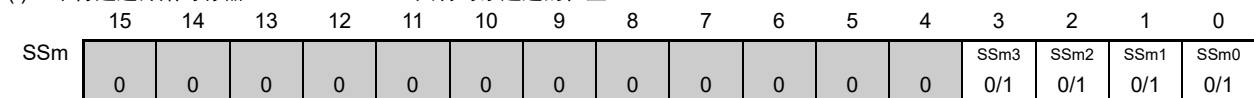
(d) 串行输出寄存器 m (SOM) 只设定对象通道的位。



(e) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。



(f) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。



注 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。

注意 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

2. : 在 CSI 从属发送和接收模式中为固定设定。 : 不能设定 (设定初始值)。
 ×: 这是在此模式中不能使用的位 (在其他模式中也不使用的情况下, 设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 操作步骤

图 11-63 从属发送和接收的初始设定步骤

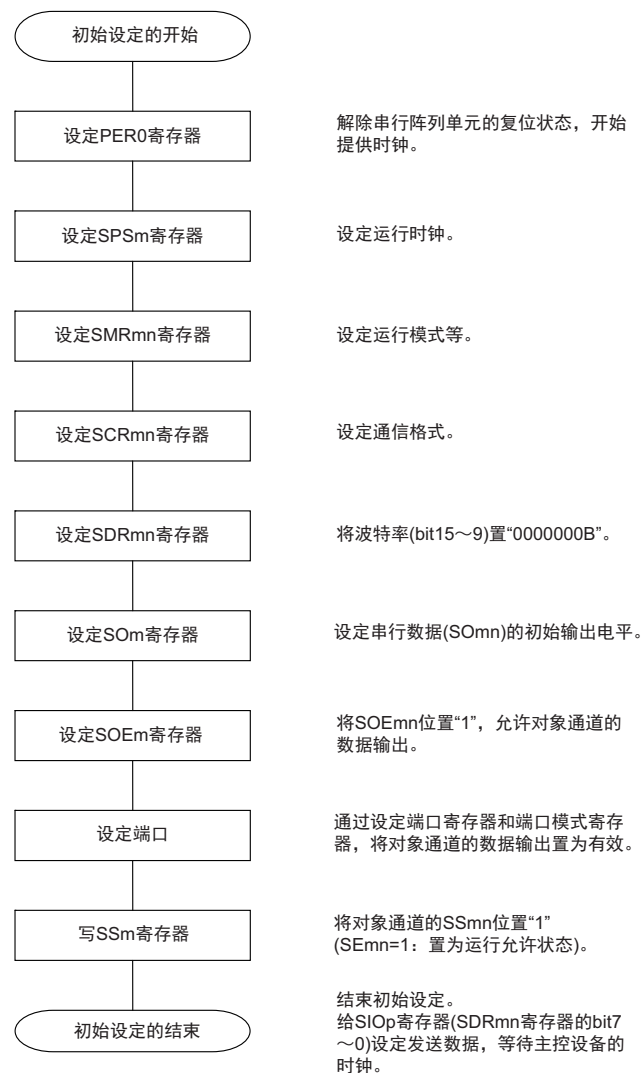


图 11-64 从属发送和接收的中止步骤

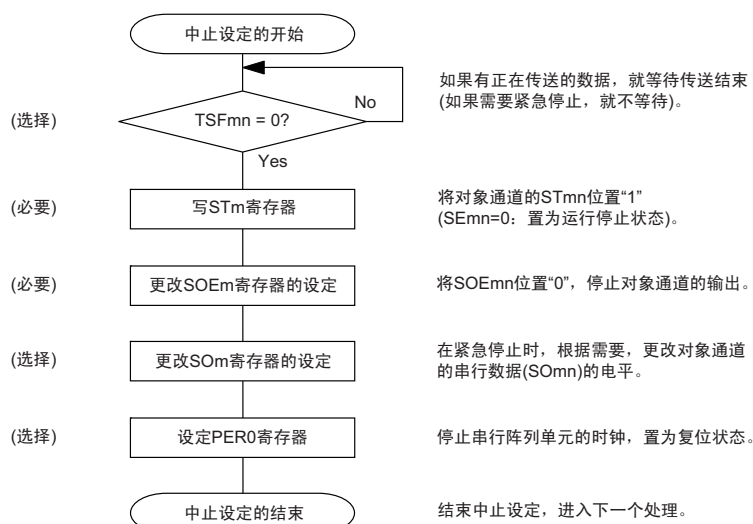
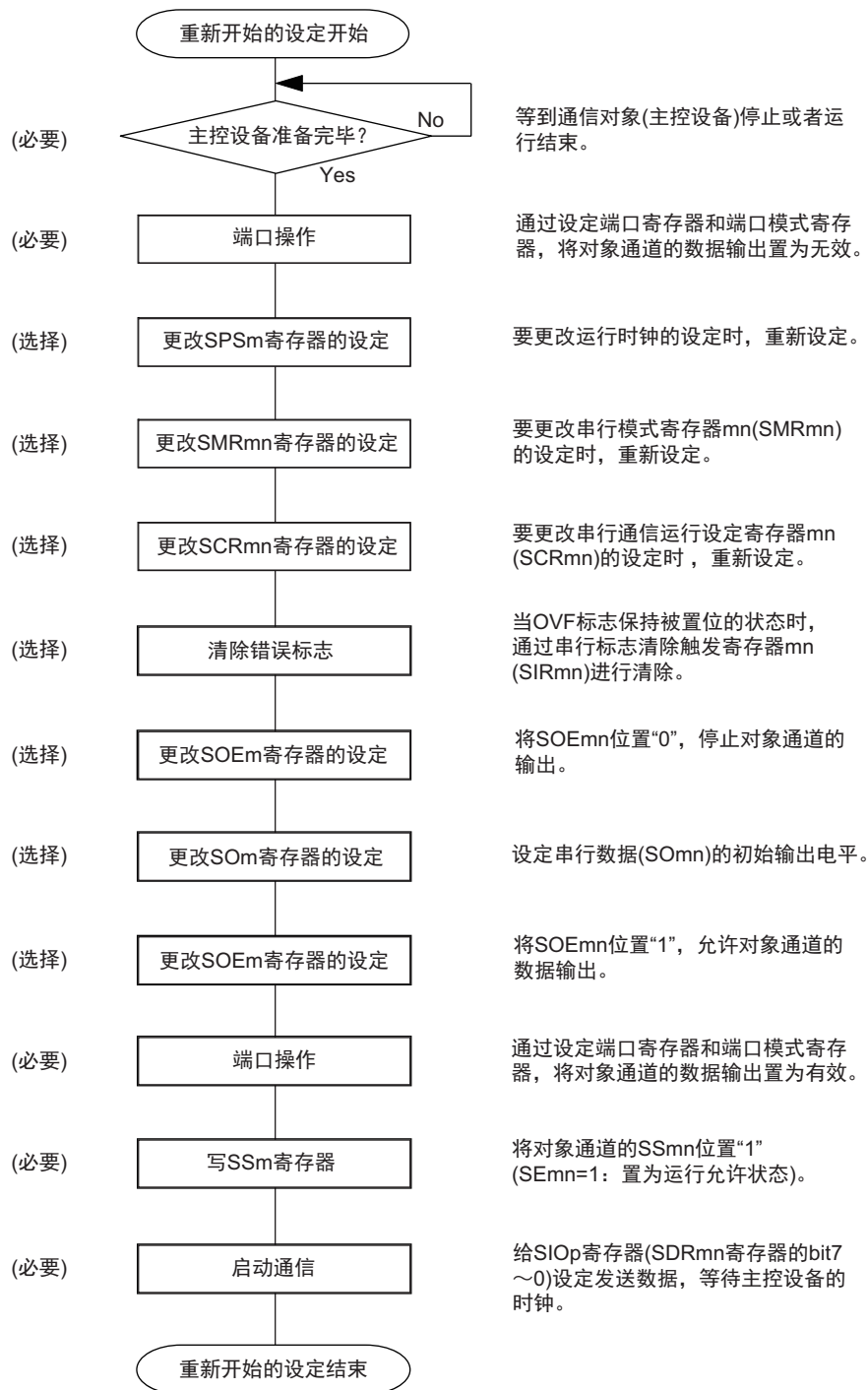


图 11-65 重新开始从属发送和接收的设定步骤

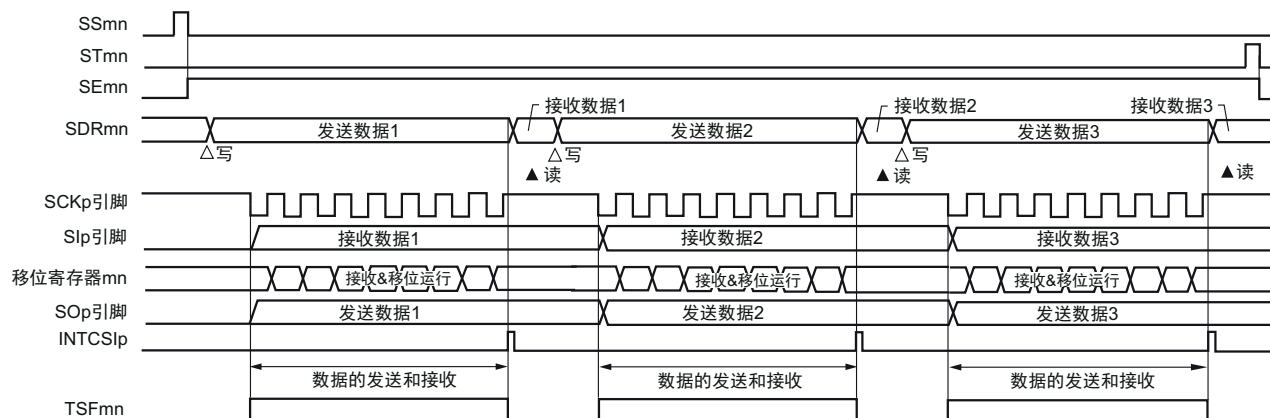


注意 1. 在主导设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

2. 如果在中止设定中改写 PER0 来停止提供时钟, 就必须在等到通信对象 (主导设备) 停止或者通信结束后进行初始设定而不是进行重新开始的设定。

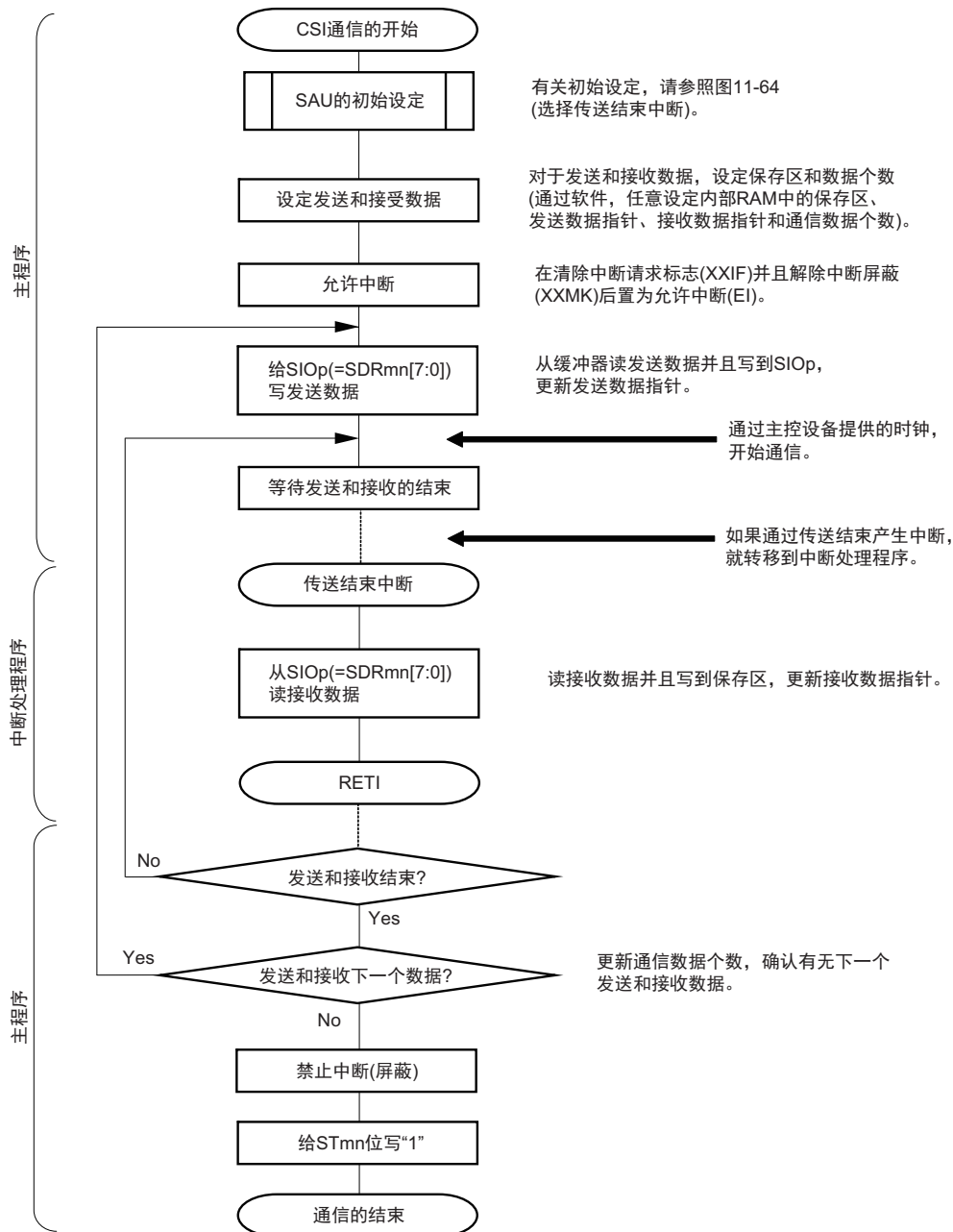
(3) 处理流程（单次发送和接收模式）

图 11-66 从属发送和接收（单次发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

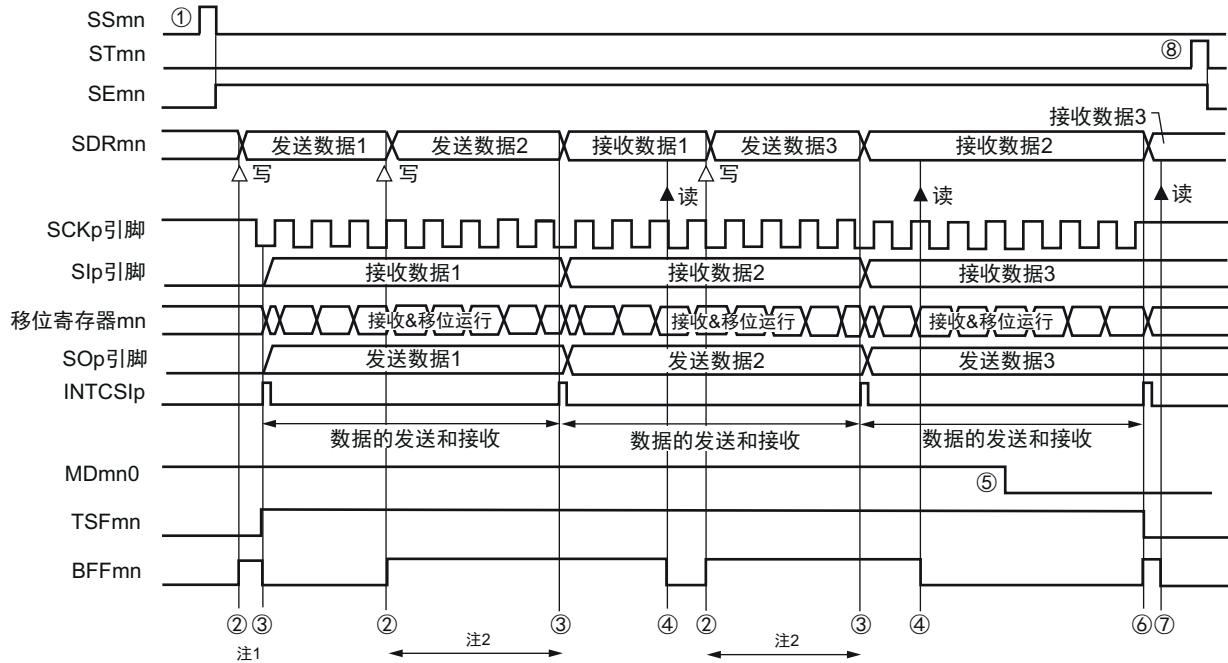
图 11-67 从属发送和接收（单次发送和接收模式）的流程图



注意 在主控设备开始输出时钟前, 必须给 SIOp 寄存器设定发送数据。

(4) 处理流程（连续发送和接收模式）

图 11-68 从属发送和接收（连续发送和接收模式）的时序图（类型 1：DAPmn=0、CKPmn=0）



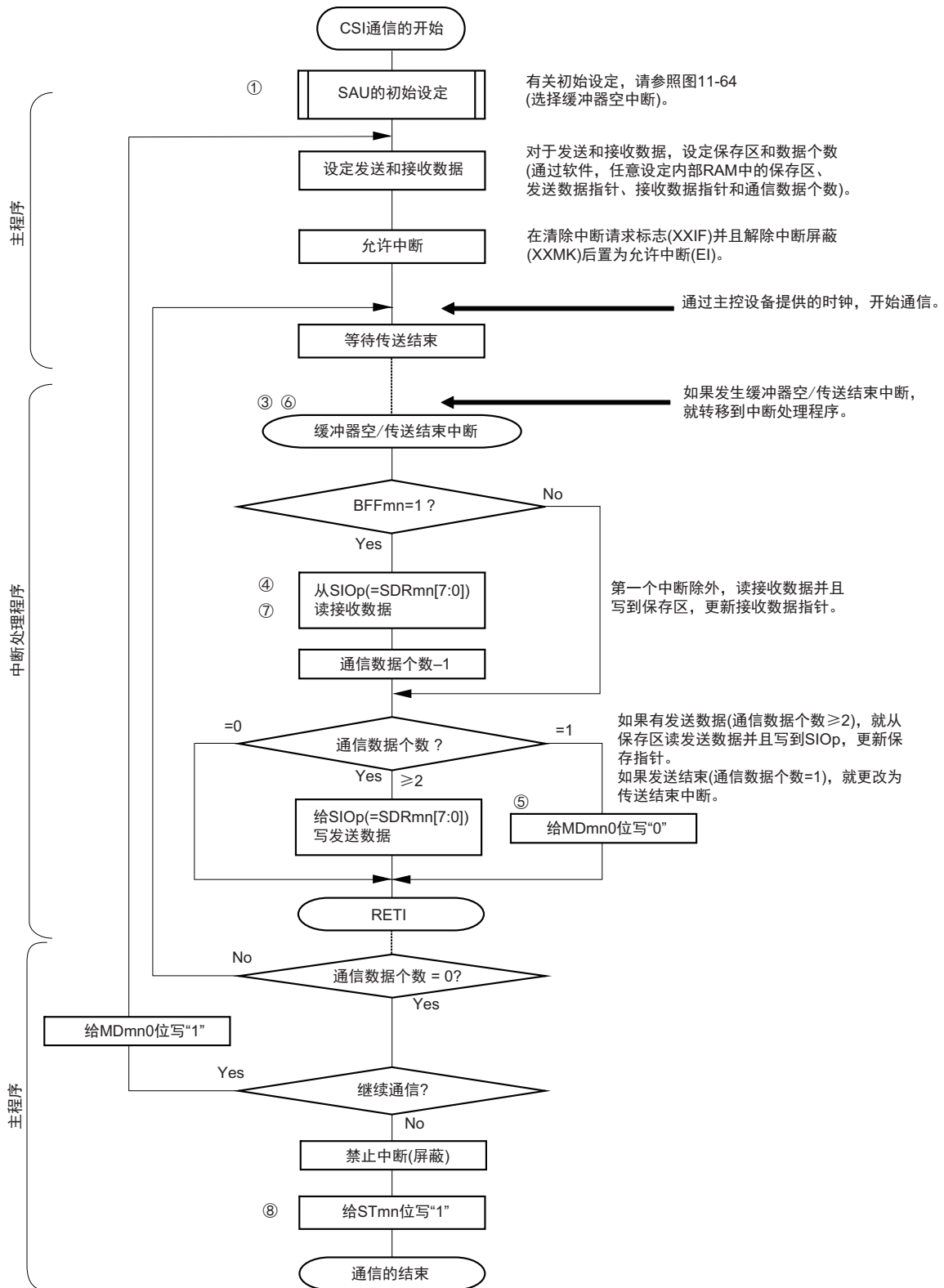
- 注 1. 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。
2. 如果在此期间读取 SDRmn 寄存器，就能读发送数据。此时，不影响传送运行。

注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 1. 图中的①~⑧对应“图 11-69 从属发送和接收（连续发送和接收模式）的流程图”中的①~⑧。

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) p: CSI 号 (p=00、11、20) mn=00、03、10

图 11-69 从属发送和接收（连续发送和接收模式）的流程图



注意 在 主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 11-68 从属发送和接收（连续发送和接收模式）的时序图（类型 1: DAPmn=0、CKPmn=0）”中的①~⑧。

11.5.7 SNOOZE 模式功能

这是在 STOP 模式中通过检测 SCK_p 引脚的输入使 CSI 进行接收的模式。通常，在 STOP 模式中 CSI 停止通信，但是如果使用 SNOOZE 模式功能，就能在 CPU 不运行的状态下通过检测 SCK_p 引脚的输入进行 CSI 接收。只有以下的通道才能设定 SNOOZE 模式。

- 本产品：CSI00

要在 SNOOZE 模式中使用 CSI 时，必须在转移到 STOP 模式前进行以下的设定（参照“图 11-71 SNOOZE 模式运行（启动一次）的流程图和图 11-73 SNOOZE 模式运行（连续启动）的流程图”）。

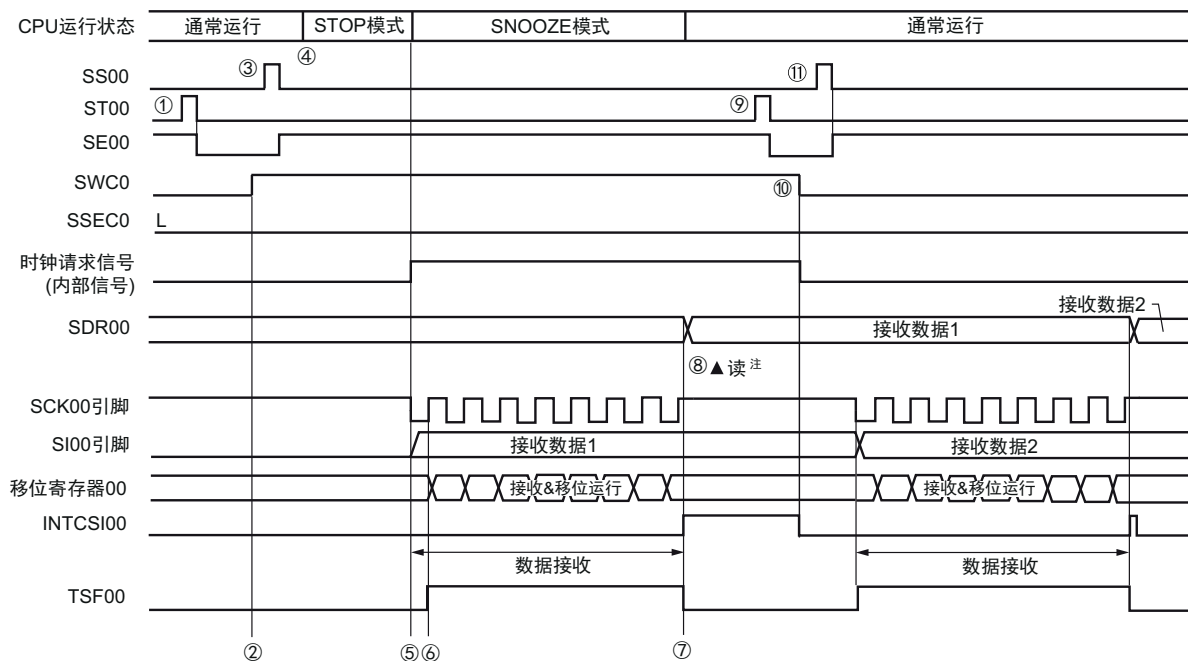
- 必须在即将转移到 STOP 模式前将串行待机控制寄存器 0（SSC0）的 SWC0 位置“1”。在初始设定结束后将串行通道开始寄存器 m（SSm）的 SSm0 位置“1”。
- 在转移到 STOP 模式后，如果检测到 SCK_p 引脚的有效边沿，就转移到 SNOOZE 模式。通过 SCK_p 引脚的串行时钟输入，开始 CSI 接收。

注意 1. 只有在选择高速内部振荡器时钟作为 f_{CLK} 时才能设定 SNOOZE 模式。

2. 在 SNOOZE 模式中使用时，最大传送速率为 1Mbps。

(1) SNOOZE 模式运行（启动一次）

图 11-70 SNOOZE 模式运行（启动一次）的时序图（类型 1：DAP_mn=0、CKP_mn=0）



注 必须在 SWC0 位为“1”的状态下并且在检测到 SCK_p 引脚输入的下一个边沿前读接收数据。

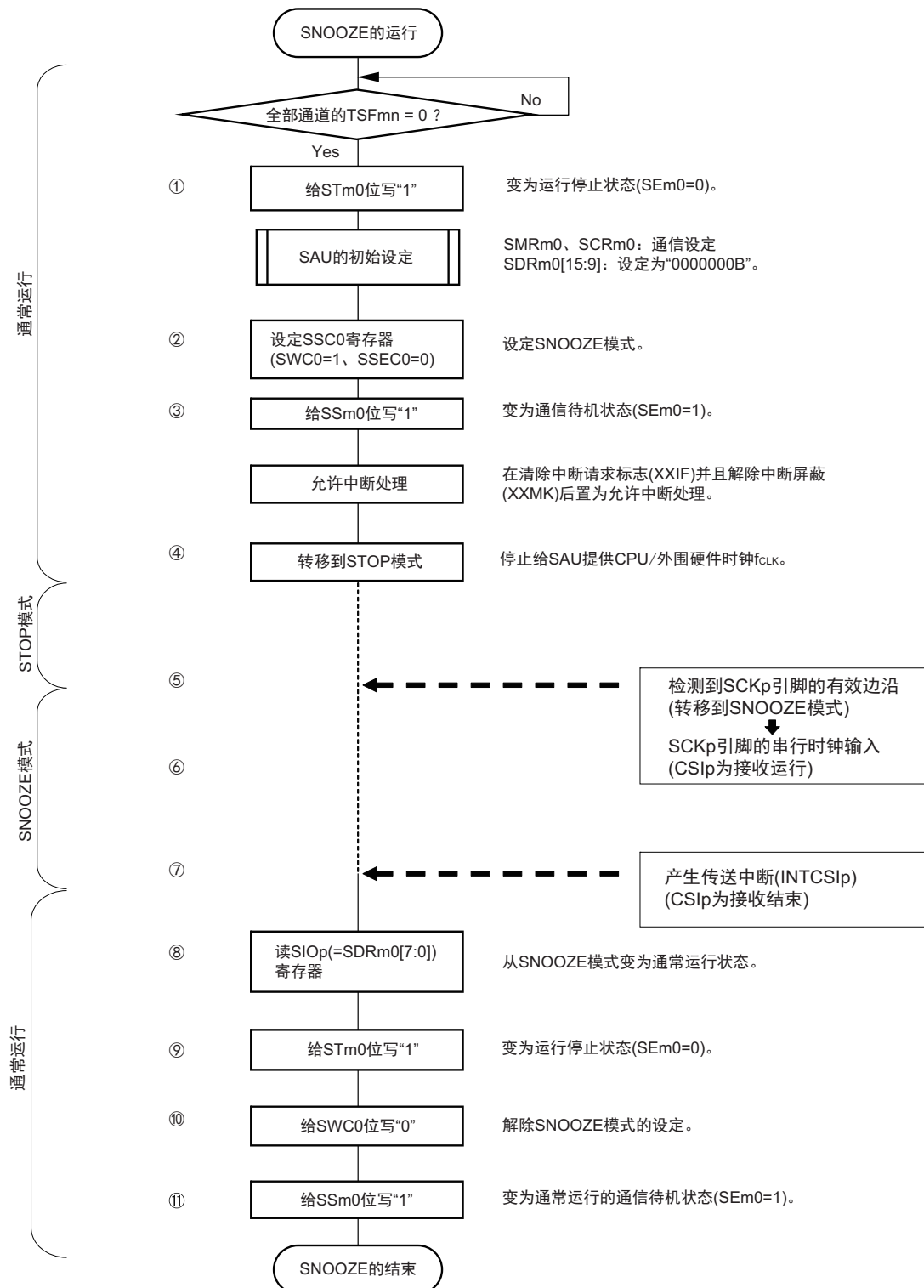
注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 ST_m0 位置“1”（清除 SE_m0 位并且停止运行），而且还必须在接收结束后清除 SWC0 位（解除 SNOOZE 模式）。

2. 当 SWC0 位为“1”时，BFF_m0 标志和 OVF_m0 标志不变。

备注 1. 图中的①~⑪对应“图 11-71 SNOOZE 模式运行（启动一次）的流程图”中的①~⑪。

2. 本产品：m=0，p=00

图 11-71 SNOOZE 模式运行（启动一次）的流程图

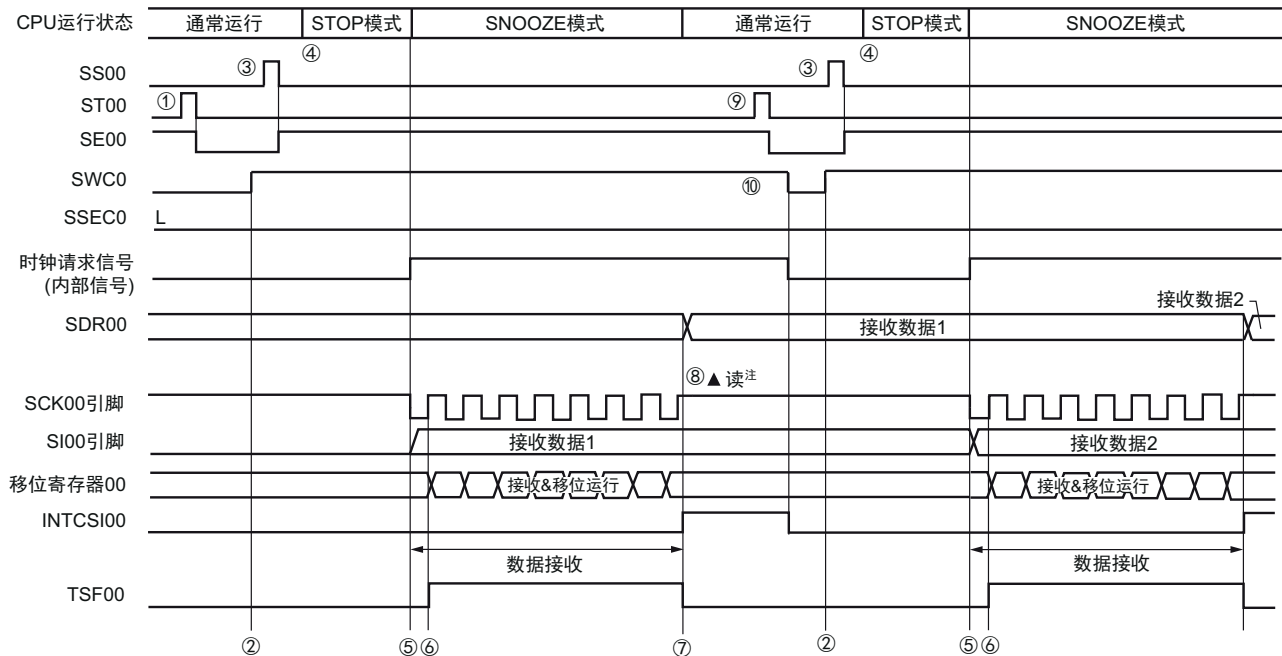


备注 1. 图中的①~⑪对应“图 11-70 SNOOZE 模式运行（启动一次）的时序图（类型 1：DAPmn=0、CKPmn=0）”中的①~⑪。

2. 本产品：m=0，p=00

(2) SNOOZE 模式运行（连续启动）

图 11-72 SNOOZE 模式运行（连续启动）的时序图（类型 1：DAPmn=0、CKPmn=0）



注 必须在 SWC0 位为“1”的状态下并且在检测到 SCKp 引脚输入的下一个边沿前读接收数据。

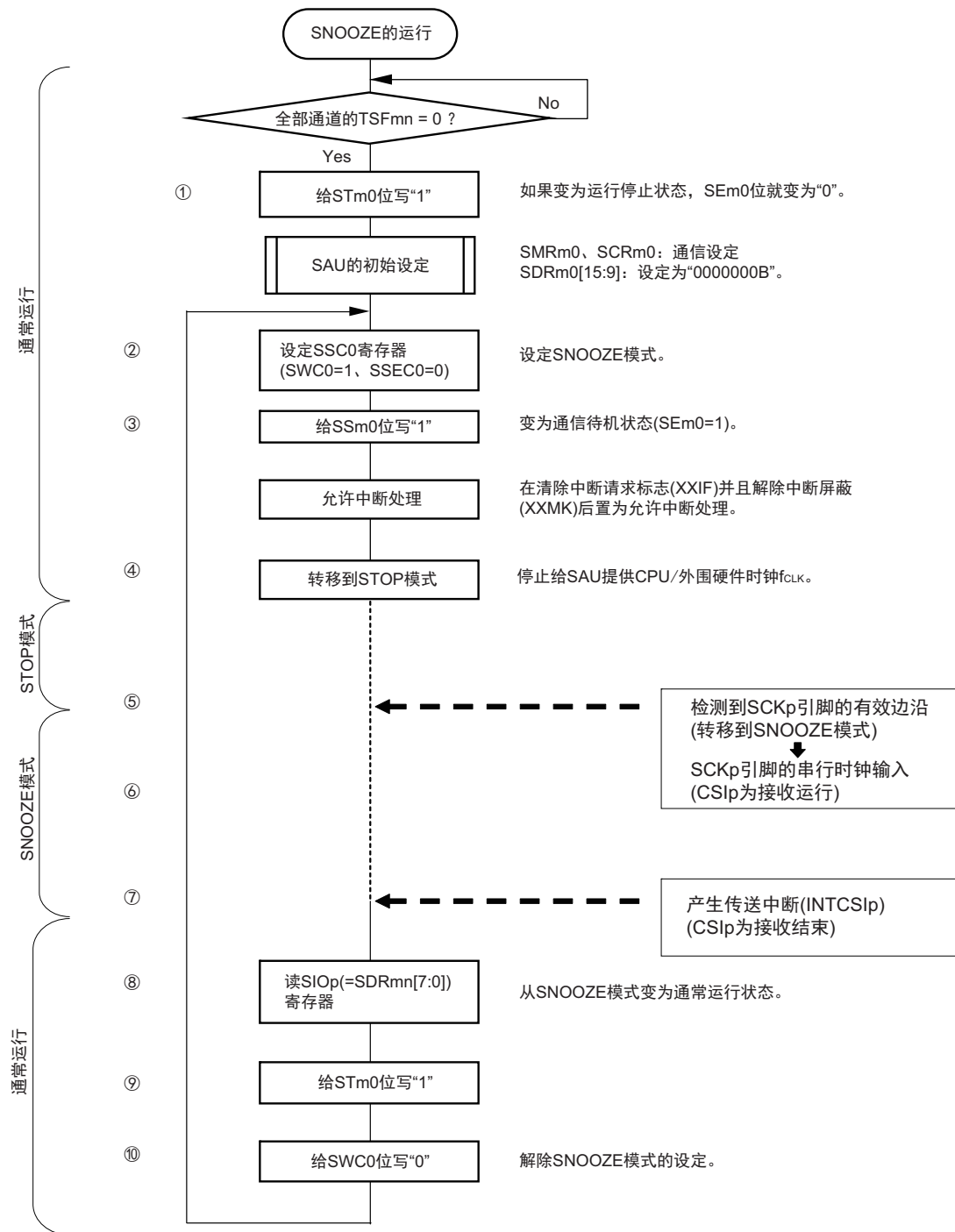
注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm0 位置“1”（清除 SEm0 位并且停止运行），而且还必须在接收结束后清除 SWC0 位（解除 SNOOZE 模式）。

2. 当 SWC0 位为“1”时，BFFm0 标志和 OVFm0 标志不变。

备注 1. 图中的①~⑩对应“图 11-73 SNOOZE 模式运行（连续启动）的流程图”中的①~⑩。

2. 本产品：m=0，p=00

图 11-73 SNOOZE 模式运行（连续启动）的流程图



备注 1. 图中的①~⑩对应“图 11-72 SNOOZE 模式运行（连续启动）的时序图（类型 1：DAPmn=0、CKPmn=0）”中的①~⑩。

2. 本产品：m=0, p=00

11.5.8 传送时钟频率的计算

3 线串行 I/O (CSI00、CSI11、CSI20) 通信的传送时钟频率能用以下计算式进行计算。

(1) 主控设备

$$\text{〈传送时钟频率〉} = \{ \text{对象通道的运行时钟 (} f_{\text{MCK}} \text{) 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) 从属设备

$$\text{〈传送时钟频率〉} = \{ \text{主控设备提供的串行时钟 (SCK) 频率} \} \text{注} \quad [\text{Hz}]$$

注 容许的最大传送时钟频率为 $f_{\text{MCK}}/6$ 。

备注 因为SDRmn[15:9]的值为串行数据寄存器mn (SDRmn) 的bit15~9的值 (0000000B~1111111B), 所以为0~127。

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的 bit15 (CKSmn)。

表 11-2 3 线串行 I/O 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f_{MCK}) 注	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	$f_{CLK}=20\text{MHz}$ 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 m（STm）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

11.5.9 在 3 线串行 I/O (CSI00、CSI11、CSI20) 通信过程中发生错误时的处理步骤

在 3 线串行 I/O (CSI00、CSI11、CSI20) 通信过程中发生错误时的处理步骤如图 11-74 所示。

图 11-74 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。→	SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SIRmn) 写“1”。	清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

11.6 UART (UART0 ~ UART2) 通信的运行

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

[数据的发送和接收]

- 7 位、8 位或者 9 位的数据长度注
- MSB/LSB 优先的选择
- 发送和接收数据的电平设定 (选择电平是否反相)
- 奇偶校验位的附加、奇偶校验功能
- 停止位的附加、停止位的检测功能

[中断功能]

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

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

以下通道的 UART 接收支持 SNOOZE 模式。SNOOZE 模式功能是指: 如果在 STOP 模式的状态下检测到 RxD 的输入, 就不需要 CPU 运行而接收数据。只有在接收时支持波特率调整功能的以下 UART 才能设定 SNOOZE 模式。

- 本产品: UART0

注 只有以下的 UART 支持 9 位数据长度。

- 本产品: UART0

UART0 使用 SAU0 的通道 0 和通道 1。

UART1 使用 SAU0 的通道 2 和通道 3。

UART2 使用 SAU1 的通道 0 和通道 1。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

各通道任意选择一个功能使用, 除了所选功能以外, 其他功能不能运行。例如, 在单元 0 的通道 0 和通道 1 使用 UART0 时, 不能使用 CSI00。但是, 在使用 UART0 的同时, 不同通道的通道 2 和通道 3 能使用 CSI11、UART1 或者 IIC11。

注意 当用作 UART 时, 发送方 (偶数通道) 和接收方 (奇数通道) 只能用于 UART。

UART 有以下 2 种通信运行:

- UART 发送 (参照 11.6.1)
- UART 接收 (参照 11.6.2)

11.6.1 UART 发送

UART 发送是 RL78 微控制器将数据异步发送到其他设备的运行。

UART 使用的 2 个通道中的偶数通道用于 UART 发送。

UART	UART0	UART1	UART2
对象通道	SAU0 的通道 0	SAU0 的通道 2	SAU1 的通道 0
使用的引脚	TxD0	TxD1	TxD2
中断	INTST0	INTST1	INTST2
	可选择传送结束中断（单次传送模式）或者缓冲器空中断（连续传送模式）。		
错误检测标志	无		
传送数据长度	7 位、8 位或者 9 位 ^{注 1}		
传送速率 ^{注 2}	Max. $f_{MCK}/6[\text{bps}]$ ($\text{SDRmn}[15:9] \geq 2$)、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$		
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： <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 1. 只有以下的 UART 支持 9 位数据长度。

- 本产品：UART0

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”）的范围内使用。

备注 1. f_{MCK} : 对象通道的运行时钟频率

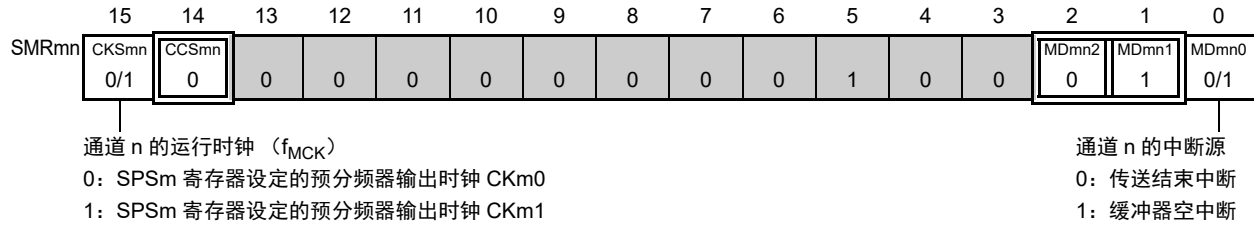
f_{CLK} : 系统时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=0、2) mn=00、02、10

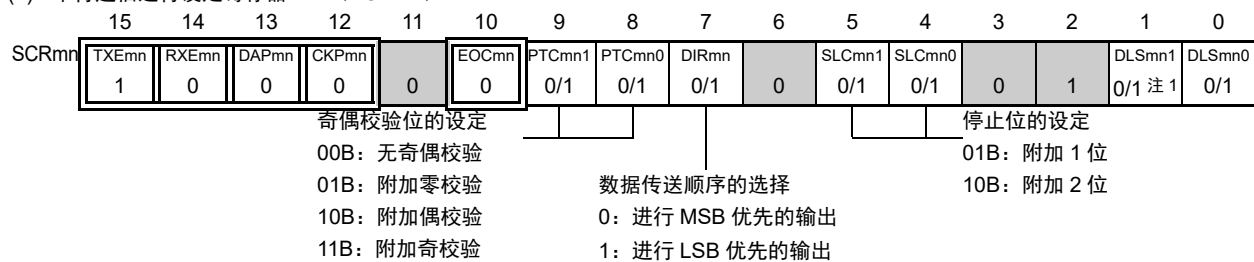
(1) 寄存器的设定

图 11-75 UART (UART0 ~ UART2) 的 UART 发送时的寄存器设定内容例子 (1/2)

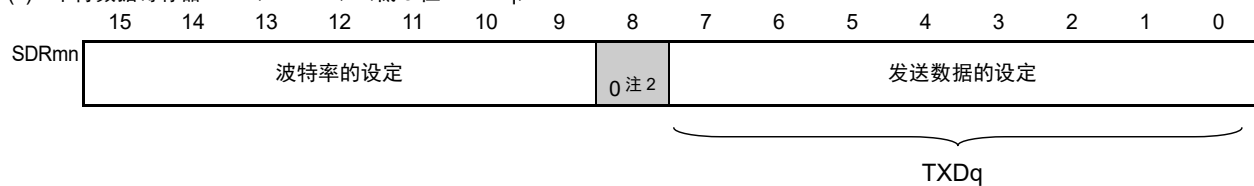
(a) 串行模式寄存器 mn (SMRmn)



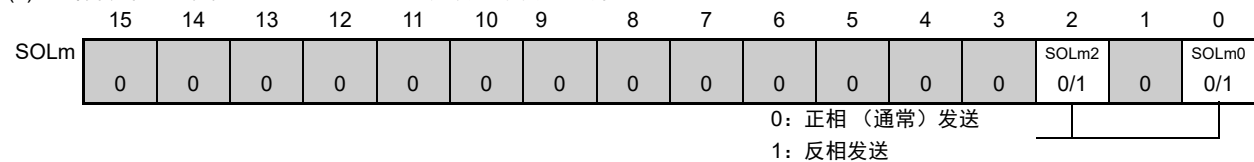
(b) 串行通信运行设定寄存器 mn (SCRmn)



(c) 串行数据寄存器 mn (SDRmn) (低 8 位: TXDq)



(d) 串行输出电平寄存器 m (SOLm) 只设定对象通道的位。



注 1. 只限于 SCR00 寄存器, 其他固定为“1”。

2. 当进行 9 位数据长度的通信时, SDRm0 寄存器的 bit0 ~ 8 为发送数据的设定区。只有以下的 UART 才能进行 9 位数据长度的通信。

- 本产品: UART0

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART 号 (q=0 ~ 2) mn=00、02、10

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 11-75 UART (UART0 ~ UART2) 的 UART 发送时的寄存器设定内容例子 (2/2)

(e) 串行输出寄存器 m (SOm) 只设定对象通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
					×	×	×	×					×	0/1 注	×	0/1 注

0: 串行数据输出值为“0”
1: 串行数据输出值为“1”

(f) 串行输出允许寄存器 m (SOEm) 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
													×	0/1	×	0/1

(g) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													×	0/1	×	0/1

注 在开始发送前，当对应通道的 SOLmn 位为“0”时，必须置“1”；当对应通道的 SOLmn 位为“1”时，必须置“0”。在通信过程中，值因通信数据而变。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0、2) mn=00、02、10

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 11-76 UART 发送的初始设定步骤

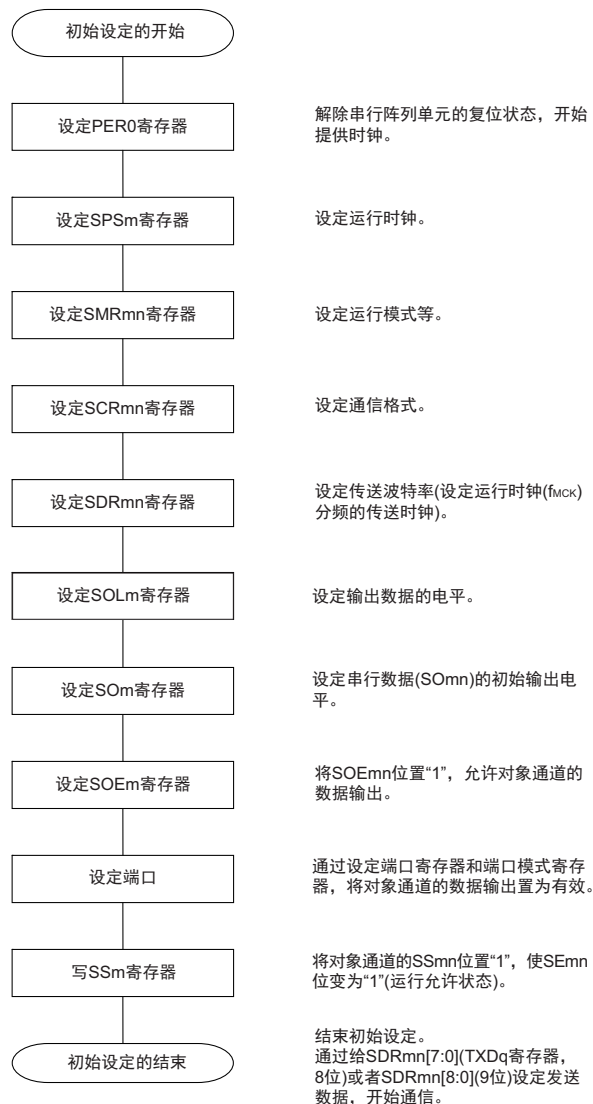


图 11-77 UART 发送的中止步骤

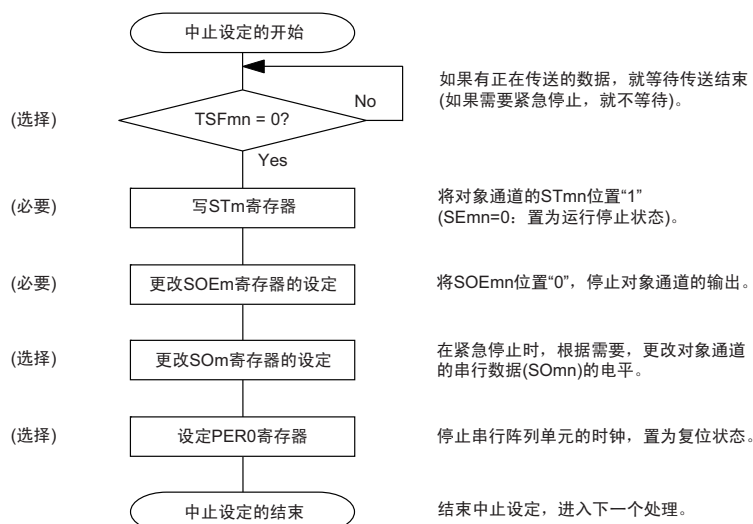
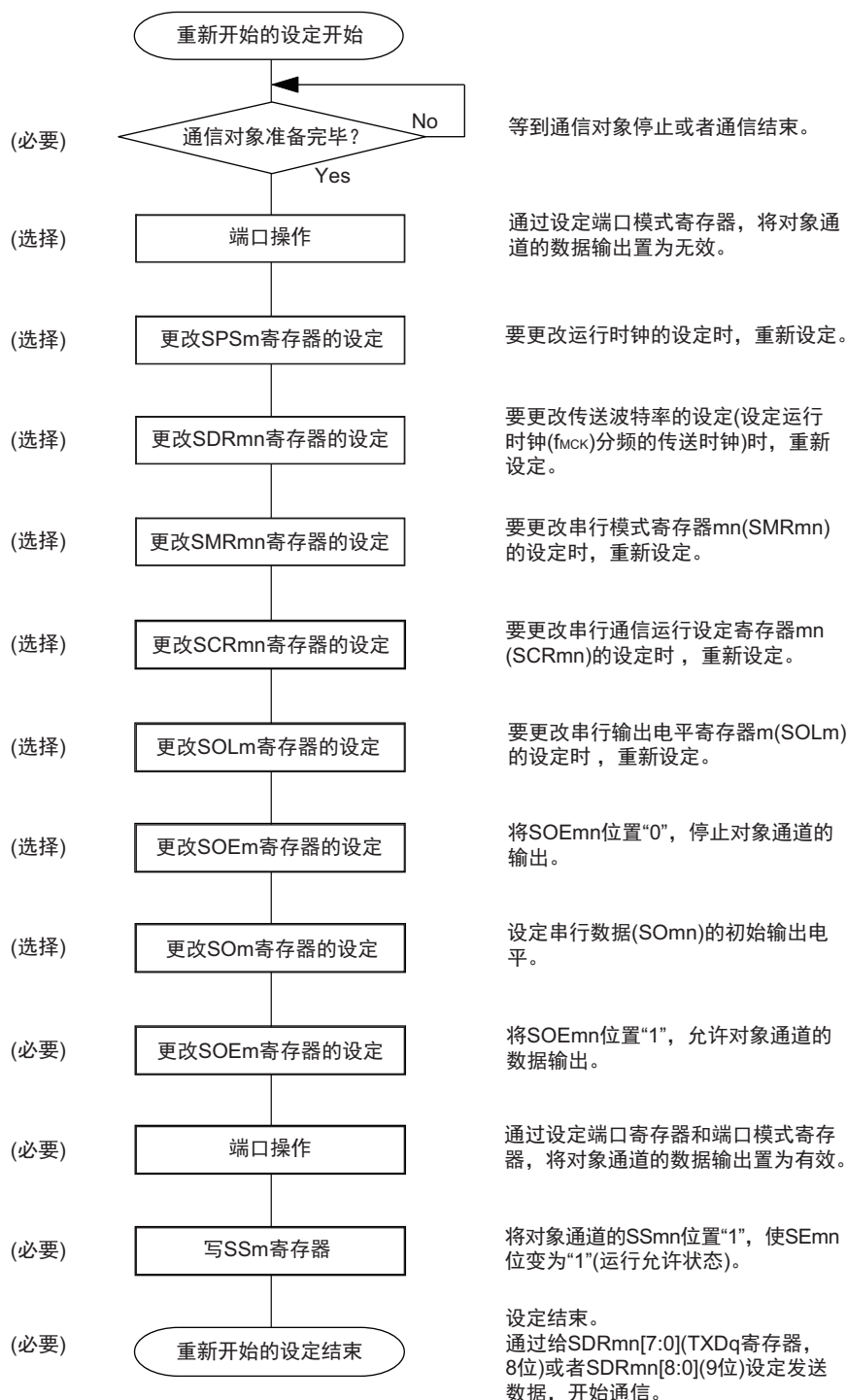


图 11-78 重新开始 UART 发送的设定步骤



备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始的设定。

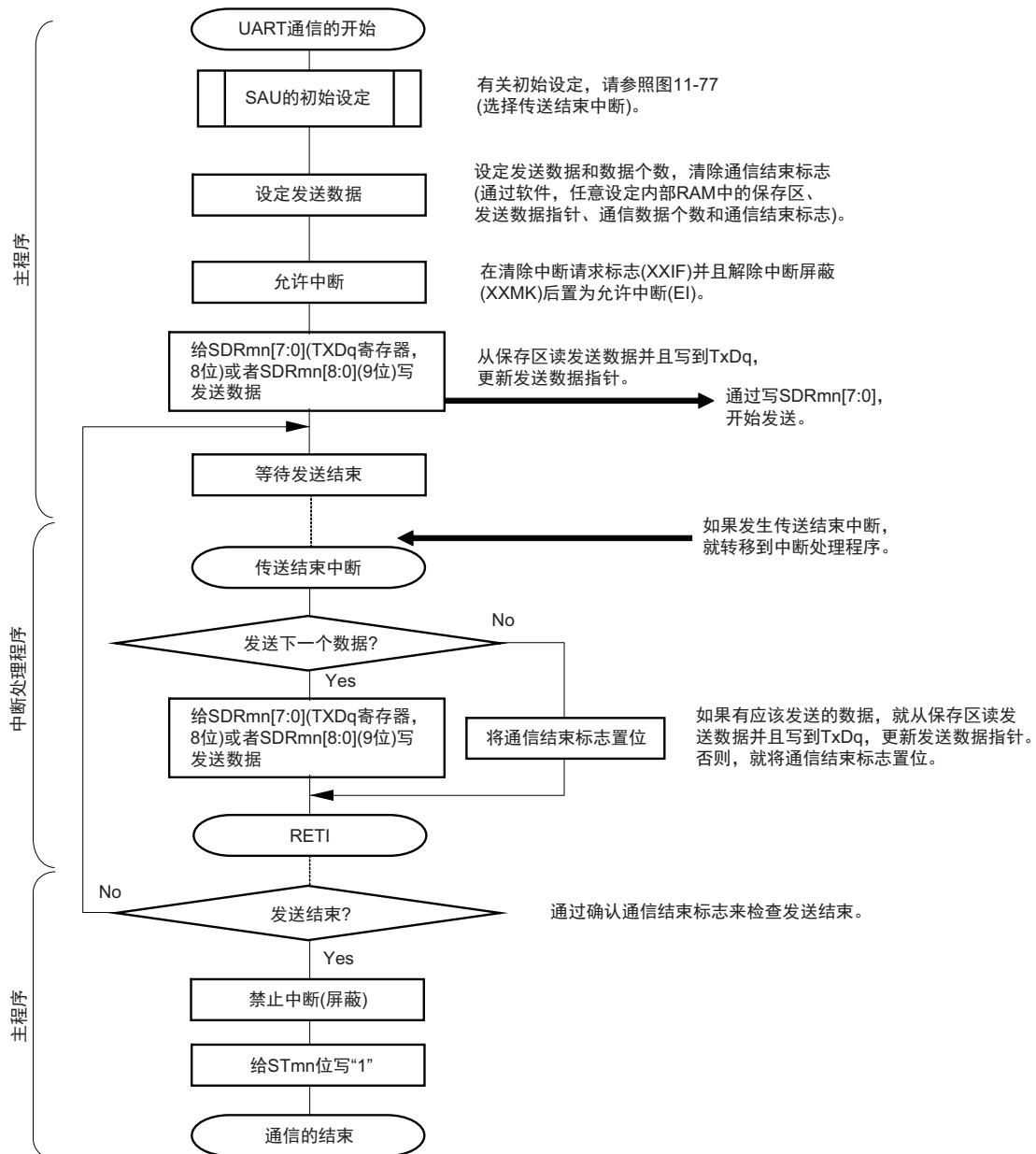
(3) 处理流程（单次发送模式）

图 11-79 UART 发送（单次发送模式）的时序图



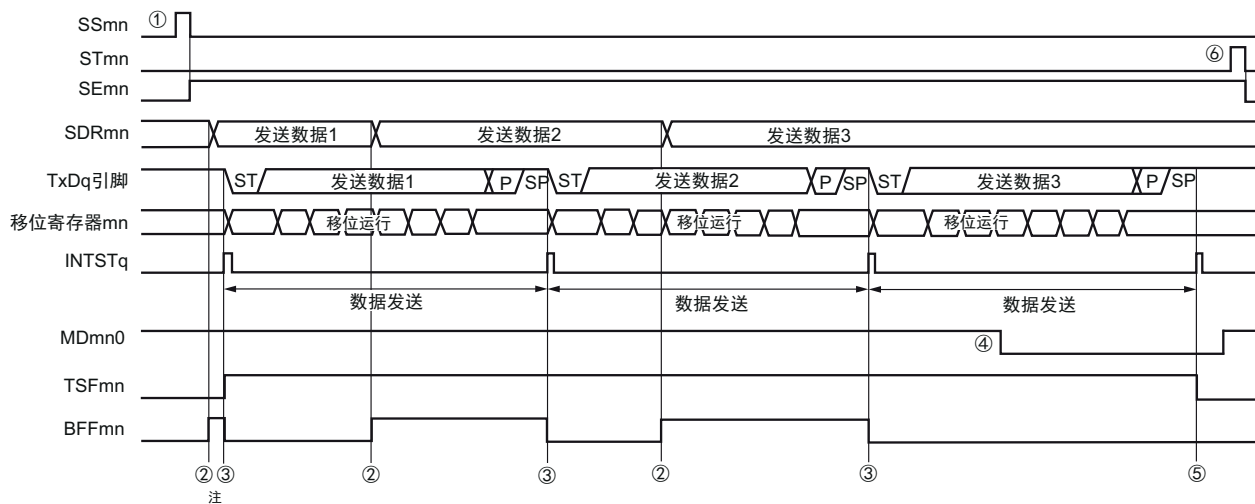
备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART 号 (q=0~2) mn=00、02、10

图 11-80 UART 发送（单次发送模式）的流程图



(4) 处理流程（连续发送模式）

图 11-81 UART 发送（连续发送模式）的时序图

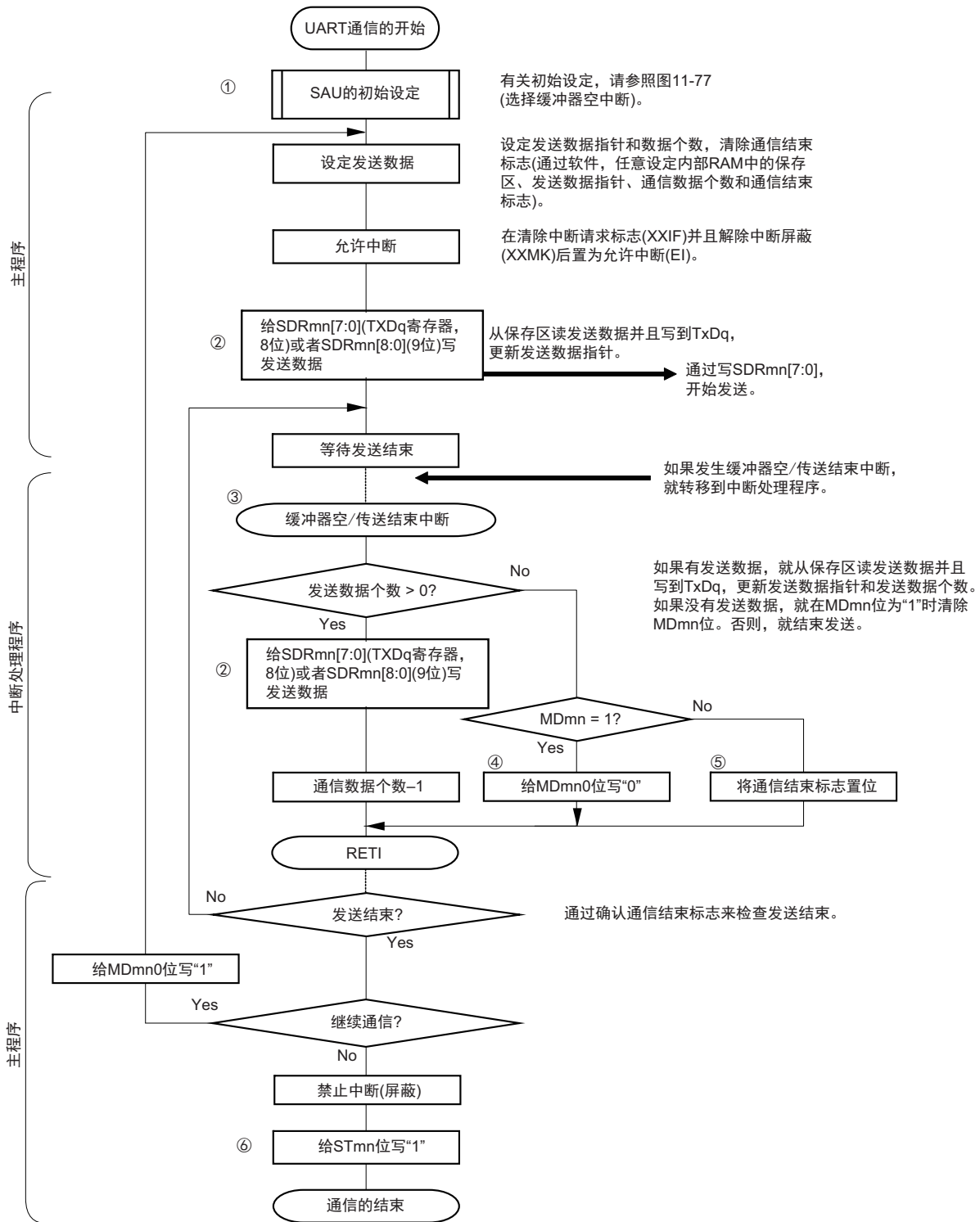


注 如果在串行状态寄存器 mn（SSRmn）的 BFFmn 位为“1”期间（有效数据保存在串行数据寄存器 mn（SDRmn）时）给 SDRmn 寄存器写发送数据，就重写发送数据。

注意 即使在运行中也能改写串行模式寄存器 mn（SMRmn）的 MDmn0 位。但是，为了能赶上最后发送数据的传送结束中断，必须在开始传送最后一位之前进行改写。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) q: UART 号 (q=0~2) mn=00、02、10

图 11-82 UART 发送（连续发送模式）的流程图



备注 图中的①~⑥对应“图 11-81 UART 发送（连续发送模式）的时序图”中的①~⑥。

11.6.2 UART 接收

UART 接收是 RL78 微控制器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	UART1	UART2
对象通道	SAU0 的通道 1	SAU0 的通道 3	SAU1 的通道 1
使用的引脚	RxD0	RxD1	RxD2
中断	INTSR0	INTSR1	INTSR2
	只限于传送结束中断（禁止设定缓冲器空中断）。		
错误中断	INTSRE0	INTSRE1	INTSRE2
错误检测标志	<ul style="list-style-type: none"> • 帧错误检测标志（FEFmn） • 奇偶校验错误检测标志（PEFmn） • 溢出错误检测标志（OVFmn） 		
传送数据长度	7 位、8 位或者 9 位注 1		
传送速率	Max. $f_{MCK}/6[\text{bps}]$ ($SDRmn[15:9] \geq 2$)、Min. $f_{CLK}/(2 \times 2^{15} \times 128)[\text{bps}]$		
数据相位	正相输出（默认值：高电平）。 反相输出（默认值：低电平）。		
奇偶校验位	可选择以下内容： <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 1. 只有以下的 UART 支持 9 位数据长度。

- 本产品：UART0

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”）的范围内使用。

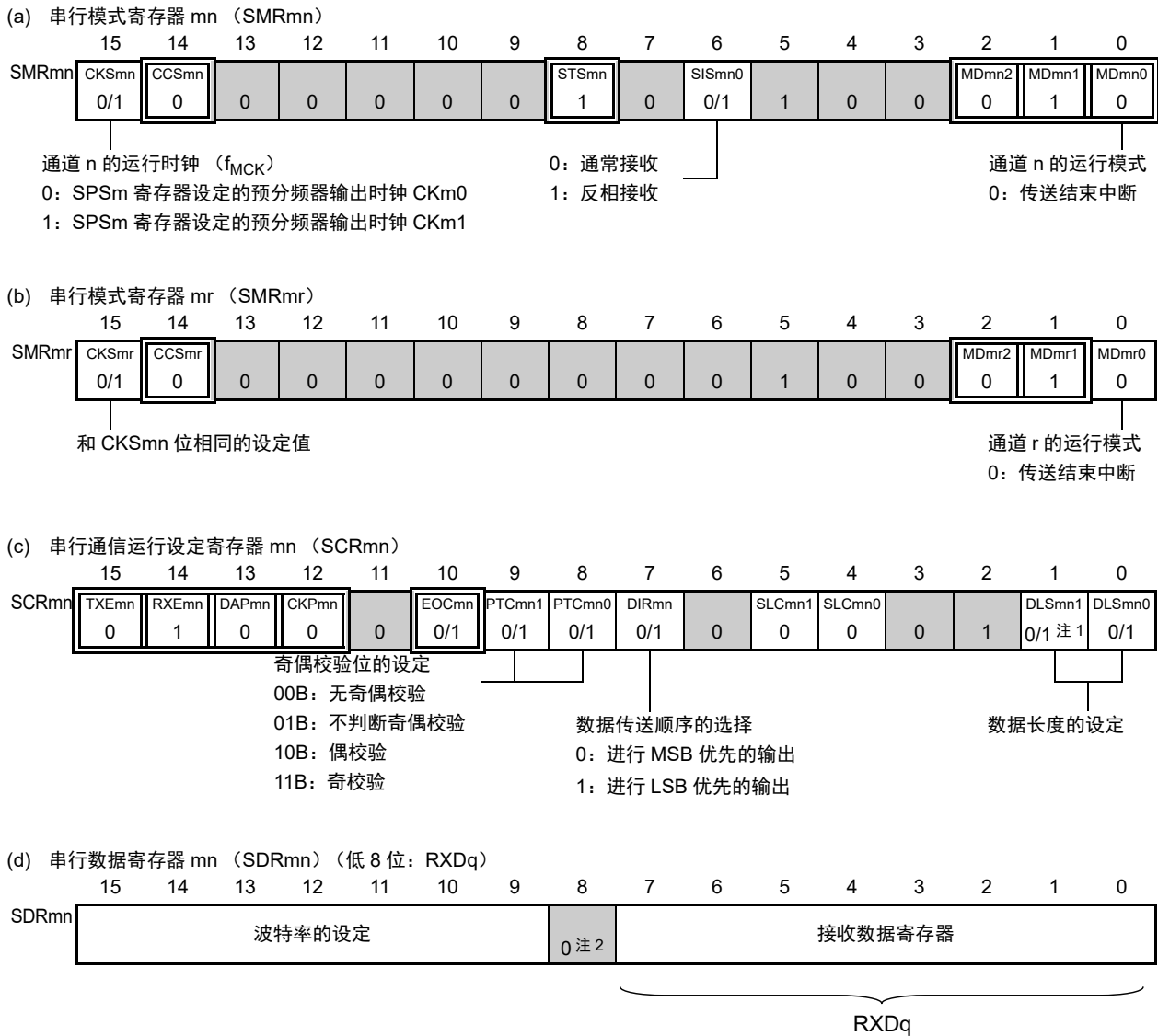
备注 1. f_{MCK} : 对象通道的运行时钟频率

f_{CLK} : 系统时钟频率

2. m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11

(1) 寄存器的设定

图 11-83 UART (UART0 ~ UART2) 的 UART 接收时的寄存器设定内容例子 (1/2)



注 1. 只限于 SCR01 寄存器，其他固定为“1”。

2. 当进行 9 位数据长度的通信时，SDRm1 寄存器的 bit0 ~ 8 为发送数据的设定区。只有以下的 UART 才能进行 9 位数据长度的通信。

- 本产品: UART0

注意 在 UART 接收时，还必须将与通道 n 成对的通道 r 的 SMRmr 寄存器设定为 UART 发送模式。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11 r: 通道号 (r=n-1) q: UART 号 (q=0 ~ 2)

2. □: 在 UART 接收模式中为固定设定。■: 不能设定 (设定初始值)。

0/1: 根据用户的用途置“0”或者“1”。

图 11-83 UART (UART0 ~ UART2) 的 UART 接收时的寄存器设定内容例子 (2/2)

(e) 串行输出寄存器 m (SOm) 在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3	CKOm2	CKOm1	CKOm0	0	0	0	0	SOm3	SOm2	SOm1	SOm0
					×	×	×	×					×	×	×	×

(f) 串行输出允许寄存器 m (SOEm) 在此模式中不使用。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3	SOEm2	SOEm1	SOEm0
													×	×	×	×

(g) 串行通道开始寄存器 m (SSm) 只将对象通道的位置“1”。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0
													0/1	×	0/1	×

备注 1. m: 单元号 (m=0、1)

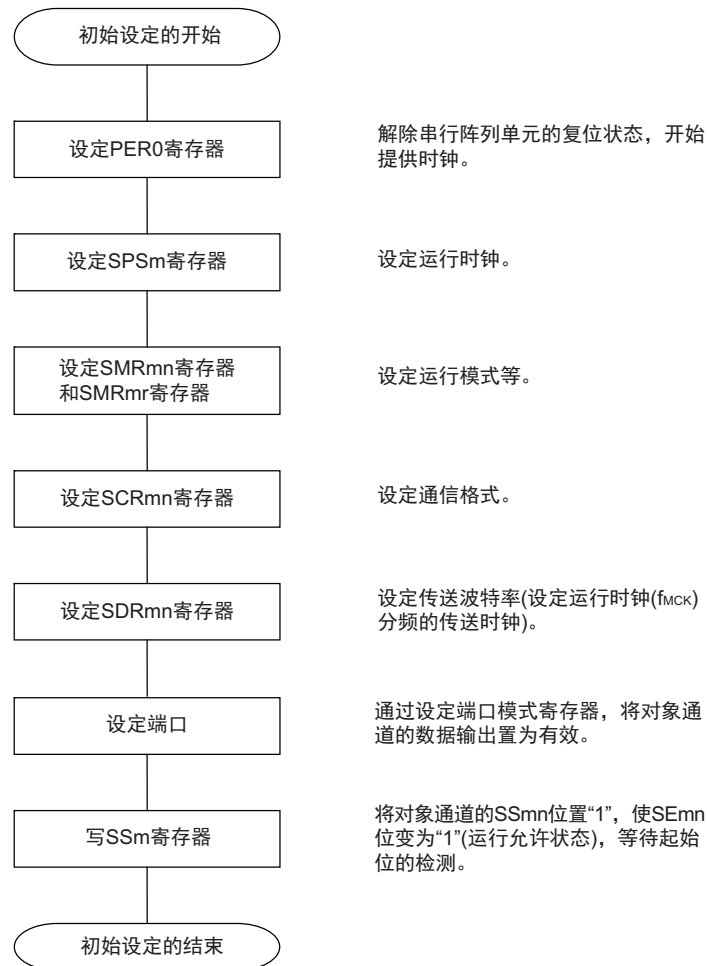
2. : 不能设定 (设定初始值)。

×: 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 11-84 UART 接收的初始设定步骤



注意 必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 f_{MCK} 时钟，然后将 SSmn 位置“1”。

图 11-85 UART 接收的中止步骤

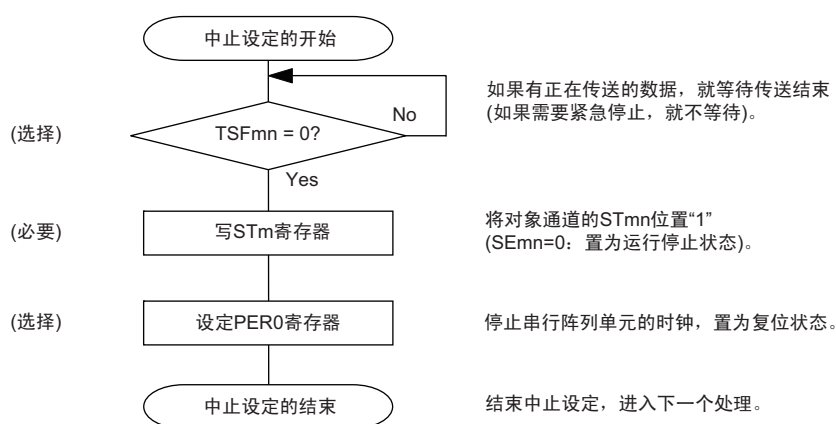
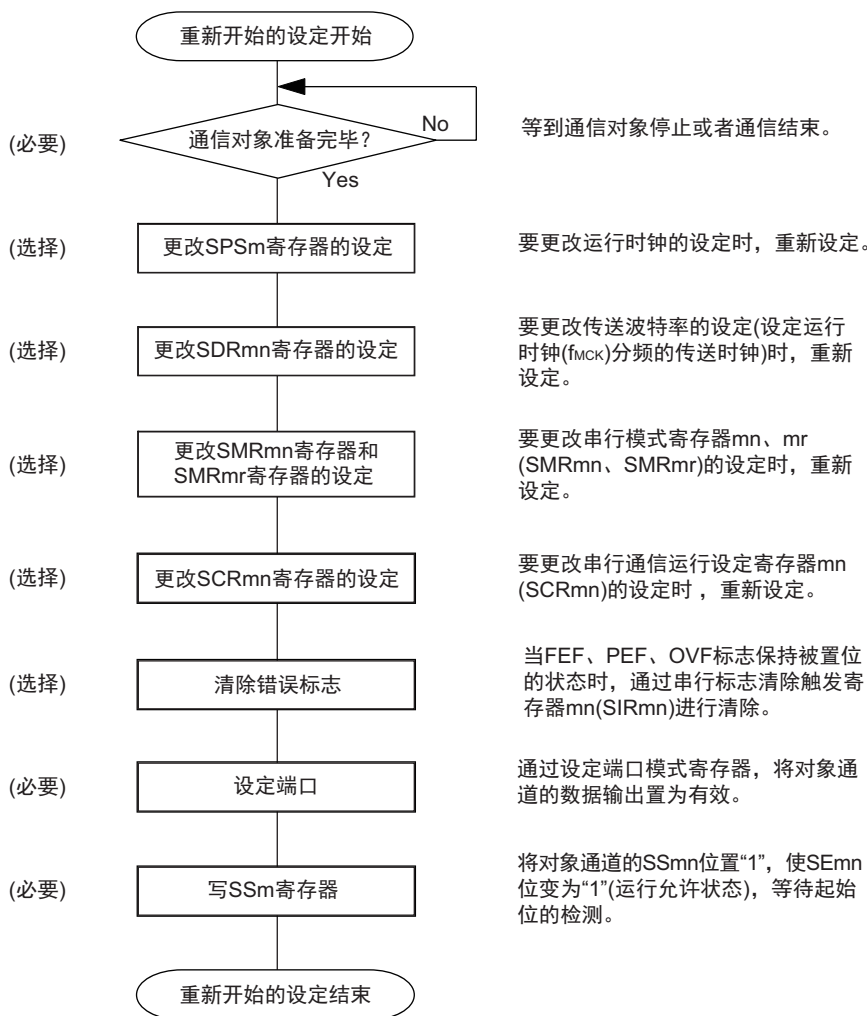


图 11-86 重新开始 UART 接收的设定步骤

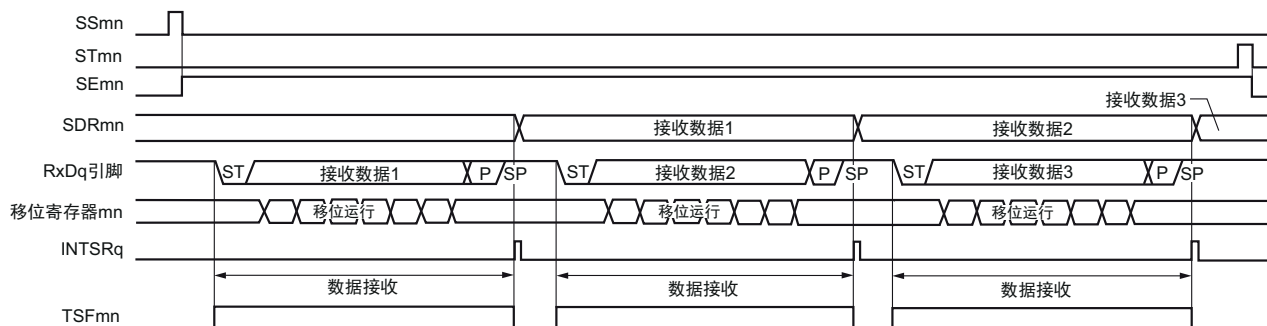


注意 必须在将 SCRmn 寄存器的 RXEmn 位置“1”后至少间隔 4 个 f_{MCK} 时钟，然后将 SSmn 位置“1”。

备注 如果在中止设定中改写 PER0 来停止提供时钟，就必须在等到通信对象停止或者通信结束后进行初始设定而不是进行重新开始的设定。

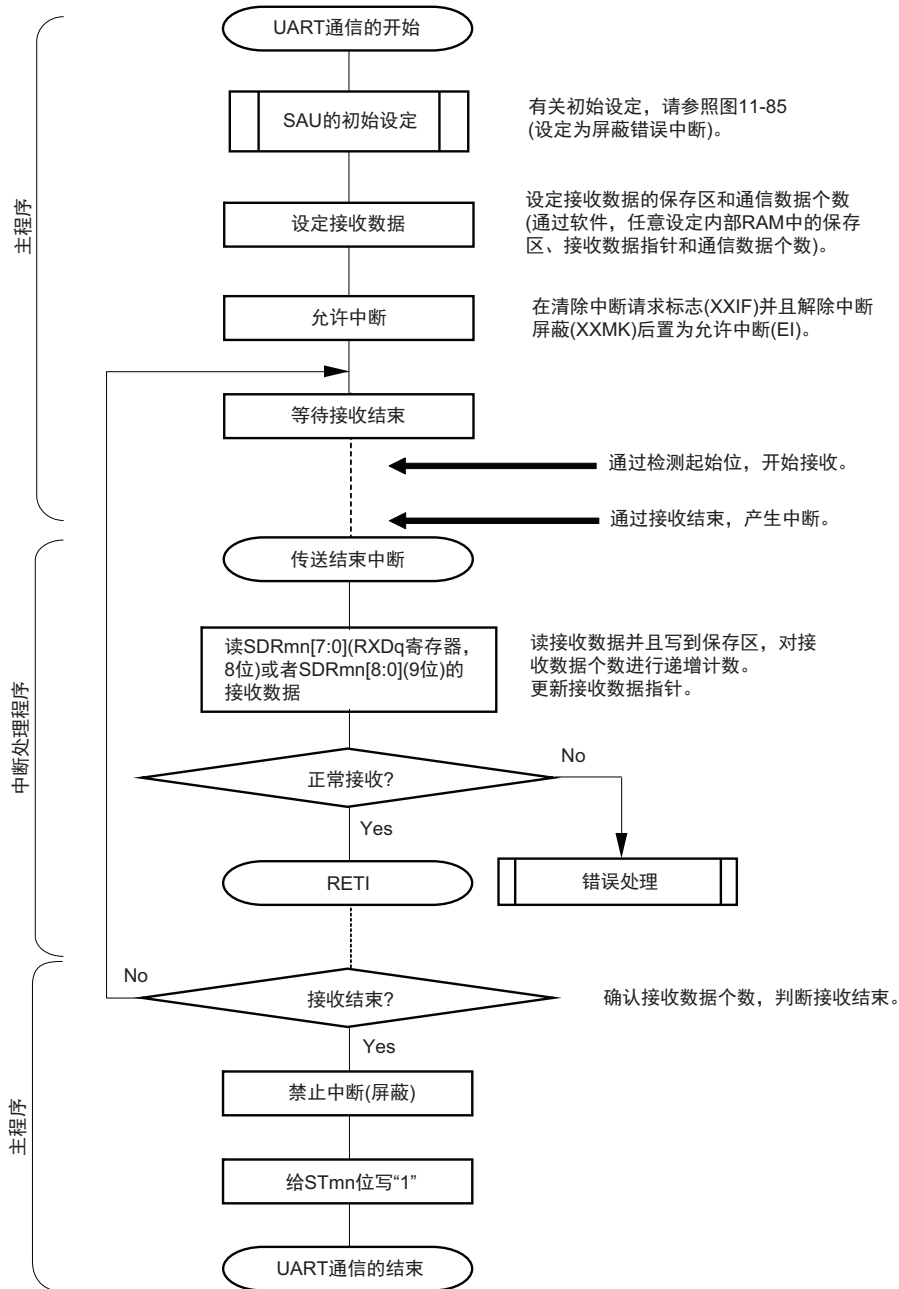
(3) 处理流程

图 11-87 UART 接收的时序图



备注 m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11 r: 通道号 (r=n-1) q: UART 号 (q=0 ~ 2)

图 11-88 UART 接收的流程图



11.6.3 SNOOZE 模式功能

这是在 STOP 模式中通过检测 RxDq 引脚的输入使 UART 进行接收的模式。通常，在 STOP 模式中 UART 停止通信，但是如果使用 SNOOZE 模式功能，就能在 CPU 不运行的状态下进行 UART 的接收。只有以下的 UART 才能设定 SNOOZE 模式。

- 本产品：UART0

要在 SNOOZE 模式中使用 UARTq 时，必须在转移到 STOP 模式前进行以下的设定（参照“图 11-91 SNOOZE 模式运行（EOCm1=0、SSEC0=0/1 或者 EOCm1=1、SSEC0=0）的流程图和图 11-93 SNOOZE 模式运行（EOCm1=1、SSEC0=1）的流程图”）。

- 在 SNOOZE 模式中，需要更改 UART 接收波特率的设定（更改为和通常运行时不同的值）。必须参照表 11-3，设定 SPSm 寄存器和 SDRmn 寄存器[15:9]。
- 设定 EOCmn 位和 SSEC0 位。能设定允许或者停止在发生通信错误时产生错误中断（INTSRE0）。
- 必须在即将要转移到 STOP 模式前将串行待机控制寄存器 0（SSC0）的 SWC0 位置“1”。在初始设定结束后，将串行通道开始寄存器 m（SSm）的 SSm1 位置“1”。
- 如果在转移到 STOP 模式后检测到 RxDq 输入的起始位，UARTq 就开始接收。

注意 1. 只有在选择高速内部振荡器时钟（ f_{IH} ）作为 f_{CLK} 时才能使用 SNOOZE 模式功能。

2. SNOOZE 模式中的传送速率只为 4800bps。
3. 如果将 SWC0 位置“1”，就只能在 STOP 模式中开始接收时使用 UARTq。当其他 SNOOZE 模式功能和中断同时使用并且在以下非 STOP 模式的状态下开始接收时，可能无法正常接收数据而发生帧错误或者奇偶校验错误。
 - 在将 SWC0 位置“1”后并且在转移到 STOP 模式前开始接收的情况
 - 在其他 SNOOZE 模式中开始接收的情况
 - 在通过中断等从 STOP 模式返回到通常运行后并且在将 SWC0 位置“0”前开始接收的情况
4. 如果将 SSEC0 位置“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFmn、FEFmn、OVFmn 标志置位，也不产生错误中断（INTSREq）。因此，在 SSEC0 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFmn、FEFmn、OVFmn 标志并且读 SDRm1 寄存器的 bit7 ~ 0（RxDq）。
5. 通过检测 RxDq 引脚的有效边沿，转移到 SNOOZE 模式。

如果接受到无法检测到输入起始位的短脉冲，就可能不开始 UART 接收而继续保持 SNOOZE 模式。此时，可能在下一轮的 UART 接收过程中无法正常接收数据而发生帧错误或者奇偶校验错误。

表 11-3 SNOOZE 模式中的 UART 接收波特率的设定

高速内部振荡器 (f_{IH})	SNOOZE 模式中的 UART 接收波特率			
	波特率 4800bps			
	运行时钟 (f_{MCK})	SDRmn[15:9]	最大容许值	最小容许值
24MHz \pm 1.0% 注	$f_{CLK}/2^5$	79	1.60%	-2.18%
16MHz \pm 1.0% 注	$f_{CLK}/2^4$	105	2.27%	-1.53%
12MHz \pm 1.0% 注	$f_{CLK}/2^4$	79	1.60%	-2.19%
8MHz \pm 1.0% 注	$f_{CLK}/2^3$	105	2.27%	-1.53%
6MHz \pm 1.0% 注	$f_{CLK}/2^3$	79	1.60%	-2.19%
4MHz \pm 1.0% 注	$f_{CLK}/2^2$	105	2.27%	-1.53%
3MHz \pm 1.0% 注	$f_{CLK}/2^2$	79	1.60%	-2.19%
2MHz \pm 1.0% 注	$f_{CLK}/2$	105	2.27%	-1.54%
1MHz \pm 1.0% 注	f_{CLK}	105	2.27%	-1.57%

注 当高速内部振荡器的频率精度为 $\pm 1.5\%$ 、 $\pm 2.0\%$ 时，以下的容许范围会变窄。

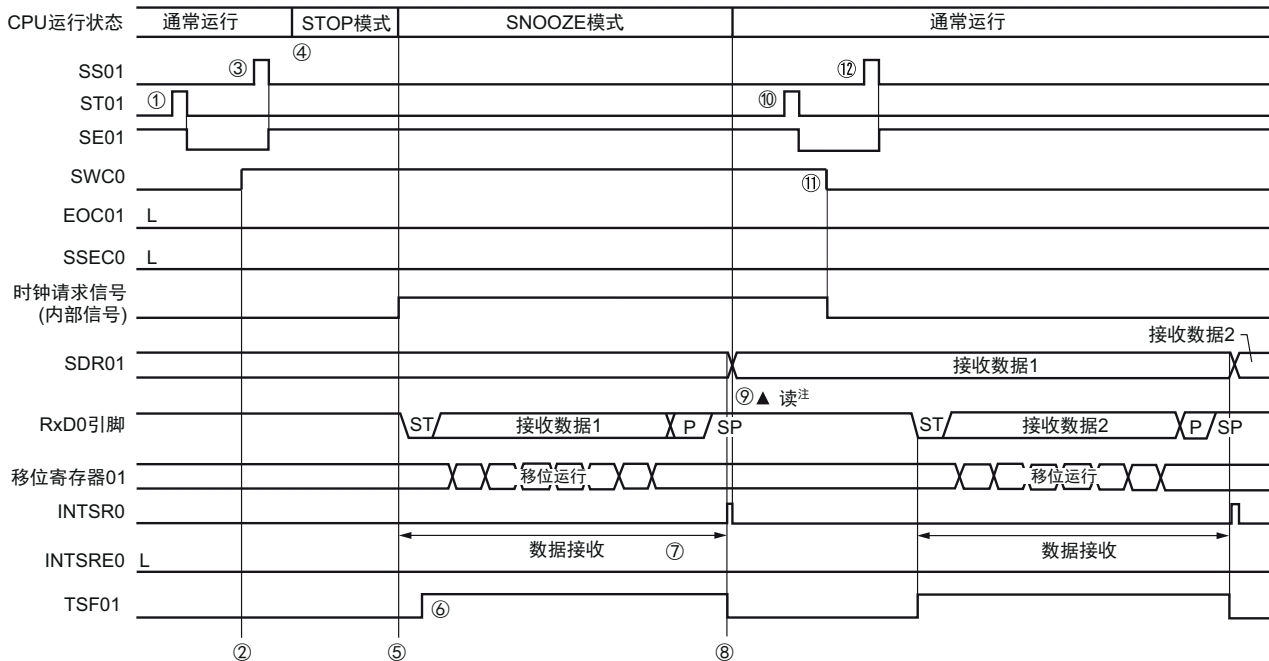
- 在 $f_{IH}\pm 1.5\%$ 的情况下，必须将上表的最大容许值设定为 -0.5% ，并且将最小容许值设定为 $+0.5\%$ 。
- 在 $f_{IH}\pm 2.0\%$ 的情况下，必须将上表的最大容许值设定为 -1.0% ，并且将最小容许值设定为 $+1.0\%$ 。

备注 最大容许值和最小容许值是 UART 接收时的波特率的容许值。
必须将发送方的波特率设定在此范围内。

(1) SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1)

因为 EOCm1 位为“0”，所以与 SSEC0 位的设定无关，即使发生通信错误也不产生错误中断 (INTSREq)。但是，产生传送结束中断 (INTSRq)。

图 11-89 SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1) 的时序图



注 必须在 SWC0 位为“1”的状态下读接收数据。

注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWC0 位（解除 SNOOZE 模式）。

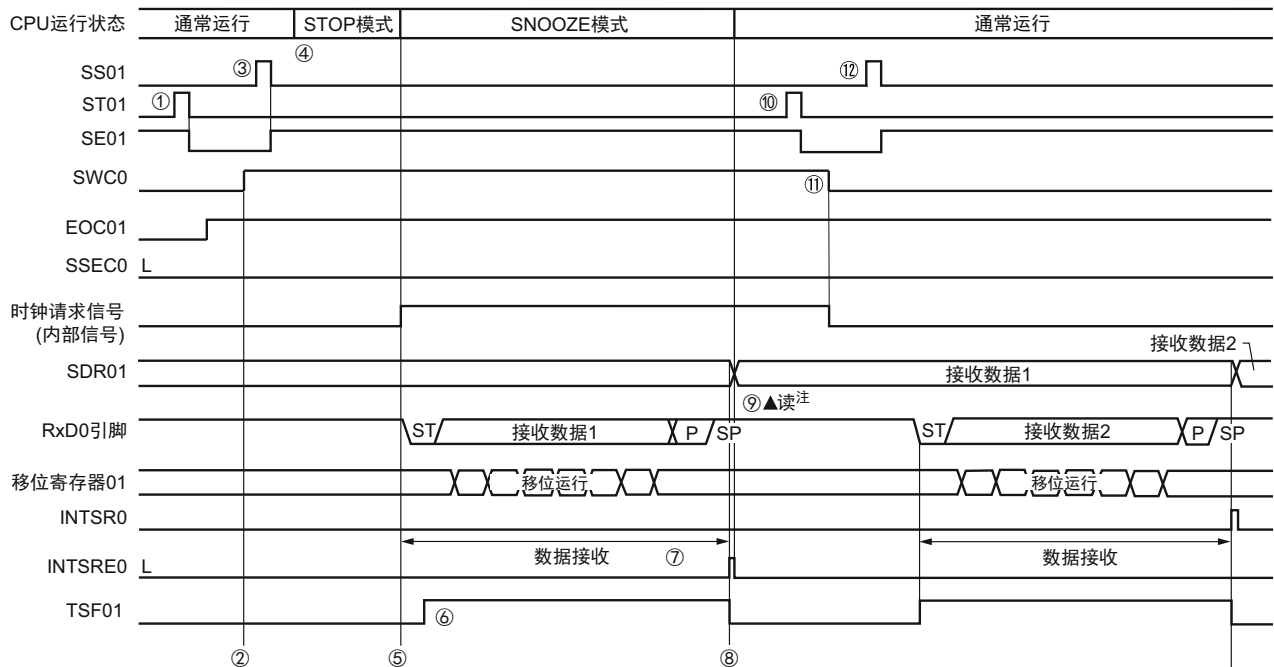
备注 1. 图中的①~⑫对应“图 11-91 SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1 或者 EOCm1=1、SSEC0=0) 的流程图”中的①~⑫。

2. 本产品：m=0，q=0

(2) SNOOZE 模式运行 (EOCm1=1、SSEC0=0: 允许产生错误中断 (INTSREq))

因为 EOCm1 位为“1”并且 SSEC0 位为“0”，所以在发生通信错误时产生错误中断 (INTSREq)。

图 11-90 SNOOZE 模式运行 (EOCm1=1、SSEC0=0) 的时序图



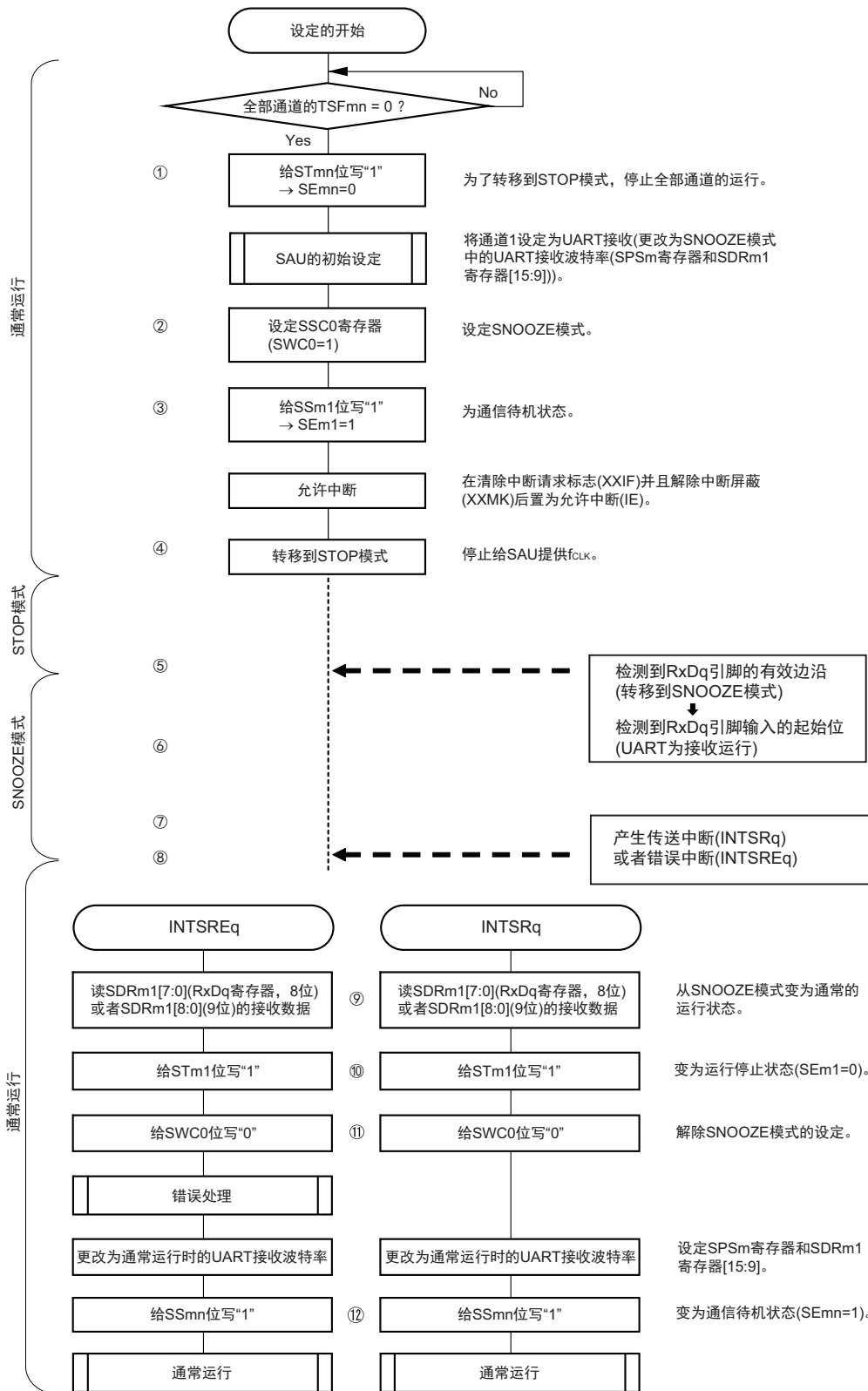
注 必须在 SWC0 位为“1”的状态下读接收数据。

注意 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWC0 位（解除 SNOOZE 模式）。

备注 1. 图中的①~⑫对应“图 11-91 SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1 或者 EOCm1=1、SSEC0=0) 的流程图”中的①~⑫。

2. 本产品: m=0, q=0

图 11-91 SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1 或者 EOCm1=1、SSEC0=0) 的流程图



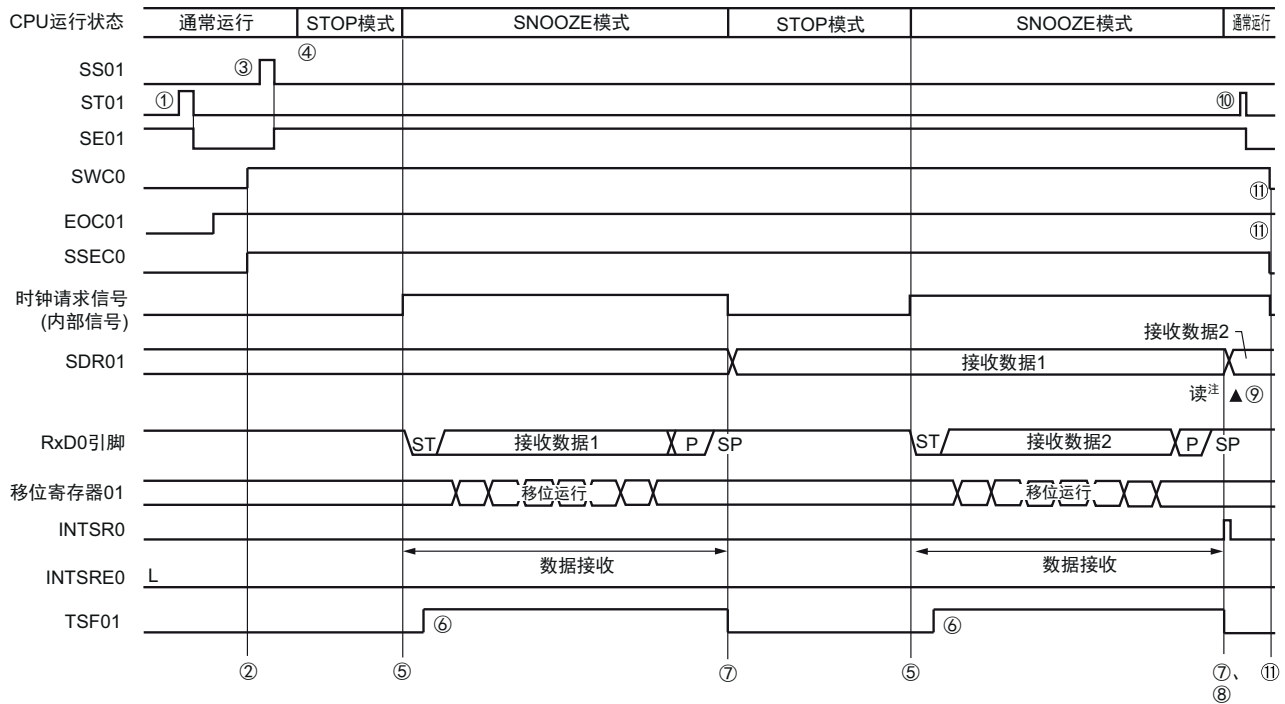
备注 1. 图中的①~⑫对应“图 11-89 SNOOZE 模式运行 (EOCm1=0、SSEC0=0/1) 的时序图”和“图 11-90 SNOOZE 模式运行 (EOCm1=1、SSEC0=0) 的时序图”中的①~⑫。

2. 本产品: m=0, q=0, n=0 ~ 3

(3) SNOOZE 模式运行 (EOCm1=1、SSEC0=1: 停止产生错误中断 (INTSREq))

因为 EOCm1 位为“1”并且 SSEC0 位为“1”，所以在发生通信错误时不产生错误中断 (INTSREq)。

图 11-92 SNOOZE 模式运行 (EOCm1=1、SSEC0=1) 的时序图



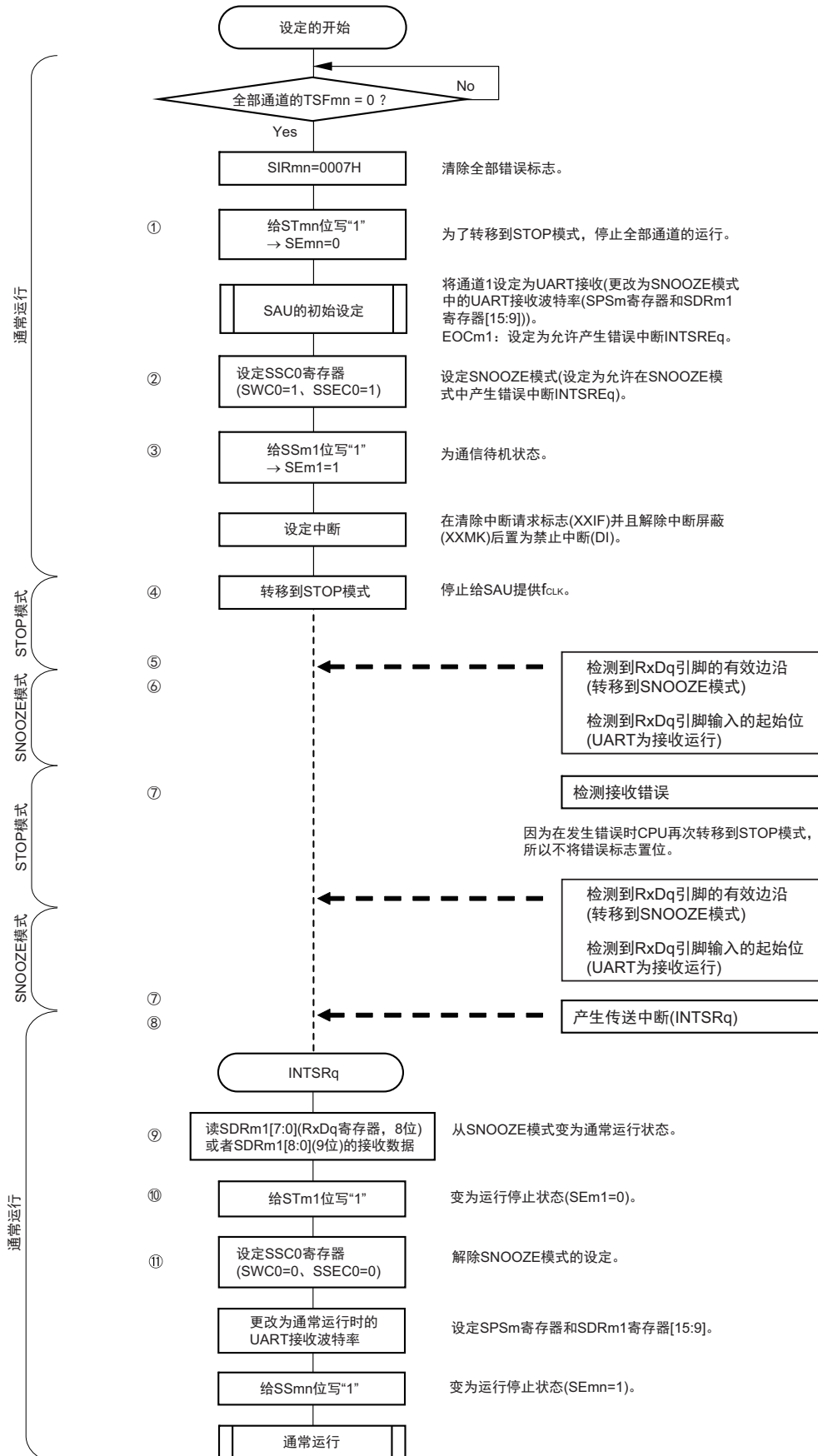
注 必须在 SWC0 位为“1”的状态下读接收数据。

- 注意 1. 在向 SNOOZE 模式转移前或者在 SNOOZE 模式中接收结束后，必须将 STm1 位置“1”（清除 SEm1 位并且停止运行），而且还必须在接收结束后清除 SWC0 位（解除 SNOOZE 模式）。
2. 如果 SSEC0 位为“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位，也不产生错误中断 (INTSREq)。因此，在 SSEC0 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0] (RXDq 寄存器，8 位) 或者 SDRm1[8:0] (9 位)。

备注 1. 图中的①~⑪对应“图 11-93 SNOOZE 模式运行 (EOCm1=1、SSEC0=1) 的流程图”中的①~⑪。

2. 本产品: m=0, q=0

图 11-93 SNOOZE 模式运行 (EOCm1=1、SSEC0=1) 的流程图



注意 如果 SSEC0 位为“1”，就在发生奇偶校验错误、帧错误或者溢出错误时不将 PEFm1、FEFm1、OVFm1 标志置位，也不产生错误中断（INTSREQ）。因此，在 SSEC0 位为“1”的情况下使用时，必须在将 SWC0 位置“1”前清除 PEFm1、FEFm1、OVFm1 标志并且读 SDRm1[7:0]（RXDq 寄存器，8 位）或者 SDRm1[8:0]（9 位）。

备注 1. 图中的①~⑪对应“图 11-92 SNOOZE 模式运行（EOCm1=1、SSEC0=1）的时序图”中的①~⑪。

2. 本产品：m=0，q=0，n=0~3

11.6.4 波特率的计算

(1) 波特率的计算式

UART（UART0 ~ UART2）通信的波特率能用以下计算式进行计算：

$$\text{（波特率）} = \{ \text{对象通道的运行时钟（} f_{\text{MCK}} \text{）频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [bps]}$$

注意 禁止将串行数据寄存器 mn（SDRmn）的 SDRmn[15:9] 设定为“0000000B”和“0000001B”。

备注 1. 因为在使用 UART 时 SDRmn[15:9] 的值为 SDRmn 寄存器的 bit15~9 的值（000010B~111111B），所以为 2~127。

2. m：单元号（m=0、1） n：通道号（n=0~3） mn=00、03、10

运行时钟（ f_{MCK} ）取决于串行时钟选择寄存器 m（SPSm）和串行模式寄存器 mn（SMRmn）的 bit15（CKSmn 位）。

表 11-4 UART 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f_{MCK}) 注	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	$f_{CLK}=20\text{MHz}$ 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 m（STm）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

(2) 发送时的波特率误差

UART (UART0 ~ UART2) 通信发送时的波特率误差能用以下计算式进行计算, 必须将发送方的波特率设定在接收方的波特率容许范围内。

$$\text{(波特率误差)} = (\text{波特率的计算值}) \div (\text{目标波特率的值}) \times 100 - 100 [\%]$$

$f_{\text{CLK}}=20\text{MHz}$ 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})	SDRmn[15:9]	波特率的计算值	与目标波特率的误差
300bps	$f_{\text{CLK}}/2^9$	103	300.48bps	+0.16%
600bps	$f_{\text{CLK}}/2^8$	103	600.96bps	+0.16%
1200bps	$f_{\text{CLK}}/2^7$	103	1201.92bps	+0.16%
2400bps	$f_{\text{CLK}}/2^6$	103	2403.85bps	+0.16%
4800bps	$f_{\text{CLK}}/2^5$	103	4807.69bps	+0.16%
9600bps	$f_{\text{CLK}}/2^4$	103	9615.38bps	+0.16%
19200bps	$f_{\text{CLK}}/2^3$	103	19230.8bps	+0.16%
31250bps	$f_{\text{CLK}}/2^3$	63	31250.0bps	$\pm 0.0\%$
38400bps	$f_{\text{CLK}}/2^2$	103	38461.5bps	+0.16%
76800bps	$f_{\text{CLK}}/2$	103	76923.1bps	+0.16%
153600bps	f_{CLK}	103	153846bps	+0.16%
312500bps	f_{CLK}	50	312500bps	$\pm 0.0\%$

备注 m: 单元号 (m=0、1) n: 通道号 (n=0、2) mn=00、02、10

(3) 接收时的波特率容许范围

UART (UART0 ~ UART2) 通信接收时的波特率容许范围能用以下计算式进行计算, 必须将发送方的波特率设定在接收方的波特率容许范围内。

$$\text{(可接收的最大波特率)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{(可接收的最小波特率)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: 接收方的波特率的计算值 (参照“11.6.4(1) 波特率的计算式”)

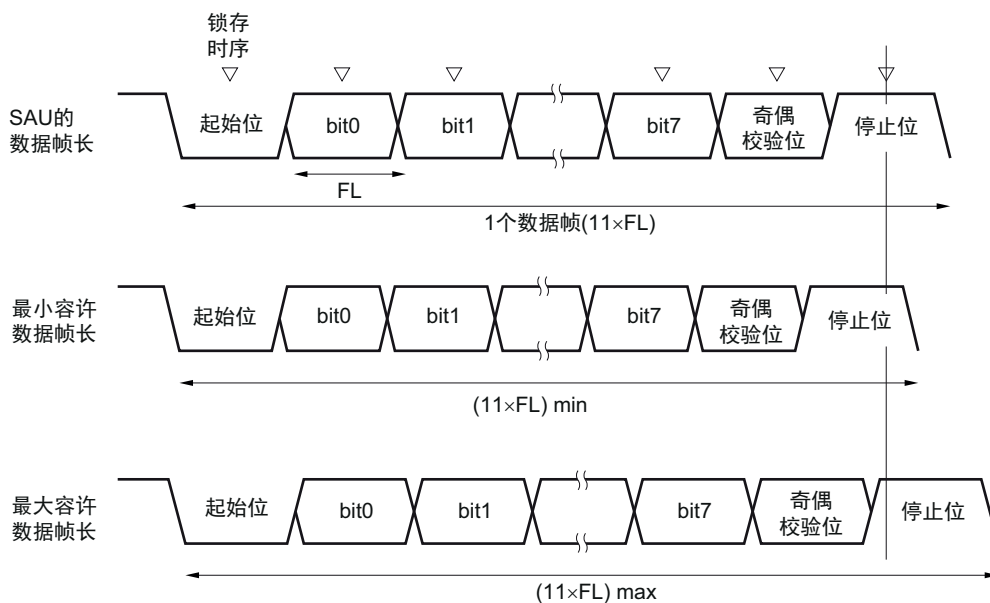
k: SDRmn[15:9]+1

Nfr: 1 个数据的数据帧长 [位]

= (起始位) + (数据长度) + (奇偶校验位) + (停止位)

备注 m: 单元号 (m=0、1) n: 通道号 (n=1、3) mn=01、03、11、13

图 11-94 接收时的波特率容许范围 (1 个数据的数据帧长 = 11 位的情况)



如图 11-94 所示, 在检测到起始位后, 接收数据的锁存时序取决于串行数据寄存器 mn (SDRmn) 的 bit15 ~ 9 设定的分频比。如果最后的数据 (停止位) 能赶上此锁存时序, 就能正常接收。

11.6.5 在 UART (UART0 ~ UART2) 通信过程中发生错误时的处理步骤

在 UART (UART0 ~ UART2) 通信过程中发生错误时的处理步骤如图 11-95 和图 11-96 所示。

图 11-95 发生奇偶校验错误或者溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。	→ SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SIRmn) 写“1”。	→ 清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

图 11-96 发生帧错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。	→ SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器 mn (SIRmn)。	→ 清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。
将串行通道停止寄存器 m (STm) 的 STmn 位置“1”。	→ 串行通道允许状态寄存器 m (SEm) 的 SEMn 位为“0”并且通道 n 为运行停止状态。	
与通信方进行同步处理。		因为起始位偏移，所以可认为发生了帧错误。因此，需要与通信方重新取得同步，重新开始通信。
将串行通道开始寄存器 m (SSm) 的 SSmn 位置“1”。	→ 串行通道允许状态寄存器 m (SEm) 的 SEMn 位为“1”并且通道 n 为可运行状态。	

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00、03、10

11.7 简易 I²C (IIC00、IIC11、IIC20) 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 SOEmn 位 (SOEm 寄存器) 写“0”来停止串行通信数据的输出，就不输出 ACK。详细内容请参照“11.7.3(2) 处理流程”。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

SAU0 的通道 0~3 以及 SAU1 的通道 0 和通道 1 是支持简易 I²C (IIC00、IIC11、IIC20) 的通道。

单元	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

简易 I²C (IIC00、IIC11、IIC20) 有以下 4 种通信运行：

- 地址段发送 (参照 11.7.1)
- 数据发送 (参照 11.7.2)
- 数据接收 (参照 11.7.3)
- 停止条件的产生 (参照 11.7.4)

11.7.1 地址段发送

地址段发送是为了特别指定传送对象（从属设备）而在 I²C 通信时最先进行的发送运行。在产生开始条件后，将地址（7 位）和传送方向（1 位）作为 1 帧进行发送。

简易 I ² C	IIC00	IIC11	IIC20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL11、SDA11 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC11	INTIIC20
	只限于传送结束中断（不能选择缓冲器空中断）。		
错误检测标志	ACK 错误检测标志（PEFmn）		
传送数据长度	8 位（将高 7 位作为地址并且将低 1 位作为 R/W 控制进行发送）		
传送速率注 2	Max. $f_{MCK}/4$ [Hz]（ $SDRmn[15:9] \geq 1$ ） 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.1MHz（增强型快速模式） • Max. 400kHz（快速模式） • Max. 100kHz（标准模式） 		
数据电平	正相输出（默认值：高电平）。		
奇偶校验位	无奇偶校验位。		
停止位	附加 1 位（用于 ACK 接收）。		
数据方向	MSB 优先		

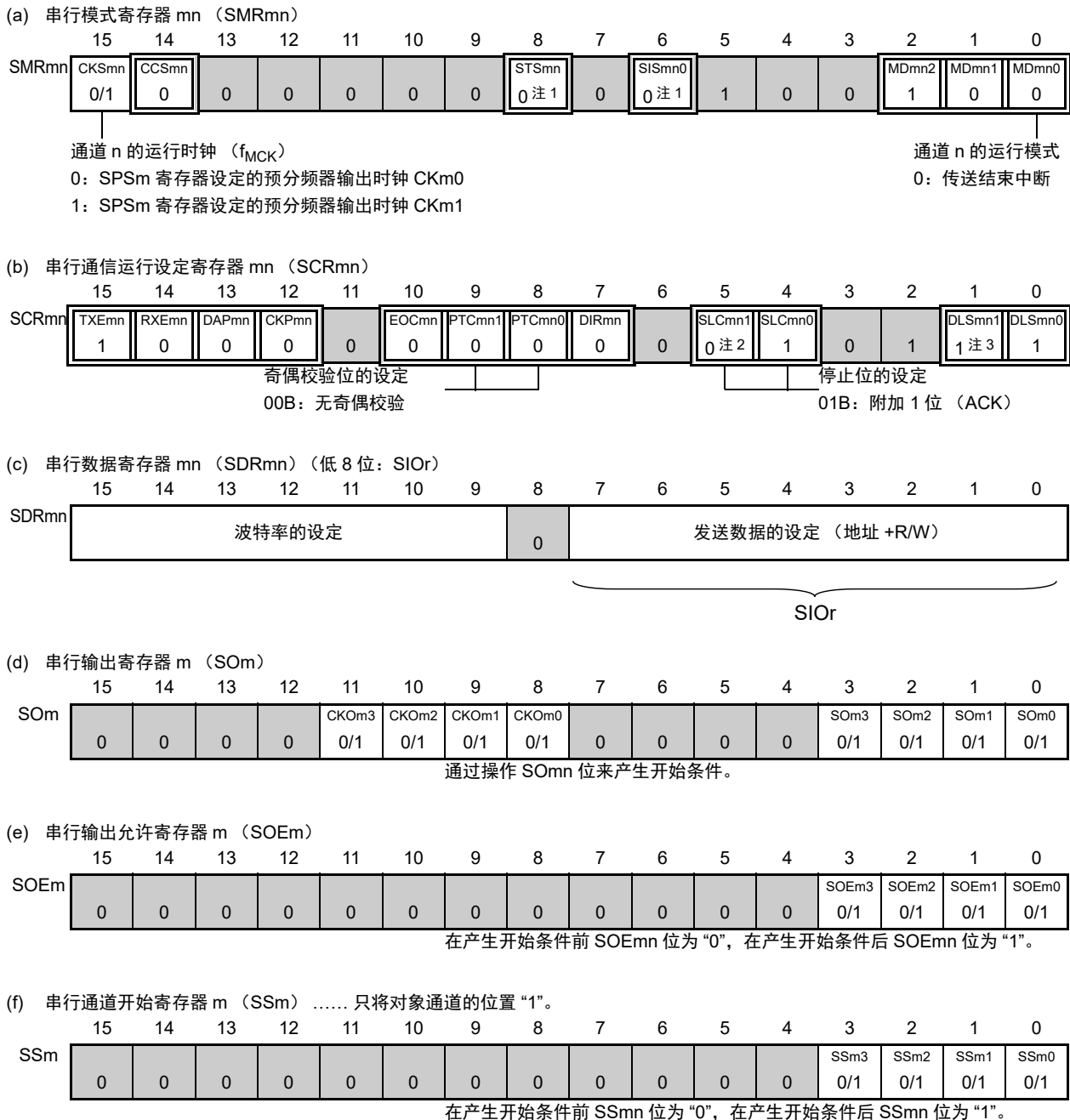
注 1. 要通过简易 I²C 进行通信时，必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出（V_{DD} 耐压）模式（POMxx=1）。详细内容请参照“4.3 控制端口功能的寄存器”和“4.5 使用复用功能时的寄存器设定”。

要将 IIC00、IIC20 与不同电位的外部设备进行通信时，还必须同样地将时钟输入 / 输出引脚（SCL00、SCL20）设定为 N 沟道漏极开路输出（V_{DD} 耐压）模式（POMxx=1）。详细内容请参照“4.4.4 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应”。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性（T_A=−40 ~ +85°C）”）的范围内使用。

备注 m：单元号（m=0、1） n：通道号（n=0 ~ 3） mn=00、03、10

(1) 寄存器的设定

图 11-97 简易 I²C (IIC00、IIC11、IIC20) 地址段发送时的寄存器设定内容例子

- 注 1. 只限于 SMR00、SMR03、SMR11 寄存器。
 2. 只限于 SCR00、SCR02、SCR10 寄存器。
 3. 只限于 SCR00 寄存器和 SCR01 寄存器，其他固定为“1”。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、11、20) mn=00、03、10

2.

--

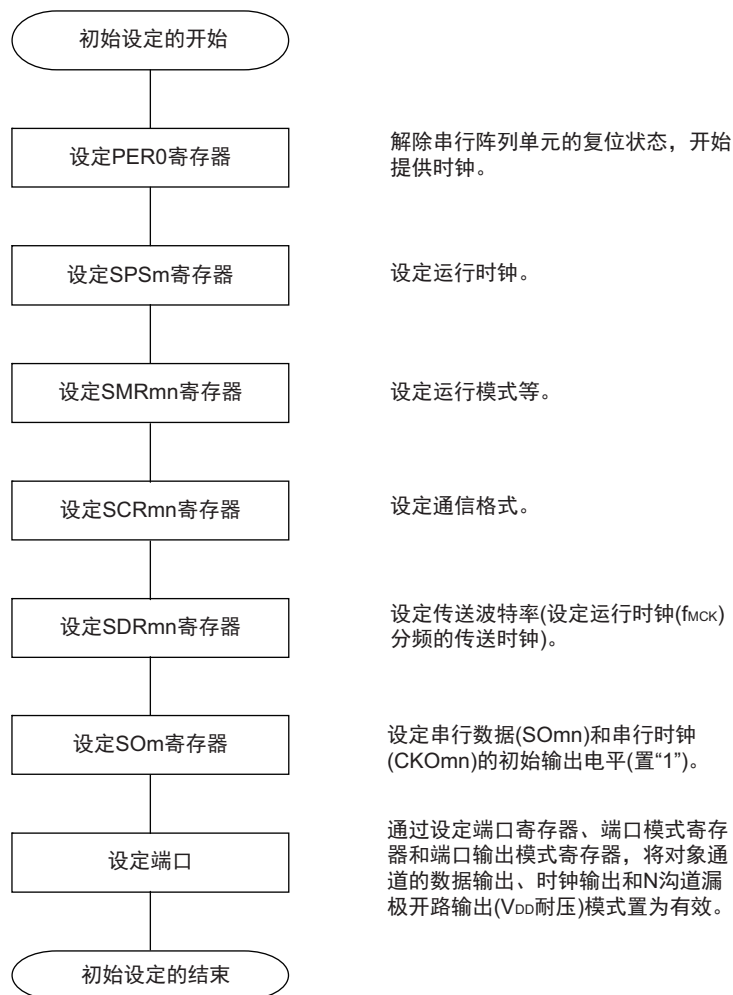
: 在 IIC 模式中为固定设定。

--

: 不能设定 (设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

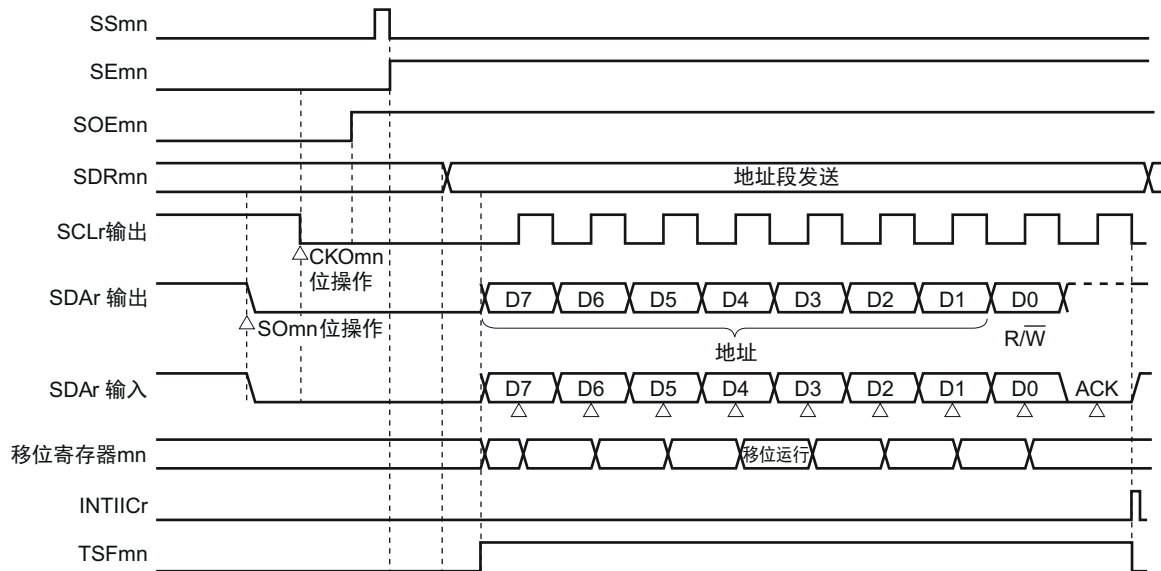
(2) 操作步骤

图 11-98 地址段发送的初始设定步骤



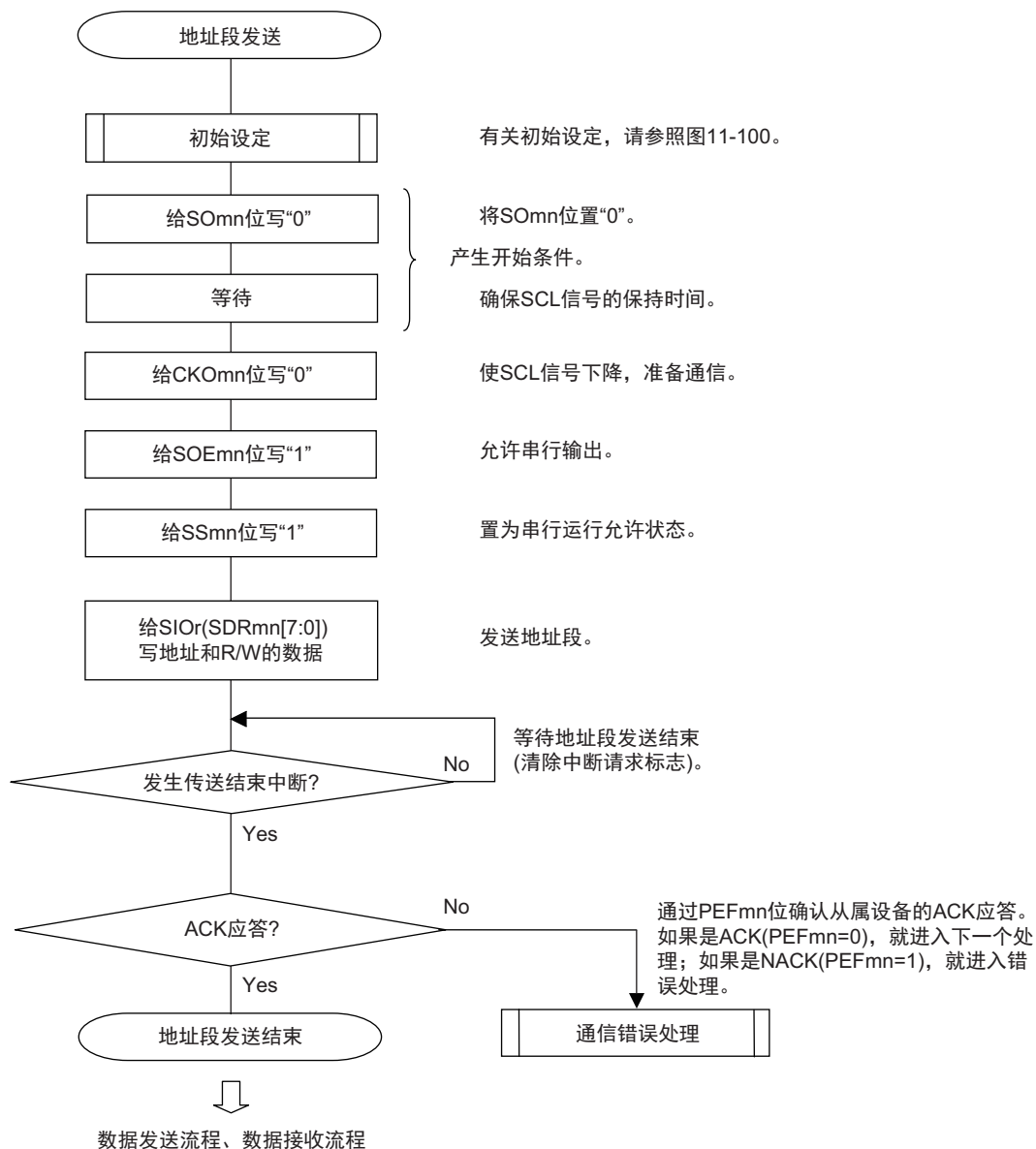
(3) 处理流程

图 11-99 地址段发送的时序图



备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC号 (r=00、11、20) mn=00、03、10

图 11-100 地址段发送的流程图



11.7.2 数据发送

数据发送是在发送地址段后给该传送对象（从属设备）发送数据的运行。在给对象从属设备发送全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00	IIC11	IIC20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL11、SDA11 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC11	INTIIC20
	只限于传送结束中断（不能选择缓冲器空中断）。		
错误检测标志	ACK 错误标志（PEFmn）		
传送数据长度	8 位		
传送速率注 2	Max. $f_{MCK}/4$ [Hz] ($SDRmn[15:9] \geq 1$) 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.1MHz（增强型快速模式） • Max. 400kHz（快速模式） • Max. 100kHz（标准模式） 		
数据电平	正相输出（默认值：高电平）。		
奇偶校验位	无奇偶校验位。		
停止位	附加 1 位（用于 ACK 接收）。		
数据方向	MSB 优先		

注 1. 要通过简易 I²C 进行通信时, 必须通过端口输出模式寄存器（POMxx）设定 N 沟道漏极开路输出（V_{DD} 耐压）模式（POMxx=1）。详细内容请参照“4.3 控制端口功能的寄存器”和“4.5 使用复用功能时的寄存器设定”。

要将 IIC00、IIC20 与不同电位的外部设备进行通信时, 还必须同样地将时钟输入 / 输出引脚（SCL00、SCL20）设定为 N 沟道漏极开路输出（V_{DD} 耐压）模式（POMxx=1）。详细内容请参照“4.4.4 通过输入 / 输出缓冲器进行的不同电位（1.8V、2.5V、3V）的对应”。

2. 必须在满足此条件并且满足电特性的外围功能特性（参照“第 27 章 电特性（T_A=−40 ~ +85°C）”）的范围内使用。

备注 m: 单元号（m=0、1） n: 通道号（n=0 ~ 3） mn=00、03、10

(1) 寄存器的设定

图 11-101 简易 I²C (IIC00、IIC11、IIC20) 数据发送时的寄存器设定内容例子

(a) 串行模式寄存器 mn (SMRmn) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0注1	0	0注1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器 mn (SCRmn) 在数据发送和接收的过程中不操作 TXEmn 位和 RXEmn 位以外的位)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0注2	1	0	1	1注3	1

(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIO_r) 在数据发送和接收的过程中只有低 8 位有效 (SIO_r)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDRmn	波特率的设定注4								0	发送数据的设定							
	SIO _r																

(d) 串行输出寄存器 m (SOM) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5

(e) 串行输出允许寄存器 m (SOEm) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

(f) 串行通道开始寄存器 m (SSm) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

- 注 1. 只限于 SMR01 寄存器和 SMR03 寄存器。
 2. 只限于 SCR00 寄存器和 SCR02 寄存器。
 3. 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。
 4. 因为已经在发送地址段时设定, 所以不需要设定。
 5. 在通信运行过程中, 值因通信数据而变。

备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、11、20) mn=00、03、10

2. □: 在 IIC 模式中为固定设定。■: 不能设定 (设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

(2) 处理流程

图 11-102 数据发送的时序图

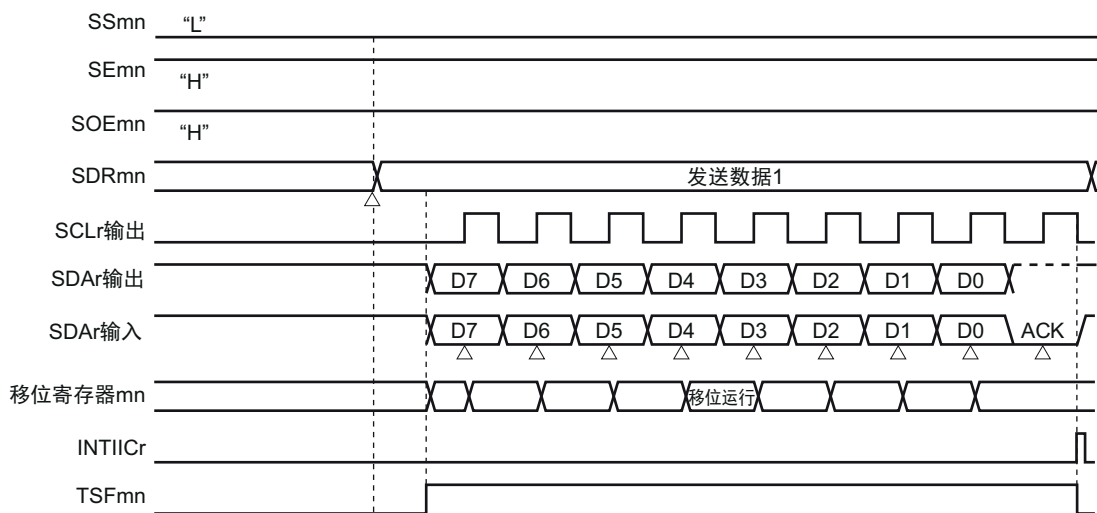
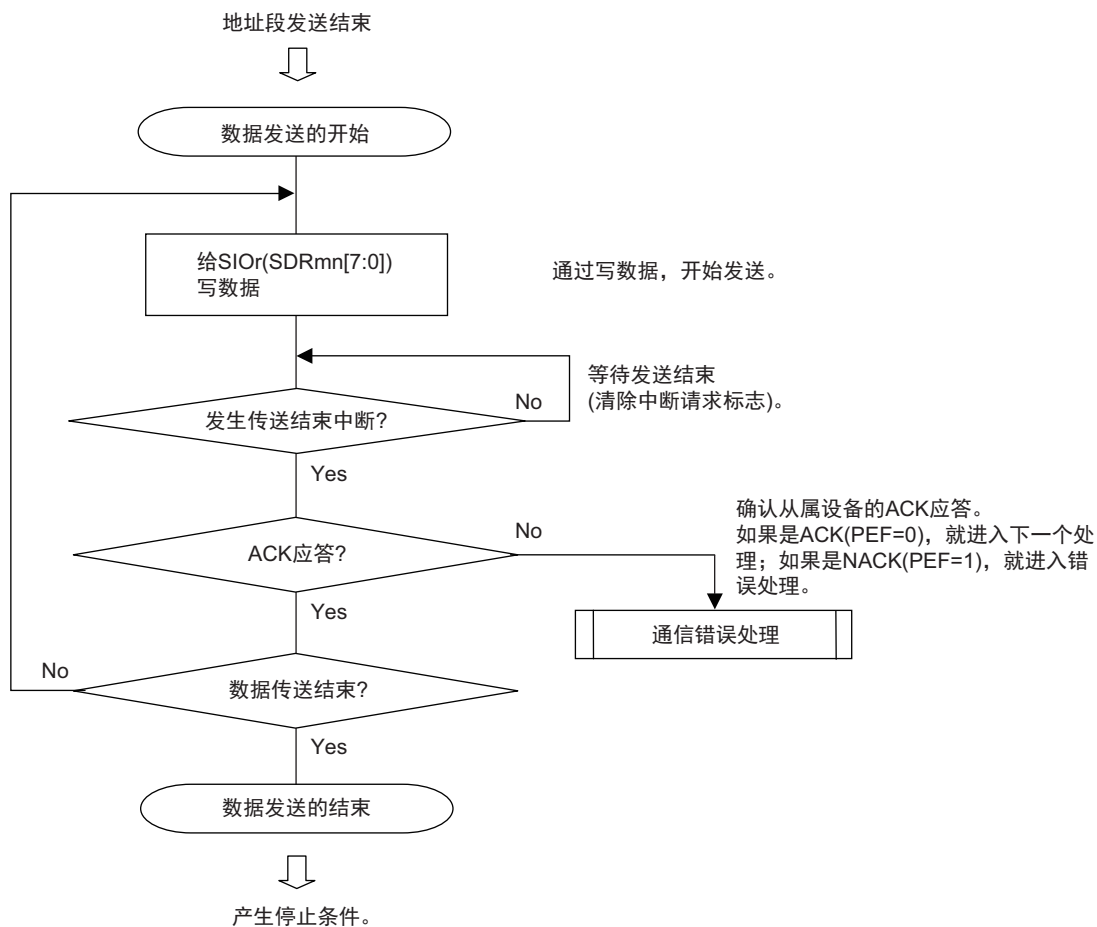


图 11-103 数据发送的流程图



11.7.3 数据接收

数据接收是在发送地址段后从传送对象（从属设备）接收数据的运行。在从对象从属设备接收全部数据后产生停止条件并且释放总线。

简易 I ² C	IIC00	IIC11	IIC20
对象通道	SAU0 的通道 0	SAU0 的通道 3	SAU1 的通道 0
使用的引脚	SCL00、SDA00 注 1	SCL11、SDA11 注 1	SCL20、SDA20 注 1
中断	INTIIC00	INTIIC11	INTIIC20
	只限于传送结束中断（不能选择缓冲器空中断）。		
错误检测标志	只有溢出错误检测标志（OVFmn）。		
传送数据长度	8 位		
传送速率注 2	Max. $f_{MCK}/4$ [Hz] ($SDRmn[15:9] \geq 1$) 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.1MHz (增强型快速模式) • Max. 400kHz (快速模式) • Max. 100kHz (标准模式) 		
数据电平	正相输出 (默认值: 高电平)。		
奇偶校验位	无奇偶校验位。		
停止位	附加 1 位 (ACK 发送)。		
数据方向	MSB 优先		

注 1. 要通过简易 I²C 进行通信时, 必须通过端口输出模式寄存器 (POMxx) 设定 N 沟道漏极开路输出 (V_{DD} 耐压) 模式 (POMxx=1)。详细内容请参照“4.3 控制端口功能的寄存器”和“4.5 使用复用功能时的寄存器设定”。

要将 IIC00、IIC20 与不同电位的外部设备进行通信时, 还必须同样地将时钟输入 / 输出引脚 (SCL00、SCL20) 设定为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式 (POMxx=1)。详细内容请参照“4.4.4 通过输入 / 输出缓冲器进行的不同电位 (1.8V、2.5V、3V) 的对应”。

2. 必须在满足此条件并且满足电特性的外围功能特性 (参照“第 27 章 电特性 ($T_A = -40 \sim +85^\circ\text{C}$)”) 的范围内使用。

备注 m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3) mn=00、03、10

(1) 寄存器的设定

图 11-104 简易 I²C (IIC00、IIC11、IIC20) 数据接收时的寄存器设定内容例子

(a) 串行模式寄存器 mn (SMRmn) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0注1	0	0注1	1	0	0	1	0	0

(b) 串行通信运行设定寄存器 mn (SCRmn) 在数据发送和接收的过程中不操作 TXEmn 位和 RXEmn 位以外的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0注2	1	0	1	1注3	1

(c) 串行数据寄存器 mn (SDRmn) (低 8 位: SIO_r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDRmn	波特率的设定注4								0	虚拟发送数据的设定 (FFH)							
	SIO _r																

(d) 串行输出寄存器 m (SOM_m) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM _m					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5	0	0	0	0	0/1注5	0/1注5	0/1注5	0/1注5

(e) 串行输出允许寄存器 m (SOEm_m) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm _m													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(f) 串行通道开始寄存器 m (SSm_m) 在数据发送和接收的过程中不操作此寄存器。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm _m													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

- 注 1. 只限于 SMR01 寄存器和 SMR03 寄存器。
 2. 只限于 SCR00 寄存器和 SCR02 寄存器。
 3. 只限于 SCR00 寄存器和 SCR01 寄存器, 其他固定为“1”。
 4. 因为已经在发送地址段时设定, 所以不需要设定。
 5. 在通信运行过程中, 值因通信数据而变。

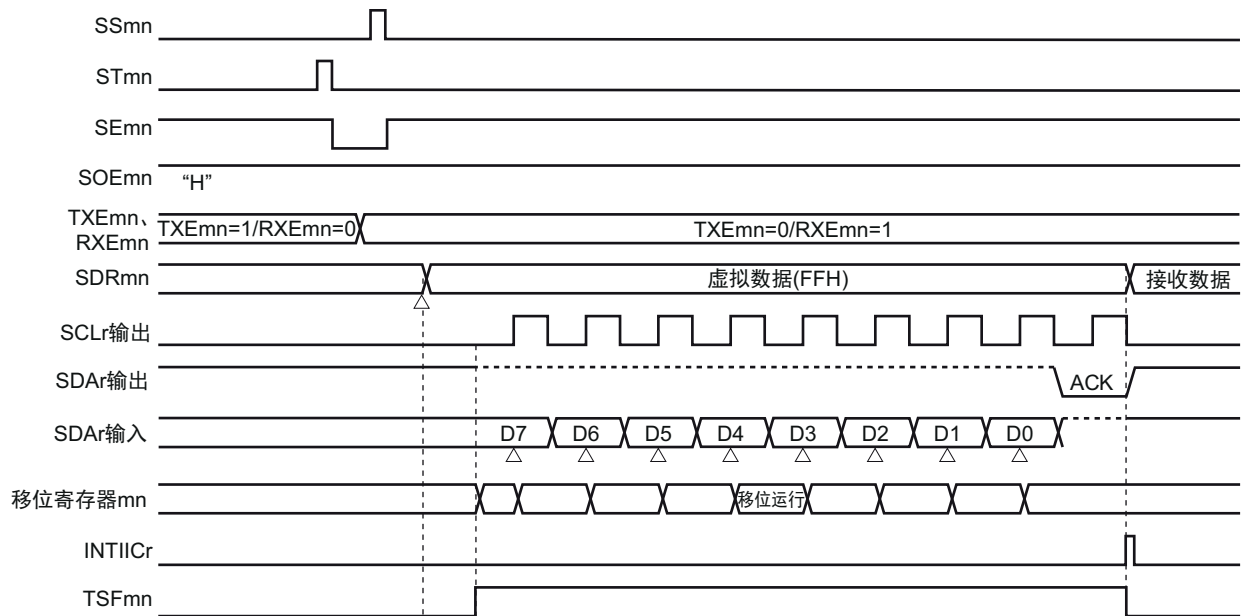
备注 1. m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、11、20) mn=00、03、10

2. □: 在 IIC 模式中为固定设定。■: 不能设定 (设定初始值)。
 0/1: 根据用户的用途置“0”或者“1”。

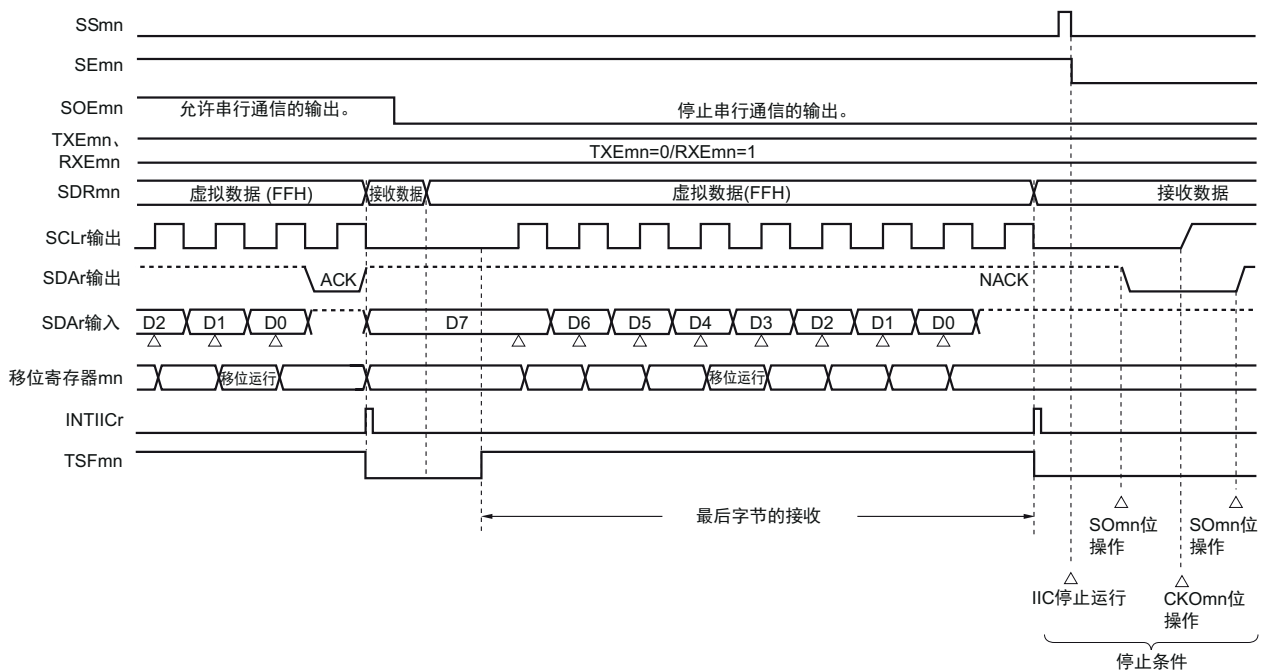
(2) 处理流程

图 11-105 数据接收的时序图

(a) 开始接收数据的情况

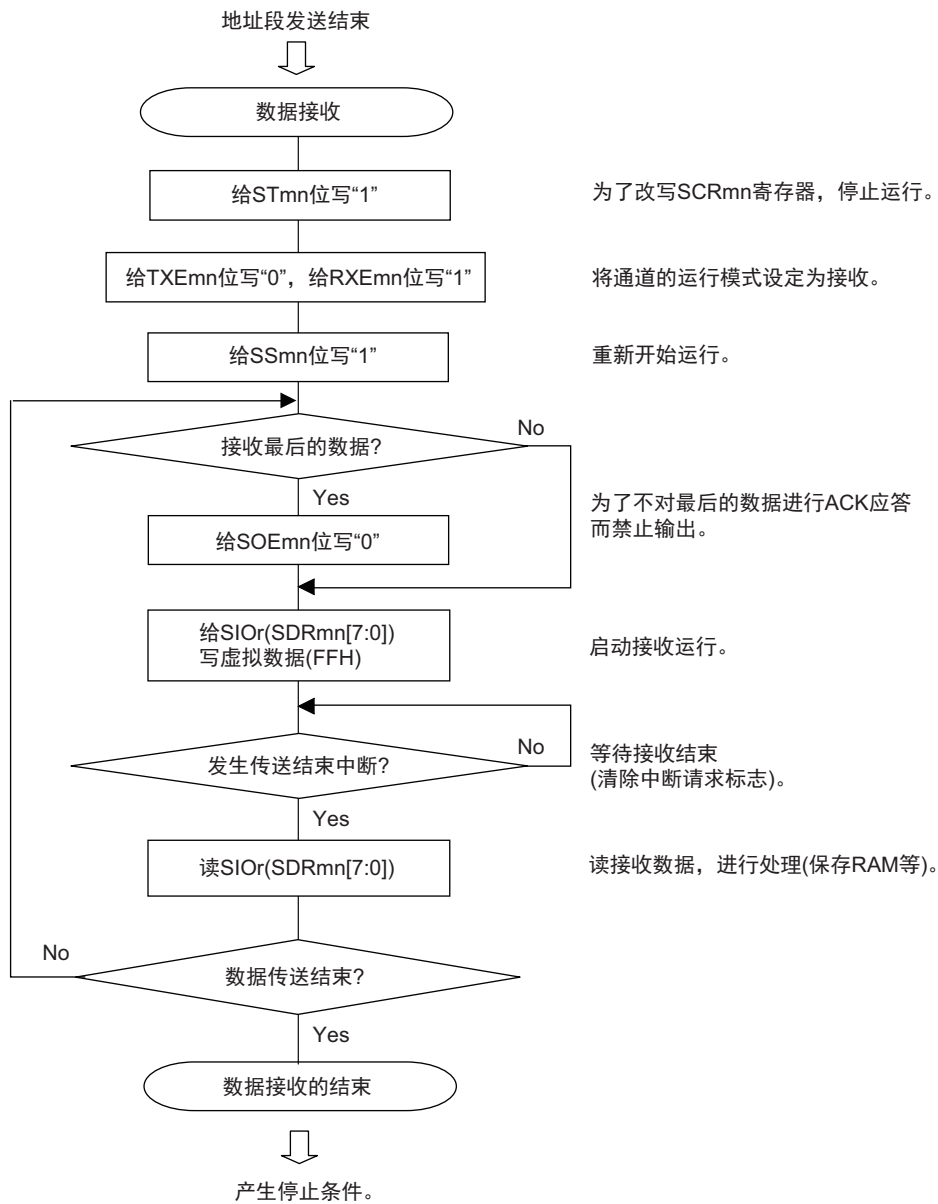


(b) 接收最后数据的情况



备注 m: 单元号 (m=0, 1) n: 通道号 (n=0~3) r: IIC号 (r=00, 11, 20) mn=00, 03, 10

图 11-106 数据接收的流程图



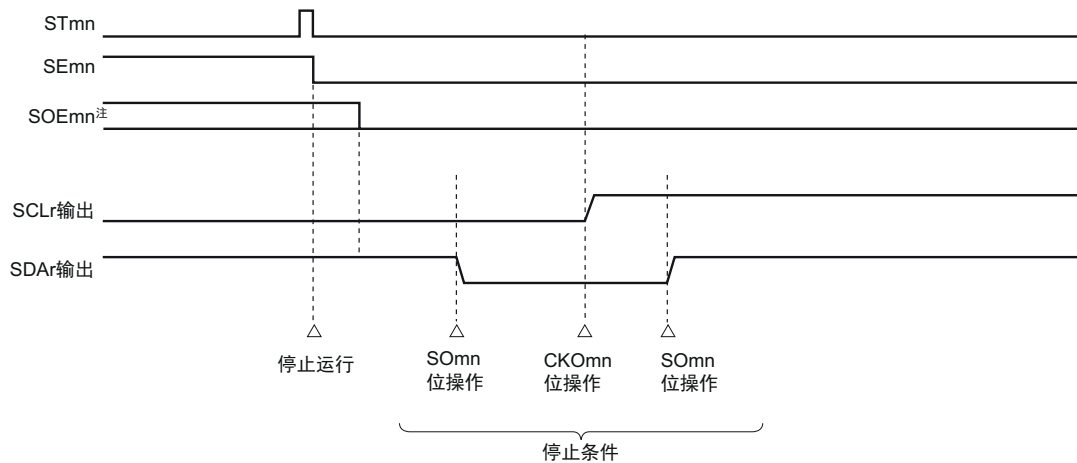
注意 在接收最后的数据时不输出 ACK (NACK)。此后，先通过将串行通道停止寄存器 m (STm) 的 STmn 位置“1”来停止运行，然后产生停止条件来结束通信。

11.7.4 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。

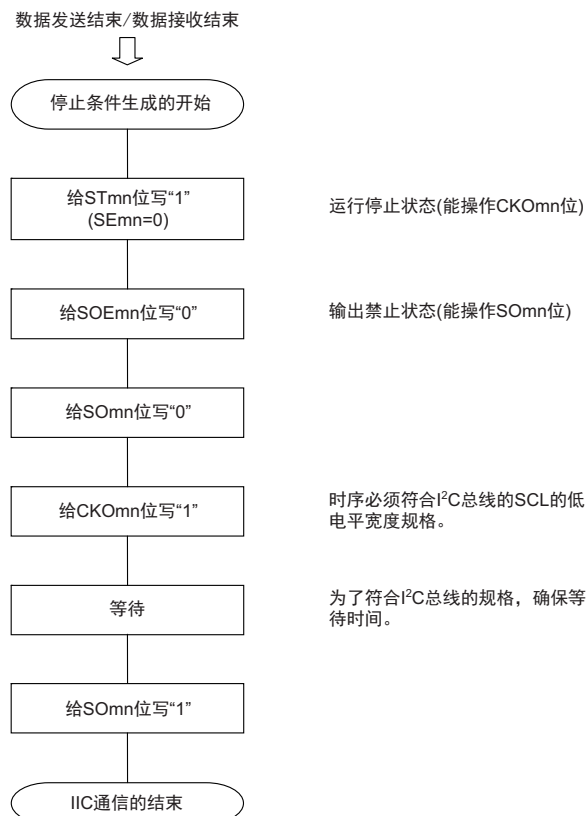
(1) 处理流程

图 11-107 产生停止条件的时序图



注 在接收时，已经在接收最后的数据前将串行输出允许寄存器 m（SOEm）的 SOEmn 位置“0”。

图 11-108 产生停止条件的流程图



11.7.5 传送速率的计算

简易 I²C (IIC00、IIC11、IIC20) 通信的传送速率能用以下计算式进行计算。

$$\text{(传送速率)} = \{ \text{对象通道的运行时钟 (f}_{\text{MCK}}) \text{ 频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2$$

注意 禁止将 SDRmn[15:9] 设定为“0000000B”，SDRmn[15:9] 的设定值必须大于等于“0000001B”。简易 I²C 输出的 SCL 信号的占空比为 50%。在 I²C 总线规格中，SCL 信号的低电平宽度大于高电平宽度。因此，如果设定为快速模式的 400kbps 或者增强型快速模式的 1Mbps，SCL 信号输出的低电平宽度就小于 I²C 总线的规格值。必须给 SDRmn[15:9] 设定能满足 I²C 总线规格的值。

备注 1. 因为 SDRmn[15:9] 的值为串行数据寄存器 (SDRmn) 的 bit15~9 的值 (0000001B~1111111B)，所以为 1~127。

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

运行时钟 (f_{MCK}) 取决于串行时钟选择寄存器 m (SPSm) 和串行模式寄存器 mn (SMRmn) 的 bit15 (CKSmn 位)。

表 11-5 简易 I²C 运行时钟的选择

SMRmn 寄存器	SPSm 寄存器								运行时钟 (f _{MCK}) 注	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f _{CLK} =20MHz 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 m（STm）=000FH）后进行更改。

备注 1. X: 忽略

2. m: 单元号 (m=0、1) n: 通道号 (n=0~3) mn=00、03、10

$f_{MCK}=f_{CLK}=20\text{MHz}$ 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})	SDRmn[15:9]	计算的传送速率	与期待的传送速率的误差
100kHz	$f_{CLK}/2$	49	100kHz	0.0%
400kHz	f_{CLK}	25	380kHz	3.8% 注

注 因为 SCL 信号的占空比为 50%，所以不能将误差设定为“0%”左右。

11.7.6 在简易 I²C (IIC00、IIC11、IIC20) 通信过程中发生错误时的处理步骤

在简易 I²C (IIC00、IIC11、IIC20) 通信过程中发生错误时的处理步骤如图 11-109 和图 11-110 所示。

图 11-109 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读串行数据寄存器 mn (SDRmn)。	→ SSRmn 寄存器的 BFFmn 位为“0”并且通道 n 为可接收状态。	这是为了防止在错误处理的过程中结束下一次接收而发生溢出错误。
读串行状态寄存器 mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
给串行标志清除触发寄存器 mn (SIRmn) 写“1”。	→ 清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。

图 11-110 在简易 I²C 模式中发生 ACK 错误时的处理步骤

软件操作	硬件状态	备注
读串行状态寄存器 mn (SSRmn)。		判断错误种类，读取值用于清除错误标志。
写串行标志清除触发寄存器 mn (SIRmn)。	→ 清除错误标志。	通过将 SSRmn 寄存器的读取值直接写到 SIRmn 寄存器，只能清除读操作时的错误。
将串行通道停止寄存器 m (STm) 的 STmn 位置“1”。	→ 串行通道允许状态寄存器 m (SEm) 的 SEmn 位为“0”并且通道 n 为运行停止状态。	因为没有返回 ACK，所以从属设备没有做好接收的准备。因此，生成停止条件并且释放总线，再次从开始条件开始通信，或者也能生成重新开始条件并且从地址发送开始重新进行。
生成停止条件。		
生成开始条件。		
将串行通道开始寄存器 m (SSm) 的 SSmn 位置“1”。	→ 串行通道允许状态寄存器 m (SEm) 的 SEmn 位为“1”并且通道 n 为可运行状态。	

备注 m: 单元号 (m=0、1) n: 通道号 (n=0~3) r: IIC 号 (r=00、11、20) mn=00、03、10

第 12 章 串行接口 IICA

串行接口 IICA 的通道个数因产品而不同。

	本产品
通道	1ch

12.1 串行接口 IICA 的功能

串行接口 IICA 有以下 3 种模式。

(1) 运行停止模式

这是用于不进行串行传送时的模式，能降低功耗。

(2) I²C 总线模式（支持多主控）

此模式通过串行时钟（SCLAn）和串行数据总线（SDAAn）的 2 条线，与多个设备进行 8 位数据传送。

符合 I²C 总线格式，主控设备能在串行数据总线上给从属设备生成“开始条件”、“地址”、“传送方向的指示”、“数据”和“停止条件”。从属设备通过硬件自动检测接收到的状态和数据。能通过此功能简化应用程序的 I²C 总线控制部分。

因为串行接口 IICA 的 SCLAn 引脚和 SDAAn 引脚用作漏极开路输出，所以串行时钟线和串行数据总线需要上拉电阻。

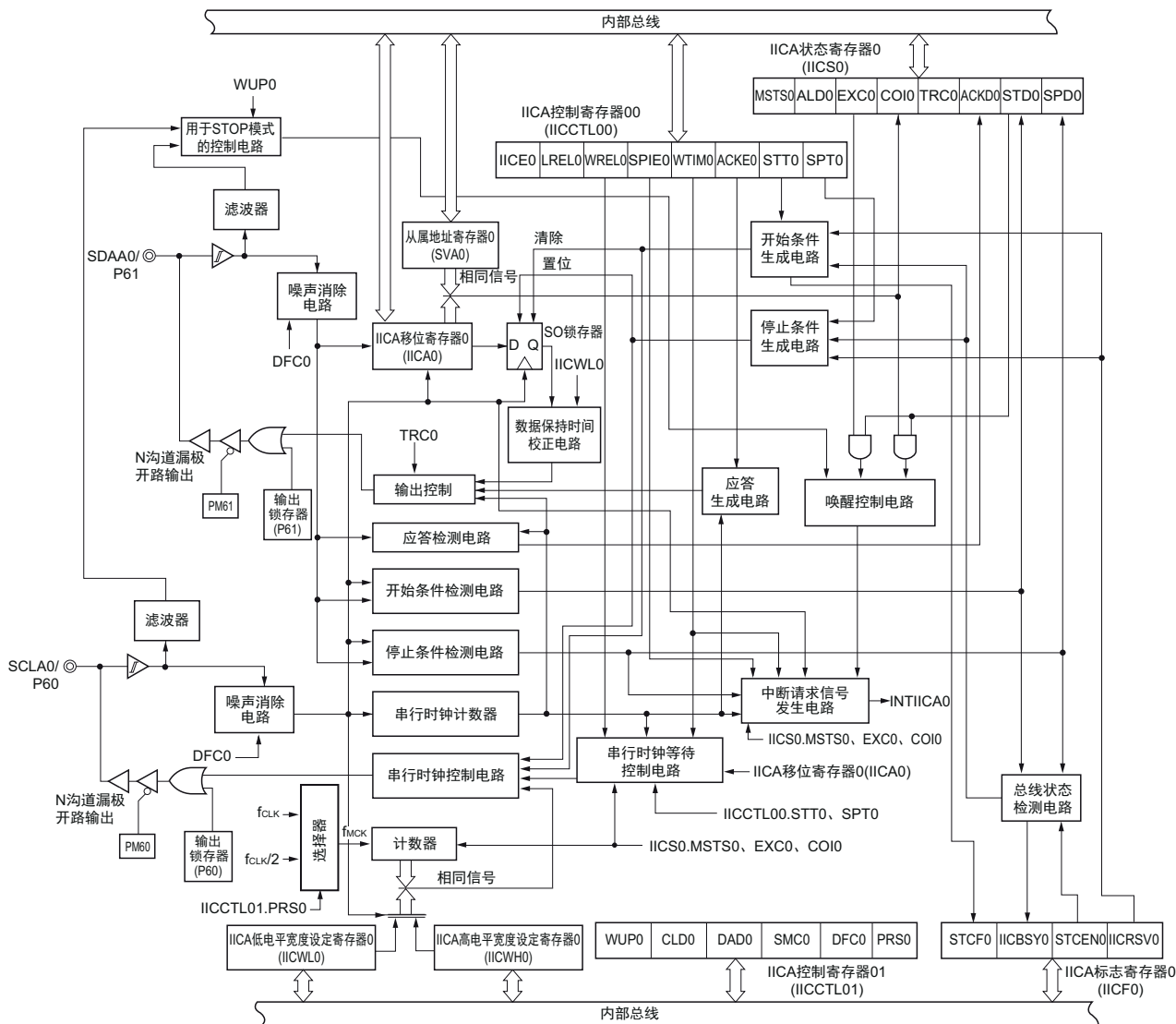
(3) 唤醒模式

在 STOP 模式中，当接收到来自主控设备的扩展码或者本地站地址时，能通过产生中断请求信号（INTIICAn）解除 STOP 模式。通过 IICA 控制寄存器 n1（IICCTLn1）的 WUPn 位进行设定。

串行接口 IICA 的框图如图 12-1 所示。

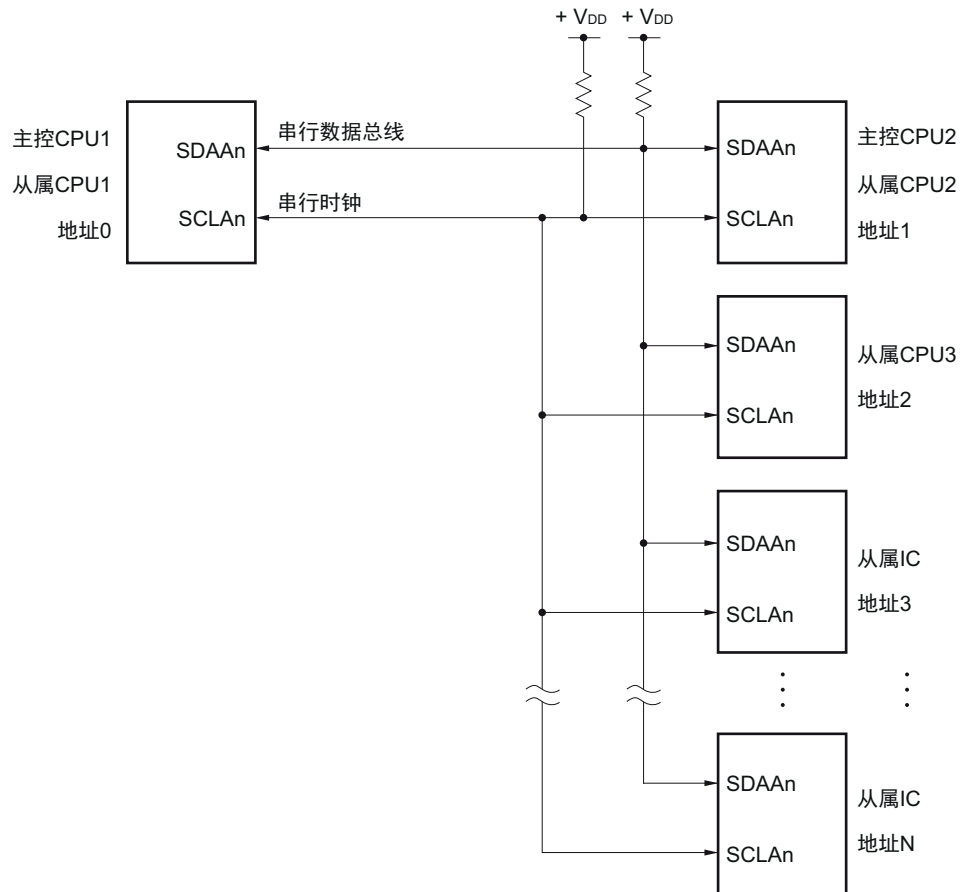
备注 n=0

图 12-1 串行接口 IICA 的框图



串行总线的结构例子如图 12-2 所示。

图 12-2 I²C 总线的串行总线结构例子



备注 n=0

12.2 串行接口 IICA 的结构

串行接口 IICA 由以下硬件构成。

表 12-1 串行接口 IICA 的结构

项目	结构
寄存器	IICA 移位寄存器 n (IICAn) 从属地址寄存器 n (SVAn)
控制寄存器	外围允许寄存器 0 (PER0) IICA 控制寄存器 n0 (IICCTLn0) IICA 状态寄存器 n (IICSn) IICA 标志寄存器 n (IICFn) IICA 控制寄存器 n1 (IICCTLn1) IICA 低电平宽度设定寄存器 n (IICWLn) IICA 高电平宽度设定寄存器 n (IICWHn) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

(1) IICA 移位寄存器 n (IICAn)

IICAn 寄存器是与串行时钟同步进行 8 位串行数据和 8 位并行数据相互转换的寄存器，用于发送和接收。

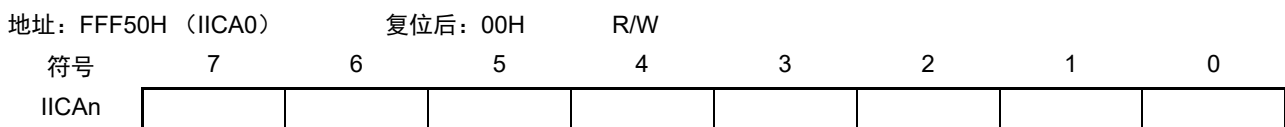
能通过读写 IICAn 寄存器来控制实际的发送和接收。

在等待期间，通过写 IICAn 寄存器来解除等待，开始传送数据。

通过 8 位存储器操作指令设定 IICAn 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-3 IICAn 移位寄存器 n (IICAn) 的格式



注意 1. 在数据传送过程中，不能给 IICAn 寄存器写数据。

2. 只能在等待期间读写 IICAn 寄存器。除了等待期间以外，禁止在通信状态下存取 IICAn 寄存器。但是，在主控设备的情况下，能在将通信触发位 (STTn) 置“1”后写一次 IICAn 寄存器。

3. 当预约通信时，必须在检测到由停止条件产生的中断后给 IICAn 寄存器写数据。

备注 n=0

(2) 从属地址寄存器 n (SVAn)

这是在用作从属设备时保存 7 位本地站地址 {A6, A5, A4, A3, A2, A1, A0} 的寄存器。

通过 8 位存储器操作指令设定 SVAn 寄存器。但是，在 STDn 位为“1”（检测到开始条件）时，禁止改写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 12-4 从属地址寄存器 n (SVAn) 的格式

地址: F0234H (SVA0)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0注

注 bit0 固定为“0”。

(3) SO 锁存器

SO 锁存器保持 SDAAn 引脚的输出电平。

(4) 唤醒控制电路

当设定在从属地址寄存器 n (SVAn) 的地址值和接收到的地址相同时或者当接收到扩展码时，此电路产生中断请求 (INTIICAn)。

(5) 串行时钟计数器

在发送或者接收过程中，此计数器对输出或者输入的串行时钟进行计数，检查是否进行了 8 位数据的发送和接收。

(6) 中断请求信号发生电路

此电路控制产生中断请求信号 (INTIICAn)。

由以下 2 种触发产生 I²C 中断请求。

- 第 8 或者第 9 个串行时钟的下降（通过 WTIMn 位进行设定）
- 因检测到停止条件而产生中断请求（通过 SPIEn 位进行设定）。

备注 WTIMn 位: IICA 控制寄存器 n0 (IICCTLn0) 的 bit3
SPIEn 位: IICA 控制寄存器 n0 (IICCTLn0) 的 bit4

(7) 串行时钟控制电路

在主控模式中，此电路从采样时钟生成输出到 SCLAn 引脚的时钟。

(8) 串行时钟等待控制电路

此电路控制等待时序。

备注 n=0

(9) 应答生成电路、停止条件检测电路、开始条件检测电路、应答检测电路

这些电路生成并且检测各种状态。

(10) 数据保持时间校正电路

此电路生成对串行时钟下降的数据保持时间。

(11) 开始条件生成电路

如果将 STTn 位置“1”，此电路就生成开始条件。

但是，在禁止预约通信的状态下（IICRSVn 位 =1）并且没有释放总线（IICBSYn 位 =1）时，忽视开始条件请求并且将 STCFn 位置“1”。

(12) 停止条件生成电路

如果将 SPTn 位置“1”，此电路就生成停止条件。

(13) 总线状态检测电路

此电路通过检测开始条件和停止条件来检测总线是否被释放。

但是，在刚运行时不能立即检测总线状态，因此必须通过 STCENn 位设定总线状态检测电路的初始状态。

备注 1. STTn 位： IICA 控制寄存器 n0（IICCTLn0）的 bit1

SPTn 位： IICA 控制寄存器 n0（IICCTLn0）的 bit0

IICRSVn 位： IICA 标志寄存器 n（IICFn）的 bit0

IICBSYn 位： IICA 标志寄存器 n（IICFn）的 bit6

STCFn 位： IICA 标志寄存器 n（IICFn）的 bit7

STCENn 位： IICA 标志寄存器 n（IICFn）的 bit1

2. n=0

12.3 控制串行接口 IICA 的寄存器

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 n0（IICCTLn0）
- IICA 标志寄存器 n（IICFn）
- IICA 状态寄存器 n（IICSn）
- IICA 控制寄存器 n1（IICCTLn1）
- IICA 低电平宽度设定寄存器 n（IICWLn）
- IICA 高电平宽度设定寄存器 n（IICWHn）
- 端口模式寄存器 6（PM6）
- 端口寄存器 6（P6）

备注 n=0

12.3.1 外围允许寄存器 0 (PER0)

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

要使用串行接口 IICAn 时，必须将 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 12-5 外围允许寄存器 0 (PER0) 的格式

地址: F00F0H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PER0	TMKAEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

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

注意 1. 要设定串行接口 IICA 时，必须先在 IICAnEN 位为“1”的状态下设定以下的寄存器。当 IICAnEN 位为“0”时，串行接口 IICA 的控制寄存器的值为初始值，忽视写操作（端口模式寄存器 6 (PM6) 和端口寄存器 6 (P6) 除外）。

- IICA 控制寄存器 n0 (IICCTLn0)
- IICA 标志寄存器 n (IICFn)
- IICA 状态寄存器 n (IICSn)
- IICA 控制寄存器 n1 (IICCTLn1)
- IICA 低电平宽度设定寄存器 n (IICWLn)
- IICA 高电平宽度设定寄存器 n (IICWHn)

2. 必须将 bit6 和 bit1 置“0”。

备注 n=0

12.3.2 IICA 控制寄存器 n0 (IICCTLn0)

这是允许或者停止 I²C 运行、设定等待时序以及设定其他 I²C 运行的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICCTLn0 寄存器。但是，必须在 IICEn 位为“0”时或者在等待期间设定 SPIEn 位、WTIMn 位和 ACKEn 位，而且在将 IICEn 位从“0”置为“1”时能同时设定这些位。

在产生复位信号后，此寄存器的值变为“00H”。

备注 n=0

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (1/4)

地址: F0230H (IICCTL00) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I ² C 运行的允许
0	停止运行。对 IICA 状态寄存器 n (IICSn) 进行复位 ^{注 1} , 并且停止内部运行。
1	允许运行。
必须在 SCLAn 线和 SDAAn 线为高电平的状态下将此位置“1”。	
清除条件 (IICEn=0)	置位条件 (IICEn=1)
<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。

LRELn ^{注 2、3}	通信的退出
0	通常运行
1	退出当前的通信, 进入待机状态。执行后自动清“0”。 在接收到与本站无关的扩展码等情况下使用。 SCLAn 线和 SDAAn 线变为高阻状态。 IICA 控制寄存器 n0 (IICCTLn0) 和 IICA 状态寄存器 n (IICSn) 中的以下标志被清“0”: •STTn •SPTn •MSTSn •EXCn •COIn •TRCn •ACKDn •STDn
变为退出通信的待机状态, 保持到满足以下的通信参加条件为止。	
<ul style="list-style-type: none"> 在检测到停止条件后作为主控设备启动。 在检测到开始条件后地址匹配或者接收到扩展码。 	
清除条件 (LRELn=0)	置位条件 (LRELn=1)
<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。

WRELn ^{注 2、3}	等待的解除
0	不解除等待。
1	解除等待。在解除等待后自动清除。
如果在发送状态下 (TRCn=1) 的第 9 个时钟等待期间将 WRELn 位 (解除等待) 置位, SDAAn 线就变为高阻抗状态 (TRCn=0)。	
清除条件 (WRELn=0)	置位条件 (WRELn=1)
<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 移位寄存器 n (IICAn)、IICA 标志寄存器 n (IICFn) 的 STCFn 位和 IICBSYn 位以及 IICA 控制寄存器 n1 (IICCTLn1) 的 CLDn 位和 DADn 位进行复位。
2. 在 IICEn 位为“0”的状态下, 此位的信号无效。
3. LRELn 位和 WRELn 位的读取值总是“0”。

注意 如果在 SCLAn 线为高电平、SDAAn 线为低电平并且数字滤波器为 ON (IICCTLn1 寄存器的 DFCn=1) 时允许 I²C 运行 (IICEn=1), 就立即检测开始条件。此时, 必须在允许 I²C 运行 (IICEn=1) 后连续通过位存储器操作指令将 LRELn 位置“1”。

备注 n=0

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (2/4)

SPIEn 注 1	允许或者禁止停止条件检测产生的中断请求	
0	禁止	
1	允许	
当 IICA 控制寄存器 n1 (IICCTLn1) 的 WUPn 位为“1”时, 即使将 SPIEn 位置“1”也不产生停止条件中断。		
清除条件 (SPIEn=0)		置位条件 (SPIEn=1)
<ul style="list-style-type: none"> 通过指令清除。 当复位时 		<ul style="list-style-type: none"> 通过指令置位。

WTIMn 注 1	等待和中断请求的控制	
0	在第 8 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 8 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 8 个时钟后, 将时钟置为低电平, 然后等待主控设备。	
1	在第 9 个时钟的下降沿产生中断请求信号。 主控设备: 在输出 9 个时钟后, 将时钟输出置为低电平进行等待。 从属设备: 在输入 9 个时钟后, 将时钟置为低电平, 然后等待主控设备。	
在地址传送期间, 与此位的设定无关, 在第 9 个时钟的下降沿产生中断; 在地址传送结束后, 此位的设定有效。主控设备在地址传送期间的第 9 个时钟下降沿进入等待状态。接收到本地站地址的从属设备在产生应答 (ACK) 后的第 9 个时钟下降沿进入等待状态, 但是接收到扩展码的从属设备在第 8 个时钟下降沿进入等待状态。		
清除条件 (WTIMn=0)		置位条件 (WTIMn=1)
<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。

ACKEn 注 1、2	应答控制	
0	禁止应答。	
1	允许应答。在第 9 个时钟期间将 SDAAn 线置为低电平。	
清除条件 (ACKEn=0)		置位条件 (ACKEn=1)
<ul style="list-style-type: none"> 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 IICEn 位为“0”的状态下, 此位的信号无效。必须在此期间设定此位。
 2. 在地址传送过程中不是扩展码时, 设定值无效。
 当为从属设备并且地址匹配时, 与设定值无关而生成应答。

备注 n=0

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (3/4)

STTn 注 1、2	开始条件的触发
0	不生成开始条件。
1	<p>当总线被释放时（待机状态，IICBSYn 位为“0”）： 如果将此位置“1”，就生成开始条件（作为主控设备的启动）。</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（IICRSVn=0） 用作开始条件预约标志。如果将此位置“1”，就在释放总线后自动生成开始条件。 • 禁止通信预约功能的情况（IICRSVn=1） 即使将此位置“1”，也清除 STTn 位并且将 STTn 清除标志（STCFn）置“1”，不生成开始条件。 <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 ACKEn 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 • 主控发送：在应答期间，可能无法正常生成开始条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 • 禁止与停止条件的触发（SPTn）同时置“1”。 • 在将 STTn 位置“1”后，禁止在满足清除条件前再次将此位“1”。 	
清除条件（STTn=0）	置位条件（STTn=1）
<ul style="list-style-type: none"> • 在禁止通信预约的状态下将 STTn 位置“1”。 • 在仲裁失败时 • 主控设备生成开始条件。 • 因 LRELn 位为“1”（退出通信）而进行的清除 • 当 IICEn 位为“0”（停止运行）时 • 当复位时 	<ul style="list-style-type: none"> • 通过指令置位。

- 注 1. 在 IICEn 位为“0”的状态下，此位的信号无效。
2. STTn 位的读取值总是“0”。

- 备注 1. IICRSVn: IICA 标志寄存器 n (IICFn) 的 bit0
STCFn: IICA 标志寄存器 n (IICFn) 的 bit7
2. n=0

图 12-6 IICA 控制寄存器 n0 (IICCTLn0) 的格式 (4/4)

SPTn ^注	停止条件的触发	
0	不生成停止条件。	
1	生成停止条件 (作为主控设备的传送结束)。	
有关置位时序的注意事项:		
<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 ACKEn 位置“0”并且通知从属设备接收已经完成后的等待期间才能将此位置“1”。 主控发送: 在应答期间, 可能无法正常生成停止条件。必须在输出第 9 个时钟后的等待期间将此位置“1”。 禁止与开始条件的触发 (STTn) 同时置“1”。 只有在主控设备的情况下才能将 SPTn 位置“1”。 在 WTIMn 位为“0”时, 必须注意: 如果在输出 8 个时钟后的等待期间将 SPTn 位置“1”, 就在解除等待后的第 9 个时钟的高电平期间生成停止条件。必须在输出 8 个时钟后的等待期间将 WTIMn 位从“0”置为“1”并且在输出第 9 个时钟后的等待期间将 SPTn 位置“1”。 在将 SPTn 位置“1”后, 禁止在满足清除条件前再次将此位置“1”。 		
清除条件 (SPTn=0)		置位条件 (SPTn=1)
<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 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 通过指令置位。

注 SPTn 位的读取值总是“0”。

注意 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICCTLn0 寄存器的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

备注 1. 如果在设定数据后读 bit0 (SPTn), 此位就变为“0”。

2. n=0

12.3.3 IICA 状态寄存器 n (IICSn)

这是表示 I²C 状态的寄存器。

只有在 STTn 位为“1”并且等待期间，才能通过 1 位或者 8 位存储器操作指令读 IICSn 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

注意 在 STOP 模式中允许地址匹配唤醒功能 (WUPn=1) 状态下，禁止读 IICSn 寄存器。在 WUPn 位为“1”的状态下，与 INTIICAn 中断请求无关，如果将 WUPn 位从“1”改为“0” (停止唤醒运行)，就在检测到下一个开始条件或者停止条件后才会反映状态的变化。因此，要使用唤醒功能时，必须允许 (SPIEn=1) 因检测到停止条件而产生的中断，并且在检测到中断后读 IICSn 寄存器。

备注 STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1
WUPn: IICA 控制寄存器 n1 (IICCTLn1) 的 bit7

图 12-7 IICA 状态寄存器 n (IICSn) 的格式 (1/3)

地址: FFF51H (IICS0)	复位后: 00H	R						
符号	7	6	5	4	3	2	1	0
IICSn	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n

MSTS _n	主控状态的确认标志
0	从属状态或者通信待机状态
1	主控通信状态
清除条件 (MSTS _n =0)	置位条件 (MSTS _n =1)
<ul style="list-style-type: none"> 当检测到停止条件时 当 ALD_n 位为“1” (仲裁失败) 时 因 LREL_n 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 当生成开始条件时

ALD _n	仲裁失败的检测
0	表示未发生仲裁或者赢得仲裁。
1	表示仲裁失败。清除 MSTS _n 位。
清除条件 (ALD _n =0)	置位条件 (ALD _n =1)
<ul style="list-style-type: none"> 在读 IICSn 寄存器后自动清除注。 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 当仲裁失败时

注 即使对 IICSn 寄存器以外的位执行位存储器操作指令，也清除此位。因此，在使用 ALD_n 位时，必须在读其他位前先读 ALD_n 位的数据。

备注 1. LREL_n: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6
IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

图 12-7 IICA 状态寄存器 n (IICSn) 的格式 (2/3)

EXCn	扩展码的接收检测	
0	未接收到扩展码。	
1	接收到扩展码。	
清除条件 (EXCn=0)		置位条件 (EXCn=1)
<ul style="list-style-type: none"> 当检测到开始条件时 当检测到停止条件时 因 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 		<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 个时钟的上升沿置位)

COIn	地址匹配的检测	
0	地址不同。	
1	地址相同。	
清除条件 (COIn=0)		置位条件 (COIn=1)
<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 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当接收地址和本地站地址 (从属地址寄存器 n (SVAn)) 相同时 (在第 8 个时钟的上升沿置位)

TRCn	发送 / 接收的状态检测	
0	处于接收状态 (发送状态除外)。将 SDAAn 线置为高阻抗。	
1	处于发送状态。设定为能将 SOn 锁存器的值输出到 SDAAn 线 (在第 1 字节的第 9 个时钟的下降沿以后有效)。	
清除条件 (TRCn=0)		置位条件 (TRCn=1)
<主控设备和从属设备> <ul style="list-style-type: none"> 当检测到停止条件时 因 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 因 WRELn 位为“1” (解除等待) 而进行的清除注 当 ALDn 位从“0”变为“1” (仲裁失败) 时 当复位时 不参加通信的情况 (MSTSn、EXCn、COIn=0) <主控设备> <ul style="list-style-type: none"> 当第 1 字节的 LSB (传送方向指示位) 输出“1”时 <从属设备> <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 		<主控设备> <ul style="list-style-type: none"> 当生成开始条件时 当第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输出“0” (主控发送) 时 <从属设备> <ul style="list-style-type: none"> 当主控设备的第 1 字节 (地址传送) 的 LSB (传送方向指示位) 输入“1” (从属发送) 时

注 在 IICA 状态寄存器 n (IICSn) 的 bit3 (TRCn) 为“1” (发送状态) 时, 如果在第 9 个时钟将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置“1”来解除等待, 就在清除 TRCn 位 (接收状态) 后将 SDAAn 线置为高阻抗。必须通过写 IICA 移位寄存器 n 进行 TRCn 位为“1” (发送状态) 时的等待解除。

备注 1. LRELn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6
 IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7
 2. n=0

图 12-7 IICA 状态寄存器 n (IICSn) 的格式 (3/3)

ACKDn	应答 (ACK) 的检测	
0	未检测到应答。	
1	检测到应答。	
清除条件 (ACKDn=0)		置位条件 (ACKDn=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当下一个字节的第 1 个时钟上升时 因 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 在 SCLAn 线的第 9 个时钟上升沿将 SDAAn 线置为低电平时

STDn	开始条件的检测	
0	未检测到开始条件。	
1	检测到开始条件, 表示处于地址传送期间。	
清除条件 (STDn=0)		置位条件 (STDn=1)
<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 LRELn 位为“1” (退出通信) 而进行的清除 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到开始条件时

SPDn	停止条件的检测	
0	未检测到停止条件。	
1	检测到停止条件, 主控设备结束通信并且已释放总线。	
清除条件 (SPDn=0)		置位条件 (SPDn=1)
<ul style="list-style-type: none"> 在将此位置位后, 当检测到开始条件后的地址传送字节的第 1 个时钟上升时 当 WUPn 位从“1”变为“0”时 当 IICEn 位从“1”变为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当检测到停止条件时

备注 1. LRELn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit6

IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

12.3.4 IICA 标志寄存器 n (IICFn)

这是设定 I²C 运行模式以及表示 I²C 总线状态的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICFn 寄存器。但是, 只能读 STTn 清除标志 (STCFn) 和 I²C 总线状态标志 (IICBSYn)。

通过 IICRSVn 位设定允许或者禁止通信预约功能, 并且通过 STCENn 位设定 IICBSYn 位的初始值。

只有在禁止 I²C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时才能写 IICRSVn 位和 STCENn 位。在允许运行后, 只能读 IICFn 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 12-8 IICA 标志寄存器 n (IICFn) 的格式

地址: FFF52H (IICF0) 复位后: 00H R/W 注

符号	7	6	5	4	3	2	1	0
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn 清除标志
0	发行开始条件。
1	无法发行开始条件而清除 STTn 标志。
清除条件 (STCFn=0)	置位条件 (STCFn=1)
<ul style="list-style-type: none"> 因 STTn 位为“1”而进行的清除 当 IICEn 位为“0” (停止运行) 时 当复位时 	<ul style="list-style-type: none"> 在设定为禁止通信预约 (IICRSVn=1) 的状态下无法发行开始条件而将 STTn 位清“0”时

IICBSYn	I ² C 总线状态标志
0	总线释放状态 (STCENn=1 时的通信初始状态)
1	总线通信状态 (STCENn=0 时的通信初始状态)
清除条件 (IICBSYn=0)	置位条件 (IICBSYn=1)
<ul style="list-style-type: none"> 当检测到停止条件时 当 IICEn 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 STCENn 位为“0” 时的 IICEn 位的置位

STCENn	初始开始允许触发
0	在允许运行 (IICEn=1) 后, 通过检测停止条件来允许生成开始条件。
1	在允许运行 (IICEn=1) 后, 不检测停止条件而允许生成开始条件。
清除条件 (STCENn=0)	置位条件 (STCENn=1)
<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。

IICRSVn	通信预约功能禁止位
0	允许通信预约。
1	禁止通信预约。
清除条件 (IICRSVn=0)	置位条件 (IICRSVn=1)
<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 (IICEn=0) 时才能写 STCENn 位。

2. 如果 STCENn 位为“1”, 就与实际的总线状态无关而认为总线为释放状态 (IICBSYn=0), 因此为了避免在发行第 1 个开始条件 (STTn=1) 时破坏其他通信, 需要确认没有正在通信的第三方。

3. 只有在停止运行 (IICEn=0) 时才能写 IICRSVn。

备注 1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

12.3.5 IICA 控制寄存器 n1 (IICCTLn1)

这是用于设定 I²C 运行模式以及检测 SCLAn 引脚状态和 SDAAn 引脚状态的寄存器。

通过 1 位或者 8 位存储器操作指令设定 IICCTLn1 寄存器。但是，只能读 CLDn 位和 DADn 位。

除了 WUPn 位以外，必须在禁止 I²C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICCTLn1 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

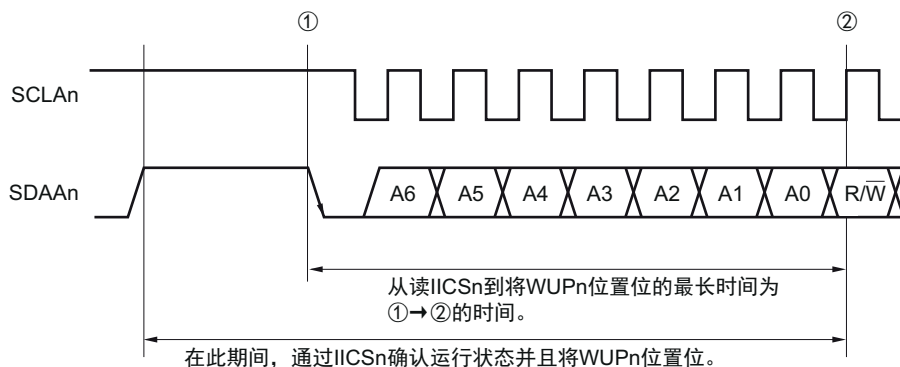
图 12-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (1/2)

地址: F0231H (IICCTL01)	复位后: 00H	R/W 注 1						
符号	7	6	5	4	3	2	1	0
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	地址匹配唤醒的控制
0	在 STOP 模式中，停止地址匹配唤醒功能的运行。
1	在 STOP 模式中，允许地址匹配唤醒功能的运行。
要通过将 WUPn 位置“1”来转移到 STOP 模式时，必须在将 WUPn 位置“1”后至少经过 3 个 f_{MCK} 时钟，然后执行 STOP 指令（参照“图 12-22 将 WUPn 位置“1”时的流程”）。 在地址匹配或者接收到扩展码后，必须将 WUPn 位清“0”。能通过将 WUPn 位清“0”来参加后续的通信（需要在将 WUPn 位清“0”后解除等待以及写发送数据）。 在 WUPn 位为“1”的状态下，地址匹配或者接收到扩展码时的中断时序与 WUPn 位为“0”时的中断时序相同（根据时钟产生采样误差的延迟差）。另外，当 WUPn 位为“1”时，即使将 SPIEn 位置“1”也不产生停止条件中断。	
清除条件 (WUPn=0)	置位条件 (WUPn=1)
<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 (MSTSn=0、EXCn=0、COIn=0 并且 STDn=0（不参加通信）) 注 2。

注 1. bit4 和 bit5 是只读位。

2. 在以下所示的期间，需要确认 IICA 状态寄存器 n (IICSn) 的状态并且将其置位。



备注 n=0

图 12-9 IICA 控制寄存器 n1 (IICCTLn1) 的格式 (2/2)

CLDn	SCLAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)	
0	检测到 SCLAn 引脚为低电平。	
1	检测到 SCLAn 引脚为高电平。	
清除条件 (CLDn=0)		置位条件 (CLDn=1)
<ul style="list-style-type: none"> 当 SCLAn 引脚为低电平时 当 IICEn 位为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当 SCLAn 引脚为高电平时

DADn	SDAAn 引脚的电平检测 (只在 IICEn 位为“1”时有效)	
0	检测到 SDAAn 引脚为低电平。	
1	检测到 SDAAn 引脚为高电平。	
清除条件 (DADn=0)		置位条件 (DADn=1)
<ul style="list-style-type: none"> 当 SDAAn 引脚为低电平时 当 IICEn 位为“0” (停止运行) 时 当复位时 		<ul style="list-style-type: none"> 当 SDAAn 引脚为高电平时

SMCn	运行模式的切换
0	在标准模式中运行 (最大传送速率: 100kbps)。
1	在快速模式 (最大传送速率: 400kbps) 或者增强型快速模式 (最大传送速率: 1Mbps) 中运行。

DFCn	数字滤波器的运行控制
0	数字滤波器 OFF
1	数字滤波器 ON
必须在快速模式或者增强型快速模式中使用数字滤波器。 数字滤波器用于消除噪声。 无论是将 DFCn 位置“1”还是清“0”，传送时钟都不变。	

PRSn	IICA 运行时钟 (f_{MCK}) 的控制
0	选择 f_{CLK} ($1MHz \leq f_{CLK} \leq 20MHz$)。
1	选择 $f_{CLK}/2$ ($20MHz < f_{CLK}$)。

注意 1. IICA 运行时钟 (f_{MCK}) 的最大工作频率为 20MHz(Max.)。只有在 f_{CLK} 超过 20MHz 时才必须将 IICA 控制寄存器 n1 (IICCTLn1) 的 bit0 (PRSn) 置“1”。

2. 在设定传送时钟的情况下, 必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 最小工作频率取决于运行模式。

快速模式: $f_{CLK}=3.5MHz$ (Min.)

增强型快速模式: $f_{CLK}=10MHz$ (Min.)

标准模式: $f_{CLK}=1MHz$ (Min.)

备注 1. IICEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit7

2. n=0

12.3.6 IICA 低电平宽度设定寄存器 n (IICWLn)

此寄存器控制串行接口 IICA 输出的 SCLAn 引脚信号低电平宽度 (t_{LOW}) 和 SDAAn 引脚信号。

通过 8 位存储器操作指令设定 IICWLn 寄存器。

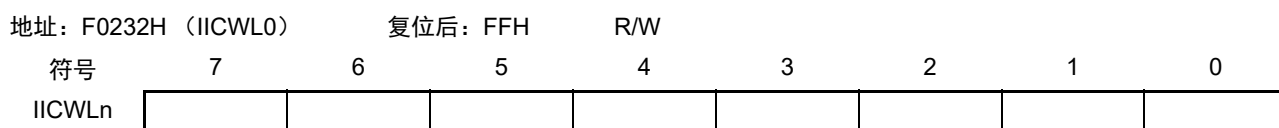
必须在禁止 I²C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICWLn 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

有关 IICWLn 寄存器的设定方法, 请参照“12.4.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法”。

数据保持时间为 IICWLn 所设时间的 1/4。

图 12-10 IICA 低电平宽度设定寄存器 n (IICWLn) 的格式



12.3.7 IICA 高电平宽度设定寄存器 n (IICWHn)

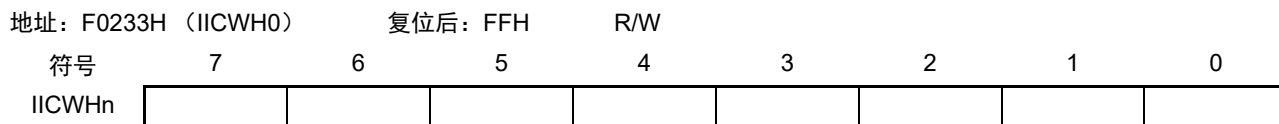
此寄存器控制串行接口 IICA 输出的 SCLAn 引脚信号高电平宽度和 SDAAn 引脚信号。

通过 8 位存储器操作指令设定 IICWHn 寄存器。

必须在禁止 I²C 运行 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) =0) 时设定 IICWHn 寄存器。

在产生复位信号后, 此寄存器的值变为“FFH”。

图 12-11 IICA 高电平宽度设定寄存器 n (IICWHn) 的格式



备注 1. 有关主控方传送时钟的设定方法, 请参照 12.4.2(1); 有关从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法, 请参照 12.4.2(2)。

2. n=0

12.3.8 端口模式寄存器 6 (PM6)

此寄存器以位为单位设定端口 6 的输入 / 输出。

在将 P60/SCLA0 引脚用作时钟输入 / 输出并且将 P61/SDAA0 引脚用作串行数据输入 / 输出时，必须将 PM60 和 PM61 以及 P60 和 P61 的输出锁存器置“0”。

当 IICEn 位 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit7) 为“0”时，P60/SCLA0 引脚和 P61/SDAA0 引脚为低电平输出 (固定)，因此必须在将 IICEn 位置“1”后切换到输出模式。

通过 1 位或者 8 位存储器操作指令设定 PM6 寄存器。

在产生复位信号后，此寄存器的值变为“FFH”。

图 12-12 端口模式寄存器 6 (PM6) 的格式

地址: FFF26H	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	PM62	PM61	PM60

PM6n	P6n 引脚的输入 / 输出模式的选择 (n=0 ~ 2)
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

12.4 I²C 总线模式的功能

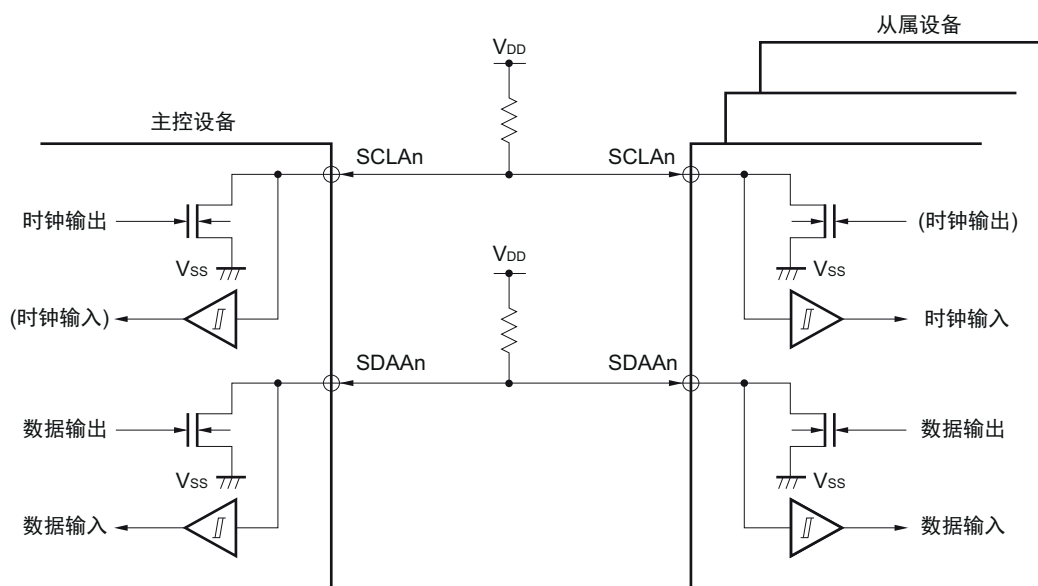
12.4.1 引脚结构

串行时钟引脚（SCLAn）和串行数据总线引脚（SDAAn）的结构如下。

- (1) SCLAn..... 串行时钟的输入/输出引脚
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。
- (2) SDAAn..... 串行数据的输入/输出复用引脚
主控设备和从属设备的输出都为N沟道漏极开路输出，输入都为施密特输入。

因为串行时钟线和串行数据总线的输出为 N 沟道漏极开路输出，所以需要外接上拉电阻。

图 12-13 引脚结构图



备注 n=0

12.4.2 通过 IICWLn 寄存器和 IICWHn 寄存器设定传送时钟的方法

(1) 主控方传送时钟的设定方法

$$\text{传送时钟} = \frac{f_{\text{MCK}}}{\text{IICWL} + \text{IICWH} + f_{\text{MCK}}(t_{\text{R}} + t_{\text{F}})}$$

此时，IICWLn 寄存器和 IICWHn 寄存器的最佳设定值如下：
(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWLn} = \frac{0.52}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.48}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWLn} = \frac{0.47}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.53}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWLn} = \frac{0.50}{\text{传送时钟}} \times f_{\text{MCK}}$$

$$\text{IICWHn} = \left(\frac{0.50}{\text{传送时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

(2) 从属方 IICWLn 寄存器和 IICWHn 寄存器的设定方法

(全部设定值的小数部分都舍入)

- 快速模式

$$\text{IICWLn} = 1.3\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (1.2\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 标准模式

$$\text{IICWLn} = 4.7\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (5.3\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- 增强型快速模式

$$\text{IICWLn} = 0.50\mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWHn} = (0.50\mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

注意 1. IICA 运行时钟 (f_{MCK}) 的最大工作频率为 20MHz(Max.)。只有在 f_{CLK} 超过 20MHz 时才必须将 IICA 控制寄存器 n1 (IICCTLn1) 的 bit0 (PRSn) 置“1”。

2. 在设定传送时钟的情况下，必须注意 f_{CLK} 的最小工作频率。串行接口 IICA 的 f_{CLK} 最小工作频率取决于运行模式。

快速模式： $f_{\text{CLK}}=3.5\text{MHz (Min.)}$

增强型快速模式： $f_{\text{CLK}}=10\text{MHz (Min.)}$

标准模式： $f_{\text{CLK}}=1\text{MHz (Min.)}$

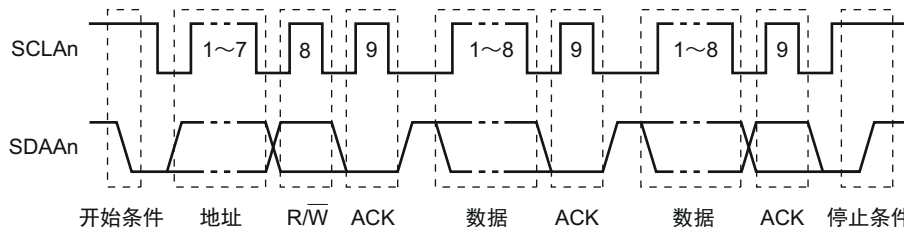
- 备注 1. 因为 SDAAn 信号和 SCLAn 信号的上升时间 (t_R) 和下降时间 (t_F) 因上拉电阻和布线电容而不同, 所以必须各自计算。
- IICWLn: IICA 低电平宽度设定寄存器 n
IICWHn: IICA 高电平宽度设定寄存器 n
 t_F : SDAAn 信号和 SCLAn 信号的下降时间
 t_R : SDAAn 信号和 SCLAn 信号的上升时间
 f_{MCK} : IICA 运行时钟频率
 - n=0

12.5 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 12-14 所示。

图 12-14 I²C 总线的串行数据传送时序



主控设备生成开始条件、从属地址和停止条件。

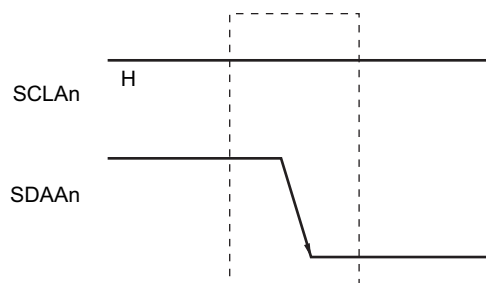
主控设备和从属设备都能生成应答 (ACK) (在一般情况下, 接收方输出 8 位数据)。

主控设备连续输出串行时钟 (SCLAn)。但是, 从属设备能延长 SCLAn 引脚的低电平期间并且插入等待。

12.5.1 开始条件

在 SCLAn 引脚为高电平时, 如果 SDAAn 引脚从高电平变为低电平, 就生成开始条件。SCLAn 引脚和 SDAAn 引脚的开始条件是在主控设备对从属设备开始串行传送时生成的信号。在用作从属设备时, 能检测到开始条件。

图 12-15 开始条件



在检测到停止条件 (SPDn: IICA 状态寄存器 n (IICSn) 的 bit0=1) 的状态下, 如果将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”, 就输出开始条件。如果检测到开始条件, 就将 IICSn 寄存器的 bit1 (STDn) 置“1”。

备注 n=0

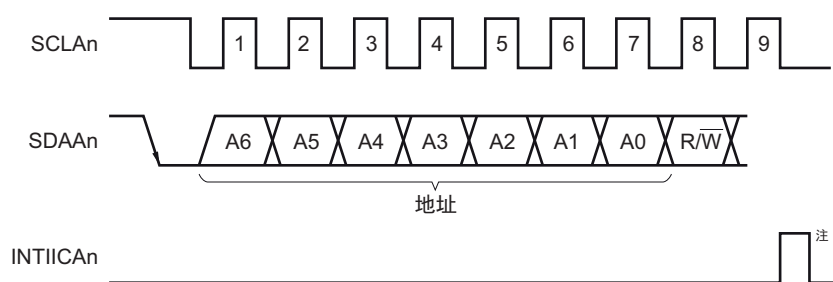
12.5.2 地址

开始条件的后续 7 位数据被定义为地址。

地址是主控设备为了从连接在总线的多个从属设备中选择特定的从属设备而输出的 7 位数据。因此，总线上的从属设备需要设定完全不同的地址。

从属设备通过硬件检测到开始条件，并且检查 7 位数据是否和从属地址寄存器 n (SVAn) 的内容相同。此时，如果 7 位数据和 SVAn 寄存器的值相同，该从属设备就被选中，在 7 位数据生成开始条件或者停止条件前，与主控设备进行通信。

图 12-16 地址



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

如果将从属地址和“12.5.3 传送方向的指定”中说明的传送方向构成的 8 位数据写到 IICA 移位寄存器 n (IICAn)，就输出地址。接收到的地址被写到 IICAn 寄存器。

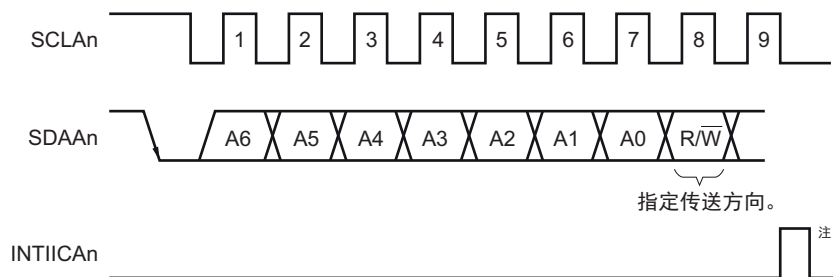
从属地址分配在 IICAn 寄存器的高 7 位。

12.5.3 传送方向的指定

主控设备在 7 位地址之后发送 1 位指定传送方向的数据。

当此传送方向指定位为“0”时，表示主控设备向从属设备发送数据；当此传送方向指定位为“1”时，表示主控设备从从属设备接收数据。

图 12-17 传送方向的指定



注 如果在从属运行时接收到本地站地址或者扩展码以外的数据，就不产生 INTIICAn。

备注 n=0

12.5.4 应答 (ACK)

能通过应答 (ACK) 确认发送方和接收方的串行数据状态。

接收方在每次接收到 8 位数据时返回应答。

通常, 发送方在发送 8 位数据后接收应答。当接收方返回应答时, 认为已正常接收, 继续处理。能通过 IICA 状态寄存器 n (IICSn) 的 bit2 (ACKDn) 确认应答的检测。

在主控设备为接收状态下接收到最后的数据时, 不返回应答而生成停止条件。在从属设备接收数据后不返回应答时, 主控设备输出停止条件或者重新开始条件, 中止发送。不返回应答的原因如下:

- ① 没有正常接收。
- ② 已结束最后数据的接收。
- ③ 不存在地址指定的接收方。

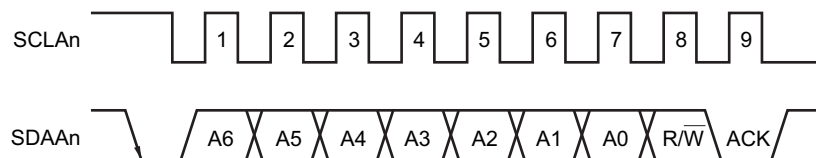
接收方在第 9 个时钟将 SDAAn 线置为低电平, 生成应答 (正常接收)。

通过将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit2 (ACKEn) 置“1”, 变为能自动生成应答的状态。通过 7 位地址信息后续的第 8 位数据设定 IICSn 寄存器的 bit3 (TRCn)。在接收 (TRCn=0) 的情况下, 通常必须将 ACKEn 位置“1”。

在从属接收运行过程中 (TRCn=0) 不能接收数据或者不需要下一个数据时, 必须将 ACKEn 位清“0”, 通知主控方不能接收数据。

在从属接收运行过程中 (TRCn=0) 不需要下一个数据时, 为了不生成应答, 必须将 ACKEn 位清“0”, 通知从属发送方数据的结束 (停止发送)。

图 12-18 应答



当接收到本地站的地址时, 与 ACKEn 位的值无关, 自动生成应答; 当接收到非本地站的地址时, 不生成应答 (NACK)。

在接收到扩展码时, 通过事先将 ACKEn 位置“1”, 生成应答。

接收数据时的应答生成方法因等待时序的设定而不同, 如下所示。

- 当选择 8 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =0):
通过在解除等待前将 ACKEn 位置“1”, 与 SCLAn 引脚的第 8 个时钟下降沿同步生成应答。
- 当选择 9 个时钟的等待时 (IICCTLn0 寄存器的 bit3 (WTIMn) =1):
通过事先将 ACKEn 位置“1”, 生成应答。

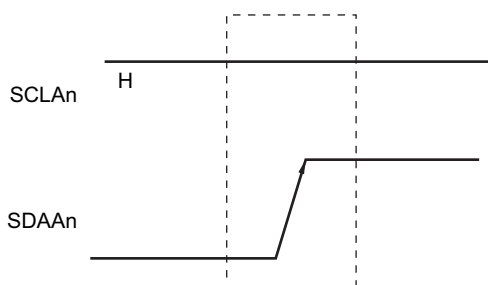
备注 n=0

12.5.5 停止条件

在 SCLAn 引脚为高电平时，如果 SDAAn 引脚从低电平变为高电平，就生成停止条件。

停止条件是在主控设备结束对从属设备的串行传送时生成的信号。在用作从属设备时，能检测到停止条件。

图 12-19 停止条件



如果将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit0 (SPTn) 置“1”，就生成停止条件。如果检测到停止条件，就将 IICA 状态寄存器 n (IICSn) 的 bit0 (SPDn) 置“1”，并且在 IICCTLn0 寄存器的 bit4 (SPIEn) 为“1”时产生 INTIICAn。

备注 n=0

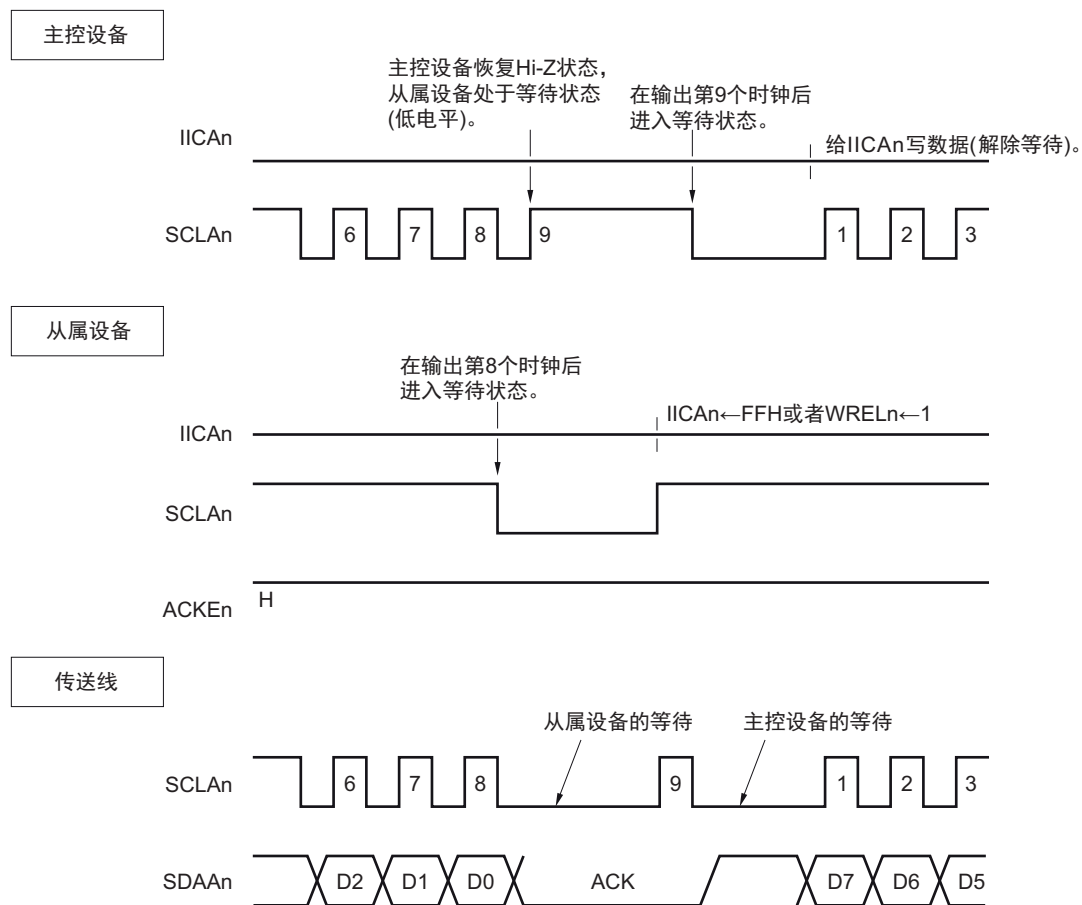
12.5.6 等待

通过等待来通知对方主控设备或者从属设备正在准备数据的发送 / 接收（等待状态）。

通过将 SCLAn 引脚置为低电平，通知对方处于等待状态。如果主控设备和从属设备的等待状态都被解除，就能开始下一次传送。

图 12-20 等待 (1/2)

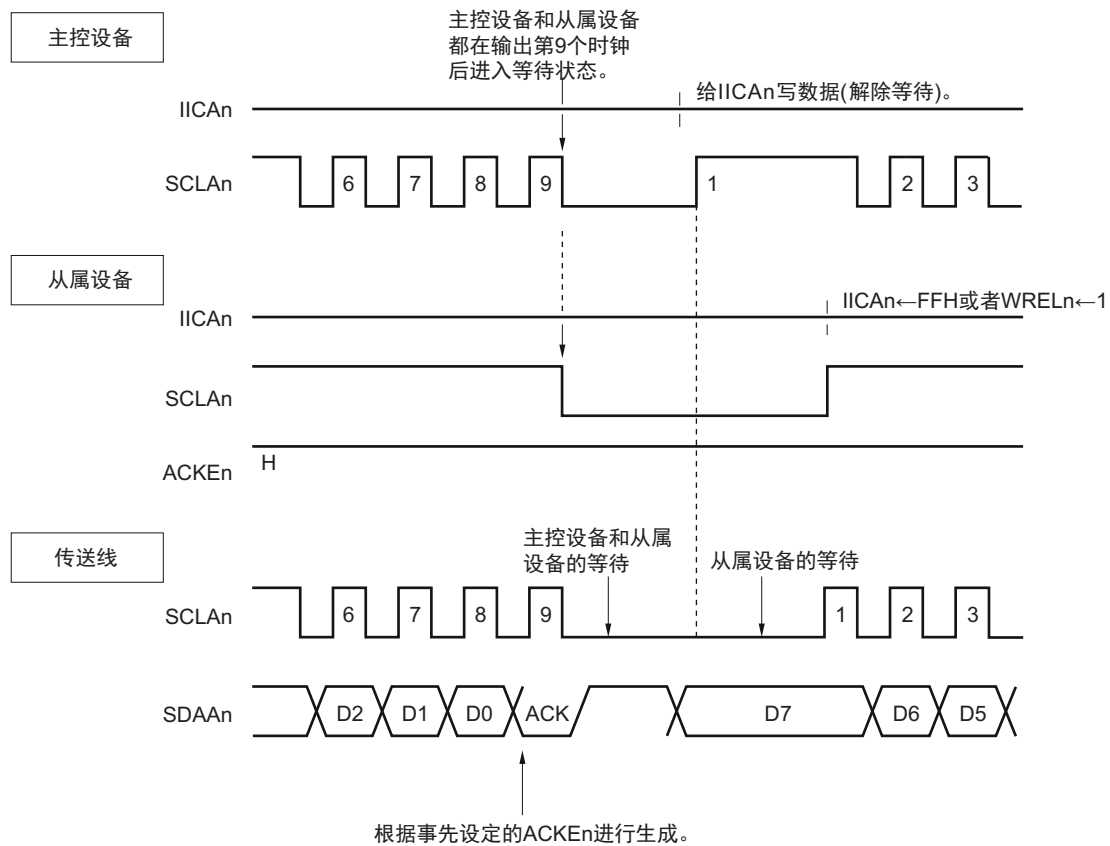
- (1) 主控设备为 9 个时钟等待，从属设备为 8 个时钟等待的情况
(主控设备：发送，从属设备：接收，ACKEn=1)



备注 n=0

图 12-20 等待 (2/2)

- (2) 主控设备和从属设备都为 9 个时钟等待的情况
(主控设备: 发送, 从属设备: 接收, ACKEn=1)



备注 ACKEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit2
WRELn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit5

通过设定 IICA 控制寄存器 n0 (IICCTLn0) 的 bit3 (WTIMn) 自动产生等待状态。

通常, 在接收方, 如果 IICCTLn0 寄存器的 bit5 (WRELn) 为“1”或者给 IICA 移位寄存器 n (IICAn) 写“FFH”, 就解除等待; 在发送方, 如果给 IICAn 寄存器写数据, 就解除等待。

主控设备还能通过以下方法解除等待:

- 将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”。
- 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

备注 n=0

12.5.7 等待的解除方法

在一般情况下，I²C 能通过以下的处理来解除等待。

- 给 IICA 移位寄存器 n (IICAn) 写数据。
- 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit5 (WRELn) 置位 (解除等待)。
- 将 IICCTLn0 寄存器的 bit1 (STTn) 置位 (生成开始条件) 注。
- 将 IICCTLn0 寄存器的 bit0 (SPTn) 置位 (生成停止条件) 注。

注 只限于主控设备。

如果执行了这些等待的解除处理，I²C 就解除等待，重新开始通信。

要在解除等待后发送数据 (包括地址) 时，必须给 IICAn 寄存器写数据。

要在解除等待后接收数据或者结束发送数据时，必须将 IICCTLn0 寄存器的 bit5 (WRELn) 置“1”。

要在解除等待后生成重新开始条件时，必须将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”。

要在解除等待后生成停止条件时，必须将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

对于一次等待只能执行一次解除处理。

例如，如果在通过将 WRELn 位置“1”来解除等待后给 IICAn 寄存器写数据，SDAAn 线的变化时序与 IICAn 寄存器的写时序就可能发生冲突，导致将错误的值输出到 SDAAn 线。

除了这些处理以外，在中途中止通信的情况下，如果将 IICEn 位清“0”，就停止通信，因此能解除等待。

在 I²C 总线状态因噪声而被死锁的情况下，如果将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”，就退出通信，因此能解除等待。

注意 如果在 WUPn 位为“1”时执行等待的解除处理，就不解除等待。

备注 n=0

12.5.8 中断请求（INTIICAn）的产生时序和等待控制

通过设定 IICA 控制寄存器 n0（IICCTLn0）的 bit3（WTIMn），在表 12-2 所示的时序产生 INTIICAn 并且进行等待控制。

表 12-2 INTIICAn 的产生时序和等待控制

WTIMn	从属运行			主控运行		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 注 1、2	8 注 2	8 注 2	9	8	8
1	9 注 1、2	9 注 2	9 注 2	9	9	9

注 1. 只有在接收的地址和从属地址寄存器 n（SVAn）的设定地址相同时，从属设备才在第 9 个时钟的下降沿产生 INTIICAn 信号并且进入等待状态。

此时，与 IICCTLn0 寄存器的 bit2（ACKEn）的设定无关，生成应答。接收到扩展码的从属设备在第 8 个时钟的下降沿产生 INTIICAn。如果在重新开始后地址不同，就在第 9 个时钟的下降沿产生 INTIICAn，但是不进入等待状态。

2. 如果接收的地址和从属地址寄存器 n（SVAn）的内容不同并且未接收到扩展码，就不产生 INTIICAn 并且也不进入等待状态。

备注 表中的数字表示串行时钟的时钟数。中断请求和等待控制都与串行时钟的下降沿同步。

(1) 地址的发送和接收

- 从属运行：与 WTIMn 位无关，根据上述注 1 和注 2 的条件决定中断和等待的时序。
- 主控运行：与 WTIMn 位无关，在第 9 个时钟的下降沿产生中断和等待的时序。

(2) 数据接收

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

(3) 数据发送

- 主控运行/从属运行：通过 WTIMn 位决定中断和等待的时序。

(4) 等待的解除方法

等待的解除方法有以下 4 种：

- 给 IICA 移位寄存器 n（IICAn）写数据。
- 将 IICA 控制寄存器 n0（IICCTLn0）的 bit5（WRELn）置位（解除等待）。
- 将 IICCTLn0 寄存器的 bit1（STTn）置位（生成开始条件）注。
- 将 IICCTLn0 寄存器的 bit0（SPTn）置位（生成停止条件）注。

注 只限于主控设备。

当选择 8 个时钟的等待（WTIMn=0）时，需要在解除等待前决定是否生成应答。

(5) 停止条件的检测

如果检测到停止条件，就产生 INTIICAn（只限于 SPIEn=1 的情况）。

备注 n=0

12.5.9 地址匹配的检测方法

在 I²C 总线模式中，主控设备能通过发送从属地址来选择特定的从属设备。

能通过硬件自动检测地址匹配。当主控设备发送的从属地址和从属地址寄存器 n (SVAn) 的设定地址相同或者只接收到扩展码时，产生 INTIICAn 中断请求。

12.5.10 错误的检测

在 I²C 总线模式中，因为发送过程中的串行数据总线 (SDAAn) 的状态被取到发送器件的 IICA 移位寄存器 n (IICAn)，所以能通过将开始发送前和发送结束后的 IICA 数据进行比较来检测发送错误。此时，如果 2 个数据不同，就判断为发生了发送错误。

12.5.11 扩展码

- (1) 当接收地址的高 4 位为“0000”或者“1111”时，作为接收到扩展码，将扩展码接收标志 (EXCn) 置“1”，并且在第 8 个时钟的下降沿产生中断请求 (INTIICAn)。不影响保存在从属地址寄存器 n (SVAn) 的本地站地址。
- (2) 当 SVAn 寄存器的设定值为“11110xx0”时，如果通过 10 位地址传送从主控设备发送“11110xx0”，就发生以下的置位。但是，在第 8 个时钟的下降沿产生中断请求 (INTIICAn)。
 - 高 4 位数据相同：EXCn=1
 - 7 位数据相同：COIn=1

备注 EXCn: IICA 状态寄存器 n (IICSn) 的 bit5
COIn: IICA 状态寄存器 n (IICSn) 的 bit4

- (3) 中断请求发生后的处理因扩展码的后续数据而不同，通过软件进行处理。如果在从属运行时接收到扩展码，即使地址不同也在参加通信。例如，在接收到扩展码后不想作为从属设备运行时，必须将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit6 (LRELn) 置“1”，进入下一次通信的待机状态。

表 12-3 主要扩展码的位定义

从属地址	R/W 位	说明
0 0 0 0 0 0 0	0	全呼地址
1 1 1 1 0 x x	0	10 位从属地址的指定 (地址认证时)
1 1 1 1 0 x x	1	10 位从属地址的指定 (在地址相同后发行读命令时)

备注 1. 有关上述以外的扩展码，请参照 NXP 公司发行的 I²C 总线规格书。

2. n=0

12.5.12 仲裁

当多个主控设备同时生成开始条件时（在 $STDn$ 位变为“1”前将 $STTn$ 位置“1”的情况），边调整时钟边进行主控设备的通信，直到数据不同为止。此运行称为仲裁。

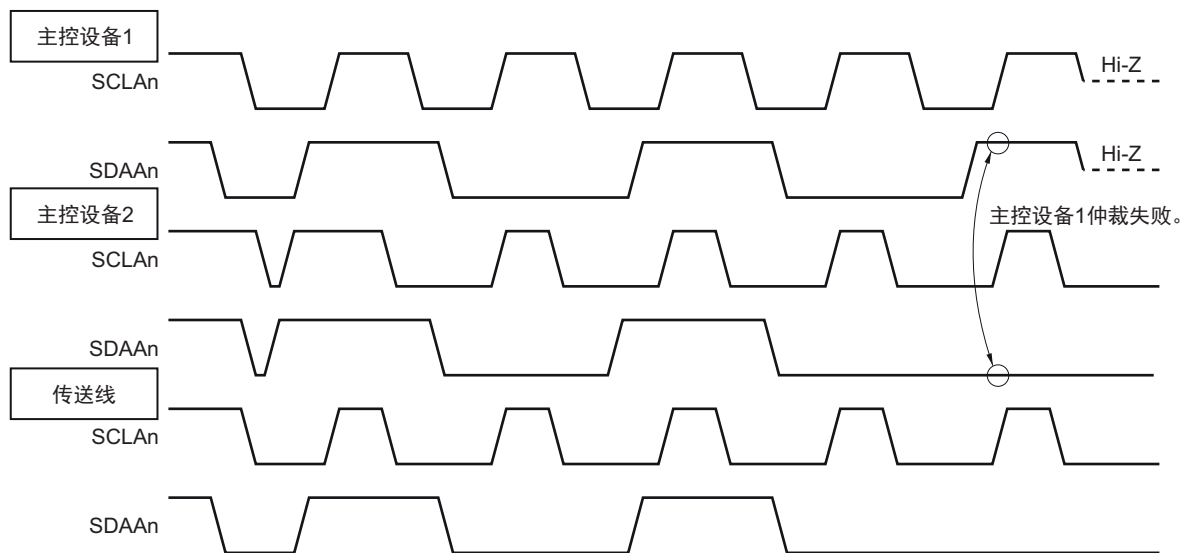
在仲裁失败时，仲裁失败的主控设备将 IICA 状态寄存器 n（ $IICSn$ ）的仲裁失败标志（ $ALDn$ ）置“1”，并且将 $SCLAn$ 线和 $SDAAn$ 线都置为高阻抗状态，释放总线。

在发生下一次中断请求时（例如：在第 8 或者第 9 个时钟检测到停止条件），用软件通过 $ALDn$ 位为“1”来检测仲裁的失败。

有关中断请求的产生时序，请参照“12.5.8 中断请求（ $INTIICAn$ ）的产生时序和等待控制”。

备注 $STDn$: IICA 状态寄存器 n（ $IICSn$ ）的 bit1
 $STTn$: IICA 控制寄存器 n0（ $IICCTLn0$ ）的 bit1

图 12-21 仲裁时序例子



备注 $n=0$

表 12-4 发生仲裁时的状态和中断请求的产生时序

发生仲裁时的状态	中断请求的产生时序
地址发送过程中	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
发送地址后的读写信息	
扩展码发送过程中	
发送扩展码后的读写信息	
数据发送过程中	
发送数据后的应答传送过程中	
在数据传送过程中检测到重新开始条件。	
在数据传送过程中检测到停止条件。	在生成停止条件时 (SPIEn=1) ^{注 2}
想要生成重新开始条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
想要生成重新开始条件, 但是检测到停止条件。	在生成停止条件时 (SPIEn=1) ^{注 2}
想要生成停止条件, 但是数据为低电平。	在字节传送后的第 8 或者第 9 个时钟的下降沿 ^{注 1}
想要生成重新开始条件, 但是 SCLAn 为低电平。	

注 1. 当 WTIMn 位 (IICA 控制寄存器 n0 (IICCTLn0) 的 bit3) 为 “1” 时, 在第 9 个时钟的下降沿产生中断请求; 当 WTIMn 位为 “0” 并且接收到扩展码的从属地址时, 在第 8 个时钟的下降沿产生中断请求。

2. 当有可能发生仲裁时, 必须在主控运行时将 SPIEn 位置 “1”。

备注 1. SPIEn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit4

2. n=0

12.5.13 唤醒功能

这是 I²C 的从属功能，是在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）的功能。

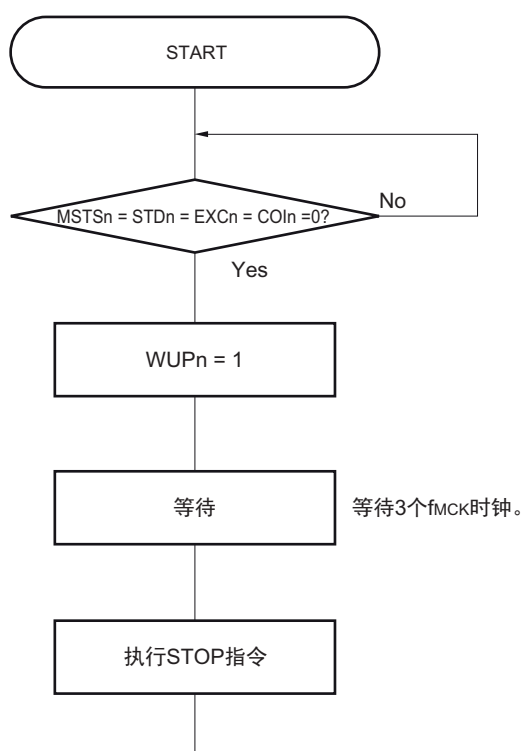
在地址不同的情况下不产生不需要的 INTIICAn 信号，能提高处理效率。

如果检测到开始条件，就进入唤醒待机状态。因为主控设备（已经生成开始条件的情况）也有可能因仲裁失败而变为从属设备，所以在发送地址的同时进入唤醒待机状态。

要在 STOP 模式中使用唤醒功能时，必须将 WUPn 位置“1”。与运行时钟无关而能接收地址。即使在这种情况下，也在接收到本地站地址和扩展码时产生中断请求信号（INTIICAn）。在产生此中断后，通过指令将 WUPn 位清“0”，返回到通常运行。

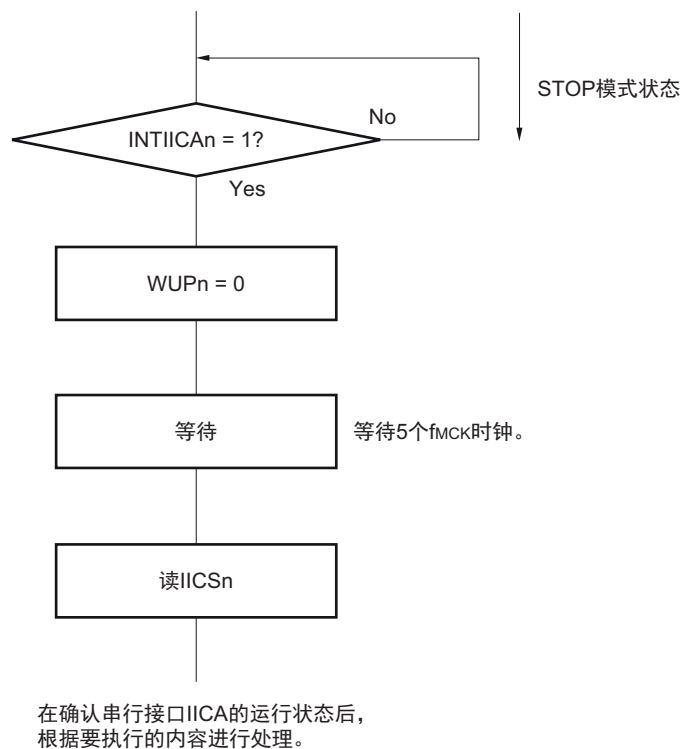
将 WUPn 位置“1”时的流程如图 12-22 所示，通过地址匹配将 WUPn 位置“0”时的流程如图 12-23 所示。

图 12-22 将 WUPn 位置“1”时的流程



备注 n=0

图 12-23 通过地址匹配将 WUPn 位置“0”时的流程（包括接收扩展码）

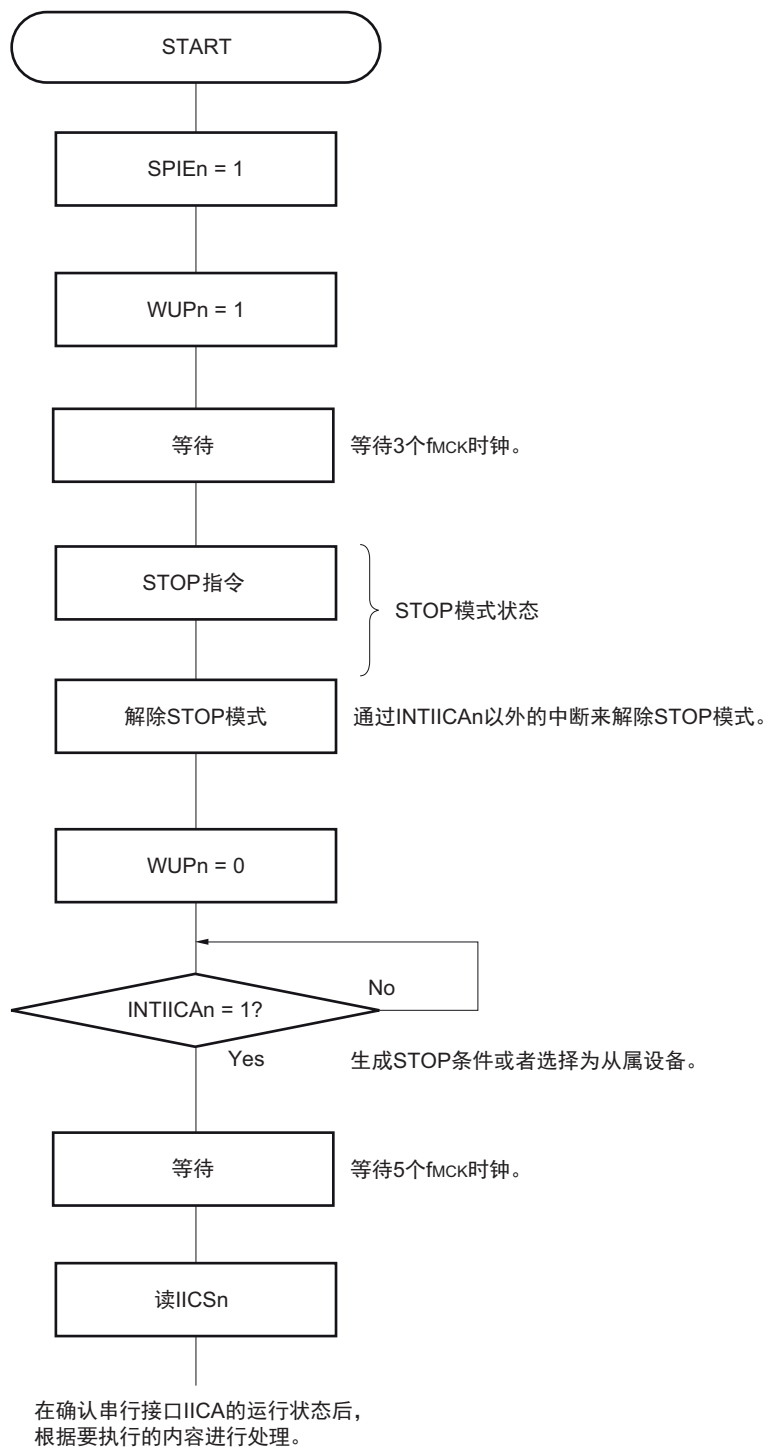


除了串行接口 IICA 产生的中断请求（INTIICAn）以外，必须通过以下的流程解除 STOP 模式。

- 下一次 IIC 通信为主控设备运行的情况：图 12-24 的流程
- 下一次 IIC 通信为从属设备运行的情况：
 - 通过 INTIICAn 中断返回的情况：和图 12-23 的流程相同。
 - 通过 INTIICAn 中断以外的中断返回的情况：必须在产生 INTIICAn 中断前保持 WUPn 位为“1”的状态继续运行。

备注 n=0

图 12-24 在通过 INTIICAn 以外的中断来解除 STOP 模式后作为主控设备运行的情况



备注 n=0

12.5.14 通信预约

(1) 允许通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit 0 (IICRSVn) =0)

要在不加入总线的状态下进行下一次主控通信时，能通过通信预约在释放总线时发送开始条件。此时的不加入总线包括以下 2 种状态：

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时（不返回应答而将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit6 (LRELn) 置“1”，退出通信后释放了总线)

如果在不加入总线的状态下将 IICCTLn0 寄存器的 bit1 (STTn) 置“1”，就在释放总线后（检测到停止条件）自动生成开始条件，进入等待状态。

将 IICCTLn0 寄存器的 bit4 (SPIEn) 置“1”，在通过产生的中断请求信号 (INTIICAn) 检测到总线的释放（检测到停止条件）后，如果给 IICA 移位寄存器 n (IICAn) 写地址，就自动作为主控设备开始通信。在检测到停止条件前，给 IICAn 寄存器写的的数据无效。

当将 STTn 位置“1”时，根据总线状态决定是作为开始条件运行还是作为通信预约运行。

- 总线处于释放状态时 生成开始条件
- 总线未处于释放状态（待机状态）时 通信预约

在将 STTn 位置“1”并且经过等待时间后，通过 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 确认是否作为通信预约运行。

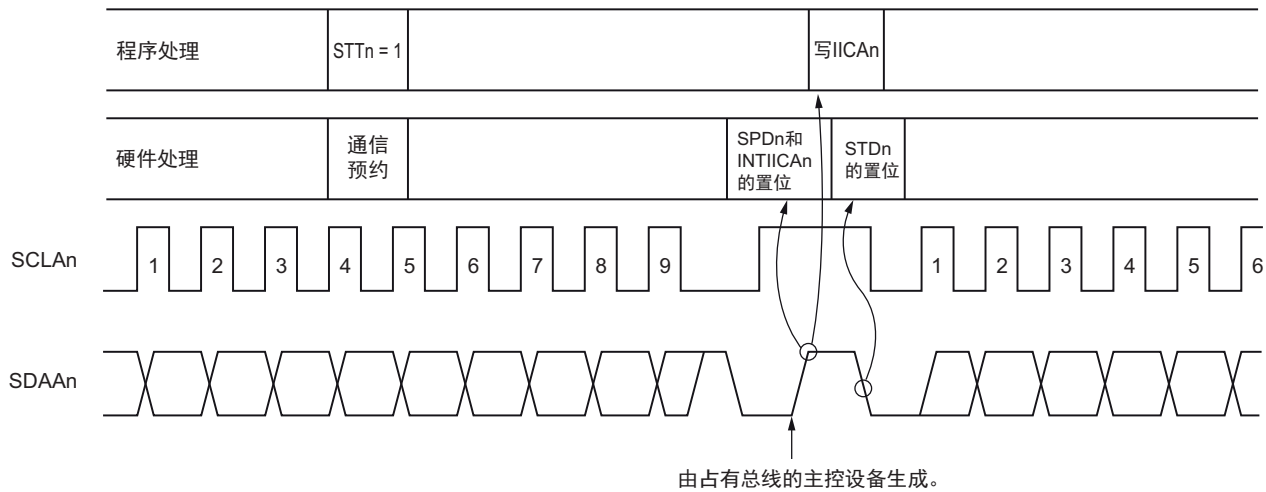
必须通过软件确保以下计算式计算的等待时间：

从将 STTn 位置“1”到确认 MSTSn 标志为止的等待时间：
 $(IICWLn \text{ 的设定值} + IICWHn \text{ 的设定值} + 4) / f_{MCK} + t_F \times 2$

- 备注 1. IICWLn: IICA 低电平宽度设定寄存器 n
 IICWHn: IICA 高电平宽度设定寄存器 n
 t_F : SDAAn 信号和 SCLAn 信号的下降时间
 f_{MCK} : IICA 运行时钟频率
2. n=0

通信预约的时序如图 12-25 所示。

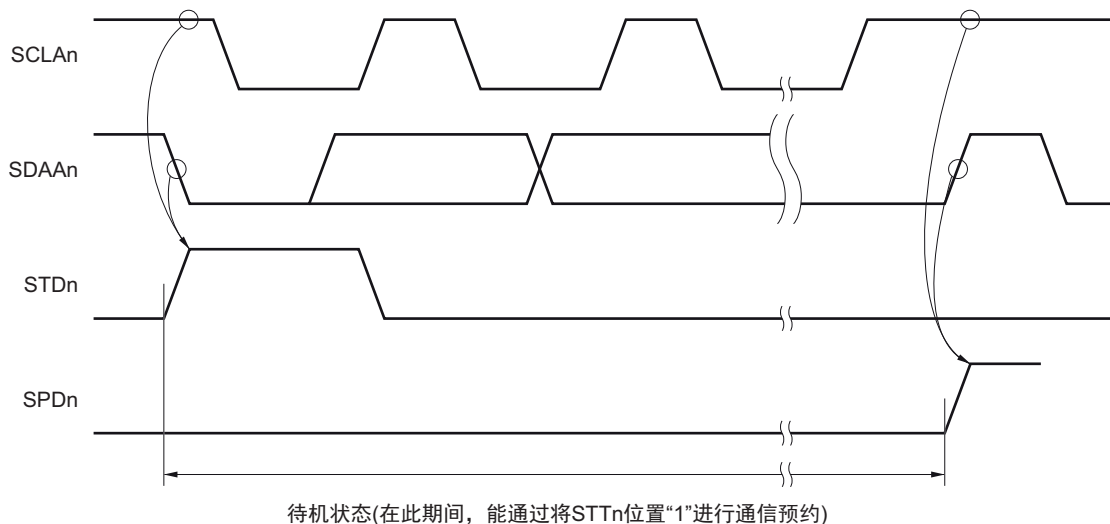
图 12-25 通信预约的时序



备注 IICAn: IICA 移位寄存器 n
 STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1
 STDn: IICA 状态寄存器 n (IICSn) 的 bit1
 SPDn: IICA 状态寄存器 n (IICSn) 的 bit0

通过图 12-26 所示的时序接受通信预约。在 IICA 状态寄存器 n (IICSn) 的 bit1 (STDn) 变为“1”后并且在检测到停止条件前，将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”进行通信预约。

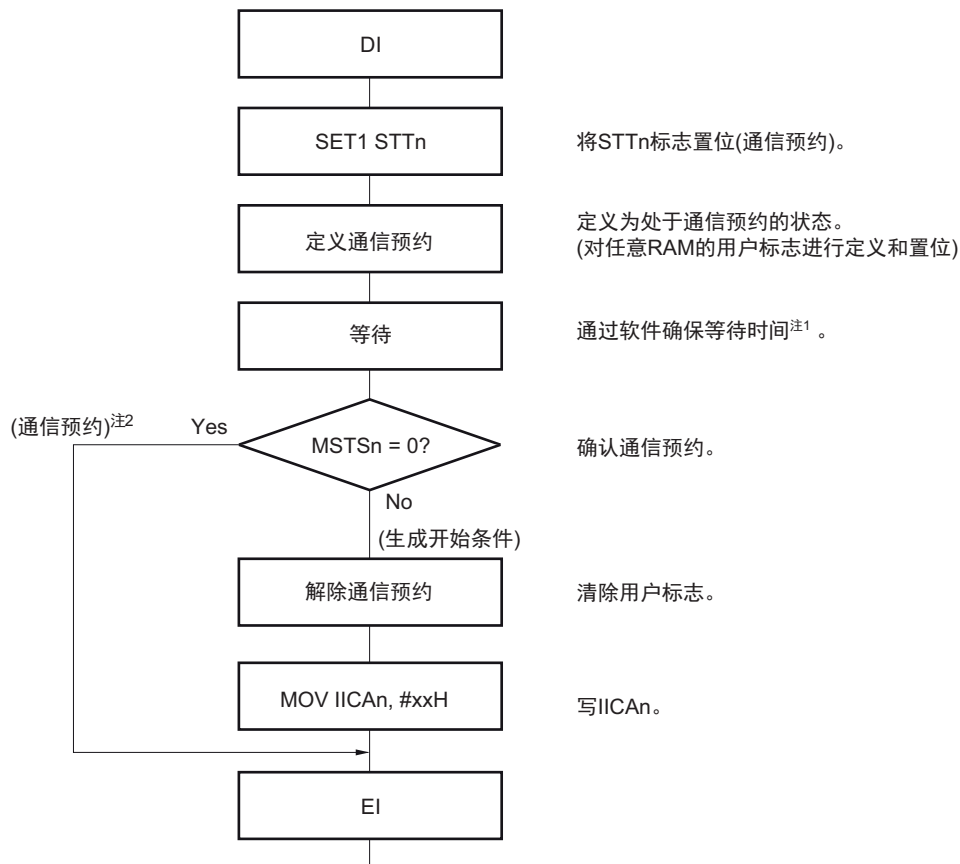
图 12-26 通信预约的接受时序



通信预约的步骤如图 12-27 所示。

备注 n=0

图 12-27 通信预约的步骤



注 1. 等待时间如下：

$$(\text{IICWLn 的设定值} + \text{IICWHn 的设定值} + 4) / f_{\text{MCK}} + t_{\text{F}} \times 2$$

2. 在通信预约运行时，通过停止条件中断请求来写 IICA 移位寄存器 n (IICAn)。

备注 1. STTn: IICA 控制寄存器 n0 (IICCTLn0) 的 bit1

MSTSn: IICA 状态寄存器 n (IICSn) 的 bit7

IICAn: IICA 移位寄存器 n

IICWLn: IICA 低电平宽度设定寄存器 n

IICWHn: IICA 高电平宽度设定寄存器 n

t_{F} : SDAAn 信号和 SCLAn 信号的下降时间

f_{MCK} : IICA 运行时钟频率

2. n=0

(2) 禁止通信预约功能的情况 (IICA 标志寄存器 n (IICFn) 的 bit0 (IICRSVn) =1)

在总线通信过程中, 如果在不参加此通信的状态下将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit1 (STTn) 置“1”, 就拒绝此请求而且不生成开始条件。此时的不加入总线包括以下 2 种状态:

- 在仲裁结果既不是主控设备也不是从属设备时
- 在接收到扩展码后不作为从属设备运行时 (不返回应答而将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 退出通信后释放了总线)

能通过 STCFn (IICFn 寄存器的 bit7) 来确认是生成了开始条件还是拒绝了请求。因为从 STTn 位为“1”到将 STCFn 位置“1”为止需要 5 个 f_{MCK} 时钟的时间, 所以必须通过软件确保此时间。

备注 n=0

12.5.15 其他注意事项**(1) STCENn 位为“0”的情况**

在刚允许 I²C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为通信状态 (IICBSYn=1)。要在没有检测到停止条件的状态下进行主控通信时, 必须先生成停止条件, 在释放总线后进行主控通信。

对于多主控, 在总线未释放 (未检测到停止条件) 的状态下不能进行主控通信。

按照以下顺序生成停止条件:

- ① 设定 IICA 控制寄存器 n1 (IICCTLn1)。
- ② 将 IICA 控制寄存器 n0 (IICCTLn0) 的 bit7 (IICEn) 置“1”。
- ③ 将 IICCTLn0 寄存器的 bit0 (SPTn) 置“1”。

(2) STCENn 位为“1”的情况

在刚允许 I²C 运行 (IICEn=1) 后, 与实际的总线状态无关而视为释放状态 (IICBSYn=0)。因此在生成第 1 个开始条件 (STTn=1) 时, 为了不破坏其他的通信, 需要确认总线已被释放。

(3) 正在和其他设备进行 I²C 通信的情况

在 SDAAn 引脚为低电平并且 SCLAn 引脚为高电平时, 如果允许 I²C 运行而且中途参加通信, I²C 的宏就视为 SDAAn 引脚从高电平变为低电平 (检测到开始条件)。如果此时总线上的值是能识别为扩展码的值, 就返回应答而妨碍和其他设备的 I²C 通信。为了避免这种情况, 必须按照以下顺序启动 I²C:

- ① 将 IICCTLn0 寄存器的 bit4 (SPIEn) 清“0”, 禁止在检测到停止条件时产生中断请求信号 (INTIICAn)。
- ② 将 IICCTLn0 寄存器的 bit7 (IICEn) 置“1”, 允许 I²C 运行。
- ③ 等待检测到开始条件。
- ④ 在返回应答前 (将 IICEn 位置“1”后的 4~72 个 f_{MCK} 时钟内) 将 IICCTLn0 寄存器的 bit6 (LRELn) 置“1”, 强制将检测置为无效。

(4) 在将 STTn 位和 SPTn 位 (IICCTLn0 寄存器的 bit1 和 bit0) 置位后, 禁止清“0”前的再置位。**(5) 如果进行了通信预约, 就必须将 SPIEn 位 (IICCTLn0 寄存器的 bit4) 置“1”, 在检测到停止条件时产生中断请求。在产生中断请求后, 通过给 IICA 移位寄存器 n (IICAn) 写通信数据, 开始传送。如果在检测到停止条件时不发生中断, 就因在开始通信时不产生中断请求而停止在等待状态。但是, 当通过软件检测 MSTSn 位 (IICA 状态寄存器 n (IICSn) 的 bit7) 时, 不需要将 SPIEn 位置“1”。**

备注 n=0

12.5.16 通信运行

在此通过流程图表示以下 3 个运行步骤。

(1) 单主控系统的主控运行

在单主控系统中用作主控设备的流程图如下所示。

此流程大体分为“初始设定”和“通信处理”。在启动时执行“初始设定”部分，如果需要和从属设备进行通信，就在进行通信时所需的准备后执行“通信处理”部分。

(2) 多主控系统的主控运行

在 I²C 总线的多主控系统中，只根据 I²C 总线的规格无法判断在参加通信的阶段总线是处于释放状态还是处于使用状态。在此，如果数据和时钟在一定时间内（1 帧）为高电平，就将总线作为释放状态而参加通信。

此流程大体分为“初始设定”、“通信等待”和“通信处理”。在此省略因仲裁失败而被指定为从属设备的处理，只表示用作主控设备的处理。在启动时执行“初始设定”部分后加入总线，然后通过“通信等待”，等待主控设备的通信请求或者从属设备的指定。实际进行通信的是“通信处理”部分，除了支持与从属设备进行数据发送和接收以外，还支持与其他主控设备的仲裁。

(3) 从属运行

用作 I²C 总线从属设备的例子如下所示。

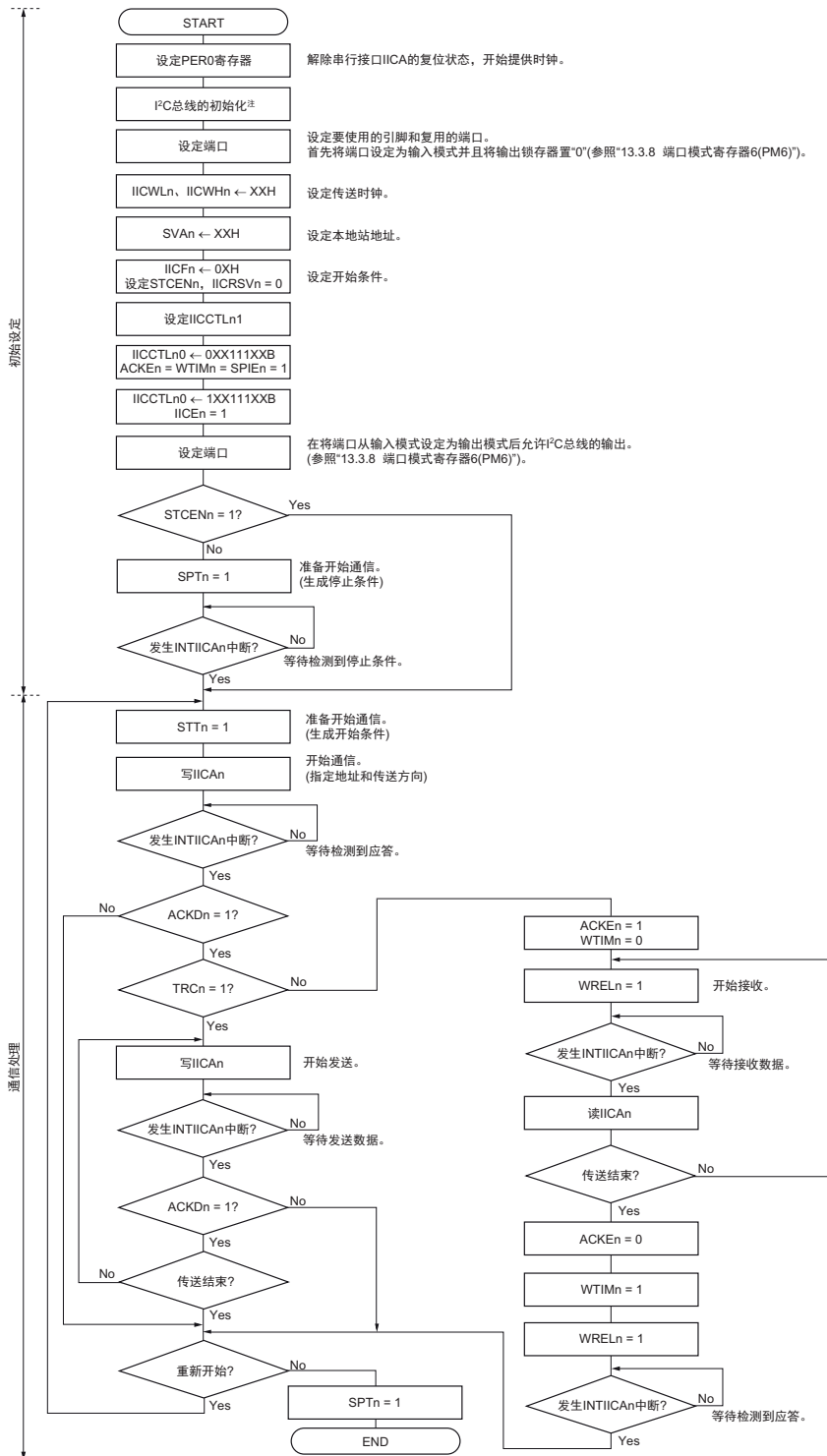
当用作从属设备时，通过中断开始运行。在启动时执行“初始设定”部分，然后通过“通信等待”，等待 INTIICAn 中断的发生。如果发生 INTIICAn 中断，就判断通信状态并且将标志传递给主处理部。

通过检查各标志，进行所需的“通信处理”。

备注 n=0

(1) 单主控系统的主控运行

图 12-28 单主控系统的主控运行

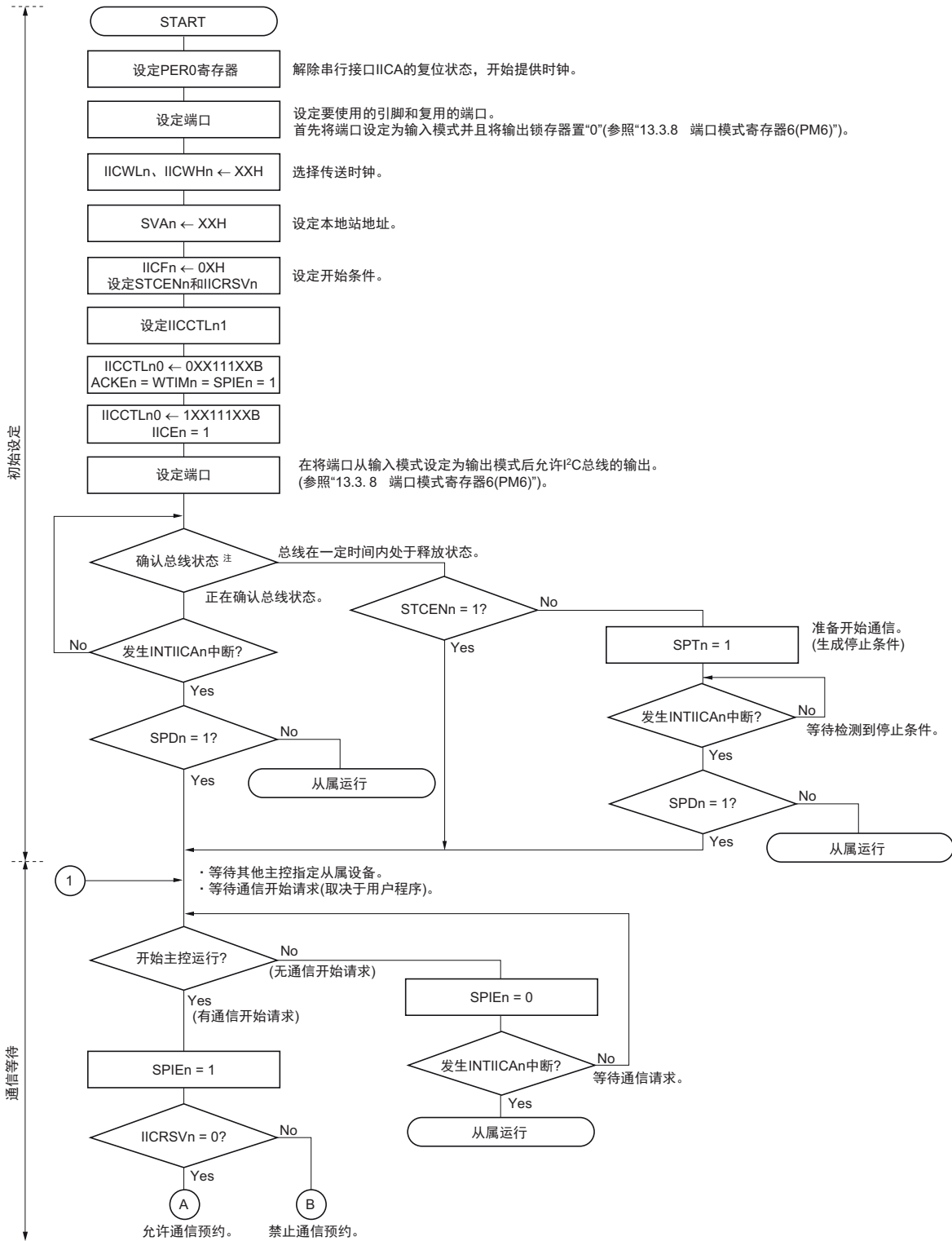


注 必须根据通信中的产品的规格，释放 I²C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。例如，如果 EEPROM 处于给 SDAAn 引脚输出低电平的状态，就必须将 SCLAn 引脚设定为输出端口，并且在 SDAAn 引脚固定为高电平前从输出端口输出时钟脉冲。

备注 1. 发送和接收的格式必须符合通信中的产品的规格。
2. n=0

(2) 多主控系统的主控运行

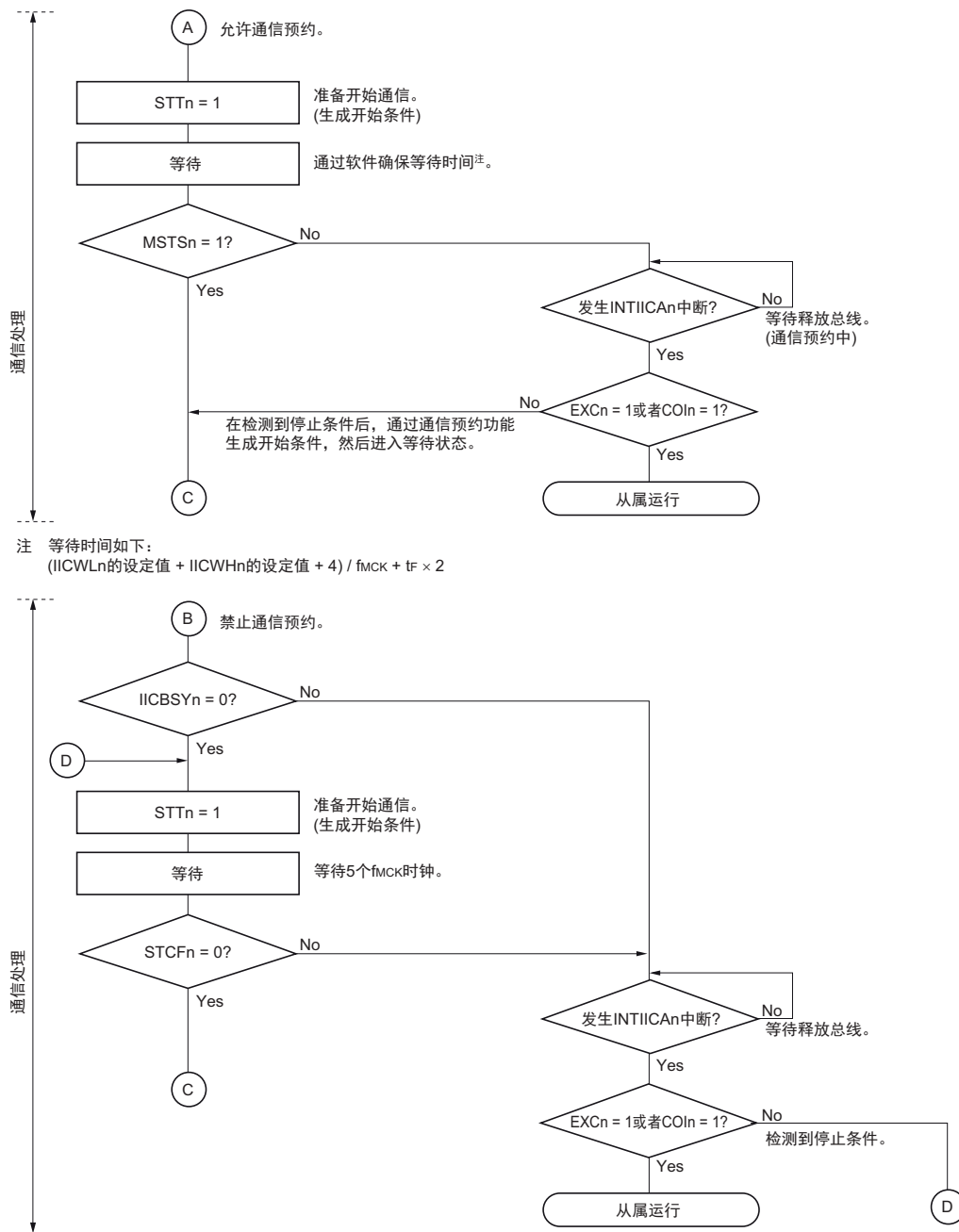
图 12-29 多主控系统的主控运行 (1/3)



注 必须确认总线在一定时间内（例如，1 帧）处于释放状态（CLDn 位 = 1、DADn 位 = 1）。当 SDAAn 引脚固定为低电平时，必须根据通信中的产品的规格，判断是否释放 I²C 总线（SCLAn 引脚和 SDAAn 引脚为高电平）。

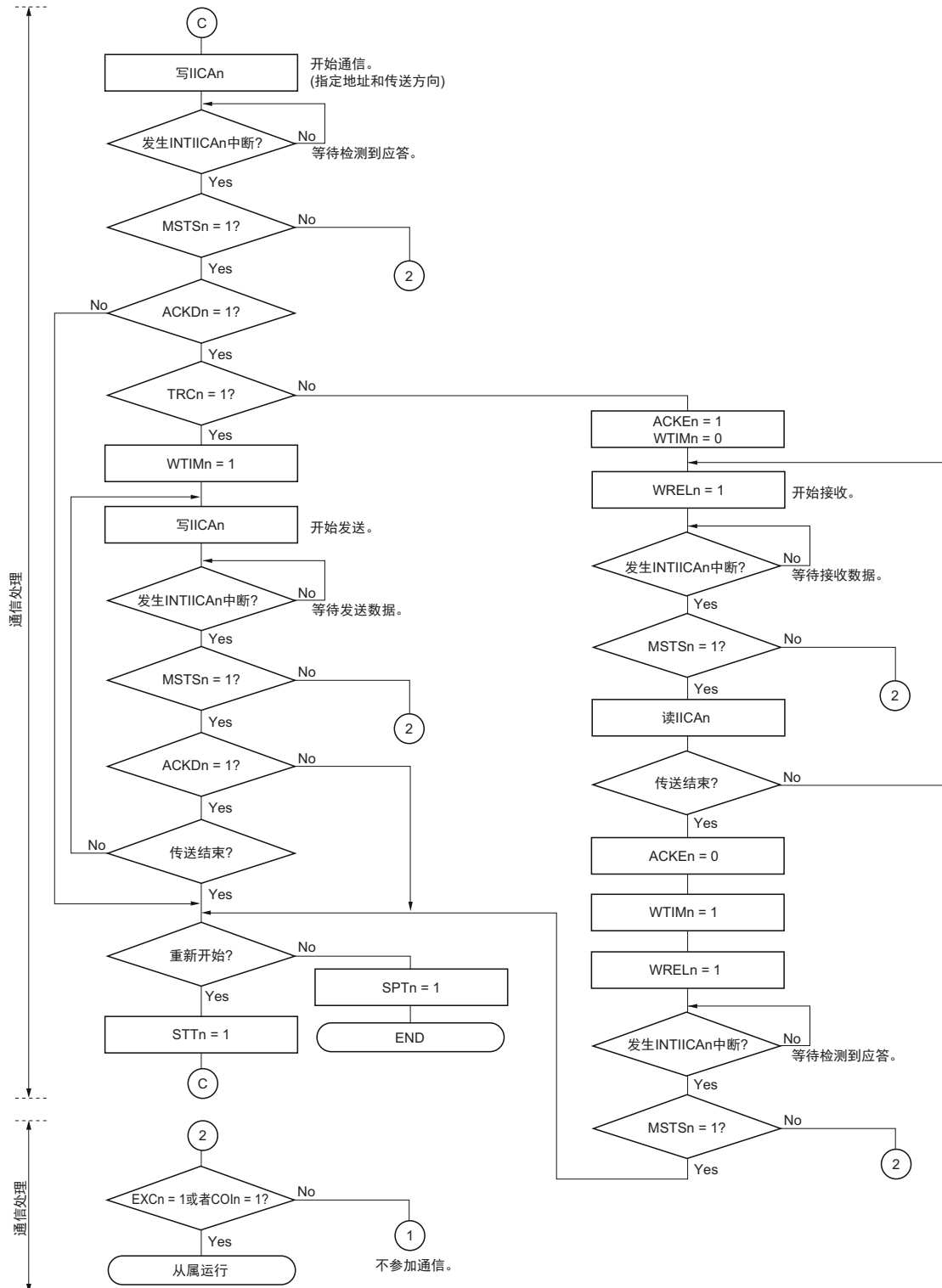
备注 n=0

图 12-29 多主控系统的主控运行 (2/3)



- 备注 1. IICWLn: IICA 低电平宽度设定寄存器 n
 IICWHn: IICA 高电平宽度设定寄存器 n
 t_F : SDAAn 信号和 SCLAn 信号的下降时间
 f_{MCK} : IICA 运行时钟频率
2. n=0

图 12-29 多主控系统的主控运行 (3/3)



备注 1. 传送和接收的格式必须符合通信中的产品的规格。

2. 在多主控系统中用作主控设备的情况下，必须在每次发生 INTIICAn 中断时读 MSTSn 位，确认仲裁结果。

3. 在多主控系统中用作从属设备的情况下，必须在每次发生 INTIICAn 中断时通过 IICA 状态寄存器 n (IICSn) 和 IICA 标志寄存器 n (IICFn) 确认状态，决定以后的处理。

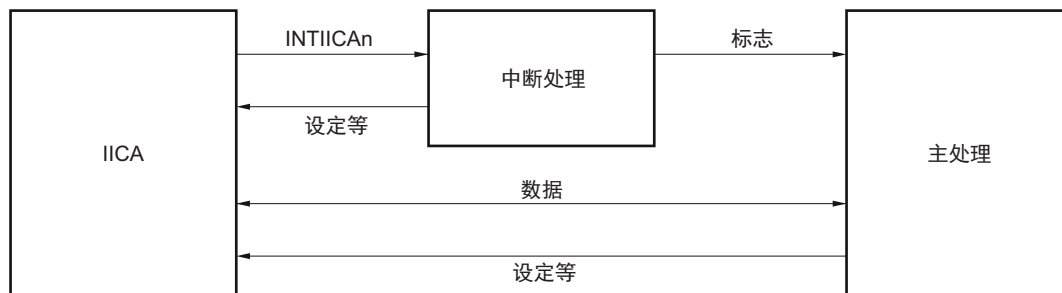
4. n=0

(3) 从属运行

从属运行的处理步骤如下所示。

从属运行基本上由事件驱动，因此需要通过 INTIICAn 中断进行处理（需要对通信中的停止条件检测等的运行状态进行很大的变更处理）。

在此说明中，假设数据通信不支持扩展码，INTIICAn 中断处理只进行状态转移处理并且实际的数据通信由主处理部进行。



因此，准备以下 3 个标志并且代替 INTIICAn 将标志传递给主处理部，进行数据通信处理。

① 通信模式标志

此标志表示以下 2 种通信状态：

- 清除模式：不在进行数据通信的状态
- 通信模式：正在进行数据通信的状态（有效地址的检测～停止条件的检测，未检测到主控设备的应答，地址不同）

② 就绪标志

此标志表示能进行数据通信。在通常的数据通信中，和 INTIICAn 中断相同，由中断处理部置位而由主处理部清除。在开始通信时，由中断处理部清除标志。但是，在发送第 1 个数据时，中断处理部不将就绪标志置位，因此在不清除标志的状态下发送第 1 个数据（地址匹配被解释为下一次数据请求）。

③ 通信方向标志

此标志表示通信方向，和 TRCn 位的值相同。

备注 n=0

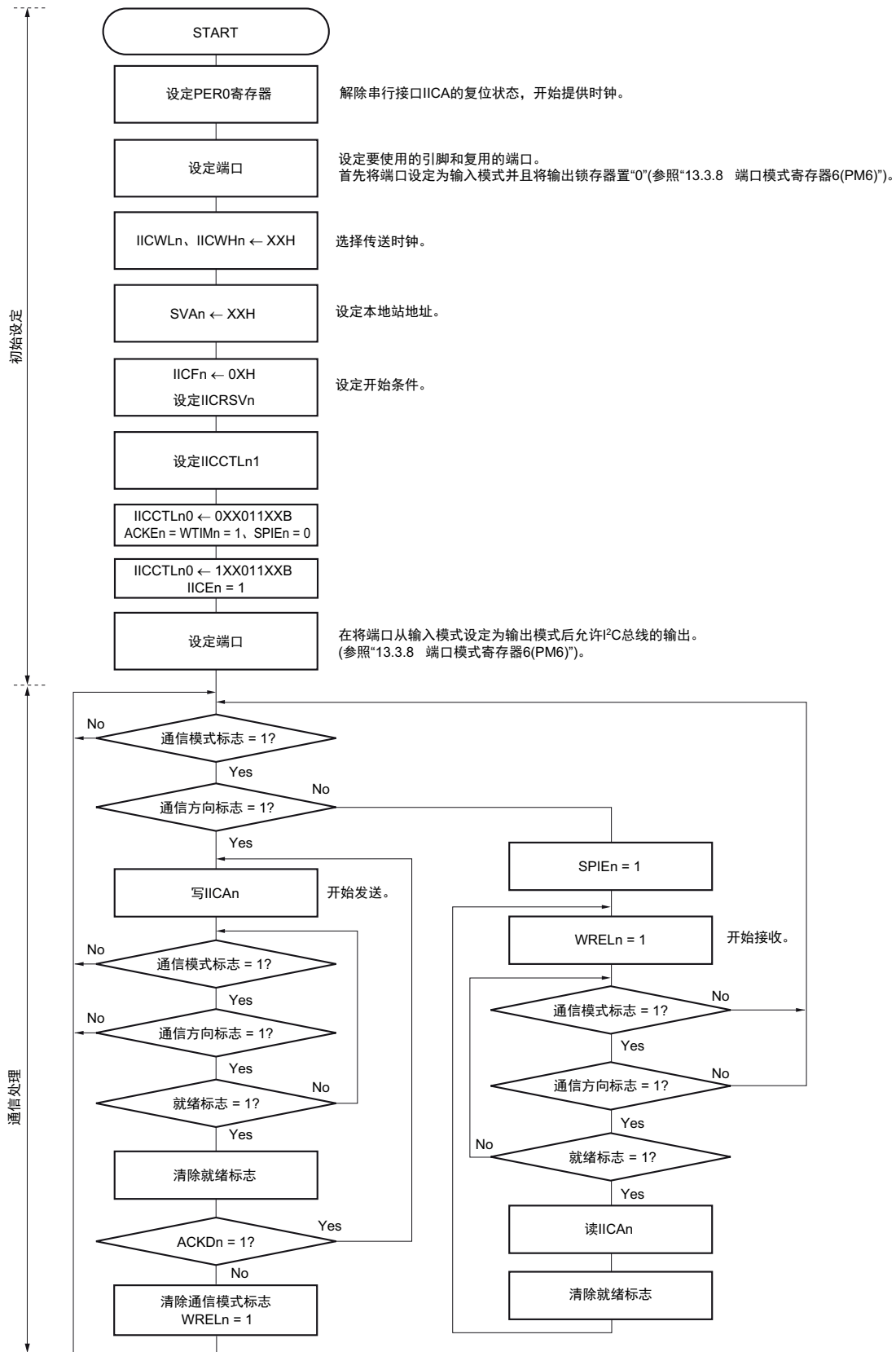
从属运行的主处理部的运行如下所示。

启动串行接口 IICA，等待变为可通信的状态。如果变为可通信的状态，就使用通信模式标志和就绪标志进行通信（因为通过中断进行停止条件和开始条件的处理，所以在此通过标志确认状态）。

在发送时，重复发送，直到主控设备不返回应答为止。如果主控设备不返回应答，就结束通信。

在接收时，接收所需数量的数据。如果通信结束，就在下一个数据时不返回应答。此后，主控设备生成停止条件或者重新开始条件，从而退出通信状态。

图 12-30 从属运行步骤 (1)



备注 1. 传送和接收的格式必须符合通信中的产品的规格。

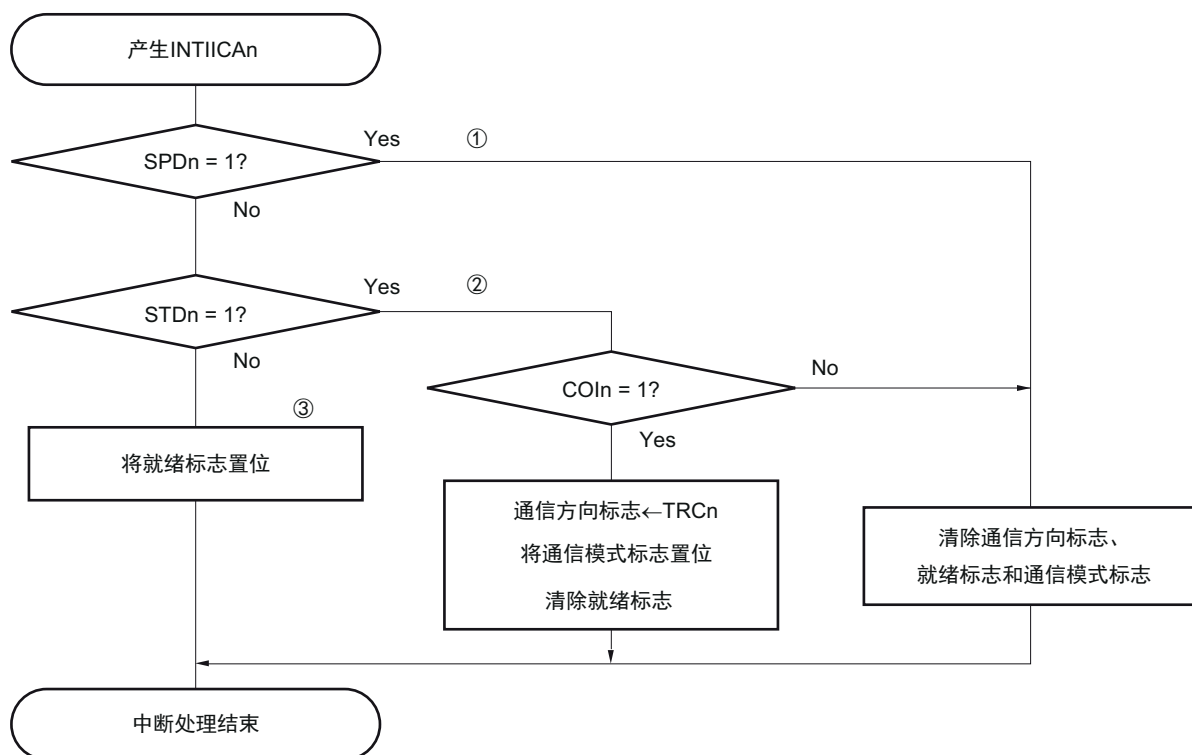
2. n=0

从属设备通过 INTIICAn 中断进行处理的步骤例子如下所示（在此假设没有用扩展码进行处理）。通过 INTIICAn 中断确认状态并且进行以下处理。

- ① 如果生成停止条件，就结束通信。
- ② 如果生成开始条件，就确认地址。如果地址不同，就结束通信。如果地址相同，就设定为通信模式并且解除等待，然后从中断返回（清除就绪标志）。
- ③ 当发送和接收数据时，只要将就绪标志置位，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 12-31 从属运行步骤 (2)”的①~③。

图 12-31 从属运行步骤 (2)



12.5.17 I²C 中断请求 (INTIICAn) 的产生时序

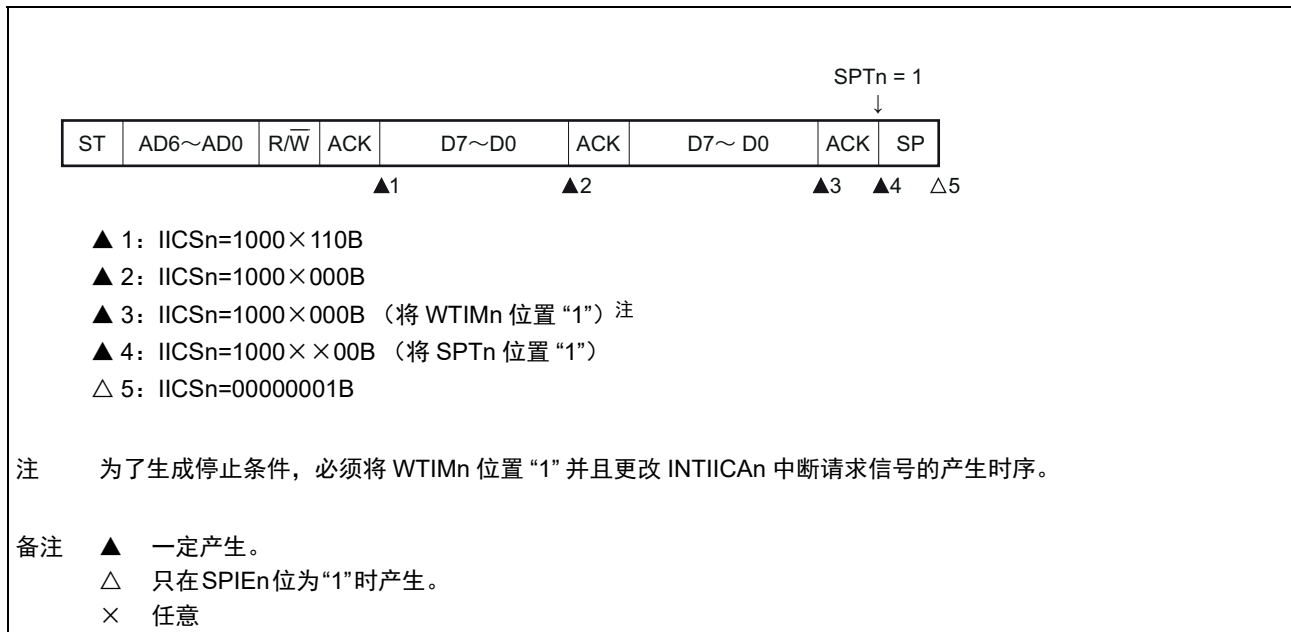
数据的发送和接收时序、INTIICAn 中断请求信号的产生时序以及产生 INTIICAn 信号时的 IICA 状态寄存器 n (IICSn) 的值如下所示。

- 备注 1. ST: 开始条件
 AD6 ~ AD0: 地址
 R/W: 传送方向的指定
 ACK: 应答
 D7 ~ D0: 数据
 SP: 停止条件
2. n=0

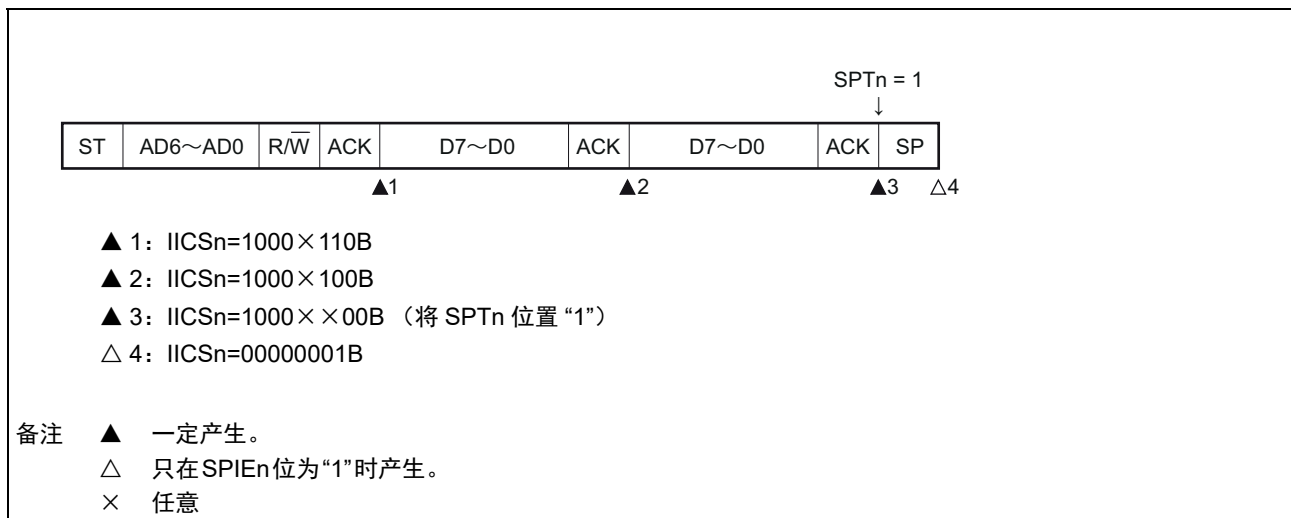
(1) 主控运行

(a) Start ~ Address ~ Data ~ Data ~ Stop (发送和接收)

(i) WTIMn=0的情况



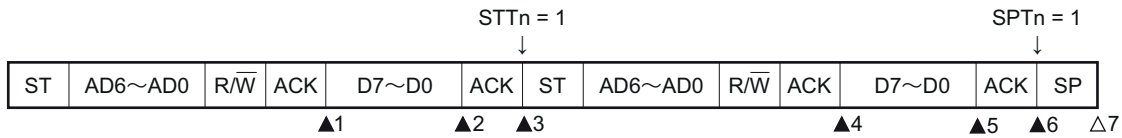
(ii) WTIMn=1的情况



备注 n=0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (重新开始)

(i) WTIMn=0的情况

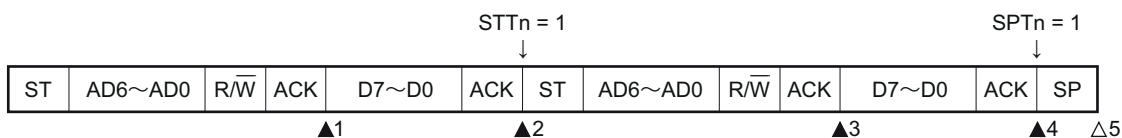


- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000×000B (将 WTIMn 位置“1”) 注 1
- ▲ 3: IICSn=1000××00B (将 WTIMn 位清“0”注 2 并且将 STTn 位置“1”)
- ▲ 4: IICSn=1000×110B
- ▲ 5: IICSn=1000×000B (将 WTIMn 位置“1”) 注 3
- ▲ 6: IICSn=1000××00B (将 SPTn 位置“1”)
- △ 7: IICSn=00000001B

- 注 1. 为了生成开始条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。
 注 2. 为了恢复原来的设定, 必须将 WTIMn 位清“0”。
 注 3. 为了生成停止条件, 必须将 WTIMn 位置“1”并且更改 INTIICAn 中断请求信号的产生时序。

备注 ▲ 一定产生
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

(ii) WTIMn=1的情况



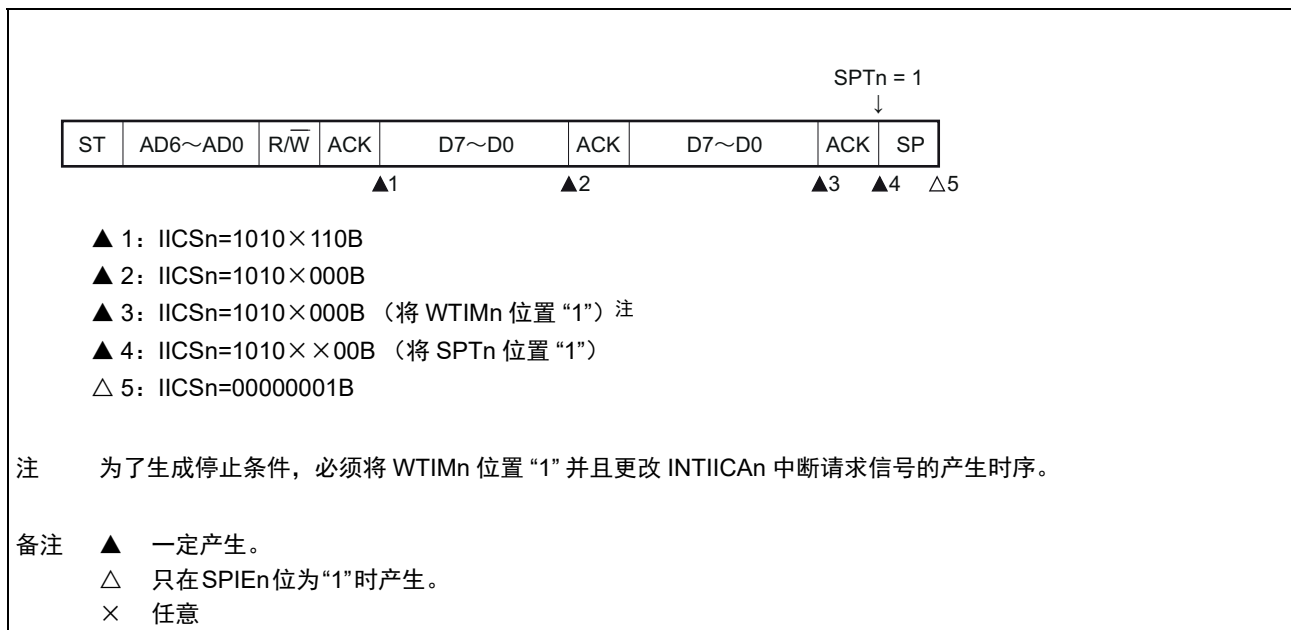
- ▲ 1: IICSn=1000×110B
- ▲ 2: IICSn=1000××00B (将 STTn 位置“1”)
- ▲ 3: IICSn=1000×110B
- ▲ 4: IICSn=1000××00B (将 SPTn 位置“1”)
- △ 5: IICSn=00000001B

备注 ▲ 一定产生。
 △ 只在 SPIEn 位为“1”时产生。
 × 任意

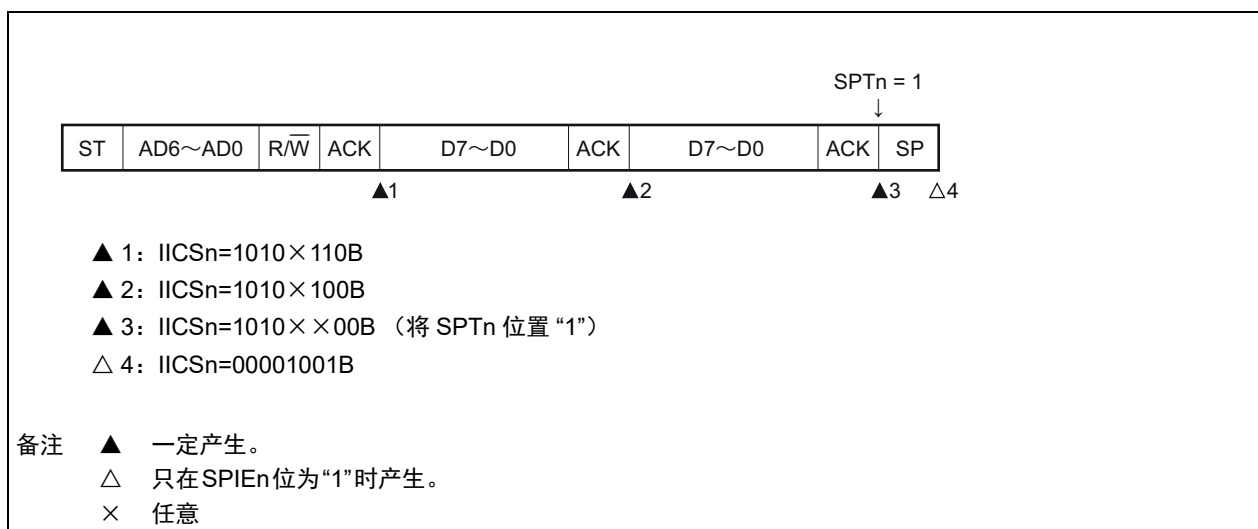
备注 n=0

(c) Start ~ Code ~ Data ~ Data ~ Stop (发送扩展码)

(i) WTIMn=0的情况



(ii) WTIMn=1的情况

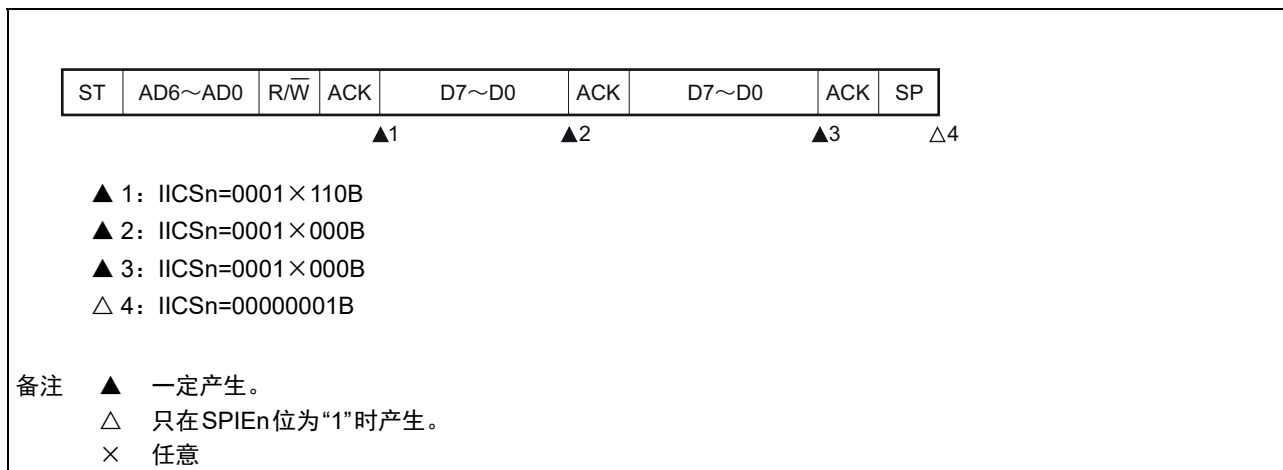


备注 n=0

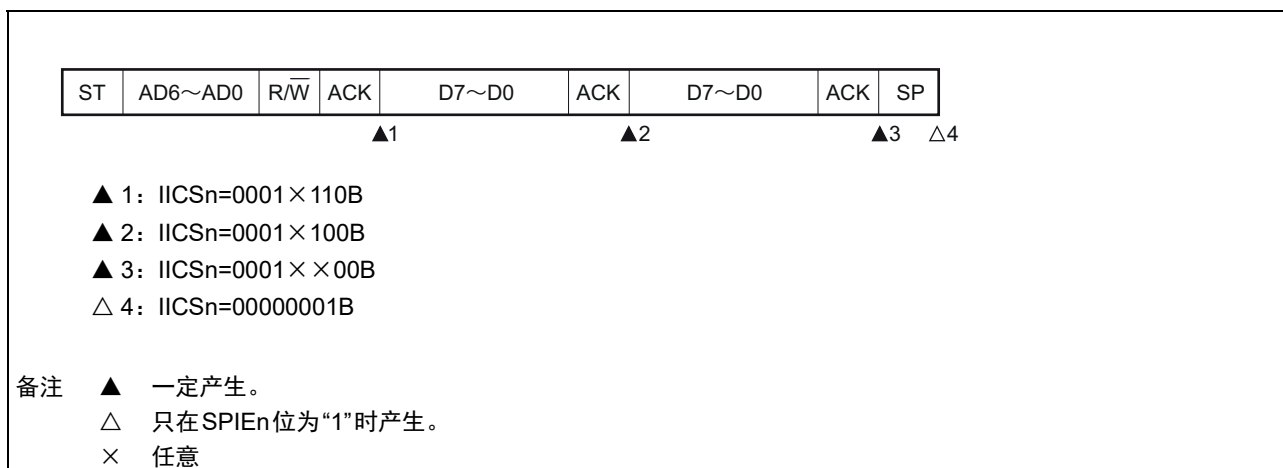
(2) 从属运行（接收从属地址的情况）

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) WTIMn=0 的情况



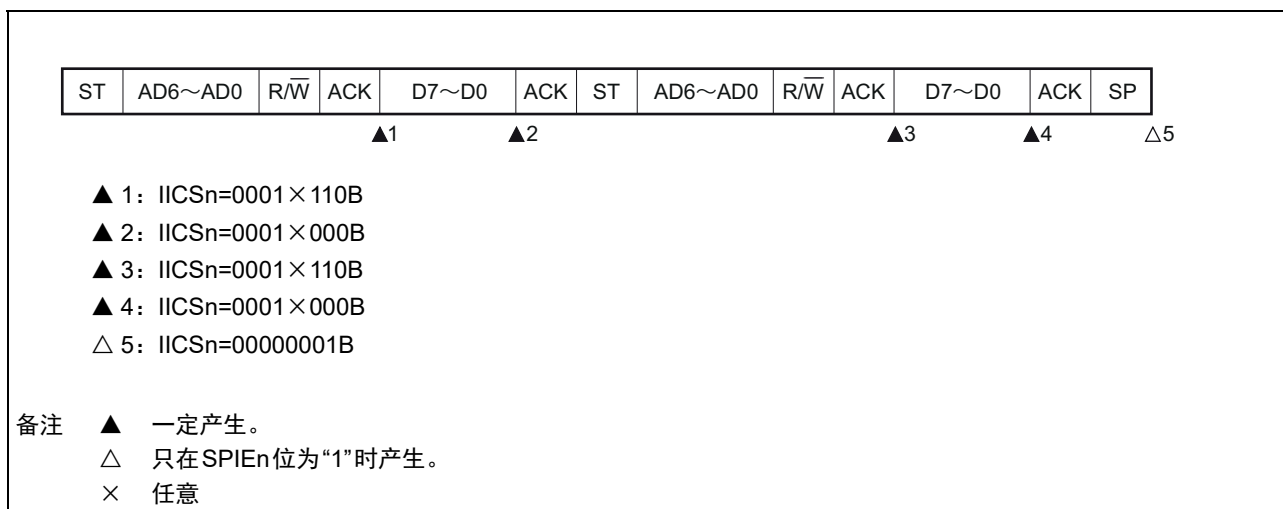
(ii) WTIMn=1 的情况



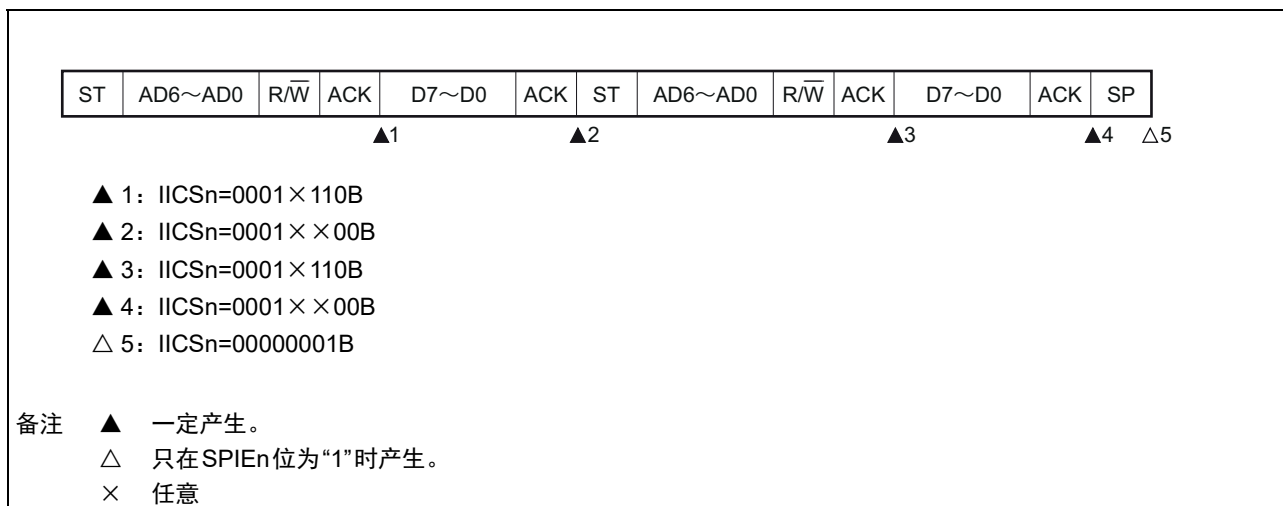
备注 n=0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0的情况 (在重新开始后SVAn相同)



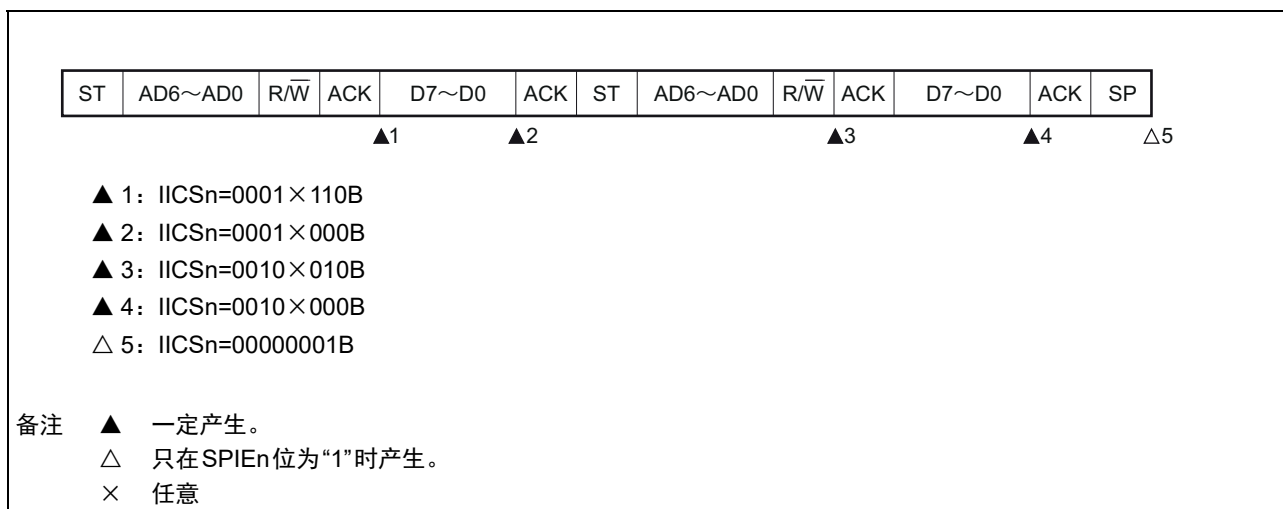
(ii) WTIMn=1的情况 (在重新开始后SVAn相同)



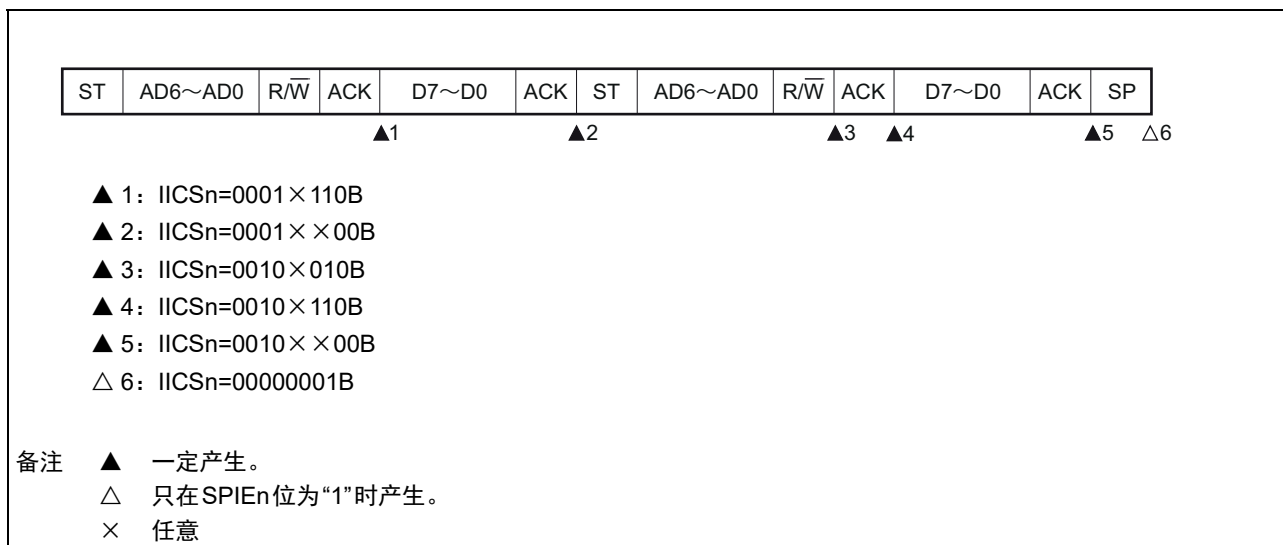
备注 n=0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIMn=0的情况 (在重新开始后地址不同 (扩展码))



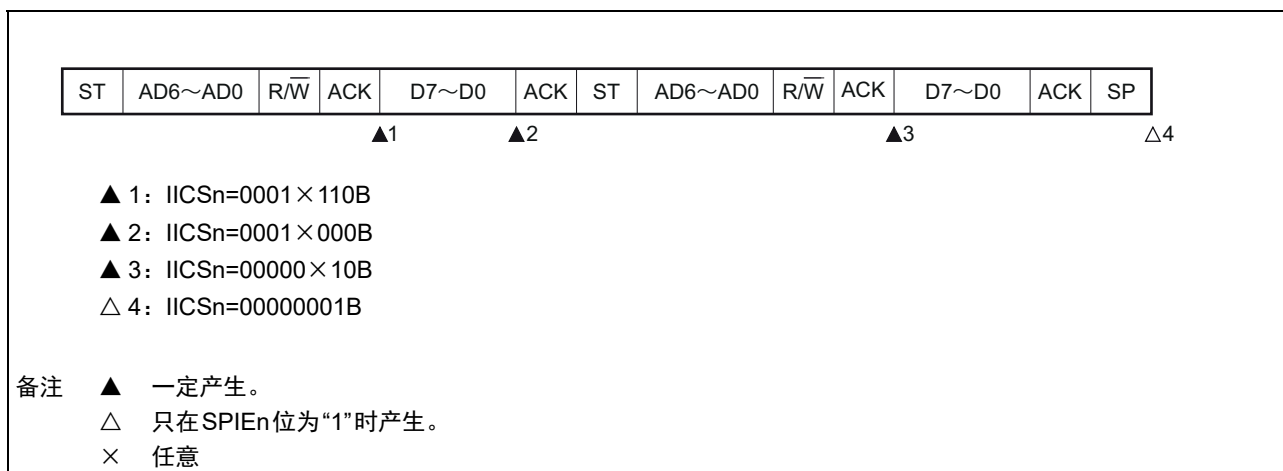
(ii) WTIMn=1的情况 (在重新开始后地址不同 (扩展码))



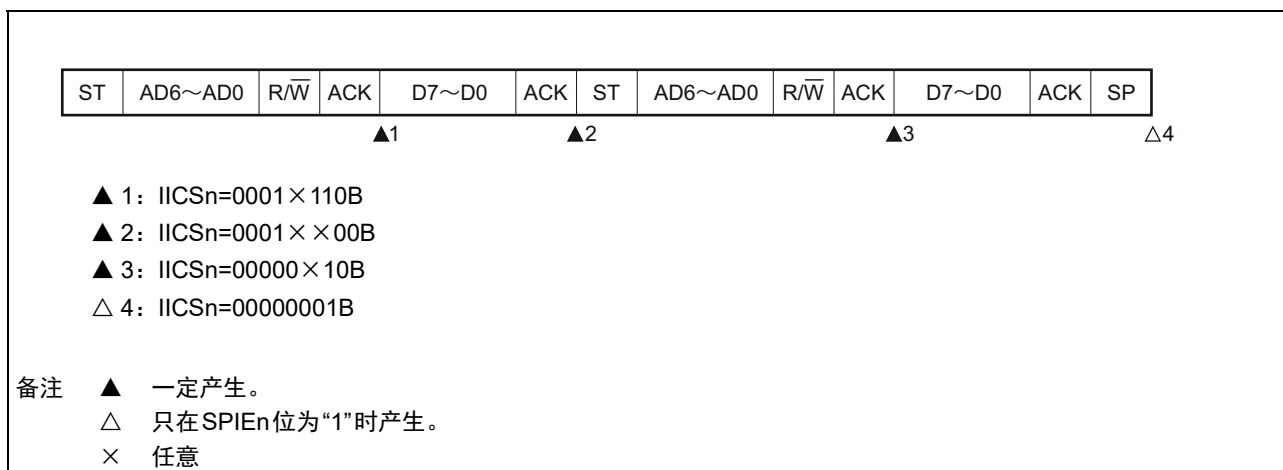
备注 n=0

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0的情况 (在重新开始后地址不同 (非扩展码))



(ii) WTIMn=1的情况 (在重新开始后地址不同 (非扩展码))



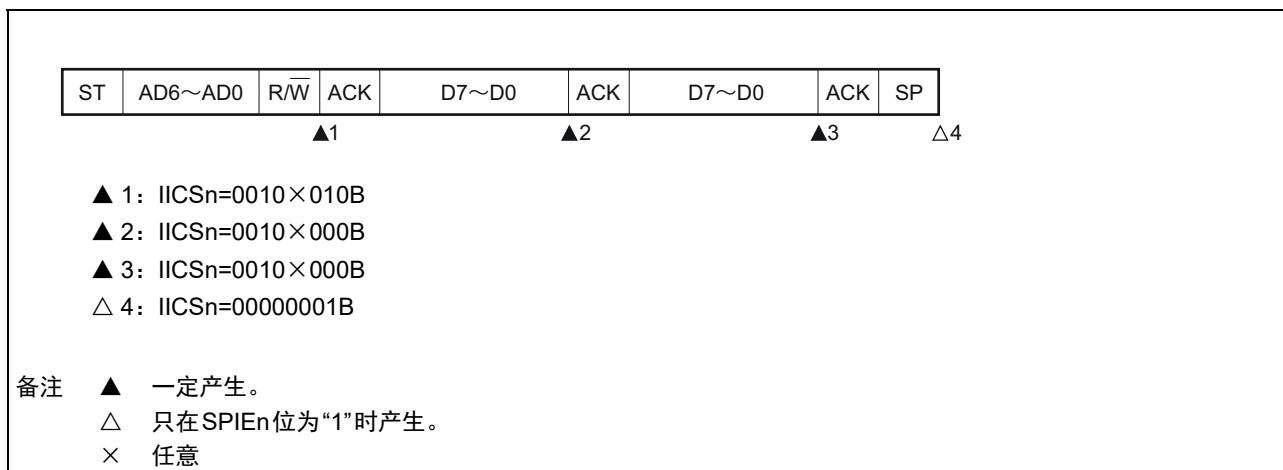
备注 n=0

(3) 从属运行（接收扩展码的情况）

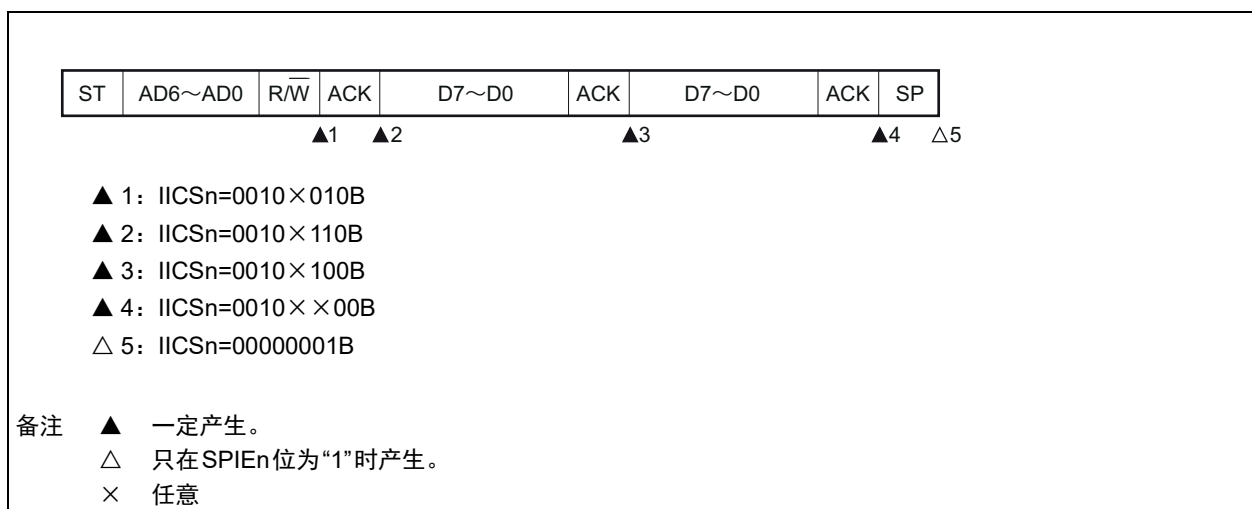
在接收扩展码时，始终参加通信。

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) WTIMn=0的情况



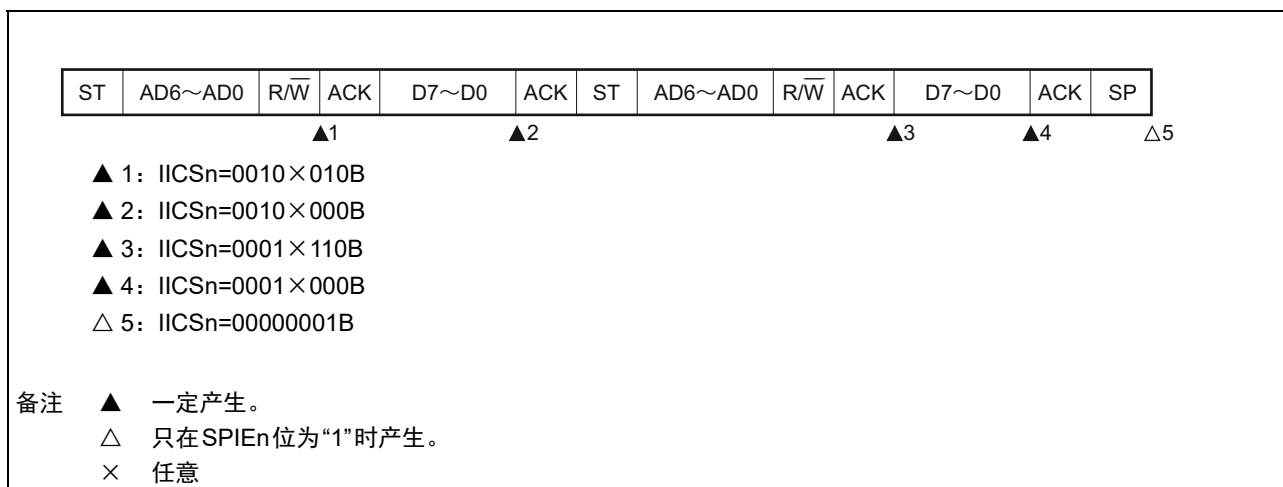
(ii) WTIMn=1的情况



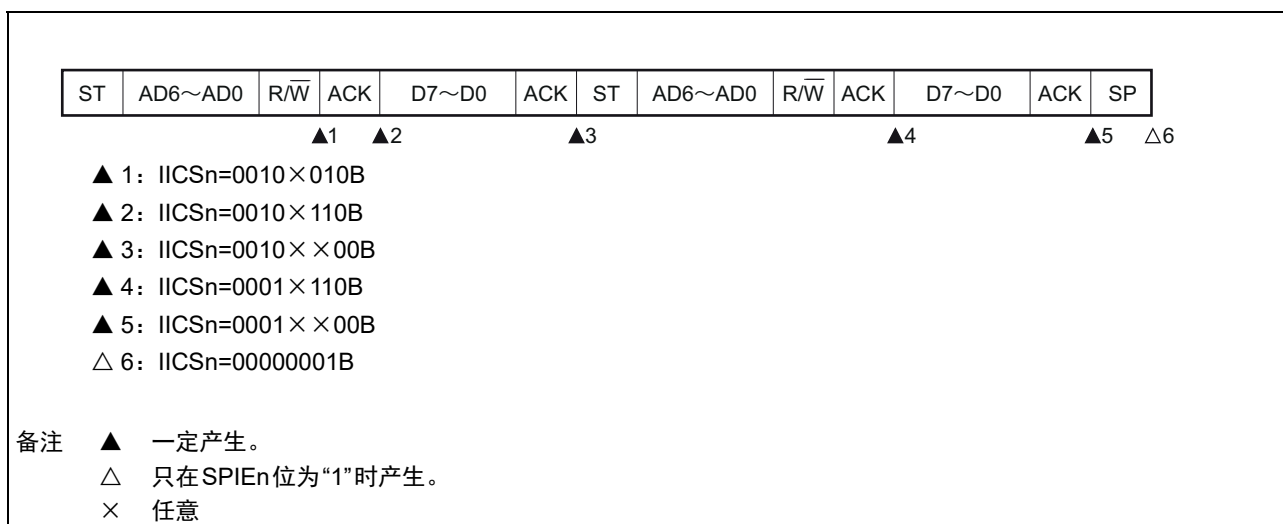
备注 n=0

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0的情况 (在重新开始后SVAn相同)



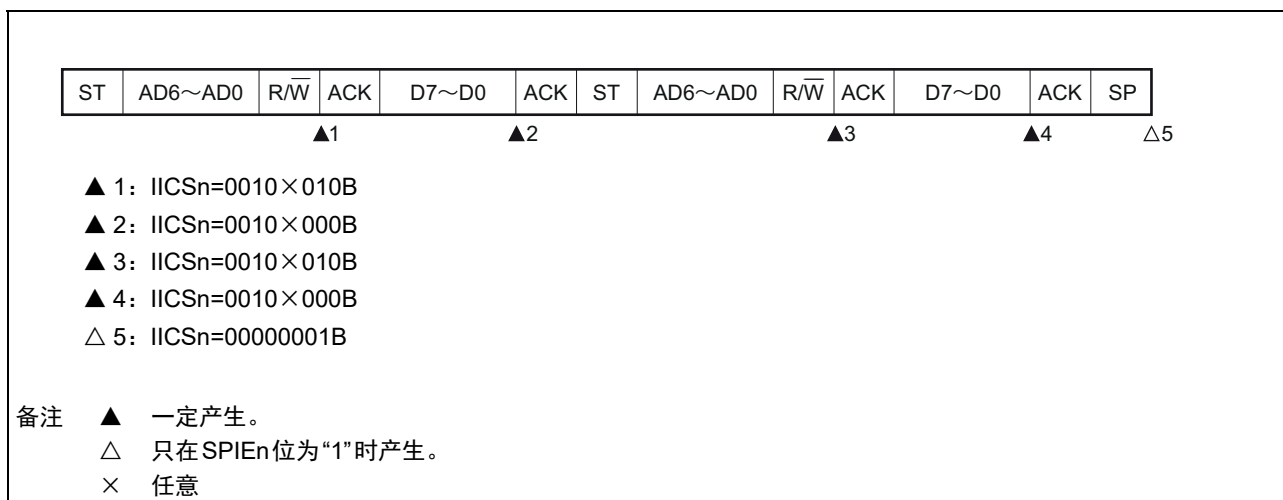
(ii) WTIMn=1的情况 (在重新开始后SVAn相同)



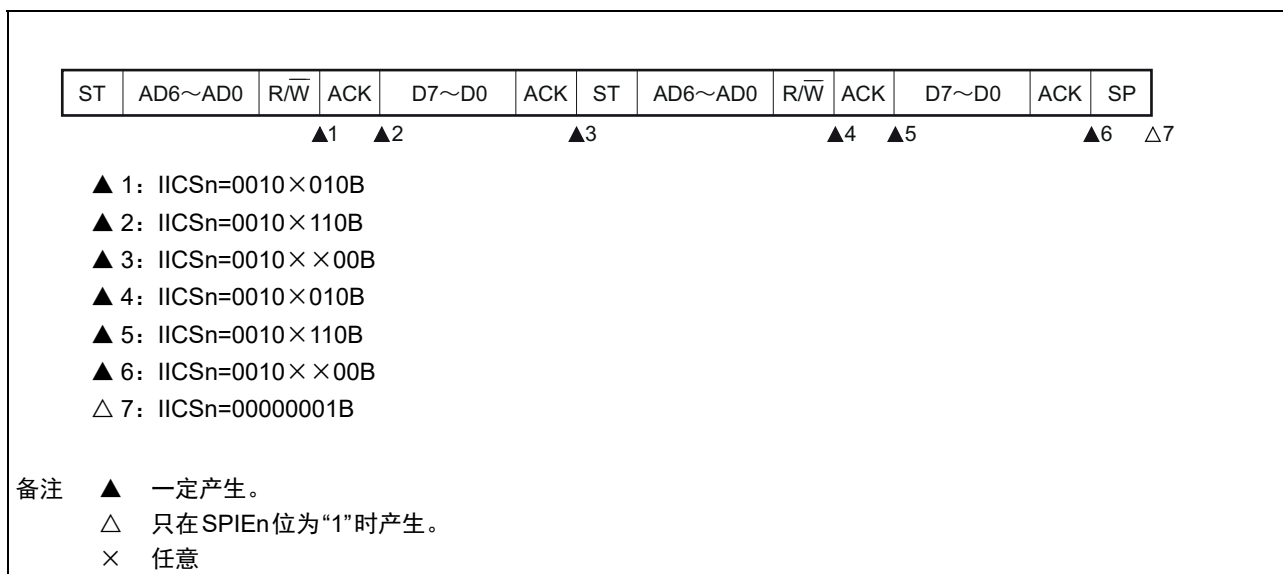
备注 n=0

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) WTIMn=0的情况（在重新开始后接收扩展码）



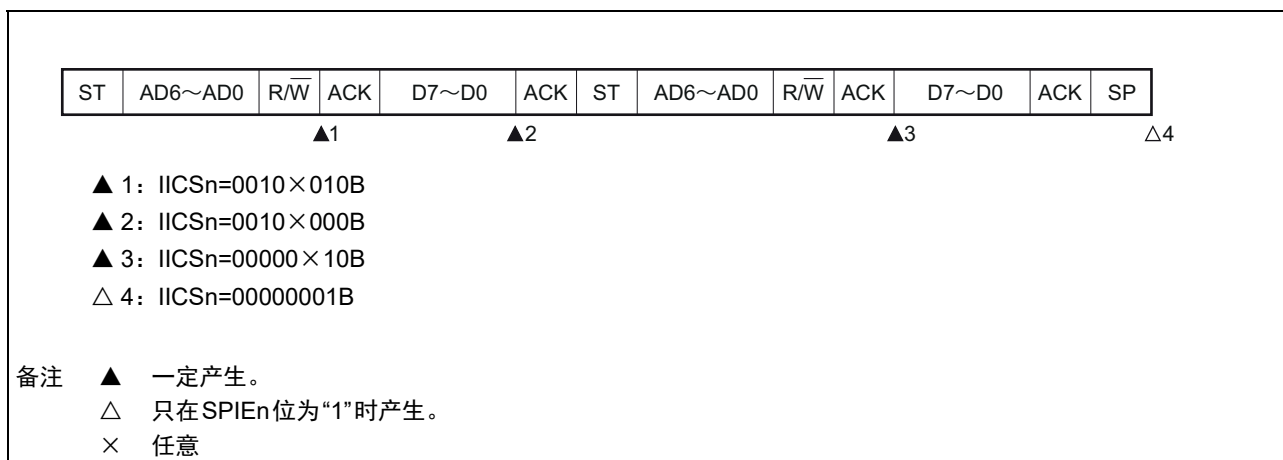
(ii) WTIMn=1的情况（在重新开始后接收扩展码）



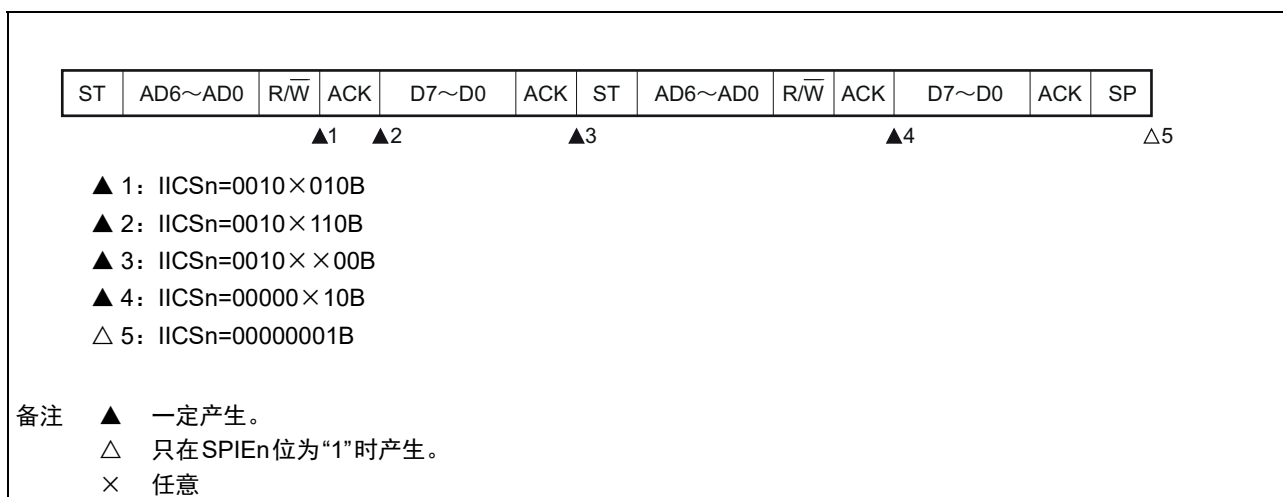
备注 n=0

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) WTIMn=0的情况 (在重新开始后地址不同 (非扩展码))



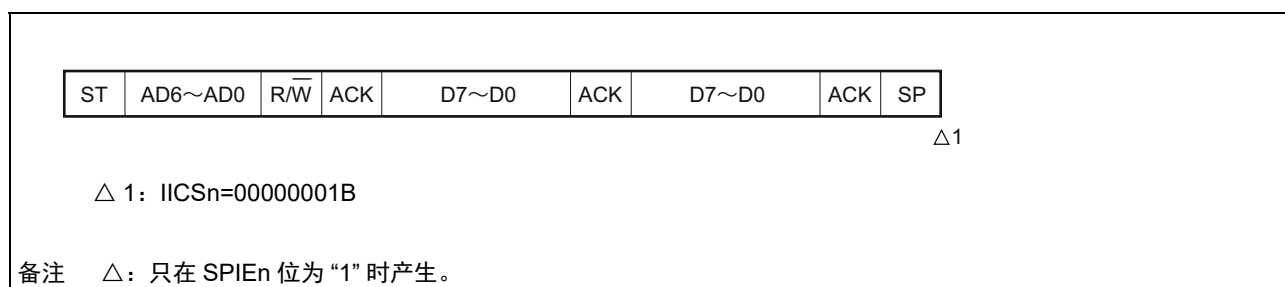
(ii) WTIMn=1的情况 (在重新开始后地址不同 (非扩展码))



备注 n=0

(4) 不参加通信的运行

(a) Start ~ Code ~ Data ~ Data ~ Stop

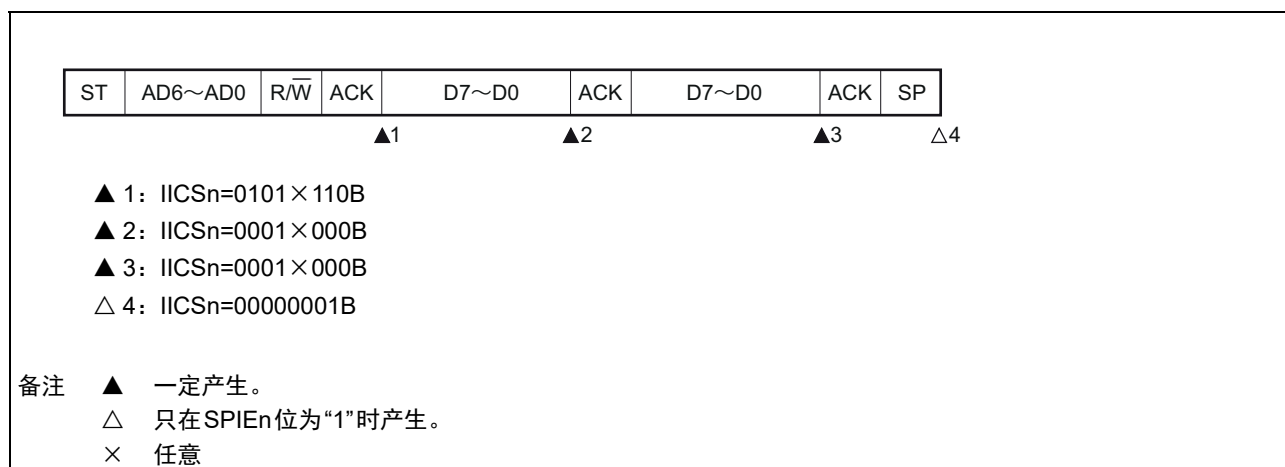


(5) 仲裁失败的运行（在仲裁失败后作为从属设备运行）

在多主控系统中用作主控设备时，必须在每次产生 INTIICAn 中断请求信号时读 MSTSn 位，确认仲裁结果。

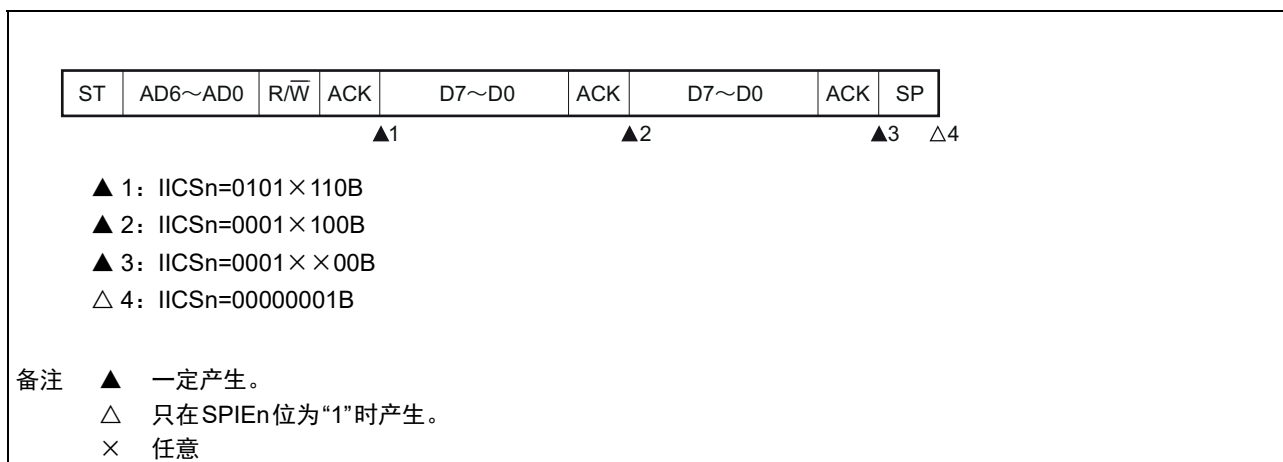
(a) 在发送从属地址数据的过程中仲裁失败的情况

(i) WTIMn=0 的情况



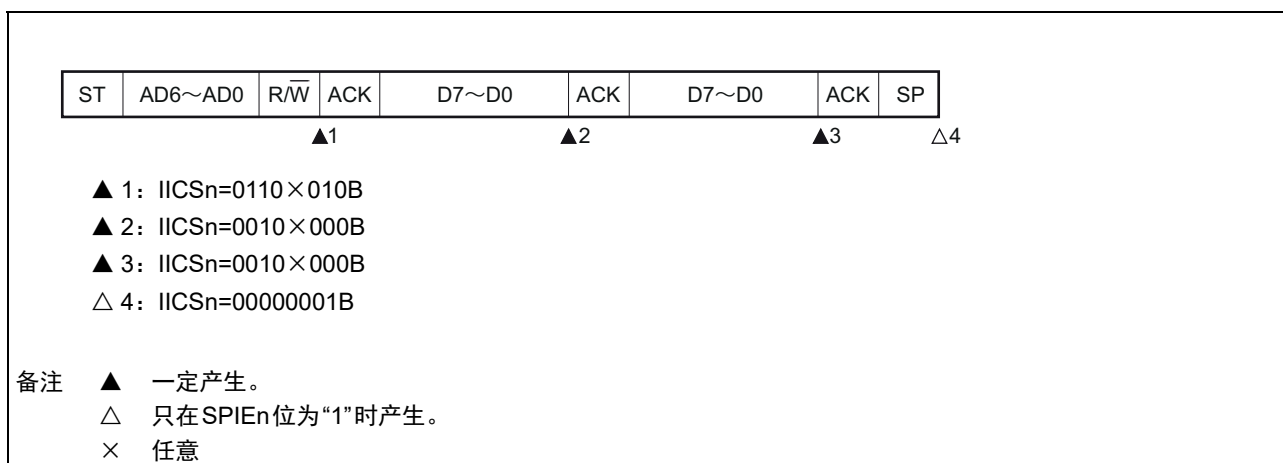
备注 n=0

(ii) WTIMn=1 的情况



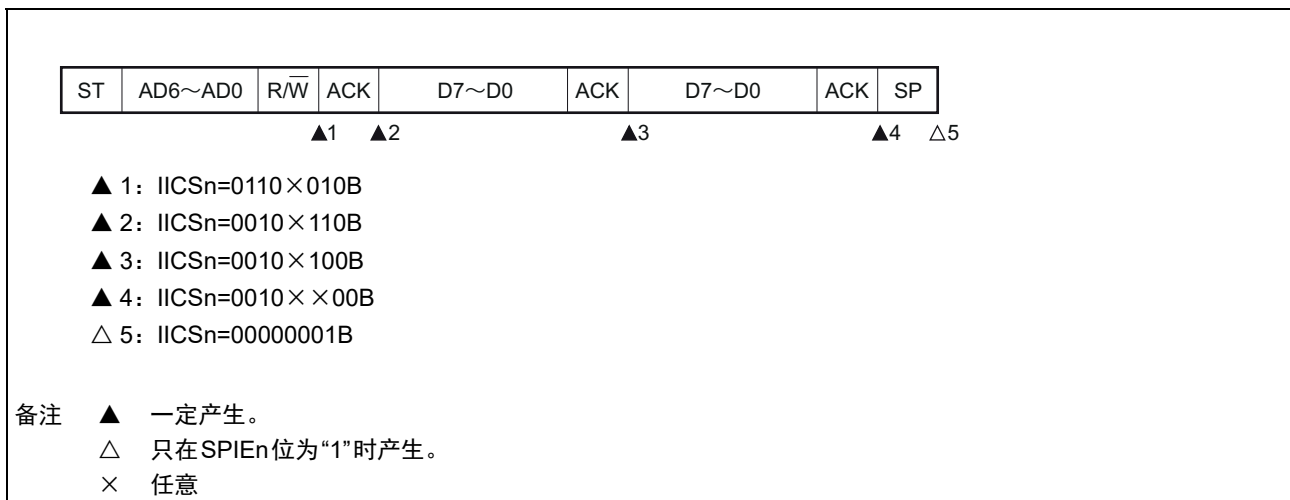
(b) 在发送扩展码的过程中仲裁失败的情况

(i) WTIMn=0 的情况



备注 n=0

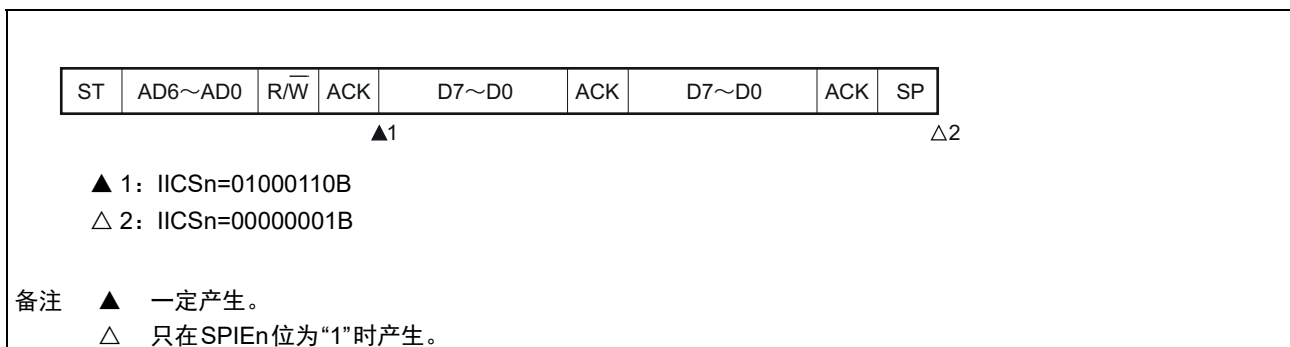
(ii) WTIMn=1 的情况



(6) 仲裁失败的运行（在仲裁失败后不参加通信）

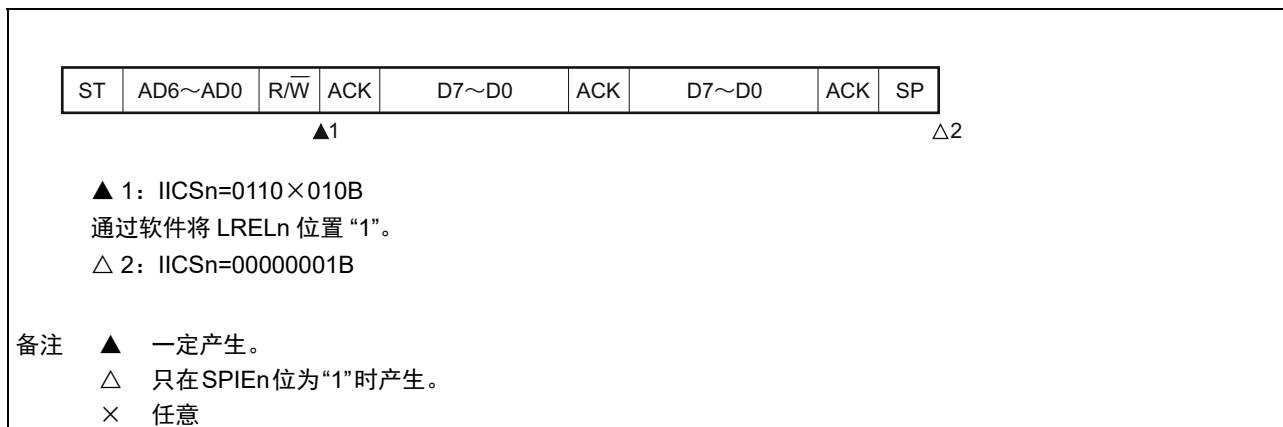
在多主控系统中用作主控设备时，必须在每次产生INTIICAn中断请求信号时读MSTSn位，确认仲裁结果。

(a) 在发送从属地址数据的过程中仲裁失败的情况（WTIMn=1）



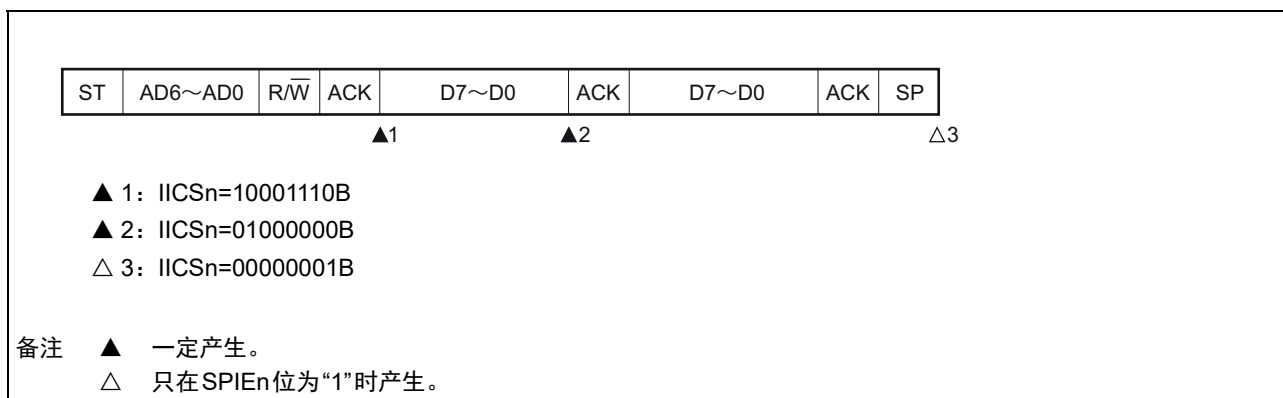
备注 n=0

(b) 在发送扩展码的过程中仲裁失败的情况



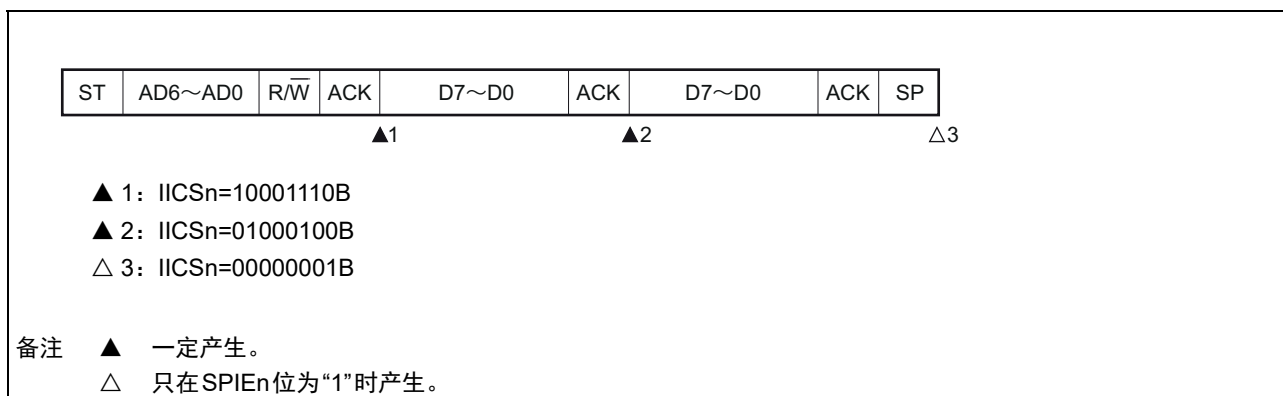
(c) 在传送数据时仲裁失败的情况

(i) WTIMn=0 的情况



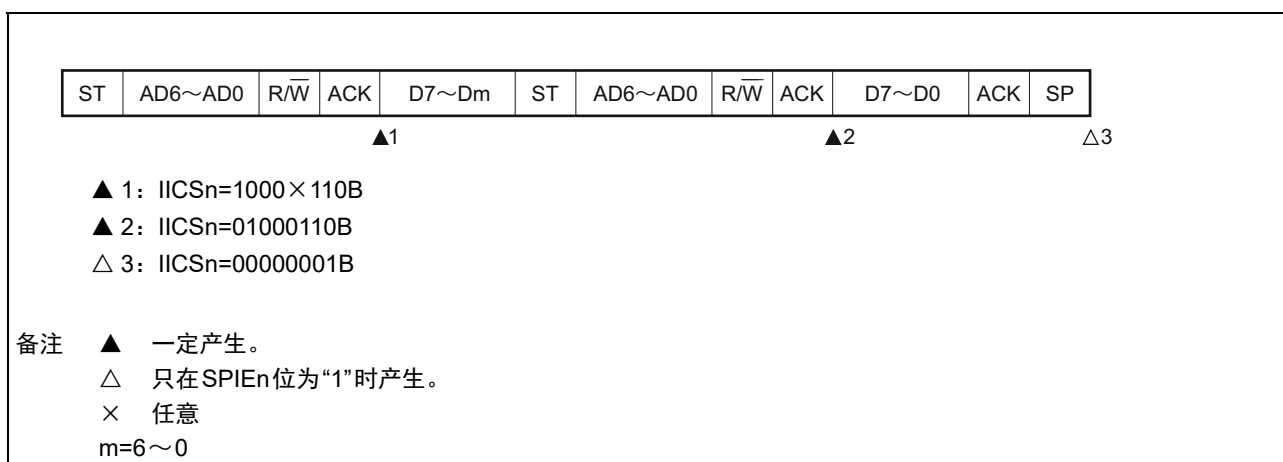
备注 n=0

(ii) WTIMn=1 的情况



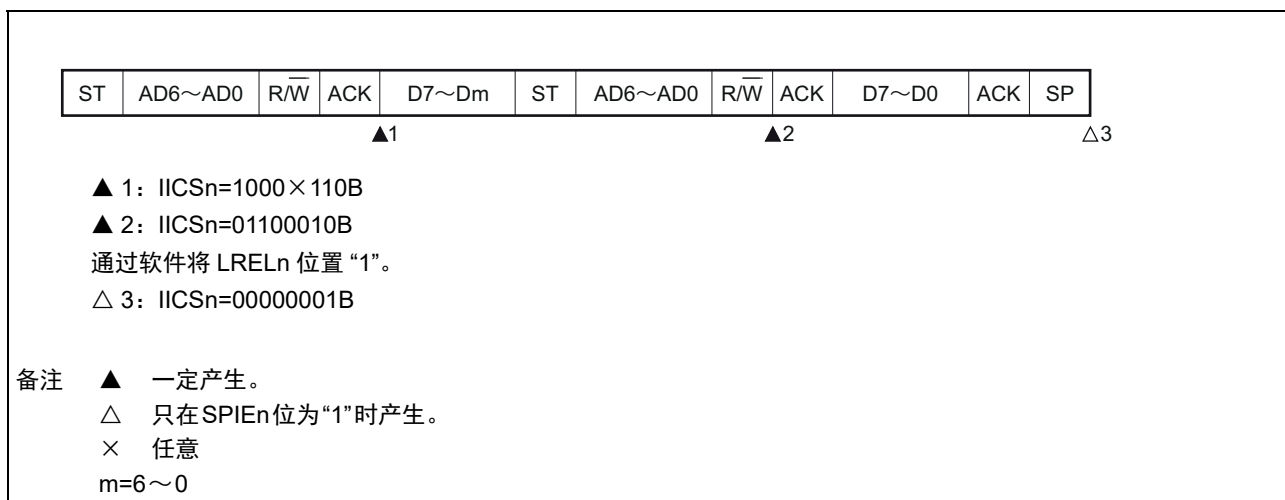
(d) 在传送数据时因重新开始条件而仲裁失败的情况

(i) 非扩展码 (例如, SVAn 不同)

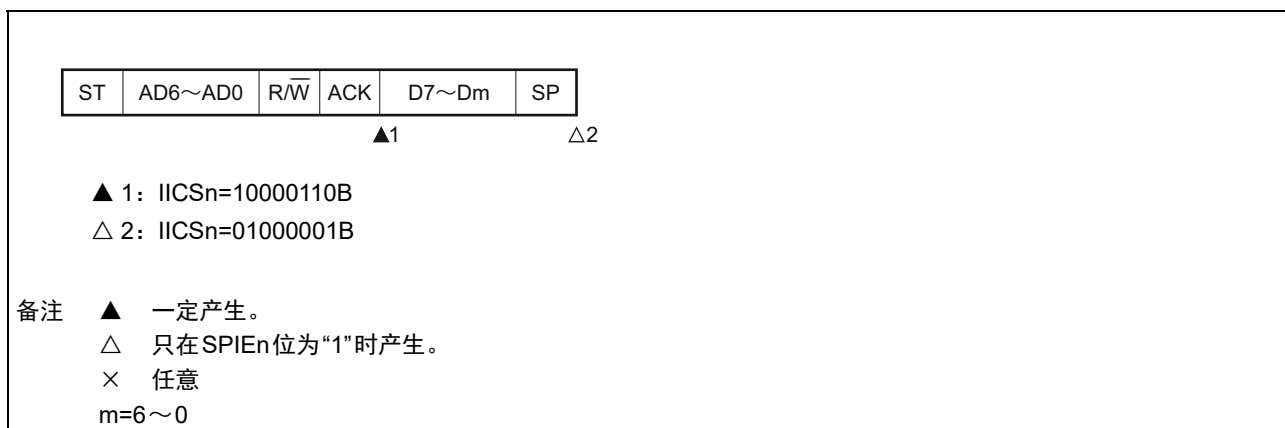


备注 n=0

(ii) 扩展码



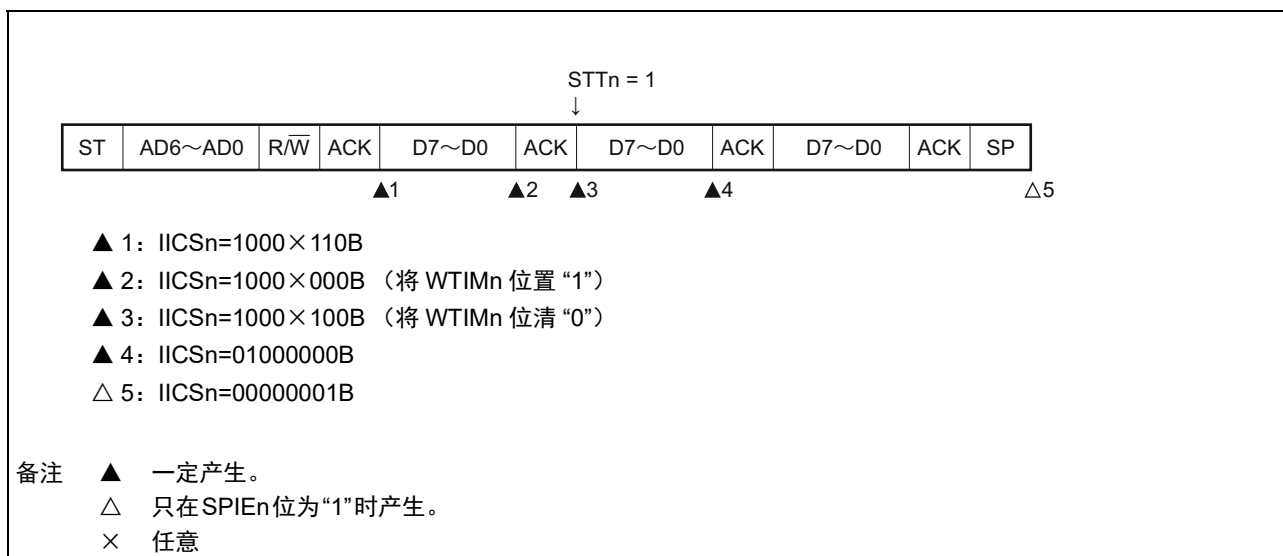
(e) 在传送数据时因停止条件而仲裁失败的情况



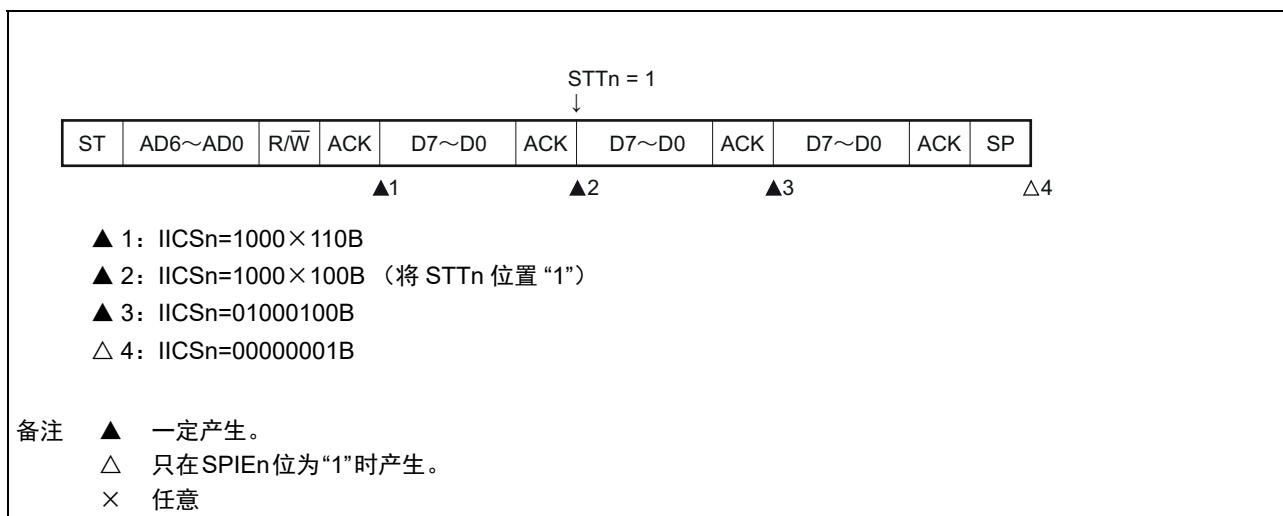
备注 n=0

(f) 在想要生成重新开始条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



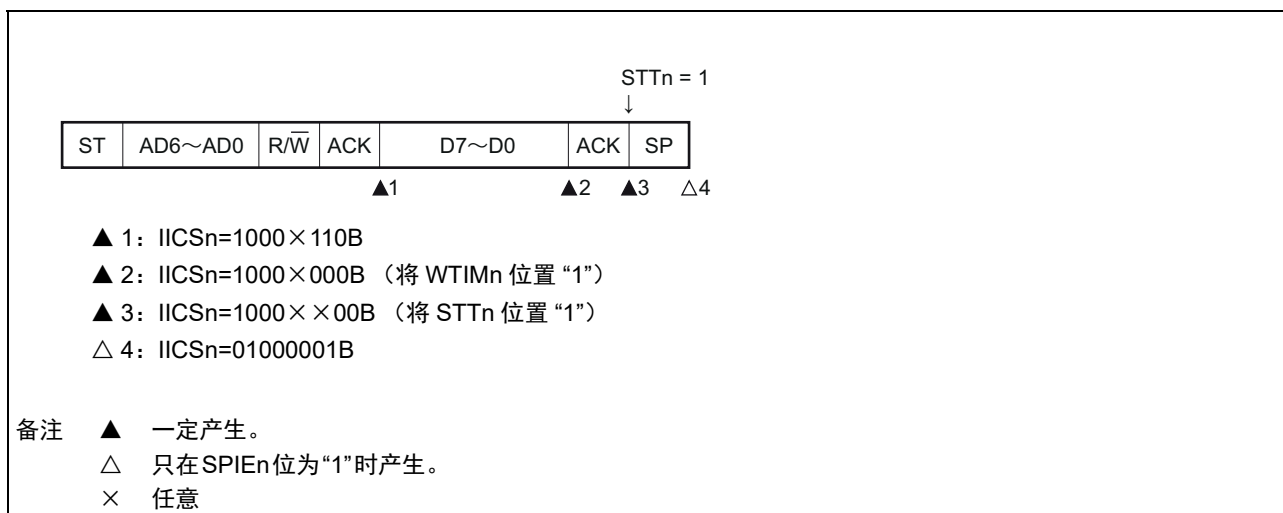
(ii) WTIMn=1的情况



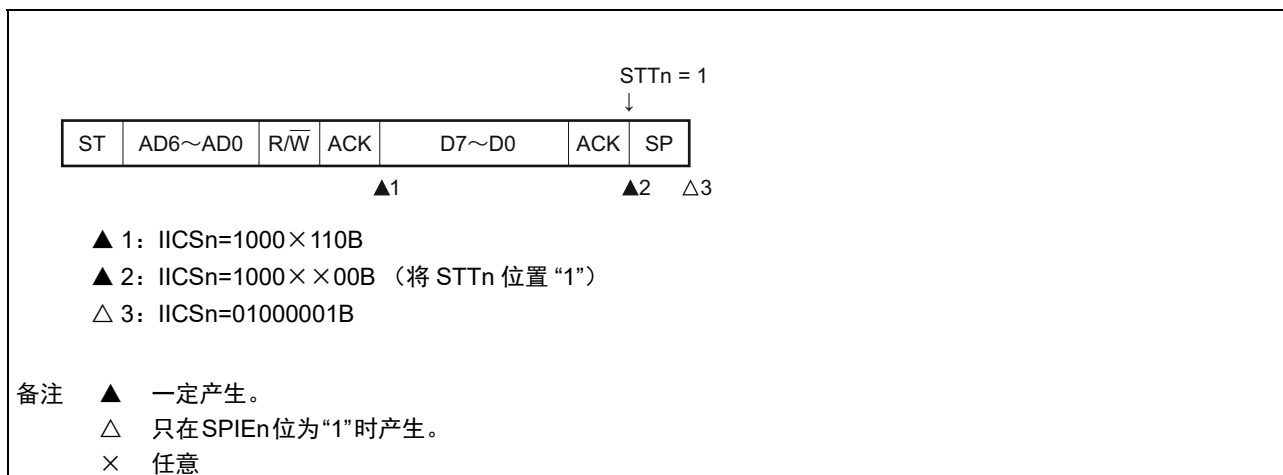
备注 n=0

(g) 在想要生成重新开始条件时因停止条件而仲裁失败的情况

(i) WTIMn=0的情况



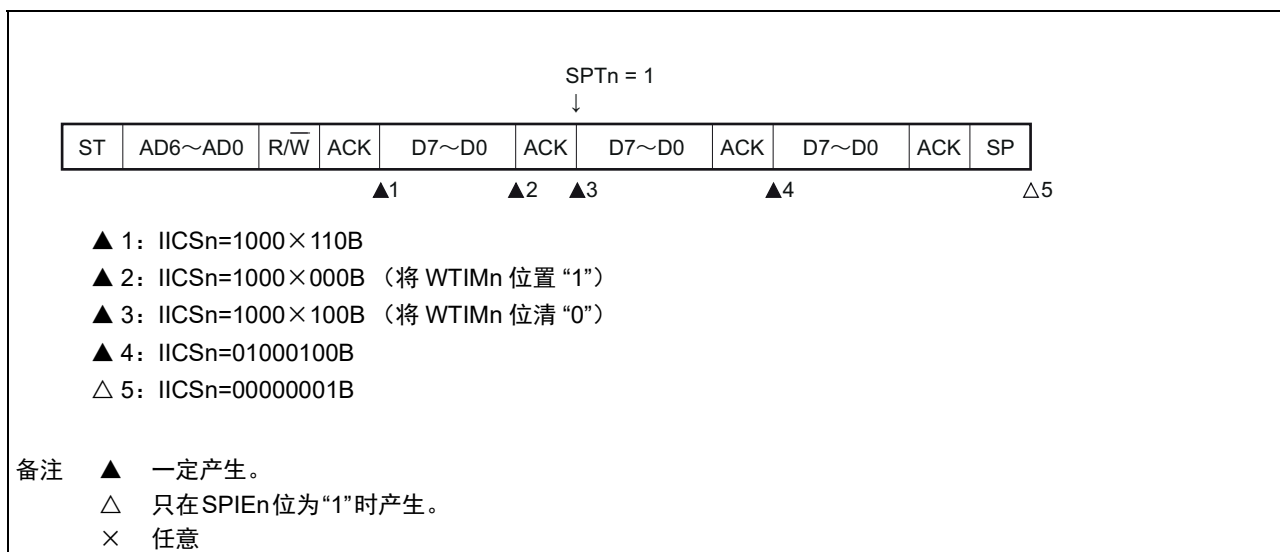
(ii) WTIMn=1的情况



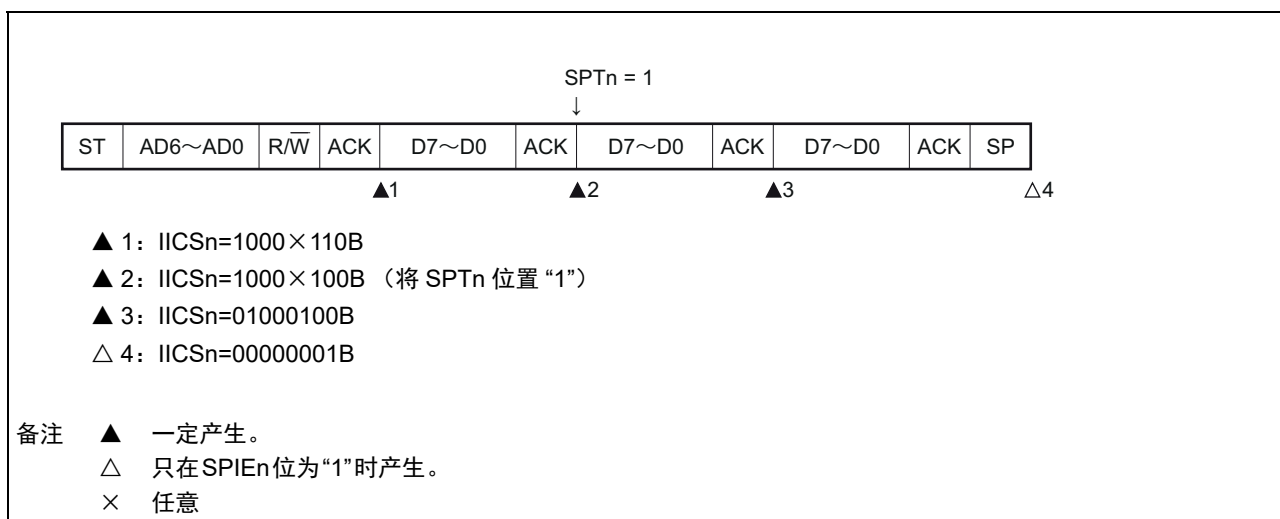
备注 n=0

(h) 在想要生成停止条件时因数据为低电平而仲裁失败的情况

(i) WTIMn=0的情况



(ii) WTIMn=1的情况



备注 n=0

12.6 时序图

在 I²C 总线模式中，主控设备通过给串行总线输出地址，从多个从属设备中选择一个通信对象的从属设备。

主控设备在从属设备地址之后发送表示数据传送方向的 TRCn 位（IICA 状态寄存器 n（IICSn）的 bit3），开始与从属设备进行串行通信。

数据通信的时序图如图 12-32 和图 12-33 所示。

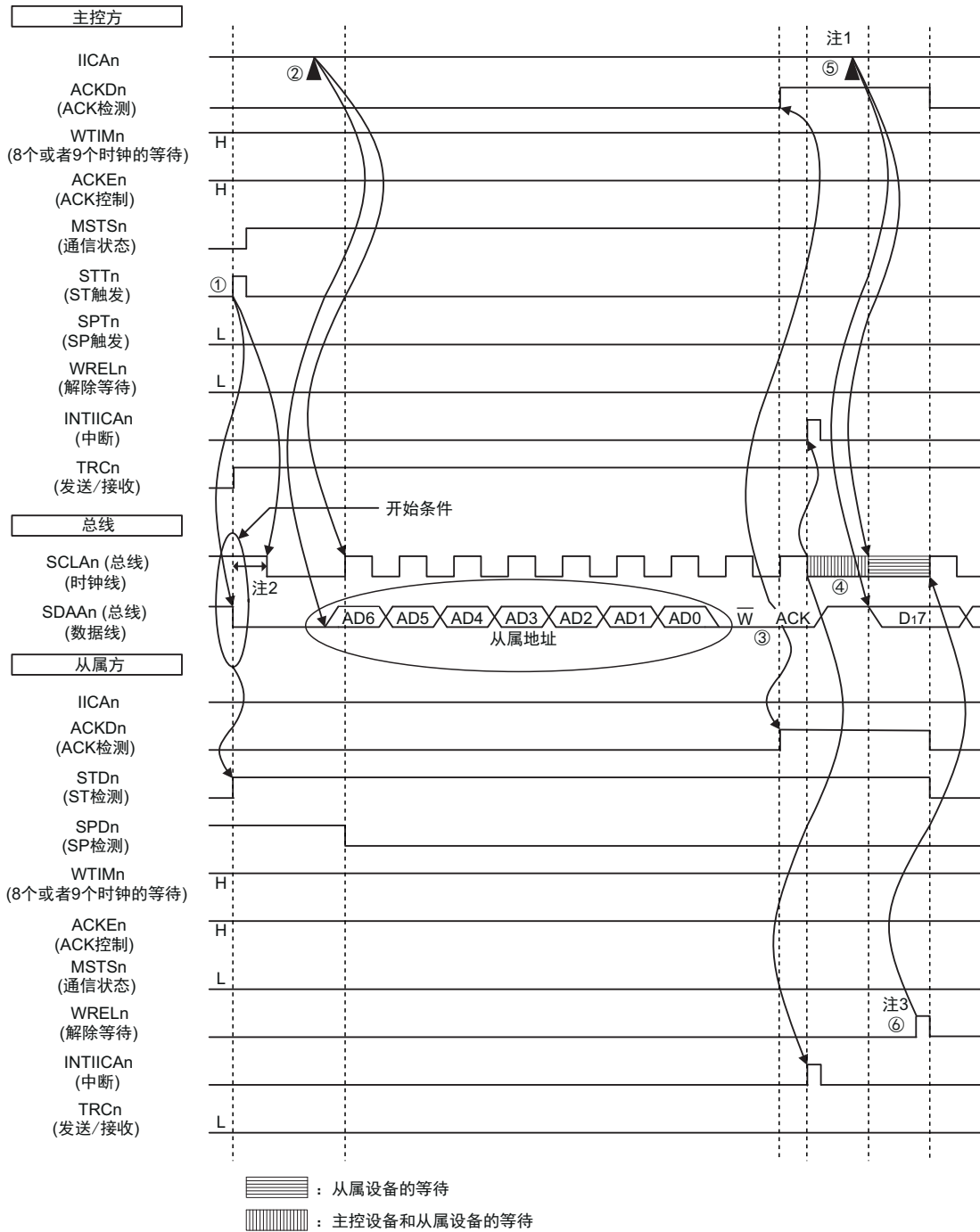
与串行时钟（SCLAn）的下降沿同步进行 IICA 移位寄存器 n（IICAn）的移位，并且将发送数据传送到 SO 锁存器，以 MSB 优先从 SDAAn 引脚输出数据。

在 SCLAn 的上升沿将 SDAAn 引脚输入的数据取到 IICAn。

备注 n=0

图 12-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (1/4)

(1) 开始条件~地址~数据



- 注 1. 要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
3. 要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

图 12-32 的“(1) 开始条件~地址~数据”的①~⑥的说明如下：

- ① 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTSn=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 n（IICAn）写地址+W（发送），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤ 主控方给 IICAn 寄存器写发送数据，解除主控方的等待。
- ⑥ 如果从属方解除等待（WRELn=1），主控方就开始给从属方传送数据。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。
但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

备注 1. 图 12-32 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件~地址~数据”说明步骤①~④。

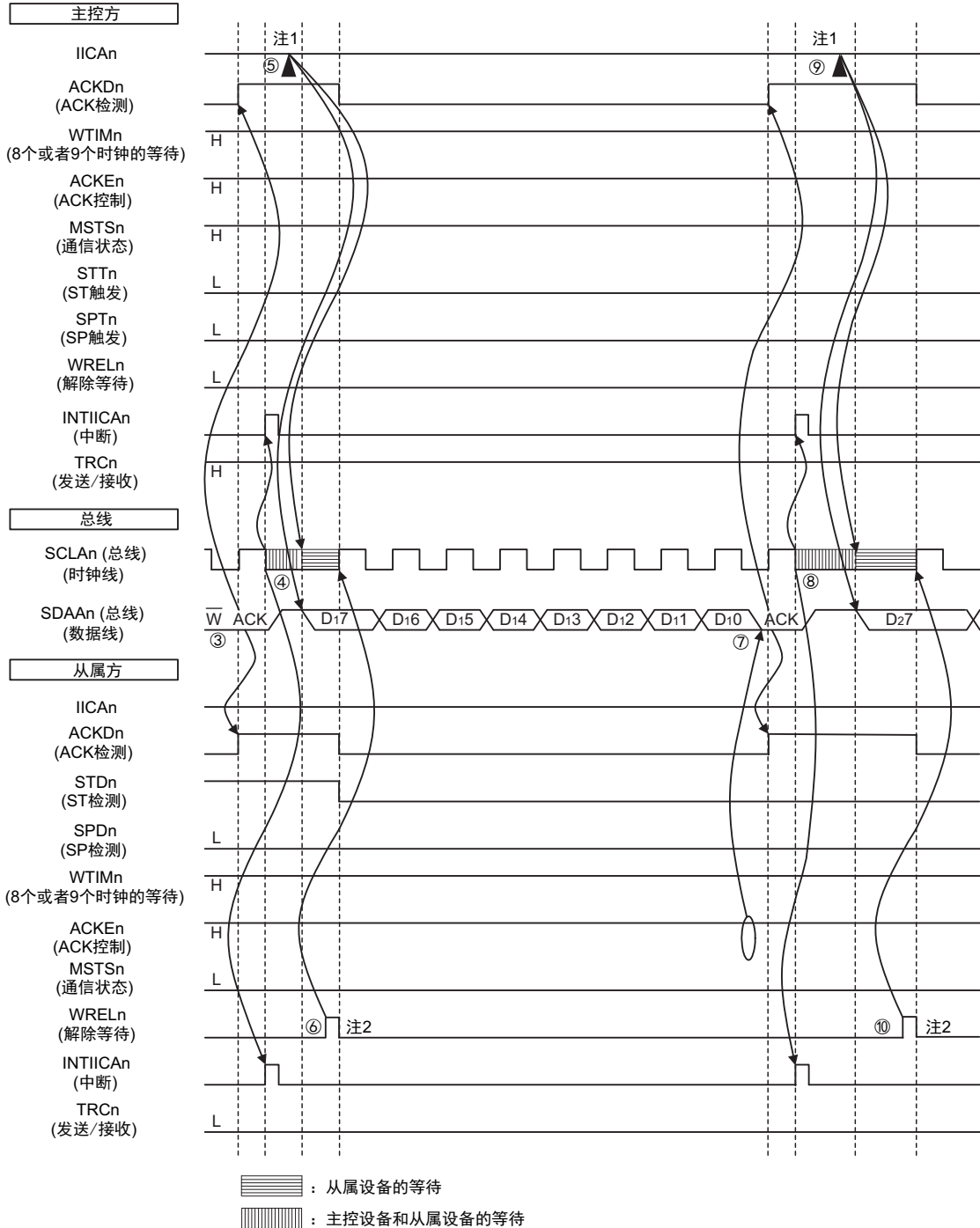
图 12-32 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 12-32 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

2. n=0

图 12-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/4)

(2) 地址~数据~数据



- 注 1. 要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。
- 2. 要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

图 12-32 的“(2) 地址~数据~数据”的③~⑩的说明如下:

- ③ 在从属方, 如果接收地址和本地站地址 (SVAn 的值) 相同注, 就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ④ 主控方在第 9 个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断)注。
- ⑤ 主控方给 IICA 移位寄存器 n (IICAn) 写发送数据, 解除主控方的等待。
- ⑥ 如果从属方解除等待 (WRELn=1), 主控方就开始给从属方传送数据。
- ⑦ 在数据传送结束后, 因为从属方的 ACKEn 位为“1”, 所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑨ 主控方给 IICAn 寄存器写发送数据, 解除主控方的等待。
- ⑩ 如果从属方读接收数据并且解除等待 (WRELn=1), 主控方就开始给从属方传送数据。

注 如果发送的地址和从属地址不同, 从属方就不给主控方返回 ACK (NACK: SDAAn=1), 并且不产生 INTIICAn 中断 (地址匹配中断), 也不进入等待状态。
但是, 主控方对于 ACK 和 NACK 都产生 INTIICAn 中断 (地址发送结束中断)。

备注 1. 图 12-32 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件~地址~数据”说明步骤①~④。

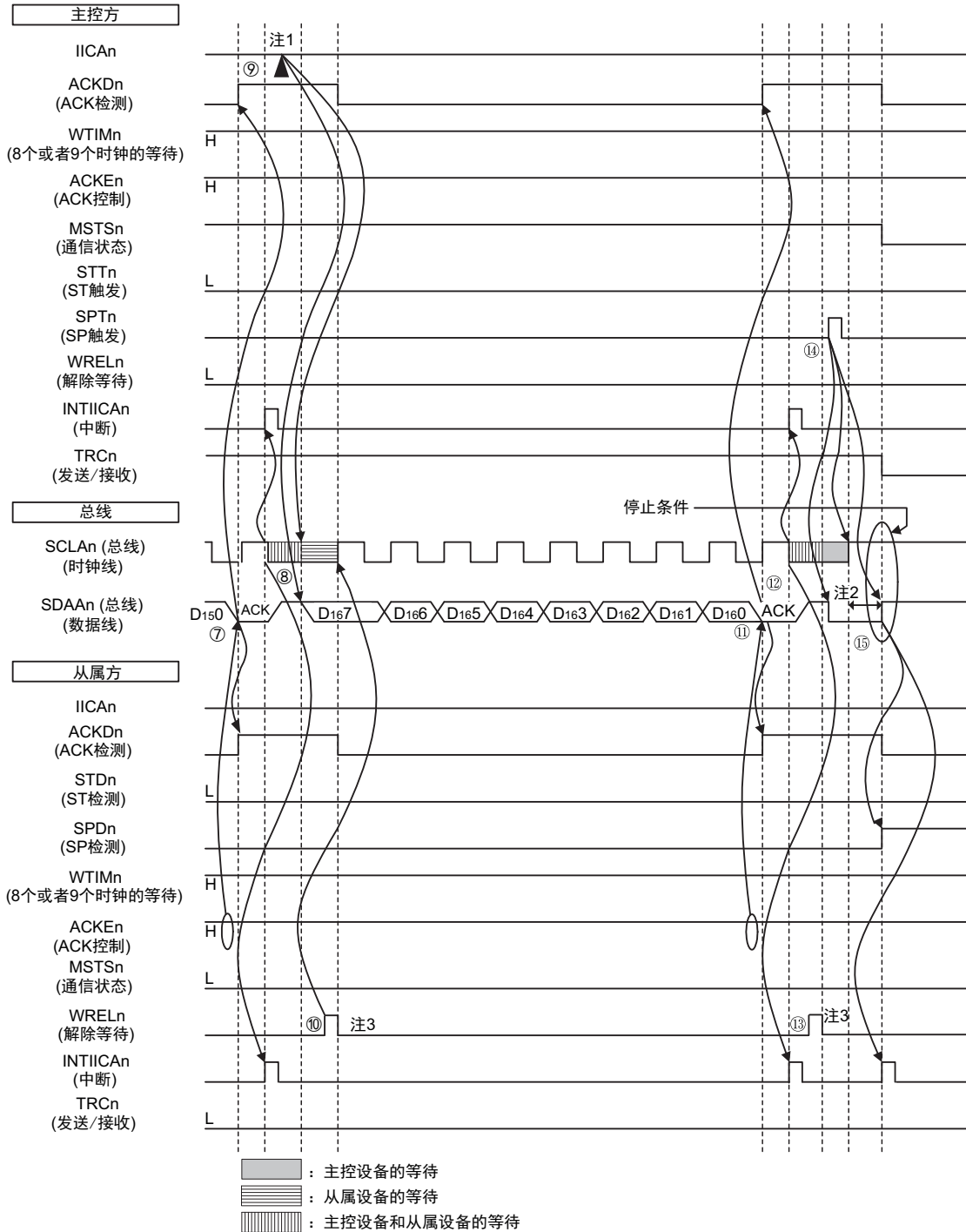
图 12-32 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 12-32 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

2. n=0

图 12-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (3/4)

(3) 数据~数据~停止条件



- 注 1. 要解除主控方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。
 2. 在发行停止条件后，从 SCLAn 引脚信号上升到生成停止条件的的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
 3. 要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

图 12-32 的“(3) 数据~数据~停止条件”的⑦~⑮的说明如下:

- ⑦ 在数据传送结束后, 因为从属方的 ACKEn 位为“1”, 所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑨ 主控方给 IICA 移位寄存器 n (IICAn) 写发送数据, 解除主控方的等待。
- ⑩ 如果从属方读接收数据并且解除等待 (WRELn=1), 主控方就开始给从属方传送数据。
- ⑪ 在数据传送结束后, 从属方 (ACKEn=1) 通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑫ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑬ 从属方读接收数据, 解除等待 (WRELn=1)。
- ⑭ 如果在主控方将停止条件触发置位 (SPTn=1), 就清除总线数据线 (SDAAn=0) 并且将总线时钟线置位 (SCLAn=1), 在经过停止条件的准备时间后将总线数据线置位 (SDAAn=1), 生成停止条件 (通过 SCLAn=1 使 SDAAn 从“0”变为“1”)。
- ⑮ 如果生成停止条件, 从属方就检测到停止条件并且产生中断 (INTIICAn: 停止条件中断)。

备注 1. 图 12-32 的①~⑮是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-32 的“(1) 开始条件~地址~数据”说明步骤①~④。

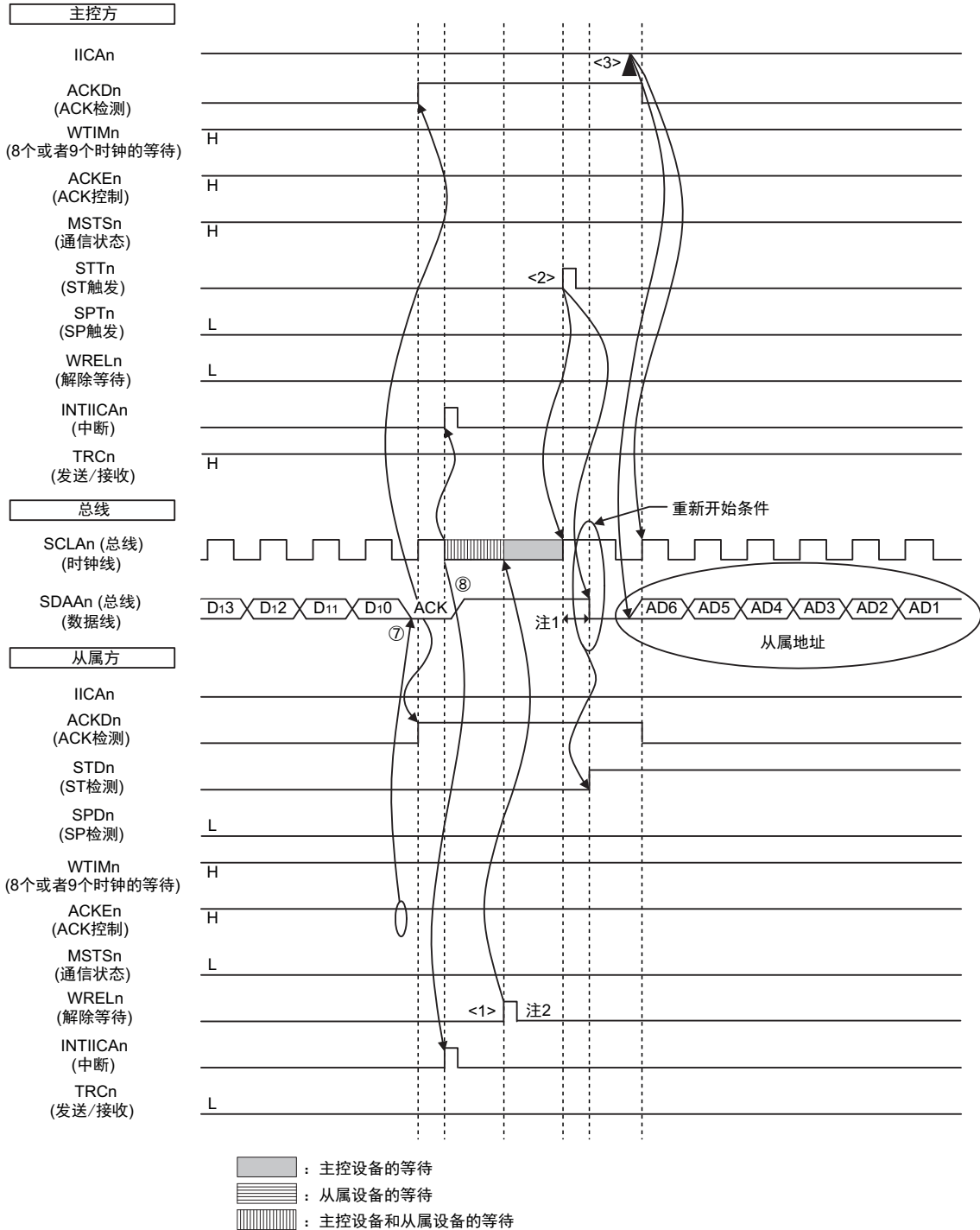
图 12-32 的“(2) 地址~数据~数据”说明步骤③~⑩。

图 12-32 的“(3) 数据~数据~停止条件”说明步骤⑦~⑮。

2. n=0

图 12-32 主控设备 → 从属设备的通信例子
 (主控设备：选择 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (4/4)

(4) 数据~重新开始条件~地址



- 注 1. 在发行重新开始条件后，从 SCLAn 引脚信号上升到生成开始条件的的时间，在设定为标准模式时至少为 4.7μs，在设定为快速模式时至少为 0.6μs。
2. 要解除从属方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。

备注 n=0

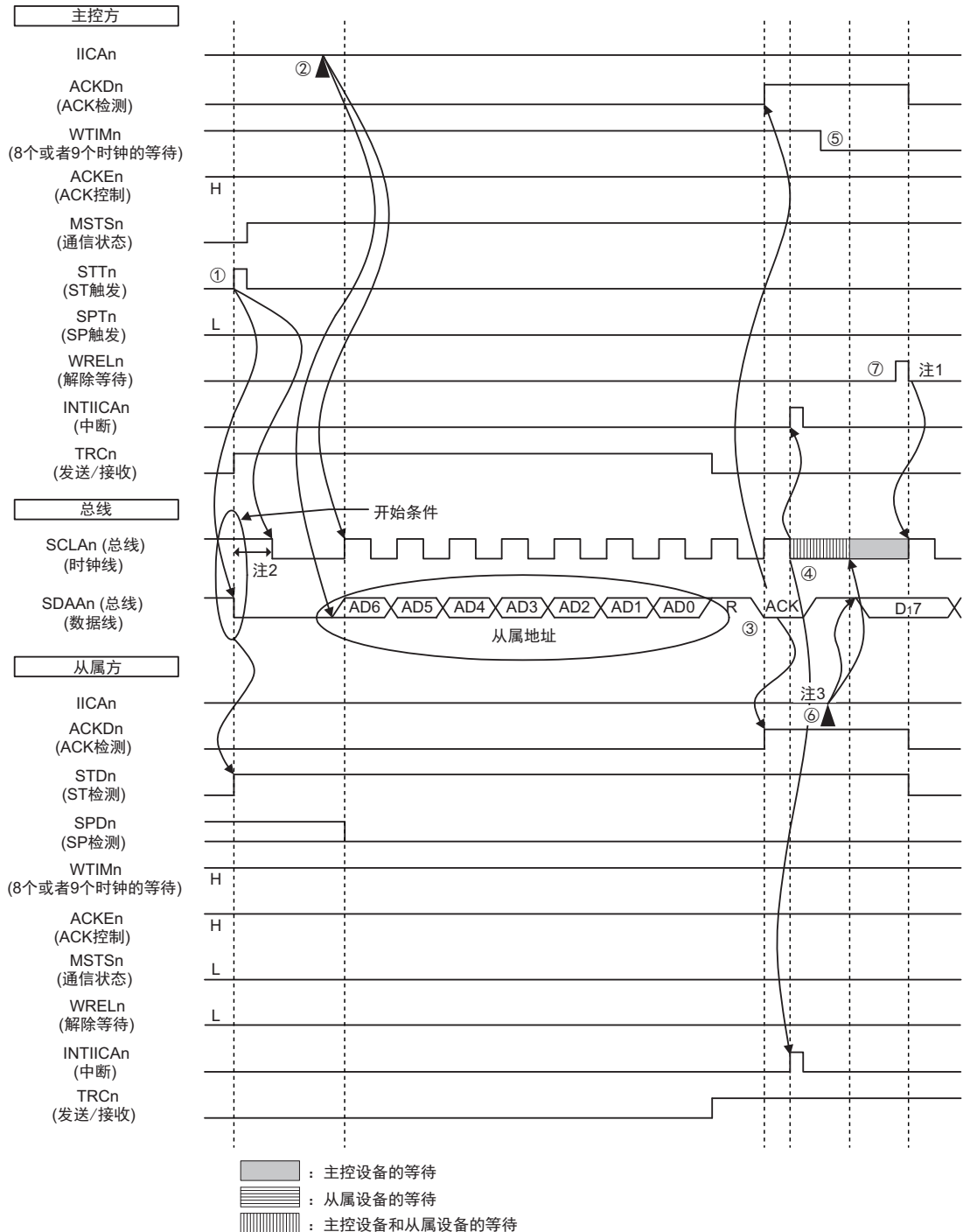
图 12-32 的“(4) 数据~重新开始条件~地址”的运行说明如下。在执行步骤⑦和⑧后执行<1>~<3>,从而返回到步骤③的数据发送步骤。

- ⑦ 在数据传送结束后,因为从属方的 ACKEn 位为“1”,所以通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑧ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0),并且都产生中断 (INTIICAn: 传送结束中断)。
 - <1> 从属方读接收数据,解除等待 (WRELn=1)。
 - <2> 如果在主控方再次将开始条件触发置位 (STTn=1),总线时钟线就上升 (SCLAn=1),而且在经过重新开始条件的准备时间后总线数据线下降 (SDAAn=0),生成开始条件 (通过 SCLAn=1 使 SDAAn 从“1”变为“0”)。然后,如果检测到开始条件,就在经过保持时间后总线时钟线下降 (SCLAn=0),结束通信准备。
 - <3> 如果主控方给 IICA 移位寄存器 n (IICAn) 写地址+R/W (发送),就发送从属地址。

备注 n=0

图 12-33 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待) (1/3)

(1) 开始条件~地址~数据



- 注 1. 要解除主控方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。
2. 从 SDAAn 引脚信号下降到 SCLAn 引脚信号下降的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
3. 要解除从属方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0

图 12-33 的“(1) 开始条件~地址~数据”的①~⑦的说明如下：

- ① 如果在主控方将开始条件触发置位（STTn=1），总线数据线（SDAAn）就下降，生成开始条件（通过 SCLAn=1 使 SDAAn 从“1”变为“0”）。此后，如果检测到开始条件，主控方就进入主控通信状态（MSTS_n=1），在经过保持时间后总线时钟线下降（SCLAn=0），结束通信准备。
- ② 如果主控方给 IICA 移位寄存器 n（IICAn）写地址+R（接收），就发送从属地址。
- ③ 在从属方，如果接收地址和本地站地址（SVAn 的值）相同注，就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK（ACKDn=1）。
- ④ 主控方在第 9 个时钟的下降沿产生中断（INTIICAn：地址发送结束中断）。相同地址的从属设备进入等待状态（SCLAn=0），并且产生中断（INTIICAn：地址匹配中断）注。
- ⑤ 主控方将等待时序改为第 8 个时钟（WTIMn=0）。
- ⑥ 从属方给 IICAn 寄存器写发送数据，解除从属方的等待。
- ⑦ 主控方解除等待（WRELn=1），开始来自从属设备的数据传送。

注 如果发送的地址和从属地址不同，从属方就不给主控方返回 ACK（NACK：SDAAn=1），并且不产生 INTIICAn 中断（地址匹配中断），也不进入等待状态。

但是，主控方对于 ACK 和 NACK 都产生 INTIICAn 中断（地址发送结束中断）。

备注 1. 图 12-33 的①~⑱是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

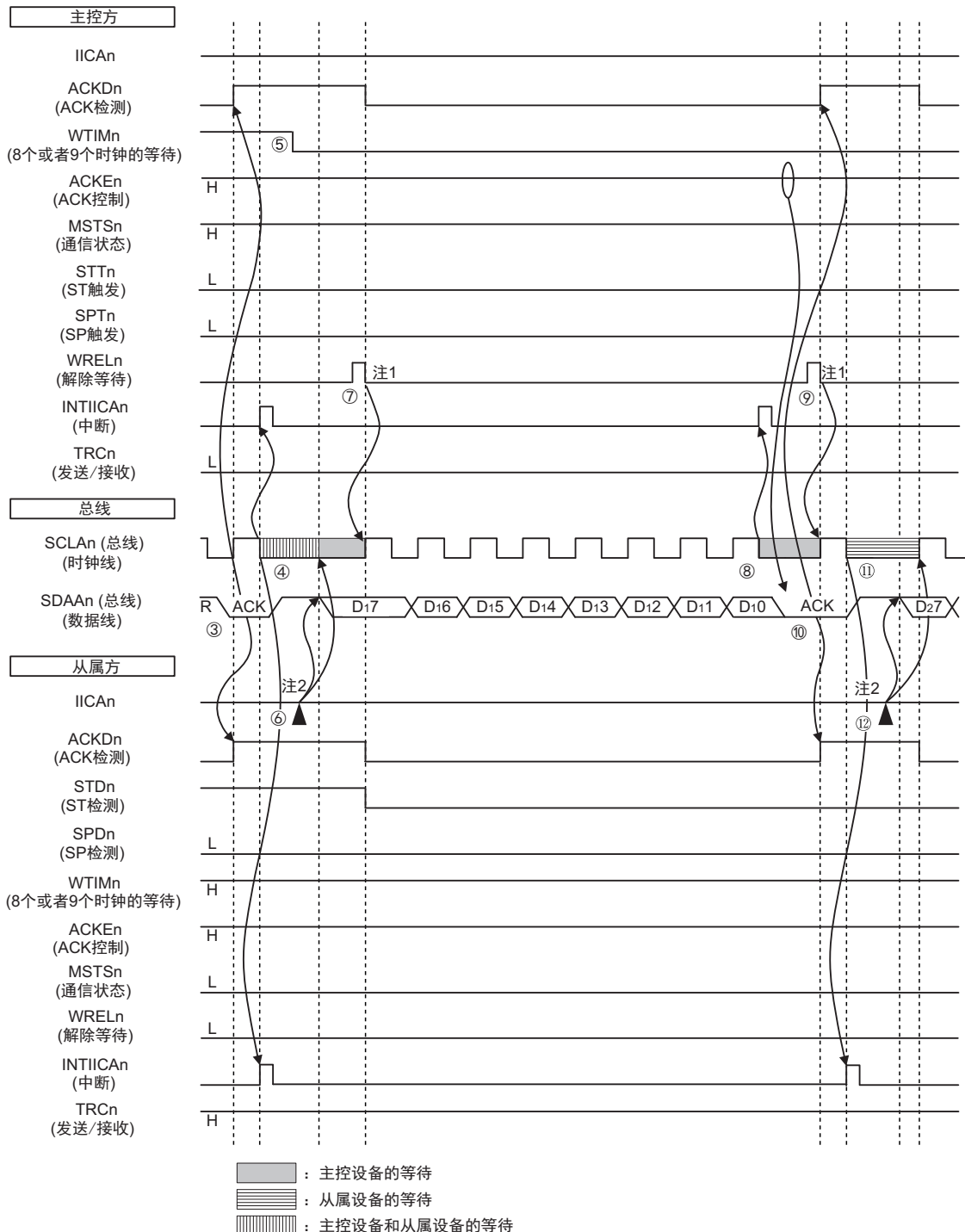
图 12-33 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 12-33 的“(3) 数据~数据~停止条件”说明步骤⑧~⑱。

2. n=0

图 12-33 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个时钟的等待，从属设备：选择 9 个时钟的等待) (2/3)

(2) 地址~数据~数据



- 注 1. 要解除主控方接收期间的等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。
- 2. 要解除从属方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。

备注 n=0

图 12-33 的“(2) 地址~数据~数据”的③~⑫的说明如下:

- ③ 在从属方, 如果接收地址和本地站地址 (SVAn 的值) 相同注, 就通过硬件给主控方发送 ACK。主控方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ④ 主控方在第 9 个时钟的下降沿产生中断 (INTIICAn: 地址发送结束中断)。相同地址的从属设备进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 地址匹配中断)注。
- ⑤ 主控方将等待时序改为第 8 个时钟 (WTIMn=0)。
- ⑥ 从属方给 IICA 移位寄存器 n (IICAn) 写发送数据, 解除从属方的等待。
- ⑦ 主控方解除等待 (WRELn=1), 开始来自从属设备的数据传送。
- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的 ACKEn 位为“1”, 所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据, 解除等待 (WRELn=1)。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0), 并且产生中断 (INTIICAn: 传送结束中断)。
- ⑫ 如果从属方给 IICAn 寄存器写发送数据, 就解除从属方的等待, 开始从属方到主控方的数据传送。

注 如果发送的地址和从属地址不同, 从属方就不给主控方返回 ACK (NACK: SDAAn=1), 并且不产生 INTIICAn 中断 (地址匹配中断), 也不进入等待状态。
但是, 主控方对于 ACK 和 NACK 都产生 INTIICAn 中断 (地址发送结束中断)。

备注 1. 图 12-33 的①~⑱是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

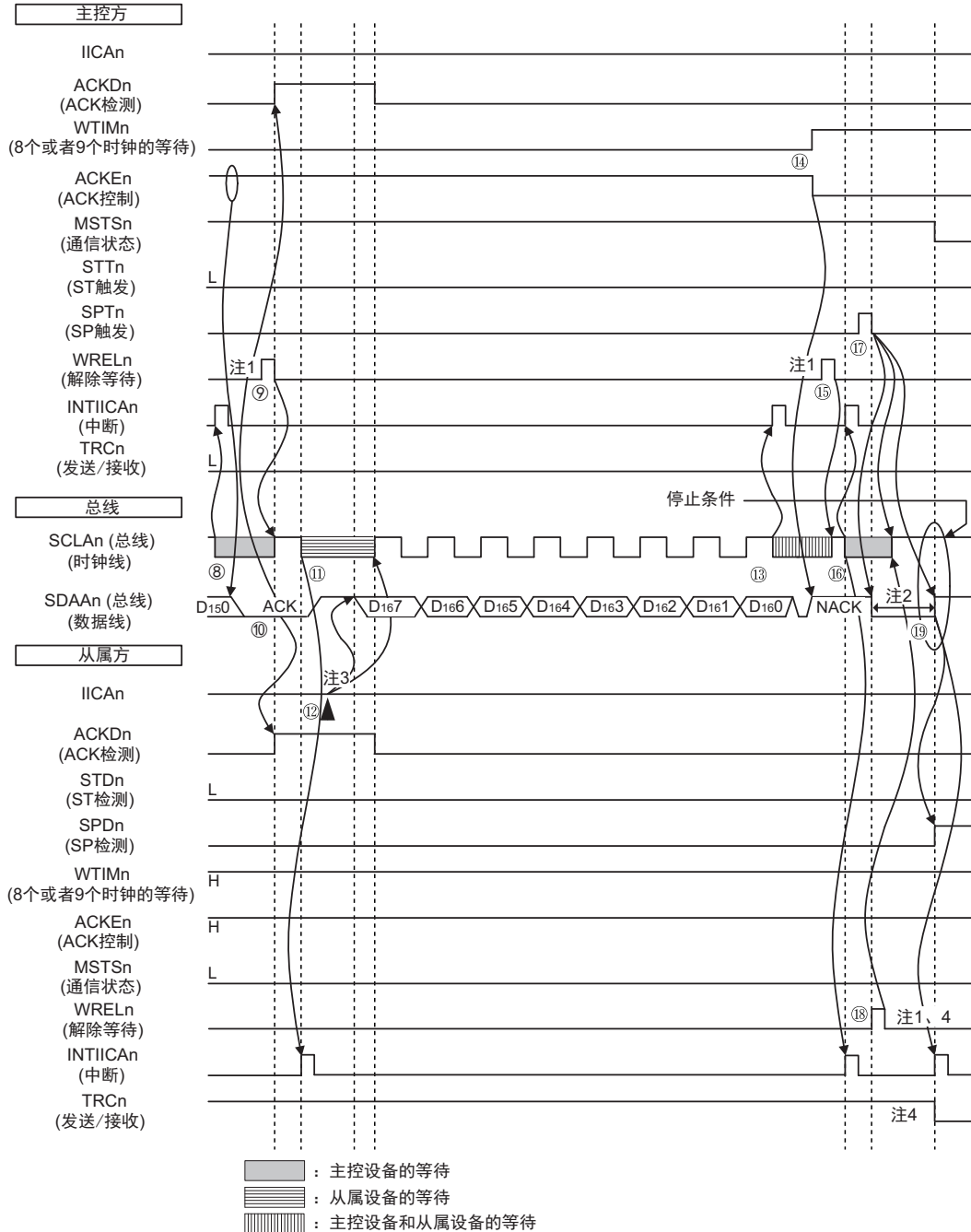
图 12-33 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 12-33 的“(3) 数据~数据~停止条件”说明步骤⑧~⑱。

2. n=0

图 12-33 从属设备 → 主控设备的通信例子
 (主控设备：选择 8 个 → 9 个时钟的等待，从属设备：选择 9 个时钟的等待) (3/3)

(3) 数据~数据~停止条件



- 注 1. 要解除等待时，必须将 IICAn 置“FFH”或者将 WRELn 位置位。
- 2. 在发行停止条件后，从 SCLAn 引脚信号上升到生成停止条件的的时间，在设定为标准模式时至少为 4.0μs，在设定为快速模式时至少为 0.6μs。
- 3. 要解除从属方发送期间的等待时，必须给 IICAn 写数据而不是将 WRELn 位置位。
- 4. 在从属方的发送期间，如果通过 WRELn 位的置位来解除等待，就清除 TRCn 位。

备注 n=0

图 12-33 的“(3) 数据~数据~停止条件”的⑧~⑲的说明如下:

- ⑧ 主控方在第 8 个时钟的下降沿进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn: 传送结束中断)。因为主控方的 ACKEn 位为“0”，所以通过硬件给从属方发送 ACK。
- ⑨ 主控方读接收数据，解除等待 (WRELn=1)。
- ⑩ 从属方在第 9 个时钟的上升沿检测到 ACK (ACKDn=1)。
- ⑪ 从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0)，并且产生中断 (INTIICAn: 传送结束中断)。
- ⑫ 如果从属方给 IICA 移位寄存器 n (IICAn) 写发送数据，就解除从属方的等待，开始从属方到主控方的数据传送。
- ⑬ 主控方在第 8 个时钟的下降沿产生中断 (INTIICAn: 传送结束中断)，并且进入等待状态 (SCLAn=0)。因为进行 ACK 控制 (ACKEn=1)，所以此阶段的总线数据线变为低电平 (SDAAn=0)。
- ⑭ 主控方设定为 NACK 应答 (ACKEn=0)，并且将等待时序改为第 9 个时钟 (WTIMn=1)。
- ⑮ 如果主控方解除等待 (WRELn=1)，从属方就在第 9 个时钟的上升沿检测到 NACK (ACKDn=0)。
- ⑯ 主控方和从属方在第 9 个时钟的下降沿进入等待状态 (SCLAn=0)，并且都产生中断 (INTIICAn: 传送结束中断)。
- ⑰ 如果主控方发行停止条件 (SPTn=1)，就清除总线数据线 (SDAAn=0)，并且解除主控方的等待。此后，主控方处于待机状态，直到将总线时钟线置位 (SCLAn=1) 为止。
- ⑱ 从属方在确认 NACK 后停止发送，为了结束通信，解除等待 (WRELn=1)。如果解除从属方的等待，就将总线时钟线置位 (SCLAn=1)。
- ⑲ 如果主控方确认到总线时钟线被置位 (SCLAn=1)，就在经过停止条件准备时间后将总线数据线置位 (SDAAn=1)，然后发行停止条件 (通过 SCLAn=1 使 SDAAn 从“0”变为“1”)。如果生成停止条件，从属方就检测到停止条件，并且产生中断 (INTIICAn: 停止条件中断)。

备注 1. 图 12-33 的①~⑲是通过 I²C 总线进行数据通信的一系列运行步骤。

图 12-33 的“(1) 开始条件~地址~数据”说明步骤①~⑦。

图 12-33 的“(2) 地址~数据~数据”说明步骤③~⑫。

图 12-33 的“(3) 数据~数据~停止条件”说明步骤⑧~⑲。

2. n=0

第 13 章 乘除器和乘加器

13.1 乘除器和乘加器的功能

乘除器和乘加器有以下功能：

- 16 位×16 位=32 位（无符号）
- 16 位×16 位=32 位（带符号）
- 16 位×16 位+32 位=32 位（无符号）
- 16 位×16 位+32 位=32 位（带符号）
- 32 位÷32 位=32 位、余数 32 位（无符号）

13.2 乘除器和乘加器的结构

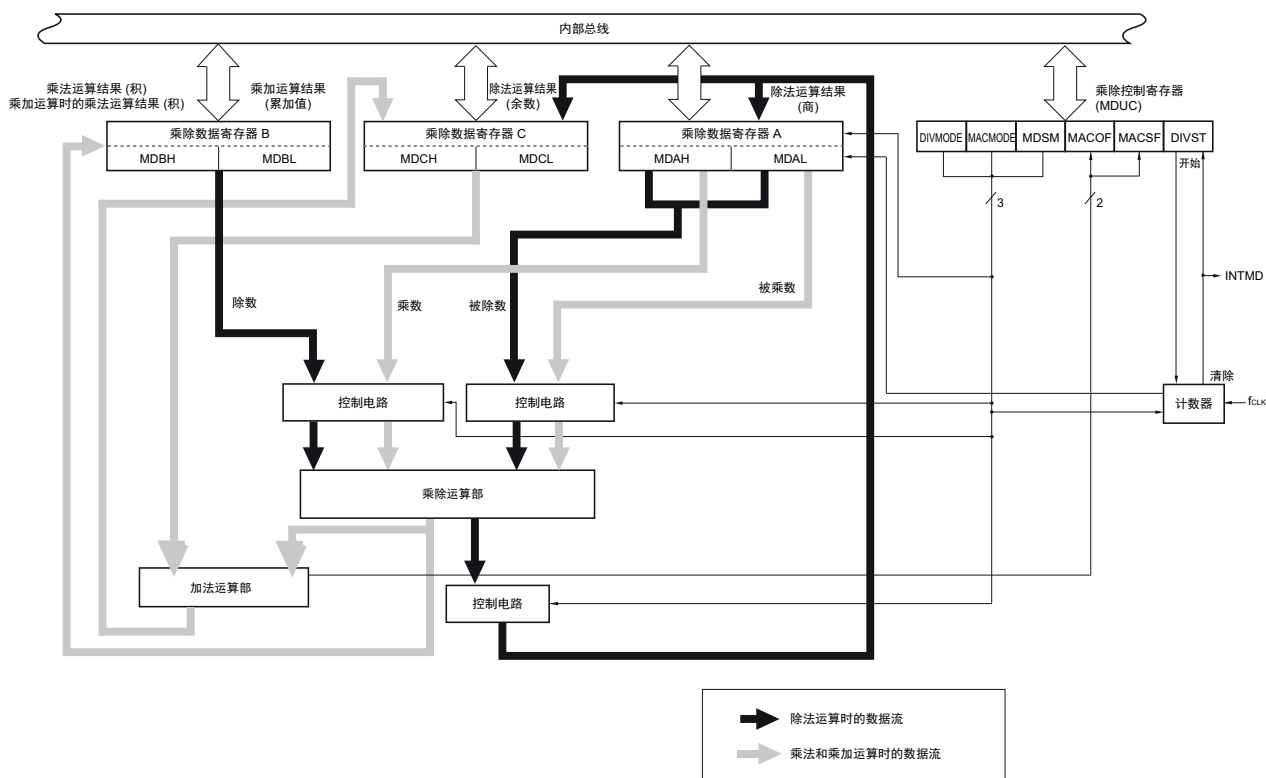
乘除器和乘加器由以下硬件构成。

表 13-1 乘除器和乘加器的结构

项目	结构
寄存器	乘除数据寄存器 A (L) (MDAL) 乘除数据寄存器 A (H) (MDAH) 乘除数据寄存器 B (L) (MDBL) 乘除数据寄存器 B (H) (MDBH) 乘除数据寄存器 C (L) (MDCL) 乘除数据寄存器 C (H) (MDCH)
控制寄存器	乘除控制寄存器 (MDUC)

乘除器和乘加器的框图如图 13-1 所示。

图 13-1 乘除器和乘加器的框图



备注 f_{CLK}: CPU/ 外围硬件的时钟频率

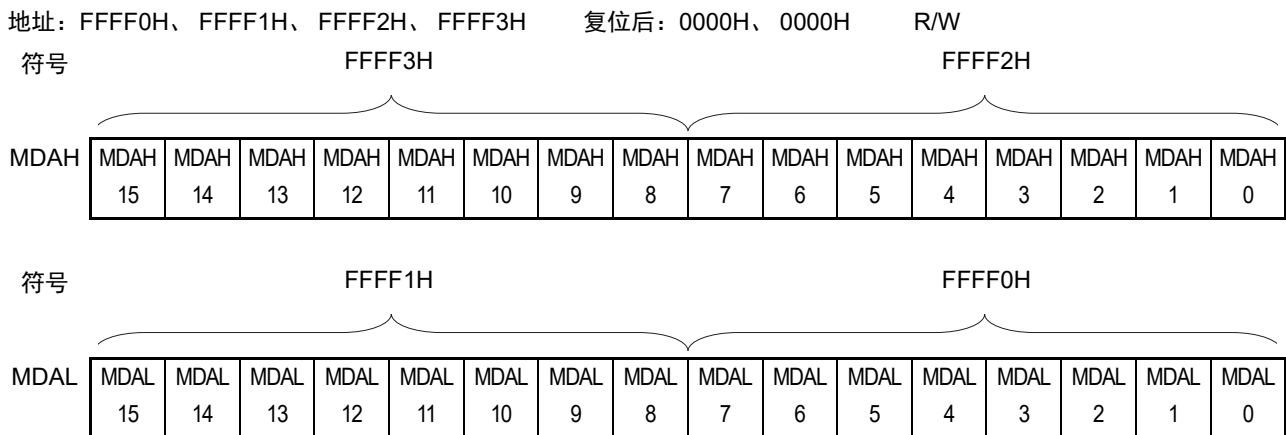
13.2.1 乘除数据寄存器 A (MDAH、MDAL)

MDAH 寄存器和 MDAL 寄存器设定用于乘除运算的值并且保存运算结果。在乘法运算模式或者乘加运算模式中，设定乘数和被乘数的数据；在除法运算模式中，设定被除数的数据。另外，在除法运算模式中，运算结果（商）保存在 MDAH 寄存器和 MDAL 寄存器。

通过 16 位操作指令设定 MDAH 寄存器和 MDAL 寄存器。

在产生复位信号后，这些寄存器的值变为“0000H”。

图 13-2 乘除数据寄存器 A (MDAH、MDAL) 的格式



- 注意 1. 在除法运算过程中（乘除控制寄存器（MDUC）为“81H”或者“C1H”），不能改写 MDAH 寄存器和 MDAL 寄存器的值。否则，虽然进行运算，但是运算结果为不定值。
2. 在除法运算过程中（MDUC 寄存器为“81H”或者“C1H”），如果读 MDAH 寄存器和 MDAL 寄存器的值，就不保证其读取值。
3. 在乘法运算模式（带符号）或者乘加运算模式（带符号）中，数据为 2 的补码。

运算时 MDAH 寄存器和 MDAL 寄存器的功能如下所示。

表 13-2 运算时 MDAH 寄存器和 MDAL 寄存器的功能

运算模式	设定	运算结果
乘法运算模式（无符号）	MDAH: 乘数（无符号）	—
乘加运算模式（无符号）	MDAL: 被乘数（无符号）	—
乘法运算模式（带符号）	MDAH: 乘数（带符号）	—
乘加运算模式（带符号）	MDAL: 被乘数（带符号）	—
除法运算模式（无符号）	MDAH: 被除数（无符号） （高 16 位） MDAL: 被除数（无符号） （低 16 位）	MDAH: 除法运算结果（商）（无符号） 高 16 位 MDAL: 除法运算结果（商）（无符号） 低 16 位

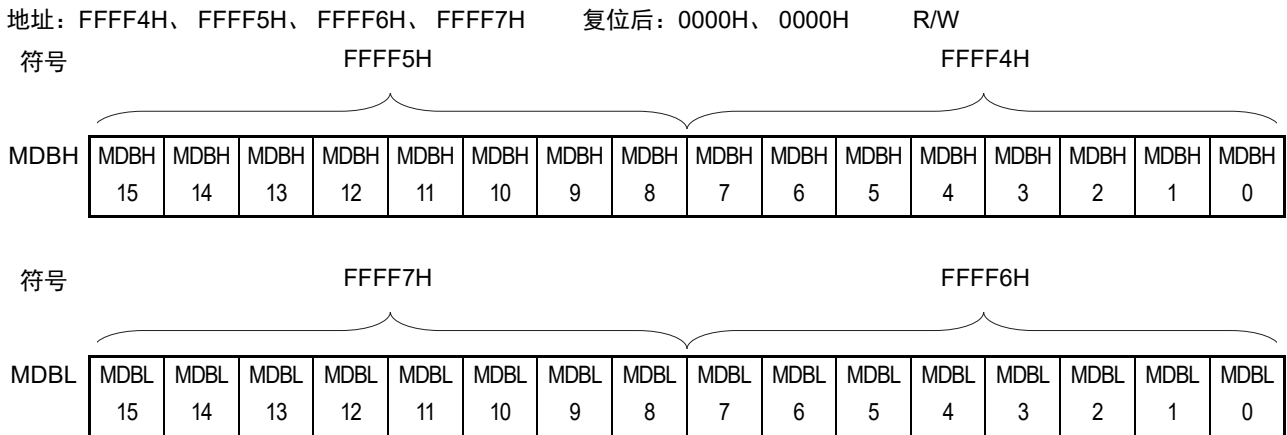
13.2.2 乘除数据寄存器 B (MDBL、MDBH)

MDBH 寄存器和 MDBL 寄存器设定用于乘除运算的值并且保存运算结果。在乘法运算模式或者乘加运算模式中，保存运算结果（积）；在除法运算模式中，设定除数的数据。

通过 16 位操作指令设定 MDBH 寄存器和 MDBL 寄存器。

在产生复位信号后，这些寄存器的值变为“0000H”。

图 13-3 乘除数据寄存器 B (MDBH、MDBL) 的格式



- 注意 1. 在除法运算（乘除控制寄存器（MDUC）为“81H”或者“C1H”）或者乘加运算过程中，不能通过软件改写 MDBH 寄存器和 MDBL 寄存器的值。否则，运算结果为不定值。
2. 在除法运算模式中，不能将 MDBH 寄存器和 MDBL 寄存器置“0000H”。否则，运算结果为不定值。
3. 在乘法运算模式（带符号）或者乘加运算模式（带符号）中，数据为 2 的补码。

运算时 MDBH 寄存器和 MDBL 寄存器的功能如下所示。

表 13-3 运算时 MDBH 寄存器和 MDBL 寄存器的功能

运算模式	设定	运算结果
乘法运算模式（无符号） 乘加运算模式（无符号）	—	MDBH: 乘法运算结果（积）（无符号）高 16 位 MDBL: 乘法运算结果（积）（无符号）低 16 位
乘法运算模式（带符号） 乘加运算模式（带符号）	—	MDBH: 乘法运算结果（积）（带符号）高 16 位 MDBL: 乘法运算结果（积）（带符号）低 16 位
除法运算模式（无符号）	MDBH: 除数（无符号） （高 16 位） MDBL: 除数（无符号） （低 16 位）	—

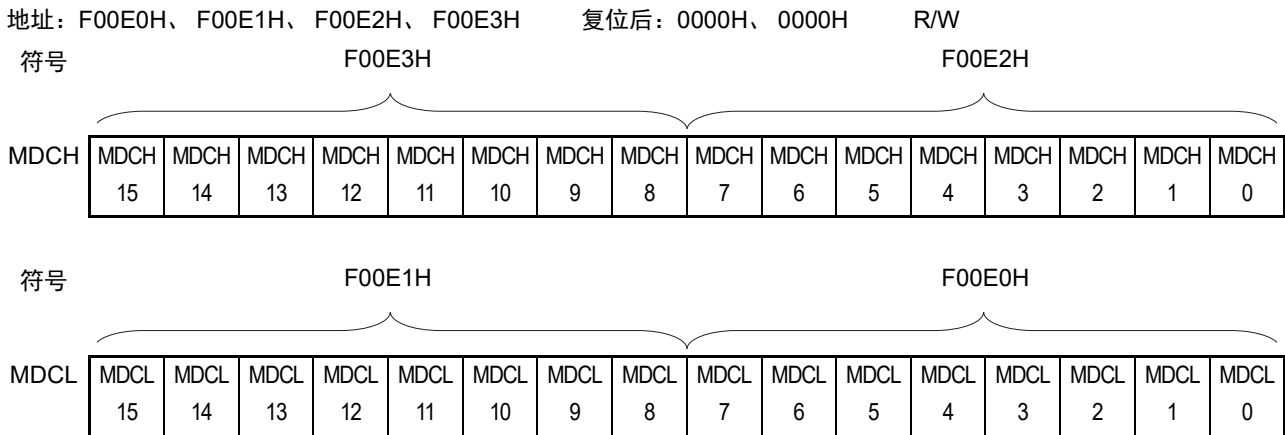
13.2.3 乘除数据寄存器 C (MDCL、MDCH)

MDCH 寄存器和 MDCL 寄存器在乘加运算模式中，保存累加结果；在除法运算模式中，保存运算结果的余数。在乘法运算模式中，不使用这些寄存器。

通过 16 位操作指令设定 MDCH 寄存器和 MDCL 寄存器。

在产生复位信号后，这些寄存器的值变为“0000H”。

图 13-4 乘除数据寄存器 C (MDCH、MDCL) 的格式



- 注意 1. 在除法运算过程中（乘除控制寄存器（MDUC）为“81H”或者“C1H”），如果读 MDCH 寄存器和 MDCL 寄存器的值，就不保证其读取值。
2. 在乘加运算过程中，不能通过软件改写 MDCH 寄存器和 MDCL 寄存器的值。否则，运算结果为不定值。
3. 在乘加运算模式（带符号）中，数据为 2 的补码。

表 13-4 运算时 MDCH 寄存器和 MDCL 寄存器的功能

运算模式	设定	运算结果
乘法运算模式 (带符号或者无符号)	—	—
乘加运算模式（无符号）	MDCH: 累加初始值（无符号）（高 16 位） MDCL: 累加初始值（无符号）（低 16 位）	MDCH: 累加值（无符号）（高 16 位） MDCL: 累加值（无符号）（低 16 位）
乘加运算模式（带符号）	MDCH: 累加初始值（带符号）（高 16 位） MDCL: 累加初始值（带符号）（低 16 位）	MDCH: 累加值（带符号）（高 16 位） MDCL: 累加值（带符号）（低 16 位）
除法运算模式（无符号）	—	MDCH: 余数（无符号）（高 16 位） MDCL: 余数（无符号）（低 16 位）

进行乘法运算和除法运算时的寄存器结构如下所示。

- 乘法运算时的寄存器结构

$$\begin{array}{ccc} \langle \text{乘数 A} \rangle & \langle \text{乘数 B} \rangle & \langle \text{积} \rangle \\ \text{MDAL (bit15} \sim \text{0)} \times \text{MDAH (bit15} \sim \text{0)} & = & [\text{MDBH (bit15} \sim \text{0)}, \text{MDBL (bit15} \sim \text{0)}] \end{array}$$

- 乘加运算时的寄存器结构

$$\begin{array}{ccc} \langle \text{乘数 A} \rangle & \langle \text{乘数 B} \rangle & \langle \text{累加值} \rangle \\ \text{MDAL (bit15} \sim \text{0)} \times \text{MDAH (bit15} \sim \text{0)} + \text{MDC (bit31} \sim \text{0)} & = & \\ \langle \text{累加结果} \rangle & & \\ [\text{MDCH (bit15} \sim \text{0)}, \text{MDCL (bit15} \sim \text{0)}] & & \\ (\text{MDBH (bit15} \sim \text{0)} \text{ 和 } \text{MDBL (bit15} \sim \text{0)}) & \text{保存乘法运算结果} & \end{array}$$

- 除法运算时的寄存器结构

$$\begin{array}{ccc} \langle \text{被除数} \rangle & & \langle \text{除数} \rangle \\ [\text{MDAH (bit15} \sim \text{0)}, \text{MDAL (bit15} \sim \text{0)}] \div [\text{MDBH (bit15} \sim \text{0)}, \text{MDBL (bit15} \sim \text{0)}] & = & \\ \langle \text{商} \rangle & & \langle \text{余数} \rangle \\ [\text{MDAH (bit15} \sim \text{0)}, \text{MDAL (bit15} \sim \text{0)}] \dots [\text{MDCH (bit15} \sim \text{0)}, \text{MDCL (bit15} \sim \text{0)}] & & \end{array}$$

13.3 控制乘除器和乘加器的寄存器

通过乘除控制寄存器（MDUC）来控制乘除器和乘加器。

13.3.1 乘除控制寄存器（MDUC）

MDUC 寄存器是控制乘除器和乘加器运算的 8 位寄存器。

通过 1 位或者 8 位存储器操作指令设定 MDUC 寄存器。

但是，只能读乘加运行结果（累加值）的上溢标志（MACOF）和符号标志（MACSF）。

在产生复位信号后，此寄存器的值变为“00H”。

图 13-5 乘除控制寄存器（MDUC）的格式

地址: F00E8H	复位后: 00H	R/W 注 1						
符号	7	6	5	4	3	2	1	0
MDUC	DIVMODE	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

DIVMODE	MACMODE	MDSM	运算模式的选择
0	0	0	乘法运算模式（无符号）（默认）
0	0	1	乘法运算模式（带符号）
0	1	0	乘加运算模式（无符号）
0	1	1	乘加运算模式（带符号）
1	0	0	除法运算模式（无符号），产生除法运算结束中断（INTMD）。
1	1	0	除法运算模式（无符号），不产生除法运算结束中断（INTMD）。
上述以外			禁止设定。

MACOF	乘加运算结果（累加值）的上溢标志
0	没有发生上溢。
1	发生上溢。

[置位条件]

- 乘加运算模式（无符号）
当累加值超出 00000000h ~ FFFFFFFFh 的范围时
- 乘加运算模式（带符号）
当正累加值与正积之和超过 7FFFFFFFh 而结果为负时
当负累加值与负积之和超过 80000000h 而结果为正时

MACSF	乘加结果（累加值）的符号标志
0	累加值为正。
1	累加值为负。

乘加运算模式（无符号）：总是“0”。

乘加运算模式（带符号）：表示累加值的符号位。

DIVST 注 2	除法运算的开始 / 停止
0	除法运算处理结束。
1	开始除法运算 / 正在进行除法运算。

- 注 1. bit1 和 bit2 是只读位。
2. 只有在除法运算模式中才能将 DIVST 位置“1”。在除法运算模式中，如果将 DIVST 位置“1”，就开始除法运算。在运算结束后，自动将 DIVST 位清“0”。在乘法运算模式中，通过给乘除数据寄存器 A（MDAH、MDAL）设定乘数和被乘数，自动开始运算。

- 注意 1. 在运算过程中（DIVST 位为“1”），不能改写 DIVMODE 位和 MDSM 位。否则，运算结果为不定值。
2. 在除法运算过程中（DIVST 位为“1”），不能通过软件将 DIVST 位清“0”。

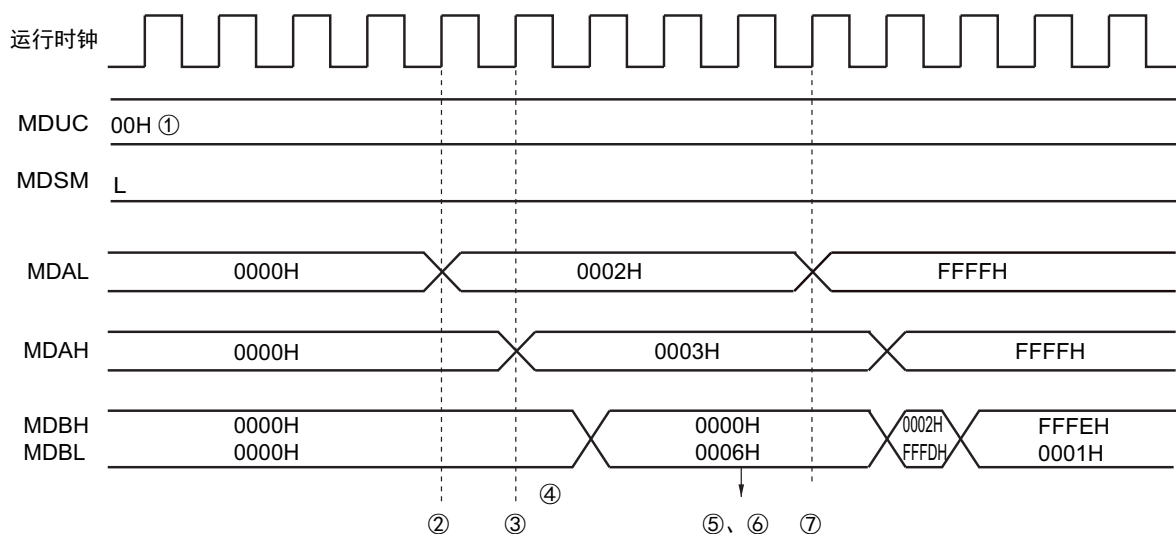
13.4 乘除器和乘加器的运算

13.4.1 乘法运算（无符号）

- 初始设定
 - ① 将乘除控制寄存器（MDUC）置“00H”。
 - ② 给乘除数据寄存器 A（L）（MDAL）设定被乘数。
 - ③ 给乘除数据寄存器 A（H）（MDAH）设定乘数。
（②和③的设定顺序不分先后。如果给MDAH寄存器和MDAL寄存器设定乘数和被乘数，就自动开始乘法运算）
- 运算处理
 - ④ 至少等待1个时钟。运算在1个时钟内完成。
- 运算结束
 - ⑤ 从乘除数据寄存器 B（L）（MDBL）读积（低16位）。
 - ⑥ 从乘除数据寄存器 B（H）（MDBH）读积（高16位）。
（⑤和⑥的读取顺序不分先后）
- 下一次运算
 - ⑦ 在更改运行模式时，必须从各运算步骤的初始设定开始进行。
在继续使用相同的运算模式时，能省略①和②的设定。

备注 步骤①~⑦对应图 13-6 中的①~⑦。

图 13-6 乘法运算（无符号）的时序图（2×3=6）



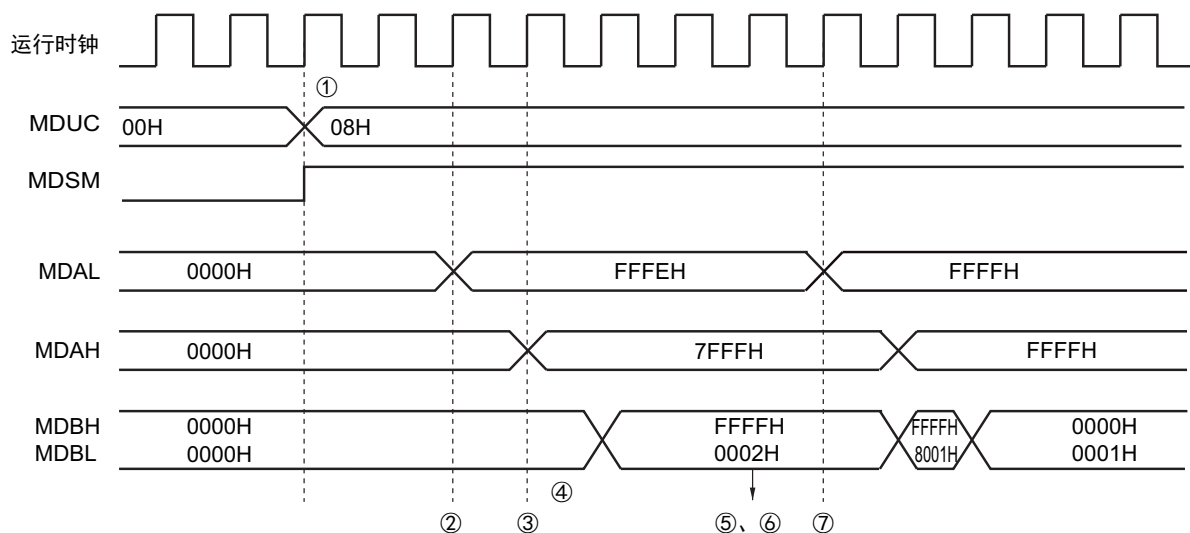
13.4.2 乘法运算（带符号）

- 初始设定
 - ① 将乘除控制寄存器（MDUC）置“08H”。
 - ② 给乘除数据寄存器 A（L）（MDAL）设定被乘数。
 - ③ 给乘除数据寄存器 A（H）（MDAH）设定乘数。
（②和③的设定顺序不分先后。如果给MDAH寄存器和MDAL寄存器设定乘数和被乘数，就自动开始乘法运算）
- 运算处理
 - ④ 至少等待1个时钟。运算在1个时钟内完成。
- 运算结束
 - ⑤ 从乘除数据寄存器 B（L）（MDBL）读积（低16位）。
 - ⑥ 从乘除数据寄存器 B（H）（MDBH）读积（高16位）。
（⑤和⑥的读取顺序不分先后）
- 下一次运算
 - ⑦ 在更改运算模式时，必须从各运算步骤的初始设定开始进行。
在继续使用相同的运算模式时，能省略①和②的设定。

注意 在乘法运算模式（带符号）中，数据为 2 的补码。

备注 步骤①~⑦对应图 13-7 中的①~⑦。

图 13-7 乘法运算（带符号）的时序图（ $-2 \times 32767 = -65534$ ）



13.4.3 乘加运算（无符号）

- 初始设定
 - ① 将乘除控制寄存器（MDUC）置“40H”。
 - ② 给乘除数据寄存器C（H）（MDCH）设定累加初始值的高16位。
 - ③ 给乘除数据寄存器C（L）（MDCL）设定累加初始值的低16位。
 - ④ 给乘除数据寄存器A（L）（MDAL）设定被乘数。
 - ⑤ 给乘除数据寄存器A（H）（MDAH）设定乘数。

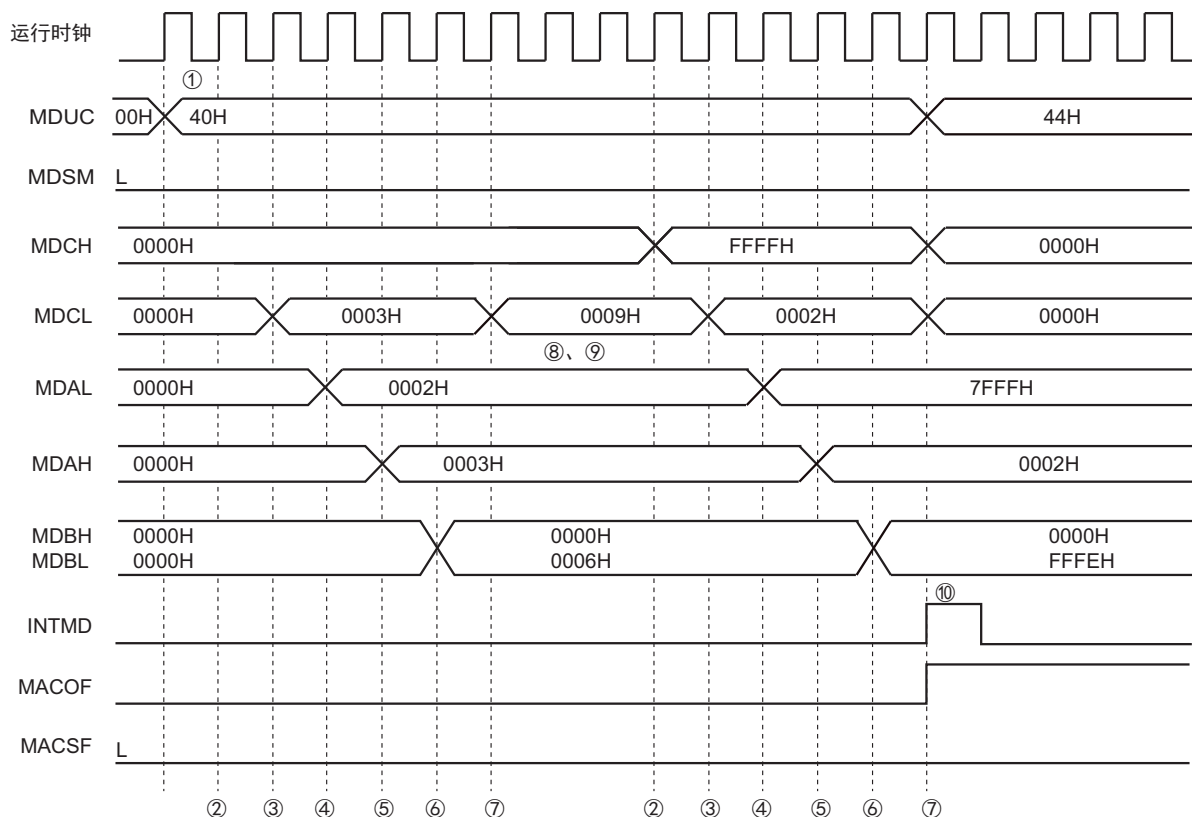
（②、③和④的设定顺序不分先后。如果给⑤的MDAH寄存器设定乘数，就自动开始乘法运算）
- 运算处理
 - ⑥ 乘法运算在1个时钟内完成。
（乘法结果保存在乘除数据寄存器B（L）（MDBL）和乘除数据寄存器B（H）（MDBH））
 - ⑦ ⑥之后，乘加运算在1个时钟内完成（在初始设定结束（⑤）后，至少等待2个时钟）。
- 运算结束
 - ⑧ 从乘除数据寄存器C（L）（MDCL）读累加值（低16位）。
 - ⑨ 从乘除数据寄存器C（H）（MDCH）读累加值（高16位）。

（⑧和⑨的读取顺序不分先后）

（⑩当乘加运算的结果发生上溢时，将MACOF位置“1”并且产生INTMD信号）
- 下一次运算
 - ⑪ 在更改运算模式时，必须从各运算步骤的初始设定开始进行。
在继续使用相同的运算模式时，能省略①~④的设定。

备注 步骤①~⑩对应图 13-8 中的①~⑩。

图 13-8 乘加运算（无符号）的时序图
 $(2 \times 3 + 3 = 9 \rightarrow 32767 \times 2 + 4294901762 = 0 \text{ (发生上溢)})$



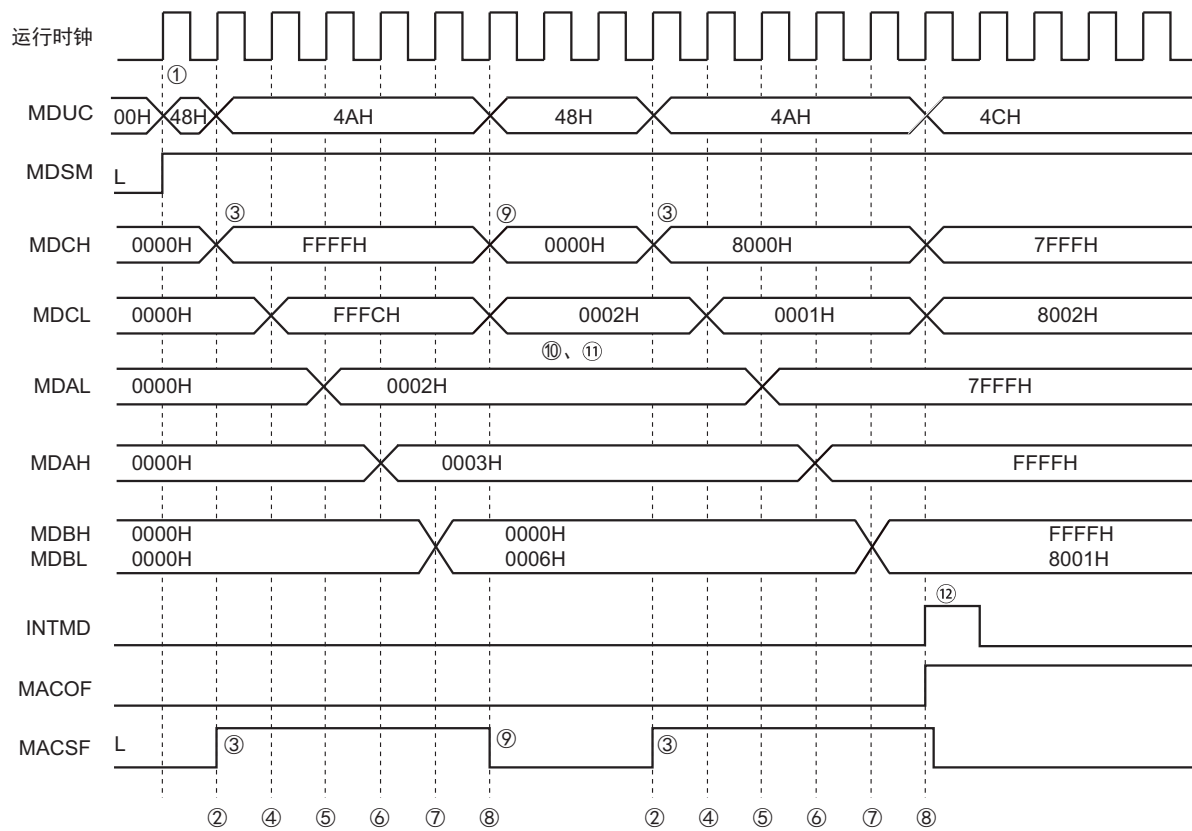
13.4.4 乘加运算（带符号）

- 初始设定
 - ① 将乘除控制寄存器（MDUC）置“48H”。
 - ② 给乘除数据寄存器C（H）（MDCH）设定累加初始值的高16位。
（③ 当MDCH寄存器的累加值为负值时，将MACSF位置“1”）
 - ④ 给乘除数据寄存器C（L）（MDCL）设定累加初始值的低16位。
 - ⑤ 给乘除数据寄存器A（L）（MDAL）设定被乘数。
 - ⑥ 给乘除数据寄存器A（H）（MDAH）设定乘数。
（②、④和⑤的设定顺序不分先后。如果给⑥的MDAH寄存器设定乘数，就自动开始乘法运算）
- 运算处理
 - ⑦ 乘法运算在1个时钟内完成。
（乘法结果保存在乘除数据寄存器B（L）（MDBL）和乘除数据寄存器B（H）（MDBH））
 - ⑧ ⑦之后，乘加运算在1个时钟内完成（在初始设定结束（⑥）后，至少等待2个时钟）。
- 运算结束
 - ⑨ 当保存在MDCL寄存器和MDCH寄存器中的累加值为正值时，将MACSF位清“0”。
 - ⑩ 从MDCL寄存器读累加值（低16位）。
 - ⑪ 从MDCH寄存器读累加值（高16位）。
（⑩和⑪的读取顺序不分先后）
 - ⑫ 当乘加运算的结果发生上溢时，将MACOF位置“1”并且产生INTMD信号）
- 下一次运算
 - ⑬ 在更改运算模式时，必须从各运算步骤的初始设定开始进行。
在继续使用相同的运算模式时，能省略①～⑤的设定。

注意 在乘加运算模式（带符号）中，数据为2的补码。

备注 步骤①～⑫对应图 13-9 中的①～⑫。

图 13-9 乘加运算（带符号）的时序图
 $(2 \times 3 + (-4)) = 2 \rightarrow 32767 \times (-1) + (-2147483647) = -2147450882$ （发生上溢）

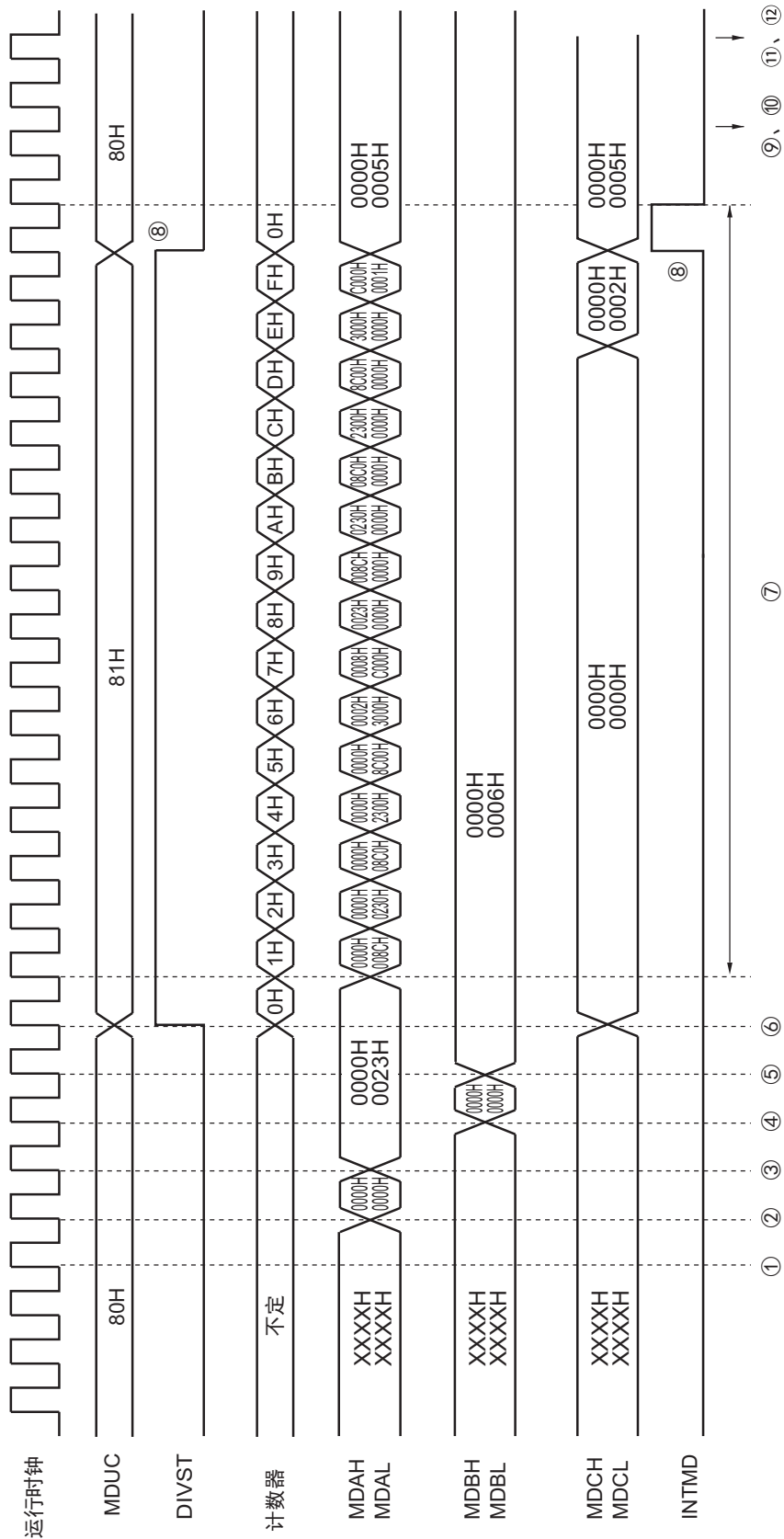


13.4.5 除法运算

- 初始设定
 - ① 将乘除控制寄存器 (MDUC) 置“80H”。
 - ② 给乘除数据寄存器 A (H) (MDAH) 设定被除数 (高 16 位)。
 - ③ 给乘除数据寄存器 A (L) (MDAL) 设定被除数 (低 16 位)。
 - ④ 给乘除数据寄存器 B (H) (MDBH) 设定除数 (高 16 位)。
 - ⑤ 给乘除数据寄存器 B (L) (MDBL) 设定除数 (低 16 位)。
 - ⑥ 将 MDUC 寄存器的 bit0 (DIVST) 置“1”。
(②~⑤的顺序不分先后)
- 运算处理
 - ⑦ 如果完成以下某种处理, 就结束运算。
 - 至少等待 16 个时钟 (运算在 16 个时钟内完成)。
 - 确认 DIVST 位的清除。
(在运算过程中, 不保证 MDBL、MDBH、MDCL、MDCH 寄存器的读取值)
- 运算结束
 - ⑧ 将 DIVST 位清“0”并且结束运算。此时, 如果是 MACMODE=0 的运算, 就产生中断请求信号 (INTMD)。
 - ⑨ 从 MDAL 寄存器读商 (低 16 位)。
 - ⑩ 从 MDAH 寄存器读商 (高 16 位)。
 - ⑪ 从乘除数据寄存器 C (L) (MDCL) 读余数 (低 16 位)。
 - ⑫ 从乘除数据寄存器 C (H) (MDCH) 读余数 (高 16 位)。
(⑨~⑫的顺序不分先后)
- 下一次运算
 - ⑬ 在更改运算模式时, 必须从各运算步骤的初始设定开始进行。
在继续使用相同的运算模式时, 能省略①~⑤的设定。

备注 步骤①~⑫对应图 13-10 中的①~⑫。

图 13-10 除法运算的时序图 (35÷6=5 余数 5)



第 14 章 DMA 控制器

R7F0C908B2、R7F0C907B2、R7F0C906B2 内置一个 DMA (Direct Memory Access) 控制器。

能在支持 DMA 外围硬件的 SFR 和内部 RAM 之间不经过 CPU 而自动传送数据。

因为能在进行通常的 CPU 内部运算和数据传送的同时进行 SFR 和内部 RAM 之间的传送, 所以能进行大容量数据的处理, 并且还能实现通过通信、定时器和 A/D 进行的实时控制。

14.1 DMA 控制器的功能

- DMA 通道个数: 2 个通道 (本产品)
- 传送单位: 8 位或者 16 位
- 最大传送单位: 1024 次
- 传送类型: 2 个周期传送 (通过 2 个时钟进行一次传送并且在传送期间 CPU 停止运行)
- 传送模式: 单次传送模式
- 传送请求: 选择以下外围硬件中断。
 - A/D 转换器
 - 串行接口 (CSI00、CSI11、CSI20、UART0~UART2)
 - 定时器 (通道 0、1、2、3)
- 传送对象: SFR 和内部 RAM 之间的传送

使用 DMA 功能的例子如下所示:

- 串行接口的连续传送
- A/D 转换结果的连续取得
- 每隔一段时间的端口值的取得

14.2 DMA 控制器的结构

DMA 控制器由以下硬件构成。

表 14-1 DMA 控制器的结构

项目	结构
地址寄存器	<ul style="list-style-type: none"> • DMA SFR 地址寄存器 0、1 (DSA0、DSA1) • DMA RAM 地址寄存器 0、1 (DRA0、DRA1)
计数寄存器	<ul style="list-style-type: none"> • DMA 字节计数寄存器 0、1 (DBC0、DBC1)
控制寄存器	<ul style="list-style-type: none"> • DMA 模式控制寄存器 0、1 (DMC0、DMC1) • DMA 运行控制寄存器 0、1 (DRC0、DRC1)

14.2.1 DMA SFR 地址寄存器 n (DSAn)

这是设定 SFR 地址的 8 位寄存器，SFR 地址是 DMA 通道 n 的传送源或者传送目标。

必须设定 SFR 地址 FFF00H ~ FFFFFH 的低 8 位。

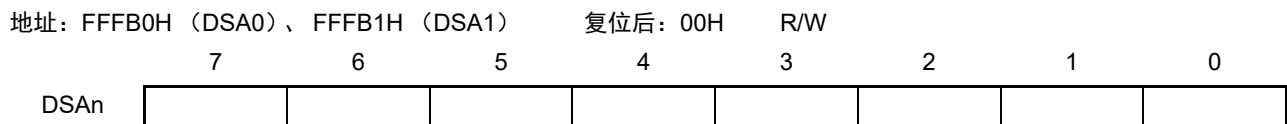
此寄存器为固定值而不会自动递增。

在 16 位传送模式中，忽略最低位而作为偶数地址处理。

能以 8 位为单位读写 DSAn 寄存器，但是不能在 DMA 传送过程中写此寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 14-1 DMA SFR 地址寄存器 n (DSAn) 的格式



备注 n: DMA 通道号 (n=0、1)

14.2.2 DMA RAM 地址寄存器 n (DRAn)

这是设定 RAM 地址的 16 位寄存器，RAM 地址是 DMA 通道 n 的传送源或者传送目标。

能设定通用寄存器以外的内部 RAM 区（参照表 14-2）的地址。

必须设定 RAM 地址的低 16 位。

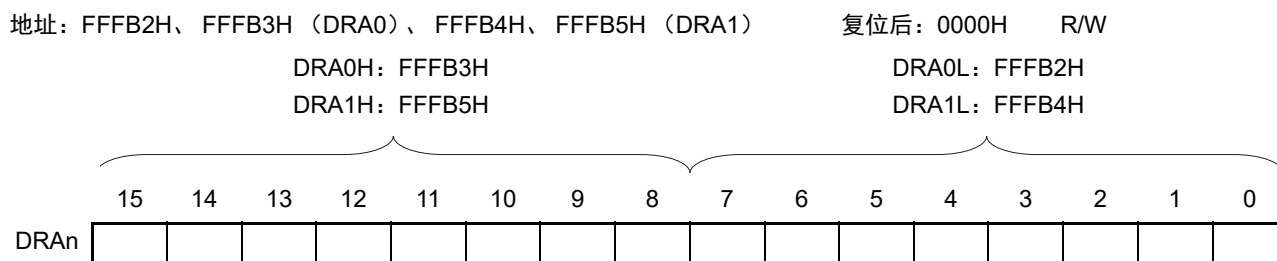
如果开始 DMA 传送，此寄存器就自动递增。在 8 位传送模式中增 1；在 16 位传送模式中增 2。如果从此 DRAn 寄存器的设定地址开始传送到最后地址，DMA 传送就停止运行。此时，在 8 位传送模式中，DRAn 寄存器为最后地址 +1；在 16 位传送模式中，DRAn 寄存器为最后地址 +2。

在 16 位传送模式中，忽略最低位而作为偶数地址处理。

能以 8 位或者 16 位为单位读写 DRAn 寄存器，但是不能在 DMA 传送过程中写此寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 14-2 DMA RAM 地址寄存器 n (DRAn) 的格式



备注 n: DMA 通道号 (n=0、1)

表 14-2 通用寄存器以外的内部 RAM 区

产品	通用寄存器以外的内部 RAM 区
R7F0C908B2	FF700H ~ FFEDFH
R7F0C907B2、R7F0C906B2	FC000H ~ FFEDFH

14.2.3 DMA 字节计数寄存器 n (DBCn)

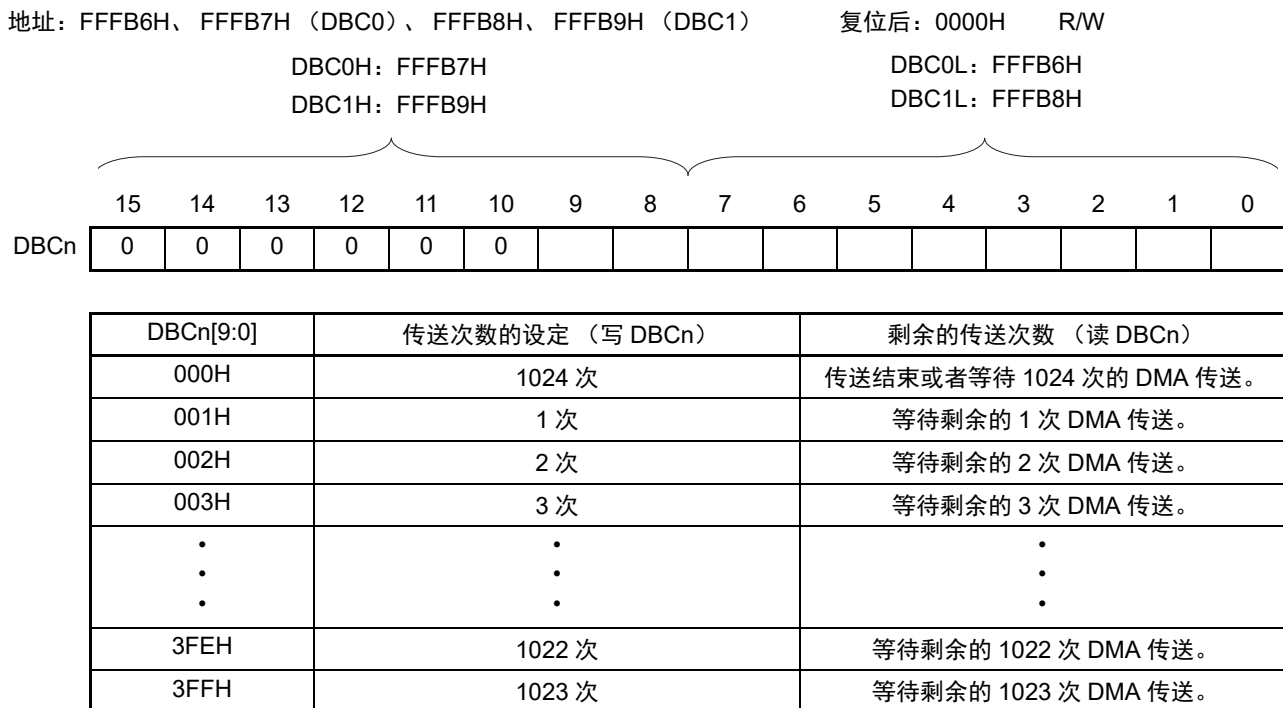
这是设定 DMA 通道 n 传送次数的 10 位寄存器。必须在 DMA 传送前给此 DBCn 寄存器设定连续的传送次数（最多 1024 次）。

每进行一次 DMA 传送，此寄存器就自动递减。在 DMA 传送过程中，能通过读此 DBCn 寄存器获得剩余的连续传送次数。

能以 8 位或者 16 位为单位读写 DBCn 寄存器，但是不能在 DMA 传送过程中写此寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 14-3 DMA 字节计数寄存器 n (DBCn) 的格式



注意 1. 必须将 bit15 ~ 10 置“0”。

2. 如果指定通用寄存器或者因连续传送而超过内部 RAM 空间，就读写通用寄存器或者 SFR 空间，导致数据的破坏。必须将传送次数设定在内部 RAM 空间范围内。

备注 n: DMA 通道号 (n=0、1)

14.3 控制 DMA 控制器的寄存器

控制 DMA 控制器的寄存器如下所示：

- DMA 模式控制寄存器 n (DMCn)
- DMA 运行控制寄存器 n (DRCn)

备注 n: DMA 通道号 (n=0、1)

14.3.1 DMA 模式控制寄存器 n (DMCn)

DMCn 寄存器设定 DMA 通道 n 的传送模式，选择传送方向、数据长度、保留设定和启动源。bit7 (STGn) 为启动 DMA 的软件触发。

禁止在 DMA 运行过程中 (DSTn=1) 改写 DMCn 寄存器的 bit6、5、3 ~ 0。

通过 1 位或者 8 位存储器操作指令设定 DMCn 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 14-4 DMA 模式控制寄存器 n (DMCn) 的格式 (1/2)

地址: FFFBAH (DMC0)、FFFBH (DMC1)	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0
STGn 注 1	DMA 传送开始的软件触发							
0	没有软件触发。							
1	在允许 DMA 运行 (DENn=1) 时，开始 DMA 传送。							
在允许 DMA 运行 (DENn=1) 时，通过给 STGn 位写“1”进行一次 DMA 传送。 此位的读取值总是“0”。								
DRSn	DMA 传送方向的选择							
0	SFR → 内部 RAM							
1	内部 RAM → SFR							
DSn	DMA 传送数据长度的指定							
0	8 位							
1	16 位							
DWAITn 注 2	DMA 传送的保留							
0	通过 DMA 启动请求进行 DMA 传送 (不保留)。							
1	即使接收到 DMA 启动请求也保留 DMA 传送。							
能通过将 DWAITn 位的值从“1”置为“0”来开始被保留的 DMA 传送。 在将 DWAITn 位的值从“0”置为“1”后，实际上需要经过 2 个时钟，然后保留 DMA 传送。								

注 1. 与 IFCn3 ~ IFCn0 位的值无关，能使用软件触发 (STGn)。

2. 要在使用 2 个或者更多通道的 DMA 过程中保留 DMA 传送时，必须保留全部通道的 DMA 传送 (DWAIT0=DWAIT1=DWAIT2=DWAIT3=1)。

备注 n: DMA 通道号 (n=0、1)

图 14-4 DMA 模式控制寄存器 n (DMCn) 的格式 (2/2)

地址: FFFBAH (DMC0)、FFFBH (DMC1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	DMA 启动源的选择注	
				触发信号	触发内容
0	0	0	0	—	禁止通过中断触发 DMA 传送。 (只能允许软件触发)
0	0	0	1	INTAD	A/D 转换结束中断
0	0	1	0	INTTM00	定时器通道 0 的计数结束或者捕捉中断
0	0	1	1	INTTM01	定时器通道 1 的计数结束或者捕捉中断
0	1	0	0	INTTM02	定时器通道 2 的计数结束或者捕捉中断
0	1	0	1	INTTM03	定时器通道 3 的计数结束或者捕捉中断
0	1	1	0	INTST0/INTCSI00	UART0 发送的传送结束或者缓冲器空中断 /CSI00 的传送结束或者缓冲器空中断
0	1	1	1	INTSR0	UART0 接收的传送结束中断
1	0	0	0	INTST1	UART1 发送的传送结束或者缓冲器空中断
1	0	0	1	INTSR1/INTCSI11	UART1 接收的传送结束 /CSI11 的传送结 束或者缓冲器空中断
1	0	1	0	INTST2/INTCSI20	UART2 发送的传送结束或者缓冲器空中断 /CSI20 的传送结束或者缓冲器空中断
1	0	1	1	INTSR2	UART2 接收的传送结束中断
上述以外				禁止设定。	

注 与 IFCn3 ~ IFCn0 位的值无关, 能使用软件触发 (STGn)。

备注 n: DMA 通道号 (n=0、1)

14.3.2 DMA 运行控制寄存器 n (DRCn)

DRCn 寄存器设定允许或者禁止 DMA 通道 n 的传送。

禁止在 DMA 运行过程中 (DSTn=1) 改写 DRCn 寄存器的 bit7 (DENn)。

通过 1 位或者 8 位存储器操作指令设定 DRCn 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 14-5 DMA 运行控制寄存器 n (DRCn) 的格式

地址: FFFBCH (DRC0)、FFFB DH (DRC1) 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA 运行允许标志
0	禁止 DMA 通道 n 的运行 (停止 DMA 的运行时钟)。
1	允许 DMA 通道 n 的运行。

在设定为允许 DMA 运行 (DENn=1) 后, 通过将 DSTn 标志置“1”进入等待 DMA 触发的状态。

DSTn	DMA 传送模式标志
0	DMA 通道 n 的 DMA 传送结束。
1	DMA 通道 n 的 DMA 传送没有结束 (正在传送)。

在设定为允许 DMA 运行 (DENn=1) 后, 通过将 DSTn 标志置“1”进入等待 DMA 触发的状态。
如果输入软件触发 (STGn) 或者输入由 IFCn3 ~ IFCn0 位设定的启动源触发, 就开始 DMA 传送。
此后, 如果 DMA 传送结束, 就自动将此标志清“0”。
要在 DMA 传送过程中强制结束传送时, 给此标志写“0”。

注意 如果 DMA 传送结束, 就自动将 DSTn 标志清“0”。

因为只有在 DSTn 标志为“0”时才允许写 DENn 标志, 所以当不等待 DMA 中断 (INTDMA n) 的发生而结束 DMA 传送时, 必须在将 DSTn 标志置“0”后将 DENn 标志置“0” (详细内容请参照“14.5.5 通过软件强制结束”)。

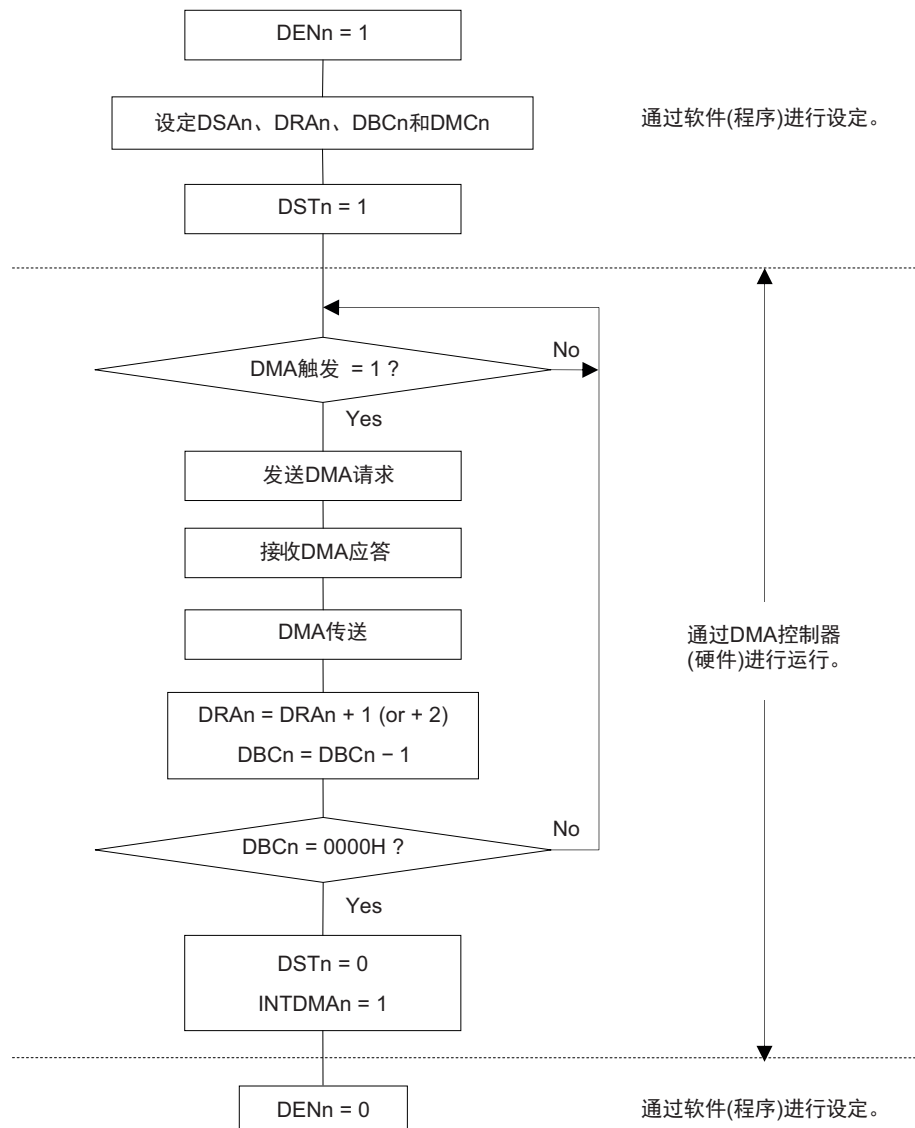
备注 n: DMA 通道号 (n=0、1)

14.4 DMA 控制器的运行

14.4.1 运行步骤

- ① 通过将DENn位置“1”，使DMA控制器变为允许运行状态。必须在将DENn位置“1”后写其他寄存器。当通过8位操作指令进行写操作时，写“80H”。
- ② 给DMA SFR 地址寄存器n（DSAn）、DMA RAM 地址寄存器n（DRAn）、DMA 字节计数寄存器n（DBCn）和DMA 模式控制寄存器n（DMCn）设定DMA 传送的SFR 地址、RAM 地址、传送次数和传送模式。
- ③ 通过将DSTn位置“1”，使DMA 变为等待触发的状态。当通过8位操作指令进行写操作时，写“81H”。
- ④ 如果输入软件触发（STGn）或者输入由IFCn3～IFCn0位设定的启动源触发，就开始DMA 传送。
- ⑤ 如果DBCn 寄存器设定的传送次数变为“0”，就完成传送并且产生中断请求（INTDMA n）而自动结束传送。
- ⑥ 此后，如果不使用DMA 控制器，就必须将DENn位置“0”，进入运行停止状态。

图 14-6 运行步骤



备注 n: DMA 通道号 (n=0、1)

14.4.2 传送模式

能通过设定 DMA 模式控制寄存器 n (DMCn) 的 bit6 和 bit5 (DRSn、DSn)，给 DMA 传送选择以下 4 种传送模式。

DRSn	DSn	DMA 传送模式
0	0	从 1 字节数据 (固定地址) 的 SFR 到 RAM (地址递增 +1) 的传送
0	1	从 2 字节数据 (固定地址) 的 SFR 到 RAM (地址递增 +2) 的传送
1	0	从 1 字节数据 (地址递增 +1) 的 RAM 到 SFR (固定地址) 的传送
1	1	从 2 字节数据 (地址递增 +2) 的 RAM 到 SFR (固定地址) 的传送

通过这些传送模式，能利用串行接口最多连续传送 1024 字节的数据，并且能连续传送 A/D 转换结果的数据以及利用定时器每隔一段时间扫描端口的数据。

14.4.3 DMA 传送的结束

如果 DBCn 变为“00H”而完成 DMA 传送，就自动将 DSTn 位清“0”，然后产生中断请求 (INTDMAn) 而结束传送。

如果为了强制结束 DMA 传送而将 DSTn 位清“0”，DMA 字节计数寄存器 n (DBCn) 和 DMA RAM 地址寄存器 n (DRAn) 就保持停止时的值。

当强制结束传送时，不产生中断请求 (INTDMAn)。

备注 n: DMA 通道号 (n=0、1)

14.5 DMA 控制器的设定例子

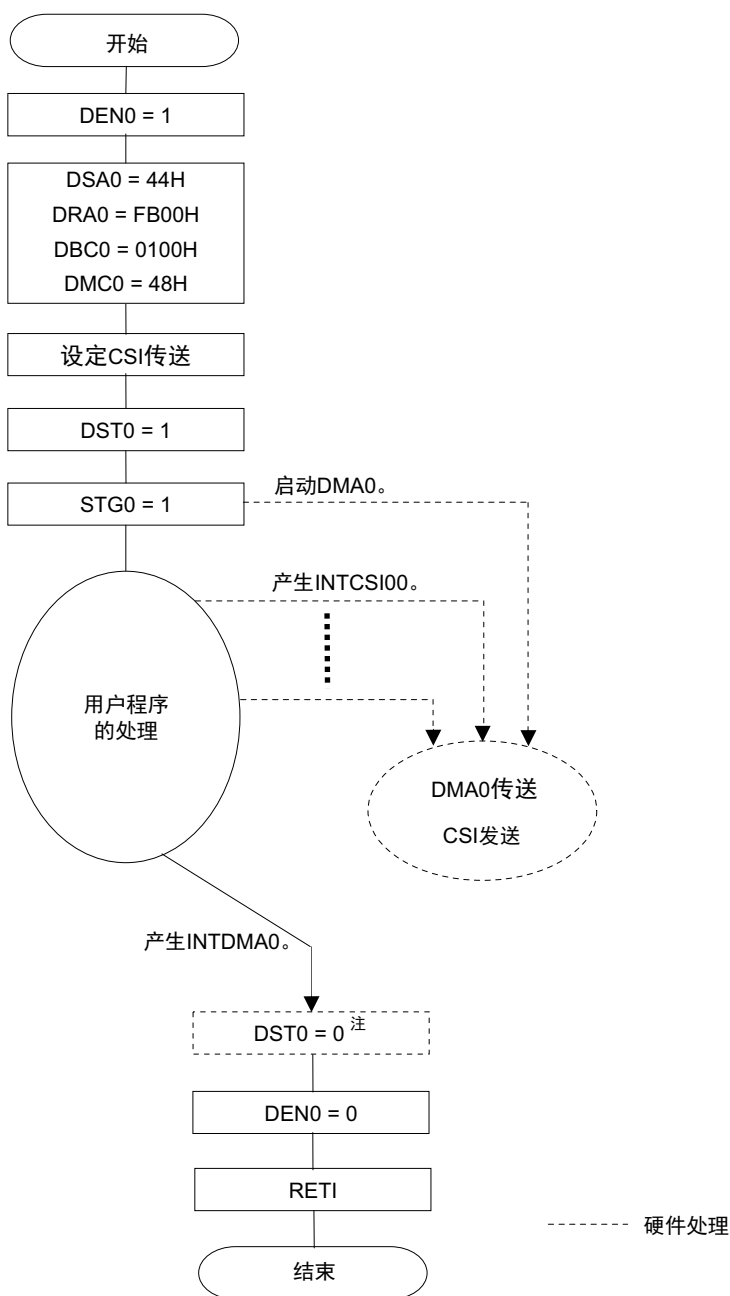
14.5.1 CSI 连续发送

CSI 连续发送的设定例子的流程图如下所示：

- CSI00 的连续发送 (256 字节)
- DMA 传送使用 DMA 的通道 0。
- DMA 启动源: INTCSI00 (软件触发 (STG0) 只限于最初的启动源)
- 通过 IFC03 ~ IFC00=0110B 指定 CSI00 的中断。
- 将 RAM 的 FFB00H ~ FFBFFH (256 字节) 传送到 CSI 的数据寄存器 (SIO00) 的 FFF10H。

备注 IFC03 ~ IFC00: DMA 模式控制寄存器 0 (DMC0) 的 bit3 ~ 0

图 14-7 CSI 连续发送的设定例子



注 如果 DMA 传送结束，就自动将 DST0 标志清“0”。

因为只有在 DST0 标志为“0”时才允许写 DEN0 标志，所以当不等待 DMA0 中断（INTDMA0）的发生而结束 DMA 传送时，必须在将 DST0 标志置“0”后将 DEN0 标志置“0”（详细内容请参照“14.5.5 通过软件强制结束”）。

在连续发送时，CSI 中断不能启动第一次触发。在此例子中，通过软件触发进行启动。

自动进行第二次及以后的 CSI 发送。

在给数据寄存器写完最后的发送数据时，产生 DMA 中断（INTDMA0）。

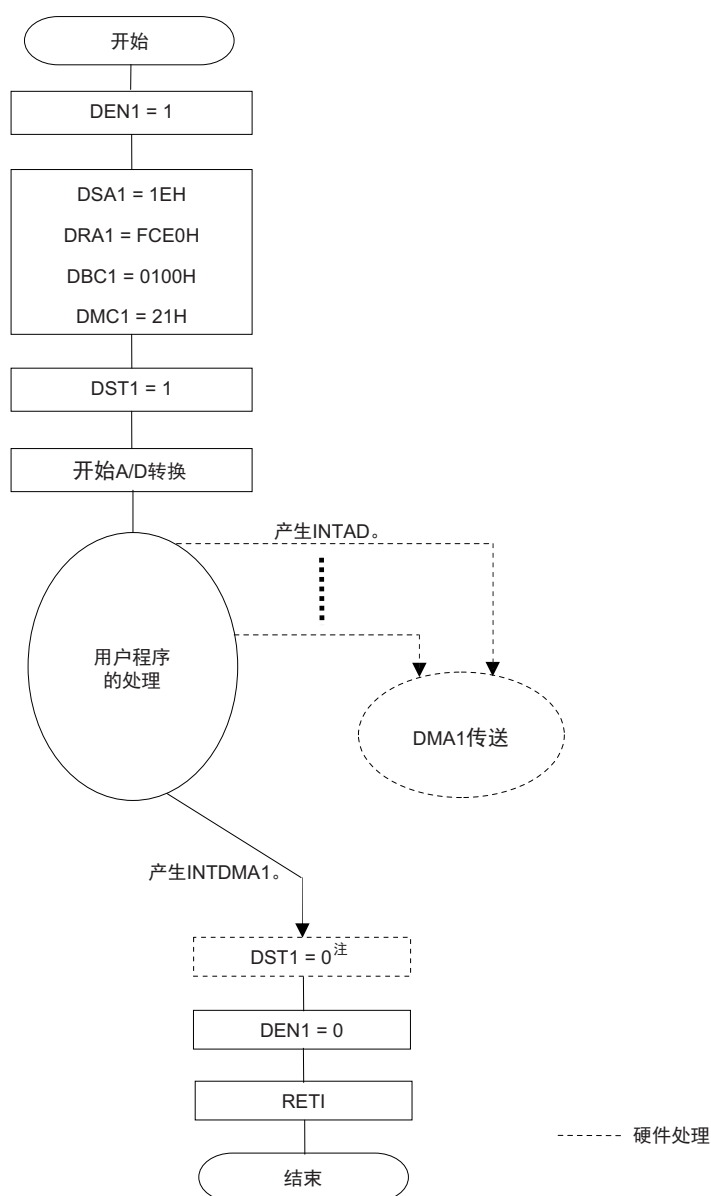
14.5.2 A/D 转换结果的连续读取

A/D 转换结果连续读取的设定例子的流程图如下所示：

- A/D 转换结果的连续读取
- DMA 传送使用 DMA 的通道 1。
- DMA 启动源：INTAD
- 通过 IFC13 ~ IFC10=0001B 指定 A/D 的中断。
- 将 10 位 A/D 转换结果寄存器（ADCR）的 FFF1EH 和 FFF1FH（2 字节）传送到 RAM 的 FFCE0H ~ FFEDFH（512 字节）。

备注 IFC13 ~ IFC10：DMA 模式控制寄存器 1（DMC1）的 bit3 ~ 0

图 14-8 A/D 转换结果连续读取的设定例子



注 如果 DMA 传送结束，就自动将 DST1 标志清“0”。

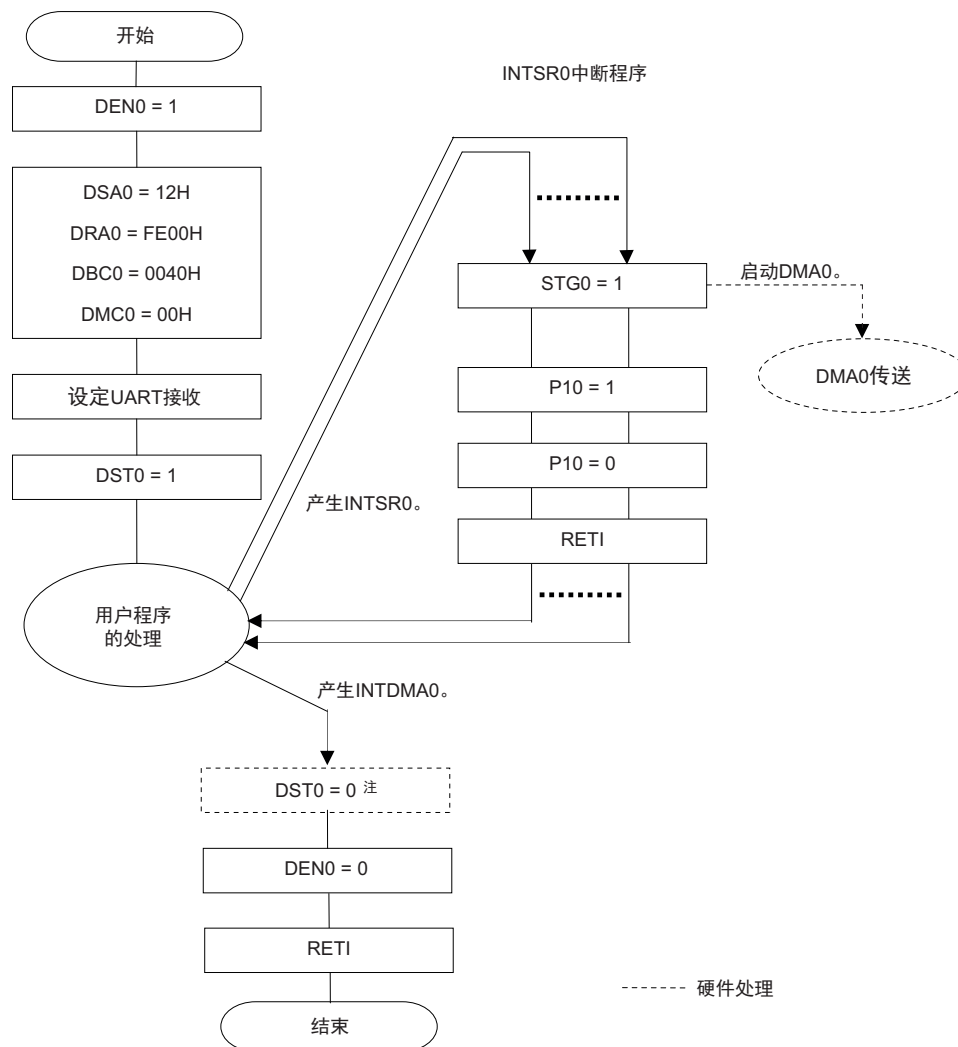
因为只有在 DST1 标志为“0”时才允许写 DEN1 标志，所以当不等待 DMA1 中断（INTDMA1）的发生而结束 DMA 传送时，必须在将 DST1 标志置“0”后将 DEN1 标志置“0”（详细内容请参照“14.5.5 通过软件强制结束”）。

14.5.3 UART 连续接收 +ACK 发送

UART 连续接收 +ACK 发送的设定例子的流程图如下所示：

- UART0 连续进行接收并且将接收结束的 ACK 输出到 P10。
- DMA 传送使用 DMA 的通道 0。
- DMA 启动源：软件触发（禁止通过中断触发 DMA 传送）
- 将 UART 接收数据寄存器 0（RxD0）的 FFF12H 传送到 RAM 的 FFE00H ~ FFE3FH（64 字节）。

图 14-9 UART 连续接收 +ACK 发送的设定例子



注 如果 DMA 传送结束，就自动将 DST0 标志清“0”。

因为只有在 DST0 标志为“0”时才允许写 DEN0 标志，所以当不等待 DMA0 中断（INTDMA0）的发生而结束 DMA 传送时，必须在将 DST0 标志置“0”后将 DEN0 标志置“0”（详细内容请参照“14.5.5 通过软件强制结束”）。

备注 这是将软件触发用于 DMA 启动源的例子。

如果不发送 ACK 而只进行 UART 的连续接收，就能将 UART 接收结束中断（INTSR0）设定为 DMA 启动源进行数据接收。

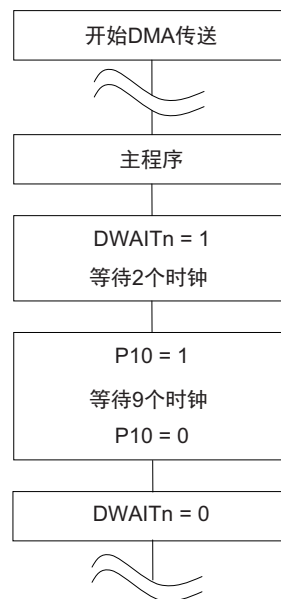
14.5.4 通过 DWAITn 位保留 DMA 传送

如果开始 DMA 传送，就在指令执行过程中进行传送。因此，此时 CPU 停止运行并且延迟 2 个时钟。如果这种情况会影响设备系统的正常运行，就能通过将 DWAITn 位置“1”，保留 DMA 传送。对于在保留期间发生的传送触发所对应的 DMA 传送，在解除保留后执行该 DMA 传送。但是，各通道只能保留一个传送触发，因此同一个通道即使在保留期间发生了 2 次或者更多次的传送触发，在解除保留后也只执行一次 DMA 传送。

以 P10 引脚输出 10 个时钟（工作频率）宽度的脉冲为例，如果在中途开始 DMA 传送，就增加到 12 个时钟。此时，能通过将 DWAITn 位置“1”来保留 DMA 传送。

在将 DWAITn 位置“1”后需要经过 2 个时钟，然后保留 DMA 传送。

图 14-10 通过 DWAITn 位保留 DMA 传送的设定例子



注意 要在使用 2 个或者更多通道的 DMA 过程中保留 DMA 传送时，必须保留全部通道的 DMA 传送（DWAIT0=DWAIT1=DWAIT2=DWAIT3=1）。如果在保留其中一个 DMA 传送期间执行另一个通道的 DMA 传送，就可能无法保留。

备注 1. n: DMA 通道号（n=0、1）

2. 1 个时钟： $1/f_{CLK}$ （ f_{CLK} : CPU 时钟）

14.5.5 通过软件强制结束

在通过软件将 DSTn 位置“0”后最多需要经过 2 个时钟，然后停止 DMA 传送并且 DSTn 位变为“0”。因此，当不等待 DMA 中断（INTDMA_n）的发生而通过软件强制结束 DMA 传送时，必须进行以下某种处理。

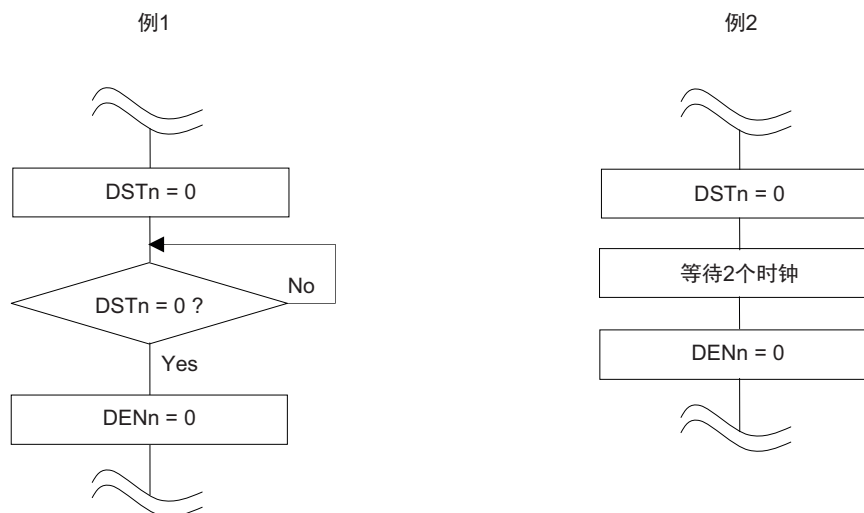
< 当使用 1 个通道的 DMA 时 >

- 在通过软件将 DSTn 位置“0”（当通过字节操作指令进行写操作时，DRCn=80H）后，通过轮询确认 DSTn 位已变为“0”，然后将 DENn 位置“0”（当通过字节操作指令进行写操作时，DRCn=00H）。
- 在通过软件将 DSTn 位置“0”（当通过字节操作指令进行写操作时，DRCn=80H）后经过 2 个时钟，然后将 DENn 位置“0”（当通过字节操作指令进行写操作时，DRCn=00H）。

< 当使用 2 个通道的 DMA 时 >

- 要在使用 2 个通道的 DMA 过程中通过软件强制结束 DMA 传送（DSTn=0）时，通过将要使用的 2 个通道的 DWAITn 位置“1”来保留 DMA 传送，然后将 DSTn 位清“0”。此后，通过将 2 个通道的 DWAITn 位清“0”来解除保留，然后将 DENn 位清“0”。

图 14-11 DMA 传送的强制结束 (1/2)



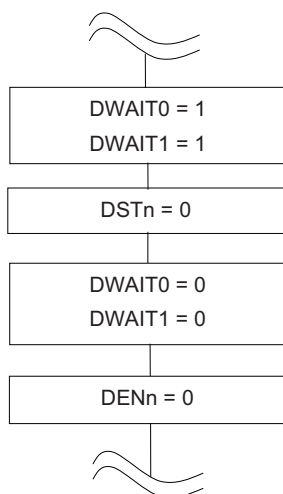
备注 1. n: DMA 通道号 (n=0、1)

2. 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

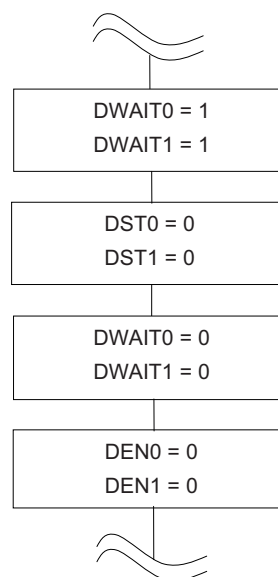
图 14-11 DMA 传送的强制结束 (2/2)

例3

- 在使用2个通道时，其中1个通道强制结束的步骤



- 在使用2个通道时，2个通道都强制结束的步骤



注意 在例 3 中，将 DWAITn 位置“1”后不需要等待 2 个时钟。因为在从 DSTn 位清“0”到 DENn 位清“0”的期间已经至少经过了 2 个时钟，所以在将 DSTn 位清“0”后不需要等待 2 个时钟。

备注 1. n: DMA 通道号 (n=0、1)

2. 1 个时钟: $1/f_{\text{CLK}}$ (f_{CLK} : CPU 时钟)

14.6 DMA 控制器的注意事项

(1) DMA 的优先级

在 DMA 传送过程中，即使发生其他 DMA 通道的请求，也保留该通道的请求。在 DMA 传送结束后开始被保留的 DMA 传送。但是，当多个 DMA 请求同时发生时，优先级为 DMA 通道 0 > DMA 通道 1 > DMA 通道 2 > DMA 通道 3。

当 DMA 请求和中断请求同时发生时，优先 DMA 传送，然后执行中断处理。

(2) DMA 应答时间

DMA 传送的应答时间如下所示。

表 14-3 DMA 传送的应答时间

	最短时间	最长时间
应答时间	3 个时钟	10 个时钟注

注 当从内部 RAM 执行指令时，最长时间为 16 个时钟。

注意 1. 以上的应答时间不包括 DMA 传送的 2 个时钟。

2. 在执行 DMA 保留指令（参照 14.6(4)）时，各条件的最长应答时间加上在其条件下保留指令的执行时间。
3. 最长应答时间 +1 个时钟内的同一个通道的连续传送触发可能被忽视，因此不能设定。

备注 1 个时钟： $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

(3) 待机模式中的运行

待机模式中的 DMA 控制器的运行如下所示。

表 14-4 待机模式中的 DMA 运行

状态	DMA 运行
HALT 模式	通常运行
STOP 模式	停止运行 当 DMA 传送和 STOP 指令发生竞争时，DMA 传送可能会遭到破坏，因此必须在执行 STOP 指令前停止 DMA。

(4) DMA 保留指令

即使发生 DMA 请求，也会在紧随着以下指令之后保留 DMA 传送。

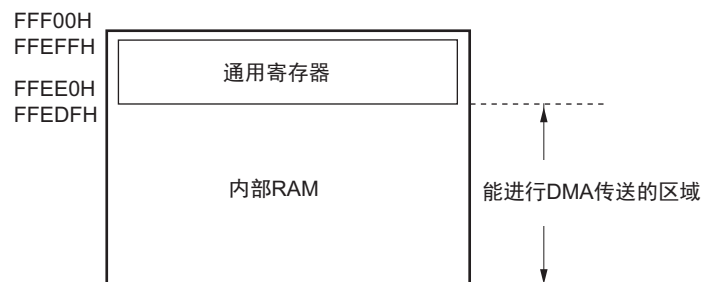
- CALL !addr16
- CALL \$!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW.bit, CY
- SET1 PSW.bit
- CLR1 PSW.bit
- POP PSW
- BTCLR PSW.bit, \$addr20
- EI
- DI
- IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H 和 PSW 的各寄存器的写指令
- 存取数据闪存的指令

(5) 指定通用寄存器区内或者内部 RAM 区外的地址时的运行

在 DMA 传送过程中，DMA RAM 地址寄存器 n (DRAn) 所示的地址进行递增。当该地址进入通用寄存器区内或者超过内部 RAM 区时，会出现以下情况。

- 从 SFR 传送到 RAM 的模式
破坏该地址的数据。
- 从 RAM 传送到 SFR 的模式
将不定的数据传送到 SFR。

以上 2 种情况都可能导致误动作或者系统的损坏，因此必须确保地址在通用寄存器区以外的内部 RAM 区内。



(6) 存取数据闪存空间时的运行

如果要在 1 条 DMA 传送指令之后存取数据闪存空间，就需要在此指令之间插入 3 个时钟的等待。

指令 1

DMA 传送

指令 2 ← 等待 3 个时钟。

MOV A, !DataFlash 空间

第 15 章 中断功能

中断功能是指在程序执行过程中将处理程序切换为其他所需处理程序的一种功能。当转移目标处理结束时，返回到原来中止的执行程序。

中断源的个数因产品而不同。

可屏蔽中断	外部	6
	内部	26

15.1 中断功能的种类

中断功能有以下 2 种。

(1) 可屏蔽中断

这是接受屏蔽控制的中断。通过设定优先级指定标志寄存器（PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H），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默认优先级进行处理。有关默认优先级，请参照表 15-1。

产生待机解除信号，解除 STOP 模式、HALT 模式和 SNOOZE 模式。

可屏蔽中断分为外部中断请求和内部中断请求。

(2) 软件中断

这是通过执行 BRK 指令而产生的向量中断。即使在禁止中断的状态下也接受软件中断，而且软件中断不受中断优先级控制。

15.2 中断源和结构

中断源除了有可屏蔽中断和软件中断以外还有最多 7 种复位源（参照表 15-1）。当发生复位或者各种中断请求而进行转移时，因为保存程序起始地址的向量码各为 2 字节，所以中断的转移目标地址为 00000H ~ 0FFFFH 的 64K 地址。

表 15-1 中断源一览表 (1/2)

中断处理	默认优先级注 1	中断源		内部/外部	向量表地址	基本结构类型注 2	
		名称	触发				
可屏蔽	0	INTWDTI	看门狗定时器的间隔注 3 (上溢时间的 75%+1/2f _{IL})	内部	00004H	(A)	○
	1	INTLVI	电压检测注 4		00006H		○
	2	INTP0	引脚输入边沿的检测	外部	00008H	(B)	○
	3	INTP1			0000AH		○
	4	INTP2			0000CH		○
	5	INTP3			0000EH		○
	6	INTP4			00010H		○
	7	INTP5			00012H		○
	8	INTST2/ INTCSI20/ INTIIC20			UART2 发送的传送结束或者缓冲器空中断 / CSI20 的传送结束或者缓冲器空中断 / IIC20 的传送结束		内部
	9	INTSR2	UART2 接收的传送结束	00016H	○		
	10	INTSRE2	发生 UART2 接收的通信错误	00018H	○		
	11	INTDMA0	DMA0 的传送结束	0001AH	○		
	12	INTDMA1	DMA1 的传送结束	0001CH	○		
	13	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	0001EH	○		
	14	INTSR0	UART0 接收的传送结束	00020H	○		
	15	INTSRE0	发生 UART0 接收的通信错误	00022H	○		
		INTTM01H	定时器通道 01 的计数结束或者捕捉结束 (高 8 位定时器工作时)		○		
	16	INTST1	UART1 发送的传送结束或者缓冲器空中断	00024H	○		
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 接收的传送结束 / CSI11 的传送结束或者缓冲器空中断 / IIC11 的传送结束	00026H	○		
	18	INTSRE1	发生 UART1 接收的通信错误	00028H	○		
		INTTM03H	定时器通道 03 的计数结束或者捕捉结束 (高 8 位定时器工作时)		○		
	19	INTIICA0	IICA0 通信结束	0002AH	○		
	20	INTTM00	定时器通道 00 的计数结束或者捕捉结束	0002CH	○		
21	INTTM01	定时器通道 01 的计数结束或者捕捉结束 (16 位 / 低 8 位定时器工作时)	0002EH	○			
22	INTTM02	定时器通道 02 的计数结束或者捕捉结束	00030H	○			
23	INTTM03	定时器通道 03 的计数结束或者捕捉结束 (16 位 / 低 8 位定时器工作时)	00032H	○			

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，31 表示最低优先级。

2. 基本构成类型 (A) ~ (D) 分别对应图 15-1 的 (A) ~ (D)。

3. 这是将选项字节 (000C0H) 的 bit7 (WDTINT) 置“1”的情况。

4. 这是将电压检测电平寄存器 (LVIS) 的 bit7 (LVIMD) 置“0”的情况。

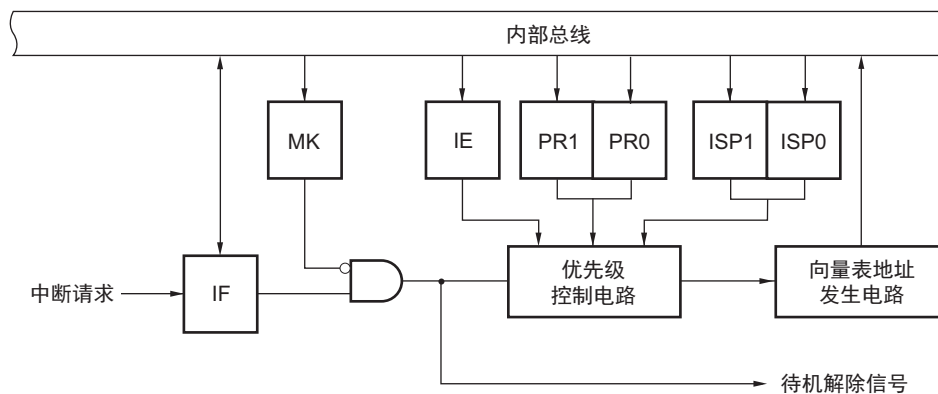
表 15-1 中断源一览表 (2/2)

中断处理	默认优先级注 1	中断源		内部 / 外部	向量表地址	基本结构类型注 2	
		名称	触发				
可屏蔽	24	INTAD	A/D 转换结束	内部	00034H	(A)	○
	25	INTIT	12 位间隔定时器的间隔信号检测		00038H		○
	26	INTTM04	定时器通道 04 的计数结束或者捕捉结束		00042H		○
	27	INTTM05	定时器通道 05 的计数结束或者捕捉结束		00044H		○
	28	INTTM06	定时器通道 06 的计数结束或者捕捉结束		00046H		○
	29	INTTM07	定时器通道 07 的计数结束或者捕捉结束		00048H		○
	30	INTMD	除法运算结束 / 乘法运算结果发生上溢		0005EH		○
	31	INTFL	保留注 3		00062H		○
软件	—	BRK	BRK 指令的执行	—	0007EH	(D)	○
复位	—	RESET	RESET 引脚的输入	—	00000H	—	○
		POR	上电复位				○
		LVD	电压检测注 4				○
		WDT	看门狗定时器的上溢				○
		TRAP	非法指令的执行注 5				○
		IAW	非法存储器的存取				○
		RPE	RAM 奇偶校验错误				○

- 注 1. 在发生多个可屏蔽中断时，默认优先级决定优先顺序。0 表示最高优先级，31 表示最低优先级。
2. 基本构成类型 (A) ~ (D) 分别对应图 15-1 的 (A) ~ (D)。
3. 用于闪存自编程库或者数据闪存库。
4. 这是将电压检测电平寄存器 (LVIS) 的 bit7 (LVIMD) 置“1”的情况。
5. 在执行指令码 FFH 时发生。
在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

图 15-1 中断功能的基本结构

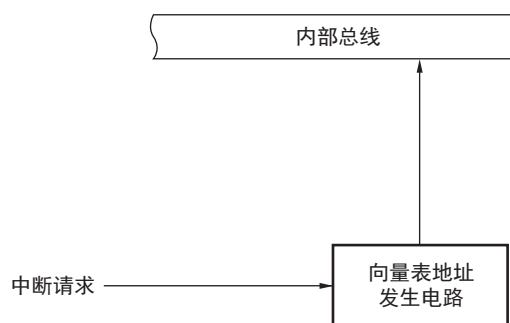
(A) 内部可屏蔽中断



- IF: 中断请求标志
 IE: 中断允许标志
 ISP0: 服务优先级标志 0
 ISP1: 服务优先级标志 1
 MK: 中断屏蔽标志
 PR0: 优先级指定标志 0
 PR1: 优先级指定标志 1

备注 n=0 ~ 5

(B) 软件中断



15.3 控制中断功能的寄存器

通过以下 6 种寄存器控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)
- 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)
- 外部中断上升沿允许寄存器 (EGP0)
- 外部中断下降沿允许寄存器 (EGN0)
- 程序状态字 (PSW)

各中断请求源对应的中断请求标志、中断屏蔽标志和优先级指定标志的名称如表 15-2 所示。

表 15-2 中断请求源对应的各种标志 (1/3)

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	寄存器	寄存器	寄存器	
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0、WDTIPR1	PR00L、 PR10L	○
INTLVI	LVIIIF		LVIMK		LVIPR0、LVIPR1		○
INTP0	PIF0		PMK0		PPR00、PPR10		○
INTP1	PIF1		PMK1		PPR01、PPR11		○
INTP2	PIF2		PMK2		PPR02、PPR12		○
INTP3	PIF3		PMK3		PPR03、PPR13		○
INTP4	PIF4		PMK4		PPR04、PPR14		○
INTP5	PIF5		PMK5		PPR05、PPR15		○

表 15-2 中断请求源对应的各种标志 (2/3)

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	
INTST2 注 1	STIF2 注 1	IF0H	STMK2 注 1	MK0H	STPR02、STPR12 注 1	PR00H、 PR10H	○
INTCSI20 注 1	CSIIF20 注 1		CSIMK20 注 1		CSIPR020、CSIPR120 注 1		○
INTIIC20 注 1	IICIF20 注 1		IICMK20 注 1		IICPR020、IICPR120 注 1		○
INTSR2	SRIF2		SRMK2		SRPR02、SRPR12		○
INTSRE2	SREIF2		SREMK2		SREPR02、SREPR12		○
INTDMA0	DMAIF0		DMAMK0		DMAPR00、DMAPR10		○
INTDMA1	DMAIF1		DMAMK1		DMAPR01、DMAPR11		○
INTST0 注 2	STIF0 注 2		STMK0 注 2		STPR00、STPR10 注 2		○
INTCSI00 注 2	CSIIF00 注 2		CSIMK00 注 2		CSIPR000、CSIPR100 注 2		○
INTIIC00 注 2	IICIF00 注 2		IICMK00 注 2		IICPR000、IICPR100 注 2		○
INTSR0	SRIF0		SRMK0		SRPR00、SRPR10		○
INTSRE0 注 3	SREIF0 注 3		SREMK0 注 3		SREPR00、SREPR10 注 3		○
INTTM01H 注 3	TMIF01H 注 3		TMMK01H 注 3		TMPR001H、TMPR101H 注 3		○

- 注 1. 如果发生 INTST2、INTCSI20 或者 INTIIC20 的中断源，IF0H 寄存器的 bit0 就被置“1”。MK0H、PR00H、PR10H 寄存器的 bit0 对应这 3 个中断源。
2. 如果发生 INTST0、INTCSI00 或者 INTIIC00 的中断源，IF0H 寄存器的 bit5 就被置“1”。MK0H、PR00H、PR10H 寄存器的 bit5 对应这 3 个中断源。
3. 因为 UART0 接收的错误中断和 TAU0 的通道 1（高 8 位定时器工作时）的中断复用中断请求源对应的各种标志，所以不能同时使用。在不使用 UATR0 接收的错误中断（EOC01=0）时，可同时使用 UART0、TAU0 的通道 1（高 8 位定时器工作时）。如果发生 INTSRE0 或者 INTTM01H 的中断源，IF0H 寄存器的 bit7 就被置“1”。MK0H、PR00H、PR10H 寄存器的 bit7 对应这 2 个中断源。

表 15-2 中断请求源对应的各种标志 (3/3)

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	
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01、STPR11	PR01L、 PR11L	○
INTSR1 注 1	SRIF1 注 1		SRMK1 注 1		SRPR01、SRPR11 注 1		○
INTCSI11 注 1	CSIIF11 注 1		CSIMK11 注 1		CSIPR011、CSIPR111 注 1		○
INTIIC11 注 1	IICIF11 注 1		IICMK11 注 1		IICPR011、IICPR111 注 1		○
INTSRE1 注 2	SREIF1 注 2		SREMK1 注 2		SREPR01、SREPR11 注 2		○
INTTM03H 注 2	TMIF03H 注 2		TMMK03H 注 2		TMPR003H、TMPR103H 注 2		○
INTIICA0	IICAIF0		IICAMK0		IICAPR00、IICAPR10		○
INTTM00	TMIF00		TMMK00		TMPR000、TMPR100		○
INTTM01	TMIF01		TMMK01		TMPR001、TMPR101		○
INTTM02	TMIF02		TMMK02		TMPR002、TMPR102		○
INTTM03	TMIF03		TMMK03		TMPR003、TMPR103		○
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0、ADPR1	PR01H、 PR11H	○
INTIT	ITIF		ITMK		ITPR0、ITPR1		○
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005、TMPR105	PR02L、 PR12L	○
INTTM06	TMIF06		TMMK06		TMPR006、TMPR106		○
INTTM07	TMIF07		TMMK07		TMPR007、TMPR107		○
INTMD	MDIF	IF2H	MDMK	MK2H	MDPR0、MDPR1	PR02H、 PR12H	○
INTFL	FLIF		FLMK		FLPR0、FLPR1		○

- 注 1. 如果发生 INTSR1、INTCSI11 或者 INTIIC11 的中断源，就将 IF1L 寄存器的 bit1 置“1”。MK1L、PR01L、PR11L 寄存器的 bit1 对应这 3 个中断源。
2. 因为 UART1 接收的错误中断和 TAU0 的通道 3（高 8 位定时器工作时）的中断共用中断请求源对应的各种标志，所以不能同时使用。当不使用 UART1 接收的错误中断（EOC03=0）时，能同时使用 UART1 和 TAU0 的通道 3（高 8 位定时器工作时）。如果发生 INTSRE1 或者 INTTM03H 的中断源，就将 IF1L 寄存器的 bit2 置“1”。MK1L、PR01L、PR11L 寄存器的 bit2 对应这 2 个中断源。

15.3.1 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)

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、IF1H、IF2L、IF2H 寄存器。当将 IF0L 寄存器和 IF0H 寄存器、IF1L 寄存器和 IF1H 寄存器以及 IF2L 寄存器和 IF2H 寄存器一起用作 IF0、IF1、IF2 的 16 位寄存器时，通过 16 位存储器操作指令进行设定。

在产生复位信号后，这些寄存器的值变为“00H”。

备注 当执行写此寄存器的指令时，指令执行时钟数增加 2 个时钟。

图 15-2 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H) 的格式 (1/2)

地址: FFFE0H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF	

地址: FFFE1H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
IF0H	SREIF0 TMIF01H	SRIF0	STIF0 CSIF00 IICIF00	DMAIF1	DMAIF0	SREIF2	SRIF2	STIF2 CSIF20 IICIF20	

地址: FFFE2H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF0	SREIF1 TMIF03H	SRIF1 CSIF11 IICIF11	STIF1	

地址: FFFE3H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
IF1H	0	0	0	0	0	ITIF	0	ADIF	

地址: FFFD0H	复位后: 00H	R/W							
符号	7	6	5	4	3	2	1	0	
IF2L	0	0	0	0	0	TMIF07	TMIF06	TMIF05	

图 15-2 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H) 的格式 (2/2)

地址: FFFD1H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IF2H	FLIF	0	MDIF	0	0	0	0	0

XXIFX	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. 分配的寄存器和位因产品而不同。有关各产品分配的寄存器和位, 请参照表 15-2。必须给未分配的位设定初始值。
2. 在操作中中断请求标志寄存器的标志时, 必须使用位存储器操作指令 (CLR1)。当使用 C 语言描述时, 编译后的汇编程序需要变为位存储器操作指令 (CLR1), 因此必须使用类似 “IF0L.0=0;” 或者 “_asm(“clr1 IF0L, 0”);” 的位操作指令。
- 如果在 C 语言中使用类似于 “IF0L &= 0xfe;” 的 8 位存储器操作指令描述的情况下进行编译, 就变为 3 条指令的汇编程序。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

此时, 即使在 “mov a, IF0L” 与 “mov IF0L, a” 之间将同一中断请求标志寄存器 (IF0L) 的其他位的请求标志置 “1”, 该标志也会被 “mov IF0L, a” 清 “0”。因此, 在 C 语言中使用 8 位存储器操作指令时, 必须注意。

15.3.2 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)

中断屏蔽标志设定允许或者禁止对应的可屏蔽中断处理。

通过 1 位或者 8 位存储器操作指令设定 MK0L、MK0H、MK1L、MK1H、MK2L、MK2H 寄存器。当将 MK0L 寄存器和 MK0H 寄存器、MK1L 寄存器和 MK1H 寄存器以及 MK2L 寄存器和 MK2H 寄存器一起用作 MK0、MK1、MK2 的 16 位寄存器时, 通过 16 位存储器操作指令进行设定。

在产生复位信号后, 这些寄存器的值变为 “FFH”。

备注 当执行写此寄存器的指令时, 指令执行时钟数增加 2 个时钟。

图 15-3 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H) 的格式

地址: FFFE4H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

地址: FFFE5H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK0H	SREMK0 TMMK01H	SRMK0	STMK0 CSIMK00 IICMK00	DMAMK1	DMAMK0	SREMK2	SRMK2	STMK2 CSIMK20 IICMK20

地址: FFFE6H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK0	SREMK1 TMMK03H	SRMK1 CSIMK11 IICMK11	STMK1

地址: FFFE7H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK1H	0	0	0	0	0	ITMK	0	ADMK

地址: FFFD4H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK2L	0	0	0	0	0	TMMK07	TMMK06	TMMK05

地址: FFFD5H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
MK2H	FLMK	0	MDMK	0	0	0	0	0

XXMKX	中断处理的控制
0	允许中断处理。
1	禁止中断处理。

注意 分配的寄存器和位因产品而不同。有关各产品分配的寄存器和位，请参照表 15-2。必须给未分配的位设定初始值。

15.3.3 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)

优先级指定标志寄存器设定对应的可屏蔽中断优先级。

通过组合 PR0xy 寄存器和 PR1xy 寄存器来设定优先级 (xy=0L、0H、1L、1H、2L、2H)。

通过 1 位或者 8 位存储器操作指令设定 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H 寄存器。当将 PR00L 寄存器和 PR00H 寄存器、PR01L 寄存器和 PR01H 寄存器、PR02L 寄存器和 PR02H 寄存器、PR10L 寄存器和 PR10H 寄存器、PR11L 寄存器和 PR11H 寄存器以及 PR12L 寄存器和 PR12H 寄存器一起用作 PR00、PR01、PR02、PR10、PR11、PR12 的 16 位寄存器时，通过 16 位存储器操作指令进行设定。

在产生复位信号后，这些寄存器的值变为“FFH”。

备注 当执行写此寄存器的指令时，指令执行时钟数增加 2 个时钟。

图 15-4 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H) 的格式 (1/2)

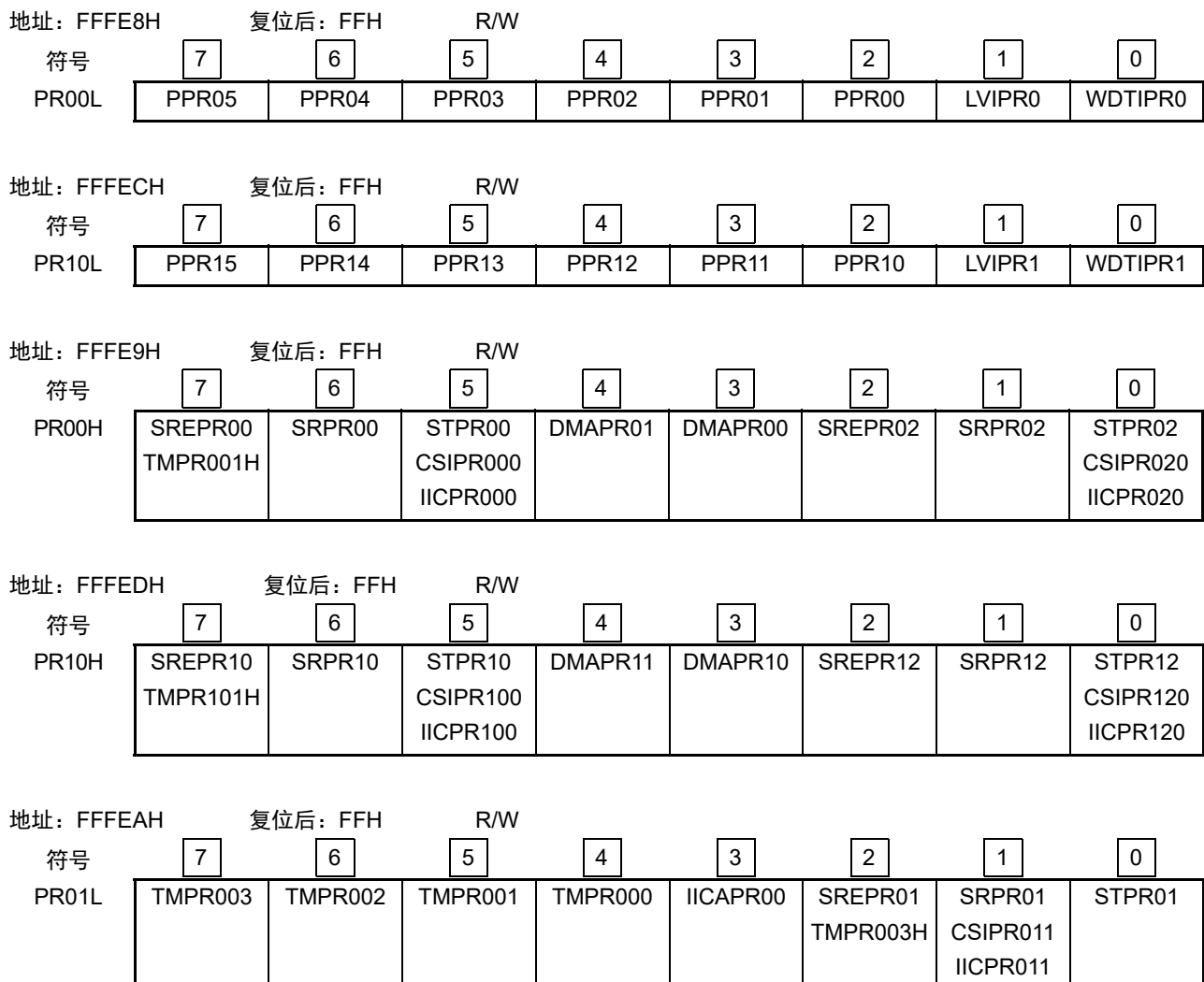


图 15-4 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H) 的格式 (2/2)

地址: FFFEEH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR10	SREPR11 TMPR103H	SRPR11 CSIPR111 IICPR111	STPR11

地址: FFFEBH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR01H	1	1	1	1	1	ITPR0	1	ADPR0

地址: FFFEFH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR11H	1	1	1	1	1	ITPR1	1	ADPR1

地址: FFFD8H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR02L	1	1	1	1	1	TMPR007	TMPR006	TMPR005

地址: FFFDCH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR12L	1	1	1	1	1	TMPR107	TMPR106	TMPR105

地址: FFFD9H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR02H	FLPR0	1	MDPR0	1	1	1	1	1

地址: FFFDDH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PR12H	FLPR1	1	MDPR1	1	1	1	1	1

XXPR1X	XXPR0X	优先级的选择
0	0	指定优先级 0 (高优先级)。
0	1	指定优先级 1。
1	0	指定优先级 2。
1	1	指定优先级 3 (低优先级)。

15.3.4 外部中断上升沿允许寄存器（EGP0）、外部中断下降沿允许寄存器（EGN0）

这些寄存器设定 INTP0 ~ INTP5 的有效边沿。

通过 1 位或者 8 位存储器操作指令设定 EGP0 寄存器和 EGN0 寄存器。

在产生复位信号后，这些寄存器的值变为“00H”。

图 15-5 外部中断上升沿允许寄存器（EGP0）和外部中断下降沿允许寄存器（EGN0）的格式

地址: FFF38H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FFF39H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚的有效边沿选择 (n=0 ~ 5)
0	0	禁止检测边沿。
0	1	下降沿
1	0	上升沿
1	1	上升和下降的双边沿

对应 EGPn 位和 EGNn 位的端口如表 15-3 所示。

表 15-3 对应 EGPn 位和 EGNn 位的中断请求信号

检测允许位		中断请求信号	
EGP0	EGN0	INTP0	○
EGP1	EGN1	INTP1	○
EGP2	EGN2	INTP2	○
EGP3	EGN3	INTP3	○
EGP4	EGN4	INTP4	○
EGP5	EGN5	INTP5	○

注意 如果将外部中断功能使用的输入端口切换到输出模式，就可能检测到有效边沿而产生 INTPn 中断。

当切换到输出模式时，必须在禁止检测边沿后（EGPn、EGNn=0、0）将端口模式寄存器（PMxx）置“0”。

备注 1. 有关边沿检测的端口，请参照“2.1 端口功能”。

2. n=0 ~ 5

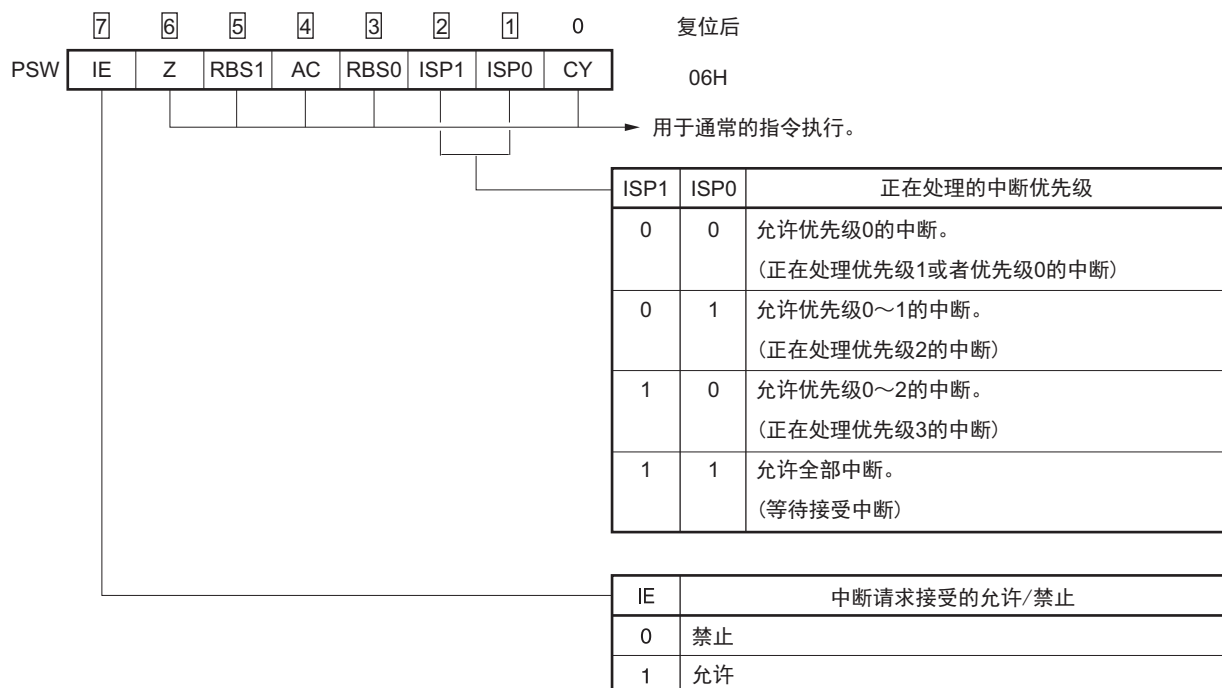
15.3.5 程序状态字 (PSW)

程序状态字是保存指令执行结果和中断请求当前状态的寄存器，映射设定允许或者禁止可屏蔽中断的 IE 标志以及控制多重中断处理的 ISP0 标志和 ISP1 标志。

除了能以 8 位为单位读写 PSW 以外，还能利用位操作指令和专用指令 (EI 和 DI) 操作 PSW。在接受向量中断请求及执行 BRK 指令时，自动将 PSW 的内容压栈，并且将 IE 标志恢复为“0”。在接受可屏蔽中断请求时，如果接受的中断优先级指定标志寄存器的内容不为“00”，就将其减 1 的值传送到 ISP0 标志和 ISP1 标志。PSW 的内容也能通过 PUSH PSW 指令进行压栈，通过 RETI、RETB、POP PSW 指令进行退栈。

在产生复位信号后，PSW 的值变为“06H”。

图 15-6 程序状态字的结构



15.4 中断处理的操作

15.4.1 可屏蔽中断请求的接受

如果中断请求标志被置“1”并且该中断请求的屏蔽（MK）标志已被清“0”，就进入能接受可屏蔽中断请求的状态，在允许中断的状态（IE 标志已被置“1”）下接受向量中断请求。但是，在处理优先级高的中断请求的过程中，不接受被指定的低优先级的中断请求。

从产生可屏蔽中断请求到进行向量中断处理的时间如表 15-4 所示。

有关中断请求的接受时序，请参照图 15-8 和图 15-9。

表 15-4 从产生可屏蔽中断请求到处理的时间

	最短时间	最长时间注
处理时间	9 个时钟	16 个时钟

注 从内部 RAM 区执行指令时除外。

备注 1 个时钟： $1/f_{\text{CLK}}$ (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。

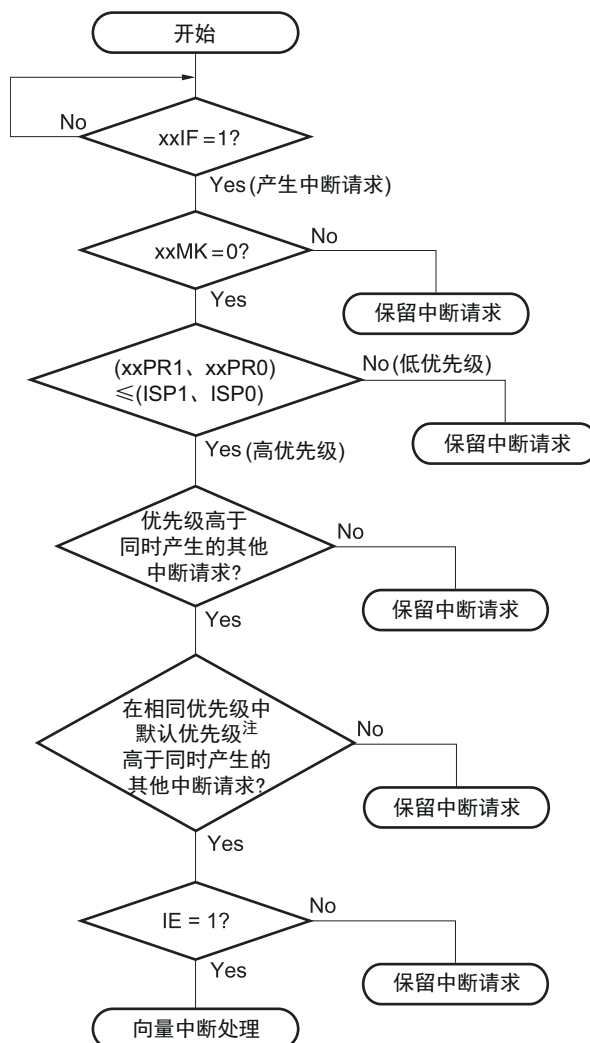
被保留的中断请求在可接受的状态时被接受。

中断请求的接受算法如图 15-7 所示。

如果接受可屏蔽中断请求，就按照程序状态字（PSW）、程序计数器（PC）的顺序将内容压栈，将 IE 标志复位为“0”，并且将接受的中断优先级指定标志的内容传送到 ISP1 标志和 ISP0 标志。将各中断请求决定的向量表中的数据装入 PC 并且进行转移。

能使用 RETI 指令从中断返回。

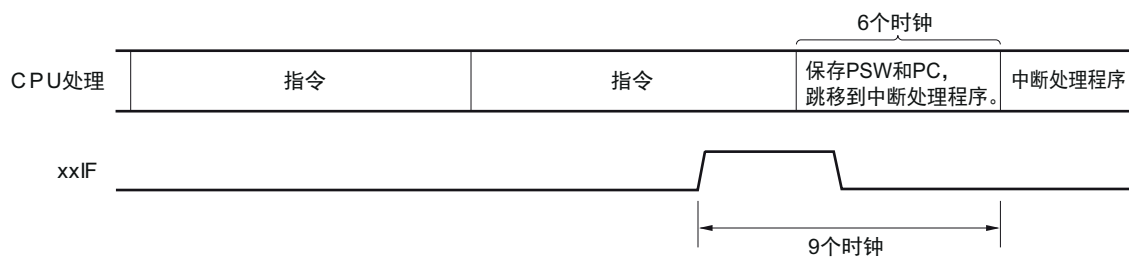
图 15-7 接受中断请求的处理算法



- xxIF: 中断请求标志
 xxMK: 中断屏蔽标志
 xxPR0: 优先级指定标志 0
 xxPR1: 优先级指定标志 1
 IE: 可屏蔽中断请求的接受控制标志 (1= 允许、0= 禁止)
 ISP0、ISP1: 表示正在处理的中断优先级的标志 (参照图 15-6)

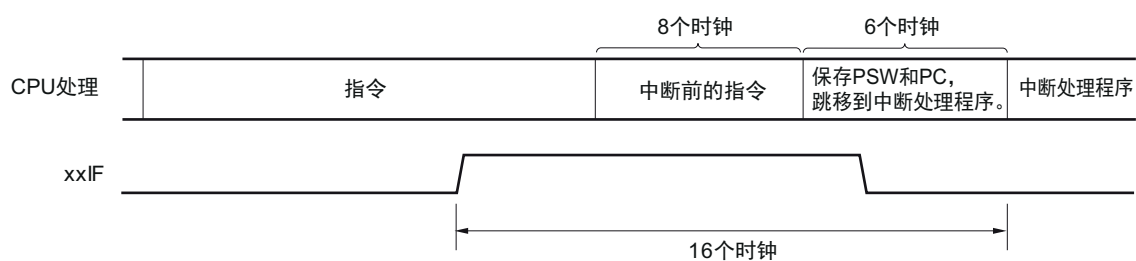
注 有关默认优先级, 请参照“表 15-1 中断源一览表”。

图 15-8 中断请求的接受时序（最短时间）



备注 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

图 15-9 中断请求的接受时序（最长时间）



备注 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

15.4.2 软件中断请求的接受

通过执行 BRK 指令接受软件中断请求，不能禁止软件中断。

如果接受软件中断请求，就按照程序状态字（PSW）、程序计数器（PC）的顺序将内容压栈，将 IE 标志复位为“0”，并且将向量表（0007EH、0007FH）的内容装入 PC 进行转移。

能使用 RETB 指令从软件中断返回。

注意 不能使用 RETI 指令从软件中断返回。

15.4.3 多重中断处理

在中断处理过程中接受了另一个中断请求的情况称为多重中断。

除非中断请求为接受允许状态（IE=1），否则不会发生多重中断。在接受中断请求时，中断请求为接受被禁止状态（IE=0）。因此，如果要允许多重中断，就需要在中断处理过程中用 EI 指令将 IE 标志置“1”，进入允许中断的状态。

另外，即使在允许中断的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。

在允许中断的状态下，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。

在中断处理过程中能发生多重中断的中断请求的关系和多重中断的例子如表 15-5 和图 15-10 所示。

表 15-5 在中断处理过程中能发生多重中断的中断请求的关系

多重中断请求		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. ○：能发生多重中断。

2. ×：不能发生多重中断。

3. ISP0、ISP1 和 IE 是 PSW 中的标志。

ISP1=0、ISP0=0：正在处理优先级 1 或者优先级 0 的中断。

ISP1=0、ISP0=1：正在处理优先级 2 的中断。

ISP1=1、ISP0=0：正在处理优先级 3 的中断。

ISP1=1、ISP0=1：等待接受中断（允许全部中断）。

IE=0：禁止接受中断请求。

IE=1：允许接受中断请求。

4. PR 是 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H 寄存器中的标志。

PR=00：通过 xxPR1x=0、xxPR0x=0 指定优先级 0（高优先级）。

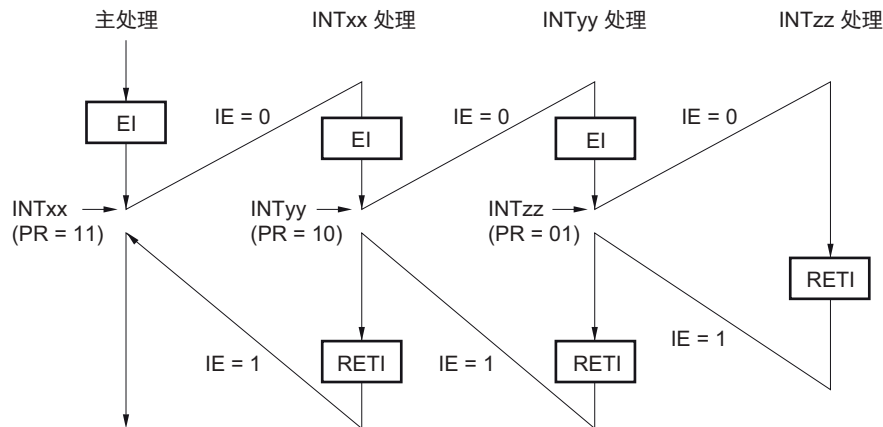
PR=01：通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

PR=10：通过 xxPR1x=1、xxPR0x=0 指定优先级 2。

PR=11：通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

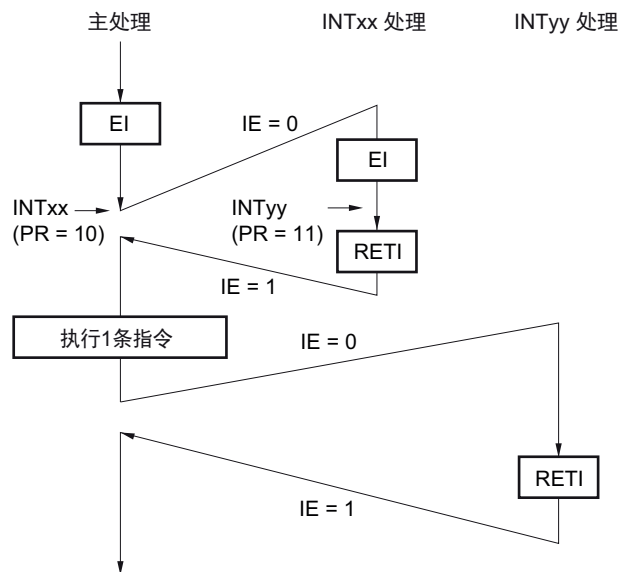
图 15-10 多重中断的例子 (1/2)

例 1. 发生 2 次多重中断的例子



在处理中断 INTxx 的过程中接受 INTyy 和 INTzz 的 2 个中断请求，发生多重中断。必须在接受各中断请求之前发行 EI 指令，使中断请求处于允许接受的状态。

例 2. 因控制优先级而不发生多重中断的例子



在处理中断 INTxx 的过程中，发生的 interrupt 请求 INTyy 因中断优先级低于 INTxx 而不被接受，所以不发生多重中断。保留 INTyy 请求，在执行一条主处理的指令后接受被保留的 INTyy 请求。

PR=00: 通过 xxPR1x=0、xxPR0x=0 指定优先级 0（高优先级）。

PR=01: 通过 xxPR1x=0、xxPR0x=1 指定优先级 1。

PR=10: 通过 xxPR1x=1、xxPR0x=0 指定优先级 2。

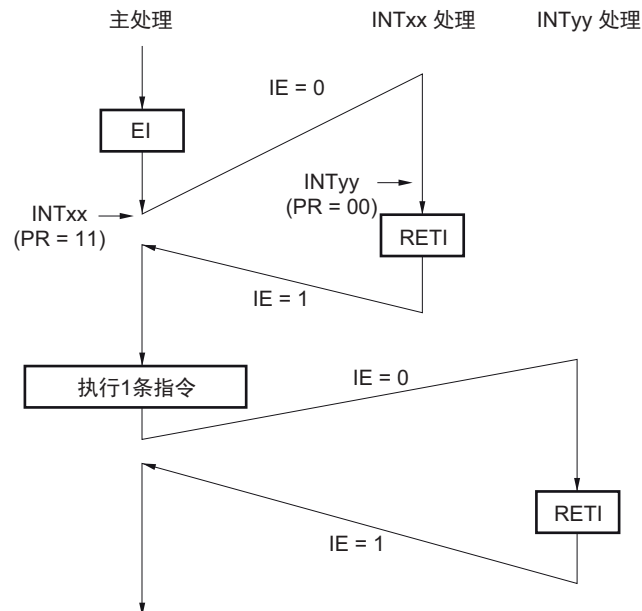
PR=11: 通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。

IE=0: 禁止接受中断请求。

IE=1: 允许接受中断请求。

图 15-10 多重中断的例子 (2/2)

例 3. 因不允许中断而不发生多重中断的例子



在处理中断 INTxx 的过程中，因不允许中断（未发行 EI 指令）而不接受中断请求 INTyy，所以不发生多重中断。保留 INTyy 请求，在执行一条主处理的指令后接受被保留的 INTyy 请求。

- PR=00: 通过 xxPR1x=0、xxPR0x=0 指定优先级 0（高优先级）。
- PR=01: 通过 xxPR1x=0、xxPR0x=1 指定优先级 1。
- PR=10: 通过 xxPR1x=1、xxPR0x=0 指定优先级 2。
- PR=11: 通过 xxPR1x=1、xxPR0x=1 指定优先级 3（低优先级）。
- IE=0: 禁止接受中断请求。
- IE=1: 允许接受中断请求。

15.4.4 中断请求的保留

有些指令，即使在执行指令的过程中发生了中断请求，也可能将中断请求保留到下一条指令执行结束为止。这些指令（中断请求保留指令）如下所示。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L和PR12H的各寄存器的写指令

中断请求的保留时序如图 15-11 所示。

图 15-11 中断请求的保留



备注 1. 指令 N: 中断请求的保留指令

2. 指令 M: 中断请求保留指令以外的指令

第 16 章 待机功能

16.1 待机功能

待机功能是进一步降低系统工作电流的功能，有以下 3 种模式。

(1) HALT 模式

通过执行 HALT 指令进入 HALT 模式。HALT 模式是停止 CPU 运行时钟的模式。在设定 HALT 模式前，如果高速系统时钟振荡电路或者高速内部振荡器正在振荡，各时钟就继续振荡。虽然此模式无法让工作电流降到 STOP 模式的程度，但是在想要通过中断请求立即重新开始处理或者想要频繁地进行间歇运行时是一种有效的模式。

(2) STOP 模式

通过执行 STOP 指令进入 STOP 模式。STOP 模式是停止高速系统时钟振荡电路和高速内部振荡器的振荡并且停止整个系统的模式。能大幅度地减小 CPU 的工作电流。

因为 STOP 模式能通过中断请求来解除，所以也能进行间歇运行。但是，在 X1 时钟的情况下，因为在解除 STOP 模式时需要确保振荡稳定的等待时间，所以如果一定要通过中断请求立即开始处理，就必须选择 HALT 模式。

(3) SNOOZE 模式

通过 CSIp 或者 UARTq 的数据接收以及由定时器触发信号（中断请求信号（INTIT））产生的 A/D 转换请求，解除 STOP 模式，并且在 CPU 不运行的状态下进行 CSIp 或者 UARTq 的数据接收以及 A/D 转换。只有在选择高速内部振荡器作为 CPU/ 外围硬件时钟（ f_{CLK} ）时才能设定 SNOOZE 模式。

在任何一种模式中，寄存器、标志和数据存储器全部保持设定为待机模式前的内容，并且还保持输入 / 输出端口的输出锁存器和输出缓冲器的状态。

注意 1. 只有在 CPU 以主系统时钟运行时才能使用 STOP 模式和 HALT 模式。

2. 在转移到 STOP 模式时，必须在停止以主系统时钟运行的外围硬件后执行 STOP 指令（SNOOZE 模式设定单元除外）。
3. 在 SNOOZE 模式中使用 CSIp、UARTq 或者 A/D 转换器时，必须在转移到 STOP 模式前设定串行待机控制寄存器 0（SSC0）和 A/D 转换器模式寄存器 2（ADM2）。详细内容请参照“11.3 控制串行阵列单元的寄存器”和“10.3 控制 A/D 转换器的寄存器”。
4. 为了减小 A/D 转换器的工作电流，必须将 A/D 转换器模式寄存器 0（ADM0）的 bit7（ADCS）和 bit0（ADCE）清“0”，在停止 A/D 转换运行后执行 STOP 指令。
5. 能通过选项字节选择在 HALT 模式或者 STOP 模式中是继续还是停止低速内部振荡器的振荡。详细内容请参照“第 22 章 选项字节”。

备注 P=00、q=0

16.2 控制待机功能的寄存器

控制待机功能的寄存器如下所示：

- 副系统时钟提供模式控制寄存器（OSMC）
- 振荡稳定时间计数器的状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

备注 有关上述寄存器的详细内容，请参照“第5章 时钟发生电路”。有关控制SNOOZE模式功能的寄存器，请参照“第10章 A/D转换器”和“第11章 串行阵列单元”。

16.3 待机功能的运行

16.3.1 HALT 模式

(1) HALT 模式

通过执行 HALT 指令设定为 HALT 模式。无论设定前的 CPU 时钟是高速系统时钟还是高速内部振荡器时钟，都能设定。

HALT 模式中的运行状态如下所示。

注意 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除 HALT 模式。因此，即使在此情况下执行 HALT 指令，也不转移到 HALT 模式。

表 16-1 HALT 模式中的运行状态

HALT 模式的设定		在 CPU 以主系统时钟运行的过程中执行 HALT 指令的情况					
		CPU 以高速内部振荡器时钟 (f_{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) 以及副系统时钟提供模式控制寄存器 (OSMC) 的 WUTMMCK0 位进行设定。 <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 (在执行 DMA 时, 能运行)。					
数据闪存							
RAM		保持设定为 HALT 模式前的状态。					
端口 (锁存器)		能运行。					
定时器阵列单元		参照“第 9 章 看门狗定时器”。					
12 位间隔定时器							
看门狗定时器		能运行。					
时钟输出 / 蜂鸣器输出		在 RAM 区的运算中执行 DMA 时, 能运行。					
A/D 转换器							
串行阵列单元 (SAU)							
串行接口 (IICA)							
乘除器和乘加器							
DMA 控制器							
上电复位功能							
电压检测功能							
外部中断							
CRC 运算功能	高速 CRC				在 RAM 区的运算中执行 DMA 时, 能运行。		
	通用 CRC						
RAM 奇偶校验功能					在执行 DMA 时, 能运行。		
RAM 保护功能					非法存储器存取检测功能		
SFR 保护功能							
非法存储器存取检测功能							

备注 停止运行: 在转移到 HALT 模式时自动停止运行。

禁止运行: 在转移到 HALT 模式前停止运行。

f_{IH} : 高速内部振荡器时钟 f_{IL} : 低速内部振荡器时钟

f_X : X1 时钟 f_{EX} : 外部主系统时钟

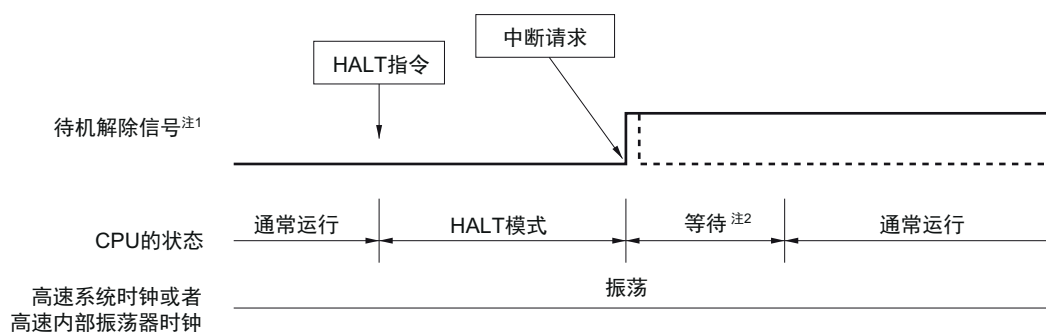
(2) HALT 模式的解除

能通过以下 2 种方法解除 HALT 模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除 HALT 模式。然后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图 16-1 通过中断请求解除 HALT 模式



注 1. 有关待机解除信号的详细内容，请参照图 15-1。

2. HALT 模式解除等待时间

- 进行向量中断处理的情况
主系统时钟：15 ~ 16 个时钟
- 不进行向量中断处理的情况
主系统时钟：9 ~ 10 个时钟

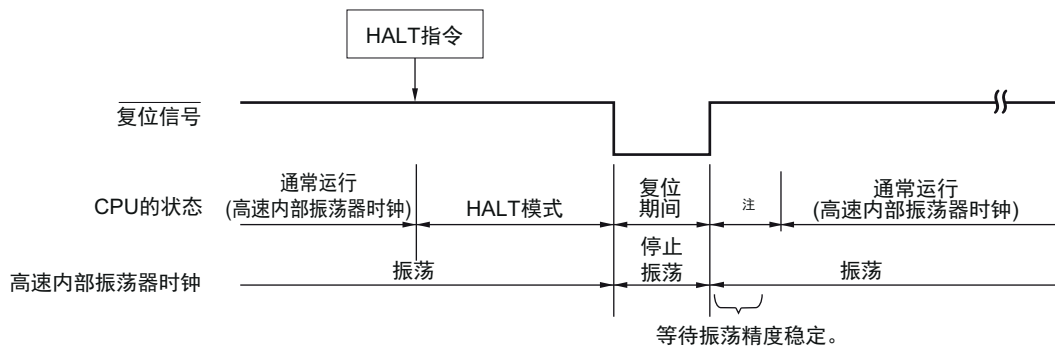
备注 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号进行的解除

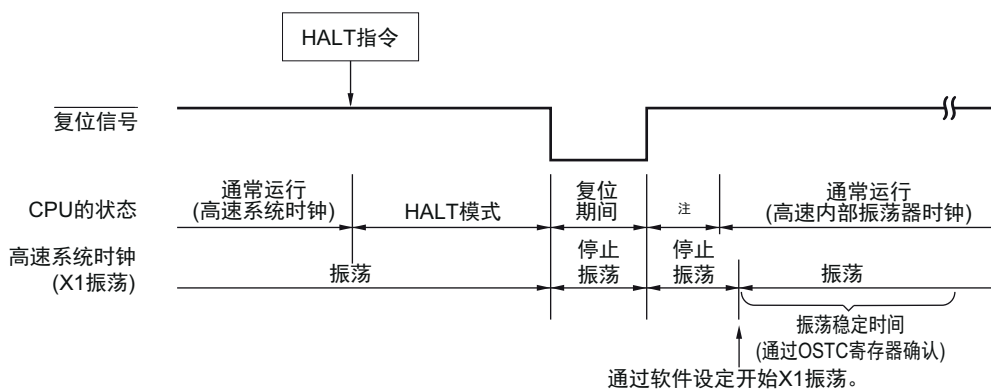
通过产生复位信号来解除 HALT 模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图 16-2 通过复位解除 HALT 模式

(1) CPU时钟为高速系统时钟的情况



(2) CPU时钟为高速内部振荡器时钟的情况



注 有关复位处理时间，请参照“第17章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第18章 上电复位电路”。

16.3.2 STOP 模式

(1) STOP 模式的设定和运行状态

通过执行 STOP 指令设定为 STOP 模式。只有在设定前的 CPU 时钟为主系统时钟的情况下才能设定。

注意 当中断屏蔽标志为“0”（允许中断处理）并且中断请求标志为“1”（产生中断请求信号）时，中断请求信号用于解除 STOP 模式。因此，如果在此情况下执行 STOP 指令，就在一旦进入 STOP 模式后立即被解除。在执行 STOP 指令并且经过 STOP 模式解除时间后返回到运行模式。

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) 以及副系统时钟提供模式控制寄存器 (OSMC) 的 WUTMMCK0 位进行设定。 <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				
数据闪存				
RAM				
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		能运行。		
看门狗定时器		参照“第 9 章 看门狗定时器”。		
时钟输出 / 蜂鸣器输出		禁止运行。		
A/D 转换器		能进行唤醒 (转移到 SNOOZE 模式)。		
串行阵列单元 (SAU)		只有 CSIp 和 UARTq 才能进行唤醒 (转移到 SNOOZE 模式)。 除了 CSIp 和 UARTq 以外, 禁止运行。		
串行阵列单元 (IICA)		能通过地址匹配进行唤醒。		
乘除器和乘加器		禁止运行。		
DMA 控制器		能运行。		
上电复位功能				
电压检测功能				
外部中断				
CRC 运算功能	高速 CRC	停止运行。		
	通用 CRC			
RAM 奇偶校验功能				
RAM 保护功能				
SFR 保护功能				
非法存储器存取检测功能				

备注 1. 停止运行: 在转移到 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

f_{EX} : 外部主系统时钟

2. p=00、q=0

(2) STOP 模式的解除

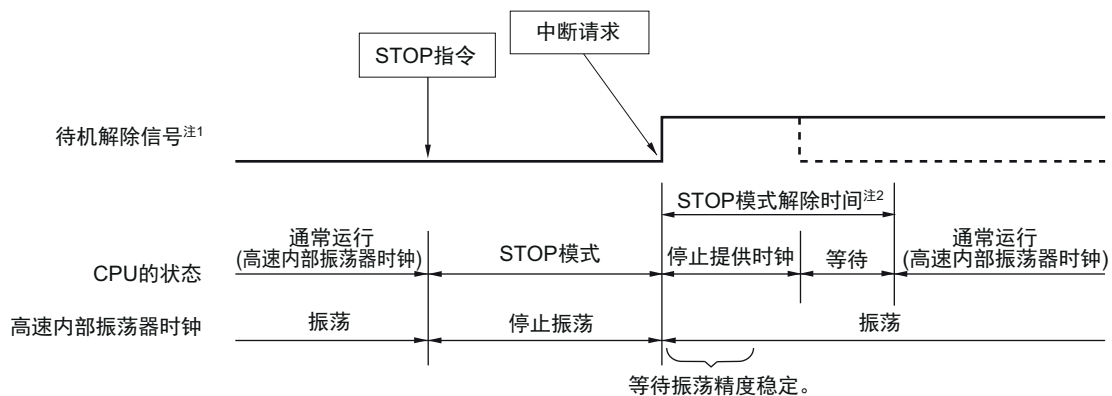
能通过以下 2 种方法解除 STOP 模式。

(a) 通过未屏蔽的中断请求进行的解除

如果发生未屏蔽的中断请求，就解除 STOP 模式。在经过振荡稳定时间后，如果处于允许接受中断的状态，就进行向量中断的处理。如果处于禁止接受中断的状态，就执行下一个地址的指令。

图 16-3 通过中断请求解除 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 15-1。

2. STOP 模式解除时间

停止提供时钟：18 μ s ~ 65 μ s

等待

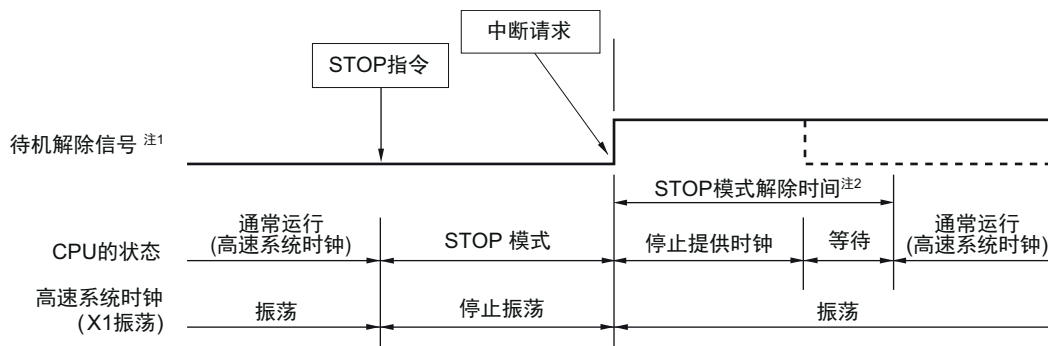
- 进行向量中断处理的情况： 7 个时钟
- 不进行向量中断处理的情况： 1 个时钟

备注 1. 时钟停止提供的时间因温度条件和 STOP 模式期间而变。

2. 虚线表示接受解除了待机模式的中断请求的情况。

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



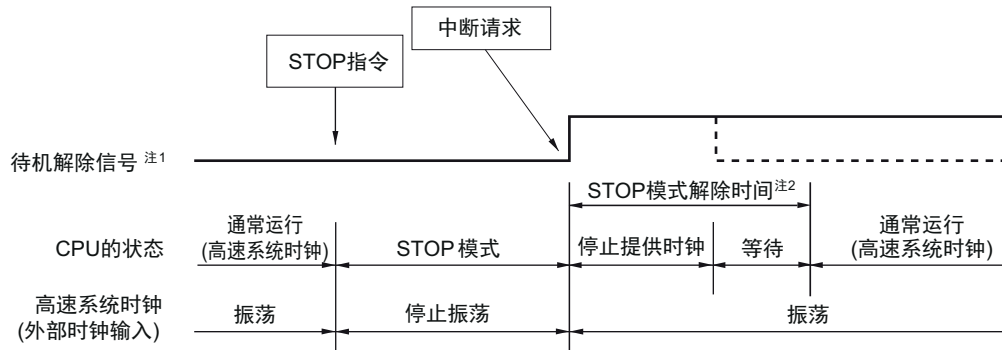
注 1. 有关待机解除信号的详细内容, 请参照图 15-1。

2. STOP 模式解除时间

停止提供时钟: $18\mu\text{s} \sim$ “ $65\mu\text{s}$ 和振荡稳定时间 (通过 OSTS 进行设定) 中较长的时间”
等待

- 进行向量中断处理的情况: 10 ~ 11 个时钟
- 不进行向量中断处理的情况: 4 ~ 5 个时钟

(3) CPU时钟为高速系统时钟 (外部时钟输入) 的情况



注 1. 有关待机解除信号的详细内容, 请参照图 15-1。

2. STOP 模式解除时间

停止提供时钟: $18\mu\text{s} \sim 65\mu\text{s}$
等待

- 进行向量中断处理的情况: 7 个时钟
- 不进行向量中断处理的情况: 1 个时钟

注意 在 CPU 以高速系统时钟 (X1 振荡) 运行并且要缩短 STOP 模式解除后的振荡稳定时间时, 必须在执行 STOP 指令前暂时将 CPU 时钟切换为高速内部振荡器时钟。

备注 1. 时钟停止提供的时间因温度条件和 STOP 模式期间而变。

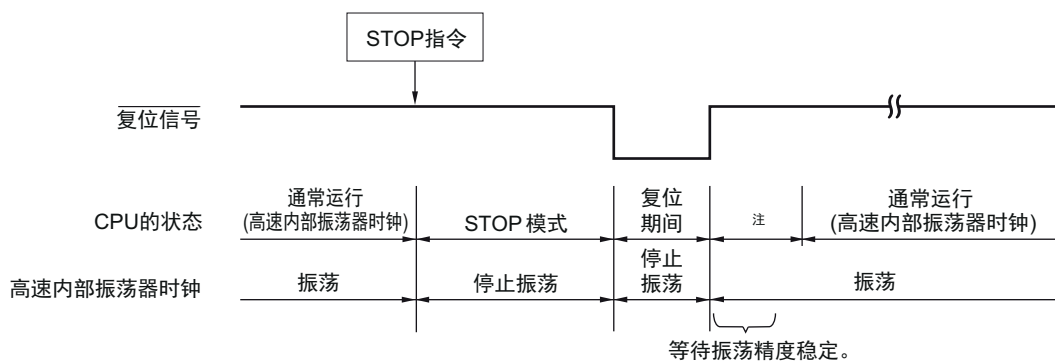
- 2. 虚线表示接受解除了待机模式的中断请求的情况。

(b) 通过产生复位信号进行的解除

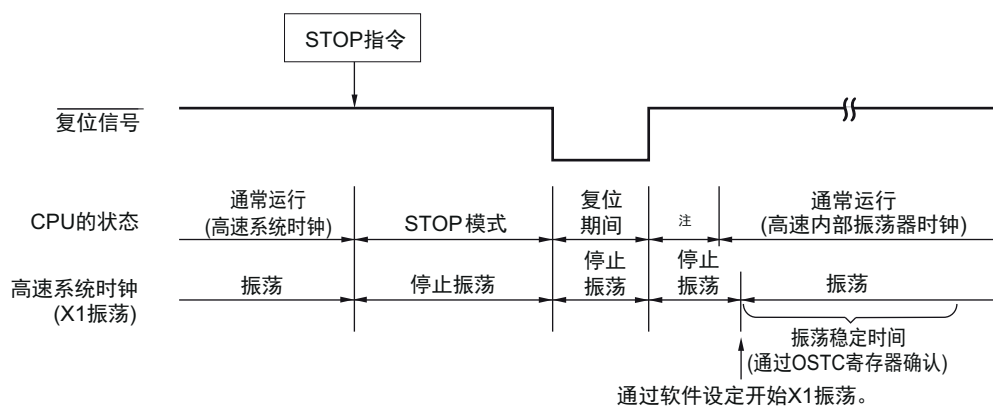
通过产生复位信号来解除 STOP 模式。然后，和通常的复位相同，在转移到复位向量地址后执行程序。

图 16-4 通过复位解除 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



注 有关复位处理时间，请参照“第17章 复位功能”。有关上电复位（POR）电路和电压检测（LVD）电路的复位处理时间，请参照“第18章 上电复位电路”。

16.3.3 SNOOZE 模式

(1) SNOOZE 模式的设定和运行状态

能给 CSIp、UARTq 或者 A/D 转换器设定 SNOOZE 模式，并且只有在设定前的 CPU 时钟为高速内部振荡器时钟的情况下才能设定。

要在 SNOOZE 模式中使用 CSIp 或者 UARTq 时，必须在即将要转移到 STOP 模式前将串行待机控制寄存器 0 (SSC0) 的 SWC0 位置“1”。详细内容请参照“11.3 控制串行阵列单元的寄存器”。

要在 SNOOZE 模式中使用 A/D 转换器时，必须在即将要转移到 STOP 模式前将 A/D 转换器模式寄存器 2 (ADM2) 的 AWC 位置“1”。详细内容请参照“10.3 控制 A/D 转换器的寄存器”。

备注 p=00、q=0

在进行 SNOOZE 模式的转移时，只在以下的时间内变为等待状态。

STOP 模式 → SNOOZE 模式的转移时间：18μs ~ 65μs

备注 STOP 模式 → SNOOZE 模式的转移时间因温度条件和 STOP 模式期间而变。

SNOOZE 模式 → 通常运行的转移时间：

- 进行向量中断处理的情况：
 - HS (高速主) 模式： “4.99 ~ 9.44μs”+7 个时钟
 - LS (低速主) 模式： “1.10 ~ 5.08μs”+7 个时钟
 - LV (低电压主) 模式： “16.58 ~ 25.40μs”+7 个时钟
- 不进行向量中断处理的情况：
 - HS (高速主) 模式： “4.99 ~ 9.44μs”+1 个时钟
 - LS (低速主) 模式： “1.10 ~ 5.08μs”+1 个时钟
 - LV (低电压主) 模式： “16.58 ~ 25.40μs”+1 个时钟

SNOOZE 模式中的运行状态如下所示。

表 16-3 SNOOZE 模式中的运行状态

STOP 模式的设定		在 STOP 模式中输入 CSIp、UARTq 的数据接收信号和 A/D 转换器的定时器触发信号的情况	
项目		CPU 以高速内部振荡器时钟 (f_{IH}) 运行	
系统时钟		停止给 CPU 提供时钟。	
主系统时钟	f_{IH}	开始运行。	
	f_X	停止	
	f_{EX}		
f_{IL}	通过选项字节 (000C0H) 的 bit0 (WDSTBYON) 和 bit4 (WDTON) 以及副系统时钟提供模式控制寄存器 (OSMC) 的 WUTMMCK0 位进行设定。 <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			
数据闪存			
RAM		停止运行 (在执行 DMA 时, 能运行)。	
端口 (锁存器)		保持 STOP 模式中的状态。	
定时器阵列单元		禁止运行。	
12 位间隔定时器		能运行。	
看门狗定时器		参照“第 9 章 看门狗定时器”。	
时钟输出 / 蜂鸣器输出		禁止运行。	
A/D 转换器		能运行。	
串行阵列单元 (SAU)		只有 CSIp 和 UARTq 才能运行。 除了 CSIp 和 UARTq 以外, 禁止运行。	
串行阵列单元 (IICA)		禁止运行。	
乘除器和乘加器			
DMA 控制器			
上电复位功能		能运行。	
电压检测功能			
外部中断			
CRC 运算功能	高速 CRC	禁止运行。	
	通用 CRC		
RAM 奇偶校验功能			
RAM 保护功能			
SFR 保护功能			
非法存储器存取检测功能			

备注 1. 停止运行: 在转移到 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

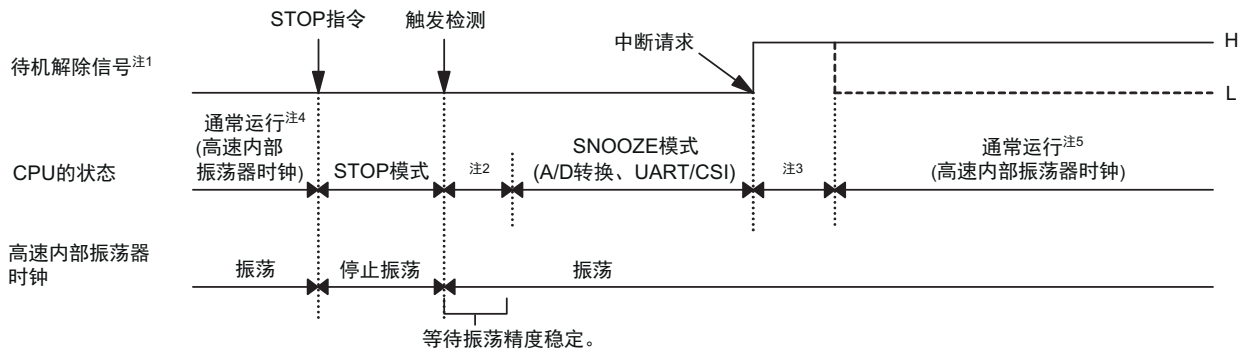
f_X : X1 时钟

f_{EX} : 外部主系统时钟

2. p=00、q=0

(2) 在 SNOOZE 模式中产生中断请求信号时的时序图

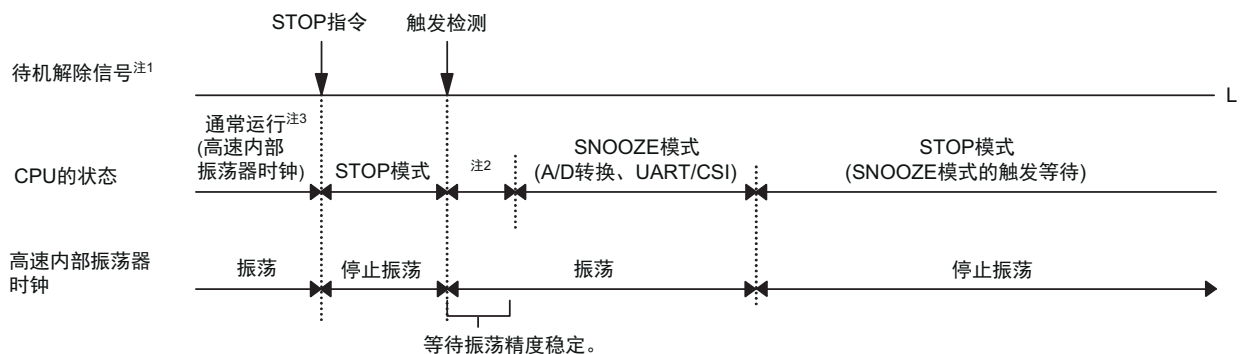
图 16-5 在 SNOOZE 模式中产生中断请求的情况



- 注 1. 有关待机解除信号的详细内容，请参照图 15-1。
 2. STOP 模式 → SNOOZE 模式的转移时间
 3. SNOOZE 模式 → 通常运行的转移时间
 4. 必须在即将要转移到 STOP 模式前设定为允许 SNOOZE 模式 (AWC=1/SWC=1)。
 5. 必须在返回到通常运行后立即设定为解除 SNOOZE 模式 (AWC=0/SWC=0)。

(3) 在 SNOOZE 模式中不产生中断请求信号时的时序图

图 16-6 在 SNOOZE 模式中不产生中断请求的情况



- 注 1. 有关待机解除信号的详细内容，请参照图 15-1。
 2. STOP 模式 → SNOOZE 模式的转移时间
 3. 必须在即将要转移到 STOP 模式前设定为允许 SNOOZE 模式 (AWC=1/SWC=1)。

备注 有关 SNOOZE 模式功能的详细内容，请参照“第 10 章 A/D 转换器”和“第 11 章 串行阵列单元”。

第 17 章 复位功能

以下 7 种方法产生复位信号。

- (1) 通过 $\overline{\text{RESET}}$ 引脚输入外部复位。
- (2) 通过看门狗定时器的程序失控检测产生内部复位。
- (3) 通过上电复位 (POR) 电路的电源电压和检测电压的比较产生内部复位。
- (4) 通过电压检测电路 (LVD) 的电源电压和检测电压的比较产生内部复位。
- (5) 因执行非法指令而产生内部复位注。
- (6) 因 RAM 奇偶校验错误而产生内部复位。
- (7) 因存取非法存储器而产生内部复位。

内部复位和外部复位相同，在产生复位信号后，从写在地址 00000H 和 00001H 中的地址开始执行程序。

当给 $\overline{\text{RESET}}$ 引脚输入低电平，或者看门狗定时器检测到程序失控，或者检测到 POR 电路和 LVD 电路的电压，或者执行非法指令注，或者发生 RAM 奇偶校验错误，或者存取非法存储器时，产生复位并且各硬件变为如表 17-1 所示的状态。

注 当执行指令码 FFH 时，发生非法指令错误。

在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

注意 1. 在进行外部复位时，必须至少给 $\overline{\text{RESET}}$ 引脚输入 10 μ s 的低电平。

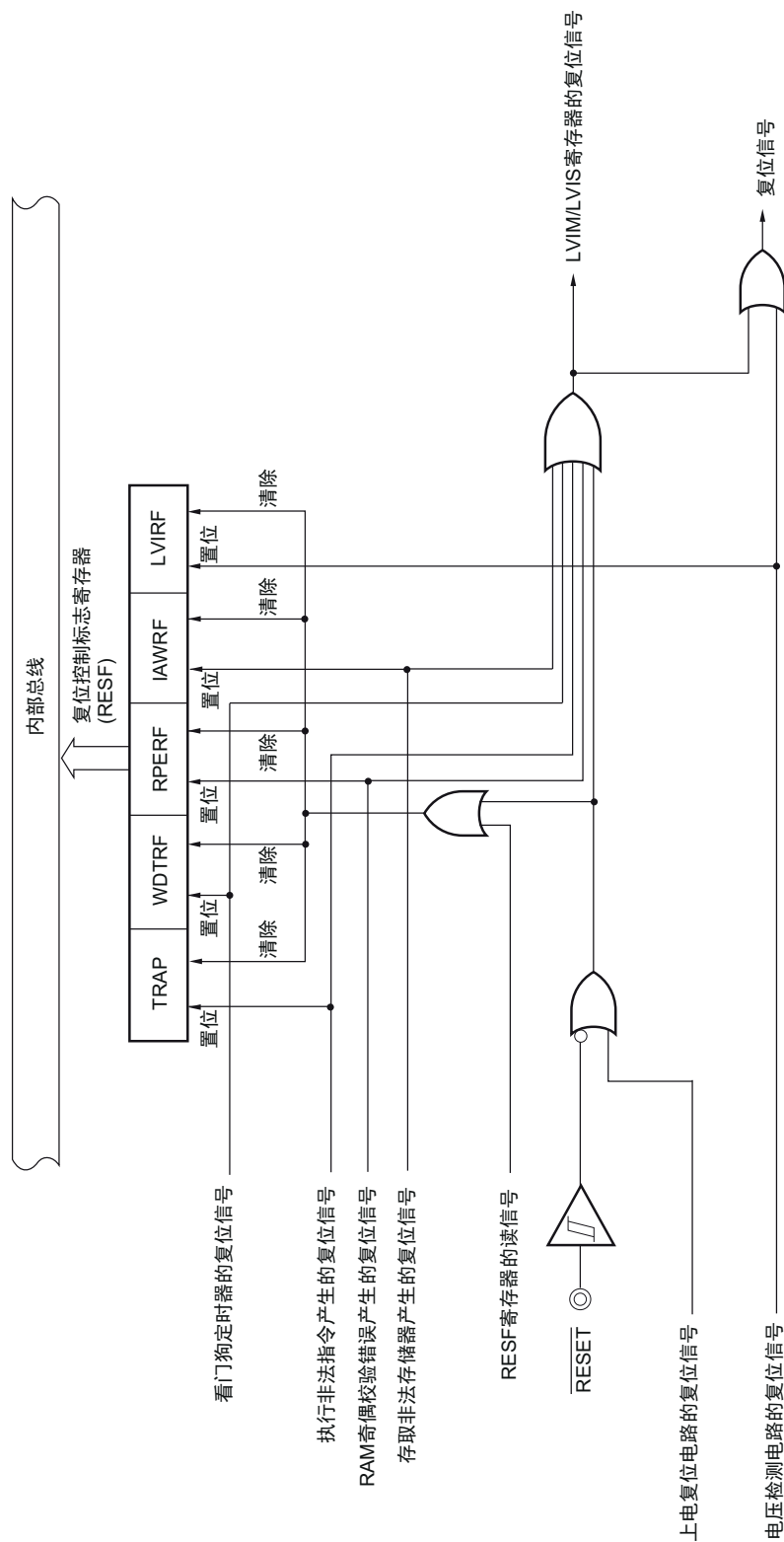
如果在电源电压上升时进行外部复位，就必须在给 $\overline{\text{RESET}}$ 引脚输入低电平后接通电源，而且在“27.4 AC 特性”所示的工作电压范围内至少保持 10 μ s 的低电平，然后输入高电平。

2. 在复位信号发生期间，停止 X1 时钟、高速内部振荡器时钟和低速内部振荡器时钟的振荡。外部主系统时钟的输入无效。
3. 如果发生复位，就对各 SFR 和 2nd SFR 进行初始化，因此端口引脚变为以下状态：
 - P40：在外部复位或者 POR 复位的期间为高阻抗。在其他复位期间以及接受复位后为高电平（连接内部上拉电阻）。
 - P40 以外的端口：在复位期间以及接受复位后为高阻抗。

备注 V_{POR} ：POR 电源电压上升检测电压

V_{LVD} ：LVD 检测电压

图 17-1 复位功能的框图



注意 LVD 电路的内部复位不会对 LVD 电路进行复位。

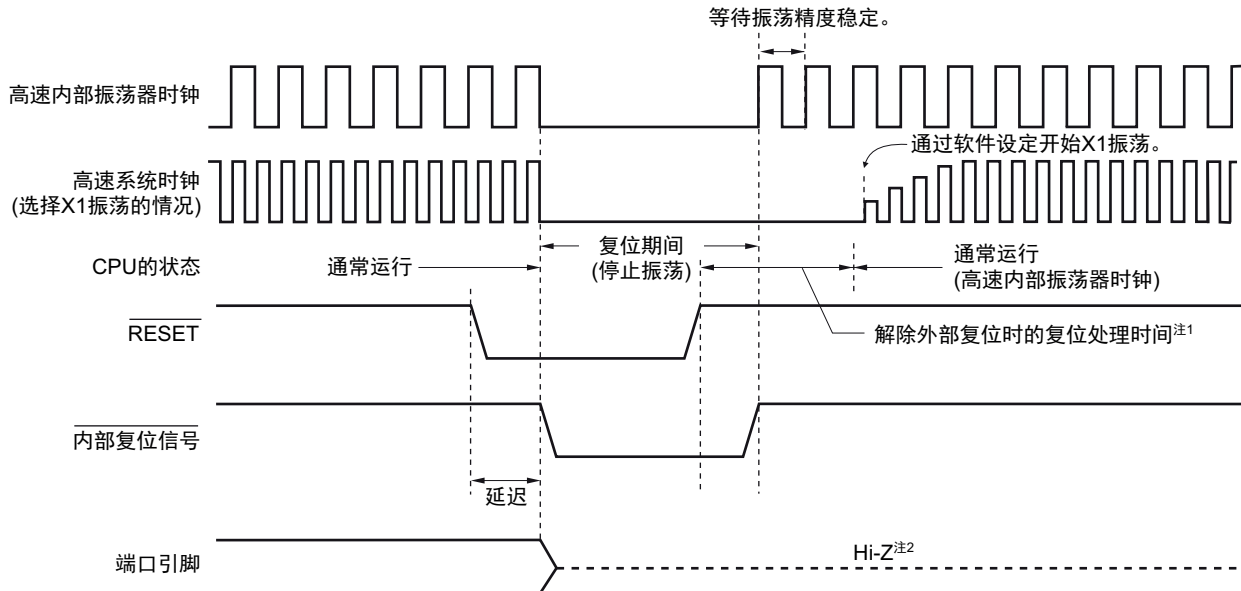
备注 1. LVIM: 电压检测寄存器

2. LVIS: 电压检测电平寄存器

17.1 复位时序

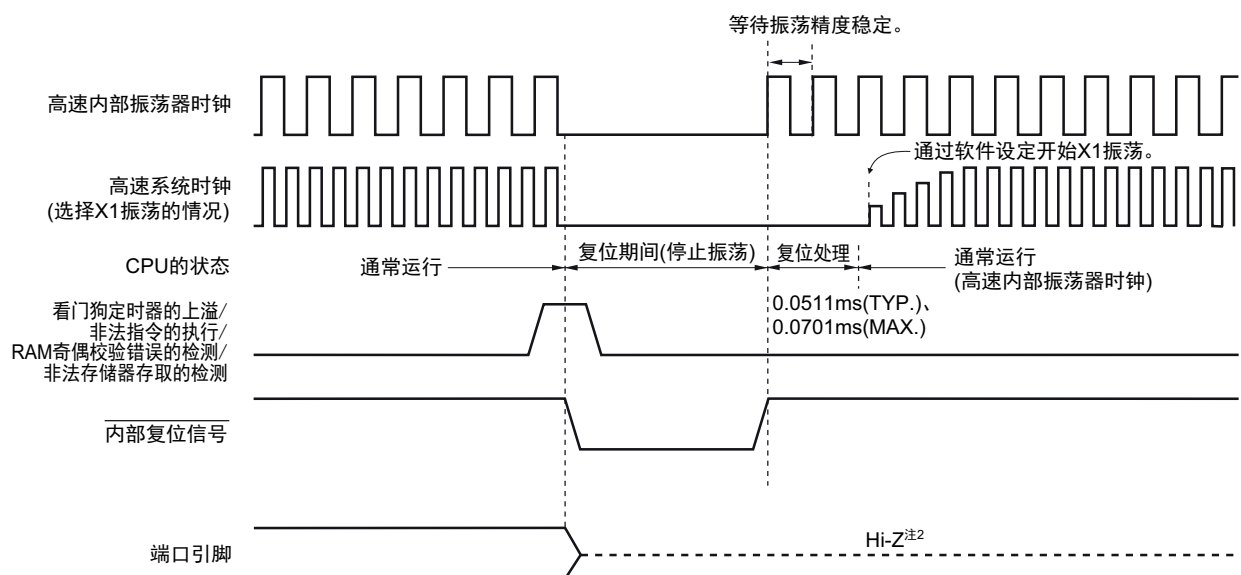
当给 $\overline{\text{RESET}}$ 引脚输入低电平时，产生复位。然后，如果给 $\overline{\text{RESET}}$ 引脚输入高电平就解除复位状态，并且在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 17-2 $\overline{\text{RESET}}$ 输入的复位时序



对于因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取检测而产生的复位，自动解除复位状态，在复位处理结束后以高速内部振荡器时钟开始执行程序。

图 17-3 因看门狗定时器的上溢、非法指令的执行、RAM 奇偶校验错误的检测或者非法存储器存取的检测而产生的复位时序



注 1. 解除外部复位时的复位处理时间:

解除 POR 后的第 1 次:	0.672ms(TYP.)、0.832ms(MAX.) (使用 LVD 的情况)
	0.399ms(TYP.)、0.519ms(MAX.) (不使用 LVD 的情况)
解除 POR 后的第 2 次以后:	0.531ms(TYP.)、0.675ms(MAX.) (使用 LVD 的情况)
	0.259ms(TYP.)、0.362ms(MAX.) (不使用 LVD 的情况)

当电源电压上升时, 在解除外部复位时的复位处理时间之前需要电压稳定等待时间0.99ms(TYP.)、2.30ms(MAX.)。

2. 端口引脚 P40 变为以下状态:

- 在外部复位或者 POR 复位的期间为高阻抗。
- 在其他复位期间以及接受复位后为高电平 (连接内部上拉电阻)。

对于由 POR 电路和 LVD 电路的电压检测产生的复位, 如果在复位后满足 $V_{DD} \geq V_{POR}$ 或者 $V_{DD} \geq V_{LVD}$, 就解除复位状态, 并且在复位处理后以高速内部振荡器时钟开始执行程序。详细内容请参照“第 18 章 上电复位电路”和“第 19 章 电压检测电路”。

17.2 复位期间的运行状态

复位期间的运行状态和接受复位后的各硬件状态分别如表 17-1 和表 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。	
数据闪存		停止运行。	
RAM		停止运行。	
端口（锁存器）		高阻抗注	
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			
串行阵列单元（SAU）			
串行接口（IICA）			
乘除器和乘加器			
DMA 控制器			
上电复位功能			能进行检测运行。
电压检测功能			在 LVD 复位时，能运行。在其他复位时，停止运行。
外部中断			停止运行。
CRC 运算功能	高速 CRC		
	通用 CRC		
RAM 奇偶校验功能			
RAM 保护功能			
SFR 保护功能			
非法存储器存取检测功能			

注 端口引脚 P40 变为以下状态：

- P40：在外部复位或者 POR 复位的期间为高阻抗。在其他复位期间为高电平（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 (00000H、00001H) 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 在复位信号发生期间和振荡稳定等待期间的各硬件状态中，只有 PC 的内容不定，而其他的硬件状态保持复位后的状态。

备注 有关特殊功能寄存器 (SFR: Special Function Register) 的接受复位后的状态，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 确认复位源的寄存器

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。

通过 RESET 的输入、上电复位（POR）电路的复位和 RESF 寄存器的读取，清除 TRAP、WDTRF、RPERF、IAWRF、LVIRF 标志。

图 17-4 复位控制标志寄存器（RESF）的格式

地址：FFFA8H 复位后：不定值^{注 1} R

符号	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	执行非法指令产生的内部复位请求 ^{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	看门狗定时器（WDT）产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

RPERF	RAM 奇偶校验错误产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

IAWRF	存取非法存储器产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

LVIRF	电压检测电路（LVD）产生的内部复位请求
0	没有产生内部复位请求或者清除了 RESF 寄存器。
1	产生内部复位请求。

- 注 1. 因复位源而不同。请参照表 17-3。
 2. 当执行指令码 FFH 时，发生非法指令错误。
 在通过在线仿真器或者片上调试仿真器进行仿真时，不会因执行非法指令而产生复位。

- 注意 1. 不能通过位存储器操作指令读数据。
 2. 在允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的情况下，当存取数据时，必须对“所用 RAM 区”进行初始化；当从 RAM 区执行指令时，必须对“所用 RAM 区 +10 字节”的区域进行初始化。
 通过产生复位，进入允许产生 RAM 奇偶校验错误复位（RPERDIS=0）的状态。详细内容请参照“20.3.3 RAM 奇偶校验错误检测功能”。

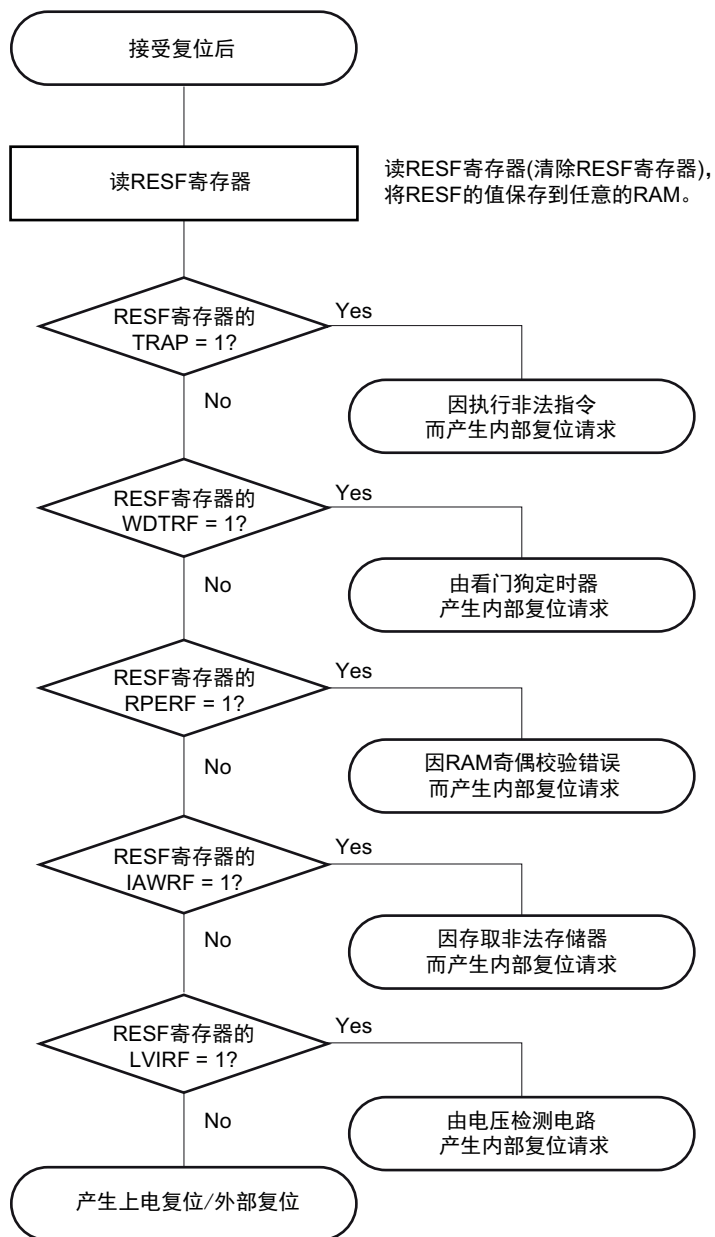
发生复位请求时的 RESF 寄存器状态如表 17-3 所示。

表 17-3 发生复位请求时的 RESF 寄存器状态

标志 \ 复位源	$\overline{\text{RESET}}$ 输入	POR 产生的复位	执行非法指令产生的复位	WDT 产生的复位	RAM 奇偶校验错误产生的复位	存取非法存储器产生的复位	LVD 产生的复位
TRAP	清“0”	清“0”	置“1”	保持	保持	保持	保持
WDTRF			保持	置“1”			
RPERF			保持	置“1”			
IAWRF			保持	置“1”			
LVIRF			保持	置“1”			

如果通过 8 位存储器操作指令读 RESF 寄存器，就自动清除此寄存器。
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



上述流程是确认步骤的例子。

第 18 章 上电复位电路

18.1 上电复位电路的功能

上电复位电路（POR）有以下功能。

- 在接通电源时产生内部复位信号。
如果电源电压（ V_{DD} ）超过检测电压（ V_{POR} ），就解除复位。但是，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态。
- 将电源电压（ V_{DD} ）和检测电压（ V_{PDR} ）进行比较。当 $V_{DD} < V_{PDR}$ 时，产生内部复位信号。但是，当电源电压下降时，必须在电源电压低于“27.4 AC 特性”所示的工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

注意 当上电复位电路产生内部复位信号时，将复位控制标志寄存器（RESF）清为“00H”。

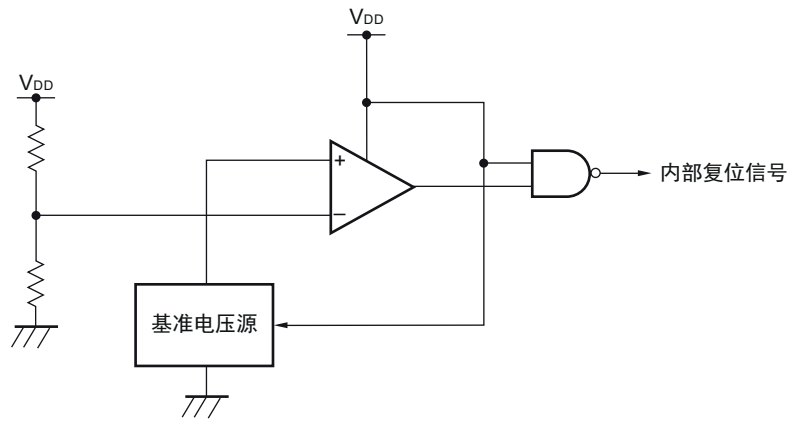
备注 1. RL78 微控制器内置多个产生内部复位信号的硬件。当由看门狗定时器（WDT）、电压检测（LVD）电路、非法指令的执行、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，用于表示复位源的标志分配在 RESF 寄存器；当由 WDT、LVD、非法指令的执行、RAM 奇偶校验错误或者非法存储器的存取而产生内部复位信号时，不将 RESF 寄存器清为“00H”而将标志置“1”。有关 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

2. V_{POR} : POR 电源电压上升检测电压
 V_{PDR} : POR 电源电压下降检测电压
详细内容请参照“27.6.3 POR 电路特性”。

18.2 上电复位电路的结构

上电复位电路的框图如图 18-1 所示。

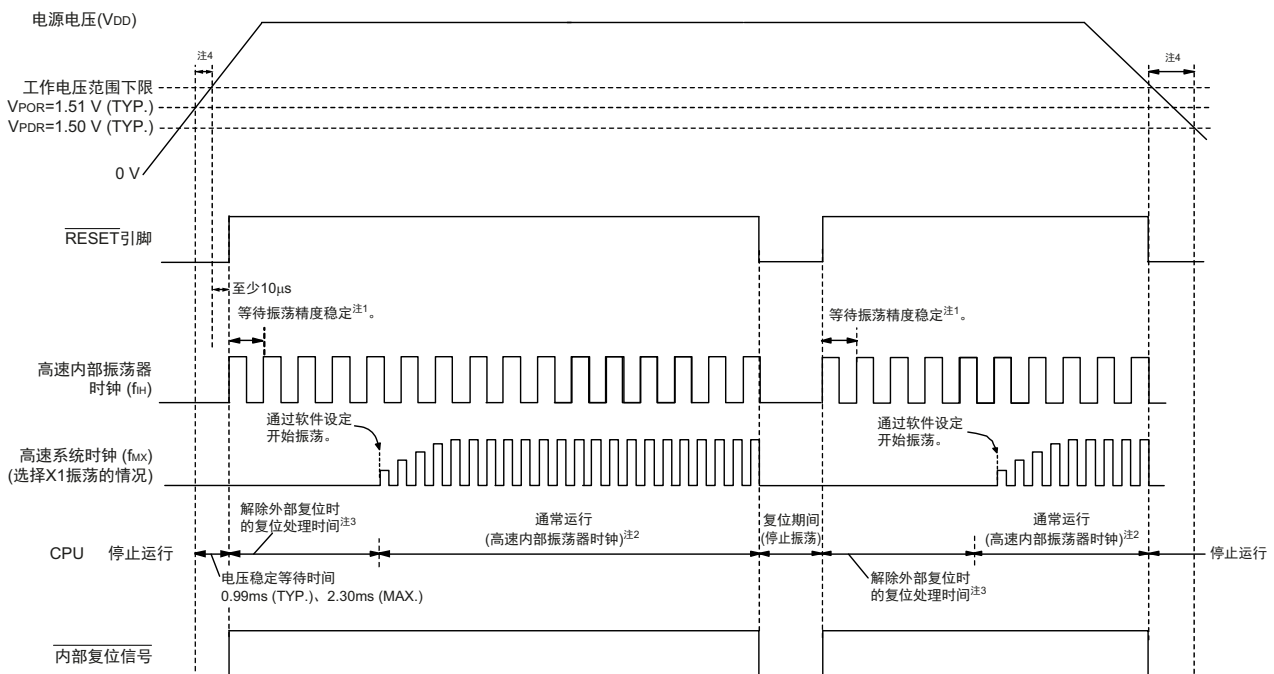
图 18-1 上电复位电路的框图



18.3 上电复位电路的运行

上电复位电路和电压检测电路的内部复位信号的产生时序如下所示。

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (1/3)

(1) 使用 $\overline{\text{RESET}}$ 引脚的外部复位输入的情况

- 注 1. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 能选择高速内部振荡器时钟或者高速系统时钟作为 CPU 时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换。
3. 到开始通常运行的时间除了达到 V_{POR}(1.51V(TYP.)) 后的“电压稳定等待时间”以外，还需要 $\overline{\text{RESET}}$ 信号变为高电平 (1) 后的如下的“外部复位解除时的复位处理时间”（POR 解除后的第 1 次）。外部复位解除时的复位处理时间如下所示。
- POR 解除后的第 1 次： 0.672ms(TYP.)、0.832ms(MAX.)（使用 LVD 时）
0.399ms(TYP.)、0.519ms(MAX.)（LVD OFF 时）
- POR 解除后的第 2 次以后（包括第 2 次）的外部复位解除时的复位处理时间如下所示。
- POR 解除后的第 2 次以后（包括第 2 次）：0.531ms(TYP.)、0.675ms(MAX.)（使用 LVD 时）
0.259ms(TYP.)、0.362ms(MAX.)（LVD OFF 时）
4. 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

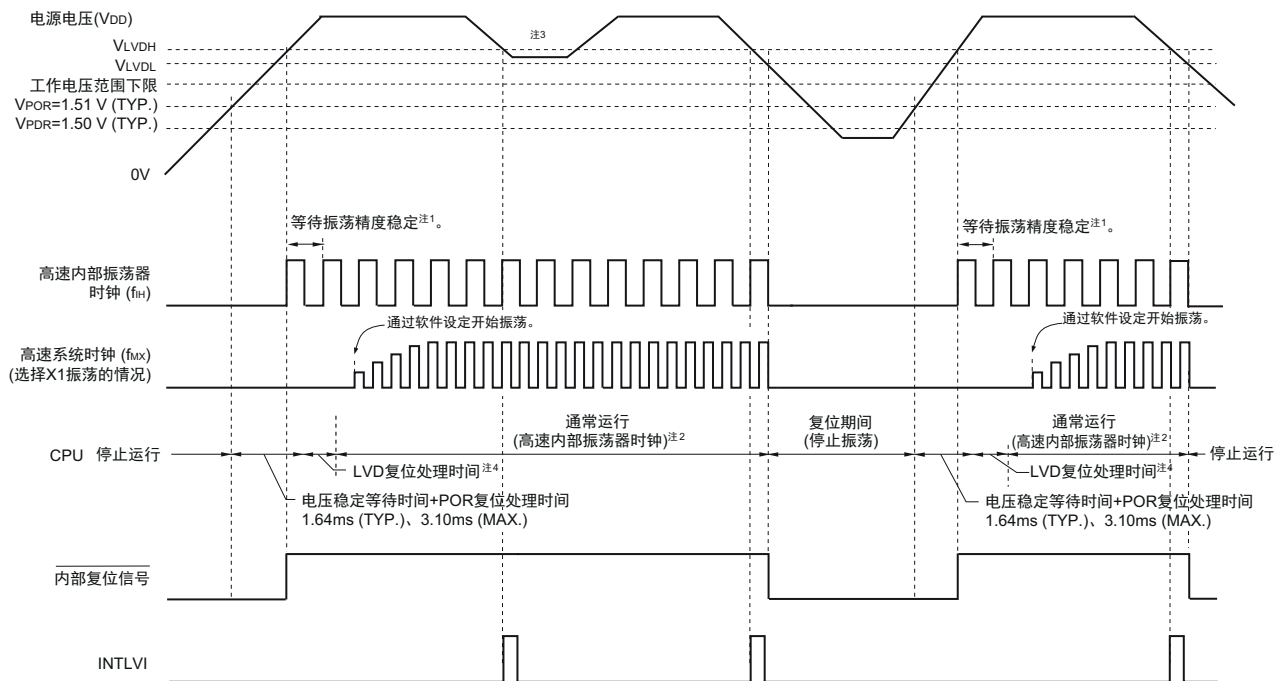
备注 V_{POR}：POR 电源电压上升检测电压

V_{PDR}：POR 电源电压下降检测电压

注意 在 LVD 为 OFF 时，必须使用 $\overline{\text{RESET}}$ 引脚的外部复位。详细内容请参照“第 19 章 电压检测电路”。

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (2/3)

(2)LVD 为中断 & 复位模式的情况 (选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、0)



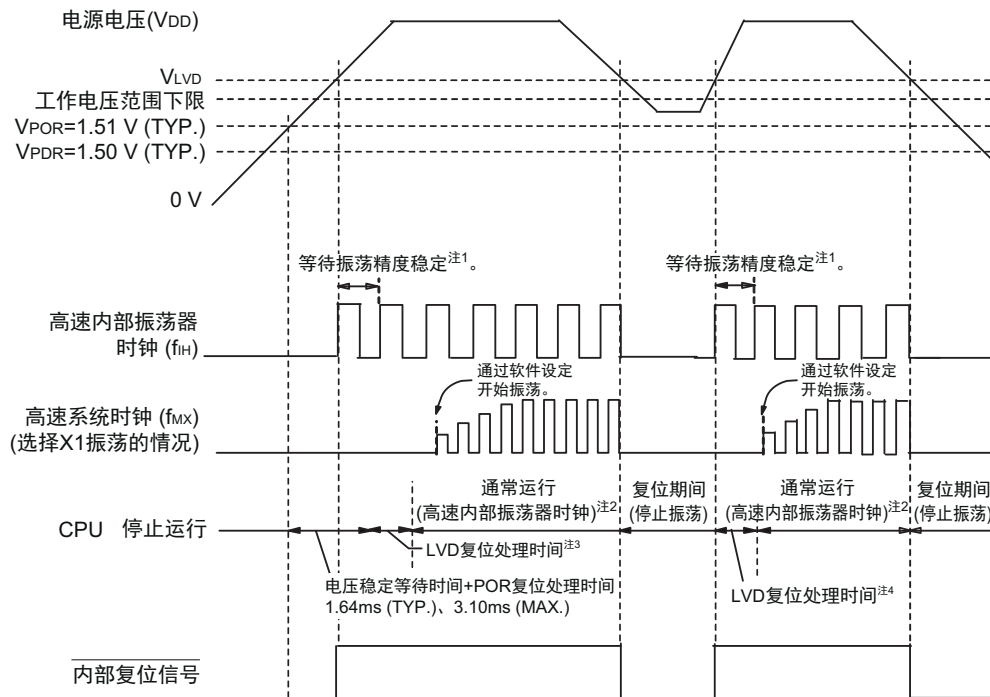
- 注 1. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 能选择高速内部振荡器时钟或者高速系统时钟作为 CPU 时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器（OSTC）确认振荡稳定时间后进行切换。
3. 在产生中断请求信号（INTLVI）后，自动将电压检测电平寄存器（LVIS）的 LVILV 位和 LVIMD 位置“1”。因此，必须考虑可能出现电源电压在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高的情况，在产生 INTLVI 后按照“图 19-8 工作电压的确认 / 复位的设定步骤”和“图 19-9 中断 & 复位模式的初始设定步骤”进行设定。
4. 到开始通常运行为止的时间除了达到 V_{POR} （1.51V(TYP.)）后的“电压稳定等待时间 + POR 复位处理时间”以外，在达到 LVD 检测电平（ V_{LVDH} ）后还需要以下的“LVD 复位处理时间”。

LVD 复位处理时间：0ms ~ 0.0701ms(MAX.)

备注 V_{LVDH} 、 V_{LVDL} ：LVD 检测电压
 V_{POR} ：POR 电源电压上升检测电压
 V_{PDR} ：POR 电源电压下降检测电压

图 18-2 上电复位电路和电压检测电路的内部复位信号的产生时序 (3/3)

(3) LVD 复位模式的情况 (选项字节 000C1H 的 LVIMDS1、LVIMDS0=1、1)



- 注 1. 内部复位处理时间包含高速内部振荡器时钟的振荡精度稳定等待时间。
2. 能选择高速内部振荡器时钟或者高速系统时钟作为 CPU 时钟。在使用 X1 时钟的情况下，必须在通过振荡稳定时间计数器的状态寄存器 (OSTC) 确认振荡稳定时间后进行切换。
3. 到开始通常运行为止的时间除了达到 V_{POR} (1.51V(TYP.)) 后的“电压稳定等待时间 + POR 复位处理时间”以外，在达到 LVD 检测电平 (V_{LVD}) 后还需要以下的“LVD 复位处理时间”。
- LVD 复位处理时间: 0ms ~ 0.0701ms(MAX.)
4. 在电源电压下降时，如果只在发生电压检测电路 (LVD) 的内部复位后恢复电源电压，就在达到 LVD 检测电平 (V_{LVD}) 后需要以下的“LVD 复位处理时间”。
- LVD 复位处理时间: 0.0511ms(TYP.)、0.0701ms(MAX.)

备注 1. V_{LVDH}、V_{LVDL}: LVD 检测电压V_{POR}: POR 电源上升检测电压V_{PDR}: POR 电源下降检测电压

2. 当选择 LVD 中断模式 (选项字节 000C1H 的 LVIMD1、LVIMD0=0、1) 时，从接通电源到开始通常运行的时间和“图 18-2 (3) LVD 复位模式的情况”的“注 3”的时间相同。

第 19 章 电压检测电路

19.1 电压检测电路的功能

电压检测电路通过选项字节（000C1H）设定运行模式和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）。电压检测（LVD）电路有以下功能。

- 将电源电压（ V_{DD} ）和检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）进行比较，产生内部复位或者内部中断信号。
- 电源电压的检测电压（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）能通过选项字节选择 14 种检测电平（参照“第 22 章 选项字节”）。
- 也能在 STOP 模式中运行。
- 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

(a) 中断 & 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）

通过选项字节 000C1H 选择 2 个检测电压（ V_{LVDH} 、 V_{LVDL} ），高电压检测电平（ V_{LVDH} ）用于解除复位或者产生中断，低电压检测电平（ V_{LVDL} ）用于产生复位。

(b) 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、1）

将选项字节 000C1H 选择的 1 个检测电压（ V_{LVD} ）用于产生或者解除复位。

(c) 中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）

将选项字节 000C1H 选择的 1 个检测电压（ V_{LVD} ）用于解除复位或者产生中断。

在各模式中，产生以下的中断信号和内部复位信号。

中断 & 复位模式 (LVIMDS1、LVIMDS0=1、0)	复位模式 (LVIMDS1、LVIMDS0=1、1)	中断模式 (LVIMDS1、LVIMDS0=0、1)
在电源电压下降时，当检测到 $V_{DD} < V_{LVDH}$ 时，产生中断请求信号；当检测到 $V_{DD} < V_{LVDL}$ 时，产生内部复位；当检测到 $V_{DD} \geq V_{LVDH}$ 时，解除内部复位。	当检测到 $V_{DD} \geq V_{LVD}$ 时，解除内部复位；当检测到 $V_{DD} < V_{LVD}$ 时，产生内部复位。	在发生复位后，LVD 的内部复位状态继续保持到 $V_{DD} \geq V_{LVD}$ 为止。当检测到 $V_{DD} \geq V_{LVD}$ 时，解除 LVD 的内部复位。在解除 LVD 的内部复位后，如果检测到 $V_{DD} < V_{LVD}$ 或者 $V_{DD} \geq V_{LVD}$ ，就产生中断请求信号（INTLVI）。

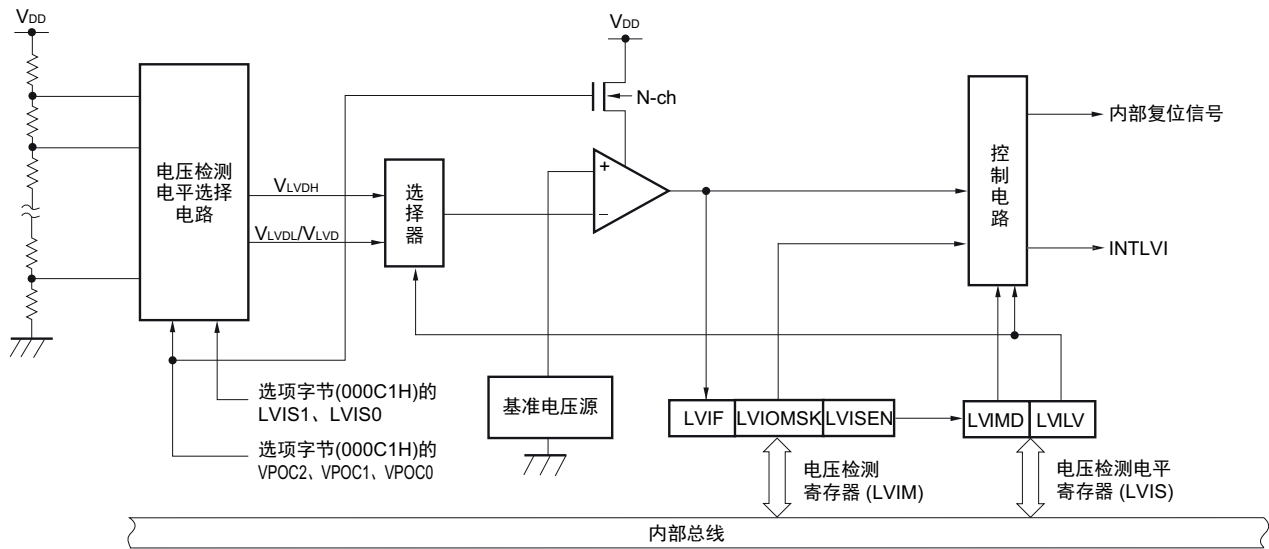
在电压检测电路运行时，能通过读电压检测标志（LVIF：电压检测寄存器（LVIM）的 bit0）来确认电源电压是大于等于检测电压还是小于检测电压。

如果发生复位，就将复位控制标志寄存器（RESF）的 bit0（LVIRF）置“1”。有关 RESF 寄存器的详细内容，请参照“第 17 章 复位功能”。

19.2 电压检测电路的结构

电压检测电路的框图如图 19-1 所示。

图 19-1 电压检测电路的框图



19.3 控制电压检测电路的寄存器

通过以下寄存器控制电压检测电路。

- 电压检测寄存器 (LVIM)
- 电压检测电平寄存器 (LVIS)

19.3.1 电压检测寄存器 (LVIM)

此寄存器设定允许或者禁止改写电压检测电平寄存器 (LVIS)，并且确认 LVD 输出的屏蔽状态。

通过 1 位或者 8 位存储器操作指令设定 LVIM 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 19-2 电压检测寄存器 (LVIM) 的格式

地址: FFFA9H	复位后: 00H 注 1	R/W 注 2						
符号	7	6	5	4	3	2	1	0
LVIM	LVISEN 注 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN 注 3	电压检测电平寄存器 (LVIS) 的允许 / 禁止改写的设定
0	禁止改写 LVIS 寄存器 (LVIOMSK=0 (LVD 输出屏蔽无效))。
1	允许改写 LVIS 寄存器 (LVIOMSK=1 (LVD 输出屏蔽有效))。

LVIOMSK	LVD 输出的屏蔽状态标志
0	LVD 输出屏蔽无效。
1	LVD 输出屏蔽有效注 4。

LVIF	电压检测标志
0	电源电压 (V_{DD}) \geq 检测电压 (V_{LVD}) 或者 LVD 为 OFF。
1	电源电压 (V_{DD}) $<$ 检测电压 (V_{LVD})

- 注 1. 复位值因复位源而变。
在 LVD 发生复位时，不对 LVIM 寄存器的值进行复位而保持原来的值；在其他复位时，将 LVISEN 清“0”。
2. bit0 和 bit1 是只读位。
3. 只有在选择中断 & 复位模式 (选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”)时才能设定，在其他模式中不能更改初始值。
4. 只有在选择中断 & 复位模式 (选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”)时，LVIOMSK 位才在以下期间自动变为“1”，屏蔽 LVD 产生的复位或者中断。
- LVISEN=1 的期间
 - 从发生 LVD 中断开始到 LVD 检测电压稳定为止的等待时间
 - 从更改 LVILV 位的值到 LVD 检测电压稳定为止的等待时间

19.3.2 电压检测电平寄存器 (LVIS)

这是设定电压检测电平的寄存器。

通过 1 位或者 8 位存储器操作指令设定 LVIS 寄存器。

在产生复位信号后，此寄存器的值变为“00H/01H/81H”注 1。

图 19-3 电压检测电平寄存器 (LVIS) 的格式

地址: FFFAAH	复位后: 00H/01H/81H 注 1							R/W
符号	7	6	5	4	3	2	1	0
LVIS	LVIMD	0	0	0	0	0	0	LVILV
LVIMD 注 2	电压检测的运行模式							
0	中断模式							
1	复位模式							
LVILV 注 2	LVD 检测电平							
0	高电压检测电平 (V_{LVDH})							
1	低电压检测电平 (V_{LVDL} 或者 V_{LVD})							

注 1. 复位值因复位源和选项字节的设定而变。

在发生 LVD 复位时，不将此寄存器清为“00H”。

在发生 LVD 以外的复位时，此寄存器的值如下：

- 选项字节的 LVIMDS1、LVIMDS0=1、0 时：00H
- 选项字节的 LVIMDS1、LVIMDS0=1、1 时：81H
- 选项字节的 LVIMDS1、LVIMDS0=0、1 时：01H

2. 只有在选择中断 & 复位模式（选项字节的 LVIMDS1 位和 LVIMDS0 位分别为“1”和“0”）时才能写“0”。在其他情况下不能设定。在中断 & 复位模式中，通过产生复位或者中断自动进行值的替换。

注意 1. 要改写 LVIS 寄存器时，必须按照图 19-8 和图 19-9 的步骤进行。

2. 通过选项字节 000C1H 选择 LVD 的运行模式和各模式的检测电压 (V_{LVDH} 、 V_{LVDL} 、 V_{LVD})。用户选项字节 (000C1H/010C1H) 的格式如图 19-4 所示。有关选项字节的详细内容，请参照“第 22 章 选项字节”。

图 19-4 用户选项字节（000C1H/010C1H）的格式 (1/2)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD的设定（中断&复位模式）

检测电压			选项字节的设定值						
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V	0	1	1	1	0	1	0
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V	1	0	1	1	0	1	0
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V	1	1	1	1	0	1	0
3.02V	2.96V					0	1		
4.06V	3.98V					0	0		
—			禁止设定上述以外的值。						

- LVD的设定（复位模式）

检测电压		选项字节的设定值						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	1	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—			禁止设定上述以外的值。					

注 在引导交换时，000C1H被010C1H替换，因此必须给010C1H设定和000C1H相同的值。

备注 1. 有关 LVD 电路的详细内容，请参照“第 19 章 电压检测电路”。

2. 检测电压是 TYP. 值。详细内容请参照“27.6.4 LVD 电路特性”。

图 19-4 用户选项字节（000C1H/010C1H）的格式 (2/2)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断模式）

检测电压		选项字节的设定值								
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.67V	1.63V	0	0	0	1	1	0	1		
1.77V	1.73V		0	0	1	0				
1.88V	1.84V		0	1	1	1				
1.98V	1.94V		0	1	1	0				
2.09V	2.04V		0	1	0	1				
2.50V	2.45V		1	0	1	1				
2.61V	2.55V		1	0	1	0				
2.71V	2.65V		1	0	0	1				
2.81V	2.75V		1	1	1	1				
2.92V	2.86V		1	1	1	0				
3.02V	2.96V		1	1	0	1				
3.13V	3.06V		0	1	0	0				
3.75V	3.67V		1	0	0	0				
4.06V	3.98V		1	1	0	0				
—	—		禁止设定上述以外的值。							

- LVD 为 OFF（使用 $\overline{\text{RESET}}$ 引脚的外部复位）

检测电压		选项字节的设定值						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—	—	禁止设定上述以外的值。						

注 在引导交换时，000C1H 被 010C1H 替换，因此必须给 010C1H 设定和 000C1H 相同的值。

注意 1. 必须给 bit4 写“1”。

- 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位位置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

备注 1. ×：忽略

- 有关 LVD 电路的详细内容，请参照“第 19 章 电压检测电路”。
- 检测电压是 TYP. 值。详细内容请参照“27.6.4 LVD 电路特性”。

19.4 电压检测电路的运行

19.4.1 用作复位模式时的设定

通过选项字节000C1H设定运行模式（复位模式（LVIMDS1、LVIMDS0=1、1）和检测电压（ V_{LVD} ）。如果设定复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“81H”。
bit7（LVIMD）为“1”（复位模式）。
bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。

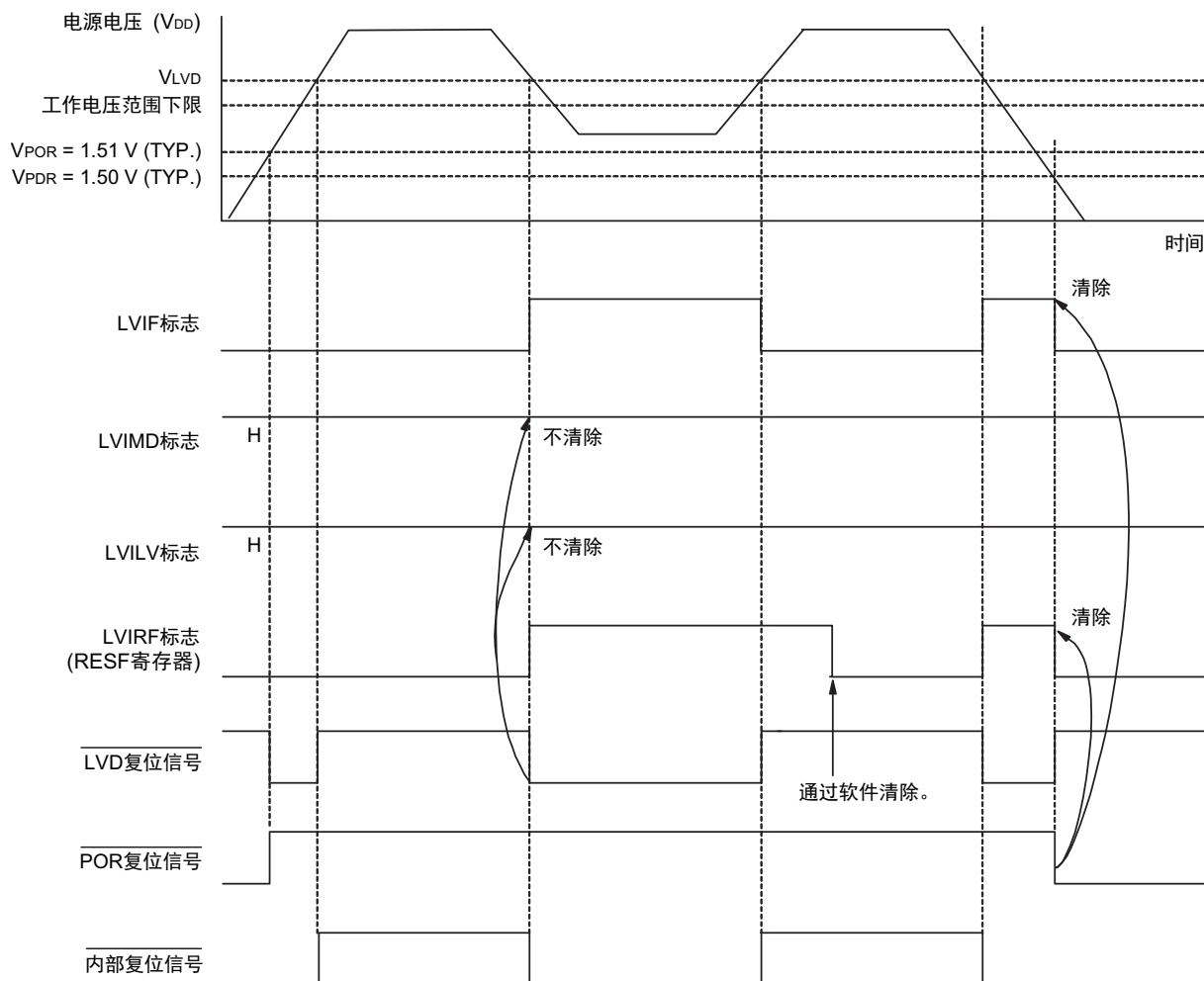
● LVD复位模式的运行

当接通电源时，复位模式（选项字节的LVIMDS1、LVIMDS0=1、1）在电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ）前保持LVD的内部复位状态。如果电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ），就解除内部复位。

当电源电压下降时，如果电源电压（ V_{DD} ）低于电压检测电平（ V_{LVD} ），就产生LVD的内部复位。

LVD复位模式的内部复位信号的产生时序如图 19-5 所示。

图 19-5 内部复位信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=1、1）



备注 V_{POR} : POR 电源电压上升检测电压

V_{PDR} : POR 电源电压下降检测电压

19.4.2 用作中断模式时的设定

通过选项字节 000C1H 设定运行模式（中断模式（LVIMDS1、LVIMDS0=0、1）和检测电压（ V_{LVD} ）。如果设定中断模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“01H”。
bit7（LVIMD）为“0”（中断模式）。
bit0（LVILV）为“1”（电压检测电平： V_{LVD} ）。

● LVD 中断模式的运行

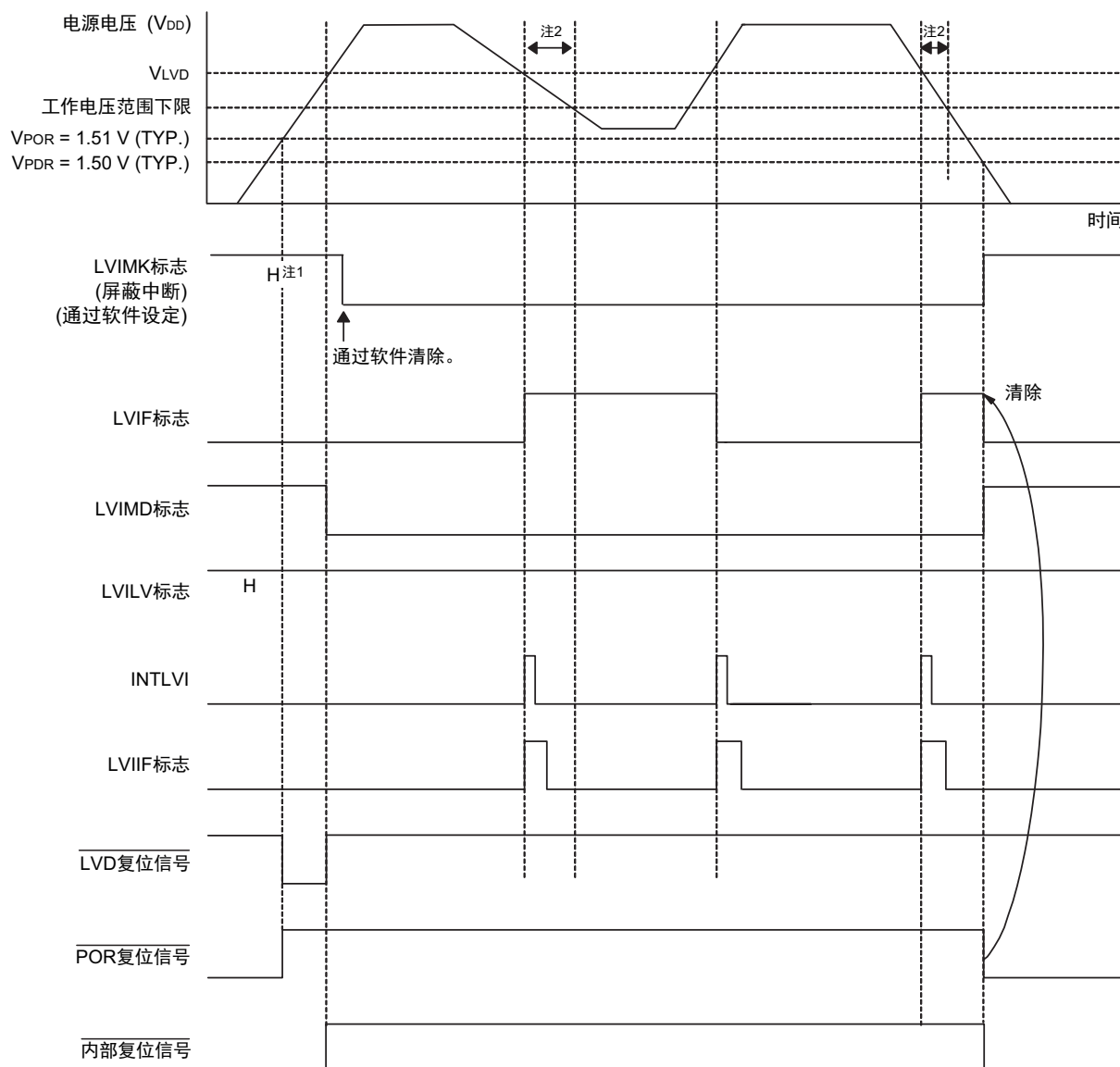
在产生复位后，中断模式（选项字节的 LVIMDS1、LVIMDS0=0、1）在电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ）前保持 LVD 的内部复位状态。如果电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ），就解除内部复位。

在解除 LVD 的内部复位后，如果电源电压（ V_{DD} ）超过电压检测电平（ V_{LVD} ），就产生 LVD 的中断请求信号（INTLVI）。

当电源电压下降时，必须在电源电压低于“27.4 AC 特性”所示的工作电压范围前，转移到 STOP 模式，或者通过外部复位置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

LVD 中断模式的中断请求信号的产生时序如图 19-6 所示。

图 19-6 中断信号的产生时序（选项字节的 LVIMDS1、LVIMDS0=0、1）



- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 当电源电压下降时，必须在电源电压低于“27.4 AC 特性”所示的工作电压范围前，转移到 STOP 模式，或者通过外部复位位置为复位状态。在重新开始运行时，必须确认电源电压是否恢复到工作电压范围。

备注 V_{POR}: POR 电源电压上升检测电压
V_{PDR}: POR 电源电压下降检测电压

19.4.3 用作中断 & 复位模式时的设定

通过选项字节 000C1H 设定运行模式（中断 & 复位模式（LVIMDS1、LVIMDS0=1、0）和检测电压（ V_{LVDH} 、 V_{LVDL} ）。

如果设定中断 & 复位模式，就在以下初始设定的状态下开始运行。

- 将电压检测寄存器（LVIM）的 bit7（LVISEN）置“0”（禁止改写电压检测电平寄存器（LVIS））。
- 将电压检测电平寄存器（LVIS）的初始值置“00H”。
 - bit7（LVIMD）为“0”（中断模式）。
 - bit0（LVILV）为“0”（高电压检测电平： V_{LVDH} ）。

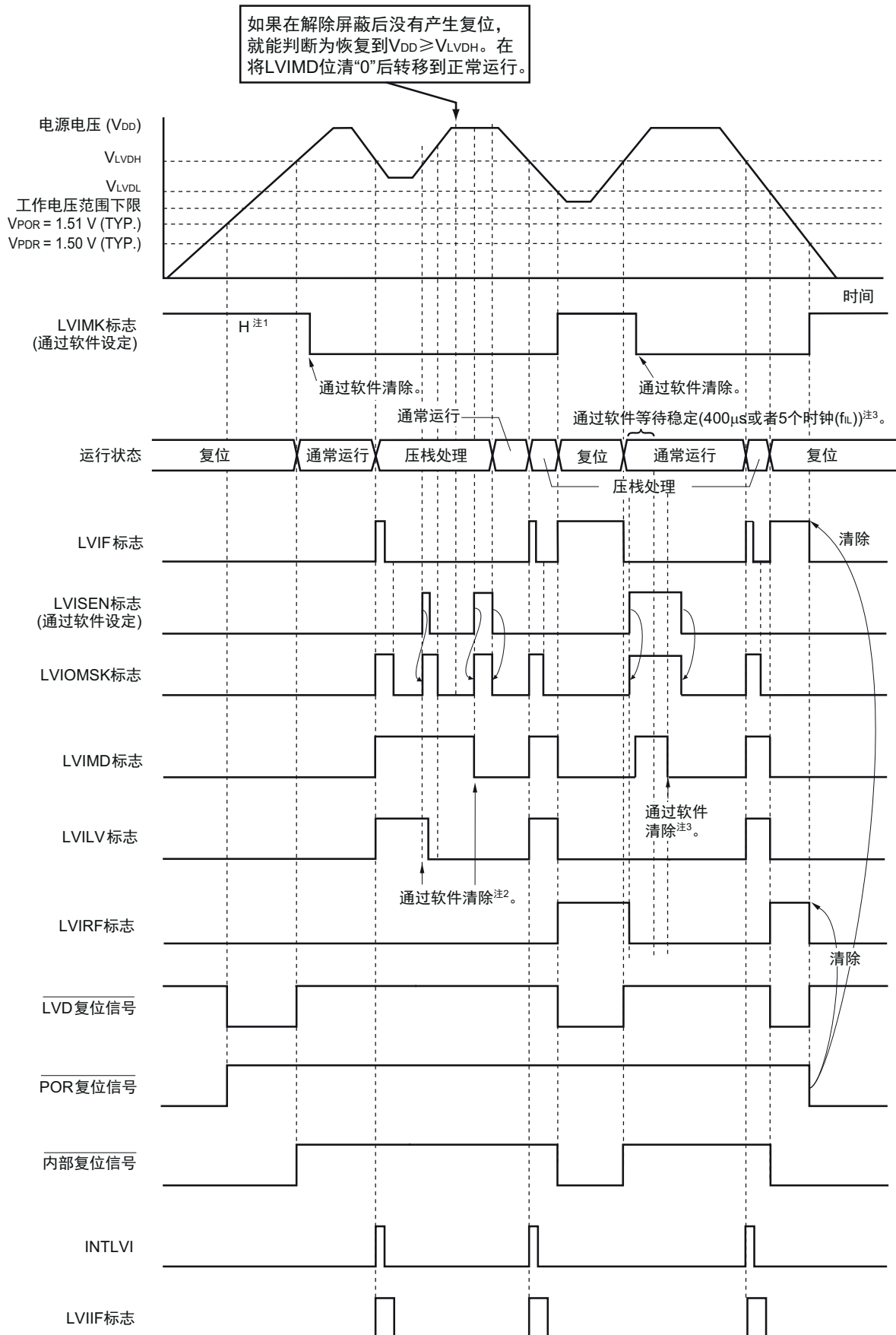
● LVD 中断 & 复位模式的运行

当接通电源时，中断 & 复位模式（选项字节的 LVIMDS1、LVIMDS0=1、0）在电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ）前保持 LVD 的内部复位状态。如果电源电压（ V_{DD} ）超过高电压检测电平（ V_{LVDH} ），就解除内部复位。

当电源电压下降时，如果电源电压（ V_{DD} ）低于高电压检测电平（ V_{LVDH} ），就产生 LVD 的中断请求信号（INTLVI）并且能进行任意的压栈处理。此后，如果电源电压（ V_{DD} ）低于低电压检测电平（ V_{LVDL} ），就产生 LVD 的内部复位。但是，在发生 INTLVI 后，即使电源电压（ V_{DD} ）在不低于低电压检测电压（ V_{LVDL} ）的状态下恢复到高电压检测电压（ V_{LVDH} ）或者更高，也不产生中断请求信号。当使用 LVD 中断 & 复位模式时，必须按照“图 19-8 工作电压的确认 / 复位的设定步骤”和“图 19-9 中断 & 复位模式的初始设定步骤”所示的流程图的步骤进行设定。

LVD 中断 & 复位模式的内部复位信号和中断信号的产生时序如图 19-7 所示。

图 19-7 复位 & 中断信号的产生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (1/2)

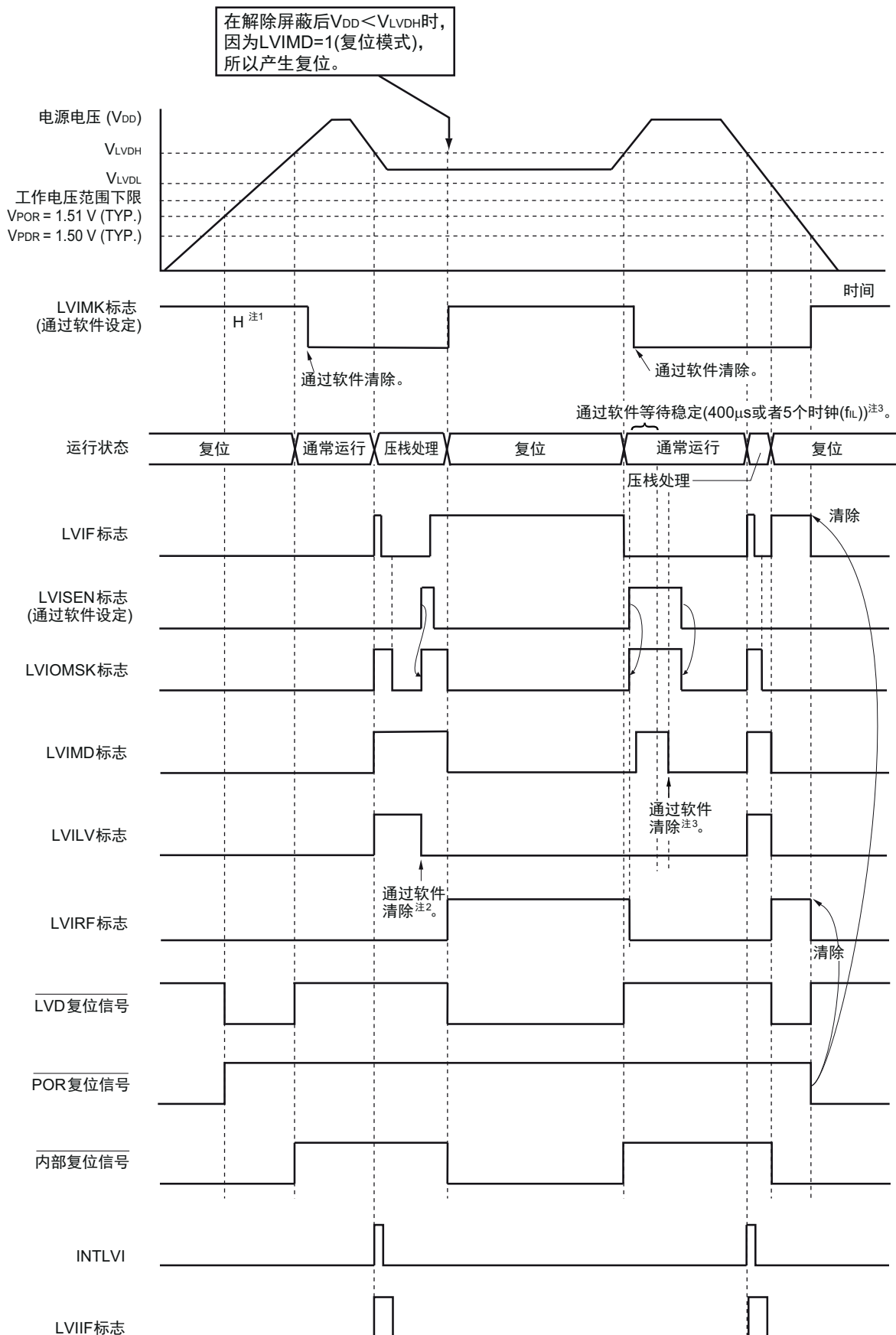


- 注
1. 在产生复位信号后，LVIMK 标志变为“1”。
 2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 19-8 工作电压的确认 / 复位的设定步骤”进行设定。
 3. 当使用中断 & 复位模式时，必须在解除复位后按照“图 19-9 中断 & 复位模式的初始设定步骤”进行设定。

备注 V_{POR} : POR 电源电压上升检测电压

V_{PDR} : POR 电源电压下降检测电压

图 19-7 中断 & 复位信号的产生时序 (选项字节的 LVIMDS1、LVIMDS0=1、0) (2/2)

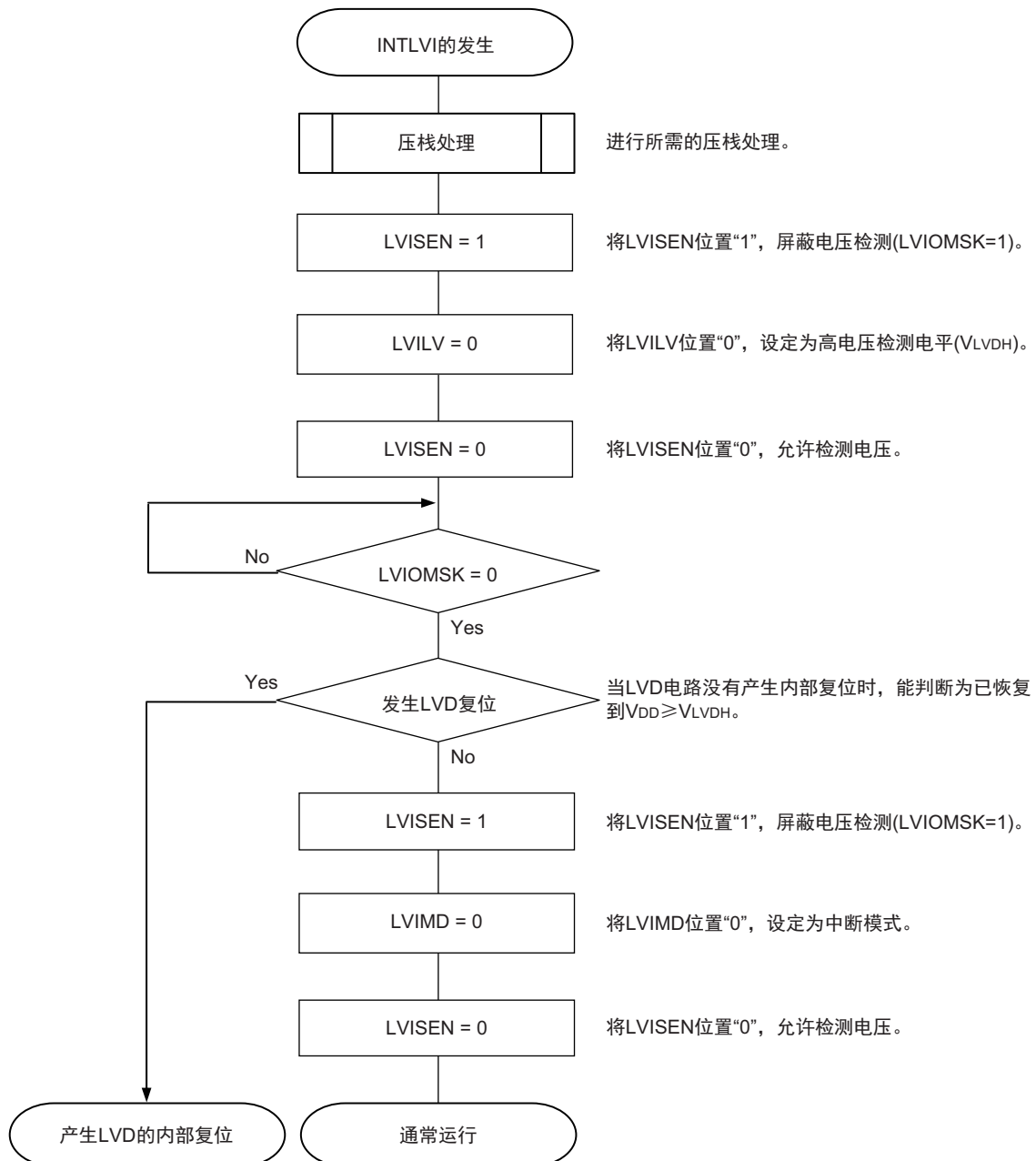


- 注 1. 在产生复位信号后，LVIMK 标志变为“1”。
2. 当使用中断 & 复位模式时，必须在发生中断后按照“图 19-8 工作电压的确认 / 复位的设定步骤”进行设定。
3. 当使用中断 & 复位模式时，必须在解除复位后按照“图 19-9 中断 & 复位模式的初始设定步骤”进行设定。

备注 V_{POR} : POR 电源电压上升检测电压

V_{PDR} : POR 电源电压下降检测电压

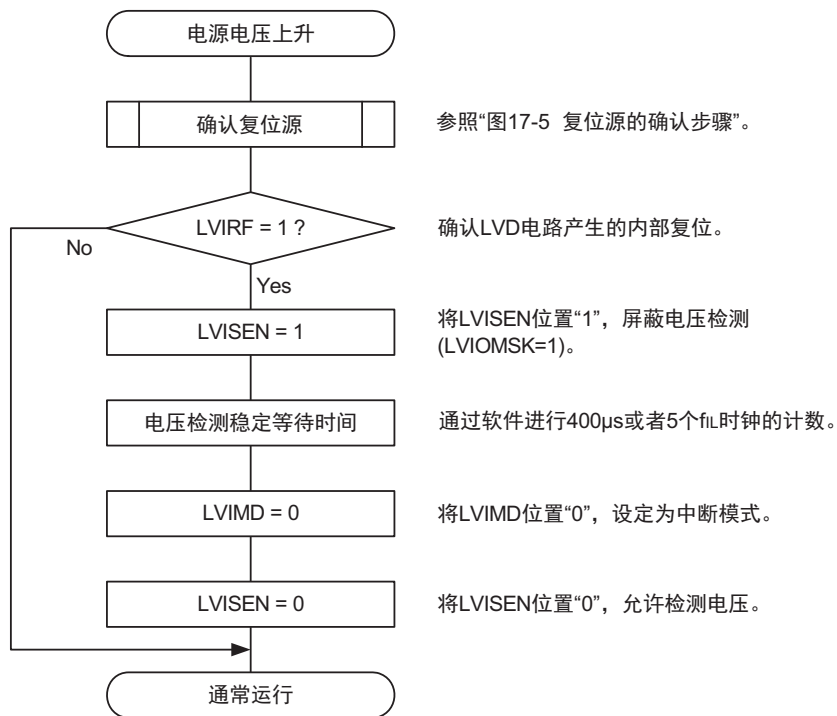
图 19-8 工作电压的确认 / 复位的设定步骤



如果设定中断 & 复位模式 (LVIMDS1、LVIMDS0=1、0)，就在解除 LVD 复位 (LVIRF=1) 后需要 $400\mu\text{s}$ 或者 5 个 f_{IL} 时钟的电压检测稳定等待时间。必须在等待电压检测稳定后将 LVIMD 位清“0”进行初始化。在电压检测稳定等待时间的计数过程中以及在改写 LVIMD 位时，必须将 LVISEN 位置“1”，屏蔽 LVD 产生的复位或者中断的产生。

中断 & 复位模式的初始设定步骤如图 19-9 所示。

图 19-9 中断 & 复位模式的初始设定步骤



备注 f_{IL} ：低速内部振荡器时钟频率

19.5 电压检测电路的注意事项

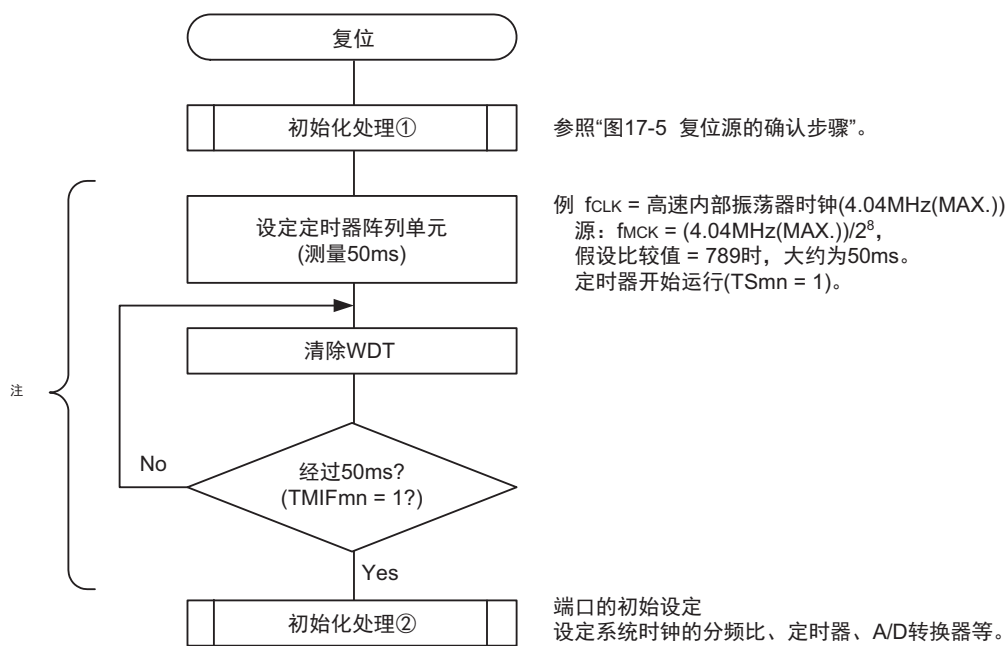
(1) 有关接通电源时的电压波动

对于电源电压 (V_{DD}) 在 LVD 检测电压附近发生一定时间波动的系统, 有可能重复进入复位状态和复位解除状态。能通过以下的处理, 任意设定解除复位到单片机开始运行的时间。

<处理>

在解除复位后, 必须通过使用定时器的软件计数器, 在等待各系统不同的电源电压波动时间后进行端口等的初始设定。

图 19-10 LVD 检测电压附近的电源电压波动不超过 50ms 时的软件处理例子



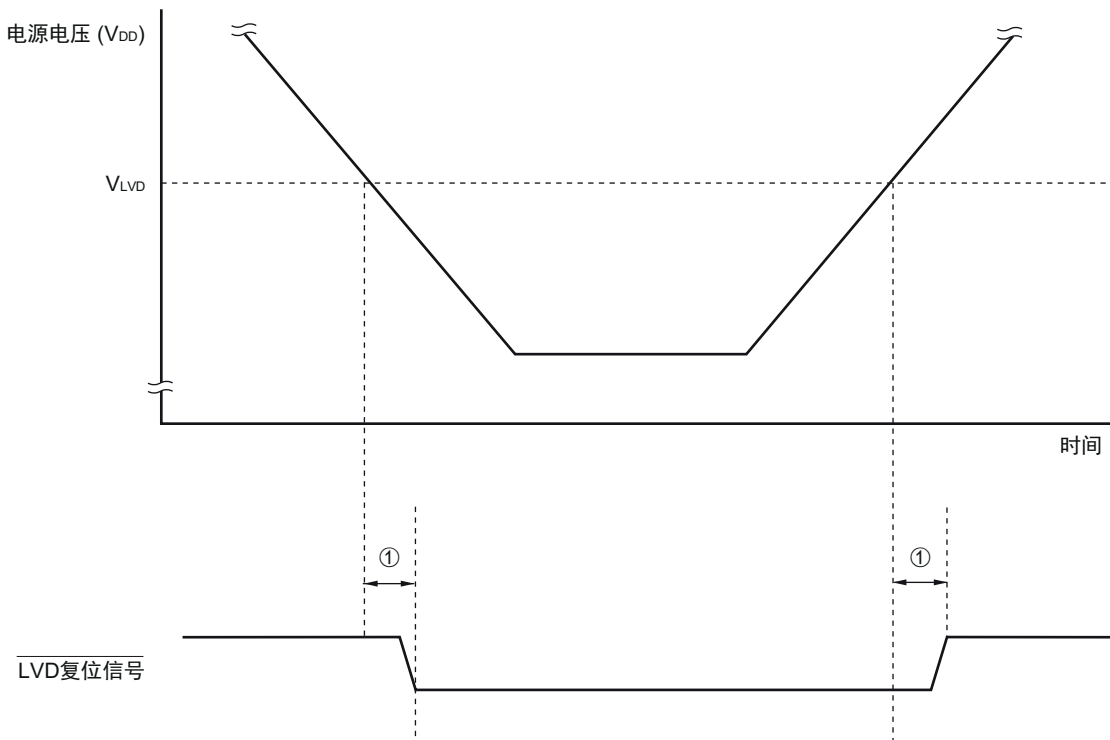
注 如果在此期间再次发生复位, 就不转移到初始化处理②。

备注 $m=0, 1, n=0 \sim 7$

(2) 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟

从满足电源电压 (V_{DD}) < LVD 检测电压 (V_{LVD}) 到产生 LVD 复位为止会发生延迟。同样, 从 LVD 检测电压 (V_{LVD}) \leq 电源电压 (V_{DD}) 到解除 LVD 复位为止也会发生延迟 (参照图 19-11)。

图 19-11 从产生 LVD 复位源到产生或者解除 LVD 复位的延迟



①: 检测延迟 (300 μ s (MAX.))

(3) 有关将 LVD 置为 OFF 时接通电源的情况

当将 LVD 置为 OFF 时, 必须使用 RESET 引脚的外部复位。

在进行外部复位时, 必须至少给 RESET 引脚输入 10 μ s 的低电平。如果在电源电压上升时进行外部复位, 就必须在给 RESET 引脚输入低电平后接通电源, 而且在“27.4 AC 特性”所示的工作电压范围内至少保持 10 μ s 的低电平, 然后输入高电平。

(4) 有关将 LVD 置为 OFF 并且设定为 LVD 中断模式时电源电压下降的情况

在将 LVD 置为 OFF 并且设定为 LVD 中断模式的情况下, 如果电源电压下降, 就必须在电源电压低于“27.4 AC 特性”所示的工作电压范围前, 转移到 STOP 模式, 或者通过外部复位置为复位状态。在重新开始运行时, 必须确认电源电压是否恢复到工作电压范围。

第 20 章 安全功能

20.1 安全功能的概要

为了对应 IEC60730 和 IEC61508 安全标准，本产品内置以下安全功能。
此安全功能的目的是通过单片机的自诊断，在检测到故障时安全地停止工作。

(1) 闪存 CRC 运算功能（高速 CRC、通用 CRC）

通过 CRC 运算检测闪存的数据错误。

能根据不同的用途和使用条件，分别使用以下 2 个 CRC。

- “高速 CRC”...在初始化程序中，能停止 CPU 的运行并且高速检查整个代码闪存区。
- “通用 CRC”...在 CPU 运行中，不限于代码闪存区而能用于多用途的检查。

(2) RAM 奇偶校验错误检测功能

在读 RAM 数据时，检测奇偶校验错误。

(3) RAM 保护功能

防止因 CPU 失控而改写 RAM 数据。

(4) SFR 保护功能

防止因 CPU 失控而改写 SFR。

(5) 非法存储器存取检测功能

检测对非法存储区（没有存储器的区域或者限制存取的区域）的非法存取。

(6) 频率检测功能

能使用定时器阵列单元进行 CPU/ 外围硬件时钟频率的自检。

(7) A/D 测试功能

能通过 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANI) 和内部基准电压的 A/D 转换进行 A/D 转换器的自检。

备注 有关对应 IEC60730 安全标准的安全功能的使用例子，请参照《RL78 MCU 系列的 IEC60730/60335 自检库应用说明》(R01AN1062、R01AN1296)。

20.2 安全功能使用的寄存器

安全功能的各功能使用以下寄存器。

寄存器名	安全功能的各功能
<ul style="list-style-type: none"> • 闪存 CRC 控制寄存器 (CRC0CTL) • 闪存 CRC 运算结果寄存器 (PGCRCL) 	闪存 CRC 运算功能 (高速 CRC)
<ul style="list-style-type: none"> • CRC 输入寄存器 (CRCIN) • CRC 数据寄存器 (CRCD) 	CRC 运算功能 (通用 CRC)
<ul style="list-style-type: none"> • RAM 奇偶校验错误控制寄存器 (RPECTL) 	RAM 奇偶校验错误检测功能
<ul style="list-style-type: none"> • 非法存储器存取检测控制寄存器 (IAWCTL) 	RAM 保护功能
	SFR 保护功能
	非法存储器存取检测功能
<ul style="list-style-type: none"> • 定时器输入选择寄存器 0 (TIS0) 	频率检测功能
<ul style="list-style-type: none"> • A/D 测试寄存器 (ADTES) 	A/D 测试功能

有关各寄存器的内容，在“20.3 安全功能的运行”中进行说明。

20.3 安全功能的运行

20.3.1 闪存 CRC 运算功能（高速 CRC）

IEC60730 标准要求确认闪存中的数据，并且建议 CRC 为确认手段。此高速 CRC 能在初始设定（初始化）程序中检查整个代码闪存区。只能通过 RAM 内的程序以主系统时钟的 HALT 模式执行高速 CRC。

高速 CRC 停止 CPU 的运行并且通过 1 个时钟从闪存读 32 位数据进行运算。因此，其特点是完成检查的时间较短。

CRC 生成多项式对应 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。

以 bit31→bit0 的 MSB 优先进行运算。

注意 在片上调式时，因为配置监视程序，所以 CRC 的运算结果不同。

备注 因为通用 CRC 为 LSB 优先，所以运算结果不同。

20.3.1.1 闪存 CRC 控制寄存器 (CRC0CTL)

这是设定高速 CRC 运算器的运行控制和运算范围的寄存器。
通过 1 位或者 8 位存储器操作指令设定 CRC0CTL 寄存器。
在产生复位信号后，此寄存器的值变为“00H”。

图 20-1 闪存 CRC 控制寄存器 (CRC0CTL) 的格式

地址: F02F0H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0
CRC0EN	高速 CRC 运算器的运行控制							
0	停止运行。							
1	通过执行 HALT 指令开始运算。							
FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	高速 CRC 的运算范围		
0	0	0	0	0	0	00000H ~ 03FFBH (16K-4 字节)		
0	0	0	0	0	1	00000H ~ 07FFBH (32K-4 字节)		
0	0	0	0	1	0	00000H ~ 0BFFBH (48K-4 字节)		
上述以外						禁止设定。		

备注 必须事先将用于比较的 CRC 运算结果期待值存入闪存的最后 4 字节，因此运算范围为减去 4 字节的范围。

20.3.1.2 闪存 CRC 运算结果寄存器 (PGCRCL)

这是保存高速 CRC 运算结果的寄存器。
通过 16 位存储器操作指令设定 PGCRCL 寄存器。
在产生复位信号后，此寄存器的值变为“0000H”。

图 20-2 闪存 CRC 运算结果寄存器 (PGCRCL) 的格式

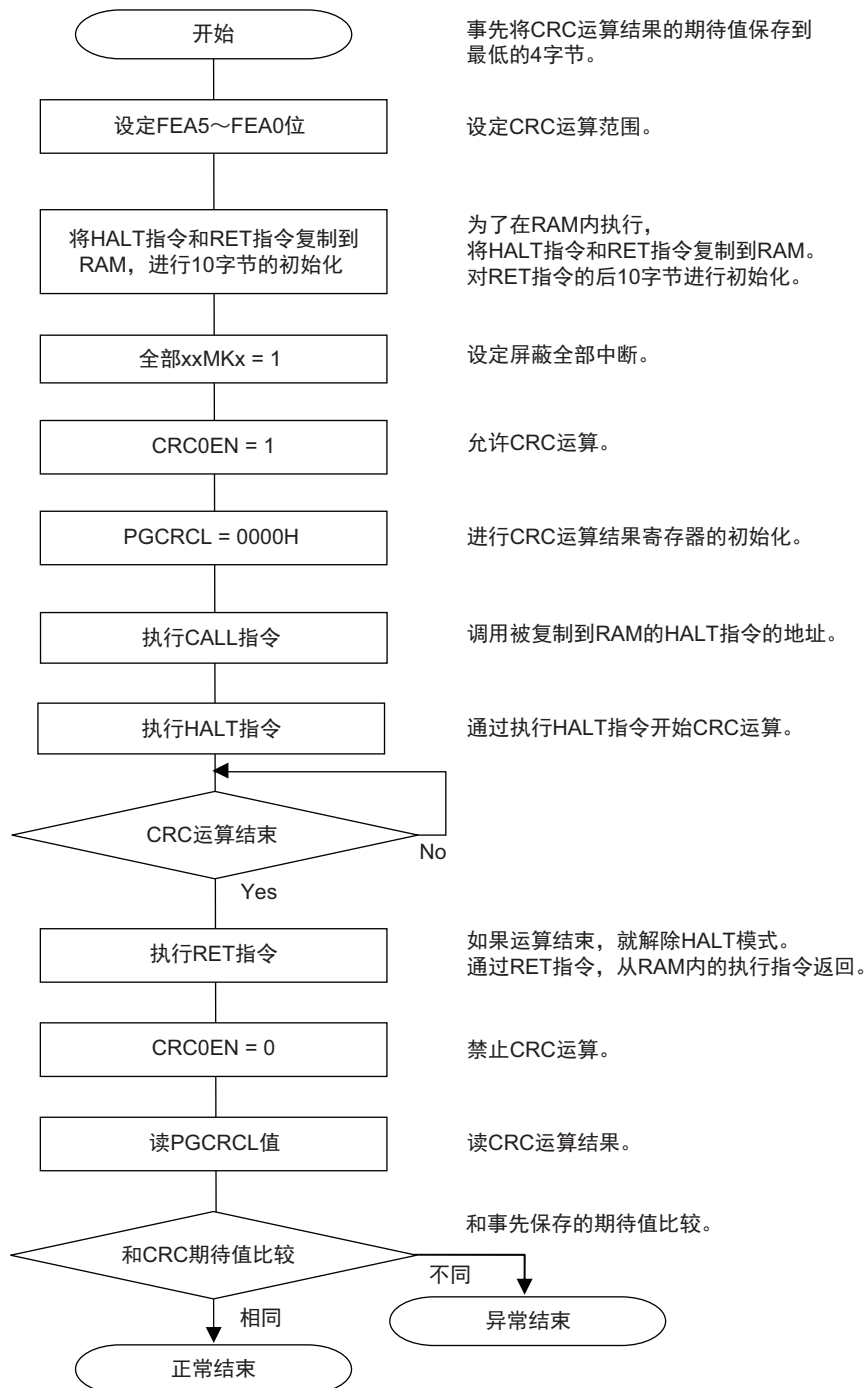
地址: F02F2H	复位后: 0000H	R/W								
符号	15	14	13	12	11	10	9	8		
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8		
	7	6	5	4	3	2	1	0		
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0		
PGCRC15 ~ 0	高速 CRC 的运算结果									
0000H ~ FFFFH	保存高速 CRC 的运算结果。									

注意 只有在 CRC0EN (CRC0CTL 寄存器的 bit7) 位为“1”时才能写 PGCRCL 寄存器。

闪存 CRC 运算功能 (高速 CRC) 的流程图如图 20-3 所示。

< 操作流程 >

图 20-3 闪存 CRC 运算功能（高速 CRC）的流程图



- 注意 1. 只以代码闪存为 CRC 运算的对象。
2. 必须将 CRC 运算的期待值保存在代码闪存中的运算范围后的区域。
3. 通过在 RAM 区执行 HALT 指令，CRC 运算变为有效。
必须在 RAM 区执行 HALT 指令。

CRC 运算的期待值能使用综合开发环境“CubeSuite+”进行计算。详细内容请参照《CubeSuite+ Development Environment User's Manual》。

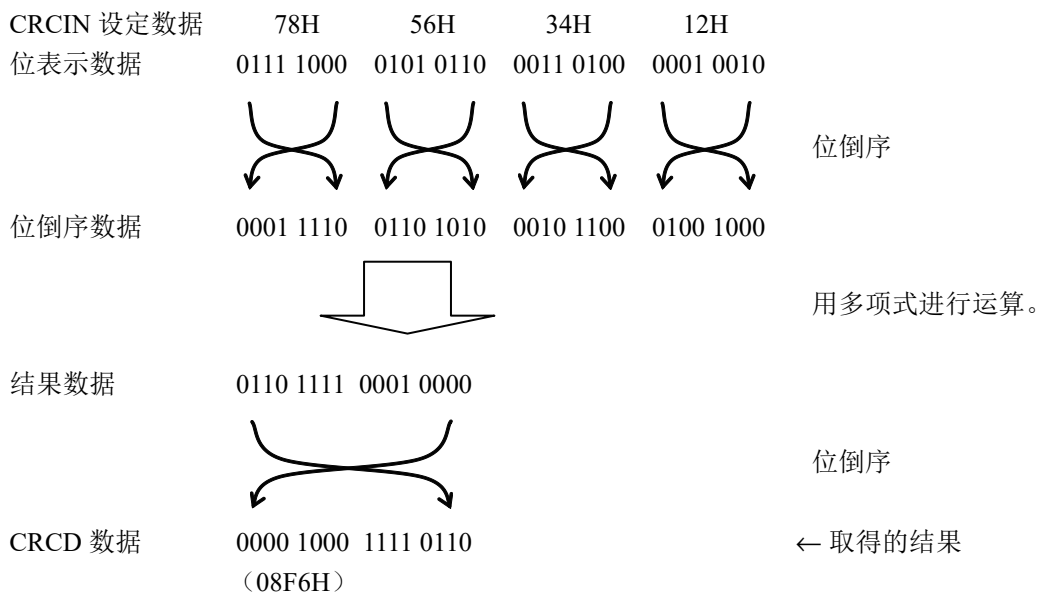
20.3.2 CRC 运算功能（通用 CRC）

为了必须保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。

此通用 CRC 能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 不限于代码闪存区而能用于多用途的检查。通过软件（用户程序）指定要确认的数据。HALT 模式中的 CRC 运算功能只能在 DMA 传送过程中使用。

在主系统时钟运行模式中，能使用 CRC 运算功能。

CRC 生成多项式使用 CRC-16-CCITT 的“ $X^{16}+X^{12}+X^5+1$ ”。因为考虑到是以 LSB 优先进行的通信，所以在将输入数据的位序颠倒后进行计算。例如，从 LSB 发送数据“12345678H”的情况，按照“78H”、“56H”、“34H”、“12H”的顺序给 CRCIN 寄存器写值，从 CRCD 寄存器得到“08F6H”的值。这是针对颠倒了数据“12345678H”的位序后的以下位序进行 CRC 运算的结果。



注意 在执行程序的过程中，因为调式程序将软件断点的设定行改写为断点指令，所以如果在 CRC 运算的对象区设定软件断点，CRC 的运算结果就不同。

20.3.2.1 CRC 输入寄存器（CRCIN）

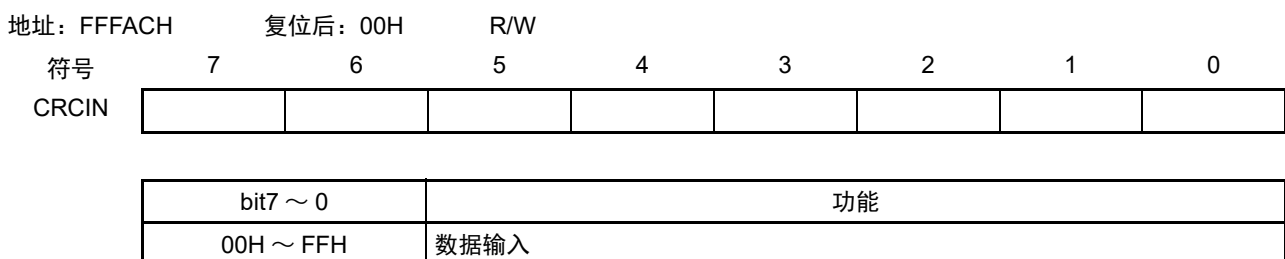
这是设定通用 CRC 的 CRC 计算数据的 8 位寄存器。

能设定的范围为“00H ~ FFH”。

通过 8 位存储器操作指令设定 CRCIN 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-4 CRC 输入寄存器（CRCIN）的格式



20.3.2.2 CRC 数据寄存器 (CRCD)

这是保存通用 CRC 运算结果的寄存器。

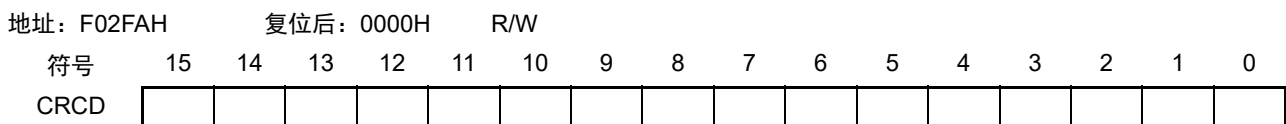
能设定的范围为“0000H ~ FFFFH”。

在写 CRCIN 寄存器后经过 1 个 CPU/ 外围硬件时钟 (f_{CLK})，将 CRC 运算结果保存到 CRCD 寄存器。

通过 16 位存储器操作指令设定 CRCD 寄存器。

在产生复位信号后，此寄存器的值变为“0000H”。

图 20-5 CRC 数据寄存器 (CRCD) 的格式

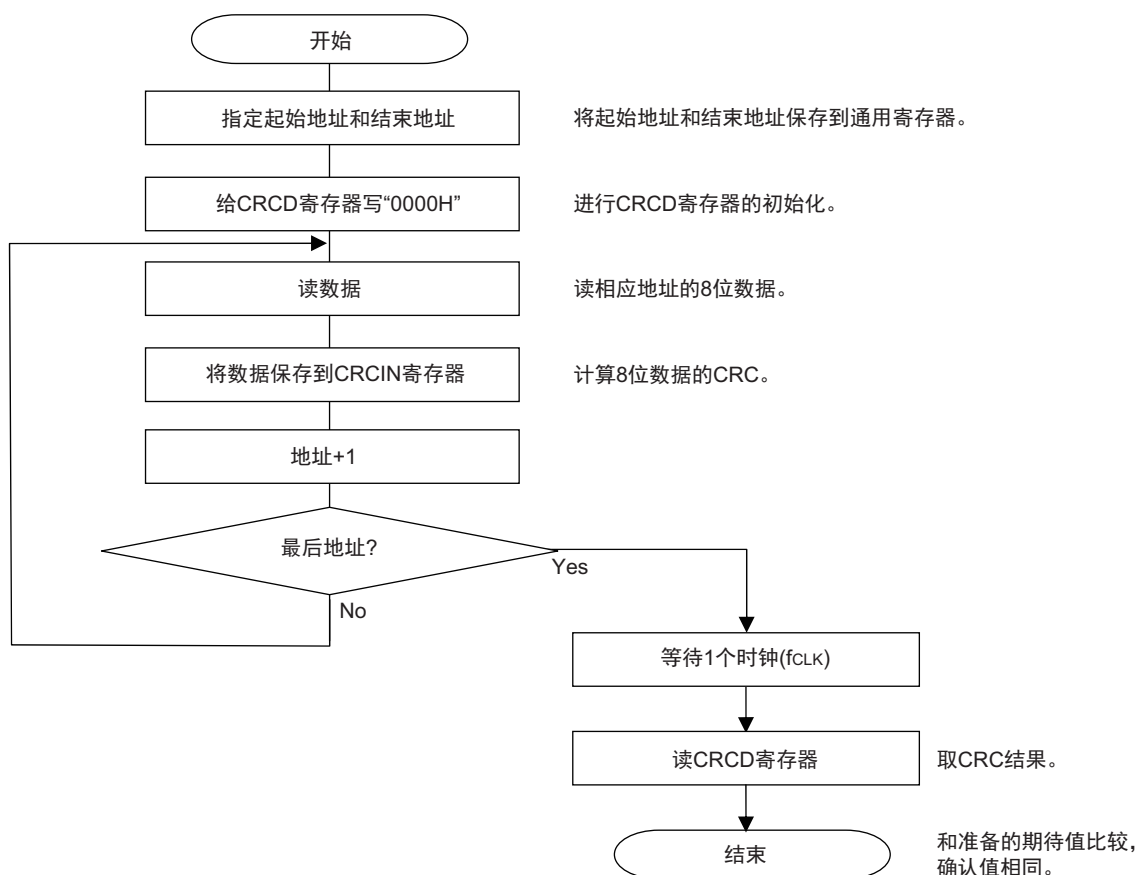


注意 1. 要读 CRCD 寄存器的写入值时，必须在写 CRCIN 寄存器前读 CRCD 寄存器。

2. 如果 CRCD 寄存器的写操作与运算结果的保存发生竞争，就忽视写操作。

<操作流程>

图 20-6 CRC 运算功能 (通用 CRC) 的流程图



20.3.3 RAM 奇偶校验错误检测功能

IEC60730 标准要求确认 RAM 数据。因此，本产品的 RAM 每 8 位附加 1 位奇偶校验位。RAM 奇偶校验错误检测功能在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位，并且能在发生奇偶校验错误时产生复位。

20.3.3.1 RAM 奇偶校验错误控制寄存器 (RPECTL)

此寄存器控制奇偶校验的错误确认位和因奇偶校验错误而产生复位。

通过 1 位或者 8 位存储器操作指令设定 RPECTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-7 RAM 奇偶校验错误控制寄存器 (RPECTL) 的格式

地址: F00F5H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	奇偶校验错误复位的屏蔽标志
0	允许产生奇偶校验错误复位。
1	禁止产生奇偶校验错误复位。

RPEF	奇偶校验错误状态标志
0	没有发生奇偶校验错误。
1	发生奇偶校验错误。

注意 在写数据时附加奇偶校验位，而在读数据时检查奇偶校验位。

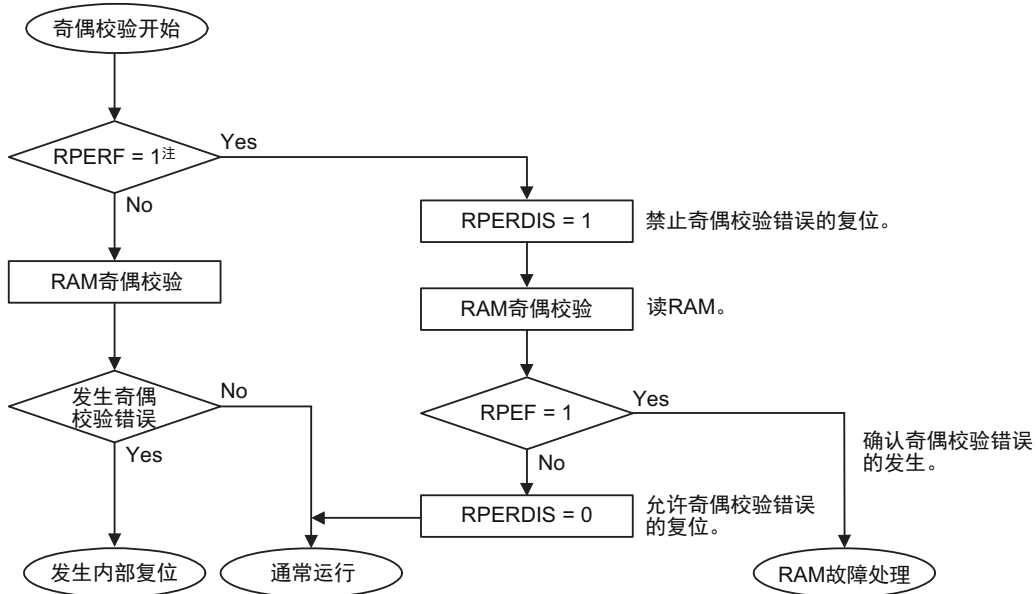
因此，当允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 时，必须在存取数据时并且在读数据前对“所用 RAM 区”进行初始化。

因为 RL78 为流水线运行，所以 CPU 进行预读，有可能因读所用 RAM 区前的未初始化 RAM 区而发生 RAM 奇偶校验错误。因此，当允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 时，必须在从 RAM 区执行指令时对“所用 RAM 区 +10 字节”的区域进行初始化；当允许产生 RAM 奇偶校验错误复位 (RPERDIS=0) 并且使用自编程功能时，必须在改写前对“要改写的 RAM 区 +10 字节”的区域进行初始化。

备注 1. 初始状态为允许产生奇偶校验错误复位 (RPERDIS=0)。

- 即使设定为禁止产生奇偶校验错误复位 (RPERDIS=1)，也在发生奇偶校验错误时将 RPEF 标志置“1”。如果在 RPEF 位为“1”的状态下设定为允许产生奇偶校验错误复位 (RPERDIS=0)，就在将 RPERDIS 清“0”时产生奇偶校验错误复位。
- 因 RAM 奇偶校验错误而将 RPECTL 寄存器的 RPEF 标志置“1”，通过写“0”或者全部复位源将 RPEF 标志清“0”。当 RPEF 标志为“1”时，即使读未发生奇偶校验错误的 RAM，RPEF 标志也保持“1”的状态。
- RAM 奇偶校验检测的范围不包括通用寄存器。

图 20-8 RAM 奇偶校验的流程



注 有关 RAM 奇偶错误的内部复位的确认，请参照“第 17 章 复位功能”。

20.3.4 RAM 保护功能

为了必须保证运行过程中的安全，IEC61508 标准要求即使 CPU 失控也需要保护存储在 RAM 的重要数据。此 RAM 保护功能用于保护被指定的 RAM 空间的数据。如果设定为 RAM 保护功能，指定空间的 RAM 写操作就无效，但是能正常读取。

20.3.4.1 非法存储器存取检测控制寄存器 (IAWCTL)

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。RAM 保护功能使用 GRAM1 位和 GRAM0 位。通过 8 位存储器操作指令设定 IAWCTL 寄存器。在产生复位信号后，此寄存器的值变为“00H”。

图 20-9 非法存储器存取检测控制寄存器 (IAWCTL) 的格式

地址: F0078H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM 保护空间注
0	0	无效。能写 RAM。
0	1	RAM 起始地址开始的 128 字节
1	0	RAM 起始地址开始的 256 字节
1	1	RAM 起始地址开始的 512 字节

注 RAM 的起始地址因产品内置的 RAM 容量而变。

20.3.5 SFR 保护功能

为了必须保证运行过程中的安全，IEC61508标准要求即使CPU失控也需要保护重要的SFR，使其免遭改写。

SFR 保护功能用于保护端口功能、中断功能、时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的数据。

如果设定为 SFR 保护功能，被保护的 SFR 的写操作就无效，但是能正常读取。

20.3.5.1 非法存储器存取检测控制寄存器（IAWCTL）

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。

SFR 保护功能使用 GPORT 位、GINT 位和 GCSC 位。

通过 8 位存储器操作指令设定 IAWCTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-10 非法存储器存取检测控制寄存器（IAWCTL）的格式

地址：F0078H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	端口功能的控制寄存器的保护
0	无效。能读写端口功能的控制寄存器。
1	有效。端口功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] PMxx、PUxx、PIMxx、POMxx、PMCxx、ADPC、PIOR 注

GINT	中断功能的寄存器的保护
0	无效。能读写中断功能的控制寄存器。
1	有效。中断功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] IFxx、MKxx、PRxx、EGPx、EGNx

GCSC	时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的保护
0	无效。能读写时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器。
1	有效。时钟控制功能、电压检测电路和 RAM 奇偶校验错误检测功能的控制寄存器的写操作无效，能读。 [被保护的 SFR] CMC、CSC、OSTS、CKC、PERx、OSMC、LVIM、LVIS、RPECTL

注 不保护 Pxx（端口寄存器）。

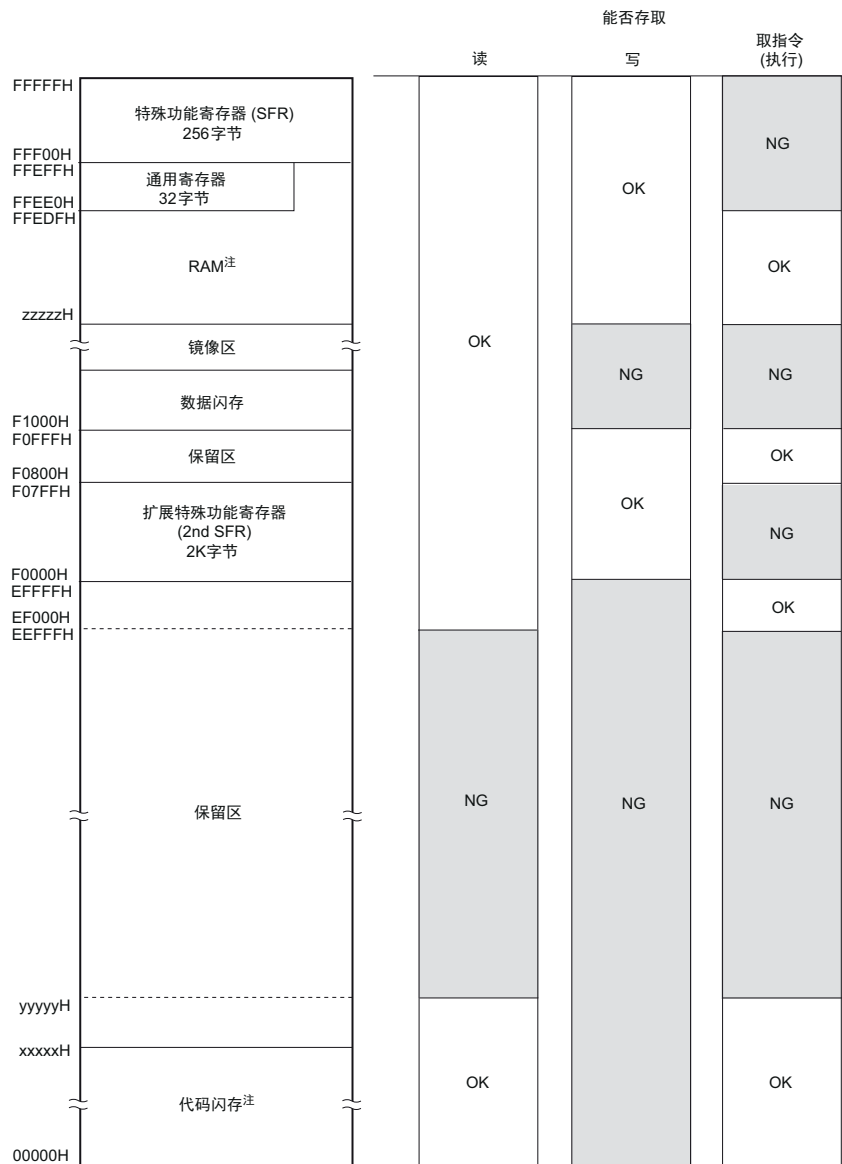
20.3.6 非法存储器存取检测功能

IEC60730 标准要求需要确认 CPU 和中断是否正常运行。

非法存储器存取检测功能在存取被规定的非法存取检测空间时产生复位。

非法存取检测空间为图 20-11 中记载为“NG”的范围。

图 20-11 非法存取检测空间



注 各产品的代码闪存、RAM 和检测最低位地址如下所示：

产品	代码闪存 (00000H ~ xxxxxH)	RAM (zzzzzH ~ FFEFFH)	读 / 取指令 (执行) 时的 检测最低位地址 (yyyyyH)
R7F0C906B2DFP-C R7F0C903B2DFP-C	16384×8 位 (00000H ~ 03FFFH)	2048×8 位 (FF700H ~ FFEFFH)	10000H
R7F0C907B2DFP-C R7F0C904B2DFP-C	32768×8 位 (00000H ~ 07FFFH)	2048×8 位 (FF700H ~ FFEFFH)	10000H
R7F0C908B2DFP-C R7F0C905B2DFP-C	49152×8 位 (00000H ~ 0BFFFH)	3072×8 位 (FF300H ~ FFEFFH)	10000H

20.3.6.1 非法存储器存取检测控制寄存器 (IAWCTL)

此寄存器控制是否允许检测非法存储器的存取以及控制 RAM/SFR 保护功能。

非法存储器存取检测功能使用 IAWEN 位。

通过 8 位存储器操作指令设定 IAWCTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 20-12 非法存储器存取检测控制寄存器 (IAWCTL) 的格式

地址: F0078H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC
IAWEN 注	非法存储器存取的检测控制							
0	非法存储器存取的检测无效。							
1	非法存储器存取的检测有效。							

注 IAWEN 位只有写“1”的操作有效，IAWEN 位为“1”后的写“0”的操作无效。

备注 当选项字节 (000C0H) 的 WDTON 位为“1” (允许看门狗定时器运行) 时，即使 IAWEN 位为“0”，非法存储器存取检测功能也有效。

20.3.7 频率检测功能

IEC60730 标准要求确认振荡频率是否正常。

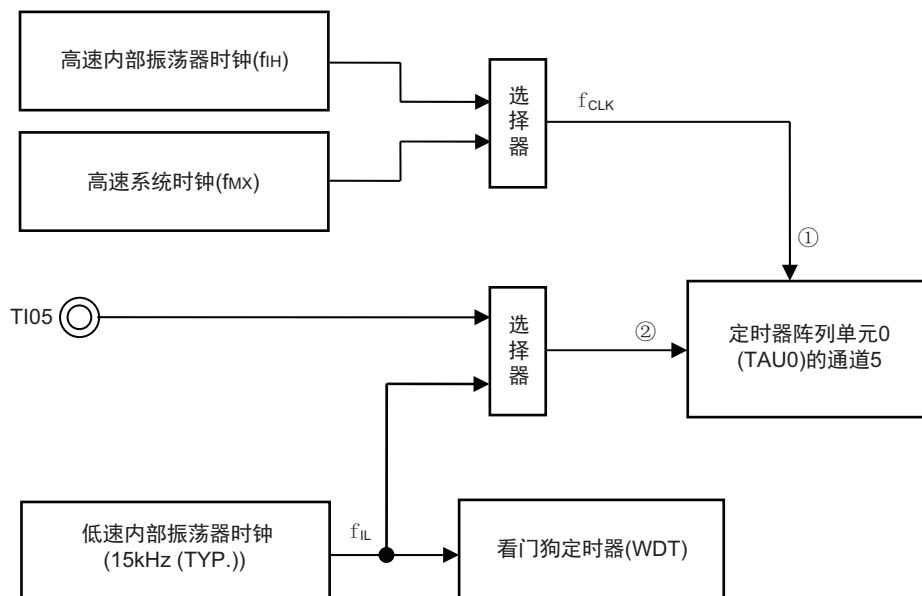
频率检测功能可使用 CPU/ 外围硬件的时钟频率 (f_{CLK})，并且能通过测量定时器阵列单元 0 (TAU0) 的通道 5 输入脉冲，判断 2 个时钟的比率关系是否正确。

但是，如果其中 1 个时钟或者 2 个时钟完全停止振荡，就不能判断 2 个时钟的比率关系。

<要比较的时钟>

- ① CPU/外围硬件的时钟频率 (f_{CLK}):
 - 高速内部振荡器时钟 (f_{IH})
 - 高速系统时钟 (f_{MX})
- ② 定时器阵列单元0的通道5输入:
 - 通道5的定时器输入 (TI05)
 - 低速内部振荡器时钟 (f_{IL} : 15kHz (TYP.))

图 20-13 频率检测功能的结构



当输入脉冲间隔的测量结果为异常值时，能判断为“时钟频率异常”。

有关输入脉冲间隔的测量方法，请参照“6.8.4 作为输入脉冲间隔测量的运行”。

20.3.7.1 定时器输入选择寄存器 0 (TIS0)

TIS0 寄存器选择定时器阵列单元 0 (TAU0) 的通道 5 的定时器输入。
通过 8 位存储器操作指令设定 TIS0 寄存器。
在产生复位信号后, 此寄存器的值变为“00H”。

图 20-14 定时器输入选择寄存器 0 (TIS0) 的格式

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

TIS02	TIS01	TIS00	通道 5 使用的定时器输入的选择
0	x	x	定时器输入引脚 (TI05) 的输入信号
1	0	0	低速内部振荡器时钟 (f_{IL})
上述以外			禁止设定。

备注 x: 忽略

20.3.8 A/D 测试功能

IEC60730 标准要求进行 A/D 转换器的测试。此 A/D 测试功能通过对 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (ANI) 和内部基准电压进行 A/D 转换, 确认 A/D 转换器是否正常运行。有关详细的确认方法, 请参照《安全功能 (A/D 测试) 的应用说明》(R01AN0955)。

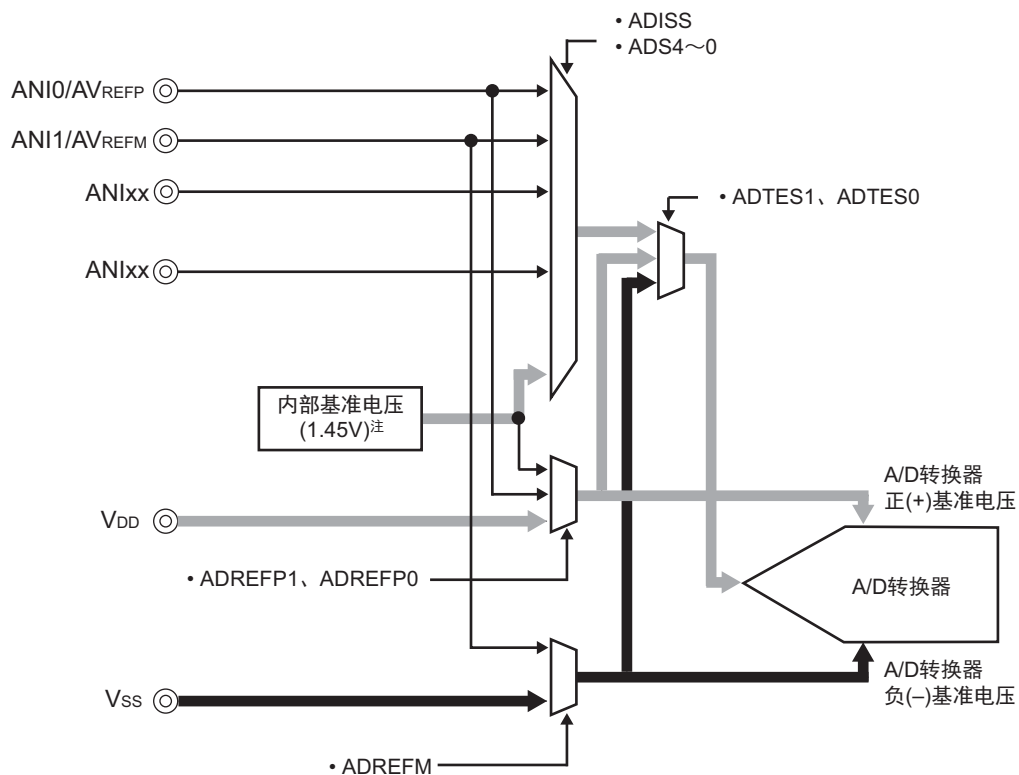
能通过以下步骤确认模拟多路转换器:

- ① 通过 ADTES 寄存器选择 ANIx 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ② 对 ANIx 引脚进行 A/D 转换 (转换结果 1-1)。
- ③ 通过 ADTES 寄存器选择 A/D 转换器的负 (-) 基准电压作为 A/D 转换对象 (ADTES1、ADTES0=1、0)。
- ④ 对 A/D 转换器的负 (-) 基准电压进行 A/D 转换 (转换结果 2-1)。
- ⑤ 通过 ADTES 寄存器选择 ANIx 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ⑥ 对 ANIx 引脚进行 A/D 转换 (转换结果 1-2)。
- ⑦ 通过 ADTES 寄存器选择 A/D 转换器的正 (+) 基准电压作为 A/D 转换对象 (ADTES1、ADTES0=1、1)。
- ⑧ 对 A/D 转换器的正 (+) 基准电压进行 A/D 转换 (转换结果 2-2)。
- ⑨ 通过 ADTES 寄存器选择 ANIx 引脚作为 A/D 转换对象 (ADTES1、ADTES0=0、0)。
- ⑩ 对 ANIx 引脚进行 A/D 转换 (转换结果 1-3)。
- ⑪ 确认“转换结果 1-1”、“转换结果 1-2”和“转换结果 1-3”相同。
- ⑫ 确认“转换结果 2-1”的 A/D 转换结果全部为“0”并且“转换结果 2-2”的 A/D 转换结果全部为“1”。

通过以上步骤, 能选择模拟多路转换器以及确认布线没有断线。

- 备注 1. 在①~⑩的转换过程中, 如果模拟输入电压可变, 就必须采用其他方法来确认模拟多路转换器。
2. 转换结果含有误差, 因此必须在比较转换结果时要适当考虑误差。

图 20-15 A/D 测试功能的结构



注 只有在 HS（高速主）模式中才能选择。

20.3.8.1 A/D 测试寄存器 (ADTES)

此寄存器选择 A/D 转换器的正 (+) 基准电压、负 (-) 基准电压、模拟输入通道 (AN_{Ixx}) 和内部基准电压 (1.45V) 作为 A/D 转换对象。

当用作 A/D 测试功能时, 进行以下的设定:

- 在测量零刻度时, 选择负 (-) 基准电压作为 A/D 转换对象。
- 在测量满刻度时, 选择正 (+) 基准电压作为 A/D 转换对象。

通过 8 位存储器操作指令设定 ADTES 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 20-16 A/D 测试寄存器 (ADTES) 的格式

地址: F0013H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D 转换对象
0	0	AN _{Ixx} / 内部基准电压 (1.45V) 注 (由模拟输入通道指定寄存器 (ADS) 进行设定)
1	0	负 (-) 基准电压 (通过 ADM2 寄存器的 ADREFM 位进行选择)
1	1	正 (+) 基准电压 (通过 ADM2 寄存器的 ADREFP1 位和 ADREFP0 位进行选择) 注
上述以外		禁止设定。

注 只有在 HS (高速主) 模式中才能选择内部基准电压 (1.45V)。

20.3.8.2 模拟输入通道指定寄存器 (ADS)

此寄存器指定 A/D 转换的模拟电压的输入通道。

要通过 A/D 测试功能测量 ANI_{xx} 或者内部基准电压 (1.45V) 时, 必须将 A/D 测试寄存器 (ADTES) 置“00H”。

通过 1 位或者 8 位存储器操作指令设定 ADS 寄存器。

在产生复位信号后, 此寄存器的值变为“00H”。

图 20-17 模拟输入通道指定寄存器 (ADS) 的格式

地址: FFF31H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○选项模式 (ADMD=0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	0	0	ANI0	P20/ANI0/AV _{REFP} 引脚
0	0	0	0	0	1	ANI1	P21/ANI1/AV _{REFM} 引脚
0	0	0	0	1	0	ANI2	P22/ANI2 引脚
0	0	0	0	1	1	ANI3	P23/ANI3 引脚
0	1	0	0	0	0	ANI16	P01/ANI16 引脚
0	1	0	0	0	1	ANI17	P00/ANI17 引脚
0	1	0	0	1	0	ANI18	P147/ANI18 引脚
0	1	0	0	1	1	ANI19	P120/ANI19 引脚
1	0	0	0	0	1	—	内部基准电压 (1.45V) 注
上述以外						禁止设定。	

注 只有在 HS (高速主) 模式中才能选择。

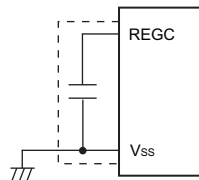
注意 1. 必须将 bit5 和 bit6 位置“0”。

- 对于由 ADPC 寄存器和 PMC 寄存器设定为模拟输入的端口, 必须通过端口模式寄存器 0、2、12、14 (PM0、PM2、PM12、PM14) 将其设定为输入模式。
- 对于由 A/D 端口配置寄存器 (ADPC) 设定为数字输入 / 输出的引脚, 不能通过 ADS 寄存器进行设定。
- 对于由端口模式控制寄存器 0、12、14 (PMC0、PMC12、PMC14) 设定为数字输入 / 输出的引脚, 不能通过 ADS 寄存器进行设定。
- 要改写 ADISS 位时, 必须在转换停止状态 (ADCS=0、ADCE=0) 下进行。
- 当将 AV_{REFP} 用作 A/D 转换器的正 (+) 基准电压时, 不能选择 ANI0 作为 A/D 转换通道。
- 当将 AV_{REFM} 用作 A/D 转换器的负 (-) 基准电压时, 不能选择 ANI1 作为 A/D 转换通道。
- 在将 ADISS 位置“1”的情况下, 不能将内部基准电压 (1.45V) 用作正 (+) 基准电压。另外, 在将 ADISS 位置“1”后, 不能使用第 1 次的转换结果。有关详细设定流程, 请参照“10.7.4 选择内部基准电压时的设定 (以软件触发模式、单次转换模式为例)”。
- 要转移到 STOP 模式时, 不能将 ADISS 位置“1”。如果将 ADISS 位置“1”, 就加上“27.3.2 电源电流特性”所示的 A/D 转换器基准电压电流 (I_{ADREF}) 的电流值。

第 21 章 稳压器

21.1 稳压器的概要

本产品内置使器件内部恒压工作的电路。为了使稳压器的输出电压稳定，必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚连接 V_{SS}。另外，为了稳定内部电压，必须使用特性好的电容器。



注意 必须尽量缩短上图虚线部分的布线。

稳压器的输出电压如表 21-1 所示。

表 21-1 稳压器的输出电压条件

模式	输出电压	条件
LV（低电压主）模式	1.8V	—
LS（低速主）模式		
HS（高速主）模式	1.8V	STOP 模式
	2.1V	上述以外（包含片上调试期间）注

注 在片上调试期间转移到 STOP 模式时，稳压器的输出电压保持 2.1V（不变为 1.8V）。

第 22 章 选项字节

22.1 选项字节的功能

本产品的闪存 000C0H ~ 000C3H 为选项字节区。

选项字节由用户选项字节（000C0H ~ 000C2H）和片上调试选项字节（000C3H）构成。

在接通电源或者复位启动时，自动参照选项字节进行指定功能的设定。在使用本产品时，必须通过选项字节进行以下功能的设定。对于没有配置功能的位，不能更改初始值。

要在自编程过程中使用引导交换功能时，因为 000C0H ~ 000C3H 被 010C0H ~ 010C3H 替换，所以必须给 010C0H ~ 010C3H 设定和 000C0H ~ 000C3H 相同的值。

注意 与是否使用各功能无关，必须设定选项字节。

22.1.1 用户选项字节（000C0H ~ 000C2H/010C0H ~ 010C2H）

(1) 000C0H/010C0H

- 看门狗定时器的运行
 - 允许或者禁止计数器的运行。
 - 在 HALT/STOP 模式中允许或者停止计数器的运行。
- 看门狗定时器的上溢时间的设定
- 看门狗定时器的窗口打开期间的设定
- 看门狗定时器的间隔中断的设定
 - 使用或者不使用间隔中断。

注意 在引导交换时，000C0H 被 010C0H 替换，因此必须给 010C0H 设定和 000C0H 相同的值。

(2) 000C1H/010C1H

- LVD 运行模式的设定
 - 中断&复位模式
 - 复位模式
 - 中断模式
 - LVD 为 OFF（使用 RESET 引脚的外部复位输入）。
- LVD 检测电平（ V_{LVDH} 、 V_{LVDL} 、 V_{LVD} ）的设定

注意 1. 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位位置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

2. 在引导交换时，000C1H 被 010C1H 替换，因此必须给 010C1H 设定和 000C1H 相同的值。

(3) 000C2H/010C2H

○ 闪存运行模式的设定

根据所用主系统时钟频率 (f_{MAIN}) 和电源电压 (V_{DD}) 进行设定。

- LV (低电压主) 模式
- LS (低速主) 模式
- HS (高速主) 模式

○ 高速内部振荡器的频率设定

- 从24MHz、16MHz、12MHz、8MHz、6MHz、4MHz、3MHz、2MHz、1MHz(TYP.)中选择。

注意 在引导交换时, 000C2H 被 010C2H 替换, 因此必须给 010C2H 设定和 000C2H 相同的值。

22.1.2 片上调试选项字节 (000C3H/010C3H)

○ 片上调试运行的控制

- 禁止或者允许片上调试运行。

○ 安全ID验证失败时闪存数据的处理

- 是否在片上调试安全ID验证失败时擦除闪存的数据。

注意 在引导交换时, 000C3H 被 010C3H 替换, 因此必须给 010C3H 设定和 000C3H 相同的值。

22.2 用户选项字节的格式

图 22-1 用户选项字节（000C0H/010C0H）的格式

地址：000C0H/010C0H 注 1

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	看门狗定时器间隔中断的使用 / 不使用
0	不使用间隔中断。
1	在达到上溢时间的 $75\%+1/2f_{IL}$ 时，产生间隔中断。

WINDOW1	WINDOW0	看门狗定时器的窗口打开期间注 2
0	0	禁止设定。
0	1	50%
1	0	75% 注 3
1	1	100%

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

WDCS2	WDCS1	WDCS0	看门狗定时器的上溢时间 ($f_{IL}=17.25\text{kHz}(\text{MAX.})$ 的情况)
0	0	0	$2^6/f_{IL}$ (3.71ms)
0	0	1	$2^7/f_{IL}$ (7.42ms)
0	1	0	$2^8/f_{IL}$ (14.84ms)
0	1	1	$2^9/f_{IL}$ (29.68ms)
1	0	0	$2^{11}/f_{IL}$ (118.72ms)
1	0	1	$2^{13}/f_{IL}$ (474.89ms)
1	1	0	$2^{14}/f_{IL}$ (949.79ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18ms)

WDSTBYON	看门狗定时器的计数器运行控制（HALT/STOP 模式）
0	在 HALT/STOP 模式中，停止计数器的运行注 2。
1	在 HALT/STOP 模式中，允许计数器的运行。

- 注 1. 在引导交换时，000C0H 被 010C0H 替换，因此必须给 010C0H 设定和 000C0H 相同的值。
2. 当 WDSTBYON 位为“0”时，与 WINDOW1 位和 WINDOW0 位的值无关，窗口打开期间为 100%。
3. 在将窗口打开期间设定为 75% 时，如果进行看门狗定时器的计数器清除（给 WDTE 写“ACH”），就必须在下表所示的计数器清除禁止期间外的时序进行看门狗定时器的间隔中断请求标志（WDTIIF）确认等。

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)	将窗口打开期间设定为 75% 时的 计数器清除禁止期间
0	0	0	$2^6/f_{IL}$ (3.71ms)	1.85 ~ 2.51ms
0	0	1	$2^7/f_{IL}$ (7.42ms)	3.71 ~ 5.02ms
0	1	0	$2^8/f_{IL}$ (14.84ms)	7.42 ~ 10.04ms
0	1	1	$2^9/f_{IL}$ (29.68ms)	14.84 ~ 20.08ms
1	0	0	$2^{11}/f_{IL}$ (118.72ms)	56.36 ~ 80.32ms
1	0	1	$2^{13}/f_{IL}$ (474.89ms)	237.44 ~ 321.26ms
1	1	0	$2^{14}/f_{IL}$ (949.79ms)	474.89 ~ 642.51ms
1	1	1	$2^{16}/f_{IL}$ (3799.18ms)	1899.59 ~ 2570.04ms

备注 f_{IL} : 低速内部振荡器的时钟频率

图 22-2 用户选项字节（000C1H/010C1H）的格式 (1/2)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD的设定（中断&复位模式）

检测电压			选项字节的设定值						
V _{LVDH}		V _{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降	下降						LVIMDS1	LVIMDS0
1.77V	1.73V	1.63V	0	0	0	1	0	1	0
1.88V	1.84V					0	1		
2.92V	2.86V					0	0		
1.98V	1.94V	1.84V	0	1	1	1	0	1	0
2.09V	2.04V					0	1		
3.13V	3.06V					0	0		
2.61V	2.55V	2.45V	1	0	1	1	0	1	0
2.71V	2.65V					0	1		
3.75V	3.67V					0	0		
2.92V	2.86V	2.75V	1	1	1	1	0	1	0
3.02V	2.96V					0	1		
4.06V	3.98V					0	0		
—			禁止设定上述以外的值。						

- LVD的设定（复位模式）

检测电压		选项字节的设定值						
V _{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
1.67V	1.63V	0	0	0	1	1	1	1
1.77V	1.73V		0	0	1	0		
1.88V	1.84V		0	1	1	1		
1.98V	1.94V		0	1	1	0		
2.09V	2.04V		0	1	0	1		
2.50V	2.45V		1	0	1	1		
2.61V	2.55V		1	0	1	0		
2.71V	2.65V		1	0	0	1		
2.81V	2.75V		1	1	1	1		
2.92V	2.86V		1	1	1	0		
3.02V	2.96V		1	1	0	1		
3.13V	3.06V		0	1	0	0		
3.75V	3.67V		1	0	0	0		
4.06V	3.98V		1	1	0	0		
—			禁止设定上述以外的值。					

注 在引导交换时，000C1H被010C1H替换，因此必须给010C1H设定和000C1H相同的值。

备注 1. 有关LVD电路的详细内容，请参照“第19章 电压检测电路”。

2. 检测电压是TYP.值。详细内容请参照“27.6.4 LVD电路特性”。

图 22-2 用户选项字节（000C1H/010C1H）的格式 (2/2)

地址：000C1H/010C1H注

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD 的设定（中断模式）

检测电压		选项字节的设定值								
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定			
上升	下降						LVIMDS1	LVIMDS0		
1.67V	1.63V	0	0	0	1	1	0	1		
1.77V	1.73V		0	0	1	0				
1.88V	1.84V		0	1	1	1				
1.98V	1.94V		0	1	1	0				
2.09V	2.04V		0	1	0	1				
2.50V	2.45V		1	0	1	1				
2.61V	2.55V		1	0	1	0				
2.71V	2.65V		1	0	0	1				
2.81V	2.75V		1	1	1	1				
2.92V	2.86V		1	1	1	0				
3.02V	2.96V		1	1	0	1				
3.13V	3.06V		0	1	0	0				
3.75V	3.67V		1	0	0	0				
4.06V	3.98V		1	1	0	0				
—			禁止设定上述以外的值。							

- LVD 为 OFF 时的设定（使用 $\overline{\text{RESET}}$ 引脚的外部复位输入）

检测电压		选项字节的设定值						
V_{LVDH}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	模式设定	
上升	下降						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		禁止设定上述以外的值。						

注 在引导交换时，000C1H 被 010C1H 替换，因此必须给 010C1H 设定和 000C1H 相同的值。

注意 1. 必须给 bit4 写“1”。

- 当电源电压上升时，必须在电源电压达到“27.4 AC 特性”所示的工作电压范围前，通过电压检测电路或者外部复位保持复位状态；当电源电压下降时，必须在电源电压低于工作电压范围前，转移到 STOP 模式，或者通过电压检测电路或外部复位位置为复位状态。工作电压范围取决于用户选项字节（000C2H/010C2H）的设定。

备注 1. ×：忽略

- 有关 LVD 电路的详细内容，请参照“第 19 章 电压检测电路”。
- 检测电压是 TYP. 值。详细内容请参照“27.6.4 LVD 电路特性”。

图 22-3 用户选项字节（000C2H/010C2H）的格式

地址：000C2H/010C2H注

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	闪存运行模式的设定		
			工作频率范围 (f_{MAIN})	工作电压范围 (V_{DD})
0	0	LV (低电压主) 模式	1MHz ~ 4MHz	1.6V ~ 5.5V
1	0	LS (低速主) 模式	1MHz ~ 8MHz	1.8V ~ 5.5V
1	1	HS (高速主) 模式	1MHz ~ 16MHz	2.4V ~ 5.5V
			1MHz ~ 24MHz	2.7V ~ 5.5V
上述以外		禁止设定。		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	高速内部振荡器的时钟频率
0	0	0	0	24MHz
1	0	0	1	16MHz
0	0	0	1	12MHz
1	0	1	0	8MHz
0	0	1	0	6MHz
1	0	1	1	4MHz
0	0	1	1	3MHz
1	1	0	0	2MHz
1	1	0	1	1MHz
上述以外				禁止设定。

注 在引导交换时，000C2H 被 010C2H 替换，因此必须给 010C2H 设定和 000C2H 相同的值。

注意 1. 必须给 bit5 和 bit4 写“10B”。

2. 工作频率范围和工作电压范围因闪存的各运行模式而不同。详细内容请参照“27.4 AC 特性”。

22.3 片上调试选项字节的格式

片上调试选项字节的格式如下所示。

图 22-4 片上调试选项字节（000C3H/010C3H）的格式

地址：000C3H/010C3H^注

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	片上调试运行的控制
0	0	禁止片上调试运行。
0	1	禁止设定。
1	0	允许片上调试运行。 在片上调试安全 ID 验证失败时，擦除闪存的数据。
1	1	允许片上调试运行。 在片上调试安全 ID 验证失败时，不擦除闪存的数据。

注 在引导交换时，000C3H 被 010C3H 替换，因此必须给 010C3H 设定和 000C3H 相同的值。

注意 只有 bit7 和 bit0（OCDENSET 和 OCDERSD）才能指定值。
必须给 bit6～1 写“000010B”。

备注 bit3～1 的值在使用片上调试功能时被改写，因此在设定后变为不定值。
但是，在设定时必须给 bit3～1 设定初始值（0、1、0）。

22.4 选项字节的设定

用户选项字节和片上调试选项字节除了在源程序中记述以外，还能通过链接选项进行设定。此时，即使在如下的源程序中有记述也优先链接选项的设定。

选项字节设定的软件记述例子如下所示。

OPT	CSEG	OPT_BYTE	
	DB	36H	; 不使用看门狗定时器的间隔中断。 ; 允许看门狗定时器运行。 ; 看门狗定时器的窗口打开期间为 50%。 ; 看门狗定时器的上溢时间为 $2^9/f_{IL}$ 。 ; 在 HALT/STOP 模式中，停止看门狗定时器的运行。
	DB	1AH	; V_{LVDL} 选择 1.63V。 ; V_{LVDH} 选择上升 1.77V、下降 1.73V。 ; 选择中断 & 复位模式作为 LVD 运行模式。
	DB	2DH	; 选择 LV（低电压主）模式作为闪存运行模式。 ; 选择高速内部振荡器的时钟频率 1MHz。
	DB	85H	; 允许片上调试运行，在安全 ID 验证失败时不擦除闪存的数据。

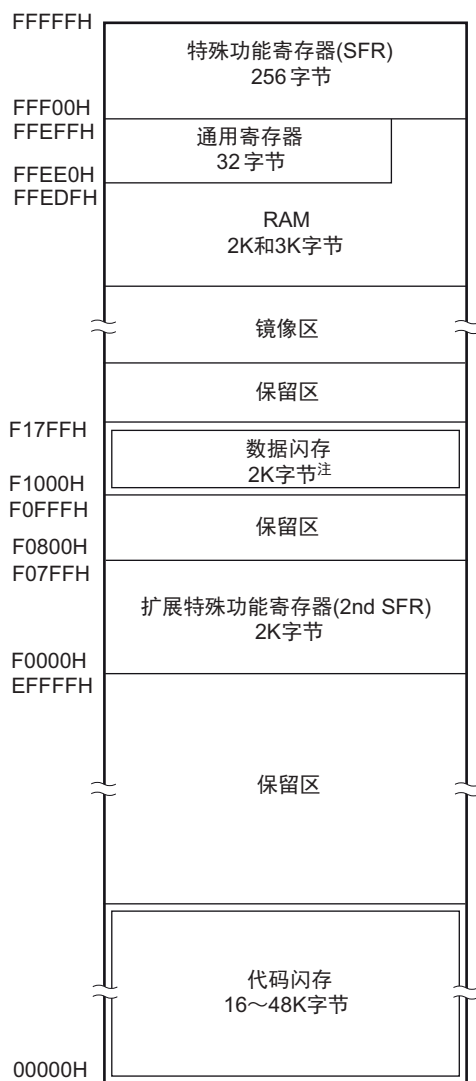
要在自编程过程中使用引导交换功能时，因为 000C0H ~ 000C3H 被 010C0H ~ 010C3H 替换，所以必须按照以下方式给 010C0H ~ 010C3H 设定和 000C0H ~ 000C3H 相同的值。

OPT2	CSEG	AT	010C0H	
	DB		36H	; 不使用看门狗定时器的间隔中断。 ; 允许看门狗定时器运行。 ; 看门狗定时器的窗口打开期间为 50%。 ; 看门狗定时器的上溢时间为 $2^9/f_{IL}$ 。 ; 在 HALT/STOP 模式中，停止看门狗定时器的运行。
	DB		1AH	; V_{LVDL} 选择 1.63V。 ; V_{LVDH} 选择上升 1.77V、下降 1.73V。 ; 选择中断 & 复位模式作为 LVD 运行模式。
	DB		2DH	; 选择 LV（低电压主）模式作为闪存运行模式。 ; 选择高速内部振荡器的时钟频率 1MHz。
	DB		85H	; 允许片上调试运行，在安全 ID 验证失败时不擦除闪存的数据。

注意 当通过汇编语言指定选项字节时，CSEG 伪指令的重定位属性名必须使用 OPT_BYTE。为了使用引导交换功能而给 010C0H ~ 010C3H 指定选项字节时，必须使用重定位属性 AT 来指定绝对地址。

第 23 章 闪存

RL78 微控制器内置在安装于电路板的状态下可进行编程、擦除和重新编程的闪存。闪存有可执行程序“代码闪存”和数据保存区的“数据闪存”。



注 在 R7F0C903B2、R7F0C904B2、R7F0C905B2 产品时，此区域为保留区。

闪存的编程方法如下：

能使用闪存编程器或者外部器件（UART 通信）对代码闪存进行串行编程或者通过自编程改写代码闪存。

- 使用闪存编程器的串行编程（参照 23.1）
能使用专用闪存编程器进行板上或者板外编程。
- 使用外部器件（UART 通信）的串行编程（参照 23.2）
能通过和外部器件（单片机或者 ASIC）的 UART 通信进行板上编程。
- 自编程（参照 23.6）
能让用户应用程序利用闪存自编程库进行代码闪存的自改写。

在用户程序执行过程中，能使用数据闪存库对数据闪存进行改写（后台操作）。有关数据闪存的存取和编程，请参照“23.8 数据闪存”。

23.1 使用闪存编程器的串行编程

能使用以下的专用闪存编程器对 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

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 Densai Machida Mfg. Co., Ltd 的产品。

表 23-1 本产品 and 专用闪存编程器的连线表

专用闪存编程器的连接引脚				引脚名	引脚号
					32 引脚
信号名		输入 / 输出	引脚功能		LQFP (7×7)
PG-FP5、 FL-PR5	E1 片上 调试仿真器			引脚号	
—	TOOL0	输入 / 输出	发送 / 接收信号	TOOL0/P40	1
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
/RESET	—	输出			
V_{DD}		输入 / 输出	V_{DD} 电压生成 / 电源监视	V_{DD}	8
GND		—	接地	V_{SS}	7
				REGC ^注	6
FLMD1	EMV_{DD}	—	TOOL0 引脚 驱动电源	V_{DD}	8

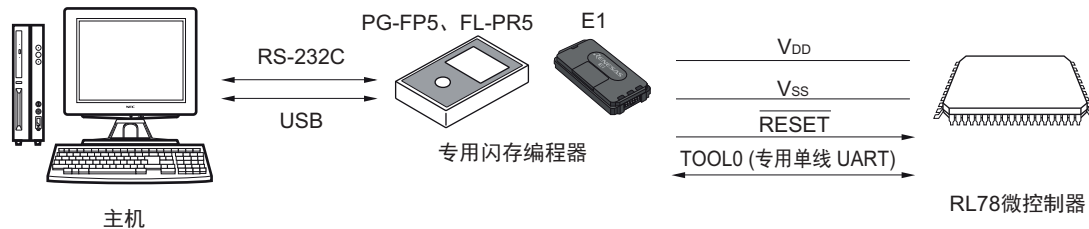
注 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。

备注 在使用闪存编程器进行编程时，此表中未记载的引脚可以开路。

23.1.1 编程环境

RL78 微控制器闪存的编程环境如下所示。

图 23-1 闪存的编程环境



需要控制专用闪存编程器的主机。

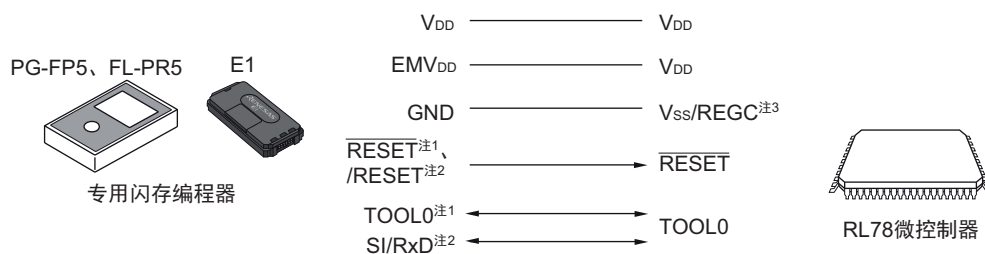
专用闪存编程器和 RL78 微控制器的接口使用 TOOL0 引脚，通过专用的单线 UART 进行编程和擦除。

23.1.2 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：1M、500k、250k、115.2kbps

图 23-2 和专用闪存编程器的通信



- 注
1. 这是使用 E1 片上调试仿真器的情况。
 2. 这是使用 PG-FP5 或者 FL-PR5 的情况。
 3. 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。

专用闪存编程器对 RL78 微控制器生成以下信号。详细内容请参照 PG-FP5、FL-PR5 或者 E1 片上调试仿真器的手册。

表 23-2 引脚连接一览表

专用闪存编程器			RL78 微控制器	
信号名		输入 / 输出	引脚功能	引脚名
PG-FP5、FL-PR5	E1 片上调试仿真器			
V_{DD}		输入 / 输出	V_{DD} 电压生成 / 电压监视	V_{DD}
GND		—	接地	V_{SS} 、REGC 注
FLMD1	EMV_{DD}	—	TOOL0 引脚驱动电源	V_{DD}
/RESET	—	输出	复位信号	\overline{RESET}
—	\overline{RESET}	输出		
—	TOOL0	输入 / 输出	发送 / 接收信号	TOOL0
SI/RxD	—	输入 / 输出	发送 / 接收信号	

注 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。

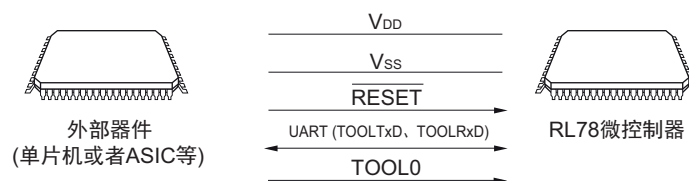
23.2 使用外部器件（内置 UART）的串行编程

能使用 RL78 微控制器和连接 UART 的外部器件（单片机或者 ASIC）对内部闪存进行数据的板上编程。有关用户闪存编程器的开发，请参照 RL78 微控制器（RL78 协议 A）编程器的应用说明（R01AN0815）。

23.2.1 编程环境

RL78 微控制器闪存的编程环境如下所示。

图 23-3 闪存的编程环境



外部器件对 RL78 微控制器进行板上编程和擦除，但是不能进行板外编程。

23.2.2 通信方式

外部器件和 RL78 微控制器的通信使用 RL78 微控制器的 TOOLTxD 引脚和 TOOLRxD 引脚，通过专用的 UART 进行串行通信。

传送速率：1M、500k、250k、115.2kbps

图 23-4 和外部器件的通信



注 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。

外部器件对 RL78 微控制器生成以下信号。

表 23-3 引脚连接一览表

外部器件			RL78 微控制器
信号名	输入 / 输出	引脚功能	引脚名
V_{DD}	输入 / 输出	V_{DD} 电压生成 / 电压监视	V_{DD}
GND	—	接地	V_{SS} 、REGC 注
RESETOUT	输出	复位信号输出	\overline{RESET}
RxD	输入	接收信号	TOOLTxD
TxD	输出	发送信号	TOOLRxD
PORT	输出	模式信号	TOOL0

注 必须通过电容器（0.47 ~ 1 μ F）将 REGC 引脚接地。

23.3 电路板上的引脚处理

要使用闪存编程器进行板上编程时，必须在目标系统上设置连接专用闪存编程器的连接器，并且在电路板上设置从通常运行模式到闪存编程模式的切换功能。

如果转移到闪存编程模式，闪存编程时未使用的全部引脚就保持刚复位后的相同状态。因此，如果外部器件不允许刚复位后的状态，就需要进行引脚处理。

备注 有关闪存编程模式，请参照“23.4.2 闪存编程模式”。

23.3.1 P40/TOOL0 引脚

在闪存编程模式中，必须在外部通过 $1\text{k}\Omega$ 电阻将此引脚上拉并且连接到专用闪存编程器。当此引脚用作端口引脚时，必须按照以下方法使用此引脚。

用作输入引脚：在解除外部复位时的 t_{HD} 期间，不能输入低电平。但是，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} ：这是在进入闪存编程模式时解除外部复位或者内部复位后保持 TOOL0 引脚低电平的时间。请参照“27.10 闪存编程模式的转移时序”。

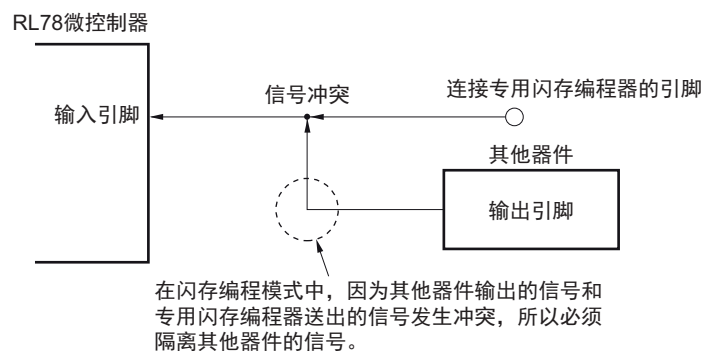
2. 因为 RL78 微控制器和专用闪存编程器的通信使用单线 UART (TOOL0 引脚)，所以不使用 SAU 引脚和 IICA 引脚。

23.3.2 $\overline{\text{RESET}}$ 引脚

如果在电路板上将专用闪存编程器和外部器件的复位信号连接到与复位信号生成电路相连的 $\overline{\text{RESET}}$ 引脚，就会发生信号冲突。为了避免此信号冲突，必须隔离与复位信号生成电路的连接。

在闪存编程模式的期间，如果从用户系统输入复位信号，就不能进行正常的编程，因此除了专用闪存编程器和外部器件的复位信号以外，不能输入其他复位信号。

图 23-5 信号冲突 ($\overline{\text{RESET}}$ 引脚)



23.3.3 端口引脚

如果转移到闪存编程模式，闪存编程时未使用的全部引脚就保持刚复位后的相同状态。因此，如果连接各端口的外部器件不允许刚复位后的端口状态，就需要通过电阻将引脚连接 V_{DD} 或者 V_{SS} 进行引脚处理。

23.3.4 REGC 引脚

和通常运行模式相同，必须通过特性好的电容器（ $0.47 \sim 1\mu\text{F}$ ）将 REGC 引脚连接 GND。另外，为了稳定内部电压，必须使用特性好的电容器。

23.3.5 X1 引脚和 X2 引脚

X1、X2 的连接方法必须和通常运行模式相同。

备注 在闪存编程模式中，使用高速内部振荡器时钟（ f_{IH} ）。

23.3.6 电源

当使用闪存编程器输出的电源时，必须将 V_{DD} 引脚连接闪存编程器的 V_{DD} ，并且将 V_{SS} 引脚连接闪存编程器的 GND。

当使用电路板上的电源时，必须按照通常运行模式进行连接。

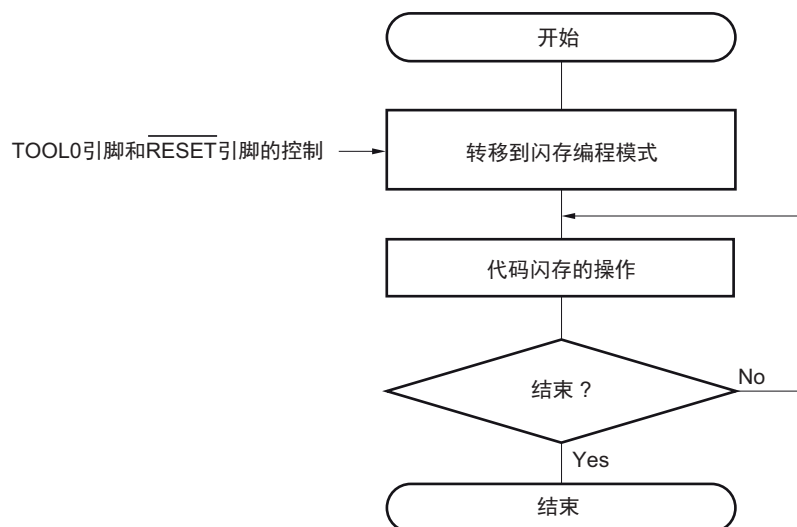
但是，在使用闪存编程器进行编程时，即使使用电路板上的电源，也必须将 V_{DD} 引脚和 V_{SS} 引脚分别连接闪存编程器的 V_{DD} 和 GND，以便通过闪存编程器监视电压。

23.4 串行编程方法

23.4.1 串行编程的步骤

通过串行编程进行代码闪存改写的流程如下所示。

图 23-6 代码闪存的操作步骤



23.4.2 闪存编程模式

在通过串行编程改写代码闪存的内容时，必须设定为闪存编程模式。要转移到闪存编程模式时，必须按照以下方法进行。

<使用专用闪存编程器进行串行编程的情况>

将 RL78 微控制器和专用闪存编程器连接。通过和专用闪存编程器的通信，自动转移到闪存编程模式。

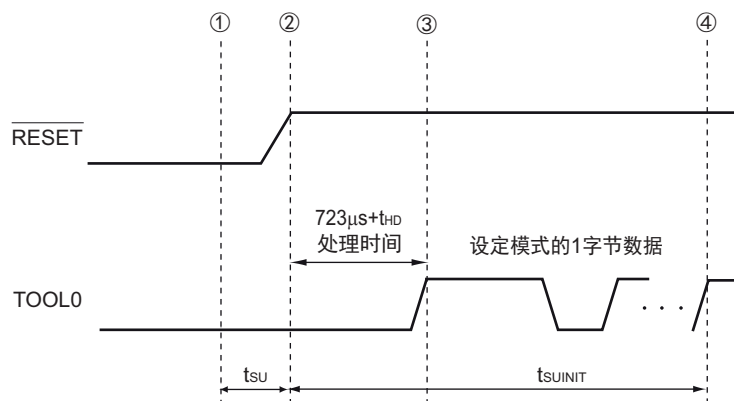
<使用外部器件（UART 通信）进行串行编程的情况>

在将 TOOL0 引脚置为低电平后解除复位（参照表 23-4），然后按照图 23-7 所示的①~④的步骤转移到闪存编程模式。详细内容请参照 RL78 微控制器（RL78 协议 A）编程器的应用说明（R01AN0815）。

表 23-4 解除复位时的 TOOL0 引脚和运行模式的关系

TOOL0	运行模式
V _{DD}	通常运行模式
0V	闪存编程模式

图 23-7 闪存编程模式的进入



- ① 给 TOOL0 引脚输入低电平。
- ② 解除外部复位（在此之前需要解除 POR 和 LVD 的复位）。
- ③ 解除 TOOL0 引脚的低电平。
- ④ 通过 UART 接收来完成波特率的设定。

备注 t_{SUINIT}: 在此区间，必须在解除外部复位后的 100ms 之内完成初始设定的通信。

t_{SU}: 这是从将 TOOL0 引脚置为低电平到解除外部复位为止的时间。

t_{HD}: 这是在解除外部复位或者内部复位后保持 TOOL0 引脚低电平的时间（闪存固件处理时间除外）。

详细内容请参照“27.10 闪存编程模式的转移时序”。

闪存编程模式有宽电压模式和全速模式共 2 种编程模式。根据编程时提供给单片机的电源电压值以及进入闪存编程模式时的用户选项字节的设定信息，决定模式的选择。

在使用专用闪存编程器进行串行编程时，通过 GUI 进行的电压设定自动选择模式。

表 23-5 编程模式和可进行编程、擦除或者校验的电压

电源电压 (V_{DD})	进入闪存编程模式时的选项字节的设定		闪存的改写模式
	闪存运行模式	工作频率	
$2.7V \leq V_{DD} \leq 5.5V$	空白状态		全速模式
	HS (高速主) 模式	1MHz ~ 24MHz	全速模式
	LS (低速主) 模式	1MHz ~ 8MHz	宽电压模式
	LV (低电压主) 模式	1MHz ~ 4MHz	宽电压模式
$2.4V \leq V_{DD} < 2.7V$	空白状态		全速模式
	HS (高速主) 模式	1MHz ~ 16MHz	全速模式
	LS (低速主) 模式	1MHz ~ 8MHz	宽电压模式
	LV (低电压主) 模式	1MHz ~ 4MHz	宽电压模式
$1.8V \leq V_{DD} < 2.4V$	空白状态		宽电压模式
	LS (低速主) 模式	1MHz ~ 8MHz	宽电压模式
	LV (低电压主) 模式	1MHz ~ 4MHz	宽电压模式

备注 1. 在同时使用宽电压模式和全速模式的情况下进行编程、擦除或者校验时，没有任何限制事项。

2. 有关通信命令的详细内容，请参照“23.4.4 通信命令”。

23.4.3 通信方式

RL78 微控制器的通信方式如下所示。

表 23-6 通信方式

通信方式	Standard 设定 ^{注 1}				使用的引脚
	Port	Speed ^{注 2}	Frequency	Multiply Rate	
单线 UART (使用闪存编程器 或者外部器件)	UART	115200bps、 250000bps、 500000bps、 1Mbps	—	—	TOOL0
专用 UART (使用外部器件)	UART	115200bps、 250000bps、 500000bps、 1Mbps	—	—	TOOLTxD、 TOOLRxD

注 1. 这是与闪存编程器 GUI Standard 设定相关的设定项目。

2. 除了波特率误差以外，信号波形的失真等还会影响 UART 通信，因此必须在评估后使用。

23.4.4 通信命令

RL78 微控制器通过表 23-7 所示的命令进行串行编程。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”，进行该命令对应的各种功能处理。详细内容请参照 RL78 微控制器（RL78 协议 A）编程器的应用说明（R01AN0815）。

表 23-7 闪存控制命令

分类	命令名称	功能
校验	Verify	将指定闪存区的内容和从编程器发送的数据进行比较。
擦除	Block Erase	擦除指定的闪存区。
空白检查	Block Blank Check	检查指定块的闪存擦除状态。
编程	Programming	将数据写到指定的闪存区注。
信息取得	Silicon Signature	取得 RL78 微控制器的信息（例如：产品名、闪存结构、用于编程的固件版本等）。
	Checksum	取得指定区域的校验和。
安全	Security Set	设定安全信息。
	Security Get	取得安全信息。
	Security Release	解除禁止编程的设定。
其他	Reset	用于通信的同步检测。
	Baud Rate Set	设定选择 UART 时的波特率。

注 必须确认编程区中还没有进行数据编程。因为在设定禁止块擦除后无法进行擦除，所以在数据没有被擦除时，不能进行数据编程。

能通过执行“Silicon Signature”命令来获取产品信息（产品名、固件版本等）。

特征数据一览表和特征数据的例子分别如表 23-8 和表 23-9 所示。

表 23-8 特征数据一览表

字段名	内容	发送字节数
器件代码	分配给器件的序列号	3 字节
器件名	器件名（ASCII 码）	10 字节
代码闪存区的结束地址	代码闪存区的结束地址 （从地址的低位开始发送。 例：00000H ~ 0BFFFH（48KB）→FFH、BFH、00H）	3 字节
数据闪存区的结束地址	数据闪存区的结束地址 （从地址的低位开始发送。 例：F1000H ~ F17FFH（2KB）→FFH、17H、0FH）	3 字节
固件版本	用于编程的固件版本信息 （从版本的高位开始发送。例：Ver.1.23→01H、02H、03H）	3 字节

表 23-9 特征数据的例子

字段名	内容	发送字节数	数据（十六进制）
器件代码	RL78 协议 A	3 字节	10 00 06
器件名	R7F0C908	10 字节	52 = "R" 37 = "7" 46 = "F" 30 = "0" 43 = "C" 39 = "9" 30 = "0" 38 = "8" 20 = "" 20 = ""
代码闪存区的结束地址	代码闪存区 00000H ~ 0BFFFH (48KB)	3 字节	FF BF 00
数据闪存区的结束地址	数据闪存区 F1000H ~ F17FFH (2KB)	3 字节	FF 17 0F
固件版本	Ver.1.23	3 字节	01 02 03

23.5 使用 PG-FP5 时的各命令处理时间（参考值）

在将 PG-FP5 用作专用闪存编程器时，各命令处理时间（参考值）如下所示：

表 23-10 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									
	16K 字节	32K 字节	48K 字节	64K 字节	96K 字节	128K 字节	192K 字节	256K 字节	384K 字节	512K 字节
擦除	1s	1s	1s	1.5s	1.5s	2s	2s	2.5s	3s	4s
编程	1.5s	1.5s	2s	2.5s	3s	3.5s	5s	6s	8.5s	11s
校验	1.5s	1.5s	2s	2s	3s	3.5s	4.5s	5.5s	8s	10.5s
擦除后的编程	1.5s	2s	2.5s	3s	4s	4.5s	6.5s	8s	11s	14.5s

备注 命令处理时间（参考值）是 TYP. 值，条件如下：

Port: TOOL0（单线 UART）

Speed: 1000000bps

Mode: 全速模式（闪存运行模式：HS（高速主）模式）

23.6 自编程

RL78 微控制器支持自编程功能，能通过用户程序改写代码闪存。因为此功能可让用户应用程序利用闪存自编程库来改写代码闪存，所以能在现场进行程序升级等。

- 注意 1. 为了在自编程过程中禁止中断，必须和通常运行模式相同，在通过 DI 指令将 IE 标志清“0”的状态下执行闪存自编程库。在允许中断的情况下，必须在通过 EI 指令将 IE 标志置“1”的状态下将接受中断的中断屏蔽标志清“0”，然后执行闪存自编程库。
2. 在自编程过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30 μ s 后执行闪存自编程库。

- 备注 1. 有关自编程功能的详细内容，请参照《RL78 Family Flash Self Programming Library Type01 User's Manual》（R01US0050）。
2. 有关自编程的执行处理时间，请参照闪存自编程库工具附属的使用时的注意点。

自编程功能有宽电压模式和全速模式共 2 种闪存编程模式。

必须根据选项字节 000C2H 的 CMODE1 位和 CMODE0 位设定的闪存运行模式进行以下任意模式的设定。

当设定为 HS（高速主）模式时，必须设定全速模式；当设定为 LS（低速主）模式或者 LV（低电压主）模式时，必须设定宽电压模式。

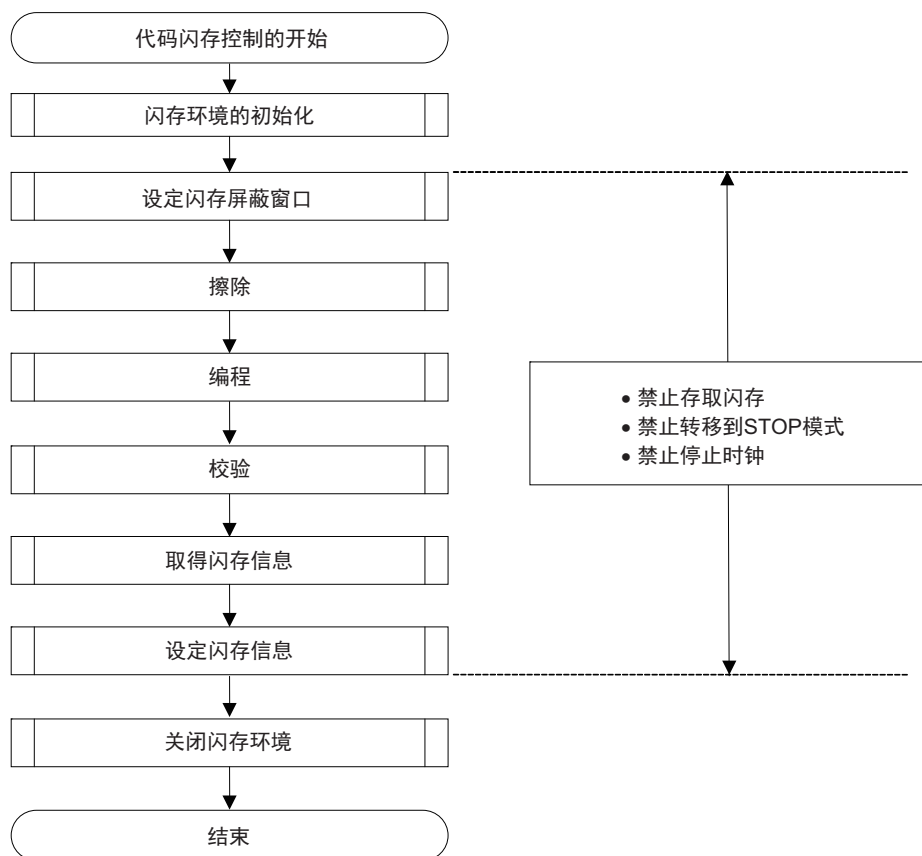
在执行本公司提供的闪存自编程库的“FSL_Init”函数时，如果参数“fsl_flash_voltage_u08”为“00H”，就设定为全速模式。否则，就设定为宽电压模式。

备注 在同时使用宽电压模式和全速模式的情况下进行编程、擦除或者校验时，没有任何限制事项。

23.6.1 自编程的步骤

利用闪存自编程库改写闪存的流程如下所示。

图 23-8 自编程流程（改写闪存）



23.6.2 引导交换功能

在通过自编程改写引导区的过程中，如果因瞬间断电等导致改写失败，引导区的数据就会遭到破坏，以致复位时无法重新启动程序并且无法重新编程。

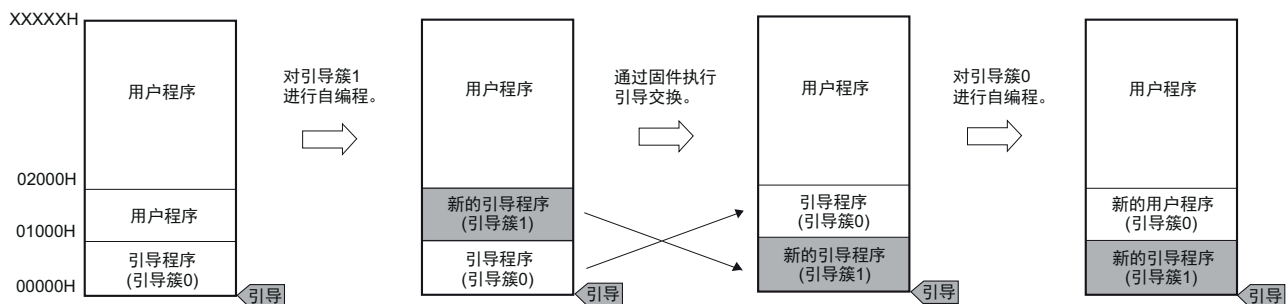
为了避免此问题，提供了引导交换功能。

在擦除引导区的引导簇 0 注前，能事先通过自编程对引导簇 1 进行新的引导程序的编程。如果对引导簇 1 的编程正常结束，就能通过 RL78 微控制器内部固件的设定信息功能进行引导簇 1 和引导簇 0 的交换，将引导簇 1 设定为引导区。此后，对原区域的引导簇 0 进行擦除和编程。

从而，即使在改写区域的过程中发生瞬间断电，也能在下次复位启动时从交换对象的引导簇 1 进行引导，正常启动程序。

注 引导簇是 4K 字节的区域，通过引导交换功能进行引导簇 0 和引导簇 1 的置换。

图 23-9 引导交换功能

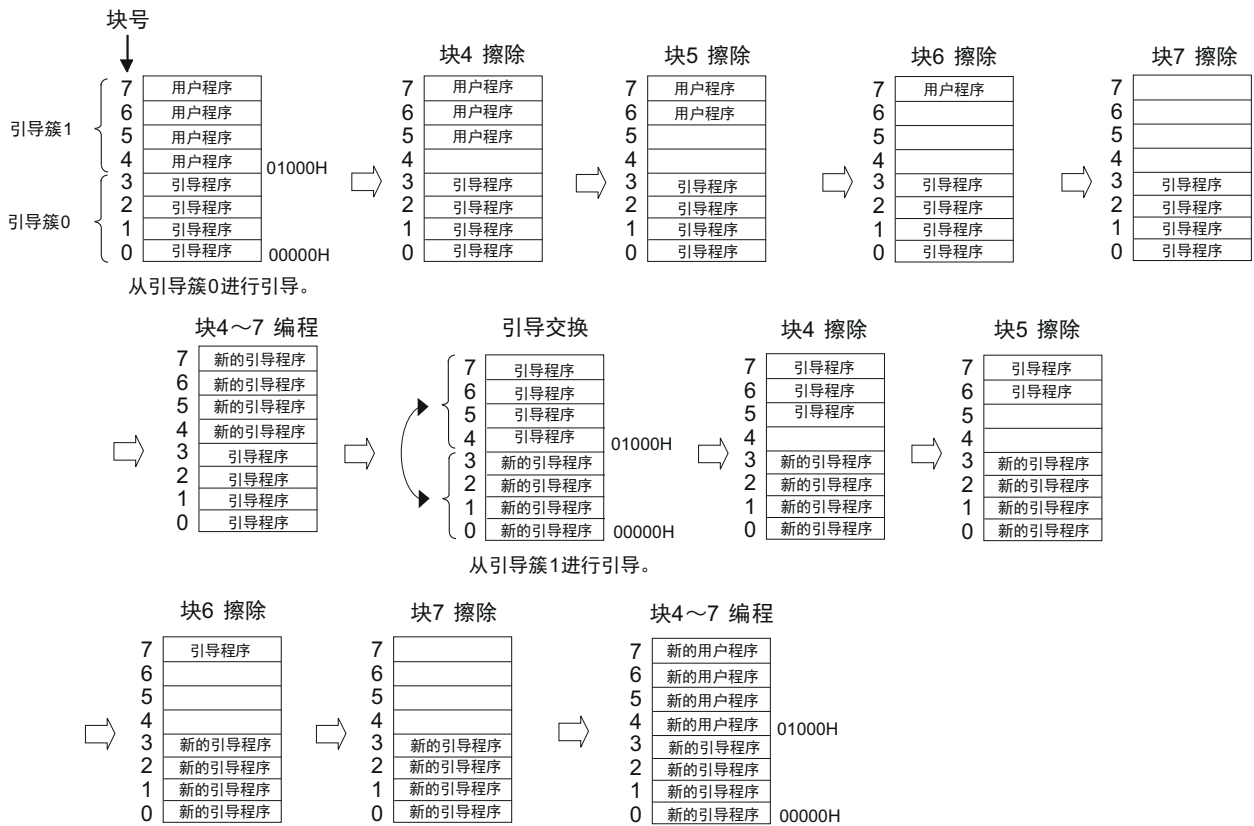


在此图的例子中，

引导簇 0：引导交换前的引导区

引导簇 1：引导交换后的引导区

图 23-10 引导交换的执行例子



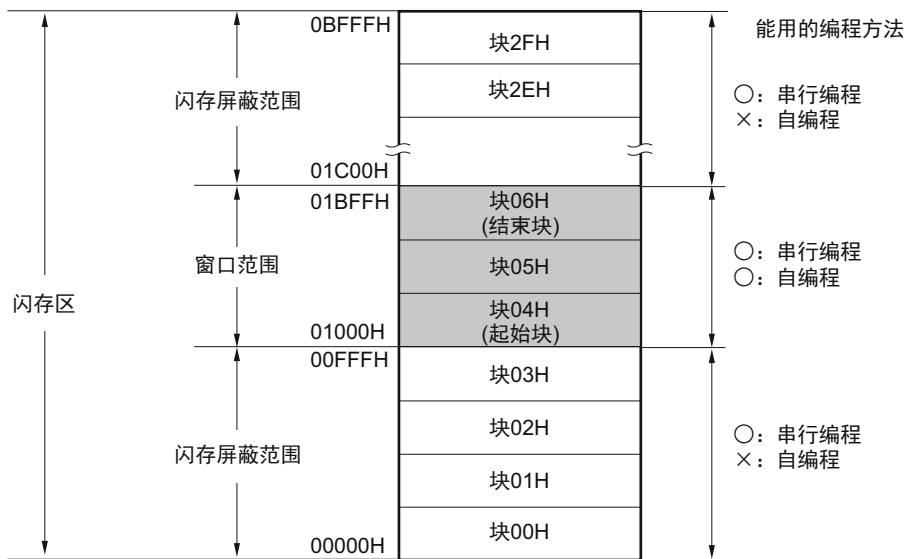
23.6.3 闪存屏蔽窗口功能

闪存屏蔽窗口功能作为自编程时的一个安全功能，只在自编程时禁止对指定的窗口范围以外的区域进行编程和擦除。

能通过指定起始块和结束块来设定窗口范围。能在串行编程和自编程时设定或者更改窗口范围的指定。

在自编程时，禁止对窗口范围以外的区域进行编程和擦除。但是，在串行编程时，也能对指定窗口范围以外的区域进行编程和擦除。

图 23-11 闪存屏蔽窗口的设定例子
(对象设备: R7F0C908, 起始块: 04H, 结束块: 06H)



注意 1. 如果引导簇 0 的禁止改写区和闪存屏蔽窗口范围重叠，就优先禁止改写引导簇 0。

2. 只能对代码闪存设定闪存屏蔽窗口（不支持数据闪存）。

表 23-11 闪存屏蔽窗口功能的设定 / 更改方法和命令的关系

编程条件	窗口范围的设定 / 更改方法	执行的命令	
		块擦除	编程
自编程	通过闪存自编程库指定窗口的起始块和结束块。	只能擦除窗口范围内的块。	只能对窗口范围以内的区域进行编程。
串行编程	通过专用闪存编程器的 GUI 等指定窗口的起始块和结束块。	也能擦除窗口范围以外的块。	也能对窗口范围以外的区域进行编程。

备注 要在串行编程过程中禁止编程和擦除时，请参照“23.7 安全设定”。

23.7 安全设定

RL78 微控制器支持安全功能，此安全功能禁止改写代码闪存中的用户程序，能防止他人更改程序。能通过使用 Security Set 命令进行以下的操作。

- 禁止块擦除
在串行编程时，禁止执行代码闪存中的块擦除命令。但是，在自编程时能进行块擦除。
- 禁止编程
在串行编程时，禁止对代码闪存中的全部块执行编程命令。但是，在自编程时能进行编程。在设定禁止编程后，Security Release 命令的解除在复位后有效。
- 禁止改写引导簇 0
禁止对代码闪存中的引导簇 0（00000H～00FFFH）执行块擦除命令和编程命令。

出厂时的初始状态为允许块擦除、编程以及改写引导簇 0。能在串行编程和自编程时进行安全功能的设定，并且能同时组合使用各种安全功能的设定。

RL78 微控制器的安全功能有效和擦除、编程命令的关系如表 23-12 所示。

注意 闪存编程器的安全功能不支持自编程。

备注 要在自编程过程中禁止编程和擦除时，使用闪存屏蔽窗口功能（详细内容请参照 23.6.3）。

表 23-12 安全功能有效和命令的关系

(1) 串行编程

有效的安全	执行的命令	
	块擦除	编程
禁止块擦除。	不能进行块擦除。	能进行编程注。
禁止编程。	能进行块擦除。	不能进行编程。
禁止改写引导簇 0。	不能擦除引导簇 0。	不能进行引导簇 0 的编程。

注 必须确认编程区中还没有进行数据编程。因为在设定禁止块擦除后无法进行擦除，所以在数据没有被擦除时，不能进行数据编程。

(2) 自编程

有效的安全	执行的命令	
	块擦除	编程
禁止块擦除。	能进行块擦除。	能进行编程。
禁止编程。		
禁止改写引导簇 0。	不能擦除引导簇 0。	不能进行引导簇 0 的编程。

备注 要在自编程过程中禁止编程和擦除时，使用闪存屏蔽窗口功能（详细内容请参照 23.6.3）。

表 23-13 各编程模式中的安全设定方法

(1) 串行编程

安全	安全设定方法	如何使安全设定无效
禁止块擦除。	通过专用闪存编程器的GUI等进行设定。	在设定后无法置为无效。
禁止编程。		通过专用闪存编程器的GUI等进行设定。
禁止改写引导簇 0。		在设定后无法置为无效。

注意 只有在没有设定为“禁止块擦除”或者“禁止改写引导簇 0”并且代码闪存区和数据闪存区为空白时才能解除“禁止编程”的设定。

(2) 自编程

安全	安全设定方法	如何使安全设定无效
禁止块擦除。	通过闪存自编程库进行设定。	在设定后无法置为无效。
禁止编程。		在自编程时无法置为无效（在串行编程时通过专用闪存编程器的 GUI 等进行设定）。
禁止改写引导簇 0。		在设定后无法置为无效。

23.8 数据闪存

23.8.1 数据闪存的概要

数据闪存的概要如下：

- 能使用数据闪存库，通过用户程序改写数据闪存。详细内容请参照《RL78 Family Data Flash Library User's Manual》。
- 能通过专用闪存编程器或者外部器件的串行编程改写数据闪存。
- 以 1K 字节为 1 块对数据闪存进行块擦除。
- 只能以 8 位为单位存取数据闪存。
- 能通过 CPU 指令直接读数据闪存。
- 在改写数据闪存的过程中，能从代码闪存执行指令（支持后台操作（BGO））。
- 数据闪存是数据专用区域，因此禁止从数据闪存执行指令。
- 在改写代码闪存的过程中（自编程的情况），禁止存取数据闪存。
- 在改写数据闪存的过程中，禁止操作 DFLCTL 寄存器。
- 在改写数据闪存的过程中，禁止转移到 STOP 模式。

注意 1. 在解除复位后，数据闪存处于停止状态。在使用数据闪存时，必须设定数据闪存控制寄存器（DFLCTL）。

2. 在改写数据闪存的过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30μs 后执行数据闪存库。

备注 有关通过用户程序改写代码闪存的详细内容，请参照“23.6 自编程”。

23.8.2 控制数据闪存的寄存器

23.8.2.1 数据闪存控制寄存器（DFLCTL）

此寄存器设定允许或者禁止存取数据闪存。

通过 1 位或者 8 位存储器操作指令设定 DFLCTL 寄存器。

在产生复位信号后，此寄存器的值变为“00H”。

图 23-12 数据闪存控制寄存器的格式（DFLCTL）

地址：F0090H	复位后：00H	R/W							
符号	7	6	5	4	3	2	1	0	
DFLCTL	0	0	0	0	0	0	0	0	DFLEN

DFLEN	数据闪存的存取控制
0	禁止存取数据闪存。
1	允许存取数据闪存。

注意 在改写数据闪存的过程中，禁止操作 DFLCTL 寄存器。

23.8.3 数据闪存的存取步骤

在解除复位后，数据闪存处于停止状态。要存取数据闪存时，必须按照以下步骤进行初始设定。

- ① 将数据闪存控制寄存器（DFCTL）的 bit0（DFLEN）置“1”。
- ② 通过软件定时器等来等待准备时间。
准备时间因主时钟的各闪存运行模式而不同。
<各闪存运行模式的准备时间>
 - HS（高速主）模式： 5 μ s
 - LS（低速主）模式： 720ns
 - LV（低电压主）模式： 10 μ s
- ③ 在等待准备时间后，能存取数据闪存。

注意 1. 在准备时间内，禁止存取数据闪存。

2. 禁止在准备时间内转移到 STOP 模式。要在准备时间内转移到 STOP 模式时，必须在将 DFLEN 位置“0”后执行 STOP 指令。
3. 在改写数据闪存的过程中，需要使高速内部振荡器振荡。当高速内部振荡器处于停止状态时，必须使高速内部振荡器时钟运行（HIOSTOP=0），并且在经过 30 μ s 后执行数据闪存库。

在初始设定后，能通过 CPU 指令读数据闪存，或者通过数据闪存库读或者改写数据闪存。

但是，如果在存取数据闪存时 DMA 控制器运行，就必须按照以下的任意一个步骤进行处理：

(A) 保留或者强制结束 DMA 传送。

在读数据闪存前，必须保留正在使用的全部通道的 DMA 传送。但是，必须在将 DWAITn 位置“1”后至少间隔 3 个时钟（ f_{CLK} ），然后读数据闪存。在读数据闪存后，必须将 DWAITn 位置“0”，解除传送的保留。

或者，在读数据闪存前，必须按照“14.5.5 通过软件强制结束”的步骤强制结束 DMA 传送，在读数据闪存后重新开始 DMA 传送。

(B) 使用库进行数据闪存的存取。

必须使用最新的数据闪存库进行数据闪存的存取。

(C) 插入 NOP。

必须在数据闪存的读指令之前插入 NOP 指令。

<例>

```
MOVW    HL, !addr16    ;读RAM。
NOP                                           ;在数据闪存的读指令之前插入NOP指令。
MOV     A, [DE]        ;读数据闪存。
```

但是，当使用 C 语言等高级语言时，编译程序可能对 1 个代码生成 2 条指令。此时，因为不在数据闪存的读指令之前插入 NOP 指令，所以必须通过 (A) 或者 (B) 读数据闪存。

备注 1. n: DMA 通道号 (n=0、1)

2. f_{CLK} : CPU/ 外围硬件的时钟频率

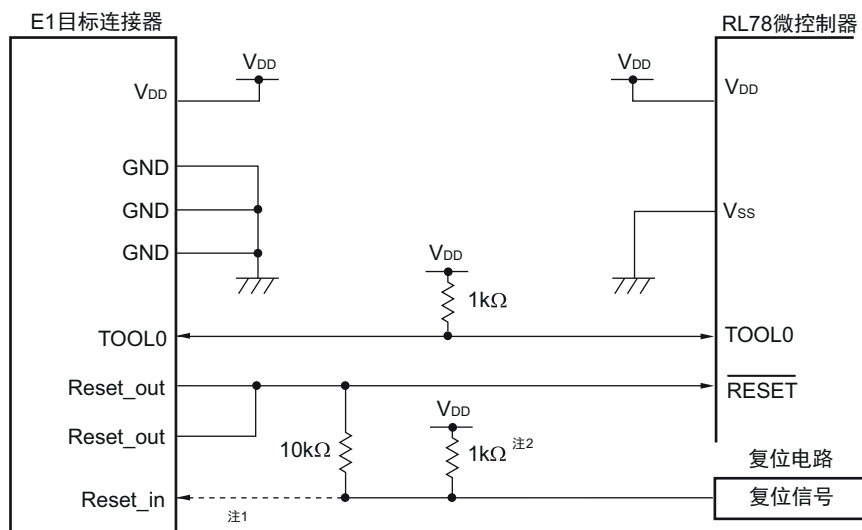
第 24 章 片上调试功能

24.1 和 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} 、 $\overline{\text{RESET}}$ 、 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 微控制器内置用于开发和评估的片上调试功能。如果使用片上调试功能，就可能会超过闪存的保证改写次数而无法保证产品的可靠性，因此批量生产的产品不能使用片上调试功能。对于使用片上调试功能的产品，不作为投诉受理对象。

图 24-1 和 E1 片上调试仿真器的连接例子



- 注 1. 在串行编程时，虚线部分不需要连接。
 2. 如果目标系统的复位电路没有缓冲器而只通过电阻和电容器生成复位信号，就不需要此上拉电阻。

注意 这是假设复位信号的输出为 N 沟道漏极开路缓冲器（输出电阻不大于 100Ω ）的电路例子。

24.2 片上调试安全 ID

为了防止他人读取存储器的内容，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 22 章 选项字节”），并且在 000C4H ~ 000CDH 中提供了片上调试安全 ID 设定区。

在自编程时使用引导交换功能的情况下，因为 000C3H 和 000C4H ~ 000CDH 分别被 010C3H 和 010C4H ~ 010CDH 替换，所以必须事先给 010C3H 和 010C4H ~ 010CDH 设定相同的值。

表 24-1 片上调试安全 ID

地址	片上调试安全 ID 码
000C4H ~ 000CDH	任意 10 字节的 ID 码注
010C4H ~ 010CDH	

注 不能设定“FFFFFFFFFFFFFFFFFFFFFH”。

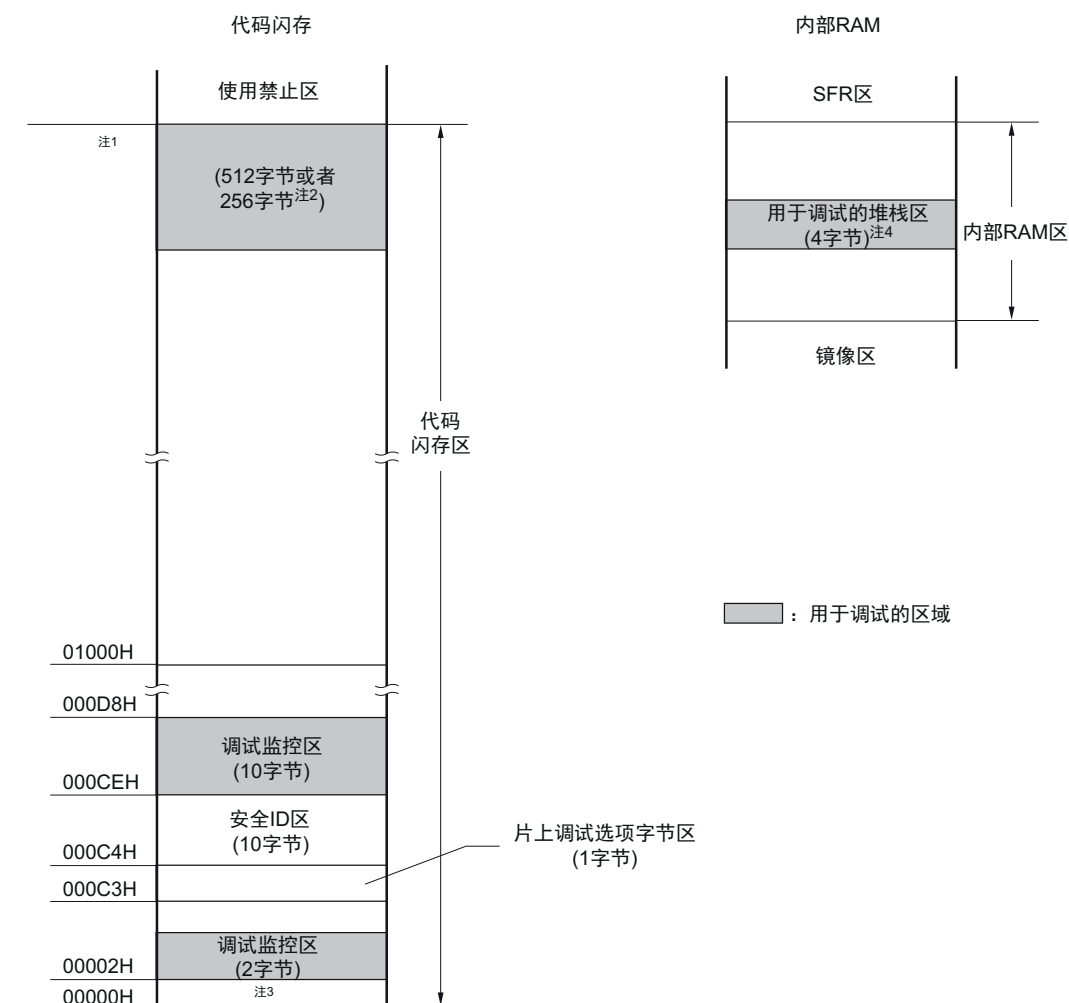
24.3 用户资源的确保

为了实现 RL78 微控制器与 E1 片上调试仿真器的通信或者实现各种调试功能，需要事先确存储空间。当使用本公司的汇编程序和编译程序时，也能通过链接选项进行设定。

(1) 存储空间的确

图 24-2 的灰色部分表示的区域保存用于调试的监控程序，是不能保存用户程序或者数据的空间。要使用片上调试功能时，需要确保不使用此空间的区域，而且不能在用户程序中改写此空间。

图 24-2 保存调试监控程序的存储空间



注 1. 如下所示，地址因产品而不同。

产品名（代码闪存容量）	注 1 的地址
R7F0C903、R7F0C906	03FFFH
R7F0C904、R7F0C907	07FFFH
R7F0C905、R7F0C908	0BFFFH

2. 在不使用实时 RAM 监控（RRM）功能和动态存储器修改（DMM）功能时，为 256 字节。
3. 在调试时，复位向量被改写为监控程序的分配地址。
4. 此区域分配在堆栈区之后，因此用于调试的堆栈区地址随着堆栈的增减而变。即，对于使用的堆栈区，额外占用 4 字节。在自编程时，额外占用 12 字节。

第 25 章 十进制校正 (BCD) 电路

25.1 十进制校正电路的功能

能以 BCD 码 (二 - 十进制) 计算 BCD 码 (二 - 十进制) 和 BCD 码 (二 - 十进制) 相加减的结果。

在执行以 A 寄存器为操作数的加减运算指令后, 通过加减 BCD 校正结果寄存器 (BCDADJ) 的值, 计算十进制的校正运算结果。

25.2 十进制校正电路使用的寄存器

十进制校正电路使用以下寄存器:

- BCD 校正结果寄存器 (BCDADJ)

25.2.1 BCD 校正结果寄存器 (BCDADJ)

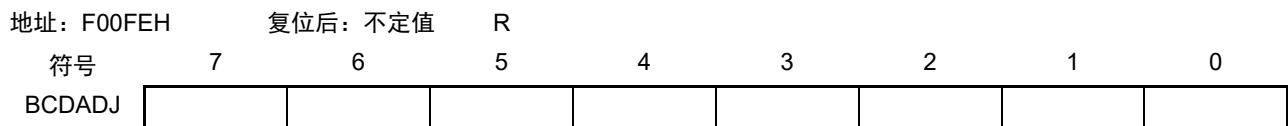
通过以 A 寄存器为操作数的加减运算指令, 将用于计算 BCD 码加减运算结果的校正值保存到 BCDADJ 寄存器。

BCDADJ 寄存器的读取值因读时的 A 寄存器、CY 标志和 AC 标志的值而变。

通过 8 位存储器操作指令读 BCDADJ 寄存器。

在产生复位信号后, 此寄存器的值变为不定值。

图 25-1 BCD 校正结果寄存器 (BCDADJ) 的格式



25.3 十进制校正电路的运行

十进制校正电路的基本运行如下所示。

(1) 加法 以 BCD 码值计算 BCD 码值加上 BCD 码值的结果

- ① 将要相加的BCD码值（被加数）保存到A寄存器。
- ② 以二进制将A寄存器的值和第2个操作数的值（另一个要相加的BCD码值，加数）相加，二进制的运算结果保存在A寄存器，校正值保存在BCD校正结果寄存器（BCDADJ）。
- ③ 以二进制将A寄存器的值（二进制的加法运算结果）和BCDADJ寄存器的值（校正值）相加进行十进制校正运算，校正结果保存在A寄存器和CY标志。

注意 BCDADJ寄存器的读取值因读时的A寄存器、CY标志和AC标志的值而变。因此，必须在②的指令后不执行其他指令而执行③的指令。要在允许中断的状态下进行BCD校正时，需要在中断函数中进行A寄存器的压栈和退栈。通过RETI指令恢复PSW（CY标志、AC标志）。

例子如下所示。

例 1 99+89=188

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #99H ; ①	99H	—	—	—
ADD A, #89H ; ②	22H	1	1	66H
ADD A, !BCDADJ ; ③	88H	1	0	—

例 2 85+15=100

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #85H ; ①	85H	—	—	—
ADD A, #15H ; ②	9AH	0	0	66H
ADD A, !BCDADJ ; ③	00H	1	1	—

例 3 80+80=160

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #80H ; ①	80H	—	—	—
ADD A, #80H ; ②	00H	1	0	60H
ADD A, !BCDADJ ; ③	60H	1	0	—

(2) 减法 以 BCD 码值计算 BCD 码值减去 BCD 码值的结果

- ① 将要减去的BCD码值（被减数）保存到A寄存器。
- ② 以二进制将A寄存器的值减去第2个操作数的值（要减去的BCD码值，减数），二进制的运算结果保存在A寄存器，校正值保存在BCD校正结果寄存器（BCDADJ）。
- ③ 以二进制将A寄存器的值（二进制的减法运算结果）减去BCDADJ寄存器的值（校正值）进行十进制校正运算，校正结果保存在A寄存器和CY标志。

注意 BCDADJ寄存器的读取值因读时的A寄存器、CY标志和AC标志的值而变。因此，必须在②的指令后不执行其他指令而执行③的指令。要在允许中断的状态下进行BCD校正时，需要在中断函数中进行A寄存器的压栈和退栈。通过RETI指令恢复PSW（CY标志、AC标志）。

例子如下所示。

例 91-52=39

指令		A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV	A, #91H ; ①	91H	—	—	—
SUB	A, #52H ; ②	3FH	0	1	06H
SUB	A, !BCDADJ ; ③	39H	0	0	—

第 26 章 指令集的概要

RL78 微控制器的指令集表示在一览表中。有关各指令的详细动作和机器码（指令码），请参照《RL78 Family User's Manual Software》（R01US0015E）。

26.1 凡例

26.1.1 操作数的表现形式和记述方法

根据指令操作数的表现形式所对应的记述方法，在各指令的操作数栏中记述了操作数（详细内容请参照汇编程序规格）。在记述方法中存在多个内容时，选择其中一个。大写字母以及 #、!、!!、\$、\$!、[] 和 ES: 的符号为关键字，应按照原样记述。符号的说明如下：

- #: 指定立即数。
- !: 指定 16 位绝对地址。
- !!: 指定 20 位绝对地址。
- \$: 指定 8 位相对地址。
- \$!: 指定 16 位相对地址。
- []: 指定间接地址。
- ES:: 指定扩展地址。

对于立即数，记述适当的数值或者标号。在使用标号时，必须记述 #、!、!!、\$、\$!、[] 和 ES: 的符号。

对于操作数的寄存器记述形式 r 和 rp，能使用功能名（X、A、C 等）或者绝对名（表 26-1 中括号内的名称，如 R0、R1、R2 等）进行记述。

表 26-1 操作数的表现形式和记述方法

表现形式	记述方法
r	X(R0)、A(R1)、C(R2)、B(R3)、E(R4)、D(R5)、L(R6)、H(R7)
rp	AX(RP0)、BC(RP1)、DE(RP2)、HL(RP3)
sfr	特殊功能寄存器的符号（SFR 符号）FFF00H ~ FFFFFH
sfrp	特殊功能寄存器的符号（可进行 16 位操作的 SFR 符号，只限于偶数地址注）FFF00H ~ FFFFFH
saddr	FFE20H ~ FFF1FH 立即数或者标号
saddrp	FFE20H ~ FFF1FH 立即数或者标号（只限于偶数地址注）
addr20	00000H ~ FFFFFH 立即数或者标号
addr16	0000H ~ FFFFH 立即数或者标号（在 16 位数据时，只限于偶数地址注）
addr5	0080H ~ 00BFH 立即数或者标号（只限于偶数地址注）
word	16 位立即数或者标号
byte	8 位立即数或者标号
bit	3 位立即数或者标号
RBn	RB0 ~ RB3

注 当指定奇数地址时，bit0 为“0”。

备注 能用符号将特殊功能寄存器记述为操作数 sfr。有关特殊功能寄存器的符号，请参照“表 3-5 SFR 一览表”。
能用符号将扩展特殊功能寄存器记述为操作数!addr16。有关扩展特殊功能寄存器的符号，请参照“表 3-6 扩展 SFR (2nd SFR) 一览表”。

26.1.2 操作栏的说明

在各指令的操作栏中，用以下符号表示指令执行时的动作。

表 26-2 操作栏的符号

符号	功能
A	A 寄存器：8 位累加器
X	X 寄存器
B	B 寄存器
C	C 寄存器
D	D 寄存器
E	E 寄存器
H	H 寄存器
L	L 寄存器
ES	ES 寄存器
CS	CS 寄存器
AX	AX 寄存器对：16 位累加器
BC	BC 寄存器对
DE	DE 寄存器对
HL	HL 寄存器对
PC	程序计数器
SP	堆栈指针
PSW	程序状态字
CY	进位标志
AC	辅助进位标志
Z	零标志
RBS	寄存器组选择标志
IE	中断请求允许标志
()	() 内的地址或者寄存器的内容所示的存储器内容
X_H 、 X_L	16 位寄存器： X_H = 高 8 位、 X_L = 低 8 位
X_S 、 X_H 、 X_L	20 位寄存器： X_S (bit19 ~ 16)、 X_H (bit15 ~ 8)、 X_L (bit7 ~ 0)
\wedge	逻辑与 (AND)
\vee	逻辑或者 (OR)
∇	异或 (exclusive OR)
—	数据取反
addr5	16 位立即数 (只限于偶数地址 0080H ~ 00BFH)
addr16	16 位立即数
addr20	20 位立即数
jdisp8	带符号的 8 位数据 (位移量)
jdisp16	带符号的 16 位数据 (位移量)

26.1.3 标志栏的说明

在各指令的标志栏中，用以下符号表示指令执行时的标志变化。

表 26-3 标志栏的符号

符号	标志的变化
(空白)	无变化。
0	清“0”。
1	置“1”。
×	根据结果置位或者复位。
R	恢复以前保存的值。

26.1.4 PREFIX 指令

ES: 所示的指令以 PREFIX 指令码为前缀，将能存取的数据区从 F0000H ~ FFFFFH 的 64K 字节空间扩展为附加 ES 寄存器值的 00000H ~ FFFFFH 的 1M 字节空间。将 PREFIX 指令码置于对象指令的前头，只有紧接在 PREFIX 指令码后的 1 条指令作为附加了 ES 寄存器值的地址进行执行。

在 PREFIX 指令码和紧随其后的 1 条指令之间，不接受中断和 DMA 传送。

表 26-4 PREFIX 指令码的使用例子

指令	指令码				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES:[HL]	11H	8BH	—	—	—

注意 在执行 PREFIX 指令前，必须事先通过 MOV ES, A 等设定 ES 寄存器的值。

26.2 操作一览表

表 26-5 操作一览表 (1/17)

指令群	助记符	操作数	字节	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据传送	MOV	r, #byte	2	1	—	r←byte			
		PSW, #byte	3	3	—	PSW←byte	×	×	×
		CS, #byte	3	1	—	CS←byte			
		ES, #byte	2	1	—	ES←byte			
		!addr16, #byte	4	1	—	(addr16)←byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16)←byte			
		saddr, #byte	3	1	—	(saddr)←byte			
		sfr, #byte	3	1	—	sfr←byte			
		[DE+byte], #byte	3	1	—	(DE+byte)←byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE)+byte)←byte			
		[HL+byte], #byte	3	1	—	(HL+byte)←byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL)+byte)←byte			
		[SP+byte], #byte	3	1	—	(SP+byte)←byte			
		word[B], #byte	4	1	—	(B+word)←byte			
		ES:word[B], #byte	5	2	—	((ES, B)+word)←byte			
		word[C], #byte	4	1	—	(C+word)←byte			
		ES:word[C], #byte	5	2	—	((ES, C)+word)←byte			
		word[BC], #byte	4	1	—	(BC+word)←byte			
		ES:word[BC], #byte	5	2	—	((ES, BC)+word)←byte			
		A, r 注3	1	1	—	A←r			
		r, A 注3	1	1	—	r←A			
		A, PSW	2	1	—	A←PSW			
		PSW, A	2	3	—	PSW←A	×	×	×
		A, CS	2	1	—	A←CS			
		CS, A	2	1	—	CS←A			
		A, ES	2	1	—	A←ES			
		ES, A	2	1	—	ES←A			
		A, !addr16	3	1	4	A←(addr16)			
		A, ES:!addr16	4	2	5	A←(ES, addr16)			
		!addr16, A	3	1	—	(addr16)←A			
ES:!addr16, A	4	2	—	(ES, addr16)←A					
A, saddr	2	1	—	A←(saddr)					
saddr, A	2	1	—	(saddr)←A					

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	—	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	—	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	—	$(\text{DE}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (\text{ES}, \text{DE})$			
		ES:[DE], A	2	2	—	$(\text{ES}, \text{DE}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	—	$(\text{HL}) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (\text{ES}, \text{HL})$			
		ES:[HL], A	2	2	—	$(\text{ES}, \text{HL}) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE+byte], A	2	1	—	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{DE}) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((\text{ES}, \text{DE}) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow (\text{HL} + \text{byte})$			
		[HL+byte], A	2	1	—	$(\text{HL} + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{HL}) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((\text{ES}, \text{HL}) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow (\text{SP} + \text{byte})$			
		[SP+byte], A	2	1	—	$(\text{SP} + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (\text{B} + \text{word})$			
		word[B], A	3	1	—	$(\text{B} + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((\text{ES}, \text{B}) + \text{word})$			
		ES:word[B], A	4	2	—	$((\text{ES}, \text{B}) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (\text{C} + \text{word})$			
		word[C], A	3	1	—	$(\text{C} + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((\text{ES}, \text{C}) + \text{word})$			
		ES:word[C], A	4	2	—	$((\text{ES}, \text{C}) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (\text{BC} + \text{word})$			
		word[BC], A	3	1	—	$(\text{BC} + \text{word}) \leftarrow A$			
A, ES:word[BC]	4	2	5	$A \leftarrow ((\text{ES}, \text{BC}) + \text{word})$					
ES:word[BC], A	4	2	—	$((\text{ES}, \text{BC}) + \text{word}) \leftarrow A$					

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	1	—	rp \leftarrow word			
			saddrp, #word	4	1	—	(saddrp) \leftarrow word			
			sfrp, #word	4	1	—	sfrp \leftarrow word			
AX, rp 注3			1	1	—	AX \leftarrow rp				
rp, AX 注3			1	1	—	rp \leftarrow AX				
AX, laddr16			3	1	4	AX \leftarrow (addr16)				
laddr16, AX			3	1	—	(addr16) \leftarrow AX				
AX, ES:laddr16			4	2	5	AX \leftarrow (ES, addr16)				
ES:laddr16, AX			4	2	—	(ES, addr16) \leftarrow AX				
AX, saddrp			2	1	—	AX \leftarrow (saddrp)				
saddrp, AX			2	1	—	(saddrp) \leftarrow AX				
AX, sfrp			2	1	—	AX \leftarrow sfrp				
sfrp, AX	2	1	—	sfrp \leftarrow AX						

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	1	4	AX←(DE)			
		[DE], AX	1	1	—	(DE)←AX			
		AX, ES:[DE]	2	2	5	AX←(ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE)←AX			
		AX, [HL]	1	1	4	AX←(HL)			
		[HL], AX	1	1	—	(HL)←AX			
		AX, ES:[HL]	2	2	5	AX←(ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL)←AX			
		AX, [DE+byte]	2	1	4	AX←(DE+byte)			
		[DE+byte], AX	2	1	—	(DE+byte)←AX			
		AX, ES:[DE+byte]	3	2	5	AX←((ES, DE)+byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE)+byte)←AX			
		AX, [HL+byte]	2	1	4	AX←(HL+byte)			
		[HL+byte], AX	2	1	—	(HL+byte)←AX			
		AX, ES:[HL+byte]	3	2	5	AX←((ES, HL)+byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL)+byte)←AX			
		AX, [SP+byte]	2	1	—	AX←(SP+byte)			
		[SP+byte], AX	2	1	—	(SP+byte)←AX			
		AX, word[B]	3	1	4	AX←(B+word)			
		word[B], AX	3	1	—	(B+word)←AX			
		AX, ES:word[B]	4	2	5	AX←((ES, B)+word)			
		ES:word[B], AX	4	2	—	((ES, B)+word)←AX			
		AX, word[C]	3	1	4	AX←(C+word)			
		word[C], AX	3	1	—	(C+word)←AX			
		AX, ES:word[C]	4	2	5	AX←((ES, C)+word)			
		ES:word[C], AX	4	2	—	((ES, C)+word)←AX			
		AX, word[BC]	3	1	4	AX←(BC+word)			
		word[BC], AX	3	1	—	(BC+word)←AX			
AX, ES:word[BC]	4	2	5	AX←((ES, BC)+word)					
ES:word[BC], AX	4	2	—	((ES, BC)+word)←AX					

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	1	4	BC←(addr16)			
		BC, ES:!addr16	4	2	5	BC←(ES, addr16)			
		DE, !addr16	3	1	4	DE←(addr16)			
		DE, ES:!addr16	4	2	5	DE←(ES, addr16)			
		HL, !addr16	3	1	4	HL←(addr16)			
		HL, ES:!addr16	4	2	5	HL←(ES, addr16)			
		BC, saddrp	2	1	—	BC←(saddrp)			
		DE, saddrp	2	1	—	DE←(saddrp)			
		HL, saddrp	2	1	—	HL←(saddrp)			
	XCHW	AX, rp 注 3	1	1	—	AX↔rp			
	ONEW	AX	1	1	—	AX←0001H			
		BC	1	1	—	BC←0001H			
	CLRW	AX	1	1	—	AX←0000H			
		BC	1	1	—	BC←0000H			
8 位 运 算	ADD	A, #byte	2	1	—	A, CY←A+byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY←(saddr)+byte	×	×	×
		A, r 注 4	2	1	—	A, CY←A+r	×	×	×
		r, A	2	1	—	r, CY←r+A	×	×	×
		A, !addr16	3	1	4	A, CY←A+(addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY←A+(ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY←A+(saddr)	×	×	×
		A, [HL]	1	1	4	A, CY←A+(HL)	×	×	×
		A, ES:[HL]	2	2	5	A, CY←A+(ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY←A+(HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A, CY←A+((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY←A+(HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A, CY←A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY←A+(HL+C)	×	×	×
A, ES:[HL+C]	3	2	5	A, CY←A+((ES, HL)+C)	×	×	×		

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	1	—	AX, CY←AX+word	×	×	×
		AX, AX	1	1	—	AX, CY←AX+AX	×	×	×
		AX, BC	1	1	—	AX, CY←AX+BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX+DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY←AX-word	×	×	×
		AX, BC	1	1	—	AX, CY←AX-BC	×	×	×
		AX, DE	1	1	—	AX, CY←AX-DE	×	×	×
		AX, HL	1	1	—	AX, CY←AX-HL	×	×	×
		AX, !addr16	3	1	4	AX, CY←AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY←AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY←AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY←AX-(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY←AX-((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX-word	×	×	×
		AX, BC	1	1	—	AX-BC	×	×	×
		AX, DE	1	1	—	AX-DE	×	×	×
		AX, HL	1	1	—	AX-HL	×	×	×
		AX, !addr16	3	1	4	AX-(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX-(ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX-(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX-(HL+byte)	×	×	×
AX, ES: [HL+byte]		4	2	5	AX-((ES:HL)+byte)	×	×	×	
乘	MULU	X	1	1	—	AX←A×X			

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	—	$r \leftarrow r+1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)+1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r-1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)-1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$	×	×	
	INCW	rp	1	1	—	$rp \leftarrow rp+1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)+1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	1	—	$rp \leftarrow rp-1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16)-1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp)-1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$			
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$			
移位	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×	

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

2. cnt 是位的移位个数。

表 26-5 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	1	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC, 1	2	1	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, A.bit	2	1	—	$CY \leftarrow A.bit$			×
		A.bit, CY	2	1	—	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	—	$CY \leftarrow PSW.bit$			×
		PSW.bit, CY	3	4	—	$PSW.bit \leftarrow CY$	×	×	
		CY, saddr.bit	3	1	—	$CY \leftarrow (saddr).bit$			×
		saddr.bit, CY	3	2	—	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	—	$CY \leftarrow sfr.bit$			×
		sfr.bit, CY	3	2	—	$sfr.bit \leftarrow CY$			
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		[HL].bit, CY	2	2	—	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			×
	ES:[HL].bit, CY	3	3	—	$(ES, HL).bit \leftarrow CY$				
	AND1	CY, A.bit	2	1	—	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \wedge sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			×
	OR1	CY, A.bit	2	1	—	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \vee PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \vee sfr.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			×

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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、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以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	3	—	(SP-2) \leftarrow (PC+2) _S , (SP-3) \leftarrow (PC+2) _H , (SP-4) \leftarrow (PC+2) _L , PC \leftarrow CS, rp, SP \leftarrow SP-4			
		\$!addr20	3	3	—	(SP-2) \leftarrow (PC+3) _S , (SP-3) \leftarrow (PC+3) _H , (SP-4) \leftarrow (PC+3) _L , PC \leftarrow PC+3+jdisp16, SP \leftarrow SP-4			
		!addr16	3	3	—	(SP-2) \leftarrow (PC+3) _S , (SP-3) \leftarrow (PC+3) _H , (SP-4) \leftarrow (PC+3) _L , PC \leftarrow 0000, addr16, SP \leftarrow SP-4			
		!!addr20	4	3	—	(SP-2) \leftarrow (PC+4) _S , (SP-3) \leftarrow (PC+4) _H , (SP-4) \leftarrow (PC+4) _L , PC \leftarrow addr20, SP \leftarrow SP-4			
	CALLT	[addr5]	2	5	—	(SP-2) \leftarrow (PC+2) _S , (SP-3) \leftarrow (PC+2) _H , (SP-4) \leftarrow (PC+2) _L , PC _S \leftarrow 0000, PC _H \leftarrow (0000, addr5+1), PC _L \leftarrow (0000, addr5), SP \leftarrow SP-4			
	BRK	—	2	5	—	(SP-1) \leftarrow PSW, (SP-2) \leftarrow (PC+2) _S , (SP-3) \leftarrow (PC+2) _H , (SP-4) \leftarrow (PC+2) _L , PC _S \leftarrow 0000, PC _H \leftarrow (0007FH), PC _L \leftarrow (0007EH), SP \leftarrow SP-4, IE \leftarrow 0			
	RET	—	1	6	—	PC _L \leftarrow (SP), PC _H \leftarrow (SP+1), PC _S \leftarrow (SP+2), SP \leftarrow SP+4			
	RETI	—	2	6	—	PC _L \leftarrow (SP), PC _H \leftarrow (SP+1), PC _S \leftarrow (SP+2), PSW \leftarrow (SP+3), SP \leftarrow SP+4	R	R	R
	RETB	—	2	6	—	PC _L \leftarrow (SP), PC _H \leftarrow (SP+1), PC _S \leftarrow (SP+2), PSW \leftarrow (SP+3), SP \leftarrow SP+4	R	R	R

注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。

2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	1	—	(SP-1)←PSW, (SP-2)←00H, SP←SP-2			
		rp	1	1	—	(SP-1)←rp _H , (SP-2)←rp _L , SP←SP-2			
	POP	PSW	2	3	—	PSW←(SP+1), SP←SP+2	R	R	R
		rp	1	1	—	rp _L ←(SP), rp _H ←(SP+1), SP←SP+2			
	MOVW	SP, #word	4	1	—	SP←word			
		SP, AX	2	1	—	SP←AX			
		AX, SP	2	1	—	AX←SP			
		HL, SP	3	1	—	HL←SP			
		BC, SP	3	1	—	BC←SP			
		DE, SP	3	1	—	DE←SP			
ADDW	SP, #byte	2	1	—	SP←SP+byte				
SUBW	SP, #byte	2	1	—	SP←SP-byte				
无条件转移	BR	AX	2	3	—	PC←CS, AX			
		\$addr20	2	3	—	PC←PC+2+jdisp8			
		\$!addr20	3	3	—	PC←PC+3+jdisp16			
		!addr16	3	3	—	PC←0000, addr16			
		!!addr20	4	3	—	PC←addr20			
条件转移	BC	\$addr20	2	2/4注3	—	PC←PC+2+jdisp8 if CY=1			
	BNC	\$addr20	2	2/4注3	—	PC←PC+2+jdisp8 if CY=0			
	BZ	\$addr20	2	2/4注3	—	PC←PC+2+jdisp8 if Z=1			
	BNZ	\$addr20	2	2/4注3	—	PC←PC+2+jdisp8 if Z=0			
	BH	\$addr20	3	2/4注3	—	PC←PC+3+jdisp8 if (Z∨CY)=0			
	BNH	\$addr20	3	2/4注3	—	PC←PC+3+jdisp8 if (Z∨CY)=1			
	BT	saddr.bit, \$addr20	4	3/5注3	—	PC←PC+4+jdisp8 if (saddr).bit=1			
		sfr.bit, \$addr20	4	3/5注3	—	PC←PC+4+jdisp8 if sfr.bit=1			
		A.bit, \$addr20	3	3/5注3	—	PC←PC+3+disp8 if A.bit=1			
		PSW.bit, \$addr20	4	3/5注3	—	PC←PC+4+jdisp8 if PSW.bit=1			
[HL].bit, \$addr20		3	3/5注3	6/7	PC←PC+3+jdisp8 if (HL).bit=1				
ES:[HL].bit, \$addr20	4	4/6注3	7/8	PC←PC+4+jdisp8 if (ES, HL).bit=1					

- 注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。
 2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 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 2 倍 +3 个时钟。

表 26-5 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	SEL注4	RBn	2	1	—	RBS[1:0]←n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE←1 (Enable Interrupt)			
	DI	—	3	4	—	IE←0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

- 注 1. 这是存取内部 RAM 区、SFR 区、扩展 SFR 区时或者不存取数据时的 CPU 时钟 (f_{CLK}) 数。
 2. 这是存取代码闪存区以及通过 8 位指令存取数据闪存区时的 CPU 时钟 (f_{CLK}) 数。
 3. 表示“条件不成立时 / 条件成立时”的时钟数。
 4. n 为寄存器组号 (n=0 ~ 3)。

备注 时钟数是在内部 ROM (闪存) 区有程序的情况。当从内部 RAM 区取指令时, 最大为 2 倍 +3 个时钟。

第 27 章 电特性 (T_A=-40 ~ +85°C)

- 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 各产品配置的功能”。

27.1 绝对最大额定值

绝对最大额定值 (T_A=25°C) (1/2)

项目	符号	条件	额定值	单位
电源电压	V _{DD}		-0.5 ~ +6.5	V
REGC 引脚的输入电压	V _I REGC	REGC	-0.3 ~ +2.8 并且 -0.3 ~ V _{DD} +0.3 注 1	V
输入电压	V _{I1}	P00、P01、P10 ~ P17、P30、P31、P40、P50、P51、P70、P120、P147	-0.3 ~ V _{DD} +0.3 注 2	V
	V _{I2}	P60 ~ P62 (N 沟道漏极开路)	-0.3 ~ +6.5	V
	V _{I3}	P20 ~ P23、P121、P122、P137、EXCLK、 $\overline{\text{RESET}}$	-0.3 ~ V _{DD} +0.3 注 2	V
输出电压	V _{O1}	P00、P01、P10 ~ P17、P30、P31、P40、P50、P51、P60 ~ P62、P70、P120、P147	-0.3 ~ V _{DD} +0.3 注 2	V
	V _{O2}	P20 ~ P23	-0.3 ~ V _{DD} +0.3 注 2	V
模拟输入电压	V _{AI1}	ANI16 ~ ANI19	-0.3 ~ V _{DD} +0.3 并且 -0.3 ~ AV _{REF(+)} +0.3 注 2、3	V
	V _{AI2}	ANI0 ~ ANI3	-0.3 ~ V _{DD} +0.3 并且 -0.3 ~ AV _{REF(+)} +0.3 注 2、3	V

- 注 1. 必须通过电容器 (0.47 ~ 1μF) 将 REGC 引脚连接 V_{SS}。此值是规定 REGC 引脚的绝对最大额定值，使用时不能外加电压。
2. 不超过 6.5V。
3. A/D 转换对象的引脚不能超过 AV_{REF(+)}+0.3。

注意 即使是各项目中的 1 个项目瞬间超过绝对最大额定值，也可能降低产品的质量。绝对最大额定值是不可能给产品带来物理性损伤的额定值，必须在不超过额定值的状态下使用产品。

- 备注 1. 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。
2. AV_{REF(+)}: A/D 转换器的正 (+) 基准电压
3. 将 V_{SS} 作为基准电压。

绝对最大额定值 ($T_A=25^{\circ}\text{C}$) (2/2)

项目	符号	条件		额定值	单位
高电平输出电流	I_{OH1}	1 个引脚	P00、P01、P10 ~ P17、P30、P31、P40、P50、P51、P62、P70、P120、P147	-40	mA
		引脚合计 -170mA	P00、P01、P40、P120	-70	mA
			P10 ~ P17、P30、P31、P50、P51、P70、P147	-100	mA
	I_{OH2}	1 个引脚	P20 ~ P23	-0.5	mA
		引脚合计		-2	mA
低电平输出电流	I_{OL1}	1 个引脚	P00、P01、P10 ~ P17、P30、P31、P40、P50、P51、P60 ~ P62、P70、P120、P147	40	mA
		引脚合计 170mA	P00、P01、P40、P120	70	mA
			P10 ~ P17、P30、P31、P50、P51、P60 ~ P62、P70、P147	100	mA
	I_{OL2}	1 个引脚	P20 ~ P23	1	mA
		引脚合计		5	mA
工作环境温度	T_A	通常运行时		-40 ~ +85	$^{\circ}\text{C}$
		闪存编程时			
保存温度	T_{stg}			-65 ~ +150	$^{\circ}\text{C}$

注意 即使是各项目中的 1 个项目瞬间超过绝对最大额定值，也可能降低产品的质量。绝对最大额定值是不可能给产品带来物理性损伤的额定值，必须在不超过额定值的状态下使用产品。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

27.2 振荡电路特性

27.2.1 X1 振荡电路特性

($T_A=-40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$)

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 / 晶体谐振器	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	1.0		20.0	MHz
		$2.4\text{V} \leq V_{DD} < 2.7\text{V}$	1.0		16.0	
		$1.8\text{V} \leq V_{DD} < 2.4\text{V}$	1.0		8.0	
		$1.6\text{V} \leq V_{DD} < 1.8\text{V}$	1.0		4.0	

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。
请委托谐振器厂商给予安装电路后的评估，并且在确认振荡特性后使用。

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”。

27.2.2 内部振荡器特性

($T_A=-40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$)

谐振器	符号	条件	MIN.	TYP.	MAX.	单位	
高速内部振荡器的时钟频率注 1、2	f_{IH}		1		24	MHz	
高速内部振荡器的时钟频率精度		$-20 \sim +85^\circ\text{C}$	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	-1.0		+1.0	%
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$	-5.0		+5.0	%
		$-40 \sim -20^\circ\text{C}$	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	-1.5		+1.5	%
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$	-5.5		+5.5	%
低速内部振荡器的时钟频率	f_{IL}			15		kHz	
低速内部振荡器的时钟频率精度			-15		+15	%	

注 1. 通过选项字节 (000C2H/010C2H) 的 bit0 ~ 3 和 HOCODIV 寄存器的 bit0 ~ 2 选择高速内部振荡器的频率。
2. 只表示振荡电路的特性，指令执行时间请参照 AC 特性。

27.3 DC 特性

27.3.1 引脚特性

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V) (1/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
高电平输出电流注 1	I _{OH1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P51、P70、P120、P147 1 个引脚	1.6V ≤ V _{DD} ≤ 5.5V			-10.0注2	mA
		P00、P01、P40、P120 合计 (占空比 ≤ 70% 时注 3)	4.0V ≤ V _{DD} ≤ 5.5V			-28.0	mA
			2.7V ≤ V _{DD} < 4.0V			-10.0	mA
			1.8V ≤ V _{DD} < 2.7V			-5.0	mA
			1.6V ≤ V _{DD} < 1.8V			-2.5	mA
		P10 ~ P17、P30、P31、 P50、P51、P70、P147 合计 (占空比 ≤ 70% 时注 3)	4.0V ≤ V _{DD} ≤ 5.5V			-80.0	mA
			2.7V ≤ V _{DD} < 4.0V			-19.0	mA
			1.8V ≤ V _{DD} < 2.7V			-10.0	mA
			1.6V ≤ V _{DD} < 1.8V			-5.0	mA
		全部引脚合计 (占空比 ≤ 70% 时注 3)	1.6V ≤ V _{DD} ≤ 5.5V			-108.0	mA
	I _{OH2}	P20 ~ P23 1 个引脚	1.6V ≤ V _{DD} ≤ 5.5V			-0.1注2	mA
		全部引脚合计 (占空比 ≤ 70% 时注 3)	1.6V ≤ V _{DD} ≤ 5.5V			-0.3	mA

注 1. 这是即使电流从 V_{DD} 引脚流到输出引脚也保证器件工作的电流值。

2. 不能超过合计的电流值。

3. 这是“占空比 ≤ 70% 条件”的输出电流值。

改为占空比 > 70% 的输出电流值能用以下的计算式进行计算 (将占空比改为 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

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。

注意 在 N 沟道漏极开路模式中，P00、P10 ~ P15、P17、P50 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。

($T_A=-40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$) (2/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位
低电平输出电流 ^{注1}	I_{OL1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P51、P70、P120、P147 1 个引脚			20.0 ^{注2}	mA
		P60 ~ P62 1 个引脚			15.0 ^{注2}	mA
		P00、P01、P40、P120 合计 (占空比 $\leq 70\%$ 时 ^{注3})	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$		56.0	mA
			$2.7\text{V} \leq V_{DD} < 4.0\text{V}$		15.0	mA
			$1.8\text{V} \leq V_{DD} < 2.7\text{V}$		9.0	mA
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$		4.5	mA
		P10 ~ P17、P30、P31、 P50、P51、P60 ~ P62、 P70、P147 合计 (占空比 $\leq 70\%$ 时 ^{注3})	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$		80.0	mA
			$2.7\text{V} \leq V_{DD} < 4.0\text{V}$		35.0	mA
			$1.8\text{V} \leq V_{DD} < 2.7\text{V}$		20.0	mA
			$1.6\text{V} \leq V_{DD} < 1.8\text{V}$		10.0	mA
	全部引脚合计 (占空比 $\leq 70\%$ 时 ^{注3})			136.0	mA	
	I_{OL2}	P20 ~ P23 1 个引脚			0.4 ^{注2}	mA
		全部引脚合计 (占空比 $\leq 70\%$ 时 ^{注3})	$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$		1.2	mA

- 注 1. 这是即使电流从输出引脚流到 V_{SS} 引脚也保证器件工作的电流值。
 2. 不能超过合计的电流值。
 3. 这是“占空比 $\leq 70\%$ 条件”的输出电流值。
 改为占空比 $> 70\%$ 的输出电流值能用以下的计算式进行计算（将占空比改为 $n\%$ 的情况）。
 • 引脚合计的输出电流 = $(I_{OL} \times 0.7) / (n \times 0.01)$
 <计算例子> $I_{OL} = 10.0\text{mA}$ 、 $n = 80\%$
 引脚合计的输出电流 = $(10.0 \times 0.7) / (80 \times 0.01) \approx 8.7\text{mA}$

各引脚的电流不会因占空比而变，而且不会流过绝对最大额定值以上的电流。

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$) (3/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
高电平输入电压	V_{IH1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P51、P70、P120、P147	通常的输入缓冲器	$0.8V_{DD}$		V_{DD}	V
	V_{IH2}	P01、P10、P11、 P13 ~ P17	TTL 输入缓冲器 $4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$	2.2		V_{DD}	V
			TTL 输入缓冲器 $3.3\text{V} \leq V_{DD} < 4.0\text{V}$	2.0		V_{DD}	V
			TTL 输入缓冲器 $1.6\text{V} \leq V_{DD} < 3.3\text{V}$	1.5		V_{DD}	V
	V_{IH3}	P20 ~ P23		$0.7V_{DD}$		V_{DD}	V
	V_{IH4}	P60 ~ P62		$0.7V_{DD}$		6.0	V
	V_{IH5}	P137、EXCLK、 $\overline{\text{RESET}}$		$0.8V_{DD}$		V_{DD}	V
低电平输入电压	V_{IL1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P57、P70、P120、P147	通常的输入缓冲器	0		$0.2V_{DD}$	V
	V_{IL2}	P01、P10、P11、 P13 ~ P17	TTL 输入缓冲器 $4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$	0		0.8	V
			TTL 输入缓冲器 $3.3\text{V} \leq V_{DD} < 4.0\text{V}$	0		0.5	V
			TTL 输入缓冲器 $1.6\text{V} \leq V_{DD} < 3.3\text{V}$	0		0.32	V
	V_{IL3}	P20 ~ P23		0		$0.3V_{DD}$	V
	V_{IL4}	P60 ~ P62		0		$0.3V_{DD}$	V
	V_{IL5}	P137、EXCLK、 $\overline{\text{RESET}}$		0		$0.2V_{DD}$	V

注意 即使在 N 沟道漏极开路模式中，P00、P10 ~ P15、P17 的 V_{IH} 最大值 (MAX.) 也为 V_{DD} 。

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。

($T_A=-40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$) (4/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
高电平输出电压	V_{OH1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P51、P120、P147	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH1}=-10.0\text{mA}$	$V_{DD}-1.5$			V
			$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH1}=-3.0\text{mA}$	$V_{DD}-0.7$			V
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH1}=-2.0\text{mA}$	$V_{DD}-0.6$			V
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH1}=-1.5\text{mA}$	$V_{DD}-0.5$			V
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH1}=-1.0\text{mA}$	$V_{DD}-0.5$			V
	V_{OH2}	P20 ~ P23	$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OH2}=-100\mu\text{A}$	$V_{DD}-0.5$			V
低电平输出电压	V_{OL1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、 P51、P70、P120、P147	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=20.0\text{mA}$			1.3	V
			$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=8.5\text{mA}$			0.7	V
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=3.0\text{mA}$			0.6	V
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=1.5\text{mA}$			0.4	V
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=0.6\text{mA}$			0.4	V
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL1}=0.3\text{mA}$			0.4	V
	V_{OL2}	P20 ~ P23	$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL2}=400\mu\text{A}$			0.4	V
	V_{OL3}	P60 ~ P62	$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL3}=15.0\text{mA}$			2.0	V
			$4.0\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL3}=5.0\text{mA}$			0.4	V
			$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL3}=3.0\text{mA}$			0.4	V
			$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL3}=2.0\text{mA}$			0.4	V
			$1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $I_{OL3}=1.0\text{mA}$			0.4	V

注意 在 N 沟道漏极开路模式中，P00、P10 ~ P15、P17、P50 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。

($T_A=-40 \sim +85^\circ\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$) (5/5)

项目	符号	条件	MIN.	TYP.	MAX.	单位		
高电平输入漏电流	I_{LH1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、P51、 P60 ~ P62、P70、120、147	$V_I=V_{DD}$			1	μA	
	I_{LH2}	P20 ~ P23、P137、 $\overline{\text{RESET}}$	$V_I=V_{DD}$			1	μA	
	I_{LH3}	X1、X2、EXCLK	$V_I=V_{DD}$	输入端口和 外部时钟输入时			1	μA
				连接谐振器时			10	μA
低电平输入漏电流	I_{LIL1}	P00、P01、P10 ~ P17、 P30、P31、P40、P50、P51、 P60 ~ P62、P70、120、147	$V_I=V_{SS}$			-1	μA	
	I_{LIL2}	P20 ~ P23、P137、 $\overline{\text{RESET}}$	$V_I=V_{SS}$			-1	μA	
	I_{LIL3}	X1、X2、EXCLK	$V_I=V_{SS}$	输入端口和 外部时钟输入时			-1	μA
				连接谐振器时			-10	μA
内部上拉电阻	R_U	P00、P01、P10 ~ P17、 P30、P31、P40、P50、P51、 P70、P120、P147	$V_I=V_{SS}$ ，输入端口时	10	20	100	$\text{k}\Omega$	

备注 在没有特别指定的情况下，复用引脚的特性和端口引脚的特性相同。

27.3.2 电源电流特性

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V) (1/2)

项目	符号	条件				MIN.	TYP.	MAX.	单位	
电源电流 ^{注1}	I _{DD1}	运行模式	HS (高速主) 模式 ^{注4}	f _{IH} =24MHz ^{注3}	基本	V _{DD} =5.0V		3.7	5.5	mA
					运行	V _{DD} =3.0V		3.7	5.5	
				基本	V _{DD} =5.0V		2.7	4.0		
					V _{DD} =3.0V		2.7	4.0		
			LS (低速主) 模式 ^{注4}	f _{IH} =8MHz ^{注3}	通常	V _{DD} =3.0V		1.2	1.8	mA
					运行	V _{DD} =2.0V		1.2	1.8	
			LV (低电压主) 模式 ^{注4}	f _{IH} =4MHz ^{注3}	通常	V _{DD} =3.0V		1.2	1.7	mA
					运行	V _{DD} =2.0V		1.2	1.7	
		HS (高速主) 模式 ^{注4}	f _{MX} =20MHz ^{注2} 、 V _{DD} =5.0V	通常	输入方波		3.0	4.6	mA	
					运行	连接谐振器		3.2		4.8
				通常	输入方波		3.0	4.6		
					运行	连接谐振器		3.2		4.8
				通常	f _{MX} =20MHz ^{注2} 、 V _{DD} =3.0V	输入方波		1.9		2.7
					运行	连接谐振器		1.9		2.7
通常	f _{MX} =10MHz ^{注2} 、 V _{DD} =5.0V			输入方波		1.9	2.7			
	运行			连接谐振器		1.9	2.7			
LS (低速主) 模式 ^{注4}	f _{MX} =8MHz ^{注2} 、 V _{DD} =3.0V	通常	输入方波		1.1	1.7	mA			
		运行	连接谐振器		1.1	1.7				
	通常	f _{MX} =8MHz ^{注2} 、 V _{DD} =2.0V	输入方波		1.1	1.7				
		运行	连接谐振器		1.1	1.7				

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、LVD 电路、I/O 端口以及内部上拉或者下拉电阻的电流，也不包含改写数据闪存时的电流。

2. 这是高速内部振荡器停止振荡的情况。
3. 这是高速系统时钟停止振荡的情况。
4. 工作电压范围、CPU 工作频率和运行模式的关系如下所示：
 HS (高速主) 模式： 2.7V ≤ V_{DD} ≤ 5.5V@1MHz ~ 24MHz
 2.4V ≤ V_{DD} ≤ 5.5V@1MHz ~ 16MHz
 LS (低速主) 模式： 1.8V ≤ V_{DD} ≤ 5.5V@1MHz ~ 8MHz
 LV (低电压主) 模式： 1.6V ≤ V_{DD} ≤ 5.5V@1MHz ~ 4MHz

备注 1. 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)

2. f_{IH}: 高速内部振荡器的时钟频率
3. TYP. 值的温度条件是 T_A=25°C。

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V) (2/2)

项目	符号	条件			MIN.	TYP.	MAX.	单位	
电源电流 ^{注1}	I _{DD2} ^{注2}	HALT 模式	HS (高速主) 模式 ^{注6}	f _{IH} =24MHz ^{注4}	V _{DD} =5.0V	0.44	1.28	mA	
				V _{DD} =3.0V	0.44	1.28			
			f _{IH} =16MHz ^{注4}	V _{DD} =5.0V	0.40	1.00			
				V _{DD} =3.0V	0.40	1.00			
			LS (低速主) 模式 ^{注6}	f _{IH} =8MHz ^{注4}	V _{DD} =3.0V	260	530		μA
				V _{DD} =2.0V	260	530			
		LV (低电压主) 模式 ^{注6}	f _{IH} =4MHz ^{注4}	V _{DD} =3.0V	420	640	μA		
				V _{DD} =2.0V	420	640			
		HS (高速主) 模式 ^{注6}	f _{MX} =20MHz ^{注3} 、 V _{DD} =5.0V	输入方波	0.28	1.00	mA		
				连接谐振器	0.45	1.17			
			f _{MX} =20MHz ^{注3} 、 V _{DD} =3.0V	输入方波	0.28	1.00			
				连接谐振器	0.45	1.17			
	f _{MX} =10MHz ^{注3} 、 V _{DD} =5.0V		输入方波	0.19	0.60				
			连接谐振器	0.26	0.67				
	f _{MX} =10MHz ^{注3} 、 V _{DD} =3.0V		输入方波	0.19	0.60				
			连接谐振器	0.26	0.67				
	LS (低速主) 模式 ^{注6}	f _{MX} =8MHz ^{注3} 、 V _{DD} =3.0V	输入方波	95	330	μA			
			连接谐振器	145	380				
		f _{MX} =8MHz ^{注3} 、 V _{DD} =2.0V	输入方波	95	330				
			连接谐振器	145	380				
I _{DD3} ^{注5}	STOP 模式	T _A =-40°C			0.18	0.50	μA		
		T _A =+25°C			0.23	0.50			
		T _A =+50°C			0.30	1.10			
		T _A =+70°C			0.46	1.90			
		T _A =+85°C			0.75	3.30			

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、LVD 电路、I/O 端口以及内部上拉或者下拉电阻的电流，也不包含改写数据闪存时的电流。

2. 这是执行闪存中的 HALT 指令的情况。
3. 这是高速内部振荡器停止振荡的情况。
4. 这是高速系统时钟停止振荡的情况。
5. 不包含流到 12 位间隔定时器和看门狗定时器的电流。
6. 工作电压范围、CPU 工作频率和运行模式的关系如下所示：
 HS (高速主) 模式： 2.7V ≤ V_{DD} ≤ 5.5V@1MHz ~ 24MHz
 2.4V ≤ V_{DD} ≤ 5.5V@1MHz ~ 16MHz
 LS (低速主) 模式： 1.8V ≤ V_{DD} ≤ 5.5V@1MHz ~ 8MHz
 LV (低电压主) 模式： 1.6V ≤ V_{DD} ≤ 5.5V@1MHz ~ 4MHz

备注 1. 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)

2. f_{IH}: 高速内部振荡器的时钟频率
3. STOP 模式以外的 TYP. 值的温度条件是 T_A=25°C。

(1) 外围功能

(T_A=-40 ~ +85°C、1.6V ≤ 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				0.20		μA
12 位间隔定时器工作电流	I _{IT} 注 1、2、3				0.02		μA
看门狗定时器工作电流	I _{WDT} 注 1、2、4	f _{IL} =15kHz			0.22		μA
A/D 转换器工作电流	I _{ADC} 注 1、5	最高速转换时	标准模式, AV _{REFP} =V _{DD} =5.0V		1.3	1.7	mA
			低电压模式, AV _{REFP} =V _{DD} =3.0V		0.5	0.7	mA
A/D 转换器基准电压电流	I _{ADREF} 注 1				75.0		μA
LVD 工作电流	I _{LVD} 注 1、6				0.08		μA
自编程工作电流	I _{FSP} 注 1、8				2.50	12.20	mA
BGO 电流	I _{BGO} 注 1、7				2.50	12.20	mA
SNOOZE 工作电流	I _{SNOZ} 注 1	ADC 运行	模式转移中注 9		0.50	0.60	mA
			转换运行中, 低电压模式 AV _{REFP} =V _{DD} =3.0V		1.20	1.44	
		CSI/UART 运行				0.70	0.84

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)。在运行模式或者 HALT 模式中 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} 或者 I_{DD2} 加上 I_{IT} 的值。另外, 当选择低速内部振荡器时, 必须加上 I_{FIL}。4. 这是只流到看门狗定时器的电流 (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} 或者 I_{DD2} 或者 I_{DD3} 加上 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 LVD 电路的电流。在 LVD 电路运行的情况下, RL78 微控制器的电流值为 I_{DD1} 或者 I_{DD2} 或者 I_{DD3} 加上 I_{LVD} 的值。

7. 这是改写数据闪存时的电流。

9. 这是自编程时的电流。

10. 有关 SNOOZE 模式的转移时间, 请参照“16.3.3 SNOOZE 模式”。

备注 1. f_{IL}: 低速内部振荡器的时钟频率2. f_{CLK}: CPU/ 外围硬件的时钟频率3. TYP. 值的温度条件是 T_A=25°C。

27.4 AC 特性

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

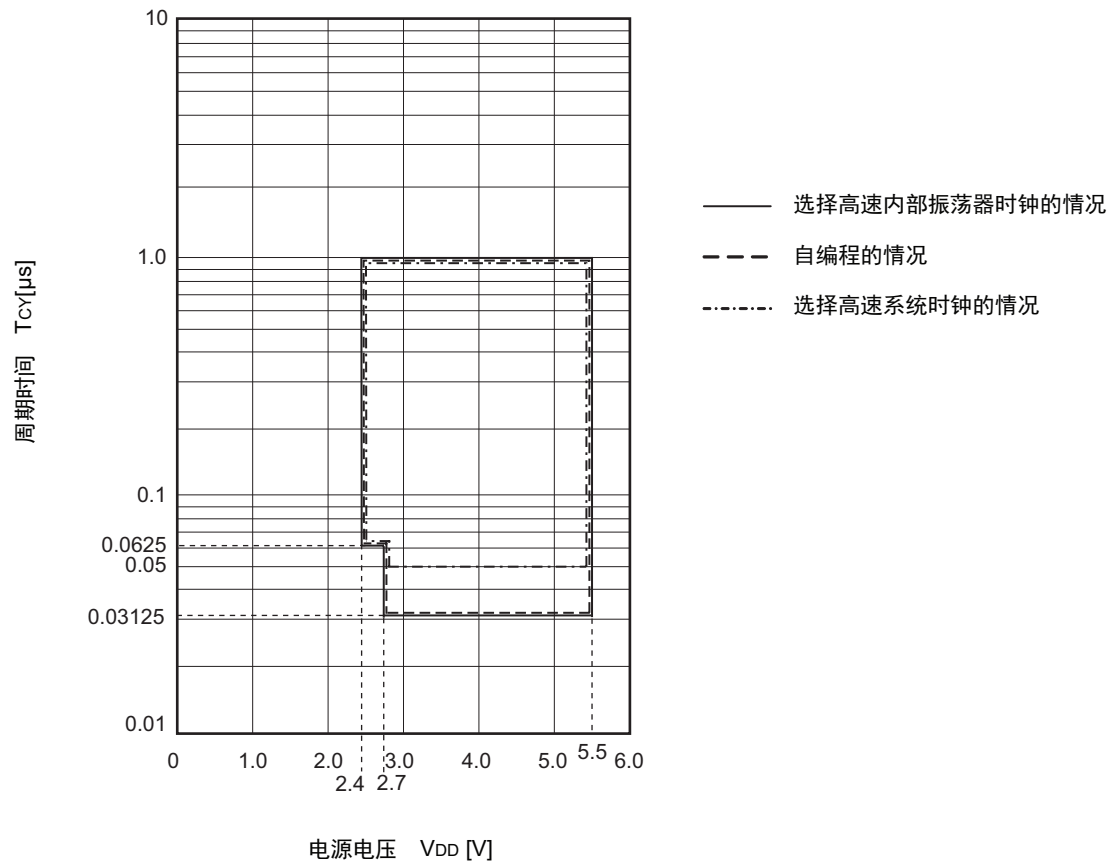
项目	符号	条件		MIN.	TYP.	MAX.	单位	
指令周期 (最短指令执行时间)	T _{CY}	主系统时钟 (f _{MAIN}) 运行	HS (高速主) 模式	2.7V ≤ V _{DD} ≤ 5.5V	0.03125		1	μs
				2.4V ≤ V _{DD} < 2.7V	0.0625		1	μs
			LS (低速主) 模式	1.8V ≤ V _{DD} ≤ 5.5V	0.125		1	μs
			LV (低电压主) 模式	1.6V ≤ V _{DD} ≤ 5.5V	0.25		1	μs
		自编程时	HS (高速主) 模式	2.7V ≤ V _{DD} ≤ 5.5V	0.03125		1	μs
				2.4V ≤ V _{DD} < 2.7V	0.0625		1	μs
			LS (低速主) 模式	1.8V ≤ V _{DD} ≤ 5.5V	0.125		1	μs
			LV (低电压主) 模式	1.8V ≤ V _{DD} ≤ 5.5V	0.25		1	μs
外部系统时钟频率	f _{EX}	2.7V ≤ V _{DD} ≤ 5.5V		1.0		20.0	MHz	
		2.4V ≤ V _{DD} < 2.7V		1.0		16.0	MHz	
		1.8V ≤ V _{DD} < 2.4V		1.0		8.0	MHz	
		1.6V ≤ V _{DD} < 1.8V		1.0		4.0	MHz	
外部系统时钟输入的高低电平宽度	t _{EXH} 、 t _{EXL}	2.7V ≤ V _{DD} ≤ 5.5V		24			ns	
		2.4V ≤ V _{DD} < 2.7V		30			ns	
		1.8V ≤ V _{DD} < 2.4V		60			ns	
		1.6V ≤ V _{DD} < 1.8V		120			ns	
TI00 ~ TI07 输入的高低电平宽度	t _{TIH} 、 t _{TIL}			1/f _{MCK} +10			ns	
TO00 ~ TO07 的输出频率	f _{TO}	HS (高速主) 模式	4.0V ≤ V _{DD} ≤ 5.5V			12	MHz	
			2.7V ≤ V _{DD} < 4.0V			8	MHz	
			1.8V ≤ V _{DD} < 2.7V			4	MHz	
			1.6V ≤ V _{DD} < 1.8V			2	MHz	
		LS (低速主) 模式	1.8V ≤ V _{DD} ≤ 5.5V			4	MHz	
			1.6V ≤ V _{DD} < 1.8V			2	MHz	
		LV (低电压主) 模式	1.6V ≤ V _{DD} ≤ 5.5V			2	MHz	
PCLBUZ0、PCLBUZ1 的输出频率	f _{PCL}	HS (高速主) 模式	4.0V ≤ V _{DD} ≤ 5.5V			16	MHz	
			2.7V ≤ V _{DD} < 4.0V			8	MHz	
			1.8V ≤ V _{DD} < 2.7V			4	MHz	
			1.6V ≤ V _{DD} < 1.8V			2	MHz	
		LS (低速主) 模式	1.8V ≤ V _{DD} ≤ 5.5V			4	MHz	
			1.6V ≤ V _{DD} < 1.8V			2	MHz	
		LV (低电压主) 模式	1.8V ≤ V _{DD} ≤ 5.5V			4	MHz	
			1.6V ≤ V _{DD} < 1.8V			2	MHz	
中断输入的高低电平宽度	t _{INTH} 、 t _{INTL}	INTP0	1.6V ≤ V _{DD} ≤ 5.5V	1			μs	
		INTP1 ~ INTP5	1.6V ≤ V _{DD} ≤ 5.5V	1			μs	
RESET 的低电平宽度	t _{RSL}			10			μs	

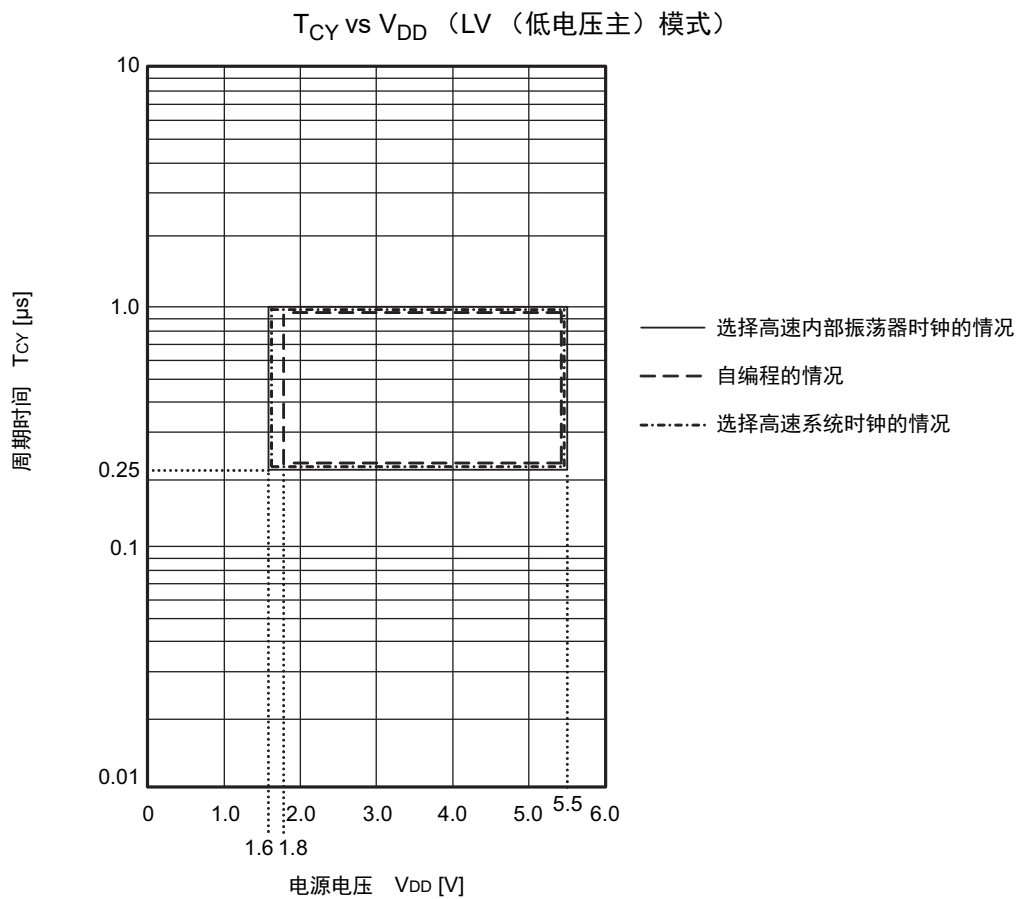
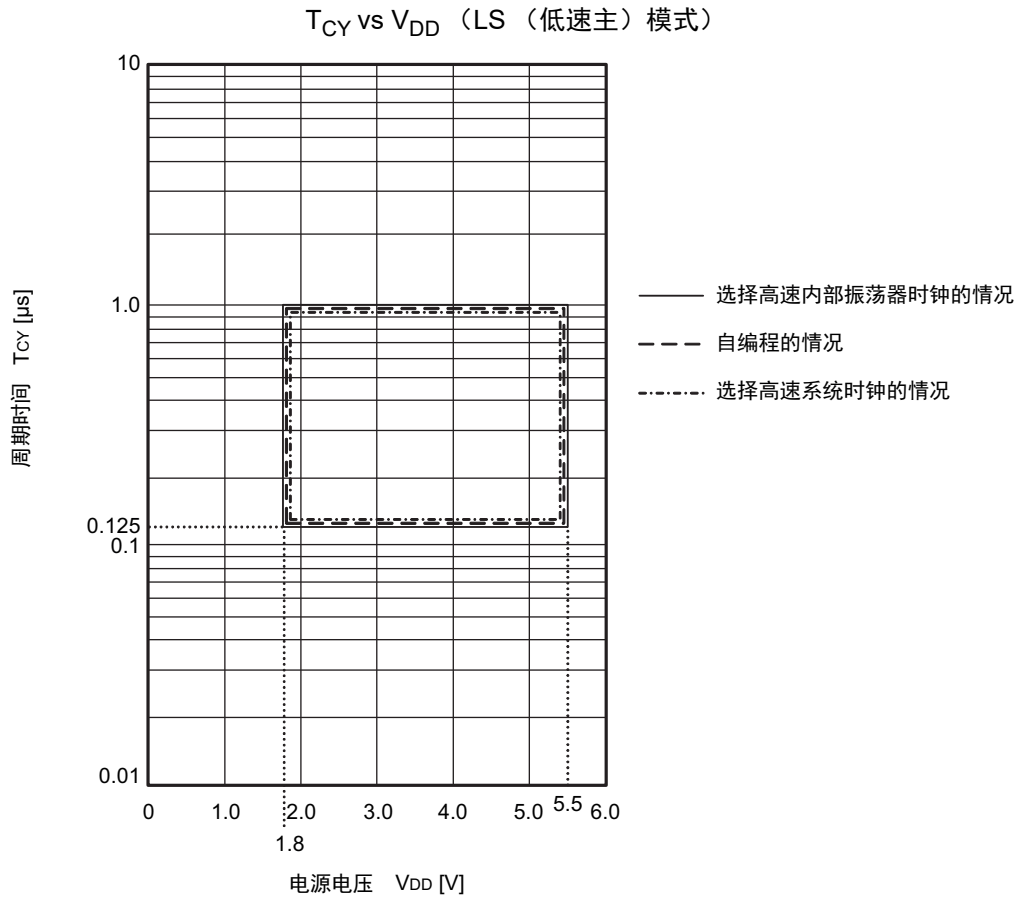
备注 f_{MCK}: 定时器阵列单元的运行时钟频率

(这是定时器模式寄存器 mn (TMRmn) 的 CKSmn0 位和 CKSmn1 位设定的运行时钟。

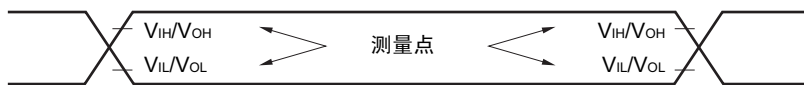
m: 单元号 (m=0、1) n: 通道号 (n=0~7))

主系统时钟运行时的最短指令执行时间

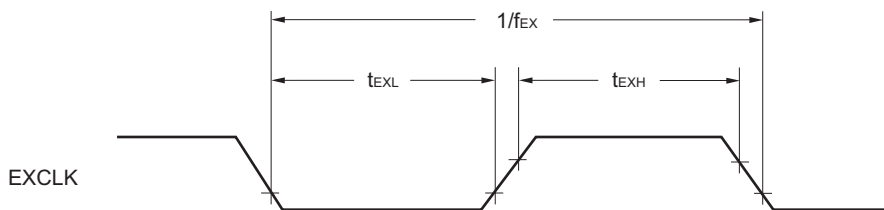
 T_{CY} vs V_{DD} (HS (高速主) 模式)



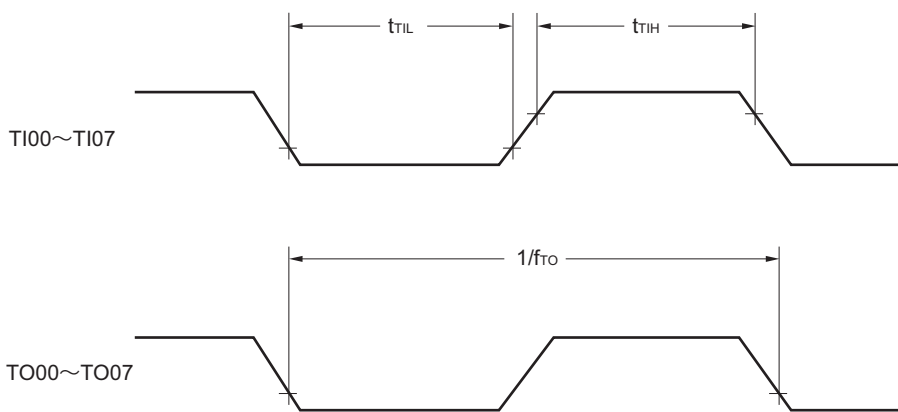
AC 时序测量点



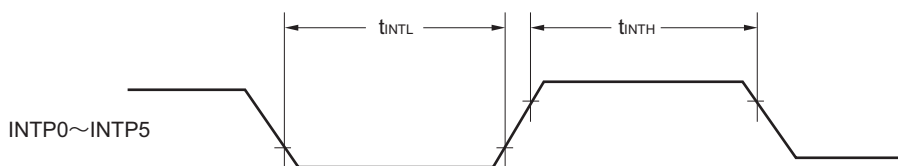
外部系统时钟的时序



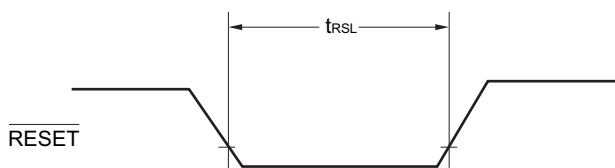
TI/TO 时序



中断请求的输入时序

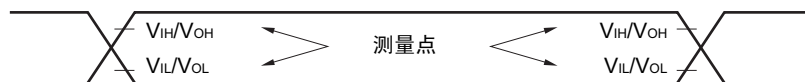


$\overline{\text{RESET}}$ 的输入时序



27.5 外围功能特性

AC 时序测量点



27.5.1 串行阵列单元

(1) 同电位的通信 (UART 模式)

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

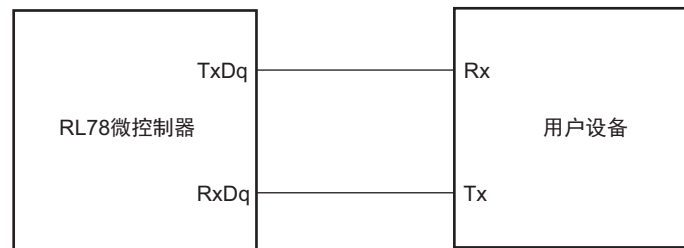
项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
传送速率注 1		2.4V ≤ V _{DD} ≤ 5.5V		f _{MCK} /6		f _{MCK} /6		f _{MCK} /6	bps
		最大传送速率的理论值 f _{MCK} =f _{CLK} 注 2		5.3		1.3		0.6	Mbps
		1.8V ≤ V _{DD} ≤ 5.5V		f _{MCK} /6		f _{MCK} /6		f _{MCK} /6	bps
		最大传送速率的理论值 f _{MCK} =f _{CLK} 注 2		5.3		1.3		0.6	Mbps
		1.7V ≤ V _{DD} ≤ 5.5V		f _{MCK} /6		f _{MCK} /6		f _{MCK} /6	bps
		最大传送速率的理论值 f _{MCK} =f _{CLK} 注 2		5.3		1.3		0.6	Mbps
1.6V ≤ V _{DD} ≤ 5.5V		—		f _{MCK} /6		f _{MCK} /6	bps		
	最大传送速率的理论值 f _{MCK} =f _{CLK} 注 2		—		1.3		0.6	Mbps	

注 1. SNOOZE 模式中的传送速率只为 4800bps。

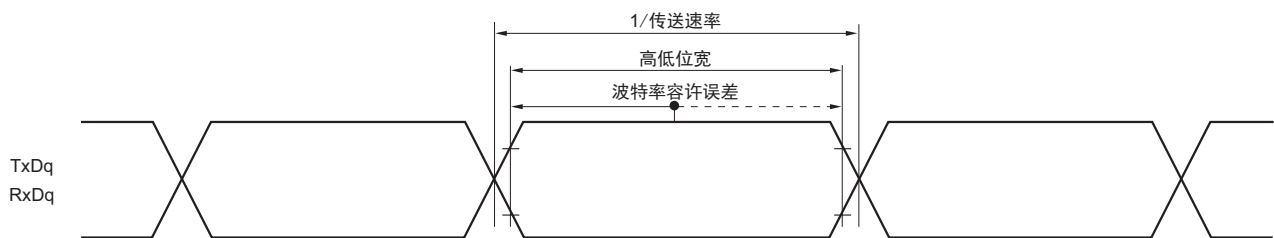
2. CPU/ 外围硬件时钟 (f_{CLK}) 的最大工作频率如下所示:HS (高速主) 模式: 24MHz (2.7V ≤ V_{DD} ≤ 5.5V)16MHz (2.4V ≤ V_{DD} ≤ 5.5V)LS (低速主) 模式: 8MHz (1.8V ≤ V_{DD} ≤ 5.5V)LV (低电压主) 模式: 4MHz (1.6V ≤ V_{DD} ≤ 5.5V)

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 RxDq 引脚选择为通常的输入缓冲器并且将 TxDq 引脚选择为通常的输出模式。

UART 模式的连接图 (同电位的通信)



UART 模式的位宽 (同电位的通信) (参考)



备注 1. q: UART 号 (q=0 ~ 2) g: PIM、POM 号 (g=0、1)

2. f_{MCK} : 串行阵列单元的运行时钟频率

(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。

m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13))

(2) 同电位的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出, 只对应 CSI00)

(T_A=-40 ~ +85°C、2.7V ≤ 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		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期时间	t _{KCY1}	t _{KCY1} ≥ 2/f _{CLK}	4.0V ≤ V _{DD} ≤ 5.5V	62.5		250		500		ns
			2.7V ≤ V _{DD} ≤ 5.5V	83.3		250		500		ns
SCKp 高低电平宽度	t _{KH1} 、 t _{KL1}	4.0V ≤ V _{DD} ≤ 5.5V		t _{KCY1} /2 -7		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns
		2.7V ≤ V _{DD} ≤ 5.5V		t _{KCY1} /2 -10		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK1}	4.0V ≤ V _{DD} ≤ 5.5V		23		110		110		ns
		2.7V ≤ V _{DD} ≤ 5.5V		33		110		110		ns
Slp 保持时间 (对 SCKp↑) 注 2	t _{KS11}	2.7V ≤ V _{DD} ≤ 5.5V		10		10		10		ns
SCKp↓→SOp 输出延迟时间 注 3	t _{KSO1}	C=20pF 注 4			10		10		10	ns

- 注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
3. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。
4. C 是 SCKp、SOp 输出线的负载电容。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 Slp 引脚选择为通常的输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为通常的输出模式。

备注 1. 此值只对应不使用 CSI00 的外围 I/O 重定向功能的情况。

2. p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0) g: PIM、POM 号 (g=1)
3. f_{MCK}: 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))

(3) 同电位的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出)

(T_A=−40 ~ +85°C、1.6V ≤ 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		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期时间	t _{KCY1}	t _{KCY1} ≥ 4/f _{CLK}	2.7V ≤ V _{DD} ≤ 5.5V	125		500		1000	ns
			2.4V ≤ V _{DD} ≤ 5.5V	250		500		1000	ns
			1.8V ≤ V _{DD} ≤ 5.5V	500		500		1000	ns
			1.7V ≤ V _{DD} ≤ 5.5V	1000		1000		1000	ns
			1.6V ≤ V _{DD} ≤ 5.5V	—		1000		1000	ns
SCKp 高低电平宽度	t _{KH1} 、 t _{KL1}	4.0V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2 −12		t _{KCY1} /2 −50		t _{KCY1} /2 −50	ns	
		2.7V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2 −18		t _{KCY1} /2 −50		t _{KCY1} /2 −50	ns	
		2.4V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2 −38		t _{KCY1} /2 −50		t _{KCY1} /2 −50	ns	
		1.8V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2 −50		t _{KCY1} /2 −50		t _{KCY1} /2 −50	ns	
		1.7V ≤ V _{DD} ≤ 5.5V	t _{KCY1} /2 −100		t _{KCY1} /2 −100		t _{KCY1} /2 −100	ns	
		1.6V ≤ V _{DD} ≤ 5.5V	—		t _{KCY1} /2 −100		t _{KCY1} /2 −100	ns	
Slp 准备时间 (对 SCKp↑) 注 1	t _{S1K1}	4.0V ≤ V _{DD} ≤ 5.5V	44		110		110	ns	
		2.7V ≤ V _{DD} ≤ 5.5V	44		110		110	ns	
		2.4V ≤ V _{DD} ≤ 5.5V	75		110		110	ns	
		1.8V ≤ V _{DD} ≤ 5.5V	110		110		110	ns	
		1.7V ≤ V _{DD} ≤ 5.5V	220		220		220	ns	
		1.6V ≤ V _{DD} ≤ 5.5V	—		220		220	ns	
Slp 保持时间 (对 SCKp↑) 注 2	t _{KS11}	1.7V ≤ V _{DD} ≤ 5.5V	19		19		19	ns	
		1.6V ≤ V _{DD} ≤ 5.5V	—		19		19	ns	
SCKp↓→SOp 输出延迟时间 注 3	t _{KSO1}	1.7V ≤ V _{DD} ≤ 5.5V C=30pF 注 4		25		25		25	ns
		1.6V ≤ V _{DD} ≤ 5.5V C=30pF 注 4		—		25		25	ns

- 注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↓”。
3. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时, 为“对 SCKp↑”。
4. C 是 SCKp、SOp 输出线的负载电容。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 Slp 引脚选择为通常的输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为通常的输出模式。

- 备注 1. p: CSI 号 (p=00、11、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、5)
2. f_{MCK}: 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13))

(4) 同电位的通信 (CSI 模式) (从属模式, SCKp..... 外部时钟输入)

(T_A=−40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

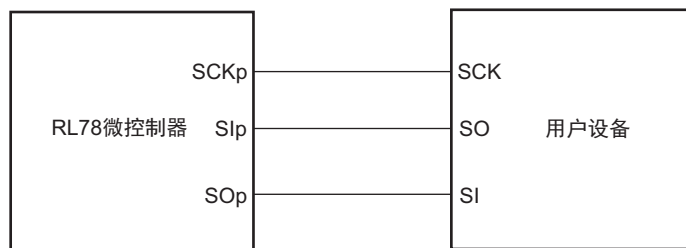
项目	符号	条件		HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间注 5	t _{KCY2}	4.0V ≤ V _{DD} ≤ 5.5V	20MHz < f _{MCK}	8/f _{MCK}	—	—	—	—	ns	
			f _{MCK} ≤ 20MHz	6/f _{MCK}	6/f _{MCK}	6/f _{MCK}	6/f _{MCK}	ns		
		2.7V ≤ V _{DD} ≤ 5.5V	16MHz < f _{MCK}	8/f _{MCK}	—	—	—	ns		
			f _{MCK} ≤ 16MHz	6/f _{MCK}	6/f _{MCK}	6/f _{MCK}	6/f _{MCK}	ns		
		2.4V ≤ V _{DD} ≤ 5.5V			6/f _{MCK} 并且 500	6/f _{MCK} 并且 500	6/f _{MCK} 并且 500	6/f _{MCK} 并且 500	ns	
		1.8V ≤ V _{DD} ≤ 5.5V			6/f _{MCK} 并且 750	6/f _{MCK} 并且 750	6/f _{MCK} 并且 750	6/f _{MCK} 并且 750	ns	
		1.7V ≤ V _{DD} ≤ 5.5V			6/f _{MCK} 并且 1500	6/f _{MCK} 并且 1500	6/f _{MCK} 并且 1500	6/f _{MCK} 并且 1500	ns	
1.6V ≤ V _{DD} ≤ 5.5V			—	6/f _{MCK} 并且 1500	6/f _{MCK} 并且 1500	6/f _{MCK} 并且 1500	ns			
SCKp 高低 电平宽度	t _{KH2} 、 t _{KL2}	4.0V ≤ V _{DD} ≤ 5.5V		t _{KCY2} /2 −7	t _{KCY2} /2 −7	t _{KCY2} /2 −7	t _{KCY2} /2 −7	ns		
		2.7V ≤ V _{DD} ≤ 5.5V		t _{KCY2} /2 −8	t _{KCY2} /2 −8	t _{KCY2} /2 −8	t _{KCY2} /2 −8	ns		
		1.8V ≤ V _{DD} ≤ 5.5V		t _{KCY2} /2 −18	t _{KCY2} /2 −18	t _{KCY2} /2 −18	t _{KCY2} /2 −18	ns		
		1.7V ≤ V _{DD} ≤ 5.5V		t _{KCY2} /2 −66	t _{KCY2} /2 −66	t _{KCY2} /2 −66	t _{KCY2} /2 −66	ns		
		1.6V ≤ V _{DD} ≤ 5.5V		—	t _{KCY2} /2 −66	t _{KCY2} /2 −66	t _{KCY2} /2 −66	ns		
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK2}	2.7V ≤ V _{DD} ≤ 5.5V		1/f _{MCK} +20	1/f _{MCK} +30	1/f _{MCK} +30	1/f _{MCK} +30	ns		
		1.8V ≤ V _{DD} ≤ 5.5V		1/f _{MCK} +30	1/f _{MCK} +30	1/f _{MCK} +30	1/f _{MCK} +30	ns		
		1.7V ≤ V _{DD} ≤ 5.5V		1/f _{MCK} +40	1/f _{MCK} +40	1/f _{MCK} +40	1/f _{MCK} +40	ns		
		1.6V ≤ V _{DD} ≤ 5.5V		—	1/f _{MCK} +40	1/f _{MCK} +40	1/f _{MCK} +40	ns		
Slp 保持时间 (对 SCKp↑) 注 2	t _{KS12}	1.8V ≤ V _{DD} ≤ 5.5V		1/f _{MCK} +31	1/f _{MCK} +31	1/f _{MCK} +31	1/f _{MCK} +31	ns		
		1.7V ≤ V _{DD} ≤ 5.5V		1/f _{MCK} +250	1/f _{MCK} +250	1/f _{MCK} +250	1/f _{MCK} +250	ns		
		1.6V ≤ V _{DD} ≤ 5.5V		—	1/f _{MCK} +250	1/f _{MCK} +250	1/f _{MCK} +250	ns		
SCKp↓→SOp 输出延迟时间 注 3	t _{KSO2}	C=30pF 注 4	2.7V ≤ V _{DD} ≤ 5.5V	2/f _{MCK} +44	2/f _{MCK} +110	2/f _{MCK} +110	2/f _{MCK} +110	ns		
			2.4V ≤ V _{DD} ≤ 5.5V	2/f _{MCK} +75	2/f _{MCK} +110	2/f _{MCK} +110	2/f _{MCK} +110	ns		
			1.8V ≤ V _{DD} ≤ 5.5V	2/f _{MCK} +100	2/f _{MCK} +110	2/f _{MCK} +110	2/f _{MCK} +110	ns		
			1.7V ≤ V _{DD} ≤ 5.5V	2/f _{MCK} +220	2/f _{MCK} +220	2/f _{MCK} +220	2/f _{MCK} +220	ns		
			1.6V ≤ V _{DD} ≤ 5.5V	—	2/f _{MCK} +220	2/f _{MCK} +220	2/f _{MCK} +220	ns		

- 注
1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↓”。
 2. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↓”。
 3. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↑”。
 4. C 是 SOp 输出线的负载电容。
 5. SNOOZE 模式中的传送速率为 MAX. 1Mbps。

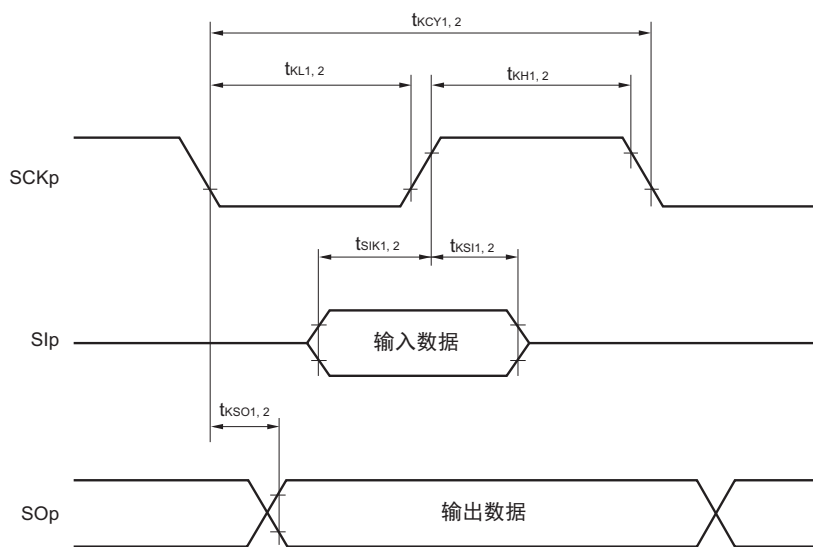
注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 Slp 引脚和 SCKp 引脚选择为通常的输入缓冲器并且将 SOp 引脚选择为通常的输出模式。

- 备注 1. p: CSI 号 (p=00、11、20) m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
g: PIM、POM 号 (g=0、1、5)
2. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13))

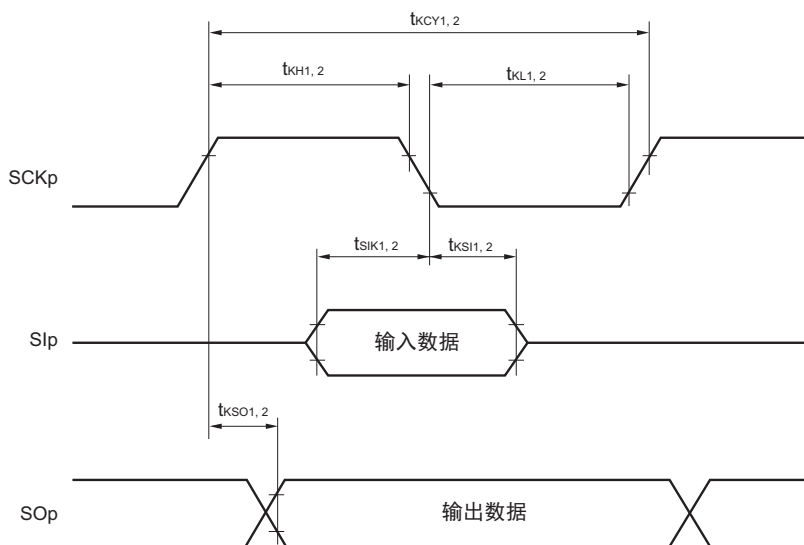
CSI 模式的连接图 (同电位的通信)



CSI 模式的串行传送时序 (同电位的通信)
(DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况)



CSI 模式的串行传送时序 (同电位的通信)
(DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 的情况)



备注 1. p: CSI 号 (p=00、11、20)

2. m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13)

(5) 同电位的通信 (简易 I²C 模式) (1/2)(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

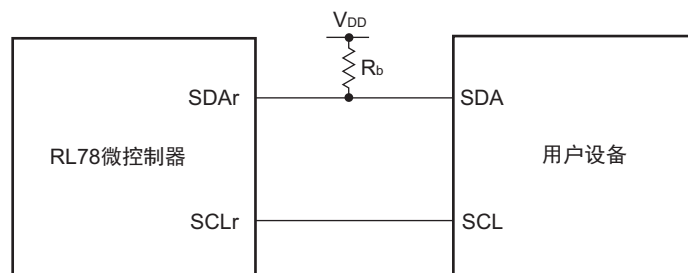
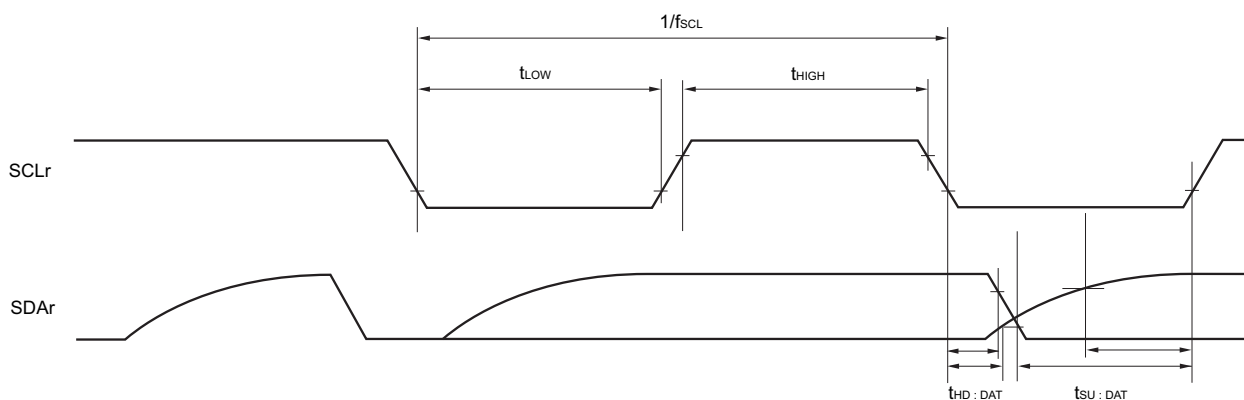
项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr 时钟频率	f _{SCL}	2.7V ≤ V _{DD} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ		1000 注1		400 注1		400 注1	kHz
		1.8V ≤ V _{DD} ≤ 5.5V、 C _b =100pF、R _b =3kΩ		400 注1		400 注1		400 注1	kHz
		1.8V ≤ V _{DD} < 2.7V、 C _b =100pF、R _b =5kΩ		300 注1		300 注1		300 注1	kHz
		1.7V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ		250 注1		250 注1		250 注1	kHz
		1.6V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ		—		250 注1		250 注1	kHz
SCLr="L" 的 保持时间	t _{LOW}	2.7V ≤ V _{DD} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	475		1150		1150		ns
		1.8V ≤ V _{DD} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	1150		1150		1150		ns
		1.8V ≤ V _{DD} < 2.7V、 C _b =100pF、R _b =5kΩ	1550		1550		1550		ns
		1.7V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ	1850		1850		1850		ns
		1.6V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ	—		1850		1850		ns
SCLr="H" 的 保持时间	t _{HIGH}	2.7V ≤ V _{DD} ≤ 5.5V、 C _b =50pF、R _b =2.7kΩ	475		1150		1150		ns
		1.8V ≤ V _{DD} ≤ 5.5V、 C _b =100pF、R _b =3kΩ	1150		1150		1150		ns
		1.8V ≤ V _{DD} < 2.7V、 C _b =100pF、R _b =5kΩ	1550		1550		1550		ns
		1.7V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ	1850		1850		1850		ns
		1.6V ≤ V _{DD} < 1.8V、 C _b =100pF、R _b =5kΩ	—		1850		1850		ns

(5) 同电位的通信 (简易 I²C 模式) (2/2)($T_A=-40 \sim +85^\circ\text{C}$, $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS}=0\text{V}$)

项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
数据准备时间 (接收时)	$t_{\text{SU: DAT}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $C_b=50\text{pF}$ 、 $R_b=2.7\text{k}\Omega$	$1/f_{\text{MCK}}$ +85 注 2		$1/f_{\text{MCK}}$ +145 注 2		$1/f_{\text{MCK}}$ +145 注 2		ns
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=3\text{k}\Omega$	$1/f_{\text{MCK}}$ +145 注 2		$1/f_{\text{MCK}}$ +145 注 2		$1/f_{\text{MCK}}$ +145 注 2		ns
		$1.8\text{V} \leq V_{DD} < 2.7\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	$1/f_{\text{MCK}}$ +230 注 2		$1/f_{\text{MCK}}$ +230 注 2		$1/f_{\text{MCK}}$ +230 注 2		ns
		$1.7\text{V} \leq V_{DD} < 1.8\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	$1/f_{\text{MCK}}$ +290 注 2		$1/f_{\text{MCK}}$ +290 注 2		$1/f_{\text{MCK}}$ +290 注 2		ns
		$1.6\text{V} \leq V_{DD} < 1.8\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	—		$1/f_{\text{MCK}}$ +290 注 2		$1/f_{\text{MCK}}$ +290 注 2		ns
数据保持时间 (发送时)	$t_{\text{HD: DAT}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $C_b=50\text{pF}$ 、 $R_b=2.7\text{k}\Omega$	0	305	0	305	0	305	ns
		$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=3\text{k}\Omega$	0	355	0	355	0	355	ns
		$1.8\text{V} \leq V_{DD} < 2.7\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	0	405	0	405	0	405	ns
		$1.7\text{V} \leq V_{DD} < 1.8\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	0	405	0	405	0	405	ns
		$1.6\text{V} \leq V_{DD} < 1.8\text{V}$ 、 $C_b=100\text{pF}$ 、 $R_b=5\text{k}\Omega$	—		0	405	0	405	ns

注 1. 并且必须至少设定为 $f_{\text{MCK}}/4$ 。2. f_{MCK} 的设定值不能超过 $\text{SCLr}=\text{“L”}$ 和 $\text{SCLr}=\text{“H”}$ 的保持时间。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 h (POMh), 将 SDAr 引脚选择为通常的输入缓冲器和 N 沟道漏极开路输出 (V_{DD} 耐压) 模式, 并且将 SCLr 引脚选择为通常的输出模式。

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 (同电位的通信)

备注 1. $R_b[\Omega]$: 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、11、20) g: PIM 号 (g=0、1)

h: POM 号 (h=0、1、5)

3. f_{MCK} : 串行阵列单元的运行时钟频率

(这是 SMRmn 寄存器的 CKSmn 位设定的运行时钟。m: 单元号 (m=0、1) n: 通道号 (n=0 ~ 3)
mn=00 ~ 03、10 ~ 13)

(6) 不同电位 (1.8V、2.5V、3V) 的通信 (UART 模式) (1/2)

(T_A=-40 ~ +85°C、1.8V ≤ 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		LV (低电压主) 模式		单位	
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
传送速率		接收	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V		f _{MCK} /6 ^{注1}		f _{MCK} /6 ^{注1}		f _{MCK} /6 ^{注1}	bps
			最大传送速率的理论值 f _{MCK} =f _{CLK} ^{注3}		5.3		1.3		0.6	Mbps
			2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V		f _{MCK} /6 ^{注1}		f _{MCK} /6 ^{注1}		f _{MCK} /6 ^{注1}	bps
			最大传送速率的理论值 f _{MCK} =f _{CLK} ^{注3}		5.3		1.3		0.6	Mbps
			1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V		f _{MCK} /6 注1、2		f _{MCK} /6 注1、2		f _{MCK} /6 注1、2	bps
			最大传送速率的理论值 f _{MCK} =f _{CLK} ^{注3}		5.3		1.3		0.6	Mbps

- 注 1. SNOOZE 模式中的传送速率只为 4800bps。
 2. 必须在 V_{DD} ≥ V_b 的情况下使用。
 3. CPU/外围硬件时钟 (f_{CLK}) 的最大工作频率如下所示：
 HS (高速主) 模式： 24MHz (2.7V ≤ V_{DD} ≤ 5.5V)
 16MHz (2.4V ≤ V_{DD} ≤ 5.5V)
 LS (低速主) 模式： 8MHz (1.8V ≤ V_{DD} ≤ 5.5V)
 LV (低电压主) 模式： 4MHz (1.6V ≤ V_{DD} ≤ 5.5V)

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 RxDq 引脚选择为 TTL 输入缓冲器并且将 TxDq 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

- 备注 1. V_b[V]: 通信线的电压
 2. q: UART 号 (q=0 ~ 2) g: PIM、POM 号 (g=0、1)
 3. f_{MCK}: 串行阵列单元的运行时钟频率
 (这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
 m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13))
 4. 当外围 I/O 重定向寄存器 (PIOR) 的 bit1 (PIOR1) 为“1”时，不能使用 UART2 进行不同电位的通信。

(6) 不同电位 (1.8V、2.5V、3V) 的通信 (UART 模式) (2/2)

(T_A=-40 ~ +85°C、1.8V ≤ 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		LV (低电压主) 模式		单位	
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.		
传送速率		发送	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V		注 1		注 1		注 1	bps
			最大传送速率的理论值 C _b =50pF、R _b =1.4kΩ、 V _b =2.7V		2.8 注 2		2.8 注 2		2.8 注 2	Mbps
			2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V		注 3		注 3		注 3	bps
			最大传送速率的理论值 C _b =50pF、R _b =2.7kΩ、 V _b =2.3V		1.2 注 4		1.2 注 4		1.2 注 4	Mbps
			1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V		注 5、6		注 5、6		注 5、6	bps
			最大传送速率的理论值 C _b =50pF、R _b =5.5kΩ、 V _b =1.6V		0.43 注 7		0.43 注 7		0.43 注 7	Mbps

注 1. f_{MCK}/6 和用以下计算式算出的最大传送速率中的较小的值为有效的最大传送速率。4.0V ≤ V_{DD} ≤ 5.5V、2.7V ≤ V_b ≤ 4.0V 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{\left(\frac{1}{\text{传送速率}}\right) \times \text{传送位数}} \times 100 \text{ [%]}$$

※ 此值为发送方和接收方的相对误差的理论值。

2. 作为例子，此值是满足条件栏中条件的计算值。有关用户条件下的最大传送速率，必须根据注 1 进行计算。

3. f_{MCK}/6 和用以下计算式算出的最大传送速率中的较小的值为有效的最大传送速率。2.7V ≤ V_{DD} < 4.0V、2.3V ≤ V_b ≤ 2.7V 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{\left(\frac{1}{\text{传送速率}}\right) \times \text{传送位数}} \times 100 \text{ [%]}$$

※ 此值为发送方和接收方的相对误差的理论值。

- 注 4. 作为例子，此值是满足条件栏中条件的计算值。有关用户条件下的最大传送速率，必须根据注 3 进行计算。
5. 必须在 $V_{DD} \geq V_b$ 的情况下使用。
6. $f_{MCK}/6$ 和用以下计算式算出的最大传送速率中的较小的值为有效的最大传送速率。
 $1.8V \leq V_{DD} < 3.3V$ 、 $1.6V \leq V_b \leq 2.0V$ 时的传送速率计算式：

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

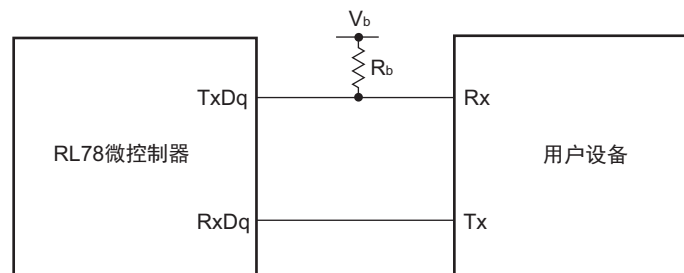
$$\text{波特率容许误差 (理论值)} = \frac{\frac{1}{\text{传送速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{传送速率}}) \times \text{传送位数}} \times 100 \text{ [%]}$$

※ 此值为发送方和接收方的相对误差的理论值。

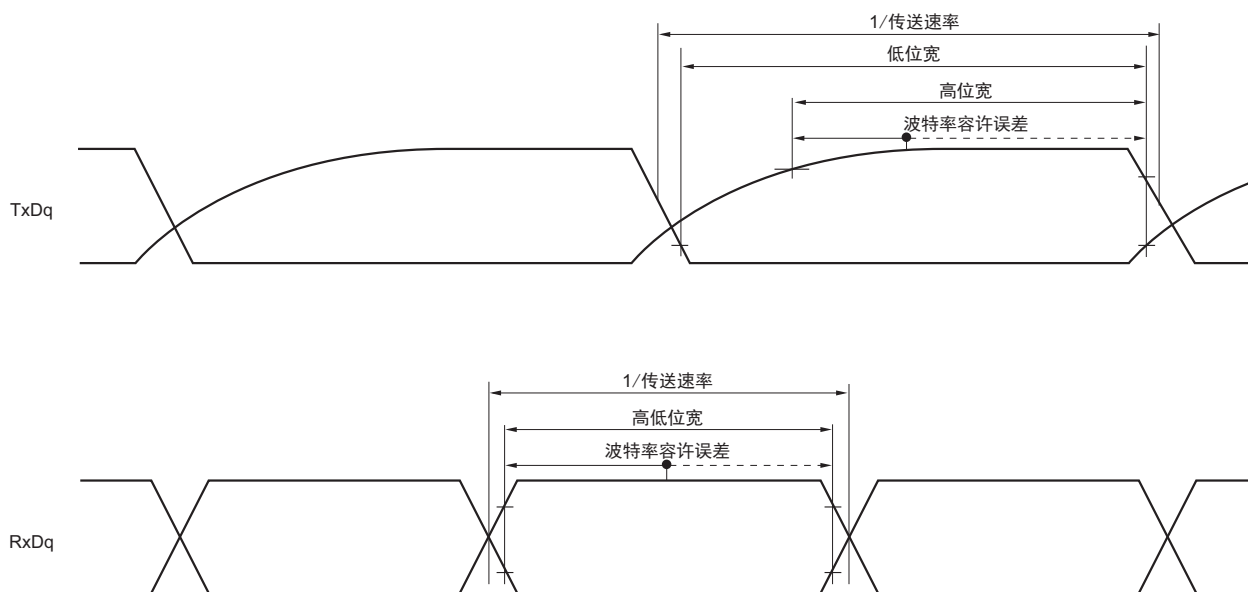
7. 作为例子，此值是满足条件栏中条件的计算值。有关用户条件下的最大传送速率，必须根据注 6 进行计算。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 RxDq 引脚选择为 TTL 输入缓冲器并且将 TxDq 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。 V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

UART 模式的连接图 (不同电位的通信)



UART 模式的位宽 (不同电位的通信) (参考)



- 备注 1. $R_b[\Omega]$: 通信线 (TxDq) 的上拉电阻值 $C_b[F]$: 通信线 (TxDq) 的负载电容值 $V_b[V]$: 通信线的电压
2. q: UART 号 (q=0 ~ 2) g: PIM、POM 号 (g=0、1)
3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
m: 单元号、n: 通道号 (mn=00 ~ 03、10 ~ 13))
4. 当外围 I/O 重定向寄存器 (PIOR) 的 bit1 (PIOR1) 为“1”时, 不能使用 UART2 进行不同电位的通信。

(7) 不同电位 (2.5V、3V) 的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出, 只限于 CSI00) (1/2)
(T_A=-40 ~ +85°C、2.7V ≤ 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		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间	t _{KCY1}	t _{KCY1} ≥ 2/f _{CLK}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ	200		1150		1150		ns
			2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ	300		1150		1150		ns
SCKp 高电平 宽度	t _{KH1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ		t _{KCY1} /2 -50		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ		t _{KCY1} /2 -120		t _{KCY1} /2 -120		t _{KCY1} /2 -120		ns
SCKp 低电平 宽度	t _{KL1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ		t _{KCY1} /2 -7		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ		t _{KCY1} /2 -10		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ		58		479		479		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ		121		479		479		ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KSI1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ		10		10		10		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ		10		10		10		ns
SCKp↓→SOp 输出延迟时间 注 1	t _{KSO1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ			60		60		60	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ			130		130		130	ns

(7) 不同电位 (2.5V、3V) 的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出, 只限于 CSI00) (2/2)
(T_A=−40 ~ +85°C、2.7V ≤ 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		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Slp 准备时间 (对 SCKp↓) 注 2	t _{SIK1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ	23		110		110		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ	33		110		110		ns
Slp 保持时间 (对 SCKp↓) 注 2	t _{KS11}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ	10		10		10		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ	10		10		10		ns
SCKp↑→SOp 输出延迟时间 注 2	t _{KSO1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =20pF、R _b =1.4kΩ		10		10		10	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =20pF、R _b =2.7kΩ		10		10		10	ns

注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。
2. 这是 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 的情况。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 Slp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

备注 1. R_b[Ω]: 通信线 (SCKp、SOp) 的上拉电阻值 C_b[F]: 通信线 (SCKp、SOp) 的负载电容值 V_b[V]: 通信线的电压
2. p: CSI 号 (p=00) m: 单元号 (m=0) n: 通道号 (n=0) g: PIM、POM 号 (g=1)
3. f_{MCK}: 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))
4. 此值只对应不使用 CSI00 的外围 I/O 重定向功能的情况。

(8) 不同电位 (1.8V、2.5V、3V) 的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出) (1/3)

(T_A=-40 ~ +85°C, 1.8V ≤ 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		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp 周期 时间	t _{KCY1}	t _{KCY1} ≥ 4/f _{CLK}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	300		1150		1150		ns
			2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	500		1150		1150		ns
			1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注、 C _b =30pF、R _b =5.5kΩ	1150		1150		1150		ns
SCKp 高电平 宽度	t _{KH1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	t _{KCY1} /2 -75		t _{KCY1} /2 -75		t _{KCY1} /2 -75		ns	
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	t _{KCY1} /2 -170		t _{KCY1} /2 -170		t _{KCY1} /2 -170		ns	
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注、 C _b =30pF、R _b =5.5kΩ	t _{KCY1} /2 -458		t _{KCY1} /2 -458		t _{KCY1} /2 -458		ns	
SCKp 低电平 宽度	t _{KL1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	t _{KCY1} /2 -12		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns	
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	t _{KCY1} /2 -18		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns	
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注、 C _b =30pF、R _b =5.5kΩ	t _{KCY1} /2 -50		t _{KCY1} /2 -50		t _{KCY1} /2 -50		ns	

注 必须在 V_{DD} ≥ V_b 的情况下使用。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 SIp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

(8) 不同电位 (1.8V、2.5V、3V) 的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出) (2/3)

(T_A=−40 ~ +85°C, 1.8V ≤ 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		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Slp 准备时间 (对 SCKp↑) 注 1	t _{SIK1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	81		479		479		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	177		479		479		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ	479		479		479		ns
Slp 保持时间 (对 SCKp↑) 注 1	t _{KS11}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	19		19		19		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	19		19		19		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ	19		19		19		ns
SCKp↓→SOp 输出延迟时间 注 1	t _{KSO1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ		100		100		100	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ		195		195		195	ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ		483		483		483	ns

注 1. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。

2. 必须在 V_{DD} ≥ V_b 的情况下使用。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 Slp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

(8) 不同电位 (1.8V、2.5V、3V) 的通信 (CSI 模式) (主控模式, SCKp..... 内部时钟输出) (3/3)

(T_A=-40 ~ +85°C, 1.8V ≤ V_{DD} ≤ 5.5V, V_{SS}=0V)

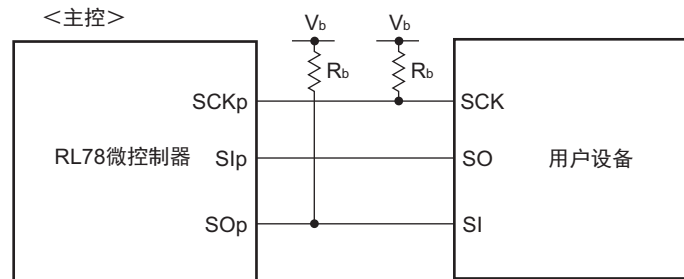
项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Slp 准备时间 (对 SCKp↓) 注 1	t _{SIK1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	44		110		110		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	44		110		110		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ	110		110		110		ns
Slp 保持时间 (对 SCKp↓) 注 1	t _{KS11}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ	19		19		19		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ	19		19		19		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ	19		19		19		ns
SCKp↑→SOp 输出延迟时间 注 1	t _{KSO1}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、R _b =1.4kΩ		25		25		25	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、R _b =2.7kΩ		25		25		25	ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注 2、 C _b =30pF、R _b =5.5kΩ		25		25		25	ns

注 1. 这是 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 的情况。

2. 必须在 V_{DD} ≥ V_b 的情况下使用。

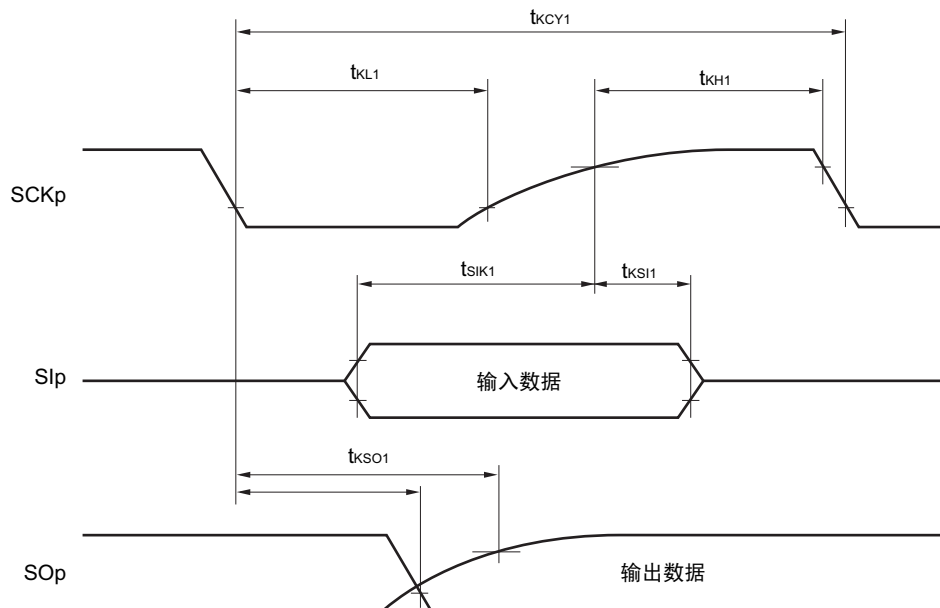
注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg), 将 Slp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚和 SCKp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

CSI 模式的连接图 (不同电位的通信)

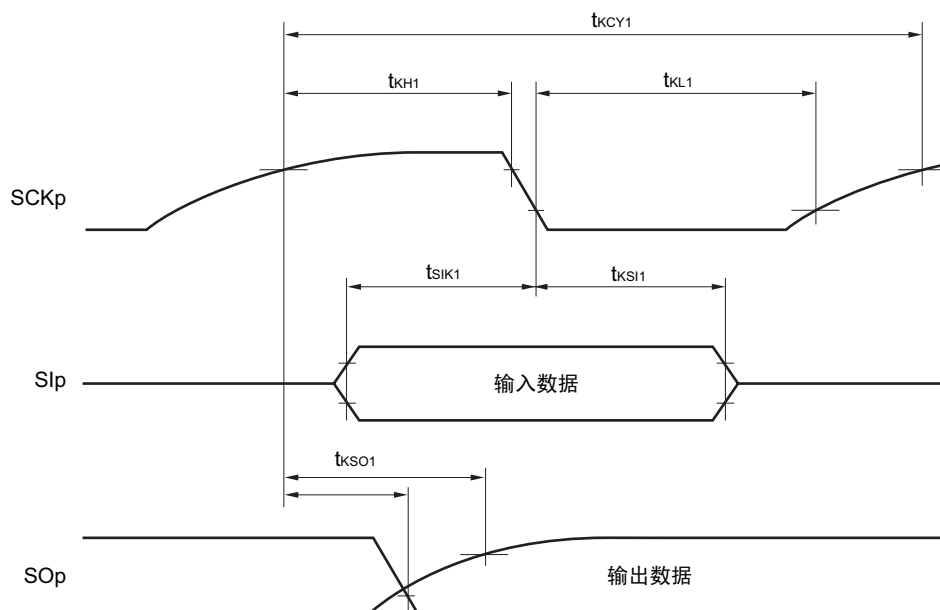


- 备注 1. $R_b[\Omega]$: 通信线 (SCKp、SOp) 的上拉电阻值 $C_b[F]$: 通信线 (SCKp、SOp) 的负载电容值 $V_b[V]$: 通信线的电压
2. p: CSI 号 (p=00、20) m: 单元号、n: 通道号 (mn=00、01、02、10、12、13)
g: PIM、POM 号 (g=0、1、5)
3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。m: 单元号、n: 通道号 (mn=00))
4. CSI11 不能进行不同电位的通信。要进行不同电位的通信时, 必须使用其他 CSI。

CSI模式的串行传送时序：主控模式（不同电位的通信）
 （DAPmn=0、CKPmn=0或者DAPmn=1、CKPmn=1的情况）



CSI模式的串行传送时序：主控模式（不同电位的通信）
 （DAPmn=0、CKPmn=1或者DAPmn=1、CKPmn=0的情况）



备注 1. p: CSI号 (p=00、20) m: 单元号、n: 通道号 (mn=00、01、02、10、12、13)
 g: PIM、POM号 (g=0、1、5)

2. CSI11 不能进行不同电位的通信。要进行不同电位的通信时，必须使用其他 CSI。

(9) 不同电位 (1.8V、2.5V、3V) 的通信 (CSI 模式) (从属模式, SCKp..... 外部时钟输入)

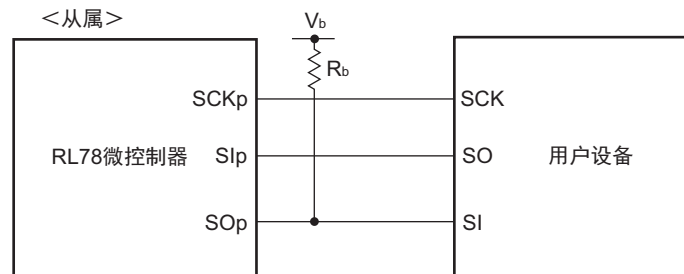
(T_A=-40 ~ +85°C, 1.8V ≤ V_{DD} ≤ 5.5V, V_{SS}=0V)

项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位			
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.				
SCKp 周期时间注 1	t _{KCY2}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V	24MHz < f _{MCK}	14/f _{MCK}	—	—	—	—	ns			
			20MHz < f _{MCK} ≤ 24MHz	12/f _{MCK}	—	—	—	—	ns			
			8MHz < f _{MCK} ≤ 20MHz	10/f _{MCK}	—	—	—	—	ns			
			4MHz < f _{MCK} ≤ 8MHz	8/f _{MCK}	16/f _{MCK}	—	—	—	ns			
			f _{MCK} ≤ 4MHz	6/f _{MCK}	10/f _{MCK}	10/f _{MCK}	10/f _{MCK}	—	ns			
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V	24MHz < f _{MCK}	20/f _{MCK}	—	—	—	—	ns			
			20MHz < f _{MCK} ≤ 24MHz	16/f _{MCK}	—	—	—	—	ns			
			16MHz < f _{MCK} ≤ 20MHz	14/f _{MCK}	—	—	—	—	ns			
			8MHz < f _{MCK} ≤ 16MHz	12/f _{MCK}	—	—	—	—	ns			
			4MHz < f _{MCK} ≤ 8MHz	8/f _{MCK}	16/f _{MCK}	—	—	—	ns			
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 2	24MHz < f _{MCK}	48/f _{MCK}	—	—	—	—	ns			
			20MHz < f _{MCK} ≤ 24MHz	36/f _{MCK}	—	—	—	—	ns			
			16MHz < f _{MCK} ≤ 20MHz	32/f _{MCK}	—	—	—	—	ns			
			8MHz < f _{MCK} ≤ 16MHz	26/f _{MCK}	—	—	—	—	ns			
			4MHz < f _{MCK} ≤ 8MHz	16/f _{MCK}	16/f _{MCK}	—	—	—	ns			
SCKp 高低电平宽度	t _{KH2} 、 t _{KL2}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V	t _{KCY2} /2	-12	t _{KCY2} /2	-50	t _{KCY2} /2	-50	ns			
			2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V	t _{KCY2} /2	-18	t _{KCY2} /2	-50	t _{KCY2} /2	-50	ns		
			1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 2	t _{KCY2} /2	-50	t _{KCY2} /2	-50	t _{KCY2} /2	-50	ns		
		Slp 准备时间 (对 SCKp↑) 注 3	t _{SIK2}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V	1/f _{MCK}	+20	1/f _{MCK}	+30	1/f _{MCK}	+30	ns	
					2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V	1/f _{MCK}	+20	1/f _{MCK}	+30	1/f _{MCK}	+30	ns
					1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 2	1/f _{MCK}	+30	1/f _{MCK}	+30	1/f _{MCK}	+30	ns
		Slp 保持时间 (对 SCKp↑) 注 4	t _{KSI2}		1/f _{MCK}	+31	1/f _{MCK}	+31	1/f _{MCK}	+31	ns	
		SCKp↓→SOp 输出延迟时间 注 5	t _{KSO2}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =30pF、 R _b =1.4kΩ	2/f _{MCK}	+120	2/f _{MCK}	+573	2/f _{MCK}	+573	ns	
				2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =30pF、 R _b =2.7kΩ	2/f _{MCK}	+214	2/f _{MCK}	+573	2/f _{MCK}	+573	ns	
				1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 2、 C _b =30pF、 R _b =5.5kΩ	2/f _{MCK}	+573	2/f _{MCK}	+573	2/f _{MCK}	+573	ns	

- 注
1. SNOOZE 模式中的传送速率为 MAX. 1Mbps。
 2. 必须在 $V_{DD} \geq V_b$ 的情况下使用。
 3. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↓”。
 4. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↓”。
 5. 这是 DAPmn=0、CKPmn=0 或者 DAPmn=1、CKPmn=1 的情况。在 DAPmn=0、CKPmn=1 或者 DAPmn=1、CKPmn=0 时，为“对 SCKp↑”。

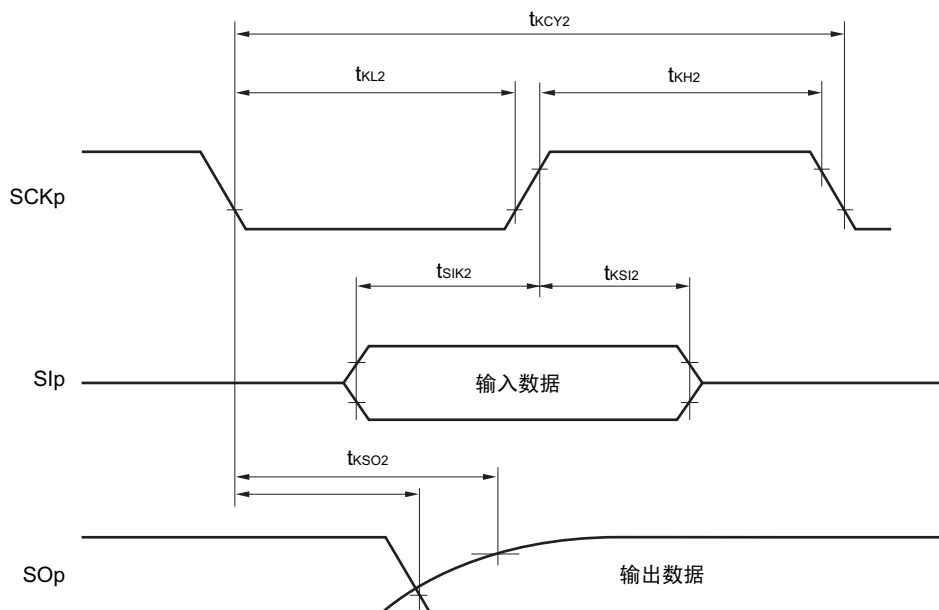
注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 Slp 引脚和 SCKp 引脚选择为 TTL 输入缓冲器并且将 SOp 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。 V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

CSI 模式的连接图 (不同电位的通信)

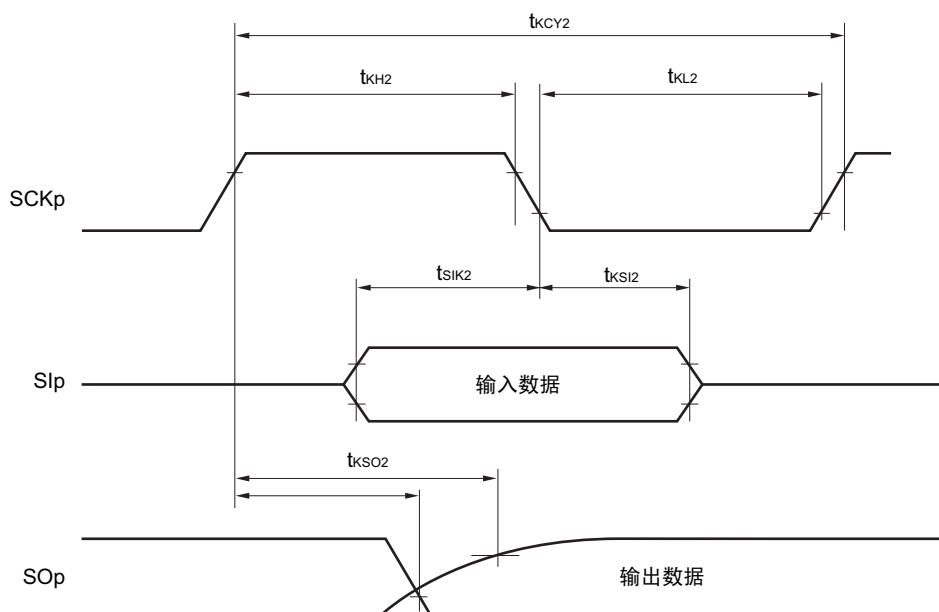


- 备注
1. $R_b[\Omega]$: 通信线 (SOp) 的上拉电阻值 $C_b[F]$: 通信线 (SOp) 的负载电容值 $V_b[V]$: 通信线的电压
 2. p: CSI 号 (p=00、20) m: 单元号、n: 通道号 (mn=00、01、02、10、12、13)
g: PIM、POM 号 (g=0、1、5)
 3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
m: 单元号、n: 通道号 (mn=00、01、02、10、12、13))
 4. CSI11 不能进行不同电位的通信。要进行不同电位的通信时，必须使用其他 CSI。

CSI模式的串行传送时序：从属模式（不同电位的通信）
 （DAPmn=0、CKPmn=0或者DAPmn=1、CKPmn=1的情况）



CSI模式的串行传送时序：从属模式（不同电位的通信）
 （DAPmn=0、CKPmn=1或者DAPmn=1、CKPmn=0的情况）



- 备注 1. p: CSI 号 (p=00、20) m: 单元号、n: 通道号 (mn=00、01、02、10、12、13)
 g: PIM、POM 号 (g=0、1、5)
 2. CSI11 不能进行不同电位的通信。要进行不同电位的通信时，必须使用其他 CSI。

(10) 不同电位 (1.8V、2.5V、3V) 的通信 (简易 I²C 模式) (1/2)(T_A=-40 ~ +85°C, 1.8V ≤ V_{DD} ≤ 5.5V, V_{SS}=0V)

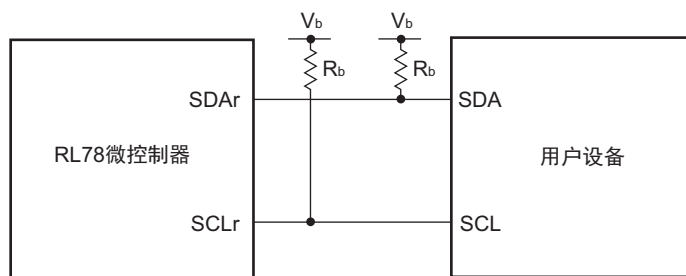
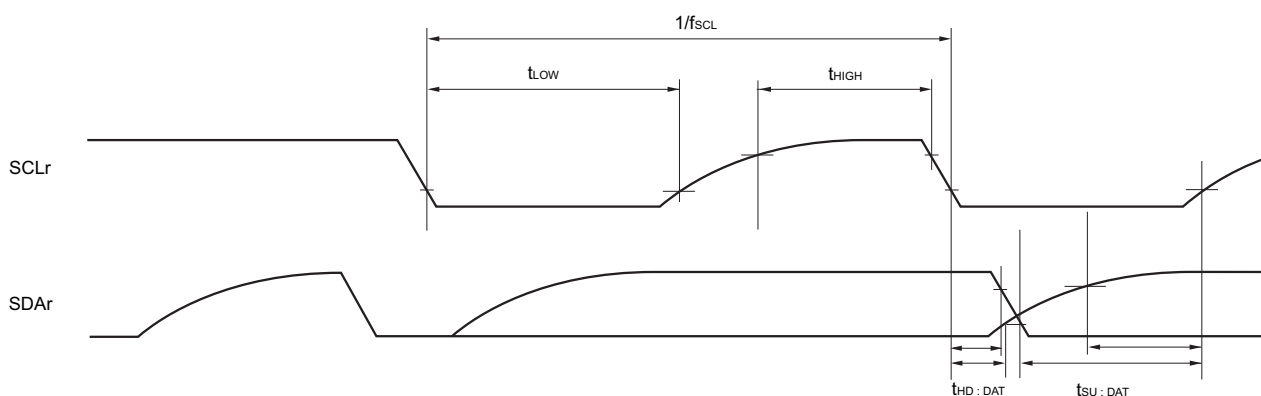
项目	符号	条件	HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr 时钟频率	f _{SCL}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ		1000 注1		300 注1		300 注1	kHz
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ		1000 注1		300 注1		300 注1	kHz
		4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ		400 注1		300 注1		300 注1	kHz
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ		400 注1		300 注1		300 注1	kHz
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注2、 C _b =100pF、R _b =5.5kΩ		300 注1		300 注1		300 注1	kHz
SCLr="L" 的 保持时间	t _{LOW}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	475		1550		1550		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	475		1550		1550		ns
		4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	1150		1550		1550		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	1150		1550		1550		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注2、 C _b =100pF、R _b =5.5kΩ	1550		1550		1550		ns
SCLr="H" 的 保持时间	t _{HIGH}	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	245		610		610		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	200		610		610		ns
		4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	675		610		610		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	600		610		610		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V 注2、 C _b =100pF、R _b =5.5kΩ	610		610		610		ns

(10) 不同电位 (1.8V、2.5V、3V) 的通信 (简易 I²C 模式) (2/2)(T_A=-40 ~ +85°C、1.8V ≤ 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		LV (低电压主) 模式		单位
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
数据准备时间 (接收时)	t _{SU} : DAT	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	1/f _{MCK} +135注 ³		1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	1/f _{MCK} +135注 ³		1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		ns
		4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 ² 、 C _b =100pF、R _b =5.5kΩ	1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		1/f _{MCK} +190注 ³		ns
数据保持时间 (发送时)	t _{HD} : DAT	4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =50pF、R _b =2.7kΩ	0	305	0	305	0	305	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =50pF、R _b =2.7kΩ	0	305	0	305	0	305	ns
		4.0V ≤ V _{DD} ≤ 5.5V、 2.7V ≤ V _b ≤ 4.0V、 C _b =100pF、R _b =2.8kΩ	0	355	0	355	0	355	ns
		2.7V ≤ V _{DD} < 4.0V、 2.3V ≤ V _b ≤ 2.7V、 C _b =100pF、R _b =2.7kΩ	0	355	0	355	0	355	ns
		1.8V ≤ V _{DD} < 3.3V、 1.6V ≤ V _b ≤ 2.0V注 ² 、 C _b =100pF、R _b =5.5kΩ	0	405	0	405	0	405	ns

- 注 1. 并且必须至少设定为 f_{MCK}/4。
 2. 必须在 V_{DD} ≥ V_b 的情况下使用。
 3. f_{MCK} 的设定值不能超过 SCLr="L" 和 SCLr="H" 的保持时间。

注意 通过端口输入模式寄存器 g (PIMg) 和端口输出模式寄存器 g (POMg)，将 SDAr 引脚选择为 TTL 输入缓冲器和 N 沟道漏极开路输出 (V_{DD} 耐压) 模式，并且将 SCLr 引脚选择为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。V_{IH} 和 V_{IL} 请参照选择为 TTL 输入缓冲器时的 DC 特性。

简易 I²C 模式连接图 (不同电位的通信)简易 I²C 模式串行传送时序 (不同电位的通信)

- 备注 1. $R_b[\Omega]$: 通信线 (SDAr、SCLr) 的上拉电阻值 $C_b[F]$: 通信线 (SDAr、SCLr) 的负载电容值 $V_b[V]$: 通信线的电压
2. r: IIC 号 (r=00、20) g: PIM、POM 号 (g=0、1、5)
3. f_{MCK} : 串行阵列单元的运行时钟频率
(这是串行模式寄存器 mn (SMRmn) 的 CKSmn 位设定的运行时钟。
m: 单元号、n: 通道号 (mn=00、01、02、10、12、13))

27.5.2 串行接口 IICA

(1) I²C 标准模式(T_A=-40 ~ +85°C, 1.6V ≤ 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		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 时钟频率	f _{SCL}	标准模式: f _{CLK} ≥ 1MHz	2.7V ≤ V _{DD} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.8V ≤ V _{DD} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.7V ≤ V _{DD} ≤ 5.5V	0	100	0	100	0	100	kHz
			1.6V ≤ V _{DD} ≤ 5.5V	—		0	100	0	100	kHz
重新开始条件的准备时间	t _{SU: STA}	2.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.7		4.7		μs
保持时间注 1	t _{HD: STA}	2.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.0		4.0		μs
SCLA0="L" 的保持时间	t _{LOW}	2.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.7		4.7		μs
SCLA0="H" 的保持时间	t _{HIGH}	2.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.0		4.0		μs
数据准备时间 (接收时)	t _{SU: DAT}	2.7V ≤ V _{DD} ≤ 5.5V		250		250		250		ns
		1.8V ≤ V _{DD} ≤ 5.5V		250		250		250		ns
		1.7V ≤ V _{DD} ≤ 5.5V		250		250		250		ns
		1.6V ≤ V _{DD} ≤ 5.5V		—		250		250		ns
数据保持时间 (发送时) 注 2	t _{HD: DAT}	2.7V ≤ V _{DD} ≤ 5.5V		0	3.45	0	3.45	0	3.45	μs
		1.8V ≤ V _{DD} ≤ 5.5V		0	3.45	0	3.45	0	3.45	μs
		1.7V ≤ V _{DD} ≤ 5.5V		0	3.45	0	3.45	0	3.45	μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		0	3.45	0	3.45	μs
停止条件的准备时间	t _{SU: STO}	2.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.0		4.0		4.0		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.0		4.0		μs
总线空闲时间	t _{BUF}	2.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.8V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.7V ≤ V _{DD} ≤ 5.5V		4.7		4.7		4.7		μs
		1.6V ≤ V _{DD} ≤ 5.5V		—		4.7		4.7		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_{HD}: DAT 的最大值 (MAX.) 是通常传送时的数值, 在进行应答 (ACK) 时需要等待。

注意 即使在外围 I/O 重定向寄存器 (PIOR) 的 bit2 (PIOR2) 为 "1" 时, 也能适用上述值。但是, 引脚特性 (I_{OH1}、I_{OL1}、V_{OH1}、V_{OL1}) 必须满足重定向目标的值。

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:
 标准模式: C_b=400pF、R_b=2.7kΩ

(2) I²C 快速模式

(T_A=-40 ~ +85°C、1.6V ≤ 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		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 时钟频率	f _{SCL}	快速模式: f _{CLK} ≥ 3.5MHz	2.7V ≤ V _{DD} ≤ 5.5V	0	400	0	400	0	400	kHz
			1.8V ≤ V _{DD} ≤ 5.5V	0	400	0	400	0	400	kHz
重新开始条件的准备时间	t _{SU} : STA	2.7V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
保持时间注 1	t _{HD} : STA	2.7V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
SCLA0="L" 的保持时间	t _{LOW}	2.7V ≤ V _{DD} ≤ 5.5V		1.3		1.3		1.3		μs
		1.8V ≤ V _{DD} ≤ 5.5V		1.3		1.3		1.3		μs
SCLA0="H" 的保持时间	t _{HIGH}	2.7V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
数据准备时间 (接收时)	t _{SU} : DAT	2.7V ≤ V _{DD} ≤ 5.5V		100		100		100		ns
		1.8V ≤ V _{DD} ≤ 5.5V		100		100		100		ns
数据保持时间 (发送时) 注 2	t _{HD} : DAT	2.7V ≤ V _{DD} ≤ 5.5V		0	0.9	0	0.9	0	0.9	μs
		1.8V ≤ V _{DD} ≤ 5.5V		0	0.9	0	0.9	0	0.9	μs
停止条件的准备时间	t _{SU} : STO	2.7V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
		1.8V ≤ V _{DD} ≤ 5.5V		0.6		0.6		0.6		μs
总线空闲时间	t _{BUF}	2.7V ≤ V _{DD} ≤ 5.5V		1.3		1.3		1.3		μs
		1.8V ≤ V _{DD} ≤ 5.5V		1.3		1.3		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_{HD}: DAT 的最大值 (MAX.) 是通常传送时的数值, 在进行应答 (ACK) 时需要等待。

注意 即使在外围 I/O 重定向寄存器 (PIOR) 的 bit2 (PIOR2) 为 "1" 时, 也能适用上述值。但是, 引脚特性 (I_{OH1}、I_{OL1}、V_{OH1}、V_{OL1}) 必须满足重定向目标的值。

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:
 快速模式: C_b=320pF、R_b=1.1kΩ

(3) I²C 增强型快速模式($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.6\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$)

项目	符号	条件		HS (高速主) 模式		LS (低速主) 模式		LV (低电压主) 模式		单位
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 时钟频率	f_{SCL}	增强型快速模式: $f_{\text{CLK}} \geq 10\text{MHz}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$	0	1000	—	—	—	—	kHz
重新开始条件的准备时间	$t_{\text{SU: STA}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.26		—	—	—	—	μs
保持时间注 1	$t_{\text{HD: STA}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.26		—	—	—	—	μs
SCLA0="L" 的保持时间	t_{LOW}	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.5		—	—	—	—	μs
SCLA0="H" 的保持时间	t_{HIGH}	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.26		—	—	—	—	μs
数据准备时间 (接收时)	$t_{\text{SU: DAT}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		50		—	—	—	—	ns
数据保持时间 (发送时) 注 2	$t_{\text{HD: DAT}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0	0.45	—	—	—	—	μs
停止条件的准备时间	$t_{\text{SU: STO}}$	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.26		—	—	—	—	μs
总线空闲时间	t_{BUF}	$2.7\text{V} \leq V_{DD} \leq 5.5\text{V}$		0.5		—	—	—	—	μs

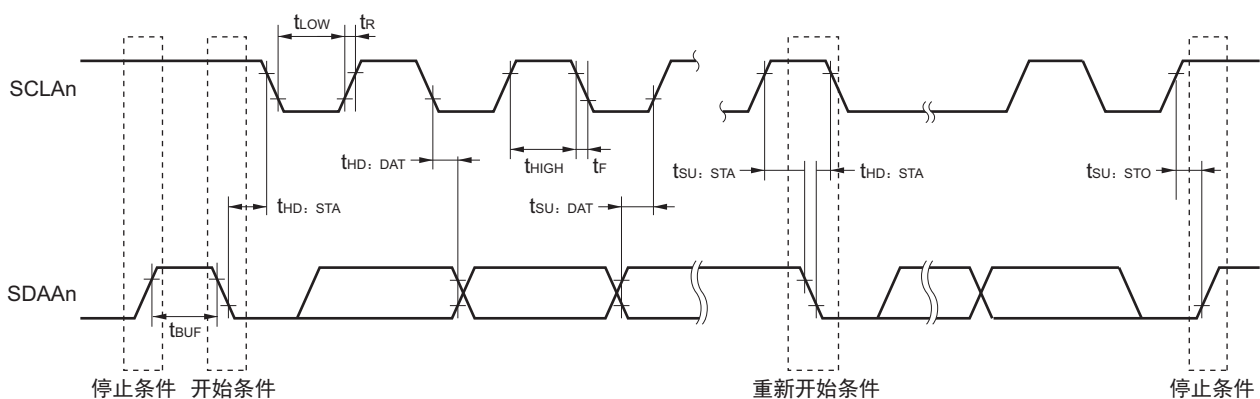
注 1. 在开始条件和重新开始条件的情况下, 在此期间之后生成第一个时钟脉冲。

2. $t_{\text{HD: DAT}}$ 的最大值 (MAX.) 是通常传送时的数值, 在进行应答 (ACK) 时需要等待。

注意 即使在外围 I/O 重定向寄存器 (PIOR) 的 bit2 (PIOR2) 为 "1" 时, 也能适用上述值。但是, 引脚特性 ($I_{\text{OH}1}$ 、 $I_{\text{OL}1}$ 、 $V_{\text{OH}1}$ 、 $V_{\text{OL}1}$) 必须满足重定向目标的值。

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:

增强型快速模式: $C_b=120\text{pF}$ 、 $R_b=1.1\text{k}\Omega$

I²C 串行传送时序

备注 n=0

27.6 模拟特性

27.6.1 A/D 转换器特性

A/D 转换器特性的区分

输入通道	基准电压	基准电压 (+)=AV _{REFP} 基准电压 (-)=AV _{REFM}	基准电压 (+)=V _{DD} 基准电压 (-)=V _{SS}	基准电压 (+)=V _{BGR} 基准电压 (-)=AV _{REFM}
ANI0 ~ ANI3		参照 27.6.1(1)。	参照 27.6.1(3)。	参照 27.6.1(4)。
ANI16 ~ ANI19		参照 27.6.1(2)。		
内部基准电压		参照 27.6.1(1)。		

(1) 选择基准电压(+)=AV_{REFP}/ANI0 (ADREFP1=0, ADREFP0=1)、基准电压(-)=AV_{REFM}/ANI1 (ADREFM=1) 的情况, 转换对象: ANI2、ANI3、内部基准电压

(T_A=-40~+85°C、1.6V≤AV_{REFP}≤V_{DD}≤5.5V、V_{SS}=0V、基准电压(+)=AV_{REFP}、基准电压(-)=AV_{REFM}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
分辨率	RES		8		10	bit	
综合误差注 1	AINL	10 位分辨率 AV _{REFP} =V _{DD} 注 3	1.8V ≤ AV _{REFP} ≤ 5.5V		1.2	±3.5	LSB
			1.6V ≤ AV _{REFP} ≤ 5.5V注 4		1.2	±7.0	LSB
转换时间	t _{CONV}	10 位分辨率 转换对象: ANI2、ANI3	3.6V ≤ V _{DD} ≤ 5.5V	2.125		39	μs
			2.7V ≤ V _{DD} ≤ 5.5V	3.1875		39	μs
			1.8V ≤ V _{DD} ≤ 5.5V	17		39	μs
			1.6V ≤ V _{DD} ≤ 5.5V	57		95	μs
		10 位分辨率 转换对象: 内部基准电压 (HS (高速主) 模式)	3.6V ≤ V _{DD} ≤ 5.5V	2.375		39	μs
			2.7V ≤ V _{DD} ≤ 5.5V	3.5625		39	μs
零刻度误差注 1、2	E _{ZS}	10 位分辨率 AV _{REFP} =V _{DD} 注 3	1.8V ≤ AV _{REFP} ≤ 5.5V			±0.25	%FSR
			1.6V ≤ AV _{REFP} ≤ 5.5V注 4			±0.50	%FSR
满刻度误差注 1、2	E _{FS}	10 位分辨率 AV _{REFP} =V _{DD} 注 3	1.8V ≤ AV _{REFP} ≤ 5.5V			±0.25	%FSR
			1.6V ≤ AV _{REFP} ≤ 5.5V注 4			±0.50	%FSR
积分线性误差注 1	ILE	10 位分辨率 AV _{REFP} =V _{DD} 注 3	1.8V ≤ AV _{REFP} ≤ 5.5V			±2.5	LSB
			1.6V ≤ AV _{REFP} ≤ 5.5V注 4			±5.0	LSB
微分线性误差注 1	DLE	10 位分辨率 AV _{REFP} =V _{DD} 注 3	1.8V ≤ AV _{REFP} ≤ 5.5V			±1.5	LSB
			1.6V ≤ AV _{REFP} ≤ 5.5V注 4			±2.0	LSB
模拟输入电压	V _{AIN}	ANI2、ANI3	0		AV _{REFP}	V	
		内部基准电压 (2.4V ≤ V _{DD} ≤ 5.5V、HS (高速主) 模式)			V _{BGR} 注 5	V	

注 1. 不包含量化误差 (±1/2 LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

3. 当 AV_{REFP} < V_{DD} 时, MAX. 值如下:

综合误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±1.0LSB。

零刻度误差 / 满刻度误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±0.05%FSR。

积分线性误差 / 微分线性误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±0.5LSB。

4. 这是将转换时间设定为 MIN.57μs、MAX.95μs 时的值。

5. 请参照“27.6.2 内部基准电压的特性”。

(2) 选择基准电压(+)=AV_{REFP}/ANI0 (ADREFP1=0, ADREFP0=1)、基准电压(-)=AV_{REFM}/ANI1 (ADREFM=1) 的情况, 转换对象: ANI16 ~ ANI19

(T_A=-40~+85°C、1.6V≤AV_{REFP}≤V_{DD}≤5.5V、V_{SS}=0V、基准电压(+)=AV_{REFP}、基准电压(-)=AV_{REFM}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
分辨率	RES		8		10	bit	
综合误差注 1	AINL	10 位分辨率	1.8V ≤ AV _{REFP} ≤ 5.5V		1.2	±5.0	LSB
		V _{DD} =AV _{REFP} =V _{DD} 注 3、4	1.6V ≤ AV _{REFP} ≤ 5.5V注 5		1.2	±8.5	LSB
转换时间	t _{CONV}	10 位分辨率 转换对象: ANI16 ~ ANI19	3.6V ≤ V _{DD} ≤ 5.5V	2.125		39	μs
			2.7V ≤ V _{DD} ≤ 5.5V	3.1875		39	μs
			1.8V ≤ V _{DD} ≤ 5.5V	17		39	μs
			1.6V ≤ V _{DD} ≤ 5.5V	57		95	μs
零刻度误差注 1、2	E _{ZS}	10 位分辨率 V _{DD} =AV _{REFP} =V _{DD} 注 3、4	1.8V ≤ AV _{REFP} ≤ 5.5V			±0.35	%FSR
			1.6V ≤ AV _{REFP} ≤ 5.5V注 5			±0.60	%FSR
满刻度误差注 1、2	E _{FS}	10 位分辨率 V _{DD} =AV _{REFP} =V _{DD} 注 3、4	1.8V ≤ AV _{REFP} ≤ 5.5V			±0.35	%FSR
			1.6V ≤ AV _{REFP} ≤ 5.5V注 5			±0.60	%FSR
积分线性误差注 1	ILE	10 位分辨率 V _{DD} =AV _{REFP} =V _{DD} 注 3、4	1.8V ≤ AV _{REFP} ≤ 5.5V			±3.5	LSB
			1.6V ≤ AV _{REFP} ≤ 5.5V注 5			±6.0	LSB
微分线性误差注 1	DLE	10 位分辨率 V _{DD} =AV _{REFP} =V _{DD} 注 3、4	1.8V ≤ AV _{REFP} ≤ 5.5V			±2.0	LSB
			1.6V ≤ AV _{REFP} ≤ 5.5V注 5			±2.5	LSB
模拟输入电压	V _{AIN}	ANI16 ~ ANI19	0		AV _{REFP} 并且 V _{DD}	V	

注 1. 不包含量化误差 (±1/2 LSB)。

2. 用对满刻度值的比率 (%FSR) 表示。

3. 当 AV_{REFP} < V_{DD} 时, MAX. 值如下:

综合误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±1.0LSB。

零刻度误差 / 满刻度误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±0.05%FSR。

积分线性误差 / 微分线性误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±0.5LSB。

4. 当 AV_{REFP} < V_{DD} 时, MAX. 值如下:

综合误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±4.0LSB。

零刻度误差 / 满刻度误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±0.20%FSR。

积分线性误差 / 微分线性误差: AV_{REFP}=V_{DD} 的 MAX. 值必须加上 ±2.0LSB。

5. 这是将转换时间设定为 MIN.57μs、MAX.95μs 时的值。

(3) 选择基准电压 (+)=V_{DD} (ADREFP1=0、ADREFP0=0)、基准电压 (-)=V_{SS} (ADREFM=0) 的情况, 转换对象: ANI0 ~ ANI3、ANI16 ~ ANI19、内部基准电压

(T_A=-40 ~ +85°C、1.6V ≤ V_{DD} ≤ 5.5V、V_{SS}=0V、基准电压 (+)=V_{DD}、基准电压 (-)=V_{SS})

项目	符号	条件		MIN.	TYP.	MAX.	单位
分辨率	RES			8		10	bit
综合误差注 1	AINL	10 位分辨率	1.8V ≤ V _{DD} ≤ 5.5V		1.2	±7.0	LSB
			1.6V ≤ V _{DD} ≤ 5.5V 注 3		1.2	±10.5	LSB
转换时间	t _{CONV}	10 位分辨率 转换对象: ANI0 ~ ANI3、 ANI16 ~ ANI19	3.6V ≤ V _{DD} ≤ 5.5V	2.125		39	μs
			2.7V ≤ V _{DD} ≤ 5.5V	3.1875		39	μs
			1.8V ≤ V _{DD} ≤ 5.5V	17		39	μs
			1.6V ≤ V _{DD} ≤ 5.5V	57		95	μs
		10 位分辨率 转换对象: 内部基准电压 (HS (高速主) 模式)	3.6V ≤ V _{DD} ≤ 5.5V	2.375		39	μs
			2.7V ≤ V _{DD} ≤ 5.5V	3.5625		39	μs
零刻度误差注 1、2	E _{ZS}	10 位分辨率	1.8V ≤ V _{DD} ≤ 5.5V			±0.60	%FSR
			1.6V ≤ V _{DD} ≤ 5.5V 注 3			±0.85	%FSR
满刻度误差注 1、2	E _{FS}	10 位分辨率	1.8V ≤ V _{DD} ≤ 5.5V			±0.60	%FSR
			1.6V ≤ V _{DD} ≤ 5.5V 注 3			±0.85	%FSR
积分线性误差注 1	ILE	10 位分辨率	1.8V ≤ V _{DD} ≤ 5.5V			±4.0	LSB
			1.6V ≤ V _{DD} ≤ 5.5V 注 3			±6.5	LSB
微分线性误差注 1	DLE	10 位分辨率	1.8V ≤ V _{DD} ≤ 5.5V			±2.0	LSB
			1.6V ≤ V _{DD} ≤ 5.5V 注 3			±2.5	LSB
模拟输入电压	V _{AIN}	ANI0 ~ ANI3	0			V _{DD}	V
		ANI16 ~ ANI19	0			V _{DD}	V
		内部基准电压 (2.4V ≤ V _{DD} ≤ 5.5V、HS (高速主) 模式)	V _{BGR} 注 4				V

- 注 1. 不包含量化误差 (±1/2 LSB)。
 2. 用对满刻度值的比率 (%FSR) 表示。
 3. 这是将转换时间设定为 MIN.57μs、MAX.95μs 时的值。
 4. 请参照“27.6.2 内部基准电压的特性”。

- (4) 选择基准电压 (+)= 内部基准电压 (ADREFP1=1, ADREFP0=0)、基准电压 (-)= $AV_{REFM}/ANI1$ (ADREFM=1) 的情况, 转换对象: ANI0、ANI2、ANI3、ANI16 ~ ANI19

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $1.6\text{V} \leq V_{DD}$ 、 $V_{SS}=0\text{V}$ 、基准电压 (+)= V_{BGR} 注 3、基准电压 (-)= AV_{REFM} 注 4=0V、HS (高速主) 模式)

项目	符号	条件		MIN.	TYP.	MAX.	单位
分辨率	RES			8			bit
转换时间	t_{CONV}	8 位分辨率	$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$	17		39	μs
零刻度误差注 1、2	E_{ZS}	8 位分辨率	$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 0.60	%FSR
积分线性误差注 1	ILE	8 位分辨率	$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 2.0	LSB
微分线性误差注 1	DLE	8 位分辨率	$2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$			± 1.0	LSB
模拟输入电压	V_{AIN}			0		V_{BGR} 注 3	V

- 注 1. 不包含量化误差 ($\pm 1/2$ LSB)。
 2. 用对满刻度值的比率 (%FSR) 表示。
 3. 请参照“27.6.2 内部基准电压的特性”。
 4. 当基准电压 (-)= V_{SS} 时, MAX. 值如下:
 零刻度误差: 基准电压 (-)= AV_{REFM} 时的 MAX. 值必须加上 $\pm 0.35\%$ FSR。
 积分线性误差: 基准电压 (-)= AV_{REFM} 时的 MAX. 值必须加上 ± 0.5 LSB。
 微分线性误差: 基准电压 (-)= AV_{REFM} 时的 MAX. 值必须加上 ± 0.2 LSB。

27.6.2 内部基准电压的特性

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $2.4\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$ 、HS (高速主) 模式)

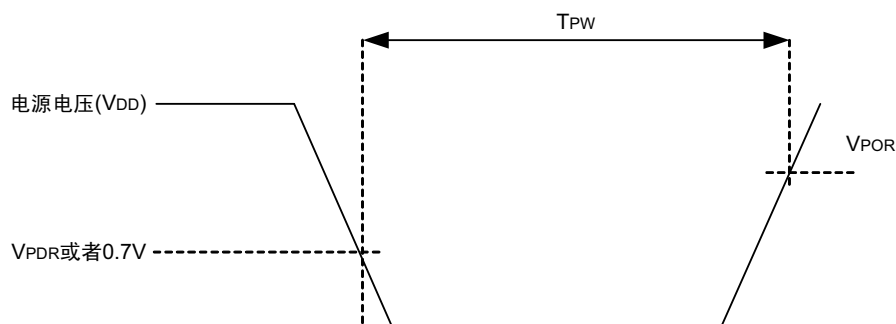
项目	符号	条件	MIN.	TYP.	MAX.	单位
内部基准电压	V_{BGR}	ADS 寄存器 =81H	1.38	1.45	1.5	V
运行稳定等待时间	t_{AMP}		5			μs

27.6.3 POR 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_{POR}	电源电压上升时	1.47	1.51	1.55	V
	V_{PDR}	电源电压下降时	1.46	1.50	1.54	V
最小脉宽注	T_{PW}		300			μs

注 这是在 V_{DD} 低于 V_{PDR} 时 POR 复位所需的时间。另外，在转移到 STOP 模式或者通过设定时钟运行状态控制寄存器 (CSC) 的 bit0 (HIOSTOP) 和 bit7 (MSTOP) 停止主系统时钟 (f_{MAIN}) 的振荡时，是从 V_{DD} 低于 0.7V 到回升超过 V_{POR} 为止的 POR 复位所需的时间。



27.6.4 LVD 电路特性

(1) 复位模式和中断模式的 LVD 检测电压

(T_A=-40 ~ +85°C、V_{PDR} ≤ V_{DD} ≤ 5.5V、V_{SS}=0V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
检测电压 电源电压电平	V _{LVD0}	电源电压上升时	3.98	4.06	4.14	V
		电源电压下降时	3.90	3.98	4.06	V
	V _{LVD1}	电源电压上升时	3.68	3.75	3.82	V
		电源电压下降时	3.60	3.67	3.74	V
	V _{LVD2}	电源电压上升时	3.07	3.13	3.19	V
		电源电压下降时	3.00	3.06	3.12	V
	V _{LVD3}	电源电压上升时	2.96	3.02	3.08	V
		电源电压下降时	2.90	2.96	3.02	V
	V _{LVD4}	电源电压上升时	2.86	2.92	2.97	V
		电源电压下降时	2.80	2.86	2.91	V
	V _{LVD5}	电源电压上升时	2.76	2.81	2.87	V
		电源电压下降时	2.70	2.75	2.81	V
	V _{LVD6}	电源电压上升时	2.66	2.71	2.76	V
		电源电压下降时	2.60	2.65	2.70	V
	V _{LVD7}	电源电压上升时	2.56	2.61	2.66	V
		电源电压下降时	2.50	2.55	2.60	V
	V _{LVD8}	电源电压上升时	2.45	2.50	2.55	V
		电源电压下降时	2.40	2.45	2.50	V
	V _{LVD9}	电源电压上升时	2.05	2.09	2.13	V
		电源电压下降时	2.00	2.04	2.08	V
	V _{LVD10}	电源电压上升时	1.94	1.98	2.02	V
		电源电压下降时	1.90	1.94	1.98	V
	V _{LVD11}	电源电压上升时	1.84	1.88	1.91	V
		电源电压下降时	1.80	1.84	1.87	V
	V _{LVD12}	电源电压上升时	1.74	1.77	1.81	V
		电源电压下降时	1.70	1.73	1.77	V
V _{LVD13}	电源电压上升时	1.64	1.67	1.70	V	
	电源电压下降时	1.60	1.63	1.66	V	
最小脉宽	t _{LW}		300			μs
检测延迟					300	μs

(2) 中断&复位模式的 LVD 检测电压

($T_A=-40 \sim +85^\circ\text{C}$, $V_{PDR} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位	
中断 & 复位 模式	V_{LVDA0}	V_{POC2} , V_{POC1} , $V_{POC0}=0, 0, 0$, 下降复位电压	1.60	1.63	1.66	V	
	V_{LVDA1}	$LVIS1$, $LVIS0=1, 0$	上升复位解除电压	1.74	1.77	1.81	V
			下降中断电压	1.70	1.73	1.77	V
	V_{LVDA2}	$LVIS1$, $LVIS0=0, 1$	上升复位解除电压	1.84	1.88	1.91	V
			下降中断电压	1.80	1.84	1.87	V
	V_{LVDA3}	$LVIS1$, $LVIS0=0, 0$	上升复位解除电压	2.86	2.92	2.97	V
			下降中断电压	2.80	2.86	2.91	V
	V_{LVDB0}	V_{POC2} , V_{POC1} , $V_{POC0}=0, 0, 1$, 下降复位电压	1.80	1.84	1.87	V	
	V_{LVDB1}	$LVIS1$, $LVIS0=1, 0$	上升复位解除电压	1.94	1.98	2.02	V
			下降中断电压	1.90	1.94	1.98	V
	V_{LVDB2}	$LVIS1$, $LVIS0=0, 1$	上升复位解除电压	2.05	2.09	2.13	V
			下降中断电压	2.00	2.04	2.08	V
	V_{LVDB3}	$LVIS1$, $LVIS0=0, 0$	上升复位解除电压	3.07	3.13	3.19	V
			下降中断电压	3.00	3.06	3.12	V
	V_{LVDC0}	V_{POC2} , V_{POC1} , $V_{POC0}=0, 1, 0$, 下降复位电压	2.40	2.45	2.50	V	
	V_{LVDC1}	$LVIS1$, $LVIS0=1, 0$	上升复位解除电压	2.56	2.61	2.66	V
			下降中断电压	2.50	2.55	2.60	V
	V_{LVDC2}	$LVIS1$, $LVIS0=0, 1$	上升复位解除电压	2.66	2.71	2.76	V
			下降中断电压	2.60	2.65	2.70	V
	V_{LVDC3}	$LVIS1$, $LVIS0=0, 0$	上升复位解除电压	3.68	3.75	3.82	V
			下降中断电压	3.60	3.67	3.74	V
	V_{LVDD0}	V_{POC2} , V_{POC1} , $V_{POC0}=0, 1, 1$, 下降复位电压	2.70	2.75	2.81	V	
	V_{LVDD1}	$LVIS1$, $LVIS0=1, 0$	上升复位解除电压	2.86	2.92	2.97	V
			下降中断电压	2.80	2.86	2.91	V
V_{LVDD2}	$LVIS1$, $LVIS0=0, 1$	上升复位解除电压	2.96	3.02	3.08	V	
		下降中断电压	2.90	2.96	3.02	V	
V_{LVDD3}	$LVIS1$, $LVIS0=0, 0$	上升复位解除电压	3.98	4.06	4.14	V	
		下降中断电压	3.90	3.98	4.06	V	

27.6.5 电源电压的上升斜率特性

($T_A=-40 \sim +85^\circ\text{C}$, $V_{SS}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
电源电压的上升斜率	S_{VDD}				54	V/ms

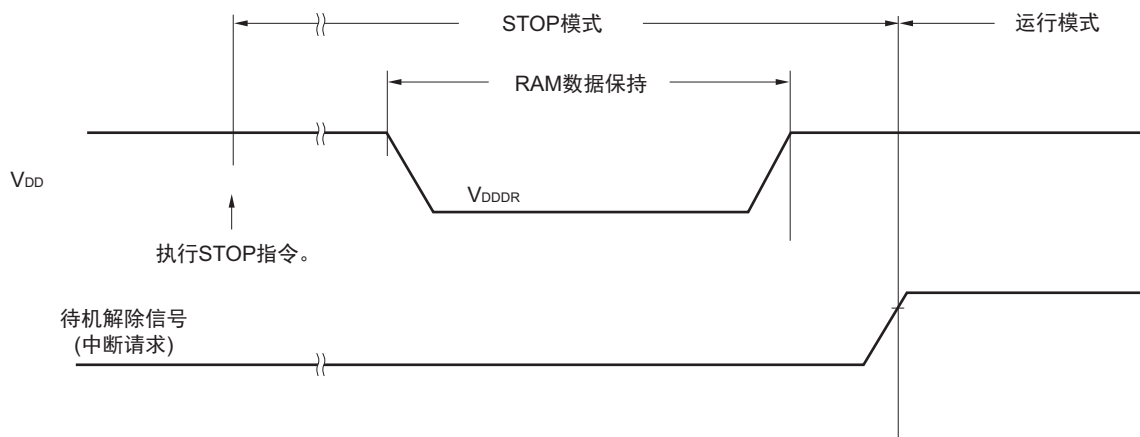
注意 必须在 V_{DD} 达到“27.4 AC 特性”所示的工作电压范围前，通过 LVD 电路或者外部复位保持内部复位状态。

27.7 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}		1.46 注		5.5	V

注 取决于 POR 检测电压。当电压下降时，保持 RAM 的数据，直到发生 POR 复位为止。但是，当发生 POR 复位时，不保持 RAM 的数据。



27.8 闪存编程特性

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
系统时钟频率	f_{CLK}	$1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$	1		24	MHz
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
数据闪存的改写次数注 1、2、3		保持年数：1 年 $T_A=25^{\circ}\text{C}$		1000000		
		保持年数：5 年 $T_A=85^{\circ}\text{C}$	100000			
		保持年数：20 年 $T_A=85^{\circ}\text{C}$	10000			

- 注 1. 1 次改写是指 1 次擦除 + 擦除后的 1 次编程，保持年数是指从 1 次改写到下一次改写为止的期间。
 2. 这是使用闪存编程器和本公司提供的库程序的情况。
 3. 此特性表示闪存的特性，是本公司的可靠性试验的结果。

27.9 专用闪存编程器通信 (UART)

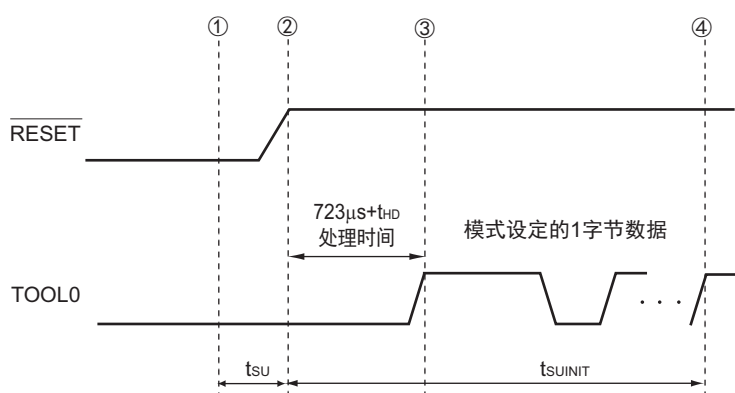
($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.8\text{V} \leq V_{DD} \leq 5.5\text{V}$ 、 $V_{SS}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率		串行编程时	115200		1000000	bps

27.10 闪存编程模式的转移时序

($T_A=-40 \sim +85^{\circ}\text{C}$ 、 $1.8\text{V} \leq V_{\text{DD}} \leq 5.5\text{V}$ 、 $V_{\text{SS}}=0\text{V}$)

项目	符号	条件	MIN.	TYP.	MAX.	单位
从解除外部复位到完成初始设定通信为止的时间	t_{SUNIT}	在解除外部复位前，解除 POR 和 LVD 的复位。			100	ms
从将 TOOL0 引脚置为低电平到解除外部复位为止的时间	t_{SU}	在解除外部复位前，解除 POR 和 LVD 的复位。	10			μs
在解除外部复位后保持 TOOL0 引脚低电平的时间 (闪存固件处理时间除外)	t_{HD}	在解除外部复位前，解除 POR 和 LVD 的复位。	1			ms



- ① 给 TOOL0 引脚输入低电平。
- ② 解除外部复位（在此之前需要解除 POR 和 LVD 的复位）。
- ③ 解除 TOOL0 引脚的低电平。
- ④ 通过 UART 接收来完成波特率的设定。

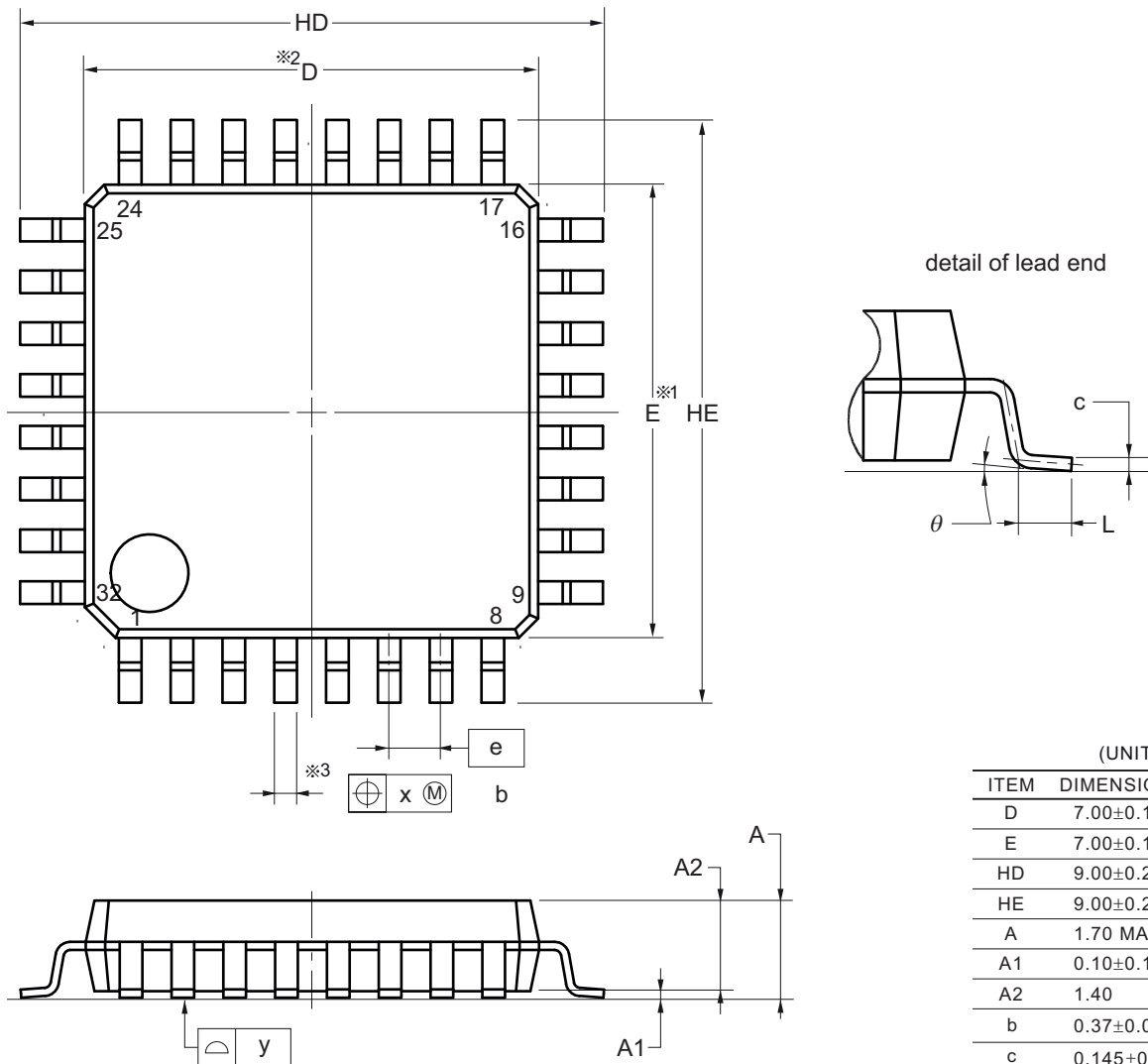
备注 t_{SUNIT} : 在此区间，必须在解除外部复位后的 100ms 之内完成初始设定的通信。

t_{SU} : 这是从将 TOOL0 引脚置为低电平到解除外部复位为止的时间。

t_{HD} : 这是在解除外部复位后保持 TOOL0 引脚低电平的时间（闪存固件处理时间除外）。

第 28 章 封装尺寸图

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

NOTE

1. Dimensions " $\ast 1$ " and " $\ast 2$ " do not include mold flash.
2. Dimension " $\ast 3$ " does not include trim offset.

附录 A 修订记录

Rev.	发行日	修订内容	
		页	修订处
1.11	2017.10.20	—	初版发行
1.20	2019.06.28	1 232 242	将 1.1 中的“ADC”更改为“A/D”。 在表 9-3 中追加注。 在图 10-4 中追加“或者在 A/D 转换结束时自动清除”。

R7F0C903-908
用户手册 硬件篇

Publication Date: Rev.1.11 Oct 20, 2017
Rev.1.20 Jun 28, 2019

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 Corporation**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351**Renesas Electronics Canada Limited**9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004**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 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, 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 200333, China
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 No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290**Renesas Electronics India Pvt. Ltd.**No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700**Renesas Electronics Korea Co., Ltd.**17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

R7F0C903-908



瑞萨电子株式会社

R01UH0629CJ0120

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [32-bit Microcontrollers - MCU category](#):

Click to view products by [Renesas manufacturer](#):

Other Similar products are found below :

[MCF51AC256AVFUE](#) [MCF51AC256BCFUE](#) [MCF51AC256BVFUE](#) [MB91F464AAPMC-GSE2](#) [R5S726B0D216FP#V0](#) [MB91F248PFV-GE1](#) [MB91243PFV-GS-136E1](#) [SAK-TC1782F-320F180HR BA](#) [TC364DP64F300WAAKXUMA1](#) [R5F566NNDDFP#30](#)
[R5F566NNDDFC#30](#) [R5F566NNDDBD#20](#) [MC96F8216ADBN](#) [A96G181HDN](#) [A96G140KNN](#) [A96G174FDN](#) [A31G213CL2N](#)
[A96G148KNN](#) [A96G174AEN](#) [AC33M3064TLBN-01](#) [V3s](#) [T3](#) [A40i-H](#) [V526](#) [A83T](#) [R11](#) [V851s](#) [A133](#) [V833](#) [F1C100S](#) [T3L](#) [T507](#) [A33](#)
[A63](#) [T113-i](#) [H616](#) [V853](#) [V533](#) [V536-H](#) [A64-H](#) [V831](#) [V3LP](#) [T113-S3](#) [F1C200S](#) [F133-A](#) [R128-S2](#) [ADUCM360BCPZ128-TR](#)
[APT32S003F8PT](#) [AT32F435VMT7](#) [AT32F435CGT7](#)